

Introduction

Navigation

Supported Platforms & Framework For Local Development

Requirements

How It Works ?

Admin Dashboard (GraphQL & REST)

Shop Frontend (GraphQL & REST)

API

Getting Started

Folder Structure

Prerequisites for Backend

Prerequisites for Frontend

Project initialization

Getting started with Windows

For MAC and Linux(with sail and docker)

Prerequisites

Installation Mac

Video

REST API

GraphQL API

Getting Started with Frontend

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

For Admin :

For Shop :

Shop config

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

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

For Shop :

Shop config

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

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

Common How to use

For Docker

For Generalize installation in local PC

PHP Artisan Oriented Scripts

Available Scripts:

Sail Oriented Scripts

Available Scripts:

Most uses Laravel commands.

Available Scripts:

Available Scripts:

Multi-Vendor

Create New Shop

 Dashboard:

 Attributes:

 Products:

 Order:

 Staff:

 WithDraw:

User Roles:

 Super Admin:

 Store Owner:

 Staff:

Withdraw Payment:

FrontEnd Shop

 Super admin

 Vendor or Store Owner

 Staff

 Customer

Introduction to Features

 Settings Management

 International Slug

Introduction to Analytics

Introduction to Shop

 Super Admin

 Vendor

 Staff

Introduction to Products management

 Products

 Product form

 Super Admin (Product form)

 Vendor (Product form)

 Inventory

 Product group

 Product Category

 Product Tags

 Product Attributes

 Product Manufacturer/Publications

 Product Authors

Export Import

 Simple Product

 variable products

Introduction to Features

 Tax Management

 Shipping Management

- Free Shipping
- Cash on delivery
- Shipping and Billing Address
- Shipping process in general
- Withdrawal Management

Refund

Introduction to Features

- Manage Product Type
- Manage FAQs
- Manage Terms & Conditions

Orders Management

- Super Admin
- Vendor

User Management

Super Admin to User Management

- Vendor Management & Staff Management
- Customer Management

Introduction to Feedback management

- Review

Question

Introduction to Promotional management

- Coupon Management
- Flash Sale

Introduction to Engagement

Message

- Store Notice
 - Notice creation for specific vendor
- Vendor
 - Store Notice Vendor profile
- Staff
 - Create a Notice for particular shop
 - Creation a notice for [furniture-shop]
 - Issued notice for a specific shop [furniture-shop]
 - Store Notice staff profile
 - Staff for [furniture-shop]

Settings Management

- General settings
- Change Logo
 - For General Logo View:
 - For Collapsible Logo View:
 - API
 - Change Site information

Shipping class, Tax class & Free Shipping
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.

To Enable or Disable Verify Email
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

Payment settings

Super Admin

Customer

Event Settings

Event Settings Sidebar panel

Real-time Notification Settings

SMS Event Settings

Email Event Settings

SEO

Shop settings

Add or Remove Delivery Schedule

Shop Settings

Company Information

Address

address with google map

address without google map

Social platform

Footer

Payment

Stripe

Stripe integrate inside PickBazar.

How to create & setup Stripe information properly?

Special Notes for Stripe users.

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

Stripe Element

Stripe Element integrate inside PickBazar.

How to create & setup Stripe Element information properly?

How can I add stripe element's another payment gateway?

How to customize your stripe element payment method.

Special Notes for Stripe element users.

Available Stripe Element Payment Method

PayPal

PayPal integrate inside PickBazar.

How to create & setup PayPal information properly?

RazorPay

RazorPay integration inside PickBazar.

How to create & setup RazorPay information properly?

Special Notes for RazorPay.

Mollie

Mollie integrate inside PickBazar.

Mollie Webhook settings

How to create & setup Mollie information properly?

Special Notes for Mollie.

Paystack

Paystack integrate inside PickBazar.

Paystack Webhook settings

How to create & setup Paystack information properly?

Special Notes for Paystack.

SSLCOMMERZ

SSLCOMMERZ integrate inside PickBazar.

How to create & setup SSLCOMMERZ information properly?

Xendit

Xendit integrate inside PickBazar.

Xendit Webhook settings

How to create & setup Xendit information properly?

Special Notes for Xendit.

Iyzico

Iyzico integrate inside PickBazar.

How to create & setup Iyzico information properly?

Paymongo

Paymongo integrate inside PickBazar.

How to create paymongo webhook url?

You can visit for paymongo.

bKash

bKash integrate inside PickBazar.

How to create & setup bKash information properly?

Special Notes for bKash.

Flutterwave

Flutterwave integrate inside PickBazar.

Flutterwave Webhook settings

How to create & setup Flutterwave information properly?

Special Notes for Flutterwave.

Integration of New Payment Gateway

Getting Started with API

Step 1: Install and configure the payment gateway package

Step 2: Add payment gateway name in the Enum.

Step 3: Configure the Payment Facade for the new payment gateway.

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

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

Getting Started with admin dashboard.

Getting Started with shop front.

 Redirect-base Payment Gateway

 Non-redirect based payment gateway

Multiple payment gateway

How to activate multiple payment gateways [in Admin] ?

 Description [sequence by number in the screenshot]

How to use multiple payment gateways from customer end [in Shop] ?

 1. Select a payment gateway during place order

 2. Select different payment gateway (if needed)

 Description [sequence by number in the screenshot]

Introduction to Flash Sale

Super Admin to Flash Sale

Key concept

Percentage basis

Fixed rate basis

All listed flash sale

Flash sale single view in admin panel.

Flash Sale for Vendors and Staffs.

Flash Sale for Customer

Social Login

Google

Facebook

OTP (Mobile Number Verification)

Configuration

 Twilio Configuration:

 Verify Geo Permissions

 MessageBird:

FrontEnd Demo

 Login with Mobile Number:

 OTP Verification During Checkout Process:

 OTP Verification For Updating Mobile Number:

Wallet

 Sign Up Points:

 Manually By Admin:

 Refund:

 Wallet Ratio:

Email Configuration

Mailgun Configuration

Gmail SMTP Configuration

Email Verification

How to turn on this feature

Admin view of this feature

Shop view of this feature

Configuration

Subscribe setup

Implementation of Subscribe

How to Get Mailchimp API Key

How to Get Mailchimp List ID

Introduction

Refund Request

Refund Reports

Customer and vendor end

Customer

Vendor

Introduction to Real-time Notification

Configure pusher

Configure settings

Key notes

Multilingual

Step 1: Enable Multilingual Feature

API:

Admin

Shop

Step 2: How to add new language in admin?

Step 3: How to add new language in shop?

Step 3: How to translate static content for admin?

Step 4: How to translate static content for shop?

Step 5: How to translate content?

Data Type

Translate Settings

Translate Order Status

Translate Group, Author, Manufacturer, Tags, Categories:

Translate Products

 Translate Simple Products:

 Translate Variable Product:

Translate Attributes

Translate Groups

Translate Variable Products

Step 6: How to add custom or solved broken font issue?

Translation

Existing Language

 For Rest

 For GraphQL

New Language

Default Language

For Rest

For GraphQL

OpenAI

OpenAI Integration inside PickBazar.

Admin Settings

Special Notes for Paystack.

Find or Search Nearby shops inside PickBazar.

Admin Settings

Demo Deployment

AWS (Amazon Web Service)

How to create ec2 server?

Domain Setup

Login to Server

Change permission .pem

VPS Server

Virtual Private Server (Automated Script)

Prerequisite

At first login your server from terminal

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

Server Environment setup script - RUN on Virtual Server

Nginx Setup And Settings - RUN on Virtual Server

Backend build - RUN on Virtual Server

Frontend build script - RUN on Local PC

Frontend run script - RUN on Virtual Server

cPanel

Access Server

Create Subdomains

Install API

Install FrontEnd

FrontEnd Project Build

step 1 - Build Custom Server

shop rest

admin rest

admin graphql

shop graphql

Step 2 - Install & Build

Step 3 - Build the project

For REST

For GraphQL

Install NodeJs Project

Vercel

vercel.com

API

Frontend

Virtual Private Server

Access Server

Install NodeJS & Required Application

 Install NodeJS

 Install Yarn

 Install Zip & Unzip

 Install PM2

Setup Server

 Introduction

 Step 1 - Installing Nginx

 Step 2: Adjusting the Firewall

 Step 3 – Checking your Web Server

 Step 4 - Install MySQL

 Step 5 - Install PHP & Composer

 Step 6 - Create MySQL Database & User

 Step 7 - Change permission for the `www` folder

 Step 8 - Upload API to Server

 Step 9: Setting Up Server & Project

 Step 10 - Create New Nginx for the domain

 For REST API

 For GraphQL API

Secure Server

 Step 1: Secure Nginx with Let's Encrypt

Install API

 Step 1: Build and Run `api`

FrontEnd Project Build

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

 Step 2 - Install & Build

 Step 3 - Build the project

 For REST

 For GraphQL

 Build Project

Install FrontEnd And Run

 Run frontend app

 For REST API

 For GraphQL API

 For GraphQL

 Uses Example:

 For REST API

 Uses Example:

For customizing the template's default site settings:

CSS styles:

Icons:

For Adding a custom Icon:

New Static Page

Shop,

Admin,

REST

GraphQL

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

PickBazar Update - Virutal Private Server

Step 1: Setup Git - Server

Install git

Config for first time

Prepare Git Repository

Git & Github

Step 2: Shut Down Current Process

Step 3: Local Repository & Updated Code

Step 4: Update API

Step 5: FrontEnd Project Build

Config Next Admin App For /admin Sub Directory

Install & Build

Build the project

Upload to GitHub

Step 6: Upload Frontend & Run

Run frontend app

FAQ Page

Customer End

Vendor End

Contact

Customer End

Vendor End

Terms and conditions

Customer End

Vendor End

Existing Project Update Guide

Why Update Your PickBazar Project?

Preparing for the Update

General FAQs

How to configure Stripe payment gateway?

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

I am changing schema files but changes is not working

Changing .env files but not getting the changes

Changing route but not getting the changes.

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

Can I use it with my existing laravel?

Why am I getting `Access denied for user`?

Why am I getting permission issue during deployment?

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

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

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

API

Admin

Shop

Build Frontend,

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

Reason 1:

PORT:

Reason 2:

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

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

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

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

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

How to rebuild the project?

How to increase upload size?

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

How to remove the existing payment gateway?

Why checkout `Place Order` button is disabled?

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

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

Breaking changes

Breaking Change Log Version: 11.0.0

Breaking Change Log Version: 10.0.0

Breaking Change Log Version: 8.0.0

Breaking Change Log Version: 6.0.0

Breaking Change Log Version: 5.0.0

Breaking Change Log Version: 2.0.0

Changelogs

Version: 11.0.0 : Major

3rd November 2023

Version: 10.6.0

28 August 2023
Version: 10.5.0

17 August 2023
Version: 10.4.0

24 July 2023
Version: 10.3.0

12 July 2023
Version: 10.2.0

13 June 2023
Version: 10.1.0

30 May 2023
Version: 10.0.0 Major

15 May 2023
Version: 9.1.0

15 May 2023
Version: 9.0.0 Major

2 May 2023
Version: 8.2.0

3 April 2023
Version: 8.1.0

21 March 2023
Version: 8.0.0 Major

15 March 2023
Version: 7.0.0 Major

February 2023
Version: 6.3.1

9 January 2023
Version: 6.3.0

22 December 2022
Version: 6.2.0

12 December 2022
Version: 6.1.0

8 December 2022
Version: 6.0.0 Major

16 November 2022
Version: 5.0.1

16 August 2022
Version: 5.0.0 Major

7th July 2022
Version: 4.6.0

18th April 2022
Version: 4.5.1

1st April 2022

Version: 4.5.0

1st March 2022

Version: 4.4.2

17th February 2022

Version: 4.4.1

17th January 2022

Version: 4.4.0

10th January 2022

Version: 4.3.0

25th October 2021

Version: 4.2.0

18th October 2021

Version: 4.1.0

4th October 2021

Version: 4.0.0 Major

26th August 2021

Version: 3.0.0 Major

22 August 2021

Version: 2.3.1

3rd August 2021

Version: 2.3.0

2nd August 2021

Version: 2.2.0

18th July 2021

Version: 2.1.0

8th July 2021

Version: 2.0.1

3rd July Friday 2021

Version: 2.0.0 Major

1st July Wednesday 2021

Version: 1.3.0

16th June Wednesday 2021

Version: 1.2.0

9th June, 2021

Version: 1.1.0

10th May Monday 2021

Version: 1.0.1

Thursday 22 April 2021

Version: 1.0.0

Thursday 14 April 2021

Introduction

Welcome to the PickBazar documentation!

We have presented one of the fastest e-commerce App built with React, NextJS, TypeScript, Laravel and Tailwind CSS. And it has both GraphQL & REST API support. Here, we have tried to cover all the available features & topics presented in this application. We are inviting your to go through the whole documentation, so that it may helps you to understand better about what we are offering.

Navigation

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

Supported Platforms & Framework For Local Development

In your production environment, maintaining similar configuration is recommended.

- MacOS, Windows, and Linux are supported
- Compatible Browsers (Firefox, Safari, Chrome, Edge)
- JavaScript (Node.js 16.15.1 or later)
- Laravel 10 environments with PHP 8.1 (at least)

Requirements

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

How It Works ?

We have used mono-repo folder structure with Yarn Workspace. In our app we have three different services & all of them has (`REST` & `GraphQL`) support.

- `api`
- `shop`
- `admin`
 - `rest`
 - `graphql`

Admin Dashboard (GraphQL & REST)

- [NextJS](#)
- [TypeScript](#)
- [Tailwind CSS](#)
- [React Hook Form](#)
- [Apollo Client](#)(For GraphQL API version)
- [React-Query](#)(For REST API version)

Shop Frontend (GraphQL & REST)

- [NextJS](#)
- [TypeScript](#)
- [Tailwind CSS](#)
- [React Hook Form](#)
- [Apollo Client](#)(For GraphQL API version)
- [React-Query](#)(For REST API version)
- [Next SEO](#)
- [Next PWA](#)

API

- Laravel

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

Getting Started

In this chapter we are going to show you the installation process of this application on different operating system.

Folder Structure

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

```
pickbazar
|-- api

|-- admin
    |-- rest
    |-- graphql

|-- shop
```

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

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

Prerequisites for Backend

- 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

Prerequisites for Frontend

- node(16.15.1 or later)
- `yarn` or `npm` or `pnpm`
- editor: [Visual Studio Code](#)(recommended, but choice is up to you)

Project initialization

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

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

Getting started with Windows

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

- Make sure you have run xamp/mamp/wamp/lamp for mysql and php by following the steps below:
Click on action button start to make it start.



Xamp has been started.

Module	Service	PID(s)	Port(s)	Actions
Apache		2120	80, 443	Stop
MySQL		3272	3306	Stop
FileZilla				Start
Mercury				Start
Tomcat				Start

```

4:03:13 PM [main] Checking for prerequisites
4:03:16 PM [main] All prerequisites found
4:03:16 PM [main] Initializing Modules
4:03:16 PM [Apache] XAMPP Apache Service is already running on port 80
4:03:16 PM [Apache] XAMPP Apache Service is already running on port 443
4:03:16 PM [mysql] XAMPP MySQL Service is already running on port 3306
4:03:16 PM [main] Starting Check-Timer
4:03:16 PM [main] Control Panel Ready
  
```

- Go to localhost/phpmyadmin

localhost/phpmyadmin/

- Click on new to create new database.

New

- information_schema
- mysql
- performance_schema
- phpmyadmin
- pickbazar
- test

General settings

Server connection collation: utf8mb4_unicode_ci

Appearance settings

Language: English

Web server

- Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.0.25
- Database client version: libmysql - mysqlnd 8.0.26
- PHP extension: mysqli curl mbstring
- PHP version: 8.0.25

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

Create database

pickbazar_laravel

utf8mb4_general_ci

Create

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

Specially check for this `env` variables

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

- Run `composer install` using terminal

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

Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: barryvdh/laravel-dompdf
Discovered Package: bensampo/laravel-enum
Discovered Package: cviebrock/eloquent-sluggable
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: fruitcake/laravel-cors
Discovered Package: ignited/laravel-omnipay
Discovered Package: intervention/image
Discovered Package: laravel/legacy-factories
Discovered Package: laravel/sail
Discovered Package: laravel/sanctum
Discovered Package: laravel/tinker
Discovered Package: 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 (79.79ms)
Migrated: 2014_10_12_000000_create_users_table (79.79ms)
Migrating: 2014_10_12_100000_create_password_resets_table (58.57ms)
Migrated: 2014_10_12_100000_create_password_resets_table (58.57ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table (79.56ms)
Migrated: 2019_08_19_000000_create_failed_jobs_table (79.56ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table (143.06ms)
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (143.06ms)
Migrating: 2020_04_17_194830_create_permission_tables (839.79ms)
Migrated: 2020_04_17_194830_create_permission_tables (839.79ms)
Migrating: 2020_06_02_051901_create_pickbazar_tables (1,601.56ms)
Migrated: 2020_06_02_051901_create_pickbazar_tables (1,601.56ms)
Migrating: 2020_10_26_163529_create_media_table (71.72ms)
Migrated: 2020_10_26_163529_create_media_table (71.72ms)

Tables Migration completed.

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

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

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

Copying resources files...
Installation Complete

```

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

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

- run `php artisan serve`

```

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

```

*NB: your frontend `NEXT_PUBLIC_GRAPHQL_API_ENDPOINT` env value will be
`localhost:8000/graphq1`*

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

For MAC and Linux(with sail and docker)

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

Prerequisites

- Docker

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

Installation Mac

Video

REST API

<https://www.youtube.com/embed/xkkGC448d1g>

Follow this tutorial for install REST API

GraphQL API

<https://www.youtube.com/embed/arwEWD4CCxA>

Follow this tutorial for install GraphQL

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

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

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

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

Getting Started with Frontend

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

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

```
- Go to your specific project root and find `graphql-let.yml`.  
- To find the file go to `admin` => `graphql` => `graphql-let-yml`  
- Replace the env variable `${NEXT_PUBLIC_GRAPHQL_API_ENDPOINT}` in  
schema field manually  
- Provide your API url in the schema field.  
  
"schema: ${NEXT_PUBLIC_GRAPHQL_API_ENDPOINT}",  
to  
"schema: https://localhost:8000/graphql",  
  
- Also change the dev command at `package.json`  
- Go to `shop/package.json` file and change the scripts:  
  
"codegen": "node -r dotenv/config $(yarn bin)/graphql-let",  
to  
"codegen": "graphql-let",
```

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

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

After choosing your working directory

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

5. Run yarn at the root directory.

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

6. Scripts To Run the fronted App

For Admin :

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

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

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

# REST
yarn dev:admin-rest
```

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

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

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

# REST
yarn dev
```

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

** Note: **

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

For Shop :

Shop config

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

At first open `pickbazar -> shop`,

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

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

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

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

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

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

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

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

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

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

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

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

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

```

► yarn build:shop-rest
yarn run v1.22.5
$ yarn workspace @pick-bozor/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]
    L css/7936e7aa084bd830e87c.css
    L grocery
    L makeup
    L bags
    [+2 more paths]
  o /404
  • /bakery (ISR: 60 Seconds)
    L css/8dcfb7860d56385c547.css
    λ /change-password
  o /checkout
  o /contact
  o /example
  o /help
  o /logout
  o /offers
  λ /order
    L css/db04274d0e1ca011303a.css
  λ /order-received/[tracking_number]
  λ /orders
    L css/d2daebf602ff72a3fc0.css
  o /privacy
  • /products/[slug]
    L css/efb94a24c22d59f4d562.css
    L products/apples
    L products/baby-spinach
    L products/blueberries
    [+27 more paths]
  λ /profile
  o /terms
  L o /ui
    8.82 kB   185 kB
+ First Load JS shared by all
  176 kB
  chunks/00682edc09922f9a0564b6d51ebfe3fe9fe3f6bc.837c2b.js   3.14 kB
  chunks/03597eb8dee95bb8e6991ab21a5b170a1bc330ab.CSS.3bbe0c.js  68 B
  chunks/1c2831c8.c533da.js                                         23.5 kB
  chunks/2c96be89bf36e1896d67e23c3a04c26631eb28.f78ccb.js     4.76 kB
  chunks/57ad4ee95d685fa85f031607d3121ddc6f6f331.d3ef7d.js    35.8 kB
  chunks/6bc2a5f5c5fb5a7770dd80e4657cc465fa5ca.173fc5.js     7.14 kB
  chunks/78478e0567901f0c00137009c7f6751fb214ebc.855475.js    15.3 kB
  chunks/ad9552d4f74729e02303abd0a7d42186859f2ec0.37727a.js  7.43 kB
  chunks/commons.ff7e9e.js                                         18.2 kB
  chunks/framework.f6e4f9.js                                       42.3 kB
  chunks/main.9c8e07.js                                         6.29 kB
  chunks/pages/_app.558fd4a.js                                    9.76 kB
  chunks/webpack.3d3782.js                                       2.3 kB
  css/669ac54f0d96e08dbac5.css                                 4.38 kB
  css/f53121b36d6036aac827.css                                11.2 kB

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

★ Done in 96.20s.
► yarn start:shop-rest
yarn run v1.22.5
$ yarn workspace @pick-bozor/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](#).

For Shop :

Shop config

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

At first open `pickbazar -> shop`,

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

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

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

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

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

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

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

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

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

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

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

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

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

```

► yarn build:shop-rest
yarn run v1.22.5
$ yarn workspace @pick-bozor/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]
    L css/7936e7aa084bd830e87c.css
    L grocery
    L makeup
    L bags
    [+2 more paths]
  o /404
  • /bakery (ISR: 60 Seconds)
    L css/8dcfb7860d56385c547.css
    λ /change-password
    o /checkout
    o /contact
    o /example
    o /help
    o /logout
    o /offers
    λ /order
      L css/db04274d0e1ca011303a.css
    λ /order-received/[tracking_number]
    λ /orders
      L css/d2daebf602ff72a3fc0.css
    o /privacy
  • /products/[slug]
    L css/efb94a24c22d59f4d562.css
    /products/apples
    /products/baby-spinach
    /products/blueberries
    [+27 more paths]
  λ /profile
  o /terms
  L o /ui
    8.82 kB   185 kB
+ First Load JS shared by all
  176 kB
  chunks/00682edc09922f9a0564b6d51ebfe3fe9fe3f6bc.837c2b.js   3.14 kB
  chunks/03597eb8dee95bb8e6991ab21a5b170a1bc330ab.CSS.3bbe0c.js  68 B
  chunks/1c2831c8.c533da.js                                         23.5 kB
  chunks/2c96be89bf36e1896607e23c3a04c26631eb28.f78ccb.js     4.76 kB
  chunks/57ad4ee95d685fa85f031607d3121ddc6f6f331.d3ef7d.js    35.8 kB
  chunks/6bc2a5f5c5fb5a7f770dd80e4657cc465fa5ca.173fc5.js    7.14 kB
  chunks/78478e0567901f0c00137009c7f6751fb214ebc.855475.js   15.3 kB
  chunks/ad9552d4f74729e02303abd0a7d42186859f2ec0.37727a.js  7.43 kB
  chunks/commons.ff7e9e.js                                         18.2 kB
  chunks/framework.f6e4f9.js                                       42.3 kB
  chunks/main.9c8e07.js                                         6.29 kB
  chunks/pages/_app.558fd4a.js                                    9.76 kB
  chunks/webpack.3d3782.js                                       2.3 kB
  css/669ac54f0d96e08dbac5.css                                 4.38 kB
  css/f53121b36d6036aac827.css                                11.2 kB

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

★ Done in 96.20s.
► yarn start:shop-rest
yarn run v1.22.5
$ yarn workspace @pick-bozor/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](#).

Common How to use

For Docker

For Docker installation in local PC check our [Docker Installation](#) Part.

For Generalize installation in local PC

For Generalize installation in local PC check our [Windows Installation](#) Part.

PHP Artisan Oriented Scripts

Available Scripts:

```
"php artisan marvel:install": "Installing Marvel Dependencies",
"php artisan marvel:create-admin": "Create an Admin",
"php artisan marvel:seed": "Import Demo Data",
"php artisan marvel:settings_seed": "Import Settings Demo Data",
"php artisan marvel:copy-files": "Copy necessary files",
```

Sail Oriented Scripts

Available Scripts:

```
"sail artisan marvel:install": "Installing Marvel Dependencies",
"sail artisan marvel:create-admin": "Create an Admin",
"sail artisan marvel:seed": "Import Demo Data",
"sail artisan marvel:settings_seed": "Import Settings Demo Data",
"sail artisan marvel:copy-files": "Copy necessary files",
```

Most uses Laravel commands.

```
"php artisan serve": "To start project",
"php artisan config:cache": "change environmental variables",
"php artisan key:generate": "Generate new application key",
"php artisan migrate": "If you want to migrate the database
tables",
"php artisan storage:link": "Create a symbolic link in Laravel
application.",
"composer require vendor_name/Package_name": "Install New packages
in Laravel application",
"composer update": "Update all your packages or specific package",
```

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

** Note: ** Also, individual Scripts are available under every individual package.

You can check out them from there individual package.json file.

Available Scripts:

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

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

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

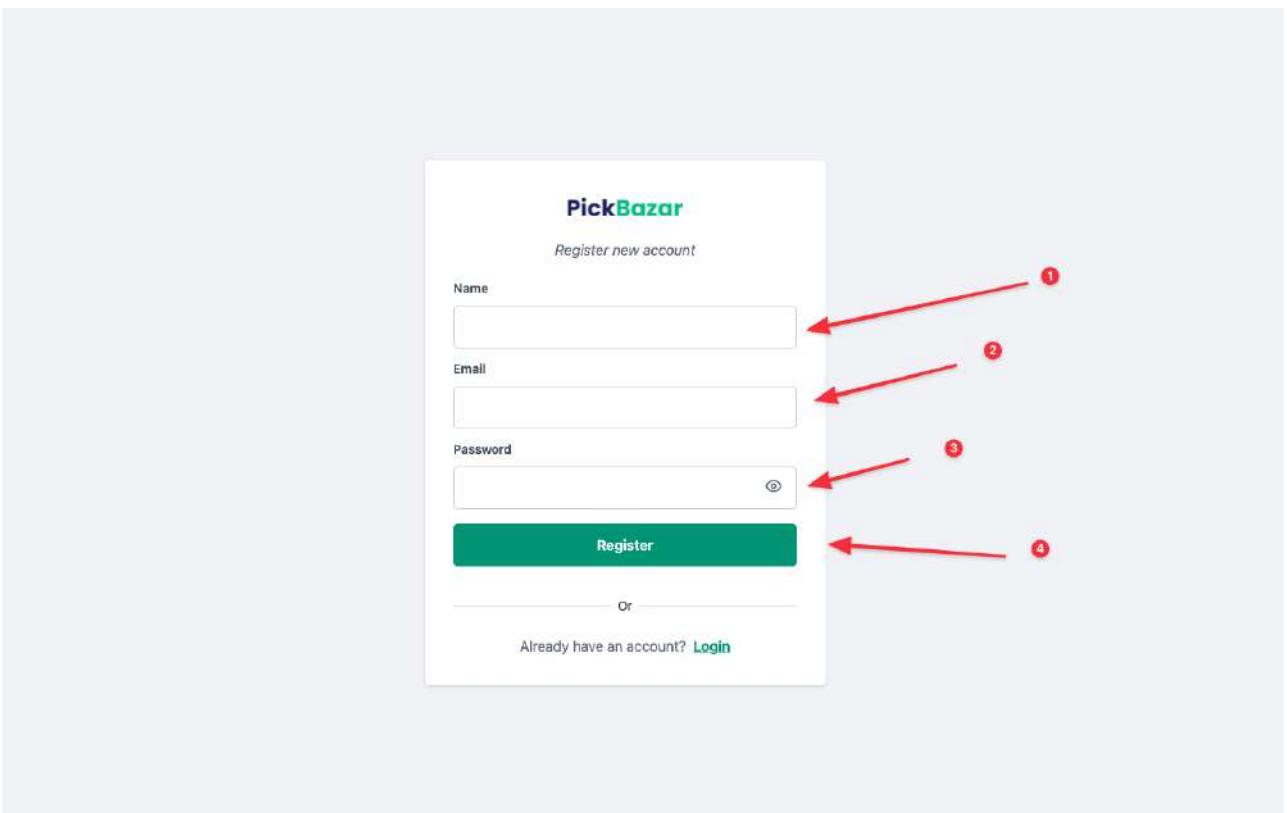
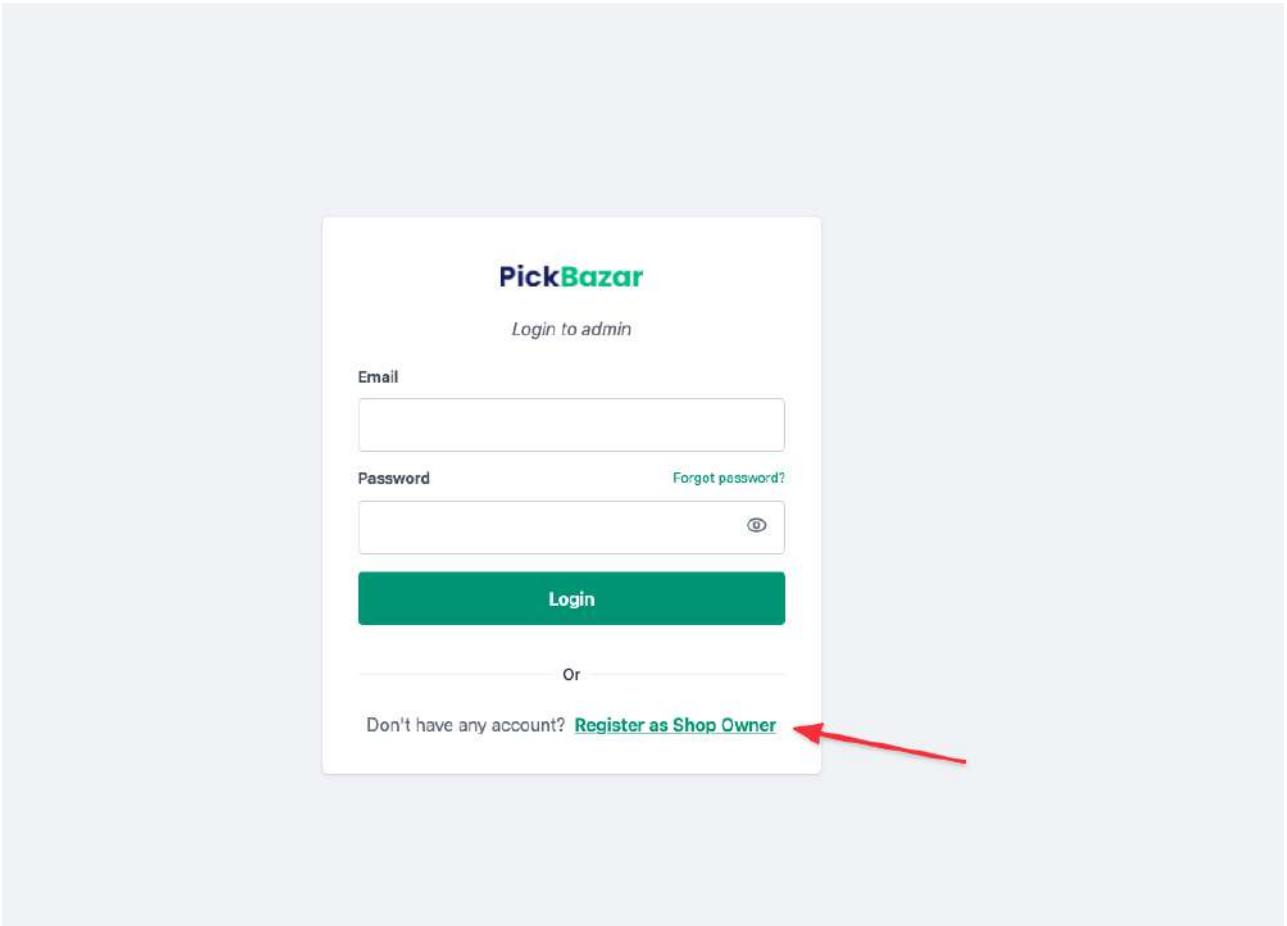
Multi-Vendor

Multi-vendor is one of the Key feature of this Pickbazar application. In this chapter we are going to discuss about how you can make a use of this application in our given multi-vendor approach.

During installation a super admin need to be created. So, you can easily check that out from installation part. Now the second part is creating a vendor and shop.

Create New Shop

To create new shop login as an `shop owner` or register a new account for creating `shop`



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

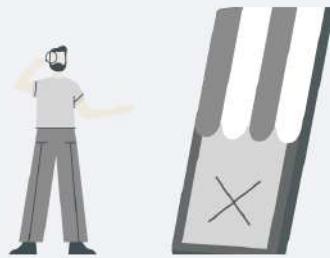


Vendor

vendor@example.com
Please add your [Profile](#)

Enabled

My Shops



No Shops Found



After that, click **Create Shop**



Vendor

vendor@example.com
Please add your [Profile](#)

Enabled

Create Shop

Logo

Upload your shop logo from here.

 Upload an Image or drag and drop.
IMG, 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.
IMG, JPG

Basic Info

Add some basic info about your shop from here.

Name

Description

Payment Info

Add your payment information from here.

Account Holder Name

localhost:3002

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

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



Vendor

vendor@example.com
Please add your [Profile](#)

Enabled

My Shops



Grocery Shop 2

Inactive



localhost:3002/grocery-shop-2

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

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

Dashboard

 Shops^①

My Shops

Products

Attributes

Groups

Categories

Tags

Orders

Order Status

Users

Coupons

Taxes

Shipments

Withdraws

Settings



Shops

Type your query and press enter

Logo	Name	Owner Name	Products	Orders	Status	Actions
	Grocery Shop 2 ^②	Vendor	0	0	Inactive	
	Bags Shop	Store Owner	15	0	Active	
	Bakery Shop	Store Owner	72	0	Active	
	Clothing Shop	Store Owner	64	0	Active	
	Grocery Shop	Store Owner	249	0	Active	
	Makeup Shop	Store Owner	81	0	Active	
	Furniture Shop	Store Owner	55	0	Active	

< 1 >

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



Vendor

vendor@example.com
Please add your [Profile](#)

Enabled

My Shops



Grocery Shop 2

Active



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

- Dashboard
- Attributes
- Products
- Orders
- Staffs
- Withdraws



Grocery Shop 2

The grocery shop is the best shop around the city. This is being run by...

[Read more](#)

📍 1986 Spinnaker Lane, Freeport, Illinois, 61032, USA, New York, New York, 10001, USA

📞 No contact number available

[Visit Shop](#)[Edit Shop](#)

Products

 0
Total Products

Revenue

 \$0.00
Gross Sales

Others

 15 %
Admin Commission Rate
Registered Since
July 3, 2021

Total Orders

 0
Total Orders

 \$0.00
Current Balance

Payment Information

Name:
Mahmud
Email:
vendor@example.com

From this dashboard, you can maintain the shop,

Dashboard:

PickBazar

Dashboard ←

Attributes Products Orders Staffs Withdraws

Grocery Shop 2

The grocery shop is the best shop around the city. This is being run ...
[Read more](#)

1986 Spinaker Lane, Freeport, Illinois, 61032, USA, New York, New York, 10001, USA
No contact number available

[Visit Shop](#)

Products

0 Total Products	\$0.00 Gross Sales	15 % Admin Commission Rate
0 Total Orders	\$0.00 Current Balance	

Revenue

Others

Registered Since
July 9, 2021

Payment Information

Name: Mahmud
Email: vendor@example.com



Attributes:

PickBazar

Dashboard ←

Attributes Products Orders Staffs Withdraws

Attributes

[+ Add Attribute](#)

ID	Name	Values	Actions
No data found			



Products:

PickBazar

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Products

Type your query and press enter

+ Add Product

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

Order:

PickBazar

Dashboard

Attributes

Products

Orders

Staffs

Withdraws

Orders

Type your query and press enter

Tracking Number	Delivery Fee	Total	Order Date	Status	Shipping Address	table-item-download	Actions
No data found							

Staff:

PickBazar

Dashboard Attributes Products Orders Staffs Withdraws

Staff

Name Email Status Actions

No data found

+ Add Staff

WithDraw:

PickBazar

Dashboard Attributes Products Orders Staffs Withdraws

Withdraws

Shop Name Amount Status Created

No data found

+ Request Withdraw

User Roles:

Super Admin:

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

Store Owner:

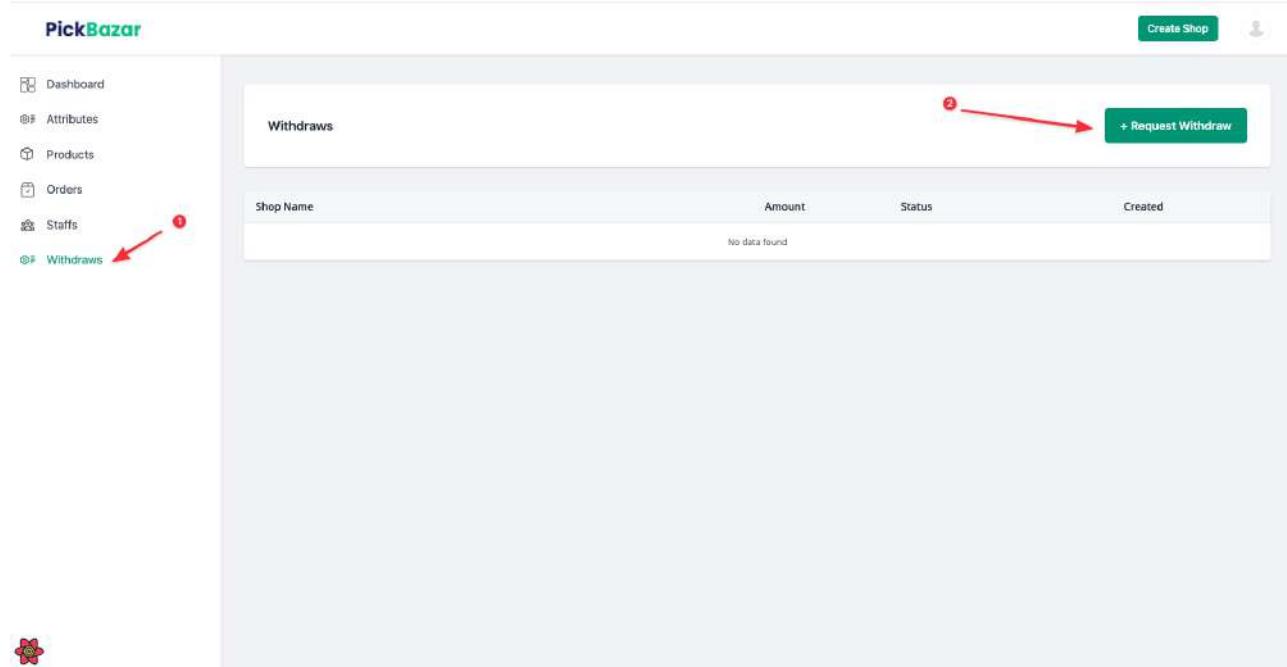
Store owner can edit or main it's store, staff or payment.

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,



- Dashboard
- Attributes
- Products
- Orders
- Staffs
- Withdraws

Create Withdraw

Description
Add withdraw request from here

①	Amount	<input type="text"/>
②	Payment Method	<input type="text"/>
③	Details	<input type="text"/>
④	Note	<input type="text"/>

⑤ → Request Withdraw



After request payment, the dashboard will be like this,

- Dashboard
- Attributes
- Products
- Orders
- Staffs
- Withdraws

Withdraws

+ Request Withdraw

Shop Name	Amount	Status	Created
Grocery Shop 2	\$500.00	Pending	a few seconds ago



< 1 >



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

- Dashboard
- Shops
- My Shops
- Products
- Attributes
- Groups
- Categories
- Tags
- Orders
- Order Status
- Users
- Coupons
- Taxes
- Shipments
- Withdraws
- Settings

Withdraws

Shop Name	Amount	Status	Created	Actions
Grocery Shop 2	\$500.00	Pending	a minute ago	



- Dashboard
- Shops
- My Shops
- Products
- Attributes
- Groups
- Categories
- Tags
- Orders
- Order Status
- Users
- Coupons
- Taxes
- Shipments
- Withdraws
- Settings

Withdrawal Information

①

②

Amount	<input type="text" value="500"/>	Status	Pending	
Payment Method	hand	Approved		
Status	Pending	On Hold		
Details:		Processing		
test		Pending		
		Rejected		

①

②



After approved,

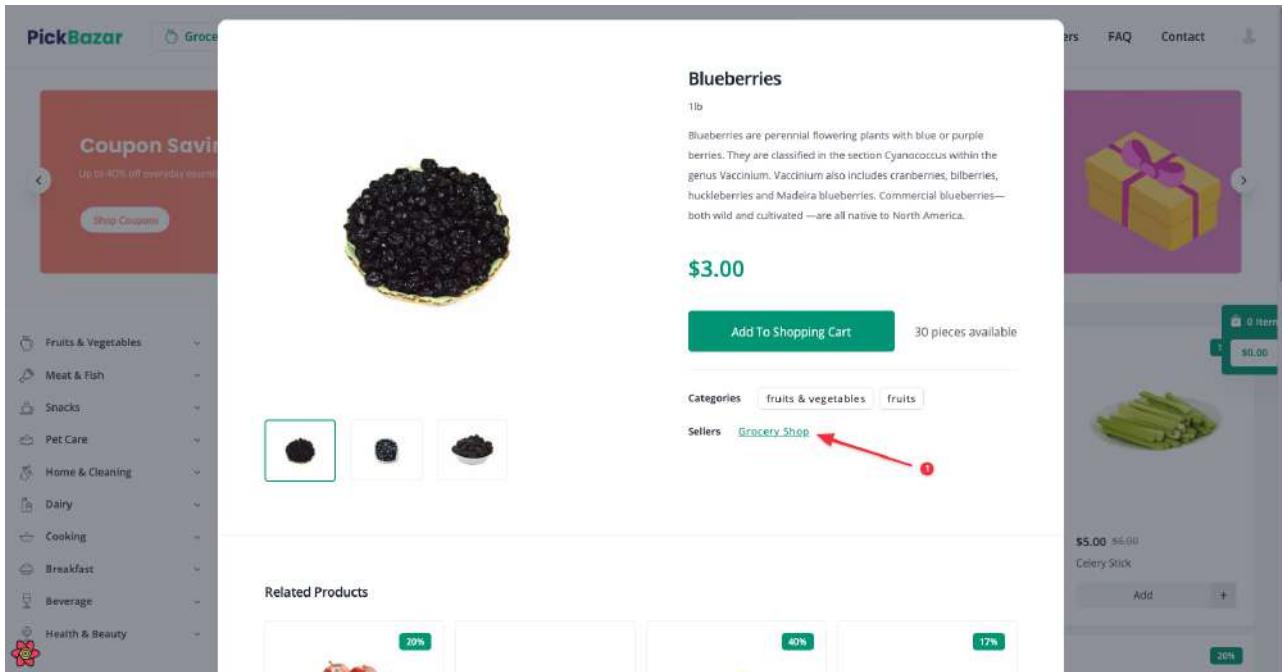
The screenshot shows the 'Withdraws' section of the PickBazar admin dashboard. On the left sidebar, there are links for Dashboard, Attributes, Products, Orders, Staffs, and Withdraws. The main content area is titled 'Withdraws' and contains a table with one row. The table columns are 'Shop Name', 'Amount', 'Status', and 'Created'. The data row shows 'Grocery Shop 2' with '\$500.00', 'Approved' status, and '3 minutes ago' created time. A green button at the top right says '+ Request Withdraw'.

FrontEnd Shop

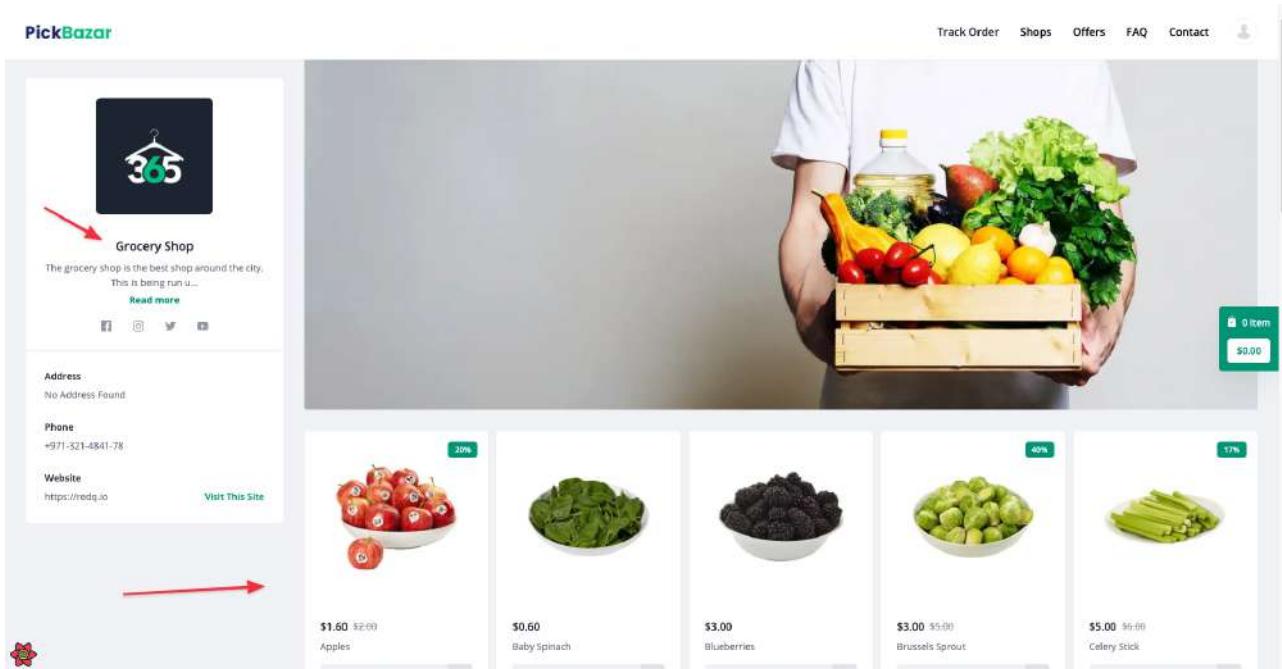
From frontend when customers click on the `shop` page they'll get all the shops as a card list,

The screenshot shows the 'All Shops' page of the PickBazar frontend. At the top, there is a navigation bar with links for Track Order, Shops (which has a red notification badge with the number 1), Offers, FAQ, Contact, and a user profile icon. Below the navigation bar, there are category tabs: Grocery, Bakery, Makeup, Bags, Clothing, and Furniture. The main content area is titled 'All Shops' and displays six shop cards in a 2x3 grid. Each card includes a small icon, the shop name, and a brief address. The cards are: Furniture Shop (Furniture icon), Clothing Shop (Clothing icon), Bags Shop (Bags icon); Makeup Shop (Makeup icon), Bakery Shop (Bakery icon), and Grocery Shop (Grocery icon). A red arrow points to the 'Shops' link in the navigation bar.

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



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



Super admin

The super admin role in an e-commerce site holds ultimate control. They manage users, permissions, and critical settings, ensuring seamless operations and security at the highest level.

As, super admin has the supreme authority over any vendor, so he/she can control any listed shops in the application. Here are the key notes for super admin,

- Super admin can see access all settings & menu.

- Super admin is also a vendor, staff, customer himself/herself. So, he/she can listed his own shop & control them as a vendor/staff/customer perspective.

ID	Shop	Products	Orders	Owner Name	Status	Actions
#ID: 7	Books Shop	87	5	Store Owner	Active	Edit Delete
#ID: 6	Grocery Shop	584	4	Store Owner	Active	Edit Delete
#ID: 5	Bakery Shop	72	0	Store Owner	Active	Edit Delete
#ID: 4	Makeup Shop	81	0	Store Owner	Active	Edit Delete
#ID: 3	Bags Shop	15	0	Store Owner	Active	Edit Delete
#ID: 2	Clothing Shop	64	1	Store Owner	Active	Edit Delete
#ID: 1	Furniture Shop	55	0	Store Owner	Active	Edit Delete

Vendor or Store Owner

Vendors are generally the shop owner. A vendor can register in the application as a seller. And he/she can open a shop based on agreement with super admin. Some key notes for Vendors,

- Vendor can listed his/her own products.
- Vendor can assign staffs for his/her shop.
- Vendor can monitor his orders, based on his provided order status, super admin will take rest actions.
- Vendor can set his/her own FAQs based on his shop individually. For example, a Grocery shop's FAQ may not be similar with a Gadget shop's FAQ.
- Vendor can set shop specific Terms & Conditions, if and only if Super admin gave permission.

The screenshot shows the PickBazar Laravel admin dashboard. On the left, there's a sidebar with various menu items: Main (Dashboard), Product Management (Products, Inventory, Attributes, Manufacturers/Publications, Authors), Financial Management (Withdrawals, Refunds), Order Management (Orders, Transactions), Features Management (Store Notice, Message), Feedback Control (Reviews, Questions), User Control (Staff), Promotional Control (Flash sale), Layout/Page Management (FAQs), and a footer note about RedQ. The main content area features a large image of a person holding a wooden crate filled with fresh produce. Below the image is a circular logo with '365' and a hanger icon. The shop information is displayed: 'Grocery Shop' (with a bio link), email 'store_owner@demo.com', address '1986 Spinnaker Lane, Freeport, Illinois, 61032, USA', and an 'Edit Shop' button. Key statistics are shown in cards: 584 Total Products, 4 Total Orders, 10% Admin Commission Rate, \$0.00 Gross Sales, and \$0.00 Current Balance.

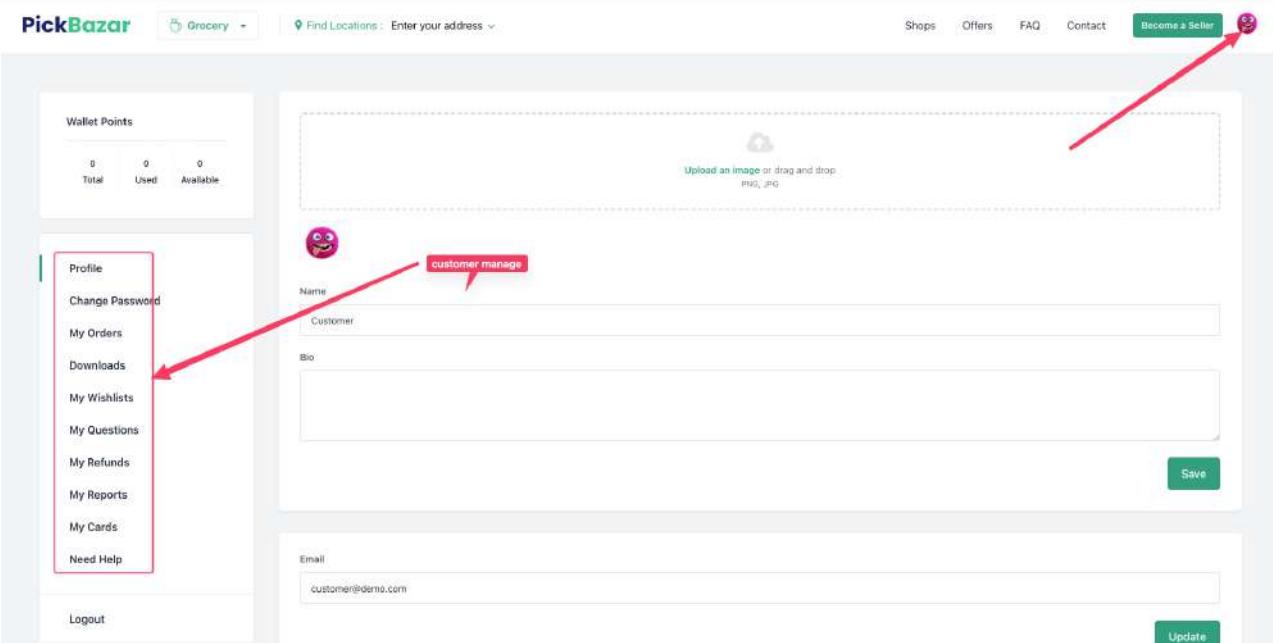
Staff

The staff role in an e-commerce website typically involves managing day-to-day operations. Staff members assist customers, process orders, update product listings, and ensure smooth website functionality. They may also handle customer inquiries, resolve issues, and collaborate with vendors and other team members to maintain a seamless shopping experience for customers.

Customer

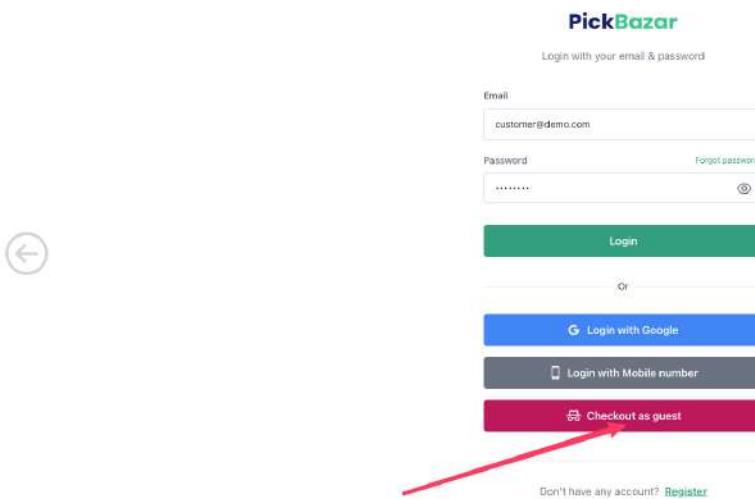
Login Customer

The customer role in PickBazar Laravel represents the end-users who visit the site to browse, select, and purchase products or services. Customers create accounts, place orders, make payments, and provide feedback.



Guest Customer

A guest user in e-commerce refers to someone who visits an online store and makes purchases without creating an account or logging in. PickBazar Guest users can browse products, add items to their cart, provide shipping and payment information, and complete a purchase without the need for a registered account. This streamlined process is convenient for those who prefer not to create accounts or for first-time shoppers.



Introduction to Features

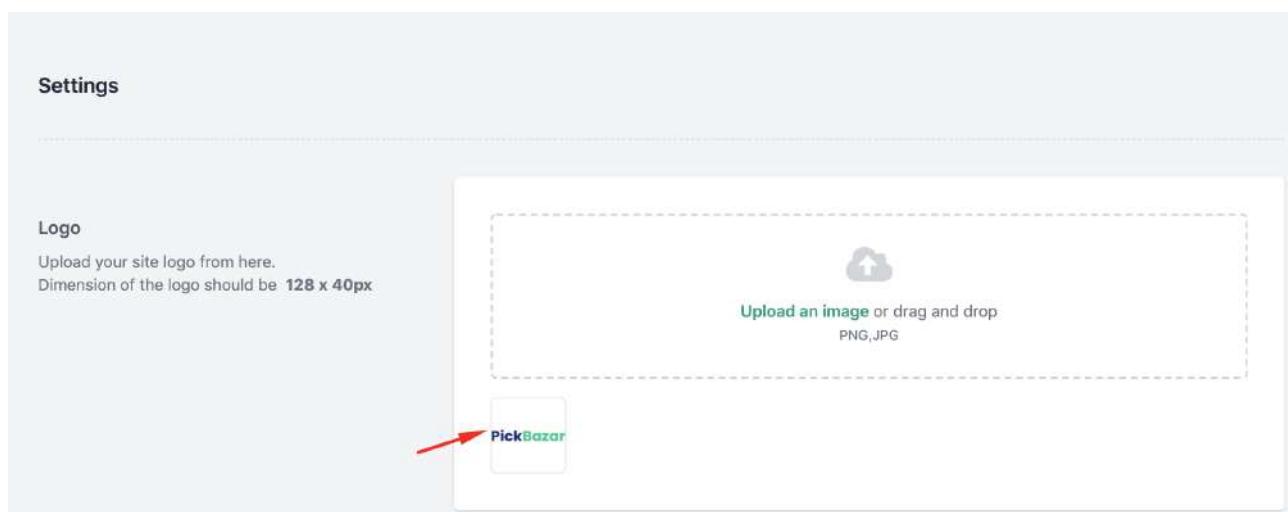
In this chapter we will provide an overview of all the available features presented in the application. As it's a multivendor application, so it has multiple user roles. (e.g. Super admin, Vendor, Staff, Customer)

So, if we cover the Super admin role, then most of the feature will be covered for other user roles too.

Let's deep dive into the feature chapter.

Settings Management

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



Information

Change your site information from here

Site Title	Pickbazar
Site Subtitle	Your next ecommerce
Currency	US Dollar
Tax Class	Global
Shipping Class	Global

Save Settings

International Slug

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

if I write name/title as **Books**, **Books store** and **we live in a beautiful world.** the slug will be look like

For English:

Books Slug: **books**

Books store Slug: **books-store**

we live in a beautiful world. Slug: **we-live-in-a-beautiful-world.**

For Arabic:

كتاب Slug: **كتاب**

مكتبة لبيع الكتب Slug: **مكتبة-لبيع-الكتب**

أوي تعيش في عالم جميل Slug: **أوي-تعيش-في-عالم-جميل**

For Korean:

책 Slug: 책

서점 Slug: 서점

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

For Japanese:

本 Slug: 本

本屋 Slug: 本屋

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

For Chinese:

書 Slug: 書

書店 Slug: 書店

阿伊生活在一個美麗的世界裡。 Slug: 阿伊生活在一個美麗的世界裡。

For Turkish:

kitap Slug: kitap

kitapçı Slug: kitapçı

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

For Telugu:

పుస్తకం Slug: పుస్తకం

పుస్తకాల దుకాణం Slug: పుస్తకాల-దుకాణం

ఒ అందమైన ప్రపంచంలో జీవిస్తున్నా . Slug: ఒ-అందమైన-ప్రపంచంలో-జీవిస్తున్నా .

For Spanish:

libros Slug: libros

tienda de libros Slug: tienda-de-libros

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

For Bengali:

বই Slug: বই

বইয়ের দোকান Slug: বইয়ের-দোকান

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

Introduction to Analytics

For a modern e-commerce application, analytics is a must needed tool. In Pickbazar, we have already some basic analytical widgets. But we are working on to add more widgets day by day.

Here is the lists of some analytical widgets presented currently.

- Basic Summary
- Order status widgets
- Recent orders
- Sale history
- Popular products
- Low stock products
- Top 10 most rated products
- Top 10 categories with most products.
- Recent withdrawal

MAIN

Dashboard

SHOP MANAGEMENT

Shops
My Shops

PRODUCT MANAGEMENT

Products
Inventory
Categories
Tags
Attributes
Manufacturers/Publications
Authors

E-COMMERCE MANAGEMENT

Taxes
Shipments
Withdrawals
Refunds
Home pages / Types
FAQs
Terms And Conditions

ORDER MANAGEMENT

Orders
Create Order
Transactions
All users
Admin list
Vendors
Staff
Customers

FEEDBACK CONTROL

Reviews
Questions

PROMOTIONAL MANAGEMENT

Coupons
Flash sale

FEATURE MANAGEMENT

Message
Store Notice

SITE MANAGEMENT

Settings

Summary

	Total Revenue \$0.00		Total Order 5		Vendor 2		Total Shops 7
--	--------------------------------	--	-------------------------	--	--------------------	--	-------------------------

Order Status

Pending Order	Processing Order	Completed Order	Cancelled Order
0	5	0	0

Today Weekly Monthly Yearly

Recent Orders

Tracking Number	Customer	Products	Order Date	Total	Status	Actions
20230920612904	Customer customer@demo.com	2	3 hours ago	\$53.12	Processing	
2023092050274	Customer customer@demo.com	3	4 hours ago	\$70.90	Processing	
20230920131981	Customer customer@demo.com	3	4 hours ago	\$70.96	Processing	
20230920504468	Customer customer@demo.com	3	4 hours ago	\$60.78	Processing	
20230920813800	Customer customer@demo.com	3	4 hours ago	\$54.75	Processing	

1 of 2 pages

Sale History



Popular Products

	Blueberries	Grocery	\$2.55
	Baby Spinach	Grocery	\$0.51
	Brussels Sprout	Grocery	\$3.00
	Celery Stick	Grocery	\$5.00
	The Bedtime Stories Part One	Books	\$100.00
	Apples	Grocery	\$1.60

Low stock products

Product ID	Product	SKU	Group	Shop	Price/Unit	Stock status	Quantity
#ID: 2	Baby Spinach	2	Grocery	Grocery Shop	\$0.60		8

Top 10 Most Sold Products



Strawberry
The garden strawberry is a widely grown hybrid species...

\$8.00

Top 10 Category with most products

Category ID	Category Name	Shop	Product Count
#ID: 7	Snacks	Grocery Shop	73
#ID: 125	Bakery	Grocery Shop	61
#ID: 122	Fruits	Grocery Shop	60
#ID: 118	Snacks	Grocery Shop	54
#ID: 117	Vegetables	Grocery Shop	47
#ID: 121	Meat	Grocery Shop	40
#ID: 72	Outer Wear	Clothing Shop	33
#ID: 124	Dairy & Egg	Grocery Shop	28
#ID: 123	Fish	Grocery Shop	26
#ID: 43	Health & Beauty	Grocery Shop	26

Recent Withdrawals

Shop ID	Shop	Amount	Created	Payment Method	Status	Actions
---------	------	--------	---------	----------------	--------	---------



Introduction to Shop

No data found
Sorry we couldn't found any data.



©2023 Pickbazar. All right reserved RedQ

10.2.0

In Pickbazar application shop is one the most important features need to understand. Based on the user role shop feature can be divided into 3 parts. Lets discuss is here.

Super Admin

As, super admin has the supreme authority over any vendor, so he/she can control any listed shops in the application. Here are the key notes for super admin,

- Super admin can see all the published & inactive shops.
- During shop approval super admin can set the commission rate for that shop.
- Super admin can control actions for approval & disapproval of any shop.
- Super admin has access to visit every shop's own dashboard & inner pages.
- Super admin is also a vendor himself/herself. So, he/she can listed his own shop & control them as a vendor perspective.

The screenshot shows the Pickbazar Admin Dashboard. The left sidebar has sections for MAIN (Dashboard), SHOP MANAGEMENT (Shops, All shops, Add new shop, Inactive/New shops, My Shops), PRODUCT MANAGEMENT (Products, Inventory, Categories, Tags, Attributes, Manufacturers/Publications, Authors), and E-COMMERCE MANAGEMENT (Taxes). The main content area is titled 'Shops' and lists seven entries:

ID	Shop	Products	Orders	Owner Name	Status	Actions
#ID: 7	Books Shop	67	5	Store Owner	Active	⋮ ⊕
#ID: 6	Grocery Shop	564	4	Store Owner	Active	⋮ ⊕
#ID: 5	Bakery Shop	72	0	Store Owner	Active	⋮ ⊕
#ID: 4	Makeup Shop	81	0	Store Owner	Active	⋮ ⊕
#ID: 3	Bags Shop	15	0	Store Owner	Active	⋮ ⊕
#ID: 2	Clothing Shop	64	1	Store Owner	Active	⋮ ⊕
#ID: 1	Furniture Shop	55	0	Store Owner	Active	⋮ ⊕

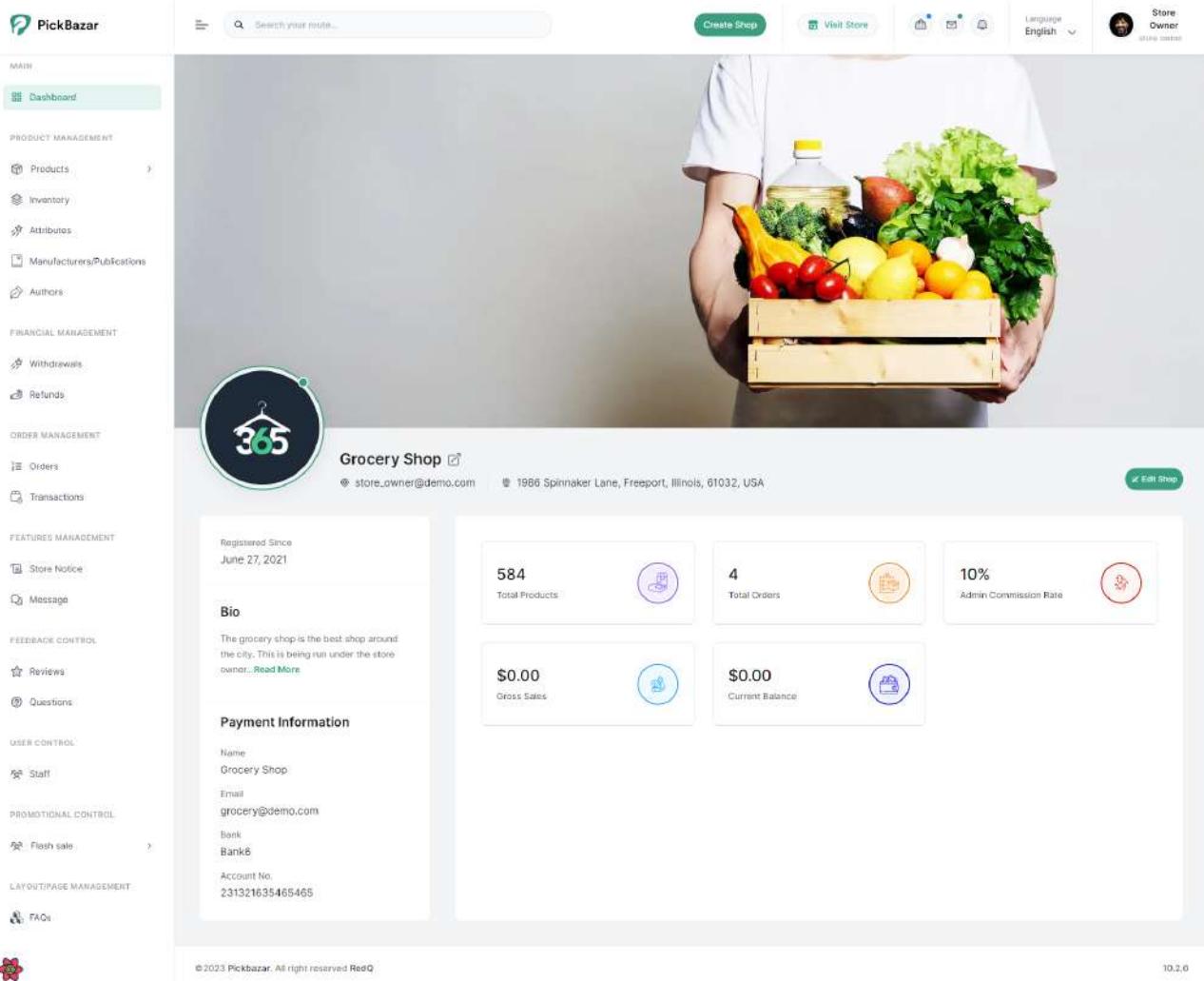
Vendor

Vendors are generally the shop owner. A vendor can register in the application as a seller. And he/she can open a shop based on agreement with super admin.

Some key notes for Vendors,

- Vendor can listed his/her own products.

- Vendor can assign staffs for his/her shop.
- Vendor can monitor his orders, based on his provided order status, super admin will take rest actions.
- Vendor can set his/her own FAQs based on his shop individually. For example, a Grocery shop's FAQ may not be similar with a Gadget shop's FAQ.
- Vendor can set shop specific Terms & Conditions, if and only if Super admin gave permission.



The screenshot shows the PickBazar dashboard interface. On the left, there is a sidebar with various management sections: Dashboard, PRODUCT MANAGEMENT (Products, Inventory, Attributes, Manufacturers/Publications, Authors), FINANCIAL MANAGEMENT (Withdrawals, Refunds), ORDER MANAGEMENT (Orders, Transactions), FEATURES MANAGEMENT (Store Notice, Message), FEEDBACK CONTROL (Reviews, Questions), USER CONTROL (Staff), PROMOTIONAL CONTROL (Flash sale), LAYOUT/PAGE MANAGEMENT (FAQs), and a footer section with a flower icon and copyright information. The main content area features a large image of a person holding a wooden crate filled with fresh vegetables and fruits. Below the image, the shop's profile is displayed: **Grocery Shop**, store_owner@demo.com, Registered Since June 27, 2021. The profile includes sections for Bio (describing it as the best shop around the city), Payment Information (Name: Grocery Shop, Email: grocery@demo.com, Bank: BankB, Account No: 231321635465465), and performance metrics: 584 Total Products, 4 Total Orders, 10% Admin Commission Rate, \$0.00 Gross Sales, and \$0.00 Current Balance. At the top right, there are buttons for Create Shop, Visit Store, and Edit Shop, along with language settings (English) and a user account for the store owner.

Staff

Staffs are assigned by vendor. A shop staff's main duty is to act as a helping hand to vendor, but his/her has some limitations in permission. Some key notes for Staff role,

- One shop staff has no access to another shop of a same vendor.
- Staffs can add FAQs to his/her assigned shop.

The screenshot shows the PickBazar dashboard interface. On the left, there's a sidebar with navigation links for MAIN, PRODUCT MANAGEMENT, FINANCIAL MANAGEMENT, ORDER MANAGEMENT, FEATURES MANAGEMENT, FEEDBACK CONTROL, USER CONTROL, PROMOTIONAL CONTROL, and LAYOUT/PAGE MANAGEMENT. The main content area features a large image of a person holding a wooden crate filled with fresh fruits and vegetables. Below this is a profile card for a store named 'Grocery Shop'. The card includes a circular logo with '365' and a hanger icon, the store name, an email address (store_owner@demo.com), and an address (1986 Spinnaker Lane, Freeport, Illinois, 61032, USA). It also displays statistics: 'Registered Since June 27, 2021', '584 Total Products' (with a shopping cart icon), '4 Total Orders' (with a clipboard icon), and '0% Admin Commission Rate' (with a gift icon). At the bottom of the page, there's a footer with a flower icon, the text '© 2023 Pickbazar. All right reserved RedQ.', and a version number '10.2.0'.

Introduction to Products management

Products is the life line of any e-commerce business. So, it's very important to manage & control products. In our application, we have create a set of products & related data group. Please check this screenshot.



PickBazar

✉ My Shops

PRODUCT MANAGEMENT

📦 Products >

📦 Inventory

📁 Categories

🏷 Tags

⭐ Attributes

👤 Manufacturers/Publications

👤 Authors

E-COMMERCE MANAGEMENT

% Taxes

🚚 Shippings

📄 Withdrawals

>Returns Refunds >

LAYOUT/PAGE CONTROL

🏡 Home pages / Types

Products

Super admin can check all the listed products in his site. Also he has ability to check all the low stock & out of stock products of all listed shop. So, he/she can contact with that shop owner about their products inventory.

ID	Product	Product Type	Shop	Price/Unit	Quantity	Status	Actions
#ID: 946	Blood Drinker Books	variable	Books Shop	\$180.00 - \$200.00	1000	publish	***
#ID: 945	Forest Killer Books	simple	Books Shop	\$280.00	500	publish	***
#ID: 944	Flesh Eater Books	simple	Books Shop	\$250.00	500	publish	***
#ID: 943	The Boneyard Man Books	variable	Books Shop	\$300.00 - \$300.00	1000	publish	***
#ID: 942	The Physico Killer First Part Books	simple	Books Shop	\$200.00	500	publish	***
#ID: 941	The Serial Killer Books	simple	Books Shop	\$200.00	500	publish	***
#ID: 940	Before The Rains Books	simple	Books Shop	\$200.00	500	publish	***
#ID: 939	Eight Perfect Murders Books	simple	Books Shop	\$200.00	500	publish	***
#ID: 938	The Hidden Planet Books	simple	Books Shop	\$300.00	500	publish	***
#ID: 937	Broken Stars Books	simple	Books Shop	\$350.00	500	publish	***
#ID: 936	Escape The Pod Books	variable	Books Shop	\$180.00 - \$200.00	1000	publish	***
#ID: 935	The Black Hole Books	simple	Books Shop	\$300.00	500	publish	***
#ID: 934	Space Wars Books	simple	Books Shop	\$280.00	500	publish	***
#ID: 933	Space Force Peace Books	variable	Books Shop	\$200.00 - \$220.00	1000	publish	***
#ID: 931	Space Force First Chapter Books	simple	Books Shop	\$200.00	500	publish	***
#ID: 932	Space Force Second Chapter Books	variable	Books Shop	\$150.00 - \$152.00	1000	publish	***
#ID: 930	Milan The Story of Love Books	simple	Books Shop	\$160.00	500	publish	***
#ID: 929	All About Love Books	simple	Books Shop	\$200.00	500	publish	***
#ID: 928	The Hero Alain Story Books	variable	Books Shop	\$300.00 - \$350.00	1000	publish	***
#ID: 927	The Dutch Princess Books	variable	Books Shop	\$100.00 - \$120.00	1000	publish	***

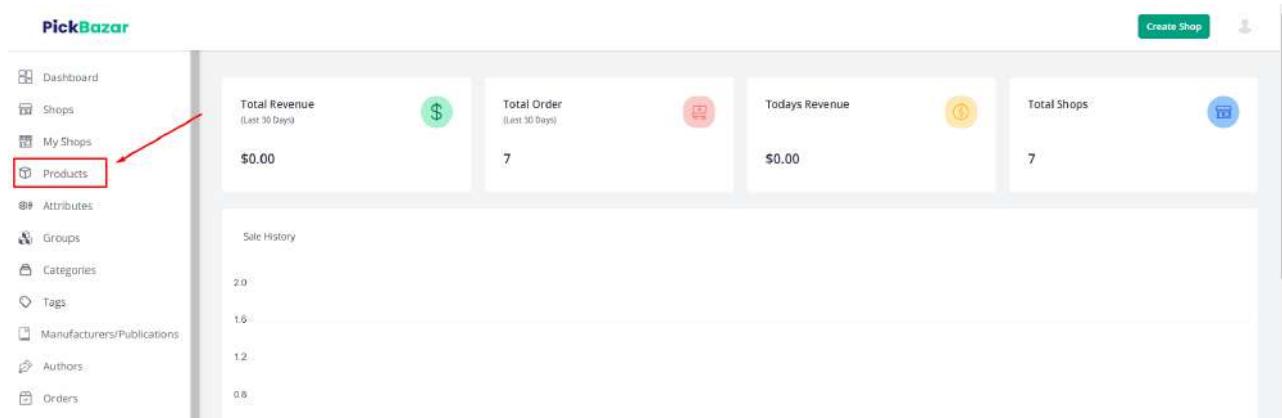
Product form

Products are the core components of the site, representing items or services available for sale. Admins add products to Pickbazar to populate the ecommerce platform with offerings that customers can purchase.

By adding products, admins provide a diverse range of options to customers, attracting and engaging a wider audience. Each product contains essential information like price, description, and images, enabling users to make informed buying decisions. Admins add products to create a comprehensive and competitive online marketplace that caters to various consumer needs and preferences.

Super Admin (Product form)

- The **Products** option is accessible through the admin dashboard.



The screenshot shows the PickBazar admin dashboard. On the left, there's a sidebar with various menu items: Dashboard, Shops, My Shops, **Products** (which has a red arrow pointing to it), Attributes, Groups, Categories, Tags, Manufacturers/Publications, Authors, and Orders. The main area displays four cards: Total Revenue (Last 30 Days) with \$0.00, Total Order (Last 30 Days) with 7, Today's Revenue with \$0.00, and Total Shops with 7. Below these cards is a section titled 'Sale History' with some numerical values.

- Once you enter the **Products**, you can sort products by name, price/unit, quantity and status. You can also search, edit, remove and filter products.

Here are the descriptions of types of sorting:



The screenshot shows a list of products in the admin dashboard. The columns include Image, Name, Group, Shop, Price/Unit, Quantity, Status, Actions, and a Publish button. Red arrows point to the 'Name +' column header, the 'Price/Unit +' column header, the 'Status +' column header, and the 'Actions' column header. Each row contains a product image, its name ('Blood Drinker' and 'Forest Killer'), its group ('Books'), shop ('Books Shop'), price range ('\$180.00 - \$200.00' and '\$260.00'), quantity (1000 and 500), a 'PUBLISH' button, and an 'Actions' column with icons for Remove and Edit.

Name : Sorting by product name arranges items in alphabetical or lexicographic order based on their names. Products will be listed in ascending (A to Z) or descending (Z to A) order, making it easier for users to find specific items or navigate through the catalog alphabetically.

Price/Unit : Sorting by product price per unit orders items based on their price. Products will be arranged from the lowest to the highest price (ascending) or vice versa (descending). This sorting option helps customers compare prices and find products that fit their budget.

Quantity : Sorting by quantity organizes products based on the available stock or inventory levels. Products can be sorted from those with the highest quantity in stock to the lowest or the other way around. This is useful for customers looking for products with ample availability.

Status : Sorting by status categorizes products based on their current publication status. Published products are those actively displayed on the ecommerce site and available for purchase, while draft products are in the editing or preparation phase and not yet visible to customers. Sorting by status helps admins manage and review the product catalog efficiently.

The screenshot shows a product management interface with the following elements:

- Search Bar:** A top navigation bar with a "Search Here" field and a "Search by Name" input field.
- Filter Buttons:** "Filter By Group" and "Filter By Category".
- Product List:** A table showing four products:

Image	Name	Group	Shop	Price/Unit
	Nair Hair Removal Cream	Grocery	Grocery Shop	\$8
	Espa Men Dual Action	Grocery	Grocery Shop	\$6
	Human Nature Shaving Cream	Grocery	Grocery Shop	\$6
	Longmarket Barber Aftershave...	Grocery	Grocery Shop	\$6.00
- Category Filter:** A dropdown menu titled "Fruits & Vegetables" listing categories like Fruits, Vegetables, Meat & Fish, Fresh Fish, and Meat.
- Buttons:** "PUBLISH", a trash icon, and a edit icon.
- Bottom Filters:** "Filter By Group" (Bakery) and "Filter By Category" (Cookies).

Search : The ability to find specific products quickly by entering keywords or phrases, helping users navigate the catalog efficiently.

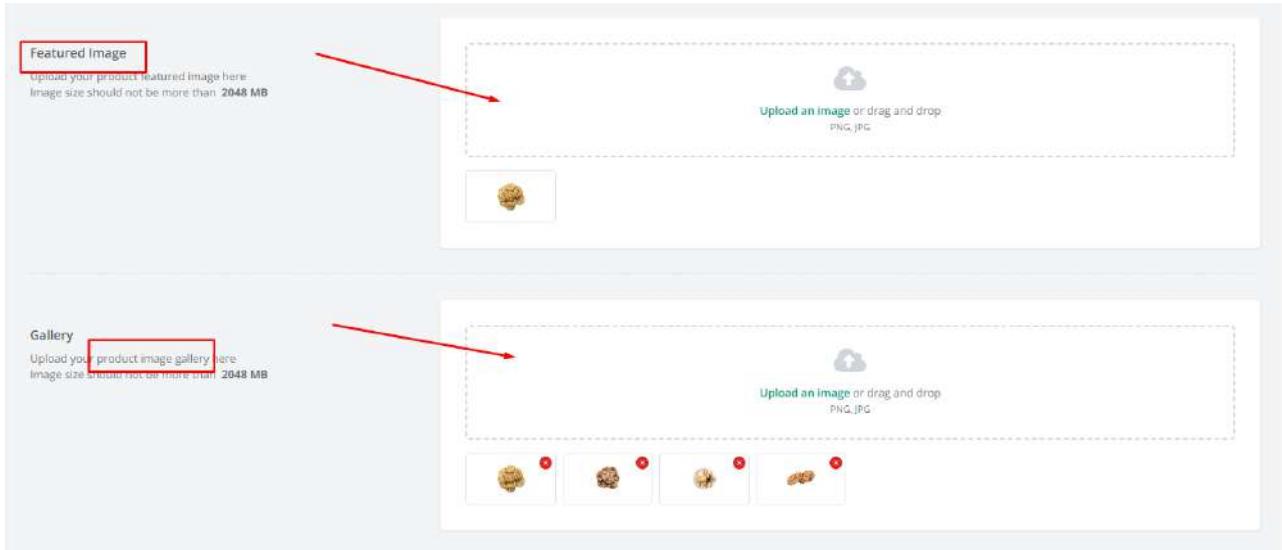
Filter by Group : A feature allowing users to refine product results by selecting a specific group, narrowing down choices to relevant categories based on the chosen group. Clicking on the cross icon clears all selected filters and restores the product listing to its default or unfiltered state. That allows users to reset or remove any applied filters

Filter by Category : A filtering option that displays relevant categories based on the chosen group, allowing users to further refine their search within the selected product group. This dynamic filtering optimizes the user experience by displaying only pertinent categories for the chosen group, enhancing navigation and selection accuracy. Clicking on the cross icon clears all selected filters and restores the product listing to its default or unfiltered state. That allows users to reset or remove any applied filters

- The administrator will click **Edit** icon. A form will appear with previous data in the input fields. Now user can modify an existing Products.

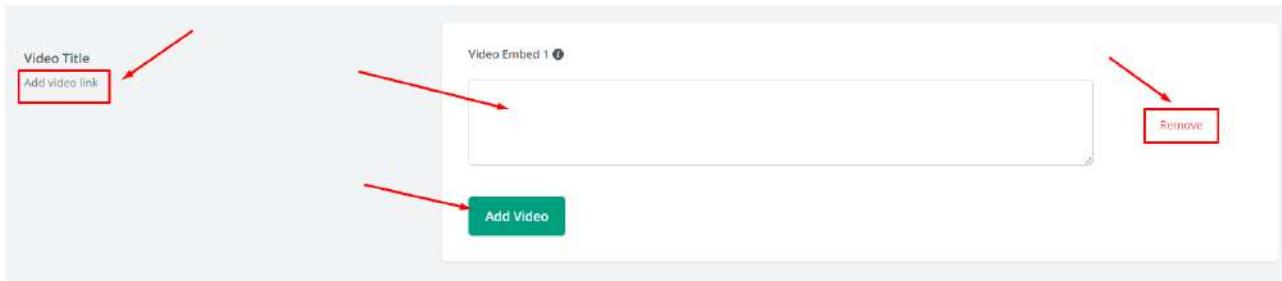
Which modification will appear in the Products list.

Here are descriptions of each field:



Product Featured Image : The primary visual representation of a product, prominently displayed on the product detail page, offering customers an initial view of the item.

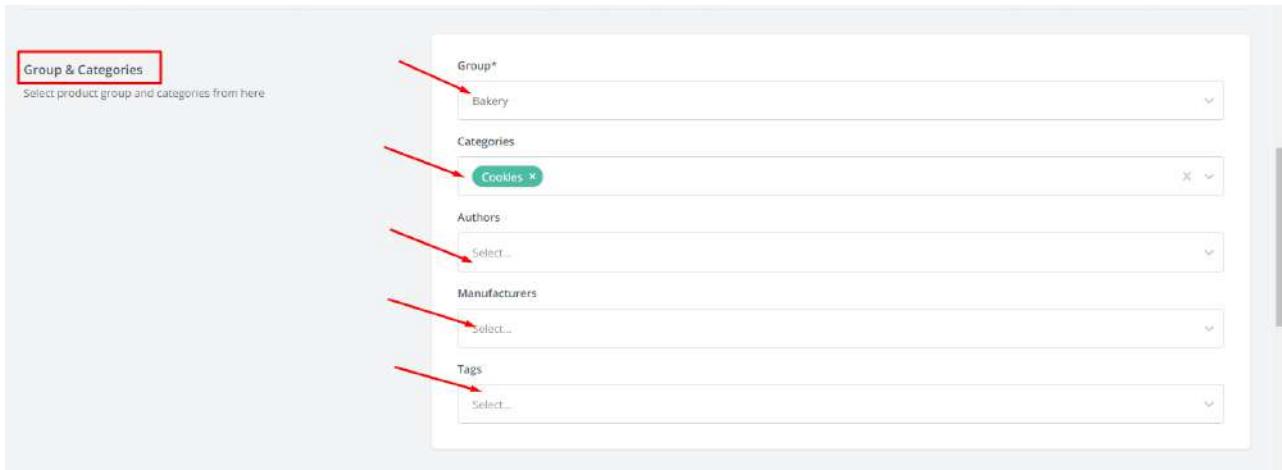
Product Image Gallery : A collection of additional images showcasing the product from various angles or depicting different variations, allowing customers to explore the product in more detail before making a purchase.



Video Link Input Field : A space for users to enter video links that provide multimedia content or demonstrations related to the product, enabling a richer product experience.

Remove : A function that allows users to individually delete video links they've added, providing control over the displayed content.

Add video : A button that permits users to include multiple video links by clicking, enhancing the product presentation with various multimedia resources.



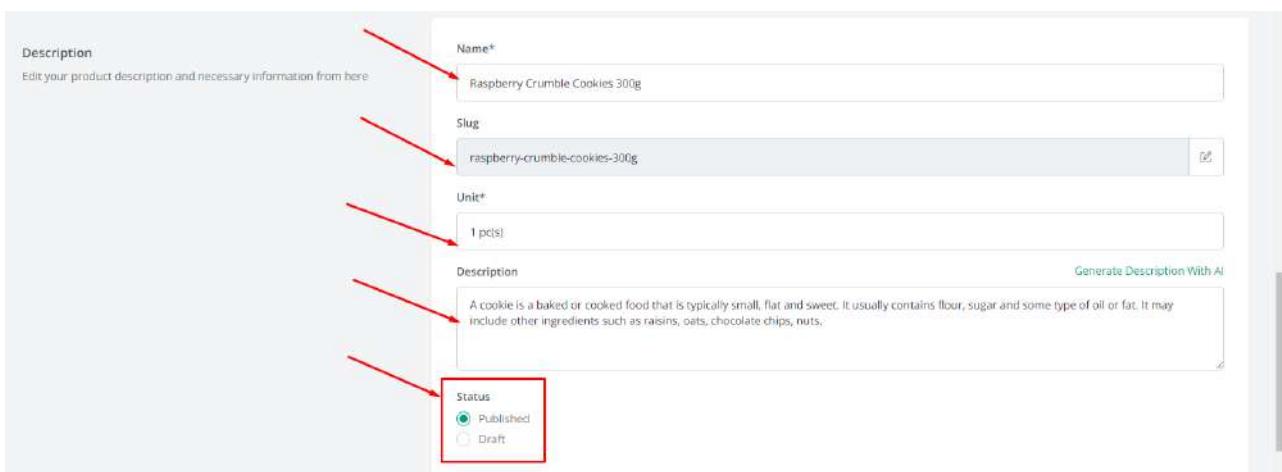
Groups : High-level categories that classify products into distinct themes or collections, such as grocery, bakery, clothing, bags, makeup, and daily needs.

Categories : Subcategories displayed based on the selected `group, providing a more specific classification for products; for example, in the "grocery" group, categories might include fruit & vegetables, meat, fish, and dairy.

Authors : The creators or writers of books or literary works associated with a product, helping customers identify and explore items by their authors.

Manufacturer : The entity or company responsible for producing or manufacturing a product, offering insights into its origin and quality.

Tags : Descriptive keywords or labels assigned to products, aiding in search and categorization, and providing additional context or features related to the product.



Name : The unique title or identifier for a product, enabling easy recognition and search by customers.

slug : A URL-friendly version of the product name used in website links, enhancing search engine optimization and accessibility.

Unit : The measurement or quantity associated with a product, specifying how it's sold or packaged, such as per kilogram, piece, or liter.

Description : A detailed text that provides information about the product's features, benefits, and specifications, aiding customers in making informed purchasing decisions.

Status : Indicates whether a product is "published" and available for customers or in "draft" mode, undergoing editing or preparation before being made visible on the website.



Simple : A product type where each item is identical and doesn't have multiple variations. Customers select the product as is without choosing different options like size or color.

Variable : A product type offering various options or variations for customers to choose from, such as different sizes, colors, or styles. Each variation may have its own price or characteristics, allowing for customization within the same product listing.

- If product type is **simple** following input fields need to be filled:

A screenshot of a 'Simple Product Information' form. At the top, it says 'Simple Product Information' and 'Edit your simple product description and necessary information from here...'. There are several input fields with red arrows pointing to them:

- 'Price*' field containing '4'.
- 'Sale Price' field containing '3.5'.
- 'Quantity*' field containing '50'.
- 'SKU*' field containing '3050'.
- 'Width' field.
- 'Height' field.
- 'Length' field.
- A group of checkboxes at the bottom:
 - Is Digital
 - Is External

The 'Update Product' button is highlighted with a red box and a red arrow pointing to it.

Price : The regular or original cost of a product before any discounts or promotions, representing its base value.

Sale Price : The temporarily reduced price at which a product is offered, typically during a promotional period or sale event, providing customers with a discounted rate.

Quantity : The number of available units of a product in stock, indicating its current availability for purchase.

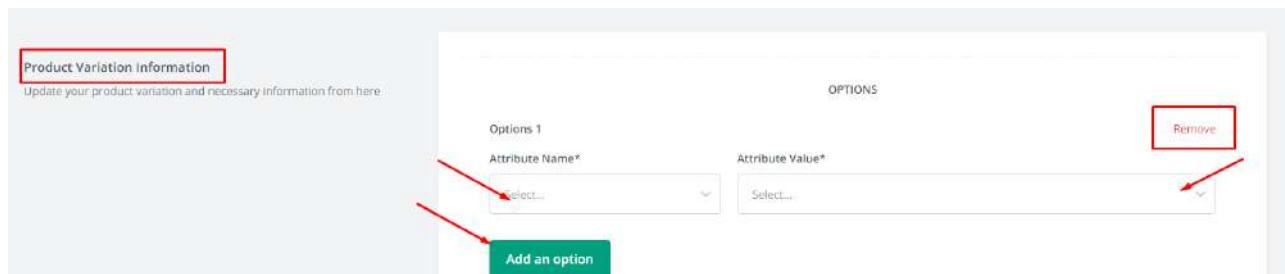
SKU : A unique identifier or code assigned to a product, simplifying inventory management and tracking.

Width, Height, Length : Dimensions that describe the physical size of a product, helpful for shipping calculations and product description.

Is Digital : A checkbox indicating whether the product is digital in nature, such as a downloadable file or digital service.

Is External : A checkbox signifying whether the product is external or linked to an external source, often used for affiliate or referral products.

- If product type is **variable** following input fields need to be filled:

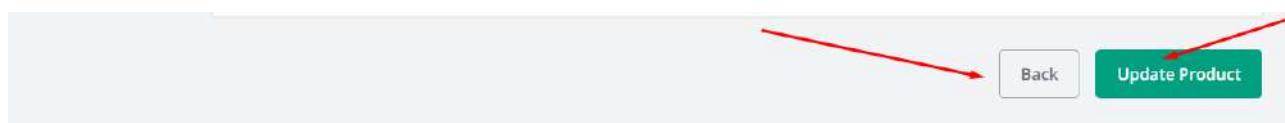


Add Option : A button that allows users to include additional variations for a variable product, revealing two input fields for attribute name and attribute value with each click.

Attribute Name : A descriptive label or characteristic associated with a product, such as size or color, used to define product variations.

Attribute value : The specific option or choice corresponding to an attribute, defining the unique variation of a product, like "Small" for size or "Red" for color.

Remove : A feature that permits users to individually eliminate specific attribute options they've added, providing flexibility in configuring product variations.



Update Product : Click to save and apply the changes made to the product.

Back Button : Return to the previous page or exit the product modification form without saving changes, ensuring data is not altered unintentionally.

Vendor (Product form)

Administrators keep control of the platform while enabling merchants to their product listings as needed. Admin can edit and delete product, but he is not permitted to add any product.

- To add product, vendors must log in to their own dashboards. Visit his particular store to add product.

The screenshot shows the PickBazar vendor dashboard. On the left, there's a sidebar with a 'Store Owner' profile, a contact number (+12365141641631), and a 'Logged in as Vendor' message. Below this are links for 'Dashboard', 'My Shops' (which is highlighted in green), 'Message', and 'Store Notice'. The main area displays a grid of nine shop cards. Each card includes a circular logo, the shop name, address, phone number, and four numerical metrics: Commission, Sale, Balance, and Withdraw. A red arrow points from the 'My Shops' link in the sidebar to the 'Furniture Shop' card.

- The **Products** option is accessible through the vendor dashboard.

The screenshot shows the PickBazar vendor dashboard for the 'Furniture Shop'. The sidebar on the left has a 'Products' link highlighted with a red box. The main content area shows the 'Furniture Shop' card with its logo, name, address, and a 'Visit Shop' button. To the right is a large image of a modern living room interior. At the top right of the main area is a 'Create Shop' button, and below it is an 'Edit Shop' button.

- Once you enter the **Products**, you can add, sort products by name, price/unit, quantity and status. You can also search, edit, remove and filter products.

You can add new product by clicking on **Add Product** Button. An empty form will appear with some input field and an **Add product** button to save.

The screenshot shows a product management interface. At the top, there is a header with a 'Create Shop' button and a user profile icon. Below the header, a search bar contains the placeholder 'Type your query and press enter'. To the right of the search bar is a green button labeled '+ Add Product' with a red box drawn around it. A tooltip above the button says 'Click here and get an empty form to add product'. Further to the right are 'Filter' and other settings icons. The main area is titled 'Products' and contains a table with two rows of data. The columns are: Image, Name, Group, Shop, Price/Unit, Quantity, Status, and Actions. Red arrows point from the column headers to their respective labels: 'Name +' points to 'Name', 'Price/Unit +' points to 'Price/Unit', 'Quantity +' points to 'Quantity', and 'Status +' points to 'Status'. In the Actions column, two rows show 'Blood Drinker' and 'Forest Killer'. Each row has a 'PUBLISH' button, a trash icon, and an edit icon. A tooltip for the trash icon says 'Click the icon to Remove' and a tooltip for the edit icon says 'Click the icon to Edit'.

Image	Name	Group	Shop	Price/Unit	Quantity	Status	Actions
	Blood Drinker	Books	Books Shop	\$180.00 - \$200.00	1000	PUBLISH	
	Forest Killer	Books	Books Shop	\$260.00	500	PUBLISH	

Inventory

This is an important route to keep the product supply flow on a good pace. By this route super admin can easily checkout the product inventory status.

Screenshot of the PickBazar Admin Dashboard showing the Inventory section.

Inventory

ID	Name	SKU	Quantity	Sold quantity	Actions
#ID: 946	Blood Drinker Books	hijklijklijklijklijklijkl	1000	0	
#ID: 945	Forest Killer Books	hijklijklijklijklijklijkl	500	0	
#ID: 944	Fleash Eater Books	ijklijklijklijklijklijkl	500	0	
#ID: 943	The Boneyard Man Books	ijklijklijklijklijklijkl	1000	0	
#ID: 942	The Psycho Killer First Part Books	5d4f95645464jk584j514jk54jk	500	0	
#ID: 941	The Serial Killer Books	654561654+58949846luyghfodf	500	0	
#ID: 940	Before The Ruins Books	ijklijklijklijklijklijkl	500	0	
#ID: 939	Eight Perfect Murders Books	kyukyhgj+ghj5+hsdg	500	0	
#ID: 938	The Hidden Planet Books	mthj.kjp/g/g+jk5jk+5gjk	500	0	
#ID: 937	Broken Stars Books	ghijklmnijklmnijklmn	500	0	
#ID: 936	Escape The Pod Books	ijklijklijklijklijklijkl	1000	0	
#ID: 935	The Black Hole Books	jkhijklhjmg-hmjnjmhjlmhjlmhj	500	0	
#ID: 934	Space Wars Books	hejuktlukenho+thg+nSihghedvn	500	0	
#ID: 933	Space Force Peace Books	df5as+15ada+15aa+15ada+5fb+sa5sf+	1000	0	
#ID: 931	Space Force First Chapter Books	df5as+15ada+15aa+15ada+5fb+sa5sf+	500	0	
#ID: 932	Space Force Second Chapter Books	df5as+15ada+15aa+15ada+5fb+sa5sf+	1000	0	
#ID: 930	Milan The Story of Love Books	+g+df5ga56l14ds5456sdfl	500	0	
#ID: 929	All About Love Books	56d165d4ga5f4ds56f4ds56f4sd+f+fsa...	500	0	
#ID: 928	The Hero Alam Story Books	ijklijklijklijklijklijkl	1000	0	
#ID: 927	The Dutch Princess Books	ijklijklijklijklijklijkl	1000	0	

Navigation: < 1 2 3 ... 47 >

Footer: ©2023 Pickbazar. All right reserved RedQ. Version 10.2.0

Product group

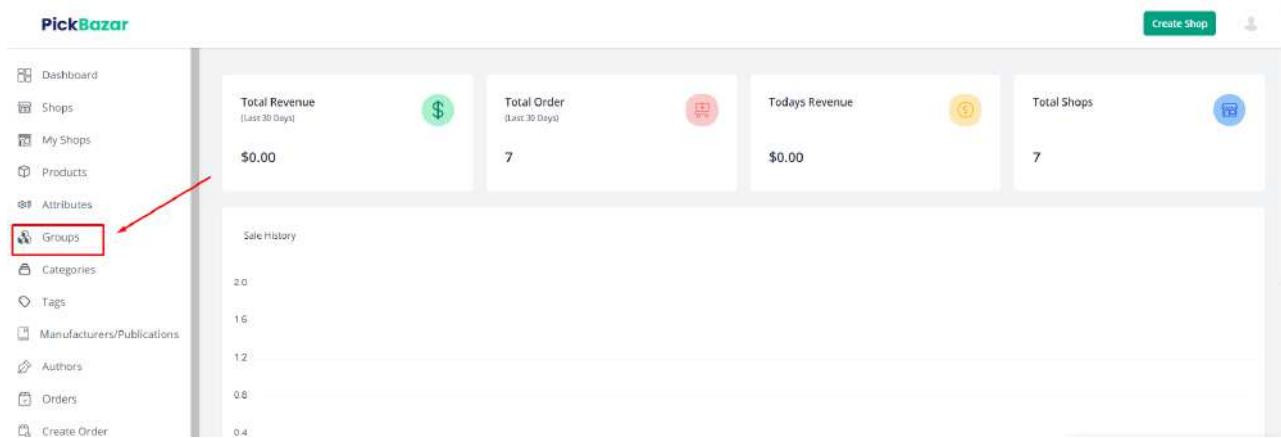
"Groups" refer to overarching thematic collections that encompass a range of related categories and products. Groups serve as high-level categories that help customers quickly identify and access products that align with their interests or needs. They simplify navigation by grouping related categories together, making it easier for users to explore a wide range of offerings. Groups also facilitate effective merchandising and marketing strategies, allowing admins to highlight specific product collections or themes.

By adding groups, admins enhance the overall organization of the site, improve customer engagement, and contribute to a more efficient and enjoyable shopping environment. For instance, the "grocery" group might include categories like "canned goods," "spices," and "beverages," making it simpler for users to find essential food items.

Similarly, the "bakery" group could house categories such as "bread," "pastries," and "cakes," creating a dedicated space for baked goods enthusiasts. "Daily needs" might include categories like "cleaning supplies," "personal care," and "health products," catering to everyday essentials. "Furniture" could encompass categories like "living room," "bedroom," and "office furniture," making it convenient for users shopping for home decor.

Groups in Super Admin perspective

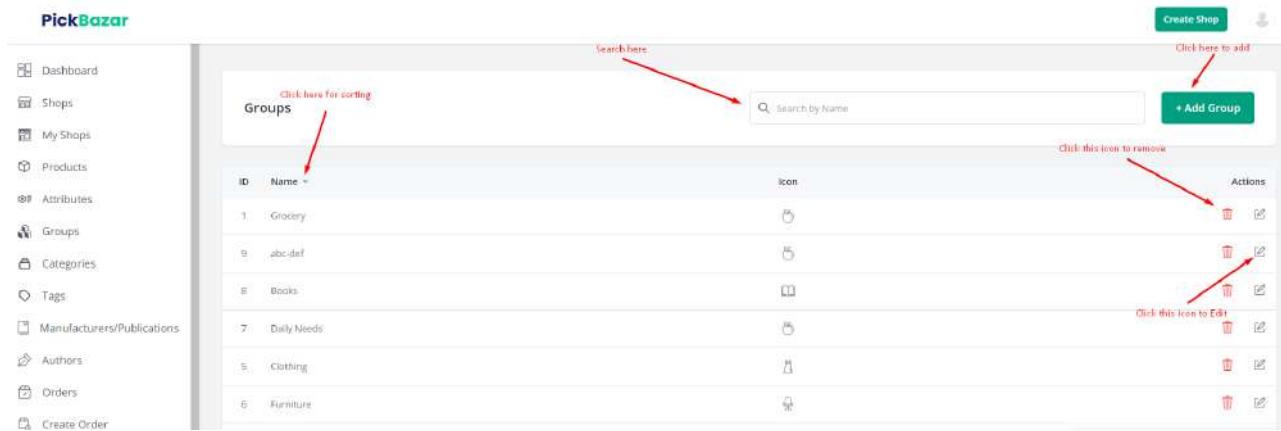
- The **Groups** option is accessible through the admin dashboard.



This screenshot shows the PickBazar admin dashboard. On the left, there is a sidebar with various menu items: Dashboard, Shops, My Shops, Products, Attributes, Groups (which is highlighted with a red box), Categories, Tags, Manufacturers/Publications, Authors, Orders, and Create Order. The main area displays four key metrics: Total Revenue (Last 30 Days) at \$0.00, Total Order (Last 30 Days) at 7, Today's Revenue at \$0.00, and Total Shops at 7. Below these metrics is a Sale History chart showing values of 2.0, 1.6, 1.2, 0.8, and 0.4.

- Once you enter the **Groups**, the **Add Groups** button will be available for adding additional Groups.

In **Groups** menu you can search, sort by name, add, remove or modify Groups from there.



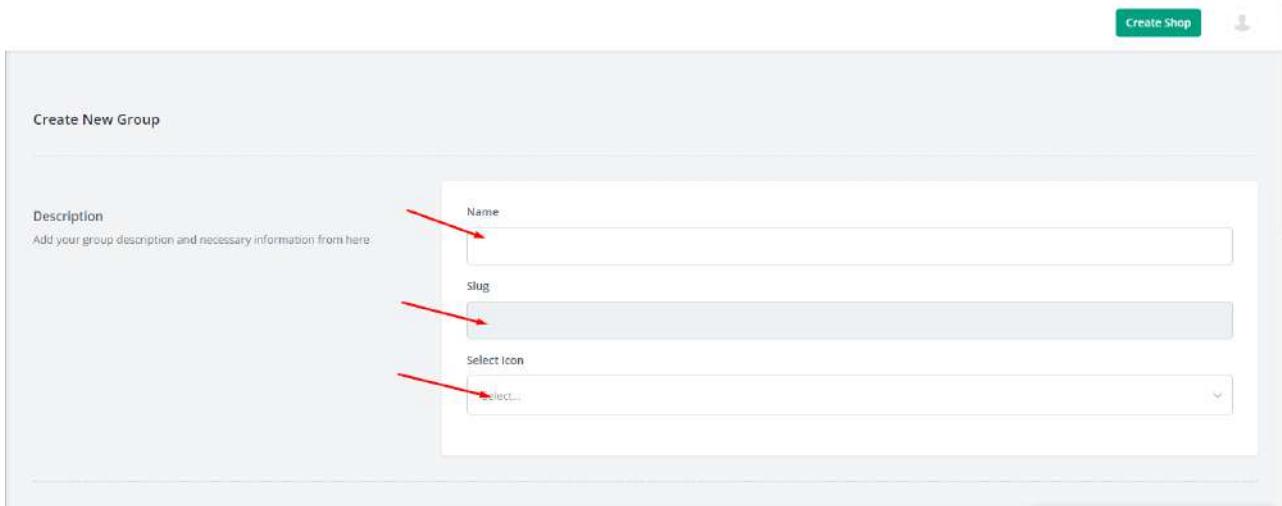
This screenshot shows the Groups management page within the PickBazar admin dashboard. The sidebar remains the same as the previous screenshot. The main area lists groups with columns for ID, Name, Icon, and Actions. A red arrow points to the "Name" column header with the text "Click here for sorting". Another red arrow points to a search bar with the placeholder "Search by Name". A third red arrow points to a green button labeled "+ Add Group". A fourth red arrow points to the "Actions" column with the text "Click this icon to remove". A fifth red arrow points to the "Actions" column with the text "Click this icon to Edit". The table data is as follows:

ID	Name	Icon	Actions
1	Grocery	Grocery icon	Remove Edit
2	abc-def	Books icon	Remove Edit
3	Books	Books icon	Remove Edit
4	Daily Needs	Books icon	Remove Edit
5	Clothing	Clothing icon	Remove Edit
6	Furniture	Furniture icon	Remove Edit

- The administrator will click **Add Groups** button after filling out the form to create a new Groups.

Which will appear in the Groups list.

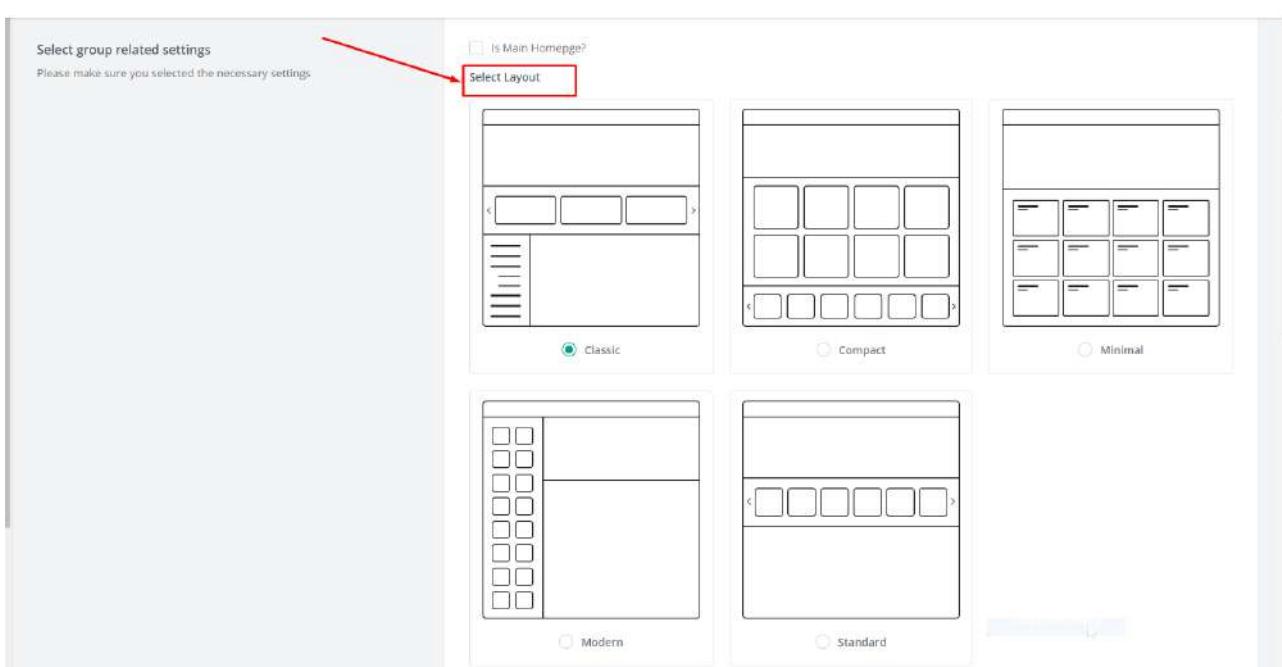
Here are descriptions of each field:



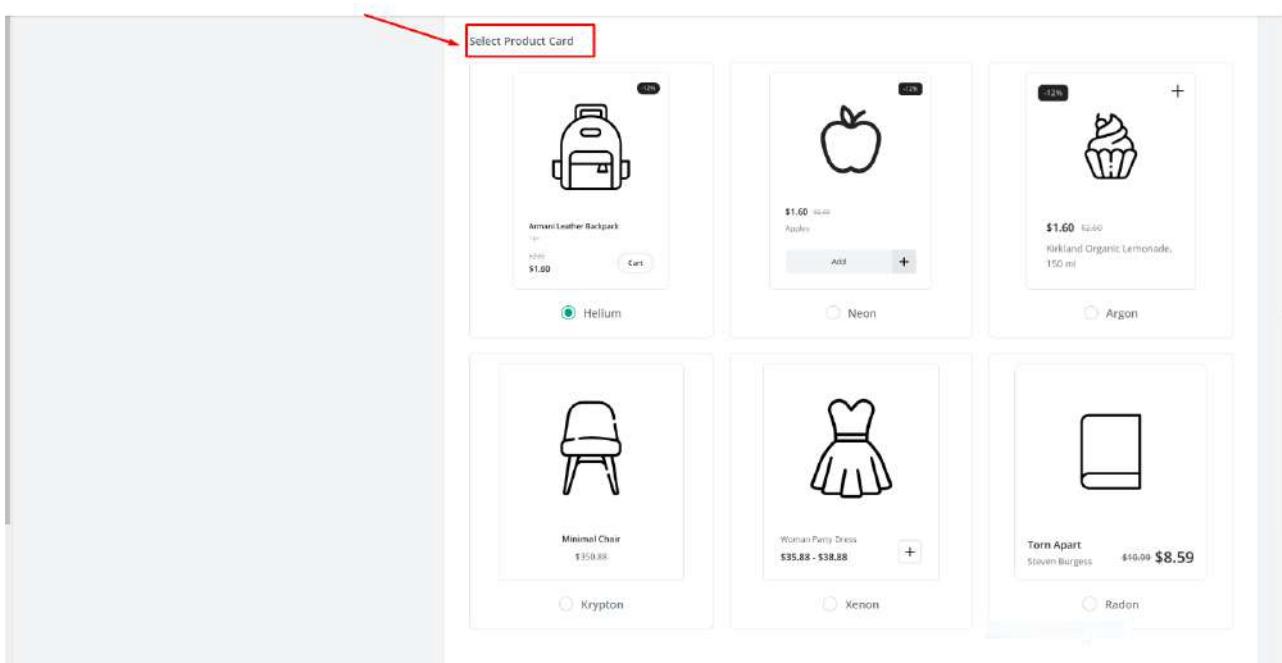
Name : Enter the name of the group, which represents a thematic collection of related categories or products.

slug : Provide a URL-friendly version of the group name used in the website's URL. It should be unique and composed of lowercase letters, hyphens, or numbers.

Select Icon : Choose an icon that visually represents the group, aiding users in quickly identifying and associating it with specific product themes.



Select Layout : Specify the layout or display style for the group, such as grid or list view, affecting how products within the group are presented.

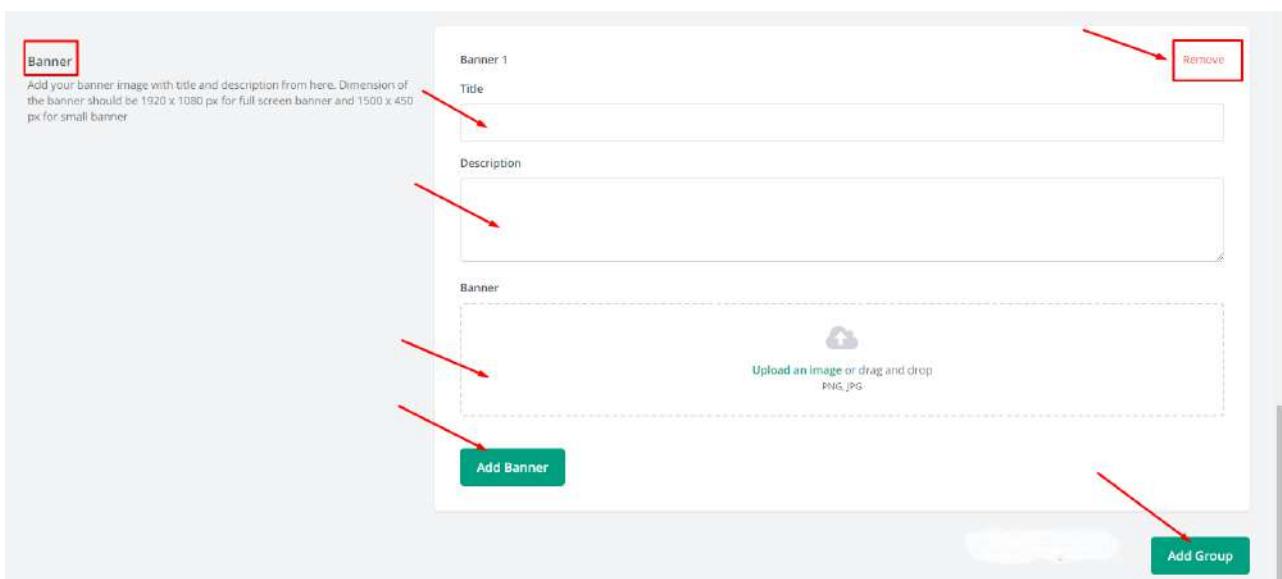


Select Product Card : Choose a template or style for displaying product information within the group, enhancing the visual presentation of items.



Upload Promotional Sliders : Add promotional banners or sliders to highlight featured products or offers within the group.

Inside **Add Banner** Button Fields:



Title : Provide a title for the promotional banner, describing the featured content or offer.

Description : Include a brief description or details about the promotional banner's content or purpose.

Banner Image : Upload an image that visually represents the banner content and entices users to explore further.

remove : Added banner can be removed by clicking on **remove**.

Add Group : Click to finalize and add the group to the ecommerce site. Ensure all necessary information, including promotional banners, is filled out accurately before adding the group.

Product Category

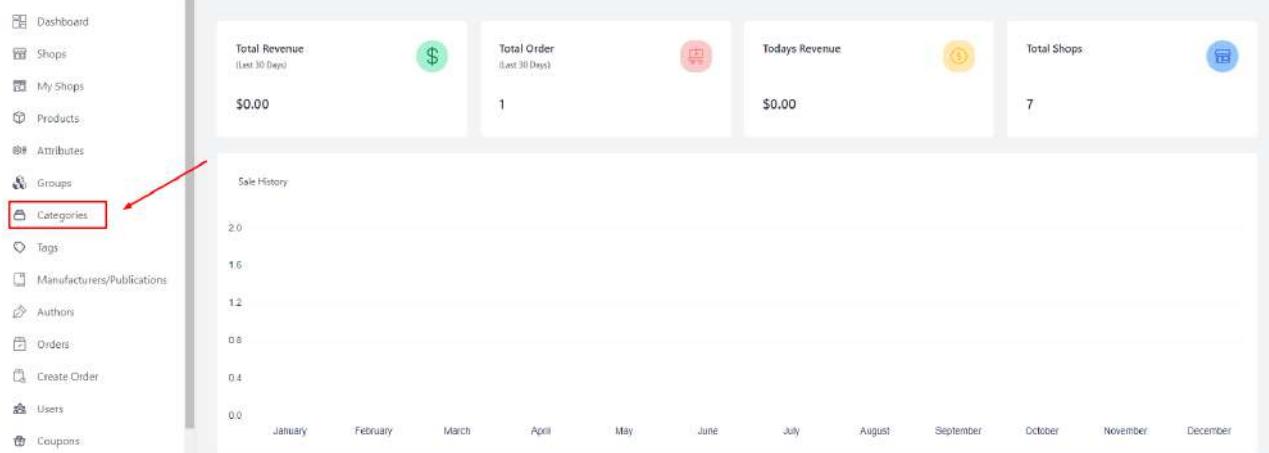
Category refers to a distinct and organized grouping of related products.

Categories are used to classify and group similar items together, making it easier for customers to browse and find products based on their interests or needs. Each category represents a specific type of merchandise or a thematic collection, such as **Thriller**, **Fiction**, **Horror**, **Children's Literature**, **Literature**, **Novels**, **Romantic Poetry**, **Comic books** are some categories of **Book** group.

By structuring products into categories, Pickbazar enhances the user experience, enabling efficient navigation and discovery of products within well-defined sections of the site. This classification also contributes to effective search engine optimization and helps users quickly identify the areas of the site that align with their preferences.

Category in Super Admin perspective

- The **Categories** option is accessible through the admin dashboard.



- Once you enter the **Categories**, the **Add Categories** button will be available for adding additional Categories.

In **Categories** menu you can search, filter by group, add, remove or modify Categories from there.

The categories management interface allows administrators to view, search, and manage category lists. It features a search bar, a dropdown filter for groups, and buttons for adding, removing, and editing categories.

ID	Name	Details	Image	Icon	Actions
179	Thriller	Cerebral suspenser is a kidney...			
178	Romantic Poetry	The runes of Lyrical Ditties de...			
177	Novels	A novel is a fairly long work of ...			
176	Horror Fiction	Horror is a kidney of academi...			
175	Comic books	A ridiculous book, also called r...			

- The administrator will click **Add Categories** button after filling out the form to create a new Categories. Which will appear in the Categories list.

Here are descriptions of each field:

The 'Create New Category' form includes fields for the category name, an image upload, and a detailed description.

Image	Upload your category image here	Upload an Image or drag and drop PNG, JPG
<p>Category Name: Thriller</p> <p>Description: Cerebral suspenser is a kidney...</p>		

Image : Upload an image that visually represents the category. This image provides a visual cue for the category and enhances the browsing experience.

The screenshot shows a user interface for adding a new category. At the top left is a section labeled "Description" with the placeholder "Add your category details and necessary information from here". To its right is a "Name" input field. A red arrow points from the "Name" label to the input field. Below it is a "Details" input area, also with a red arrow pointing to it. On the far right of the "Details" area is a "Generate Description With AI" button. Next is a "Select Icon" dropdown menu with a red arrow pointing to the "Select..." button. Following is a "Types" dropdown menu with a red arrow pointing to the "Select..." button. Then comes a "Parent Category" dropdown menu with a red arrow pointing to the "Select..." button. In the bottom right corner of the form area is a large green rectangular button with the text "Add Category" in white.

Name : Enter the name of the category. This is the primary identifier for the category and will be displayed throughout the site.

slug : The URL-friendly version of the category name used in the website's URL. It should be unique and contain lowercase letters, hyphens, or numbers.

Details : Provide additional information about the category, including its characteristics, products it encompasses, or any relevant details.

Select Icon : Choose an icon that symbolizes the category. Icons aid in quick recognition, making categories more intuitive for users to identify.

Types : Specify the type of category, such as "Grocery," "Clothing," etc. This classification helps manage different types of categories effectively.

Parent Category : If applicable, select a parent category under which this category falls. Useful for organizing hierarchical relationships between categories.

Add Category : Click to finalize and add the category to the ecommerce site's category structure. Ensure all essential information is provided accurately before adding.

Product Tags

Tags refer to descriptive keywords or labels that are assigned to products or content. These tags serve as identifiers that categorize and group items based on specific attributes, characteristics, or themes.

Admins add tags to products on an ecommerce site to enhance searchability and improve user navigation. Tags are descriptive keywords or labels that categorize products based on attributes like size, color, type, and more.

By using tags, customers can easily find relevant products, leading to a smoother shopping experience and increased customer satisfaction. Tags also aid in organizing the site's inventory, making it easier for admins to manage and update product listings.

Tags in Super Admin perspective

- The **Tags** option is accessible through the admin dashboard.

The screenshot shows the PickBazar admin dashboard. On the left, there is a sidebar with various menu items: Dashboard, Shops, My Shops, Products, Attributes, Groups, Categories, Tags (which is highlighted with a red box and has a red arrow pointing to it), Manufacturers/Publications, Authors, Orders, Create Order, Users, and Coupons. The main area displays real-time statistics: Total Revenue (\$0.00), Total Order (1), Today's Revenue (\$0.00), and Total Shops (7). Below these stats is a "Sale History" chart showing monthly sales from January to December. The chart indicates a peak in sales during the first quarter (January, February, March) and a decline towards the end of the year.

- Once you enter the **Tags**, the **Add Tags** button will be available for adding additional Tags.

In **Tags** menu you can search, add, remove or modify Tags from there.

The screenshot shows the Tags management page. At the top, there is a search bar with the placeholder "Search here" and a button labeled "+ Add Tag". Below the search bar is a table listing existing tags. The table columns are ID, Name, Slug, Group, and Actions. Each row contains a set of icons for modification and removal. The table lists the following tags:

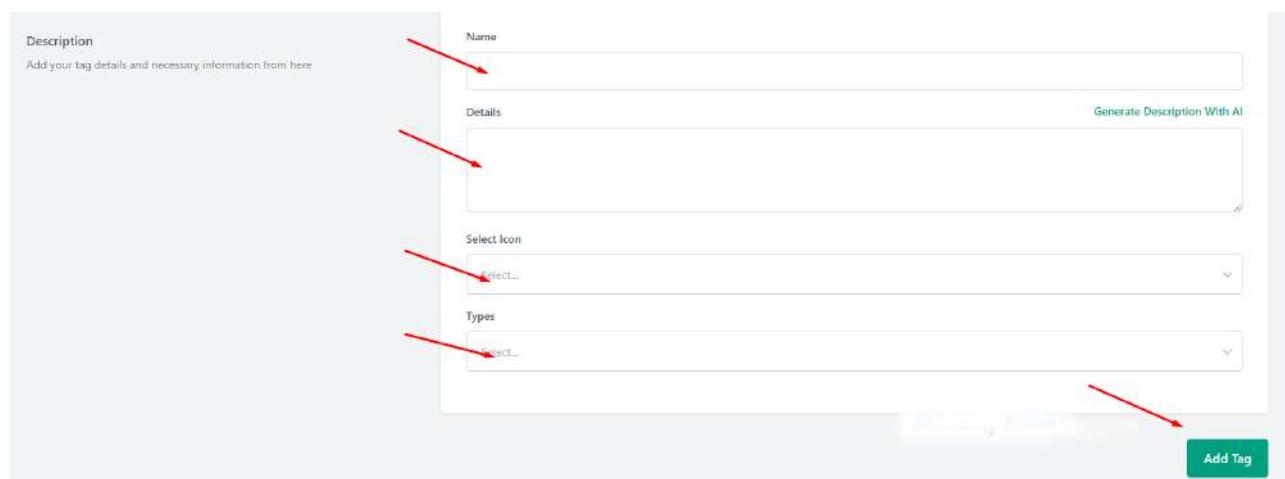
ID	Name	Slug	Group	Actions
26	Combo	combo	Books	
25	Touching	touching	Books	
24	Fantasy	fantasy	Books	
23	Space	space	Books	
22	Tear	tear	Books	
21	Sad	sad	Books	
20	Kids Story	kids-story	Books	

- The administrator will click **Add Tags** button after filling out the form to create a new Tags. Which will appear in the Tags list.

Here are descriptions of each field:



Image : Upload an image that visually represents the tag. This image provides a quick visual reference for the tag, helping users identify it easily.



Name : Enter the name of the tag. This is the main identifier for the tag and will be displayed alongside products associated with it.

slug : A slug is a URL-friendly version of the tag name, often used in the website's URL. It should be unique and composed of lowercase letters, hyphens, or numbers.

Details : Provide additional information about the tag. Describe its purpose, characteristics, or any relevant details that can help users understand its context.

Select Icon : Choose an icon that visually represents the tag. Icons can convey meaning at a glance, making tags more recognizable and enhancing the user experience.

Type : Specify the type of the tag, such as "category," "color," "material," etc. This helps categorize tags and provides context for their use.

Add Tag : Click this button to finalize and add the tag to the ecommerce site's tag collection. Ensure that all necessary information is filled out accurately before adding the tag.

Product Attributes

An **Attribute** refers to a characteristic or feature associated with a product that can vary or be customized by the customer. Attributes help describe and differentiate products within the site, such as size, color, material, or any other specification that can affect a customer's choice. For example, in the context of clothing, attributes might include size options like small, medium, and large, as well as color choices such as red, blue, and black. By using attributes, Pickbazar allows customers to easily select and customize products according to their preferences, providing a more personalized shopping experience.

- Super admin has no access to add any attributes.
- Only vendor can add attributes to the system.

The screenshot shows the PickBazar Admin Dashboard. On the left, there is a sidebar with various menu items: Dashboard, Shops, My Shops, Products, Inventory, Categories, Tags, **Attributes** (which is highlighted in green), Manufacturers/Publications, Authors, and Withdrawals. The main content area is titled "Attributes". It displays a table with five rows of data:

ID	Name	Values	Slug	Actions
#ID: 7	Type	one two	type	[Edit] [Delete] [More]
#ID: 6	Book Type	Hardcover Book Picture Book Paperback Book PDF Book	book-type	[Edit] [Delete] [More]
#ID: 5	Language	English Arabic Spanish French Hindi Hebrew	language	[Edit] [Delete] [More]
#ID: 4	Size	S M L XL	size	[Edit] [Delete] [More]
#ID: 3	Color	Red Blue White	color	[Edit] [Delete] [More]

Attributes in Super Admin perspective

- The **Attributes** option is accessible through the admin dashboard.

The screenshot shows the PickBazar Admin Dashboard. On the left, there is a sidebar with various menu items: Dashboard, Shops, My Shops, Products, **Attributes** (which is highlighted with a red box and has a red arrow pointing to it from the bottom-left), Groups, Categories, Tags, Manufacturers/Publications, Authors, Orders, and a few other items that are partially cut off. The main content area displays some statistics: Total Revenue (\$Last 10 Days) is \$0.00, Total Order (\$Last 30 Days) is 7, Todays Revenue is \$0.00, and Total Shops is 7. Below these stats, there is a "Sale History" table with three rows:

2.0	
1.6	
1.2	
0.8	

- Once you enter the **Attributes**, you can sort by name, remove or modify Attributes from there.

The screenshot shows a table titled "Attributes" with columns: ID, Name, Shop, Values, and Actions. Red arrows highlight specific features: one arrow points to the "Name" column header with the text "Click here to sort by Name"; another arrow points to the "Actions" column with the text "Click the icon to remove"; a third arrow points to the "Actions" column with the text "Click the icon to Edit". The data in the table includes:

ID	Name	Shop	Values	Actions
3	Color	Clothing Shop	Red, Blue, White	
4	Size	Clothing Shop	S, M, L, XL	
5	Language	Books Shop	English, Arabic, Spanish, French, Hindi, Hebrew	
6	Book Type	Books Shop	Hardcover Book, Picture Book, Paperback Book, PDF Book	

- The administrator will click **Edit** icon. A form will appear with previous data in the input fields. Now user can modify an existing Attributes.

Which modification will appear in the Attributes list.

Here are descriptions of each field:

The screenshot shows the "Attribute" editing form. It has two main sections: "Attribute" and "Attribute Values".

Attribute Section:

- Name:** A red box highlights the "Name" input field where "Color" is entered.
- Value:** A red box highlights the "Value" input field where "Red" is entered.
- Meta:** A red box highlights the "Meta" input field where "#cc1f6a" is entered.
- Remove:** A red box highlights the "Remove" button.

Attribute Values Section:

- Add Value:** A red box highlights the "Add Value" button.
- Back:** A red box highlights the "Back" button.
- Update Attribute:** A red box highlights the "Update Attribute" button.

Name : Enter or modify the name of the attribute, defining a characteristic or feature of the product. This name serves as the primary identifier for the attribute.

values : List the values associated with the attribute, representing different options or choices for that attribute. These values help customers select product variations.

Meta : Provide additional information or metadata about the attribute, such as measurement units or specific details that clarify its use.

Remove : Click to delete an existing attribute value or associated metadata. Use this option to refine or update the attribute's options.

Add Value : Click to add a new attribute value. After clicking, two input fields for "value" and "meta" will appear, allowing the addition of new options.

Update Attribute : Click to save and apply the changes made to the attribute, including modifications to attribute name, values, and metadata.

Back Button : Return to the previous page or exit the attribute modification form without saving changes, ensuring data is not altered unintentionally.

Attributes in Vendor perspective

Attribute on a vendor dashboard denotes additional details or qualities that suppliers might include with their goods or services. Vendors can create these characteristics on their own dashboards; administrators are not able to do so. Vendors can provide details about their items, such as size, color, or material, while adding attribute. These characteristics provide prospective purchasers thorough information.

Administrators keep control of the platform while enabling merchants to add attributes to their product listings as needed. Admin can edit and delete attribute, but he is not permitted to add any product attributes.

- To add attribute, vendors must log in to their own dashboards. Visit his particular store to add product attribute.

The screenshot shows the PickBazar vendor dashboard. On the left, there's a sidebar with the user's profile picture, name 'Store Owner', email 'store_owner@demo.com', and a message: 'This is the store owner and we have 6 shops under our banner. We are running all the shops to give our customers hassle-free service and quality products.' Below this is a 'Read more' link and a 'Contact' number '+12365141641631'. A red box highlights the 'Logged in as Vendor' status. The main area has a header with a search bar, 'Create Shop' button, and 'Visit Site' button. Below the header is a red button labeled 'Go to a partial Shop'. The main content is a grid of shop cards. Each card includes a shop icon, name, address, phone number, and four numerical stats: Commission, Sale, Balance, and Withdraw. A red arrow points from the 'Go to a partial Shop' button to the 'Furniture Shop' card. Other visible shop cards include Clothing Shop, Bags Shop, Makeup Shop, Bakery Shop, Grocery Shop, Books Shop, Gadget, and Medicine.

- The **Attributes** option is accessible through the vendor dashboard.



The screenshot shows the PickBazar dashboard. On the left, there's a sidebar with various menu items: Dashboard, Attributes (which is highlighted with a red box and has a red arrow pointing to it), Products, Authors, Manufacturers/Publications, Orders, Refunds, Staff, Withdrawals, Reviews, and Questions. The main area displays a shop named "Furniture Shop" with a brown circular logo. Below the logo, there's a brief description: "The furniture shop is the best shop around the city. This is being run...". A "Read more" link is present. Address details are shown: "588 Firwood Road, East Dover, New Jersey, 08753, USA" and phone number "2134212121". A "Visit Shop" button is at the bottom. In the top right corner, there's a "Edit Shop" button.

- Once you enter the **Attributes**, you can sort by name, remove or modify Attributes from there.

The screenshot shows the "Attributes" management page. At the top, there's a header with "Attributes" and a "Click here to sort by Name" link. Below is a table with the following data:

ID	Name	Shop	Values	Actions
3	Color	Clothing Shop	Red, Blue, White	Delete Edit
4	Size	Clothing Shop	S, M, L, XL	Delete Edit
5	Language	Books Shop	English, Arabic, Spanish, French, Hindi, Hebrew	Delete Edit
6	Book Type	Books Shop	Hardcover Book, Picture Book, Paperback Book, PDF Book	Delete Edit

- You can add new attribute by clicking on **Add Attribute** Button. An empty form will appear with some input field and an "Add Attribute" button to save.

The screenshot shows the "Add Attribute" page. It has a header with "Attributes" and a "Click here to add" link. Below is a form with a single input field labeled "Name" and a green "Add Attribute" button.

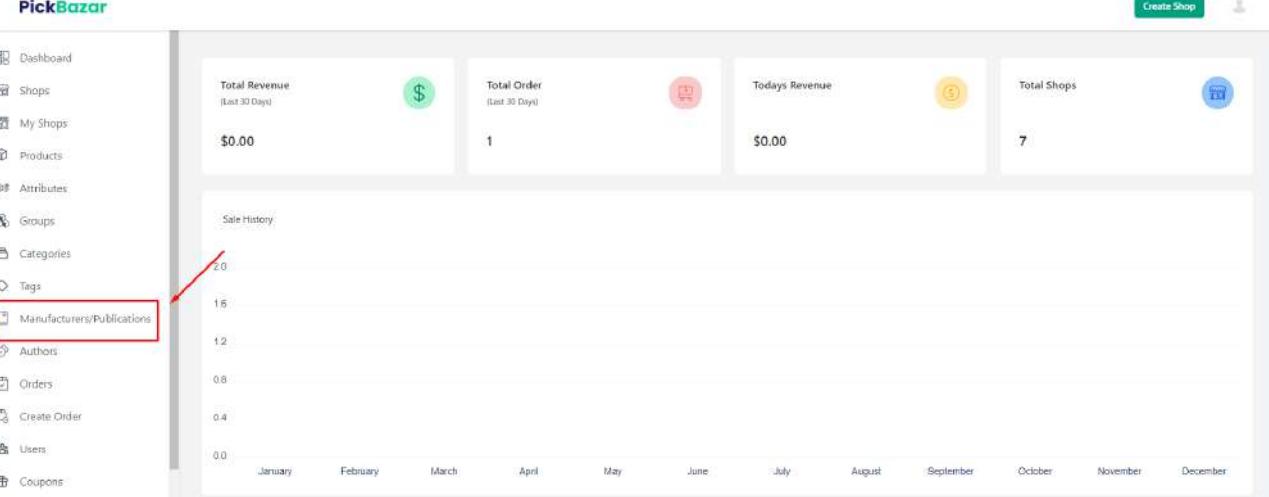
Product Manufacturer/Publications

Admins add manufacturer/publication information to an ecommerce site to provide customers with transparency about the origins of products. This helps build trust and credibility. Manufacturer/publication details offer insights into product quality and authenticity, allowing customers to make informed purchasing decisions.

Manufacturers information and publications contribute to a well-rounded ecommerce platform that prioritizes customer satisfaction and informed shopping.

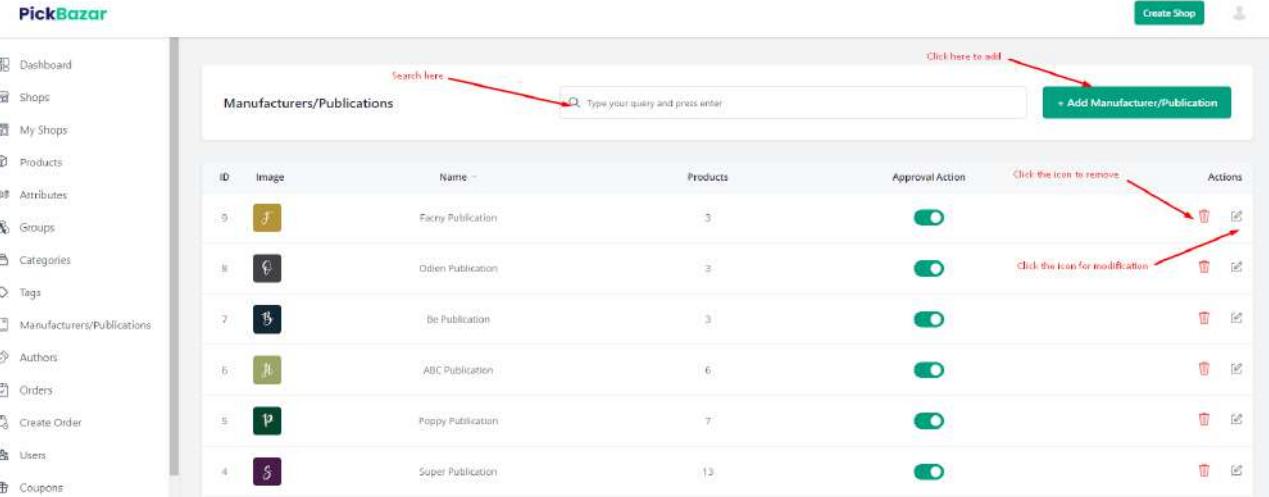
Manufacturer/Publications in Super Admin perspective

- The **Manufacturers/Publications** option is accessible through the admin dashboard.



The screenshot shows the PickBazar admin dashboard. On the left sidebar, there is a list of menu items: Dashboard, Shops, My Shops, Products, Attributes, Groups, Categories, Tags, Manufacturers/Publications (which is highlighted with a red box), Authors, Orders, Create Order, Users, and Coupons. The main content area displays four summary cards: Total Revenue (\$0.00), Total Order (1), Today's Revenue (\$0.00), and Total Shops (7). Below these cards is a chart titled "Sale History" showing monthly sales volume from January to December. A red arrow points from the "Manufacturers/Publications" menu item on the sidebar to the same item on the chart.

- Once you enter the **Manufacturers/Publications**, the **Add Manufacturers/Publications** button will be available for adding additional Manufacturers/Publications.
And you can search, add, remove or modify Manufacturers/Publications from there.



The screenshot shows the Manufacturers/Publications list page in the PickBazar admin. The left sidebar has the same menu as the dashboard. The main area has a search bar labeled "Search here" and a button "+ Add Manufacturer/Publication". Below is a table with the following data:

ID	Image	Name	Products	Approval Action	Click the icon to remove	Actions
9		Faery Publication	3	<input checked="" type="checkbox"/>		
8		Odien Publication	3	<input checked="" type="checkbox"/>		
7		Be Publication	3	<input checked="" type="checkbox"/>		
6		ABC Publication	6	<input checked="" type="checkbox"/>		
5		Poppy Publication	7	<input checked="" type="checkbox"/>		
4		Super Publication	13	<input checked="" type="checkbox"/>		

- The administrator will click **Add Manufacturers/Publications** button after filling out the form to create a new Manufacturers/Publications. Which will appear in the Manufacturers/Publications list.
Here are descriptions of each field:

- [Dashboard](#)
- [Shops](#)
- [My Shops](#)
- [Products](#)
- [Attributes](#)
- [Groups](#)
- [Categories](#)
- [Tags](#)
- [Manufacturers/Publications](#)
- [Authors](#)
- [Orders](#)
- [Create Order](#)
- [Users](#)
- [Coupons](#)

Create Manufacturer/Publication

Logo
 Upload your Manufacturer/Publication logo from here. Dimension should be 160 x 160 px.

Upload an image or drag and drop.

PNG, JPG

Cover Image
 Upload your Manufacturer/Publication cover image from here. Dimension should be 960 x 340 px.

Upload an image or drag and drop.

PNG, JPG

Manufacturer/Publication's Logo : Upload a distinct logo that represents the manufacturer/publication, enhancing brand recognition across the platform.

Manufacturer/Publication's Cover Image : Provide an eye-catching cover image that visually reflects the manufacturer/publication's identity and offerings.

Description
Add Add some information and manufacturer description from here.

Name

Website

Description

Group*

Generate Description With AI

Add New Social Profile

Add Manufacturer/Publication

Name : Input the official name of the manufacturer/publication to ensure accurate identification throughout the site.

slug : Create a URL-friendly slug for the manufacturer/publication's page, optimizing website navigation and search engine visibility.

website : Include the website URL of the manufacturer/publication for direct access to more information about their products or services.

Description : Craft a concise and engaging description that highlights key features, products, or values of the manufacturer/publication. This text helps users understand the essence of the brand. Utilize AI to automatically generate a concise and engaging description of the manufacturer/publication, saving time and ensuring consistent presentation.

Groups : Select relevant shop groups (e.g., grocery, bakery, makeup, clothing) to categorize the manufacturer/publication, ensuring that customers can easily find and navigate products of their interest. Select appropriate groups to categorize the manufacturer/publication's offerings, aiding customers in discovering relevant products.

Add New Social Profile : Click this button to reveal input fields for social media profiles, enabling direct connections between customers and the manufacturer/publication.

Select Social Platform : Choose the relevant social platforms where the manufacturer/publication is active, allowing users to stay updated and engaged.

Add Social Profile URL : Input the URLs to the manufacturer/publication's social media profiles, enabling seamless interactions and access to updates.

Manufacturer/Publications in Vendor perspective

Administrators keep control of the platform while enabling merchants to their publication listings as needed. Admin can edit, delete and add any manufacturer/publication. But vendors can only add manufacturer/publication for a particular shop.

- To add publication, vendors must log in to their own dashboards. Visit his particular store to add publication.

The screenshot shows the PickBazar vendor dashboard. On the left, there is a sidebar with a profile picture, the title 'Store Owner', an email address, a message about running 6 shops, and contact information. Below this is a 'Logged in as Vendor' message. The main area displays a grid of publication cards. Each card includes a thumbnail, the shop name, address, phone number, and four performance metrics: Commission, Sale, Balance, and Withdraw. A red arrow points from the text 'The publications option is accessible through the vendor dashboard.' to the 'Go to a partial Shop' button located above the first row of publications.

Shop Name	Address	Commission	Sale	Balance	Withdraw
Furniture Shop	688 Pinewood Road, East Dover, New Jersey, USA	10	1233	1233	0
Clothing Shop	4885 Spring Street, Lincoln, Illinois, 62657, USA	10	0	0	0
Bags Shop	1740 Bedford Street, Michigan, Alabama, 36001, USA	10	369	369	0
Makeup Shop	2960 Rose Avenue, Metairie, Louisiana, 70002, USA	10	167.4	167.4	0
Bakery Shop	4422 Fort Street, Rocky Mount, Carolina, 27801, USA	10	0	0	0
Grocery Shop	1988 Spinnaker Lane, Freeport, Illinois, 61028, USA	10	0	0	0
Books Shop	44444, Zurich, California, 8021, Switzerland	10	1030.5	1030.5	0
Gadget	1740 Bedford Street, Michigan, Alabama, 36001, USA	10	0	0	0
Medicine	East Avenue 1743, West Tower, Manhattan, New York, USA	10	0	0	0

- The **publications** option is accessible through the vendor dashboard.

- [Dashboard](#)
- [Attributes](#)
- [Products](#)
- [Authors](#)
- [Manufacturers/Publications](#)
- [Orders](#)
- [Refunds](#)
- [Staff](#)
- [Withdrawals](#)
- [Reviews](#)
- [Questions](#)



Furniture Shop

The furniture shop is the best shop around the city. This is being run...

[Read more](#)

588 Firwood Road, East Dover, New Jersey, 08753, USA
213421221

[Visit Shop](#)

- Once you enter the **publications**, you can search, add and sort publications by name but can not able to edit, remove and filter publications.

You can add new publication by clicking on **Add Manufacturer/Publication** Button. An empty form will appear with some input field and an "Add publication" button to save.

Click here and get an empty form to add new manufacturer/publication

Manufacturers/Publications

[+ Add Manufacturer/Publication](#)

PickBazar

[Create Shop](#)

[Visit Site](#)

MAIN

- [Dashboard](#)

SHOP MANAGEMENT

- [Shops](#)
- [My Shops](#)

PRODUCT MANAGEMENT

- [Products](#)
- [Inventory](#)
- [Categories](#)
- [Tags](#)
- [Attributes](#)
- [Manufacturers/Publications](#)
- [Authors](#)

E-COMMERCE MANAGEMENT

- [Taxes](#)
- [Shipments](#)
- [Withdrawals](#)
- [Refunds](#)

LAYOUT/PAGE CONTROL

- [Home pages / Types](#)

Manufacturers/Publications

ID	Name	Products	Approval Action	Slug	Actions
#ID: 9	Fancy Publication	3		fancy-publication	
#ID: 8	Odien Publication	3		odien-publication	
#ID: 7	Be Publication	3		be-publication	
#ID: 6	ABC Publication	6		abc-publication	
#ID: 5	Poppy Publication	7		poppy-publication	
#ID: 4	Super Publication	13		super-publication	
#ID: 3	Wonder Publications	13		wonder-publications	
#ID: 2	Jeremy Publications	11		jeremy-publications	
#ID: 1	Too cool publication	8		too-cool-publication	

© 2023 Pickbazar. All right reserved RedQ

10.2.0

Product Authors

Authors refer to the individuals who have written the content for the books available for sale on the platform. These authors are responsible for creating the written material that makes up the content of the books, including the storyline, characters, information, and ideas presented within the pages. The authors play a crucial role in shaping the quality, uniqueness, and appeal of each book.

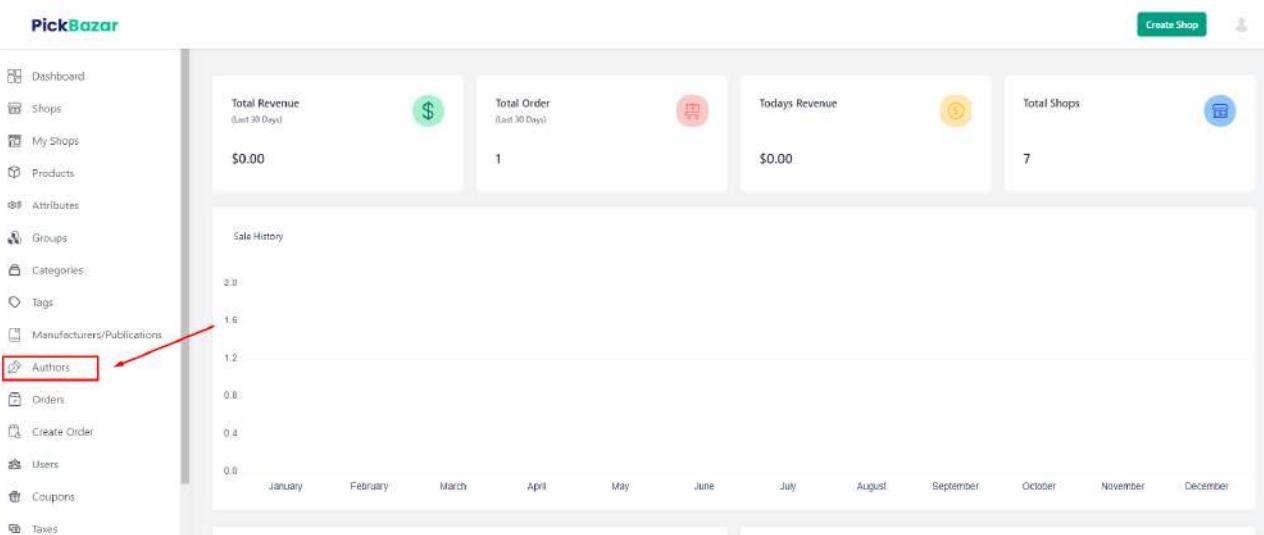
On **Pickbazar**, the information about authors includes their images, names, biographies, date of birth and date of death. This information helps customers understand the background and expertise of the authors, allowing them to make more informed decisions when purchasing books. It can also add a personal touch to the shopping experience, enabling readers to connect with their favorite authors and discover new ones.

Incorporating details about books authors on an ecommerce site can enhance the overall shopping experience for book enthusiasts, fostering a sense of connection between readers and the creative minds behind the stories they love.

Authors in Super Admin perspective

- In **Authors** menu you will get the authors list and you can search, add, remove or modify author from there.

The **Authors** option is accessible through the admin dashboard. Once you enter the **Authors**, the **Add Author** button will be available for adding additional authors.



Authors

ID	Image	Name	Products	Approval Action	Actions
12		Kaitly Jerry	5	<input checked="" type="checkbox"/>	Click to Remove Click to Modify
11		Jakob Dillon	6	<input checked="" type="checkbox"/>	Click to Remove Click to Modify
10		Dorian P Price	9	<input checked="" type="checkbox"/>	Click to Remove Click to Modify
9		Kelly White	7	<input checked="" type="checkbox"/>	Click to Remove Click to Modify
8		Jenifer Wickham	7	<input checked="" type="checkbox"/>	Click to Remove Click to Modify
7		Sharone C Stone	4	<input checked="" type="checkbox"/>	Click to Remove Click to Modify
6		Kelton Benjamin	6	<input checked="" type="checkbox"/>	Click to Remove Click to Modify

- The administrator will click **add author** button after filling out the form to create a new author. Which will appear in the author list.

Authors

ID	Image	Name	Products	Approval Action	Actions
12		Kaitly Jerry	5	<input checked="" type="checkbox"/>	Click to Remove Click to Modify
11		Jakob Dillon	6	<input checked="" type="checkbox"/>	Click to Remove Click to Modify
10		Dorian P Price	9	<input checked="" type="checkbox"/>	Click to Remove Click to Modify
9		Kelly White	7	<input checked="" type="checkbox"/>	Click to Remove Click to Modify
8		Jenifer Wickham	7	<input checked="" type="checkbox"/>	Click to Remove Click to Modify
7		Sharone C Stone	4	<input checked="" type="checkbox"/>	Click to Remove Click to Modify

- Here are descriptions of each field:

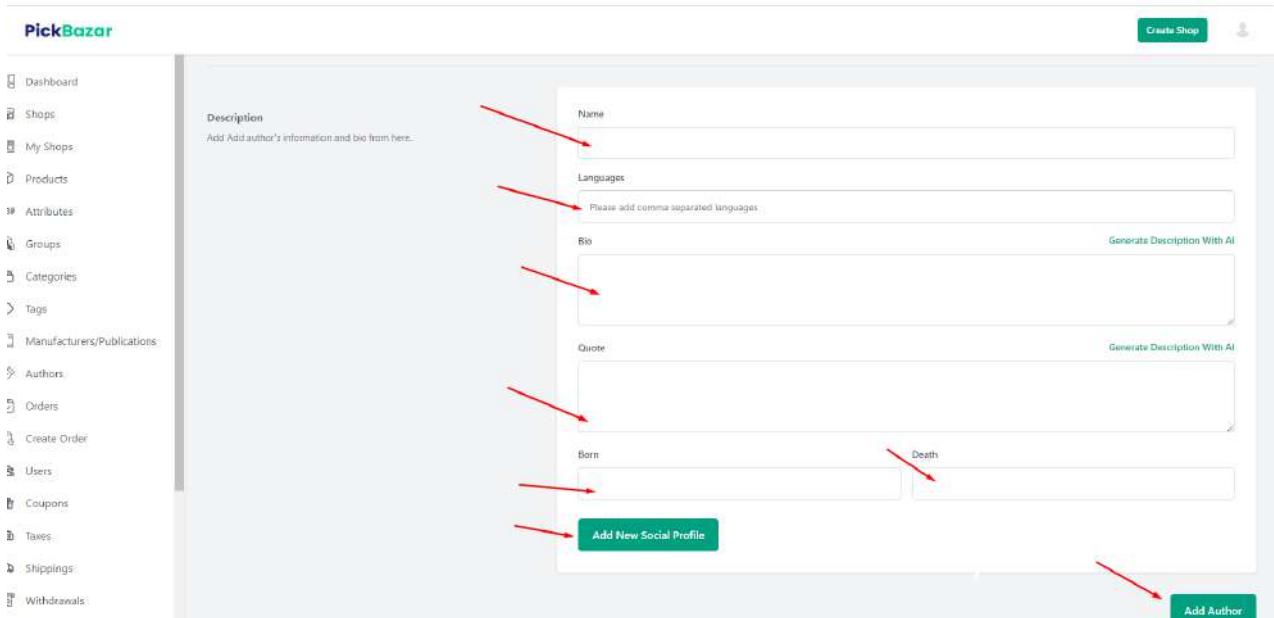
Create Author

Image
Add author's profile image from here. Dimension should be 480 x 450 px.

Cover Image
Add author's cover image from here.

Image : Upload a representative image of the author to personalize their presence and establish recognition throughout the site.

Cover Image : Provide a cover image that encapsulates the author's style or complements their works, offering an appealing visual introduction.



Name : Enter the full name of the author to ensure accurate identification and enable seamless searchability.

slug : Create a user-friendly URL slug for the author's page, enhancing website navigation and search engine optimization.

Languages : Specify the languages in which the author's works are available, aiding users in discovering content in their preferred language.

Bio : Craft a concise biography of the author, offering insights into their background, achievements, and writing journey.

Quote : Share a compelling quote from the author that captures their essence and resonates with readers.

Born : Enter the author's date of birth to provide readers with a glimpse into their life timeline.

Death : If applicable, include the author's date of death to provide comprehensive biographical information.

Select Social Platform : Choose the relevant social media platforms where the author has a presence, facilitating direct engagement with readers.

Add Profile URL : Include the URLs to the author's social media profiles, enabling users to connect and follow their updates easily.

Authors in Vendor perspective

Administrators keep control of the platform while enabling merchants to their author listings as needed. Admin can edit, delete and add any author. But vendors can only add author for a particular shop.

- To add author, vendors must log in to their own dashboards. Visit his particular store to add author.

The screenshot shows the PickBazar vendor dashboard. On the left, there's a sidebar with options like Dashboard, My Shops (which is selected), Message, and Store Notice. The main area displays a grid of nine shop cards. Each card includes a thumbnail, the shop name, address, and some stats (Commission, Sale, Balance, Withdraw). A red arrow points from the 'Go to a partial Shop' button at the top to the 'Clothing Shop' card. The 'Clothing Shop' card shows the following details:

Clothing Shop	
Address: 4885 Spring Street, Lincoln, Illinois, 62655...	Phone: +212 90121221
Commission: 10	Sale: 0
Balance: 0	Withdraw: 0

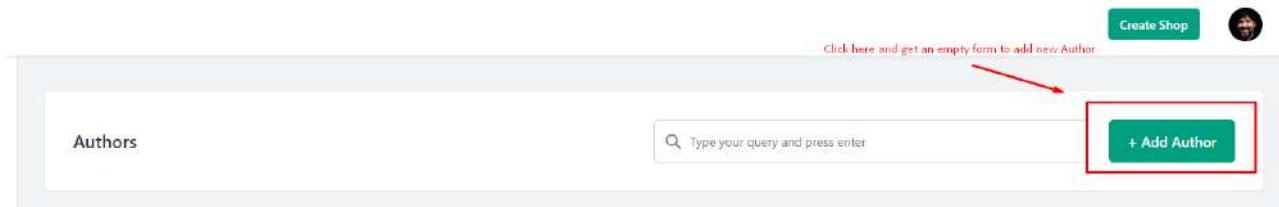
- The **authors** option is accessible through the vendor dashboard.

The screenshot shows the PickBazar vendor dashboard with the 'Authors' section selected in the sidebar. The main area shows a card for a 'Furniture Shop'. The 'Authors' section contains the following information:

Author Name	Commission	Sale	Balance	Withdraw
John Doe	10	167.4	167.4	0
Jane Smith	10	1030.5	1030.5	0

- Once you enter the **authors**, you can add, search and sort authors by name but can not able to edit, remove and filter authors.

You can add new author by clicking on **Add Author** Button. An empty form will appear with some input field and an "Add author" button to save.



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.

Image	Name	Group	Product Type	Price/Unit	Quantity	Status	Actions
	Nair Hair Removal Cream	Grocery	simple	\$8.00	50	<button>publish</button>	
	Espa Men Dual Action	Grocery	simple	\$6.00	50	<button>publish</button>	
	Human Nature Shaving Cream	Grocery	simple	\$6.00	50	<button>publish</button>	
	Longmarket Barber Aftershave Balm	Grocery	simple	\$6.00	40	<button>publish</button>	
	Pacific Natural Shaving Cream	Grocery	simple	\$6.00	50	<button>publish</button>	
	Nuace Men Facial Wash	Grocery	simple	\$3.00	50	<button>publish</button>	
	Garnier Pure Active	Grocery	simple	\$2.00	50	<button>publish</button>	
	Frei Öl	Grocery	simple	\$2.00	50	<button>publish</button>	
	Face Republic Sun Gel	Grocery	simple	\$2.00	50	<button>publish</button>	

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.

Image	Name	Group	Product Type	Price/Unit	Quantity	Status	Actions
	Nair Hair Removal Cream	Grocery	simple	\$6.00	50	publish	trash edit
	Nuace Men Facial Wash	Grocery	simple	\$3.00	50	publish	trash edit
	Garnier Pure Active	Grocery	simple	\$2.00	50	publish	trash edit
	Fris Oil	Grocery	simple	\$2.00	50	publish	trash edit
	Face Republic Sun Gel	Grocery	simple	\$2.00	50	publish	trash edit

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

Image	Name	Group	Product Type	Price/Unit	Quantity	Status	Actions
	Nair Hair Removal Cream	Grocery	simple	\$6.00	50	publish	trash edit
	Nuace Men Facial Wash	Grocery	simple	\$3.00	50	publish	trash edit
	Garnier Pure Active	Grocery	simple	\$2.00	50	publish	trash edit
	Fris Oil	Grocery	simple	\$2.00	50	publish	trash edit
	Face Republic Sun Gel	Grocery	simple	\$2.00	50	publish	trash edit

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.

The image consists of three vertically stacked screenshots of the PickBazar admin dashboard, each highlighting specific features with red numbered arrows.

Screenshot 1: Attributes Section

- Arrow 1:** Points to the 'Products' menu item in the left sidebar.
- Arrow 2:** Points to the 'Actions' dropdown menu at the top right of the table.
- Arrow 3:** Points to the 'Export Attributes' button in the 'Export/Import' modal.

ID	Name	Values
5	colors	red, green

Screenshot 2: Products Section

- Arrow 1:** Points to the 'Products' menu item in the left sidebar.
- Arrow 2:** Points to the 'Actions' dropdown menu at the top right of the table.
- Arrow 3:** Points to the 'Export Product Variations' button in the 'Export/Import' modal.

Image	Name	Group	Product Type	Price/Unit	Quantity	Status	Actions
	Nair Hair Removal Cream	Grocery	simple	\$10.00	50	published	
	Huace Men Facial Wash	Grocery	simple	\$3.00	50	published	
	Garner Pure Active	Grocery	simple	\$2.00	50	published	
	Frei Oil	Grocery	simple	\$2.00	50	published	
	Face Republic Sun Gel	Grocery	simple	\$2.00	50	published	

Screenshot 3: Attributes Section

- Arrow 1:** Points to the 'Products' menu item in the left sidebar.
- Arrow 2:** Points to the 'Actions' dropdown menu at the top right of the table.
- Arrow 3:** Points to the 'Import Attributes' button in the 'Export/Import' modal.

ID	Name	Values
5	colors	red, green

PickBazar

Dashboard Attributes Products Orders Staffs Withdraws

Products

Image Name Group Product Type Price/Unit Quantity Status Actions

Image	Name	Group	Product Type	Price/Unit	Quantity	Status	Actions
	Nair Hair Removal Cream	Grocery	simple	\$10.00	50	publish	
	Nuace Men Facial Wash	Grocery	simple	\$3.00	50	publish	
	Garnier Pure Active	Grocery	simple	\$2.00	50	publish	
	Fast Oil	Grocery	simple	\$2.00	50	publish	
	Face Republic Sun Gel	Grocery	simple	\$2.00	50	publish	

Export/Import

Import Products Import Product Variations Export Products Export Product Variations

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

Create Shop

PickBazar

Dashboard Attributes Products Orders Staffs Withdraws

Attributes

Order By DESC + Add Attribute

ID	Name	Values	Actions
5	color	red, green	

Export/Import

Import Attributes Export Attributes

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

Create Shop

PickBazar

The screenshot shows the PickBazar admin interface. On the left sidebar, under the 'Products' section, there is a red arrow labeled '1' pointing to the 'Products' link. At the top right, there is a search bar with placeholder text 'Type your query and press enter', a green button '+ Add Product', a 'Filter' dropdown, and a small icon with a question mark and a red arrow labeled '2'. A modal window titled 'Export/Import' is open, containing four options: 'Import Products', 'Import Product Variations', 'Export Products', and 'Export Product Variations'. Red arrows labeled '3' point to the 'Import Product Variations' button and the 'Grocery' group name in the products list.

Image	Name	Group	Product Type	Price/Unit	Quantity	Status	Action
	Hair Hair Removal Cream	Grocery	simple	\$8.00	50	publish	
	Nuace Men Facial Wash	-Grocery	simple	\$3.00	50	publish	
	Garnier Pure Active	Grocery	simple	\$2.00	50	publish	
	Free Oil	Grocery	simple	\$2.00	50	publish	
	Face Republic Sun Gel	Grocery	simple	\$2.00	50	publish	

PickBazar

This screenshot is identical to the one above, showing the PickBazar admin interface. It features the same sidebar with a red arrow '1' pointing to 'Products', the same top navigation with a question mark icon and a red arrow '2', and the same 'Export/Import' modal with its four options. Red arrows labeled '3' point to the 'Import Product Variations' button and the 'Grocery' group name in the products list.

Image	Name	Group	Product Type	Price/Unit	Quantity	Status	Action
	Hair Hair Removal Cream	Grocery	simple	\$8.00	50	publish	
	Nuace Men Facial Wash	-Grocery	simple	\$3.00	50	publish	
	Garnier Pure Active	Grocery	simple	\$2.00	50	publish	
	Free Oil	Grocery	simple	\$2.00	50	publish	
	Face Republic Sun Gel	Grocery	simple	\$2.00	50	publish	

NB: Here, type_id is the group id like Grocery or clothing id. You can get type info from [api_url/types](#) url;

Introduction to Features

Tax Management

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

The screenshot shows a table titled "Taxes" with a search bar at the top. The table has columns for ID, Name, Rate (%), Country, City, State, ZIP, and Actions. A single row is visible with ID 1, Name Global, Rate 2, and the Actions column showing a trash bin icon and an edit icon. Red arrows point to the search bar, the "+ Add Tax" button, the Rate (%) column header, and the Actions column header.

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

+ Add Tax

Like any other company, Pickbazar must pay taxes to the government. The government uses taxes as a means of raising money to support public infrastructure and services like roads, schools, and hospitals.

Pickbazar is an online marketplace that links buyers and sellers so that anyone may purchase and sell goods. The merchant makes money when a consumer uses Pickbazar to buy something, and as a result, is obligated to pay taxes on that money.

Tax in Super Admin perspective

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

The screenshot shows a table titled "Taxes" with a search bar at the top. The table has columns for ID, Name, Rate (%), Country, City, State, ZIP, and Actions. A single row is visible with ID 1, Name Global, Rate 2, and the Actions column showing a trash bin icon and an edit icon. Red arrows point to the search bar, the "+ Add Tax" button, the Rate (%) column header, and the Actions column header.

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

+ Add Tax

To add a new tax, the super administrator must now specify its name and value.

Create New Tax

Information

Add your tax information from here

Name: new tax
Rate: 10
Country:
City:
State:
ZIP:
Add Tax

The list of taxes will display the newly established tax.

ID	Name	Rate (%)	Country	City	State	ZIP	Actions
2	new tax	10					
1	Global	2					

He may access the settings from the admin dashboard and configure the tax he wants to enable.

PickBazar

- Dashboard
- Shops
- My Shops
- Products
- Attributes
- Groups
- Categories
- Tags
- Manufacturers/Publications
- Authors
- Orders
- Create Order
- Users
- Coupons
- Taxes
- Shipping
- Withdrawals
- Message
- Refunds
- Questions
- Store Notice
- Reviews
- Settings

Payment
Configure Payment Option

Site Slogan: Your next ecommerce
Currency: US Dollar
Minimum Order Amount: 0
Wallet Currency Ratio: 1
Sign Up Points: 100
Maximum Question Limit: 5
Use CTP at checkout:
Tax Class: new tax
Shipping Class: Global
Enable Free Shipping:
Enable Cash On Delivery:
Select Payment Gateway: Stripe

Tax Class

new tax

Tax in Vendor perspective

When a consumer adds items to his cart and checks out, he may discover that the tax has been added to the subtotal.

In order to place an order, the consumer must now click "place order."

PickBazar

Dashboard Shops My Shops Products Attributes Groups Categories Tags Manufacturers/Publications Authors Orders Create Order Users Coupons

Customer

Contact Number

Billing Address

Your Order

8 x Celery Stick 1lb	\$40.00	
3 x Clementines 1lb	\$7.50	
Sub Total	\$47.50	
Tax	Calculated at checkout	
Estimated Shipping	Calculated at checkout	

Check Availability

PickBazar

Dashboard Shops My Shops Products Attributes Groups Categories Tags Manufacturers/Publications Authors Orders Create Order Users Coupons Taxes

Customer

Contact Number

Billing Address

Your Order

8 x Celery Stick 1lb	\$40.00	
3 x Clementines 1lb	\$7.50	
Sub Total	\$47.50	
Tax	\$0.95	
Shipping	\$50.00	
Discount (-\$0.00) X	-\$4.00	
Total	\$94.45	
Wallet points	0	
Wallet currency	\$0.00	

Do you want to use wallet?

Choose Payment Method

Cash Cash On Delivery

Please pay with cash

Place Order

Shipping Management

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

Shipments

Type your query and press enter

+ Add Shipping

ID	Name	Amount	Global	Type	Actions
1	Global	50	1	fixed	

When anything needs to be physically moved from an online store to a customer's delivery location, it is referred to as shipping. This mostly entails taking an order, processing it, and getting it ready for delivery.

Free Shipping

Online companies who provide free delivery do so when shoppers buy products from them. Online sellers encourage customers to buy specific products by reducing delivery fees from those products.

Cash on delivery

Cash on Delivery (C.O.D.) is a word used frequently in commerce to denote that payment for products must be made at the time of delivery. The main method of payment is cash.

Shipping and Billing Address

An address selected for shipping is where you will send the order. The address associated with the customer's payment method is known as the billing address. But not usually, billing and shipping addresses are frequently the same.

Shipping process in general

On Pickbazar, the merchants who are selling their goods on the site are the ones who receive the shipping money. When clients place an order, the vendors charge them shipping fees, which they then keep. The vendors utilize the shipping charges to cover the expenses incurred when sending the customers' orders of the items.

You could get access to the shipping rates and shipping cost information for each seller on the platform if you're a Super Admin. This might assist you in keeping an eye on the shipping charges and fees to make sure they comply with the platform's shipping regulations.

On the shipping charges that the merchants receive through Pickbazar, a commission fee may be assessed. This commission can be used to pay for platform management expenses, customer service expenses, and platform development expenses for new products and services. Depending on the vendor's subscription plan and the number of sales generated via the platform, the commission portion may change.

The platform may impose a commission on these payments to offset the costs of running the platform. Ultimately, the shipping fees are collected by the sellers. As a Super Admin, you would be in charge of keeping an eye on the shipping charges and fees information and making sure that it complies with the platform's rules.

Shipping in Super Admin perspective

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

ID	Name	Amount	Global	Type	Actions
1	Global	50	1	fixed	

+ Add Shipping

Once you click on "Add Shipping," the following form will display for you to fill out.

Create New Shipping

Description
Add your shipping description and necessary information from here.

Name

Type Free Fixed Percentage

Amount

Add Shipping

ID	Name	Amount	Global	Type	Actions
1	Global	50	1	fixed	

The shipment name, quantity, and type are necessary. Once added, this will show up on the shipment list.

The administrator will then pick the shipping class name from the dropdown menu under "settings" after that. Also, he might provide the merchant the option of free shipment.

Shipments

Type your query and press enter

+ Add Shipping

ID	Name	Amount	Global	Type	Actions
1	Global	50	1	fixed	

Shipping in Vendor perspective

Upon the purchase of the product, the seller will receive the shipping cost together with the subtotal.

Withdrawal Management

Withdrawal Lists will be here.

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

Shop ID	Shop	Amount	Created	Payment Method	Status	Actions
6	Grocery Shop	\$100.00	a few seconds ago	via Bank	Pending	

Withdrawal details

The screenshot shows the PickBazar Admin Dashboard. On the left sidebar, there are several sections: MAIN (Dashboard), SHOP MANAGEMENT (Shops, My Shops), PRODUCT MANAGEMENT (Products, Inventory, Categories, Tags, Attributes, Manufacturers/Publications, Authors), and E-COMMERCE MANAGEMENT (Taxes, Shipments). The main content area is titled "Withdrawal Information". It shows a withdrawal request with the following details: Amount: 100, Payment Method: via Bank, Status: Pending. There is a "Change Status" button. Below this, a "Details" box shows account information: Account No : 9973 272001 23434, Account Name : Jhon Doe, Routing number : 887300. A "Note" box contains placeholder text about the use of Lorem Ipsum in web design. At the bottom, there is a footer with the text "©2023 Pickbazar. All right reserved RedQ" and a version number "10.2.0".

The procedure by which a seller can move their profits or balance from the Pickbazar platform to their own bank account or another payment method is known as a withdrawal mechanism in Pickbazar.

In most withdrawal systems, the seller must ask to remove money from their account balance. With the seller's identification being confirmed, the platform may impose conditions regarding the minimum withdrawal amounts or withdrawal costs.

The platform will transfer the money to the seller's chosen payment method as soon as the withdrawal is accepted. The payment method and the platform's rules will determine how long this procedure will take, which might be several business days.

The ability for sellers to obtain their revenues and use them for either personal or professional objectives makes withdrawal mechanisms an essential component.

Withdrawal in Vendor perspective

The store owner must first log in, then from the dashboard, we must locate "my shops". The seller's shop is where he may locate withdrawal options. The options below will be displayed if he visits his own store.

The screenshot shows the PickBazar Vendor Dashboard. On the left sidebar, there is a profile picture of a store owner and a status indicator "Enabled". The main content area is titled "My Shops". It shows a grid of six active shops: Furniture Shop (Active), Clothing Shop (Active), Bags Shop (Active), Makeup Shop (Inactive), Bakery Shop (Active), and Grocery Shop (Inactive). Each shop has a small icon and a status badge. At the top of the "My Shops" section, there are tabs for "My Shops", "Message", and "Store Notice".



- [Dashboard](#)
- [Attributes](#)
- [Products](#)
- [Authors](#)
- [Manufacturers/Publications](#)
- [Orders](#)
- [Refunds](#)
- [Staff](#)
- [Withdrawals](#)
- [Reviews](#)
- [Questions](#)
- [Store Notice](#)

The screenshot shows a seller's shop dashboard. On the left, there's a sidebar with navigation links. In the center, there's a large image of a living room with a teal sofa, wooden shelves, and various decorative items. Below the image, there are summary statistics: 55 Total Products, \$0.00 Gross Sales, and 10% Admin Commission Rate. At the bottom right, it says "Registered Since June 27, 2021". A red arrow points to the "Withdrawals" link in the sidebar.

The seller may submit withdrawal requests under the withdrawals area.

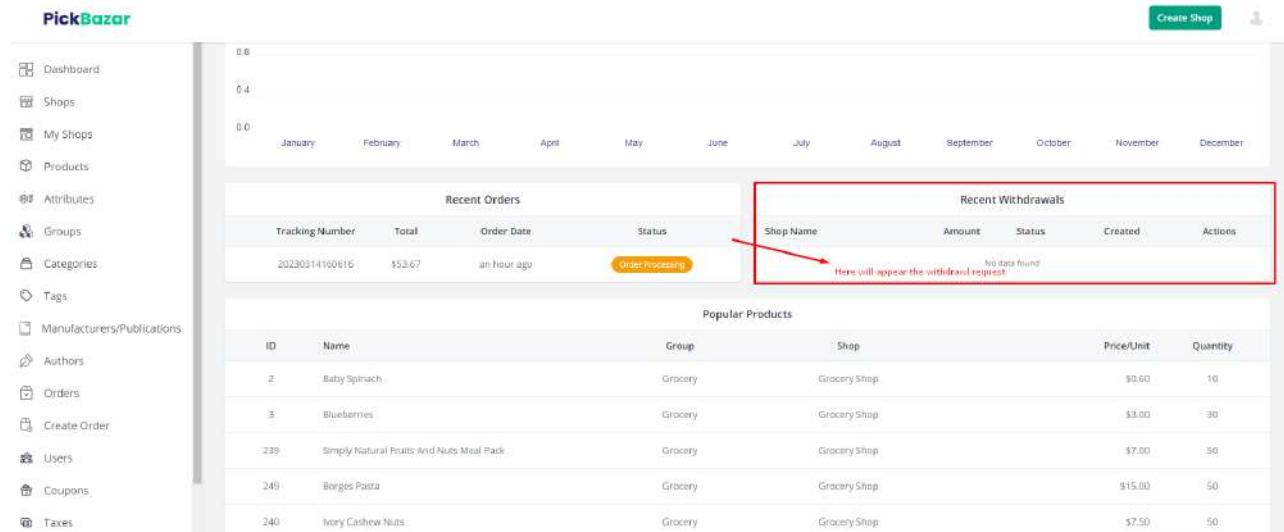
The screenshot shows the "Withdrawals" section. It has a table with columns for Shop Name, Amount, Status, and Created. A red arrow points to the green "Request Withdrawal" button at the top right of the table area.

The seller can select **request withdrawals** and enter a request for a lesser quantity than what is now available.

The screenshot shows the "Create Withdrawal" form. It includes fields for Description, Amount (with a note "Write an amount which is less than the available balance"), Payment Method, Details, Note, and a "Request Withdrawal" button. A red arrow points to the "Amount" field, and another red arrow points to the "Request Withdrawal" button.

Withdrawal in Super Admin perspective

Requested Withdrawals will show up in the "recent Withdrawals" section of the admin dashboard. The administrator can use the action function and change the withdrawal request's status from pending to approved by clicking on **change status** button..



The screenshot shows the PickBazar admin dashboard. On the left is a sidebar with various menu items: Dashboard, Shops, My Shops, Products, Attributes, Groups, Categories, Tags, Manufacturers/Publications, Authors, Orders, Create Order, Users, Coupons, and Taxes. At the top right are buttons for 'Create Shop' and user profile. Below the sidebar are two main sections: 'Recent Orders' and 'Recent Withdrawals'. The 'Recent Orders' section has columns for Tracking Number, Total, Order Date, and Status (with a dropdown currently set to 'Order Processing'). The 'Recent Withdrawals' section is highlighted with a red border and contains columns for Shop Name, Amount, Status, Created, and Actions. A note at the bottom of this section says 'Here will appear the withdrawal request' and 'No data found'. Below these sections are 'Popular Products' tables.

Recent Withdrawals						
Shop Name	Amount	Status	Created	Actions		
Chawkbazar Vendor shop	\$500.00	Approved	a year ago			
Chawkbazar Vendor shop	\$250.00	On hold	a year ago			
Chawkbazar Vendor shop	\$6,500.00	Rejected	a year ago			
Chawkbazar Vendor shop	\$600.00	On hold	a year ago			
Chawkbazar Vendor shop	\$10.00	Pending	a few seconds ago			

Recent Withdrawals

Shop Name	Amount	Status	Created	Actions
Chawkbazar Vendor shop	\$500.00	Approved	a year ago	(o)
Chawkbazar Vendor shop	\$250.00	On hold	a year ago	(o)
Chawkbazar Vendor shop	\$6,500.00	Rejected	a year ago	(o)
Chawkbazar Vendor shop	\$600.00	On hold	a year ago	(o)
Chawkbazar Vendor shop	\$10.00	Pending	a minute ago	(o)

Withdrawal Information

Amount	10	Pending	Change Status
Payment Method	N/A		
Status	Pending		

Withdrawal Information

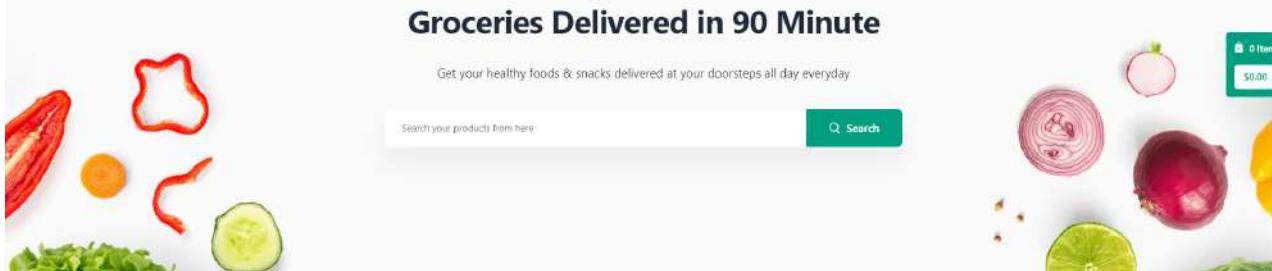
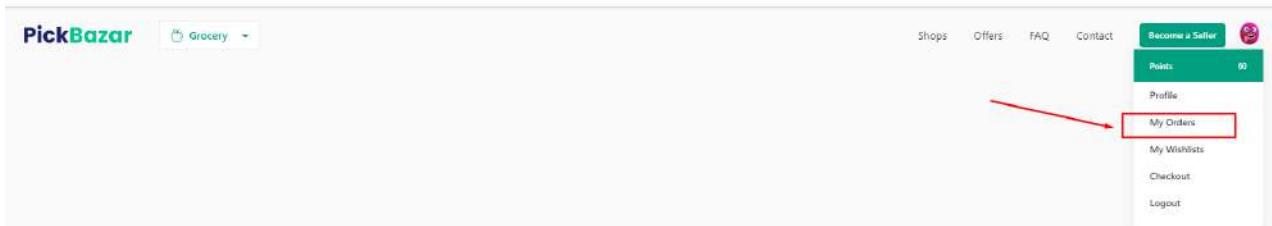
Amount	10	Approved	Change Status
Payment Method	N/A		
Status	Pending		

Refund

A refund is the procedure of returning money to a consumer who has made a purchase of a good or service and is unhappy with it or runs into trouble with the transaction. In order to ensure customer pleasure and uphold client confidence in the company, refunds are a crucial component of customer care in eCommerce.

Refund in Customer perspective

- After successfully placing a purchase, the customer will select the profile icon in the top right corner. He must choose **My Orders** from the drop-down option.



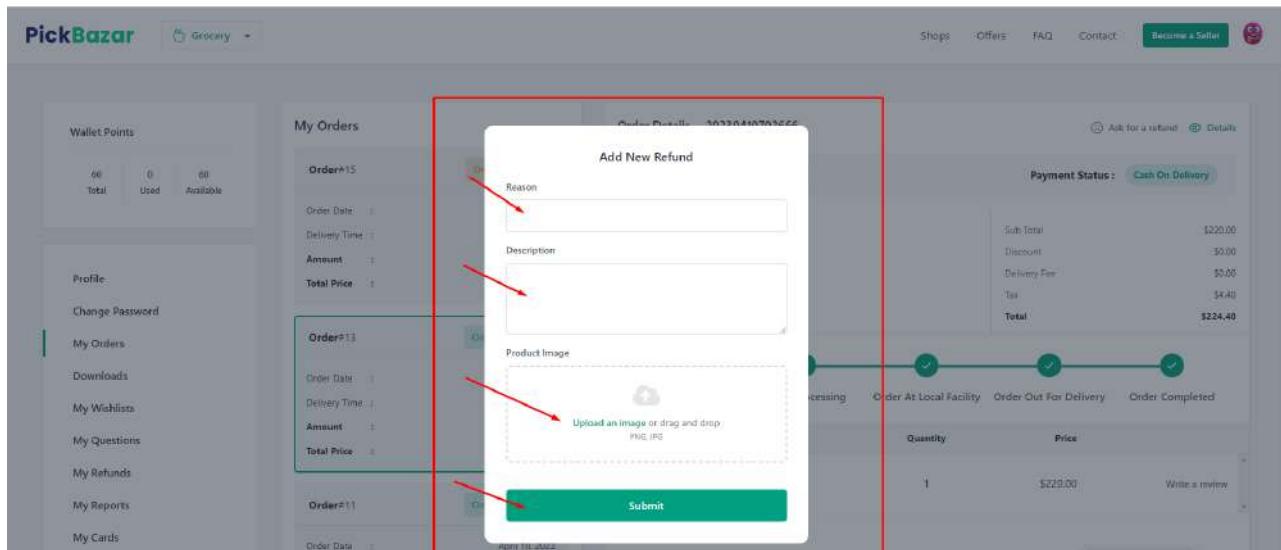
- The consumer will discover a **order details** segment after clicking on individual order under **My Orders**.

Order #	Status	Date	Amount	Total Price
Order#15	Order Processing	August 16, 2023	\$20.00	\$70.40
Order#13	Order Completed	April 10, 2023	\$220.00	\$224.40
Order#11	Order Completed	April 10, 2023		

- The consumer will find **Ask for a refund** option in **Order details** section which is at the top right corner of each individual order details.

Order Status	Payment Status
Order Completed	Cash On Delivery

- After selecting **Ask for a refund**, a modal window with a few input fields will show up for you to fill out and submit.



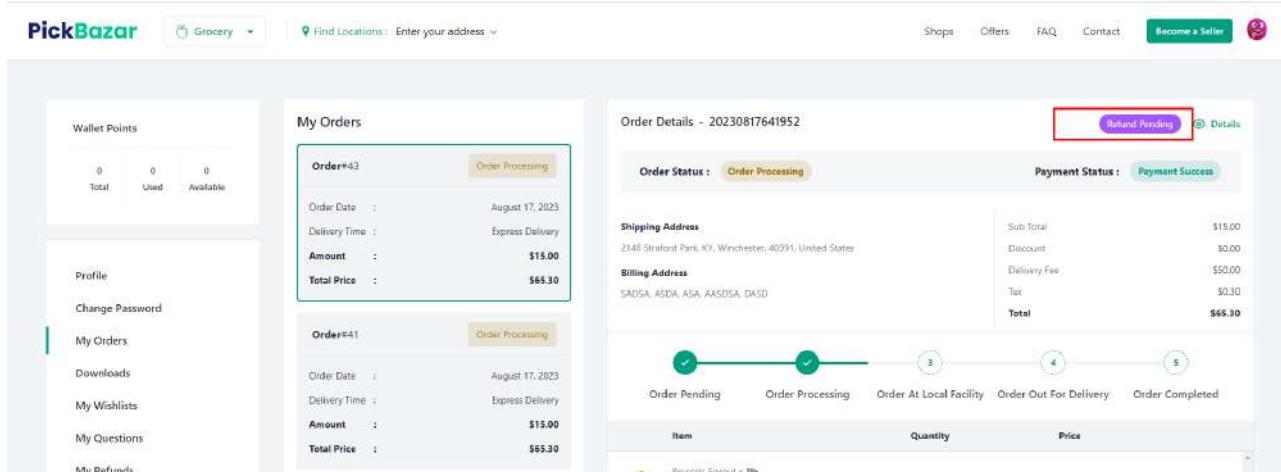
Reason text area enabling customers to provide clear and specific details within a reasonable character limit to help the support team understand the reason for the refund.

Description text area further provides customers with the opportunity to provide additional context or details related to their refund request. Customers can elaborate on specific issues or concerns regarding the ordered product.

Product Image allows customers to upload image files (e.g., JPEG, PNG) with a clear view of the product's condition, highlighting any defects, damages, or discrepancies.

Submit button confirms the customer's refund request and sends the provided information to the customer support team for further processing.

- When a refund request is made, the **Ask for a Refund** option changes to **Refund Pending** until admin approves or rejects the request.



Refund in Super Admin perspective

- In **Refund** menu you will get the refund list which refund has been requested by a customer.

The screenshot shows the PickBazar dashboard. On the left, there is a sidebar with various menu items: Groups, Categories, Tags, Manufacturers/Publications, Authors, Orders, Create Order, Users, Coupons, Taxes, Shipments, Withdrawals, Message, Refunds (which is highlighted with a red box and a red arrow), and Questions. The main area displays four key metrics: Total Revenue (\$0.00), Total Order (1), Today's Revenue (\$0.00), and Total Shops (7). Below these metrics is a chart titled "Sale History" showing monthly sales volume from January to December. The "Refunds" menu item is clearly identified as the target for this section.

- Listed refund item status is initially pending. To change the status admin needs to click on **action** icon.

The screenshot shows the "Refunds" list page. The sidebar on the left includes: Dashboard, Shops, My Shops, Products, Attributes, Groups, Categories, Tags, Manufacturers/Publications, Authors, Orders, Create Order, and The main content area is titled "Refunds" and lists two items:

ID	Reason	Customer Email	Amount	Tracking Number	Created	Order Date	Status	Actions
3	broken	customer@demo.com	\$220.00	20230418702666	A Few Seconds Ago	4 Months Ago	pending	
1	damaged	customer@demo.com	\$8.00	20230818284645	2 Minutes Ago	3 Minutes Ago	pending	

A red box highlights the "Status" column for the first item, and a red arrow points to the "pending" status value. Another red arrow points to the edit icon in the "Actions" column for the same row.

- Admin can select the status from dropdown and click on **Change Status** as following:

Refund ID - 3 (Pending)

Tracking Number : 20230410702665
Order Status :
Customer Email : customer@demo.com
Contact Number : 19365141641631

Reason : broken
Description : broken
Images :

Change status and click here

Change Status

- Status of the refund will be updated in the refund table.

ID	Reason	Customer Email	Amount	Tracking Number	Created	Order Date	Status	Actions
3	broken	customer@demo.com	\$220.00	20230410702666	7 Minutes Ago	4 Months Ago	approved	
1	damaged	customer@demo.com	\$8.00	20230818284645	9 Minutes Ago	9 Minutes Ago	rejected	

Introduction to Features

In this chapter we will mainly discuss about how we can build pages more specially to fullfill our needs.

Manage Product Type

In **Homepages / Groups** menu you will get the product types and you can add, remove or modify product type from there. Products group also act as a controller to setup a homepage. So, that all the products of similar group (e.g. Grocery) can be unified under one tree.

Types

Type your query and press enter

+ Add Types

ID	Name	Icon	Actions
6	Furniture	Chair icon	trash edit
5	Clothing	Dress icon	trash edit
4	Bags	Luggage icon	trash edit
3	Makeup	Brush icon	trash edit
2	Bakery	Cake icon	trash edit
1	Grocery	Apple icon	trash edit

Manage FAQs

In **FAQs** menu super admin will get the form to add, remove or modify FAQs from there. In vendor end, shop owner can also add specific FAQs based on their specific shop.

PickBazar

Search your route...

Create Shop Visit Site Language English Admin Name super admin

FAQs

Type your query and press enter

+ Add FAQs

ID	Title	Description	Type	Issued By	Actions
#ID: 3	Where does it come from?	contrary to popular belief, Lorem Ipsum is not simply random text. It ...	global	Super Admin	edit trash more
#ID: 2	What is Lorem Ipsum?	Loem Ipsum is simply dummy text of the printing and typesetting in...	global	Super Admin	edit trash more
#ID: 1	Why do we use it?	It is a long established fact that a reader will be distracted by the re...	global	Super Admin	edit trash more

For More Details [FAQS](#)

Manage Terms & Conditions

In **Terms & Conditions** menu super admin will get the form to add, remove or modify terms & conditions from there. There are some keynotes about this,

- Super admin can enable the accessibility for vendors to add terms & conditions.

- Vendor's added terms & conditions must have to go through approval phase via Super admin.

For More Details [Terms & Conditions](#)

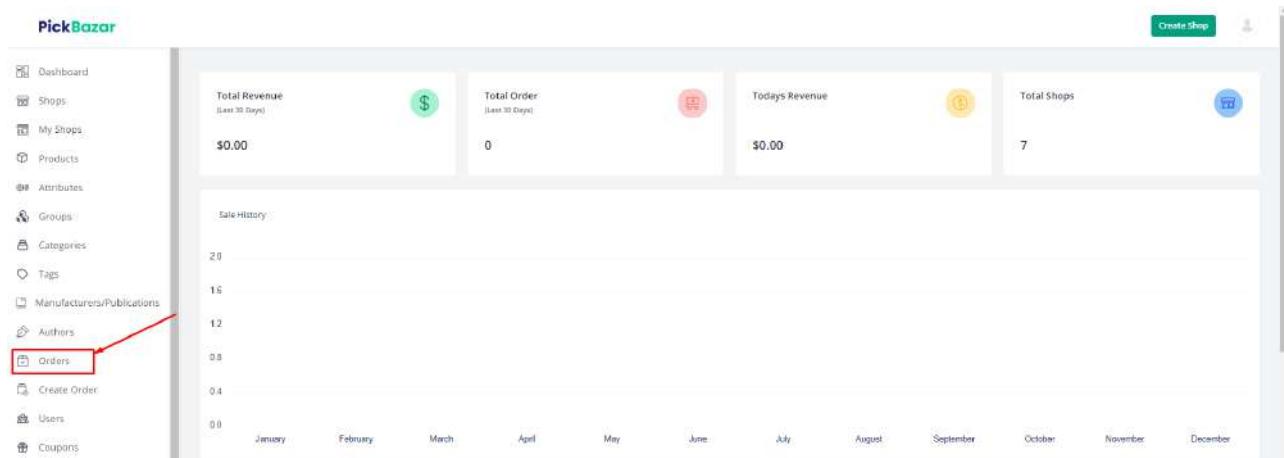
Orders Management

Order refers to a customer's request to purchase one or more products from the platform. It represents a completed transaction where the customer has selected items, provided shipping and payment information, and confirmed their purchase.

The **orders** feature in Pickbazar is used to manage and track these transactions.

Super Admin

- The **Orders** option is accessible through the admin dashboard.



- Once you enter the **Orders**, you can sortOrders by name, price/unit, quantity and status. You can also search, edit, remove and filterOrders.

Here are the descriptions of types of sorting:

The screenshot shows a table of orders with the following columns: Tracking Number, Delivery Fee, Total, Order Date, Status, Shipping Address, and Actions. A red box highlights the 'Status' column, which contains 'Order Completed' status for all four rows. Red arrows point from the text definitions to the corresponding UI elements: 'Search by Tracking Number' points to the search bar at the top; 'Sort by Total' and 'Sort by Order Date' point to the sort dropdowns; 'Status' points to the status column header; 'Actions' points to the 'Action page' link; and 'Export Orders' points to the export button.

Orders							
	Tracking Number	Delivery Fee	Total	Order Date	Status	Shipping Address	Actions
[+]	20230410702666	\$0.00	\$224.40	4 months ago	Order Completed Status will update according to action	2148 Stratford Park, Winchester, KY, 40391, United States	Action page
[+]	20230410885225	\$0.00	\$122.40	4 months ago	Order Completed	2148 Stratford Park, Winchester, KY, 40391, United States	Action page
[+]	20230410230333	\$0.00	\$229.50	4 months ago	Order Completed	2148 Stratford Park, Winchester, KY, 40391, United States	Action page
[+]	20230410631164	\$0.00	\$102.00	4 months ago	Order Completed		Action page

Search : A functionality allowing administrators to find specific orders by entering the tracking number, aiding in quick order retrieval and management.

Sort by Total : A feature enabling administrators to arrange orders in ascending or descending order based on the total amount spent by customers, aiding in financial analysis and decision-making.

Sort by Order Date : An option permitting administrators to organize orders chronologically, either in ascending or descending order by their respective order dates, facilitating efficient tracking and management.

Status : A column in the order table that displays the current status of each order, indicating its progress or status within the order fulfillment process.

Action : A feature that leads users to an action page where they can manage order statuses and perform operations like changing the status, exporting orders, and downloading invoices.

Export Orders : The process of generating and saving order data, often in a file format like CSV or Excel, for external use or record-keeping purposes.

The screenshot shows the details for Order ID - 20230410702666. It features a timeline with three stages: Order Pending, Order Processing, and Order At Local Facility. Below the timeline is a 'Products' section. To the right is a 'Change Status' dropdown menu with the following options: Order Pending, Order Processing, Order at local facility, Order Out For Delivery, Order Completed, and Order Cancelled. A red box highlights the 'Order Completed' option. Red arrows point from the text definitions to the corresponding UI elements: 'Order Status' points to the status dropdown; 'Payment Status' points to the payment method indicator; 'Download Invoice' points to the download invoice button; 'Change Status' points to the 'Change Status' button; and 'Click here to Change status' points to the 'Order Completed' option in the dropdown menu.

Order Pending, Order Processing, Order At Local Facility, Order Out For Delivery, Order Completed, Order Cancelled : Status options representing various stages of an order's fulfillment process, allowing administrators to track and update the order's progress.

Change Status: A button that, when clicked, saves the selected status from the dropdown input field, updating the status column in the order table.

Download Invoice: A button that generates and allows users to retrieve the invoice for a specific order in PDF format.

Vendor

To manage the status of an order on PickBazare Commerce, log in to your vendor account, go to specific shop and navigate to the "Orders" section.

Click on the specific order in the list. Review the order details, ensure availability of items, click the "Save Changes" button after changing the status of an order. This ensures that your actions are recorded. In this way vendor can effectively manage the status of orders and maintain transparent communication with customers on PickBazare Commerce.

The screenshot displays the PickBazare Commerce vendor dashboard. On the left, there is a sidebar with navigation links: Dashboard, Attributes, Products, Authors, Manufacturers/Publications, Orders (which is highlighted with a red box), Refunds, Staff, Withdrawals, Reviews, Questions, and Store Notice. The main content area has two main sections. The top section, titled 'My Shops', shows six active shop cards: Furniture Shop, Clothing Shop, Bags Shop, Makeup Shop, Bakery Shop, and Grocery Shop. Each card includes a thumbnail, shop name, and status (Active). A red arrow points to the 'Visit Specific Shop' link above the Furniture Shop card. The bottom section, titled 'Furniture Shop', shows a large image of a modern living room interior. Below the image, it says 'The furniture shop is the best shop around the city. This is being run...'. It includes a 'Read more' button, address (588 Firwood Road, New Jersey, East Dover, 08753, USA), phone number (21342121221), and a 'Visit Shop' button. A red arrow points to the 'Orders' link in the sidebar. The right side of the dashboard shows summary statistics: Products (55 Total Products), Revenue (\$0.00 Gross Sales, \$0.00 Current Balance), Others (10% Admin Commission Rate), and Payment Information (Name: furniture shop, Email: furniture@demo.com). It also indicates that the shop was registered since June 27, 2021.

The screenshot shows a table of orders with columns for Tracking Number, Delivery Fee, Total, Order Date, and Status. A search bar at the top right allows users to type queries. Sorting options for Total and Order Date are available. An 'Export Orders' button is present, along with a 'Click here' link for more actions.

Orders					Search here	Export Orders	Click here
Tracking Number	Delivery Fee	Total	Order Date	Status	Shipping Address	Actions	
20230829451702	\$0.00	\$510.00	18 minutes ago	Order Processing Status will update according to action Order Completed	2148 Stratford Park, Winchester, KY, 40391, United States	Click here to visit action	
20230825848907	\$0.00	\$300.00	4 days ago	Order Completed	2148 Stratford Park, Winchester, KY, 40391, United States	Click here to visit action	

This screenshot shows the details for Order ID 20230410702666. It includes a status history timeline, payment information, and a dropdown menu for changing the order status. A 'Click here to Change status' link is highlighted with a red arrow.

Order Status : Order Completed

Payment Status : Cash On Delivery

Order ID - 20230410702666

Order Pending Order Processing Order At Local Facility

Products

Order status: Order Pending, Order Processing, Order at local facility, Order Out For Delivery, Order Completed, Order Cancelled.

Change Status

Click here to Change status

User Management

Super Admin to User Management

Super admin is a role who has the authority to run a complete **CRUD** (Create, Read, Update, Delete) operation on this feature, Like Super admin Can add or block Vendor, Staff & Customer. Also can change permissions to any user.

PickBazar

ORDER MANAGEMENT

- Orders
- Create Order
- Transactions

USER CONTROL

- All users
- Admin list
- Vendors
- Staff

 - My Staffs
 - Vendor Staffs

- Customers

FEEDBACK CONTROL

- Reviews
- Questions

PROMOTIONAL MANAGEMENT

Users

See All users

ID Name Permissions Available wallet points Status Actions

#ID: 6	admin admin@gmail.com	super_admin customer store_owner	Available wallet points	Active		
#ID: 5	customer3 customer3@demo.com	customer	Available wallet points	Active		
#ID: 4	customer2 customer2@demo.com	customer	Available wallet points	Active		
#ID: 2	Customer customer@demo.com	customer	Available wallet points	Active		
#ID: 1	Store Owner store_owner@demo.com	customer store_owner	Available wallet points	Active		

Search by Name

+ Add User

ID Name Permissions Available wallet points Status Actions

#ID: 6	admin admin@gmail.com	super_admin customer store_owner	Available wallet points	Active		
#ID: 5	customer3 customer3@demo.com	customer	Available wallet points	Active		
#ID: 4	customer2 customer2@demo.com	customer	Available wallet points	Active		
#ID: 2	Customer customer@demo.com	customer	Available wallet points	Active		
#ID: 1	Store Owner store_owner@demo.com	customer store_owner	Available wallet points	Active		

Add Wallet Points

super admin can add wallet points

Submit

Vendor Management & Staff Management

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

ORDER MANAGEMENT

- Orders
- Create Order
- Transactions

USER CONTROL

- All users
- Admin list
- Vendors

 - All vendors
 - Pending vendors

- Staff

 - My Staffs
 - Vendor Staffs

- Customers

Users

Search by Name

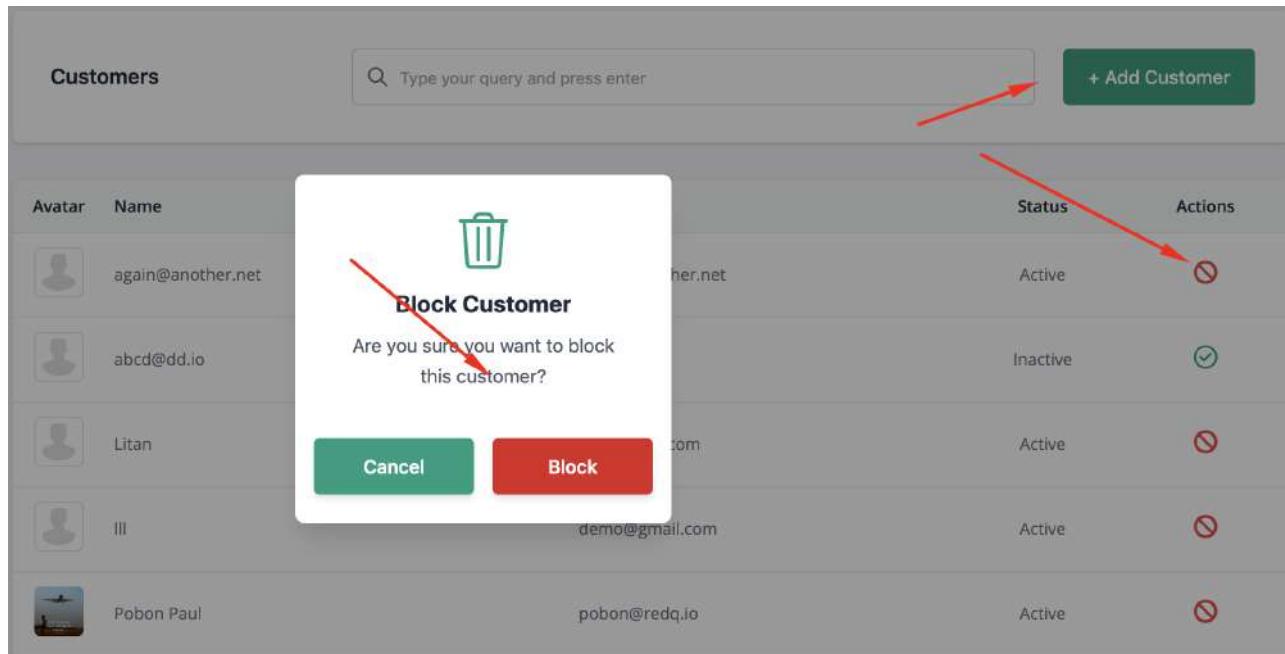
+ Add User

ID Name Permissions Available wallet points Status Actions

#ID: 6	admin admin@gmail.com	super_admin customer store_owner	Available wallet points	Active		
#ID: 5	customer3 customer3@demo.com	customer	Available wallet points	Active		
#ID: 4	customer2 customer2@demo.com	customer	Available wallet points	Active		
#ID: 2	Customer customer@demo.com	customer	Available wallet points	Active		
#ID: 1	Store Owner store_owner@demo.com	customer store_owner	Available wallet points	Active		

Customer Management

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



Introduction to Feedback management

Feedback play a very essential role in any e-commerce business. So, it's very important to manage & control Feedback.

Review

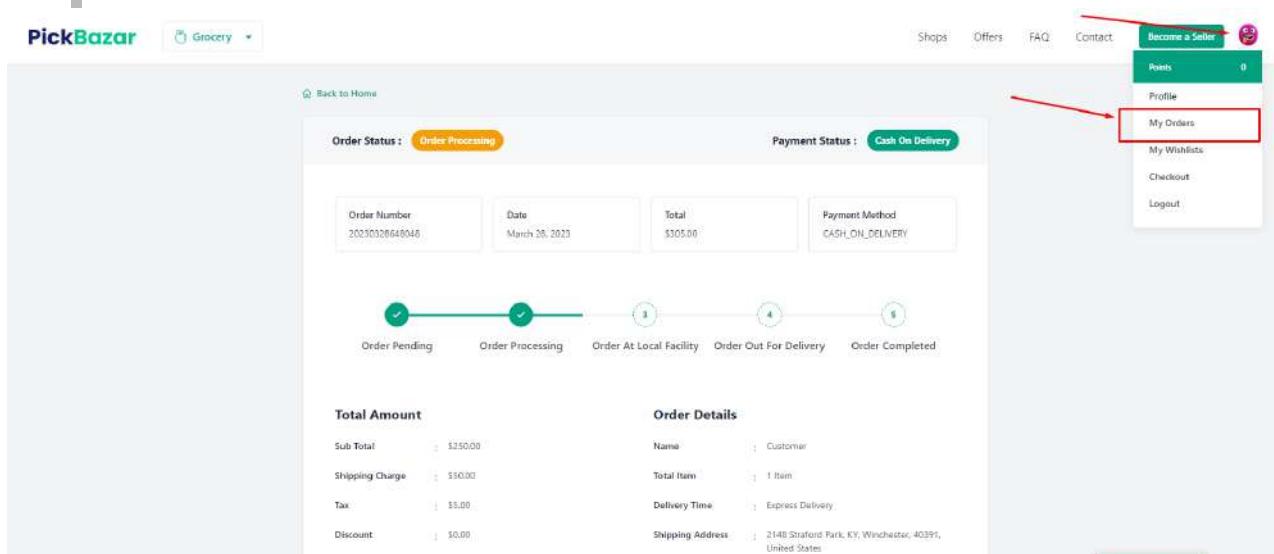
Customers are able to provide comments on things they have purchased using Pickbazar's review function.

Customers may rate a product (often on a scale of 1 to 5) and write a review about it using the review option. Other future buyers who are thinking about buying the same product might find this feedback useful. While it enables them to learn more about how their product is being received by clients and make adjustments as needed, it may also be useful to the vendor or seller.

The Pickbazar review option is a crucial instrument for gaining the respect and trustworthiness of customers. Allowing customers to offer their insights and viewpoints promotes openness and authenticity in the purchasing process.

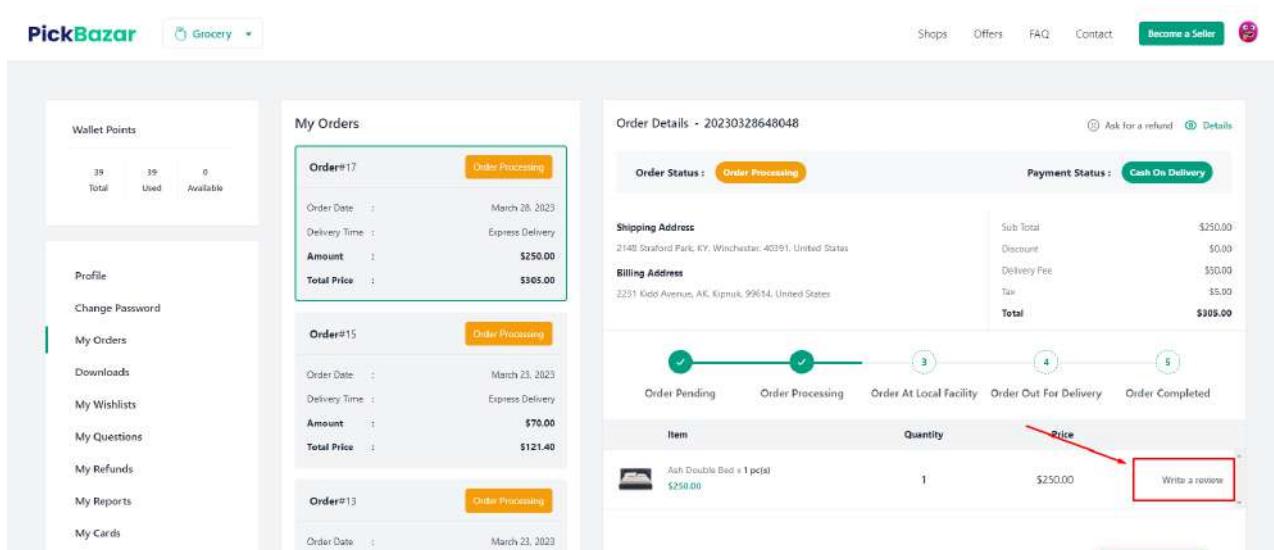
Review in Customer perspective

After successfully placing a purchase, the customer will select the profile icon in the top right corner. He must choose **My Orders** from the drop-down option.



The screenshot shows the PickBazar mobile application interface. At the top, there is a navigation bar with the 'PickBazar' logo, a 'Grocery' category, and links for 'Shops', 'Offers', 'FAQ', 'Contact', 'Become a Seller', 'Pools' (with a value of 0), 'Profile', 'My Wishlists', 'Checkout', and 'Logout'. A red arrow points from the 'Profile' link to the 'My Orders' link in the dropdown menu. Below the navigation bar, the main content area displays the 'Order Status' as 'Order Processing' and the 'Payment Status' as 'Cash On Delivery'. It shows details for Order Number 20230328648046, dated March 28, 2023, with a total of \$305.00 and payment method CASH_ON_DELIVERY. A progress bar indicates the order status: Order Pending (green checkmark), Order Processing (green checkmark), Order At Local Facility (grey circle with number 3), Order Out For Delivery (grey circle with number 4), and Order Completed (grey circle with number 5). Below the progress bar, there are sections for 'Total Amount' and 'Order Details'. The 'Total Amount' section lists Sub Total (\$250.00), Shipping Charge (\$50.00), Tax (\$5.00), and Discount (\$0.00). The 'Order Details' section includes Name (Customer), Total Item (1 item), Delivery Time (Express Delivery), and Shipping Address (2140 Stratford Park, KY, Winchester, 40391, United States).

The consumer will discover a **write a review** option under **My Orders** to the right of each individual order.



The screenshot shows the 'My Orders' section of the PickBazar app. On the left, there is a sidebar with 'Wallet Points' (39 Total, 39 Used, 0 Available) and a 'Profile' section with links for 'Change Password', 'My Orders', 'Downloads', 'My Wishlists', 'My Questions', 'My Refunds', 'My Reports', and 'My Cards'. The main content area shows three orders: Order #17 (Status: Order Processing, Date: March 28, 2023, Amount: \$250.00, Total Price: \$305.00), Order #15 (Status: Order Processing, Date: March 23, 2023, Amount: \$70.00, Total Price: \$121.40), and Order #13 (Status: Order Processing, Date: March 23, 2023). To the right of the order details, there is a 'Order Details - 20230328648048' section with 'Ask for a refund' and 'Details' buttons. It shows the same order information as the first screenshot. Below this, there is a progress bar and a table for the order items. The table has columns for 'Item', 'Quantity', and 'Price'. The 'Price' column for the first item (Ash Double Bed x 1 pc(s)) is \$250.00. A red box highlights the 'Write a review' button next to the price.

After selecting **write a review**, a modal window with a few input fields will show up for you to fill out and publish.

The screenshot shows a user interface for leaving a review. At the top left is a small image of a bed. To its right is the product name, "Ash Double Bed". Below the product name is a section titled "Give Ratings" which contains five empty star icons. A red arrow points from the text "Give a rating" to the rightmost star icon. Below the ratings is a text input field labeled "Description" with a red arrow pointing to it. Further down is a section titled "Upload Images" containing a dashed box with a cloud icon and the text "Upload an image or drag and drop PNG, JPG". A red arrow points from the bottom of this section towards a large green "Submit" button.

Ash Double Bed

Give Ratings

Give a rating

Description

Write a description

Upload Images

Upload an image or drag and drop
PNG, JPG

Submit

ratings : A review rating is a point total or tally that sums up how well a user or customer feels about a certain item, service, or experience. Customer reviews are frequently available on websites, and these evaluations are sometimes rated using a scale, such as a five-star rating system, where the more stars, the better the product or service is thought to be.

description : A written review that offers more information about the user's experience and might be useful to other shoppers when making judgments is frequently included with review ratings.

One can amend the review after submitting it by doing the following.

The screenshot shows the PickBazar dashboard. On the left, there's a sidebar with 'Wallet Points' (39 Total, 39 Used, 0 Available), 'Profile', 'Change Password', and a list of 'My Orders', 'Downloads', 'My Wishlists', 'My Questions', 'My Refunds', 'My Reports', and 'My Cards'. The main area has a section titled 'My Orders' with three order cards: Order#17 (Processing), Order#15 (Processing), and Order#13 (Processing). To the right is 'Order Details - 20230328648048' showing shipping and billing addresses, payment status (Cash On Delivery), and a timeline from Order Pending to Order Completed. A red arrow points to the 'Update review' button next to the product item.

Each product detail page's review section will contain any recently published reviews.

The screenshot shows the product detail page for 'Ash Double Bed'. It displays a star rating summary (1 rating, 1★) and two reviews. The first review is by 'Customer2' (1★) dated March 18, 2023, with the text 'had to wait long time for delivery,'. The second review is by 'Customer2' (1★) dated March 28, 2023, with the text 'this is a demo review description..'. Red arrows point to the star ratings and the review descriptions.

Review in Super Admin perspective

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

The screenshot shows the super admin 'Reviews' list. The left sidebar includes 'Create Order', 'Transactions', 'USER CONTROL' (All users, Admin list, Vendors, Staff, Customers), 'FEEDBACK CONTROL' (Reviews, Questions), and 'PROMOTIONAL MANAGEMENT' (Coupons, Flash sale). The main area lists reviews for products like 'Milan The Story of Love', 'Space Force First Chapter', 'Space Force Second Chapter', 'Greddy Love', and 'Partex Cushioned Double Bed'. Each review includes the customer name, product ID, rating (e.g., 3★, 5★), date (e.g., 2 years ago), and an edit icon. Red arrows point to the 'Reviews' menu item in the sidebar and to the edit icons in the review list.

Question

A comprehensive and scalable e-commerce site may be created using Pickbazar, an open-source Laravel-based e-commerce platform. Before making a purchase, shoppers may ask questions about a certain product using Pickbazar's "Question" function.

Customers' worries and questions regarding the product are clarified thanks to this feature, which may eventually result in higher customer satisfaction and fewer product returns. On the backend admin panel, the store owner or administrator may view and respond to the queries.

Customers may enable the "Question" function, specify the minimum number of questions for each product, and score how helpful other customers' or the business owner's responses were. Customers' confidence and credibility in the store and its items are increased thanks to this feature.

Question in Super Admin perspective

On the admin dashboard, the super admin will choose `settings` and then select `maximum question limit` to enter a figure. A consumer is only permitted to ask the `maximum question limit` in relation to a certain product. An error warning will show up if he asks more questions than the `maximum question limit`."



The administrator set a `maximum question limit` of three, as seen in the image below. In order to prevent further inquiries from the consumer, an error notice will be displayed.

A screenshot of the PickBazar frontend. At the top, there is a navigation bar with the 'PickBazar' logo and a 'Grocery' category. Below the navigation, a user profile shows 'by Customer2' and a message 'had to wait long time for delivery.' The main content area displays a list of 'Questions and Answers (3)'. One question is visible: 'Q: What other color you can offer? A: Any color of your choice.' A red arrow points from the text 'An error notice will be displayed.' in the accompanying text to a red error box in the top right corner of the page. The error box contains the text 'Maximum Question Limit Exceeded'. At the bottom right of the page, there is a green button labeled 'Activate Windows'.

On the admin dashboard, choose **question** to provide the customer's inquiry with an answer. Next, in order to respond to the newly posed question, he will click on **reply**.

The screenshot shows the PickBazar admin dashboard. On the left sidebar, there is a list of menu items: Groups, Categories, Tags, Manufacturers/Publications, Authors, Orders, Create Order, Users, Coupons, Taxes, Shipments, Withdrawals, Message, Refunds, Questions, Store Notice, Reviews, and Settings. The 'Questions' item is highlighted with a red box and has a red arrow pointing to it from the bottom-left.

The main content area displays various statistics: Total Revenue (\$0.00), Total Order (8), Today's Revenue (\$0.00), and Total Shops (7). Below these are sections for Sale History (a line chart) and Recent Orders (a table). The Recent Orders table shows one entry: Tracking Number 2023032348226, Total \$121.40, Order Date 4 days ago, and Status Order Processing.

The screenshot shows the PickBazar admin dashboard under the 'Questions' section. The left sidebar remains the same as the previous screenshot. The main content area shows a table of questions and answers. One specific question is highlighted with a red box and a red arrow pointing to it from the bottom-left. This question is: "Q: this is question 1 for furniture shop". The answer is: "A: replied which is updated". A modal box is overlaid on this row, containing the question and answer text along with a 'Reply' button, also highlighted with a red box.

A modal box with a text field for responding will display. The question and answer will both show underneath the store product in the question and answer area when the response has been submitted.



Pacific Natural Shaving Cream
Product ID: 407

Q: It showed a lady fitted out with a fur hat and fur boa who sat upright, raising a heavy fur muff that covered the whole of her lower arm towards the viewer. Gregor then turned to look out the window at the dull weather. Drops of rain could be heard hitting the pane, which made him feel quite sad.

Type your answer here

If admin give answer of this question this question and answer will appear in the shop for a specific product details

Reply

Also, the administrator has the ability to edit, remove, and rate the question using like and dislike.



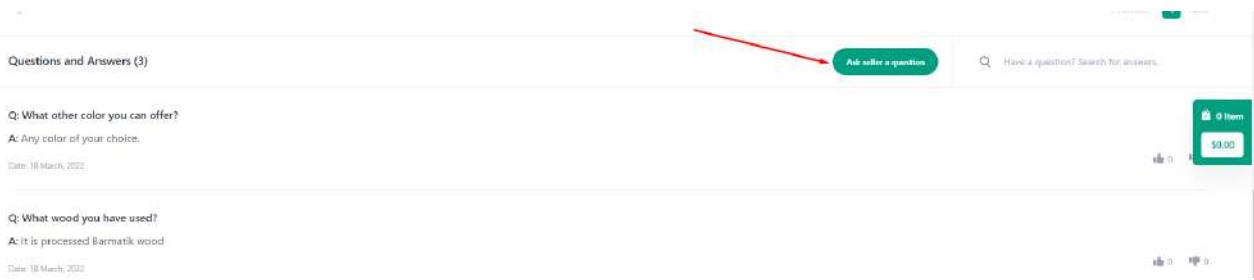
Q: Is it sweet or sour?
A: Its sweet as it contains sugar.

Customer Sun Tropics Organic Mango Nectar,250ml 0 0 a year ago Delete Edit

Question in Customer perspective

The consumer will visit the specifics of a given product and scroll down to see the FAQ section. He will see the phrase **Ask seller a question** there.

If he selects **Ask seller a Question**, a modal window will open, allowing him to type and submit the query.



Questions and Answers (3)

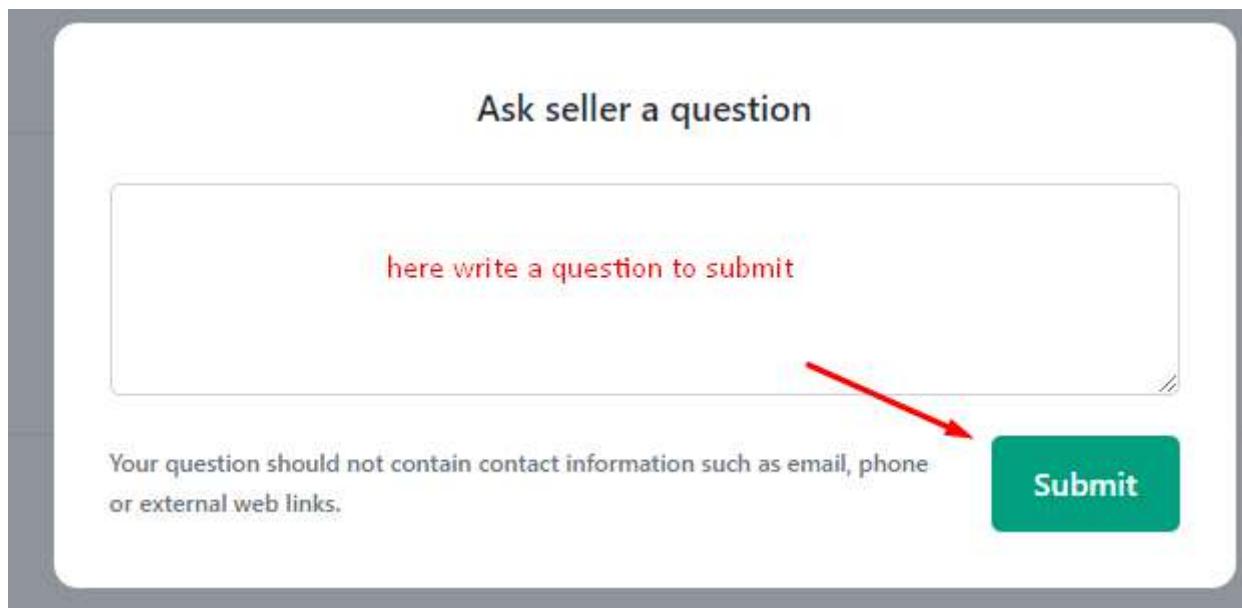
Q: What other color you can offer?
A: Any color of your choice.
Date: 18 March, 2022

Q: What wood you have used?
A: It is processed Barrattik wood
Date: 18 March, 2022

Ask seller a question

Have a question? Search for answers.

0 Item \$0.00



Introduction to Promotional management

Shop smarter and save big with our essential e-commerce promotions: coupons & flash sales. Grab exclusive discounts with coupons, making your favorite items more affordable. Stay tuned for our exciting flash sales, offering unbeatable deals on a variety of products. Don't miss out – start saving now!

Coupon Management

Pickbazar is an online store-building tool that enables companies to launch online stores and sell goods. A coupon is a code that a consumer may enter on Pickbazar during the checkout process to get a discount or other promotional offer on their purchase.

On Pickbazar, companies may generate coupons in a variety of ways for their consumers, including percentage-based discounts, fixed amount discounts, free delivery, and buy one, get one free offers. Moreover, they may specify the coupon's validity period, the minimum order amount needed to utilize it, and the maximum number of times it can be used.

Consumers can use the coupon code to get a discount or take advantage of a promotion during the checkout process. Businesses may use coupons as effective marketing tools to draw in new clients, keep old ones coming back, and boost sales at their Pickbazar shop.

Coupon in Super Admin perspective

- The coupon option is accessible through the admin dashboard. Once you enter the coupons, the "Add Coupons" option will be available for adding additional coupons.

Coupons

ID	Banner	Code	Amount	Minimum Amount	Active	Expired	Actions
10		4OFF	\$4.00	\$0.00	2 years ago	In a year	
9		5OFF	\$5.00	\$0.00	2 years ago	In a year	
8		OFF%	\$6.00	\$0.00	2 years ago	In a year	
7		BAZAR10	\$10.00	\$0.00	2 years ago	In 2 years	
6		BAZAR20	\$20.00	\$0.00	2 years ago	In a year	
5		BAZAR18	\$18.00	\$0.00	2 years ago	In a year	
4		HELLO15	\$15.00	\$0.00	2 years ago	In a year	
3		BIDIZ	\$12.00	\$0.00	2 years ago	In a year	
2		FOD	\$8.00	\$0.00	2 years ago	In a year	

- The administrator will click "add coupon" after filling out all the forms to create a new coupon. Which will appear in the coupon list. Here are descriptions of each field:

Description
Add your coupon description and necessary information from here

Code
4OFF

Description

Type
 Fixed
 Percentage
 Free Shipping

Coupon discount amount (USD)
0

Minimum cart amount (USD)
0

Active from
15/03/2023

Will Expire
15/03/2023

Activate Window Add Coupon

code : While making a purchase online or in-person, customers can enter a coupon code, which is a combination of letters and/or numbers, to get a discount or exclusive deal. Often, firms or shops use coupon codes as a marketing strategy to get consumers to purchase their goods or services.

description : The offer that a discount code makes to clients is briefly described in the description of the coupon code. The sort of discount or offer, the goods or services it applies to, any limits or limitations, and the expiration date are often all listed. Customers may determine if they wish to use a discount code to make a purchase by reading the

description of the code, which explains the terms and limitations of the deal.

minimum cart amount : A consumer must have a certain minimum total number of goods or services in their shopping cart in order to use a coupon and benefit from the accompanying discount or deal.

coupon discount amount : The amount of the discount that a consumer receives when using a coupon code is known as the coupon discount amount.

type : Two distinct sorts of coupons—the fixed coupon type and the percentage coupon type—offer clients various kinds of savings.

Regardless of the overall order quantity, a fixed coupon type provides a particular monetary discount on a purchase.

On the other hand, a percentage coupon type provides a discount based on a percentage of the entire order value.

active date : The date on which a coupon becomes active and may be used by customers to obtain the related discount or offer is referred to as the active date.

expire date : The date on which a coupon becomes invalid and can no longer be used by consumers to enjoy the accompanying discount or offer is referred to as the coupon's expiry date.

Customers should be aware of a coupon's active and expire date since attempting to use one before or after that date might render it invalid and prevent them from enjoying the discount or deal.

Coupon in Customer perspective

- A customer may see an offer printed in **do you have a coupon?** when he adds things to his basket and proceeds to the checkout.

- [Dashboard](#)
- [Shops](#)
- [My Shops](#)
- [Products](#)
- [Attributes](#)
- [Groups](#)
- [Categories](#)
- [Tags](#)
- [Manufacturers/Publications](#)
- [Authors](#)
- [Orders](#)
- [Create Order](#)
- [Users](#)
- [Coupons](#)
- [Taxes](#)
- [Shipments](#)
- [Withdrawals](#)

Your Order

8 x Celery Stick 1lb	\$40.00
3 x Clementines 1lb	\$7.50
Sub Total	\$47.50
Tax	\$0.95
Shipping	\$50.00
Do you have a coupon?	
Total	\$98.45
Wallet points	0
Wallet currency	\$0.00
<input type="checkbox"/> Do you want to use wallet?	

Choose Payment Method

Cash Cash On Delivery

Please pay with cash

Place Order

- The consumer now clicks on the offer text and enters a discount code to receive a discount.

- [Dashboard](#)
- [Shops](#)
- [My Shops](#)
- [Products](#)
- [Attributes](#)
- [Groups](#)
- [Categories](#)
- [Tags](#)
- [Manufacturers/Publications](#)
- [Authors](#)
- [Orders](#)
- [Create Order](#)
- [Users](#)
- [Coupons](#)
- [Taxes](#)
- [Shipments](#)
- [Withdrawals](#)

Your Order

8 x Celery Stick 1lb	\$40.00
3 x Clementines 1lb	\$7.50
Sub Total	\$47.50
Tax	\$0.95
Shipping	\$50.00
40OFF	Apply
Total \$98.45	
Wallet points	0
Wallet currency	\$0.00
<input type="checkbox"/> Do you want to use wallet?	

Choose Payment Method

Cash Cash On Delivery

Please pay with cash

Place Order

- The discount will now show up and be deducted from the subtotal. and after placing an order, the consumer can take advantage of a discount.

- [Dashboard](#)
- [Shops](#)
- [My Shops](#)
- [Products](#)
- [Attributes](#)
- [Groups](#)
- [Categories](#)
- [Tags](#)
- [Manufacturers/Publications](#)
- [Authors](#)
- [Orders](#)
- [Create Order](#)
- [Users](#)
- [Coupons](#)
- [Taxes](#)
- [Shipments](#)
- [Withdrawals](#)

Your Order

8 x Celery Stick 1lb	\$40.00
3 x Clementines 1lb	\$7.50
Sub Total	\$47.50
Tax	\$0.95
Shipping	\$50.00
Discount: 40OFF	-\$4.00
Total	\$94.45
Wallet points	0
Wallet currency	\$0.00
<input type="checkbox"/> Do you want to use wallet?	

Choose Payment Method

Cash Cash On Delivery

Please pay with cash

Place Order

Flash Sale

A flash sale is a limited-time promotion offered by retailers, both online and offline, where products are sold at significantly discounted prices for a short duration, typically just a few hours or a day. Flash sales are often used to create a sense of urgency and excitement among shoppers, encouraging them to make quick purchasing decisions. These sales can feature a wide range of products, from electronics to fashion items, and are known for their deeply discounted prices, but they are available only for a brief period or until the limited stock runs out.

For more details [Flash Sale](#)

Introduction to Engagement

In an e-commerce website, messages and store notices play a crucial role in keeping customers informed. Messages offer personalized communication, enhancing customer relationships. Meanwhile, store notices provide essential updates, promotions, or policy changes, ensuring transparency and a seamless shopping experience. Together, they enhance user engagement and satisfaction.

Message

With the use of messaging, you can make sure that your shop owner have easy access to the special offers, marketing promotions, and stock updates you're providing them. With a 98% open rate, text message links are more likely to be clicked by recipients than email ones.

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

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

The screenshot shows the PickBazar admin dashboard. On the left, there's a sidebar with categories like User Control, Feedback Control, Promotional Management, Feature Management, and Site Management. Under Feature Management, the 'Message' option is highlighted with a red arrow pointing to it. In the main content area, there's a search bar at the top, followed by a section titled 'Search by shop (type at least 3 character)'. Below this is a message inbox with a placeholder 'Your inbox is Empty'. To the right, there's an illustration of a person using a laptop with speech bubbles above it, and a button labeled 'Select Your Conversation'. At the bottom of the main area, there's a 'Compose Message' button.

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

This screenshot is similar to the one above, but the 'Compose Message' button at the bottom of the main content area is now highlighted with a red box. The rest of the interface, including the sidebar and the message inbox, remains the same.

Starting Chat

The screenshot shows a chat window. At the top, there's a profile icon and the text 'Furniture Shop'. Below this is a message input field containing the text 'Starting Conversation'. A red arrow points from the text in the input field towards a green button labeled 'Starting Conversation' which is highlighted with a red border.

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

- Dashboard
- Shops
- My Shops
- Products
- Attributes
- Groups
- Categories
- Tags
- Manufacturers/Publications
- Authors
- Orders
- Create Order

Search by shop (type at least 3 character)



Furniture Shop

a few seconds...

Furniture Shop

...



No Messages Found!

Compose

Type your message here..



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

- Dashboard
- Shops
- My Shops
- Products
- Attributes
- Groups
- Categories
- Tags
- Manufacturers/Publications
- Authors
- Orders
- Create Order

Search by shop (type at least 3 character)



Furniture Shop

a few seconds...

Furniture Shop

...

No Messages Found!

Type your message here..



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



Furniture Shop

this is furnitur shop

3 days ago

hello furniture

3 days ago

- Now they will get details of the shop.



Furniture Shop

The furniture shop is the best shop around the city. This is being run...

[Read more](#)

 588 Finwood Road,
East Dover,
New Jersey,
08753,
USA

 21342121221

[Visit Shop](#)



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

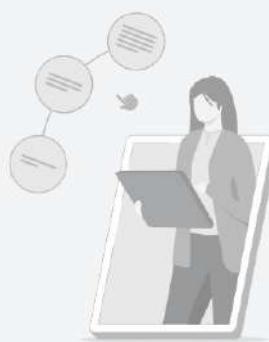
[Dashboard](#)[Shops](#)[My Shops](#)[Products](#)[Attributes](#)[Groups](#)[Categories](#)[Tags](#)[Manufacturers/Publications](#)[Authors](#)[Orders](#)

Admin

admin@demo.com

Please add your [Profile](#)[Enabled](#)Search by shop (type at least 3 character) 

Your Inbox is Empty

[Compose](#)

Select Your Conversation

Basic Info

Add some basic info about your shop from here

Name

Name is required

Description

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

My Shops

New Test Shop

Inactive

Shops						
<input type="text"/> Type your query and press enter						
Logo	Name	Owner Name	Products	Orders	Status	Actions
	New Test Shop	Admin	0	0	Inactive	click here to active



Admin
 admin@demo.com
 Please add your [Profile](#)
 Enabled

My Shops

New Test Shop

Active

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

PickBazar

Manufacturers/Publications

Authors

Orders

Create Order

Users

Coupons

Taxes

Shipments

Search by shop (type at least 3 character)

Starting Chat

New Test Shop

Starting Conversation

PICKBAZAR_ERROR_YOU_CAN_NO

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



Store Owner

store_owner@demo.com
12365141641631

Enabled

Basic Info

Add some basic info about your shop from here

Name

Name is required

Description



PickBazar

Create Shop



Store Owner

store_owner@demo.com
12365141641631

Enabled

My Shops

Message

Store Notice



Furniture Shop

Active



Clothing Shop

Active



Bags Shop

Active



Makeup Shop

Active



Bakery Shop

Active



Grocery Shop

Active



Books Shop

Active



New store owner shop

Active

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

Starting Chat

New store owner shop

X ▾



Starting Conversation

My Shops Message Store Notice

Search by shop (type at least 3 character)

hello from admin 4 minutes a...
New store owner shop

Enabled

Search by shop (type at least 3 character)

New store owner shop

hello from admin a few seconds...
New store owner shop

hello from admin a few seconds ago

Type your message here..

Compose

Store Notice

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

- **Store Notice Hierarchy**
 - Admin ⇒ Vendor/All Vendor/Selected Vendors
 - Vendor ⇒ Shop
- **Priority**

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

Super Admin

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

The screenshot shows the PickBazar dashboard with a sidebar containing various management sections: User Control, Feedback Control, Promotional Management, Feature Management, and Site Management. Under Feature Management, the 'Store Notice' link is highlighted with a red arrow. The main content area displays a 'Store Notices' table with columns for ID, Notice, Description, Type, Effective From, Expired At, Issued By, Priority, and Actions. A search bar at the top right allows filtering by notice. Below the table, a message says 'No data found'.

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

This screenshot shows the 'Store Notices' page. It features a search bar with the placeholder 'Type your query and press enter'. To the right of the search bar is a green button labeled '+ Add Store Notice'.

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

- **Notice Title:** The title of a store notification attracts attention; it should be short and to the point in order to convey the essence of the notice and prevent the merchant from getting lost in a sea of details.
- **Notice description:** Notice descriptions are a type of formal letter that is sent to a certain vendor or set of suppliers. Similar to a news story, it alerts the appropriate party or parties to a significant incident.
- **Notice Priority:** There are three ways to set a priority. Low, medium, and high. Vendors may assign importance based on the urgency of a message.

- **Active Date**: The term "active date" refers to a notice's start date or effective period.
- **Expire Date**: A notice's expiration date indicates that it was over on that particular day.
- **Notice Type**: An individual or business that sells products or services to another is referred to as a vendor or supplier. This vendor may be a general vendor, a particular vendor, or an administrator.

Notice creation for specific vendor

Description
Add Store Notice necessary information from here.

Priority*
Medium

Store Notice*
Notice to Vendors

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

Active
20/01/2023

Expire
25/01/2023

Type*
Specific Vendor

Received By*
Select...

Add Store Notice

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

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

Store Notices

Type your query and press enter

+ Add Store Notice

ID	Notice	Description	Type	Effective From	Effective At	Issued By	Priority	Actions
22	FSG	Prepare for FSG's approval (which approval ...	All vendor	a day ago	in 11 days	Super admin	LOW	
21	A letter for each of the ven...	A letter for each of the vendors of the Servic...	All vendor	a day ago	in 4 days	Super admin	MEDIUM	
20	Notice to Vendors	NMS shall notify each Vendor servicing GECS...	All vendor	a day ago	in 14 days	Super admin	HIGH	

6. Update a notice by following the below picture.

Store Notices								<input type="text" value="Type your query and press enter"/>	<button>+ Add Store Notice</button>
ID	Notice	Description	Type	Effective From	Effective At	Issued By	Priority	Actions	
14	new notice	description	All vendor	in 9 hours	in 2 days	Super admin	HIGH		
									

7. Update notice

Description

Edit Store Notice necessary information from here

Priority*	<input type="text" value="High"/>
Store Notice*	<input type="text" value="new notice updated"/>
Description	<input type="text"/>
Active	<input type="text" value="19/01/2023"/>
Expire	<input type="text" value="26/01/2023"/>
Type*	<input type="text" value="All Vendor"/>
	<button>Back</button>
	<button>Update Store Notice</button>

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

- Priority

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

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

Vendor

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

Store Notice Vendor profile

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

The screenshot shows a user interface for managing store notices. At the top, there are tabs: 'My Shops', 'Message', and 'Store Notice'. The 'Store Notice' tab is active, indicated by a green underline. Below the tabs, there are three notices listed:

- Notice to Vendors** (with a blue 'Read more' button)
- A letter for each of the vendors** (with an orange 'Read more' button)
- FSG** (with a yellow 'Read more' button)

Each notice has a small circular icon with an 'i' and a checkmark or a question mark icon to its right. Red arrows point from the text above to the 'Read more' buttons of the first two notices.

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

Staff

Create a Notice for perticular shop

Creation a notice for [furniture-shop]

Create Store Notice

Description	Priority*
Add Store Notice necessary information from here.	Medium
Store Notice*	Offer
Description	Buy one Get one Offer from 20/01/2023 to 25/01/2023.
Active	20/01/2023
Expire	25/01/2023
<input type="button" value="Add Store Notice"/>	

Issued notice for a specific shop [furniture-shop]

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

Store Notice staff profile

Staff for [furniture-shop]

The screenshot shows a staff profile interface. On the left, there's a placeholder for a profile picture with the text "Staff" and an email address "store_staff@domain.com". Below this, a button says "Please add your [Profile](#)". To the right, there's a "Enabled" status indicator with a green checkmark. The main area displays a "Store Notice" card. The card has a yellow info icon, the word "Offer", and a "MEDIUM" priority level. It contains the text "Buy one Get one Offer from 20/01/2023 to 25/01/2023..." and a "Read more" link. At the top of this card, there are tabs for "My Shops", "Message", and "Store Notice", with "Store Notice" being the active tab.

Settings Management

Settings system is divided into multiple parts for better usage & understanding. Lets checkout all the sub chapter about settings.

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.

The screenshot shows the PickBazar admin dashboard. On the left, there is a sidebar with various menu items: Create Order, Transactions, User Control (All users, Admin List), Vendors, Staff, Customers, Feedback Control (Reviews, Questions), Promotional Management (Coupons, Flash Sale), Feature Management (Message, Store Notice), and Site Management (Settings). A red arrow points from the 'Settings' menu item to the main content area. The main content area has three sections: Summary, Order Status, and Recent Orders. The Summary section displays metrics like Total Revenue (\$2,488.92), Total Order (10), Vendor (2), and Total Shops (9). The Order Status section shows Pending Order (0), Processing Order (0), Completed Order (0), and Cancelled Order (0). The Recent Orders section lists four recent orders with details: Tracking Number, Customer, Products, Order Date, Total, Status, and Actions. A search bar labeled 'Search by Name' is also present in the Recent Orders section.

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:

The screenshot shows the PickBazar admin interface. On the left, there's a sidebar with various management sections like Dashboard, Shops, Products, etc. The main area is titled 'Settings' and has tabs for General settings, Payment settings, SEO settings, Events settings, Shop settings, and Company Information. Under 'General settings', there's a 'Logo' section with instructions to upload a logo (138x34 Pixel, 100 MB). Below it is a 'Collapse Logo' section with similar upload instructions. At the bottom right of the main area is a green 'Save Settings' button.

For Collapsible Logo View:

This screenshot shows the same PickBazar admin interface as above, but with a collapsed sidebar. The sidebar icons are visible as small squares in a vertical list. The main area is identical to the first screenshot, with the 'Logo' and 'Collapse Logo' sections and the 'Save Settings' button.

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

Both uploading systems follow this procedure,

API

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

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

If you're using windows and your API is running with ports like `localhost:8000` or `127.0.0.1:8000`, then make sure you add the domain with port to `APP_URL`. Like this `API_URL=http://localhost:8000`

After that, clear your API cache by using this command from the `api` folder,

```
php artisan optimize:clear
```

Then link the storage system by using this command,

```
php artisan storage:link
```

If you're using AWS S3, then update `MEDIA_DISK=s3` and add all `AWS_` credentials For AWS s3, make sure your properly setup permission of the bucket with ACL enable by follow this link -> <https://stackoverflow.com/a/70603995/2158023>

Change Site information

This is the Site `Information` area of the admin settings. Here you can change your Site Title, Site Subtitle, Currency, Minimum Order Amount, Wallet Currency Ratio, Sign Up Points, Maximum Question Limit, OTP checkout, Tax Class, Shipping Class and handle Free Shipping.

The screenshot shows the 'Information' section of a settings page. It includes fields for 'Site Title' (Pickbazar), 'Site Subtitle' (Your next ecommerce), 'Currency' (US Dollar), 'Minimum Order Amount' (0), and 'Wallet Currency Ratio' (3). Red arrows point from the left margin to each of these five fields.

Information	
Change your site information from here	
Site Title	Pickbazar
Site Subtitle	Your next ecommerce
Currency	US Dollar
Minimum Order Amount	0
Wallet Currency Ratio	3

Shipping class, Tax class & Free Shipping

You may find the shipping class at the conclusion of the information section. You may choose the precise class you wish to utilize from this menu. These classes were developed within the shipping category. The classes include a value for shipping that should be added to the total price bought.

For tax classes, the same applies. You may create tax classes in the admin settings' tax area by entering a dollar amount or percentage that will be added to the total amount of your order.

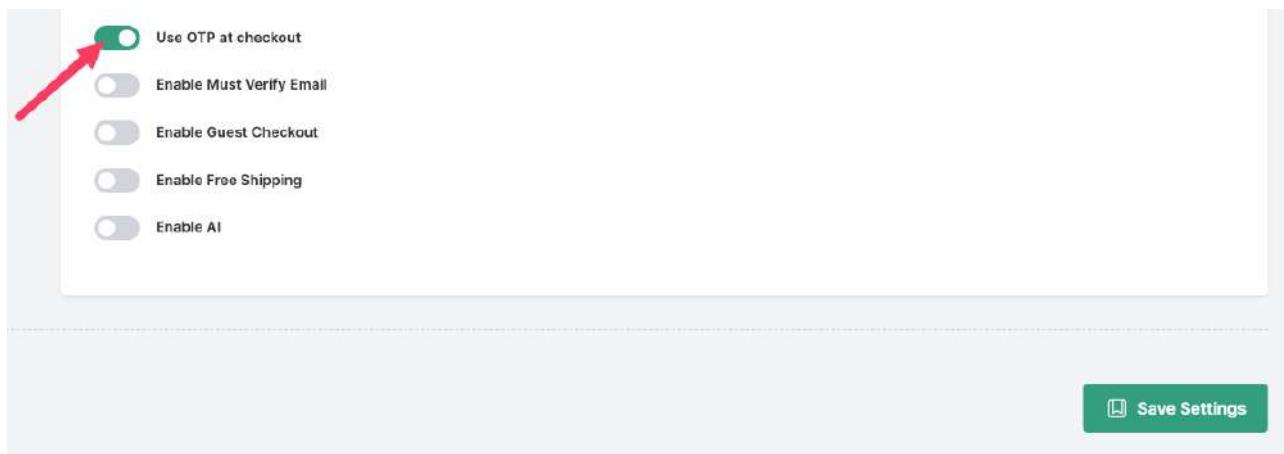
The option to allow free delivery is another one. It may be turned off to stop offering free delivery. Customers will receive free delivery if it is activated. If not, the cost of shipment must be paid.

The screenshot shows the 'Admin Settings' page. It includes dropdown menus for 'Tax Class' (Global) and 'Shipping Class' (Global), both with red arrows pointing to them. Below these are several toggle switches: 'Use OTP at checkout' (off), 'Enable Must Verify Email' (off), 'Enable Guest Checkout' (off), and 'Enable Free Shipping' (on, indicated by a green arrow). A red box highlights the 'minimum cart amount' input field, which is currently empty. At the bottom right is a 'Save Settings' button.

Tax Class	Global
Shipping Class	Global
<input type="checkbox"/> Use OTP at checkout	
<input type="checkbox"/> Enable Must Verify Email	
<input type="checkbox"/> Enable Guest Checkout	
<input checked="" type="checkbox"/> Enable Free Shipping	
Minimum cart amount for free shipping	minimum cart amount
<input type="checkbox"/> Enable AI	

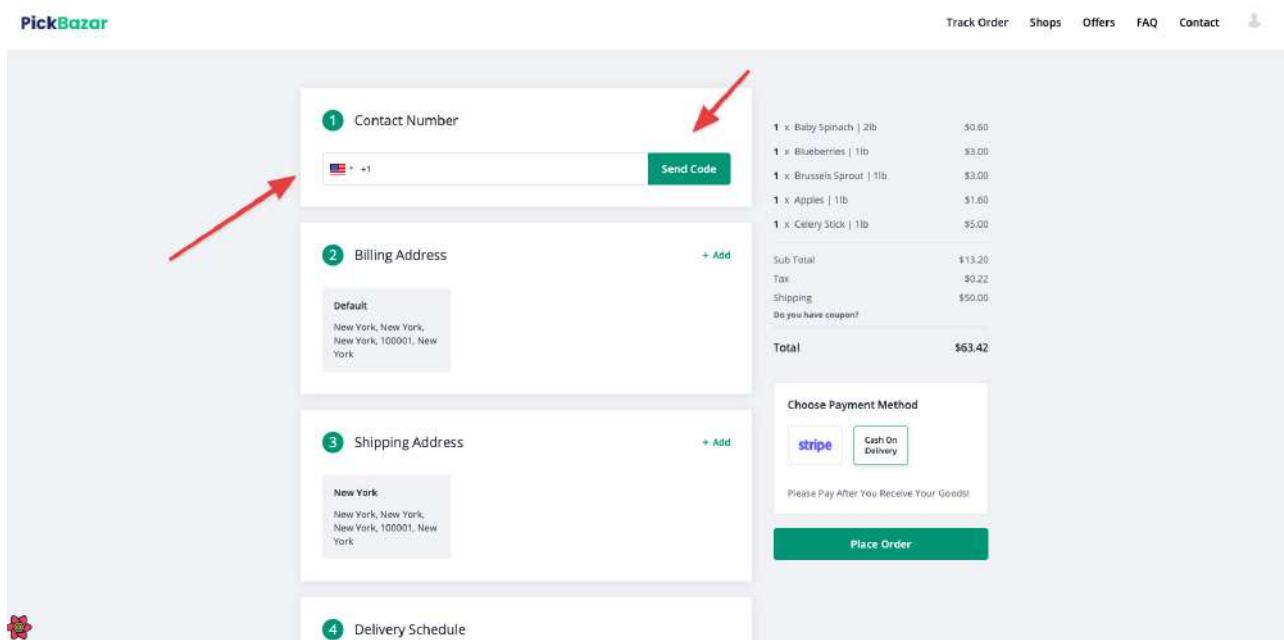
Save Settings

To Enable or Disable Use OTP at checkout

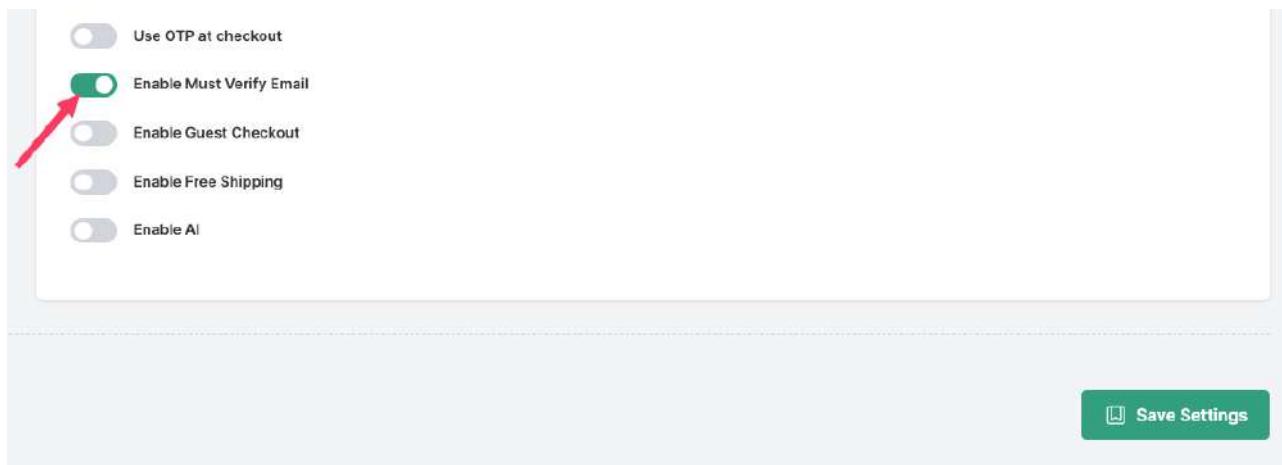


If Enable OTP at checkout, When users go to the checkout page, they'll get a form to verify their mobile number.

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..

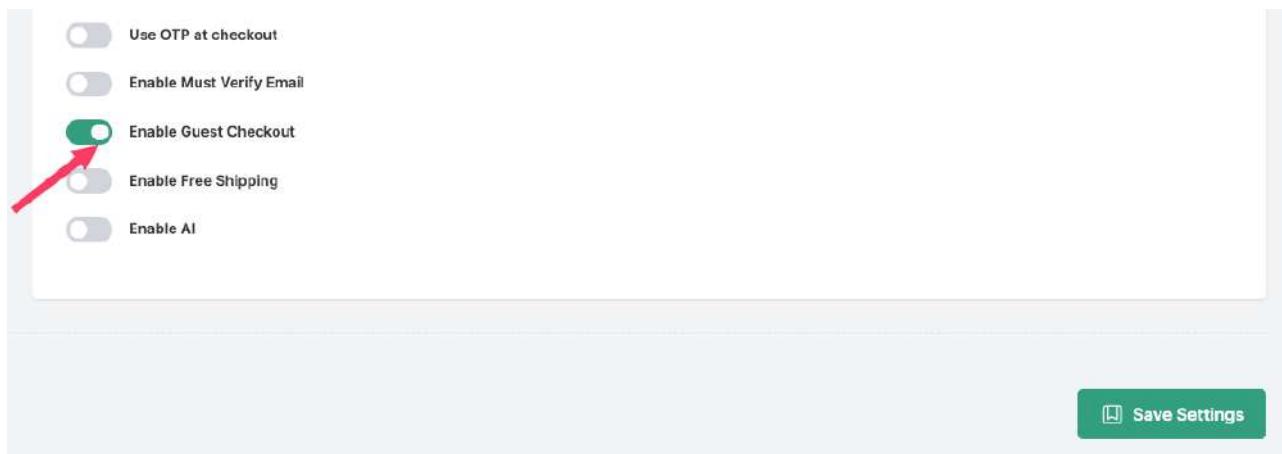


To Enable or Disable Verify Email

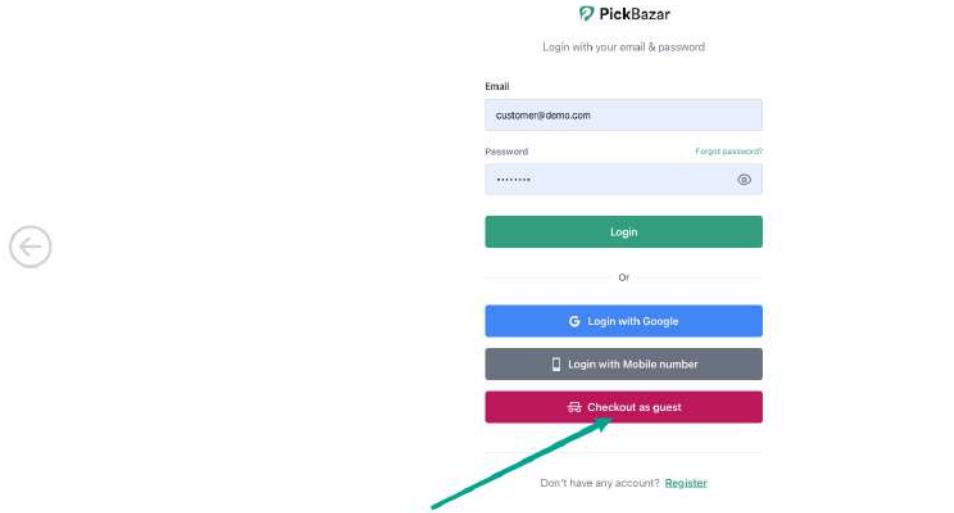


For more information check our [Email-Verification](#) Section.

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

A screenshot of the AI settings page. It shows several toggle switches: "Use OTP at checkout" (off), "Enable Must Verify Email" (off), "Enable Guest Checkout" (off), "Enable Free Shipping" (off), and "Enable AI" (on, indicated by a red arrow). Below the switches is a dropdown menu with "openai" selected. At the bottom right is a green "Save Settings" button.

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 Settings

General settings **Payment settings** SEO settings Events settings Shop settings Company Information

Payment
Configure Payment Option

Enable Cash On Delivery

Currency: US Dollar

Enable Gateway

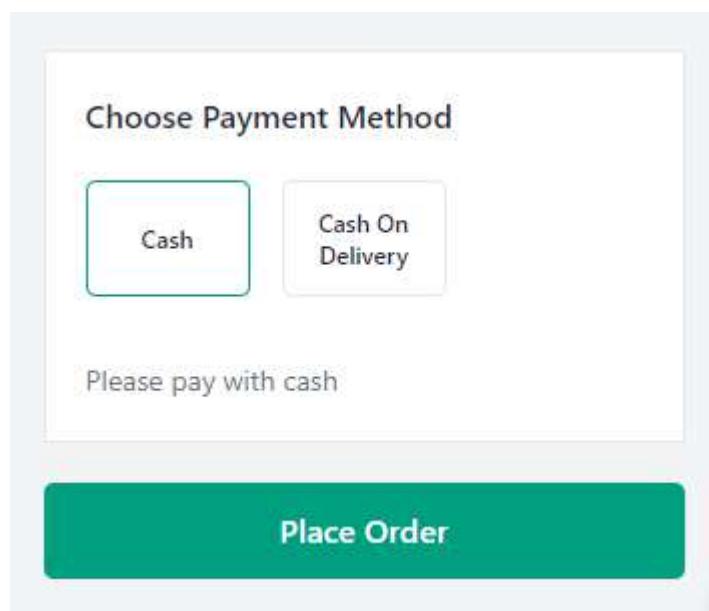
Select Payment Gateway:

- stripe
- PayPal
- Razorpay
- mollie
- paymongo
- pystack
- ssocommerce
- iyzico
- Flutterwave

Set default payment gateway: Stripe

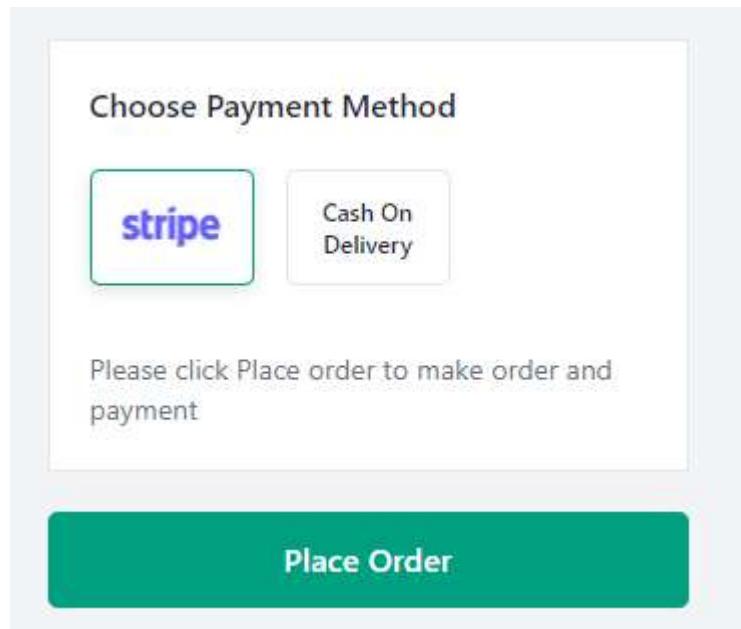
Super Admin

The admin will see **cash** and **cash on delivery** when he will go to **Choose Payment Method** after generating an order from the shop. Because the buyer may pay while purchasing the product directly from the administrator. For customers who will pay cash after getting the merchandise distant from the admin, there is also the option of cash on delivery.



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.



Event Settings

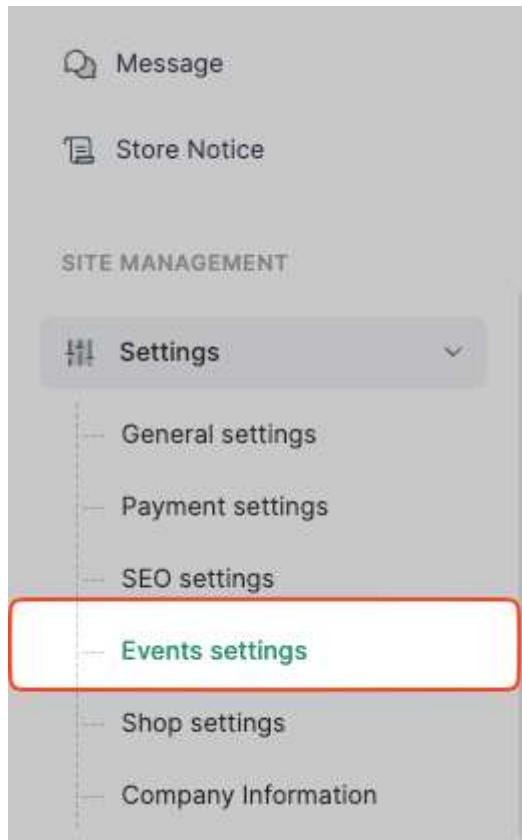
These days, every software incorporates some form of notification or alert system. In order to enhance the customer-friendliness of **Pickbazar Laravel**, we have implemented such alert systems. However, it is often the case that such systems come with a high cost. Recognizing the importance of cost 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



The screenshot shows the 'Events Settings' page. On the left is a sidebar with navigation items: Vendors, Staff, Customers, FEEDBACK CONTROL (Reviews, Questions), PROMOTIONAL MANAGEMENT (Coupons, Flash Sale), FEATURE MANAGEMENT (Message, Store Notice), SITE MANAGEMENT (Settings, General settings, Payment settings, SEO settings, Events settings, Shop settings, Company information). The 'Events settings' tab is selected and highlighted with a red box. The main content area has three tabs: 'Realtime notification setting', 'SMS Event Setting', and 'Email Event Setting'. The 'Realtime notification setting' tab is active, showing a sub-section titled 'Realtime notification setting' with the instruction 'Set this to get notification in realtime' and a 'Select notification options' dropdown. The other two tabs show similar structures for SMS and Email event settings. A green 'Save Settings' button is located at the bottom right.

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.

The screenshot shows a configuration panel for 'Realtime notification setting'. On the left, there's a section titled 'Realtime notification setting' with the sub-instruction 'Set this to get notification in realtime'. On the right, a larger section titled 'Select notification options' contains a dropdown menu labeled 'Select...' and a list of notification types under the heading 'AVAILABLE FOR ADMIN & VENDOR': 'Admin & Vendor : Store notice notification', 'Admin & Vendor : Order notification', and 'Admin & Vendor : Message notification'. A small note at the bottom right says 'PICKBAZAR'.

SMS Event Settings

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

The screenshot shows a configuration panel for 'SMS Event Setting'. On the left, there's a section titled 'SMS Event Setting' with the sub-instruction 'Set This to Send SMS on Selected Event'. On the right, a larger section titled 'Select SMS Options' contains a dropdown menu labeled 'Select...' and a list of notification types categorized by role: 'ADMIN' includes 'Admin: Status Change Order', 'Admin: Refund Order', and 'Admin: Payment'; 'STORE OWNER' includes 'Owner: Status Change Order' and 'Owner: Refund Order'. A vertical scroll bar is visible on the right side of the list.

Email Event Settings

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

The screenshot shows a configuration panel for 'Email Event Setting'. On the left, there's a section titled 'Email Event Setting' with the sub-instruction 'Set This to Send SMS on Selected Event'. On the right, a larger section titled 'Select Email Options' contains a dropdown menu labeled 'Select...' and a list of notification types categorized by role: 'ADMIN' includes 'Admin: Status Change Order', 'Admin: Refund Order', and 'Admin: Payment'; 'STORE OWNER' includes 'Owner: Status Change Order'. A vertical scroll bar is visible on the right side of the list. At the bottom left, there's a copyright notice: '©2023 Pickbazar. Copyright © REDQ. All rights reserved worldwide. REDQ'.

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.

The screenshot shows the SEO Settings page in the PickBazar admin interface. On the left, there's a sidebar with various management sections like Mail, Dashboard, Shop Management (Shops, My Shops), Product Management (Products, Inventory, Categories, Tags, Attributes, Manufacturers/Publications, Authors), E-commerce Management (Taxes, Shipments, Withdrawals, Refunds), and Outbound Control. The main content area is titled "SEO Settings" and has tabs for General settings, Payment settings, SEO settings (which is selected and highlighted with a red box), Events settings, Shop settings, and Company Information. Under the SEO tab, there's a section titled "SEO" with the sub-instruction "Change your site SEO from here." It contains six input fields: "Meta Title", "Meta Description", "Meta Tags", "Canonical URL", "OG Title", and "OG Description". Each field has a red arrow pointing to it from the left side of the image. At the bottom right of the form is a green "Save Settings" button.

Shop settings

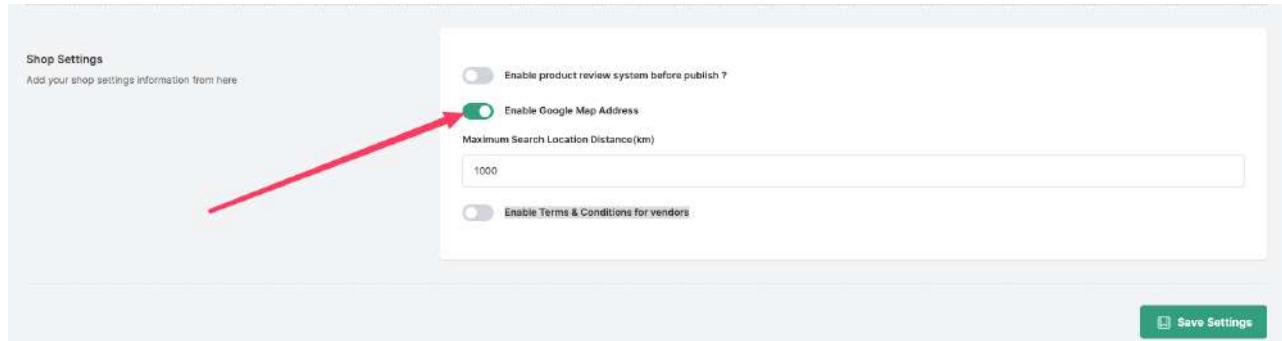
Add or Remove Delivery Schedule

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.

The screenshot shows the Delivery Schedule settings page. At the top left, there's a red box around the title "Delivery Schedule" and the sub-instruction "Add your delivery schedule time with greater description from here." Below this, there are three entries for delivery times, each with a red arrow pointing to its "Title/Time" field. The first entry is "Morning" with the time "8:00 AM - 11:30 AM". The second entry is "Noon" with the time "11:00 AM - 2:00 PM". The third entry is "Evening" with the time "2:00 PM - 5:00 PM". Each entry has a red "Remove" button to its right. At the bottom of the list is a green "Add Delivery time" button with a red arrow pointing to it from the bottom left of the image.

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.



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

A screenshot of the 'Company Information' page under the 'Settings' tab. The top navigation bar includes 'General settings', 'Payment settings', 'SEO settings', 'Events settings', 'Shop settings', and 'Company Information' (which is highlighted). The main content area starts with an 'Address' section where users can add their address. A note says 'addresses are automatically fetched from the Google API'. Below this are fields for 'Contact Number' (containing '+12929012212') and 'Website' (containing 'https://rediq.io'). There are also fields for 'Email' (containing 'demo@demo.com') and 'Social Media'. Under 'Select social platform', there are dropdown menus for 'Facebook' and 'Twitter', each with an 'Add profile url' field (containing 'https://www.facebook.com' and 'https://twitter.com/home' respectively) and a 'Remove' button. At the bottom right is a green 'Save Settings' button.

address without google map

The screenshot shows the 'Company Information' tab selected in the navigation bar. The 'Address' section contains fields for City, State, Street Address, Contact Number, Website, and Email, all with 'manual entry' placeholder text. Below these are sections for 'Select social platform' and 'Add profile url'. A green 'Save Settings' button is at the bottom right.

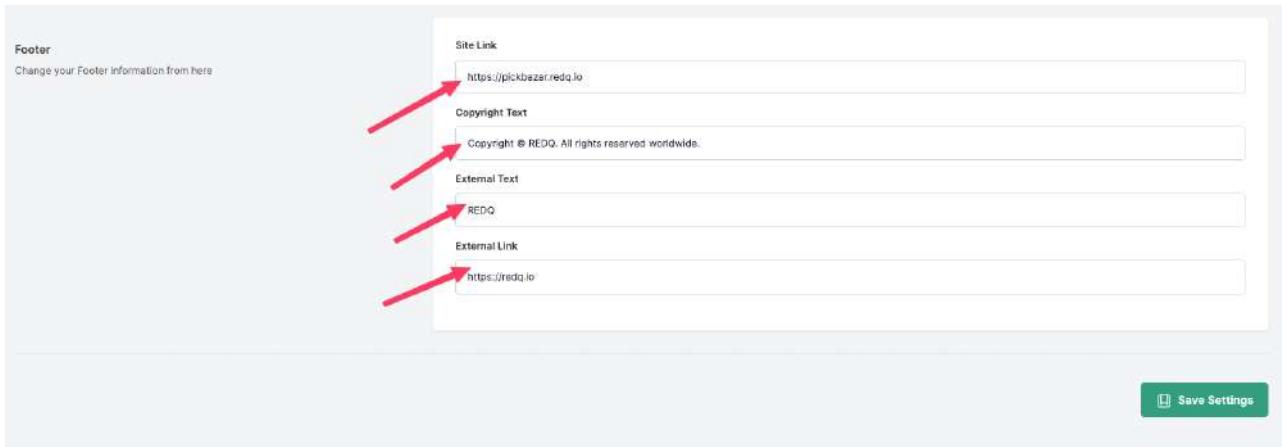
Social platform

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

The screenshot shows three social platform accounts listed: Facebook, Twitter, and Instagram. Each account has a 'Remove' link next to its profile URL. A red arrow points to the 'Add New Social Profile' button at the bottom left. A second red arrow points to the 'Remove' link for the Twitter account. A green 'Save Settings' button is at the bottom right.

Footer

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



Payment

We have introduced the new **Payment** architecture feature on PickBazar **v6.0.0**. So if you want to use **Payment** with **PickBazar**, then make sure your PickBazar is **v6.0.0** or later.

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

- **Stripe**
- **Stripe Element**
- **PayPal**
- **RazorPay**
- **Mollie**
- **Xendit**
- **Paystack**
- **SSLCOMMERZ**
- **Iyzico**
- **bKash**
- **Paymongo**
- **Flutterwave**

Let's discuss those sequentially.

Stripe

Stripe is a financial service & Software as a service (SaaS) company. It offers payment processing software & API for e-commerce applications. In PickBazar we have integrated Stripe API for payment system. Though currently card based features are available only, Stripe Element for other payment options will be integrated in future updates.

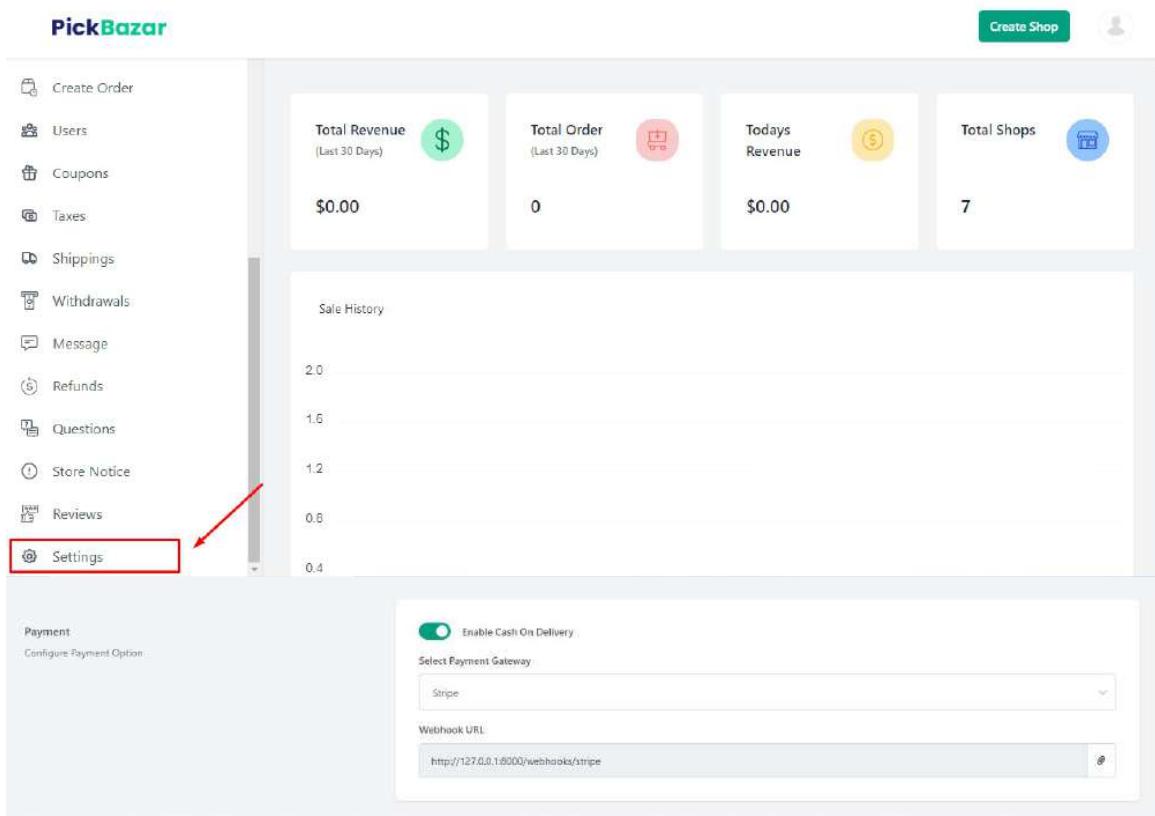
Stripe integrate inside PickBazar.

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

- Inside api there is .env file. You have to copy & paste this line of codes inside the .env file. We will discuss later about how to create those API keys in stripe.

```
STRIPE_API_KEY=[YOUR_STRIPE_API_SECRET_KEY]  
STRIPE_WEBHOOK_SECRET_KEY=[YOUR_STRIPE_API_WEBHOOK_SECRET_KEY]
```

- To activate Stripe go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option.(e.g. webhook URL is coming from local development. This static link will dynamically generated in live environment)

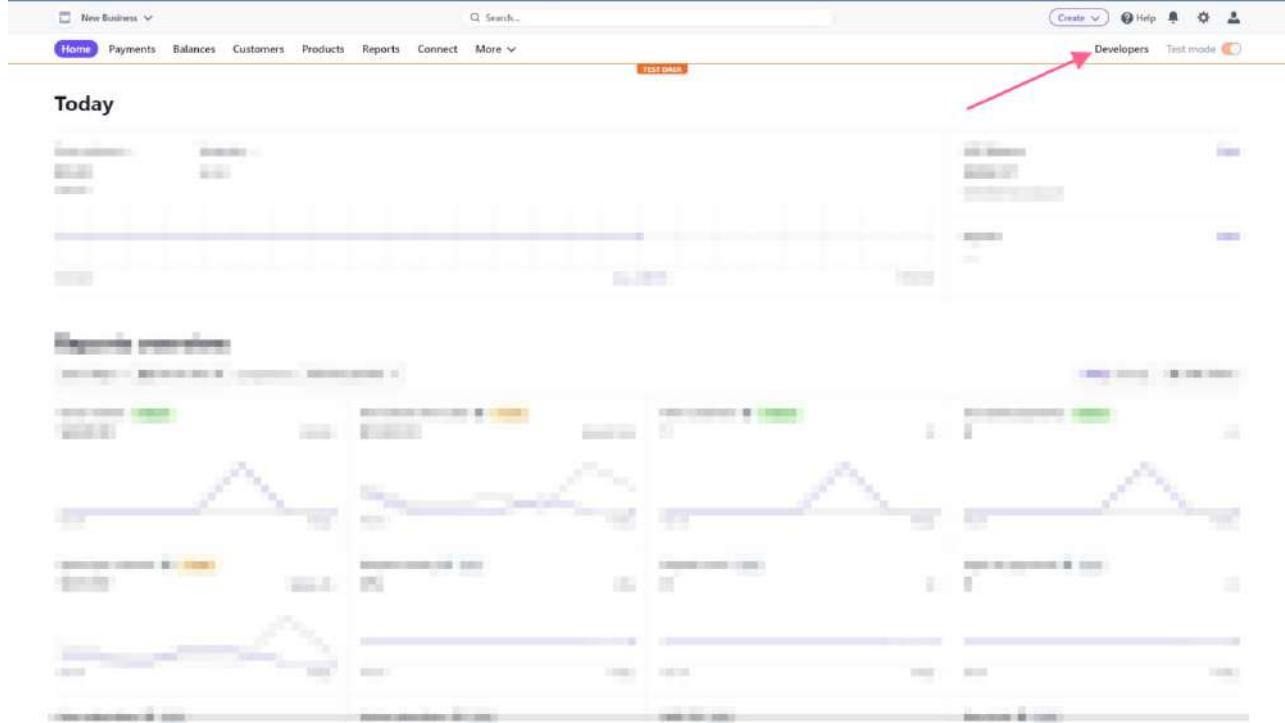


- Add Stripe publishable key inside .env file of pickbazar shop.

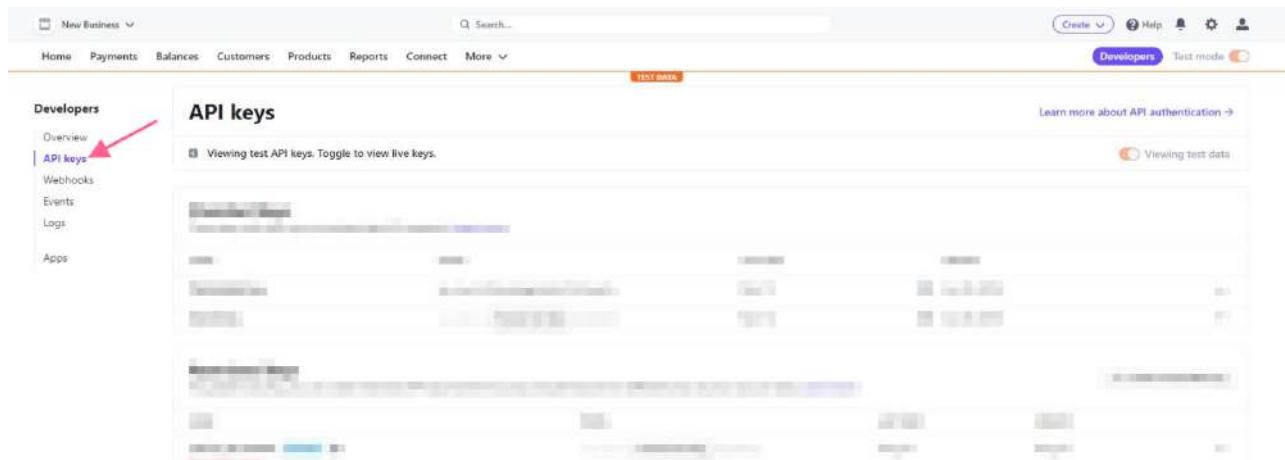
```
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=[YOUR_STRIPE_PUBLISHABLE_KEY]
```

How to create & setup Stripe information properly?

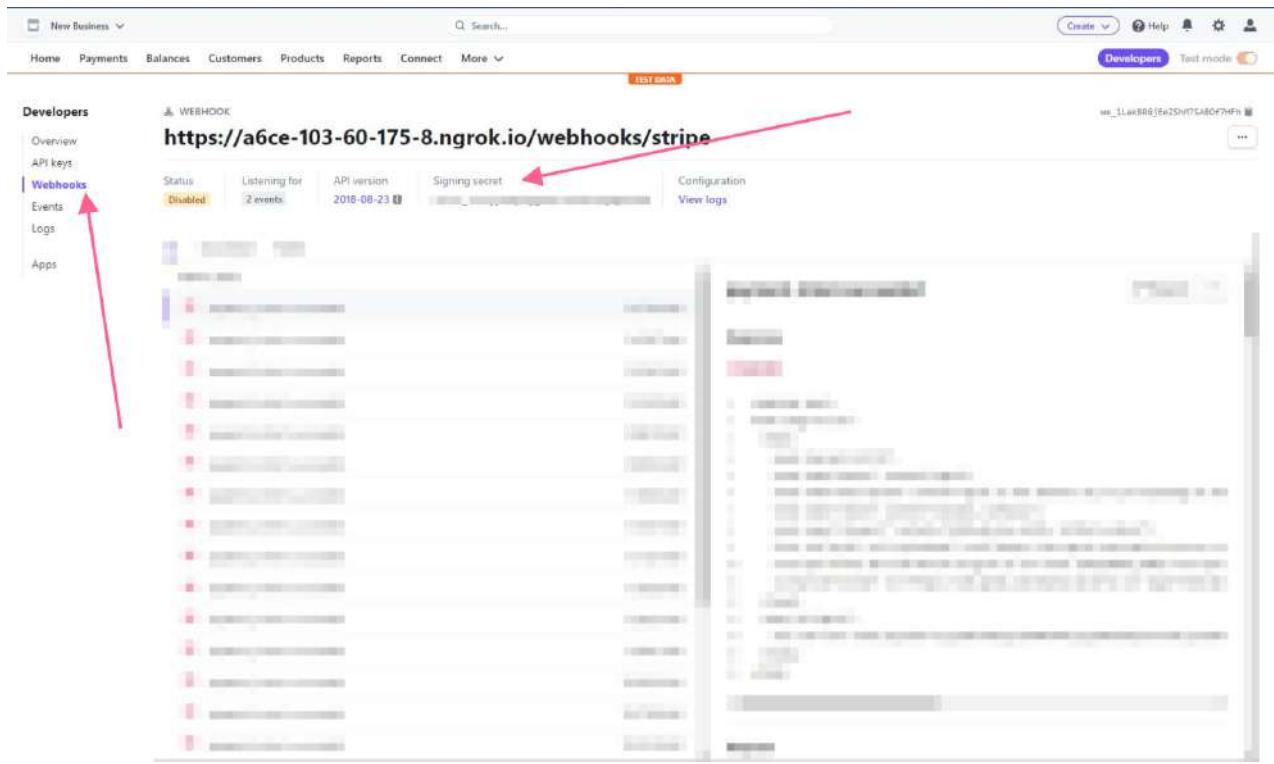
- Go to the stripe website and login <https://stripe.com> If you aren't a registered user, complete the stripe registration first.
- After logged in into stripe dashboard, follow the developer link to create the API keys.



- 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 these two webhook events for monitoring the payment flow.
 - **payment_intent.succeeded**
 - **payment_intent.payment_failed**

Special Notes for Stripe users.

If we have used any third party system/plugin/packages, then we have always encouraged our respected customers to follow the official documentation for detailed & in-depth knowledge.

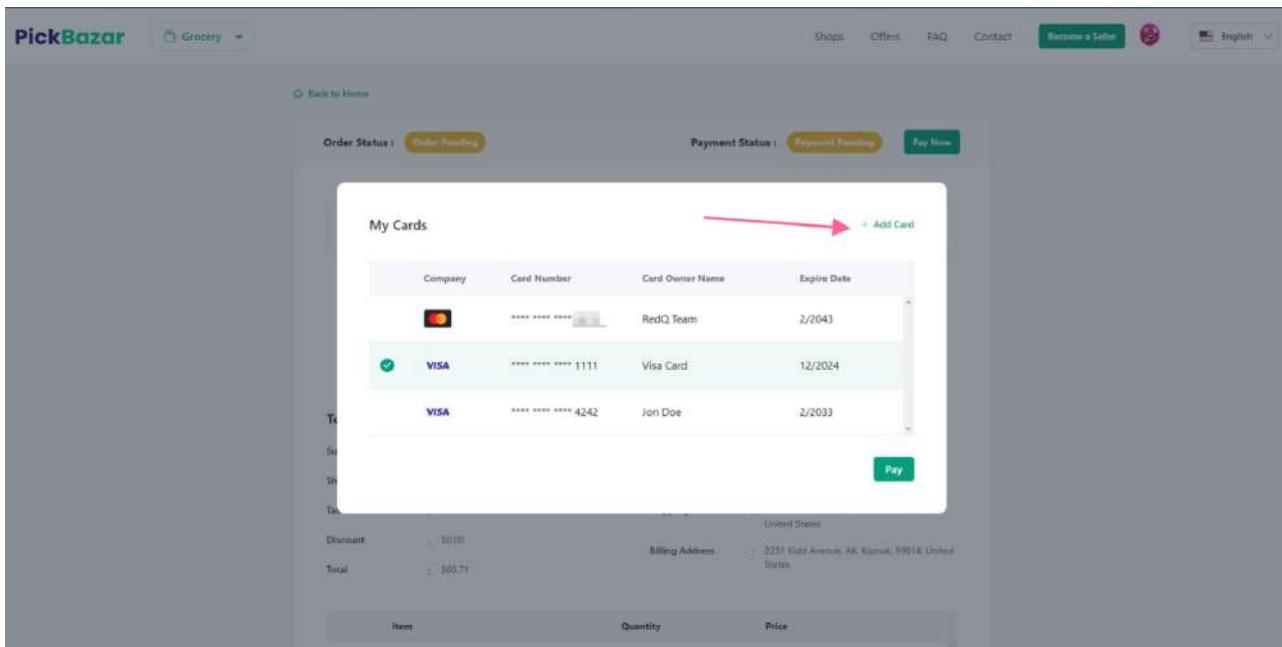
- Stripe DOCS [home page] -- <https://stripe.com/docs>
- Stripe API Docs -- <https://stripe.com/docs/api>
- Stripe JavaScript Docs -- <https://stripe.com/docs/js>

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

In PickBazar we have provided a 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 processes.

- Save card during checkout process.



The screenshot shows the PickBazar checkout page. At the top, there are tabs for 'Grocery' and other categories. Below the tabs, the 'Order Status' is 'Order Pending' and the 'Payment Status' is 'Payment Pending'. A large central modal window titled 'My Cards' displays three saved payment methods:

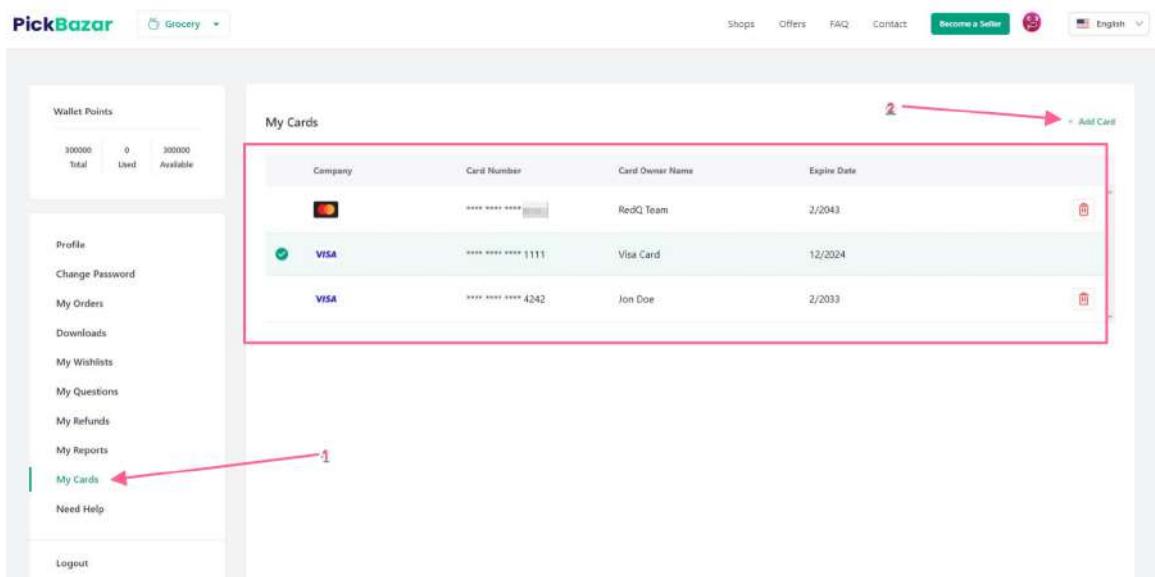
Company	Card Number	Card Owner Name	Expire Date
	**** **** 0123	RedQ Team	2/2043
	**** **** 1111	Visa Card	12/2024
	**** **** 4242	Jon Doe	2/2033

Below the modal, there's a summary of the order details:

- Discount: \$0.00
- Total: \$60.71
- Billing Address: 2231 Kidd Avenue, AK, Ekonik, 99614 United States

A green 'Pay' button is located at the bottom right of the modal. A red arrow points from the text above to the 'Add Card' button in the top right corner of the modal.

- Save card in his user profile **My Card** section.



The screenshot shows the PickBazar user profile page. On the left, a sidebar menu includes options like 'Profile', 'Change Password', 'My Orders', 'Downloads', 'My Wishlists', 'My Questions', 'My Refunds', 'My Reports', 'My Cards' (which is highlighted with a green bar), and 'Need Help'. At the bottom of the sidebar is a 'Logout' link. On the right, the main content area is titled 'My Cards' and displays the same three saved cards as the checkout page. A red box highlights the 'My Cards' section in the sidebar, and a red arrow labeled '1' points to it. Another red arrow labeled '2' points to the 'Add Card' button in the top right corner of the 'My Cards' list.

There are something needs to keep in mind.

Please note

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

Stripe Element

Stripe Elements is a set of prebuilt UI components which is provided by stripe itself for building your web payment flow. It's available as a feature of Stripe.js, Stripe Elements foundational JavaScript library for building functional checkout flows.

Stripe Element integrate inside PickBazar.

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

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in stripe.

```
STRIPE_API_KEY=[YOUR_STRIPE_API_SECRET_KEY]  
STRIPE_WEBHOOK_SECRET_KEY=[YOUR_STRIPE_API_WEBHOOK_SECRET_KEY]
```

- To activate Stripe go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option.(e.g. webhook URL is coming from local development. This static link will dynamically generated in live environment)

PickBazar

[Create Order](#)

[Users](#)

[Coupons](#)

[Taxes](#)

[Shipments](#)

[Withdrawals](#)

[Message](#)

[Refunds](#)

[Questions](#)

[Store Notice](#)

[Reviews](#)

[Settings](#)

Enable Gateway

Select Payment Gateway

stripe	PayPal	Razorpay	mollie
paymongo	paystack	xendit	sslcommerz
iyzico	bKash	Flutterwave	

Set default payment gateway

stripe

Enable Stripe Element

Webhook URL

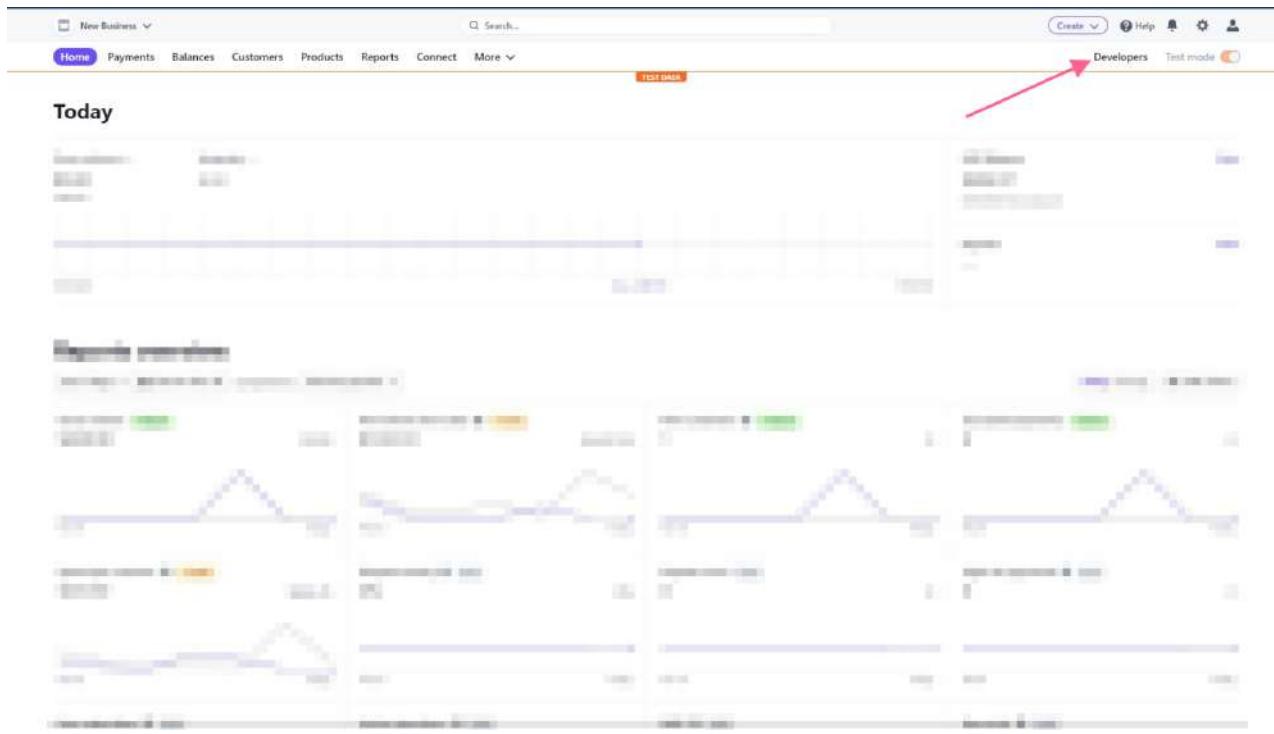
stripe http://localhost:8000/webhooks/stripe

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

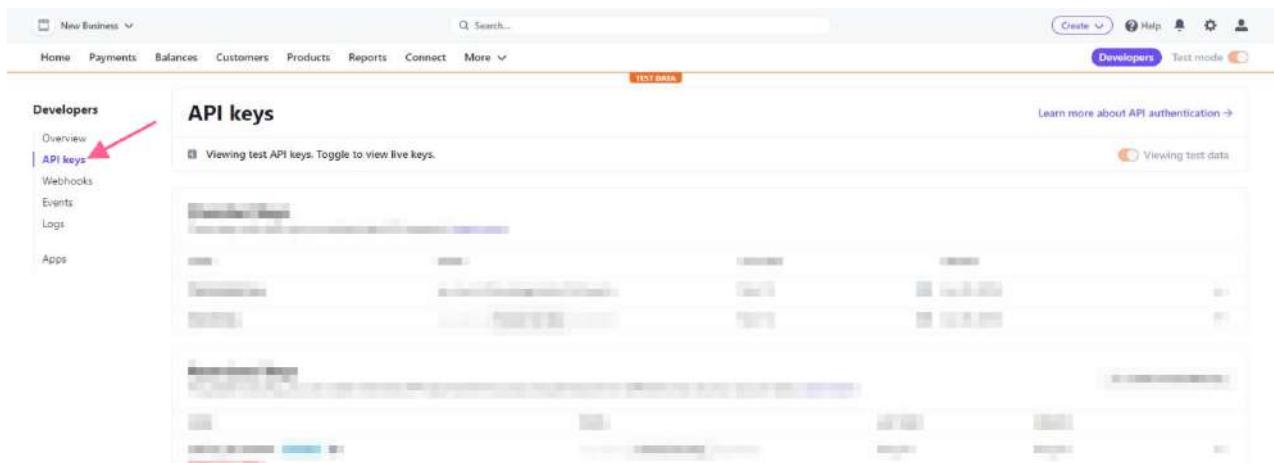
```
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=[YOUR_STRIPE_PUBLISHABLE_KEY]
```

How to create & setup Stripe Element information properly?

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



- Then create API keys from there. for more details follow up this official documentation <https://stripe.com/docs/keys>



If you want to integrate stripe element in pickbazar then you must have to configure webhook properly.



- Drop your webhook endpoint and select all event then "Add endpoint"

- **Note:** We recommended you to enable all event because stripe element's payment method vary on currency. But you can enable those event that are specifically depend on your currency or you can enable all event as recommend.

Listen to Stripe events

[Add an endpoint](#)

[Test in a local environment](#)

Set up your webhook endpoint to receive live events from Stripe or [learn more about Webhooks](#).

Endpoint URL

https://

Description

An optional description of what this webhook endpoint is used for...

Listen to

Events on your account Events on Connected accounts [?](#)

Version

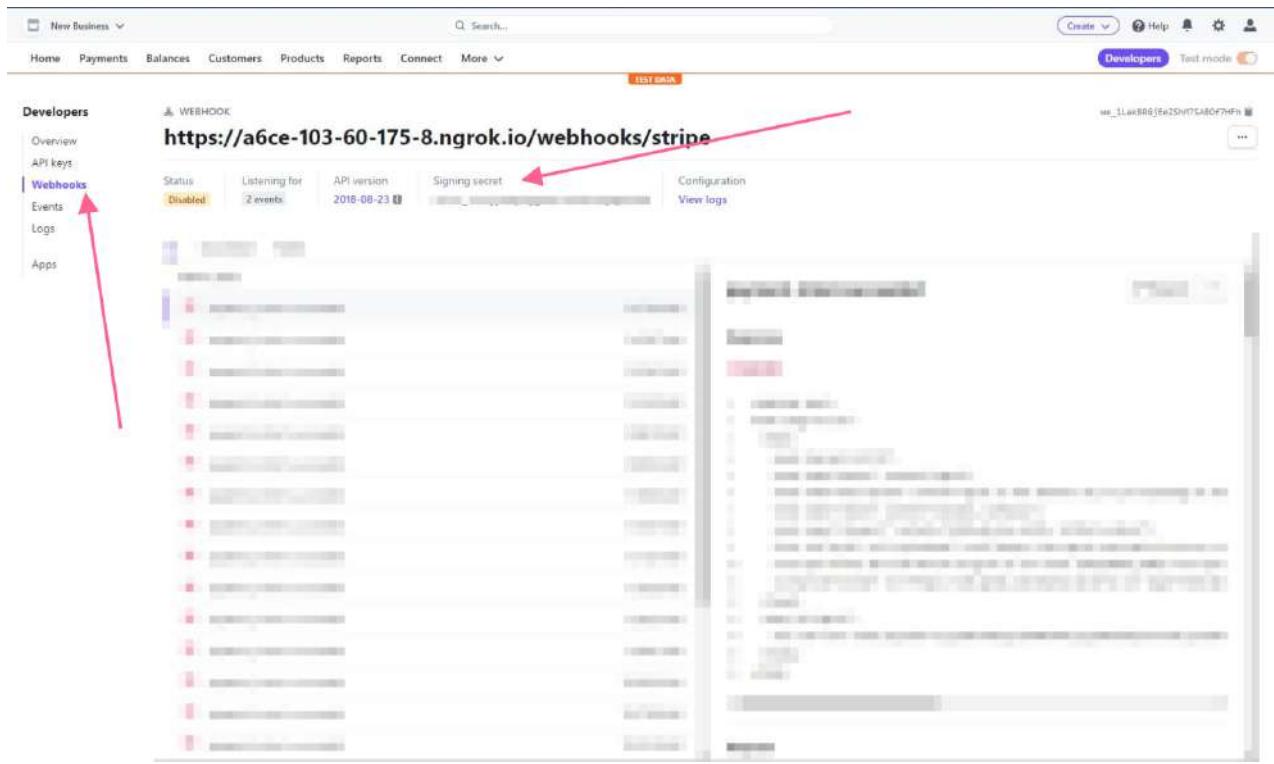
Your current version (2019-12-03) [▼](#)

Select events to listen to

[+ Select events](#)

[Add endpoint](#)

[Cancel](#)



Again! To integrate stripe element in pickbazar then you must have to configure webhook properly.

How can I add stripe element's another payment gateway?

We have a fully functional card management feature in Pickbazar. Where you can save card for further use.

[Click here for more details.](#)

If you enable stripe element from admin settings, then you will get "Others element" option in stripe card.

My Cards

Company	Card Number	Card Owner Name	Expire Date
VISA	**** **** **** 4242	shamim dewan	2/2042

Total Amount: [Redacted] **Order Details:** [Redacted]

Pay

Card **Cash App Pay** **US bank account** **Klarna** **Card**

Secure, 1-click checkout with Link

Card number: 1234 1234 1234 1234 **VISA** Expiration: MM / YY CVC: CVC

Country: United States ZIP: 12345

Save your info for secure 1-click checkout with Link
Pay faster at localhost:3003 and everywhere Link is accepted.

[link](#) • Learn more

Pay **Pay Later** **Back**

: \$0.00 **Shipping Address** : 2148 Straford Park, KY, Winchester,

- Enable your desired stripe element payment gateway from stripe dashboard.

Product settings

- Payments**: Accept payments globally.
- Checkout and Payment Links**: **Payment methods** (highlighted with a red box).
- Billing**: Create and manage subscriptions and invoices.
- Connect**: Embed payments and financial services on your platform.
- Discover more products**: Enable Stripe products to boost revenue, fight fraud, manage finances, and more.

- Turn on your gateway.

Cards
Popular for consumers and businesses to pay online or in person. Stripe supports global and local card networks. [Learn more](#)

Payment method:

- > Cards Popular globally On
- > Cards Bancaires Popular in France Off

Wallets
Improve conversion and reduce fraud on mobile. Customers pay with a stored card or balance. [Learn more](#)

Payment method:

- > Alipay Popular in China Off
- > Apple Pay Popular globally Off
- > Cash App Pay Popular in financial terms Off
- > Google Pay Popular globally Off
- > Link Popular globally Off
- > WeChat Pay Popular in China Off

Bank redirects
Improve convenience and reduce fraud with non-US consumers. Customers pay online using their bank account. [Learn more](#)

Payment method:

- > Bancard Popular in Belgium Off
- > EPS Popular in Austria Off
- > giropay Popular in Germany Off
- > DEAL Popular in the Netherlands Off
- > Rozetka24 Popular in Poland Off
- > Sofort Popular in Germany, Austria, and Switzerland Turn on

Buy now, pay later
Let customers who want to pay in installments. Your business still receives full payment at checkout. [Learn more](#)

Payment method:

- > Affirm Popular in United States and Canada Off
- > Afterpay / Clearpay Popular in Australia, France, New Zealand, Spain, the United Kingdom, and the United States Off
- > Klarna Popular in Europe and the United States Off

Bank debits
Popular for high-value, low-churn payments. Customers give permission for their bank account to be debited. [Learn more](#)

Payment method:

- > SEPA Direct Debit Widely recommended Popular in Europe Off
- > Canadian pre-authorized debit Widely recommended Popular in Canada Off
- > ACH Direct Debit Popular in US Off

How to customize your stripe element payment method.

If you want selective payment method type then list those method name in the "payment_method_types" array
and uncomment this

```
'payment_method_types' =>
['card', 'cashapp', 'alipay', 'wechat_pay', 'etc']
```

and comment

```
'automatic_payment_methods' => ['enabled' => true,],
```

lines

or you can enabled true "automatic_payment_methods" array. so that stripe can handle everything

```
/*
public function getIntent($data): array
{
    try {
        extract($data);
        $intent_array = [];
        $intent_array = [
            'amount' => round($amount, 2) * 100,
            'currency' => $this->currency,
            'description' => 'Marvel Payment',
            /**
             * If you want selective payment method type then list those method name in the "payment_method_types" array
             * and uncomment this "payment_method_types" and comment "automatic_payment_methods" lines
             * or you can enabled true "automatic_payment_methods" array. so that stripe can handle everything
            */
            // 'payment_method_types' => ['card','cashapp','alipay','wechat_pay','etc'],
            'automatic_payment_methods' => [
                'enabled' => true,
            ],
            'metadata' => [
                'order_tracking_number' => $order_tracking_number,
            ]
        ];
        if (isset($customer)) {
            $intent_array['customer'] = $customer;
        }
        $intent = $this->stripe->paymentIntents->create($intent_array);

        return [
            'client_secret' => $intent->client_secret,
            'payment_id' => $intent->id,
            'is_redirect' => false
        ];
    } catch (\Stripe\Exception\CardException $e) {
        throw new HttpException(400, INVALID_CARD);
    }
}
```

Special Notes for Stripe element users.

If we have used any third party system/plugin/packages, then we have always encouraged our respected customers to follow the official documentation for detailed & in-depth knowledge.

Available Stripe Element Payment Method

Stripe element's payment method vary on currency

```
acss_debit, affirm, afterpay_clearpay, alipay, au_becs_debit,
bacs_debit, bancontact, blik, boleto,
card, cashapp, customer_balance, eps, fpx, giropay, grabpay, ideal,
klarna, konbini, link,
oxxo, p24, paynow, paypal, pix, promptpay, sepa_debit, sofort,
us_bank_account, wechat_pay, zip,
```

- Stripe DOCS [home page] -- <https://stripe.com/docs>
- Stripe API Docs -- <https://stripe.com/docs/api>
- Stripe JavaScript Docs -- <https://stripe.com/docs/js>
- Stripe Webhooks Settings -- <https://dashboard.stripe.com/webhooks>
- Stripe Payment Method Settings -- https://dashboard.stripe.com/test/settings/payment_methods
- Stripe Payment Method List -- https://stripe.com/docs/api/payment_methods/list

PayPal

PayPal is an American multinational financial technology company operating an online payments system. In PickBazar we have integrated PayPal APIs which may cover a vast area of PayPal supported region. It will help your business to grow and reach to a wide region.

PayPal integrate inside PickBazar.

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

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in PayPal.

```
# values: sandbox or live (Default: live)
PAYPAL_MODE=sandbox

# Add currency like USD
PAYPAL_CURRENCY=USD

# Change this accordingly for your application.
PAYPAL_NOTIFY_URL=

# force gateway language i.e. it_IT, es_ES, en_US ... (for express
# checkout only)
PAYPAL_LOCALE=en

#validate SSL when creating api client.
PAYPAL_VALIDATE_SSL=
```

```
# PayPal Setting & API Credentials -> sandbox

PAYPAL_SANDBOX_CLIENT_ID=[YOUR_PAYPAL_SANDBOX_CLIENT_ID]
PAYPAL_SANDBOX_CLIENT_SECRET=
[YOUR_PAYPAL_SANDBOX_CLIENT_SECRET_KEY]

# PayPal Setting & API Credentials -> live

PAYPAL_LIVE_CLIENT_ID=[YOUR_PAYPAL_LIVE_CLIENT_ID]
PAYPAL_LIVE_CLIENT_SECRET=[YOUR_PAYPAL_LIVE_CLIENT_SECRET_KEY]

# PayPal webhook settings

PAYPAL_WEBHOOK_ID=[YOUR_PAYPAL_WEBHOOK_URL]
SHOP_URL=[YOUR_SHOP_URL]
```

SHOP_URL=[YOUR_SHOP_URL] This parameter is must have in .env file when PayPal is using. Otherwise the payment redirection will be broken.

- To activate Paypal go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. webhook URL is coming from local development. This static link will dynamically generated in live environment)

[Create Order](#)[Users](#)[Coupons](#)[Taxes](#)[Shipments](#)[Withdrawals](#)[Message](#)[Refunds](#)[Questions](#)[Store Notice](#)[Reviews](#)[Settings](#)Total Revenue
(Last 30 Days)

\$0.00

Total Order
(Last 30 Days)

0

Todays Revenue



\$0.00

Total Shops



7

Sale History

2.0

1.6

1.2

0.8

0.4

Payment

Configure Payment Option

 Enable Cash On Delivery.

Select Payment Gateway

Paypal

Webhook URL

http://127.0.0.1:8000/webhooks/paypal



How to create & setup PayPal information properly?

- Create an account in <https://developer.paypal.com>
- Choose sandbox if you are testing your development environment.
- Click on Default Application. or create a new App.
- You will get your Sandbox API credentials. Copy & paste those inside PickBazar app .env file like mentioned above.

Default Application

App display name: Default Application

SANDBOX API CREDENTIALS

Sandbox Account: [REDACTED]

Client ID: [REDACTED]

Secret: [show](#)

SANDBOX APP SETTINGS

Return URL - Users are redirected to this URL after live transactions. Allow up to three hours for the change to take effect. [Show](#)

App feature options

Account payments: Enable payments and subscriptions payments. [Advanced options](#)

- You will find the Webhook ID and link too.

SANDBOX WEBHOOKS

Configure webhooks to notify your app when certain events occur. To configure a webhook, define your webhook listener URL and a list of events for which to listen. You can configure up to ten webhooks. Each webhook can subscribe to either specific events or all events. To learn more about webhooks, see [webhooks notifications](#).

Webhook ↑	Webhook Id	Event Tracked
[REDACTED]	[REDACTED]	Payment capture completed, Payment capture denied, Payment capture pending, Payment capture refunded, Payment capture reversed

Add Webhook

- 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)

- Create Order
- Users
- Coupons
- Taxes
- Shipments
- Withdrawals
- Message
- Refunds
- Questions
- Store Notice
- Reviews
- Settings

Total Revenue
(Last 30 Days)



\$0.00

Total Order
(Last 30 Days)



0

Todays Revenue



\$0.00

Total Shops



7

Sale History

2.0
1.6
1.2
0.8
0.4

Enable Cash On Delivery

Select Payment Gateway

RazorPay

Webhook URL

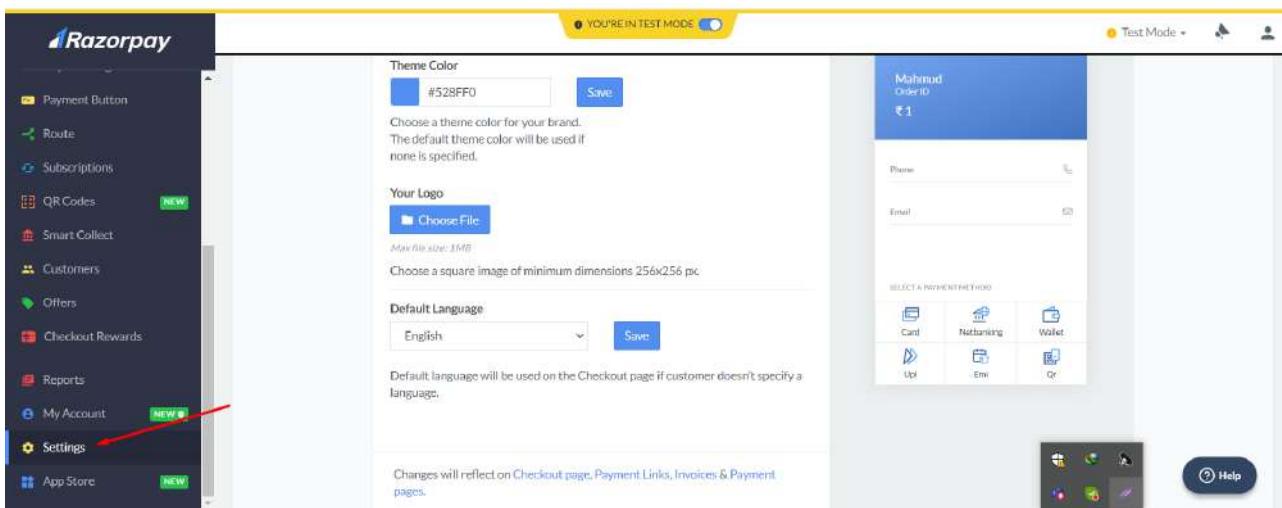
http://localhost:8000/webhooks/razorpay

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

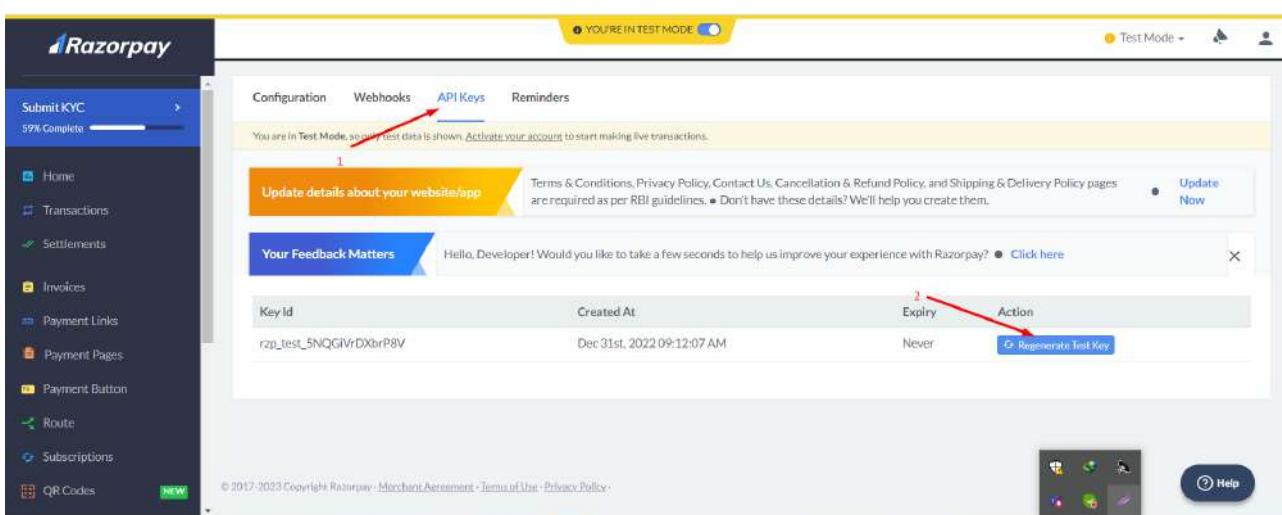
```
NEXT_PUBLIC_RAZORPAY_KEY=[YOUR_RAZORPAY_KEY_ID]
```

How to create & setup RazorPay information properly?

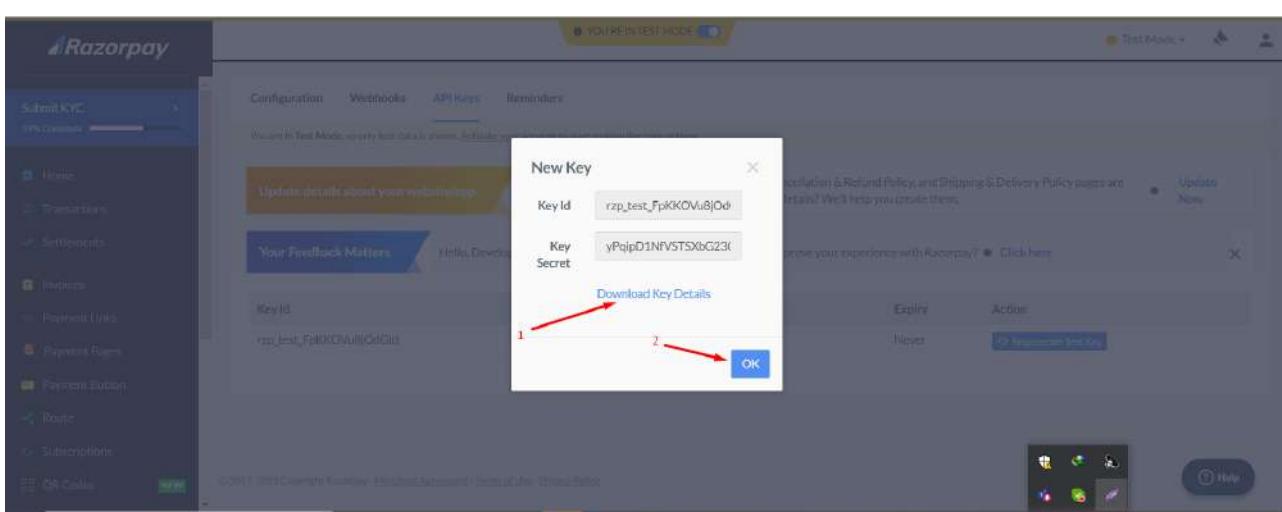
- Go to the RazorPay website & login <https://razorpay.com> If you aren't a registered user, Complete the RazorPay registration first.
- After logged in into RazorPay dashboard, Click on Settings



- 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.

You are in Test Mode, so only test data is shown. [Activate your account](#) to start making live transactions.

Your Feedback Matters Hello, Developer! Would you like to take a few seconds to help us improve your experience with razorpay? • [Click here](#)

URL	Status	Events	Last Updated
REDACTED	Disabled	12 events	12 Dec 2022, 11:39:45 am

Showing 1 - 5

Documentation [+ Add New Webhook](#)

© 2017-2023 Copyright Razorpay | [Merchant Agreement](#) | [Terms of Use](#) | [Privacy Policy](#)

- Enter Webhook URL , Secret key , Alert Email and tick mark the checkbox in Active Event then click on create Webhook

Webhook Setup

Webhook URL: https://yourdomain.com/webhooks/razorpay

Secret: REDACTED

We strongly recommend using secrets for security. [View Secured](#)

Alert Email: youremail@demo.com

Receive email alerts for webhook failures

Search

Active Events

Payment Events

payment.authorized
 payment.authorized
 payment.failed
 payment.captured
 payment.dispute.created
 payment.dispute.won
 payment.dispute.lost
 payment.refund.created
 payment.refund.won
 payment.refund.lost

3 events selected

Cancel Create Webhook

- Copy & paste those inside PickBazar app .env file like mentioned above. after that test your RazorPay Payment Status

Payments Refunds Batch Refunds Orders Disputes

You are in Test Mode, so only test data is shown. [Activate your account](#) to start making live transactions.

Payment Id	Duration	Status	Email	Notes	Status	
pay_KzaX5rehTtoAUF	order_KzaX5rehTtoAUF	\$53.08	customer@demo.com	+19365141641631	03 Jan 2023, 11:25:35 am	Captured
pay_Kza7u304PkiCle	order_Kza7u304PkiCle	\$54.90	customer@demo.com	+19365141641631	03 Jan 2023, 11:13:08 am	Captured
pay_KzcEcqgvoRA31ldI	order_KzcEcqgvoRA31ldI	€30.45	customer@demo.com	+19365141641631	29 Dec 2022, 11:47:29 am	Captured
pay_KzcOvYvourBDQsw	order_KzcOvYvourBDQsw	€57.75	customer@demo.com	+19365141641631	29 Dec 2022, 11:45:21 am	Captured
pay_KzbmIB1dL9qgQ	order_KzbmIB1dL9qgQ	€30.45	customer@demo.com	+19365141641631	29 Dec 2022, 11:20:40 am	Captured
pay_KzcbkklouHrq1H	order_KzcbkklouHrq1H	€19.95	customer@demo.com	+19365141641631	29 Dec 2022, 11:18:45 am	Captured
pay_KzabPQe5Hbk2rU	order_KzabPQe5Hbk2rU	€51.45	customer@demo.com	+19365141641631	29 Dec 2022, 11:18:45 am	Captured

Submit KYC 59% Complete

Home Transactions Settlements Invoices Payment Links Payment Pages Payment Buttons Route Subscriptions QR Codes Smart Collect Customers

Test Mode Documentation + Add New Webhook

Help

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-payment>

Mollie

Mollie is a Payment Service Provider (PSP) that processes online payments for companies. If you buy something online from one of our merchants, we make sure that your money is transferred safely from your bank to the merchant's bank. Since we arrange the payment process, you may see Mollie or Stg Mollie Payments on your bank statement.

Mollie integrate inside PickBazar.

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

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in Mollie.

```
MOLLIE_KEY=[YOUR_MOLLIE_API_KEY]
```

Mollie Webhook settings

```
SHOP_URL=[YOUR_SHOP_URL]  
MOLLIE_WEBHOOK_URL=[YOUR_MOLLIE_WEBHOOK_URL]
```

SHOP_URL=[YOUR_SHOP_URL] This parameter is must have in `.env` file when Mollie is using. Otherwise the payment redirection will be broken.

- To activate Mollie go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. Mollie Webhook URL is coming from local development. This static link will dynamically generated in live environment)

[Create Order](#)[Users](#)[Coupons](#)[Taxes](#)[Shipments](#)[Withdrawals](#)[Message](#)[Refunds](#)[Questions](#)[Store Notice](#)[Reviews](#)[Settings](#)Total Revenue
(Last 30 Days)Total Order
(Last 30 Days)

Todays Revenue



Total Shops



\$0.00

0

\$0.00

7

Sale History

2.0
1.6
1.2
0.8
0.4**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 http http://localhost:8000
(Ctrl+C to quit)

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

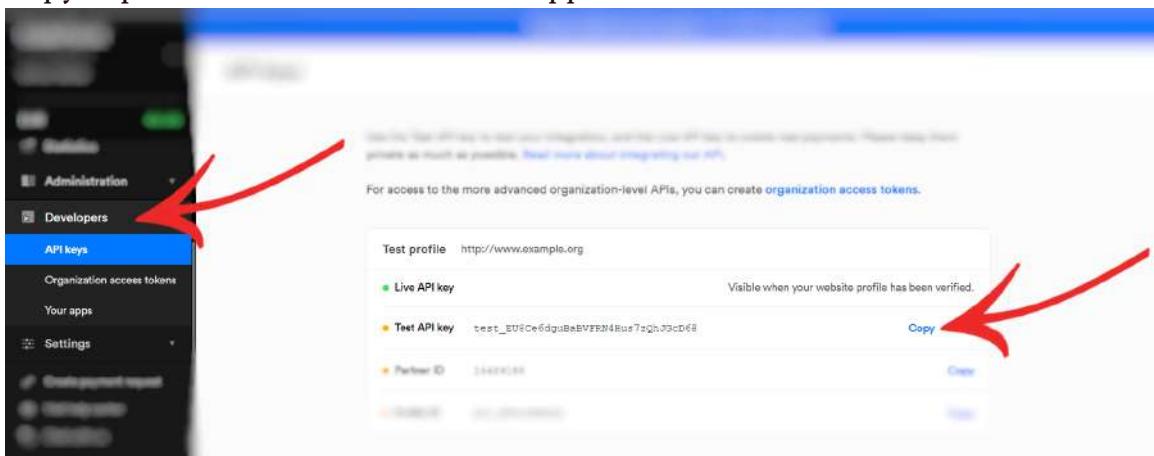
Connections        ttl     opn     rt1     rt5     p50     p90
                  0       0     0.00    0.00    0.00    0.00
```

- 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.id/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, 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>

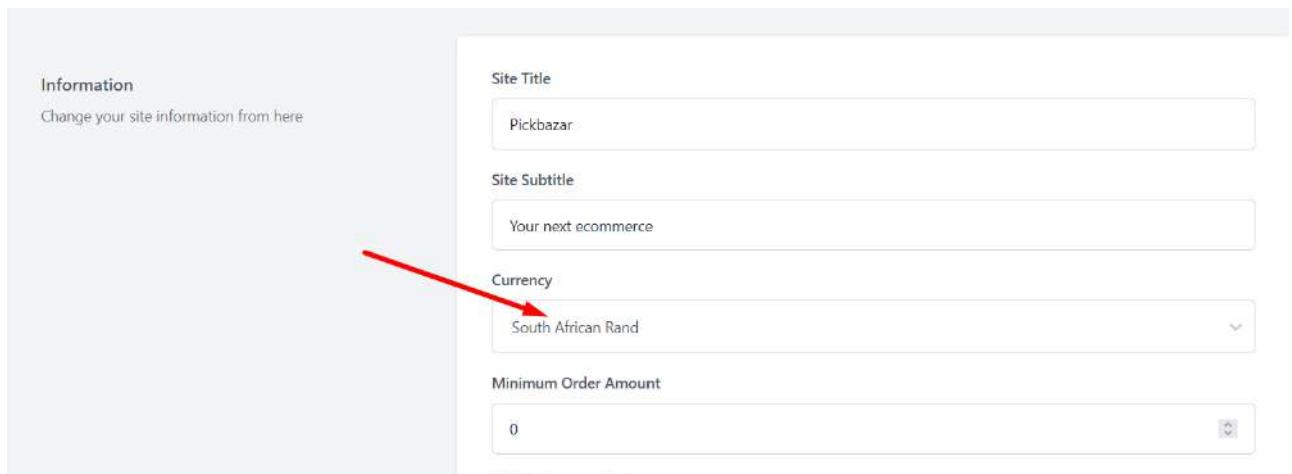
Paystack

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).



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_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)

The screenshot shows the PickBazar admin dashboard. On the left, there is a sidebar with various menu items: Create Order, Users, Coupons, Taxes, Shipments, Withdrawals, Message, Refunds, Questions, Store Notice, Reviews, and Settings. The 'Settings' item is highlighted with a red box and has a red arrow pointing to it from the bottom-left. The main content area displays four summary cards: Total Revenue (\$0.00), Total Order (0), Today's Revenue (\$0.00), and Total Shops (7). Below these cards is a chart titled 'Sale History' with data points at 0.4, 0.6, 0.8, 1.2, 1.6, and 2.0. A modal window is open over the main content, titled 'Payment'. It contains the following fields:

- Enable Cash On Delivery (checkbox)
- Select Payment Gateway (dropdown menu showing 'Paystack')
- Webhook URL (input field containing 'http://localhost:8000/webhooks/paystack')

- Copy the webhook url & registered in Paystack Dashboard.

The screenshot shows a configuration interface with two input fields and a button. The top field is labeled 'Test Callback URL' and contains the placeholder 'Test Callback URL'. Below it is another field labeled 'Test Webhook URL' containing the URL 'https://pickbazarapi.com/webhooks/paystack'. A red arrow labeled '1' points from the 'Test Webhook URL' field to the 'Save Changes' button at the bottom right. Another red arrow labeled '2' points from the 'Test Webhook URL' field to the same button.

- 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.

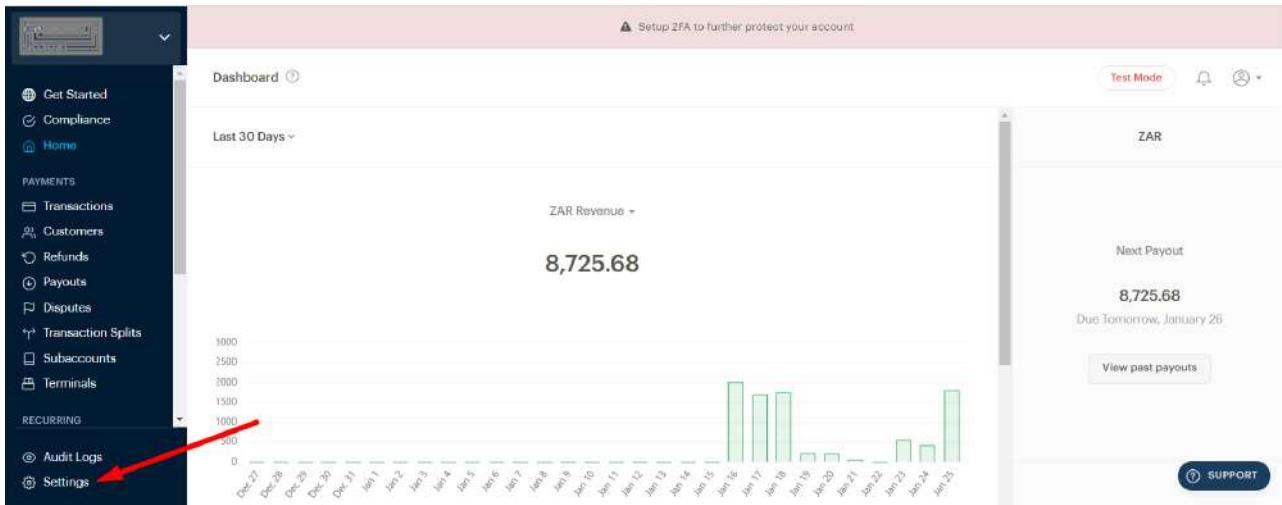
The screenshot shows a terminal window titled 'ngrok'. It displays the command '- ngrok http http://localhost:8000'. Below the command, it says '(Ctrl+C to quit)'. The output shows session details: Session Status is 'online', Account is [REDACTED], Version is '3.1.0', Region is 'India (in)', Latency is '116ms', and the Web Interface is 'http://127.0.0.1:4040'. Under Forwarding, it shows a tunnel mapping from 'https://c852-37-111-243-195.in.ngrok.io' to 'http://localhost:8000'. The Connections table shows 0 ttl, 0 open connections, and 0.00 ms for rt1, rt5, p50, and p90.

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

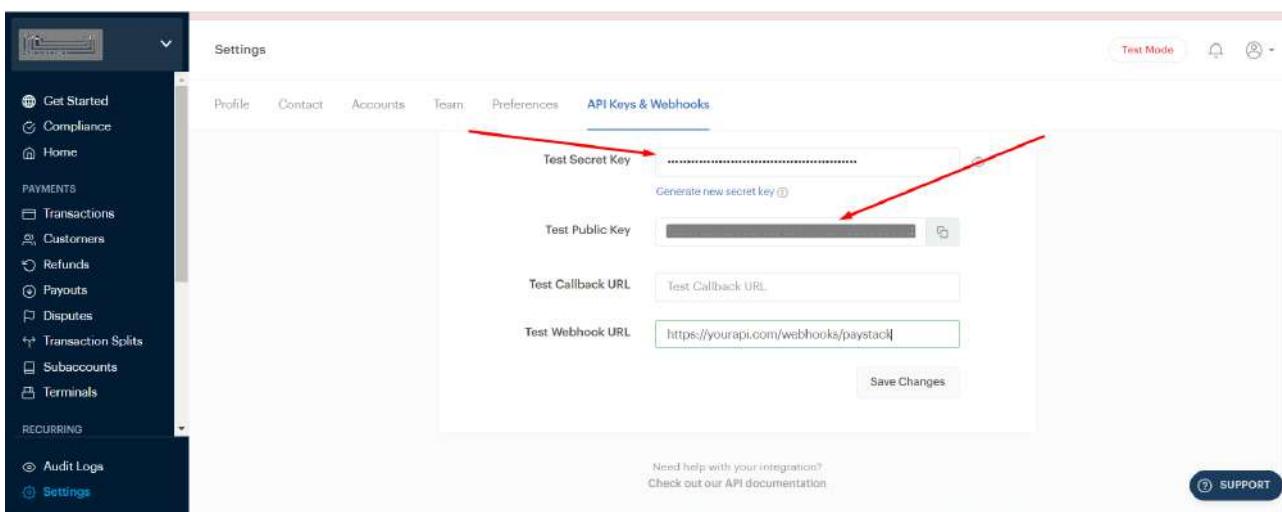
The screenshot shows a configuration interface similar to the first one. It has 'Test Callback URL' and 'Test Webhook URL' fields. The 'Test Webhook URL' field contains the URL 'https://dd15-2400-c600-345f-88ce-a1d9-b41-3c43-cf65.in.ngrok.io/webhooks/paystack'. A red box highlights this URL. A 'Save Changes' button is visible at the bottom right.

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.



- 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)

SSLCOMMERZ

SSLCOMMERZ is the largest payment gateway aggregator in Bangladesh. In PickBazar we have integrated SSLCOMMERZ APIs which may cover a vast area of SSLCOMMERZ supported region. It will help your business to grow and reach to a wide region.

SSLCOMMERZ integrate inside PickBazar.

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

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in SSLCOMMERZ.

```
SHOP_URL=[YOUR_SHOP_URL]

# Values: true or false (Default: true)
SSLC_ALLOW_LOCALHOST=true

# SSLCOMMERZ setting & API Credentials

SSLC_STORE_ID=[YOUR_SSLCOMMERZ_STORE_ID]
SSLC_STORE_PASSWORD=[YOUR_SSLCOMMERZ_STORE_PASSWORD]
```

SHOP_URL=[YOUR_SHOP_URL] This parameter is must have in .env file when SSLCOMMERZ is using. Otherwise the payment redirection will be broken.

Package usaged Inside PickBazar [SSLCOMMERZ-Package](#)

- To activate SSL-Commerz go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. webhook URL is coming from local development. This static link will dynamically generated in live environment)

[Create Order](#)[Users](#)[Coupons](#)[Taxes](#)[Shipments](#)[Withdrawals](#)[Message](#)[Refunds](#)[Questions](#)[Store Notice](#)[Reviews](#)[Settings](#)Total Revenue
(Last 30 Days)

\$0.00

Total Order
(Last 30 Days)

0

Todays Revenue



\$0.00

Total Shops



7

Sale History

2.0

1.6

1.2

0.8

0.4

Payment

Configure Payment Option

 Enable Cash On Delivery

Select Payment Gateway

SslCommerz

Webhook URL

http://localhost:8000/webhooks/sslcommerz

How to create & setup SSLCOMMERZ information properly?

- For **local** Create an account in <https://developer.sslcommerz.com/registration/>
- For going **live** Create an account in <https://signup.sslcommerz.com/register>
- After successfully creating your sslcommerz account you will get your **SSLC_STORE_ID** and **SSLC_STORE_PASSWORD** through email.

Online Payment SSLCommerz <developer@sslcommerz.com>

☆ ⏪ ⏴

SSLCommerz



Welcome to SSLCommerz Sandbox

Congratulations Sandbox Store ID Created

We highly appreciate if you first connect with the sandbox and then move to the live system. Please inform us if you need to change Registered Store URL for testbox.



Store ID: [REDACTED]
Store Password (API/Secret Key): [REDACTED]

Merchant Panel URL: <https://sandbox.sslcommerz.com/manage/>
(Credential as you inputted in the time of registration)

Store name: [REDACTED]
Registered URL: [REDACTED]
Session API to generate transaction: <https://sandbox.sslcommerz.com/gwprocess/v3/api.php>
Validation API: <https://sandbox.sslcommerz.com/validator/api/validationserverAPI.php?wsdl>
Validation API (Web Service) name: <https://sandbox.sslcommerz.com/validator/api/validationserverAPI.php>

You may check our plugins available for multiple carts and libraries:
<https://github.com/sslcommerz>

YOUR INFORMATION AT OUR SYSTEM

Merchant Id: [REDACTED]
Merchant Name: [REDACTED]
Name: [REDACTED]

- You will find the IPN setting under **My Stores**.

IPN Settings

IPN at HTTP Listener

Enable HTTP Listener

HTTP POST Parameters will be thrown to the IPN_HTTP_URL as amount=1150.00&bank_tran_id=151114130739MqCBNx5&card_brand=VISA&card_issuer=BRAC+BANK%2C+LTD.&card_issuer_country=Bangladesh&card_issuer_country_code=BD&card_no=432149XXXXXX0567&card_type=VISA_Brac+Bank+LTD&status=VALID&store_amount=1104.00&store_id=cfts14b3ac58771e&tran_date=2015-11-14+13%3A07%3A12&tran_id=5645cd94ab48&val_id=1511141307426jHIBUKtDEGRj&verify_sign=9ae83ce8b421937fe9de93a379983&verify_key=amount%2Cbank_tran_id%2Ccard_brand%2Ccard_issuer%2Ccard_issuer_country%2Ccard_issuer_country_code%2Ccard_no%2Ccard_type%2Ccurrency%2Cstatus%2Cstore_amount%2Cstore_id%2Ctran_date%2Ctran_id%2Cval_id

Requirement of your HTTP IPN Listener

- The Listener must use the common port like 80 or 443
- Your IPN Listener must be reachable from Internet
- White-list the SSLCommerz IPs at your network firewall
 - Sandbox Access Requirement
 - TCP 80 or 443 needs to be opened at your system from 103.26.139.87
 - Your system needs to be able to reach TCP 443 of 103.26.139.87
 - Live Access Requirement
 - TCP 80 or 443 needs to be opened at your system from 103.26.139.81
 - Your system needs to be able to reach TCP 443 of 103.26.139.148

2011-2023 © SSLCommerz - All rights reserved. It is powered by SSL Wireless.

- Then Add the IPN notification if you want to up & running the services.

- VALID** : A successful transaction.
- FAILED** : Transaction is declined by customer's Issuer Bank.
- CANCELLED** : Transaction is cancelled by the customer.
- UNATTEMPTED** : Customer did not choose to pay any channel.
- EXPIRED** : Payment Timeout.
- At last, For going live with your application please follow this official documentation. <https://developer.sslcommerz.com/doc/v4/#grab-the-notification>
- For any kind of further customization you can also see this documentation <https://github.com/dgvai/laravel-sslcommerz>

Xendit

Xendit is a digital payments infrastructure provider that operates in Southeast Asia, particularly in Indonesia, the Philippines, and Singapore. Xendit offers a variety of payment services, including online and offline payments, virtual accounts, and digital wallets. It also supports multiple payment methods, such as credit and debit cards, bank transfers, and e-wallets.

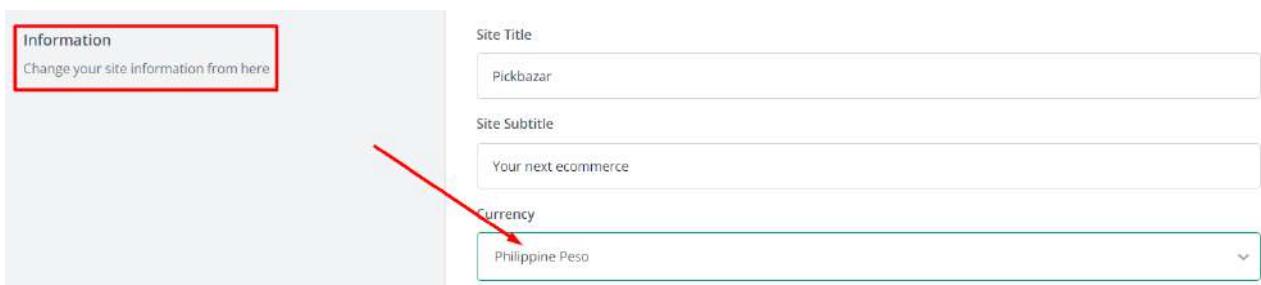
Xendit integrate inside PickBazar.

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

- First go to settings from PickBazar admin dashboard. Inside settings you will find Currency option. Select your Currency. If you want to Test Indonesian Account, then Select Indonesian rupiah(IDR).



Or If you want to Test philippine Account, then Select Philippine peso(PHP).



WARNING: Any order below the minimum amount of **10000 IDR or PHP** will not be processed, please make sure to meet the minimum order requirement before placing an order.

For more information: [Xendit currency is available.](#)

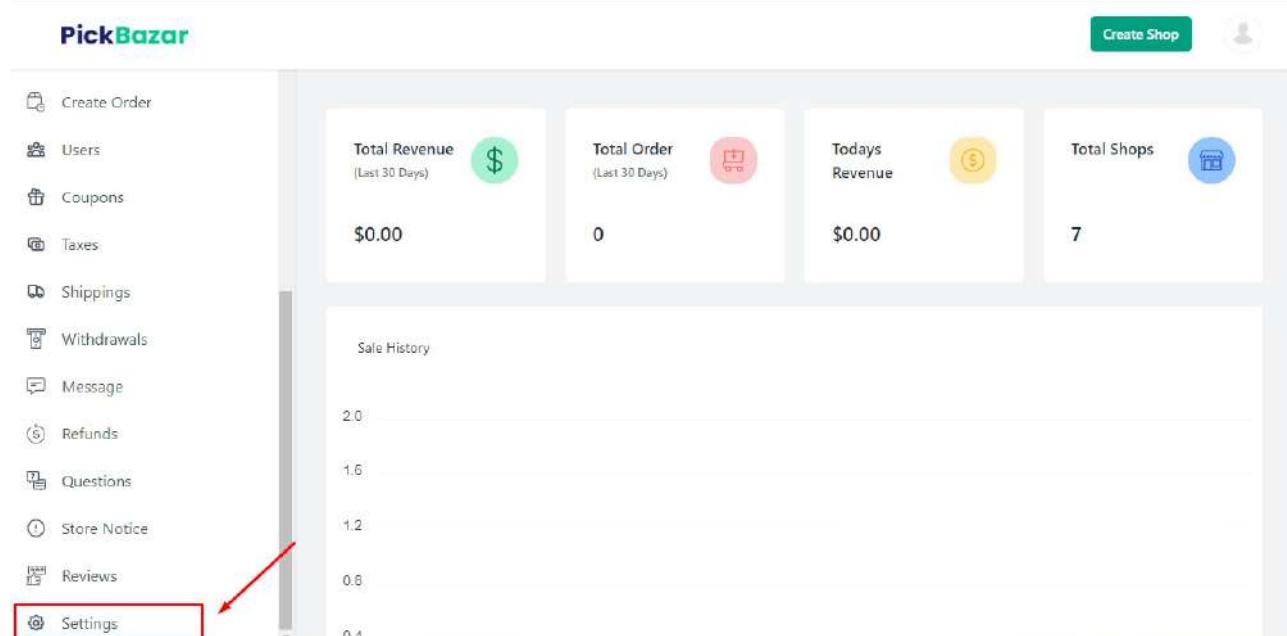
- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in [Xendit official website](#).

```
API_KEY=[YOUR_XENDIT_SECRET_KEY]
XENDIT_WEBHOOK_URL=[YOUR_XENDIT_WEBHOOK_URL]
XENDIT_CALLBACK_TOKEN=[YOUR_XENDIT_CALLBACK_VERIFICATION_TOKEN]
```

SHOP_URL=[YOUR_SHOP_URL] This parameter is must have in .env file when Xendit is using. Otherwise the payment redirection will be broken.

Xendit Webhook settings

- To activate Xendit go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. Xendit Webhook URL is coming from local development. This static link will dynamically generated in live environment)



The screenshot shows the PickBazar admin dashboard. On the left, there is a sidebar with various menu items: Create Order, Users, Coupons, Taxes, Shipments, Withdrawals, Message, Refunds, Questions, Store Notice, Reviews, and Settings. The 'Settings' item is highlighted with a red box and has a red arrow pointing to it from the bottom-left. The main content area displays summary statistics: Total Revenue (\$0.00), Total Orders (0), Today's Revenue (\$0.00), and Total Shops (7). Below these stats is a chart titled 'Sale History' with a Y-axis ranging from 0.4 to 2.0.

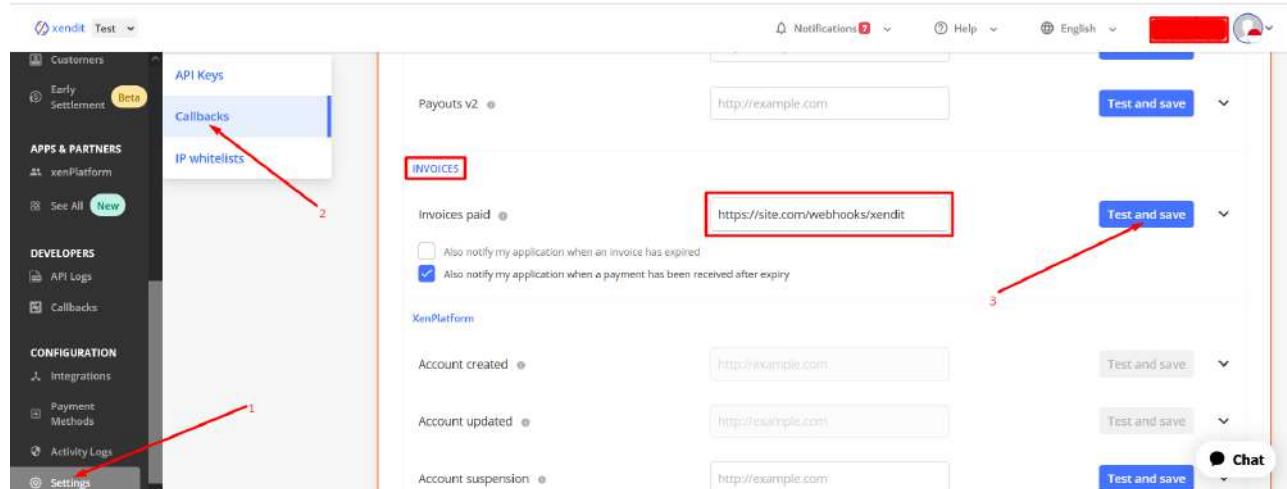
Payment
Configure Payment Option

Enable Cash On Delivery

Select Payment Gateway
Xendit

Webhook URL
`http://localhost:8000/webhooks/xendit`

- Copy the webhook url & registered in Xendit Dashboard.



The screenshot shows the Xendit dashboard. On the left, there is a sidebar with sections: Customers, APPS & PARTNERS (xenPlatform), DEVELOPERS (API Logs, Callbacks), and CONFIGURATION (Integrations, Payment Methods, Activity Logs, Settings). The 'Settings' item is highlighted with a red box and has a red arrow pointing to it from the bottom-left. In the main content area, under 'API Keys', the 'Callbacks' tab is selected (highlighted with a blue box and a red arrow labeled 1). Under the 'INVOICES' section, the 'Invoices paid' field contains the URL `https://site.com/webhooks/xendit`. There are two checkboxes below it: 'Also notify my application when an invoice has expired' (unchecked) and 'Also notify my application when a payment has been received after expiry' (checked). A red box highlights this field, and a red arrow labeled 2 points to it. To the right, there are 'Test and save' buttons for each section. A red arrow labeled 3 points to the 'Test and save' button for the 'Invoices paid' section.

- 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 Status      online
Account             [REDACTED]
Version            3.1.0
Region              India (in)
Latency            116ms
Web Interface     http://127.0.0.1:4040
Forwarding         https://c852-37-111-243-195.in.ngrok.io -> http://localhost:8000

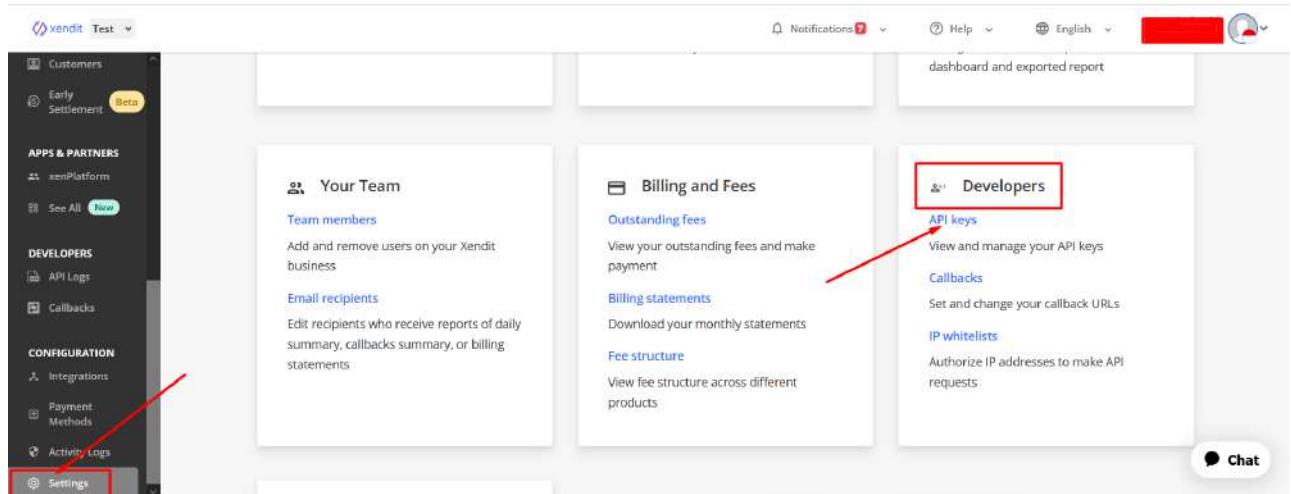
Connections        ttl     opn     rt1     rt5     p50     p90
                    0       0       0.00   0.00   0.00   0.00
```

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

```
94 # Payment -> Xendit
95 API_KEY=
96 XENDIT_WEBHOOK_URL=https://45ae-2400-c600-3425-9e4-f4b6-285d-ab9d-70a1.in.ngrok.io/webhooks/xendit
97 XENDIT_CALLBACK_TOKEN=
```

How to create & setup Xendit information properly?

- Go to the [Xendit official website](#) & login If you aren't a registered user, Complete the Xendit registration first.
- After logged in into Xendit dashboard, Click on Settings, after that click of API keys.



- After That Click on **Generate secret key** for new Xendit Secret Key.

API Keys

Callbacks

IP whitelists

Test data

API Keys

Secret keys

Secret keys are used to authenticate API requests coming from your servers.

Key name	Permissions	Created (GMT +6)	Last Used (GMT +6)	Beta
You don't have any secret API keys yet.				

[+ Generate secret key](#)

- Then API key name, give the permissions and click on Generate key button.
- Next, for Callback verification token, Go to settings and click on Callbacks.

Send Payments

Customers

Early Settlement Beta

APPS & PARTNERS

See All New

Developers

API Logs

Callbacks Callbacks

Configuration

Integrations

Payment Methods

Activity Log

Settings

Configure Checkout appearance, available payment methods and notifications

Disbursement

Disbursement notifications

Set how recipients are notified about disbursements you send

Security settings, and find your referral code

Time zone

Configure the time zone preference on the dashboard and exported report

Your Team

Team members

Add and remove users on your Xendit business

Email recipients

Edit recipients who receive reports of daily summary, callbacks summary, or billing statements

Billing and Fees

Outstanding fees

View your outstanding fees and make payment

Billing statements

Download your monthly statements

Fee structure

View fee structure across different products

Developers

API keys

View and manage your API keys

Callbacks Callbacks

Set and change your callback URLs

IP whitelists

Authorize IP addresses to make API requests

Make sure you enable auto-retry for failed callback, by enabling this, you will have better experience in handling your callback response.

Next, is to obtain your callback verification token. Click View Callback Verification Token, Enter your password, Copy & paste those inside PickBazar API `.env` file like mentioned above.

Test data

Callbacks

Add your callback URL's here. To learn how to add and manage your callbacks, visit our [documentation page](#).

Enable auto-retry for failed callback

Automatic retry for failed callback with exponential delay. Learn more [here](#)

Callback verification token

Sent with every callback, use the token to validate that a callback came from our servers

Callback verification token

[View Callback Verification Token](#)

Special Notes for Xendit.

If we have used any third party system/plugin/packages, then we have always encouraged our respected customers to follow the official documentation for detailed & in-depth knowledge.

- [Xendit official documentation](#)
- [Xendit Webhooks documentation](#)

Iyzico

iyzico is one of Turkey's most well-known payment gateways. iyzico provides safe payment solutions to internet businesses and individuals of various sizes. You can quickly and easily incorporate the iyzico payment into your e-commerce website and begin accepting payments.

Iyzico integrate inside PickBazar.

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

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in Iyzico.

```
# Remove sandbox- for Production

IYZIPAY_BASE_URL=[https://sandbox-api.iyzipay.com]

IYZIPAY_API_KEY=[YOUR_KEY_HERE]

IYZIPAY_SECRET_KEY=[YOUR_SECRET_KEY_HERE]

SHOP_URL=[YOUR_SHOP_URL]
```

SHOP_URL=[YOUR_SHOP_URL] This parameter is must have in .env file when Iyzico is using. Otherwise the payment redirection will be broken.

- To activate Paypal go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. webhook URL is coming from local development. This static link will dynamically generated in live environment)

[Create Order](#)[Users](#)[Coupons](#)[Taxes](#)[Shipments](#)[Withdrawals](#)[Message](#)[Refunds](#)[Questions](#)[Store Notice](#)[Reviews](#)[Settings](#)Total Revenue
(Last 30 Days)

\$0.00

Total Order
(Last 30 Days)

0

Todays Revenue



\$0.00

Total Shops



7

Sale History

2.0

1.6

1.2

0.8

0.4

Payment

Configure Payment Option

 Enable Cash On Delivery

Select Payment Gateway

Iyzico

Webhook URL

http://localhost:8000/webhooks/izico

How to create & setup Iyzico information properly?

- Create an account in <https://merchant.iyzipay.com/>
- Create a sandbox account in <https://sandbox-merchant.iyzipay.com/>
- Choose sandbox if your are testing your development environment.
- Click on **Setting** then **Merchant Settings..**
- You will get your Sandbox API credentials. Copy & paste those inside PickBazar app .env file like mentioned above.

The screenshot shows the 'Merchant Settings' page in the iyzico sandbox. On the left, a sidebar menu is open, with 'Settings' highlighted by a red box and a red arrow pointing to it. The main content area is titled 'Merchant Settings' and contains two sections: 'API Keys' and 'Cep POS API Keys'. In the 'API Keys' section, there are fields for 'API Key' and 'Secret Key', each with a 'Copy' button. A 'RESET SECRET KEY' button is also present. In the 'Cep POS API Keys' section, there are similar fields for 'API Key' and 'Secret Key', each with a 'Show detail' link.

- You will find the Webhook ID and logo link too.

The screenshot shows the 'Merchant Settings' page in the iyzico sandbox. The sidebar menu has 'Settings' highlighted by a red box and a red arrow pointing to it. The main content area contains two sections: 'Merchant Logo' and 'Merchant Notifications'. In the 'Merchant Logo' section, there is a 'Merchant Logo Url' input field with a red arrow pointing to it. Below it is a 'SAVE' button. In the 'Merchant Notifications' section, there is a toggle switch labeled 'Receive notifications for payments.' followed by a 'Merchant Notification Url' input field with a red arrow pointing to it. Below it is another 'SAVE' button.

- At last, For going live with your application please follow this official documentation. <https://dev.iyzipay.com/en/api>

Paymongo

The PayMongo API allows you to build customized and unique payment experiences for your website, application or service. Our goal is to give you access to pretty much all the necessary resources to integrate payments into your platform.

Paymongo integrate inside PickBazar.

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

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in Paymongo.

```
PAYMONGO_PUBLIC_KEY=[YOUR_PAYMONGO_PUBLIC_KEY]
PAYMONGO_SECRET_KEY=[YOUR_PAYMONGO_SECRET_KEY]

# Paymongo webhook settings

PAYMONGO_WEBHOOK_SIG=[YOUR_PAYMONGO_WEBHOOK_URL]
```

- To activate Paymongo go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. webhook URL is coming from local development. This static link will dynamically generated in live environment)

[Create Order](#)[Users](#)[Coupons](#)[Taxes](#)[Shipments](#)[Withdrawals](#)[Message](#)[Refunds](#)[Questions](#)[Store Notice](#)[Reviews](#)[Settings](#)**Information**

Change your site information from here

Site Title

Pickbazar

Site Subtitle

Your next ecommerce

Currency

Philippine Peso

Minimum Order Amount

0

Payment

Configure Payment Option

Enable Cash On Delivery**Select Payment Gateway**

Paymongo

Webhook URL

http://localhost:8000/webhooks/paymongo

Menu

- Activate your account
2 out of 4 steps completed
- [Home](#)
- [Links](#)
- [Pages](#)
- [Payments](#)
- [Paymoto](#)
- [Developers](#)
- [Security logs](#)
- [Settings](#)

TEST API Keys

[Public Key](#) [Secret Key](#) [Read our documentation](#)

Plugins

Install PayMongo payment methods on Shopify, Prestashop, WooCommerce, and Magento
<https://developers.paymongo.com/docs/plugins>

How to create paymongo webhook url?

- Open your terminal in API project for artisan command
- Then run this command

```
php artisan paymongo:webhook
```

You can visit for paymongo.

- [paymongo official documentation](#)
- [paymongo Webhooks documentation](#)

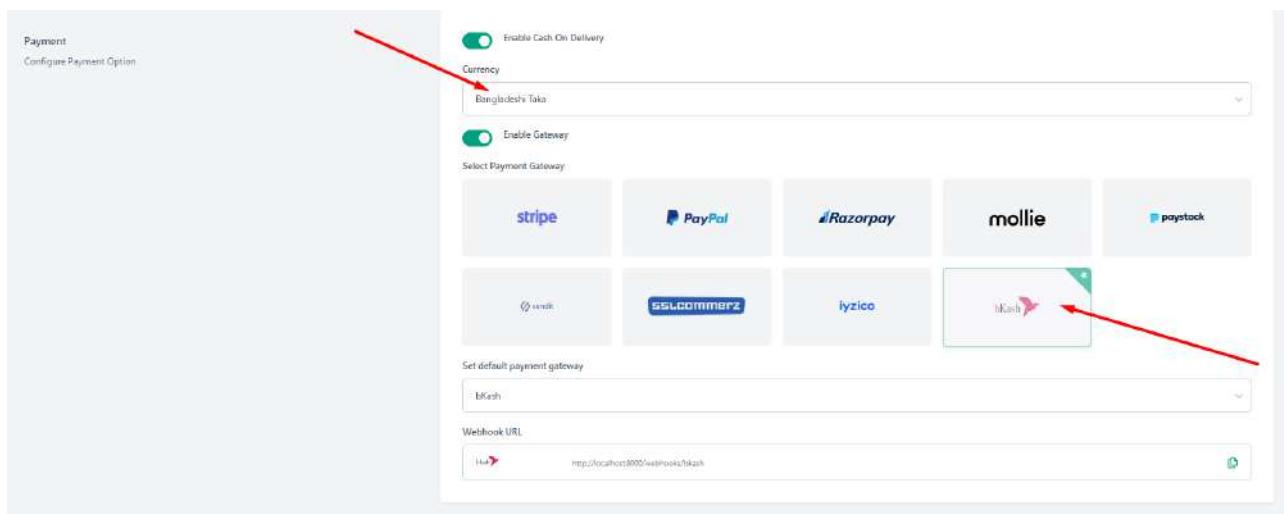
bKash

bKash is a mobile financial service in Bangladesh that allows users to make financial transactions through their mobile phones. It was launched in 2011 as a subsidiary of BRAC Bank, one of the leading banks in Bangladesh. bKash provides a range of services such as money transfer, mobile recharge, bill payment, merchant payments, and more.

bKash integrate inside PickBazar.

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

- First go to settings from PickBazar admin dashboard. Inside settings you will find Currency option. Select Bangladeshi Taka (BDT) Currency. bKash primarily operates in the currency of Bangladesh, which is the Bangladeshi Taka (BDT).



- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in [bkash official website](#).

```
BKASH_SANDBOX=true #for production use false
BKASH_APP_KEY=
BKASH_APP_SECRET=
BKASH_USERNAME=
BKASH_PASSWORD=
```

SHOP_URL=[YOUR_SHOP_URL] This parameter is must have in .env file when bKash is using. Otherwise the payment redirection will be broken.

How to create & setup bKash information properly?

- Go to the [bKash official website](#) & Click on Live Chat Button,



- After That Click on `customer service` for new merchant API Keys.



- Then apply API keys for merchant mobile Number.
- 3-10days after they will give you your

`BKASH_APP_KEY`

`BKASH_APP_SECRET`

`BKASH_USERNAME`

& `BKASH_PASSWORD`.

Special Notes for bKash.

If we have used any third party system/plugin/packages, then we have always encouraged our respected customers to follow the official documentation for detailed & in-depth knowledge.

- [bKash official documentation](#)
- [bKash API reference](#)

Flutterwave

Flutterwave is a secure and versatile payment gateway that facilitates seamless online transactions. It enables businesses to accept payments from various sources, including credit cards, debit cards, and mobile money, in multiple currencies. With user-friendly integration and advanced fraud protection, Flutterwave ensures secure and hassle-free payment processing. It also offers comprehensive analytics and reporting tools, empowering businesses to track and manage their transactions effectively. With its reliability and convenience, Flutterwave is a trusted solution for online payments, providing a smooth and secure experience for both merchants and customers.

Flutterwave integrate inside PickBazar.

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

- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in [RazorPay](#).
- Inside api there is `.env` file. You have to copy & paste this line of codes inside the `.env` file. We will discuss later about how to create those API keys in [Paystack official website](#).

```
FLW_PUBLIC_KEY=[YOUR_FLW_PUBLIC_KEY]
FLW_SECRET_KEY=[YOUR_FLW_SECRET_KEY]
FLW_SECRET_HASH=[FLW_SECRET_HASH]
```

SHOP_URL=[YOUR_SHOP_URL] This parameter is must have in .env file when Flutterwave is using. Otherwise the payment redirection will be broken.

Flutterwave Webhook settings

- To activate Flutterwave go to settings from PickBazar admin dashboard. Inside settings you will find configure payment option. (e.g. Paystack Webhook URL is coming from local development. This static link will dynamically generated in live environment)

The screenshot shows the PickBazar admin dashboard. On the left, there is a sidebar with various menu items: Create Order, Users, Coupons, Taxes, Shipments, Withdrawals, Message, Refunds, Questions, Store Notice, Reviews, and Settings. The 'Settings' item is highlighted with a red box and has a red arrow pointing to it from the top-left. The main content area displays real-time statistics: Total Revenue (\$0.00), Total Orders (0), Today's Revenue (\$0.00), and Total Shops (7). Below these stats is a chart titled 'Sale History' with a Y-axis ranging from 0.4 to 2.0. The 'Settings' page itself shows payment gateway options. It includes an 'Enable Gateway' toggle switch, a 'Select Payment Gateway' section with icons for stripe, PayPal, Razorpay, mollie, paymongo, paystack, venmo, SSLCOMMERZ, lyzico, and bKash. A specific 'Flutterwave' button is highlighted with a red box and has a red arrow pointing to it from the bottom-left. Below this, there is a 'Set default payment gateway' dropdown set to 'Flutterwave'. Under 'Webhook URL', there are two entries: 'stripe' with the URL 'http://localhost:8000/webhooks/stripe' and 'flutterwave' with the URL 'http://localhost:8000/webhooks/flutterwave'. Both URLs have green copy icons at the end of their respective lines.

- Copy the webhook url & registered in Paystack Dashboard.

General Bank Accounts Users API **Webhooks** Account Settings

Live Webhooks

URL

Secret hash What is secret hash?

Receive Webhook response in JSON format
 Enable Webhook retries
 Enable failed transactions webhook

<https://yourdomain/webhooks/flutterwave> [Edit Webhook](#)

Enable V3 Webhooks

- 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 Status      online
Account             [REDACTED]
Version            3.1.0
Region              India (in)
Latency            116ms
Web Interface     http://127.0.0.1:4040
Forwarding          https://c852-37-111-243-195.in.ngrok.io-> http://localhost:8000

Connections
  ttl     opn      rtt1     rt5      p50      p90
    0       0      0.00     0.00     0.00     0.00
```

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

Live Webhooks

URL

Secret hash What is secret hash?

How to create & setup Flutterwave information properly?

- Go to the [Paystack official website](#) & login If you aren't a registered user, Complete the Flutterwave registration first.
- After logged in into Flutterwave dashboard, Click on Settings.



- After That Click on API Keys for Paystack Secret Key & Paystack Public Key.



- Copy & paste those inside PickBazar API and PickBazar Shop .env file like mentioned above.

Special Notes for Flutterwave.

If we have used any third party system/plugin/packages, then we have always encouraged our respected customers to follow the official documentation for detailed & in-depth knowledge.

- Flutterwave official documentation
- Flutterwave Webhooks documentation

Integration of New Payment Gateway

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

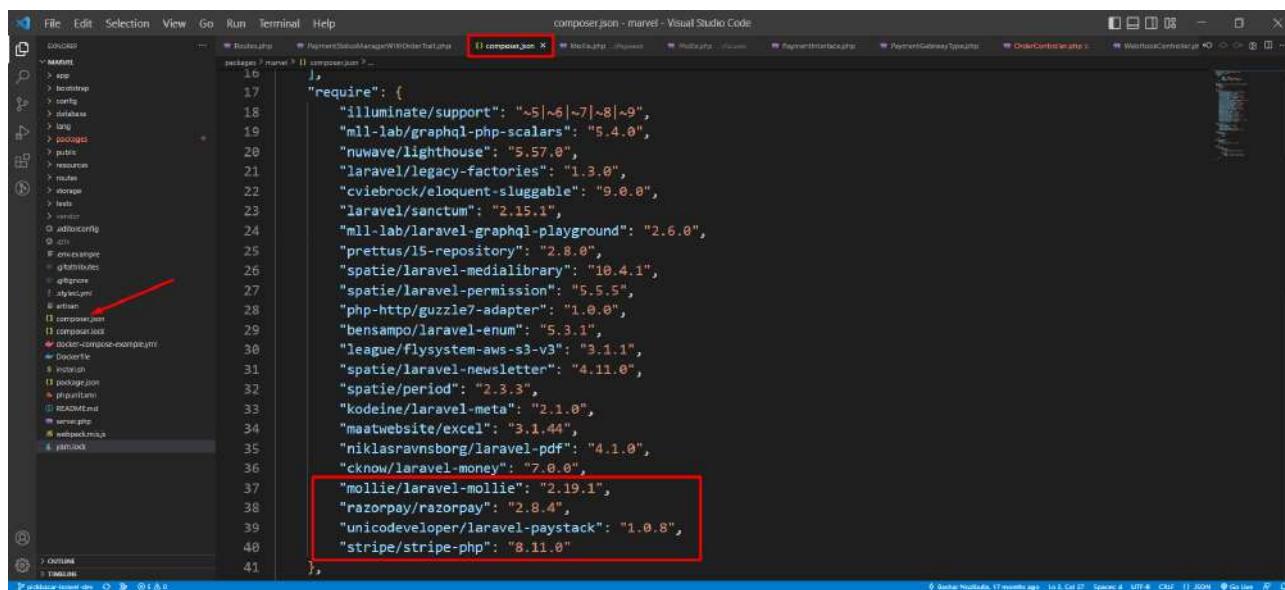
Getting Started with API

Step 1: Install and configure the payment gateway package

First, you will need to install the payment gateway package for Laravel (if there are any available package). There are several packages available that provide integration with various payment gateways. Example is given in the screenshot.

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

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



A screenshot of the Visual Studio Code interface. The left sidebar shows a project structure with a 'composer.json' file selected. The main area is a code editor with the following content:

```
{
    "name": "marvel/PickBazar-Laravel",
    "description": "A Laravel application for a grocery delivery service called PickBazar.",
    "version": "1.0.0",
    "require": {
        "illuminate/support": "~5|~6|~7|~8|~9",
        "mll-lab/graphql-php-scalars": "5.4.0",
        "nuwave/lighthouse": "5.57.0",
        "laravel/legacy-factories": "1.3.0",
        "cviebroek/eloquent-sluggable": "9.0.0",
        "laravel/sanctum": "2.15.1",
        "mll-lab/laravel-graphql-playground": "2.6.0",
        "prettus/l5-repository": "2.8.0",
        "spatie/laravel-medialibrary": "10.4.1",
        "spatie/laravel-permission": "5.5.5",
        "php-http/guzzle7-adapter": "1.0.0",
        "bensampo/laravel-enum": "5.3.1",
        "league/flysystem-aws-s3-v3": "3.1.1",
        "spatie/laravel-newsletter": "4.11.0",
        "spatie/period": "2.3.3",
        "kodeine/laravel-meta": "2.1.0",
        "maatwebsite/excel": "3.1.44",
        "niklasrvnsborg/laravel-pdf": "4.1.0",
        "cknow/laravel-money": "7.0.0",
        "mollie/laravel-mollie": "2.19.1",
        "razorpay/razorpay": "2.8.4",
        "unicodeveloper/laravel-paystack": "1.0.5",
        "stripe/stripe-php": "8.11.0"
    },
    "autoload": {
        "psr-4": {
            "App\\": "app/",
            "Database\\Factories\\": "database/factories/",
            "Database\\Migrations\\": "database/migrations/",
            "Database\\Seeders\\": "database/seeders/"
        }
    }
}
```

A red box highlights the 'require' section of the JSON object.

Step 2: Add payment gateway name in the Enum.

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

```

12 final class PaymentGatewayType extends Enum
13 {
14     public const STRIPE = 'STRIPE';
15     public const CASH_ON_DELIVERY = 'CASH_ON_DELIVERY';
16     public const MOLLIE = 'MOLLIE';
17     public const CASH = 'CASH';
18     public const RAZORPAY = 'RAZORPAY';
19     public const FULL_WALLET_PAYMENT = 'FULL_WALLET_PAYMENT';
20     public const PAYPAL = 'PAYPAL';
21 }

```

Step 3: Configure the Payment Facade for the new payment gateway.

Now go to `API -> package -> marvel -> src -> Payment` then create new payment Class (e.g. Stripe, PayPal, Razorpay, Mollie) for implements the `PaymentInterface`.The Class must implements all of the methods defined in the `PaymentInterface`.

The screenshot shows the Visual Studio Code interface with the following details:

- File Path:** Razorpay.php - marvel - Visual Studio Code
- Code Content:**

```

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.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     {

```
- Explorer View:** Shows the project structure under `MARVEL`, specifically the `Payment` folder which contains files like `Base.php`, `Mollie.php`, `Payment.php`, `PaymentInterface.php`, `Payload.php`, `PayloadTrait.php`, `Razorpay.php`, and `Stripe.php`. Arrows point from the code annotations to these files.
- Status Bar:** Shows the file path as `pickme-laravel-dev`, line count as `1 & 0`, and other standard VS Code status indicators.

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             }
94         } catch (Exception $e) {
95             throw new MarvelException(SOMETHING_WENT_WRONG_WITH_PAYMENT);
96         }
97     }
98 }
```

```

93             http_response_code(400);
94             exit();
95         }
96     } catch (SignatureVerificationError $e) {
97         // Invalid signature
98         http_response_code(400);
99         exit();
100    }
101    $eventStatus = (string) $env['of($request->event)']->replace(['payment_type' => $request->event]);
102
103    switch ($eventStatus) {
104        case 'dispute.won':
105        case 'dispute.created':
106        case 'authorized':
107            $this->updatePaymentOrderStatus($request, OrderStatus::PENDING, PaymentStatus::PROCESSING);
108            break;
109        case 'captured':
110            $this->updatePaymentOrderStatus($request, OrderStatus::PROCESSING, PaymentStatus::SUCCESS);
111        case 'failed':
112            $this->updatePaymentOrderStatus($request, OrderStatus::PENDING, PaymentStatus::FAILED);
113    }

```

Note : * It's 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 documentation for your specific payment gateway.

Step 4: Using the 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 like this:

case `GatewayType`:

`function`

```

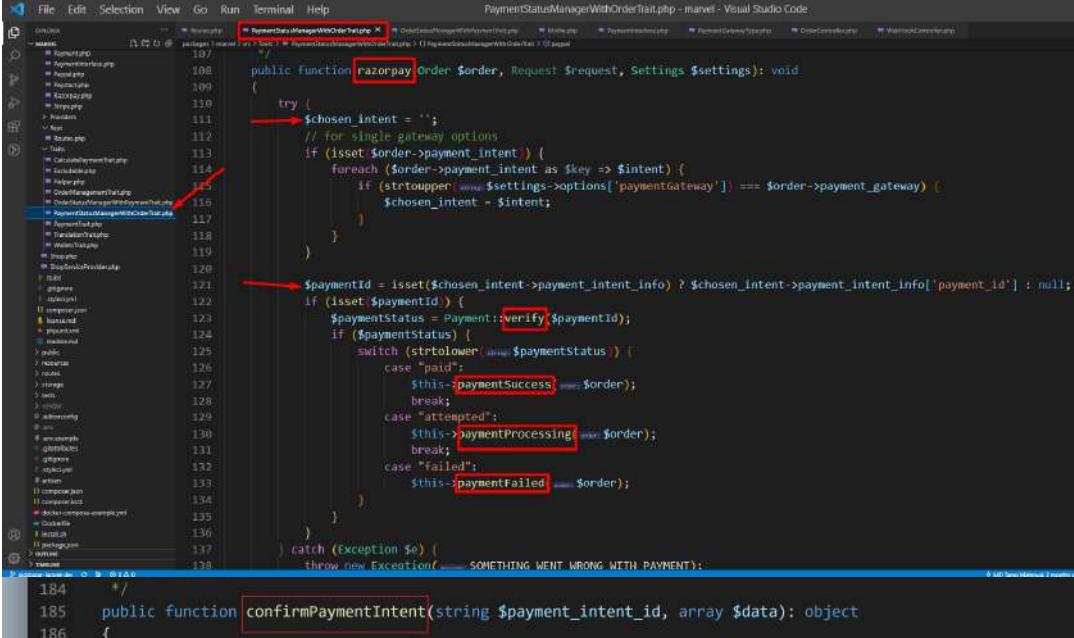
116
117     case 'stripe':
118         http_response_code(200);
119         exit();
120     }
121
122     /**
123      * @throws Exception
124     */
125     public function submitPayment(Request $request): void
126     {
127         $tracking_number = $request->tracking_number ?? null;
128         try {
129             $order = $this->repository->with(['products', 'children.shop', 'wallet_point', 'payment_intent'])
130                 ->findOneByFieldOrFail('tracking_number', $tracking_number);
131
132             switch ($order->payment_gateway) {
133                 case PaymentGatewayType::STRIPE:
134                     $this->stripe($order, $request, $this->settings);
135                     break;
136
137                 case PaymentGatewayType::PAYPAL:
138                     $this->paypal($order, $request, $this->settings);
139                     break;
140
141                 case PaymentGatewayType::MOLLIE:
142                     $this->mollie($order, $request, $this->settings);
143                     break;
144
145                 case PaymentGatewayType::RAZORPAY:
146                     $this->razorpay($order, $request, $this->settings);
147                     break;
148             }
149         }
150     }

```

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.

`public function attachPaymentMethodToCustomer(string $retrieved_payment_method, object $request): object`



```

151     * attachPaymentMethodToCustomer
152     * @param string $retrieved_payment_method
153     * @param object $request
154     * @return object
155     */
156     public function attachPaymentMethodToCustomer(string $retrieved_payment_method, object $request): object
157     {
158         /**
159          * @throws PaymentStatusManagerWithOrderTraitException
160          */
161         $this->razorpay($order, $request, $this->settings);
162
163         if (isset($chosen_intent)) {
164             $chosen_intent = '';
165             // for single gateway options
166             if (isset($order->payment_intent)) {
167                 foreach ($order->payment_intent as $key => $intent) {
168                     if (striptoupper($settings->options['paymentgateway']) === $order->payment_gateway) {
169                         $chosen_intent = $intent;
170                     }
171                 }
172             }
173
174             $paymentId = isset($chosen_intent->payment_intent_info) ? $chosen_intent->payment_intent_info['payment_id'] : null;
175             if (isset($paymentId)) {
176                 $paymentstatus = Payment::verify($paymentId);
177                 if ($paymentstatus) {
178                     switch (striptoupper($paymentstatus)) {
179                         case "paid":
180                             $this->paymentSuccess($order);
181                             break;
182                         case "attempted":
183                             $this->paymentProcessing($order);
184                             break;
185                         case "failed":
186                             $this->paymentFailed($order);
187                     }
188                 }
189             }
190         }
191     }
192
193     /**
194      * @throws PaymentStatusManagerWithOrderTraitException
195     */
196     public function confirmPaymentIntent(string $payment_intent_id, array $data): object
197     {

```

Step 5: How to Setup webhook in pickbazar-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.

```
187     return (object) [];
188 }
189 /**
190 * setIntent
191 */
192 /**
193 * @param array $data
194 * @return array
195 */
196 public function setIntent(array $data): array
197 {
198     return [];
199 }
200 /**
201 */
202
203 /**
204 * @param string $method_key
205 * @return object
206 */
207 Route::post('webhooks/razorpay', [WebHookController::class, 'razorpay']);
208 Route::post('webhooks/stripe', [WebHookController::class, 'stripe']);
209 Route::post('webhooks/paypal', [WebHookController::class, 'paypal']);
210 Route::post('webhooks/mollie', [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     }
}
```

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                 $this->apiUtility->verifyWebhookSignature($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) $request->of($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    }
```

* 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.

```
77   > = {
78     STRIPE: {
79       name: 'Stripe',
80       value: PaymentGateway.STRIPE,
81       icon: '/payment/stripe.png',
82       component: PaymentOnline,
83     },
84     PAYPAL: {
85       name: 'Paypal',
86       value: PaymentGateway.PAYPAL,
87       icon: '',
88       // icon: '/payment/paypal.png',
89       component: PaymentOnline,
90     },
91     RAZORPAY: {
92       name: 'RazorPay',
93       value: PaymentGateway.RAZORPAY,
94       icon: '/payment/razorpay.png',
95       component: PaymentOnline,
96     },
97     MOLLIE: {
98       name: 'Mollie',
99       value: PaymentGateway.MOLLIE,
100      icon: '/payment/mollie.png',
101      component: PaymentOnline,
102    },
103  },
```

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.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "PICKBAZAR-SHOP". The "payment" folder contains "razorpay" and "stripe" subfolders. "razorpay" contains "razorpay-payment-modal.tsx". "stripe" contains "stripe-base-form.tsx", "stripe-payment-form.tsx" (which is selected), "stripe-payment-modal.tsx", "stripe-saved-cards-list.tsx", "add-new-payment-modal.tsx", "pay-now-button.tsx", "payment-modal.tsx", and "saved-card-view-header.tsx".
- Code Editor:** Displays the "stripe-payment-form.tsx" file. The code defines an interface "Props" and a functional component "PaymentForm".
- Status Bar:** Shows developer information: MD Tareq Mahmud, 2 months ago, Ln 1, Col 1, Spaces: 2, UTF-8, CRLF, TypeScript React, Go Live, and other icons.

```
16 interface Props {  
17     paymentIntentInfo: PaymentIntentInfo;  
18     trackingNumber: string;  
19     paymentGateway: PaymentGateway;  
20 }  
21  
22 const PaymentForm: React.FC<Props> = ({  
23     paymentIntentInfo,  
24     trackingNumber,  
25     paymentGateway,  
26 }) => {  
27     const { t } = useTranslation('common');  
28     const stripe = useStripe();  
29     const elements = useElements();
```

That's all for today. If you need any more help you can always contact with our support agents in our support portal.

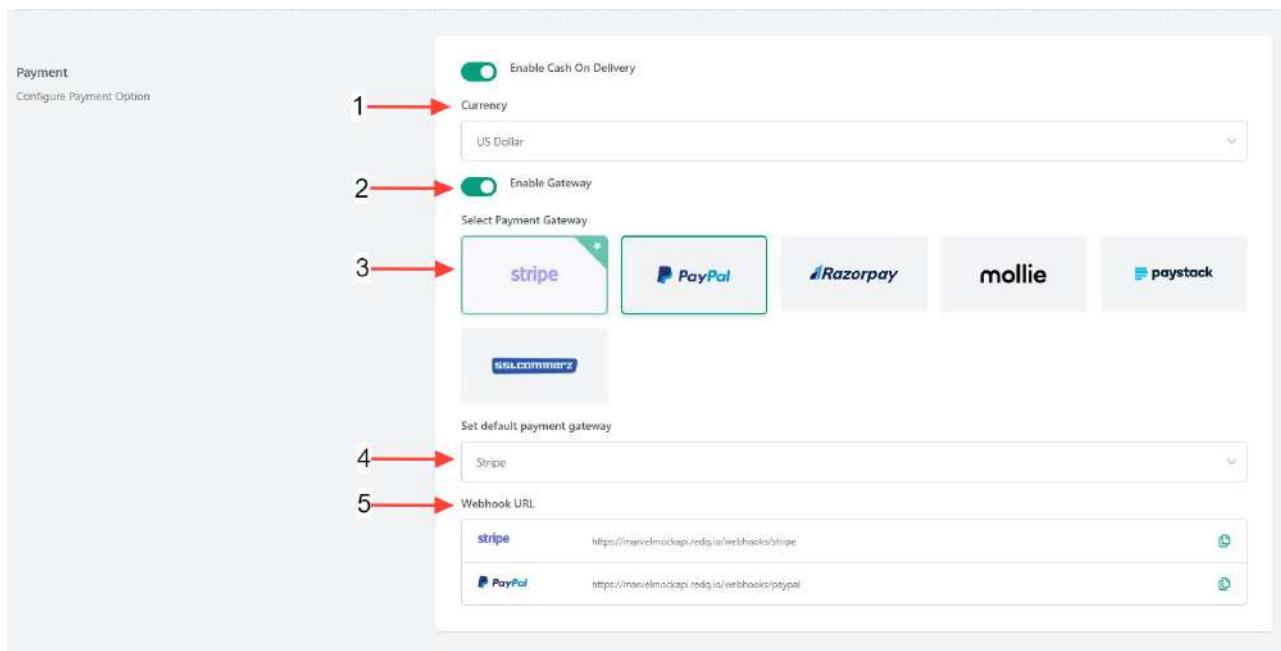
Multiple payment gateway

In this chapter we are going to discuss about how you can use multiple payment gateway in our application. But before jumping into this feature user needs to keep some points in mind first.

1. Choose your desired payment gateway based on availability in your region.
2. Choose your currency correctly before activating that payment gateway.
3. Set up all the necessary .env keys and payment gateway internal settings properly before starting operations.
4. Configure webhooks properly if needed.

How to activate multiple payment gateways [in Admin] ?

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



Description [sequence by number in the screenshot]

1. Choose the currency first
2. Enable gateway if you want to use any
3. Select the payment gateway based on your region and supported currency.
4. Select default payment gateway. Default gateway only available from selected ones.
5. Webhooks URL will be generated. You can use this URL for webhooks setting in payment gateways.

How to use multiple payment gateways from customer end [in Shop] ?

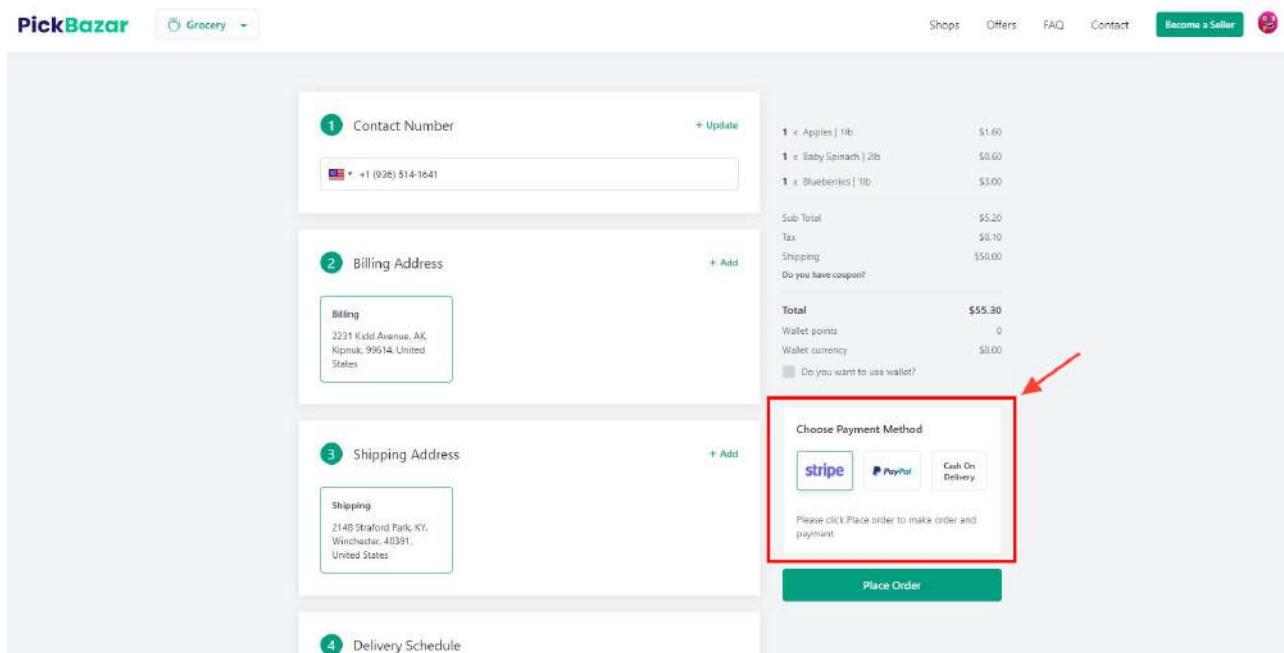
Payment gateways activated by super-admin will be available in the checkout process. There are two steps of operation,

1. Select a payment gateway during place order
2. Select different payment gateway (if needed), after the place order.

This two points are cleared below with necessary screenshot.

1. Select a payment gateway during place order

Please check this screenshot first,



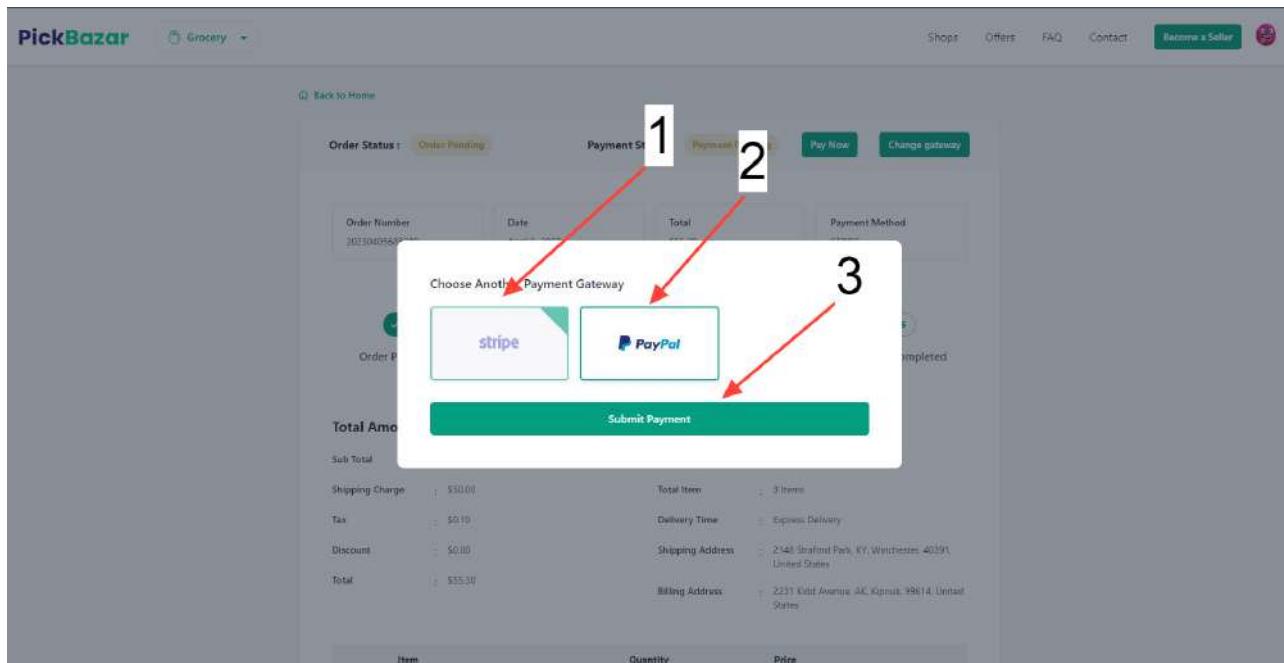
Select a payment gateway just before placing order.

** 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.

Introduction to Flash Sale

In the modern days of e-commerce, businesses are constantly innovating to capture the attention of online shoppers. One such step that has taken the online retail space by storm is the "Flash Sale." A flash sale is a marketing strategy that creates a sense of urgency and excitement by offering limited-time discounts or promotions on products.

We have introduced this feature on PickBazar v11.0.0. So if you want to use `Flash Sale` with `PickBazar`, then make sure your PickBazar is v11.0.0 or later.

We have plan to enrich this feature area by integrating more functionalites in future. Before go through lets denotes the key points. We have divided this feature into 3 different areas.

- **Super Admin**
- **Vendors & Staffs**
- **Customers**

Let's discuss those sequentially in upcoming chapters.

Super Admin to Flash Sale

Super admin is a role who has the authority to run a complete **CRUD** (Create, Read, Update, Delete) operation on this feature. But there are some key concepts we are considering to build a flash sale data.

Key concept

There some key concepts need to understand first before proceed on. Flash sale feature's data building is based on this procedure.

- Products which have already **sale price**, those **are not eligible** for any Flash sale campaign.
- Products which are **already in a running flash sale campaign**, those **are not eligible** to chose for another campaign. When the current campaign is over then those products will be free to use in another campaign.
- Super admin can remove any products from on going campaign, then those products will be available to assign for any campaign.
- Super admin can choose products independently.
- Also, super admin can filter out products via Group & Categories for choosing on a campaign.

Here is a sample of **flash sale form** for creating any campaign.

Main

- Dashboard

Shop Management

- Shops
- My Shops

Product Management

- Products
- Inventory
- Categories
- Tags
- Attributes
- Manufacturers/Publications
- Authors

E-commerce Management

- Taxes
- Shipments
- Withdrawals
- Refunds

Layout/Page Control

- Home pages / Types
- FAQs
- Terms And Conditions

Order Management

- Orders
- Create Order
- Transactions

User Control

- All users
- Admin list
- Vendors
- Staff
- Customers

Feedback Control

Create flash sale schedule

Flash sale thumbnail:
This thumbnail will be used for flash sale grid.
Recommended dimension of the image : 520 x 288px

Cover Image
Set cover image
Dimension of the cover image should be 1920 x 700px

Description
Add campaign here.

Title*

Description*

Please select dates in which no other flash sale is on going.

Active Date*

Expire Date*

Select which type of offering is applicable in this campaign *

Fixed rate

Percentage

Choose products
Select all the products you want to set for this flash sale campaign. You can see details from the campaign list

Select products filter option.*

Handpicked products

Filter products by group & related Category

Add Campaign

©2023 Pickbazar. Howdy External

10.2.0

In our application, we have denoted each flash sale period as a campaign. Each campaign has two different sets of choices

- Percentage basis campaign
- Fixed rate basis campaign

Let's discuss about those & see how can we build one.

Percentage basis

Here is a construction for a flash sale on percentage basis.

MAIN

[Dashboard](#)

SHOP MANAGEMENT

[Shops](#)[My Shops](#)

PRODUCT MANAGEMENT

[Products](#)[Inventory](#)[Categories](#)[Tags](#)[Attributes](#)[Manufacturers/Publications](#)[Authors](#)

E-COMMERCE MANAGEMENT

[Taxes](#)[Shipments](#)[Withdrawals](#)[Refunds](#)

Edit campaign : 15% Flash Sale: Don't Miss Out on Super Savings!

Flash sale thumbnail:

This thumbnail will be used for flash sale grid.
Recommended dimension of the image : 520 x 288px



Upload an image or drag and drop
PNG, JPG

Cover Image

Set cover image
Dimension of the cover image should be 1920 x 700px



Upload an image or drag and drop
PNG, JPG

Description

Edit campaign here

Title*

15% Flash Sale: Don't Miss Out on Super Savings!

Description*

Ready to ignite your shopping spree? Don't miss our electrifying "Sizzle into Savings" Flash Sale! Get a dazzling 15% off on your favorite products for a limited time only!

Dive into a world of savings as you explore our vast selection of top-notch merchandise. From fashion-forward clothing to cutting-edge electronics, we have something for everyone.

Please select dates in which no other flash sale is on going.

Active Date*

20/09/2023

Expire Date*

22/09/2023

Select which type of offering is applicable in this campaign *

Fixed rate

Percentage

Amount applicable for this campaign (USD)

15

Enable flash deals

Choose products

Select all the products you want to set for this flash sale campaign. You can see details from the campaign list.

Select products filter option.*

Handpicked products

Filter products by group & related Category

Filter By Group

Filter by Group

Filter By Category

Filter by Category

Choose Products*

Estee Lauder Double Wear Stay-in-Place Foundation - USD 40.00 Blueberries - USD 3.00 Mango - USD 2.50 Cucumber - USD 2.50
 Clinique Everlasting Compact Foundation - USD 54.00 Dior Backstage Contour Palette - USD 44.00
 Dr. Hauschka Compact Powder - USD 18.00 L'Oréal Face Powder - USD 15.00

Back

Update Campaign

FEATURE MANAGEMENT

[Message](#)[Store Notice](#)

SITE MANAGEMENT

[Settings](#)

©2023 Pickbazar. Howdy External

10.2.0

In percentage basis, product pricing will be calculated based on the current regular price. Suppose if any product's regular price is \$150, then if it's on a campaign with 15% rate, then the sale price will be \$127.5

Fixed rate basis

Here is a construction for fixed rate basis flash sale.

The screenshot shows the 'Create flash sale schedule' page in the PickBazar admin interface. The left sidebar contains navigation links for Main, Shop Management, Product Management, E-commerce Management, Layout/Theme Control, Order Management, User Control, Feedback Control, Promotional Management, Feature Management, and Site Management. The main content area is titled 'Create flash sale schedule' and includes the following sections:

- Flash sale thumbnail:** A placeholder for a 520x288px image, with a note that it will be used for flash sale grids. An 'Upload an image or drag and drop' button is provided.
- Cover Image:** A placeholder for a 1920x700px image, with a note that the dimension of the cover image should be 1920x700px. An 'Upload an image or drag and drop' button is provided.
- Description:** A form section with:
 - Title***: Fixed Rate : \$15 Flash Sale Extravaganza!
 - Description***: A rich text area containing promotional text: "Our wait is over! This incredible deal won't last long... The clock is ticking, and once the time is up, so is your chance to score these incredible savings. Join us at [Store Name] and experience the thrill of grabbing your favorite items for a mere \$15. Don't miss out - mark your calendar, set your alarms, and be there when the '\$15 Flash Sale Extravaganza' begins!"
 - Please select dates in which no other flash sale is on going.**
 - Active Date***: 26/09/2023
 - Expire Date***: 29/09/2023
- Select which type of offering is applicable in this campaign ***:
 - Fixed rate
 - Percentage
- Amount applicable for this campaign (USD)**: 15
- Enable flash deals**: A toggle switch that is turned off.
- Please select those products only, which prices will not compromise with this campaign rate**: A note for selecting products.
- Choose products**: A section for selecting products from a list. It includes:
 - Select products filter option.***:
 - Handpicked products
 - Filter products by group & related Category
 - Filter By Group**: A dropdown menu showing 'Bags'.
 - Filter By Category**: A dropdown menu showing 'Purse'.
 - Choose Products**: A list of selected products: Estee Lauder - USD 200, Gucci Handbag - USD 380, Armani Handbag - USD 300, See by Chloé - USD 780, Salvatore Ferragamo - USD 100, Gucci Purse - USD 90, Armani Silver Purse - USD 120, KAAI Pyramid Bag - USD 60.
- Add Campaign**: A green button at the bottom right.

In fixed rate basis, product pricing will be calculated based on the current regular price. Suppose if any product's regular price is \$150, then if it's on a campaign with **fixed rate of \$30**, then the sale price will be \$120

Note : If your selected product's regular price is less than the campaign rate, then the sale price will not be calculated & customers will see the regular price during purchase.

All listed flash sale

Super admin can run CRUD operation from this panel for flash sale. All the listed campaigns are presented here with pagination.

If admin click on eye icon in the **Details** column, then it will redirect to a single page where that flash sale information will be shown in details.

ID	Title	Description	Start date	End date	Details	Actions
#ID: 2	\$15 Flash Sale Extravaganza	Get ready for the shopping event of the season! For a limited time...	9/26/2023	9/28/2023		
#ID: 1	15% Flash Sale: Don't Miss Out!	Ready to ignite your shopping spree? Don't miss our electrifying "Si...	9/20/2023	9/22/2023		

Flash sale single view in admin panel.

In campaign details page, it was divided into two sections.

- **Information section**

In this section, all the campaign related information are displayed.

- **Product table section**

In this section, all the selected products are displayed in a tabluar format. Data was fetched with pagination.

Please check this screenshot below.

Flash Sale

15% Flash Sale: Don't Miss Out on Super Savings!

Ready to Ignite your shopping spree? Don't miss our electrifying "Sizzle into Savings" Flash Sale! Get a dazzling 15% off on your favorite products for a limited time only! Dive into a world of savings as you explore our vast selection of top-notch merchandise. From fashion-forward clothing to cutting-edge electronics, we have something for everyone. Why wait? The clock is ticking! Grab your shopping carts and rush in to seize this incredible deal before it's gone. Our shelves are stocked, our staff is ready, and the discounts are smoking hot! Hurry, this sale is here today, gone tomorrow. Don't let the chance to save big slip through your fingers. Join us now and let the shopping spree begin! Terms and conditions apply. Offer valid for a limited time. See in-store for details.

Campaign status : On going
Active Date: 20 Sep 2023
Expire Date: 22 Sep 2023
Campaign type on : Percentage
Deals rate : 15% off.

Deals						
Image	SKU	Product Type	Price/Unit	Deal offering	Quantity	Status
	2	Simple	\$0.60	\$0.51	10	<button>Publish</button>
	3	Simple	\$3.00	\$2.55	30	<button>Publish</button>
	8	Simple	\$2.50	\$2.13	25	<button>Publish</button>
	13	Simple	\$2.50	\$2.13	50	<button>Publish</button>
	1004	Simple	\$44.00	\$37.40	45	<button>Publish</button>

©2023 Pickbazar. Howdy External 10.2.0

Flash Sale for Vendors and Staffs.

For shop owner, flash sale is very important for feature to engage with more customer. In our application Flash sale feature is working following this procedure from Vendor (shop owner) & Staff (shop staff) perspective.

- Vendor & Staff both role has ability to checkout all the **on going** & **upcoming** campaigns. So that they can prepare their strategies for campaigns, manage inventories etc.
- Both can check which products of their shops are selected for the campaigns.

This screenshots will helps to understand the feature better.

Listed campaigns for Vendor & Staff

PickBazar

Refunds

Order Management

Orders

Transactions

Features Management

Store Notice

Message

Feedback Control

Reviews

Questions

User Control

Staff

Promotional Control

Flesh sale

Available flash deals

My products in deals

©2023 Pickbazar. All right reserved RedQ

10.2.0

Currently on going flash sales.

ID	Title	Description	Start date	End date	Type	Rate	Details
2	\$15 Flash Sale Extravag...	Get ready for the shopping event of the season! For a limited time only, our store is hosting an electrifying "\$15 Flash Sale Extravaganza" where you can snag a fantastic array of products for just \$15 each. Yes, you read that right – only \$15! From fashion-forward clothing and accessories to must-have gadgets and home essentials, we've curated a selection of items that will leave you amazed. But hurry! This incredible deal won't last long. The clock is ticking, and once the time is up, so is your chance to score these incredible savings. Join us at [Store Name] and experience the thrill of grabbing your favorite items for a mere \$15. Don't miss out – mark your calendar, set your alarms, and be there when the "\$15 Flash Sale Extravaganza" begins!	9/26/2023	9/29/2023	fixed_rate	15	
1	15% Flash Sale! Don't Mi...	Ready to ignite your shopping spree? Don't miss our electrifying "15% Flash Sale!" We're offering a massive discount of 15% off sitewide. Whether you're looking for clothing, accessories, or home essentials, this deal has something for everyone. Hurry, this offer is limited time only! Don't let this opportunity pass you by – shop now and save big!	9/26/2023	9/22/2023	percentage	15	

Visit Site

Language English

Store Owner Store owner

Campaign details for Vendor & Staff

PickBazar

Dashboard

Products

Inventory

Attributes

Manufacturers/Publications

Authors

Withdrawals

Refunds

Orders

Transactions

Store Notice

Message

Reviews

Questions

Active Date: 26 Sep 2023

Expire Date: 29 Sep 2023

Campaign type on: Fixed Rate

Deals rate: 15

©2023 Pickbazar. All right reserved RedQ

10.2.0

Flash Sale

\$15 Flash Sale Extravaganza!

Get ready for the shopping event of the season! For a limited time only, our store is hosting an electrifying "\$15 Flash Sale Extravaganza" where you can snag a fantastic array of products for just \$15 each. Yes, you read that right – only \$15! From fashion-forward clothing and accessories to must-have gadgets and home essentials, we've curated a selection of items that will leave you amazed. But hurry! This incredible deal won't last long. The clock is ticking, and once the time is up, so is your chance to score these incredible savings. Join us at [Store Name] and experience the thrill of grabbing your favorite items for a mere \$15. Don't miss out – mark your calendar, set your alarms, and be there when the "\$15 Flash Sale Extravaganza" begins!

Visit Site

Language English

Store Owner Store owner

Listed products of any shop.

The screenshot shows the PickBazar application interface. On the left, there's a sidebar with various management sections like Order Management, Features Management, Feedback Control, User Control, Promotional Control, and Layout/Page Management. Under Promotional Control, a dropdown menu is open for 'Flash sale' with options 'Available flash deals' and 'My products in deals'. The main content area has a heading 'My products on going flash sales.' and a search bar. A table lists four products: Mango, Cucumber, Blueberries, and Baby Spinach, each with a 'Publish' button. At the bottom right of the main area, there are navigation arrows. The footer includes a 'FAQs' link and copyright information: '©2023 Pickbazar. All rights reserved RedQ' and '10.2.0'.

Product	Slug	SKU	Product Type	Regular price	Quantity	Sold Quantity	Status
Mango	mango	13	Simple	\$2.50	50	0	Publish
Cucumber	cucumber	8	Simple	\$2.50	25	0	Publish
Blueberries	blueberries	3	Simple	\$3.00	30	0	Publish
Baby Spinach	baby-spinach	2	Simple	\$0.60	10	0	Publish

Flash Sale for Customer

For any customer flash sale is an opportunity to collect his/her desired products with a discounted price. In our application, there are some ways to engage a customer with flash sale feature.

- There is page where all the **on going** & **upcoming** flash sales are listed. So customers can checkout the products, prepare their plans before the campaign.

Flash Sale lists

Available Flash Sale

[Home](#) > Available Flash Sale



Limited time discounts.

Offer Till: 16 Oct 2023 - 29 Nov 2023

Campaign status : On going

Campaign type on : Fixed Rate

Deals rate : 10



Super Savings Spectacular: Discounts Await You!

Offer Till: 20 Oct 2023 - 30 Oct 2023

Campaign status : On going

Campaign type on : Percentage

Deals rate : 50



Limited-Time Offer: Act Fast! 🔥

Offer Till: 31 Oct 2023 - 29 Nov 2023

Campaign status : On going

Campaign type on : Percentage

Deals rate : 50

Flash sale details



Sale Starts In: 6d 8h 38m 55s

Fixed Rate : \$15 Flash Sale!

🌟 Get ready for the shopping event of the season! 🌟 For a limited time only, our store is hosting an electrifying "\$15 Flash Sale Extravaganza" where you can snag a fantastic array of products for just \$15 each. Yes, you read that right – only \$15! 🎉 From fashion-forward clothing and accessories to must-have gadgets and home essentials, we've curated a selection of items that will leave you amazed. ⏰ But hurry! This incredible deal won't last long. The clock is ticking, and once the time is up, so is your chance to score these incredible savings. Join us at [Store Name] and experience the thrill of grabbing your favorite items for a mere \$15. Don't miss out – mark your calendar, set your alarms, and be there when the "\$15 Flash Sale Extravaganza" begins!

Campaign status: On hold

Offer Till: 26 Sep 2023 - 29 Sep 2023

Campaign type on : Fixed Rate

Deals rate : 15

Gucci Purse
\$90.00
★ 5 (1) Reviews

Armani Silver Purse
\$120.00
★ 5 (1) Reviews

KAAI Pyramid Bag
\$60.00
★ 0 (0) Reviews

Balenciaga
\$200.00
★ 0 (0) Reviews

Gucci Handbag
\$380.00
★ 0 (0) Reviews

See by Chloé
\$280.00
★ 0 (0) Reviews

Armani Handbag
\$300.00
★ 0 (0) Reviews

Salvatore Ferragamo
\$200.00
★ 0 (0) Reviews

0 Items
\$0.00

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.**

A screenshot of the "APIs & Services" page in the Google Cloud Console. On the left, there's a sidebar with several options: "Dashboard", "Library", "Credentials", "OAuth consent screen" (which is highlighted with a red arrow), "Domain verification", and "Page usage agreements". The main content area is titled "OAuth consent screen" and contains a message: "because your OAuth request includes additional scopes that haven't been approved." Below this is a progress bar showing "0 users / 100 user cap". There's also a "SHOW LESS" link. The next section is titled "OAuth rate limits" with a sub-section for "Your token grant rate". It shows a current limit of "10,000 grants per day" and a "Token grant rates limit how quickly your application can get new users." A "Raise daily token limit" link is provided. Below this is a chart showing token grants over time, with a note "No data is available for the selected time frame." A red arrow points from the text "After creating the project, go to APIs & Services" in the previous step to the "OAuth consent screen" option in the sidebar.

4. Create an **External OAuth Consent**. Add all the required fields.

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

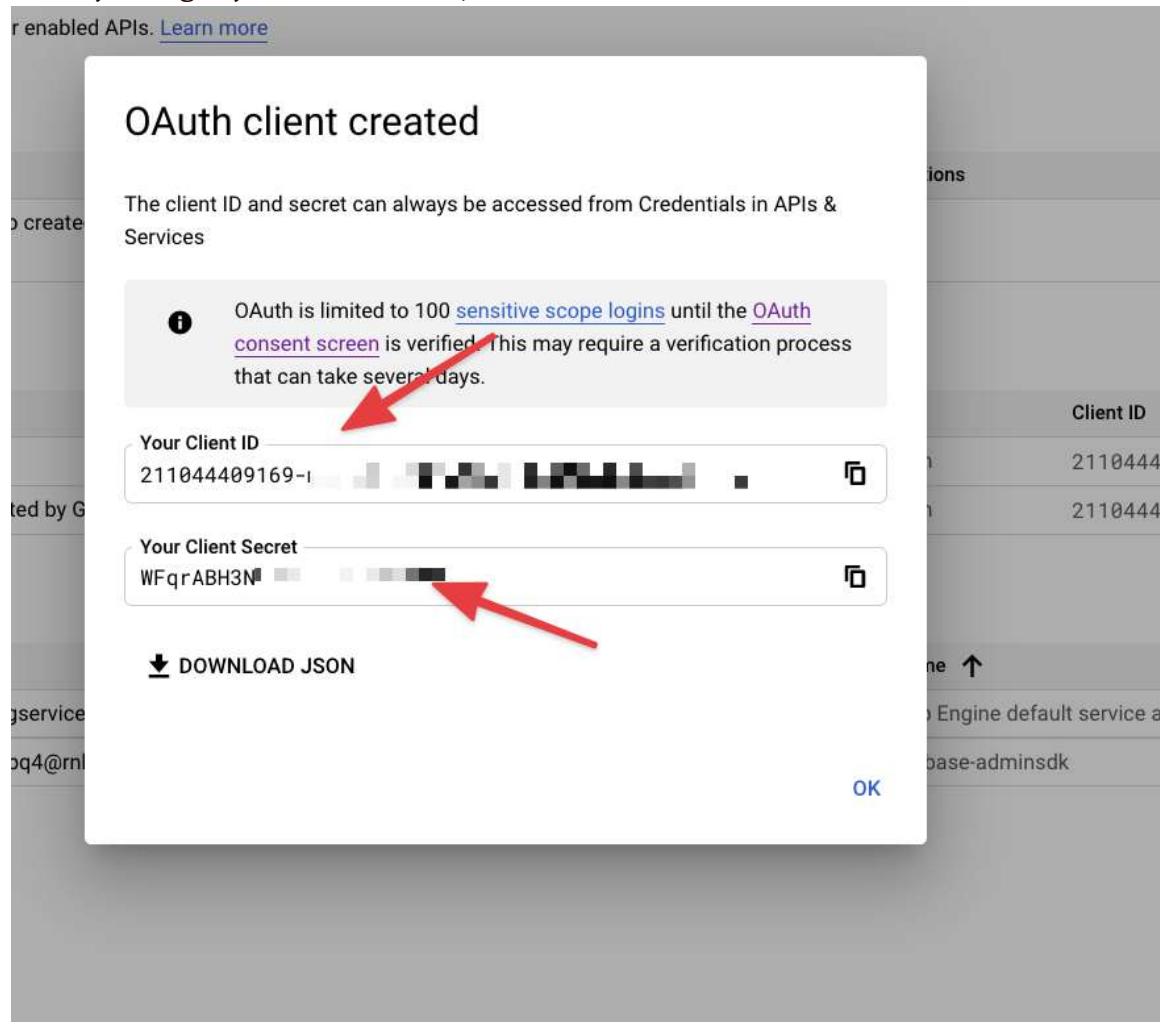
The screenshot shows the 'Credentials' section of the Google Cloud Platform API & Services page. A red arrow points to the 'OAuth client ID' option under 'API Keys'. The 'OAuth client ID' row is highlighted, showing 'Name: AlzabyAinp...', 'Restrictions: None', and 'Key: hHhc-1ots'. A tooltip for 'Help me choose' provides information about the different credential types.

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

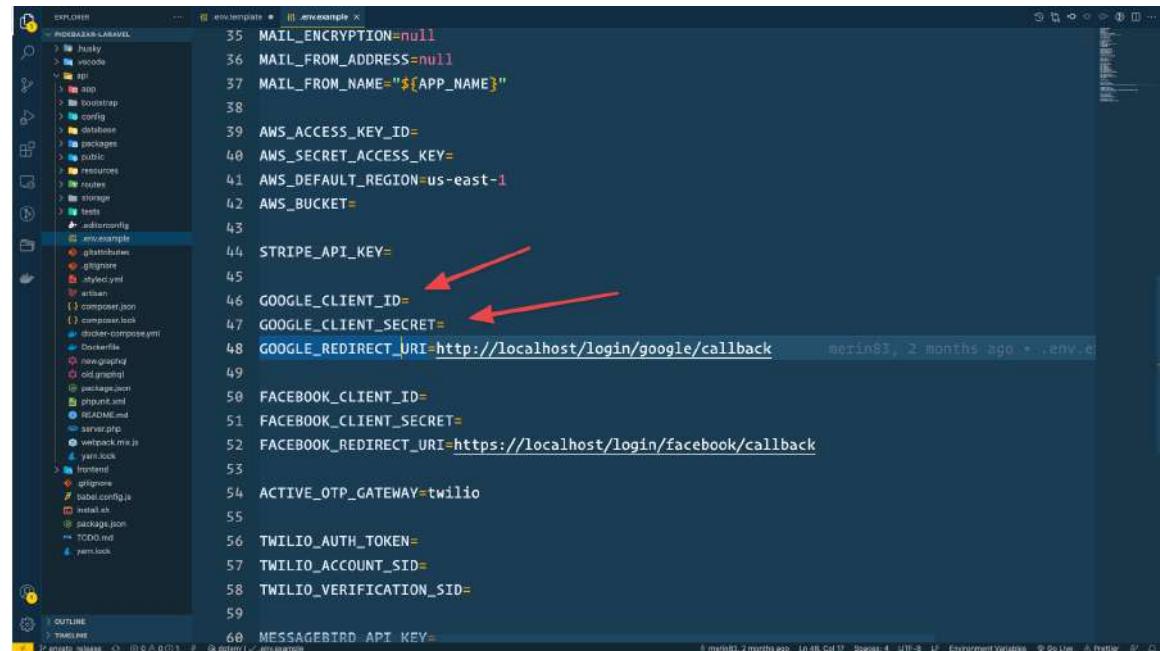
`http://yourdomain.com/api/auth/callback/google`

The screenshot shows the 'Client ID for Web application' configuration page. The 'Name' field is set to 'Web client 1'. A note explains that this name identifies the client in the console. Below this, a box contains a note about adding authorized domains to the OAuth consent screen. The 'Authorized JavaScript origins' section is shown with a note for browser requests and a '+ ADD URI' button. The 'Authorized redirect URIs' section is shown with a note for web server requests and a red box highlighting the 'URIs 1' field containing 'https://yourdomain.com/api/auth/callback/google'. A '+ ADD URI' button is also present here.

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

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

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

For REST API

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

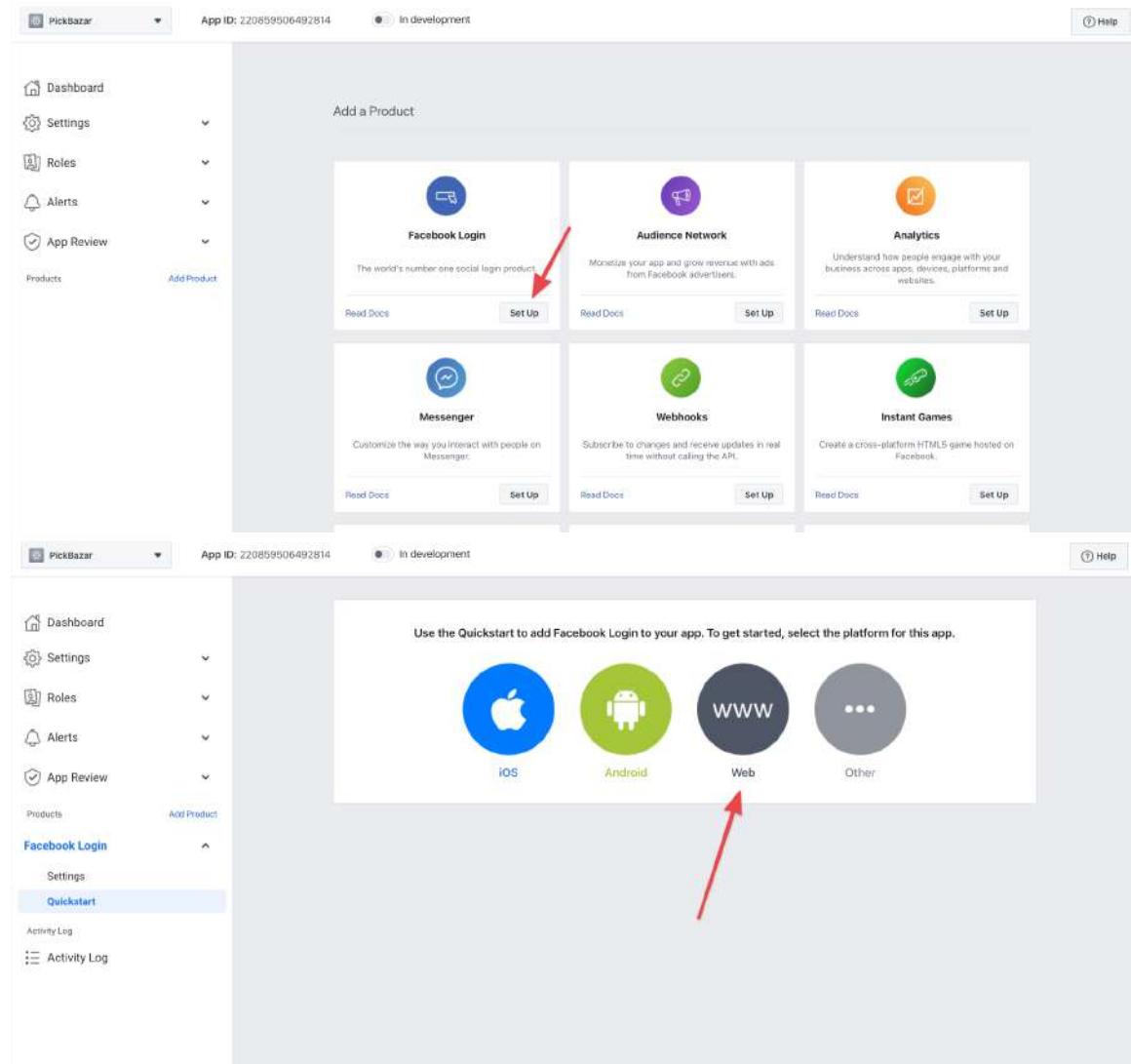
And for GraphQL,

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

Facebook

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

3. After that, set up Facebook Login.



The screenshot shows the Facebook Developer Platform interface. The top navigation bar includes 'PickBazar' (App ID: 220859506492814), 'In development', and a 'Help' link. On the left, a sidebar lists 'Dashboard', 'Settings', 'Roles', 'Alerts', 'App Review', 'Products' (selected), and 'Add Product'. The main content area is titled 'Add a Product' and features several cards:

- Facebook Login**: 'The world's number one social login product.' A red arrow points to the 'Set Up' button.
- Audience Network**: 'Monetize your app and grow revenue with ads from Facebook advertisers.'
- Analytics**: 'Understand how people engage with your business across apps, devices, platforms and websites.'
- Messenger**: 'Customize the way you interact with people on Messenger.'
- Webhooks**: 'Subscribe to changes and receive updates in real time without calling the API.'
- Instant Games**: 'Create a cross-platform HTML5 game hosted on Facebook.'

The bottom section is titled 'Use the Quickstart to add Facebook Login to your app. To get started, select the platform for this app.' It shows four icons: iOS (blue circle with white Apple logo), Android (green circle with white Android logo), Web (dark blue circle with white 'WWW'), and Other (grey circle with three dots). A red arrow points to the 'Web' icon.

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,

The top part of the image shows the Facebook App Dashboard for 'PickBazar'. It displays the App ID (220969506492814) and App Secret (redacted). Red arrows point to both the App ID and the App Secret field. The bottom part shows a code editor with the .env file open. The FACEBOOK_CLIENT_ID and FACEBOOK_CLIENT_SECRET lines are highlighted with red arrows.

```
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS=null
MAIL_FROM_NAME="${APP_NAME}"
AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
STRIPE_API_KEY=
GOOGLE_CLIENT_ID=
GOOGLE_CLIENT_SECRET=
GOOGLE_REDIRECT_URI=http://localhost/login/google/callback
FACEBOOK_CLIENT_ID=
FACEBOOK_CLIENT_SECRET=
FACEBOOK_REDIRECT_URI=https://localhost/login/facebook/callback
ACTIVE_OTP_GATEWAY=twilio
TWILIO_AUTH_TOKEN=
TWILIO_ACCOUNT_SID=
TWILIO_VERIFICATION_SID=
MESSAGEBIRD_APT_KEY=
NEXTAUTH_URL=YOUR_SHOP_DOMAIN
FACEBOOK_CLIENT_ID=
FACEBOOK_CLIENT_SECRET=
SECRET=
```

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

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

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

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

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

For REST API

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

And for GraphQL,

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

OTP (Mobile Number Verification)

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

With the OTP feature, you can do,

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

Configuration

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

Twilio Configuration:

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

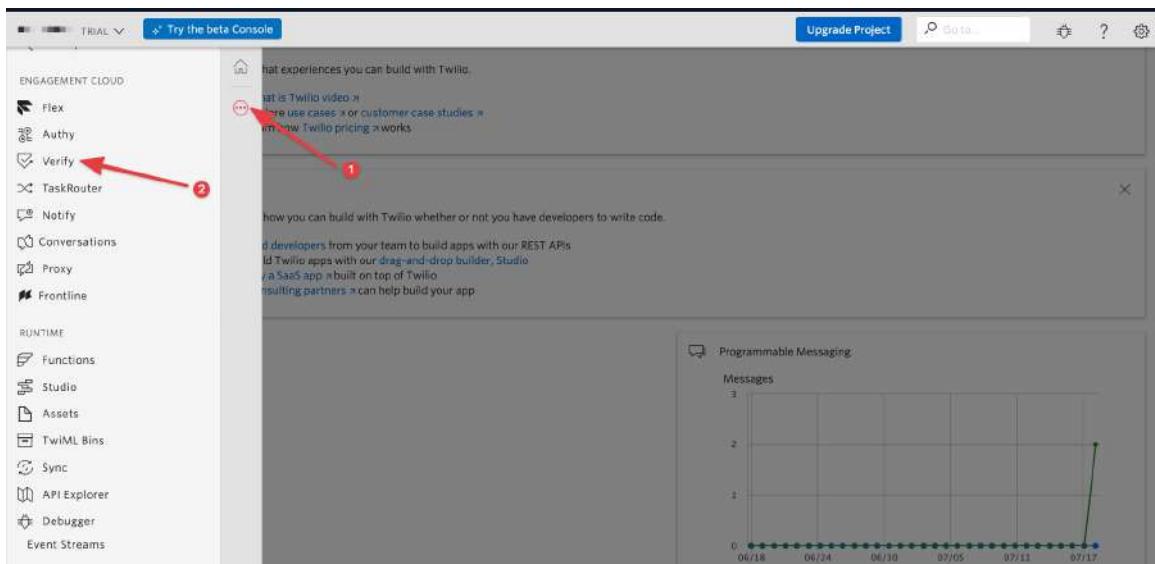
1. Create an account at [Twilio](#)
2. After that, go to the `Twilio Console Panel` and copy the `ACCOUNT SID`, and `AUTH TOKEN` key and add this key to your `api/.env` file

```

57 FACEBOOK_CLIENT_SECRET=
58 FACEBOOK_REDIRECT_URI=https://localhost/login/facebook/callback
59
60 ACTIVE_OTP_GATEWAY=twilio
61
62 TWILIO_AUTH_TOKEN= TWILIO_AUTH_TOKEN
63 TWILIO_ACCOUNT_SID= TWILIO_ACCOUNT_SID
64 TWILIO_VERIFICATION_SID=
65
66 MESSAGEBIRD_API_KEY=
67 MESSAGEBIRD_ORIGINATOR=Pickbazar
68

```

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`

General Settings

Properties

SERVICE NAME: PickBazar

SERVICE SID: VA4b...3d73

CODE LENGTH: 6 digits

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: **ENABLED** Will return line type and carrier information in API response.

LANDLINE VALIDATION: **ENABLED** 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: **DISABLED**

Text-to-speech

```

FACEBOOK_CLIENT_SECRET=
FACEBOOK_REDIRECT_URI=https://localhost/login/facebook/callback
ACTIVE_OTP_GATEWAY=twilio
TWILIO_AUTH_TOKEN=
TWILIO_ACCOUNT_SID=
TWILIO_VERIFICATION_SID= 3
MESSAGEBIRD_API_KEY=
MESSAGEBIRD_ORIGINATOR=Pickbazar

```

Verify Geo Permissions

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

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

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

Verify geographic permissions

*Beta

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

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

1 2 3 4 5 → 26
1-10 of 252 items

Without this verification, no SMS will be delivered.

With that, your configuration is complete.

MessageBird:

Use only one service provider for OTP service, so use this [MessageBird configuration](#) only if you want to use [MessageBird](#) instead of [Twilio](#).

To configure [MessageBird](#), follow this procedure,

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

2. After creating the account, go to the [API Getting Started](#) and follow all the procedures step by step.

The screenshot shows the 'Your getting started guide' section of the MessageBird API Getting Started page. A progress bar at the top indicates 'SMS API • 4 steps left' and '20%'. Below the bar, five steps are listed: 1. Create your account, 2. Get started with free test credits, 3. Install the official SDK for SMS, 4. Send your first SMS, and 5. Go live with SMS. Red arrows numbered 1 through 6 point to each of these steps in sequence from left to right.

3. After completing the profile, copy the [API](#) key from [MessageBird](#) and paste it to [API/.env](#)

The screenshot shows the 'Your getting started guide' and 'Your API Keys' sections of the MessageBird API Getting Started page. The 'Your API Keys' section displays two keys: 'Live' and 'Test'. The 'Live' key is shown as a series of asterisks with a red arrow pointing to it. The 'Test' key is also shown as a series of asterisks. Below the keys, there is a note: 'Use this API Key in your production environment. It works with all of our APIs.' and a link to 'Manage API Keys'.

```

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

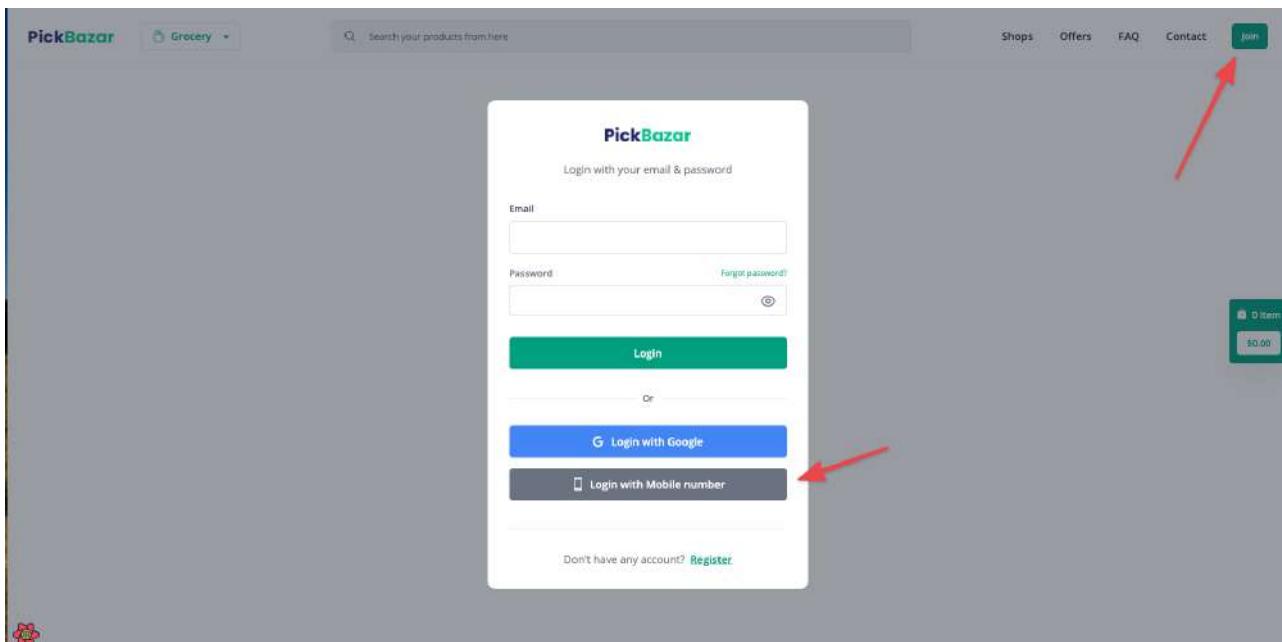
```

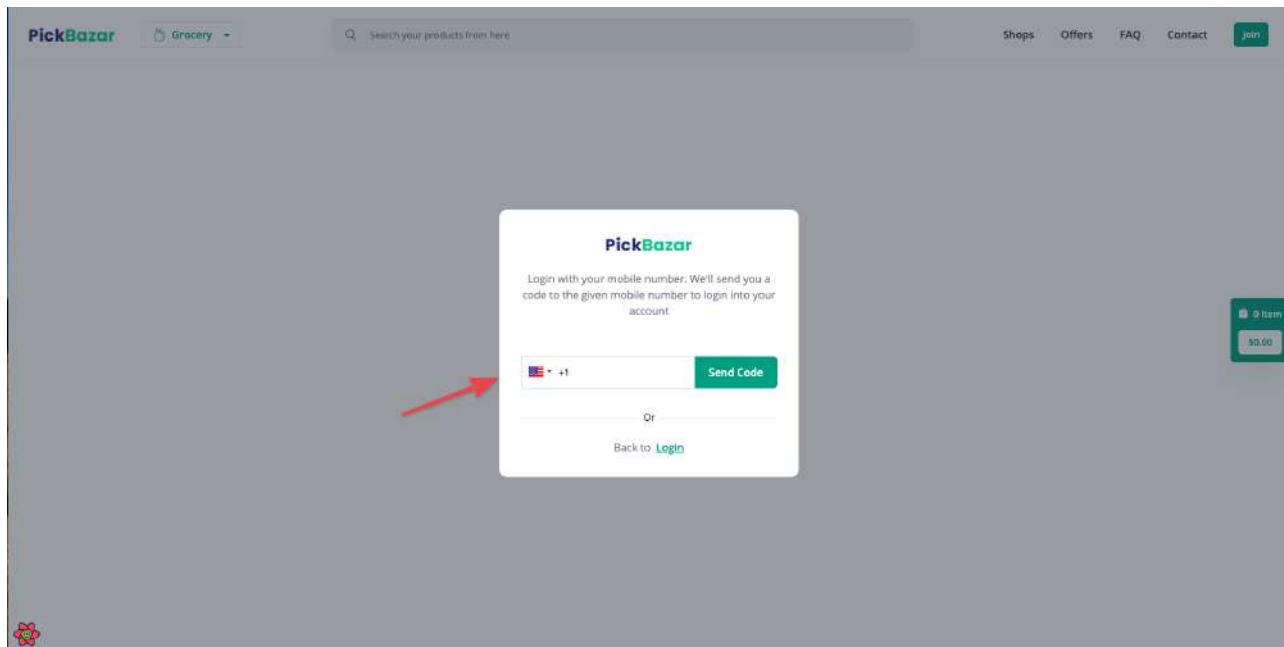
With that, `MessageBird` is appropriately configured for the OTP Service.

FrontEnd Demo

Login with Mobile Number:

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

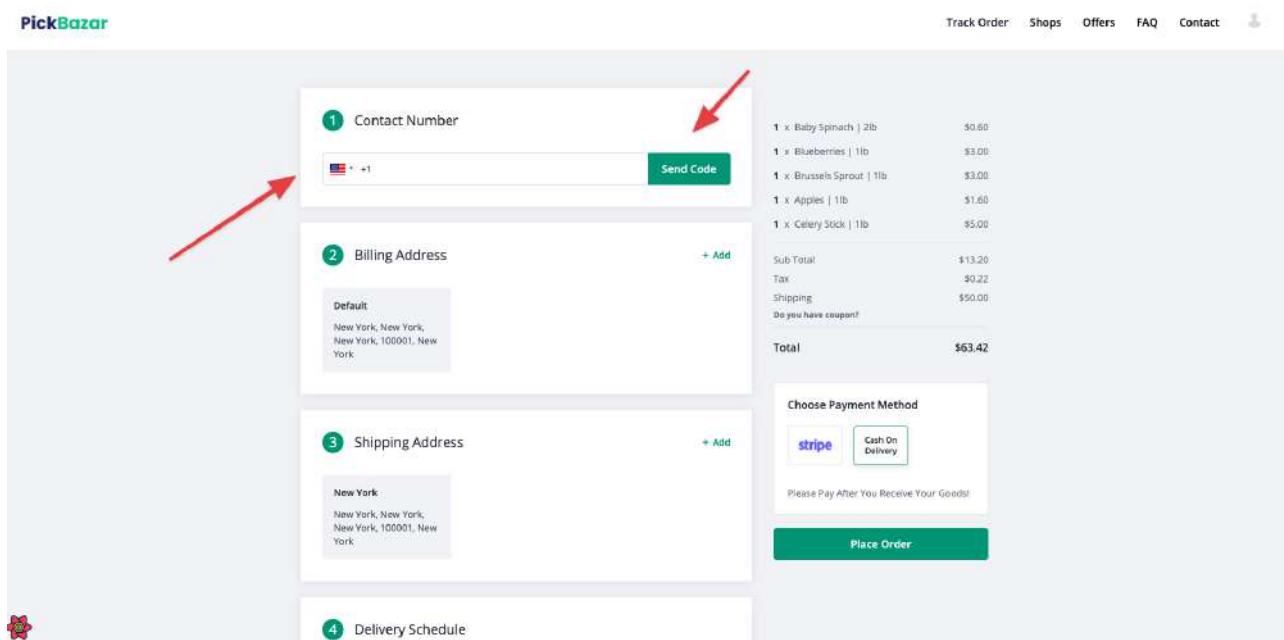




OTP Verification During Checkout Process:

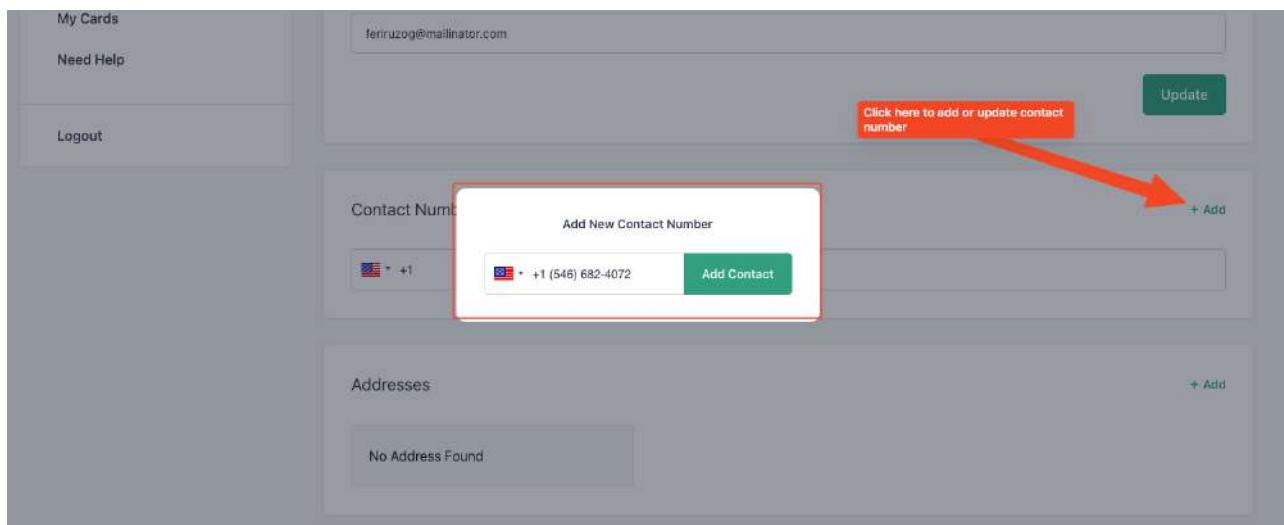
Similarly, when users go to the checkout page, they'll get a form to verify their mobile number.

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



OTP Verification For Updating Mobile Number:

Users also can change the number from their profile.



Wallet

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

Sign up points.

Manually by admin

Refund

Sign Up Points:

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

PickBazar

[Create Shop](#)

- [Categories](#)
- [Tags](#)
- [Manufacturers/Publications](#)
- [Authors](#)
- [Orders](#)
- [Order Status](#)
- [Create Order](#)
- [Users](#)
- [Coupons](#)
- [Taxes](#)
- [Shipments](#)
- [Withdrawals](#)
- [Refunds](#)
- [Questions](#)
- [Reviews](#)
- [Settings](#)

Site Subtitle

Currency

Minimum Order Amount

Wallet Currency Ratio

Sign Up Points

Maximum Question Limit

Use OTP at checkout

Tax Class

Shipping Class

Manually By Admin:

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

PickBazar

[Create Shop](#)

- [Categories](#)
- [Tags](#)
- [Manufacturers/Publications](#)
- [Authors](#)
- [Orders](#)
- [Order Status](#)
- [Create Order](#)
- [Users](#)
- [Coupons](#)
- [Taxes](#)
- [Shipments](#)
- [Withdrawals](#)
- [Refunds](#)
- [Questions](#)
- [Reviews](#)
- [Settings](#)

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

PickBazar

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

Customers

Type your query and press enter

+ Add Customer

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

< 1 >

Refund:

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

PickBazar

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

Refund ID - 1 (Approved)

Approved Change Status

Tracking Number : cWz1Dw7Tq1Yw
Order Status : Order Received
Customer Email : customer@demo.com
Contact Number : 1936514641631

Reason : Emmi
Description : No Reason
Images : No image found

Order Details

Products	Total
Apples x 1	\$1.60

Refund Created : 24 May 2022
Order Created : 24 May 2022



The screenshot shows the PickBazar customer profile interface. At the top left, there's a 'Wallet Points' section with a red arrow pointing to the 'Available' value of 24. To the right is a placeholder for a user profile picture with a 'Upload an image' button. Below these are sections for 'Profile' (with links to Change Password, My Orders, Downloads, My Wishlists, My Questions, My Refunds, My Reports, and Need Help), 'Name' (set to 'Customer'), 'Bio' (empty), and a 'Save' button. At the bottom, there's a 'Contact Number' field with a '+ Update' link.

Wallet Ratio:

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

The screenshot shows the PickBazar admin settings page. On the left is a sidebar with various management links: Categories, Tags, Manufacturers/Publications, Authors, Orders, Order Status, Create Order, Users, Coupons, Taxes, Shipments, Withdrawals, Refunds, Questions, Reviews, and Settings. A red arrow points from the 'Settings' link in the sidebar to the 'Wallet Currency Ratio' input field on the right. The main panel contains fields for 'Your next ecommerce', 'Currency' (set to 'US Dollar'), 'Minimum Order Amount' (0), 'Wallet Currency Ratio' (3), 'Sign Up Points' (100), 'Maximum Question Limit' (5), 'Use OTP at checkout' (switched on), 'Tax Class' (Global), and 'Shipping Class' (Global).

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

Email Configuration

Laravel supports a variety of mail transport drivers to be used while sending an e-mail. Today we will walk you through two types of email service configuration step by step. We will discuss Mailgun and Gmail SMTP configuration here.

Mailgun Configuration

Sandbox domains are restricted to authorized recipients only. If you want to receive your email in sandbox account. Please authorize your email. Up to 5 Authorized Recipients can be added.

The screenshot shows the Mailgun dashboard with the 'Overview' tab selected. On the left, there's a sidebar with various options like 'Dashboard', 'Reporting', 'Sending', 'Domains', 'Overview' (which is highlighted), 'Logs', 'Analytics', 'Templates', 'Suppressions', 'Webhooks', 'IPs', 'Mailing lists', 'Domain settings', 'Receiving', and 'Verifications'. The main area has a heading 'Overview' and a sub-heading 'How would you like to send your emails from sandbox8423fe4c116f4922b95e739ab1cdebb8.mailgun.org?'. It features two boxes: 'API' (described as 'The most flexible, most popular way to send email. Use languages like Ruby, Python, PHP, C# and more.') and 'SMTP' (described as 'The easiest way to send email. Grab your SMTP credentials and plug them into your app's settings.'). To the right, under 'Authorized Recipients', there's a form with 'Email address*' containing 'YOUR_MAIL@EXAMPLE.COM' and a green 'Save Recipient' button. A red arrow points to this button. Below the form, a note says 'Sandbox domains are restricted to authorized recipients only.' At the bottom right of the page, there are links for 'API keys', 'Help center', 'API documentation', 'Postman', and 'HTML email templates'.

For more information, Please follow [Authorized Recipients](#)

To setup and configure Mailgun you need to set these below `.env` keys. We will discuss how you can get those env variable from mailgun.

```
MAIL_MAILER=
MAIL_HOST=
MAILGUN_DOMAIN=
MAILGUN_SECRET=
MAIL_FROM_ADDRESS=
MAIL_PORT=587
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
ADMIN_EMAIL=support@example.com
```

- `MAIL_MAILER=mailgun`
- `MAIL_HOST=smtp.mailgun.org`

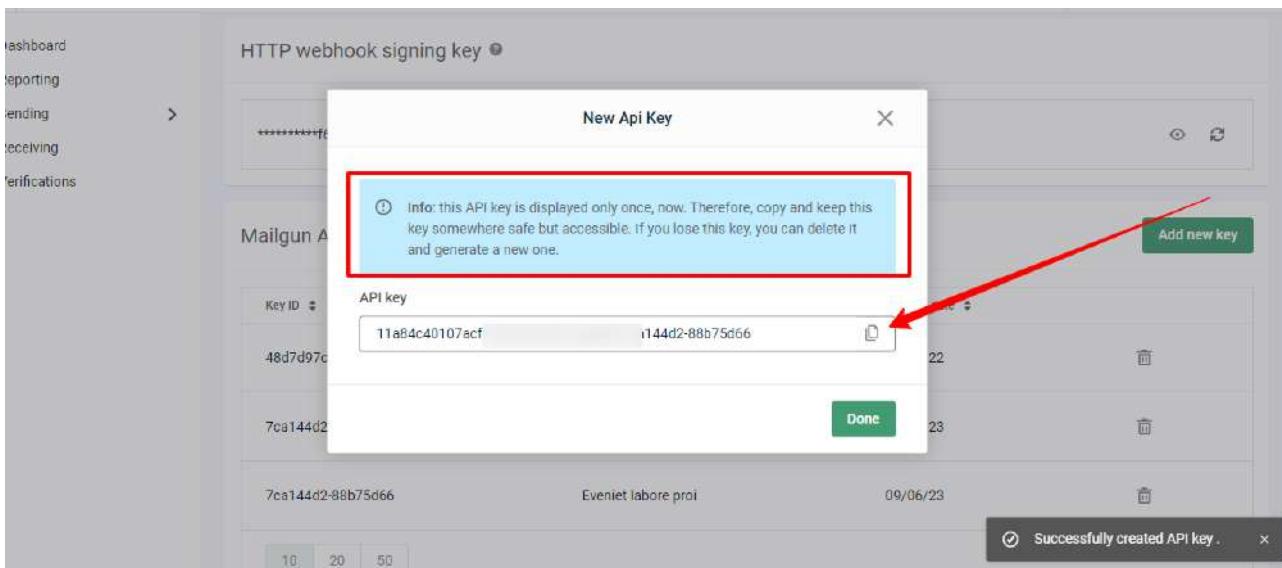
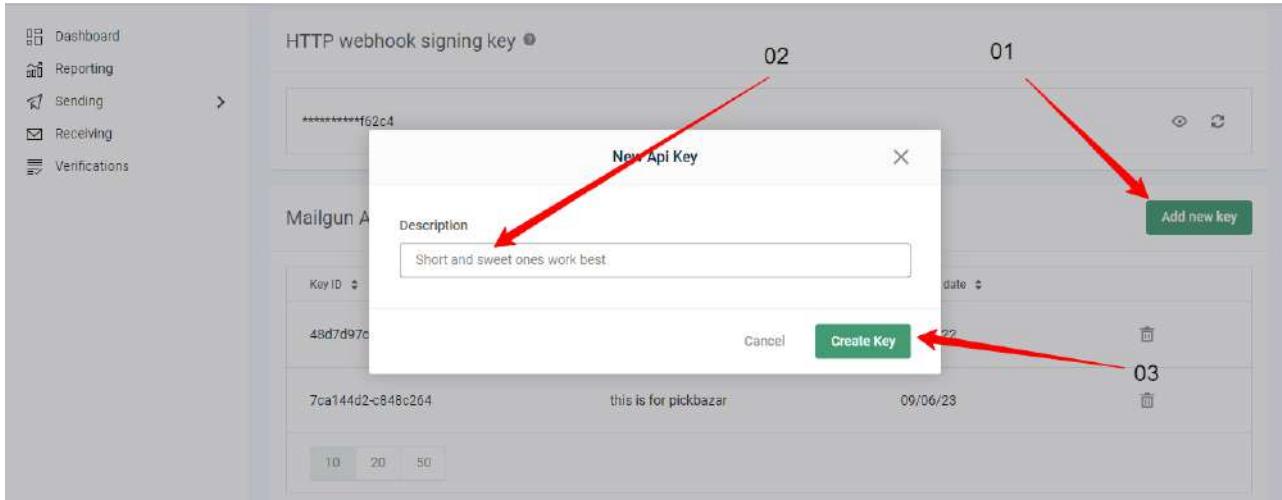
- To get **MAILGUN_DOMAIN** Create or SignIn a Mailgun account from here <https://login.mailgun.com/login> After login click sending you will see a domain list there. I will go with this domain below. Just copy this domain and set this as your **MAILGUN_DOMAIN**

The screenshot shows the Mailgun dashboard with the 'Domains' section selected in the sidebar. The main area displays a list of domains, with one specific domain highlighted: 'sandbox303fb2f5a17b4a7fb87467d3a26eace6.mailgun.org'. This domain is shown with a green checkmark and a small American flag icon.

- Now to obtain **MAILGUN_SECRET** click on on that domain. You will redirect to an over page like below click on API Keys.

The screenshot shows the Mailgun overview page. The sidebar has 'Overview' selected. On the right side, there is a sidebar menu with several options: 'Save Recipient', 'Sandbox domains are restricted to authorized recipients only.', 'API keys' (which is highlighted with a red arrow), 'Help center', 'API documentation', 'Postbin', and 'HTML email templates'.

Then it will redirect to page find **Mailgun API keys** then click on **Add new key**



- To set `MAIL_FROM_ADDRESS` just type your mail address.
- Now go to the dashboard -> sending -> Domain Click on the domain then select SMTP. You will see

```
MAIL_PORT=587  
MAIL_USERNAME=null
```

configuration information there like below.

How would you like to send your emails from sandbox303fb2f5a17b4a7fb87467d3a26eace6.mailgun.org?

API

The most flexible, most popular way to send email.

Use languages like Ruby, Python, PHP, C# and more.

Select

SMTP

The easiest way to send email.

Grab your SMTP credentials and plug them into your app's settings.

Select

Sandbox domains are restricted to authorized recipients only.

How to send with SMTP

Grab your SMTP credentials:

- SMTP hostname: smtp.mailgun.org
- Port: 587 (recommended)
- Username: postmaster@sandbox303fb2f5a17b4a7fb87467d3a26eace6.mailgun.org
- Default password:

[Manage SMTP credentials](#)

Integrate with your app:

- In a Mailgun configuration, you typically don't need to specify a `MAIL_PASSWORD` because Mailgun uses API keys for authentication, not SMTP passwords. You can set `MAIL_PASSWORD=null` or remove it from your .env file.
- Set your `MAIL_ENCRYPTION` `MAIL_ENCRYPTION=tls`
- You can set your admin email `ADMIN_EMAIL=your_email@example.com`

Gmail SMTP Configuration

To setup Gmail SMTP Configuration you need to set these below `.env` keys. We will discuss how you can get those env variable from Gmail.

```
MAIL_MAILER=
MAIL_HOST=
MAILGUN_DOMAIN=
MAILGUN_SECRET=
MAIL_FROM_ADDRESS=
MAIL_PORT=587
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
ADMIN_EMAIL=support@example.com
```

- `MAIL_MAILER=smtp`

- `MAIL_HOST=smtp.gmail.com`
- You can ignore `MAILGUN_DOMAIN` `MAILGUN_SECRET` for gmail.
- To set `MAIL_FROM_ADDRESS` just type your mail address.
- `MAIL_PORT=587`
- To set `MAIL_USERNAME=your@gmail.com` just type your gmail address that you are using for configuration.
- To get `MAIL_PASSWORD` from google Create or SignIn to a gmail account. Then go to your account or click the URL <https://myaccount.google.com/security> Then turn on 2-Step Verification.

The screenshot shows the 'Security' section of the Google Account settings. On the left, there's a sidebar with links: Data & privacy, Security (which is selected and highlighted in blue), People & sharing, Payments & subscriptions, and About. The main content area is titled 'Review security activity' and contains a section titled 'How you sign in to Google'. It says 'Make sure you can always access your Google Account by keeping this information up to date.' Below this, there are three items: '2-Step Verification' (with a red arrow pointing to it), 'Passkeys', and 'Password'. Each item has a status indicator (green checkmark for 2-Step Verification, grey for Passkeys and Password), a last updated date ('On since Oct 30, 2022' for 2-Step Verification, 'Last changed Nov 14, 2021' for others), and a 'View details' link.

Under 2-Step Verification settings you will see app passwords and click that -

The screenshot shows the '2-Step Verification' settings page. At the top, there's a link to 'Devices that don't need a second step' with a note: 'You can skip the second step on devices you trust, such as your own computer.' Below this is a section titled 'Devices you trust' with a 'REVOKE ALL' button. Further down is an 'App passwords' section with a note: 'App Passwords aren't recommended and are unnecessary in most cases. To help keep your account secure, use "Sign in with Google" to connect apps to your Google Account.' A red arrow points to the 'App passwords' link. At the bottom of the page, there are links for Privacy, Terms, Help, and About.

now set your application name under other (custom name)

← App passwords

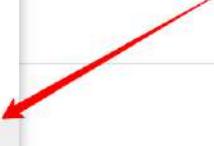
App passwords let you sign in to your Google Account from apps on devices that don't support 2-Step Verification. You'll only need to enter it once so you don't need to remember it. [Learn more](#)

You don't have any app passwords.

Select the app and device you want to generate the app password for.

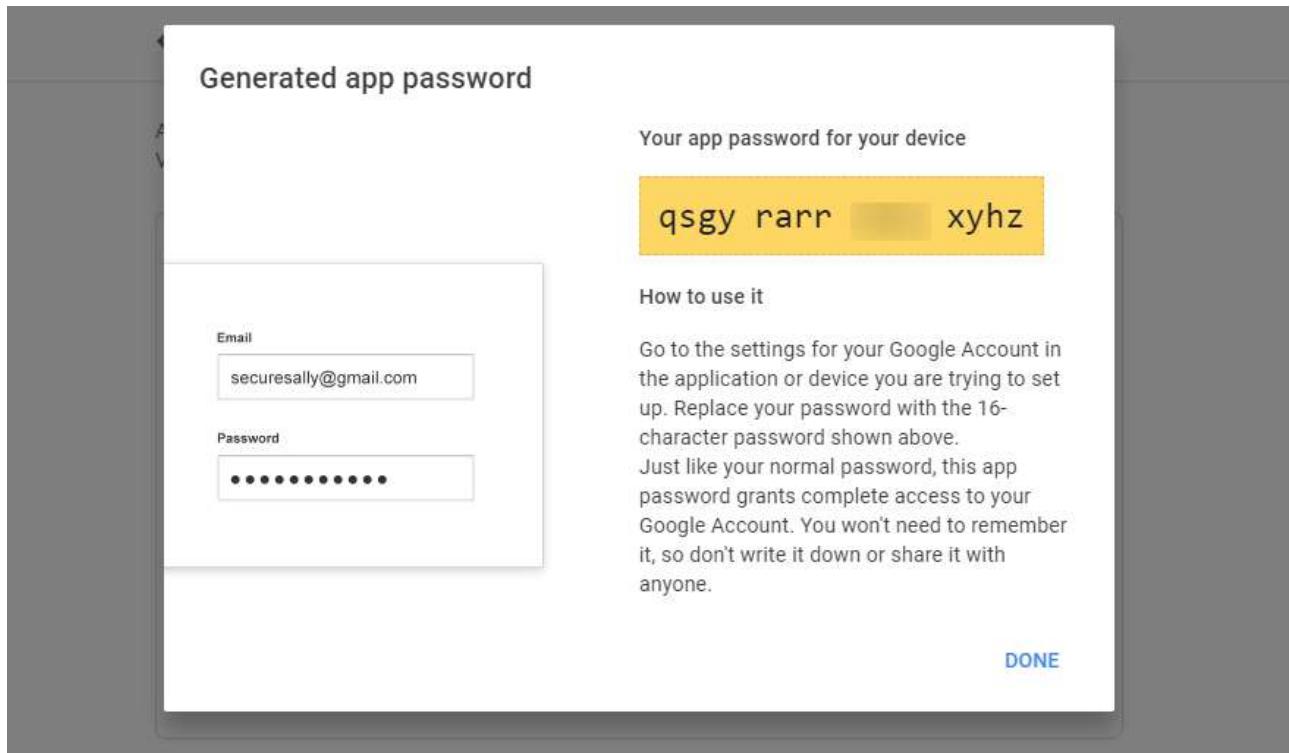
Select app	Select device
Mail	
Calendar	
Contacts	
YouTube	
Other (Custom name)	

GENERATE



[Privacy](#) [Terms](#) [Help](#) [About](#)

after typing the name click on generate a propt will popup. like below That is your **MAIL_PASSWORD** copy that.



- Set your MAIL_ENCRYPTION `MAIL_ENCRYPTION=tls`
- You can set your admin email `ADMIN_EMAIL=your_email@example.com`

Email Verification

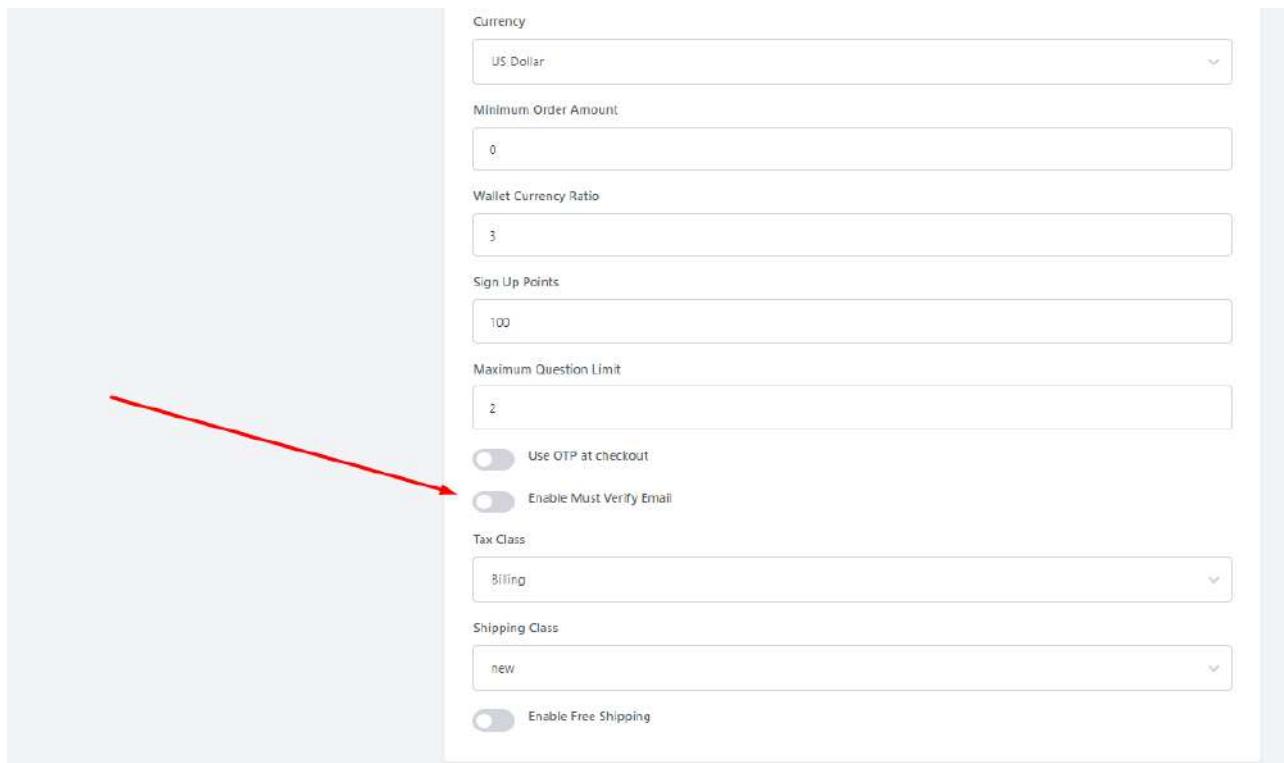
We implement the **Email verification** feature on PickBazar v8.2.0. So if you want to use **Email verification** with **PickBazar**, then make sure your PickBazar is **v8.2.0** or later.

With the Email Verification feature, you can do,

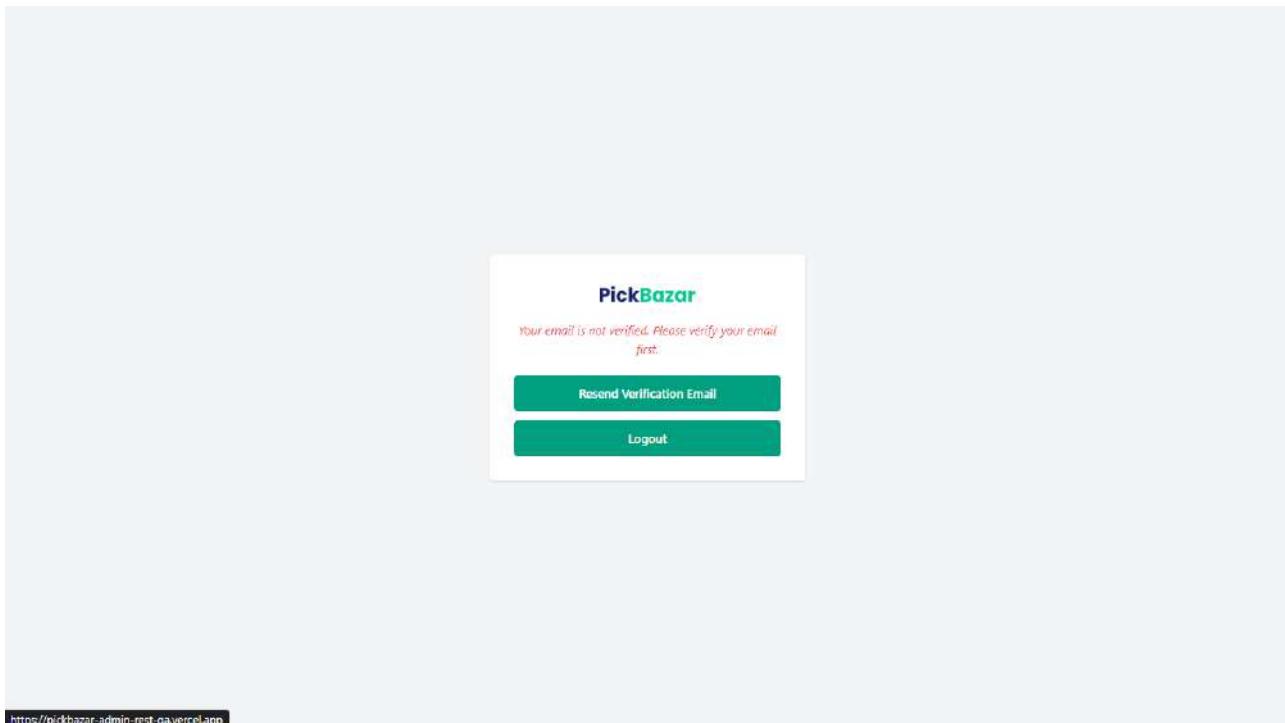
- Get Email After Registration
- Email verification for updating Email in profile update section
- when Email verification is on in settings then users will be forced to verify their email.

How to turn on this feature

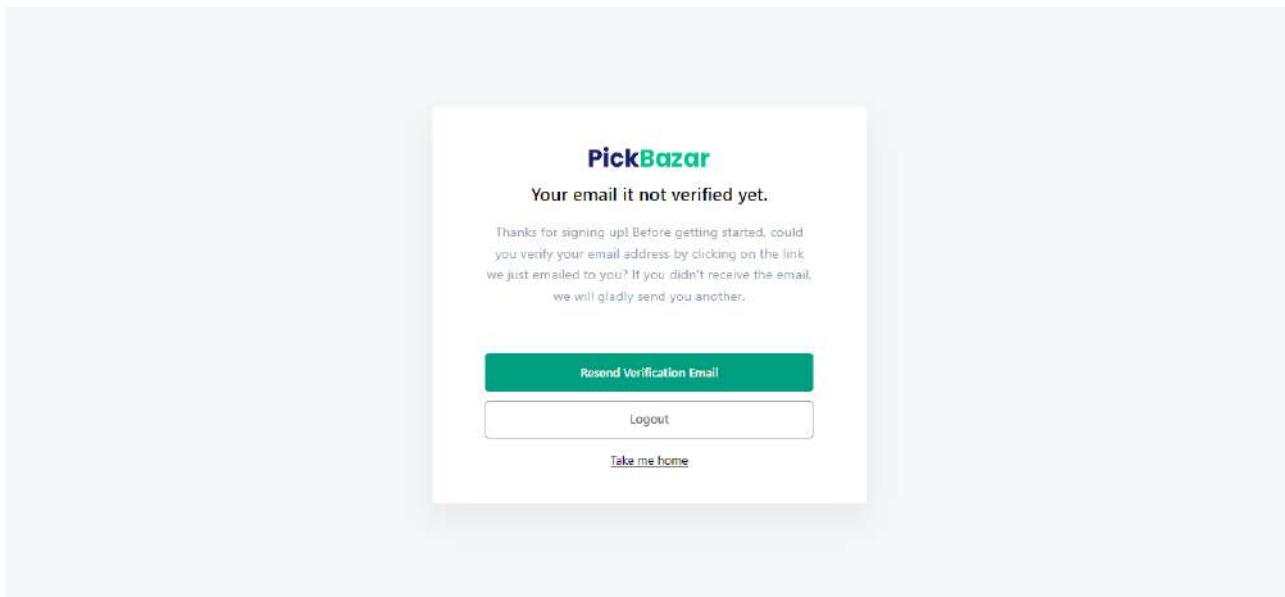
To turn on this feature go to settings of your site then find **Enable Must Verify Email** switch.



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
```

Subscribe setup

In Pickbazar we implement Subscribe email through our API. To make this work properly in shop End. You have to configure it in the API (BackEnd).

Below .env variables are required to work Subscribe email. We will go through the whole process how you can get those env variable.

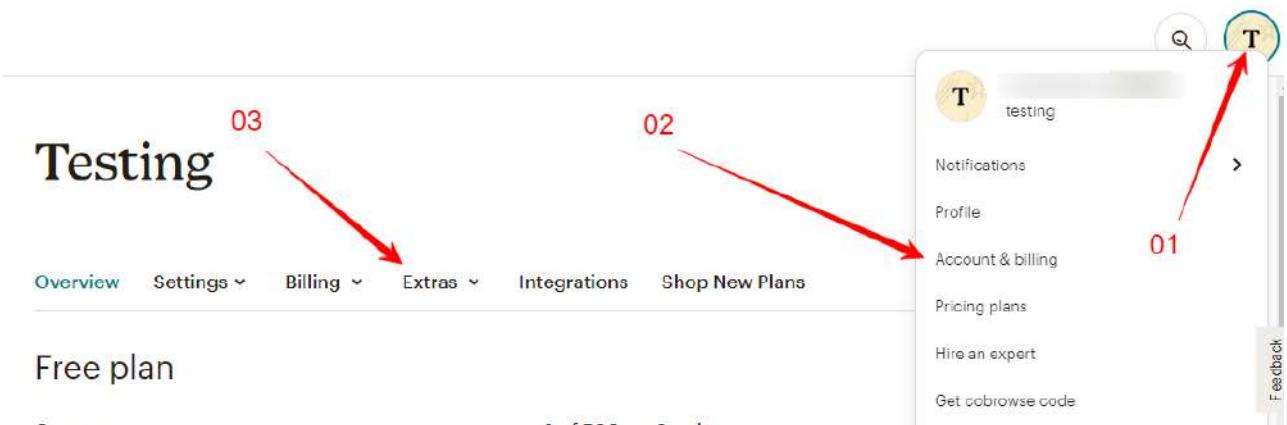
```
# Mailchimp Newsletter
NEWSLETTER_API_KEY=
NEWSLETTER_LIST_ID=
```

Implementation of Subscribe

In Pickbazar we used `spatie/laravel-newsletter` package for as Newsletter manager. The `spatie/laravel-newsletter` dependant on two Newsletter manager `mailcoach` and `mailchimp`. But in Pickbazar we used and configure only `mailchimp`.

How to Get Mailchimp API Key

- Go to <https://mailchimp.com/signup/> to signup for a mailchimp account.
- Once logged in, click on your profile name at the top right corner of the page. In the dropdown menu, select "Account."



- Within the Account page, navigate to the "Extras" dropdown menu and choose "API keys."
- On the API keys page, you'll see a section titled "Your API keys." If you have previously generated API keys, they will be listed here.

Your API keys

You can review, revoke or generate new API keys below. [Learn more about generating, revoking, and accessing API keys here.](#)

Created	User	Label	API key	i
			e8ca*****-us21	<button>Revoke</button>

- If you don't have an existing API key, click on the "Create A Key" button. Otherwise, you can select an existing key from the list.

Mailchimp will generate a new API key for you. You'll see a string of letters and numbers similar to the following format:

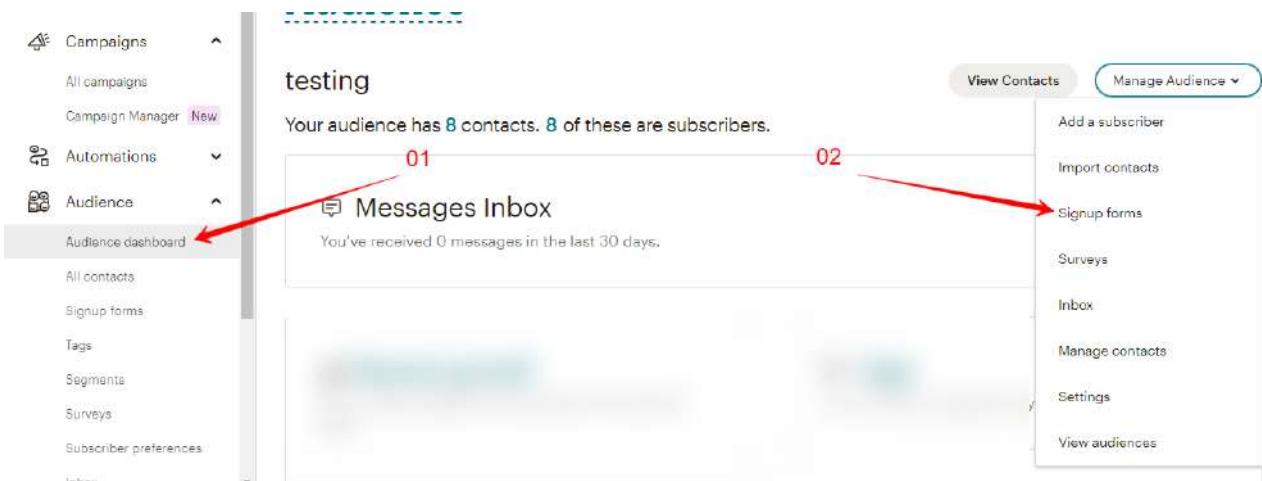
1234567890abcdef1234567890abcdef-us1

- Click the "Copy" button to copy the API key to your clipboard. Make sure to securely store this key as it provides access to your Mailchimp account.

That's it! You now have your Mailchimp API key, which you can use to interact with the Mailchimp API and perform various tasks, such as managing subscribers, creating campaigns, and more.

How to Get Mailchimp List ID

- From the Audience dashboard, select the desired audience or create a new one by clicking on the "Manage Audience" dropdown and choosing "View Audience" or "Create Audience," respectively.
- Within the Audience page, click on the "Manage Audience" dropdown again and select "Signup forms."



- You will be presented with various options for your signup form. Choose **Embedded forms** for Form integrations."

The screenshot shows the Mailchimp Form builder interface. At the top, there are navigation links: Overview, Manage contacts, Add contacts, Subscriber preferences, Settings, and a search bar. Below these are three sections: 'Form builder' (Build, design, and translate forms and response emails), 'Embedded forms' (Generate HTML code to embed in your site or blog to collect signups), and 'Subscriber pop-up' (Design a pop-up signup form that can be embedded on any site). Each section has a 'Select' button to its right. A large red arrow points from the 'Embedded forms' section towards the 'Select' button.

- After selecting the desired form type, you'll be directed to the form builder. Click continue

The screenshot shows the Mailchimp Embedded form builder interface. On the left, there's a sidebar with options: 'Form Fields', 'Settings', 'Tags', and 'Referral Badge'. The main area displays a 'Subscribe' form with fields for 'Email Address' and a 'Subscribe' button. A small note at the bottom says 'Keep the focus on your brand' and 'Upgrade to remove Mailchimp branding.' In the top right corner, there are 'Finish later' and 'Continue' buttons, with a red arrow pointing to the 'Continue' button.

- Locate the form code and look for the HTML snippet that contains the "form action" attribute. It should resemble the following code:

```
form action="https://yourusername.us#list-id-number.campaign-archive.com/subscribe/post?u=xxxxxxxx&id=xxxxxxx"
```

- In the HTML snippet, find the "id" parameter in the form action URL. The value after the "id=" is your List ID. It will be a combination of letters and numbers.
- For example, in the above code snippet, the List ID would be represented by the "xxxxxxxx" placeholder.

That's it! You have now found the List ID for your Mailchimp newsletter subscription list.

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

Introduction

In business, refunds are often issued to clients when a service or product cannot be delivered as promised. This could be due to unexpected circumstances such as illness, family emergencies, or miscommunication.

Key Notes

- First, after completing successful order through **online payment** except 'Cash on delivery (payment must be successful) he has to claim a refund.
- After completing payment, for the order he wants to refund. Now he can select the order from the link on the **My Order** page and click on the **Ask For A Refund** button to claim the refund.
- A modal form will open. If the form is filled out properly, the refund request will go to admin.
- The rest of the issues will be handled by the admin and vendor.

Returns are a common occurrence in the retail industry. Understanding the reasons behind returns can help merchants improve their business practices and increase customer satisfaction.

First go to this link [/refund-reasons](#).

If you have no refund reason you see an empty refund reason list.

Now create your first refund reason.

[Manufacturers/Publications](#)[Authors](#)[E-COMMERCE MANAGEMENT](#)[Taxes](#)[Shipments](#)[Withdrawals](#)[Refunds](#)[LAYOUT/PAGE CONTROL](#)[Home pages / Types](#)[FAQs](#)[Terms And Conditions](#)[ORDER MANAGEMENT](#)[Orders](#)[Create Order](#)[Transactions](#)[USER CONTROL](#)[Logout](#)

Add New Refund Reason

Description

Add Refund Reason's information from here.

Refund Reason Title*

Title is required

Slug

[Add Refund Reason](#)

©2023 Pickbazar.

Your created refund reason list here.

[Manufacturers/Publications](#)[Authors](#)[E-COMMERCE MANAGEMENT](#)[Taxes](#)[Shipments](#)[Withdrawals](#)[Refunds](#)[LAYOUT/PAGE CONTROL](#)[Home pages / Types](#)[FAQs](#)[Terms And Conditions](#)[ORDER MANAGEMENT](#)[Orders](#)[Create Order](#)[Transactions](#)[USER CONTROL](#)[Logout](#)

Refund Reasons

[+ Add Refund Reason](#)

ID	Name	Slug	Actions
5	Incorrect fit	incorrect-fit	
4	Damaged or defective products	damaged-or-defective-products	
3	Unmet expectations	unmet-expectations	
2	Broken item	broken-item	

< 1 >

©2023 Pickbazar.

Finally, you find the refund reason here.

My Orders

Order#15

Order Date: [dropdown]

Delivery Time: [dropdown]

Amount: [dropdown]

Total Price: [dropdown]

Order#13

Order Date: [dropdown]

Delivery Time: [dropdown]

Amount: [dropdown]

Total Price: [dropdown]

Order#11

Order Date: [dropdown]

Delivery Time: [dropdown]

Amount: [dropdown]

Total Price: [dropdown]

Processing

Order At Local Facility

Order Out For Delivery

Order Completed

Quantity

Price

Sub Total: \$14.70

Discount: \$0.00

Delivery Fee: \$0.00

Total: \$14.70

Payment Status: Payment Success

Ask for a refund Details

Logout

others are magical keywords for refund reasons. If you select *others* reason you find a custom input field and you describe your reason here.

My Orders

Order#22

Order Date: [dropdown]

Delivery Time: [dropdown]

Amount: [dropdown]

Total Price: [dropdown]

Order#19

Order Date: [dropdown]

Delivery Time: [dropdown]

Amount: [dropdown]

Total Price: [dropdown]

Order#17

Order Date: [dropdown]

Delivery Time: [dropdown]

Amount: [dropdown]

Total Price: [dropdown]

Processing

Order At Local Facility

Order Out For Delivery

Order Completed

Quantity

Price

Sub Total: \$6.00

Discount: \$0.00

Delivery Fee: \$0.00

Total: \$6.00

Payment Status: Payment Success

Ask for a refund Details

Logout

A **refund** is a repayment of a sum of money. It is typically given back to a customer who is not satisfied with goods or services bought.

First, create your order. Cash on delivery is not acceptable for refund requests.

1 Contact Number

+ Update

+1 (936) 514-1641

2 Billing Address

+ Add

Billing
2321 Kidd Avenue, AK, Kipnuk, 99614, United States

3 Shipping Address

+ Add

Shipping
2148 Stratford Park, KY, Winchester, 40391, United States

4 Delivery Schedule

Morning
9:00 - 10:00 AM

5 Order Note

Please click Place order to make order and payment

Choose Payment Method

stripe **PayPal**

Do you want to use wallet?

Place Order

Now complete your payment.

Back to Home

Order Status : **Order Processing** Payment Status : **Payment Success**

Order Number 20230919921759	Date September 20, 2023	Total \$11.00	Payment Method STRIPE
--------------------------------	----------------------------	------------------	--------------------------

Order Progress:

1 Order Pending 2 Order Processing 3 Order At Local Facility 4 Order Out For Delivery 5 Order Completed

Total Amount

Sub Total : \$11.00	Name : Customer
Shipping Charge : \$0.00	Total Item : 3 items
Tax : \$0.00	Delivery Time : Morning
Discount : \$0.00	Shipping Address : 2148 Stratford Park, KY, Winchester, 40391, United States
Total : \$11.00	Billing Address : 2321 Kidd Avenue, AK, Kipnuk, 99614, United States

Order Details

Item	Quantity	Price
Blueberries x 1lb \$3.00	1	\$3.00
Brussels Sprout x 1lb \$3.00	1	\$3.00
Celery Stick x 1lb \$5.00	1	\$5.00

See your last order here [/orders](#) in the shop end.



Wallet Points

0	0	0
Total	Used	Available

Profile

Change Password

My Orders

Downloads

My Wishlists

My Questions

My Refunds

My Reports

My Cards

Need Help

Logout

My Orders

Order#17 Order Processing

Order Date: September 20, 2023
Delivery Time: Morning
Amount: \$11.00
Total Price: \$11.00

Order#15 Order Processing

Order Date: September 19, 2023
Delivery Time: Morning
Amount: \$14.70
Total Price: \$14.70

Order#13 Order Completed

Order Date: April 10, 2023
Delivery Time:

Order Details - 20230919921759

Order Status: Order Processing Payment Status: Payment Success

Shipping Address
2148 Stratford Park, KY, Winchester, 40391, United States

Billing Address
2231 Kidd Avenue, AK, Kipmuk, 99614, United States

Sub Total	\$11.00
Discount	\$0.00
Delivery Fee	\$0.00
Tax	\$0.00
Total	\$11.00

Order Progress

Item **Quantity** **Price**

- Blueberries x 1lb \$1.00
- Brussels Sprout x 1lb \$1.00
- Celery Stick x 1lb \$5.00

Actions: Update review

See your last order here `/orders/orderID` in the admin end.

PickBazar

Search your route...

Create Shop **Visit Site** **Language English** **Admin super admin**

MAIN

Dashboard

SHOP MANAGEMENT

Shops **My Shops**

PRODUCT MANAGEMENT

Products **Inventory**

Categories **Tags** **Attributes**

Manufacturers/Publications **Authors**

E-COMMERCE MANAGEMENT

Taxes **Shipments** **Withdrawals**

Order Status: Processing **Payment Status: Payment Success**

Order ID - 20230919921759

Download Invoice **Change Status**

Order Progress

Products

	Total
Blueberries x 1	\$3.00
Brussels Sprout x 1	\$3.00
Celery Stick x 1	\$5.00

Order Details

3 Items
Morning
Payment Method: STRIPE

Billing Address

Customer: 2231 Kidd Avenue, Kipmuk, AK, 99614, United States
+1 9365141641631

Shipping Address

Customer: 2148 Stratford Park, Winchester, KY, 40391, United States
+1 9385141641631

©2023 Pickbazar.

Refund Request

Now create a refund request. Press the `Ask for a refund` button now open a modal. Fill up everything perfectly.

The refund was created successfully.

Refund Reports

My refunds list. You find the refund list here `/refunds` in the shop end.

This screenshot shows a user profile page from PickBazar. On the left, there's a sidebar with links like Profile, Change Password, My Orders, Downloads, My Wishlists, My Questions, My Refunds (which is currently selected and highlighted in green), My Reports, My Cards, and Need Help. Below that is a Logout button. The main content area has a section titled "My Refunds" with a table showing one refund entry:

ID	Refund Reason	Status	Tracking Number	Amount	Date	Details
1	Damaged or defective products	Pending	20230919921759	\$11.00	a few seconds ago	@

Refunds list. You find the refund list here `/refunds` on the admin end.

This screenshot shows the admin dashboard for PickBazar. The left sidebar includes sections for Manufacturers/Publications, Authors, E-commerce Management, Taxes, Shipments, Withdrawals, Refunds (selected and expanded), Refund Policies, Add new refund policy, Refund Reasons, Add new refund reason, Home pages / Types, FAQs, Terms And Conditions, and Orders. The main content area is titled "Refunds" and displays a table of refund details. A filter bar at the top right allows filtering by reason.

ID	Refund Reason	Customer Email	Amount	Tracking Number	Created	Order Date	Status	Actions
#ID: 1	Damaged or defective products	customer@demo.com	\$11.00	20230919921759	A Minute Ago	A Minute Ago	Pending	@

Refunds list. You find the refund list here `/shopName/refunds` on the vendor end.

PickBazar

Search your route...

Create Shop Visit Site Language English Store Owner

MAIN

Dashboard

PRODUCT MANAGEMENT

- Products
- Inventory
- Attributes
- Manufacturers/Publications
- Authors

FINANCIAL MANAGEMENT

- Withdrawals
- Refunds**

ORDER MANAGEMENT

- Orders
- Transactions

FEATURE MANAGEMENT

- Store Notice
- Messana

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

10.2.0

Check refund details here on the admin end.

PickBazar

Search your route...

Create Shop Visit Site Language English Store Owner

MAIN

Dashboard

PRODUCT MANAGEMENT

- Products
- Inventory
- Attributes
- Manufacturers/Publications
- Authors

FINANCIAL MANAGEMENT

- Withdrawals
- Refunds

ORDER MANAGEMENT

- Orders
- Transactions

FEATURE MANAGEMENT

- Store Notice
- Messana

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

10.2.0

Refund Status : Pending

Refund ID - #2

Refund Created	01 November 2023
Order Created	01 November 2023
Tracking Number	20231025441702
Customer Email	Customer@Demo.Com
Contact Number	19385141641631

Refund Reason

Product Not as Described

Description

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totem rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptates sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut erim ad minimam veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quia autem vel eum lire reprehenderit qui in ea voluptate velt esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla paratur? At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque compiti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere.

Images

Order Details

Products	Total
Brothers Stylish Dining Table x 1	\$720.00
Sub total	\$720.00
Tax	\$0.00
Delivery fee	\$0.00
Discount	\$0.00
Total	\$720.00

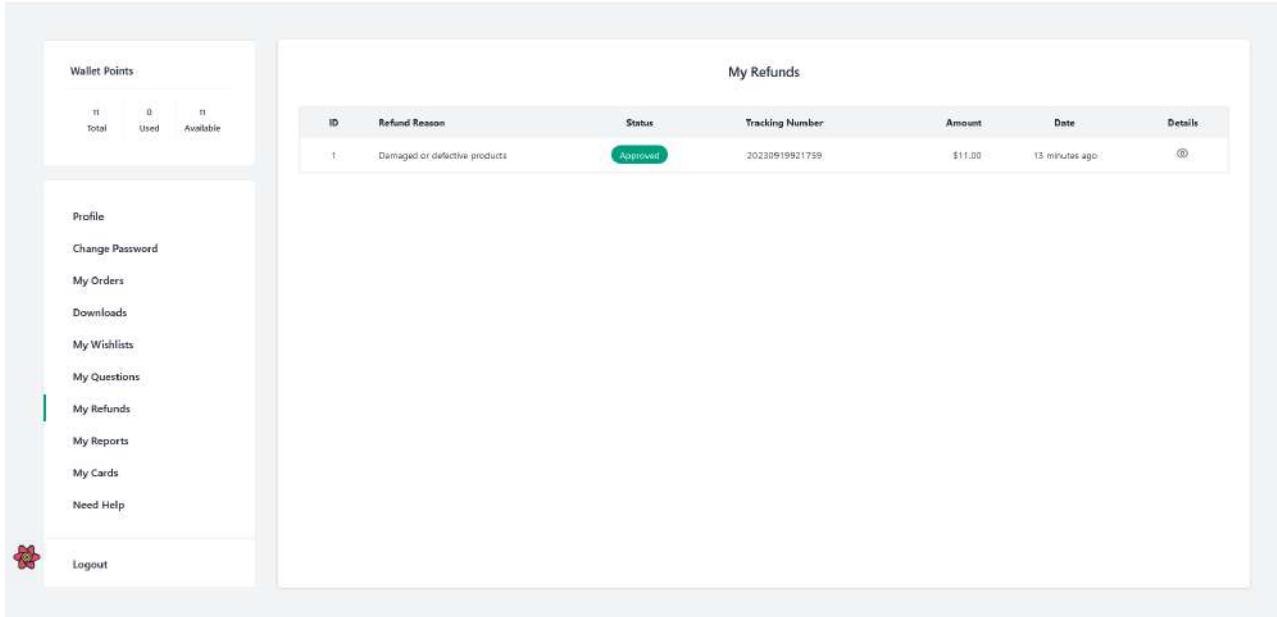
Billing Address

Customer
2231 Kidd Avenue, Kipnuk, AK, 99614, United States
19385141641631

Shipping Address

Customer
2148 Stratford Park, Winchester, KY, 40391, United States
19385141641631

Approved refunds.



The screenshot shows the customer dashboard of the PickBazar platform. On the left, there's a sidebar with links for Profile, Change Password, My Orders, Downloads, My Wishlists, My Questions, My Refunds (which is currently selected and highlighted with a green border), My Reports, My Cards, and Need Help. Below the sidebar is a 'Logout' button with a user icon. The main content area has two sections: 'Wallet Points' (Total: 11, Used: 0, Available: 11) and 'My Refunds'. The 'My Refunds' section displays a table with one row of data:

ID	Refund Reason	Status	Tracking Number	Amount	Date	Details
1	Damaged or defective products	Approved	20230919921759	\$11.00	13 minutes ago	

Customer and vendor end

A **refund policy** is a document that outlines the rules for getting refunds for purchased goods and services. It often details the eligibility requirements for refunds, types of refunds given, the refund time frame, and the return process.

A well-written refund policy can help businesses retain customers and increase sales. However, overly restrictive or nonexistent policies tend to discourage customers.

It's important to note that the specific details of a refund policy may vary depending on the nature of the business and the products or services being sold. Additionally, legal requirements for refunds may vary by jurisdiction.

Only approved policy shown in the front end.

You find the refund policy list here [/refund-policies](#) on the admin end.

PickBazar

Search your route...

Create Shop Visit Site Language English Admin super admin

Refunds

- Reported refunds
- Refund Policies**
- Add new refund policy
- Refund Reasons
- Add new refund reason

LAYOUT/PAGE CONTROL

- Home pages / Types
- FAQs
- Terms And Conditions

ORDER MANAGEMENT

- Orders
- Create Order
- Transactions

USER CONTROL

- All users
- Admin list

No data found Sorry we couldn't find any data.

©2023 Pickbazar.

Create your own refund policy.

PickBazar

Search your route...

Create Shop Visit Site Language English Admin super admin

Add New Refund Policy

Description Add Refund Policy's Information from here.

Refund Policy Heading* Enter Your Refund Policy Heading

Slug

Refund Policy Description

Select Refund Policy For* Select Refund Policy Users

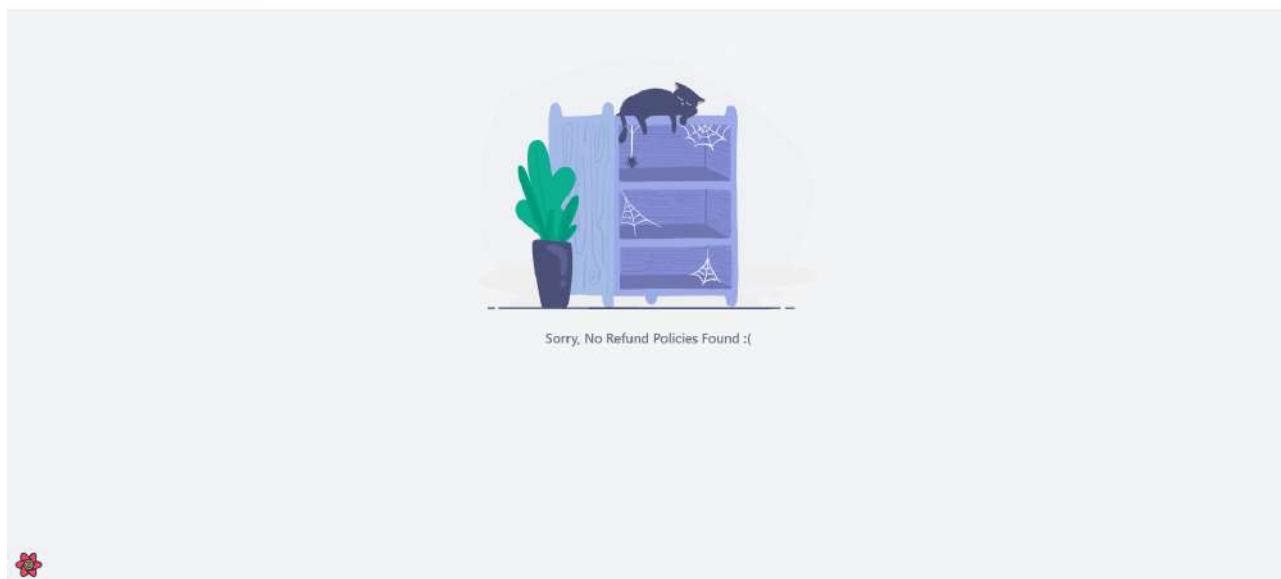
Status

Approved
 Pending

Add Refund Policy

©2023 Pickbazar.

You find the refund policies list here on the shop end `/vendor-refund-policies` and customer end `/customer-refund-policies`.



Customer

Customer refund policies are listed on the admin end.

Screenshot of the PickBazar admin end showing the 'Refund Policies' page. The page displays a list of five refund policies, each with a heading, description, status (approved or pending), and actions (edit, delete, etc.).

ID	Heading	Description	Applied on	Status	Actions
8	And yet I feel that I never was a greater artist than now.	A wonderful serenity has taken possession of my entire soul, like these sweet mornings of spring which I enjoy with my whole heart, I am alone, and feel the charm of existence in this spot, which was created for the bliss of souls like mine. I am so happy, my dear friend, so absorbed in the exquisite sense of mere tranquill existence, that I neglect my talents.	CUSTOMER	approved	  
7	It and seemed ready to slide off any moment.	One morning, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin. He lay on his armour-like back, and if he lifted his head a little he could see his brown belly, slightly domed and divided by arches into stiff sections. The bedding was hardly able to cover it and seemed ready to slide off any moment. His many legs, pitifully thin compared with the size of the rest of him, waved about helplessly as he looked. "What's happened to me?" he thought. It wasn't a dream. His room, a proper human room although a little too small, lay peacefully between its four familiar walls.	CUSTOMER	approved	  
6	Blowzy red vixens fight for a quick jump.	Joaquin Phoenix was gazed by MTV for luck. A wizard's job is to vex chumps quickly in fog. Witch "Leopardyl", Alex Trebek's fun TV quiz game. Woven silk pajamas exchanged for blue quartz. Brawny gods just	CUSTOMER	approved	  
5	A very bad quack might jinx zippy fowls. Few quips galvanized the mock jury box.	A very bad quack might jinx zippy fowls. Few quips galvanized the mock jury box. Quick brown dogs jump over the lazy fox. The jay, pig, fox, zebra, and my wolves quack!	CUSTOMER	pending	  

Customer refund policies are listed on the shop end.



Customer Refund Policy

[Home](#) > Customer Refund Policy

Blowzy red vixens fight for a quick jump.

It and seemed ready to slide off any moment.

And yet I feel that I never was a greater artist than now.

Blowzy red vixens fight for a quick jump.

Joaquin Phoenix was gazed by MTV for luck. A wizard's job is to vex chumps quickly in fog. Watch "Jeopardy!", Alex Trebek's fun TV quiz game. Woven silk pyjamas exchanged for blue quartz. Brawny gods just

It and seemed ready to slide off any moment.

One morning, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin. He lay on his armour-like back, and if he lifted his head a little he could see his brown belly, slightly domed and divided by arches into stiff sections. The bedding was hardly able to cover it and seemed ready to slide off any moment. His many legs, pitifully thin compared with the size of the rest of him, waved about helplessly as he lay. "What's happened to me?" he thought. It wasn't a dream. His room, a proper human room although a little too small, lay peacefully between its four familiar walls.

And yet I feel that I never was a greater artist than now.

A wonderful serenity has taken possession of my entire soul, like these sweet mornings of spring which I enjoy with my whole heart. I am alone, and feel the charm of existence in this spot, which was created for the bliss of souls like mine. I am so happy, my dear friend, so absorbed in the exquisite sense of mere tranquill existence, that I neglect my talents.



Vendor

Vendor refund policies are listed on the admin end.

The screenshot shows the PickBazar Admin dashboard with the 'Refund Policies' section highlighted. The page includes a search bar, filter options (Search by Heading, Vendor, Filter by Status), and a 'Create Shop' button. Two refund policies are listed:

ID	Heading	Description	Applied on	Status	Actions
10	Pityful a rhetoric question ran over her cheek.	Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regelialia.	VENDOR	approved	
9	I should be incapable of drawing a single stroke.	A wonderful serenity has taken possession of my entire soul, like these sweet mornings of spring which I enjoy with my whole heart. I am alone, and feel the charm of existence in this spot, which was created for the bliss of souls like mine. I am so happy, my dear friend, so absorbed in the exquisite sense of mere tranquill existence, that I neglect my talents. I should be incapable of drawing a single stroke at the present moment; and yet I feel that I never was a greater artist than now.	VENDOR	approved	

At the bottom left, there are navigation icons for 'Dashboard', 'My Shops', 'Products', 'Inventory', 'Categories', 'Tags', 'Attributes', 'Manufacturers/Publications', and 'Authors'. On the right, there are links for 'Visit Site', 'Language English', and 'Admin super admin'.

Vendor refund policies are listed on the shop end.

Vendor Refund Policy

[Home](#) > Vendor Refund Policy

I should be incapable of drawing a single stroke.

Pityful a rhetoric question ran over her cheek.

I should be incapable of drawing a single stroke.

A wonderful serenity has taken possession of my entire soul, like these sweet mornings of spring which I enjoy with my whole heart. I am alone, and feel the charm of existence in this spot, which was created for the bliss of souls like mine. I am so happy, my dear friend, so absorbed in the exquisite sense of mere tranquill existence, that I neglect my talents. I should be incapable of drawing a single stroke at the present moment; and yet I feel that I never was a greater artist than now.

Pityful a rhetoric question ran over her cheek.

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regelialia.



Introduction to Real-time Notification

In today's fast-growing digital era, staying connected and informed is paramount. Real-time notifications have appeared as an necessary tool, revolutionizing the way we receive and process information. These notifications are instant, actionable, and personalized, enabling us to stay in the loop, make quick decisions, and engage with our digital environments like never before.

We have introduced this feature on PickBazar **v11.0.0**. So if you want to use this with **PickBazar**, then make sure your PickBazar is **v11.0.0** or later.

We have plans to enrich this feature area by integrating more functionalites in future. Before go through lets denotes the key points. We have divided this feature into 3 different areas.

- **Store notices**
- **Orders**
- **Messaging**

But before make this usable we need to configure some settings and our application environments. We are using **Pusher** for this real time notification feature.

Let's discuss those sequentially in upcoming chapters.

Configure pusher

At first we need to configure pusher service into our application. It involves several steps to set up real-time messaging and notifications. Here's a step-by-step guide on how to configure Pusher in a Laravel app,

1. Create a Pusher Account:

- Go to the Pusher website and sign up for an account if you don't have one already. <https://pusher.com>
- Once you're logged in, create a new app in your Pusher dashboard.
- Then You will find an app credentials, which will need for configuration.

2. Configure API .env variables

Open Pickbazar `api` .env file and add the following Pusher configuration variables with the credentials from your Pusher dashboard,

```
BROADCAST_DRIVER=pusher

PUSHER_APP_ID=your_app_id
PUSHER_APP_KEY=your_app_key
PUSHER_APP_SECRET=your_app_secret
PUSHER_APP_CLUSTER=your_app_cluster
```

3. Configure Admin .env variables

Open Pickbazar `admin` .env file and add the following Pusher configuration variables. You also get example from .env.example

```
# pusher config
NEXT_PUBLIC_PUSHER_DEV_MOOD=true
NEXT_PUBLIC_PUSHER_APP_KEY='4ef94e52833c9542eadc'
NEXT_PUBLIC_PUSHER_APP_CLUSTER='ap2'
NEXT_PUBLIC_BROADCAST_AUTH_URL='http://127.0.0.1:8000/broadcasting/
auth'
```

Then add this Channel variables too below that.

```
# Channel name from PHP
NEXT_PUBLIC_STORE_NOTICE_CREATED_CHANNEL_PRIVATE=private-
store_notice.created
NEXT_PUBLIC_ORDER_CREATED_CHANNEL_PRIVATE=private-order.created
NEXT_PUBLIC_MESSAGE_CHANNEL_PRIVATE=private-message.created

# Event name from PHP
NEXT_PUBLIC_STORE_NOTICE_CREATED_EVENT=store.notice.event
NEXT_PUBLIC_ORDER_CREATED_EVENT=order.create.event
NEXT_PUBLIC_MESSAGE_EVENT=message.event
```

Now, your application is configured with pusher.

Tips : For this real-time notifications feature, you must run event queues.

Configure settings

After configuring pusher, now it's time to configure settings from `admin dashboard`. A super admin can enable which real time notification settings he want to keep enable in his site. Settings can be found inside `Settings > Event settings` menu.

1. Configuration settings page

Events Settings

Realtime notification setting
Set this to get notification in realtime

Select notification options
Admin & Vendor : Order, Admin & Vendor : Message, Admin & Vendor : Store Notice

SMS Event Setting
Set This to Send SMS on Selected Event.

Select SMS Options
Select...

Email Event Setting
Set This to Send SMS on Selected Event.

Select Email Options
Select...

Save Settings

1. All notifications page

All notifications

Customer Created a new order: #20230920913800
Sep 20, 2023 at 09:32 AM

Customer Created a new order: #20230920504469
Sep 20, 2023 at 09:33 AM

Store Owner Created a new order: #1
Sep 20, 2023 at 09:35 AM

Customer Created a new order: #20230920131961
Sep 20, 2023 at 09:38 AM

Customer Created a new order: #20230920950274
Sep 20, 2023 at 09:40 AM

Customer Created a new order: #20230920612904
Sep 20, 2023 at 10:41 AM

Filter By Notification Type

Key notes

- After activating the settings, notification panel will be available in app Header section.
- Currently only Super Admin & Vendors are inside this feature criteria.
- Clicking on single notify-log, it will redirected to it's targetted page.
- Store notice real-time notification is available only in Vendor end.

- Order real-time notification is splitted & redirected to it's targetted user. For example, super admin will receive only Parent order notifications. Child order notifications will be redirected to respected shop owners.

Multilingual

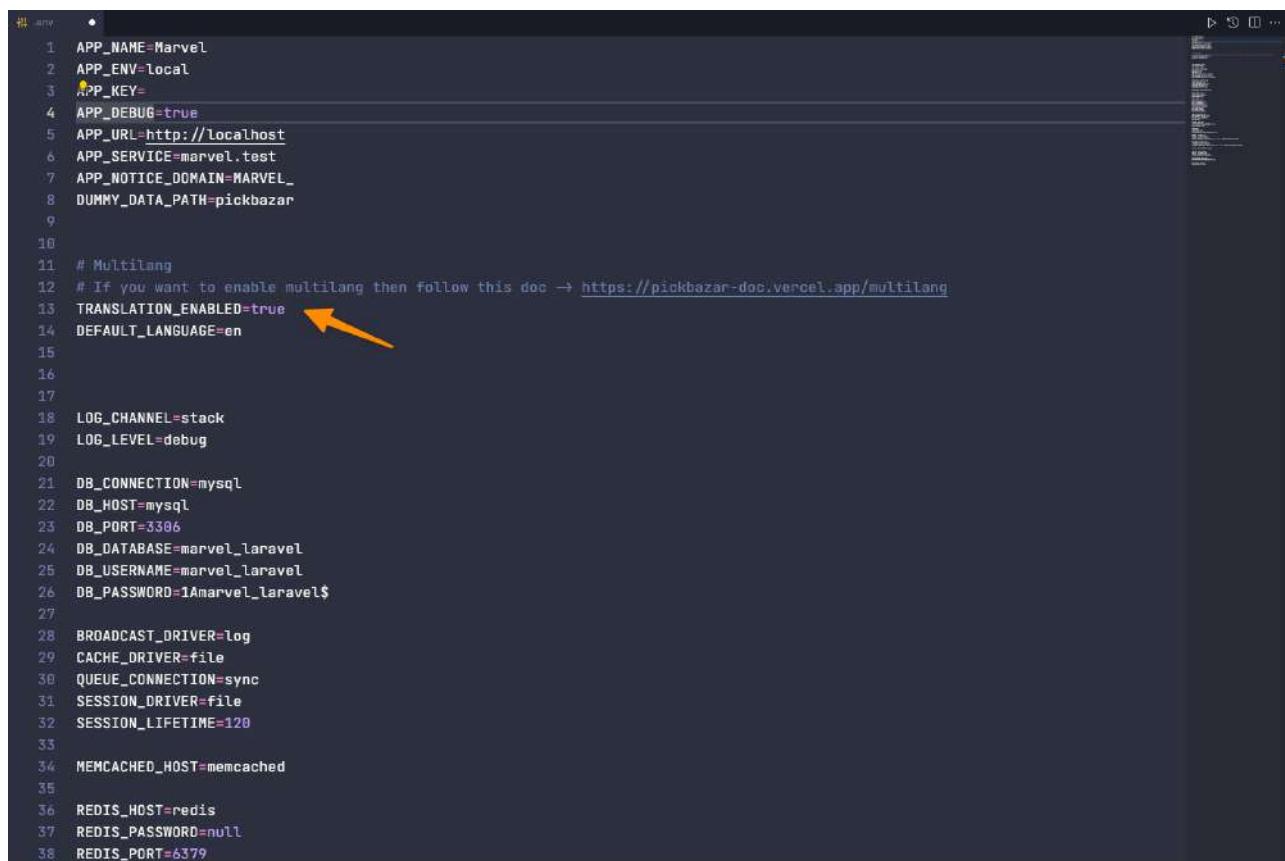
If your business or site doesn't require multiple languages, then we don't recommend enabling the multilanguage feature. It'll increase the complexity of maintaining all the language simultaneously. Only enable it if you know what multilanguage is and your business or site needs that feature.

Step 1: Enable Multilingual Feature

API:

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

```
TRANSLATION_ENABLED=true
```



```

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

```

Admin

For admin,

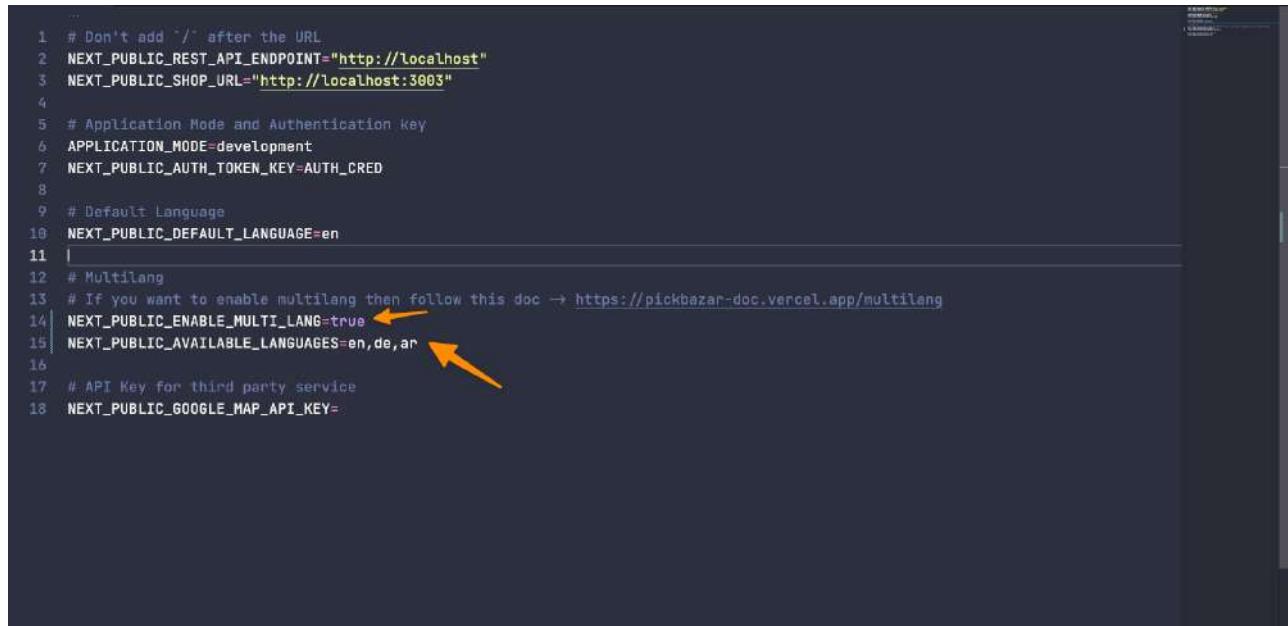
update `admin -> rest -> .env` and update,

```
NEXT_PUBLIC_ENABLE_MULTI_LANG=true
```

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

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

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



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

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

Shop

For shop, similarly,

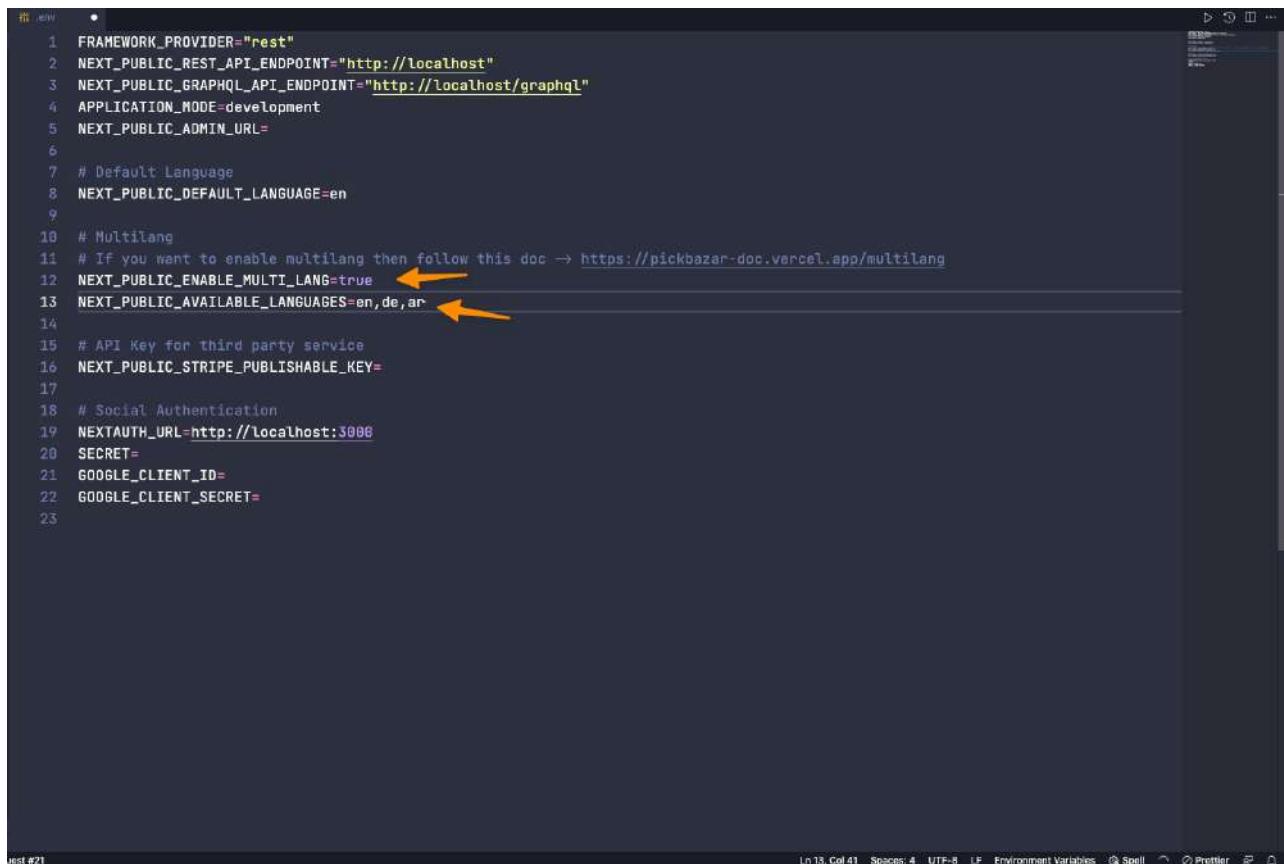
update `shop -> .env` and update,

```
NEXT_PUBLIC_ENABLE_MULTI_LANG=true
```

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

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

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



```
1 FRAMEWORK_PROVIDER="rest"
2 NEXT_PUBLIC_REST_API_ENDPOINT="http://localhost"
3 NEXT_PUBLIC_GRAPHQL_API_ENDPOINT="http://localhost/graphql"
4 APPLICATION_MODE=development
5 NEXT_PUBLIC_ADMIN_URL=
6
7 # Default Language
8 NEXT_PUBLIC_DEFAULT_LANGUAGE=en
9
10 # Multilang
11 # If you want to enable multilang then follow this doc → https://pickbazar-doc.vercel.app/multilang
12 NEXT_PUBLIC_ENABLE_MULTI_LANG=true ←
13 NEXT_PUBLIC_AVAILABLE_LANGUAGES=en,de,ar ←
14
15 # API Key for third party service
16 NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=
17
18 # Social Authentication
19 NEXTAUTH_URL=http://localhost:3008
20 SECRET=
21 GOOGLE_CLIENT_ID=
22 GOOGLE_CLIENT_SECRET=
23
```

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

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

Step 2: How to add new language in admin?

The screenshot shows two instances of Visual Studio Code. The top instance displays the file `multilingual.mdx` with the following content:

```
1 # Secrets like the one below should go into ".env.local" instead to avoid pushing
2 # them to a repository.
3 # NEXT_PUBLIC_ENABLE_MULTI_LANG value true or blank
4 NEXT_PUBLIC_REST_API_ENDPOINT="http://localhost:8000"
5 NEXT_PUBLIC_SHOP_URL="http://localhost:3003"
6 NEXT_PUBLIC_GOOGLE_MAP_API_KEY=
7 NEXT_PUBLIC_DEFAULT_LANGUAGE=en
8 NEXT_PUBLIC_AVAILABLE_LANGUAGES=en,de,sg
9 APPLICATION_MODE=development
10 NEXT_PUBLIC_AUTH_TOKEN_KEY=AUTH_CRED
11 NEXT_PUBLIC_ENABLE_MULTI_LANG=true
```

The bottom instance shows the file structure of the `marvel-admin` directory. A red box highlights the `locales` folder under the `public` directory. Red arrows point from the `locales` folder to the `.env` file in both instances, indicating where to copy and paste the folder.

Copy and paste the folder.

Make sure copy and paste en folder

File Location

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

File Location

admin\rest\src\utils\locals.tsx

```

1 "text-read-more": "Read more",
2 "text-less": "Less",
3 "text-or": "Or",
4 "description": "Track Order",
5 "error-heading": "Error code: 404",
6 "nav-menu-track-order": "Track Order",
7 "nav-menu-offer": "Offers",
8 "nav-menu-faq": "FAQ",
9 "nav-menu-contact": "Contact",
10 "user-avatar": "User Avatar",
11 "auth-menu-profile": "Contact",
12 "auth-menu-checkout": "Checkout",
13 "auth-menu-my-orders": "My Orders",
14 "auth-menu-logout": "Logout",
15 "join-button": "Join",
16 "change-locale": "Switch to German",
17 "admin-login-title": "Login to Admin",
18 "admin-register-title": "Register New Account",
19 "billing-address": "Billing Address",
20 "shipping-address": "Shipping Address",
21 "no-order-found": "No Order Found",
22 "order-sub-total": "Sub Total",
23 "order-tax": "Tax",
24 "order-delivery-fee": "Delivery Fee",
25 "order-discount": "Discount",
26 "order-total": "Total",
27 "signing-out-text": "Signing Out...",
28 "sale-history": "Sale History",
29 "january": "January",
30 "february": "February",
31

```

In 10, Col 1 · Spaces: 2 · UTF-8 · ORF · JSON · Go Live · Spell · Prettier

File Location

admin\rest\src\utils\locals.tsx

```

1 const flags = [
2   {
3     id: 'de',
4     name: 'Deutsch',
5     value: 'de',
6     icon: <DEFlag width="20px" height="15px" />
7   },
8   {
9     id: 'he',
10    name: 'עברית',
11    value: 'he',
12    icon: <ILFlag width="20px" height="15px" />
13  },
14  {
15    id: 'es',
16    name: 'Español',
17    value: 'es',
18    icon: <ESFlag width="20px" height="15px" />
19  },
20  {
21    id: 'sg',
22    name: 'Singapore',
23    value: 'sg',
24    icon: <SGFlag width="20px" height="15px" />
25  }
];

```

You, 3 minutes ago In 8, Col 56 (5 selected) Spaces: 2 · UTF-8 · ORF · TypeScript React · Go Live · Spell · TSLint · Prettier

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

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** On the left, it displays a tree view of files and folders. The root folder is 'multilingual'. Inside 'multilingual', there are subfolders like 'multilingual.mdx', 'localstx.M', 'SGFlag.jsx', 'translation.mdx', and 'common.mdx'. Under 'multilingual.mdx', there are further subfolders: 'pickbazar', 'marvel-admin', 'test', 'src', 'components', 'icons', and 'flags'. The 'flags' folder contains several files: 'CHFlag.jsx', 'DEFlag.jsx', 'ESFlag.jsx', 'FRFlag.jsx', 'SAFlag.jsx', 'SGFlag.jsx', and 'USTFlag.jsx'.
 - Code Editor:** The main editor area shows the content of the 'SGFlag.jsx' file. The code is an SVG definition for a flag icon, starting with the following lines:

```
1 export const SgFlag = ({ width = '640px', height = '480px' }) => {
```
 - Status Bar:** At the bottom, the status bar shows 'In 1, Col 20' and other standard status indicators.

Step 3: How to add new language in shop?

```
multilingual.mdx pickbazar-doc+inc M .env - Untitled (Workspace) - Visual Studio Code
pickbazar > pickbazar-shop > .env
1 FRAMEWORK_PROVIDER=rest
2 NEXT_PUBLIC_REST_API_ENDPOINT=http://localhost:8000
3 NEXT_PUBLIC_GRAPHQL_API_ENDPOINT=http://localhost:8000/graphql
4 NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=
5 NEXT_PUBLIC_ADMIN_URL=http://localhost:3002
6 NEXT_PUBLIC_DEFAULT_LANGUAGE=en
7 NEXT_PUBLIC_AVAILABLE_LANGUAGES=en,de,sg
8 NEXT_PUBLIC_ENABLE_MULTI_LANG=true
9
10 # Social Authentication
11 SECRET=c8245e4a46c398b6e2241a8070e10b71
12 GOOGLE_CLIENT_ID=131043867871-nc3nff7qtu4lkr1hb6hi6rfcud53m2f2.apps.googleusercontent.com
13 GOOGLE_CLIENT_SECRET=GLuPrBZY9Jun0IkbbbQ961lw
```

```
multilingual.mdx pickbazar-doc+inc M .env - Untitled (Workspace) - Visual Studio Code
pickbazar > pickbazar-shop > .env
1 FRAMEWORK_PROVIDER=rest
2 NEXT_PUBLIC_REST_API_ENDPOINT=http://localhost:8000
3 NEXT_PUBLIC_GRAPHQL_API_ENDPOINT=http://localhost:8000/graphql
4 NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=
5 NEXT_PUBLIC_ADMIN_URL=http://localhost:3002
6 NEXT_PUBLIC_DEFAULT_LANGUAGE=en
7 NEXT_PUBLIC_AVAILABLE_LANGUAGES=en,de,sg
8 NEXT_PUBLIC_ENABLE_MULTI_LANG=true
9
10 # Social Authentication
11 SECRET=c8245e4a46c398b6e2241a8070e10b71
12 GOOGLE_CLIENT_ID=131043867871-nc3nff7qtu4lkr1hb6hi6rfcud53m2f2.apps.googleusercontent.com
13 GOOGLE_CLIENT_SECRET=GLuPrBZY9Jun0IkbbbQ961lw
```

File Location: shop\public\locales\sg\banner.json

A screenshot of Visual Studio Code showing a JSON file named `common.json`. The file contains various key-value pairs for static text, such as "404-heading", "404-sub-heading", "404-back-home", "nav-menu-track-order", "nav-menu-offer", "nav-menu-faq", "nav-menu-contact", "user-avatar", "auth-menu-profile", "auth-menu-checkout", "auth-menu-my-orders", "auth-menu-logout", "join-button", "change-locale", "text-search", "text-burger-menu", "text-close", "text-or", "text-terms", "text-policy", "text-accept", "cookie-message", "text-minus", "text-plus", "text-copy", "text-copied", "text-expired", "text-card-number", and "text-read-more". A red box highlights the text "Translate the static text here" at the top right of the code editor.

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
```

Translate the static text here

File Location

shop\src\lib\locals.tsx

A screenshot of Visual Studio Code showing a TypeScript file named `locals.tsx`. The file defines a class with methods for getting reviews, addresses, products, strings, and preview images. It also includes a `locals` method that returns an array of language objects. Each object has properties: `id`, `name`, `value`, and `icon` (using `<Flag>` or `<FlagRound>` components). A red box highlights the text "Set flag here for lang switcher" near the icon definitions. The code editor shows syntax highlighting for TypeScript and React components.

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
```

Set flag here for lang switcher

File Location

shop\src\components\icons\flags\

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a tree view of files and folders. In the 'multilingual' folder, there are sub-folders like 'CNFlag', 'CNFlagRound', 'DEFlag', 'DEFlagRound', 'ESFlag', 'ESFlagRound', 'FRFlag', 'FRFlagRound', 'USFlag', and 'USFlagRound'. There are also 'groups', 'social', and 'flags' sections.
- Editor:** Displays an SVG file named 'SgFlag.tsx'. The code defines a flag icon with a width of 640px and height of 480px, using a specific path and stroke width.
- Bottom Status Bar:** Shows 'In 1. Col 16 Tab Size: 2 UTF-8 CRLF JSON Go Live Spell TSLint Prettier'.

Step 3: How to translate static content for admin?

File Location

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

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure. The 'common.json' file is located in the 'de' folder under 'public\locales'. Other files like 'banner.json', 'form.json', 'table.json', and 'widgets.json' are also visible.
- Editor:** Displays the content of 'common.json'. It contains various German translations such as 'Weiterlesen', 'Weniger', 'Oden', 'Bestellung verfolgen', 'Fehlercode: 404!', 'Bestellung verfolgen', 'Bietet an', 'FAQ', 'Kontakt german', 'Benutzer Avatar', 'Kontakt', 'Überprüfen', 'meine Bestellungen', 'Ausloggen', 'Beitreten', 'Wechseln Sie zu Deutsch', 'Melden Sie sich bei admin an', 'Rechnungsadresse', 'Lieferanschrift', 'Kein Bestellung gefunden', 'Zwischensumme', 'MwSt', 'Liefergebühr', 'Rabatt', 'Gesamt', 'Abmelden...', 'Verkaufsgeschichte', 'Januar', 'Februar', etc.
- Bottom Status Bar:** Shows 'In 10. Col 42 Spaces: 2 UTF-8 CRLF JSON Go Live Spell TSLint Prettier'.

Step 4: How to translate static content for shop?

File Location

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

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure:
 - multilingual.mdx
 - common.json (highlighted)
 - translation.mdx
 - .env
 - public (highlighted)
 - locales (highlighted)
 - de
 - behind.json (highlighted)
 - common.json (highlighted)
 - faq.json
 - policy.json
 - terms.json
 - en
 - es
 - fr
 - zh
 - payment
 - promotion
 - favicon.ico
 - manifest.json
 - master.png
 - mockup-img.png
- Editor:** The content of the common.json file is displayed:

```
1 "404-heading": "Fehlercode 404",
2 "404-sub-heading": "Hoppa! Sieht so aus, als wäre dies keine Seite",
3 "404-back-home": "Bring mich nach Hause",
4 "nav-menu-track-order": "Reihenfolge verfolgen",
5 "nav-menu-offer": "Angebote",
6 "nav-menu-faq": "FAQ",
7 "nav-menu-contact": "Kontakt",
8 "user-avatar": "Benutzer-Avatar",
9 "auth-menu-profile": "Profil",
10 "auth-menu-checkout": "Kasse",
11 "auth-menu-my-orders": "Meine Bestellungen",
12 "auth-menu-logout": "Abmelden",
13 "join-button": "Beitreten",
14 "change-locale": "Zu Deutsch wechseln",
15 "text-search": "Suchen",
16 "text-burger-menu": "Burger-Menu",
17 "text-close": "Schließen",
18 "text-or": "Oder",
19 "text-terms": "Begriffe",
20 "text-policy": "Richtlinien",
21 "text-accept": "Akzeptieren",
22 "cookie-message": "Diese Website verwendet Cookies, um Ihre Erfahrung zu verbessern. Durch Klicken auf erklären Sie sich mit unserer",
23 "text-privacy-policy": "Datenschutzrichtlinie",
24 "text-minus": "Minus",
25 "text-plus": "Plus",
26 "text-copy": "Kopie",
27 "text-copied": "Kopiert!",
28 "text-expired": "Abgelaufen",
29
```
- Bottom Status Bar:** Shows file information like Line 2, Col 2, Spaces: 2, UTF-8, CR LF, JSON, Go Live, 100 Spell, Prettier.

Step 5: How to translate content?

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

Data Type

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

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

For example,

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

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

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

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

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

Translate Settings

If you want to translate the settings,

just go to your admin -> settings

It'll open default language settings.

The screenshot shows the PickBazar admin dashboard. On the left is a sidebar with various menu items: Dashboard, Shops, My Shops, Products, Attributes, Groups, Categories, Tags, Manufacturers/Publications, Authors, Orders, Order Status, Create Order, Users, Coupons, Taxes, Shipments, Withdrawals, and Refunds. The 'Settings' option is selected. The main content area has a title 'Default Language'. It includes a 'Logo' section with a placeholder for a site logo and a file upload input. Below this is a 'Information' section with fields for 'Site Title' (set to 'Pickbazar'), 'Site Subtitle' (set to 'Your next ecommerce'), 'Currency' (set to 'US Dollar'), and 'Minimum Order Amount' (set to '0'). At the top right of the main content area, there is a dropdown menu for language selection, currently set to 'English'. An orange arrow points to this language dropdown.

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

PickBazar

- Dashboard
- Shops
- My Shops
- Products
- Attributes
- Groups
- Categories
- Tags
- Manufacturers/Publications
- Authors
- Orders
- Order Status
- Create Order
- Users
- Coupons
- Taxes
- Shipments
- Withdrawals
- Refunds

Settings

Logo
Upload your site logo from here.
Dimension of the logo should be 128x40 Pixel.

Information
Change your site information from here

Site Title	Pickbazar
Site Subtitle	Your next ecommerce
Currency	US Dollar
Minimum Order Amount	0

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

Translate Order Status

To translate the order status,

1. Go to the order status page from the admin
2. Then click **Three Dot** and Create translate,

PickBazar

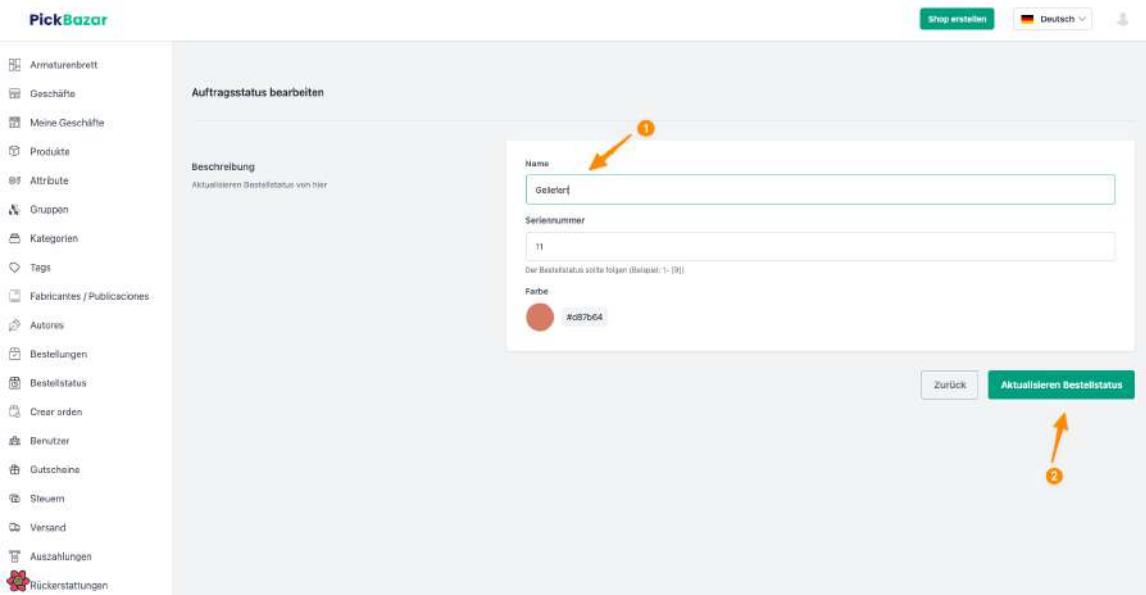
- Dashboard
- Shops
- My Shops
- Products
- Attributes
- Groups
- Categories
- Tags
- Manufacturers/Publications
- Authors
- Orders
- Order Status**
- Create Order
- Users
- Coupons
- Taxes
- Shipments
- Withdrawals
- Refunds

Order Status

ID	Name	Serial	Action
7	Delivered	11	
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	

Create +

3. Then translate the text and click Save.



Similarly, translate all the order status.

Translate Group, Author, Manufacturer, Tags, Categories:

Follow order status procedure to translate **Group**, **Author**, **Manufacturer**, **Tags**, and **Categories**.

Translate Products

Translate Simple Products:

Similarly, just edit three dots and translate the product.

*But during the translation, don't change the **SKU**. SKU for a product has to be similar for all languages. We calculate the quantity based on that field.*

- Armaturenrett
- Geschäfte
- Meine Geschäfte
- Produkte
- Attribute
- Gruppen
- Kategorien
- Tags
- Fabricantes / Publicaciones
- Autoren
- Bestellungen
- Bestellstatus
- Crear orden
- Benutzer
- Gutscheine
- Steuern
- Versand
- Auszahlungen
- Rückerstattungen

Don't change
SKU during the
translation

The screenshot shows a product update form. On the left, there's a sidebar with various shop management options. The main area has fields for quantity (Menge) set to 500, article number (Artikelnummer) containing a long, random string of letters and numbers, width (Breite), height (Höhe), and length (Länge). There are checkboxes for 'Ist digital' (checked) and 'Ist extern'. Below these is a file upload section with a placeholder 'Bild hochladen oder Drag & Drop' and extensions 'PNG, JPG'. At the bottom right are 'Zurück' and 'Produkt aktualisieren' buttons.

Translate Variable Product:

For variable products, you must have to translate attributes first. Without attributes, you can't get an attribute list during the translation.

Click on **Three Dots** to translate the item.

But for the variable product, you must create the variation for translation again.

Translate Attributes

PickBazar

Dashboard
Shops
My Shops
Products
Attributes
Groups
Categories
Tags
Manufacturers/Publications
Authors
Orders
Order Status
Create Order
Users
Coupons
Taxes
Shipments
Withdrawals

Attributes

ID	Name	Shop	Values	Actions
6	Book Type	Books Shop	Hardcover Book, Picture Book, Paperback Book, PDF Book	
5	Language	Books Shop	English, Arabic, Spanish, French, Hindi, Hebrew	
4	Size	Clothing Shop	S, M, L, XL	
3	Color	Clothing Shop	Red, Blue, White	

PickBazar

Armaturenbrett
Geschäfte
Meine Geschäfte
Produkte
Attribute
Gruppen
Kategorien
Tags
Fabricantes / Publicaciones
Autores
Bestellungen
Bestellstatus
Crear orden
Benutzer
Gutscheine:
Steuern
Versand
Auszahlungen

Attributwerte
Aktualisieren Ihr Attributwert und die erforderlichen Informationen von hier.

Wert	Meta	Aktionen
Gebundenes Buch		
Bilderbuch		
Taschenbuch		
PDF-Buch		

Wert hinzufügen

Zurück **Aktualisieren Attribut**

Translate Groups

PickBazar

Dashboard Shops My Shops Products Attributes Groups Categories Tags Manufacturers/Publications Authors Orders Order Status Create Order Users Coupons Taxes Shipments

[https://www.pickbazar.com/groups/books/translates](#)

Groups

Type your query and press enter

+ Add Group

ID	Name	Icon	Actions
8	Books	book	...
7	Daily Needs	apple	...
6	Furniture	chair	...
5	Clothing	clothes	...
4	Bags	bag	...
3	Makeup	woman	...
2	Bakery	bread	...
1	Grocery	grocery	...

Create Deutsch

PickBazar

Armaturenbrett Geschäfte Meine Geschäfte Produkte Attribute Gruppen Kategorien Tags Fabricantes / Publicaciones Autores Bestellungen Bestellstatus Clear orden Benutzer Gutscheine Steuern Versand Auszahlungen

[https://www.pickbazar.com/groups/books/translates](#)

Banner
Fügen Sie hier Ihr Bannerbild mit Titel und Beschreibung hinzu. Die Größe des Banners sollte 1920 x 1080 px für das Vollbild-Banner und 1500 x 450 px für das kleine Banner betragen.

Banner 1

Entfernen

Titel: Buchbanner

Beschreibung: Dies ist das Buch-Demo-Banner

Galerie

Bild hochladen oder Drag & Drop
PNG, JPG



Banner hinzufügen

Zurück Gruppe aktualisieren

Translate Variable Products

PickBazar

- Dashboard
- Shops
- My Shops
- Products
- Attributes
- Groups
- Categories
- Tags
- Manufacturers/Publications
- Authors
- Orders
- Order Status
- Create Order
- Users
- Coupons
- Taxes
- Shipments

Products

Image	Name	Group	Shop	Product Type	Price/Unit	Quantity	Status	Actions
	Blood Drinker	Books	Books Shop	variable	\$180.00 - \$200.00	1000	published	trash edit more
	Forest Killer	Books	Books Shop	simple	\$260.00	500	published	trash edit more
	Flesh Eater	Books	Books Shop	simple	\$250.00	500	published	trash edit more
	The Boneyard Man	Books	Books Shop	variable	\$300.00 - \$300.00	1000	published	trash edit more
	The Psycho Killer First Part	Books	Books Shop	simple	\$200.00	500	published	trash edit more
	The Serial Killer	Books	Books Shop	simple	\$200.00	500	published	trash edit more
	Before The Ruins	Books	Books Shop	simple	\$200.00	500	published	trash edit more
	Eight Perfect Murders	Books	Books Shop	simple	\$200.00	500	published	trash edit more

PickBazar

- Armaturenbrett
- Geschäfte
- Meine Geschäfte
- Produkte
- Attribute
- Gruppen
- Kategorien
- Tags
- Fabricantes / Publicaciones
- Autores
- Bestellungen
- Bestellstatus
- Crear orden
- Benutzer
- Gutscheine
- Steuern
- Versand
- Auszahlungen

Gruppe & Kategorien
Wählen Sie hier Produktgruppe und Kategorien aus.

Beschreibung
Bearbeiten Ihre Produktbeschreibung und die notwendigen Informationen finden Sie hier

Gruppe*
Bücher

Kategorien
Thriller

Autoren
Katy Iery

Fabricantes
Factory-Veröffentlichung

Tags
Combo

Name*
Blut-Trinker

Einhheit*

PickBazar

- Armaturenbrett
- Geschäfte
- Meine Geschäfte
- Produkte
- Attribute
- Gruppen
- Kategorien
- Tags
- Fabricantes / Publicaciones
- Autoren
- Bestellungen
- Bestellstatus
- Crear orden
- Benutzer
- Gutscheine
- Steuern
- Versand
- Auszahlungen

Produktart
Wählen Sie hier das Produkttypformular aus.

Informationen zu Produktvariationen
Aktualisieren Ihre Produktvariation und die erforderlichen Informationen von hier

Einhalt*
1 pc

Beschreibung:
Cerebral suspenser is a kidney combining the suspenser and cerebral fabrication stripes. It's generally used to describe literature or flicks that deal with cerebral narratives in a suspenser or thrilling setting.
In terms of environment and convention, it's a subgenre of the broader ranging suspenser narrative structure, with parallels to Gothic and operatic fabrication in the sense of occasionally having a "dissolving sense of reality". It's frequently told through the standpoint of psychologically stressed characters, often dealing with their distorted internal perceptions and reactions on the outside, and frequently involving mysterious, horrific, and sometimes violent events.

Status
Veröffentlicht Entwurf

Produktart
Variable

OPTIONEN

Option hinzufügen

Zurück Produkt aktualisieren

- [Armaturenbrett](#)
- [Geschäfte](#)
- [Meine Geschäfte](#)
- [Produkte](#)
- [Attribute](#)
- [Gruppen](#)
- [Kategorien](#)
- [Tags](#)
- [Fabricantes / Publicaciones](#)
- [Autoren](#)
- [Bestellungen](#)
- [Bestellstatus](#)
- [Crear orden](#)
- [Benutzer](#)
- [Gutscheine](#)
- [Steuern](#)
- [Versand](#)
- [Auszahlungen](#)

Optionen 1

Attributname*	Attributwert*
Buchtyp	Gebundenes Buch

Optionen 2

Attributname*	Attributwert*
Sprache	English *

[Option hinzufügen](#)

1 VARIANTE

Variante: Gebundenes Buch/English

Preis*	Verkaufspreis
200	180

Artikelnummer*	Menge*
51dg65sd1g65g4aB9ab65b	500

Stellen Sie sicher, dass die SKU für alle Sprachen identisch ist.

Bild

Make sure translated variable product variation value same like as default language example (attribute name, attribute value, price, sale price, quantity and SKU).

PickBazar

[Create Shop](#) | [English](#) | [Logout](#)

[Dashboard](#)

[Shops](#)

[My Shops](#)

[Products](#)

[Attributes](#)

[Groups](#)

[Categories](#)

[Tags](#)

[Manufacturers/Publications](#)

[Authors](#)

[Orders](#)

[Order Status](#)

[Create Order](#)

[Users](#)

[Coupons](#)

[Taxes](#)

[Shipments](#)

[Withdrawals](#)

Product Variation Information
Update your product variation and necessary information from here .

OPTIONS

Options 1

Attribute Name*	Attribute Value*	Remove
Book Type	Hardcover Book *	X

Options 2

Attribute Name*	Attribute Value*	Remove
Language	English * Arabic * French * Spanish *	X

[Add an option](#)

4 VARIANT

Variant: Hardcover Book/English

Price*	Sale Price
200	180

SKU* **Quantity***

51dg5sd1g55g489cb65b	500
----------------------	-----

[Image](#)

PickBazar

[Create Shop](#) | [Singapore](#) | [Logout](#)

[Papan pemuka](#)

[Shops](#)

[My Shops](#)

[Produk](#)

[Atribut](#)

[Groups](#)

[Categories](#)

[Tags](#)

[Manufacturers/Publications](#)

[Authors](#)

[Orders](#)

[Order Status](#)

[Create Order](#)

[Users](#)

[Coupons](#)

[Taxes](#)

[Shipments](#)

[Withdrawals](#)

Product Variation Information
Update your product variation and necessary information from here .

OPTIONS

Options 1

Attribute Name*	Attribute Value*	Remove
Jenis Buku	Buku Kulit Keras *	X

Options 2

Attribute Name*	Attribute Value*	Remove
Bahasa	English * Arabic * French * Spanish *	X

[Add an option](#)

4 VARIANT

Variant: Buku Kulit Keras/English

Price*	Sale Price
200	180

SKU* **Quantity***

51dg5sd1g55g489cb65b	500
----------------------	-----

[Image](#)

Step 6: How to add custom or solved broken font issue?

Invoice No: LpDNRSepc3MU

Date: 7th July, 2022

Pobon

pobon@redq.io
19365141641631
2148 Stratford Park
Winchester
KY
40391
United States

Pickbazar

<https://redq.io>
+129290122122
NY State Thruway, New York, USA

Products	Quantity	Total
ମୁଣ୍ଡା ମୁଣ୍ଡା ମୁଣ୍ଡା ମୁଣ୍ଡା	1	\$5.00

Sub total : \$5.00
Discount : \$0.00
Tax : \$0.10
Delivery fee : \$50.00
Total : **\$55.10**

File Location

api\config\pdf.php

```
'font_path' => base_path('resources/fonts/'),
'font_data' => [
    'example' => [
        'R' => 'example.ttf',      // regular font
        'B' => 'example.ttf',      // optional: bold font
        'I' => 'example.ttf',      // optional: italic font
        'BI' => 'example.ttf', // optional: bold-italic font
        'useOTL' => 0xFF,
        'useKashida' => 75,
    ]
],
```

```
<?php

return [
    'mode' => 'utf-8',
    'format' => 'A4',
    'author' => '',
    'subject' => '',
    'keywords' => '',
    'creator' => 'Laravel Pdf',
    'display_mode' => 'fullpage',
    'tempDir' => storage_path('app/temp'), // base_path(storage_path('app/public')),
    'pdf_a' => false,
    'pdf_a_auto' => false,
    'icc_profile_path' => '',
    'font_path' => base_path('resources/fonts/'),
    'font_data' => [
        'example' => [
            'R' => 'example.ttf', // regular font
            'B' => 'example.ttf', // optional: bold font
            'I' => 'example.ttf', // optional: italic font
            'BI' => 'example.ttf', // optional: bold-italic font
            'useOTL' => 0xFF,
            'useKashida' => 75,
        ]
    ],
];
```

File Location

api\resources\fonts\

```
<?php

return [
    'mode' => 'utf-8',
    'format' => 'A4',
    'author' => '',
    'subject' => '',
    'keywords' => '',
    'creator' => 'Laravel Pdf',
    'display_mode' => 'fullpage',
    'tempDir' => storage_path('app/temp'), // base_path(storage_path('app/public')),
    'pdf_a' => false,
    'pdf_a_auto' => false,
    'icc_profile_path' => '',
    'font_path' => base_path('resources/fonts/'),
    'font_data' => [
        'example' => [
            'R' => 'example.ttf', // regular font
            'B' => 'example.ttf', // optional: bold font
            'I' => 'example.ttf', // optional: italic font
            'BI' => 'example.ttf', // optional: bold-italic font
            'useOTL' => 0xFF,
            'useKashida' => 75,
        ]
    ],
];
```

File Location

api\packages\marvel\stubs\resources\views\pdf\order-invoice.blade.php
api\resources\views\pdf\order-invoice.blade.php

```

@if($language === "ex")
<style type="text/css">
    body {
        font-family: Arial, sans-serif, example;
    }
</style>
@endif

```

File Edit Selection View Go Run Terminal Help

order-invoice.blade.php - Untitled (Workspace) - Visual Studio Code

pickbazar > marvel > packages > marvel > stubs > resources > views > pdf > order-invoice.blade.php > head

order-invoice.blade.php M example.ttf U env _yaml localstax M order-invoice.blade.php M env pickedash-pickbazar-shop banner.js

pickbazar > marvel > packages > marvel > stubs > resources > views > pdf > order-invoice.blade.php > head

You 3 seconds ago | 3 authors (MD Tareq Mahmud and others)

```

1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6      <title>Order Invoice</title>
7      <style type="text/css">
8          MD Tareq Mahmud, yesterday | 2 authors (Roman Uli Ferdosh and others)
9          body {
10              font-family: Arial, sans-serif;
11          }
12      </style>
13      @if($language === "ex")
14          You, now + uncommitted changes
15          body {
16              font-family: Arial, sans-serif, example;
17          }
18      </style>
19      <endif>
20  </head>
21  <body>
22      <php
23          $contactDetails = $settings->options['contactDetails'];
24          $customer = $order->customer;
25          $shippingAddress = $order->shipping_address;
26          $products = $order->products;
27          $settings = $settings->options;
28          $authorDetails = $settings['contactDetails'];
29          $authorLocation = $authorDetails['location'];

```

You, now | Line 12, Col 28 | Spaces: 4 | UTF-8 | CRLF | Blade | Go Live | Spell | Prettier



Stack trace Request App User Context Debug Share ↗

Mpdf\TTFontFile::__getGDEFTables
E:\Server\www\wordpress\redq\laravel\pickbazar-video\pickbazar-laravel\api\vendor\mpdf\mpdf\src\TTFontFile.php:1355

45 Mpdf\TTFontFile 1340 \$markSetTableFormat = \$this->read_ushort();
1341 \$markSetCount = \$this->read_ushort();
44 Mpdf\TTFontFile 1342 \$markSetOffset = [];
1343 for (\$i = 0; \$i < \$markSetCount; \$i++) {
43 Mpdf\TTFontFile 1344 \$markSetOffset[] = \$this->read_ulong();
1345 }
1346 for (\$i = 0; \$i < \$markSetCount; \$i++) {
1347 \$this->seek(\$markSetOffset[\$i]);
1348 \$glyphs = \$this->_getCoverage();
1349 \$this->MarkGlyphSets[\$i] = '' . implode(' ', \$glyphs);
42 Mpdf\Fonts\MetricsGenerator 1350
1348
1349
41 Mpdf\Mpdf 1350
1351 } else {
40 Mpdf\Mpdf 1352 \$this->MarkGlyphSets = [];
1353 }
39 Mpdf\Mpdf 1354 } else {
38 Mpdf\Mpdf 1355 throw new \Mpdf\Exception\FontException(sprintf('Unable to set font "%s" to use OTL as'));
1356 }

If you are facing above the issue just replace these code.

File Location

api\config\pdf.php

```
'font_path' => base_path('resources/fonts/'),
'font_data' => [
    'example' => [
        'R' => 'example.ttf',      // regular font
        'B' => 'example.ttf',      // optional: bold font
        'I' => 'example.ttf',      // optional: italic font
        'BI' => 'example.ttf', // optional: bold-italic font
        // 'useOTL' => 0xFF,
        'useKashida' => 75,
    ],
],
```

If you face any issues with Multilingual, or you've any suggestions to improve the Multilingual feature, then please open a ticket at (<https://redqsupport.ticksy.com/>) [<https://redqsupport.ticksy.com/>]

Translation

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

Existing Language

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

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

For Rest

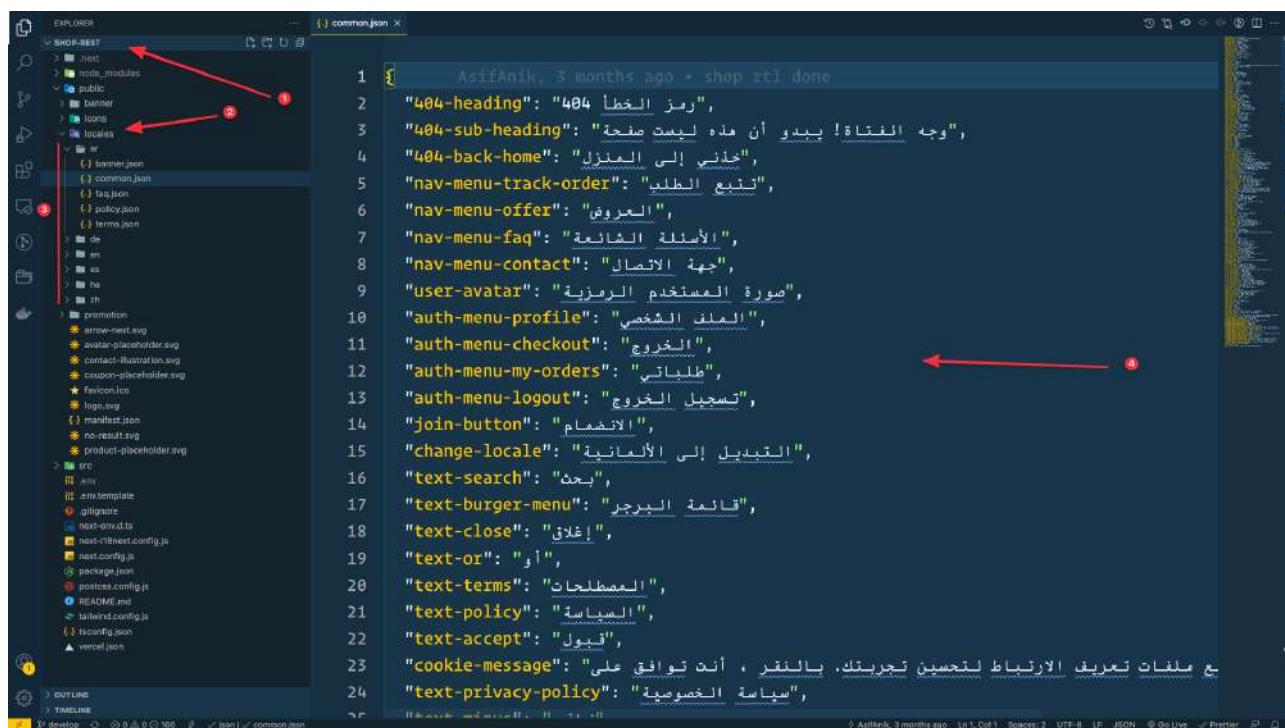
```
pickbazar-laravel -> shop -> public -> locales -> YOUR LANGUAGE
```

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

For GraphQL

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

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

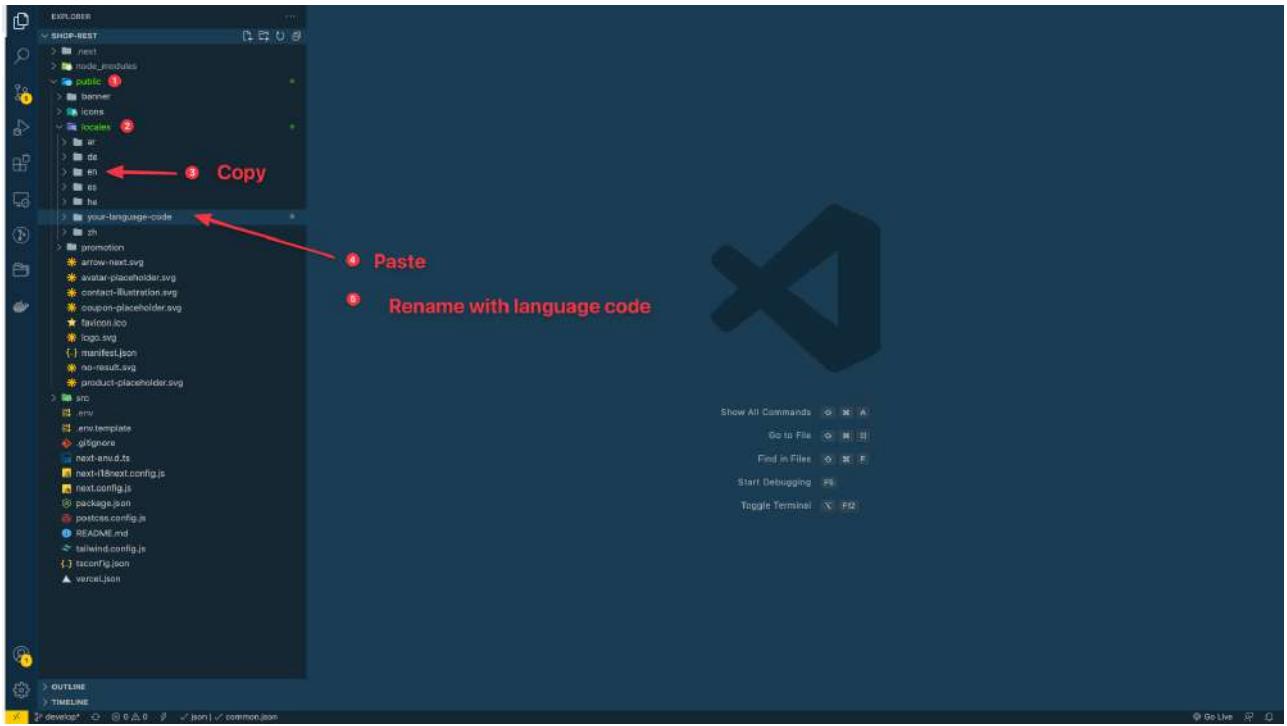


```
AsifAnik, 3 months ago + shop xti done
1 "404-heading": "رمز الخطأ 404",
2 "404-sub-heading": "وجه الفتاة! يبدو أن هذه ليست صنعة",
3 "404-back-home": "اخذني إلى المنزل",
4 "nav-menu-track-order": "ترتيب الطلب",
5 "nav-menu-offer": "العروض",
6 "nav-menu-faq": "الأسئلة الشائعة",
7 "nav-menu-contact": "جهة الاتصال",
8 "user-avatar": "صورة المستخدم الرمزية",
9 "auth-menu-profile": "المملة الشخصية",
10 "auth-menu-checkout": "الخروج",
11 "auth-menu-my-orders": "طلباتي",
12 "auth-menu-logout": "تسجيل الخروج",
13 "join-button": "انضم",
14 "change-locale": "التبديل إلى الألمانية",
15 "text-search": "بحث",
16 "text-burger-menu": "قائمة البرجر",
17 "text-close": "إغلاق",
18 "text-or": "أو",
19 "text-terms": "المصطلحات",
20 "text-policy": "السياسة",
21 "text-accept": "قبول",
22 "cookie-message": "أنت توافق على",
23 "text-privacy-policy": "سياسة الخصوصية"
```

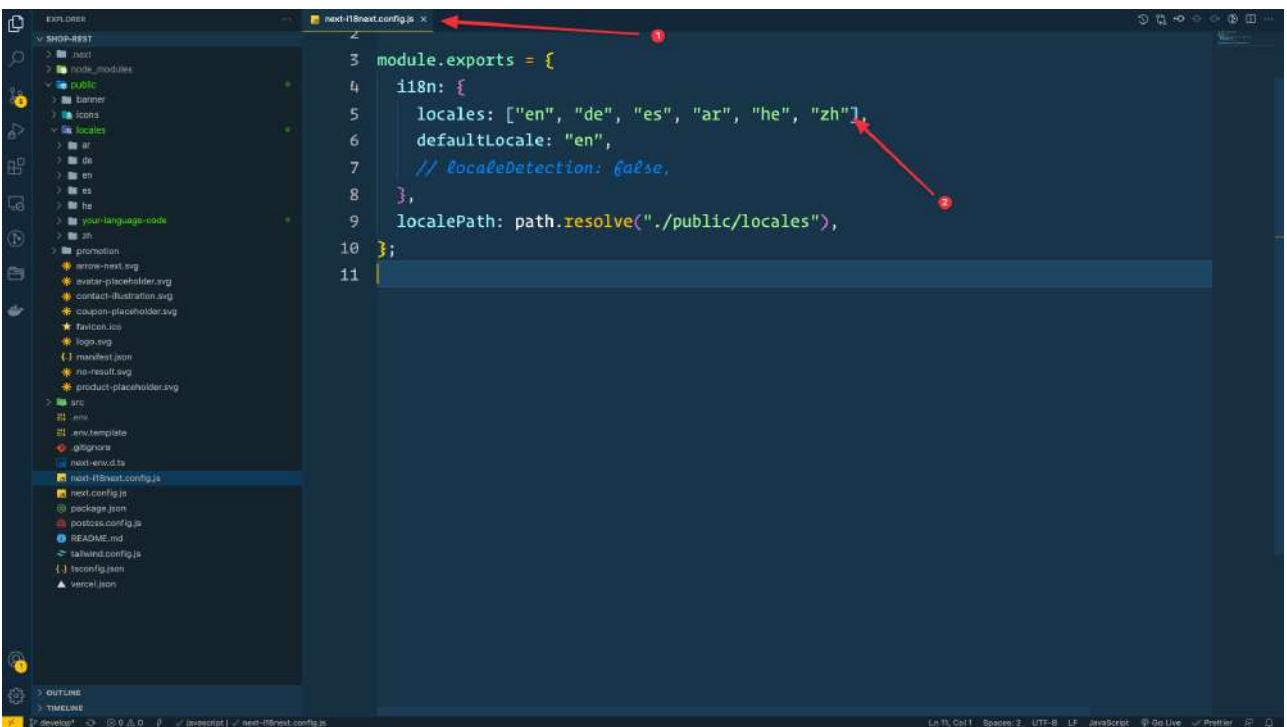
After translating `string` from `JSONre-run or rebuild`` project it'll translate the static string.

New Language

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



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



Default Language

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

For Rest

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

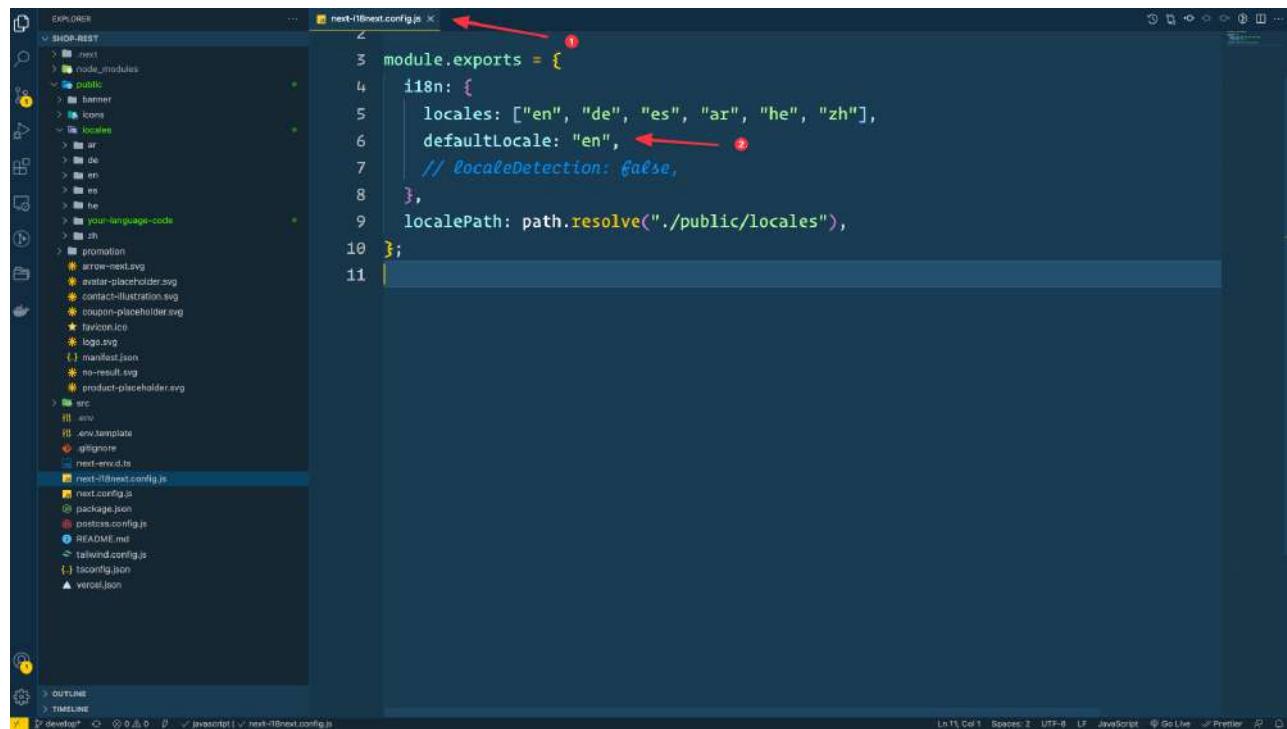
```
pickbazar-laravel -> admin -> rest -> public -> locales -> YOUR  
LANGUAGE -> next-i18next.config.js
```

For GraphQL

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

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

And change `defaultLocale` to your language code.



```
module.exports = {  
  i18n: {  
    locales: ["en", "de", "es", "ar", "he", "zh"],  
    defaultLocale: "en",  
    // localeDetection: false,  
  },  
  localePath: path.resolve("./public/locales"),  
};
```

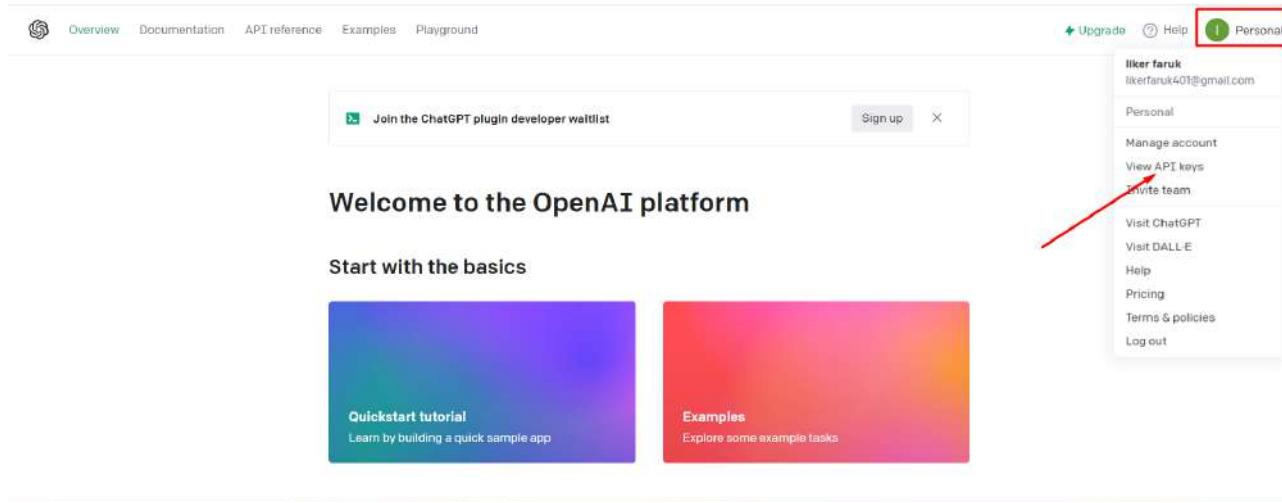
OpenAI

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.

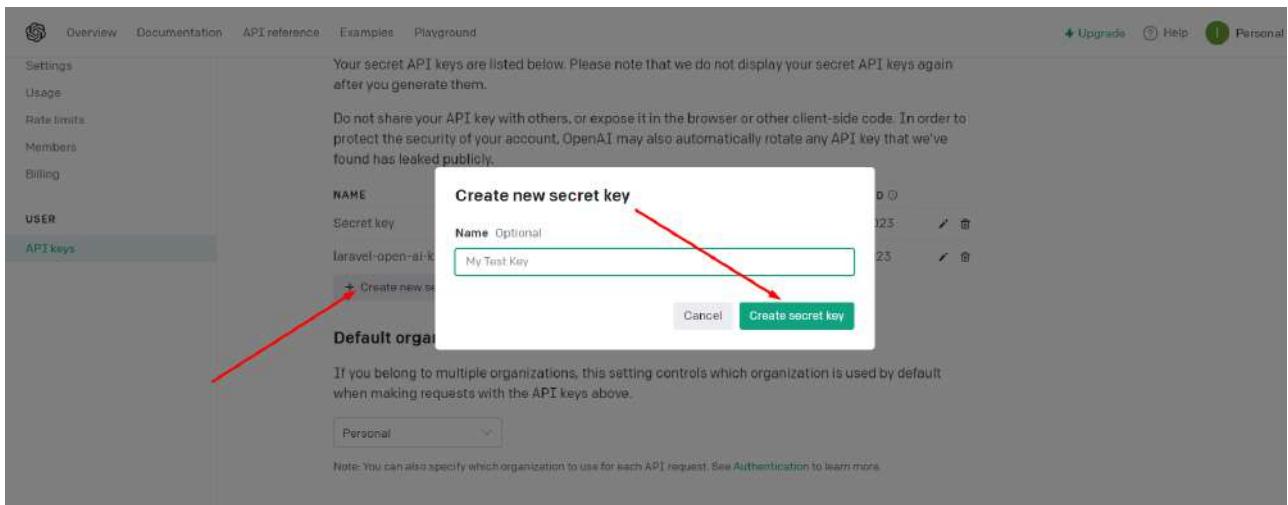
OpenAI Integration inside PickBazar.

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

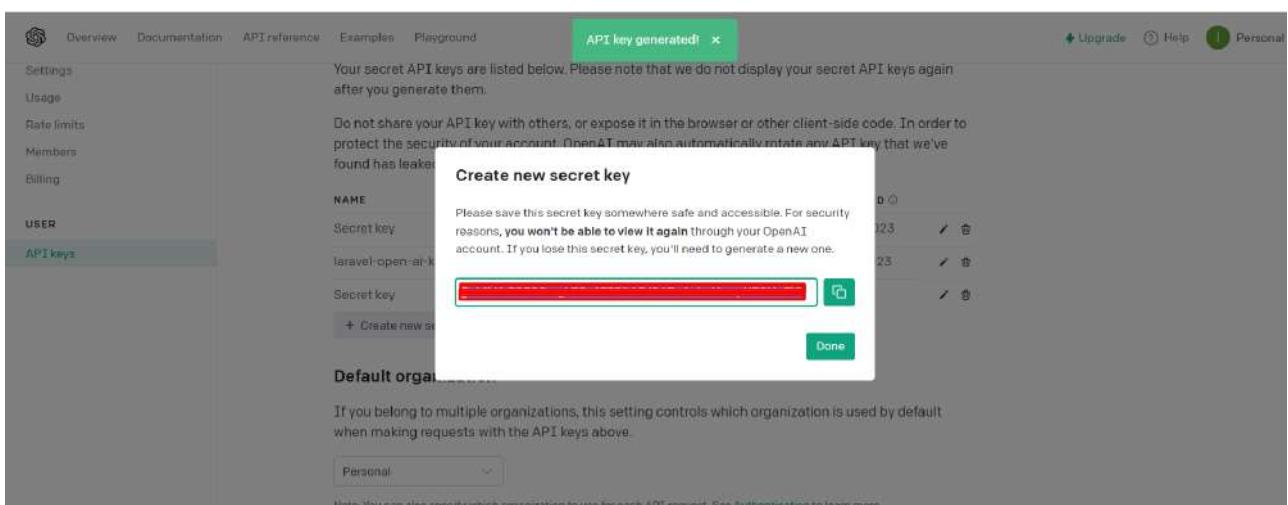
- Go to the [OpenAI Platform Website](#) & login If you aren't a registered user, Complete the OpenAI registration first.
- After logged in into OpenAI dashboard, Click on Personal, then click on View API Keys.



- 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.



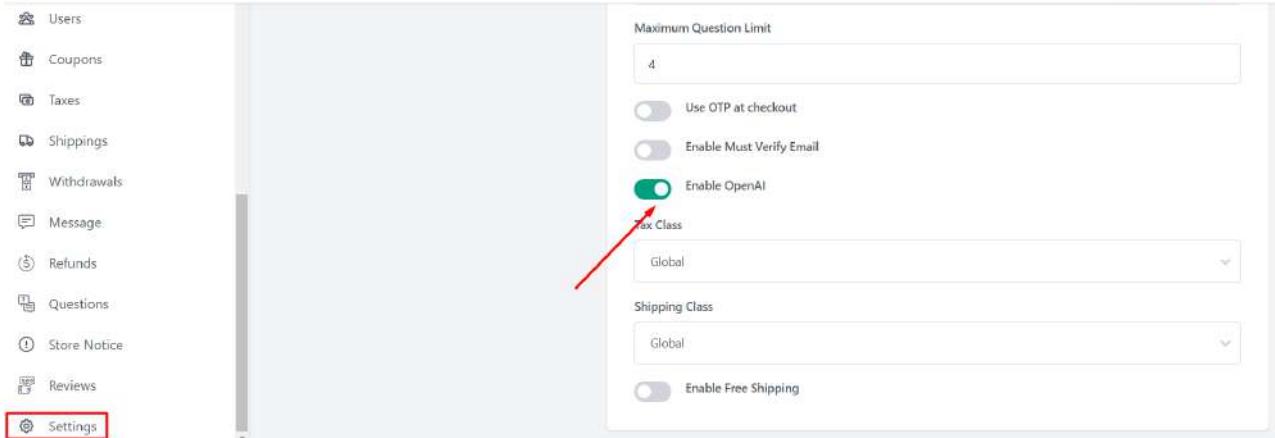
- Pickbazar api .env

```
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.



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](#)

Find or Search Nearby shops inside PickBazar.

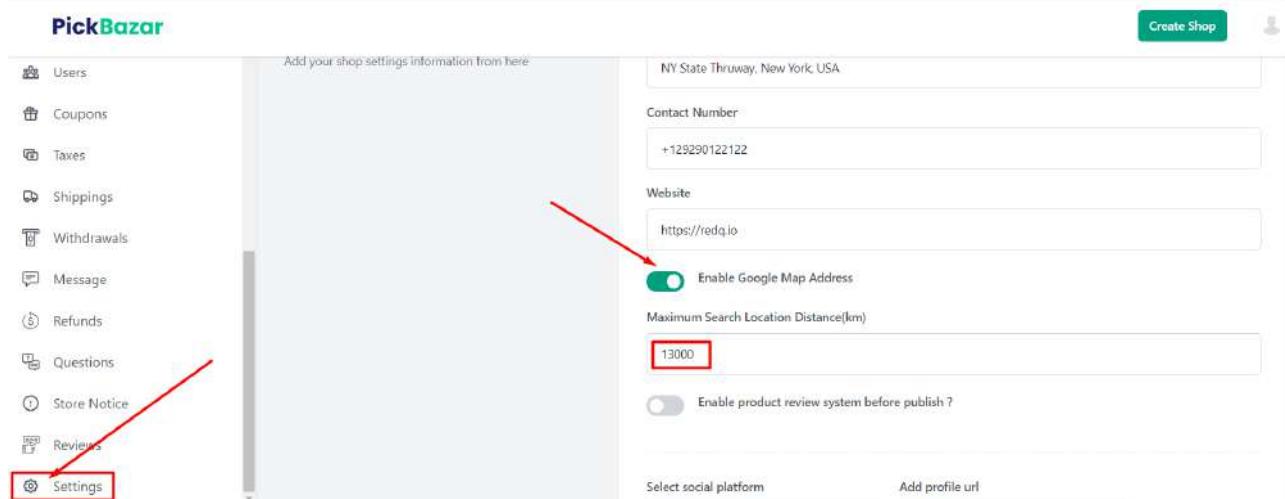
Please follow & complete this steps for Find or Search Nearby shops Enable for your e-commerce system.

- Add Google Map API key inside `.env` file of Pickbazar shop.

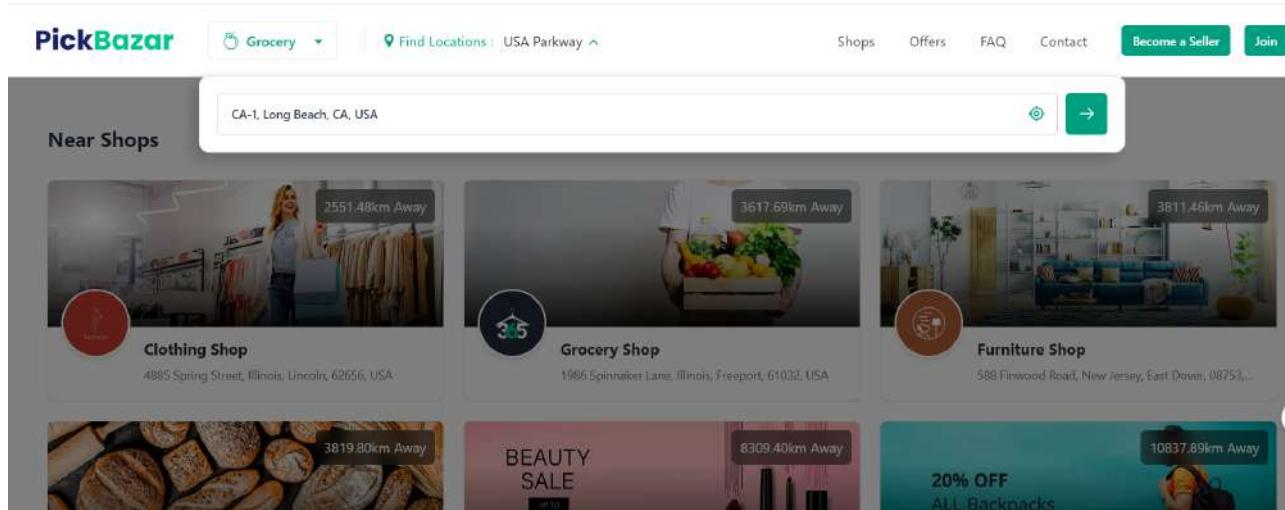
```
NEXT_PUBLIC_GOOGLE_MAP_API_KEY=
```

Admin Settings

- First go to settings from PickBazar admin dashboard. Inside settings you will find Enable Google Map Address switch button. Enable the Google Map Address & add value of Maximum Search Location Distance.



Now you can search your nearby shop



Demo Deployment

For Virtual Private Server Deployment check our [Manual Installation](#) guide.

For Virtual Private Server check our [Automated Script Installation](#) guide.

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

The screenshot shows the AWS Management Console homepage. On the left, there's a navigation sidebar with sections like 'AWS services', 'Recently visited services' (with 'Compute' and 'EC2' highlighted by a red arrow), 'All services' (listing services like Lightsail, Lambda, Batch, Elastic Beanstalk, Serverless Application Repository, AWS Outposts, EC2 Image Builder, Containers, Storage, and others), and 'Feedback' and language settings ('English (US)'). The main content area features a 'Stay connected to your AWS resources on-the-go' section with a mobile app icon and a 'Explore AWS' section with links to Amazon S3 on Outposts, Free Digital Training, and Amazon SageMaker Resources.

The screenshot shows the 'Instances' page under the EC2 service. A red arrow points to the 'Instances' link in the left sidebar. The main table lists one instance: 'i-00fb9dda581f15817' (terminated, t2.micro, us-east-1e). A red arrow points to the 'Launch instances' button at the top right of the table. The left sidebar also includes sections for 'Images', 'Elastic Block Store', and 'Metrics & CloudWatch Metrics'.

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	Actions
HTTP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop	X
HTTP	TCP	80	Custom ::/0	e.g. SSH for Admin Desktop	X
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop	X
HTTPS	TCP	443	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop	X
HTTPS	TCP	443	Custom ::/0	e.g. SSH for Admin Desktop	X

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.

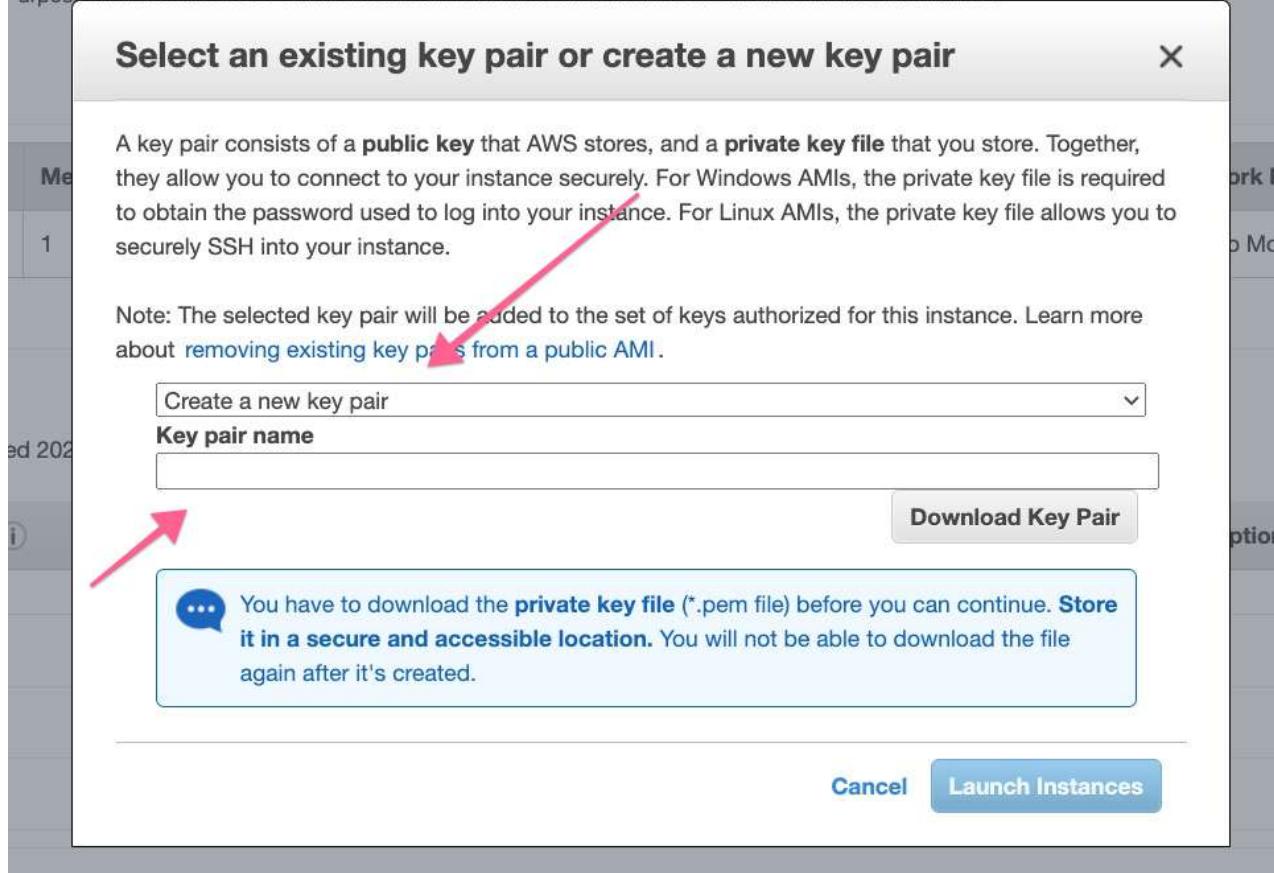
Cancel **Save**

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>)



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

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like New EC2 Experience, EC2 Dashboard, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, and Elastic Block Store. The main area displays a table of instances. A red arrow points to the "pickbazar" instance, which has a checked checkbox. Another red arrow points to the "Public IPv4 DNS" column, which shows "ec2-54-90-111-167". Below the table, a detailed view for the "pickbazar" instance is shown, with a red arrow pointing to the "Public IPv4 address" field, which contains "54.90.111.167".

Domain Setup

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

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'Instances' section, 'pickbazar' is selected. In the main table, there are two rows: one terminated instance and one running instance named 'pickbazar'. The running instance has a red arrow pointing to its name. Below the table, a detailed view for 'pickbazar' is shown, with another red arrow pointing to the 'Public IPv4 address' field, which contains the value '54.90.111.167'.

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

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

To use Cloudflare,

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

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

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

Instances (1/2) [Info](#)

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS |
|------------------|---------------------|----------------------------|---------------|--------------|--------------------------|-------------------|-------------------|
| - | i-00fb9dda581f15817 | Terminated | t2.micro | - | 2 alarms | 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 ID: i-0b3ad96ed56b60940 (pickbazar) Public IPv4 address: 54.90.168.167, [open address](#)

Private IPv4 addresses: 54.90.168.167

Add more DNS records for jsfinity.com

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

Cloud icon: Proxied: Accelerates and protects traffic
No Cloud icon: DNS resolution only: Bypasses Cloudflare

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

DNS management for .com

+ Add record Advanced

.com points to 54.90. and has its traffic proxied through Cloudflare.

| Type | Name | IPv4 address | TTL | Proxy status |
|------|------|---------------|------|--------------|
| A | @ | 54.90.168.167 | Auto | Proxied |

Cancel Save

Continue

Then add another CNAME record for www

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

Cloud icon: Proxied: Accelerates and protects traffic
No Cloud icon: DNS resolution only: Bypasses Cloudflare

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

DNS management for .com

+ Add record Advanced

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

| Type | Name | Target | TTL | Proxy status |
|-------|------|--------|------|--------------|
| CNAME | www | .com | Auto | Proxied |

domain name Cancel Save

A www.:com 54.90.168.167 Auto Proxied Delete

Continue

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

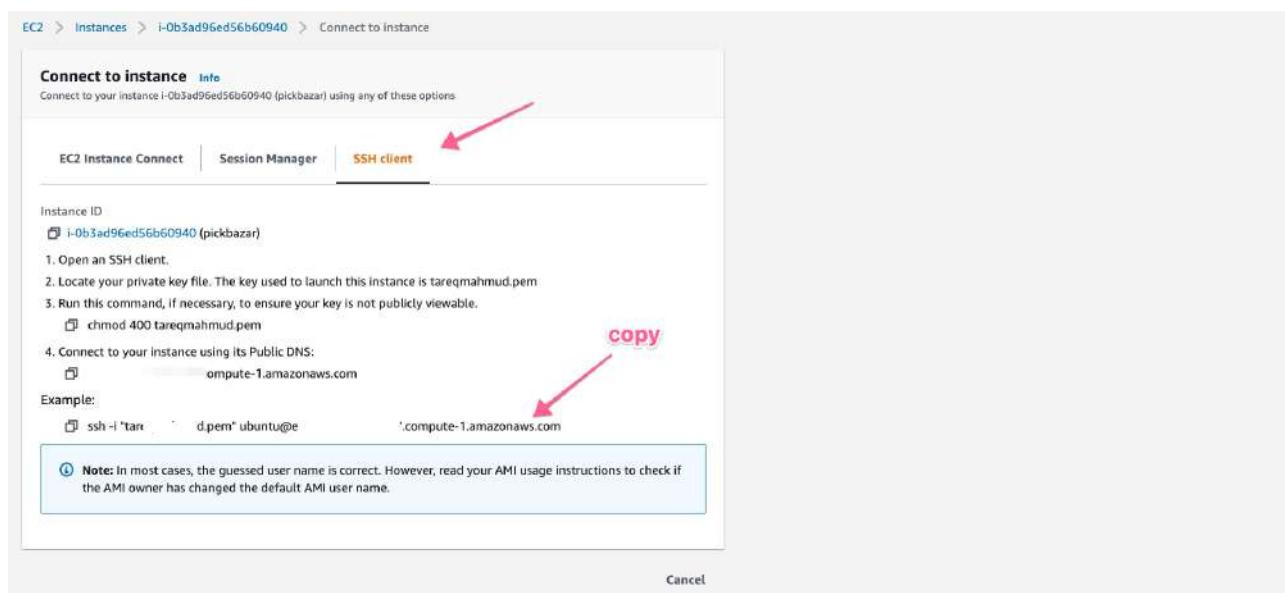
Login to Server

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

then click **Connect**



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



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

Change permission .pem

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

```
chmod 400 pickbazar.pem
```

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

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

Virtual Private Server (Automated Script)

With this tutorial, you can install PickBazar to any type of blank or empty ubuntu server.

For example, `Digital Ocean Droplets, Amazon Lightsail, AWS, Google Cloud`

`Virtual Private Server, Azure Ubuntu Virtual Private Server`, etc.

If you want to use all the scripts (`shop`, `admin`, `api`) on the same server as this tutorial, then we recommend creating a blank ubuntu-based (`v20.0.4 7ts`) 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.

https://www.youtube.com/embed/_z7VnfHcvnE

Prerequisite

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

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

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

Now you can follow the script installation procedure,

At first login your server from terminal

```
ssh SERVER_USERNAME@SERVERIP
```

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

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

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

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

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

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

Server Environment setup script - RUN on Virtual Server

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

Nginx Setup And Settings - RUN on Virtual Server

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

Backend build - RUN on Virtual Server

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

Frontend build script - RUN on Local PC

Run the below command from your pickbazar-laravel project root

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

Frontend run script - RUN on Virtual Server

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

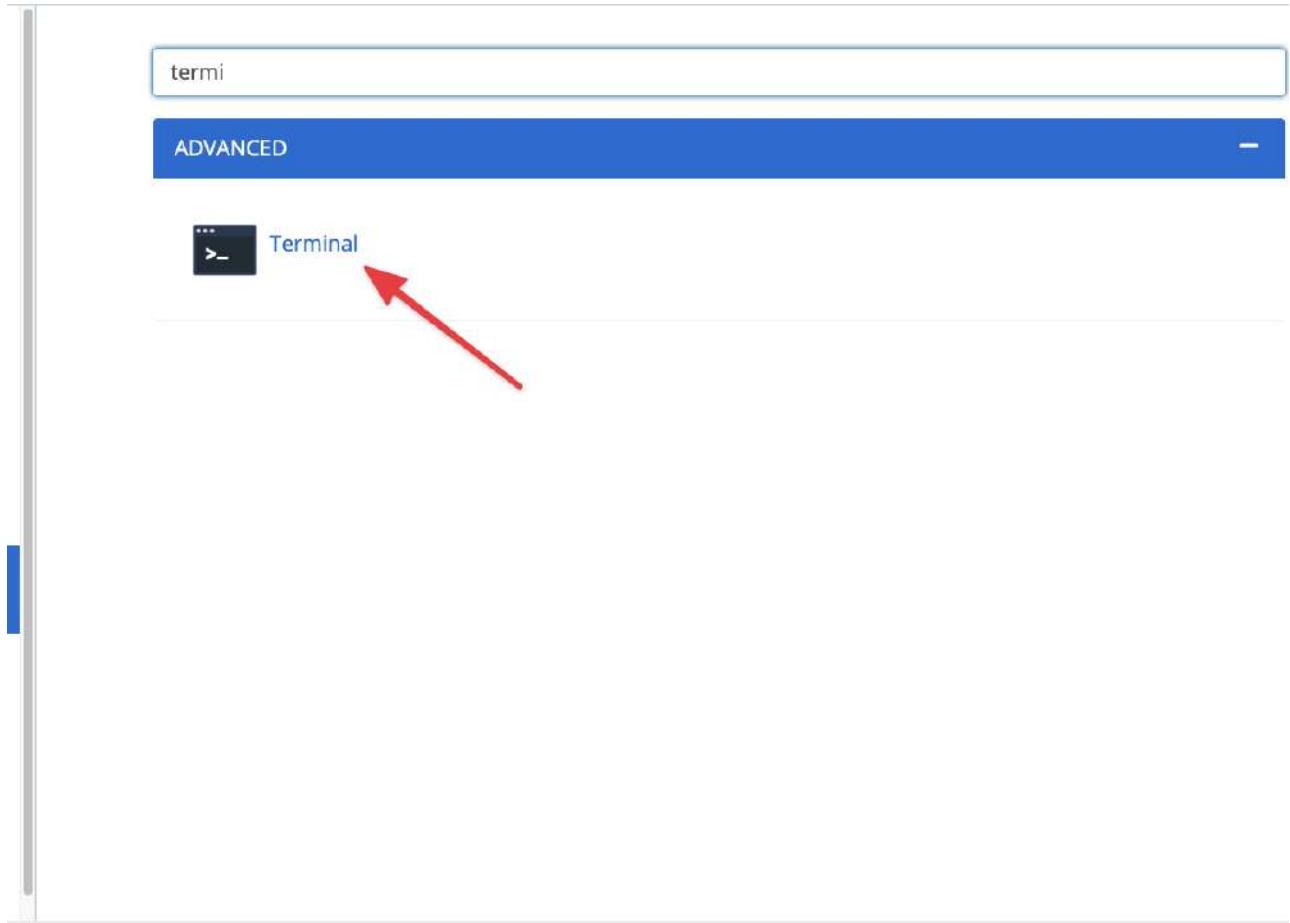
cPanel

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

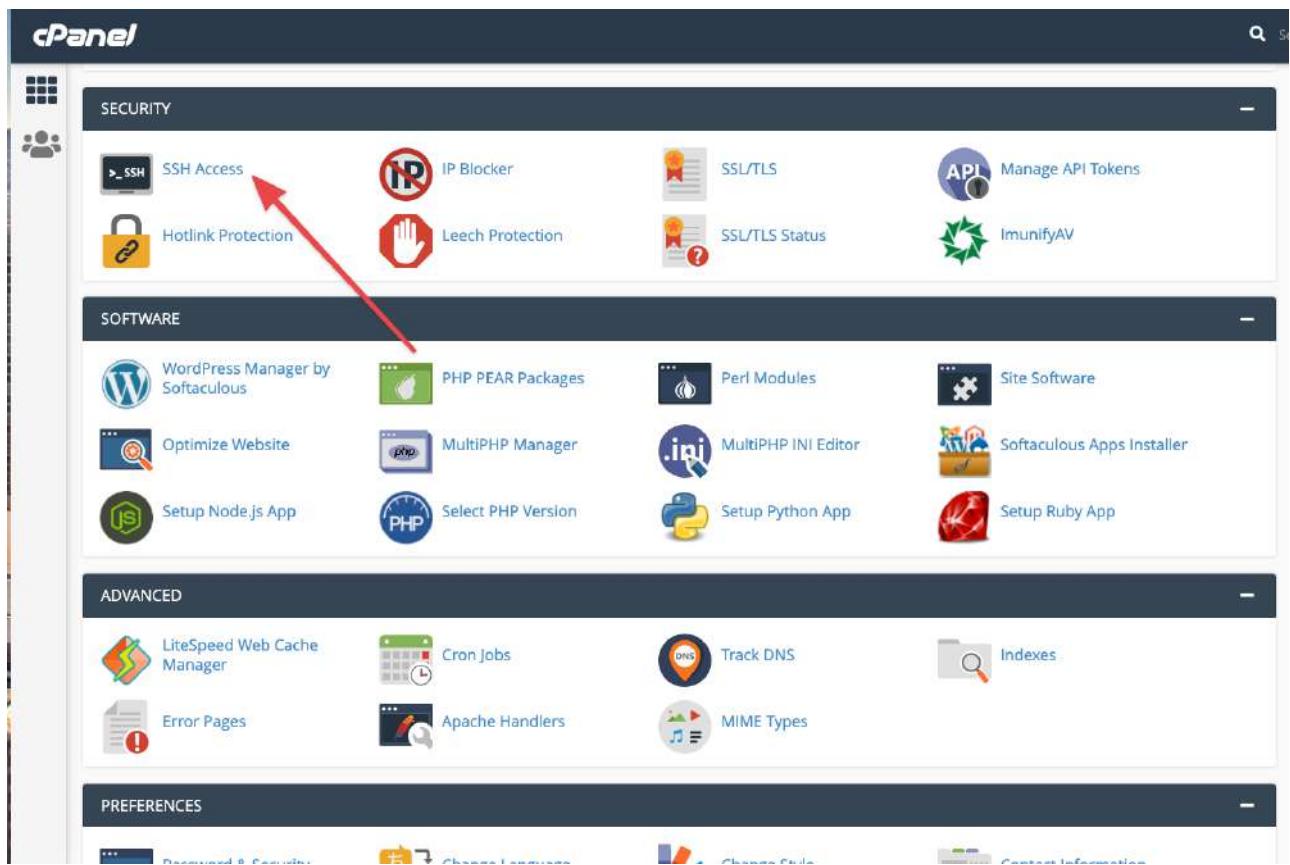
If you still decide to proceed with panel based server, our support team won't be able to help you. We have put some resources for Cpanel in this documentation section to help our users to get started but other than that, we don't have much to offer with Cpanel or any Managed Server

Access Server

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



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



After enabling the ssh login to your server using ssh,

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

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

```
composer -v
```



A screenshot of a terminal window titled "mahmud (ssh)". It shows the command "composer --version" being run, and the output "Composer version 2.0.6 2020-11-07 11:21:17" is displayed. The terminal has three tabs open: "yarn (node)", ".pickbazar-doc (~zsh)", and "mahmud (ssh)".

```
[~]# composer --version
Composer version 2.0.6 2020-11-07 11:21:17
```

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

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

After that, check the PHP version using,

```
php -v
```

make sure it's `8.1`

Create Subdomains

Now create two subdomains, for example,

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

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

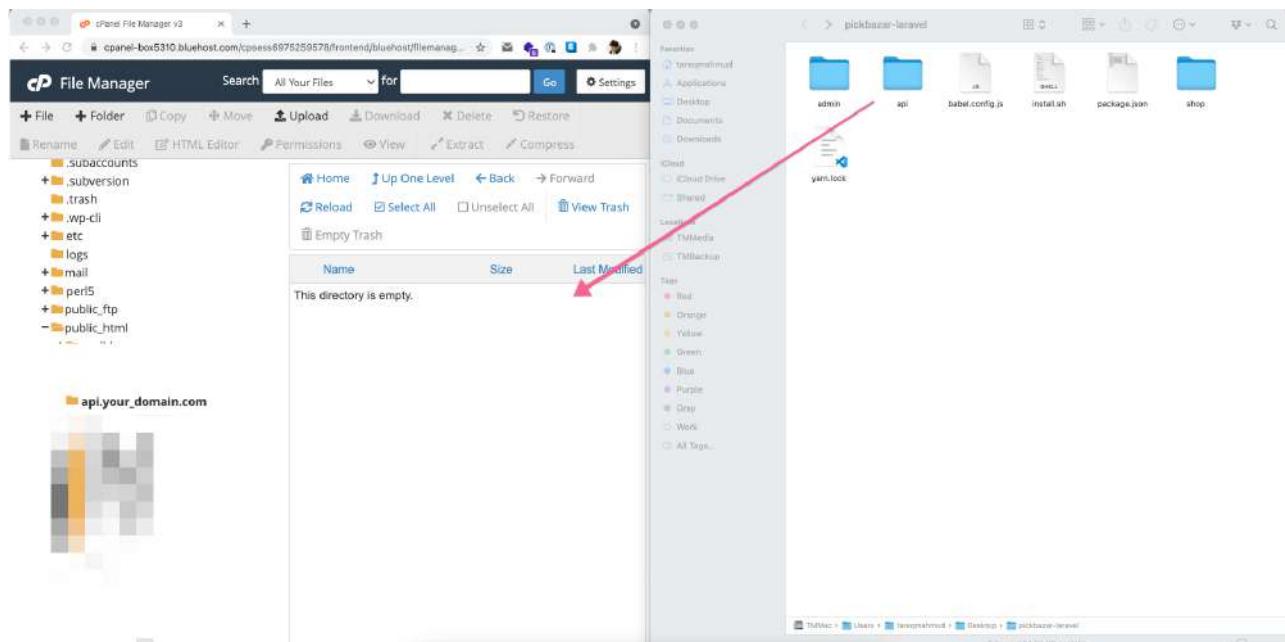
```
-> store.your_domain.com -> host frontend store.
-> api.your_domain.com -> host laravel API.
-> admin.your_domain.com -> host admin dashboard.
```

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

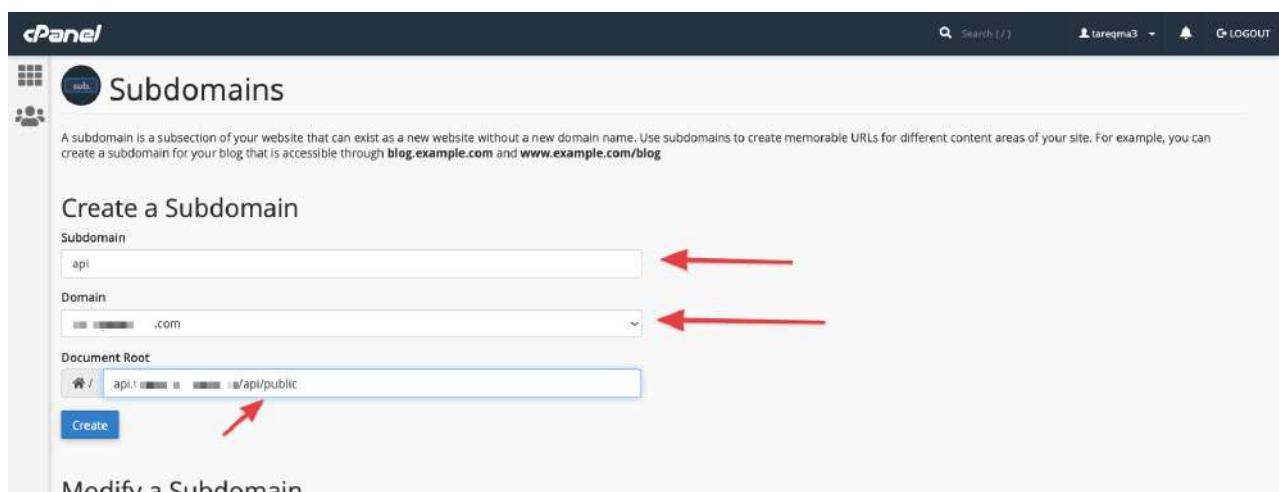
Install API

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

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



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



Now create a MySQL database and user from MySQL wizard

The screenshot shows the cPanel dashboard. At the top, there are various icons for web disk, FTP accounts, backup, and backup wizard. Below that is a section for databases, featuring phpMyAdmin, MySQL® Databases (which has a red arrow pointing to it), MySQL® Database Wizard, and Remote MySQL®. The next section is for domains, including Site Publisher, Domains, Addon Domains, Subdomains, Aliases, Redirects, Zone Editor, and Dynamic DNS. The third section is for metrics, showing Visitors, Errors, Bandwidth, Raw Access, Awstats, Analog Stats, Webalizer, and Webalizer FTP. At the bottom, there's a security bar with a URL: https://jupiter.exonhost.com:2083/cpaccess5845560484/frontend/paper_lantern/sql/index.html.

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	<input checked="" type="checkbox"/> Select All	<input type="checkbox"/> 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,

Editing: `your_domain.com/api/.env` Encoding: utf-8 Re-open

Keyboard shortcuts

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

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

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

Then go to your `ssh terminal` again and,

go to `api` folder and run,

```
composer install
```

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

```
php artisan key:generate
```

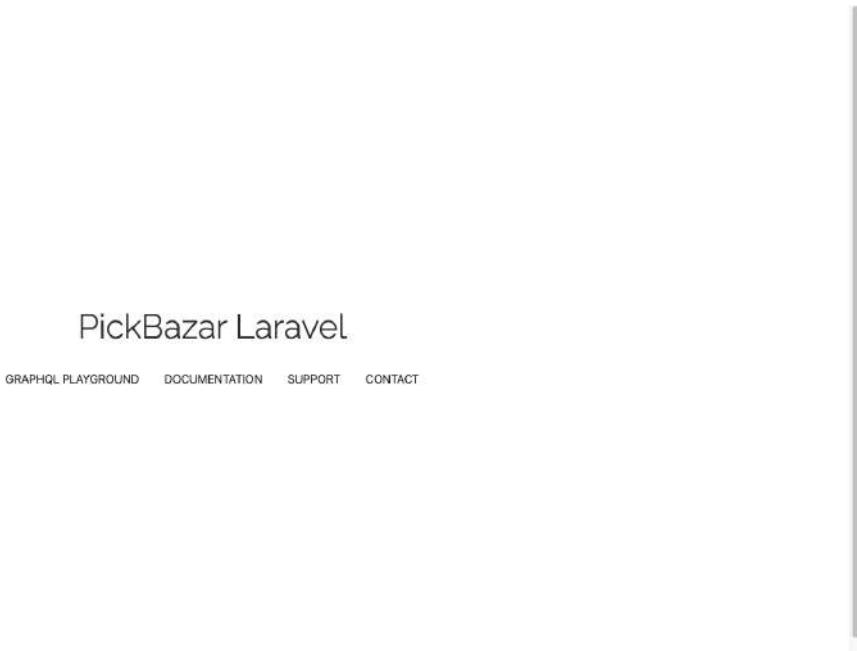
```
php artisan marvel:install
```

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

After that, run this command to link storage,

```
php artisan storage:link
```

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



PickBazar Laravel

GRAPHQL PLAYGROUND DOCUMENTATION SUPPORT CONTACT

Install 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

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

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

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

step 1 - Build Custom Server

go to your `pickbazar-laravel` folder

shop rest

Create custom server for `shop rest`,

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

and paste this code,

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

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

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

```

const { pathname, query } = parsedurl

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

}).listen(3003, (err) => {
  if (err) throw err
  console.log('> Ready on http://localhost:3003')
})
}

```

Now update package.json for shop rest,

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

and replace `start` script with this,

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

```

package.json (~\Desktop\pick-bazar-server\shop) - VM
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/fi
amework/graphql/**/*.d.ts\"",
7     "dev:rest": "next dev -p 3003",
8     "build:rest": "next build",
9     "codegen": "node -r dotenv/config $(yarn bin)/graphql-let",
10    "dev:gql": "yarn codegen && next dev -p 3001",
11    "build:gql": "yarn codegen && next build",
12    "start": "next start",
13    "lint": "next lint"
14  },
15  "dependencies": {
16    "@apollo/client": "^3.4.16",
17    "@headlessui/react": "1.4.1",
18    "@hookform/resolvers": "2.8.2",
19    "@reach/portal": "0.16.0",
20    "@stripe/react-stripe-js": "1.5.0",
21    "@stripe/stripe-js": "1.20.2",
22    "apollo-upload-client": "16.0.0",
23    "axios": "0.23.0",
24    "body-scroll-lock": "4.0.0-beta.0",
25    "camelcase-keys": "7.0.1",
26    "classnames": "2.3.1",
27    "dayjs": "1.10.7",
28    "deepmerge": "4.2.2",
29    "framer-motion": "4.1.17",
30  }
31 ]
32 package.json (~\Desktop\pick-bazar-server\shop) - VM
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/fi
amework/graphql/**/*.d.ts\"",
7     "dev:rest": "next dev -p 3003",
8     "build:rest": "next build",
9     "codegen": "node -r dotenv/config $(yarn bin)/graphql-let",
10    "dev:gql": "yarn codegen && next dev -p 3001",
11    "build:gql": "yarn codegen && next build",
12    "start": "NODE_ENV=production node server.js",
13    "lint": "next lint"
14  },
15  "dependencies": {
16    "@apollo/client": "3.4.16",
17    "@headlessui/react": "1.4.1",
18    "@hookform/resolvers": "2.8.2",
19    "@reach/portal": "0.16.0",
20    "@stripe/react-stripe-js": "1.5.0",
21    "@stripe/stripe-js": "1.20.2",
22    "apollo-upload-client": "16.0.0",
23    "axios": "0.23.0",
24    "body-scroll-lock": "4.0.0-beta.0",
25    "camelcase-keys": "7.0.1",
26    "classnames": "2.3.1",
27    "dayjs": "1.10.7",
28    "deepmerge": "4.2.2",
29    "framer-motion": "4.1.17",
30  }
31 ]
32 package.json[+]
33 -- INSERT --

```

admin rest

Similarly, create custom server for `admin rest`,

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

and paste this code,

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

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

app.prepare().then(() => {
  createServer((req, res) => {
    // Be sure to pass `true` as the second argument to
    `url.parse`.

    // This tells it to parse the query portion of the URL.
    const parsedUrl = parse(req.url, true)
    const { pathname, query } = parsedUrl

    if (pathname === '/a') {
      app.render(req, res, '/a', query)
    } else if (pathname === '/b') {
      app.render(req, res, '/b', query)
    } else {
      handle(req, res, parsedUrl)
    }
  }).listen(3002, (err) => {
    if (err) throw err
    console.log('> Ready on http://localhost:3002')
  })
})
```

Now update package.json for `admin rest`,

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

and replace `start` script with this,

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

admin graphql

Skip this step if you want to use REST API

Create custom server for `admin graphql`,

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

and paste this code,

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

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

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

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

```
        }
    }).listen(3000, (err) => {
    if (err) throw err
    console.log('> Ready on http://localhost:3000')
})
})
```

Now update package.json for `admin graphql`,

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

and replace `start` script with this,

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

shop graphql

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

Create custom server for `shop graphql`,

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

and paste this code,

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

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

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

```
const parsedUrl = parse(req.url, true)
const { pathname, query } = parsedUrl

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

)).listen(3001, (err) => {
  if (err) throw err
  console.log('> Ready on http://localhost:3001')
})
})
```

Now update package.json for `shop graphql`,

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

and replace `start` script with this,

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

Step 2 - Install & Build

go to your `pickbazar-laravel` folder

To install all the npm packages run this command,

```
yarn
```

Step 3 - Build the project

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

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

For REST

Go to,

```
cd shop
```

then use this command to copy,

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

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

```
nano .env
```

and use

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

and

```
FRAMEWORK_PROVIDER="rest"
```

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

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

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

then use this command to copy,

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

```
nano .env
```

and use

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

Now go to the `pickbazar-laravel` folder,

and run,

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

For GraphQL

Skip this step if you want to use REST API

and go to,

```
cd shop
```

then use this command to copy,

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

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

```
nano .env
```

and use

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

and

```
FRAMEWORK_PROVIDER="graphql"
```

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

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

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

then use this command to copy,

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

Again edit `.env`

```
nano .env
```

and use

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

Now go to the `pickbazar-laravel` folder,

and run,

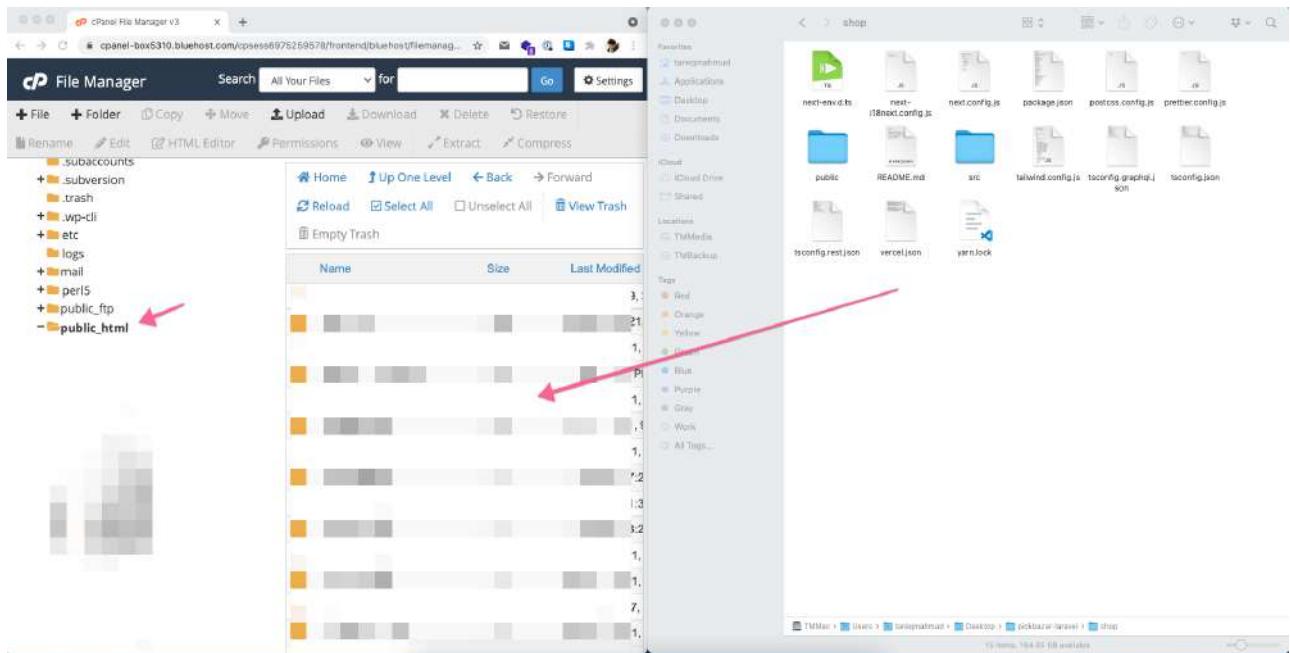
```
yarn build:shop-gql  
yarn build:admin-gql
```

After build the project upload the

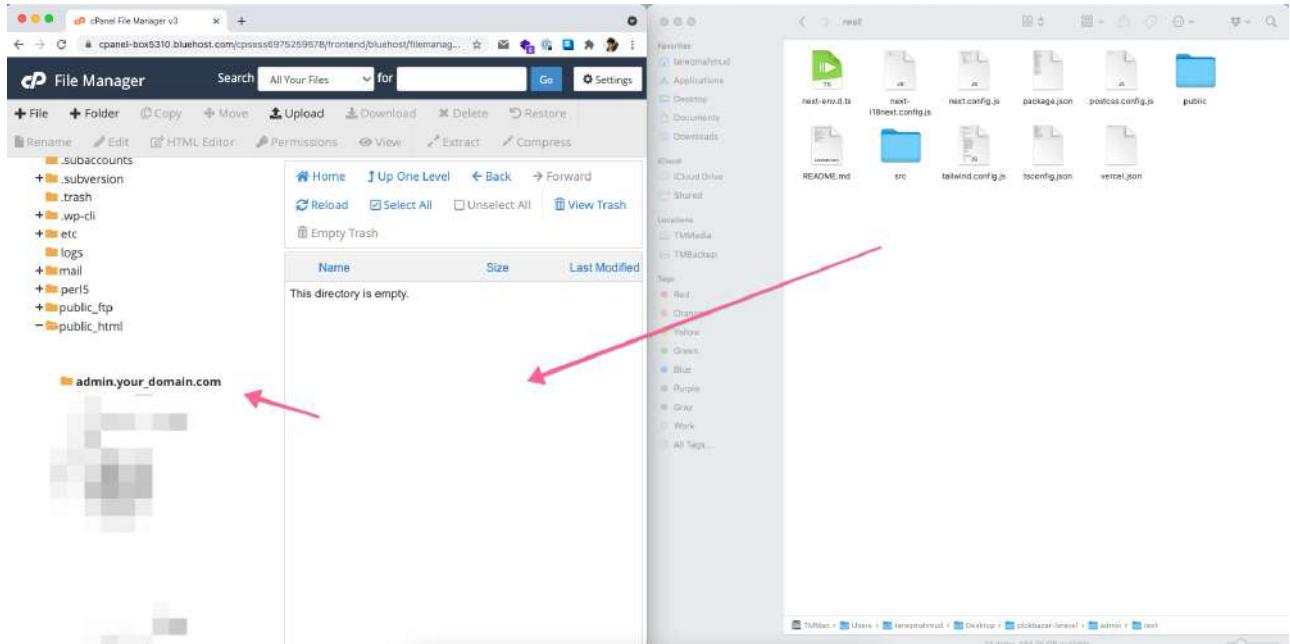
- `shop` to `root_domain -> public_html` folder
- `admin-rest` to `admin.your_domain.com` folder

For graphql upload admin-graphql instead of admin-rest

`shop`,



`shop-admin`,



Install NodeJs Project

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

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

For `shop`,

The screenshot shows the cPanel interface under the 'SOFTWARE' section. It includes links for various software management tools like WordPress Manager, PHP PEAR Packages, Perl Modules, Site Software, Optimize Website, MultiPHP Manager, MultiPHP INI Editor, Softaculous Apps Installer, Setup Node.js App (which has a red arrow pointing to it), Select PHP Version, Setup Python App, and Setup Ruby App.

The screenshot shows the 'Node.js' application setup page in cPanel. It features a 'WEB APPLICATIONS' section with a 'CREATE APPLICATION' button. A red arrow points to this button. Below it, a message says 'NO APPLICATIONS FOUND'.

Now,

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

cPanel

WEB APPLICATIONS [CREATE APPLICATION](#)

Node.js version: 12.22.1

Application mode: Production

Application root: /home/tareqma3/public_html

Application URL:

Application startup file: server.js

Passenger log file: /home/tareqma3/logs/passenger.log

Environment variables: [ADD VARIABLE](#)

NO RESULT FOUND

You can get the Application Path from your cPanel file manager

cP File Manager

File Manager

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

Search: All Your Files for Go Settings

Collapse All

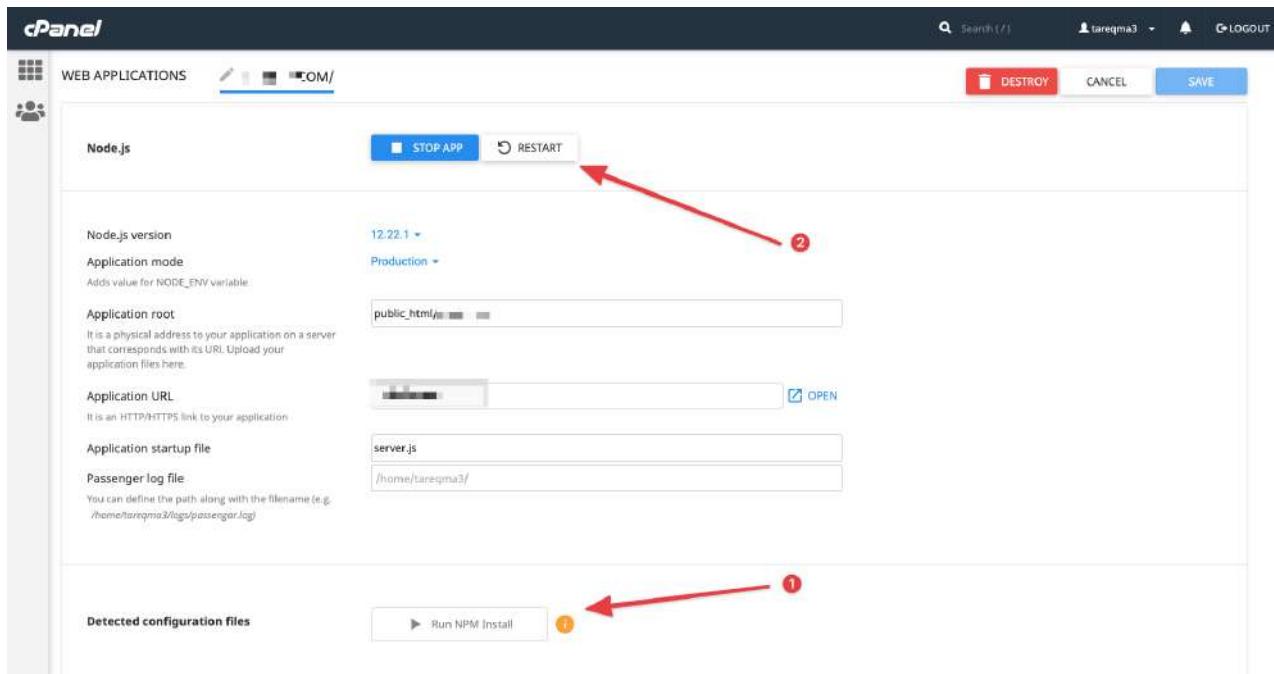
Name	Size	Last Modified	Type	Permissions
.cagefs	39 bytes	Feb 12, 2021, 2:10 PM	httpd/unix-directory	0771
.cl.selector	48 bytes	May 13, 2021, 1:44 PM	httpd/unix-directory	0755
.cpanel	195 bytes	Today, 5:16 PM	httpd/unix-directory	0700
.cphorde	70 bytes	Mar 24, 2021, 9:28 PM	httpd/unix-directory	0700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

(/home/tareqma3)

public_html

admin.your_domain.com

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



For `admin`,

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

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

Thank You!

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
```

For graphql admin -> graphql -> vercel.json

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

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

After that, log in to vercel using this command,

```
vercel login
```

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

```
vercel
```

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

```
vercel
```

For graphql admin -> graphql -> vercel.json

For more details,

Please follow [nextjs deployment docs](#):

Virtual Private Server

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

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

If you want to use all the scripts (`shop`, `admin`, `api`) on the same server as this tutorial, then we recommend creating a blank ubuntu-based (`v20.0.4 7ts`) 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.

<https://www.youtube.com/embed/1ptQKY8BycI>

Access Server

At first login your server using `SSH` and `Terminal`

Install NodeJS & Required Application

Install NodeJS

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

```
sudo apt-get update
```

```
curl -sL https://deb.nodesource.com/setup_16.x | sudo -E bash -
```

```
sudo apt-get install -y nodejs
```

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

```
sudo apt update
```

Install Yarn

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

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

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

Npm Permission Issue

Install Zip & Unzip

```
sudo apt install zip unzip
```

Install PM2

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

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

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

```
pm2 startup systemd
```

Setup Server

Introduction

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

Step 1 - Installing Nginx

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

```
sudo apt update
```

Add PPA to get the specific php version

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

```
sudo apt update
```

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

```
sudo apt install nginx
```

Step 2: Adjusting the Firewall

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

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

```
sudo ufw app list
```

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

Available applications:

- Nginx Full
- Nginx HTTP
- Nginx HTTPS
- OpenSSH

At first, add ssh to the firewall,

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

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

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

Now enable the firewall,

```
sudo ufw enable
```

```
sudo ufw default deny
```

You can verify the change by typing:

```
sudo ufw status
```

The output will be indicated which HTTP traffic is allowed:

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

Step 3 – Checking your Web Server

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

```
systemctl status nginx
```

You'll get an output like this,

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

Step 4 - Install MySQL

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

Step 5 - Install PHP & Composer

```
sudo apt install php8.1-fpm php8.1-mysql
```

```
sudo apt install php8.1-mbstring php8.1-xml php8.1-bcmath php8.1-
simplexml php8.1-intl php8.1-gd php8.1-curl php8.1-zip php8.1-gmp
```

```
php -r "copy('https://getcomposer.org/installer', 'composer-
setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'e21205b207c3ff031906575712edab6f13eb0b361f2085f1f1237b7126d785e826
a450292b6cf1d64d92e6563bbde02') { echo 'Installer verified'; }
else { echo 'Installer corrupt'; unlink('composer-setup.php'); }
echo PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

```
sudo mv composer.phar /usr/bin/composer
```

Step 6 - Create MySQL Database & User

```
sudo mysql
```

```
CREATE DATABASE pickbazar;

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

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

FLUSH PRIVILEGES;
```

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

Step 7 - Change permission for the `www` folder

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

Step 8 - Upload API to Server

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

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

Then, go to your `local computer`

1. Extract the `pickbazar` package that you download from [CodeCanyon](#).
2. Rename that folder as `redq-ecommerce`
3. On that folder, you'll get another `folder` called `pickbazar-laravel`.
4. On that folder, you'll get a folder called `api`

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

Step 9: Setting Up Server & Project

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

At first, we'll disable the default configuration.

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

Step 10 - Create New Nginx for the domain

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

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

Add this Nginx config file to that edited file,

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

For REST API

```
server {
    listen 80;

    server_name YOUR_DOMAIN.com;

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

    index index.html index.htm index.php;

    charset utf-8;

    # For API
    location /backend {
        alias /var/www/pickbazar/api/public;
        try_files $uri $uri/ @backend;
        location ~ \.php$ {
            
```

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

}

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

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

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

error_page 404 /index.php;

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

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

```
}
```

For GraphQL API

Skip this step if you want to use REST API

```
server {
    listen 80;

    server_name YOUR_DOMAIN.com;

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

    index index.html index.htm index.php;

    charset utf-8;

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

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

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

```

}

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

error_page 404 /index.php;

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

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

```

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

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

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

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

Then enable the config

```

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

```

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

```
sudo nginx -t
```

Next, restart Nginx:

```
sudo systemctl restart nginx
```

Secure Server

Step 1: Secure Nginx with Let's Encrypt

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

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

```
sudo certbot --nginx -d YOUR_DOMAIN
```

After this command, you'll get several command 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 `api` folder, then copy `.env.example` to `.env`,

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

```
cp .env.example .env
```

Edit `.env`

```
nano .env
```

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

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

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

Then install all the packages and install `api`

```
composer install
```

```
php artisan key:generate
```

```
php artisan marvel:install
```

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

Enable `Laravel storage`,

```
php artisan storage:link
```

Then give proper `permission` for laravel folder,

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

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



PickBazar Laravel

GRAPHQL PLAYGROUND DOCUMENTATION SUPPORT CONTACT

FrontEnd Project Build

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

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

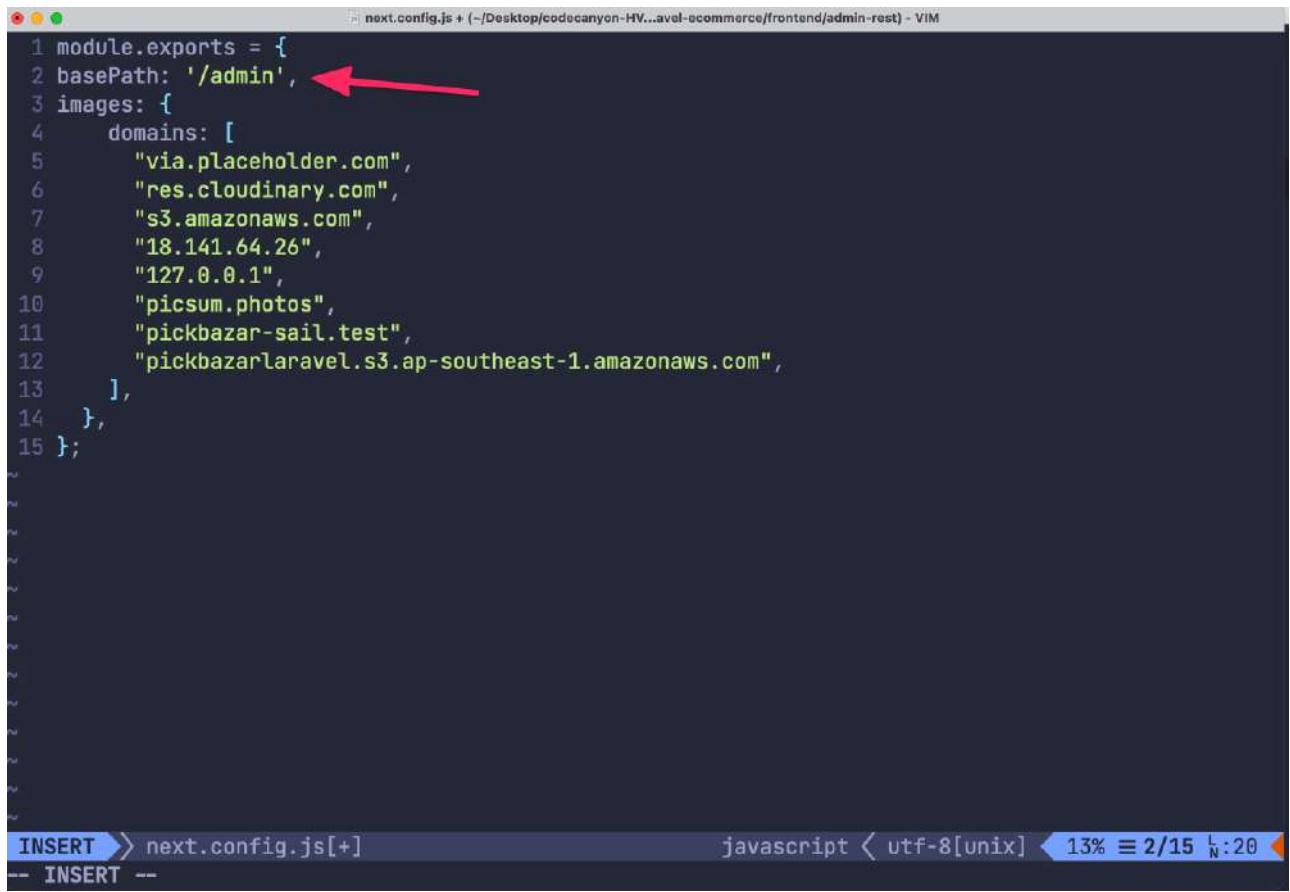
We'll suggest you build the frontend part on your computer and then upload the build file to the server.

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

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

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

add `basePath` for '/admin'



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

INSERT > next.config.js[+] javascript < utf-8[unix] 13% 2/15 L:20
-- INSERT --

Again,

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

add `basePath` for '/admin'

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

```

INSERT next.config.js[+] javascript < utf-8[unix] 5% 2/36 N:1
-- INSERT --

Step 2 - Install & Build

go to your `pickbazar-laravel` folder again

To install all the npm packages run this command,

```
yarn
```

Step 3 - Build the project

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

For REST

Go to,

```
cd shop
```

then use this command to copy,

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

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

and use

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

and

```
FRAMEWORK_PROVIDER="rest"
```

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

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

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

then use this command to copy,

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

and use

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

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

and add your domain to images object

```
next.config.js [root] - vim
3 const { i18n } = require("./next-i18next.config");
4
5 module.exports = withPWA({
6   i18n,
7   pwa: {
8     disable: process.env.NODE_ENV === "development",
9     dest: "public",
10    runtimeCaching,
11  },
12  async redirects() {
13    return [
14      {
15        source: "/",
16        destination: "/grocery",
17        permanent: false,
18      },
19    ];
20  },
21  images: {
22    domains: [
23      "YOUR_API_DOMAIN", | ←
24      "res.cloudinary.com",
25      "s3.amazonaws.com",
26      "18.141.64.26",
27      "via.placeholder.com",
28      "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
29      "picsum.photos",
30    ],
31  },
32  // ...
33  // ...
34  // ...
35}
INSERT ↵ next.config.js[+]
-- INSERT --
```

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

For GraphQL

Skip this step if you want to use REST API

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

```
cd shop
```

then use this command to copy,

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

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

and use

```
NEXT_PUBLIC_GRAPHQL_API_ENDPOINT="https://YOUR_DOMAIN/backend/graph
ql"
```

and

```
FRAMEWORK_PROVIDER="graphql"
```

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

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

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

then use this command to copy,

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

Again edit `.env`

```
NEXT_PUBLIC_GRAPHQL_API_ENDPOINT="https://YOUR_DOMAIN/backend/graph  
ql"
```

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

and add your domain to `images` object

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

javascript < utf-8[unix] 65% ↻ 25 ↺

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

Build Project

Now go to the `pickbazar-laravel` folder,

and run,

For `REST api`

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

And for `GraphQL`,

Skip this step if you want to use REST API

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

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

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

Then `unzip` the `frontend` zip file.

Install FrontEnd And Run

Then install all the node packages,

```
yarn
```

Run frontend app

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

For REST API

For `shop rest` app, use this command,

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

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

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

For GraphQL API

Skip this step if you want to use REST API

For `shop gql` app, use this command,

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

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

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

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

For API Integration:

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

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

For example:

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

for `GraphQL`

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

or for `REST`

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

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

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

For GraphQL

We have used GraphQL Query and Mutation which is defined in

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

For details documentation:

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

You can check and customize it as your need.

Uses Example:

- Query

```
import {  
    useOrdersQuery,  
    OrdersDocument,  
    OrderDocument,  
    OrdersOrderByColumn,  
    SortOrder,  
} from "@graphql/orders.graphql";  
  
// Fetch list of orders  
  
const { data, error, loading, fetchMore } = useOrdersQuery({  
    variables: {  
        orderBy: [{ field: OrdersOrderByColumn.CreatedAt, order:  
SortOrder.Desc }],  
        first: 10,  
        page: 1,  
        customer_id: 1,  
    },  
});  
  
fetchMore({  
    variables: {  
        page,  
        first,  
    },  
};
```

```

    },
});

// Fetch Single Order

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

```

- Mutation

```

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

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

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

```

```
active_from: "2020-12-20 13:43:32",
expire_at: "2020-12-30 13:43:32",
type: CouponType.FixedCoupon,
};

const { data, error, loading } = await updateCoupon({
variables: { id: 22, input: updateInput },
});

const { data, error, loading } = await deleteCoupon({
variables: { id: 22 },
});

};
```

For REST API

We have used React Query and Axios which is defined in

```
[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,
  isFetchingNextPage,
  fetchNextPage,
  hasNextPage,
  isError,
  data,
  error,
} = useProductsQuery({
  type: "grocery",
```

```
    text: "",
    category: "grocery",
});
```

- Mutation

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

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

function handleSubmit(inputValues) {
  createProduct(
    {
      ...inputValues,
    },
    {
      onError: (error: any) => {
        Object.keys(error?.response?.data).forEach((field: any) =>
        {
          setError(field, {
            type: "manual",
            message: error?.response?.data[field][0],
          });
        });
      },
    }
  );
}
```

For customizing the template's default site settings:

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

CSS styles:

`[your-frontend-project] = admin [rest/graphql] 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 refference you can see one of our icon.)
5. If your custom SVG code has any single custom color then change them into fillColor.

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

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

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

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

New Static Page

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

Shop,

```
pickbazar-laravel/shop/src/pages
```

Admin,

REST

```
pickbazar-laravel/admin/rest/src/pages
```

GraphQL

```
pickbazar-laravel/admin/graphql/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>

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

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

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

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

Then remove the composer,

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

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

```
cd /var/www/pickbazar-laravel/api  
  
sudo chown -R $USER:www-data storage  
sudo chown -R $USER:www-data bootstrap/cache  
  
rm vendor -rf
```

Then install PHP 8.1, and its extensions,

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

```
sudo apt install php8.1-fpm php8.1-mysql
```

```
sudo apt install php8.1-mbstring php8.1-xml php8.1-bcmath php8.1-simplexml php8.1-intl php8.1-gd php8.1-curl php8.1-zip php8.1-gmp
```

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'e21205b207c3ff031906575712edab6f13eb0b361f2085f1f1237b7126d785e826
a450292b6cf1d64d92e6563bbde02') { echo 'Installer verified'; }
else { echo 'Installer corrupt'; unlink('composer-setup.php'); }
echo PHP_EOL;"
```

```
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

```
sudo mv composer.phar /usr/bin/composer
```

Then update 8.0 fpm to 8.1 fpm from,

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

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

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

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

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

error_page 404 /index.php;

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

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

```
sudo nginx -t
```

Next, restart Nginx:

```
sudo systemctl restart nginx
```

Then install composer packages,

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

```
composer install
```

```
php artisan optimize:clear
```

```
php artisan marvel:install
```

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

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



PickBazar Update - Virutal Private Server

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

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

Step 1: Setup Git - Server

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

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

Install git

```
sudo apt install git
```

Config for first time

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

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

Prepare Git Repository

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

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

2. Then initialize a git on that folder,

```
git init
```

3. Create a new `.gitignore` file

```
nano .gitignore
```

and paste this code to that `.gitignore`,

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

# dependencies
node_modules
/.pnp
.pnp.js

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

# misc
.DS_Store

# If Packages have their independent repo
# /packages

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

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

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

```
git init
```

```
git add .
```

```
git commit -m "Initial commit"
```

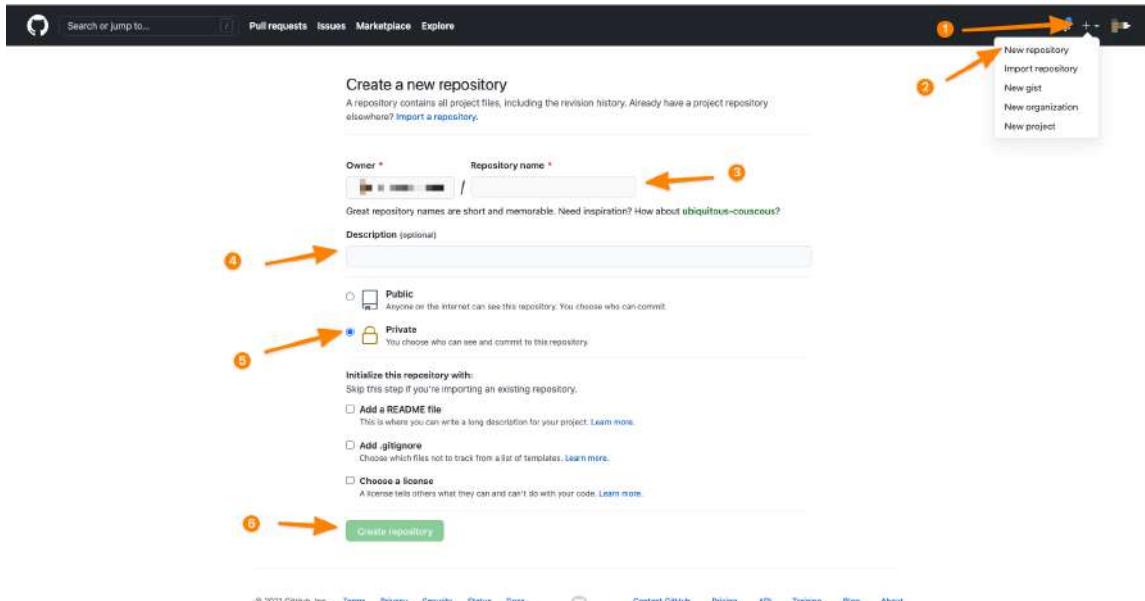
Create a separate branch to maintain the updated code.

```
git branch pickbazar
```

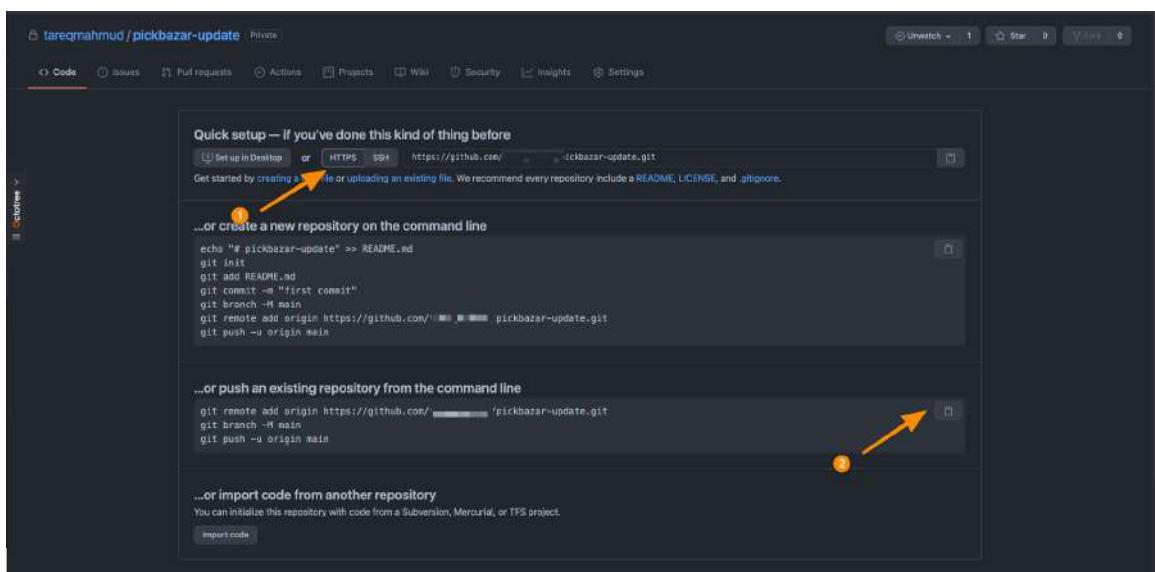
Git & Github

1. At first, go to <https://github.com/> and create an account first. If you already have an account then `sign in` on that account.

2. Then create a new repository,



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



And paste that copied command to pickabazar folder and press enter

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

```
root@redq:/var/www/pickbazar# git remote add origin https://github.com/██████████/pickbazar-update.git
fatal: remote origin already exists.
root@redq:/var/www/pickbazar# git branch -M main
root@redq:/var/www/pickbazar# git push -u origin main
Username for 'https://github.com':
```

Your existing repository is successfully connected with GitHub.

Step 2: Shut Down Current Process

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

```
pm2 stop 'all'
```

Step 3: Local Repository & Updated Code

In this step,

1. Clone that `GitHub repository` to your `local computer`
2. Download `update package` from `CodeCanyon`
3. Open your terminal and clone that repository to your computer
4. Checkout to `pickbazar` branch

```
git checkout -b pickbazar
```

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

```
git add .
```

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

7. Merge new code with `main` branch

```
git checkout -b main
```

```
git merge pickbazar
```

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

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

```
git push origin main
```

Step 4: Update API

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

```
git pull origin main
```

After pull go to `api` folder,

```
cd api
```

and install composer package and optimize compiled file,

```
composer install
```

```
php artisan optimize:clear
```

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

Step 5: 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.

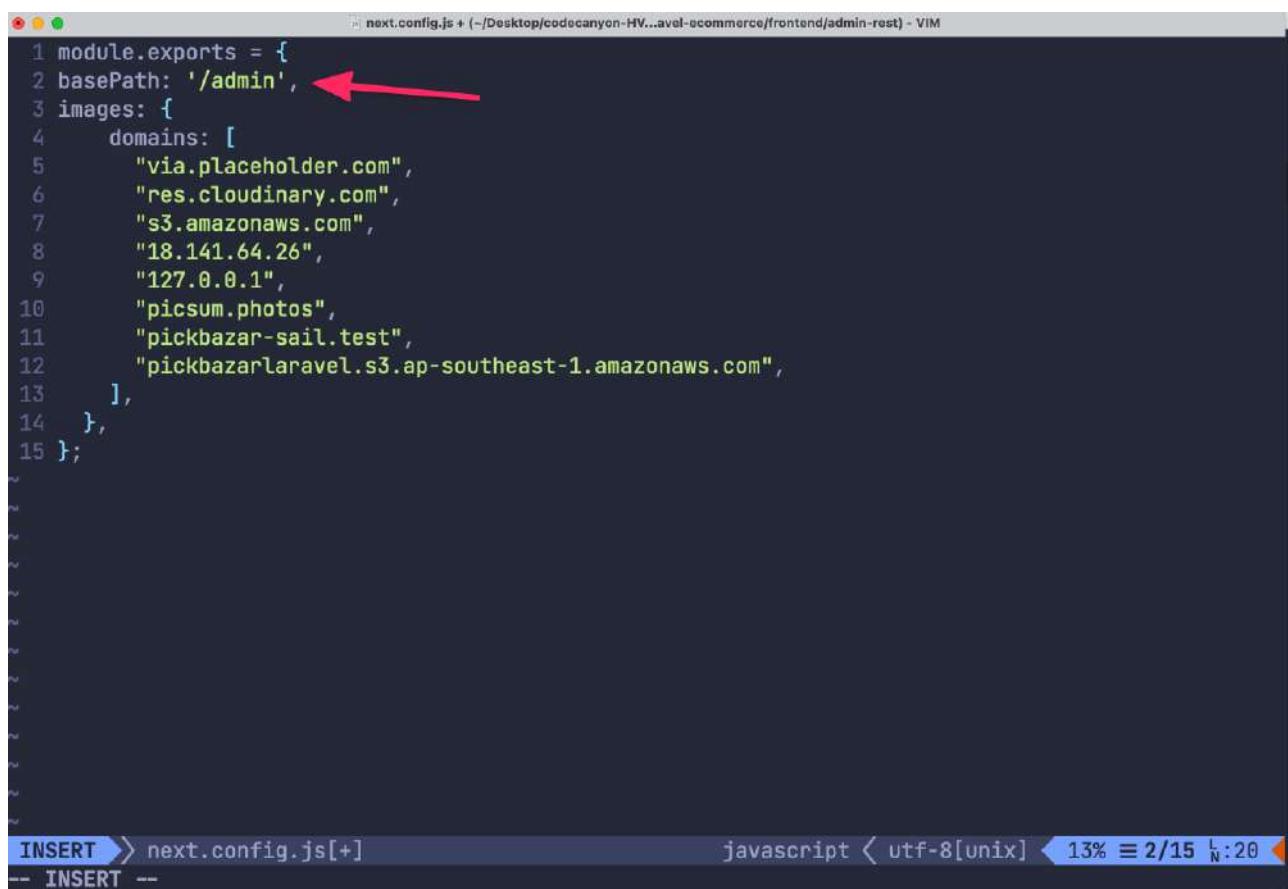
We'll suggest you build the frontend part on your computer and then move the build file to the server using git and github.

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'



```
next.config.js + (~/Desktop/codecanyon-HV...avel-commerce/frontend/admin-rest) - VIM
1 module.exports = {
2   basePath: '/admin', ←
3   images: {
4     domains: [
5       "via.placeholder.com",
6       "res.cloudinary.com",
7       "s3.amazonaws.com",
8       "18.141.64.26",
9       "127.0.0.1",
10      "picsum.photos",
11      "pickbazar-sail.test",
12      "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
13    ],
14  },
15};
```

INSERT > next.config.js[+] javascript < utf-8[unix] 13% 2/15 L:20
-- INSERT --

Again,

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

add `basePath` for '/admin'

```
1 module.exports = {
2   basePath: '/admin', // ←
3   images: {
4     domains: [
5       "via.placeholder.com",
6       "res.cloudinary.com",
7       "s3.amazonaws.com",
8       "18.141.64.26",
9       "127.0.0.1",
10      "picsum.photos",
11      "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
12    ],
13  },
14
15  webpack(config, options) {
16    config.module.rules.push({
17      test: /\.graphql$/,
18      exclude: /node_modules/,
19      use: [options.defaultLoaders.babel, { loader: "graphql-let/loader" }],
20    });
21
22    config.module.rules.push({
23      test: /\.graphqls$/,
24      exclude: /node_modules/,
25      use: ["graphql-let/schema/loader"],
26    });
27
28    config.module.rules.push({
29
30      test: /\.js$/,
31      exclude: /node_modules/,
32      use: ["babel-loader"]
33    });
34  }
35
36  export default config;

```

INSERT next.config.js[+]

-- INSERT --

javascript < utf-8[unix] 5% 2/36 ↵:1

Install & Build

go to your `root` folder again

Clean previous build file,

```
yarn clean
```

To install all the npm packages run this command,

```
yarn
```

Build the project

For `REST api`

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

And for `GraphQL`,

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

Upload to GitHub

Use this command to add all files to git,

```
git add .
```

```
git commit -m "Build frontend"
```

```
git push origin main
```

Step 6: Upload Frontend & Run

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

```
git pull origin main
```

Then install all the node packages,

```
yarn
```

Run frontend app

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

```
pm2 start 'all'
```

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

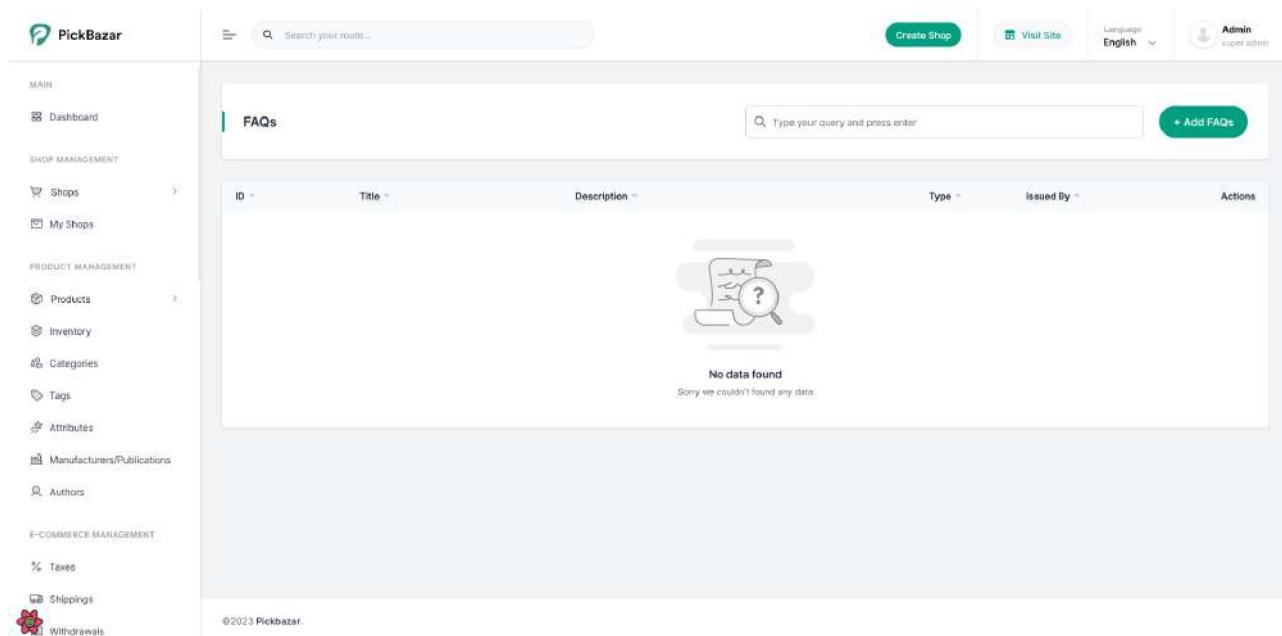
FAQ Page

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.

Customer End

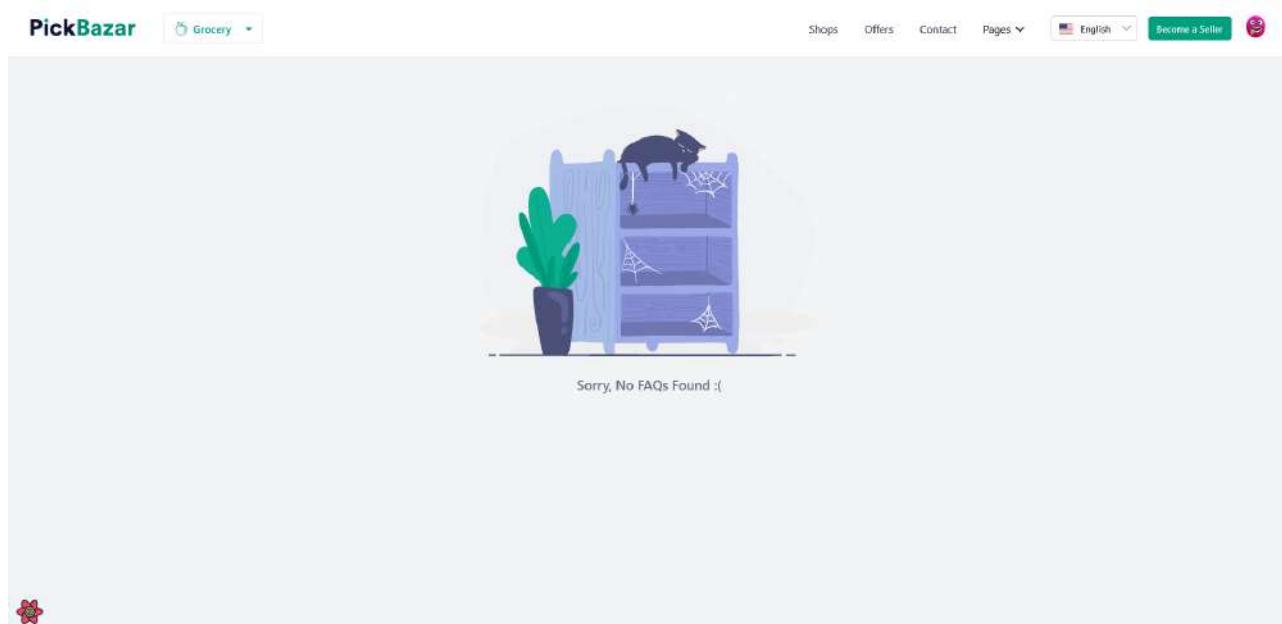
You find FAQ here [/help](#).

Admin end empty FAQ lists.



The screenshot shows the PickBazar Admin dashboard. On the left, there's a sidebar with categories like MAIN, SHOP MANAGEMENT, PRODUCT MANAGEMENT, and E-COMMERCE MANAGEMENT. The main area is titled 'FAQs' and contains search and filter fields. A large central area displays a message: 'No data found' with the subtext 'Sorry we couldn't find any data'. There's also a small icon of a book with a question mark.

Shop end empty FAQ lists.



The screenshot shows the PickBazar Shop interface. At the top, there are navigation links for Shops, Offers, Contact, Pages, Language (English), and Become a Seller. Below the header is a decorative illustration of a purple cabinet with a black cat and a green plant. A message at the bottom center says 'Sorry, No FAQs Found :(

FAQ create or edit settings.

MAIN

SHOP MANAGEMENT

PRODUCT MANAGEMENT

E-COMMERCE MANAGEMENT

MAIN

Create FAQ

Description

Add Store Notice necessary information from here

Title*

Description*

Add FAQ

MAIN

SHOP MANAGEMENT

PRODUCT MANAGEMENT

E-COMMERCE MANAGEMENT

MAIN

Edit Store Notice

Description

Edit Store Notice necessary information from here

Title*

Neque porro quisquam est

Description*

Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur?

Back

Update FAQ

FAQ admin lists.

ID	Title	Description	Type	Issued By	Actions
#ID: 2	Neque porro quisquam est	Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, c...	global	Super Admin	
#ID: 1	Quis autem vel eum iure reprehenderit qui...	Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit ...	global	Super Admin	

FAQ shop lists.

Quis autem vel eum iure reprehenderit qui.	
Neque porro quisquam est	

Vendor End

You find FAQ here `/shops/shopName/faqs`.

Vendor end empty FAQ lists.

PickBazar

- Refunds
- ORDER MANAGEMENT
 - Orders
 - Transactions
- FEATURE MANAGEMENT
 - Store Notice
 - Message
- FEEDBACK CONTROL
 - Reviews
 - Questions
- USER CONTROL
 - Staff
- PROMOTIONAL CONTROL
 - Flash Sale
- LAYOUT/PAGE MANAGEMENT

FAQs

Search your route...

Create Shop Visit Site Language English Store Owner

FAQs

+ Add FAQ

ID	Title	Description	Type	Issued By	Actions
 No data found Sorry we couldn't find any data					

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

10.2.0

Vendor end FAQ create or edit settings.

PickBazar

- MAIN
 - Dashboard
- PRODUCT MANAGEMENT
 - Products
 - Inventory
 - Attributes
 - Manufacturers/Publications
 - Authors
- FINANCIAL MANAGEMENT
 - Withdrawals
 - Refunds
- ORDER MANAGEMENT
 - Orders
 - Transactions
- FEATURE MANAGEMENT
 - Store Notice

FAQs

Search your route...

Create Shop Visit Site Language English Store Owner

FAQs

+ Add FAQ

ID	Title	Description	Type	Issued By	Actions
#ID: 6	What is the delivery process for furniture p...	We offer convenient and reliable furniture delivery services. After m...	shop	Furniture Shop	  
#ID: 7	Do you provide warranty coverage for furnit...	Yes, many of our furniture items come with manufacturer warranties...	shop	Furniture Shop	  

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

10.2.0

Vendor Shop end FAQ lists.

FAQs

Home > FAQs

Quick wafting zephyrs vex bold Jim.

Quick wafting zephyrs vex bold Jim. Quick zephyr blow, vexing daft Jim. Sex-charged fop blew my junk TV quiz. How quickly daft jumping zebras vex. Two driven jocks help fax my big quiz. Quick, Baz, get my woven flask jodhpurs! "Now fax quiz Jack!" my brave ghost pled.

Five quacking zephyrs.

+



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

Contact

A good contact page should be easy to find, explain why someone should contact your business, describe how your business can help solve the visitors' problems, include an email and phone number so visitors can get in touch with you on their first attempt, include a short form using fields that'll help your business understand who's contacting them.

Customer End

Contact page details admin part. You find the details here [/settings/company-information](#).

SC Vouchers

FEEDBACK CONTROL

 Reviews Questions

PROMOTIONAL MANAGEMENT

 Coupons Flash sale

FEATURE MANAGEMENT

 Message Store Notice

SITE MANAGEMENT

 Settings

General settings

Payment settings

SEO settings

Events settings

Shop settings

 Company Information

Settings

General settings Payment settings SEO settings Events settings Shop settings Company information

Address

Add your address from here

Set location from map

New York Ave, Quezon City, Metro Manila, Philippines

Contact Number

234456896

Website

https://www.youtube.com/

Email

demo@dmeo.com

Select social platform

 Facebook

Add profile url

https://www.facebook.com/

Remove

 Save Settings

Select social platform

 Youtube

Add profile url

https://www.youtube.com/

Remove

Select social platform

 Instagram

Add profile url

https://www.instagram.com/

Remove

Select social platform

 Twitter

Add profile url

https://twitter.com/

Remove

 Add New Social Profile

Footer

Change your Footer Information from here

Site Link

Copyright Text

External Text

External Link

©2023 Pickbazar.

 Contact page in shop end.



Address
New York Ave, Quezon City, Metro Manila, Philippines

Phone
234456896

Email Address
demo@dmeo.com

Website
<https://www.youtube.com> [Visit This Site](#)

Follow Us



How can we improve your experience?

Name Email

Subject

Description

Submit

PickBazar**Explore**

New York Ave, Quezon City, Metro Manila, Philippines

demo@dmeo.com
234456896**About us**Sitemap
Bookmarks
Sign In/Out**Customer Service**FAQ & Help
Returns
Accessibility
Contact Us
Store Pickup**Our information**Privacy policy update
Terms & conditions
Return Policy
Shipping**Subscribe Now**

Subscribe your email for newsletter and featured news, based on your interest.

Write your email here

©2023 Pickbazar.

stripe

Vendor End

Contact page details vendor part. You find the details here </shopName/edit>.

MAIN

Dashboard

PRODUCT MANAGEMENT

- Products
- Inventory
- Attributes
- Manufacturers/Publications
- Authors

FINANCIAL MANAGEMENT

Withdrawals

Refunds

ORDER MANAGEMENT

Orders

Transactions

FEATURES MANAGEMENT

Store Notice

Message

Edit Shop

Logo

Upload your shop logo from here



Upload an Image or drag and drop
(PNG, JPG)

Cover image

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



Upload an Image or drag and drop
(PNG, JPG)

<https://pickbazar.com/southeast-1/join?newsr>

Update

Basic Info

Add some basic info about your shop from here

Name

Grocery Shop

edit

Slug

grocery-shop

edit

Description

The grocery shop is the best shop around the city. This is being run under the store owner and our aim is to provide fresh and quality product and hassle free customer service.

Payment Info

Add your payment information from here

Account Holder Name

Grocery Shop

Account Holder Email

grocer@demo.com

Bank Name

Bank6

Account Number

231321635465465

Shop Address

Add your physical shop address from here

Country

USA

City

Freeport

State

Illinois

ZIP

61032

Street Address

1988 Spinnaker Lane

Email Notification

Set your email notification for messaging feature

Notification email

 Enable Notification

Shop Settings

Add your shop settings information from here.

Contact Number

123456789

Website

<https://redq.io>

Select social platform

Url

Social Profile Settings

Contact page in shop end. You find the details here [/shops/shopName/contact](#).

Select social platform

Url

Shops Offers Contact Pages Join Become a Seller

We're always here for you

Grocery Shop is going to receive an email. we'd love to hear from you.

Name Email

Subject

Description

Submit

Address
1986 Spinaker Lane, Illinois, Freeport, 61032, USA

Phone
123456789

Website
<https://edq10.com> Visit This Site

Follow Us



PickBazar

New York Ave, Quezon City, Metro Manila, Philippines

demo@dmepo.com
234456896

Explore

About us
Sitemap
Bookmarks
Sign In/Out

Customer Service

FAQ & Help
Returns
Accessibility
Contact Us
Store Pickup

Our information

Privacy policy update
Terms & conditions
Return Policy
Sitemap

Subscribe Now

Subscribe your email for newsletter and featured news based on your interest.

Write your email here

©2023 Pickbazar.

stripe

Terms and conditions

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.

You find terms and conditions here [/terms](#).

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

Customer End

Admin end empty Terms and Condition lists.

Manufacturers/Publications Authors

E-COMMERCE MANAGEMENT

 Taxes Shipments Withdrawals Refunds

LAYOUT/PAGE CONTROL

 Home pages / Types FAQs Terms And Conditions

All Terms

Add new Terms

ORDER MANAGEMENT

 Orders Create Order Transactions

Terms And Conditions

Type your query and press enter

+ Add Terms And Conditions

ID - Title - Description - Type - Issued By - Status - Approval Action Actions



No data found

Sorry we couldn't find any data

 *Shop end empty Terms and Condition lists.*



Sorry, No Terms & Conditions Found :(



 *Create or edit terms and condition form.*

PickBazar

- Manufacturers/Publications
- Authors
- E-COMMERCE MANAGEMENT
 - Taxes
 - Shipments
 - Withdrawals
 - Refunds
- LAYOUT/PAGE CONTROL
 - Home pages / Types
 - FAQs
 - Terms And Conditions
- ORDER MANAGEMENT
 - Orders
 - Create Order
 - Transactions

©2023 Pickbazar.

PickBazar

Create Terms And Conditions

Description Add Terms and Condition necessary information from here

Title*

Description*

Add Terms & Condition

PickBazar

- MAIN
- Dashboard
- SHOP MANAGEMENT
 - Shops
 - My Shops
- PRODUCT MANAGEMENT
 - Products
 - Inventory
 - Categories
 - Tags
 - Attributes
- Manufacturers/Publications
- Authors
- E-COMMERCE MANAGEMENT
 - Taxes
 - Shipments
 - Withdrawals

©2023 Pickbazar.

Edit Store Notice

Description Edit Terms and Condition necessary information from here

Title* But I must explain to you how all this mistaken.

Description* On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal blame belongs to those who fail in their duty through weakness of will, which is the same as saying through shrinking from toil and pain. These cases are perfectly simple and easy to distinguish.

Back **Update Terms & Conditions**

Admin end terms and condition lists.

PickBazar

- MAIN
- Dashboard
- SHOP MANAGEMENT
 - Shops
 - My Shops
- PRODUCT MANAGEMENT
 - Products
 - Inventory
 - Categories
 - Tags
 - Attributes
- Manufacturers/Publications
- Authors
- E-COMMERCE MANAGEMENT
 - Taxes
 - Shipments
 - Withdrawals

©2023 Pickbazar.

Terms And Conditions

Type your query and press enter

+ Add Terms And Conditions

ID	Title	Description	Type	Issued By	Status	Approval Action	Actions
#ID: 2	But who has any right to...	In a free hour, when our p...	global	Super Admin	approved		
#ID: 1	But I must explain to you...	Nor again is there anyone...	global	Super Admin	approved		

Shop end terms and condition lists.

The screenshot shows the 'Terms & Condition' page on the PickBazar platform. At the top, there are navigation links for 'Shops', 'Offers', 'Contact', 'Pages', language selection ('English'), and a 'Become a Seller' button. Below the header, the title 'Terms & Condition' is displayed, along with a breadcrumb trail: 'Home > Terms & Condition'. The main content area contains three entries:

- But I must explain to you how all this mistaken.**
Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain, but because occasionally circumstances occur in which toll and pain can procure him some great pleasure. To take a trivial example, which of us ever undertakes laborious physical exercise, except to obtain some advantage from it?
- But who has any right to find fault.**
In a free hour, when our power of choice is untrammelled and when nothing prevents our being able to do what we like best, every pleasure is to be welcomed and every pain avoided. But in certain circumstances and owing to the claims of duty or the obligations of business it will frequently occur that pleasures have to be repudiated and annoyances accepted.



Not approved terms and condition item.

The screenshot shows the 'Terms And Conditions' list on the PickBazar Admin dashboard. The left sidebar includes sections for MAIN (Dashboard), SHOP MANAGEMENT (Shops, My Shops), PRODUCT MANAGEMENT (Products, Inventory, Categories, Tags, Attributes, Manufacturers/Publications, Authors), and E-COMMERCE MANAGEMENT (Taxes, Shipments, Withdrawals). The main content area displays a table of terms and conditions:

ID	Title	Description	Type	Issued By	Status	Approval Action	Actions
#ID: 3	On the other hand, we d...	On the other hand, we de...	global	Super Admin	approved	✓ ⟳	✓ ✗ ...
#ID: 2	But who has any right to...	In a free hour, when our p...	global	Super Admin	waiting for approval	⟳ ⟳	✓ ✗ ...
#ID: 1	But I must explain to you...	Nor again is there anyone...	global	Super Admin	approved	✓ ⟳	✓ ✗ ...

At the bottom of the page, there is a footer note: '©2023 Pickbazar.' and a small decorative flower icon.

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

Terms & Condition

Home > Terms & Condition

But I must explain to you how all this mistaken.

On the other hand, we denounce with righteous

But I must explain to you how all this mistaken.

Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain, but because occasionally circumstances occur in which toil and pain can procure him some great pleasure. To take a trivial example, which of us ever undertakes laborious physical exercise, except to obtain some advantage from it?

On the other hand, we denounce with righteous

On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal blame belongs to those who fall in their duty through weakness of will, which is the same as saying through shrinking from toil and pain. These cases are perfectly simple and easy to distinguish.



Vendor End

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

Inactive terms and conditions.

[Create Shop](#)
[Visit Store](#)
Language English
Store Owner

Furniture Shop
(verified)

Registered Since
June 27, 2021

55
Total Products

0
Total Orders

Bio
furniture shop

\$0.00
Gross Sales

10%
Admin Commission Rate

Name
furniture shop

Email
furniture@demo.com

Bank
Bank1

Account No.
1121213131414141

You find terms and conditions settings here [/settings/shop](#).

The screenshot shows the PickBazar admin interface. On the left, there's a sidebar with categories like ORDER MANAGEMENT, FEEDBACK CONTROL, PROMOTIONAL MANAGEMENT, FEATURE MANAGEMENT, and SITE MANAGEMENT. Under SITE MANAGEMENT, 'Settings' is selected, which further branches into General settings, Payment settings, SEO settings, Events settings, Shop settings (highlighted in green), and Company information. The main content area is titled 'Settings' and has tabs for General settings, Payment settings, SEO settings, Events settings, Shop settings (active), and Company information. A section titled 'Delivery Schedule' allows adding delivery times with descriptions. Another section titled 'Shop Settings' contains checkboxes for enabling product review systems, Google Map addresses, and vendor terms & conditions. A 'Save Settings' button is at the bottom right.

Active terms and conditions.

The screenshot shows the PickBazar store owner dashboard. On the left, there's a sidebar with categories like ORDER MANAGEMENT, FEATURES MANAGEMENT, FEEDBACK CONTROL, USER CONTROL, PROMOTIONAL CONTROL, LAYOUT/PAGE MANAGEMENT, and a 'FAQs' section. The main content area features a large image of a modern living room with a blue sofa and wooden furniture. Below the image, the store name 'Furniture Shop' is displayed along with its email (store_owner@demo.com) and address (588 Finwood Road, East Dover, New Jersey, 08753, USA). A phone number (+213 42 12 12 21) and an 'Edit Shop' button are also present. The store's bio indicates it was registered since June 27, 2021. It lists 55 total products, 0 total orders, and 10% admin commission rate. Payment information shows gross sales of \$0.00 and current balance of \$0.00. At the bottom, there's a 'FAQs' section and a 'Terms And Conditions' link.

Admin end vendor terms and condition lists.

PickBazar

Search your route...

Create Shop Visit Site Language English Admin super admin

- Manufacturers/Publications
- Authors
- E-COMMERCE MANAGEMENT
 - Taxes
 - Shipments
 - Withdrawals
 - Refunds
- LAYOUT/PAGE CONTROL
 - Home pages / Groups
 - FAQs
 - Terms And Conditions
 - All Terms
 - Add new Terms
- ORDER MANAGEMENT
 - Orders
 - Create Order
 - Transactions

©2023 Pickbazar.

Vendor end terms and conditions list.

PickBazar

Search your route...

Create Shop Visit Site Language English Store Owner

- ORDER MANAGEMENT
 - Orders
 - Transactions
- FEATURES MANAGEMENT
 - Store Notice
 - Message
- FEEDBACK CONTROL
 - Reviews
 - Questions
- USER CONTROL
 - Staff
- PROMOTIONAL CONTROL
 - Flesh sale
- LAYOUT/PAGE MANAGEMENT
 - FAQs
- Terms And Conditions

©2023 Pickbazar.

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

Terms & Condition

[Home](#) > Terms & Condition

Donec quam felis, ultricies nec

Donec quam felis, ultricies nec

Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu.



Explore

About us
Sitemap
Bookmarks
Sign In/Join

Customer Service

FAQ & Help
Returns
Accessibility
Contact Us
Store Pickup

Our Information

Privacy Policy Update
Terms & Conditions
Return Policy
Sitemap

Subscribe Now

Subscribe your email for newsletter and featured news based on your interest.

Write your email here

©2023 PickBazar.

stripe

Existing Project Update Guide

NB: Before Update Your Existing Project Please Read the Full Documentation.

Why Update Your PickBazar Project?

Before we dive into the update process, let's understand why it's crucial to keep your PickBazar Laravel project up to date:

- Security: Before we dive into the update process, let's understand why it's crucial to keep your PickBazar Laravel project up to date: updates often include important security patches. Failing to update your project may leave it vulnerable to security threats.
- Bug Fixes: Updates address bugs and issues reported by the community. Staying up to date ensures that your project remains stable and reliable.

- **Performance:** New versions of Before we dive into the update process, let's understand why it's crucial to keep your PickBazar Laravel project up to date: may introduce performance improvements that can enhance the speed and efficiency of your application.
- **New Features:** Before we dive into the update process, let's understand why it's crucial to keep your PickBazar Laravel project up to date.

Preparing for the Update

Before you begin the update process, there are several crucial steps to take to ensure a smooth transition:

- **Backup Your Project**

Always start by creating a backup of your project. This includes both your codebase and your database. In case anything goes wrong during the update, you can easily restore your project to its previous state.

- **Review PickBazar Laravel change Log**

Carefully review the change Log for the version you plan to update to. This will help you understand what changes and new features are introduced and what breaking changes might affect your project.

- **Update Dependencies**

Update your project's dependencies, including PHP, Composer.

General FAQs

How to configure Stripe payment gateway?

To configure `stripe` payment gateway,

1. At first, create a developer account from stripe developer dashboard (<https://dashboard.stripe.com/>)

2. log in to that account

3. Then go to the `Developer -> API Keys` section and on that section create an API key. After creating, you'll get one `Secret key` and one `Publishable key`.

The screenshot shows the Stripe API Keys page. On the left, there's a sidebar with 'New Business' selected. Under 'Developers', 'API keys' is highlighted with a red arrow. The main area shows 'Standard keys' with two entries:

NAME	TOKEN	LAST USED	CREATED
Publishable key	pk_test_26a02U...9vgNer1...	Jul 18	Apr 11
Secret key	(REDACTED)	Jul 18	Jul 10

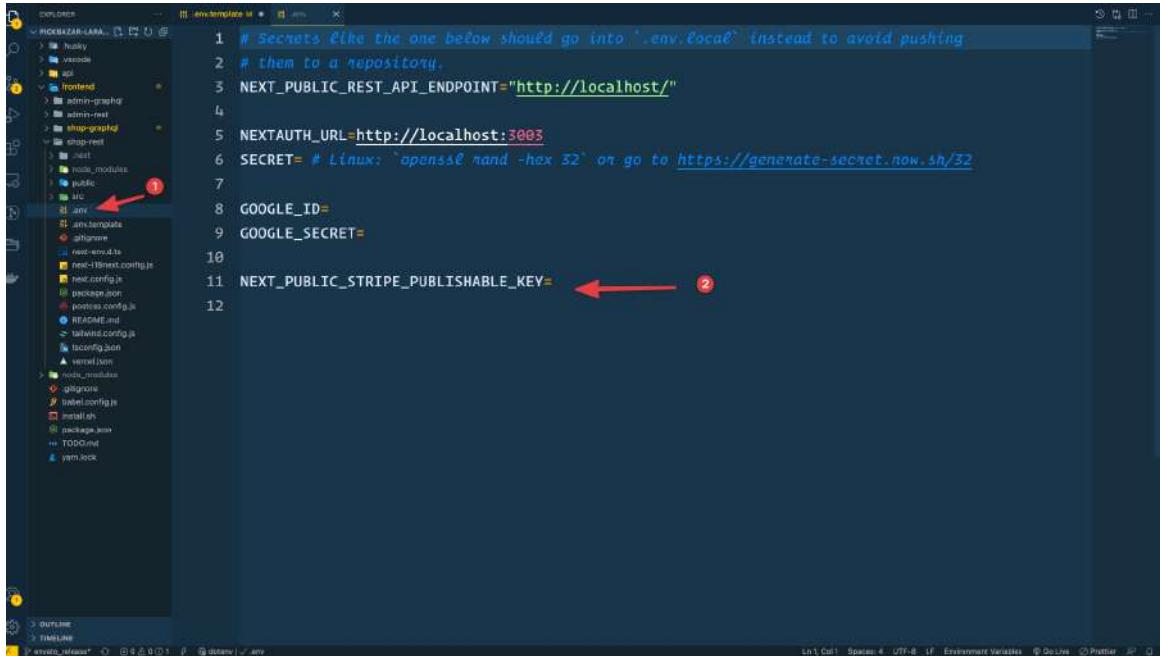
Below this is a section for 'Restricted keys' which is currently empty.

4. Then copy `Secret key` to your pickbazar `api->.env`

The screenshot shows the VS Code interface with the '.env' file open in the editor. The file contains several environment variables. Red arrows point to three specific lines:

```
AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=
AWS_BUCKET=
STRIPE_API_KEY= (arrow 3)
LIGHHOUSE_CACHE_ENABLE=false
MEDIA_DISK=public
SHOP_URL=
ADMIN_EMAIL=support@example.com
GOOGLE_CLIENT_ID=
GOOGLE_CLIENT_SECRET=
GOOGLE_REDIRECT_URI=http://localhost/login/google/callback
FACEBOOK_CLIENT_ID=
FACEBOOK_CLIENT_SECRET=
FACEBOOK_REDIRECT_URI=https://localhost/login/facebook/callback
ACTIVE_OTP_PROVIDER=messagebird
TWILIO_AUTH_TOKEN=
TWILIO_ACCOUNT_SID=
TWILIO_VERIFICATION_SID=
```

5. And similarly, add `PUBLISHABLE_KEY` to your `frontend/shop-rest/.env` for rest and `frontend/shop-graphql/.env` for graphql.



```
1 # Secrets like the one below should go into '.env.local' instead to avoid pushing
2 # them to a repository.
3 NEXT_PUBLIC_REST_API_ENDPOINT="http://localhost/"
4
5 NEXTAUTH_URL=http://localhost:3005
6 SECRET= # Linux: `openssl rand -hex 32` or go to https://generate-secret.now.sh/32
7
8 GOOGLE_ID=
9 GOOGLE_SECRET=
10
11 NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY= ②
12
```

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

For REST API

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

And for GraphQL,

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

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

Ans: Please run `yarn clean` then `yarn` then run `yarn dev:shop-rest`

I am changing schema files but changes is not working

Your changes might not work because schema is cached. SO you might need to clear schema cache using the below command `php artisan lighthouse:clear-cache`.

Changing .env files but not getting the changes

Run Below command `php artisan optimize:clear`

Changing route but not getting the changes.

Run `php artisan optimize:clear` or `php artisan route:clear`

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

In some cases `STRIPE_API_KEY` value can't read from .env in those cases you have to put the key in the config file directly in `api/packages/marvel/src/Config/laravel-omnipay.php`

6. Getting error on forget password email sending

Make sure you have run the `php artisan marvel:install` commands successfully and copied the necessary email templates to your resources folder. You can also do it by `php artisan marvel:copy-files` command.

NB: This same issue can occur during order creation.

Can I use it with my existing laravel?

Yes, you can. Follow the below steps.

- Make sure you are using laravel 8.
- Copy `api/packages` folder from the downloaded files into your laravel root folder.
- Put below code in your laravel `composer.json` file into `require` section.

```
"ignited/laravel-omnipay": "dev-master",
"omnipay/common": "dev-master",
"omnipay/stripe": "dev-master",
"pickbazar/shop": "dev-master"
```

- Put below code in bottom of your `composer.json`. If you already have an `repositories` section then put code inside `repositories` to your existing `repositories`

```
"repositories": {
    "pickbazar/shop": {
        "type": "path",
        "url": "packages/marvel"
    },
    "ignited/laravel-omnipay": {
        "type": "path",
        "url": "packages/laravel-omnipay"
    },
    "omnipay/common": {
        "type": "path",
        "url": "packages/omnipay-common"
    },
    "omnipay/stripe": {
        "type": "path",
        "url": "packages/omnipay-stripe"
    }
}
```

- Now run `composer install`
- Copy necessary env variables from `.env.example` to your `env` file.
- Run `php artisan marvel:install` and follow necessary steps.
- To run server `php artisan serve`
- For image upload to work properly you need to run `php artisan storage:link`.

Why am I getting `Access denied for user?`

navigate to `api` then run `./vendor/bin/php down -v`. It will delete any of your existing mysql volumes. Now run `./vendor/bin/php up -d` on same directory or run `bash install.sh` on root directory

Why am I getting permission issue during deployment?

Run below commands for fixing permission issue in your laravel app during deployment

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

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

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

Run `php artisan optimize:clear`

How to resolve the Load More Infinity loading issue?

If you click on the Load More button and the button is spinning continuously, then check API request from the network tab. If it is HTTP and your API is hosted on HTTPS, then open,

```
/users/tareqmahmud/tmbox/tmserver/larabox/work/redq/laravel-
ecommerce/api/app/Providers/AppServiceProvider.php
```

And add this code to boot method,

```
if (!\App::environment('local')) {
    $this->app['request']->server->set('HTTPS', true);
}
```

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

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

Both uploading systems follow this procedure,

API

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

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

If you're using windows and your API is running with ports like `localhost:8000` or `127.0.0.1:8000`, then make sure you add the domain with port to `APP_URL`. Like this `API_URL=http://localhost:8000`

After that, clear your API cache by using this command from the `api` folder,

```
php artisan optimize:clear
```

Then link the storage system by using this command,

```
php artisan storage:link
```

If you're using AWS S3, then update `MEDIA_DISK=s3` and add all `AWS_` credentials For AWS s3, make sure your properly setup permission of the bucket with ACL enable by follow this link -> <https://stackoverflow.com/a/70603995/2158023>

Edit Object Ownership Info

Object Ownership

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner preferred

If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

Object writer

The object writer remains the object owner.

[Cancel](#)

[Save changes](#)

Admin

Then add your domain to your `pickbazar-laravel -> admin -> rest -> next.config.js` for `rest` or `pickbazar-laravel -> admin -> graphql -> next.config.js` for `graphql`

```
images: {
  domains: [
    "YOUR_DOMAIN.COM", ←
    "via.placeholder.com",
    "res.cloudinary.com",
    "s3.amazonaws.com",
    "18.141.64.26",
    "127.0.0.1",
    "localhost",
    "picsum.photos",
    "pickbazar-sail.test",
    "pickbazarlaravel.s3.ap-southeast-1.amazonaws.com",
    "chawkbazarlaravel.s3.ap-southeast-1.amazonaws.com",
    "lh3.googleusercontent.com",
  ],
},
```

If you're using AWS S3, then also add your root S3 domain like this,

```
images: {
  domains: [
    "YOUR_DOMAIN.COM", ← Like this
    "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",
  ],
},
```

Shop

Similarly, add your domain to your `pickbazar-laravel -> shop -> next.config.js`

```
images: {
  domains: [
    "YOUR_DOMAIN.COM", ← Like this
    "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", ← Like this
    "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",
  ],
},
```

Build Frontend,

If you run in dev mode, then re-run will affect your update. But if you're running as a production mode, then you've to rebuild the project and re-run again. Otherwise, image upload will not work.

<https://pickbazar-doc.vercel.app/vps-server#build-project>

After building the project, replace your server shop and admin folder using this rebuild folder and re-run the app using this command,

```
pm2 restart all
```

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

There can be several reasons that throw 502 errors for the frontend in the production server.

Reason 1:

On your Nginx file, a specific port is set for shop or the admin, so when you run the script as a PM2` instance, the script has to be run that specific port; otherwise, your domain/subdomain will throw a 502 error.

PORT:

```
3000 -> Shop Rest  
3001 -> Admin Rest  
  
3000 -> Shop GraphQL  
3004 -> Admin GraphQL
```

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

```
pm2 stop 'all'
```

then go to the pickbazar folder from your server and try to run the script using yarn and check which port it is running,

Rest:

```
yarn start:admin-rest
```

```
yarn shop-rest
```

GraphQL:

```
yarn start:admin-gql
```

```
yarn shop-gql
```

And, check if the port matched with the Nginx port or not? If not matched, then change the port to **Nginx** config.

Reason 2:

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

```
pm2 stop 'all'
```

then go to the **pickbazar** folder from your server and try to run the script using yarn and check which port it is running,

Rest:

```
yarn start:admin-rest
```

```
yarn shop-rest
```

GraphQL:

```
yarn start:admin-gql
```

```
yarn shop-gql
```

With this command, it'll give you an error that no build file is found. To resolve that, rebuild your project and then rerun the scripts.

<https://pickbazar-doc.vercel.app/vps-server#frontend-project-build>

How to resolve javascript heap out of memory issue?

If you're using cPanel based server, then you maybe get an error like "call_and_retry_last allocation failed - javascript heap out of memory".

To resolve that issue, you've to add a limit of the memory bound,

Please check this,

<https://stackoverflow.com/questions/38558989/node-js-heap-out-of-memory>

<https://support.snyk.io/hc/en-us/articles/360002046418-JavaScript-heap-out-of-memory>

<https://blog.openreplay.com/javascript-heap-out-of-memory-error>

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

For uploading, the php8.1-gd library is required. So make sure you install the php8.1-gd library on your server.

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

To resolve that issue, you've to move the `Laravel-ecommerce` folder from `Pickbazar Laravel - React Next REST & GraphQL Ecommerce` to a generic folder where you store your code. After that, use this command from your `Laravel-ecommerce` folder,

```
chmod +x install.sh  
bash install.sh
```

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

In general, this happens mainly when your upstream server fails to handle its processing power when there is lots of traffic or requires lots of operation. Please upgrade your server configuration to double to check the issue. Please check this video,

<https://www.youtube.com/watch?v=4pIqAz5bzX0>

If that doesn't solve the issue, then you've to redeploy again with a new ec2 instance.

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

For AWS s3, make sure your properly setup permission of the bucket with ACL enable by follow this link

<https://stackoverflow.com/a/70603995/2158023>

How to rebuild the project?

When you upload the code to the server as a production, NextJS fetch its code from the build file. So after deployed, when you update or change code on scripts, the updated scripts don't work at the front end until you rebuild the project again. NextJs works this way.

Now there are two ways you can follow to rebuild the project,

If your server has more than 2+ CPU core and 6GB+ of memory, then you can directly edit code at your server then rebuild the project from the server directly using this command,

At first go to `/var/www/pickbazar-laravel`

For `shop rest.`

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

for graphql `yarn build:shop-gql`

And for `admin rest.`

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

for graphql `yarn build:admin-gql`

After completing the build, restart the PM2 process using this command,

```
pm2 restart 'all'
```

But suppose your server doesn't have that processing power, and you try to build your scripts directly from the server. In that case, your server will be shut down during the build, and you've to restart the server.

In that case, you can rebuild the scripts at your local computer and upload them to your server. If you follow our [VPS Server Deployment](#) guide, then you already know the process. To build the scripts on your pc, follow this procedure,

At first, download the `pickbazar-laravel` folder from the server to your computer, and customize the code at your computer. Then use this command from the `root` folder,

To install all the packages,

```
yarn
```

Then build the shop rest,

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

for graphql `yarn build:shop-gql`

Then build the admin rest,

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

for graphql `yarn build:admin-gql`

Then delete the shop and admin folder from the server and upload your build shop and admin folder to the server at the exact same location and use this command to restart the PM2,

```
pm2 restart 'all'
```

How to increase upload size?

The API supports **10MB** of upload limit for a single file by default. If your upload file is more than **10MB**, then please follow this,

At first, login to the server using ssh, and then,

Edit,

```
/etc/php/8.1/fpm/php.ini
```

Then update,

```
upload_max_filesize = 256M  
post_max_size = 256M  
max_execution_time = 300
```

Then restart the PHP,

```
sudo service php8.1-fpm restart
```

Then update,

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

and Add **client_max_body_size 256M**

```
client_max_body_size 256M
```

```
server {  
    server_name [REDACTED];  
    client_max_body_size 256M; ←  
    add_header X-Frame-Options "SAMEORIGIN";  
    add_header X-XSS-Protection "1; mode=block";  
    add_header X-Content-Type-Options "nosniff";  
    index index.html index.htm index.php;  
    charset utf-8;
```

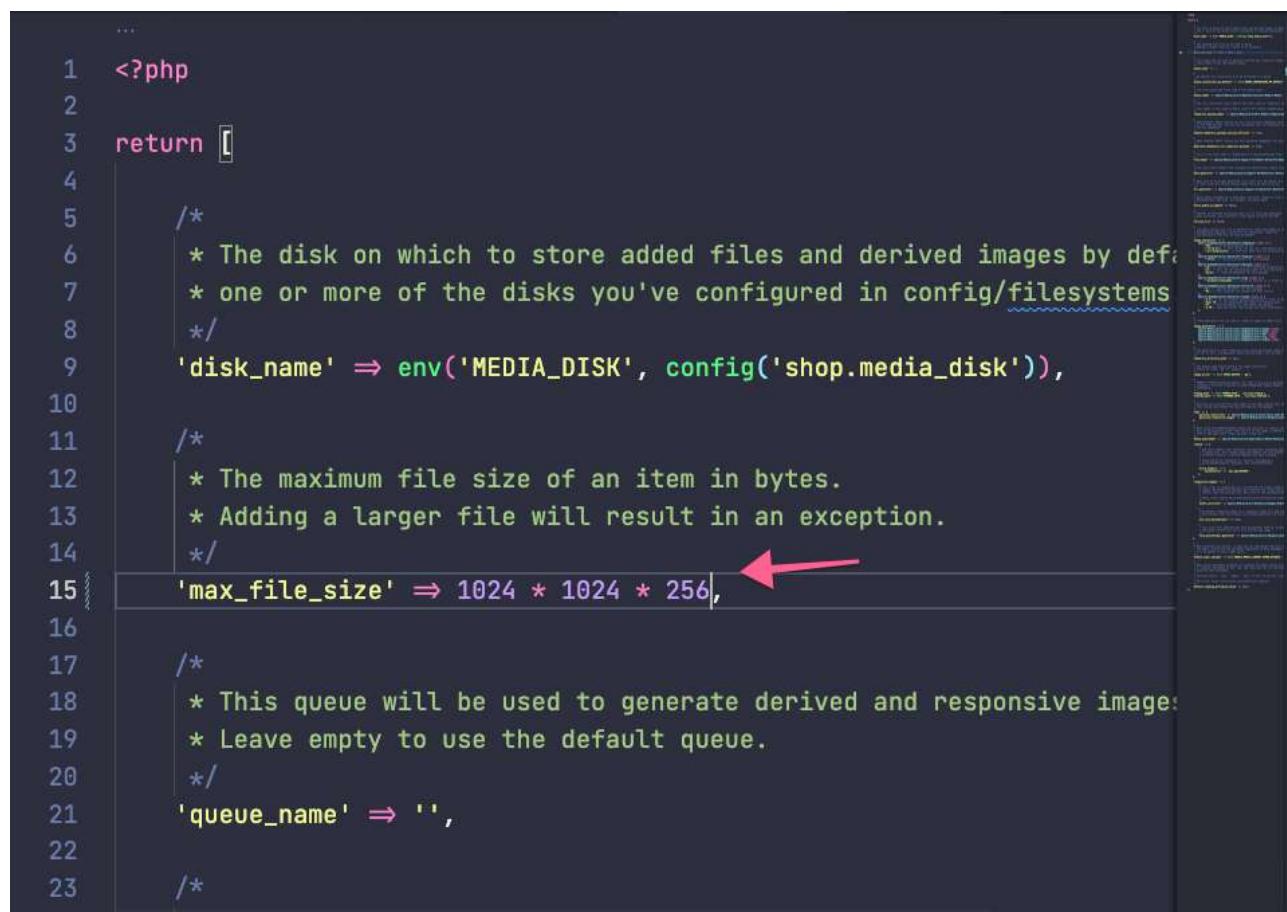
Then restart the server,

```
sudo nginx -t  
sudo service nginx restart
```

After that update,

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

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



```
1 <?php  
2  
3 return [  
4  
5     /*  
6      * The disk on which to store added files and derived images by def:  
7      * one or more of the disks you've configured in config/filesystems  
8      */  
9     'disk_name' => env('MEDIA_DISK', config('shop.media_disk')),  
10  
11    /*  
12     * The maximum file size of an item in bytes.  
13     * Adding a larger file will result in an exception.  
14     */  
15     'max_file_size' => 1024 * 1024 * 256, ←  
16  
17    /*  
18     * This queue will be used to generate derived and responsive images.  
19     * Leave empty to use the default queue.  
20     */  
21     'queue_name' => '',  
22  
23     /*
```

After that, run this command from the `api` folder,

```
php artisan optimize:clear
```

And for `admin`, increase `timeout` from,

```
admin/rest/src/utils/api/http.ts
```

```
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";      Bashar Nozibulla, 8 months ago ·
5 import Router from "next/router";
6
7 const http = axios.create({
8     baseURL: process.env.NEXT_PUBLIC_REST_API_ENDPOINT, // TODO: take this from env
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;
25    }
26);
27
```

```
timeout: 600000
```

After that [Rebuild Your Project](#) and restart again.

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

Please open your terminal and use this command,

```
brew services stop mysql
```

It'll stop any MySQL instance that is installed using brew.

Then use this command to check is there MySQL running or not,

```
mysql -v
```

If no MySQL is running then, go to,

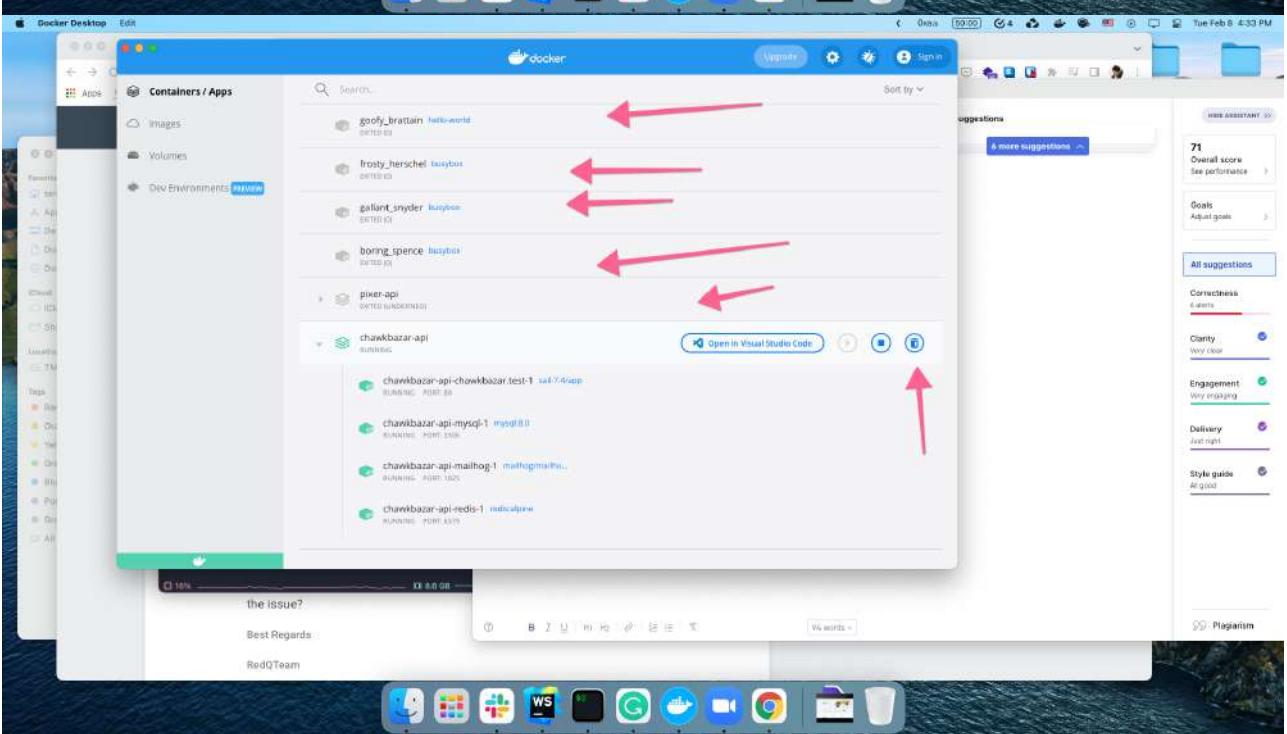
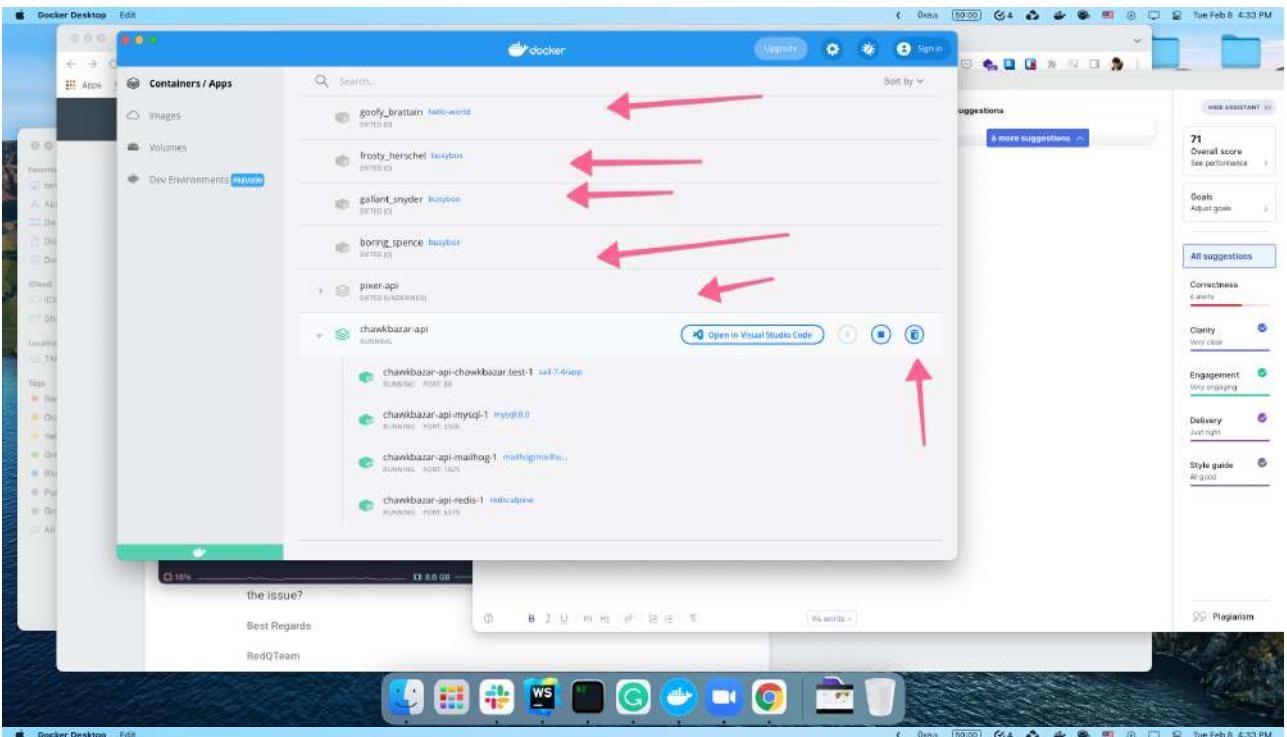
```
pickbazar-laravel -> api
```

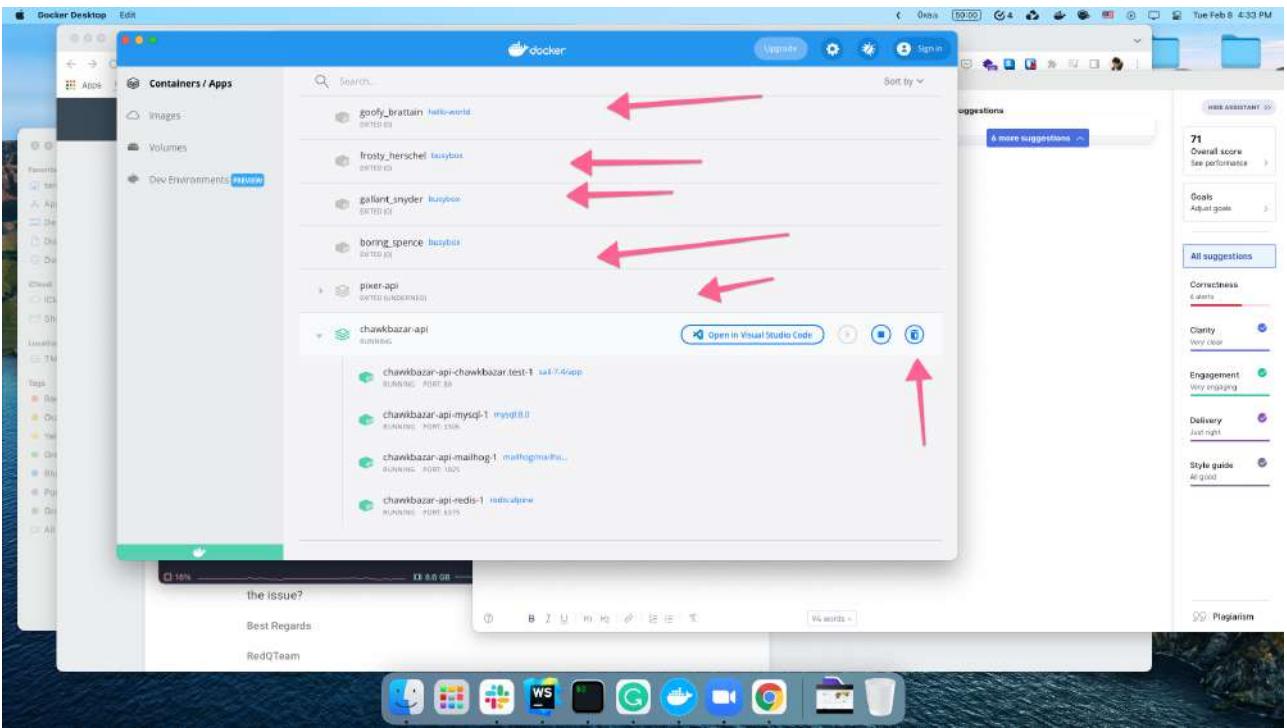
and use this command,

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

After that,

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





And after that remove,

```
pickbazar-laravel -> api -> .env
```

And then go to your root `pickbazar-laravel` folder and use this command again,

```
chmod +x install.sh
bash install.sh
```

How to remove the existing payment gateway?

You can remove the Stripe Payment from the available payment array,

```
shop/src/components/checkout/payment/payment-grid.tsx
```

Why `Checkout Place Order` button is disabled?

The place order button will be activated only when you fill in this information,

- OTP (Mobile number)
- Billing Address
- Shipping Address
- Delivery Schedule

- Payment Gateway

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

When you install the API using `php artisan marvel:install` or using `bash install.sh(docker)`, you'll get a prompt to create an admin account, and you've to create admin credentials that time. And then, you can use that credentials for the main admin account.

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

At first, remove all the existing `php 7.4` and its extensions by using this command,

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

Then remove the composer,

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

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

```
cd /var/www/pickbazar-laravel/api  
  
sudo chown -R $USER:www-data storage  
sudo chown -R $USER:www-data bootstrap/cache  
  
rm vendor -rf
```

Then install PHP 8, and its extensions,

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

```
sudo apt install php8.0-fpm php8.0-mysql
```

```
sudo apt install php8.0-mbstring php8.0-xml php8.0-bcmath php8.0-simplexml php8.0-intl php8.0-mbstring php8.0-gd php8.0-curl php8.0-zip
```

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'e21205b207c3ff031906575712edab6f13eb0b361f2085f1f1237b7126d785e826
a450292b6cf1d64d92e6563bbde02') { echo 'Installer verified'; }
else { echo 'Installer corrupt'; unlink('composer-setup.php'); }
echo PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

```
sudo mv composer.phar /usr/bin/composer
```

Then update 7.4 fpm to 8.0 fpm from,

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

```
charset utf-8;

# For API
location /backend {
    alias /var/www/pickbazar/api/public;
    try_files $uri $uri/ @backend;
    location ~ \.php$ {
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $request_filename;
        fastcgi_pass unix:/run/php/php8.0-fpm.sock;
    }
}

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

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

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

error_page 404 /index.php;

location ~ \.php$ {
    fastcgi_pass unix:/var/run/php/php8.0-fpm.sock;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
    include fastcgi_params;
}

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

Then install composer packages,

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

```
composer install
```

```
php artisan optimize:clear
```

```
php artisan marvel:install
```

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

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

Breaking changes

Purpose of this chapter is to help our users to identify the major changes he/she should notice before upgrading his/her existing application. It's always recommended to keep the `current site & database` backup before any upgradation.

Breaking Change Log Version: 11.0.0

Some new exciting feature is added like Flash sale, FAQs builder for super admin & vendor shop, Terms & Condition builder etc. For that reason there were some changes introduced in the database level along with API, Admin & Shop end. Here is the given key points to know before any upgradation,

- Pickbazar admin dashboard design facelifted
- Dashboard analytics section updated along with new analytics widgets
- Settings page is divided into multi parts.
- Flash sale builder added
- Terms & Condition builder added
 - Super admin can perform CRUD operation

- Vendor also can perform CRUD operation if and only if super admin enable this feature from settings.
- FAQs builder added
 - Super admin can add FAQs for global scope
 - Vendor can add FAQs based on his each shop
- Refund policy builder added
- Refund reason builder added, so that Super Admin can preset some refund reasons during customer asking for any refund.
- Product inventory route added
- Order transaction route added for better viewing about the recent transactions.
- User section divided into multiple layer for better management.

Some new database table is introduced along with some older table modification. So it's always safe to keep your existing database backup, then run migration for getting the new changes.

Breaking Change Log Version: 10.0.0

- Set SMS Events from settings for Admin, Vendor & Customer. [New Feature]
- Set Email Event from settings for Admin, Vendor & Customer. [New Feature]
 - Email Event Setting option [New Feature]
 - SMS Settings for both Graphql and REST [New Feature]
 - Added Setting Seeder [New Feature]
 - New Console Command `marvel:settings_seed` [New Feature]
 - Added Settings Dummy data seeder command [New Feature]
 - Added OpenAI support in every possible description like Create Shop → Description, Create Manufacturer/Publication → Description, Create New Category → Details, Add Tag → Details, Create Author → Bio, Create Author → Quote, Create New Coupon → Bio, Create New Coupon → Description, Settings → Meta Description, Settings → OG Description [New Feature]
 - Added Video attribute for a product. [Enhancement]
 - Order cancel Recalculate [Enhancement]
 - Order Note attribute added for Order [Enhancement]
 - Guest Checkout Enable/Disable option on setting. [Enhancement]
 - Video Preview in Single Product view page. [Enhancement]

- Handled Unused Order Creation and other minor [Issue Fix]
- Fixed Settings issue [Issue Fix].
- Search Page redirection issue fixed.[Issue Fix].
- Next router URL issue fixed on search component[Issue Fix].
- Homepage control issue fixed[Issue Fix].

Breaking Change Log Version: 8.0.0

- Upgraded Pickbazar API to Laravel 10 & all other packages
- Upgraded Pickbazar Admin to Next JS 13 & all other packages
- Upgraded Pickbazar Shop to Next JS 13 & updated all other packages
- Fixed permission issue
- Fixed Slug issue

Breaking Change Log Version: 6.0.0

- Upgrade API to laravel 9
- Restructure the entire payment gateway system
- Added Stripe (Intent Based) and PayPal payment gateway
- Added card management feature
- Removed omniipay packages
- Added support for status-based inventory management option
- Added balance dispatch option based on status
- Added option to disable cash on delivery system
- Added product slug edit feature
- Fixed docker zip extension issue
- Fixed wallet points order issue
- Fixed order fail issue
- Fixed excel package cache issue
- Fixed user block and vendor inactive issue
- Fixed blank order created issue
- Fixed repeated new user wallet add issue for OTP login
- Fixed negative subtotal for coupon

Breaking Change Log Version: 5.0.0

- Multilingual support added
- Added toast notification after payment confirmation
- Fixed invoice issue for Unicode language
- Fixed wallet issue for social login
- Fixed file upload security issue
- Fixed book demo search page broken issue
- Fixed few minor bugs

Breaking Change Log Version: 2.0.0

- Multivendor support
- Single owner multiple shop
- Withdraws
- Order Management
- Shop Management
- Staff Management
- Product Management
- Attribute Management
- Shop Archives
- Shop details
- Internationalization support with RTL
- Product Tags support

Changelogs

Version: 11.0.0 : Major

3rd November 2023

Caution! Breaking Changes! see all breaking change list [All Breaking Changes](#)

- [Enhancement] & [Shop end] Two new demos.
- [Enhancement] & [Shop end] One new product grid for flash deals.
- [Enhancement] & [Shop end] Pickbazar shop design facelift
- [Enhancement] & [Shop end] Pickbazar shop Menu items updated

- [Enhancement] & [Shop end] Pickbazar shop Footer updated
- [Enhancement] Pickbazar admin dashboard design facelift
- [Enhancement] Pickbazar vendor & Staff dashboard design facelift
- [Enhancement] Settings page is divided into multi parts.
- [Enhancement] user section divided into multiple layer for better management.
- [Enhancement] Shop section divided into multiple layer for better management.
- [Enhancement] Products section divided into multiple layer for better management.
- [Enhancement] Settings cache system for performance optimization.
- [Enhancement] Documentation updated
- [Enhancement] Database seeder added
 - Settings seeder with Console command
 - Demo FAQs seeder
 - Demo refund reason & policy seeder
- [New Feature] Dashboard analytics section updated along with new analytics widgets
- [New Feature] Product Preview system added.
- [New Feature] Flash sale builder added
- [New Feature] Terms & Condition builder added
 - Super admin can perform CRUD operation
 - Vendor also can perform CRUD operation if and only if super admin enable this feature from settings.
- [New Feature] FAQs builder added
 - Super admin can add FAQs for global scope
 - Vendor can add FAQs based on his each shop
- [New Feature] Refund policy builder added
- [New Feature] Refund reason builder added.
- [New Feature] Product inventory route added
- [New Feature] Order transaction route added.
- [New Feature] Real-time notification system using Pusher
 - Order notification
 - Message notification
 - Store notice notification
- [New Feature] New filter option added in different routes
- [New Feature] New Dashboard topbar
 - With route searching
 - Shop redirection & creating
 - Order, Store notice & messaging dropdown bar
 - New Profile navigation

- [Issue fix] Variable products issue fix on digital item
- [Issue fix] Route permission issue fix for vendor & staff
- [Issue fix] Analytics issue fix on Staff user
- [Issue fix] Drafted product republish issue fix
- [Issue fix] Vendor page profile information issue fix
- [Issue fix] Currency cache issue fix
- [Issue fix] Product carting count issue based on available inventory
- [Issue fix] Messege converstation page CSS issue fix
- [Issue fix] Vendor root page shop grid issue fix
- [Issue fix] variable product inventory increment/decerement issue fix based on ordering.
- [Issue fix] Application fresh installation issue fix.
- [Issue fix] OpenAI chatbot suggestion issue fix.
- [Issue fix] Order status issue fix during order from different vendor
- [Issue fix] Customer order details payment and order status fix in case of refund
- [Issue fix] Toast issue fix throughout the dashboard & shop project
- [Issue fix] User order note issue fix
- [Issue fix] Site logo issue fix
- [Issue fix] Minimum order amount issue fix
- [Issue fix] Multilingual issue fix for some text strings
- [Issue fix] CSS issue fix throughout the shop end

Version: 10.6.0

28 August 2023

- [Enhancement] Added currency in the exported Order Excel sheet
- [Enhancement] Modified Order details in both shop and admin panel
- [Issue Fix] Partial wallet Payment
- [Issue Fix] Customer product review
- [Issue Fix] Send notification when order status changes

- [Issue Fix] Sorting and filtering issue fix for both Rest and Graphql

Version: 10.5.0

17 August 2023

- [New Feature] Shop Address autocomplete using Google map
- [Enhancements] External Product feature enhancements
- [Enhancements] International slug support added
- [Issue Fix] Import Export issue fixed
- [Issue Fix] Refund issue fixed
- [Issue Fix] Coupon date picker bug fixed

Version: 10.4.0

24 July 2023

- [New Feature] Google Maps find or Search Nearby shops
- [Issue Fix] Development mode build issue fixed
- [Issue Fix] Graphql playground issue fixed
- [Issue Fix] International Slug Generate Issue Fixed

Version: 10.3.0

12 July 2023

- [New Feature] Stripe Element support added
- [New Feature] Making sku unique
- [Issue Fix] Admin place order GraphQL
- [Issue Fix] FullWallet payment
- [Issue Fix] Change Payment Gateway as Guest user
- [Issue Fix] Admin File validation
- [Issue Fix] wishlist Issue Graphql
- [Issue Fix] Full Order Details based on User Role
- [Issue Fix] Store notice

Version: 10.2.0

13 June 2023

- [New Feature] Added Flutterwave Payment Gateway
- [New Feature] Added Flutterwave Configure Documentation.
- [Enhancements] User Documentation added(Question,Review,Withdrawals,Coupon,Tax & Shipping)

Version: 10.1.0

30 May 2023

- [New Feature] Added Paymongo Payment Gateway
- [New Feature] Added bKash Payment Gateway

Version: 10.0.0 Major

15 May 2023

Caution! Breaking Changes! see all breaking change list [All Breaking Changes](#)

- [New Feature] Set SMS Events from settings for Admin, Vendor & Customer.
- [New Feature] Set Email Event from settings for Admin, Vendor & Customer.
- [New Feature] Email Event Setting option
- [New Feature] SMS Settings for both Graphql and REST
- [New Feature] Added Setting Seeder
- [New Feature] New Console Command `marvel:settings_seed`
- [New Feature] Added Settings Dummy data seeder command
- [New Feature] Added OpenAI support in every possible description,
 - Create Shop → Description,
 - Create Manufacturer/Publication → Description,
 - Create New Category → Details,
 - Create Tag → Details,
 - Create Author → Bio,

- Create Author → Quote,
- Create New Coupon → Bio,
- Create New Coupon → Description,
- Settings → Meta Description,
- Settings → OG Description
- [Enhancement] Added video attribute for a product.
- [Enhancement] Order cancel Recalculate
- [Enhancement] Order Note attribute added for Order
- [Enhancement] Guest Checkout Enable/Disable option on setting
- [Enhancement] Video Preview in Single Product view page
- [Issue Fix] Handled Unused Order Creation and other minor
- [Issue Fix] Fixed Settings issue.
- [Issue Fix] Search Page redirection issue fixed.
- [Issue Fix] Next router URL issue fixed on search component
- [Issue Fix] Homepage control issue fixed

Version: 9.1.0

15 May 2023

- [New Feature] OpenAI integration added.
- [New Feature] Generate product description with AI.
- [New Feature] Generate store notice with AI.
- [New Feature] Predefined prompt suggestions with appropriate tone for both product description and store notice.
- [New Feature] Added Country wise currency formation support

Version: 9.0.0 Major

2 May 2023

- [New feature] Multiple payment gateway management
- [New feature] Product publish management control [new feature]
- [New feature] Best selling product [new feature]
- [Enhancement] New color scheme for order status
- [Enhancement] Payment modal steps modified
- [issue fix] Product sale price
- [issue fix] Multi-language auto translation

- [issue fix] Related product
- [issue fix] product out-of-stock quantity
- [issue fix] Backend order status GraphQL
- [issue fix] Unpublished/hidden product add-to-cart

Version: 8.2.0

3 April 2023

- [New feature] Added Xendit (Southeast Asia) payment gateway
- [New feature] Added Iyzico (Turkish) Payment Gateway
- [New feature] Added Email verification for all user and
- [New feature] Added Must verify email switch in the setting
- [Enhancement] Re-organized Exception handling for proper Rest status code

Version: 8.1.0

21 March 2023

- [New feature] Added Paystack Payment Gateway
- [New feature] Added SSLCOMMERZ Payment Gateway
- [issue fix] Fixed Default Card Issue

Version: 8.0.0 Major

15 March 2023

Caution! Breaking Changes! see all breaking change list [All Breaking Changes](#)

- [System Upgrade] Upgraded Pickbazar API to Laravel 10 & all other packages
- [System Upgrade] Upgraded Pickbazar Admin to Next JS 13 & all other packages
- [System Upgrade] Upgraded Pickbazar Shop to Next JS 13 & updated all other packages
- [issue fix] Fixed permission issue
- [issue fix] Fixed slug issue

Version: 7.0.0 Major

February 2023

- [New feature] Add Admin dashboard store notice CRUD for specific vendor/ All
- [New feature] Add vendor profile store notice list with read unread feature
- [New feature] Add Staff profile store notice list with read unread feature
- [New feature] Add Vendor Shop dashboard store notice CRUD
- [New feature] Add Store notice in shop for customer.
- [New feature] Add google map autocomplete address form
- [New feature] Add useGoogleMap switch in settings
- [New feature] Add Dashboard Menu message for admin , store owner and staff

Version: 6.3.1

9 January 2023

- [issue fix] Minimum order amount calculation issue fix shop & admin
- [Enhancement] Razorpay Documentation added.

Version: 6.3.0

22 December 2022

- [New feature] Added Mollie payment gateway
- [New feature] Added percentage, free-shipping coupon option
- [New feature] Added automatic free-shipping coupon option
- [New feature] Added option to set minimum amount for coupon
- [Enhancement] Display customer name for guest checkout

Version: 6.2.0

12 December 2022

- [New feature] Added RazorPay (India) payment gateway
- [New feature] Added name input options for guest checkout
- [issue fix] Fixed single-order issue
- [issue fix] Fixed maximum question limit issue

Version: 6.1.0

8 December 2022

- [New feature] Added RazorPay (India) payment gateway
- [New feature] Added name input options for guest checkout
- [issue fix] Fixed single-order issue
- [issue fix] Fixed maximum question limit issue

Version: 6.0.0 Major

16 November 2022

Caution! Breaking Changes! see all breaking change list [All Breaking Changes](#)

- [System Upgrade] Upgrade API to laravel 9

- [Enhancement] Restructure the entire payment gateway system
- [New feature] Added Stripe (Intent Based) and PayPal payment gateway
- [New feature] Added card management feature
- [Enhancement] Removed omnipay packages
- [New feature] Added support for status-based inventory management option
- [New feature] Added balance dispatch option based on status
- [New feature] Added option to disable cash on delivery system
- [New feature] Added product slug edit feature
- [issue fix] Fixed docker zip extension issue
- [issue fix] Fixed wallet points order issue
- [issue fix] Fixed order fail issue
- [issue fix] Fixed excel package cache issue
- [issue fix] Fixed user block and vendor inactive issue
- [issue fix] Fixed blank order created issue
- [issue fix] Fixed repeated new user wallet add issue for OTP login
- [issue fix] Fixed negative subtotal for coupon

Version: 5.0.1

16 August 2022

- [issue fix] Fixed dashboard recent withdraw issue
- [issue fix] Fixed vendor item edit path issue
- [issue fix] Fixed translation text issue
- [issue fix] Fixed add new address issue at backend booking
- [issue fix] Fixed admin profile image update issue
- [issue fix] Fixed composer v2 package installation issue
- [issue fix] Fixed category filter issue
- [issue fix] Fixed backend order address responsive issue

Version: 5.0.0 Major

7th July 2022

Caution! Breaking Changes! see all breaking change list [All Breaking Changes](#)

- [New feature] Multilingual support added
- [New feature] Added toast notification after payment confirmation
- [issue fix] Fixed invoice issue for Unicode language
- [issue fix] Fixed wallet issue for social login
- [issue fix] Fixed file upload security issue
- [issue fix] Fixed book demo search page broken issue
- [issue fix] Fixed few minor bugs

Version: 4.6.0

18th April 2022

- [New feature] Introduced order export support
- [New feature] Introduced my question page
- [New feature] Introduced my report page
- [New feature] Added variation support for review
- [New feature] Added question rate limit option
- [New feature] Added email notification for questions
- [New feature] Added support for auto remove review after refund
- [New feature] Added support for fixed report limit
- [issue fix] Fixed review bugs
- [issue fix] Fixed zip upload placeholder issue
- [issue fix] Fixed logout issue for mobile device
- [issue fix] Fixed export-import issue

Version: 4.5.1

1st April 2022

- [issue fix] Fixed category create issue
- [issue fix] Fixed OTP form issue
- [issue fix] Fixed feedback issue
- [issue fix] Fixed route issue

Version: 4.5.0

1st March 2022

- [New feature] Introduced Review support for products
- [New feature] Introduced Report abusive review
- [New feature] Introduced Review Feedback
- [New feature] Introduced Question and Answer support for products
- [New feature] Introduced Q&A Feedback
- [New feature] Introduced wishlist support
- [New feature] Introduced My wishlist page
- [issue fix] Fixed withdraw form balance issue
- [issue fix] Order Page selection issue fixed
- [issue fix] Packages version is locked to a fixed version
- [issue fix] Safari and Firefox scroll issue fixed
- [Enhancement] Out of stock notice

Version: 4.4.2

17th February 2022

- [System Upgrade] Upgrade shop component to tailwind 3
- [System Upgrade] Upgrade shop packages
- [issue fix] Fixed a few minor bugs

Version: 4.4.1

17th January 2022

- [New feature] Guest checkout support provided
- [New feature] Enable/Disable OTP in checkout option in admin settings
- [Enhancement] Restructured order details page.

Version: 4.4.0

10th January 2022

- [New feature] Digital Product support for both simple and variable product
- [New feature] New Book Demo introduced for digital download
- [New feature] Digital checkout page introduced
- [New feature] New search page introduced
- [New feature] Email newsletter subscription
- [New feature] Digital file download option
- [New feature] Backend order support
- [New feature] Variation specific image support
- [New feature] External product support
- [New feature] Author and Manufacturer support for products
- [Enhancement] Author and Manufacturer Archive page
- [Enhancement] Author and Manufacturer details page
- [Issue Fix] Rest vendor permission issue fixed
- [Issue Fix] Super admin role user creation issue fixed

Version: 4.3.0

25th October 2021

- [Issue Fix] Fixed rest home page query issue
- [Issue Fix] Fixed admin product edit issue
- [Issue Fix] Fixed ios issue

Version: 4.2.0

18th October 2021

- [New feature] Wallet points
- [New feature] Refunds using wallet points
- [New feature] Signup wallet points

Version: 4.1.0

4th October 2021

- [New feature] Introduced new Layout(Daily Needs)
- [Enhancement] Moved sorting to table column in admin
- [Issue Fix] Added few missing translation

Version: 4.0.0 Major

26th August 2021

- [Enhancement] Folder structure changed
- [Enhancement] api namespaces and prefix changed
- [Enhancement] Changed console command prefix to "marvel"
- [Enhancement] Updated doc
- [Issue Fix] Delivery time selection bug fixed
- [Issue Fix] Rest order creation bug fixed

Version: 3.0.0 Major

22 August 2021

- [System Upgrade] Updated necessary packages to latest version
- [Enhancement] Restructured shop code
- [New feature] Added layout changing option in admin
- [New feature] Added group based layout system

- [New feature] Dynamic promotional slider
- [New feature] Type based promotional slider
- [New feature] Single checkout page
- [New feature] Added dynamic homepage selection settings
- [New feature] added contact details for admin in settings
- [Issue Fix] Fixed cart update check availability issue
- [Issue Fix] Fixed color picker issue in order status form
- [Issue Fix] Fixed staff delete issue
- [Issue Fix] Fixed Inactive shop orders page issue
- [Issue Fix] Order details rest variable product title issue fixed
- [Issue Fix] Missing translation updated
- [Issue Fix] Minor css fix

Version: 2.3.1

3rd August 2021

- [Issue Fix] mobile navigation issue in admin
- [Issue Fix] cart update checkout data
- [Issue Fix] analytics data issue

Version: 2.3.0

2nd August 2021

- [New feature] product import export
- [New feature] minimum order amount
- [Issue Fix] env issue foxed for API
- [Issue Fix] rest draft product issue
- [Issue Fix] group delete issue fixed

Version: 2.2.0

18th July 2021

- [New feature] OTP Login
- [New feature] OTP verification in profile contact
- [New feature] OTP verification in checkout customer contact
- [New feature] Added group banner adding an option in the admin dashboard
- [New feature] Dynamic delivery time added in settings.
- [Enhancement] Stripe latest API version support with token
- [Enhancement] Vendor email notification
- [Enhancement] Restructured checkout procedure
- [Enhancement] Minor design improvement

Version: 2.1.0

8th July 2021

- [New feature] Added Sorting for dashboard listing pages
- [New feature] Added Filtering for product and category in the admin dashboard
- [New feature] Added shop location and settings
- [New feature] Added google places autocomplete
- [Enhancement] Minor design improvement
- [Issue Fix] Fixed SSR issue in the shop
- [Issue Fix] Fixed shop single page not found issue

Version: 2.0.1

3rd July Friday 2021

- [Issue Fix] Fixed "Invalid Version: 2.0" issue

Version: 2.0.0 Major

1st July Wednesday 2021

Caution! Breaking Changes! see all breaking change list [All Breaking Changes](#)

- [New feature] Multivendor support
- [New feature] Single owner multiple shop
- [New feature] withdraws
- [New feature] Order Management
- [New feature] Shop Management
- [New feature] Staff Management
- [New feature] Product Management
- [New feature] Attribute Management
- [New feature] Shop Archives
- [New feature] Shop details
- [New feature] Internationalization support with RTL
- [New feature] Product Tags support

Version: 1.3.0

16th June Wednesday 2021

- [Issue Fix] Product adding fix for rest api
- [Issue Fix] cart item delete fix for rest api

Version: 1.2.0

9th June, 2021

- [New feature] Added social authentication [facebook & google]
- [Issue Fix] Fixed password mismatched issue
- [Issue Fix] Fixed GraphQL admin product display issue
- [Issue Fix] Fixed GraphQL env issue
- [Issue Fix] Fixed few minor bugs

Version: 1.1.0

10th May Monday 2021

- [New feature] Variable Product support added
- [New feature] invoice download option added
- [New feature] SEO settings added
- [New feature] New modal layout design
- [New feature] New product single page layout design

Version: 1.0.1

Thursday 22 April 2021

- [Issue Fix] Fixed rest admin child product select issue

Version: 1.0.0

Thursday 14 April 2021

- Initial release.