

Welcome to Chawkbazar

Welcome to the ChawkBazar Laravel documentation! Fastest Ecommerce App built with React, NextJS, TypeScript, Laravel and Tailwind CSS.

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 Hamberger in the top right corner.

Supported platforms & ecosystem for local development

- Compatible Browsers (Firefox, Safari, Chrome, Edge)
- Node.js 16.15.0 or later
- PHP 8.1
- MySQL 8
- MacOS, Windows, and Linux are supported

Requirements

- Node.js 16.15.0 or later
- yarn
- PHP 8.1
- MySQL 8
- ext_curl
- ext_gmp
- editor: [Visual Studio Code](#)(recommended)

Special Notice ****

One new chapter named **Upgrade** was introduced along with some sub-chapters for easy migration process of app & server ecosystem. Please visit this chapter each time to check if any new changes are introduced. [Breaking changes](#)

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



How it works

Tech We Have Used

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

- [api](#)
- [shop](#)
- [admin](#)

Tech specification for specific part is given below:

Admin Dashboard (GraphQL & REST)

- [NextJs](#)
- [Typescript](#)
- [Tailwindcss](#)
- [React Hook Form](#)
- [React-Query](#)

Shop Frontend (GraphQL & REST)

- [NextJs](#)
- [Typescript](#)
- [Tailwindcss](#)
- [React Hook Form](#)
- [React-Query](#)

API

- [Laravel](#)

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 Video

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



Prerequisites

- PHP 8.1
- Composer
- Xamp/Wamp/Lamp/Mamp 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 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

```
chawkbazar
|-- chawkbazar-api
|-- admin
|-- shop
```

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

Getting started with api

- Run and start xamp/wamp/lamp/mamp and start all the services
- 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=chawkbazar_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: your frontend `NEXT_PUBLIC_REST_API_ENDPOINT` env value will be `localhost:8000/`

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

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

Video



- Run Docker application first
- Now go to your chawkbazar-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_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

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

```
cd admin/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 chawkbazar/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 chawkbazar directory, you can run the below commands)

```
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
# 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

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

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

```
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`)

```
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 chawkbazar directory, you can run the below commands)

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

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

Shop (At the root of the chawkbazar directory, you can run the below commands)

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

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

```
► 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                                    2.74 kB      179 kB
├ • /bakery (ISR: 60 Seconds)              2.66 kB      243 kB
├   └ css/0dcfbd7860d56385c547.css         4.01 kB
├ λ /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
├ λ /order                                15.4 kB      209 kB
├   └ css/db04274d0e1caa11303a.css         351 B
├ λ /order-received/[tracking_number]      6.85 kB      186 kB
├ λ /orders                                28.5 kB      208 kB
├   └ css/da2daebf602ff72a3fc0.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]
├ λ /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/03597eb9dee95bb8e6991ab21a5b170a1bc330ab.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.173fc5.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/669ac54f0d96e08d6ac5.css        4.38 kB
├   └ css/f53121b36d6036aac827.css       11.2 kB
├ λ (Server) server-side renders at runtime (uses getInitialProps or getServerSideProps)
├ ○ (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.

7. 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.

8. For further development & customization check our [Frontend Customization](#) guide.

Upgrade

ChawkBazar Update - Virutal Private Server

If you follow this [Virtual Private Server](#) docs to host your site, then follow this documentation to update ChawkBazar 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 [chawkbazar](#) directory on your server,

```
cd /var/www/chawkbazar
```

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 chawkbazar
```


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,


The screenshot shows the GitHub 'Create a new repository' page. It includes a search bar at the top, navigation links for Pull requests, Issues, Marketplace, and Explore. The main form has fields for Repository name, Description (optional), and a choice between Public and Private visibility. Below these are checkboxes for 'Add a README file', 'Add .gitignore', and 'Choose a license'. A green 'Create repository' button is at the bottom. Numbered annotations (1-6) point to specific elements: 1 points to the top right corner, 2 points to the 'New repository' dropdown menu, 3 points to the 'Repository name' field, 4 points to the 'Description' field, 5 points to the 'Private' radio button, and 6 points to the 'Create repository' button.

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**,

Quick setup — if you've done this kind of thing before

 Set up in Desktop or **HTTPS** **SSH** `https://github.com/ /chawkbazar-update.git`

Get started by [creating a new repository](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


1 

...or create a new repository on the command line

```
echo "# chawkbazar-update" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ /chawkbazar-update.git
git push -u origin main
```

...or push an existing repository from the command line


```
git remote add origin https://github.com/ /chawkbazar-update.git
git branch -M main
git push -u origin main
```

2 

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

 **ProTip!** Use the URL for this page when adding GitHub as a remote.

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 **chawkbazar** branch

```
git checkout -b chawkbazar
```

5. replace **repository** file with downloaded **chawkbazar-laravel** folder content.
6. Then use this command to add all files to git

```
git add .
```

```
git commit -m "Update ChawkBazar API"
```

7. Merge new code with **main** branch

```
git checkout -b main
```

```
git merge chawkbazar
```


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/chawkbazar` 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: FrontEnd 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 const withPWA = require("next-pwa");
2 const runtimeCaching = require("next-pwa/cache");
3 const { i18n } = require("./next-i18next.config");
4
5 module.exports = withPWA({
6   basePath: "/admin",
7   i18n,
8   pwa: {
9     disable: process.env.NODE_ENV === "development",
10    dest: "public",
11    runtimeCaching,
12  },
13
14  images: {
15    domains: [
16      "via.placeholder.com",
17      "res.cloudinary.com",
18      "s3.amazonaws.com",
19      "18.141.64.26",
20      "127.0.0.1",
21      "localhost",
22      "picsum.photos",
23      "pickbazar-sail.test",
24      "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
25      "chawkbazarlaravel.s3.ap-southeast-1.amazonaws.com",
26      "lh3.googleusercontent.com",
27    ],
28  },
```

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

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

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 `chawkbazar` 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.

Server migration guide

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 `chawkbazar-laravel` -> `chawkbazar-api` folder.

```
cd /var/www/chawkbazar-laravel/chawkbazar-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-gd php8.0-curl php8.0-zip  
php8.0-gmp
```

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===  
'e21205b207c3ff031906575712edab6f13eb0b361f2085f1f1237b7126d785e826a450292b6cfd1d64d92e6563bbde02') { 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/chawkbazar
```

```

# 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/chawkbazar-laravel/chawkbazar-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

```

Laravel version upgrade

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

At first, remove all the existing `php 7.4` 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-gd php8.0-curl php8.0-zip
php8.0-gmp composer
```

Then remove the composer,

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

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

```
cd /var/www/chawkbazar-laravel/chawkbazar-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') ===
'e21205b207c3ff031906575712edab6f13eb0b361f2085f1f1237b7126d785e826a450292b6cfd1d64d92e6563bbde02') { 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/chawkbazar
```

```

# For API
location /backend {
    alias /var/www/chawkbazar-laravel/chawkbazar-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;
}

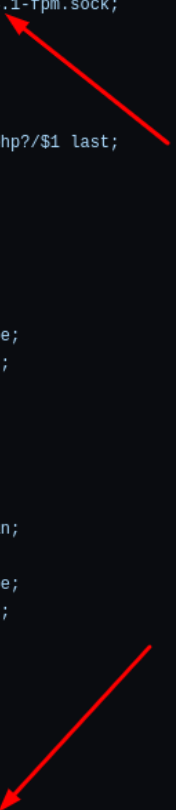
# For FrontEnd
location /{
    proxy_pass http://localhost:3003;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}

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;

```



Then install composer packages,

```
cd /var/www/chawkbazar-laravel/chawkbazar-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
```

Breaking changes

In this chapter the breaking changes will be listed here version wise sequentially. So that our customers can easily track down the changes & take necessary steps during their migration process.

Version 2.2.2 to 3.0.0 update

- Laravel version 8 to 9 upgradation process was introduced. [Laravel 8 to 9 upgrade](#)
- There are changes coming in database section. Before update & run any database migration command, please keep the database backup.
 - Order table was updated in the database.
 - Order-Status table was deleted. Now, order statuses are managed dynamically
- Before update please keep your existing code backup. Because there are changes inside order code architecture & payment gateway integration system.
- Product inventory system was managed based on new order & payment status.

Admin dashboard

- Analytics Dashboard

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



Recent Orders

Tracking Number	Total	Order Date	Status
GkszijsVAcae	\$2,744.33	5 days ago	Order Received
Irrdghqshkys	\$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
TfzYmld1AR32	\$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



















• Manage Product Type

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

Types

Type your query and press enter

+ Add Types

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	Actions
8	Women				women	
7	Watch				watch	
6	Sunglass				sunglass	
5	Sports				sports	
4	Sneakers				sneakers	
3	Men				men	
2	Kids				kids	
1	Bags				bags	

<

1

>

• Product Management

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

Products

Type your query and press enter

+ Add Product

Filter


Image	Name	Brand	Product Type	Price/Unit	Quantity	Status	Actions
	Zara Monte Carlo	Fania Fashion	variable	\$90.00 - \$100.00	240	<div>publish</div>	
	Zara Miss Chase	Vintgae	simple	\$100.00	120	<div>publish</div>	
	White Oxford Shirt	Fusion	variable	\$35.00 - \$40.00	80	<div>publish</div>	
	The Horse Original	Hoppister	simple	\$200.00	250	<div>publish</div>	
	Roadster Women Round Neck	Hairstore Vintage	variable	\$150.00 - \$200.00	200	<div>publish</div>	
	Reyban Havana Phantos Sunglasses	Hipster	simple	\$100.00	50	<div>publish</div>	
	Phillip Lim Leather Shoulder Bag	Phonix Bags	simple	\$260.00	100	<div>publish</div>	
	Nike Pro Mesh Top with Leggings	Vintgae	variable	\$30.00 - \$35.00	100	<div>publish</div>	
	Nike Comfy Vapor Maxpro	AB Shoes	variable	\$220.00 - \$250.00	2000	<div>publish</div>	

A portion of product form

Gallery

Upload your product image gallery here

Upload an image or drag and drop
PNG, JPG



Brand & Categories

Select product group and categories from here

Brand*

Fania Fashion

Categories

Kids x

Tags

Gift Collection x Kids Collection x

Description

Edit your product description and necessary information from here

Name*

Zara Monte Carlo

• Order Status

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

Dashboard

Shops

My Shops

Products

Attributes

Brands

Categories

Tags

Orders

Order Status

Users

Coupons

Taxes

Shippings





















Withdraws

Settings

Order Status

Q Type your query and press enter

+ Add Order Status

ID	Name	Serial	Actions
7	Delivered	10	 
9	Failed to Contact Consignee	9	 
8	Failed To collect payment	8	 
10	Shipment Refused By Consignee	7	 
6	Out For Delivery	6	 
5	At Local Facility	5	 
4	Order Dispatched	4	 
3	Ready to Dispatch	3	 
2	Order Processing	2	 
1	Order Received	1	 

<

1

>

Order Management

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

Orders

Type your query and press enter

Tracking Number	Delivery Fee	Total	Order Date	Status	Shipping Address	Download	Actions
<div>+ a1bKQKf4na1K</div>	\$50.00	\$243.80	2 hours ago	Order Received	3886 Froe Street, Mill Creek, WV, 26280, United States	Download	<div></div>
<div>+ 0DGIGYdkMSw8</div>	\$50.00	\$2,804.00	2 days ago	Order Received	f, fsdfs, fsdf, fsdf	Download	<div></div>
<div>+ 4yQV4m38mbBe</div>	\$50.00	\$845.60	3 days ago	Order Received	Cumilla, Cumilla, Cumilla, 1600, Cumilla	Download	<div></div>
<div>+ lZndjRHpKCc3</div>	\$50.00	\$1,136.30	5 days ago	Order Received	5/1-A, SBC Lane, Holiday Park, Gandaria, Dhaka, Dhupkhola maath, Pukur par, Dhaka, Dhaka, 1204, Bangladesh	Download	<div></div>
<div>+ PGCb83dPpUc3</div>	\$50.00	\$4,181.00	5 days ago	Order Received	5/1-A, SBC Lane, Holiday Park, Gandaria, Dhaka, Dhupkhola maath, Pukur par, Dhaka, Dhaka, 1204, Bangladesh	Download	<div></div>

<

1

>

A portion of order management. Order status change.

Order ID - PGCb83dPpUc3

Order Received

Order Processing

Ready To Ship

At Local Facility

2

3

5

Select...

Order Received

Order Processing

Ready To Dispatch

Order Dispatched

At Local Facility

Out For Delivery

Change Status

Products

Tay Ben Aviator x 3

Sub total\$4,050.00

Tax\$81.00

Delivery fee\$50.00

Discount\$0.00

Total\$4,181.00

Billing Address

Shipping Address

Coupon Management

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

22 / 169

Coupons

Q Type your query and press enter

+ Add Coupon

ID	Banner	Code	Amount	Active	Expired	Actions
10		20OFF	\$20.00	22 days ago	22 days ago	
9		CHRISTMAS18	\$18.00	22 days ago	in a month	
8		SUMMER15	\$15.00	22 days ago	in 3 months	
7		12OFF	\$12.00	22 days ago	in 14 days	
6		WINTER10	\$10.00	22 days ago	in 2 months	
5		SUMMER8	\$8.00	22 days ago	in a month	
4		6OFF	\$6.00	22 days ago	in 2 months	
3		RAMADAN5	\$5.00	22 days ago	in a month	
2		4OFF	\$4.00	22 days ago	in a month	

Customer Management

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

Customers

Q Type your query and press enter

+ Add Customer

Avatar	Name	Email	Status	Actions
	Vendor	vendor@demo.com	Active	
	Admin	admin@demo.com	Active	

Block Customer

Are you sure you want to block this customer?

Cancel

Block

Tax Management

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

Taxes

Q Type your query and press enter

+ Add Tax

ID	Name	Rate (%)	Country	City	State	ZIP	Actions
1	Global	2					<div><div></div><div></div></div>

Shipping Management

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

+ Add Shipping

ID	Name	Amount	Global	Shipping Type	Actions
1	Global	50	1	fixed	<div><div></div><div></div></div>

Shipping Management

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

+ Add Shipping

ID	Name	Amount	Global	Shipping Type	Actions
1	Global	50	1	fixed	<div><div></div><div></div></div>


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 128x40 Pixel





Upload an image or drag and drop
PNG, JPG

Information

Change your site information from here

Site Title

Site Subtitle

Currency

US Dollar

Minimum Order Amount

Tax Class

Global

Shipping Class

Global

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",
"dev:shop-rest": "yarn workspace @chawkbazar/shop dev",
"build:shop-rest": "yarn workspace @chawkbazar/shop build",
"start:shop-rest": "yarn workspace @chawkbazar/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",
"prepare": "husky install",
```

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

Customization

For customizing the template's default site settings:

> **[your-frontend-project]** = admin **[rest]** or shop

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

```
[your-frontend-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 `[your-frontend-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:

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

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

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

For tailwindcss documentation:

Go to [Tailwindcss](#)

Icons:

for our icons

```
open [your-frontend-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 reffernce you can see one of our icon.)
5. If your custom SVG code has any single custom color then change them into fillColor.

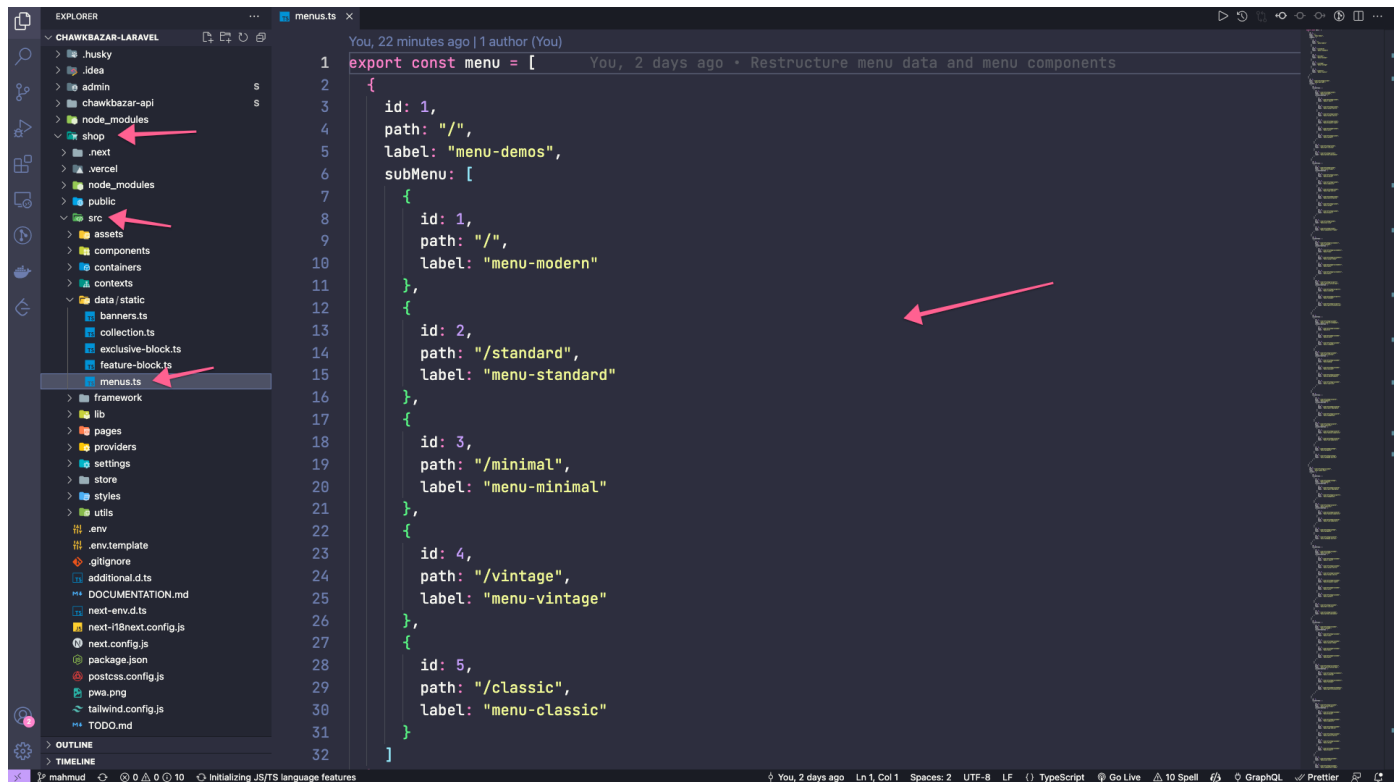
Menu

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

https://www.youtube.com/watch?v=ewNGG_i3r-Y&feature=emb_title

Mega menu for the **shop** are coming from a JSON file which is located at,

```
shop -> public -> data -> menus.json
```

You've to update this JSON to build the mega menu.

Data Structure

id -> ID for menu or submenu
path -> After clicking the menu in frontend which path will be redirect
label -> Label of that menu
subMenu -> JSON for submenu
mobileMenu -> Different menu for mobile.

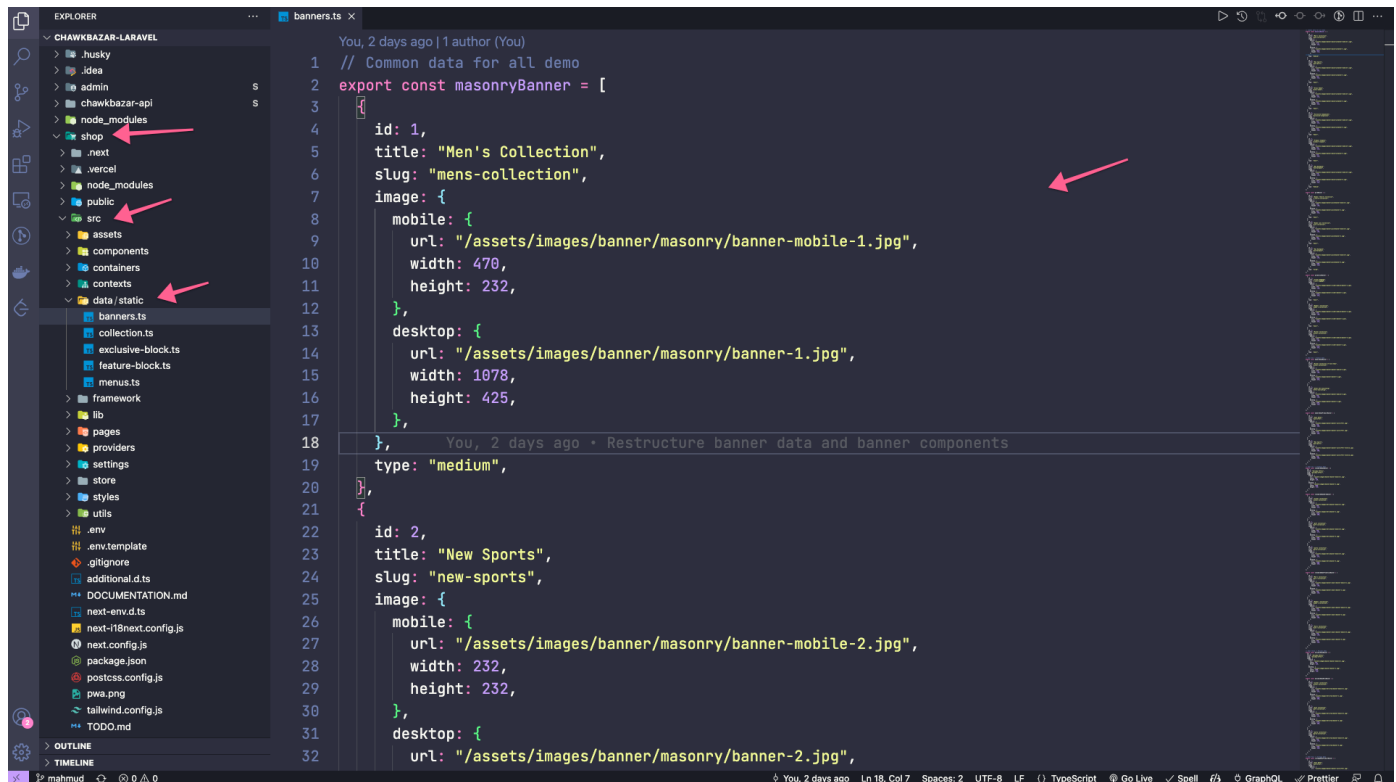
Banner

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

<https://www.youtube.com/watch?v=AM0exuHTKa0>

The banner data are coming from a JSON file which is located at,

shop -> public -> data -> banner.json



You've to update this JSON to build the mega menu.

Data Structure

id -> ID for a banner
 title -> Title for a banner
 slug -> This is a tag slug

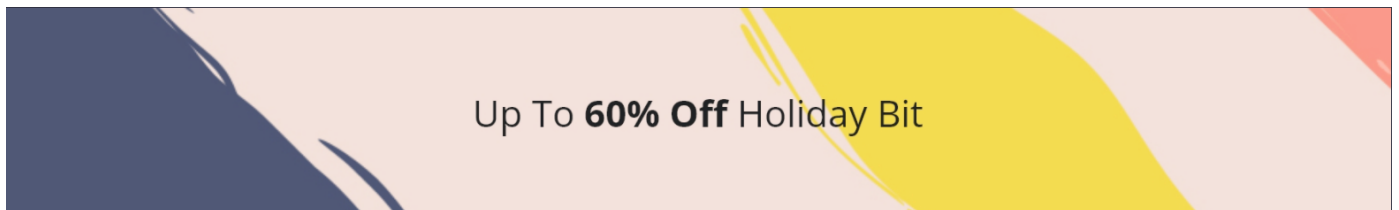
At first, create the tag and add that tag slug on the **slug** path.

For example, if you want to start a special campaign for some products, just create a tag for that campaign in admin, add that tag to their related products, and on **banners.json**, add that tag as a slug.

image -> Image path
 mobile -> JSON object for mobile image
 desktop -> JSON object for desktop image
 url -> Path of the image
 width -> Width of the image for next/image optimization
 height -> Height of the image for next/image optimization

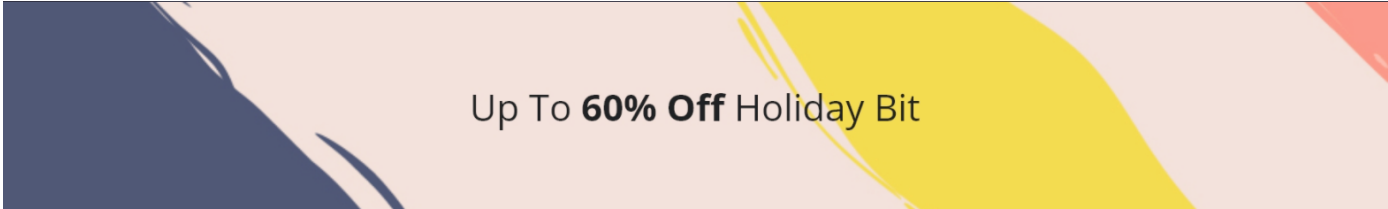
Data Mapping

homeOneBanner



homeOneHeroBanner

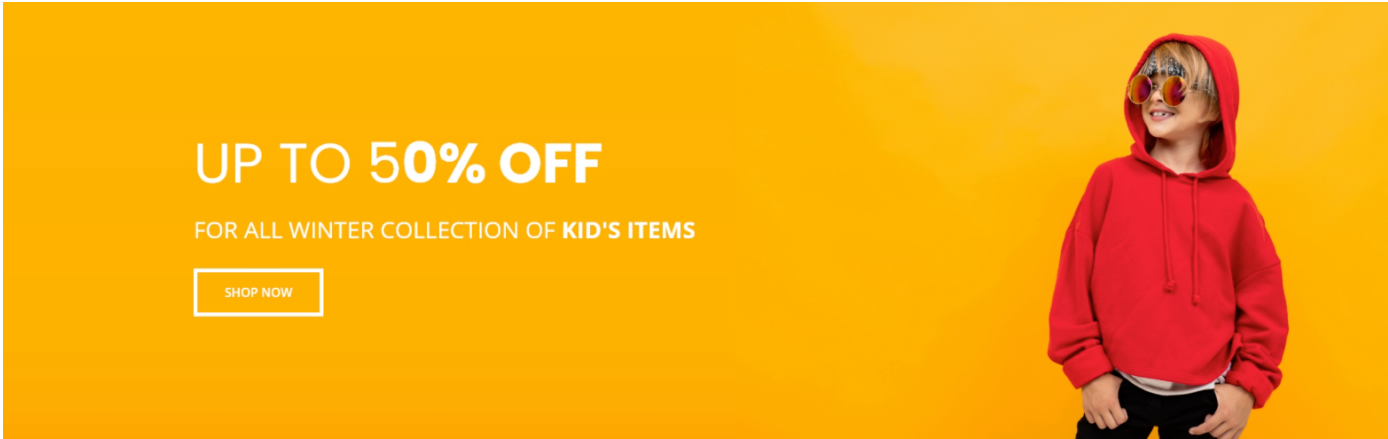
[1]



[2]

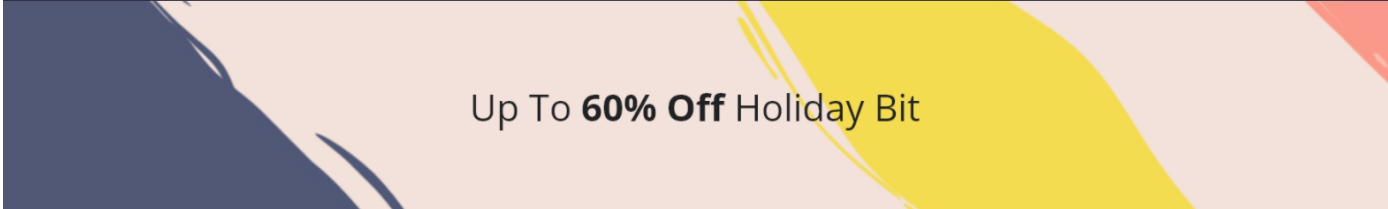


[3]



homeTwoHeroBanner

[1]



[2]



[3]



homeThreeBanner

[1]

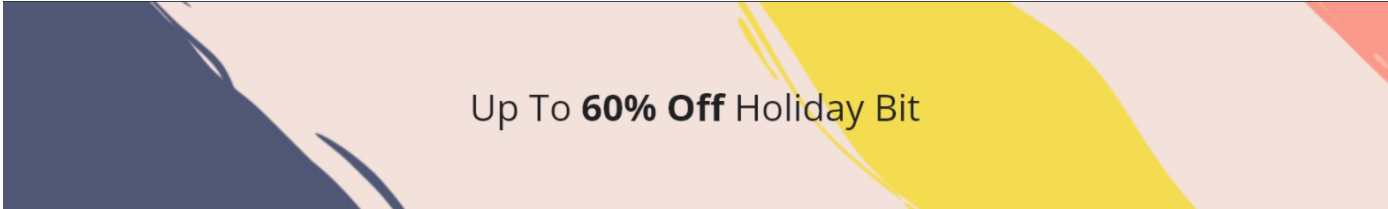


[2]



homeThreeMasonryBanner

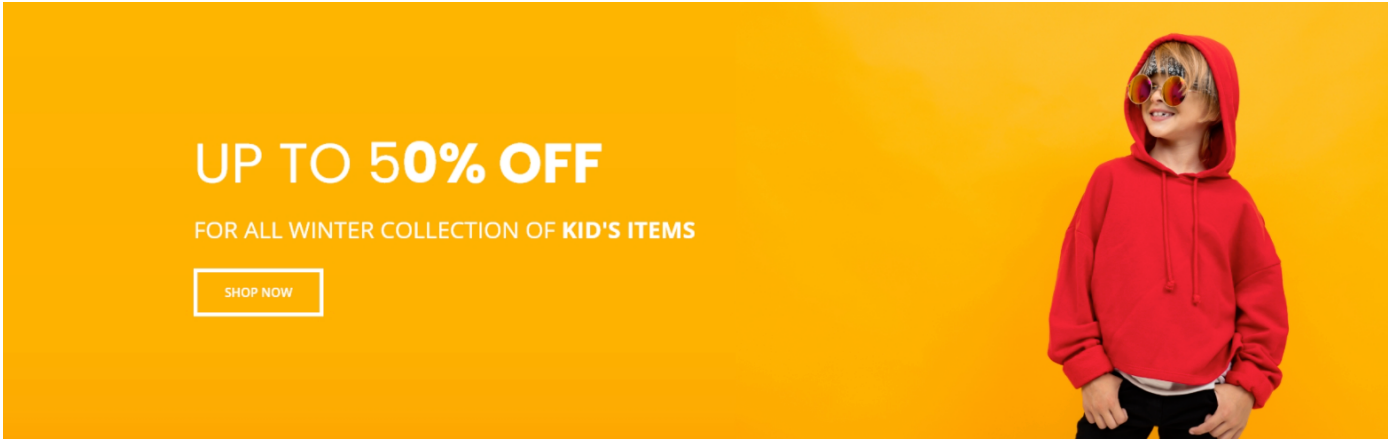
[1]



[2]



[3]



[4]

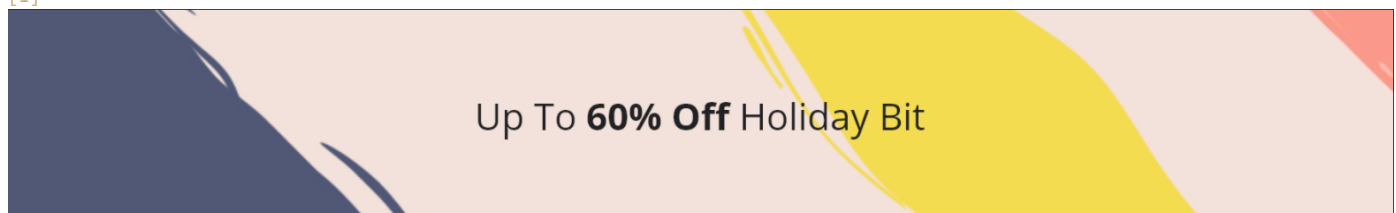


[5]

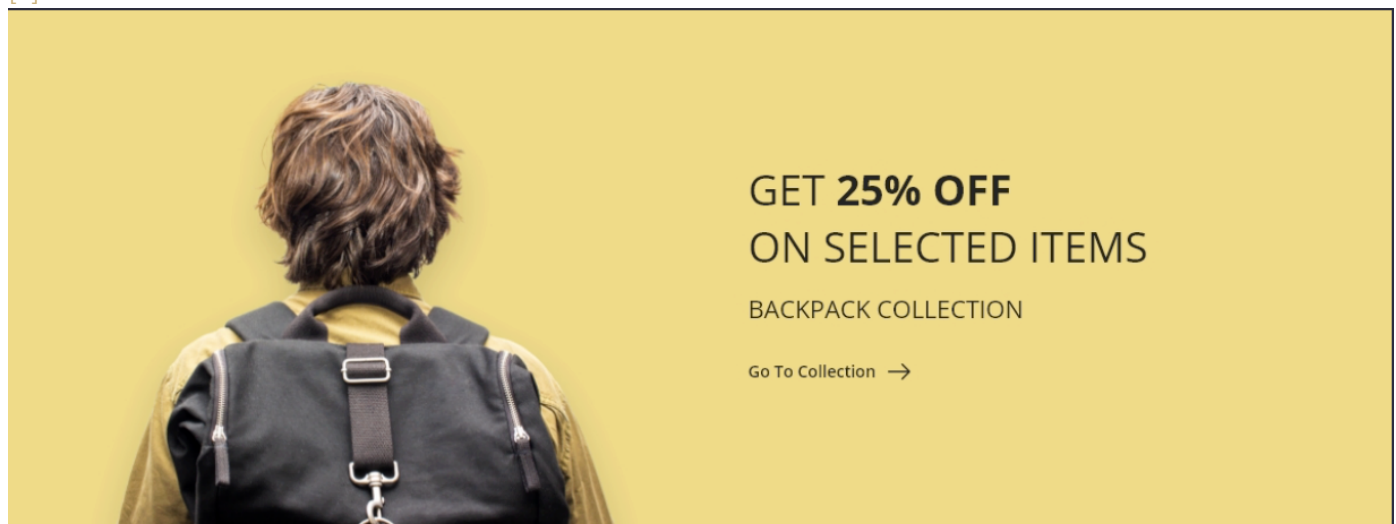


bannerGrid

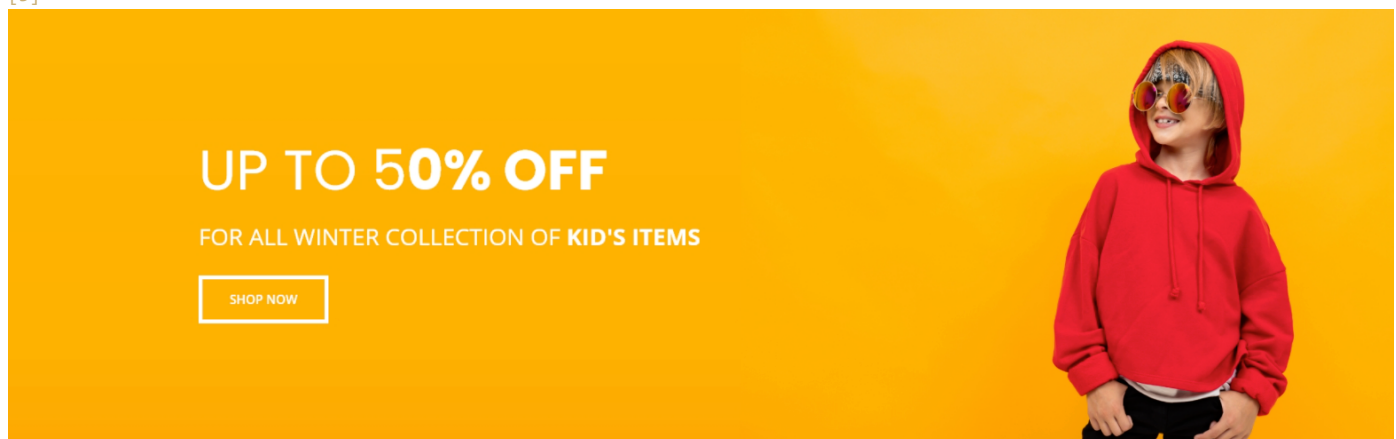
[1]



[2]

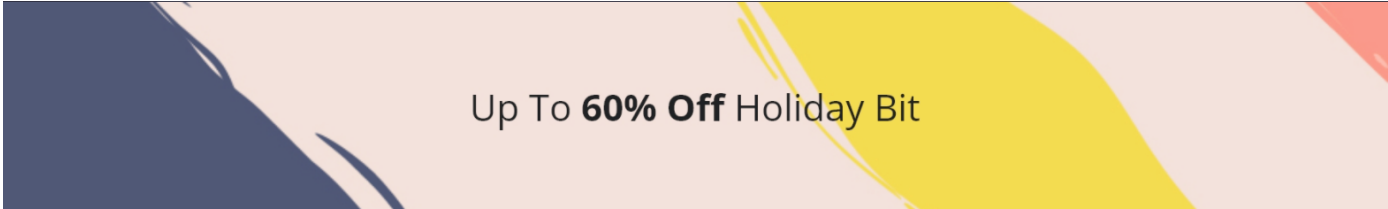


[3]



promotionBanner

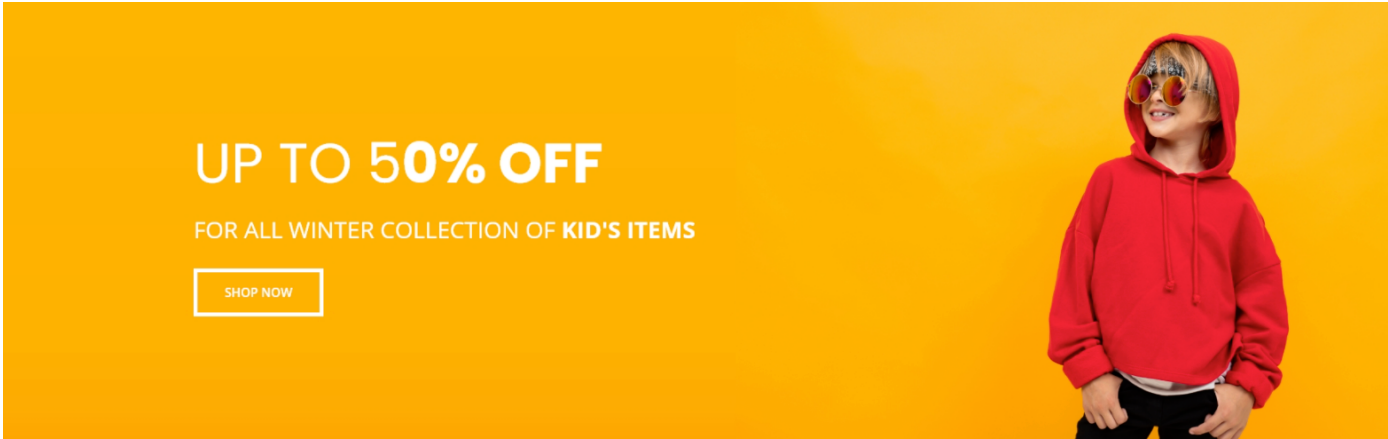
[1]



[2]

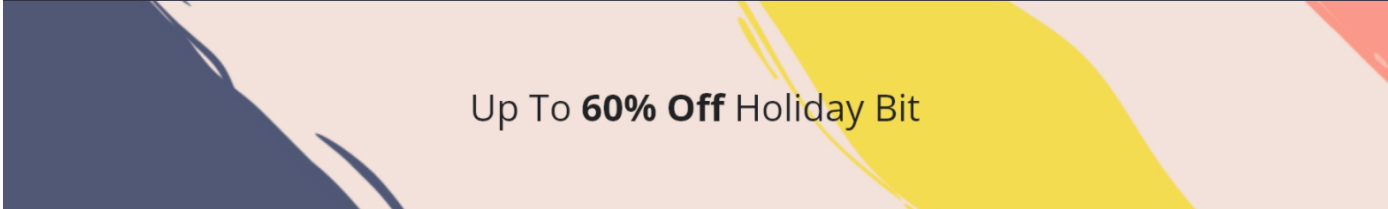


[3]



promotionBannerTwo

[1]



[2]

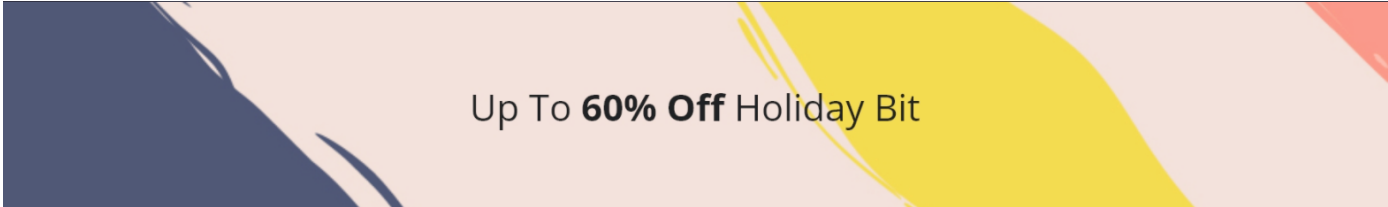


[3]

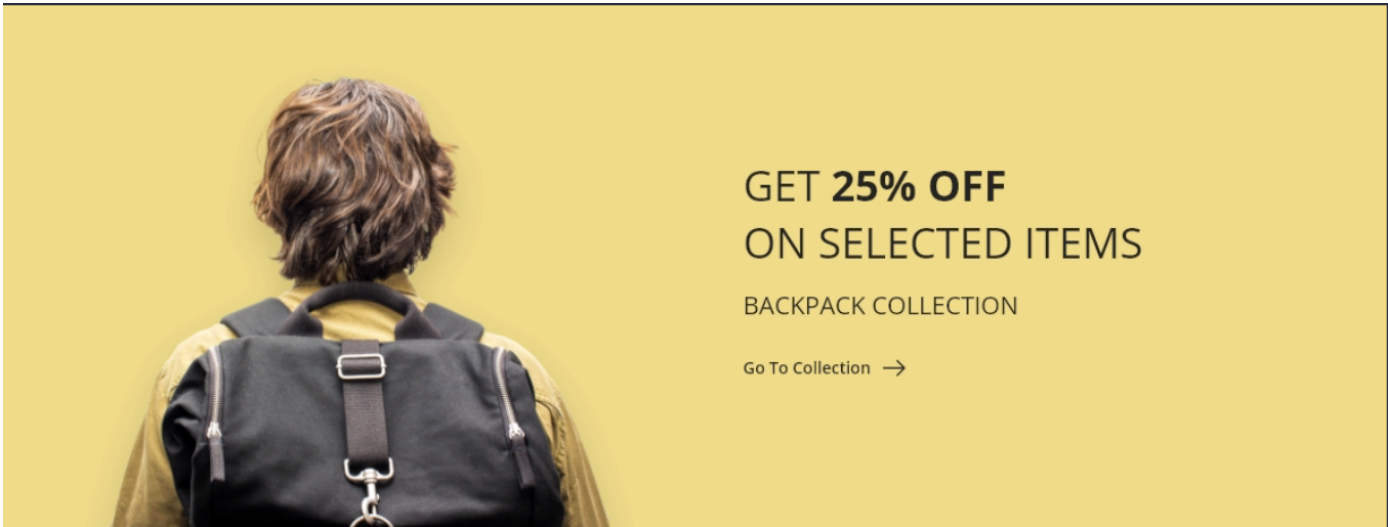


bannerDataOne

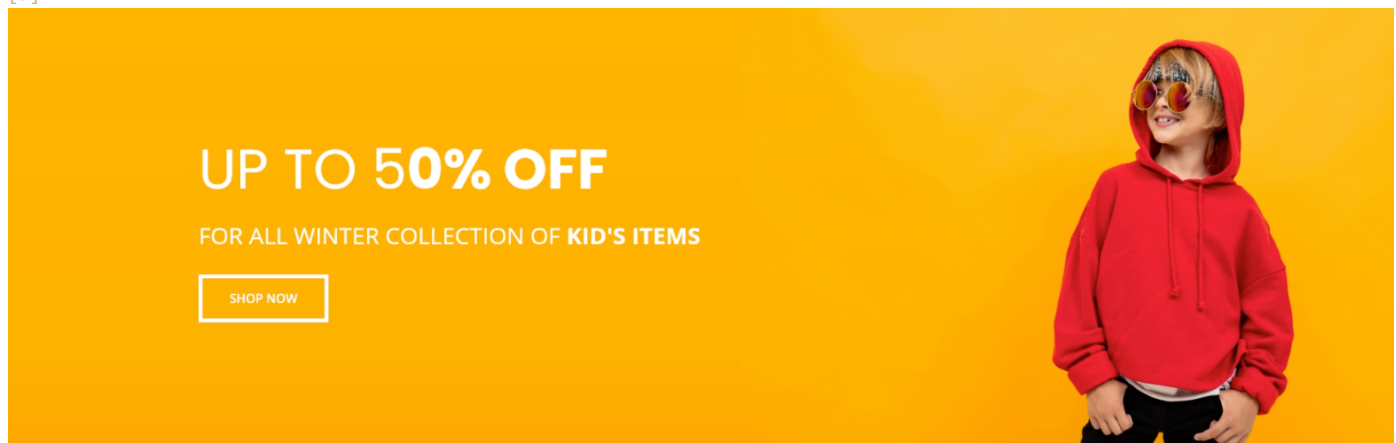
[1]



[2]

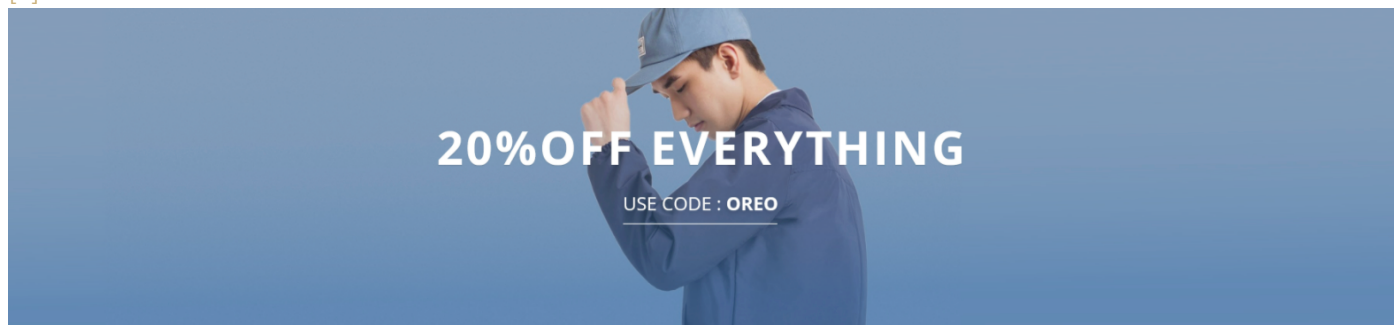


[3]



bannerDataTwo

[1]



[2]



[3]



homeThreeProductsBanner

[1]



[2]



homeFourBanner

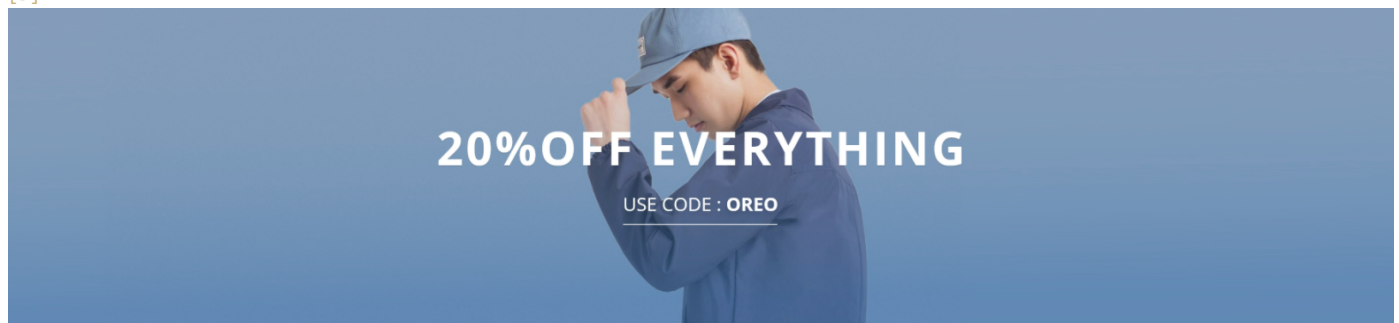
[1]



[2]

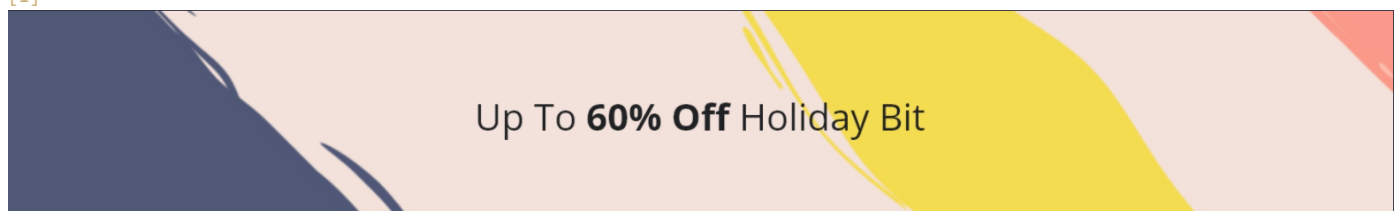


[3]



homeFourGridBanners

[1]



[2]



[3]



homeFiveBanner

[1]



Brand

We have multiple types of brand layouts.

- 1. Slider Layout -> For Minimal Demo
- 2. Grid Layout -> For Others Demo

For slider layout, you've to upload a single image, and for grid layout, you've to upload two images; one is a cover, and another is a logo.

You'll get a layout option when you create a brand,

The screenshot shows the CHAWK BAZAAR admin dashboard. On the left is a sidebar with navigation links: Dashboard, Shops, My Shops, Products, Attributes, Brands, Categories, Tags, Orders, Order Status, Users, Coupons, Taxes, Shippings, Withdraws, and Settings. The main content area has two sections: 'Description' with a placeholder text 'Add type-description-help-text' and 'Brand Images' with a placeholder text 'Add your brand images for different layout'. A red arrow points to the 'Brand Images' section. In the 'Brand Images' section, there is a 'Select layout type' dropdown menu with options 'Grid Layout' and 'Slider Layout'. The 'Slider Layout' is currently selected. Below the dropdown is an 'Upload an image or drag and drop' area with a dashed border and a note 'PNG, JPG'. There is an 'Add Image' button below the upload area. At the bottom right of the 'Brand Images' section is an 'Add Brand' button. A 'Remove' link is also visible next to the 'Slider Layout' option.

If you use a minimal home page, then add a slider layout; otherwise, use Grid Layout.

Utilities

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

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

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

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

Backend Integration

API Integration:

For API Integration:

```
> [your-frontend-project] = admin [rest] 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`

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

And run your frontend project

Data Fetching

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

We have used React Query and Axios which is defined in

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

For details documentation:

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

Uses 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(inputVaues) {
  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
```

AWS Management Console

The screenshot shows the AWS Management Console interface. The top section displays 'AWS services' with a list of recently visited services and all services categorized by function (Compute, Containers, Storage, Quantum Technologies, Management & Governance, Security, Identity, & Compliance). The right sidebar contains promotional banners for mobile apps, S3 on Outposts, digital training, and SageMaker resources. The bottom section shows the 'Instances (1)' page with a table of instances. A red arrow points to the 'Launch instances' button in the top right corner of the 'Instances' page.

AWS services

Recently visited services

All services

- Compute
 - EC2
 - Lightsail
 - Lambda
 - Batch
 - Elastic Beanstalk
 - Serverless Application Repository
 - AWS Outposts
 - EC2 Image Builder
- Containers
 - Elastic Container Registry
 - Elastic Container Service
 - Elastic Kubernetes Service
- Storage
 - S3
- Quantum Technologies
 - Amazon Braket
- Management & Governance
 - AWS Organizations
 - CloudWatch
 - AWS Auto Scaling
 - CloudFormation
 - CloudTrail
 - Config
 - OpsWorks
 - Service Catalog
 - Systems Manager
 - AWS AppConfig
 - Trusted Advisor
 - Control Tower
 - AWS License Manager
- Security, Identity, & Compliance
 - IAM
 - Resource Access Manager
 - Cognito
 - Secrets Manager
 - GuardDuty
 - Inspector
 - Amazon Macie
 - AWS Single Sign-On
 - Certificate Manager
 - Key Management Service
 - CloudHSM
 - Directory Service
 - WAF & Shield
 - AWS Firewall Manager
 - Artifact

Stay connected to your AWS resources on-the-go

Download the AWS Console Mobile App to your iOS or Android mobile device. [Learn more](#)

Explore AWS

Amazon S3 on Outposts

You can now use S3 object storage in your on-premises environment. [Learn more](#)

Free Digital Training

Get access to 350+ self-paced online courses covering AWS products and services. [Learn more](#)

Amazon SageMaker Resources

Explore features, use cases, and tutorials for every developer. [Learn more](#)

Feedback English (US)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

New EC2 Experience

EC2 Dashboard

Events

Tags

Limits

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Images

AMIs

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Instances (1)

Filter instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-00fb9dda581f15817	Terminated	t2.micro	-	1 alarm...	us-east-1e	-

Select an instance above

Launch instances

Feedback English (US)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

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.

Volume Type - ami-0885b1f6bd170450c

Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>)

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMLs, the private key file is required to obtain the password used to log into your instance. For Linux AMLs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AML](#).

Create a new key pair ▾

Key pair name

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel Launch Instances

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

New EC2 Experience

EC2 Dashboard **New**

Events

Tags

Limits

Instances

Instances **New**

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts **New**

Scheduled Instances

Capacity Reservations

Images

AMIs

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Feedback English (US)

Instances (1/2) Info

Filter instances

1

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-00fb9dda581f15817	Termina...	t2.micro	-	2 alar...	us-east-1e	-
pickbazar	i-0b3ad96ed56b60940	Running	t2.micro	-	No alarms	us-east-1d	ec2-54-90-111-167

Instance: i-0b3ad96ed56b60940 (pickbazar)

Details Security Networking Storage Status Checks Monitoring Tags

Instance summary Info

Instance ID

i-0b3ad96ed56b60940 (pickbazar)

Public IPv4 address

54.90... , open address

Private IPv4 addresses

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Domain Setup

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

New EC2 Experience

EC2 Dashboard **New**

Events

Tags

Limits

Instances

Instances **New**

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts **New**

Scheduled Instances

Capacity Reservations

Images

AMIs

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Feedback English (US)

Instances (1/2) Info

Filter instances

1

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-00fb9dda581f15817	Termina...	t2.micro	-	2 alar...	us-east-1e	-
pickbazar	i-0b3ad96ed56b60940	Running	t2.micro	-	No alarms	us-east-1d	ec2-54-90-111-167

Instance: i-0b3ad96ed56b60940 (pickbazar)

Details Security Networking Storage Status Checks Monitoring Tags

Instance summary Info

Instance ID

i-0b3ad96ed56b60940 (pickbazar)

Public IPv4 address

54.90... , open address

Private IPv4 addresses

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

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**

Instances (1/2) Info

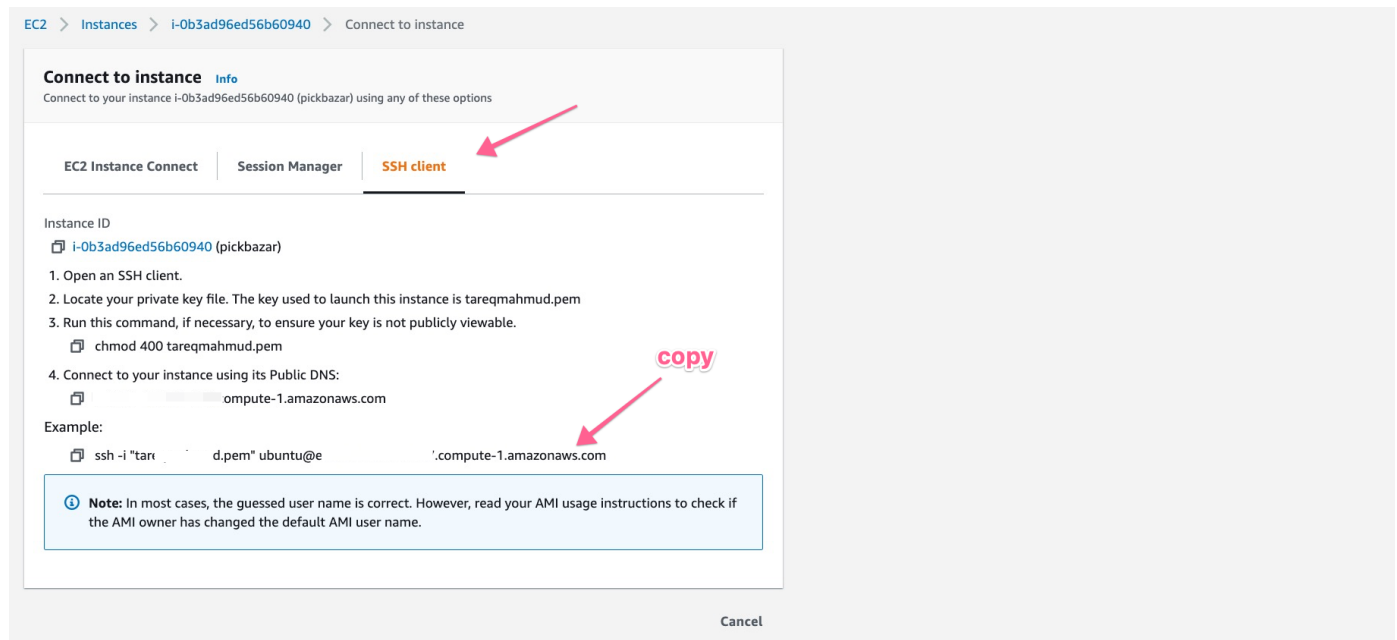
Connect

Instance state

Actions

Launch instances

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 chawkbazar.pem
```

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

Now go to the [VPS Server](#) section for deploy the [Chawkbazar Laravel](#)

Virtual Private Server (Automated Script)

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

With this tutorial, you can install ChawkBazar 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 your PC - RUN on Local PC

To upload the zipped `chawkbazar-api` and `deployment` files to server you need to run the below command form your chawkbazar 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 `chawkbazar-api.zip` and `deployment.zip` files path and the path will be look like `/home/your_project_folder_path/chawkbazar-laravel/chawkbazar-api.zip` for chawkbazar-api.zip file so forth for `deployment.zip`

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

Then login your server using `ssh` and,

Server Environment setup script - RUN on Virtual Server

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

Nginx Setup And Settings - RUN on Virtual Server

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

Backend build - RUN on Virtual Server

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

Frontend build script - RUN on Local PC

Run the below command from your chawkbazar project root

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

Frontend run script - RUN on Virtual Server

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

Vercel

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
```

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 more details,

Please follow [nextjs deployment docs](#):

Virtual Private Server (Manual)

With this tutorial, you can install ChawkBazar 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 server with at least 2+ CPU cores and 2GB+ memory.

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

<https://www.youtube.com/watch?v=cESkBAmmDPc>

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 chawkbazar app. To install NodeJS and npm, run this command on your terminal,

```
sudo apt-get update
curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
sudo apt-get install -y nodejs
```

Install Yarn

Chawkbazar 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 chawkbazar

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
```

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

```
sudo apt install php-fpm php-mysql
sudo apt install php-fpm php-mysql php-mbstring php-xml php-bcmath php-simplexml php-mbstring php7.4-gd php7.4-curl composer
```

Step 6 - Create MySQL Database & User

```
sudo mysql
CREATE DATABASE chawkbazar;

CREATE USER 'chawkbazar_user'@'%' IDENTIFIED WITH mysql_native_password BY 'chawkbazar1';

GRANT ALL ON chawkbazar.* TO 'chawkbazar_user'@'%';

FLUSH PRIVILEGES;
```

We use MySQL user name `chawkbazar_user` and MYSQL password `chawkbazar1`. 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/chawkbazar
```

Then, go to your `local computer`

1. Extract the `chawkbazar` package that you download from `CodeCanyon`.
2. On that folder, you'll get another `folder` called `chawkbazar-laravel`.
3. On that folder, you'll get a folder called `chawkbazar-api`

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

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/chawkbazar
sudo nano /etc/nginx/sites-available/chawkbazar
```

Add this Nginx config file to that edited file,

```
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 /api {
        alias /var/www/chawkbazar/chawkbazar-api/public;
        try_files $uri $uri/ @api;
        location ~ /\.php$ {
            include fastcgi_params;
            fastcgi_param SCRIPT_FILENAME $request_filename;
            fastcgi_pass unix:/run/php/php7.4-fpm.sock;
        }
    }

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

    # For FrontEnd -> Rest
    location /{
        proxy_pass http://localhost:3003;
        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/php7.4-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/chawkbazar /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 prompt. 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 `chawkbazar-api` folder, then copy `.env.example` to `.env`,

```
cd /var/www/chawkbazar/chawkbazar-api
cp .env.example .env
```

Edit `.env`

```
nano .env
```

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

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

```
1 APP_NAME=ChawkBazar
2 APP_ENV=production
3 APP_KEY=base64:d\adddfsfdfsdfssdf/sdfdsfer
4 APP_DEBUG=true
5 APP_URL=https://YOUR_DOMAIN.COM/backend
6 APP_SERVICE=chawkbazar.test
7 APP_NOTICE_DOMAIN=CHAWKBAZAR_
8
9 DUMMY_DATA_PATH=chawkbazar
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/api` you'll get a `welcome` page like this

Marvel Laravel

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

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 `chawkbazar-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 const withPWA = require("next-pwa");
2 const runtimeCaching = require("next-pwa/cache");
3 const { i18n } = require("./next-i18next.config");
4
5 module.exports = withPWA({
6   basePath: "/admin",
7   i18n,
8   pwa: {
9     disable: process.env.NODE_ENV === "development",
10    dest: "public",
11    runtimeCaching,
12  },
13
14  images: {
15    domains: [
16      "via.placeholder.com",
17      "res.cloudinary.com",
18      "s3.amazonaws.com",
19      "18.141.64.26",
20      "127.0.0.1",
21      "localhost",
22      "picsum.photos",
23      "pickbazar-sail.test",
24      "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
25      "chawkbazarlaravel.s3.ap-southeast-1.amazonaws.com",
26      "lh3.googleusercontent.com",
27    ],
28  },
```

Step 2 - Install & Build

go to your `chawkbazar-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.

Shop

Go to `shop` and copy `.env.template` to `.env`

Then edit `.env` and add your `API` url to that file

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

Also, update all the necessary configurations like `OTP`, `Mailchimp Token`, `Instagram Token`, `Google MAP API`, etc.

Admin

After that, Go to `admin -> rest` and copy `.env.template` to `.env`

Then edit `.env` and add your `API` url to that file

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

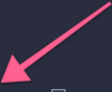
Also, update all the necessary configurations like `Google MAP API`, etc.

Image Config

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

and add your domain to `images` object

```
1 const withPWA = require("next-pwa");
2 const runtimeCaching = require("next-pwa/cache");
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
13  images: {
14    domains: [
15      "YOUR_API_DOMAIN",
16      "via.placeholder.com",
17      "res.cloudinary.com",
18      "s3.amazonaws.com",
19      "18.141.64.26",
20      "127.0.0.1",
21      "localhost",
22      "picsum.photos",
23      "pickbazar-sail.test",
24      "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
25      "chawkbazarlaravel.s3.ap-southeast-1.amazonaws.com",
26      "lh3.googleusercontent.com",
27    ],
28  },
29 });
```



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 `chawkbazar-laravel` folder,

run this for build `admin`

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

And for the `shop`,

at first, run the shop in one terminal

```
yarn dev:shop-rest
```

and then build the shop,

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

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

Now go to the server `/var/www/chawkbazar` using terminal

Then `unzip` that `zip` file.

Install FrontEnd And Run

Then install all the node packages,

```
yarn
```

Run frontend app

For `shop` app, use this command,

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

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

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

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

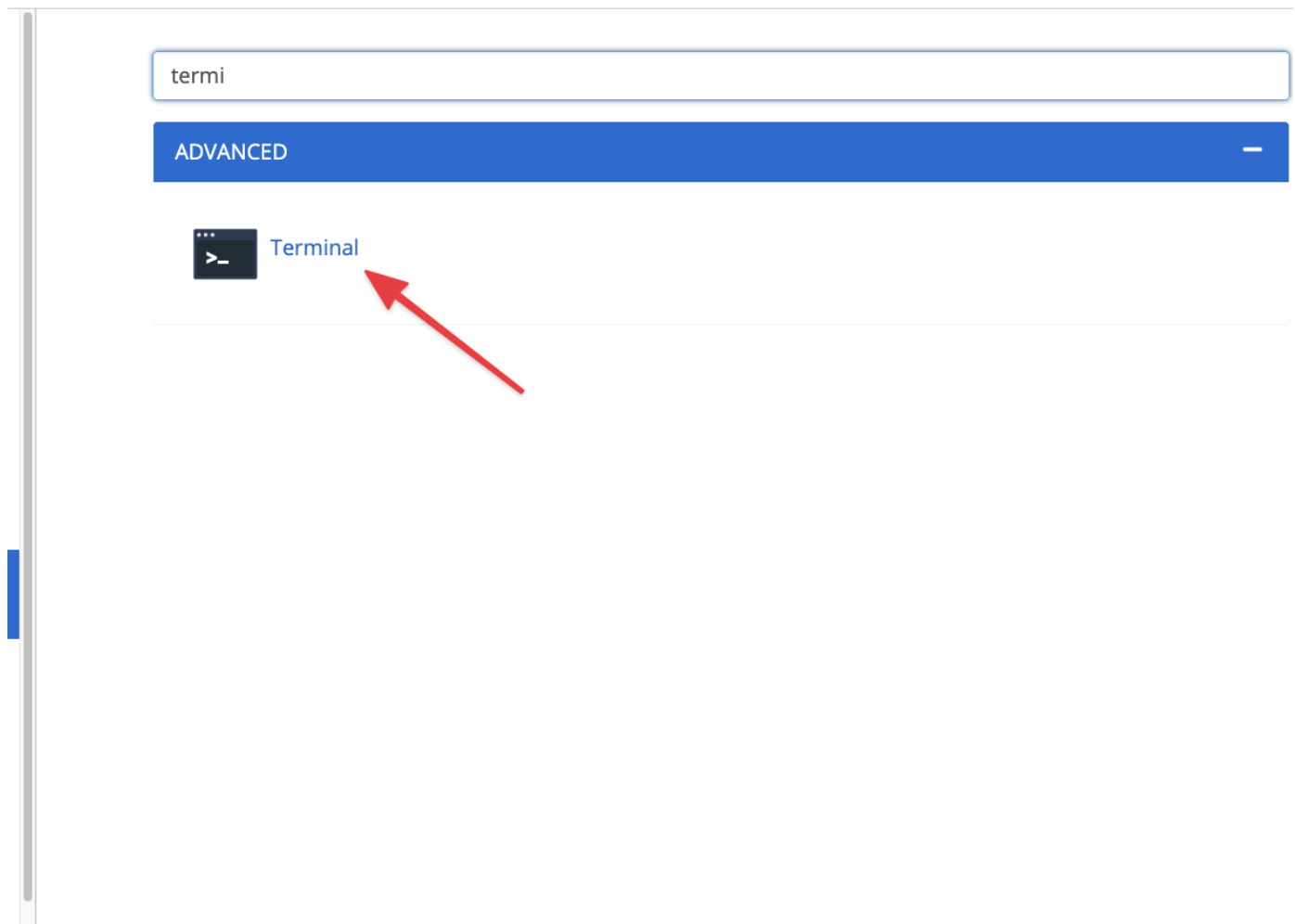
cPanel

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

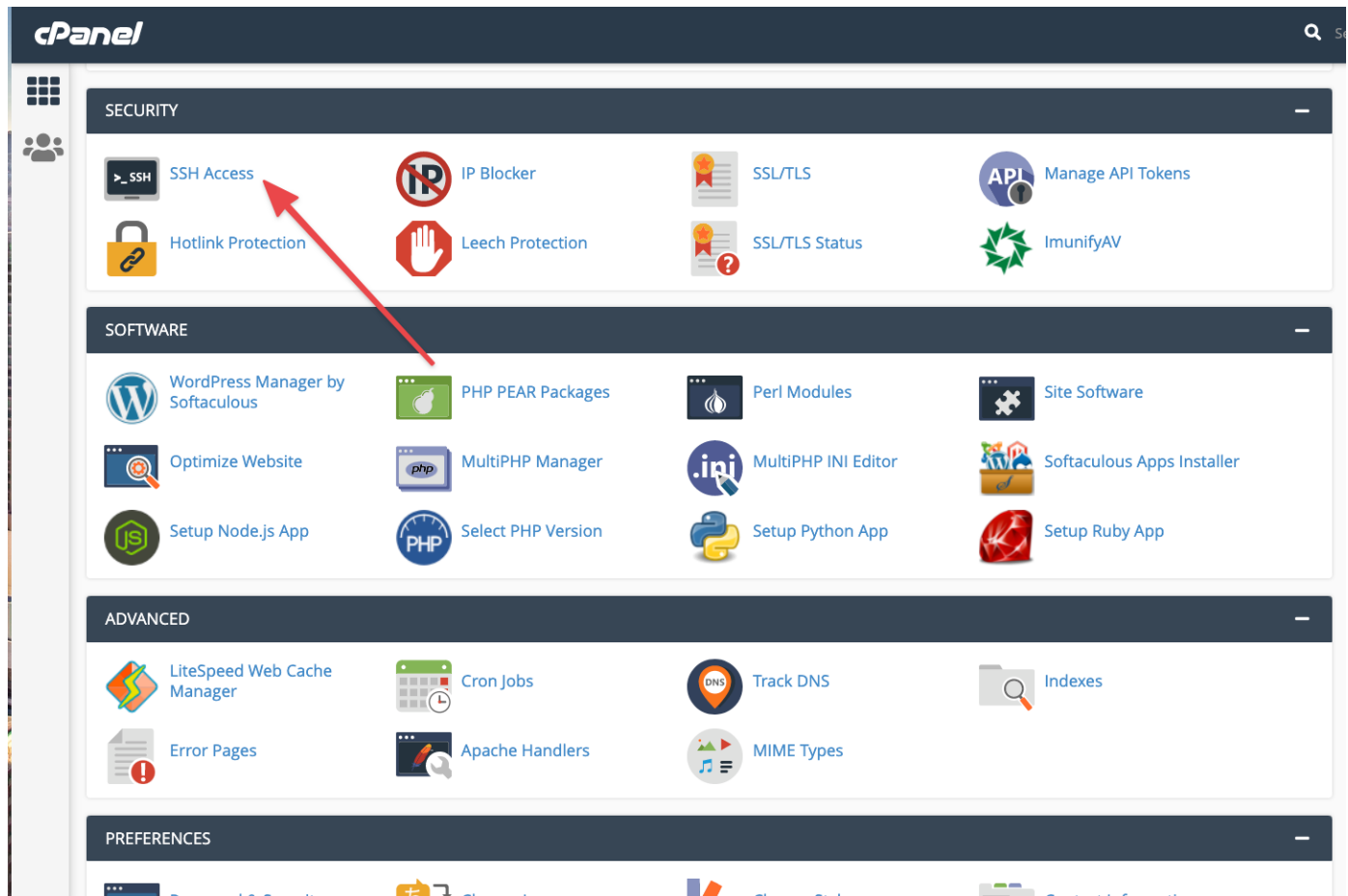
If you still decide to proceed with panel based server, our support team won't be able to help you. We have put some resources for Cpanel in this documentation section to help our users to get started but other than that, we don't have much to offer with Cpanel or any Managed Server.

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.

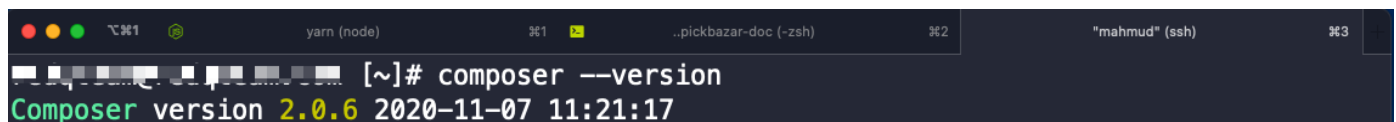


After enabling the ssh login to your server using ssh,

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
```



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,

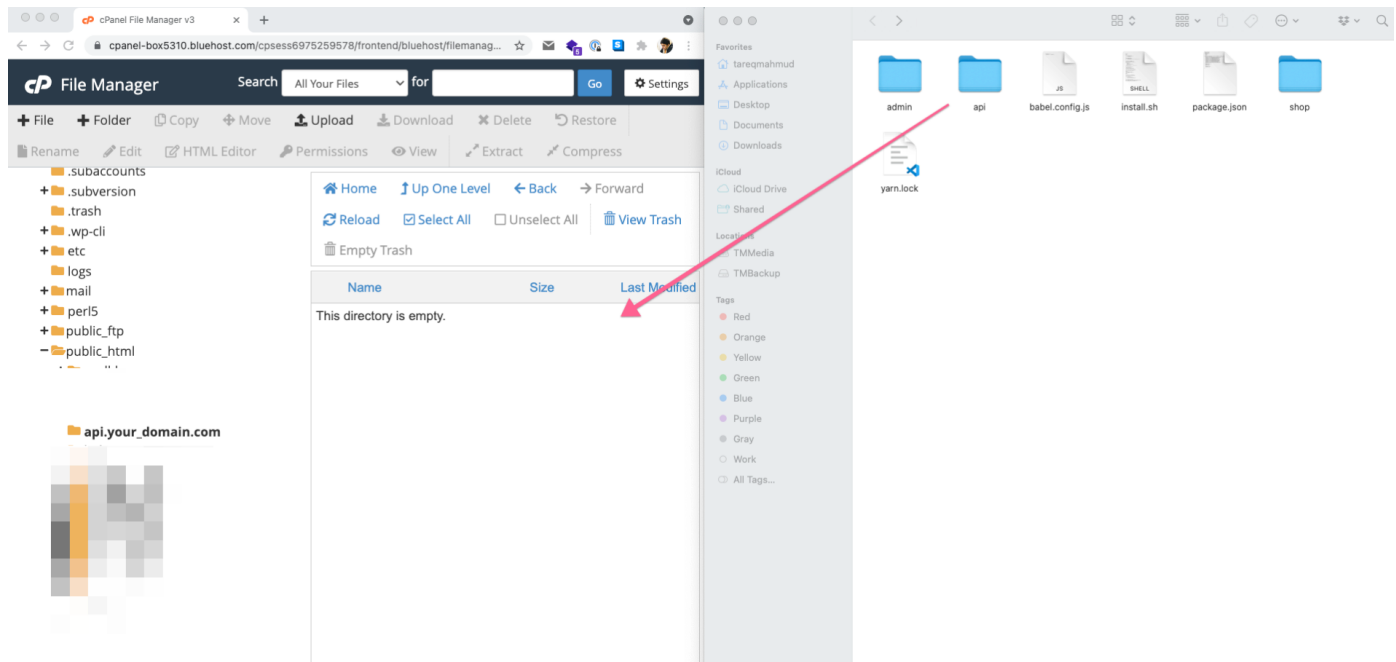
```
-> store.your_domain.com -> host frontend store.
-> api.your_domain.com -> host laravel API.
-> admin.your_domain.com -> host admin dashboard.
```

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.

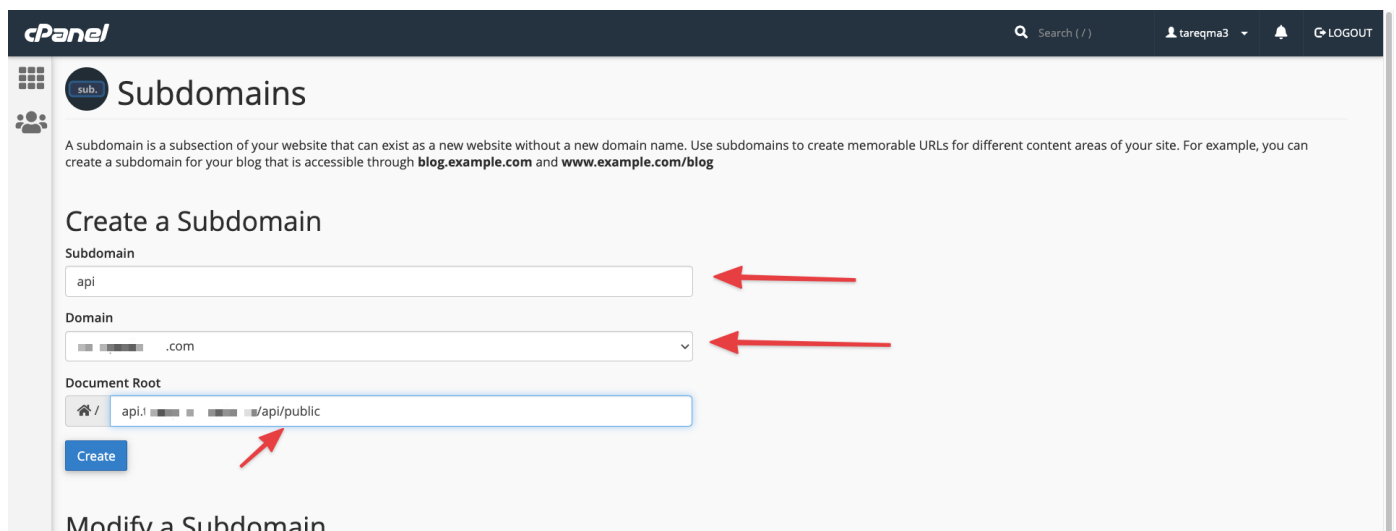
Install API

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

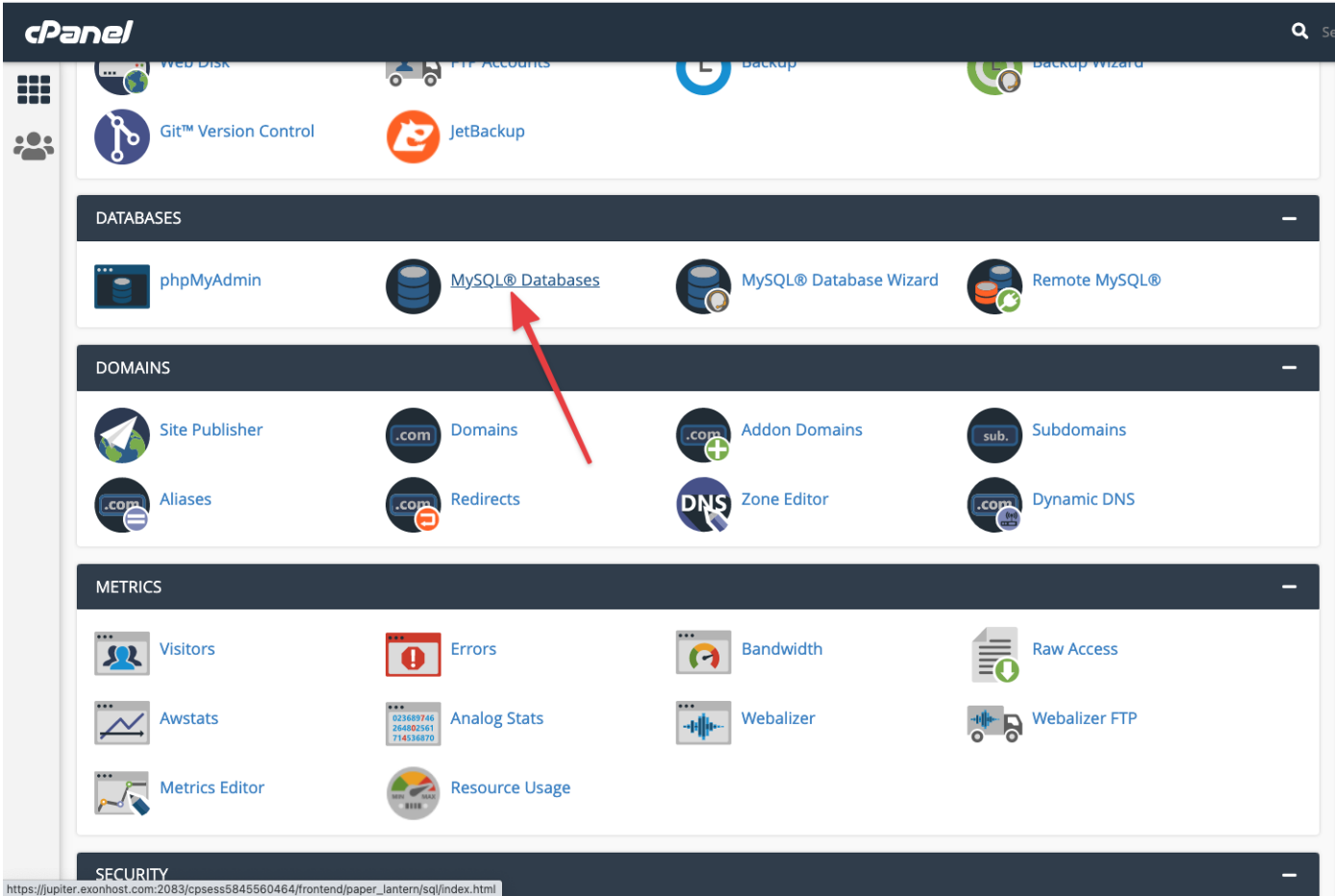
Now upload this **api** folder to the **api.your_domain.com** folder in your server



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



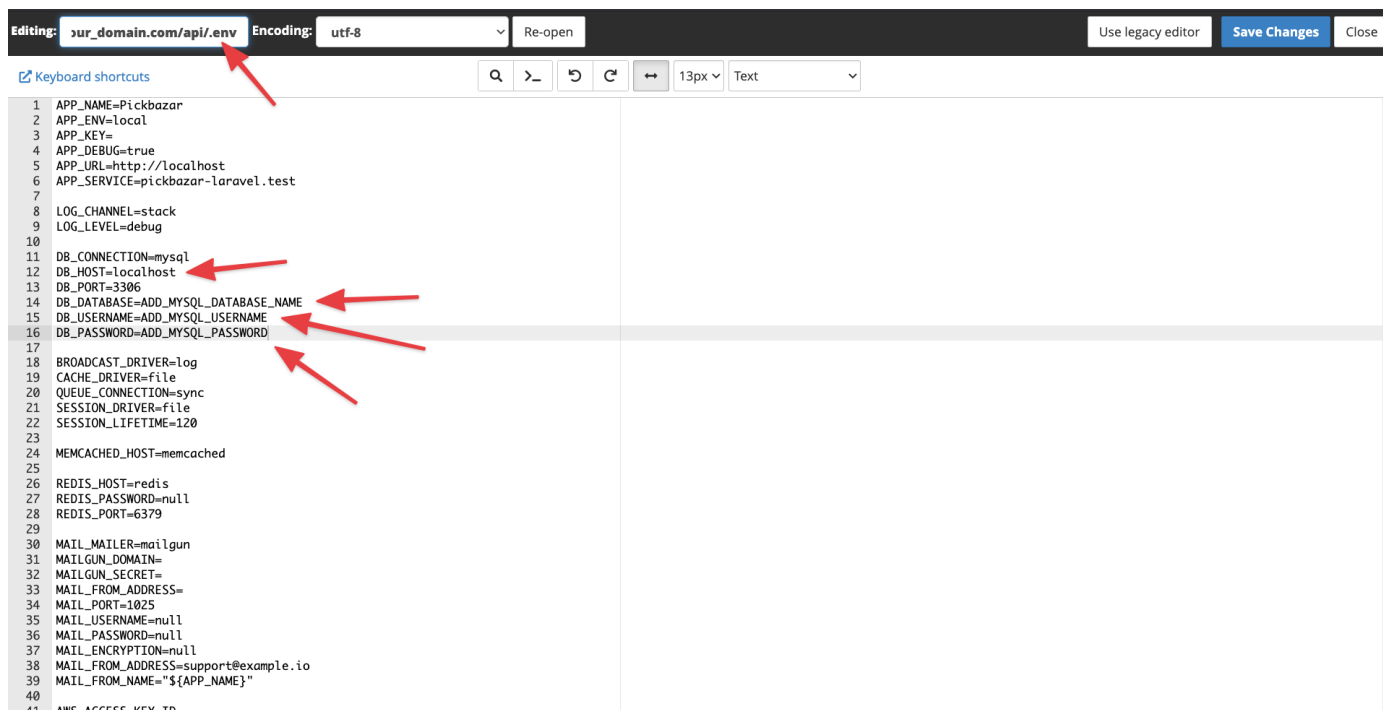
Now create a MySQL database and user from MySQL wizard



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

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



```

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.



```

1 APP_NAME=ChawkBazar
2 APP_ENV=production
3 APP_KEY=base64:d1addfdsfdfdsfssdf/sdfdsfer
4 APP_DEBUG=true
5 APP_URL=https://YOUR_DOMAIN.COM/backend
6 APP_SERVICE=chawkbazar.test
7 APP_NOTICE_DOMAIN=CHAWKBAZAR_
8
9 DUMMY_DATA_PATH=chawkbazar
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,

Marvel Laravel

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

Install FrontEnd

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.
```

FrontEnd 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 upload the build file to the server.

step 1 - Build Custom Server

go to your `chawkbazar-laravel` folder

shop

Create custom server for `shop`,

create a `server.js` file on,

```
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/package.json,

and replace start script with this,

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

The screenshot shows two side-by-side views of a 'package.json' file. The left pane shows the original configuration, and the right pane shows the updated configuration. A red arrow points from the 'start' script in the left pane to the updated 'start' script in the right pane.

```

1 {
2   "name": "@pick-bazar/shop",
3   "version": "4.2.0",
4   "private": true,
5   "scripts": {
6     "clean": "rimraf \"{.next,node_modules,__generated__,.cache,src/fr
7     amework/graphql/**/*.d.ts}\"",
8     "dev:rest": "next dev -p 3003",
9     "build:rest": "next build",
10    "codegen": "node -r dotenv/config $(yarn bin)/graphql-let",
11    "dev:gql": "yarn codegen && next dev -p 3001",
12    "build:gql": "yarn codegen && next build",
13    "start": "next start",
14    "lint": "next lint"
15  },
16  "dependencies": {
17    "@apollo/client": "^3.4.16",
18    "@headlessui/react": "^1.4.1",
19    "@hookform/resolvers": "^2.8.2",
20    "@reach/portal": "^0.16.0",
21    "@stripe/react-stripe-js": "^1.5.0",
22    "@stripe/stripe-js": "^1.20.2",
23    "apollo-upload-client": "^16.0.0",
24    "axios": "^0.23.0",
25    "body-scroll-lock": "^4.0.0-beta.0",
26    "camelcase-keys": "^7.0.1",
27    "classnames": "^2.3.1",
28    "dayjs": "^1.10.7",
29    "deepmerge": "^4.2.2",
30    "framer-motion": "^4.1.17",

```

admin rest

Similarly, create custom server for admin rest,

create another server.js file on,

```
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/package.json,

and replace start script with this,

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

Step 2 - Install & Build

go to your hawkbazar-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.

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/
```

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 **chawkbazar-laravel** folder,

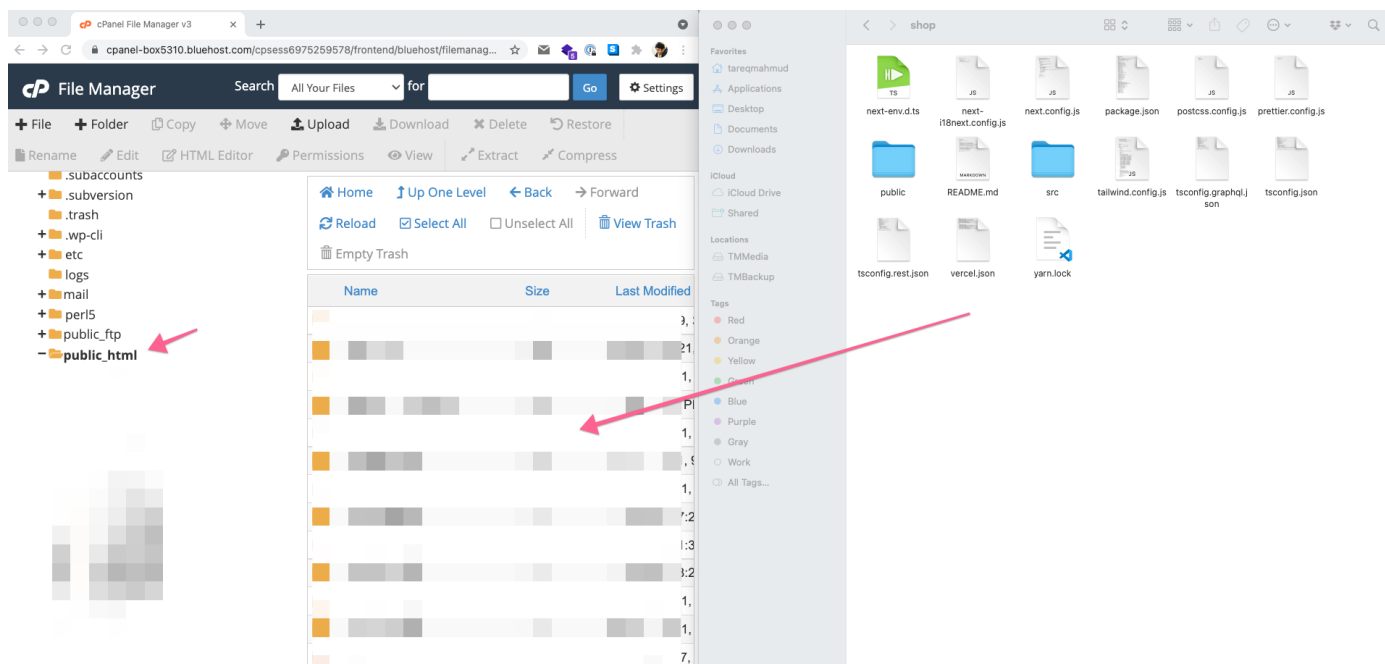
and run,

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

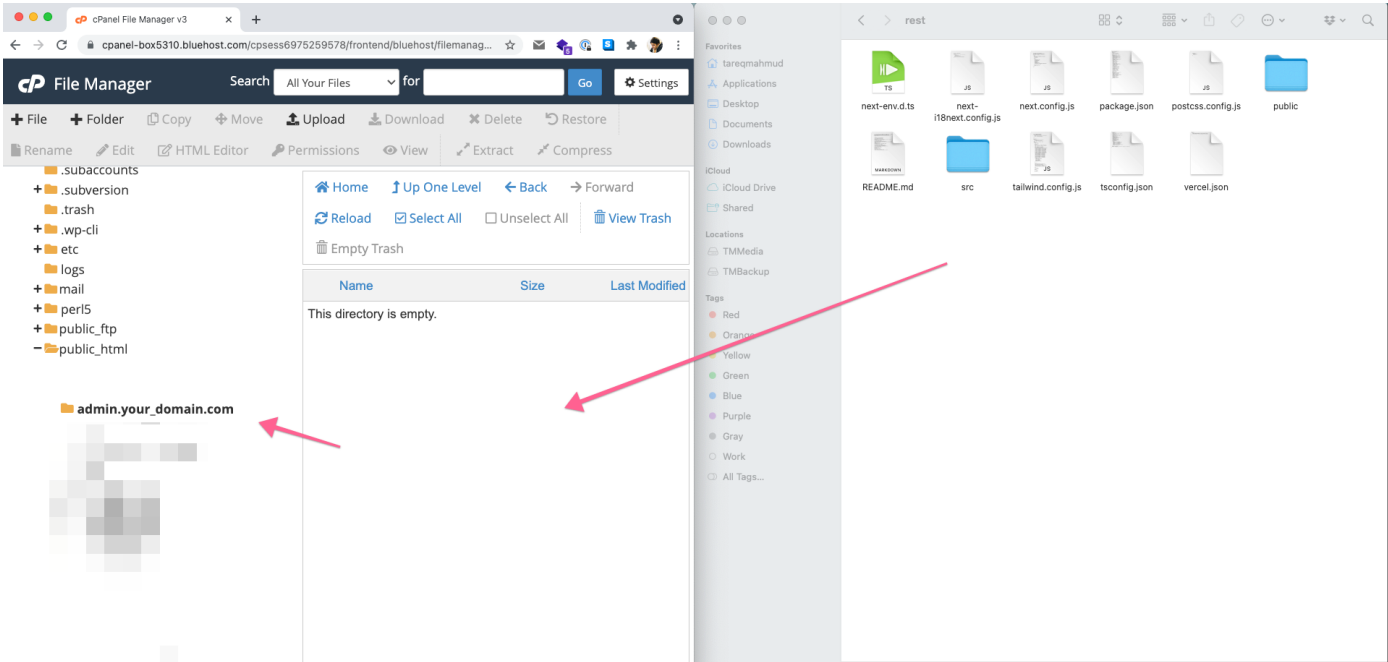
After build the project upload the

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

shop,



admin-rest,

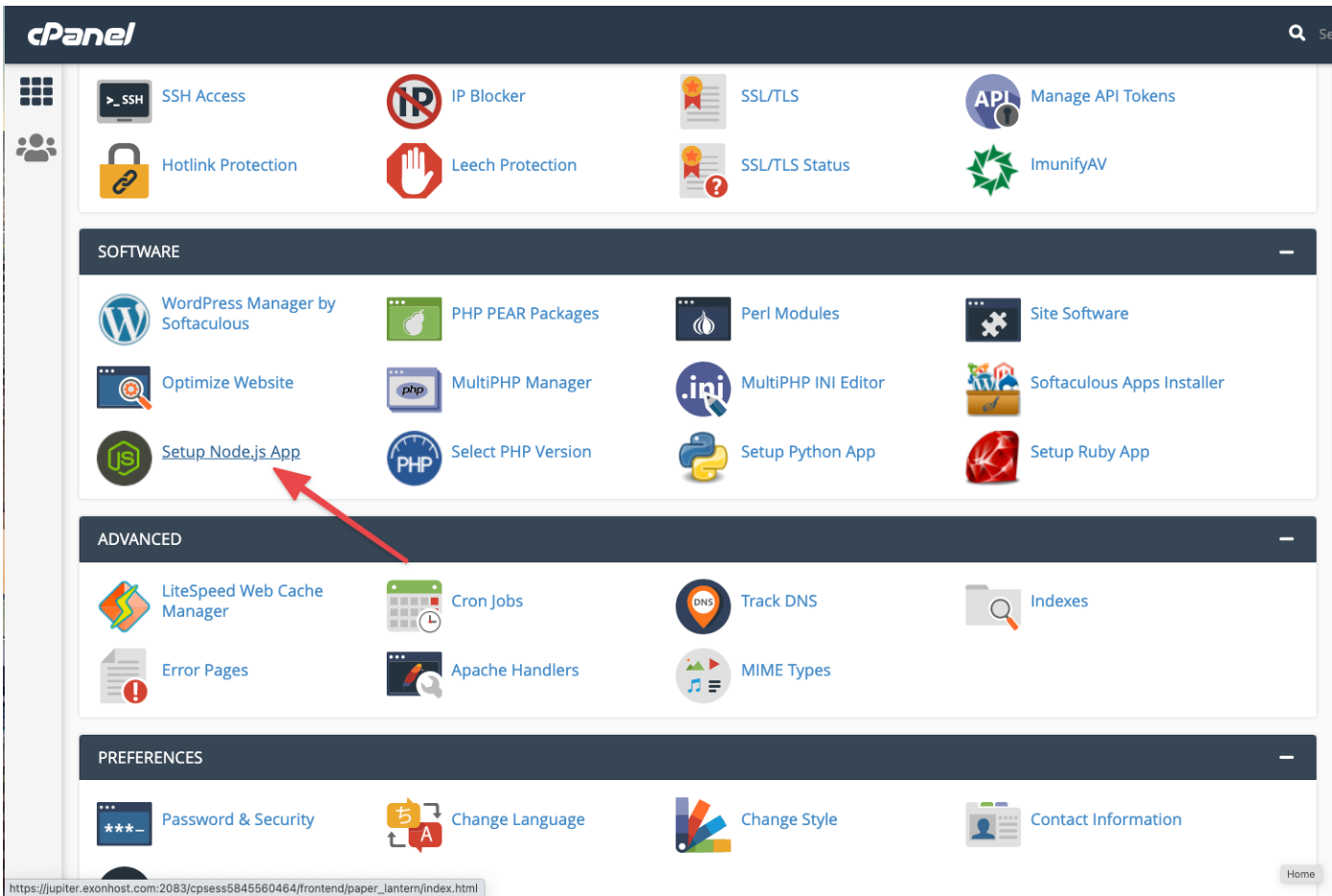


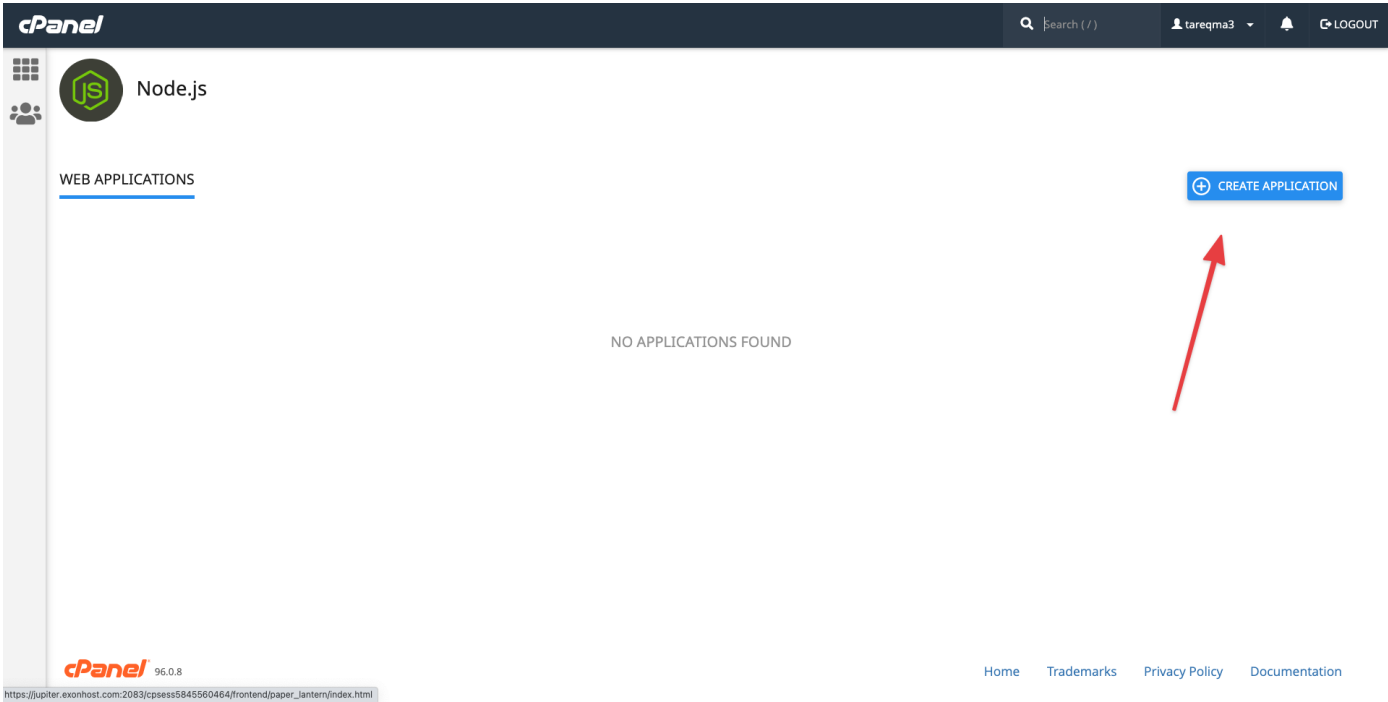
Install NodeJs Project

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,

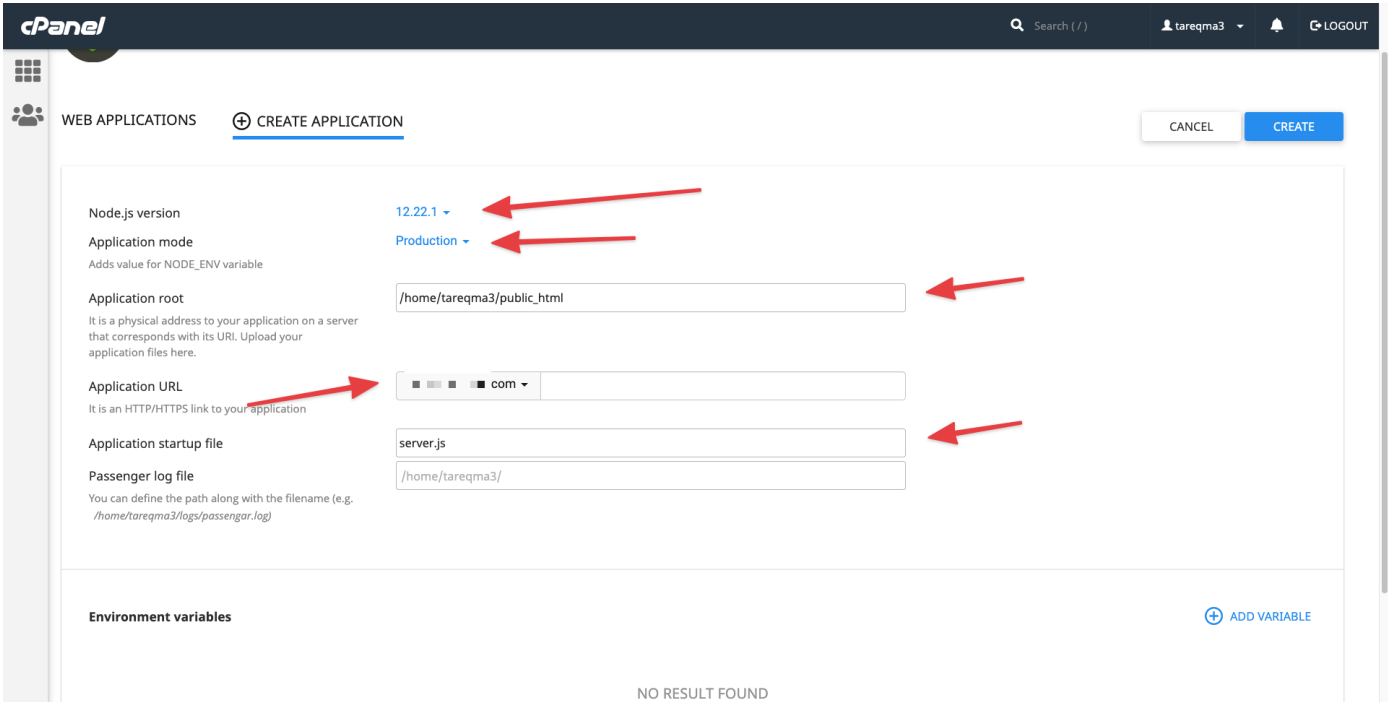
For `shop`,



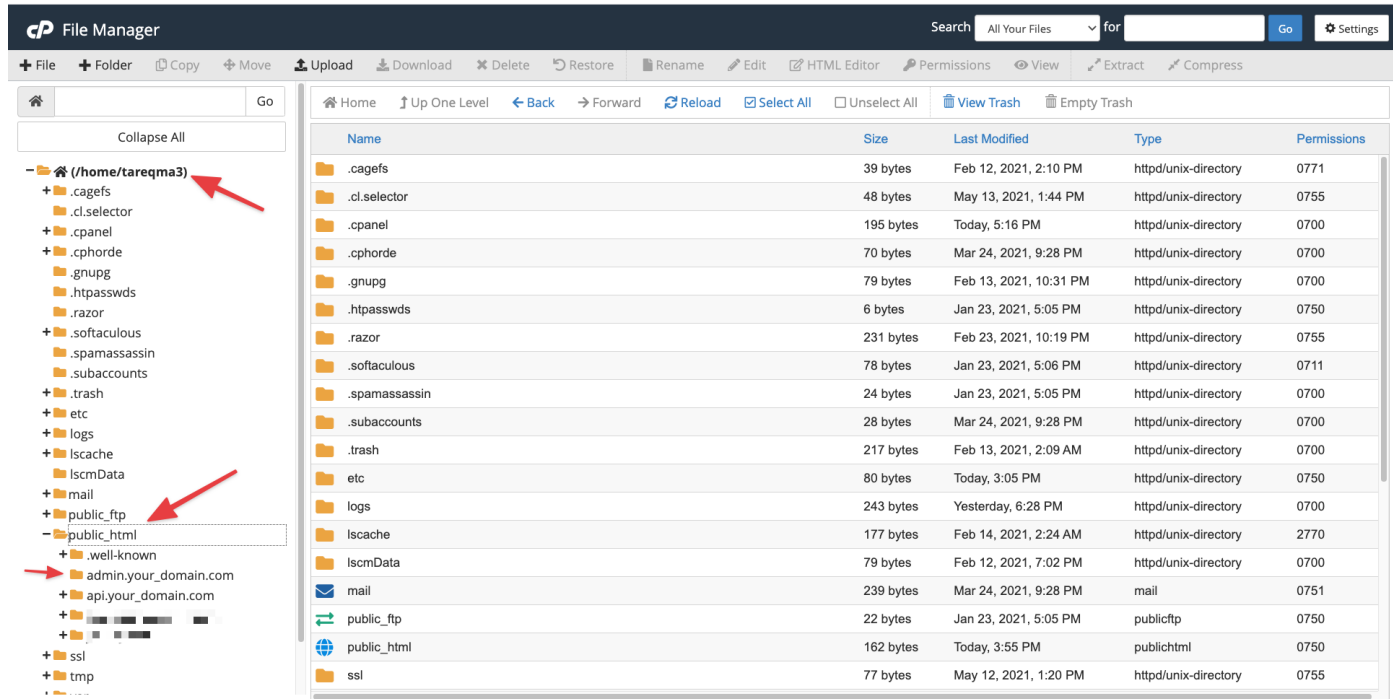


Now,

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



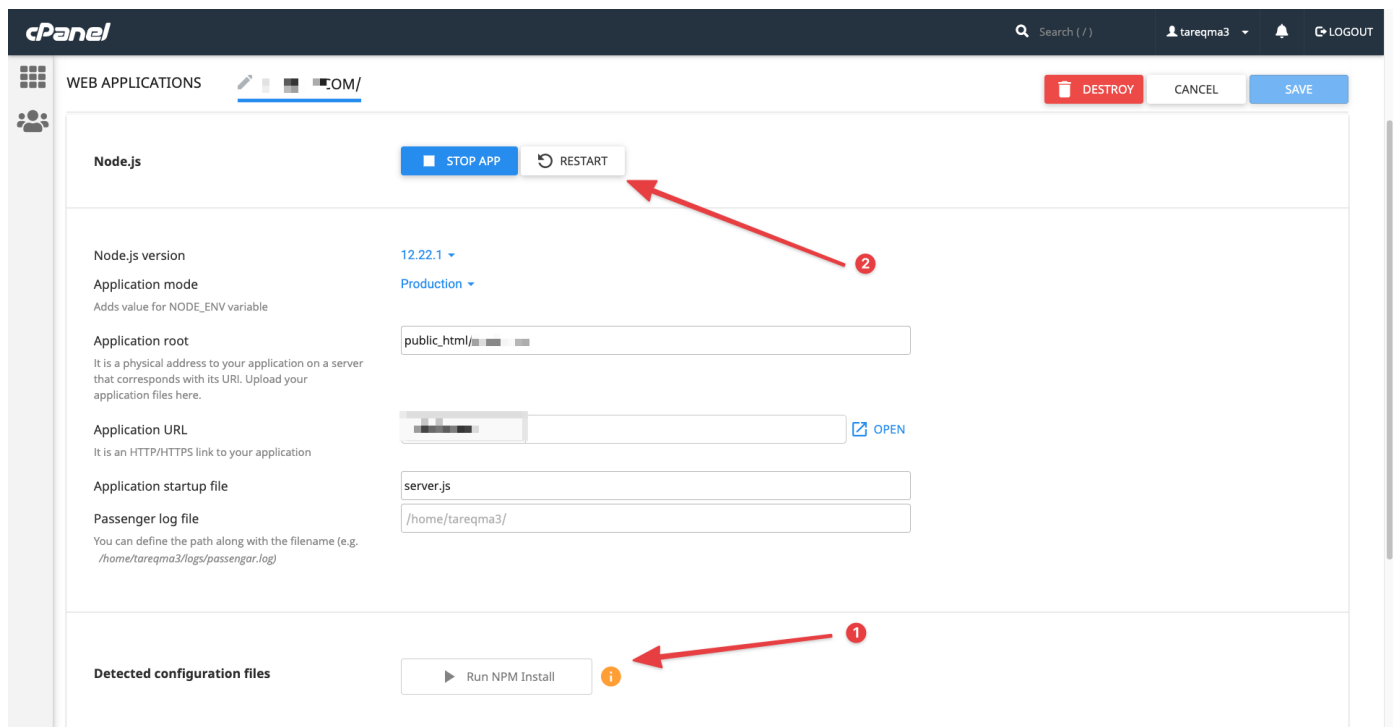
You can get the Application Path from your cPanel file manager



The screenshot shows the cPanel File Manager interface. On the left, a sidebar displays a tree view of the file system. A red arrow points to the 'public_html' directory. The main area shows a table of files and folders. A red arrow points to the 'public_html' folder in the table.

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
lscache	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_html	22 bytes	Jan 23, 2021, 5:05 PM	publichtml	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,



The screenshot shows the cPanel Web Applications interface. A red arrow points to the 'RESTART' button. Another red arrow points to the 'Run NPM Install' button.

Node.js

Node.js version: 12.22.1

Application mode: Production

Application root: public_html/

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 ChawkBazar,

Thank You!

SEO and Analytics

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 chawkbazar 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

Introduction

Chawkbazar is a laravel multi api package for ecommerce. This package is for building dynamic ecommerce site using chawkbazar package.

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 Video



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/10.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

```
"require": {
    "php": "^8.0|^8.1",
    "barryvdh/laravel-dompdf": "2.0.1",
    "doctrine/dbal": "3.6.0",
    "guzzlehttp/guzzle": "7.5.0",
    "laravel/framework": "10.1.5",
    "laravel/socialite": "5.6.1",
    "laravel/tinker": "2.8.1",
    "marvel/shop": "dev-master",
    "messagebird/php-rest-api": "3.1.4",
    "openai-php/client": "^0.4.1",
    "psr/log": "2.0.0",
    "srmklive/paypal": "3.0.19",
    "stevebauman/purify": "5.1.1",
    "symfony/http-client": "6.2.6",
    "symfony/mailgun-mailer": "6.2.5",
    "twilio/sdk": "6.44.4"
},
"require-dev": {
    "spatie/laravel-ignition": "1.3.1",
    "fakerphp/faker": "1.20.0",
    "laravel/sail": "1.15.1",
    "mockery/mockery": "1.5.0",
    "nunomaduro/collision": "6.2.1",
    "phpunit/phpunit": "9.5.21",
    "squizlabs/php_codesniffer": "3.7.1"
},
```

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=chawkbazar_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



- Run Docker application first
- Now go to your chawkbazar-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 `15-repository` (<https://github.com/andersao/15-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 `ShopServiceProviders` 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.

Social Login

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**.

API APIs & Services

Dashboard

Library

Credentials

OAuth consent screen

Domain verification

Page usage agreements

OAuth consent screen

because your OAuth request includes additional scopes that haven't been approved.

0 users / 100 user cap

SHOW LESS

OAuth rate limits

Your token grant rate ?

Token grant rates limit how quickly your application can get new users.

Your current per day token grant rate limit is 10,000 grants per day. The per day token grant rate resets every day. [Raise daily token limit](#)

5 minutes 1 day

No data is available for the selected time frame.

10,000

3 PM 6 PM 9 PM Thu 05 3 AM 6 AM 9 AM 12 PM 9,999

SHOW LESS

[Let us know what you think](#) about our OAuth experience

4. Create an **External OAuth Consent**.

5. Then, go to the **credentials** section and create **OAuth Client ID**.

API APIs & Services

Dashboard

Library

Credentials

OAuth consent screen

Domain verification

Page usage agreements

Credentials

+ CREATE CREDENTIALS

DELETE

Create credentials to access your API

API Keys

API key

Identifies your project using a simple API key to check quota and access

OAuth client ID

Requests user consent so your app can access the user's data

Service account

Enables server-to-server, app-level authentication using robot accounts

Help me choose

Asks a few questions to help you decide which type of credential to use

OAuth 2.0 Client ID

Restrictions	Key
None	AIzaSyA1np...-hFhc-1ots

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.

i

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

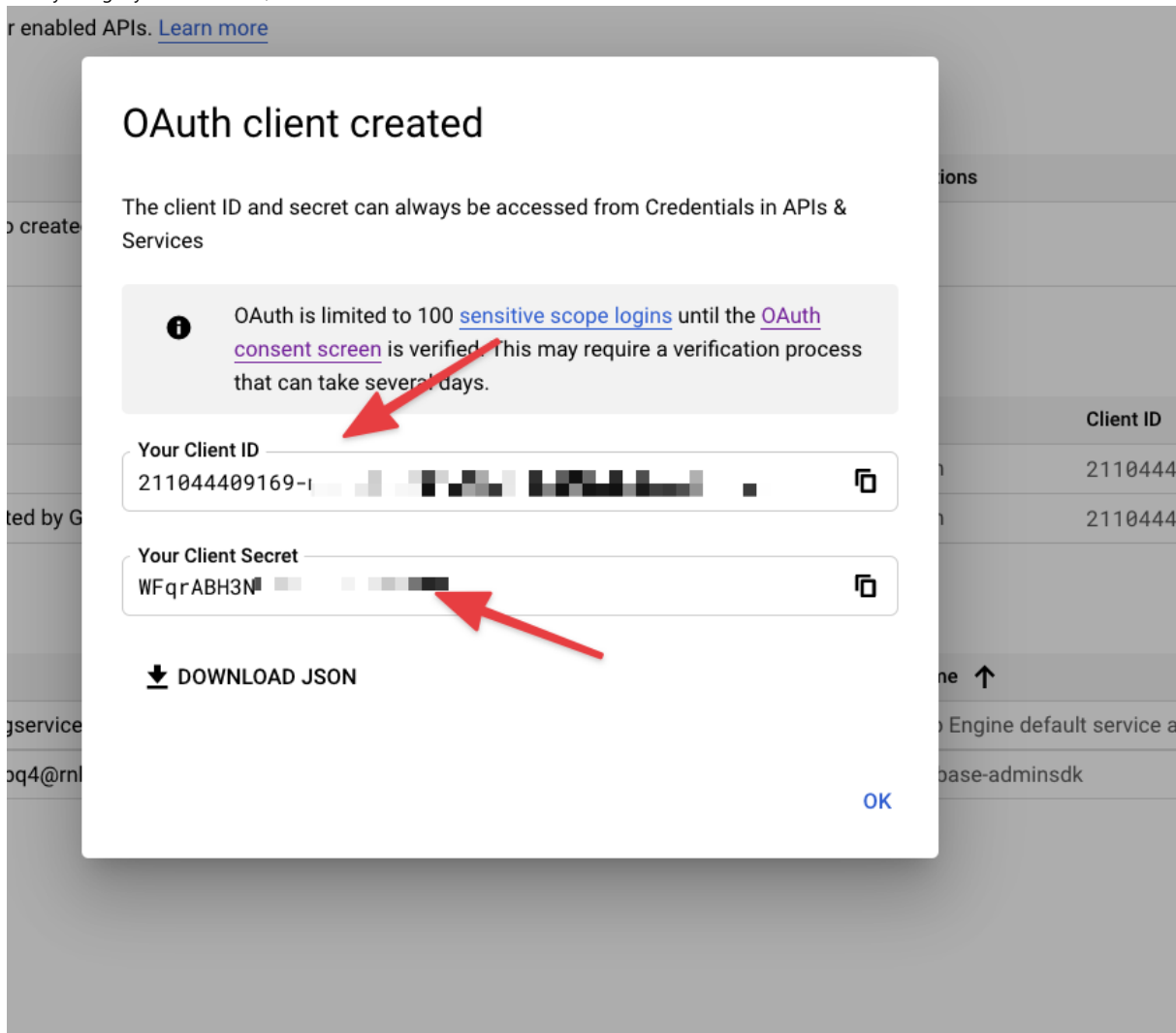
URIs 1 *

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

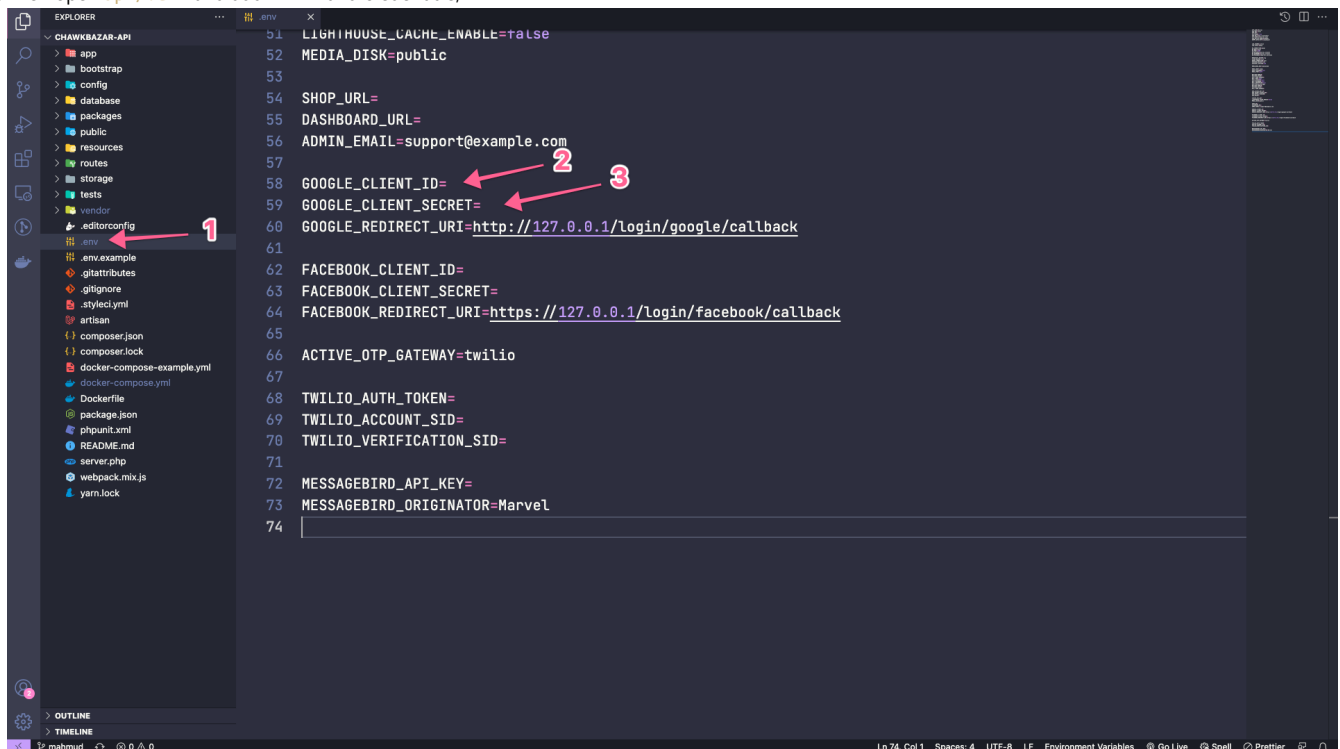
+ ADD URI

75 / 169

6. Then you'll get your credentials,



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



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` also add a `SECRET` code.

```

1  NEXT_PUBLIC_REST_API_ENDPOINT=http://localhost/
2
3  NEXT_PUBLIC_SITE_URL=http://localhost:3000
4
5  NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=
6
7  # Social Authentication
8  NEXTAUTH_URL=http://localhost:3003
9
10
11  SECRET= # Linux: `openssl rand -hex 32` or go to https://generate-secret.now.sh/32
12
13  GOOGLE_ID=
14  GOOGLE_SECRET=
15
16  FACEBOOK_ID=
17  FACEBOOK_SECRET=
18
19  NEXT_PUBLIC_GOOGLE_MAP_API_KEY=ADD_GOOGLE_MAP_API_KEY
20
21  NEXT_PUBLIC_INSTAGRAM_BASIC_DISPLAY_USER_TOKEN=your_instagram_basic_display_user_token_here
  
```

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

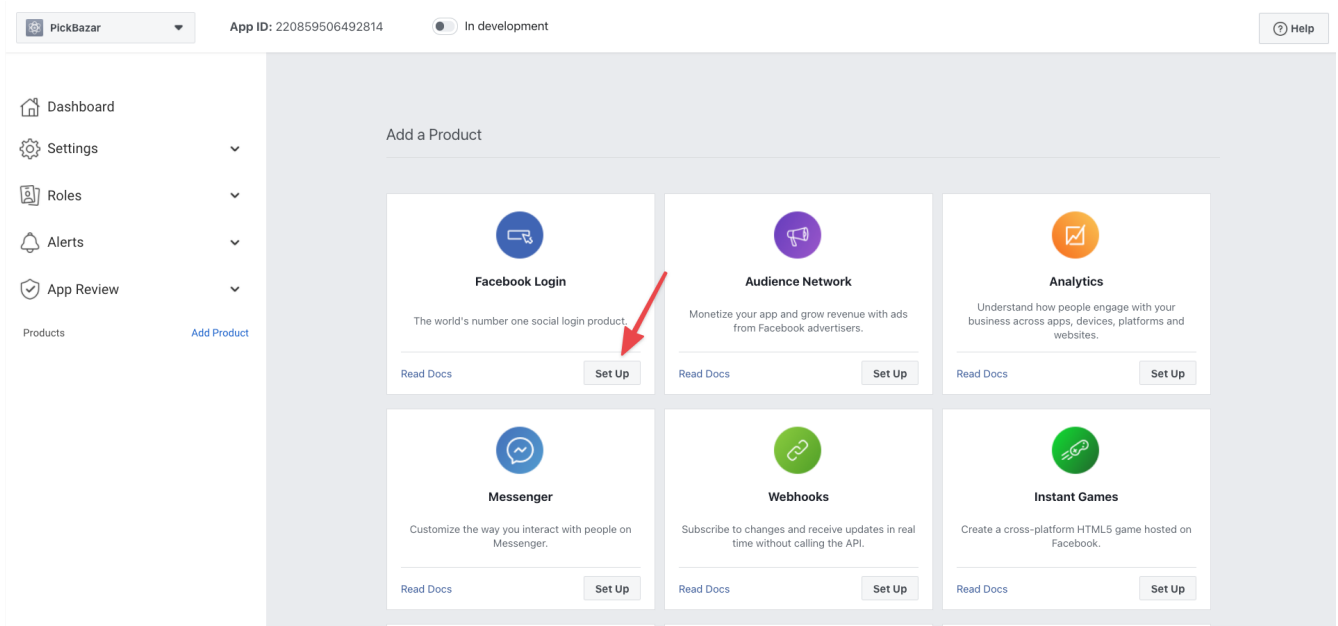
For REST API

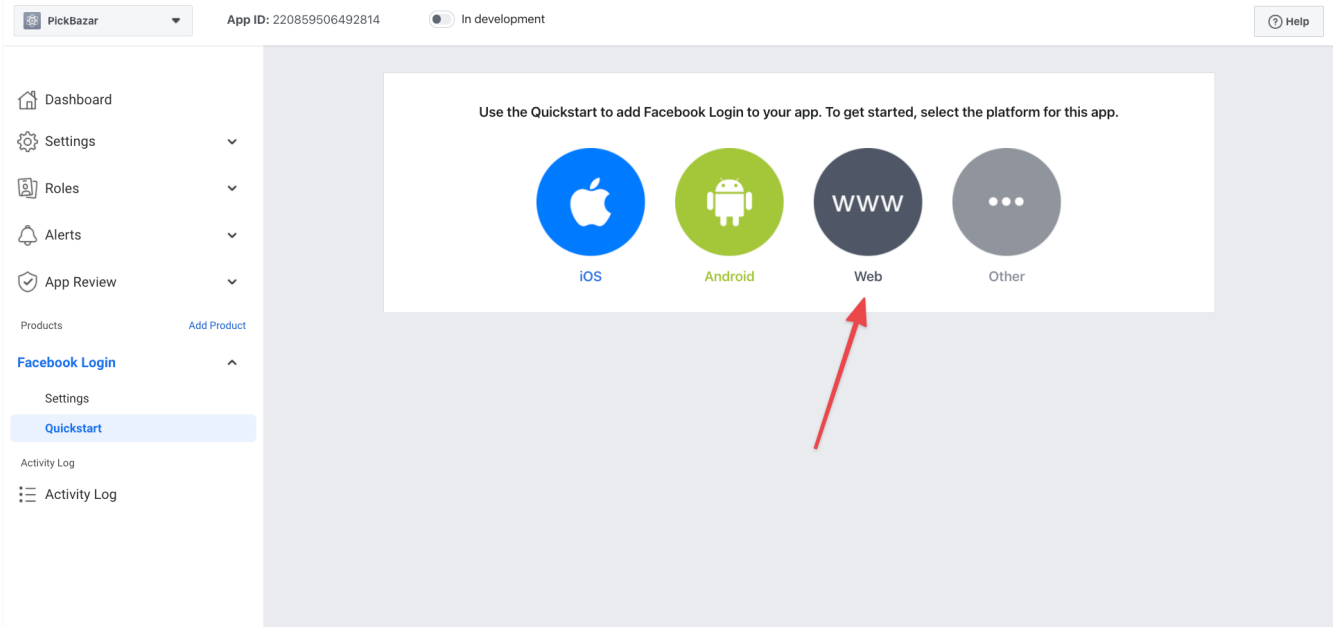
```

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

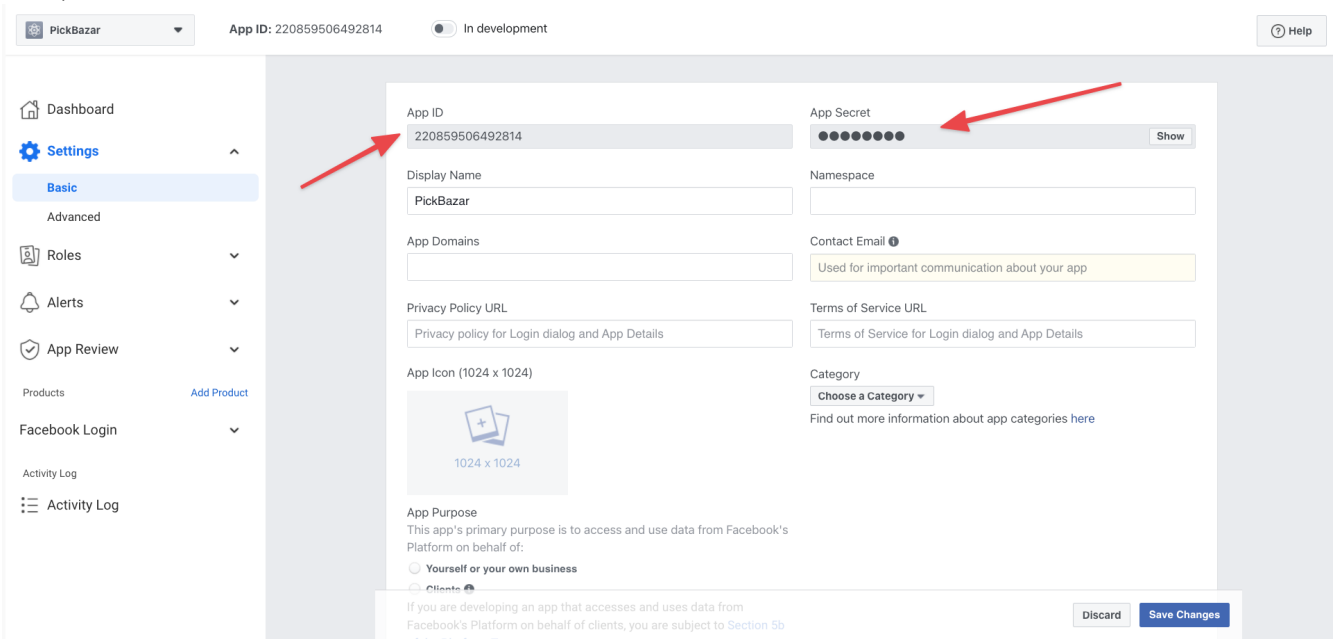
Facebook

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



```

1 NEXT_PUBLIC_REST_API_ENDPOINT=http://localhost/
2
3 NEXT_PUBLIC_SITE_URL=http://localhost:3000
4
5 NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=
6
7 # Social Authentication
8 NEXTAUTH_URL=http://localhost:3003
9
10
11 SECRET= # Linux: `openssl rand -hex 32` or go to https://generate-secret.now.sh/32
12
13 GOOGLE_ID=
14 GOOGLE_SECRET=
15
16 FACEBOOK_ID=
17 FACEBOOK_SECRET=
18
19 NEXT_PUBLIC_GOOGLE_MAP_API_KEY=ADD_GOOGLE_MAP_API_KEY
20
21 NEXT_PUBLIC_INSTAGRAM_BASIC_DISPLAY_USER_TOKEN=your_instagram_basic_display_user_token_here

```

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` and `FACEBOOK_CLIENT_SECRET` also add a `SECRET` code.

```

51 LIGHTHOUSE_CACHE_ENABLE=false
52 MEDIA_DISK=public
53
54 SHOP_URL=
55 DASHBOARD_URL=
56 ADMIN_EMAIL=support@example.com
57
58 GOOGLE_CLIENT_ID=
59 GOOGLE_CLIENT_SECRET=
60 GOOGLE_REDIRECT_URI=http://127.0.0.1/login/google/callback
61
62 FACEBOOK_CLIENT_ID=
63 FACEBOOK_CLIENT_SECRET=
64 FACEBOOK_REDIRECT_URI=https://127.0.0.1/login/facebook/callback
65
66 ACTIVE_OTP_GATEWAY=twilio
67
68 TWILIO_AUTH_TOKEN=
69 TWILIO_ACCOUNT_SID=
70 TWILIO_VERIFICATION_SID=
71
72 MESSAGEBIRD_API_KEY=
73 MESSAGEBIRD_ORIGINATOR=Marvel
74

```

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

```

OTP (Mobile Number Verification)

We implement the **OTP** feature on ChawkBazar **v2.2.0**. So if you want to use **OTP** with ChawkBazar, then make sure your ChawkBazar 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:

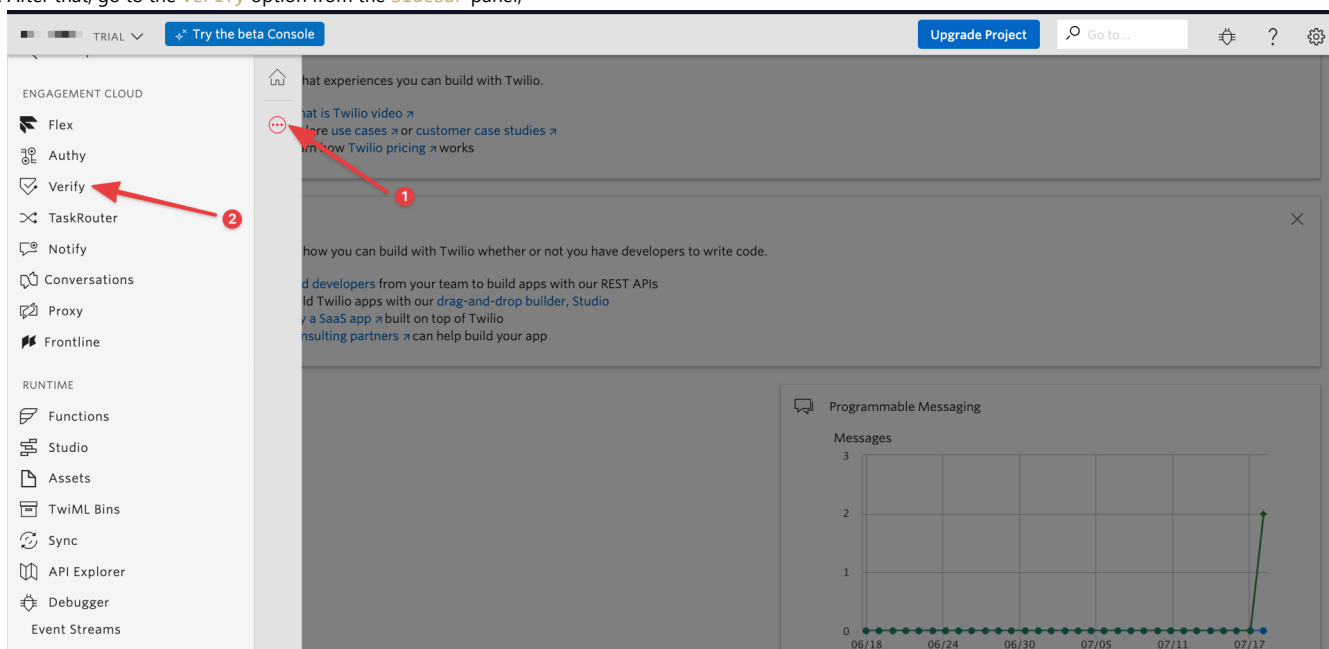
If you want to use [Twilio](#) as your OTP service provider, then follow this procedure,

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

```

51 LIGHTHOUSE_CACHE_ENABLE=false
52 MEDIA_DISK=public
53
54 SHOP_URL=
55 DASHBOARD_URL=
56 ADMIN_EMAIL=support@example.com
57
58 GOOGLE_CLIENT_ID=
59 GOOGLE_CLIENT_SECRET=
60 GOOGLE_REDIRECT_URI=http://127.0.0.1/login/google/callback
61
62 FACEBOOK_CLIENT_ID=
63 FACEBOOK_CLIENT_SECRET=
64 FACEBOOK_REDIRECT_URI=https://127.0.0.1/login/facebook/callback
65
66 ACTIVE_OTP_GATEWAY=twilio
67
68 TWILIO_AUTH_TOKEN=
69 TWILIO_ACCOUNT_SID=
70 TWILIO_VERIFICATION_SID=
71
72 MESSAGEBIRD_API_KEY=
73 MESSAGEBIRD_ORIGINATOR=Marvel
74
  
```

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 1

PickBazar

Insights
Logs
Settings 2
Webhooks

< Back

General Settings

Properties

SERVICE NAME

SERVICE SID 3

CODE LENGTH

Phone Number Validation

Numbers will be automatically validated to check that they are well formatted and can receive SMS before a verification code is sent.

LOOKUP Will return line type and carrier information in API response.

LANDLINE VALIDATION When enabled, Verify will not send SMS messages to landlines.

Transaction Verification

Send verification messages that include payee and amount to fulfill the PSD2 requirements of dynamic linking and Strong Customer Authentication (SCA). After enabling, both payee and amount parameters will be required to start verifications. [Learn more about PSD2](#)

PSD2 MODE

Text-to-speech

...

```

51 LIGHTHOUSE_CACHE_ENABLE=false
52 MEDIA_DISK=public
53
54 SHOP_URL=
55 DASHBOARD_URL=
56 ADMIN_EMAIL=support@example.com
57
58 GOOGLE_CLIENT_ID=
59 GOOGLE_CLIENT_SECRET=
60 GOOGLE_REDIRECT_URI=http://127.0.0.1/login/google/callback
61
62 FACEBOOK_CLIENT_ID=
63 FACEBOOK_CLIENT_SECRET=
64 FACEBOOK_REDIRECT_URI=https://127.0.0.1/login/facebook/callback
65
66 ACTIVE_OTP_GATEWAY=twilio
67
68 TWILIO_AUTH_TOKEN=
69 TWILIO_ACCOUNT_SID=
70 TWILIO_VERIFICATION_SID=
71
72 MESSAGEBIRD_API_KEY=
73 MESSAGEBIRD_ORIGINATOR=Marvel
74

```

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.

Contact

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

SupportPricing

API Getting Started

Your getting started guide

SMS API • 4 steps left

20%

✓ Create your account

2

2 Get started with free test credits

3

3 Install the official SDK for SMS

4

4 Send your first SMS

5

5 Go live with SMS

6

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%

1 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

0 Free SMS Remaining

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

1

Show

Test

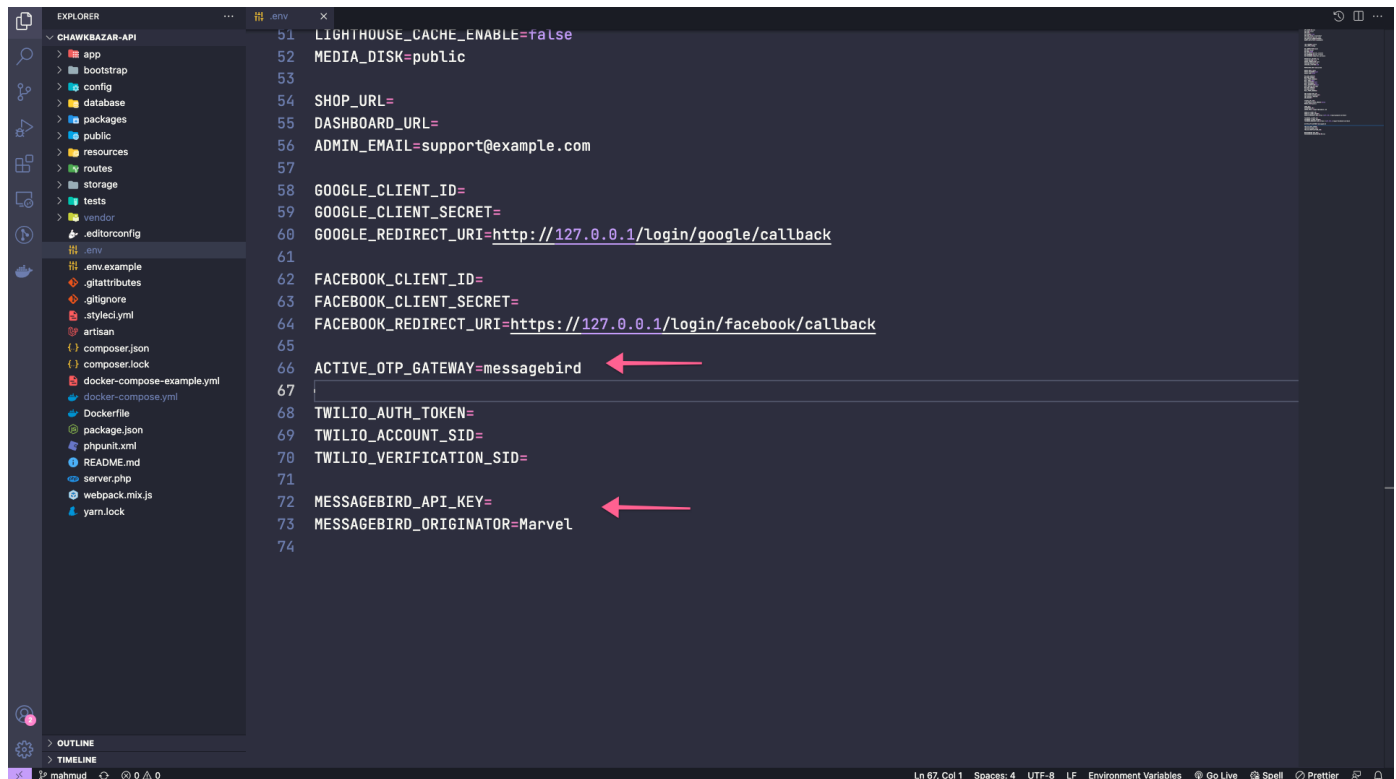
Use this API Key in your test environment. It will return a response but not work in production or deduct balance.

Manage API Keys

Resources

SMS API Quickstarts

SMS API Tutorials



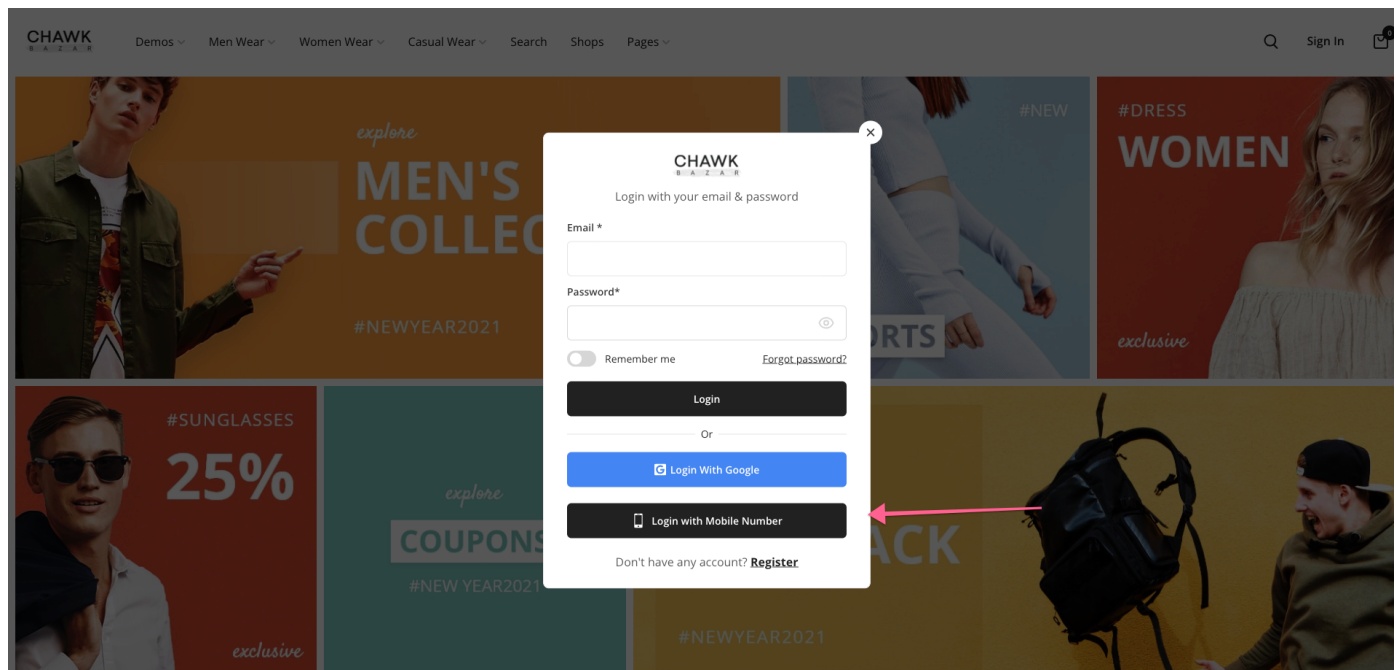
```
51 LIGHTHOUSE_CACHE_ENABLE=false
52 MEDIA_DISK=public
53
54 SHOP_URL=
55 DASHBOARD_URL=
56 ADMIN_EMAIL=support@example.com
57
58 GOOGLE_CLIENT_ID=
59 GOOGLE_CLIENT_SECRET=
60 GOOGLE_REDIRECT_URI=http://127.0.0.1/login/google/callback
61
62 FACEBOOK_CLIENT_ID=
63 FACEBOOK_CLIENT_SECRET=
64 FACEBOOK_REDIRECT_URI=https://127.0.0.1/login/facebook/callback
65
66 ACTIVE_OTP_GATEWAY=messagebird
67
68 TWILIO_AUTH_TOKEN=
69 TWILIO_ACCOUNT_SID=
70 TWILIO_VERIFICATION_SID=
71
72 MESSAGEBIRD_API_KEY=
73 MESSAGEBIRD_ORIGINATOR=Marvel
74
```

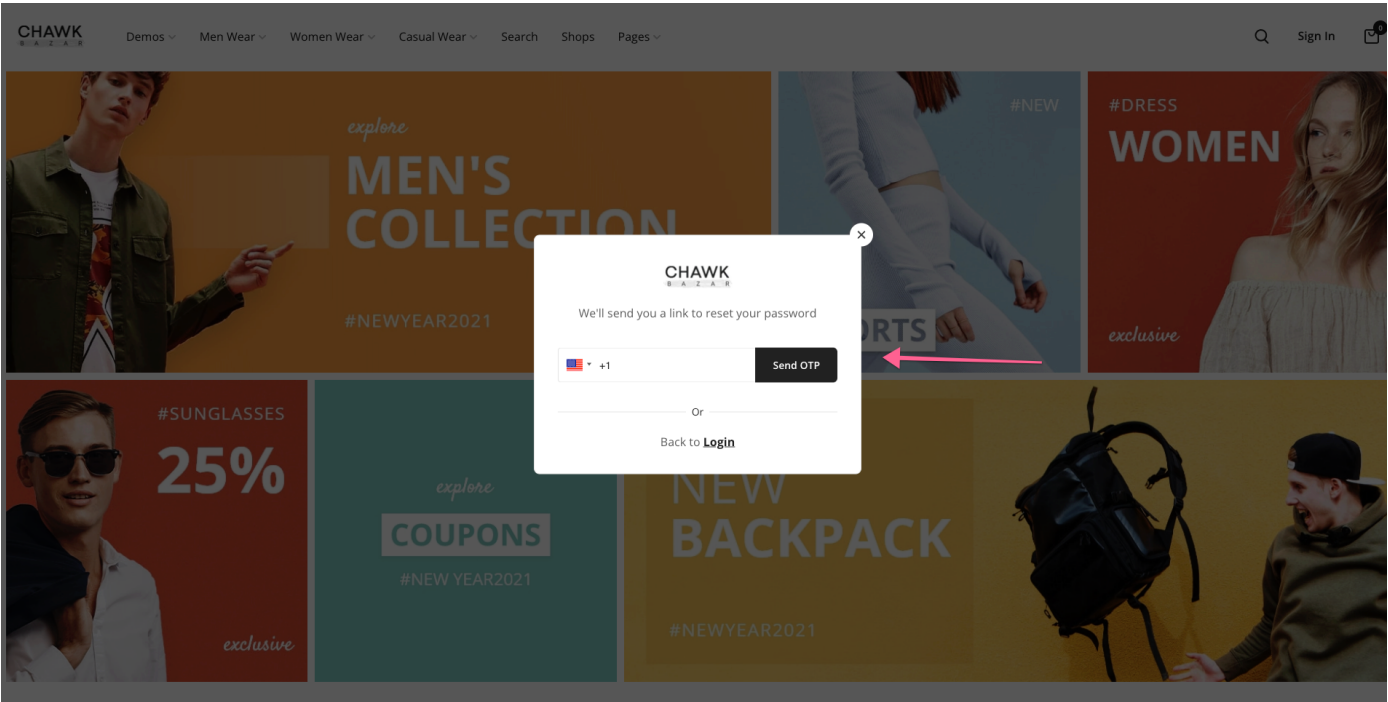
With that, **MessageBird** is appropriately configured for the OTP Service.

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.

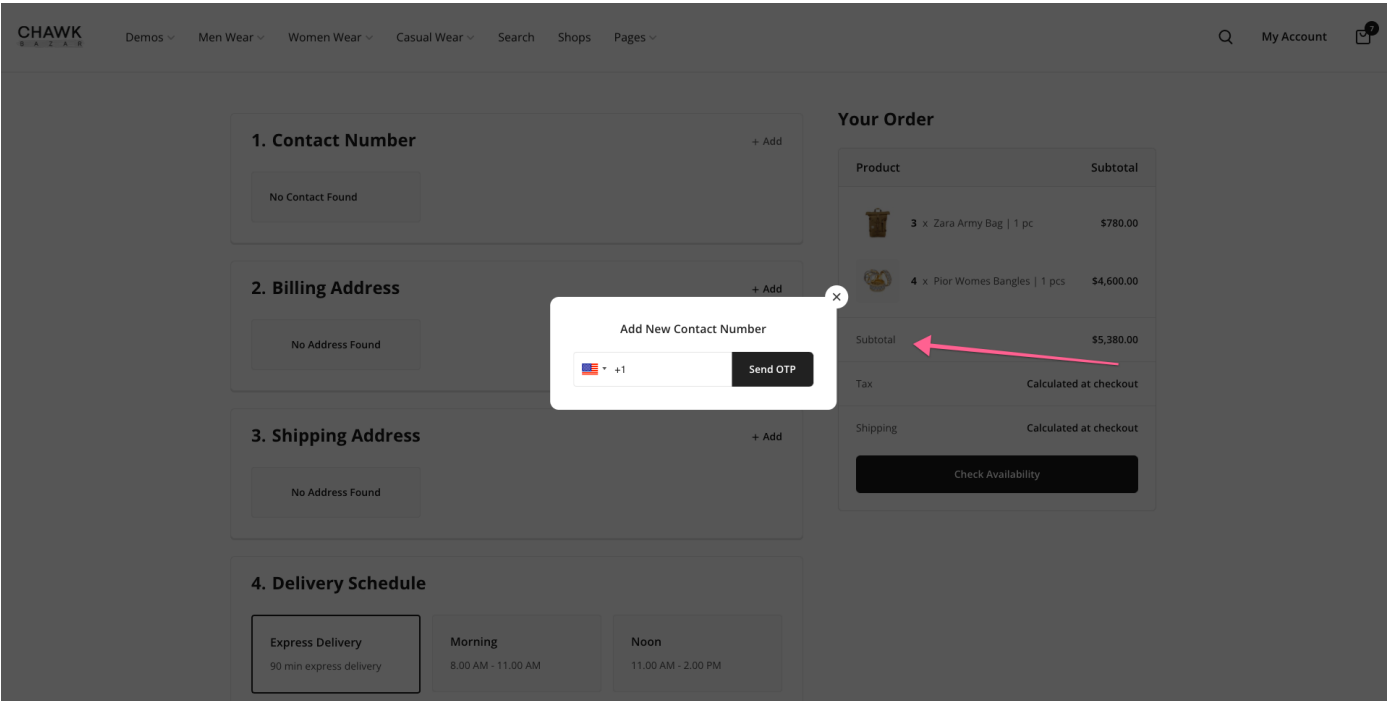




OTP Verification During Checkout Process:

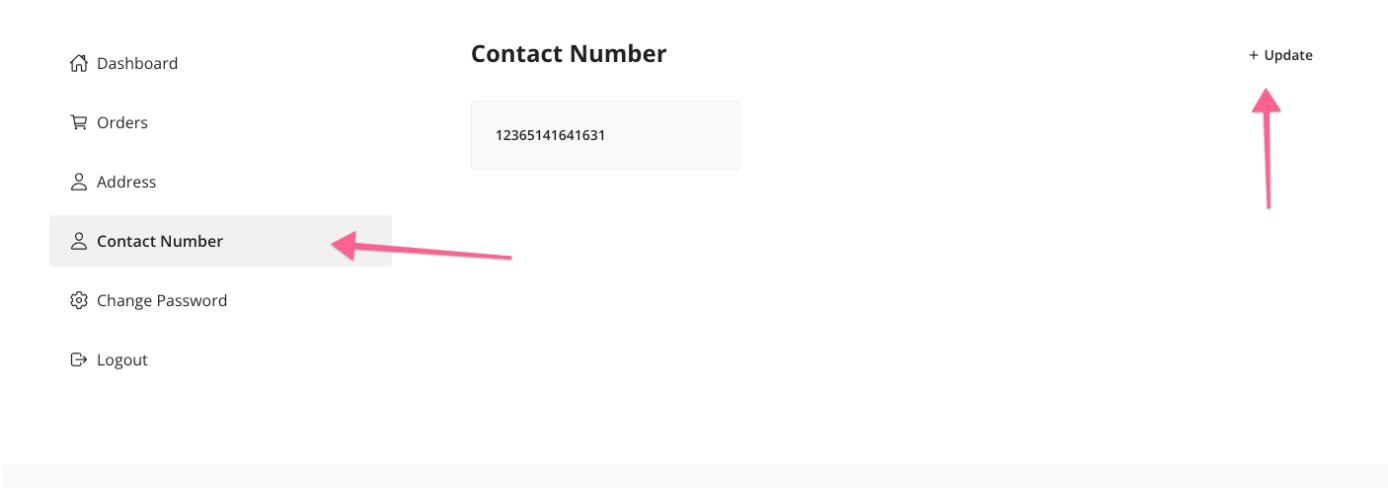
Similarly, when users go to the checkout page, they'll get a form to verify their mobile number.

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

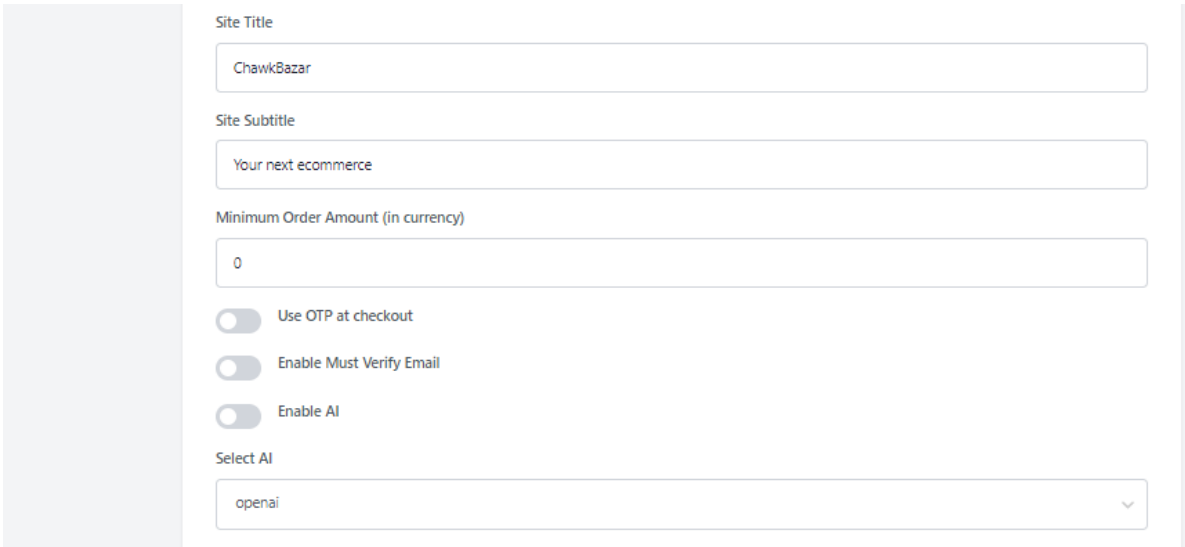
We implement the **Email Verification** feature on Chawkbazar v5.1.0. So if you want to use **Email Verification** with Chawkbazar, then make sure your Chawkbazar is v5.1.0 or later.

With the Email Verification feature, you can do,

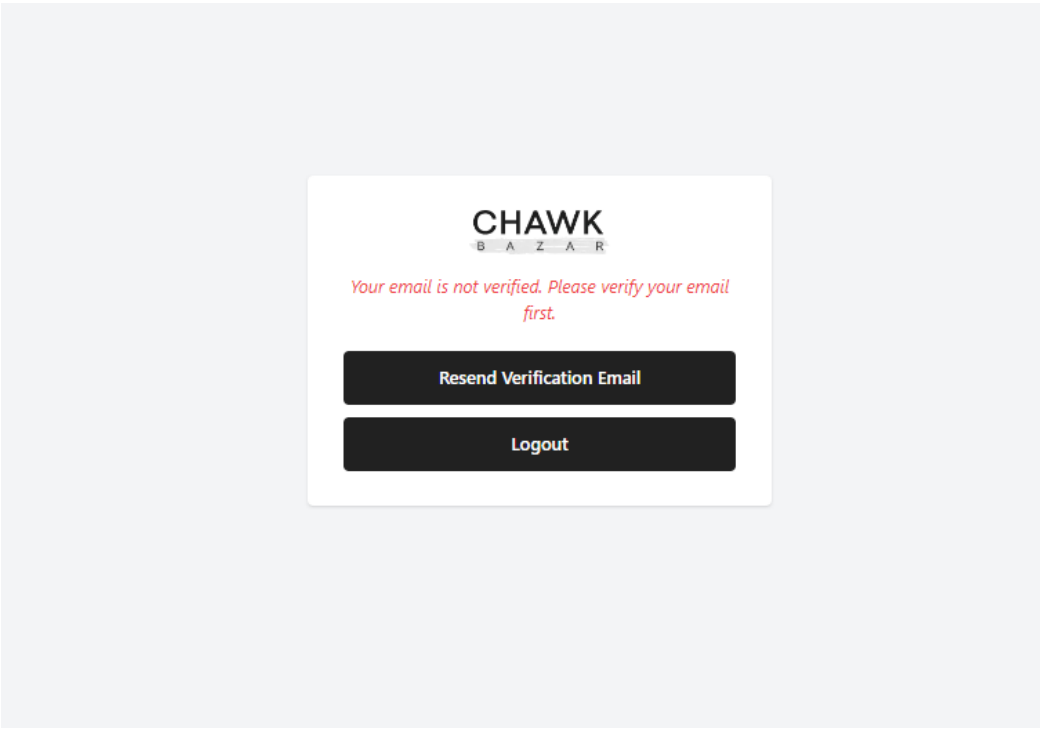
- Get Email After Registration
- Email verification for updating Email in profile update section
- When Email verification is on in settings then users will be forced to verify their email.

How to turn on this feature

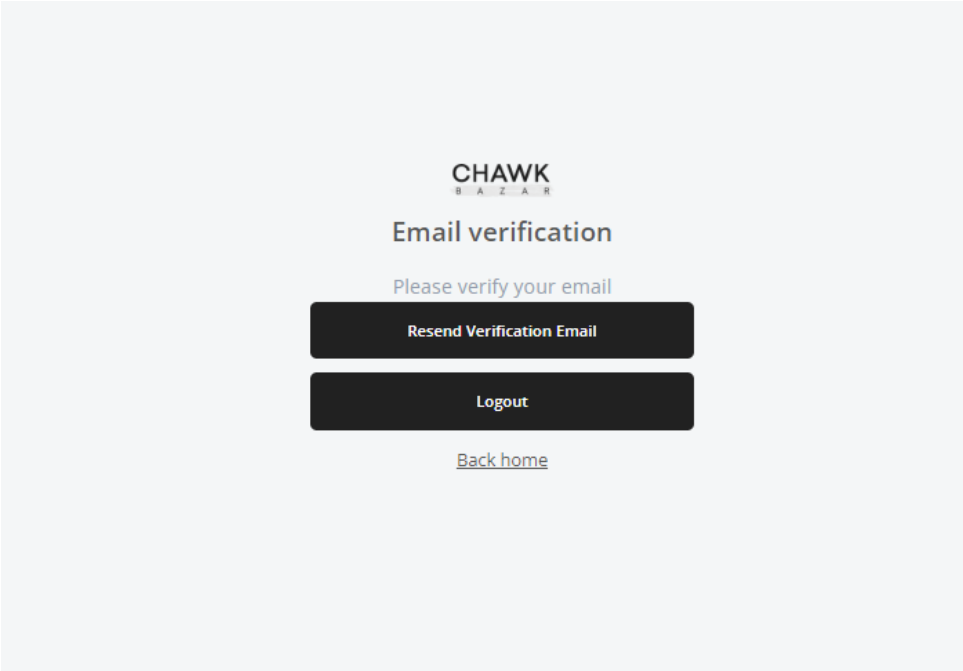
To turn on this feature go to settings of your site then find **Enable Must Verify Email** switch.



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
```

Payment

We have introduced the new **Payment** architecture feature on ChawkBazar **v3.0.0**. So if you want to use **Payment** with **ChawkBazar**, then make sure your ChawkBazar is **v3.0.0** or later.

We have plan to enrich this feature area by integrating more payment gateways in future. List of available payment gateways now.

- **Stripe**
- **PayPal**
- **Rezorpay**
- **Mollie**
- **Paystack**

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 ChawkBazar 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 ChawkBazar.

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 ChawkBazar 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)

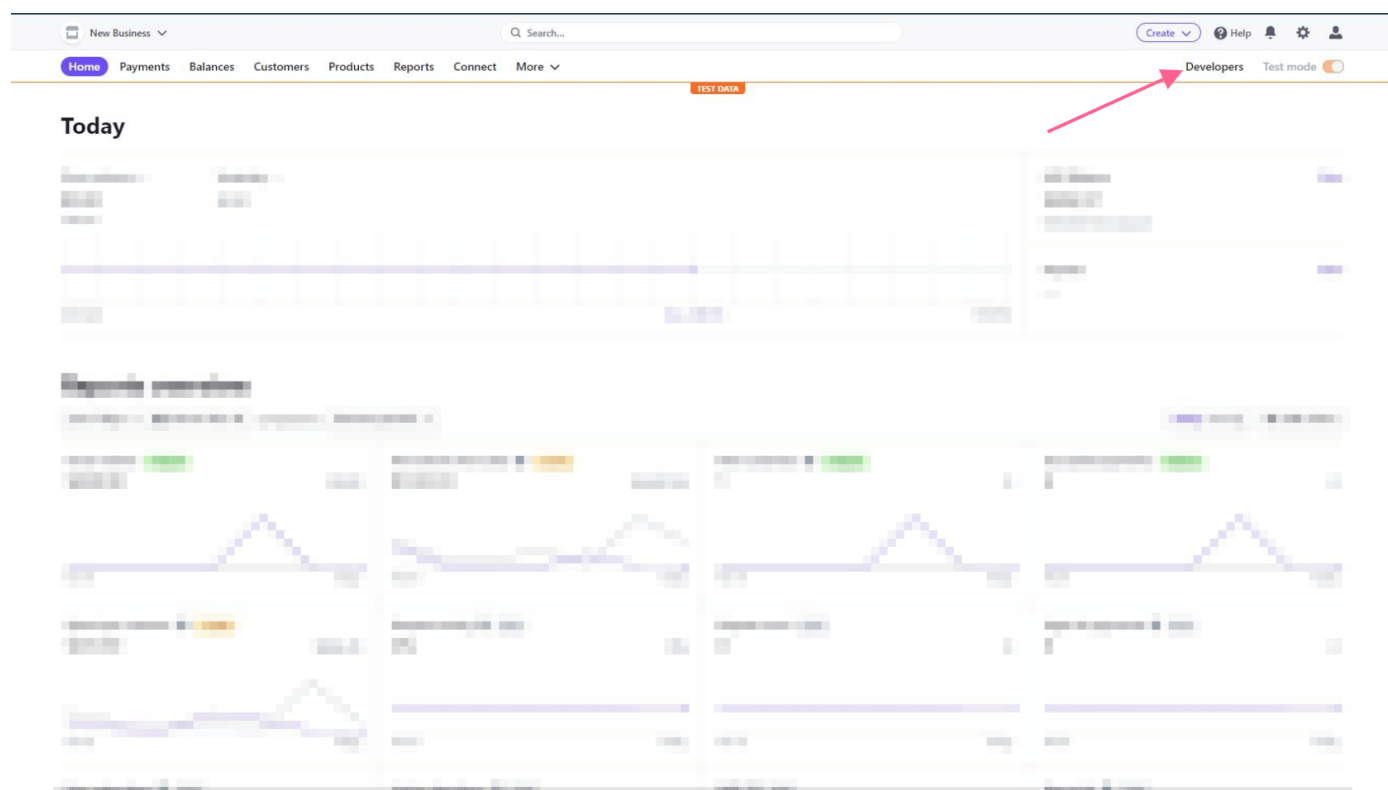
The screenshot displays the ChawkBazar admin dashboard. The sidebar on the left contains navigation links for various sections. The main area shows a summary of key metrics: Total Revenue (Last 30 Days) at \$0.00, Total Order (Last 30 Days) at 9, Todays Revenue at \$0.00, and Total Shops at 2. Below these metrics is a Sales History chart showing data from January to December. Further down are tables for Recent Orders and Recent Withdrawals. The Settings page is highlighted, showing the Payment configuration section. It includes a toggle for 'Enable Cash On Delivery', a dropdown for 'Select Payment Gateway' (currently set to Stripe), and a text field for 'Webhook URL' with a copy icon.

- Add Stripe publishable key inside **.env** file of ChawkBazar shop.

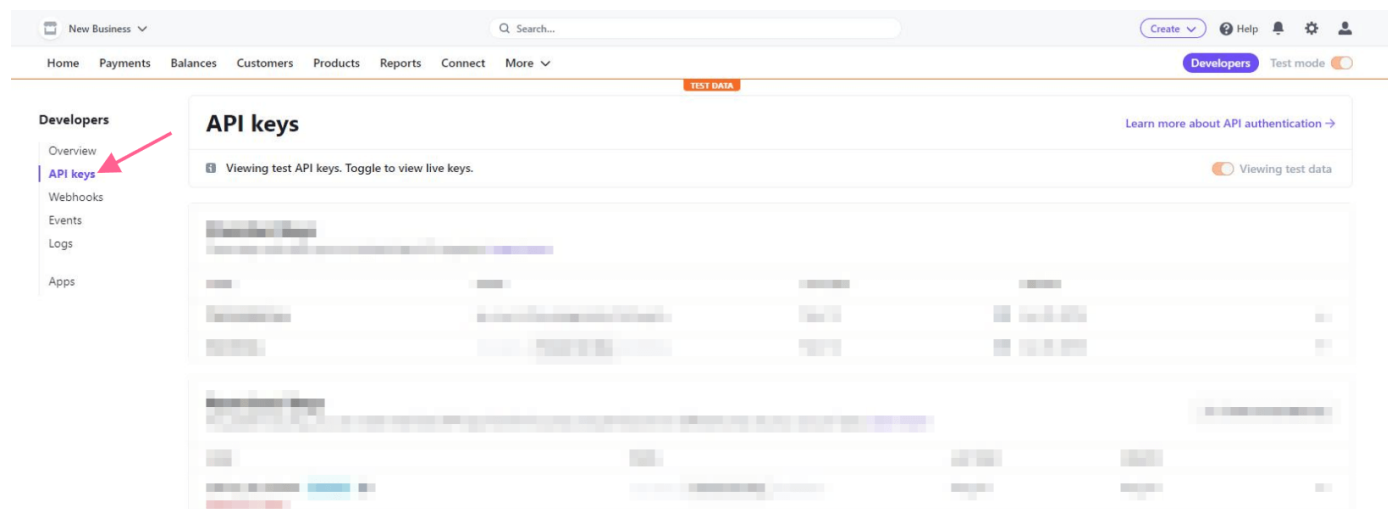
```
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=[YOUR_STRIPE_PUBLISHABLE_KEY]
```

How to create & setup Stripe information properly?

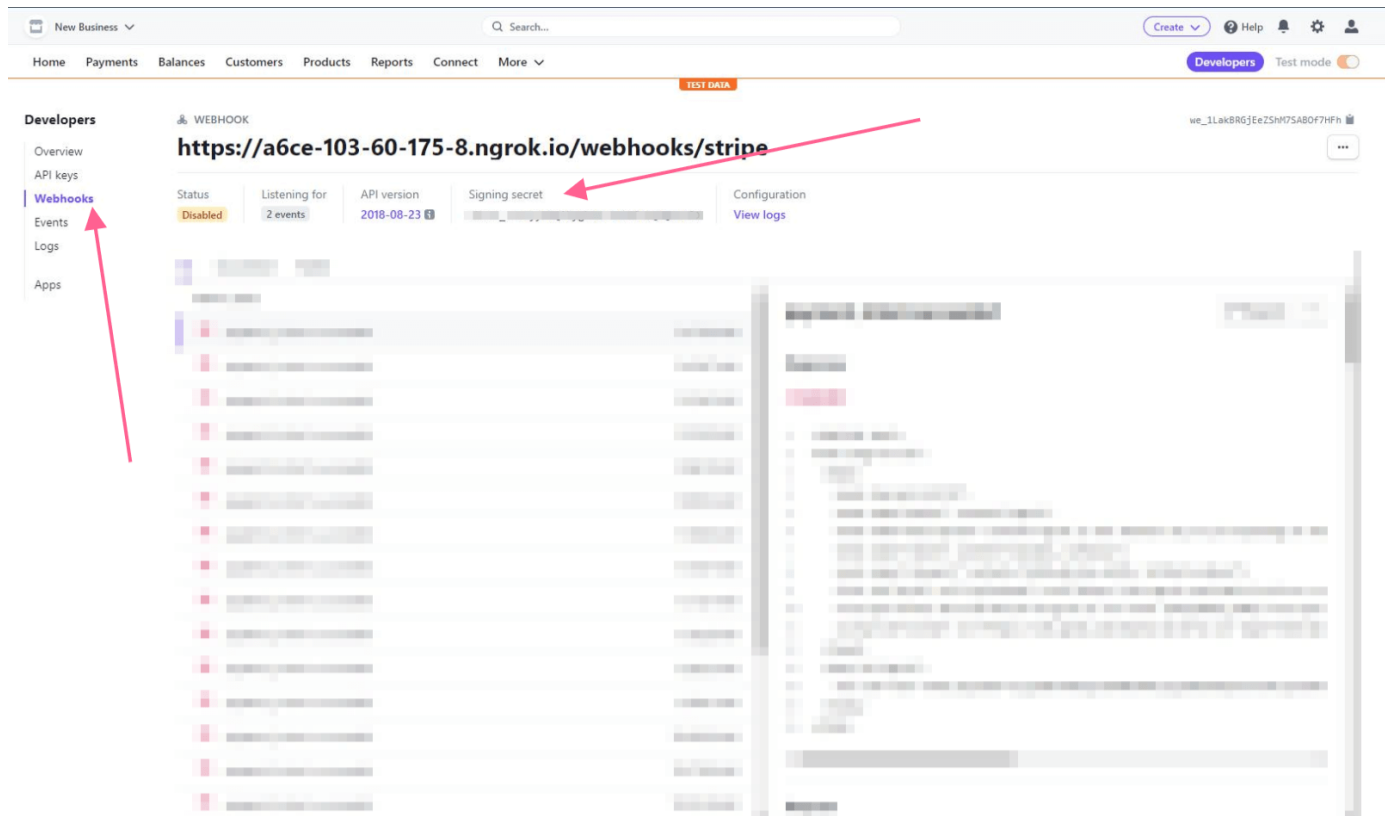
- Go to the stripe website and login <https://stripe.com> If you aren't a registered user, the complete the stripe registration first.
- After logged in into stripe dashboard, follow the developer link to create the API keys.



- 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.

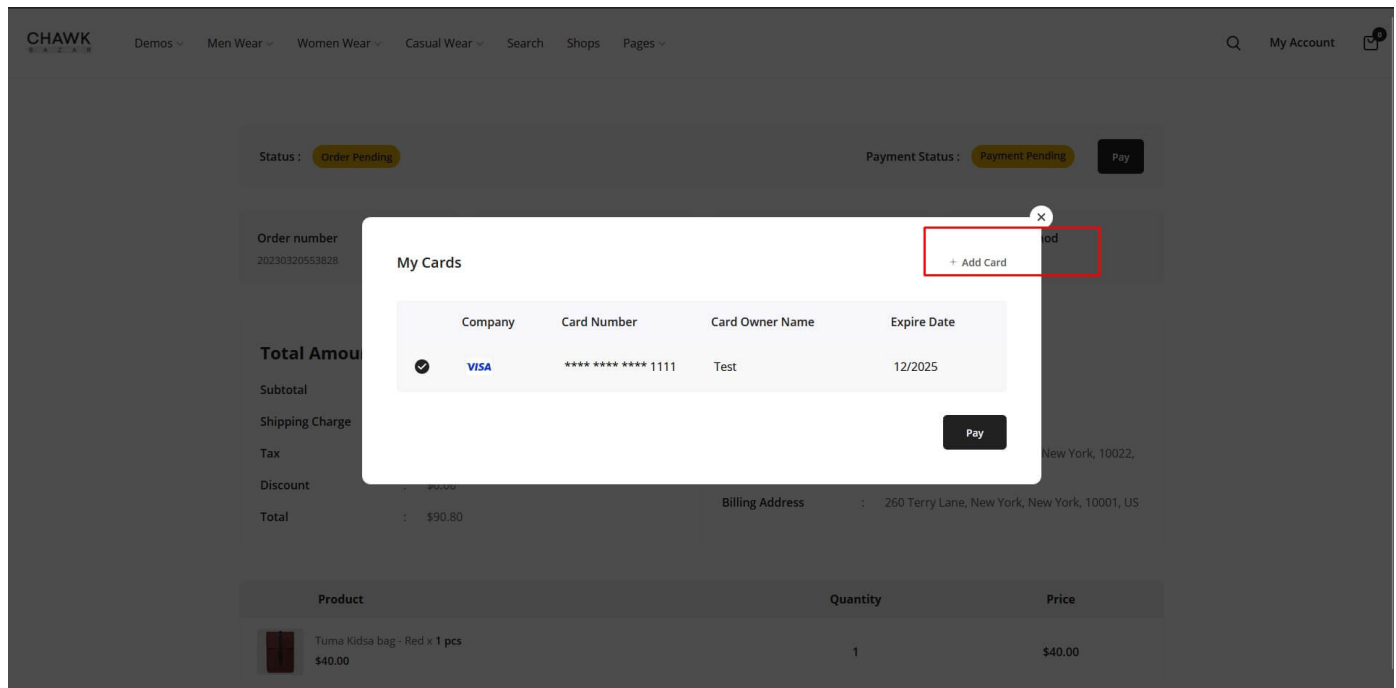
- Stripe DOCS [home page] -- <https://stripe.com/docs>
- Stripe API Docs -- <https://stripe.com/docs/api>
- Stripe JavaScript Docs -- <https://stripe.com/docs/js>

How can I add card in my user profile for future payments in Stripe?

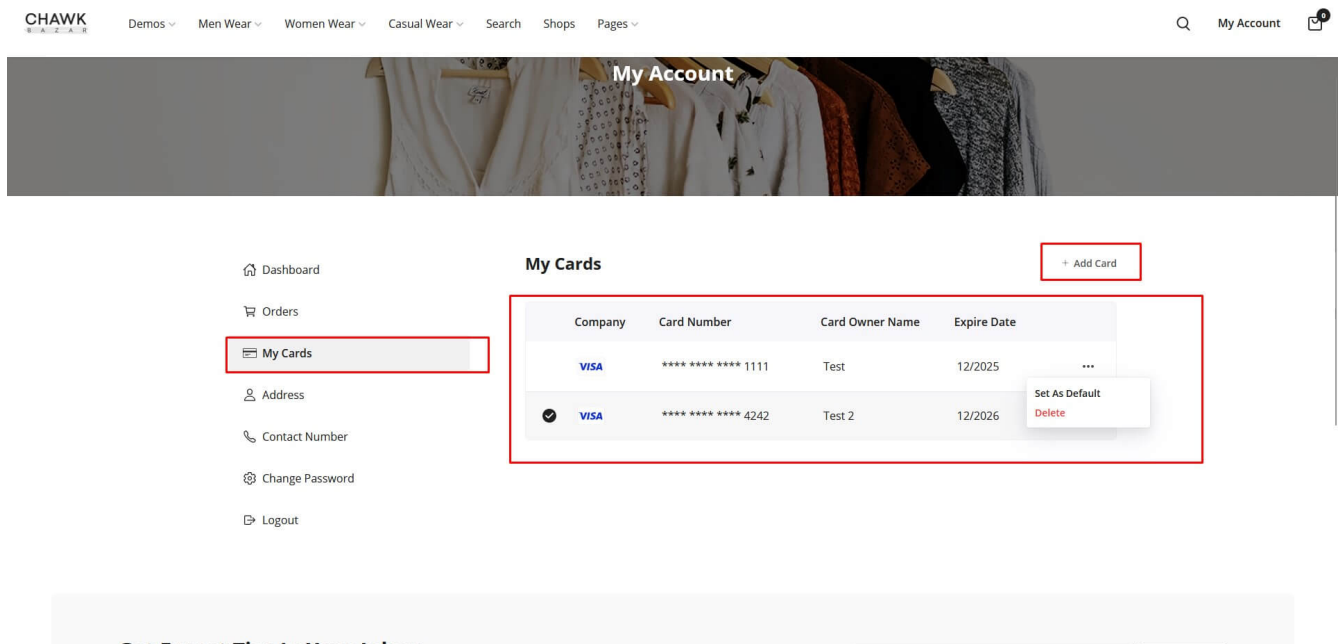
In ChawkBazar we have provided an feature for saving card in case of future usages. This is an **on-session** process. So, if a customer wants to pay via Cards, s/he must have present in the application lively. No **off-session** payment was applied here.

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 some things that need to be kept in mind.

Please note

- No confidential information was saved in this feature. By maintaining the guideline only available information which is permitted to be saved via Stripe is implemented here.
- Guest user can't save card for future payment.

PayPal

PayPal is an American multinational financial technology company operating an online payments system. In ChawkBazar 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 integrates inside ChawkBazar.

Please follow & complete these 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 ChawkBazar 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 shows the ChawkBazar admin dashboard. The sidebar on the left contains navigation links: Dashboard, Shops, My Shops, Products, Attributes, Brands, Categories, Tags, Orders, Users, Coupons, Taxes, Shippings, Withdrawals, and Settings (highlighted with a red box). The main content area displays summary cards for Total Revenue (\$0.00), Total Order (9), Today's Revenue (\$0.00), and Total Shops (2). Below these is a Sales History chart. At the bottom, there are tables for Recent Orders and Recent Withdrawals. The Recent Orders table shows an order with tracking number 20230318657888, total \$611.00, and status 'Order Processing'. The Recent Withdrawals table shows a withdrawal of \$500.00 from 'Chawkbazar Vendor shop' with status 'Approved'. The bottom section, titled 'Payment', allows configuring payment options, including enabling 'Cash On Delivery', selecting 'Paypal' as the payment gateway, and setting a webhook URL.

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 ChawkBazar app .env file like mentioned above.

The screenshot shows the PayPal Developer dashboard. On the left sidebar, 'My Apps & Credentials' is highlighted. The main content area is titled 'Default Application'. It shows the 'App display name' as 'Default Application'. Below this is the 'SANDBOX API CREDENTIALS' section, which contains 'Sandbox Account', 'Client ID', and 'Secret'. The 'Secret' field has a 'show' link. Below this is the 'SANDBOX APP SETTINGS' section, which includes 'Return URL' and 'App feature options'.

- You will find the Webhook ID and link too.

The screenshot shows the 'Sandbox Webhooks' section of the PayPal Developer dashboard. It includes a search bar and a table of webhooks. The table has columns for 'Webhook', 'Webhook Id', and 'Event Tracked'. One webhook is listed with the event 'Payment capture completed, Payment capture denied, Payment capture pending, Payment capture refunded, Payment capture reversed'. There is an 'Add Webhook' button at the bottom.

- 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/>

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.

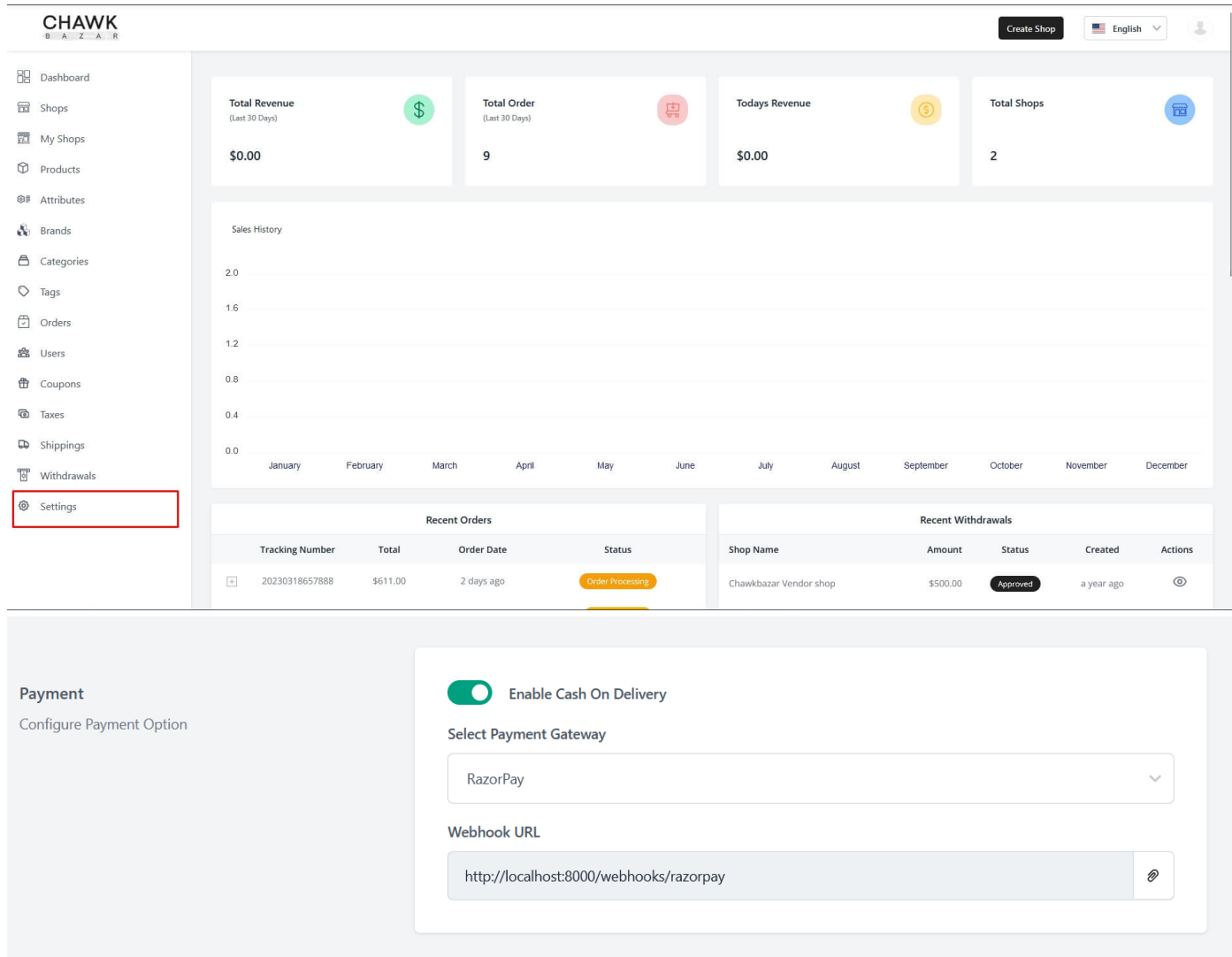
RazorPay integration inside PickBazar.

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)



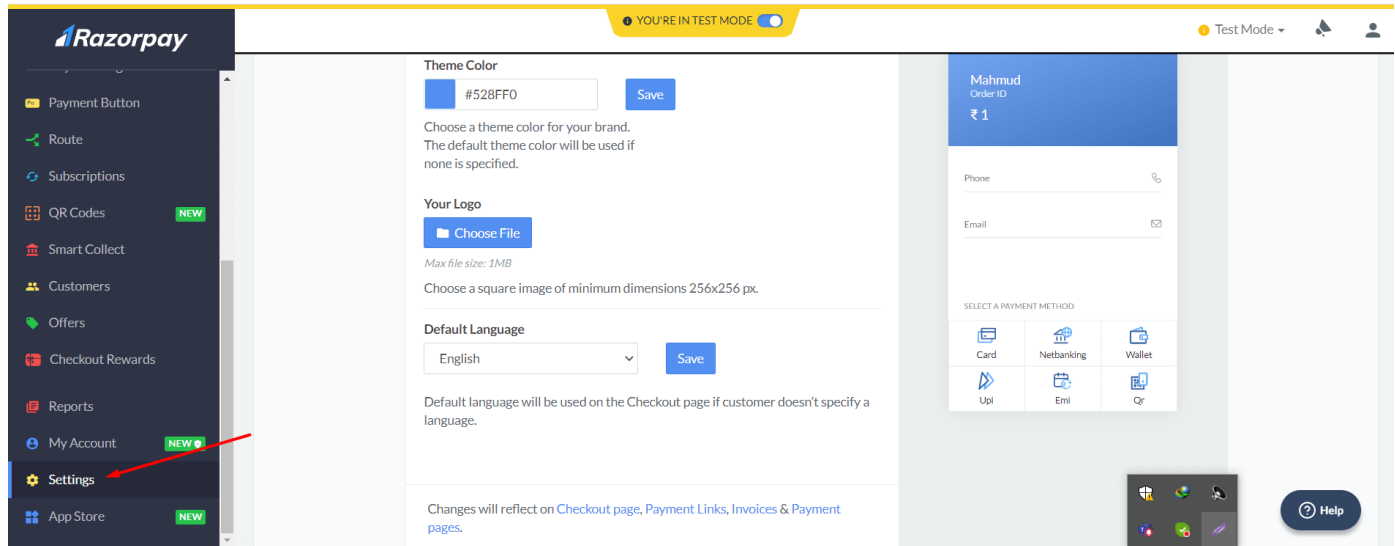
The screenshot displays the CHAWK BAZAR admin dashboard. The left sidebar contains a menu with items like Dashboard, Shops, My Shops, Products, Attributes, Brands, Categories, Tags, Orders, Users, Coupons, Taxes, Shippings, Withdrawals, and Settings (highlighted with a red box). The main content area shows a dashboard overview with four cards: Total Revenue (\$0.00), Total Order (9), Todays Revenue (\$0.00), and Total Shops (2). Below these is a Sales History chart. At the bottom, there are tables for Recent Orders and Recent Withdrawals. The 'Payment' settings section is expanded, showing options to enable Cash On Delivery, select a payment gateway (RazorPay is selected), and set a webhook URL (http://localhost:8000/webhooks/razorpay).

- Add RazorPay publishable key inside `.env` file of pickbazar shop.

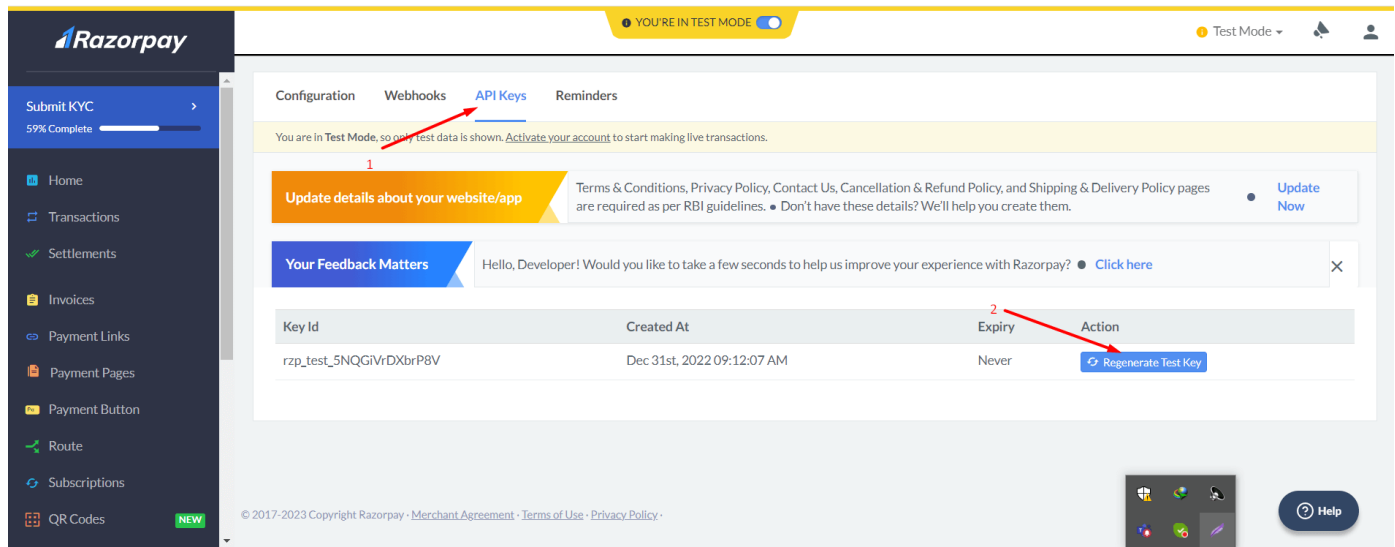
```
NEXT_PUBLIC_RAZORPAY_KEY=[YOUR_RAZORPAY_KEY_ID]
```

How to create & setup RazorPay information properly?

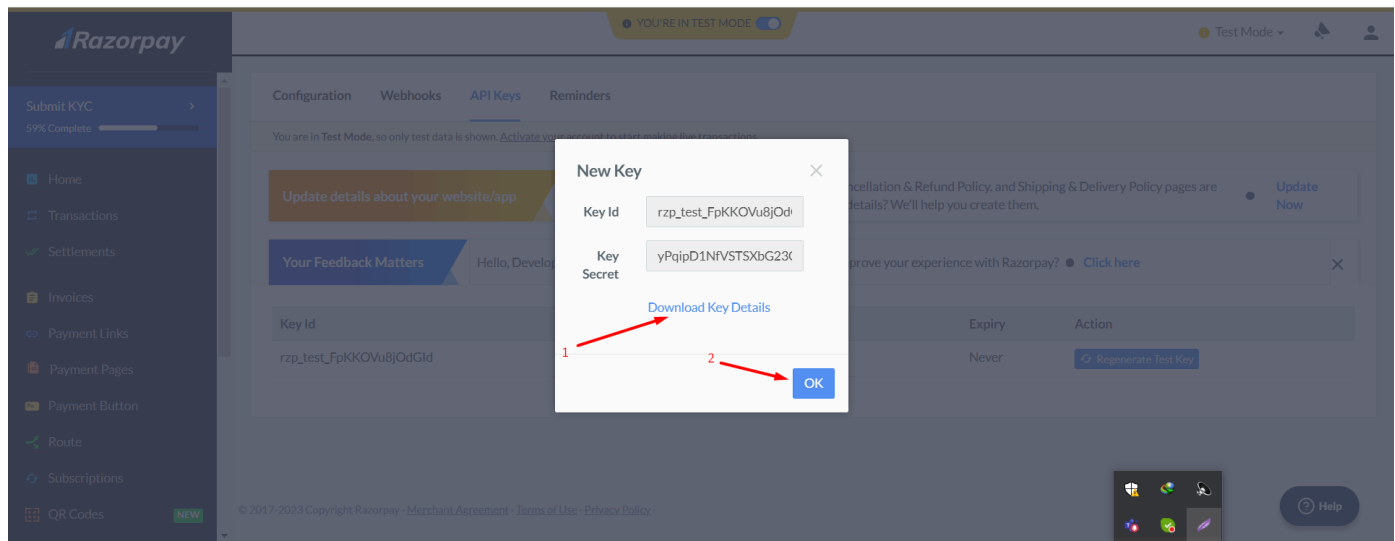
- Go to the RazorPay website & login <https://razorpay.com> If you aren't a registered user, Complete the RazorPay registration first.
- After logged in into RazorPay dashboard, Click on Settings



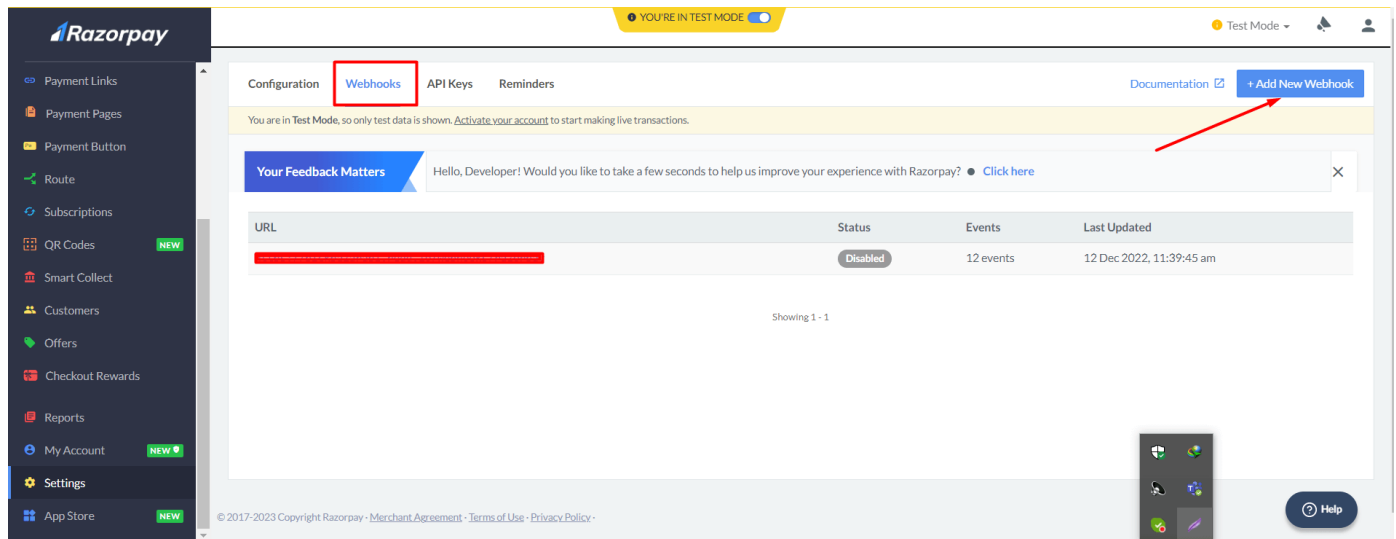
- Select API Keys option & click on Generate Test Key.



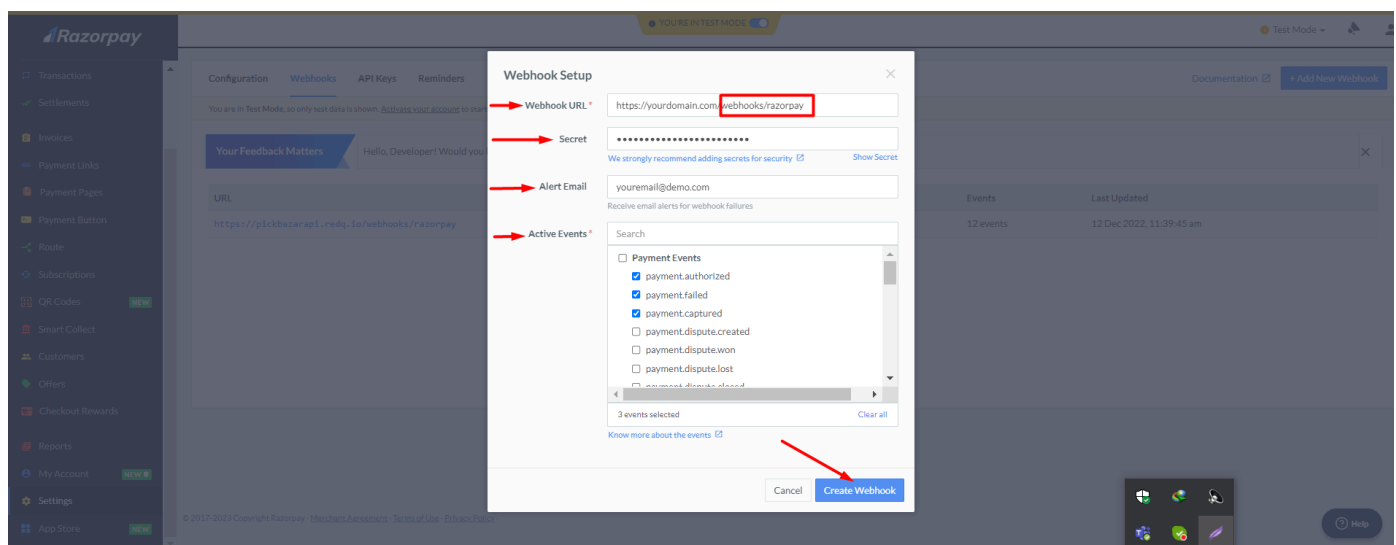
- The Client Id and Secret Key will appear. Download these keys and click on OK



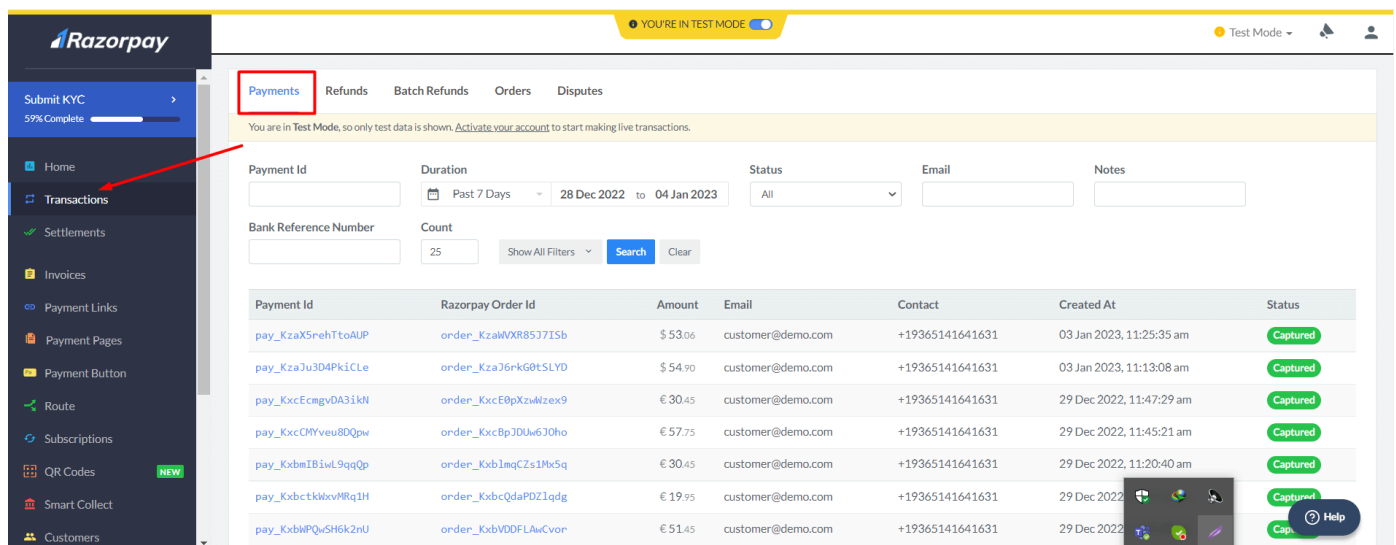
- Select Webhooks option & click on Add New Webhooks.



- 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>

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)

CHAWK
B A Z A R

Create Shop

English

Dashboard

Shops

My Shops

Products

Attributes

Brands

Categories

Tags

Orders

Users

Coupons

Taxes

Shippings

Withdrawals

Settings

Total Revenue
(Last 30 Days)

\$0.00

Total Order
(Last 30 Days)

9

Todays Revenue

\$0.00

Total Shops

2

Sales History

2.0

1.6

1.2

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
20230318657888	\$611.00	2 days ago	Order Processing

Recent Withdrawals

Shop Name	Amount	Status	Created	Actions
Chawkbazar Vendor shop	\$500.00	Approved	a year ago	

Payment

Configure Payment Option

Enable Cash On Delivery

Select Payment Gateway

Mollie

Webhook URL

http://localhost:8000/webhooks/mollie

- 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.

```

ngrok
Check which logged users are accessing your tunnels in real time https://ngrok.com/s/app-users

Session Status      online
Account
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
  
```

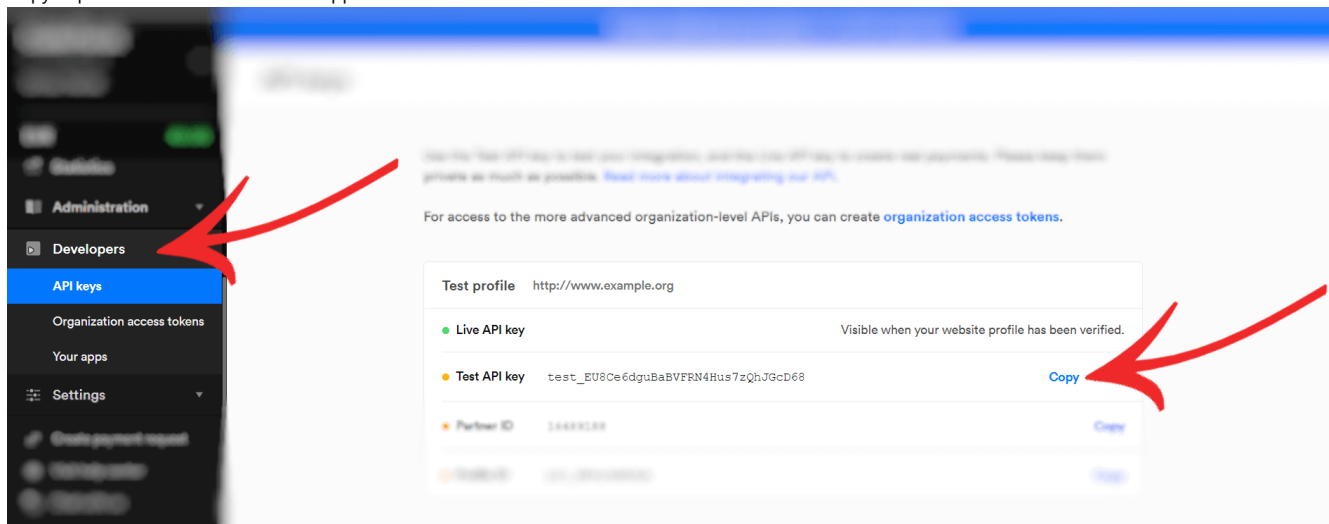
- Copy the forwarding https link & paste .env file after link add line- /webhooks/mollie

```

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>

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)

CHAWK

B A Z A A R

Create ShopEnglish

Dashboard

Shops

My Shops

Products

Attributes

Brands

Categories

Tags

Orders

Users

Coupons

Taxes

Shippings

Withdrawals

Settings

Total Revenue

Last 30 Days

\$0.00

Total Order

Last 30 Days

9

Todays Revenue

\$0.00

Total Shops

2

Sales History

Recent Orders

Tracking Number	Total	Order Date	Status
20230318657888	\$611.00	2 days ago	Order Processing

Recent Withdrawals

Shop Name	Amount	Status	Created	Actions
Chawkbazar Vendor shop	\$500.00	Approved	a year ago	

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 Callback URL

Test Webhook URL

https://pickbazarapi.com/webhooks/paystack

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

(Ctrl+C to quit)

Check which logged users are accessing your tunnels in real time https://ngrok.com/s/app-users

Session Statusonline

Account

Version3.1.0

RegionIndia (in)

Latency116ms

Web Interfacehttp://127.0.0.1:4040

Forwardinghttps://c852-37-111-243-195.in.ngrok.io -> http://localhost:8000

Connections

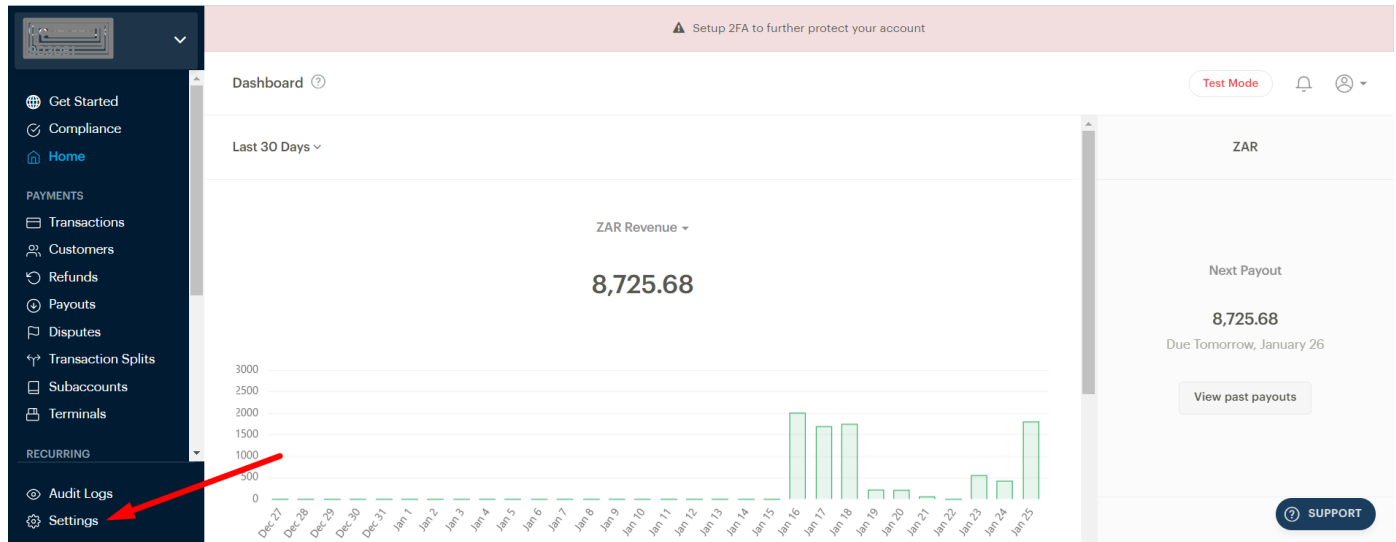
tll	opn	rt1	rt5	p50	p90
0	0	0.00	0.00	0.00	0.00

- Copy the forwarding https link & paste on Test Webhook URL, after paste forwarding https link, add line- /webhooks/paystack.

Test Callback URL	<input type="text" value="Test Callback URL"/>
Test Webhook URL	<input type="text" value="https://dd15-2400-c600-345f-88ce-a1d9-b41-3c43-cf65.in.ngrok.io/webhooks/paystack"/>
<input type="button" value="Save Changes"/>	

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 in Paystack. The left sidebar is the same as the dashboard. The main area has tabs for Profile, Contact, Accounts, Team, Preferences, and API Keys & Webhooks. The 'API Keys & Webhooks' tab is active. It contains fields for 'Test Secret Key' (with a 'Generate new secret key' link), 'Test Public Key' (with a copy icon), 'Test Callback URL', and 'Test Webhook URL' (pre-filled with 'https://yourapi.com/webhooks/paystack'). A 'Save Changes' button is at the bottom. A red arrow points to the 'Test Secret Key' field. At the bottom, there is a link to 'Need help with your integration? Check out our API documentation' and a 'SUPPORT' button.

- 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 \(USD\)](#)

lyzico

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.

lyzico integrate inside Chawkbazar.

Please follow & complete this steps for lyzico 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 lyzico.

```
# 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 Chawkbazar 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 Chawkbazar admin dashboard. The sidebar on the left contains navigation links: Dashboard, Shops, My Shops, Products, Attributes, Brands, Categories, Tags, Orders, Users, Coupons, Taxes, Shippings, Withdrawals, and Settings (highlighted with a red box). The main content area shows four summary cards: Total Revenue (\$0.00), Total Order (9), Today's Revenue (\$0.00), and Total Shops (2). Below these is a Sales History chart and two tables: Recent Orders and Recent Withdrawals. The Settings page is shown below, with a red arrow pointing to the 'lyzico' payment gateway option. The settings page includes options for enabling cash on delivery, currency (US Dollar), enabling the gateway, and selecting a payment gateway. The 'lyzico' gateway is selected, and its webhook URL is shown as `https://chawkbazarapi.redq.io/admin/webhooks/lyzico`, which is highlighted with a red box.

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 you are testing your development environment.
- Click on **Setting** then **Merchant Settings**.
- You will get your Sandbox API credentials. Copy & paste those inside Chawkbazar 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>

Integration of new payment gateway

To integrate a payment gateway in Chawkbazar-Laravel, you will need to follow these general steps:

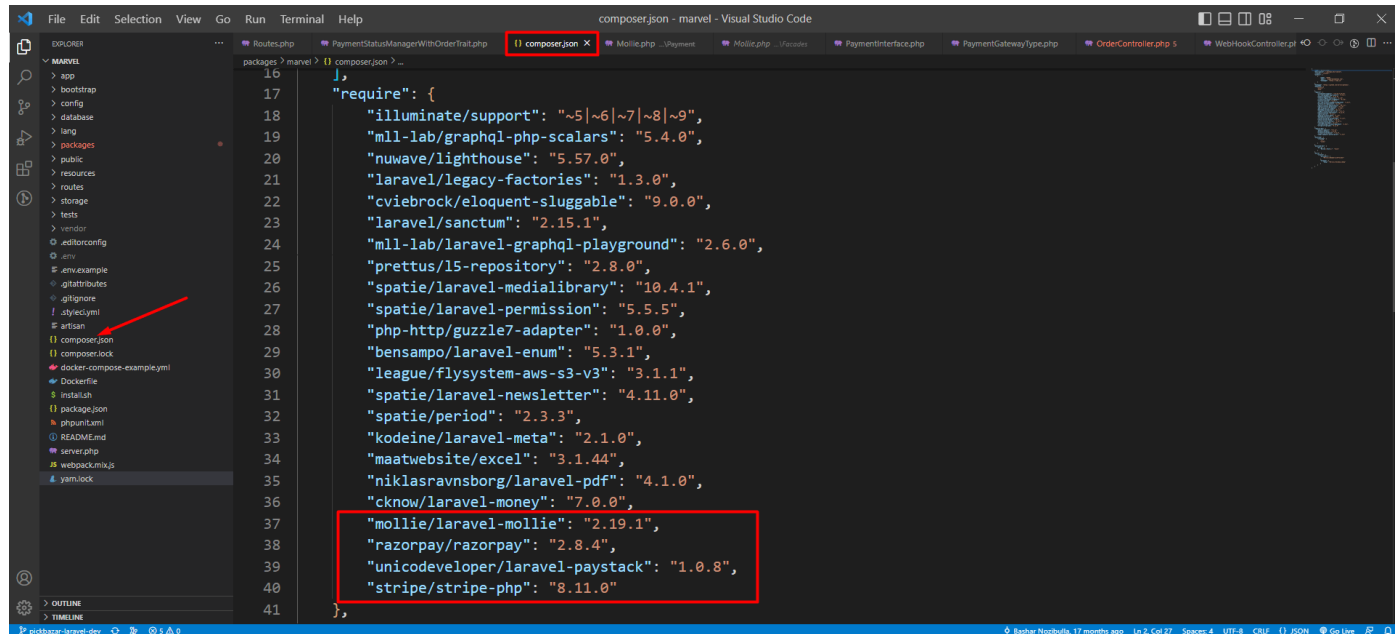
Getting Started with API

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.

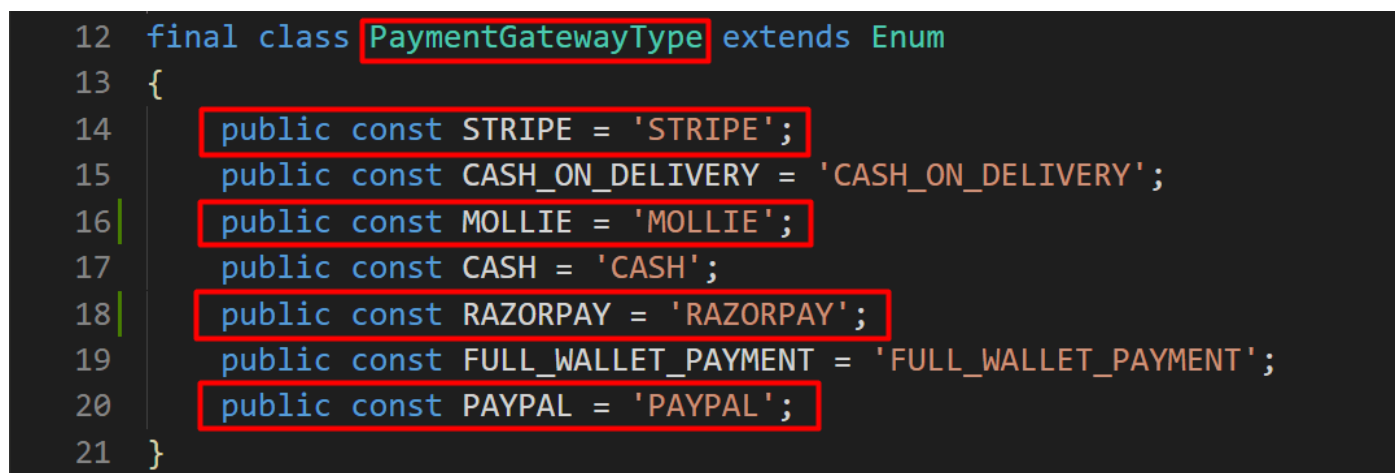
Once you have chosen and installed a package, you will need to configure it by adding your payment gateway credentials and any other required settings.

To check the installed dependencies of payment gateway in ChawkBazar-Laravel, you can open the composer.json file in a text editor and find that.



Step 2: Add payment gateway name in the Enum.

Add PaymentGateway Enum `API -> package -> marvel -> src -> Enums -> PaymentGatewayType.php`



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 PaymentTrait;
20
21     public Api $api;
22
23     public function __construct()
24     {
25         parent::__construct();
26         $this->api = new Api($key: config($key: 'shop.razorpay.key_id'), $secret: config($key: 'shop.razorpay.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     {

```

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 }

```

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             $this->updatePaymentOrderStatus($request, OrderStatus::PENDING, PaymentStatus::FAILED);
115     }
116
117     // To prevent loop for any case
118     http_response_code(200);
119     exit();
120 }
121
122 /**
123  * Update Payment and Order Status
124  *
125  * @param $request
126  * @param $orderStatus
127  * @param $paymentStatus
128  * @return void
129  */
130 public function updatePaymentOrderStatus($request, $orderStatus, $paymentStatus): void
131 {
132     $payload = $request->payload['payment']['entity'];
133     $paymentIntent = PaymentIntent::whereJsonContains('payment_intent_info', ['payment_id' => $payload['order_id']])->first();
134     $trackingId = $paymentIntent->tracking_number;
135     $order = Order::where('tracking_number', '=', $trackingId)->first();
136     $this->webhookSuccessResponse($order, $orderStatus, $paymentStatus);
137 }
138
139 /**
140  * createCustomer
141  *
142  * @param mixed $request
143  * @return array
144  */
145 public function createCustomer($request): array
146 {
147     return [];
148 }
149
150 /**
151  * attachPaymentMethodToCustomer
152  *
153  * @param string $retrieved_payment_method
154  * @param object $request
155  * @return object
156  */
157 public function attachPaymentMethodToCustomer(string $retrieved_payment_method, object $request): object
158 {
159     return (object) [];
160 }
161
162 /**
163  * detachPaymentMethodToCustomer
164  *
165  * @param string $retrieved_payment_method
166  * @return object
167  */
168 public function detachPaymentMethodToCustomer(string $retrieved_payment_method): object
169 {
170     return (object) [];
171 }
172
173 public function retrievePaymentIntent($payment_intent_id): object
174 {
175     return (object) [];
176 }
177
178 /**
179  * confirmPaymentIntent
180  *
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
201 /**
202  * retrievePaymentMethod
203  *
204  * @param string $method_key
205  * @return object
206  */
207 public function retrievePaymentMethod(string $method_key): object
208 {
209     return (object) [];
210 }
211 }
212

```

*** Note : *** It is important to note that each payment gateway has its own set of requirements and may have different methods for processing payments. You will need to follow the official documentation of that specific payment gateway.

Step 4: Using that payment gateway for submitting the order.

Go to API -> package -> marvel -> src -> Http -> Controllers -> OrderController.php -> submitPayment then you've to add your PaymentGatewayType and function name in switch case.

```

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(['products', 'children.shop', 'wallet_point', 'payment_intent'])
366             ->findOneByFieldOrFail('tracking_number', $tracking_number);
367
368         switch ($order->payment_gateway) {
369             case PaymentGatewayType::STRIPE:
370                 $this->stripe($order, $request, $settings: $this->settings);
371                 break;
372             case PaymentGatewayType::PAYPAL:
373                 $this->paypal($order, $request, $settings: $this->settings);
374                 break;
375             case PaymentGatewayType::MOLLIE:
376                 $this->mollie($order, $request, $settings: $this->settings);
377                 break;
378             case PaymentGatewayType::RAZORPAY:
379                 $this->razorpay($order, $request, $settings: $this->settings);
380                 break;
381         }
382     } catch (Exception $e) {
383         throw new Exception($e->getMessage());
384     }
385 }

```

After That go to API -> package -> marvel -> src -> Traits -> PaymentStatusManagerWithOrderTrait.php for verify your payment status update, you've to add function like Stripe, PayPal, Razorpay or Mollie.

```

PaymentStatusManagerWithOrderTrait.php - marvel - Visual Studio Code
107 public function razorpay(Order $order, Request $request, Settings $settings): void
108 {
109     try {
110         // for single gateway options
111         if (isset($order->payment_intent)) {
112             foreach ($order->payment_intent as $key => $intent) {
113                 if (strtolower($settings->options['paymentGateway']) === $order->payment_gateway) {
114                     $chosen_intent = $intent;
115                 }
116             }
117         }
118
119         $paymentId = isset($chosen_intent->payment_intent_info) ? $chosen_intent->payment_intent_info['payment_id'] : null;
120         if (isset($paymentId)) {
121             $paymentStatus = Payment::verify($paymentId);
122             if ($paymentStatus) {
123                 switch (strtolower($paymentStatus)) {
124                     case "paid":
125                         $this->paymentSuccess($order);
126                         break;
127                     case "attempted":
128                         $this->paymentProcessing($order);
129                         break;
130                     case "failed":
131                         $this->paymentFailed($order);
132                         break;
133                 }
134             }
135         }
136     } catch (Exception $e) {
137         throw new Exception($e->getMessage(), 500, "SOMETHING WENT WRONG WITH PAYMENT");
138     }
139 }

```

Step 5: How to Setup webhook in ChawkBazar-laravel follow the steps

To use 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.

First go to **API** -> **package** -> **marvel** -> **src** -> **Routes** then add a post route.

```
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']);
```

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): 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, $eventStatus::PENDING, $eventStatus::PROCESSING);
109             break;
110         case 'captured':
111             $this->updatePaymentOrderStatus($request, $eventStatus::PROCESSING, $eventStatus::SUCCESS);
112             break;
113         case 'failed':
114             $this->updatePaymentOrderStatus($request, $eventStatus::PENDING, $eventStatus::FAILED);
115             break;
116     }
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

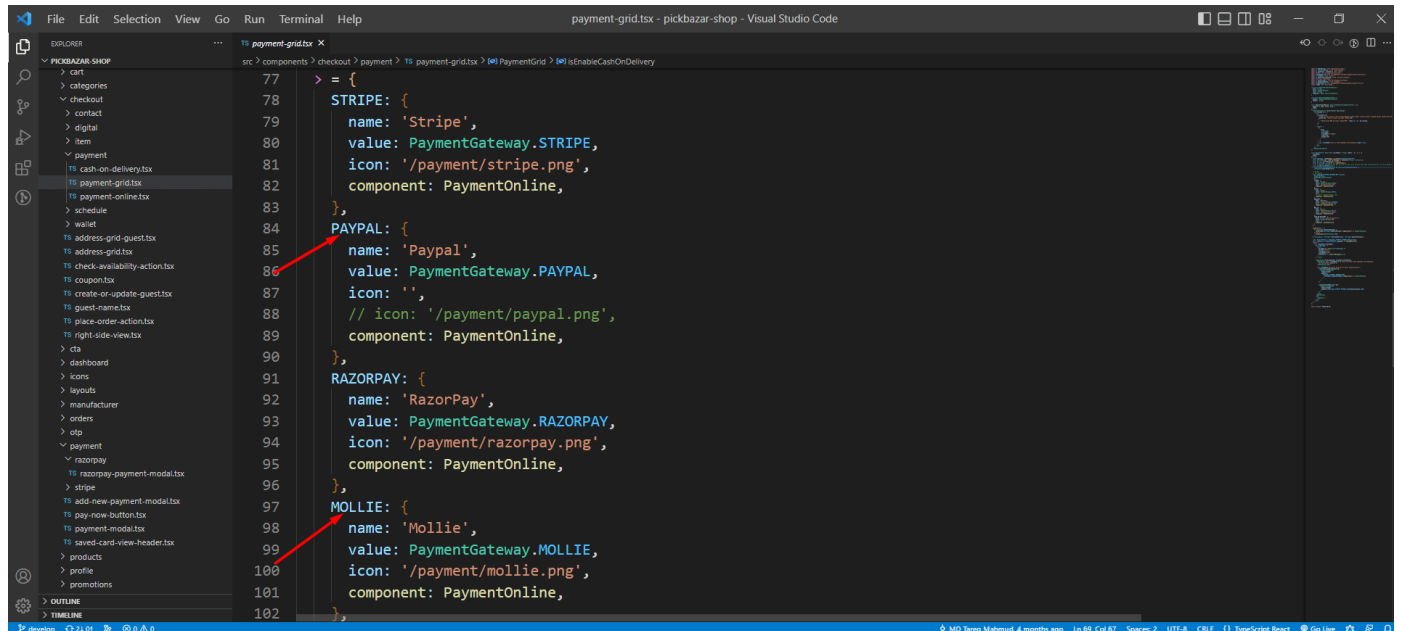
First go to `shop -> src -> types -> index.ts -> export enum PaymentGateway` then add PaymentGateway Name.

```

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 }

```

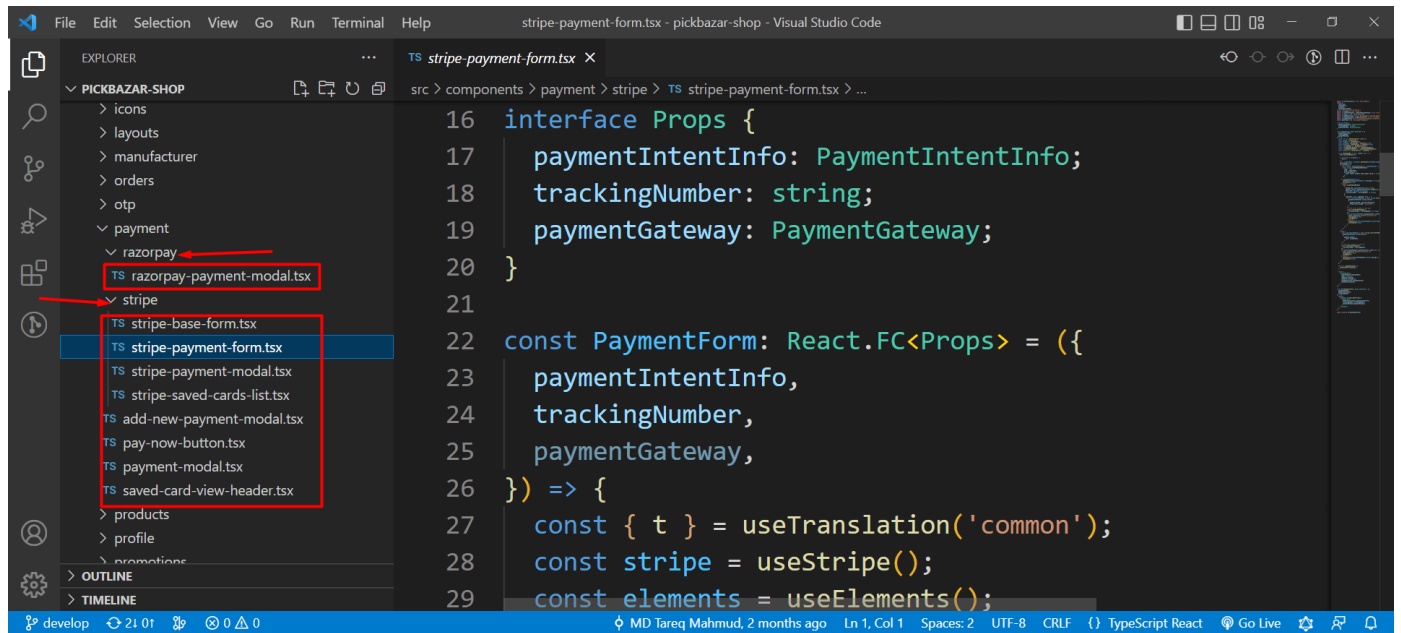
after that go to `shop -> src -> components -> checkout -> payment -> Payment-grid.tsx` then add your PaymentGateway object.



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.

How to activate multiple payment gateways [in Admin] ?

In admin dashboard there are new settings introduced for this feature. Please check this screenshot first,

Payment
Configure Payment Option

☐ Enable Cash On Delivery

Currency
US Dollar

☐ Enable Gateway

Select Payment Gateway

stripe PayPal Razorpay mollie paystack

Set default payment gateway
Stripe

Webhook URL

Gateway	Webhook URL	Copy
stripe	http://localhost:8000/webhooks/stripe	
PayPal	http://localhost:8000/webhooks/paypal	
Razorpay	http://localhost:8000/webhooks/razorpay	

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.

1. Select a payment gateway during place order

Please check this screenshot first,

CHAWK

Demos Men Wear Women Wear Casual Wear Search Shops Pages

My Account

1. Contact Number + Update

19365141641631

2. Billing Address + Add

Billing
260 Terry Lane, New York,
New York, 10001, US

3. Shipping Address + Add

Shipping
1780 Angus Road, New
York, New York, 10022, US

4. Delivery Schedule

Express Delivery
90 min express delivery

Morning
8.00 AM - 11.00 AM

Noon
11.00 AM - 2.00 PM

Afternoon
Evening

Product	Subtotal
1 x Zara Army Bag 1 pc	\$260.00
Subtotal	\$260.00
Tax	\$5.20
Shipping	\$50.00
Do you have coupon?	
Total	\$315.20

Choose Payment Method

stripe PayPal Razorpay

Cash On Delivery

Please click Place order to make order and payment

Place Order

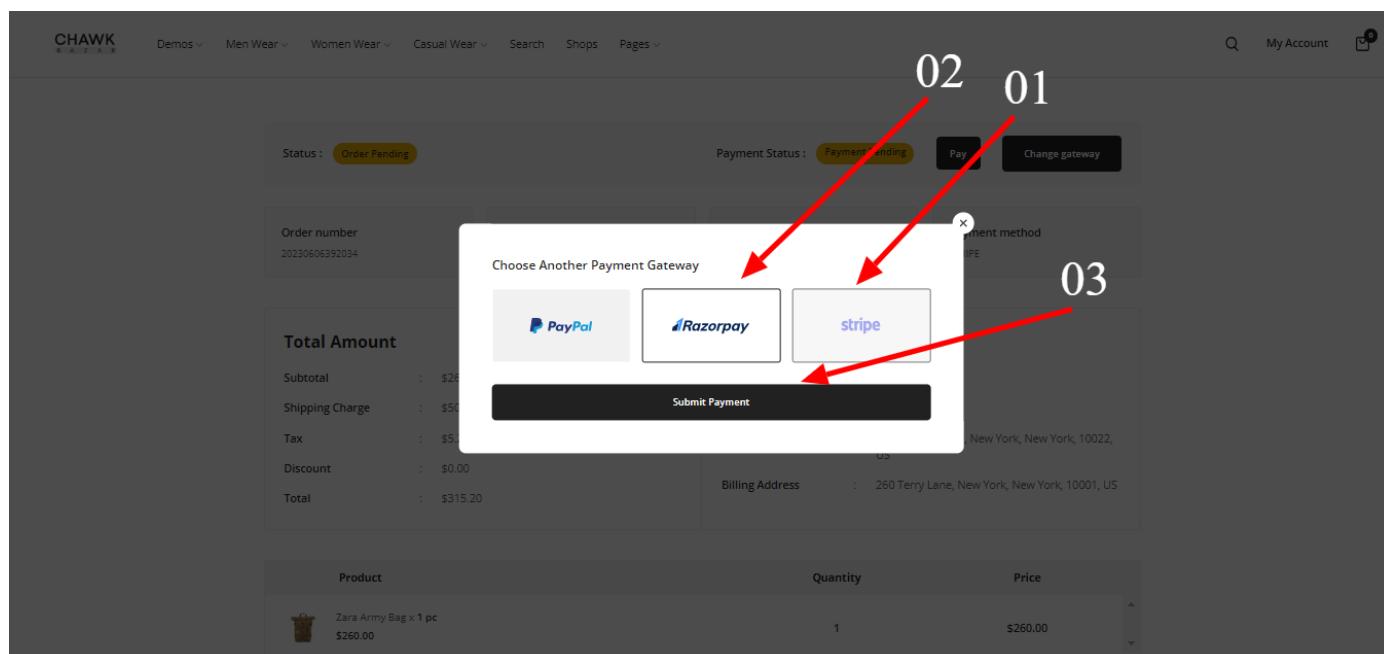
Select a payment gateway just before placing order.

** Notes for customer : ** Please choose this carefully. Otherwise you may need to re-select the payment procedure again.

2. 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 placeorder 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,



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.

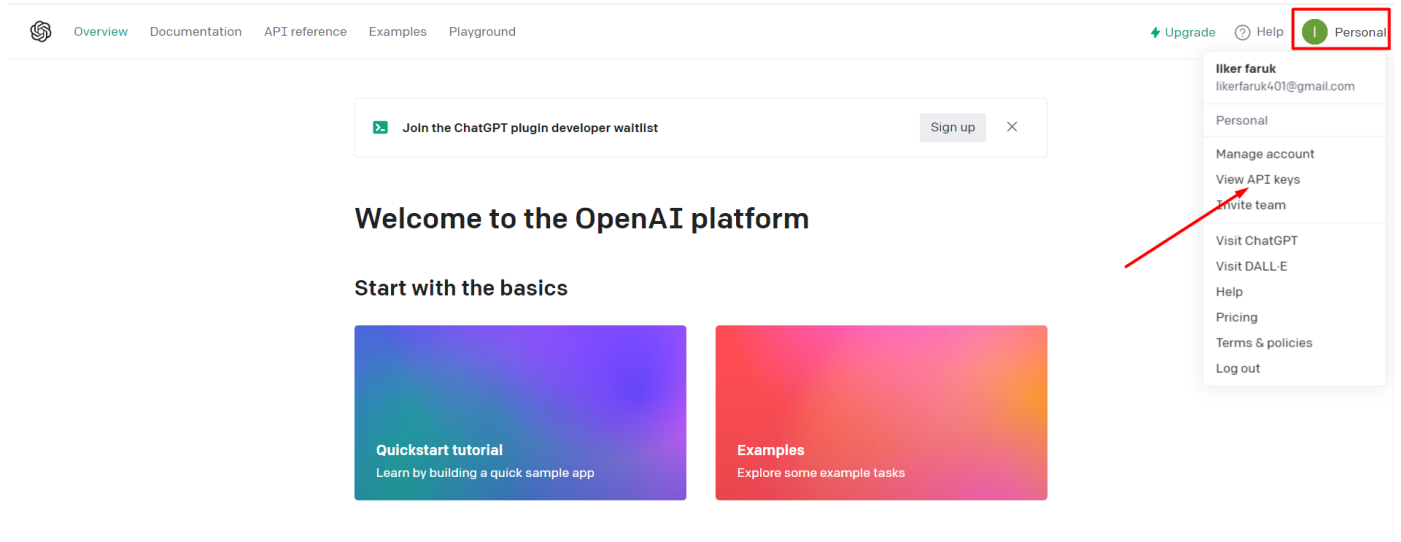
Third-party Integrations - OpenAI

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.

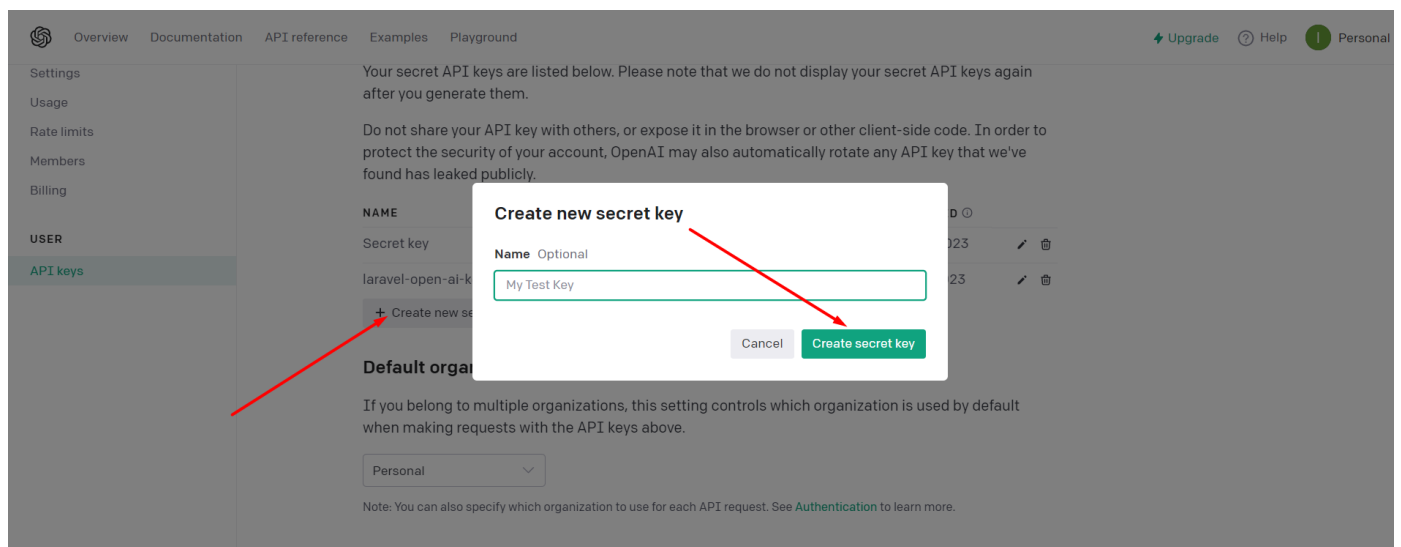
OpenAI Integration inside PickBazar.

Please follow & complete this steps for OpenAI integration for your e-commerce system.

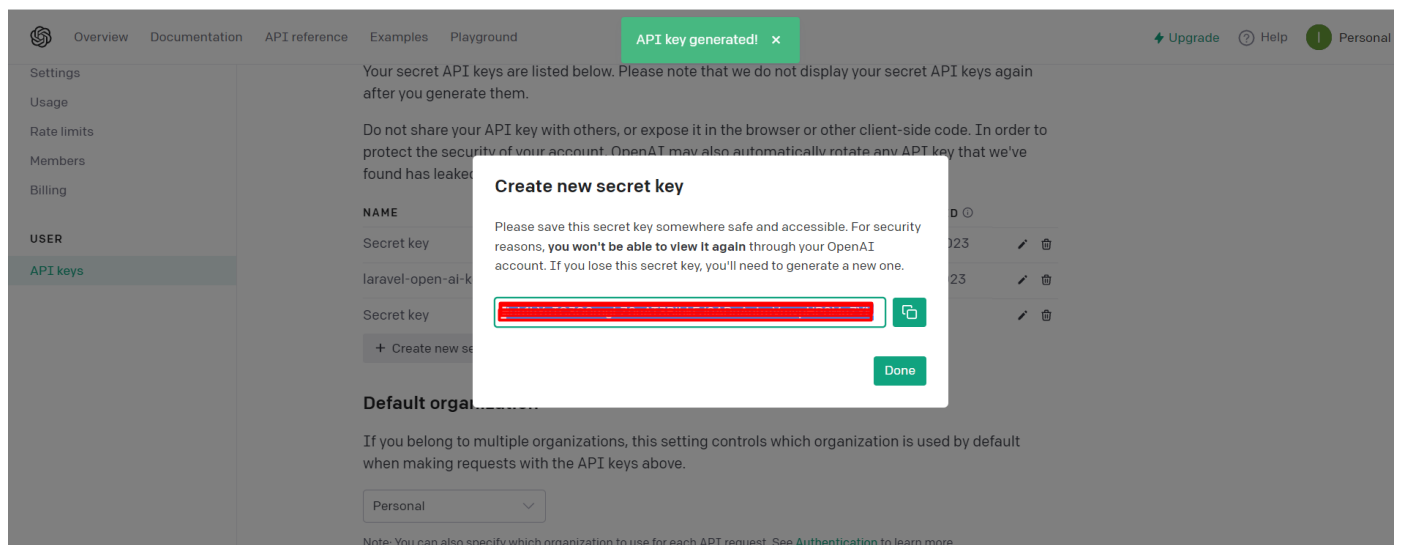
- Go to the [OpenAI Platform Website](#) & login If you aren't a registered user, Complete the OpenAI registration first.
- After logged in into OpenAI dashboard, Click on Personal, then click on View API Keys.



- 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=[REDACTED]
```

Admin Settings

- First go to settings from PickBazar admin dashboard. Inside settings you will find Enable OpenAI switch button. Enable the OpneAi.

PickBazar

Create Shop

Users

Coupons

Taxes

Shippings

Withdrawals

Message

Refunds

Questions

Store Notice

Reviews

Settings

Maximum Question Limit

4

Use OTP at checkout

Enable Must Verify Email

Enable OpenAI

Tax Class

Global

Shipping Class

Global

Enable Free Shipping

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

FAQ Page

An **FAQ page** is a web page that lists common questions and answers about a specific product, service, or topic. It is designed to provide quick and easy access to information that users frequently seek.

Admin End

Admin end empty FAQ lists.

CHAWK

Search your route...

Create Shop

Visit Site

Admin
Super Admin

MAIN

Dashboard

SHOP MANAGEMENT

Shops

My Shops

PRODUCT MANAGEMENT

Products

Inventory

Brands

Categories

Tags

Attributes

E-COMMERCE MANAGEMENT

Taxes

Shippings

FAQs

Type your query and press enter

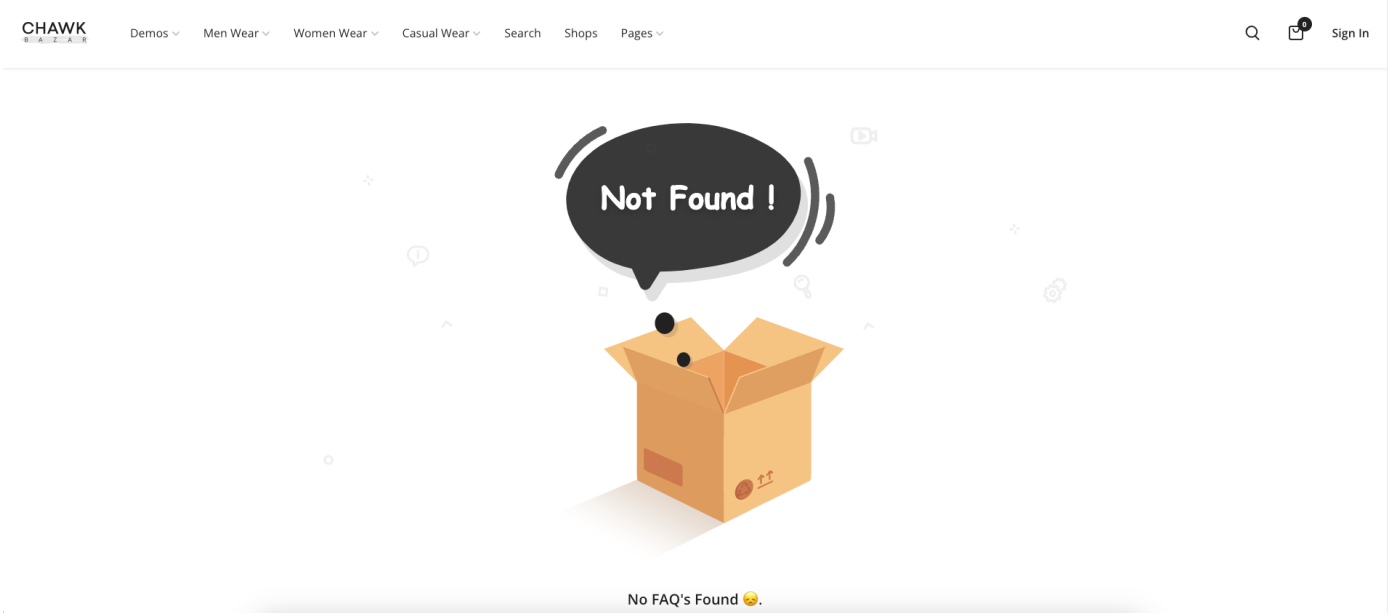
+ Add FAQ

ID	Title	Description	Type	Issued By	Actions
<div><div></div><div>No data found</div><div>Sorry we couldn't found any data</div></div>					

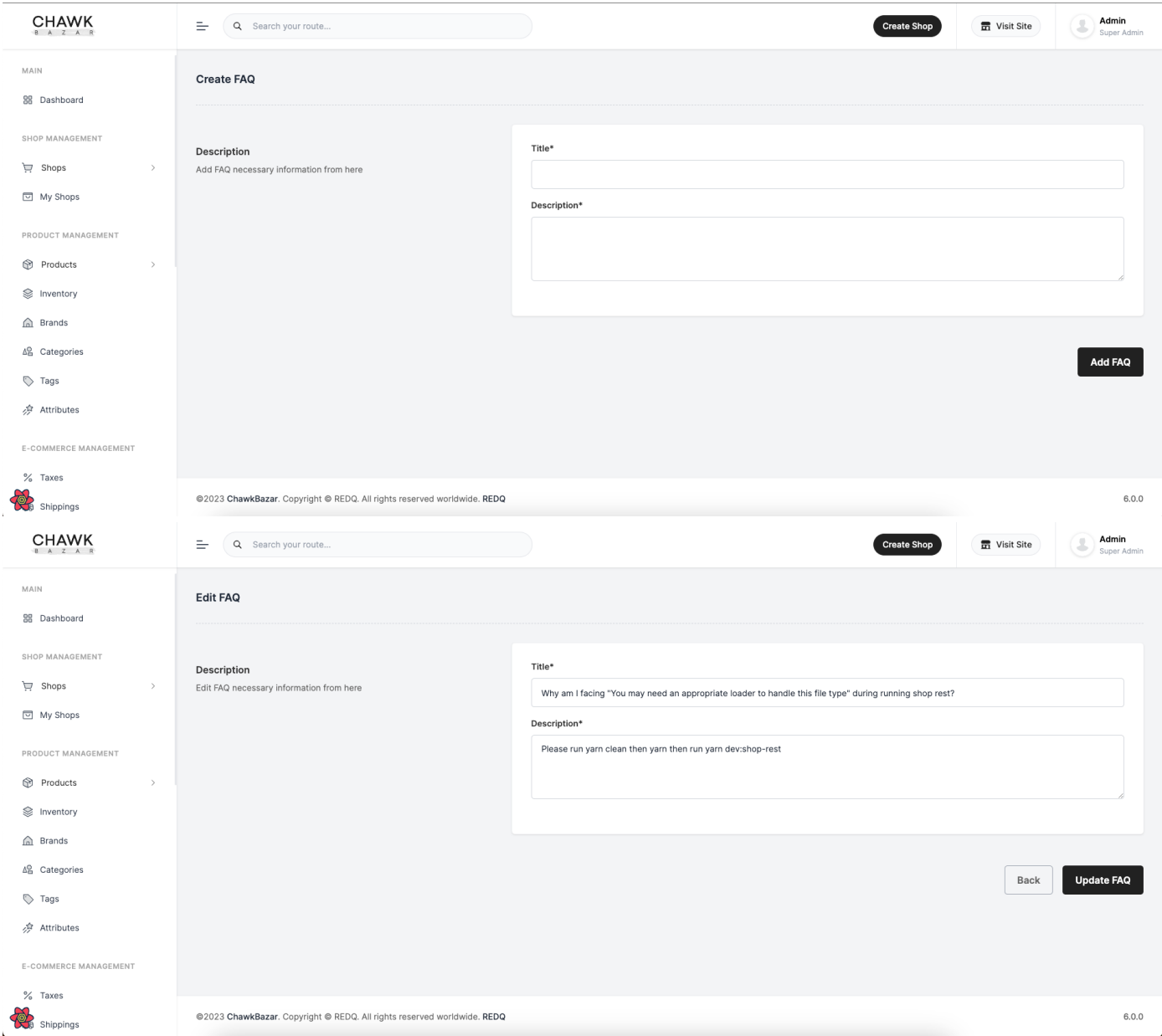
©2023 ChawkBazar. Copyright © REDQ. All rights reserved worldwide. REDQ

6.0.0

Shop end empty FAQ lists.



FAQ create or edit settings.



FAQ admin lists.

CHAWK

MAIN

Dashboard

SHOP MANAGEMENT

Shops

My Shops

PRODUCT MANAGEMENT

Products

Inventory

Brands

Categories

Tags

Attributes

E-COMMERCE MANAGEMENT

Taxes

Shipplings

Search your route...

Create Shop

Visit Site

Admin
Super Admin

FAQs

Type your query and press enter

Add FAQ

ID	Title	Description	Type	Issued By	Actions
#ID: 25	Changing .env files but not getting the cha...	Run Below command php artisan optimize:clear	global	Super Admin	✎ 🗑
#ID: 24	Why am I facing "You may need an appropri...	Please run yarn clean then yarn then run yarn dev:shop-rest	global	Super Admin	✎ 🗑

< 1 >

©2023 ChawkBazar. Copyright © REDQ. All rights reserved worldwide. REDQ

6.0.0

FAQ shop lists.

CHAWK

Demos Men Wear Women Wear Casual Wear Search Shops Pages

Sign In

explore

Frequently Asked Questions

Changing .env files but not getting the changes

Run Below command php artisan optimize:clear

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

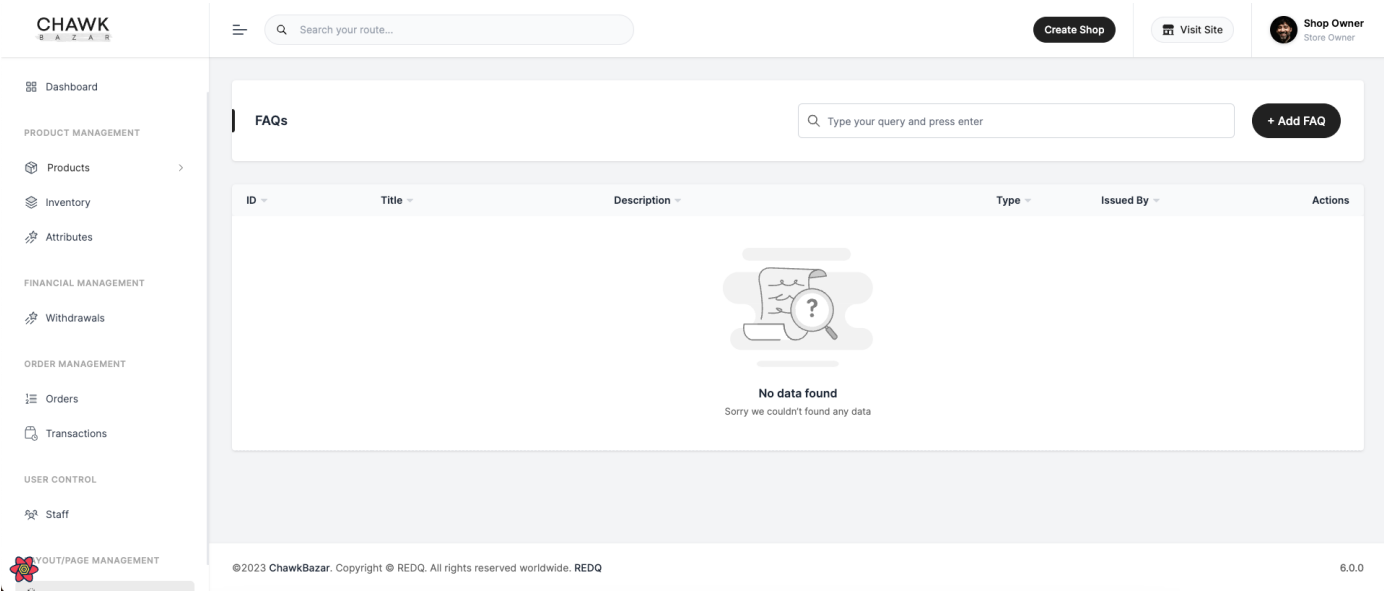
Shop End

Finally you find FAQ here [/help](#).

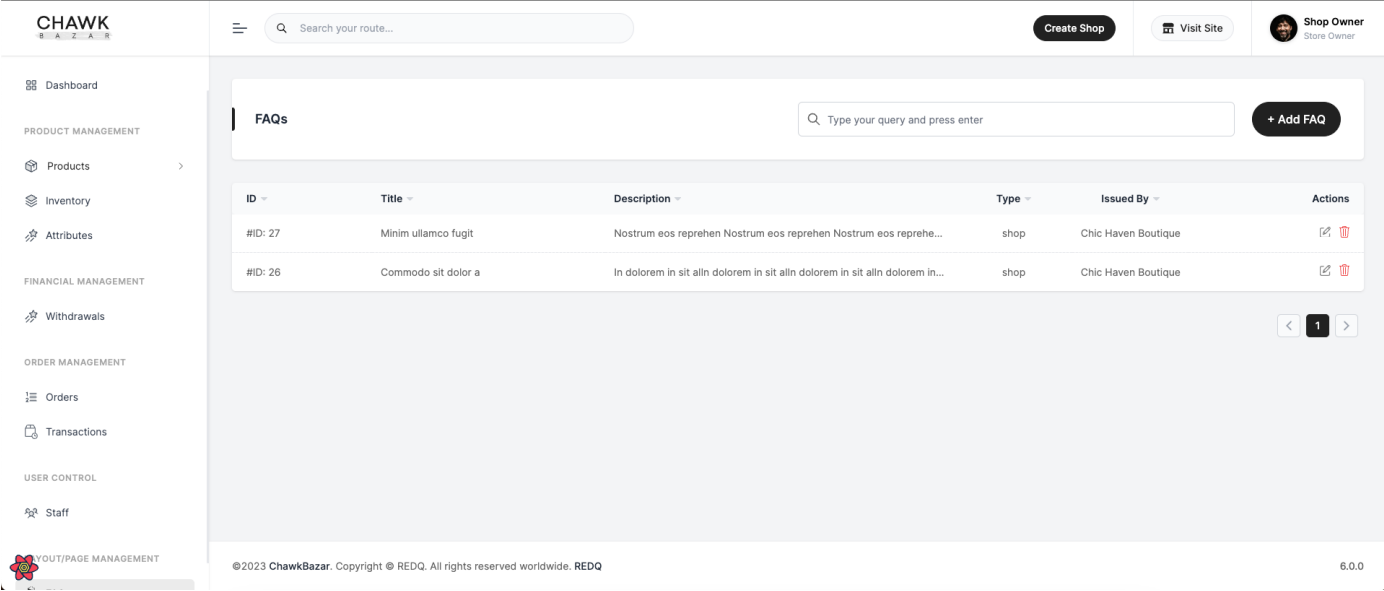
Vendor Shop End

You find FAQ here [/shops/shopName/faqs](#).

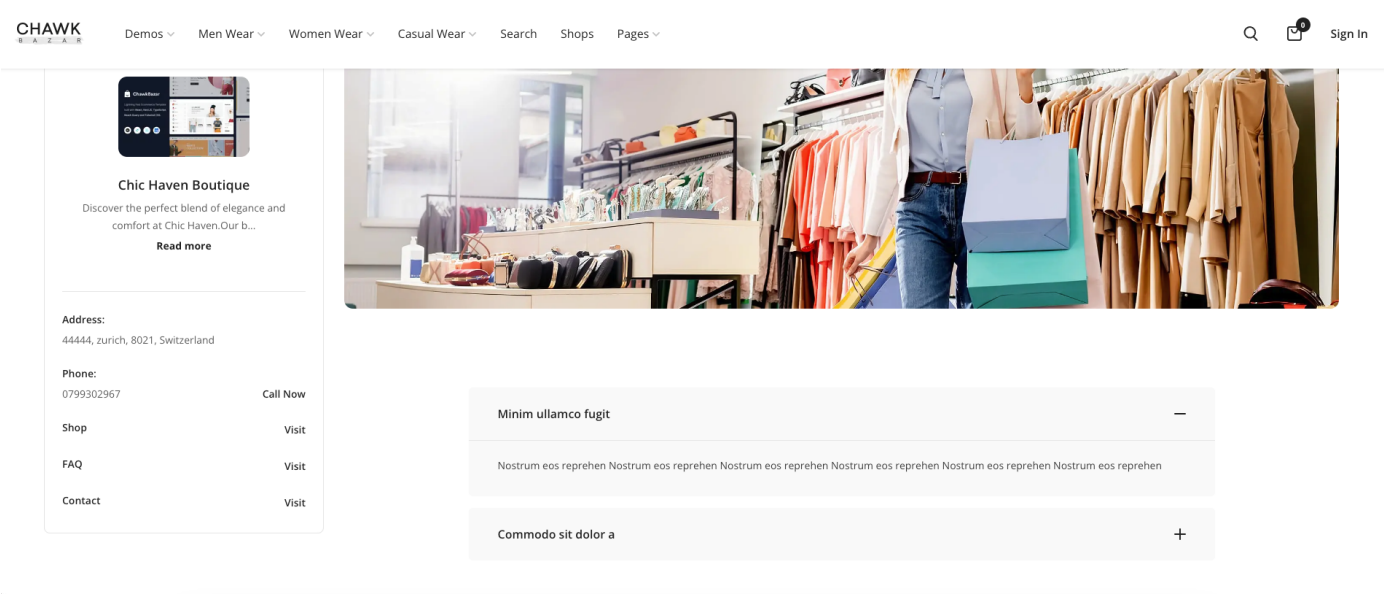
Vendor shop end empty FAQ lists.



Vendor shop end FAQ create or edit settings.



Vendor shop end FAQ lists.



Multiple payment gateway

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.

A **terms and conditions page** is a web page that outlines the rules and restrictions for the use of a website. It acts as a contract between the website owner and its users. The primary purpose of a terms and conditions page is to establish legal guidelines for the website and protect the interests of both parties involved.

If terms and conditions are not approved it's not appear on the shop page.

Admin end empty Terms and Condition lists.

Shop end empty Terms and Condition lists.

Create or edit terms and condition form.

CHAWK

Terms And Conditions

ORDER MANAGEMENT

Orders

Transactions

USER CONTROL

All users

Admin list

Vendors

Staff

Customers

PROMOTIONAL MANAGEMENT

Coupons

SITE MANAGEMENT

Settings

CHAWK

Search your route...

Create Shop

Visit Site

Admin
Super Admin

Create Terms And Conditions

Description

Add Terms and Condition necessary information from here

Title*

Description*

Add terms and conditions.

©2023 ChawkBazar. Copyright © REDQ. All rights reserved worldwide. REDQ6.0.0

CHAWK

Terms And Conditions

ORDER MANAGEMENT

Orders

Transactions

USER CONTROL

All users

Admin list

Vendors

Staff

Customers

PROMOTIONAL MANAGEMENT

Coupons

SITE MANAGEMENT

Settings

CHAWK

Search your route...

Create Shop

Visit Site

Admin
Super Admin

Edit Store Notice

Description

Edit Terms and Condition necessary information from here

Title*

User Accounts

Description*

Users are responsible for maintaining the confidentiality of their account information and password.

BackUpdate terms and conditions.

©2023 ChawkBazar. Copyright © REDQ. All rights reserved worldwide. REDQ6.0.0

Admin end terms and condition lists.

CHAWK

All Terms

Add new Terms

ORDER MANAGEMENT

Orders

Transactions

USER CONTROL

All users

Admin list

Vendors

Staff

Customers

PROMOTIONAL MANAGEMENT

Coupons

SITE MANAGEMENT

Settings

CHAWK

Search your route...

Create Shop

Visit Site

Admin
Super Admin

Terms And Conditions

Search

+ Add Terms And Conditions

ID	Name	Description	Type	Issued By	Status	Approval Action	Actions
#ID: 7	User Accounts	Users are responsible for ...	global	Super Admin	Approved	<div></div>	<div></div>
#ID: 6	Acceptance of Terms	Users must agree to the t...	global	Super Admin	Approved	<div></div>	<div></div>

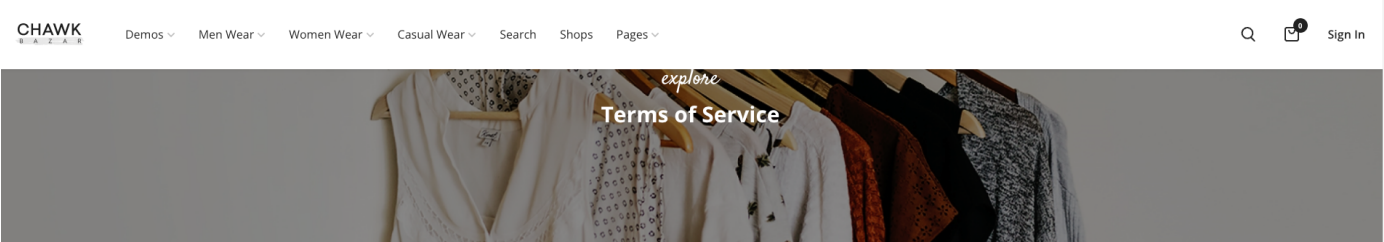
<

1

>

©2023 ChawkBazar. Copyright © REDQ. All rights reserved worldwide. REDQ6.0.0

Shop end terms and condition lists.



00 USER ACCOUNTS

01 ACCEPTANCE OF TERMS

User Accounts

Users are responsible for maintaining the confidentiality of their account information and password.

Acceptance of Terms

Users must agree to the terms and conditions before using the platform.

Not approved terms and condition item.

CHAWK BAZAR

All Terms

Add new Terms

ORDER MANAGEMENT

Orders

Transactions

USER CONTROL

All users

Admin list

Vendors

Staff

Customers

PROMOTIONAL MANAGEMENT

Coupons

SITE MANAGEMENT

Settings

Search your route...

Create Shop

Visit Site

Admin Super Admin

Terms And Conditions

Type your query and press enter

+ Add Terms And Conditions

ID	Name	Description	Type	Issued By	Status	Approval Action	Actions
#ID: 8	Product Information	The platform provides ac...	shop	Denim Delight Co.	Waiting for approval		
#ID: 7	User Accounts	Users are responsible for ...	global	Super Admin	Approved		
#ID: 6	Acceptance of Terms	Users must agree to the t...	global	Super Admin	Approved		

<

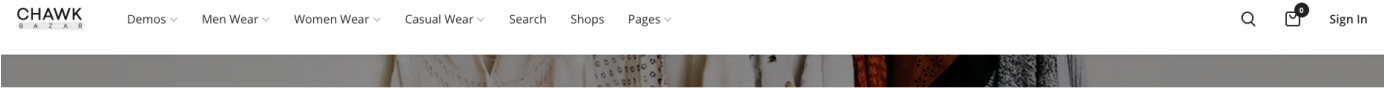
1

>

©2023 ChawkBazar. Copyright © REDQ. All rights reserved worldwide. REDQ

6.0.0

Non approved terms and condition item not showing in shop end.



00 USER ACCOUNTS

01 ACCEPTANCE OF TERMS

User Accounts

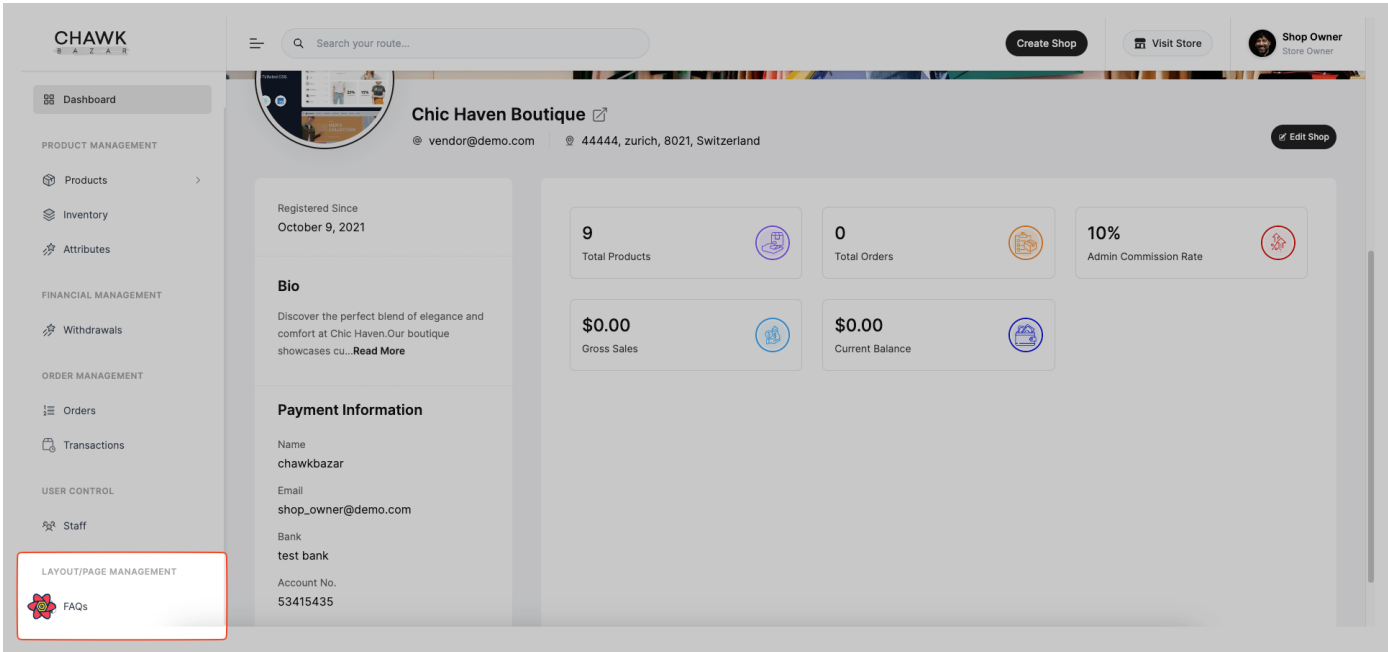
Users are responsible for maintaining the confidentiality of their account information and password.

Acceptance of Terms

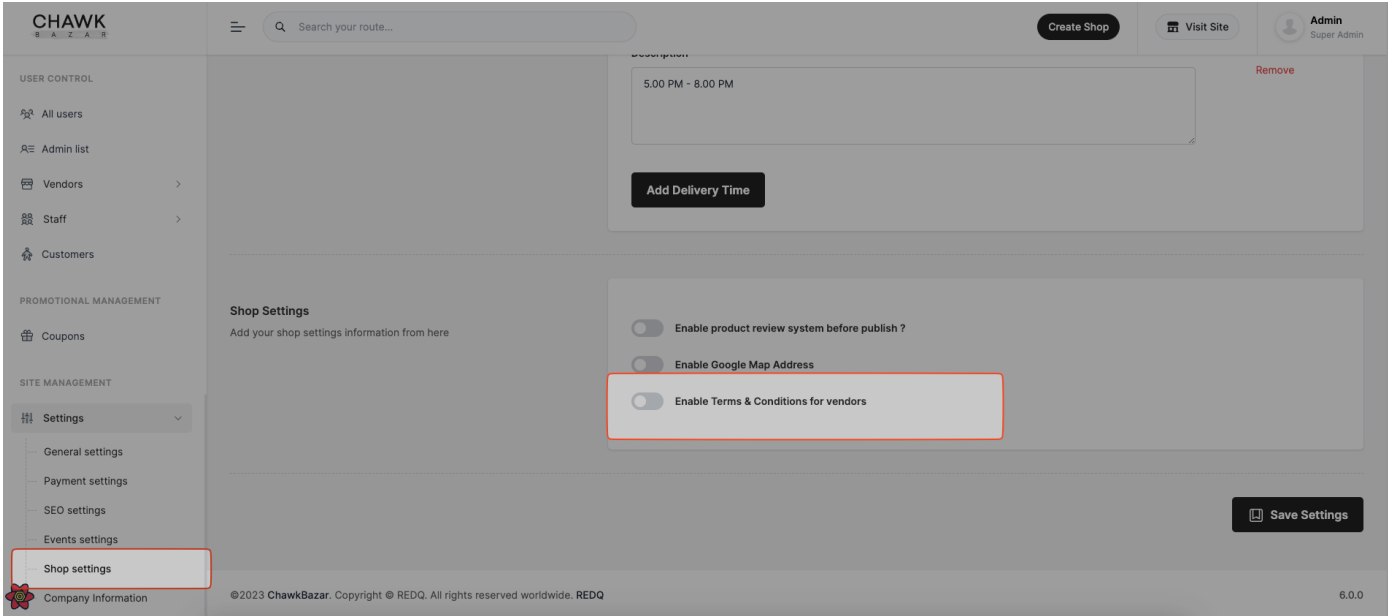
Users must agree to the terms and conditions before using the platform.

If admin active terms and condition for vendor it's available for shop and vendor.

Vendor shop end inactive terms and conditions.



You find vendor shop terms and conditions settings here </settings/shop>.



Vendor shop end active terms and conditions.

CHAWK

PRODUCT MANAGEMENT

Products

Inventory

Attributes

FINANCIAL MANAGEMENT

Withdrawals

ORDER MANAGEMENT

Orders

Transactions

USER CONTROL

Staff

LAYOUT/PAGE MANAGEMENT

FAQs

Terms And Conditions

Search your route...

Create Shop

Visit Site

Shop Owner

Listed Terms And Conditions

Type your query and press enter

+ Add Terms And Conditions

ID	Name	Description	Type	Issued By	Status	Approval Action	Actions
<div><div></div><div>No data found</div><div>Sorry we couldn't found any data</div></div>							

©2023 ChawkBazar. Copyright © REDQ. All rights reserved worldwide. REDQ6.0.0

Admin end vendor shop terms and condition lists.

CHAWK

MAIN

Dashboard

SHOP MANAGEMENT

Shops

My Shops

PRODUCT MANAGEMENT

Products

Inventory

Brands

Categories

Tags

Attributes

E-COMMERCE MANAGEMENT

Taxes

Shippings

Search your route...

Create Shop

Visit Site

Admin

Terms And Conditions

Type your query and press enter

+ Add Terms And Conditions

ID	Name	Description	Type	Issued By	Status	Approval Action	Actions
#ID: 10	Nemo anim et invento	Excepturi quia repel	shop	Denim Delight Co.	Approved	<div><div></div><div></div></div>	<div><div></div><div></div></div>
#ID: 9	Laboris minim et aspern	Proident nostrud et	global	Super Admin	Waiting for approval	<div><div></div><div></div></div>	<div><div></div><div></div></div>
#ID: 8	Product Information	The platform provides ac...	shop	Denim Delight Co.	Waiting for approval	<div><div></div><div></div></div>	<div><div></div><div></div></div>
#ID: 7	User Accounts	Users are responsible for ...	global	Super Admin	Approved	<div><div></div><div></div></div>	<div><div></div><div></div></div>
#ID: 6	Acceptance of Terms	Users must agree to the L...	global	Super Admin	Approved	<div><div></div><div></div></div>	<div><div></div><div></div></div>

1

©2023 ChawkBazar. Copyright © REDQ. All rights reserved worldwide. REDQ6.0.0

Vendor shop end terms and conditions list.

CHAWK

PRODUCT MANAGEMENT

Products

Inventory

Attributes

FINANCIAL MANAGEMENT

Withdrawals

ORDER MANAGEMENT

Orders

Transactions

USER CONTROL

Staff

LAYOUT/PAGE MANAGEMENT

FAQs

Terms And Conditions

Search your route...

Create Shop

Visit Site

Shop Owner

Listed Terms And Conditions

Type your query and press enter

+ Add Terms And Conditions

ID	Name	Description	Type	Issued By	Status	Approval Action	Actions
#ID: 10	Nemo anim et invento	Excepturi quia repel	shop	Denim Delight Co.	Approved	<div><div></div></div>	<div><div></div><div></div></div>
#ID: 8	Product Information	The platform provides ac...	shop	Denim Delight Co.	Waiting for approval	<div><div></div></div>	<div><div></div><div></div></div>

1

©2023 ChawkBazar. Copyright © REDQ. All rights reserved worldwide. REDQ6.0.0

Vendor shop end vendor terms and conditions list. You find vendor end terms and conditions here </shops/shopName/terms>.

CHAWKBAZAR

Demos

Men Wear

Women Wear

Casual Wear

Search

Shops

Pages

Q

9

Sign In

T

Denim Delight Co.

Experience the joy of denim at Denim Delight Co.

Our curated collectio...

Read more

Address:

44444, California, zurich, 8021, Switzerland

Phone:

7033542601

Website:

demodemo.com

Shop

Call Now

Visit This Site

Visit

SALE UP TO 20% OFF

Meet Your Next Favorite Books.

Shop Now

00 NEMO ANIM ET INVENTO

Nemo anim et invento

Excepturi quia repel

Part of the Party

The Messy Man

The Cautious Caterpillar

Charlie the Firefighter

Back to Earth with a Bump

The Zoo Vet

Translation

If you're using `chawkbazar-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`.

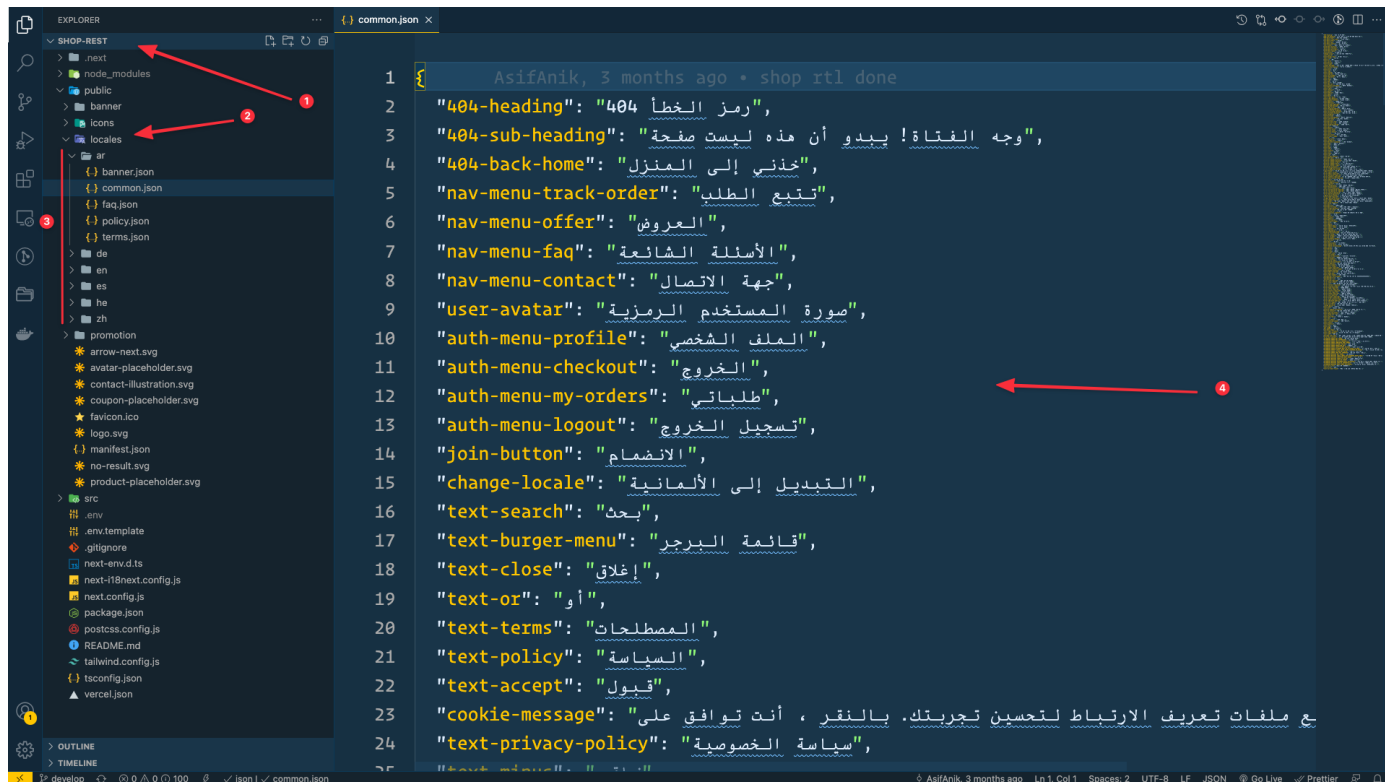
If your translation language is from them then just edit the translation string from,

For Rest

`chawkbazar-laravel -> shop -> public -> locales -> YOUR LANGUAGE`

`chawkbazar-laravel -> admin -> rest -> public -> locales -> YOUR LANGUAGE`

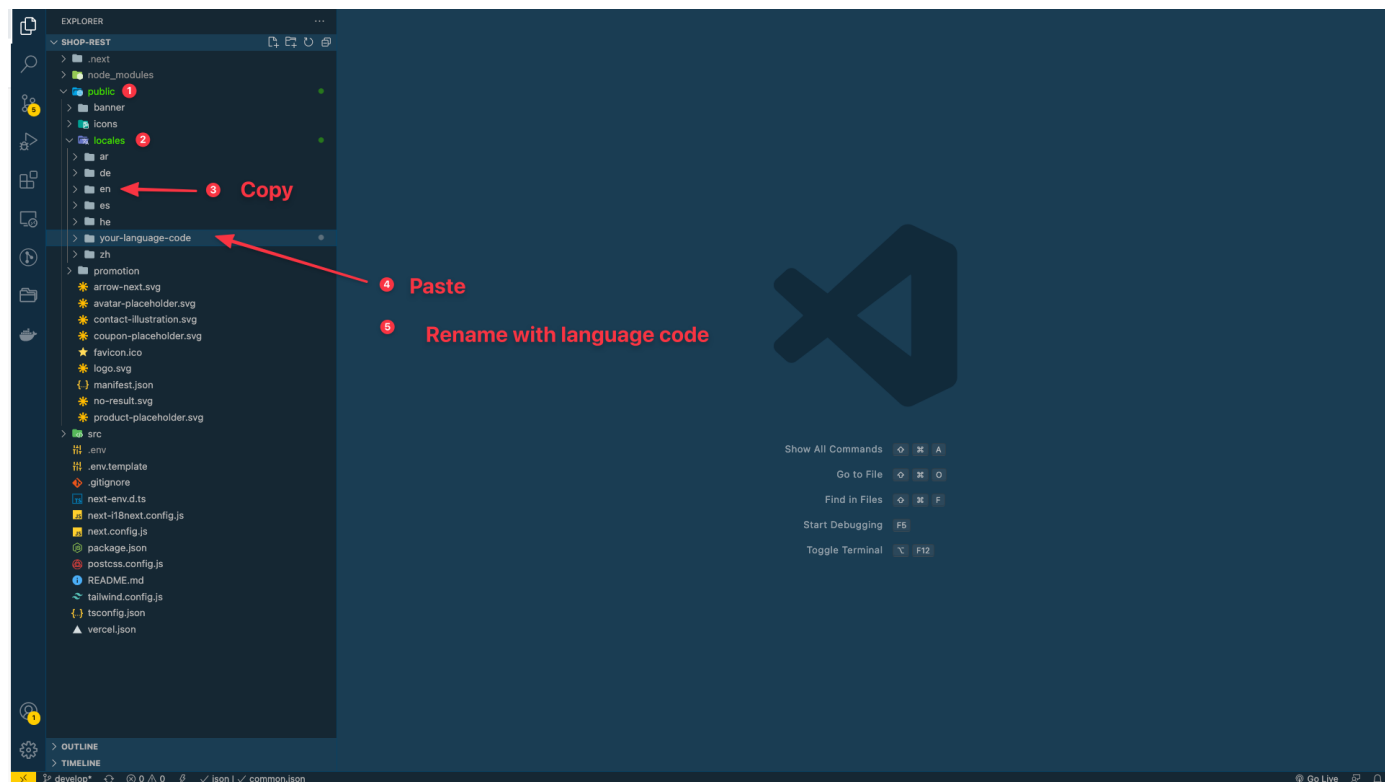
123 / 169



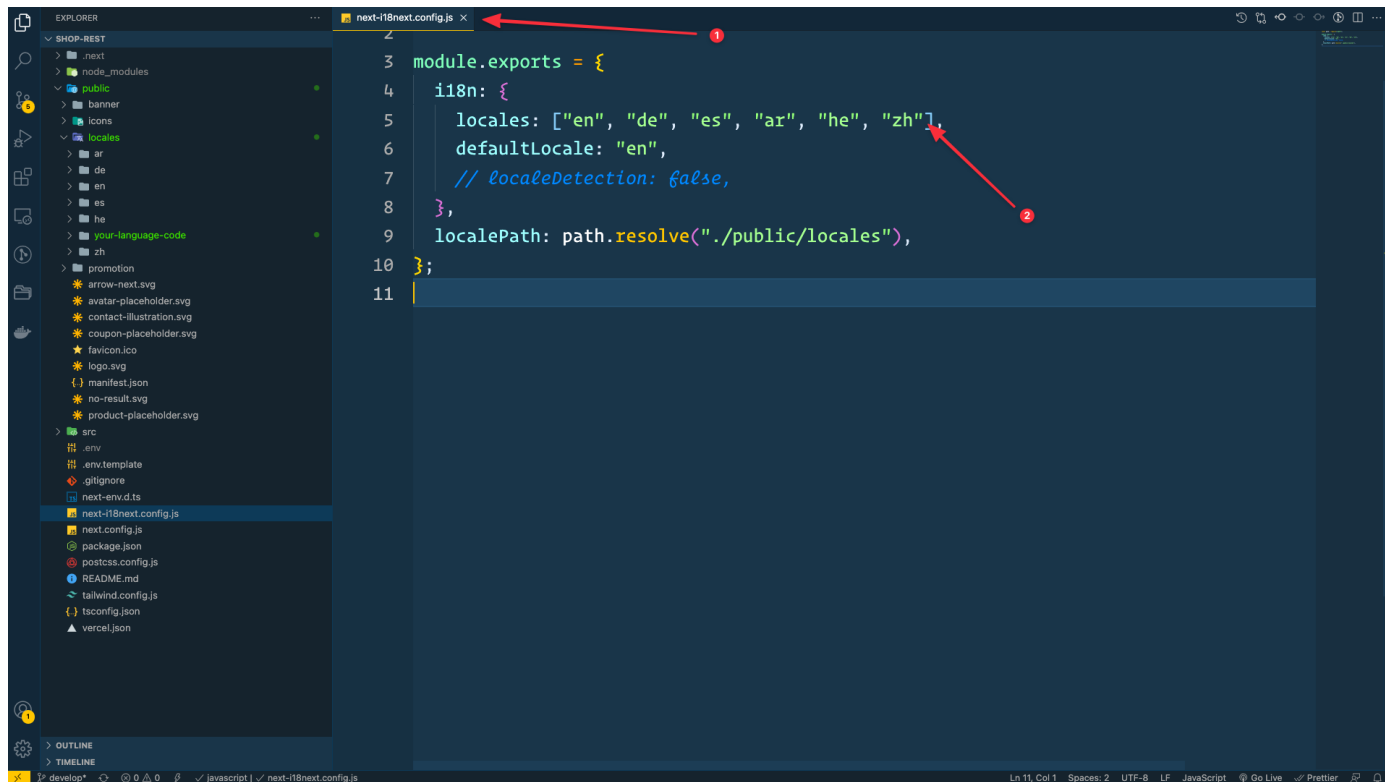
After translating **string** from **JSON** re-run **rebuild** project it'll translate the static string.

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,

```
chawkbazar-laravel -> shop -> public -> locales -> YOUR LANGUAGE -> next-i18next.config.js
```

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

Multivendor

Create New Shop

To create new shop login as an **administrator** or create a new account for creating **shop**

CHAWK BAZAR

Login to admin

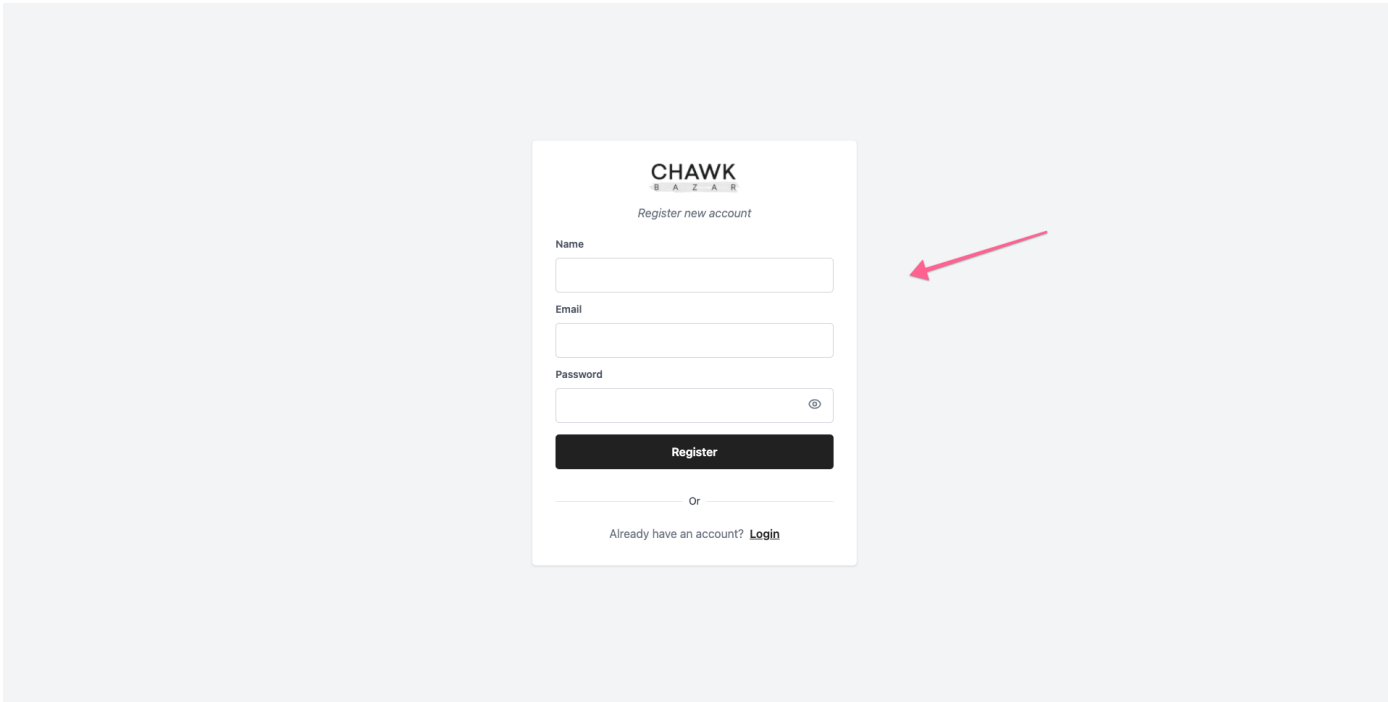
Email

Password [Forgot password?](#)

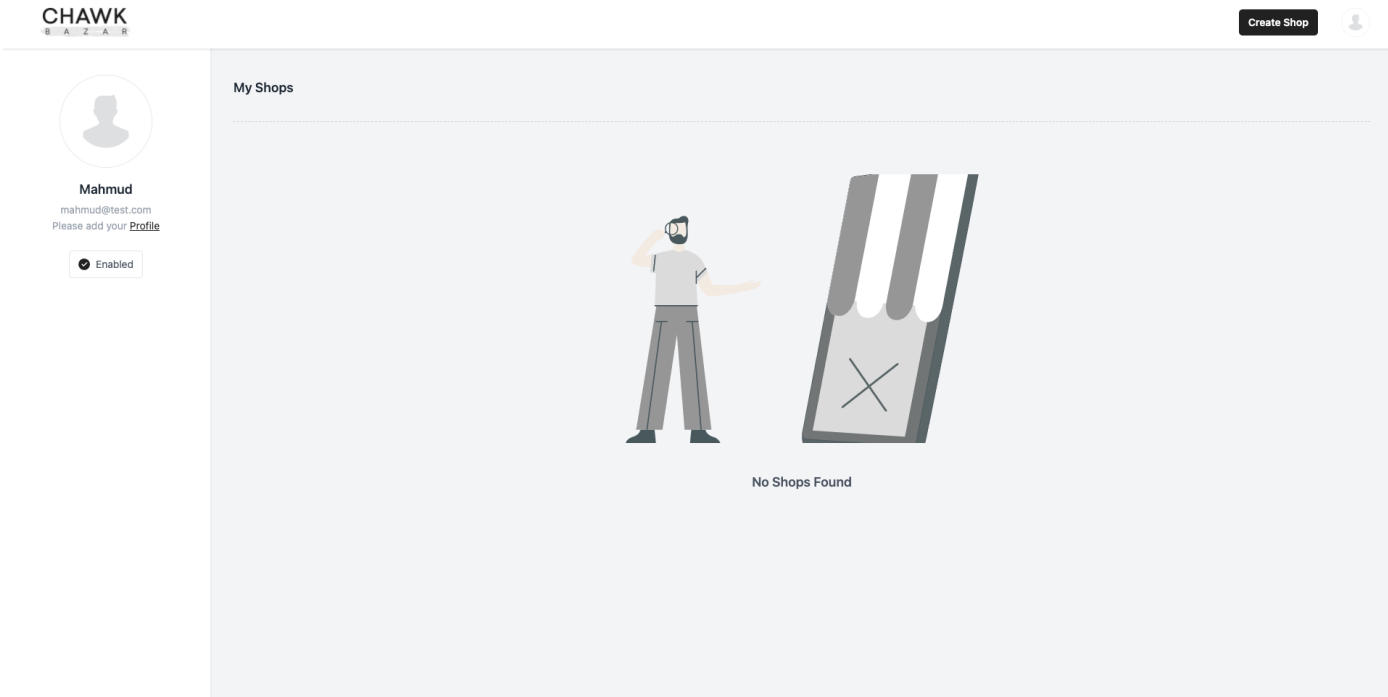
Login

Or

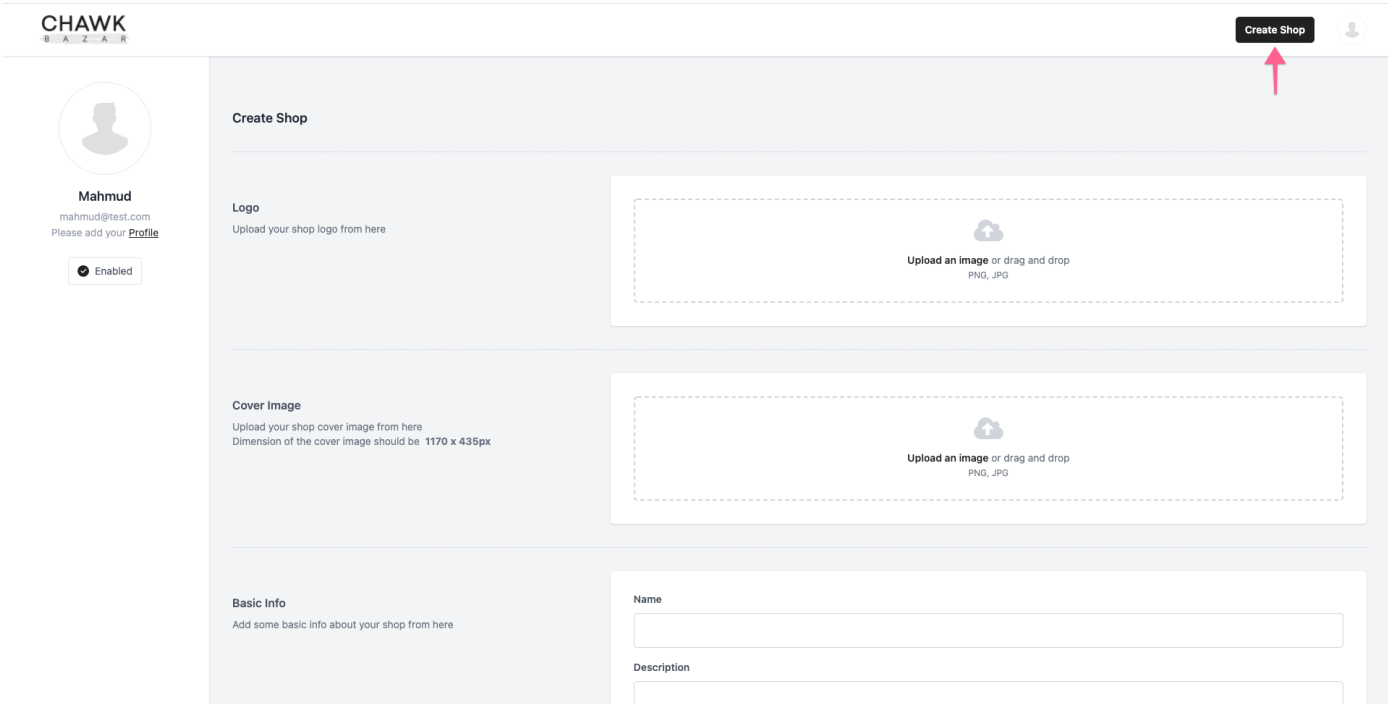
Don't have any account? [Register as Shop Owner](#)



After creating the account you'll be redirected to this page,

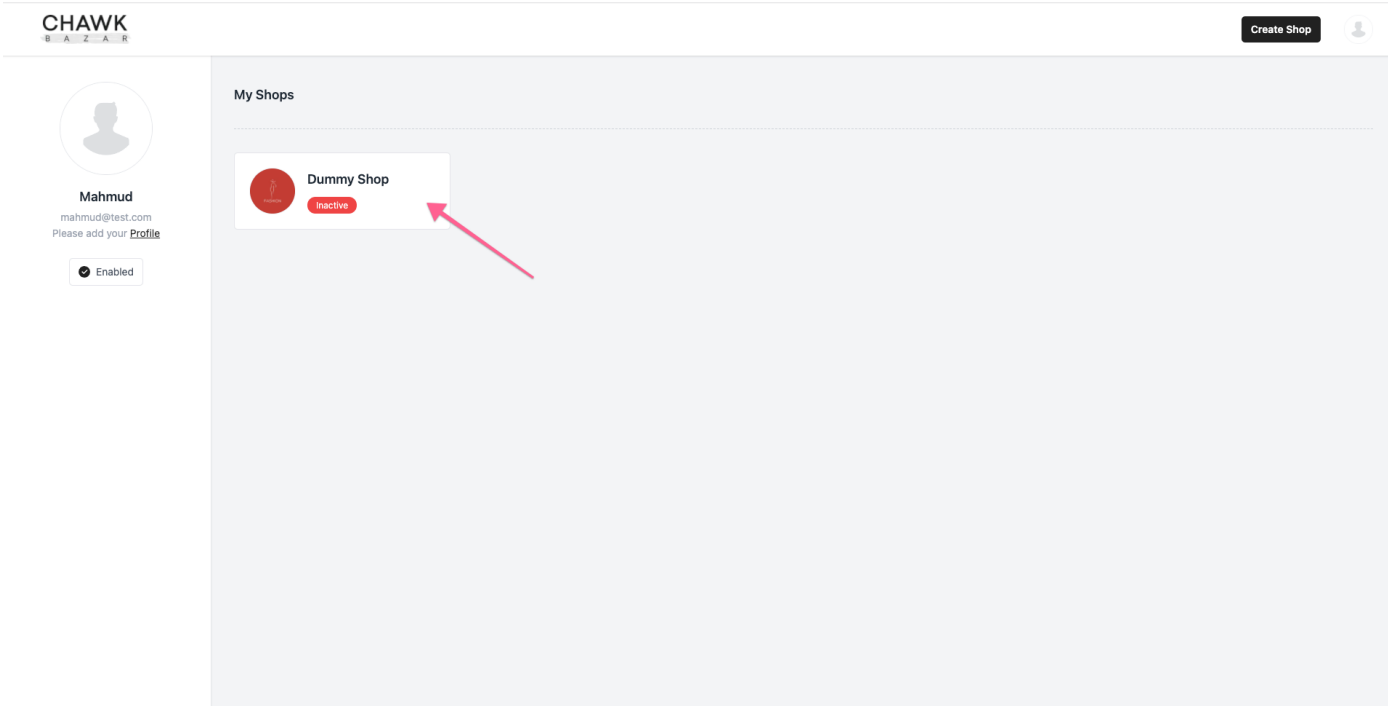


After that, click [Create Shop](#)



And provide all the information for the **store**.

After creating the shop you'll redirect to this page,



By default, the shop will be inactive. Only **administrator** can active a shop.

From **administrator** account go to shop and click **mark to activate** or **deactivate** a shop.

CHAWK

B A Z A R

Create Shop

Dashboard

Shops

My Shops

Products

Attributes

Brands

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
	Dummy Shop	Mahmud	0	0	Inactive	<div><div></div><div></div></div>
	name	Admin	0	0	Inactive	<div><div></div><div></div></div>
	Chawkbazar Vendor shop	Vendor	57	5	Active	<div><div></div><div></div></div>
	Chawkbazar Shop	Admin	0	0	Active	<div><div></div><div></div></div>

<

1


>

After activate the shop by administrator the vendor dashboard will be like this,

CHAWK

B A Z A R

Create Shop




Mahmud

mahmud@test.com

Please add your Profile

Enabled

My Shops



Dummy Shop

Active

After click on shop, you'll be redirected to dashboard page,

CHAWK
BAZAR

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

FASHION

Dummy Shop

This is a vendor shop of chawkbazar

4360 Hampton Meadows, Boston,
Massachusetts, 02210, USA

01236547852

Visit Shop

57

Total Products

5

Total Orders

\$7,906.50

Gross Sales

\$7,828.50

Current Balance

10 %

Admin Commission Rate

Registered Since
October 9, 2021

Payment Information

Name:
vendor

Email:
vendor@demo.com

Bank:
vendor bank

Account No.:
6523498651

From this dashboard, you can maintain you shop,

Dashboard:

CHAWK
BAZAR

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

FASHION

Dummy Shop

This is a vendor shop of chawkbazar

4360 Hampton Meadows, Boston,
Massachusetts, 02210, USA

01236547852

Visit Shop

57

Total Products

5

Total Orders

\$7,906.50

Gross Sales

\$7,828.50

Current Balance

10 %

Admin Commission Rate

Registered Since
October 9, 2021

Payment Information

Name:
vendor

Email:
vendor@demo.com

Bank:
vendor bank

Account No.:
6523498651

Attributes:

CHAWK

B A Z A R

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Attributes

+ Add Attribute

ID	Name	Values	Actions
No data found			

Products:

CHAWK

B A Z A R

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Products

+ Add Product

Filter

Image	Name	Brand	Product Type	Price/Unit	Quantity	Status	Actions
No data found							

Order:

CHAWK

B A Z A R

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	Download	Actions
No data found							

Staff:

CHAWK

B A Z A R

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

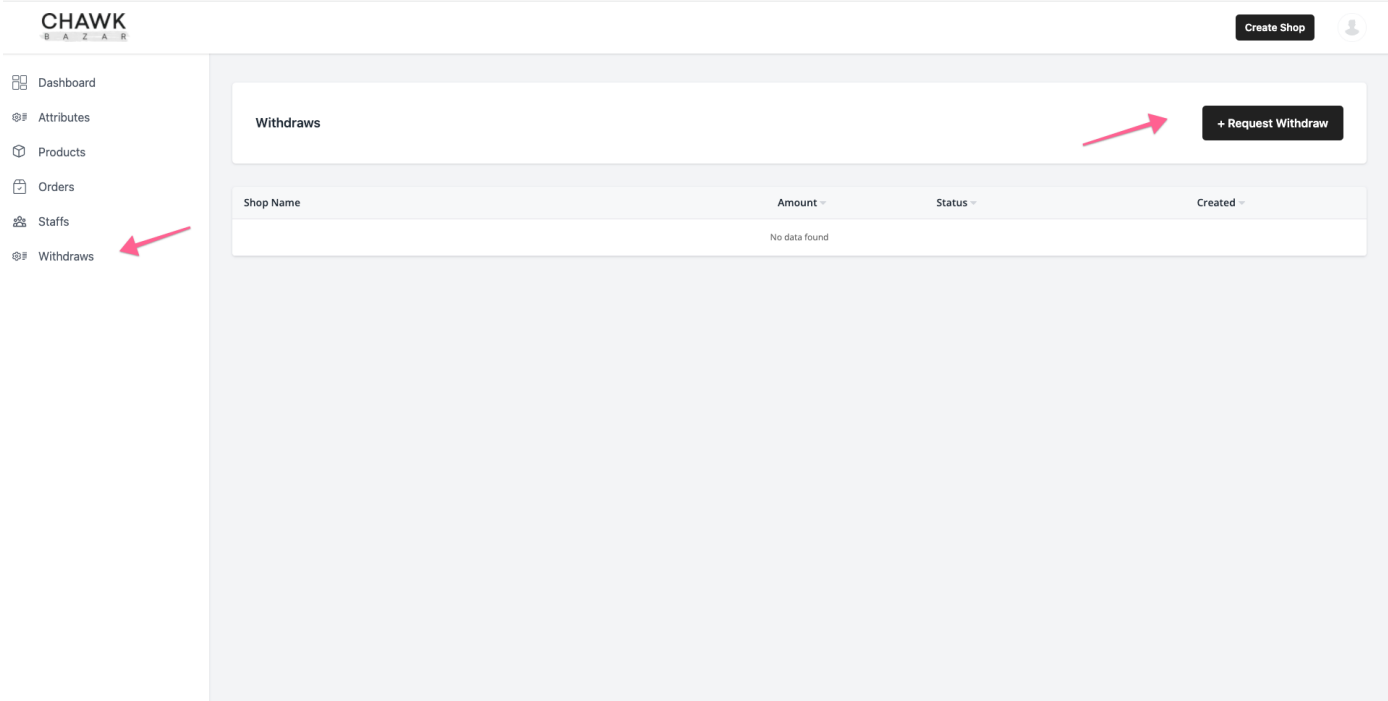
Withdraws

Staff

+ Add Staff

Name	Email	Status	Actions
No data found			

WithDraw:



User Roles:

Super Admin:

Super admin can do everything. the admin can maintain and edit every store on the site.

Store Owner:

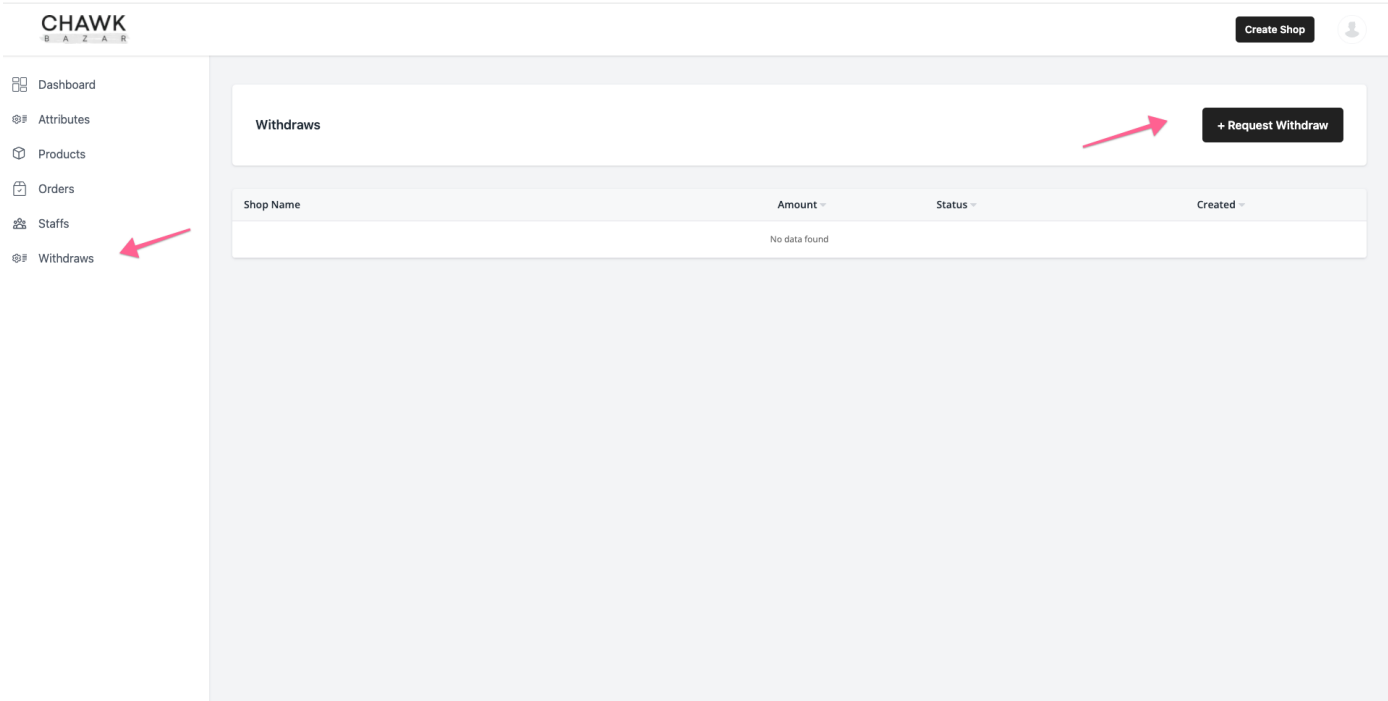
Store owner can edit or main it's store, staff or payment.

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,



CHAWK

B A Z A R

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,

CHAWK

B A Z A R

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Withdraws

+ Request Withdraw

Shop Name	Amount	Status	Created
Chawkbazar Vendor shop	\$500.00	Pending	a few seconds ago
Chawkbazar Vendor shop	\$78.00	Approved	5 hours ago

<

1

>

After request, the admin has to be approved the admin.

CHAWK

Dashboard

Shops

My Shops

Products

Attributes

Brands

Categories

Tags

Orders

Order Status

Users

Coupons

Taxes

Shippings

Withdraws

Settings

Create Shop

Withdraws

Shop Name	Amount	Status	Created	Actions
Chawkbazar Vendor shop	\$500.00	Pending	a few seconds ago	
Chawkbazar Vendor shop	\$78.00	Approved	5 hours ago	

<

1

>

CHAWK

Dashboard

Shops

My Shops

Products

Attributes

Brands

Categories

Tags

Orders

Order Status

Users

Coupons

Taxes

Shippings

Withdraws

Settings

Create Shop

Withdrawal Information

Amount : 500

Payment Method : N/A

Status : Pending

Pending

Approved

On Hold

Processing

Pending

Rejected

Change Status

After approved,

CHAWK

BAZAR

Create Shop

Dashboard

Shops

My Shops

Products

Attributes

Brands

Categories

Tags

Orders

Order Status

Users

Coupons

Taxes

Shippings

Withdraws

Settings

Withdraws

Shop Name	Amount	Status	Created	Actions
Chawkbazar Vendor shop	\$500.00	Approved	a minute ago	
Chawkbazar Vendor shop	\$78.00	Approved	5 hours ago	

<

1

>

FrontEnd Shop

From frontend when customers click on the **shop** page they'll get all the shops as a card list,

CHAWK

BAZAR

Demos

Men Wear

Women Wear

Casual Wear

Search

Pages

Account

Super Shops Near You

Dummy Shop

NEW

Chawkbazar Vendor shop

NEW

Chawkbazar Shop

NEW

Get Expert Tips In Your Inbox

Subscribe to our newsletter and stay updated.

Write your email here

Subscribe

Social

Facebook

Twitter

Instagram

Contact

Contact Us

Email: admin@chawkbazar.demo

Website: https://redq.io

About

Support Center

Customer Support

About Us

Copyright

Customer Care

FAQ & Helps

Shipping & Delivery

Return & Exchanges

Our Information

Privacy policy update

Terms & conditions

Return Policy

Site Map

Community

Announcements

Answer center

Discussion boards

Giving works

Also when customers click on a product they will get the seller link,

135 / 169

CHAWK
B A Z A R

Demos

Men Wear

Women Wear

Casual Wear


Search

Pages

Q

Account

Home / Products / Dido Pilot Glass



Dido Pilot Glass

Polarized sunglasses reduce glare reflected off of roads, bodies of water, snow and other horizontal surfaces. Restore true color. Vision lenses are 400UV rated, meaning it can block UVA and UVB radiation.

\$300.00 ~~\$350.00~~

-

1

+

Add To Cart

SKU: kjjkjk894561230

Category: Sunglass, Sports

Tags: Kids Collection, Flash Sale, Sunglass, Sports, Men's Collection

Brand: Roseban

Shop: Chawkbazar Vendor shop

Related Products

After clicking the **seller link** they will redirect to the seller shop page,

CHAWK
B A Z A R

Demos

Men Wear

Women Wear


Casual Wear

Search

Pages

Q

Account



Chawkbazar Vendor shop

This is a vendor shop of chawkbazar


Address:






4360 Hampton Meadows, Massachusetts, Boston, 02210, USA

Phone:

01236547852

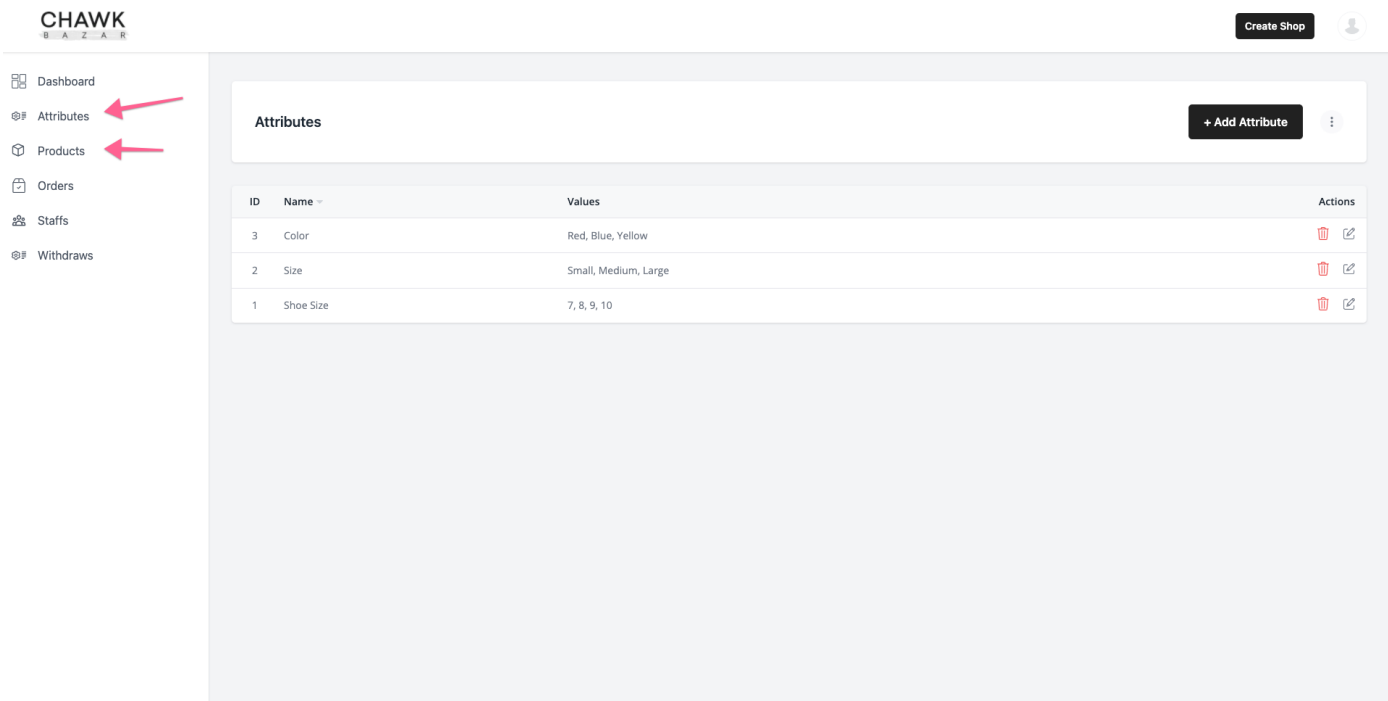
Call Now





Export Import

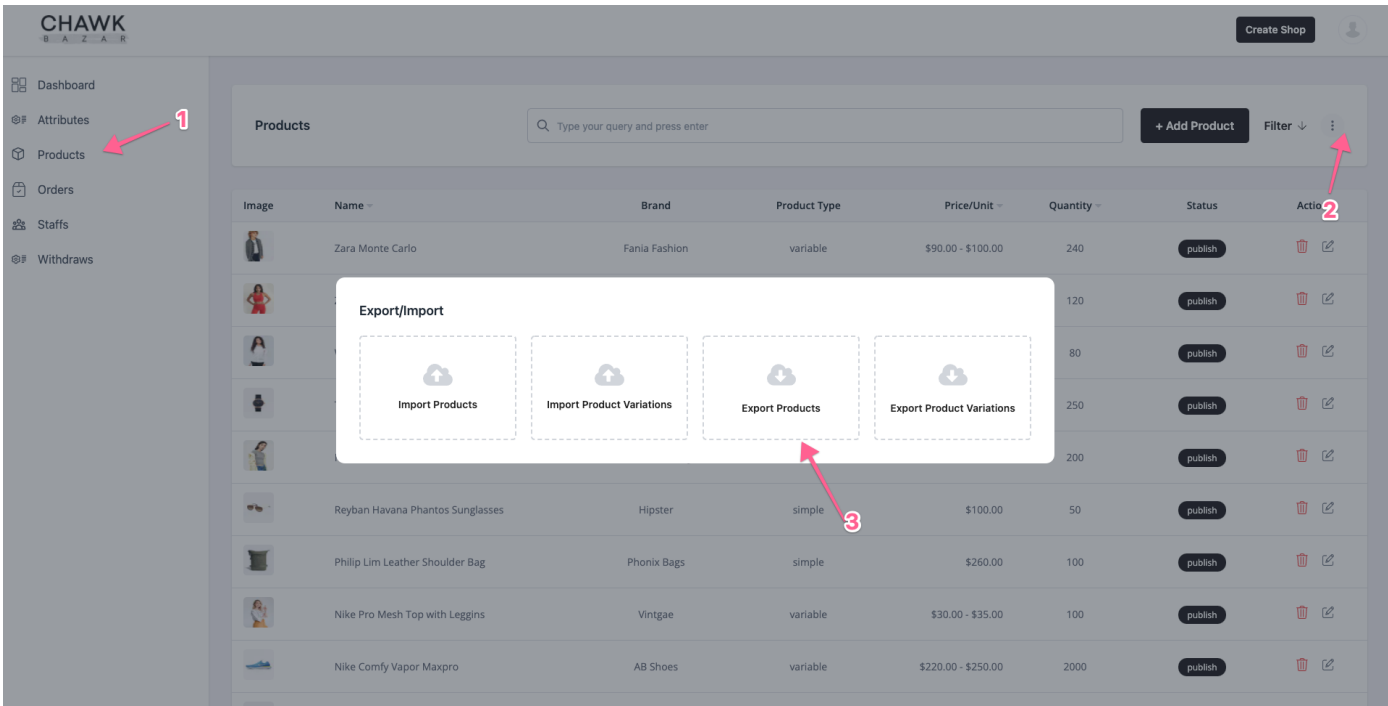
Export Import works under a shop. So, You have to go to the **shop page** -> **products menu** for product or **shop page** -> **attributes menu** to import or export attributes.



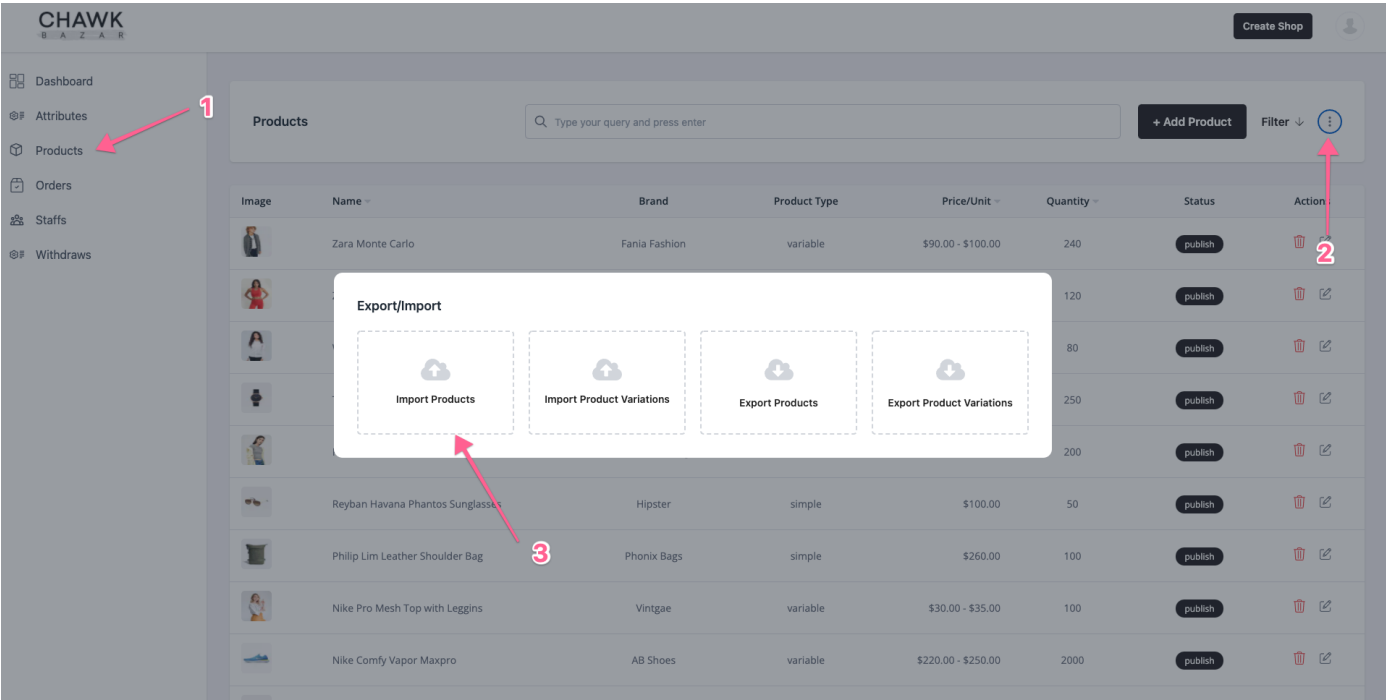
Export Import work different way for simple and variable product

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.



Similarly, for import **csv**, go to the same option and import exported **csv**

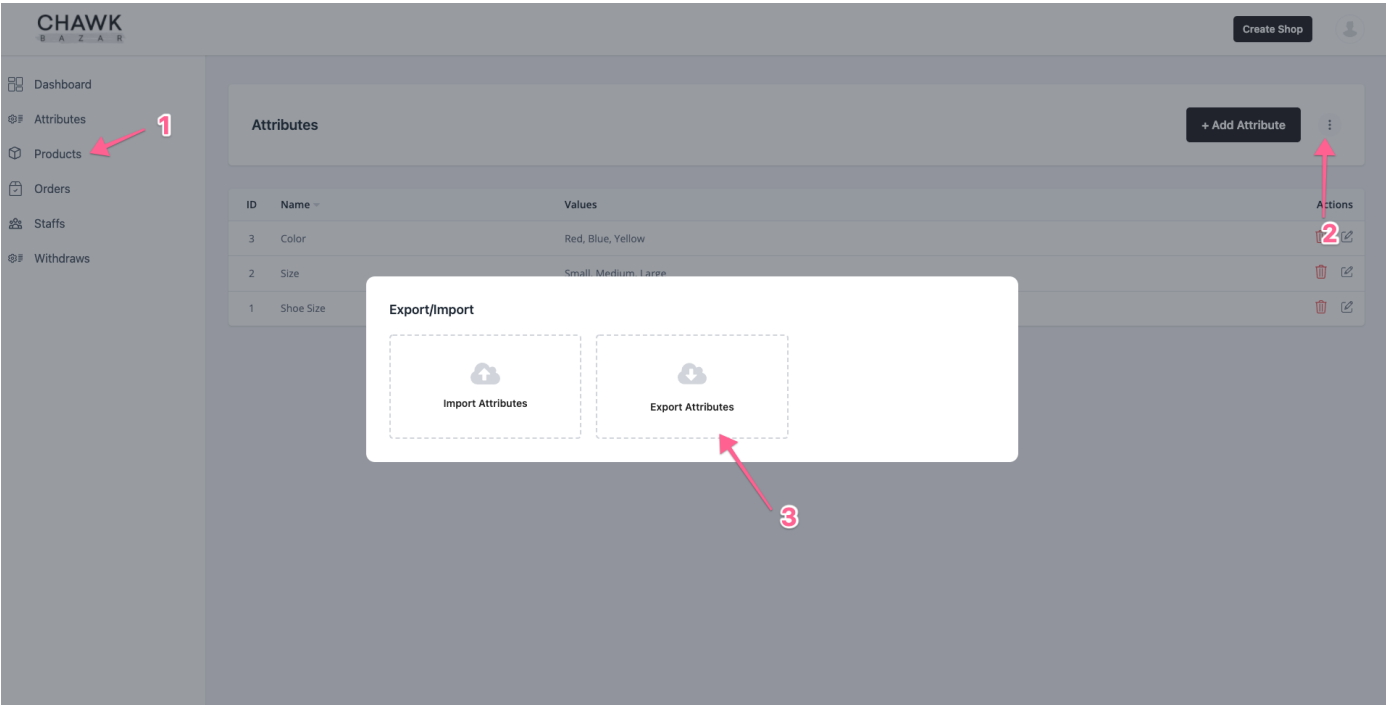


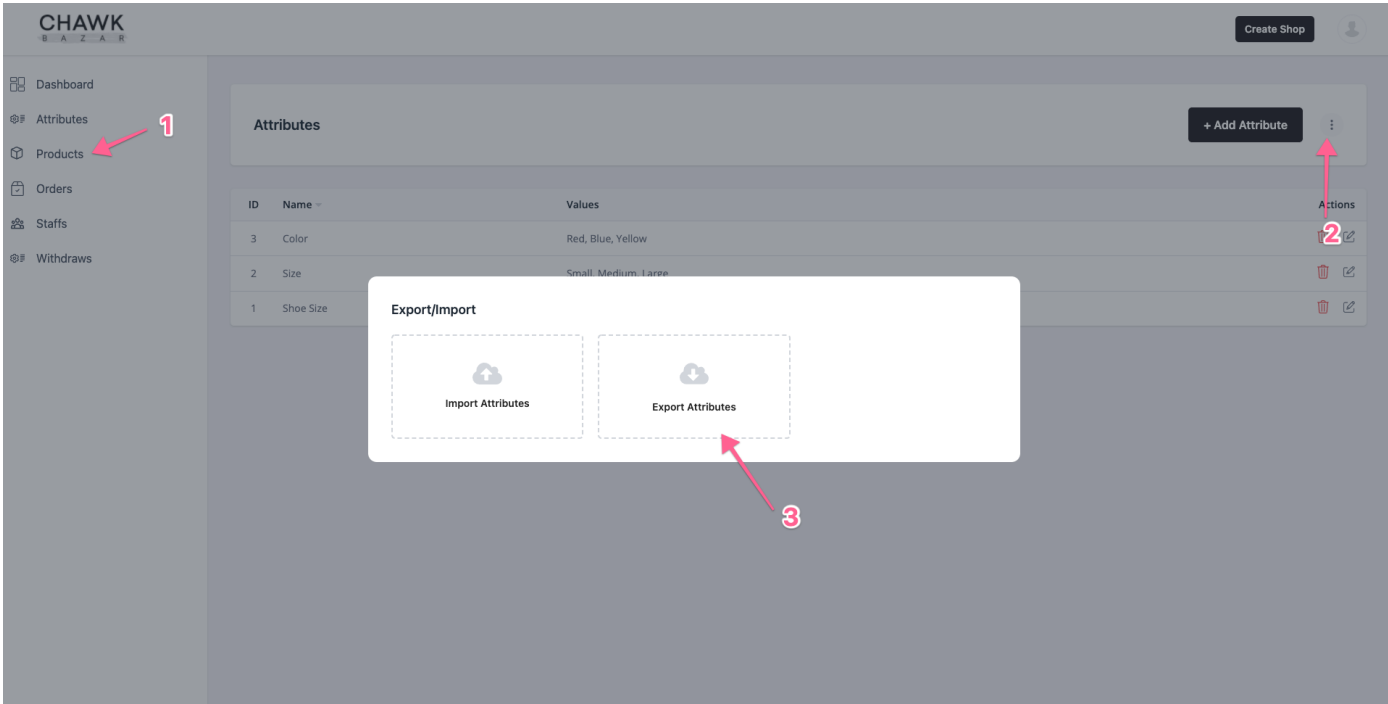
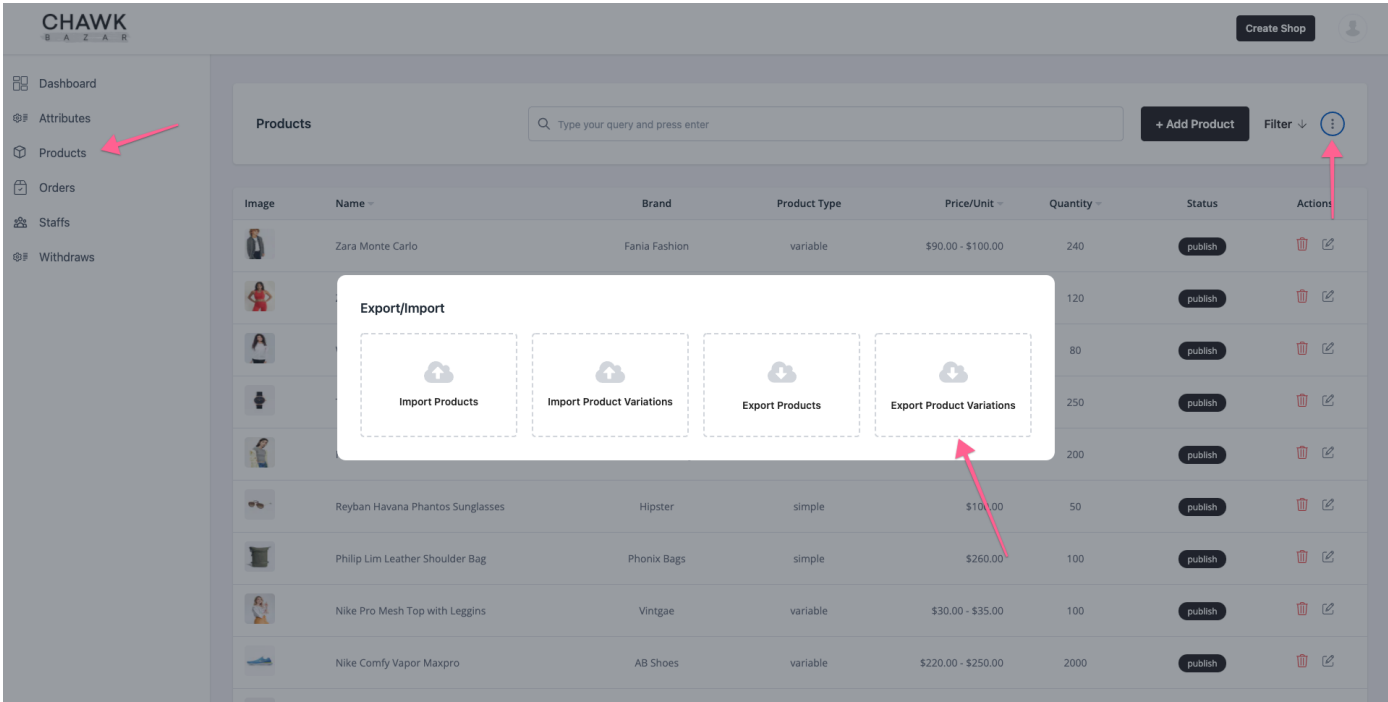
variable products

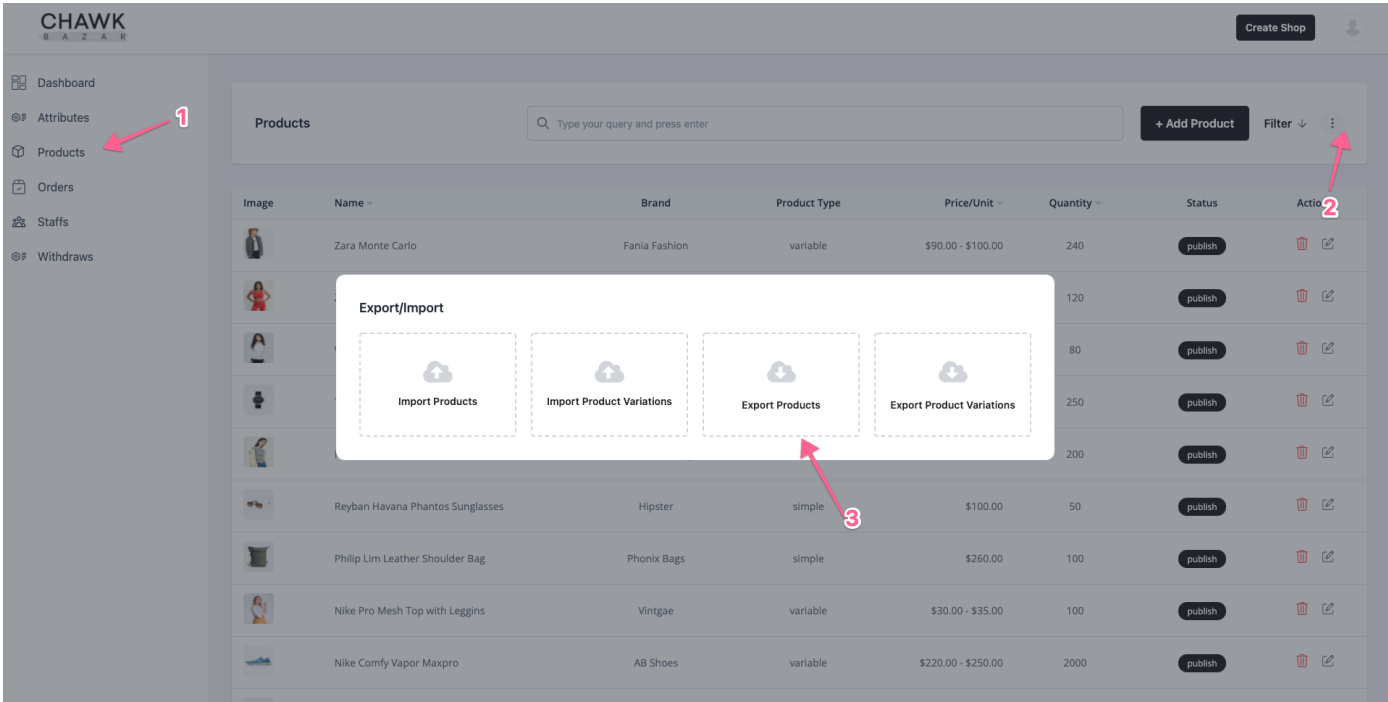
For variable products, you will need three different **csv**.

- Attributes **csv**
- products **csv**
- variation options **csv**

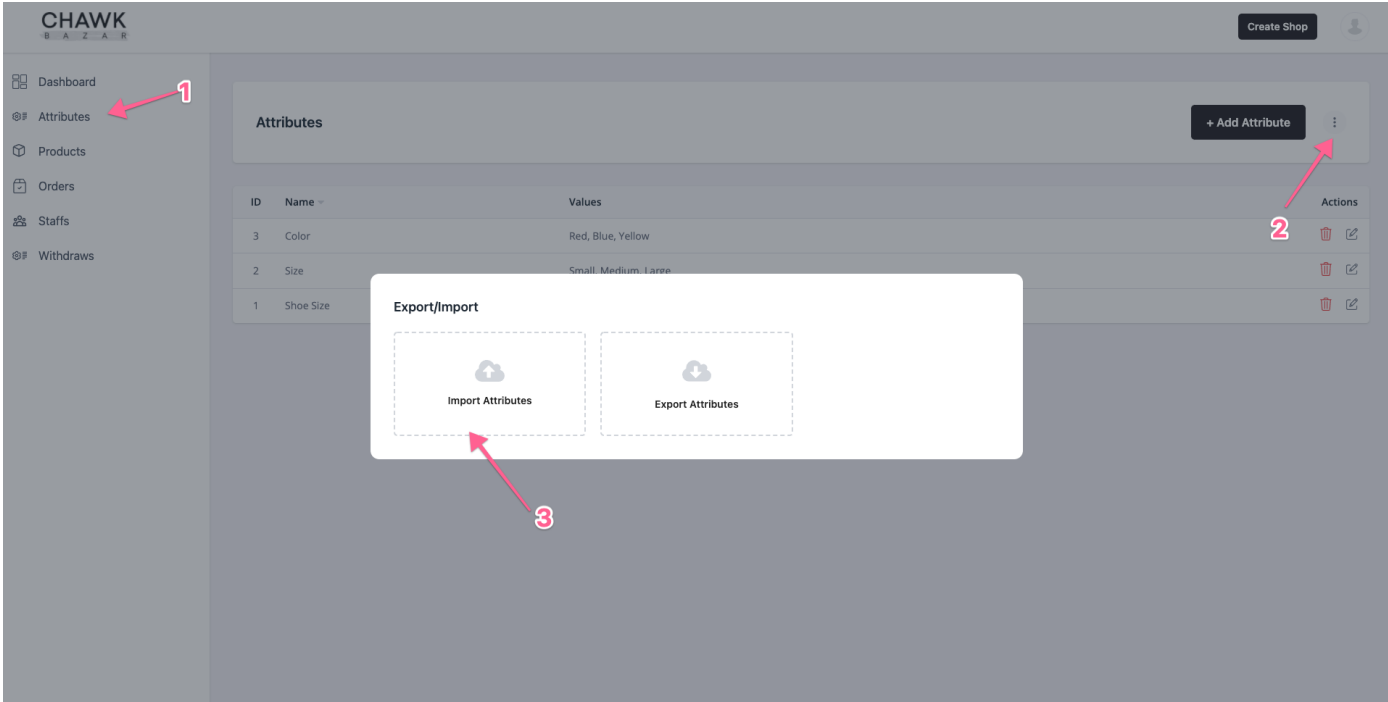
Make sure to re-import you export these three **csv**.

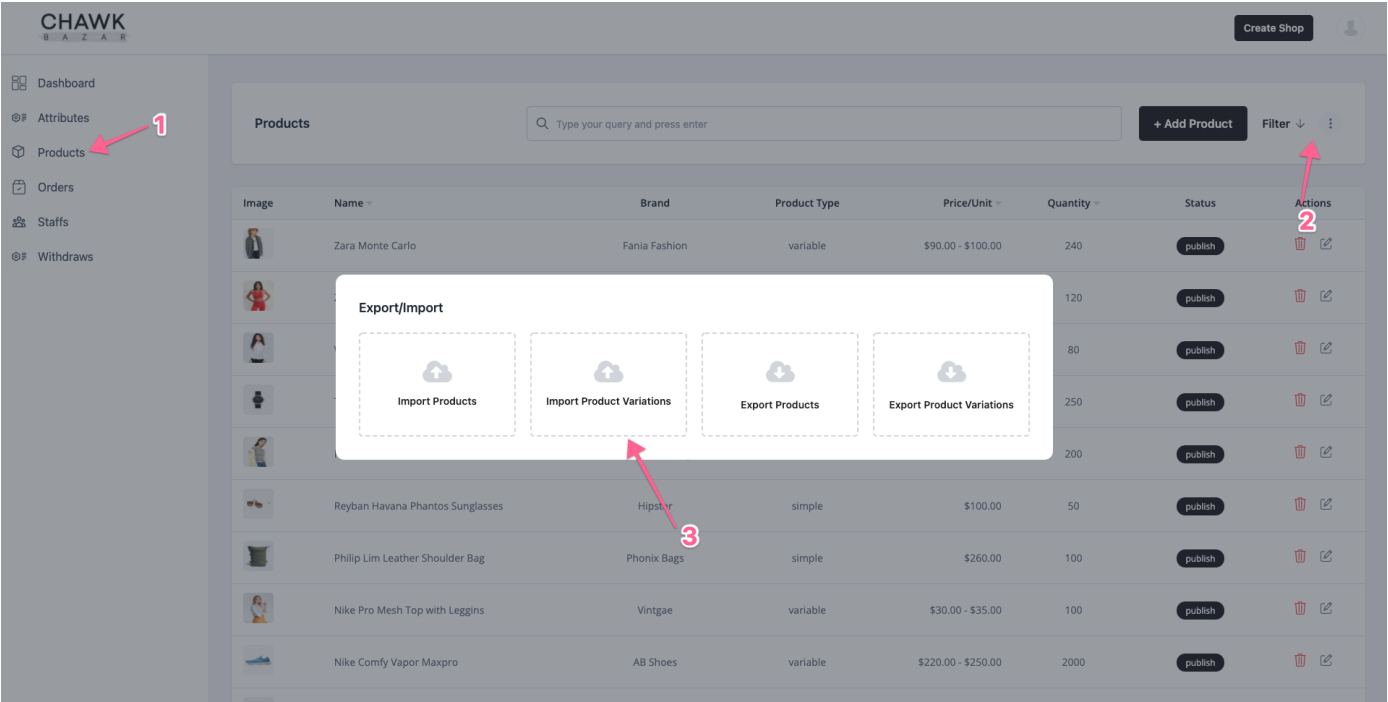
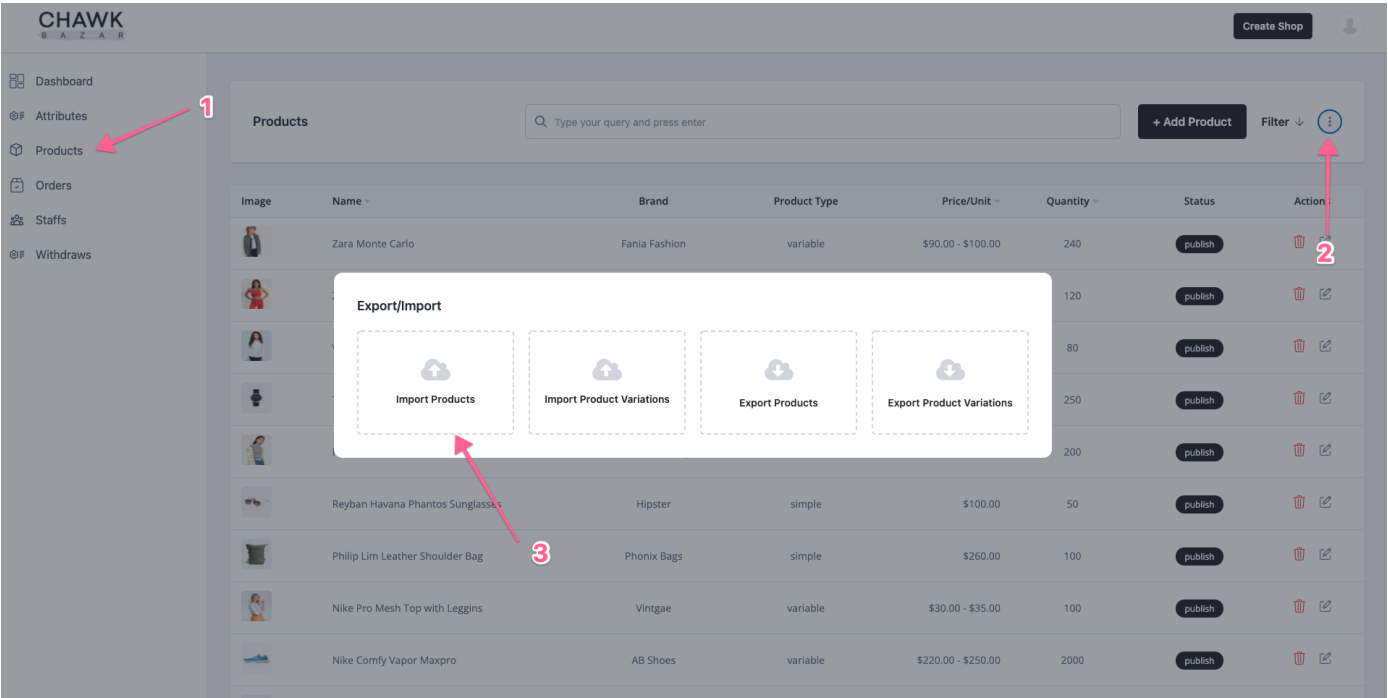






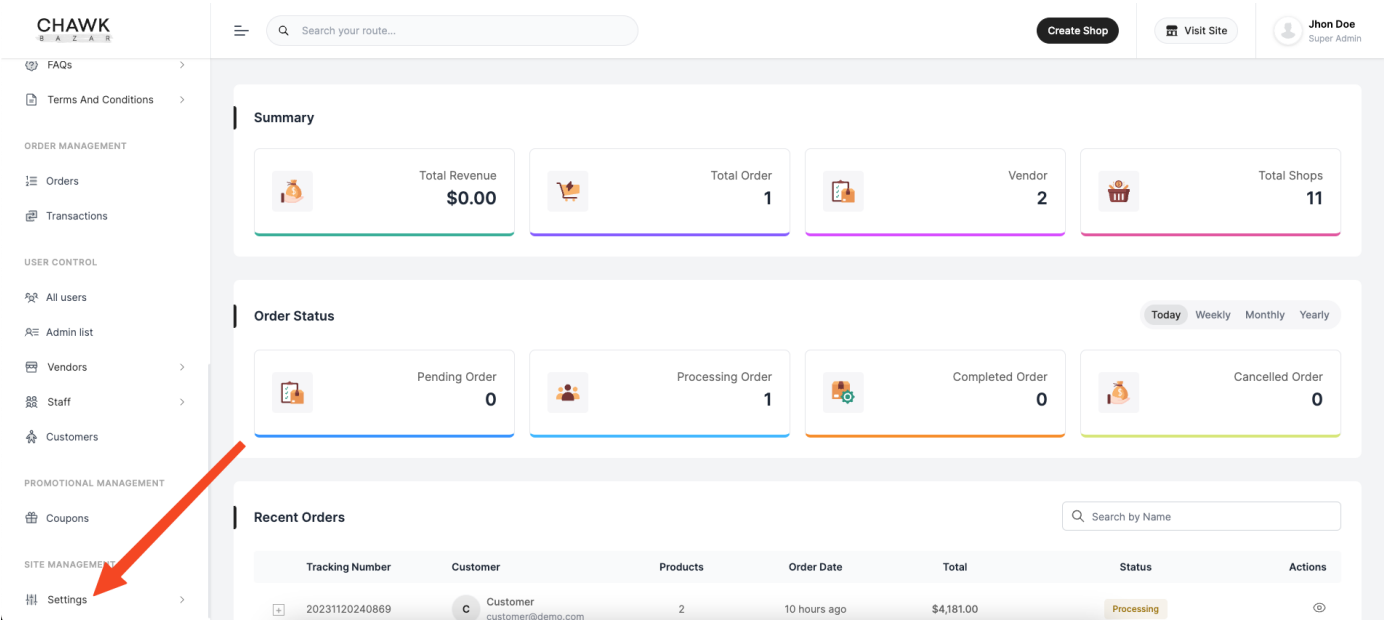
Now, during import variations product, At First, you have to import Attributes then Products and Then products variation.





General settings

In **General 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, Special Features Enable or Disable option etc.

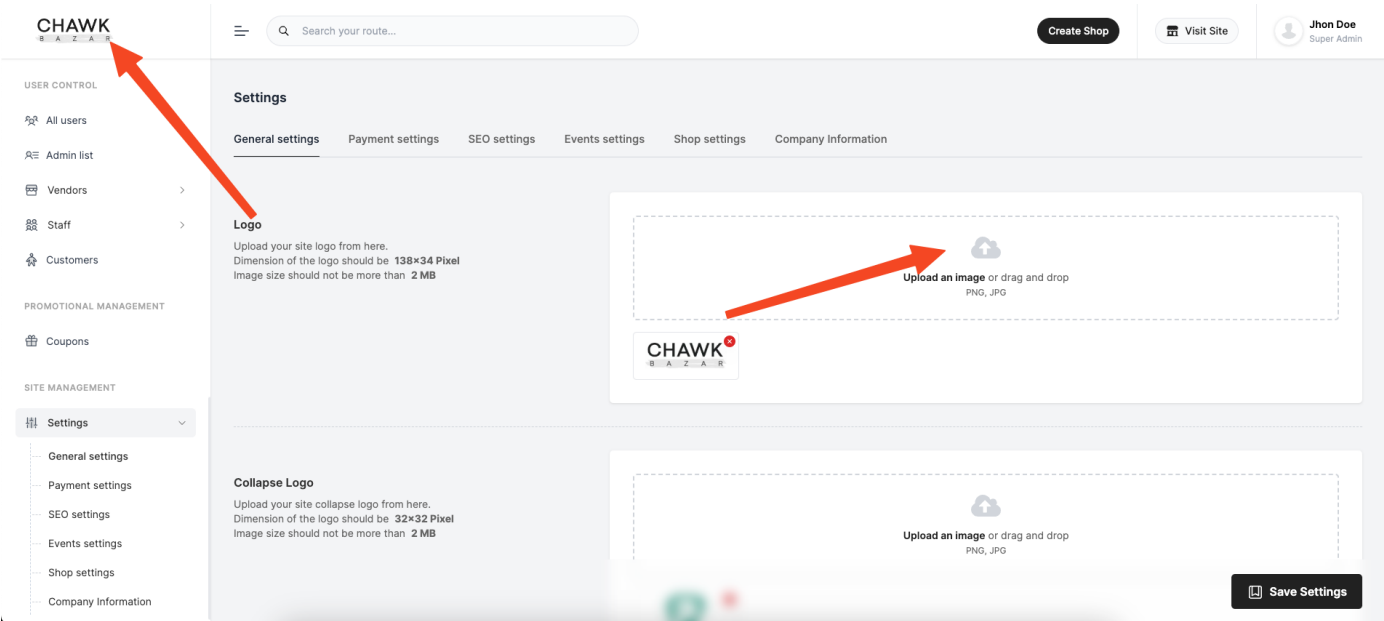


Note: If you make changes or update any fields, it's important to save those changes by clicking on the **Save Settings** button bottom of the page.

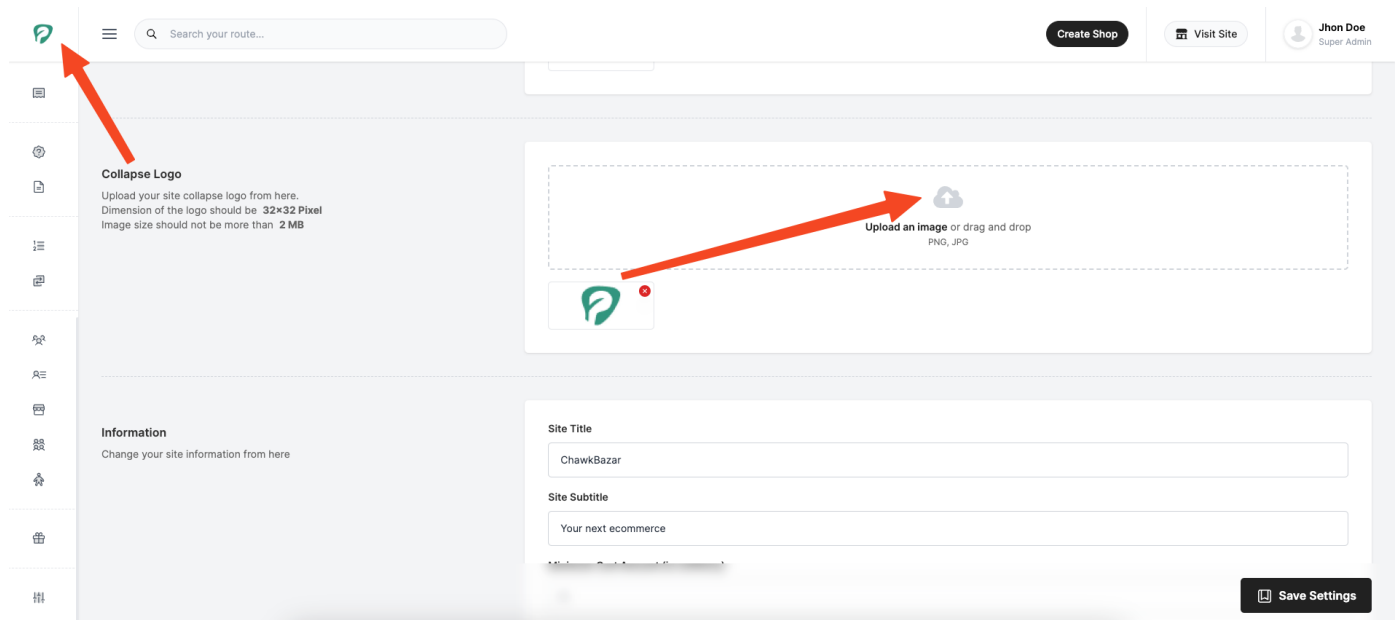
Change Logo

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**)

For General Logo View:



For Collapsible Logo View:



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`.

```
1 APP_NAME=ChawkBazar
2 APP_ENV=production
3 APP_KEY=base64:dLaddfdsfdfdsfssdf/sdfdsfer
4 APP_DEBUG=true
5 APP_URL=https://YOUR_DOMAIN.COM/backend
6 APP_SERVICE=chawkbazar.test
7 APP_NOTICE_DOMAIN=CHAWKBAZAR_
8
9 DUMMY_DATA_PATH=chawkbazar
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

ChawkBazar

Site Subtitle

Your next ecommerce

Minimum Order Amount (in currency)

0

Use OTP at checkout

Enable Must Verify Email

Enable AI

Select AI

openai

Tax Class

Global

Shipping Class

Global

Enable Guest Checkout

Enable Free Shipping

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

Global

Use OTP at checkout

Enable Must Verify Email

Enable Guest Checkout

Enable Free Shipping

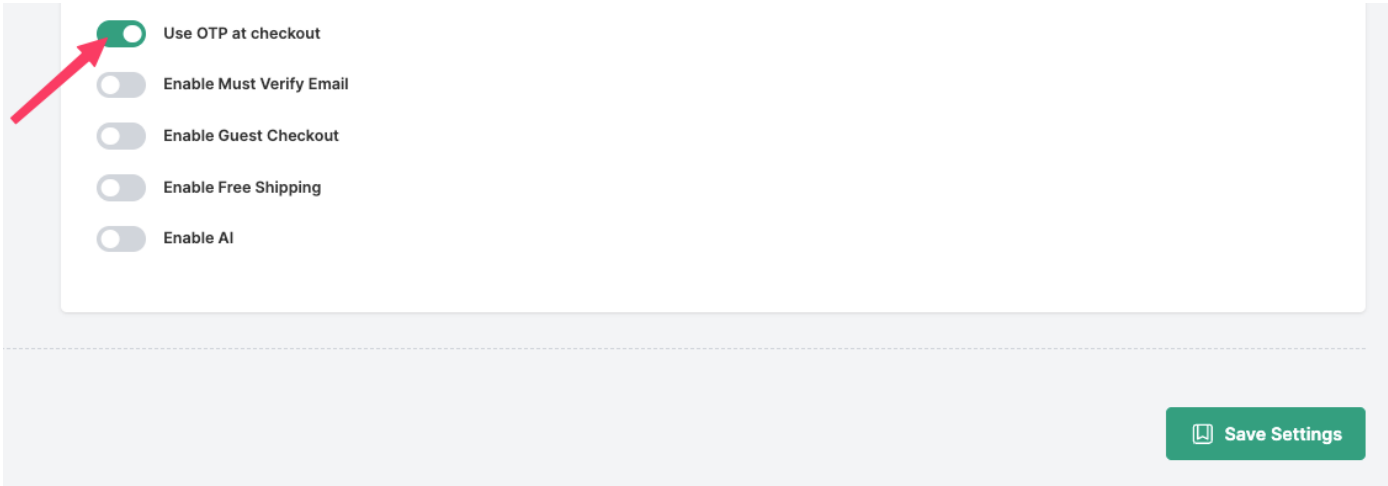
Minimum cart amount for free shipping

minimum cart amount

Enable AI

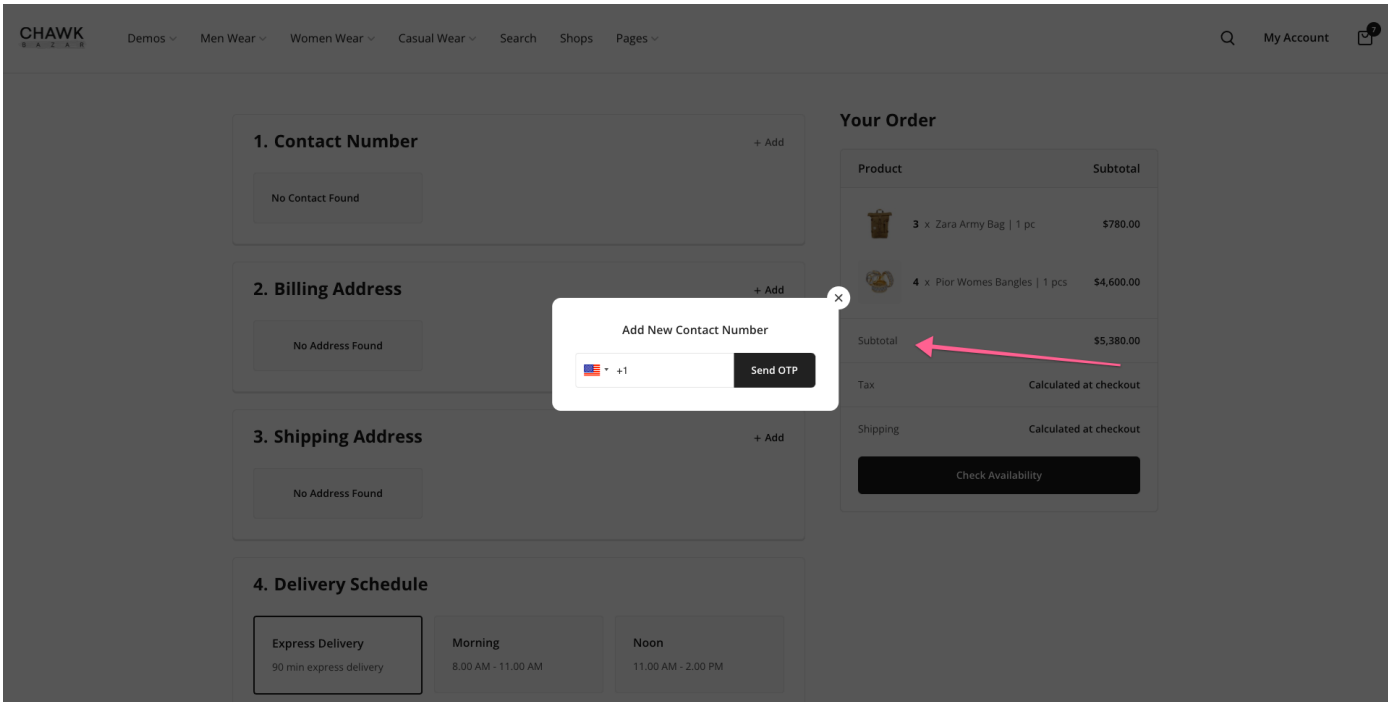
Save Settings

To Enable or Disable Use OTP at checkout

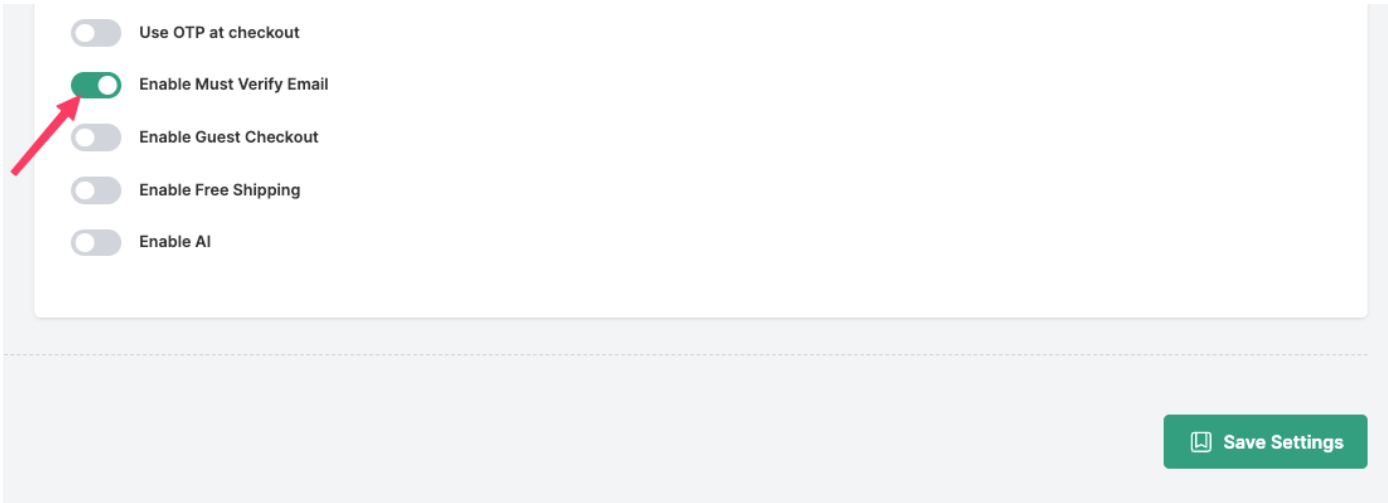


If Enable OTP at checkout, When users go to the checkout page, they'll get a form to verify their mobile number.

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..

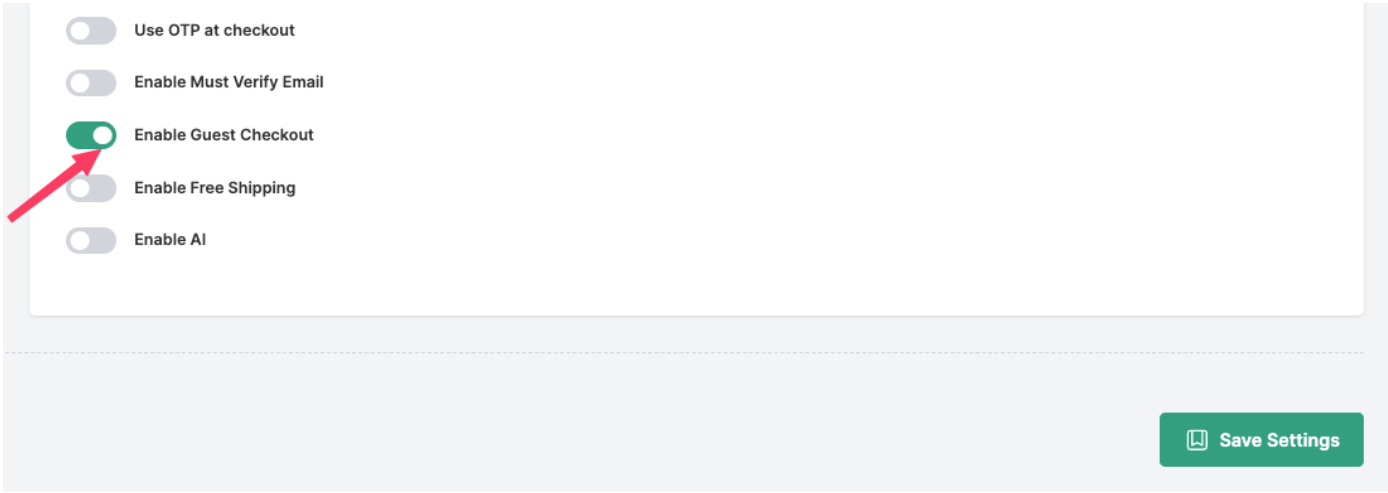


To Enable or Disable Verify Email

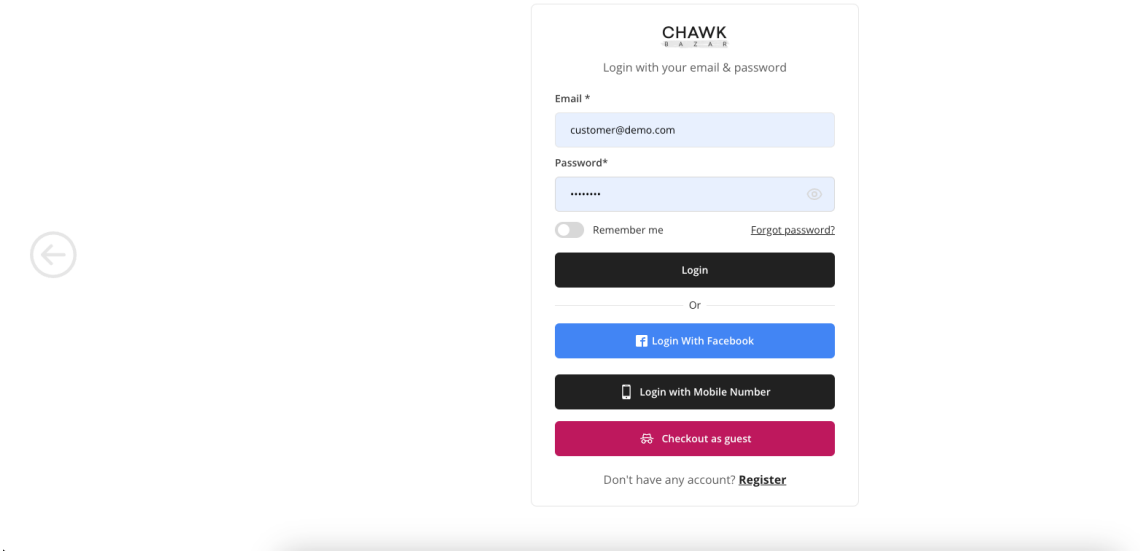


For more information check our [Email-Verification](#) Section.

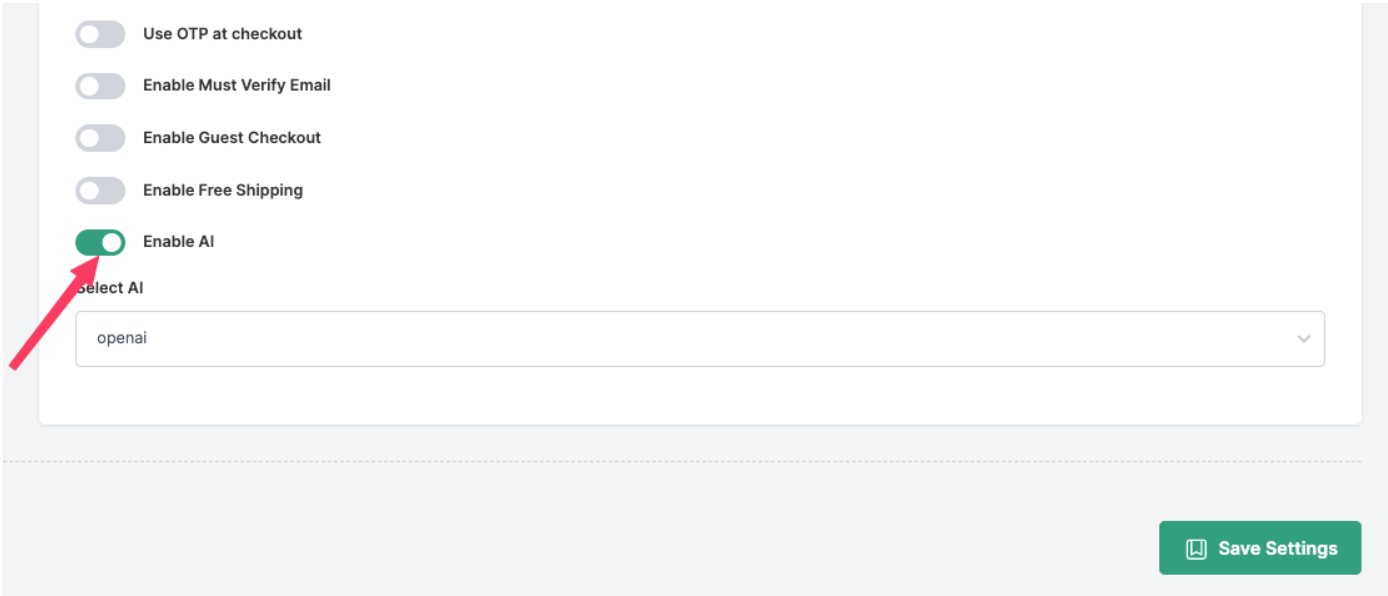
To Enable or Disable Guest Checkout



If guest checkout is enabled, the shop view will allow customers to make purchases without requiring them to create an account or log in.



To Enable or Disable AI



For more information check our [open-ai](#) Docs Section.

Payment settings

This is the **payment** area of the super admin settings. This will allow you to choose a payment gateway such as Stripe, Rezorpay, Paypal, Mollie Paystack etc. You can also enable or disable Cash On Delivery.

Payment

Configure Payment Option

Enable Cash On Delivery

Currency

US Dollar

Enable Gateway

Select Payment Gateway

stripe

PayPal

Razorpay

mollie

paystack

iyzico

Set default payment gateway

Stripe

Webhook URL

stripe

https://chawkbazar-ga-api.redg.io/webhooks/stripe

Save Settings

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.

Total\$265.20

Choose Payment Method

PayPal

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

OG Image

Upload an image or drag and drop

PNG, JPG

Event Settings

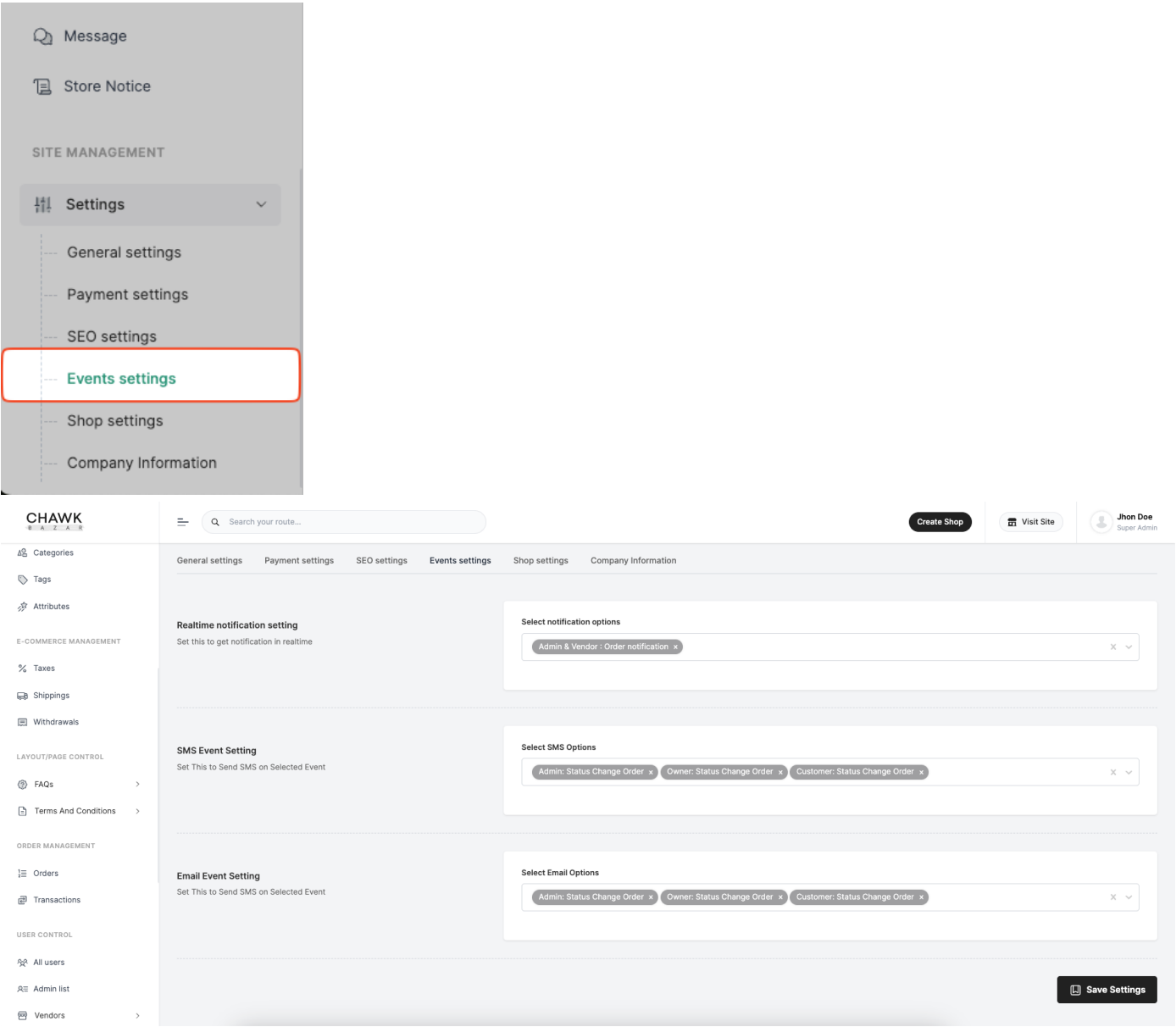
These days, every software incorporates some form of notification or alert system. In order to enhance the customer-friendliness of [Pickbazar Laravel](#), we have implemented such alert systems. However, it is often the case that such systems come with a high cost. Recognizing the importance of cost management, we have developed a solution that is tailored to fit within your budget.

In [Pickbazar Laravel](#), we offer three main types of alert systems, allowing you to toggle them on and off as needed:

All of These panel supports multi-select functionality, allowing you to choose the specific options that suit your needs.

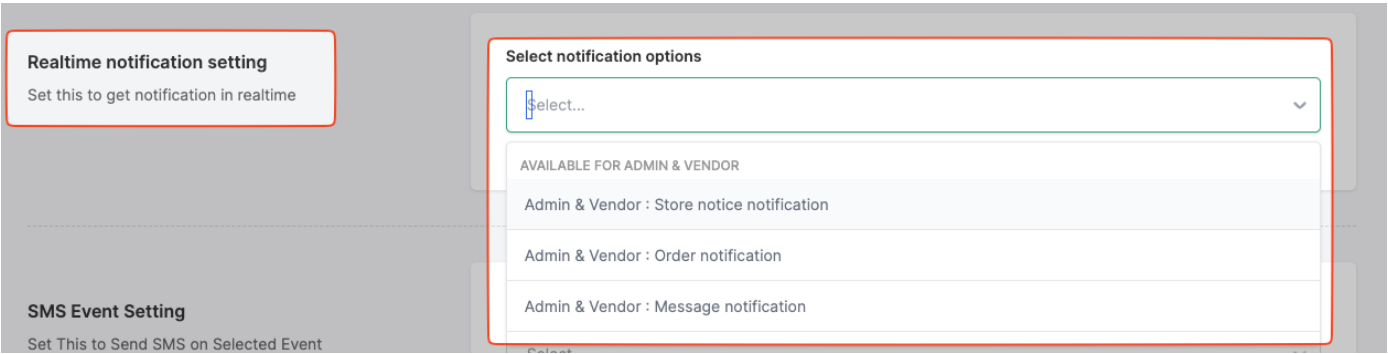
- Real-time Notification Settings
- SMS Event Settings
- Email Event Settings

Event Settings Sidebar panel



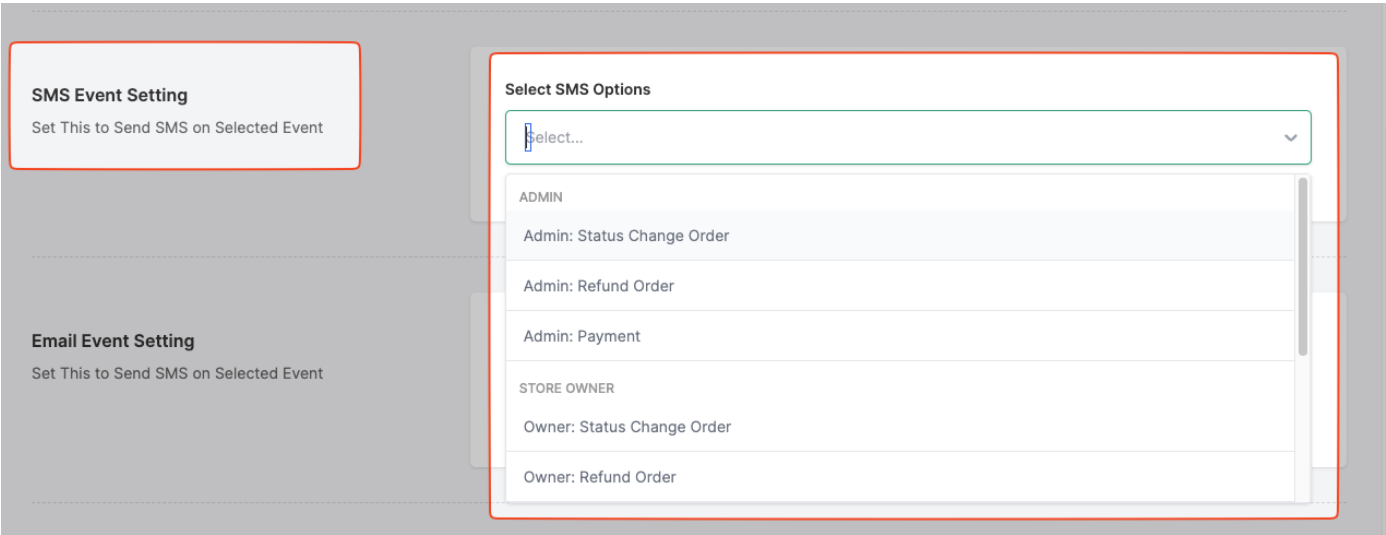
Real-time Notification Settings

These settings provide real-time notifications for store notices, new orders, and new messages. New order and message notifications are applicable to all users with access to the admin panel, excluding store notice notifications for administrators.



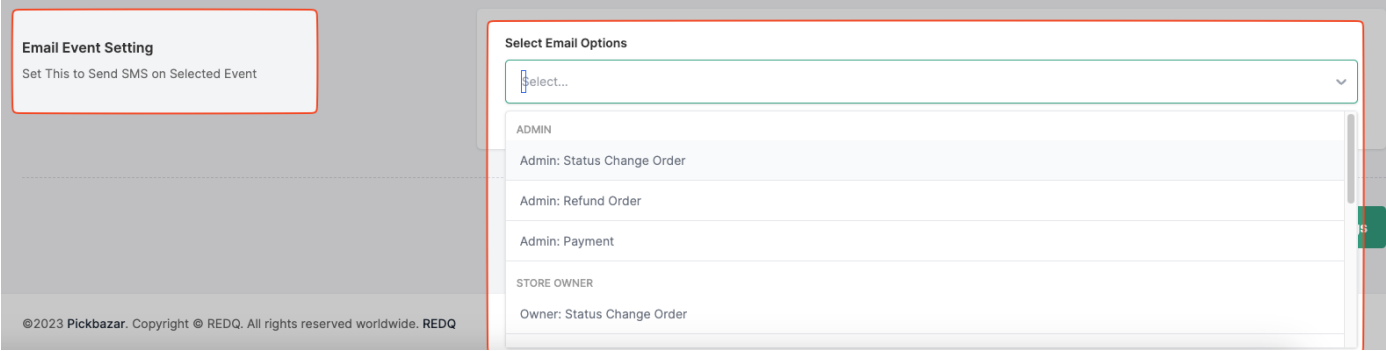
SMS Event Settings

These settings enable SMS notification in various scenarios, categorized by user roles with distinct prefixes.



Email Event Settings

These settings enable Email notification in various scenarios, categorized by user roles with distinct prefixes.



Shop settings

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

Title/Time

Express Delivery

Description

90 min express delivery

Remove

Title/Time

Morning

Description

8.00 AM - 11.00 AM

Remove

Title/Time

Noon

Description

11.00 AM - 2.00 PM

Remove

Title/Time

Shop Settings

This is the **Shop Settings** area of the settings. This will allow you to change Enable product review system before publish, Enable or disable Google Map Address, Enable disable Terms & Conditions for vendors.

Shop Settings

Add your shop settings information from here

Enable product review system before publish ?

Enable Google Map Address

Enable Terms & Conditions for vendors

Save Settings

Company Information

Address

Pickbazar offers a user-friendly experience, giving customers the option to streamline address input. When Google Maps integration is enabled, addresses are automatically fetched from the Google API. Alternatively, users can opt for manual entry, offering flexibility and convenience for all.

address with google map

Settings

General settings

Payment settings

SEO settings

Events settings

Shop settings

Company Information

Address

Add your address from here

Set location from map

Search location form here

addresses are automatically fetched from the Google API

Contact Number

+129290122122

Website

https://redq.io

Email

demo@demo.com

Select social platform

Facebook

X

Add profile url

https://www.facebook.com

Remove

Select social platform

Twitter

X

Add profile url

https://twitter.com/home

Remove

Save Settings

address without google map

Settings

General settings

Payment settings

SEO settings

Events settings

Shop settings

Company Information

Address

Add your address from here

City

manual entry

Country

United States

manual entry

State

NY

manual entry

ZIP

Street Address

manual entry

Contact Number

+129290122122

Website

https://redq.io

Email

demo@demo.com

Select social platform

Add profile url

Save Settings

Social platform

This section you can add new social platform account or remove existing account,

Select social platform

Facebook

X

▼

Add profile url

https://www.facebook.com/redqinc

Remove

Select social platform

Twitter

X

▼

Add profile url

https://twitter.com/RedqTeam

Remove

Select social platform

Instagram

X

▼

Add profile url

https://www.instagram.com/redqteam

Remove

Add New Social Profile

Footer

This section you can add Site Link, Copyright Text, External Text and External Link.

Footer

Change your Footer information from here

Site Link

https://chawkbazar.redq.io/

Copyright Text

Copyright © REDQ. All rights reserved worldwide.

External Text

REDQ

External Link

https://redq.io

Save Settings

New Static Page

Both shop and admin are built using React NextJS framework. So all the existing pages are available to this location,

Shop,

/chawkbazar-laravel/shop/src/pages

Admin,

/chawkbazar-laravel/admin/rest/src/pages

You can use the NextJS routing feature for the new pages. Check these official NextJS docs for pages and routing,

<https://nextjs.org/docs/basic-features/pages>

<https://nextjs.org/docs/routing/introduction>

FAQ:

How to change the default layout?

Please go to,

chawkbazar-laravel/shop/src/pages

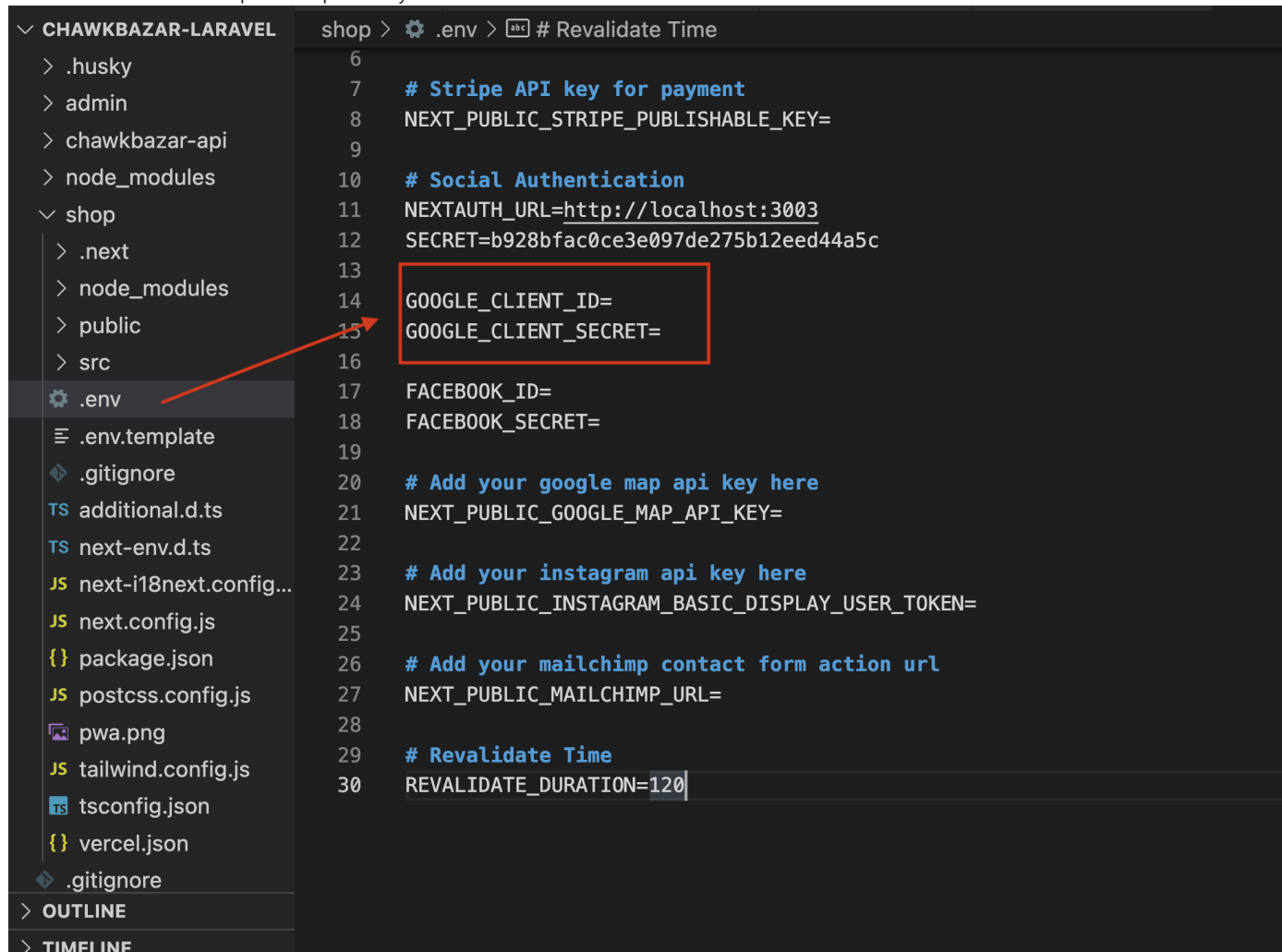
And delete or change the name of `index.tsx`

Then rename `your specific demo.tsx` to `index.tsx`

After that, make sure you rebuild your project to run in production.

Why showing client_id error while Google social login

Ans: Go to `.env` file and change the variables from `GOOGLE_ID=` to `GOOGLE_CLIENT_ID=` and `GOOGLE_SECRET=` to `GOOGLE_CLIENT_SECRET=`. Make this changes in the `.env` file of `chawkbazar-api` and `shop` directory.



```

shop > .env > # Revalidate Time
6
7 # Stripe API key for payment
8 NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=
9
10 # Social Authentication
11 NEXTAUTH_URL=http://localhost:3003
12 SECRET=b928bfac0ce3e097de275b12eed44a5c
13
14 GOOGLE_CLIENT_ID=
15 GOOGLE_CLIENT_SECRET=
16
17 FACEBOOK_ID=
18 FACEBOOK_SECRET=
19
20 # Add your google map api key here
21 NEXT_PUBLIC_GOOGLE_MAP_API_KEY=
22
23 # Add your instagram api key here
24 NEXT_PUBLIC_INSTAGRAM_BASIC_DISPLAY_USER_TOKEN=
25
26 # Add your mailchimp contact form action url
27 NEXT_PUBLIC_MAILCHIMP_URL=
28
29 # Revalidate Time
30 REVALIDATE_DURATION=120

```

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.

API keys [Learn more about API authentication →](#)

Viewing test API keys. Toggle to view live keys. ☐ Viewing test data

Standard keys
These keys will allow you to authenticate API requests. [Learn more](#)

NAME	TOKEN	LAST USED	CREATED
Publishable key	pk_test_26a0ZU... 9vgNqr...	Jul 18	Apr 11
Secret key	Reveal test key	Jul 18	Jul 10

Restricted keys
For greater security, you can create restricted API keys that limit access and permissions for different areas of your account data. [Learn more](#) [+ Create restricted key](#)

NAME	TOKEN	LAST USED	CREATED
No restricted keys			

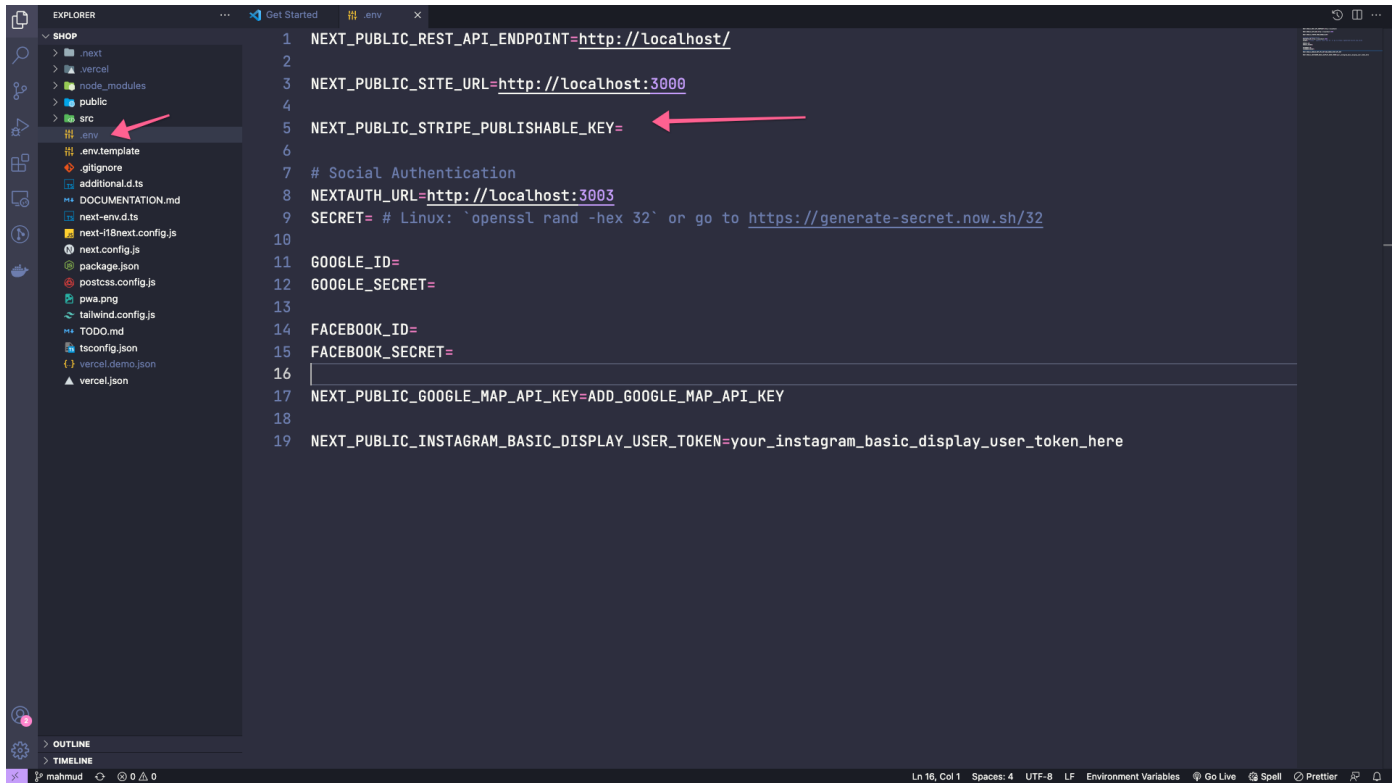
4. Then copy **Secret key** to your **chawkbazar api->.env**

```

43 MAIL_FROM_ADDRESS=
44
45 AWS_ACCESS_KEY_ID=
46 AWS_SECRET_ACCESS_KEY=
47 AWS_DEFAULT_REGION=
48 AWS_BUCKET=
49
50 STRIPE_API_KEY=
51
52
53 LIGHTHOUSE_CACHE_ENABLE=false
54 MEDIA_DISK=public
55
56 SHOP_URL=
57 DASHBOARD_URL=
58 ADMIN_EMAIL=support@example.com
59
60 GOOGLE_CLIENT_ID=
61 GOOGLE_CLIENT_SECRET=
62 GOOGLE_REDIRECT_URI=http://127.0.0.1/login/google/callback
63
64 FACEBOOK_CLIENT_ID=
65 FACEBOOK_CLIENT_SECRET=
66 FACEBOOK_REDIRECT_URI=https://127.0.0.1/login/facebook/callback
67
68 ACTIVE_OTP_GATEWAY=twilio
69
70 TWILIO_AUTH_TOKEN=
71 TWILIO_ACCOUNT_SID=
72 TWILIO_VERIFICATION_SID=
73
74 MESSAGEBIRD_API_KEY=
75 MESSAGEBIRD_ORIGINATOR=Marvel

```

5. And similarly, add **Publishable key** to your **frontend/shop/.env**



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

For REST API

```

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

```

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`

- 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",
"chawkbazar/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": {
    "chawkbazar/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`

I am getting "Cannot use 'Mixed' as class name as it is reserved" after access graphql endpoint.

You are using the application in PHP 8. Downgrade it to PHP 7.4. Mixed is a reserved word in PHP 8.

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?

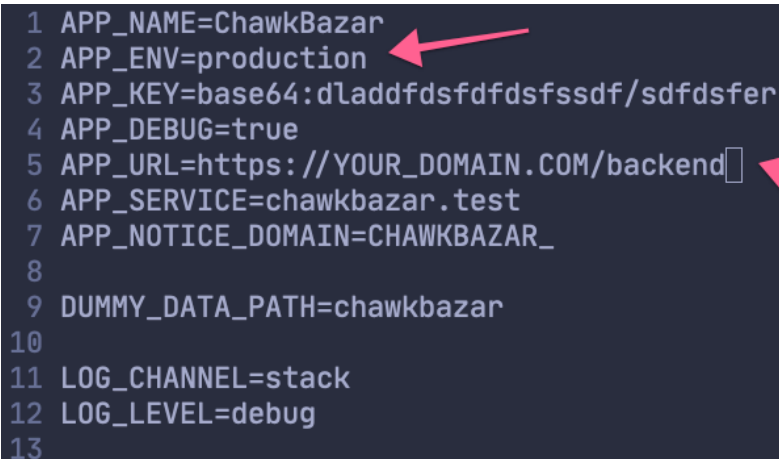
We support two types of file uploads; one is `local`, and another is `AWS S3`.

Both uploading systems follow this procedure,

API

At first, add your API URL to your `chawkbazar-api` -> `.env`.

```
1 APP_NAME=ChawkBazar  
2 APP_ENV=production  
3 APP_KEY=base64:d1addfdsfdfsfssdf/sdfdsfer  
4 APP_DEBUG=true  
5 APP_URL=https://YOUR_DOMAIN.COM/backend  
6 APP_SERVICE=chawkbazar.test  
7 APP_NOTICE_DOMAIN=CHAWKBAZAR_  
8  
9 DUMMY_DATA_PATH=chawkbazar  
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 `chawkbazar-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 `chawkbazar-laravel` -> `admin` -> `rest` -> `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",
  ],
},
```

Like this

Shop

Similarly, add your domain to your `chawkbazar-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",
  ],
},
```

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://chawkbazar-laravel-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:

```
3003 -> Shop Rest  
3002 -> Admin Rest
```

To check that at the first stop the PM2 instance by this command,

```
pm2 stop 'all'
```

then go to the **chawkbazar** 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
```

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 **chawkbazar** 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
```

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://chawkbazar-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.

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/chawkbazar-laravel`

For `shop`,

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

And for `admin`,

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

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 `chawkbazar-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,

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

Then build the admin,

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

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/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/chawkbazar
```

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,

```
chawkbazar-api/packages/marvel/config/media-library.php
```

```
'max_file_size' => 1024 * 1024 * 256,
```

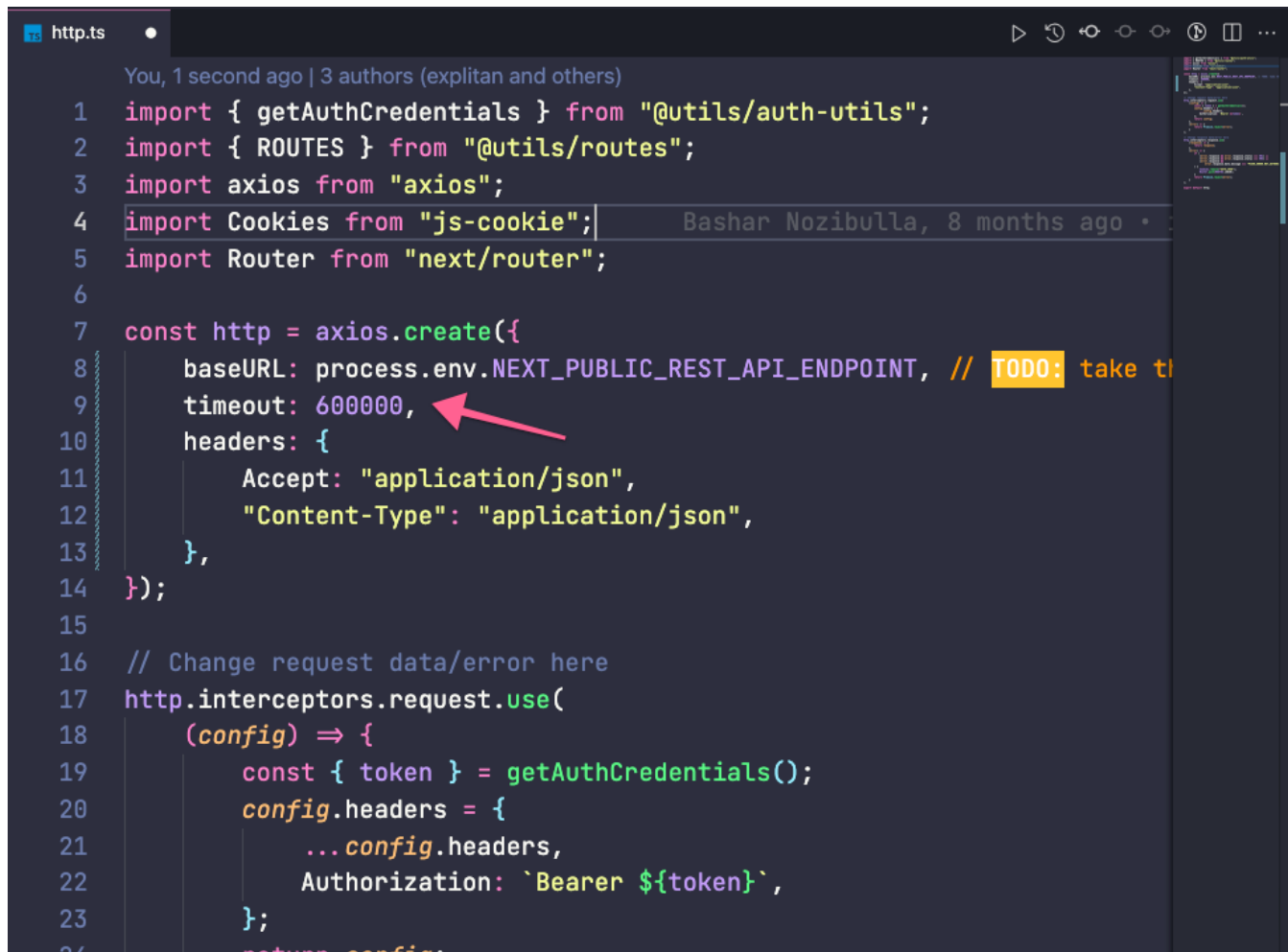
```
...
1  <?php
2
3  return [
4
5      /*
6       * The disk on which to store added files and derived images by default.
7       * one or more of the disks you've configured in config/filesystems.
8       */
9      'disk_name' => env('MEDIA_DISK', config('shop.media_disk')),
10
11     /*
12      * 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 `chawkbazar-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 ti
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,

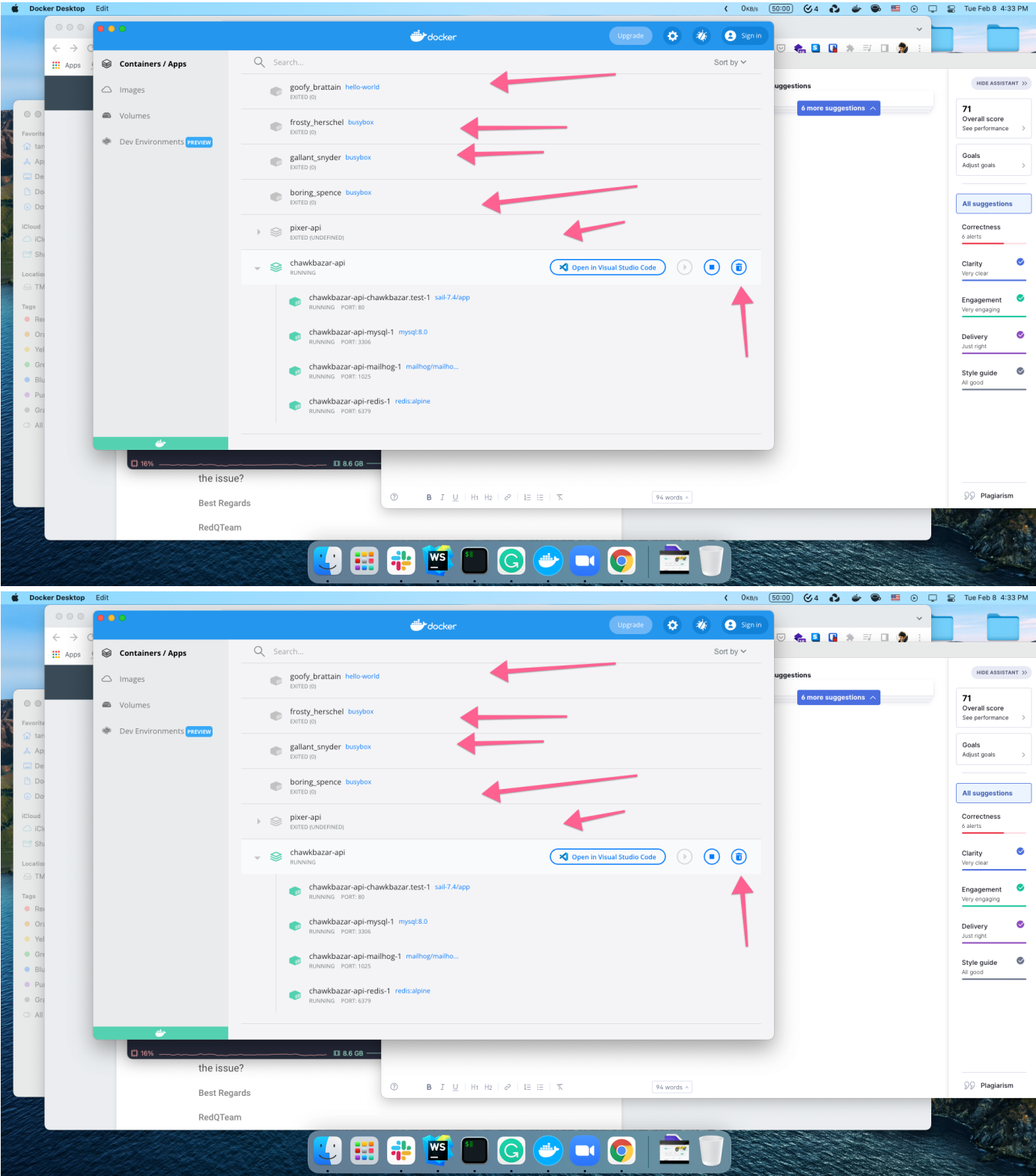
```
chawkbazar-laravel -> chawkbazar-api
```

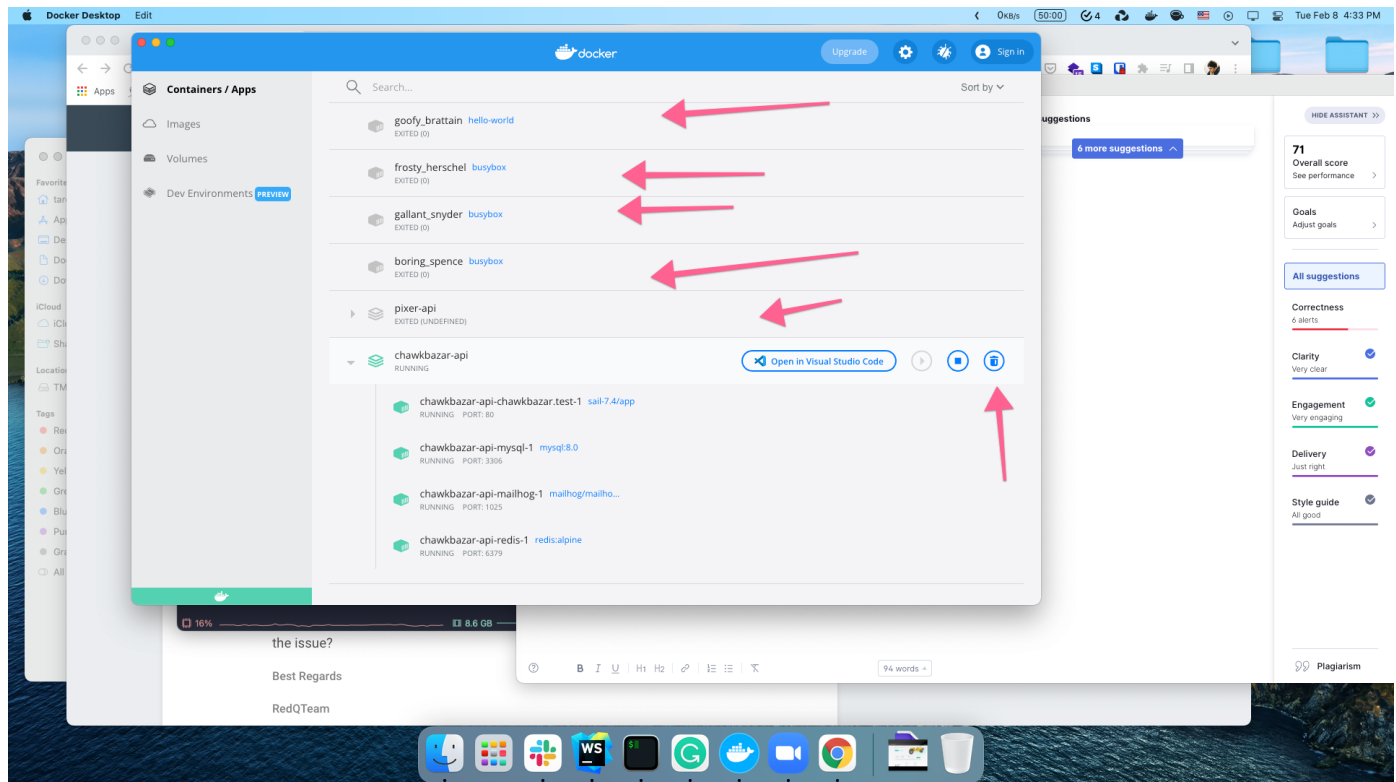
and use this command,

```
./vendor/bin/sail down -v
```

After that,

Open docker dashboard and delete all the existing container and images,





And after that remove,

```
chawkbazar-laravel -> chawkbazar-api -> .env
```

And then go to your root `chawkbazar-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 disable product popup view and redirect to a single product page?

At first edit `shop/src/components/product/product-card.tsx`

Import these two lines,

```
import { useRouter } from "next/router";
import { ROUTES } from "@lib/routes";
```

```

1 import cn from "classnames";
2 import Image from "next/image";
3 import type { FC } from "react";
4 import { useUI } from "@contexts/ui.context";
5 import usePrice from "@lib/use-price";
6 import { Product } from "@framework/types";
7 import { siteSettings } from "@settings/site.settings";
8 import { useRouter } from "next/router";
9 import { ROUTES } from "@lib/routes";
10
11 interface ProductProps {
12   product: Product;
13   className?: string;
14   contactClassName?: string;
15   imageContentClassName?: string;
16   variant?: "grid" | "gridSlim" | "list" | "listSmall";
17   imgWidth?: number | string;
18   imgHeight?: number | string;
19   imgLoading?: "eager" | "lazy";
20 }
21
22 const ProductCard: FC<ProductProps> = ({
23   product,
24   className = "",
25   contactClassName = "",
26   imageContentClassName = "",
27   variant = "list",
28   imgWidth = 340,
29   imgHeight = 440,
30   imgLoading,
31 }) => {
32   const router = useRouter();
33   const { openModal, setModalView, setModalData } = useUI();
34   const { name, image, min_price, max_price, product_type, description } =
35     product ?? {};
36
37   const { price, basePrice } = usePrice({
38     amount: product?.sale_price ? product?.sale_price : product?.price!

```

Then call useRouter,

```
const router = useRouter();
```

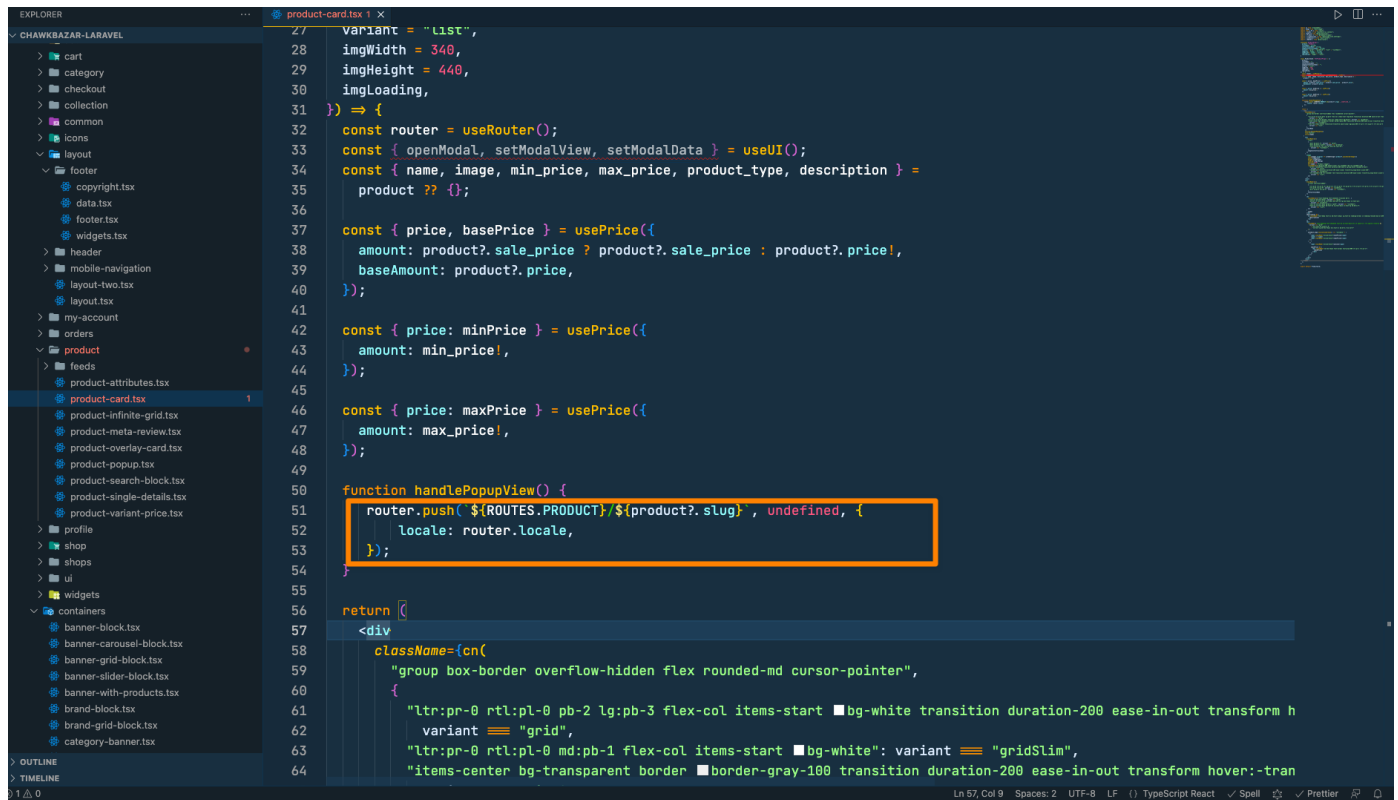
```

23   product,
24   className = "",
25   contactClassName = "",
26   imageContentClassName = "",
27   variant = "list",
28   imgWidth = 340,
29   imgHeight = 440,
30   imgLoading,
31 ) => {
32   const router = useRouter();
33   const { openModal, setModalView, setModalData } = useUI();
34   const { name, image, min_price, max_price, product_type, description } =
35     product ?? {};
36
37   const { price, basePrice } = usePrice({
38     amount: product?.sale_price ? product?.sale_price : product?.price!,
39     baseAmount: product?.price,
40   });
41
42   const { price: minPrice } = usePrice({
43     amount: min_price!,
44   });
45
46   const { price: maxPrice } = usePrice({
47     amount: max_price!,
48   });
49
50   function handlePopupView() {
51     router.push(`${ROUTES.PRODUCT}/${product?.slug}`, undefined, {
52       locale: router.locale,
53     });
54   }
55
56   return (
57     <div
58       className={cn(
59         "group box-border overflow-hidden flex rounded-md cursor-pointer",
60

```

Then update handlePopupView with this code,

```
router.push(`${ROUTES.PRODUCT}/${product?.slug}`, undefined, {
  locale: router.locale,
});
```



```
27 variant = "list",
28 imgWidth = 340,
29 imgHeight = 440,
30 imgLoading,
31 } => {
32   const router = useRouter();
33   const { openModal, setModalView, setModalData } = useUI();
34   const { name, image, min_price, max_price, product_type, description } =
35     product ?? {};
36
37   const { price, basePrice } = usePrice({
38     amount: product?.sale_price ? product?.sale_price : product?.price!,
39     baseAmount: product?.price,
40   });
41
42   const { price: minPrice } = usePrice({
43     amount: min_price!,
44   });
45
46   const { price: maxPrice } = usePrice({
47     amount: max_price!,
48   });
49
50   function handlePopupView() {
51     router.push(`${ROUTES.PRODUCT}/${product?.slug}`, undefined, {
52       locale: router.locale,
53     });
54   }
55
56   return (
57     <div
58       className={cn(
59         "group box-border overflow-hidden flex rounded-md cursor-pointer",
60         "ltr:pr-0 rtl:pl-0 pb-2 lg:pb-3 flex-col items-start bg-white transition duration-200 ease-in-out transform h",
61         variant === "grid",
62         "ltr:pr-0 rtl:pl-0 md:pb-1 flex-col items-start bg-white": variant === "gridSlim",
63         "items-center bg-transparent border border-gray-100 transition duration-200 ease-in-out transform hover:-tran
64   )}
```

Support

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)
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 to provide standard support for our items

Thank You!