Experiment 06

Aim : To Connect Flutter UI with fireBase database

Theory :

Firebase is a comprehensive mobile and web application development platform provided by Google. It offers a wide range of features and services that enable developers to build high-quality apps quickly and efficiently.

At its core, Firebase provides tools for various aspects of app development, including authentication, real-time database, cloud storage, hosting, and more. Here's a breakdown of some key Firebase components:

Authentication: Firebase Authentication allows developers to easily integrate secure authentication methods into their apps. It supports various authentication methods, including email/password, phone number, Google, Facebook, Twitter, and more. This feature handles user management, authentication, and security, making it simple to add user sign-up, sign-in, and access control to apps.

Realtime Database: Firebase Realtime Database is a NoSQL cloud database that enables developers to store and sync data in real time. It's particularly useful for applications that require real-time updates, such as chat apps, collaboration tools, and gaming apps. The database is JSON-based, which makes it flexible and easy to use. It automatically synchronizes data across all connected clients and devices, ensuring that changes made by one user are immediately reflected on others' screens.

Cloud Firestore: Cloud Firestore is Firebase's newer, more scalable NoSQL database solution. It offers more powerful querying capabilities, better scalability, and improved performance compared to the Realtime Database. Firestore organizes data into collections and documents, allowing for more structured and efficient data storage. It also supports real-time updates, offline data access, and powerful querying capabilities.

Cloud Storage: Firebase Cloud Storage provides secure and reliable cloud storage for user-generated content, such as images, videos, and documents. It allows developers to easily upload and download files from their apps using simple APIs. Cloud Storage integrates seamlessly with other Firebase services, making it easy to store and serve user-generated content in Firebase-powered apps.

Hosting: Firebase Hosting allows developers to quickly and securely deploy web apps and static content to a global content delivery network (CDN). It provides fast and reliable hosting with SSL encryption, custom domain support, and automatic scaling. With Firebase Hosting, developers can deploy web apps with just a few simple commands, eliminating the need for complex server configurations and maintenance.

Cloud Functions: Firebase Cloud Functions allow developers to run backend code in response to events triggered by Firebase features and HTTPS requests. It enables developers to extend the functionality of their apps without managing servers or infrastructure. Cloud Functions are written in JavaScript or TypeScript and can be deployed with a single command using the Firebase CLI (Command Line Interface).

Analytics: Firebase Analytics provides insights into app usage and user behavior, helping developers understand how users interact with their apps. It tracks key metrics such as active users, retention, engagement, and conversion, allowing developers to make informed decisions about app improvements and optimizations.

Performance Monitoring: Firebase Performance Monitoring helps developers identify and fix performance issues in their apps. It provides detailed insights into app performance, including app startup time, network latency, and UI responsiveness. Performance Monitoring helps developers optimize their apps for better user experiences and higher ratings.

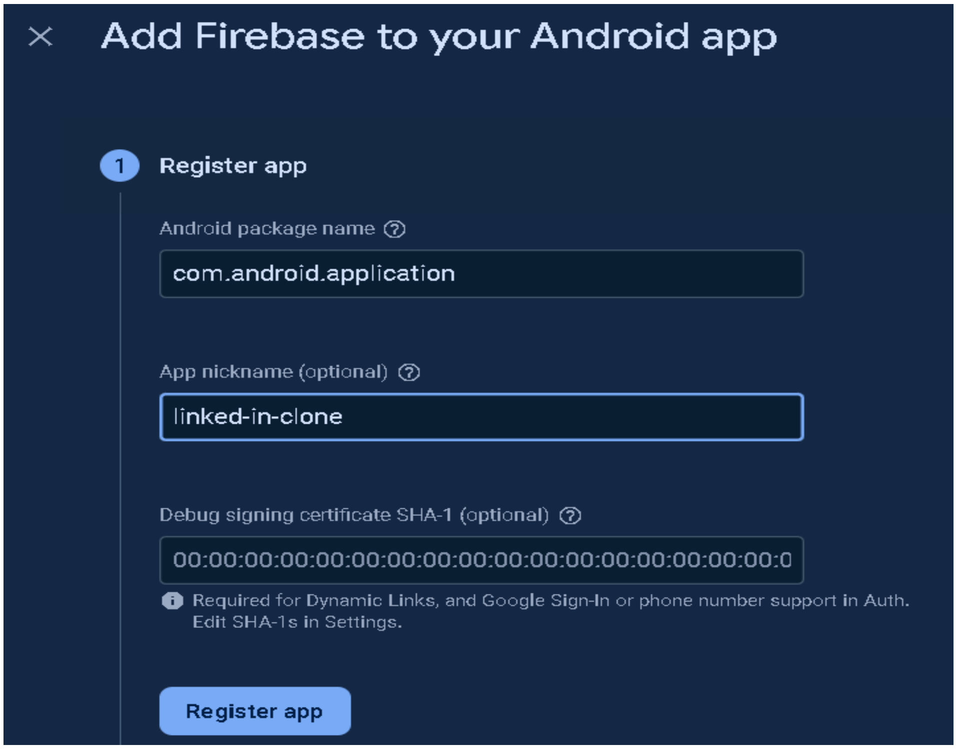
In addition to these core features, Firebase offers many other services, including Cloud Messaging for push notifications, Remote Config for dynamic app configuration, Machine Learning for integrating machine learning models into apps, and more. Overall, Firebase provides a powerful and comprehensive platform for building, managing, and scaling mobile and web applications.

Code:

```
import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:cloud_firestore/cloud_firestore.dart'; // Import Firestore
import 'package:linkedin_mobile_ui/theme/styles.dart';
```

```
class CreatePage extends StatefulWidget {
  final VoidCallback? onCloneClickListener;
  const CreatePage({Key? key, required this.onCloneClickListener})
      : super(key: key);

  @override
  State<CreatePage> createState() => _CreatePageState();
}

class _CreatePageState extends State<CreatePage> {

  final TextEditingController _postBodyController = TextEditingController();

  bool _openTwoBottomModalSheetsOnce = false;

  final FocusScopeNode _subPostBottomModalSheetFocusNode = FocusScopeNode();
  final FocusScopeNode _superPostBottomModalSheetFocusNode = FocusScopeNode();

  @override
  void dispose() {
    _postBodyController.dispose();
    super.dispose();
  }

  // Function to add post to Firestore
  Future<void> _addPostToFirestore(String postBody) async {
    try {
      await FirebaseFirestore.instance.collection('posts').add({
        'body': postBody,
        'timestamp': DateTime.now(),
      });
      print('Post added to Firestore');
    } catch (e) {
      print('Error adding post to Firestore: $e');
    }
  }

  @override
  Widget build(BuildContext context) {
    if(_openTwoBottomModalSheetsOnce == false) {
      WidgetsBinding.instance!.addPostFrameCallback((timeStamp) {
        _createSuperPostBottomModalSheet();
        _createSubPostBottomModalSheet();
        print("value before = $_openTwoBottomModalSheetsOnce");
        setState(() {
          _openTwoBottomModalSheetsOnce = true;
        });
        print("value after = $_openTwoBottomModalSheetsOnce");
      });
    }

    return const Scaffold();
  }

  _createSuperPostBottomModalSheet() {
    showModalBottomSheet(
      isScrollControlled: true,
      enableDrag: false,
      isDismissible: false,
      context: context,
      builder: (context) {
        return StatefulBuilder(
            builder: (context, void Function(void Function()) setState) {
              return FocusScope(
                node: _superPostBottomModalSheetFocusNode,
                child: Container(
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      Container(
                        padding: const EdgeInsets.symmetric(horizontal: 10),
                        width: double.infinity,
                        height: 110,
                        decoration:
                        const BoxDecoration(color: linkedInWhiteFFFFFF, boxShadow: [
                          BoxShadow(
                              offset: Offset(0, 2),
                              color: linkedInLightGreyCACCCE,
                              blurRadius: 5,
                              spreadRadius: 0.1),
                        ]),
                        child: Padding(
                          padding: const EdgeInsets.only(bottom: 15.0),
                          child: Row(
                            crossAxisAlignment: CrossAxisAlignment.end,
                            mainAxisAlignment: MainAxisAlignment.spaceBetween,
                            children: [
                              Row(
                  children: [
                   GestureDetector(
                     onTap: widget.onCloneClickListener,
                     child: const Icon(
                      Icons.close_outlined,
                      size: 30,
                     )),
                   const SizedBox(
                    width: 15,
                   ),
                   const Text(
                    "Share Post",
                    style: TextStyle(
                      fontSize: 25,
                      fontWeight: FontWeight.bold,
                      color: linkedInMediumGrey86888A),
                   )
                  ],
                 ),
                 Text(
                  "Post",
                  style: TextStyle(
                    fontSize: 22,
                    fontWeight: FontWeight.bold,
                    color: _postBodyController.text.isEmpty? linkedInLightGreyCACCCE :
  linkedInBlue0077B5),
                 )
                ],
               ),
              ),
             ),
             const SizedBox(
              height: 30,
             ),
             Expanded(
              child: Container(
               margin: const EdgeInsets.symmetric(horizontal: 20),
               child: Column(
                children: [
                 Row(
                  children: [
                   Container(
                    width: 60,
                    height: 60,
        child: ClipRRect(
          borderRadius: BorderRadius.circular(30),
          child: Image.asset("assets/profile_1.jpeg"),
        ),
      ),
      const SizedBox(
        width: 10,
      ),
      Column(
        mainAxisAlignment: MainAxisAlignment.end,
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          _switchWidget(
              title: "Nidhi Gawde",
              prefixIcon: Icons.account_circle_rounded,
              suffixIcon: Icons.arrow_drop_down_outlined),
          const SizedBox(
            height: 5,
          ),
          _switchWidget(
              title: "Anyone",
              prefixIcon: FontAwesomeIcons.earth,
              suffixIcon: Icons.arrow_drop_down_outlined),
        ],
      ),
    ],
  ),
  const SizedBox(
    height: 20,
  ),
  TextFormField(
    controller: _postBodyController,
    onTap: () {
      _postBodyController.addListener(() {
        if(_postBodyController.text.length == 1) {
          setState(() {});
          print("call first IF setState");
        } else if (_postBodyController.text.length < 1) {
          setState(() {});
          print("2nd ELSE IF setState");
        }

        print("onTap called");
      });
```

```
          },
          style: const TextStyle(fontSize: 22),
          maxLines: 15,
          decoration: const InputDecoration(
              hintText: "What do you want to talk about?",
              border: InputBorder.none),
        ),
      ],
    ),
  ),
),
Container(
  margin: const EdgeInsets.symmetric(horizontal: 30),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      Row(
        children: [
          const Icon(
            Icons.camera_alt,
            color: linkedInMediumGrey86888A,
          ),
          const SizedBox(
            width: 15,
          ),
          const Icon(
            Icons.video_call,
            color: linkedInMediumGrey86888A,
          ),
          const SizedBox(
            width: 15,
          ),
          const Icon(
            Icons.image,
            color: linkedInMediumGrey86888A,
          ),
          const SizedBox(
            width: 25,
          ),
          GestureDetector(
              onTap: () {
                _createSubPostBottomModalSheet();
              },
              child: const Icon(
                                    Icons.more_horiz,
                                    color: linkedInMediumGrey86888A,
                                  )),
                            ],
                          ),
                          const Row(
                            children: [
                              Icon(
                                Icons.message_outlined,
                                color: linkedInMediumGrey86888A,
                              ),
                              SizedBox(
                                width: 10,
                              ),
                              Text(
                                "Anyone",
                                style: TextStyle(
                                    fontWeight: FontWeight.bold,
                                    color: linkedInMediumGrey86888A),
                              )
                            ],
                          )
                        ],
                      ),
                    ),


                  ],
                ),
              ),
            );
          }
      );
    },
  ).then((value) {
    _superPostBottomModalSheetFocusNode.unfocus();
  });
}

_switchWidget({String? title, IconData? prefixIcon, IconData? suffixIcon}) {
  return Container(
    height: 30,
    padding: const EdgeInsets.symmetric(horizontal: 10),
    decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(20),
        border: Border.all(width: 1, color: linkedInMediumGrey86888A)),
    child: Row(
      children: [
        Icon(
          prefixIcon,
          color: linkedInMediumGrey86888A,
          size: 18,
        ),
        const SizedBox(
          width: 5,
        ),
        Text(
          "$title",
          style: const TextStyle(fontWeight: FontWeight.bold, fontSize: 12),
        ),
        const SizedBox(
          width: 5,
        ),
        Icon(
          suffixIcon,
          size: 30,
        ),
      ],
    ),
  );
}

_createSubPostBottomModalSheet() {
  showModalBottomSheet(
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(20),
    ),
    barrierColor: Colors.transparent,
    context: context,
    builder: (context) {
      return FocusScope(
        node: _subPostBottomModalSheetFocusNode,
        child: Container(
          decoration: BoxDecoration(color: linkedInWhiteFFFFFF, boxShadow: [
            BoxShadow(
                offset: const Offset(5, 0),
                blurRadius: 1,
                color: linkedInLightGreyCACCCE.withOpacity(.6),
                  spreadRadius: 0.5)
            ]),
            child: Padding(
              padding: const EdgeInsets.symmetric(horizontal: 20.0, vertical: 20),
              child: SingleChildScrollView(
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Center(
                      child: Container(
                        width: 80,
                        height: 6,
                        decoration: BoxDecoration(
                            borderRadius: BorderRadius.circular(10),
                            color: linkedInMediumGrey86888A),
                      ),
                    ),
                    const SizedBox(height: 20,),
                    _createSubPostNavigationItem(title: "Add a photo", iconData: Icons.image),
                    const SizedBox(height: 25,),
                    _createSubPostNavigationItem(title: "Take a video", iconData: Icons.video_call),
                    const SizedBox(height: 25,),
                    _createSubPostNavigationItem(title: "Use a template", iconData:
Icons.temple_buddhist),
                    const SizedBox(height: 25,),
                    _createSubPostNavigationItem(title: "Celebrate an occasion", iconData:
Icons.celebration),
                    const SizedBox(height: 25,),
                    _createSubPostNavigationItem(title: "Add a document", iconData:
Icons.document_scanner),
                    const SizedBox(height: 25,),
                    _createSubPostNavigationItem(title: "Share that you're hiring", iconData:
Icons.work),
                    const SizedBox(height: 25,),
                    _createSubPostNavigationItem(title: "Find an expert", iconData:
Icons.account_circle_rounded),
                    const SizedBox(height: 25,),
                    _createSubPostNavigationItem(title: "Create a poll", iconData: Icons.bar_chart),
                    const SizedBox(height: 25,),
                    _createSubPostNavigationItem(title: "Create an event", iconData: Icons.event),
                    const SizedBox(height: 25,),
                  ],
                ),
              ),
            ),
          ),
        );
      },
    ).then((value) {
      _subPostBottomModalSheetFocusNode.unfocus();
    });
  }

  _createSubPostNavigationItem({IconData? iconData, String? title}) {
    return Row(
      children: [
        Icon(iconData, size: 25,color: linkedInMediumGrey86888A,),
        const SizedBox(width: 10,),
        Text("$title", style: const TextStyle(fontSize: 16, color: linkedInMediumGrey86888A,
fontWeight: FontWeight.bold),)
      ],
    );
  }
}
```



Add Firebase to your Android app

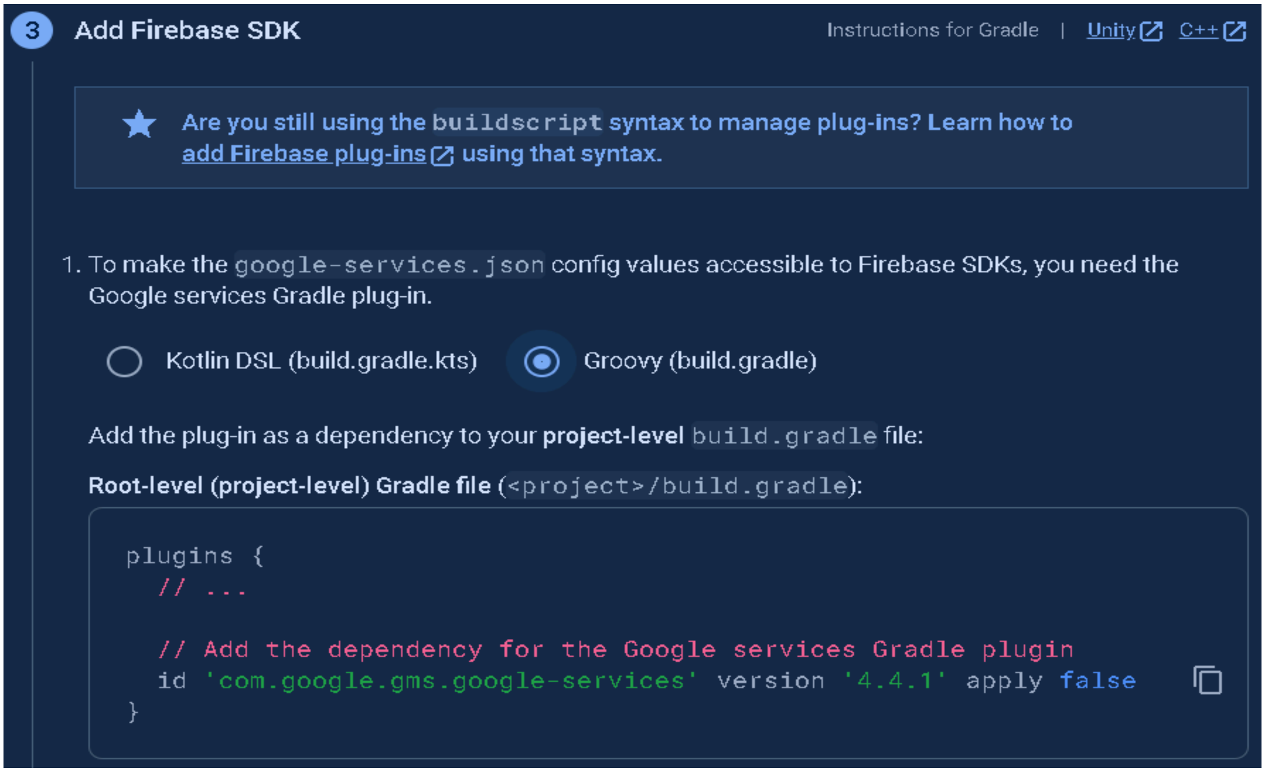
1 Register app

Android package name: com.android.application

App nickname (optional): linked-in-clone

Debug signing certificate SHA-1 (optional): 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0

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



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.

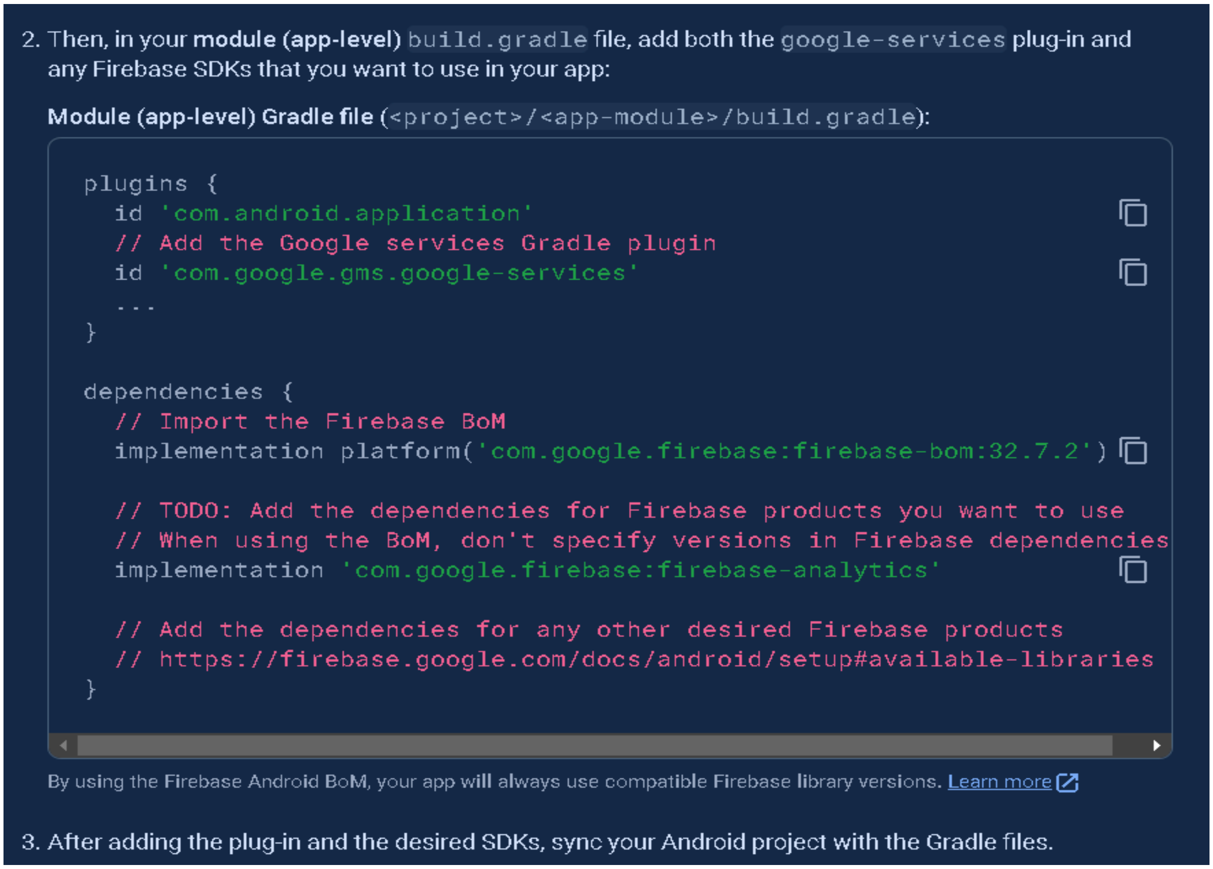
1. To make the `google-services.json` config values accessible to Firebase SDKs, you need the Google services Gradle plug-in.

   ○ 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 '4.4.1' apply false
}
```



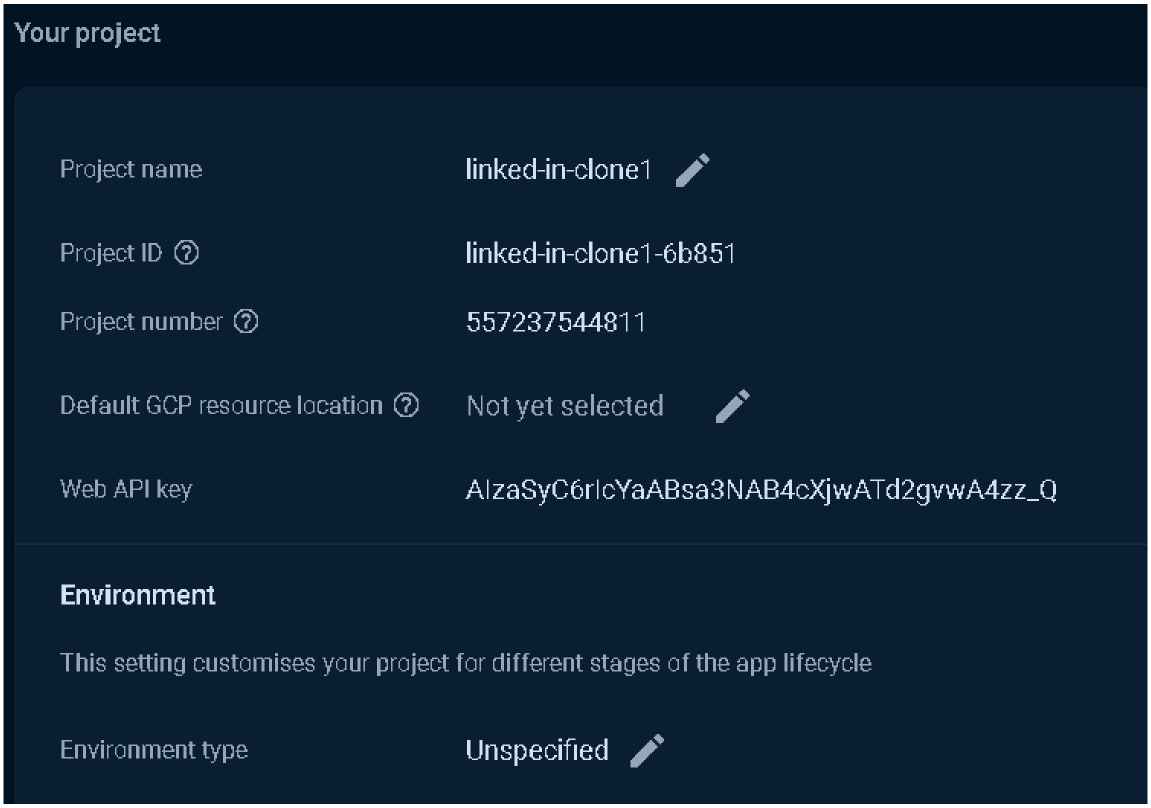
2. Then, in your **module (app-level)** `build.gradle` file, add both the `google-services` plug-in and any Firebase SDKs that you want to use in your app:

   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 {
  // Import the Firebase BoM
  implementation platform('com.google.firebase:firebase-bom:32.7.2')

  // TODO: Add the dependencies for Firebase products you want to use
  // When using the BoM, don't specify versions in Firebase dependencies
  implementation 'com.google.firebase:firebase-analytics'

  // Add the dependencies for any other desired Firebase products
  // 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

3. After adding the plug-in and the desired SDKs, sync your Android project with the Gradle files.



Your project

Project name: linked-in-clone1

Project ID: linked-in-clone1-6b851

Project number: 557237544811

Default GCP resource location: Not yet selected

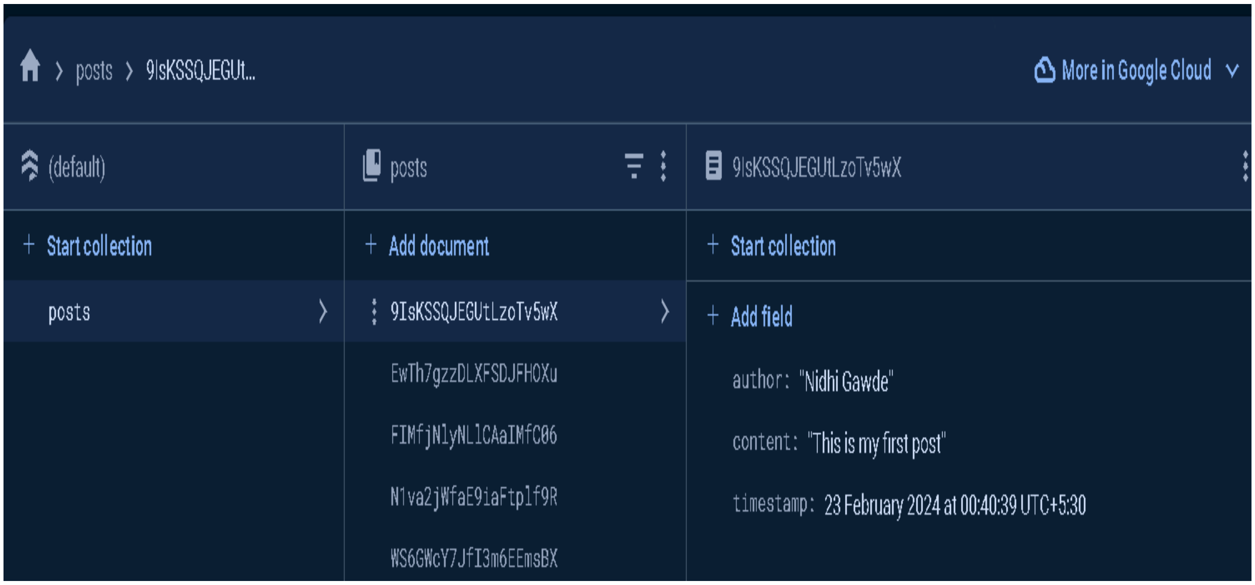
Web API key: AIzaSyC6rIcYaABsa3NAB4cXjwATd2gvwA4zz_Q

Environment

This setting customises your project for different stages of the app lifecycle

Environment type: Unspecified

Firestore:



posts

9IsKSSQJEGUtLzoTv5wX

EwTh7gzzDLXFSDJFHOXu

FIMfjNlyNLlCAaIMfC06

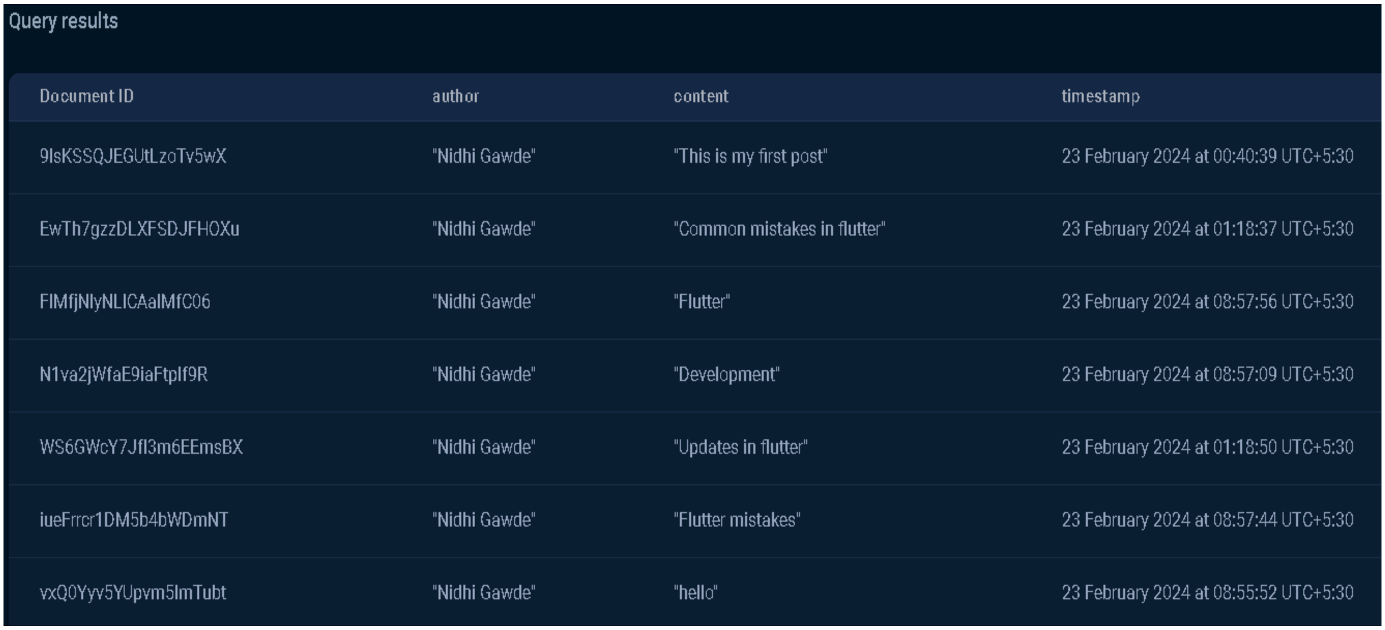
N1va2jWfaE9iaFtplf9R

WS6GWcY7JfI3m6EEmsBX

author: "Nidhi Gawde"

content: "This is my first post"

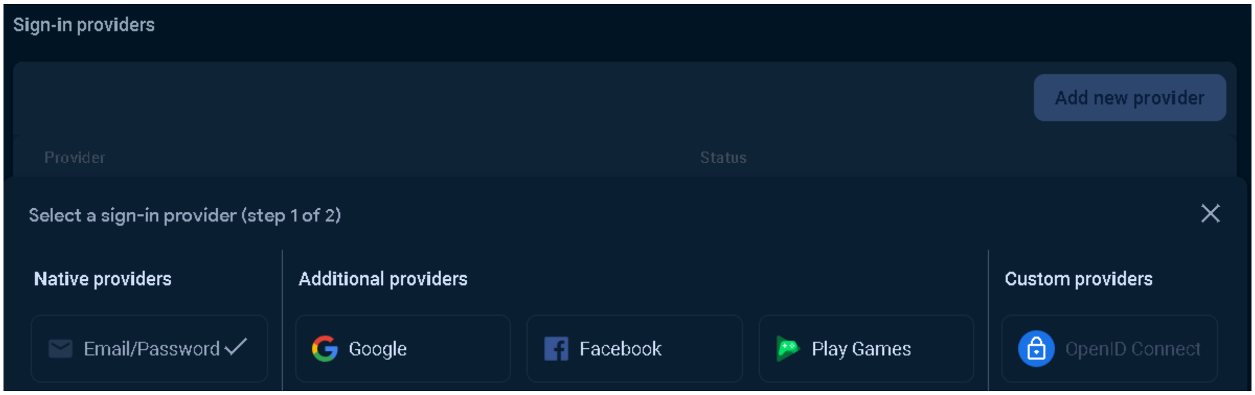
timestamp: 23 February 2024 at 00:40:39 UTC+5:30



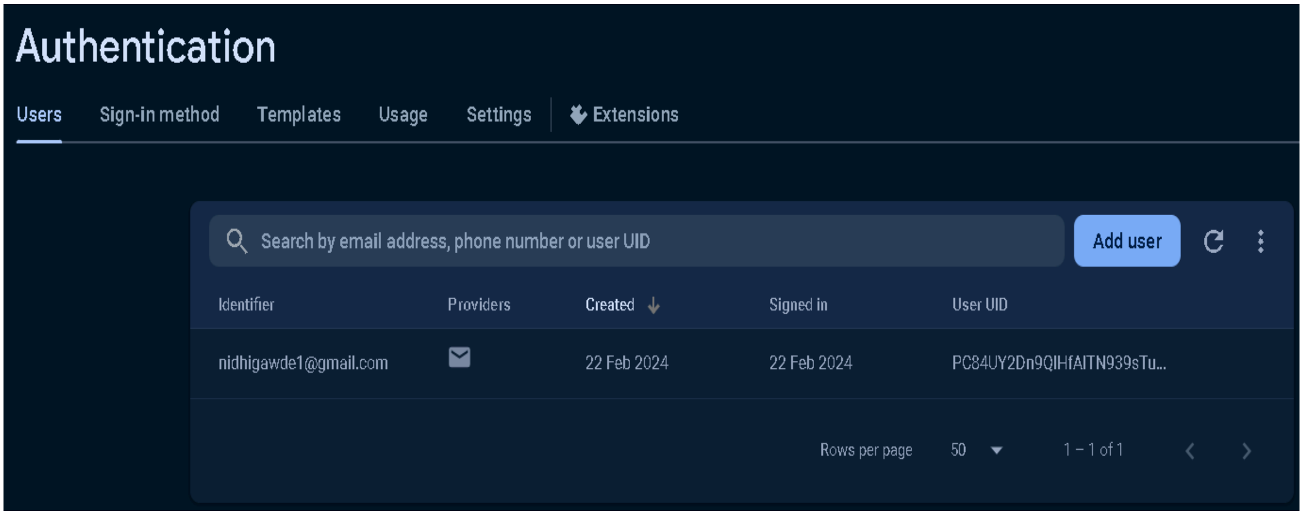
Query results

Document ID	author	content	timestamp
9IsKSSQJEGUtLzoTv5wX	"Nidhi Gawde"	"This is my first post"	23 February 2024 at 00:40:39 UTC+5:30
EwTh7gzzDLXFSDJFHOXu	"Nidhi Gawde"	"Common mistakes in flutter"	23 February 2024 at 01:18:37 UTC+5:30
FIMfjNlyNLICAalMfC06	"Nidhi Gawde"	"Flutter"	23 February 2024 at 08:57:56 UTC+5:30
N1va2jWfaE9iaFtplf9R	"Nidhi Gawde"	"Development"	23 February 2024 at 08:57:09 UTC+5:30
WS6GWcY7Jfl3m6EEmsBX	"Nidhi Gawde"	"Updates in flutter"	23 February 2024 at 01:18:50 UTC+5:30
iueFrrcr1DM5b4bWDmNT	"Nidhi Gawde"	"Flutter mistakes"	23 February 2024 at 08:57:44 UTC+5:30
vxQ0Yyv5YUpvm5lmTubt	"Nidhi Gawde"	"hello"	23 February 2024 at 08:55:52 UTC+5:30

Authentication:



Select a sign-in provider (step 1 of 2)



Identifier	Providers	Created ↓	Signed in	User UID
nidhigawde1@gmail.com	✉	22 Feb 2024	22 Feb 2024	PC84UY2Dn9QlHfAITN939sTu...