

Introduction

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

Navigation

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

Supported Platforms For Local Development

- Compatible Browsers (Firefox, Safari, Chrome, Edge)
- Server(Node.js 12.13.0 or later)
- MacOS, Windows, and Linux are supported

Requirements

- node(12.13.0 or later)
- yarn
- PHP 7.4
- MySQL 8
- ext_curl
- editor: Visual Studio Code(recommended)

“

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

<https://www.youtube.com/playlist?list=PLUT1MYLrVpA9-p3tDlWzJgHmJNBwoz7IQ>

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

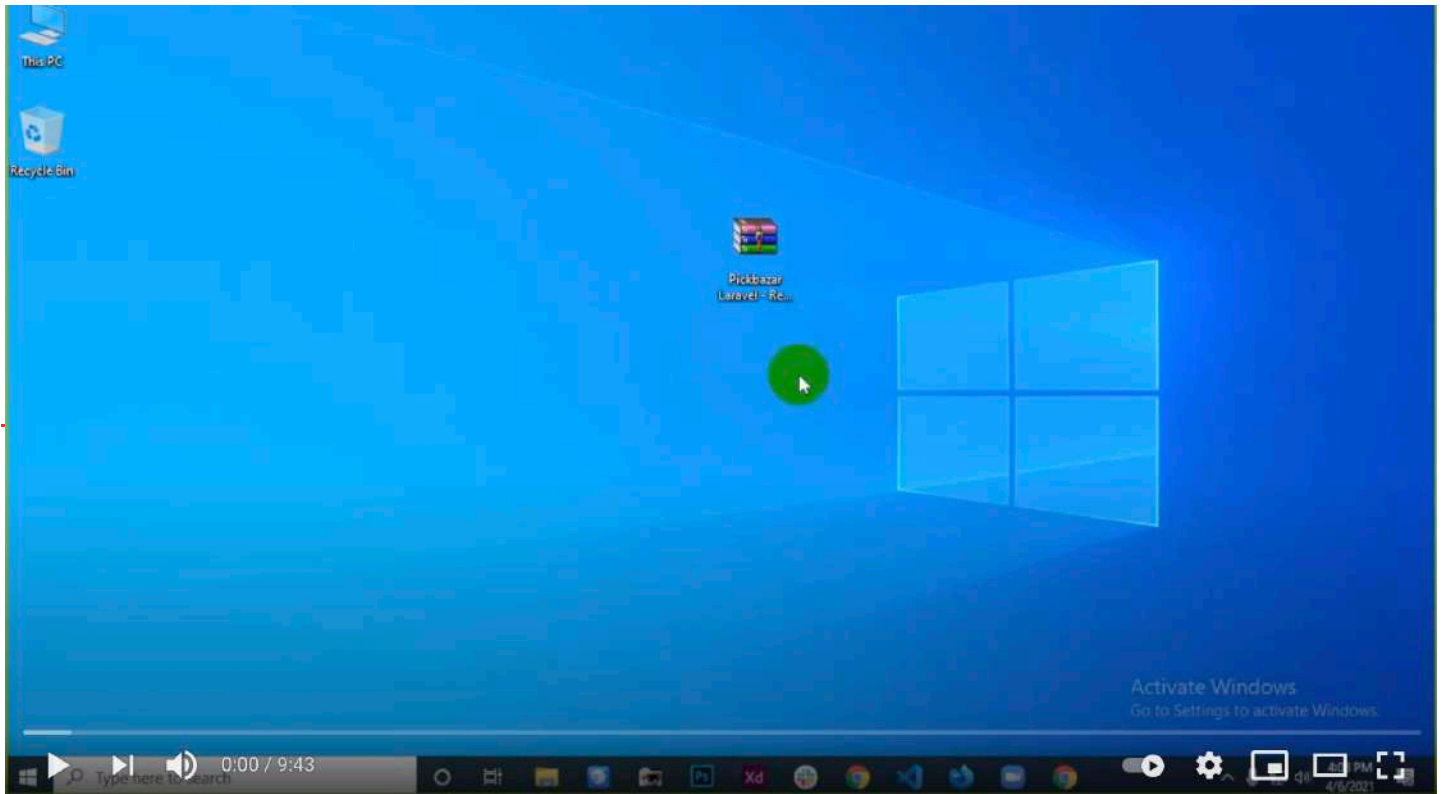
| API

- 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



Prerequisites

- PHP 7.4
- 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(12.13.0 or later)
- yarn
- editor: Visual Studio Code (recommended)

| Getting Started

First download the file from codecanyon.

Unzip the downloaded file and folder structure you get



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

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: ml-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 chawkbazar:install` and follow necessary steps.

```

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**

https://www.youtube.com/watch?v=Cb_jb2N2UYM

- 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

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`

Run yarn at the root directory.

```
# on chawkbazar/root directory
yarn
```

• 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 \Rightarrow <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 Check :

- For snop :


- 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 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]
├   └ 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/a09552d4f74729e02303abd0a7d42186859f2ec0.37727a.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.

For development purpose we use yarn workspace
if you want to use it then see the

if you want to use it then see the `package.json` file at root, for various workspace specific command.

- if you prefer single template then just copy the required template folder and separate them. you'll find their `package.json` file within them and follow the command for dev, build, start.

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

Admin dashboard

- **Analytics Dashboard**

You will get a complete analytics dashboard to

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

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



Recent Orders



















Tracking Number	Total	Order Date	Status
GkszijsVAcac	\$2,744.33	5 days ago	Order Received
lrrdghqshkys	\$1,748.30	14 days ago	Order Dispatched
iTUVhT87QdE8	\$526.98	17 days ago	Order Dispatched
A9mvTnzhGxkc	\$294.49	17 days ago	Order Received
BLOnTtmsHHHE	\$12.80	17 days ago	Order Received
ruIb1kz4UIIN	\$2.45	19 days ago	Shipment Refused by Consignee
NqlykeIW6awe	\$97.80	18 days ago	Delivered
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			
		<input type="text" value="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

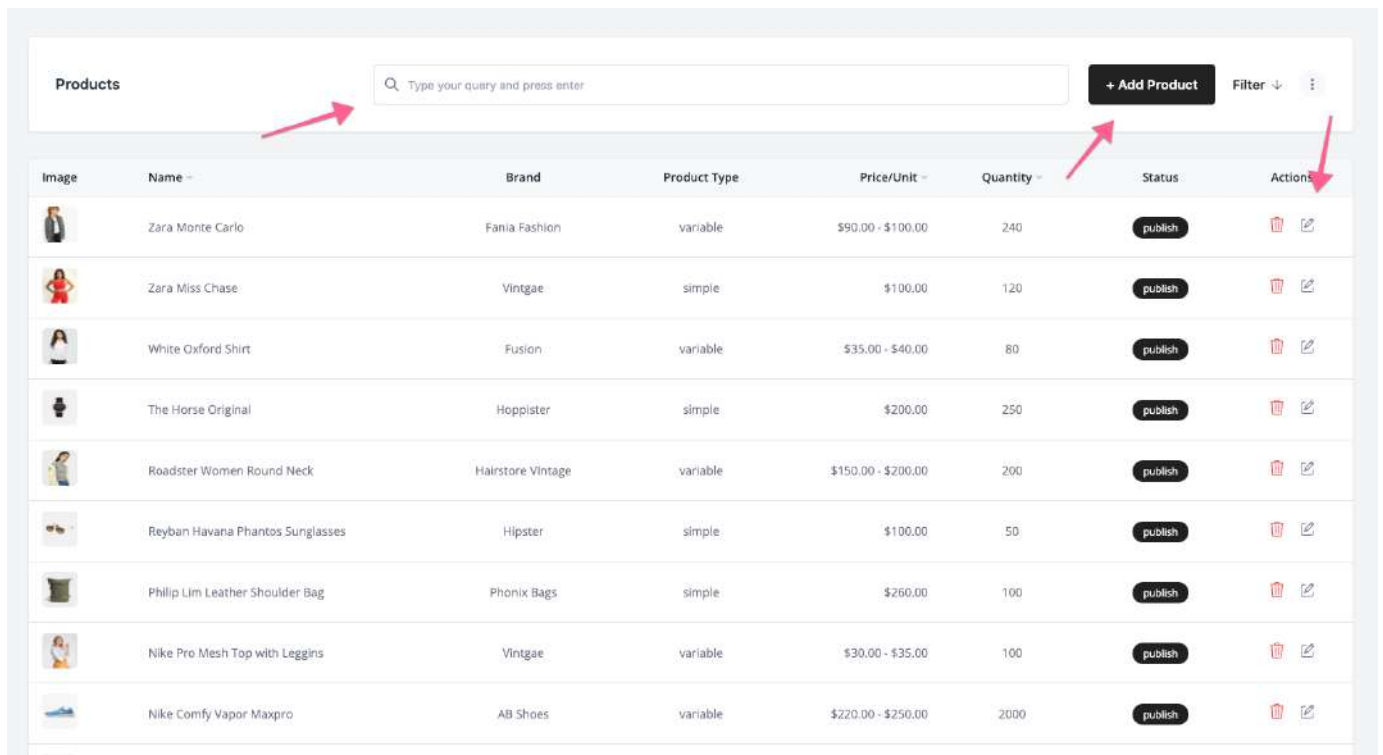
In **Categories** menu you will get the product types and you can add, remove or modify product categories from there.

Categories						
				<input type="text" value="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	 

Product Management

- PRODUCT Management

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



The screenshot shows a 'Products' management interface. At the top, there is a search bar with the placeholder text 'Type your query and press enter', a '+ Add Product' button, and a 'Filter' dropdown. Below this is a table with the following columns: Image, Name, Brand, Product Type, Price/Unit, Quantity, Status, and Actions. The table contains ten product entries. Red arrows point to the search bar, the '+ Add Product' button, and the Actions column.

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


- 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

Tags

Gift Collection Kids Collection

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.

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	

Order Management

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

Orders							
<input type="text" value="Type your query and press enter"/>							
	Tracking Number	Delivery Fee	Total	Order Date	Status	Shipping Address	Download
	a1bkQKf4na1K	\$50.00	\$243.80	2 hours ago	Order Received	3886 Froe Street, Mill Creek, WV, 26280, United States	Download
	0DGfGYdkM5w8	\$50.00	\$2,804.00	2 days ago	Order Received	f, fsdfs, fsdf, fsdf	Download
	4yQV4m38mbBe	\$50.00	\$845.60	3 days ago	Order Received	Cumilla, Cumilla, Cumilla, 1600, Cumilla	Download
	lZndjRHpKCc3	\$50.00	\$1,136.30	5 days ago	Order Received	5/1-A, 5BC Lane, Holiday Park, Gandaria, Dhaka, Dhupkhola maath, Pukur par, Dhaka, Dhaka, 1204, Bangladesh	Download
	PGCb83dPpUc3	\$50.00	\$4,181.00	5 days ago	Order Received	5/1-A, 5BC Lane, Holiday Park, Gandaria, Dhaka, Dhupkhola maath, Pukur par, Dhaka, Dhaka, 1204, Bangladesh	Download

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

Order ID - PGCb83dPpUc3

Order Received

Order Processing

Ready To

Order Received

Order Processing

Ready To

Products

Tay Ben Aviator x 3

Sub total

Tax

Delivery fee

Discount

Total

\$4,050.00

\$81.00

\$50.00

\$0.00

\$4,181.00

Select...

Order Received

Order Processing

Ready To Dispatch

Order Dispatched

At Local Facility

Out For Delivery

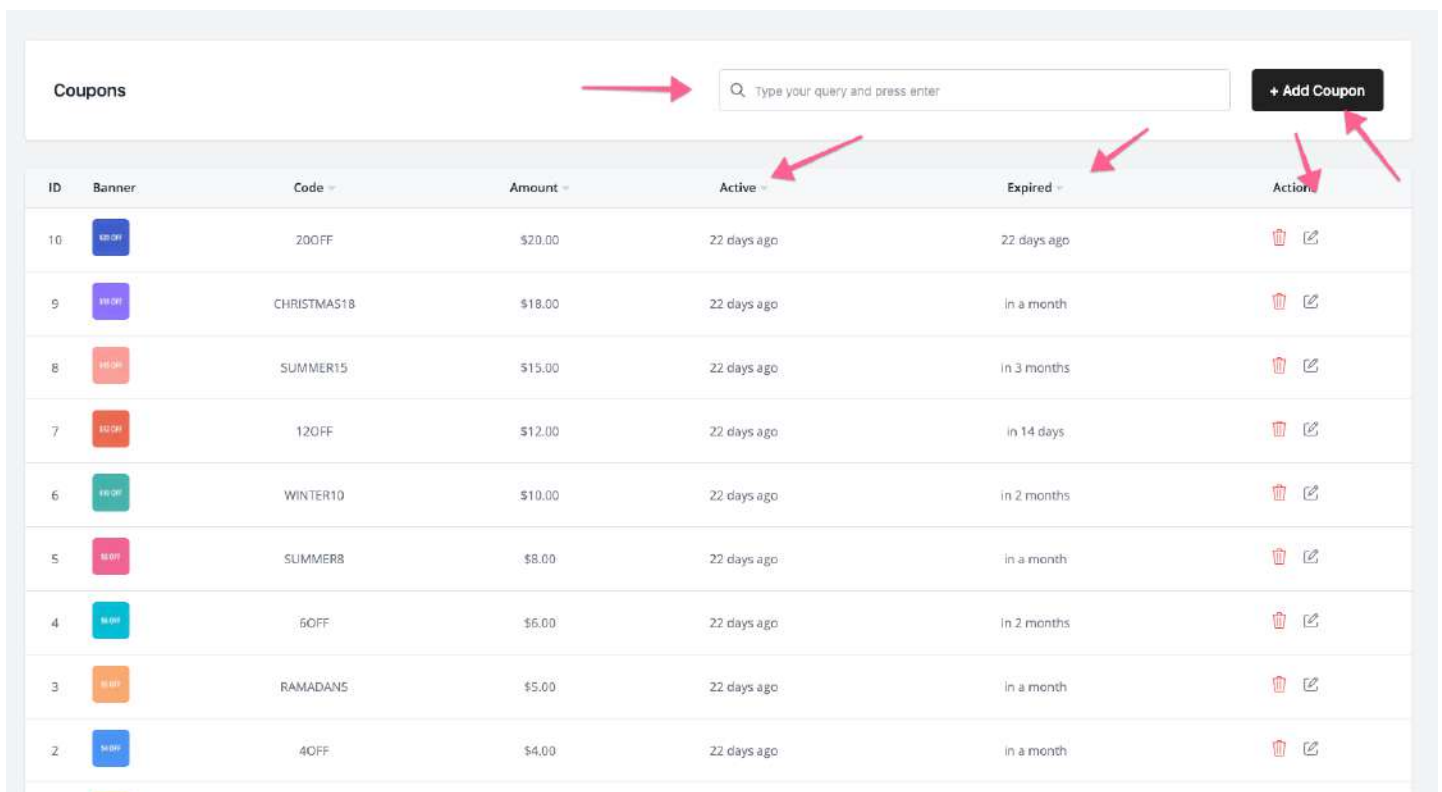
Change Status

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.



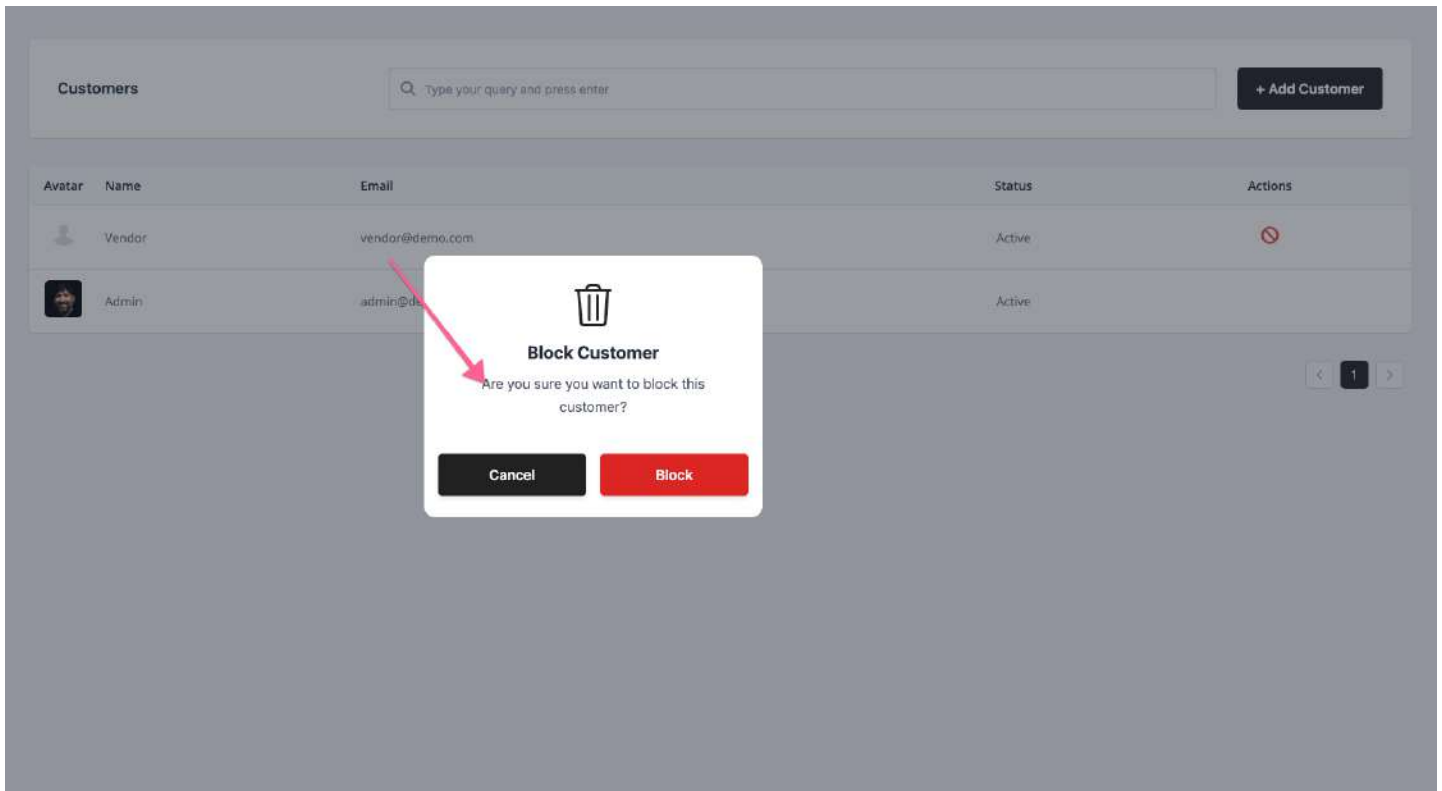
The screenshot shows a 'Coupons' management interface. At the top, there is a search bar with the placeholder text 'Q. Type your query and press enter' and a '+ Add Coupon' button. Below the search bar is a table with the following columns: ID, Banner, Code, Amount, Active, Expired, and Action. The table contains 10 rows of coupon data. Red arrows point to the search bar, the 'Active' column header, the 'Expired' column header, and the '+ Add Coupon' button.

ID	Banner	Code	Amount	Active	Expired	Action
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		120FF	\$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		60FF	\$6.00	22 days ago	in 2 months	 
3		RAMADANS	\$5.00	22 days ago	in a month	 
2		40FF	\$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.



Tax Management

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



ID	Name	Rate (%)	Country	City	State	ZIP	Actions
1	Global	2					

Shipping Management

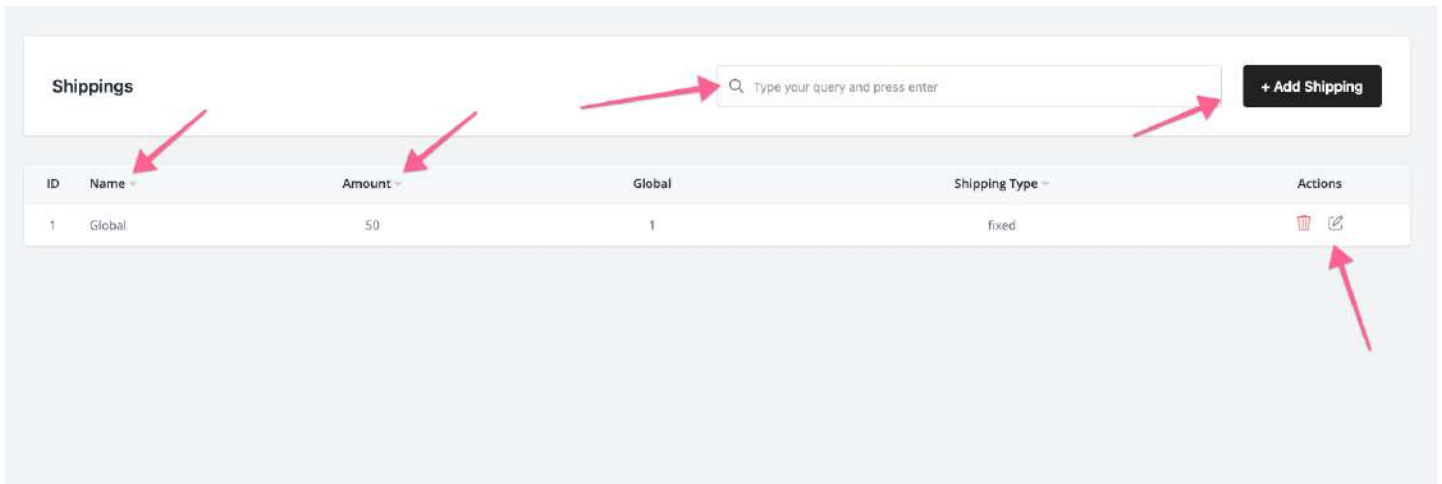
In **shipping** menu you will get the shipping list and you can add, remove or modify shipping from there.

ID	Name	Amount	Global	Shipping Type	Actions
1	Global	50	1	fixed	

Shipping Management

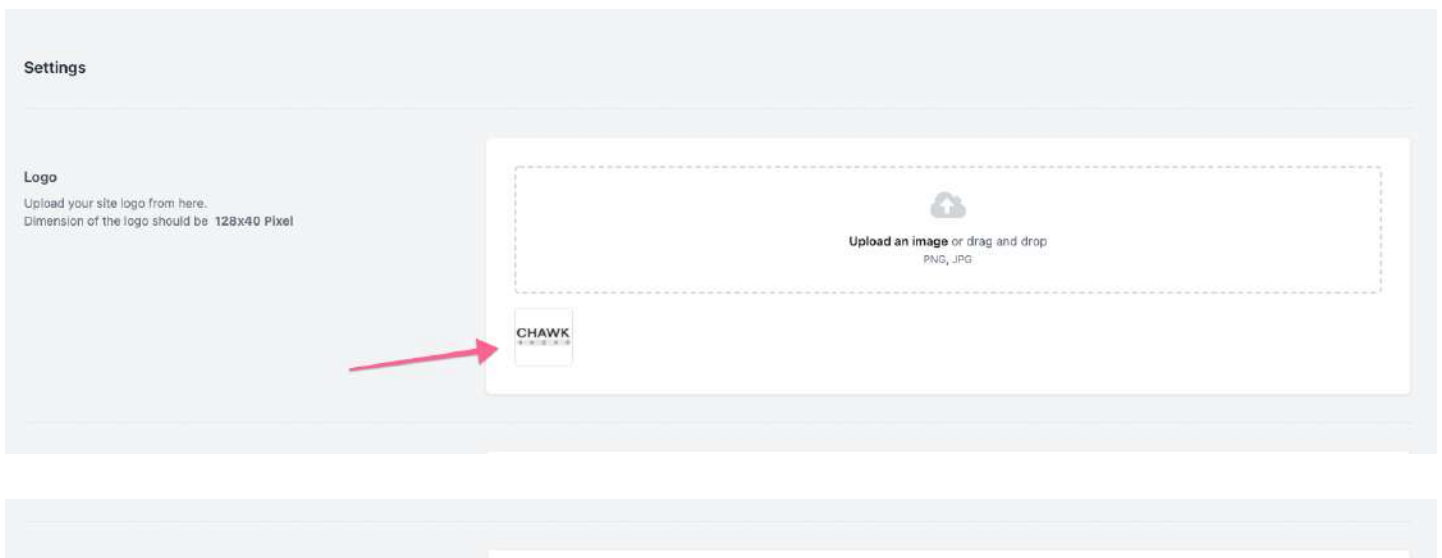
In **shipping** menu you will get the shipping list

and you can add, remove or modify shipping from there.



Settings Management

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



Information
Change your site information from here

Site Title

Chawkbazar

Site Subtitle

Your next ecommerce

Currency

US Dollar

Minimum Order Amount

0

Tax Class

Global

Shipping Class

Global

Available Scripts:

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

```
" 7 " " | : 6 \ "
```

```
"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

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 its 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.

- Open your custom SVG icon file in the code editor and copy all the code.
- Then Go to src → components → icons folder and create a new .tsx file.
- Then here create a function component and paste your copied SVG code inside the return statement.
- 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.)
- If your custom SVG code has any single custom color then change them into fillColor.

MENU

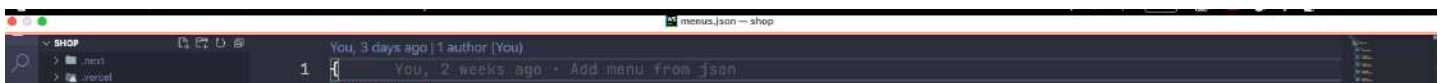
“

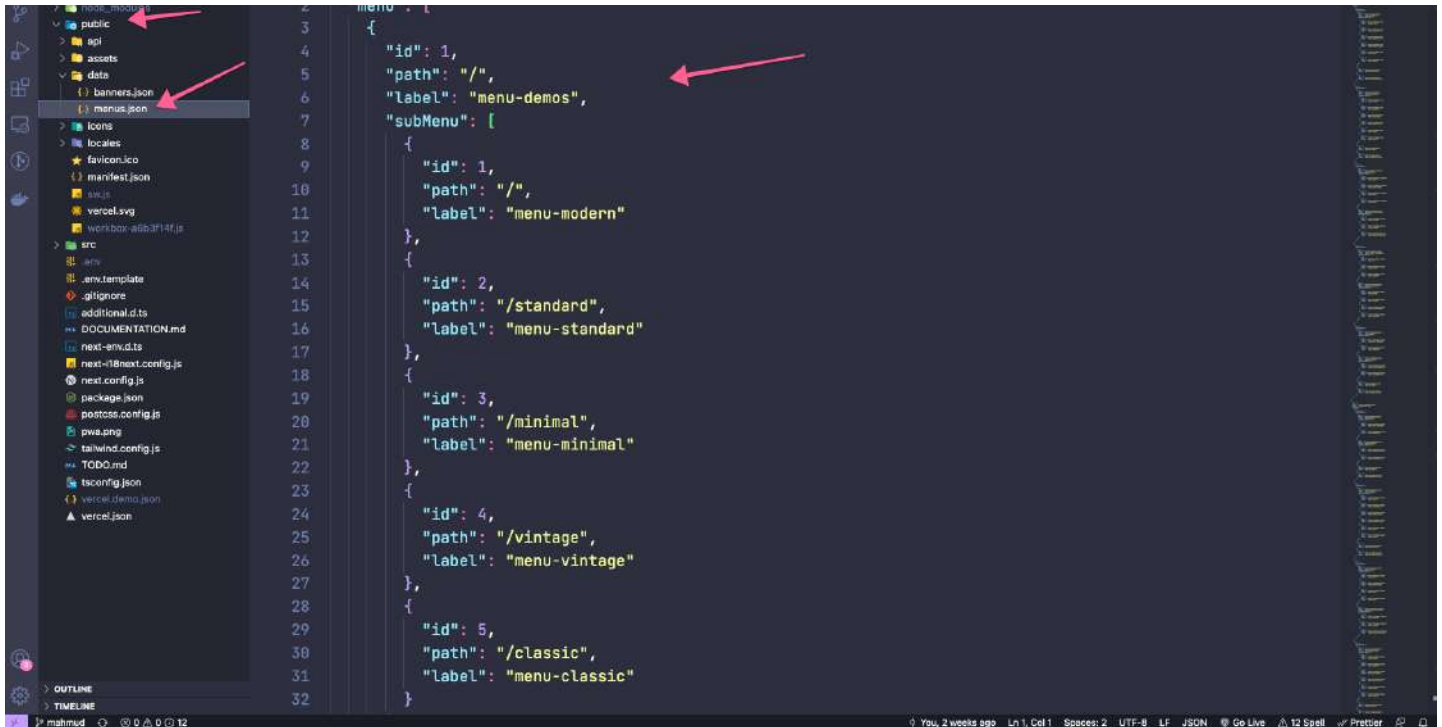
Please follow this video with the documentation, and it'll make the menu build process relatively easy.

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

Banner

“

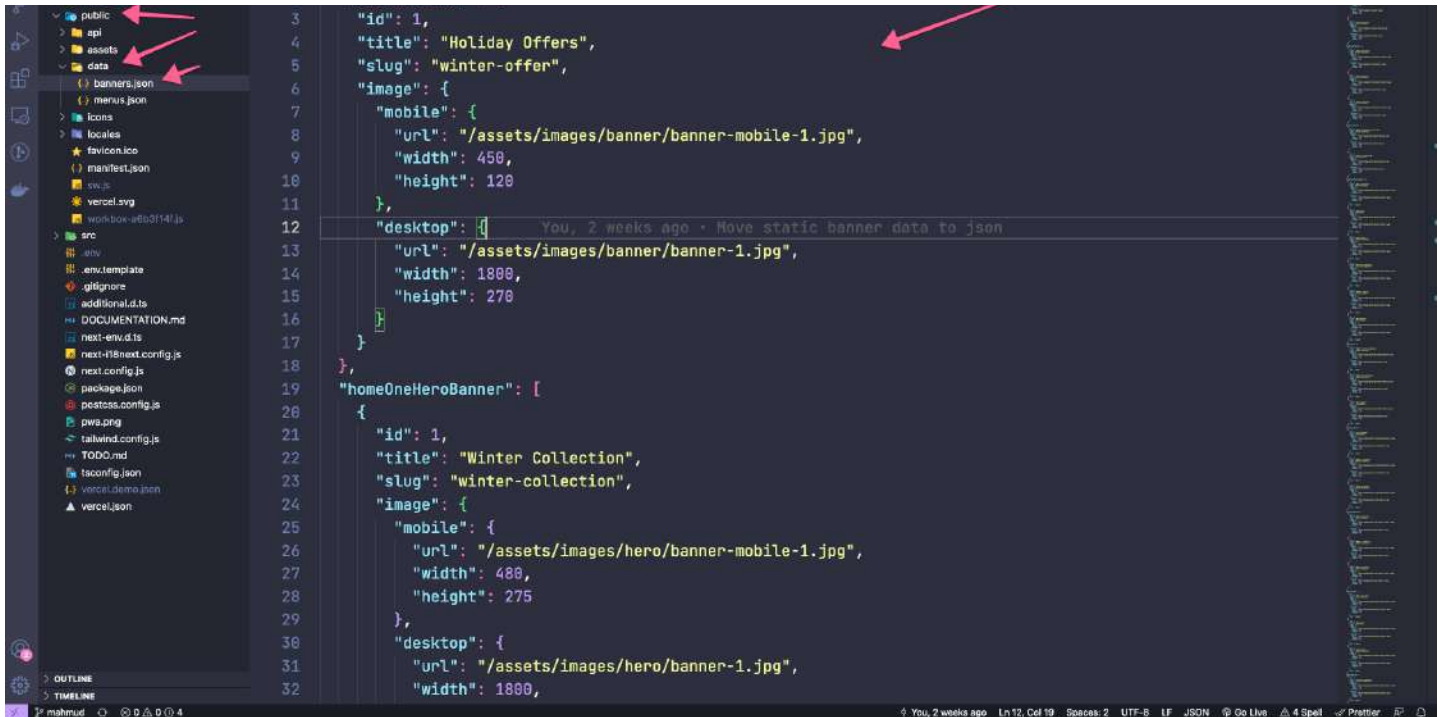
Please follow this video with the documentation, and it'll make the banner update process relatively easy.

<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 doc and add that doc

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

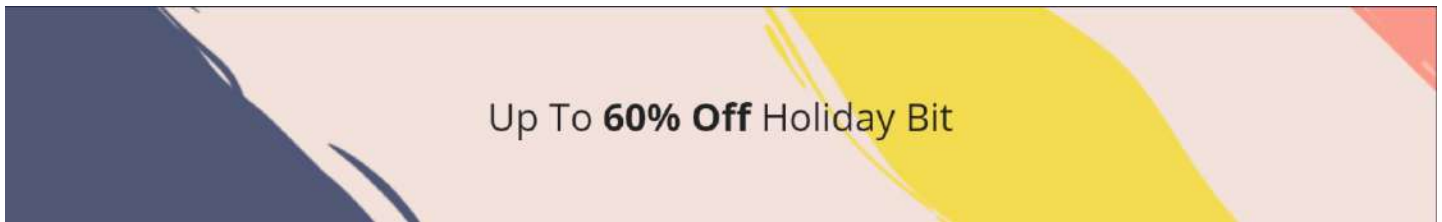
optimization

height -> Height of the image for next/image

optimization

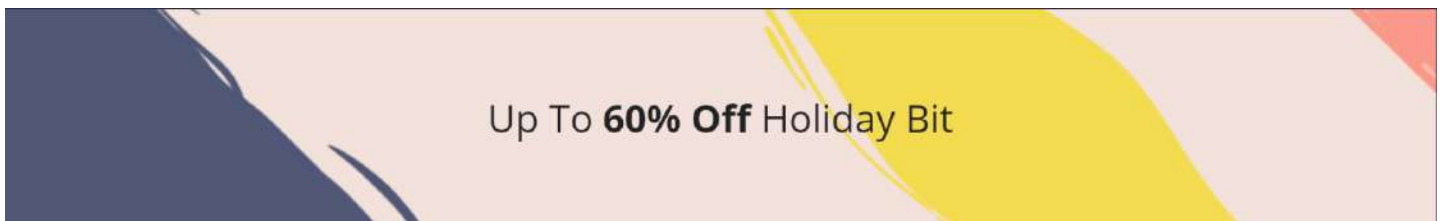
• Data Mapping

- homeOneBanner

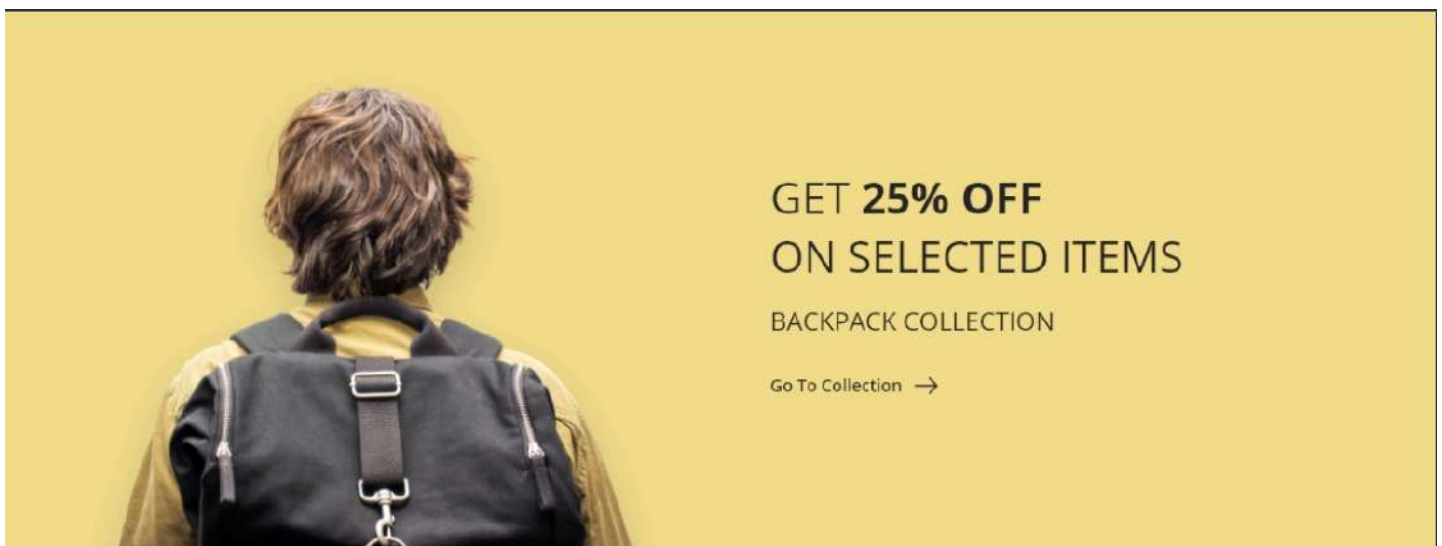


- homeOneHeroBanner

[1]



[2]



[3]

UP TO **50% OFF**

FOR ALL WINTER COLLECTION OF **KID'S ITEMS**

SHOP NOW



- homeTwoHeroBanner

[1]

Up To **60% Off** Holiday Bit

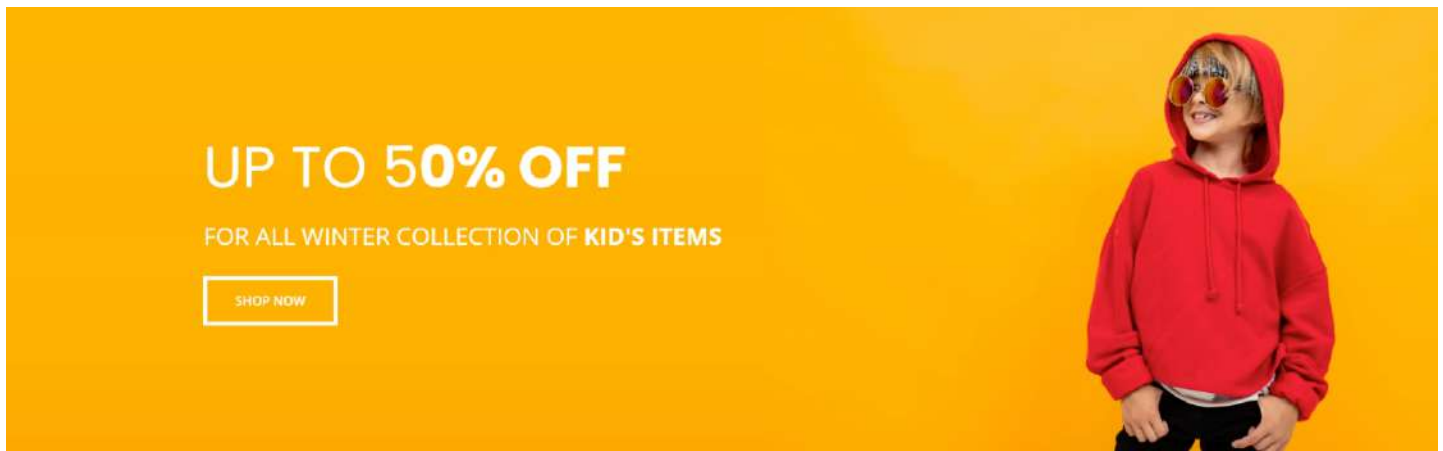
[2]

GET **25% OFF**
ON SELECTED ITEMS

BACKPACK COLLECTION

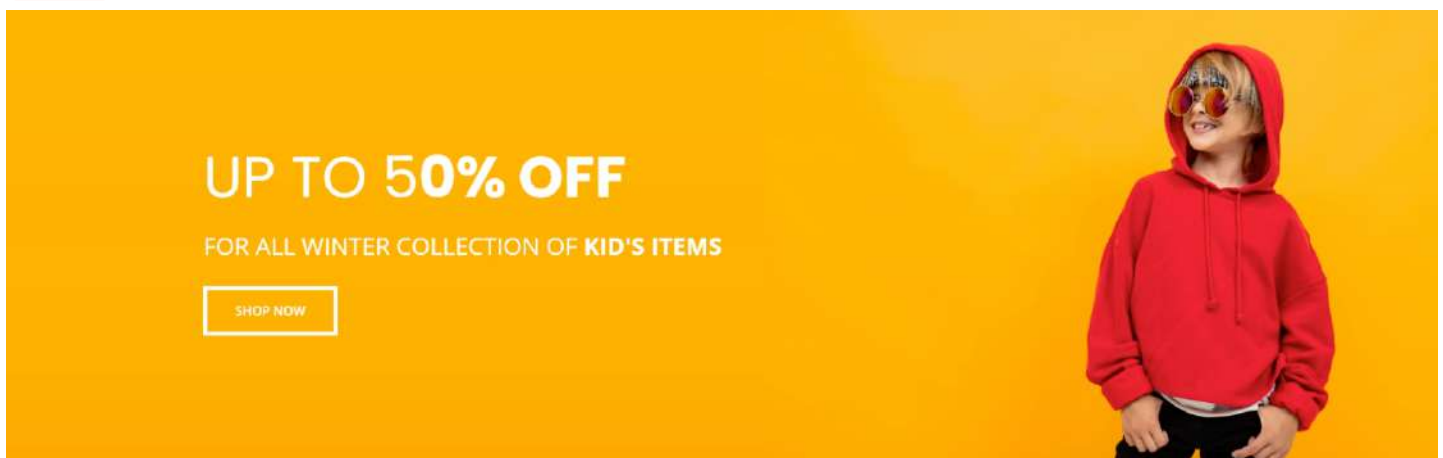
Go To Collection →

[3]



- homeThreeBanner

[1]



[2]



- homeThreeMasonryBanner

[1]

Up To **60% Off** Holiday Bit

[2]

GET **25% OFF**
ON SELECTED ITEMS

BACKPACK COLLECTION

[Go To Collection](#) →

[3]

UP TO 50% OFF

FOR ALL WINTER COLLECTION OF **KID'S ITEMS**

[SHOP NOW](#)

[4]



20%OFF EVERYTHING

USE CODE : OREO

[5]



explore

MEN'S
COLLECTION

#NEWYEAR2021

- bannerGrid

[1]



Up To **60% Off** Holiday Bit

[2]



[3]



- promotionBanner

[1]



[2]



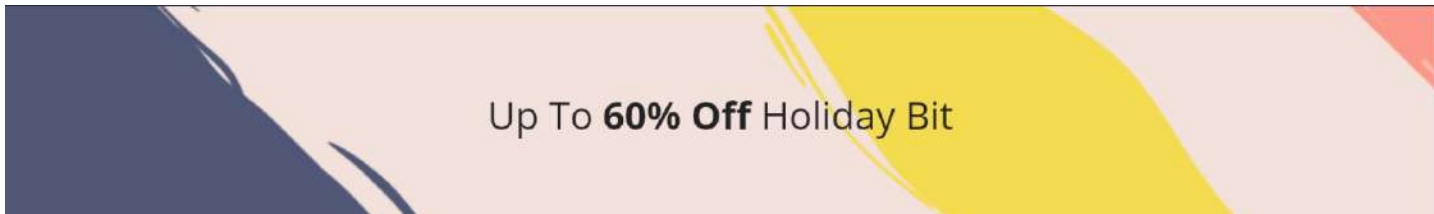


[3]



- promotionBannerTwo

[1]

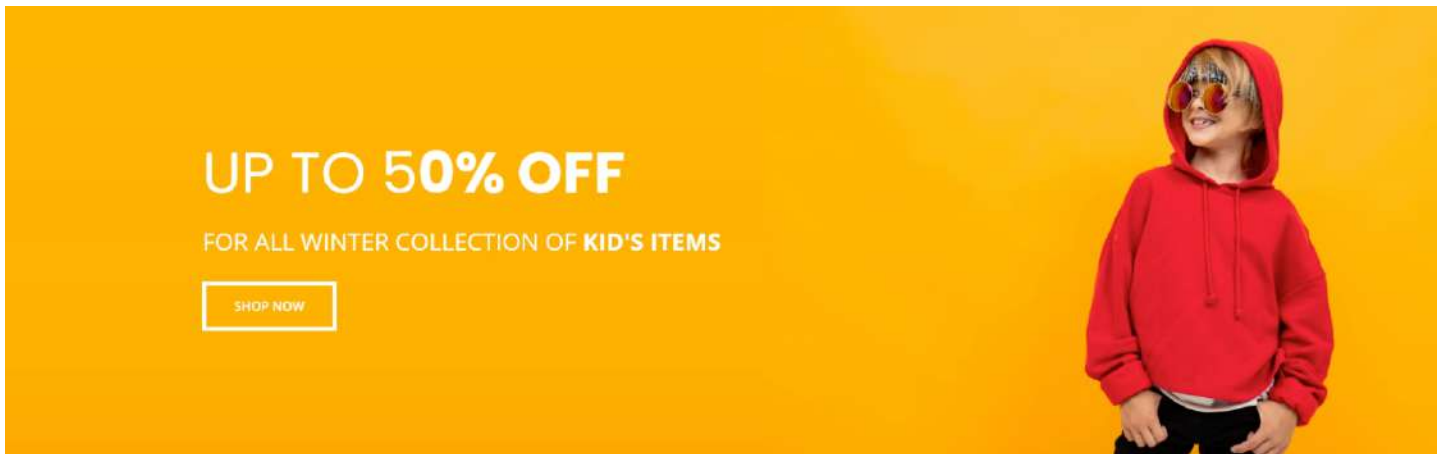


[2]



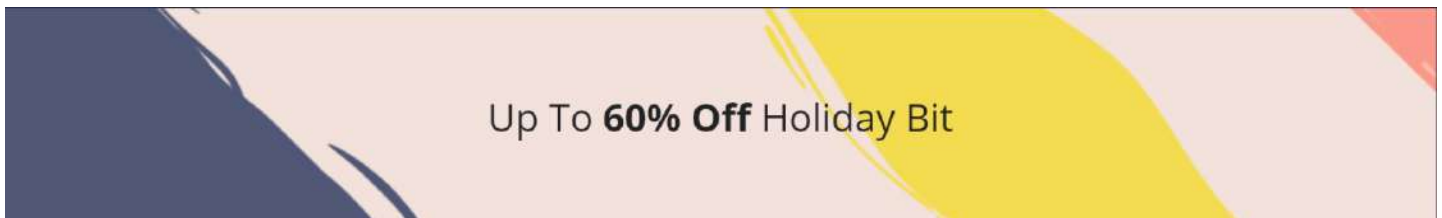


[3]



- bannerData0ne

[1]



[2]





[3]



- bannerDataTwo

[1]



[2]





[3]



- homeThreeProductsBanner

[1]



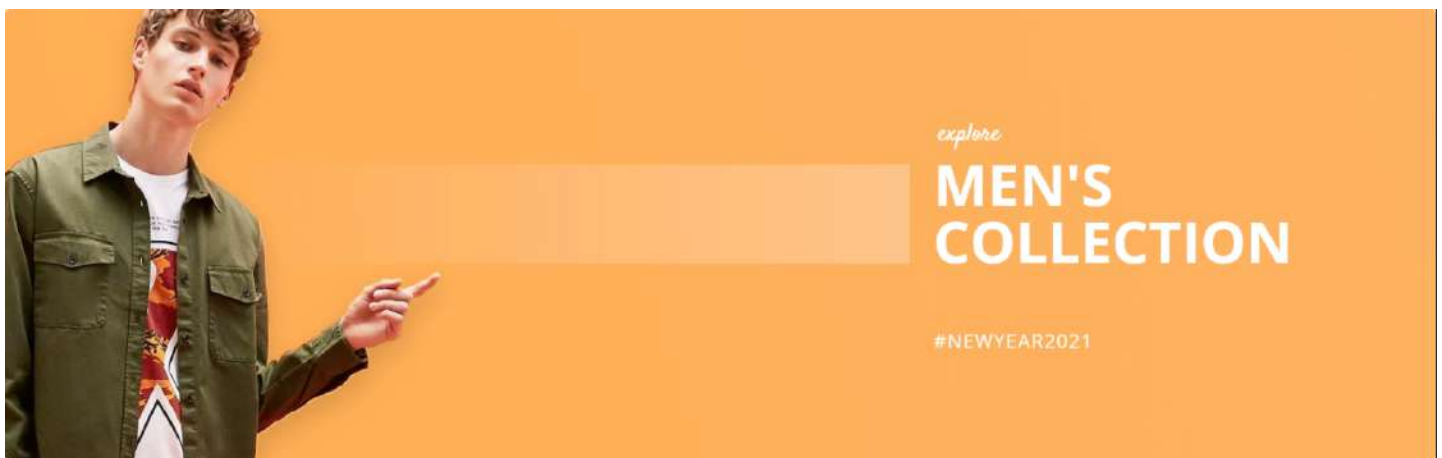




- homeFourBanner



[2]



[3]



- homeFourGridBanners

[1]

Up To **60% Off** Holiday Bit

[2]

GET **25% OFF**
ON SELECTED ITEMS

BACKPACK COLLECTION

[Go To Collection](#) →

[3]

UP TO **50% OFF**

FOR ALL WINTER COLLECTION OF **KID'S ITEMS**

[SHOP NOW](#)



- homeFiveBanner

[1]



Brand

We have multiple types of brand layouts.

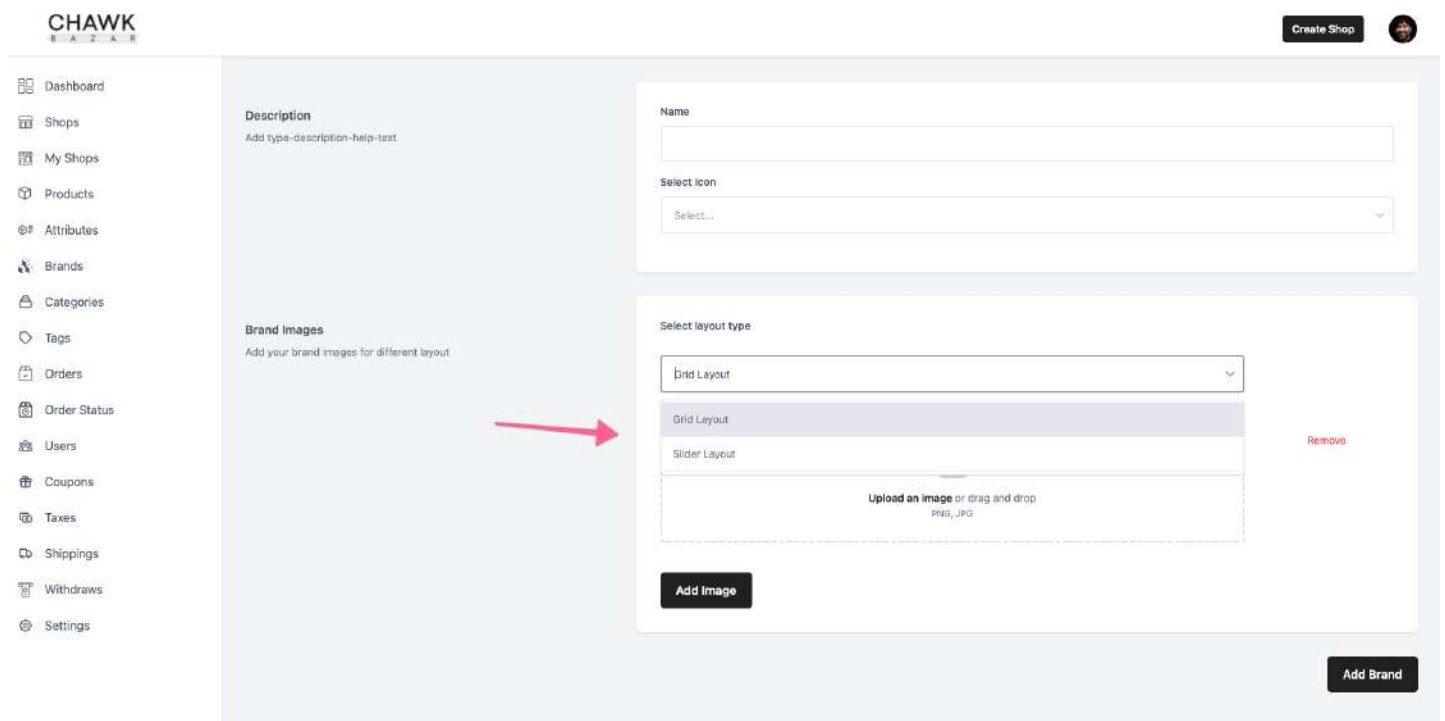
- Slider Layout → For Minimal Demo
- Grid Layout → For Others Demo

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 RAZAR dashboard with a sidebar menu on the left containing: Dashboard, Shops, My Shops, Products, Attributes, Brands, Categories, Tags, Orders, Order Status, Users, Coupons, Taxes, Shippings, Withdraws, and Settings. The main content area is divided into two sections: 'Description' and 'Brand Images'. The 'Description' section has a text input field with placeholder text 'Add type-description-help-text'. The 'Brand Images' section has a sub-section 'Select layout type' with a dropdown menu set to 'Grid Layout'. Below the dropdown are two options: 'Grid Layout' (highlighted) and 'Slider Layout'. To the right of these options is a 'Remove' link. Below the layout options is a dashed box for uploading an image, with the text 'Upload an image or drag and drop' and 'PNG, JPG'. There is an 'Add Image' button below the upload area. At the bottom right of the main content area is an 'Add Brand' button. A red arrow points from the 'Brand Images' section to the 'Grid 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/utils/
```

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

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)

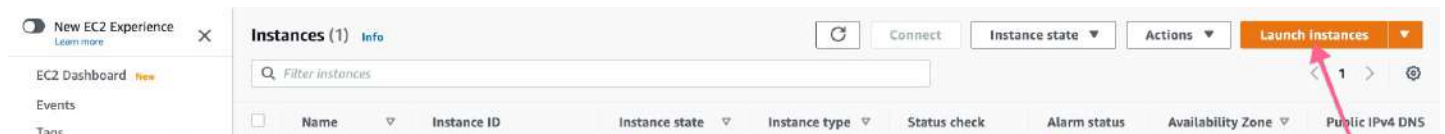
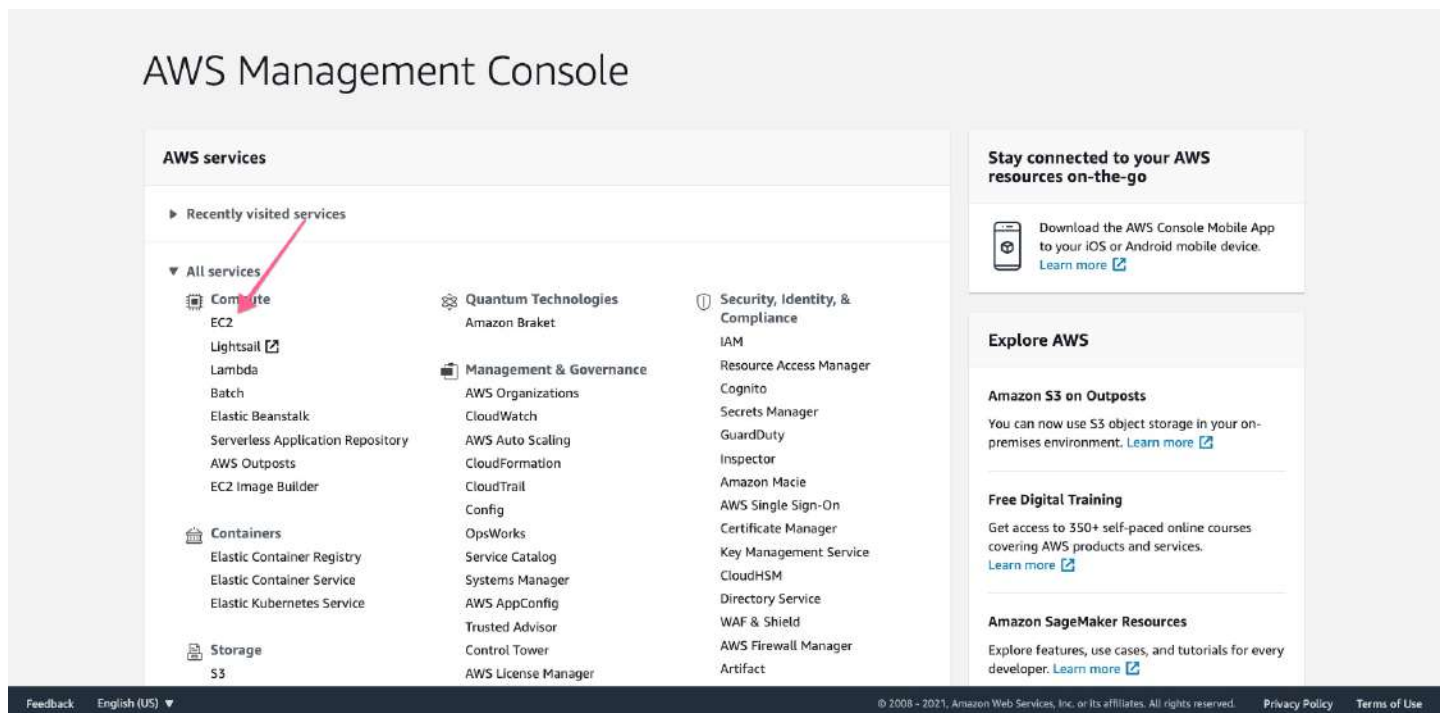
“

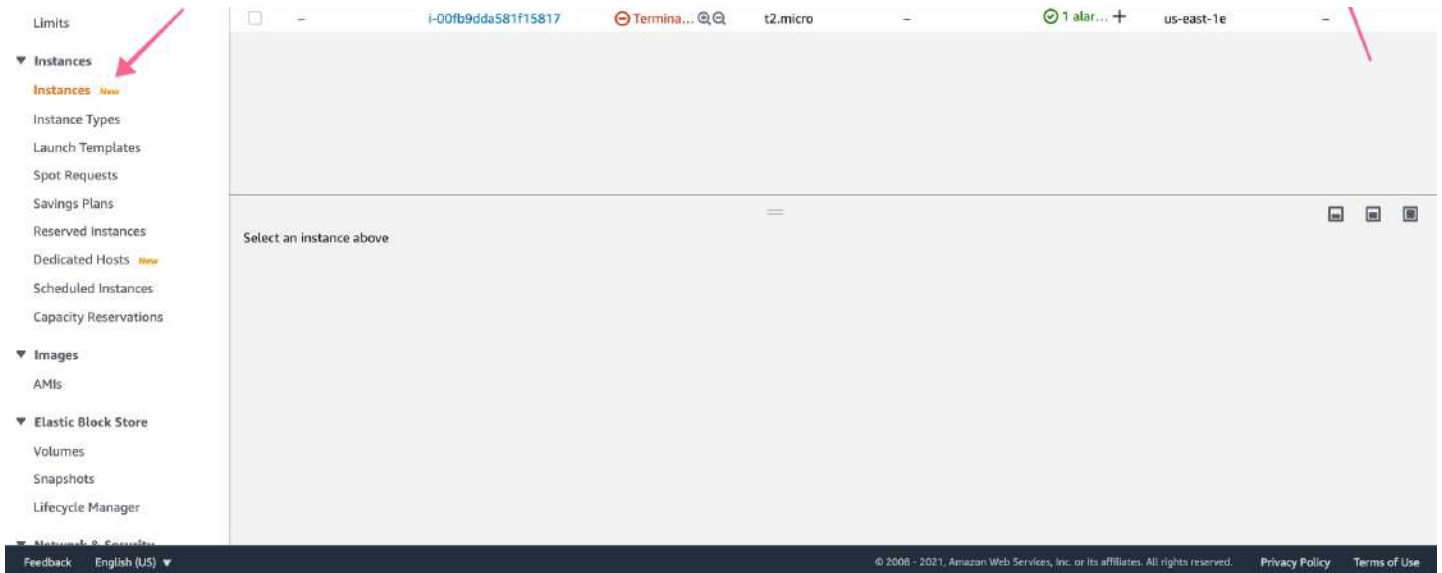
If you want to use all the scripts (`shop`, `admin`, `api`) on the same server as this tutorial, then we recommend creating a blank ubuntu-based server with at least 2+ CPU cores and 2GB+ memory.

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





Then select a ubuntu 20.04 server

After that, click **Next** → **Next** → **Next** → **Next**

And on security pages, add a rule for **HTTP**,

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

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

Add Rule

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous Review and Launch

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

After review, click **Launch**, and you'll get and nonun for KeyPair which will be required to

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.

Select an existing key pair or create a new key pair [X]

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 AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, 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 AMI](#).

Create a new key pair [v]

Key pair name

[Input field]

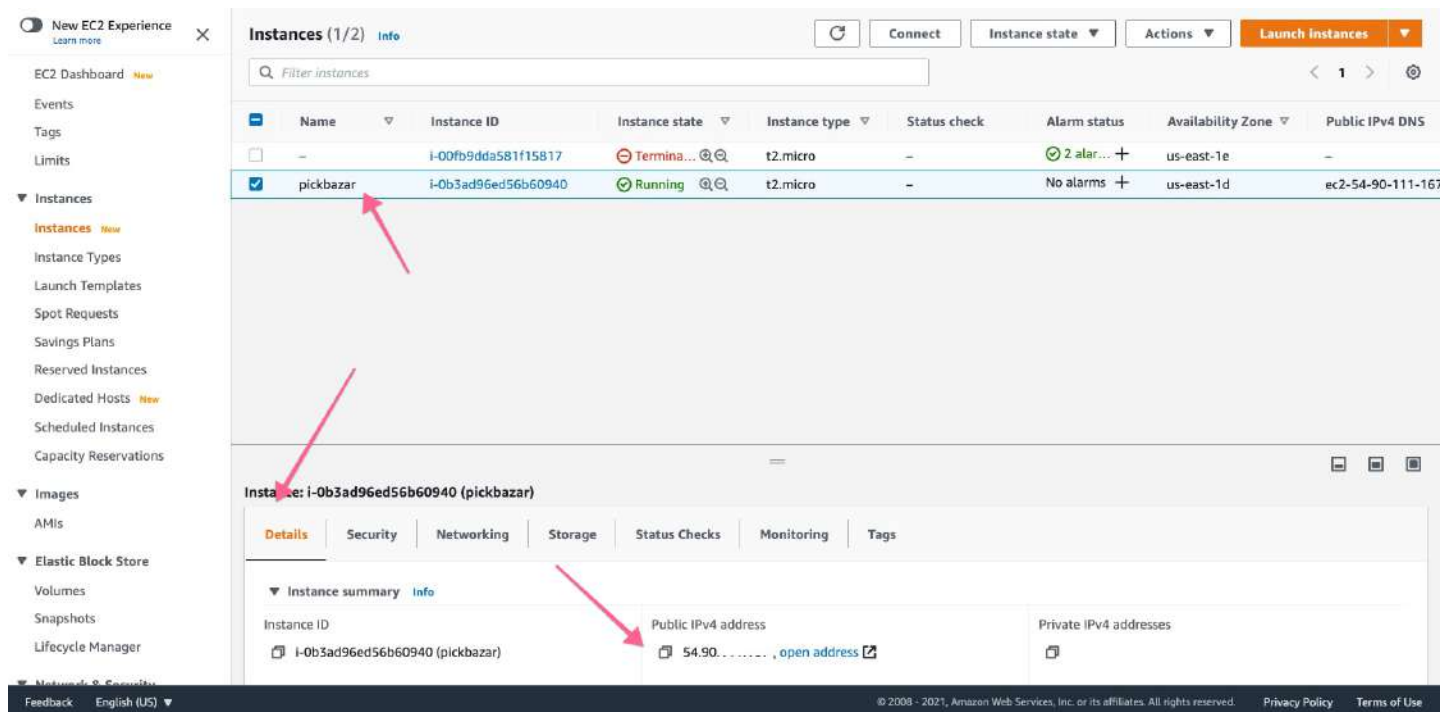
Download Key Pair

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

server IP, which will be required to login into ssh.



Domain Setup

Our server setup is done; now, we're going to set up the domain.

For domain and DNS, we're going to use

Cloudflare. It's a free DNS service that is

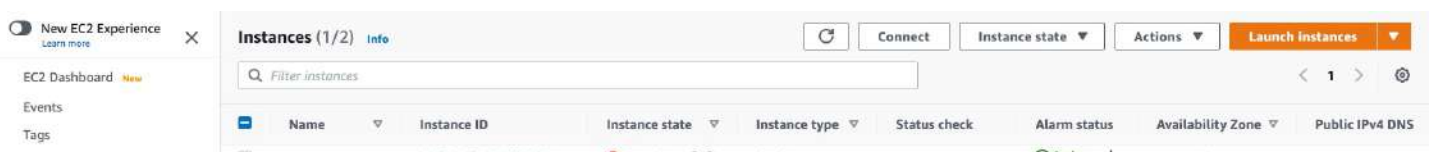
Cloudflare is a free DNS service that is used for DNS management and CDN. Not only that, it can be used to improve the security of your website as all the network come through your website by Cloudflare, and Cloudflare can prevent most of the security threats based on their traffic filtering.

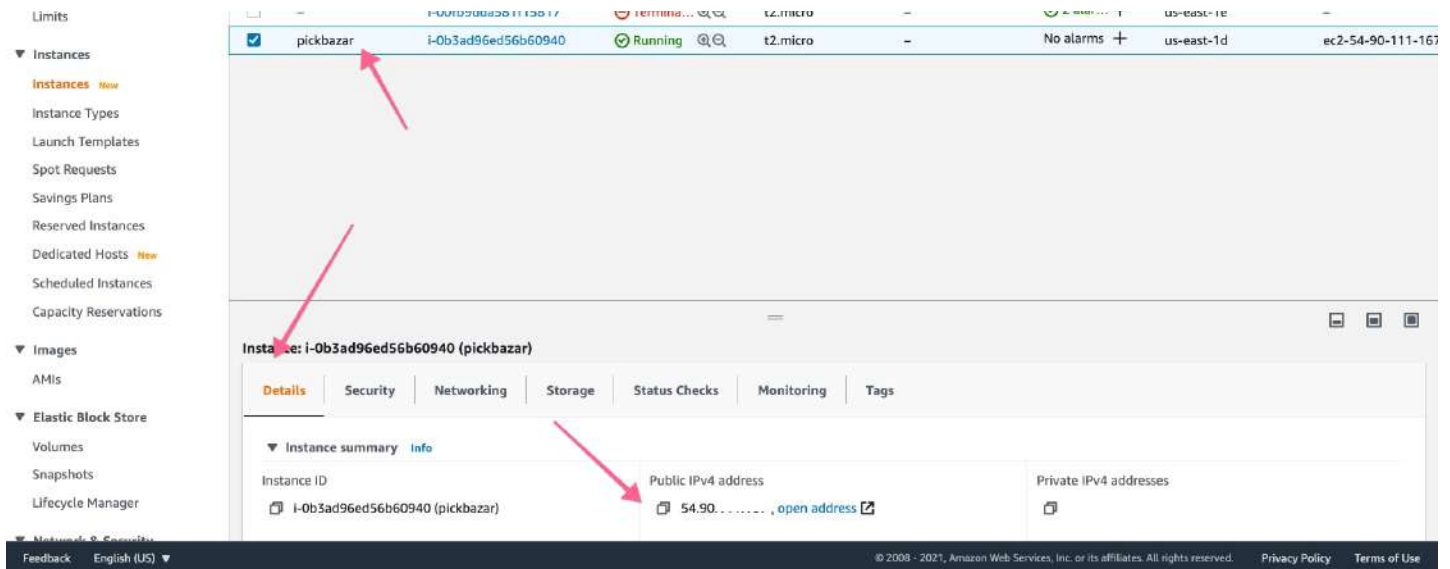
To use Cloudflare,

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

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

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





Add more DNS records for jsfinty.com

Proxy traffic for A, AAAA, and CNAME records by clicking the cloud icon.

Proxied: Accelerates and protects traffic

DNS resolution only: Bypasses Cloudflare

Note: Records with no cloud icon use DNS resolution but cannot be proxied.

DNS management for **jsfinty.com**

[+ Add record](#) [Advanced](#)

jsfinty.com points to **54.90.1.100** and has its traffic proxied through Cloudflare.

Type	Name	IPv4 address	TTL	Proxy status
A	@	54.90.1.100	Auto	Proxied

[Cancel](#) [Save](#)

[Continue](#)

Then add and another **CNAME** record for **www**

Proxy traffic for A, AAAA, and CNAME records by clicking the cloud icon.

Proxied: Accelerates and protects traffic

DNS resolution only: Bypasses Cloudflare

Note: Records with no cloud icon use DNS resolution but cannot be proxied.

DNS management for .com

+ Add record Search DNS Records Advanced

www. .com is an alias of .com and has its traffic proxied through Cloudflare.

Type	Name	Target	TTL	Proxy status
CN...	www	.com	Auto	Proxied

Cancel Save

Type	Name	Content	TTL	Proxy status
A	.com	54.90.	Auto	Proxied

Continue

We successfully create our server and domain.
Now we're going to set up the server to host
the **chawkbazar**.

Login to Server

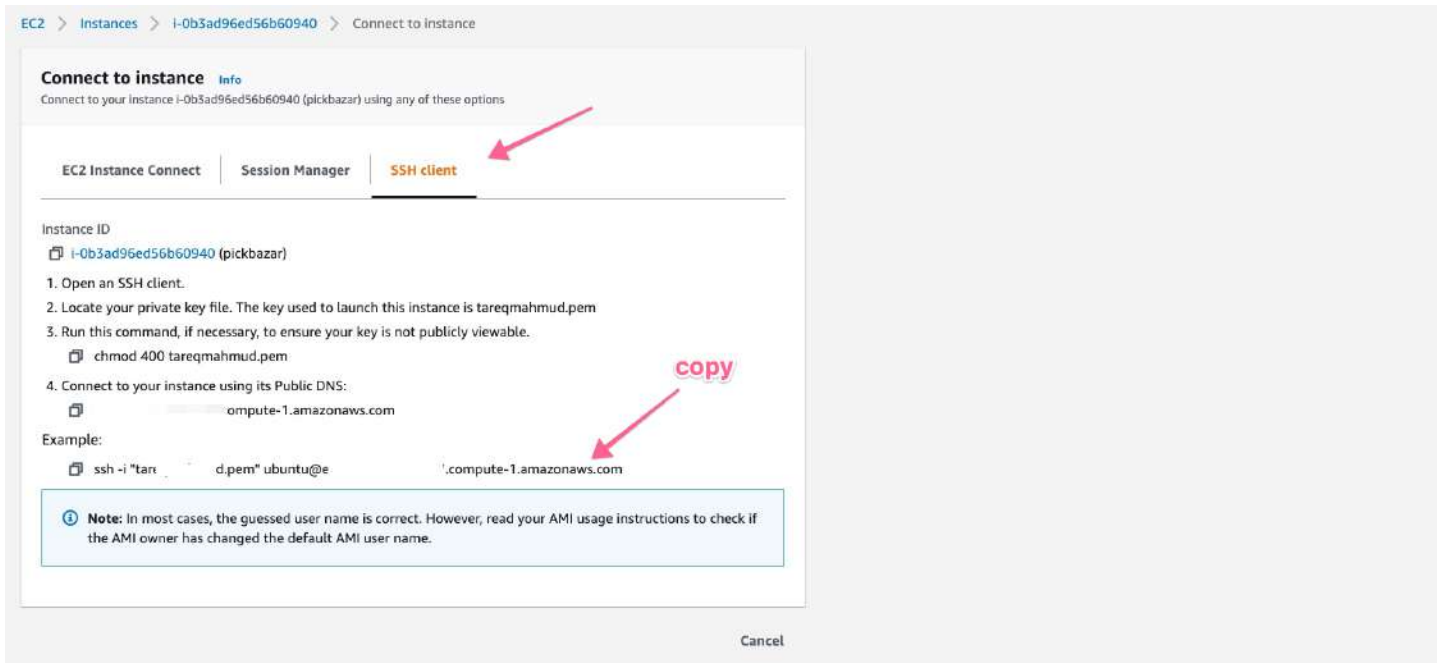
At first, login to your **AWS** server using ssh.
to do that, go to the folder from the terminal
where **KeyPair** is downloaded.

then click **Connect**



From the **Connect** dashboard, go to **SSH Client**
and copy the example line and paste it to your

terminal.



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

• Change permission .pem

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

```
chmod 400 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

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

| 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 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`

Extract the `chawkbazar` package that you download from `CodeCanyon`.

On that folder, you'll get another `folder` called `chawkbazar-laravel`.

On that folder, you'll get a folder called `chawkbazar-api`

Now upload this `chawkbazar-api` folder to the

now upload this `chawkbazar.apk` 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;
    }

    location = /favicon.ico { access_log off;
log_not_found off; }

    location = /robots.txt { access_log off;
log_not_found off; }

    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

... then, 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=local
3 APP_KEY=base64:NTLdF5T0UznjFSE/mbp5wEBRt0MsevSKmtIowxah3yA=
4 APP_DEBUG=true
5 APP_URL=https://YOUR_DOMAIN.COM/api
6 APP_SERVICE=chawkbazar.test
```



```
7 APP_NOTICE_DOMAIN=CHAWKBAZAR_  
8 DUMMY_DATA_PATH=chawkbazar  
9  
10  
11 LOG_CHANNEL=stack  
12 LOG_LEVEL=debug  
13
```

Then install all the packages and install `api`

```
composer install
```

```
php artisan key:generate
```

```
php artisan marvel:install
```

“

You'll get several confirmations for migration, dummy data, and admin account.

Make sure you check the confirmation step and take the necessary actions based on your requirement.

Enable `laravel storage,`

```
php artisan storage:link
```

Then give proper **permission** for laravel folder,

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

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

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

... as suggest , we build the production 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.

Vercel

“

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

• vercel.com

If you want to host the template in vercel.com

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

• Frontend

- Step 1:

- After deploying the api you will get the api endpoint url. Put that url in the `[your-frontend-project]/.env` and `vercel.json` file.

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

- Step 2:

Run this command to `shop` for `frontend` and to `admin -> rest` for `admin`

```
vercel
```

```
vercel --prod
Vercel CLI 23.1.2
Inspect: https://vercel.com/tareqmahmud/chawkbazar-shop-dev/8828NrnrtPNKPedRxowT5dLnvv84 [6s]
Production: https://[redacted].vercel.app [copied to clipboard] [10m]
```

**** NOTE: **** for deploying to **vercel** you need to install **vercel-cli** on your machine for more information please visit [here](#)

- **For others hosting providers:**

Please follow [nextjs deployment docs](#):

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

email/username and password. To set up social login, follow the below procedure,

|Google

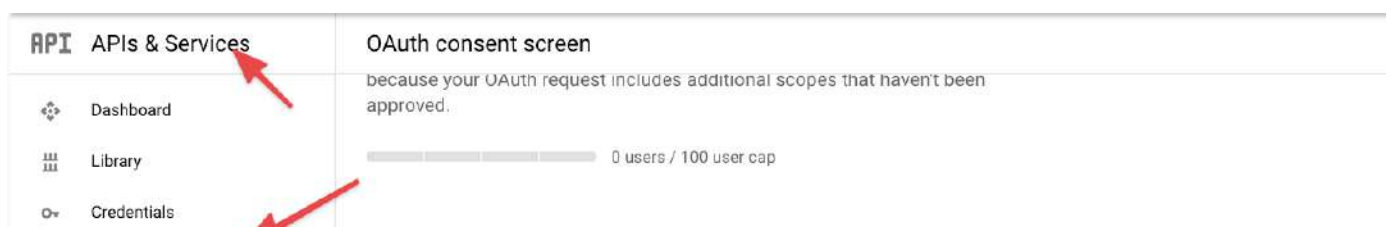
For Google, follow this procedure,

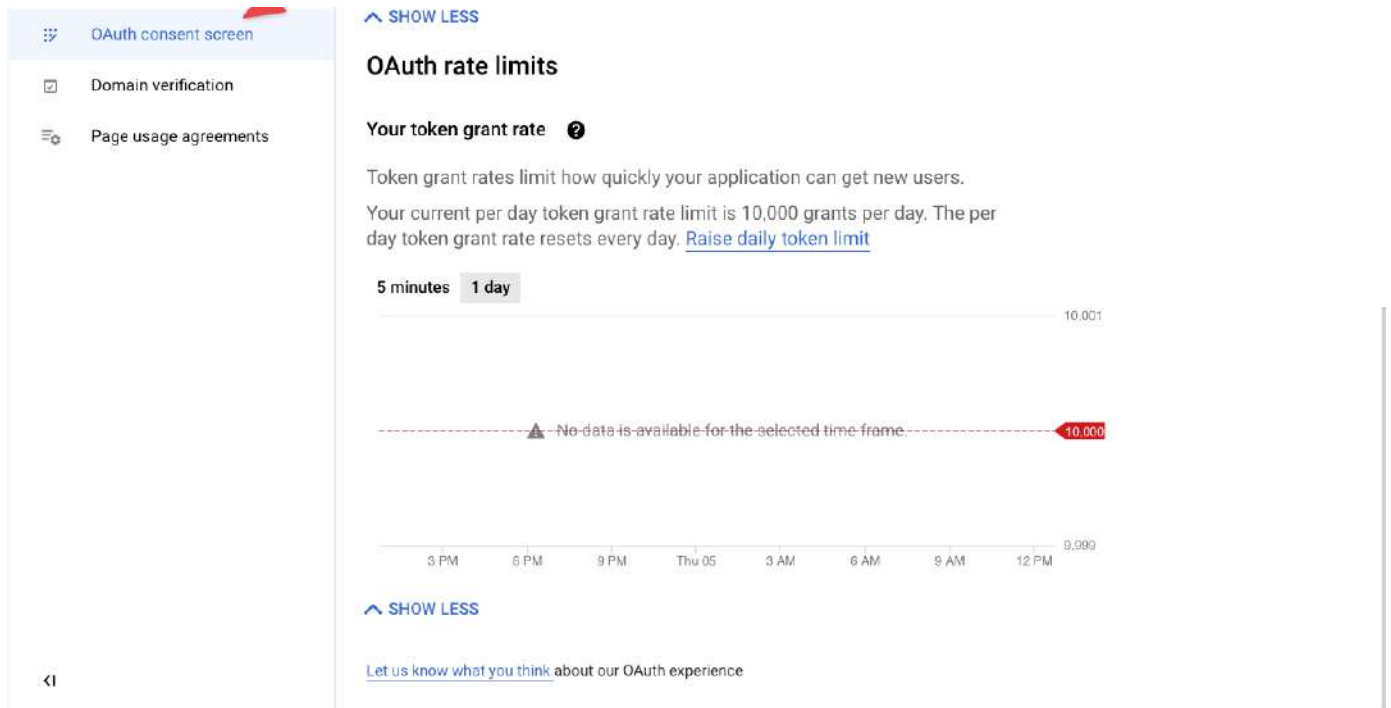
At first go to Google Console dashboard (<https://console.cloud.google.com/>)

From the console, create a new project

After creating the project, go to **APIs & Services**

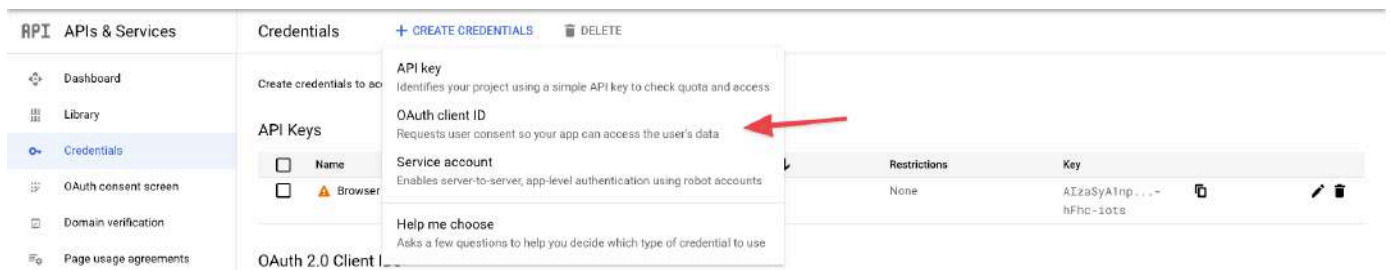
-> OAuth consent screen.





Create an **External OAuth Consent**.

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



Then you'll get your credentials,



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

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

Your Client ID

211044409169-1

Your Client Secret

WFqrABH3N

↓ DOWNLOAD JSON

OK

Client ID

2110444

2110444

ne ↑

Engine default service a

base-adminsdk

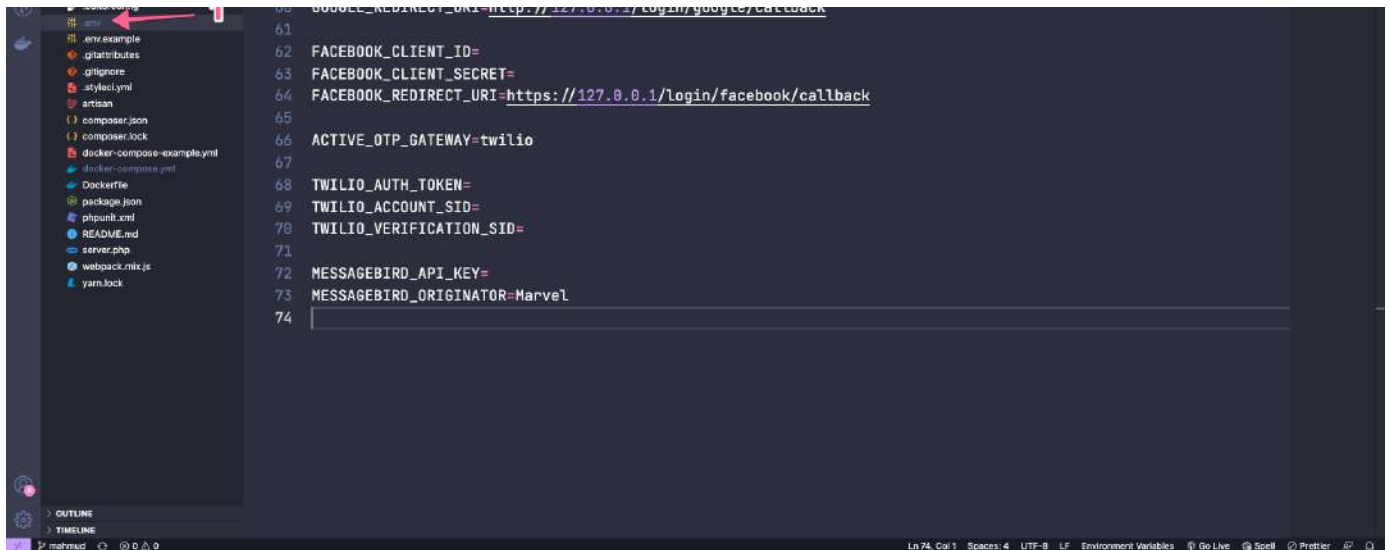
Then open `api/.env` and add API ID and credentials,



```
EXPLORER
CHAWKBAZAR-API
  app
  bootstrap
  config
  database
  packages
  public
  resources
  routes
  storage
  tests
  vendor
  .env

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_REAUTH_URL=http://127.0.0.1/login/google/callback
```

The screenshot shows a code editor with the `.env` file open. The file contains several environment variables. Red arrows and numbers point to the `GOOGLE_CLIENT_ID` and `GOOGLE_CLIENT_SECRET` lines, indicating where the credentials from the previous image should be pasted. The `GOOGLE_CLIENT_ID` line is labeled with a red '2' and the `GOOGLE_CLIENT_SECRET` line is labeled with a red '3'.

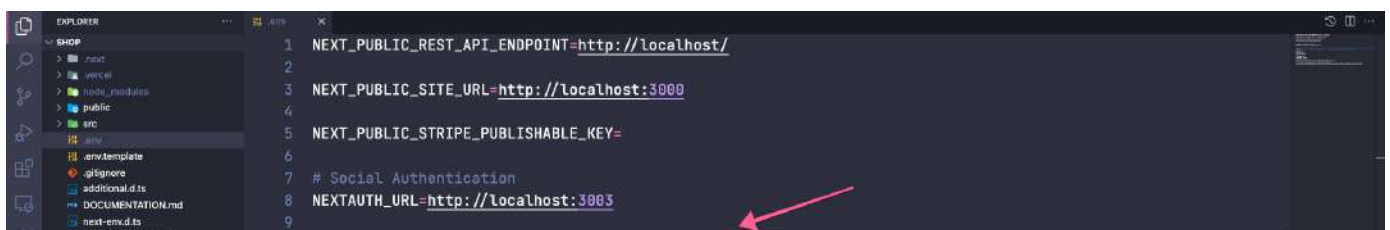


A screenshot of a code editor with a dark theme. On the left, a file explorer shows a project structure with files like .env.example, .gitattributes, .gitignore, .stylelintrc, artisan, composer.json, composer.lock, docker-compose-example.yml, docker-compose.yml, Dockerfile, package.json, phpunit.xml, README.md, server.php, webpack.mix.js, and yarn.lock. A red arrow points to the .env file. The main editor area shows the contents of the .env file, which includes various environment variables for social media and authentication services. The variables are as follows:

```
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
```

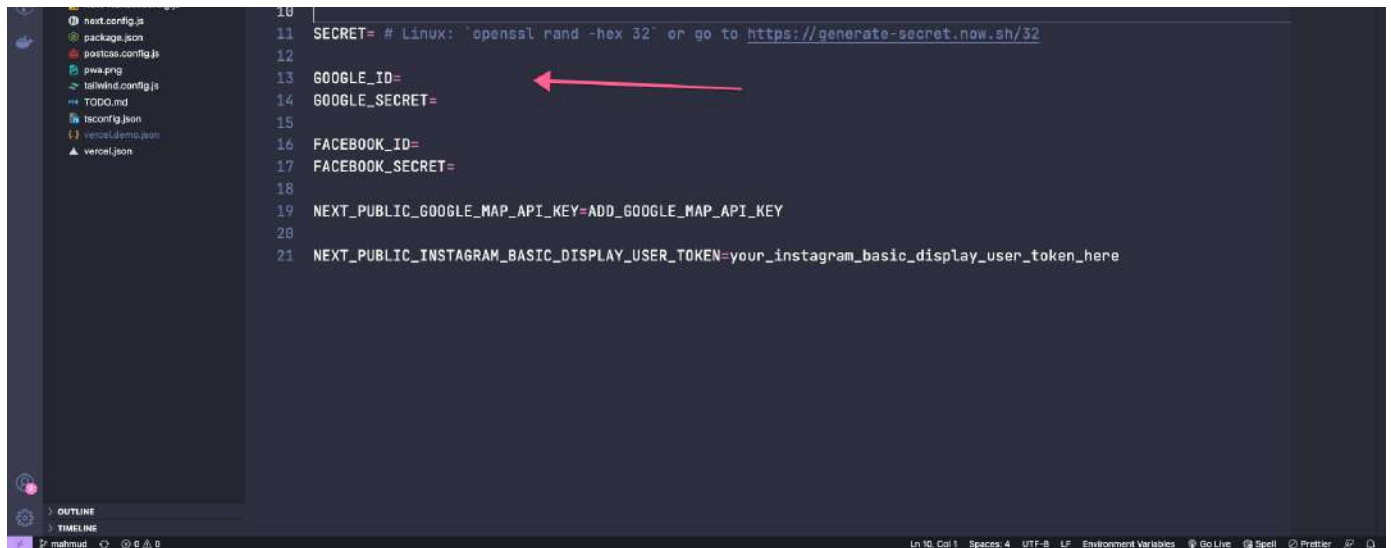
Then go to this <https://generate-secret.now.sh/32> site and generate a secret key.

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



A screenshot of a code editor with a dark theme. On the left, a file explorer shows a project structure with files like .env, .env.example, .gitattributes, .gitignore, .stylelintrc, artisan, composer.json, composer.lock, docker-compose-example.yml, docker-compose.yml, Dockerfile, package.json, phpunit.xml, README.md, server.php, webpack.mix.js, and yarn.lock. A red arrow points to the .env file. The main editor area shows the contents of the .env file, which includes various environment variables for social media and authentication services. The variables are as follows:

```
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
```

A screenshot of a code editor with a dark theme. On the left, a file explorer shows a project structure with files like 'next.config.js', 'package.json', 'postcss.config.js', 'pwa.png', 'tailwind.config.js', 'TODO.md', 'tsconfig.json', 'vercelDemo.json', and 'vercel.json'. The main editor area displays configuration code for environment variables. Line 11: 'SECRET= # Linux: `openssl rand -hex 32` or go to <https://generate-secret.now.sh/32>'. Line 13: 'GOOGLE_ID=' with a red arrow pointing to it. Line 14: 'GOOGLE_SECRET='. Line 16: 'FACEBOOK_ID='. Line 17: 'FACEBOOK_SECRET='. Line 19: 'NEXT_PUBLIC_GOOGLE_MAP_API_KEY=ADD_GOOGLE_MAP_API_KEY'. Line 21: 'NEXT_PUBLIC_INSTAGRAM_BASIC_DISPLAY_USER_TOKEN=your_instagram_basic_display_user_token_here'. The status bar at the bottom shows 'Ln 10, Col 1', 'Spaces: 4', 'UTF-8', 'LF', 'Environment Variables', 'Go Live', 'Spell', 'Prettier', and a file icon.

```
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
```

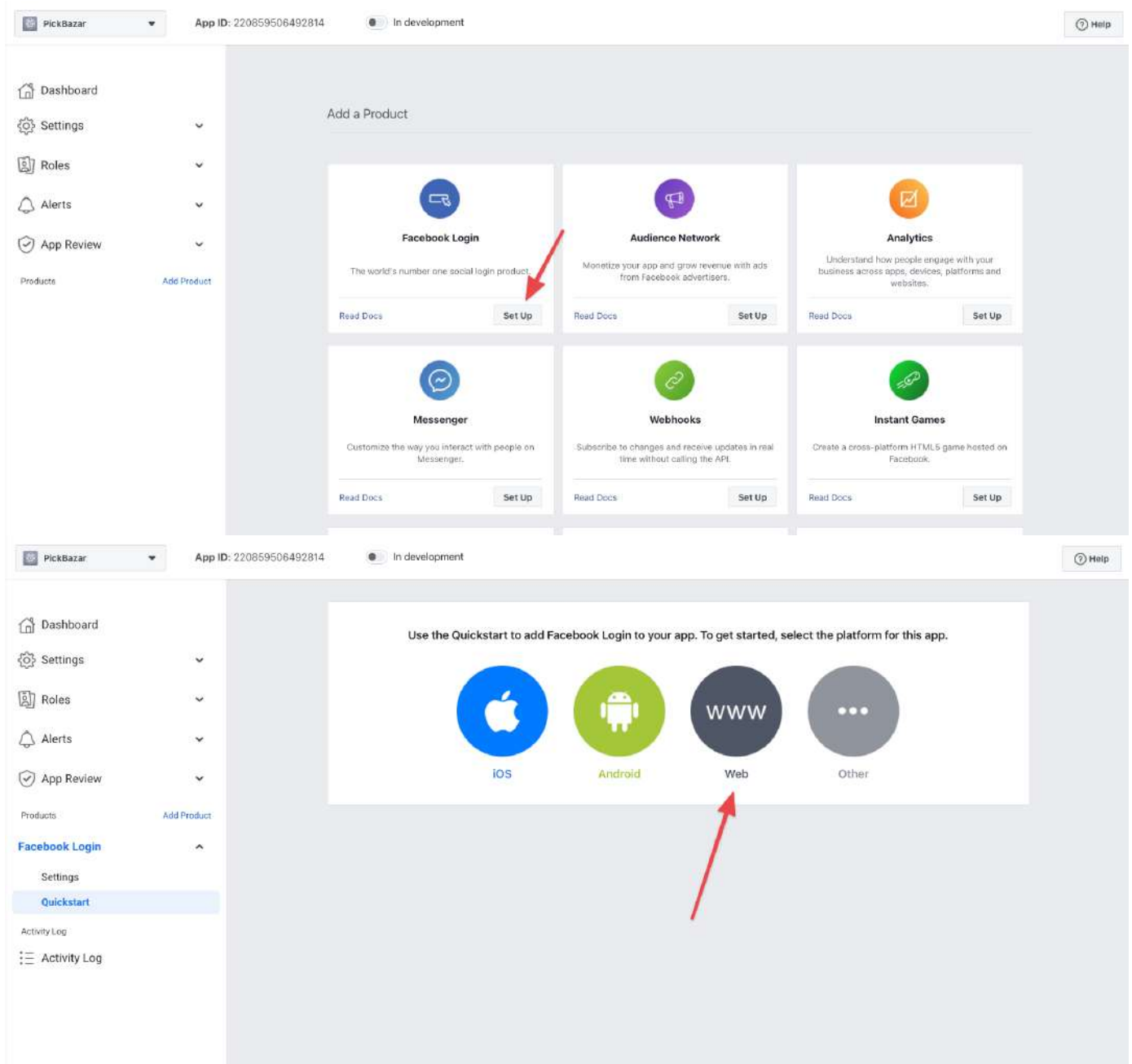
| Facebook

Go to Facebook developer dashboard (<https://developers.facebook.com/apps/>)

Then create a new app for chawkhazar

then create a new app for `android`.

After that, set up **Facebook Login**.



Then provide all the necessary information.

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

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

Facebook App

The image shows the Facebook App Settings page for an application named 'PickBazar'. The App ID is 220859506492814. The App Secret is masked with dots and a red arrow points to the 'Show' button. The Display Name is 'PickBazar'. The App Domains field is empty. The Privacy Policy URL is 'Privacy policy for Login dialog and App Details'. The App Icon (1024 x 1024) is a placeholder image. The App Purpose is 'Yourself or your own business'. The Contact Email field is empty. The Terms of Service URL is 'Terms of Service for Login dialog and App Details'. The Category is 'Choose a Category'. The 'Save Changes' button is highlighted in blue.

Dashboard

Settings

Basic

Advanced

Roles

Alerts

App Review

Products

Facebook Login

Activity Log

App ID: 220859506492814

In development

App ID: 220859506492814

App Secret: [Masked] Show

Display Name: PickBazar

Namespace:

App Domains:

Contact Email: [Empty]

Privacy Policy URL: Privacy policy for Login dialog and App Details

Terms of Service URL: Terms of Service for Login dialog and App Details

App Icon (1024 x 1024): [Placeholder]

App Purpose: Yourself or your own business

Category: Choose a Category

Find out more information about app categories here

Discard Save Changes

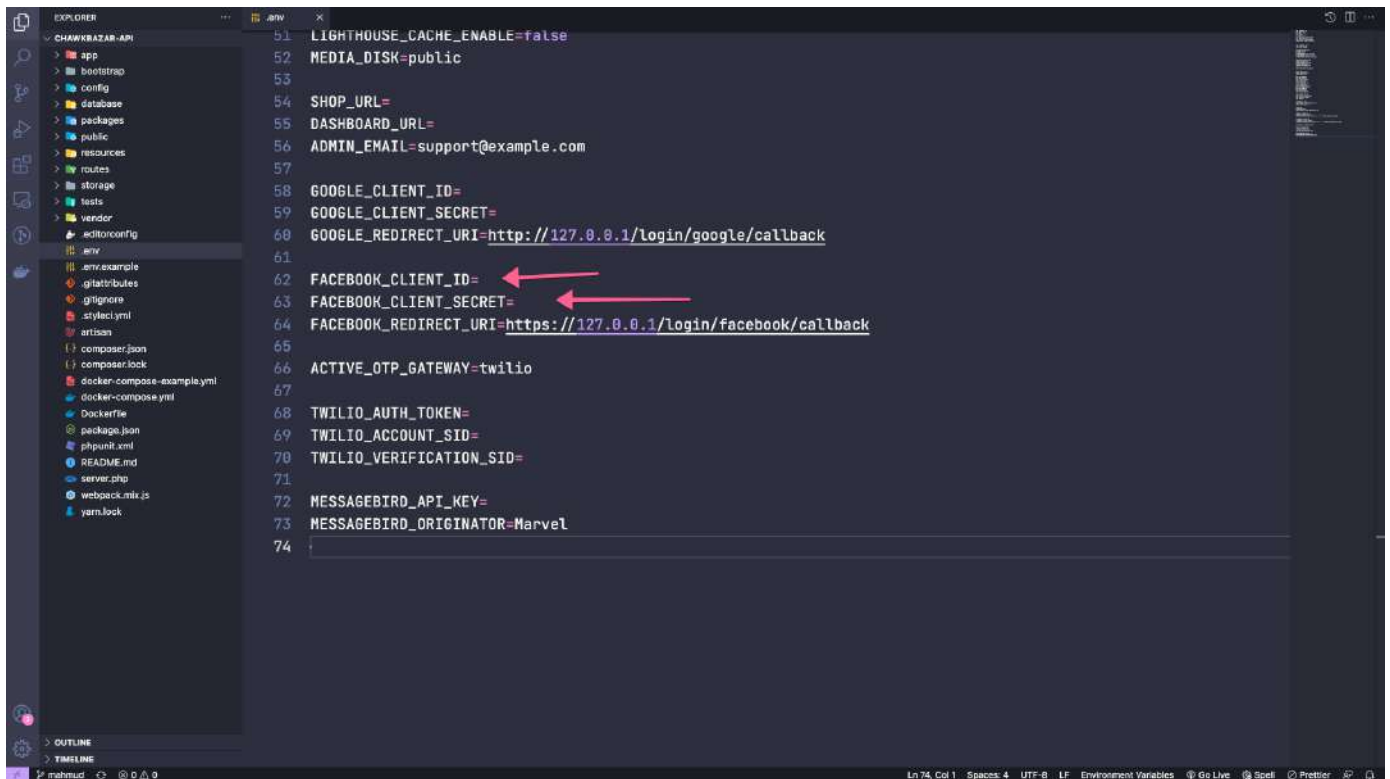
The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with files like .env, .env.template, .gitignore, additional.d.ts, DOCUMENTATION.md, next-env.d.ts, next.config.js, package.json, postcss.config.js, pwa.png, tailwind.config.js, TODO.md, tsconfig.json, vercel.json, and vercel.d.ts. The code editor shows the content of the .env file, which includes various environment variables for a Next.js application. A red arrow points to the line defining the SECRET variable.

```
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 SECRET= # Linux: 'openssl rand -hex 32' or go to https://generate-secret.now.sh/32
11
12 GOOGLE_ID=
13 GOOGLE_SECRET=
14
15 FACEBOOK_ID=
16 FACEBOOK_SECRET=
17
18 NEXT_PUBLIC_GOOGLE_MAP_API_KEY=ADD_GOOGLE_MAP_API_KEY
19
20 NEXT_PUBLIC_INSTAGRAM_BASIC_DISPLAY_USER_TOKEN=your_instagram_basic_display_user_token_here
21
```

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

Then open `shop/.env` and add credentials to

then open `shop/.env` and add the following to `FACEBOOK_CLIENT_ID` and `FACEBOOK_CLIENT_SECRET` also add a `SECRET` code.

A screenshot of a code editor showing the configuration of an environment file (.env). The file contains various settings for a Laravel application, including database credentials, shop URLs, and API keys. Two red arrows point to the `FACEBOOK_CLIENT_ID` and `FACEBOOK_CLIENT_SECRET` lines, which are currently empty, indicating where the user should paste their Facebook app credentials. The editor's interface includes a file explorer on the left, a main code area, and a status bar at the bottom.

```
51 LIGHHOUSE_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

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`.

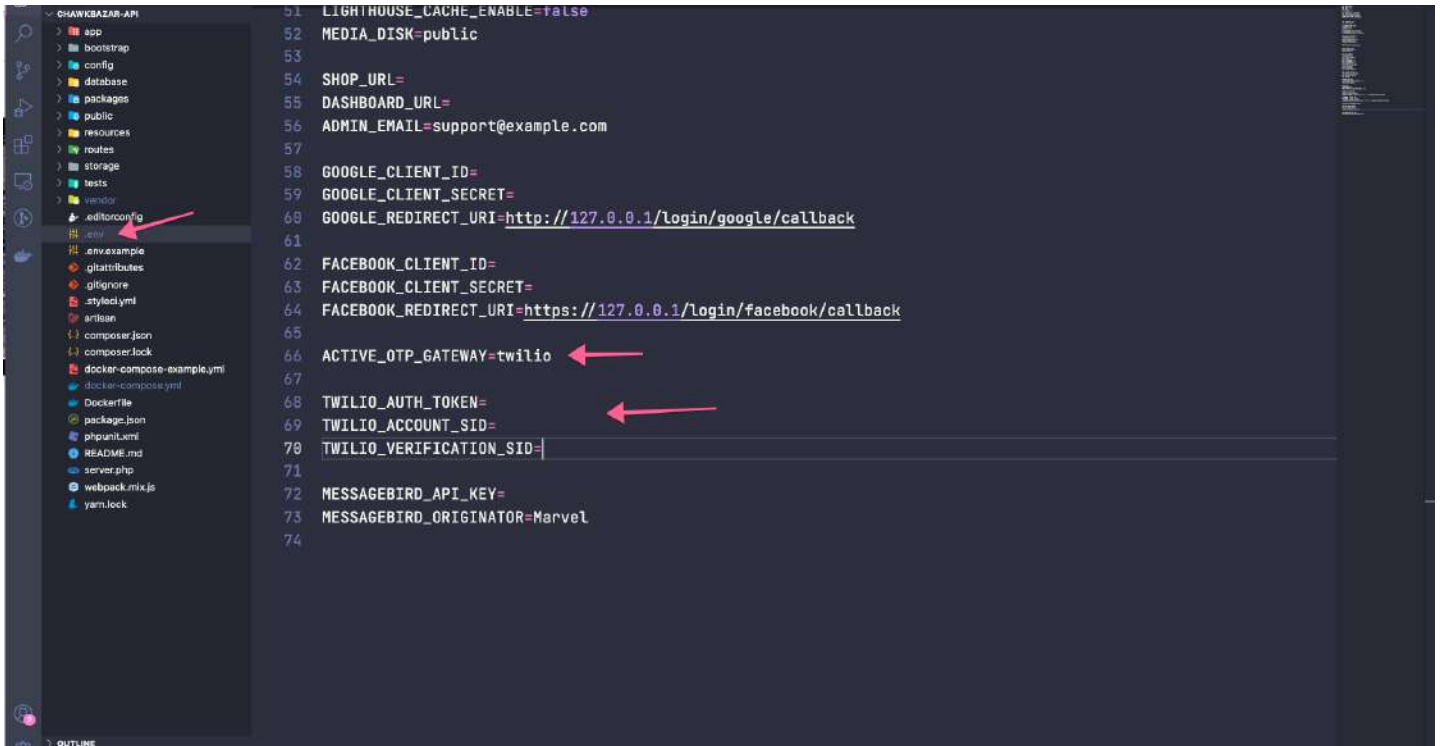
• Twilio Configuration:

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

Create an account at [Twilio](#)

After that, go to the `Twilio Console Panel` and

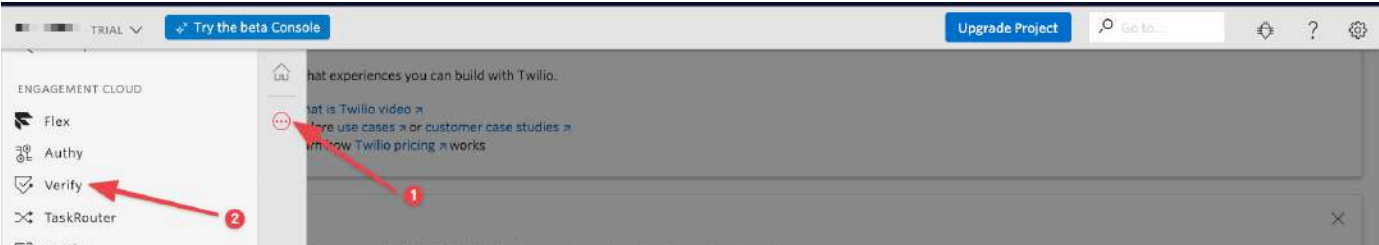
copy the `ACCOUNT SID`, and `AUTH TOKEN` key and add this key to your `api/.env` file

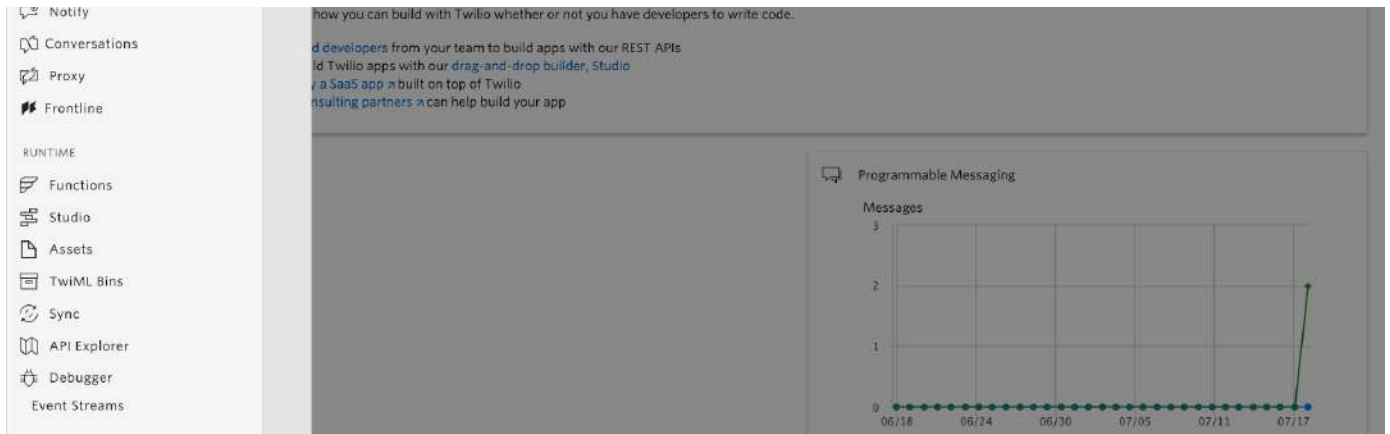


A screenshot of a code editor with a dark theme. On the left is a sidebar showing a file tree with folders like 'app', 'bootstrap', 'config', 'database', 'packages', 'public', 'resources', 'routes', 'storage', 'tests', 'vendor', and files like '.editorconfig', '.env', '.env.example', 'gitattributes', 'gitignore', 'stylelint.yml', 'artisan', 'composer.json', 'composer.lock', 'docker-compose-example.yml', 'docker-compose.yml', 'Dockerfile', 'package.json', 'phpunit.xml', 'README.md', 'server.php', 'webpack.mix.js', and 'yarn.lock'. The '.env' file is selected and highlighted with a red arrow. The main editor area shows the contents of the '.env' file, which includes various environment variables for a Laravel application, such as database settings, shop URL, dashboard URL, admin email, Google OAuth credentials, Facebook OAuth credentials, Twilio credentials, and MessageBird API key. Red arrows point to the 'ACTIVE_OTP_GATEWAY=twilio' line (line 66) and the 'TWILIO_AUTH_TOKEN=' line (line 68).

```
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
```

After that, go to the **Verify** option from the **Sidebar** panel,

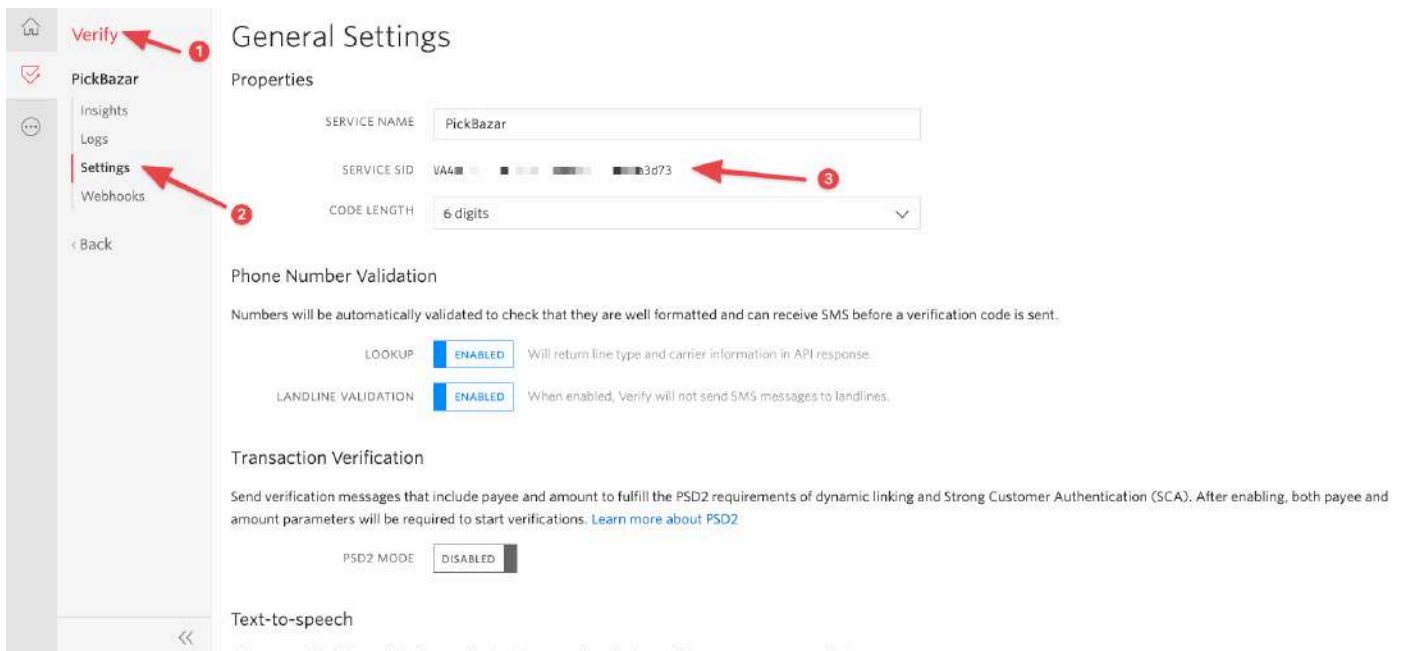




And from **verify dashboard**, create a service.

Add add that service key to **api/.env ->**

TWILIO_VERIFICATION_SID



```
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:

“

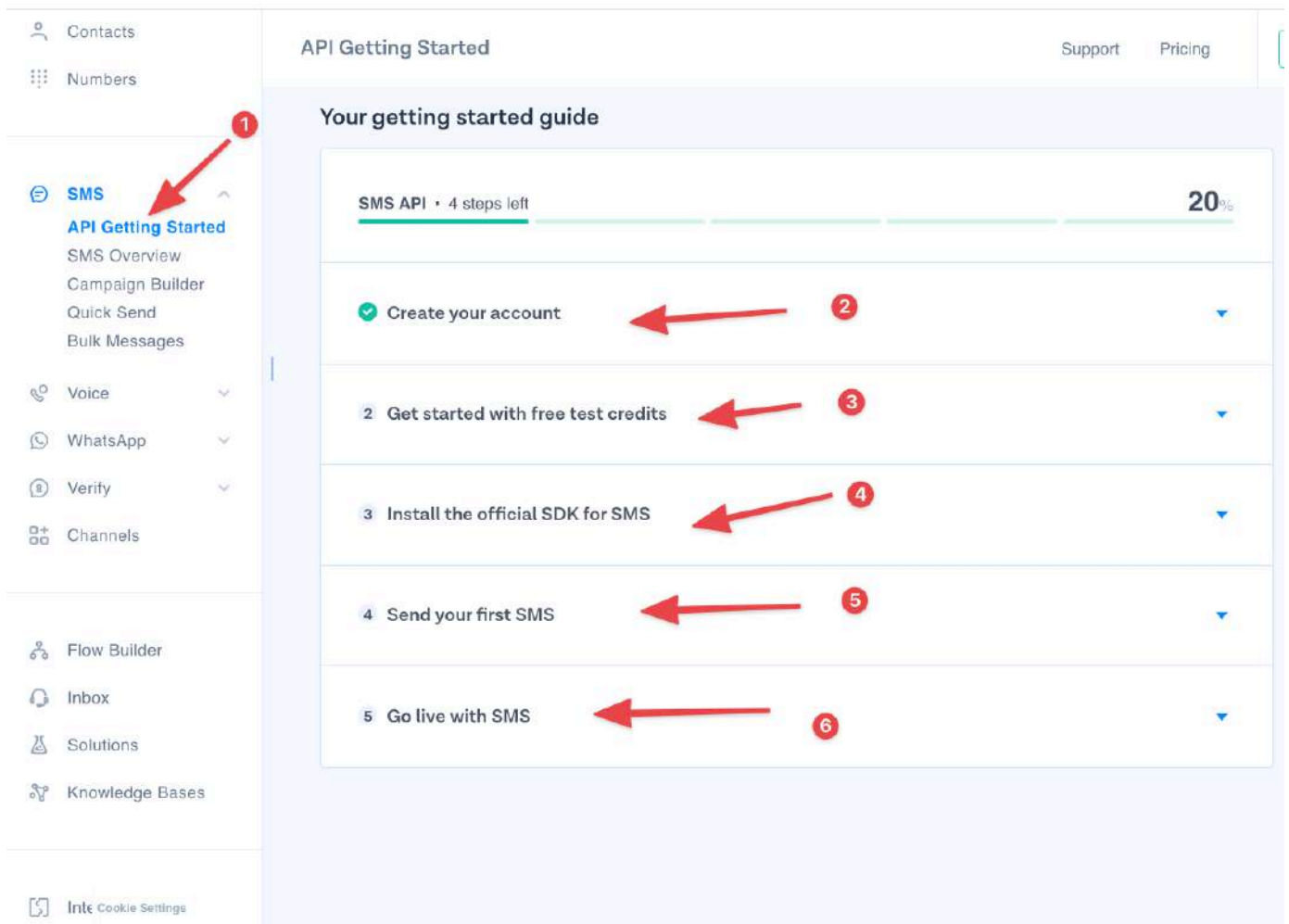
Use only one service provider for OTP service, so use this **MessageBird** configuration only if you want to use **MessageBird** instead of **Twilio**.

To configure **MessageBird**, follow this procedure,

Create an account from **MessageBird**

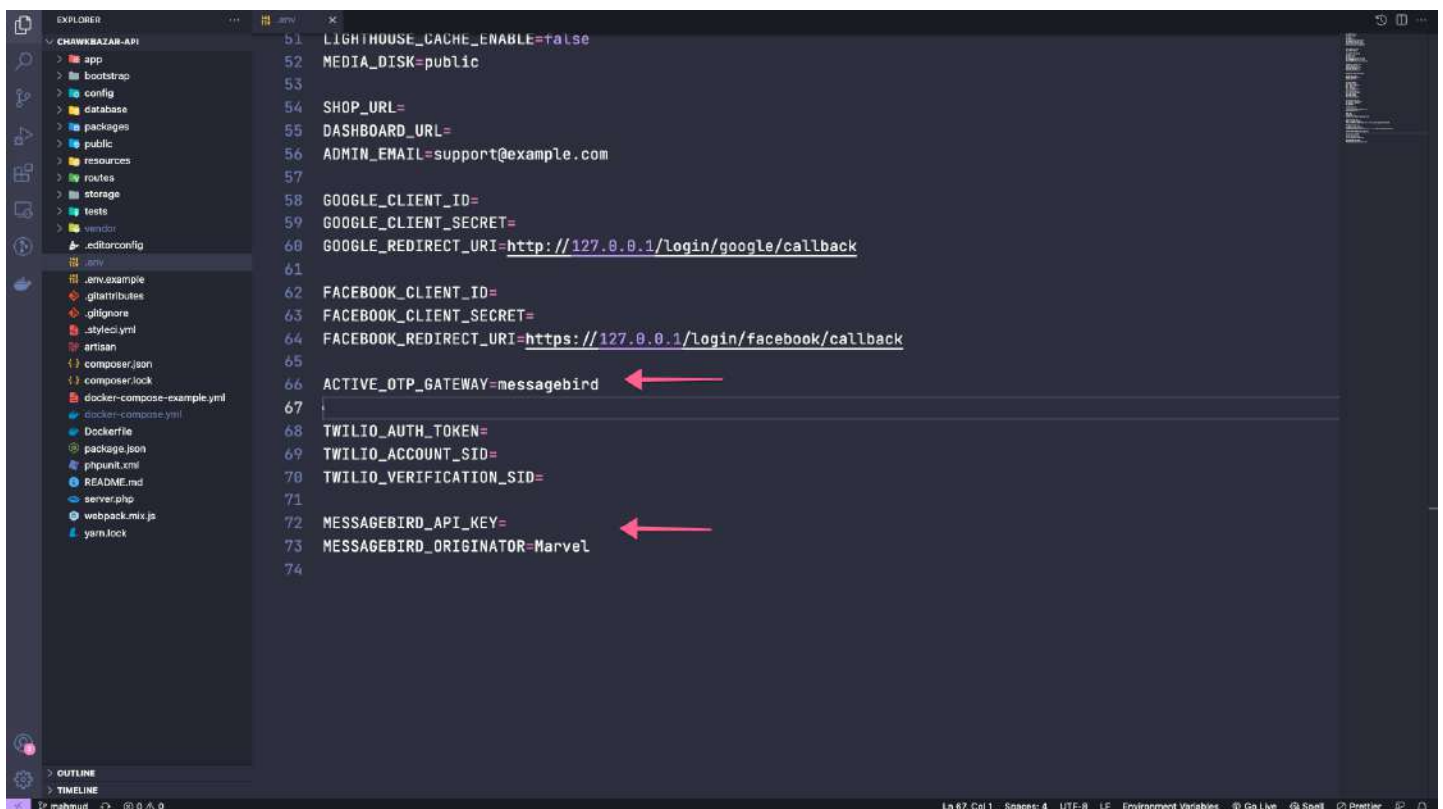
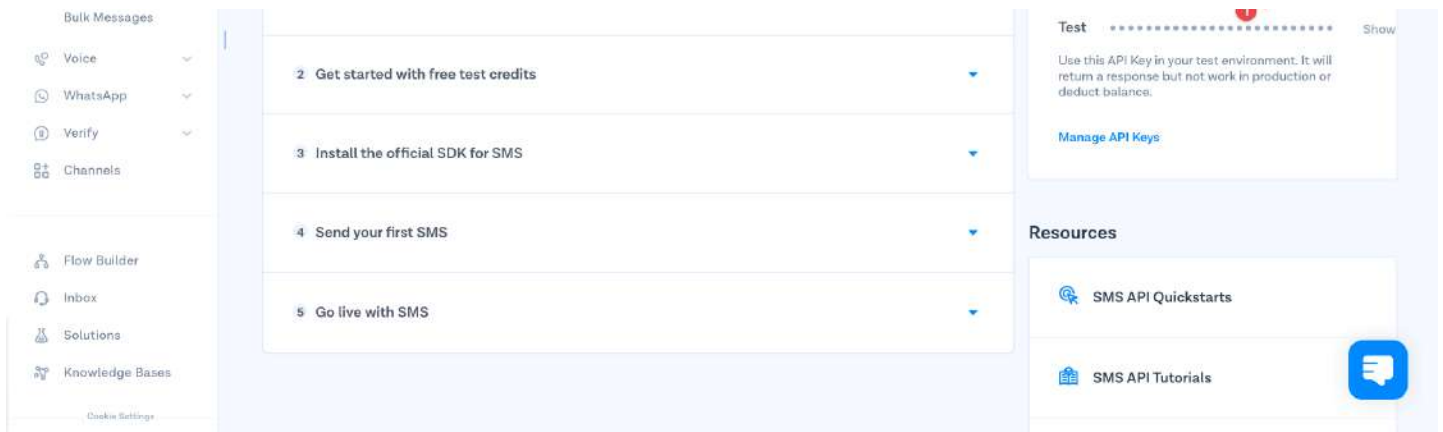
After creating the account, go to the **API Getting Started** and follow all the procedures

step by step.



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





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

FrontEnd Demo

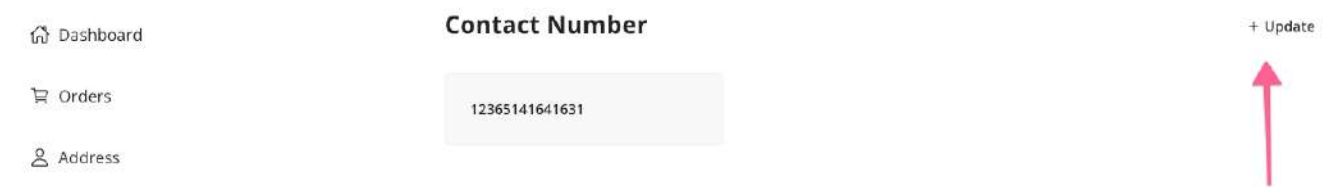
OTP Verification During Checkout Process:

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.



 [Contact Number](#)

 [Change Password](#)

 [Logout](#)

Multivendor

| Create New Shop

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

Email

Password [Forgot password?](#)

Login

Or

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

CHAWK
B A Z A A R

Register new account

Name

Email

Password

Register


Or


Already have an account? [Login](#)

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

CHAWK
B A Z A A R



[Create Shop](#)



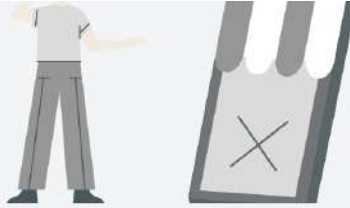


Mahmud
mahmud@test.com
[Please add your Profile](#)

My Shops



Enabled



No Shops Found

After that, click **Create Shop**

CHAWK BAZAAR

Create Shop

Mahmud
mahmud@test.com
Please add your [Profile](#)

Enabled

Logo
Upload your shop logo from here

Upload an image or drag and drop
PNG, JPG

Cover Image
Upload your shop cover image from here
Dimension of the cover image should be 1170 x 435px

Upload an image or drag and drop
PNG, JPG

Basic Info
Add some basic info about your shop from here

Name

Description

Create Shop

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

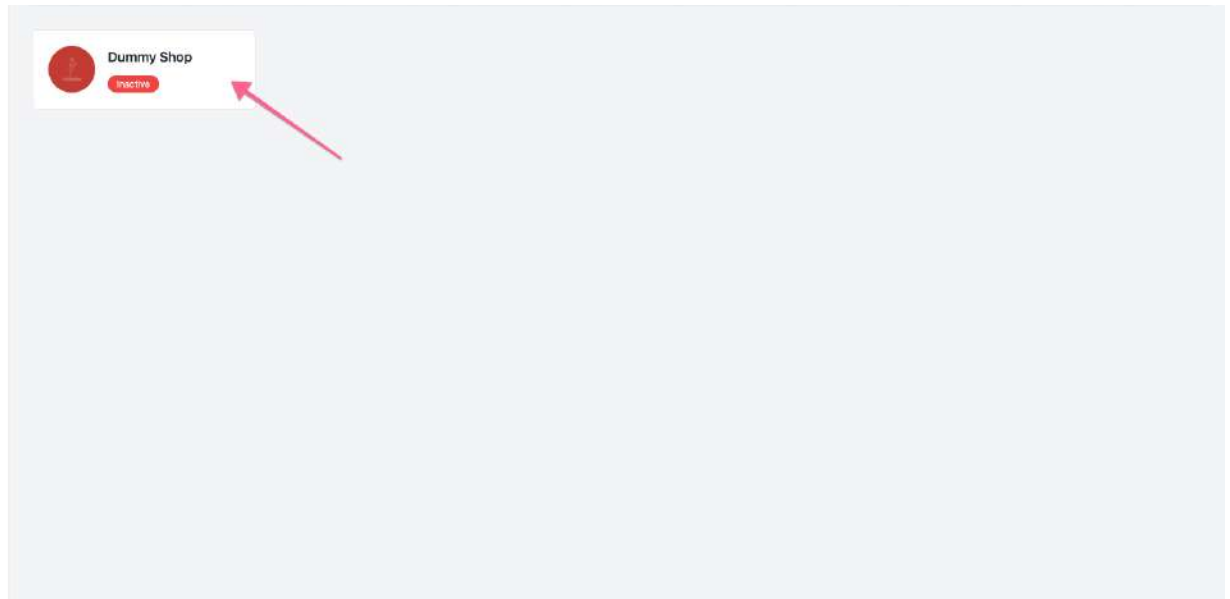
After creating the shop you'll redirect to this page,

CHAWK BAZAAR

Create Shop


My Shops







Mahmud
mahmud@test.com
Please add your [Profile](#)
✔ Enabled






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

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

CHAWK W A Z A R Create Shop 

 Dashboard  **Shops**  My Shops  Products  Attributes  Brands

Shops

Logo	Name	Owner Name	Products	Orders	Status	Actions
	Dummy Shop	Mahmud	0	0	inactive	 

Categories									
Tags									
Orders									
Order Status									
Users									
Coupons									
Taxes									
Shippings									
Withdraws									
Settings									

name	Admin	0	0	Inactive	✓	✖
Chawkbazar Vendor shop	Vendor	57	5	Active	✖	✖
Chawkbazar Shop	Admin	0	0	Active	✖	✖

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

CHAWK BAZAR

Create Shop

Mahmud

mahmud@test.com

Please add your [Profile](#)

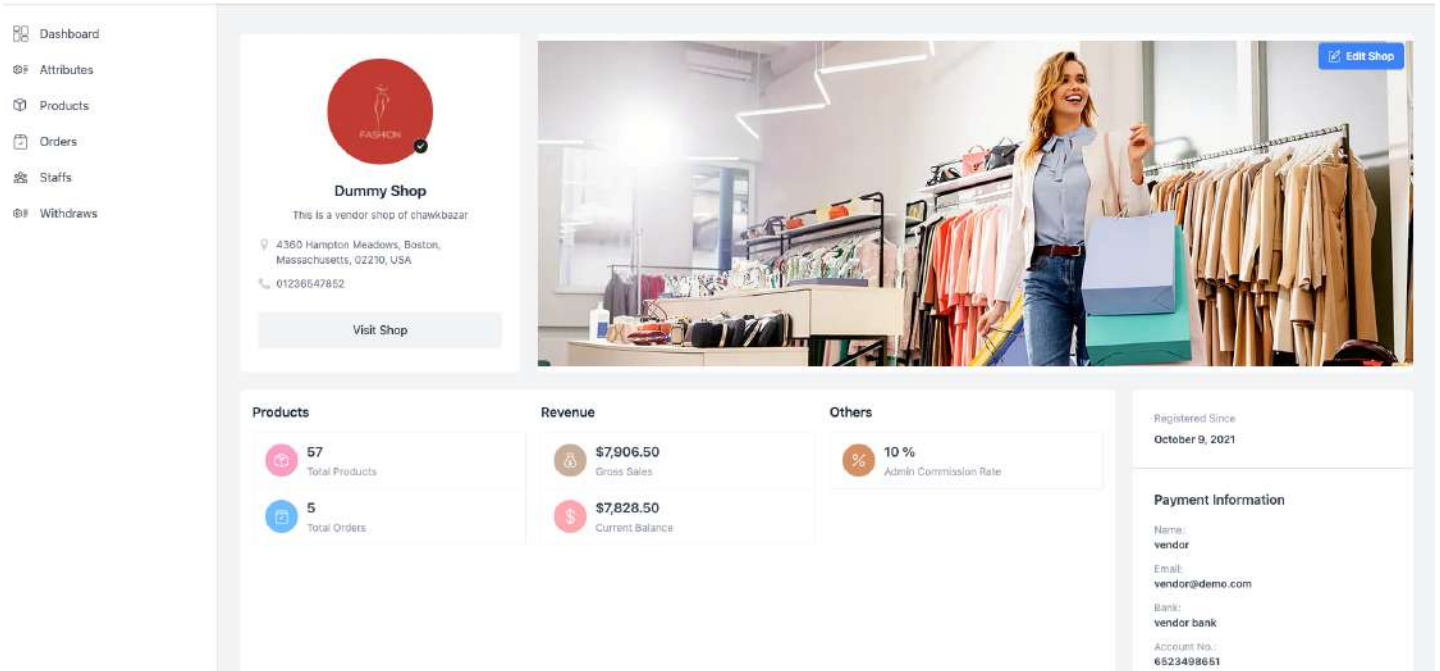
Enabled

My Shops

Dummy Shop

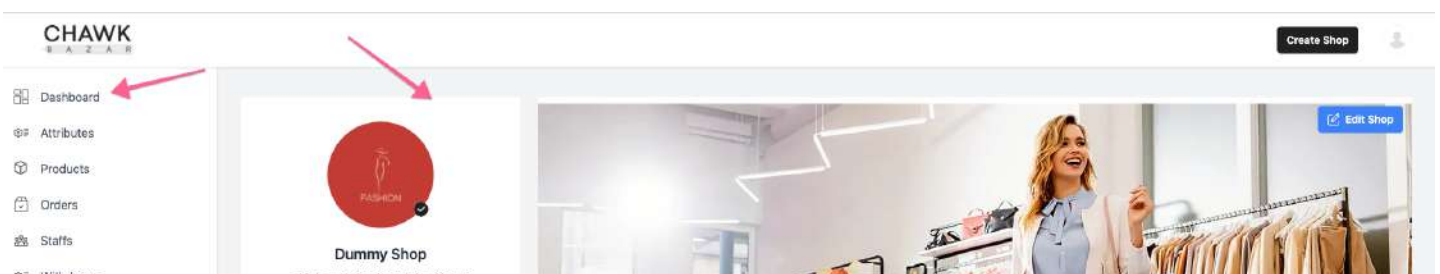
Active

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



From this dashboard, you can maintain you shop,

- Dashboard:



4360 Hampton Meadows, Boston,
Massachusetts, 02210, USA
01238547852

Visit Shop



Products

57
Total Products

5
Total Orders

Revenue

\$7,906.50
Gross Sales

\$7,828.50
Current Balance

Others

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

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Attributes

+ Add Attribute

ID	Name	Values	Actions
No data found			

- Products:

CHAWK

Create Shop

Dashboard

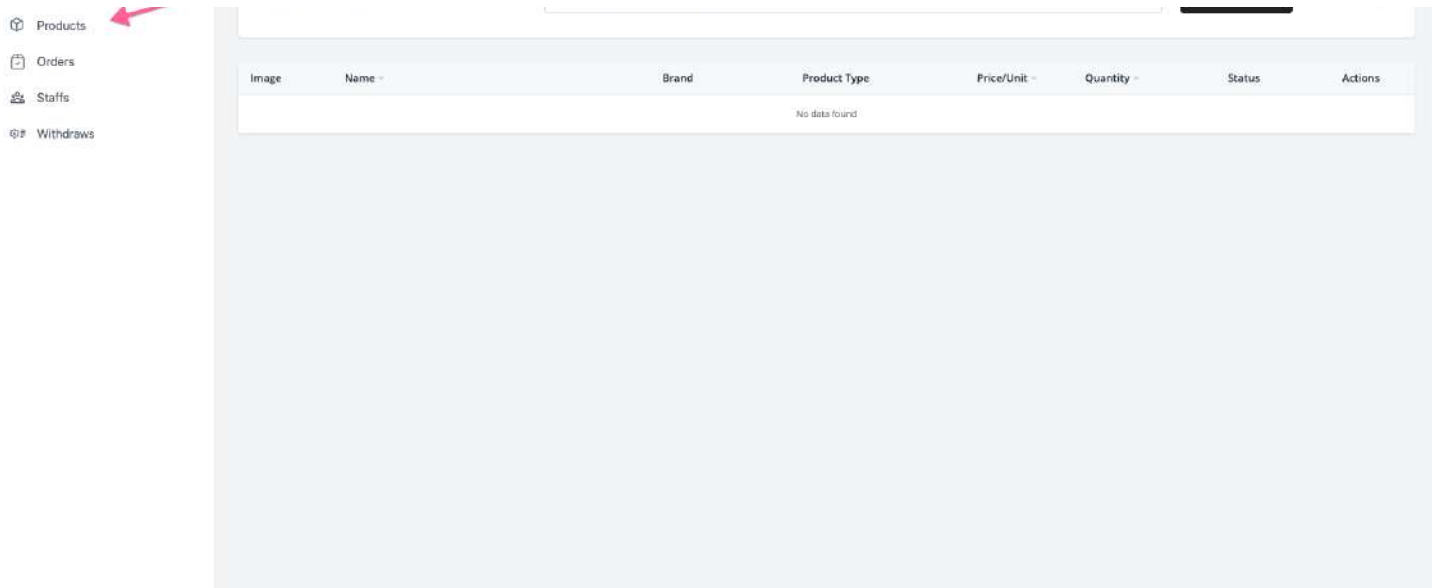
Attributes

Products

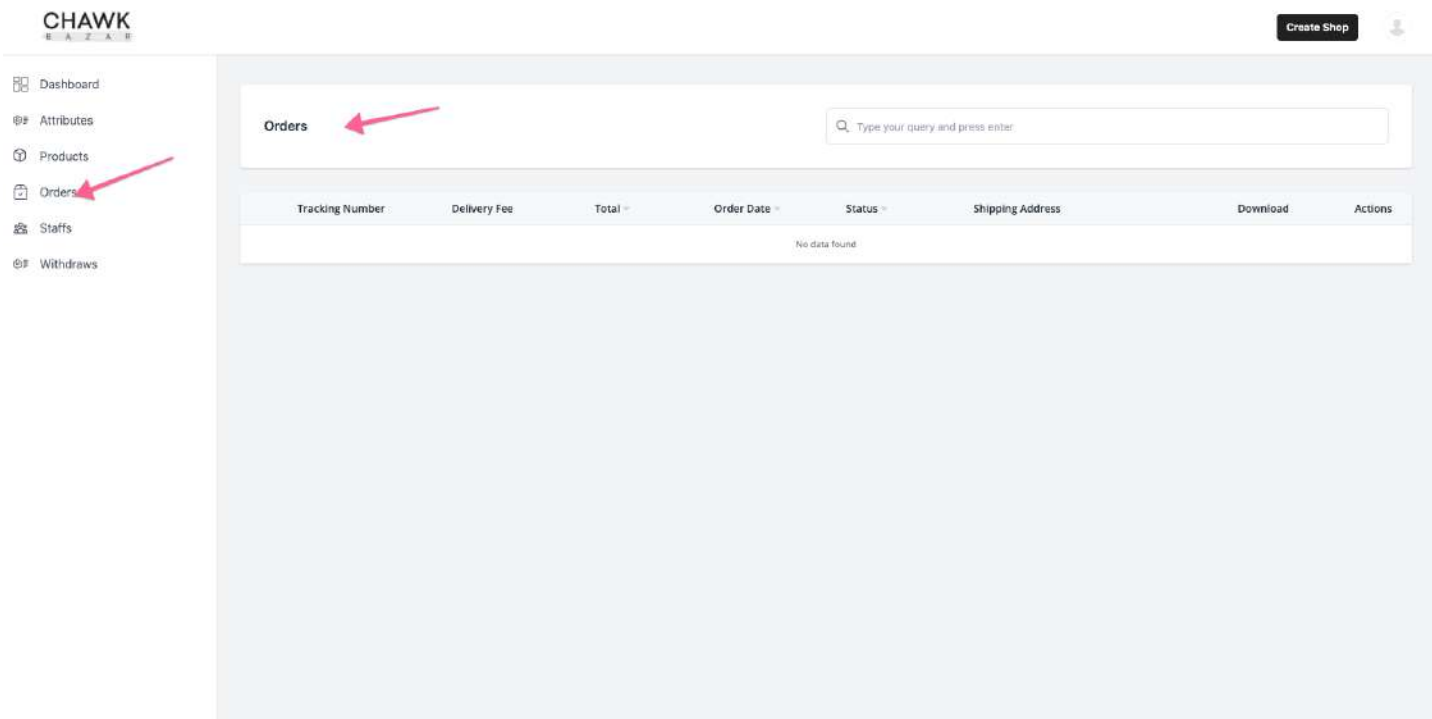
Products

+ Add Product

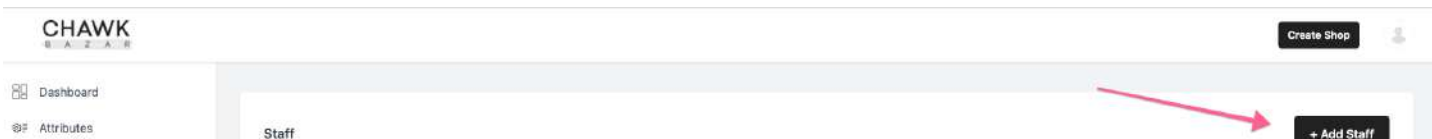
Filter



- Order:



- Staff:



- Products
- Orders
- Staffs
- Withdraws

Name	Email	Status	Actions
No data found			

- Withdraw:

CHAWK

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Withdraws

+ Request Withdraw

Shop Name	Amount	Status	Created
No data found			

User Roles:

- Super Admin -

Super Admin:

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

- Store Owner:

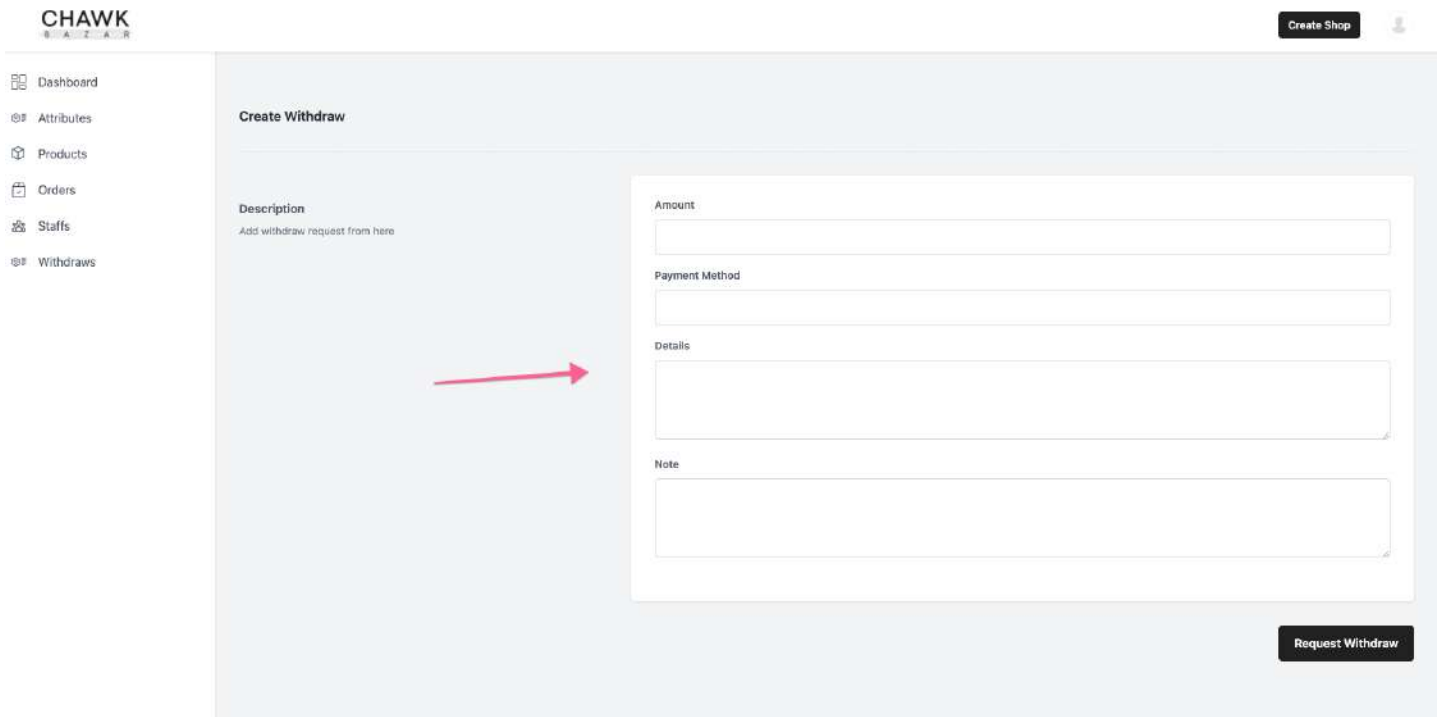
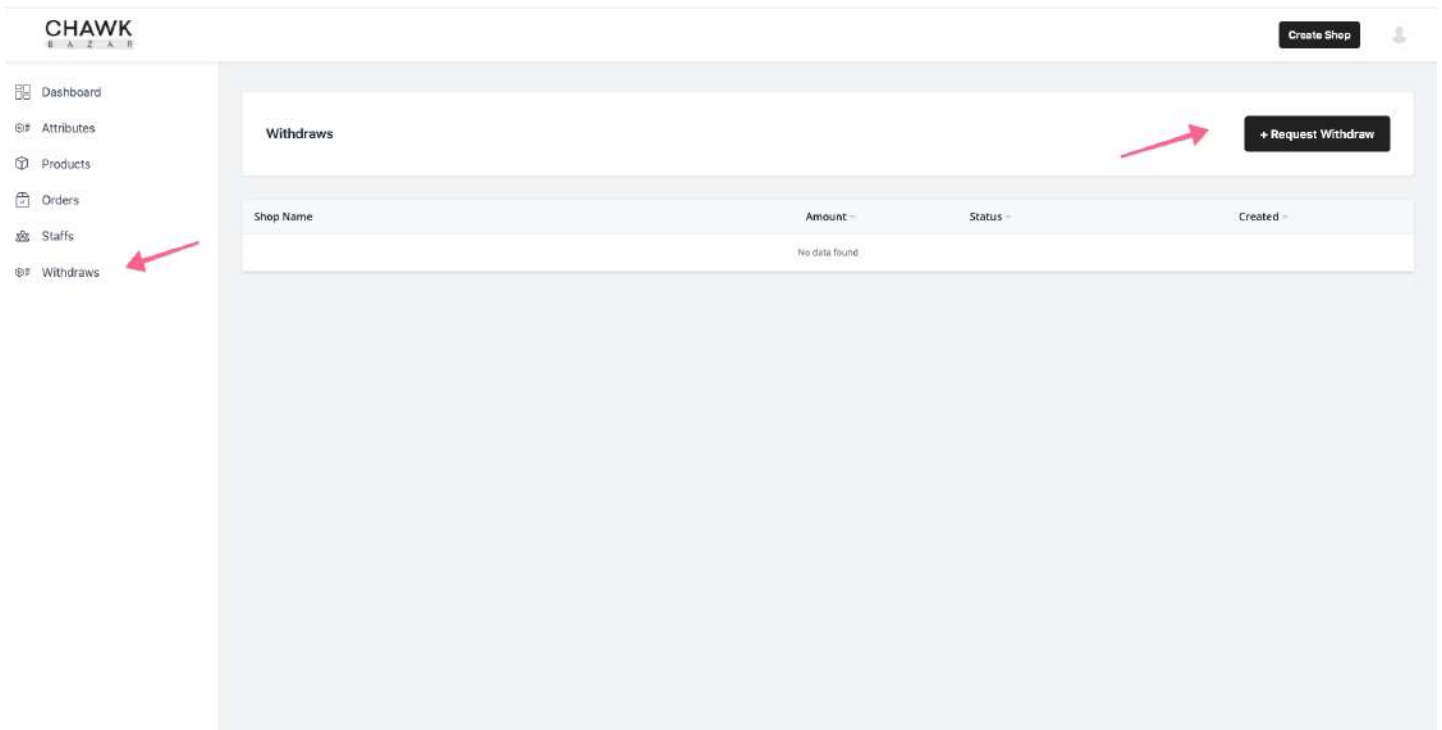
Store owner can edit or main it's store, staff or payment.

- 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,



After request payment, the dashboard will be like this,



- 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
BAZAR

- Dashboard
- Shops
- My Shops
- Products
- Attributes
- Brands

Withdraws

Create Shop

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	

Categories

Tags

Orders

Order Status

Users

Coupons

Taxes

Shippings

Withdraws

Settings

CHAWK

RAZAR

Create Shop

Dashboard

Shops

My Shops

Products

Attributes

Brands

Categories

Tags

Orders

Order Status

Users

Coupons

Taxes

Shippings

Withdraws

Settings

Withdrawal Information

Amount: 500

Payment Method: N/A

Status: Pending

Pending

Approved

On Hold

Processing

Pending

Rejected

Change Status

After approved,

CHAWK

RAZAR

Create Shop

Dashboard

Shops

My Shops

Products

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
Demos ▾ Men Wear ▾ Women Wear ▾ Casual Wear ▾ Search Pages ▾

Q Account

Super Shops Near You

Dummy Shop
NEW

Chawkbazar Vendor shop
NEW
 4360 Hampton Meadows,
 Massachusetts, Boston, 02210, USA

Chawkbazar Shop
NEW
 444-04, Zurich, 8021, Switzerland

Get Expert Tips In Your Inbox

Subscribe to our newsletter and stay updated.

[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,

CHAWK

[Demos](#) [Men Wear](#) [Women Wear](#) [Casual Wear](#) [Search](#) [Pages](#)



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: kjkjnk894561230

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

[Demos](#) [Men Wear](#) [Women Wear](#) [Casual Wear](#) [Search](#) [Pages](#)



Account



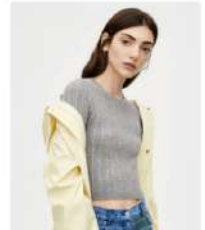


Chawkbazar Vendor shop
This is a vendor shop of chawkbazar

Address:
4360 Hampton Meadows, Massachusetts, Boston,
02210, USA

Phone:
01236547852

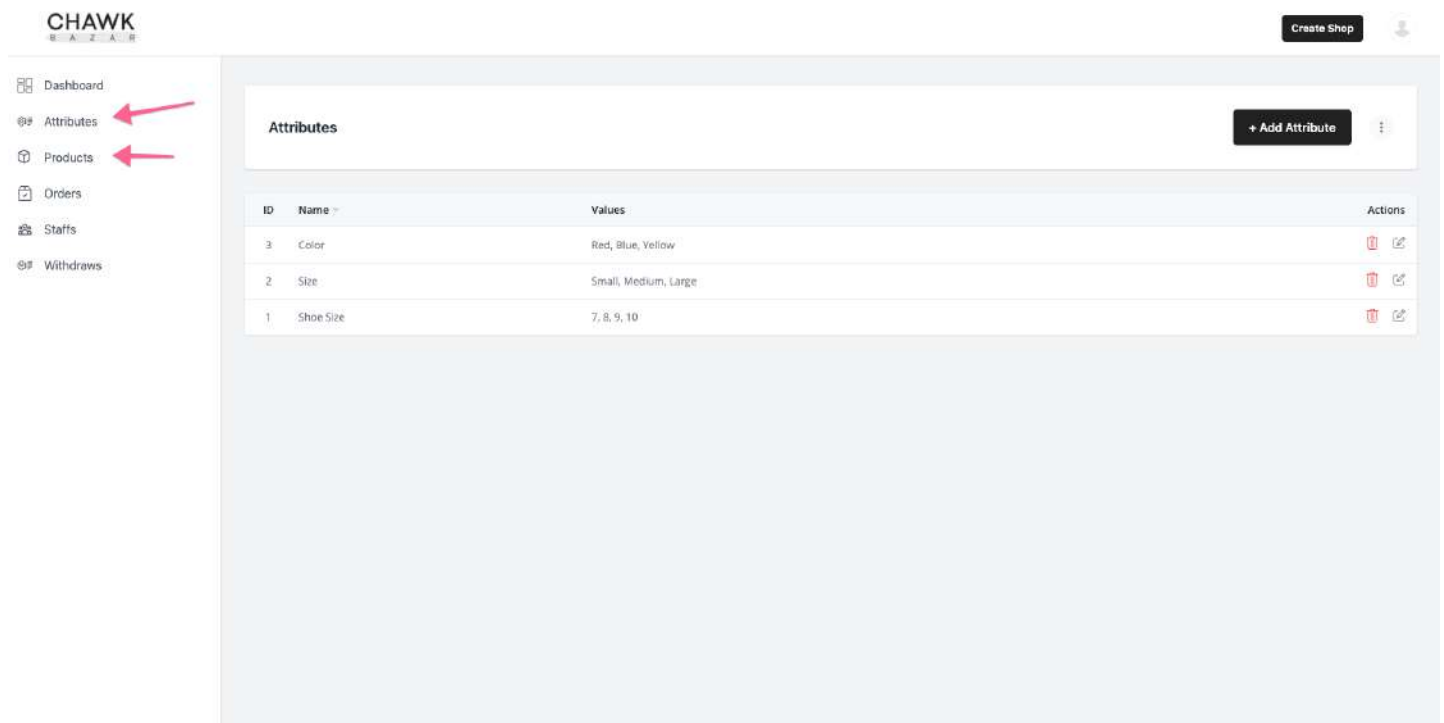
[Call Now](#)



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.

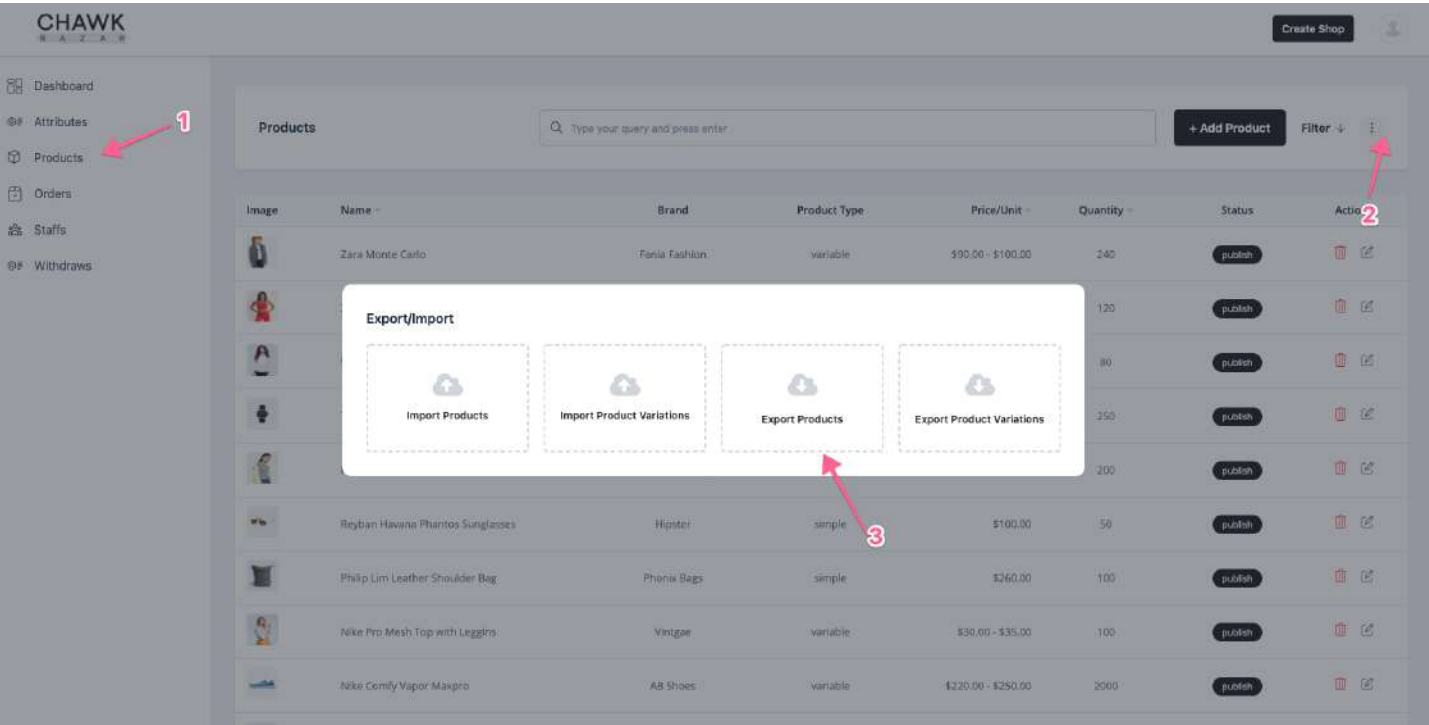


Export Import work different way for simple and variable product

| Simple Product

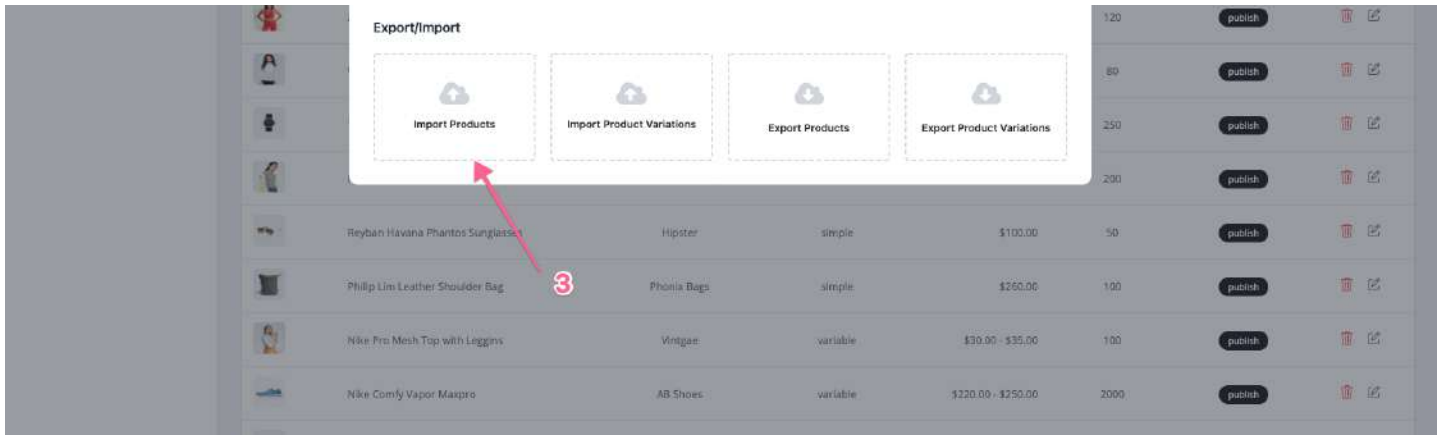
For simple products, you've to export only products data. To do that, go to **Your Shop -> products**. Then click on **three dots** and export

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** .



Attributes

Products

Orders

Staffs

Withdraws

Attributes

+ Add Attribute

Actions

ID	Name	Values
3	Color	Red, Blue, Yellow
2	Size	Small, Medium, Large
1	Shoe Size	

Export/Import

Import Attributes

Export Attributes

CHAWK

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Products

Q

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	publish	<div><div></div><div></div></div>
					120	publish	<div><div></div><div></div></div>
					80	publish	<div><div></div><div></div></div>
					250	publish	<div><div></div><div></div></div>
					200	publish	<div><div></div><div></div></div>
	Rayban Havana Phantos Sunglasses	Hipster	simple	\$100.00	50	publish	<div><div></div><div></div></div>
	Philip Lim Leather Shoulder Bag	Phonix Bags	simple	\$260.00	100	publish	<div><div></div><div></div></div>
	Nike Pro Mesh Top with Leggings	Vintage	variable	\$30.00 - \$35.00	100	publish	<div><div></div><div></div></div>
	Nike Comfy Vapor Maxpro	AB Shoes	variable	\$220.00 - \$250.00	2000	publish	<div><div></div><div></div></div>

Export/Import

Import Products

Import Product Variations

Export Products

Export Product Variations

Attributes

Products

Orders

Staffs

Withdraws

Attributes

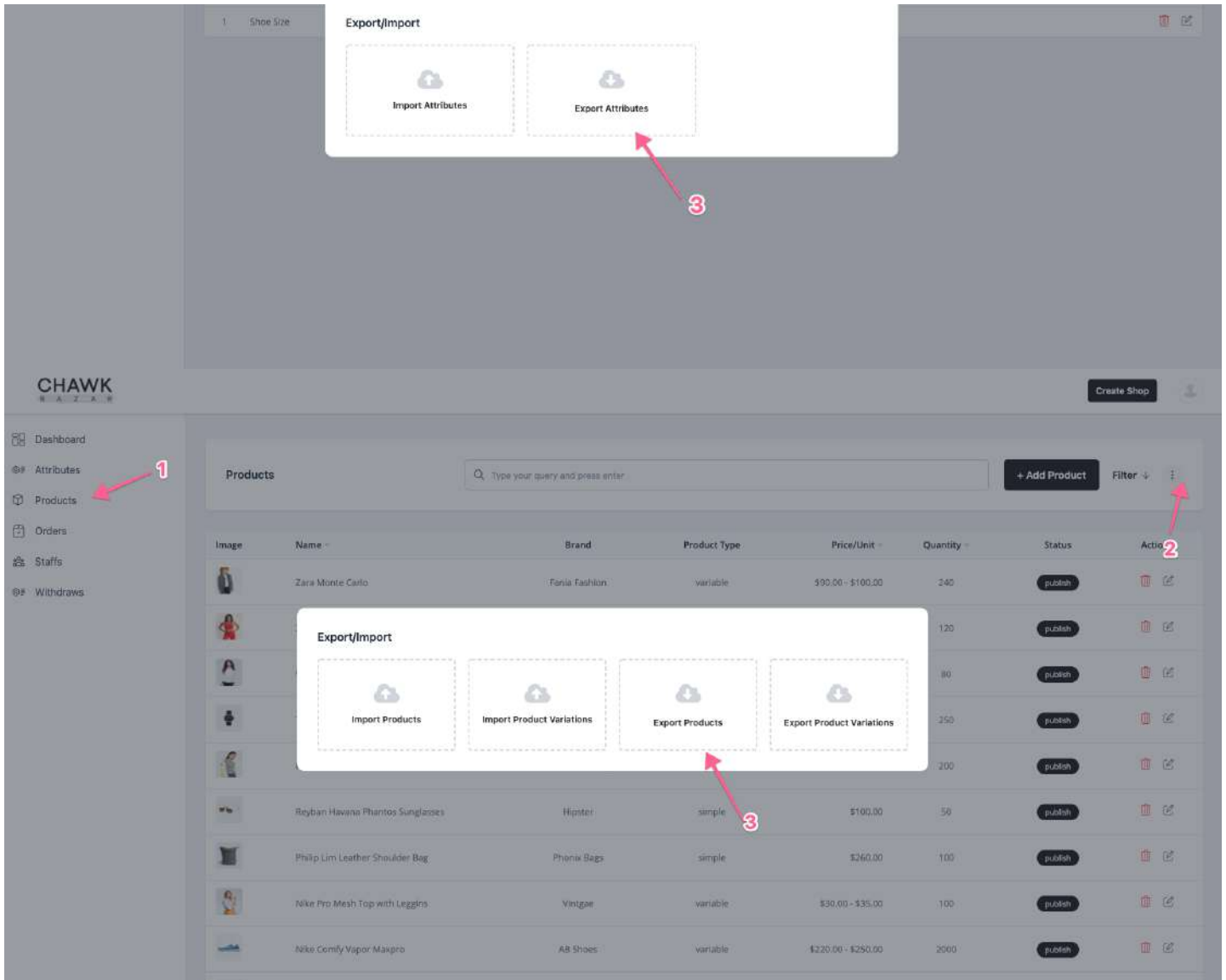
+ Add Attribute

Actions

3	Color	Red, Blue, Yellow
2	Size	Small, Medium, Large

CHAWK

Create Shop



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



1 Shoe Size

Export/Import

Import AttributesExport Attributes

CHAWKRAZAR

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Products

Type your query and press enter

+ Add Product

Filter

Image	Name	Brand	Product Type	Price/Unit	Quantity	Status	Action
	Zara Monte Carlo	Fania Fashion	variable	\$90.00 - \$100.00	240	publish	<div></div>
					120	publish	<div></div>
					80	publish	<div></div>
					250	publish	<div></div>
					200	publish	<div></div>
	Reyban Havana Phantos Sunglasses	Hipster	simple	\$100.00	50	publish	<div></div>
	Phillip Lim Leather Shoulder Bag	Phonia Bags	simple	\$260.00	100	publish	<div></div>
	Nike Pro Mesh Top with Leggings	Vintage	variable	\$30.90 - \$35.00	100	publish	<div></div>
	Nike Comfy Vapor Maxpro	AB Shoes	variable	\$220.00 - \$250.00	2000	publish	<div></div>

Export/Import

Import ProductsImport Product VariationsExport ProductsExport Product Variations

CHAWKRAZAR

Create Shop

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

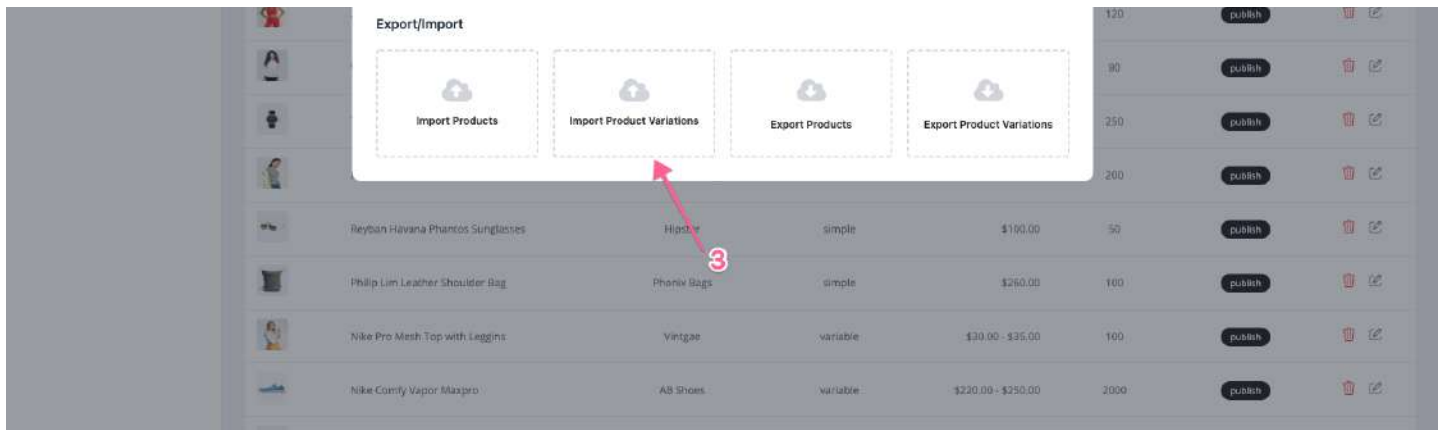
Products

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	publish	<div></div>



Thank You!
