

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

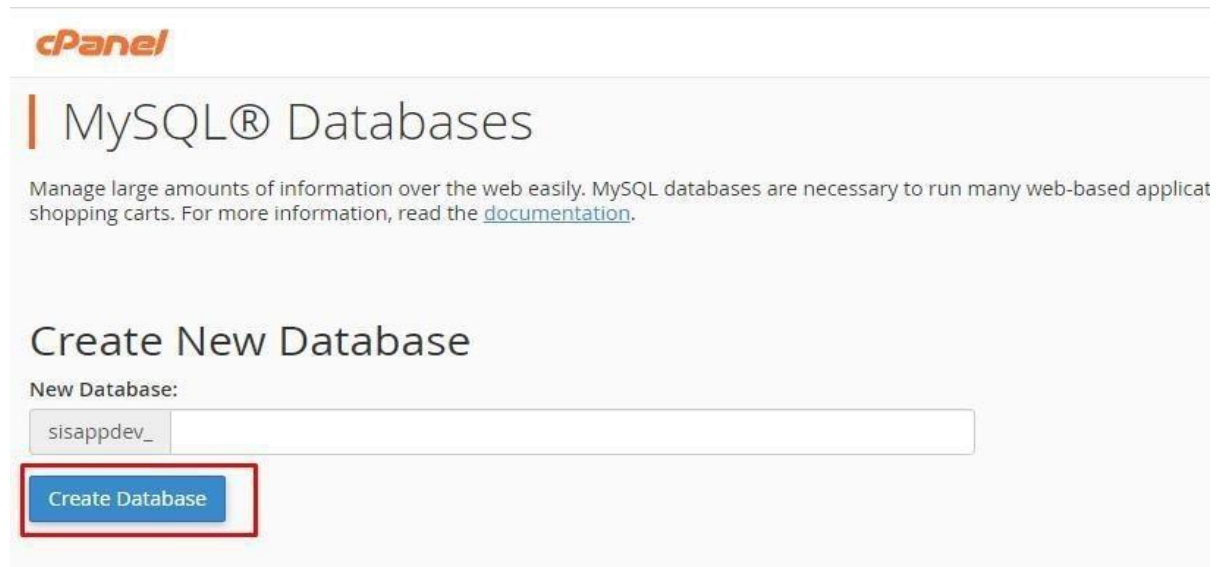
1. Laravel 10.x requires a minimum PHP version of 8.1
2. Apache Server (Recommended)
3. Mod_rewrite Apache
4. PDO Extension and other required modules of PHP

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

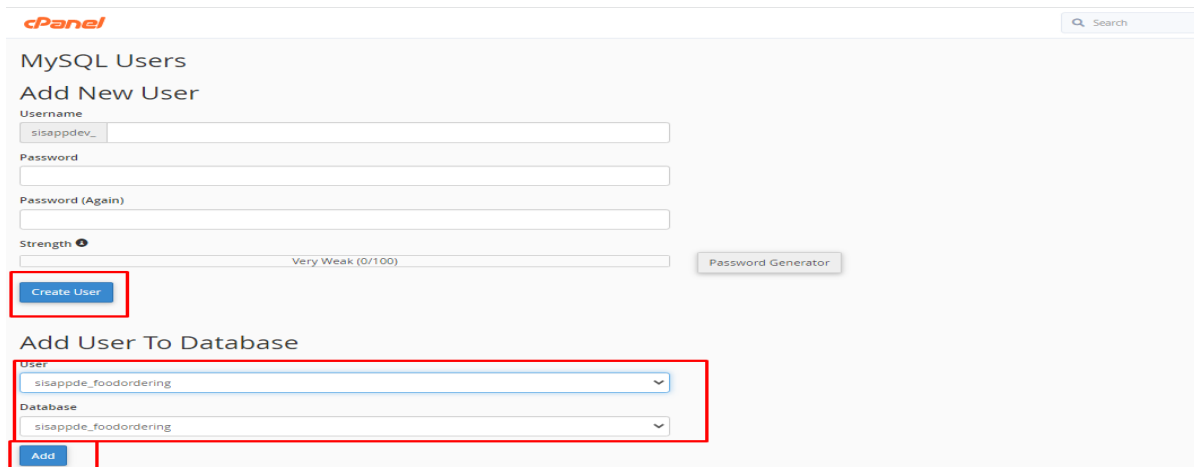
Step 2: Create the database properly



The screenshot shows the cPanel interface for MySQL Databases. At the top, there's a cPanel logo and the title 'MySQL® Databases'. Below the title, a brief description 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](#).' The main section is titled 'Create New Database'. Underneath, there's a label 'New Database:' followed by a text input field containing 'sisappdev_'. Below the input field, there is a blue button labeled 'Create Database', which is highlighted with a red rectangular border.

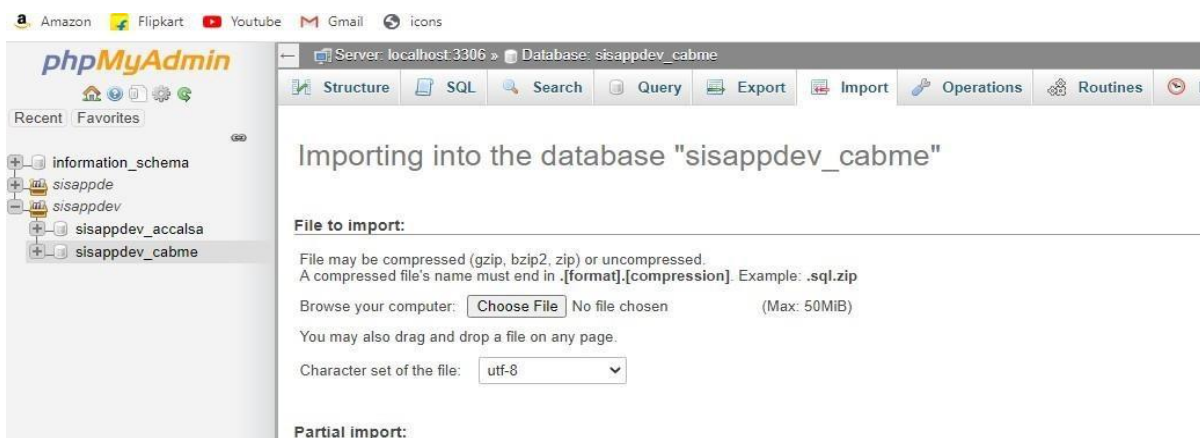
Create a new database from your server MYSQL database

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



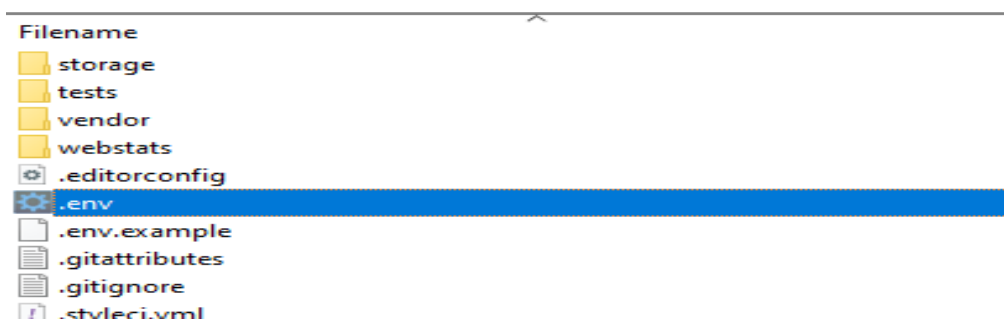
The image shows the cPanel MySQL Users interface. Under the 'Add New User' section, the 'Username' field is filled with 'sisappdev_'. The 'Password' and 'Password (Again)' fields are empty. The 'Strength' indicator shows 'Very Weak (0/100)'. A 'Create User' button is highlighted with a red box. Below 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. The left sidebar shows a tree view of databases, with 'sisappdev_cabme' selected. The main panel displays the 'Import' tab. The title is 'Importing into the database "sisappdev_cabme"'. Under 'File to import:', there is a text area for the file path, a 'Choose File' button, and a 'No file chosen' message. Below this, there is a 'Character set of the file:' dropdown menu set to 'utf-8'. A 'Partial import:' section is visible at the bottom.

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

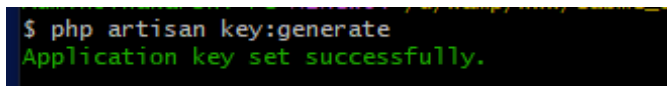
For landing page setup, extract given zip file code and you just need to follow above step 4

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 the dispatcher panel also.

“php artisan key:generate”



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

If you're facing any errors while creating a key then please run the following command to update all system packages.

“composer update”

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

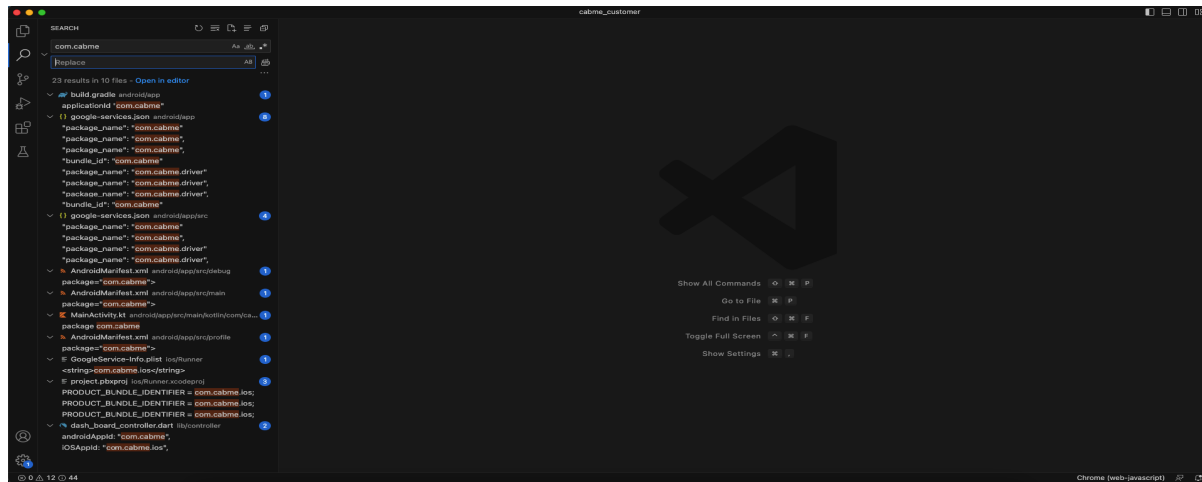
How to set up CabMe application

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

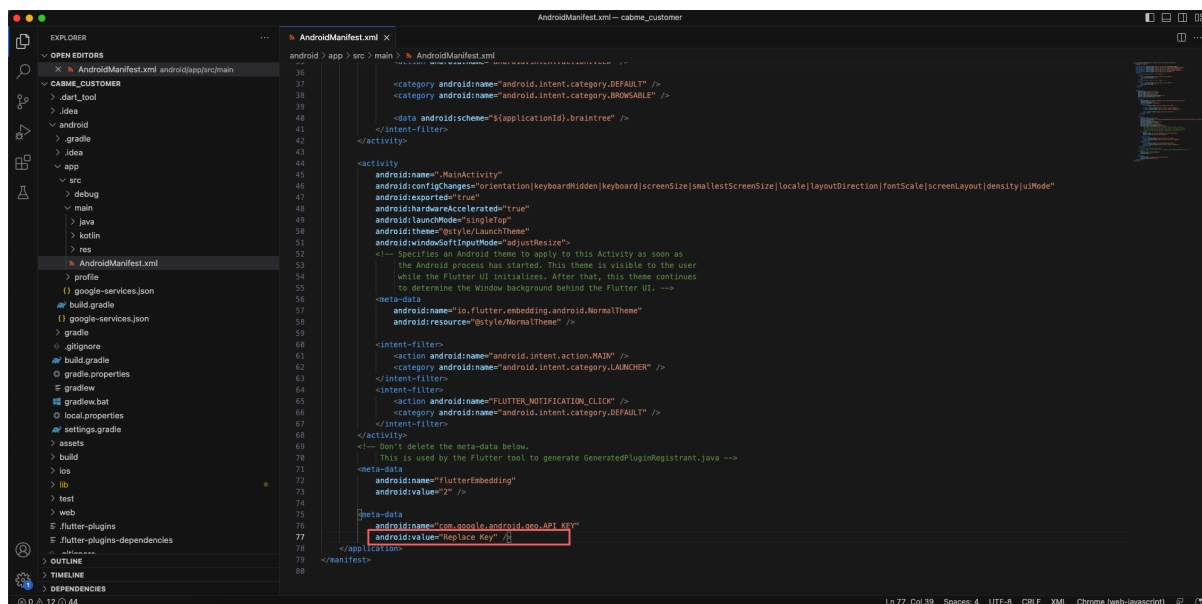
First you have to find out the existing package name. You can find it out from the top of the `/app/src/main/AndroidManifest.xml` file. Now on the right side of the project folder from VSCode. In the first box you have to put the existing package name that you saw in the `AndroidManifest.xml` file previously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



Replace Map key

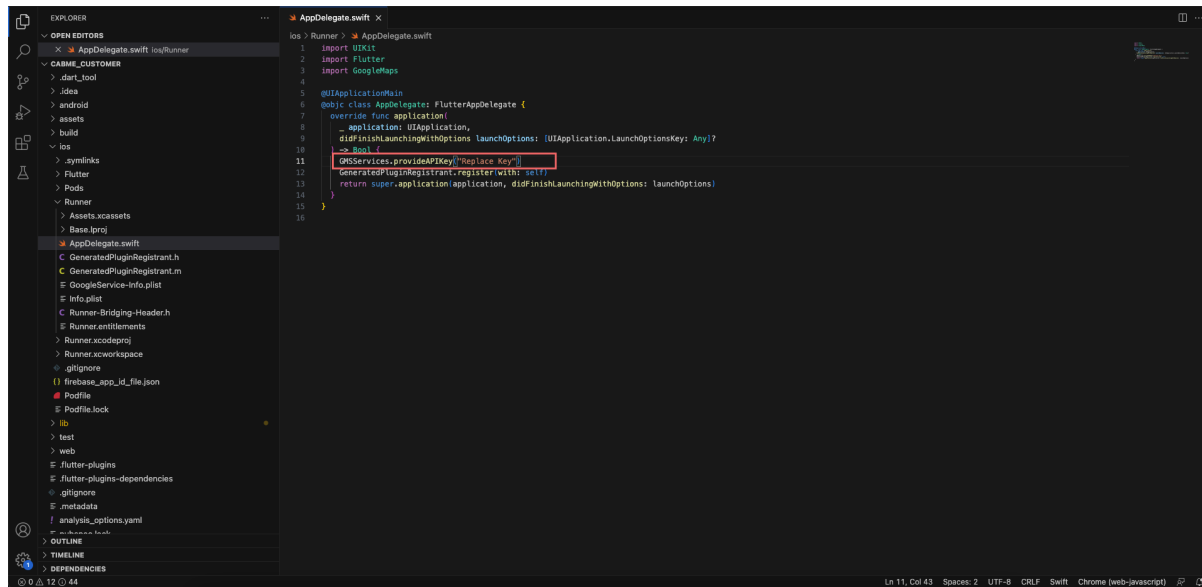
1. Android

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



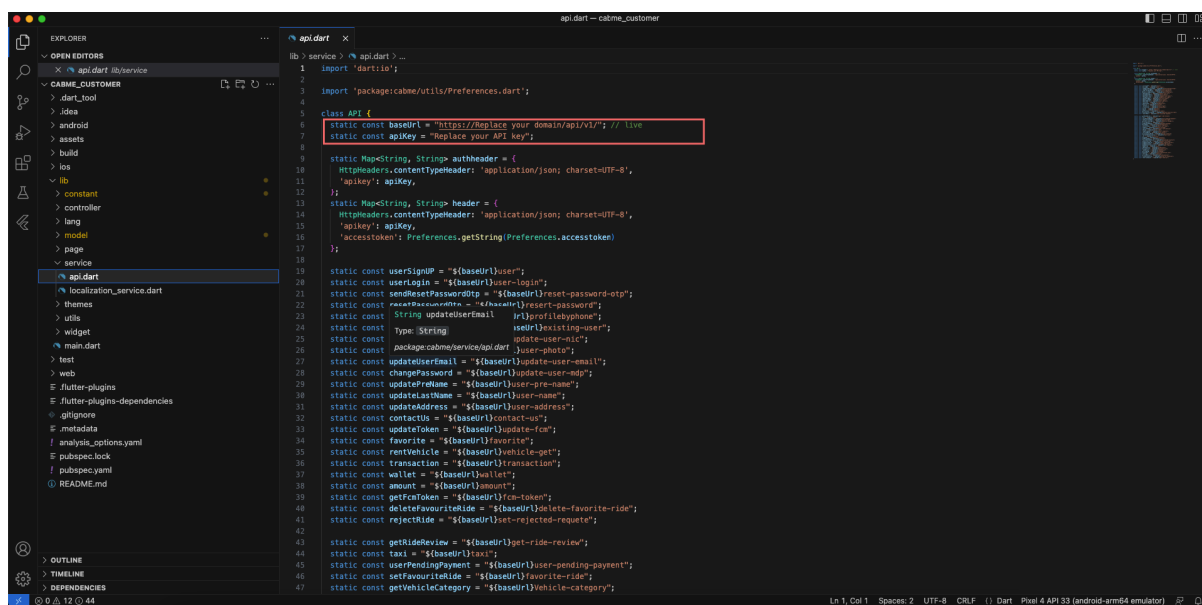
2. IOS

<project>/ios/Runner/AppDelegate.swift



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

Application Path: lib > service > api.dart



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>

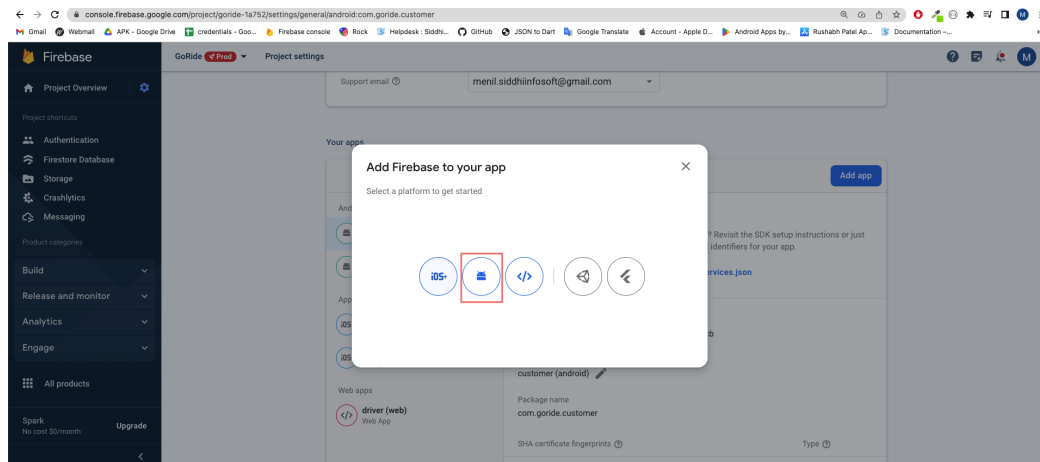
b. Video link :-

 **Add Firebase to your Flutter app: The fast way**

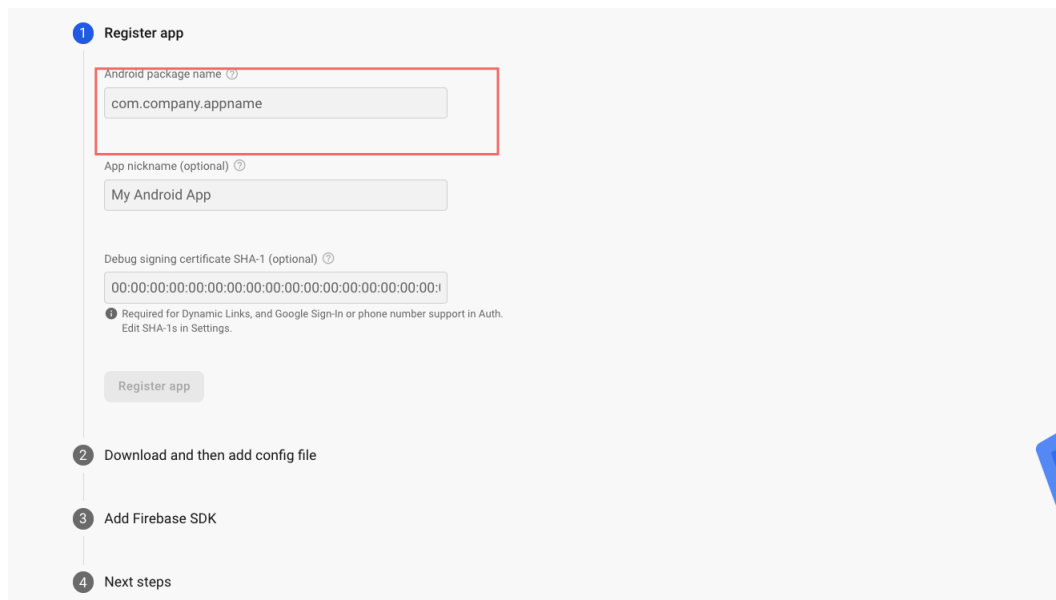
2. Useful link :- **How to set up CabMe Flutter application?**

3. Setup manually.

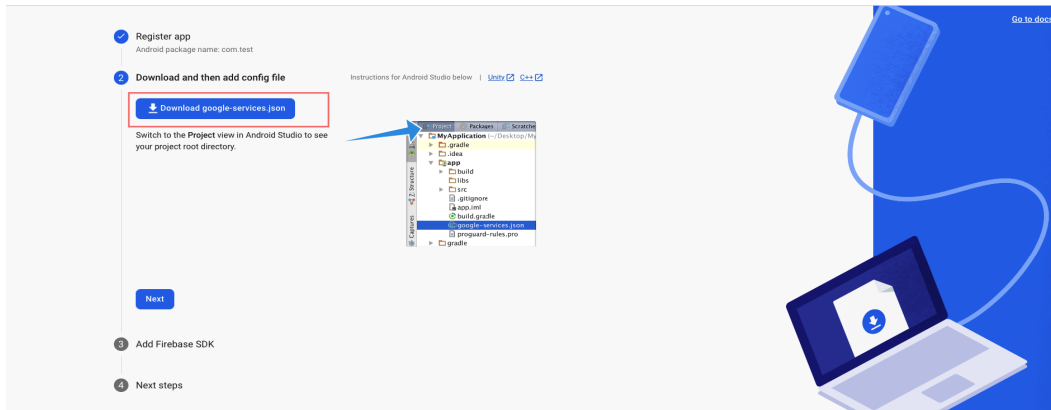
3(a) For Android



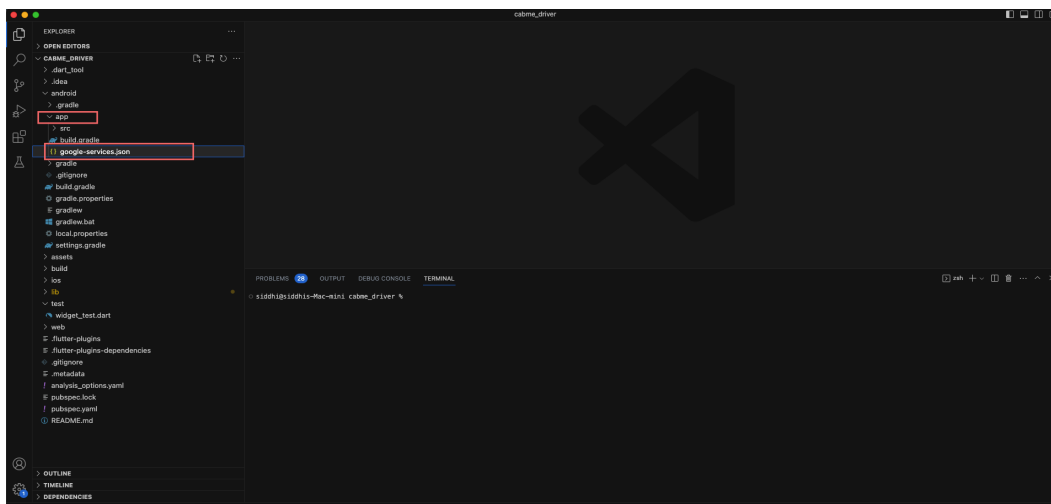
3(a) i. Enter package name



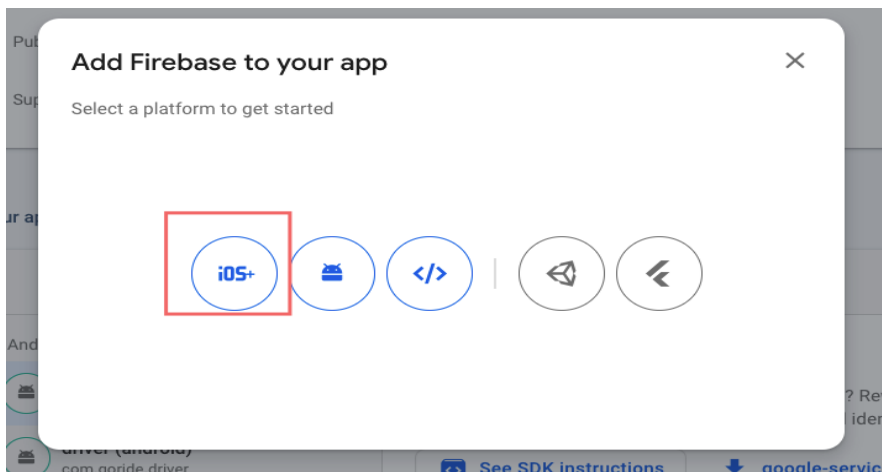
3(a) ii. Download google-services.json file.



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

