

Exp 6

Aim :- To connect Flutter UI with Firebase Database

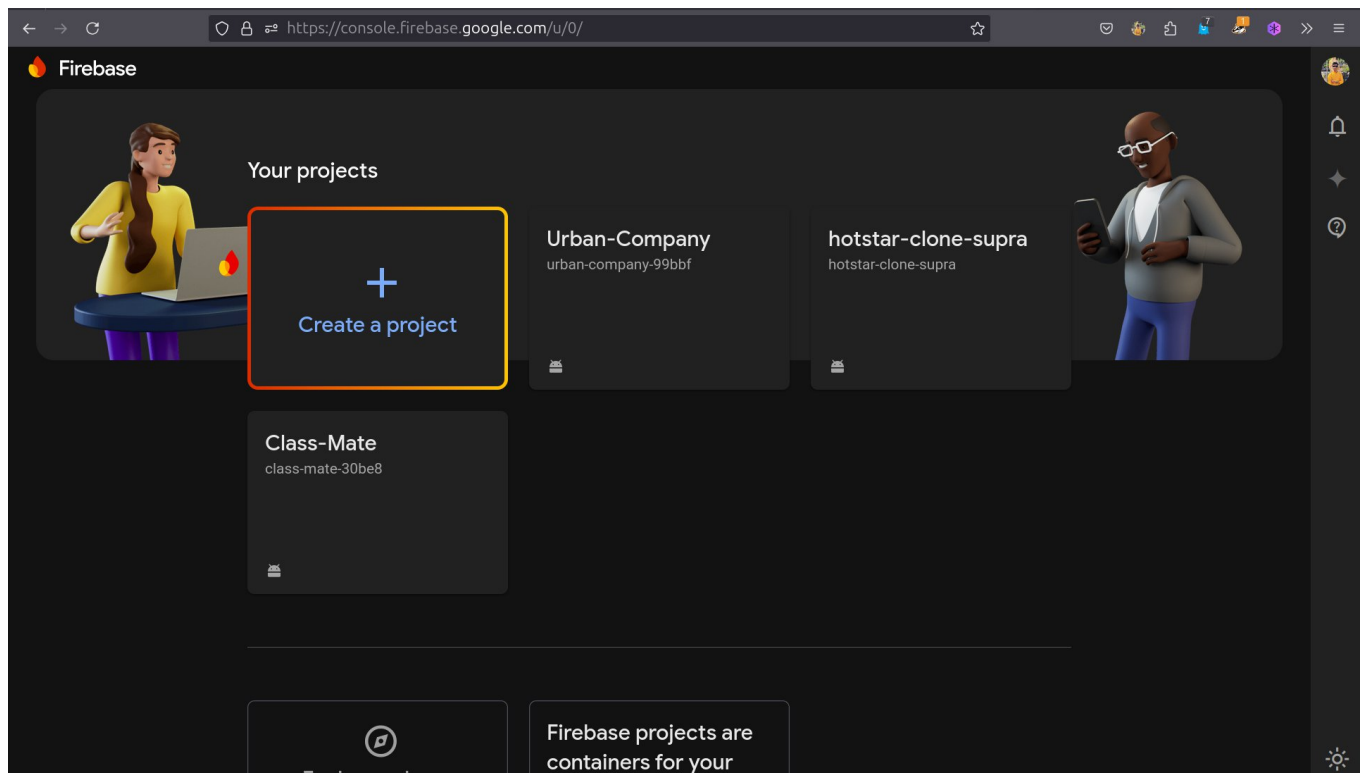
Theory :-

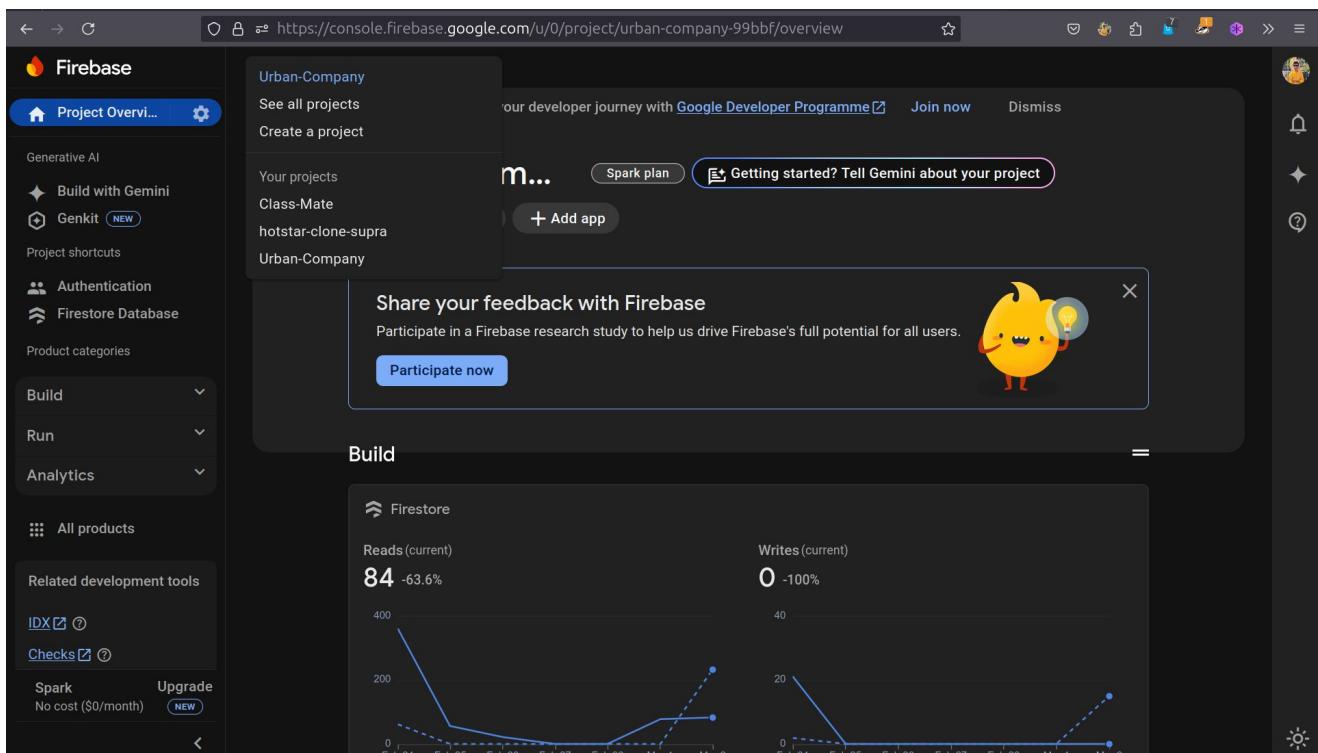
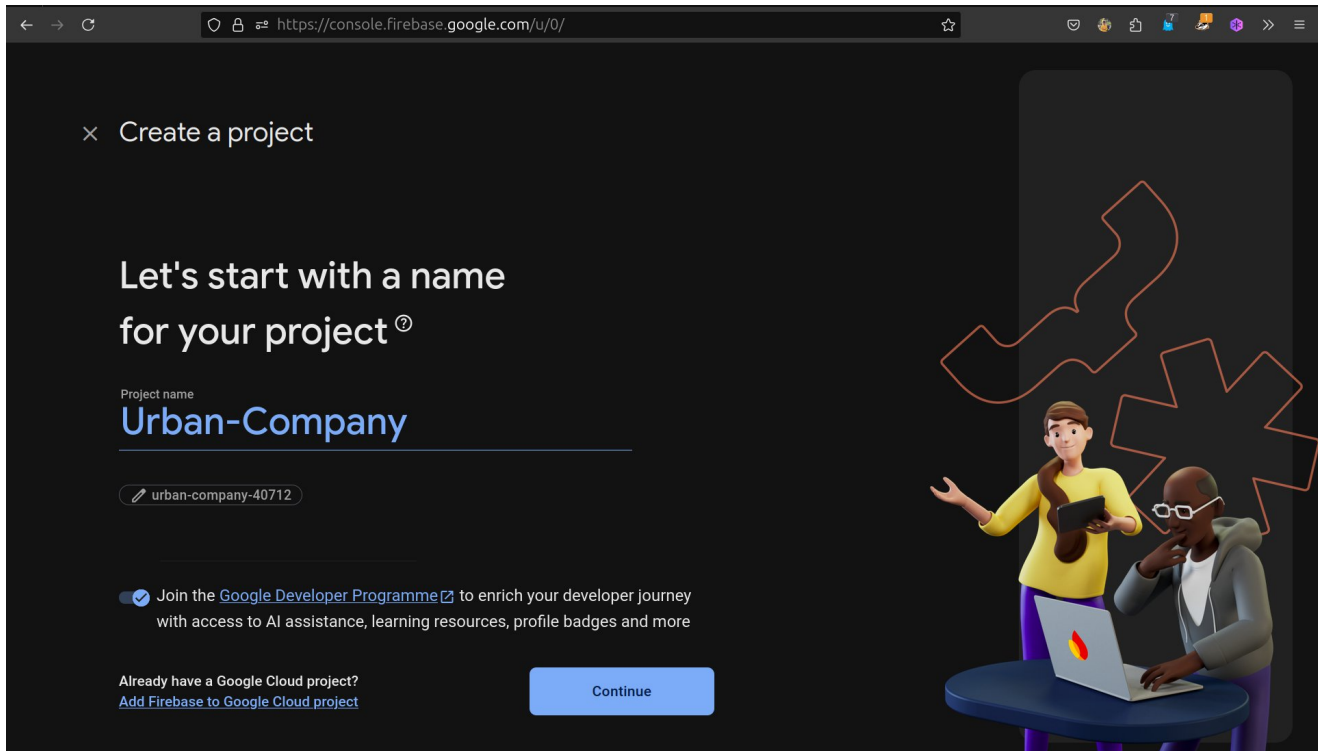
Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. Firebase, a Backend-as-a-Service (BaaS) platform, provides real-time database, authentication, and cloud storage services, making it a powerful backend solution for Flutter applications.

By integrating Firebase with Flutter, developers can store and retrieve data in real time, authenticate users, and manage cloud-based data efficiently. This is particularly useful for applications requiring dynamic content updates and user interactions.

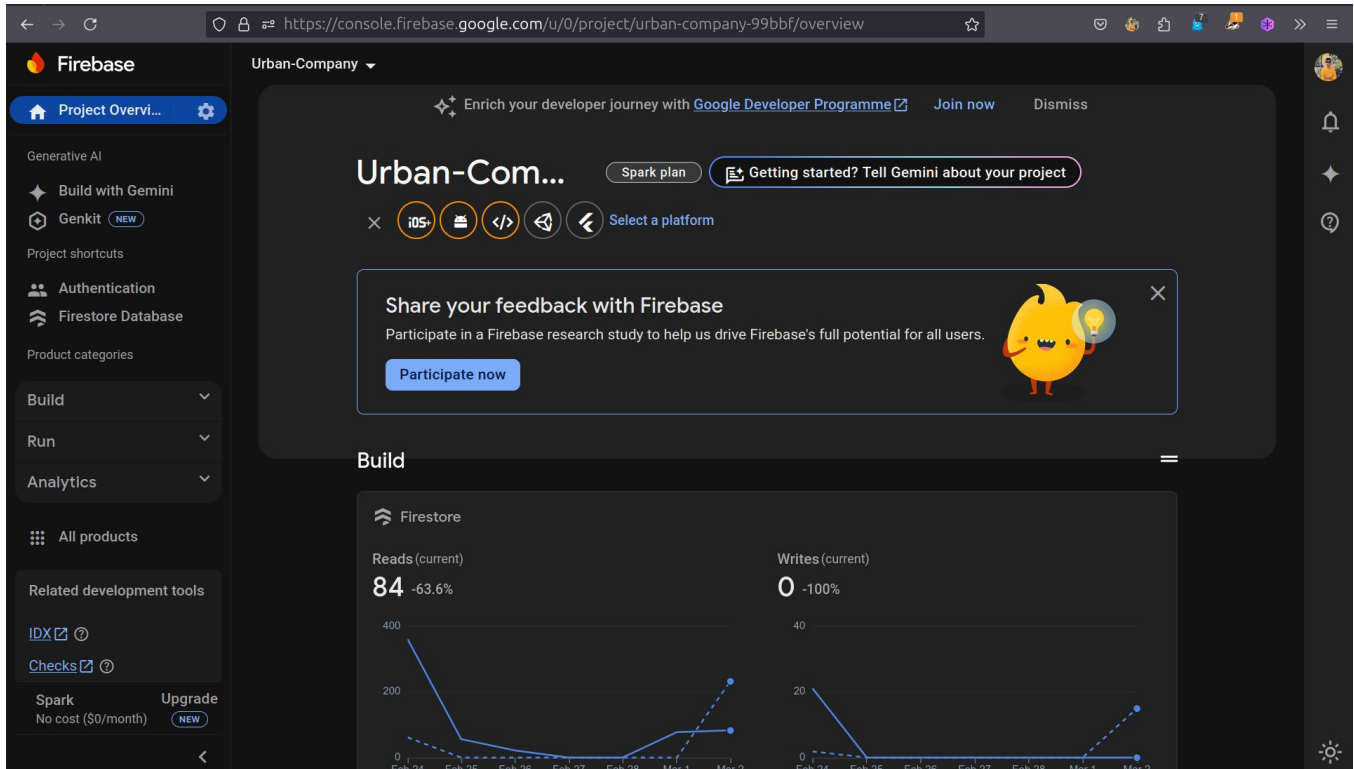
Practical :-

1. Creating new app in Firebase console





2. Adding Firebase dependencies and SDK to Flutter app



Go to docs

×

Add Firebase to your Android app

1 Register app

Android package name

com.company.urban

App nickname(optional)

UrbanCompany

Debug signing certificate SHA-1 (optional)

00:00:00:00:00:00:00:00:00:00:00:00:00:00:C

ⓘ

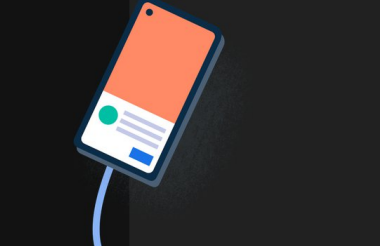
Required for Dynamic Links and Google Sign-In or phone number support in Auth. Edit SHA-1s in settings.

Register app

2 Download and then add config file

3 Add Firebase SDK

4 Next steps



× Add Firebase to your Android app

1 Register app

Android package name: com.example.agriapp, app nickname: AgriApp

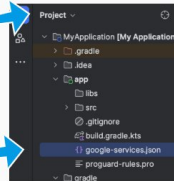
2 Download and then add config file

Instructions for Android Studio below | [Unity](#) [C++](#)

[Download google-services.json](#)

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

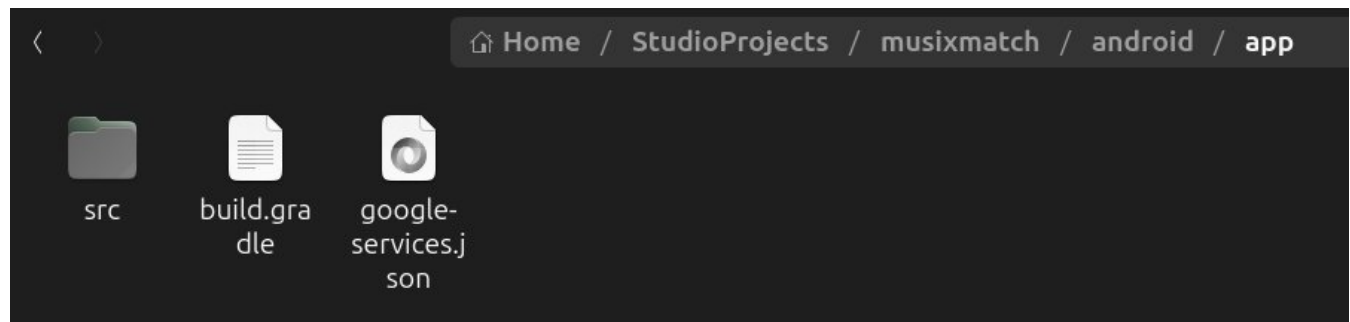
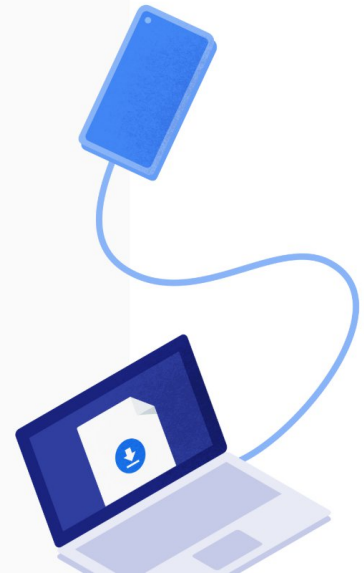
Move your downloaded `google-services.json` file into your module (app-level) root directory.



[Next](#)

3 Add Firebase SDK

4 Next steps



3 Add Firebase SDK

Instructions for Gradle | [Unity](#) [C++](#)

★ Are you still using the **buildscript** syntax to manage plug-ins? Learn how to [add Firebase plug-ins](#) using that syntax.

1. To make the `google-services.json` config values accessible to Firebase SDKs, you need the Google services Gradle plug-in.

☐ Kotlin DSL (build.gradle.kts) ☒ Groovy (build.gradle)

Add the plug-in as a dependency to your **project-level** `build.gradle` file:

Root-level (project-level) Gradle file (<project>/build.gradle):

```
plugins {
    // ...

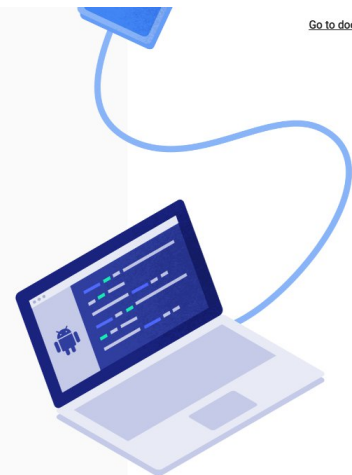
    // Add the dependency for the Google services Gradle plugin
    id 'com.google.gms.google-services' version '4.4.2' apply false
}
```

2. Then, in your **module (app-level)** `build.gradle` file, add both the `google-services` plug-in and any Firebase SDKs that you want to use in your app:

Module (app-level) Gradle file (<project>/<app-module>/build.gradle):

```
plugins {
    id 'com.android.application'
    // Add the Google services Gradle plugin
    id 'com.google.gms.google-services'
    ...
}

dependencies {
```



```

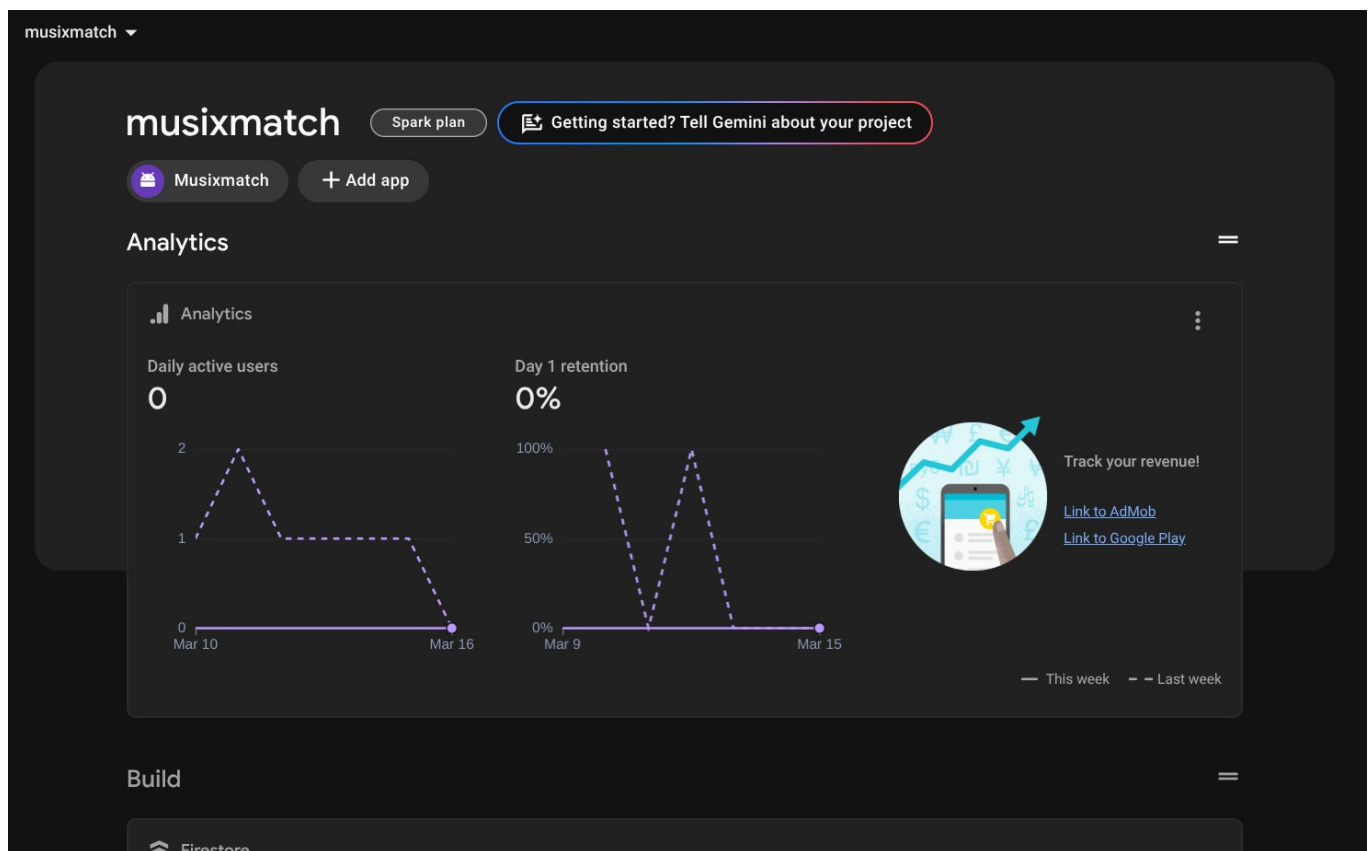
Open  build.gradle
~/StudioProjects/musixmatch/android/app

plugins {
    id "com.android.application"
    id "kotlin-android"
    // The Flutter Gradle Plugin must be applied after the Android and Kotlin Gradle plugins.
    id "dev.flutter.flutter-gradle-plugin"
    id 'com.google.gms.google-services'
}

dependencies {
    implementation platform('com.google.firebase:firebase-bom:33.10.0')
    implementation 'com.google.firebase:firebase-analytics'
}

android {
    namespace "com.example.musixmatch"
    compileSdkVersion flutter.compileSdkVersion
    ndkVersion flutter.ndkVersion

```



3. Configuring Firebase real-time Database

main.dart :-

```
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'navigation_bar.dart'; // Update this import

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Musixmatch',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const MainScreen(), // Change HomePage to MainScreen
    );
  }
}
```

```
}  
}
```

Profile_page.dart :-

```
import 'package:flutter/material.dart';  
import 'package:cloud_firestore/cloud_firestore.dart';  
  
class ProfilePage extends StatefulWidget {  
  const ProfilePage({super.key});  
  
  @override  
  _ProfilePageState createState() => _ProfilePageState();  
}  
  
class _ProfilePageState extends State<ProfilePage> {  
  // Controllers to capture form input  
  final TextEditingController nameController = TextEditingController();  
  final TextEditingController nicknameController = TextEditingController();  
  final TextEditingController dobController = TextEditingController();  
  final TextEditingController cityController = TextEditingController();  
  String? designation;  
  
  // Submit form data to Firestore  
  Future<void> submitForm() async {  
    // Gather all input data  
    final profileData = {
```

```
'name': nameController.text,  
'nickname': nicknameController.text,  
'dob': dobController.text,  
'designation': designation,  
'city': cityController.text,  
};
```

```
try {  
  // Save data to Firestore under the 'profiles' collection  
  await FirebaseFirestore.instance.collection('profiles').add(profileData);  
  
  // Show a confirmation message  
  ScaffoldMessenger.of(context).showSnackBar(  
    const SnackBar(content: Text('Profile saved successfully')),  
  );  
} catch (e) {  
  // Handle error (e.g., show an error message)  
  ScaffoldMessenger.of(context).showSnackBar(  
    SnackBar(content: Text('Error saving profile: $e')),  
  );  
}  
}
```

@override

```
Widget build(BuildContext context) {  
  return Scaffold(  

```



```
appBar: AppBar(  
  title: const Text('Profile'),  
  leading: IconButton(  
    icon: const Icon(Icons.arrow_back),  
    onPressed: () => Navigator.pop(context),  
  ),  
,  
body: SingleChildScrollView(  
  padding: const EdgeInsets.all(16.0),  
  child: Column(  
    crossAxisAlignment: CrossAxisAlignment.start,  
    children: [  
      // Image Placeholder  
      Center(  
        child: Container(  
          width: 100,  
          height: 100,  
          decoration: BoxDecoration(  
            color: Colors.grey[300],  
            borderRadius: BorderRadius.circular(50),  
          ),  
        child: const Icon(  
          Icons.person,  
          size: 50,  
          color: Colors.grey,  
        ),  
      ],  
    ),  
  ),  
)
```

```
    ),  
    ),  
    const SizedBox(height: 16),  
  
    // Name Field  
    TextField(  
      controller: nameController,  
      decoration: const InputDecoration(  
        labelText: 'Name',  
        border: OutlineInputBorder(),  
      ),  
    ),  
    const SizedBox(height: 16),  
  
    // Nickname Field  
    TextField(  
      controller: nicknameController,  
      decoration: const InputDecoration(  
        labelText: 'Nickname',  
        border: OutlineInputBorder(),  
      ),  
    ),  
    const SizedBox(height: 16),  
  
    // Date of Birth Field  
    TextField(  

```

```
controller: dobController,

decoration: const InputDecoration(

  labelText: 'Date of Birth',

  border: OutlineInputBorder(),

  hintText: 'YYYY-MM-DD',

),

keyboardType: TextInputType.datetime,

),

const SizedBox(height: 16),

// Designation Dropdown

DropDownButtonFormField<String>(

  value: designation,

  decoration: const InputDecoration(

    labelText: 'Designation',

    border: OutlineInputBorder(),

  ),

  items: const [

    DropdownMenuItem(value: 'Singer', child: Text('Singer')),

    DropdownMenuItem(value: 'Drummer', child: Text('Drummer')),

    DropdownMenuItem(value: 'Pianist', child: Text('Pianist')),

    DropdownMenuItem(value: 'Guitar player', child: Text('Guitar player')),

    DropdownMenuItem(value: 'Writer', child: Text('Writer')),

    DropdownMenuItem(value: 'DJ', child: Text('DJ')),

    DropdownMenuItem(value: 'Composer', child: Text('Composer')),

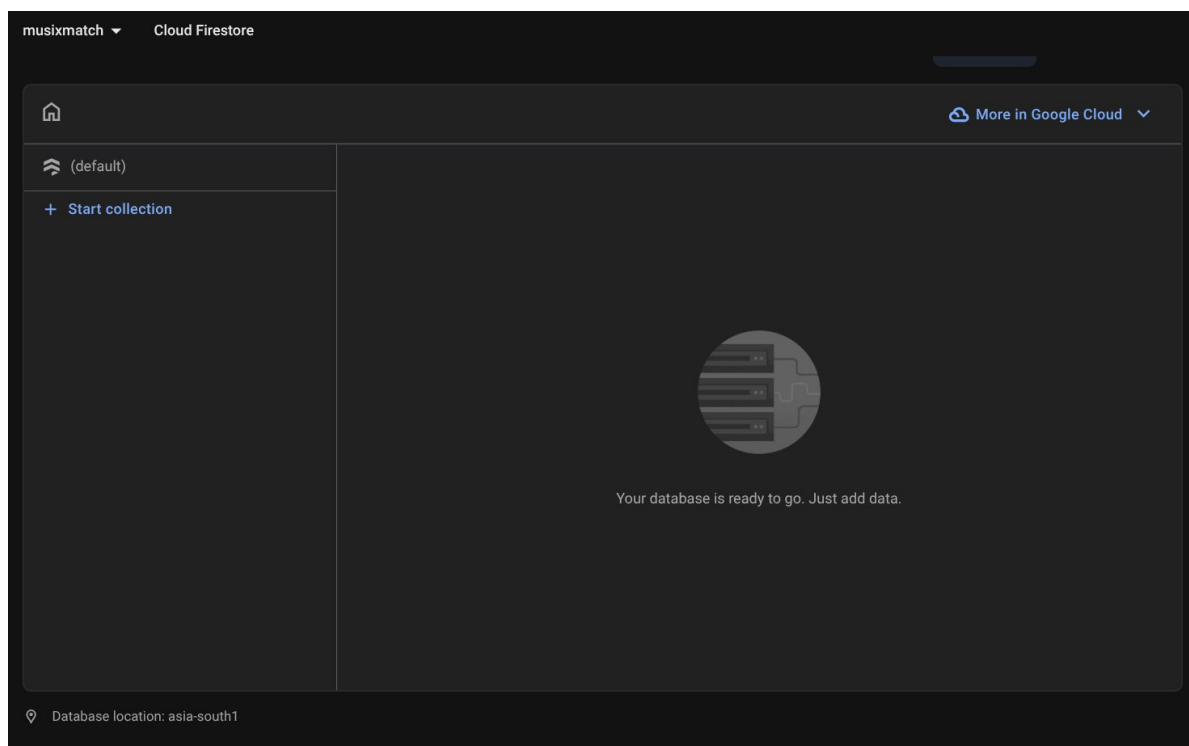
  ],
```

```
        onChanged: (value) {  
          setState(() {  
            designation = value;  
          });  
        },  
      ),  
      const SizedBox(height: 16),  
  
      // Home City Field  
      TextField(  
        controller: cityController,  
        decoration: const InputDecoration(  
          labelText: 'Home City',  
          border: OutlineInputBorder(),  
        ),  
      ),  
      const SizedBox(height: 16),  
  
      // Submit Button  
      ElevatedButton(  
        onPressed: submitForm,  
        child: const Text('Submit'),  
      ),  
    ],  
  ),  
),
```

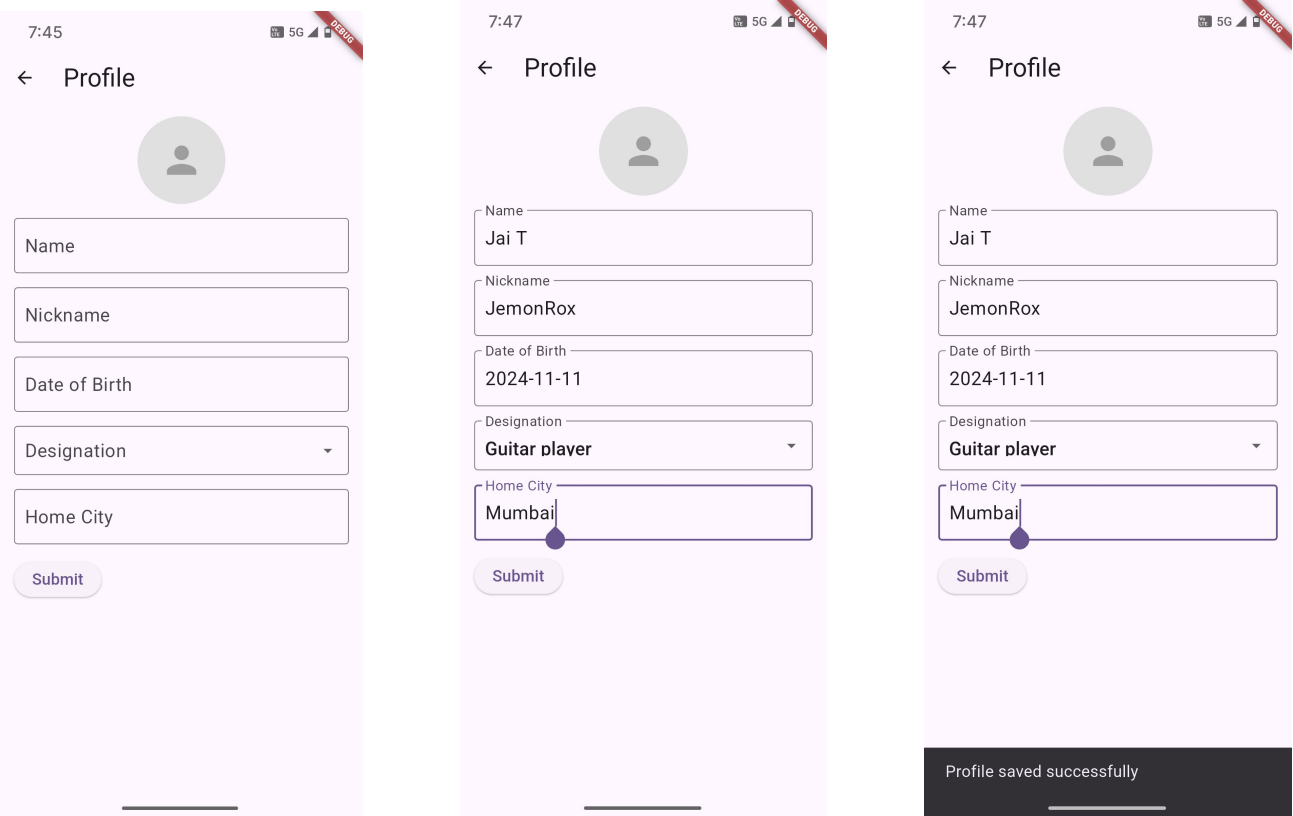
```
);  
}  
}
```

4. Final result

Before filling data in the app



Filling data in the app



After filling data in the app

