

## **MPL EXPERIMENT NO:- 6**

**Name:-Sanket More**

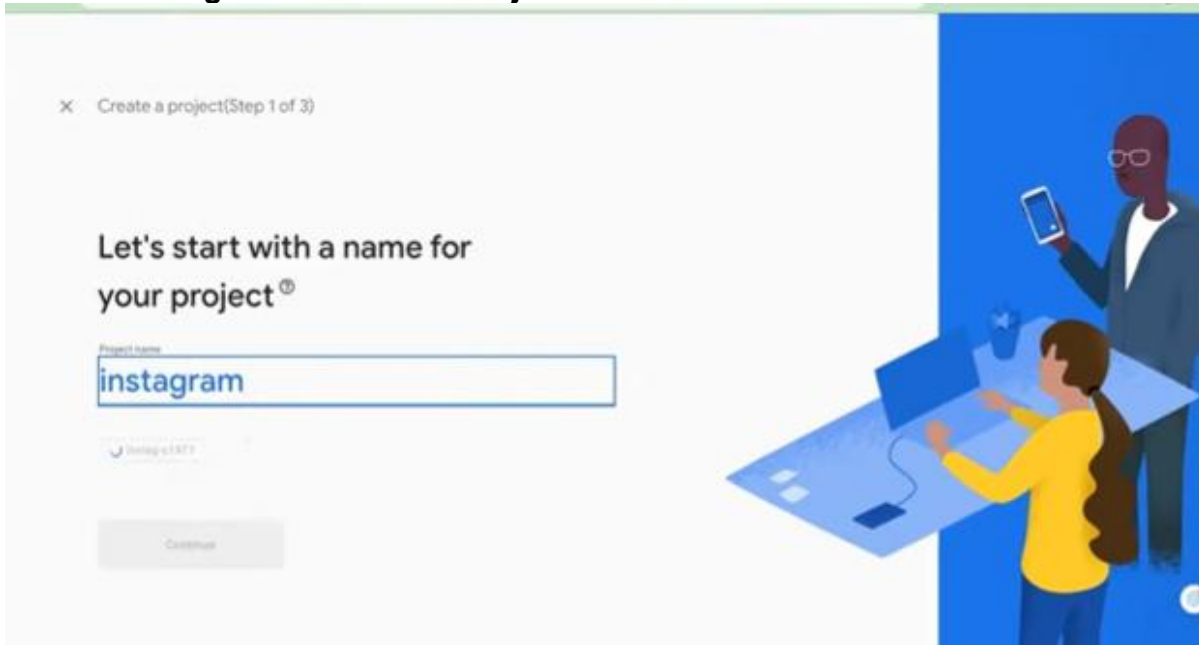
**D15A**

**Roll-no:-29**

Aim:- To Connect flutter UI with firebase database

---

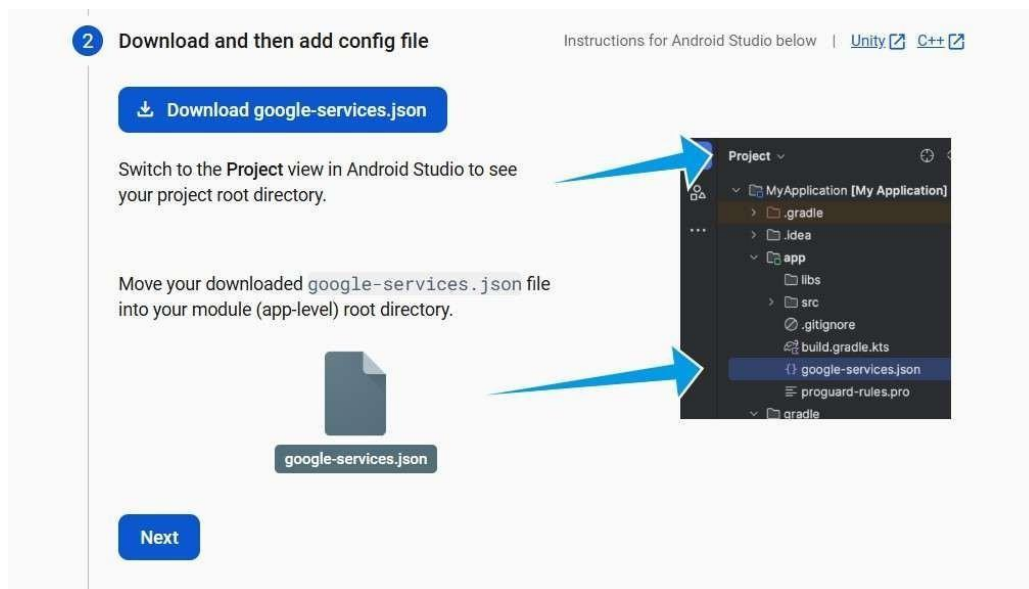
### **Creating a New Firebase Project**



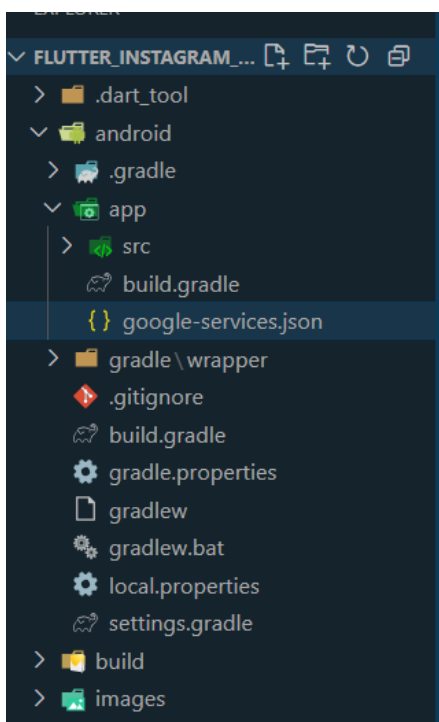
First, log in with your Google account to manage your Firebase projects. From within the Firebase dashboard, select the Create new project button and give it a name

In order to add Android support to our Flutter application, select the Android logo from the dashboard. This brings us to the following screen:

Then download the `google-services.json` file, that you will get.



put that file in the android folder (root level)



then select the `build.gradle.kts` (Kotlin DSL) part, and then follow the rest instructions

### 3 Add Firebase SDK

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

★ Are you still using the `buildscript` syntax to manage plugins? [Learn how to add Firebase plugins](#) using that syntax.

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

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

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

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

```
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.kts` file, add both the `google-services` plugin and any Firebase SDKs that you want to use in your app:

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

```
plugins {  
    id("com.android.application")  
    // Add the Google services Gradle plugin  
    id("com.google.gms.google-services")  
    ...  
}  
  
dependencies {  
    // Import the Firebase BoM  
    implementation(platform("com.google.firebase:firebase-bom:33.9.0"))  
  
    // TODO: Add the dependencies for Firebase products you want to use  
    // When using the BoM, don't specify versions in Firebase dependencies  
    // https://firebase.google.com/docs/android/setup#available-libraries  
}
```

By using the Firebase Android BoM, your app will always use compatible Firebase library versions. [Learn more](#)

### 4 Next steps

You're all set!

Make sure to check out the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

You can also explore [sample Firebase apps](#).

Or, continue to the console to explore Firebase.

[Previous](#)

[Continue to console](#)

Generate the `firebase_options.dart` file, based on the `google-services.json` file

```

class DefaultFirebaseOptions {

  static const FirebaseOptions web = FirebaseOptions(
    apiKey: 'AIzaSyDvhua8saPw1WH4VAdePLF7AoR7RwhhZmA',
    appId: '1:167400934834:web:e3046e038154729c1e68ab',
    messagingSenderId: '167400934834',
    projectId: 'instagram-7555e',
    authDomain: 'instagram-7555e.firebaseio.com',
    storageBucket: 'instagram-7555e.appspot.com',
  );

  static const FirebaseOptions android = FirebaseOptions(
    apiKey: 'AIzaSyC01Z6y2-6nrBl7HXyxFbqH2_292jEyo4g',
    appId: '1:167400934834:android:9f1c6eb6431a5d481e68ab',
    messagingSenderId: '167400934834',
    projectId: 'instagram-7555e',
    storageBucket: 'instagram-7555e.appspot.com',
  );

  static const FirebaseOptions ios = FirebaseOptions(
    apiKey: 'AIzaSyBlPS99WXA_v15pkf10GHkT2t2BKqptnp4',
    appId: '1:167400934834:ios:b9d046db16efc3a91e68ab',
    messagingSenderId: '167400934834',
    projectId: 'instagram-7555e',
    storageBucket: 'instagram-7555e.appspot.com',
    iosBundleId: 'com.example.flutterInstagramClone',
  );
}

```


In your part select the sign-in method and enable it.

inshortsclone ▾

## Authentication

Users **Sign-in method** Templates Usage Settings Extensions


Sign-in providers

 Email/Password
 

Enable

Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery, and email address change primitives. [Learn more](#)

Enable

 Email link (passwordless sign-in)
 

Enable

Cancel

Save

## Code:- Authenction\_logic

```
import 'dart:io';

import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter_instagram_clone/data/firebase_service/firestor.dart';
import 'package:flutter_instagram_clone/data/firebase_service/storage.dart';

import 'package:flutter_instagram_clone/util/exeption.dart';

class Authentication {

  final FirebaseAuth _auth = FirebaseAuth.instance;

  Future<void> Login({
    required String email,
    required String password,
  }) async {
    try {
      await _auth.signInWithEmailAndPassword(
        email: email.trim(), password: password.trim());
    } on FirebaseException catch (e) {
      throw exceptions(e.message.toString());
    }
  }

  Future<void> Signup({
    required String email,
    required String password,
    required String passwordConfirme,
    required String username,
    required String bio,
    required File profile,
  }) async {
    String URL;
```

```

try {
  if (email.isNotEmpty &&
      password.isNotEmpty &&
      username.isNotEmpty &&
      bio.isNotEmpty) {
    if (password == passwordConfirme) {
      // create user with email and password
      await _auth.createUserWithEmailAndPassword(
        email: email.trim(),
        password: password.trim(),
      );
      // upload profile image on storage

      if (profile != File("")) {
        URL =
          await StorageMethod().uploadImageToStorage('Profile', profile);
      } else {
        URL = "";
      }

      // get information with firestore

      await Firebase_Firestor().CreateUser(
        email: email,
        username: username,
        bio: bio,
        profile: URL == ""
          ? 'https://firebasestorage.googleapis.com/v0/b/instagram-
8a227.appspot.com/o/person.png?alt=media&token=c6fcbe9d-f502-4aa1-8b4b-ec37339e78ab'
          : URL,
      );
    } else {
      throw exceptions('password and confirm password should be same');
    }
  }
}

```

```

    }
  } else {
    throw exceptions('enter all the fields');
  }
} on FirebaseException catch (e) {
  throw exceptions(e.message.toString());
}
}
}

```

## Login\_screen

```

import 'dart:io';

import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter_instagram_clone/data/firebase_service/firestor.dart';
import 'package:flutter_instagram_clone/data/firebase_service/storage.dart';

import 'package:flutter_instagram_clone/util/exeption.dart';

class Authentication {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  Future<void> Login({
    required String email,
    required String password,
  }) async {
    try {
      await _auth.signInWithEmailAndPassword(
        email: email.trim(), password: password.trim());
    } on FirebaseException catch (e) {
      throw exceptions(e.message.toString());
    }
  }

  Future<void> Signup({
    required String email,
    required String password,
    required String passwordConfirme,
    required String username,
    required String bio,
  })

```



```

    required File profile,
  }) async {
    String URL;
    try {
      if (email.isNotEmpty &&
          password.isNotEmpty &&
          username.isNotEmpty &&
          bio.isNotEmpty) {
        if (password == passwordConfirme) {
          // create user with email and password
          await _auth.createUserWithEmailAndPassword(
            email: email.trim(),
            password: password.trim(),
          );
          // upload profile image on storage

          if (profile != File("")) {
            URL =
              await StorageMethod().uploadImageToStorage('Profile', profile);
          } else {
            URL = "";
          }

          // get information with firestor

          await Firebase_Firestor().CreateUser(
            email: email,
            username: username,
            bio: bio,
            profile: URL == ""
              ? 'https://firebasestorage.googleapis.com/v0/b/instagram-
8a227.appspot.com/o/person.png?alt=media&token=c6fcbe9d-f502-4aa1-8b4b-ec37339e78ab'
              : URL,
          );
        } else {
          throw exceptions('password and confirm password should be same');
        }
      } else {
        throw exceptions('enter all the fields');
      }
    } on FirebaseException catch (e) {
      throw exceptions(e.message.toString());
    }
  }
}

```

## Home\_Screen

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter_instagram_clone/widgets/post_widget.dart';
import 'package:flutter_screenutil/flutter_screenutil.dart';

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

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  FirebaseAuth _auth = FirebaseAuth.instance;
  FirebaseFirestore _firestore = FirebaseFirestore.instance;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      resizeToAvoidBottomInset: false,
      backgroundColor: Colors.white,
      appBar: AppBar(
        centerTitle: true,
        elevation: 0,
        title: SizedBox(
          width: 105.w,
          height: 28.h,
          child: Image.asset('images/instagram.jpg'),
        ),
        leading: Image.asset('images/camera.jpg'),
        actions: [
          const Icon(
            Icons.favorite_border_outlined,
            color: Colors.black,
            size: 25,
          ),
          Image.asset('images/send.jpg'),
        ],
        backgroundColor: const Color(0xffFAFAFA),
      ),
      body: CustomScrollView(
        slivers: [
          StreamBuilder(
            stream: _firestore
              .collection('posts')
              .orderBy('time', descending: true)
              .snapshots(),
            builder: (context, snapshot) {
              return SliverList(
                delegate: SliverChildBuilderDelegate(
                  (context, index) {
                    if (!snapshot.hasData) {
                      return Center(child: CircularProgressIndicator());
                    }
                    return PostWidget(snapshot.data!.docs[index].data());
                  },
                ),
              );
            },
          ),
        ],
      ),
    );
  }
}
```

```
        childCount:
          snapshot.data == null ? 0 : snapshot.data!.docs.length,
      ),
    );
  },
)
],
),
);
}
}
```

OUTPUT :



12:44

Instagram



Email

username

bio

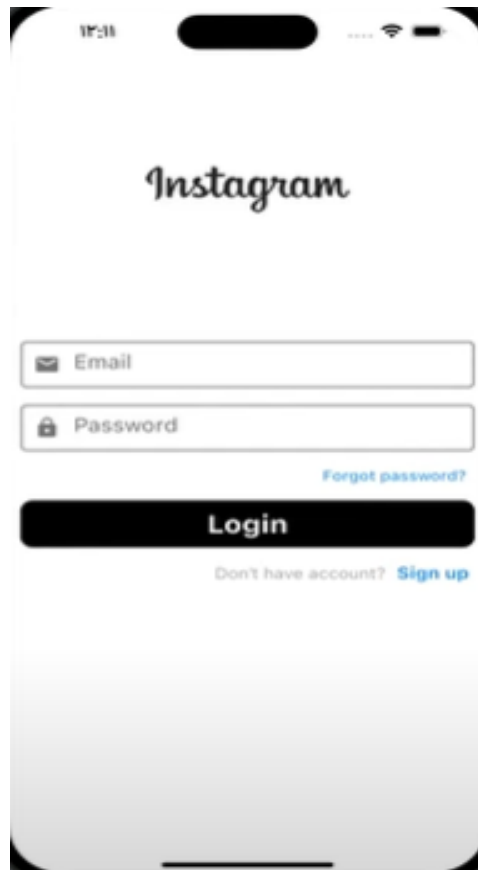
Password

PasswordConfirm

**Sign up**

Don't have account? [Login](#)

This is a screenshot of the Instagram sign-up screen. It features a white background with the Instagram logo at the top. Below the logo is a circular profile picture placeholder. The form consists of five input fields: Email, username, bio, Password, and PasswordConfirm. A black button labeled 'Sign up' is positioned below the fields. At the bottom, there is a link that says 'Don't have account? Login'.



12:44

Instagram

Email

Password

[Forgot password?](#)

**Login**

Don't have account? [Sign up](#)

This is a screenshot of the Instagram login screen. It features a white background with the Instagram logo at the top. Below the logo are two input fields: Email and Password. A link labeled 'Forgot password?' is located to the right of the Password field. A black button labeled 'Login' is positioned below the fields. At the bottom, there is a link that says 'Don't have account? Sign up'.