

Thank you for your purchase! To ensure seamless after-sales support, please activate your product using the link below:

[Activate Now](#)

Your activation is essential for us to provide you with the best assistance. We appreciate your cooperation.

Technology Used

- **Flutter** with Dart Language for Mobile App
- **DB Browser for SQLite** for Database

Please follow the below steps to set up the App.

(We have provided the steps to set up using Visual Studio Code Editor. You can use other editors also. Steps may vary based on your editor.)

1. Setup Prerequisite (If not available)

- a. Install and set up Flutter from [this link](#)
- b. Install DB Browser for SQLite from [this link](#)

2. Setup Mobile App (Technology Flutter)

- a. Initial steps to set up and run mobile app
 - i. Open the **App** folder in the VSCode
 - ii. Run the following commands in the VSCode Terminal

```
flutter clean  
flutter pub get
```

- iii. Additional steps to set up for iOS (You can skip these steps if you don't want to set up for iOS)
 1. In the VSCode terminal, go to the ios directory
(using the command **cd ios**)
 2. Run the following command to install pods

pod install

- iv. Connect your Android or iOS device with your machine

(To run on an Apple device, you must have an Apple computer)

- v. Run the following command to run on an Android or iOS device

flutter run

- vi. To upgrade the Flutter version run the following command in the VSCode Terminal

(Only if your Flutter version is lower than mentioned in this document)

flutter upgrade**b. Changes for About screen**

You have to change your about us screen URLs, for that go to the file located at **lib\screens\aboutScreen.dart**

1. Search **<YOUR EMAIL>** and replace it with your email
2. Search **<YOUR WEBSITE>** and replace it with your website
3. Search **<YOUR FB PAGE URL>** and replace it with your facebook page URL
4. Search **<YOUR INSTAGRAM PAGE URL>** and replace it with your instagram page URL
5. Search **<YOUR LINKEDIN PAGE URL>** and replace it with your linkedin page URL
6. Search **<YOUR TWITTER PAGE URL>** and replace it with your twitter page URL
7. Search **<YOUR WEB SITE URL>** and replace it with your website URL

c. Changes for App store and Play store

You have to change your app store and play store URLs, for that go to the file located at **lib\utils\global.dart**

1. Search **<PLAY STORE URL>** and replace it with your play store URL
2. Search **<APP STORE URL>** and replace it with your app store URL

d. Changes for Privacy policy

You have to change your privacy policy URL, for that go to the file located at **lib\screens\privacyPolicyScreen.dart**

1. Search **<PRIVACY POLICY URL>** and replace it with your privacy policy URL

e. Change Package Name/Bundle ID

An app's package name is a unique identifier that is automatically created when you create an app. The term used for iOS apps is "**bundle ID**" and for Android apps, it is "**package name**".

i. Set Package Name for Android App

1. Change the package name in the file located at **android/app/src/main/AndroidManifest.xml**

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  package="com.demo.accountingapp">
```

2. Change the Package Name in the file which is located at **android/app/build.gradle**

```
defaultConfig {
    // TODO: Specify your own unique Application ID (https://developer.android.com/studio/about/application-ids)
    applicationId "com.demo.accountingapp"
    minSdkVersion 22
    versionCode flutterVersionCode.toInteger()
    versionName flutterVersionName
    multiDexEnabled true
    targetSdkVersion 34
}
```

3. Change the folder structure for the below path as per your package name.

android\app\src\main\java\com\demo\accountingapp

Ex. If your package name is com.app.accountingapp

android\app\src\main\java\com\app\accountingapp

4. Change Package Name in file which is located at **android\app\src\main\java\com\demo\accountingapp\MainActivity.java**

```
d > app > src > main > java > com > demo > accountingapp > MainActivity.java
package com.demo.accountingapp;
import io.flutter.embedding.android.FlutterActivity;
import io.flutter.embedding.android.FlutterFragmentActivity;
```

- ii. Set Bundle ID for iOS App

1. In VSCode

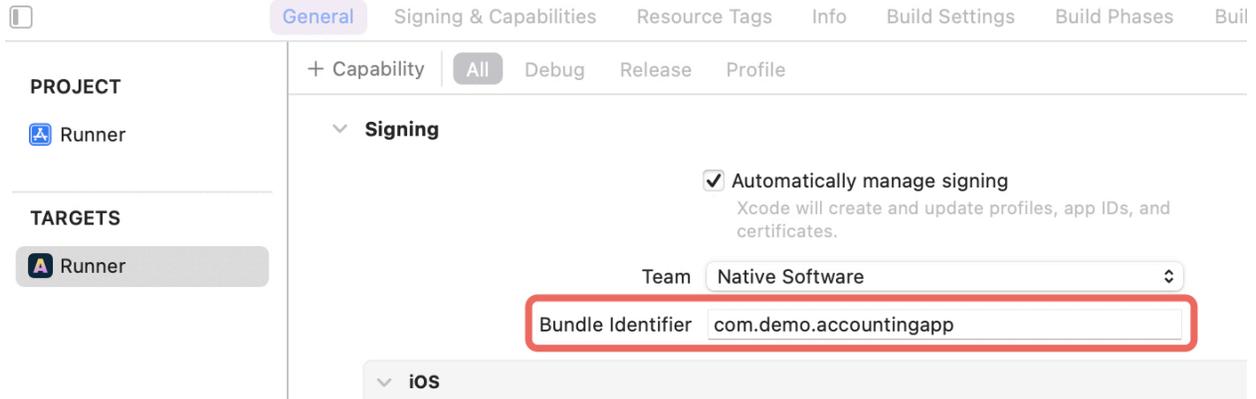
- a. Go to **ios/Runner/info.plist**
- b. Change the string of key **CFBundleIdentifier**

```
<key>CFBundleExecutable</key>
<string>$(EXECUTABLE_NAME)</string>
<key>CFBundleIdentifier</key>
<string>$(PRODUCT_BUNDLE_IDENTIFIER)</string>
<key>CFBundleInfoDictionaryVersion</key>
<string>5.0</string>
```

2. In XCode

- a. Right-click on the **IOS** folder and Choose Open in Xcode Option
- b. Click on the folder icon left side of the XCode window
- c. Select Runner.
- d. Select Target runner
- e. Go to identity

f. Change Bundle Identifier



f. Create and set Keystore file for Android

- i. Create a keystore.jks file, if not exist, use the below command in the terminal

```
keytool -genkey -v -keystore "path\keystore.jks" -storetype JKS -keyalg  
RSA -keysize 2048 -validity 10000 -alias keystore
```

- ii. Fill in all the details asked while executing the above command
- iii. Recommended. After creating your keystore.jks file, please put it in the **android/app** folder
- iv. Create key.properties file in the **android** folder and add the details in the file as per the below screenshot.

```
storePassword=android  
keyPassword=android  
keyAlias=keystore  
storeFile=keystore.jks
```



NOTE:

- If you have changed any default value for any of these keys (storePassword, keyPassword, keyAlias, storeFile) while creating the keystore.jks file, then please also change them to key.properties file.
- If you place your keystore.jks file somewhere else in the project than mentioned in step 5.c.iii then please change storeFile key value accordingly.

- For more details please refer to [this link](#)

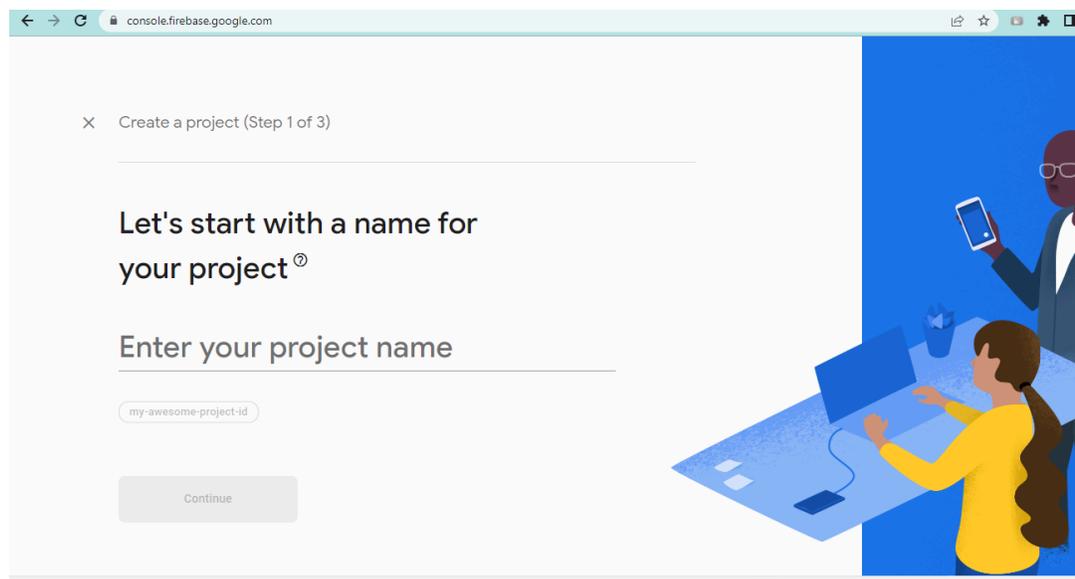
g. Create Firebase Account & Project

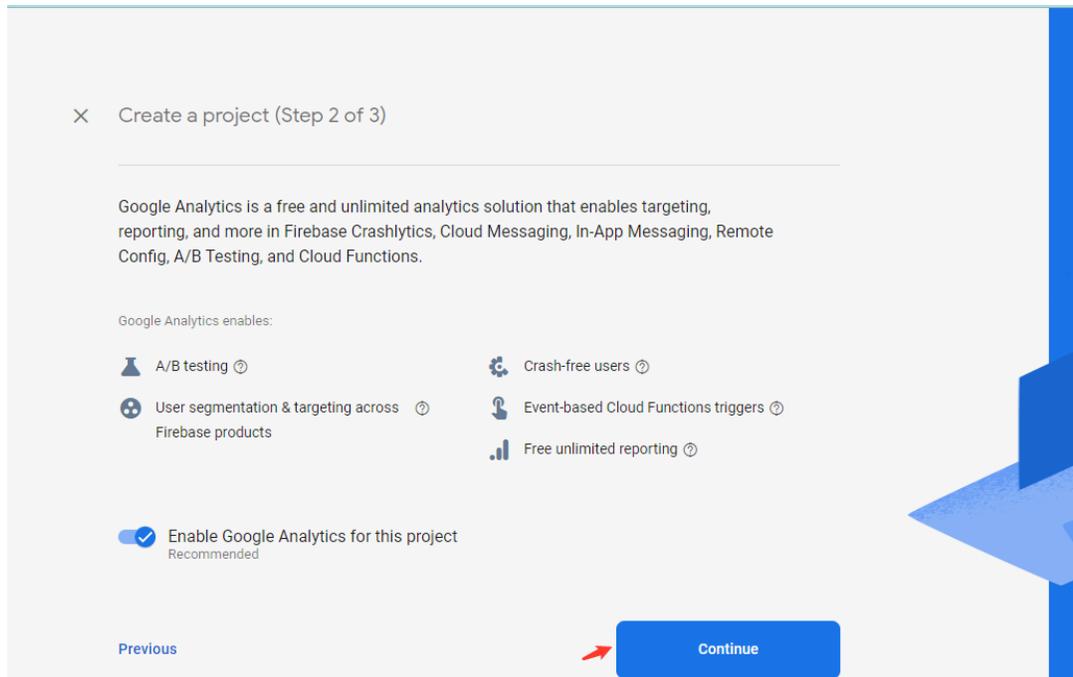
In this project, we are using the following Firebase services.

- i. Push Notification
- ii. Phone Authentication
- iii. Firestore Database
- iv. Firebase Analytics

For this, you need a Firebase account and a project set up in the Firebase. Please follow the below steps for this,

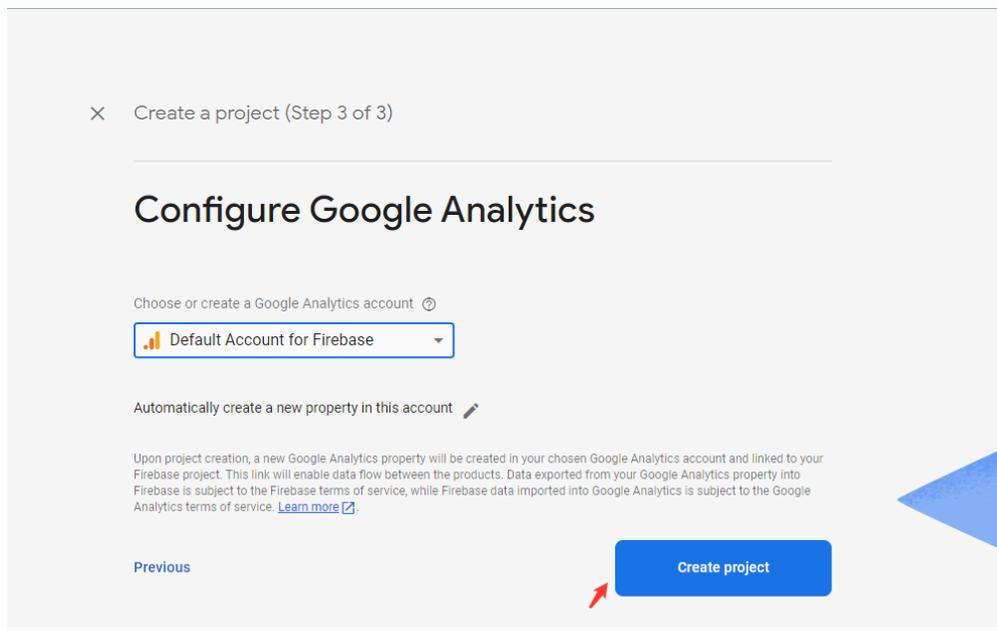
- i. Go to the [Firebase console](#)
- ii. Sign up if you don't have a Google Account or you want to create a new account for your project. Otherwise, sign in with your Google Account.
- iii. Click on **Add Project**
- iv. Enter your project name





v. Select Default Account for Firebase

(or you can create a new account)

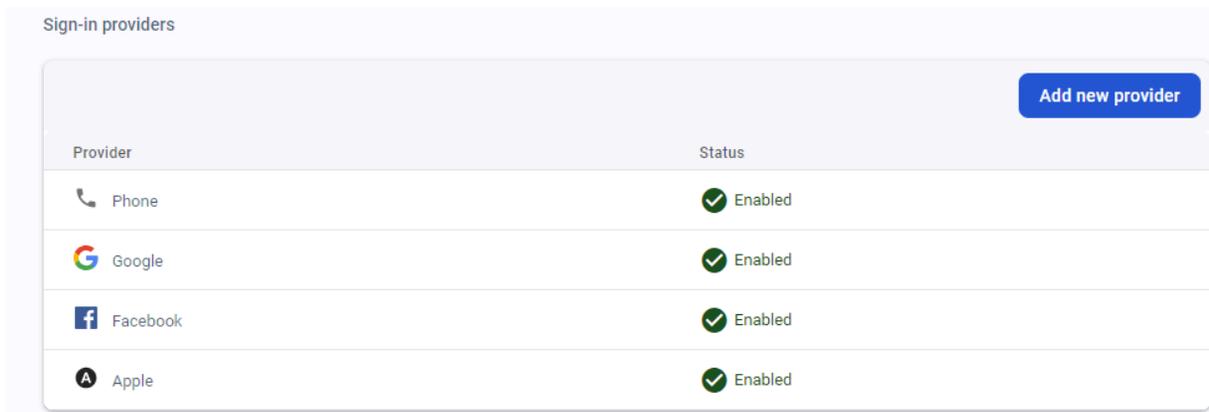


vi. Create project

h. Set up Android App in Firebase Project

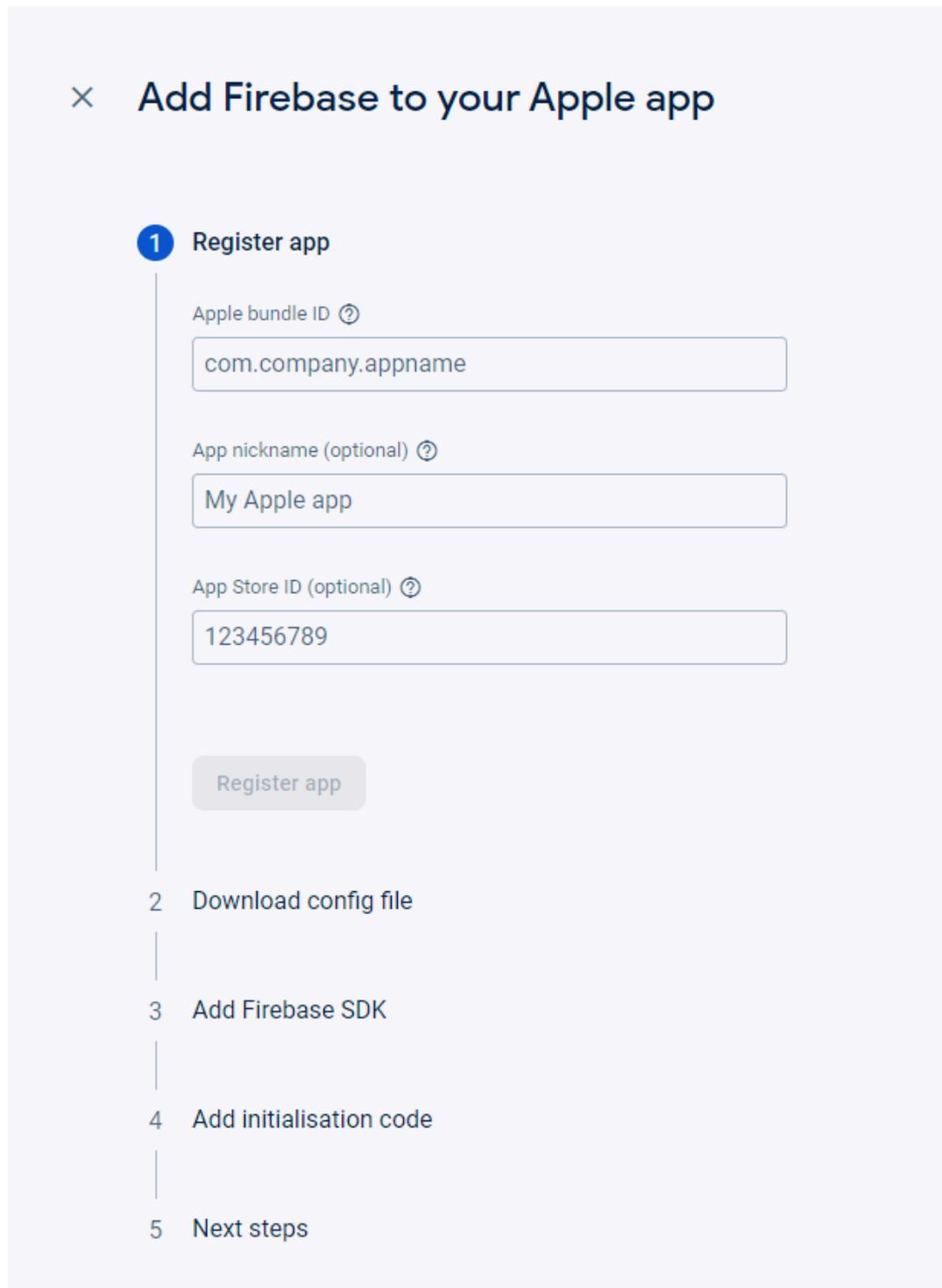
- i. Go to the Firebase console
- ii. Select the project you created in step 5.d.vi
- iii. Go to **Project Setting**
- iv. In the **General** Tab click on the **Add App** button
- v. Select **Android**
- vi. Fill out the form and click on the **Register App Button**

(Please check the below screenshot for reference)



i. Setup Firebase iOS App

- i. Go to the Firebase console
 - ii. Select the project you created in step 5.d.vi
 - iii. Go to **Project Setting**
 - iv. In the **General** Tab click on the Add App button
 - v. Select **iOS**
 - vi. Fill out the form and click on the **Register App** Button
- (Please check the below screenshot for reference)



- vii. Download the GoogleService-info.plist file from Firebase project settings and paste it at the **ios/Runner** location in the app

- viii. Go to the `ios\Runner\AppDelegate.m` file and replace “YOUR-API-KEY” with your Api key.

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    [GMServices provideAPIKey:@"YOUR-API-KEY"];
    [GeneratedPluginRegistrant registerWithRegistry:self];
    // Override point for customization after application launch.
    return YES;
}
```

- ix. Replace “YOUR-REVERSED-CLIENT-ID” with your reversed client ID. You can find this Id from the `GoogleService-info.plist` file you added from step 5.g.vii.

```
<string>Editor</string>
<key>CFBundleURLName</key>
<string>$(REVERSED_CLIENT_ID)</string>
<key>CFBundleURLSchemes</key>
<array>
    <string>YOUR-REVERSED-CLIENT-ID</string>
</array>
</dict>
<dict>
```

- x. XCode Project Setting

This step covers reviewing the most important settings in the XCode workspace. For detailed procedures and descriptions, see [Prepare for App Distribution](#)

1. Navigate to your target’s settings in XCode:

- a. Open the default Xcode workspace in your project by running the below command in a terminal window from your Flutter project directory.

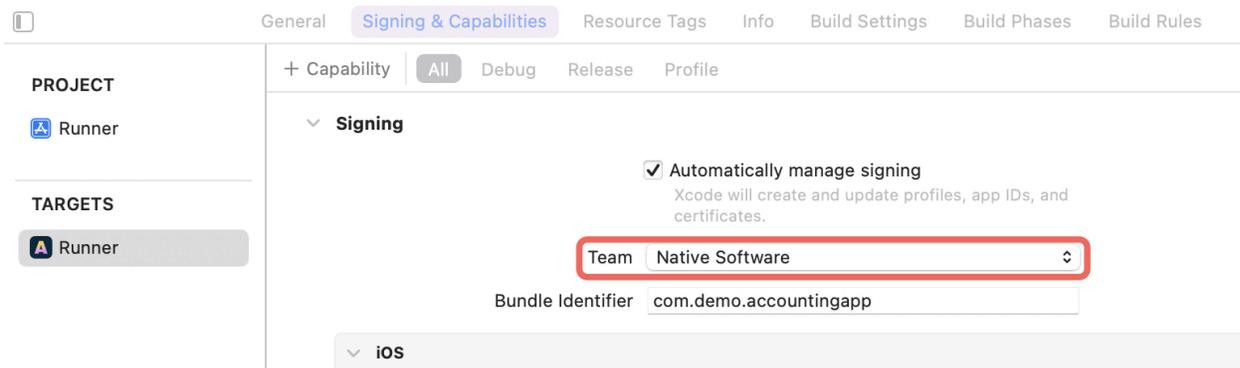
```
open ios/Runner.xcworkspace
```

- b. Select the Runner target in the Xcode navigator to view your app’s settings.

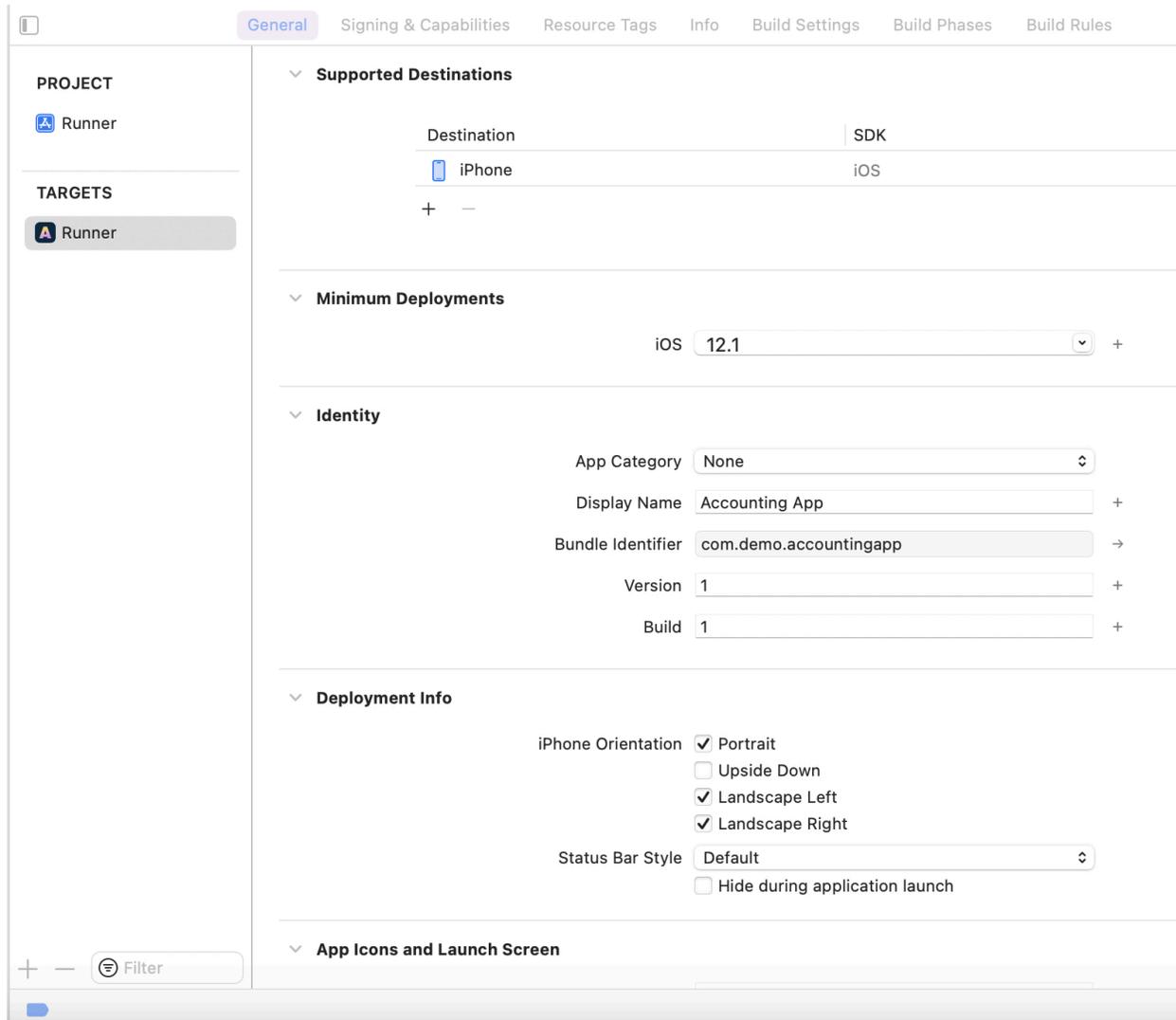
2. Verify the most important settings

- a. In the Identity section of the General tab

- i. **Display Name** (The display name of your app.)
 - ii. **Bundle Identifier** (The App ID you registered on App Store Connect.)
- b. In the Signing & Capabilities tab
- i. **Automatically manage signing** (Xcode should automatically manage app signing and provisioning. This is set true by default, which should be sufficient for most apps. For more complex scenarios, see the [Code Signing Guide](#))
 - ii. **Team** (Select the team associated with your registered Apple Developer account. If required, select Add Account..., then update this setting.)



- c. In the deployment section of the build settings tab:
- i. iOS Deployment Target
 1. The minimum iOS version that the app supports is 11.0.
 2. The General tab of your project settings should resemble the following:

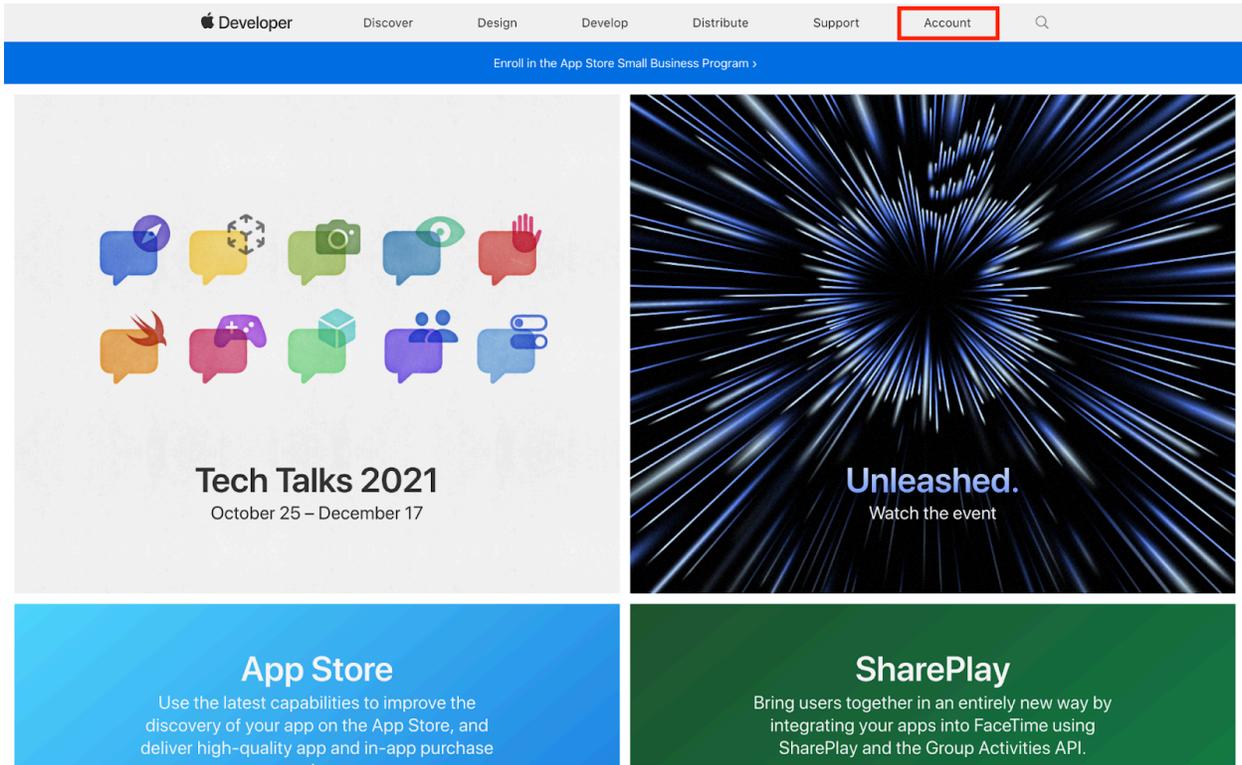


- For a detailed overview of app signing, see [Create, export, and Delete signing certificates](#).

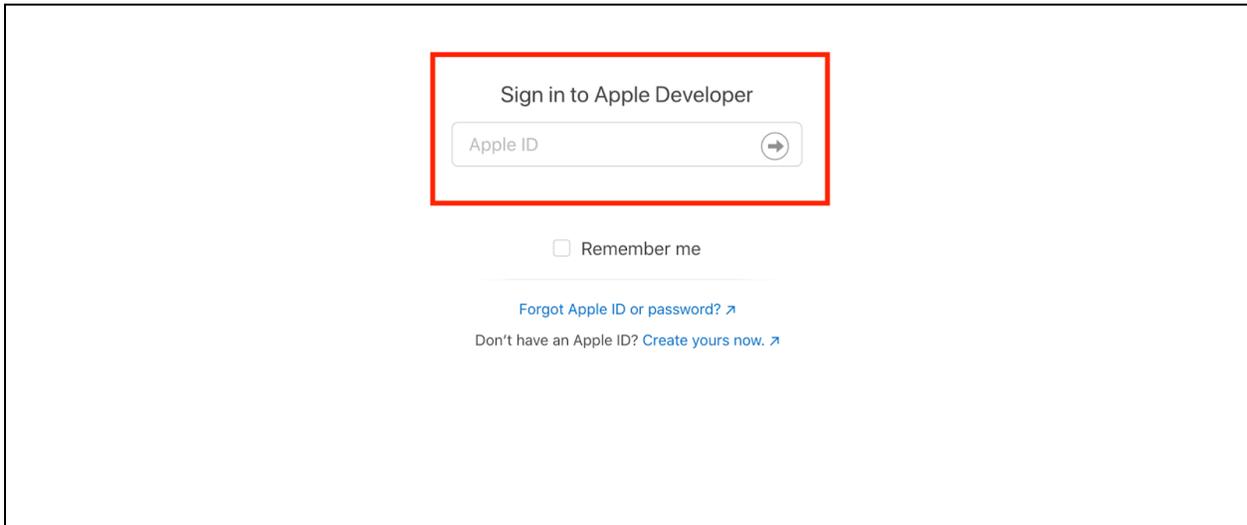
xi. Upload your APNs authentication key

If you don't already have an APNs authentication key, make sure to create one.

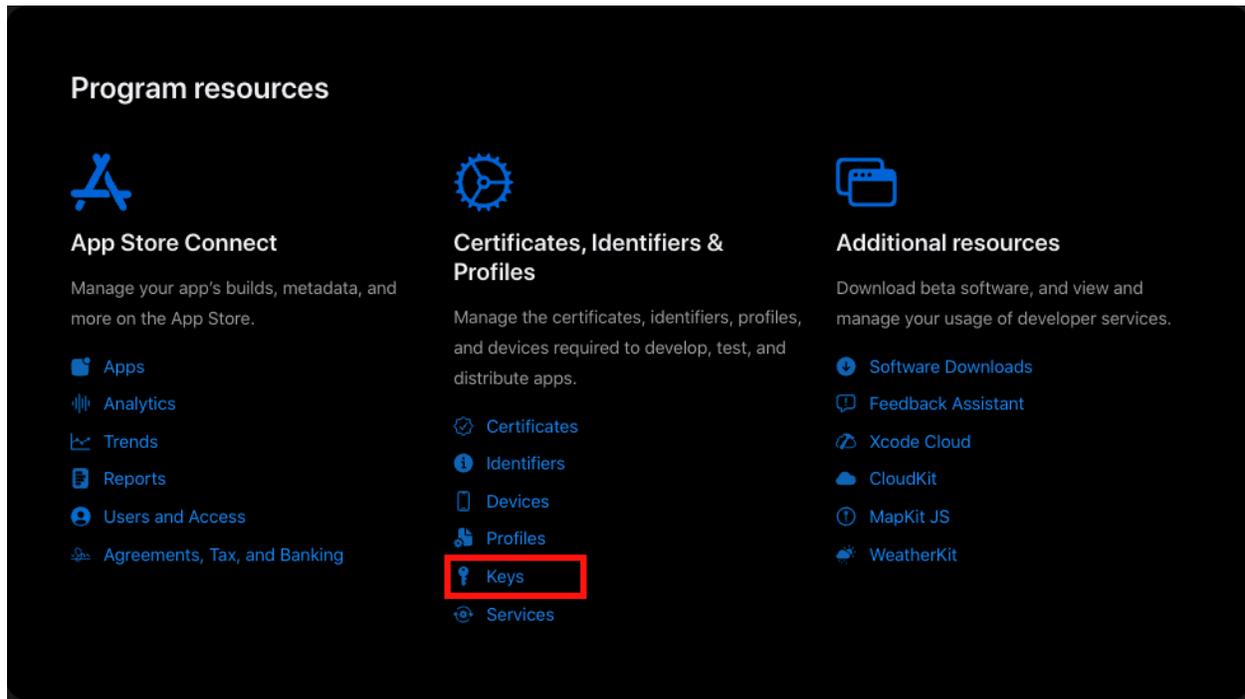
1. Go to <https://developer.apple.com> and click Account



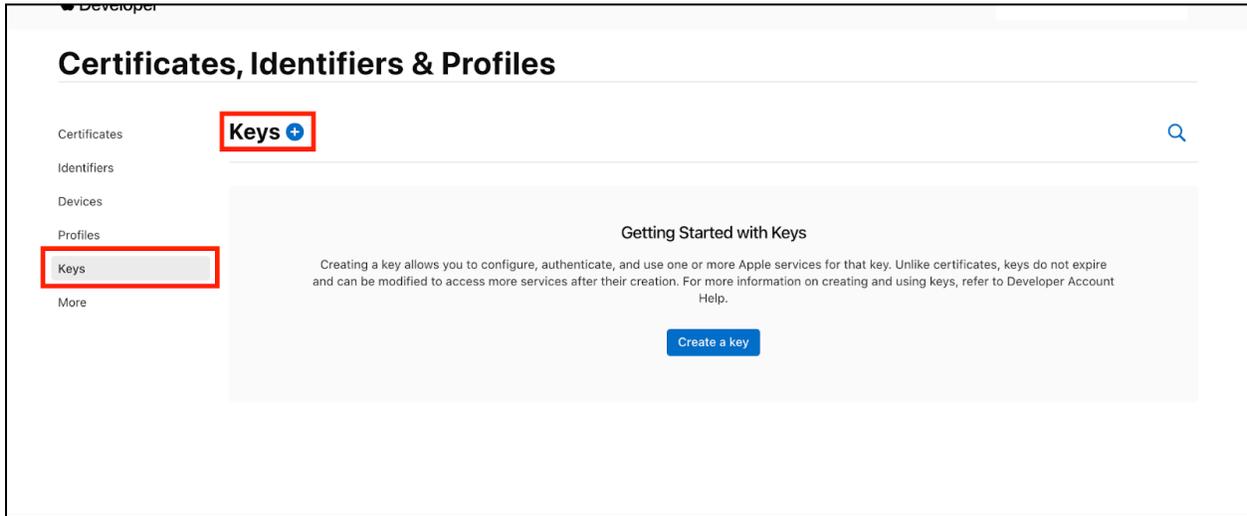
2. Log in with your Apple Developer account



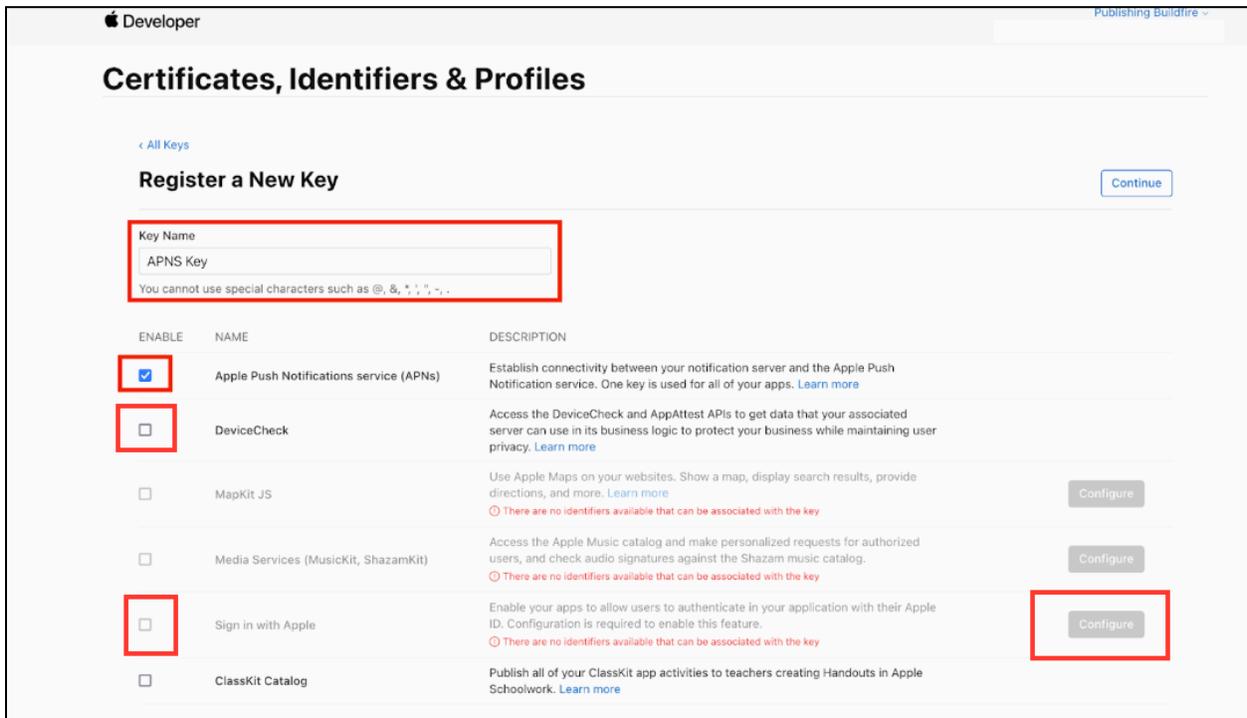
3. Click on Certificates, IDs & Profiles



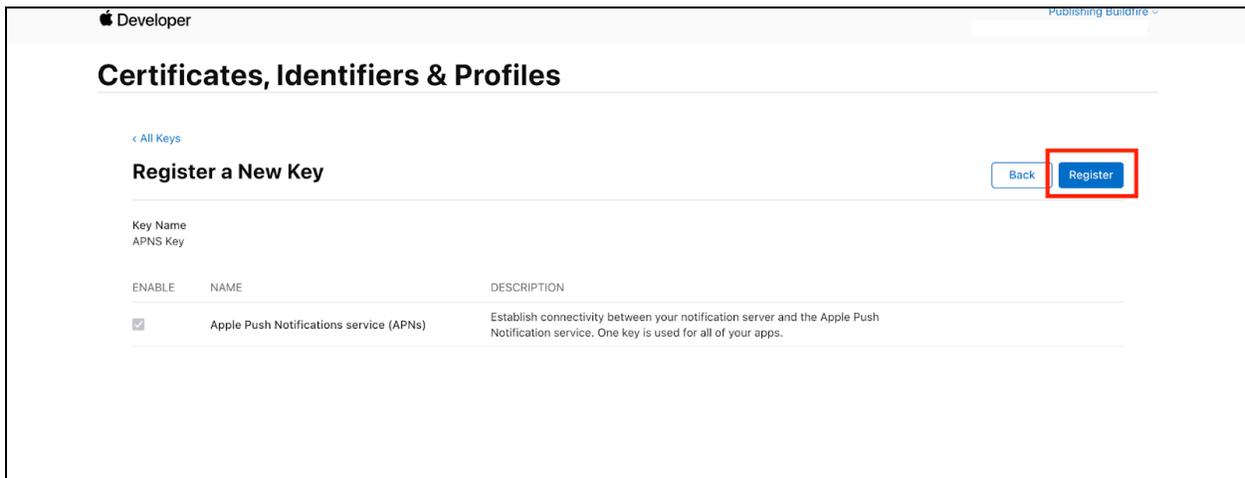
- Click on Keys and add a new key by clicking on the blue plus (+) icon next to the title Keys.



- On the next page, enter 'APNS Key' in the Key Name field and click the checkbox to enable Apple Push Notifications service (APNs), Device Check and Sign in with Apple. Also configure the Sign in with Apple.

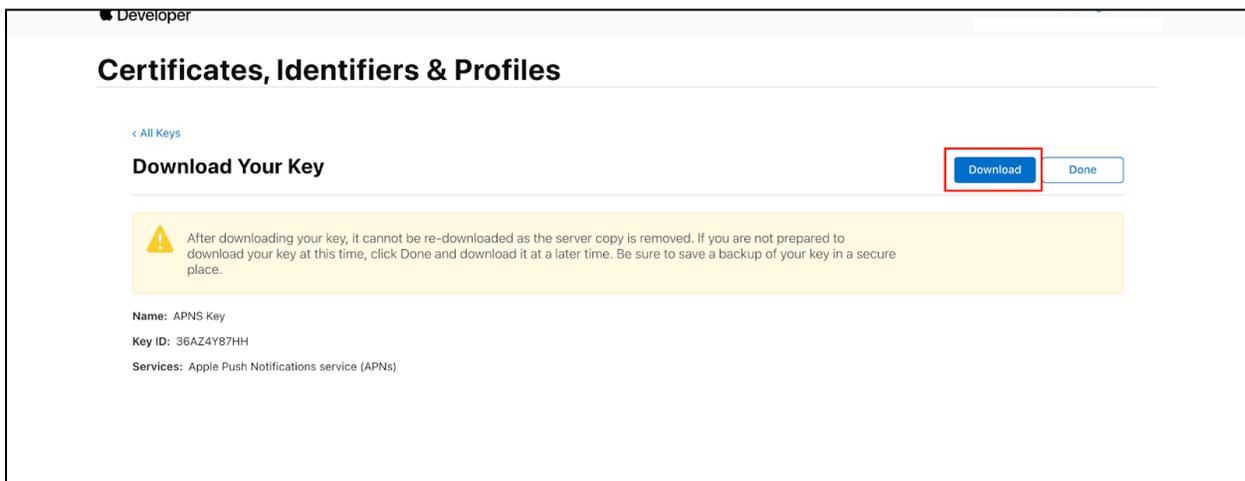


6. Click Register



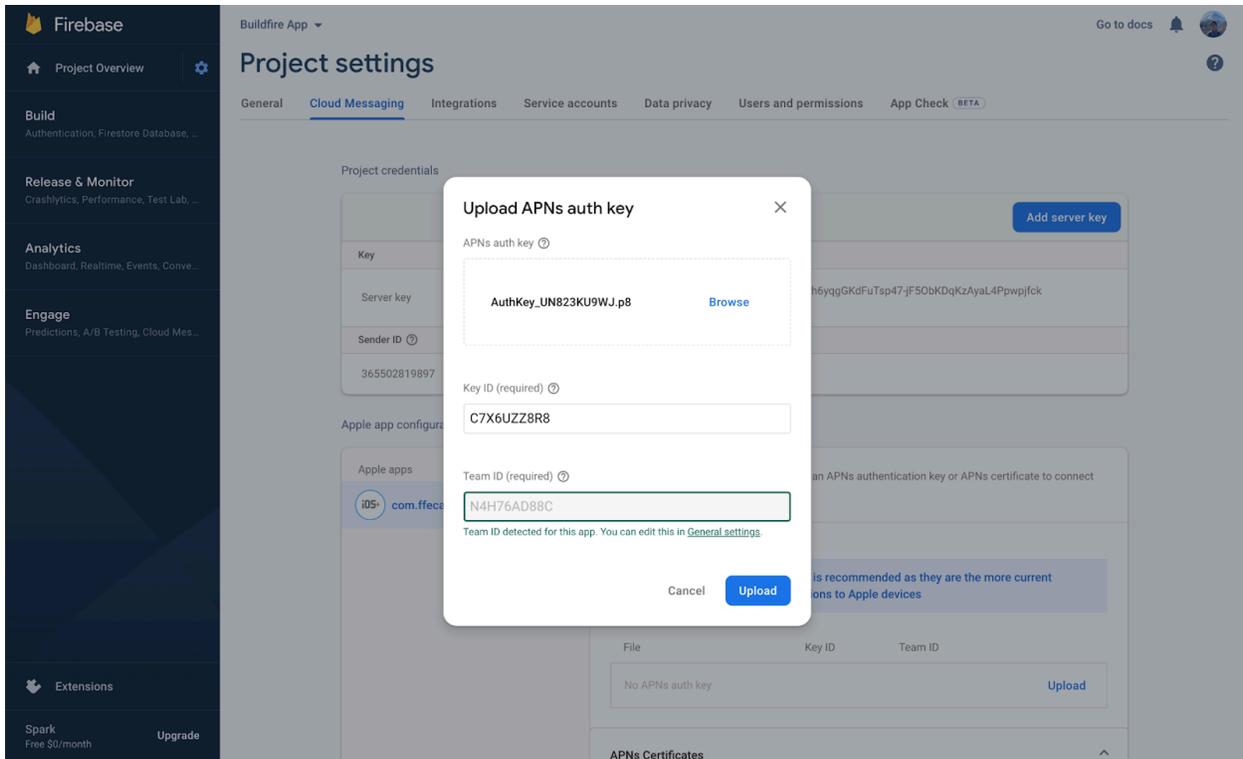
7. Click Download. This will download the APNs key that you will upload to Firebase. Please keep this page open to obtain the Key ID and Team ID for Firebase.

NOTE: Once the key has been downloaded, it cannot be retrieved again.

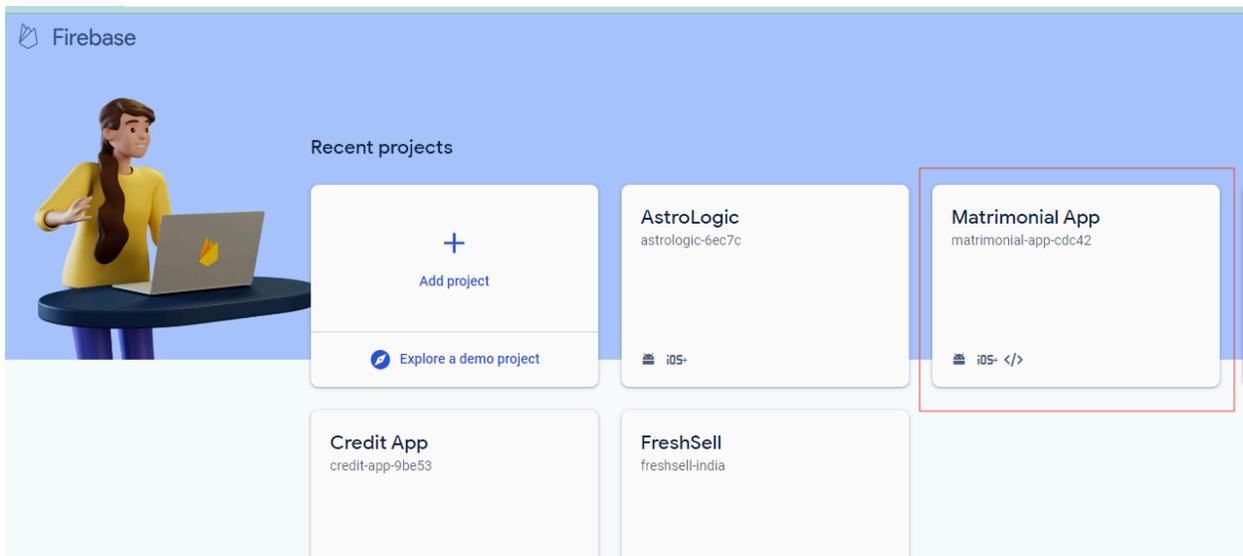


8. Now that you have the APNS key downloaded, you will need to upload this to Firebase. Open up a new browser tab or window and navigate to <https://console.firebase.google.com/>

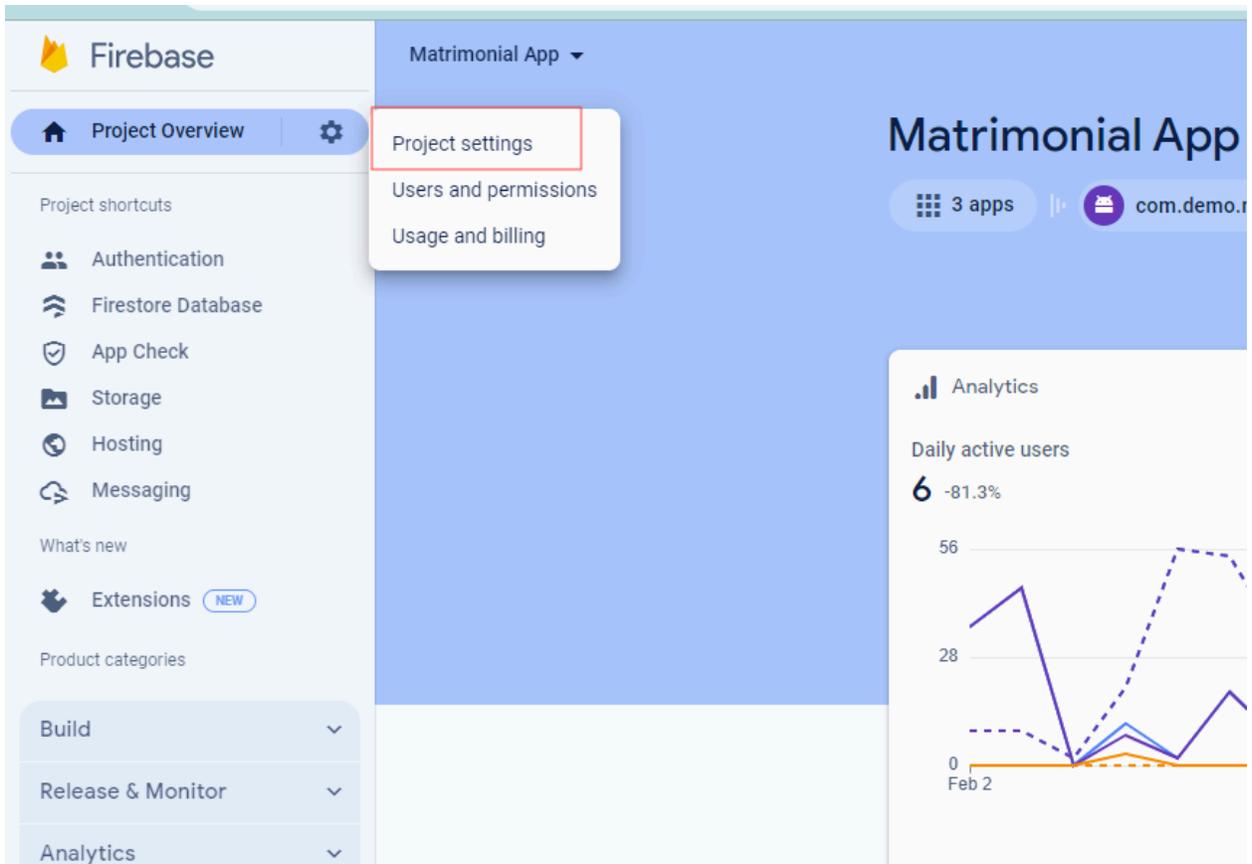
NOTE: The next few steps require you to have an iOS Firebase Certificate. If you have not done this yet, please check out our [iOS Firebase Certificate](#) article before continuing.



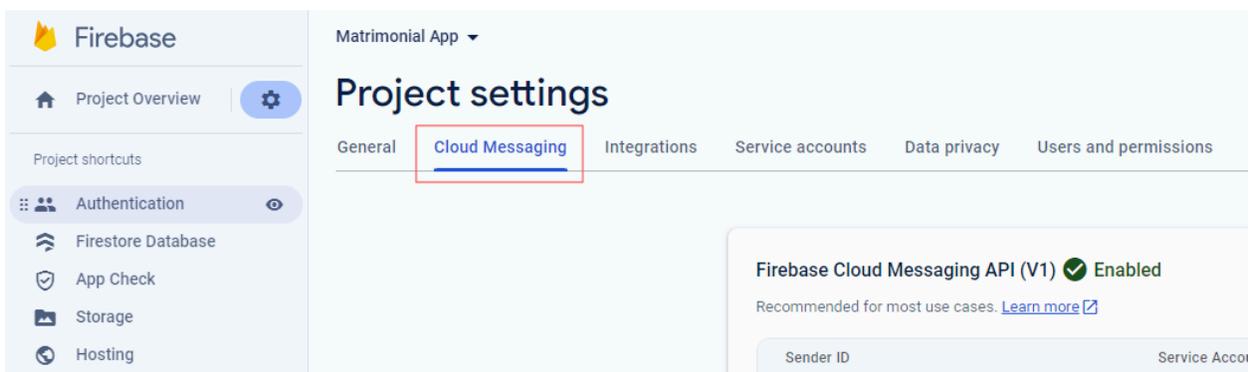
9. Click on your App project

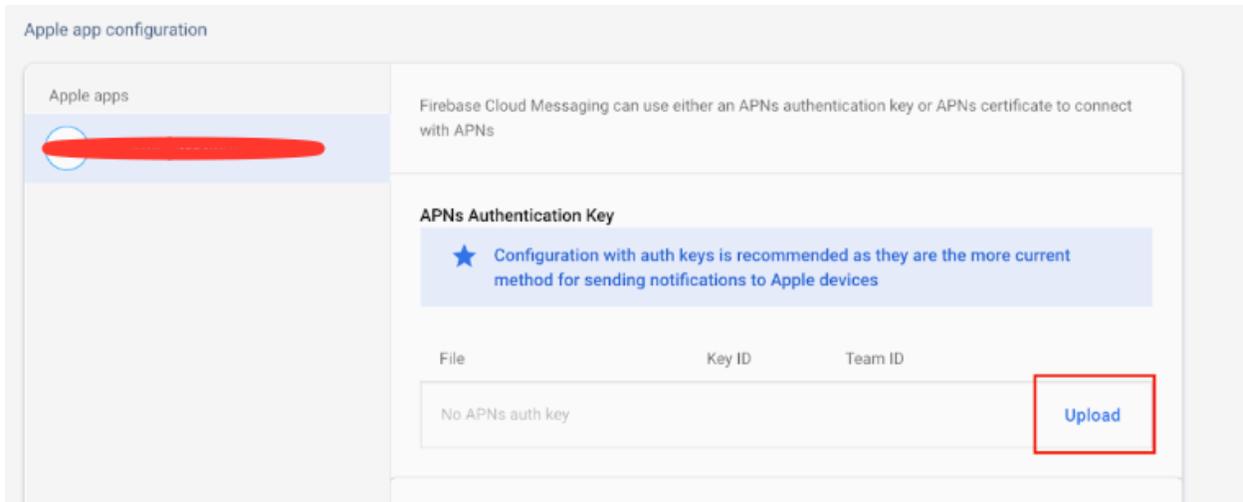


10. On the top left, click on the gear icon on the right side of Project Overview and select Project Settings

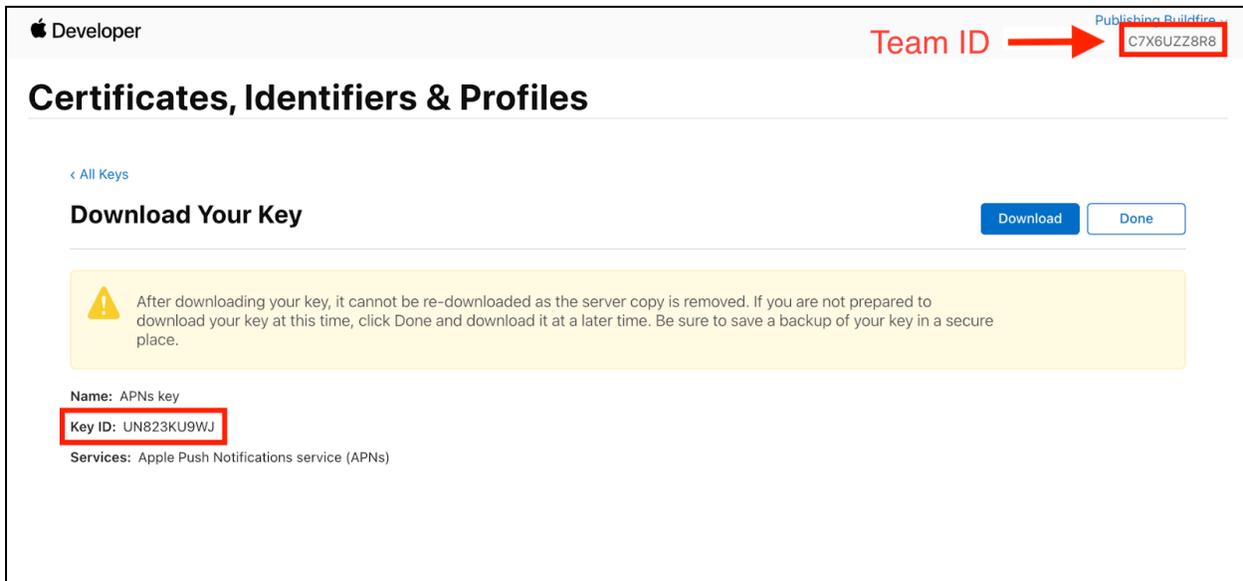


11. Click on the Cloud Messaging tab and in the Apple app configuration section, click Upload

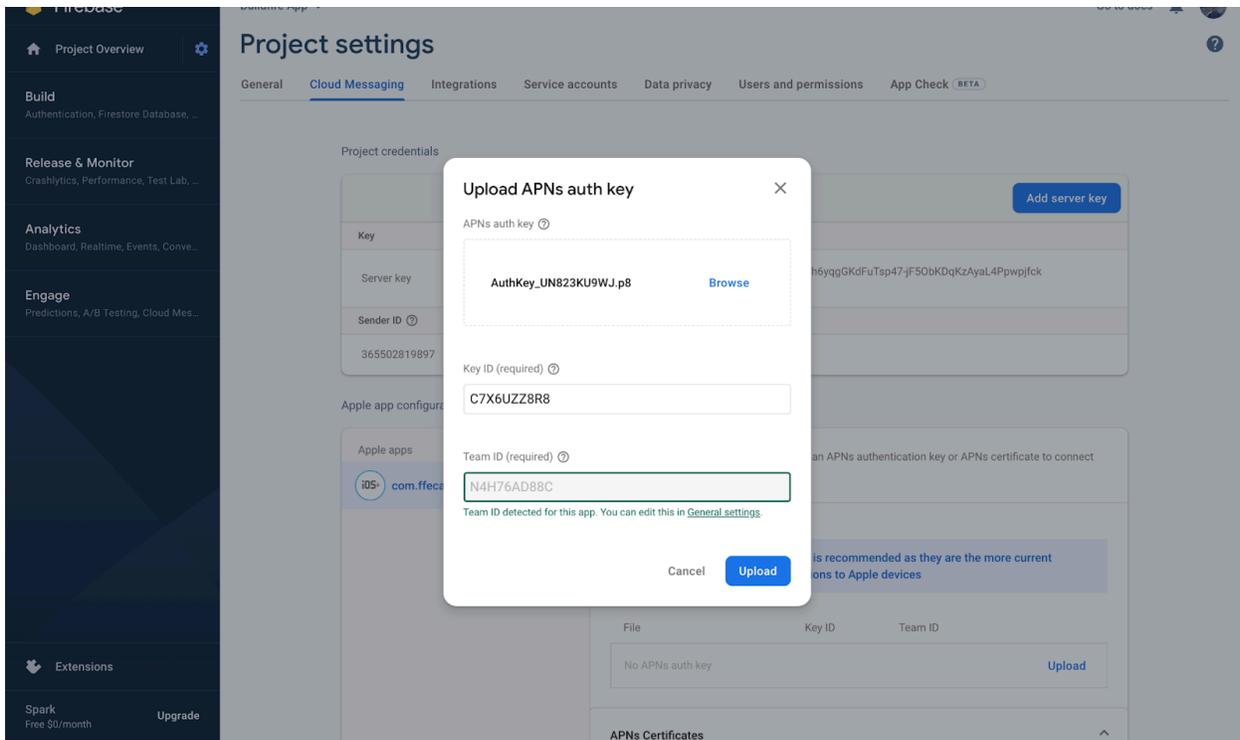




- 12. Here you will upload the key file by clicking Browse. Select the file that ends with .p8 that was downloaded in the previous steps. The file name will look like this: AuthKey_UN823KU9WJ.p8
- 13. Now you will have to copy the Key ID and Team ID by going back to your Apple Developer account. The Key ID is located below the key name and the Team ID is located in the top right corner, next to your Apple developer name.



- 14. Go back to the Firebase page and copy and paste the Key ID and Team ID. Lastly, click on the Upload button.



j. Configure the Firebase setting to the Project

Go to the `lib\firebase_option.dart` file

- For Android settings replace your credentials in the android method

```
static const FirebaseOptions android = FirebaseOptions(  
  apiKey: "Your key",  
  authDomain: "*Your Firebase Project Id*.firebaseapp.com",  
  projectId: "Your Firebase Project Id",  
  storageBucket: "*Your Firebase Project Id*.appspot.com",  
  messagingSenderId: "Your message Sender Id",  
  appId: "Your Firebase App Id",  
);
```

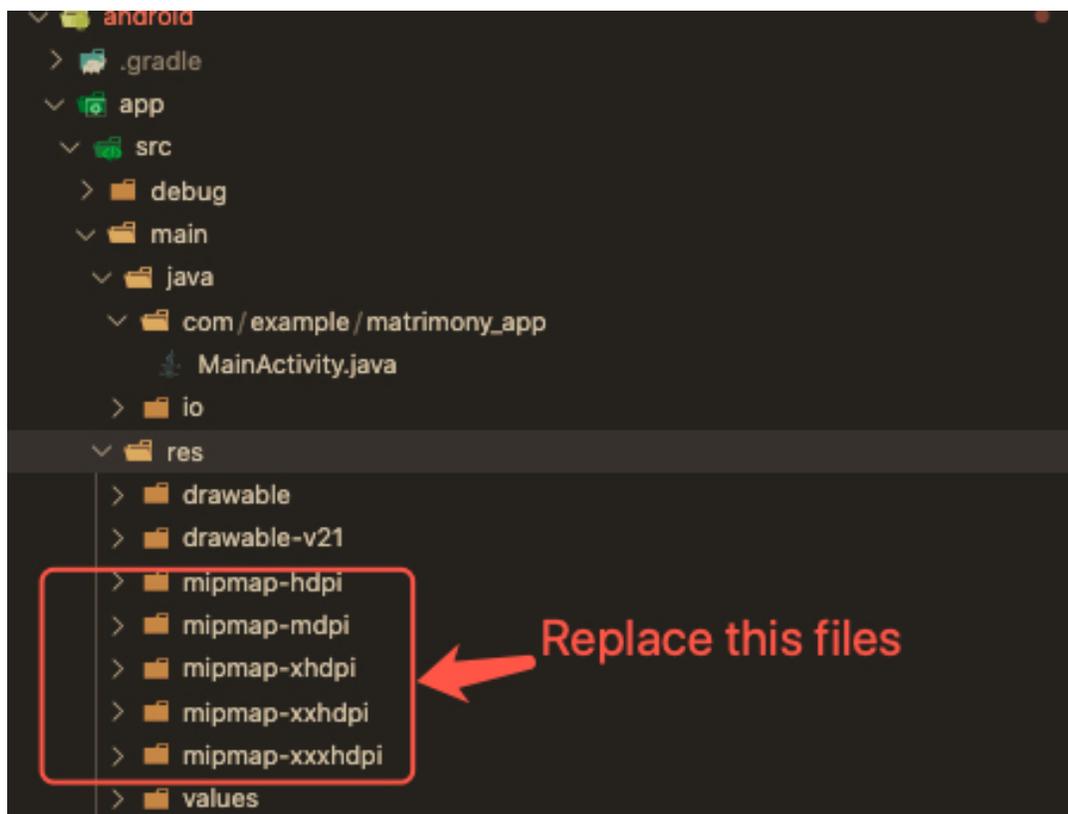
- For iOS settings replace your credentials in the ios method

```
static const FirebaseOptions ios = FirebaseOptions(  
    apiKey: "Your key",  
    authDomain: "*Your Firebase Porject Id*.firebaseapp.com",  
    projectId: "Your Firebase Porject Id",  
    storageBucket: "*Your Firebase Porject Id*.appspot.com",  
    messagingSenderId: "Your message Sender Id",  
    appId: "Your Firebase App Id",  
    androidClientId: 'Your Android Client Id',  
    iosClientId: 'Your Ios Client Id',  
    iosBundleId: 'com.demo.matrimonyapp',  
);
```

k. Change App Icon

i. For Android

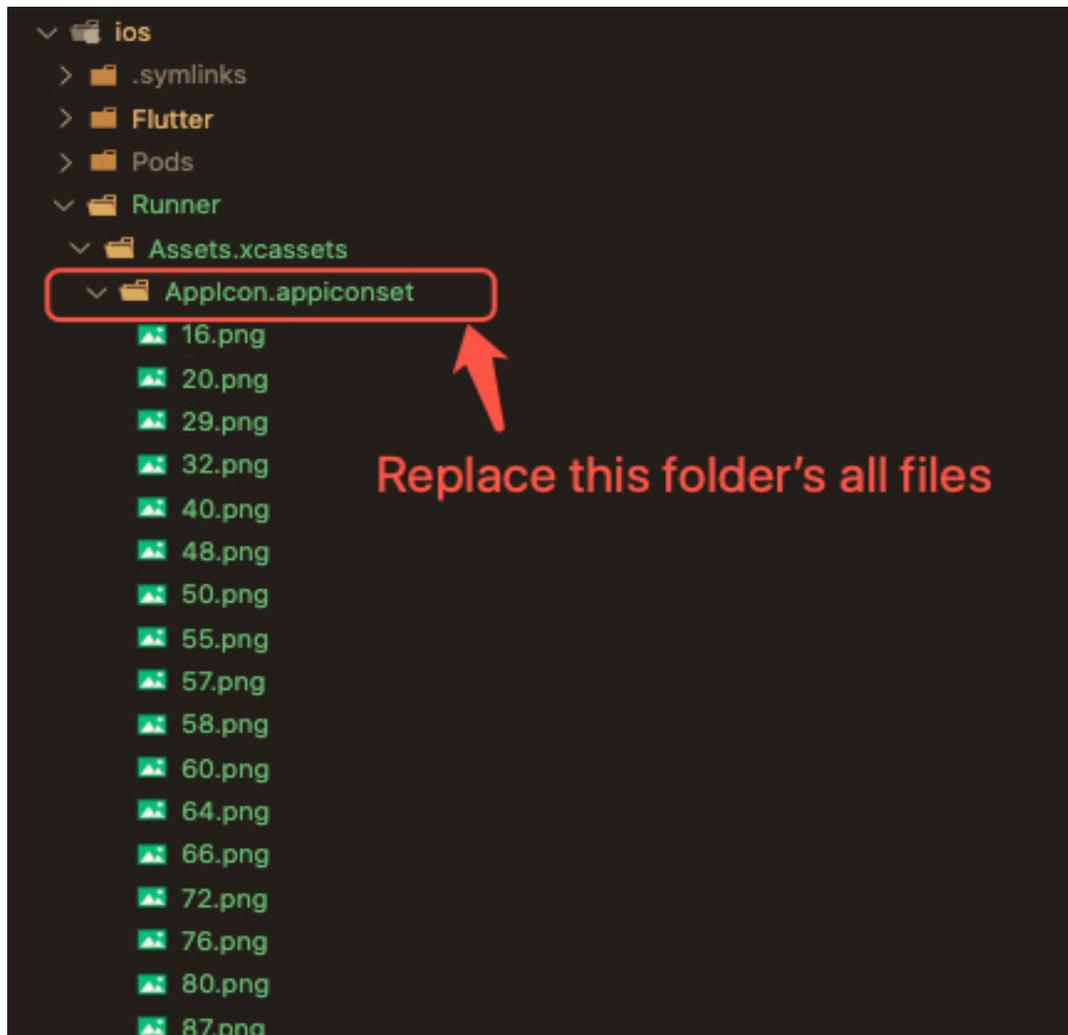
Replace the icons in the **android\app\src\main\res** folder as shown in the below image.



ii. For iOS

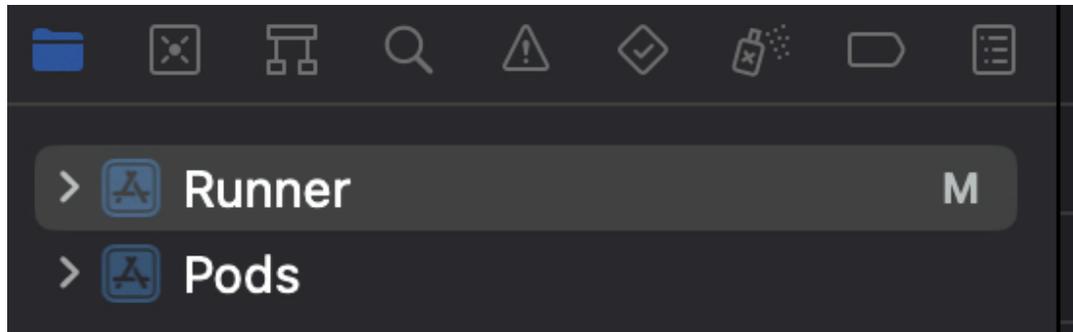
1. Replace the icons in the below folder as shown in the below image

ios\Runner\Assets.xcassets\AppIcon.appiconset

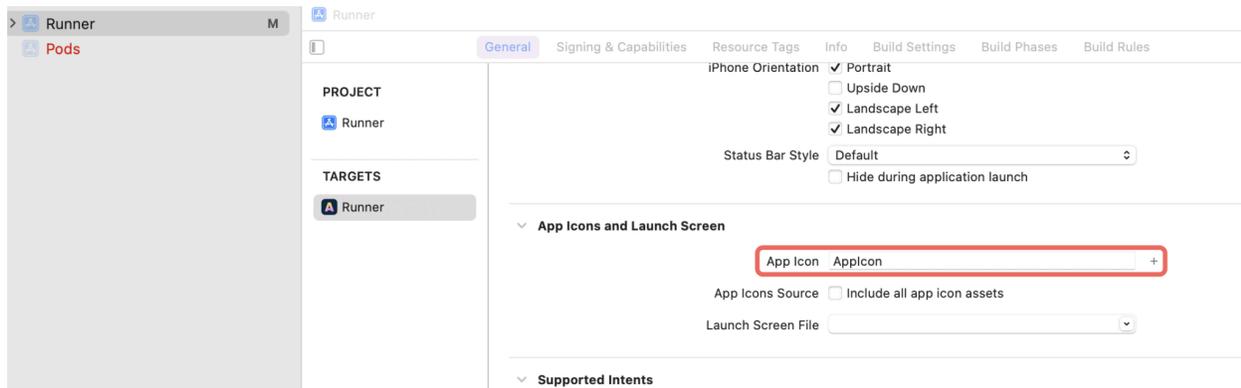


2. Change icons using XCode

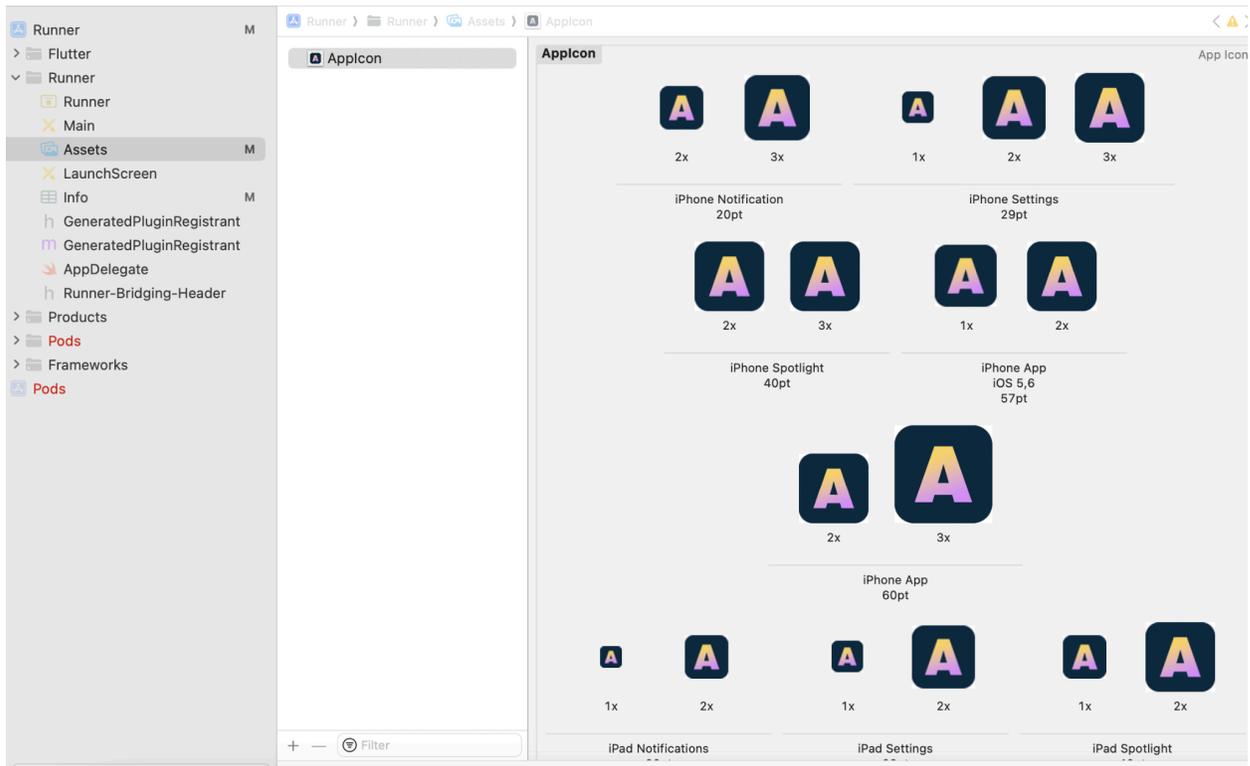
- a. Right-click on the iOS folder Choose Open in Xcode Option
- b. Click on the folder icon left side of the XCode window



- c. Select Runner.
- d. Select Target runner
- e. Go to App Icons And Launch Images
- f. Click the right arrow button of the app icons source



g. Replace all the icons according to their size



NOTE:

- If you want to generate the App icon bundle from any image you have, you can generate it from publicly available websites like

<https://www.appicon.co/>

I. Build Release for Android

- Open Project in VS Code
- In Terminal Execute the below commands

```
flutter clean  
flutter pub get  
flutter build apk --release
```

- After making the release, to generate the release bundle Execute the below command

```
flutter build appbundle --release
```

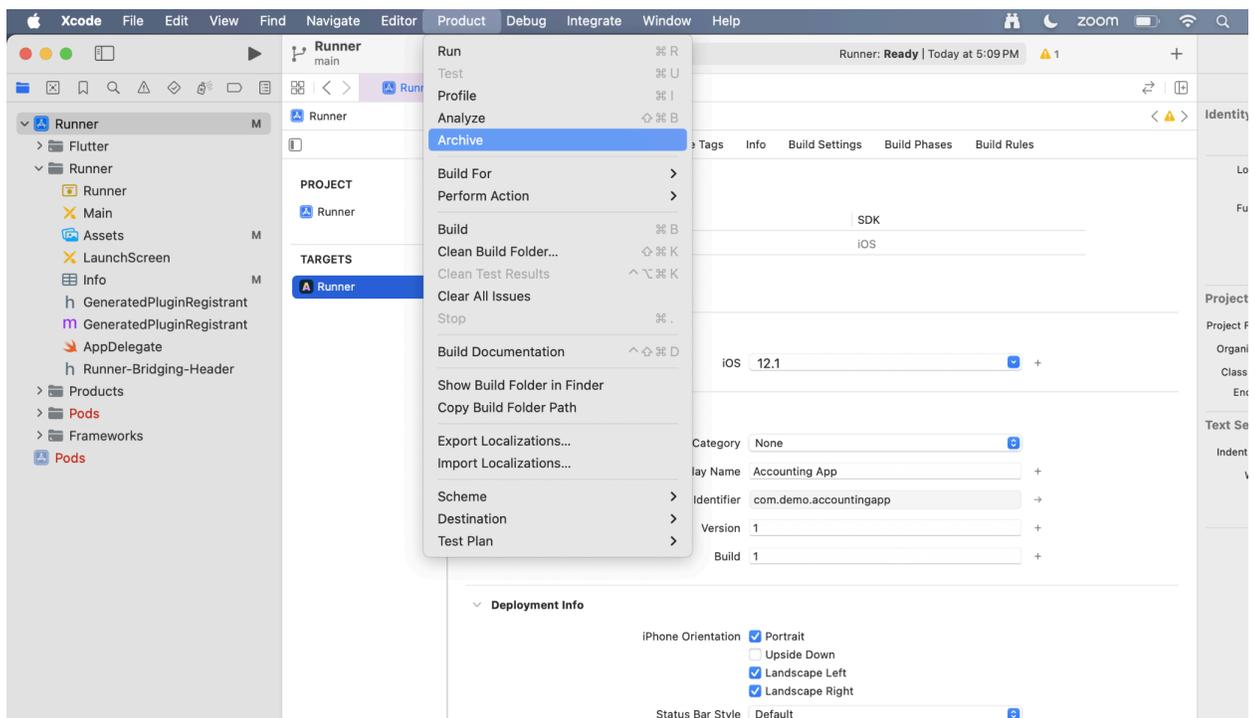
iv. Get the APK from the below path

```
build\app\outputs\flutter-apk\app-release.apk
```

m. Build Release for iOS

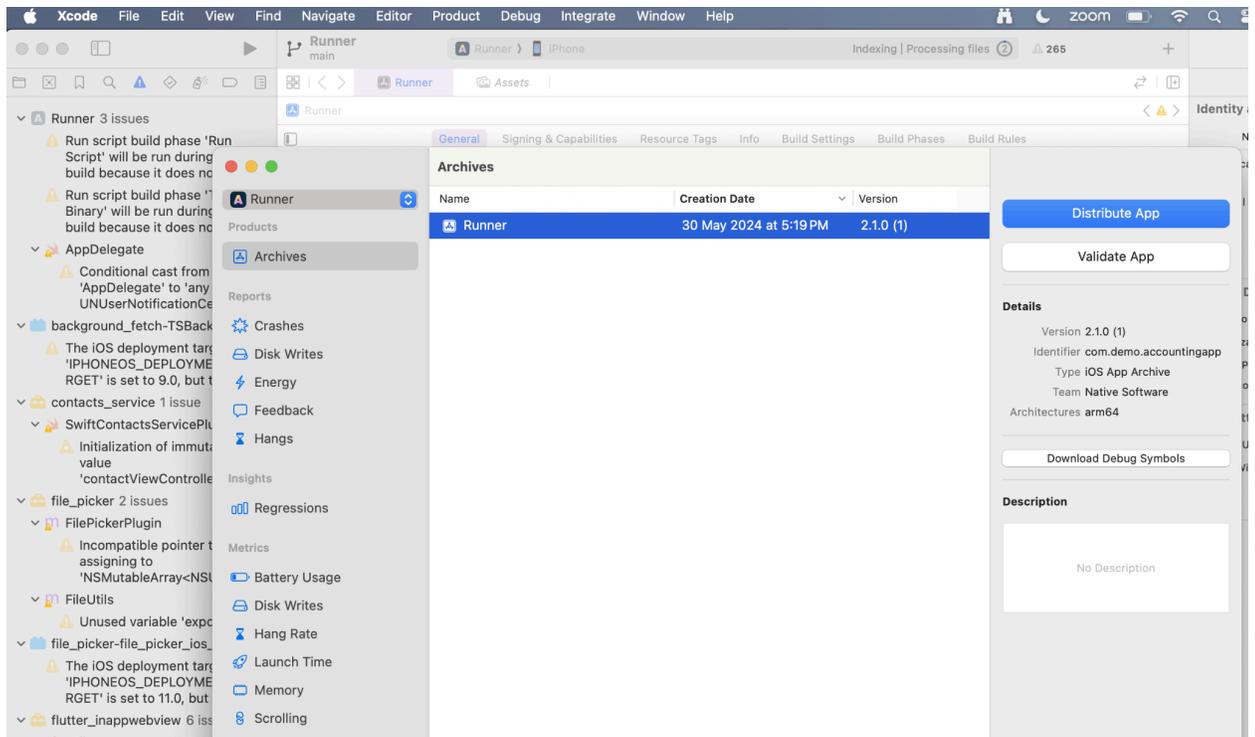
i. Open Project in XCode

ii. Select **Archive** from the **Product Menu**



iii. After successfully archiving select the **Organizer** option from the **Windows menu**

iv. After clicking on it opens one popup for Archive, Click on the **Distribute App Button**



- v. After successfully done, you can upload this app to your Apple developer account in the TestFlight
- vi. To publish your app from TestFlight please follow [this link](#)

n. Other Options for the Advanced User

- i. Paths to the images used in the app

Images	Path	Screen Path
About Screen	assets/images/logo.png	lib/screen/aboutScreen.dart
Splash Screen	assets/images/logo.png	lib/screen/splashScreen.dart
Backup & Restore Screen	assets/images/googleicon.png	lib/screen/backupAndRestoreScreen.dart
Lock Screen	assets/images/logo.png	lib/screen/lockScreen.dart
Other Apps Screen	assets/images/eventdecor144x144.png	lib/screen/otherAppsScreen.dart
	assets/images/securewall144x144.png	lib/screen/otherAppsScreen.dart
	assets/images/wordpuzzle128x128.png	lib/screen/otherAppsScreen.dart
	assets/images/kidsnursery144x144.png	lib/screen/otherAppsScreen.dart
	assets/images/yoga4me144x144.png	lib/screen/otherAppsScreen.dart
	assets/images/kidsabcd144x144.png	lib/screen/otherAppsScreen.dart
	assets/images/wishkarts150x150.png	lib/screen/otherAppsScreen.dart
	assets/images/easycounter144x144.png	lib/screen/otherAppsScreen.dart

ii.

iii. Colors used in the app. If you want to change the colors you can make the changes in the file **lib/Theme/nativeTheme.dart**

#	Color code
Primary Color	0xFF0076bc
Primary Color Dark	0xFF000000
Primary Color Light	0xFF7c7e7d
Primary Icon Theme Color	0xFF0076bc
Canvas Color	white
Scaffold Background Color	white
Card Theme Shadow Color	grey
AppBar theme - color	0xFF0076bc

iv. Packages used in the app are listed below. You can find them in **pubspec.yaml** file.

Package Name - Version	Description
package_info_plus: ^8.0.0	Check here
webview_flutter: ^4.7.0	Check here
shared_preferences: ^2.2.3	Check here
url_launcher: ^6.2.6	Check here
sqlite_sqlcipher: ^3.0.0	Check here
material_design_icons_flutter: 7.0.7296	Check here
http: ^1.2.1	Check here
email_validator: 2.1.17	Check here
font_awesome_flutter: ^10.7.0	Check here
image_cropper: ^6.0.0	Check here
flutter_mailer: ^2.1.2	Check here
ribbon_widget: ^1.0.5	Check here
printing: ^5.12.0	Check here
page_transition: ^2.1.0	Check here
number_to_words: 1.0.0	Check here
mobile_number: 2.1.1	Check here
local_auth: ^2.2.0	Check here
open_file: 3.3.2	Check here
mailer: 6.0.1	Check here
ulid: 2.0.0	Check here
badges: 3.1.2	Check here
encrypt: 5.0.3	Check here
flutter_local_notifications: ^17.1.2	Check here
device_info_plus: ^10.1.0	Check here

fl_chart: ^0.68.0	Check here
flutter_randomcolor: ^1.0.15	Check here
contacts_service: 0.6.3	Check here
multi_image_picker_plus: ^0.0.4	Check here
intro_slider: 4.2.1	Check here
permission_handler: ^11.3.1	Check here
google_sign_in: ^6.2.1	Check here
googleapis: ^13.1.0	Check here
image_picker: ^1.1.1	Check here
share_plus: ^9.0.0	Check here
file_picker: ^8.0.3	Check here
connectivity_plus: ^6.0.3	Check here
intl: ^0.18.0	Check here
path_provider: ^2.1.3	Check here
flutter_inappwebview: ^5.8.0	Check here
background_fetch: ^1.3.4	Check here
in_app_update: ^4.2.2	Check here
archive: ^3.5.1	Check here
pdf: ^3.10.8	Check here
image_picker_for_web: ^3.0.4	Check here
universal_html: 2.2.4	Check here
flutter_html_to_pdf: ^0.7.0	Check here

USEFUL LINKS

- For more information on iOS refer to [this link](#)

This document was last updated on 30 May 2024.