

GoRide Web Documentation

1. Introduction

The online cab booking service providers care the price of traveling based on the distance of travel trip and type of car, traffic, and waiting prices. There are numerous apps available in the android play store and apple app store for cab booking in India. Selecting the greatest taxi booking apps in India is tough, which may vary from town to town.

1.1 Basic knowledge required to setup

1.1.1 : Real server Server related knowledge like apache or local machine server, we preferred to use a real server.

1.1.2 : Server related knowledge and we preferred cPanel in your server for quick installation.

1.1.3 : Basic knowledge in PHP, Laravel and Flutter if you want to do some customization yourself (Not compulsory).

1.1.4 : Basic knowledge about google cloud and firebase.

2. Server Requirement

2.1 : Laravel 10.x requires a minimum PHP version of 8.1

2.2 : Support for MySQL v5.7

2.3 : Apache Server (Recommended)

2.4 : Mod_rewrite Apache

2.5 : PDO Extension and other required modules of PHP

2.6 : Firebase Account

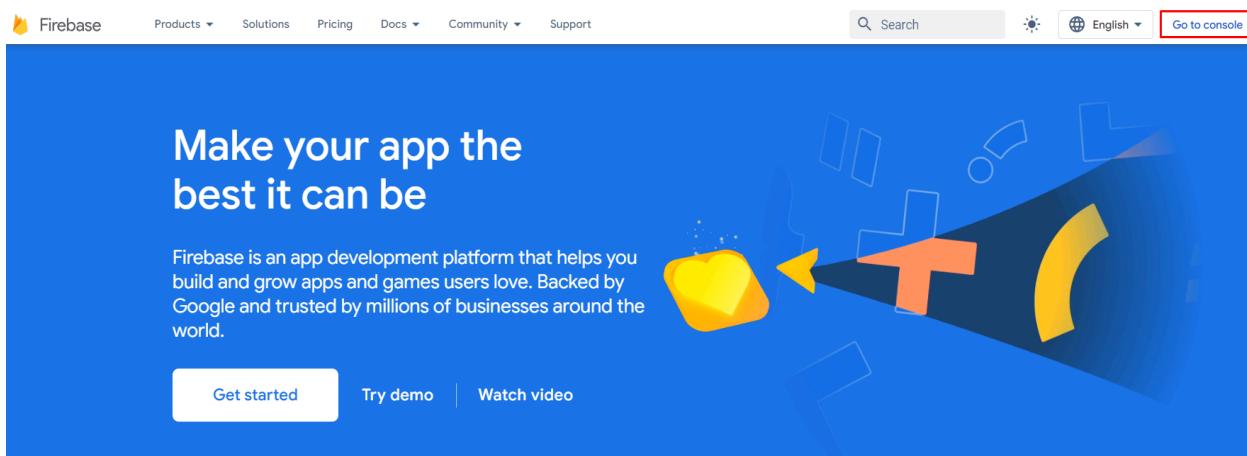
2.7 : Firebase Database

2.8 : Google MAP API Key

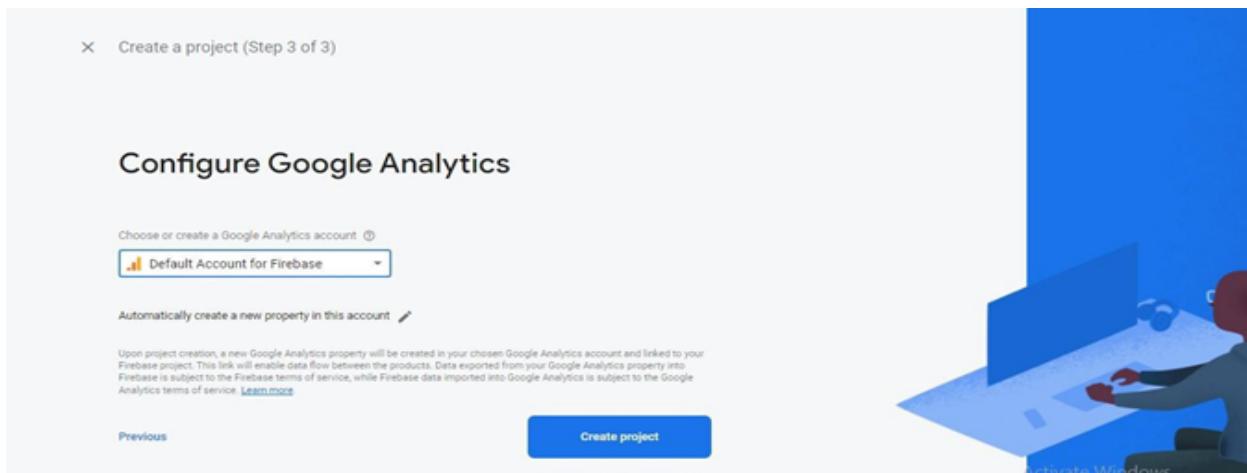
3. Create Firebase Project

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

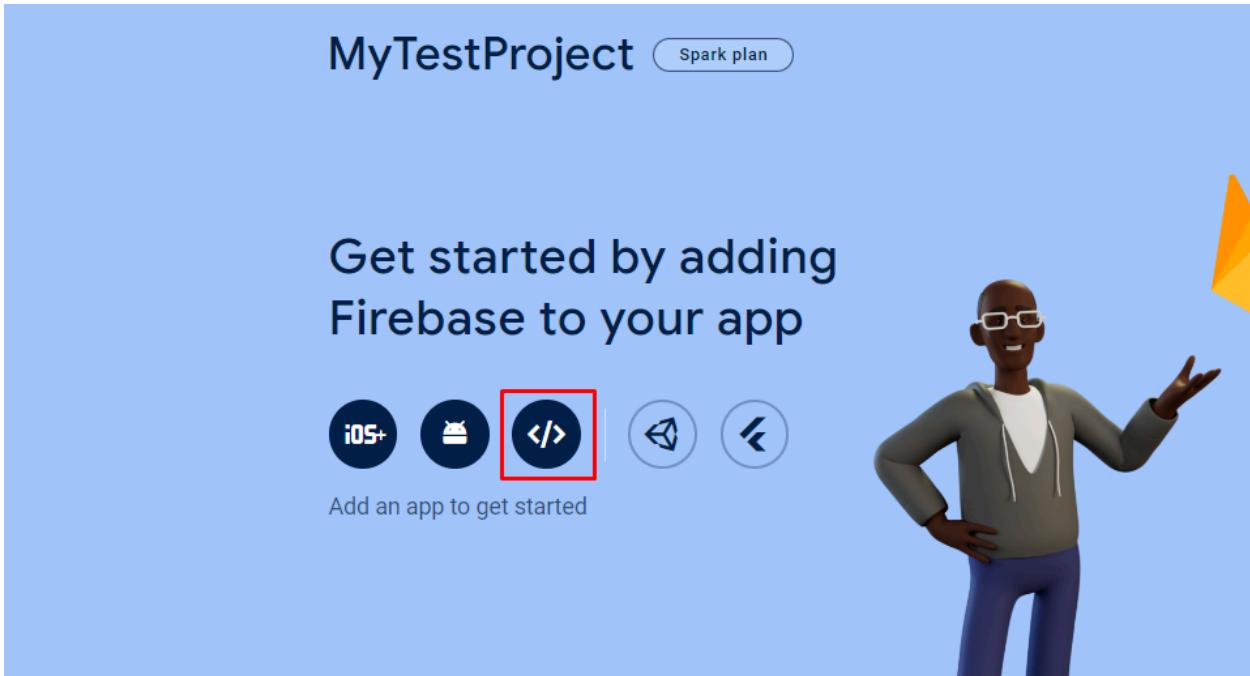
3.2 : Click on “Go to console” in the top right corner.



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: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  authDomain: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  projectId: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  storageBucket: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  messagingSenderId: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  appId: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  measurementId: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
};

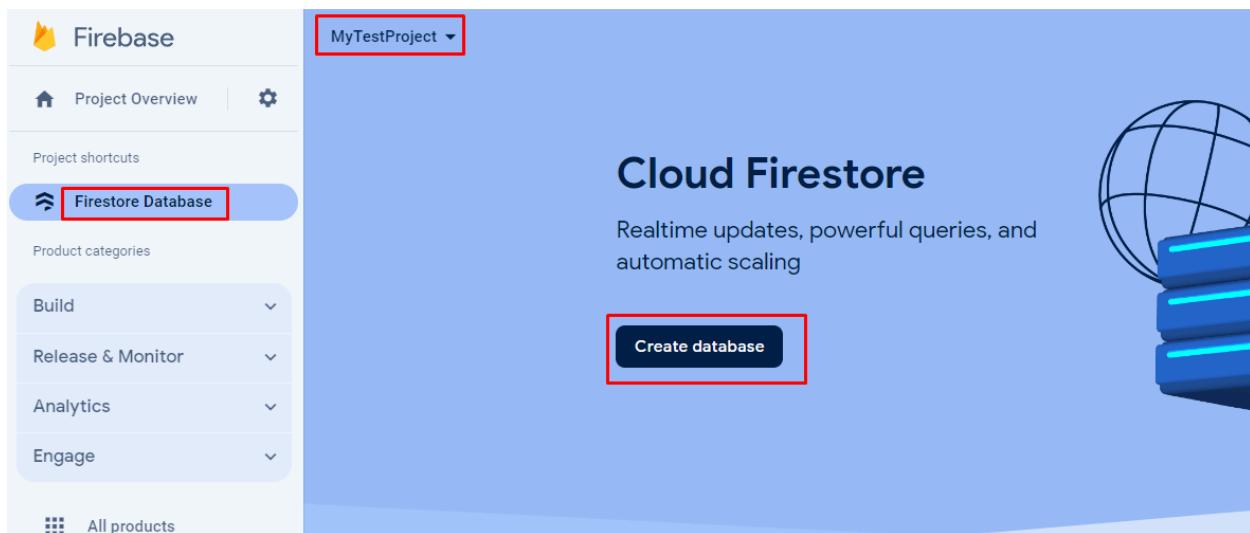
// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);

```



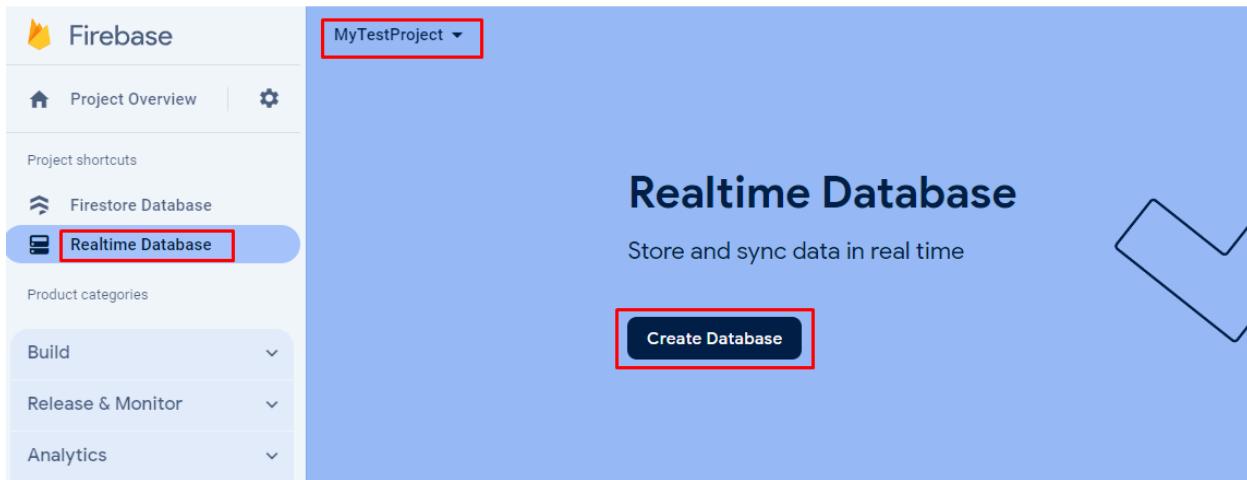
All of these details will be incorporated into both the app and admin panel documentation for comprehensive reference and seamless integration.

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'.



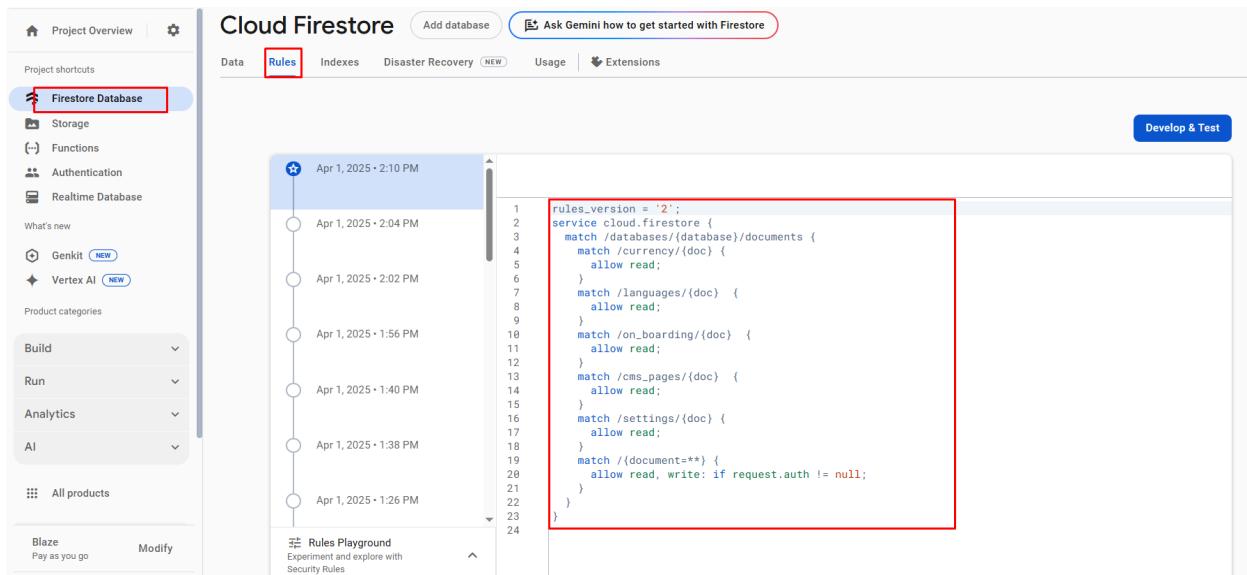
3.6 : Choose your preferred option and click on the 'Next' button. Subsequently, proceed by clicking on "Enable"

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.

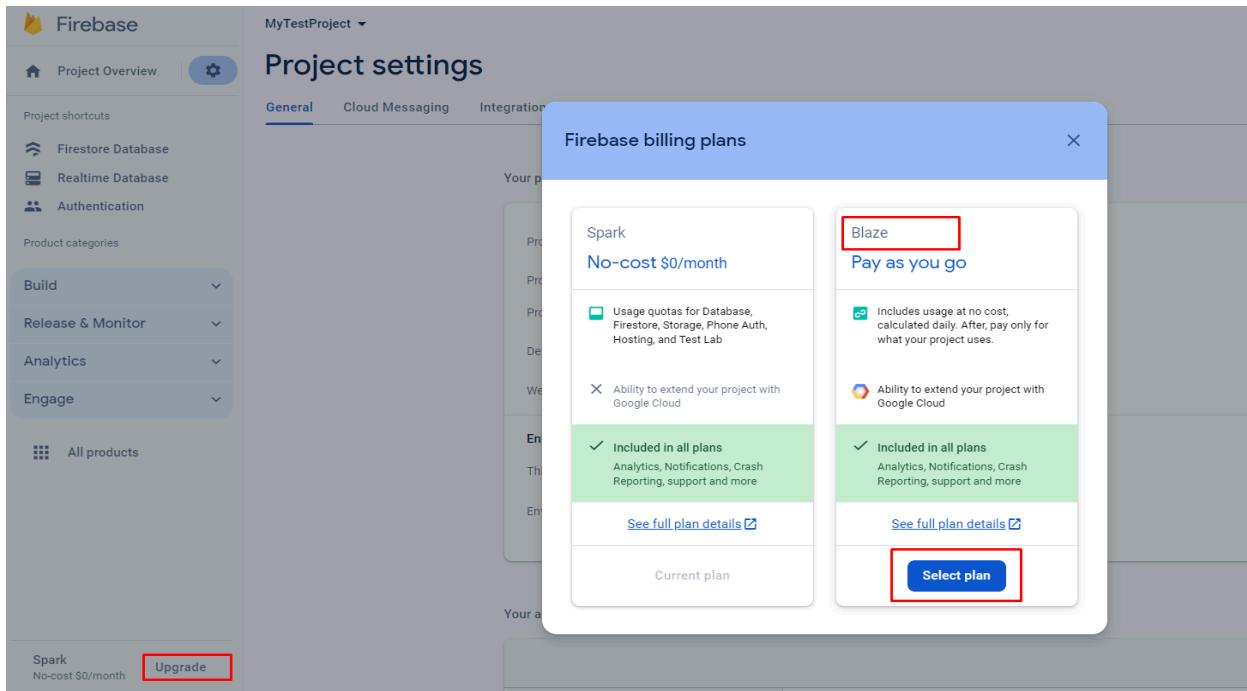
3.9 : Firestore Database Rules Update.



Important: Please update the following rules in your firestore database.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /currency/{doc} {
      allow read;
    }
    match /languages/{doc} {
      allow read;
    }
    match /on_boarding/{doc} {
      allow read;
    }
    match /cms_pages/{doc} {
      allow read;
    }
    match /settings/{doc} {
      allow read;
    }
    match /{document=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```

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-firebase-import-export firestore-import -a  
credentials.json -b collections.json
```

To export all collections, execute the following command:

```
npx -p node-firebase-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. 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)

5.1 : 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.

5.2 : 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
```

5.3 : 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.

5.4 : 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?](#)

6. Firestore Database Indexing

To perform Firebase indexing, follow these straightforward steps:

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

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

6.3 : 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.

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

6.5 : Execute the command `firebase init`

6.6 : Proceed with Y and press the enter button.

6.7 : 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.

6.8 : Choose the Option > Use an existing project

6.9 : Choose your project

6.10 : Press Enter > ? What file should be used for Firestore Rules? `firebase.rules`

6.11 : Press Enter > ? What file should be used for Firestore indexes? (`firebase.indexes.json`)

6.12: Now, the `firebase.indexes.json` file will be downloaded. Open this file and copy all the code from `firebase_indexes.json` file, then paste it into `firebase.indexes.json` file.

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

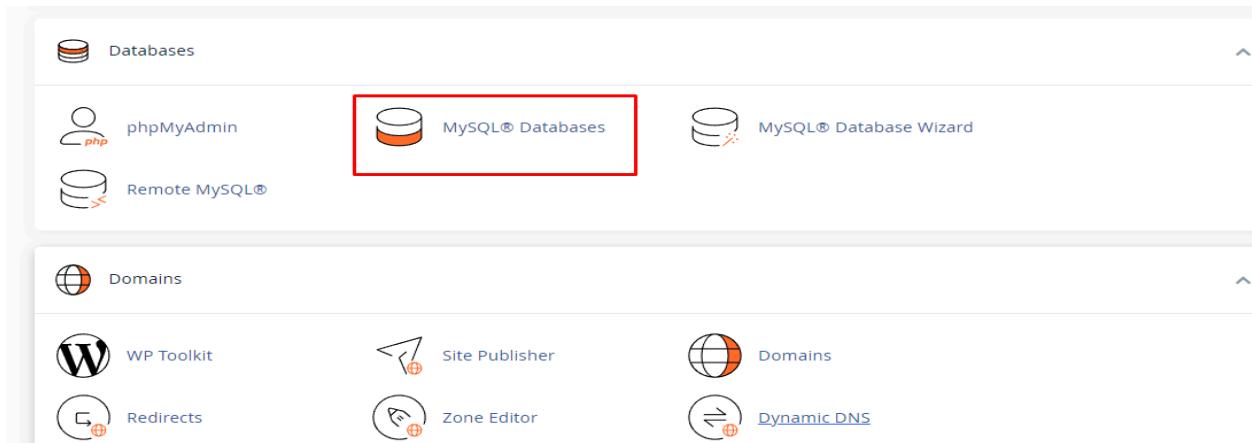
See Video: [How to Import Firestore Database Indexing in Firebase?](#)

7. 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.

7.1 Create Database

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



7.1.2 : Create New Database

A screenshot of the 'MySQL® Databases' page in cPanel. The title is 'MySQL® Databases'. Below it, a sub-section titled 'Create New Database' is shown. It has a 'New Database:' label and a text input field containing 'democom_'. Below the input field is a blue 'Create Database' button.

7.1.3 : Create a new database user

A screenshot of the 'MySQL Users' page in cPanel. The title is 'MySQL Users'. Below it, a sub-section titled 'Add New User' is shown. It has fields for 'Username' (containing 'democom_'), 'Password' (empty), 'Password (Again)' (empty), and 'Strength' (labeled 'Very Weak (0/100)'). There is also a 'Password Generator' button. At the bottom is a blue 'Create User' button.

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

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

ALL PRIVILEGES	
<input type="checkbox"/> ALTER	<input type="checkbox"/> ALTER ROUTINE
<input type="checkbox"/> CREATE	<input type="checkbox"/> CREATE ROUTINE
<input type="checkbox"/> CREATE TEMPORARY TABLES	<input type="checkbox"/> CREATE VIEW
<input type="checkbox"/> DELETE	<input type="checkbox"/> DROP
<input type="checkbox"/> EVENT	<input type="checkbox"/> EXECUTE
<input type="checkbox"/> INDEX	<input type="checkbox"/> INSERT
<input type="checkbox"/> LOCK TABLES	<input type="checkbox"/> REFERENCES
<input type="checkbox"/> SELECT	<input type="checkbox"/> SHOW VIEW
<input type="checkbox"/> TRIGGER	<input type="checkbox"/> UPDATE

Make Changes **Reset**

7.2 Configure Admin Panel

7.2.1 : 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.

7.2.2 : 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_MESSAGING_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.



```
// TODO: Add code 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",
  databaseURL: "REDACTED",
  projectId: "REDACTED",
  storageBucket: "REDACTED",
  messagingSenderId: "REDACTED",
  appId: "REDACTED",
  measurementId: "REDACTED"
};

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

7.2.3 : Import the provided `goride_admin_database.sql` file to initialize the database.

7.2.4: 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
```

7.2.5: 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:

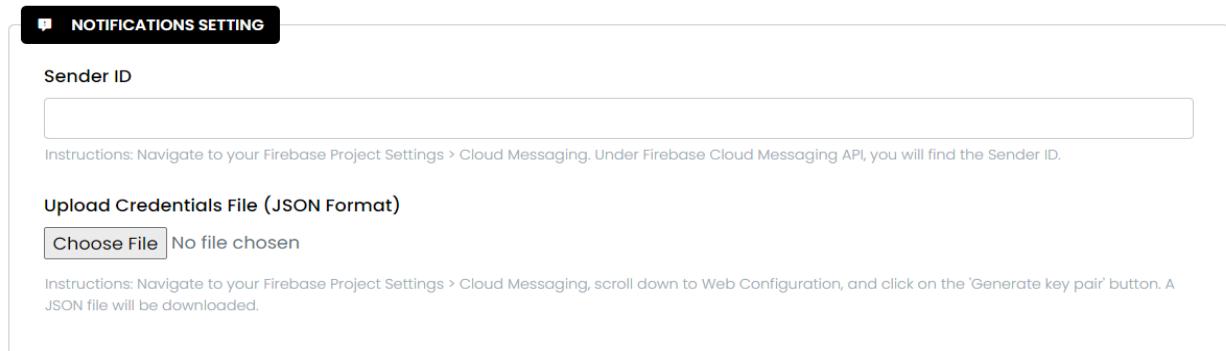
/usr/local/bin/node

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

```
[u351281892@in-mum-web1275 ~]$ which node
~/nvm/versions/node/v22.1.0/bin/node
[u351281892@in-mum-web1275 ~]$
```

Copy this path and update it in your NODE_PATH variable.

7.2.6 : 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.

8. Landing Page Setup

8.1. After downloading the source code, unzip `goride_source_code.zip` and upload the `goride_landing_panel.zip` file and extract to the root directory of your server.

Like: <http://landing.yourdomain.com/> or whatever which you like.
Please use subdomain of your main domain for landing page setup
(Recommended)

8.2. Update firebase credentials at .env file of landing page site.

Notes: Please use this same firebase credentials for landing page setup which you will add for admin panel setup on .env file.

Now you can access your landing page at your desired domain
For ex: <http://landing.yourdomain.com/>

9. Create Google Map API Key

9.1. Go to the [Google Maps Platform](#)

9.2. Click the Get Started button in the middle of the screen.

9.3. Click on the Google Cloud Platform home in the upper left corner.

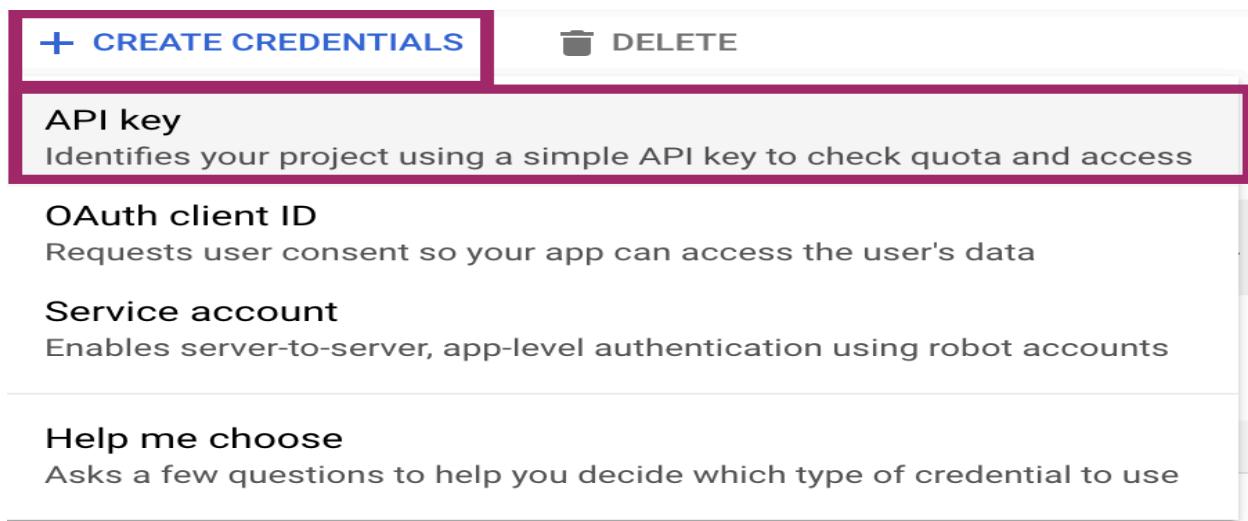
9.4. Click on Billing to make sure your billing details are up-to-date. If they are not, your Google Maps will not work properly.

9.5. Once you've confirmed your billing is up-to-date, click on the Google Cloud Platform home in the upper left corner again.

9.6. Hover to APIs & Services and go to Credentials.

9.7. If you want to use an existing project, please select it from the list. Otherwise, select 'Create a new project' and enter a project name.

9.8. Click Create credentials and select API key. You will see a new dialog that displays the newly created API key.



9.9. Click the Close button in the API key dialogue. Your new API key will be listed on the Credentials page under API keys

10. Admin Panel - Update

To update the admin panel just upload our latest source code zip file in your project root folder and extract it.

If you customized something on the code and want to update to our latest version then you can follow any one option from following this.

10.1. First push your code on a git branch then download our latest version source code from codecanyon and push it to another branch. And then you can merge the changes between current version and previous version code and merge both branches and it is possible to get conflicts on branches.

10.2. Download the our latest version source code from codecanyon and apply.

Thank You

© 2023-2025 GoRide. All Rights Reserved.