MAD and PWA Lab Mini Project

Name: Shubham Nakashe Class:D15A Roll no: 38

Aim: To connect Flutter UI with Firebase Database

Theory:

Step 1: Create a Firebase Project

- 1. Go to the [Firebase Console](https://console.firebase.google.com/) and create a new project.
- 2. Follow the on-screen instructions to set up your project.

Step 2: Add Firebase to Your Flutter Project

1. Add the `firebase_core,firebase_auth,firebase_storage,cloud_storage` package to your `pubspec.yaml` file:

```
yaml
dependencies:
flutter:
sdk: flutter
firebase_auth: ^4.14.1
firebase_core: ^2.23.0
firebase_storage: ^11.5.2
```

2. Run 'flutter pub get' to install the package.

Step 3: Initialize Firebase in Your Flutter App

1. Import the `firebase_core` package in your Dart code:

```
dart import 'package:firebase core/firebase core.dart';
```

2. Initialize Firebase in your app's `main` function:

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

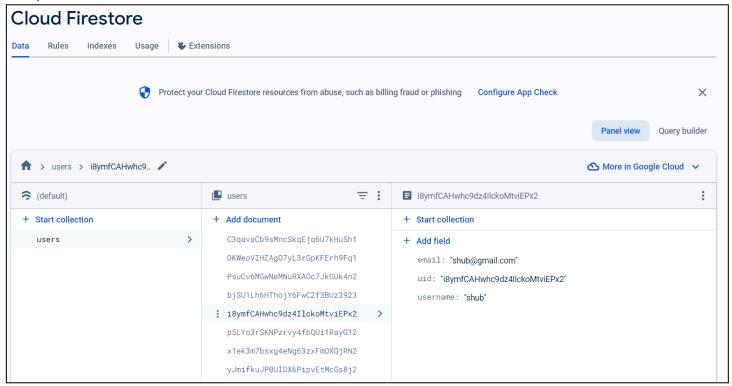
```
CODE:
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:twitch_clone/firebase_options.dart';
import 'package:twitch_clone/providers/user_provider.dart';
import 'package:twitch_clone/resources/auth_methods.dart';
import 'package:twitch_clone/screens/feed_screen.dart';
import 'package:twitch_clone/screens/home_screen.dart';
import 'package:twitch_clone/screens/login_screen.dart';
import 'package:twitch_clone/screens/onboarding_screen.dart';
import 'package:twitch_clone/screens/profile.dart';
import 'package:twitch_clone/screens/signup_screen.dart';
import 'package:twitch_clone/utils/colors.dart';
import 'package:twitch_clone/models/user.dart' as model;
import 'package:twitch_clone/widgets/loading_indicator.dart';
Future<void> main() async {
 WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
   options: DefaultFirebaseOptions.currentPlatform,
  );
 runApp(MultiProvider(providers: [
  ChangeNotifierProvider(
   create: (_) => UserProvider(),
  ),
 ], child: const MyApp()));
class MyApp extends StatelessWidget {
 const MyApp({super.key});
 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   debugShowCheckedModeBanner: false,
   title: 'Twitch Clone',
   theme: ThemeData.light().copyWith(
     scaffoldBackgroundColor: backgroundColor,
     appBarTheme: AppBarTheme.of(context).copyWith(
        backgroundColor: backgroundColor,
```

```
elevation: 0,
     titleTextStyle: const TextStyle(
       color: primaryColor,
       fontSize: 20,
       fontWeight: FontWeight.w600,
     iconTheme: const IconThemeData(
       color: primaryColor,
     ))),
 routes: {
  OnBoardingScreen.routeName: (context) => const OnBoardingScreen(),
  LoginScreen.routeName: (context) => const LoginScreen(),
  SignupScreen.routeName: (context) => const SignupScreen(),
  HomeScreen.routeName: (context) => const HomeScreen(),
  ProfileScreen.routeName: (context) => ProfileScreen(),
 },
 home: FutureBuilder(
   future: AuthMethods()
      .getCurrentUser(FirebaseAuth.instance.currentUser != null
        ? FirebaseAuth.instance.currentUser!.uid
        : null)
     .then((value) {
    if (value != null) {
     Provider.of<UserProvider>(context, listen: false)
        .setUser(model.User.fromMap(value));
    }
    return value;
   }),
   builder: (context, snapshot) {
    if (snapshot.connectionState == ConnectionState.waiting) {
     return const LoadingIndicator();
    }
    if (snapshot.hasData) {
     return const HomeScreen();
    return const OnBoardingScreen();
   }),
);
```

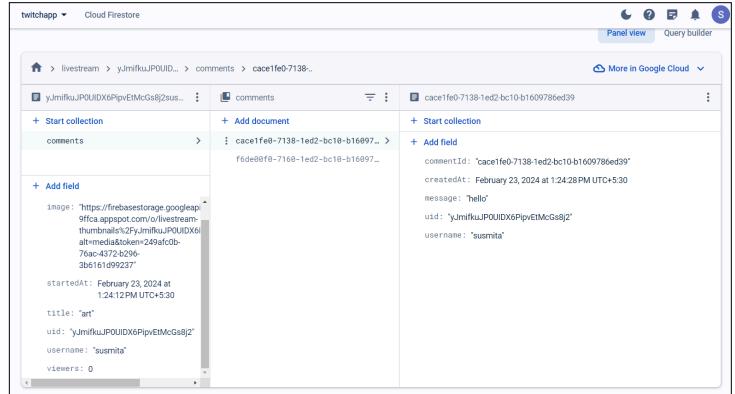
}

OUTPUT:

1) <u>USER DATA IS STORED IN FIREBASE</u>



2) LiveStream detail



Comment by users

3) Thumbnail of live stream

