

App/Website/Admin panel Reskin Guidelines

Create a unique bundle identifier for the iOS app

Go to developer.apple.com and login using credentials

Follow the instructions given in the link below to create bundle identifier

<https://www.appcodezip.com/2020/05/app-id.html>

Keep note of bundle identifier as we need it in firebase ios app setup and flutter ios app setup

Create new android app on google developer play console

Go to https://play.google.com/console and login using credentials

Create your app using the new package name

Keep note of package name as we need it in firebase android app setup and flutter ios app setup

Firebase setup

1. Login into your Firebase account and create an app if it's not created yet.
2. Create an app if its not created

Chrome File Edit View History Bookmarks Profiles Tab Window Help

console.firebaseio.google.com/u/0/

Create a project(Step 1 of 3)

Let's start with a name for your project[®]

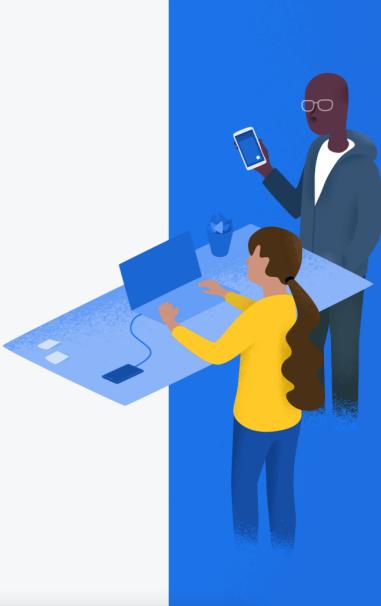
Project name

Musicy

musicy-46533

I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft or profession.

Continue



Wednesday, June 8, 2023 4:56 PM

Chrome File Edit View History Bookmarks Profiles Tab Window Help

console.firebaseio.google.com/u/0/

Create a project(Step 2 of 3)

Google Analytics for your Firebase project

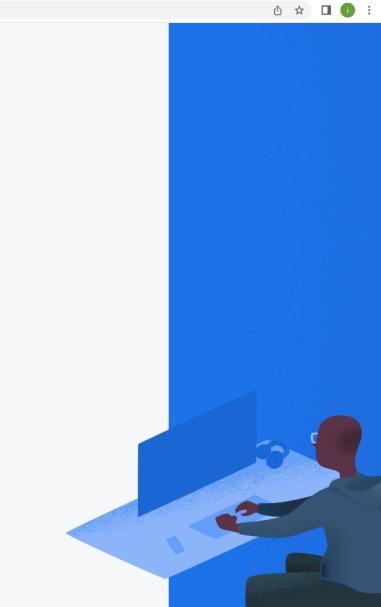
Google Analytics is a free and unlimited analytics solution that enables targeting, reporting and more in Firebase Crashlytics, Cloud Messaging, in-app messaging, Remote Config, A/B Testing and Cloud Functions.

Google Analytics enables:

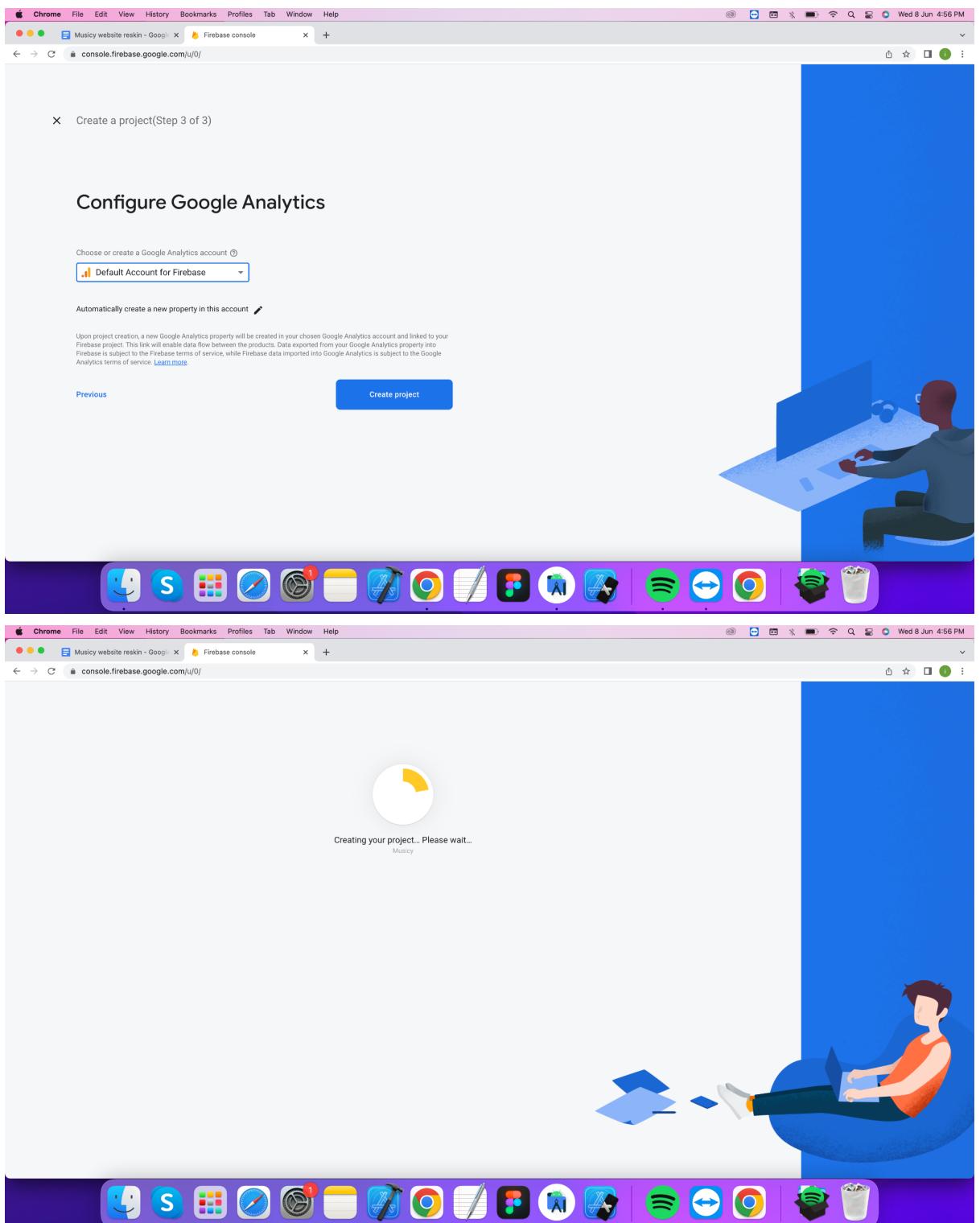
- A/B testing
- User segmentation and targeting across Firebase products
- Crash-free users
- Event-based Cloud Functions triggers
- Free unlimited reporting

Enable Google Analytics for this project
Recommended

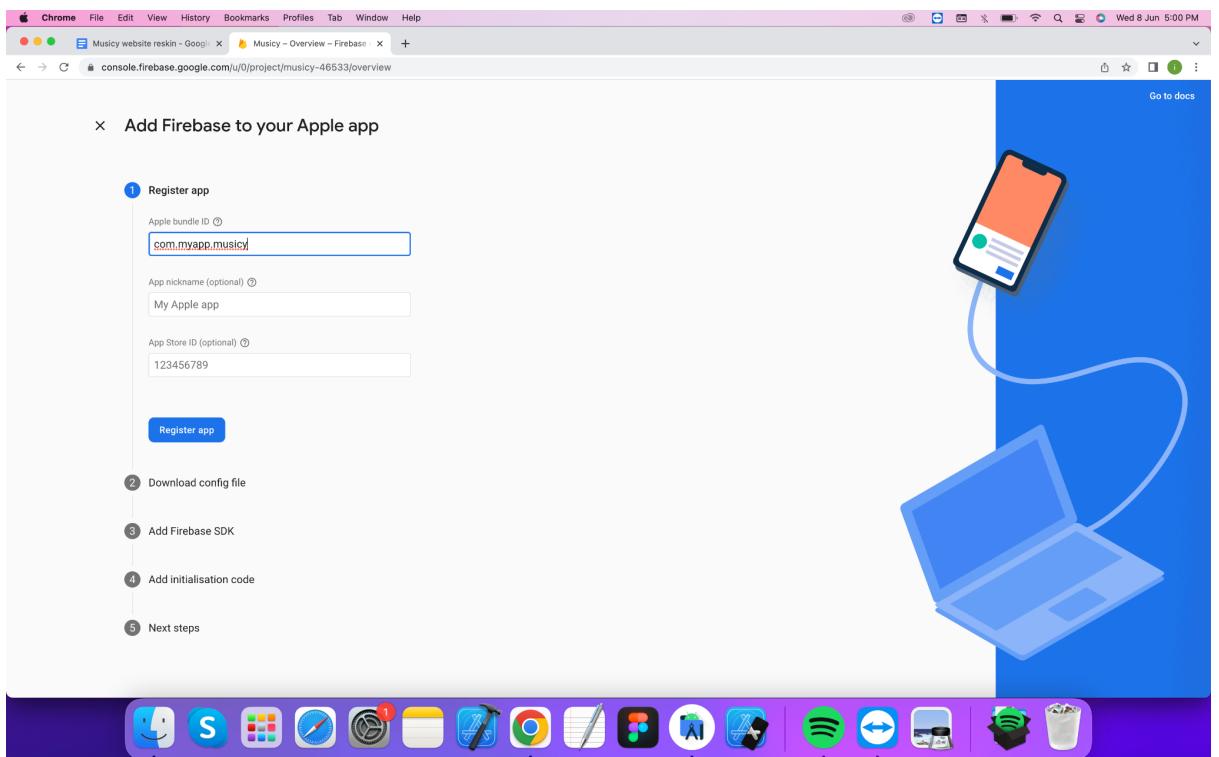
Previous Continue



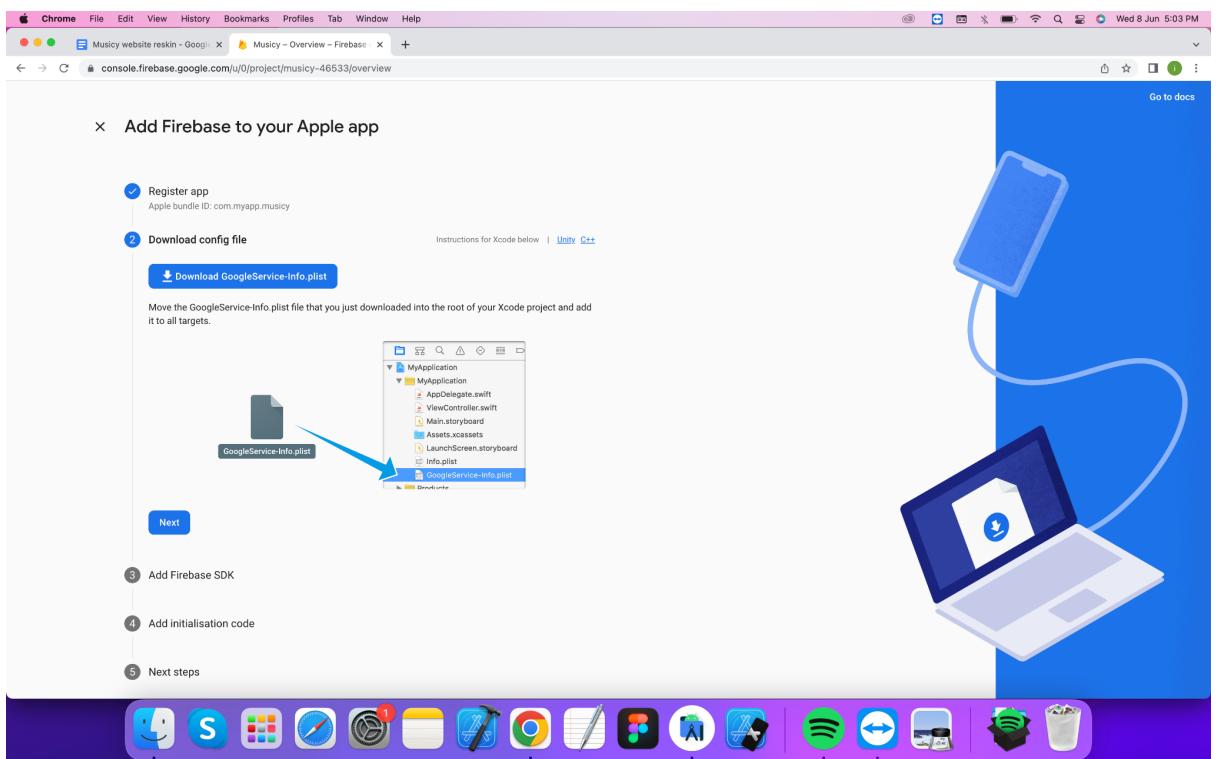
Wednesday, June 8, 2023 4:56 PM



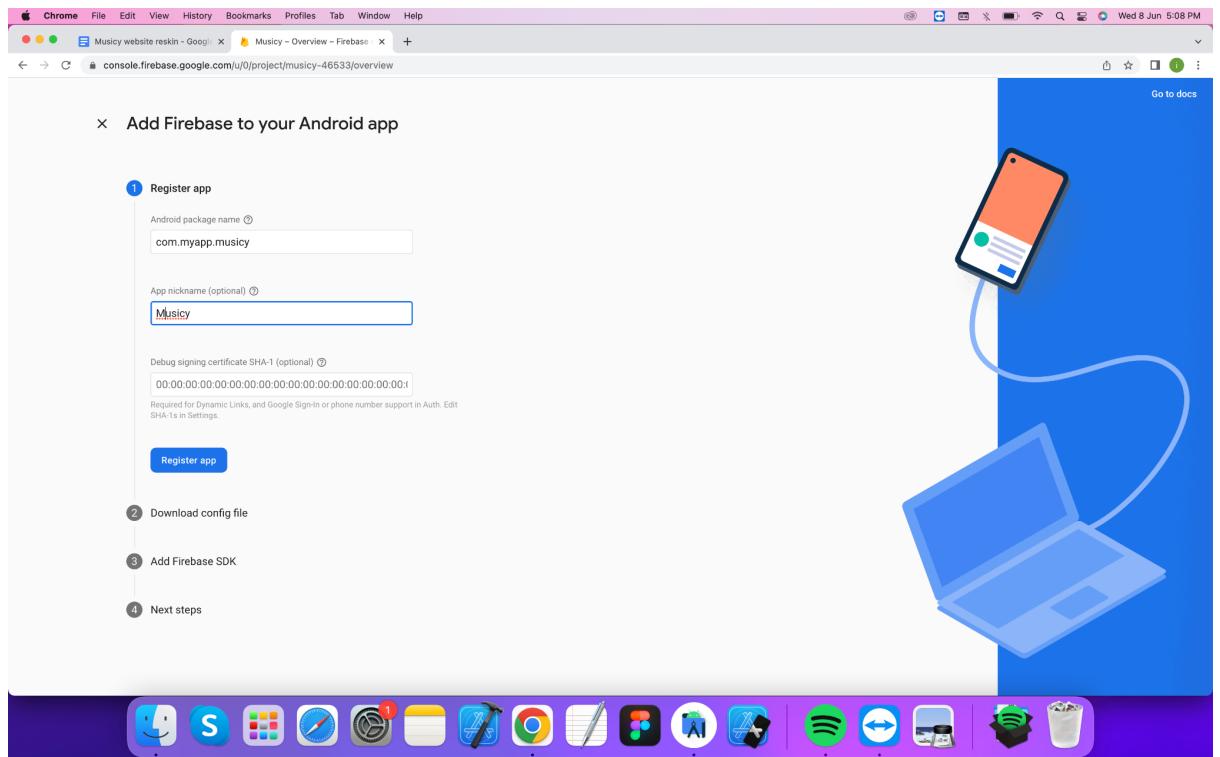
3. After creating firebase project , create ios,android and web app one by one as shown in below screenshots
4. First go it iOS app
5. Add ios bundle identifier that you have created in iOS app setup in apple developer account above, Enter app name also



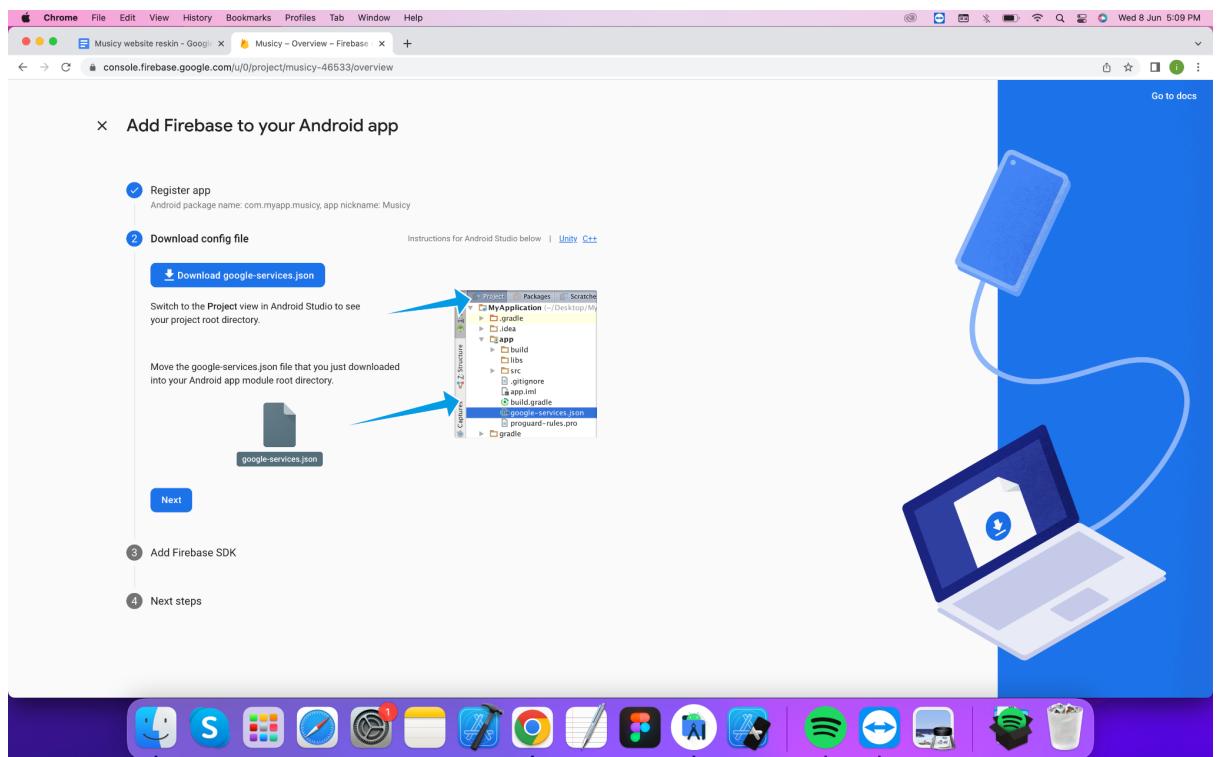
6. Click on register app and download the config plist file , This will be needed in flutter ios app reskin section



7. Now setup android app, Create on android icon
8. Enter package name that you created above



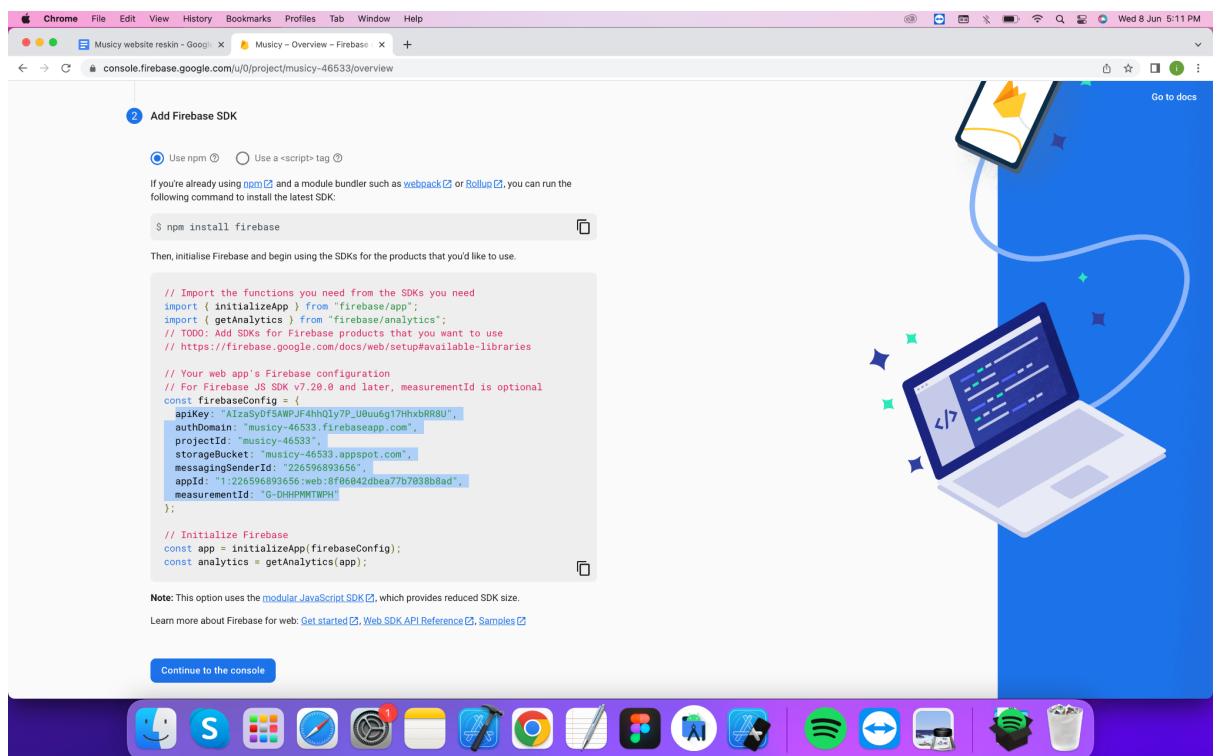
9. Click on register app and download the config.json file, This will be needed in flutter android setup later



10. Now add the web app

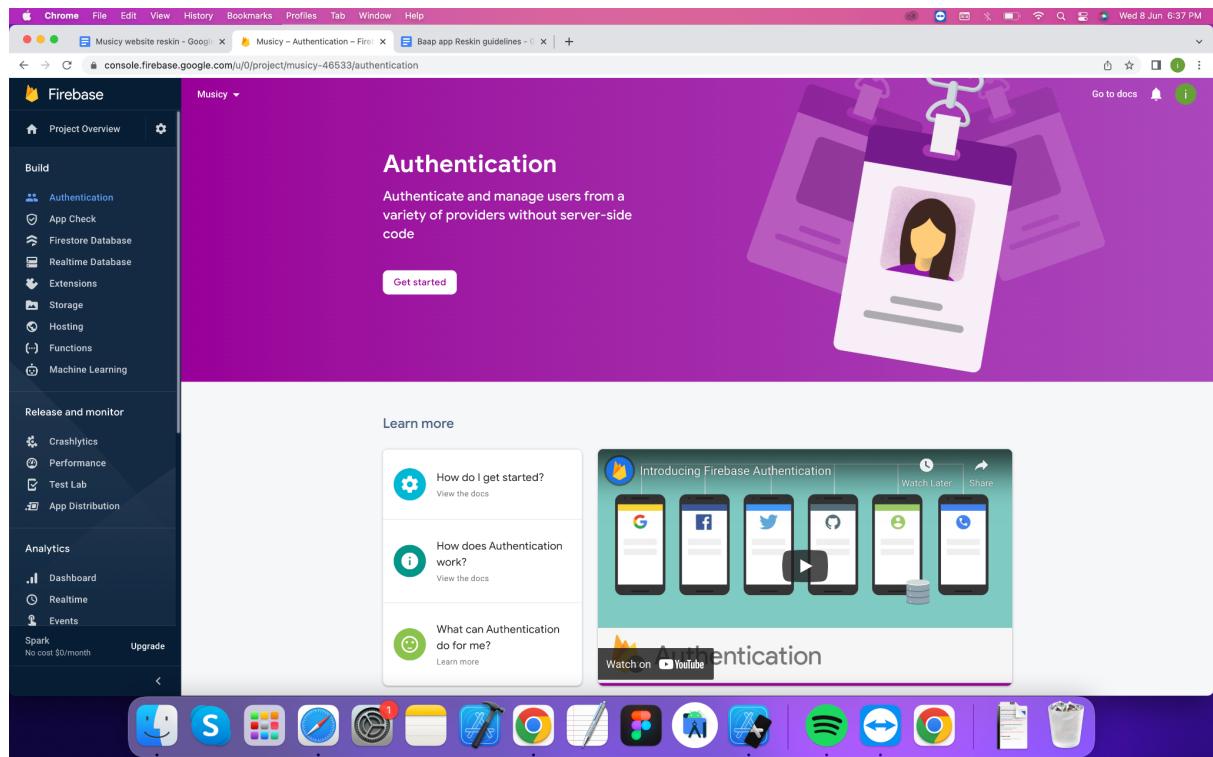
11. Enter the web app name and register the app.

12. Copy the required key values mentioned in the below screenshot. This will be needed in Admin and website setup later



Enable signin method for firebase

Go to Authentication tab from left menu and click on get started



Select email/password and enable it

The screenshot shows the Firebase Authentication interface. On the left, a sidebar lists various services: Build (Authentication, App Check, Firestore Database, Realtime Database, Extensions, Storage, Hosting, Functions, Machine Learning), Release and monitor (Crashlytics, Performance, Test Lab, App Distribution), and Analytics (Dashboard, Realtime, Events). The main area is titled 'Authentication' and has tabs for Users, Sign-in method, Templates, and Usage. Under 'Sign-in providers', the 'Email/Password' provider is listed with its 'Enable' switch turned on. Below it, there's a note about allowing users to sign up with email and password, mentioning SDKs for verification, recovery, and address change. At the bottom right of the provider card are 'Cancel' and 'Save' buttons. A large 'Authorised domains' section follows, listing 'localhost' and three project domains ('musicy-46533.firebaseio.com', 'musicy-46533.web.app') all marked as 'Default'. An 'Add domain' button is at the top right of this section. The bottom of the screen shows a Mac OS X dock with various application icons.

Select “Phone” and enable it also

This screenshot is identical to the one above, showing the 'Email/Password' provider enabled. However, the 'Phone' provider is now highlighted, indicating it has been selected. Its 'Enable' switch is also turned on. A note below the provider explains that it allows users to sign in with a mobile phone number using Firebase SDK phone verification. There is a dropdown menu for 'Phone numbers for testing (optional)'. The 'Authorised domains' section remains the same, listing the three project domains as 'Default'. The bottom of the screen shows the Mac OS X dock.

Open Firestore database menu from left menu and click on “Create database” button

Open Storage menu from left menu and click on “Get started” button

Flutter website and Admin panel reskin

Here we will guide you reskin of website only , you can repeat the same process for Admin panel also

- Create admin user in firebase**

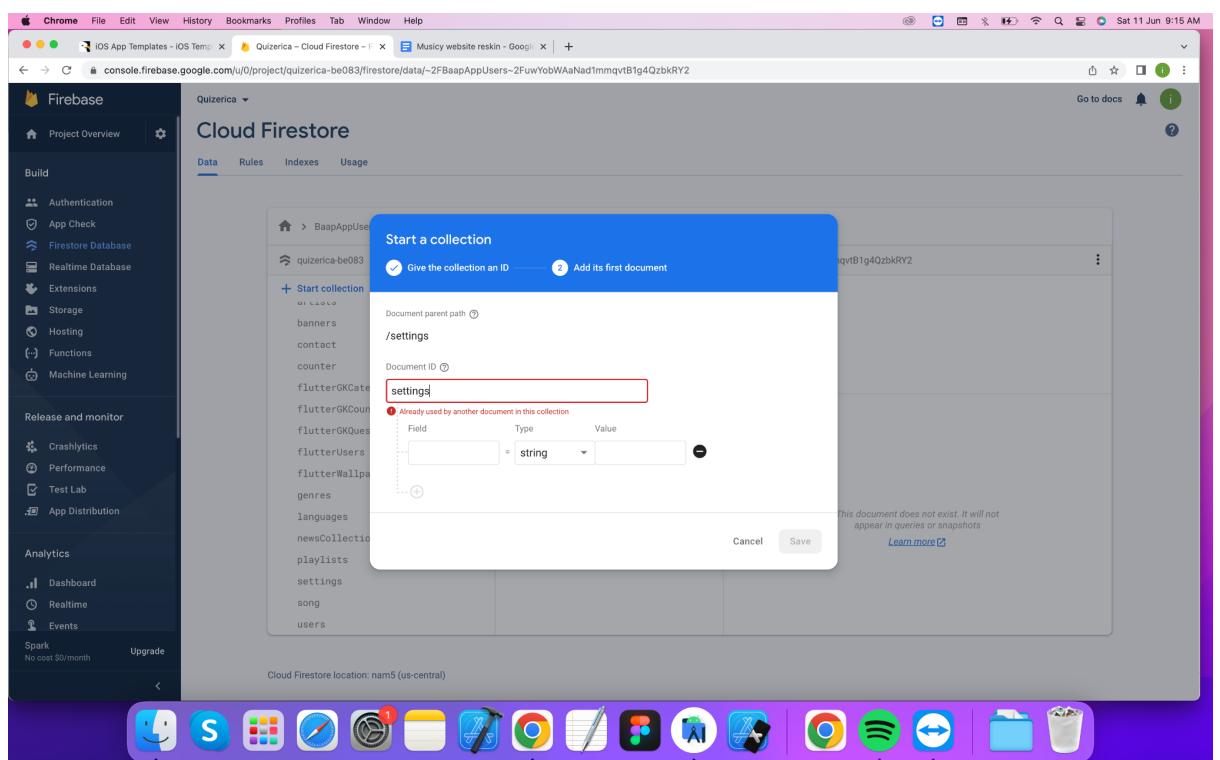
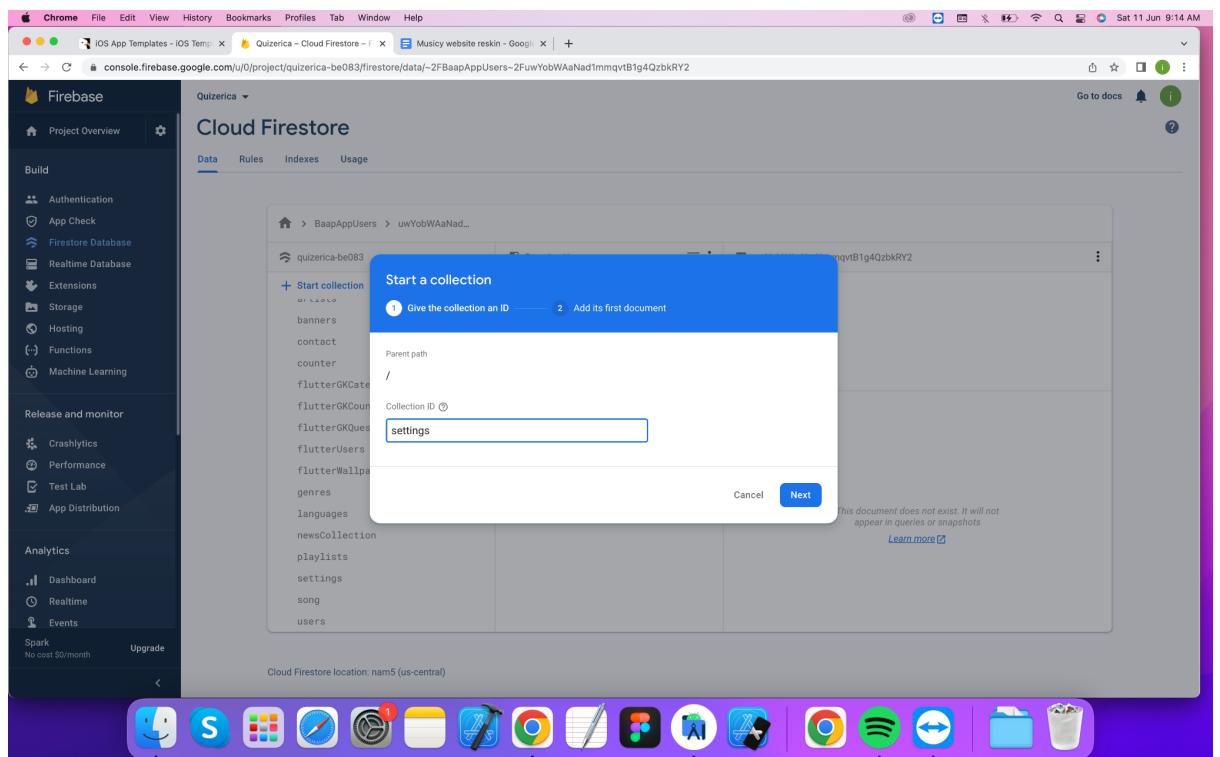
1. Open firebase project created above
2. Goto authentication menu from left menu.
3. Then open “Users” tab from top bar
4. Now click on add user and add email and password that you want to use for admin user
5. Please refer to screenshot below

Identifier	Created	Signed in	User UID
+17740075442	8 Jun 2022	11 Jun 2022	Y64v5qlpGRf3F6dGEx4Wv91eaM2
(anonymous)	7 Jun 2022	7 Jun 2022	4sVVVUWucFMbGUwTSPUNrHw...
(anonymous)	7 Jun 2022	7 Jun 2022	dTGNHxExaFORP5mQ0Wfkkp1rPX...
(anonymous)	7 Jun 2022	7 Jun 2022	poglCk1Q8zcRQvtdmGFAwwT11is1
(anonymous)	6 Jun 2022	6 Jun 2022	xlB7gDCpSTc1mBUZEN2JRH8UOH...
+19780676973	6 Jun 2022	9 Jun 2022	T00kNcYDR8Ysf97dGteB0KEDw02
(anonymous)	5 Jun 2022	5 Jun 2022	eAN6vmC27eTKoGM10ANvz4L1G...
(anonymous)	5 Jun 2022	5 Jun 2022	12bYgMuJ8H0d33hsnfVhXrRdMAi2
(anonymous)	5 Jun 2022	5 Jun 2022	dtKTUoarY9WhQxGLW9IFCvOzzy2

- Create some default collections**

1. Go to “Firestore database” from left menu in firebase project
2. Open “Data” tab from top bar
3. Now create a new collection named ‘settings’ and add an empty document ‘settings’ under “settings” collection
4. Repeat the same process again for ‘counter’ collection and created ‘counter’ document

Make sure you have entered exact same words, dont make spelling mistake and all are in small letters

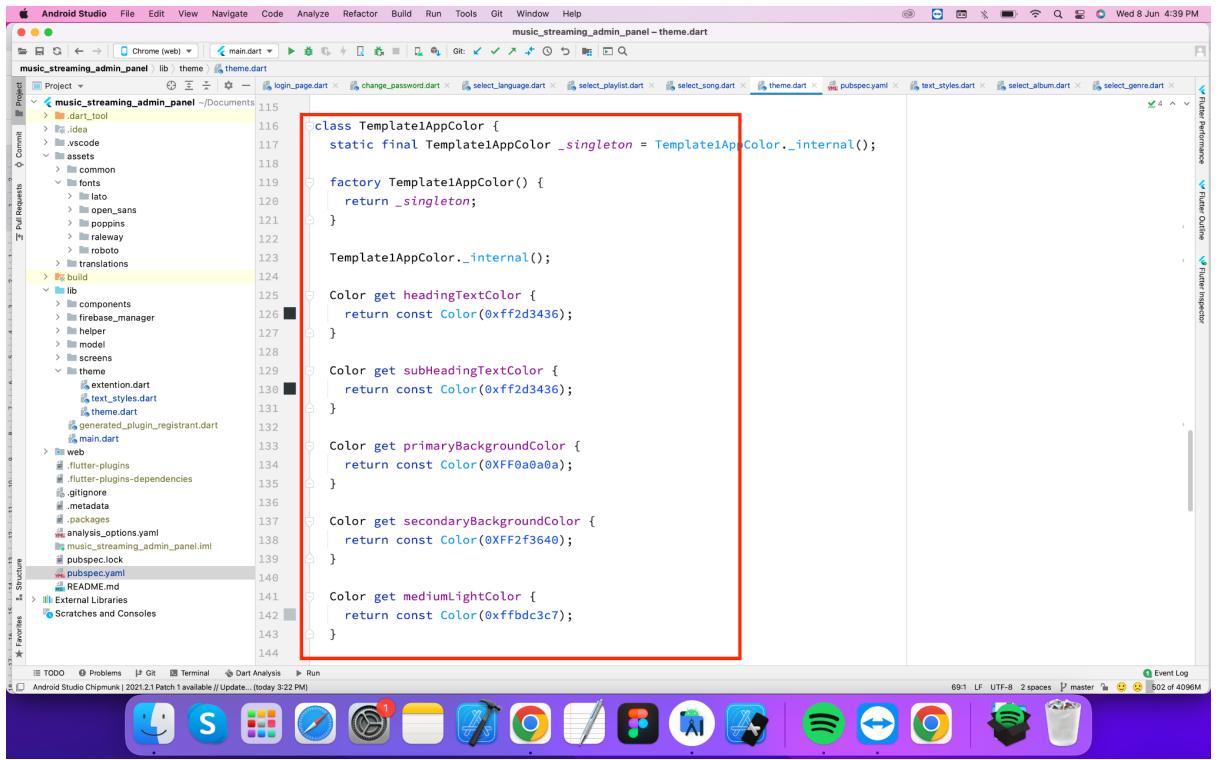


• Change colour theme

1. Open project under 'website' folder
2. Goto the 'lib/theme' folder and open the theme.dart file.
3. Under 'Template1AppColor' class you will find all the defined colours and you can

change hex colour codes.

Please refer to below screenshots for reference



The screenshot shows the Android Studio interface with the code editor open to the `theme.dart` file. A red box highlights the following code block:

```
class Template1AppColor {
    static final Template1AppColor _singleton = Template1AppColor._internal();

    factory Template1AppColor() {
        return _singleton;
    }

    Template1AppColor._internal();

    Color get headingTextColor {
        return const Color(0xff2d3436);
    }

    Color get subHeadingTextColor {
        return const Color(0xff2d3436);
    }

    Color get primaryBackgroundColor {
        return const Color(0xFF0a0a0a);
    }

    Color get secondaryBackgroundColor {
        return const Color(0XFF2f3640);
    }

    Color get mediumLightColor {
        return const Color(0xffbd3c7);
    }
}
```

• Change or add new fonts

1. Download some fonts from free fonts websites
2. Add the font in folder and drag that folder in project under 'assets/fonts' folder(refer to attached screenshot)
3. After adding the fonts goto 'pubspec.yaml' file and goto fonts section. And enter your fonts references. Please check already added fonts reference
4. After adding font reference go to theme.dart file again and add font name in 'Font' enum' - Line number 3
5. After that define the font name in switch case in line number 15
6. Then finally define the font name that you want to use in the website. Check line number 7

Drag your new font folder here

```

import 'package:flutter/material.dart';

enum Font { lato, openSans, poppins, raleway, roboto }

factory AppTheme() {
  return singleton;
}

AppTheme._internal();

String get fontName {
  switch (fontType) {
    case Font.roboto:
      return 'Roboto';
    case Font.raleway:
      return 'Raleway';
    case Font.poppins:
      return 'Poppins';
    case Font.openSans:
      return 'OpenSans';
    case Font.lato:
      return 'Lato';
  }
}

double get iconSize {
}

```

Define the font reference here

```

# example:
fonts:
  - family: Roboto
    fonts:
      - asset: assets/fonts/roboto/Roboto-Regular.ttf
      - asset: assets/fonts/roboto/Roboto-Bold.ttf

  - family: Lato
    fonts:
      - asset: assets/fonts/lato/Lato-Bold.ttf
      - asset: assets/fonts/lato/Lato-Regular.ttf

  - family: OpenSans
    fonts:
      - asset: assets/fonts/open_sans/OpenSans-Bold.ttf
      - asset: assets/fonts/open_sans/OpenSans-Regular.ttf

  - family: Poppins
    fonts:
      - asset: assets/fonts/poppins/Poppins-Bold.ttf
      - asset: assets/fonts/poppins/Poppins-Regular.ttf

  - family: Raleway
    fonts:
      - asset: assets/fonts/raleway/Raleway-Bold.ttf
      - asset: assets/fonts/raleway/Raleway-Regular.ttf

# For details regarding fonts from package dependencies,
# see https://flutter.dev/custom-fonts#from-packages

```

```

import 'package:flutter/material.dart';

enum Font { lato, openSans, poppins, raleway, roboto }

class AppTheme {
  static AppTheme singleton = AppTheme._internal();
  Font fontType = Font.roboto;

  factory AppTheme() {
    return singleton;
  }

  AppTheme._internal();

  String get fontName {
    switch (fontType) {
      case Font.roboto:
        return 'Roboto';
      case Font.raleway:
        return 'Raleway';
      case Font.poppins:
        return 'Poppins';
      case Font.openSans:
        return 'OpenSans';
      case Font.lato:
        return 'Lato';
    }
  }

  double get iconSize {
  }
}

```

• Change icons

1. Download the icons that you want to replace.
2. After downloading, give them the same name as already given.
3. Delete the old icon and drag the new icon in 'assets/icons' folder

• Firebase linking

1. Open main.dart file and replace the firebase related key values with your own create firebase project key value(that you already done above in step 12 under firebase setup)
2. To change firebase storage bucket url goto 'helper/contant_util.dart' file' and replace 'firebaseStorageBucketUrl' value to your firebase storage bucket url, please get it from firebase app keys that you created above. Please refer to screenshot below

Android Studio - Fri 10 Jun 12:29 PM

Project: music_streaming_admin_panel / lib / helper / constant_util.dart

```
music_streaming_admin_panel - constant_util.dart
1 class ConstantUtil {
2     static String projectName = 'Musicy';
3     static String firebaseStorageBucketUrl = 'quizerica-be083.appspot.com';
4
5     static bool isLoggedIn = false;
6     static bool isRightToLeft = false;
7 }
```

File Structure

ConstantUtil.dart

Localization

Model

Screens

Theme

UI

Utilities

Widgets

Event Log

3:30 (74 chars, 1 line break) LF: UTF-8 2 spaces master 687 of 4096M

Android Studio - Wed 8 Jun 5:16 PM

Project: music_streaming_admin_panel / lib / main.dart

```
music_streaming_admin_panel - main.dart
1 import 'package:music_streaming_admin_panel/screens/user/login_page.dart';
2
3 void main() async{
4     WidgetsFlutterBinding.ensureInitialized();
5     await EasyLocalization.ensureInitialized();
6
7     await Firebase.initializeApp(
8         // Replace with actual values
9         options: const FirebaseOptions(
10             apiKey: "AIzaSyC7Ar10koCMiiAEzGLOZ1wcTeaARYZW150",
11             appId: "1:500884745411:web:8e96fa15d302baf6f6c676",
12             messagingSenderId: "500884745411",
13             projectId: "quizerica-be083",
14             storageBucket: "quizerica-be083.appspot.com",
15         ),
16     );
17
18     await FirebaseAuth.instance.authStateChanges().first;
19
20     runApp(
21         EasyLocalization(
22             useOnlyLangCode: true,
23             supportedLocales: const [
24                 Locale('en', 'US'),
25             ],
26         ),
27     );
}
```

Messages: [music_streaming_admin_panel] Flutter

```
flutter pub get
Running "flutter pub get" in music_streaming_admin_panel... 3.9s
Process finished with exit code 0
```

File Structure

ConstantUtil.dart

Localization

Model

Screens

Theme

UI

Utilities

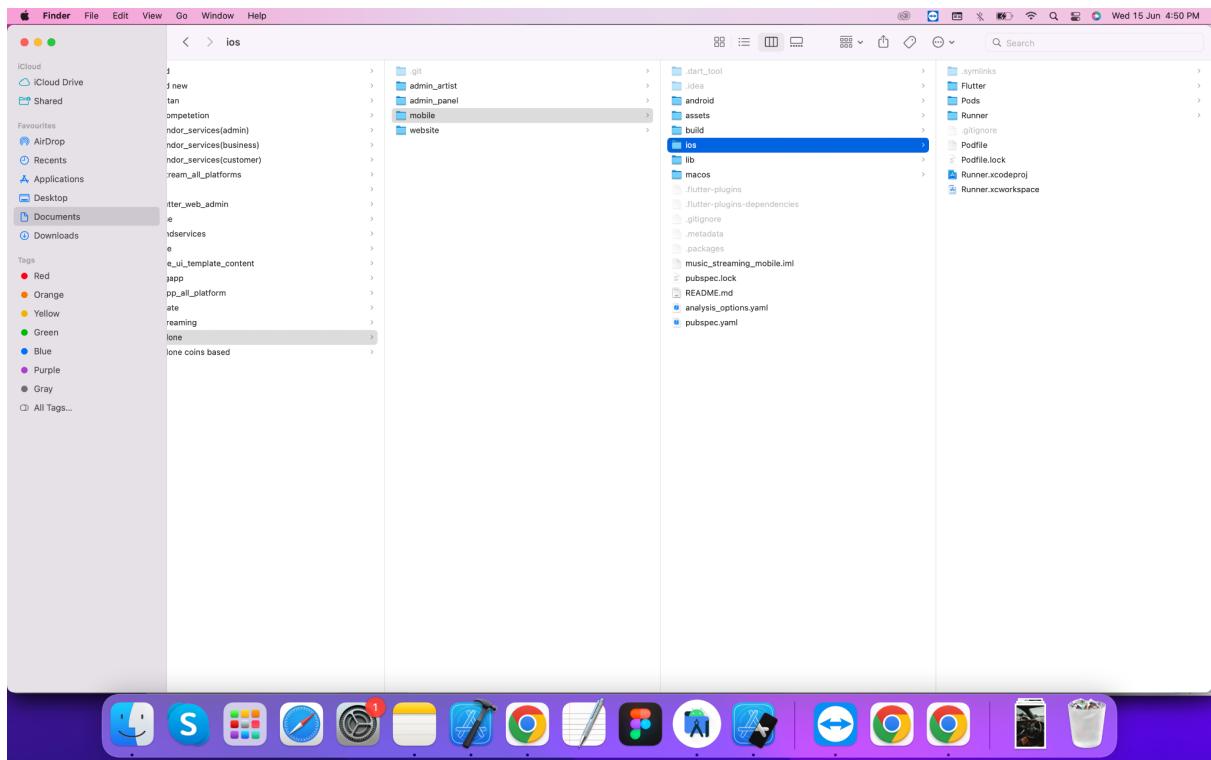
Widgets

Event Log

13:16 LF: UTF-8 2 spaces master 615 of 4096M

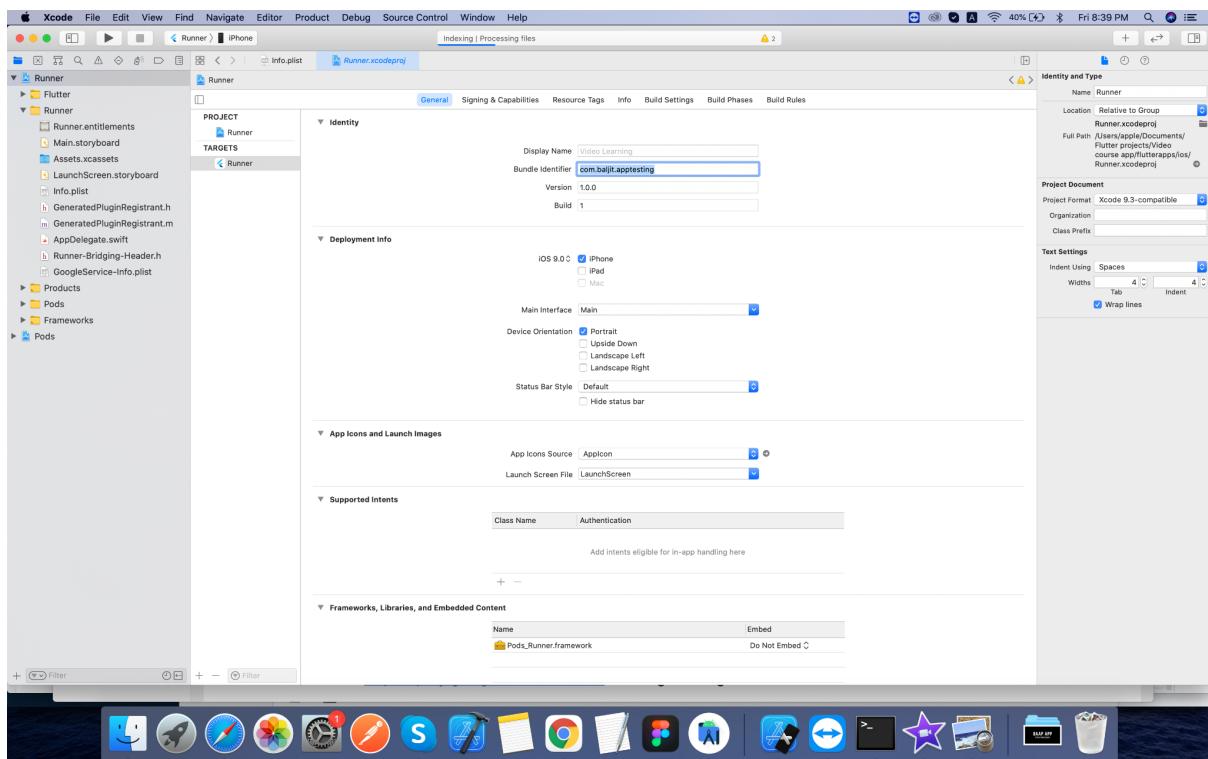
iOS App setup

1. Open ios project from 'mobile' project in xcode.
2. You need to double click on 'Runner.xcworkspace' file to open project



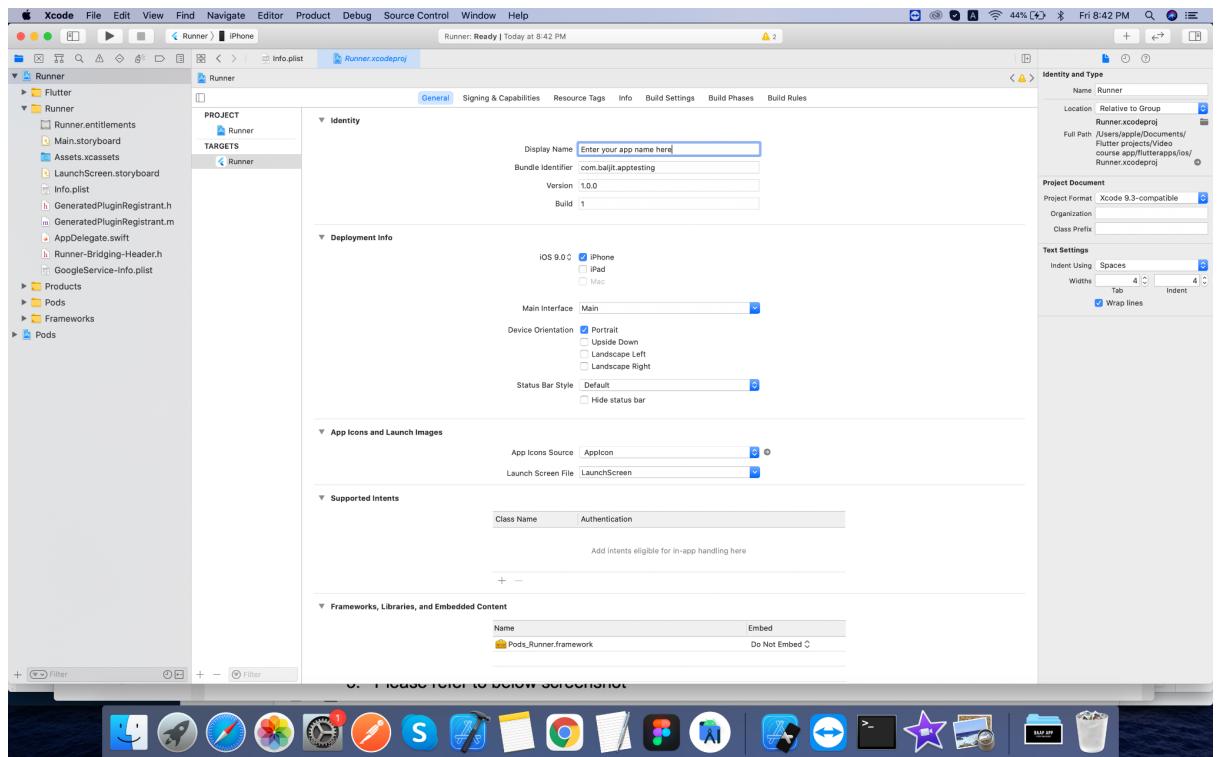
- **Change bundle identifier**

1. Replace the bundle identifier with your created bundle identifier above in ios app setup in apple developer account
2. Refer to screenshot below



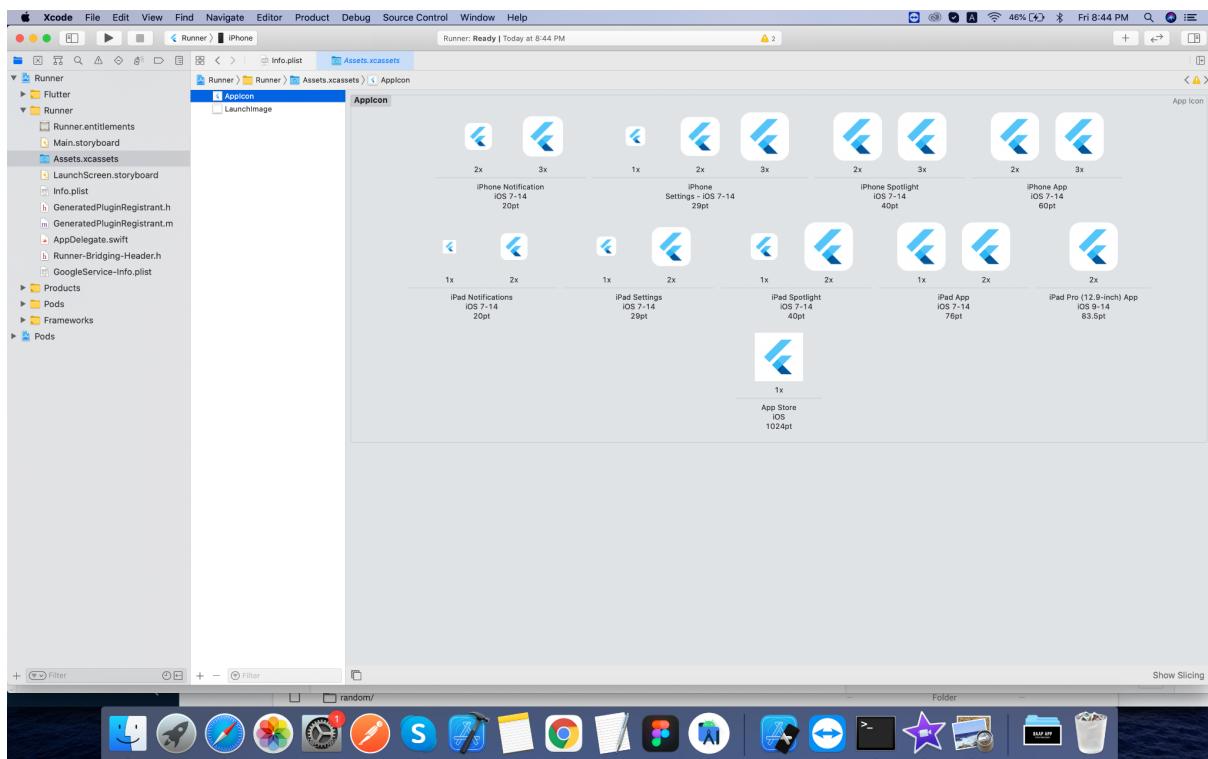
• Change App Name

1. Add new app name in display name field
2. Please refer to below screenshot



Change app icon

Go to "Assets.xcassets" folder and replace app icon as shown in below screenshot



Change app graphics

Open ‘mobile’ project in android studio

Change graphics in the assets folder of the project. Please refer to the screenshot below.

Make sure you dont change the name of new assets. Names are case sensitive also

```

import 'main.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await EasyLocalization.ensureInitialized();

  await setupServiceLocator();

  await Firebase.initializeApp();
  if (FirebaseAuth.instance.currentUser != null) {
    await UserProfileManager().refreshProfile();
  }

  runApp(
    EasyLocalization(
      useOnlyLangCode: true,
      supportedLocales: const [
        Locale('en', 'US'),
        Locale('ar', 'AE'),
        Locale('ar', 'DZ'),
        Locale('de', 'DE'),
        Locale('fr', 'FR'),
        Locale('ru', 'RU')
      ],
      path: 'assets/translations',
      // <-- change the path of the translation files
      fallbackLocale: const Locale('en', 'US'),
      child: const MainApp(), // EasyLocalization
    );
}

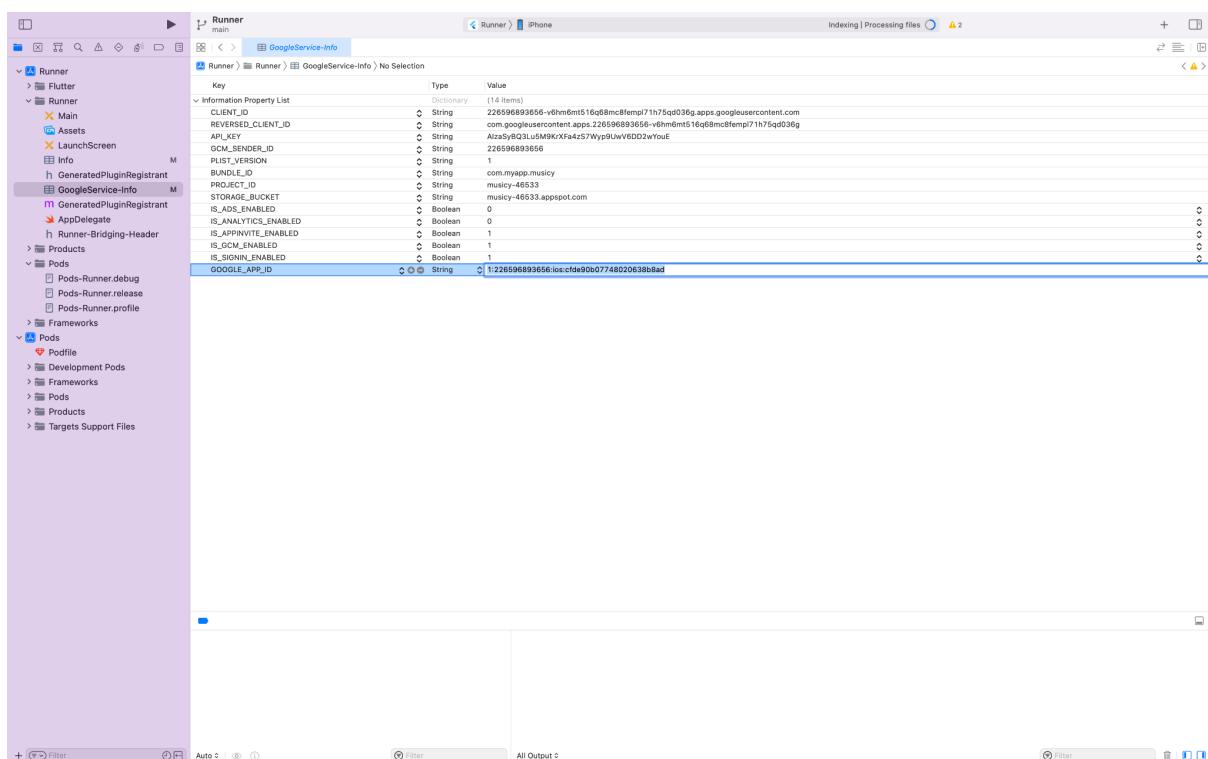
```

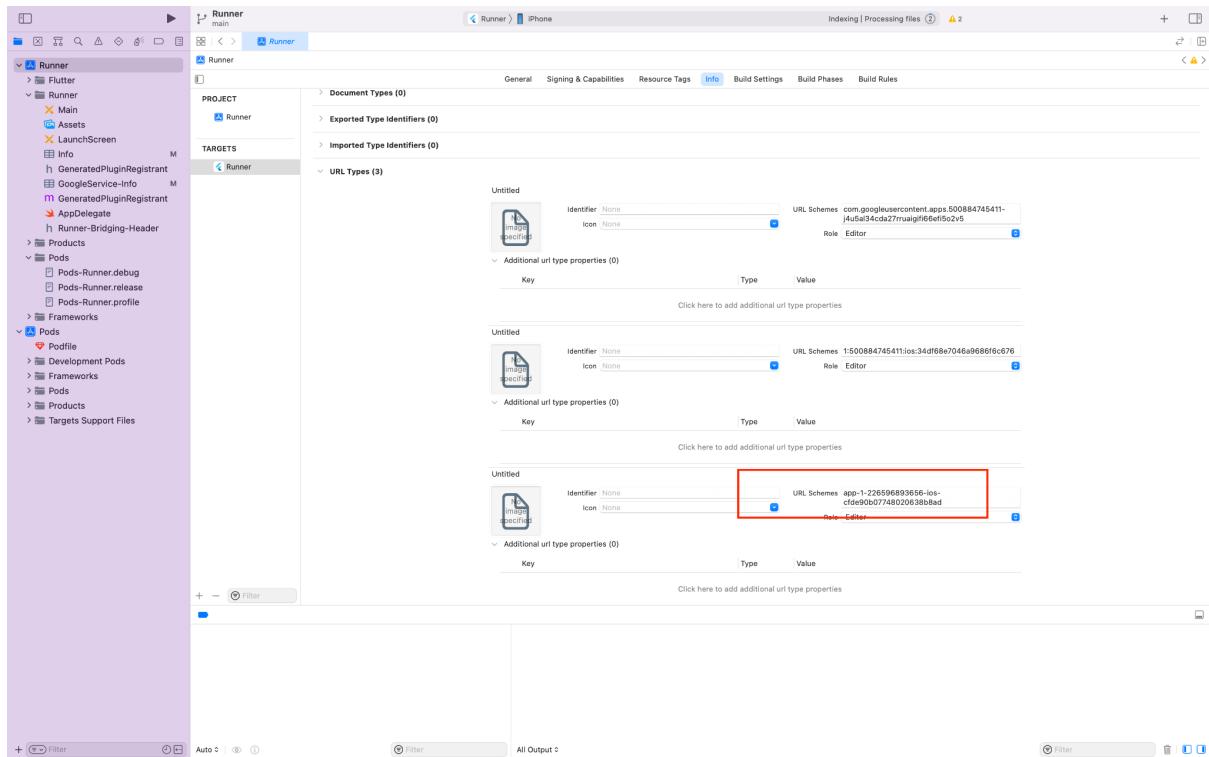
Link with firebase project

- Open ‘mobile’ project in android studio
- Open iOS/Runner folder and delete the older “GoogleService-Info.plist” file
- Drag and drop the new “GoogleService-Info.plist” config file that you have created above for your ios app. Make sure you drag the file at same location from where you delete the older file
- Now register the Custom url scheme for login via phone
- Copy the google app id from “GoogleService-Info.plist” file as shown in screenshot below screenshot 2
- Add this google app id in the custom URL scheme as shown in the below screenshot. While adding please add ‘app-’ in front of google app id. Screenshot 3

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure for "music_streaming_mobile". It includes:
 - gradle**: build.gradle, gradle.properties, gradlew, gradlew.bat, local.properties.
 - ios**: GoogleService-Info.plist, Runner.xcodeproj, Runner.xcworkspace, Base.lproj, AppDelegate.swift, GeneratedPluginRegistrant.h, GeneratedPluginRegistrant.m, Info.plist, Runner-Bridging-Header.h.
 - lib**: flutter-plugins-dependencies, flutter-plugins.
 - macos**: Runner-Bridging-Header.h.
 - Pods**: Pods-Runner.debug, Pods-Runner.release, Pods-Runner.profile.
 - Frameworks**: Podfile.
- Code Editor:** The main window displays the content of the Podfile. The code defines a project named 'Runner' with build configurations for 'Debug', 'Profile', and 'Release'. It also sets the flutter root directory and performs a foreach loop to find and remove matches for the FLUTTER_ROOT environment variable. Finally, it requires the flutter_tools package and runs the flutter_iOS_podfile_setup command.
- Toolbar:** The bottom toolbar includes icons for various Mac applications like Finder, Mail, Safari, and others.
- Status Bar:** The status bar at the bottom shows the date (Wed 8 Jun 5:53 PM), time (13:17), file encoding (UTF-8), and line count (804 of 496M).





Android App setup

Replace the package name

Now replace the package with your own created package name as shown in below screenshot. There will be two files where you need to change the package name.

1. AndroidManifest.xml
2. build.gradle

The screenshot shows the Android Studio interface with the following details:

- Project Tree:** The left sidebar shows the project structure under "flutterapps". Key files visible include `AndroidManifest.xml`, `ConfigConstants.dart`, `main.dart`, and `NetworkConstant.dart`.
- Code Editor:** The main window displays the content of `AndroidManifest.xml`. The code is as follows:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.jaspj.baaapp"
    >
    <application
        android:label="Baa App"
        android:icon="@mipmap/ic_launcher">
        <activity
            android:name=".MainActivity"
            android:launchMode="singleTop"
            android:theme="@style/LaunchTheme"
            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|locale|layoutDirection|fontScale|screenLayout|density|uiMode"
            android:hardwareAccelerated="true"
            android:windowSoftInputMode="adjustResize">
            <!-- Specifies an Android theme to apply to this Activity as soon as
                the Android process has started. This theme is visible to the user
                while the Flutter UI initializes. After that, this theme continues
                to determine the Window background behind the Flutter UI. -->
            <meta-data
                android:name="io.flutter.embedding.android.NormalTheme"
                android:resource="@style/NormalTheme"
            />
            <!-- Displays an Android View that continues showing the launch screen
                Drawable until Flutter paints its first frame, then this splash
                screen fades out. A splash screen is useful to avoid any visual
                gap between the end of Android's launch screen and the painting of
                Flutter's first frame. -->
            <meta-data
                android:name="io.flutter.embedding.android.SplashScreenDrawable"
                android:resource="@drawable/launch_background"
            />
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <!-- Facebook Login configuration -->
        <meta-data android:name="com.facebook.sdk.ApplicationId"
            android:value="@string/facebook_app_id"/>
    </application>
</manifest>
```

The code editor includes syntax highlighting for XML tags and Java/Android code. The status bar at the bottom shows "18 chars 2:32 LF UTT-0 4 spaces".

Change app name

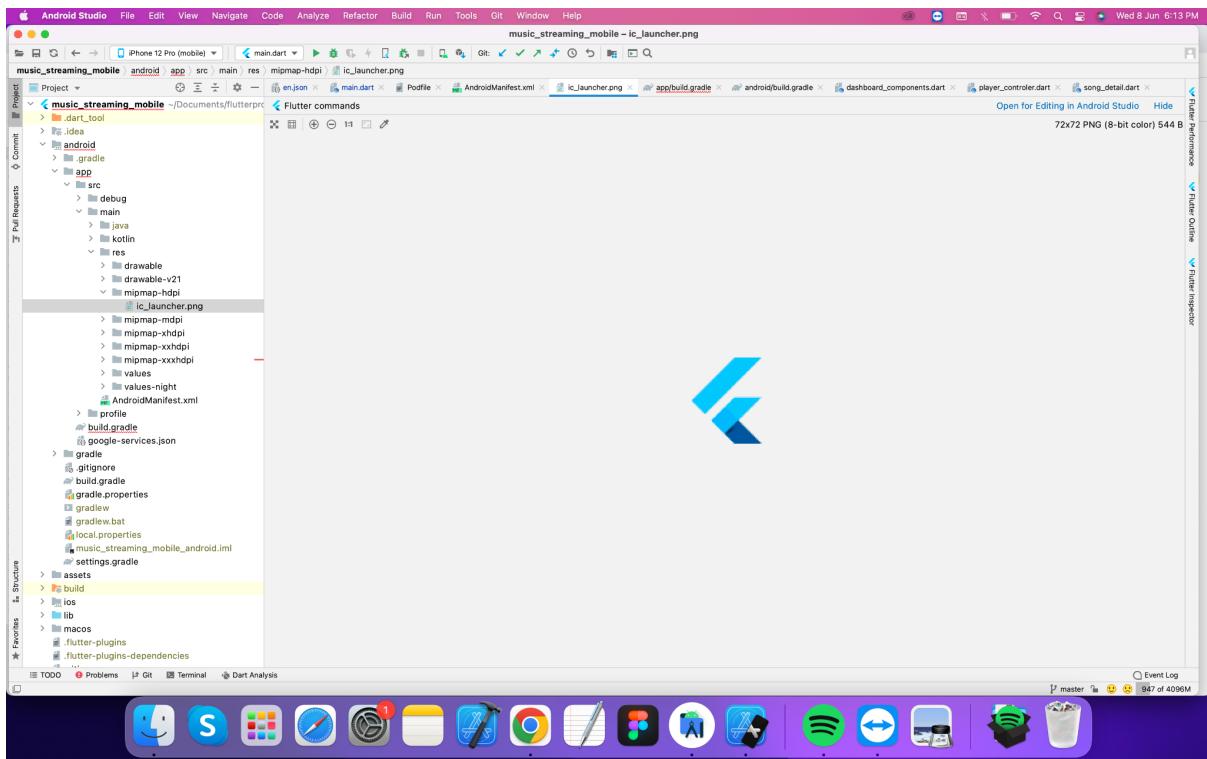
1. Go to project and then open folder in sequence android -> app -> src -> main -> `AndroidManifest.xml`
2. Add new app name in `android:label` in application tag
3. Please refer to below screenshot

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.music_streaming_mobile"
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:label="music_streaming_mobile"
        android:name="${applicationName}"
        android:icon="@mipmap/ic_launcher">
        <activity
            android:name=".AudioServiceActivity"
            android:exported="true"
            android:launchMode="singleTop"
            android:theme="@style/LaunchTheme"
            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|locale|layoutDirection|fontScale|density|ActionBar|color|size"/>
            <!-- Specifies an Android theme to apply to this Activity as soon as
                the Android process has started. This theme is visible to the user
                while the Flutter UI initializes. After that, this theme continues
                to determine the Window background behind the Flutter UI. -->
        <meta-data
            android:name="io.flutter.embedding.android.NormalTheme"
            android:resource="@style/NormalTheme"/>
        </activity>
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    
```

Change app icon

1. Go to project and then open folder in sequence android -> app -> src -> main -> res
2. There will be folders like mipmap-hdpi, mipmap-mdpi etc. Replace icons in these folders as shown in the below screenshot.



Once you are all done. Run the app and then you will get some error logs to create indexes in the firebase. It will show some clickableUrls , Click on that url and click create index. This process will be repeated for 15-20 indexes.

Thanks