

Documentation for CabMe

Introduction

CabMe is a taxi/cab booking mobile application developed by using latest & secure technologies like Flutter (For Mobile App development) & Laravel /PHP Framework(For Admin Panel).

With the CabMe, we help startups, taxi/cab companies to expand their business & brand identity drastically. CabMe allows cab companies to manage their fleet operations in real time.

In CabMe, we provide an all-in-one solution for taxi booking services for Cab companies & Individual drivers. It includes: 2 Mobile Applications and 1 Admin Panel.

- 1. Customer taxi booking application for Android & iOS developed using Flutter.**
- 2. Driver application for Android & iOS developed using Flutter**
- 3. Super Admin web panel developed using Laravel.**

Guideline for basic installation

CabMe contains varieties of modules to set up web and mobile apps, so, to make successful connection & installation for your apps on your own then it's necessary to have a basic knowledge in both server side and mobile app development.

- 1. Real server server related knowledge like apache or local machine server, we preferred to use a real server.**
- 2. Server related knowledge and we preferred cPanel in your server for quick installation**
- 3. Basic knowledge in PHP, Laravel and Flutter if you want to do some customization yourself (Not compulsory).**
- 4. Basic knowledge about google cloud and firebase**

Server requirement

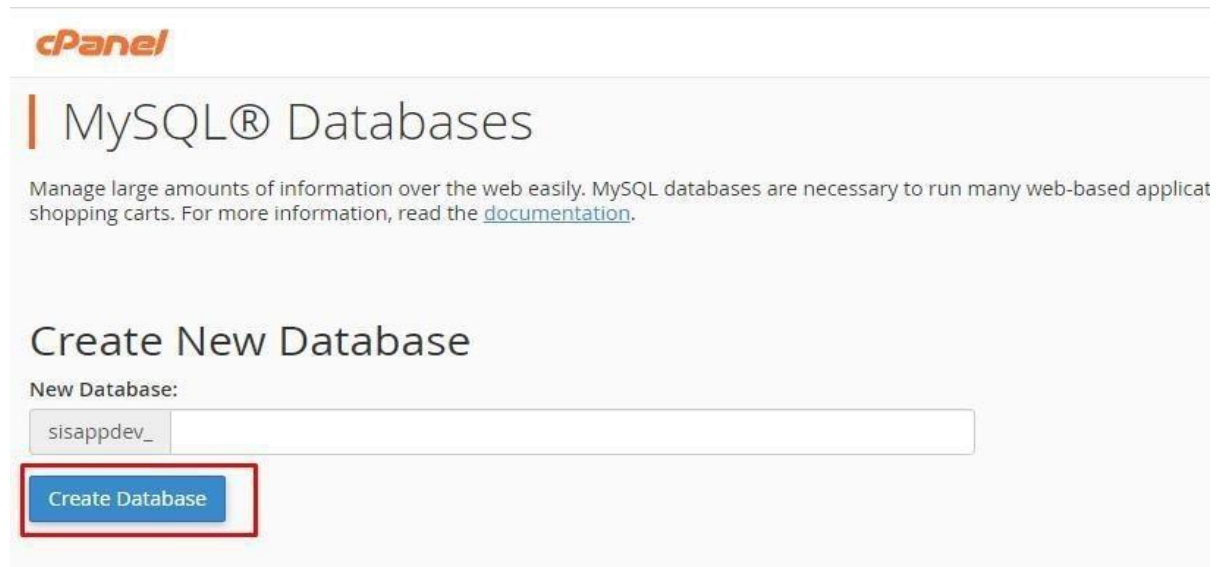
1. Laravel 10.x requires a minimum PHP version of 8.1
2. Apache Server (Recommended)
3. Mod_rewrite Apache
4. PDO Extension and other required modules of PHP

In most servers these extensions are enabled by default, but you should check with your hosting provider

Admin panel setup

Step 1: After downloading the code, extract the given zip file and upload the cabme_admin_panel.zip file in your server following your expected directory and extract the zip file. CabMe admin panel can be installed on a domain or subdomain: For Example, yourdomain.com OR admin.yourdomain.com OR yourdomain.com/admin

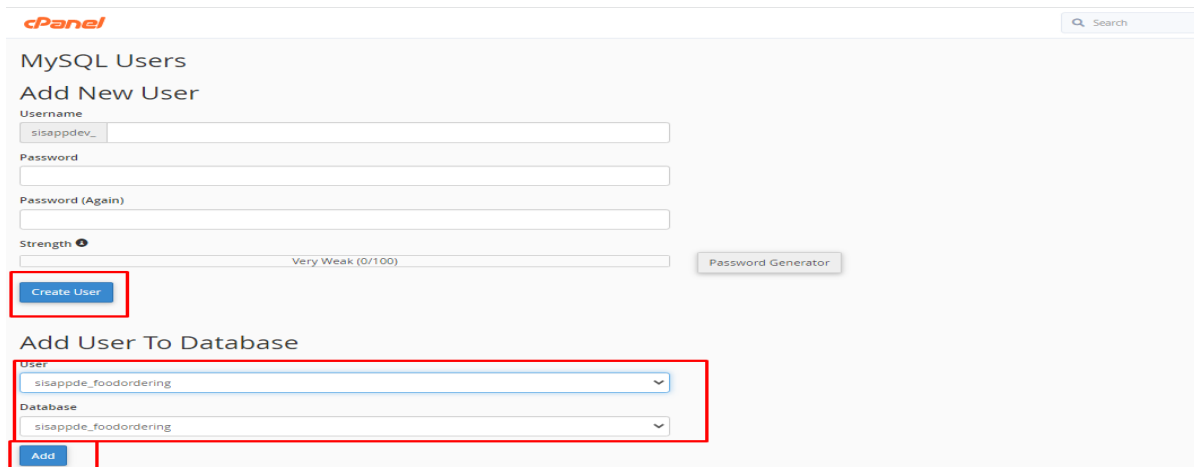
Step 2: Create the database properly



The image shows the cPanel interface for managing MySQL databases. At the top, the cPanel logo is visible. Below it, the heading "MySQL® Databases" is displayed. A descriptive paragraph states: "Manage large amounts of information over the web easily. MySQL databases are necessary to run many web-based applications, shopping carts. For more information, read the [documentation](#)." Below this, the section "Create New Database" is shown. It includes a label "New Database:" followed by a text input field containing the text "sisappdev_". Below the input field is a blue button labeled "Create Database", which is highlighted with a red rectangular border.

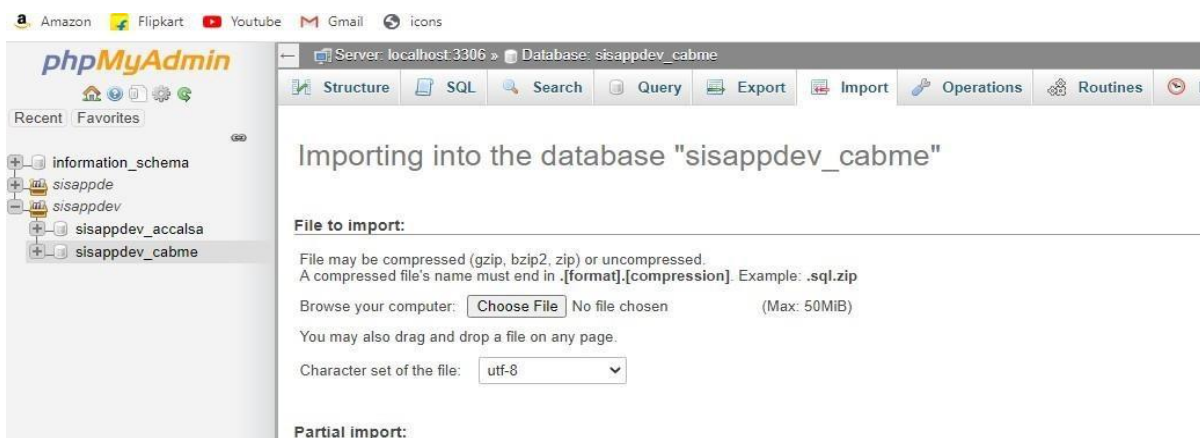
Create a new database from your server MYSQL database

Create a DB user to the database and link that database to the DB user



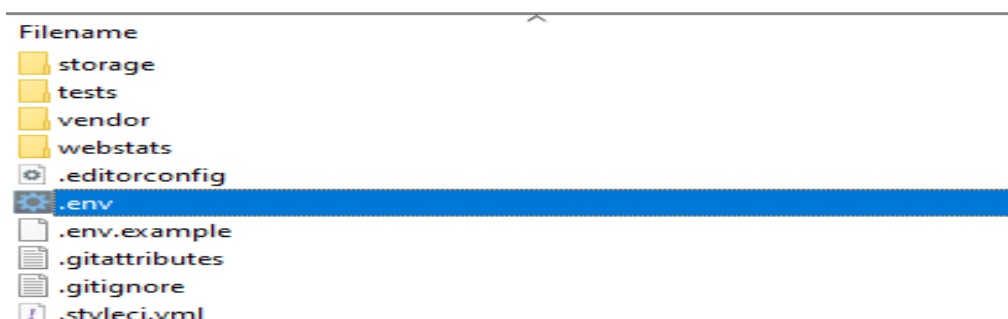
The image shows the cPanel MySQL Users interface. Under the 'Add New User' section, the 'Username' field is filled with 'sisappdev_'. The 'Password' and 'Password (Again)' fields are empty. The 'Strength' indicator shows 'Very Weak (0/100)'. A 'Create User' button is highlighted with a red box. Below this, the 'Add User To Database' section shows the 'User' dropdown set to 'sisappdev_foodordering' and the 'Database' dropdown set to 'sisappdev_foodordering'. An 'Add' button is also highlighted with a red box.

Step 3: Import cabme_admin_database.sql in to phpMyadmin of your server



The image shows the phpMyAdmin interface for the 'sisappdev_cabme' database. The 'Import' tab is selected. The 'File to import:' section shows a 'Choose File' button and a 'No file chosen' message. The 'Character set of the file:' dropdown is set to 'utf-8'. The 'Partial import:' section is empty.

Step 4: Update your database credentials (DB_DATABASE, DB_USERNAME & DB_PASSWORD) to .env file which exist at root path of admin panel



Now you can access your admin panel at the desired domain.

Dispatcher panel setup

For dispatcher panel setup, extract given zip file code and you just need to follow above step 4

Use the same database credentials of the admin for dispatcher panel setup.

Landing page setup

For landing page setup, extract given zip file code and you just need to follow above step 4

Use the same database credentials of the admin for landing page setup.

Important Notes: Please generate a new application key by running below command at your terminal (CLI) of admin panel root and use this application key to your apps, otherwise the admin panel, customer & driver application will not work.

Follow the same steps for dispatcher panel also.

“php artisan key:generate”

```
$ php artisan key:generate
Application key set successfully.
```

Now you can see generated application key at .env file of admin panel root path

```
> Users > Admin > AppData > Local > Temp > fz3temp-2 > .env 3
1 APP_NAME=Cabme
2 APP_ENV=local
3 APP_KEY=base64:9wMwQDEBjAK5OVvehRlQhF5PE1dNk6xK3RRIUkc
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
```

How to set up CabMe admin panel on your server?

See video: [▶ How to set up CabMe admin panel to your server?](#)

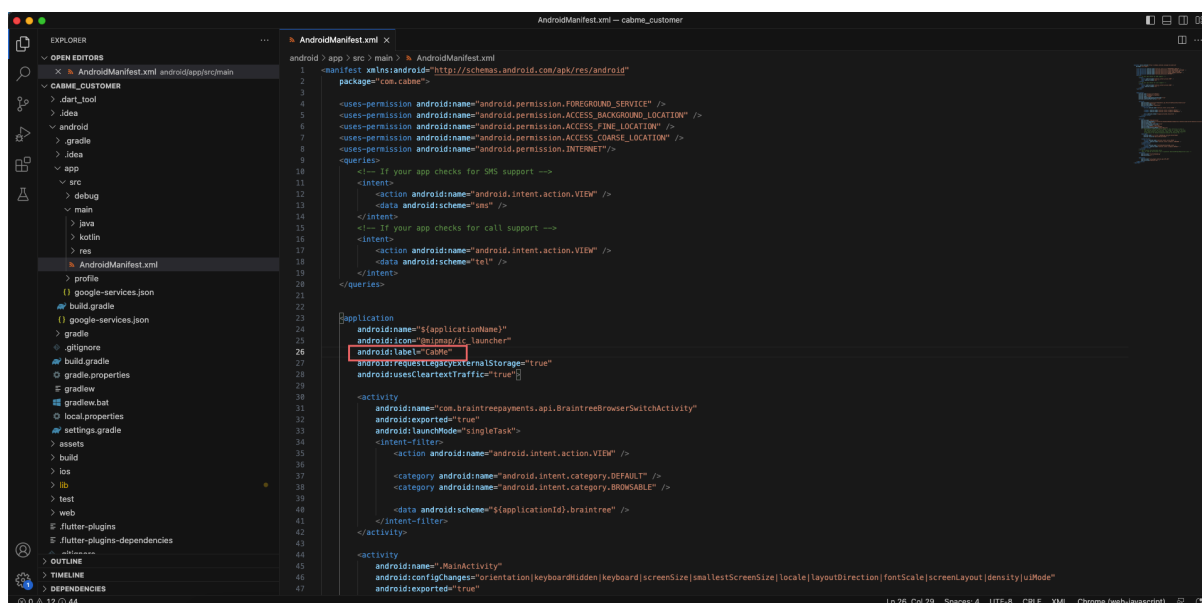
Flutter Application Setup

How to set up CabMe application

Change App Name **(For both application customer and driver follow same step)**

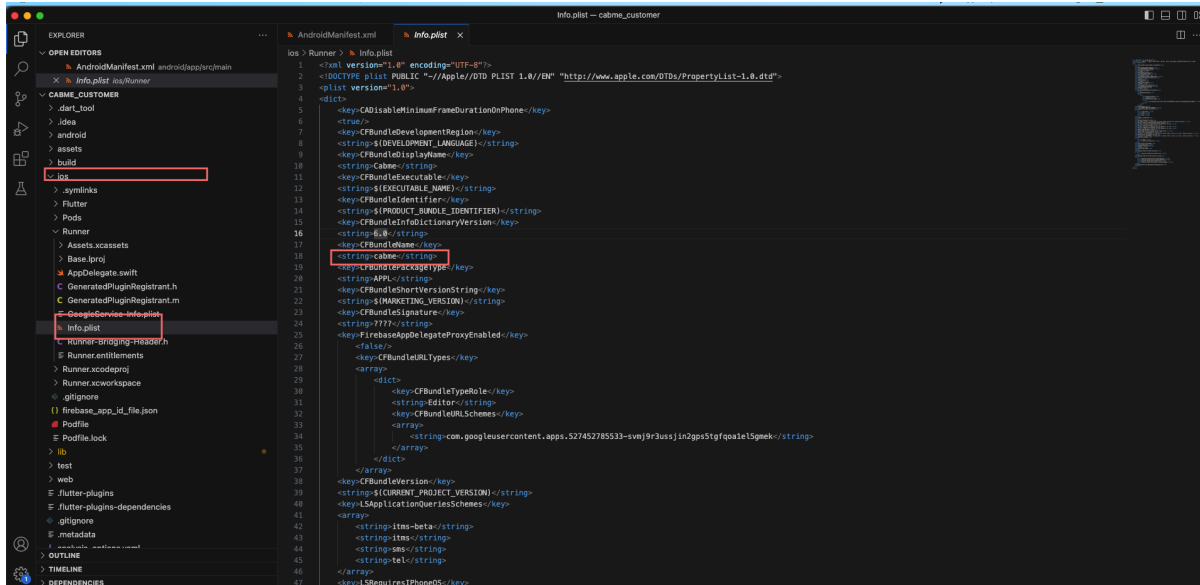
1. Change the value of label from

<project>/android/app/src/main/AndroidManifest.xml



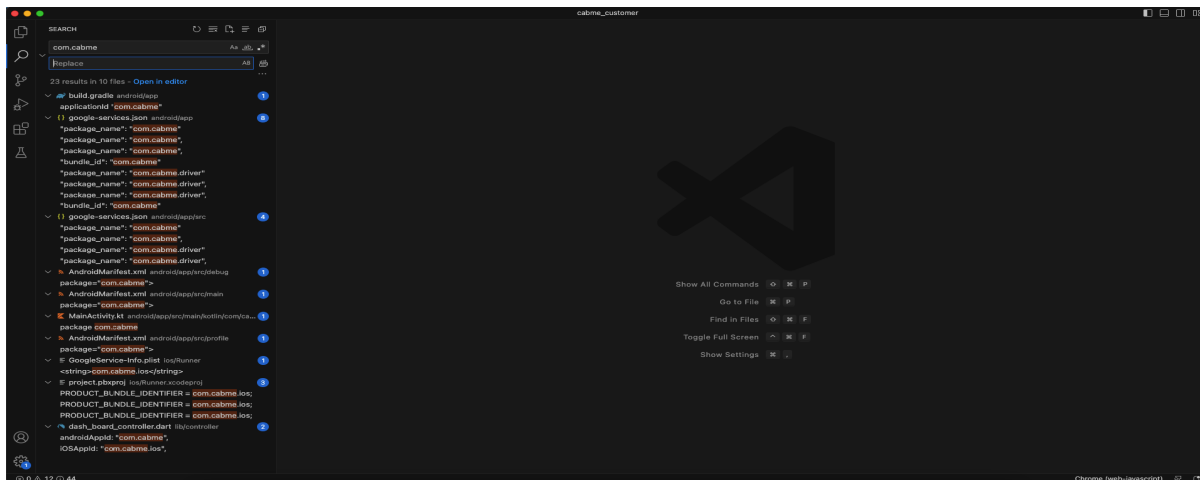
2. Change the value of CFBundleName from **(For both application customer and driver follow same step)**

<project>/iOS/Runner/info.plist



Change App Package (For both application customer and driver follow same step)

First you have to find out the existing package name. You can find it out from the top of the /app/src/main/AndroidManifest.xml file. Now on the right side of the project folder from VS Code. In the first box you have to put the existing package name that you saw in the AndroidManifest.xml file previously and write down your preferred package name in the second box and then click on Replace All button.



1. Android

The screenshot shows the AndroidManifest.xml file in an IDE. The left sidebar shows the project structure with 'AndroidManifest.xml' selected. The main editor displays the XML code for the manifest. A red box highlights the 'android:label' attribute in the application tag, which is set to 'com.google.android.get.AP'.

<project>/ios/Runner/AppDelegate.swift

The screenshot shows an IDE with the AppDelegate.swift file open. The file contains the following code:

```

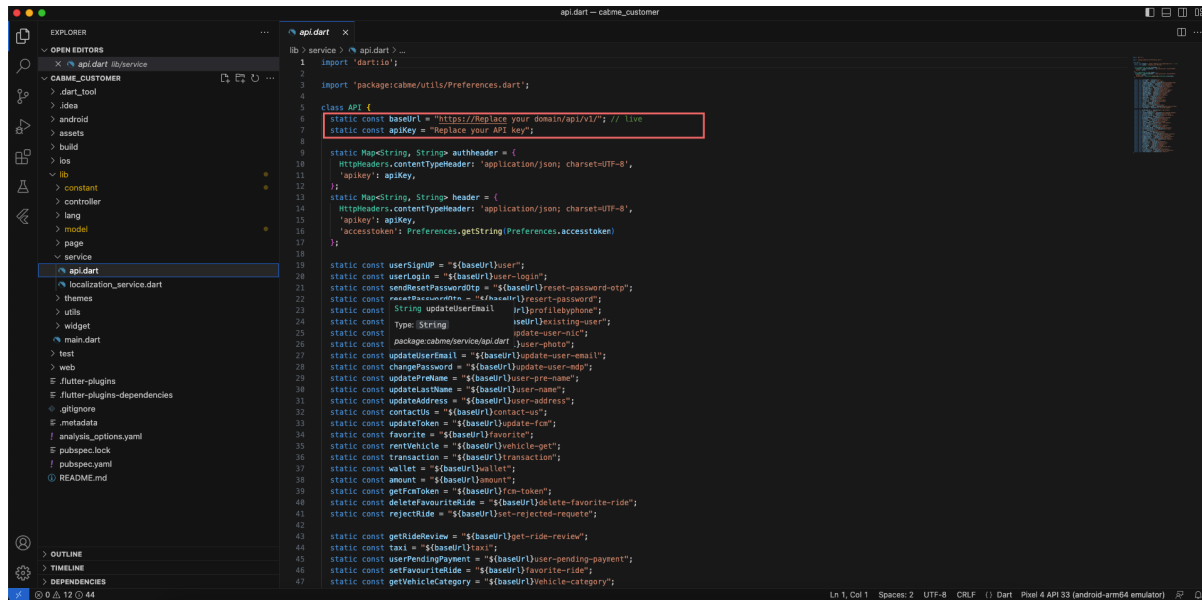
1  import UIKit
2  import Flutter
3  import GoogleMaps
4
5  @UIApplicationMain
6  @objc class AppDelegate: FlutterAppDelegate {
7      override func application(
8          _ application: UIApplication,
9          didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?
10         ) -> Bool {
11         OMServices.provideAPIKey("Replace Key")
12         GeneratedPluginRegistrant.register(with: self)
13         return super.application(application, didFinishLaunchingWithOptions: launchOptions)
14     }
15 }
16

```

The IDE interface includes a sidebar on the left with icons for Explorer, Search, Source Control, Run and Debug, and Extensions. The Explorer view shows the project structure, including the AppDelegate.swift file. The main editor area displays the code for AppDelegate.swift. The status bar at the bottom shows the current line and column (Ln 11, Col 44) and the active file (AppDelegate.swift).

Change URL and access token (For both application customer and driver follow same step)

Application Path: lib > service > api.dart



```
lib > service > api.dart
1 import 'dart:io';
2
3 import 'package:cabme/utils/Preferences.dart';
4
5 class API {
6   static const baseUrl = "https://replace your domain/api/v1/"; // Live
7   static const apiKey = "replace your API key";
8
9   static Map<String, String> authHeader = {
10     'headers': {'contentType': 'application/json'; charset=UTF-8',
11     'apiKey': apiKey,
12   };
13   static Map<String, String> header = {
14     'headers': {'contentType': 'application/json'; charset=UTF-8',
15     'apiKey': apiKey,
16     'accessToken': Preferences.getString(Preferences.accessToken);
17   };
18
19   static const userSignUp = "${baseUrl}user-signup";
20   static const userLogin = "${baseUrl}user-login";
21   static const sendResetPasswordOtp = "${baseUrl}reset-password-otp";
22   static const resetPasswordOtp = "${baseUrl}reset-password";
23   static const String updateUserEmail = "${baseUrl}update-user-email";
24   static const Type <String> userPhoto = "${baseUrl}update-user-photo";
25   static const package.cabme.service.dart update-user-nic = "${baseUrl}update-user-nic";
26   static const updateUserEmail = "${baseUrl}update-user-email";
27   static const updateToken = "${baseUrl}update-user-token";
28   static const changePassword = "${baseUrl}update-user-mp";
29   static const updateProfileName = "${baseUrl}update-user-pre-name";
30   static const updateLastName = "${baseUrl}update-user-name";
31   static const updateAddress = "${baseUrl}update-user-address";
32   static const contactUs = "${baseUrl}contact-us";
33   static const updateToken = "${baseUrl}update-fcm";
34   static const favorite = "${baseUrl}favorite";
35   static const rentVehicle = "${baseUrl}vehicle-get";
36   static const transaction = "${baseUrl}transaction";
37   static const wallet = "${baseUrl}wallet";
38   static const amount = "${baseUrl}amount";
39   static const getFcmToken = "${baseUrl}fcm-token";
40   static const deleteFavoriteRide = "${baseUrl}delete-favorite-ride";
41   static const rejectRide = "${baseUrl}reject-ride-request";
42
43   static const getRideReview = "${baseUrl}get-ride-review";
44   static const taxi = "${baseUrl}taxi";
45   static const userPendingPayment = "${baseUrl}user-pending-payment";
46   static const setFavoriteRide = "${baseUrl}favorite-ride";
47   static const getVehicleCategory = "${baseUrl}vehicle-category";
```

Setup with firebase

1. Setup with firebase using flutterFire.

a. Follow this link to setup with firebase :-

<https://firebase.google.com/docs/flutter/setup?platform=ios>

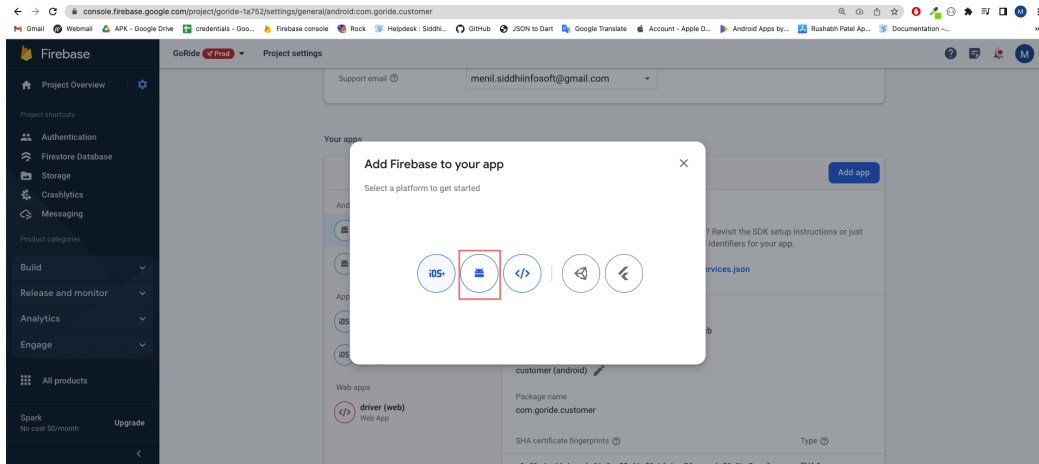
b. Video link :-

▶ Add Firebase to your Flutter app: The fast way

2. Useful link :- ▶ How to set up CabMe Flutter application?

3. Setup manually.

3(a) For Android



3(a) i. Enter package name

[illegible]

3(a) ii. Download google-services.json file.

1 Register app
Android package name: com.test

2 Download and then add config file

Download google-services.json

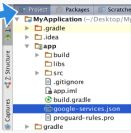
Switch to the Project view in Android Studio to see your project root directory.

Next

3 Add Firebase SDK


4 Next steps

Instructions for Android Studio below | [Watch](#) [Code](#) [Help](#)



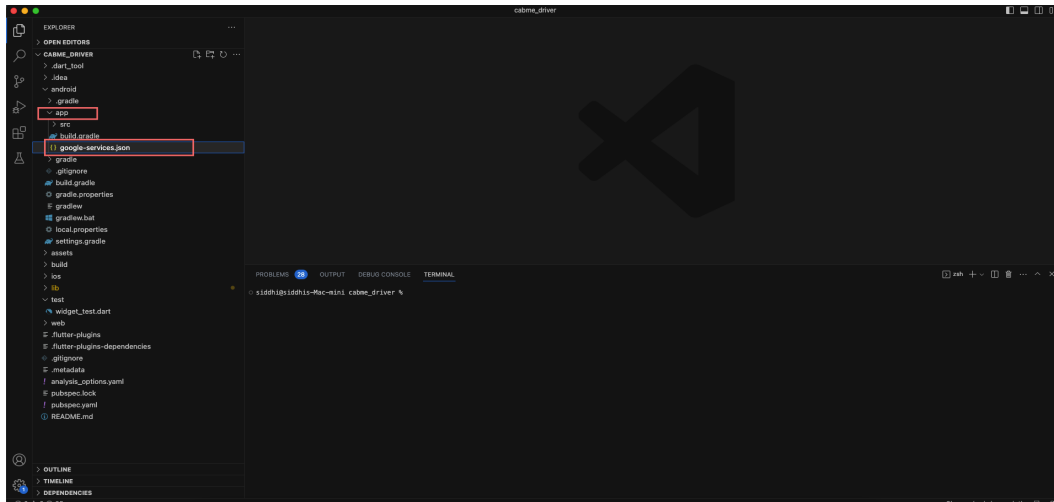
The screenshot shows the Android Studio interface with the Project view selected. The project structure is visible, showing the 'app' module under 'My Applications'. The 'google-services.json' file is highlighted in the 'app' module's file list.

Go to docs

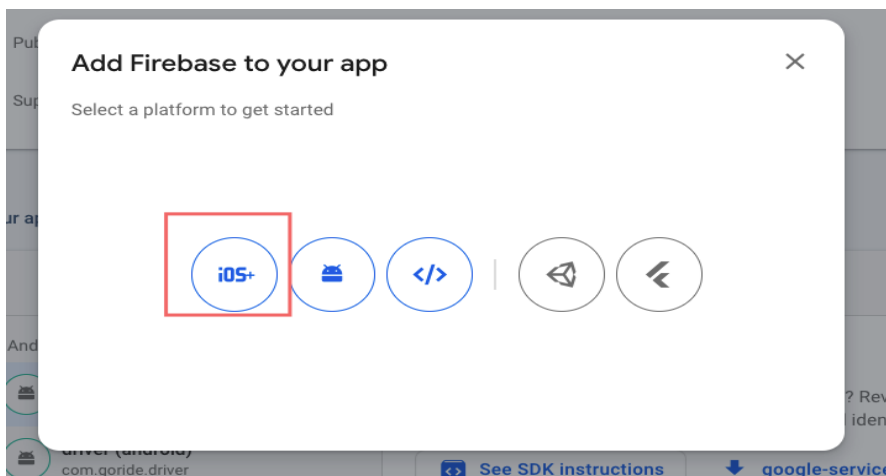


The illustration shows a blue smartphone at the top right and a laptop at the bottom right. A white cable connects the two devices. The laptop screen displays a white document icon with a blue circular logo featuring a white 'd'.

3(a) iii. Place google-services.json file in <project>/android/app/



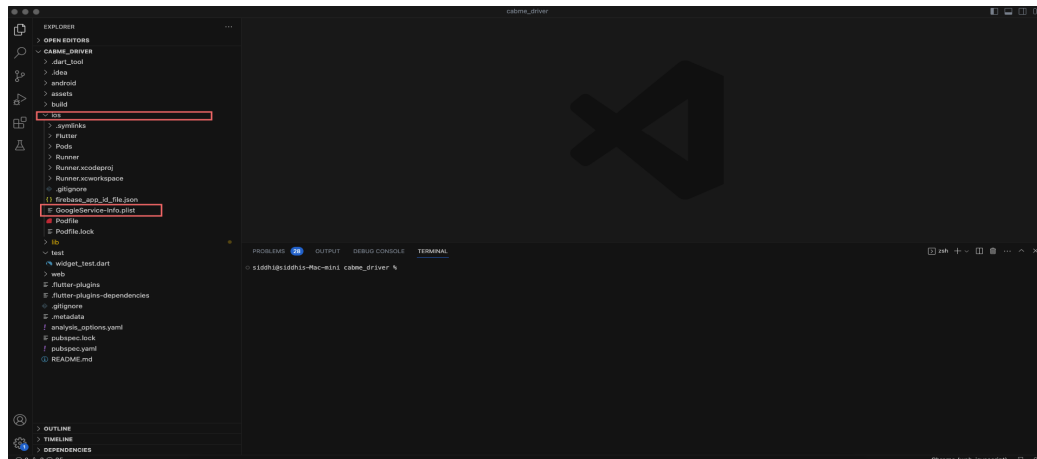
3(b) i. For IOS - Add App



3(b) ii. Enter package name and download GoogleService-Info.plist

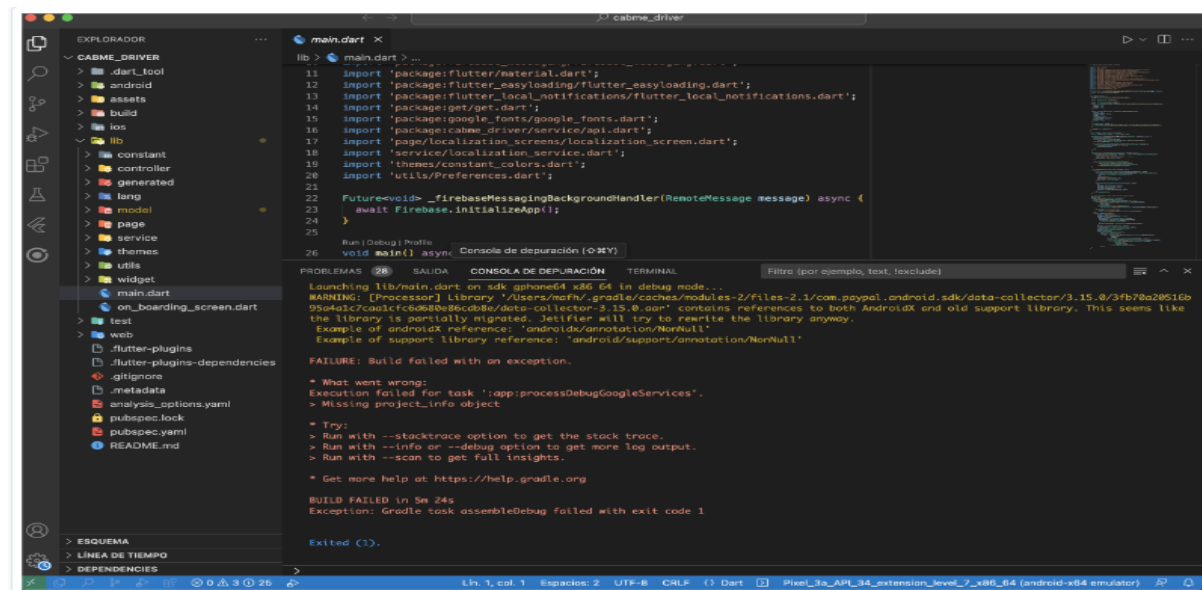
A screenshot of the 'Register app' form in the Firebase console. The form has a title '1 Register app'. It contains three input fields: 'Apple bundle ID' with the value 'com.company.appname' (highlighted with a red rectangle), 'App nickname (optional)' with the value 'My Apple app', and 'App Store ID (optional)' with the value '123456789'. At the bottom, there is a 'Register app' button.

3(b) iii. Place GoogleService-Info.plist in <project>/iOS/



IMPORTANT NOTES

If you will face the below attached error while setting up the project, make sure you have downloaded google-service.json from the firebase and replace the original one. If you do not replace google-service.json then you will face the below image issue. Although, if you will face any kind of issue related to this, feel free to raise a support ticket to <https://support.siddhiinfosoft.com>



Thank You