



Flutter App Setup Guide

Thank You for your purchase.....

Thank you for choosing our code. We appreciate your purchase and aim to provide exceptional service. If you need help with the code or documentation, contact our [Support Team](#).

Supported Flutter and Dart version

- Flutter:3.35.5 (Stable)
- Dart: 3.9.2
- DevTools: 2.48.0
- Java (JDK): 17.0.13 (LTS)

Also Ensure that you have installed JDK version

java 17.0.13 2024-10-15 LTS

Java(TM) SE Runtime Environment (build 17.0.13+10-LTS-268)

Java HotSpot(TM) 64-Bit Server VM (build 17.0.13+10-LTS-268, mixed mode, sharing)

- You Can find your JDK version using below command paste in your ide terminal and press enter button
> **java --version**
- You Can find your flutter version using below command type in terminal of vscode or android studio
> **flutter --version**
- And you can upgrade your flutter version Using below command
> **flutter upgrade**
- If you're still facing trouble, please explore the official Flutter documentation.
👉 Official guide: [Flutter Installation Guide](#)
- Prefer learning through videos?
👉 Check out this Flutter playlist: [Links](#)

Steps to Run This Project Successfully

1. Begin by extracting the downloaded code, resulting in the creation of the eSimTel - Flutter Code zip folder.
2. Next, extract the contents of the eSimTel App - Flutter Code folder and launch it in either Android Studio or Visual Studio Code(Any one of them you can prefer).
3. Access the IDE terminal and enter the command "flutter pub get" to fetch the necessary dependencies that we have used.
4. If you want to run the app in IOS simulator (**otherwise skip this step**), then execute the following commands in the terminal:

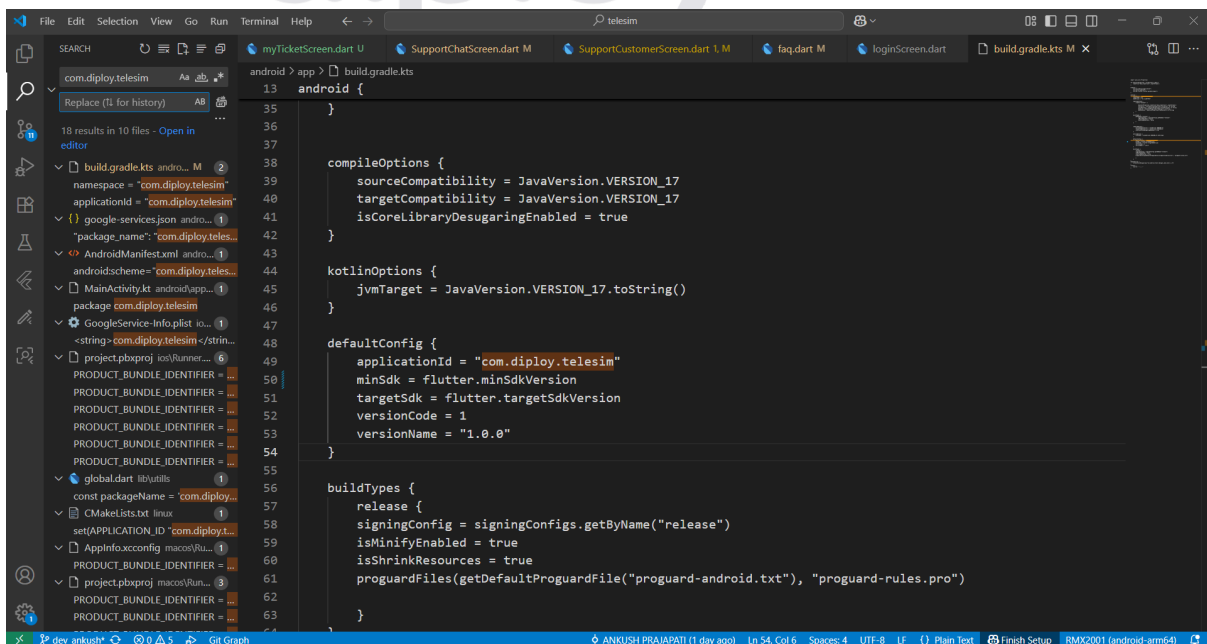
```
cd ios
pod install
cd ..
```

5. Finally, open the terminal and input "flutter run" to run the application. This will initiate the execution of the project.

Updating App Assets and Package Name

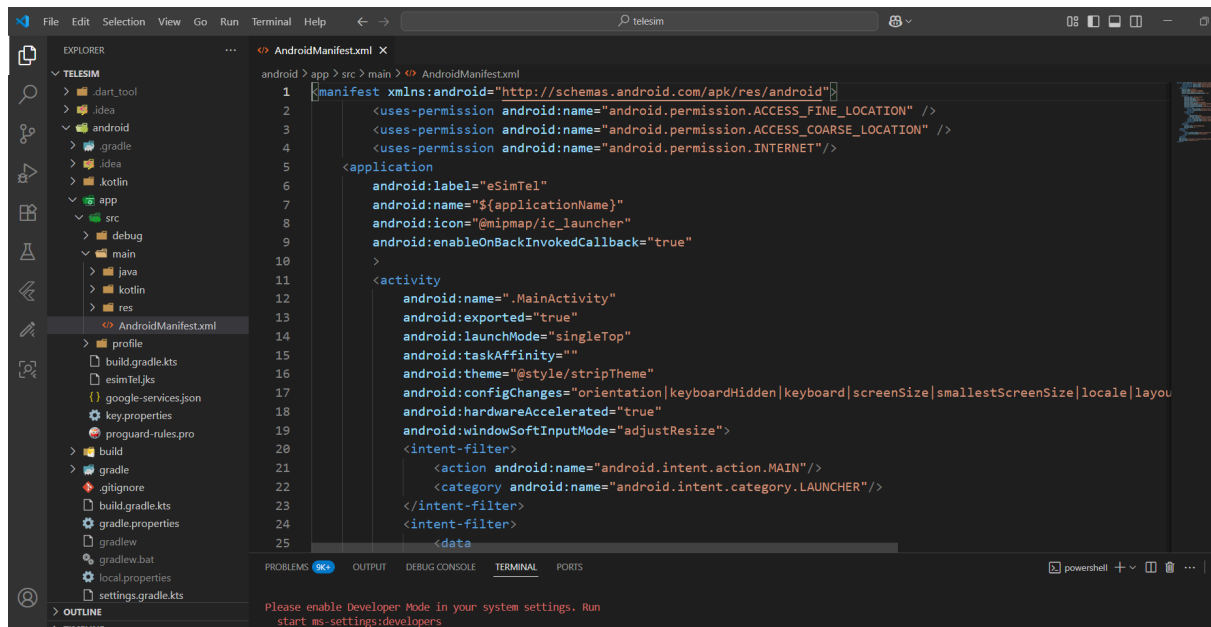
Replace package Name --

1. If you are using Android Studio, you can press **CTRL+SHIFT+R** to open Replace in Files tool or in case of VS Code you can press **CTRL+SHIFT+F** then search our package name **com.diploy.telesim**
2. Paste your package name and replace it in all places



Change Application Name –

Similarly you can change the app name by finding and replacing in all the places. Only you Need to Follow this path [android\app\src\main\AndroidManifest.xml](#) and change your app name on the `android:label` copy the label name and press **CTRL+SHIFT+F** check out the below screenshots



Change App Logo

For Creating the logo Go to the web site <https://www.appicon.co/> paste your app logo here and tick the android and ios boxes only, now click on the Generate button it will generate a zip folder for you, now extract this folder and follow the below instructions for changing the app logo.

Note: Make sure your app icons are in **pixel format of 512×512** and in **.png** format only.

Change the Brand text Logo in the App Bar and on the Package listing Card

Note: Please note that your text logo should be in **pixel format of 263x63** and also should be in **.png or .jpeg** format with transparent background, Also ensure that logo can not contain any surrounding wide spaces, for proper fitting.

Open your vscode and paste the logo inside the [assets/images](#) folder and copy the name of logo go to `image.dart` file and press **CTRL+F** search for `"ESimTel_TextLogo"` and update the name with your logo name

EXPLORER

TELESIM

- lib
- utils
 - services
 - api_end_points.dart
 - app_providers.dart
 - appColors.dart
 - config.dart
 - failurewidget.dart
 - FallbackLocalizationDelegate.dart
 - firebase_options.dart
 - global.dart
 - image.dart
 - notificationUtils.dart
 - randomColors.dart
 - TimeZoneHelper.dart
 - UserService.dart
- views
- widgets
- main.dart
- linux
- macos

OUTLINE

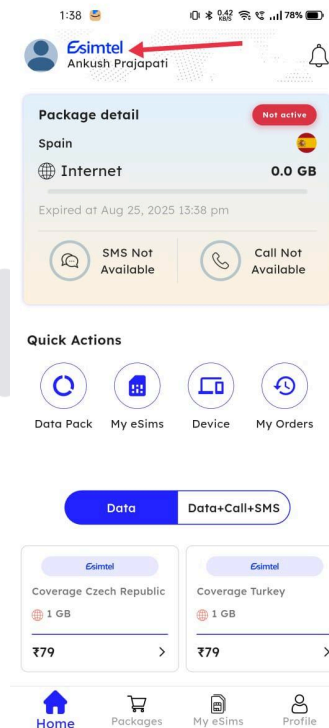
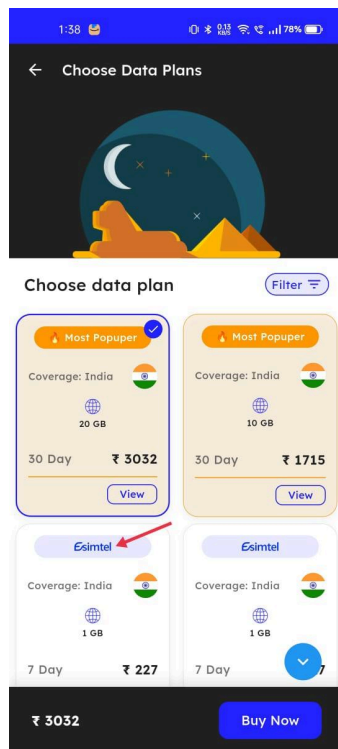
TIMELINE

DEPENDENCIES

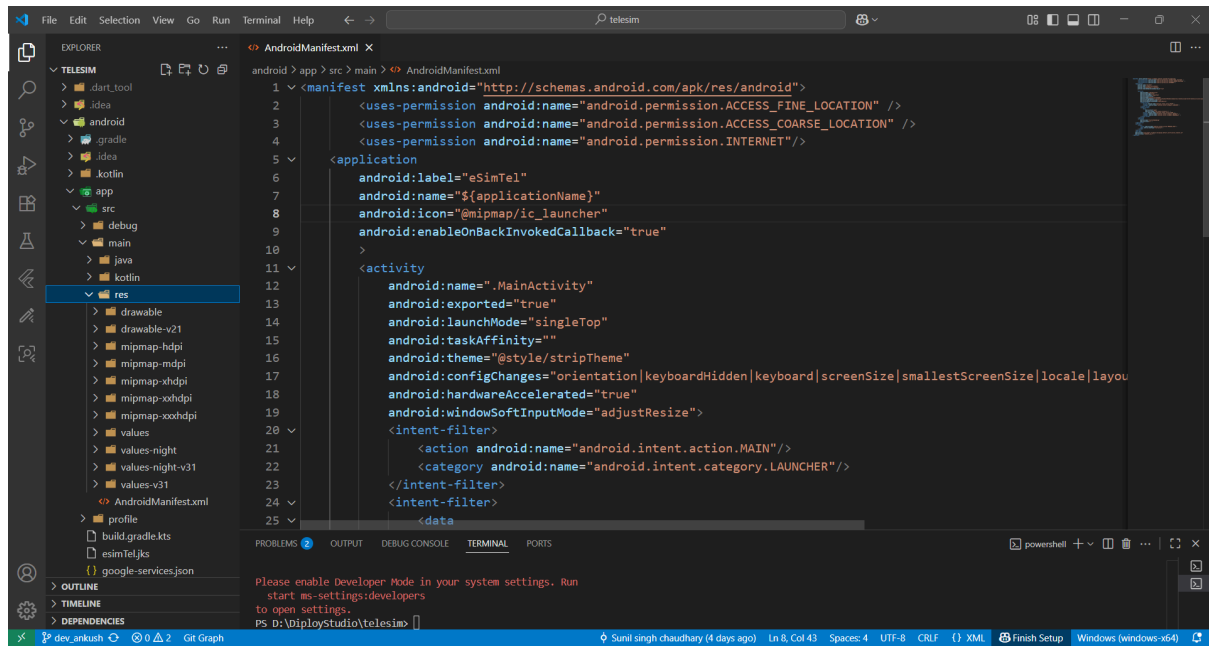
JAVA PROJECTS

lib > utils > image.dart > ESIMTel_TextLogo

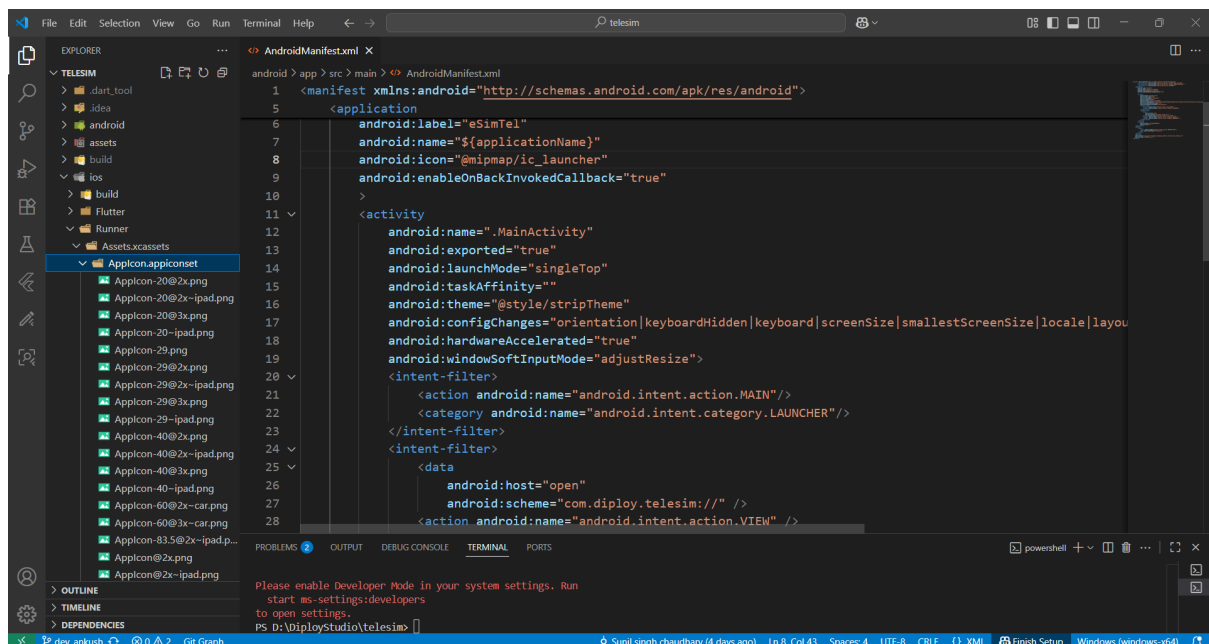
```
1 class Images {
2
3 // -----World Map-----
4 static const String worldMapImage = "assets/images/worldMap.png";
5
6 // -----Link Share-----
7 static const String linkShare = "assets/images/link.png";
8 static const String referAndEarn = "assets/images/referandEarn.png";
9 static const String emptyCart = "assets/images/empty_Cart.jpg";
10 static const String circleLoader = "assets/lottie/circular_loader.json";
11 static const String dotLoader = "assets/lottie/circle_Dots_loader.json";
12 static const String uploadImage = "assets/images/upload.png";
13
14 // -----KYC Related Images-----
15 static const String kycImage = "assets/images/kyc.png";
16 static const String ESIMTel_TextLogo = "assets/images/sintel-main.png";
17 static const String kycrejected = "assets/images/reject.png";
18 static const String kycapproved = "assets/images/verified.png";
19 static const String kycpending = "assets/images/pending.png";
20
21 // -----MyEsim Card Image-----
22 static const String earthImage = "assets/images/earth.png";
23 static const String lockImage = "assets/images/lock.png";
24 static const String packageImage = "assets/images/packaging.png";
25 static const String calenderImage = "assets/images/time-and-date.png";
26 static const String priceImage = "assets/images/dollar.png";
27 static const String validityImage = "assets/images/checked.png";
28 static const String infoImage = "assets/images/info.png";
29
30 static const String loginImage = "assets/images/login.png";
31
32 }
```



1. For Android, open android > app > src > main > res and add here your logo according to device screen size



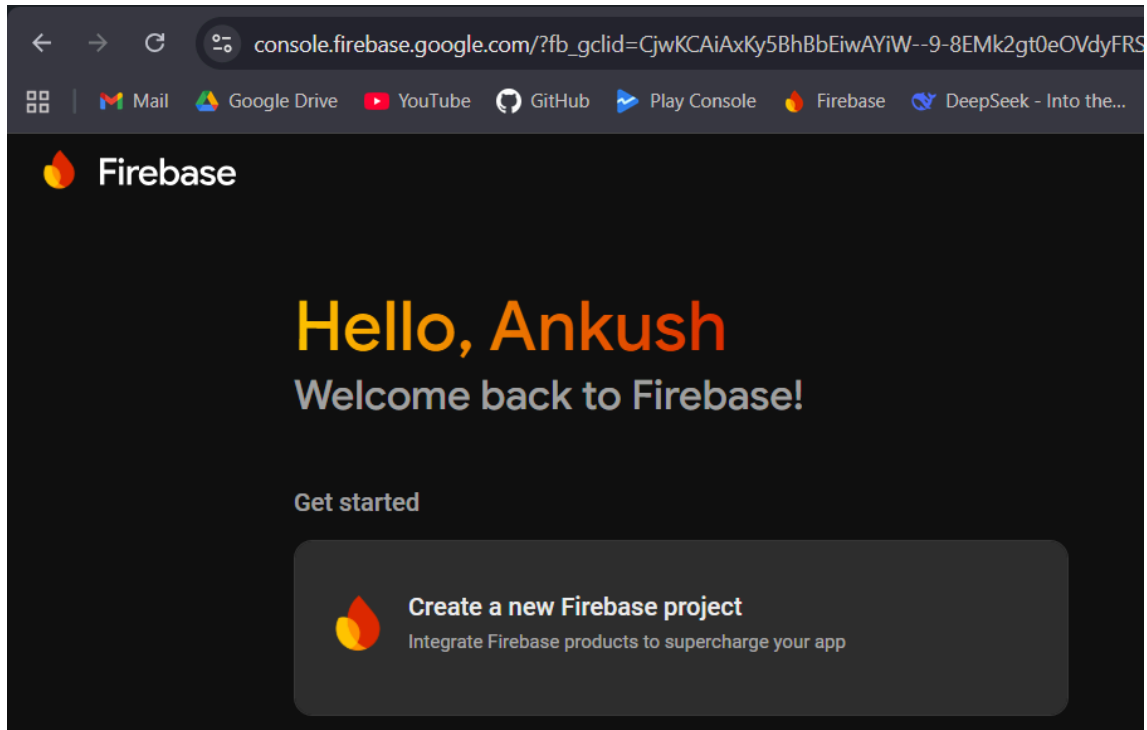
2. For IOS open, ios > Runner > Assets.xcassets > AppIcon.appiconset here and add your logo according to different size.



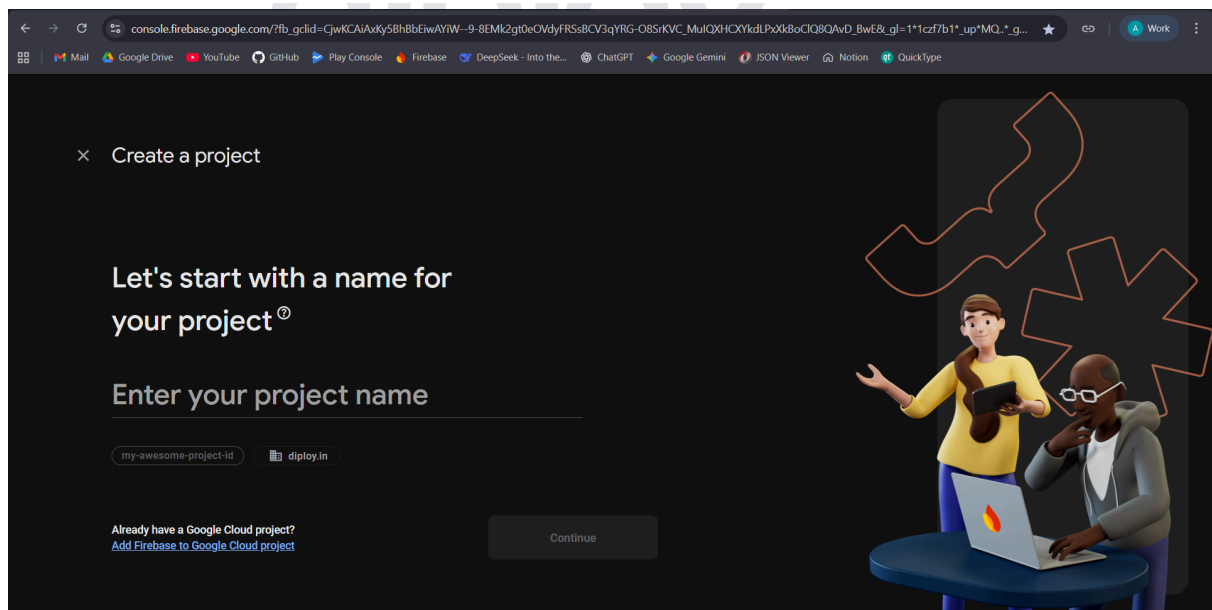
Now Let's begin with **Firebase integration in your Flutter app** 🚀.

- First, Create a new Firebase Project
- Go to Firebase Console <https://console.firebase.google.com/>

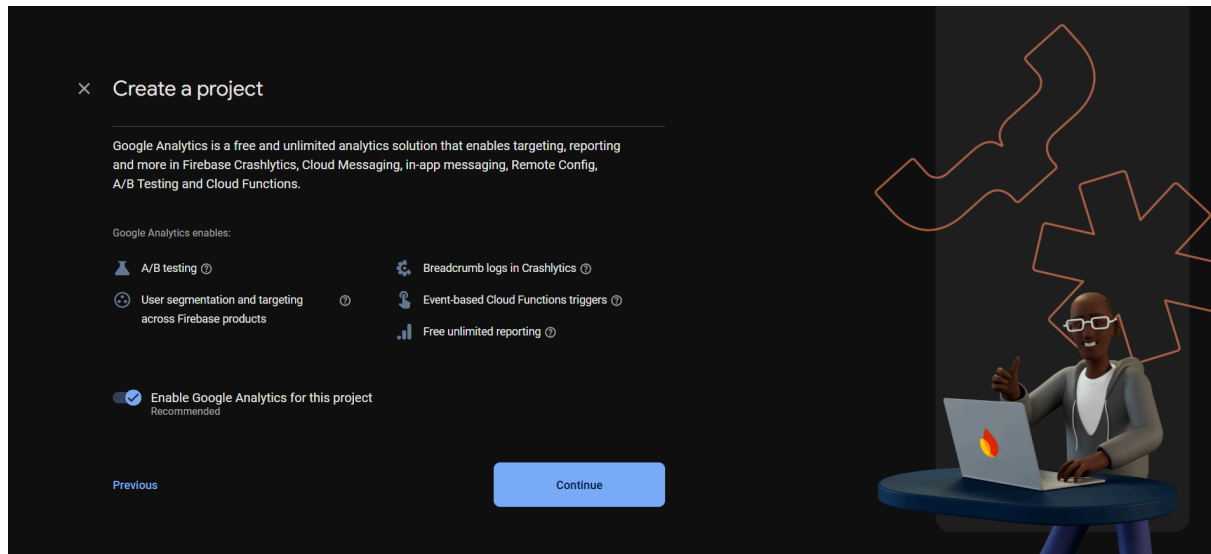
Step 1: Click on Create a New Firebase Project.



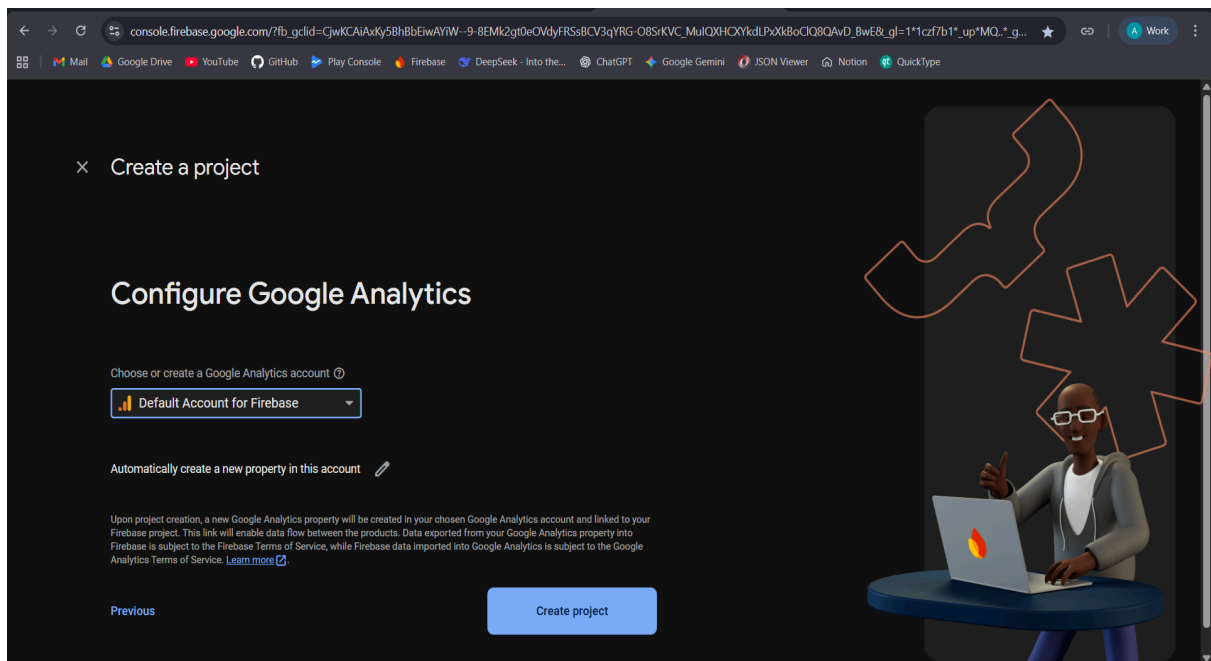
Step 2: Enter your project name and click on Continue



Step 3: Click on Continue



Step 4: Select default Account For Firebase and Click on Create Project.

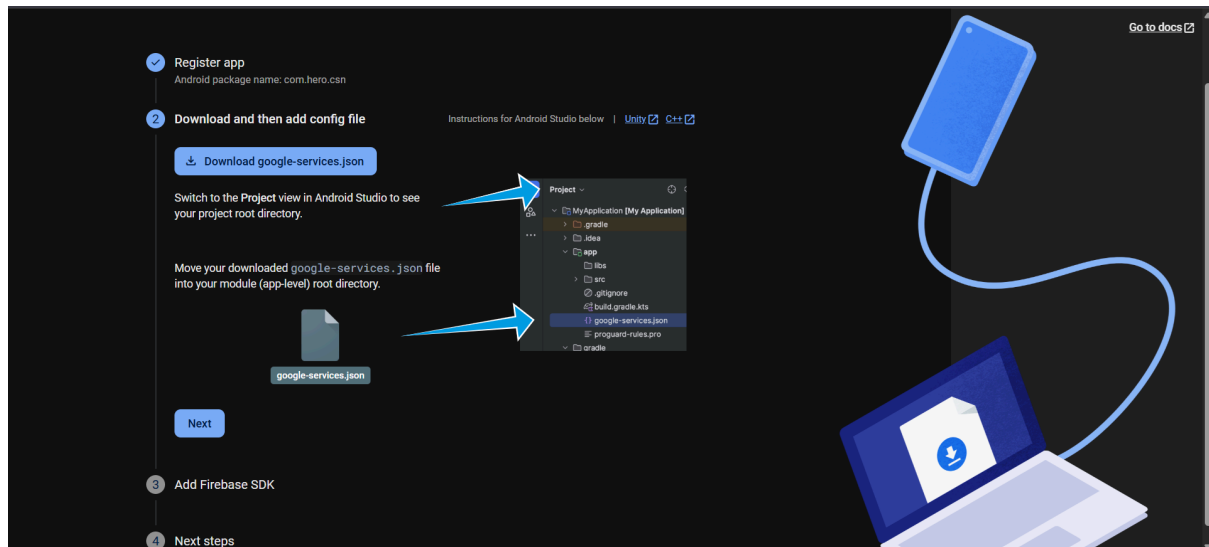


Wooh! You have created the Flutter project successfully, now it's time to connect your Android and iOS Flutter projects, respectively.

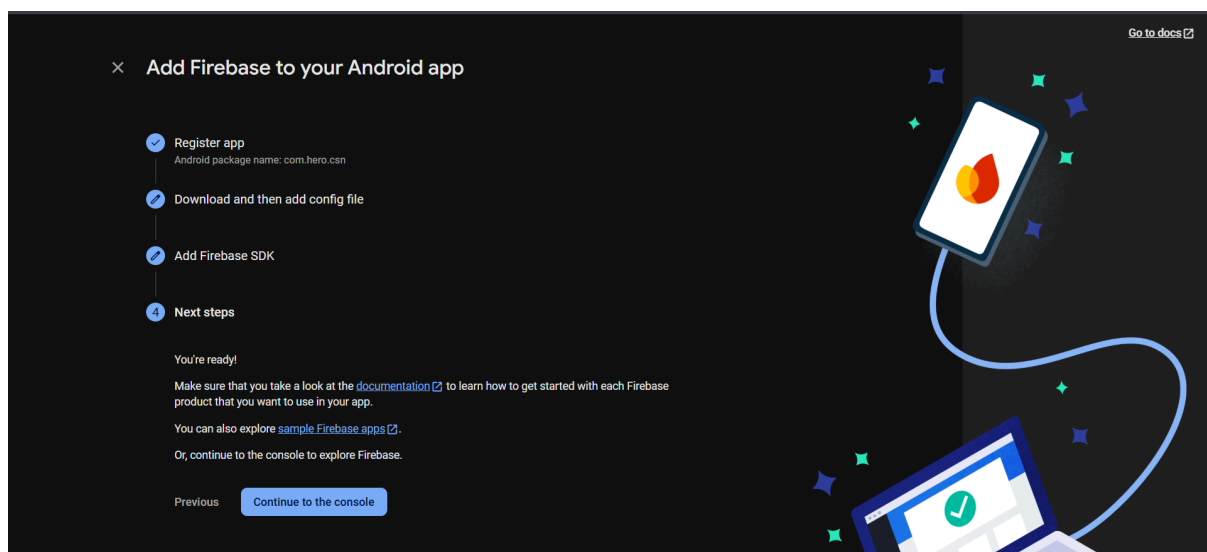
Step-1: Click on android Icon

[illegible]

Step- 3: Download the **google-Service.json** file and place the at the same location given in the below screenshot

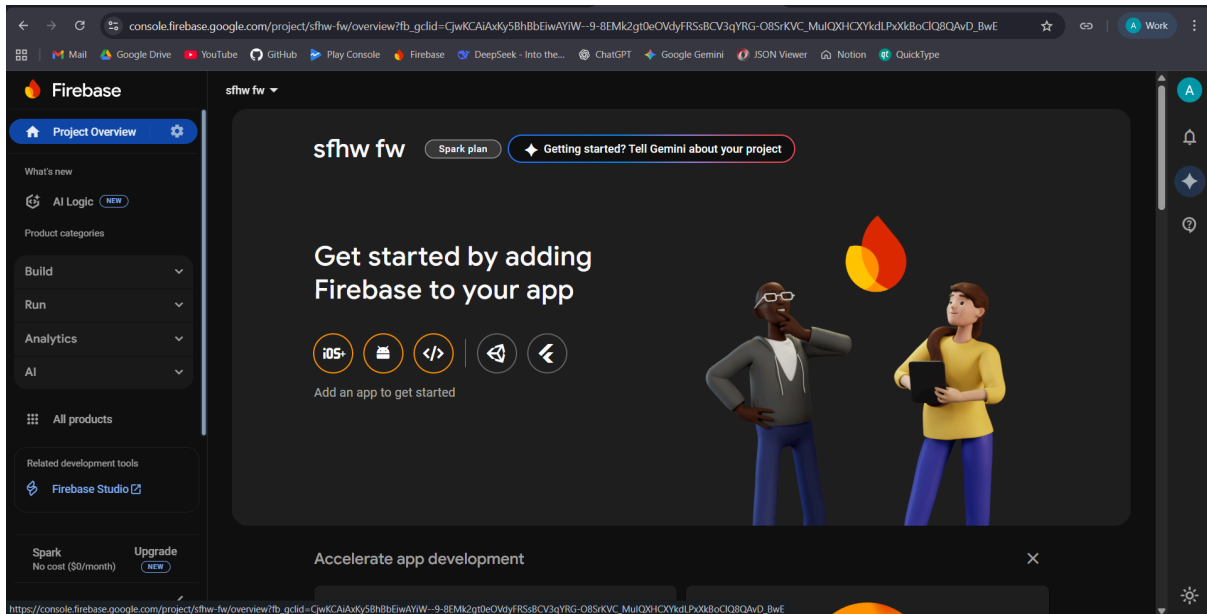


Step- 4: Now Click on Next and finally click on continue to console

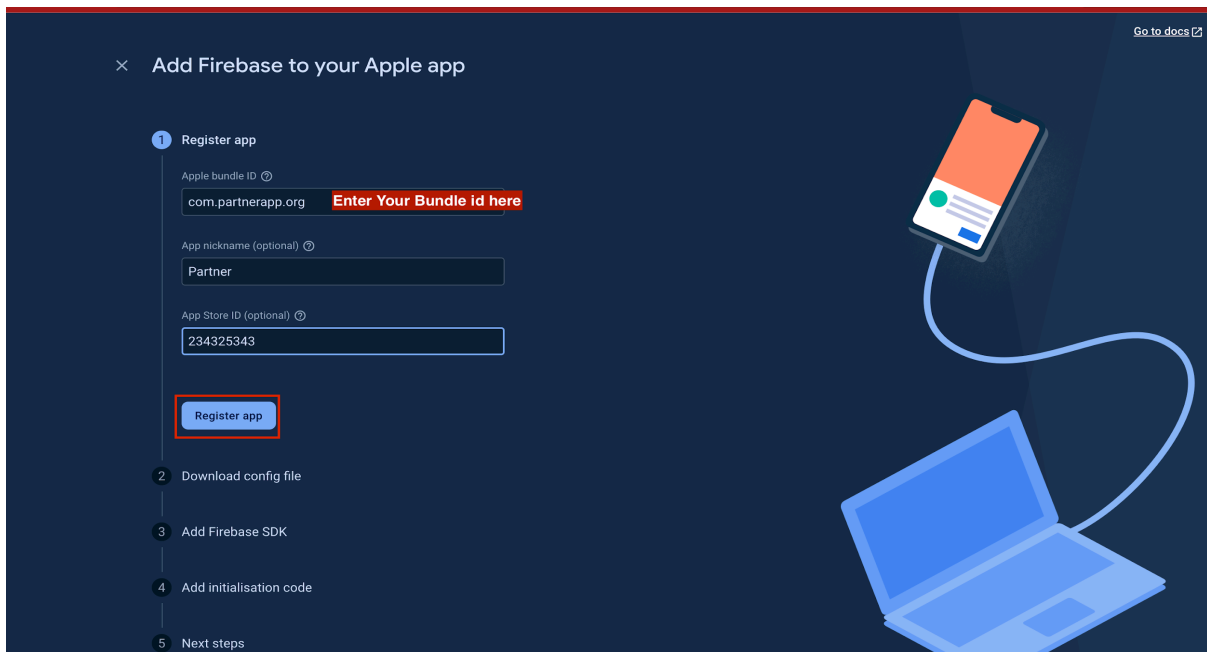


Connect Your **IOS Flutter Project** with Firebase...

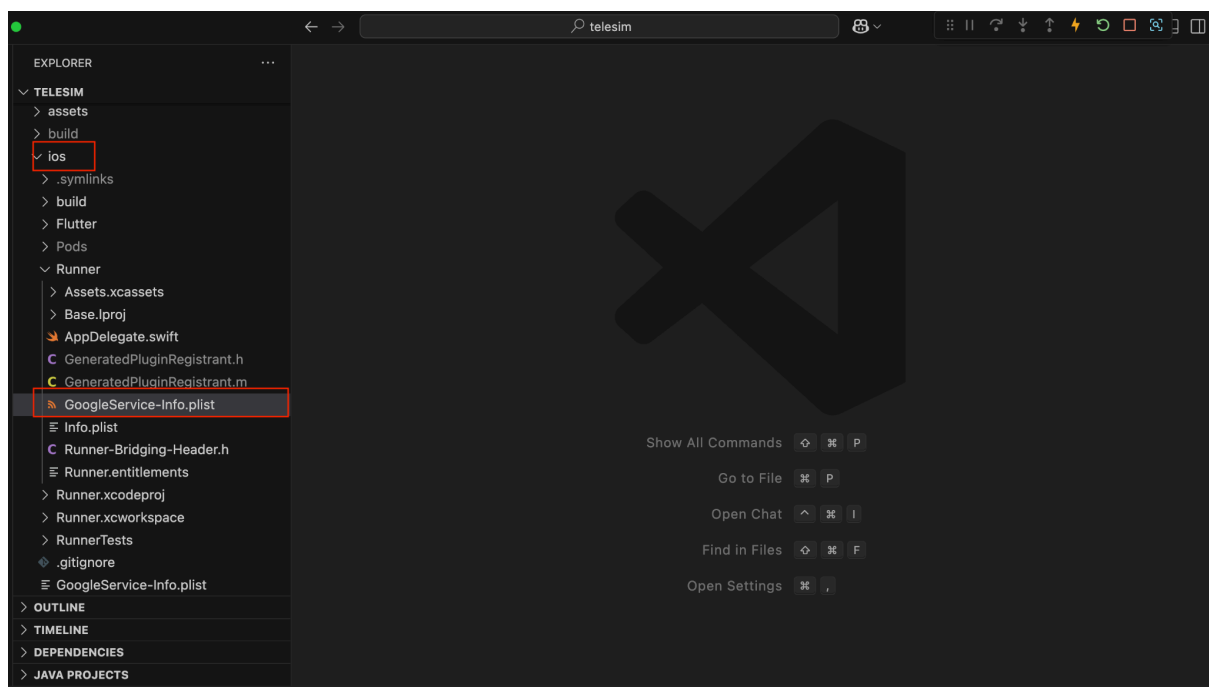
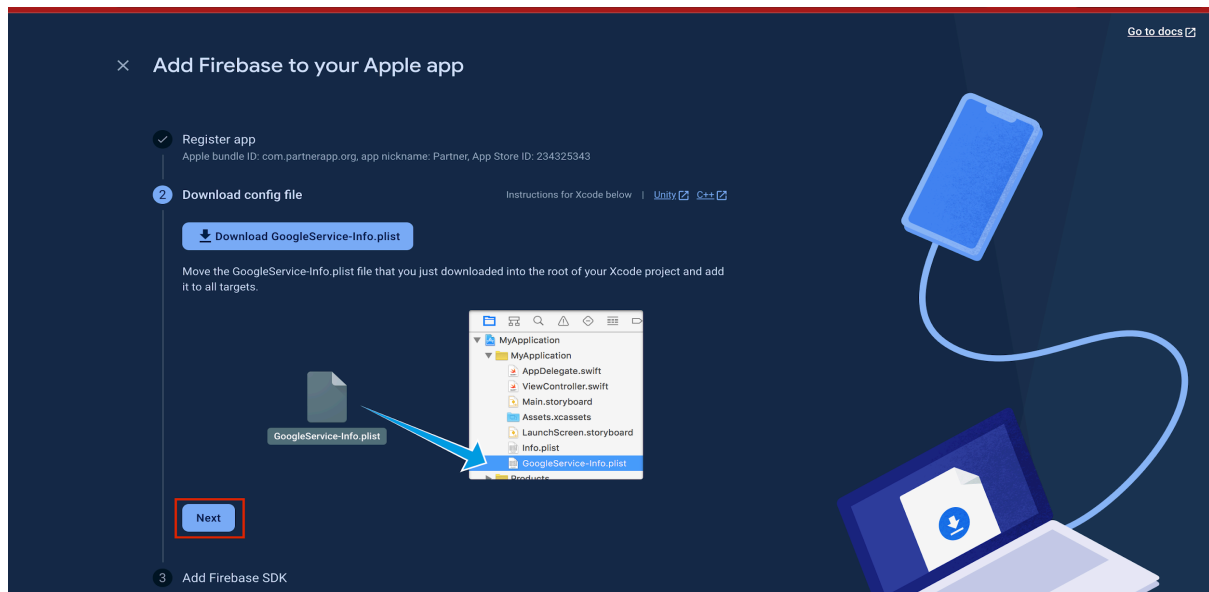
Step-1: Click on Ios Icon



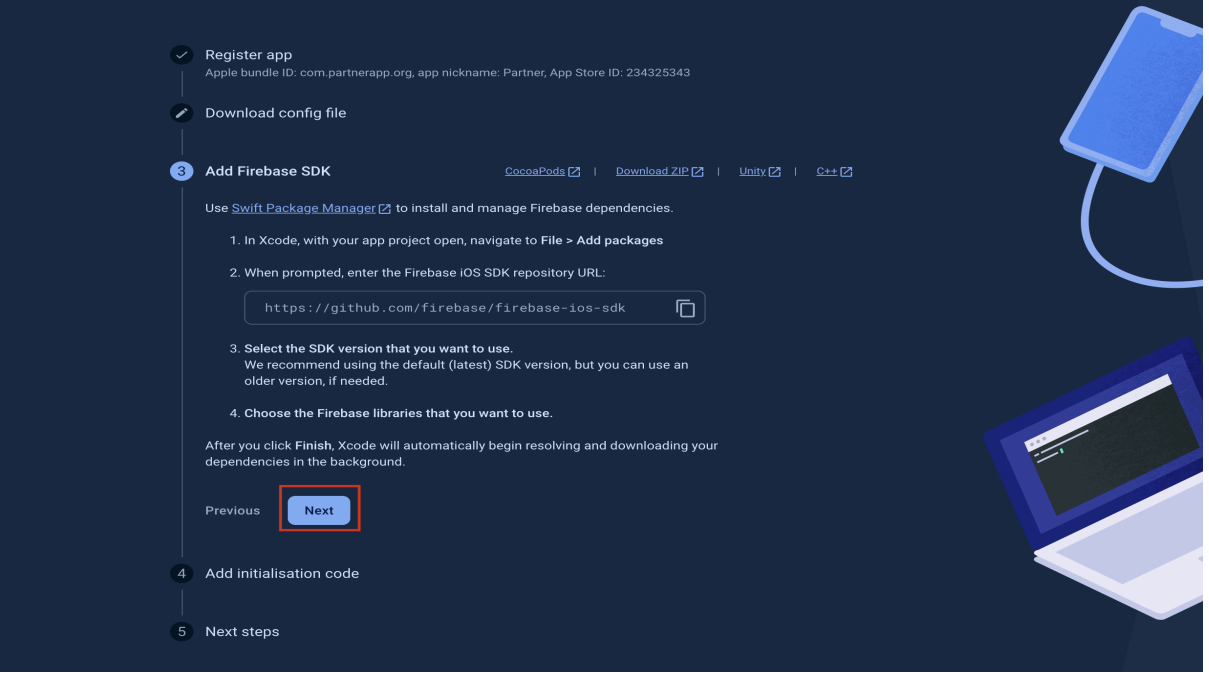
Step-2: add your Package name , SHA Key(Optional) and click on Register App



Step-3: Download the **google-Service.json** file and place the at the same location given in the below screenshot



Step-3: Click on Next



The screenshot shows the 'Add Firebase SDK' step in a tutorial. It includes a progress bar with five steps: 1. Register app, 2. Download config file, 3. Add Firebase SDK (current), 4. Add initialisation code, and 5. Next steps. Step 3 is highlighted with a blue circle. The instructions for step 3 are: 'Use Swift Package Manager to install and manage Firebase dependencies.' followed by four numbered steps: 1. In Xcode, with your app project open, navigate to File > Add packages; 2. When prompted, enter the Firebase iOS SDK repository URL: `https://github.com/firebase/firebase-ios-sdk`; 3. Select the SDK version that you want to use. We recommend using the default (latest) SDK version, but you can use an older version, if needed.; 4. Choose the Firebase libraries that you want to use. Below the instructions, there is a 'Previous' button and a 'Next' button, which is highlighted with a red rectangle. The 'Next' button is a blue button with white text. The background of the screen is dark blue with a faint illustration of a smartphone and a laptop.

✓ Register app
Apple bundle ID: com.partnerapp.org, app nickname: Partner, App Store ID: 234325343

✓ Download config file

3 Add Firebase SDK [CocoaPods](#) | [Download ZIP](#) | [Unity](#) | [C++](#)

Use [Swift Package Manager](#) to install and manage Firebase dependencies.

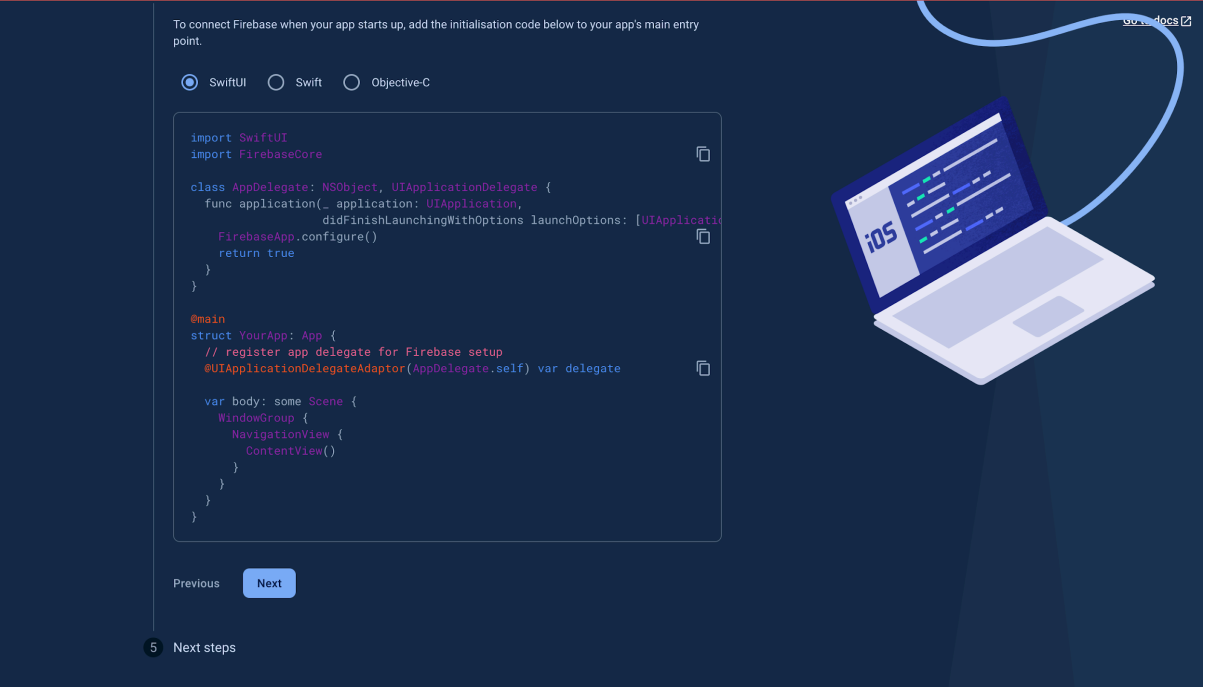
1. In Xcode, with your app project open, navigate to File > Add packages
2. When prompted, enter the Firebase iOS SDK repository URL:
`https://github.com/firebase/firebase-ios-sdk`
3. Select the SDK version that you want to use.
We recommend using the default (latest) SDK version, but you can use an older version, if needed.
4. Choose the Firebase libraries that you want to use.

After you click Finish, Xcode will automatically begin resolving and downloading your dependencies in the background.

Previous **Next**

4 Add initialisation code

5 Next steps



The screenshot shows the 'Add initialisation code' step in a tutorial. It includes a progress bar with five steps: 1. Register app, 2. Download config file, 3. Add Firebase SDK, 4. Add initialisation code (current), and 5. Next steps. Step 4 is highlighted with a blue circle. The instructions for step 4 are: 'To connect Firebase when your app starts up, add the initialisation code below to your app's main entry point.' Below the instructions, there are three radio buttons: 'SwiftUI' (selected), 'Swift', and 'Objective-C'. Below the radio buttons, there is a code editor with the following Swift code:

```
import SwiftUI
import FirebaseCore

class AppDelegate: NSObject, UIApplicationDelegate {
    func application(_ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]? = nil) -> Bool {
        FirebaseApp.configure()
        return true
    }
}

@main
struct YourApp: App {
    // register app delegate for Firebase setup
    @UIApplicationDelegateAdaptor(AppDelegate.self) var delegate

    var body: some Scene {
        WindowGroup {
            NavigationView {
                ContentView()
            }
        }
    }
}
```

 Below the code editor, there is a 'Previous' button and a 'Next' button, which is highlighted with a red rectangle. The 'Next' button is a blue button with white text. The background of the screen is dark blue with a faint illustration of a smartphone and a laptop.

To connect Firebase when your app starts up, add the initialisation code below to your app's main entry point.

☒ SwiftUI ☐ Swift ☐ Objective-C

```
import SwiftUI
import FirebaseCore

class AppDelegate: NSObject, UIApplicationDelegate {
    func application(_ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]? = nil) -> Bool {
        FirebaseApp.configure()
        return true
    }
}

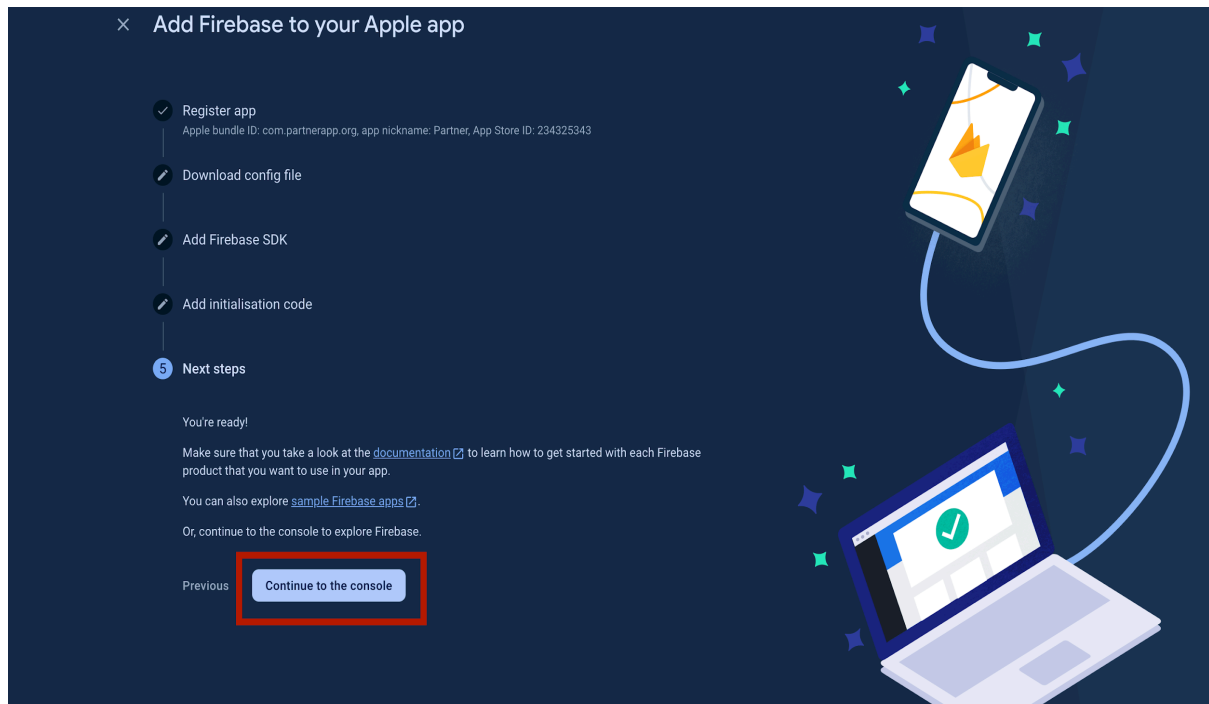
@main
struct YourApp: App {
    // register app delegate for Firebase setup
    @UIApplicationDelegateAdaptor(AppDelegate.self) var delegate

    var body: some Scene {
        WindowGroup {
            NavigationView {
                ContentView()
            }
        }
    }
}
```

Previous **Next**

5 Next steps

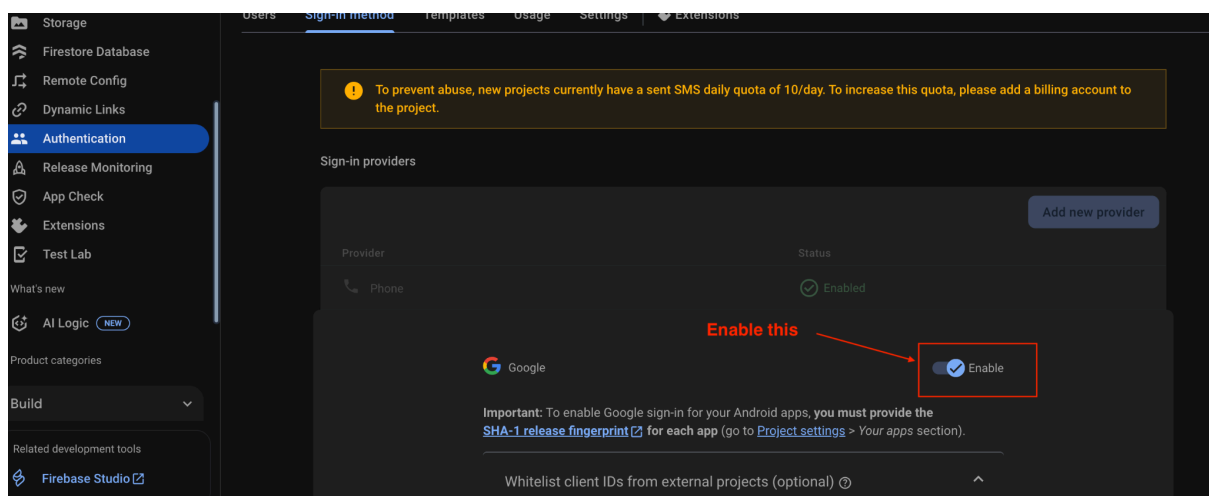
Step-4: Now Click on Continue to the Console

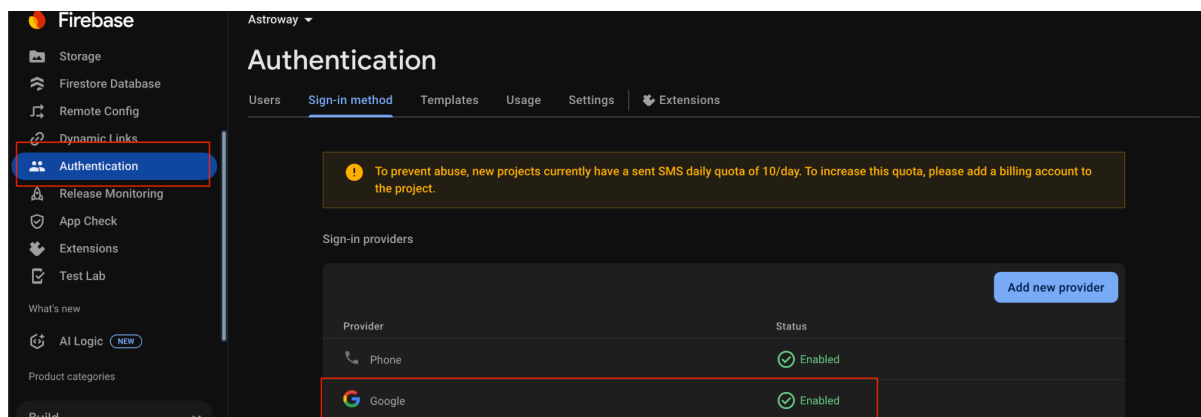


Congrats! You have successfully connected your android & IOS app to your firebase project.

Enable Firebase Authentication

- Go to firebase and in left side search Authentication and enable google sign in like in below screenshot





please remember you have to add first jks file in **android** → **app** → **yourjksfile.jks**

How to Generate the .jks file

For Android go to terminal and go to **android** -> **app** right click and click in copy relative path

```
> keytool -genkeypair -v \
-keystore "paste your relative path"/your-key-name.jks \
-keyalg RSA \
-keysize 2048 \
-validity 10000 \
-alias your-key-alias-name \
-storepass your-store-password \
-keypass your-key-password \
-dname "CN=YourName, OU=YourUnit, O=YourCompany, L=YourCity, S=YourState, C=YourCountryCode"
```

For Example

```
keytool -genkeypair -v \
-keystore
/Users/sunilsingh/Desktop/clientsApps/arthJyoti-astrologer-app/android/app/esim_tel.jks \
-keyalg RSA \
-keysize 2048 \
-validity 10000 \
-alias esim_tel\
-storepass 111111 \
-keypass 111111 \
-dname "CN=esimTel, OU=N/A, O=N/A, L=delhi, S=delhi, C=IN"
```

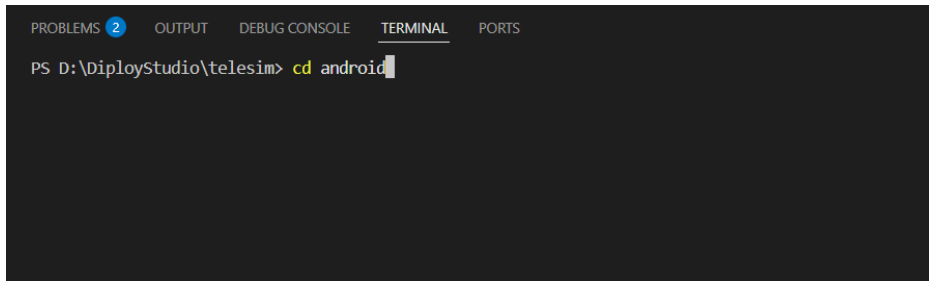
Or checkOut the official link to generate the SHA key [Link](#)

For IOS Use this Command

```
openssl req -new -key ios_distribution.key -out ios_distribution.csr -subj
"/CN=YourName/OU=YourUnit/O=YourCompany/L=YourCity/ST=YourState/C=IN"
```

- Get SHA-1 and SHA-256 keys [debug and release both] and add in below firebase setting to Get the SHA Key's Open your project go to the terminal paste the below command to navigate the android folder

> **cd android**



A screenshot of an IDE terminal window. The terminal has tabs for PROBLEMS (2), OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active. The prompt is 'PS D:\DiployStudio\telesim>' and the command 'cd android' is being entered.

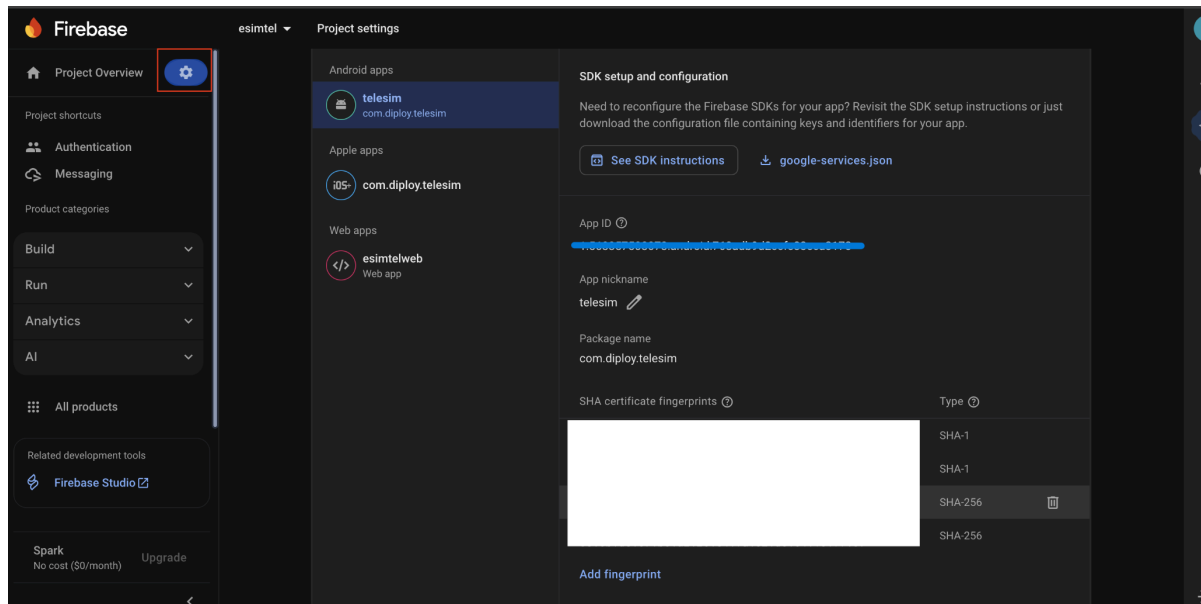
- Now once you have done this paste this command to generate the SHA keys

> **./gradlew signingReport**



A screenshot of an IDE terminal window. The terminal has tabs for PROBLEMS (2), OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active. The prompt is 'PS D:\DiployStudio\telesim\android>' and the command './gradlew signingReport' is being entered.

This command will generate sha keys copy the now copy the SHA-1 and SHA-256(both release and debug) keys from the terminal and go to firebase console-> project settings -> under general tab-> click on **add FingerPrint** add your SHA key's

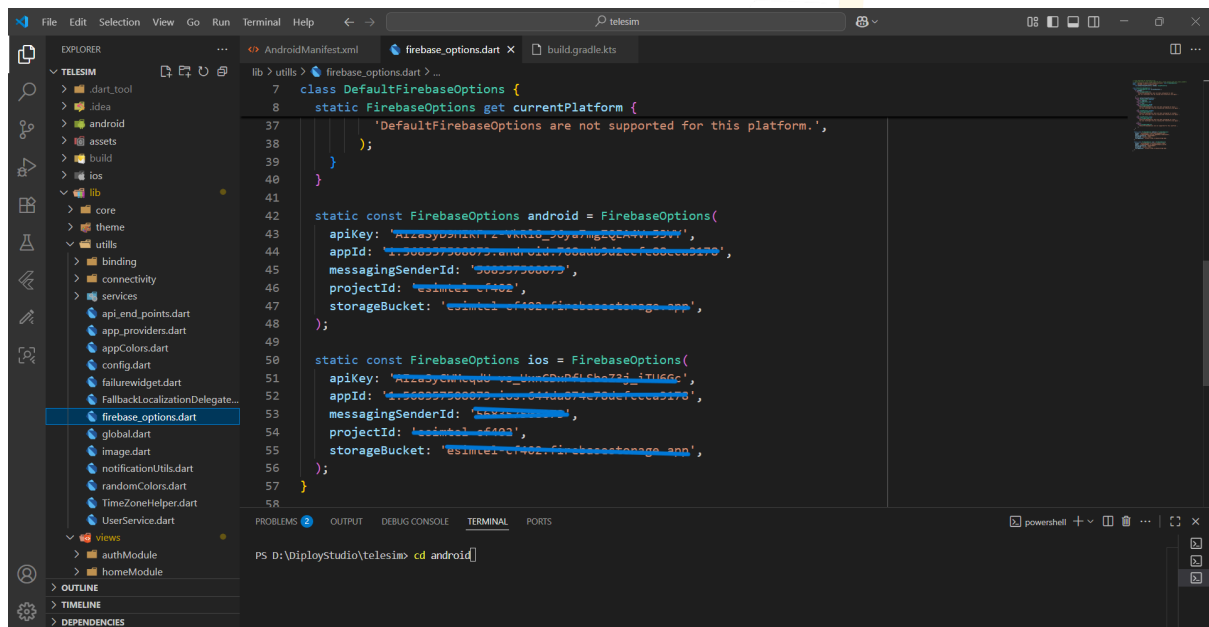


Changes in firebase_options.dart file

Now you need to make changes in firebase_options.dart file

File path:- **project>lib>utils>firebase_options.dart**

For this file you can get data from the **google-services.json** please make sure all the data that you are changing in **firebase_options.dart** is same as in **google-service.json**



Set app's default language

The App's default language is English, but you can easily modify it. To do so, navigate to the **lib/main.dart** file and locate the section displaying the language codes, as illustrated in the image below.

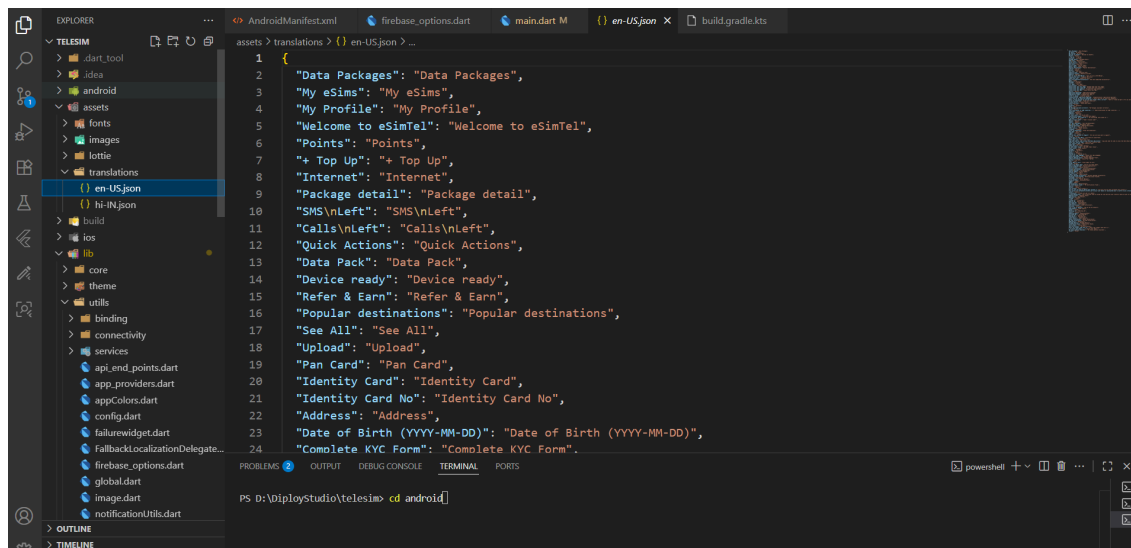
You can obtain your desired language code from the following link:

<https://developers.google.com/admin-sdk/directory/v1/languages>.

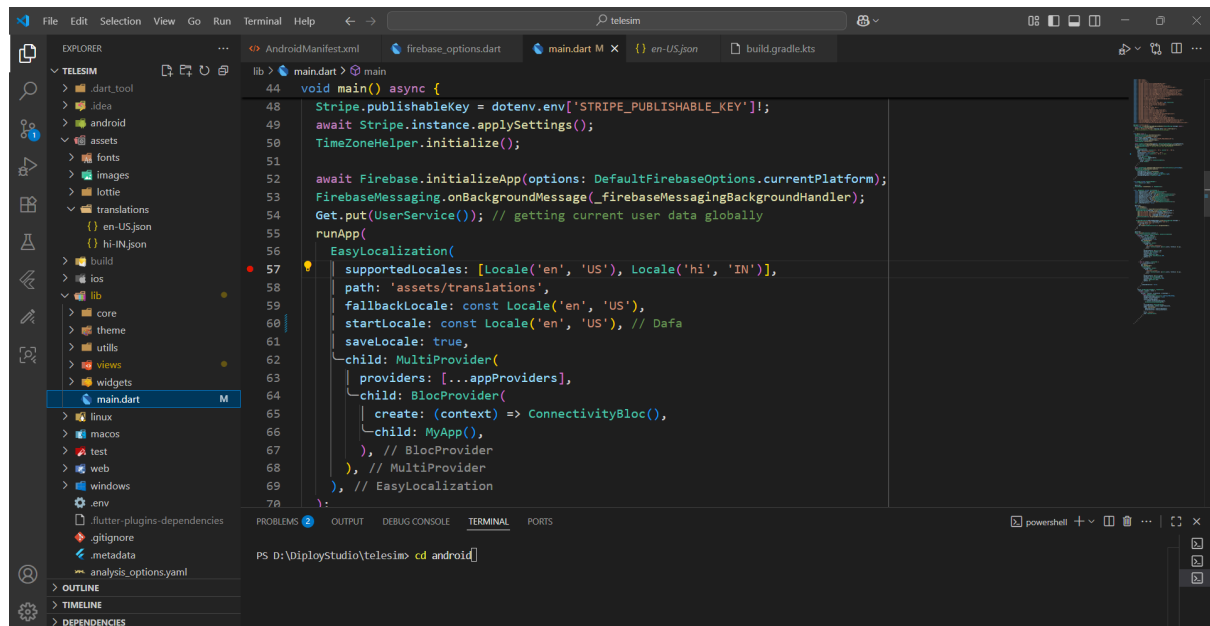
Just ensure that the language code you select is available in the aforementioned link for proper functionality.

```
runApp(  
  EasyLocalization(  
    supportedLocales: [Locale('en', 'US'), Locale('hi', 'IN')],  
    path: 'assets/translations',  
    fallbackLocale: const Locale('en', 'US'),  
    startLocale: const Locale('en', 'US'), // default app language  
    saveLocale: true,  
    child: MultiProvider(  
      providers: [...appProviders],  
      child: BlocProvider(  
        create: (context) => ConnectivityBloc(),  
        child: MyApp(),  
      ),  
    ),  
  ),  
);
```

If you are adding a new language to the application, make sure to create a separate file inside **assets/translations/** with the appropriate language code (e.g., **en-US.json** for English with the United States country code). If your desired language is not available. Copy all labels from en-US.json and translate the values accordingly. And add Your language code in main.dart file

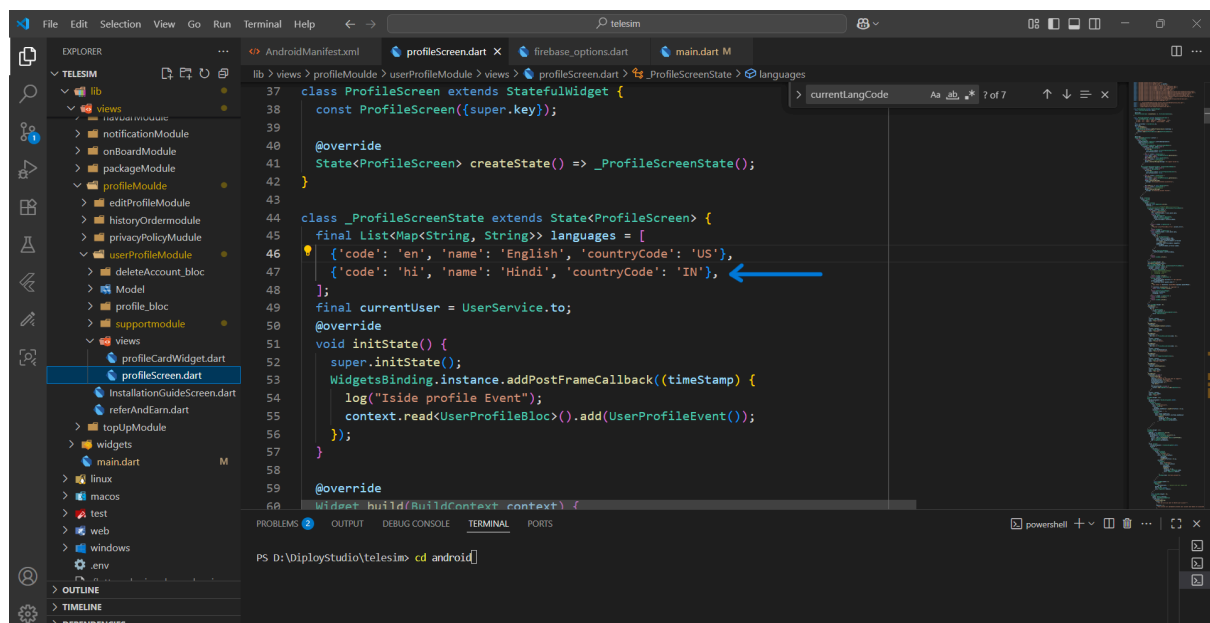


Add your desired language to the `supportedLocales` list as well to the `main.dart`



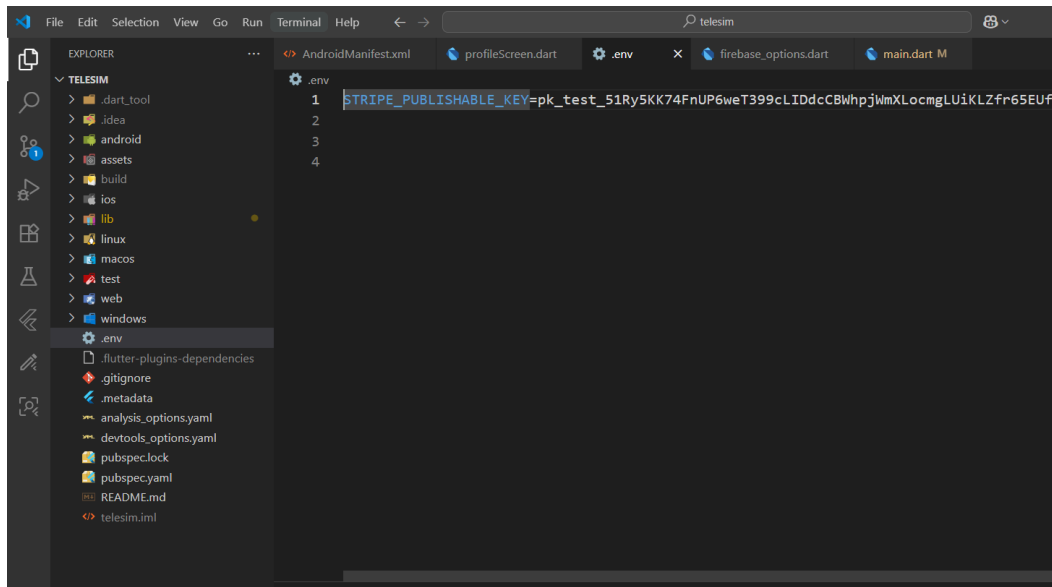
file

Make Sure you have added the language code, country code, name as well in the `profilescreen.dart` file



Change the App Credentials and keys

- Change the Stripe payment gateway key, search for .env file and replace the keys



Setting Up In-App Purchase (iOS)

Follow the steps below to configure your **Apple In-App Purchase Key** for the app.

Step 1: Verify Apple Developer Account

If you already have an active Apple Developer account, you can proceed directly to the next step.

Otherwise, make sure you have:

- An **Apple Developer Account** with the necessary permissions (Account Holder or Admin role).
- Access to **App Store Connect**.

Step 2: Get Your Apple Team ID

1. Go to [Apple Developer Account](https://developer.apple.com/account/).
2. Scroll to the bottom of the page — you'll find your **Team ID** under the **Membership Information** section.
3. Copy this **Team ID** for later use.

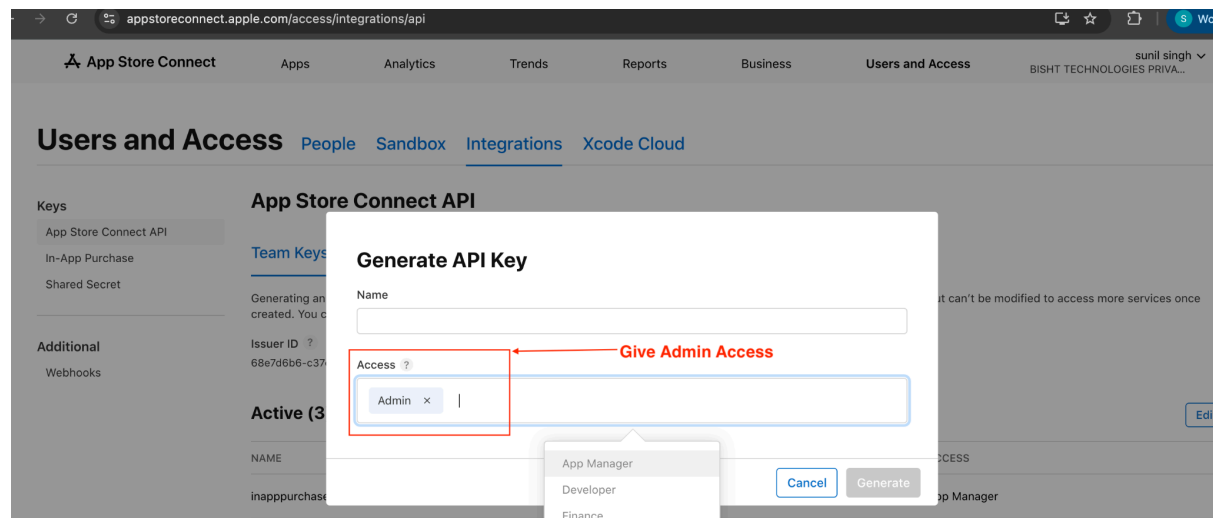
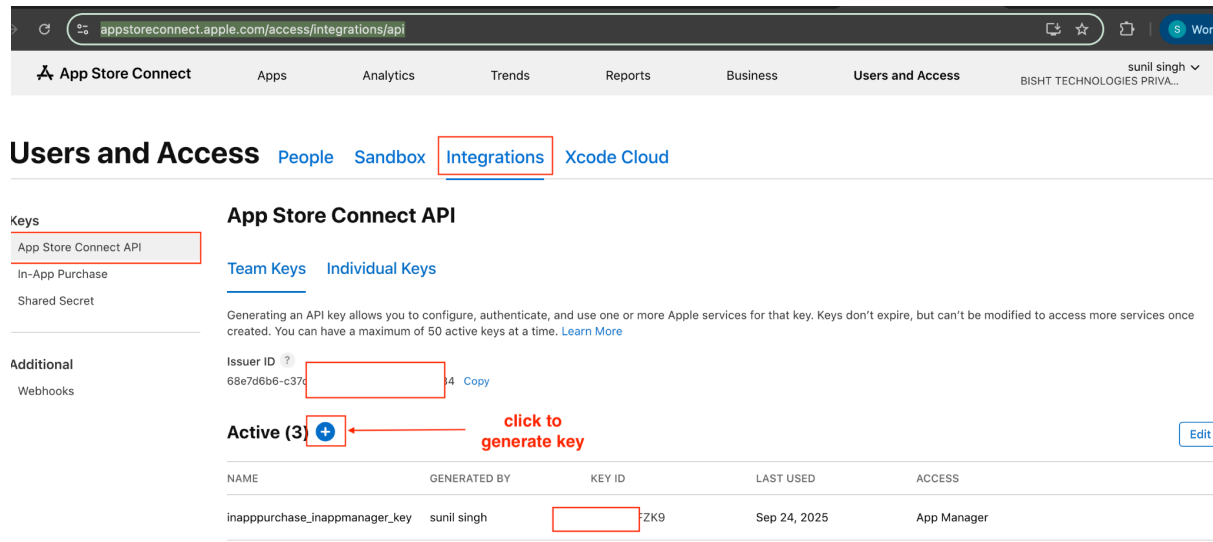
The screenshot shows the 'Account' page for a user named 'sunil singh' from 'BISHT TECHNOLOGIES PRIV...'. The 'Membership details' section is visible, containing the following information:

Entity name	<input type="text" value="PRIVATE LIMITED"/>
Team ID	9A[redacted]P3
Program	Apple Developer Program
Enrolled as	Organization
Phone	91-9[redacted]3
Street address	House No. C-3, Top Floor Kapil Complex, Haldwani, Mukhani Nainital Haldwani, Uttarakhand 262402 India
Account Holder	N t
Your role	Developer
Renewal date	September 17, 2026
Annual fee	US\$99

A red box highlights the 'Team ID' field, and a red arrow points to it with the text 'Copy your TeamID from here'.

Step 3: Create a New In-App Purchase API Key

1. Visit [App Store Connect → API Keys](#).
2. Click on “+” to create a new API key.
3. Enter a suitable **name** (e.g., **InAppPurchaseKey**).
4. Set the **Access Role** to **App Manager** or **Developer** (depending on your needs).
5. Click **Generate**.
6. After generation, download the **.p8 key file** and note down the following details:
 - **Key ID**
 - **Issuer ID**
 - **Team ID** (from Step 2)
 - **Private Key (.p8 file)**



✓ You Now Have:

- **.p8 private key file**
- **Key ID**
- **Issuer ID**
- **Team ID**

These will be required for integrating and authenticating In-App Purchase APIs in your app or backend.

1. **Upload Your .p8 private key file** into backend Location **storage/app/apple/**

Contact Support

Delighted to serve and support you! Contact our support team. We're available Mon-Fri, 9:00 am - 6:00 pm IST (GMT +5.30),

India - Asia. Expect prompt responses within 24 hours via comments, forum, or email.

Reach Us

Unlock the potential of diploy, your go-to hub for top-notch Flutter app development and web applications. Reach out to us today for quality solutions and exciting freelance opportunities.

Visit us at diploy.in or drop an email to start@diploy.in

Thank you for considering us!

