

Ameya Chaudhary
D15a
09

EXPT 6

To connect Flutter UI with firebase database

THEORY:

Connecting a Flutter app with Firebase database and implementing email/password authentication through Firebase involves several steps

Set Up Firebase Project:

- Create a new project on the Firebase Console.
- Follow the prompts to add your app to the project. Download the config file (google-services.json for Android or GoogleService-Info.plist for iOS) and place it in the appropriate location in your Flutter project.

Add Dependencies:

- Open your pubspec.yaml file and add the necessary dependencies. For Firebase, you'll typically need:
- yaml
- Copy code

dependencies:

```
firebase_core: ^latest_version  
firebase_auth: ^latest_version  
cloud_firestore: ^latest_version
```

•

Initialize Firebase:

- In your main Dart file or application entry point, initialize Firebase with the config file:
- dart
- Copy code

```
import 'package:firebase_core/firebase_core.dart';
```

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

•

Implementing Authentication

```
import 'package:firebase_auth/firebase_auth.dart';
```

•

Connect to Firestore Database:

- Use the cloud_firestore package to interact with Firestore:
- dart
- Copy code

```
import 'package:cloud_firestore/cloud_firestore.dart';
```

// Example to add data to Firestore

```
Future<void> addUserData(String uid, String name, String email) async {  
  await FirebaseFirestore.instance.collection('users').doc(uid).set({  
    'name': name,  
    'email': email,  
  });  
}
```

•

Use Firebase Services in Your UI:

- Integrate authentication and database functions into your UI to enable user registration, login, and data retrieval.

Handle User Authentication State:

- Implement a mechanism to listen to the authentication state changes and update the UI accordingly. You can use a StreamBuilder or a state management solution like Provider or Bloc

CODE:

//main.dart

```
import 'package:flutter/material.dart';  
import 'package:provider/provider.dart';  
import 'package:first_project/firebase_options.dart';  
import 'package:firebase_core/firebase_core.dart';  
import 'home_screen.dart';  
import 'song_details_screen.dart';  
import 'library_screen.dart';  
import 'login_screen.dart';  
import 'splash_screen.dart';  
import 'package:firebase_auth/firebase_auth.dart';  
import 'package:just_audio/just_audio.dart'; // Import the just_audio package  
import 'liked_songs.dart';
```

```

class AudioPlayerProvider extends ChangeNotifier {
  final AudioPlayer _audioPlayer = AudioPlayer(); // Create an instance of AudioPlayer

  AudioPlayer get audioPlayer => _audioPlayer;
}

```

```

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  try {
    await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);
    runApp(
      MultiProvider(
        providers: [
          ChangeNotifierProvider<AudioPlayerProvider>(
            create: (_) => AudioPlayerProvider(),
          ),
          ChangeNotifierProvider(create: (_) => LikedSongs()),
        ],
        child: MyApp(),
      ),
    );
  } catch (e) {
    print('Error initializing Firebase: $e');
  }
}

```

```

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'youtube_music',
      theme: ThemeData(
        primaryColor: Colors.black,
        scaffoldBackgroundColor: Colors.black,
        textTheme: TextTheme(
          bodyText2: TextStyle(color: Colors.white),
        ),
        appBarTheme: AppBarTheme(
          backgroundColor: Colors.black,
          foregroundColor: Colors.red,
        ),
      ),
    );
  }
}

```

```

        home: (FirebaseAuth.instance.currentUser != null) ? SplashScreen() : LoginScreen(),
    );
}
}

```

// register_screen.dart

```

import 'package:flutter/material.dart';
import 'home_screen.dart';
import 'dart:developer';
import 'package:firebase_auth/firebase_auth.dart';

```

```

class RegisterScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Register'),
        centerTitle: true,
        backgroundColor: Colors.black,
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: SingleChildScrollView( // Wrap with SingleChildScrollView
          child: RegisterForm(),
        ),
      ),
    );
  }
}

```

```

class RegisterForm extends StatefulWidget {
  @override
  _RegisterFormState createState() => _RegisterFormState();
}

```

```

class _RegisterFormState extends State<RegisterForm> {
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  final TextEditingController cpasswordController = TextEditingController();

```

```

  void createAccount() async {
    String email = emailController.text.trim();
    String password = passwordController.text.trim(); // Corrected
    String cPassword = cpasswordController.text.trim(); // Corrected

```

```

if (email == "" || password == "" || cPassword == "") {
  log("Please fill all the details");
} else if (password != cPassword) {
  log("Password mismatched");
} else {
  try {

    UserCredential userCredential = await FirebaseAuth.instance
      .createUserWithEmailAndPassword(email: email, password: password);
    log("User Created!");
    if(userCredential.user != null) {
      Navigator.pop(context);
    }
    // Redirect to HomeScreen or perform any other action
  } catch (e) {
    log("Error creating user: $e");
    // Handle the error, you can log it or show an error message
  }
}
}

```

```

@override
Widget build(BuildContext context) {
  return Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Image.asset('assets/ytlogo2.png', height: 300, width: 300),
      SizedBox(height: 20),
      TextField(
        controller: emailController,
        decoration: InputDecoration(labelText: 'Email'),
      ),
      TextField(
        controller: passwordController,
        decoration: InputDecoration(labelText: 'Password'),
      ),
      TextField(
        controller: cpasswordController,
        obscureText: true,
        decoration: InputDecoration(labelText: 'Confirm Password'),
      ),
      SizedBox(height: 20),
      ElevatedButton(

```

```

        onPressed: () {
          createAccount();
        },
        style: ButtonStyle(
          backgroundColor: MaterialStateProperty.all<Color>(Colors.red),
        ),
        child: Text('Register'),
      ),
    ],
  );
}
}

```

//Login Search

```

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'home_screen.dart';
import 'register_screen.dart';

```

```

class LoginScreen extends StatelessWidget {
  TextEditingController emailController = TextEditingController();
  TextEditingController passwordController = TextEditingController();

  void login(BuildContext context) async {
    String email = emailController.text.trim();
    String password = passwordController.text.trim();

    if (email == "" || password == "") {
      print("Fill in all the fields");
    } else {
      try {
        UserCredential userCredential = await FirebaseAuth.instance
          .signInWithEmailAndPassword(email: email, password: password);
        if (userCredential.user != null) {
          Navigator.pushReplacement(
            context,
            CupertinoPageRoute(builder: (context) => HomeScreen()),
          );
        }
      } catch (e) {
        print("Error during login: $e");
      }
    }
  }
}

```

```
}
```

```
@override
```

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

```
  return Scaffold(
```

```
    appBar: AppBar(
```

```
      title: const Text('Login'),
```

```
      centerTitle: true,
```

```
    ),
```

```
    body: SingleChildScrollView(
```

```
      child: Container(
```

```
        color: Colors.white,
```

```
        padding: const EdgeInsets.all(16.0),
```

```
        child: Column(
```

```
          mainAxisAlignment: MainAxisAlignment.center,
```

```
          children: [
```

```
            Image.asset('assets/ytlogo2.png', height: 300, width: 300),
```

```
            SizedBox(height: 20),
```

```
            TextField(
```

```
              controller: emailController,
```

```
              decoration: InputDecoration(labelText: 'Email'),
```

```
            ),
```

```
            SizedBox(height: 10),
```

```
            TextField(
```

```
              controller: passwordController,
```

```
              decoration: InputDecoration(labelText: 'Password'),
```

```
              obscureText: true,
```

```
            ),
```

```
            SizedBox(height: 20),
```

```
            ElevatedButton(
```

```
              onPressed: () {
```

```
                login(context);
```

```
              },
```

```
              style: ElevatedButton.styleFrom(
```

```
                primary: Colors.red,
```

```
            ),
```

```
            child: Text('Login'),
```

```
          ),
```

```
            SizedBox(height: 10),
```

```
            Row(
```

```
              mainAxisAlignment: MainAxisAlignment.center,
```

```
              children: [
```

```
                Text("New?"),
```

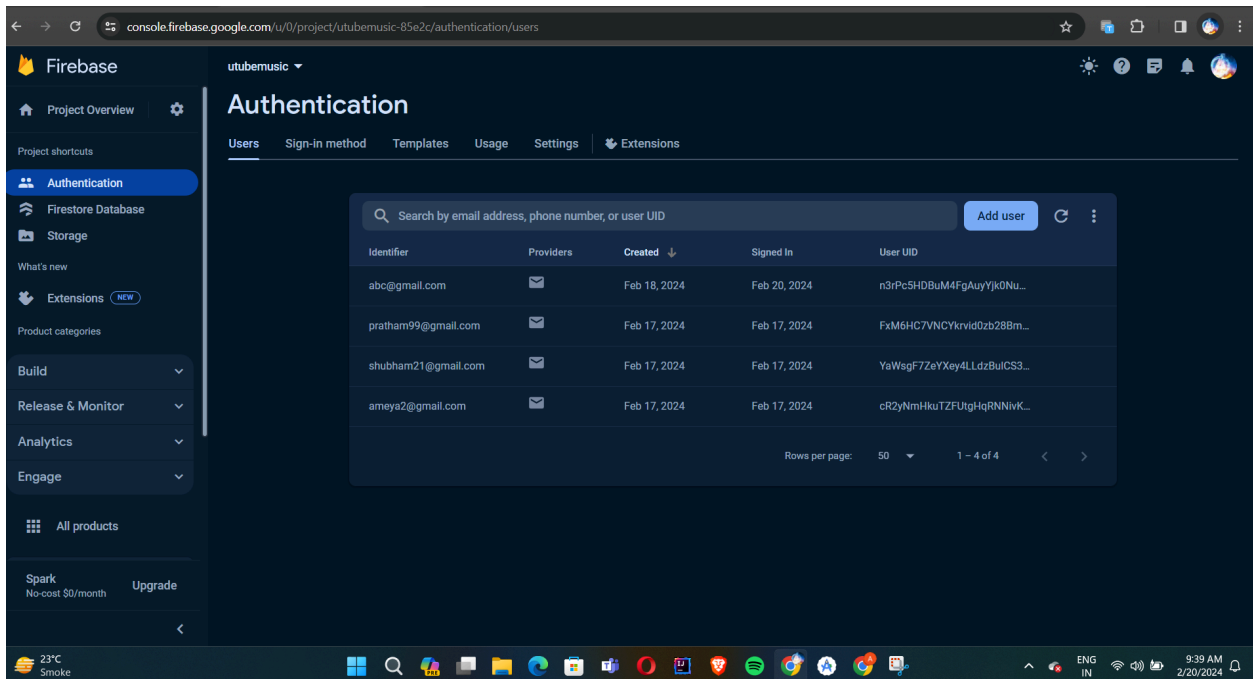
```
                TextButton(
```

```

onPressed: () {
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) => RegisterScreen(),
    ),
  );
},
child: Text('Register'),
),
],
),
],
),
),
),
);
}
}

```

FIREBASE CONSOLE:



The screenshot shows the Firebase Authentication console for the project 'utubemusic'. The 'Users' tab is selected, displaying a table of registered users. The table has columns for Identifier, Providers, Created, Signed In, and User UID. There are four users listed, all created on February 17, 2024, and signed in on February 20, 2024. The left sidebar shows the project overview and various Firebase services. The bottom of the screen shows a Windows taskbar with the date and time as 9:39 AM on 2/20/2024.

Identifier	Providers	Created	Signed In	User UID
abc@gmail.com	📧	Feb 18, 2024	Feb 20, 2024	n3rPcSHDBuM4FgAuyYjk0Nu...
pratham99@gmail.com	📧	Feb 17, 2024	Feb 17, 2024	FxM6HC7VNCYkrvid0zb28Bm...
shubham21@gmail.com	📧	Feb 17, 2024	Feb 17, 2024	YaWsgF7ZeYXey4LLdzBulCS3...
ameya2@gmail.com	📧	Feb 17, 2024	Feb 17, 2024	cR2yNmtHkuTZFUtHgRNNivK...

OUTPUT:

9:33



DEBUG



Register



Email

abc@gmail.com

Password

123456

Confirm Password

Register

9:35



DEBUG

Login



Email

abc@gmail.com

Password

Login

Register

CONCLUSION:

In conclusion, integrating Firebase with a Flutter app for email/password authentication and database connectivity provides a robust and scalable backend solution. The combination of `firebase_auth` and `cloud_firestore` packages streamlines user authentication and real-time data management. Leveraging Firebase services enhances the overall functionality and user experience of your Flutter application, allowing seamless integration of authentication features and cloud-based storage.