

## Introduction

- Navigation
- Supported Platforms For Local Development
- Requirements

## How it works

- Tech We Have Used
- Admin Dashboard (GraphQL & REST)
- Shop Frontend (GraphQL & REST)
- API

## Upgrade guide

- How to upgrade the existing deployed laravel 9 server to laravel 10?

## Getting Started

- route: /getting-started
- Installation Windows
- Prerequisites
  - Frontend
- Getting Started
  - For MAC and Linux(with sail and docker)
    - Prerequisites
    - Installation Mac
    - Video
    - REST API
    - GraphQL API
  - Getting Started with Frontend
    - NOTE: only for windows user ( if you want to use GraphQL version)
    - For Admin :
    - For Shop :
    - Shop config
    - Admin (At the root of the pickbazar directory, you can run the below commands)
    - Shop (At the root of the pickbazar directory, you can run the below commands)

## Dashboard Features

- A portion of order management. Order status change.
- Coupon Management
- Customer Management
- Tax Management
- Shipping Management
- Shipping Management
- Settings Management

## Available Scripts:

### Customization

- For customizing the template's default site settings:

### Styles and Assets

- CSS styles:
- Icons:
- For Adding a custom Icon:

## Utilities

### Backend Integration

- For API Integration:
  - Data Fetching
    - For GraphQL
    - Example:
      - Query
      - Mutation

For REST API

Example:

Query

Mutation

## Deployment

AWS (Amazon Web Service)

How to create ec2 server?

Domain Setup

Login to Server

Change permission .pem

Virtual Private Server (Automated Script)

Prerequisite

At first login your server from terminal

Server Environment setup script - RUN on Virtual Server

Nginx Setup And Settings - RUN on Virtual Server

Backend build - RUN on Virtual Server

Frontend build script - RUN on Local PC

Frontend run script - RUN on Virtual Server

Virtual Private Server (Manual)

Access Server

Install NodeJS & Required Application

Install NodeJS

Install Yarn

Install Zip & Unzip

Install PM2

Setup Server

Introduction

Step 1 - Installing Nginx

Step 2: Adjusting the Firewall

Step 3 – Checking your Web Server

Step 4 - Install MySQL

Step 5 - Install PHP & Composer

Step 6 - Create MySQL Database & User

Step 7 - Change permission for the `www` folder

Step 8 - Upload API to Server

Step 9: Setting Up Server & Project

Step 10 - Create New Nginx for the domain

For REST API

For GraphQL API

Secure Server

Step 1: Secure Nginx with Let's Encrypt

Install API

Step 1: Build and Run `api`

FrontEnd Project Build

Step 1 - Config Next Admin App For /admin Sub Directory

Step 2 - Install & Build

Step 3 - Build the project

For REST

For GraphQL

Build Project

Install FrontEnd And Run

Run frontend app

For REST API

For GraphQL API

Vercel

- vercel.com
- API
- Frontend
- cPanel
  - Access Server
  - Create Subdomains
  - Install API
  - Install Front-end
    - Front-end Project Build
    - Step 1 - Build Custom Server
      - shop rest
      - admin rest
      - admin graphql
      - shop graphql
    - Step 2 - Install & Build
    - Step 3 - Build the project
      - For REST
      - For GraphQL
  - Install Nodejs Project

### **PickBazar Update - Virutal Private Server**

- Step 1: Setup Git - Server
  - Install git
  - Config for first time
  - Prepare Git Repository
  - Git & Github
- Step 2: Shut Down Current Process
- Step 3: Local Repository & Updated Code
- Step 4: Update API
- Step 5: Front-end Project Build
  - Config Next Admin App For /admin Sub Directory
  - Install & Build
  - Build the project
  - Upload to GitHub
- Step 6: Upload Frontend & Run
  - Run frontend app

### **SEO**

- SEO
- Analytics

### **Laravel API**

- Getting Started
  - Installation Windows
  - Prerequisites
  - Resources you might need
  - Packages we have used
- Installation
  - For MAC and Linux(with sail and docker)
    - Prerequisites
    - Installation Mac
    - Video
    - REST API
    - GraphQL API
- Configuration
- Console Commands
- Development
- REST API
  - Endpoints

- Folder structure

- config

- database

- Enums

- Events

- Listeners

- Mail

- Notifications

- Providers

- stubs

- GraphQL API

- Laravel GraphQL API Endpoint

- Alternatives

- Folder Structure

- Payment Gateway

- Extending The Functionality

- Deployment

## **Messaging (Feature)**

### **Store Notice (Feature)**

- Super Admin

- Notice creation for specific vendor

- Vendor

- Store Notice Vendor profile

- Staff

- Create a Notice for particular shop

- Creation a notice for [furniture-shop]

- Issued notice for a specific shop [furniture-shop]

- Store Notice staff profile

- Staff for [furniture-shop]

### **Digital Wallet (Feature)**

- Sign Up Points

- Manually By Admin

- Refund

- Wallet Ratio

### **Social Login (Feature)**

- Google

- Facebook

### **OTP - Mobile Number Verification (Feature)**

- Configuration

- Twilio Configuration:

- Verify Geo Permissions

- MessageBird:

- FrontEnd Demo

- Login with Mobile Number:

- OTP Verification During Checkout Process:

- OTP Verification For Updating Mobile Number:

### **Email Verification (Feature)**

- How to turn on this feature

- Admin view of this feature

- Shop view of this feature

- Configuration

### **Translation (Feature)**

- Existing Language

- For Rest

- For GraphQL

- New Language
- Default Language
  - For Rest
  - For GraphQL

### **Multilingual (Feature)**

- Step 1: Enable Multilingual Feature
  - API:
  - Admin
  - Shop
- Step 2: How to add new language in admin?
- Step 3: How to add new language in shop?
- Step 3: How to translate static content for admin?
- Step 4: How to translate static content for shop?
- Step 5: How to translate content?
  - Data Type
  - Translate Settings
  - Translate Order Status
  - Translate Group, Author, Manufacturer, Tags, Categories:
  - Translate Products
    - Translate Simple Products:
    - Translate Variable Product:
  - Translate Attributes
  - Translate Groups
  - Translate Variable Products
- Step 6: How to add custom or solved broken font issue?

### **Multivendor (Feature)**

- Create New Shop
  - Dashboard:
  - Attributes:
  - Products:
  - Order:
  - Staff:
  - Withdraw:
- User Roles:
  - Super Admin:
  - Store Owner:
  - Staff:
- Withdraw Payment:
- FrontEnd Shop

### **Export Import (Feature)**

- Simple Product
- variable products

### **Settings Management**

- Change Logo
  - API
- Change Site information
  - Shipping class, Tax class & Free Shipping
- Payment Gateway Management
  - Super Admin
  - Customer
- SEO
- Add or Remove Delivery Schedule
- Shop Settings & Google Map

### **Payment**

- Stripe
  - Stripe integrate inside PickBazar.

[How to create & setup Stripe information properly?](#)

[Special Notes for Stripe users.](#)

[How can I add card in my user profile for future payments in Stripe?](#)

#### Stripe Element

[Stripe Element integrate inside PickBazar.](#)

[How to create & setup Stripe Element information properly?](#)

[How can I add stripe element's another payment gateway?](#)

[How to customize your stripe element payment method.](#)

[Special Notes for Stripe element users.](#)

[Available Stripe Element Payment Method](#)

#### Xendit

[Xendit integrate inside PickBazar.](#)

[Xendit Webhook settings](#)

[How to create & setup Xendit information properly?](#)

[Special Notes for Xendit.](#)

#### SSLCOMMERZ

[SSLCOMMERZ integrate inside PickBazar.](#)

[How to create & setup SSLCOMMERZ information properly?](#)

#### RazorPay

[RazorPay integration inside PickBazar.](#)

[How to create & setup RazorPay information properly?](#)

[Special Notes for RazorPay.](#)

#### Paystack

[Paystack integrate inside PickBazar.](#)

[Paystack Webhook settings](#)

[How to create & setup Paystack information properly?](#)

[Special Notes for Paystack.](#)

#### PayPal

[PayPal integrate inside PickBazar.](#)

[How to create & setup PayPal information properly?](#)

#### Mollie

[Mollie integrate inside PickBazar.](#)

[Mollie Webhook settings](#)

[How to create & setup Mollie information properly?](#)

[Special Notes for Mollie.](#)

#### Iyzico

[Iyzico integrate inside PickBazar.](#)

[How to create & setup Iyzico information properly?](#)

#### Multiple payment gateway

[How to activate multiple payment gateways \[ in Admin \] ?](#)

[Description \[ sequence by number in the screenshot \]](#)

[How to use multiple payment gateways from customer end \[ in Shop \] ?](#)

[Select a payment gateway during place order](#)

[Select different payment gateway \( if needed \)](#)

[Description \[ sequence by number in the screenshot \]](#)

#### Integration of new payment gateway

[Getting Started with API](#)

[Step 1: Install and configure the payment gateway package](#)

[Step 2: Add payment gateway name in the Enum.](#)

[Step 3: Configure the Payment Facade for the new payment gateway.](#)

[Step 4: Using that payment gateway for submitting the order.](#)

[Step 5: How to Setup webhook in pickbazar-laravel follow the steps](#)

[Getting Started with admin dashboard.](#)

[Getting Started with shop front.](#)

[Redirect-base Payment Gateway](#)

[Non-redirect based payment gateway](#)

## bKash

bKash integrate inside PickBazar.

How to create & setup bKash information properly?

Special Notes for bKash.

## Paymongo

Paymongo integrate inside PickBazar.

How to create paymongo webhook url?

You can visit for paymongo.

## Flutterwave

Flutterwave integrate inside PickBazar.

Flutterwave Webhook settings

### How to create & setup Flutterwave information properly?

Special Notes for Flutterwave.

## Support Portal

### FAQs

How to configure Stripe payment gateway?

Why am I facing "You may need an appropriate loader to handle this file type" during running shop rest?

I am changing schema files but changes is not working

Changing .env files but not getting the changes

Changing route but not getting the changes.

I have set `STRIPE_API_KEY` in .env but still getting error.

Can I use it with my existing laravel?

Why am I getting `Access denied for user?`

Why am I getting permission issue during deployment?

Why am I getting "The GET method is not supported for this route. Supported methods: HEAD"?

How to resolve the `Load More` `Infinity` loading issue?

I'm trying to upload images, but the images are not displayed on the frontend?

API

Admin

Shop

Build Frontend,

How to resolve the `502 Bad Gateway` error for frontend/admin?

Reason 1:

PORT:

Reason 2:

How to resolve `javascript heap out of memory` issue?

Image upload throw Internal Server Error; how to resolve that?

How to resolve docker `invalid reference format: repository name must be lowercase?`

Sometimes my server shutdown or halts automatically. After restart, it works fine again; how to resolve that?

S3 uploading doesn't work after adding credentials to .env; how to resolve that?

How to rebuild the project?

How to increase upload size?

How to resolve `docker: invalid reference format: repository name must be lowercase.?`

How to remove the existing payment gateway?

Why checkout `Place Order` button is disabled?

How do I log in to the admin panel as the main administrator?

How to upgrade the existing deployed laravel 8 server to laravel 9?

### Coupon

Super Admin

Customer

### Find or Search Nearby shops inside PickBazar.

Admin Settings

### OpenAI

OpenAI Integration inside PickBazar.

Admin Settings  
Special Notes for Paystack.

#### Question

Super Admin  
Customers

#### Review

Customer

#### Shipping

Free Shipping  
Cash on delivery  
Shipping and Billing Address  
Shipping process ingeneral  
Super Admin  
Vendor

#### Taxes

Super Admin  
Vendor

#### Withdrawals

Vendor  
Super Admin

## Introduction

---

Welcome to the PickBazar documentation! Fastest Ecommerce App built with React, NextJS, TypeScript, Laravel(GraphQL, REST API) and Tailwind.

## Navigation

---

You can find different topics in the table of contents. On desktop, you should see it in the left sidebar. On mobile, you should see it after pressing an icon with Hamburger in the top right corner.

## Supported Platforms For Local Development

---

- Compatible Browsers (Firefox, Safari, Chrome, Edge)
- Server(Node.js 16.15.1 or later)
- MacOS, Windows, and Linux are supported

## Requirements

---

- node(16.15.1 or later)
- yarn
- PHP 8.1
- MySQL 8
- ext\_curl
- editor: [Visual Studio Code](#)(recommended)

# How it works

---

## Tech We Have Used

---

We have used monorepo folder structure with Yarn Workspace. In our app we have two different services:

- api
- shop
  - shop-graphql
  - shop-rest.
- admin
  - admin-graphql
  - admin-rest

Tech specification for specific part is given below:

## Admin Dashboard (GraphQL & REST)

---

- [Nextjs](#)
- [Typescript](#)
- [Tailwindcss](#)
- [React Hook Form](#)
- [Apollo Client](#)(For GraphQL API version)
- [React-Query](#)(For REST API version)

## Shop Frontend (GraphQL & REST)

---

- [Nextjs](#)
- [Typescript](#)
- [Tailwindcss](#)
- [React Hook Form](#)
- [Apollo Client](#)(For GraphQL API version)
- [React-Query](#)(For REST API version)

## API

---

- Laravel

## Upgrade guide

---

# How to upgrade the existing deployed laravel 9 server to laravel 10?

You can check older laravel version upgrade guides [here](#)

At first, remove all the existing `php 8.0` and its extensions by using this command,

```
sudo apt purge php8.0-fpm php8.0-mysql
sudo apt purge php8.0-mbstring php8.0-xml php8.0-bcmath php8.0-simplexml php8.0-intl php8.0-mbstring php8.0-gd php8.0-curl php8.0-zip composer
```

Then remove the composer,

```
sudo rm /usr/bin/composer
```

Then delete the `vendor` folder from the `pickbazar-laravel -> api` folder.

```
cd /var/www/pickbazar-laravel/api

sudo chown -R $USER:www-data storage
sudo chown -R $USER:www-data bootstrap/cache

rm vendor -rf
```

Then install PHP 8.1, and its extensions,

```
sudo add-apt-repository ppa:ondrej/php
sudo apt update
sudo apt install php8.1-fpm php8.1-mysql
sudo apt install php8.1-mbstring php8.1-xml php8.1-bcmath php8.1-simplexml
php8.1-intl php8.1-gd php8.1-curl php8.1-zip php8.1-gmp
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'e21205b207c3ff031906575712edab6f13eb0b361f2085f1f1237b7126d785e826a450292b6cfd1d
64d92e6563bbde02') { echo 'Installer verified'; } else { echo 'Installer
corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"
sudo mv composer.phar /usr/bin/composer
```

Then update 8.0 fpm to 8.1 fpm from,

```
/etc/nginx/sites-enabled/pickbazar
```

```

# For API
location /backend {
    alias /var/www/pickbazar/api/public;
    try_files $uri $uri/ @backend;
    location ~ /\.php$ {
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $request_filename;
        fastcgi_pass    unix:/run/php/php8.1-fpm.sock;
    }
}

location @backend {
    rewrite /backend/(.*)$ /backend/index.php?/$1 last;
}

# For FrontEnd → GraphQL
location /{
    proxy_pass http://localhost:3000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}

location /admin{
    proxy_pass http://localhost:3004/admin;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}

error_page 404 /index.php;

location ~ /\.php$ {
    fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
}

```

Make sure you didn't introduce any syntax errors by typing:

sites-enabled.png

```
sudo nginx -t
```

Next, restart Nginx:

```
sudo systemctl restart nginx
```

Then install composer packages,

```
cd /var/www/pickbazar-laravel/api  
composer install  
php artisan optimize:clear  
php artisan marvel:install
```

php artisan marvel:install will remove all of your existing data. Ensure you export or backup your data before using that command.

```
sudo chown -R www-data:www-data storage  
sudo chown -R www-data:www-data bootstrap/cache
```

## Getting Started

---

name: Getting Started

### route: /getting-started

---

For getting started with the template you have to follow the below procedure. For quick guide you can check below videos for installation.

## Installation Windows

---

### Prerequisites

---

- PHP 8.1
- Composer
- Xamp/Wamp/Lamp for any such application for apache, nginx, mysql
- PHP plugins you must need
  - simplexml
  - PHP's dom extension
  - mbstring
  - GD Library

# Frontend

- node(16.15.1 or later)
- yarn
- editor: [Visual Studio Code](#)(recommended)

## Getting Started

1. First download the file from codecanyon.
2. Unzip the downloaded file and folder structure you get

```
pickbazar
|-- api
|-- admin
|   |-- rest
|   |-- graphql
|-- shop
```

3. From the above folder structure you should notice that our app has three parts `api` , `shop` and `admin`. So you have to run all the parts separately and sequentially.

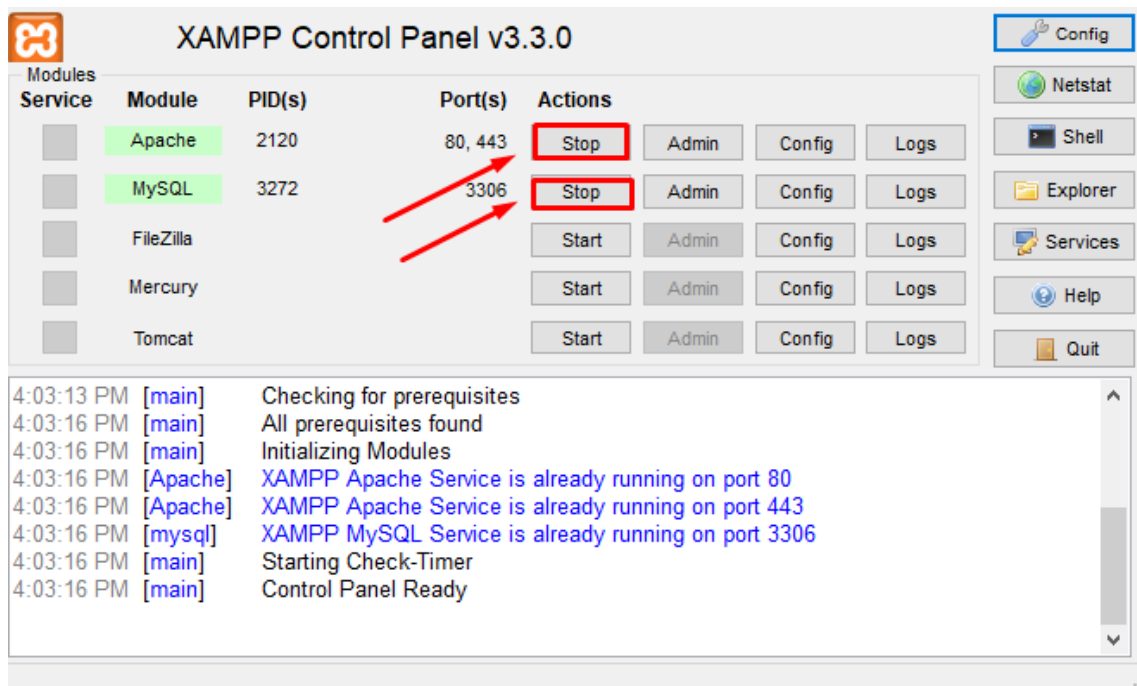
### Getting started with api

- Make sure you have run xamp/mamp/wamp/lamp for mysql and php by following the steps below:

Click on action button start to make it start.



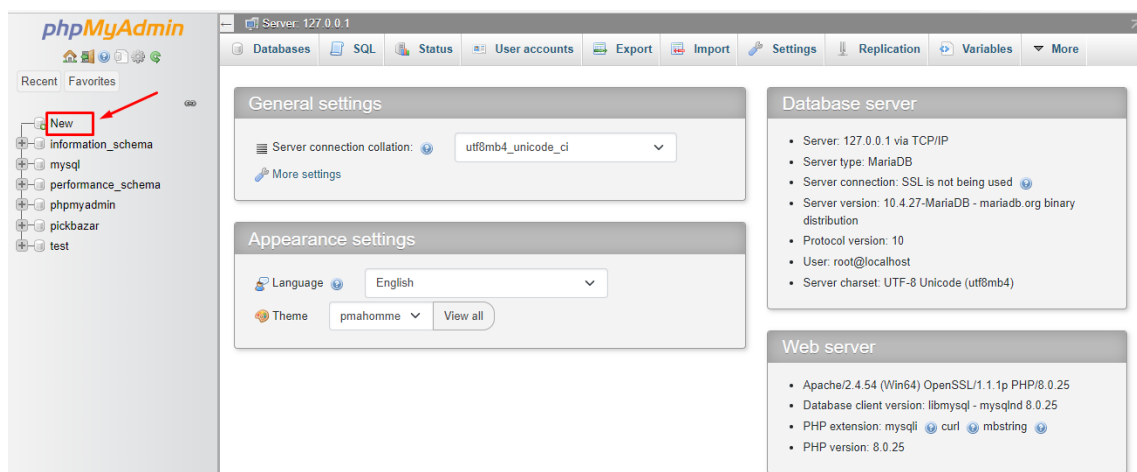
Xamp has been started.



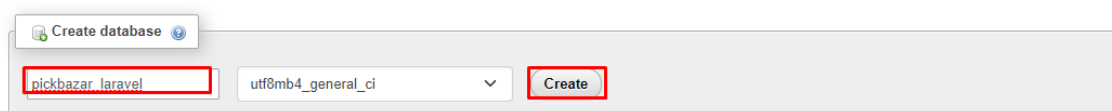
- Go to localhost/phpmyadmin

localhost/phpmyadmin/

- Click on new to create new database.



- Create a database in your mysql and put those info in next step.



- Inside api folder rename .env.example file to .env and provide necessary credentials. Like database credentials, stripe credentials, s3 credentials(only if you use s3 disk) admin email shop url etc. Specially check for this `env` variables

```

DB_HOST=localhost
DB_DATABASE=pickbazar_laravel
DB_USERNAME=root
DB_PASSWORD=
  
```

- Run `composer install` using terminal

```
► composer install
Installing dependencies from lock file (including require-dev)
Verifying lock file contents can be installed on current platform.

Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: barryvdh/laravel-dompdf
Discovered Package: bensampo/laravel-enum
Discovered Package: cviebrock/eloquent-sluggable
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: fruitcake/laravel-cors
Discovered Package: ignited/laravel-omnipay
Discovered Package: intervention/image
Discovered Package: laravel/legacy-factories
Discovered Package: laravel/sail
Discovered Package: laravel/sanctum
Discovered Package: laravel/tinker
Discovered Package: mill-lab/laravel-graphql-playground
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Discovered Package: nuwave/lighthouse
Discovered Package: pickbazar/shop
Discovered Package: prettus/l5-repository
Discovered Package: spatie/laravel-medialibrary
Discovered Package: spatie/laravel-permission
Package manifest generated successfully.
95 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
```

- run `php artisan key:generate`

```
► php artisan key:generate

Application key set successfully.
```

- Run `php artisan marvel:install` and follow necessary steps.

```
Installing Pickbazar Dependencies...

Do you want to migrate Tables? If you have already run this command or migrated tables then be aware, it will erase all of your data. (yes/no) [no]:
> yes

Migrating Tables Now....
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (79.70ms)
Migrating: 2014_10_12_100000_create_password_resets_table (58.57ms)
Migrated: 2014_10_12_100000_create_password_resets_table (58.57ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table (79.56ms)
Migrated: 2019_08_19_000000_create_failed_jobs_table (79.56ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table (143.06ms)
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (143.06ms)
Migrating: 2020_04_17_194830_create_permission_tables (839.79ms)
Migrated: 2020_04_17_194830_create_permission_tables (839.79ms)
Migrating: 2020_06_02_051901_create_pickbazar_tables (1,601.56ms)
Migrated: 2020_06_02_051901_create_pickbazar_tables (1,601.56ms)
Migrating: 2020_10_26_163529_create_media_table (71.72ms)
Migrated: 2020_10_26_163529_create_media_table (71.72ms)
Tables Migration completed.

Do you want to seed dummy data? (yes/no) [no]:
> yes

Copying necessary files for seeding....
File copying successful
Seeding....
Seed completed successfully!

Do you want to create an admin? (yes/no) [no]:
> no

Copying resources files...
Installation Complete
```

- For image upload to work properly you need to run `php artisan storage:link`.

```
The [/var/www/html/public/storage] link has been connected to [/var/www/html/storage/app/public].
The links have been created.
```

- run `php artisan serve`

```
▶ php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
[Sat Apr 10 16:35:26 2021] PHP 8.0.0 Development Server (http://127.0.0.1:8000) started
```

NB: your frontend `NEXT_PUBLIC_GRAPHQL_API_ENDPOINT` env value will be

`localhost:8000/graphql`

NB: your frontend `NEXT_PUBLIC_REST_API_ENDPOINT` env value will be `localhost:8000/`

## For MAC and Linux(with sail and docker)

There is an alternate installation procedure for linux and mac. You can follow below procedure to getting started with `sail`

### Prerequisites

- Docker

NB: Move pickbazar-laravel folder from Pickbazar Laravel - React Next Rest & GraphQL Ecommerce With Multivendor folder

### Installation Mac

#### Video

#### REST API

Follow this tutorial for install REST API

#### GraphQL API

Follow this tutorial for install GraphQL

- Run Docker application first
- Now go to your pickbazar-laravel root directory and run `bash install.sh`. It will guide you through some process. Follow those steps carefully and your app will be up and running
- Navigate to `api` then `sail down` to stop the container. If you want to remove the volumes then `sail down -v`

NB: your frontend `NEXT_PUBLIC_GRAPHQL_API_ENDPOINT` env value will be

`localhost/graphql`

NB: your frontend `NEXT_PUBLIC_REST_API_ENDPOINT` env value will be `localhost/`

For details api doc and requirements details you can go to [Laravel API](#)

## Getting Started with Frontend

4. After configuring API & running it successfully you can choose the directory where you need to work

## NOTE: only for windows user ( if you want to use GraphQL version)

- Go to your specific project root and find ``.graphql-let.yml``.
- To find the file go to ``admin` => `graphql` => `.graphql-let.yml``
- Replace the env variable ``${NEXT_PUBLIC_GRAPHQL_API_ENDPOINT}`` in schema field manually
- Provide your API url in the schema field.

```
"schema: ${NEXT_PUBLIC_GRAPHQL_API_ENDPOINT}",  
to  
"schema: https://localhost:8000/graphql",
```

- Also change the dev command at ``package.json``
- Go to ``shop/package.json`` file and change the scripts:

```
"codegen": "node -r dotenv/config $(yarn bin)/graphql-let",  
to  
"codegen": "graphql-let",
```

Below are the directories where you will choose to work for frontend stuffs

```
# For Admin GraphQL  
cd admin/graphql  
# For Admin REST  
cd admin/rest  
# For Shop GraphQL or REST  
cd shop
```

After choosing your working directory

**Go to specific folder and rename the `.env.template` => `.env` and put your api endpoint here. You will find `.env.template` file at the root of your `admin/{chosen-directory-name}` or `shop`**

5. Run yarn at the root directory.

```
# on pickbazar/root directory  
yarn
```

## 6. Scripts To Run the fronted App


### For Admin :

For starting the admin dashboard part with corresponding api data run below commands.

- using workspace (At the root of the pickbazar directory, you can run the below commands)

```
# for dev mode run below command  
# GraphQL  
yarn dev:admin-gql  
  
# REST  
yarn dev:admin-rest
```

```
► yarn dev:admin-rest
yarn run v1.22.5
$ yarn workspace @pick-bazar/admin-rest dev
$ next dev -p 3002
ready - started server on 0.0.0.0:3002, url: http://localhost:3002
info - Loaded env from /Users/bashar/Codes/pickbazar-sail/frontend/admin-rest/.env.local
event - compiled successfully
```



- without workspace(if you want to run the command within specific project root of `admin/{chosen-directory-name}`)

```
# for dev mode run below command
# GraphQL
yarn dev

# REST
yarn dev
```

This command will run the app in development mode. Open the suggested url in your terminal.  
like => <http://localhost:3000>.

**\*\* Note: \*\***

- The page will automatically reload if you make changes to the code. You will see the build errors and lint warnings in the console.
- If you saw any error while running make Sure you setup your API endpoint properly at `.env` file.

## For Shop :

### Shop config

From `v4.0.0`, we changed the structure of the `shop` directory. So if you're using `v4.0.0` or later, then you've to do a little bit of customization of the `shop` directory.

At first open `pickbazar -> shop`,

- For `REST`, copy `tsconfig.rest.json` content to `tsconfig.json`  
or
- For `GraphQL` copy `tsconfig.graphql.json` content to `tsconfig.json`

Then open `pickbazar -> shop -> .env` and change `FRAMEWORK_PROVIDER` to `rest` for REST installation or `graphql` for Graphql Installation.


Then to starting the shop part with corresponding api run below commands.

- using workspace (At the root of the pickbazar directory, you can run the below commands)

```
# for dev mode run below command
# GraphQL
yarn dev:shop-gql

# REST
yarn dev:shop-rest
```

```
► yarn dev:shop-rest
yarn run v1.22.5
$ yarn workspace @pick-bazar/shop-rest dev
$ next dev -p 3003
ready - started server on 0.0.0.0:3003, url: http://localhost:3003
info - Loaded env from /Users/bashar/Codes/pickbazar-sail/frontend/shop-rest/.env.local
```



- without workspace(if you want to run the command within specific project root of `shop`)

```
# for dev mode run below command
# GraphQL
yarn dev

# REST
yarn dev
```

**\*\* If you want to test your production build admin or shop in local environment then run the below commands. \*\***

### **Admin (At the root of the pickbazar directory, you can run the below commands)**

```
# build admin for production
yarn build:admin-gql

#start admin in production mode
yarn start:admin-gql
```

### **Shop (At the root of the pickbazar directory, you can run the below commands)**

```
# build shop for production
yarn build:shop-gql

# start shop in production mode
yarn start:shop-gql
```

```

▶ yarn build:shop-rest
yarn run v1.22.5
$ yarn workspace @pick-bazar/shop-rest build
$ next build
info - Loaded env from /Users/bashar/Codes/pickbazar-sail/frontend/shop-rest/.env.local
info - Creating an optimized production build
info - Compiled successfully
info - Collecting page data
info - Generating static pages (48/48)
info - Finalizing page optimization

Page                                     Size      First Load JS
┌ ────────────────────────────────────┴──────────────────┴──────────────────┐
│ /_app                               0 B        176 kB
│ └─ [Type]                           27.2 kB    268 kB
│   └─ css/7936e7aa084bd830e87c.css   4.01 kB
│   └─ /grocery
│   └─ /makeup
│   └─ /bags
│   └─ [+2 more paths]
│   └─ /404
│   └─ /bakery (ISR: 60 Seconds)       2.66 kB    243 kB
│     └─ css/0dcfbd7860d56385c547.css  4.01 kB
│   └─ /change-password                3.25 kB    191 kB
│   └─ /checkout                       4.85 kB    189 kB
│   └─ /contact                        2.83 kB    191 kB
│   └─ /example                        6.71 kB    183 kB
│   └─ /help                           2.75 kB    182 kB
│   └─ /logout                         4.99 kB    181 kB
│   └─ /offers                         9.81 kB    189 kB
│   └─ /order                          15.4 kB    209 kB
│     └─ css/db04274d0e1caa11303a.css  351 B
│   └─ /order-received/[tracking_number] 6.85 kB    186 kB
│   └─ /orders                         28.5 kB    208 kB
│     └─ css/da2aebf602ff72a3fc0.css   2.1 kB
│   └─ /privacy                       1.99 kB    188 kB
│   └─ /products/[slug]               7.33 kB    211 kB
│     └─ css/efb94a24c22d59f4d562.css  4.17 kB
│     └─ /products/apples
│     └─ /products/baby-spinach
│     └─ /products/blueberries
│     └─ [+27 more paths]
│   └─ /profile                       11.8 kB    200 kB
│   └─ /terms                         1.98 kB    188 kB
│   └─ /ui                            8.82 kB    185 kB
└─ First Load JS shared by all
  └─ 176 kB
    └─ chunks/00682edc09922f9a0564b6d51ebfe3fe9fe3f6bc.837c2b.js  3.14 kB
    └─ chunks/03597eb9de95bb8e6991ab21a5b170a1bc330ab.CSS.3bbe0c.js  68 B
    └─ chunks/1c2031c8_c533da.js  23.5 kB
    └─ chunks/2c96be8e9bf36e109a607a23c3ac04c26631eb28_f78ccb.js  4.76 kB
    └─ chunks/57ad4ee9d685f4a85f091607d3121ddc66f6f331_d3ef7d.js  35.8 kB
    └─ chunks/6bc2a55fc5fb5a7f77a0d800e4657cca465fac5a_d73fc5.js  7.14 kB
    └─ chunks/78478e0567901f0a00137009c7f6751fab214ebc.855475.js  15.3 kB
    └─ chunks/aa9552d4f74729e02303abd0a7d42186859f2ec0_37727a.js  7.43 kB
    └─ chunks/commons.ff7e9e.js  18.2 kB
    └─ chunks/framework.f6e4f9.js  42.3 kB
    └─ chunks/main.9c0e07.js  6.29 kB
    └─ chunks/pages/_app.558fda.js  9.76 kB
    └─ chunks/webpack.3d3782.js  2.3 kB
    └─ css/669ac54fd96e08d6ac5.css  4.38 kB
    └─ css/fs3121b36d6036aac827.css  11.2 kB

λ (Server) server-side renders at runtime (uses getInitialProps or getServerSideProps)
o (Static) automatically rendered as static HTML (uses no initial props)
• (SSG) automatically generated as static HTML + JSON (uses getStaticProps)
  (ISR) incremental static regeneration (uses revalidate in getStaticProps)

*+ Done in 96.20s.
▶ yarn start:shop-rest
yarn run v1.22.5
$ yarn workspace @pick-bazar/shop-rest start
$ next start -p 3003
ready - started server on 0.0.0.0:3003, url: http://localhost:3003
info - Loaded env from /Users/bashar/Codes/pickbazar-sail/frontend/shop-rest/.env.local

```

**\*\* Note \*\*:**





Please see `package.json` file for other builtin helper commands.

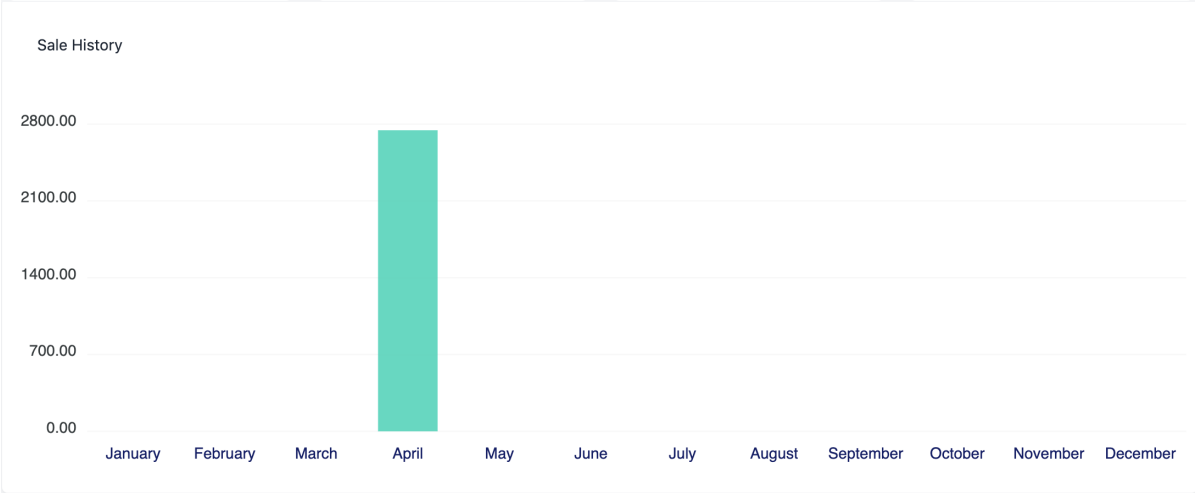
- For development purpose we use yarn workspace if you want to use it then see the `package.json` file at root, for various workspace specific command.
- if you prefer single template then just copy the required template folder and separate them. you'll find their `package.json` file within them and follow the command for dev, build, start.
- For further development & customization check our [Frontend Customization](#) guide.

# Dashboard Features

## • Analytics Dashboard

You will get a complete analytics dashboard to know the overview of your shop.

<b>Total Revenue</b> (Last 30 Days) 	<b>Total Order</b> (Last 30 Days) 	<b>New Customer</b> (Last 30 Days) 	<b>Todays Revenue</b> 
<b>\$7,844.05</b>	<b>25</b>	<b>16</b>	<b>\$0.00</b>



Recent Orders

Tracking Number	Total	Order Date	Status
GkszijsVAcae	\$2,744.33	5 days ago	Order Received
lrrdghqshkys	\$1,748.30	14 days ago	Order Dispatched
iTUVhT87QdE8	\$526.98	17 days ago	Order Dispatched
A9mvTnzhGxkc	\$294.49	17 days ago	Order Received
BlOnTtmsHHHE	\$12.80	17 days ago	Order Received
ruIb1kz4UIIN	\$2.45	19 days ago	Shipment Refused by Consignee
NqlykeIW6awe	\$97.80	18 days ago	Delivered
TfzYmId1AR32	\$18.00	18 days ago	Ready To Dispatch
PXqs8RXQBGm5	\$7.20	19 days ago	Delivered
K2isul12Cetf	\$540.00	19 days ago	Ready To Dispatch

## Popular Products

ID	Name	Type	Price/Unit	Quantity
4	Brussels Sprout		\$5.00	17
3	Blueberries		\$3.00	30
2	Baby Spinach		\$0.60	10
6	Clementines		\$3.00	50
5	Celery Stick		\$6.00	18
7	Sweet Corn		\$5.00	50
8	Cucumber		\$2.50	25
103	Armani Purse		\$80.00	50
9	Dates		\$10.00	50
102	Armani Leather Purse		\$50.00	50

## • Store Notice

Store Notice allows you to enter a site-wide message shown to everyone visiting your store. Send messages to vendors or for the multivendor website. The message can be: Informational - **A note that tells customers of shipping delays due to weather conditions.** In **Pickbazar** you can send messages to vendors or all customers with the colored base priority. Admin can set notice for specific vendor or all vendors. Vendors can set notice for a specific shop or all shops independently. Notices are set for the shop will be visible to everyone visiting that shop.

- Store Notice Hierarchy
  - Admin ⇒ Vendor/All Vendor/Selected Vendors
  - Vendor ⇒ Shop
- Priority
  - High (Blue)
  - Medium (Yellow)
  - Low (Light Yellow)

# Store Notice integration inside PickBazar

## Store Notice admin panel

Store Notices								
<div><div>Type your query and press enter</div><div>+ Add Store Notice</div></div>								
ID	Notice	Description	Type	Effective From	Effective At	Issued By	Priority	Actions
22	FSG	Prepare for FSG's approval (which approval ...	All vendor	a day ago	in 11 days	Super admin	LOW	
21	A letter for each of the ven...	A letter for each of the vendors of the Servic...	All vendor	a day ago	in 4 days	Super admin	MEDIUM	
20	Notice to Vendors	NMS shall notify each Vendor servicing GEC...	All vendor	a day ago	in 14 days	Super admin	HIGH	
<div>&lt; 1 &gt;</div>								

## Notice creation for specific vendor

Description  
Add Store Notice necessary information from here

Priority\*

Medium

Store Notice\*

Notice to Vendors

Description

NMS shall notify each Vendor servicing GECS of new enrolments and of the date on which coverage will be effective via electronic file transfer if possible. If a Vendor is unable to accept an electronic file or a smaller number of Company's Employees are enrolled, new enrollment information will be sent via a paper enrollment report. NMS shall use its best efforts to assure that the Vendors receive notification of such enrollment as soon as possible. NMS shall provide notice, in the first weekly transmission of eligibility information after NMS receipt of information, to the appropriate Vendor when new enrolments occur.

Active

20/01/2023

Expire

25/01/2023

Type\*

Specific Vendor

Received By\*

Select...

Add Store Notice

## Store Notice Vendor profile

Notices for vendor from admin will appear here. Expired notices will disappear from here automatically.

My Shops

Message

Store Notice

Notice to Vendors

HIGH

NMS shall notify each Vendor servicing GECS of new enrollments and of the date on which coverage will be effective via electronic file transfer if possible. If a Vendor is unable to accept an electronic file or a smaller number of Company's Employees...

[Read more](#)

A letter for each of the vendors

MEDIUM

A letter for each of the vendors of the Service Contracts in the form attached hereto as Exhibit "O," duly executed by Seller, dated as of the Closing Date and addressed to each Service Contract vendor, informing said vendors of the assignment of the...

[Read more](#)

FSG

LOW

Prepare for FSG's approval (which approval shall not be unreasonably withheld or delayed) a written notice to be delivered to all vendors of FSG notifying such vendors of the termination of this Agreement...

[Read more](#)

## Create a Notice for shop

Creation a notice for [furniture-shop]

Create Store Notice

Description  
Add Store Notice necessary information from here

Priority\*  
Medium

Store Notice\*  
Offer

Description  
Buy one Get one Offer from 20/01/2023 to 25/01/2023.

Active  
20/01/2023

Expire  
25/01/2023

Add Store Notice


## Issued notice for a specific shop [furniture-shop]

Store Notices								
<input type="text" value="Type your query and press enter"/> <a href="#">+ Add Store Notice</a>								
ID	Notice	Description	Type	Effective From	Effective At	Issued By	Priority	Actions
23	Offer	Buy one Get one Offer from 20/01/2023 to ...	Specific shop	2 hours ago	In 5 days	Store owner	MEDIUM	<a href="#">Delete</a> <a href="#">Edit</a>

< 1 >


## Store Notice staff profile

Staff for [furniture-shop]

  
**Staff**  
store\_staff@demo.com  
Please add your [Profile](#)

☒ Enabled

My Shops Message **Store Notice**



















 Offer **MEDIUM**

Buy one Get one Offer from 20/01/2023 to 25/01/2023...

[Read more](#)

## • Manage Product Type

In **Types** menu you will get the product types and you can add, remove or modify product type from there.

Types			
		<input type="text" value="Type your query and press enter"/>	<a href="#">+ Add Types</a>
ID	Name	Icon	Actions
6	Furniture		 
5	Clothing		 
4	Bags		 
3	Makeup		 
2	Bakery		 
1	Grocery		 
































## • Manage Product Category

In **Categories** menu you will get the product types and you can add, remove or modify product categories from there.

Categories

Type your query and press enter

+ Add Categories

	ID	Name	Details	Image	Icon	Slug	Type	Actions
	82	Skirts				skirts	Clothing	 
	113	Table				table	Furniture	 
	109	Sofa				sofa	Furniture	 
	105	Chair				chair	Furniture	 
	101	Bed				bed	Furniture	 
	100	Soft Bread				soft-bread	Bakery	 
	99	Toast & Loaf				toast-loaf	Bakery	 
	98	Feeter & Pies				feeter-pies	Bakery	 
	97	Croissants				croissants	Bakery	 

## • Product Management

In **Products** menu you will get the products and you can add, remove or modify products from there.

Add Products

Products

Type your query and press enter

ID	Image	Name	Type	Price/Unit	Quantity	Description	Status	Actions
400		Absolute Plus Meat Denta...	Grocery	\$10.00	50	Oral hygiene is the practice o...	<div>publish</div>	<div></div> <div></div>
401		Dentek Oral Care Kit	Grocery	\$10.00	50	Oral hygiene is the practice o...	<div>publish</div>	<div></div> <div></div>
402		Gum Oral Care Cleaning Kit	Grocery	\$15.00	50	Oral hygiene is the practice o...	<div>publish</div>	<div></div> <div></div>
407		Pacific Natural Shaving Cr...	Grocery	\$6.00	50	Shaving is the removal of hai...	<div>publish</div>	<div></div> <div></div>
408		Longmarket Barber Afters...	Grocery	\$6.00	40	Shaving is the removal of hai...	<div>publish</div>	<div></div> <div></div>
409		Human Nature Shaving Cr...	Grocery	\$6.00	50	Shaving is the removal of hai...	<div>publish</div>	<div></div> <div></div>
410		Espa Men Dual Action	Grocery	\$6.00	50	Shaving is the removal of hai...	<div>publish</div>	<div></div> <div></div>

## A portion of product form

Featured Image

Upload your product featured image here

Upload an image or drag and drop

PNG, JPG

Gallery

Upload your product image gallery here

Upload an image or drag and drop

PNG, JPG

Type & Categories

Select product type and categories form here

Type\*

Grocery

Categories

Fruits & Vegetables x

Fruits x

x

## • Order Management

In **Order** menu you will get the order list and you can add, remove or modify order from there.

Orders						
<input type="text" value="Type your query and press enter"/>						
Tracking Number	Delivery fee	Total	Order Date	Status	Shipping Address	Actions
GkszijsVAcae	\$50.00	\$2,744.33	6 days ago	Order Received	22, 149/2A, Mohakhali DOHS, Dhaka 1206, Bangladesh, Illinois, Dhaka	
Irrdghqshkys	\$50.00	\$1,748.30	14 days ago	Order Dispatched	22, 149/2A, Mohakhali DOHS, Dhaka 1206, Bangladesh, Illinois, Dhaka	

## A portion of order management. Order status change.

Order ID - GkszijsVAcae

Order status

Change Status

✓

2

3

Order Received

Order Processing

Ready To

At Local Facility

Out For Delivery

Failed to collect payment

failed to contact Consignee

Shipment Refused by Consignee

Delivered








5

At Local Facility

Products	Total
Bourjois Little Round Pot Blusher x 1	\$8.00
Brussels Sprout x 1	\$3.00
Celery Stick x 1	\$5.00
Armani Leather Purse x 46	\$1,840.00






## Coupon Management


In **Coupon** menu you will get the Coupon list and you can add, remove or modify Coupon from there.

Coupons						
		<input type="text" value="Type your query and press enter"/>				<input type="button" value="+ Add Coupon"/>
ID	Banner	Code	Amount	Active	Expired	Actions
3		EID12	\$12.00	15 days ago	in a month	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
4		HELLO15	\$15.00	15 days ago	in 12 days	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
5		BAZAR18	\$18.00	15 days ago	in 4 days	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
6		BAZAR20	\$20.00	15 days ago	in a year	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
7		BAZAR10	\$10.00	14 days ago	in a month	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
10		4OFF	\$4.00	14 days ago	14 days ago	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
9		5OFF	\$5.00	14 days ago	in 2 months	<input type="button" value="Delete"/> <input type="button" value="Edit"/>

## Customer Management

In **Customer** menu you will get the Customer list and you can add, remove or modify Customer from there.

Customers			
		<input type="text" value="Type your query and press enter"/>	<input type="button" value="+ Add Customer"/>
Avatar	Name	Status	Actions
	again@another.net	Active	<input type="button" value="Block"/>
	abcd@dd.io	Inactive	<input type="button" value="Unblock"/>
	Litan	Active	<input type="button" value="Block"/>
	III	Active	<input type="button" value="Block"/>
	Pobon Paul	Active	<input type="button" value="Block"/>



**Block Customer**

Are you sure you want to block this customer?

## Tax Management

In **Tax** menu you will get the Tax list and you can add, remove or modify Tax from there.

Taxes							
<div> <input type="text" value="Type your query and press enter"/> <input type="button" value="+ Add Tax"/> </div>							
ID	Name	Rate (%)	Country	City	State	ZIP	Actions
1	Global	2					<input type="button" value="Delete"/> <input type="button" value="Edit"/>

## Shipping Management

In **shipping** menu you will get the shipping list and you can add, remove or modify shipping from there.

Shippings					
<div> <input type="text" value="Type your query and press enter"/> <input type="button" value="+ Add Shipping"/> </div>					
ID	Name	Amount	Global	Type	Actions
1	Global	50	1	fixed	<input type="button" value="Delete"/> <input type="button" value="Edit"/>

## Shipping Management

In **shipping** menu you will get the shipping list and you can add, remove or modify shipping from there.

Shippings					
<div> <input type="text" value="Type your query and press enter"/> <input type="button" value="+ Add Shipping"/> </div>					
ID	Name	Amount	Global	Type	Actions
1	Global	50	1	fixed	<input type="button" value="Delete"/> <input type="button" value="Edit"/>

## Settings Management

In **settings** menu you will get the settings management form there.

### Settings

---

#### Logo

Upload your site logo from here.  
Dimension of the logo should be 128 x 40px

PNG,JPG

Information

Change your site information from here

Site Title

Pickbazar

Site Subtitle

Your next ecommerce

Currency

US Dollar

Tax Class

Global

Shipping Class

Global

Save Settings

## • International Slug

You will get International slug in some name/title link `products` , `category` , `Attribute` , `Manufacturer` , `Store notice` , etc.

if I write name/title as `Books` , `Books Store` and `We live in a beautiful world` . the slug will be look like

For English:

`Books` Slug: `books`

`Books Store` Slug: `books-store`

`We live in a beautiful world` . Slug: `we-live-in-a-beautiful-world` .

For Arabic:

`كتاب` Slug: `كتاب`

`مكتبة لبيع الكتب` Slug: `مكتبة-لبيع-الكتب`

`أوي تعيش في عالم جميل` Slug: `أوي-تعيش-في-عالم-جميل` .

For Korean:

`책` Slug: `책`

`서점` Slug: `서점`

`오이는 아름다운 세상에 살고 있습니다` . Slug: `오이는-아름다운-세상에-살고-있습니다` .

For Japanese:

本 Slug: 本

本屋 Slug: 本屋

ああ、美しい世界に住んでいます。 Slug: ああ、美しい世界に住んでいます。

---

For Chinese:

書 Slug: 書

書店 Slug: 書店

阿伊生活在一個美麗的世界裡。 Slug: 阿伊生活在一個美麗的世界裡。

---

For Turkish:

kitap Slug: kitap

kitapçı Slug: kitapçı

Oi güzel bir dünyada yaşıyorum. Slug: oi-güzel-bir-dünyada-yaşıyorum.

---

For Telugu:

పుస్తకం Slug: పుస్తకం

పుస్తకాల దుకాణం Slug: పుస్తకాల-దుకాణం

ఓ అందమైన ప్రపంచంలో జీవిస్తున్నా. Slug: ఓ-అందమైన-ప్రపంచంలో-జీవిస్తున్నా.

---

For Spanish:

libros Slug: libros

tienda de libros Slug: tienda-de-libros

Oi vivo en un mundo hermoso. Slug: oi-vivo-en-un-mundo-hermoso.

---

For Bengali:

বই Slug: বই

বইয়ের দোকান Slug: বইয়ের-দোকান

আমরা একটি সুন্দর পৃথিবীতে বাস করি। Slug: আমরা-একটি-সুন্দর-পৃথিবীতে-বাস-করি।

## Available Scripts:

You can run below commands in the root folder for your need.

```
"clean": "yarn workspaces run rimraf \"
{.next,node_modules,__generated__,.cache,src/graphql/*.d.ts,src/framework/graphql
/**/*.d.ts}\" && rimraf node_modules",
"gql-clean": "yarn workspaces run rimraf \"
{__generated__,src/graphql/*.d.ts,src/framework/graphql/**/*.d.ts}\"",
"dev:shop-rest": "yarn workspace @pick-bazar/shop dev:rest",
"build:shop-rest": "yarn workspace @pick-bazar/shop build:rest",
"start:shop-rest": "yarn workspace @pick-bazar/shop start",
"dev:admin-rest": "yarn workspace @marvel/admin-rest dev",
"build:admin-rest": "yarn workspace @marvel/admin-rest build",
"start:admin-rest": "yarn workspace @marvel/admin-rest start",
"dev:shop-gql": "yarn workspace @pick-bazar/shop dev:gql",
"build:shop-gql": "yarn workspace @pick-bazar/shop build:gql",
"start:shop-gql": "yarn workspace @pick-bazar/shop start",
"dev:admin-gql": "yarn workspace @marvel/admin-graphql dev",
"build:admin-gql": "yarn workspace @marvel/admin-graphql build",
"start:admin-gql": "yarn workspace @marvel/admin-graphql start",
"prepare": "husky install"
```

**\*\* Note: \*\*** Also, individual Scripts are available under every individual package. You can check out them from thier individual pacakge.json file.

## Customization

### For customizing the template's default site settings:

If you want to customize the site settings such as site info, default (logo, language , currency code), navigation etc, you can easily change those settings from

```
/frontend/[your-project]/src/settings/site.settings.ts
```

In this file, find your required settings key and change it's value according to your need.

For example, If you want to change the currencyCode : please go to `/frontend/[your-project]/src/settings/site.settings.ts` and find the following portions and change it.

```
currencyCode: "USD";
```

NOTE **\*\*** Some of these options are customizable through ADMIN Dashboard.

## Styles and Assets

### CSS styles:

We use tailwindcss framework with some customization which you find at :

```
open /frontend/[your-project]/tailwind.config.js
```

For tailwindcss documentation:

Go to [Tailwindcss](#)

## Icons:

---

for our icons

```
open /frontend/[your-project]/src/components/icons
```

## For Adding a custom Icon:

---

To add a custom icon please follow this procedure.

1. Open your custom SVG icon file in the code editor and copy all the code.
2. Then Go to src -> components -> icons folder and create a new .tsx file.
3. Then here create a function component and paste your copied SVG code inside the return statement.
4. Then covert all the SVG's kebab-cases properties into camelCase format except the data-name property. For ex. change the stroke-width and fill-color into strokeWidth and fillColor. (for reference you can see one of our icon. )
5. If your custom SVG code has any single custom color then change them into fillColor.

## Utilities

---

In this template, We have used some custom helper functions which is located in

```
cd /frontend/[your-project]/src/utlis/
```

You can use or customize these helper functions according to your needs.

## Backend Integration

---

### For API Integration:

---

```
[your-frontend-project] = admin [rest/graphql] or shop
```

We have used env variables using `.env` file format. You have to put your API url within this file.

For example:

Put that url in the `shop/.env` and `admin/rest/.env` or `admin/graphql/.env`

for `GraphQL`

```
NEXT_PUBLIC_GRAPHQL_API_ENDPOINT= '{put_your_api_url_here}'
```

or for `REST`

```
NEXT_PUBLIC_REST_API_ENDPOINT= '{put_your_api_url_here}'
```

- Navigate to `shop`
- and change `FRAMEWORK_PROVIDER` to `rest` for REST installation or `graphql` for GraphQL Installation from `.env`.
- For `REST`, copy `tsconfig.rest.json` content to `tsconfig.json` or
- For `GraphQL` copy `tsconfig.graphql.json` content to `tsconfig.json`

## Data Fetching

### For GraphQL

We have used GraphQL Query and Mutation which is defined in

```
cd /frontend/[your-project-graphql]/src/graphql/
```

For details documentation:

- [Apollo Client](#)
- [GraphQL Let](#)

You can check and customize it as your need.

### Example:

#### Query

```
import {
  useOrdersQuery,
  OrdersDocument,
  OrderDocument,
  OrdersOrderByColumn,
  SortOrder,
} from "@graphql/orders.graphql";

// Fetch list of orders

const { data, error, loading, fetchMore } = useOrdersQuery({
  variables: {
    orderBy: [{ field: OrdersOrderByColumn.CreatedAt, order: SortOrder.Desc }],
    first: 10,
    page: 1,
    customer_id: 1,
  },
},
```

```
});

fetchMore({
  variables: {
    page,
    first,
  },
});

// Fetch Single Order

const { data, error, loading } = useOrderQuery({
  variables: {
    id,
    tracking_number,
  },
});
```

## Mutation

```
import {
  useCreateCouponMutation,
  useUpdateCouponMutation,
  useDeleteCouponMutation,
  CouponInput,
  CouponUpdateInput,
  CouponType,
} from "@graphql/coupons.graphql";

const [createCoupon] = useCreateCouponMutation();
const [updateCoupon] = useUpdateCouponMutation();
const [deleteCoupon] = useDeleteCouponMutation();

const handleSubmit = async () => {
  const createInput = {
    code: "eid50",
    amount: 50,
    active_from: "2020-12-20 13:43:32",
    expire_at: "2020-12-30 13:43:32",
    type: CouponType.FixedCoupon,
  };
  const { data, error, loading } = await createCoupon({
    variables: { input: createInput },
  });
  const updateInput: CouponUpdateInput = {
    code: "eid50",
    amount: 100,
    active_from: "2020-12-20 13:43:32",
    expire_at: "2020-12-30 13:43:32",
    type: CouponType.FixedCoupon,
  };
  const { data, error, loading } = await updateCoupon({
    variables: { id: 22, input: updateInput },
  });
  const { data, error, loading } = await deleteCoupon({
```

```
variables: { id: 22 },
});
};
```

## For REST API

We have used React Query and Axios which is defined in

```
/frontend/[your-project-rest]/src/data/
```

For details documentation:

- [React-Query](#).  
You can check and customize it as your need.

## Example:

### Query

```
import { useProductsQuery } from "@data/product/use-products.query";

const {
  isFetching: loading,
  isFetchingNextPage: loadingMore,
  fetchNextPage,
  hasNextPage,
  isError,
  data,
  error,
} = useProductsQuery({
  type: "grocery",
  text: "",
  category: "grocery",
});
```

### Mutation

```
import { useCreateProductMutation } from "@data/product/product-create.mutation";

const {
  mutate: createProduct,
  isLoading: creating,
} = useCreateProductMutation();

function handleSubmit(inputValues) {
  createProduct(
    {
      ...inputValues,
    },
    {
      onError: (error: any) => {
        Object.keys(error?.response?.data).forEach((field: any) => {
          setError(field, {
```

```

    type: "manual",
    message: error?.response?.data[field][0],
  });
});
},
}
);
}

```

# Deployment

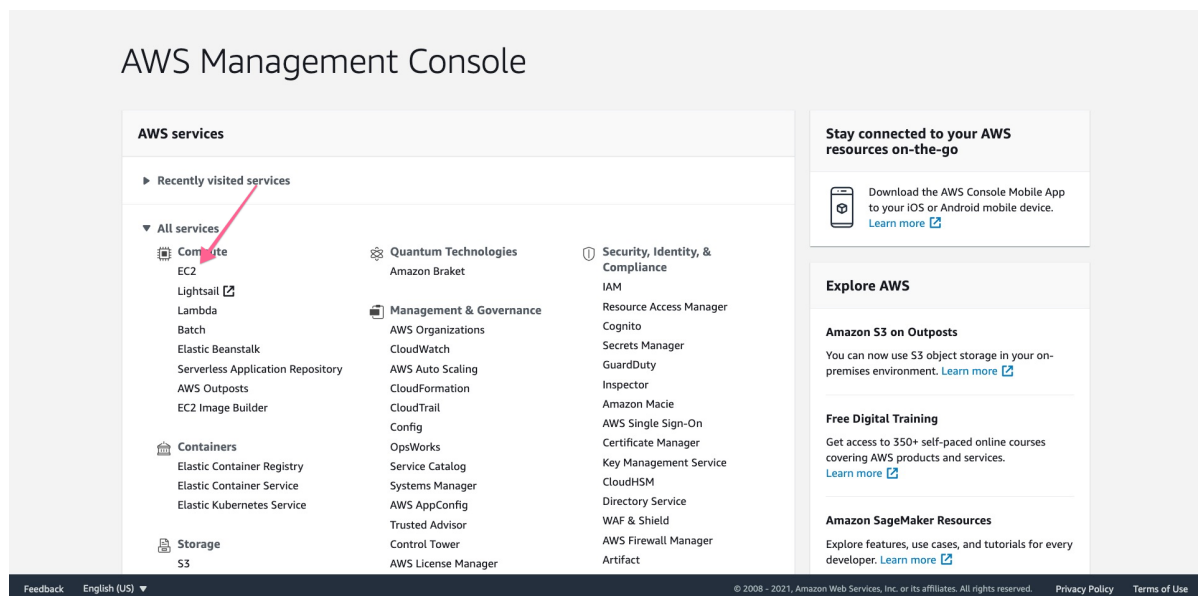
## AWS (Amazon Web Service)

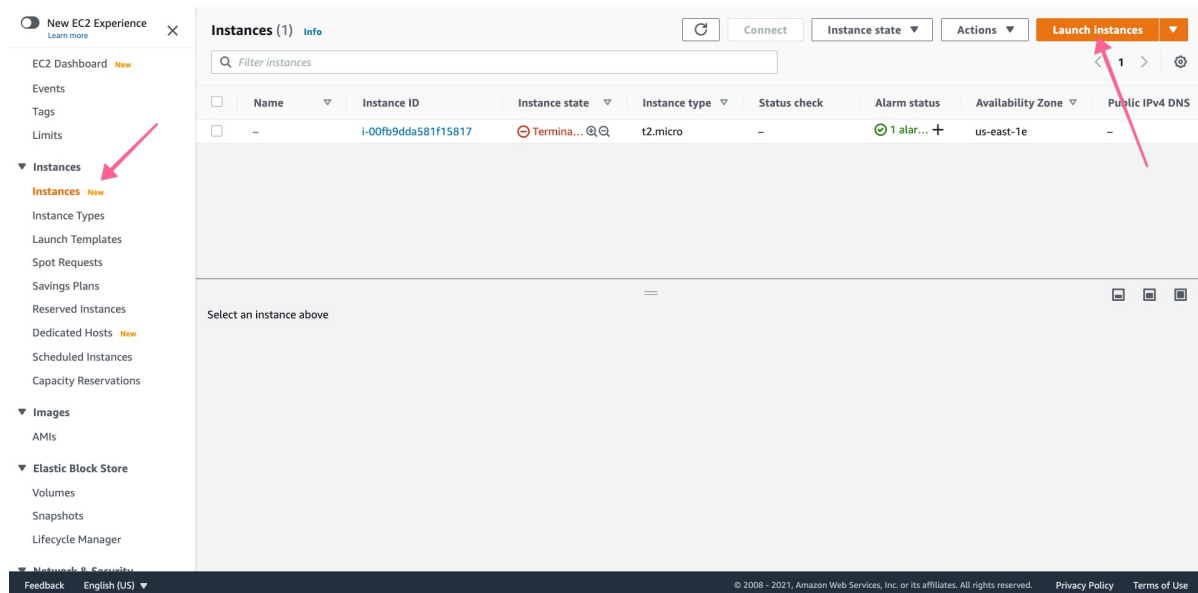
If you want to use all the scripts ( `shop`, `admin`, `api` ) on the same server as this tutorial, then we recommend creating a blank ubuntu-based server with at least 2+ CPU cores and 2GB+ memory.

### How to create ec2 server?

In this AWS tutorial, we're going to create an ec2 server. To do that at first, login to your AWS account and then click,

ec2 -> Instance -> Launch Instance





Then select a ubuntu 20.04 server

After that, click **Next** -> **Next** -> **Next** -> **Next**

And on security pages, add a rule for **HTTP**, **HTTPS** and **SSH**,

Our automation scripts setup **HTTPS** in your domain so you should open **HTTPS**

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom ::/0	e.g. SSH for Admin Desktop
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom ::/0	e.g. SSH for Admin Desktop

**Add Rule**

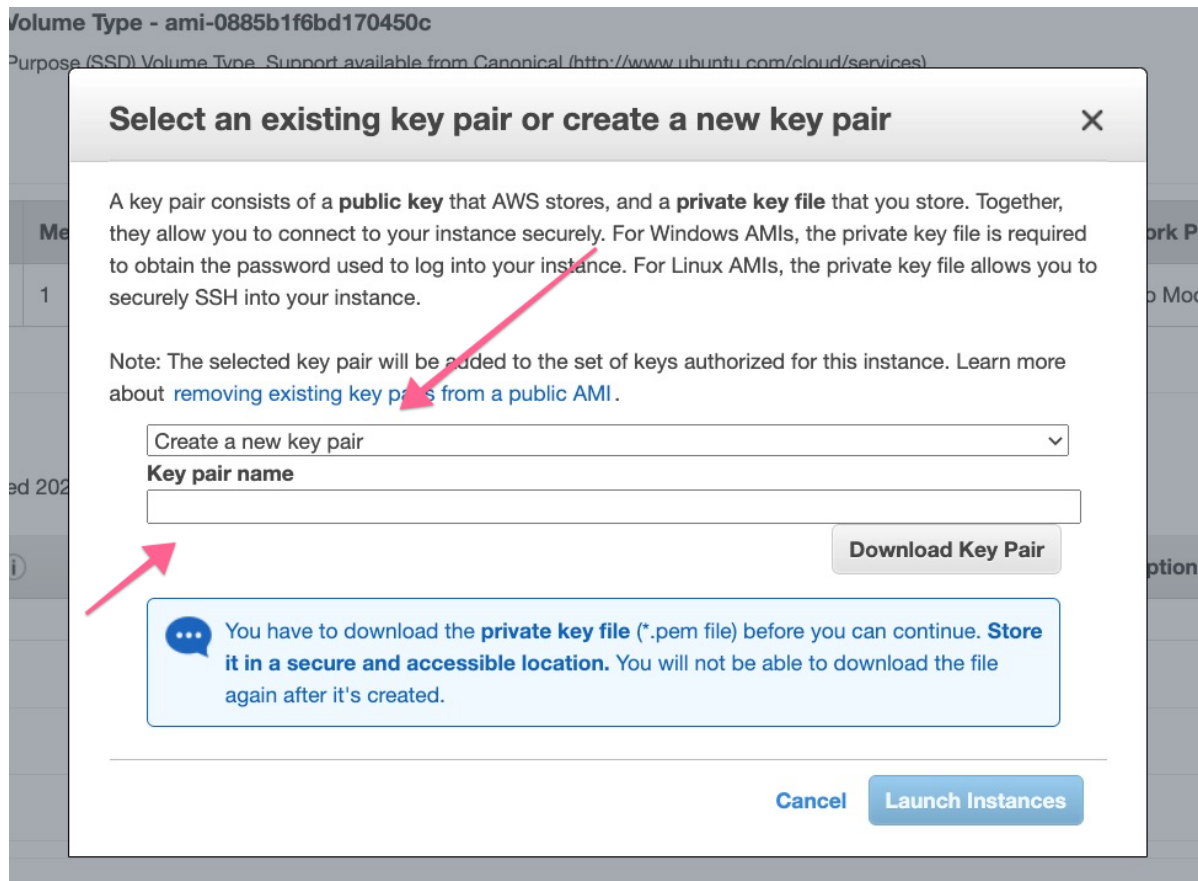
NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

**Cancel** **Save**

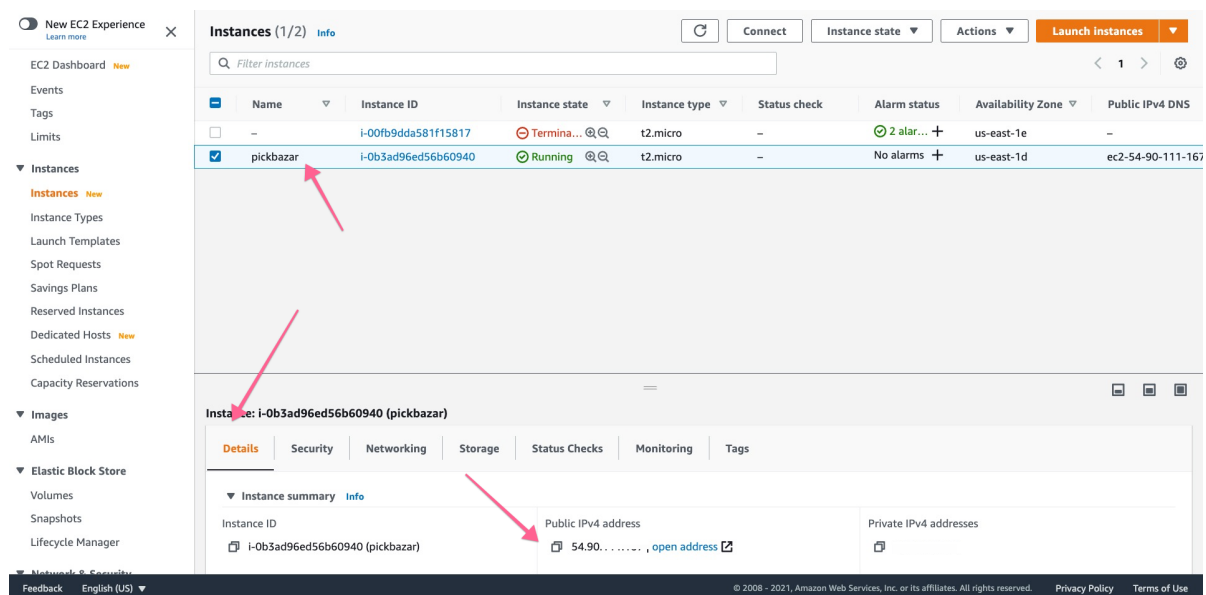
After review, click **Launch**, and you'll get and popup for KeyPair, which will be required to login to the server using ssh.

If you already have a previous KeyPair, you can use that; otherwise, you can create a new one.

After completing that, make sure you download that KeyPair.



After launching the instance, you'll get the server IP, which will be required to login into ssh.



## Domain Setup

Now copy the server IP and connect it with your domain name.

The screenshot shows the AWS Management Console interface. On the left is the navigation menu with options like EC2 Dashboard, Events, Tags, Limits, Instances, Images, and Elastic Block Store. The main panel displays the 'Instances (1/2)' page. A table lists two instances: one in a 'Terminated' state and another named 'pickbazar' in a 'Running' state. Below the table, the details for the 'pickbazar' instance are shown, including its ID, public IPv4 address, and private IPv4 addresses. Red arrows highlight the instance name in the table, the instance ID in the details panel, and the public IPv4 address in the details panel.

Please contact your domain provider for detailed explanation of how to do that.

For domain and DNS, we're going to use **Cloudflare**. It's a free DNS service that is used for DNS management and CDN. Not only that, it can be used to improve the security of your website as all the network come through your website by Cloudflare, and Cloudflare can prevent most of the security threats based on their traffic filtering.

To use Cloudflare,

1. Create a Cloudflare account
2. Add a site
3. Input your domain name
4. And change the nameserver to your domain provider admin panel. -->

Now we'll be connecting our created AWS server with Cloudflare.

To do that, copy the **IP Address** from the **AWS** panel and then add it to **Cloudflare** as an **A** record.

This screenshot is identical to the one above, showing the AWS Management Console with the 'pickbazar' instance selected. Red arrows point to the instance name in the table, the instance ID in the details panel, and the public IPv4 address in the details panel.

Please contact your domain provider for detailed explanation of how to do that.

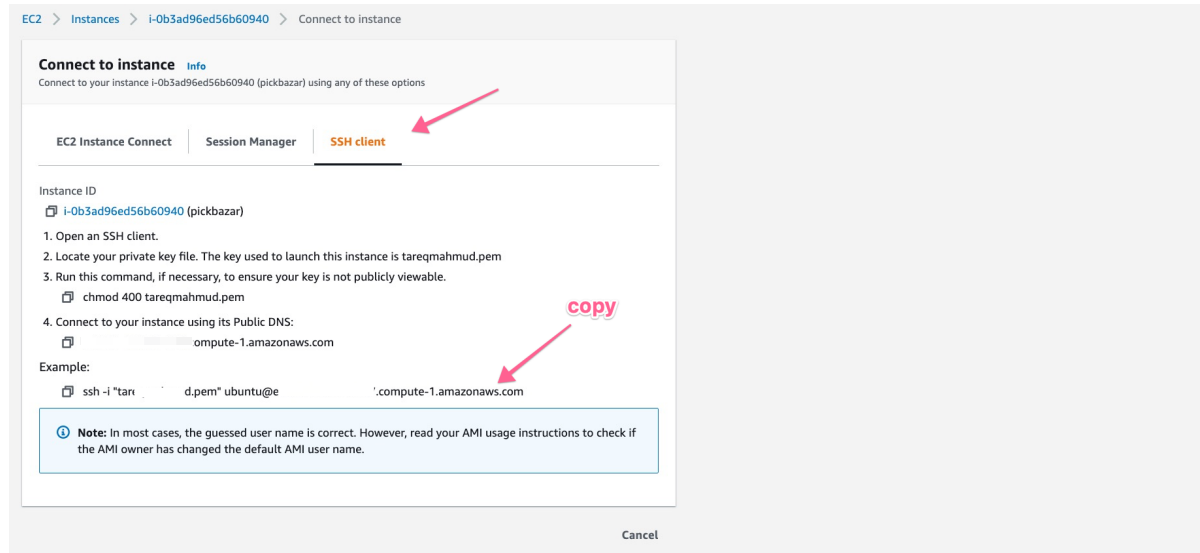
## Login to Server

At first, login to your `AWS` server using `ssh`. to do that, go to the folder from the terminal where `KeyPair` is downloaded.

then click `Connect`



From the `Connect` dashboard, go to `SSH client` and copy the example line and paste it to your terminal.



With this command, you will successfully connect to your server throw `ssh`.

## Change permission .pem

You've to change the permission downloaded `.pem` file to `400` to access the server. To do that, at first go to the location where `.pem` store then run,

```
chmod 400 pickbazar.pem
```

Change the `pickbazar.pem` filename if you use a different name during generate the key.

Now go to the `VPS Server` section for deploy the `Pickbazar Laravel`

## Virtual Private Server (Automated Script)

With this tutorial, you can install PickBazar to any type of blank or empty ubuntu server. For example, `Digital Ocean Droplets`, `Amazon Lightsail`, `AWS`, `Google Cloud Virtual Private Server`, `Azure Ubuntu Virtual Private Server`, etc.

If you want to use all the scripts (`shop`, `admin`, `api`) on the same server as this tutorial, then we recommend creating a blank ubuntu-based (`v20.0.4 lts`) server with at least 2+ CPU cores and 2GB+ memory.

Please connect your `domain` with `server`. We don't recommend/support deployment the project via `IP`.

Please follow this video with the documentation, and it'll make the installation process relatively easy.

## Prerequisite

This automated script is for the \*nix system. So if you are using mac or Linux, then you're good to go. But if you are using windows, then install WSL on your computer and use this script using [WSL](#) or follow this [manual installation](#)

Before starting, the procedure ensures that NodeJS 16 (the latest) is installed on your computer.

```
npm i -g yarn zx
```

Now you can follow the script installation procedure,

## At first login your server from terminal

```
ssh SERVER_USERNAME@SERVERIP
```

Make sure that you are logged in your server then follow the next step and run suggested command.

###Upload api and deployment project to Virtual Server form youp PC - RUN on Local PC

To upload the zipped `api` and `deployment` files to server you need to run the below command form your pickbazar project root

while running below command you will asked for enter your server `username` and `ip address` by entering and a successful connection you will also asked for enter your `api.zip` and `deployment.zip`

files path and the path will be look like `/home/your_project_folder_path/pickbazar-laravel/api.zip` for api.zip file so forth for `deployment.zip`

```
bash deployment/deployment.sh
```

## Server Environment setup script - RUN on Virtual Server

```
bash /var/www/pickbazar-laravel/deployment/nodesetup.sh
```

## Nginx Setup And Settings - RUN on Virtual Server

```
zx /var/www/pickbazar-laravel/deployment/setenv.mjs
```

## Backend build - RUN on Virtual Server

```
sudo zx /var/www/pickbazar-laravel/deployment/backendbuildscript.mjs
```

## Frontend build script - RUN on Local PC

Run the below command from your pickbazar-laravel project root

```
zx deployment/frontendbuildscript.mjs
```

## Frontend run script - RUN on Virtual Server

```
zx /var/www/pickbazar-laravel/deployment/frontendrunscript.mjs
```

## Virtual Private Server (Manual)

If you want to deploy the app using automated script then [follow this](#)

With this tutorial, you can install PickBazar to any type of blank or empty ubuntu server. For example, Digital Ocean Droplets, Amazon Lightsail, AWS, Google Cloud Virtual Private Server, Azure Ubuntu Virtual Private Server, etc.

If you want to use all the scripts (shop, admin, api) on the same server as this tutorial, then we recommend creating a blank ubuntu-based (v20.04 LTS) server with at least 2+ CPU cores and 2GB+ memory.

Please connect your domain with server. We don't recommend/support deployment the project via IP.

Please follow this video with the documentation, and it'll make the installation process relatively easy.

## Access Server

At first login your server using SSH and Terminal

## Install NodeJS & Required Application

### Install NodeJS

At first, we've to install NodeJS and npm to run the pickbazar app. To install NodeJS and npm, run this command on your terminal,

```
sudo apt-get update
curl -sL https://deb.nodesource.com/setup_16.x | sudo -E bash -
sudo apt-get install -y nodejs
sudo add-apt-repository ppa:ondrej/php
sudo apt update
```

### Install Yarn

Pickbazar is highly dependent on yarn, it would be best to handle all the script parts using yarn. So to install yarn, use this command,

```
sudo npm i -g yarn
```

If you face any permission issue, then please check this official doc to resolve that,

[Npm Permission Issue](#)

## Install Zip & Unzip

```
sudo apt install zip unzip
```

## Install PM2

Now we will install `PM2`, which is a process manager for Node.js applications. `PM2` provides an easy way to manage and daemonize applications (run them in the background as a service). To install `PM2` use this command,

```
sudo npm install -g pm2
```

After restarting the server or if the server crash, then pm2 will halt the process. To prevent that, we'll add pm2 as a `startup` process to run automatically after restart the server.

```
pm2 startup systemd
```

## Setup Server

### Introduction

Nginx is one of the most popular web servers in the world. In this deployment tutorial, we're going to use Nginx to host our website. In this tutorial, we're going to use ubuntu 20.04 to host pickbazar

### Step 1 - Installing Nginx

After creating the server, make sure the apt library is up to date. To update the apt library, use this command,

```
sudo apt update
```

Add PPA to get the specific php version

```
sudo add-apt-repository ppa:ondrej/php  
sudo apt update
```

After the update apt, we're going to install Nginx. To do that, use this command

```
sudo apt install nginx
```

### Step 2: Adjusting the Firewall

Before testing Nginx, the firewall software needs to be adjusted to allow access to the service. Nginx registers itself as a service with `ufw` upon installation, making it straightforward to allow Nginx access.

To check the `ufw` list, use this command,

```
sudo ufw app list
```

You will get a listing of an application list like this,

```
Available applications:
Nginx Full
Nginx HTTP
Nginx HTTPS
OpenSSH
```

At first, add ssh to the firewall,

```
sudo ufw allow ssh
sudo ufw allow OpenSSH
```

After that, to enable Nginx on the firewall, use this command,

```
sudo ufw allow 'Nginx HTTP'
```

Now enable the firewall,

```
sudo ufw enable
sudo ufw default deny
```

You can verify the change by typing:

```
sudo ufw status
```

The output will be indicated which HTTP traffic is allowed:

```
Status: active

To Action From
---
Nginx HTTP ALLOW Anywhere
22/tcp ALLOW Anywhere
Nginx HTTP (v6) ALLOW Anywhere (v6)
22/tcp (v6) ALLOW Anywhere (v6)
```

### Step 3 – Checking your Web Server

Now check the status of the Nginx web server by using this command,

```
systemctl status nginx
```

You'll get an output like this,

```
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-01-12 07:35:20 UTC; 7min ago
     Docs: man:nginx(8)
  Main PID: 2940 (nginx)
    Tasks: 2 (limit: 1164)
   Memory: 5.1M
    CGroup: /system.slice/nginx.service
            └─2940 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
               └─2941 nginx: worker process
```

## Step 4 - Install MySQL

```
sudo apt install mysql-server
```

## Step 5 - Install PHP & Composer

```
sudo apt install php8.1-fpm php8.1-mysql
sudo apt install php8.1-mbstring php8.1-xml php8.1-bcmath php8.1-simplexml
php8.1-intl php8.1-gd php8.1-curl php8.1-zip php8.1-gmp
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'e21205b207c3ff031906575712edab6f13eb0b361f2085f1f1237b7126d785e826a450292b6cfd1d
64d92e6563bbde02') { echo 'Installer verified'; } else { echo 'Installer
corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"
sudo mv composer.phar /usr/bin/composer
```

## Step 6 - Create MySQL Database & User

```
sudo mysql
CREATE DATABASE pickbazar;

CREATE USER 'pickbazar_user'@ '%' IDENTIFIED WITH mysql_native_password BY
'pickbazar1';

GRANT ALL ON pickbazar.* TO 'pickbazar_user'@ '%';

FLUSH PRIVILEGES;
```

We use MySQL user name `pickbazar_user` and MySQL password `pickbazar1`. Make sure you change at least MySQL password for security.

## Step 7 - Change permission for the `www` folder

```
sudo chown -R $USER:$USER /var/www/
```

## Step 8 - Upload API to Server

At first, use this command to create a directory on `/var/www/`

```
mkdir /var/www/pickbazar
```

Then, go to your `local computer`

1. Extract the `pickbazar` package that you download from `CodeCanyon`.

2. Rename that folder as `redq-ecommerce`
3. On that folder, you'll get another `folder` called `pickbazar-laravel`.
4. On that folder, you'll get a folder called `api`

Now upload this `api` folder to the server `/var/www/pickbazar/`

## Step 9: Setting Up Server & Project

In this chapter, we'll set up our server and also will set up Reverse Proxy to host all of our sites from the same server.

At first, we'll disable the default configuration.

```
sudo rm /etc/nginx/sites-enabled/default
```

## Step 10 - Create New Nginx for the domain

```
sudo touch /etc/nginx/sites-available/pickbazar
sudo nano /etc/nginx/sites-available/pickbazar
```

Add this Nginx config file to that edited file,

Make Sure You use only one config. If you want to use REST API, then use rest Nginx config or use GraphQL config.

### For REST API

```
server {
    listen 80;

    server_name YOUR_DOMAIN.com;

    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-XSS-Protection "1; mode=block";
    add_header X-Content-Type-Options "nosniff";

    index index.html index.htm index.php;

    charset utf-8;

    # For API
    location /backend {
        alias /var/www/pickbazar/api/public;
        try_files $uri $uri/ @backend;
        location ~ /\.php$ {
            include fastcgi_params;
            fastcgi_param SCRIPT_FILENAME $request_filename;
            fastcgi_pass    unix:/run/php/php8.1-fpm.sock;
        }
    }

    location @backend {
        rewrite /backend/(.*)$ /backend/index.php?/$1 last;
    }
}
```

```

# For FrontEnd -> Rest
location /{
    proxy_pass http://localhost:3000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}

location /admin{
    proxy_pass http://localhost:3002/admin;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}

error_page 404 /index.php;

location ~ /\.php$ {
    fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
    include fastcgi_params;
}

location ~ /\.(!well-known).* {
    deny all;
}
}

```

## For GraphQL API

Skip this step if you want to use REST API

```

server {
    listen 80;

    server_name YOUR_DOMAIN.com;

    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-XSS-Protection "1; mode=block";
    add_header X-Content-Type-Options "nosniff";

    index index.html index.htm index.php;

    charset utf-8;

    # For API
    location /backend {
        alias /var/www/pickbazar/api/public;
        try_files $uri $uri/ @backend;
    }
}

```

```

        location ~ /\.php$ {
            include fastcgi_params;
            fastcgi_param SCRIPT_FILENAME $request_filename;
            fastcgi_pass    unix:/run/php/php8.1-fpm.sock;
        }
    }

    location @backend {
        rewrite /backend/(.*)$ /backend/index.php?/$1 last;
    }

    # For FrontEnd -> GraphQL
    location /{
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    location /admin{
        proxy_pass http://localhost:3004/admin;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    error_page 404 /index.php;

    location ~ /\.php$ {
        fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~ /\.(!well-known).* {
        deny all;
    }
}

```

Make sure you change `YOUR_DOMAIN.com` to your specific `domain name`

You can change `api` path, if you want to change the the domain path for the laravel application

You can change `admin` path, if you want to change the the domain path for the frontend admin

Save and close the file by typing `CTRL` and `X`, then `Y` and `ENTER` when you are finished.

Then enable the config

```
sudo ln -s /etc/nginx/sites-available/pickbazar /etc/nginx/sites-enabled/
```

Make sure you didn't introduce any syntax errors by typing:

```
sudo nginx -t
```

Next, restart Nginx:

```
sudo systemctl restart nginx
```

## Secure Server

### Step 1: Secure Nginx with Let's Encrypt

```
sudo apt install certbot python3-certbot-nginx
sudo ufw status

sudo ufw allow 'Nginx Full'
sudo ufw delete allow 'Nginx HTTP'

sudo ufw status
sudo certbot --nginx -d YOUR_DOMAIN
```

After this command, you'll get several command prompts. Make sure you take the necessary steps and provide information on that command prompt.

## Install API

### Step 1: Build and Run `api`

At first, go to the `api` folder, then copy `.env.example` to `.env`,

```
cd /var/www/pickbazar/api
cp .env.example .env
```

Edit `.env`

```
nano .env
```

And add `MySQL`, `stripe`, `mail` or others configuration.

Also, add `https://YOUR_DOMAIN.COM/backend` to `APP_URL`. Without this, the `upload` function will be broken.

```
2 APP_ENV=production
3 APP_KEY=
4 APP_DEBUG=true
5 APP_URL=https://YOUR_DOMAIN.COM/backend
6 APP_SERVICE=marvel.test
7 APP_NOTICE_DOMAIN=PICKBAZAR_
8 DUMMY_DATA_PATH=pickbazar
9
10
11 LOG_CHANNEL=stack
12 LOG_LEVEL=debug
13
```

Then install all the packages and install `api`

```
composer install

php artisan key:generate

php artisan marvel:install
```

You'll get several confirmations for migration, dummy data, and admin account. Make sure you check the confirmation step and take the necessary actions based on your requirement.

Enable `laravel storage,`

```
php artisan storage:link
```

Then give proper `permission` for laravel folder,

```
sudo chown -R www-data:www-data storage
sudo chown -R www-data:www-data bootstrap/cache
```

Now, when you go to the `YOUR_DOMAIN/backend` you'll get a `welcome` page like this

PickBazar Laravel

[GRAPHQL PLAYGROUND](#) [DOCUMENTATION](#) [SUPPORT](#) [CONTACT](#)

# FrontEnd Project Build

It's not possible to run both REST and GraphQL versions at the same time. So make sure to follow only the `For REST` section or `For GraphQL` section.

Typescript requires a huge chunk of memory to build the project, so if your server has at least 8gb+ of memory, then you can build the project on your server directly. If not, then build the project on your server, then move the folder to the server then serve the project. We'll do the second method in this tutorial.

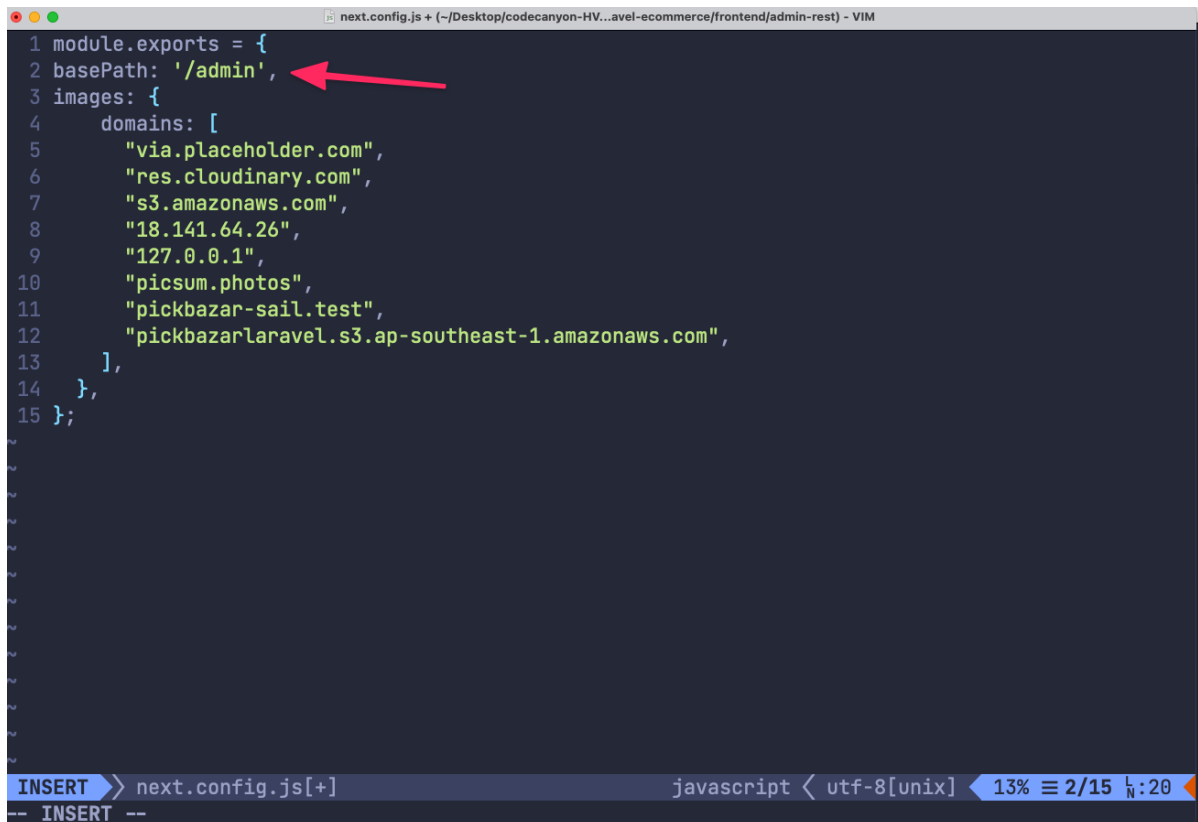
We'll suggest you build the frontend part on your computer and then upload the build file to the server.

Go to your `pickbazar-laravel` folder from your `local computer`.

## Step 1 - Config Next Admin App For /admin Sub Directory

Edit `admin/rest/next.config.js`,

add `basePath` for  `'/admin'`

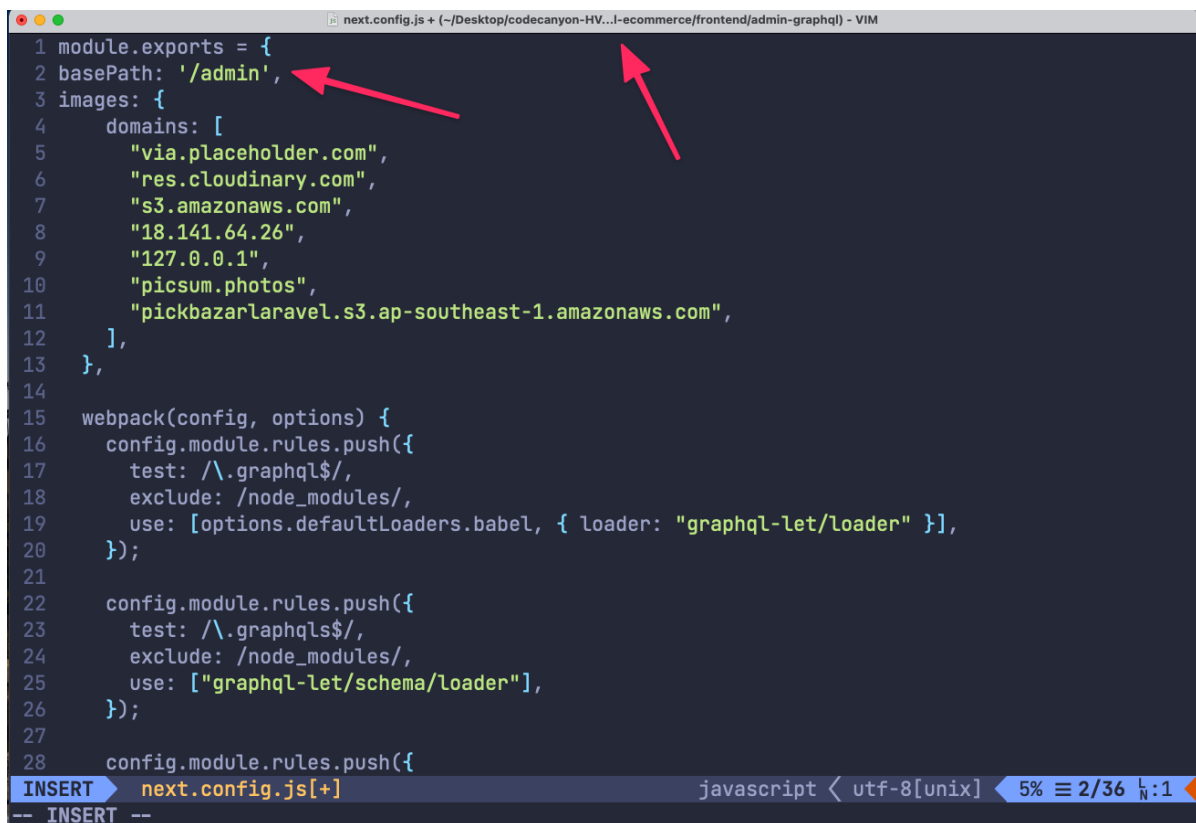


```
1 module.exports = {
2   basePath: '/admin',
3   images: {
4     domains: [
5       "via.placeholder.com",
6       "res.cloudinary.com",
7       "s3.amazonaws.com",
8       "18.141.64.26",
9       "127.0.0.1",
10      "picsum.photos",
11      "pickbazar-sail.test",
12      "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
13    ],
14  },
15 };
-- INSERT --
```

Again,

Edit `admin/graphql/next.config.js`,

add `basePath` for  `'/admin'`



```
1 module.exports = {
2   basePath: '/admin',
3   images: {
4     domains: [
5       "via.placeholder.com",
6       "res.cloudinary.com",
7       "s3.amazonaws.com",
8       "18.141.64.26",
9       "127.0.0.1",
10      "picsum.photos",
11      "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
12    ],
13  },
14
15  webpack(config, options) {
16    config.module.rules.push({
17      test: /\.graphql$/,
18      exclude: /node_modules/,
19      use: [options.defaultLoaders.babel, { loader: "graphql-let/loader" }],
20    });
21
22    config.module.rules.push({
23      test: /\.graphqls$/,
24      exclude: /node_modules/,
25      use: ["graphql-let/schema/loader"],
26    });
27
28    config.module.rules.push({
```

## Step 2 - Install & Build

go to your `pickbazar-laravel` folder again

To install all the npm packages run this command,

```
yarn
```

## Step 3 - Build the project

At first, we've to copy the sample `.env.template` to production `.env` for the shop and admin first.

### For REST

Go to,

```
cd shop
```

then use this command to copy,

```
cp .env.template .env
```

Now edit `.env` and add you `API` url to `.env`

and use

```
NEXT_PUBLIC_REST_API_ENDPOINT="https://YOUR_DOMAIN/backend"
```

and

```
FRAMEWORK_PROVIDER="rest"
```

then copy `tsconfig.rest.json` content to `tsconfig.json`

After that, go to the `admin -> rest` folder,

```
cd ../admin/rest
```

then use this command to copy,

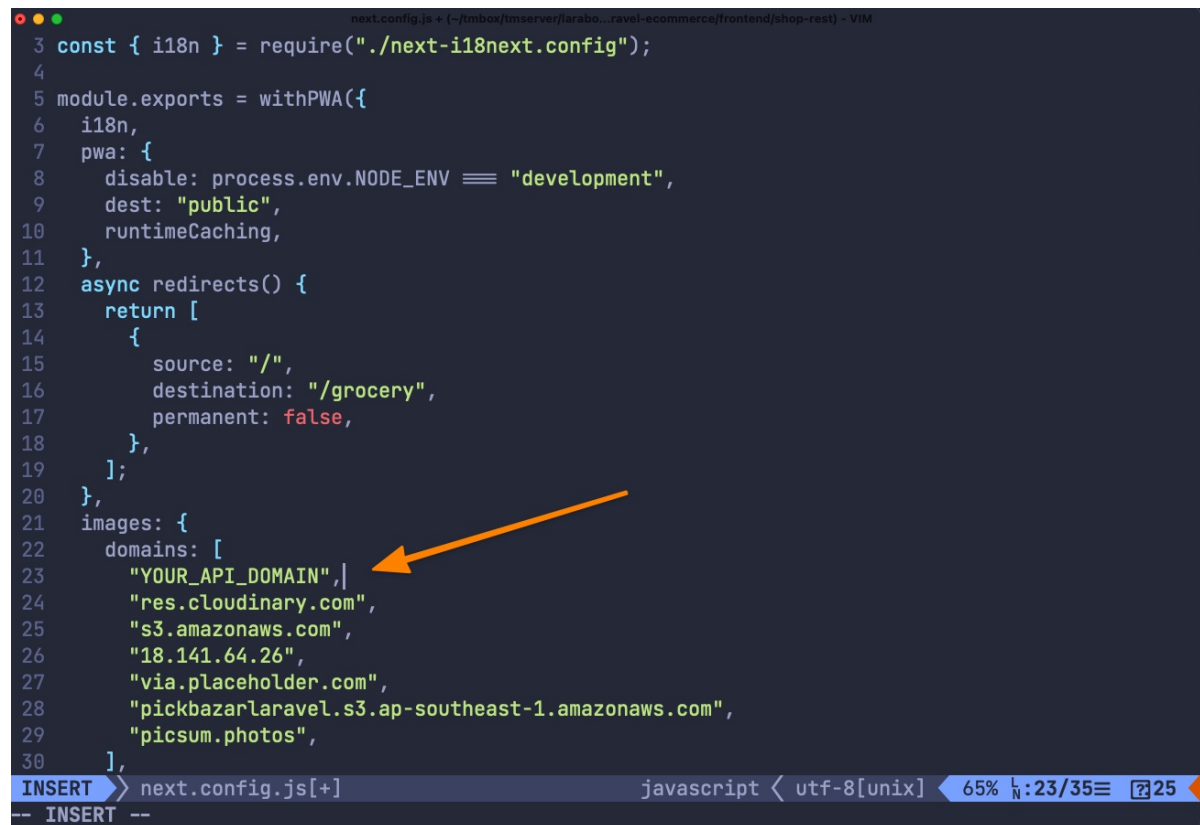
```
cp .env.template .env
```

and use

```
NEXT_PUBLIC_REST_API_ENDPOINT="https://YOUR_DOMAIN/backend"
```

Then open `shop -> next.config.js` and `admin -> rest -> next.config.js`

and add your domain to `images` object



```
3 const { i18n } = require("./next-i18next.config");
4
5 module.exports = withPWA({
6   i18n,
7   pwa: {
8     disable: process.env.NODE_ENV === "development",
9     dest: "public",
10    runtimeCaching,
11  },
12  async redirects() {
13    return [
14      {
15        source: "/",
16        destination: "/grocery",
17        permanent: false,
18      },
19    ];
20  },
21  images: {
22    domains: [
23      "YOUR_API_DOMAIN",
24      "res.cloudinary.com",
25      "s3.amazonaws.com",
26      "18.141.64.26",
27      "via.placeholder.com",
28      "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
29      "picsum.photos",
30    ],
31  },
32 });
```

If your API is hosted on a subdomain, then add that subdomain with root domain on `next.config.js`

## For GraphQL

Skip this step if you want to use REST API

go to `shop` folder from `root` folder,

```
cd shop
```

then use this command to copy,

```
cp .env.template .env
```

Now edit .env and add you `API` url to `.env`

and use

```
NEXT_PUBLIC_GRAPHQL_API_ENDPOINT="https://YOUR_DOMAIN/backend/graphql"
```

and

```
FRAMEWORK_PROVIDER="graphql"
```

then copy `tsconfig.graphql.json` content to `tsconfig.json`

After that, go to the `admin -> graphql` folder,

```
cd ../admin/graphql
```

then use this command to copy,

```
cp .env.template .env
```

Again edit `.env`


and use

```
NEXT_PUBLIC_GRAPHQL_API_ENDPOINT="https://YOUR_DOMAIN/backend/graphql"
```

Then open `shop -> next.config.js` and `admin -> graphql -> next.config.js`

and add your domain to `images` object

```
next.config.js + (-)tmbox/tmserver/laravel-e-commerce/frontend/shop-rest) - VIM
3 const { i18n } = require("./next-i18next.config");
4
5 module.exports = withPWA({
6   i18n,
7   pwa: {
8     disable: process.env.NODE_ENV === "development",
9     dest: "public",
10    runtimeCaching,
11  },
12  async redirects() {
13    return [
14      {
15        source: "/",
16        destination: "/grocery",
17        permanent: false,
18      },
19    ];
20  },
21  images: {
22    domains: [
23      "YOUR_API_DOMAIN",
24      "res.cloudinary.com",
25      "s3.amazonaws.com",
26      "18.141.64.26",
27      "via.placeholder.com",
28      "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
29      "picsum.photos",
30    ],
31  },
32});
```



If your API is hosted on a subdomain, then add that subdomain with root domain on  
next.config.js

## Build Project

Now go to the pickbazar-laravel folder,

and run,

For REST api

```
yarn build:shop-rest
yarn build:admin-rest
```

And for GraphQL,

Skip this step if you want to use REST API

```
yarn build:shop-gql
yarn build:admin-gql
```

Now zip admin, shop, package.json, babel.config.js and yarn.lock files and upload them to the server /var/www/pickbazar

Now go to the server /var/www/pickbazar using terminal

Then unzip the frontend zip file.

# Install FrontEnd And Run

Then install all the node packages,

```
yarn
```

## Run frontend app

Make Sure You use only one run command. If you want to use REST API, then use the rest command or use the GraphQL command.

### For REST API

For `shop rest` app, use this command,

```
pm2 --name shop-rest start yarn -- run start:shop-rest
```

Then to run the `admin rest` app, use this command,

```
pm2 --name admin-rest start yarn -- run start:admin-rest
```

### For GraphQL API

Skip this step if you want to use REST API

For `shop gql` app, use this command,

```
pm2 --name shop-gql start yarn -- run start:shop-gql
```

Then to run the `admin gql` app, use this command,

```
pm2 --name admin-gql start yarn -- run start:admin-gql
```

Now go to Now, go to your `YOUR_DOMAIN` to access the shop page and `YOUR_DOMAIN/admin` for the access admin section.

## Vercel

[vercel.com](https://vercel.com)

If you want to host the template in vercel.com then follow the below command

### API

It's not possible to host the API to vercel. Vercel doesn't support laravel API deployment. So you've to host the API on a separate server. We suggest you create a VPS server and host the API there. For more details, follow this [VPS Deployment](#) docs.

After the host, you'll get your API URL.

## Frontend

Now for frontend add API URL and other necessary config details to,

```
shop -> vercel.json  
admin -> rest -> vercel.json
```

For graphql `admin -> graphql -> vercel.json`

after that, install vercel-cli on your computer using this command,

```
npm i -g vercel-cli
```

After that, log in to vercel using this command,

```
vercel login
```

Then go to the `shop` directory and use this command to deploy,

```
vercel
```

Similarly, go to the `admin -> rest` directory and use this command to deploy,

```
vercel
```

For graphql `admin -> graphql -> vercel.json`

For more details,

Please follow [nextjs deployment docs](#):

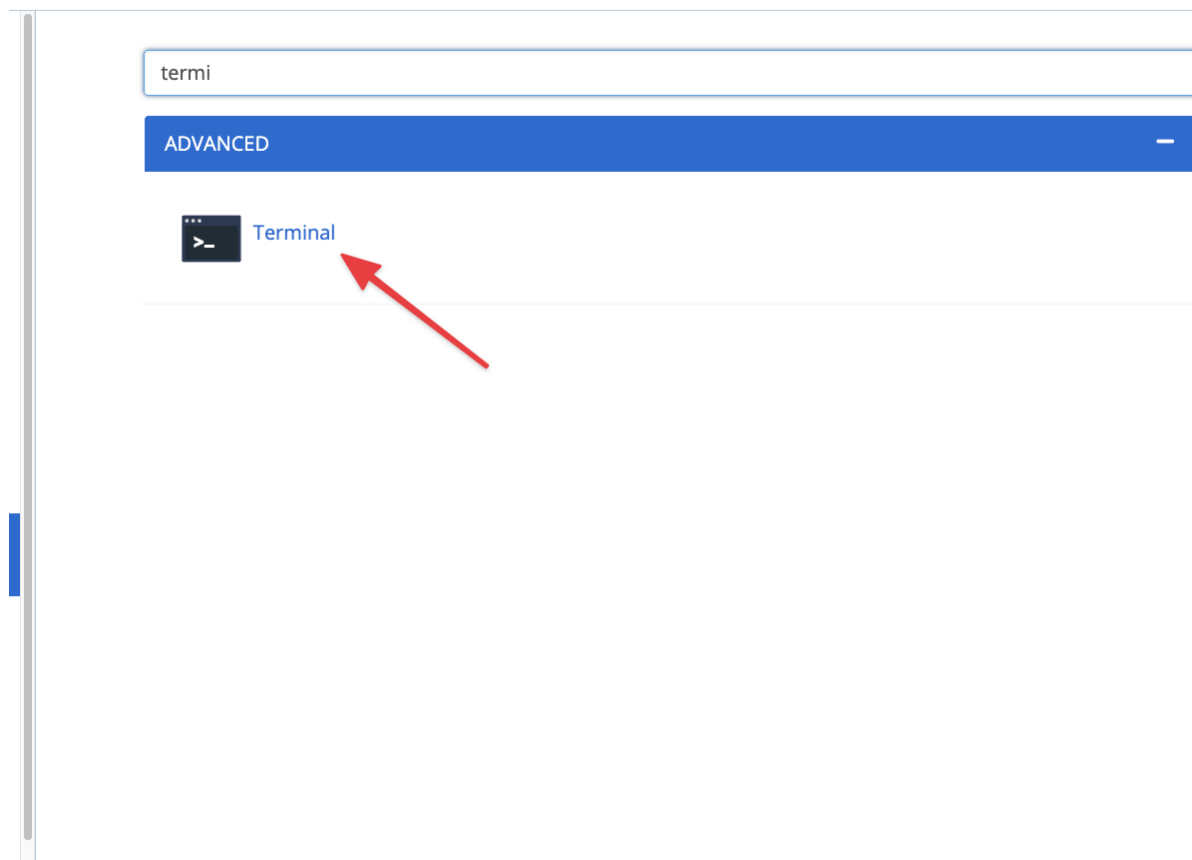
## cPanel

It's quite hard to debug any deployment issue on the cPanel or any managed server as the provider manages this type of server, and they've complete control of the server. And for that, We don't recommend Cpanel or any managed server for deployment. We suggest you use any VPS server where you have complete control of it. you can purchase any \$5 – \$10/mo server from amazon lightsail, ec2 or digitalocean or any ubuntu server

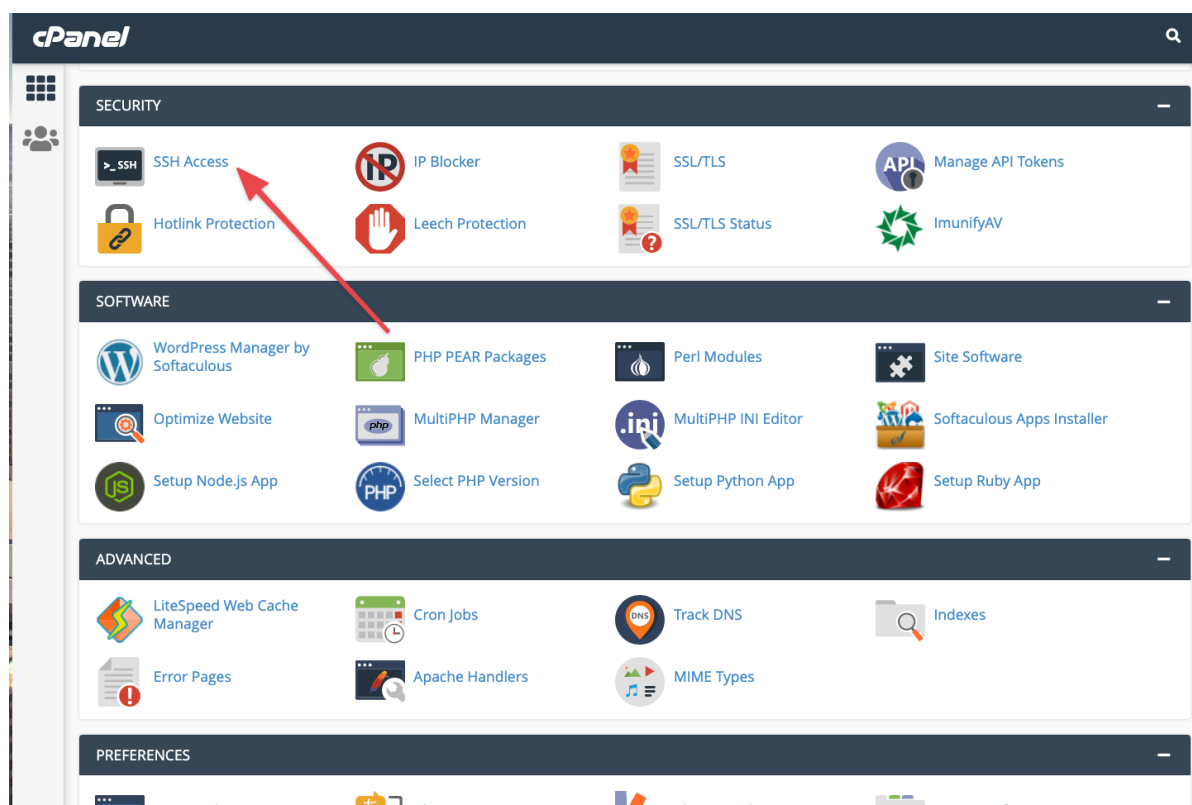
If you still decide to proceed with panel based server, our support team won't be able to help you. We have put some resources for Cpanel in this documentation section to help our users to get started but other than that, we don't have much to offer with Cpanel or any Managed Server

## Access Server

To install the API, access the server using the cPanel terminal first,



If you don't find the terminal, then login to your local computer terminal or [putty](#) for Windows using SSH.

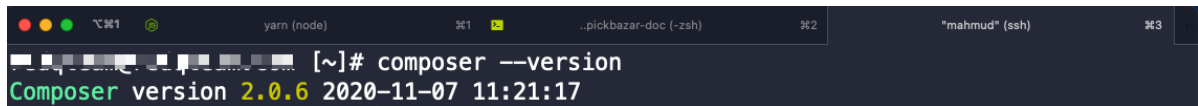


After enabling the ssh login to your server using ssh,

If you don't see any option, then contact your hosting provider as cPanel control by hosting provider.

After logging in, Check if the composer is already installed or not using this command,

```
composer -v
```

A terminal window with a dark background. The prompt is '[~]#'. The command 'composer --version' has been entered. The output is 'Composer version 2.0.6 2020-11-07 11:21:17'. The terminal has tabs at the top: 'yarn (node)', 'pickbazar-doc (-zsh)', and '"mahmud" (ssh)'.

If composer is not installed then, install `composer` to your server.

Check this [YouTube Video](#) for install `composer` on your server,

After that, check the PHP version using,

```
php -v
```

make sure it's `8.1`

## Create Subdomains

Now create two subdomains, for example,

```
-> your_domain.com -> host frontend store.  
-> api.your_domain.com -> host laravel API.  
-> admin.your_domain.com -> host admin dashboard.
```

Or if you want to host all the script on subdomains, then create subdomains like this,

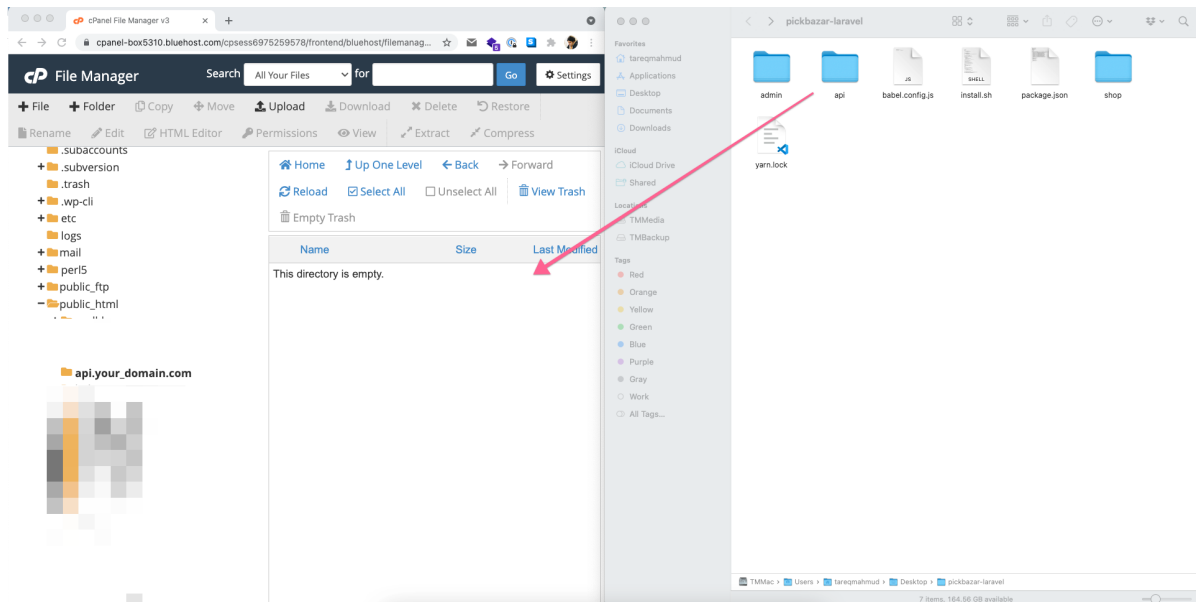
```
-> store.your_domain.com -> host frontend store.  
-> api.your_domain.com -> host laravel API.  
-> admin.your_domain.com -> host admin dashboard.
```

After creating domain/subdomains, make sure all the domain/subdomains are HTTPS enabled. Please contact your hosting provider to enable this, as most hosting providers provide some sort of free SSL.

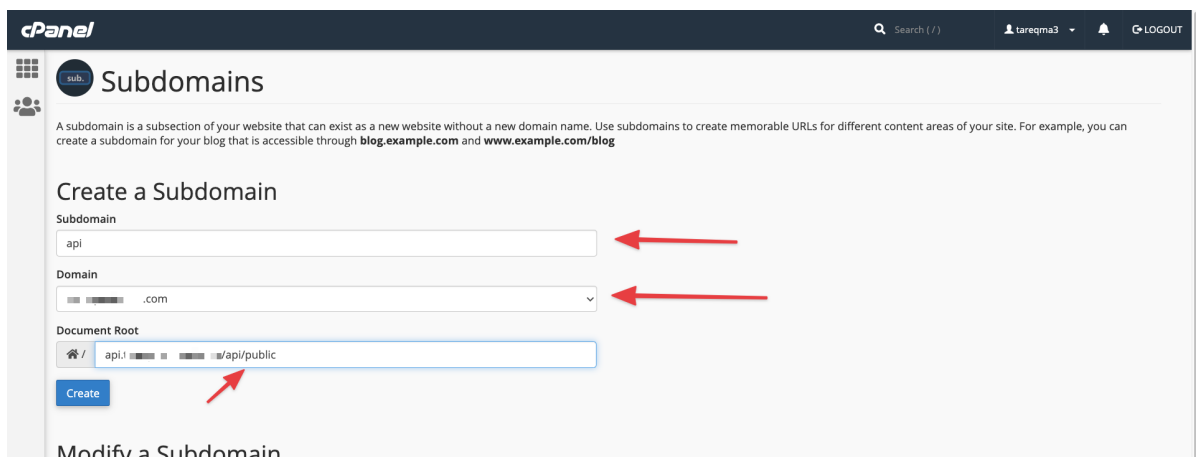
## Install API

1. Extract the `pickbazar` package that you download from `CodeCanyon`.
2. On that folder, you'll get another `zip` called `pickbazar-laravel.zip`.
3. Now extract this `pickbazar-laravel.zip` file.
4. On that file, you'll get a folder called `api`

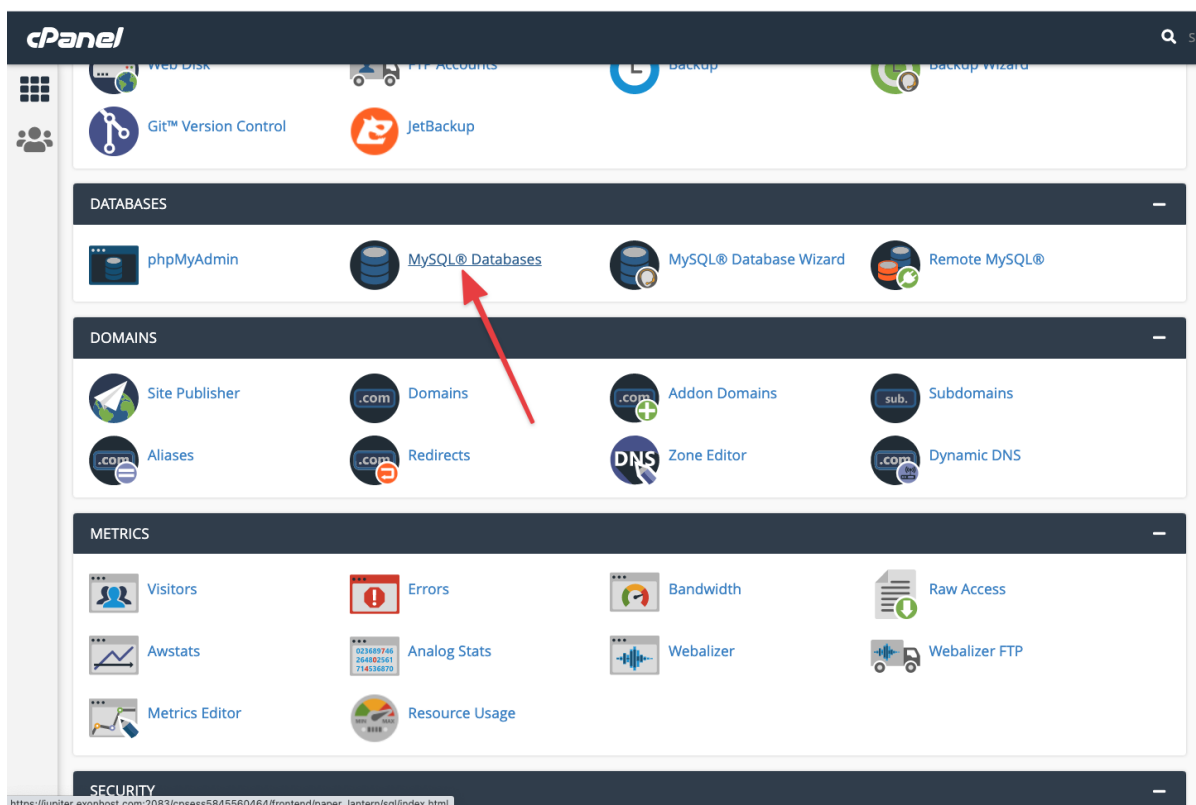
Now upload this `api` folder to the `api.your_domain.com` folder in your server



Make sure your `api.your_domain.com` subdomain Document Root points to that `api/public` folder.



Now create a MySQL database and user from MySQL wizard



After creating the MySQL database, go to your `api` folder from your cPanel file manager and copy `.env.example` to `.env`.

<a href="#">Home</a>		<a href="#">↑ Up One Level</a>	<a href="#">← Back</a>	<a href="#">→ Forward</a>	<a href="#">↻ Reload</a>	<a href="#">☑ Select All</a>	<a href="#">☐ Unselect All</a>	<a href="#">🗑 View Trash</a>	<a href="#">🗑 Empty Trash</a>
Name	Size	Last Modified	Type	Permissions					
app	82 bytes	Apr 4, 2021, 4:53 PM	httpd/unix-directory	0775					
bootstrap	34 bytes	Apr 4, 2021, 4:53 PM	httpd/unix-directory	0775					
config	300 bytes	Apr 4, 2021, 4:53 PM	httpd/unix-directory	0775					
database	74 bytes	Apr 4, 2021, 4:53 PM	httpd/unix-directory	0775					
packages	95 bytes	Apr 4, 2021, 4:53 PM	httpd/unix-directory	0775					
public	106 bytes	Apr 4, 2021, 4:53 PM	httpd/unix-directory	0775					
resources	52 bytes	Apr 4, 2021, 4:53 PM	httpd/unix-directory	0775					
routes	75 bytes	Apr 4, 2021, 4:53 PM	httpd/unix-directory	0775					
storage	46 bytes	Apr 4, 2021, 4:53 PM	httpd/unix-directory	0775					
tests	83 bytes	Apr 4, 2021, 4:53 PM	httpd/unix-directory	0775					
.editorconfig	220 bytes	Apr 4, 2021, 4:53 PM	text/x-generic	0664					
.env	871 bytes	Today, 9:33 PM	text/x-generic	0664					
.env.example	871 bytes	Apr 4, 2021, 4:53 PM	text/x-generic	0664					
.gitattributes	111 bytes	Apr 4, 2021, 4:53 PM	text/x-generic	0664					
.gitignore	216 bytes	Apr 4, 2021, 4:53 PM	text/x-generic	0664					
.styleci.yml	181 bytes	Apr 4, 2021, 4:53 PM	text/x-generic	0664					
artisan	1.65 KB	Apr 4, 2021, 4:53 PM	text/x-generic	0755					
composer.json	2.38 KB	Apr 4, 2021, 4:53 PM	text/x-generic	0664					
composer.lock	390.9 KB	Apr 4, 2021, 4:53 PM	text/x-generic	0664					
docker-compose.yml	1.23 KB	Apr 4, 2021, 4:53 PM	text/x-generic	0664					

After the copy, edit `.env` and add MySQL credentials,

Editing: `your_domain.com/api/.env` Encoding: `utf-8` Re-open Use legacy editor Save Changes Close

Keyboard shortcuts

```
1 APP_NAME=Pickbazar
2 APP_ENV=local
3 APP_KEY=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6 APP_SERVICE=pickbazar-laravel.test
7
8 LOG_CHANNEL=stack
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=localhost
13 DB_PORT=3306
14 DB_DATABASE=ADD_MYSQL_DATABASE_NAME
15 DB_USERNAME=ADD_MYSQL_USERNAME
16 DB_PASSWORD=ADD_MYSQL_PASSWORD
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 QUEUE_CONNECTION=sync
21 SESSION_DRIVER=file
22 SESSION_LIFETIME=120
23
24 MEMCACHED_HOST=memcached
25
26 REDIS_HOST=redis
27 REDIS_PASSWORD=null
28 REDIS_PORT=6379
29
30 MAIL_MAILER=mailgun
31 MAILGUN_DOMAIN=
32 MAILGUN_SECRET=
33 MAIL_FROM_ADDRESS=
34 MAIL_PORT=1025
35 MAIL_USERNAME=null
36 MAIL_PASSWORD=null
37 MAIL_ENCRYPTION=null
38 MAIL_FROM_ADDRESS=support@example.io
39 MAIL_FROM_NAME="${APP_NAME}"
40
41 AWS_ACCESS_KEY_ID=
```

Also, add `https://YOUR_DOMAIN.COM/api` to `APP_URL`. Without this, the `upload` function will be broken.

```
2 APP_ENV=production
3 APP_KEY=
4 APP_DEBUG=true
5 APP_URL=https://YOUR_DOMAIN.COM/backend
6 APP_SERVICE=marvel.test
7 APP_NOTICE_DOMAIN=PICKBAZAR_
8 DUMMY_DATA_PATH=pickbazar
9
10
11 LOG_CHANNEL=stack
12 LOG_LEVEL=debug
13
```

Then go to your `ssh terminal` again and,

go to `api` folder and run,

```
composer install
```

If `composer` installs all the packages successfully, then run this command on the `api` folder,

```
php artisan key:generate  
  
php artisan marvel:install
```

You'll get several confirmations for migration, dummy data, and admin account. Make sure you check the confirmation step and take the necessary actions based on your requirement.

After that, run this command to link storage,

```
php artisan storage:link
```

After install, go to your `api.your_domain_name.com`, and you'll get a webpage like this,

PickBazar Laravel

[GRAPHQL PLAYGROUND](#) [DOCUMENTATION](#) [SUPPORT](#) [CONTACT](#)

## Install Front-end

Before proceeding next step, make sure you already create two subdomains like this,

```
-> your_domain.com -> host frontend store.  
-> admin.your_domain.com -> host admin dashboard.
```

OR

```
-> store.your_domain.com -> host frontend store.  
-> admin.your_domain.com -> host admin dashboard.
```

## Front-end Project Build

It's not possible to run both REST and GraphQL versions at the same time. So make sure to follow only the `For REST` section or `For GraphQL` section.

Typescript requires a huge chunk of memory to build the project, so if your server has at least 8gb+ of memory, then you can build the project on your server directly. If not, then build the project on your server, then move the folder to the server then serve the project. We'll do the second method in this tutorial.

We'll suggest you build the frontend part on your computer and then upload the build file to the server.

### Step 1 - Build Custom Server

go to your `pickbazar-laravel` folder

#### shop rest

Create custom server for `shop rest`,

```
nano shop/server.js
```

and paste this code,

```
// server.js
const { createServer } = require('http')
const { parse } = require('url')
const next = require('next')

const dev = process.env.NODE_ENV !== 'production'
const app = next({ dev })
const handle = app.getRequestHandler()

app.prepare().then(() => {
  createServer((req, res) => {
    // Be sure to pass `true` as the second argument to `url.parse`.
    // This tells it to parse the query portion of the URL.
    const parsedUrl = parse(req.url, true)
    const { pathname, query } = parsedUrl

    if (pathname === '/a') {
      app.render(req, res, '/a', query)
    } else if (pathname === '/b') {
      app.render(req, res, '/b', query)
    } else {
      handle(req, res, parsedUrl)
    }
  }).listen(3003, (err) => {
    if (err) throw err
    console.log('> Ready on http://localhost:3003')
  })
})
```

Now update package.json for `shop rest`,

```
nano shop/package.json
```

and replace `start` script with this,

```
"start": "NODE_ENV=production node server.js"
```

The screenshot shows a side-by-side comparison of a `package.json` file. On the left, the original file has a `"start": "next start"` script. On the right, the file is being edited, and the `"start"` script is being replaced with `"start": "NODE_ENV=production node server.js"`. A red arrow points from the original script to the new one. The editor's status bar at the bottom indicates the file is `package.json` and shows a diff view with 13% changes.

## admin rest

Similarly, create custom server for `admin rest`,

```
nano admin/rest/server.js
```

and paste this code,

```
// server.js
const { createServer } = require('http')
const { parse } = require('url')
const next = require('next')

const dev = process.env.NODE_ENV !== 'production'
const app = next({ dev })
const handle = app.getRequestHandler()

app.prepare().then(() => {
  createServer((req, res) => {
    // Be sure to pass `true` as the second argument to `url.parse`.
    // This tells it to parse the query portion of the URL.
    const parsedUrl = parse(req.url, true)
    const { pathname, query } = parsedUrl

    if (pathname === '/a') {
      app.render(req, res, '/a', query)
    } else if (pathname === '/b') {
      app.render(req, res, '/b', query)
    } else {
      handle(req, res, parsedUrl)
    }
  })
})
```

```
}).listen(3002, (err) => {  
  if (err) throw err  
  console.log('> Ready on http://localhost:3002')  
})  
})
```

Now update package.json for `admin rest`,

```
nano admin/rest/package.json
```

and replace `start` script with this,

```
"start": "NODE_ENV=production node server.js"
```

## admin graphql

Skip this step if you want to use REST API

Create custom server for `admin graphql`,

```
nano admin/graphql/server.js
```

and paste this code,

```
// server.js  
const { createServer } = require('http')  
const { parse } = require('url')  
const next = require('next')  
  
const dev = process.env.NODE_ENV !== 'production'  
const app = next({ dev })  
const handle = app.getRequestHandler()  
  
app.prepare().then(() => {  
  createServer((req, res) => {  
    // Be sure to pass `true` as the second argument to `url.parse`.  
    // This tells it to parse the query portion of the URL.  
    const parsedUrl = parse(req.url, true)  
    const { pathname, query } = parsedUrl  
  
    if (pathname === '/a') {  
      app.render(req, res, '/a', query)  
    } else if (pathname === '/b') {  
      app.render(req, res, '/b', query)  
    } else {  
      handle(req, res, parsedUrl)  
    }  
  }).listen(3000, (err) => {  
    if (err) throw err  
    console.log('> Ready on http://localhost:3000')  
  })  
})
```

Now update package.json for `admin graphql`,

```
nano admin/graphql/package.json
```

and replace `start` script with this,

```
"start": "NODE_ENV=production node server.js"
```

## shop graphql

You can skip this step if you want to use REST API

Create custom server for `shop graphql`,

```
nano shop/server.js
```

and paste this code,

```
// server.js
const { createServer } = require('http')
const { parse } = require('url')
const next = require('next')

const dev = process.env.NODE_ENV !== 'production'
const app = next({ dev })
const handle = app.getRequestHandler()

app.prepare().then(() => {
  createServer((req, res) => {
    // Be sure to pass `true` as the second argument to `url.parse`.
    // This tells it to parse the query portion of the URL.
    const parsedUrl = parse(req.url, true)
    const { pathname, query } = parsedUrl

    if (pathname === '/a') {
      app.render(req, res, '/a', query)
    } else if (pathname === '/b') {
      app.render(req, res, '/b', query)
    } else {
      handle(req, res, parsedUrl)
    }
  }).listen(3001, (err) => {
    if (err) throw err
    console.log('> Ready on http://localhost:3001')
  })
})
```

Now update package.json for `shop graphql`,

```
nano shop/package.json
```

and replace `start` script with this,

```
"start": "NODE_ENV=production node server.js"
```

## Step 2 - Install & Build

go to your `pickbazar-laravel` folder

To install all the npm packages run this command,

```
yarn
```

## Step 3 - Build the project

At first, we've to copy the sample `.env.template` to production `.env` for the shop and admin first.

If you want to use `REST API`, then follow [REST Procedure](#), or if you want to use `GraphQL API`, then follow [GraphQL Procedure](#)

### For REST

Go to,

```
cd shop
```

then use this command to copy,

```
cp .env.template .env
```

Now edit `.env` and add you `API` url to `.env`

```
nano .env
```

and use

```
NEXT_PUBLIC_REST_API_ENDPOINT=https://api.YOUR_DOMAIN.com/
```

and

```
FRAMEWORK_PROVIDER="rest"
```

then copy `tsconfig.rest.json` content to `tsconfig.json`

After that, go to the `admin -> rest` folder,

```
cd ../admin/rest
```

then use this command to copy,

```
cp .env.template .env  
nano .env
```

and use

```
NEXT_PUBLIC_REST_API_ENDPOINT=https://api.YOUR_DOMAIN.com/
```

Now go to the `pickbazar-laravel` folder,

and run,

```
yarn build:shop-rest  
yarn build:admin-rest
```

## For GraphQL

Skip this step if you want to use REST API

and go to,

```
cd shop
```

then use this command to copy,

```
cp .env.template .env
```

Now edit `.env` and add you `API` url to `.env`

```
nano .env
```

and use

```
NEXT_PUBLIC_GRAPHQL_API_ENDPOINT=https://api.YOUR_DOMAIN.com/
```

and

```
FRAMEWORK_PROVIDER="graphql"
```

then copy `tsconfig.graphql.json` content to `tsconfig.json`

After that, go to the `admin -> graphql` folder,

```
cd ../admin/graphql
```

then use this command to copy,

```
cp .env.template .env
```

Again edit `.env`

```
nano .env
```

and use

```
NEXT_PUBLIC_GRAPHQL_API_ENDPOINT=https://api.YOUR_DOMAIN.com/
```

Now go to the `pickbazar-laravel` folder,  
and run,

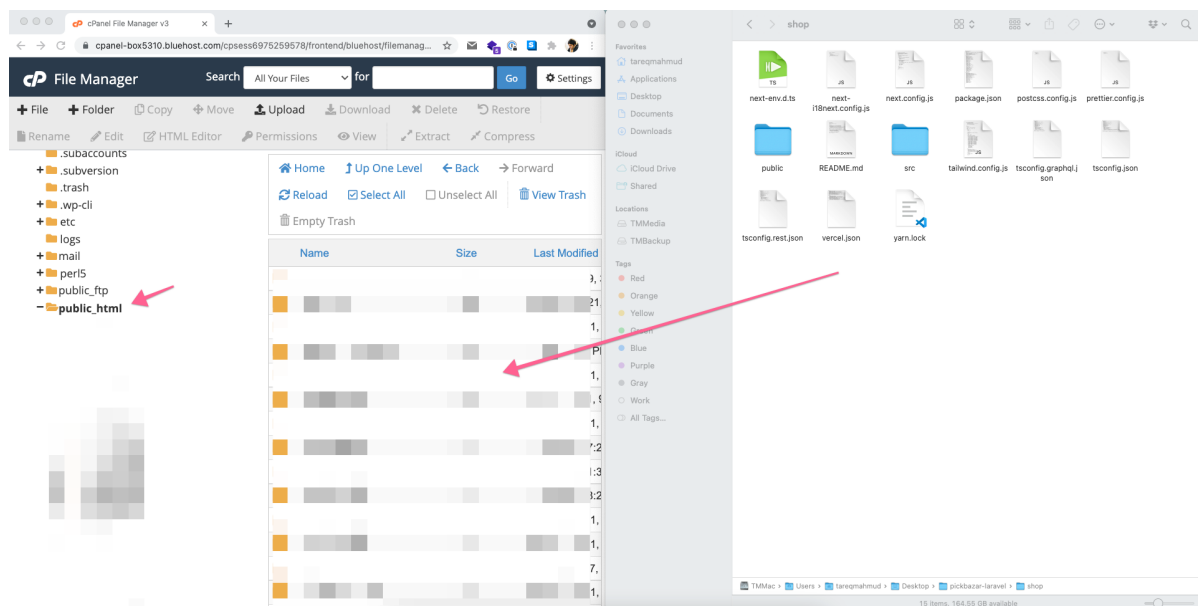
```
yarn build:shop-gql  
yarn build:admin-gql
```

After build the project upload the

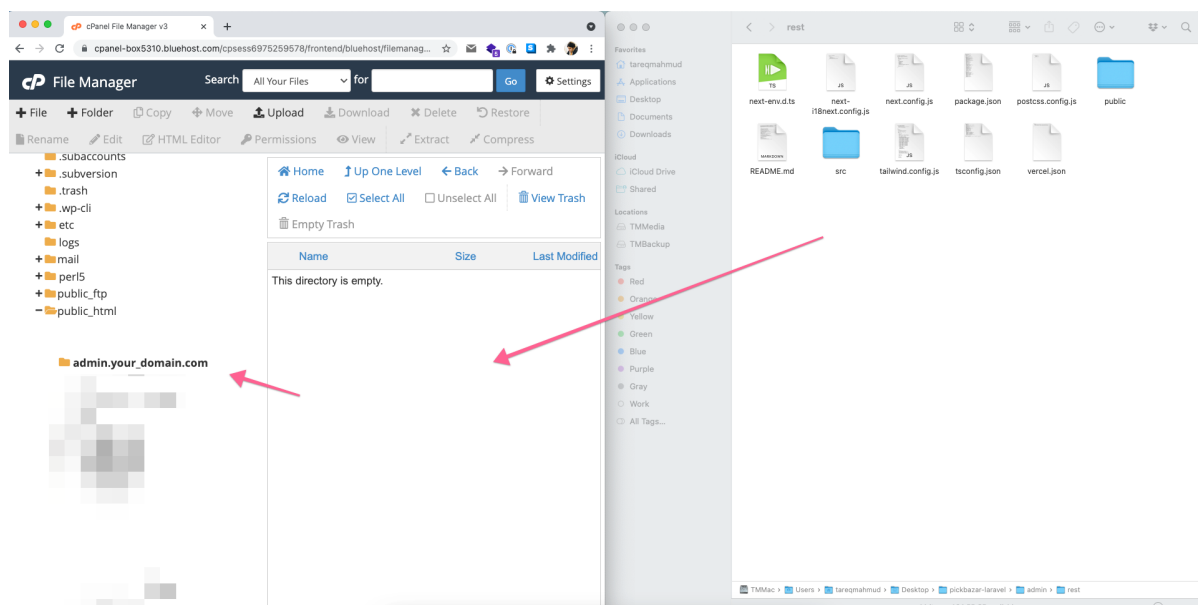
- `shop` to `root_domain -> public_html` folder
- `admin-rest` to `admin.your_domain.com` folder

For graphql upload admin-graphql instead of admin-rest

`shop`,



`shop-admin`,

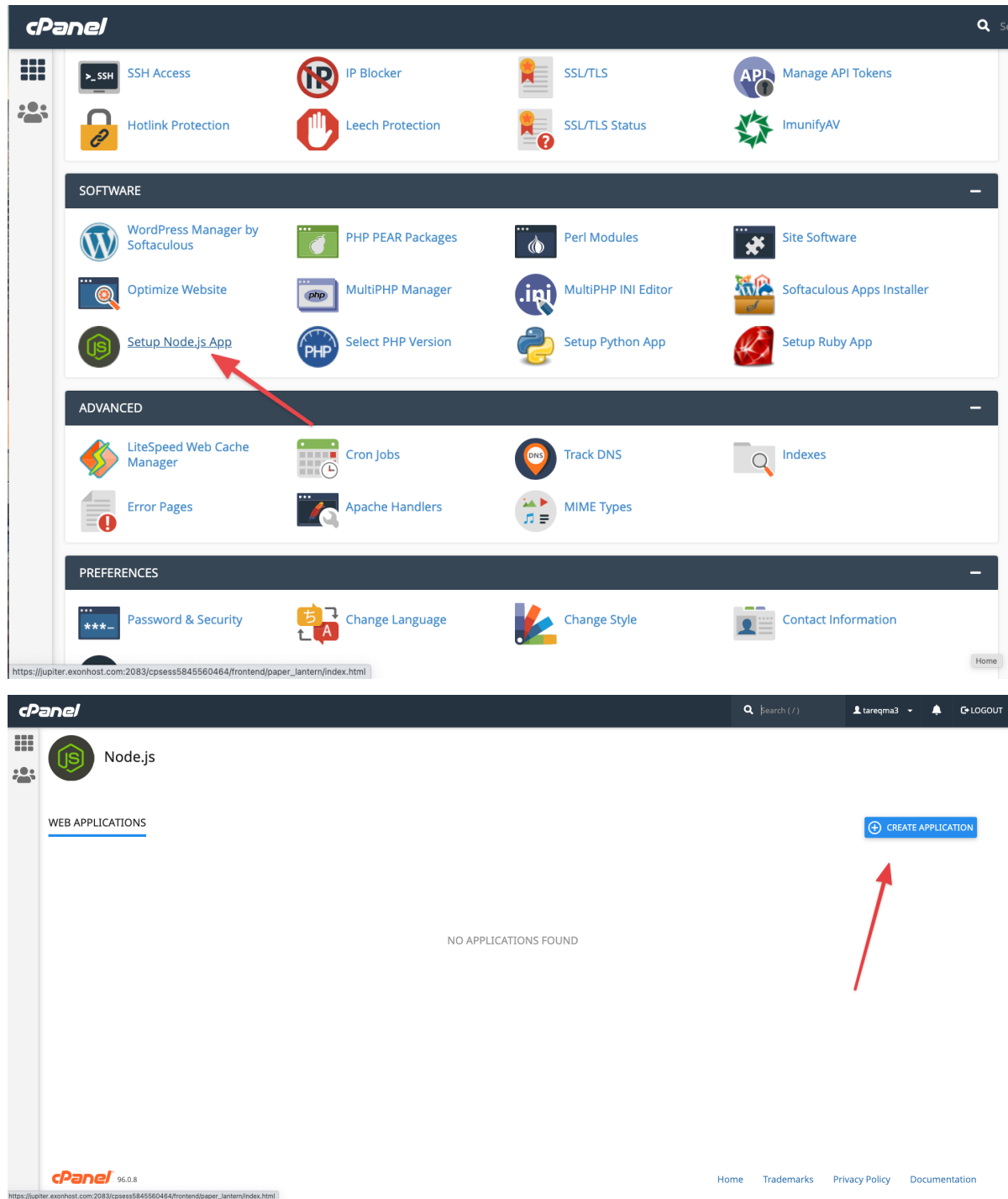


# Install Nodejs Project

We'll run both `shop` and `admin` using the cPanel Nodejs application in this step.

To do that at first go to the Nodejs section from your cPanel,

For `shop`,



Now,

- Select Nodejs version
- Make environment `production`.
- Set Application Root
- And application startup file as `server.js`

**cPanel** Search (/) tareqma3 LOGOUT

WEB APPLICATIONS [+ CREATE APPLICATION](#) CANCEL CREATE

Node.js version: 12.22.1  
 Application mode: Production  
 Application root: /home/tareqma3/public\_html  
 Application URL: [redacted] com  
 Application startup file: server.js  
 Passenger log file: /home/tareqma3/

Environment variables [+ ADD VARIABLE](#)

NO RESULT FOUND

You can get the Application Path from your cPanel file manager

**cP File Manager** Search All Your Files for Go Settings

+ File + Folder Copy Move Upload Download Delete Restore Rename Edit HTML Editor Permissions View Extract Compress

Home Up One Level Back Forward Reload Select All Unselect All View Trash Empty Trash

Name	Size	Last Modified	Type	Permissions
.cagefs	39 bytes	Feb 12, 2021, 2:10 PM	httpd/unix-directory	0771
.cl.selector	48 bytes	May 13, 2021, 1:44 PM	httpd/unix-directory	0755
.cpanel	195 bytes	Today, 5:16 PM	httpd/unix-directory	0700
.cphorde	70 bytes	Mar 24, 2021, 9:28 PM	httpd/unix-directory	0700
.gnupg	79 bytes	Feb 13, 2021, 10:31 PM	httpd/unix-directory	0700
.htpasswd	6 bytes	Jan 23, 2021, 5:05 PM	httpd/unix-directory	0750
.razor	231 bytes	Feb 23, 2021, 10:19 PM	httpd/unix-directory	0755
.softaculous	78 bytes	Jan 23, 2021, 5:06 PM	httpd/unix-directory	0711
.spamassassin	24 bytes	Jan 23, 2021, 5:05 PM	httpd/unix-directory	0700
.subaccounts	28 bytes	Mar 24, 2021, 9:28 PM	httpd/unix-directory	0700
.trash	217 bytes	Feb 13, 2021, 2:09 AM	httpd/unix-directory	0700
.etc	80 bytes	Today, 3:05 PM	httpd/unix-directory	0750
.logs	243 bytes	Yesterday, 6:28 PM	httpd/unix-directory	0700
.iscache	177 bytes	Feb 14, 2021, 2:24 AM	httpd/unix-directory	2770
.lscmData	79 bytes	Feb 12, 2021, 7:02 PM	httpd/unix-directory	0700
.mail	239 bytes	Mar 24, 2021, 9:28 PM	mail	0751
.public_ftp	22 bytes	Jan 23, 2021, 5:05 PM	publicftp	0750
.public_html	162 bytes	Today, 3:55 PM	publichtml	0750
.ssl	77 bytes	May 12, 2021, 1:20 PM	httpd/unix-directory	0755

After create NodeJS app, install all the packages and restart the app,

**cPanel** Search (/) tareqma3 LOGOUT

WEB APPLICATIONS [+ TOM/](#) DESTROY CANCEL SAVE

Node.js STOP APP RESTART

Node.js version: 12.22.1  
 Application mode: Production  
 Application root: public\_html/[redacted]  
 Application URL: [redacted] OPEN  
 Application startup file: server.js  
 Passenger log file: /home/tareqma3/

Detected configuration files [Run NPM Install](#)

For admin,

Similarly, create a another NodeJS application for admin with `admin subdomain` and `admin subdirectory`

After installing and run both NodeJS application, you can access your domain to check PickBazar,

Thank You!

## PickBazar

**Please backup your site before upgrading to latest version.**



## PickBazar Update - Virutal Private Server

If you follow this [Virtual Private Server](#) docs to host your site, then follow this documentation to update PickBazar to a new version.

To build the frontend you've to update the `API` first. But before that this time, we'll use git and GitHub to make upload and download relatively easy.

### Step 1: Setup Git - Server

This step is only for first update. From second update start from [Step 2](#)

At first, we've to install git on our server and config it.

#### Install git

```
sudo apt install git
```

#### Config for first time

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

Make sure you change `you@example.com` and `Your Name` with your `email` and `name`.

# Prepare Git Repository

At first, go to your `pickbazar` directory on your server,

```
cd /var/www/pickbazar
```

2. Then initialize a git on that folder,

```
git init
```

3. Create a new `.gitignore` file

```
nano .gitignore
```

and paste this code to that `.gitignore`,

```
# See https://help.github.com/articles/ignoring-files/ for more about ignoring
files.

# dependencies
node_modules
/.pnp
.pnp.js

# testing
/coverage
# *~
*.swp
tmp/

# misc
.DS_Store

# If Packages have their independent repo
# /packages

# ignore log files
npm-debug.log*
yarn-debug.log*
yarn-error.log*
.idea/
.vscode/
node_modules/
.DS_Store
*.tgz
my-app*
lerna-debug.log
npm-debug.log*
yarn-debug.log*
yarn-error.log*
/.changelog
.npm/
packages/server/node_modules
```

```
packages/server/error.log
packages/server/debug.log
packages/web/.env
packages/reusecore
packages/cloud
```

```
.docz
.now
.vercel
```

After that, `save` that file and use this command for the initial commit,

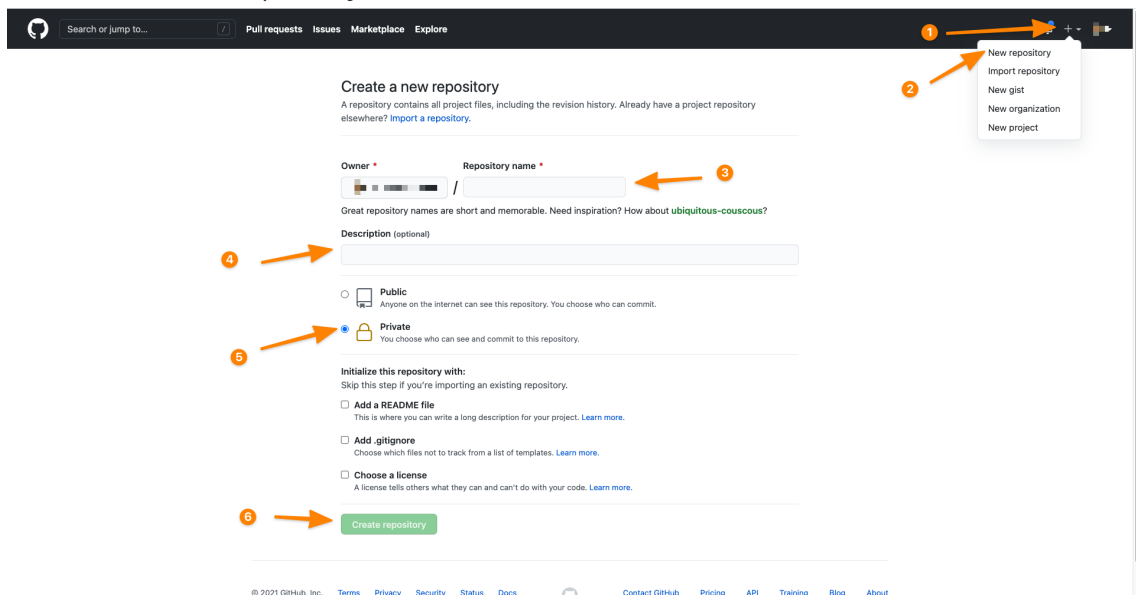
```
git init
git add .
git commit -m "Initial commit"
```

Create a separate branch to maintain the updated code.

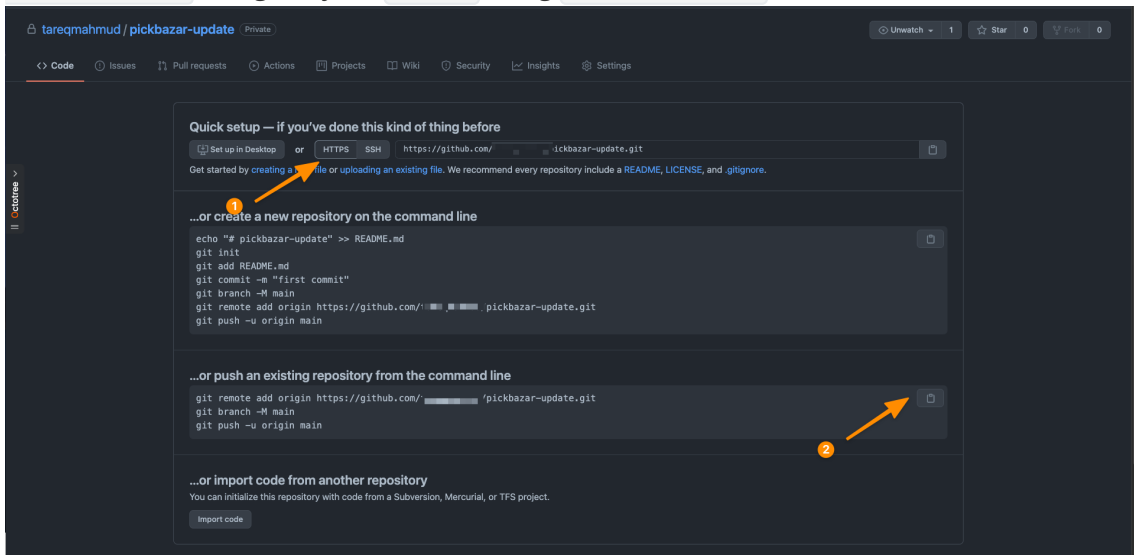
```
git branch pickbazar
```

## Git & Github

1. At first, go to <https://github.com/> and create an account first. If you already have an account then `Sign In` on that account.
2. Then create a new repository,

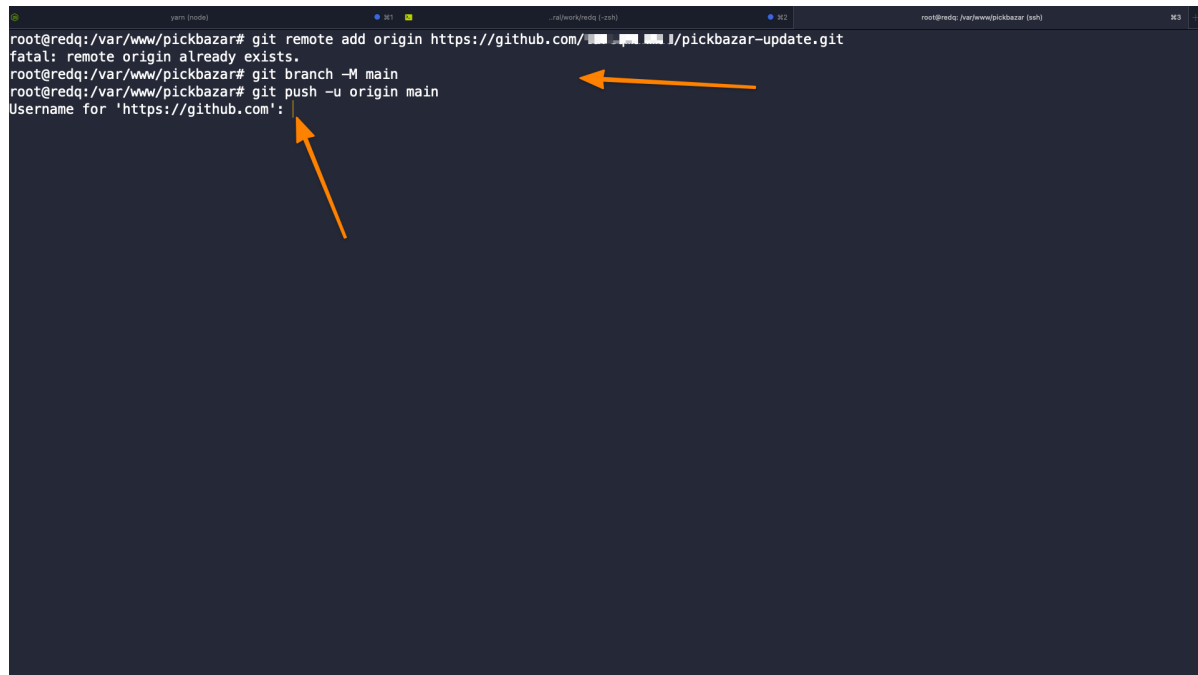


3. After creating the repository you'll get a page like this and from this page copy the `second command block` and go to your `server` using `SSH & terminal`,



And `paste` that copied command to `pickabazar` folder and press `enter`

It'll ask your GitHub `username` and `password`, provide your GitHub `username` and `password`



Your existing repository is successfully connected with GitHub.

## Step 2: Shut Down Current Process

At first use this command to shut down all the applications for update,

```
pm2 stop 'all'
```

## Step 3: Local Repository & Updated Code

In this step,

1. Clone that `GitHub repository` to your `local computer`
2. Download `update package` from `CodeCanyon`

3. Open your terminal and clone that repository to your computer

4. Checkout to `pickbazar` branch

```
git checkout -b pickbazar
```

5. replace `repository` file with downloaded `pickbazar-laravel` folder content.

6. Then use this command to add all files to git

```
git add .  
git commit -m "Update PickBazar API"
```

7. Merge new code with `main` branch

```
git checkout -b main  
git merge pickbazar
```

In this step, you'll face a merge conflict issue. Make sure you resolve all the conflicts to maintain your customization with the updated code. You can check this [video](#) about resolve merge conflict.

After `resolve` and `commit`, push the code to GitHub.

```
git push origin main
```

## Step 4: Update API

In this step, go to your server terminal and go to `/var/www/pickbazar` directory and pull all updated code,

```
git pull origin main
```

After pull go to `api` folder,

```
cd api
```

and install composer package and optimize compiled file,

```
composer install  
php artisan optimize:clear
```

With that updated API will be installed. To check go to your `YOUR_API_DOMAIN/products`

## Step 5: Front-end Project Build

Typescript requires a huge chunk of memory to build the project, so if your server has at least 8gb+ of memory, then you can build the project on your server directly. If not, then build the project on your server, then move the folder to the server then serve the project. We'll do the second method in this tutorial.

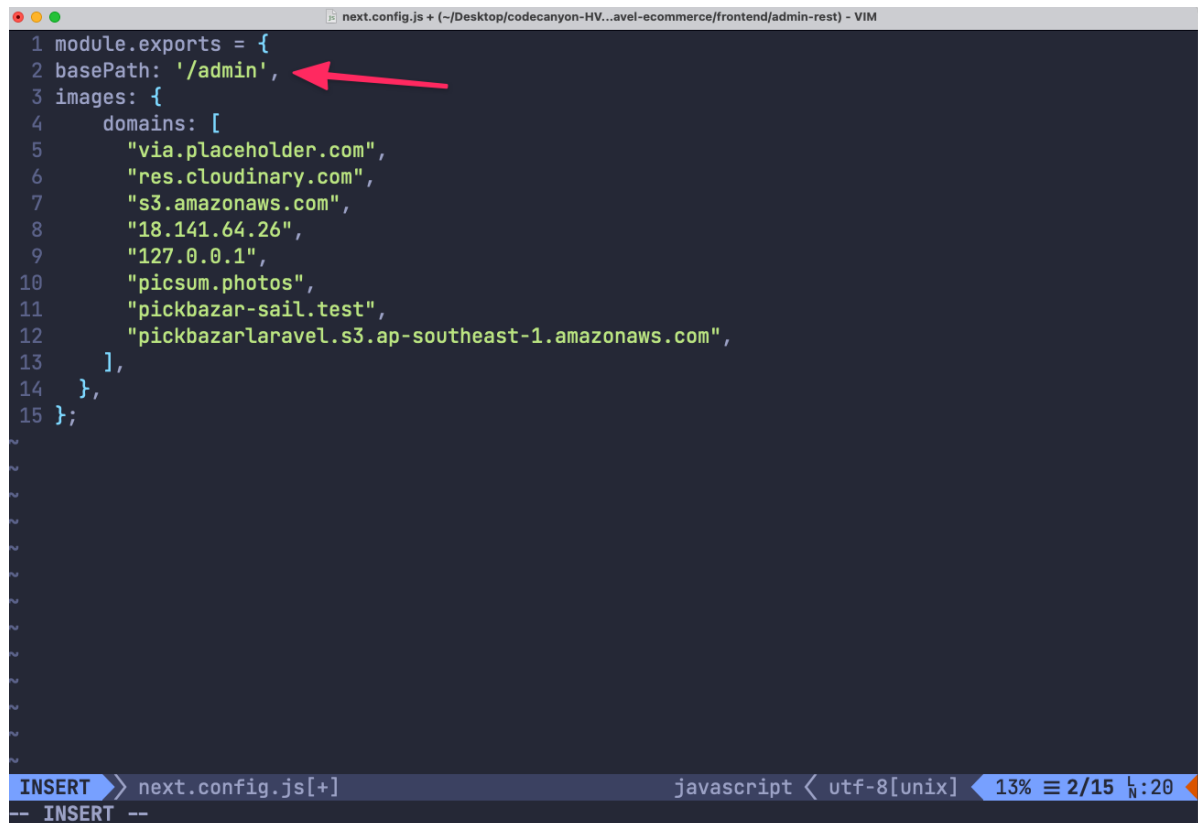
We'll suggest you build the frontend part on your computer and then move the build file to the server using git and github.

Go to your `local git repository` using terminal.

## Config Next Admin App For /admin Sub Directory

Edit `admin/rest/next.config.js`,

add `basePath` for  `'/admin'`

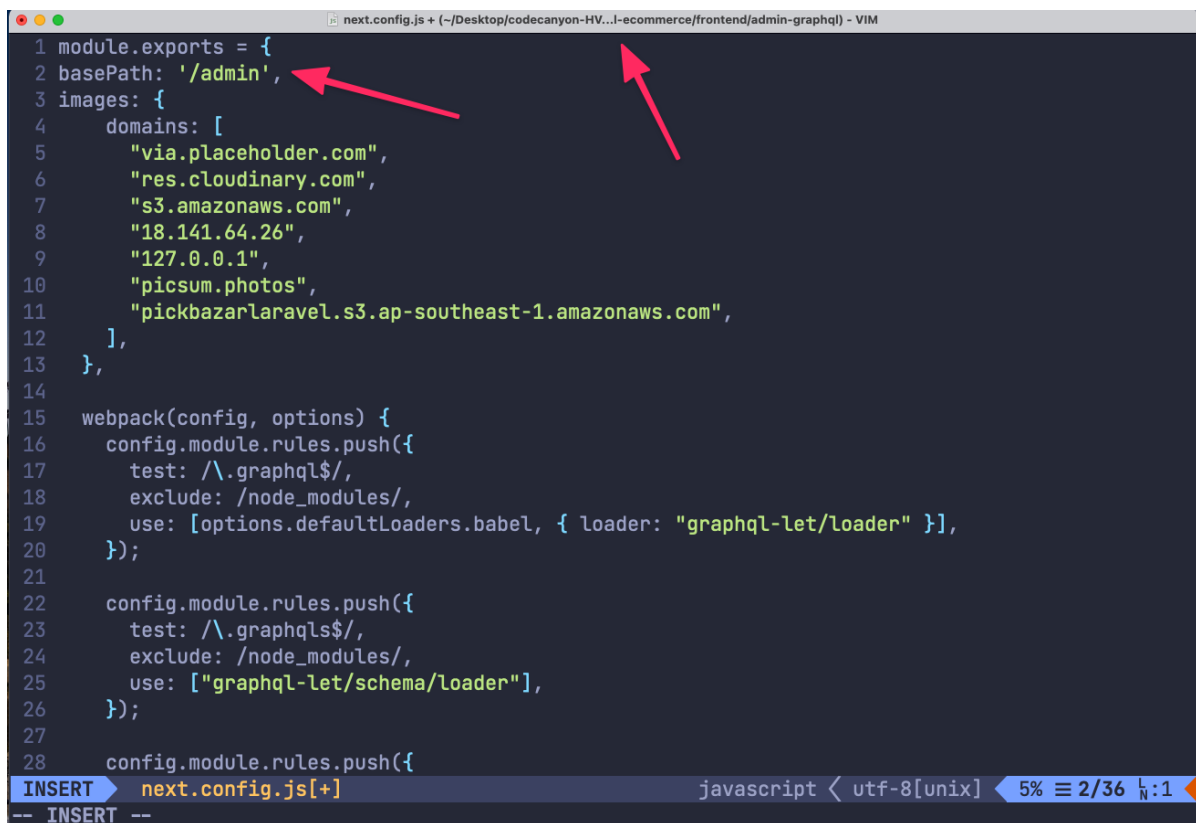


```
1 module.exports = {
2   basePath: '/admin',
3   images: {
4     domains: [
5       "via.placeholder.com",
6       "res.cloudinary.com",
7       "s3.amazonaws.com",
8       "18.141.64.26",
9       "127.0.0.1",
10      "picsum.photos",
11      "pickbazar-sail.test",
12      "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
13    ],
14  },
15 };
-- INSERT --
```

Again,

Edit `admin/graphql/next.config.js`,

add `basePath` for  `'/admin'`



```
1 module.exports = {
2   basePath: '/admin',
3   images: {
4     domains: [
5       "via.placeholder.com",
6       "res.cloudinary.com",
7       "s3.amazonaws.com",
8       "18.141.64.26",
9       "127.0.0.1",
10      "picsum.photos",
11      "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
12    ],
13  },
14
15  webpack(config, options) {
16    config.module.rules.push({
17      test: /\.graphql$/,
18      exclude: /node_modules/,
19      use: [options.defaultLoaders.babel, { loader: "graphql-let/loader" }],
20    });
21
22    config.module.rules.push({
23      test: /\.graphqls$/,
24      exclude: /node_modules/,
25      use: ["graphql-let/schema/loader"],
26    });
27
28    config.module.rules.push({
29
```

## Install & Build

go to your `root` folder again

Clean previous build file,

```
yarn clean
```

To install all the npm packages run this command,

```
yarn
```

## Build the project

For `REST api`

```
yarn build:shop-rest
yarn build:admin-rest
```

And for `GraphQL`,

```
yarn build:shop-gql
yarn build:admin-gql
```

## Upload to GitHub

Use this command to add all files to git,

```
git add .  
git commit -m "Build frontend"  
git push origin main
```

## Step 6: Upload Frontend & Run

---

At first go to your server `pickbazar` or `git` folder and use this command to pull all the build file,

```
git pull origin main
```

Then install all the node packages,

```
yarn
```

## Run frontend app

Use this command to run frontend app as `PM2` again.

```
pm2 start 'all'
```

Now go to Now, go to your `YOUR_DOMAIN` to access the shop page and `YOUR_DOMAIN/admin` for the access admin section.

## SEO

---

### SEO

---

For SEO, we use `Next SEO` packages, and we provide two boilerplates at

```
shop -> src -> components -> seo
```

With that boilerplate, check `Next SEO` docs to use SEO on your pickbazar site,

<https://github.com/garmeeh/next-seo>

## Analytics

---

We have not used any analytics integration in this template yet. But, you can easily integrate any analytics using [Next JS examples](#).

## Laravel API

---

Pickbazar is a laravel multi api package for ecommerce. This package is for building dynamic ecommerce site using pickbazar package with graphql/rest.

# Getting Started

---

For getting started with the template you have to follow the below procedure. For quick guide you can check below videos for installation.

## Installation Windows

### Prerequisites

- PHP 8.1
- Composer
- Xamp/Wamp/Lamp for any such application for apache, nginx, mysql
- PHP plugins you must need
  - simplexml
  - PHP's dom extension
  - mbstring
  - GD Library

### Resources you might need

1. <https://laravel.com/docs/8.x>
2. <https://lighthouse-php.com/5/getting-started/installation.html>
3. <https://github.com/spatie/laravel-medialibrary>
4. <https://github.com/andersao/l5-repository>
5. <https://spatie.be/docs/laravel-permission/v3/introduction>

### Packages we have used

```
"laravel/socialite": "5.5.2",
"laravel/tinker": "2.7.2",
"messagebird/php-rest-api": "3.1.2",
"symfony/http-client": "6.0.9",
"psr/log": "2.0.0",
"symfony/mailgun-mailer": "6.0.7",
"twilio/sdk": "6.40.0",
"srmklive/paypal": "3.0",
"mll-lab/graphql-php-scalars": "5.4.0",
"nuwave/lighthouse": "5.57.0",
"laravel/legacy-factories": "1.3.0",
"cviebrock/eloquent-sluggable": "9.0.0",
"laravel/sanctum": "2.15.1",
"mll-lab/laravel-graphql-playground": "2.6.0",
"prettus/l5-repository": "2.8.0",
"spatie/laravel-medialibrary": "10.4.1",
"spatie/laravel-permission": "5.5.5",
"php-http/guzzle7-adapter": "1.0.0",
"bensampo/laravel-enum": "5.3.1",
"league/flysystem-aws-s3-v3": "3.1.1",
"spatie/laravel-newsletter": "4.11.0",
```

```
"spatie/period": "2.3.3",
"kodeine/laravel-meta": "2.1.0",
"maatwebsite/excel": "3.1.44",
"niklasravnsborg/laravel-pdf": "4.1.0",
"cknow/laravel-money": "7.0.0",
"mollie/laravel-mollie": "2.19.1",
"razorpay/razorpay": "2.8.4",
"unicodeveloper/laravel-paystack": "1.0.8",
"stripe/stripe-php": "8.11.0"
```

## Installation

- Make sure you have run xamp/mamp/wamp/lamp for mysql and php
  - Create a database in your mysql and put those info in next step
  - Rename .env.example file to .env and provide necessary credentials. Like database credentials, stripe credentials, s3 credentials(only if you use s3 disk) admin email shop url etc.

Specially check for this `env` variables

```
DB_HOST=localhost
DB_DATABASE=pickbazar_laravel
DB_USERNAME=root
DB_PASSWORD=
```

- Run `composer install`

```
► composer install
Installing dependencies from lock file (including require-dev)
Verifying lock file contents can be installed on current platform.
```

```
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: barryvdh/laravel-dompdf
Discovered Package: bensampo/laravel-enum
Discovered Package: cviebrock/eloquent-sluggable
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: fruitcake/laravel-cors
Discovered Package: ignited/laravel-omnipay
Discovered Package: intervention/image
Discovered Package: laravel/legacy-factories
Discovered Package: laravel/sail
Discovered Package: laravel/sanctum
Discovered Package: laravel/tinker
Discovered Package: mll-lab/laravel-graphql-playground
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Discovered Package: nuwave/lighthouse
Discovered Package: pickbazar/shop
Discovered Package: prettus/l5-repository
Discovered Package: spatie/laravel-medialibrary
Discovered Package: spatie/laravel-permission
Package manifest generated successfully.
95 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
```

- run `php artisan key:generate`

```
► php artisan key:generate

Application key set successfully.
```

- Run `php artisan marvel:install` and follow necessary steps.

```

Installing Pickbazar Dependencies...

Do you want to migrate Tables? If you have already run this command or migrated tables then be aware, it will erase all of your data. (yes/no) [no]:
> yes

Migrating Tables Now....
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (79.70ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (58.57ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (79.56ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (143.06ms)
Migrating: 2020_04_17_194830_create_permission_tables
Migrated: 2020_04_17_194830_create_permission_tables (839.79ms)
Migrating: 2020_06_02_051901_create_pickbazar_tables
Migrated: 2020_06_02_051901_create_pickbazar_tables (1,601.56ms)
Migrating: 2020_10_26_163529_create_media_table
Migrated: 2020_10_26_163529_create_media_table (71.72ms)
Tables Migration completed.

Do you want to seed dummy data? (yes/no) [no]:
> yes

Copying necessary files for seeding...
File copying successful
Seeding...
Seed completed successfully!

Do you want to create an admin? (yes/no) [no]:
> no

Copying resources files...
Installation Complete

```

- For image upload to work properly you need to run `php artisan storage:link`.

```

The [/var/www/html/public/storage] link has been connected to [/var/www/html/storage/app/public].
The links have been created.

```

- run `php artisan serve`

```

▶ php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
[Sat Apr 10 16:35:26 2021] PHP 8.0.0 Development Server (http://127.0.0.1:8000) started

```

NB: You must need to run `php artisan marvel:install` to finish the installation.

Otherwise your api will not work properly. Run the command and follow the necessary steps.

NB: your frontend `NEXT_PUBLIC_GRAPHQL_API_ENDPOINT` env value will be

```
localhost:8000/graphql
```

NB: your frontend `NEXT_PUBLIC_REST_API_ENDPOINT` env value will be `localhost:8000/`

## For MAC and Linux(with sail and docker)

There is an alternate installation procedure for linux and mac. You can follow below procedure to getting started with `sail`

### Prerequisites

- Docker

### Installation Mac

### Video

### REST API

Follow this tutorial for install REST API

### GraphQL API

Follow this tutorial for install GraphQL

- Run Docker application first

- Now go to your pickbazar-laravel root directory and run `bash install.sh`. It will guide you through some process. Follow those steps carefully and your app will be up and running
- Navigate to `api` then `sail down` to stop the container. If you want to remove the volumes then `sail down -v`

NB: your frontend `NEXT_PUBLIC_GRAPHQL_API_ENDPOINT` env value will be `localhost/graphql`

NB: your frontend `NEXT_PUBLIC_REST_API_ENDPOINT` env value will be `localhost/`

## Configuration

---

All the configurations files are in `packages/marvel/config` folder. You can change any necessary configuration from these files. You can also publishes the shop configuration using `artisan vendor:publish --provider="Marvel\ShopServiceProvider" --tag="config"` command in your root folder.

- Create `.env` file from our example.env file and put necessary configuration
- By default s3 is using for media storage but you can also use local folder. Change `MEDIA_DISK` IN `.env` file as your need. Supported options are `public` and `'s3'`
- Set Payment related configuration to `STRIPE_API_KEY` `.env` variable
- Set `ADMIN_EMAIL`, `SHOP_URL` and necessary Database credentials.

## Console Commands

---

- `php artisan marvel:install` complete installation with necessary steps
- `php artisan marvel:seed` seeding demo data
- `php artisan marvel:copy-files` copy necessary files
- `php artisan vendor:publish --provider="Marvel\ShopServiceProvider" --tag="config"` published the configuration file

All of the above custom command you will find in `packages/marvel/src/Console` folder.

## Development

---

We have provided below two api.

- REST API
- GraphQL API

## REST API

---

All the rest routes is resides in `packages/marvel/src/Rest/Routes.php` file and you can easily navigate to corresponding controller and necessary files.

## Endpoints

<https://documenter.getpostman.com/view/11693148/UVC5Fo9R>

## Folder structure

### config

The `packages/marvel/config` folder contains all the `config` for our app.

### database

The `packages/marvel/database` folder contains all the `factories` and `migrations`.

- **Http:**

Contains two folders. `Controllers` and `Requests`. All the necessary controllers and requests are in this two folder.

- **Database:**

Contains `Models` and `Repositories`. For repositories we have used `l5-repository` (<https://github.com/andersao/l5-repository>).

### Enums

All the `enums` that are used throughout the app is in `packages/marvel/src/Enums` folder.

### Events

All the events are in `packages/marvel/src/Events` folder.

### Listeners

All the listeners corresponding to the above events are in `packages/marvel/src/Listeners` folder

### Mail

All the mailables are in `packages/marvel/src/Mails` folder.

### Notifications

Notifications related to order placed is reside `packages/marvel/src/Notifications`. Currently we have provided mail notification but you can easily add others notification system using laravel conventions.

### Providers

All the secondary service providers that we have used in our app resides in `packages/marvel/src/Providers` folder. The main `ShopServiceProvider` reside in `packages/marvel/src/` folder.

## stubs

The `packages/marvel/stubs` folder contains all the necessary email templates and demo data related resources for the app.

## GraphQL API

### Laravel GraphQL API Endpoint

```
your_domain/graphql
```

To access the graphql api you will need a graphql client. GraphiQL is one of them. You can install it from below link(<https://www.electronjs.org/apps/graphiql>)

### Alternatives

- GraphQL Playground
- <https://altair.sirmuel.design/>

### Folder Structure

All the code specifically related to GraphQL reside in `packages/marvel/src/GraphQL/` folder.

- **Mutations:**

Folder contain necessary mutations files which is connected to rest Controller file for code reusability.

- **Queries:**

Folder contain necessary queries files which is connected to rest Controller file for code reusability.

- **Schema**

This is the most important part of the graphql api. Check the lighthouse-php(<https://lighthouse-php.com/4.18/getting-started/installation.html>) doc for understanding how schema works. We have provided schema for all the models in our app. If you check the above `lighthouse-php` doc you will understand how schema works and how you can modify it to your need.

### Before Finishing up

Before you finishes the installation process make sure you have completed the below steps.

- Copied necessary files and content to your existing laravel projects(if using existing projects)
- Installed all the necessary dependencies.
- Ran `marvel:install` commands and followed the necessary steps.
- Created a `.env` file with all the necessary env variables in the provided projects.
- Put `DISK_NAME` configuration for `public` or `'s3'`

- Set Payment related configuration to `STRIPE_API_KEY`

## Payment Gateway

We have used `omnipay` for payment and given `stripe` and `cash_on_delivery` default. We have used `ignited/laravel-omnipay` by forking it in our packages due to some compatibility issue with Laravel 8.

## Extending The Functionality

If you want to extend the functionality of the app you can easily do it on your app. You would not need to modify code from our packages folder. Like you can add any `routes` and corresponding `controller` in your laravel app as your need. We highly suggest you to do all the modification in your app so you can update the package easily.

## Deployment

Its a basic laravel application so you can deploy it as any other laravel application. Make sure you have installed all the php required plugins we mentioned above.

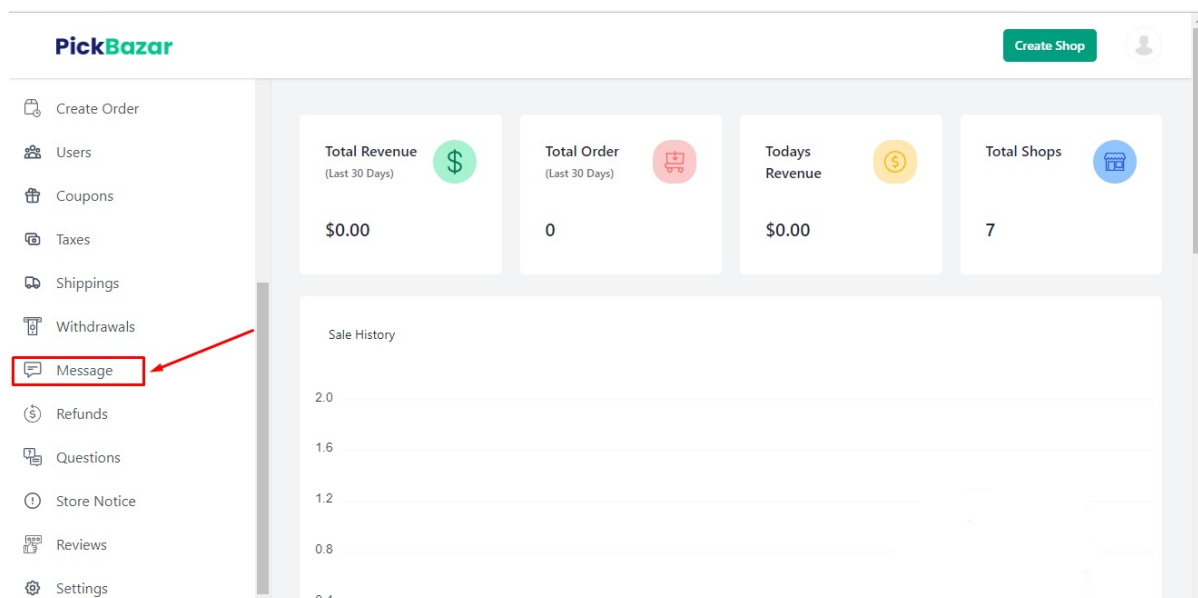
## Messaging (Feature)

With the use of messaging, you can make sure that your shop owner have easy access to the special offers, marketing promotions, and stock updates you're providing them.

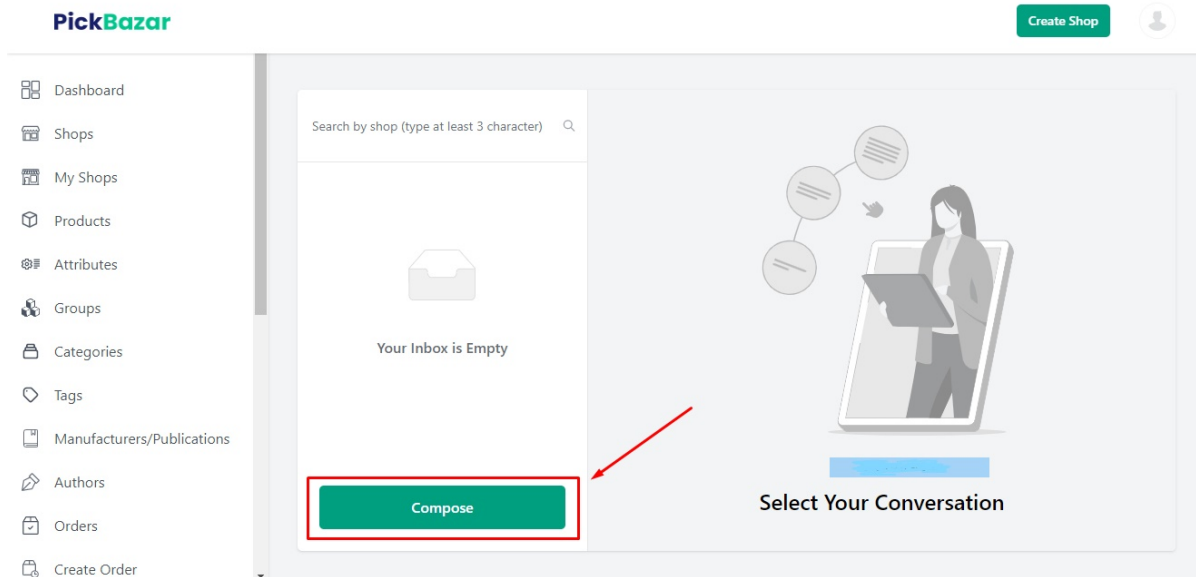
With a 98% open rate, text message links are more likely to be clicked by recipients than email ones.

Private chat between two or more shops and the administrator is possible on PickBazar. Although an admin can build shops, he cannot message those shops that he has made. Admin can send messages to any new shops that any shop owner creates.

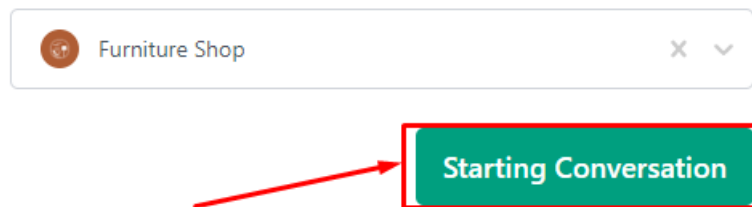
- At first go to dashboard and find message like the picture below.



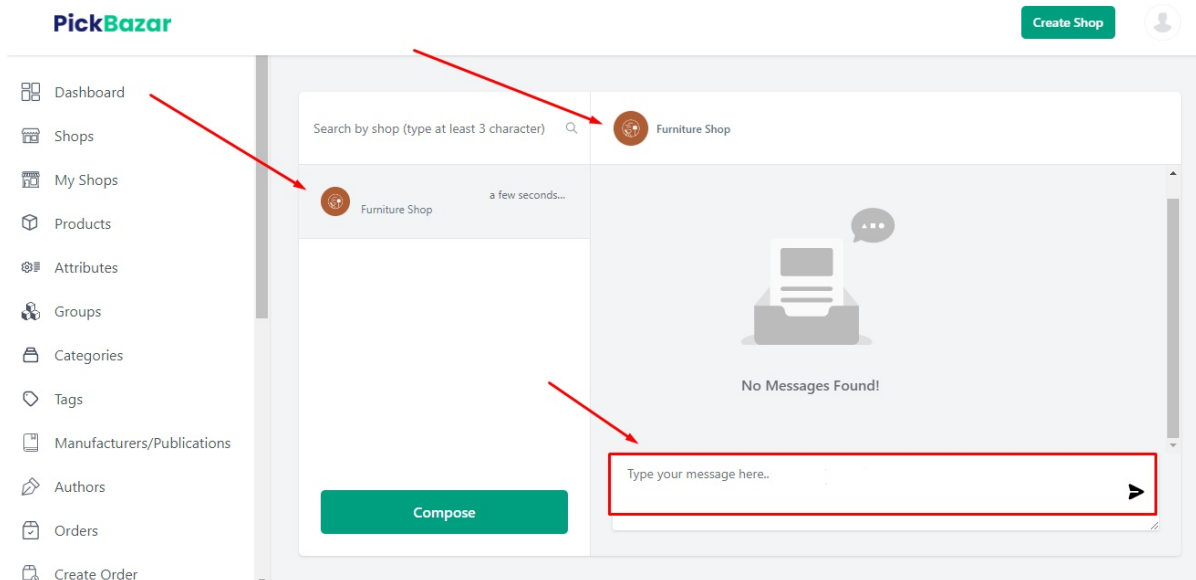
- To start new conversation with a new shop admin click on compose. It will appear a box to select specific shop and start conversation.



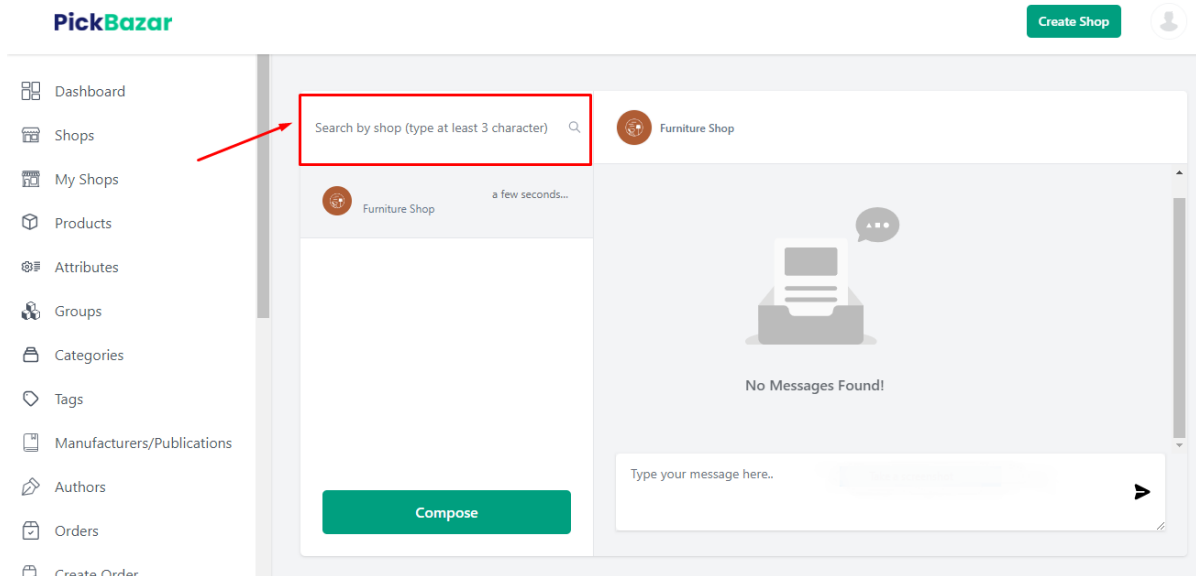
### Starting Chat



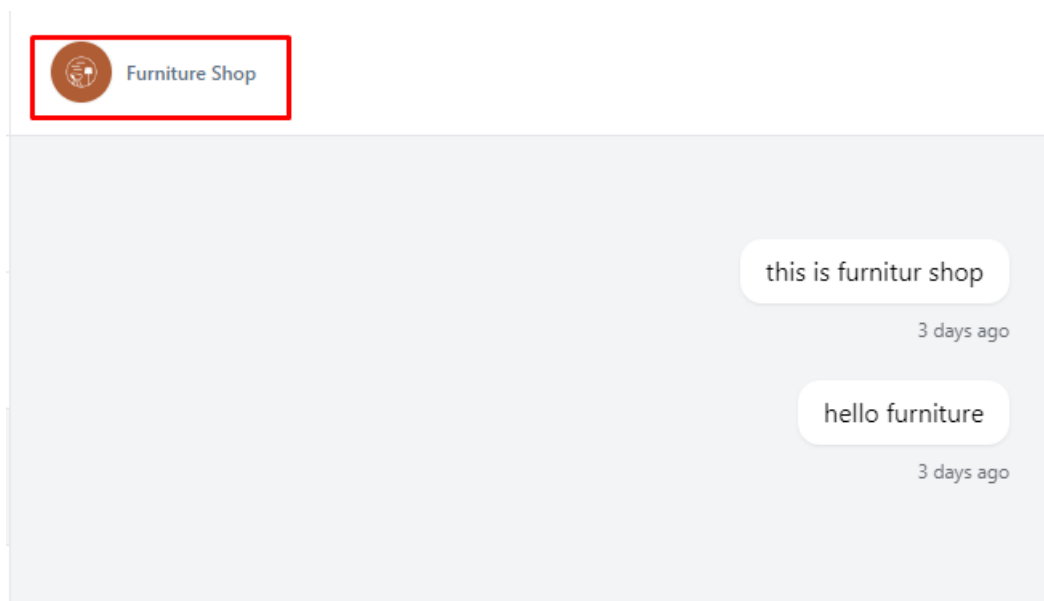
- New conversation has started. Now admin can write a message in the message box and send it to the shop owner by pressing the enter key or by clicking on the send icon.



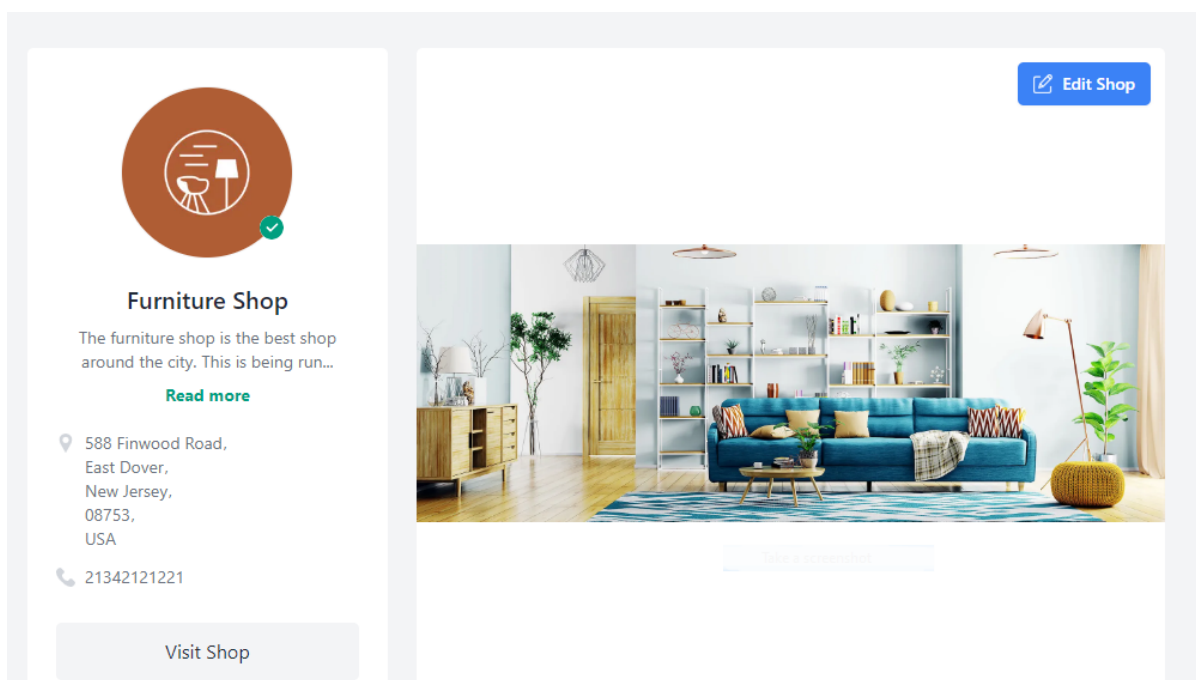
- Admin can search a shop by using the search option at the top message.



- Admin and store-owner can go to specific shop by clicking on the icon of shop.



- Now they will get details of the shop.



- An admin can create stores, but he is unable to message those shops that he has created. He can create a shop after filling up all the required field as following.

The screenshot shows the PickBazar admin dashboard. On the left is a sidebar with navigation links: Dashboard, Shops, My Shops, Products, Attributes, Groups, Categories, Tags, Manufacturers/Publications, Authors, and Orders. The main content area has a top bar with a 'Create Shop' button (highlighted with a red box and arrow) and a user profile icon. Below this is a 'Select Your Conversation' section with a search bar and a 'Compose' button. The 'Basic Info' form is visible, with a red arrow pointing to the 'Name' field, which has a red error message 'Name is required' below it. The 'Description' field is also present.

- Created shop will appear at my shop of admin dashboard which is inactive. the shop created by store owner needs to be active if the admin wants to send a message. Admin will active the created shop as following:

The screenshot shows the 'My Shops' section of the admin dashboard. It displays a 'New Test Shop' card with a placeholder image and an 'Inactive' status. Below this is a table titled 'Shops' with columns: Logo, Name, Owner Name, Products, Orders, Status, and Actions. The table contains one row for 'New Test Shop' owned by 'Admin'. The 'Status' column shows 'Inactive' (highlighted with a red arrow) and the 'Actions' column shows a green checkmark icon (highlighted with a red arrow and the text 'click here to active').

Logo	Name	Owner Name	Products	Orders	Status	Actions
	New Test Shop	Admin	0	0	Inactive	



Admin

admin@demo.com

Please add your [Profile](#)

Enabled

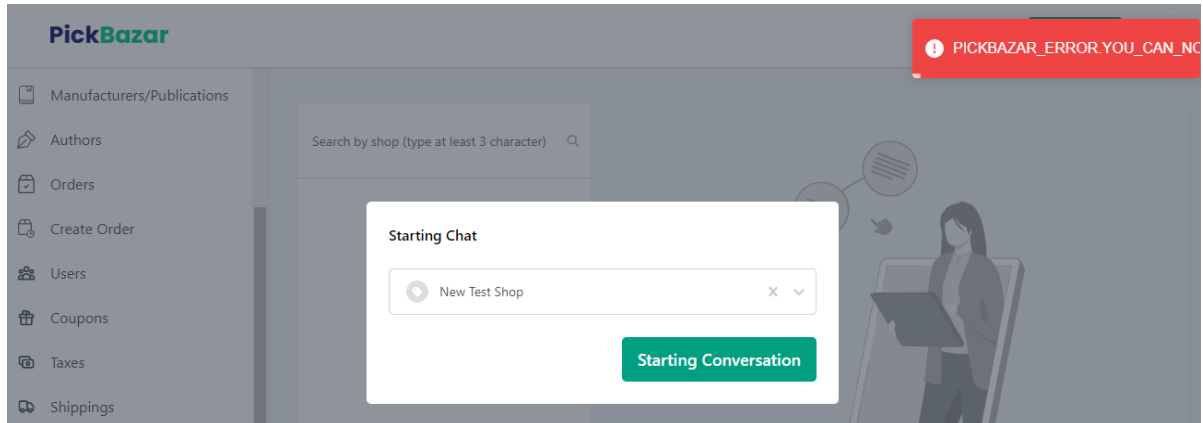
## My Shops



New Test Shop

Active

- But the admin can not send message to his own created shop. So error occurred.



- Now the store owner will create a shop which will added into the store owner shops.



Store Owner

store\_owner@demo.com

12365141641631

Enabled

### Basic Info

Add some basic info about your shop from here

Name

Name is required

Description

PickBazar

Create Shop



Store Owner

store\_owner@demo.com

12365141641631

Enabled

### My Shops

### Message

### Store Notice



Furniture Shop

Active



Clothing Shop

Active



Bags Shop

Active



Makeup Shop

Active



Bakery Shop

Active



Grocery Shop

Active



Books Shop

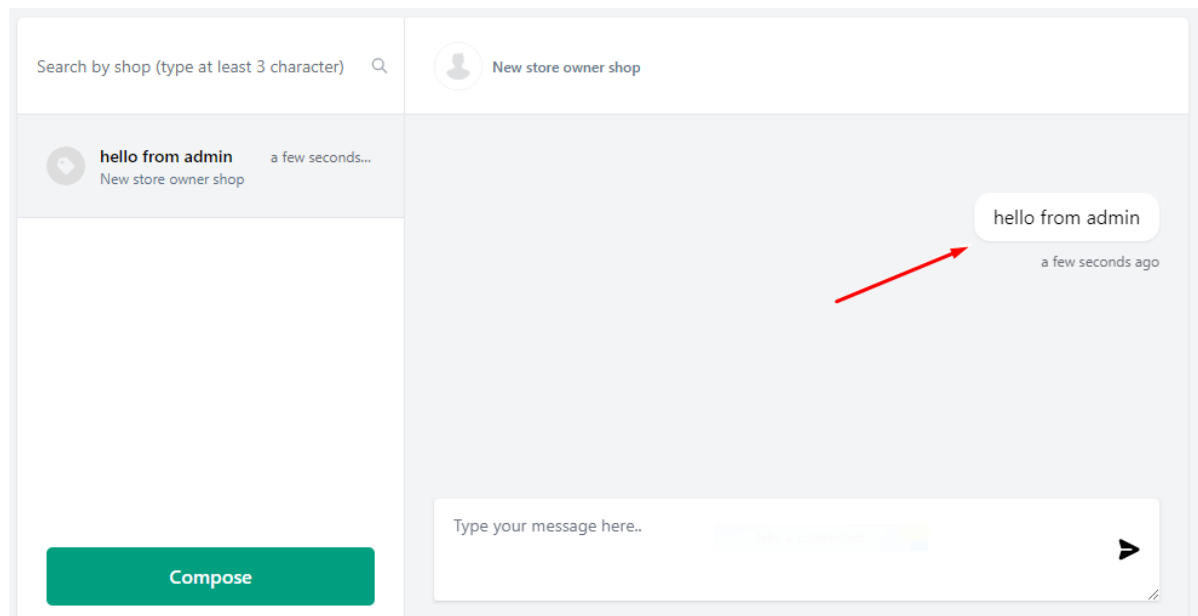
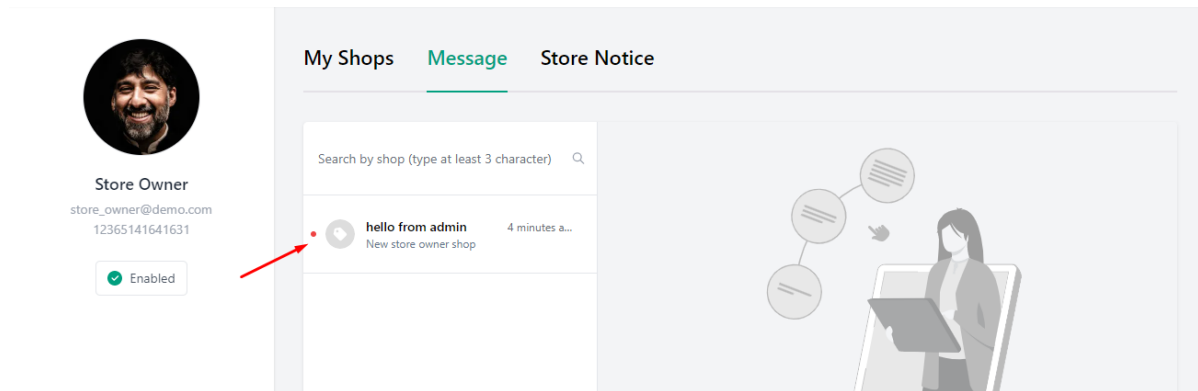
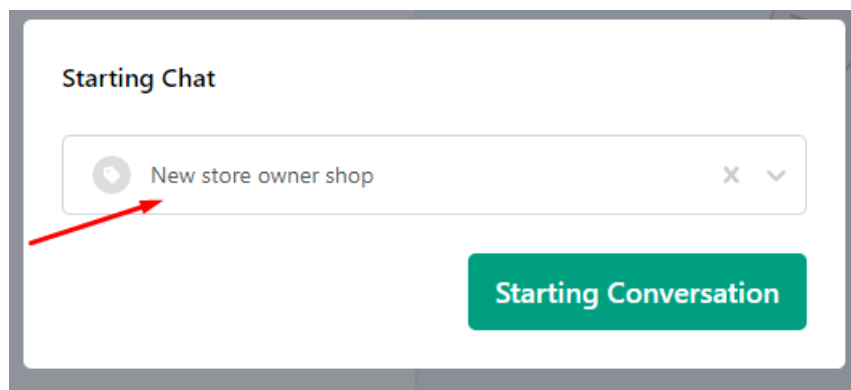
Active



New store owner shop

Active

- Now the admin can send message to the shore owner created shop after activate the shop by the admin.



## Store Notice (Feature)

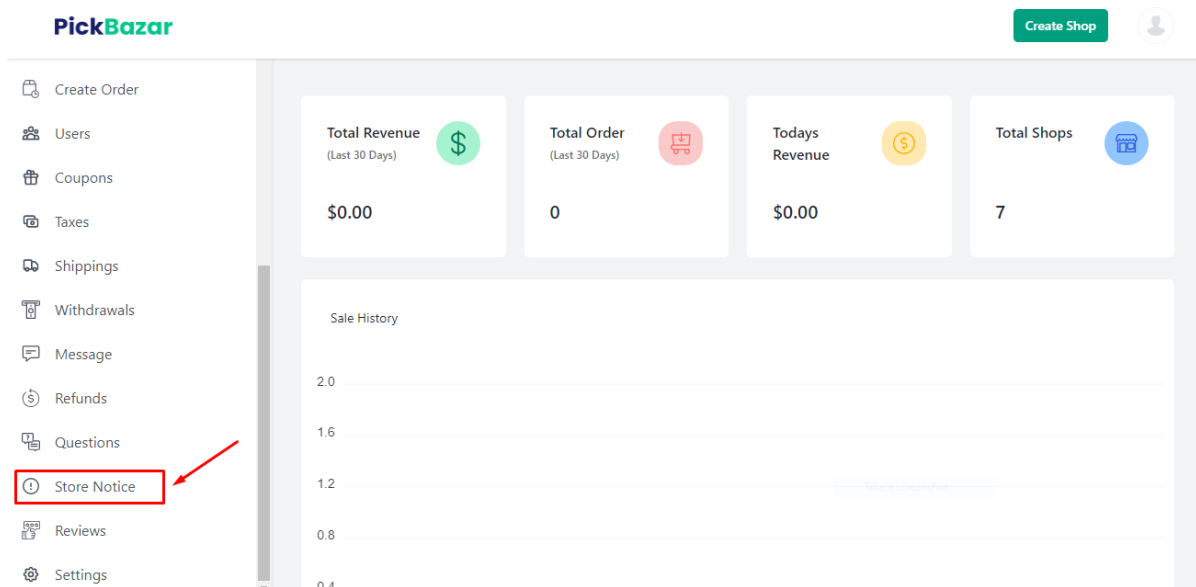
**Store Notice** allows you to enter a site-wide message shown to everyone visiting your store. Send messages to vendors or for the **multivendor** website. The message can be: Informational - **A note that tells customers of shipping delays due to weather conditions.** In **Pickbazar** you can send messages to vendors or all customers with the **colored base priority**. Admin can set notice for specific vendor or all vendors. Vendors can set notice for a specific shop or all shops independently. Notices are set for the shop will be visible to everyone visiting that shop.

- **Store Notice Hierarchy**

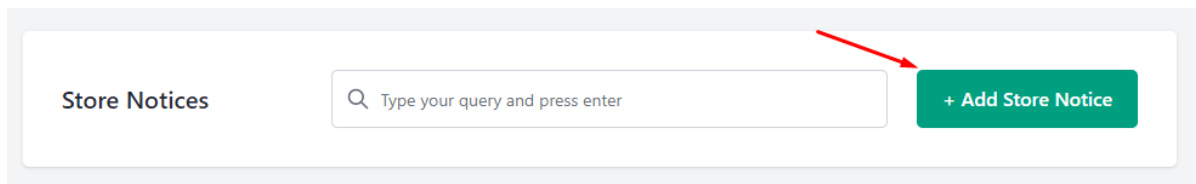
- Admin ⇒ Vendor/All Vendor/Selected Vendors
- Vendor ⇒ Shop
- **Priority**
  - High (Blue)
  - Medium (Yellow)
  - Low (Light Yellow)

## Super Admin

1. At first go to dashboard and find Store Notice like the picture below.



2. Click on add store notice to add new notice.



3. Fill up all the required field and create new notice for vendor. Here is field for a notice title, a notice description, notice priority, active date, expired date and vendor type.

- **Notice Title**: The title of a store notification attracts attention; it should be short and to the point in order to convey the essence of the notice and prevent the merchant from getting lost in a sea of details.
- **Notice description**: Notice descriptions are a type of formal letter that is sent to a certain vendor or set of suppliers. Similar to a news story, it alerts the appropriate party or parties to a significant incident.
- **Notice Priority**: There are three ways to set a priority. Low, medium, and high. Vendors may assign importance based on the urgency of a message.
- **Active Date**: The term "active date" refers to a notice's start date or effective period.
- **Expire Date**: A notice's expiration date indicates that it was over on that particular day.
- **Notice Type**: An individual or business that sells products or services to another is referred to as a vendor or supplier. This vendor may be a general vendor, a particular vendor, or an administrator.

## Notice creation for specific vendor

**Description**  
Add Store Notice necessary information from here

Priority\*  
Medium

Store Notice\*  
Notice to Vendors

Description  
NMS shall notify each Vendor servicing GECS of new enrollments and of the date on which coverage will be effective via electronic file transfer if possible. If a Vendor is unable to accept an electronic file or a smaller number of Company's Employees are enrolled, new enrollment information will be sent via a paper enrollment report. NMS shall use its best efforts to assure that the Vendors receive notification of such enrollment as soon as possible. NMS shall provide notice, in the first weekly transmission of eligibility information after NMS' receipt of information, to the appropriate Vendor when new enrollments occur.

Active  
20/01/2023

Expire  
25/01/2023







Type\*  
Specific Vendor

Received By\*  
Select...



**Add Store Notice**

4. Now the notice will be shown in the store notice list.

5. Any notice can be deleted by using delete icon.

Store Notices								
<div><div>Type your query and press enter</div><div>+ Add Store Notice</div></div>								
ID	Notice	Description	Type	Effective From	Effective At	Issued By	Priority	Actions
22	FSG	Prepare for FSG's approval (which approval ...	All vendor	a day ago	in 11 days	Super admin	LOW	 
21	A letter for each of the ven...	A letter for each of the vendors of the Servic...	All vendor	a day ago	in 4 days	Super admin	MEDIUM	 
20	Notice to Vendors	NMS shall notify each Vendor servicing GEC...	All vendor	a day ago	in 14 days	Super admin	HIGH	 
<div><div>&lt; 1 &gt;</div></div>								

6. Update a notice by following the below picture.

Store Notices								
<div><div>Type your query and press enter</div><div>+ Add Store Notice</div></div>								
ID	Notice	Description	Type	Effective From	Effective At	Issued By	Priority	Actions
14	new notice	description	All vendor	in 9 hours	in 2 days	Super admin	HIGH	 
<div><div>&lt; 1 &gt;</div></div>								

7. Update notice

**Description**  
Edit Store Notice necessary information from here

**Priority\***  
High

**Store Notice\***  
new notice updated

**Description**

**Active**  
19/01/2023

**Expire**  
26/01/2023

**Type\***  
All Vendor

Back

Update Store Notice

8. According to priority colors are shown in priority column.

- Priority
  - High (Blue)
  - Medium (Yellow)
  - Low (Light Yellow)

ID	Notice	Description	Type	Effective From	Effective At	Issued By	Priority	Actions
16	Medium priority	description for mdium priority	Specific vendor	15 hours ago	in 2 days	Super admin	LOW	
15	High priority	description	Specific vendor	in 5 days	in 11 days	Super admin	HIGH	
14	new notice for medium pri...	description	All vendor	in 3 hours	in 2 days	Super admin	MEDIUM	

## Vendor

9. Store Owner will get this notice in notice section.

### Store Notice Vendor profile

Notices for vendor from admin will appear here. Expired notices will disappear from here automatically.

My Shops
Message
Store Notice

**Notice to Vendors** HIGH

NMS shall notify each Vendor servicing GECS of new enrollments and of the date on which coverage will be effective via electronic file transfer if possible. If a Vendor is unable to accept an electronic file or a smaller number of Company's Employees...

[Read more](#)

**A letter for each of the vendors** MEDIUM

A letter for each of the vendors of the Service Contracts in the form attached hereto as Exhibit "D," duly executed by Seller, dated as of the Closing Date and addressed to each Service Contract vendor, informing said vendors of the assignment of the...

[Read more](#)

**FSG** LOW

Prepare for FSG's approval (which approval shall not be unreasonably withheld or delayed) a written notice to be delivered to all vendors of FSG notifying such vendors of the termination of this Agreement...

[Read more](#)

10. New notice will add at the top of the notice list. You will see a read more option to see detail notice. Detail notice will appear into a new page. Go back option can be used to go back to the previous notice list. The notice will show a blur color if it is seen.

## Staff

### Create a Notice for perticular shop

Creation a notice for [furniture-shop]

Create Store Notice

Description  
Add Store Notice necessary information from here

Priority\*

Medium

Store Notice\*

Offer

Description

Buy one Get one Offer from 20/01/2023 to 25/01/2023.

Active

20/01/2023

Expire

25/01/2023

Add Store Notice

### Issued notice for a specific shop [furniture-shop]

Store Notices							
<div><div>Q</div>Type your query and press enter</div> <div>+ Add Store Notice</div>							
ID	Notice	Description	Type	Effective From	Effective At	Issued By	Priority
23	Offer	Buy one Get one Offer from 20/01/2023 to ...	Specific shop	2 hours ago	in 5 days	Store owner	MEDIUM

<

1

>

### Store Notice staff profile

Staff for [furniture-shop]

Staff

store\_staff@demo.com

Please add your [Profile](#)

Enabled

My Shops

Message

Store Notice

Offer

MEDIUM

Buy one Get one Offer from 20/01/2023 to 25/01/2023...

[Read more](#)

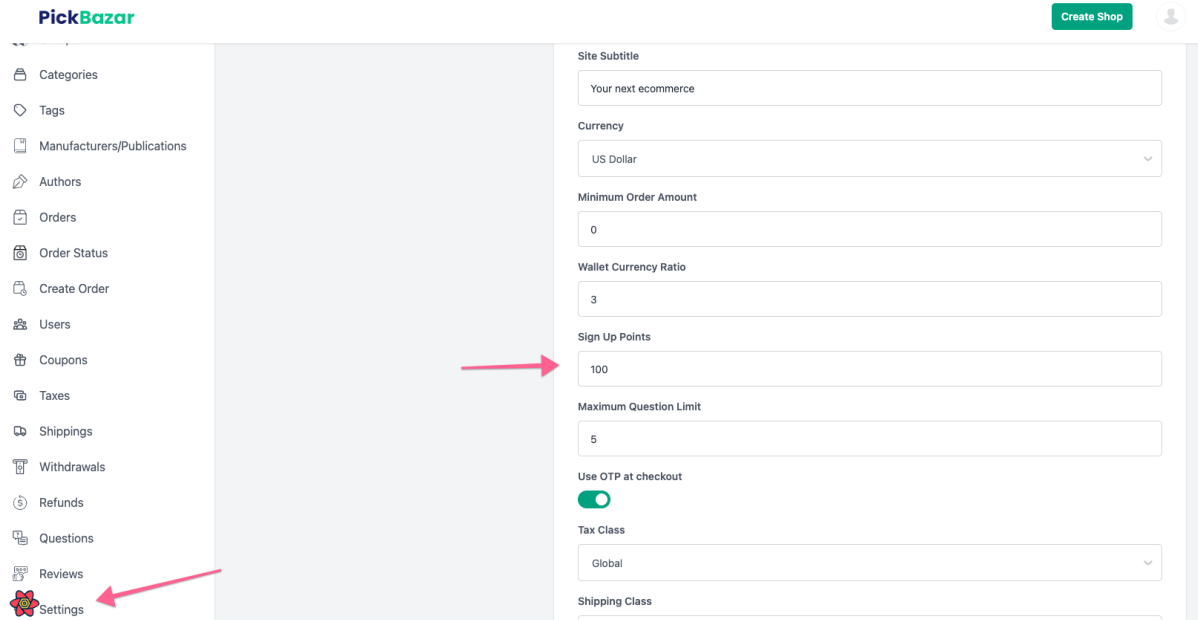
## Digital Wallet (Feature)

The wallet is a virtual currency that a customer can use for purchase items. On pickbazar, there are three ways that can be used to generate wallet points for a customer.

Sign up points.  
Manually by admin  
Refund

## Sign Up Points

When new users sign up, they will get signup rewards points for the signup. You can set that signup points from `admin -> settings`



**PickBazar** Create Shop

- Categories
- Tags
- Manufacturers/Publications
- Authors
- Orders
- Order Status
- Create Order
- Users
- Coupons
- Taxes
- Shippings
- Withdrawals
- Refunds
- Questions
- Reviews
- Settings**

Site Subtitle  
Your next ecommerce

Currency  
US Dollar

Minimum Order Amount  
0

Wallet Currency Ratio  
3

Sign Up Points  
100

Maximum Question Limit  
5

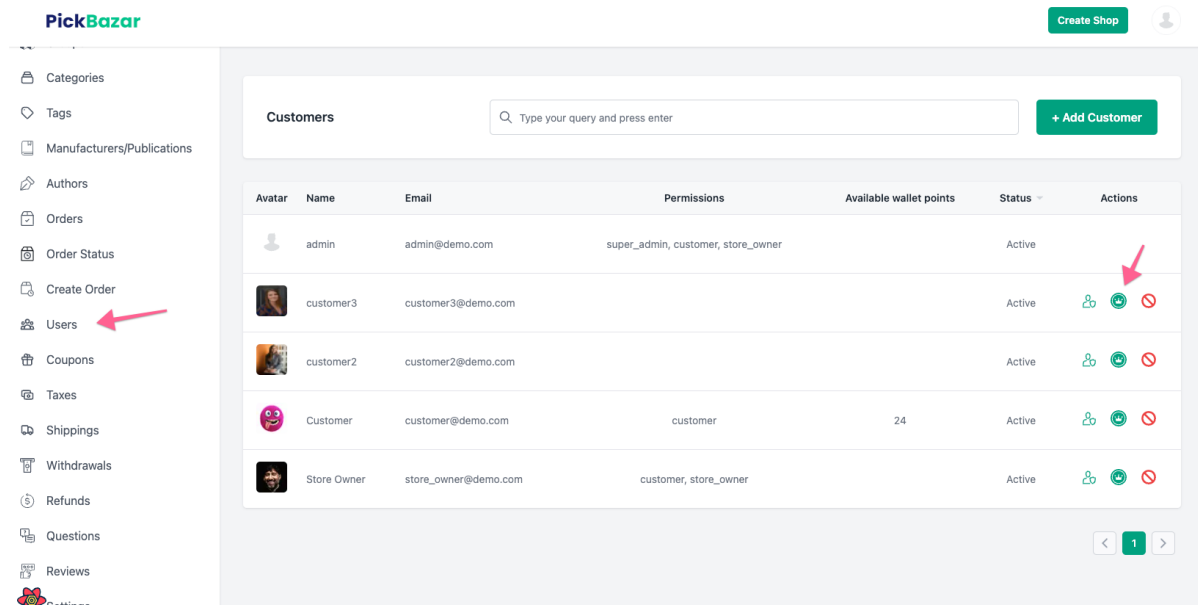
Use OTP at checkout  
☒

Tax Class  
Global

Shipping Class

## Manually By Admin

You can give wallet points to a specific user from `admin -> users`



**PickBazar** Create Shop

**Customers**  + Add Customer

Avatar	Name	Email	Permissions	Available wallet points	Status	Actions
	admin	admin@demo.com	super_admin, customer, store_owner		Active	
	customer3	customer3@demo.com			Active	
	customer2	customer2@demo.com			Active	
	Customer	customer@demo.com	customer	24	Active	
	Store Owner	store_owner@demo.com	customer, store_owner		Active	

< 1 >

PickBazar

Create Shop

Categories

Tags

Manufacturers/Publications

Authors

Orders

Order Status

Create Order

Users

Coupons

Taxes

Shippings

Withdrawals

Refunds

Questions

Reviews

Settings

Customers

+ Add Customer

Avatar	Name	Email	Permissions	Available wallet points	Status	Actions
	admin	admin@demo.com	super admin, customer, store_owner		Active	
	customer3	customer3@demo.com			Active	
	customer2	customer2@demo.com			Active	
	Customer	customer@demo.com	customer	24	Active	
	Store Owner	store_owner@demo.com	customer, store_owner		Active	

<

1

>

Add Wallet Points

Submit

## Refund

The third and final way is a refund. When the customer asks for a refund for the order, and you as an admin approve it, then the amount will be delivered to the customer as a wallet point.

PickBazar

Create Shop

Categories

Tags

Manufacturers/Publications

Authors

Orders

Order Status

Create Order

Users

Coupons

Taxes

Shippings

Withdrawals

Refunds

Questions

Reviews

Settings

Refund ID - 1 (Approved)

Approved

Change Status

Tracking Number : cWz1Dw7Tq1Yw

Order Status : Order Received

Customer Email : customer@demo.com

Contact Number : 19365141641631

Refund Created : 24 May 2022

Order Created : 24 May 2022

Reason

Emni

Description

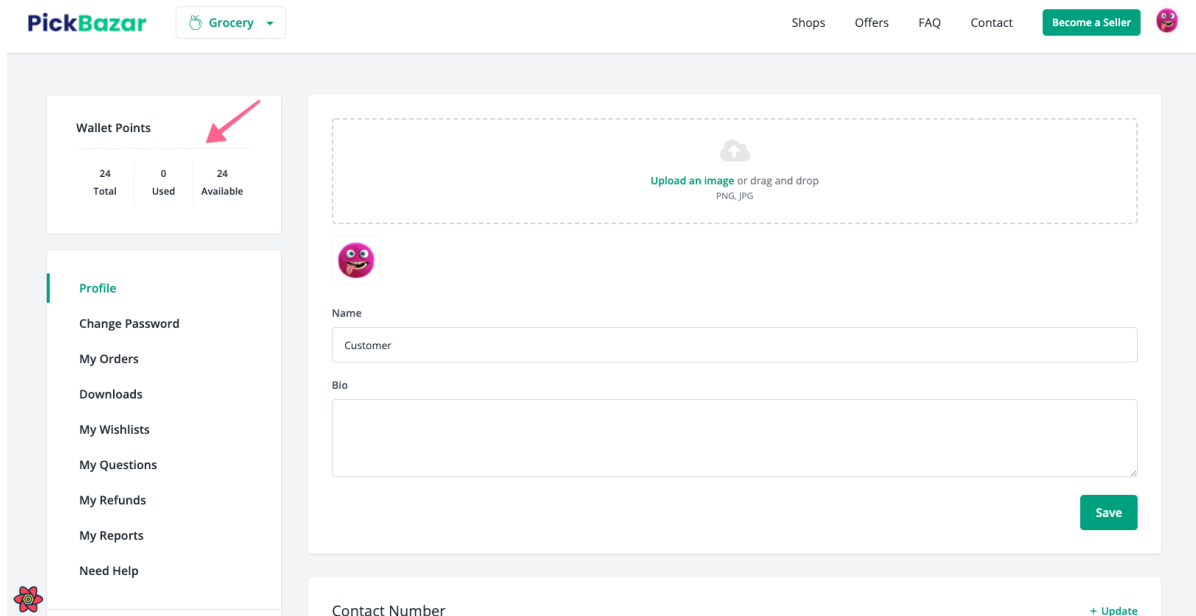
No Reason

Images

No Image found

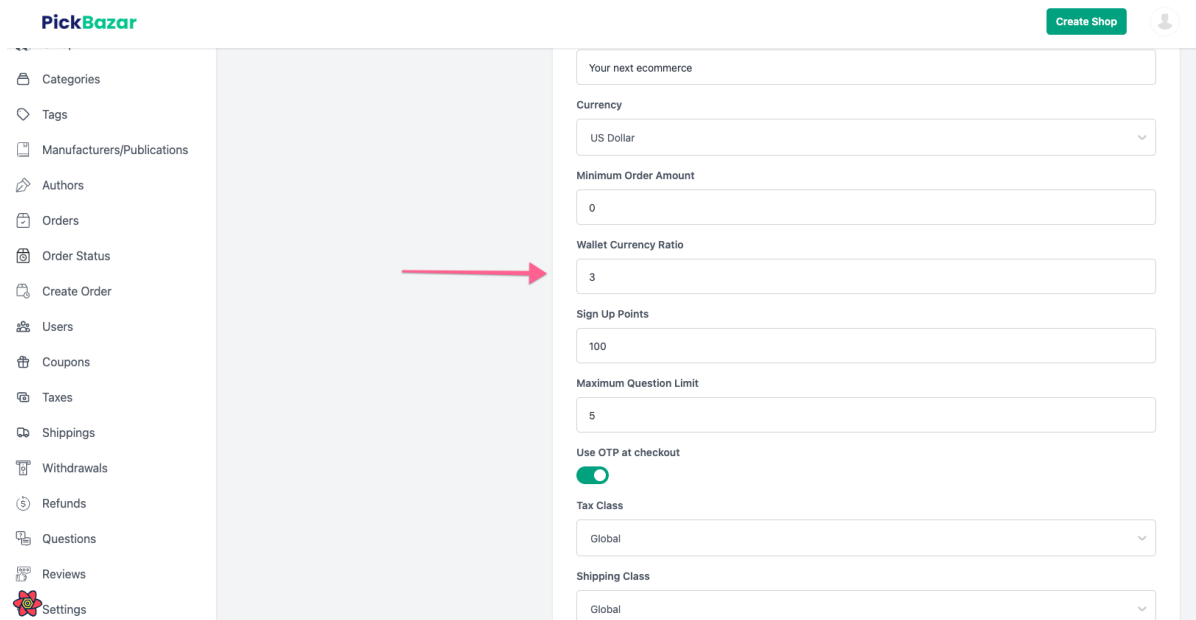
Order Details

Products	Total
Apples x 1	\$1.60



## Wallet Ratio

You can set the conversion ratio for currency amount with wallet from `admin -> settings`



For example, you set `3` for the currency ratio. Now, if the customer asks for a refund that is `8$` price and you approve it, then the customer will get `8x3=24` wallet points. Similarly, if the customer tries to use this wallet to purchase an `8$` item, then the wallet deduction will be like this, `24/3=8$`.

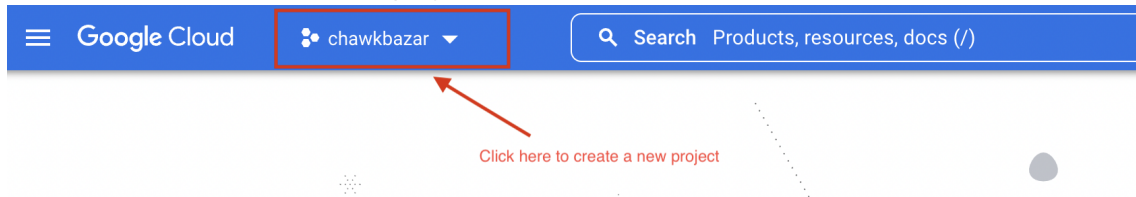
## Social Login (Feature)

You can use a social network like `Google` or `Facebook` as a login provider like email/username and password. To set up social login, follow the below procedure,

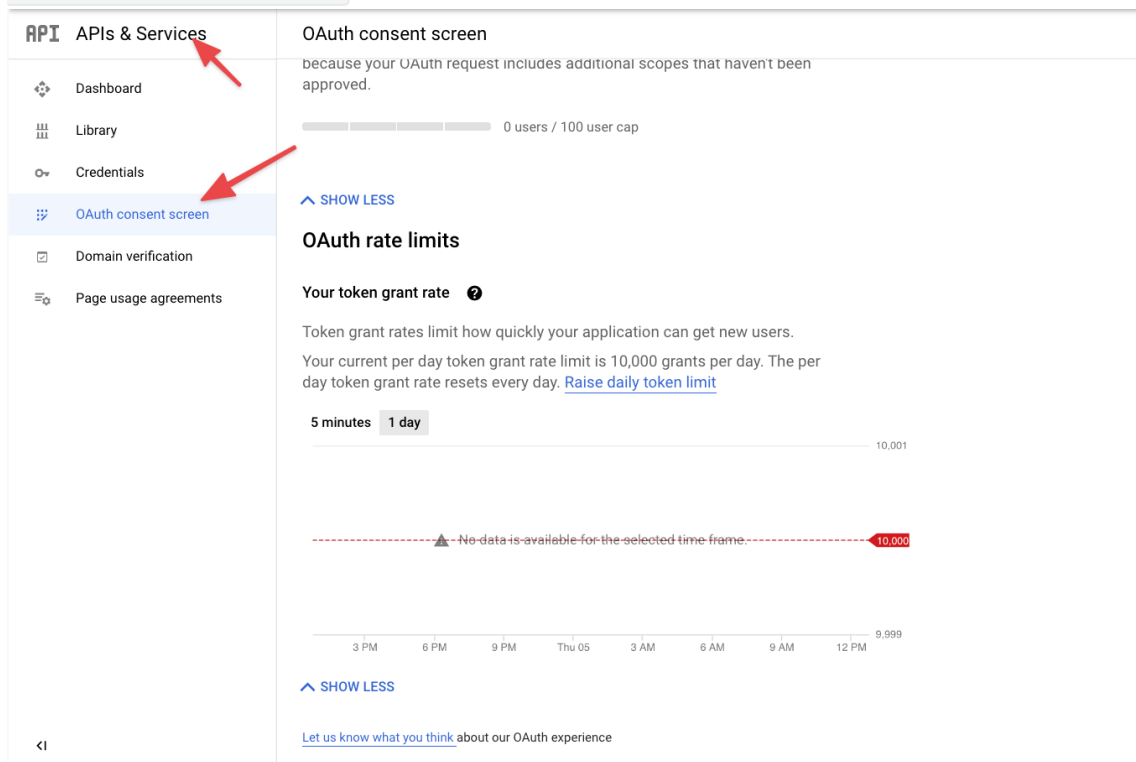
# Google

For Google, follow this procedure,

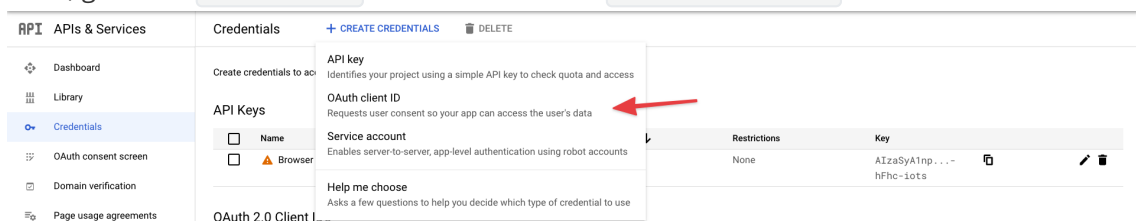
1. At first go to Google Console dashboard (<https://console.cloud.google.com/>)
2. From the console, create a new project



3. After creating the project, go to **APIs & Services**  
-> **OAuth consent screen**.



4. Create an **External OAuth Consent**. Add all the required fields.
5. Then, go to the **credentials** section and create **OAuth Client ID**.



Add an **Authorized redirect URIs**. The format of the URIs should be:

<http://yourdomain.com/api/auth/callback/google>

API APIs & Services

Enabled APIs & services

Library

Credentials

OAuth consent screen

Domain verification

Page usage agreements

Client ID for Web application

DOWNLOAD JSON

RESET SE

Name \*

Web client 1

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorized domains](#).

Authorized JavaScript origins ?

For use with requests from a browser

+ ADD URI

Authorized redirect URIs ?

For use with requests from a web server

URIs 1 \*

https://yourdomain.com/api/auth/callback/google

+ ADD URI

6. Then you'll get your credentials,

OAuth client created

The client ID and secret can always be accessed from Credentials in APIs & Services

OAuth is limited to 100 [sensitive scope logins](#) until the [OAuth consent screen](#) is verified. This may require a verification process that can take several days.

Your Client ID

211044409169-1 [redacted]

Your Client Secret

WFqrABH3N [redacted]

DOWNLOAD JSON

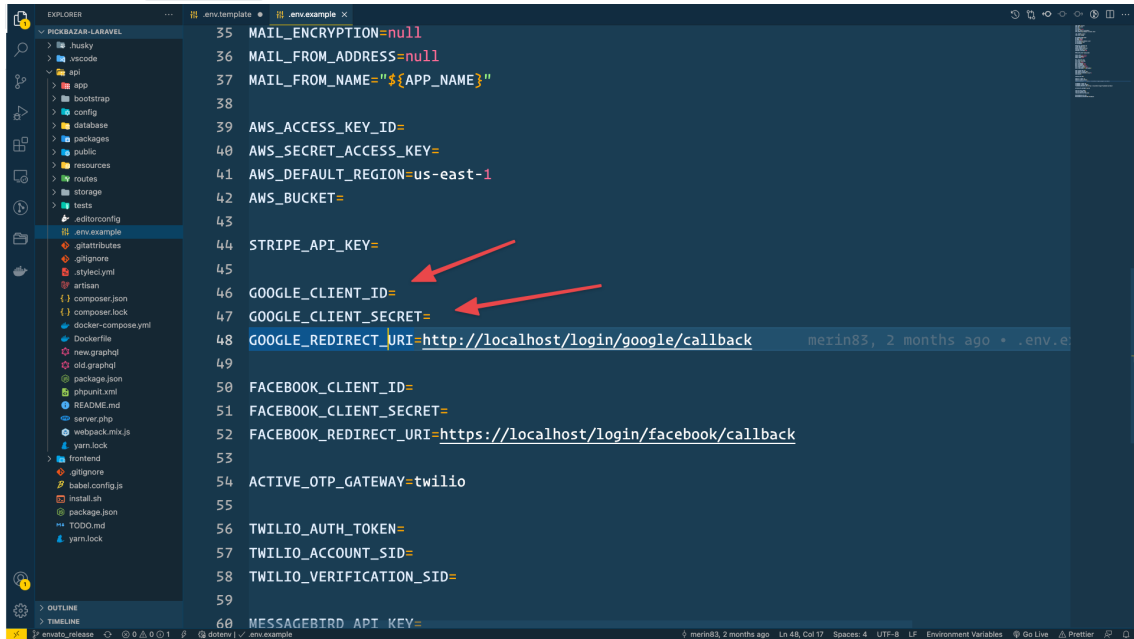
OK

Client ID

211044409169-1 [redacted]

211044409169-1 [redacted]

7. Then open `api/.env` and add API ID and credentials,



```
35 MAIL_ENCRYPTION=null
36 MAIL_FROM_ADDRESS=null
37 MAIL_FROM_NAME="${APP_NAME}"
38
39 AWS_ACCESS_KEY_ID=
40 AWS_SECRET_ACCESS_KEY=
41 AWS_DEFAULT_REGION=us-east-1
42 AWS_BUCKET=
43
44 STRIPE_API_KEY=
45
46 GOOGLE_CLIENT_ID=
47 GOOGLE_CLIENT_SECRET=
48 GOOGLE_REDIRECT_URI=http://localhost/login/google/callback
49
50 FACEBOOK_CLIENT_ID=
51 FACEBOOK_CLIENT_SECRET=
52 FACEBOOK_REDIRECT_URI=https://localhost/login/facebook/callback
53
54 ACTIVE_OTP_GATEWAY=twilio
55
56 TWILIO_AUTH_TOKEN=
57 TWILIO_ACCOUNT_SID=
58 TWILIO_VERIFICATION_SID=
59
60 MESSAGEBIRD_API_KEY=
```

8. Then go to this <https://generate-secret.now.sh/32> site and generate a secret key.

9. Then open `shop/.env` and add credentials to `GOOGLE_CLIENT_ID` and `GOOGLE_CLIENT_SECRET` and `NEXTAUTH_URL` also add a `SECRET` code.

```
NEXTAUTH_URL=YOUR_SHOP_DOMAIN
GOOGLE_CLIENT_ID=
GOOGLE_CLIENT_SECRET=
SECRET=
```

After configuration, make sure you rebuild your project using this command,

For REST API

```
yarn build:shop-rest
yarn build:admin-rest
```

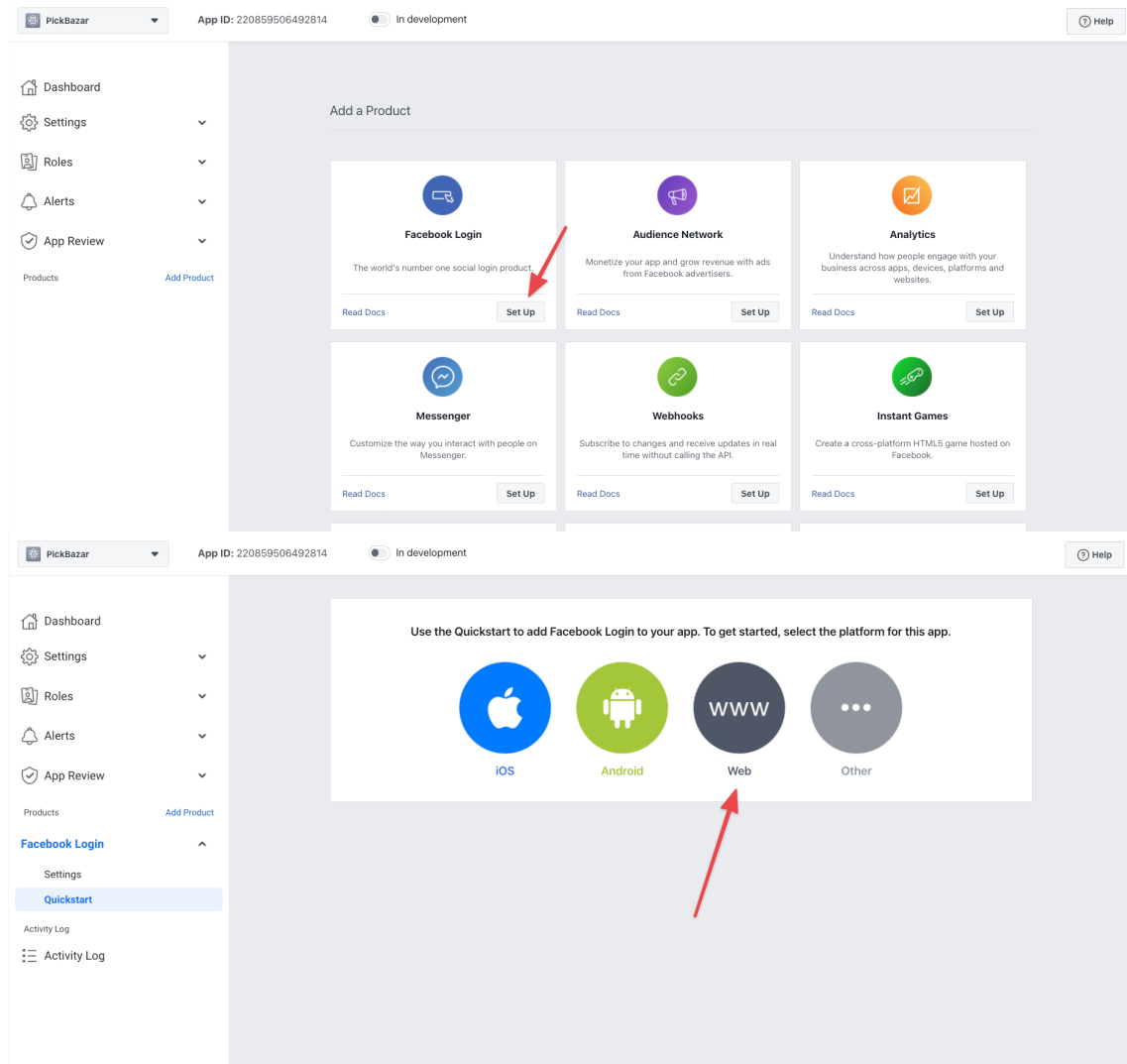
And for GraphQL,

```
yarn build:shop-gql
yarn build:admin-gql
```

## Facebook

1. Go to Facebook developer dashboard (<https://developers.facebook.com/apps/>)
2. Then create a new app for pickbazar

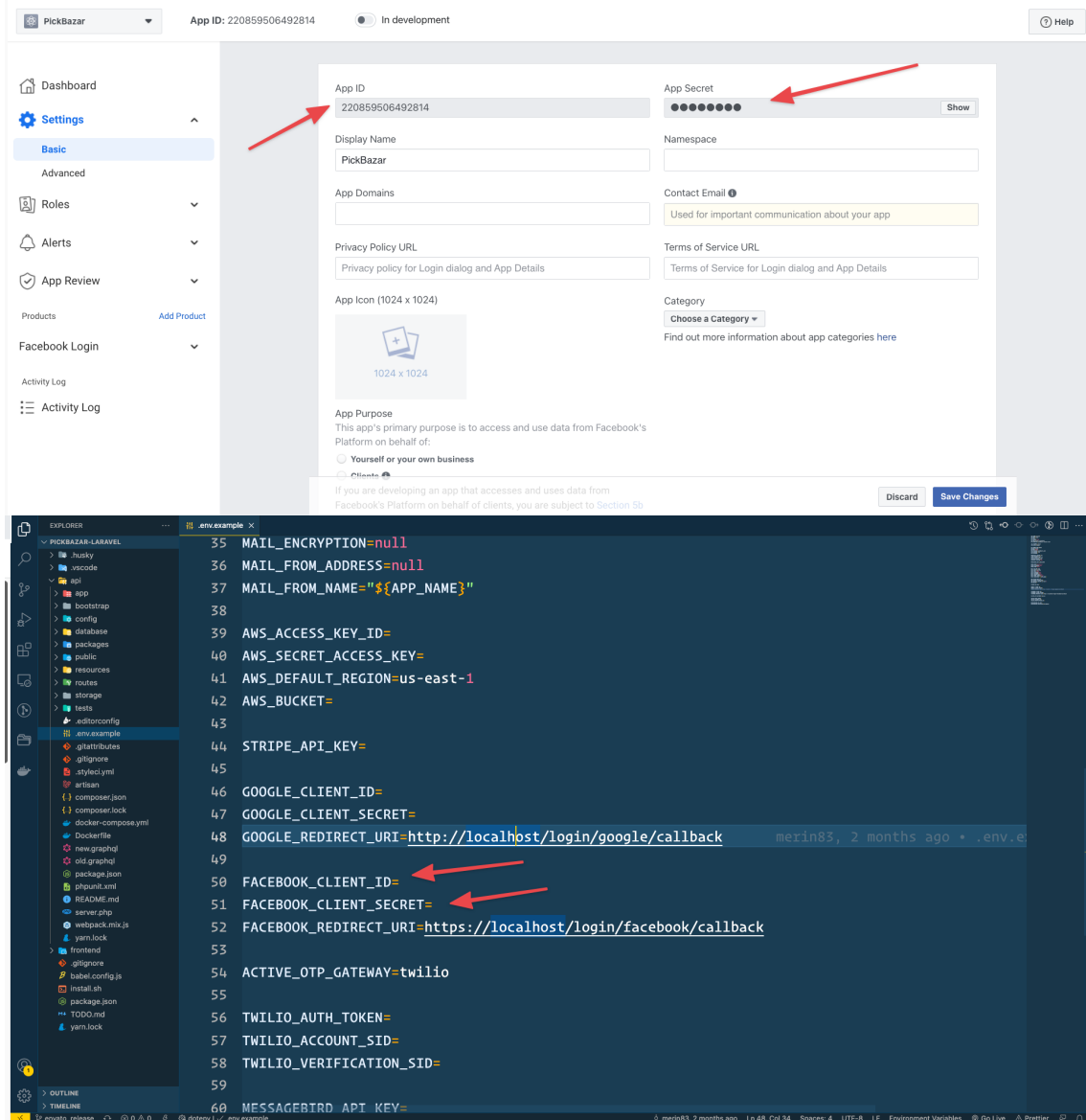
### 3. After that, set up Facebook Login.



4. Then provide all the necessary information.

5. After that, go to the settings page, and on that page, you'll get all the credentials.

6. Then open `api/.env` and add API ID and credentials,



7. Then go to this <https://generate-secret.now.sh/32> site and generate a secret key.

8. Then open `shop/.env` and add credentials to `FACEBOOK_CLIENT_ID`, `FACEBOOK_CLIENT_SECRET`, `NEXTAUTH_URL` also add a `SECRET` code.

```
NEXTAUTH_URL=YOUR_SHOP_DOMAIN
FACEBOOK_CLIENT_ID=
FACEBOOK_CLIENT_SECRET=
SECRET=
```

For Facebook, make sure you apply verification for the APP. Otherwise, the Facebook login won't work. (<https://developers.facebook.com/docs/app-review/>)

After configuration, make sure you rebuild your project using this command,

For REST API

```
yarn build:shop-rest
yarn build:admin-rest
```

And for GraphQL,

```
yarn build:shop-gql
yarn build:admin-gql
```

# OTP - Mobile Number Verification (Feature)

We implement the `OTP` feature on PickBazar `v2.2.0`. So if you want to use `OTP` with `PickBazar`, then make sure your `PickBazar` is `v2.2.0` or later.

With the `OTP` feature, you can do,

- Login with Mobile Number
- OTP verification for updating Mobile Number in profile update section
- OTP verification during checkout process in customer contact section

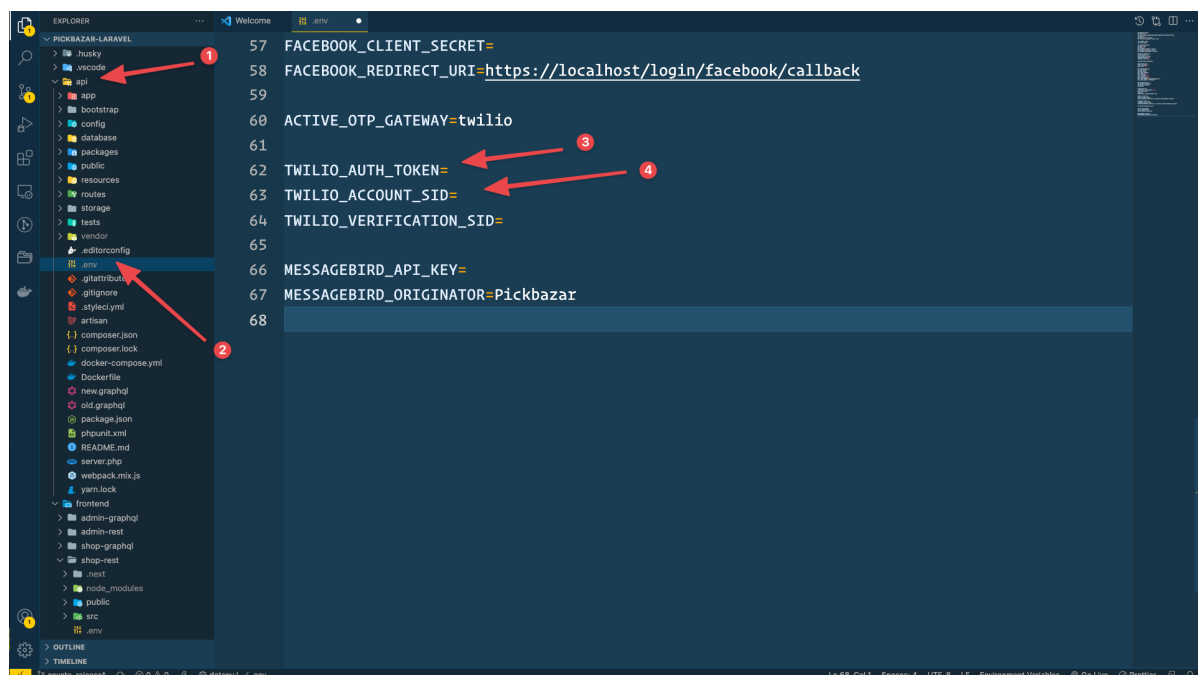
## Configuration

As of today, we implement two service providers for `OTP`. One is `Twilio`, and another is `MessageBird`.

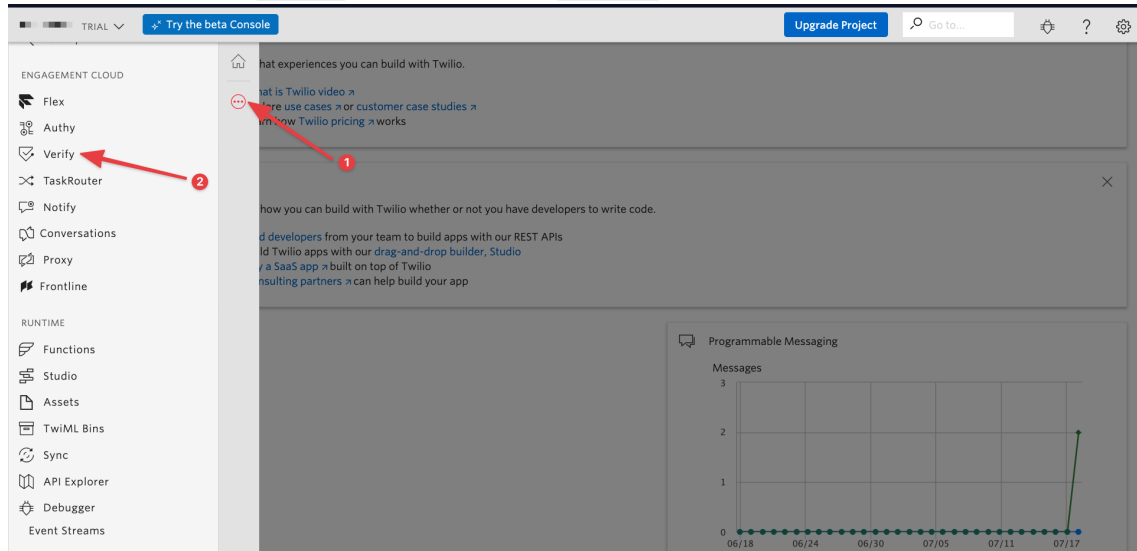
### Twilio Configuration:

If you want to use `Twilio` as your `OTP` service provider, then follow this procedure,

1. Create an account at [Twilio](#)
2. After that, go to the `Twilio Console Panel` and copy the `ACCOUNT SID`, and `AUTH TOKEN` key and add this key to your `api/.env` file

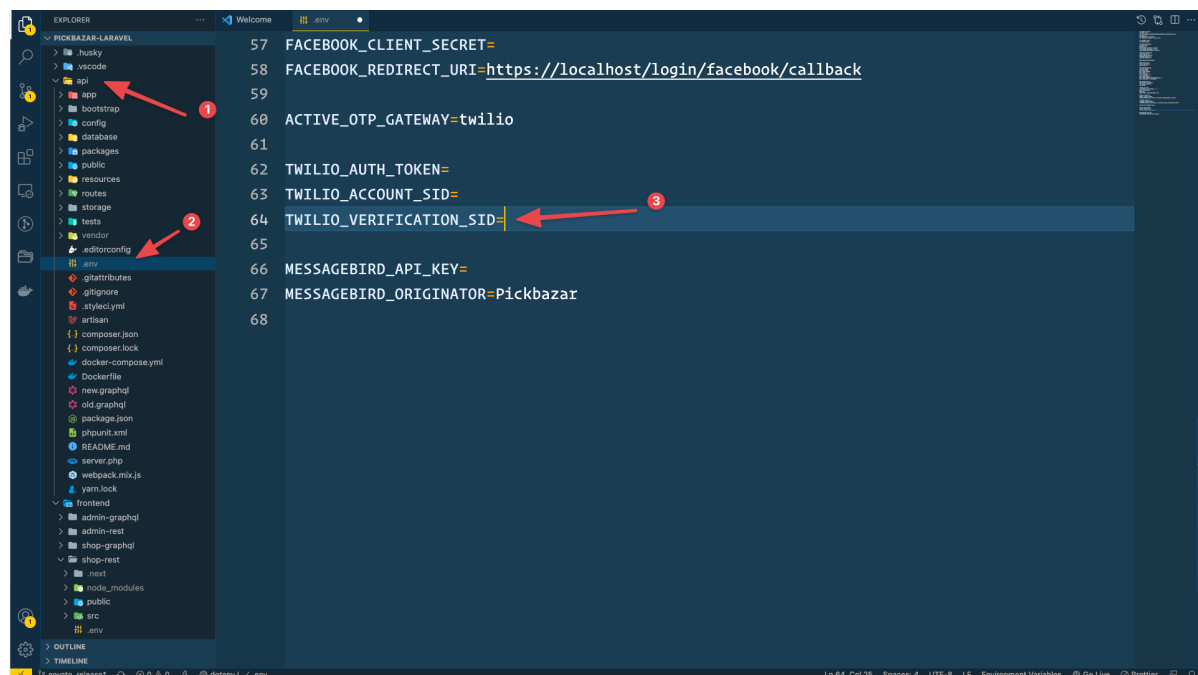
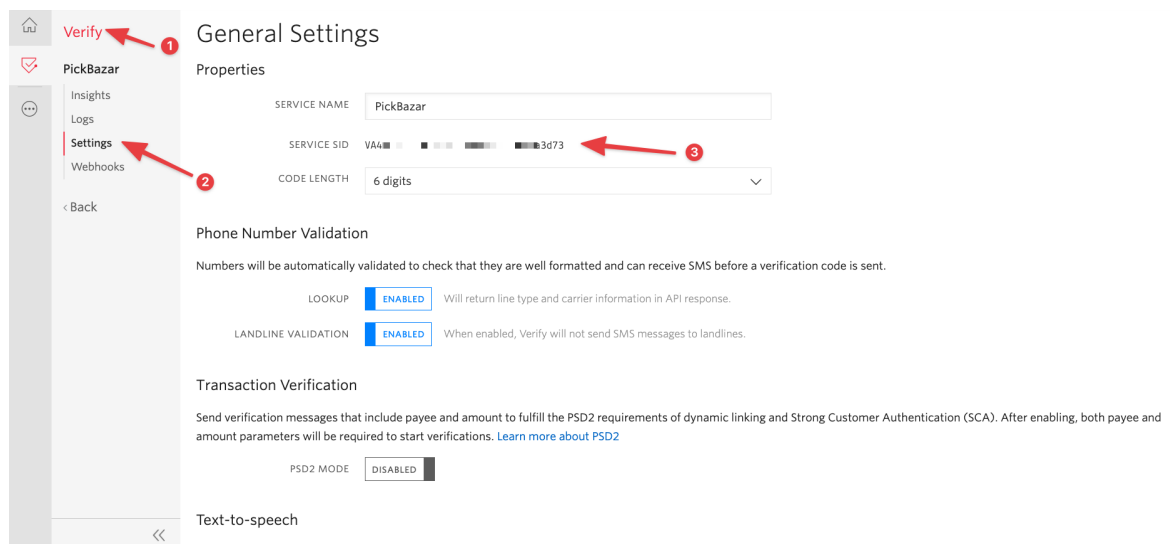


3. After that, go to the `verify` option from the `Sidebar` panel,



4. And from `verify` dashboard, create a service.

5. Add add that service key to `api/.env -> TWILIO_VERIFICATION_SID`



# Verify Geo Permissions

To enable an SMS system in your country, go to this link,

<https://www.twilio.com/docs/verify/preventing-toll-fraud/verify-geo-permissions>

and enable your country's `SMS Channel` and `Voice Channel`.

## Verify geographic permissions <sup>✦Beta</sup>

Select the countries you expect to send verification messages to. All messages sent to countries that are not enabled will be blocked. Defining your enabled countries helps to prevent unexpected high voice or messaging costs caused by malicious actors creating unwanted verification requests. [Learn more.](#)

Q

Search by country name or country code...

Save

Country	<input checked="" type="checkbox"/> Enable SMS channel	<input checked="" type="checkbox"/> Enable voice channel	Last updated
Afghanistan (+93)	<input type="checkbox"/> SMS channel	<input type="checkbox"/> Voice channel	-
Aland Islands (+35818)	<input checked="" type="checkbox"/> SMS channel	<input checked="" type="checkbox"/> Voice channel	-
Albania (+355)	<input checked="" type="checkbox"/> SMS channel	<input checked="" type="checkbox"/> Voice channel	-
Algeria (+213)	<input type="checkbox"/> SMS channel	<input type="checkbox"/> Voice channel	-
American Samoa (+1684)	<input checked="" type="checkbox"/> SMS channel	<input checked="" type="checkbox"/> Voice channel	-
Andorra (+376)	<input checked="" type="checkbox"/> SMS channel	<input checked="" type="checkbox"/> Voice channel	-
Angola (+244)	<input type="checkbox"/> SMS channel	<input type="checkbox"/> Voice channel	-
Anguilla (+1264)	<input checked="" type="checkbox"/> SMS channel	<input checked="" type="checkbox"/> Voice channel	-
Antarctica (+672)	<input checked="" type="checkbox"/> SMS channel	<input checked="" type="checkbox"/> Voice channel	-
Antigua and Barbuda (+1268)	<input checked="" type="checkbox"/> SMS channel	<input checked="" type="checkbox"/> Voice channel	-

1

2345...26

→

1-10 of 252 items

Without this verification, no SMS will be delivered.

With that, your configuration is complete.

## MessageBird:

Use only one service provider for OTP service, so use this `MessageBird` configuration only if you want to use `MessageBird` instead of `Twilio`.

To configure `MessageBird`, follow this procedure,

1. Create an account from [MessageBird](#)

2. After creating the account, go to the **API Getting Started** and follow all the procedures step by step.

Contacts  
Numbers  
SMS  
API Getting Started  
SMS Overview  
Campaign Builder  
Quick Send  
Bulk Messages  
Voice  
WhatsApp  
Verify  
Channels  
Flow Builder  
Inbox  
Solutions  
Knowledge Bases  
Inte Cookie Settings

### API Getting Started

Support Pricing

#### Your getting started guide

SMS API • 4 steps left **20%**

- ✓ Create your account
- 2 Get started with free test credits
- 3 Install the official SDK for SMS
- 4 Send your first SMS
- 5 Go live with SMS

3. After completing the profile, copy the **API** key from **MessageBird** and paste it to **API/.env**

Contacts  
Numbers  
SMS  
API Getting Started  
SMS Overview  
Campaign Builder  
Quick Send  
Bulk Messages  
Voice  
WhatsApp  
Verify  
Channels  
Flow Builder  
Inbox  
Solutions  
Knowledge Bases  
Cookie Settings

### Your getting started guide

SMS API • 4 steps left **20%**

- ✓ Create your account
- 2 Get started with free test credits
- 3 Install the official SDK for SMS
- 4 Send your first SMS
- 5 Go live with SMS

#### Your API Keys

Live ..... Show

0 Free SMS Remaining ..... Show

Use this API Key in production. It works with all of our APIs.

Test ..... Show

Use this API Key in your test environment. It will return a response but not work in production or deduct balance.

[Manage API Keys](#)

#### Resources

- [SMS API Quickstarts](#)
- [SMS API Tutorials](#)

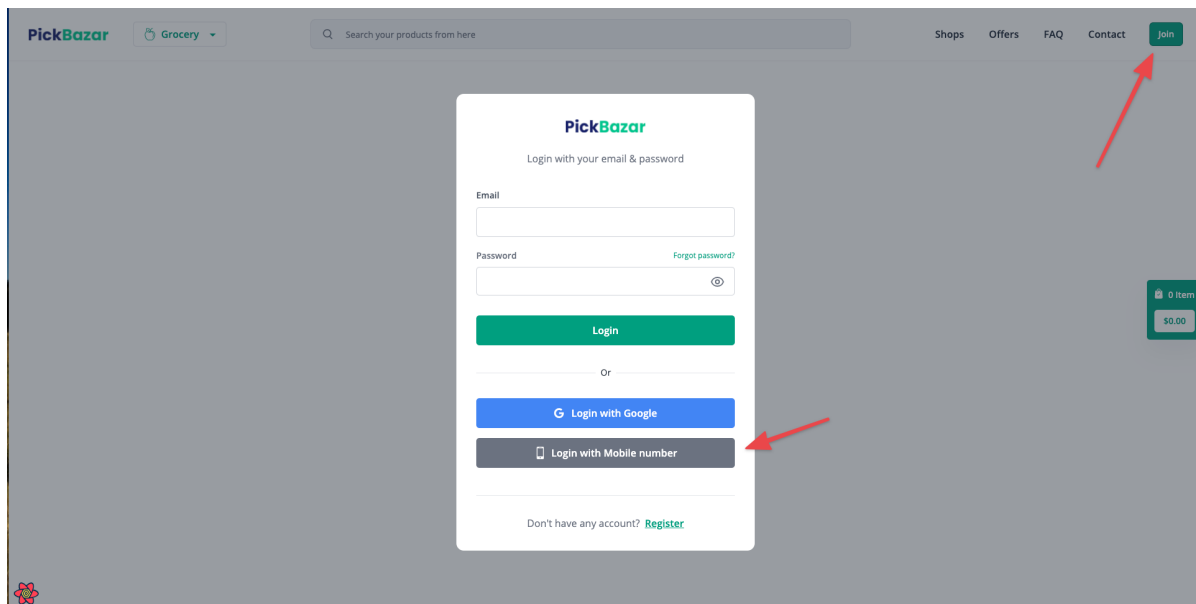
```
48 MEDIA_DISK=public
49 SHOP_URL=
50 ADMIN_EMAIL=support@example.com
51
52 GOOGLE_CLIENT_ID=
53 GOOGLE_CLIENT_SECRET=
54 GOOGLE_REDIRECT_URI=http://localhost/login/google/callback
55
56 FACEBOOK_CLIENT_ID=
57 FACEBOOK_CLIENT_SECRET=
58 FACEBOOK_REDIRECT_URI=https://localhost/login/facebook/callback
59
60
61 ACTIVE_OTP_PROVIDER=messagebird
62
63 TWILIO_AUTH_TOKEN=
64 TWILIO_ACCOUNT_SID=
65 TWILIO_VERIFICATION_SID=
66
67 MESSAGEBIRD_API_KEY=
68 MESSAGEBIRD_ORIGINATOR=Pickbazar
```

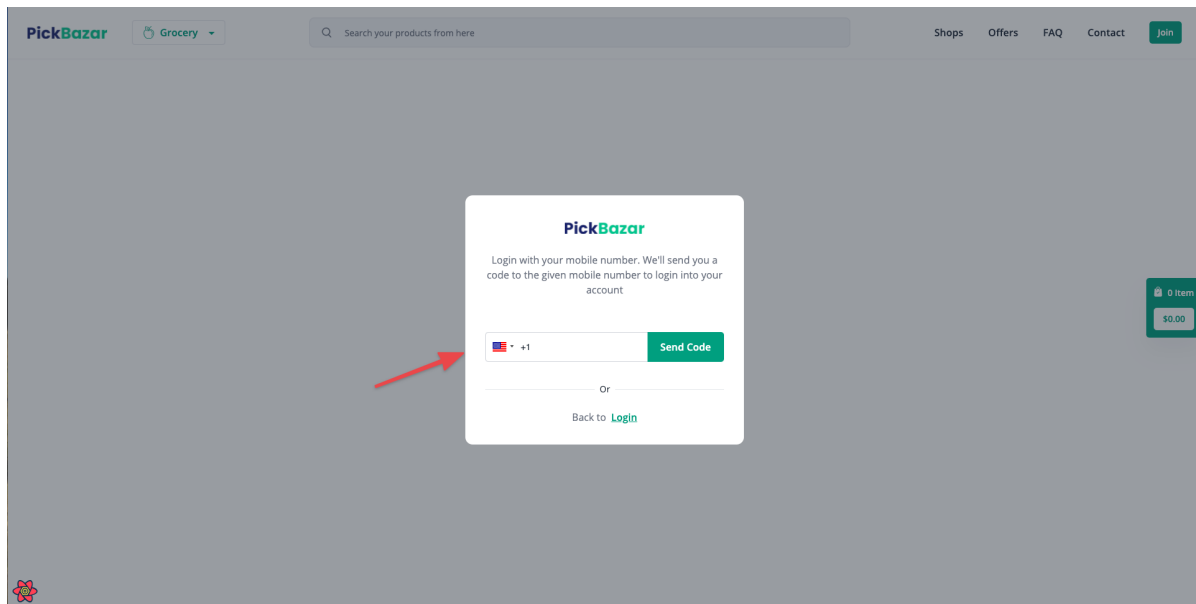
With that, `MessageBird` is appropriately configured for the OTP Service.

## FrontEnd Demo

### Login with Mobile Number:

After configuration, you will be able to use `mobile number` as a login provider. When users select the `Join` button, they'll get an option `Login with mobile` to use a mobile number as a login provider.

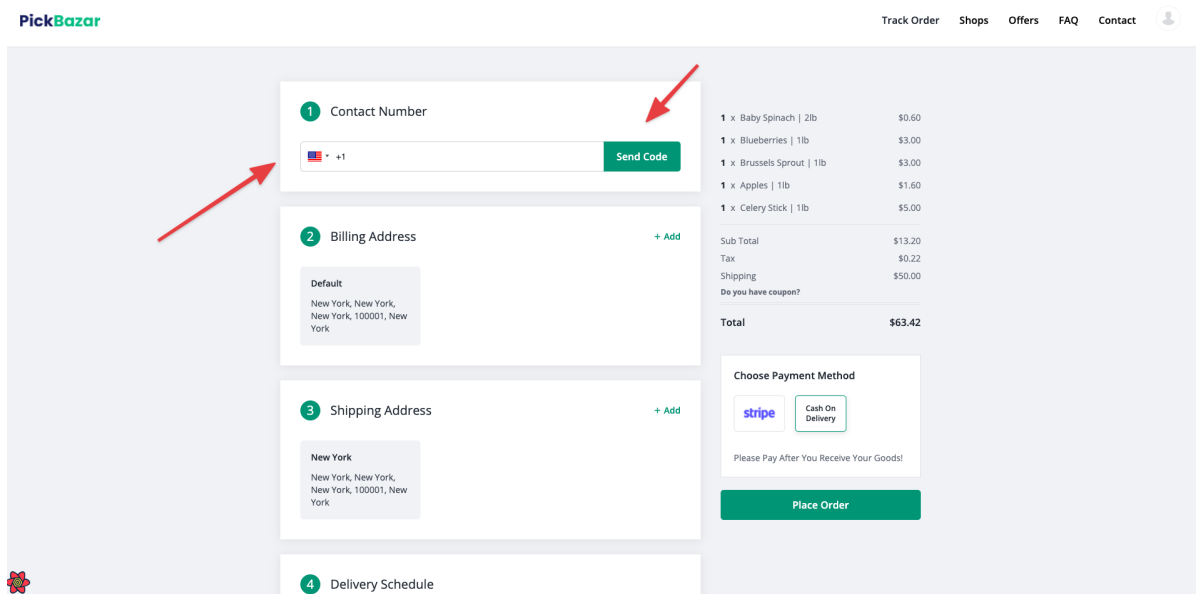




## OTP Verification During Checkout Process:

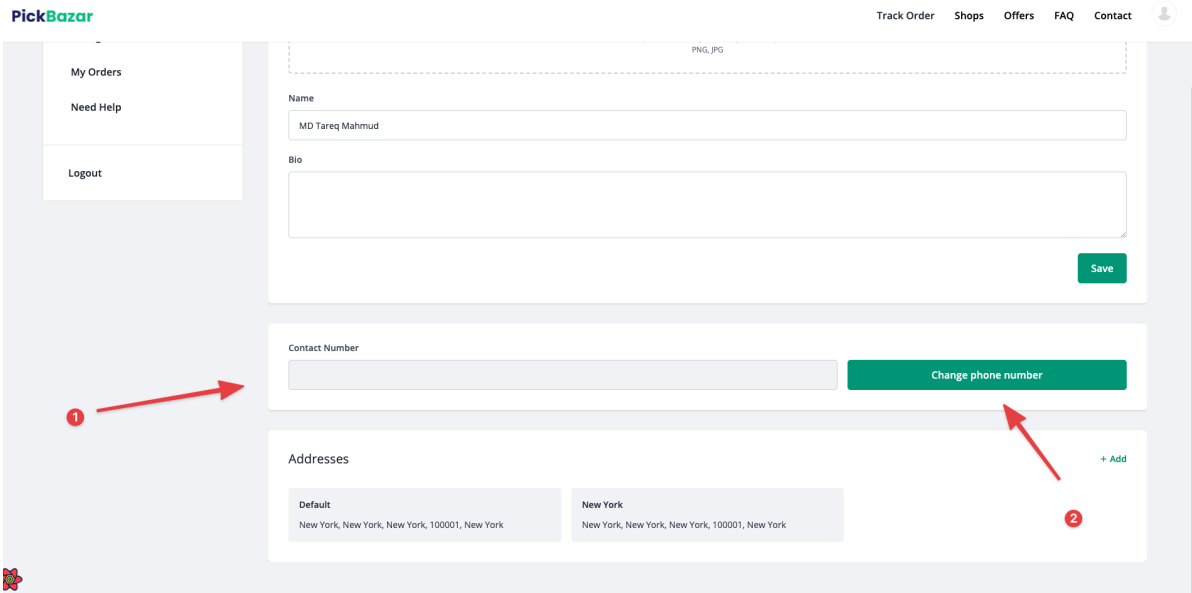
Similarly, when users go to the checkout page, they'll get a form to verify their mobile number.

For the front end, you don't have to do anything. When a user goes to the `checkout` page, they get a phone number box to add their number and verify it.



## OTP Verification For Updating Mobile Number:

Users also can change the number from their profile.



## Email Verification (Feature)

We implement the `Email Verification` feature on PickBazar `v8.2.0`. So if you want to use `Email Verification` with `PickBazar`, then make sure your PickBazar is `v8.2.0` or later.

With the Email Verification feature, you can do,

- Get Email After Registration
- Email verification for updating Email in profile update section
- When Email verification is on in settings then users will be forced to verify their email.

## How to turn on this feature

To turn on this feature go to settings of your site then find `Enable Must Verify Email` switch.

Currency

US Dollar

Minimum Order Amount

0

Wallet Currency Ratio

3

Sign Up Points

100

Maximum Question Limit

2

☐ Use OTP at checkout

☐ Enable Must Verify Email

Tax Class

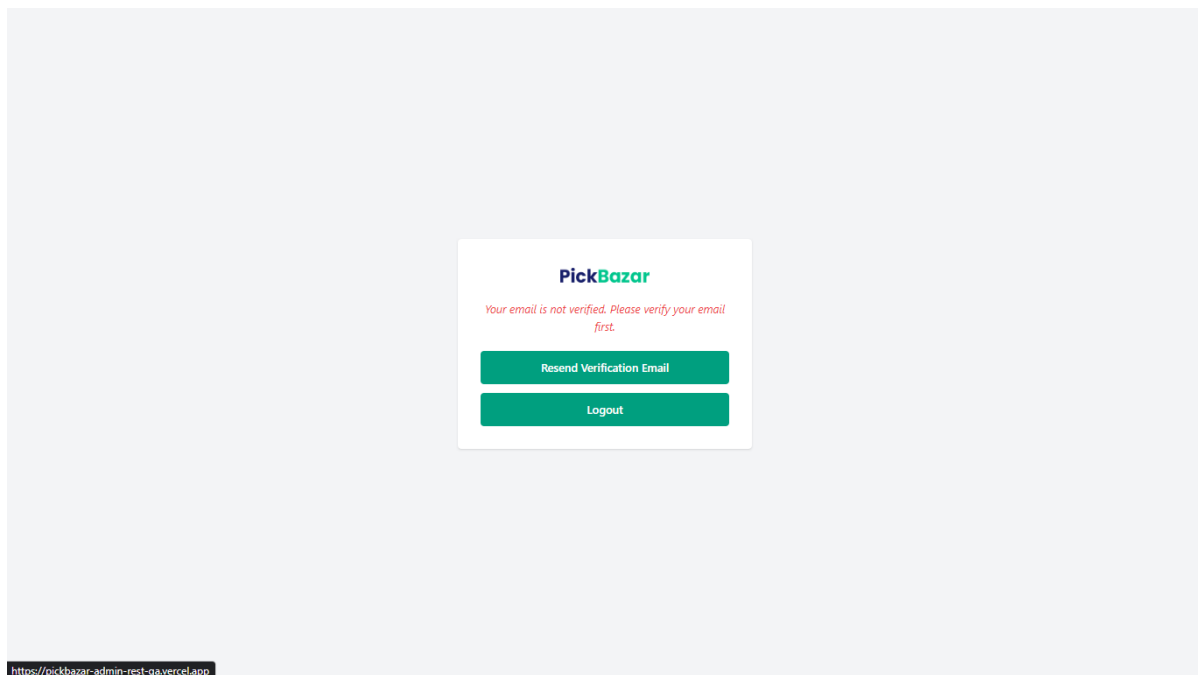
Billing

Shipping Class

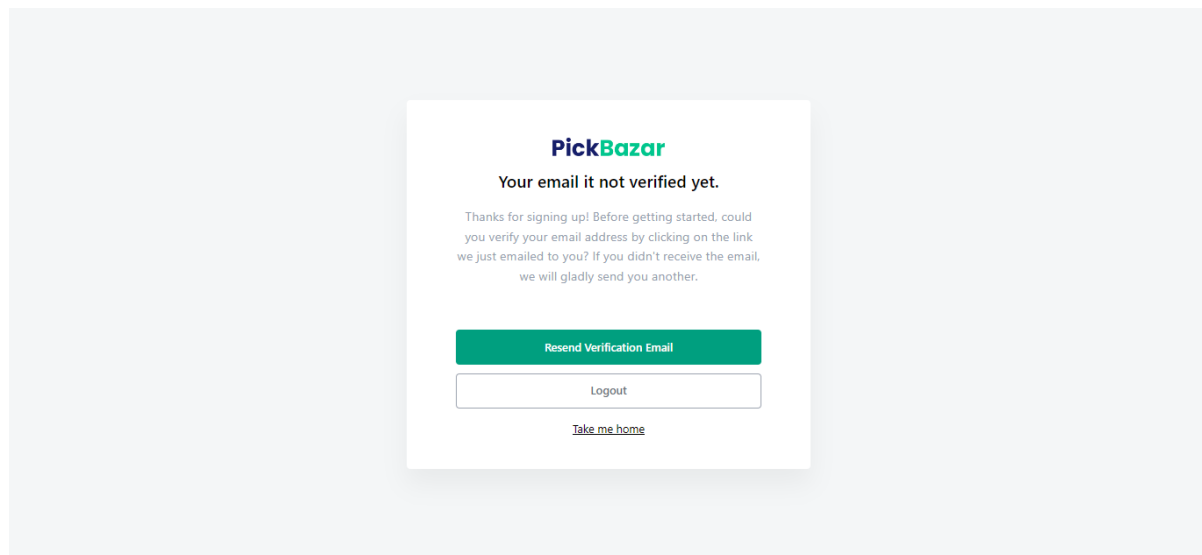
new

☐ Enable Free Shipping

## Admin view of this feature



## Shop view of this feature



## Configuration

To use this feature all you need to configure mail in your application.

```
MAIL_MAILER=mailgun
MAILGUN_DOMAIN=
MAILGUN_SECRET=
MAIL_FROM_ADDRESS=
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
ADMIN_EMAIL=support@example.com
```

## Translation (Feature)

If you're using `pickbazar-laravel` after `v2.0.0` then you can translate static content for multilanguage. It doesn't matter if the language is RTL or not, you can do both. For translation we use this [next-i18next](#) package.

## Existing Language

Right now, we provide translation files for `ar`, `de`, `en`, `es`, `he`, and `zh`.

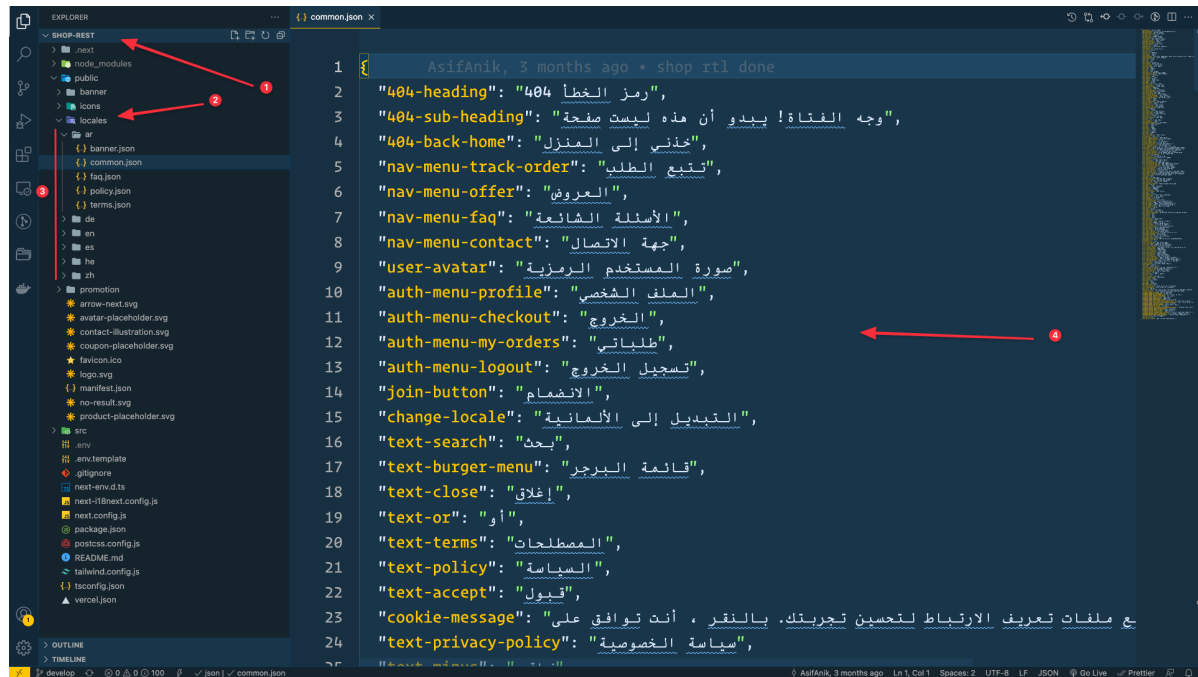
If your translation language is from them then just edit the translation string from,

### For Rest

```
pickbazar-laravel -> shop -> public -> locales -> YOUR LANGUAGE
pickbazar-laravel -> admin -> rest -> public -> locales -> YOUR LANGUAGE
```

## For GraphQL

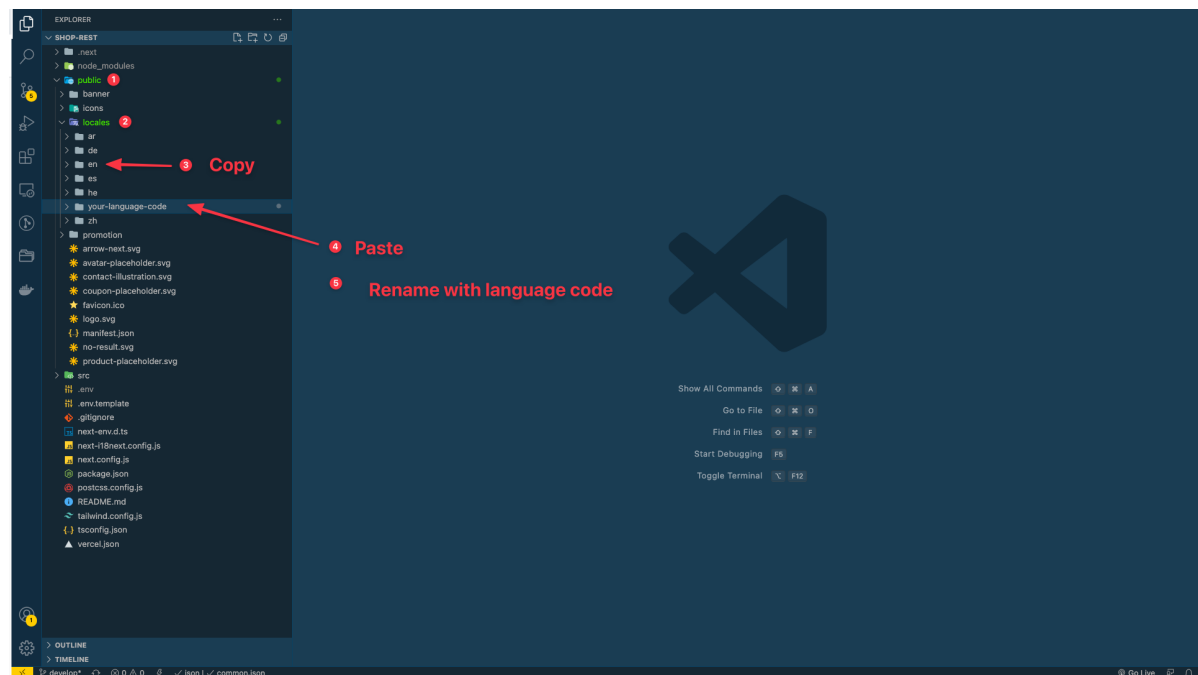
```
pickbazar-laravel -> shop -> public -> locales -> YOUR LANGUAGE  
pickbazar-laravel -> admin -> graphql -> public -> locales -> YOUR LANGUAGE
```



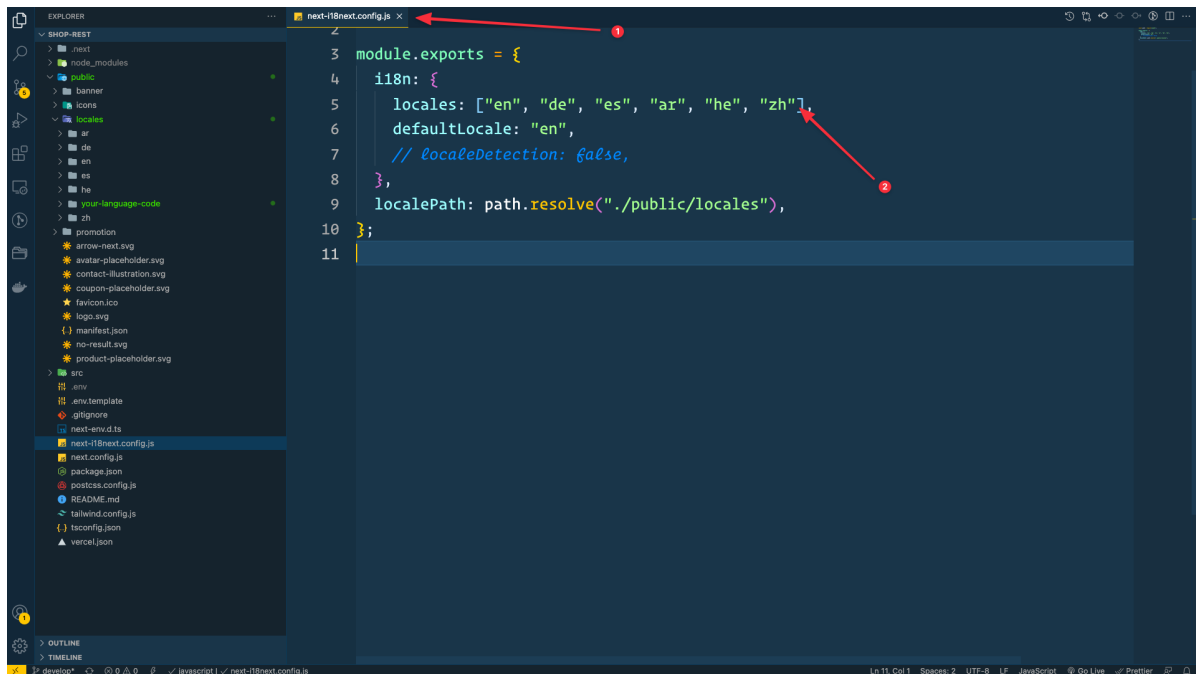
After translating `string` from `JSON` re-run or rebuild `project` it'll translate the static string.

## New Language

If your language is not in the lists then just copy the `en` folder from `locale` and paste it in the same folder by rename your language code.



After creating that folder edit `next-i18next.config.js` and add the `language-code` at `locales` object.



## Default Language

By default, English is the default language. To change that edit,

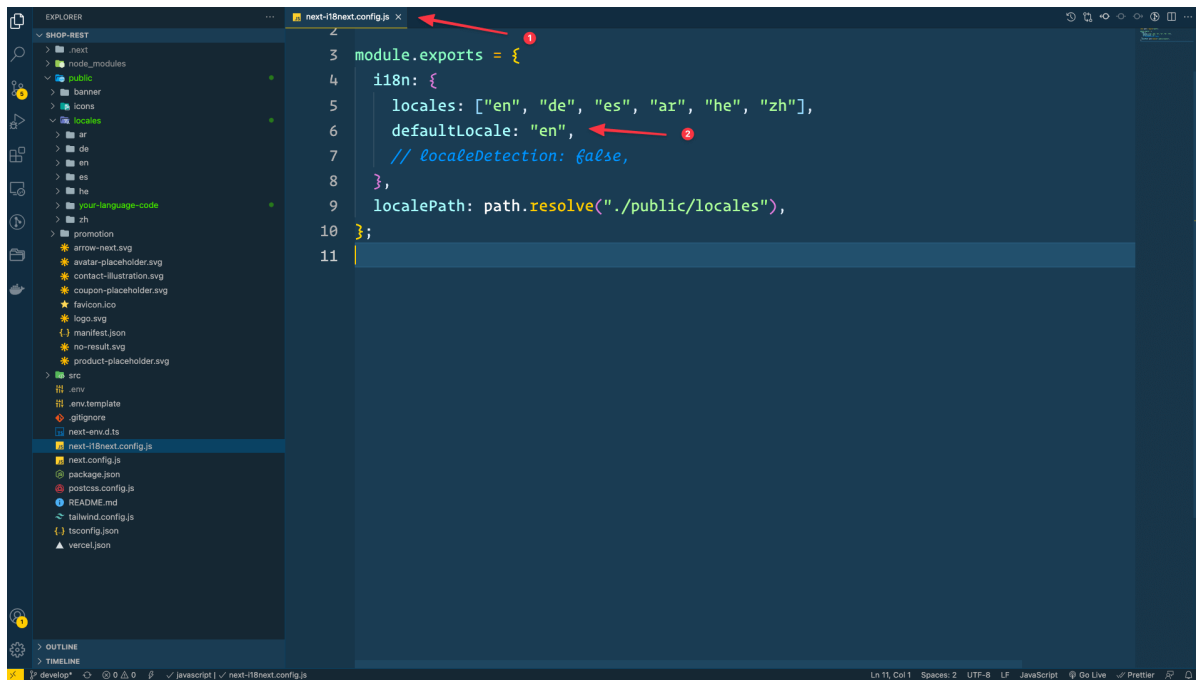
### For Rest

```
pickbazar-laravel -> shop -> public -> locales -> YOUR LANGUAGE -> next-i18next.config.js
pickbazar-laravel -> admin -> rest -> public -> locales -> YOUR LANGUAGE -> next-i18next.config.js
```

### For GraphQL

```
pickbazar-laravel -> shop -> public -> locales -> YOUR LANGUAGE -> next-i18next.config.js
pickbazar-laravel -> admin -> graphql -> public -> locales -> YOUR LANGUAGE -> next-i18next.config.js
```

And change `defaultLocale` to your language code.



## Multilingual (Feature)

If your business or site doesn't require multiple languages, then we don't recommend enabling the multilanguage feature. It'll increase the complexity of maintaining all the language simultaneously. Only enable it if you know what multilanguage is and your business or site needs that feature.

### Step 1: Enable Multilingual Feature

#### API:

At first open `api -> .env` and make true of ``

```
TRANSLATION_ENABLED=true
```

```
1 APP_NAME=Marvel
2 APP_ENV=local
3 APP_KEY=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6 APP_SERVICE=marvel.test
7 APP_NOTICE_DOMAIN=MARVEL_
8 DUMMY_DATA_PATH=pickbazar
9
10
11 # Multilang
12 # If you want to enable multilang then follow this doc → https://pickbazar-doc.vercel.app/multilang
13 TRANSLATION_ENABLED=true
14 DEFAULT_LANGUAGE=en
15
16
17
18 LOG_CHANNEL=stack
19 LOG_LEVEL=debug
20
21 DB_CONNECTION=mysql
22 DB_HOST=mysql
23 DB_PORT=3306
24 DB_DATABASE=marvel_laravel
25 DB_USERNAME=marvel_laravel
26 DB_PASSWORD=1Amarvel_laravel$
27
28 BROADCAST_DRIVER=log
29 CACHE_DRIVER=file
30 QUEUE_CONNECTION=sync
31 SESSION_DRIVER=file
32 SESSION_LIFETIME=120
33
34 MEMCACHED_HOST=memcached
35
36 REDIS_HOST=redis
37 REDIS_PASSWORD=null
38 REDIS_PORT=6379
```

## Admin

For admin,

update `admin -> rest -> .env` and update,

```
NEXT_PUBLIC_ENABLE_MULTI_LANG=true
```

And add available language to `NEXT_PUBLIC_AVAILABLE_LANGUAGES` with a comma separator.

For example, at your site, you want to support three languages, one is English, and the rest of the two will be german and Arabic. Then `NEXT_PUBLIC_AVAILABLE_LANGUAGES` will be like this,

```
NEXT_PUBLIC_AVAILABLE_LANGUAGES=en,de,ar
```

```
...
1 # Don't add '/' after the URL
2 NEXT_PUBLIC_REST_API_ENDPOINT="http://localhost"
3 NEXT_PUBLIC_SHOP_URL="http://localhost:3003"
4
5 # Application Mode and Authentication key
6 APPLICATION_MODE=development
7 NEXT_PUBLIC_AUTH_TOKEN_KEY=AUTH_CRED
8
9 # Default Language
10 NEXT_PUBLIC_DEFAULT_LANGUAGE=en
11 |
12 # Multilang
13 # If you want to enable multilang then follow this doc → https://pickbazar-doc.vercel.app/multilang
14 NEXT_PUBLIC_ENABLE_MULTI_LANG=true
15 NEXT_PUBLIC_AVAILABLE_LANGUAGES=en,de,ar
16
17 # API Key for third party service
18 NEXT_PUBLIC_GOOGLE_MAP_API_KEY=
```

For graphql `.env` will be at `admin -> graphql -> .env`

# Shop

For shop, similarly,

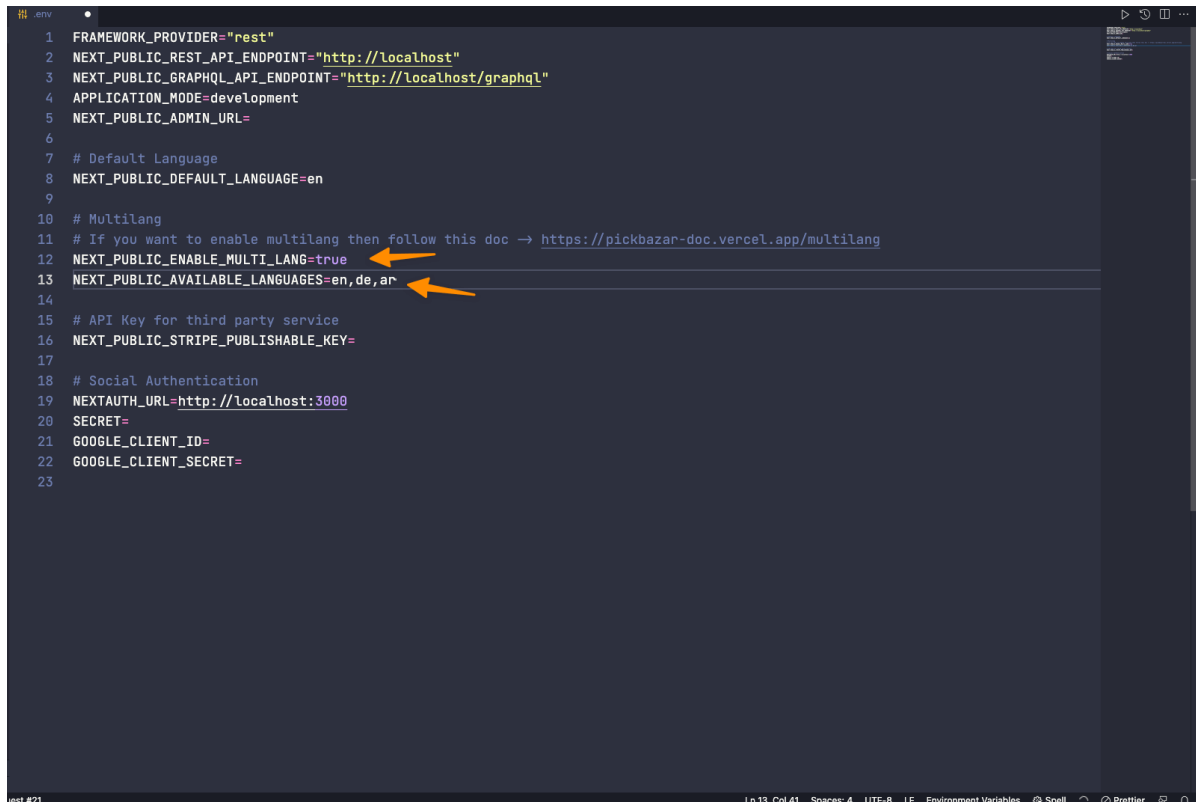
update `shop -> .env` and update,

```
NEXT_PUBLIC_ENABLE_MULTI_LANG=true
```

And add available language to `NEXT_PUBLIC_AVAILABLE_LANGUAGES` with a comma separator.

For example, at your site, you want to support three languages, one is English, and the rest of the two will be german and Arabic. Then `NEXT_PUBLIC_AVAILABLE_LANGUAGES` will be like this,

```
NEXT_PUBLIC_AVAILABLE_LANGUAGES=en,de,ar
```

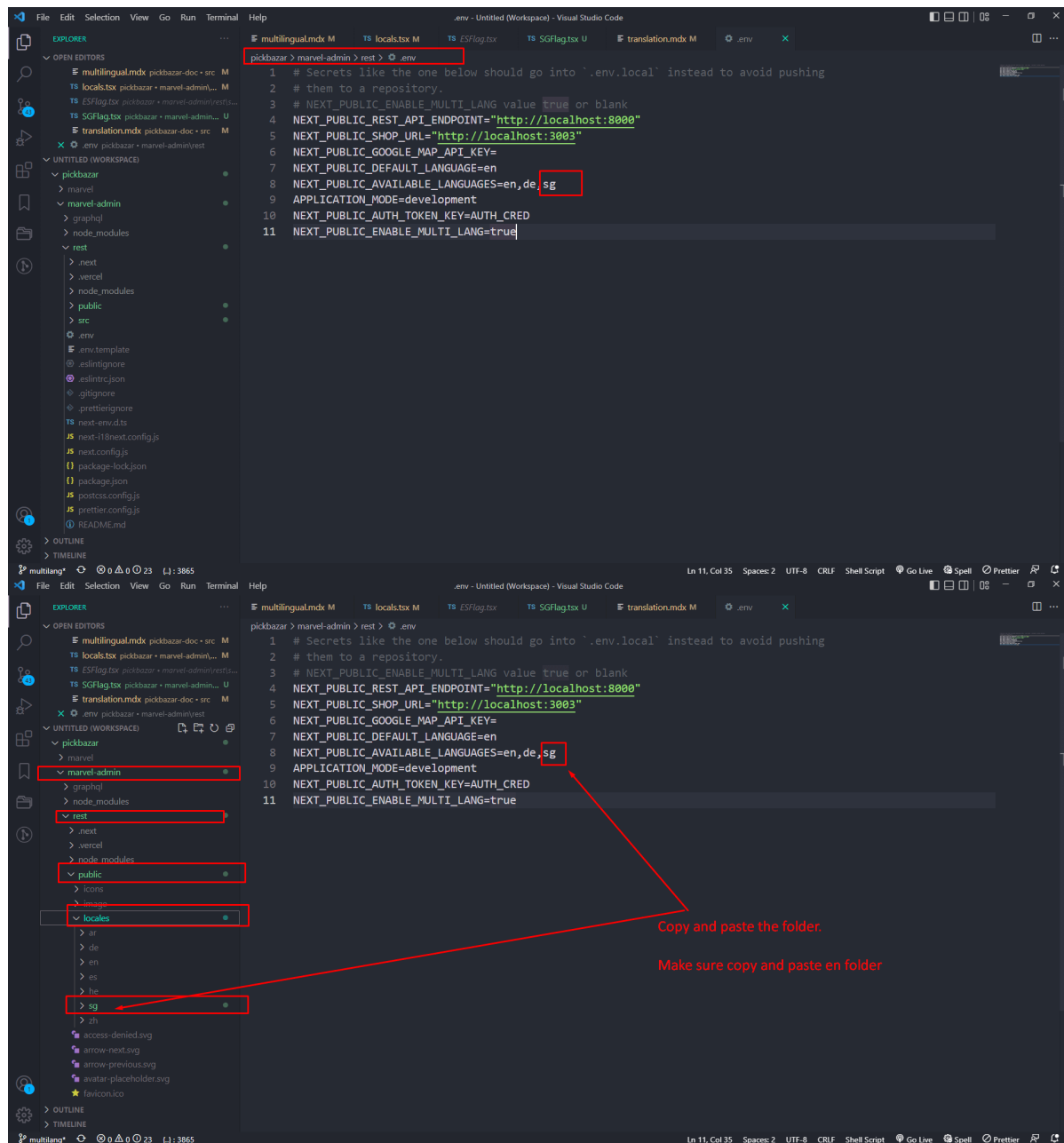


```
1  FRAMEWORK_PROVIDER="rest"
2  NEXT_PUBLIC_REST_API_ENDPOINT="http://localhost"
3  NEXT_PUBLIC_GRAPHQL_API_ENDPOINT="http://localhost/graphql"
4  APPLICATION_MODE=development
5  NEXT_PUBLIC_ADMIN_URL=
6
7  # Default Language
8  NEXT_PUBLIC_DEFAULT_LANGUAGE=en
9
10 # Multilang
11 # If you want to enable multilang then follow this doc → https://pickbazar-doc.vercel.app/multilang
12 NEXT_PUBLIC_ENABLE_MULTI_LANG=true
13 NEXT_PUBLIC_AVAILABLE_LANGUAGES=en,de,ar
14
15 # API Key for third party service
16 NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=
17
18 # Social Authentication
19 NEXTAUTH_URL=http://localhost:3000
20 SECRET=
21 GOOGLE_CLIENT_ID=
22 GOOGLE_CLIENT_SECRET=
23
```

After that, if you already host your site on server then make sure you redeploy the API and rebuild the frontend,

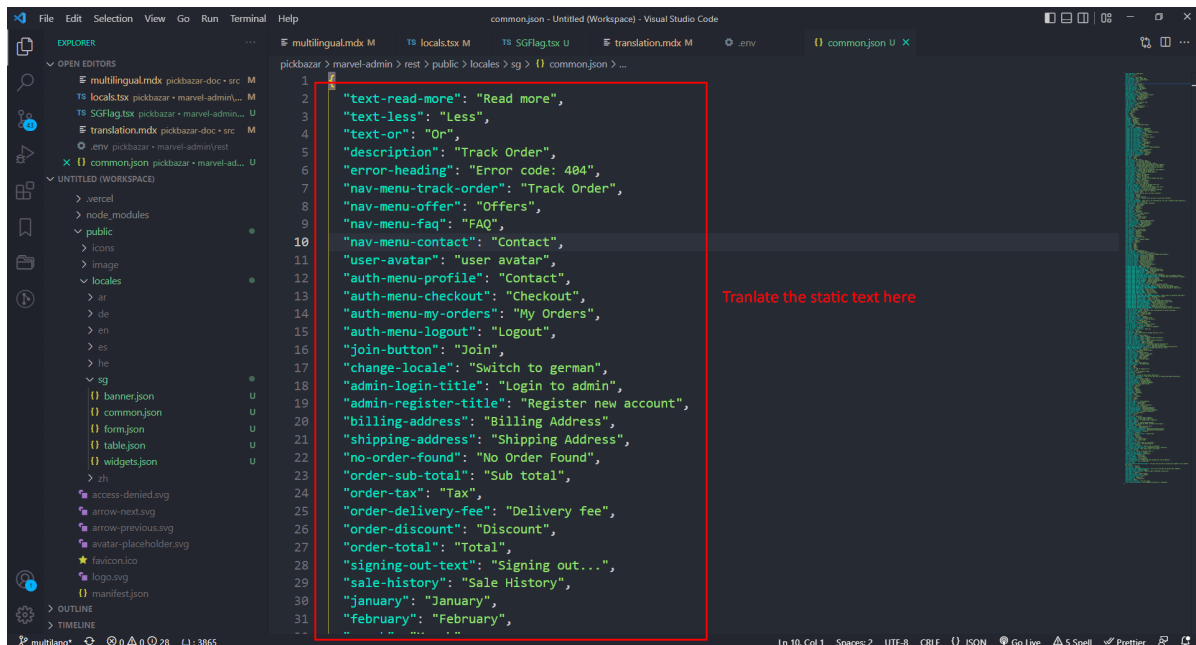
<https://pickbazar-doc.vercel.app/faq#how-to-rebuild-the-project>

## Step 2: How to add new language in admin?



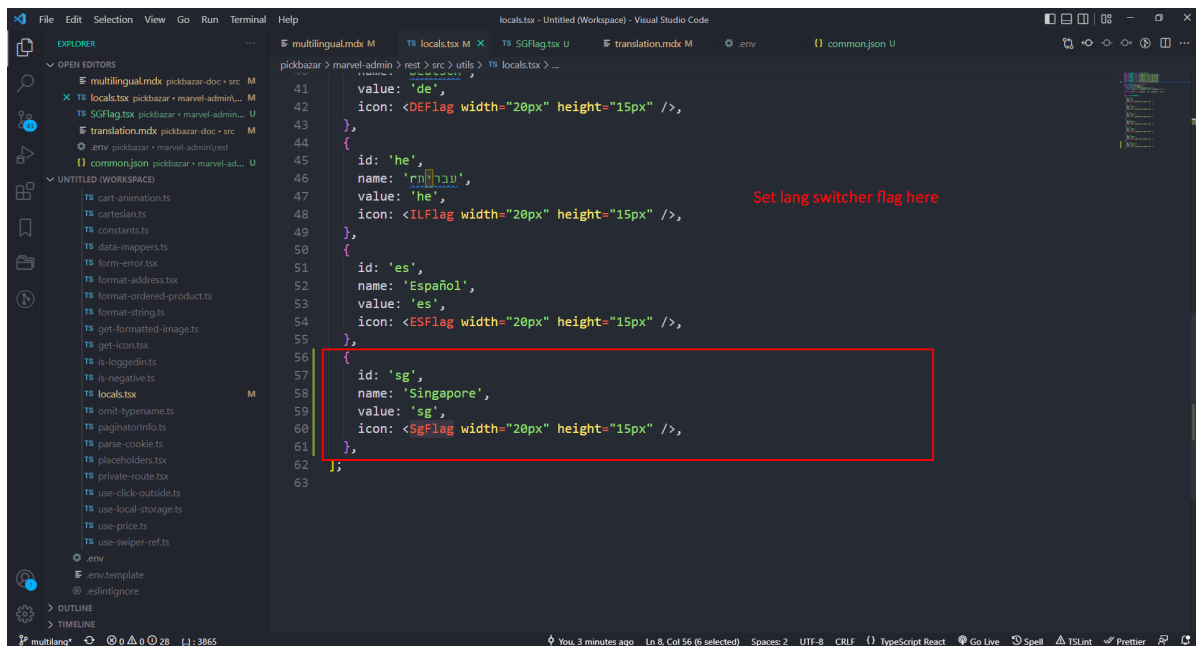
### File Location

admin\rest\public\locales\sg\banner.json



## File Location

admin\rest\src\utils\locals.tsx



## File Location

admin\rest\src\components\icons\flags\

```
export const SgFlag = ({ width = '640px', height = '480px' }) => {
  return (
    <svg
      xmlns="http://www.w3.org/2000/svg"
      id="flag-icon-css-es"
      viewBox="0 0 640 480"
      width={width}
      height={height}
    >
      <path fill="#AA151B" d="M0 0h640v480H0z" />
      <path fill="#F1BF00" d="M0 120h640v240H0z" />
      <path
        fill="#ad1519"
        d="M127.3 213.31-.8-.1-1-1-1-.7-.4-.6-.8-.7-1.1-.4-2c.3-.9-9-1.2 1.4-1.5a12 12 0 0 11.5-.511-.4 1.3-.3-5-.3c.2 0 .7 0 1-.2 1.1-.2 1.6 1h4.8c.4 0 1.2 3 1.4 4a35 35 0 0 2 .7c.5 1 1.6 3 2.2 6.5 3.9 7 1.1 11.5 1v1.1-.5-8-.6 1-.8-6-.5-5-1 .4c-.4 0-.4-.8-.7-6-.8-7.3-9-7.3-9"
      />
      <path
        fill="none"
        stroke="#000"
        strokeLinejoin="round"
        strokeWidth=".3"
        d="M127.3 213.31-.8-.1-1-1-1-.7-.4-.6-.8-.7-1.1-.4-2c.3-.9-9-1.2 1.4-1.5a12 12 0 0 11.5-.511-.4 1.3-.3-5-.3c.2 0 .7 0 1-.2 1.1-.2 1.6 1h4.8c.4 0 1.2 3 1.4 4a35 35 0 0 2 .7c.5 1 1.6 3 2.2 6.5 3.9 7 1.1 11.5 1v1.1-.5-8-.6 1-.8-6-.5-5-1 .4c-.4 0-.4-.8-.7-6-.8-7.3-9-7.3-9"
      />
      <path
        fill="#c8b100"
        d="M133.3 207c0-1.3-6-2.3 1.3-2.3 8 0 1.4 1 1.4 2.4 0 1.3-.6 2.4-1.4 2.4s-1.3-1.1-1.3-2.5"
      />
    </path>
  )
}
```

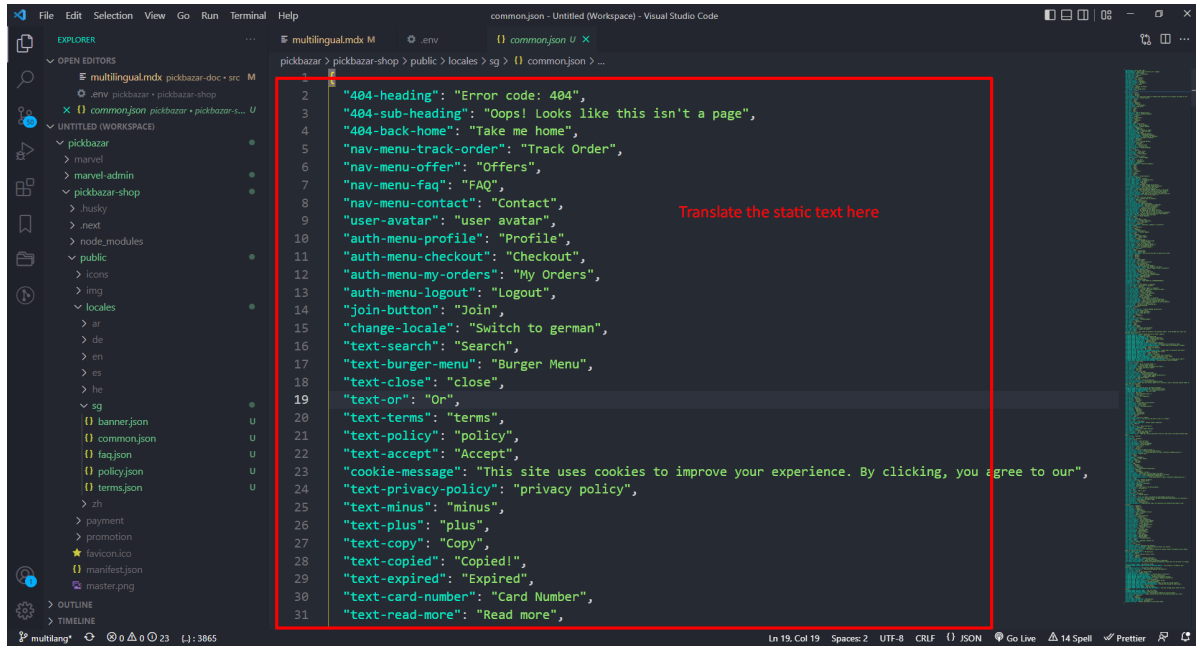
## Step 3: How to add new language in shop?

Copy and paste folder and rename it

Make sure you copy and paste en folder

File Location

shop\public\locales\sg\banner.json

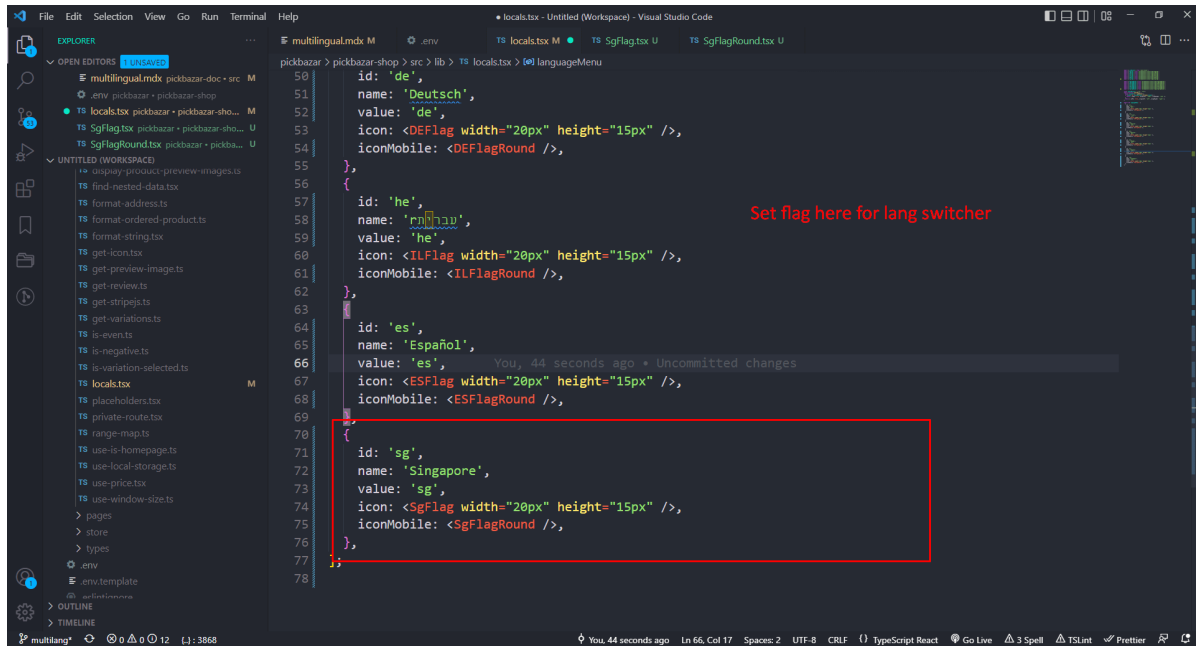


A screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows a project structure with folders like 'pickbazar', 'public', and 'locales'. The 'locales' folder is expanded, showing 'sg' and 'banner.json'. The main editor window displays the 'common.json' file, which is a JSON object containing various static text strings. A red rectangular box highlights the entire JSON content. A red text annotation 'Translate the static text here' is placed to the right of the highlighted area. The status bar at the bottom indicates 'Ln 19, Col 19'.

```
1 {
2   "404-heading": "Error code: 404",
3   "404-sub-heading": "Oops! Looks like this isn't a page",
4   "404-back-home": "Take me home",
5   "nav-menu-track-order": "Track Order",
6   "nav-menu-offer": "Offers",
7   "nav-menu-faq": "FAQ",
8   "nav-menu-contact": "Contact",
9   "user-avatar": "user avatar",
10  "auth-menu-profile": "Profile",
11  "auth-menu-checkout": "Checkout",
12  "auth-menu-my-orders": "My Orders",
13  "auth-menu-logout": "Logout",
14  "join-button": "Join",
15  "change-locale": "Switch to german",
16  "text-search": "Search",
17  "text-burger-menu": "Burger Menu",
18  "text-close": "close",
19  "text-or": "Or",
20  "text-terms": "terms",
21  "text-policy": "policy",
22  "text-accept": "Accept",
23  "cookie-message": "This site uses cookies to improve your experience. By clicking, you agree to our",
24  "text-privacy-policy": "privacy policy",
25  "text-minus": "minus",
26  "text-plus": "plus",
27  "text-copy": "Copy",
28  "text-copied": "Copied!",
29  "text-expired": "Expired",
30  "text-card-number": "Card Number",
31  "text-read-more": "Read more",
}
```

## File Location

shop\src\lib\locals.tsx

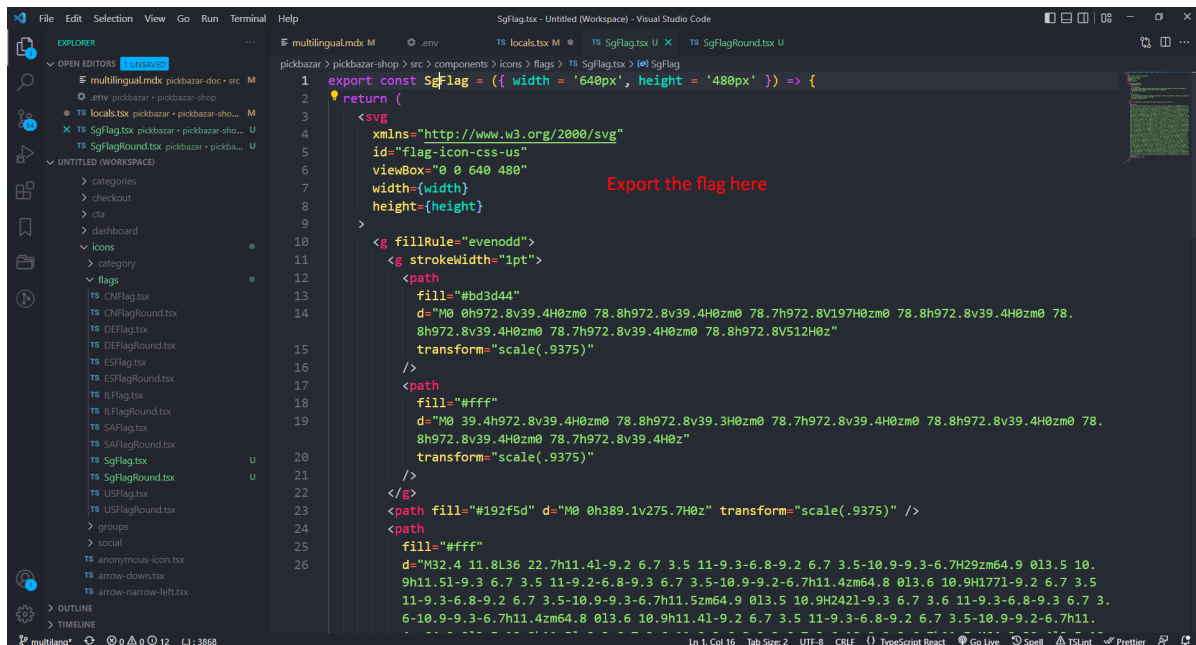


A screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows a project structure with folders like 'pickbazar', 'src', and 'lib'. The 'lib' folder is expanded, showing 'locals.tsx'. The main editor window displays the 'locals.tsx' file, which contains a TypeScript array of objects representing different languages. A red rectangular box highlights the 'sg' (Singapore) object. A red text annotation 'Set flag here for lang switcher' is placed to the right of the highlighted area. The status bar at the bottom indicates 'Ln 66, Col 17'.

```
50 {
51   id: 'de',
52   name: 'Deutsch',
53   value: 'de',
54   icon: <DEFlag width="20px" height="15px" />,
55   iconMobile: <DEFlagRound />,
56 },
57 {
58   id: 'he',
59   name: 'עברית',
60   value: 'he',
61   icon: <ILFlag width="20px" height="15px" />,
62   iconMobile: <ILFlagRound />,
63 },
64 {
65   id: 'es',
66   name: 'Español',
67   value: 'es',
68   icon: <ESFlag width="20px" height="15px" />,
69   iconMobile: <ESFlagRound />,
70 },
71 {
72   id: 'sg',
73   name: 'Singapore',
74   value: 'sg',
75   icon: <SgFlag width="20px" height="15px" />,
76   iconMobile: <SgFlagRound />,
77 },
78 }
```

## File Location

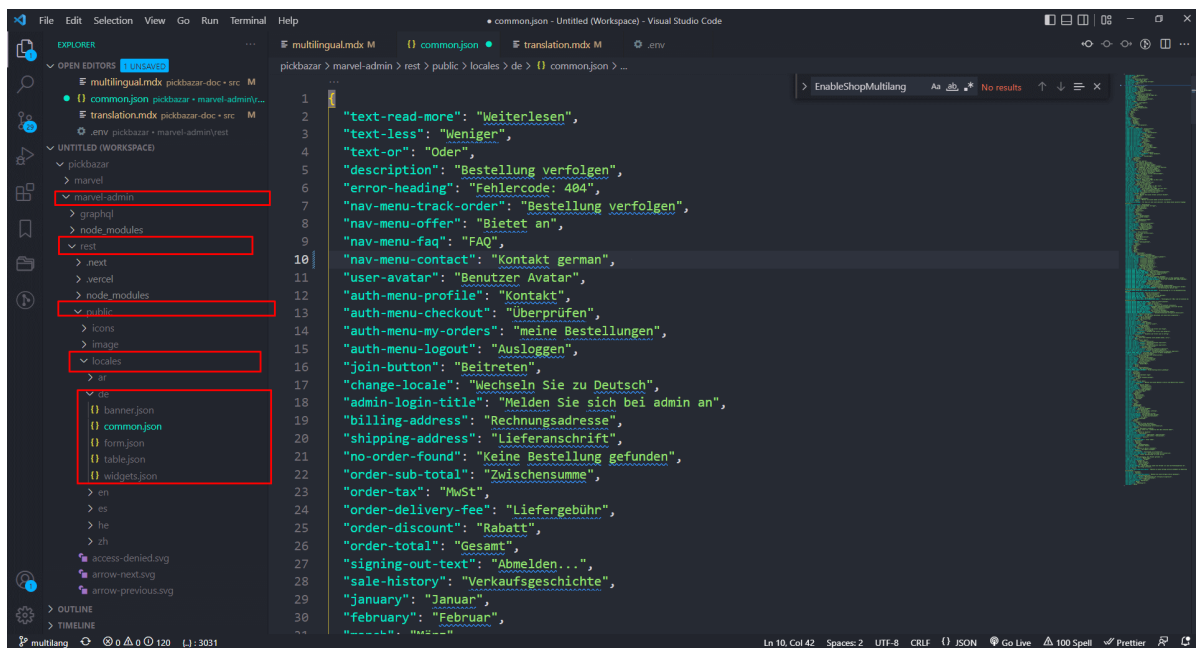
shop\src\components\icons\flags\



## Step 3: How to translate static content for admin?

File Location

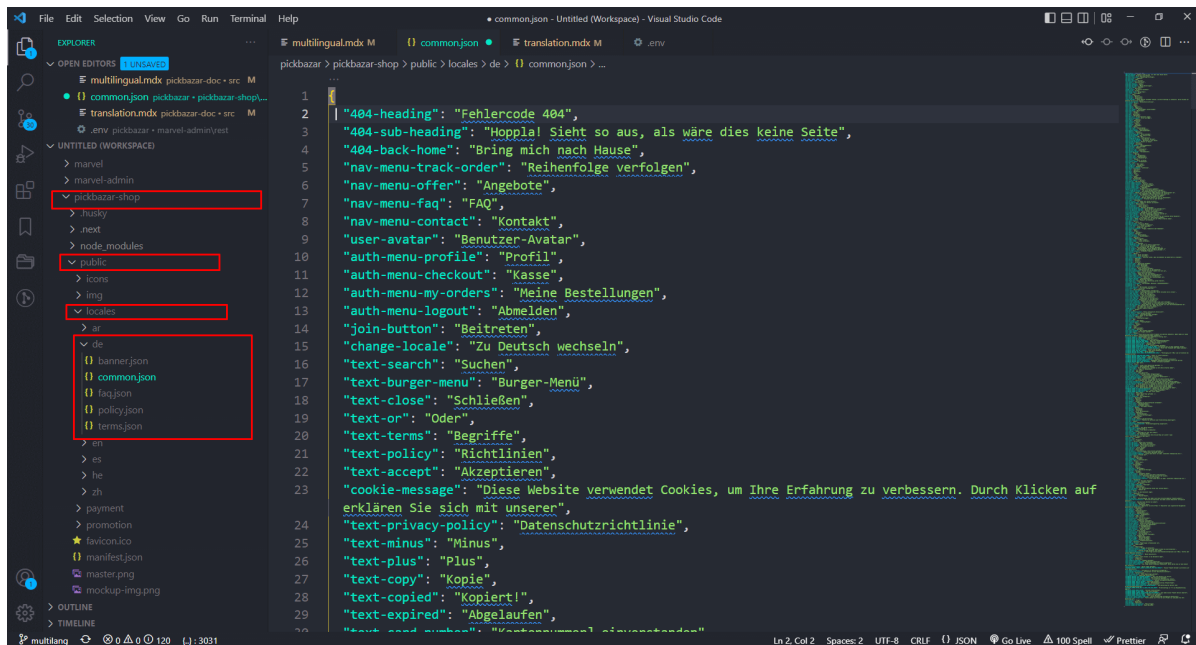
admin\rest\public\locales\de\common.json



## Step 4: How to translate static content for shop?

File Location

shop\rest\public\locales\de\common.json



## Step 5: How to translate content?

Now after successfully configuring the Multilingual follow this procedure to translate your existing content to another one.

### Data Type

Now before starting the translation, you've to understand about dependent and non-dependent data at pickbazar.

At pickbazar, some data are required to create another data, and it's called dependent data, and the independent data is called non-dependent data.

For example,

To create product types, categories, tags, author, manufacture, attributes, etc are required.

And for coupons or settings, no prior data is required.

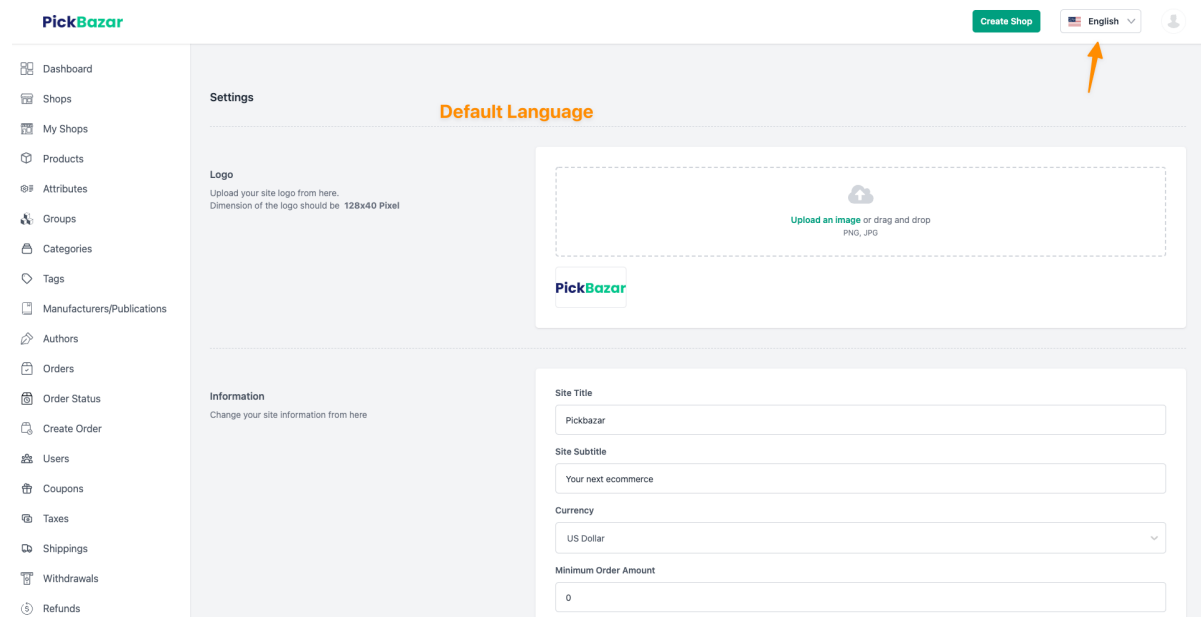
Now, to translate a product, you must have to translate types, categories, tags, authors, manufacturers, and attributes at first. Without that, when you translate a product, you can't add your categories or types as those data are not translated yet.

To simplify the process, we made a serial for you. Just follow this serial, and it'll make the process relatively easy.

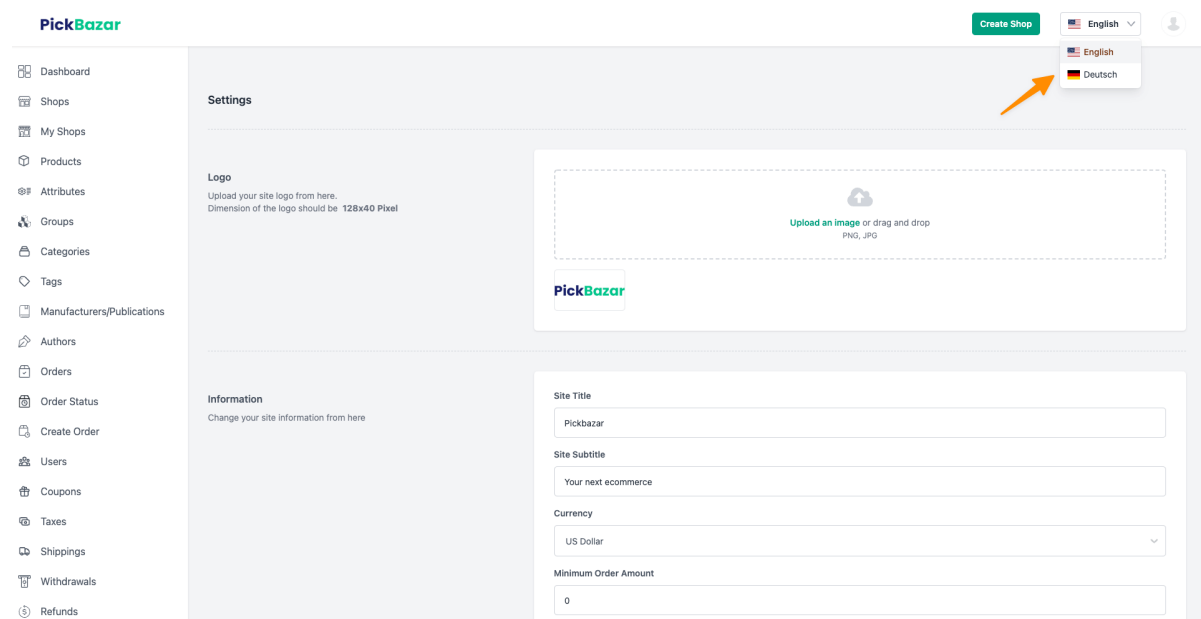
1. Settings
2. Order Status
3. Group
4. Author
5. Manufacturer
6. Tags
7. Categories
8. Attributes
9. Products

# Translate Settings

If you want to translate the settings,  
just go to your admin -> settings  
It'll open default language settings.



Then change the language from the navbar, and the settings page will be redirected,



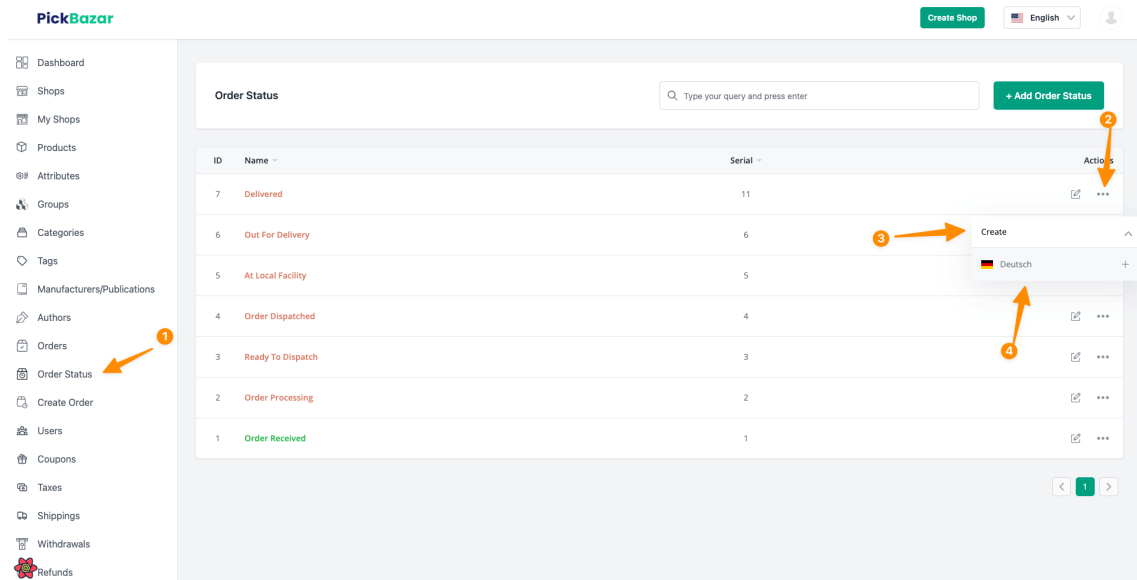
Then update the text of the settings page and click save.

# Translate Order Status

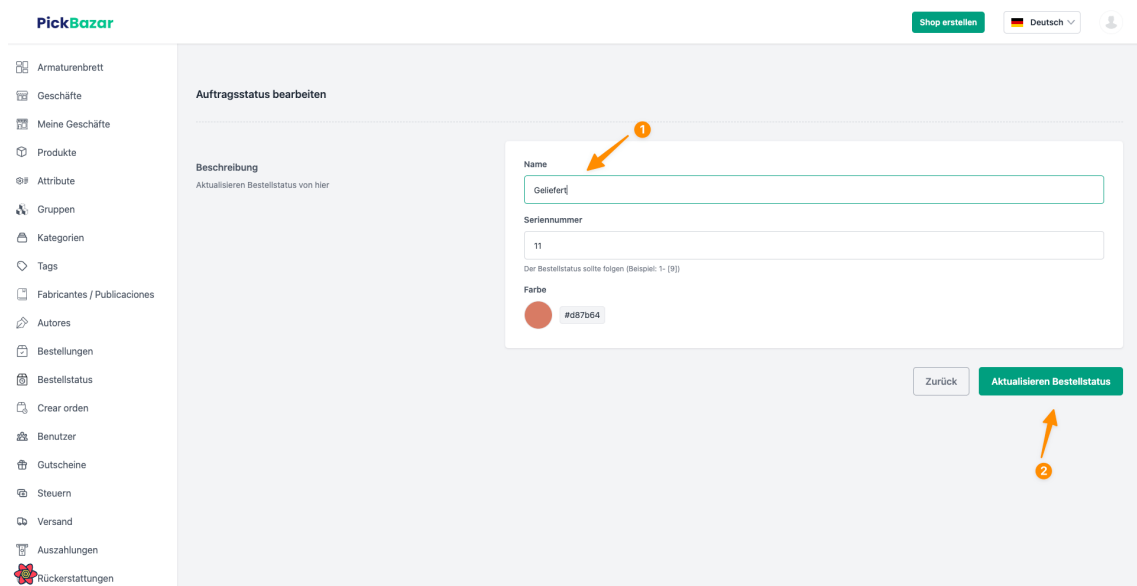
To translate the order status,

1. Go to the order status page from the admin

2. Then click **Three Dot** and Create translate,



3. Then translate the text and click Save.



Similarly, translate all the order status.

## Translate Group, Author, Manufacturer, Tags, Categories:

Follow order status procedure to translate **Group**, **Author**, **Manufacturer**, **Tags**, and **Categories**.

## Translate Products

### Translate Simple Products:

Similarly, just edit three dots and translate the product.

But during the translation, don't change the **SKU**. SKU for a product has to be similar for all languages. We calculate the quantity based on that field.

PickBazar

Shop erstellen

Deutsch

Armaturenbrett

Geschäfte

Meine Geschäfte

Produkte

Attribute

Gruppen

Kategorien

Tags

Fabricantes / Publicaciones

Autores

Bestellungen

Bestellstatus

Crear orden

Benutzer

Gutscheine

Steuern

Versand

Auszahlungen

Rückstellungen

Don't change SKU during the translation

Menge\*

500

Artikelnummer\*

hgkjhkhghfdfsfxvcbnghi+++g+5dfg+6

Stellen Sie sicher, dass die SKU für alle Sprachen identisch ist.

Breite

Höhe

Länge

☒ Ist digital

☐ Ist extern

Digitale Datei

Bild hochladen oder Drag & Drop

PNG, JPG

Zurück

Produkt aktualisieren

## Translate Variable Product:

For variable products, you must have to translate attributes first. Without attributes, you can't get an attribute list during the translation.

Click on **Three Dots** to translate the item.

But for the variable product, you must create the variation for translation again.

# Translate Attributes

PickBazar

Create Shop

English

Dashboard

Shops

My Shops

Products

Attributes

Groups

Categories

Tags

Manufacturers/Publications

Authors

Orders

Order Status

Create Order

Users

Coupons

Taxes

Shippings

Withdrawals

Attributes

ID	Name	Shop	Values	Actions
6	Book Type	Books Shop	Hardcover Book, Picture Book, Paperback Book, PDF Book	<div><div></div><div></div><div></div></div>
5	Language	Books Shop	English, Arabic, Spanish, French, Hindi, Hebrew	<div><div></div><div></div><div></div></div>
4	Size	Clothing Shop	S, M, L, XL	<div><div></div><div></div><div></div></div>
3	Color	Clothing Shop	Red, Blue, White	<div><div></div><div></div><div></div></div>

Create

Deutsch

PickBazar

Shop erstellen

Deutsch

Armaturen Brett

Geschäfte

Meine Geschäfte

Produkte

Attribute

Gruppen

Kategorien

Tags

Fabricantes / Publicaciones

Autores

Bestellungen

Bestellstatus

Crear orden

Benutzer

Gutscheine

Steuern

Versand

Auszahlungen

Attributwerte

Aktualisieren Ihr Attributwert und die erforderlichen Informationen von hier

Wert

Gebundenes Buch

Meta

Entfernen

Wert

Bilderbuch

Meta

Entfernen

Wert

Taschenbuch

Meta

Entfernen

Wert

PDF-Buch

Meta

Entfernen

Wert hinzufügen

Zurück

Aktualisieren Attribut

# Translate Groups

PickBazar

Create Shop

English

Dashboard

Shops

My Shops

Products

Attributes

Groups

Categories

Tags

Manufacturers/Publications

Authors

Orders

Order Status

Create Order

Users

Coupons

Taxes

Shippings

Groups

Groups

Type your query and press enter

+ Add Group

ID	Name	Icon	Actions
8	Books		<div><div></div><div></div><div></div></div>
7	Daily Needs		<div><div></div><div></div><div></div></div>
6	Furniture		<div><div></div><div></div><div></div></div>
5	Clothing		<div><div></div><div></div><div></div></div>
4	Bags		<div><div></div><div></div><div></div></div>
3	Makeup		<div><div></div><div></div><div></div></div>
2	Bakery		<div><div></div><div></div><div></div></div>
1	Grocery		<div><div></div><div></div><div></div></div>

Create

Deutsch

PickBazar

Shop erstellen

Deutsch

Armaturenbrett

Geschäfte

Meine Geschäfte

Produkte

Attribute

Gruppen

Kategorien

Tags

Fabricantes / Publicaciones

Autores

Bestellungen

Bestellstatus

Crear orden

Benutzer

Gutscheine

Steuern

Versand

Auszahlungen

Banner

Banner

Fügen Sie hier Ihr Bannerbild mit Titel und Beschreibung hinzu. Die Größe des Banners sollte 1920 x 1080 px für das Vollbild-Banner und 1500 x 450 px für das kleine Banner betragen

Banner 1

Entfernen

Titel

Buchbanner

Beschreibung

Dies ist das Buch-Demo-Banner

Galerie

Bild hochladen oder Drag & Drop  
PNG, JPG

Banner hinzufügen

Zurück

Gruppe aktualisieren

# Translate Variable Products

PickBazar

Create Shop

English

Dashboard

Shops

My Shops

Products

Attributes

Groups

Categories

Tags

Manufacturers/Publications

Authors

Orders

Order Status

Create Order

Users

Coupons

Taxes

Shippings

Products

Type your query and press enter

Filter

Image	Name	Group	Shop	Product Type	Price/Unit	Quantity	Status	Actions
	Blood Drinker	Books	Books Shop	variable	\$180.00 - \$200.00	1000	publish	<div><div></div><div></div><div></div></div>
	Forest Killer	Books	Books Shop	simple	\$250.00	500	publish	<div><div></div><div></div><div></div></div>
	Fleash Eater	Books	Books Shop	simple	\$250.00	500	publish	<div><div></div><div></div><div></div></div>
	The Boneyard Man	Books	Books Shop	variable	\$300.00 - \$300.00	1000	publish	<div><div></div><div></div><div></div></div>
	The Physco Killer First Part	Books	Books Shop	simple	\$200.00	500	publish	<div><div></div><div></div><div></div></div>
	The Serial Killer	Books	Books Shop	simple	\$200.00	500	publish	<div><div></div><div></div><div></div></div>
	Before The Ruins	Books	Books Shop	simple	\$200.00	500	publish	<div><div></div><div></div><div></div></div>
	Eight Purfect Murders	Books	Books Shop	simple	\$200.00	500	publish	<div><div></div><div></div><div></div></div>

Create

Deutsch

PickBazar

Shop erstellen

Deutsch

Armaturenbrett

Geschäfte

Meine Geschäfte

Produkte

Attribute

Gruppen

Kategorien

Tags

Fabricantes / Publicaciones

Autores

Bestellungen

Bestellstatus

Crear orden

Benutzer

Gutscheine

Steuern

Versand

Auszahlungen

Gruppe & Kategorien

Wählen Sie hier Produktgruppe und Kategorien aus

Gruppe\*

Bücher

Kategorien

Thriller

Autores

Kaity lerry

Fabricantes

Facry-Veröffentlichung

Tags

Combo

Beschreibung

Bearbeiten Ihre Produktbeschreibung und die notwendigen Informationen finden Sie hier

Name\*

Blut-Trinker

Einheit\*

PickBazar

Shop erstellen

Deutsch

Armaturenbrett

Geschäfte

Meine Geschäfte

Produkte

Attribute

Gruppen

Kategorien

Tags

Fabricantes / Publicaciones

Autores

Bestellungen

Bestellstatus

Crear orden

Benutzer

Gutscheine

Steuern

Versand

Auszahlungen

Einheit\*

1 pc

Beschreibung

Cerebral suspenser is a kidney combining the suspenser and cerebral fabrication stripes. It's generally used to describe literature or flicks that deal with cerebral narratives in a suspenser or thrilling setting.  
In terms of environment and convention, it's a subgenre of the broader ranging suspenser narrative structure, with parallels to Gothic and operative fabrication in the sense of occasionally having a "dissolving sense of reality". It's frequently told through the standpoint of psychologically stressed characters, revealing their distorted internal consciousness and fixation on the complex and frequently featured associations between consciousness and...

Status

Veröffentlicht

Entwurf

Produktart

Variable

Informationen zu Produktvariationen

Aktualisieren Ihre Produktvariation und die erforderlichen Informationen von hier

Option hinzufügen

OPTIONEN

Zurück

Produkt aktualisieren



# Step 6: How to add custom or solved broken font issue?

Invoice No: LpDNRSepec3MU

Date: 7th July, 2022

**Pobon**

pobon@redq.io  
19365141641631  
2148 Straford Park  
Winchester  
KY  
40391  
United States

**Pickbazar**

https://redq.io  
+129290122122  
NY State Thruway, New York, USA

Products	Quantity	Total
□□□□□ □□□□□ □□□□□□□ □□□□□	1	\$5.00

Sub total : \$5.00  
Discount : \$0.00  
Tax : \$0.10  
Delivery fee : \$50.00  
**Total : \$55.10**

```
File Location
api\config\pdf.php
'font_path' => base_path('resources/fonts/'),
'font_data' => [
    'example' => [
        'R' => 'example.ttf',    // regular font
        'B' => 'example.ttf',    // optional: bold font
        'I' => 'example.ttf',    // optional: italic font
        'BI' => 'example.ttf', // optional: bold-italic font
        'useOTL' => 0xFF,
        'useKashida' => 75,
    ]
],
```

```
1 <?php
2
3 return [
4     'mode' => 'utf-8',
5     'format' => 'A4',
6     'author' => '',
7     'subject' => '',
8     'keywords' => '',
9     'creator' => 'Laravel Pdf',
10    'display_mode' => 'fullpage',
11    'tempDir' => storage_path('app/temp'), // base_path(storage_path('app/public')),
12    'pdf_a' => false,
13    'pdf_a_auto' => false,
14    'icc_profile_path' => '',
15    'font_path' => base_path('resources/fonts/'),
16    'font_data' => [
17        'example' => [ // You, 2 seconds ago • Uncommitted changes
18            'R' => 'example.ttf', // regular font
19            'B' => 'example.ttf', // optional: bold font
20            'I' => 'example.ttf', // optional: italic font
21            'BI' => 'example.ttf', // optional: bold-italic font
22            'useOTL' => 0xFF,
23            'useKashida' => 75,
24        ],
25    ],
26 ];
27
```

## File Location

api\resources\fonts\

```
1 <?php
2
3 return [
4     'mode' => 'utf-8',
5     'format' => 'A4',
6     'author' => '',
7     'subject' => '',
8     'keywords' => '',
9     'creator' => 'Laravel Pdf',
10    'display_mode' => 'fullpage',
11    'tempDir' => storage_path('app/temp'), // base_path(storage_path('app/public')),
12    'pdf_a' => false,
13    'pdf_a_auto' => false,
14    'icc_profile_path' => '',
15    'font_path' => base_path('resources/fonts/'),
16    'font_data' => [
17        'example' => [ // You, 2 seconds ago • Uncommitted changes
18            'R' => 'example.ttf', // regular font
19            'B' => 'example.ttf', // optional: bold font
20            'I' => 'example.ttf', // optional: italic font
21            'BI' => 'example.ttf', // optional: bold-italic font
22            'useOTL' => 0xFF,
23            'useKashida' => 75,
24        ],
25    ],
26 ];
27
```

## File Location

api/packages/marvel/stubs/resources/views/pdf/order-invoice.blade.php

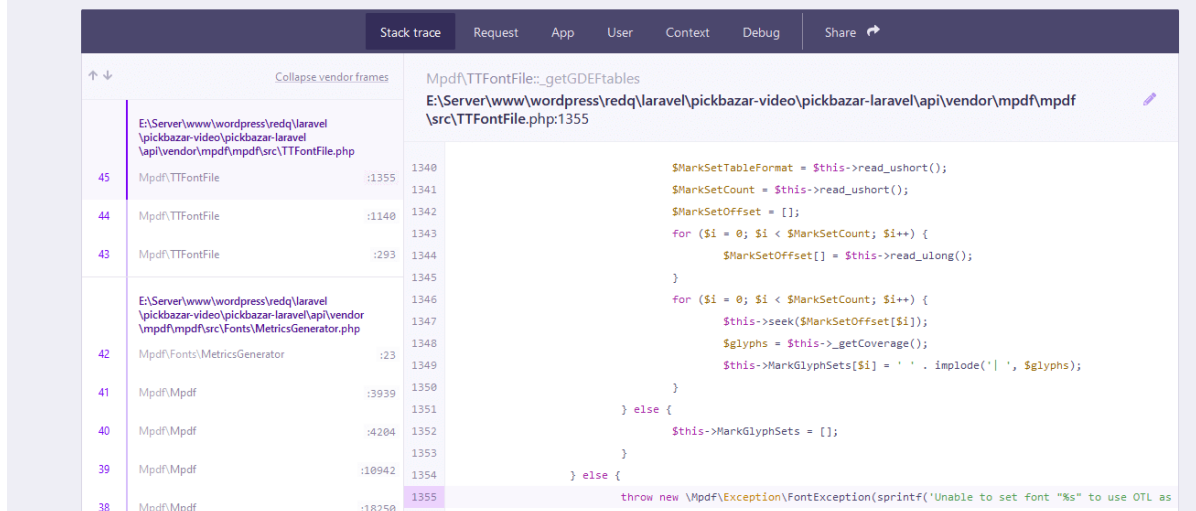
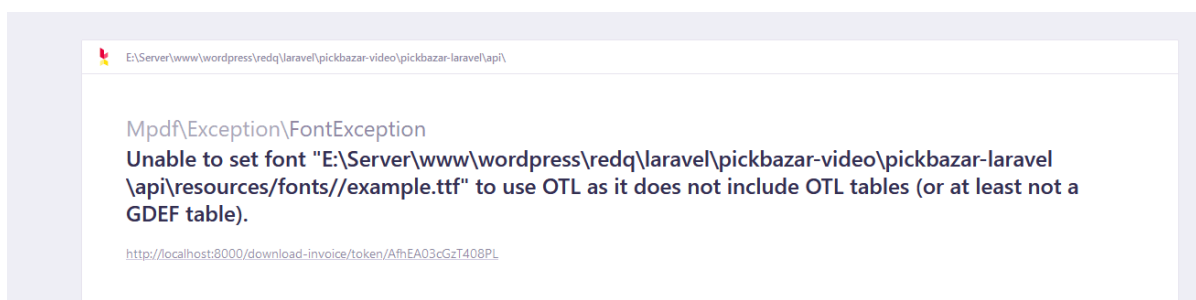
api/resources/views/pdf/order-invoice.blade.php

```
@if($language === "ex")
    <style type="text/css">
        body {
            font-family: Arial, sans-serif, example;
        }
    </style>
@endif
```

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6     <title>Order Invoice</title>
7     <style type="text/css">
8         body {
9             font-family: Arial, sans-serif;
10        }
11    </style>
12    @if($language == "ex")
13    <style type="text/css">
14        body {
15            font-family: Arial, sans-serif, example;
16        }
17    </style>
18    @endif
19 </head>
20
21 <body>
22     @php
23         $contactDetails = $settings->options['contactDetails'];
24         $customer = $order->customer;
25         $shippingAddress = $order->shipping_address;
26         $products = $order->products;
27         $settings = $settings->options;
28         $authorDetails = $settings['contactDetails'];
29         $authorLocation = $authorDetails['location'];

```



If you are facing above the issue just replace these code.

File Location

api\config\pdf.php

'font\_path' => base\_path('resources/fonts/'),

'font\_data' => [

    'example' => [

        'R' => 'example.ttf', // regular font

        'B' => 'example.ttf', // optional: bold font

        'I' => 'example.ttf', // optional: italic font

        'BI' => 'example.ttf', // optional: bold-italic font

        // 'useOTL' => 0xFF,

        'useKashida' => 75,

    ]

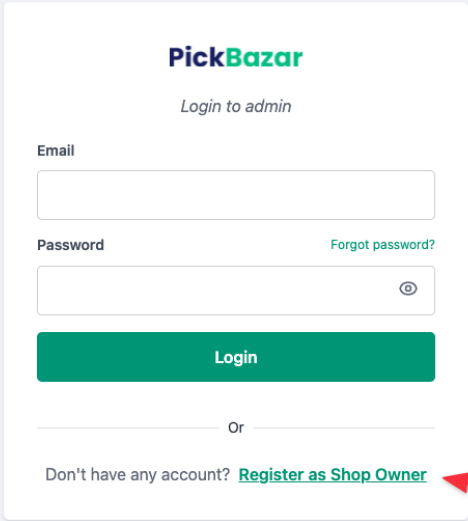
],

If you face any issues with Multilingual, or you've any suggestions to improve the Multilingual feature, then please open a ticket at (<https://redqsupport.ticksy.com/>)[<https://redqsupport.ticksy.com/>]

# Multivendor (Feature)

## Create New Shop

To create new shop login as an **administrator** or create a new account for creating **shop**



The image shows a login form for 'PickBazar'. The form is titled 'PickBazar' and 'Login to admin'. It has two input fields: 'Email' and 'Password'. Below the 'Password' field is a 'Forgot password?' link. A green 'Login' button is positioned below the password field. Below the login button is an 'Or' separator. At the bottom of the form, there is a link that says 'Don't have any account? Register as Shop Owner'. A red arrow points to this link.



PickBazar Create Shop

**Vendor**  
vendor@example.com  
Please add your [Profile](#)  
Enabled

### Create Shop

**Logo**  
Upload your shop logo from here

Upload an image or drag and drop  
PNG, JPG

**Cover Image**  
Upload your shop cover image from here  
Dimension of the cover image should be 1170 x 435px

Upload an image or drag and drop  
PNG, JPG

**Basic Info**  
Add some basic info about your shop from here

Name

Description

**Payment Info**  
Add your payment information from here

Account Holder Name

localhost:3002


And provide all the information for the `store`.

After creating the shop you'll redirect to this page,

PickBazar Create Shop

**Vendor**  
vendor@example.com  
Please add your [Profile](#)  
Enabled

### My Shops

 **Grocery Shop 2**  
Inactive

localhost:3002/grocery-shop-2

By default, the shop will be inactive. Only `administrator` can active a shop.

From `administrator` account go to shop and click tick mark to activate or deactivate` a shop.

PickBazar

Create Shop

Dashboard

Shops

My Shops

Products

Attributes

Groups

Categories

Tags

Orders

Order Status

Users

Coupons

Taxes

Shippings

Withdraws

Settings

Shops

Type your query and press enter

Logo	Name	Owner Name	Products	Orders	Status	Actions
	Grocery Shop 2	Vendor	0	0	Inactive	<a href="#">✓</a> <a href="#">✕</a>
	Bags Shop	Store Owner	15	0	Active	<a href="#">✕</a> <a href="#">✕</a>
	Bakery Shop	Store Owner	72	0	Active	<a href="#">✕</a> <a href="#">✕</a>
	Clothing Shop	Store Owner	64	0	Active	<a href="#">✕</a> <a href="#">✕</a>
	Grocery Shop	Store Owner	249	0	Active	<a href="#">✕</a> <a href="#">✕</a>
	Makeup Shop	Store Owner	81	0	Active	<a href="#">✕</a> <a href="#">✕</a>
	Furniture Shop	Store Owner	55	0	Active	<a href="#">✕</a> <a href="#">✕</a>

<

1

>

After `activate` the shop by `administrator` the `vendor` dashboard will be like this,

PickBazar

Create Shop

Vendor

vendor@example.com

Please add your [Profile](#)

Enabled

My Shops

Grocery Shop 2

Active

After click on `shop`, you'll be redirected to dashboard page,

PickBazar

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Grocery Shop 2

The grocery shop is the best shop around the city. This is being run u...

[Read more](#)

1986 Spinnaker Lane, Freeport, Illinois, 61032, USA, New York, New York, 10001, USA

No contact number available

Visit Shop

Edit Shop

Products

0 Total Products

0 Total Orders

Revenue

\$0.00 Gross Sales

\$0.00 Current Balance

Others

15 % Admin Commission Rate

Registered Since

July 3, 2021

Payment Information

Name: Mahmud

Email: vendor@example.com

From this dashboard, you can maintain you shop,

Dashboard:

PickBazar

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

365

Grocery Shop 2

The grocery shop is the best shop around the city. This is being run u...

Read more

1986 Spinnaker Lane, Freeport, Illinois, 61032, USA, New York, New York, 10001, USA

No contact number available

Visit Shop

Edit Shop

Products

0

Total Products

0

Total Orders

Revenue

\$0.00

Gross Sales

\$0.00

Current Balance

Others

15 %

Admin Commission Rate

Registered Since

July 3, 2021

Payment Information

Name: Mahmud

Email: vendor@example.com

Attributes:

PickBazar

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Attributes

+ Add Attribute

ID	Name	Values	Actions
No data found			

Products:

PickBazar

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Products

Type your query and press enter

+ Add Product

ID	Image	Name	Group	Product Type	Price/Unit	Quantity	Status	Actions
No data found								

Order:

PickBazar

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Orders

Type your query and press enter

Tracking Number	Delivery Fee	Total	Order Date	Status	Shipping Address	table-item-download	Actions
No data found							

## Staff:

PickBazar

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Staff

+ Add Staff

Name	Email	Status	Actions
No data found			



## Withdraw:

PickBazar

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Withdraws

+ Request Withdraw

Shop Name	Amount	Status	Created
No data found			



## User Roles:

### Super Admin:

Super admin can do everything. the admin can maintain and edit every store on the site.

### Store Owner:

Store owner can edit or main it's store, staff or payment.

### Staff:

The staff of a store has similar permission as store owner, but the staff can't update the store or withdraw payment.

# Withdraw Payment:

Only the store owner can withdraw its payment. To do that go to your shop dashboard -> withdraws -> Request a withdraw,

PickBazar

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Withdraws

+ Request Withdraw

Shop Name	Amount	Status	Created
No data found			

PickBazar

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Create Withdraw

Description

Add withdraw request from here

Amount

Payment Method

Details

Note

Request Withdraw

After request payment, the dashboard will be like this,

## + Request Withdraw

< 1 >



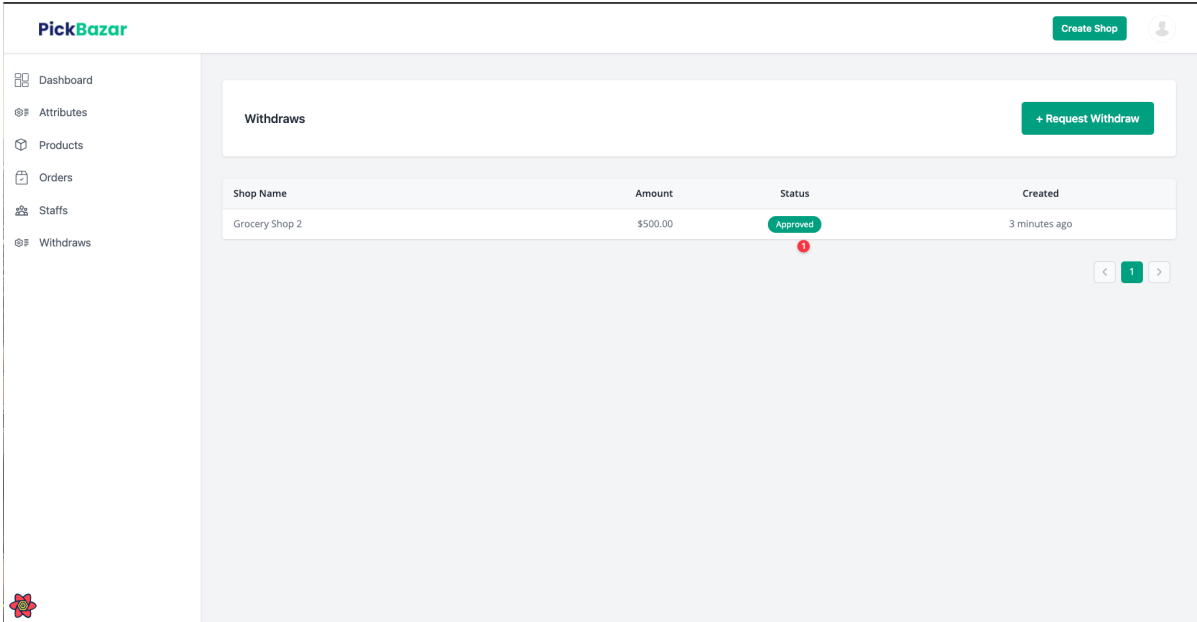
< 1 >

## 1

Pending
Approved
On Hold
Processing
Pending
Rejected

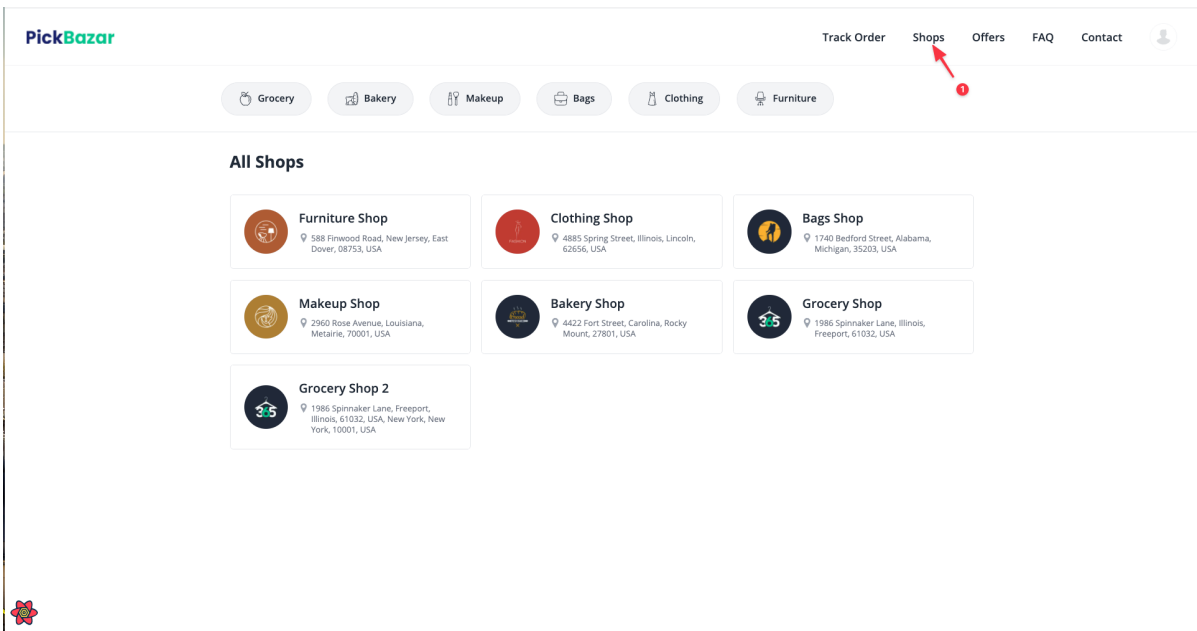
Change Status

After approved,

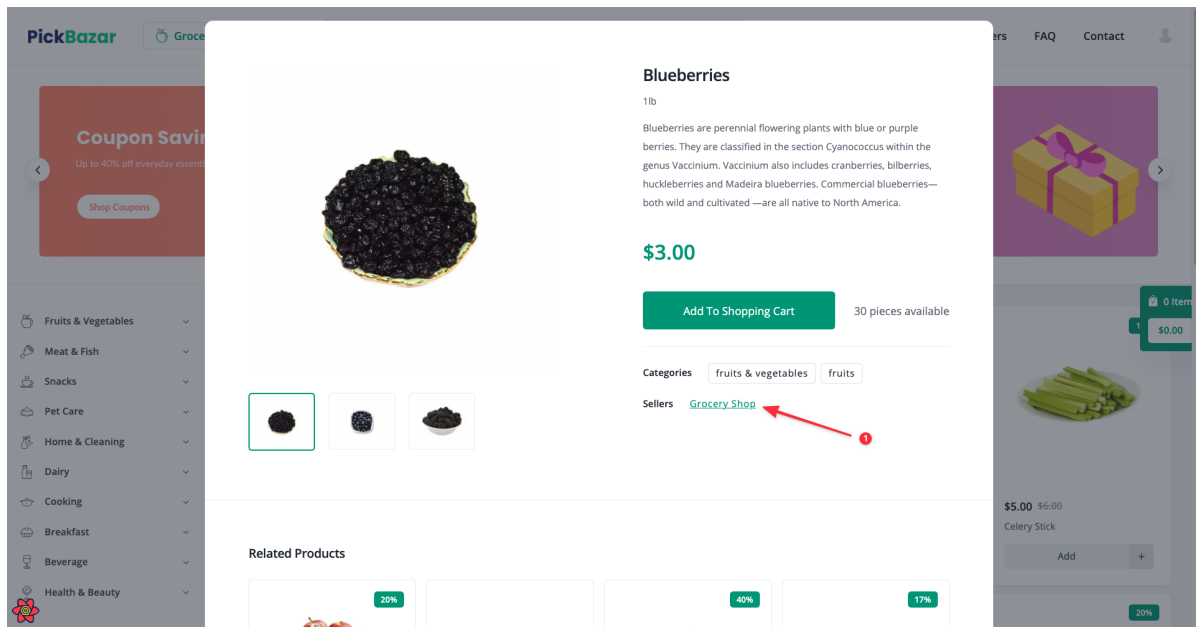


## FrontEnd Shop

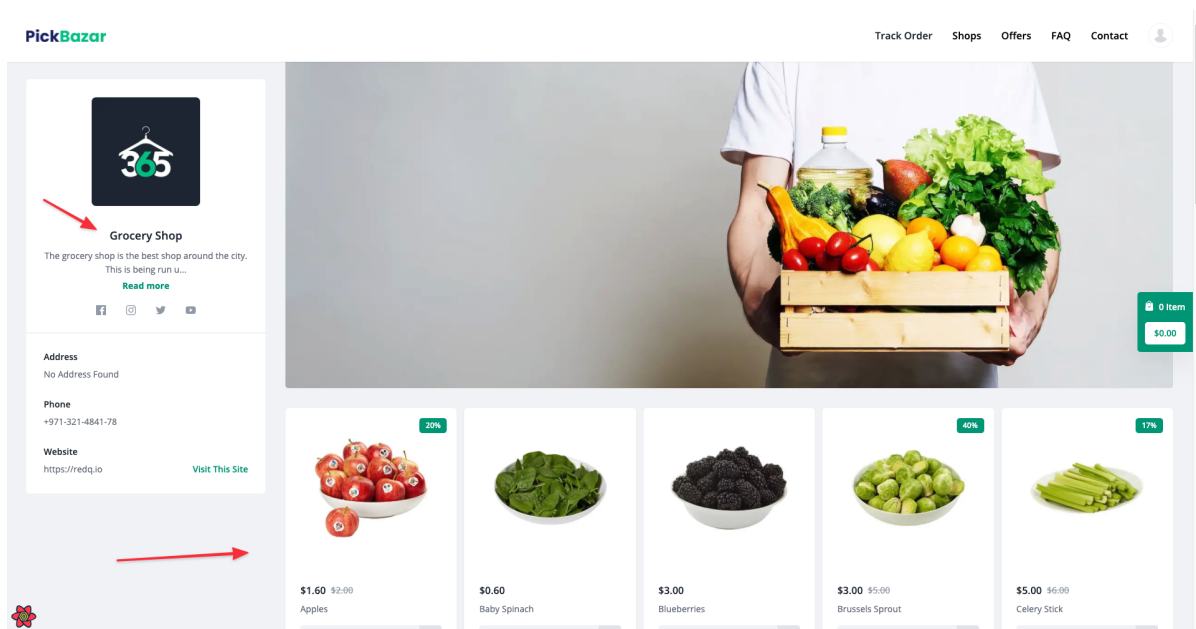
From frontend when customers click on the `shop` page they'll get all the shops as a card list,



Also when customers click on a product they will get the seller link,



After clicking the **seller link** they will redirect to the seller shop page,



## Export Import (Feature)

Export Import works under a shop. So, You have to go to the **shop page** -> **products menu** for product or **shop page** -> **attributes menu** to import or export attributes.

PickBazar

Shop Page

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Products

Type your query and press enter

+ Add Product

Filter

Image	Name	Group	Product Type	Price/Unit	Quantity	Status	Actions
	Nair Hair Removal Cream	Grocery	simple	\$8.00	50	publish	
	Espa Men Dual Action	Grocery	simple	\$6.00	50	publish	
	Human Nature Shaving Cream	Grocery	simple	\$6.00	50	publish	
	Longmarket Barber Aftershave Balm	Grocery	simple	\$6.00	40	publish	
	Pacific Natural Shaving Cream	Grocery	simple	\$6.00	50	publish	
	Nuace Men Facial Wash	Grocery	simple	\$3.00	50	publish	
	Garnier Pure Active	Grocery	simple	\$2.00	50	publish	
	Frei Oil	Grocery	simple	\$2.00	50	publish	
	Face Replublic Sun Gel	Grocery	simple	\$2.00	50	publish	

Export Import work different way for simple and variable product

## Simple Product

For simple products, you've to export only products data. To do that, go to **Your Shop -> products**. Then click on **three dots** and export the simple products.

PickBazar

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Products

Type your query and press enter

+ Add Product

Filter

Image	Name	Group	Product Type	Price/Unit	Quantity	Status	Actions
	Nair Hair Removal Cream	Grocery	simple	\$8.00	50	publish	
	Espa Men Dual Action	Grocery	simple	\$6.00	50	publish	
	Human Nature Shaving Cream	Grocery	simple	\$6.00	50	publish	
	Longmarket Barber Aftershave Balm	Grocery	simple	\$6.00	40	publish	
	Pacific Natural Shaving Cream	Grocery	simple	\$6.00	50	publish	
	Nuace Men Facial Wash	Grocery	simple	\$3.00	50	publish	
	Garnier Pure Active	Grocery	simple	\$2.00	50	publish	
	Frei Oil	Grocery	simple	\$2.00	50	publish	
	Face Replublic Sun Gel	Grocery	simple	\$2.00	50	publish	

Export/Import

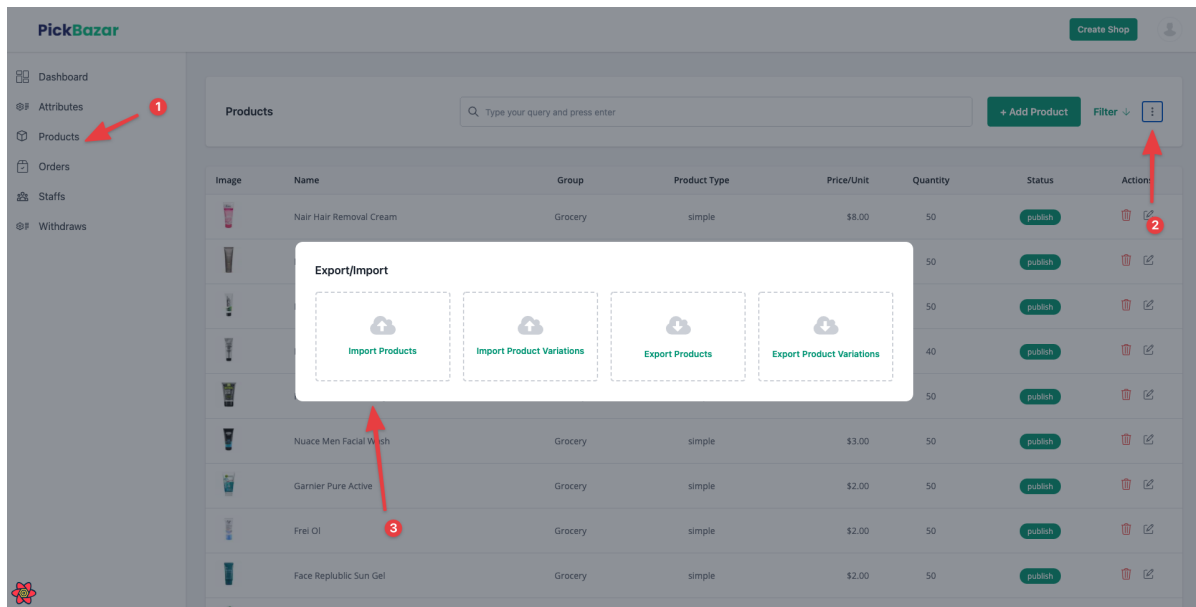
Import Products

Import Product Variations

Export Products

Export Product Variations

Similarly, for import **csv**, go to the same option and import exported **csv**



## variable products

For variable products, you will need three different `csv`.

- Attributes `csv`
- products `csv`
- variation options `csv`

Make sure to re-import you export these three csv .

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Attributes

Order By

DESC

+ Add Attribute

ID	Name	Values	Actions
5	colors	red, green	

Export/Import

Import Attributes

Export Attributes

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Products

Type your query and press enter

+ Add Product

Filter

Image	Name	Group	Product Type	Price/Unit	Quantity	Status	Actions
	Nair Hair Removal Cream	Grocery	simple	\$8.00	50	publish	
					50	publish	
					50	publish	
					40	publish	
					50	publish	
	Nuace Men Facial Wash	Grocery	simple	\$3.00	50	publish	
	Garnier Pure Active	Grocery	simple	\$2.00	50	publish	
	Frei Oil	Grocery	simple	\$2.00	50	publish	
	Face Republic Sun Gel	Grocery	simple	\$2.00	50	publish	

Export/Import

Import Products

Import Product Variations

Export Products

Export Product Variations

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Attributes

Order By

DESC

+ Add Attribute

ID	Name	Values	Actions
5	colors	red, green	

Export/Import

Import Attributes

Export Attributes

PickBazar

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Products

Search: Type your query and press enter

+ Add Product

Filter

Image

Name

Group

Product Type

Price/Unit

Quantity

Status

Actions

Nair Hair Removal Cream

Grocery

simple

\$8.00

50

publish

Export/Import

Import Products

Import Product Variations

Export Products

Export Product Variations

Nuace Men Facial Wash

Grocery

simple

\$3.00

50

publish

Garnier Pure Active

Grocery

simple

\$2.00

50

publish

Frei Oil

Grocery

simple

\$2.00

50

publish

Face Replublic Sun Gel

Grocery

simple

\$2.00

50

publish

Now, during import variations product,  
At First, you have to import Attributes  
then Products and  
Then products variation.

PickBazar

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Attributes

Order By

DESC

+ Add Attribute

ID

Name

Values

Actions

5

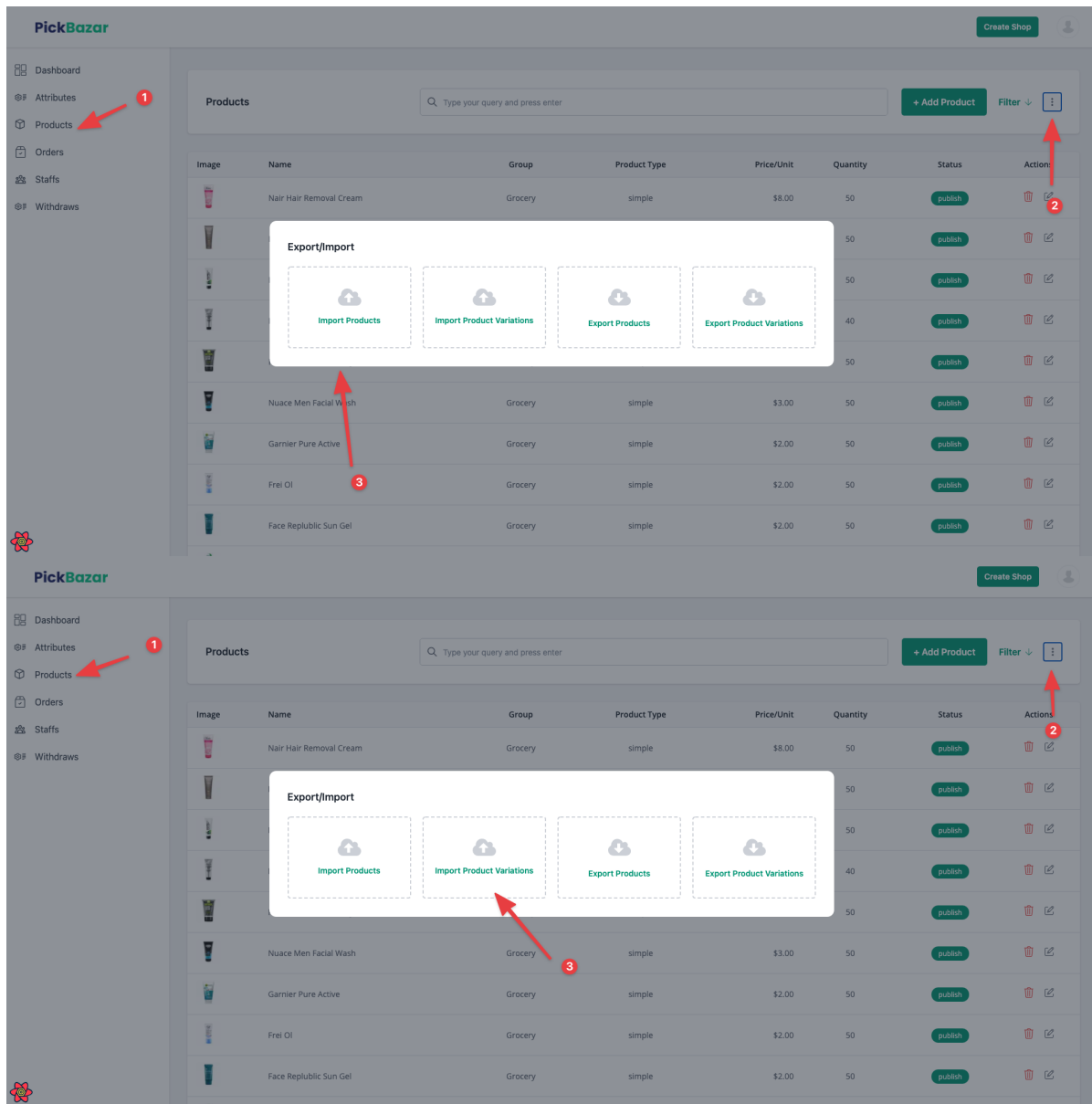
colors

red, green

Export/Import

Import Attributes

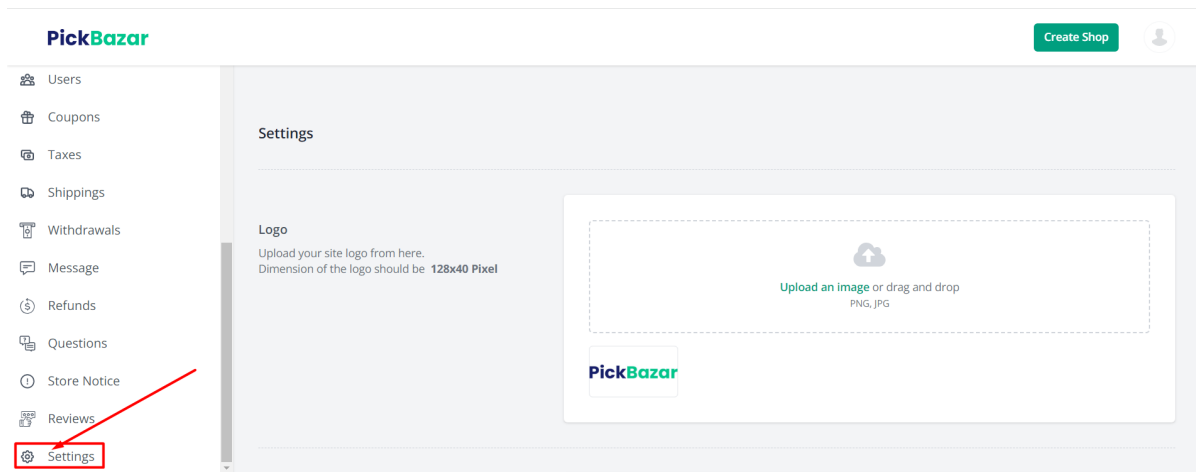
Export Attributes



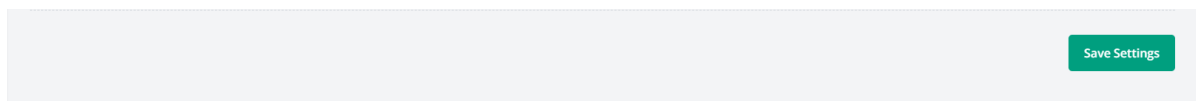
NB: Here, type\_id is the group id like Grocery or clothing id. You can get type info from `api_url/types url;`

## Settings Management

In `settings` menu you will get the settings management form there. You can modify the basic information about your E-Commerce System, such as the Change Logo, Site Title, Site Information, Payment Information, SEO, Delivery Schedule, Shop Settings, etc.

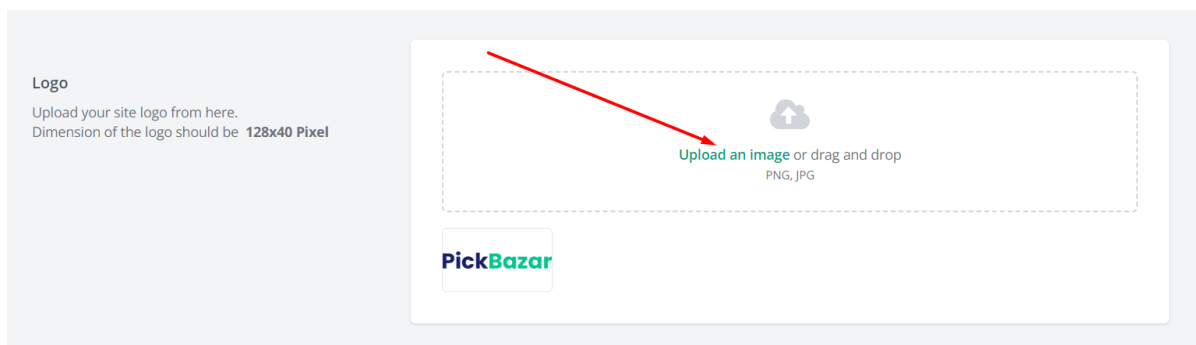


**Note:** If you make changes or update any fields, it's important to save those changes by clicking on the `Save Settings` button bottom of the page.



## Change Logo

Click on Upload an image and select the file for your new logo from your Device. or Drag and Drop your new logo. (Supported Formats JPG or PNG)



We support two types of file uploads; one is `local`, and another is `AWS S3`.

Both uploading systems follow this procedure,

## API

At first, add your API URL to your `api -> .env`.

```
2 APP_ENV=production
3 APP_KEY=
4 APP_DEBUG=true
5 APP_URL=https://YOUR_DOMAIN.COM/backend
6 APP_SERVICE=marvel.test
7 APP_NOTICE_DOMAIN=PICKBAZAR_
8 DUMMY_DATA_PATH=pickbazar
9
10
11 LOG_CHANNEL=stack
12 LOG_LEVEL=debug
13
```

If you're using windows and your API is running with ports like `localhost:8000` or `127.0.0.1:8000`, then make sure you add the domain with port to `APP_URL`. Like this

`API_URL=http://localhost:8000`

After that, clear your API cache by using this command from the `api` folder,

```
php artisan optimize:clear
```

Then link the storage system by using this command,

```
php artisan storage:link
```

If you're using AWS S3, then update `MEDIA_DISK=s3` and add all `AWS_` credentials  
For AWS s3, make sure your properly setup permission of the bucket with ACL enable by follow this link -> <https://stackoverflow.com/a/70603995/2158023>

## Change Site information

This is the Site `Information` area of the admin settings. Here you can change your Site Title, Site Subtitle, Currency, Minimum Order Amount, Wallet Currency Ratio, Sign Up Points, Maximum Question Limit, OTP checkout, Tax Class, Shipping Class and handle Free Shipping.

Information  
Change your site information from here

Site Title  
Pickbazar

Site Subtitle  
Your next ecommerce

Currency  
US Dollar

Minimum Order Amount  
0

Wallet Currency Ratio  
3

## Shipping class, Tax class & Free Shipping

You may find the shipping class at the conclusion of the information section. You may choose the precise class you wish to utilize from this menu. These classes were developed within the shipping category. The classes include a value for shipping that should be added to the total price bought.

For tax classes, the same applies. You may create tax classes in the admin settings' tax area by entering a dollar amount or percentage that will be added to the total amount of your order.

The option to allow free delivery is another one. It may be turned off to stop offering free delivery. Customers will receive free delivery if it is activated. If not, the cost of shipment must be paid.

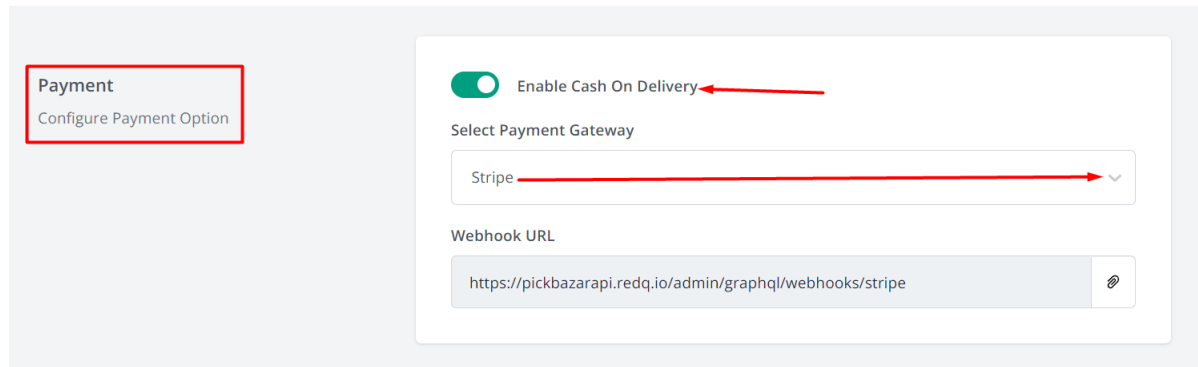
Tax Class  
Global

Shipping Class  
Select...

☐ Enable Free Shipping

# Payment Gateway Management

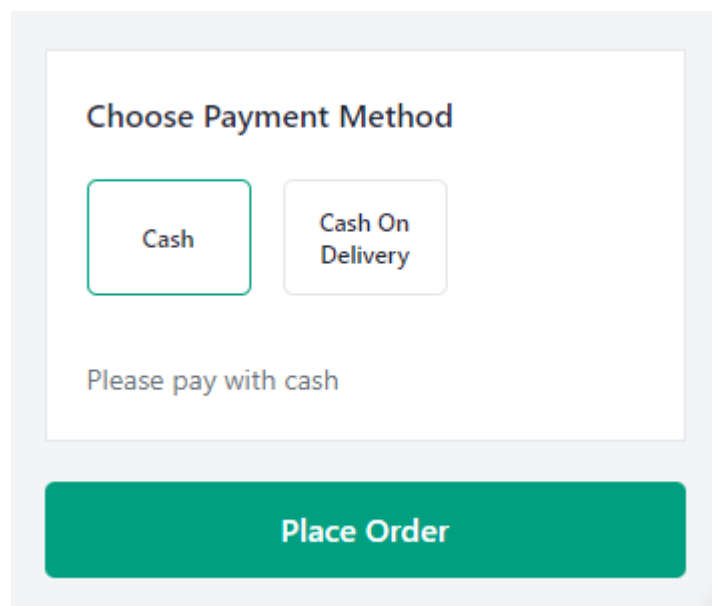
This is the **payment** area of the admin settings. This will allow you to choose a payment gateway such as Stripe, Rezipay, Paypal, Mollie Paystack etc. You can also enable or disable Cash On Delivery.



The screenshot shows the 'Payment' settings section in an admin dashboard. On the left, a sidebar menu has 'Payment' and 'Configure Payment Option' highlighted with a red box. The main content area contains a toggle switch for 'Enable Cash On Delivery' which is turned on, with a red arrow pointing to it. Below this is a 'Select Payment Gateway' dropdown menu with 'Stripe' selected, also indicated by a red arrow. At the bottom, there is a 'Webhook URL' field containing the text 'https://pickbazarapi.redq.io/admin/graphql/webhooks/stripe' and a copy icon.

## Super Admin

The admin will see **Cash** and **Cash on delivery** when he will go to **Choose Payment Method** after generating an order from the shop. Because the buyer may pay while purchasing the product directly from the administrator. For customers who will pay cash after getting the merchandise distant from the admin, there is also the option of cash on delivery.



The screenshot displays the 'Choose Payment Method' screen for a customer. It features two buttons: 'Cash' and 'Cash On Delivery'. The 'Cash' button is highlighted with a green border. Below the buttons, the text 'Please pay with cash' is displayed. At the bottom of the screen is a large green button labeled 'Place Order'.

## Customer

When a client goes to a store to buy anything, he adds it to his cart. When he goes to pick a payment option, he will see **Cash on delivery** as well as payment gateways such as Stripe, Paypal, Razorpay, and Mollie, which are configured in the payment settings area of the dashboard.

### Choose Payment Method

stripe

Cash On Delivery

Please click Place order to make order and payment

Place Order

## SEO

Search Engine Optimization ( **SEO** ) is a process of optimizing your website to rank higher in search engine results pages (SERPs) for specific keywords and phrases. you can add Meta Title, Meta Description, Meta Tags, Canonical URL, OG Title, OG Description etc.

SEO  
Change your site SEO from here

Meta Title

Meta Description

Meta Tags

Canonical URL

OG Title

OG Description

## Add or Remove Delivery Schedule

This is the **Delivery Schedule** area of the settings. This will allow you to add new Delivery Schedule such as Morning, Noon, Afternoon, Evening etc. You can also remove any existing Delivery Schedule.

Delivery Schedule  
Add your delivery schedule time with proper description from here

Express Delivery

90 min express delivery

Remove

Morning

8:00 AM - 11:00 AM

Remove

Noon

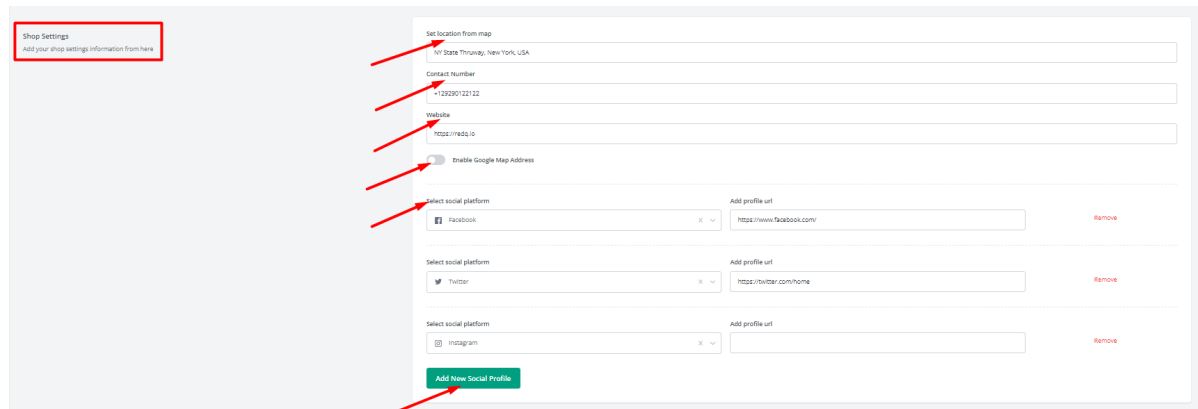
11:00 AM - 2:00 PM

Remove

Add Delivery Time

# Shop Settings & Google Map

This is the **Shop Settings** area of the settings. This will allow you to change location from map, Contact Number, Website, Enable or Disable Google Map Address, or add new social platform account, etc.



The screenshot shows the 'Shop Settings' form. A red box highlights the 'Shop Settings' header. Red arrows point to the following fields: 'Set location from map' (with a map icon), 'Contact Number' (with a phone icon), 'Website' (with a globe icon), 'Enable Google Map Address' (a toggle switch), 'Select social platform' (a dropdown menu), 'Add profile url' (a text input field), and 'Add New Social Profile' (a green button at the bottom).

## Payment

We have introduced the new **Payment** architecture feature on PickBazar **v6.0.0**. So if you want to use **Payment** with **PickBazar**, then make sure your PickBazar is **v6.0.0** or later.

We have plan to enrich this feature area by integrating more payment gateways in future. List of available payment gateways now.

- **Stripe**
- **Stripe Element**
- **PayPal**
- **RazorPay**
- **Mollie**
- **Xendit**
- **Paystack**
- **SSLCOMMERZ**
- **Iyzico**
- **bKash**
- **Paymongo**
- **Flutterwave**

Let's discuss those sequentially.

# Stripe

Stripe is a financial service & Software as a service (SaaS) company. It offers payment processing software & API for e-commerce applications. In PickBazar we have integrated Stripe API for payment system. Though currently `Card` based features are available only, `Stripe Element` for other payment options will be integrate in future updates.

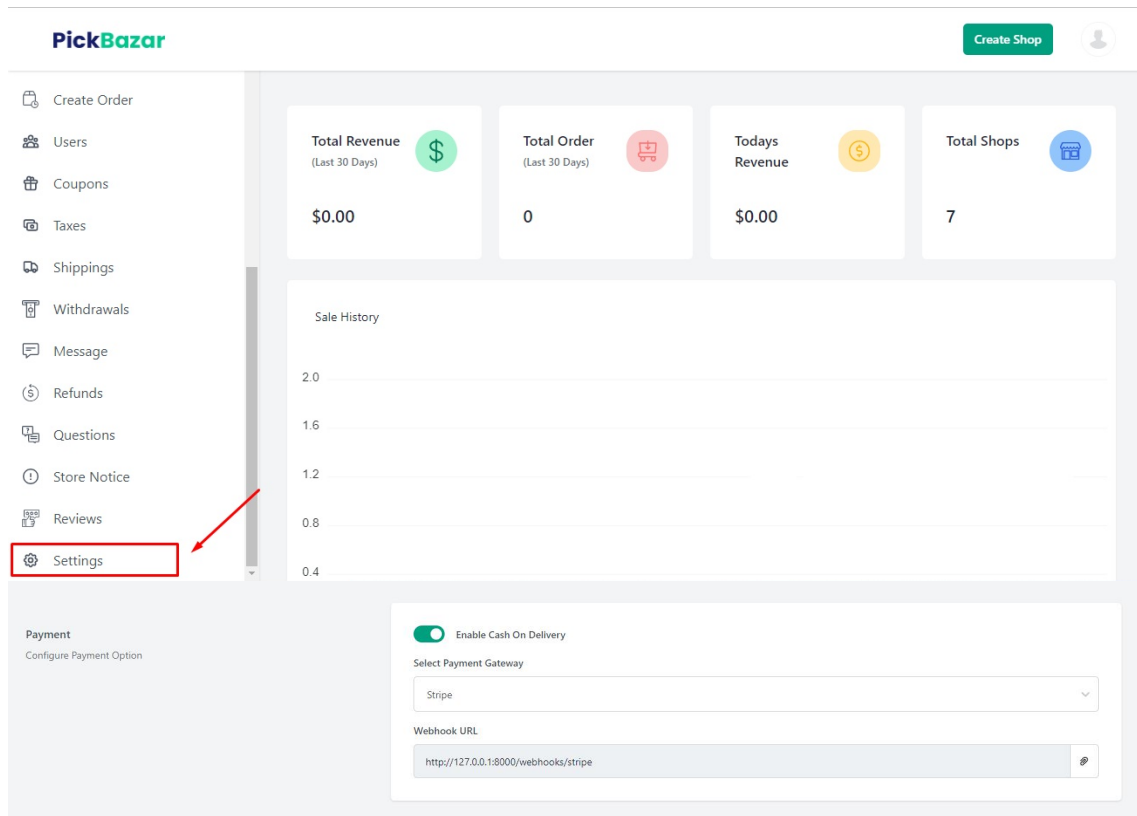
## Stripe integrate inside PickBazar.

Please follow & complete this steps for stripe integration for your e-commerce system.

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in stripe.

```
STRIPE_API_KEY=[YOUR_STRIPE_API_SECRET_KEY]
STRIPE_WEBHOOK_SECRET_KEY=[YOUR_STRIPE_API_WEBHOOK_SECRET_KEY]
```

- To activate Stripe go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option.(e.g. webhook URL is coming from local development. This static link will dynamically generated in live environment)

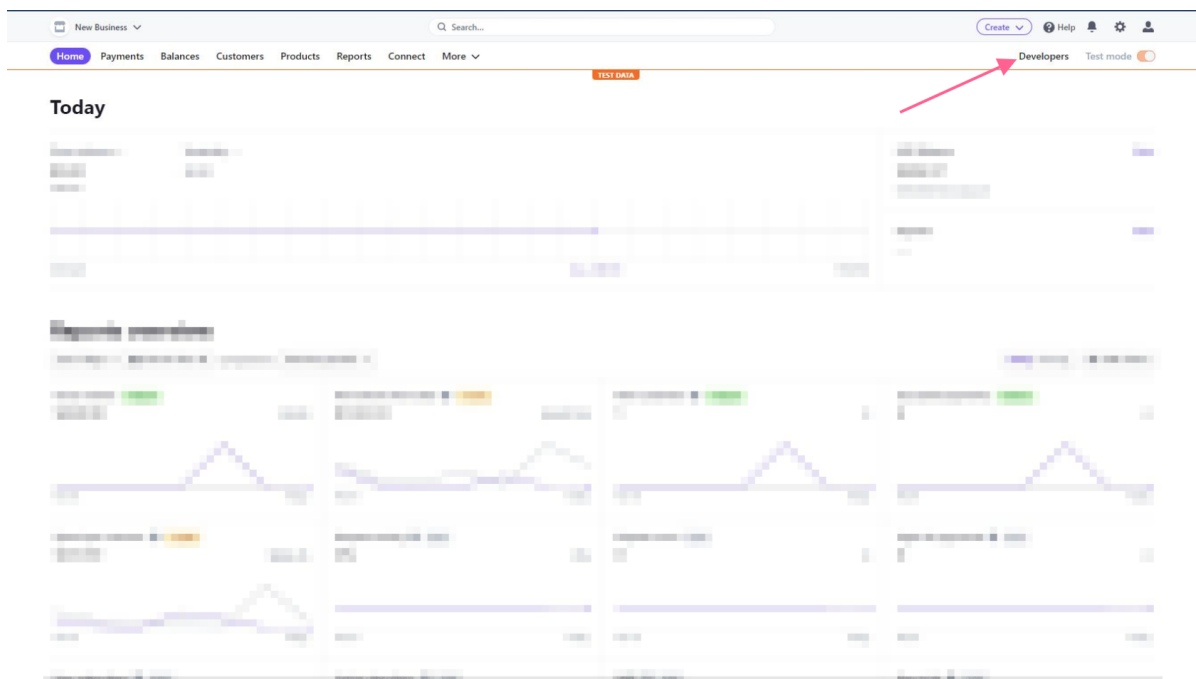


- Add Stripe publishable key inside `.env` file of pickbazar shop.

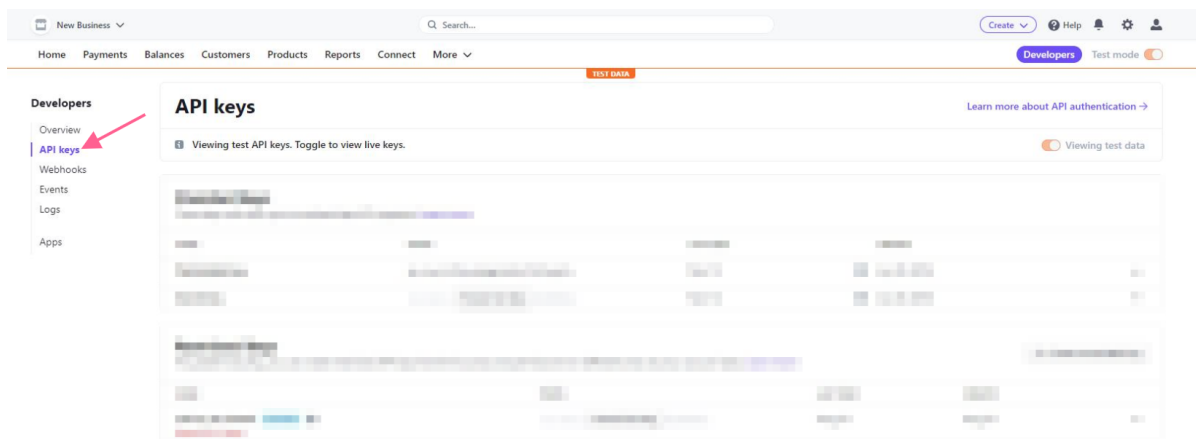
```
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=[YOUR_STRIPE_PUBLISHABLE_KEY]
```

## How to create & setup Stripe information properly?

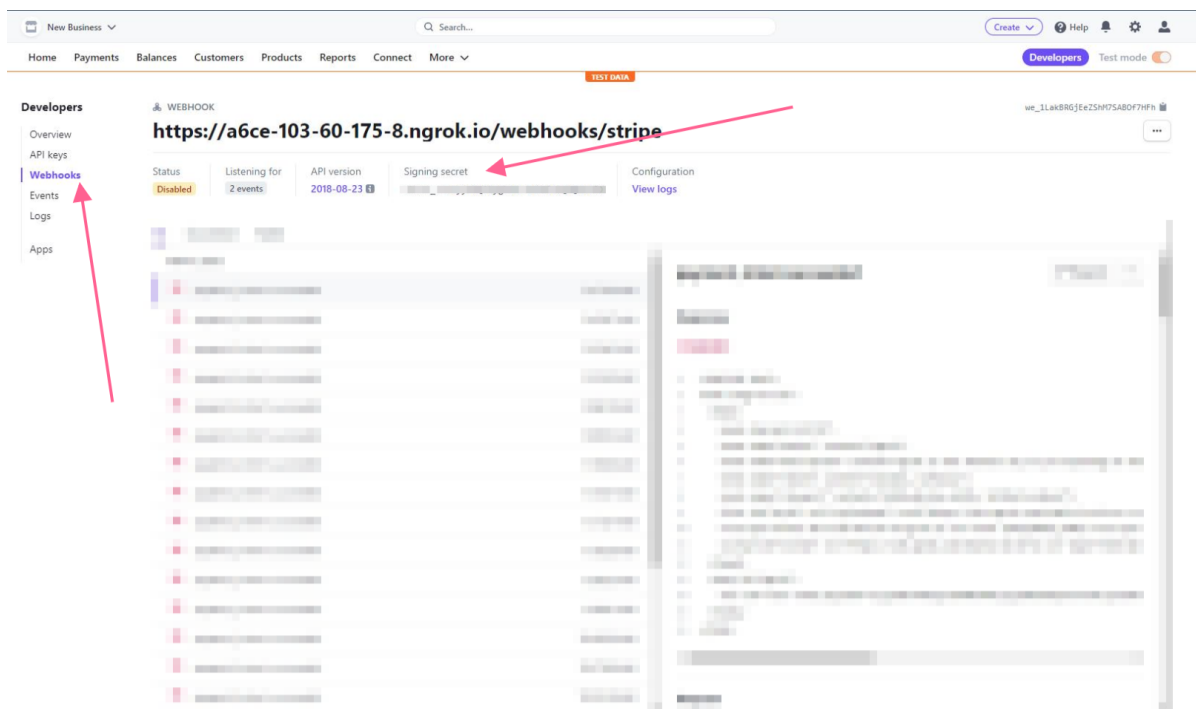
- Go to the stripe website and login <https://stripe.com> If you aren't a registered user, the complete the stripe registration first.
- After logged in into stripe dashboard, follow the developer link to create the API keys.



- Then create API keys from there. for more details follow up this official documentation <https://stripe.com/docs/keys>



- Create Webhook secret key if you decide to up & running webhooks in your App.



- Create this two webhook events for monitoring the payment flow.
  - **payment\_intent.succeeded**
  - **payment\_intent.payment\_failed**

## Special Notes for Stripe users.

If we have used any third party system/plugin/packages, then we have always encouraged our respected customers to follow the official documentation for detailed & in-depth knowledge.

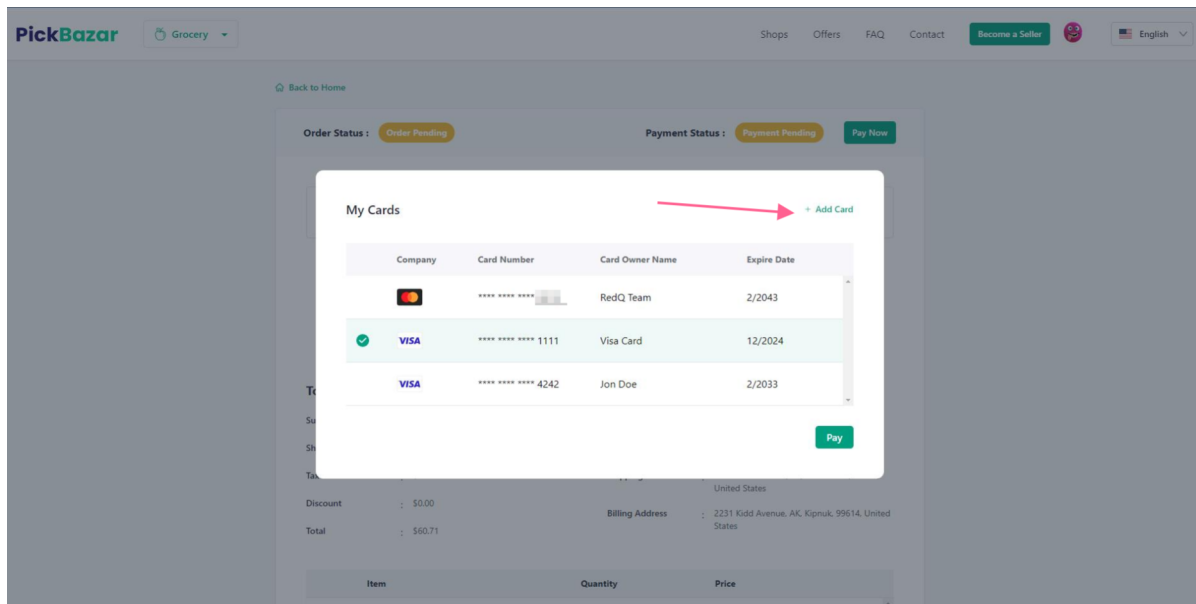
- Stripe DOCS [home page] -- <https://stripe.com/docs>
- Stripe API Docs -- <https://stripe.com/docs/api>
- Stripe JavaScript Docs -- <https://stripe.com/docs/js>

## How can I add card in my user profile for future payments in Stripe?

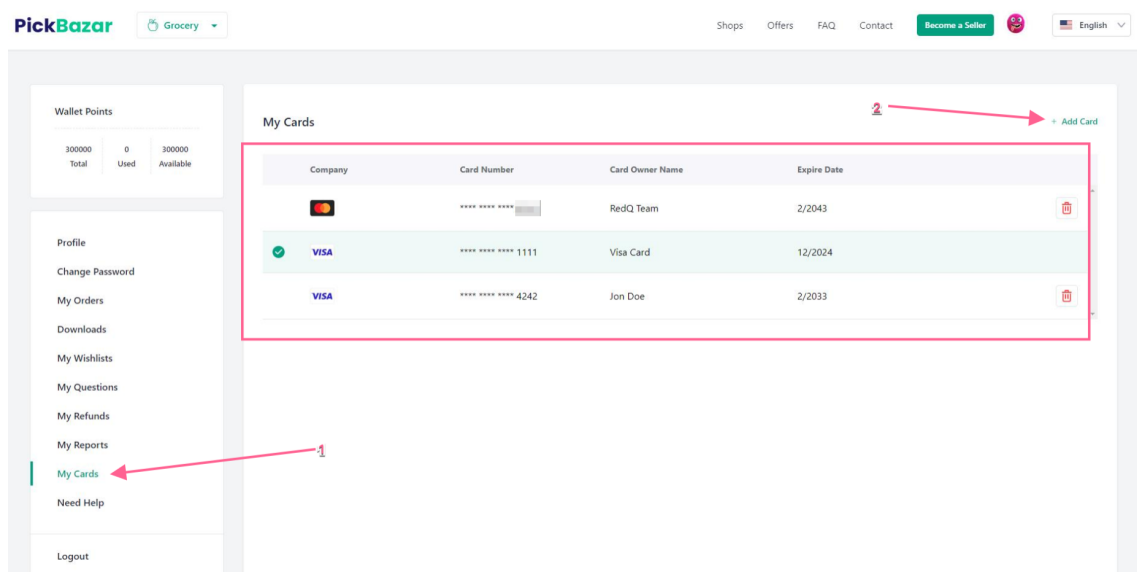
In PickBazar we have provided an feature for saving card in case of future usages. This is an **on-session** process. So, if a customer wants to pay via Cards, s/he must have present in the application lively. No **off-session** payment was applied here.

A customer can save a card via two process.

- Save card during checkout process.



- Save card in his user profile **My Card** section.



There are something needs to keep in mind.

### Please note

- No confidential information was saved in this features. By maintaing the guideline only available information which are permitted to save via Stripe is implemented here.
- Guest user can't save card for future payment.

## Stripe Element

Stripe Elements is a set of prebuilt UI components which is provided by stripe itself for building your web payment flow. It's available as a feature of Stripe.js, Stripe Elements foundational JavaScript library for building functional checkout flows.

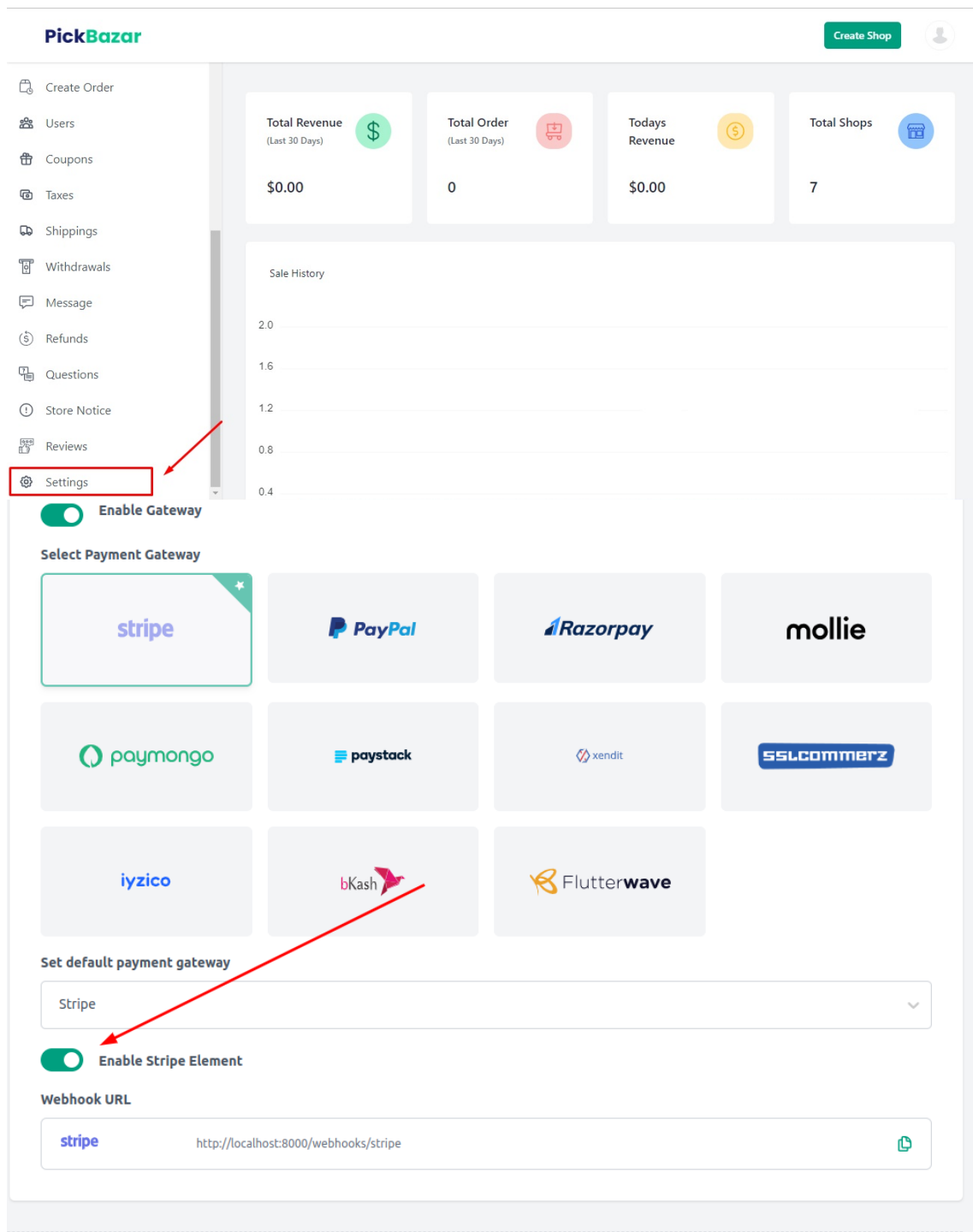
# Stripe Element integrate inside PickBazar.

Please follow & complete this steps for stripe integration for your e-commerce system.

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file.  
We will discuss later about how to create those API keys in stripe.

```
STRIPE_API_KEY=[YOUR_STRIPE_API_SECRET_KEY]
STRIPE_WEBHOOK_SECRET_KEY=[YOUR_STRIPE_API_WEBHOOK_SECRET_KEY]
```

- To activate Stripe go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option.(e.g. webhook URL is coming from local development. This static link will dynamically generated in live environment)

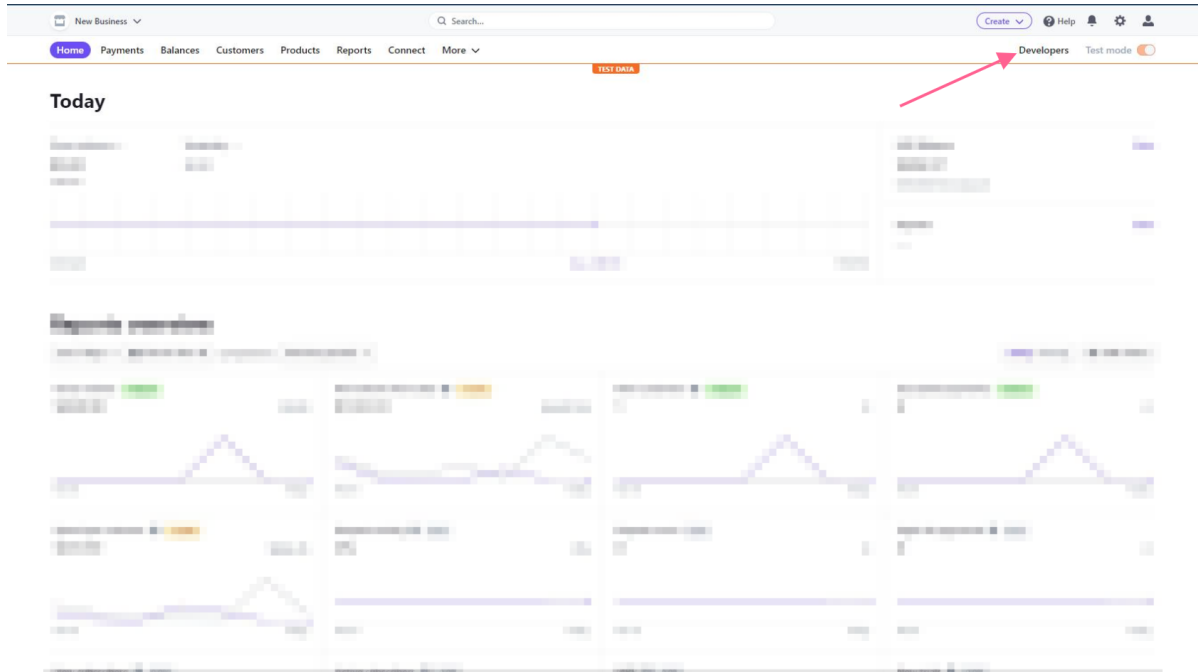


- Add Stripe publishable key inside `.env` file of pickbazar shop.

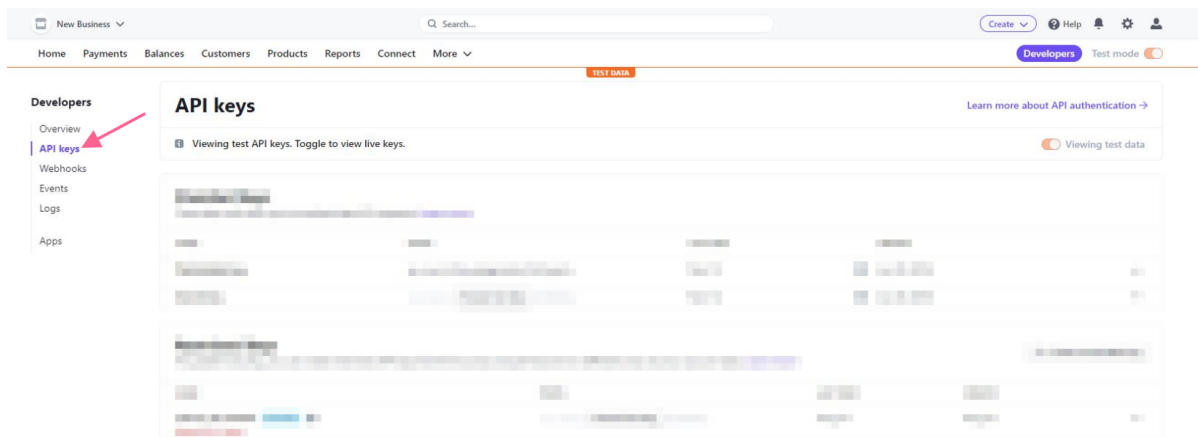
NEXT\_PUBLIC\_STRIPE\_PUBLISHABLE\_KEY=[YOUR\_STRIPE\_PUBLISHABLE\_KEY]

## How to create & setup Stripe Element information properly?

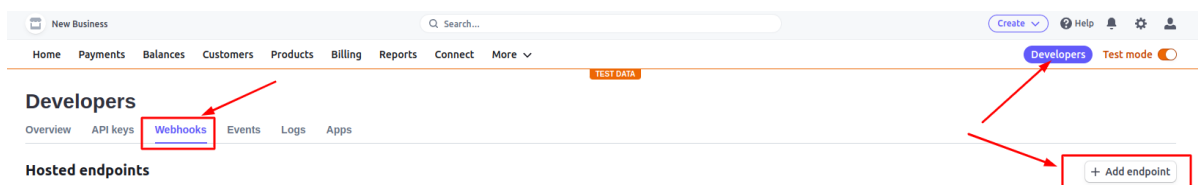
- Go to the stripe website and login <https://stripe.com> If you aren't a registered user, the complete the stripe registration first.
- After logged in into stripe dashboard, follow the developer link to create the API keys.



- Then create API keys from there. for more details follow up this official documentation <https://stripe.com/docs/keys>



If you want to integrate stripe element in pickbazar then you must have to configure webhook properly.



- Drop your webhook endpoint and select all event then "Add endpoint"

- **Note:** We recommended you to enable all event because stripe element's payment method vary on currency. But you can enable those event that are specifically depend on your currency or you can enable all event as recommend.

## Listen to Stripe events

[Add an endpoint](#) [Test in a local environment](#)

Set up your webhook endpoint to receive live events from Stripe or [learn more about Webhooks](#).

**Endpoint URL**

**Description**

An optional description of what this webhook endpoint is used for...

**Listen to**

☒ Events on your account ☐ Events on Connected accounts ⓘ

**Version**

Your current version (2019-12-03) ⇅

**Select events to listen to**

[+ Select events](#)

[Add endpoint](#) [Cancel](#)

The screenshot shows the Stripe Developers dashboard. On the left, the 'Webhooks' menu item is highlighted. The main content area shows a specific webhook configuration for the URL `https://a6ce-103-60-175-8.ngrok.io/webhooks/stripe`. The status is 'Disabled', it is listening for '2 events', and the API version is '2018-08-23'. A 'Signing secret' is displayed and redacted. The 'Configuration' tab is active, showing a list of events and their details. A red arrow points from the 'Select events to listen to' button in the first image to the '2 events' status in this image. Another red arrow points from the 'Add endpoint' button in the first image to the 'Webhooks' menu item in this image.

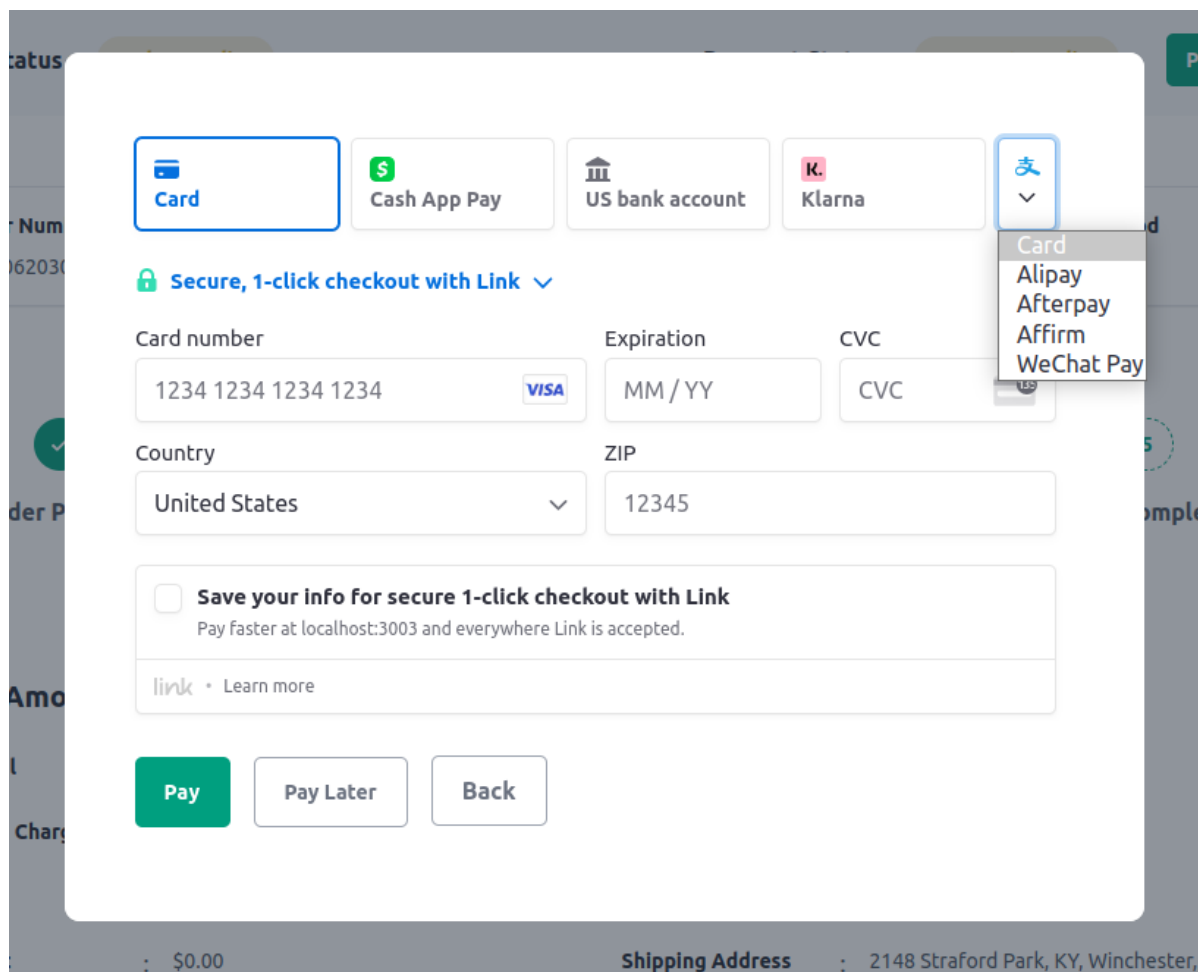
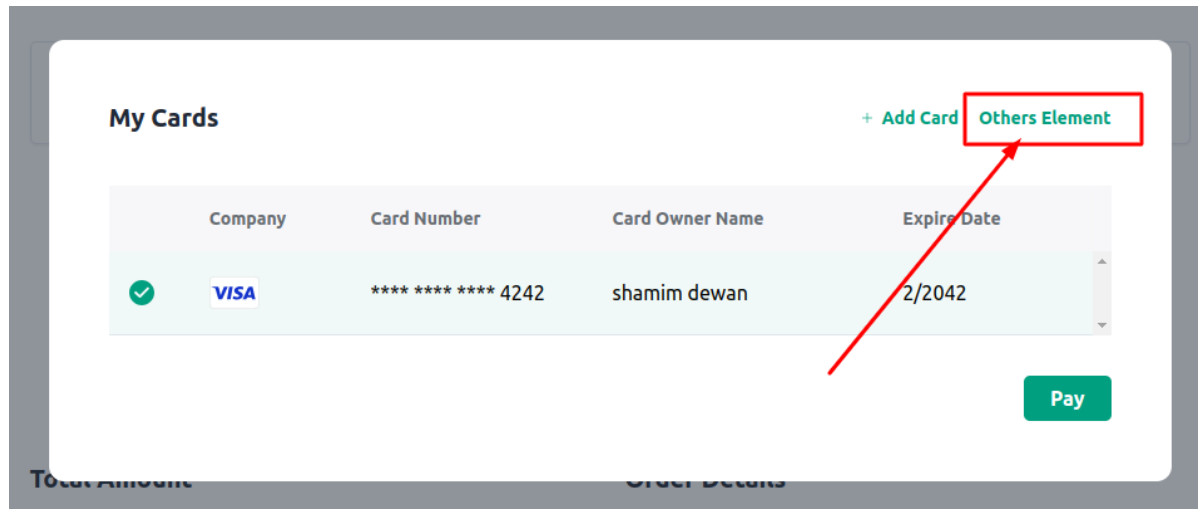
Again! To integrate stripe element in pickbazar then you must have to configure webhook properly.

## How can I add stripe element's another payment gateway?

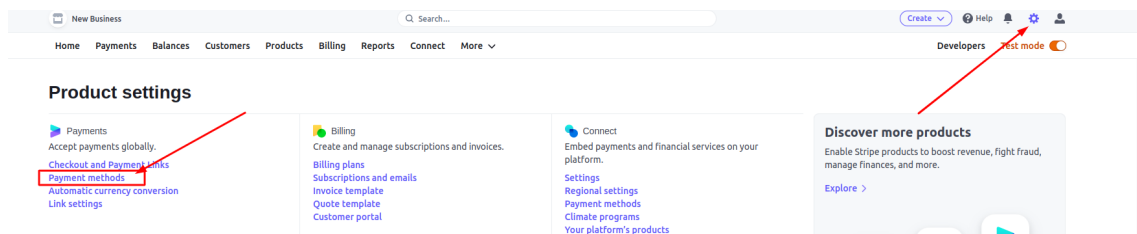
We have a fully functional card management feature in Pickbazar. Where you can save card for further use.

[Click here for more details.](#)

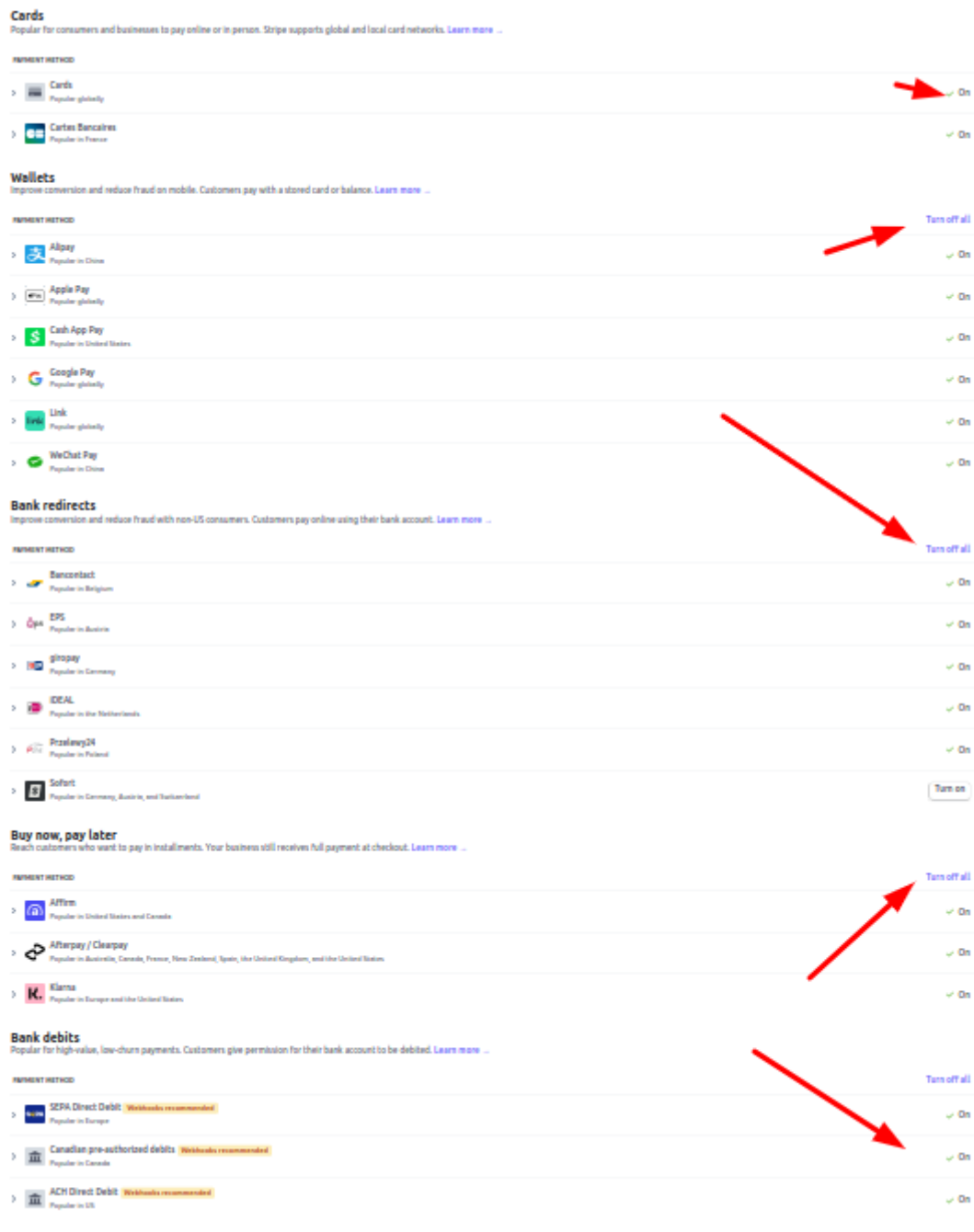
If you enable stripe element from admin settings, then you will get "Others element" option in stripe card.



- Enable your desired stripe element payment gateway from stripe dashboard.



- Turn on your gateway.



## How to customize your stripe element payment method.

If you want selective payment method type then list those method name in the "payment\_method\_types" array and uncomment this

```
'payment_method_types' => ['card', 'cashapp', 'alipay', 'wechat_pay', 'etc']
```

and comment

```
'automatic_payment_methods' => ['enabled' => true,],
```

lines

or you can enabled true "automatic\_payment\_methods" array. so that stripe can handle everything

## Special Notes for Stripe element users.

If we have used any third party system/plugin/packages, then we have always encouraged our respected customers to follow the official documentation for detailed & in-depth knowledge.

## Available Stripe Element Payment Method

Stripe element's payment method vary on currency

```
acss_debit, affirm, afterpay_clearpay, alipay, au_becs_debit, bacs_debit,
bancontact, blik, boleto,
card, cashapp, customer_balance, eps, fpx, giropay, grabpay, ideal, klarna,
konbini, link,
oxxo, p24, paynow, paypal, pix, promptpay, sepa_debit, sofort, us_bank_account,
wechat_pay, zip,
```

- Stripe DOCS [home page] -- <https://stripe.com/docs>
- Stripe API Docs -- <https://stripe.com/docs/api>
- Stripe JavaScript Docs -- <https://stripe.com/docs/js>
- Stripe Webhooks Settings -- <https://dashboard.stripe.com/webhooks>
- Stripe Payment Method Settings -- [https://dashboard.stripe.com/test/settings/payment\\_methods](https://dashboard.stripe.com/test/settings/payment_methods)
- Stripe Payment Method List -- [https://stripe.com/docs/api/payment\\_methods/list](https://stripe.com/docs/api/payment_methods/list)

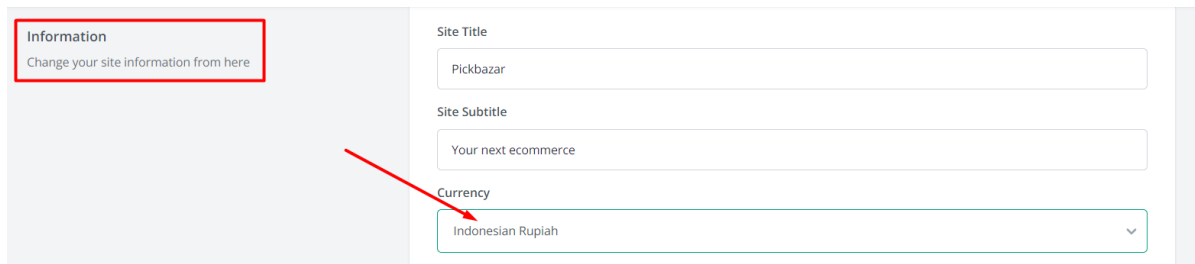
## Xendit

Xendit is a digital payments infrastructure provider that operates in Southeast Asia, particularly in Indonesia, the Philippines, and Singapore. Xendit offers a variety of payment services, including online and offline payments, virtual accounts, and digital wallets. It also supports multiple payment methods, such as credit and debit cards, bank transfers, and e-wallets.

## Xendit integrate inside PickBazar.

Please follow & complete this steps for Xendit integration for your e-commerce system.

- First go to settings from PickBazar admin dashboard. Inside settings you will find Currency option. Select your Currency. If you want to Test Indonesian Account, then Select Indonesian rupiah (IDR).



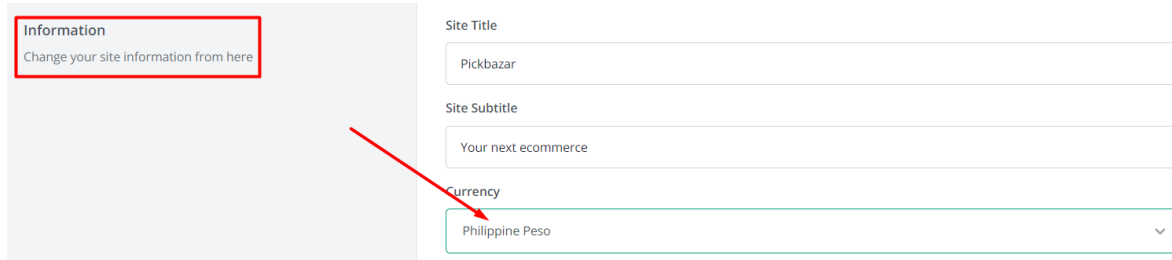
Information  
Change your site information from here

Site Title  
Pickbazar

Site Subtitle  
Your next ecommerce

Currency  
Indonesian Rupiah

Or If you want to Test philippine Account, then Select Philippine peso(PHP).



Information  
Change your site information from here

Site Title  
Pickbazar

Site Subtitle  
Your next ecommerce

Currency  
Philippine Peso

***WARNING:*** Any order below the minimum amount of `10000` IDR or PHP will not be processed, please make sure to meet the minimum order requirement before placing an order.

For more information: [Xendit currency is available](#).

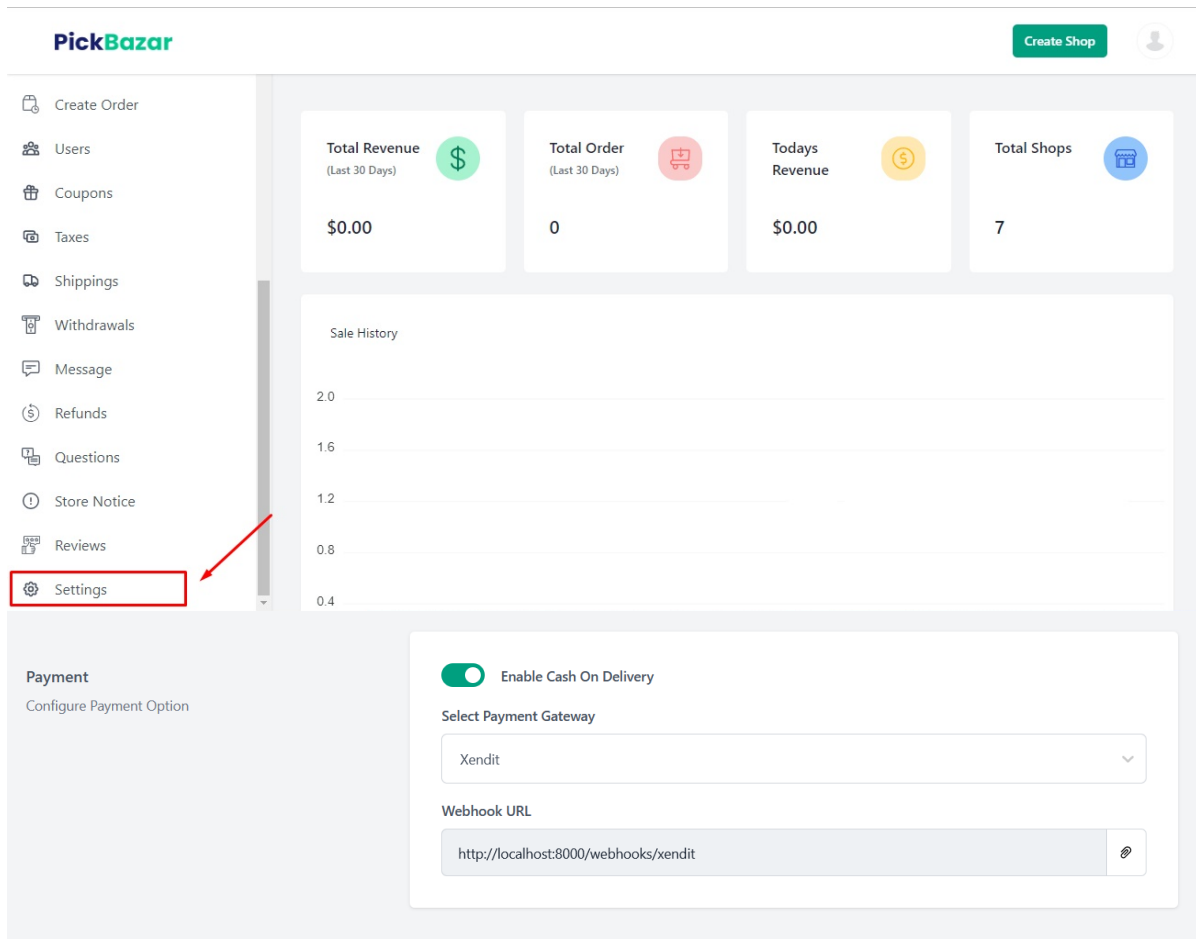
- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in [Xendit official website](#).

```
API_KEY=[YOUR_XENDIT_SECRET_KEY]
XENDIT_WEBHOOK_URL=[YOUR_XENDIT_WEBHOOK_URL]
XENDIT_CALLBACK_TOKEN=[YOUR_XENDIT_CALLBACK_VERIFICATION_TOKEN]
```

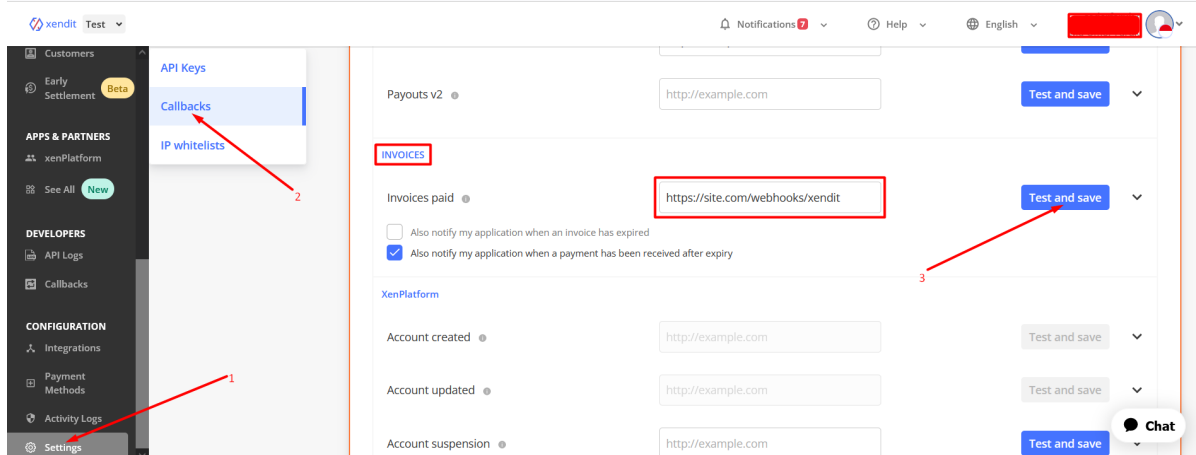
**SHOP\_URL=[YOUR\_SHOP\_URL]** This parameter is must have in `.env` file when Xendit is using. Otherwise the payment redirection will be broken.

## Xendit Webhook settings

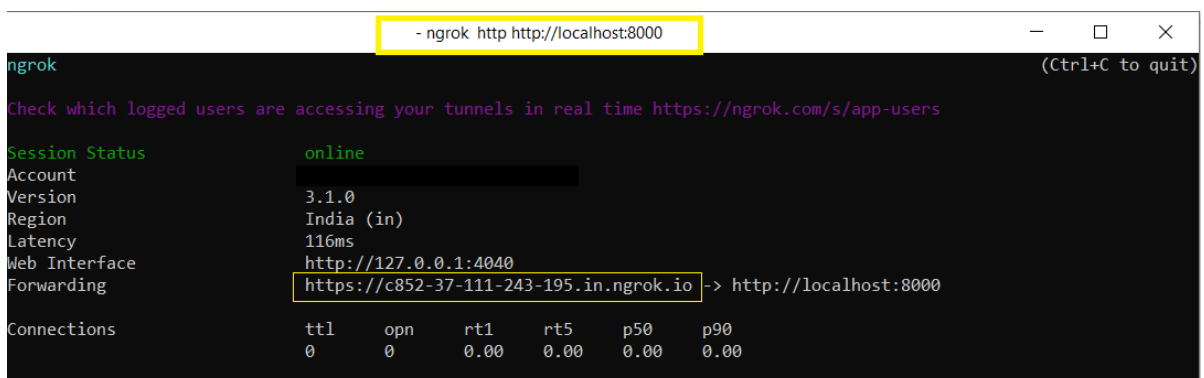
- To activate Xendit go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. Xendit Webhook URL is coming from local development. This static link will dynamically generated in live environment)



- Copy the webhook url & registered in Xendit Dashboard.



- If you want to use webhook during development on localhost, you must use a tools like ngrok to have the webhooks delivered to your local machine.



- Copy the forwarding https link & paste on Test Webhook URL, after paste forwarding https link, add line- /webhooks/xendit.

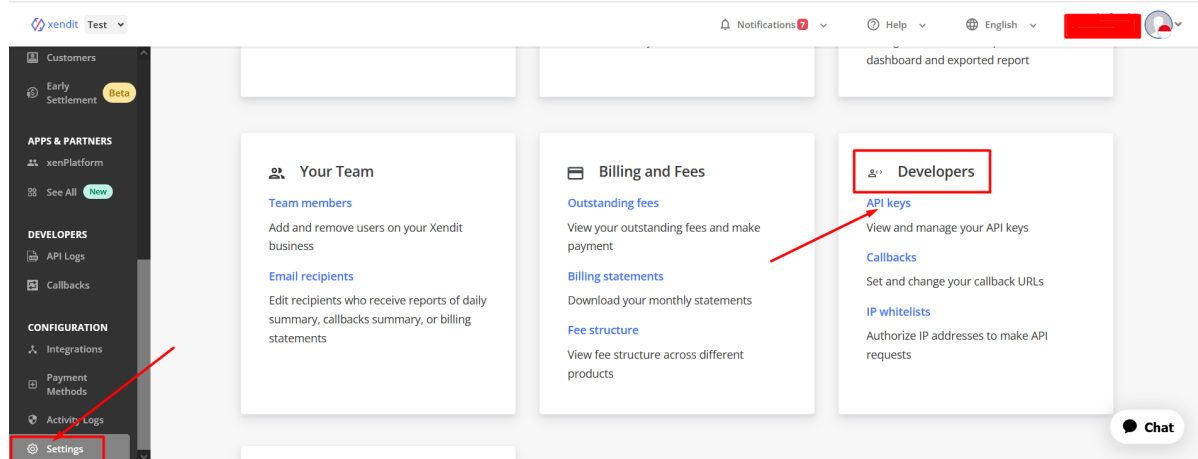
```

94 # Payment -> Xendit
95 API_KEY=
96 XENDIT_WEBHOOK_URL=https://45ae-2400-c600-3425-9e4-f4b6-285d-ab9d-70a1.in.ngrok.io/webhooks/xendit
97 XENDIT_CALLBACK_TOKEN=

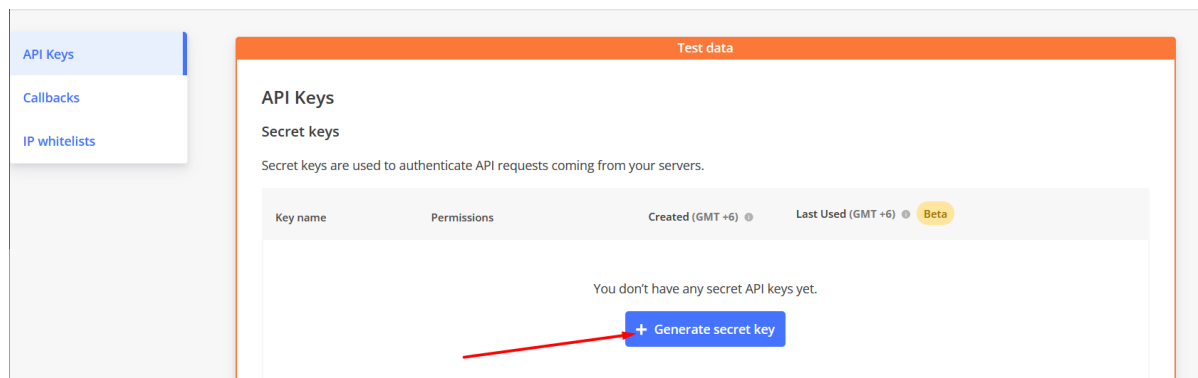
```

## How to create & setup Xendit information properly?

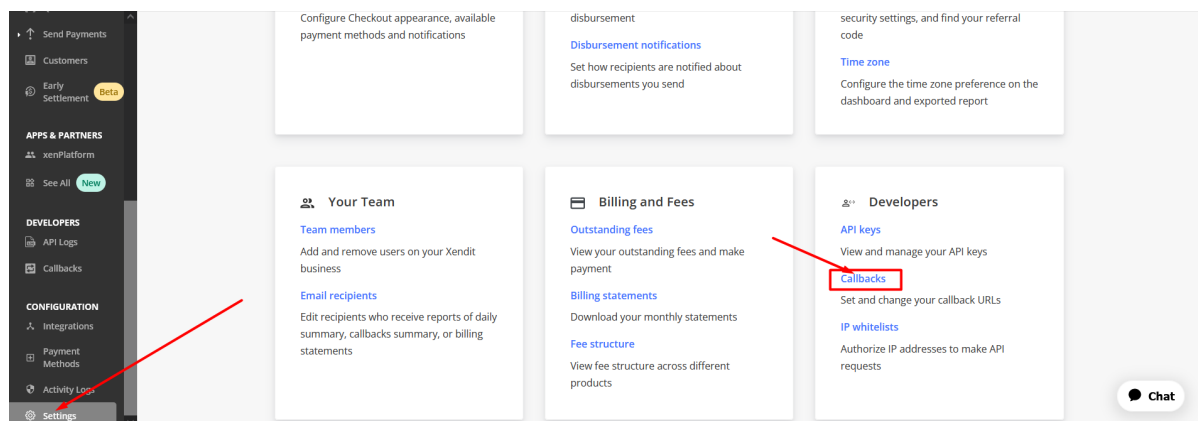
- Go to the [Xendit official website](#) & login. If you aren't a registered user, Complete the Xendit registration first.
- After logged in into Xendit dashboard, Click on Settings, after that click of API keys.



- After That Click on `Generate secret key` for new Xendit Secret Key.

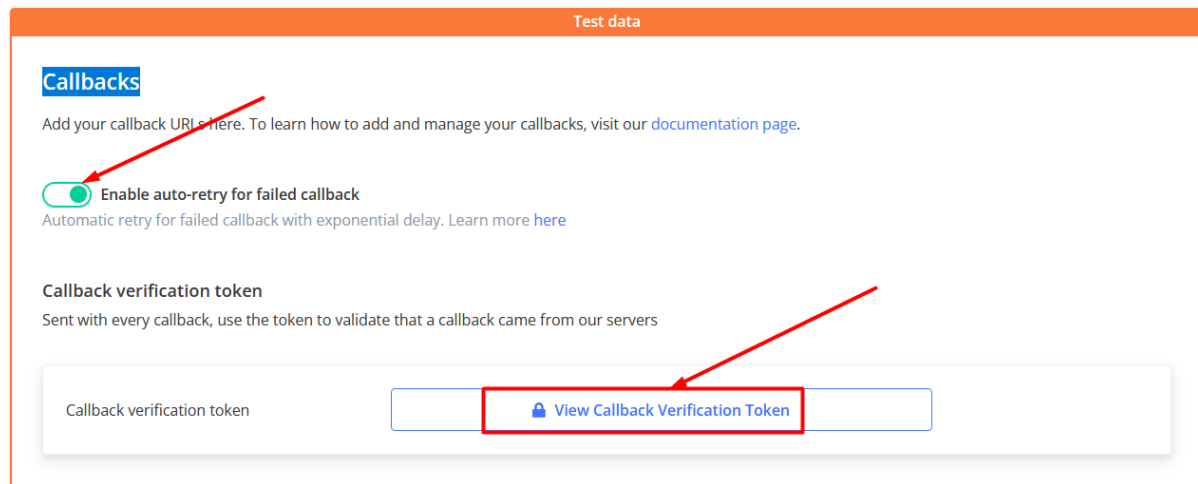


- Then API key name, give the permissions and click on Generate key button.
- Next, for Callback verification token, Go to settings and click on Callbacks.



Make sure you enable auto-retry for failed callback, by enabling this, you will have better experience in handling your callback response.

Next, is to obtain your callback verification token. Click View Callback Verification Token, Enter your password, Copy & paste those inside PickBazar API `.env` file like mentioned above.



## Special Notes for Xendit.

If we have used any third party system/plugin/packages, then we have always encouraged our respected customers to follow the official documentation for detailed & in-depth knowledge.

- [Xendit official documentation](#)
- [Xendit Webhooks documentation](#)

## SSLCOMMERZ

SSLCOMMERZ is the largest payment gateway aggregator in Bangladesh. In PickBazar we have integrated SSLCOMMERZ APIs which may cover a vast area of SSLCOMMERZ supported region. It will help your business to grow and reach to a wide region.

### SSLCOMMERZ integrate inside PickBazar.

Please follow & complete this steps for SSLCOMMERZ integration for your e-commerce system.

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in SSLCOMMERZ.

```
SHOP_URL=[YOUR_SHOP_URL]

# Values: true or false (Default: true)
SSLC_ALLOW_LOCALHOST=true

# SSLCOMMERZ Setting & API Credentials

SSLC_STORE_ID=[YOUR_SSLCOMMERZ_STORE_ID]
SSLC_STORE_PASSWORD=[YOUR_SSLCOMMERZ_STORE_PASSWORD]
```

**SHOP\_URL=[YOUR\_SHOP\_URL]** This parameter is must have in `.env` file when SSLCOMMERZ is using. Otherwise the payment redirection will be broken.

Package used Inside PickBazar [SSLCOMMERZ-Package](#)

- To activate SSL-Commerz go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. webhook URL is coming from local development. This static link will dynamically generated in live environment)

The screenshot displays the PickBazar admin dashboard. On the left is a sidebar menu with options: Create Order, Users, Coupons, Taxes, Shippings, Withdrawals, Message, Refunds, Questions, Store Notice, Reviews, and Settings. The 'Settings' option is highlighted with a red box and a red arrow. The main content area shows four summary cards: Total Revenue (\$0.00), Total Order (0), Todays Revenue (\$0.00), and Total Shops (7). Below these is a 'Sale History' chart. At the bottom, the 'Payment' section is expanded, showing 'Configure Payment Option'. An orange arrow points to the 'SslCommerz' gateway selection. The 'Enable Cash On Delivery' toggle is turned on. The 'Webhook URL' field contains the text 'http://localhost:8000/webhooks/sslcommerz'.

## How to create & setup SSLCOMMERZ information properly?

- For **local** Create an account in <https://developer.sslcommerz.com/registration/>
- For going **live** Create an account in <https://signup.sslcommerz.com/register>
- After successfully creating your sslcommerz account you will get your `SSLC_STORE_ID` and `SSLC_STORE_PASSWORD` through email.



## Welcome to SSLCommerz Sandbox

### Congratulations Sandbox Store ID Created

We highly appreciate if you first connect with the sandbox and then move to the live system. Please inform us if you need to change

Registered Store URL for textbox.



**Store ID:** [REDACTED]

**Store Password (API/Secret Key):**

[REDACTED]

---

**Merchant Panel URL:** <https://sandbox.sslcommerz.com/manage/>  
(Credential as you inputted in the time of registration)

---

**Store name:** [REDACTED]

**Registered URL:** [REDACTED]

**Session API to generate transaction:** <https://sandbox.sslcommerz.com/gwprocess/v3/api.php>

**Validation API:** <https://sandbox.sslcommerz.com/validator/api/validationserverAPI.php?wsdl>

**Validation API (Web Service) name:** <https://sandbox.sslcommerz.com/validator/api/validationserverAPI.php>

---

**You may check our plugins available for multiple carts and libraries:**  
<https://github.com/sslcommerz>

#### YOUR INFORMATION AT OUR SYSTEM

**Merchant Id:** [REDACTED]

**Merchant Name:** [REDACTED]

**Name:** [REDACTED]

**Mailing Address:** [REDACTED]

**Email:** [REDACTED]

**Mobile:** [REDACTED]

- MERCHANT PANEL - WEBSITE - DEVELOPER PAGE
- You will find the IPN setting under **My Stores**

#### IPN Settings

[Back](#)

##### IPN at HTTP Listener

☒ Enable HTTP Listener

HTTP POST Parameters will be throw to the IPN\_HTTP\_URL as amount=1150.00&bank\_tran\_id=151114130739MqCBN5&card\_brand=VISA&card\_issuer=BRAC+BRAC+LTD.&card\_issuer\_country=Bangladesh&card\_issuer\_country\_code=BD&card\_no=432149XXXXX0667&card\_type=VISA-Brac+bankocv=BDT&status=VALID&store\_amount=1104.00&store\_id=ci614b3ac58771e&tran\_date=2015-11-14+13%3A07%3A12&tran\_id=5846dd9d4b484&val\_id=151114130742Bj94IBUk4uE5GRJ&verify\_sign=9ae83ce8b4821937fa6b9de93a379983&verify\_key=amount%2Cbank\_tran\_id%2Ccard\_brand%2Ccard\_issuer%2Ccard\_issuer\_country%2Ccard\_issuer\_country\_code%2Ccard\_no%2Ccard\_type%2Ccurrency%2Cstatus%2Cstore\_amount%2Cstore\_id%2Ctran\_date%2Ctran\_id%2Cval\_id

[Save](#)

##### Requirement of your HTTP IPN Listener

- The Listener must use the common port like 80 or 443
- Your IPN Listener must be reachable from Internet
- White-list the SSLCommerz IPs at your network firewall
  - Sandbox Access Requirement
    - TCP 80 or 443 needs to be opened at your system from 103.26.139.87
    - Your system needs to be able to reach TCP 443 of 103.26.139.87
  - Live Access Requirement
    - TCP 80 or 443 needs to be opened at your system from 103.26.139.81
    - Your system needs to be able to reach TCP 443 of 103.26.139.148

2011-2023 © SSLCommerz - All rights reserved. It is powered by SSL Wireless.

- Then Add the IPN notification if you want to up & running the services.
  - VALID** : A successful transaction.
  - FAILED** : Transaction is declined by customer's Issuer Bank.
  - CANCELLED** : Transaction is cancelled by the customer.
  - UNATTEMPTED** : Customer did not choose to pay any channel.
  - EXPIRED** : Payment Timeout.
- At last, For going live with your application please follow this official documentation. <https://developer.sslcommerz.com/doc/v4/#grab-the-notification>
- For any kind of further customization you can also can see this documentation <https://github.com/dgvai/laravel-sslcommerz>

## RazorPay

Razorpay is the only payments solution in India that allows businesses to accept, process and disburse payments with its product suite. It gives you access to all payment modes including credit card, debit card, netbanking, UPI and popular wallets including JioMoney, Mobikwik, Airtel Money, FreeCharge, Ola Money and PayZapp.

## RazorPay integration inside PickBazar.

Please follow & complete this steps for RazorPay integration with your e-commerce system.

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in RazorPay.

```
RAZORPAY_KEY_ID=[YOUR_RAZORPAY_KEY_ID]
RAZORPAY_KEY_SECRET=[YOUR_RAZORPAY_KEY_SECRET]
RAZORPAY_WEBHOOK_SECRET_KEY=[YOUR_RAZORPAY__WEBHOOK_URL]
```

- Activate RazorPay from PickBazar admin dashboard. (e.g. webhook URL is coming from local development. This static link will dynamically generated in live environment)

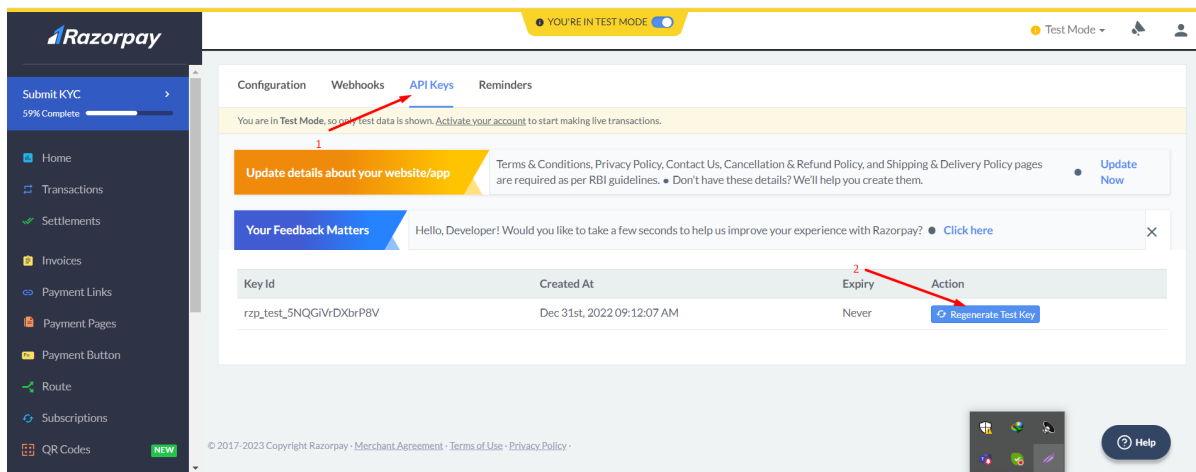
- Add RazorPay publishable key inside `.env` file of pickbazar shop.

```
NEXT_PUBLIC_RAZORPAY_KEY=[YOUR_RAZORPAY_KEY_ID]
```

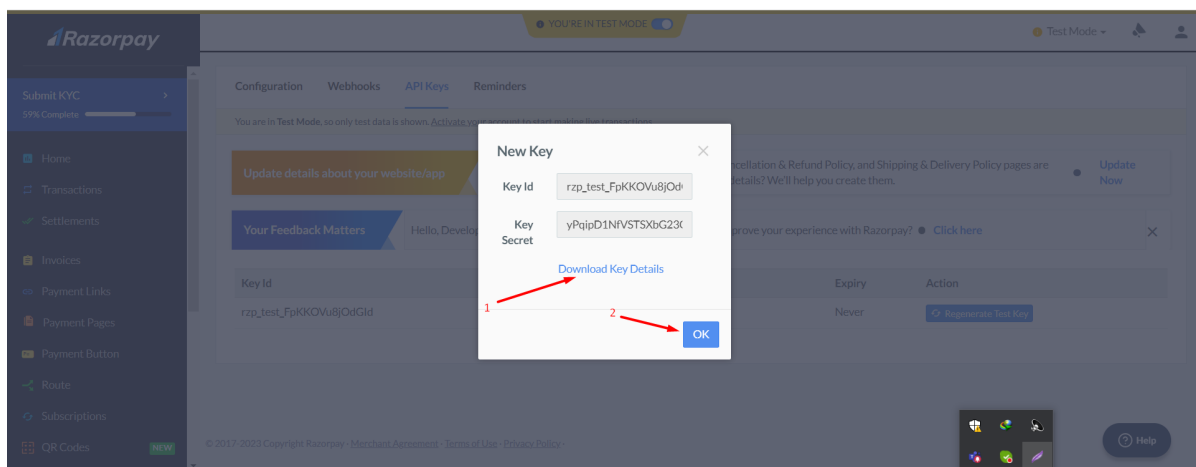
## How to create & setup RazorPay information properly?

- Go to the RazorPay website & login <https://razorpay.com> If you aren't a registered user, Complete the RazorPay registration first.
- After logged in into RazorPay dashboard, Click on Settings

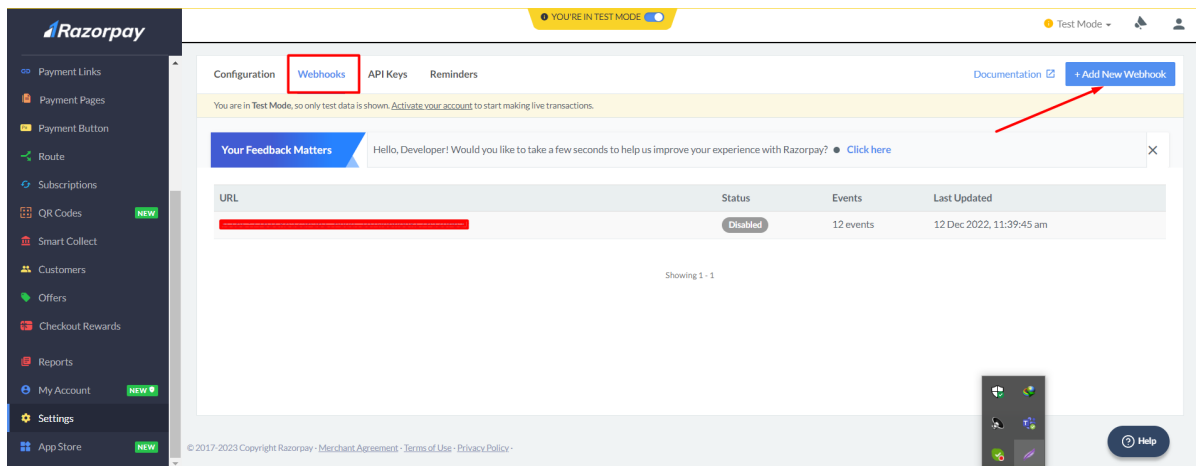
- Select API Keys option & click on Generate Test Key.



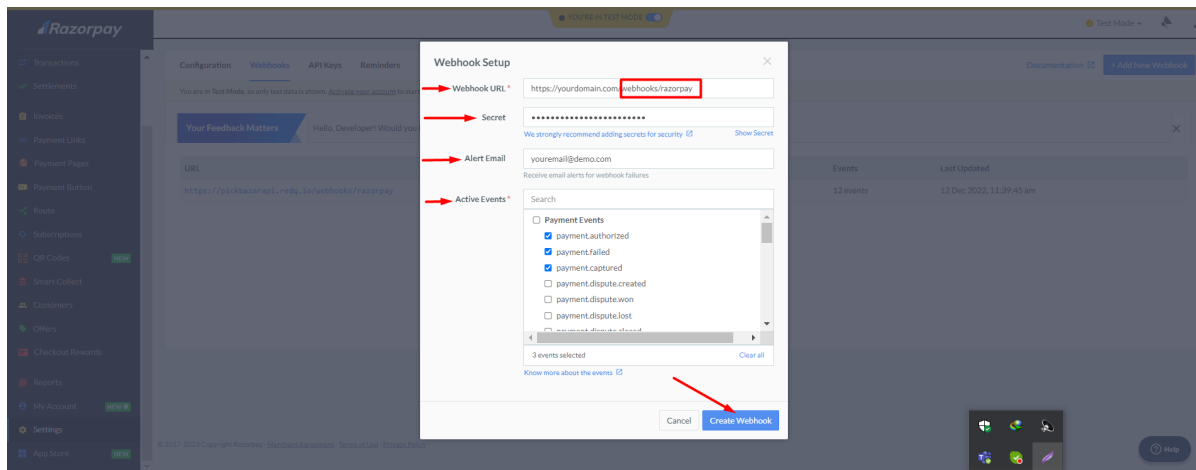
- The Client Id and Secret Key will appear. Download these keys and click on OK



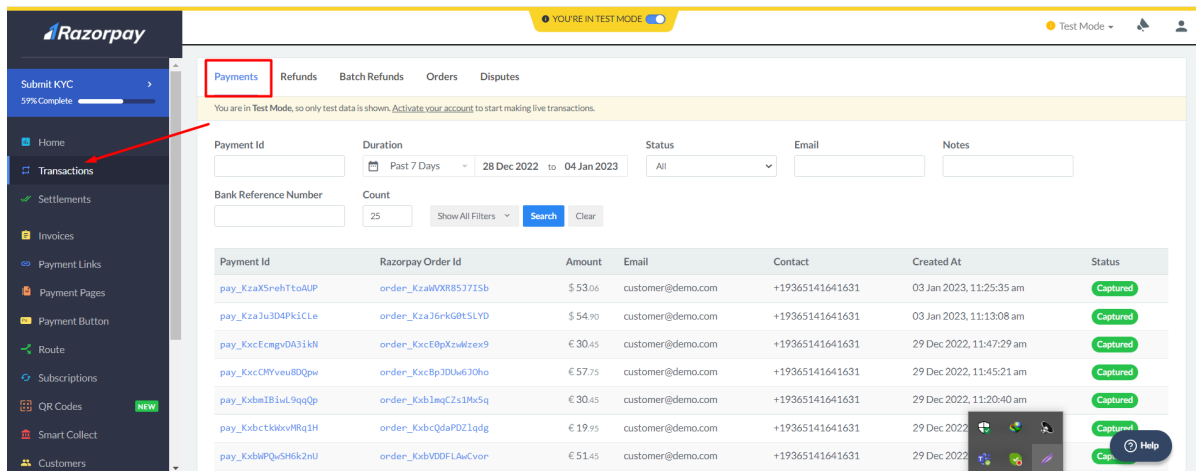
- Select Webhooks option & click on Add New Webhooks.



- Enter Webhook URL , Secret key , Alert Email and tick mark the checkbox in Active Event then click on create Webhook



- Copy & paste those inside PickBazar app .env file like mentioned above. after that test your RazorPay Payment Status



## Special Notes for RazorPay.

If we have used any third party system/plugin/packages, then we have always encouraged our respected customers to follow the official documentation for detailed & in-depth knowledge.

- RazorPay DOCS -- <https://razorpay.com/docs/#home-payments>
- RazorPay Webhooks -- <https://razorpay.com/docs/webhooks/setup-edit-payments>

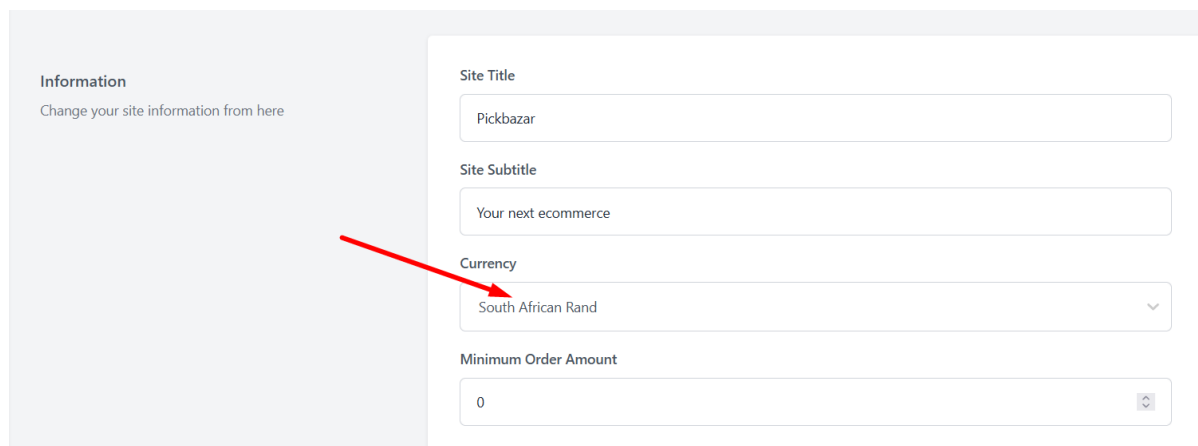
## Paystack

Paystack is a Nigerian fintech company that provides online payment solutions for businesses. It offers a range of services, including online and offline payments, subscriptions, and invoicing. Paystack's platform enables businesses to easily and securely accept payments from customers via credit card, debit card, and mobile money.

## Paystack integrate inside PickBazar.

Please follow & complete this steps for Paystack integration for your e-commerce system.

- First go to settings from PickBazar admin dashboard. Inside settings you will find Currency option. Select your Currency. If you want to Test development on localhost. then Select South African Rand(ZAR).



**Information**  
Change your site information from here

Site Title  
Pickbazar

Site Subtitle  
Your next ecommerce

Currency  
South African Rand

Minimum Order Amount  
0

Here is a breakdown of countries and the currencies they can accept payments in.

Country	Available Currencies
Ghana	GHS
Nigeria	NGN, USD
South Africa	ZAR

For more information: [Paystack currency is available](#).

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file.  
We will discuss later about how to create those API keys in [Paystack official website](#).

```
PAYSTACK_PUBLIC_KEY=[YOUR_PAYSTACK_PUBLIC_KEY]
PAYSTACK_SECRET_KEY=[YOUR_PAYSTACK_SECRET_KEY]
PAYSTACK_PAYMENT_URL=https://api.paystack.co
MERCHANT_EMAIL=[YOUR_PAYSTACK_MERCHANT_EMAIL]
```

- Add Paystack publishable key inside pickbazar shop.

```
NEXT_PAYSTACK_PUBLIC_KEY=[YOUR_YOUR_PAYSTACK_PUBLIC_KEY]
```

**SHOP\_URL=[YOUR\_SHOP\_URL]** This parameter is must have in `.env` file when Paystack is using.  
Otherwise the payment redirection will be broken.

## Paystack Webhook settings

- To activate Paystack go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. Paystack Webhook URL is coming from local development. This static link will dynamically generated in live environment)

**PickBazar** Create Shop

- Create Order
- Users
- Coupons
- Taxes
- Shippings
- Withdrawals
- Message
- Refunds
- Questions
- Store Notice
- Reviews
- Settings**

**Total Revenue**  
(Last 30 Days)  
\$0.00

**Total Order**  
(Last 30 Days)  
0

**Todays Revenue**  
\$0.00

**Total Shops**  
7

Sale History

2.0  
1.6  
1.2  
0.8  
0.4

**Payment**  
Configure Payment Option

☒ Enable Cash On Delivery

Select Payment Gateway

Paystack

Webhook URL

http://localhost:8000/webhooks/paystack

- Copy the webhook url & registered in Paystack Dashboard.

**Test Callback URL**

**Test Webhook URL**

**Save Changes**

- If you want to use webhook during development on localhost, you must use a tools like ngrok to have the webhooks delivered to your local machine.

```

- ngrok http http://localhost:8000
ngrok
Check which logged users are accessing your tunnels in real time https://ngrok.com/s/app-users
Session Status      online
Account              [REDACTED]
Version              3.1.0
Region               India (in)
Latency              116ms
Web Interface        http://127.0.0.1:4040
Forwarding            https://c852-37-111-243-195.in.ngrok.io -> http://localhost:8000
Connections          ttl    opn    rt1    rt5    p50    p90
                    0      0      0.00   0.00   0.00   0.00
  
```

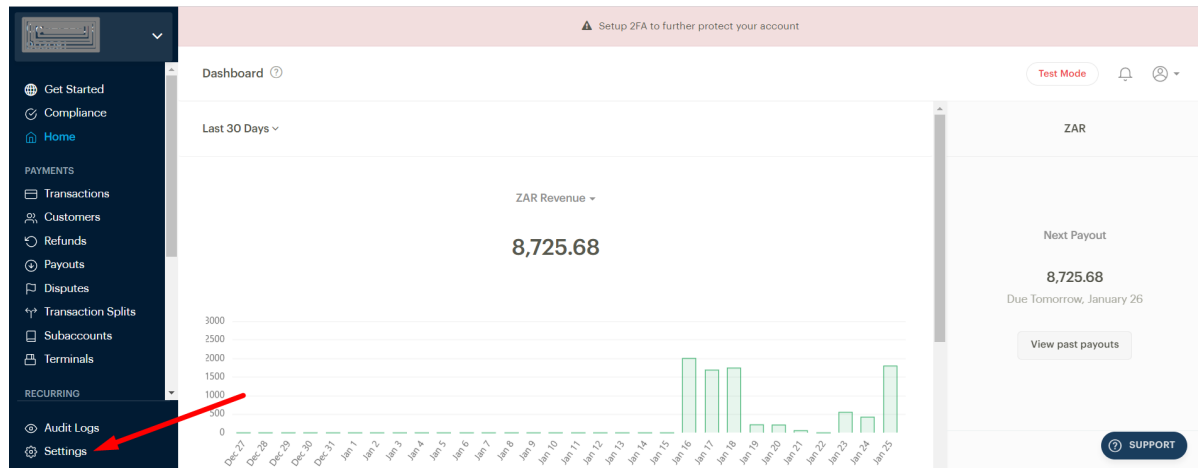
- Copy the forwarding https link & paste on Test Webhook URL, after paste forwarding https link, add line- /webhooks/paystack.

Test Callback URL

Test Webhook URL

## How to create & setup Paystack information properly?

- Go to the [Paystack official website](#) & login. If you aren't a registered user, Complete the Paystack registration first.
- After logged in into Paystack dashboard, Click on Settings.



- After That Click on API Keys & Webhooks for Paystack Secret Key & Paystack Public Key.

The screenshot shows the 'Settings' page with the 'API Keys & Webhooks' tab selected. It contains fields for 'Test Secret Key', 'Test Public Key', 'Test Callback URL', and 'Test Webhook URL'. A red arrow points to the 'Test Secret Key' field, and another red arrow points to the 'Test Public Key' field. A 'Generate new secret key' link is below the secret key field. A 'Save Changes' button is at the bottom right.

- Copy & paste those inside PickBazar API and PickBazar Shop `.env` file like mentioned above.

## Special Notes for Paystack.

If we have used any third party system/plugin/packages, then we have always encouraged our respected customers to follow the official documentation for detailed & in-depth knowledge.

- [Paystack official documentation](#)
- [Paystack Webhooks documentation](#)
- [Accept Payments in US Dollars \(USD\)](#)

# PayPal

PayPal is an American multinational financial technology company operating an online payments system. In PickBazar we have integrated PayPal APIs which may cover a vast area of PayPal supported region. It will help your business to grow and reach to a wide region.

## PayPal integrate inside PickBazar.

Please follow & complete this steps for PayPal integration for your e-commerce system.

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in PayPal.

```
# Values: sandbox or live (Default: live)
PAYPAL_MODE=sandbox

# Add currency like USD
PAYPAL_CURRENCY=USD

# Change this accordingly for your application.
PAYPAL_NOTIFY_URL=

# force gateway language i.e. it_IT, es_ES, en_US ... (for express checkout only)
PAYPAL_LOCALE=en

#Validate SSL when creating api client.
PAYPAL_VALIDATE_SSL=

# PayPal Setting & API Credentials -> sandbox

PAYPAL_SANDBOX_CLIENT_ID=[YOUR_PAYPAL_SANDBOX_CLIENT_ID]
PAYPAL_SANDBOX_CLIENT_SECRET=[YOUR_PAYPAL_SANDBOX_CLIENT_SECRET_KEY]

# PayPal Setting & API Credentials -> live

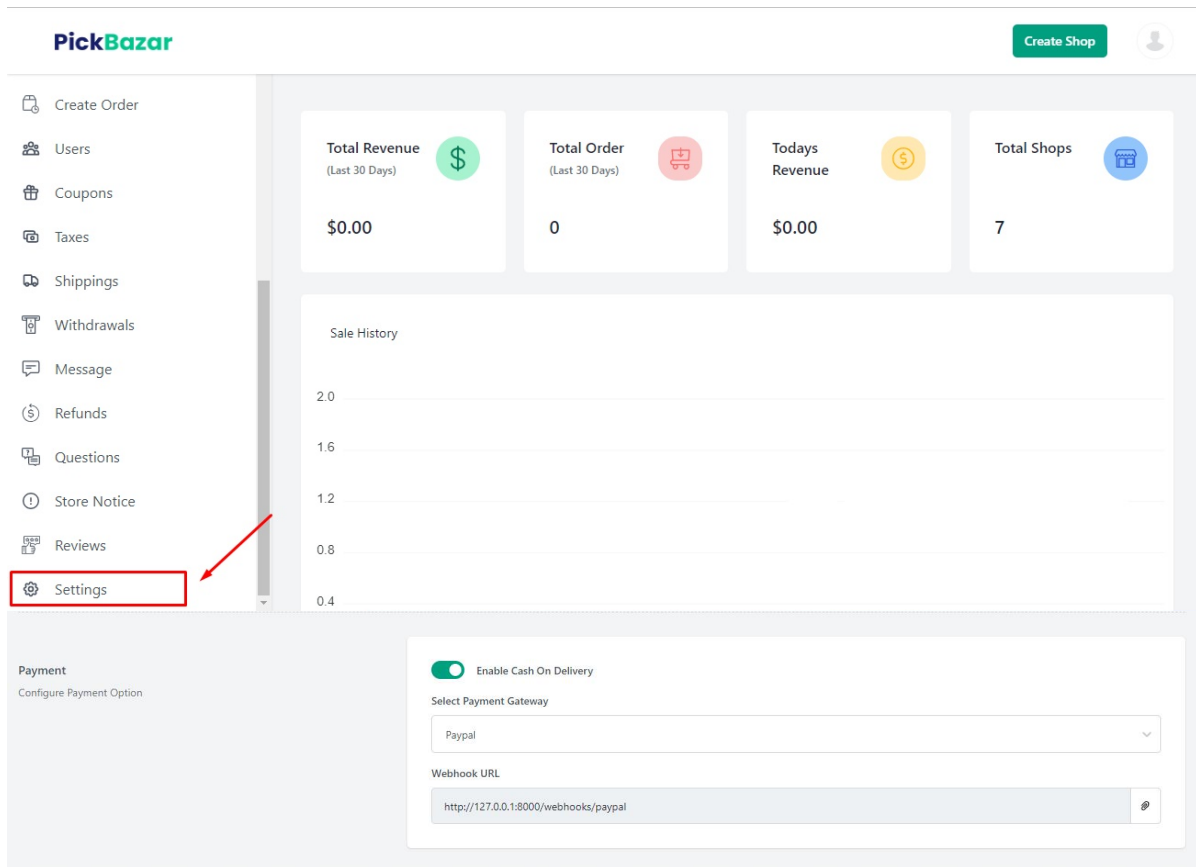
PAYPAL_LIVE_CLIENT_ID=[YOUR_PAYPAL_LIVE_CLIENT_ID]
PAYPAL_LIVE_CLIENT_SECRET=[YOUR_PAYPAL_LIVE_CLIENT_SECRET_KEY]

# PayPal Webhook settings

PAYPAL_WEBHOOK_ID=[YOUR_PAYPAL_WEBHOOK_URL]
SHOP_URL=[YOUR_SHOP_URL]
```

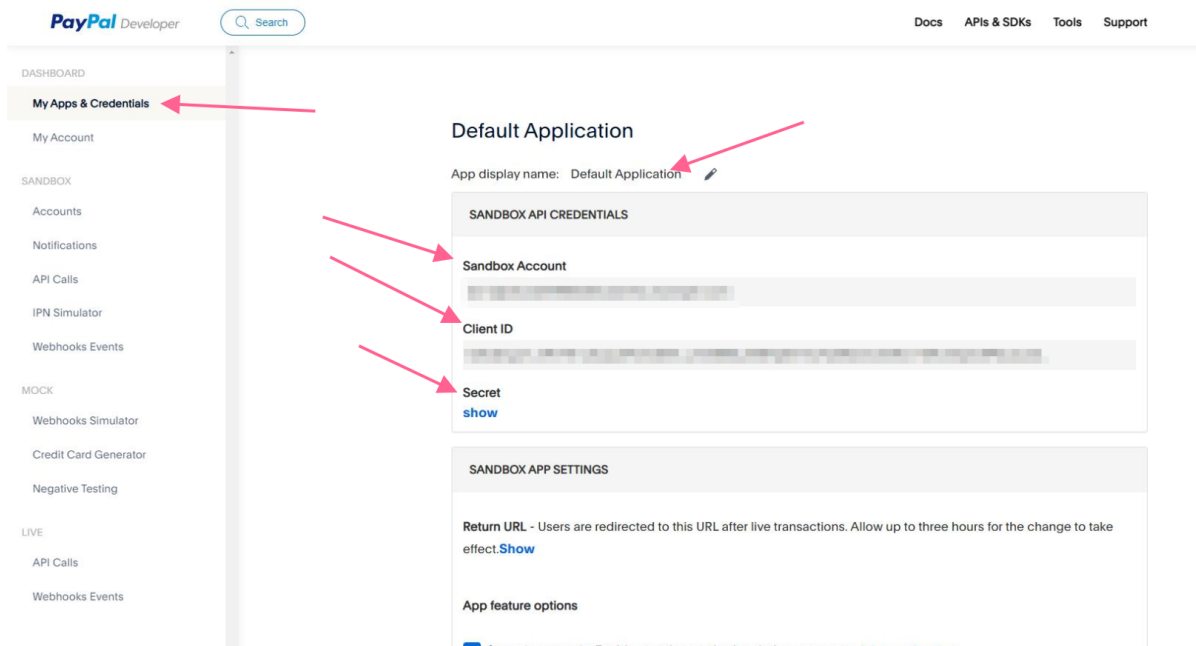
**SHOP\_URL=[YOUR\_SHOP\_URL]** This parameter is must have in `.env` file when PayPal is using. Otherwise the payment redirection will be broken.

- To activate Paypal go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. webhook URL is coming from local development. This static link will dynamically generated in live environment)

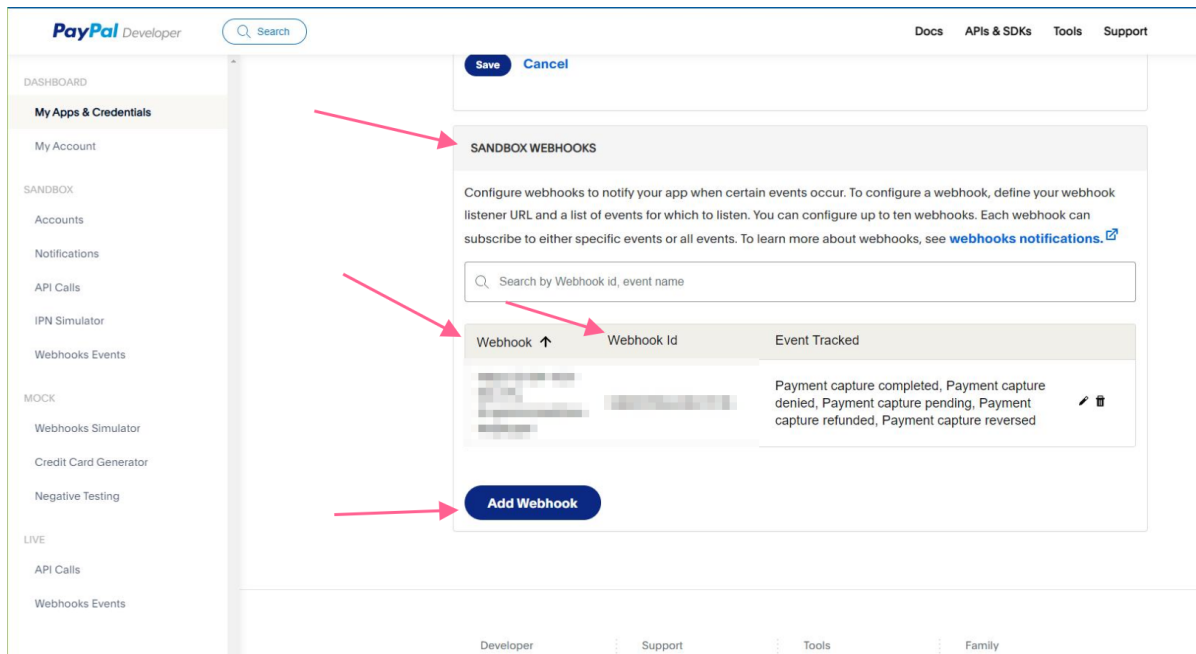


## How to create & setup PayPal information properly?

- Create an account in <https://developer.paypal.com>
- Choose sandbox if your are testing your development environment.
- Click on **Default Application** . or create a new App.
- You will get your Sandbox API credentials. Copy & paste those inside PickBazar app .env file like mentioned above.



- You will find the Webhook ID and link too.



- Then Add the webhook events if you want to up & running the services.
  - **PAYMENT.CAPTURE.COMPLETED**
  - **PAYMENT.CAPTURE.PENDING**
  - **PAYMENT.CAPTURE.CANCELLED**
  - **PAYMENT.CAPTURE.REVERSED**
- At last, For going live with your application please follow this official documentation. <https://developer.paypal.com/docs/archive/paypal-here/sdk-dev/going-live/>

## Mollie

Mollie is a Payment Service Provider (PSP) that processes online payments for companies. If you buy something online from one of our merchants, we make sure that your money is transferred safely from your bank to the merchant's bank. Since we arrange the payment process, you may see Mollie or Stg Mollie Payments on your bank statement.

## Mollie integrate inside PickBazar.

Please follow & complete this steps for Mollie integration for your e-commerce system.

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in Mollie.

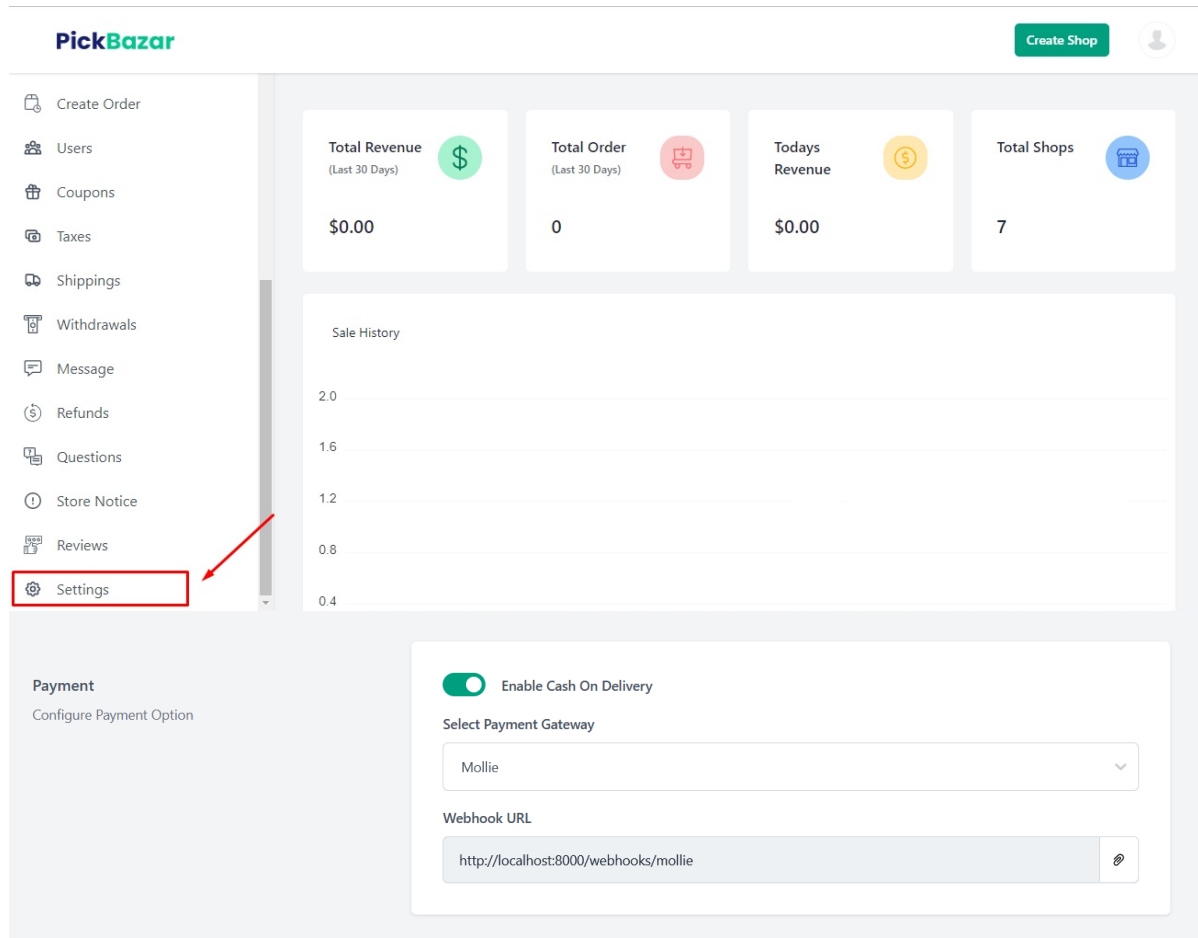
```
MOLLIE_KEY=[YOUR_MOLLIE_API_KEY]
```

## Mollie Webhook settings

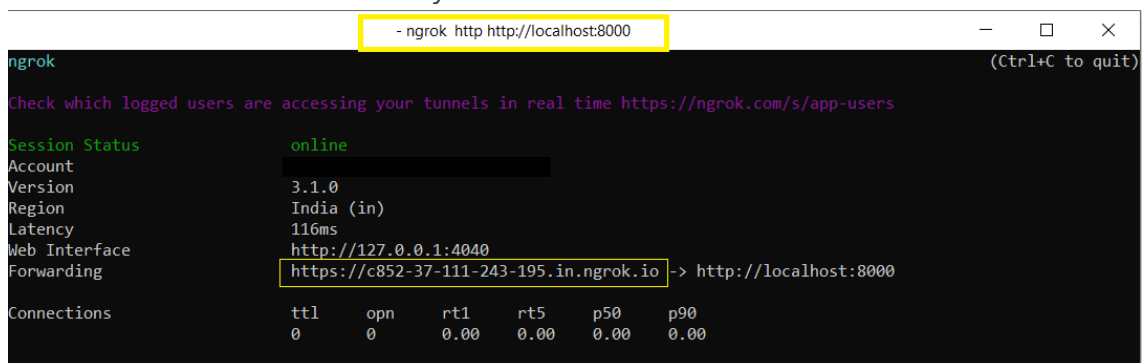
```
SHOP_URL=[YOUR_SHOP_URL]
MOLLIE_WEBHOOK_URL=[YOUR_MOLLIE_WEBHOOK_URL]
```

**SHOP\_URL=[YOUR\_SHOP\_URL]** This parameter is must have in .env file when Mollie is using. Otherwise the payment redirection will be broken.

- To activate Mollie go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. Mollie Webhook URL is coming from local development. This static link will dynamically generated in live environment)



- Copy & paste those inside PickBazar app .env file like mentioned above.
- If you want to use webhook during development on localhost, you must use a tool like ngrok to have the webhooks delivered to your local machine.

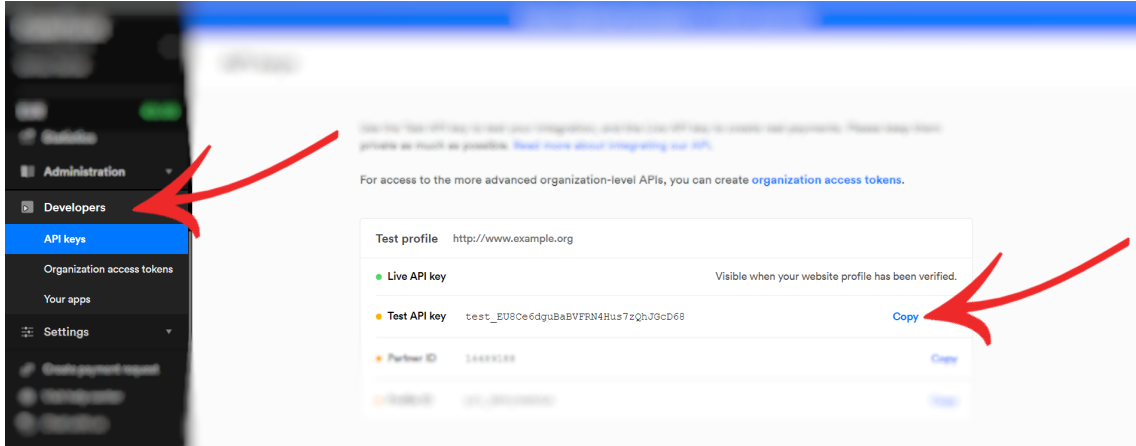


- Copy the forwarding https link & paste .env file after link add line- /webhooks/mollie

```
84 # Payment -> Mollie
85 MOLLIE_KEY=test_EU8Ce6dguBaBVFRN4Hus7zQhJGcD68
86 MOLLIE_WEBHOOK_URL=https://c852-37-111-243-195.in.ngrok.io/webhooks/mollie
```

## How to create & setup Mollie information properly?

- Go to the Mollie website & login <https://www.mollie.com> If you aren't a registered user, the complete the Mollie registration first.
- After logged in into Mollie dashboard, follow the developer link to copy the API keys.
- Choose Test API key your are testing your development environment.
- Copy & paste those inside PickBazar app .env file like mentioned above.



## Special Notes for Mollie.

If we have used any third party system/plugin/packages, then we have always encouraged our respected customers to follow the official documentation for detailed & in-depth knowledge.

- Mollie DOCS -- <https://docs.mollie.com/>
- Mollie Webhooks -- <https://docs.mollie.com/overview/webhooks>

## Iyzico

Iyzico is one of Turkey's most well-known payment gateways. Iyzico provides safe payment solutions to internet businesses and individuals of various sizes. You can quickly and easily incorporate the Iyzico payment into your e-commerce website and begin accepting payments.

## Iyzico integrate inside PickBazar.

Please follow & complete this steps for Iyzico integration for your e-commerce system.

- Inside api there is .env file. You have to copy & paste this line of codes inside the .env file. We will discuss later about how to create those API keys in Iyzico.

```
# Remove sandbox- for Production

IYZIPAY_BASE_URL=[https://sandbox-api.iyzipay.com]

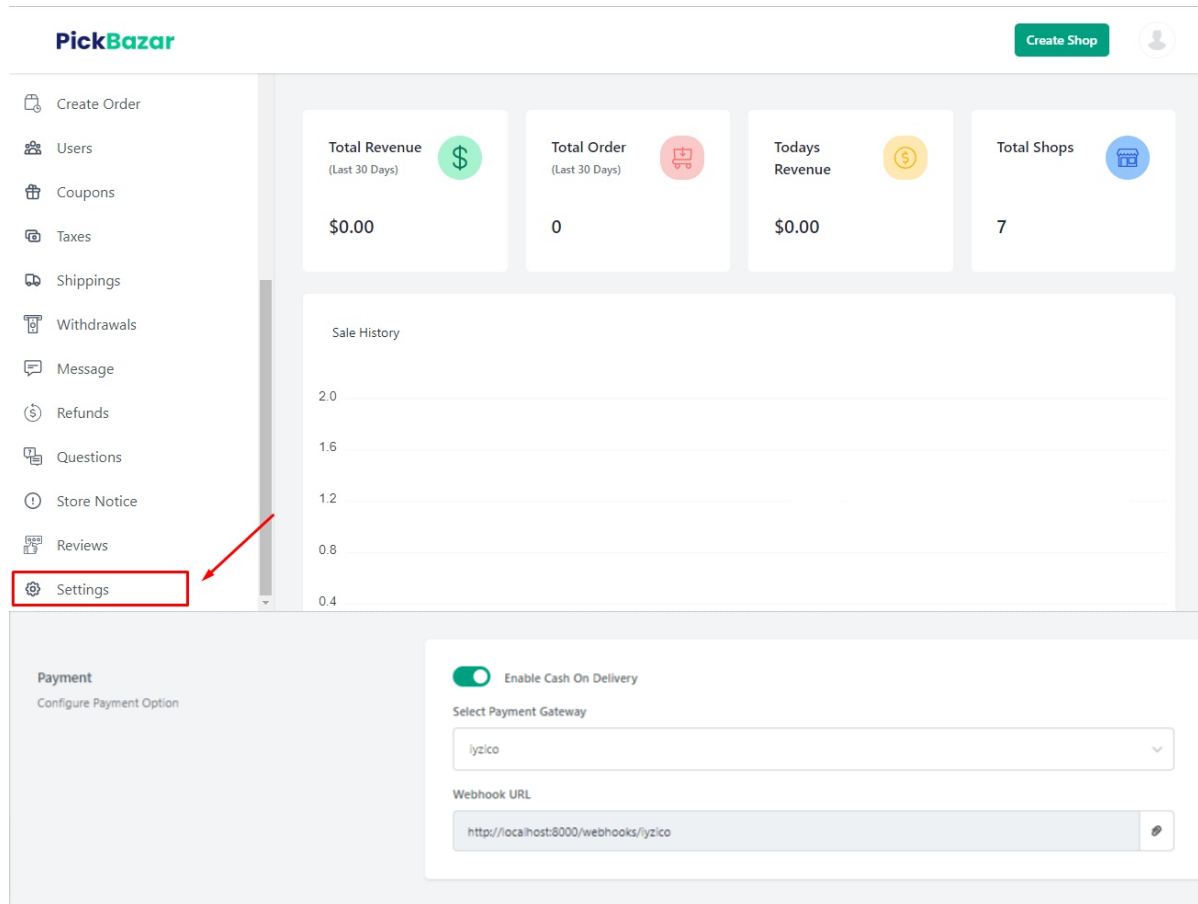
IYZIPAY_API_KEY=[YOUR_KEY_HERE]

IYZIPAY_SECRET_KEY=[YOUR_SECRET_KEY_HERE]

SHOP_URL=[YOUR_SHOP_URL]
```

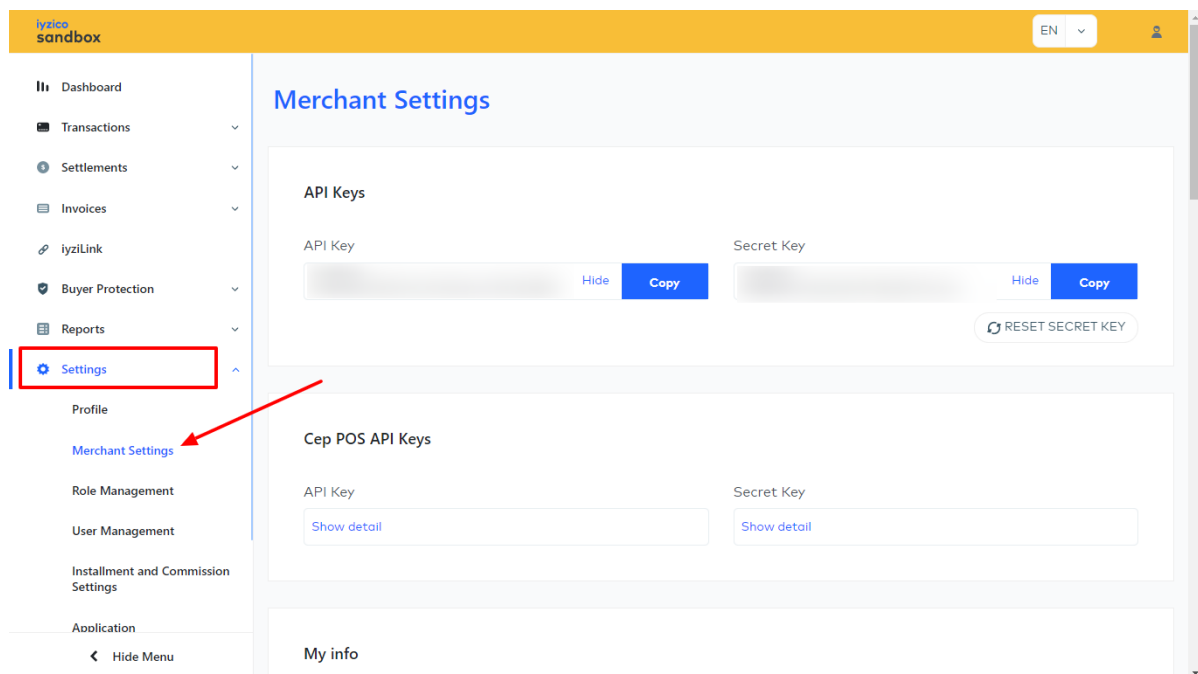
**SHOP\_URL=[YOUR\_SHOP\_URL]** This parameter is must have in .env file when lyzico is using. Otherwise the payment redirection will be broken.

- To activate Paypal go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. webhook URL is coming from local development. This static link will dynamically generated in live environment)

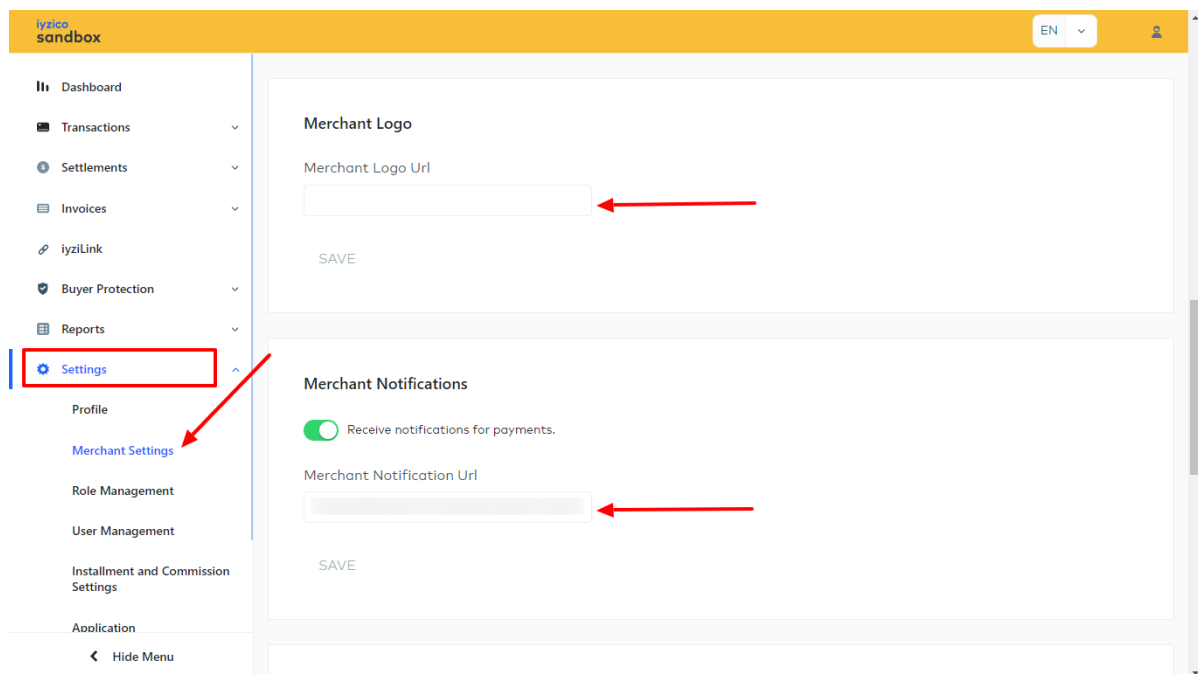


## How to create & setup lyzico information properly?

- Create an account in <https://merchant.iyzipay.com/>
- Create a sandbox account in <https://sandbox-merchant.iyzipay.com/>
- Choose sandbox if your are testing your development environment.
- Click on **Setting** then **Merchant Settings** . .
- You will get your Sandbox API credentials. Copy & paste those inside PickBazar app .env file like mentioned above.



- You will find the Webhook ID and logo link too.



- At last, For going live with your application please follow this official documentation. <https://dev.iyzipay.com/en/api>

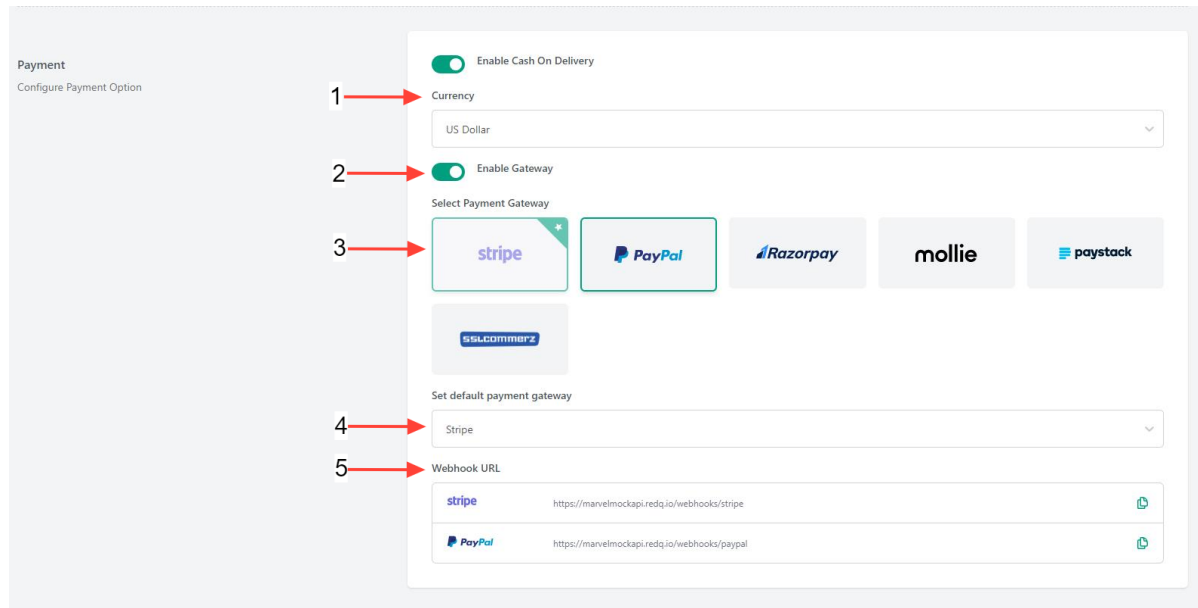
## Multiple payment gateway

In this chapter we are going to discuss about how you can use multiple payment gateway in our application. But before jumping into this feature user needs to keep some points in mind first.

1. Choose your desired payment gateway based on availability in your region.
2. Choose your currency correctly before activating that payment gateway.
3. Set up all the necessary .env keys and payment gateway internal settings properly before starting operations.
4. Configure webhooks properly if needed.

## How to activate multiple payment gateways [ in Admin ] ?

In admin dashboard there are new settings introduced for this feature. Please check this screenshot first,



## Description [ sequence by number in the screenshot ]

1. Choose the currency first
2. Enable gateway if you want to use any
3. Select the payment gateway based on your region and supported currency.
4. Select default payment gateway. Default gateway only available from selected ones.
5. Webhooks URL will be generated. You can use this URL for webhooks setting in payment gateways.

## How to use multiple payment gateways from customer end [ in Shop ] ?

Payment gateways activated by super-admin will be available in the checkout process. There are two steps of operation,

1. Select a payment gateway during place order
2. Select different payment gateway ( if needed ), after the place order.

This two points are cleared below with necessary screenshot.

## Select a payment gateway during place order

Please check this screenshot first,

The screenshot shows the PickBazar checkout process. On the left, there are four steps: 1. Contact Number, 2. Billing Address, 3. Shipping Address, and 4. Delivery Schedule. On the right, there is a summary of the order items and their prices, followed by a 'Choose Payment Method' dialog box. The dialog box is highlighted with a red rectangle and a red arrow pointing to it. The dialog box contains three options: stripe, PayPal, and Cash On Delivery. Below the options, there is a note: 'Please click Place order to make order and payment'. At the bottom of the dialog box is a green button labeled 'Place Order'.

Select a payment gateway just before placing order.

**\*\* Notes for customer : \*\*** Please choose this carefully. Otherwise you may need to re-select the payment procedure again.

## Select different payment gateway ( if needed )

In order details page there is a option to re-select the payment gateway. If any customer select a gateway and place order but found out any issue of his/her selected gateway, then he/she will have a opportunity to re-select the payment gateway again and complete the order process.

Please check the screenshot below carefully,

The screenshot shows the PickBazar order details page. The order status is 'Order Pending'. There are buttons for 'Pay Now' and 'Change gateway'. A modal dialog box titled 'Choose Another Payment Gateway' is open, showing two options: stripe and PayPal. The dialog box is highlighted with a red rectangle and a red arrow pointing to it. The dialog box also has a green button labeled 'Submit Payment'. The background shows the order details, including the order number, date, total, and payment method.

## Description [ sequence by number in the screenshot ]

1. Stripe is the previous selected payment gateway. Which do not intend to use now. So he/she might change the gateway and submit the payment.
2. PayPal was selected payment gateway now.
3. Submit payment button need to be clicked, otherwise the operation will not be continued.

## Integration of new payment gateway

To integrate a payment gateway in PickBazar-Laravel, you will need to follow these general steps:

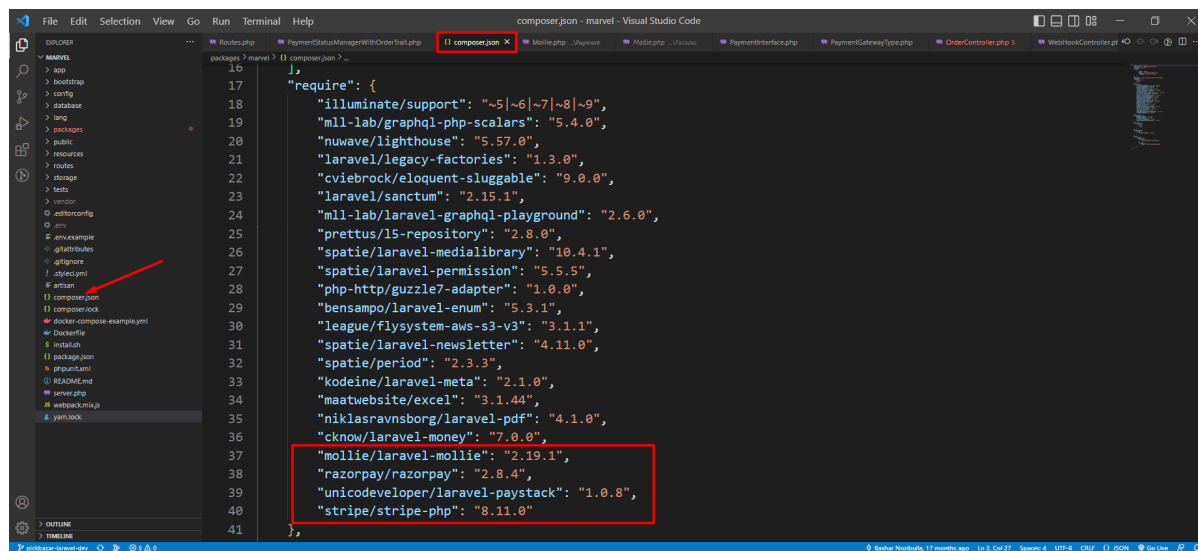
### Getting Started with API

#### Step 1: Install and configure the payment gateway package

First, you will need to install the payment gateway package for Laravel (if there are any available package). There are several packages available that provide integration with various payment gateways. Example is given in the screenshot.

Once you have chosen and installed a package, you will need to configure it by adding your payment gateway credentials and any other required settings.

To check the installed dependencies of payment gateway in PickBazar-Laravel, you can open the composer.json file in a text editor and find that.



```
16  },
17  "require": {
18      "illuminate/support": "~5|~6|~7|~8|~9",
19      "mll-lab/graphql-php-scalars": "5.4.0",
20      "nuwave/lighthouse": "5.57.0",
21      "laravel/legacy-factories": "1.3.0",
22      "cviebrock/eloquent-sluggable": "9.0.0",
23      "laravel/sanctum": "2.15.1",
24      "mll-lab/laravel-graphql-playground": "2.6.0",
25      "pretorius/l5-repository": "2.8.0",
26      "spatie/laravel-medialibrary": "10.4.1",
27      "spatie/laravel-permission": "5.5.5",
28      "php-http/guzzle7-adapter": "1.0.0",
29      "bensampo/laravel-enum": "5.3.1",
30      "league/flysystem-aws-s3-v3": "3.1.1",
31      "spatie/laravel-newsletter": "4.11.0",
32      "spatie/period": "2.3.3",
33      "kodeine/laravel-meta": "2.1.0",
34      "maatwebsite/excel": "3.1.44",
35      "niklasravnsborg/laravel-pdf": "4.1.0",
36      "cknow/laravel-money": "7.0.0",
37      "mollie/laravel-mollie": "2.19.1",
38      "razorpay/razorpay": "2.8.4",
39      "unicodeveloper/laravel-paystack": "1.0.8",
40      "stripe/stripe-php": "8.11.0"
41  },
```

#### Step 2: Add payment gateway name in the Enum.

Add PaymentGateway Enum API -> package -> marvel -> src -> Enums ->

PaymentGatewayType.php

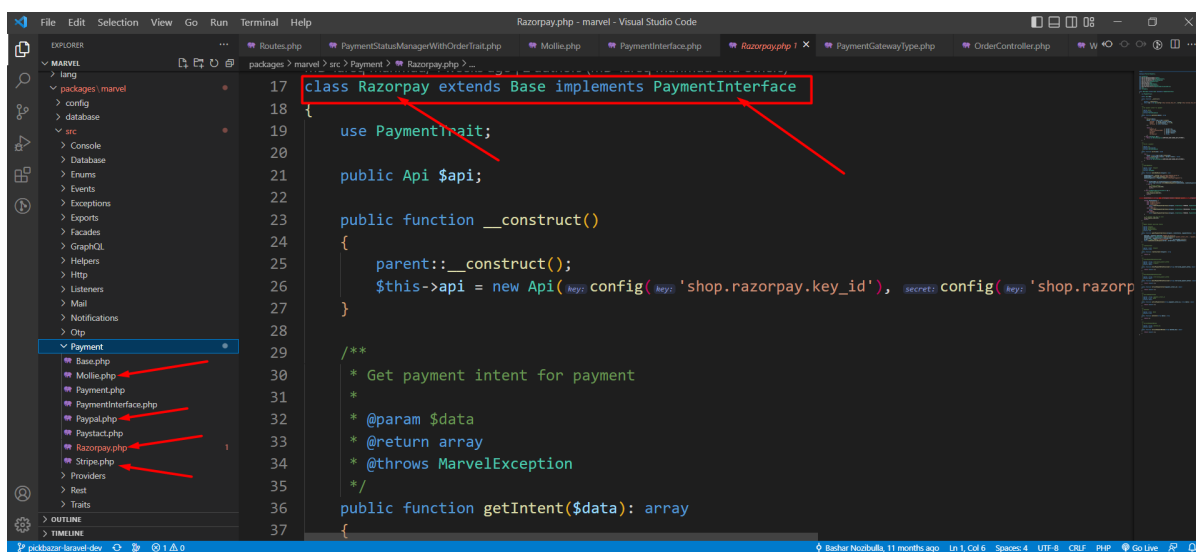
```

12 final class PaymentGatewayType extends Enum
13 {
14     public const STRIPE = 'STRIPE';
15     public const CASH_ON_DELIVERY = 'CASH_ON_DELIVERY';
16     public const MOLLIE = 'MOLLIE';
17     public const CASH = 'CASH';
18     public const RAZORPAY = 'RAZORPAY';
19     public const FULL_WALLET_PAYMENT = 'FULL_WALLET_PAYMENT';
20     public const PAYPAL = 'PAYPAL';
21 }

```

### Step 3: Configure the Payment Facade for the new payment gateway.

Now go to `API -> package -> marvel -> src -> Payment` then create new payment Class (e.g. Stripe, PayPal, Razorpay, Mollie) for implements the PaymentInterface. The Class must implements all of the methods defined in the PaymentInterface.

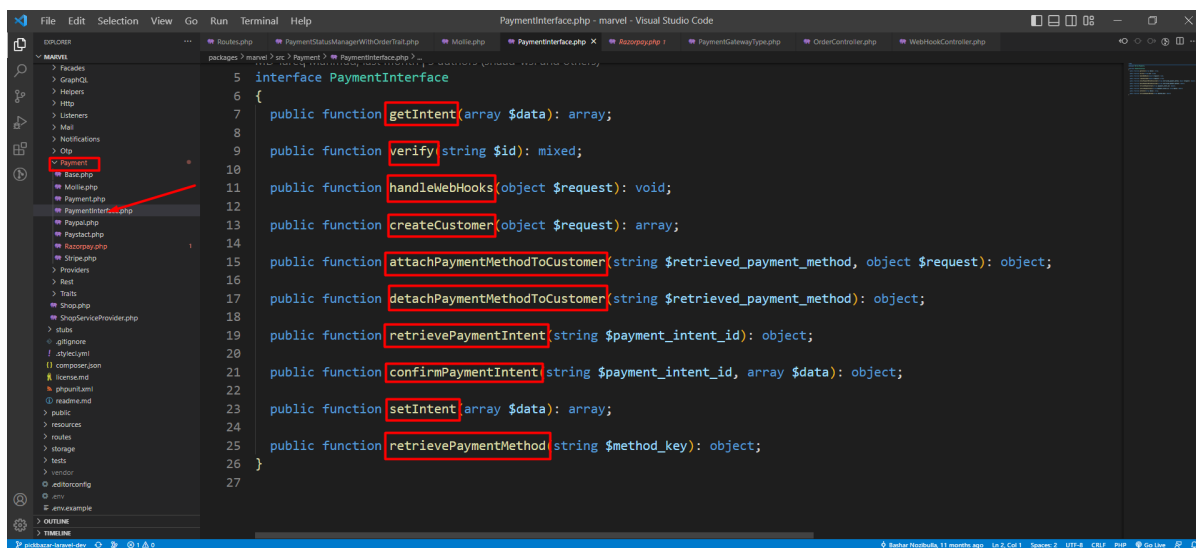


```

17 class Razorpay extends Base implements PaymentInterface
18 {
19     use PaymentInterface;
20
21     public Api $api;
22
23     public function __construct()
24     {
25         parent::__construct();
26         $this->api = new Api(
27             config('shop.razorpay.key_id'),
28             config('shop.razorpay.secret')
29         );
30
31         /**
32          * Get payment intent for payment
33          *
34          * @param $data
35          * @return array
36          * @throws MarvelException
37          */
38     public function getIntent($data): array
39     {
40
41     }
42 }

```

Methods defined in the PaymentInterface



```

5 interface PaymentInterface
6 {
7     public function getIntent(array $data): array;
8
9     public function verify(string $id): mixed;
10
11     public function handleWebHooks(object $request): void;
12
13     public function createCustomer(object $request): array;
14
15     public function attachPaymentMethodToCustomer(string $retrieved_payment_method, object $request): object;
16
17     public function detachPaymentMethodToCustomer(string $retrieved_payment_method): object;
18
19     public function retrievePaymentIntent(string $payment_intent_id): object;
20
21     public function confirmPaymentIntent(string $payment_intent_id, array $data): object;
22
23     public function setIntent(array $data): array;
24
25     public function retrievePaymentMethod(string $method_key): object;
26 }
27

```

Here's an example of a class that implements that PaymentInterface:

```

1 <?php
2
3 namespace Marvel\Payments;
4
5 use Exception;
6 use Marvel\Database\Models\Order;
7 use Marvel\Database\Models\PaymentIntent;
8 use Marvel\Exceptions\MarvelException;
9 use Marvel\Traits\PaymentTrait;
10 use Razorpay\Api\Api;
11 use Marvel\Enums\OrderStatus;
12 use Marvel\Enums\PaymentStatus;
13 use Razorpay\Api\Errors\SignatureVerificationError;
14 use Str;
15 use Throwable;
16
17 class Razorpay extends Base implements PaymentInterface
18 {
19     use PaymentTrait;
20
21     public Api $api;
22
23     public function __construct()
24     {
25         parent::__construct();
26         $this->api = new Api(config('shop.razorpay.key_id'), config('shop.razorpay.key_secret'));
27     }
28
29     /**
30      * Get payment intent for payment
31      *
32      * @param $data
33      * @return array
34      * @throws MarvelException
35      */
36     public function getIntent($data): array
37     {
38         try {
39             extract($data);
40             $order = $this->api->order->create([
41                 'receipt' => $order_tracking_number,
42                 'amount' => round($amount, 2) * 100,
43                 'currency' => $this->currency,
44             ]);
45
46             return [
47                 'payment_id' => $order->id,
48                 'order_tracking_number' => $order->receipt,
49                 'currency' => $order->currency,
50                 'amount' => $order->amount,
51                 'is_redirect' => false,
52             ];
53         } catch (Exception $e) {
54             throw new MarvelException(SOMETHING_WENT_WRONG_WITH_PAYMENT);
55         }
56     }
57
58     /**
59      * Verify a payment
60      *
61      * @param $id
62      * @return false|mixed
63      * @throws MarvelException
64      */
65     public function verify($id): mixed
66     {
67         try {
68             $order = $this->api->order->fetch($id);
69             return isset($order->status) ? $order->status : false;
70         } catch (Exception $e) {
71             throw new MarvelException(SOMETHING_WENT_WRONG_WITH_PAYMENT);
72         }
73     }
74
75     /**
76      * handleWebHooks
77      *
78      * @param mixed $request
79      * @return void
80      * @throws Throwable
81      */
82     public function handleWebHooks($request): void
83     {
84         $webhookSecret = config('shop.razorpay.webhook_secret');
85         $webhookBody = @file_get_contents('php://input');
86         $webhookSignature = $request->header('X-Razorpay-Signature');
87
88         try {
89             if ($webhookBody && $webhookSignature && $webhookSecret) {
90                 $this->api->utility->verifyWebhookSignature($webhookBody, $webhookSignature, $webhookSecret);
91             } else {
92                 // Invalid request
93                 http_response_code(400);
94                 exit();
95             }
96         } catch (SignatureVerificationError $e) {
97             // Invalid signature
98             http_response_code(400);
99             exit();

```

```

100     }
101
102     $eventStatus = (string) Str::of($request->event)->replace('payment.', '', $request->event);
103
104     switch ($eventStatus) {
105         case 'dispute.won':
106             case 'dispute.created':
107             case 'authorized':
108                 $this->updatePaymentOrderStatus($request, OrderStatus::PENDING, PaymentStatus::PROCESSING);
109                 break;
110             case 'captured':
111                 $this->updatePaymentOrderStatus($request, OrderStatus::PROCESSING, PaymentStatus::SUCCESS);
112                 break;
113             case 'failed':
114
115     }

```

API -> package -> marvel -> src -> Http -> Controllers -> OrderController.php -

```

> submitPayment
117 // to prevent loop for any case
118 http_response_code(200);
119 exit();
120
121 }

```

packages > marvel > src > Http > Controllers > OrderController.php > submitPayment

```

359  * @throws Exception
360  */
361  public function submitPayment(Request $request): void
362  {
363      $tracking_number = $request->tracking_number ?? null;
364      try {
365          $order = $this->repository->with($relations: ['products', 'children.shop', 'wallet_point', 'payment_intent'])
366              ->findOneByFieldOrFail($field: 'tracking_number', $value: $tracking_number);
367
368          switch ($order->payment_gateway) {
369
370              Gateway Type
371              case PaymentGatewayType::STRIPE:
372                  $this->stripe($order, $request, $settings: $this->settings);
373                  break;
374
375              case PaymentGatewayType::PAYPAL:
376                  $this->paypal($order, $request, $settings: $this->settings);
377                  break;
378
379              case PaymentGatewayType::MOLLIE:
380                  $this->mollie($order, $request, $settings: $this->settings);
381                  break;
382
383              case PaymentGatewayType::RAZORPAY:
384                  $this->razorpay($order, $request, $settings: $this->settings);
385                  break;
386
387          }
388      } catch (Exception $e) {
389          http_response_code(500);
390          return [];
391      }
392  }

```

API -> package -> marvel -> src -> Traits -> PaymentStatusManagerWithOrderTrait.php

```

152  * @param string $retrieved_payment_method
153  * @param object $request
154
155  */
156  public function razorpay(Order $order, Request $request, Settings $settings): void
157  {
158      try {
159          $chosen_intent = '';
160          // for single gateway options
161          if (isset($order->payment_intent)) {
162              foreach ($order->payment_intent as $key => $intent) {
163                  if (strtoupper($settings->options['payment_gateway']) === $order->payment_gateway) {
164                      $chosen_intent = $intent;
165                  }
166              }
167          }
168
169          $paymentId = isset($chosen_intent->payment_intent_info) ? $chosen_intent->payment_intent_info['payment_id'] : null;
170          if (isset($paymentId)) {
171              $paymentStatus = Payment::verify($paymentId);
172              if ($paymentStatus) {
173                  switch (strtolower($paymentStatus)) {
174                      case "paid":
175                          $this->paymentSuccess($order);
176                          break;
177                      case "attempted":
178                          $this->paymentProcessing($order);
179                          break;
180                      case "failed":
181                          $this->paymentFailed($order);
182                      }
183              }
184          }
185      } catch (Exception $e) {
186          throw new Exception('SOMETHING WENT WRONG WITH PAYMENT');
187      }
188  }

```

Step 5: Setup webhook in pickbazar-laravel follow the steps

To create a payment gateway webhook, you would first configure the webhook URL in your payment gateway account. Then, whenever the specified events occur, the payment gateway will send an HTTP POST request to the webhook URL with a payload of data about the event.

API -> package -> marvel -> src -> Routes

```

181  * @param string $payment_intent_id
182  * @param array $data
183  * @return object
184  */
185  public function confirmPaymentIntent(string $payment_intent_id, array $data): object
186  {
187      return (object) [];
188  }
189
190  /**
191   * setIntent
192   */
193  * @param array $data
194  * @return array
195  */
196  public function setIntent(array $data): array
197  {
198      return [];
199  }
200

```

```

73 Route::post(uri: 'webhooks/razorpay', action: [WebHookController::class, 'razorpay']);
74 Route::post(uri: 'webhooks/stripe', action: [WebHookController::class, 'stripe']);
75 Route::post(uri: 'webhooks/paypal', action: [WebHookController::class, 'paypal']);
76 Route::post(uri: 'webhooks/mollie', action: [WebHookController::class, 'mollie']);

208 {
209     return (object) [];
210 }
211 }

```

Add your function in WebHookController

```

8 class WebHookController extends CoreController
9 {
10
11     public function stripe(Request $request)
12     {
13         return Payment::handleWebHooks($request);
14     }
15
16     public function paypal(Request $request)
17     {
18         return Payment::handleWebHooks($request);
19     }
20
21     public function razorpay(Request $request)
22     {
23         return Payment::handleWebHooks($request);
24     }
25
26     public function mollie(Request $request)
27     {
28         return Payment::handleWebHooks($request);
29     }
30 }

```

To handle **webhook events** follow your payment gateway official webhook documentation

```

82 public function handleWebHooks(Request $request): void
83 {
84     $webhookSecret = config('shop.razorpay.webhook_secret');
85     $webhookBody = @file_get_contents('php://input');
86     $webhookSignature = $request->header('X-Razorpay-Signature');
87
88     try {
89         if ($webhookBody && $webhookSignature && $webhookSecret) {
90             $this->api->utility->verifyWebhookSignature($webhookBody, $webhookSignature, $webhookSecret);
91         } else {
92             // Invalid request
93             http_response_code(400);
94             exit();
95         }
96     } catch (SignatureVerificationError $e) {
97         // Invalid signature
98         http_response_code(400);
99         exit();
100     }
101
102     $eventStatus = (string) Str::of($request->event)->replace('payment.', '', $request->event);
103
104     switch ($eventStatus) {
105         case 'dispute.won':
106             $this->updatePaymentOrderStatus($request, $eventStatus::PENDING, $eventStatus::PROCESSING);
107             break;
108         case 'dispute.created':
109             $this->updatePaymentOrderStatus($request, $eventStatus::PROCESSING, $eventStatus::SUCCESS);
110             break;
111         case 'captured':
112             $this->updatePaymentOrderStatus($request, $eventStatus::PENDING, $eventStatus::FAILED);
113             break;
114         case 'failed':
115             $this->updatePaymentOrderStatus($request, $eventStatus::PENDING, $eventStatus::FAILED);
116             break;
117     }
118     // To prevent loop for any case
119     http_response_code(200);
120     exit();
121 }

```

**\* Note : \*** For locally webhook testing you can use ngrok tools for that. Please follow their official documentation.

## Getting Started with admin dashboard.

First go to `marvel-admin -> rest -> types -> index.ts -> export enum PaymentGateway` then add PaymentGateway Name.

```

24 export enum PaymentGateway {
25     STRIPE = 'STRIPE',
26     COD = 'CASH_ON_DELIVERY',
27     CASH = 'CASH',
28     FULL_WALLET_PAYMENT = 'FULL_WALLET_PAYMENT',
29     PAYPAL = 'PAYPAL',
30     MOLLIE = 'MOLLIE',
31     RAZORPAY = 'RAZORPAY',
32 }

```

after that go to `marvel-admin -> rest -> src -> components -> settings -> payment.ts` add name & Title.

```
1 export const PAYMENT_GATEWAY = [  
2   { name: 'stripe', title: 'Stripe' },  
3   { name: 'paypal', title: 'Paypal' },  
4   { name: 'razorpay', title: 'RazorPay' },  
5   { name: 'mollie', title: 'Mollie' },  
6 ];
```

Then it will automatically add that payment gateway in the marvel-admin settings.

## Getting Started with shop front.

There are two types of payment gateway system can be integrated here.

- Redirect based payment gateway (e.g PayPal). Where the customer will redirect to that payment gateway site during order checkout. Complete the payment there. And then comeback to the application.
- Non redirect based payment gateway. Where the customer will stay on the application and complete the whole payment process here. Here we consider Stripe as a non-redirect based payment gateway. Though Stripe has features too similar to redirect based payment gateway.

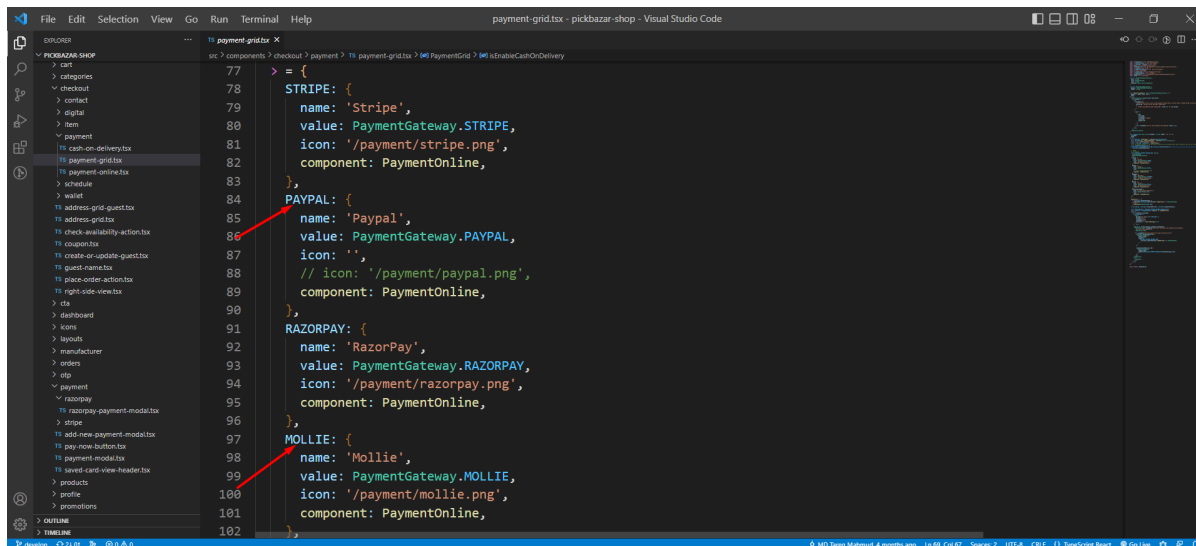
## Redirect-base Payment Gateway

if you want to integrate redirect based payment gateway (e.g. PayPal, Mollie). follow the steps

First go to `shop -> src -> types -> index.ts -> export enum PaymentGateway` then add PaymentGateway Name.

```
24 export enum PaymentGateway {  
25   STRIPE = 'STRIPE',  
26   COD = 'CASH_ON_DELIVERY',  
27   CASH = 'CASH',  
28   FULL_WALLET_PAYMENT = 'FULL_WALLET_PAYMENT',  
29   PAYPAL = 'PAYPAL',  
30   MOLLIE = 'MOLLIE',  
31   RAZORPAY = 'RAZORPAY',  
32 }
```

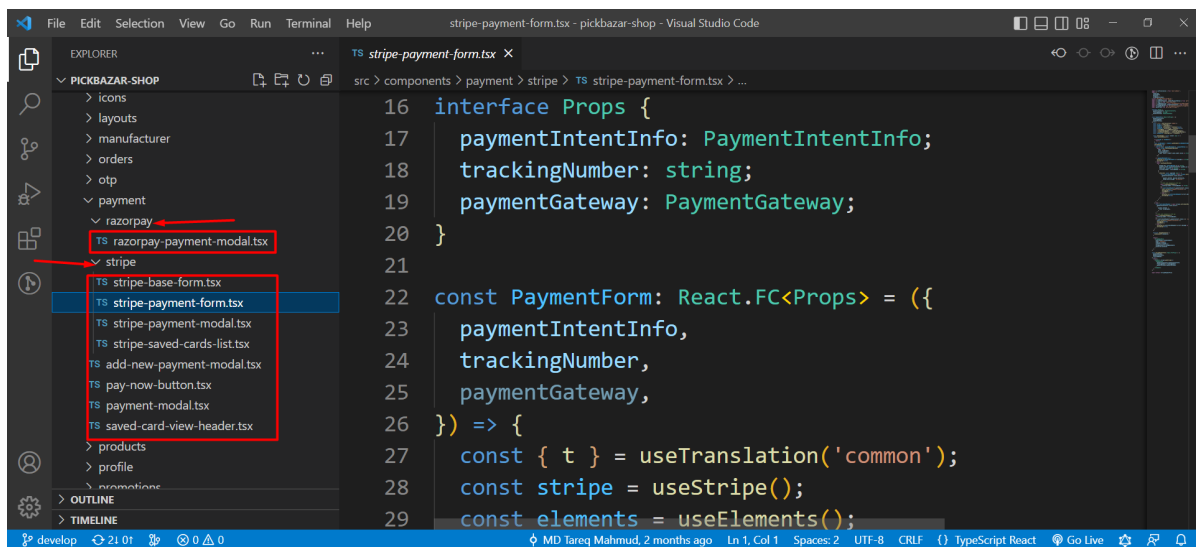
after that go to `shop -> src -> components -> checkout -> payment -> Payment-grid.tsx` then add your PaymentGateway object.



## Non-redirect based payment gateway

First, complete the redirect-based payment gateway steps mentioned above. Because that two steps is universal for all payment gateway to apply.

After That go to `shop -> src -> components -> payment` then add a folder like Stripe, Razorpay. Inside your payment folder you can create your required typescript files with related functionalities for your Payment Gateway. For example, you can checkout the Stripe folder. You can find all the necessary indication, components guide, payment-method (card) saving options etc in the Stripe folder.



That's all for today. If you need any more help you can always contact with our support agents in our support portal.

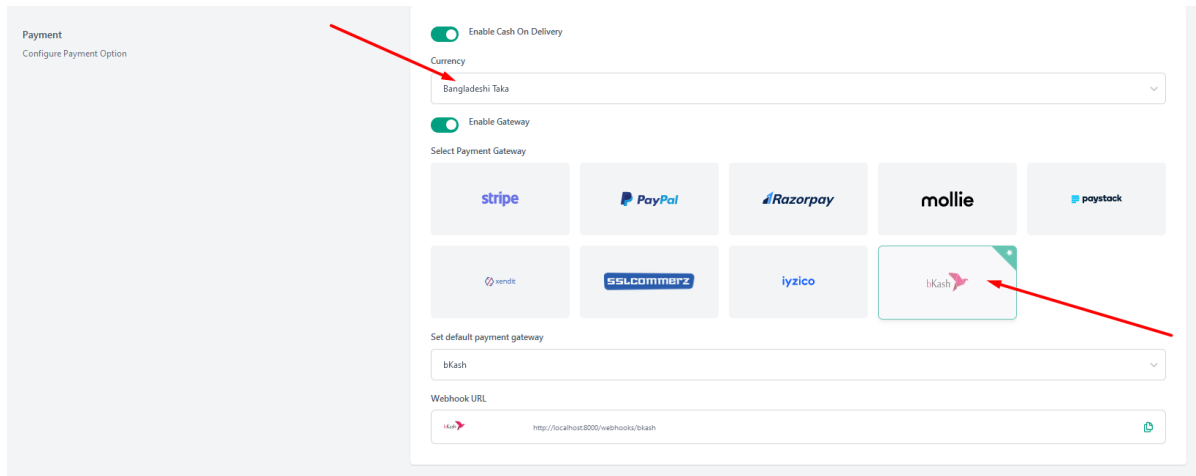
## bKash

bKash is a mobile financial service in Bangladesh that allows users to make financial transactions through their mobile phones. It was launched in 2011 as a subsidiary of BRAC Bank, one of the leading banks in Bangladesh. bKash provides a range of services such as money transfer, mobile recharge, bill payment, merchant payments, and more.

## bKash integrate inside PickBazar.

Please follow & complete this steps for bKash integration for your e-commerce system.

- First go to settings from PickBazar admin dashboard. Inside settings you will find Currency option. Select Bangladeshi Taka (BDT) Currency. bKash primarily operates in the currency of Bangladesh, which is the Bangladeshi Taka (BDT).



The screenshot shows the 'Payment' settings page in the PickBazar admin dashboard. The 'Currency' dropdown is set to 'Bangladeshi Taka'. The 'Enable Gateway' toggle is turned on. Under 'Select Payment Gateway', the 'bKash' option is highlighted with a red arrow. Below this, the 'Set default payment gateway' dropdown is also set to 'bKash'. The 'Webhook URL' field contains the URL 'http://localhost:8000/webhooks/bkash'.

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in [bkash official website](#).

```
BKASH_SANDBOX=true #for production use false
BKASH_APP_KEY=
BKASH_APP_SECRET=
BKASH_USERNAME=
BKASH_PASSWORD=
```

**SHOP\_URL=[YOUR\_SHOP\_URL]** This parameter is must have in `.env` file when bKash is using. Otherwise the payment redirection will be broken.

## How to create & setup bKash information properly?

- Go to the [bKash official website](#) & Click on Live Chat Button,



- After That Click on `customer service` for new merchant API Keys.



- Then apply API keys for merchant mobile Number.
- 3-10days after they will give you your

BKASH\_APP\_KEY

BKASH\_APP\_SECRET

BKASH\_USERNAME

& BKASH\_PASSWORD .

## Special Notes for bKash.

If we have used any third party system/plugin/packages, then we have always encouraged our respected customers to follow the official documentation for detailed & in-depth knowledge.

- [bKash official documentation](#)
- [bKash API reference](#)

## Paymongo

The PayMongo API allows you to build customized and unique payment experiences for your website, application or service. Our goal is to give you access to pretty much all the necessary resources to integrate payments into your platform.

## Paymongo integrate inside PickBazar.

Please follow & complete these steps for Paymongo integration for your e-commerce system.

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file.  
We will discuss later about how to create those API keys in Paymongo.

```
PAYMONGO_PUBLIC_KEY=[YOUR_PAYMONGO_PUBLIC_KEY]
PAYMONGO_SECRET_KEY=[YOUR_PAYMONGO_SECRET_KEY]

# Paymongo Webhook settings

PAYMONGO_WEBHOOK_SIG=[YOUR_PAYMONGO_WEBHOOK_URL]
```

- To activate Paymongo go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. webhook URL is coming from local development. This static link will dynamically generated in live environment)

**PickBazar** Create Shop

**Left Sidebar:**

- Create Order
- Users
- Coupons
- Taxes
- Shippings
- Withdrawals
- Message
- Refunds
- Questions
- Store Notice
- Reviews
- Settings**

**Main Content Area:**

**Summary Cards:**

- Total Revenue (Last 30 Days): \$0.00
- Total Order (Last 30 Days): 0
- Todays Revenue: \$0.00
- Total Shops: 7

**Sale History:**

2.0
1.6
1.2
0.8
0.4

**Information**  
Change your site information from here

**Site Title:** Pickbazar

**Site Subtitle:** Your next ecommerce

**Currency:** Philippine Peso

**Minimum Order Amount:** 0

**Enable Cash On Delivery:** ☒

**Select Payment Gateway:** Paymongo

**Webhook URL:** http://localhost:8000/webhooks/paymongo

**Payment**  
Configure Payment Option

**Footer:**

**Menu:**

- Activate your account (2 out of 4 steps completed)
- Home
- Links
- Pages
- Payments
- Payouts
- Developers**
- Security logs
- Settings

**TEST API Keys:**

Public Key	[Redacted]
Secret Key	[Redacted]

**Plugins:**

Install PayMongo payment methods on Shopify, Prestashop, WooCommerce, and Magento  
<https://developers.paymongo.com/docs/plugins>

## How to create paymongo webhook url?

- Open your terminal in API project for artisan commend
- Then run this commend

```
php artisan paymongo:webhook
```

## You can visit for paymongo.

- [paymongo official documentation](#)
- [paymongo Webhooks documentation](#)

## Flutterwave

---

Flutterwave is a secure and versatile payment gateway that facilitates seamless online transactions. It enables businesses to accept payments from various sources, including credit cards, debit cards, and mobile money, in multiple currencies. With user-friendly integration and advanced fraud protection, Flutterwave ensures secure and hassle-free payment processing. It also offers comprehensive analytics and reporting tools, empowering businesses to track and manage their transactions effectively. With its reliability and convenience, Flutterwave is a trusted solution for online payments, providing a smooth and secure experience for both merchants and customers.

### Flutterwave integrate inside PickBazar.

Please follow & complete this steps for Flutterwave integration for your e-commerce system.

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in RazorPay.
- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in [Paystack official website](#).

```
FLW_PUBLIC_KEY=[YOUR_FLW_PUBLIC_KEY]
FLW_SECRET_KEY=[YOUR_FLW_SECRET_KEY]
FLW_SECRET_HASH=[FLW_SECRET_HASH]
```

**SHOP\_URL=[YOUR\_SHOP\_URL]** This parameter is must have in `.env` file when Flutterwave is using. Otherwise the payment redirection will be broken.

### Flutterwave Webhook settings

- To activate Flutterwave go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. Paystack Webhook URL is coming from local development. This static link will dynamically generated in live environment)

**PickBazar** Create Shop

- Create Order
- Users
- Coupons
- Taxes
- Shippings
- Withdrawals
- Message
- Refunds
- Questions
- Store Notice
- Reviews
- Settings**

**Total Revenue**  
(Last 30 Days)

\$0.00

**Total Order**  
(Last 30 Days)

0

**Todays Revenue**

\$0.00

**Total Shops**

7

**Sale History**

2.0

1.6

1.2

0.8

0.4

Enable Gateway

Select Payment Gateway

stripe PayPal Razorpay mollie paymongo

paystack vendr SSLCOMMERZ iyzico bKash

Flutterwave

Set default payment gateway

Flutterwave

Webhook URL

stripe http://localhost:8000/webhooks/stripe

Flutterwave http://localhost:8000/webhooks/flutterwave

- Copy the webhook url & registered in Paystack Dashboard.

General Bank Accounts Users API **Webhooks** Account Settings

**Live Webhooks**

URL

https://yourdomain/webhooks/flutterwave

Secret hash

pay

What is secret hash?

☒ Receive Webhook response in JSON format

☒ Enable Webhook retries @

☒ Enable failed transactions webhook

https://yourdomain/webhooks/flutterwave Edit Webhook

☒ Enable V3 Webhooks

Save

- If you want to use webhook during development on localhost, you must use a tools like ngrok to have the webhooks delivered to your local machine.

```
ngrok - ngrok http http://localhost:8000 (Ctrl+C to quit)
Check which logged users are accessing your tunnels in real time https://ngrok.com/s/app-users
Session Status online
Account [REDACTED]
Version 3.1.0
Region India (in)
Latency 116ms
Web Interface http://127.0.0.1:4040
Forwarding https://c852-37-111-243-195.in.ngrok.io -> http://localhost:8000
Connections
  ttl   opn   rt1   rt5   p50   p90
    0     0    0.00  0.00  0.00  0.00
```

- Copy the forwarding https link & paste on Test Webhook URL, after paste forwarding https link, add line- /webhooks/flutterwave.

Live Webhooks

URL

http://c852-37-111-243-195.in.ngrok.io/webhooks/flutterwave

Secret hash

What is secret hash?

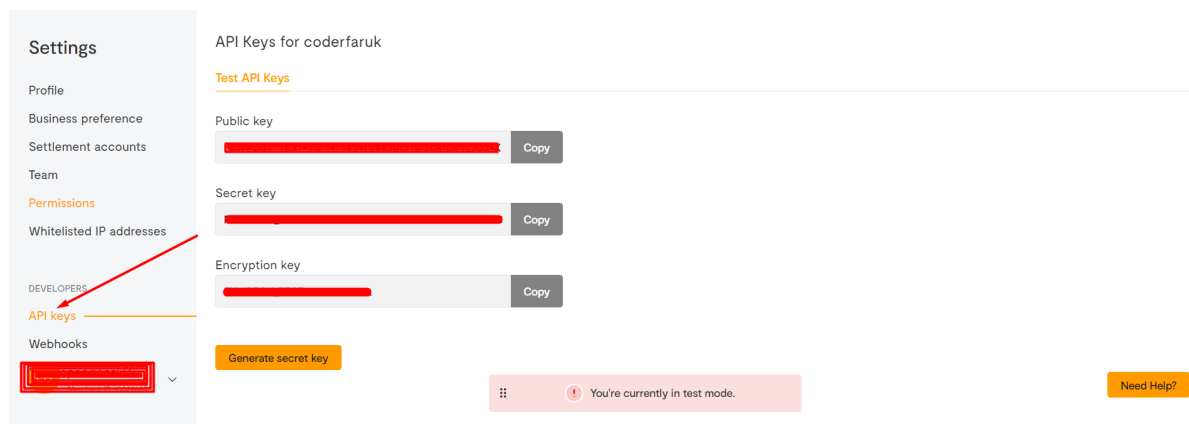
pay

## ### How to create & setup Flutterwave information properly?

- Go to the [Paystack official website](#) & login If you aren't a registered user, Complete the Flutterwave registration first.
- After logged in into Flutterwave dashboard, Click on Settings.



- After That Click on API Keys for Paystack Secret Key & Paystack Public Key.



- Copy & paste those inside PickBazar API and PickBazar Shop `.env` file like mentioned above.

## Special Notes for Flutterwave.

If we have used any third party system/plugin/packages, then we have always encouraged our respected customers to follow the official documentation for detailed & in-depth knowledge.

- [Flutterwave official documentation](#)
- [Flutterwave Webhooks documentation](#)

## Support Portal

For support, Please open a ticket in the support forum

<https://redqsupport.ticksy.com/>

There are some basic requirements for our application so that support team can help you quick such as,

1. Asking queries regarding feature that is already implemented in the application
2. Following recommended configuration, environments & server which you need to met first before you proceed with installation, deployment in your server to receive support.
3. Support query need to be within Envato item support policy. ([https://themeforest.net/page/item\\_support\\_policy](https://themeforest.net/page/item_support_policy))
4. You should maintain only one support ticket at a time. Creating multiple ticket can cause unexpected delays.
5. Ticket should provide as much details as possible related to the issue such as screenshot, video explanation, access, how to reproduce, environments, if any changes or customisations are made etc

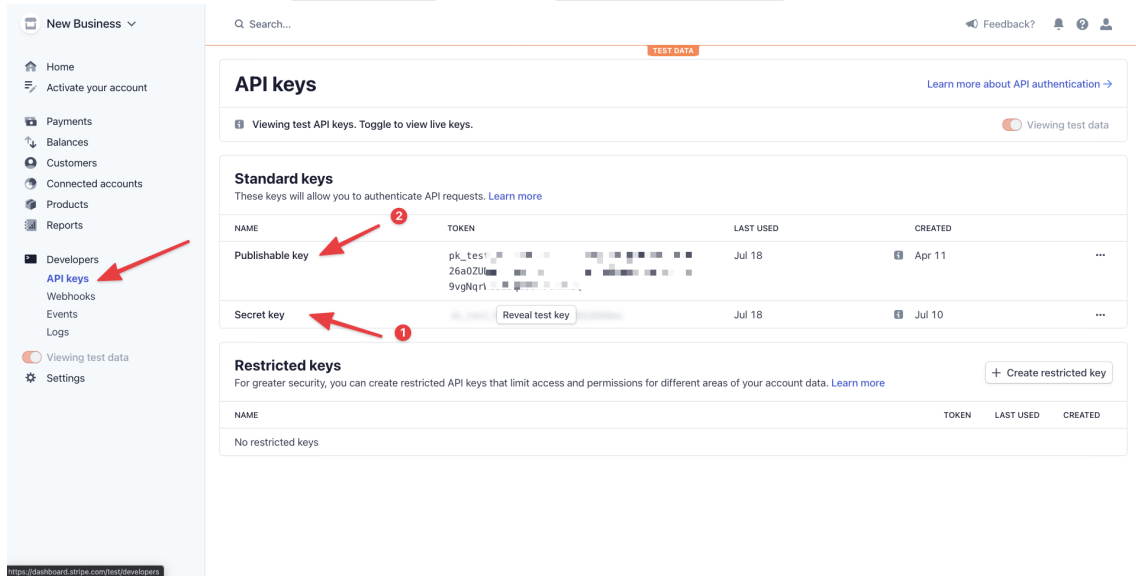
During working days our ticket response can take 24 hours to 48 hours depending on the volume of tickets pending prior to your ticket . We follow Envato Item support policy [https://themeforest.net/page/item\\_support\\_policy](https://themeforest.net/page/item_support_policy) to provide standard support for our items

# FAQs

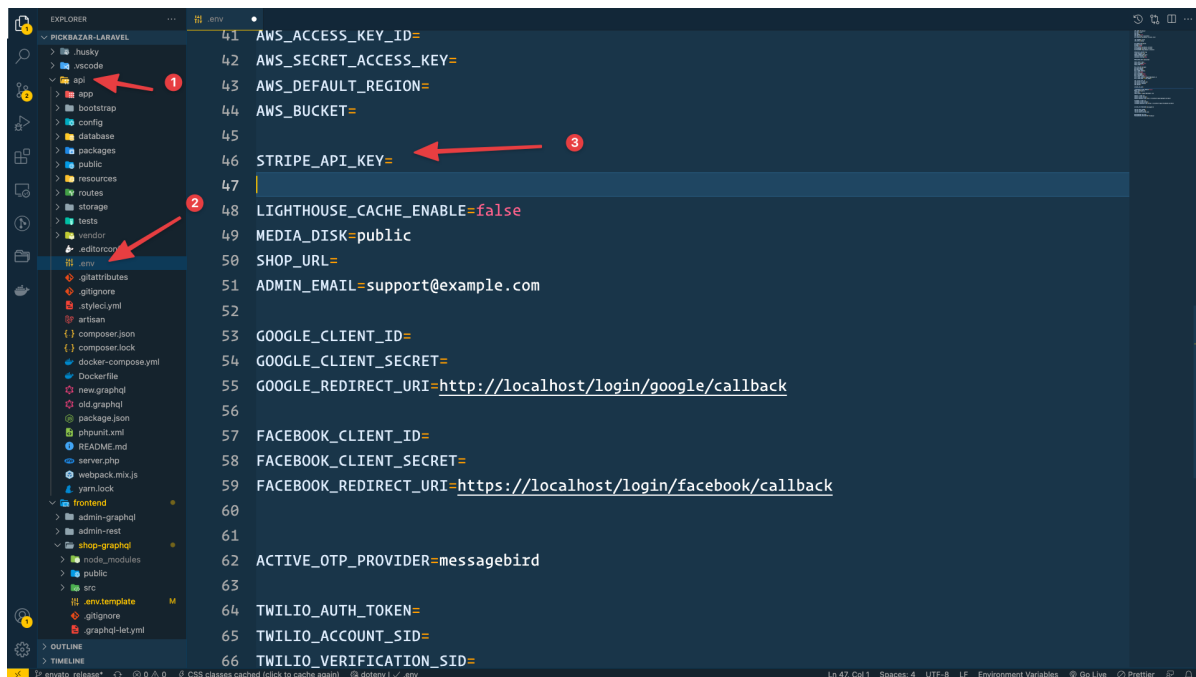
## How to configure Stripe payment gateway?

To configure **Stripe** payment gateway,

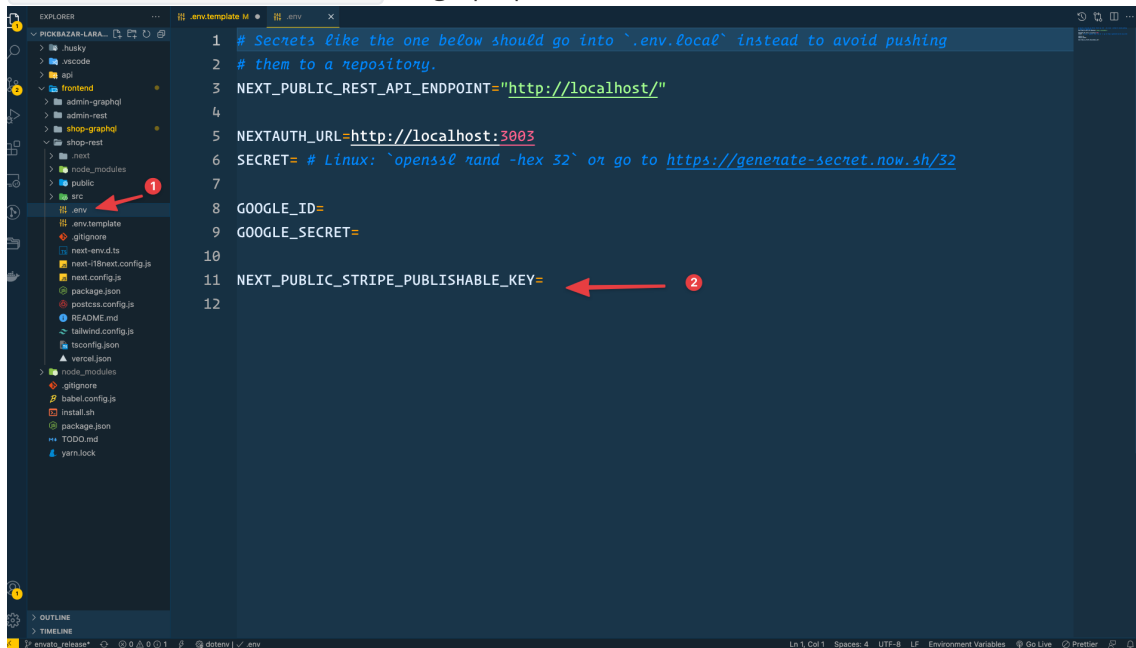
1. At first, create a developer account from stripe developer dashboard (<https://dashboard.stripe.com/>)
2. log in to that account
3. Then go to the **Developer -> API keys** section and on that section create an API key. After creating, you'll get one **Secret key** and one **Publishable key**.



4. Then copy **Secret key** to your **pickbazar api->.env**



5. And similarly, add `Publishable` key to your `frontend/shop-rest/.env` for rest and `frontend/shop-graphql/.env` for graphql.



After configuration, make sure you rebuild your project using this command,

For REST API

```
yarn build:shop-rest
yarn build:admin-rest
```

And for GraphQL,

```
yarn build:shop-gql
yarn build:admin-gql
```

## Why am I facing "You may need an appropriate loader to handle this file type" during running shop rest?

Ans: Please run `yarn clean` then `yarn` then run `yarn dev:shop-rest`

## I am changing schema files but changes is not working

Your changes might not work because schema is cached. SO you might need to clear schema cache using the below command `php artisan lighthouse:clear-cache`.

## Changing .env files but not getting the changes

Run Below command `php artisan optimize:clear`

## Changing route but not getting the changes.

Run `php artisan optimize:clear` or `php artisan route:clear`

# I have set `STRIPE_API_KEY` in `.env` but still getting error.

In some cases `STRIPE_API_KEY` value can't read from `.env` in those cases you have to put the key in the config file directly in `api/packages/marvel/src/Config/laravel-omnipay.php`

## 6. Getting error on forget password email sending

Make sure you have run the `php artisan marvel:install` commands successfully and copied the necessary email templates to your resources folder. You can also do it by `php artisan marvel:copy-files` command.

NB: This same issue can occur during order creation.

## Can I use it with my existing laravel?

Yes, you can. Follow the below steps.

- Make sure you are using laravel 8.
- Copy `api/packages` folder from the downloaded files into your laravel root folder.
- Put below code in your laravel `composer.json` file into `require` section.

```
"ignited/laravel-omnipay": "dev-master",  
"omnipay/common": "dev-master",  
"omnipay/stripe": "dev-master",  
"pickbazar/shop": "dev-master"
```

- Put below code in bottom of your `composer.json`. If you already have an `repositories` section then put code inside `repositories` to your existing `repositories`

```
"repositories": {  
    "pickbazar/shop": {  
        "type": "path",  
        "url": "packages/marvel"  
    },  
    "ignited/laravel-omnipay": {  
        "type": "path",  
        "url": "packages/laravel-omnipay"  
    },  
    "omnipay/common": {  
        "type": "path",  
        "url": "packages/omnipay-common"  
    },  
    "omnipay/stripe": {  
        "type": "path",  
        "url": "packages/omnipay-stripe"  
    }  
}
```

- Now run `composer install`
- Copy necessary env variables from `.env.example` to you env file.

- Run `php artisan marvel:install` and follow necessary steps.
- To run server `php artisan serve`
- For image upload to work properly you need to run `php artisan storage:link`.

## Why am I getting Access denied for user?

navigate to `api` then run `./vendor/bin/php down -v`. It will delete any of your existing mysql volumes. Now run `./vendor/bin/php up -d` on same directory or run `bash install.sh` on root directory

## Why am I getting permission issue during deployment?

Run below commands for fixing permission issue in your laravel app during deployment

```
sudo chown -R $USER:www-data storage
sudo chown -R $USER:www-data bootstrap/cache

sudo chown -R www-data:www-data storage
sudo chown -R www-data:www-data bootstrap/cache
```

## Why am I getting "The GET method is not supported for this route. Supported methods: HEAD"?

Run `php artisan optimize:clear`

## How to resolve the Load More Infinity loading issue?

If you click on the `Load More` button and the button is spinning continuously, then check API request from the network tab. If it is HTTP and your API is hosted on HTTPS, then open,

```
/Users/tareqmahmud/tmbox/tmserver/larabox/work/redq/laravel-ecommerce/api/app/Providers/AppServiceProvider.php
```

And add this code to `boot` method,

```
if (!\App::environment('local')) {
    $this->app['request']->server->set('HTTPS', true);
}
```

## I'm trying to upload images, but the images are not displayed on the frontend?

We support two types of file uploads; one is `local`, and another is `AWS S3`.

Both uploading systems follow this procedure,

## API

At first, add your API URL to your `api -> .env`.

```
2 APP_ENV=production
3 APP_KEY=
4 APP_DEBUG=true
5 APP_URL=https://YOUR_DOMAIN.COM/backend
6 APP_SERVICE=marvel.test
7 APP_NOTICE_DOMAIN=PICKBAZAR_
8 DUMMY_DATA_PATH=pickbazar
9
10
11 LOG_CHANNEL=stack
12 LOG_LEVEL=debug
13
```

If you're using windows and your API is running with ports like `localhost:8000` or `127.0.0.1:8000`, then make sure you add the domain with port to `APP_URL`. Like this  
`API_URL=http://localhost:8000`

After that, clear your API cache by using this command from the `api` folder,

```
php artisan optimize:clear
```

Then link the storage system by using this command,

```
php artisan storage:link
```

If you're using AWS S3, then update `MEDIA_DISK=s3` and add all `AWS_` credentials  
For AWS s3, make sure your properly setup permission of the bucket with ACL enable by  
follow this link -> <https://stackoverflow.com/a/70603995/2158023>

Amazon S3 > Buckets > pickbazarlaravel > Edit Object Ownership

### Edit Object Ownership Info

#### Object Ownership

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☐ ACLs disabled (recommended)

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☒ ACLs enabled

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

#### Object Ownership

☐ Bucket owner preferred

If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

☒ Object writer

The object writer remains the object owner.

Cancel

Save changes

## Admin

Then add your domain to your `pickbazar-laravel -> admin -> rest -> next.config.js` for `rest` or `pickbazar-laravel -> admin -> graphql -> next.config.js` for `graphql`

```
images: {
  domains: [
    "YOUR_DOMAIN.COM",
    "via.placeholder.com",
    "res.cloudinary.com",
    "s3.amazonaws.com",
    "18.141.64.26",
    "127.0.0.1",
    "localhost",
    "picsum.photos",
    "pickbazar-sail.test",
    "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
    "chawkbazarlaravel.s3.ap-southeast-1.amazonaws.com",
    "lh3.googleusercontent.com",
  ],
},
```

If you're using `AWS S3`, then also add your root S3 domain like this,

```
images: {
  domains: [
    "YOUR_DOMAIN.COM",
    "via.placeholder.com",
    "res.cloudinary.com",
    "s3.amazonaws.com",
    "18.141.64.26",
    "127.0.0.1",
    "localhost",
    "picsum.photos",
    "pickbazar-sail.test",
    "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
    "chawkbazarlaravel.s3.ap-southeast-1.amazonaws.com",
    "lh3.googleusercontent.com",
  ],
},
```

You, 7 minutes ago • Uncommitted changes

Like this

## Shop

Similarly, add your domain to your `pickbazar-laravel -> shop -> next.config.js`

```
images: {
  domains: [
    "YOUR_DOMAIN.COM",
    "via.placeholder.com",
    "res.cloudinary.com",
    "s3.amazonaws.com",
    "18.141.64.26",
    "127.0.0.1",
    "localhost",
    "picsum.photos",
    "pickbazar-sail.test",
    "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
    "chawkbazarlaravel.s3.ap-southeast-1.amazonaws.com",
    "lh3.googleusercontent.com",
  ],
},
```

If you're using `AWS S3`, then also add your root S3 domain like this,

```
images: {  
  domains: [  
    "YOUR_DOMAIN.COM",  
    "via.placeholder.com",  
    "res.cloudinary.com",  
    "s3.amazonaws.com",  
    "18.141.64.26",  
    "127.0.0.1",  
    "localhost",  
    "picsum.photos",  
    "pickbazar-sail.test",  
    "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",  
    "chawkbazarlaravel.s3.ap-southeast-1.amazonaws.com",  
    "lh3.googleusercontent.com",  
  ],  
}
```

You, 7 minutes ago • Uncommitted changes

**Like this** →

## Build Frontend,

If you run in dev mode, then re-run will affect your update. But if you're running as a production mode, then you've to rebuild the project and re-run again. Otherwise, image upload will not work.

<https://pickbazar-doc.vercel.app/vps-server#build-project>

After building the project, replace your server shop and admin folder using this rebuild folder and re-run the app using this command,

```
pm2 restart all
```

## How to resolve the 502 Bad Gateway error for frontend/admin?

There can be several reasons that throw 502 errors for the frontend in the production server.

### Reason 1:

On your `Nginx` file, a specific port is set for shop or the admin, so when you run the script as a `PM2`` instance, the script has to be run that specific port; otherwise, your domain/subdomain will throw a 502 error.

### PORT:

```
3000 -> Shop Rest  
3001 -> Admin Rest  
  
3000 -> Shop GraphQL  
3004 -> Admin GraphQL
```

To check that at the first stop the `PM2` instance by this command,

```
pm2 stop 'all'
```

then go to the `pickbazar` folder from your server and try to run the script using `yarn` and check which port it is running,

```
Rest:
yarn start:admin-rest
yarn shop-rest
GraphQL:
yarn start:admin-gql
yarn shop-gql
```

And, check if the port matched with the Nginx port or not? If not matched, then change the port to `Nginx` config.

## Reason 2:

To check that at the first stop the PM2 instance by this command,

```
pm2 stop 'all'
```

then go to the `pickbazar` folder from your server and try to run the script using yarn and check which port it is running,

```
Rest:
yarn start:admin-rest
yarn shop-rest
GraphQL:
yarn start:admin-gql
yarn shop-gql
```

With this command, it'll give you an error that no build file is found. To resolve that, rebuild your project and then rerun the scripts.

<https://pickbazar-doc.vercel.app/vps-server#frontend-project-build>

## How to resolve javascript heap out of memory issue?

If you're using cPanel based server, then you maybe get an error like "call\_and\_retry\_last allocation failed - javascript heap out of memory".

To resolve that issue, you've to add a limit of the memory bound,

Please check this,

<https://stackoverflow.com/questions/38558989/node-js-heap-out-of-memory>

<https://support.snyk.io/hc/en-us/articles/360002046418-JavaScript-heap-out-of-memory>

<https://blog.openreplay.com/javascript-heap-out-of-memory-error>

## Image upload throw Internal Server Error; how to resolve that?

For uploading, the php8.1-gd library is required. So make sure you install the php8.1-gd library on your server.

## How to resolve docker invalid reference format: repository name must be lowercase?

To resolve that issue, you've to move the `laravel-ecommerce` folder from `Pickbazar Laravel - React Next REST & GraphQL Ecommerce` to a generic folder where you store your code. After that, use this command from your `laravel-ecommerce` folder,

```
chmod +x install.sh
bash install.sh
```

## Sometimes my server shutdown or halts automatically. After restart, it works fine again; how to resolve that?

In general, this happens mainly when your upstream server fails to handle its processing power when there is lots of traffic or requires lots of operation. Please upgrade your server configuration to double to check the issue. Please check this video,

<https://www.youtube.com/watch?v=4plqAz5bzX0>

If that doesn't solve the issue, then you've to redeploy again with a new ec2 instance.

## S3 uploading doesn't work after adding credentials to .env; how to resolve that?

For AWS s3, make sure your properly setup permission of the bucket with ACL enable by follow this link

<https://stackoverflow.com/a/70603995/2158023>

## How to rebuild the project?

When you upload the code to the server as a production, NextJS fetch its code from the build file. So after deployed, when you update or change code on scripts, the updated scripts don't work at the front end until you rebuild the project again. Nextjs works this way.

Now there are two ways you can follow to rebuild the project,

If your server has more than 2+ CPU core and 6GB+ of memory, then you can directly edit code at your server then rebuild the project from the server directly using this command,

At first go to `/var/www/pickbazar-laravel`

For `shop rest`.

```
yarn build:shop-rest
```

for graphql `yarn build:shop-gql`

And for `admin rest`.

```
yarn build:admin-rest
```

```
for graphql yarn build:admin-gql
```

After completing the build, restart the PM2 process using this command,

```
pm2 restart 'all'
```

But suppose your server doesn't have that processing power, and you try to build your scripts directly from the server. In that case, your server will be shut down during the build, and you've to restart the server.

In that case, you can rebuild the scripts at your local computer and upload them to your server. If you follow our [VPS Server Deployment](#) guide, then you already know the process. To build the scripts on your pc, follow this procedure,

At first, download the `pickbazar-laravel` folder from the server to your computer, and customize the code at your computer. Then use this command from the `root` folder,

To install all the packages,

```
yarn
```

Then build the shop rest,

```
yarn build:shop-rest
```

```
for graphql yarn build:shop-gql
```

Then build the admin rest,

```
yarn build:admin-rest
```

```
for graphql yarn build:admin-gql
```

Then delete the shop and admin folder from the server and upload your build shop and admin folder to the server at the exact same location and use this command to restart the PM2,

```
pm2 restart 'all'
```

## How to increase upload size?

The API supports `10MB` of upload limit for a single file by default. If your upload file is more than `10MB`, then please follow this,

At first, login to the server using ssh, and then,

Edit,

```
/etc/php/8.1/fpm/php.ini
```

Then update,

```
upload_max_filesize = 256M
post_max_size = 256M
max_execution_time = 300
```

Then restart the PHP,

```
sudo service php8.1-fpm restart
```

Then update,

```
/etc/nginx/sites-enabled/pickbazar
```

and Add `client_max_body_size 256M`

```
client_max_body_size 256M
```

```
server {
    server_name ██████████.
    client_max_body_size 256M;
    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-XSS-Protection "1; mode=block";
    add_header X-Content-Type-Options "nosniff";

    index index.html index.htm index.php;

    charset utf-8;
```

Then restart the server,

```
sudo nginx -t
sudo service nginx restart
```

After that update,

```
api/packages/marvel/config/media-library.php
'max_file_size' => 1024 * 1024 * 256,
```

```

1  <?php
2
3  return [
4
5      /*
6       * The disk on which to store added files and derived images by default.
7       * one or more of the disks you've configured in config/filesystems.
8       */
9      'disk_name' => env('MEDIA_DISK', config('shop.media_disk')),
10
11     /*
12      * The maximum file size of an item in bytes.
13      * Adding a larger file will result in an exception.
14      */
15     'max_file_size' => 1024 * 1024 * 256,
16
17     /*
18      * This queue will be used to generate derived and responsive images.
19      * Leave empty to use the default queue.
20      */
21     'queue_name' => '',
22
23     /*

```

After that, run this command from the `api.` folder,

```
php artisan optimize:clear
```

And for `admin`, increase `timeout` from,

```
admin/rest/src/utils/api/http.ts
```

```

http.ts
You, 1 second ago | 3 authors (explitan and others)
1  import { getAuthCredentials } from "@utils/auth-utils";
2  import { ROUTES } from "@utils/routes";
3  import axios from "axios";
4  import Cookies from "js-cookie";
5  import Router from "next/router";
6
7  const http = axios.create({
8      baseURL: process.env.NEXT_PUBLIC_REST_API_ENDPOINT, // TODO: take t
9      timeout: 600000,
10     headers: {
11         Accept: "application/json",
12         "Content-Type": "application/json",
13     },
14 });
15
16 // Change request data/error here
17 http.interceptors.request.use(
18     (config) => {
19         const { token } = getAuthCredentials();
20         config.headers = {
21             ...config.headers,
22             Authorization: `Bearer ${token}`,
23         };
24         return config;

```

```
timeout: 600000
```

After that [Rebuild Your Project](#) and restart again.

## How to resolve **docker: invalid reference format: repository name must be lowercase.?**

Please open your terminal and use this command,

```
brew services stop mysql
```

It'll stop any MySQL instance that is installed using brew.

Then use this command to check is there MySQL running or not,

```
mysql -v
```

If no MySQL is running then, go to,

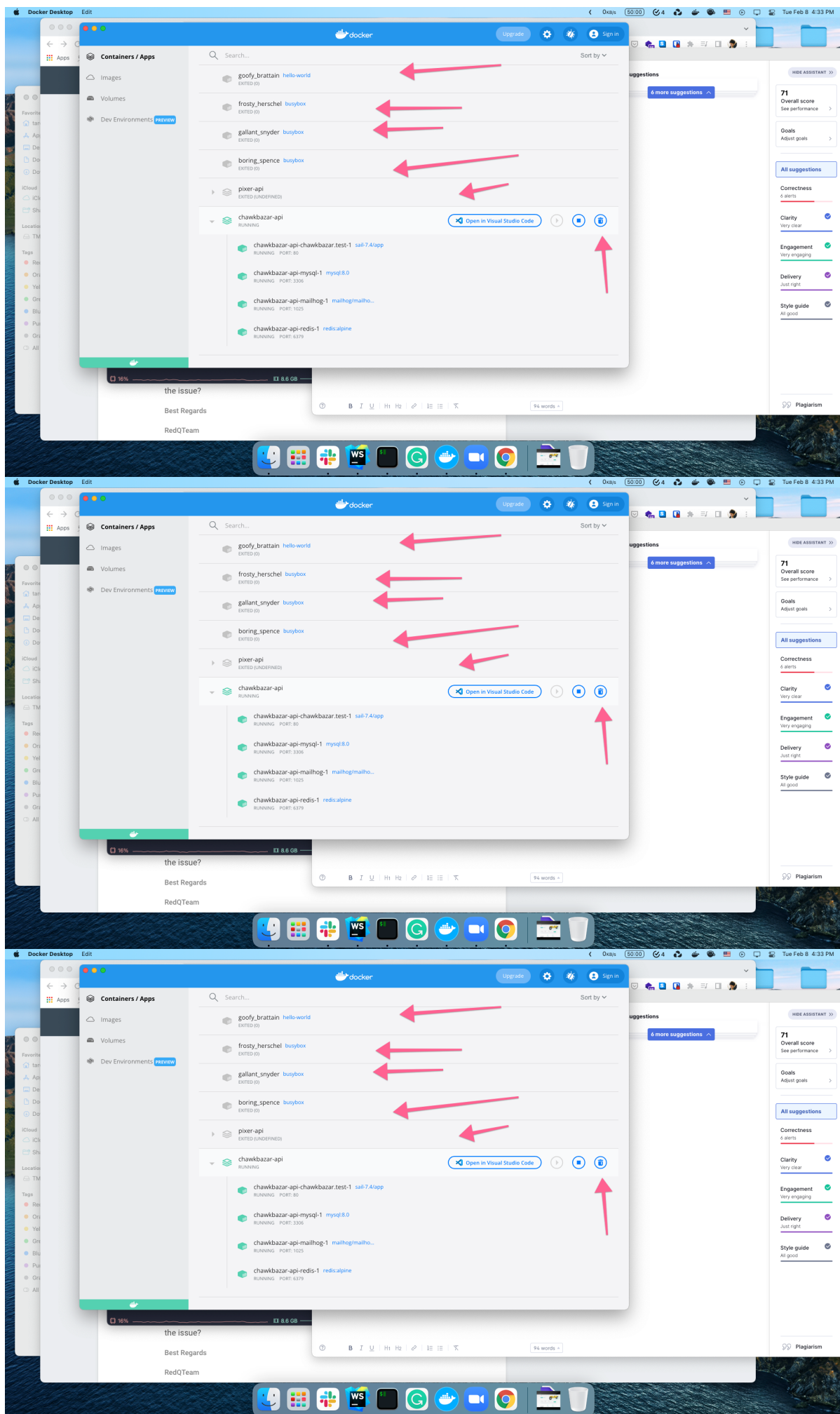
```
pickbazar-laravel -> api
```

and use this command,

```
./vendor/bin/sail down -v
```

After that,

Open docker dashboard and delete all the existing container and images,



And after that remove,

```
pickbazar-laravel -> api -> .env
```

And then go to your root `pickbazar-laravel` folder and use this command again,

```
chmod +x install.sh  
bash install.sh
```

## How to remove the existing payment gateway?

You can remove the Stripe Payment from the available payment array,

```
shop/src/components/checkout/payment/payment-grid.tsx
```

## Why checkout **Place Order** button is disabled?

The place order button will be activated only when you fill in this information,

- OTP (Mobile number)
- Billing Address
- Shipping Address
- Delivery Schedule
- Payment Gateway

## How do I log in to the admin panel as the main administrator?

When you install the API using `php artisan marvel:install` or using `bash install.sh (docker)`, you'll get a prompt to create an admin account, and you've to create admin credentials that time. And then, you can use that credentials for the main admin account.

## How to upgrade the existing deployed laravel 8 server to laravel 9?

At first, remove all the existing `php 7.4` and its extensions by using this command,

```
sudo apt purge php-fpm php-mysql  
sudo apt purge php-mbstring php-xml php-bcmath php-simplexml php-intl php-mbstring php7.4-gd php7.4-curl php7.4-zip composer
```

Then remove the composer,

```
sudo rm /usr/bin/composer
```

Then delete the `vendor` folder from the `pickbazar-laravel -> api` folder.

```
cd /var/www/pickbazar-laravel/api

sudo chown -R $USER:www-data storage
sudo chown -R $USER:www-data bootstrap/cache

rm vendor -rf
```

Then install PHP 8, and its extensions,

```
sudo add-apt-repository ppa:ondrej/php
sudo apt update
sudo apt install php8.0-fpm php8.0-mysql
sudo apt install php8.0-mbstring php8.0-xml php8.0-bcmath php8.0-simplexml
php8.0-intl php8.0-mbstring php8.0-gd php8.0-curl php8.0-zip
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'e21205b207c3ff031906575712edab6f13eb0b361f2085f1f1237b7126d785e826a450292b6cfd1d
64d92e6563bbde02') { echo 'Installer verified'; } else { echo 'Installer
corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"
sudo mv composer.phar /usr/bin/composer
```

Then update 7.4 fpm to 8.0 fpm from,

```
/etc/nginx/sites-enabled/pickbazar
```

```
charset utf-8;

# For API
location /backend {
    alias /var/www/pickbazar/api/public;
    try_files $uri $uri/ @backend;
    location ~ /\.php$ {
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $request_filename;
        fastcgi_pass unix:/run/php/php8.0-fpm.sock;
    }
}

location @backend {
    rewrite /backend/(.*)$ /backend/index.php?$1 last;
}

# For FrontEnd -> Rest
location /{
    proxy_pass http://localhost:3000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}

location /admin{
    proxy_pass http://localhost:3002/admin;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}

error_page 404 /index.php;

location ~ /\.php$ {
    fastcgi_pass unix:/var/run/php/php8.0-fpm.sock;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
    include fastcgi_params;
}

location ~ /\.(!well-known).* {
    deny all;
}
```

Then install composer packages,

```
cd /var/www/pickbazar-laravel/api
composer install
php artisan optimize:clear
php artisan marvel:install
```

php artisan marvel:install will remove all of your existing data. Ensure you export or backup your data before using that command.

```
sudo chown -R www-data:www-data storage
sudo chown -R www-data:www-data bootstrap/cache
```

## Coupon

Pickbazar is an online store-building tool that enables companies to launch online stores and sell goods. A coupon is a code that a consumer may enter on Pickbazar during the checkout process to get a discount or other promotional offer on their purchase.

On Pickbazar, companies may generate coupons in a variety of ways for their consumers, including percentage-based discounts, fixed amount discounts, free delivery, and buy one, get one free offers. Moreover, they may specify the coupon's validity period, the minimum order amount needed to utilize it, and the maximum number of times it can be used.

Consumers can use the coupon code to get a discount or take advantage of a promotion during the checkout process. Businesses may use coupons as effective marketing tools to draw in new clients, keep old ones coming back, and boost sales at their Pickbazar shop.

## Super Admin

- The coupon option is accessible through the admin dashboard. Once you enter the coupons, the "Add Coupons" option will be available for adding additional coupons.

The screenshot displays the PickBazar Super Admin dashboard. On the left is a sidebar menu with options: Dashboard, Shops, My Shops, Products, Attributes, Groups, Categories, Tags, Manufacturers/Publications, Authors, Orders, Create Order, Users, Coupons (highlighted with a red arrow), Taxes, Shippings, and Withdrawals. The main content area is titled 'Coupons' and features a search bar with the placeholder 'Type your query and press enter' and an '+ Add Coupon' button. Below this is a table listing existing coupons with columns for ID, Banner, Code, Amount, Minimum Amount, Active status, Expiry date, and Actions. The table contains 10 rows of coupon data.

ID	Banner	Code	Amount	Minimum Amount	Active	Expired	Actions
10		4OFF	\$4.00	\$0.00	2 years ago	in a year	
9		5OFF	\$5.00	\$0.00	2 years ago	in a year	
8		OFF6	\$6.00	\$0.00	2 years ago	in a year	
7		BAZAR10	\$10.00	\$0.00	2 years ago	in 2 years	
6		BAZAR20	\$20.00	\$0.00	2 years ago	in a year	
5		BAZAR18	\$18.00	\$0.00	2 years ago	in a year	
4		HELLO15	\$15.00	\$0.00	2 years ago	in a year	
3		EID12	\$12.00	\$0.00	2 years ago	in a year	
2		FOO	\$8.00	\$0.00	2 years ago	in a year	

- The administrator will click "add coupon" after filling out all the forms to create a new coupon. Which will appear in the coupon list. Here are descriptions of each field:

The screenshot shows the 'PickBazar' coupon creation interface. On the left is a sidebar with various navigation options. The main content area is titled 'Description' and contains a form for creating a new coupon. The form includes a 'Code' field with the value '4OFF', a 'Description' text area, a 'Type' section with radio buttons for 'Fixed' (selected), 'Percentage', and 'Free Shipping', a 'Coupon discount amount (USD)' field with the value '0', a 'Minimum cart amount (USD)' field with the value '0', and two date fields: 'Active From' (15/03/2023) and 'Will Expire' (15/03/2023). A green 'Add Coupon' button is located at the bottom right of the form.

**code** : While making a purchase online or in-person, customers can enter a coupon code, which is a combination of letters and/or numbers, to get a discount or exclusive deal. Often, firms or shops use coupon codes as a marketing strategy to get consumers to purchase their goods or services.

**description** : The offer that a discount code makes to clients is briefly described in the description of the coupon code. The sort of discount or offer, the goods or services it applies to, any limits or limitations, and the expiration date are often all listed. Customers may determine if they wish to use a discount code to make a purchase by reading the description of the code, which explains the terms and limitations of the deal.

**minimum cart amount** : A consumer must have a certain minimum total number of goods or services in their shopping cart in order to use a coupon and benefit from the accompanying discount or deal.

**coupon discount amount** : The amount of the discount that a consumer receives when using a coupon code is known as the coupon discount amount.

**type** : Two distinct sorts of coupons—the fixed coupon type and the percentage coupon type—offer clients various kinds of savings.

Regardless of the overall order quantity, a fixed coupon type provides a particular monetary discount on a purchase.

On the other hand, a percentage coupon type provides a discount based on a percentage of the entire order value.

**active date** : The date on which a coupon becomes active and may be used by customers to obtain the related discount or offer is referred to as the active date.

**expire date** : The date on which a coupon becomes invalid and can no longer be used by consumers to enjoy the accompanying discount or offer is referred to as the coupon's expiry date.

Customers should be aware of a coupon's active and expire date since attempting to use one before or after that date might render it invalid and prevent them from enjoying the discount or deal.

## Customer

- A customer may see an offer printed in **do you have a coupon?** when he adds things to his basket and proceeds to the checkout.

**PickBazar** Create Shop English

- Dashboard
- Shops
- My Shops
- Products
- Attributes
- Groups
- Categories
- Tags
- Manufacturers/Publications
- Authors
- Orders
- Create Order
- Users
- Coupons
- Taxes

1 Customer + Update

Customer

1 Contact Number + Update

19365141641631

2 Billing Address + Add

Billing

2231 Kidd Avenue,  
Kipnuk, AK, 99614,  
United States

**Your Order**

8 x Celery Stick   1lb	\$40.00
3 x Clementines   1lb	\$7.50
Sub Total	\$47.50
Tax	\$0.95
Shipping	\$50.00
Do you have a coupon?	
<b>Total</b>	<b>\$98.45</b>
Wallet points	0
Wallet currency	\$0.00
<input type="checkbox"/> Do you want to use wallet?	

**Choose Payment Method**

Cash

Cash On Delivery

Please pay with cash

**Place Order**

- The consumer now clicks on the offer text and enters a discount code to receive a discount.

**PickBazar** Create Shop English

- Dashboard
- Shops
- My Shops
- Products
- Attributes
- Groups
- Categories
- Tags
- Manufacturers/Publications
- Authors
- Orders
- Create Order
- Users
- Coupons
- Taxes
- Shipplings
- Withdrawals

1 Customer + Update

Customer

1 Contact Number + Update

19365141641631

2 Billing Address + Add

Billing

2231 Kidd Avenue,  
Kipnuk, AK, 99614,  
United States

3 Shipping Address + Add

**Your Order**

8 x Celery Stick   1lb	\$40.00
3 x Clementines   1lb	\$7.50
Sub Total	\$47.50
Tax	\$0.95
Shipping	\$50.00
40OFF <span>Apply</span>	
<b>Total</b>	<b>\$98.45</b>
Wallet points	0
Wallet currency	\$0.00
<input type="checkbox"/> Do you want to use wallet?	

**Choose Payment Method**

Cash

Cash On Delivery

Please pay with cash

**Place Order**

- The discount will now show up and be deducted from the subtotal. and after placing an order, the consumer can take advantage of a discount.

**PickBazar** Create Shop English

- Dashboard
- Shops
- My Shops
- Products
- Attributes
- Groups
- Categories
- Tags
- Manufacturers/Publications
- Authors
- Orders
- Create Order
- Users
- Coupons
- Taxes
- Shipplings
- Withdrawals

1 Customer + Update

Customer

1 Contact Number + Update

19365141641631

2 Billing Address + Add

Billing

2231 Kidd Avenue,  
Kipnuk, AK, 99614,  
United States

3 Shipping Address + Add

**Your Order**

8 x Celery Stick   1lb	\$40.00
3 x Clementines   1lb	\$7.50
Sub Total	\$47.50
Tax	\$0.95
Shipping	\$50.00
Discount (OFF) x	-\$4.00
<b>Total</b>	<b>\$94.45</b>
Wallet points	0
Wallet currency	\$0.00
<input type="checkbox"/> Do you want to use wallet?	

**Choose Payment Method**

Cash

Cash On Delivery

Please pay with cash

**Place Order**

# Find or Search Nearby shops inside PickBazar.

Please follow & complete this steps for Find or Search Nearby shops Enable for your e-commerce system.

- Add Google Map API key inside `.env` file of pickbazar shop.

```
NEXT_PUBLIC_GOOGLE_MAP_API_KEY=
```

## Admin Settings

- First go to settings from PickBazar admin dashboard. Inside settings you will find Enable Google Map Address switch button. Enable the Google Map Address & add value of Maximum Search Location Distance.

**PickBazar** Create Shop

Users  
Coupons  
Taxes  
Shippings  
Withdrawals  
Message  
Refunds  
Questions  
Store Notice  
Reviews  
**Settings**

Add your shop settings information from here

NY State Thruway, New York, USA

Contact Number  
+129290122122

Website  
https://redq.io

☒ Enable Google Map Address

Maximum Search Location Distance(km)

☐ Enable product review system before publish ?

Select social platform Add profile url

Now you can search your nearby shop

**PickBazar** Grocery Find Locations : USA Parkway Shops Offers FAQ Contact Become a Seller Join

CA-1, Long Beach, CA, USA

**Near Shops**

- Clothing Shop**  
4885 Spring Street, Illinois, Lincoln, 62656, USA  
2551.48km Away
- Grocery Shop**  
1986 Spinnaker Lane, Illinois, Freeport, 61032, USA  
3617.69km Away
- Furniture Shop**  
588 Finwood Road, New Jersey, East Dover, 08753, USA  
3811.46km Away
- Bread Shop**  
3819.80km Away
- BEAUTY SALE**  
UP TO 10% OFF  
8309.40km Away
- 20% OFF ALL Backpacks**  
10837.89km Away

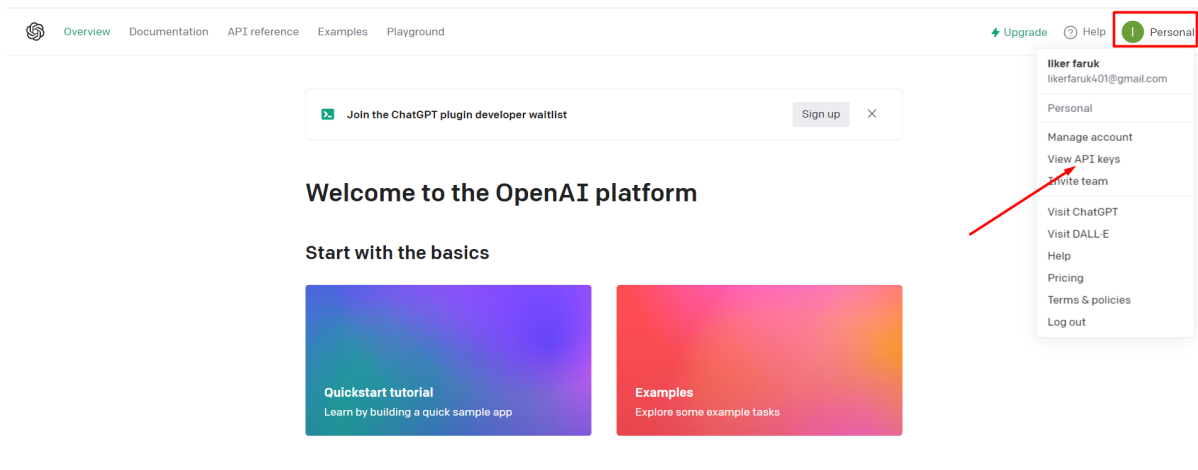
# OpenAI

OpenAI is an artificial intelligence research organization that aims to advance the field of AI in a safe and beneficial way. The organization was founded in 2015 by a group of prominent technology leaders, including Elon Musk, and has since made significant contributions to the field of AI. OpenAI conducts research in areas such as natural language processing, reinforcement learning, computer vision, and robotics. They have developed advanced language models such as GPT-3, which can generate human-like text and carry out various language-related tasks. OpenAI's mission is to ensure that artificial intelligence is developed and used in a responsible and ethical manner to benefit humanity.

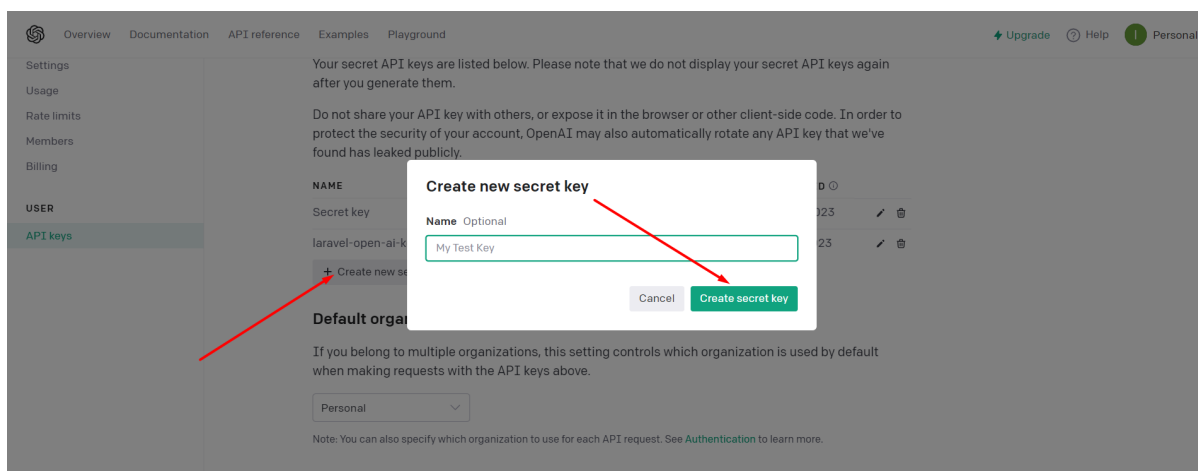
## OpenAI Integration inside PickBazar.

Please follow & complete this steps for OpenAI integration for your e-commerce system.

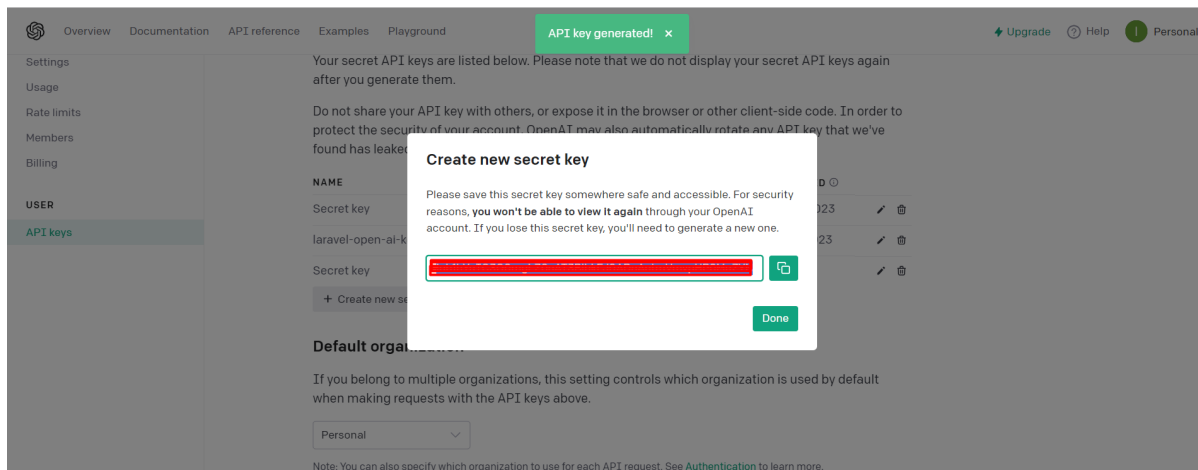
- Go to the [OpenAI Platform Website](#) & login If you aren't a registered user, Complete the OpenAI registration first.
- After logged in into OpenAI dashboard, Click on Personal, then click on View API Keys.



- After that Click on Create new Secret key, then open a modal, type your key name and submit on Create secret key.



- After that copy the generated secret key, & paste on the pickbazar api .env file.



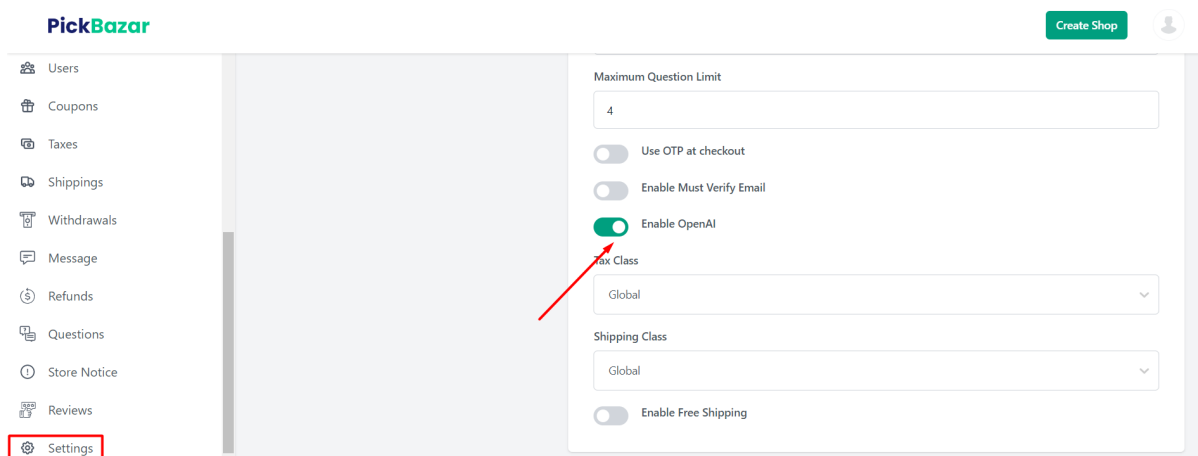
- Pickbazar api .env

```
OPENAI_SECRET_KEY=[YOUR_OPENAI_SECRET_KEY]
```

```
127 # AI --> OpenAi
128 OPENAI_SECRET_KEY=[YOUR_OPENAI_SECRET_KEY]
```

## Admin Settings

- First go to settings from PickBazar admin dashboard. Inside settings you will find Enable OpenAI switch button. Enable the OpneAi.



## Special Notes for Paystack.

If we have used any third party system/plugin/packages, then we have always encouraged our respected customers to follow the official documentation for detailed & in-depth knowledge.

- [OpenAI official website](#)
- [OpenAI official documentation](#)
- [OpenAI API-Reference](#)

# Question

A comprehensive and scalable e-commerce site may be created using Pickbazar, an open-source Laravel-based e-commerce platform. Before making a purchase, shoppers may ask questions about a certain product using Pickbazar's "Question" function.

Customers' worries and questions regarding the product are clarified thanks to this feature, which may eventually result in higher customer satisfaction and fewer product returns. On the backend admin panel, the store owner or administrator may view and respond to the queries.

Customers may enable the "Question" function, specify the minimum number of questions for each product, and score how helpful other customers' or the business owner's responses were. Customers' confidence and credibility in the store and its items are increased thanks to this feature.

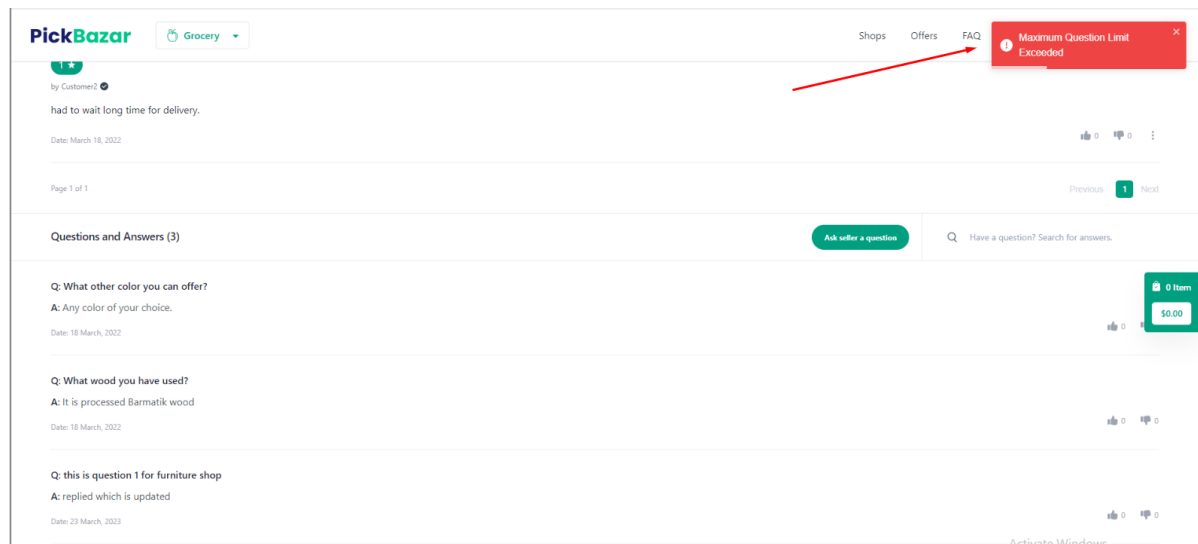
## Super Admin

On the admin dashboard, the super admin will choose **settings** and then select **maximum question limit** to enter a figure. A consumer is only permitted to ask the **maximum question limit** in relation to a certain product. An error warning will show up if he asks more questions than the **maximum question limit**."



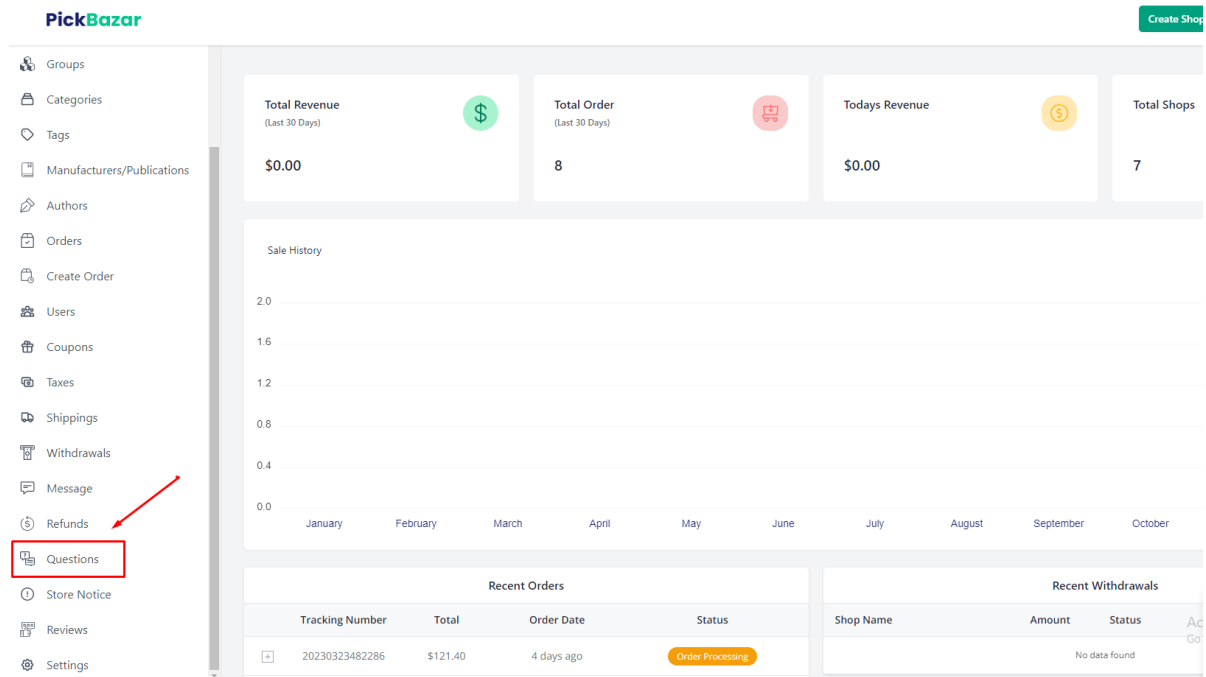
The image shows a settings form for 'Maximum Question Limit'. The label 'Maximum Question Limit' is above a text input field. The input field contains the number '3'.

The administrator set a **maximum question limit** of three, as seen in the image below. In order to prevent further inquiries from the consumer, an error notice will be displayed.



The image is a screenshot of the Pickbazar website. At the top, there's a navigation bar with 'Shops', 'Offers', and 'FAQ' links. A red error banner at the top right says 'Maximum Question Limit Exceeded'. Below the banner, a question is displayed: 'Q: What other color you can offer?'. The answer is 'A: Any color of your choice.' Below this, another question is shown: 'Q: What wood you have used?'. The answer is 'A: It is processed Barmatik wood'. At the bottom, a third question is visible: 'Q: this is question 1 for furniture shop'. The answer is 'A: replied which is updated'. On the right side, there's a green button that says 'Ask seller a question'. At the bottom right, there's a green button that says '0 Item \$0.00'.

On the admin dashboard, choose **question** to provide the customer's inquiry with an answer. Next, in order to respond to the newly posed question, he will click on **reply**.




**PickBazar** Create Shop English

- Dashboard
- Shops
- My Shops
- Products
- Attributes
- Groups
- Categories
- Tags
- Manufacturers/Publications
- Authors
- Orders
- Create Order
- Users

### Questions

Image	Question & Answer	Customer Name	Product Name	Feedbacks	Date	Actions
	Q: this is question 1 for furniture shop A: replied which is updated	Customer	Ash Double Bed	0 likes, 0 replies	4 days ago	
	Q: It showed a lady fitted out with a fur hat and fur boa who sat upright, raising a heavy fur muff that covered the whole of her lower arm towards the viewer. Gregor then turned to look out the window at the dull weather. Drops of rain could be heard hitting the pane, which made him feel quite sad. <a href="#">Reply</a>	Customer	Pacific Natural Shaving Cream	0 likes, 0 replies	9 months ago	
	Q: His room, a proper human room although a little too small, lay peacefully between its four familiar walls. A collection of textile samples lay spread out on the table - Samsa was a travelling salesman - and above it there hung a picture that he had recently cut out of an illustrated magazine and housed in a nice, gilded frame. <a href="#">Reply</a>	Customer	Pacific Natural Shaving Cream	0 likes, 0 replies	9 months ago	

A modal box with a text field for responding will display. The question and answer will both show underneath the store product in the question and answer area when the response has been submitted.



## Pacific Natural Shaving Cream

Product ID: 407


**Q:** It showed a lady fitted out with a fur hat and fur boa who sat upright, raising a heavy fur muff that covered the whole of her lower arm towards the viewer. Gregor then turned to look out the window at the dull weather. Drops of rain could be heard hitting the pane, which made him feel quite sad.

Type your answer here



If admin give answer of this question this question and answer will appear in the shop for a specific product details



Reply

Also, the administrator has the ability to edit, remove, and rate the question using like and dislike.



**Q:** Is it sweet or sour?  
**A:** Its sweet as it contains suger.

Customer      Sun Tropics Organic Mango Nectar,250ml       0    0      a year ago






## Customers

The consumer will visit the specifics of a given product and scroll down to see the FAQ section. He will see the phrase `Ask seller a question` there.

If he selects `Ask Seller a Question`, a modal window will open, allowing him to type and submit the query.



Questions and Answers (3)



 Have a question? Search for answers.



**Q:** What other color you can offer?  
**A:** Any color of your choice.

Date: 18 March, 2022

 0    0

**Q:** What wood you have used?  
**A:** It is processed Barmatik wood

Date: 18 March, 2022

 0    0

0 Item

\$0.00

## Ask seller a question

here write a question to submit

Your question should not contain contact information such as email, phone or external web links.

Submit

## Review

Customers are able to provide comments on things they have purchased using Pickbazar's review function.

Customers may rate a product (often on a scale of 1 to 5) and write a review about it using the review option. Other future buyers who are thinking about buying the same product might find this feedback useful. While it enables them to learn more about how their product is being received by clients and make adjustments as needed, it may also be useful to the vendor or seller.

The Pickbazar review option is a crucial instrument for gaining the respect and trustworthiness of customers. Allowing customers to offer their insights and viewpoints promotes openness and authenticity in the purchasing process.

## Customer

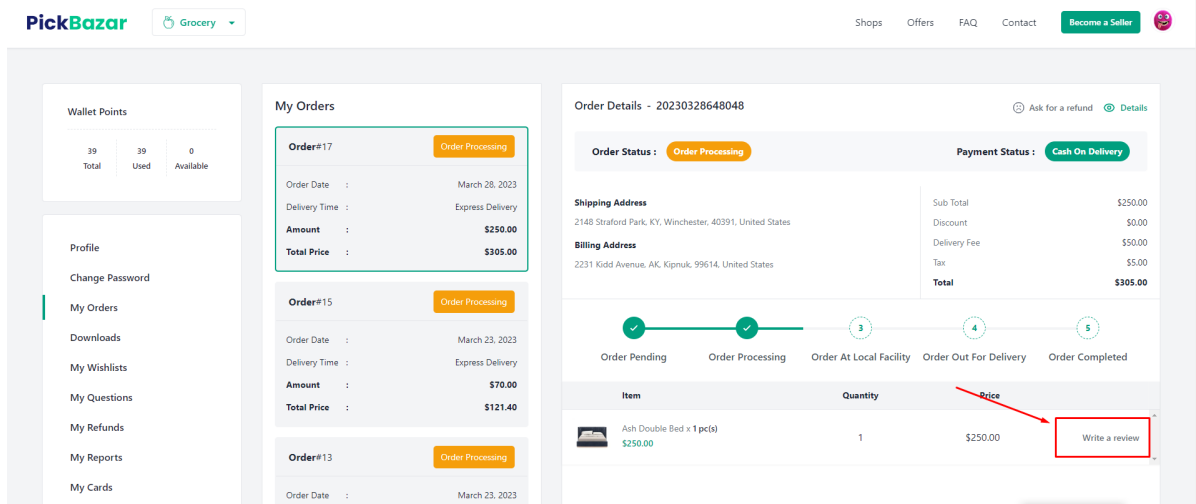
After successfully placing a purchase, the customer will select the profile icon in the top right corner. He must choose **My Orders** from the drop-down option.

The screenshot displays the PickBazar website interface. At the top right, a user profile icon is highlighted with a red arrow. A dropdown menu is open, showing options: Points, Profile, **My Orders** (highlighted with a red box), My Wishlists, Checkout, and Logout. The main content area shows an order processing status page. The order status is 'Order Processing' and the payment status is 'Cash On Delivery'. The order details include:

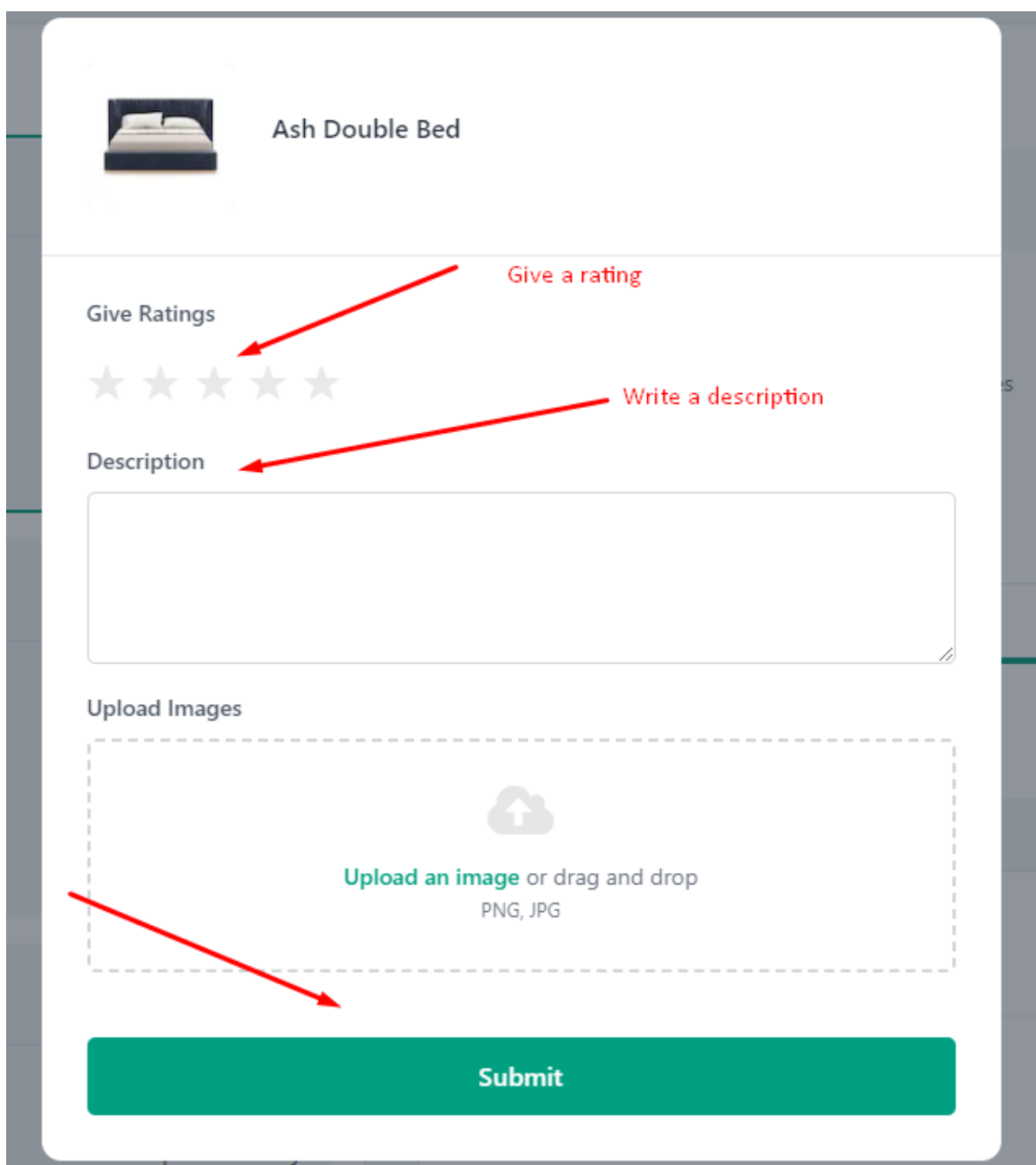
Total Amount		Order Details	
Sub Total	: \$250.00	Name	: Customer
Shipping Charge	: \$50.00	Total Item	: 1 Item
Tax	: \$5.00	Delivery Time	: Express Delivery
Discount	: \$0.00	Shipping Address	: 2148 Straford Park, KY, Winchester, 40391, United States

The order progress bar shows five steps: Order Pending (checked), Order Processing (checked), Order At Local Facility (3), Order Out For Delivery (4), and Order Completed (5).

The consumer will discover a **write a review** option under **My Orders** to the right of each individual order.



After selecting `write a review`, a modal window with a few input fields will show up for you to fill out and publish.



**ratings** : A review rating is a point total or tally that sums up how well a user or customer feels about a certain item, service, or experience. Customer reviews are frequently available on websites, and these evaluations are sometimes rated using a scale, such as a five-star rating system, where the more stars, the better the product or service is thought to be.

**description** : A written review that offers more information about the user's experience and might be useful to other shoppers when making judgments is frequently included with review ratings.

One can amend the review after submitting it by doing the following.

The screenshot shows the PickBazar user dashboard. On the left is a sidebar with navigation links: Profile, Change Password, My Orders, Downloads, My Wishlists, My Questions, My Refunds, My Reports, and My Cards. The main content area is divided into three sections. The 'My Orders' section lists three orders: Order#17 (March 28, 2023, \$305.00), Order#15 (March 23, 2023, \$121.40), and Order#13 (March 23, 2023). The 'Order Details' section for Order#17 shows the status as 'Order Processing' and 'Cash On Delivery'. It includes shipping and billing addresses, a price breakdown (Sub Total: \$250.00, Discount: \$0.00, Delivery Fee: \$50.00, Tax: \$5.00, Total: \$305.00), and a progress bar with five steps: Order Pending, Order Processing, Order At Local Facility, Order Out For Delivery, and Order Completed. Below the progress bar is a table with columns for Item, Quantity, and Price. The first item is 'Ash Double Bed x 1 pc(s)' with a quantity of 1 and a price of \$250.00. A red box highlights an 'Update review' button next to the item.

Each product detail page's review section will contain any recently published reviews.

The screenshot shows the product review section for 'Ash Double Bed'. At the top, it says 'Ratings & Reviews of Ash Double Bed'. Below this is a star rating system with five stars. The first star is selected, indicating a 1-star rating. Below the stars, it says '1 ratings'. To the right of the stars is a table showing the number of ratings for each star: 5 stars (0), 4 stars (0), 3 stars (0), 2 stars (0), and 1 star (1). Below the ratings is a section titled 'Product Reviews (2)'. It shows two reviews. The first review is by 'Customer2' and has a 1-star rating. The text of the review is 'had to wait long time for delivery.' and the date is 'Date: March 18, 2022'. The second review is by 'Customer' and has a 4-star rating. The text of the review is 'this is a demo review description..' and the date is 'Date: March 28, 2023'. Red arrows point to the star ratings of both reviews. On the right side of the reviews, there is a 'Sort By: Recent' dropdown and a 'Filter: All Star' dropdown. At the bottom right, there is a green button that says '0 Item \$0.00'.

## Shipping

When anything needs to be physically moved from an online store to a customer's delivery location, it is referred to as shipping. This mostly entails taking an order, processing it, and getting it ready for delivery.

## Free Shipping

Online companies who provide free delivery do so when shoppers buy products from them. Online sellers encourage customers to buy specific products by reducing delivery fees from those products.

## Cash on delivery

Cash on Delivery (C.O.D.) is a word used frequently in commerce to denote that payment for products must be made at the time of delivery. The main method of payment is cash.

## Shipping and Billing Address

An address selected for shipping is where you will send the order. The address associated with the customer's payment method is known as the billing address. But not usually, billing and shipping addresses are frequently the same.

## Shipping process ingeneral

On Pickbazar, the merchants who are selling their goods on the site are the ones who receive the shipping money. When clients place an order, the vendors charge them shipping fees, which they then keep. The vendors utilize the shipping charges to cover the expenses incurred when sending the customers' orders of the items.

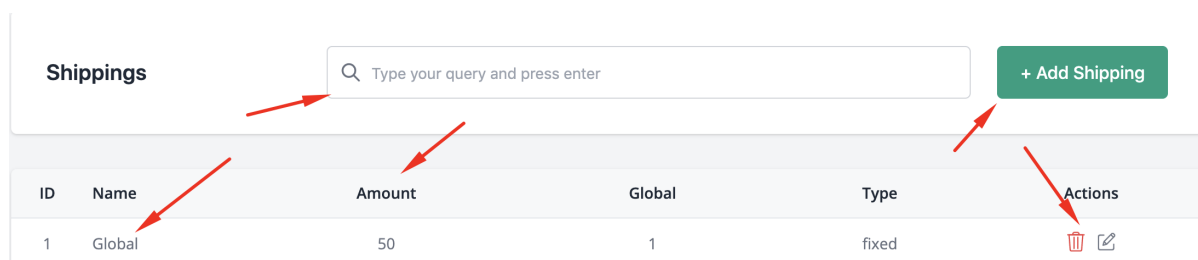
You could get access to the shipping rates and shipping cost information for each seller on the platform if you're a Super Admin. This might assist you in keeping an eye on the shipping charges and fees to make sure they comply with the platform's shipping regulations.



On the shipping charges that the merchants receive through Pickbazar, a commission fee may be assessed. This commission can be used to pay for platform management expenses, customer service expenses, and platform development expenses for new products and services. Depending on the vendor's subscription plan and the number of sales generated via the platform, the commission portion may change.

The platform may impose a commission on these payments to offset the costs of running the platform. Ultimately, the shipping fees are collected by the sellers. As a Super Admin, you would be in charge of keeping an eye on the shipping charges and fees information and making sure that it complies with the platform's rules.

## Super Admin

In `shipping` menu you will get the shipping list and you can add, remove or modify shipping from there.



Shippings					
Q Type your query and press enter					
ID	Name	Amount	Global	Type	Actions
1	Global	50	1	fixed	 

Once you click on "Add Shipping," the following form will display for you to fill out.

Create New Shipping

Description

Add your shipping description and necessary information from here

Name

Type

☐ Free
 ☒ Fixed
 ☐ Percentage

Amount

Add Shipping

Shippings

+ Add Shipping

ID	Name	Amount	Global	Type	Actions
1	Global	50	1	fixed	

The shipment name, quantity, and type are necessary. Once added, this will show up on the shipment list.

The administrator will then pick the shipping class name from the dropdown menu under "settings" after that. Also, he might provide the merchant the option of free shipment.

Shipping Class

☒ Enable Free Shipping

Shippings

+ Add Shipping

ID	Name	Amount	Global	Type	Actions
1	Global	50	1	fixed	

## Vendor

Upon the purchase of the product, the seller will receive the shipping cost together with the subtotal.

## Taxes

Like any other company, Pickbazar must pay taxes to the government. The government uses taxes as a means of raising money to support public infrastructure and services like roads, schools, and hospitals.

Pickbazar is an online marketplace that links buyers and sellers so that anyone may purchase and sell goods. The merchant makes money when a consumer uses Pickbazar to buy something, and as a result, is obligated to pay taxes on that money.

# Super Admin

In **Tax** menu you will get the Tax list and you can add, remove or modify Tax from there.

Taxes

+ Add Tax

ID	Name	Rate (%)	Country	City	State	ZIP	Actions
1	Global	2					

To add a new tax, the super administrator must now specify its name and value.

Create New Tax

Information  
Add your tax information from here

Name

new tax

Rate

10

Country

City

State

ZIP

Add Tax

The list of taxes will display the newly established tax.

Taxes

+ Add Tax

ID	Name	Rate (%)	Country	City	State	ZIP	Actions
2	new tax	10					
1	Global	2					

He may access the settings from the admin dashboard and configure the tax he wants to enable.

PickBazar

Dashboard

Shops

My Shops

Products

Attributes

Groups

Categories

Tags

Manufacturers/Publications

Authors

Orders

Create Order

Users

Coupons

Taxes

Shippings

Withdrawals

Message

Refunds

Questions

Store Notice

Reviews

Settings

Site Subtitle

Your next ecommerce

Currency

US Dollar

Minimum Order Amount

0

Wallet Currency Ratio

3

Sign Up Points

100

Maximum Question Limit

5

Use OTP at checkout

Tax Class

new tax

Shipping Class

Global

Enable Free Shipping

Enable Cash On Delivery

Select Payment Gateway

Stripe

Payment

Configure Payment Option

Tax Class

new tax

## Vendor

When a consumer adds items to his cart and checks out, he may discover that the tax has been added to the subtotal.

In order to place an order, the consumer must now click "place order."

PickBazar

Create Shop

English

Dashboard

Shops

My Shops

Products

Attributes

Groups

Categories

Tags

Manufacturers/Publications

Authors

Orders

Create Order

Users

Coupons

1 Customer

Customer

+ Update

1 Contact Number

19365141641631

+ Update

2 Billing Address

Billing

2231 Kidd Avenue,  
Kipruk, AK, 99614,  
United States

+ Add

Your Order

8 x Celery Stick | 1lb

\$40.00

3 x Clementines | 1lb

\$7.50

Sub Total

\$47.50

Tax

Calculated at checkout

Estimated Shipping

Calculated at checkout

Check Availability

**PickBazar** Create Shop English

**1 Customer** + Update

Customer

**1 Contact Number** + Update

19365141641631

**2 Billing Address** + Add

Billing  
2231 Kidd Avenue,  
Kipmuk, AK, 99614,  
United States

**Your Order**

8 x Celery Stick   1lb	\$40.00
3 x Clementines   1lb	\$7.50
Sub Total	\$47.50
Tax	\$0.95
Shipping	\$50.00
Discount (40FF) x	- \$4.00
<b>Total</b>	<b>\$94.45</b>
Wallet points	0
Wallet currency	\$0.00

☐ Do you want to use wallet?

**Choose Payment Method**

Please pay with cash

**Place Order**

# Withdrawals

The procedure by which a seller can move their profits or balance from the Pickbazar platform to their own bank account or another payment method is known as a withdrawal mechanism in Pickbazar.

In most withdrawal systems, the seller must ask to remove money from their account balance. With the seller's identification being confirmed, the platform may impose conditions regarding the minimum withdrawal amounts or withdrawal costs.

The platform will transfer the money to the seller's chosen payment method as soon as the withdrawal is accepted. The payment method and the platform's rules will determine how long this procedure will take, which might be several business days.

The ability for sellers to obtain their revenues and use them for either personal or professional objectives makes withdrawal mechanisms an essential component.

# Vendor

The store owner must first log in, then from the dashboard, we must locate "my shops". The seller's shop is where he may locate withdrawal options. The options below will be displayed if he visits his own store.

**PickBazar** Create Shop

**Store Owner**  
store\_owner@demo.com  
12365141641631  
☒ Enabled

**My Shops** **Message** **Store Notice**

**Furniture Shop** Active

**Clothing Shop** Active

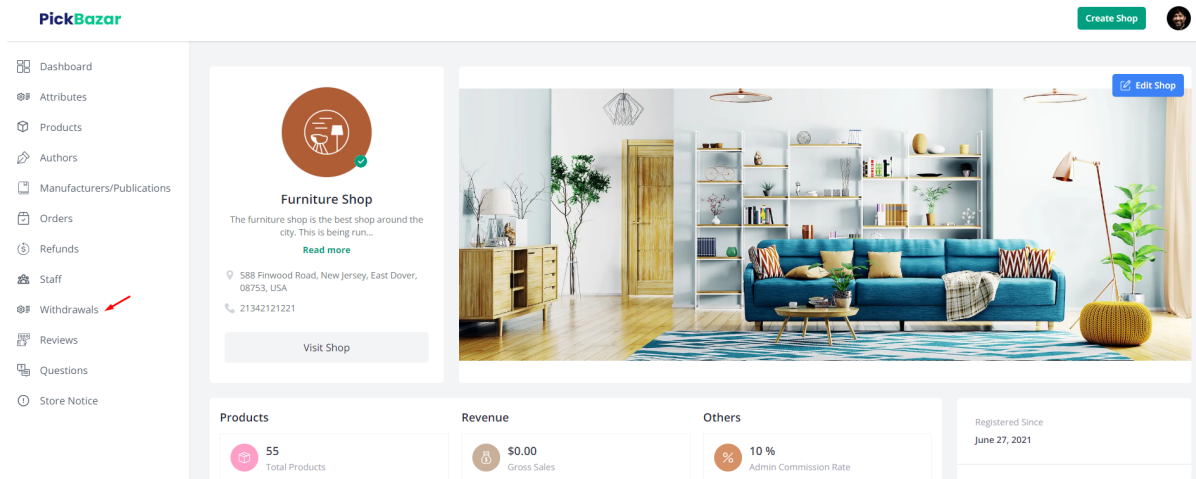
**Bags Shop** Active

**Makeup Shop** Inactive

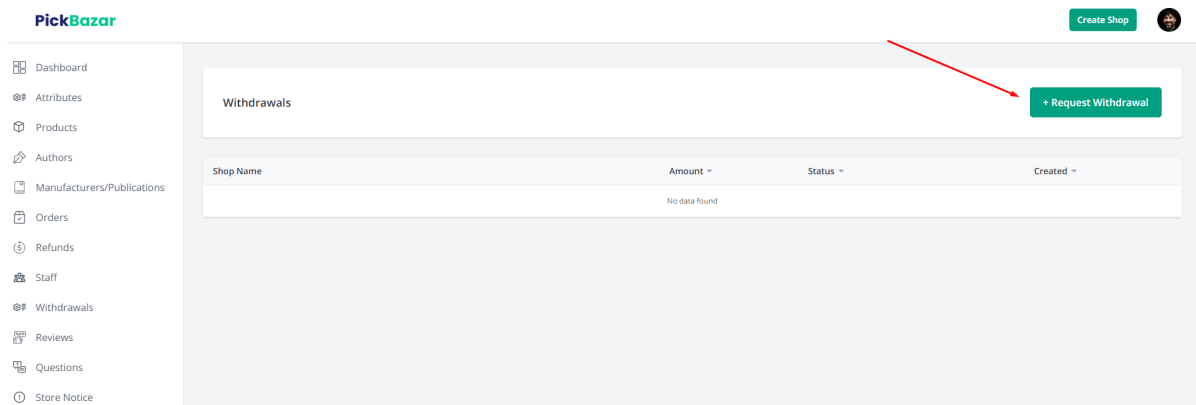
**Bakery Shop** Active

**Grocery Shop** Inactive

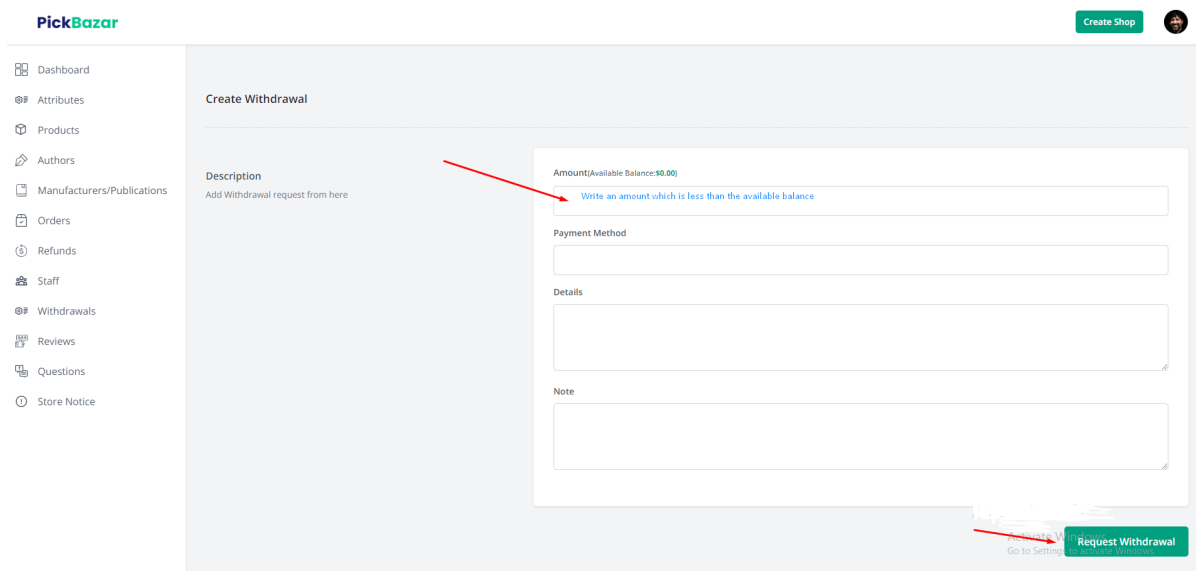
**Books Shop** Active



The seller may submit withdrawal requests under the withdrawals area.



The seller can select **request withdrawals** and enter a request for a lesser quantity than what is now available.



## Super Admin

Requested Withdrawals will show up in the "recent Withdrawals" section of the admin dashboard. The administrator can use the action function and change the withdrawal request's status from pending to approved by clicking on **change status** button..

PickBazar

Create Shop

Dashboard

Shops

My Shops

Products

Attributes

Groups

Categories

Tags

Manufacturers/Publications

Authors

Orders

Create Order

Users

Coupons

Taxes

0.8

0.4

0.0

January

February

March

April

May

June

July

August

September

October

November

December

Recent Orders

Tracking Number	Total	Order Date	Status
20230314160616	\$53.67	an hour ago	Order Processing

Recent Withdrawals

Shop Name	Amount	Status	Created	Actions
No data found				

Popular Products

ID	Name	Group	Shop	Price/Unit	Quantity
2	Baby Spinach	Grocery	Grocery Shop	\$0.60	10
3	Blueberries	Grocery	Grocery Shop	\$3.00	30
239	Simply Natural Fruits And Nuts Meal Pack	Grocery	Grocery Shop	\$7.00	50
249	Borges Pasta	Grocery	Grocery Shop	\$15.00	50
240	Ivory Cashew Nuts	Grocery	Grocery Shop	\$7.50	50

Shop Name	Amount	Status	Created	Actions
Chawkbazar Vendor shop	\$500.00	Approved	a year ago	
Chawkbazar Vendor shop	\$250.00	On hold	a year ago	
Chawkbazar Vendor shop	\$6,500.00	Rejected	a year ago	
Chawkbazar Vendor shop	\$600.00	On hold	a year ago	
Chawkbazar Vendor shop	\$10.00	Pending	a few seconds ago	

Shop Name	Amount	Status	Created	Actions
Chawkbazar Vendor shop	\$500.00	Approved	a year ago	
Chawkbazar Vendor shop	\$250.00	On hold	a year ago	
Chawkbazar Vendor shop	\$6,500.00	Rejected	a year ago	
Chawkbazar Vendor shop	\$600.00	On hold	a year ago	
Chawkbazar Vendor shop	\$10.00	Pending	a minute ago	

Withdrawal Information

Amount : 10

Payment Method : N/A

Status : Pending

Pending

Change Status

Withdrawal Information

Amount	:	<input type="text" value="10"/>	<div>Approved</div>	<div>Change Status</div>
Payment Method	:	N/A		
Status	:	Pending		