

Prachit P. Paralikar
D15A 43

MPL EXPT 6

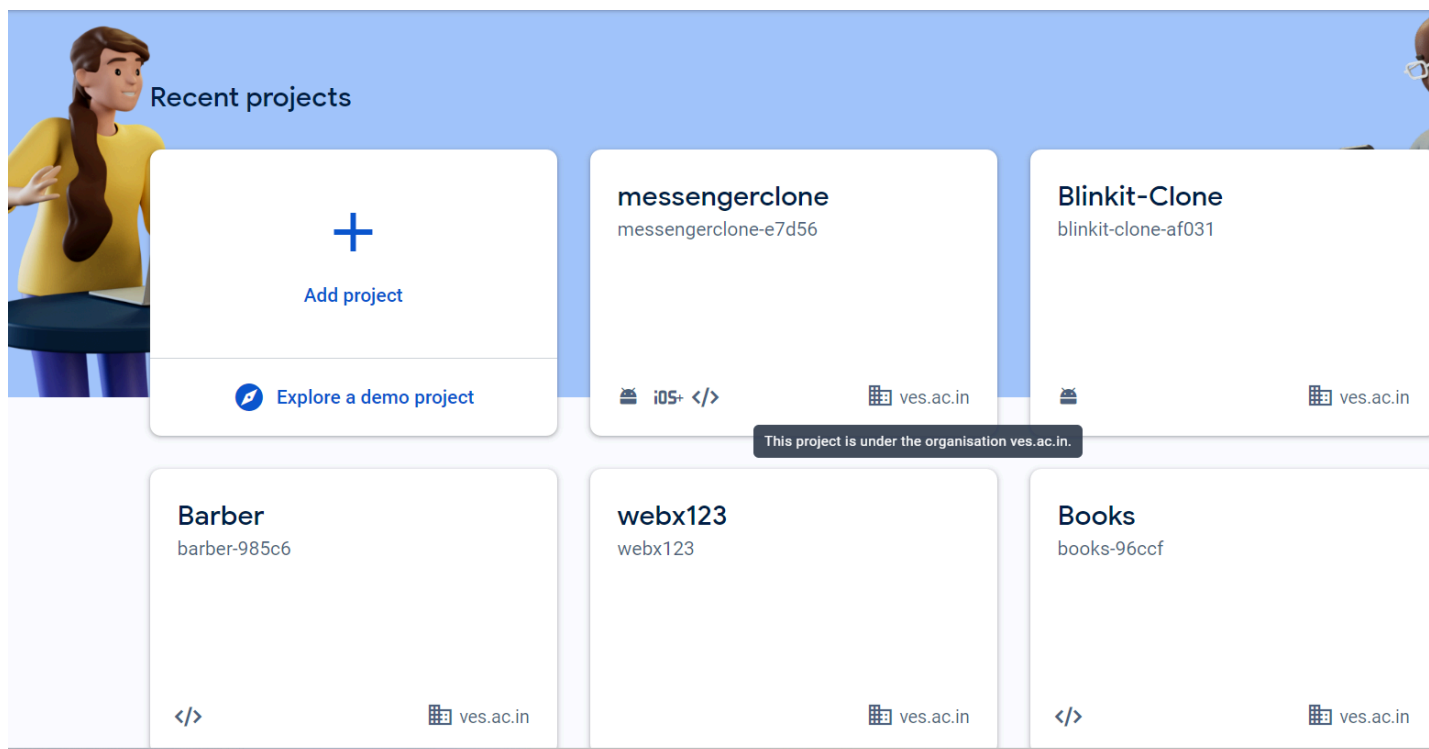
AIM:

How To Set Up Firebase with Flutter for iOS and Android Apps

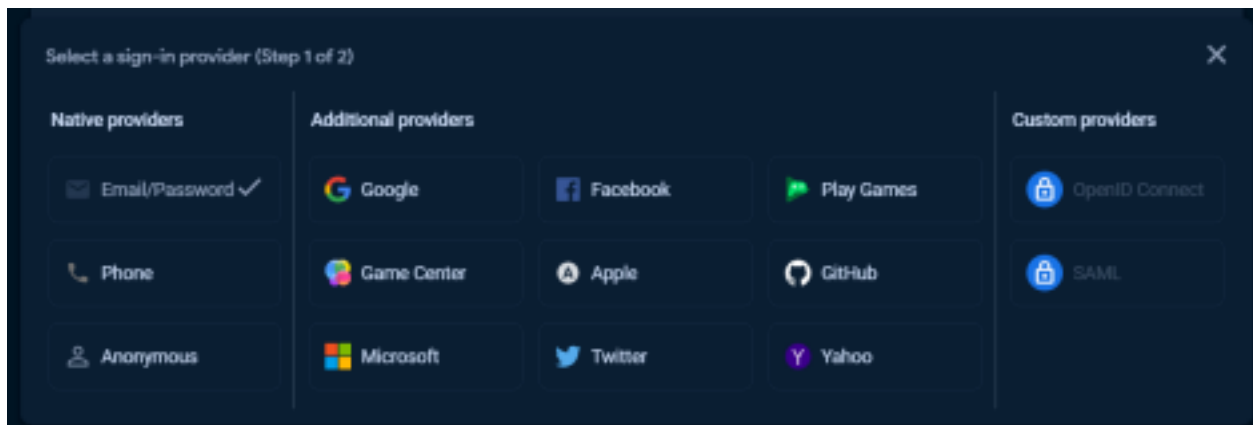
THEORY / STEPS:

(Wherever testing mode is available, I have opted for testing mode.)

1.Create a project in firebase console.



2.Go to authentication section inside the project we have created and enable email/password verification under sign-in method.



3. Similarly enable firestore database.

(Further part is done using Flutter CLI i.e Flutter command line interface)
Following are the commands for the same

4.Installing firebase CLI

npm install -g firebase-tools

5. We have to connect our project with the google account we have made out firebase project on.

firebase login

It would open a browser where we can choose our account.

6. Activation of flutter fire cli

Dart pub global activate flutterfire_cli

7. Configuration of flutterfire

Flutterfire configure

This shows us our firebase projects and we can select our project that we created for this application. I.e app1

8. It would then show us the platforms that we want to make our application compatible for.

9. Configured applications will be now available in firebase.

10. We check the flutter initialization in main function of our application. `future: Firebase.initializeApp(),`

11. We add dependencies:

```
cupertino_icons: ^1.0.2
  firebase_core: ^2.25.4
  firebase_auth: ^4.17.4
  cloud_firestore: 4.15.5
  provider:
```

These should be compatible with the flutter SDK version we are using.

If we get an error with any of these we can try running:

Flutter pub upgrade

Flutter pub get

CODE:

Auth

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_sign_in/google_sign_in.dart';
import 'package:messenger_clone/helperfunctions/sharedpref_helper.dart';
```

```

import 'package:messenger_clone/services/database.dart';
import 'package:messenger_clone/views/home.dart';
import 'package:shared_preferences/shared_preferences.dart';

class AuthMethods {
  final FirebaseAuth auth = FirebaseAuth.instance;

  getCurrentUser() async {
    return await auth.currentUser;
  }

  Future<void> signInWithGoogle(BuildContext context) async {
    final FirebaseAuth _firebaseAuth = FirebaseAuth.instance;
    final GoogleSignIn _googleSignIn = GoogleSignIn();

    try {
      final GoogleSignInAccount? googleSignInAccount =
        await _googleSignIn.signIn();

      if (googleSignInAccount != null) {
        final GoogleSignInAuthentication googleSignInAuthentication =
          await googleSignInAccount.authentication;

        final AuthCredential credential = GoogleAuthProvider.credential(
          idToken: googleSignInAuthentication.idToken,
          accessToken: googleSignInAuthentication.accessToken,
        );

        try {
          final UserCredential result =
            await _firebaseAuth.signInWithCredential(credential);
          final User? userDetails = result.user;

          if (userDetails != null) {
            SharedPreferenceHelper().saveUserEmail(userDetails.email ??
            "");

            SharedPreferenceHelper().saveUserId(userDetails.uid ?? "");
            SharedPreferenceHelper().saveUserName(

```

```

        userDetails?.email?.replaceAll("@gmail.com", "") ?? "");
    SharedPreferencesHelper()
        .saveDisplayName(userDetails.displayName ?? "");
    SharedPreferencesHelper()
        .saveUserProfileUrl(userDetails.photoURL ?? "");

    Map<String, dynamic> userInfoMap = {
        "email": userDetails.email,
        "username": userDetails.email!.replaceAll("@gmail.com", ""),
        "name": userDetails.displayName,
        "imgUrl": userDetails.photoURL
    };

    await DatabaseMethods().addUserInfoToDB(
        userDetails.uid!, userInfoMap); // Fixed this line
    Navigator.pushReplacement(
        context, MaterialPageRoute(builder: (context) => Home()));
} else {
    print('User details are null');
}
} catch (e) {
    print('Error signing in with credential: $e');
}
} else {
    print('Google sign-in account is null');
}
} catch (e) {
    print('Error signing in with Google: $e');
}
}

Future<void> signOut(BuildContext context) async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    prefs.clear();
    await auth.signOut();

    // Navigate to the sign-in screen or perform any other required action
}
}

```

OUTPUT:
Authentication:

messengerclone

?

P

Authentication

Users

Sign-in method

Templates

Usage

Settings

Extensions

Search by email address, phone number or user UID

Add user

Identifier	Providers	Created	Signed in	User UID
prachitparalikar07@gm...		20 Feb 2024	21 Feb 2024	G9ZQSa61ejfXY5r7L8kEi2jKU...
2021.prachit.paralikar@...		20 Feb 2024	20 Feb 2024	yzMfyBszypNXwiYvQ5NioS5T...

Rows per page

50

1 – 2 of 2

<

>

Chats in firestore database:

Home

>

chatrooms

>

2021.prachit.par.

More in Google Cloud

(default)

chatrooms

+ Start collection

chatrooms

stories

users

chatrooms

+ Add document

2021.prachit.paralika...

2021.prachit.paralikar@ves.ac.in_prachitparalikar07

chats

+ Add field

lastMessage: "kkkkk"

lastMessageSendBy: "prachitparalikar07"

lastMessageSendTs: 22 February 2024 at 09:54:59 UTC+5:30

users

0

"prachitparalikar07"

1

"2021.prachit.paralikar@ves.ac.in"

With this connection we are able to achieve the following:

- **Saving email and password in database.**
- **Saving chats and their timestamps in the database under current user's uid.**
- **Fetching chats from the database to display on the profile page.**
- **Updating the values in the database for password in the authentication database.**

CONCLUSION:

Connected firebase with twitter clone project and handled related errors.