eMart Web Documentation

1. Introduction

Welcome to eMart, the all-in-one solution for your everyday needs. Our platform seamlessly integrates multiple services into one convenient app, providing you with everything from delicious meals to reliable transportation, all with the click of a button. With eMart, convenience is just a tap away.

eMart is a comprehensive solution for multi-vendor delivery systems, created with Flutter, Firebase, and the Laravel Framework, known for its expressive and elegant syntax. This product includes three apps: Customer, Store, and Driver. This documentation provides guidance for setting up the eMart website and applications. It's recommended to use Laravel Homestead, a virtual machine that satisfies all system requirements, as your local development environment. Ensure your server meets the following requirements.

Key Features:

1.1 : Multi Vendor Items Delivery:

Craving your favorite dish? Explore a wide range of cuisines from local restaurants and food vendors. Order in with ease and have your meal delivered straight to your doorstep. With eMart's multivendor food delivery service, you're spoiled for choice.

1.2 : eCommerce Marketplace:

Experience non-stop shopping excitement with our extensive eCommerce marketplace. Browse through a diverse selection of products from various vendors, ranging from electronics and clothing to home goods and accessories. Find exactly what you need, all in one place.

1.3 : Parcel Delivery:

Need to send a package? No problem. Our parcel delivery service connects you with reliable couriers who can deliver your parcels safely and on time. Whether it's a small envelope or a large package, we've got you covered.

1.4 : Taxi Booking:

Getting around town has never been easier. Book a taxi with just a few taps and enjoy hassle-free transportation to your destination. Our network of drivers ensures prompt service and comfortable rides wherever you need to go.

1.5 : Car Rental:

Planning a road trip? Rent a car through eMart and hit the road with confidence. Choose from a variety of vehicles to suit your needs, whether it's a compact car for city driving or an SUV for a family adventure. Our seamless booking process makes renting a car a breeze.

1.6 : Admin Panel:

For vendors, our user-friendly admin panel provides all the tools you need to manage your products, orders, and deliveries efficiently. Monitor sales, update inventory, and communicate with customers—all from one centralized dashboard.

1.7 : Website Integration:

Our website complements our app, offering users another convenient way to access our services. Whether you prefer browsing on your desktop or mobile device, our website provides the same seamless experience as our app.

1.8 : Secure Payments:

Shop and pay with confidence thanks to our secure payment system. We support various payment methods, including credit/debit cards, mobile wallets, and cash on delivery, ensuring a smooth and secure transaction every time.

1.9 : Customer Support:

Have questions or need assistance? Our dedicated customer support team is here to help. Whether it's tracking an order, resolving an issue, or providing general assistance, we're just a message away.

1.10 : Promotions and Deals:

Take advantage of special promotions and deals offered by our vendors. From discounts on food orders to exclusive offers on car rentals, there's always a way to save with eMart.

Experience the Convenience of eMart Today:

Whether you're craving a meal, shopping for essentials, sending a package, booking a ride, or renting a car, eMart has you covered. Download the app or visit our website to discover the ultimate convenience of eMart. Welcome to a world where everything you need is just a tap away. Welcome to eMart.

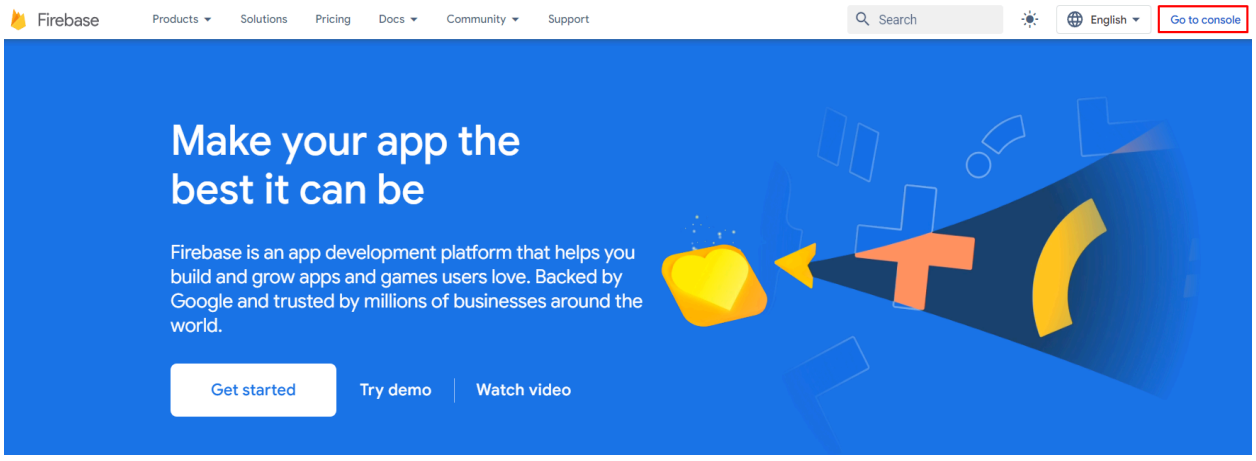
2. Server Requirement

PHP Version (8.2 or above)
SSL Certificate
Google Map API Key
Google Firebase Account with Upgraded (Blaze) Plan

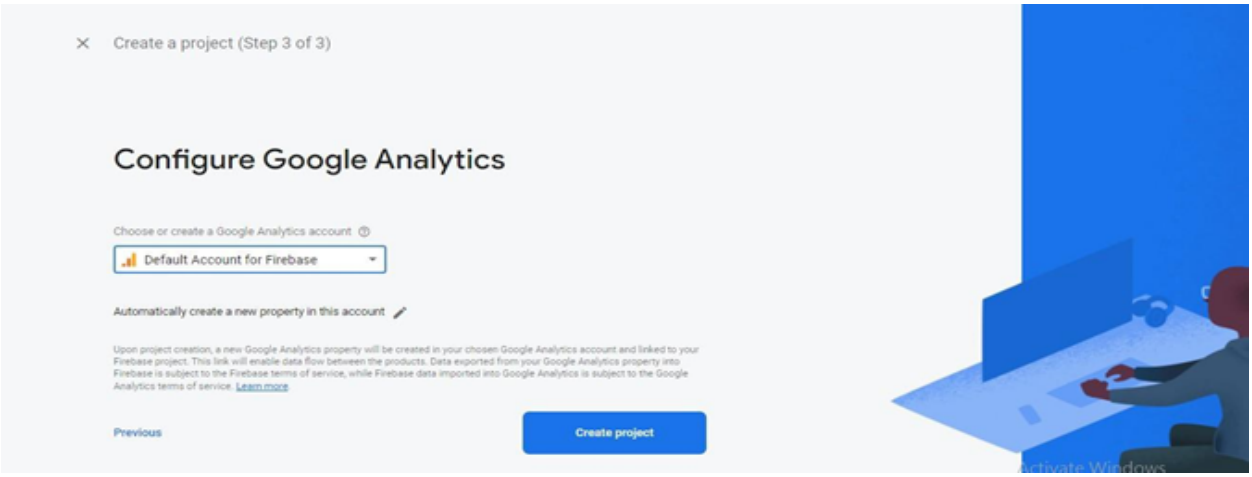
3. Create Firebase Project

3.1 : Go to firebase console through this link: https://firebase.google.com/

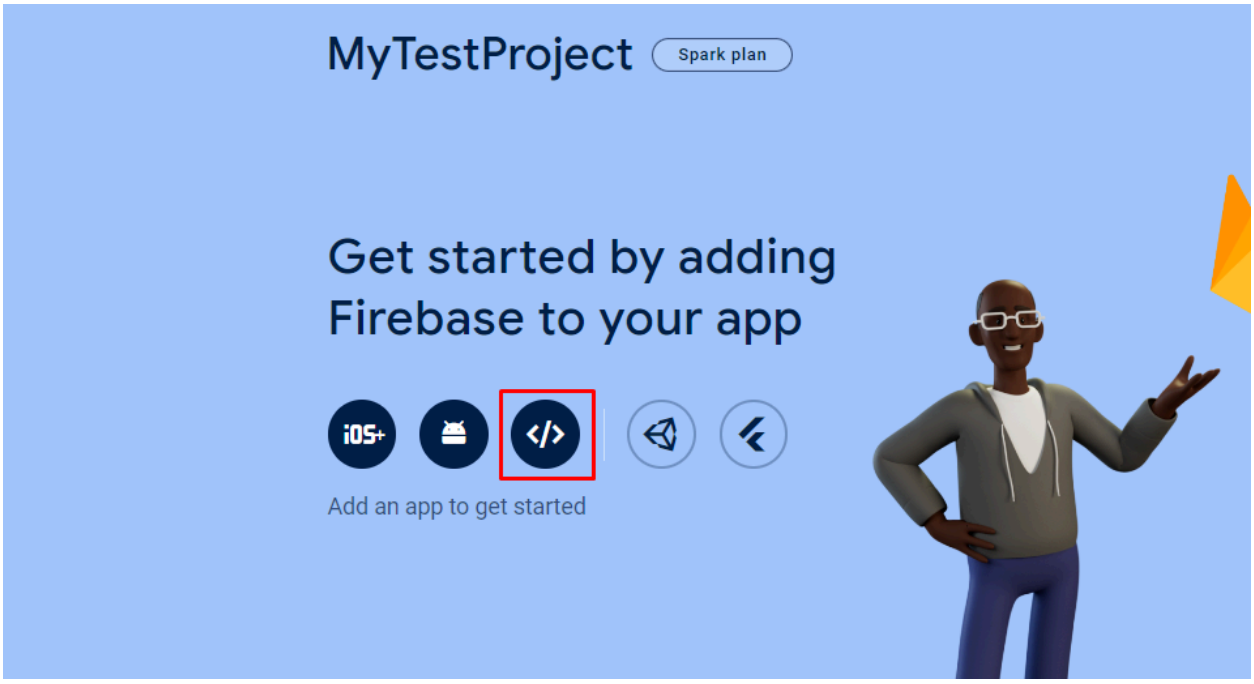
3.2 : Click on “Go to console” in the top right corner.



3.3 : 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”



3.4 : After successfully creating a new project, you will be directed to the overview page. From there, simply navigate to the icons link to proceed further.



Upon redirection, you'll land on the 'Add Firebase to your web app' page. Here, input your app's nickname and proceed by clicking 'Register app'. Afterward, scroll down to find detailed information as outlined below.

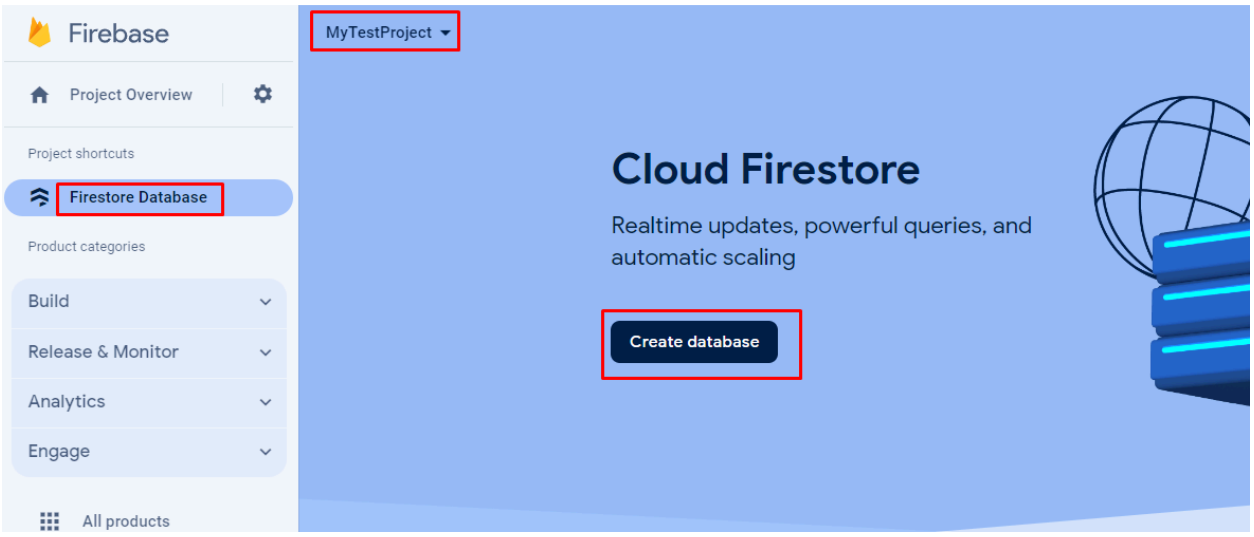
```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "[redacted]",
  authDomain: "[redacted]",
  projectId: "[redacted]",
  storageBucket: "[redacted]",
  messagingSenderId: "[redacted]",
  appId: "[redacted]",
  measurementId: "[redacted]"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

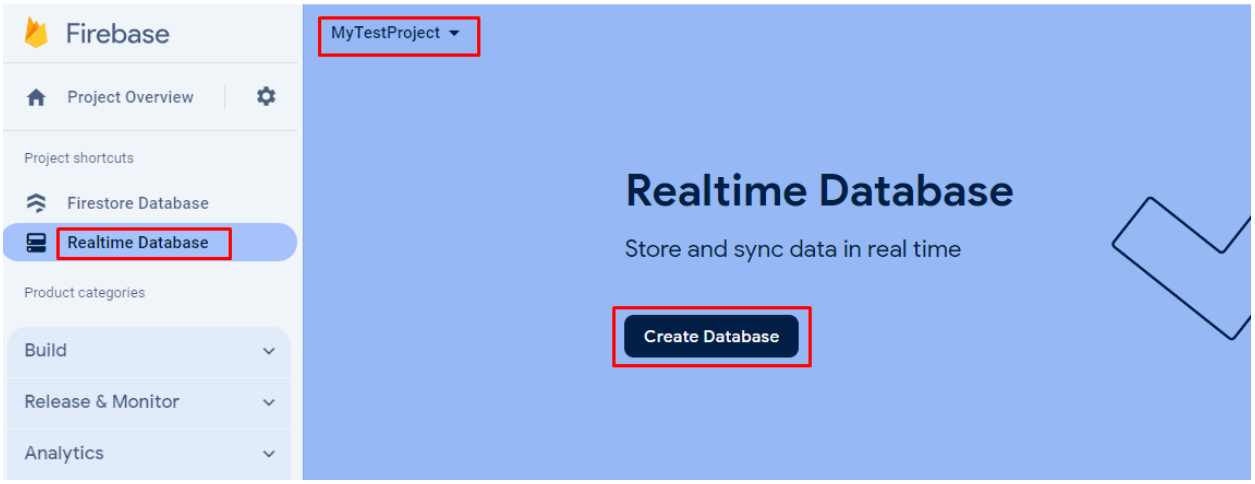
All of these details will be incorporated into both the app and admin panel documentation for comprehensive reference and seamless integration.

3.5 : Next, navigate to 'Firestore Database' in the left sidebar. Select your project name from the drop-down menu, and then proceed by clicking on 'Create database'.



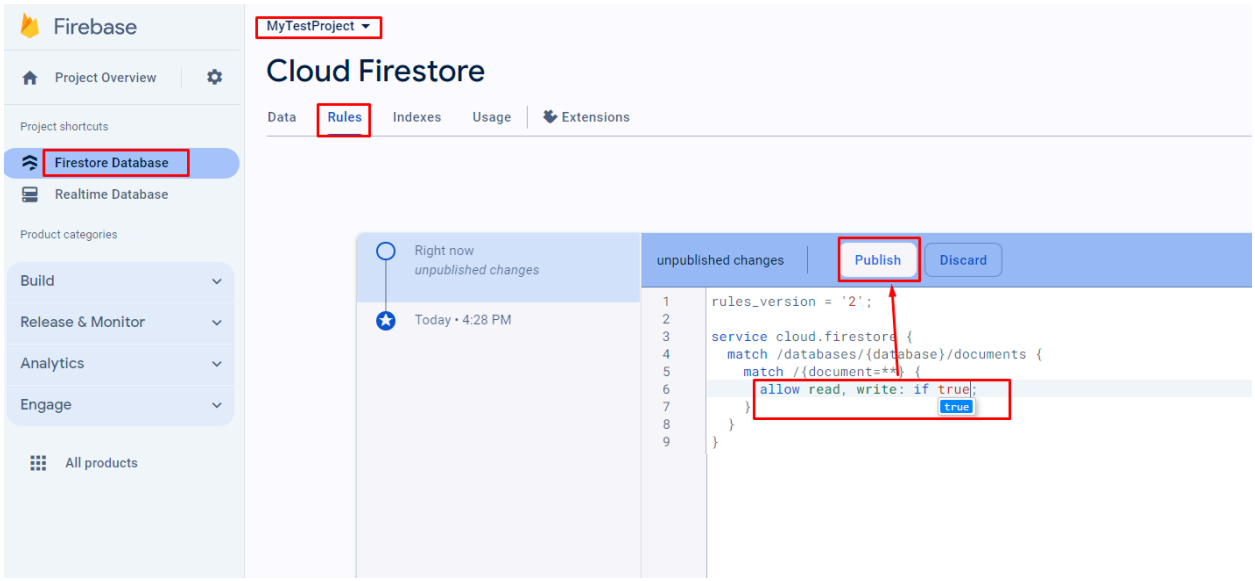
3.6 : Choose your preferred option and click on the 'Next' button. Subsequently, proceed by clicking on "Enable"

3.7 : Following that, locate "Realtime Database" in the left sidebar. Select your project name from the drop-down menu, and proceed by clicking on "Create Database"

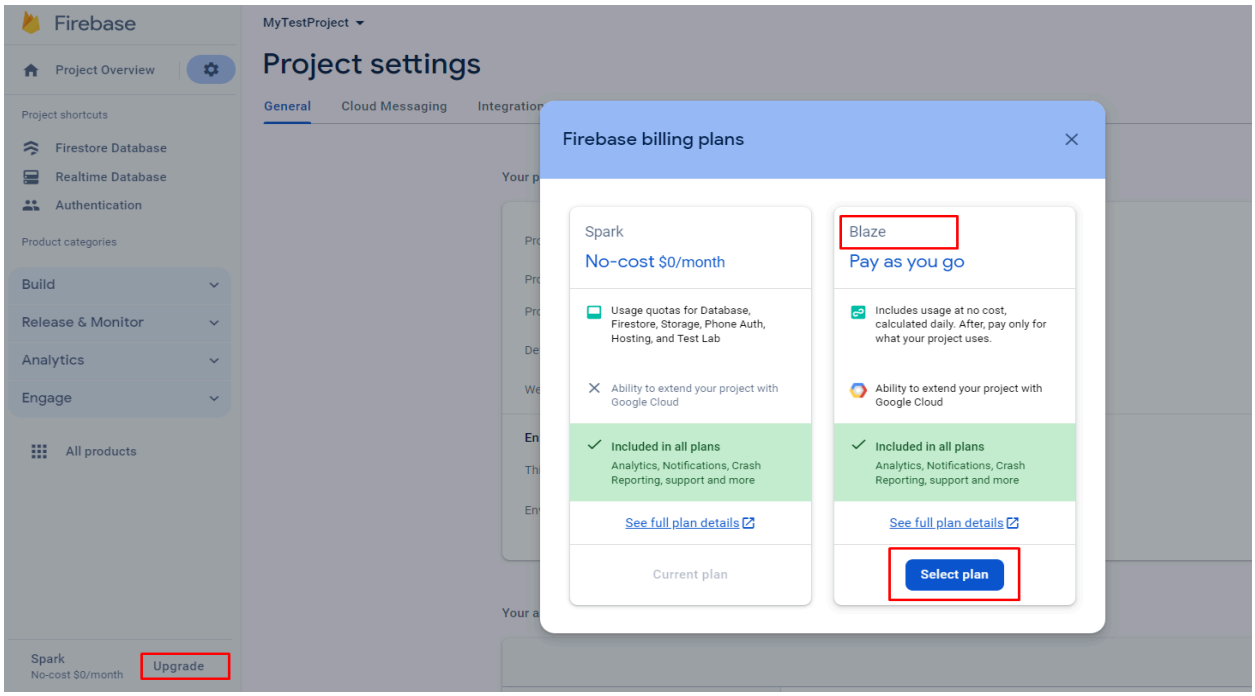


3.8 : Choose your preferred option and proceed by clicking the "Next" button, then click on "Enable" to activate the selected feature.

3.9 : Firstore Database Rules Update.



Step 10: Upgrade your firebase plan.



See Video: [How to setup & configure Firebase Project?]

4. Firestore Database Collection Import Export

To perform Firebase Collection Import Export, follow these straightforward steps:

4.1 : To set up NPM on your computer, download Node.js from the following link: https://nodejs.org/en/download/ Node.js Download Page.

4.2 : Unzip the source code file named "Firebase Import Export Collections.zip"

4.3 : If you haven't already created a Firebase project, set it up now.

4.4 : Configure the credentials.json file, which you can obtain from your Firebase Project settings. Then navigate to the Service account, then select Node.js. Generate a new private key and wait until the key is created. It will automatically download and replace the current credentials.json file.

4.5 Navigate to the extracted Firebase Import Export Collections zip file path and then press and hold the Ctrl+Shift buttons. While holding them, right-click the mouse button, and select "Open PowerShell window here" from the context menu. This will open a PowerShell window where you can run the import/export command.

4.6 : Execute the following commands to perform import/export operations for collections:

To import all collections, execute the following command:

```
npx -p node-firestore-import-export firestore-import -a
credentials.json -b collections.json
```

To export all collections, execute the following command:

```
npx -p node-firestore-import-export firestore-export -a
credentials.json -b collections.json
```

Once the export command is executed, the collections.json file will be downloaded.

Please Note: Prior to executing the above command, ensure that you have correctly updated your Firebase credentials in the credentials.json file.

See video: How to Import Export Collections in Firebase?

5. Order Tracking & Delivery Dispatching

5.1 : The customer initiates a new order

5.2 : The store approves/rejects the new order

5.3 : If the store accepts the order, it is assigned to a nearby available driver.

5.4 : The driver can choose to either accept or decline the order. If the driver accepts the order, they are assigned to pick up and deliver the items.

5.5 : The driver arrives at the store to collect the order. Once picked up, the driver updates the order status in the app for the customer. After delivery, the driver can mark the order as completed in the driver app.

5.6 : If the driver rejects the order, the dispatch process restarts to find another available driver nearby.

5.7 : Customers can track the delivery status of all their orders on their delivery app. The app utilizes Flutter code to handle most order updates. However, the backend system, which manages access to all available nearby drivers and stores, facilitates the delivery process.

6. Deploy Firebase Cloud Functions

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 Firebase Functions folder to your account. If a Firebase account has been created, find a Firebase project, and 2 apps (iOS and Android)

6.1 : Setting up Node.js and the Firebase CLI

For comprehensive guidance, refer to the official Firebase documentation on getting started: "Write, test, and deploy your first functions" in Cloud Functions for Firebase.

You'll need Node.js and the Firebase CLI to write functions and deploy them to Cloud Functions.

To set up NPM on your computer, download Node.js from the following link: Node.js download. Once Node.js is installed, proceed to install the Firebase CLI.

Use the following command to install the CLI via npm:

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

If you've already set up the Firebase tools, you can simply run the following command:

```
npm install
```

These steps will ensure you're ready to start writing and deploying functions with Firebase. If you encounter any issues, consult the Firebase documentation for troubleshooting assistance.

6.2 : Initialize your project

To initialize the project, authenticate the Firebase tool by running the following command. You'll be prompted to log in to your account via your web browser:

```
firebase login
```

6.3 : Implementing Cloud Functions

Since we're providing the complete source code for your Firebase Cloud Functions:

Extract the zip file Order Tracking Firebase Function folder.zip.

Fill in the necessary credentials in the following files located within the zip:

1. .firebaserc (Add your Firestore Project ID)
2. index.js (Add your Firestore database URL)
3. serviceAccountKey.json (Add your Firebase service account credentials)

With these steps, you've successfully set up the required credentials.

6.4 : Deploy Firebase Functions

Simply run the following command in the Order Tracking Firebase Function > functions directory.

```
firebase deploy --only functions
```

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.

See Video: **How to deploy cloud functions in firebase?**

7. Demo User Authentication Import

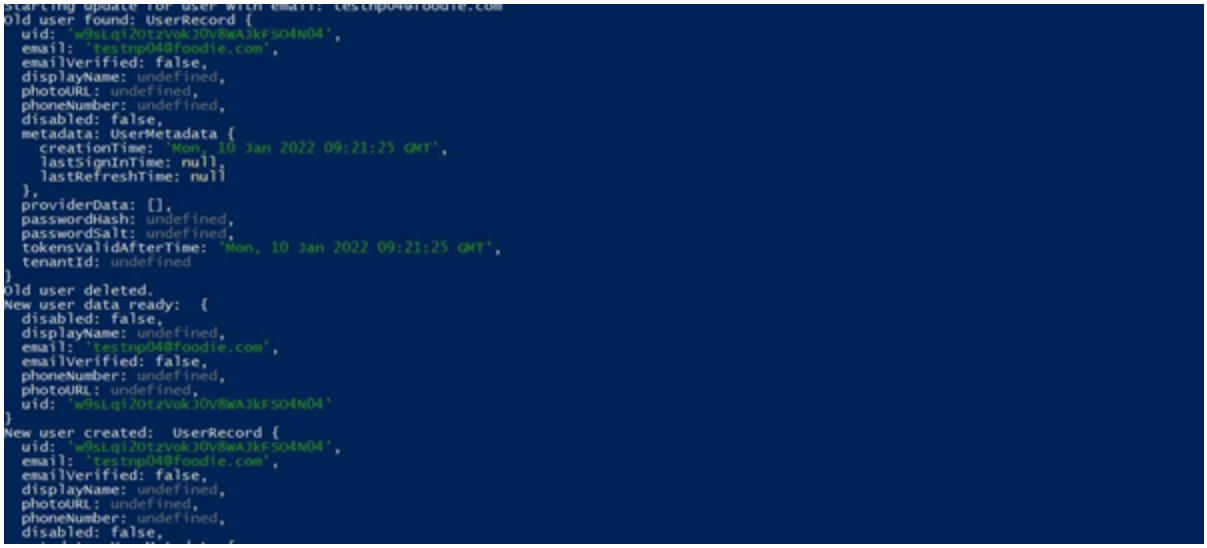
7.1 : Get files from the source zip

7.2 : Get serviceAccountKey.json from firebase configuration

7.3 : `npm install`

7.4 : Run command `node import-user.js`

You can see the process of importing users in the firestore.
(Note: Do not close the terminal during the running process)



8. Firestore Database Indexing

To perform Firebase indexing, follow these straightforward steps:

8.1 : To set up NPM on your computer, download Node.js from the following link: https://nodejs.org/en/download/

8.2 : Unzip the source code file named "Firebase Indexing.zip" here.

8.3 : Navigate to the extracted directory of "Firebase Indexing" zip file. Press and hold the ctrl+shift buttons, then right-click the mouse button. From the context menu, select "Open PowerShell window here" to launch Windows PowerShell and execute import/export commands.

8.4 : Execute the command `firebase login` to log in to Firebase, if you haven't already done so.

8.5 : Execute the command `firebase init`

8.6 : Proceed with Y and press the enter button.

8.7 : Choose the Option > Firestore: Configure security rules and index files for Firestore.

Please Note: Choose the arrow down key to navigate and select options, and press the space button to confirm your selection.

8.8 : Choose the Option > Use an existing project

8.9 : Choose your project

8.10 : Press Enter > ? What file should be used for Firestore Rules? firestore.rules

8.11 : Press Enter > ? What file should be used for Firestore indexes? (firestore.indexes.json)

8.12: Now, the firestore.indexes.json file will be downloaded. Open this file and copy all the code from firestore_indexes.json file, then paste it into firestore.indexes.json file.

8.13: Now execute the command `firebase deploy --only firestore:indexes` to import indexing in firestore.

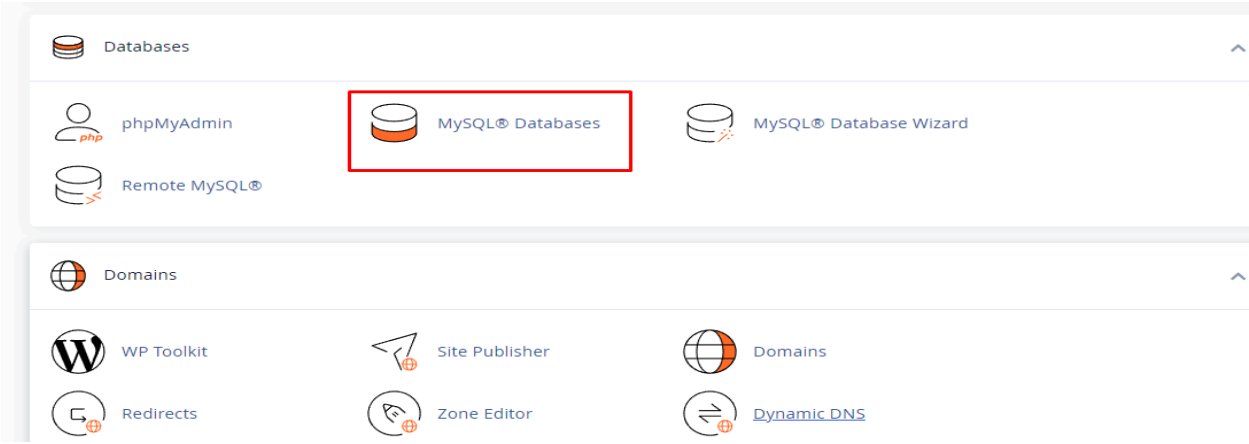
See video: How to Import Firestore Database Indexing in Firebase?

9. Admin Panel Setup

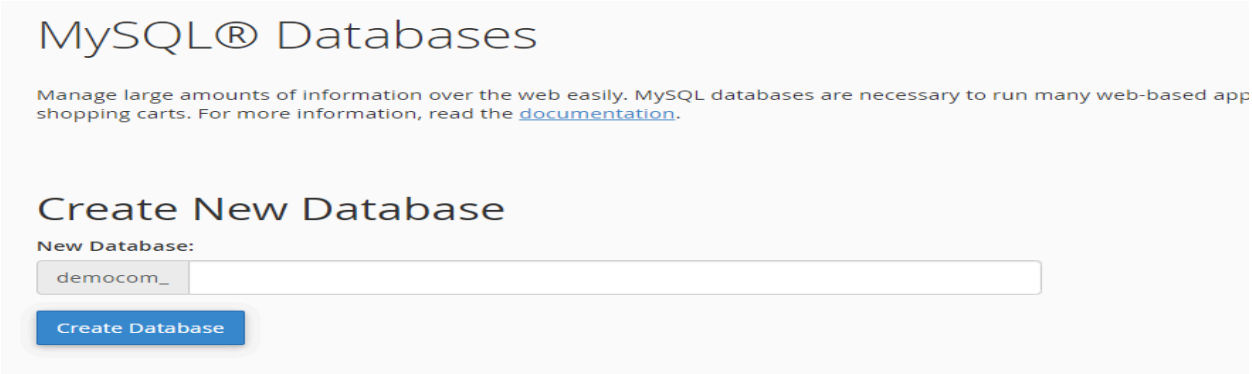
Ensure that your server meets the requirements outlined in the documentation's Server Requirement section. Upon downloading the code, upload the admin zip file to your server and extract it. The admin panel can be installed on either a domain or a subdomain.

9.1 Create Database

9.1.1 : Create a new database from your server's MYSQL database.

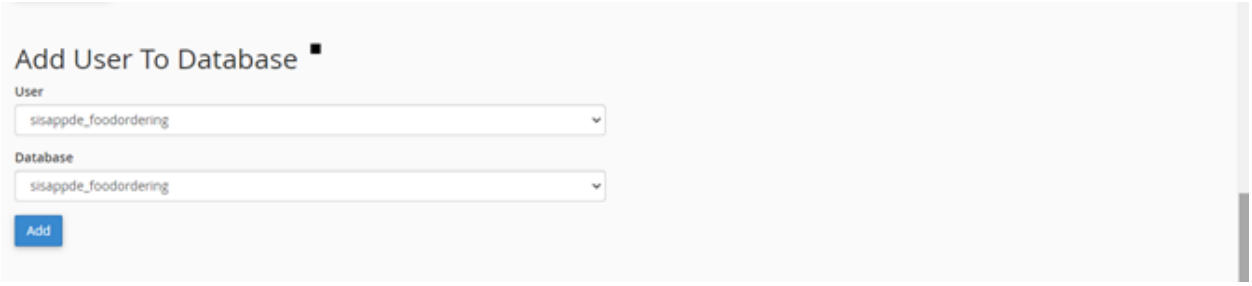


9.1.2 : Create New Database

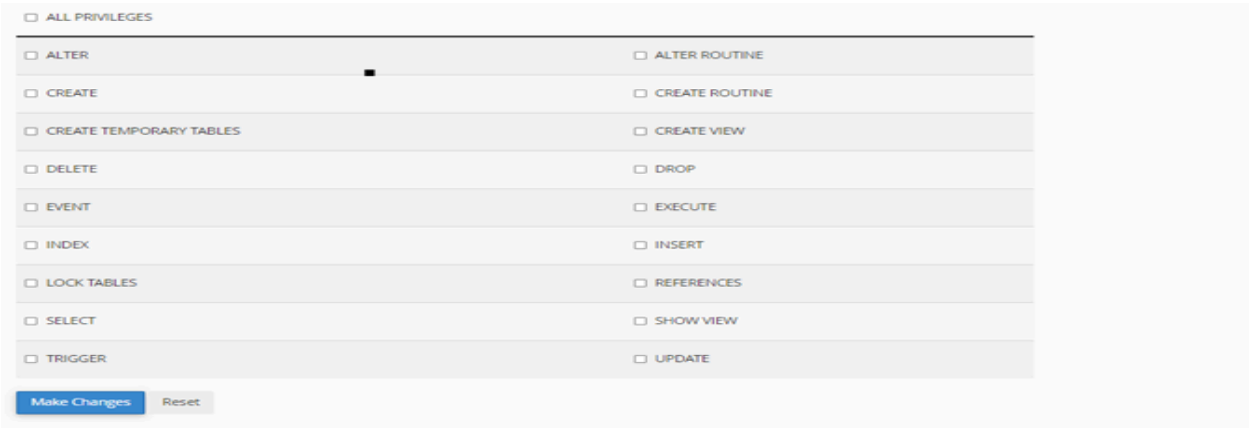


9.1.3 : Create a new database user

9.1.4 : Connect the database to the newly created database user.



9.1.5 : Grant all privileges to your user by selecting "All PRIVILEGES" and then clicking on "Make Changes"



9.2 Configure Admin Panel

9.2.1 : Once you've downloaded the code, upload the admin zip file to your server and extract it. The admin panel can be installed on either a domain or subdomain.

9.2.2 : Next, you'll need to configure the following settings in your .env file: Database host, Database name, Database username, Database password, and your Firebase project credentials.

```
FIREBASE_APIKEY=
FIREBASE_AUTH_DOMAIN=
FIREBASE_DATABASE_URL=
FIREBASE_PROJECT_ID=
FIREBASE_STORAGE_BUCKET=
FIREBASE_MESSAAGING_SENDER_ID=
FIREBASE_APP_ID=
FIREBASE_MEASUREMENT_ID=
FIREBASE_KEY=
```

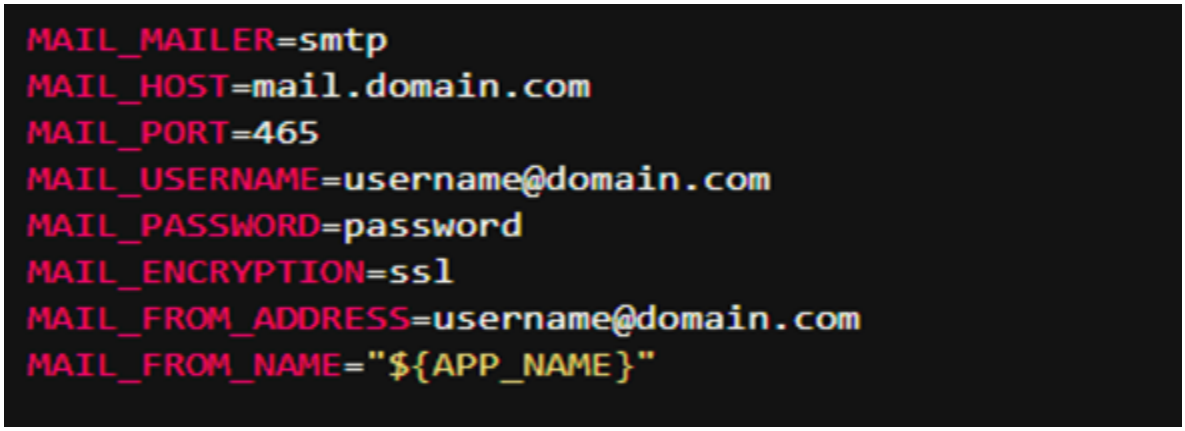
To obtain Firebase credentials, navigate to Project Settings > General > Web apps, then select your web app.

```
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "[redacted]",
  authDomain: "[redacted]",
  databaseURL: "[redacted]",
  projectId: "[redacted]",
  storageBucket: "[redacted]",
  messagingSenderId: "[redacted]",
  appId: "[redacted]",
  measurementId: "[redacted]"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

Please Note: Please ensure to update your mail SMTP credentials in the .env file of your admin panel. Failure to do so will result in the mail functionality not working. Refer to the following snapshot for guidance.



```
MAIL_MAILER=smtp
MAIL_HOST=mail.domain.com
MAIL_PORT=465
MAIL_USERNAME=username@domain.com
MAIL_PASSWORD=password
MAIL_ENCRYPTION=ssl
MAIL_FROM_ADDRESS=username@domain.com
MAIL_FROM_NAME="${APP_NAME}"
```

9.2.3 : Import the provided emart_admin_database.sql file to initialize the database.

9.2.4: Important: Run the following command from the root directory of your admin panel terminal to install Firebase dependency modules required for scheduler functions.

```
npm install
```

9.2.5: Important: Setup Node Js Path

Please update the NODE_PATH variable value in the .env file of your admin panel directory.

```
NODE_PATH=/home/user/bin/node
```

To get the Node path from the terminal, you typically want to determine the location of Node.js or a specific file related to Node. Here are common ways to do this from the terminal:

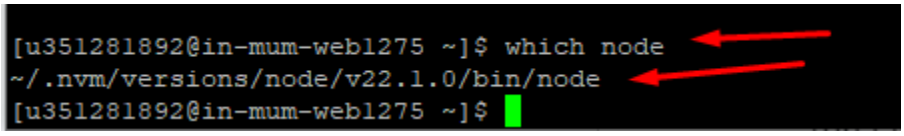
To find out where Node.js is installed on your system, you can use the which command (on Linux/macOS):

```
which node
```

This will return the full path to the Node.js binary. For example:

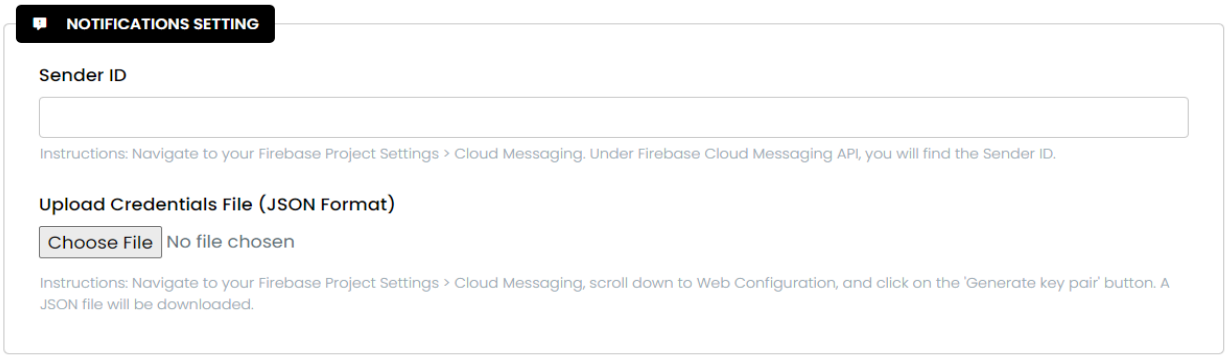
`/usr/local/bin/node`

See this example: In our case, the Node.js path is something like the one below.



Copy this path and update it in your NODE_PATH variable.

9.2.6 : Important: To work firebase push notification feature functions correctly, please upload your Firebase credentials file in the Admin Panel by navigating to Settings > Global Settings > Notifications Settings.



Now the admin panel is ready to run.

10. Website Panel Setup

10.1 : Once you've downloaded the code, upload the web panel zip file to your server and extract it. You can install the web panel on your main domain.

10.2 : Now you'll need to configure the .env file with your Database host, Database name, Database username, Database password, and your Firebase project credentials.

10.3 : Import the provided emart_website_database.sql file to import the database.

10.4 : Now the website panel is ready to run.

11. Store Panel Setup

11.1 : Once you've downloaded the code, upload the store panel zip file to your server and extract it. You can install the web panel on your main domain or subdomain.

11.2 : Now you'll need to configure the .env file with your Database host, Database name, Database username, Database password, and your Firebase project credentials.

11.3 : Import the provided emart_store_database.sql file to import the database.

11.4 : Now the store panel is ready to run.

12. Setup CRON JOB

A cron job is a scheduled task in Unix-based operating systems that runs scripts or commands at specified intervals. Here's how to set up a cron job on a server:

Steps to Set Up a Cron Job:

Step 1: Log in to cPanel or in your web server

Step 2: Navigate to the "Cron Jobs" Section
Step 3: Add a New Cron Job

In the **Command** field, enter the command or the path to the script you want to run.

For example: Run a script every minutes:

```
* * * * * /usr/local/bin/php /home/username/public_html/artisan schedule:run >> /dev/null 2>&1
```

For example: Run a script every day at midnight:

```
0 0 * * * /usr/local/bin/php /home/username/public_html/artisan schedule:run >> /dev/null 2>&1
```

Step 4: Save the Cron Job

Notes: Ensure the script you're running has the correct permissions and is executable. You need to specify the correct PHP path (/usr/local/bin/php is common but may vary depending on your server setup) and please use your admin panel path in the CRON JOB command.

Thank You