# ✅ App Documentation #

Please follow the steps mentioned on the left side bar for easy setup

Introduction

eMart is a one-stop solution for Multi Store Item delivery system developed using Flutter, Firebase, and Laravel Framework with an expressive and elegant syntax.
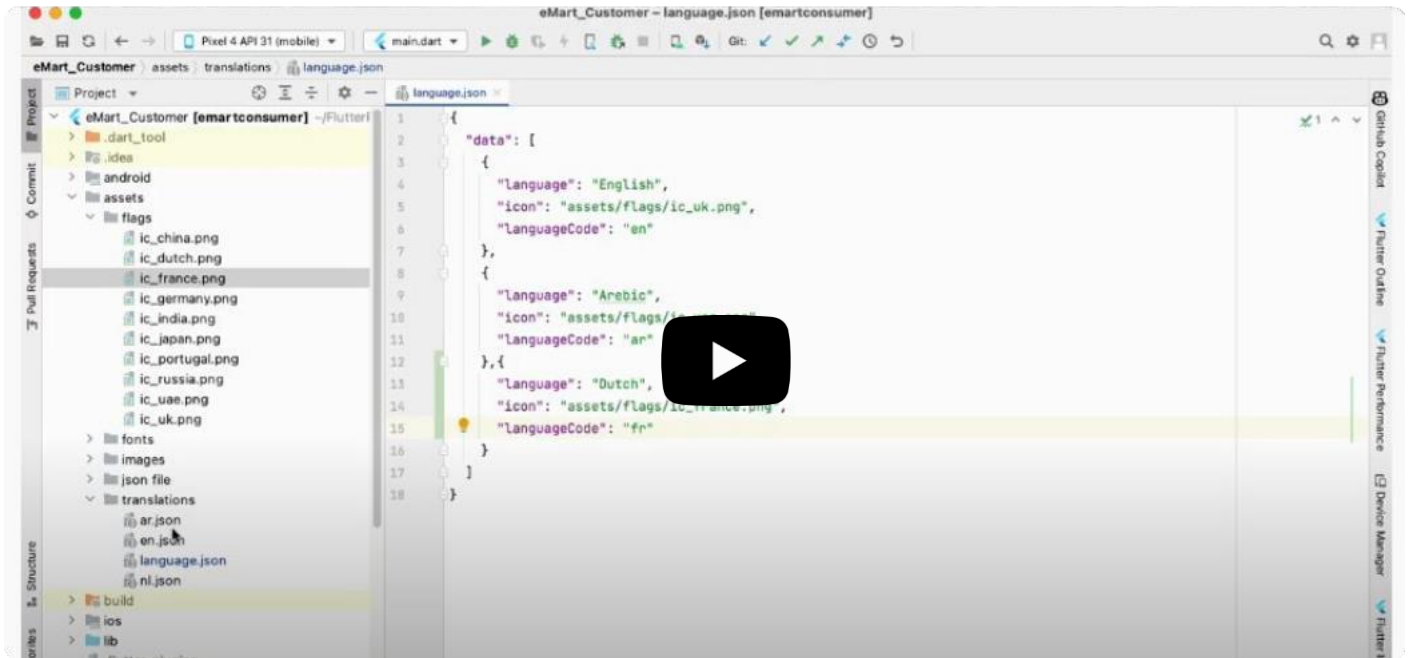
This product has three apps which are Customer app,Vendor app and Driver app. This documentation will be helpful in setting up eMart website and application.

How to Setup eMart App in Flutter (Customer App , Driver App , Store App)

(https://youtu.be/VSfgEPwWo7Y)
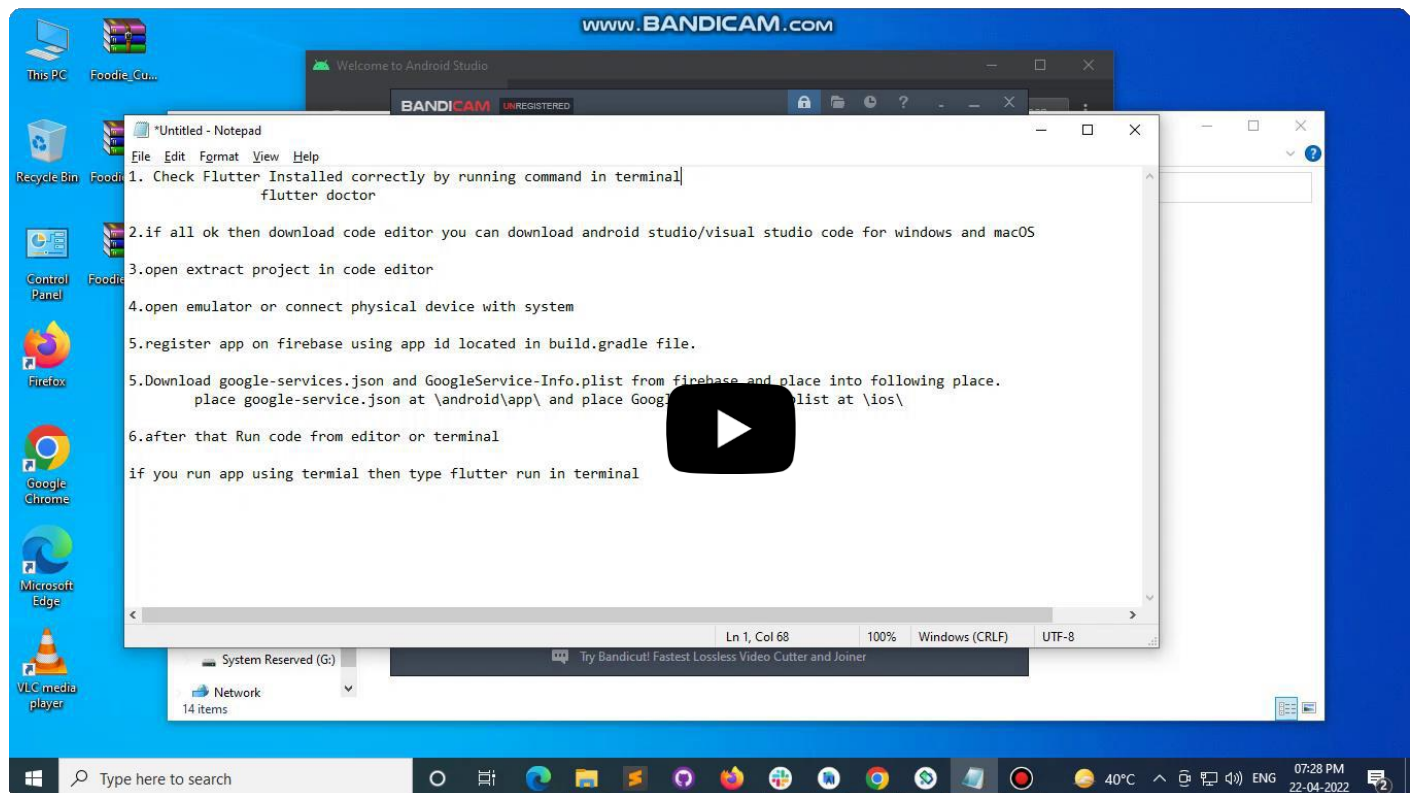
How to add new language file for the eMart APP ?

(https://youtu.be/vvjtVJXwDNM)

Link Firebase acc. & Push Notifications

Firebase Integration

a) Create a Firebase Account

Create a Firebase account (https:// rebase.google.com/) and then gain access to your own Firebase Console. A Firebase account can host multiple Firebase Projects that can host multiple mobile apps.

b) Configure a Firebase Project

Create a mobile app project in Firebase for each app that you are planning to publish. If you are publishing to both iOS and Android, you will need to create separate apps in Firebase, since technically they are different apps.

If you want your Flutter app to run on both iOS and Android, then you need to create a different mobile app for each (in Firebase). Simply click on the "Add App" button.

After you select the platform (iOS or Android), you need to provide the app's identi er.
Depending on which Flutter template you've purchased. Find the correct identi er in the les you've downloaded

for Android, go to android/app/build.gradle and get the applicationId

for iOS, open the app in Xcode and locate the Bundle ID field in the project's configuration

Once you have the bundle ID, just use it in Firebase, add your sha1 or sha256 key (https://developers.google.com/android/guides/client-auth) (For Mobile veri cation) and create the app.

> ℹ **Note:**
>
> Before you submit your app to the App Store or Google Play, you'll need to update these bundle identi ers with your unique identi ers, since the app stores don't allow duplicate application ids.

c) Enable Firebase Authentication

To enable Firebase Authentication, go to Firebase Console (https://console. rebase.google.com/?pli=1) -> Authentication -> Sign-in Methods and enable the methods that you are going to support in your app. By default, our Flutter apps have integration with Email/Password, Phone, and Facebook.
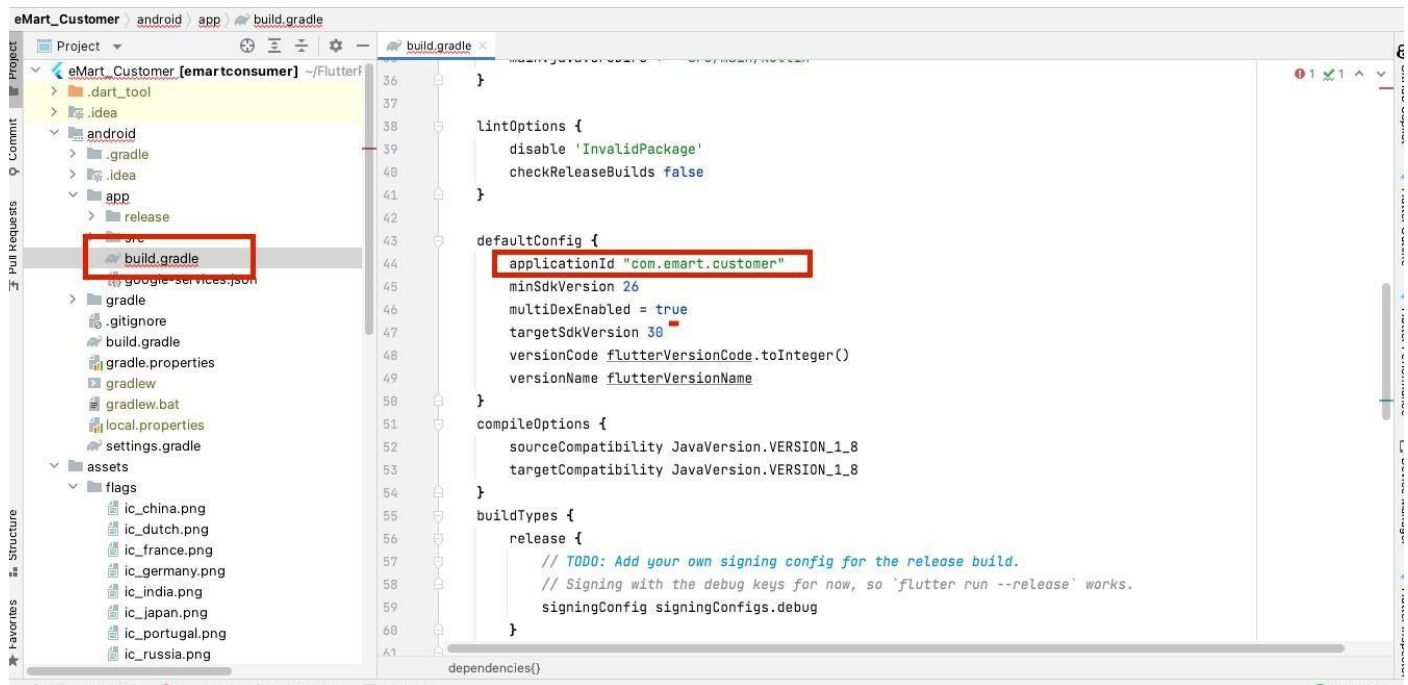
For Facebook log-in, you'll also need your Facebook App ID. You can add that later when you set up Facebook.

d) Enable Firebase Firestore

To allow the mobile app to read and write data to/from Firebase Firestore, set up the correct access permissions. To do that, just head over to Database -> Cloud Firestore and set the Rules for writes and reads to the public.

```
service cloud.firestore
{match/databases/}database},/documents {match/{document=**}, {allo
w read, write: if true;}, }, },
```

e) Enable Firebase Storage

If your mobile app needs access to Firebase Storage (e.g. for uploading photos and videos, for instance), you have to enable Firebase Storage, so that the functionality works properly. To enable it, just go to Storage in the left menu.

```
service cloud.firestore
{service firebase.storage {match /b/{bucket}/o {match /{allPaths=*
*} {allow read, write: if request.auth != null; } } }
```

f) Link Firebase Account to Your Mobile App

Once you've created the app, Firebase will generate a con guration le name(for android google-service.json and for ios GoogleService-Info.plist). You have to add this   le to your Flutter app. This is how the Flutter app can use your own Firebase backend. To do that, just download the con guration le and replace the existing mock les:

iOS: Download the GoogleService-Info-plist file and override the existing ios/NameOfApp/GoogleService-Info.plist file.
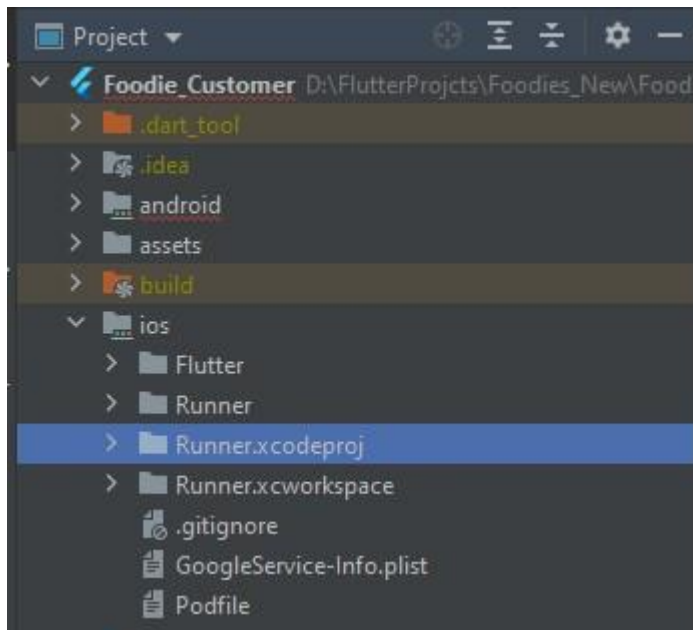
Android: Download the google-service.json file and replace the existing android/app/google-service.json file.
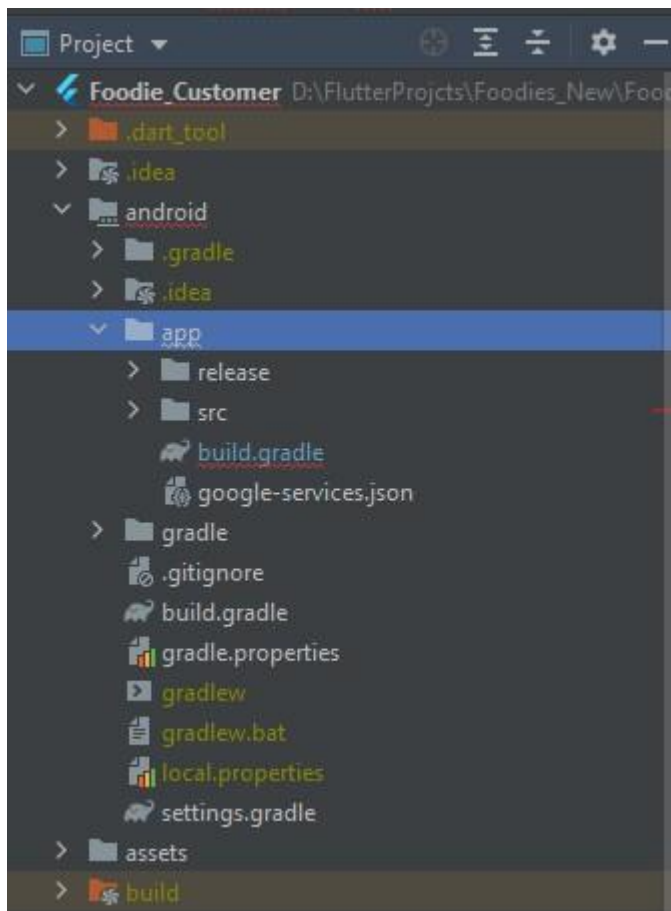

Add those les in all three apps as mentioned.

If you already have an app in Firebase, you can nd and download this con guration le in Firebase Console -> Project Settings.

Run your brand new Flutter template, the mobile app will use your own Firebase backend, as opposed to our default one. Make sure you add all the tables and the required data in your Firebase so that the app will have items to display (e.g. Item categories, chat messages, etc.). To quickly test the Flutter rebase integration, try registering a new user and see if they show up in Firebase -> Authentication tab.

g) Enable SMS Phone Authentication

To enable SMS authentication with Firebase, there are a few things we need to con gure, that will allow the app to send SMS to the users.

Enable Phone Authentication in Firebase In Firebase, go to Authentication -> Sign-in method -> Phone Authentication and check the Enable switch.
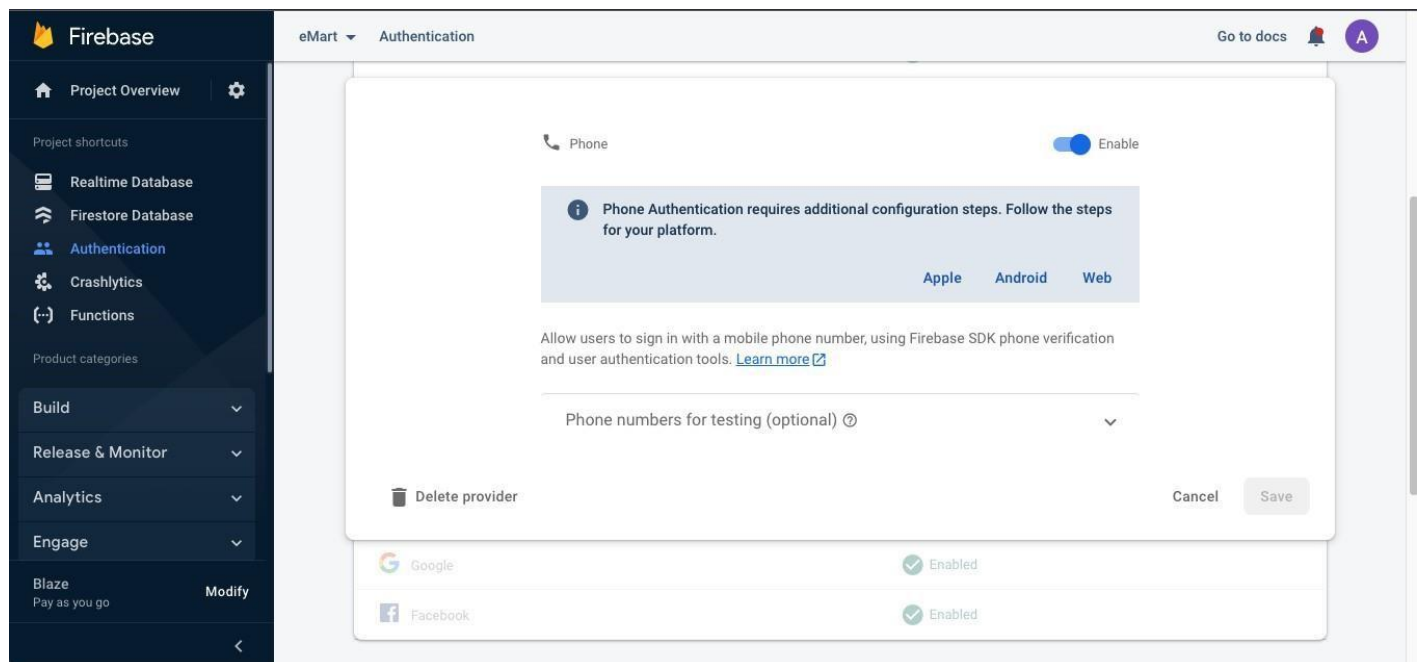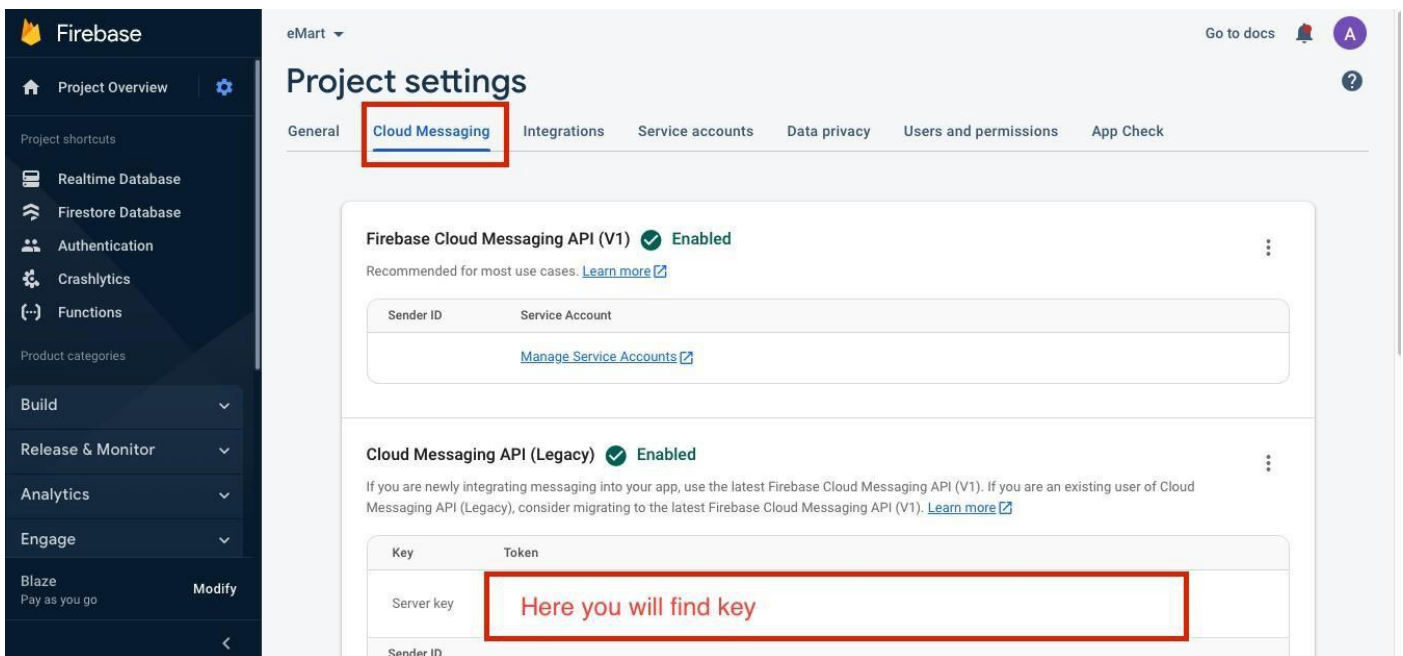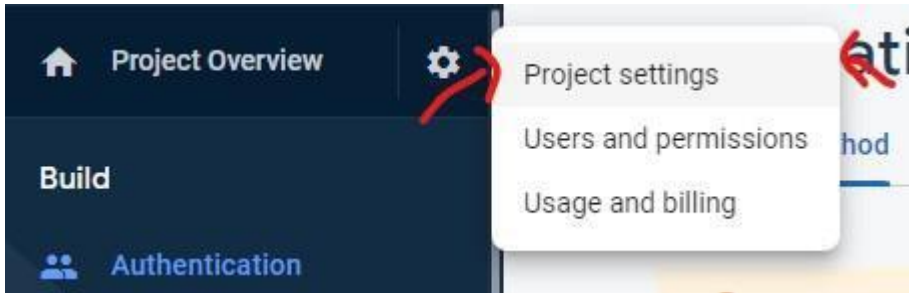

Push Notifications

This app uses Google Cloud Messaging (within Firebase) to send push noti cations to both iOS and Android devices.

Push noti cations are enabled by default in all of our Flutter templates. However, they are set up to work with our staging Firebase project, so you'll need to switch to your own project, similar to how you've done it for Firestore.

Setting Up Push Notifications with Your Own Firebase

To set them up with your own Firebase, all you need to do is to replace the Server Key of the lib/constants.dart le with your own.

a) In Firebase, go to Project Settings -> Cloud Messaging and copy the Server Key to clipboard

b) In the source code, go to lib/constants.dart and replace the Server Key (SERVER_KEY constant) with your own:

const SERVER_KEY = "YOUR_KEY_HERE";
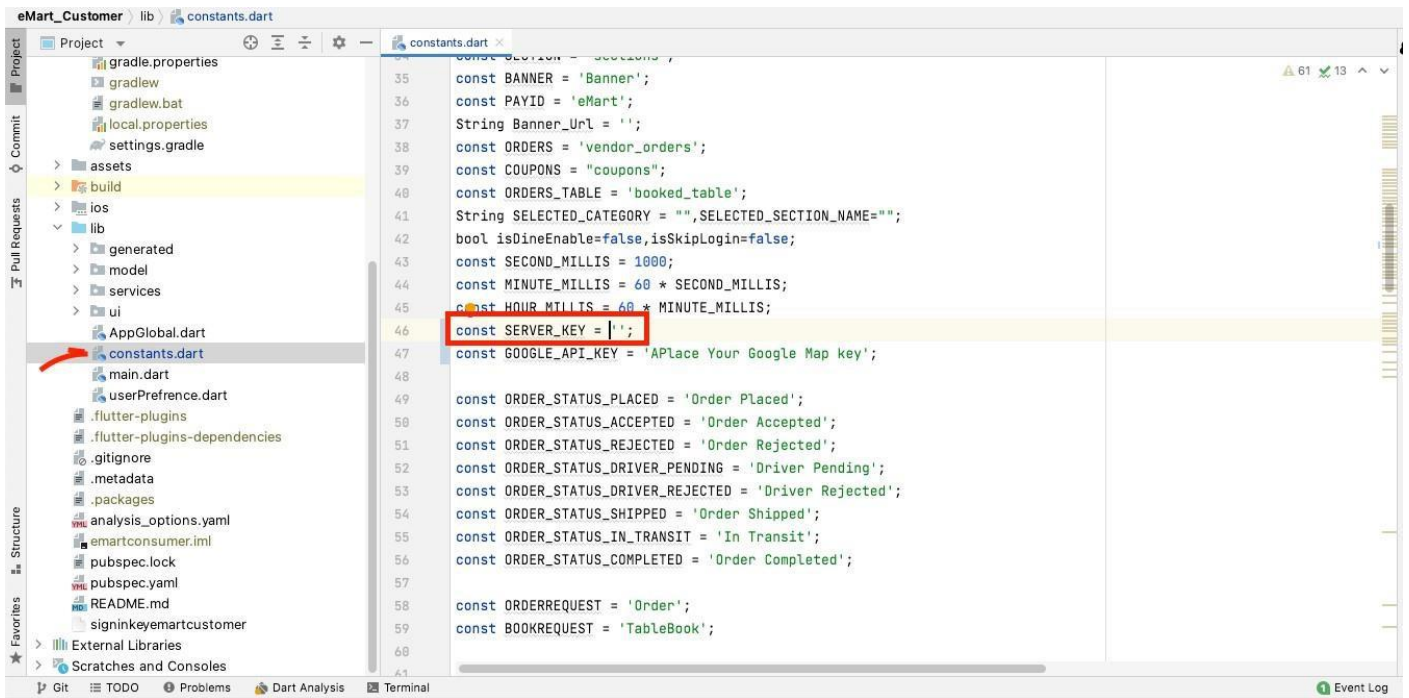
Code Documentation

sendNotification(String token, String title, String body) async {}

sendFcmMessage(String title, String message, String Token)

To identify all the places that trigger push noti cations, simply search for this method in the project.

Facebook Setup

To use facebook authentication register your app on Facebook Devloper Console Create an app (https://developers.facebook.com/) using app id .After that open facebook dev console and navigate to Settings/basic there you will get App ID and App secret copy both value to clipboard. Now open   rebase console and navigate to Authentication/ Sign-in method, click on edit icon and submit value in respective elds.
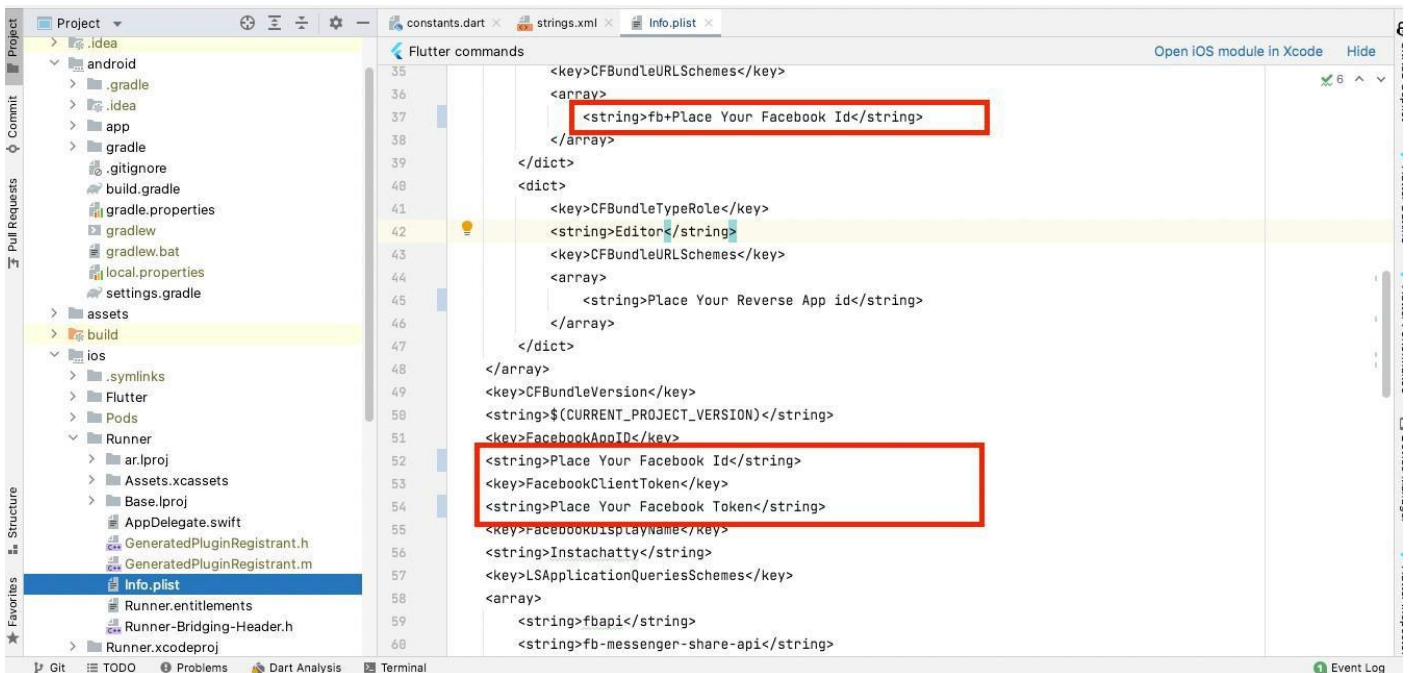
For Android:

- open   le :androidappsrcmainresvaluesstrings.xml
- Put app id at facebook_app_id and write "fb+app id" at fb_login_protocol_scheme.
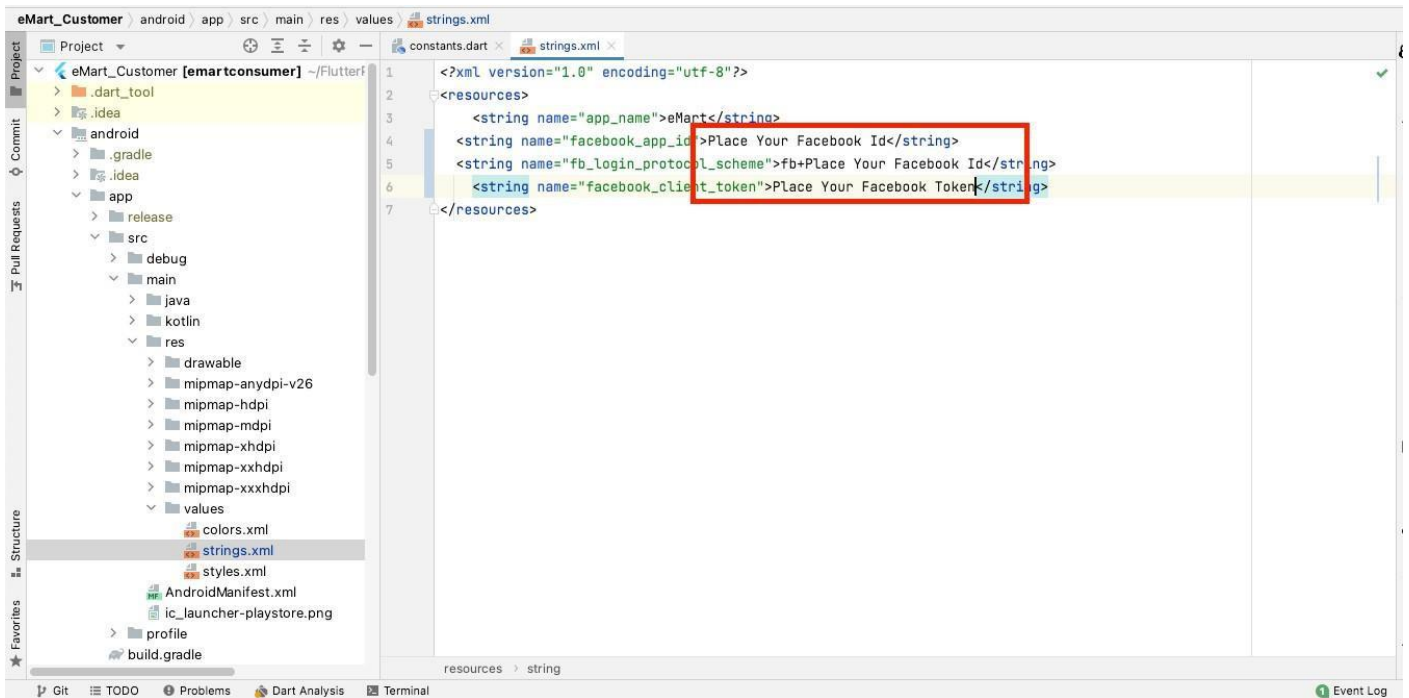
For iOS:

◇ Open file: eMart_Customer/ios/Runner/Info.plist

◇ Put app id at FacebookAppID and write "fb+app id" at CFBundleURLSchemes.

For Example your facebook id is 12345 then your fb_login_protocol_scheme and CFBundleURLSchemes will be fb12345.

Now facebook is set up.

Running on Android

Follow these simple steps to run Flutter templates on Android

Step 1: Plugin your Android device or open an emulator

Step 2: Open a Terminal window and run:

- 1. cd ~/path/to/template

- 2. ½utter run

Replace `~/path/to/template` with the correct path to the folder where you extracted the archive downloaded from our server. To make sure you are in the right folder, you can run "pwd" to see the current path. It must be the folder with the template, otherwise, the app won't run.

```
> flutter run with this one flutter run --no-sound-null-safety
```

Plug in an Android device or emulator

In order to run Flutter apps on Android, you need an Android device or an emulator. If you have an Android phone or tablet, simply plug it in. You might need to enable USB debugging in Device Settings, under Developer Tools. Follow the Android Installation Guide (https://developer.android.com/studio/run/device).

Android emulators are bundled into Android Studio, so please install Android Studio (https://developer.android.com/studio), open it, go to Tools -> AVD Manager, and start an emulator of your choosing: You can also create new emulators of your own, with your own hardware requirements. Once you have an emulator up and running, proceed to the next step.

Run the Flutter App

For MacOS / UNIX simply run the two commands we described above:

- 1. cd ~/path/to/template

- 2. ½utter run

If you are using Flutter >= 2.0 or Dart version >=2.12 with Null Safety you might want to replace this command

> utter run with this one   utter run –no-sound-null-safety

Visual Studio Code can be used that's directly located at the right folder. With it you can simply run "½utter run" and the app will just start.

> **ⓘ** Note:
>
> If you are using Flutter >= 2.0 or Dart version >=2.12 with Null Safety you might want to replace this command

> utter run with this one   utter run –no-sound-null-safety

Running on iOS

You would need Mac OS and Xcode to run a Flutter app on iOS, but there's no need for an Apple developer account or an iPhone — you can simply run an Xcode project on the iOS simulator.

If you are not new to iOS / Flutter development, and you already have the dev environment set up, simply do the following steps:

- Be sure you have the latest STABLE Xcode version (DO NOT USE XCODE BETA)
- Run "½utter run" in the root folder of the downloaded project:
    1. ½utter run

Note that if you are using Flutter >= 2.0 or Dart version >=2.12 with Null Safety you might want to run
1. ½utter run –no-sound-null-safety

- If the project did not start, open the .xcworkspace le in Xcode, which got generated at the previous step (you can nd it in the ios folder). Also, make sure a Pods folder got generated.
- In Xcode, select the iOS simulator/device and build and run the app (Command + R) If this is the rst time you are running an iOS project on your machine, you need to set up your environment   rst. We recommend you follow the Flutter Guide (https://½utter.dev/docs/get- started/test-drive?tab=vscode).
- There are a few prerequisites that are iOS speci c (you don't need them for Android):
    - Install Xcode
    - (https://developer.apple.com/xcode/) Install

        CocoaPods (https://cocoapods.org/)

    - Install Flutter (https://½utter.dev/)

Follow these steps to get your app running on an iOS device or iOS simulator.

1. Run " utter run" in the project root directory

if the "flutter run"command fails, you might want to run : > ½utter run –no-sound-
null- safety

2. Locate the ios folder and run "pod update" This will install all the pods that the Flutter
app depends on, such as Firebase Auth, Storage, etc. Make sure you have Cocoapods installed
beforehand.

3. Open the .xcworkspace   le in Xcode Simply double click on the le in the terminal (make
sure you are in the ios folder).

4. Run Xcode project
Choose a simulator or device in Xcode then go to Product -> Run, or simply press Command
+ R. This will build and run the chat project in the selected device/simulator.

Map Setup

To use map in our app we have to enable multiple apis from google console.Visit this link to
enable apis. Visit here
(https://developers.google.com/maps/documentation/javascript/cloud- setup).

After that enable below apis from list:

- Maps SDK for iOS

- Maps SDK for

- Android Directions

- API Geocoding API

that's it all required apis are now enabled and you can see map in your apps.

Run on iOS and Android

Setting up

In this module, we are going to outline how to con gure your Flutter development.

Install Basic Flutter Development Tools

View the Flutter Installation Guide to know about basic ½utter development
(https://½utter.dev/docs/get-started/install).

Android

Configure Flutter Development for Android

To set up Flutter for Android you need to install Dart, Flutter, Visual Studio Code (Optional), and
Android Studio and follow these simple steps.

- Install & Con gure with Flutter Installation Guide
  (https://½utter.dev/docs/get- started/install)
- Download & Install Visual Studio Code (https://code.visualstudio.com/) (recommended
  for developing Flutter apps on Windows)
- Download & Install Android Studio (https://developer.android.com/studio) (only if you
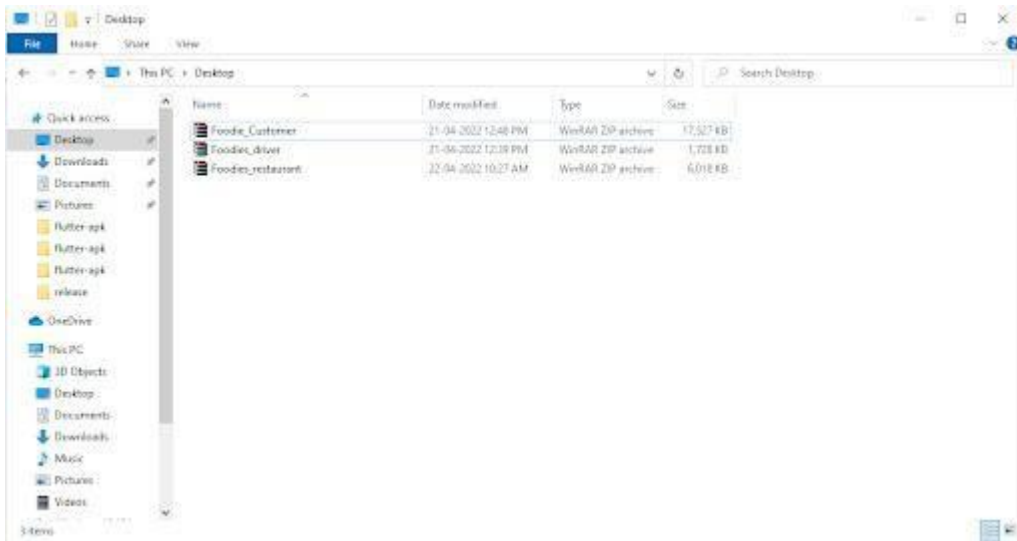 don't have an Android device)

iOS

Configure Flutter Development for iOS

Install Cocoapods, Xcode (Make sure to check the latest version of Xcode).

Setting up Project

Download zip les which contain apps and extract them to the destination place.Open project
in visual studio/android studio/ xcode.

Quick Fix

Quick Fix for Android and iOS

If sometimes you face errors while setting up project you can follow this path to debug and solve that yourself. you can see video for these.

- 1. Find if there is any issue in code most of IDE show red dot or line for that

- 2. Check if you missed and API or key from above setup

- 3. if noting found in above two steps then run ½utter clean it will clean your code

- 4. check if there is build folder deleted or not, if it still there then delete it.

- 5. Delete pubsec.lock le

- 6. if you run for android delete .gradle folder and for ios delete below all les and folder

    - 1.delete .syblinks folder

    - 2.delete pods folder

    - 3. delete pod le.lock

- 7. sometime for android need to update gradle. For that perform all above steps then perform pub get and open android folder as project in new window and from tool bar you can update gradle.You can see video for more details.

- 8. After nishing all above steps just run apps most of problem solved by following these steps.
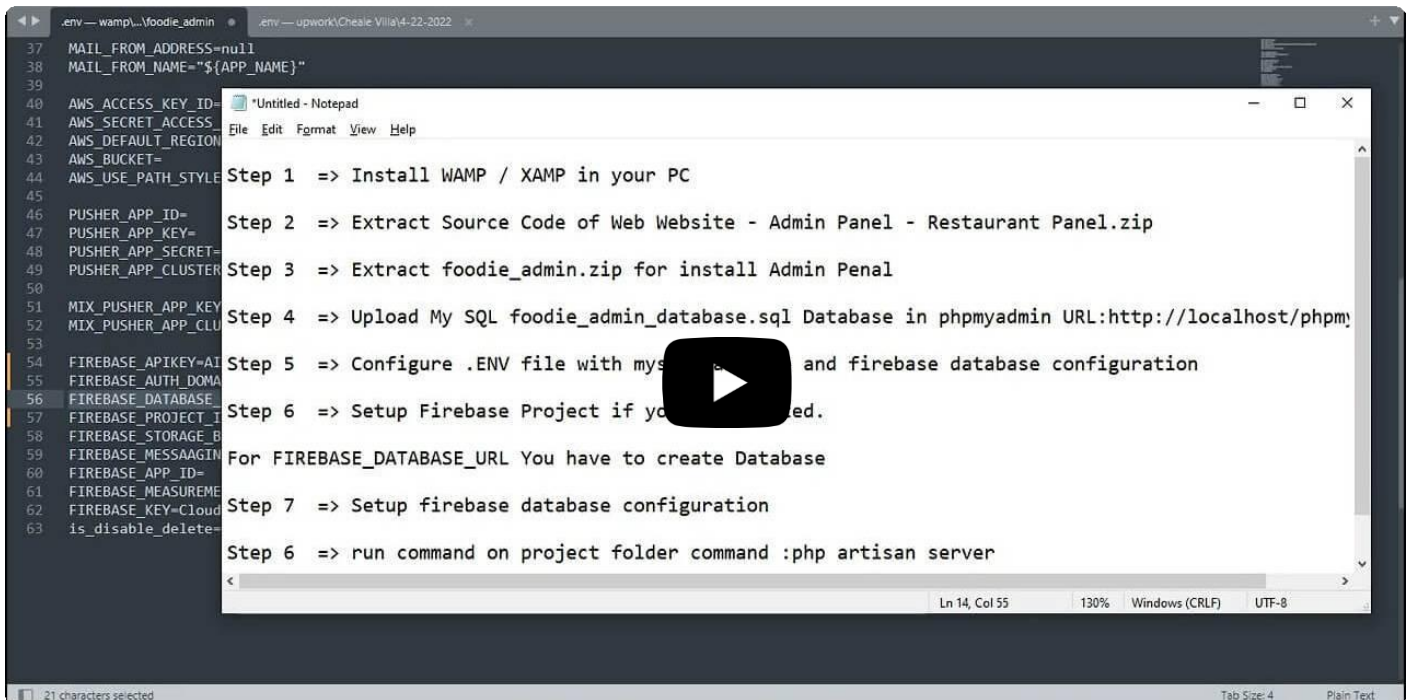
☑️ Web Documentation

Introduction

eMart is a one-stop solution for Multi Store Item delivery system developed using Flutter, Firebase, and Laravel Framework with an expressive and elegant syntax.

All the system requirements are satis ed by the Laravel Homestead virtual machine, so it's highly recommended to use Homestead as the local Laravel development environment.

Make sure your server meets the following requirements: (If you aren't using Homestead)

○ PHP Version (8.0 or above)

Php Laravel Admin Panel

Introduction

eMart is a one-stop solution for Multi Store Item delivery system developed using Flutter, Firebase, and Laravel Framework with an expressive and elegant syntax.

All the system requirements are satis ed by the Laravel Homestead virtual machine, so it's highly recommended to use Homestead as the local Laravel development environment.

Make sure your server meets the following requirements: (If you aren't using Homestead)

○ PHP Version (8.0 or above)

○ SSL

○ You must need google rebase account with upgraded plan.

  ○ Firebase Account (note: we need to upgrade rebase functions click here (https://laravelwith rebase.blogspot.com/2020/09/chapter-21-what-are- rebase-cloud- functions-and-install-deploy.html))

  ○ Google MAP API keys

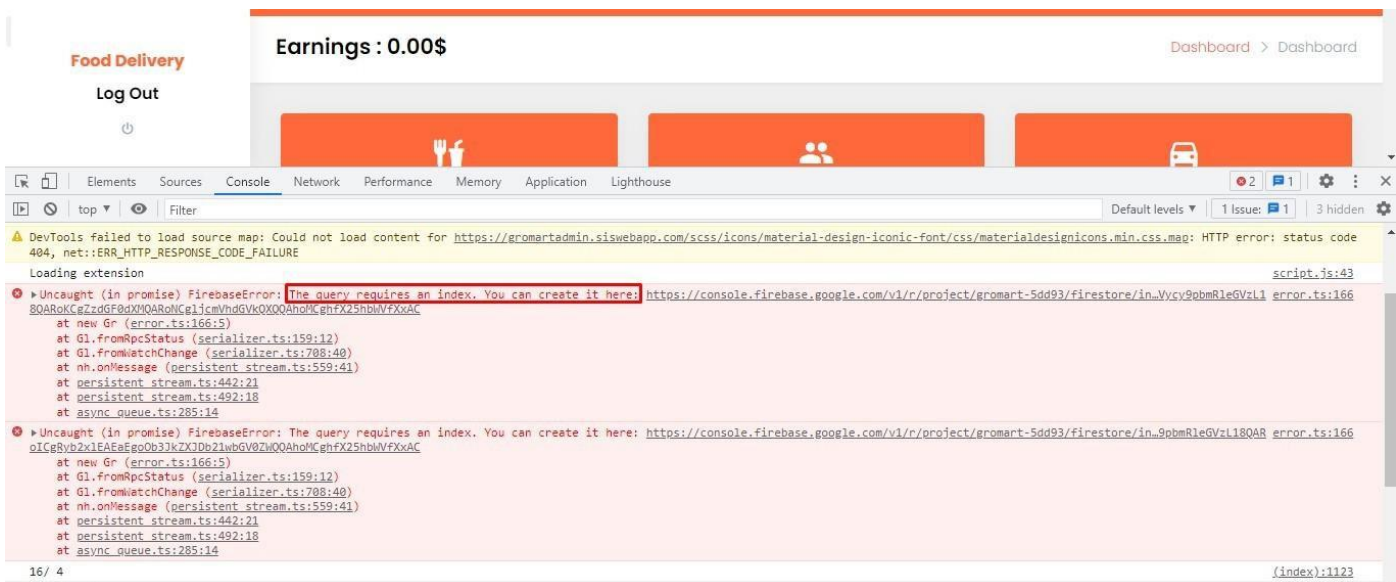## Setup eMart Admin Panel Laravel + Firebase Database

(https://www.youtube.com/watch?v=K6M3_XqjzZY)

Query Indexing

You can check indexing URL in Inspect Element.

Click On URL it will index automatically.

Upload on Server

After downloading the code, upload the Admin zip le in your server following your expected directory and extract the zip le. eMart admin panel can be installed on a domain or subdomain.

> ℹ️ **Note:**
>
> Don't install the admin panel or web app in a subdirectory. If you want the web app in your main domain then you need to install the admin panel in a subdomain.

Add folder to a ZIP

After you have signed in to your Bluehost account, nd the folder named /laravel_application zip it, and then use the le manager to upload this le inside your /public_html folder.
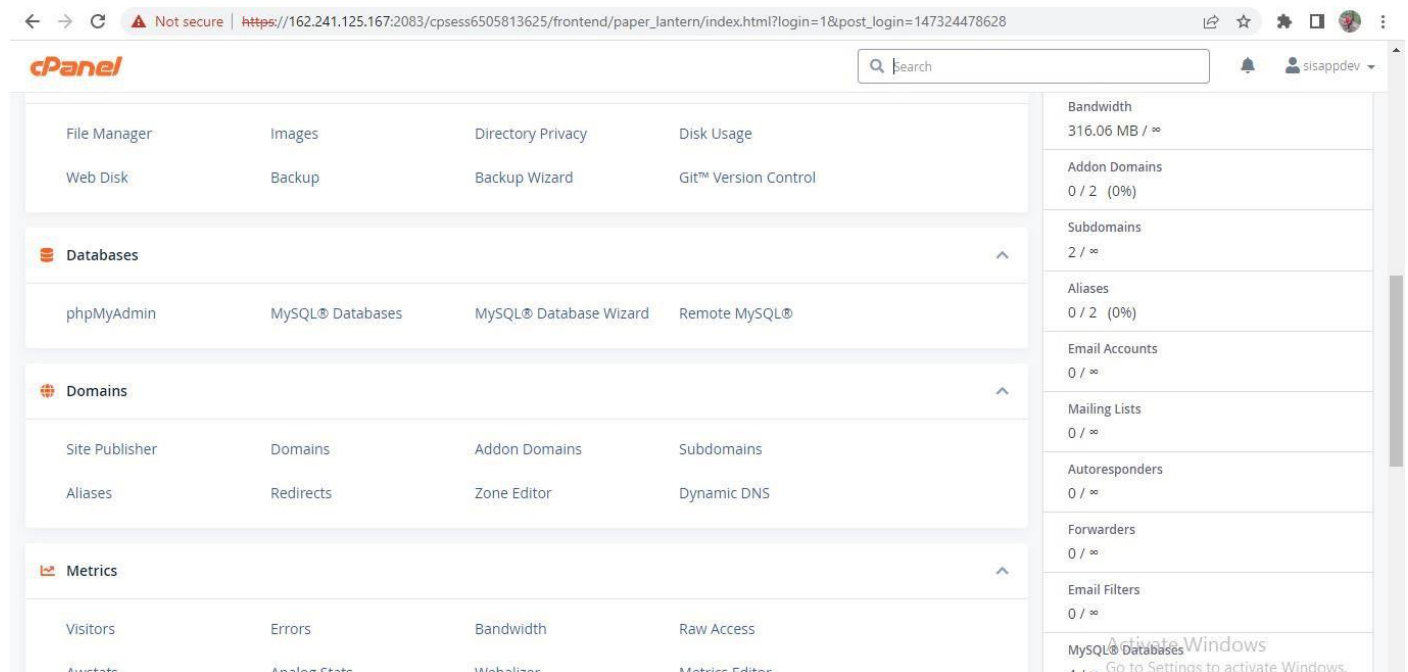
Extract the ZIP file

To get the les on your le manager, click right on your uploaded zip le and choose extract.
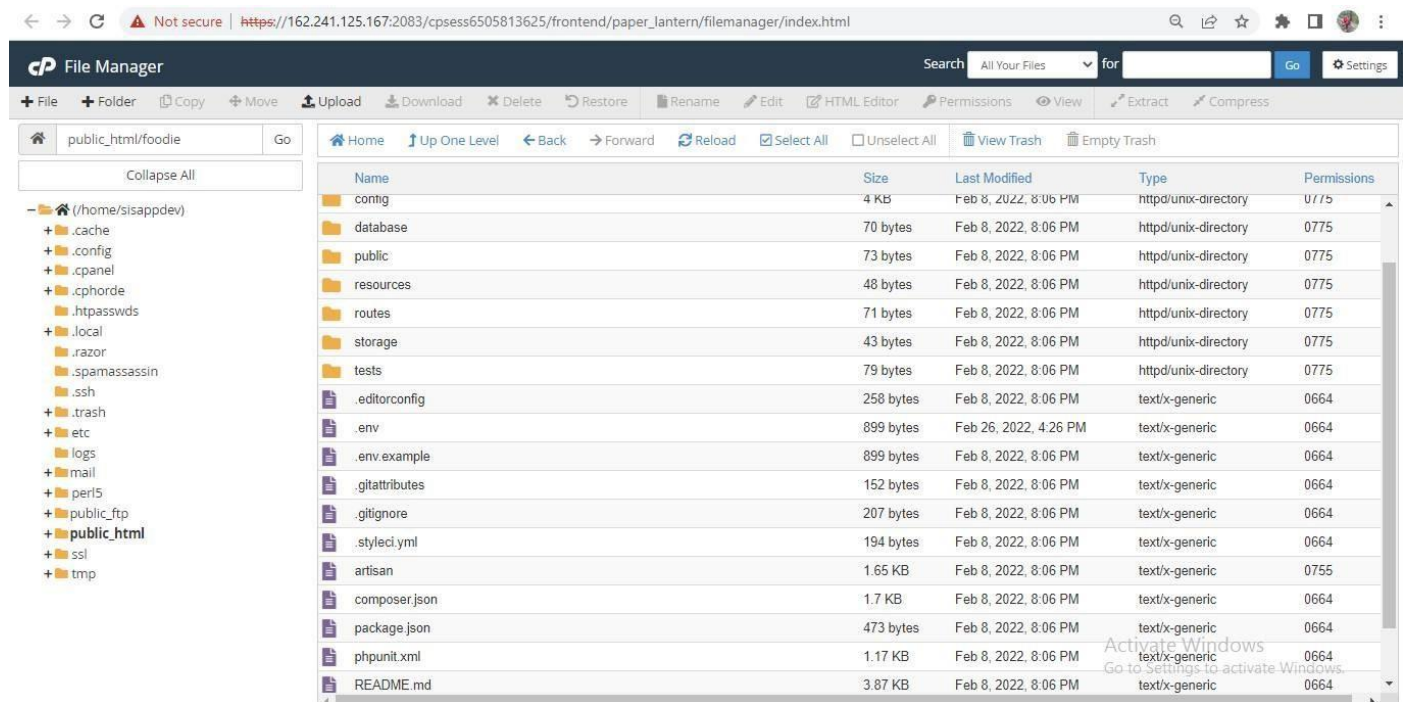
Admin Panel Installation

please watch video how to install admin penal in local server.   click here.

Step 1:

After downloading the code upload the admin install zip in your directory and extract the zip le. eMart can be installed on your main domain or subdomain.

eMart requires an SSL certi cate to be installed on your domain to work with all the services.

Step 2:

Requirements:

○ PHP Version (8.0 or above)

○ SSL

○ Firebase Account (note: we need to upgrade rebase functions click here (https://laravelwith rebase.blogspot.com/2020/09/chapter-21-what-are- rebase-cloud- functions-and-install-deploy.html))

○ Google MAP API keys

Step 3:

Now you need to set Database host, Database name, Database username, Database password, and click continue if the database is in your server then the host is localhost.

Step 4:

Click on the Import Database button to import the SQL le.
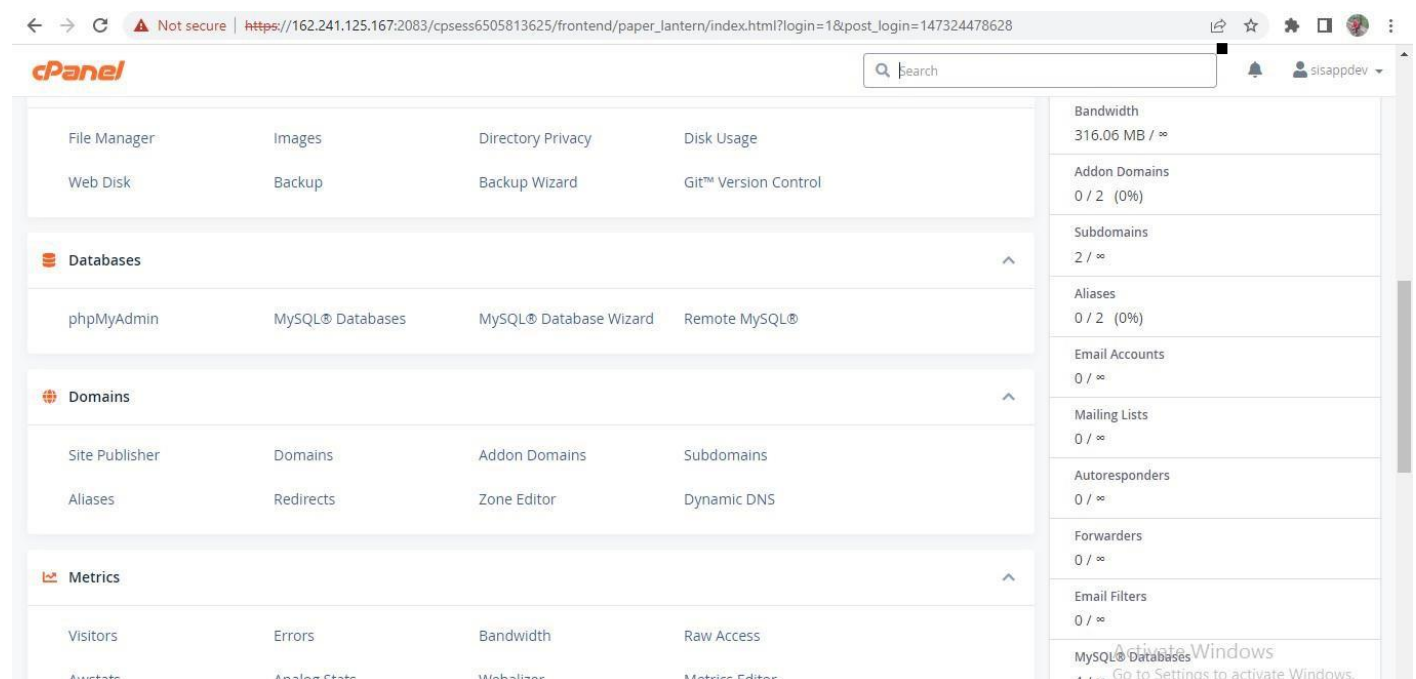
Step 5:

Click on continue, after   lling up the information.
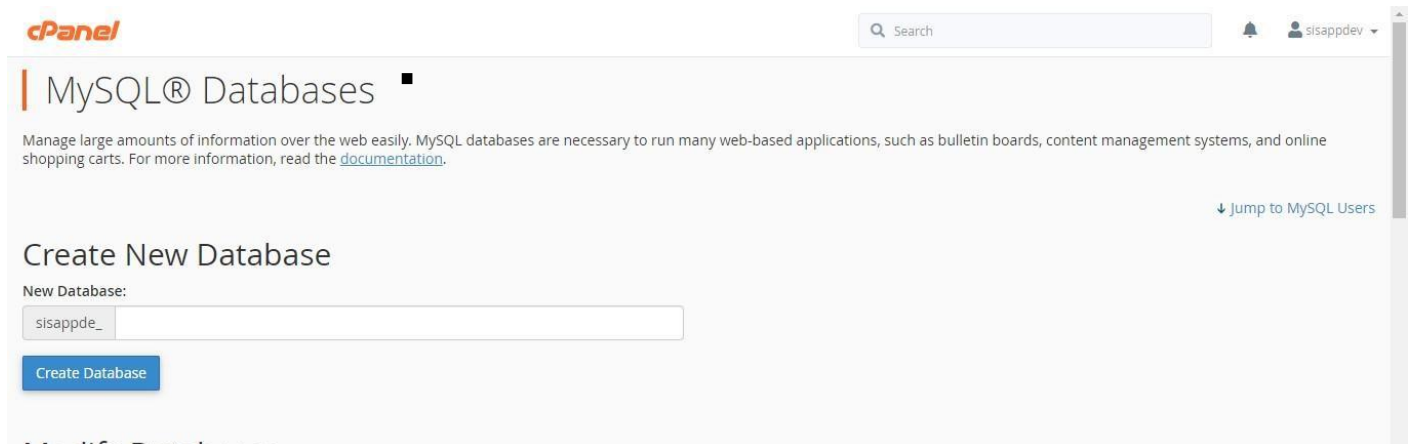
Step 6:

Software is ready to Run. Click on the admin panel or view the landing page.

Create Database

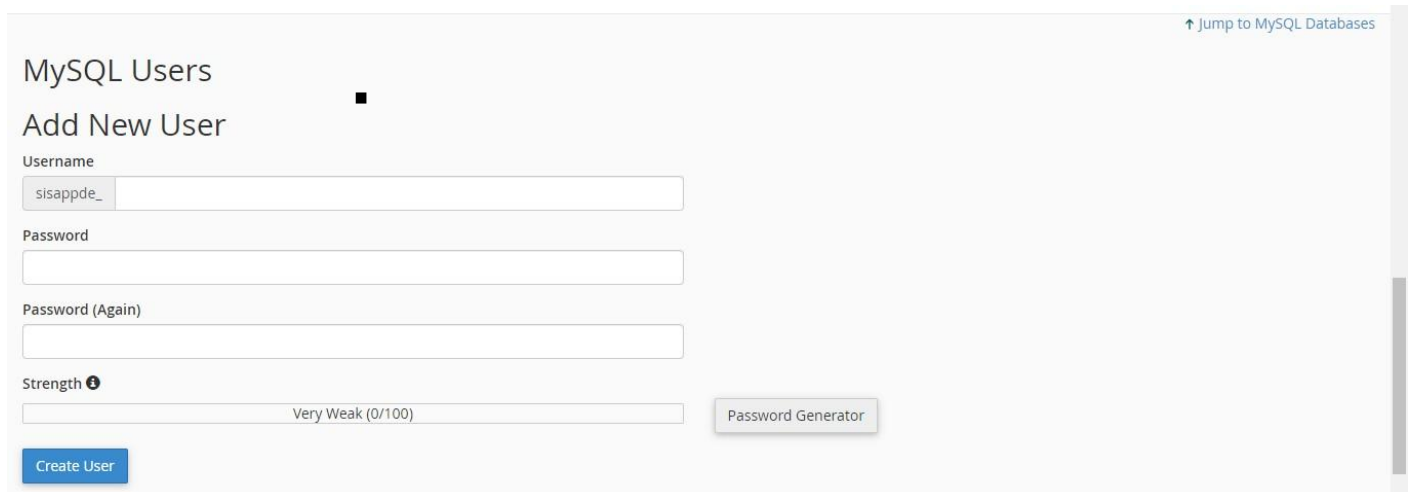**Step 1:** Create a new database from your server MYSQL database.
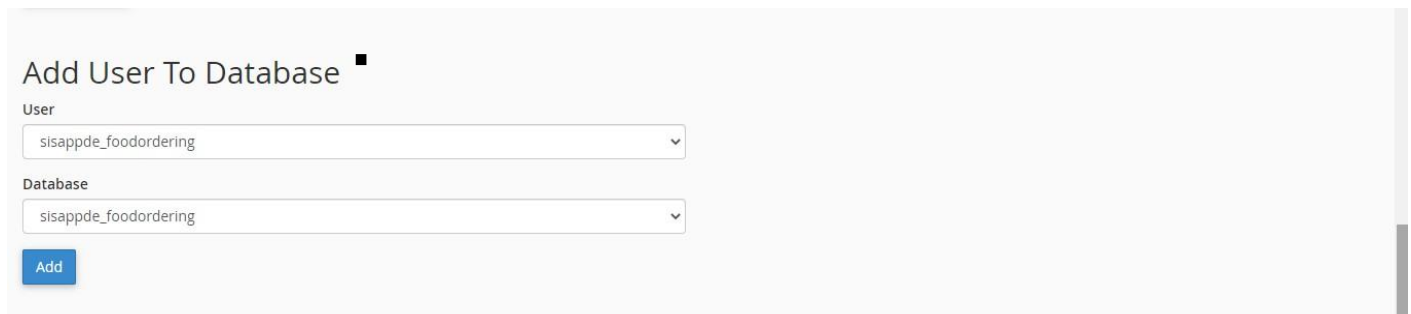
Step   2: Create New Database

Step   3: Create a new database user

Step 4: Link the database to the newly created DB user.

Step 5: Give all permission to your user by Check on "All PRIVILEGES" and clicking on "Make Changes"

Website

Introduction

eMart is a one-stop solution for Multi Store Item delivery system developed using Flutter, Firebase, and Laravel Framework with an expressive and elegant syntax.

All the system requirements are satis ed by the Laravel Homestead virtual machine, so it's highly recommended to use Homestead as the local Laravel development environment.

Make sure your server meets the following requirements: (If you aren't using Homestead)

- PHP Version (8.0 or above)

(https://www.youtube.com/watch?v=sex4t1U6iZQ)

Upload on Server

Upload the ZIP

After you sign in to your hosting go to your le manager and upload the downloaded zip le then extract it in your /public_html folder.

Extract the ZIP file

To get the les on your   le manager, click right on your uploaded zip   le and choose extract

Configure Environment File

Inside the root folder of your admin panel, you will nd a le named .env copy and past it into the root folder of the extracted zip le.

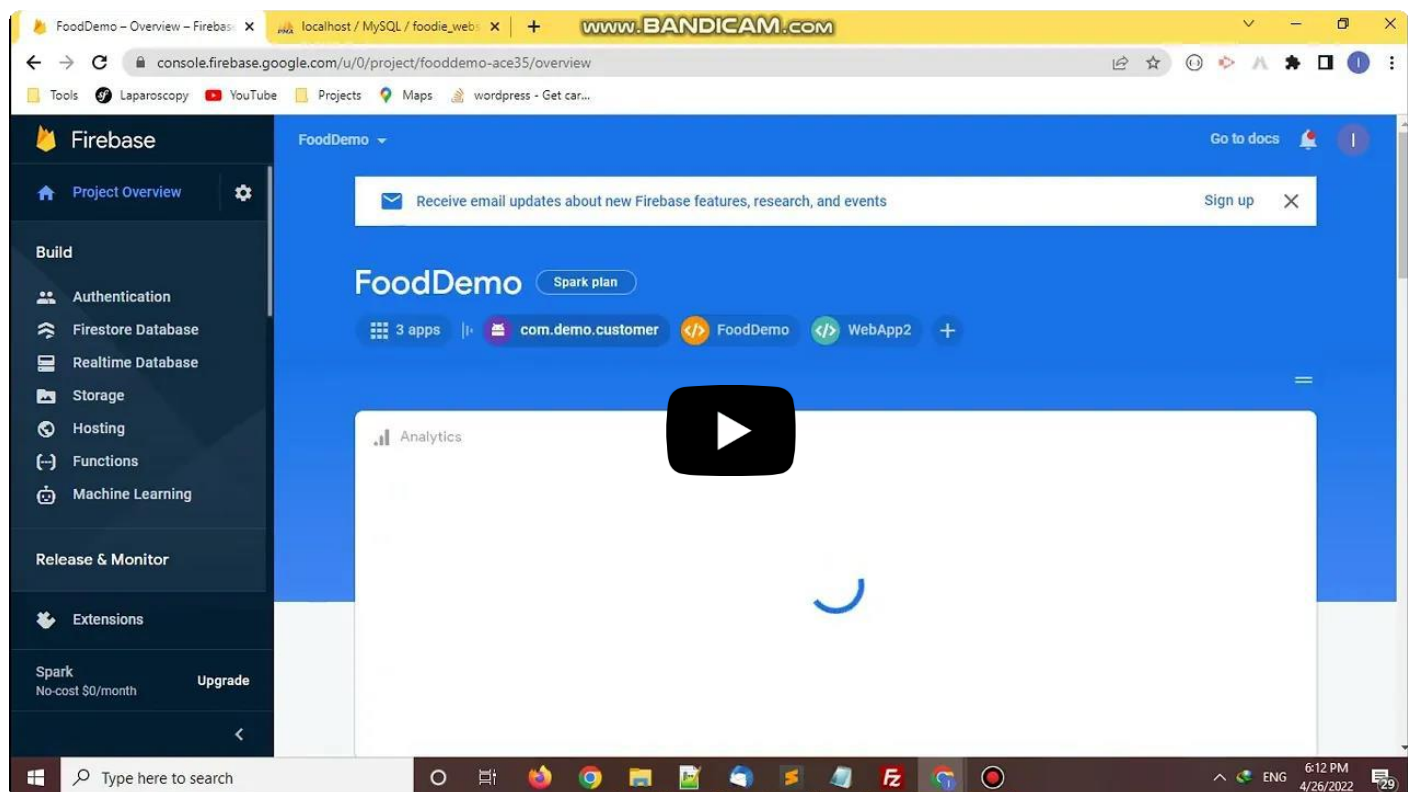| Name	Size	Last Modified	Type	Permissions
config	4 KB	Feb 8, 2022, 8:06 PM	httpd/unix-directory	0775
database	70 bytes	Feb 8, 2022, 8:06 PM	httpd/unix-directory	0775
public	73 bytes	Feb 8, 2022, 8:06 PM	httpd/unix-directory	0775
resources	48 bytes	Feb 8, 2022, 8:06 PM	httpd/unix-directory	0775
routes	71 bytes	Feb 8, 2022, 8:06 PM	httpd/unix-directory	0775
storage	43 bytes	Feb 8, 2022, 8:06 PM	httpd/unix-directory	0775
tests	79 bytes	Feb 8, 2022, 8:06 PM	httpd/unix-directory	0775
.editorconfig	258 bytes	Feb 8, 2022, 8:06 PM	text/x-generic	0664
.env	899 bytes	Feb 26, 2022, 4:26 PM	text/x-generic	0664
.env.example	899 bytes	Feb 8, 2022, 8:06 PM	text/x-generic	0664
.gitattributes	152 bytes	Feb 8, 2022, 8:06 PM	text/x-generic	0664
.gitignore	207 bytes	Feb 8, 2022, 8:06 PM	text/x-generic	0664
.styleci.yml	194 bytes	Feb 8, 2022, 8:06 PM	text/x-generic	0664
artisan	1.65 KB	Feb 8, 2022, 8:06 PM	text/x-generic	0755
composer.json	1.7 KB	Feb 8, 2022, 8:06 PM	text/x-generic	0664
package.json	473 bytes	Feb 8, 2022, 8:06 PM	text/x-generic	0664
phpunit.xml	1.17 KB	Feb 8, 2022, 8:06 PM	text/x-generic	0664
README.md	3.87 KB	Feb 8, 2022, 8:06 PM	text/x-generic	0664

Store Panel

Introduction

eMart is a one-stop solution for Multi Store Item delivery system developed using Flutter, Firebase, and Laravel Framework with an expressive and elegant syntax.

All the system requirements are satis ed by the Laravel Homestead virtual machine, so it's highly recommended to use Homestead as the local Laravel development environment.

Make sure your server meets the following requirements: (If you aren't using Homestead)

○ PHP Version (8.0 or above)

(https://www.youtube.com/watch?v=wJrtg6bwtxs)

Upload on Server

Upload the ZIP

After you sign in to your hosting go to your le manager and upload the downloaded zip le
then extract it in your /public_html folder.

Extract the ZIP file

To get the les on your le manager, click right on your uploaded zip le and choose extract.

Configure Environment File

Inside the root folder of your admin panel, you will nd a le named .env copy and past it
into the root folder of the extracted zip le.

Firebase

Create Firebase project

Step 1: Go to firebase console using this link Firebase (https:/ firebase.google.com/)

Step   2: Click on "Go to console" in the top right corner.
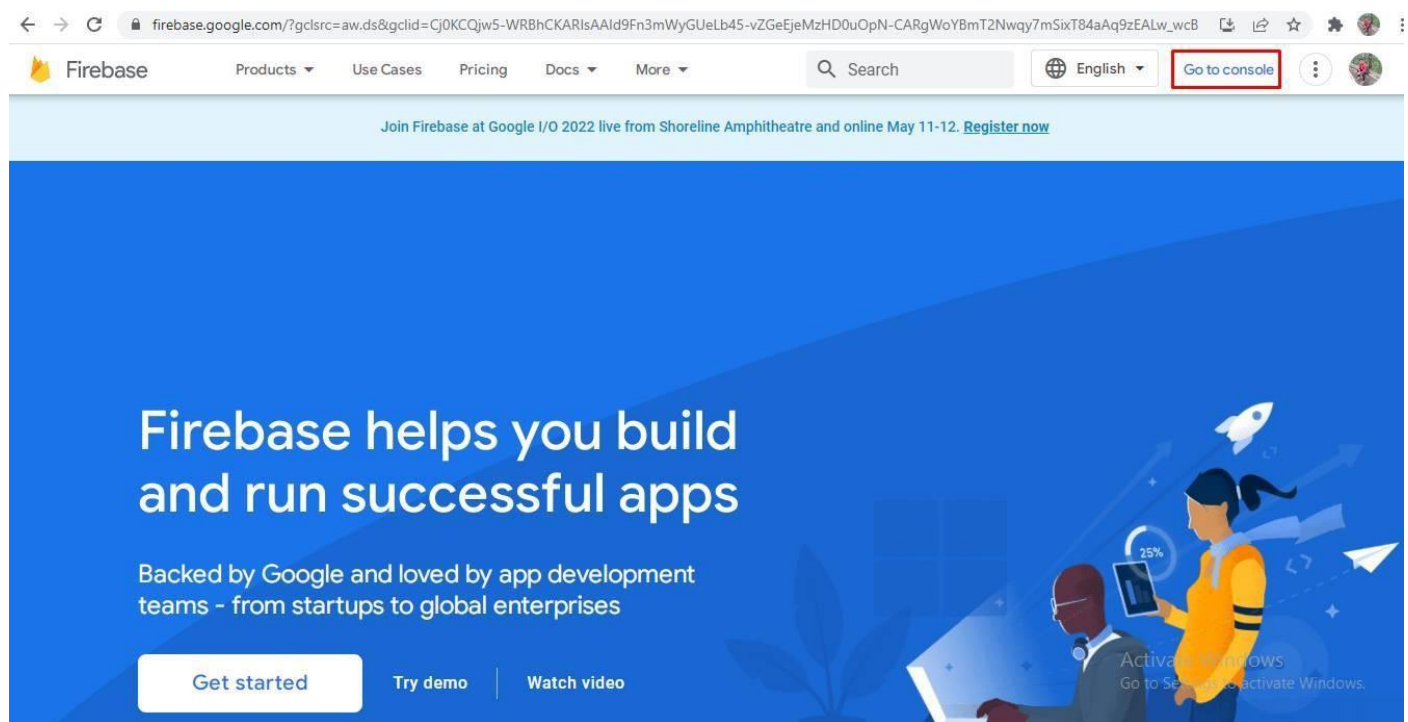
Step   3: Click on "Add project", it will redirect you to the new project creation page Enter your project name and click on "Continue" again click on "Continue" after that select "Default Account fo firebase" and then click on "Create Project" See

---
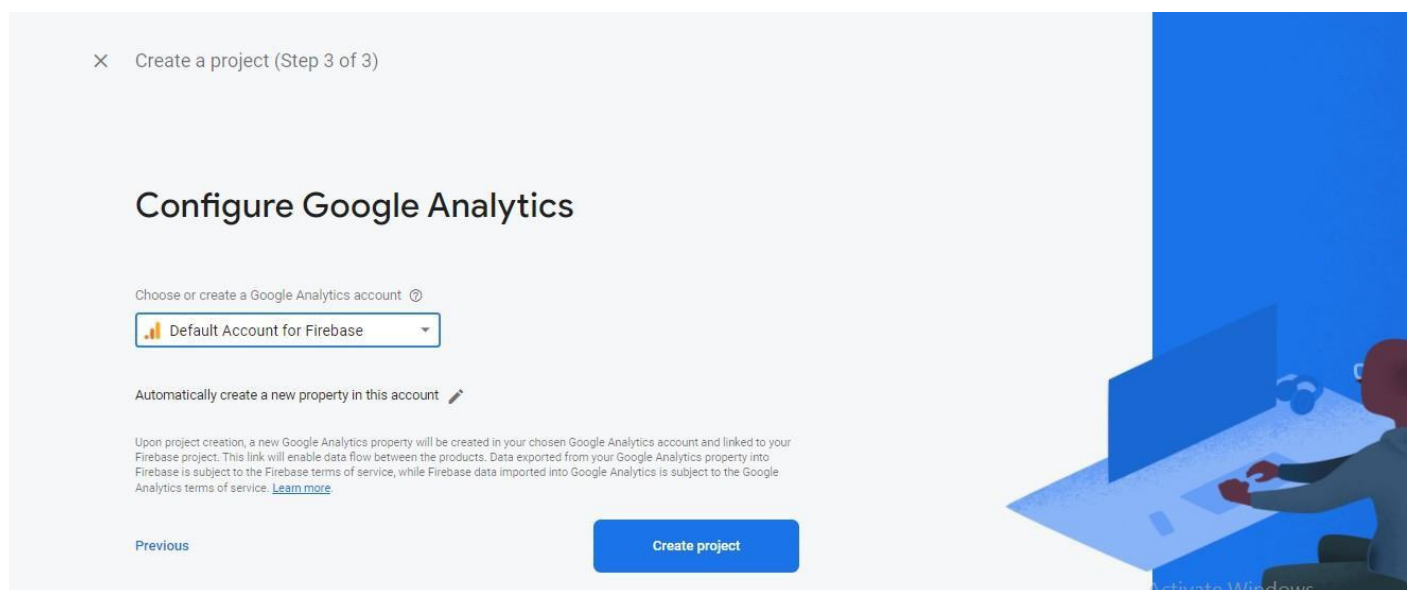
Step 4: After creating a new project it will redirect you to the overview page on this page click on icons like
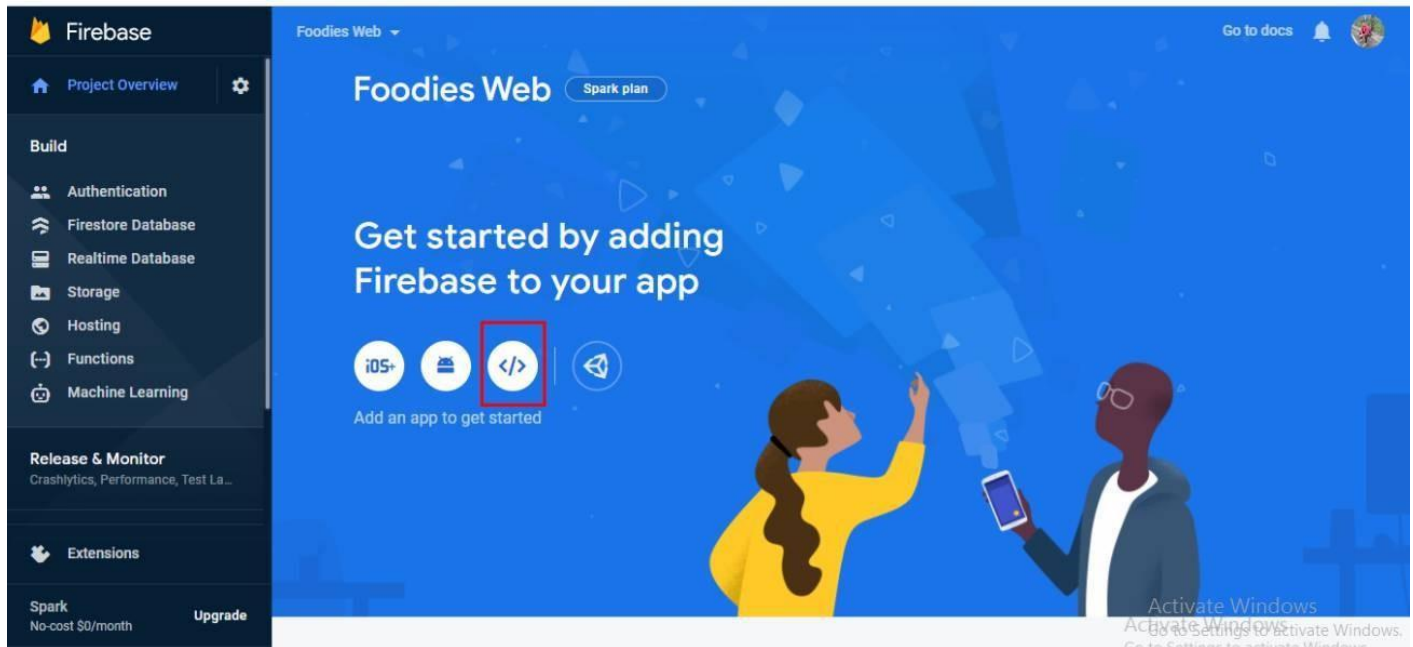
It will redirect you to the "Add Firebase to your web app" page enter your app nickname and click on "Register app" after clicking on it scroll down you can see details like below
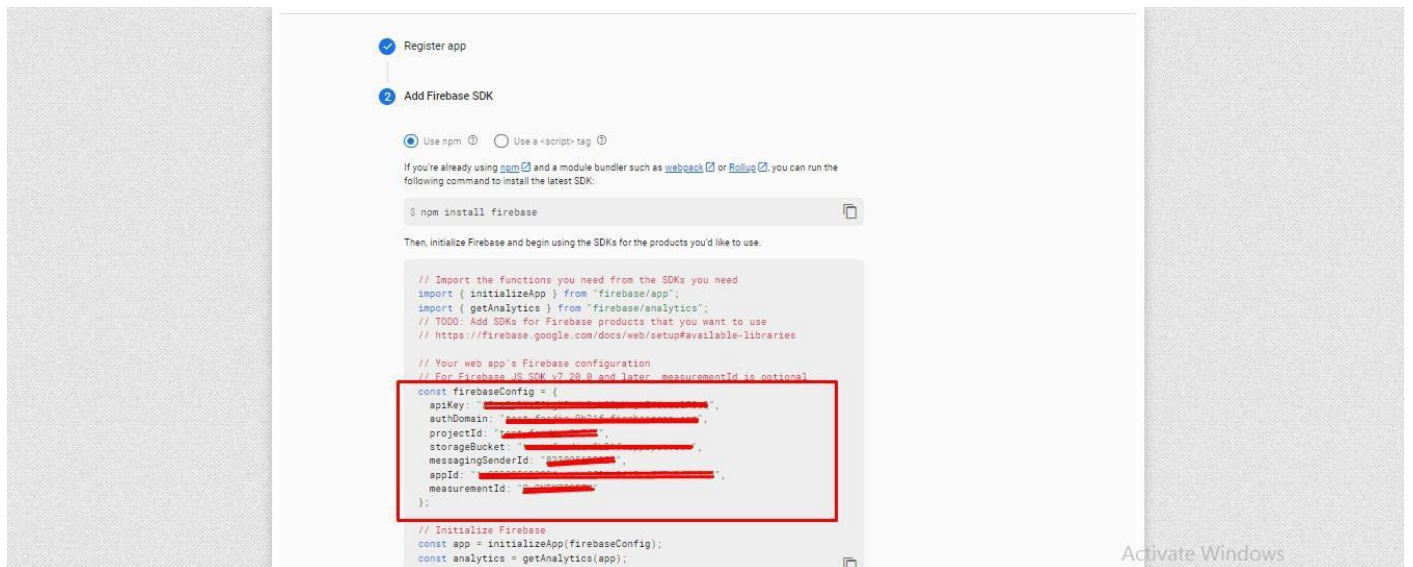
You can add these rebase project details in the project's .env le.

Step 5: After that click on "Firestore Database" in the left sidebar and select your project name from the drop-down and click on "Create database"
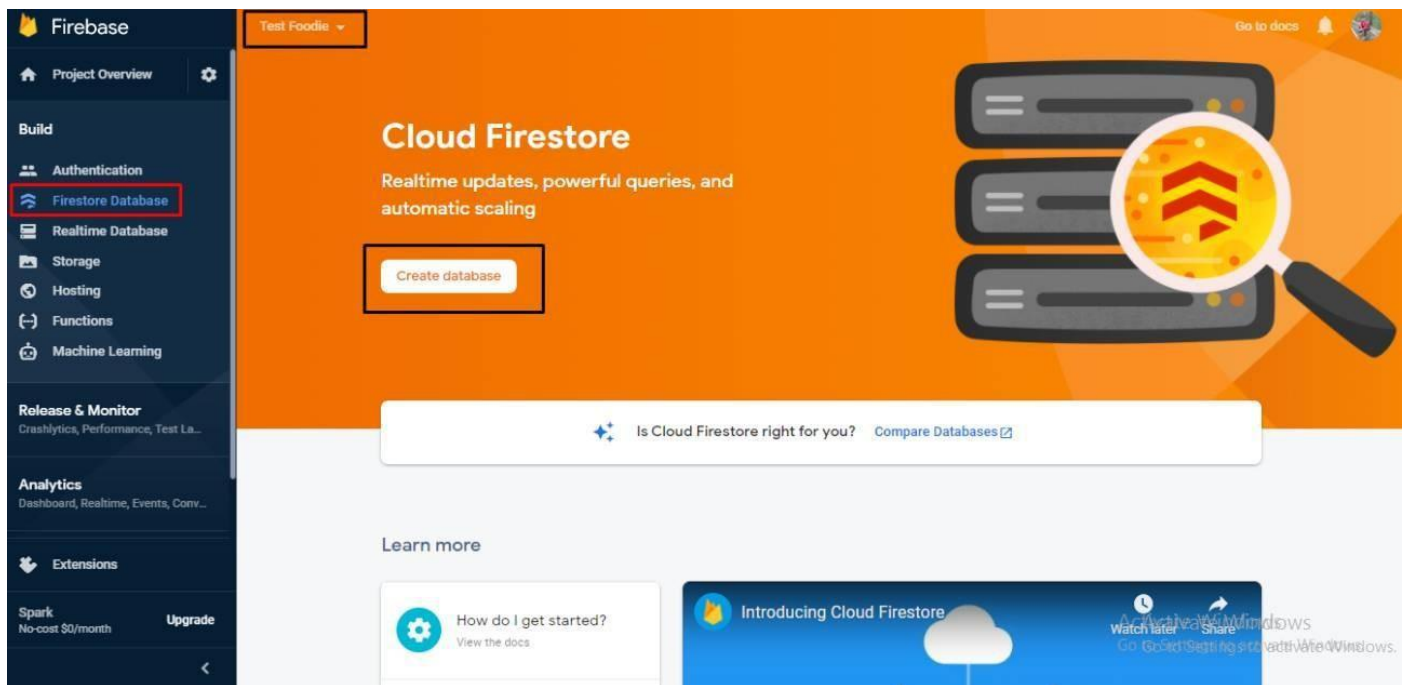
Step 6: Select the preferable option for you and click on the Next button then click on "Enable".

Step 7: Firstore Database Rules Update.
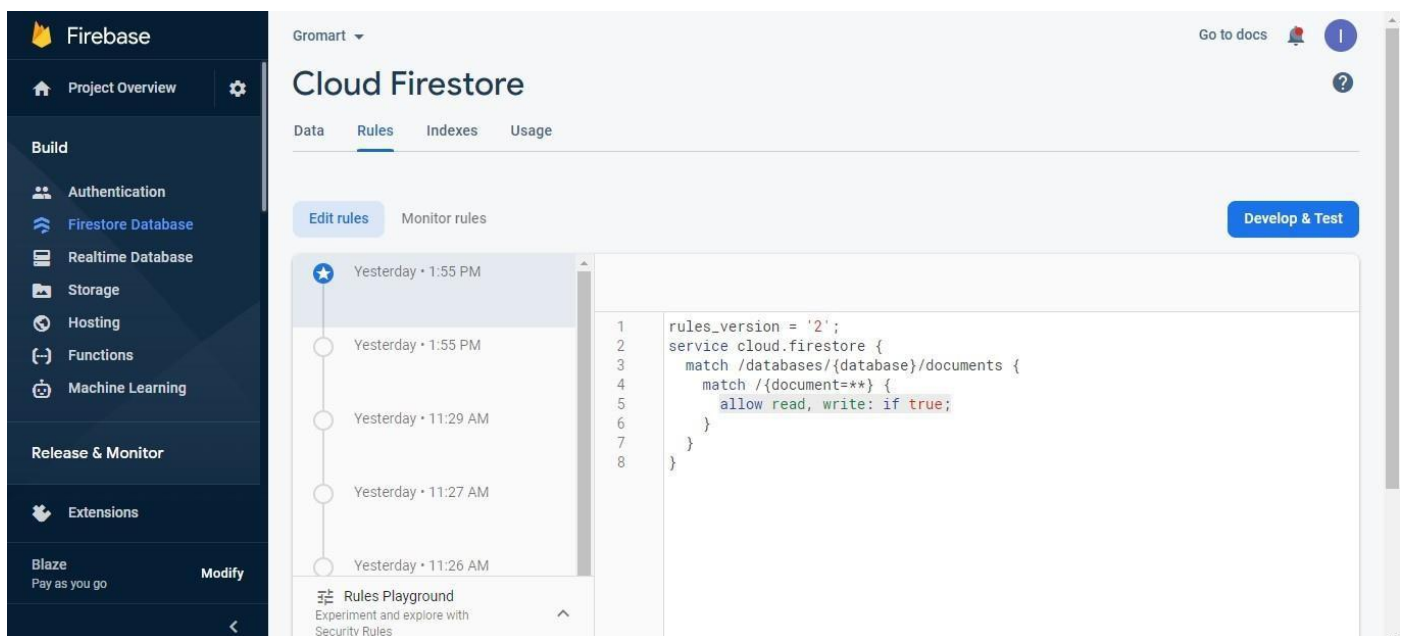
Order Tracking & Delivery Dispatching

☑

UberEats Clone Working Process:

- The customer places a new order

- The Store accepts/rejects the new order

- If the Store accepts the order, it is assigned to a driver that is available nearby the Store.

- The driver can decide to either accept or reject the order

- If the driver accepts the order, he is assigned to pick up and drop delivery

- The driver reaches the Store to pick up the order. Once the order is picked, the driver updates the status in the app for the customer
  After delivery, the driver can mark the order as completed on the driver app

- If the driver rejects the order, the dispatch starts again from a driver available nearby.

Customers can track all the order delivery status on their delivery app. App uses Flutter code for making most of the order updates mentioned above.

But there is some difference in the delivery system as it works on the backend side, it will have access to all available nearby drivers and Stores.

The delivery system is implemented via Firebase Function. You will be able to nd this delivery function in the downloaded archive, under `Firebase/functions/products/delivery.js`. Have a look at it to understand the delivery system closely.

The delivery system is implemented via Firebase Function. You will be able to nd this delivery function in the downloaded archive, under `Firebase/functions/products/delivery.js`. Have a look at it to understand the delivery system closely.

(https://youtu.be/mXDjXQ3YRys)
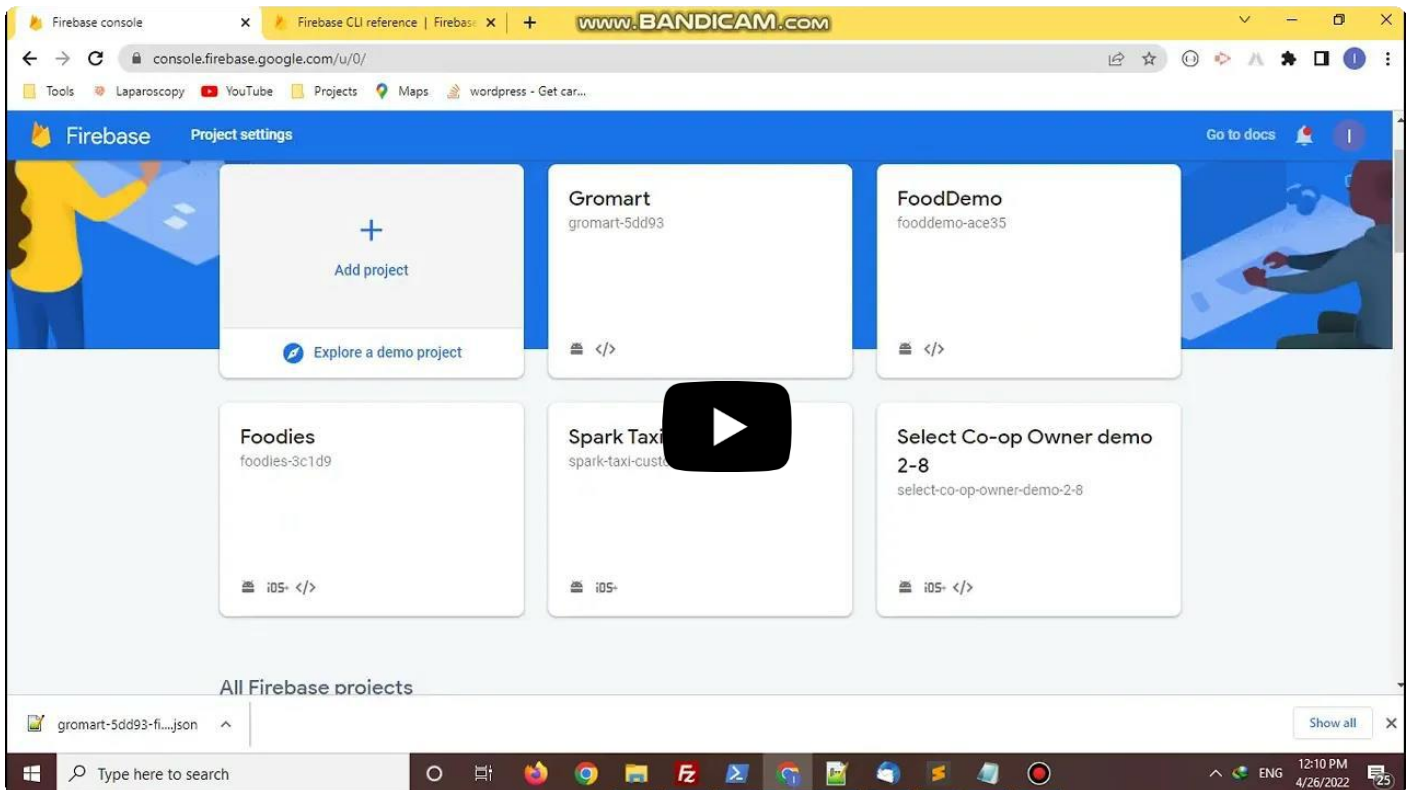
To operate your Firebase backend you'll need to follow these steps.

Step 1: Install Firebase Tools

If you haven't used Firebase functions before, you need to install this in the command line:

```
npm install -g firebase-tools
```

Step 2: Deploy the Firebase dispatch function

From the download archive locate the Firebase folder and run:

```
cd ~/Downloads/YOUR_PATH_TO_FIREBASE_FOLDER
npm install & firebase deploy
```

When it has been deployed, you'll be able to nd this function in your Firebase Console, under the Function tab.

It is even possible for you to inspect the logs of the delivery order, which will be shown for each order delivery that is in progress.

Deploy Firebase Functions

> **ⓘ** Note:
>
> This section has to be considered if you have purchased code that contains a "Firebase" or "FirebaseFunctions" folder inside.

All the necessary Firebase Functions have been initially coded by us, you just need to deploy these functions to your own Firebase account. This means that you need to upload the source code inside the FirebaseFunctions folder to your account. If a Firebase account has been created, nd a Firebase project, and 2 apps (iOS and Android).

Follow these steps for setup

Step 1: Setting up Node.js and the Firebase CLI

- Use the of cial Firebase doc (https:// rebase.google.com/docs/functions/get-started) if you encounter any issues.

- Node.js and Firebase CLI is needed to write functions and deploy them to the Cloud Functions. Install Node.js (https://nodejs.org/en/) and npm (https://www.npmjs.com/), Node Version Manager (https://github.com/nvm-sh/nvm/blob/master/README.md). After Node.js and npm are installed, install the Firebase CLI. Use: npm install -g rebase-tools, to install the CLI via npm.

Step 2: Initialize your project

An empty project containing sample code is created when you initialized Firebase SDK for cloud functions.

For initializing the project:

- Authenticate the rebase tool by running the rebase login via browser

- You need to create a new "MyFirebaseFunctions" empty folder

- Go to the MyFirebaseFunctions project directory and run rebase init functions

- With the tool, you can install npm. Before emulating or deploying your functions, you'll need to run npm install

- The tool gives you two options for language support:
    - — JavaScript
    - — TypeScript

- Please select JavaScript.

During initialization, package.json le is created that contains an important key: "engines": {"node": "12"}. Make sure it is Node.js version 12, before writing and deploying functions.

Step 3: Adding Instamobile Functions

As we are providing the full source code for your Firebase Functions, all you need to do is add that source code into your newly created MyFirebaseFunctions project. Follow these steps to proceed:

- — In the Instamobile's FirebaseFunctions folder, copy the "functions" subfolder to the clipboard.
- — Paste it to MyFirebaseFunctions folder, and OVERRIDE the functions folder. Now you have a Firebase Functions project, that is linked to your Firebase account with our source code that works on the mobile app

Step 4: Deploy Firebase Functions

To deploy Firebase Functions, you need to make sure you upgrade your Firebase Pricing Plan to Blaze.

Simply run the command (rebase deploy –only functions) in the project directory

Now you can go to your Firebase Console and check, as the functions have been deployed. It is possible to see the logs for each function, understand the output, and know when it gets called.

Step 5: Watch Firebase Functions for Errors

To run properly, some Firebase functions need the creation of indexes on certain Firestore collections. Use the app and watch the logs for the Firebase Functions in the console. If you happen to get an error of missing index, simply click on the URL of that error, and the index will get created automatically. There is no need to deploy functions again, but wait until the index is created, before testing the app again.

✅ GroMart Firestore Database Collection Import Export

Video For eMart Firestore Database Collection Import Export

(https://www.youtube.com/watch?v=REGdo2qqI9I)
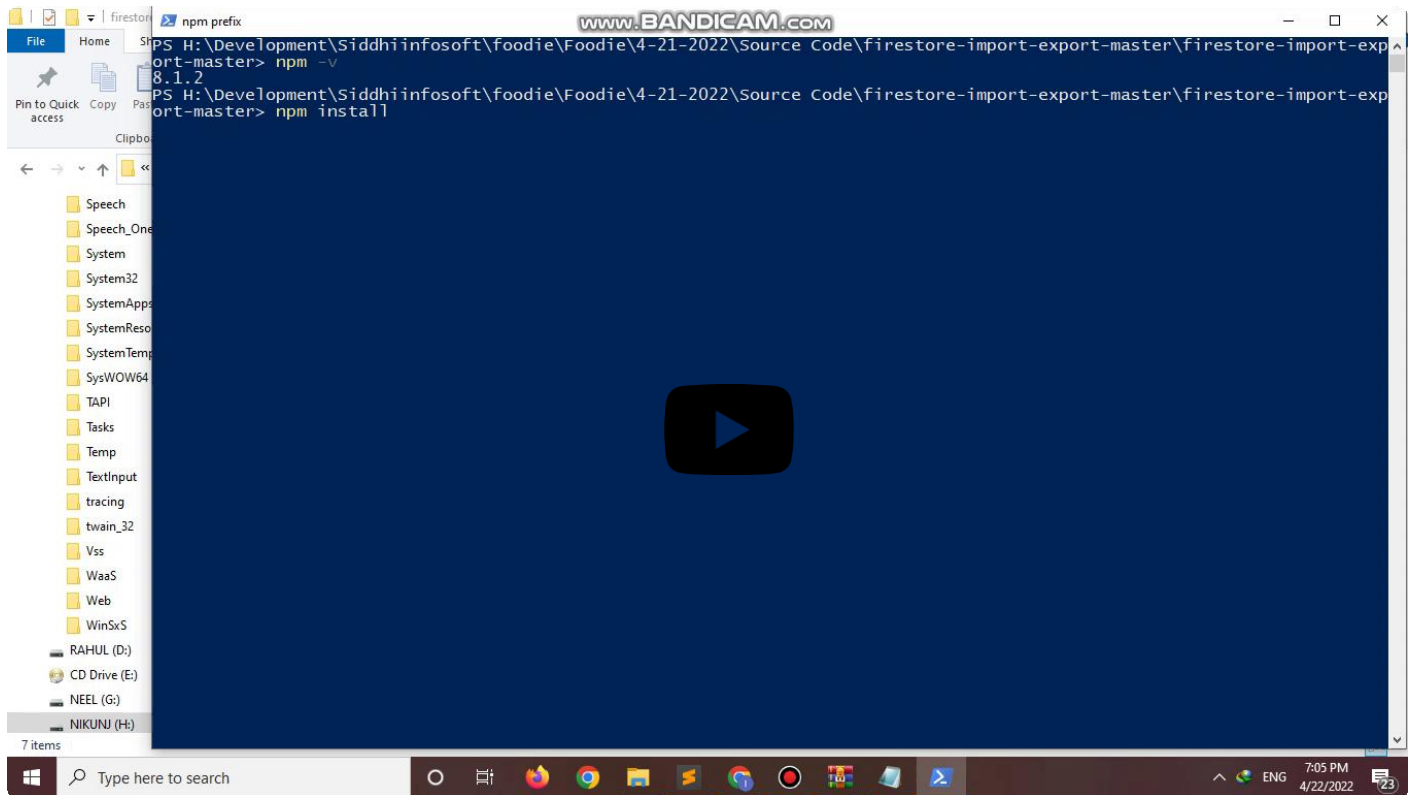
Step   1: Setup NPM in your Computer URL:

https://nodejs.org/en/download/

Step 2: Extract Source Code of firestore-import-export-master.zip

Step 3: We have to open command prompt on project folder and run command :
npm install

Step 4: Setup Firebase Project if you not created.

Step 5: Configure serviceAccountKey.json file. you can get from forbase Project settings

Go to ->Service account -> Select Node.js -> Generate new private key
Wait untill create key this will auto download.
and Replace with current serviceAccountKey.json

Step   6: Run command on project folder bellow import commands

All Collections are in Folder name "eMartsDataSeed"
One By one You have to run command for import each collection.

IMPORT Commands:

```
node import.js eMartDataSeed/SOS.json
node import.js eMartDataSeed/banner_items.json
node import.js eMartDataSeed/booked_table.json
node import.js eMartDataSeed/brands.json
node import.js eMartDataSeed/car_make.json
node import.js eMartDataSeed/car_model.json
node import.js eMartDataSeed/channel_participation.json
node import.js eMartDataSeed/channels.json
node import.js eMartDataSeed/cms_pages.json
node import.js eMartDataSeed/complaints.json
```

```
node import.js eMartDataSeed/coupons.json
node import.js eMartDataSeed/currencies.json
node import.js eMartDataSeed/driver_payouts.json
node import.js eMartDataSeed/dynamic_notification.json
node import.js eMartDataSeed/email_templates.json
node import.js eMartDataSeed/favorite_item.json
node import.js eMartDataSeed/favorite_vendor.json
node import.js eMartDataSeed/items_review.json
node import.js eMartDataSeed/notifications.json
node import.js eMartDataSeed/order_transactions.json
node import.js eMartDataSeed/parcel_categories.json
node import.js eMartDataSeed/parcel_coupons.json
node import.js eMartDataSeed/parcel_orders.json
node import.js eMartDataSeed/parcel_weight.json
node import.js eMartDataSeed/payouts.json
node import.js eMartDataSeed/popular_destinations.json
node import.js eMartDataSeed/promos.json
node import.js eMartDataSeed/rating.json
node import.js eMartDataSeed/referral.json
node import.js eMartDataSeed/rental_coupons.json
node import.js eMartDataSeed/rental_orders.json
node import.js eMartDataSeed/rental_vehicle_type.json
node import.js eMartDataSeed/reports.json
node import.js eMartDataSeed/review_attributes.json
node import.js eMartDataSeed/rides.json
node import.js eMartDataSeed/sections.json
node import.js eMartDataSeed/services.json
node import.js eMartDataSeed/settings.json
node import.js eMartDataSeed/story.json
node import.js eMartDataSeed/tax.json
node import.js eMartDataSeed/users.json
node import.js eMartDataSeed/vehicle_type.json
node import.js eMartDataSeed/vendor_attributes.json
node import.js eMartDataSeed/vendor_categories.json
node import.js eMartDataSeed/vendor_orders.json
node import.js eMartDataSeed/vendor_products.json
node import.js eMartDataSeed/vendors.json
node import.js eMartDataSeed/wallet.json
```

You also can check how to export collection in readme.txt file

```
# Export commands:

node export.js SOS
node export.js banner_items
node export.js booked_table
node export.js brands
node export.js car_make
node export.js car_model
node export.js channel_participation
node export.js channels
node export.js cms_pages
node export.js complaints
node export.js coupons
node export.js currencies
node export.js driver_payouts
node export.js dynamic_notification
node export.js email_templates
node export.js favorite_item
node export.js favorite_vendor
node export.js items_review
node export.js notifications
node export.js order_transactions
node export.js parcel_categories
node export.js parcel_coupons
node export.js parcel_orders
node export.js parcel_weight
node export.js payouts
node export.js popular_destinations
node export.js promos
node export.js rating
node export.js referral
node export.js rental_coupons
node export.js rental_orders
node export.js rental_vehicle_type
node export.js reports
node export.js review_attributes
node export.js rides
node export.js sections
node export.js services
node export.js settings
node export.js story
node export.js tax
node export.js users
node export.js vehicle_type
node export.js vendor_attributes
node export.js vendor_categories
node export.js vendor_orders
node export.js vendor_products
node export.js vendors
node export.js wallet
```

✔️ Demo User Authentication Import

Step 1: Get files from the source zip

Step 2: Get serviceAccountKey.json from firbase configuration

link:https://clemfournier.medium.com/how-to-get-my- rebase-service-account-
    key- le-f0ec97a21620 (https://clemfournier.medium.com/how-to-get-my- rebase-
    service-
account-key- le-f0ec97a21620)

Step 3: npm install

Step 4: run command: "node import-user.js"

You can see proccess of import users in rstore.(Note: Do not close terminal on
runing process.)

```
starting update for user with email: testnp04@foodie.com
Old user found: UserRecord {
  uid: 'w9sLqi2OtzVokJOV8WAJkFSO4N04',
  email: 'testnp04@foodie.com',
  emailVerified: false,
  displayName: undefined,
  photoURL: undefined,
  phoneNumber: undefined,
  disabled: false,
  metadata: UserMetadata {
    creationTime: 'Mon, 10 Jan 2022 09:21:25 GMT',
    lastSignInTime: null,
    lastRefreshTime: null
  },
  providerData: [],
  passwordHash: undefined,
  passwordSalt: undefined,
  tokensValidAfterTime: 'Mon, 10 Jan 2022 09:21:25 GMT',
  tenantId: undefined
}
Old user deleted.
New user data ready:  {
  disabled: false,
  displayName: undefined,
  email: 'testnp04@foodie.com',
  emailVerified: false,
  phoneNumber: undefined,
  photoURL: undefined,
  uid: 'w9sLqi2OtzVokJOV8WAJkFSO4N04'
}
New user created:  UserRecord {
  uid: 'w9sLqi2OtzVokJOV8WAJkFSO4N04',
  email: 'testnp04@foodie.com',
  emailVerified: false,
  displayName: undefined,
  photoURL: undefined,
  phoneNumber: undefined,
  disabled: false,
```