120A3051

Shreya Idate

Batch: E3

Experiment No: 7

AIM: To connect Flutter UI with Firebase realtime database

THEORY: Firebase Database is a cloud-based database that stores data within a JSON structure. Flutter with Firebase is now officially named FlutterFire. All devices connected with Firebase Realtime Database whether it is a mobile device or website automatically receive updates from Firebase with the newest data. You don't need to call any external services to get data from the firebase realtime database in flutter. It is necessary to include the *firebase_core* as well as the *firebase_database* plugins to the Flutter project.

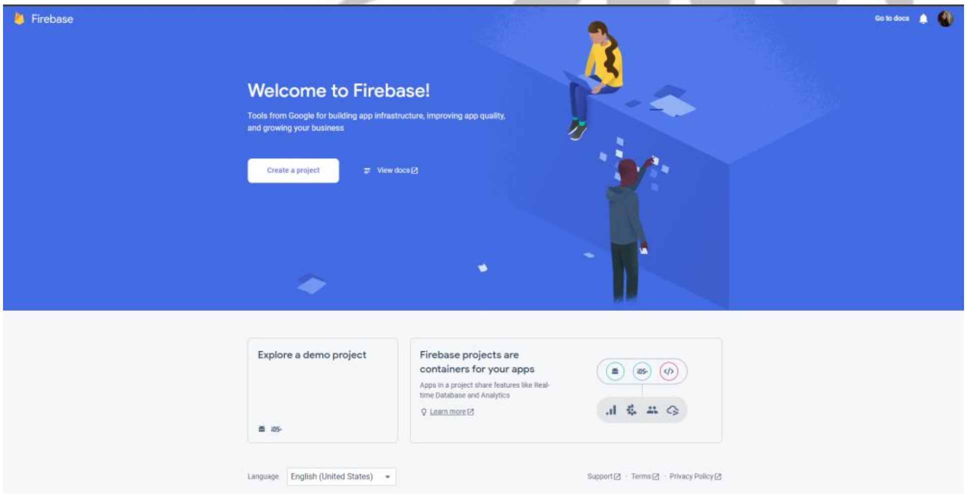
Flutter is a cross-platform development framework. So Flutter 2 now officially releases the stable version of flutter web. Same like flutter App, to read data from realtime database in flutter web, no need to add extra code. The same code base will be executed on the flutter web as well. All you need to do is just add two JS scripts on the index.html file inside the web folder.

Note – Firebase real-time databases and Firebase Firestore are distinct things. Firebase Real-Time Database stores every information in a single large JSON file. However, Firestore is an array made up of many JSON files known as *collections*.

Creating a Firebase Account

In order to use a Firestore database, you need a Firebase account. Go to https://firebase.google.com/ and sign up for an account.

On the Welcome to Firebase page, click the **Create a project** button. If Firebase shows a different page, you can click **Go to console** in the top right corner and then click **Add project**.



Welcome to Firebase!

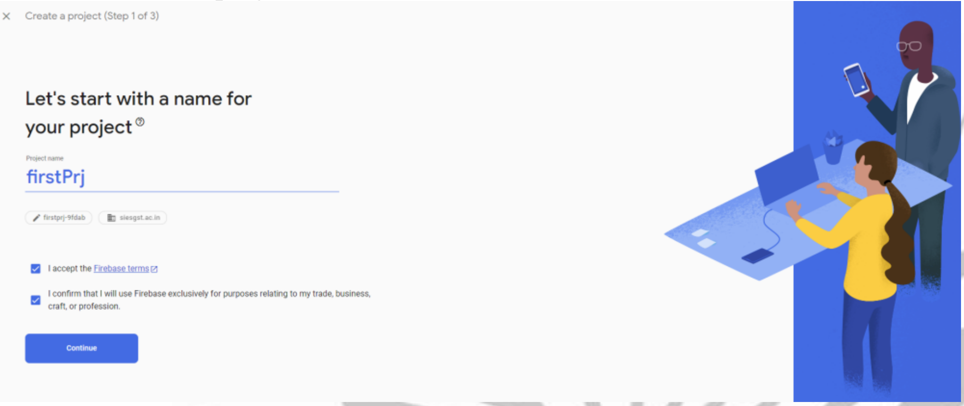
Tools from Google for building app infrastructure, improving app quality, and growing your business

Firebase projects are containers for your apps

Apps in a project share features like Real-time Database and Analytics

Language: English (United States)

Now, enter the project name: **first** and click the **Continue** button.



Create a project (Step 1 of 3)

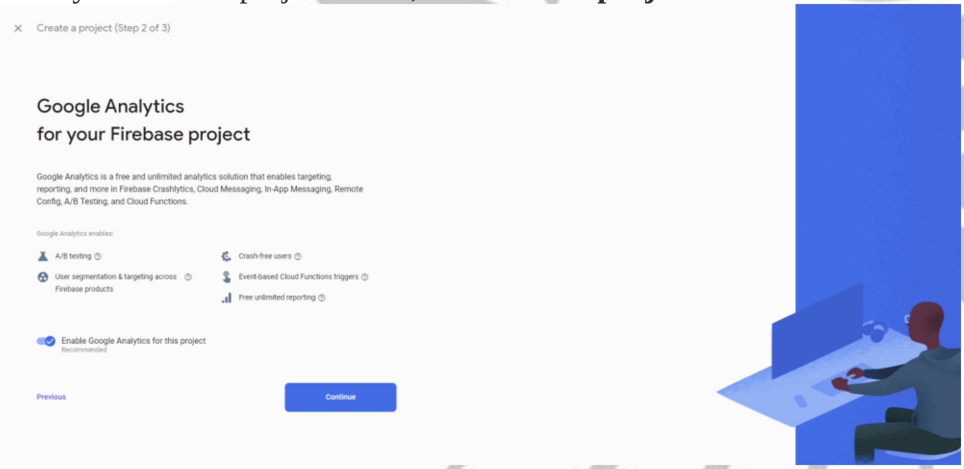
Let's start with a name for your project

Project name: firstPrj

☑ I accept the Firebase terms

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

On the next page, toggle the switch for **Enable Google Analytics** to the **off** position. You won't use analytics for this project. Then, click **Create project**.



Create a project (Step 2 of 3)

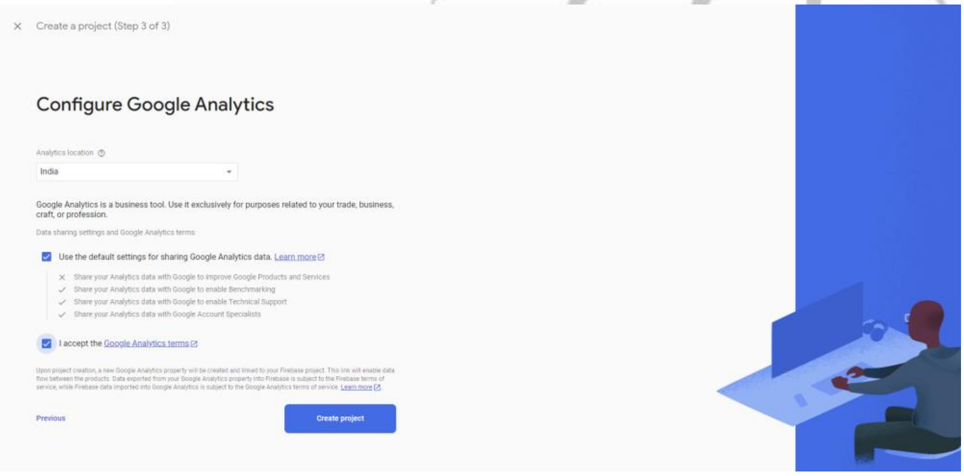
Google Analytics for your Firebase project

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 & targeting across Firebase products
- Crash-free users
- Event-based Cloud Functions triggers
- Free unlimited reporting

Enable Google Analytics for this project: on (Recommended)



Create a project (Step 3 of 3)

Configure Google Analytics

Analytics location: India

Google Analytics is a business tool. Use it exclusively for purposes related to your trade, business, craft, or profession.

Data sharing settings and Google Analytics terms

☑ Use the default settings for sharing Google Analytics data. Learn more

- ✗ Share your Analytics data with Google to improve Google Products and Services
- ✓ Share your Analytics data with Google to enable Benchmarking
- ✓ Share your Analytics data with Google to enable Technical Support
- ✓ Share your Analytics data with Google Account Specialists

☑ I accept the Google Analytics terms

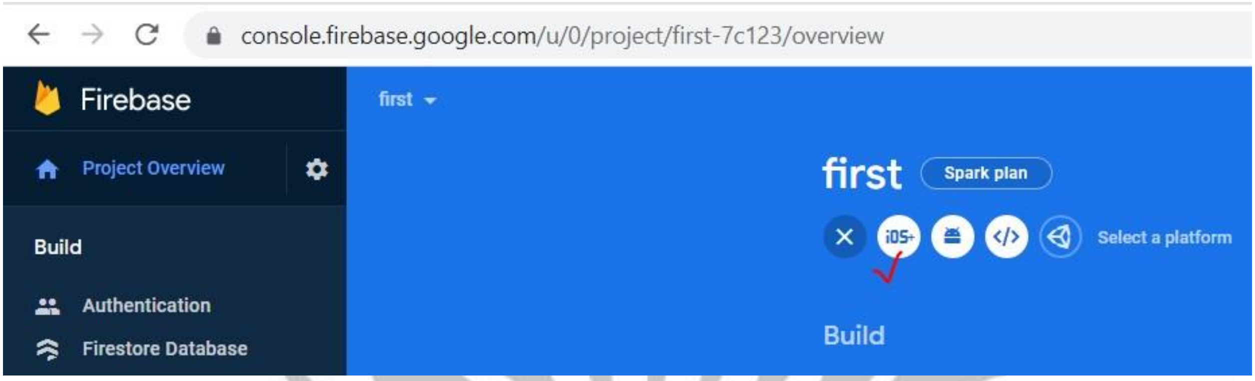
Upon project creation, a new Google Analytics property will be created and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. Learn more.

You'll see a few progress dialogs as Firebase allocates resources to your new project

Once your project is ready, click **Continue** to move to the page where you'll add Firebase to both your iOS and Android apps. Start with the iOS app.

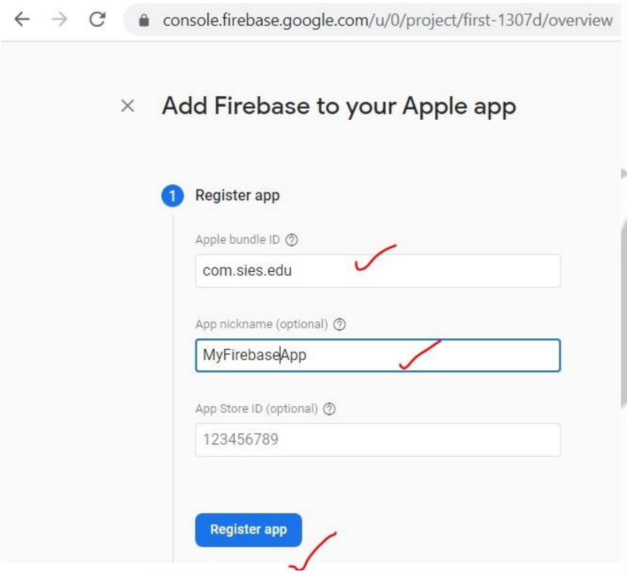
Registering an iOS App

To register the iOS app, click the iOS circle:



You'll see a dialog to register your app. Enter **com.sies.edu** for the iOS bundle ID and click the **Register app** button.

Note: If you created the Flutter app from scratch enter the bundle ID you used to create the app.

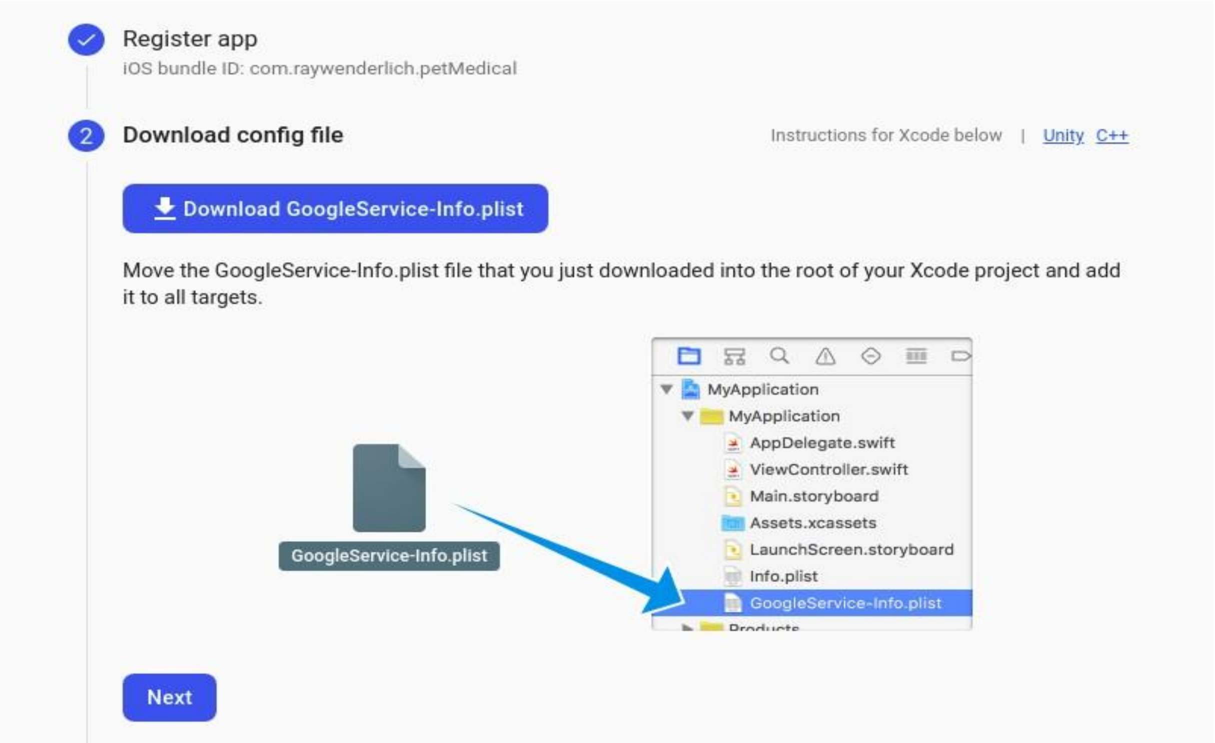

console.firebase.google.com/u/0/project/first-1307d/overview
×
Add Firebase to your Apple app
1
Register app
Register app

Apple bundle ID: com.sies.edu

App nickname (optional): MyFirebaseApp

App Store ID (optional): 123456789

Next, click the **Download GoogleService-Info.plist** button.



Register app

iOS bundle ID: com.raywenderlich.petMedical

2 Download config file

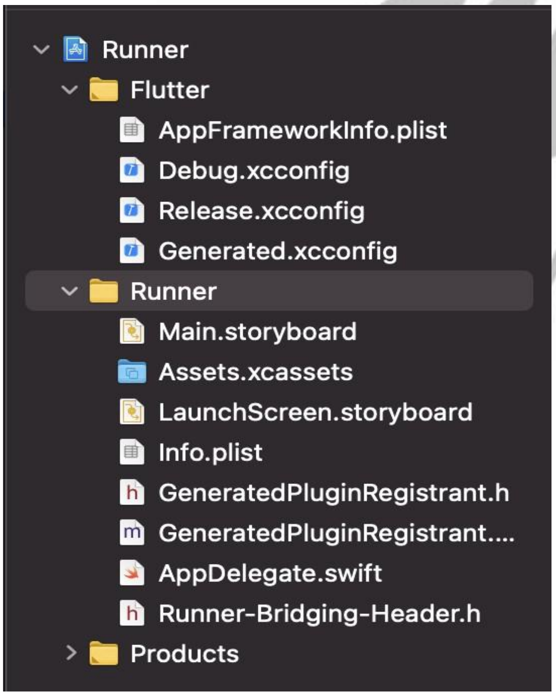
Instructions for Xcode below | Unity C++

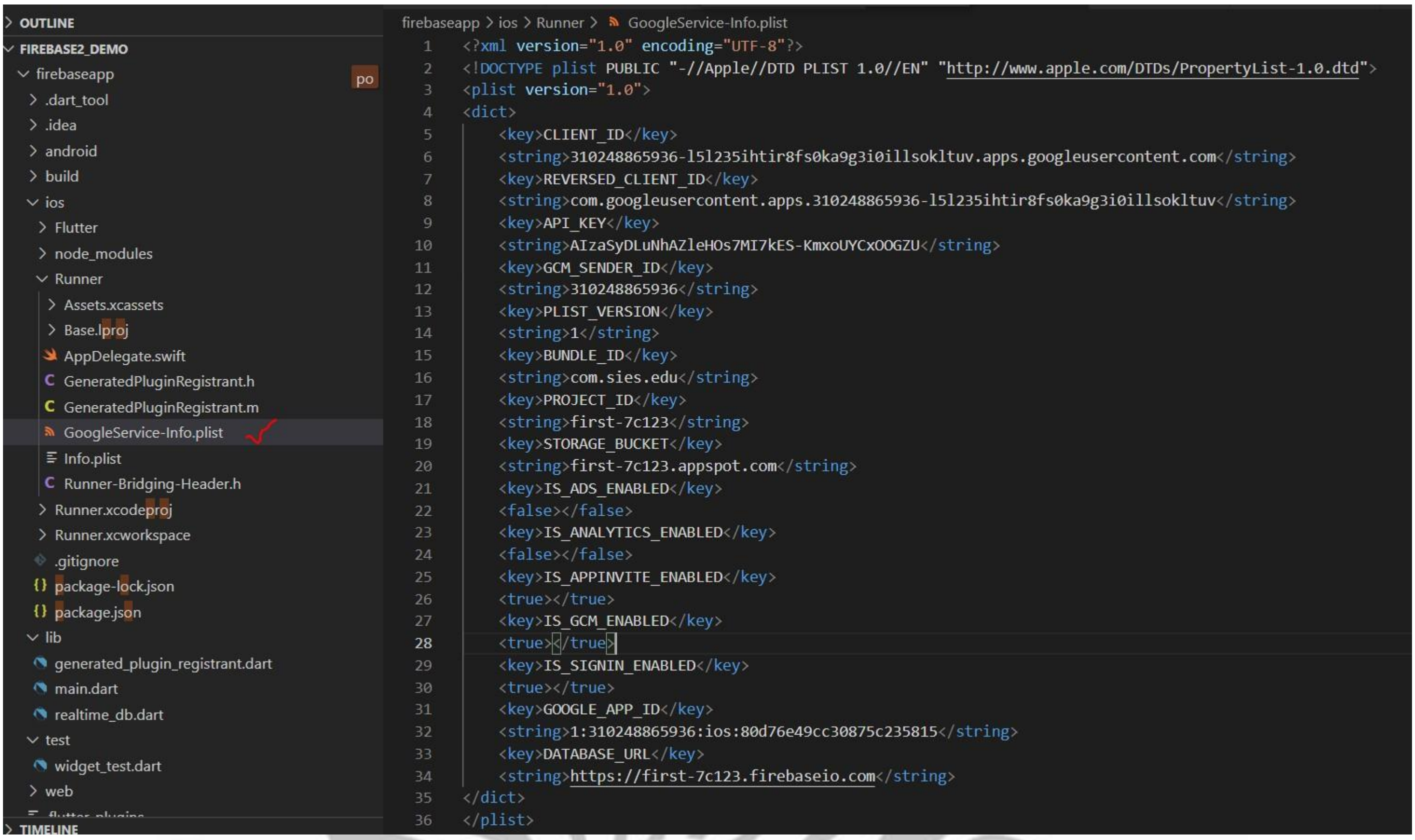
Download GoogleService-Info.plist

Move the GoogleService-Info.plist file that you just downloaded into the root of your Xcode project and add it to all targets.

Now, move this file into the **iOS ‣ Runner** folder. If using Android Studio, go to **Tools ‣ Flutter** menu, choose **Open iOS module in Xcode**. In Xcode, right-click the **Runner** folder and choose **Add files to Runner…**.

Next, add **GoogleService-Info.plist**:



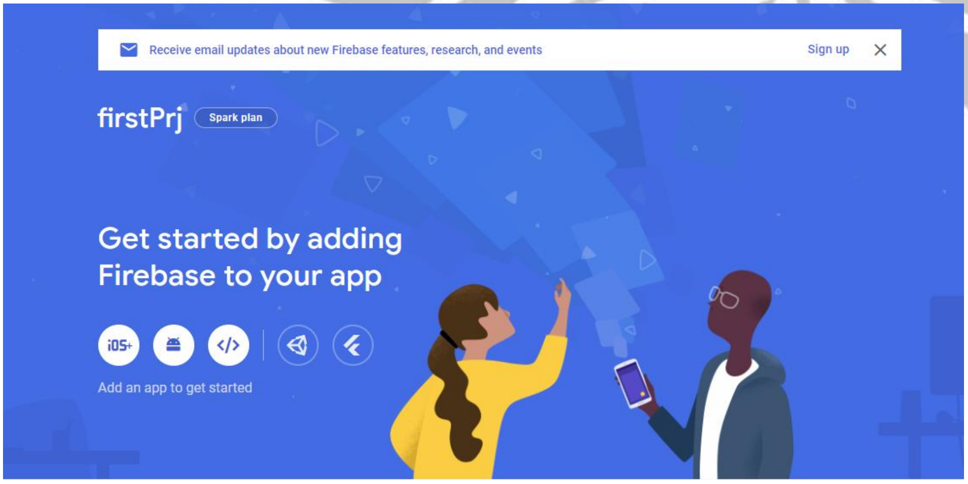


```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>CLIENT_ID</key>
    <string>310248865936-15l235ihtir8fs0ka9g3i0illsokltuv.apps.googleusercontent.com</string>
    <key>REVERSED_CLIENT_ID</key>
    <string>com.googleusercontent.apps.310248865936-15l235ihtir8fs0ka9g3i0illsokltuv</string>
    <key>API_KEY</key>
    <string>AIzaSyDLuNhAZleHOs7MI7kES-KmxoUYCxOOGZU</string>
    <key>GCM_SENDER_ID</key>
    <string>310248865936</string>
    <key>PLIST_VERSION</key>
    <string>1</string>
    <key>BUNDLE_ID</key>
    <string>com.sies.edu</string>
    <key>PROJECT_ID</key>
    <string>first-7c123</string>
    <key>STORAGE_BUCKET</key>
    <string>first-7c123.appspot.com</string>
    <key>IS_ADS_ENABLED</key>
    <false></false>
    <key>IS_ANALYTICS_ENABLED</key>
    <false></false>
    <key>IS_APPINVITE_ENABLED</key>
    <true></true>
    <key>IS_GCM_ENABLED</key>
    <true></true>
    <key>IS_SIGNIN_ENABLED</key>
    <true></true>
    <key>GOOGLE_APP_ID</key>
    <string>1:310248865936:ios:80d76e49cc30875c235815</string>
    <key>DATABASE_URL</key>
    <string>https://first-7c123.firebaseio.com</string>
</dict>
</plist>
```

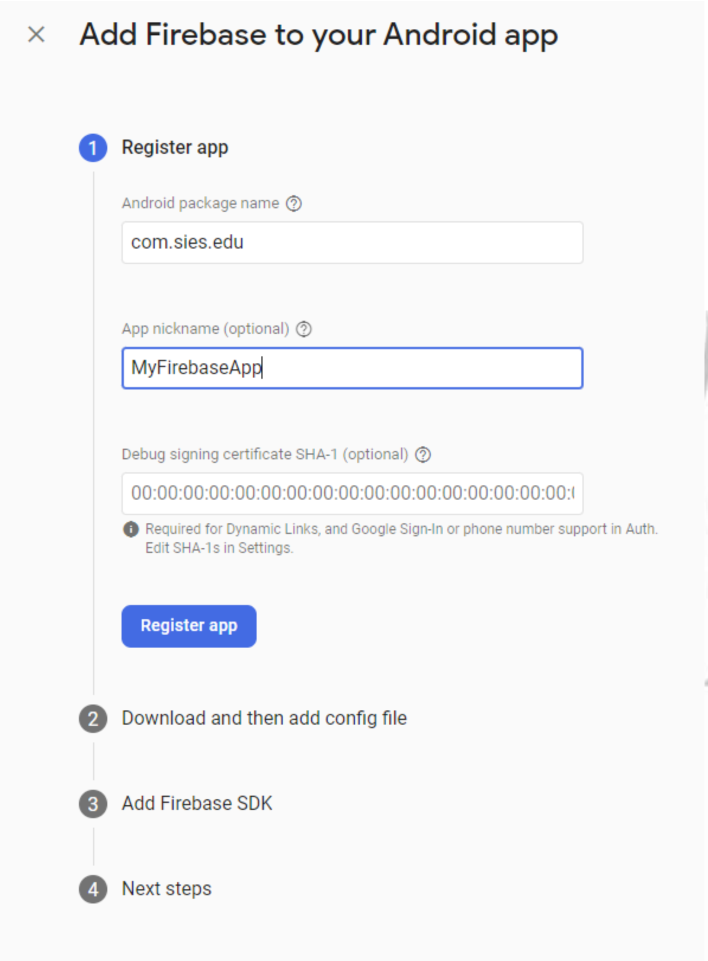
Nice job! Now it’s time to register the Android app.

Registering an Android App

First, go back to the Firebase page. Click the **Android** circle to start the process of adding Firebase to Android.


Receive email updates about new Firebase features, research, and events
Sign up
firstPrj
Spark plan
Get started by adding Firebase to your app
iOS+
Add an app to get started

You’ll see a dialog to register your app. Enter **com.sies.edu** in the Android package name field. Next, click **Register app**:



× Add Firebase to your Android app

1 Register app

Android package name: com.sies.edu

App nickname (optional): MyFirebaseApp

Debug signing certificate SHA-1 (optional): 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0

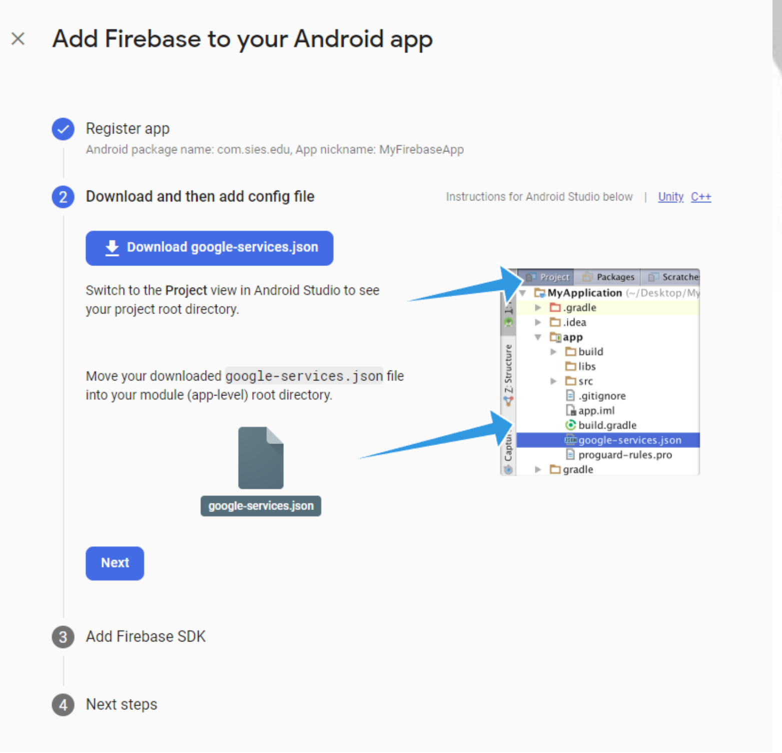
Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

2 Download and then add config file

3 Add Firebase SDK

4 Next steps

Then, click the **Download google-services.json** button and move this file into the **android** ▸ **app** folder.



× Add Firebase to your Android app

✓ Register app

Android package name: com.sies.edu, App nickname: MyFirebaseApp

2 Download and then add config file

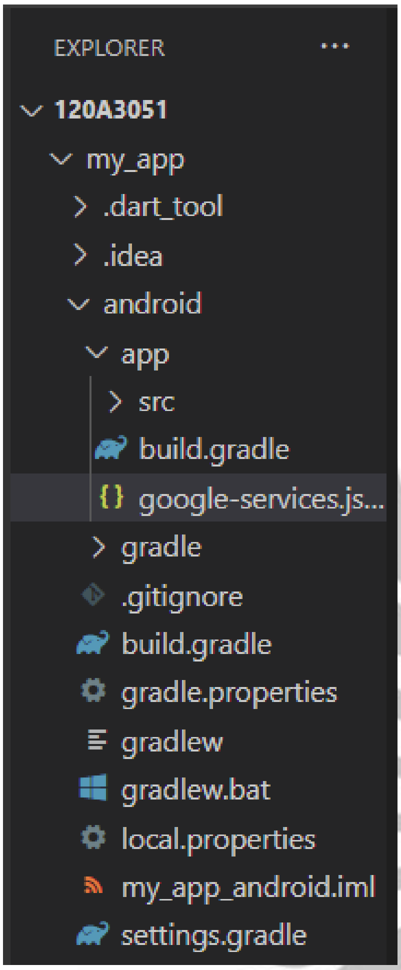
Instructions for Android Studio below | Unity C++

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

Move your downloaded `google-services.json` file into your module (app-level) root directory.

3 Add Firebase SDK

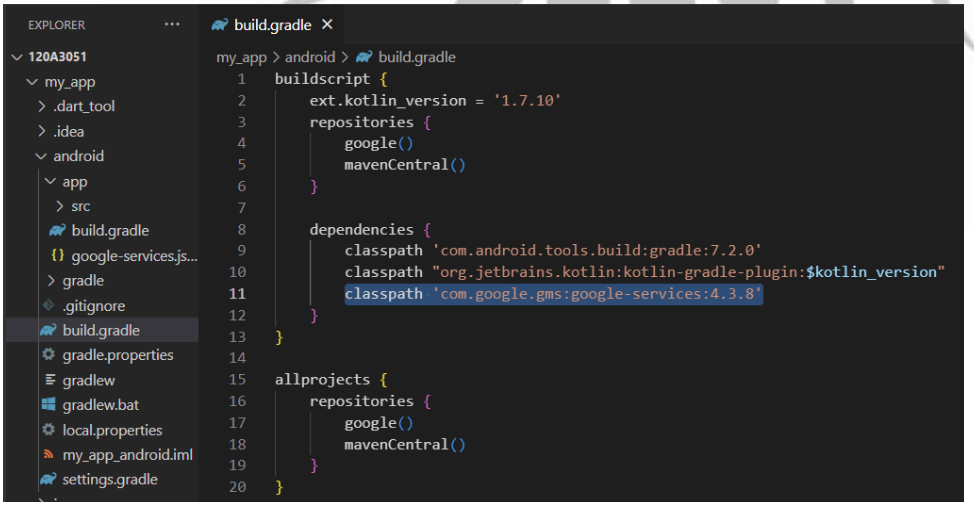
4 Next steps



```
EXPLORER ...
120A3051
  my_app
    .dart_tool
    .idea
    android
      app
        src
        build.gradle
        google-services.js...
      gradle
      .gitignore
      build.gradle
      gradle.properties
      gradlew
      gradlew.bat
      local.properties
      my_app_android.iml
      settings.gradle
```

Now, open **android/build.gradle**. Then, add the following dependency after the last classpath entry:

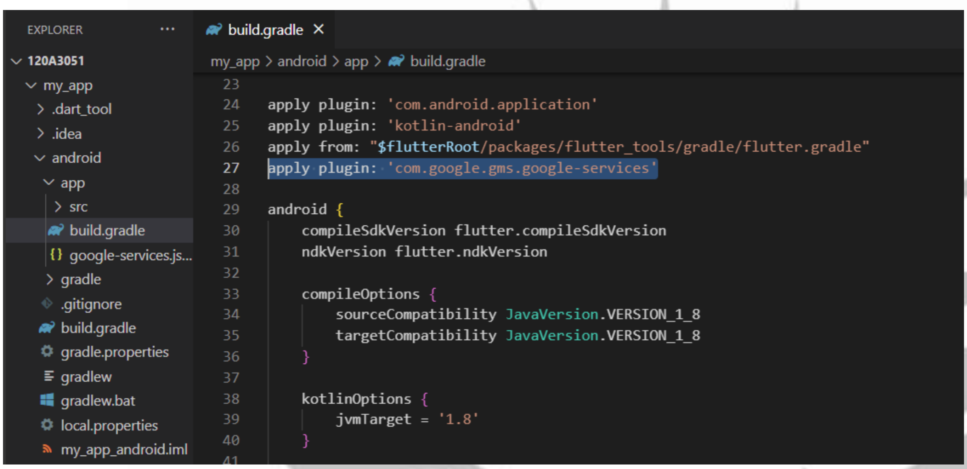
```
classpath 'com.google.gms:google-services:4.3.8'
```



```
buildscript {
    ext.kotlin_version = '1.7.10'
    repositories {
        google()
        mavenCentral()
    }

    dependencies {
        classpath 'com.android.tools.build:gradle:7.2.0'
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
        classpath 'com.google.gms:google-services:4.3.8'
    }
}

allprojects {
    repositories {
        google()
        mavenCentral()
    }
}
```

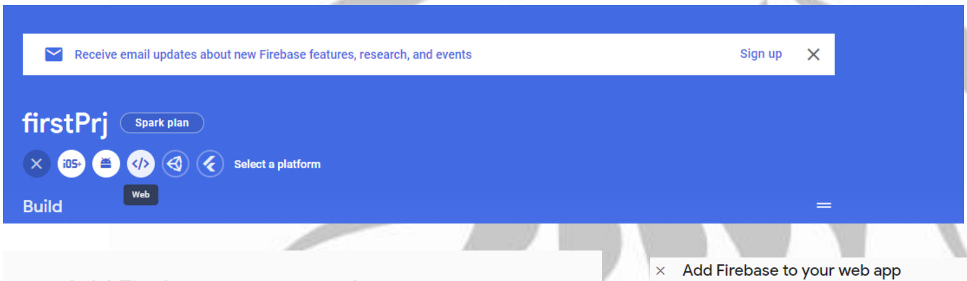
Then, open **android/app/build.gradle** and add the following plugin after the ‘`apply from’` entry:

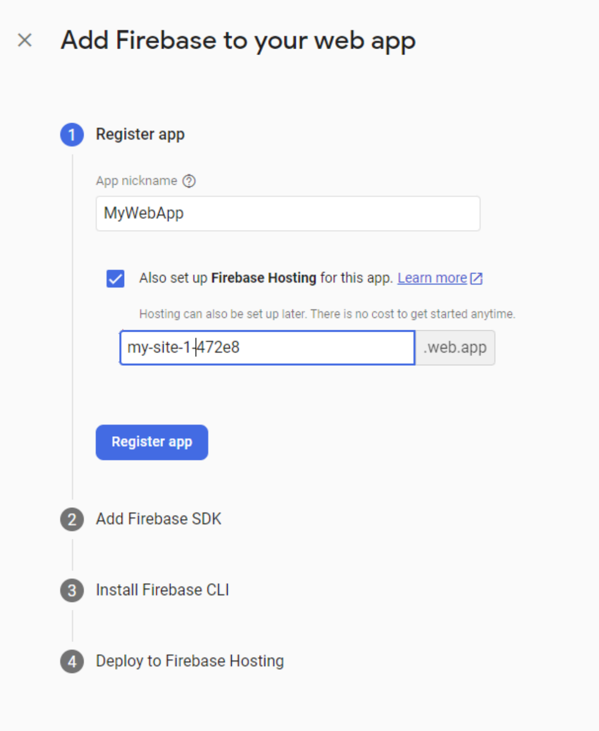
```
apply plugin: 'com.google.gms.google-services'
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply from: "$flutterRoot/packages/flutter_tools/gradle/flutter.gradle"
apply plugin: 'com.google.gms.google-services'

android {
    compileSdkVersion 30
```



```

apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply from: "$flutterRoot/packages/flutter_tools/gradle/flutter.gradle"
apply plugin: 'com.google.gms.google-services'

android {
    compileSdkVersion flutter.compileSdkVersion
    ndkVersion flutter.ndkVersion

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }

    kotlinOptions {
        jvmTarget = '1.8'
    }
```

Registering on Web: In the web section, copy the firebaseConfig details


Receive email updates about new Firebase features, research, and events
Sign up
firstPrj
Spark plan
iOS+
Web
Select a platform
Build



1 Register app

App nickname

MyWebApp

☑ Also set up **Firebase Hosting** for this app. Learn more

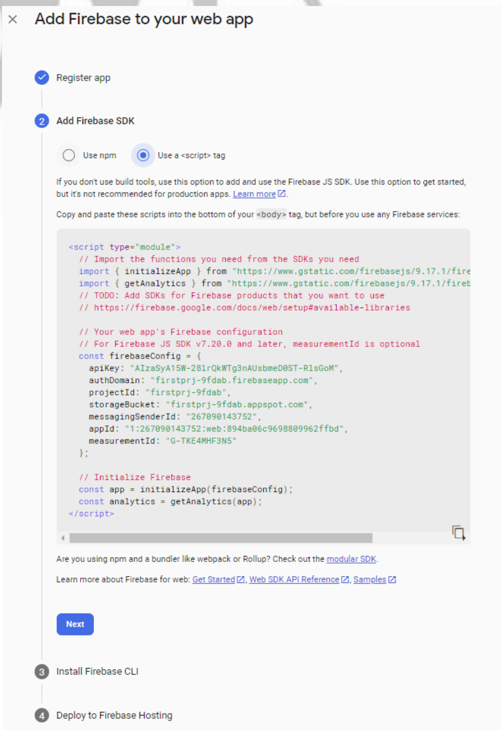
Hosting can also be set up later. There is no cost to get started anytime.

my-site-1472e8 .web.app

2 Add Firebase SDK

3 Install Firebase CLI

4 Deploy to Firebase Hosting



✓ Register app

2 Add Firebase SDK

○ Use npm ● Use a <script> tag

If you don't use build tools, use this option to add and use the Firebase JS SDK. Use this option to get started, but it's not recommended for production apps. Learn more.

Copy and paste these scripts into the bottom of your <body> tag, but before you use any Firebase services:

```
<script type="module">
  // Import the functions you need from the SDKs you need
  import { initializeApp } from "https://www.gstatic.com/firebasejs/9.17.1/fire
  import { getAnalytics } from "https://www.gstatic.com/firebasejs/9.17.1/fireb
  // TODO: Add SDKs for Firebase products that you want to use
  // https://firebase.google.com/docs/web/setup#available-libraries

  // Your web app's Firebase configuration
  // For Firebase JS SDK v7.20.0 and later, measurementId is optional
  const firebaseConfig = {
    apiKey: "AIzaSyA15W-28lrQkWTg3nAUsbmeD0ST-RlsGoM",
    authDomain: "firstprj-9fdab.firebaseapp.com",
    projectId: "firstprj-9fdab",
    storageBucket: "firstprj-9fdab.appspot.com",
    messagingSenderId: "267090143752",
    appId: "1:267090143752:web:894ba06c9698809962ffbd",
    measurementId: "G-TKE4MHF3N5"
  };

  // Initialize Firebase
  const app = initializeApp(firebaseConfig);
  const analytics = getAnalytics(app);
</script>
```

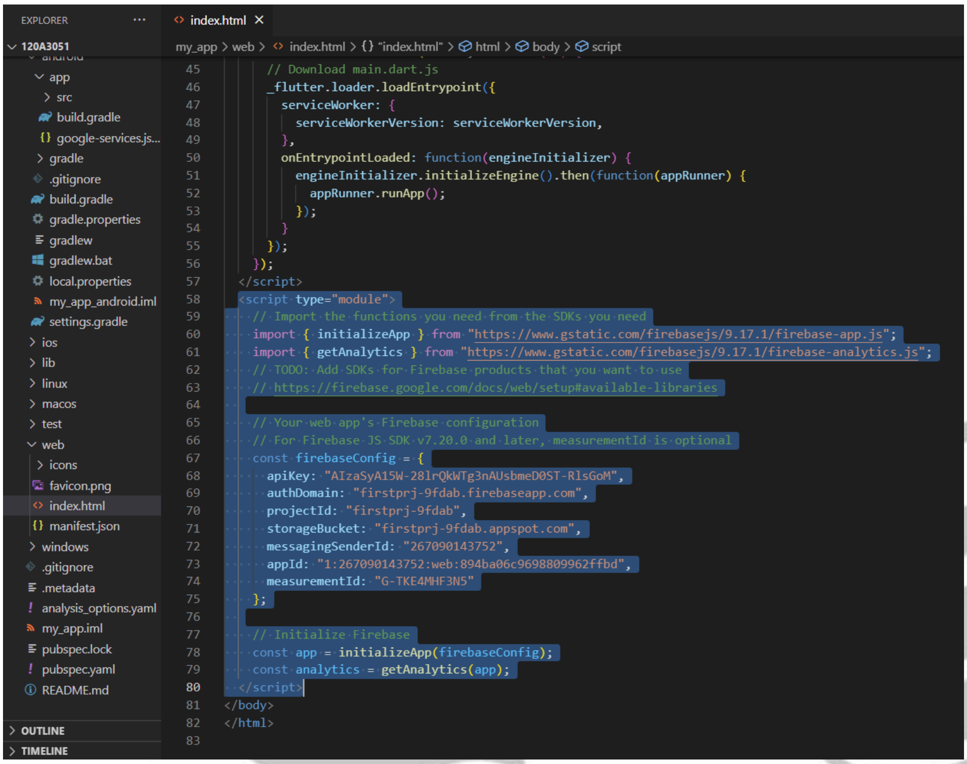
Are you using npm and a bundler like webpack or Rollup? Check out the modular SDK.

Learn more about Firebase for web: Get Started, Web SDK API Reference, Samples

3 Install Firebase CLI

4 Deploy to Firebase Hosting

And paste it inside the index.html using <script> tag as shown below:



```
      // Download main.dart.js
      _flutter.loader.loadEntrypoint({
        serviceWorker: {
          serviceWorkerVersion: serviceWorkerVersion,
        },
        onEntrypointLoaded: function(engineInitializer) {
          engineInitializer.initializeEngine().then(function(appRunner) {
            appRunner.runApp();
          });
        }
      });
    });
  </script>
  <script type="module">
    // Import the functions you need from the SDKs you need
    import { initializeApp } from "https://www.gstatic.com/firebasejs/9.17.1/firebase-app.js";
    import { getAnalytics } from "https://www.gstatic.com/firebasejs/9.17.1/firebase-analytics.js";
    // TODO: Add SDKs for Firebase products that you want to use
    // https://firebase.google.com/docs/web/setup#available-libraries

    // Your web app's Firebase configuration
    // For Firebase JS SDK v7.20.0 and later, measurementId is optional
    const firebaseConfig = {
      apiKey: "AIzaSyA15W-28lrQkWTg3nAUsbmeD0ST-RlsGoM",
      authDomain: "firstprj-9fdab.firebaseapp.com",
      projectId: "firstprj-9fdab",
      storageBucket: "firstprj-9fdab.appspot.com",
      messagingSenderId: "267090143752",
      appId: "1:267090143752:web:894ba06c9698809962ffbd",
      measurementId: "G-TKE4MHF3N5"
    };

    // Initialize Firebase
    const app = initializeApp(firebaseConfig);
    const analytics = getAnalytics(app);
  </script>
</body>
</html>
```

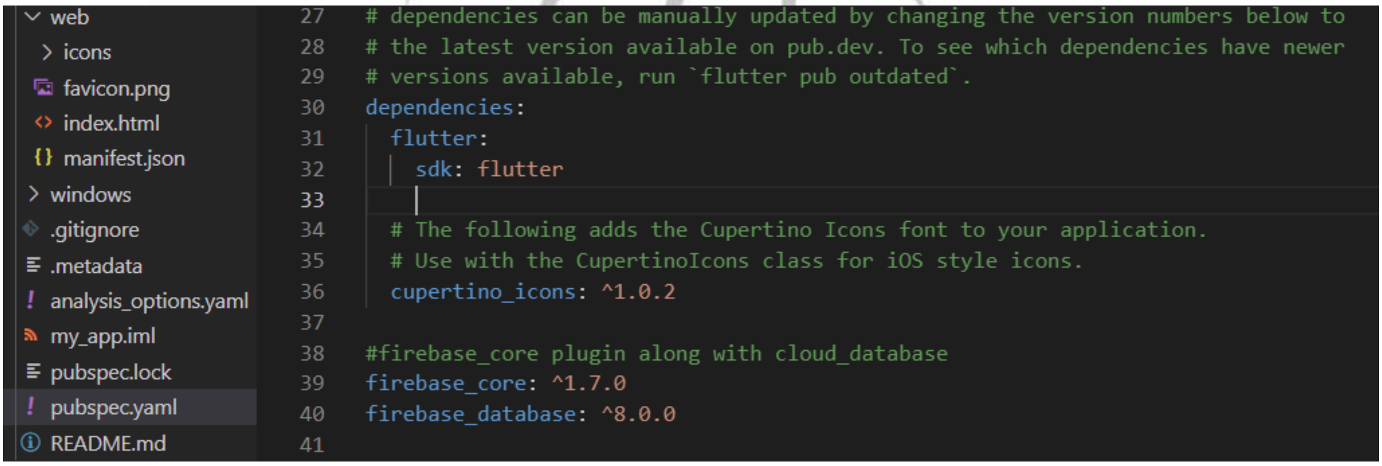
Initializing Firebase App

Before you use Firebase services, you'll need to initialize your Firebase App. To do this, you'll need to add the **firebase_core** plugin along with cloud_database.

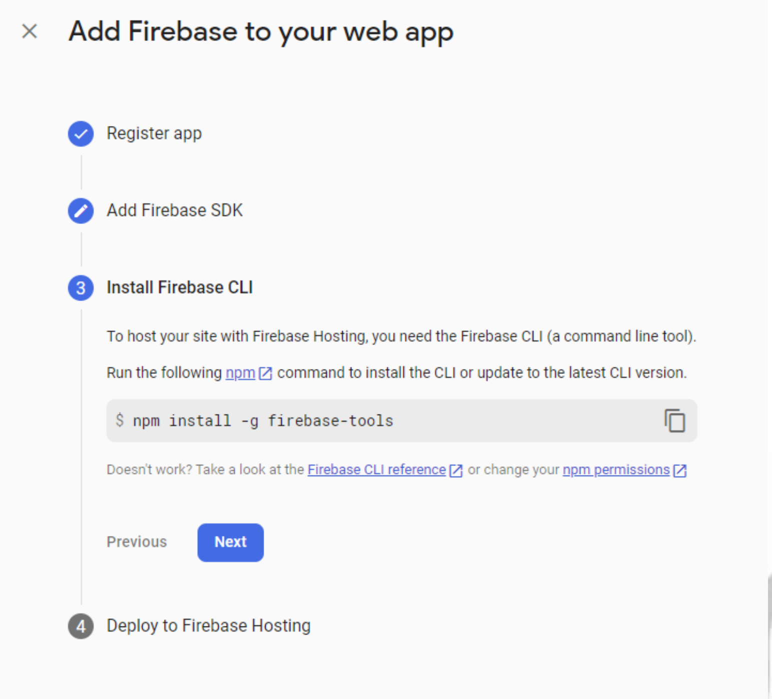
Open the **pubspec.yaml** file in your project and add the following dependencies, then click Pub get:

firebase_core: ^1.7.0

firebase_database: ^8.0.0



```
# dependencies can be manually updated by changing the version numbers below to
# the latest version available on pub.dev. To see which dependencies have newer
# versions available, run `flutter pub outdated`.
dependencies:
  flutter:
    sdk: flutter

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2

#firebase_core plugin along with cloud_database
firebase_core: ^1.7.0
firebase_database: ^8.0.0

```



Add Firebase to your web app

✓ Register app

✓ Add Firebase SDK

3 Install Firebase CLI

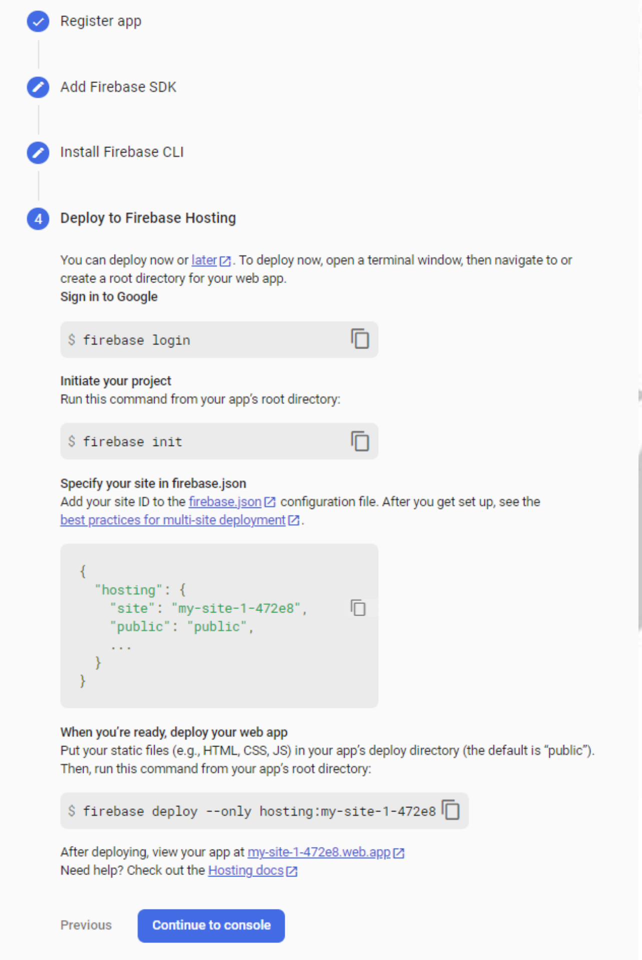
To host your site with Firebase Hosting, you need the Firebase CLI (a command line tool).

Run the following npm command to install the CLI or update to the latest CLI version.

```
$ npm install -g firebase-tools
```

Doesn't work? Take a look at the Firebase CLI reference or change your npm permissions

4 Deploy to Firebase Hosting



✓ Register app

✓ Add Firebase SDK

✓ Install Firebase CLI

4 Deploy to Firebase Hosting

You can deploy now or later. To deploy now, open a terminal window, then navigate to or create a root directory for your web app.

Sign in to Google

```
$ firebase login
```

Initiate your project

Run this command from your app's root directory:

```
$ firebase init
```

Specify your site in firebase.json

Add your site ID to the firebase.json configuration file. After you get set up, see the best practices for multi-site deployment.

```
{
  "hosting": {
    "site": "my-site-1-472e8",
    "public": "public",
    ...
  }
}
```

When you're ready, deploy your web app

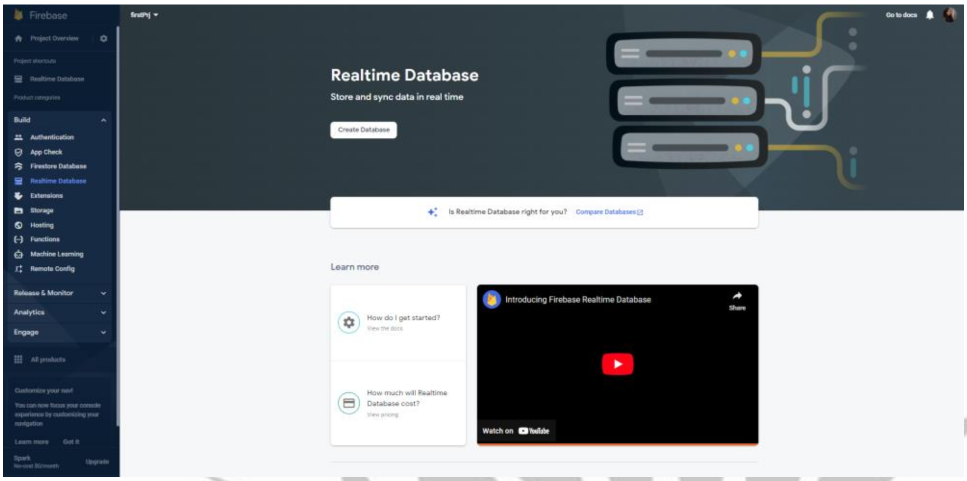
Put your static files (e.g., HTML, CSS, JS) in your app's deploy directory (the default is "public"). Then, run this command from your app's root directory:

```
$ firebase deploy --only hosting:my-site-1-472e8
```

After deploying, view your app at my-site-1-472e8.web.app

Need help? Check out the Hosting docs

Firebase Console project setup:



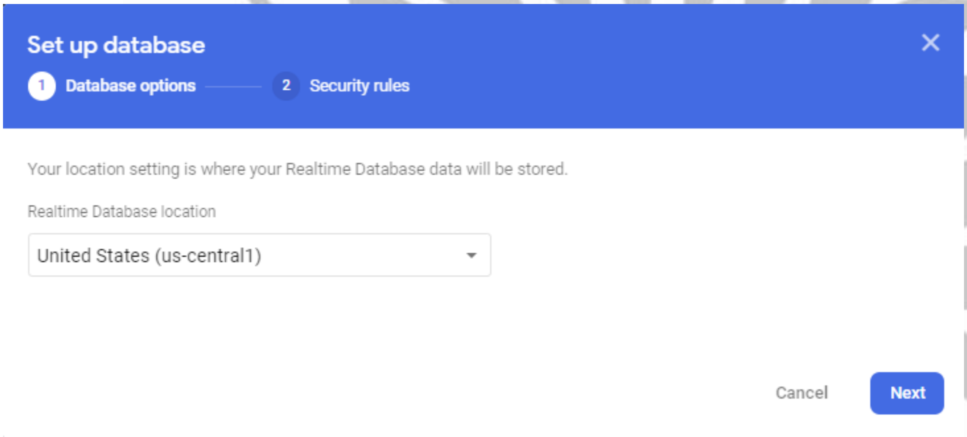
Realtime Database

Store and sync data in real time

Is Realtime Database right for you?

How do I get started?
View the docs

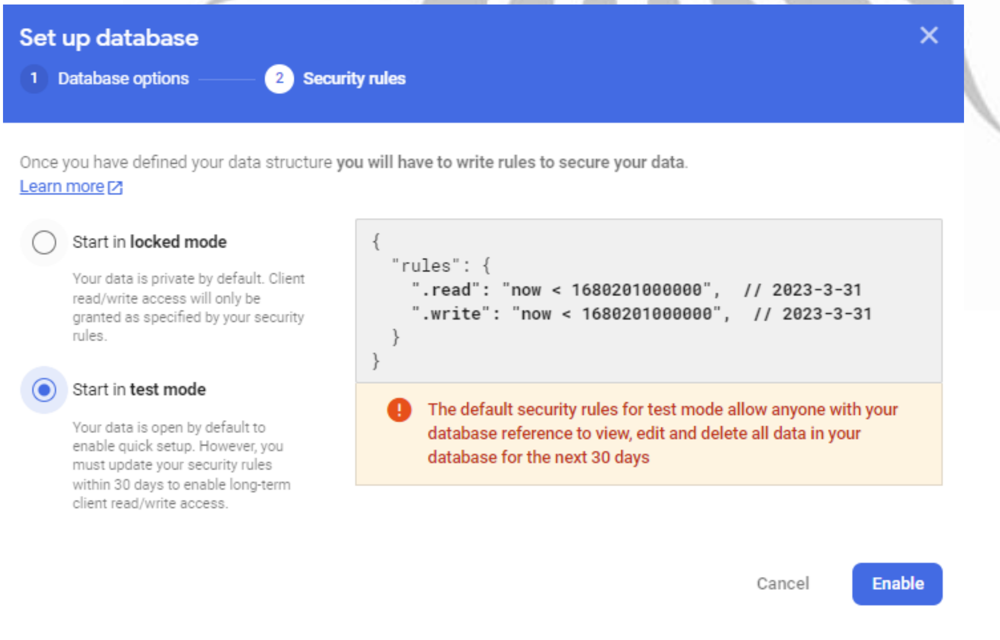
How much will Realtime Database cost?
View pricing



Set up database

Your location setting is where your Realtime Database data will be stored.

Realtime Database location: United States (us-central1)



Set up database

Once you have defined your data structure **you will have to write rules to secure your data.**

○ Start in **locked mode**

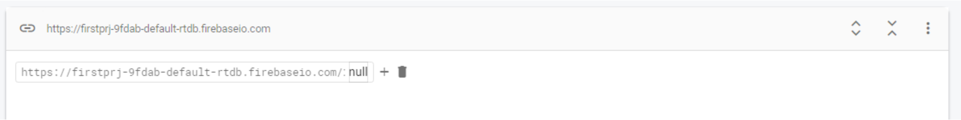
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

● Start in **test mode**

Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
{
  "rules": {
    ".read": "now < 1680201000000",  // 2023-3-31
    ".write": "now < 1680201000000",  // 2023-3-31
  }
}
```

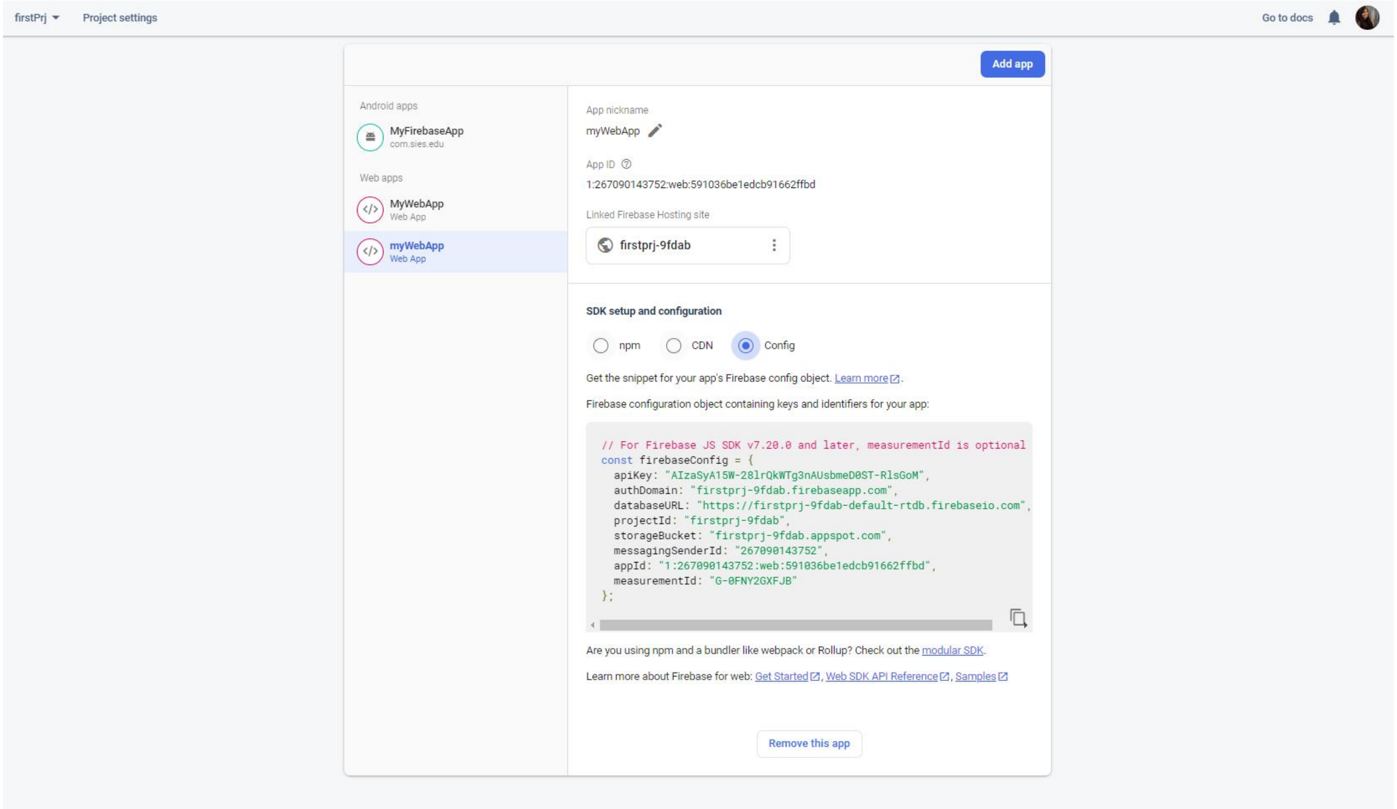
The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days



https://firstprj-9fdab-default-rtdb.firebaseio.com

```
https://firstprj-9fdab-default-rtdb.firebaseio.com/: null
```

Go to your Firebase Project Settings and copy the **config code** to be added into your flutter app ie. main.dart



App nickname
myWebApp

App ID
1:267090143752:web:591036be1edcb91662ffbd

Linked Firebase Hosting site
firstprj-9fdab

SDK setup and configuration

○ npm ○ CDN ● Config

Get the snippet for your app's Firebase config object. Learn more.

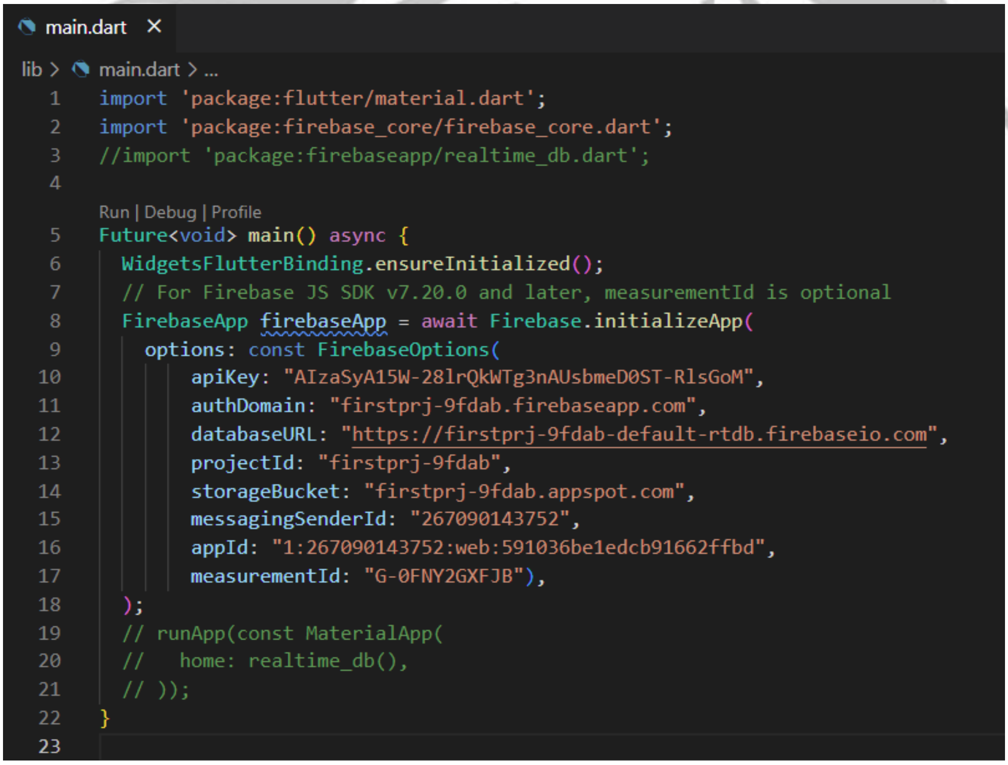
Firebase configuration object containing keys and identifiers for your app:

```
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyA15W-28lrQkWTg3nAUsbmeD0ST-RlsGoM",
  authDomain: "firstprj-9fdab.firebaseapp.com",
  databaseURL: "https://firstprj-9fdab-default-rtdb.firebaseio.com",
  projectId: "firstprj-9fdab",
  storageBucket: "firstprj-9fdab.appspot.com",
  messagingSenderId: "267090143752",
  appId: "1:267090143752:web:591036be1edcb91662ffbd",
  measurementId: "G-0FNY2GXFJB"
};
```

Are you using npm and a bundler like webpack or Rollup? Check out the modular SDK.

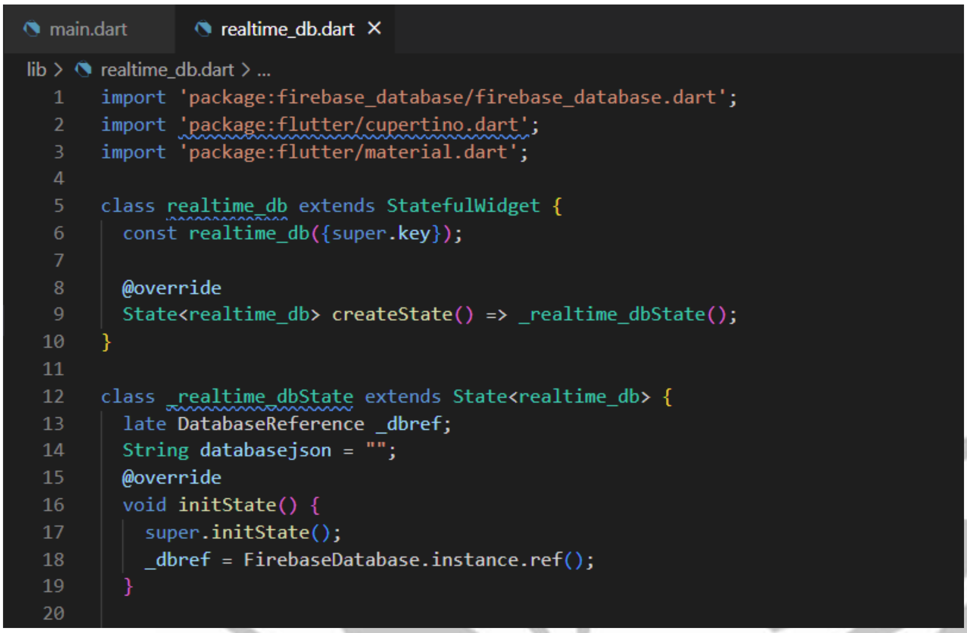
Learn more about Firebase for web: Get Started, Web SDK API Reference, Samples

Your **main.dart** should have the following code:

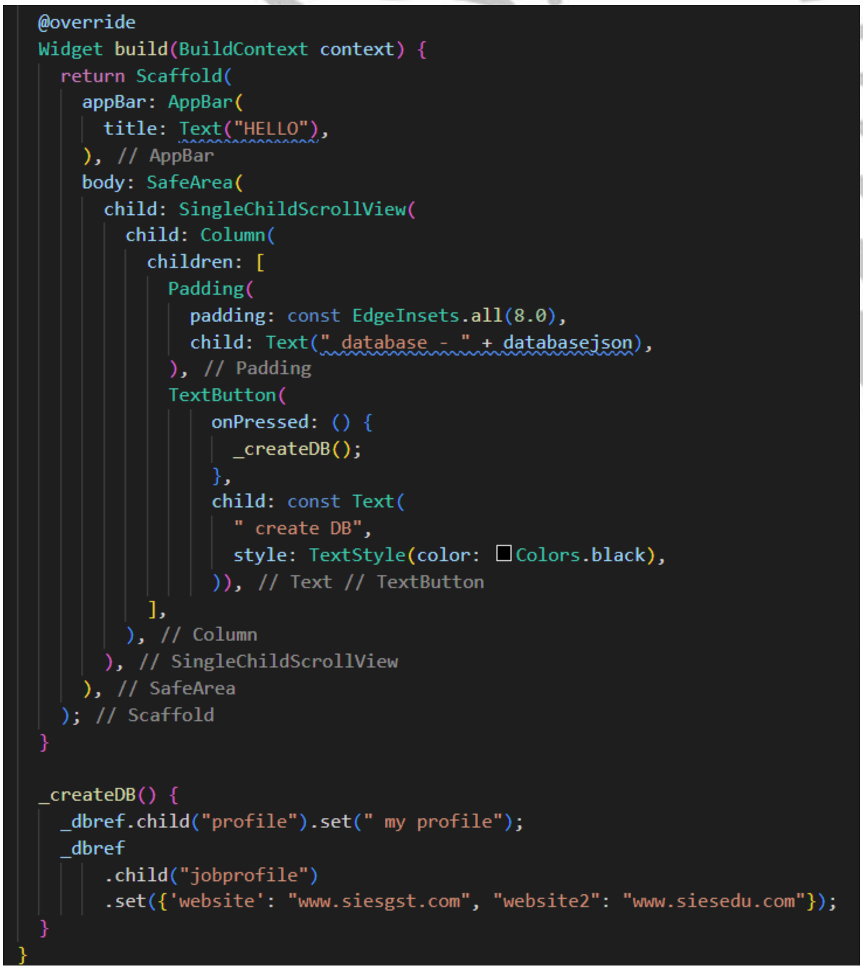


```dart
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
//import 'package:firebaseapp/realtime_db.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  // For Firebase JS SDK v7.20.0 and later, measurementId is optional
  FirebaseApp firebaseApp = await Firebase.initializeApp(
    options: const FirebaseOptions(
        apiKey: "AIzaSyA15W-28lrQkWTg3nAUsbmeD0ST-RlsGoM",
        authDomain: "firstprj-9fdab.firebaseapp.com",
        databaseURL: "https://firstprj-9fdab-default-rtdb.firebaseio.com",
        projectId: "firstprj-9fdab",
        storageBucket: "firstprj-9fdab.appspot.com",
        messagingSenderId: "267090143752",
        appId: "1:267090143752:web:591036be1edcb91662ffbd",
        measurementId: "G-0FNY2GXFJB"),
  );
  // runApp(const MaterialApp(
  //   home: realtime_db(),
  // ));
}
```

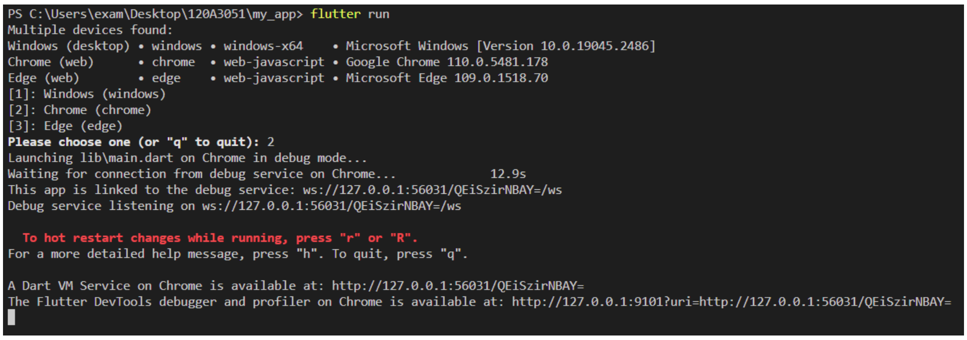
Create an another file as **realtime_db.dart** inside your project and write the following code as shown:



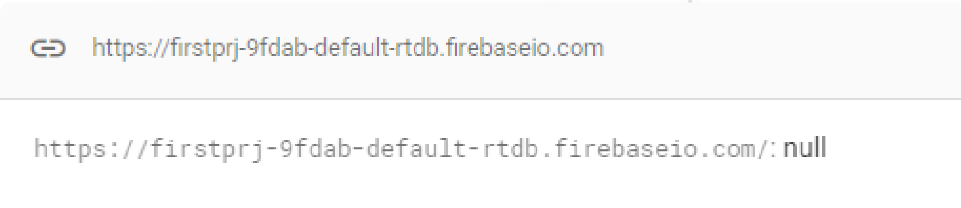
```dart
import 'package:firebase_database/firebase_database.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';

class realtime_db extends StatefulWidget {
  const realtime_db({super.key});

  @override
  State<realtime_db> createState() => _realtime_dbState();
}

class _realtime_dbState extends State<realtime_db> {
  late DatabaseReference _dbref;
  String databasejson = "";
  @override
  void initState() {
    super.initState();
    _dbref = FirebaseDatabase.instance.ref();
  }

```



```dart
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("HELLO"),
      ), // AppBar
      body: SafeArea(
        child: SingleChildScrollView(
          child: Column(
            children: [
              Padding(
                padding: const EdgeInsets.all(8.0),
                child: Text(" database - " + databasejson),
              ), // Padding
              TextButton(
                  onPressed: () {
                    _createDB();
                  },
                  child: const Text(
                    " create DB",
                    style: TextStyle(color: □Colors.black),
                  )), // Text // TextButton
            ],
          ), // Column
        ), // SingleChildScrollView
      ), // SafeArea
    ); // Scaffold
  }

  _createDB() {
    _dbref.child("profile").set(" my profile");
    _dbref
        .child("jobprofile")
        .set({'website': "www.siesgst.com", "website2": "www.siesedu.com"});
  }
}
```

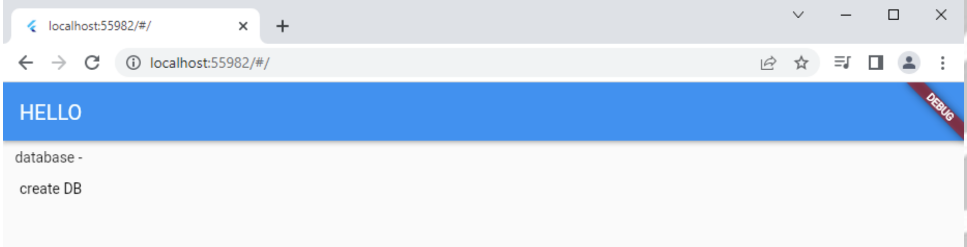


```
PS C:\Users\exam\Desktop\120A3051\my_app> flutter run
Multiple devices found:
Windows (desktop) • windows • windows-x64    • Microsoft Windows [Version 10.0.19045.2486]
Chrome (web)      • chrome  • web-javascript • Google Chrome 110.0.5481.178
Edge (web)        • edge    • web-javascript • Microsoft Edge 109.0.1518.70
[1]: Windows (windows)
[2]: Chrome (chrome)
[3]: Edge (edge)
Please choose one (or "q" to quit): 2
Launching lib\main.dart on Chrome in debug mode...
Waiting for connection from debug service on Chrome...              12.9s
This app is linked to the debug service: ws://127.0.0.1:56031/QEiSzirNBAY=/ws
Debug service listening on ws://127.0.0.1:56031/QEiSzirNBAY=/ws

  To hot restart changes while running, press "r" or "R".
For a more detailed help message, press "h". To quit, press "q".

A Dart VM Service on Chrome is available at: http://127.0.0.1:56031/QEiSzirNBAY=
The Flutter DevTools debugger and profiler on Chrome is available at: http://127.0.0.1:9101?uri=http://127.0.0.1:56031/QEiSzirNBAY=
```



https://firstprj-9fdab-default-rtdb.firebaseio.com

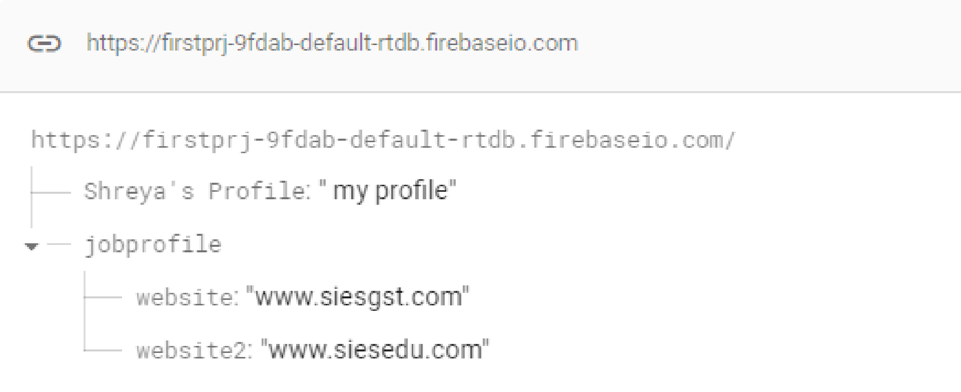
https://firstprj-9fdab-default-rtdb.firebaseio.com/: null



HELLO

database -

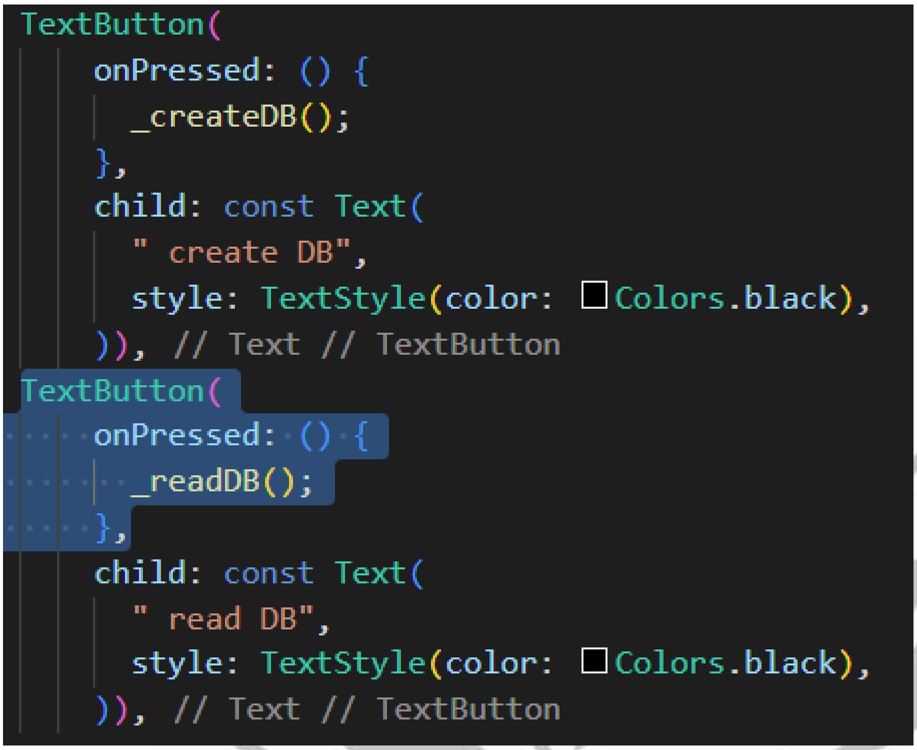
create DB



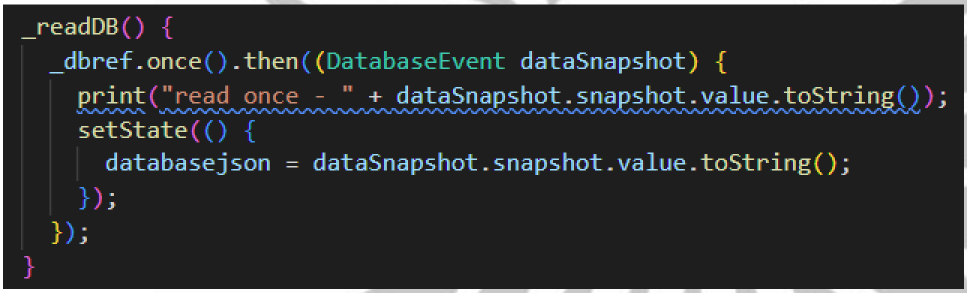
https://firstprj-9fdab-default-rtdb.firebaseio.com

```
https://firstprj-9fdab-default-rtdb.firebaseio.com/
├── Shreya's Profile: "my profile"
└── jobprofile
    ├── website: "www.siesgst.com"
    └── website2: "www.siesedu.com"
```

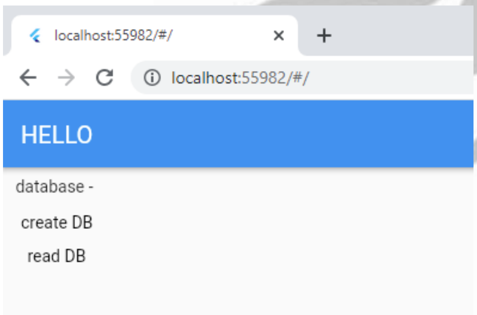
Read all the data at once: Extend the code of **realtime_db.dart** as shown below



```
TextButton(
    onPressed: () {
      _createDB();
    },
    child: const Text(
      " create DB",
      style: TextStyle(color: Colors.black),
    )), // Text // TextButton
TextButton(
    onPressed: () {
      _readDB();
    },
    child: const Text(
      " read DB",
      style: TextStyle(color: Colors.black),
    )), // Text // TextButton
```



```
_readDB() {
  _dbref.once().then((DatabaseEvent dataSnapshot) {
    print("read once - " + dataSnapshot.snapshot.value.toString());
    setState(() {
      databasejson = dataSnapshot.snapshot.value.toString();
    });
  });
}
```

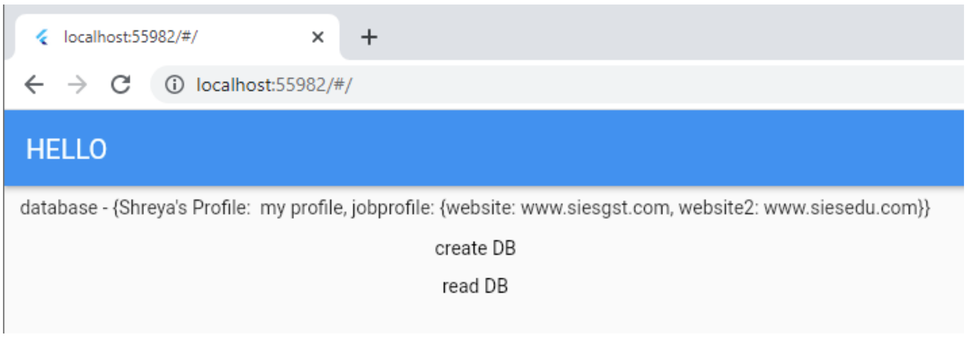


HELLO

database -

create DB

read DB



HELLO

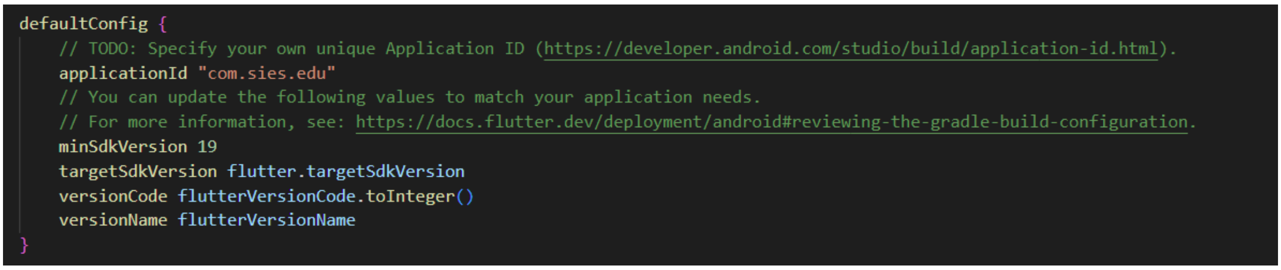
database - {Shreya's Profile: my profile, jobprofile: {website: www.siesgst.com, website2: www.siesedu.com}}

create DB

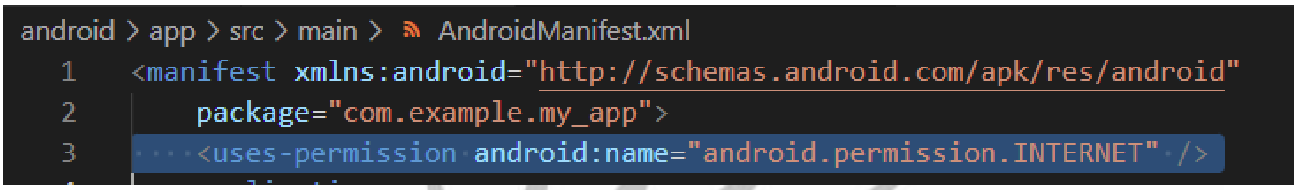
read DB

Realtime Database on android emulator:

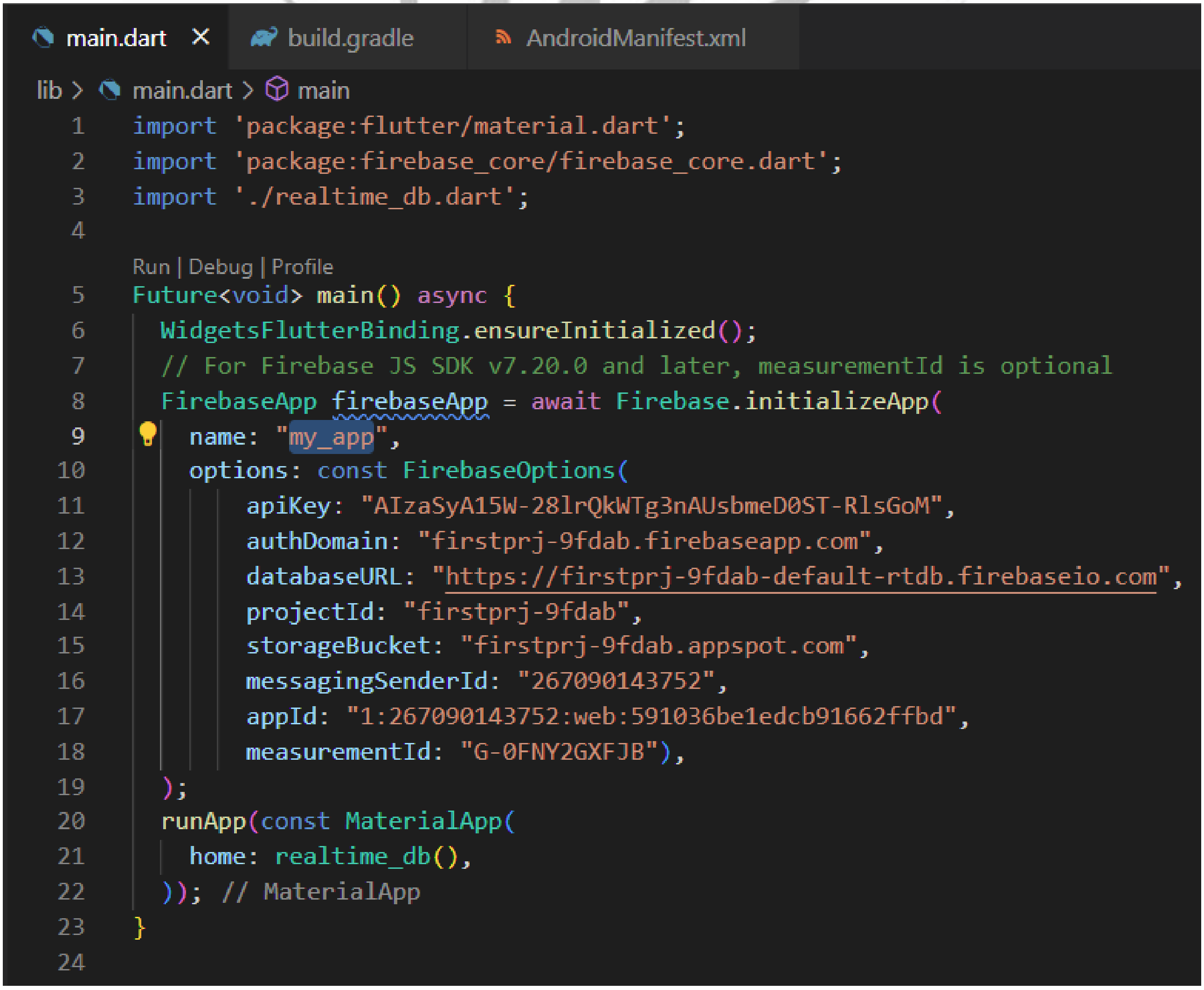
Android->app->build.gradle



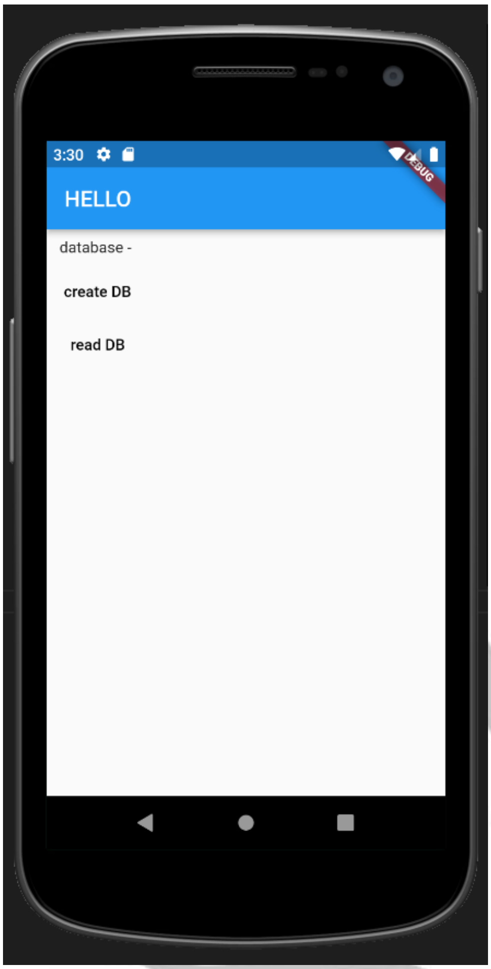
```
defaultConfig {
    // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).
    applicationId "com.sies.edu"
    // You can update the following values to match your application needs.
    // For more information, see: https://docs.flutter.dev/deployment/android#reviewing-the-gradle-build-configuration.
    minSdkVersion 19
    targetSdkVersion flutter.targetSdkVersion
    versionCode flutterVersionCode.toInteger()
    versionName flutterVersionName
}
```



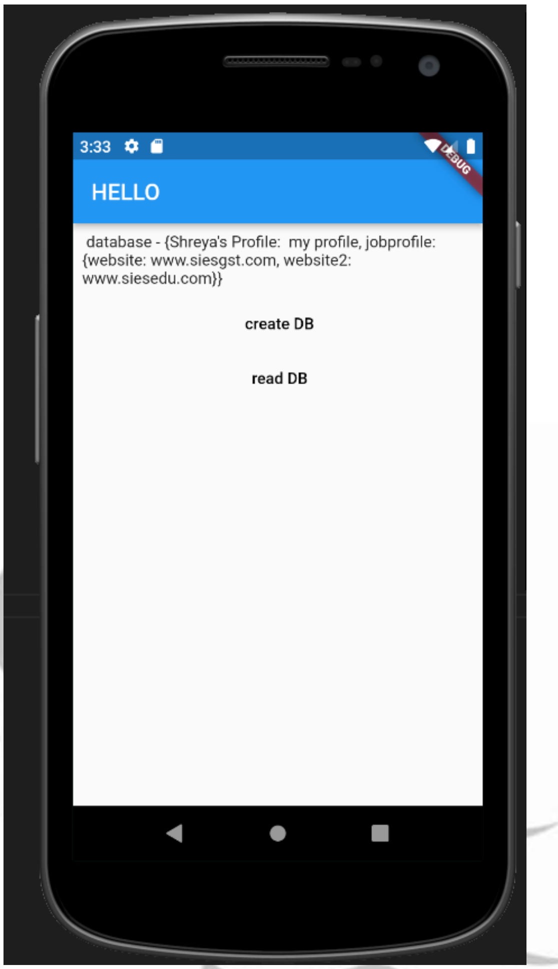
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.my_app">
    <uses-permission android:name="android.permission.INTERNET" />
```



```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import './realtime_db.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  // For Firebase JS SDK v7.20.0 and later, measurementId is optional
  FirebaseApp firebaseApp = await Firebase.initializeApp(
    name: "my_app",
    options: const FirebaseOptions(
        apiKey: "AIzaSyA15W-28lrQkWTg3nAUsbmeD0ST-RlsGoM",
        authDomain: "firstprj-9fdab.firebaseapp.com",
        databaseURL: "https://firstprj-9fdab-default-rtdb.firebaseio.com",
        projectId: "firstprj-9fdab",
        storageBucket: "firstprj-9fdab.appspot.com",
        messagingSenderId: "267090143752",
        appId: "1:267090143752:web:591036be1edcb91662ffbd",
        measurementId: "G-0FNY2GXFJB"),
  );
  runApp(const MaterialApp(
    home: realtime_db(),
  )); // MaterialApp
}

```



database -



database - {Shreya's Profile: my profile, jobprofile: {website: www.siesgst.com, website2: www.siesedu.com}}

Conclusion: Successfully connected Flutter UI with Firebase realtime database.