EXPERIMENT 6

AIM: To connect Flutter UI with firebase

THEORY:

Firebase is a comprehensive backend-as-a-service (BaaS) platform developed by Google. It provides a wide range of tools and services such as authentication, real-time databases, cloud storage, and analytics to help developers build and scale modern mobile and web applications.

Integrating Firebase with Flutter allows developers to enhance their apps with powerful backend capabilities while focusing on building rich user interfaces using Flutter.

---

1. Why Use Firebase with Flutter?

- Real-time Functionality: Firebase offers real-time database and cloud-based data synchronization, making it ideal for chat apps, live feeds, and collaborative apps.
- User Authentication: Secure user sign-in and sign-up using email/password, phone number, or social providers (Google, Facebook, etc.).
- Data Storage: Store structured data in Firestore or files (like images and videos) in Firebase Cloud Storage.
- Notifications: Send push notifications using Firebase Cloud Messaging (FCM).
- Analytics & Crash Reporting: Monitor user behavior and app performance in real-time.

---

2. Common Firebase Services Used in Flutter Apps

- Firebase Authentication: For managing user login and identity verification.
- Cloud Firestore: A scalable NoSQL database used to store and sync data in real time.

- Firebase Realtime Database: A JSON-based database with real-time syncing.
- Firebase Storage: For uploading and retrieving media files like images and videos.
- Firebase Cloud Messaging: Enables sending push notifications to user devices.

---

3. Benefits of Connecting Flutter UI with Firebase

- Cross-Platform Support: Firebase works seamlessly with Android, iOS, and web, which fits perfectly with Flutter's cross-platform nature.
- Scalability: Easily scales from a small prototype to a large production app.
- Security: Firebase provides built-in security rules and user authentication.
- Time-Saving: Reduces the need to build a backend from scratch, speeding up development.

Conclusion

Connecting Flutter UI with Firebase combines the power of a beautiful front-end framework with a robust and feature-rich backend platform. This integration allows developers to build modern, responsive, and scalable apps quickly and efficiently, while still offering advanced features like real-time updates, user management, and secure data handling.

CODE:

post_list.dart:

```
import 'package:flutter/material.dart';
import 'post_card.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
```

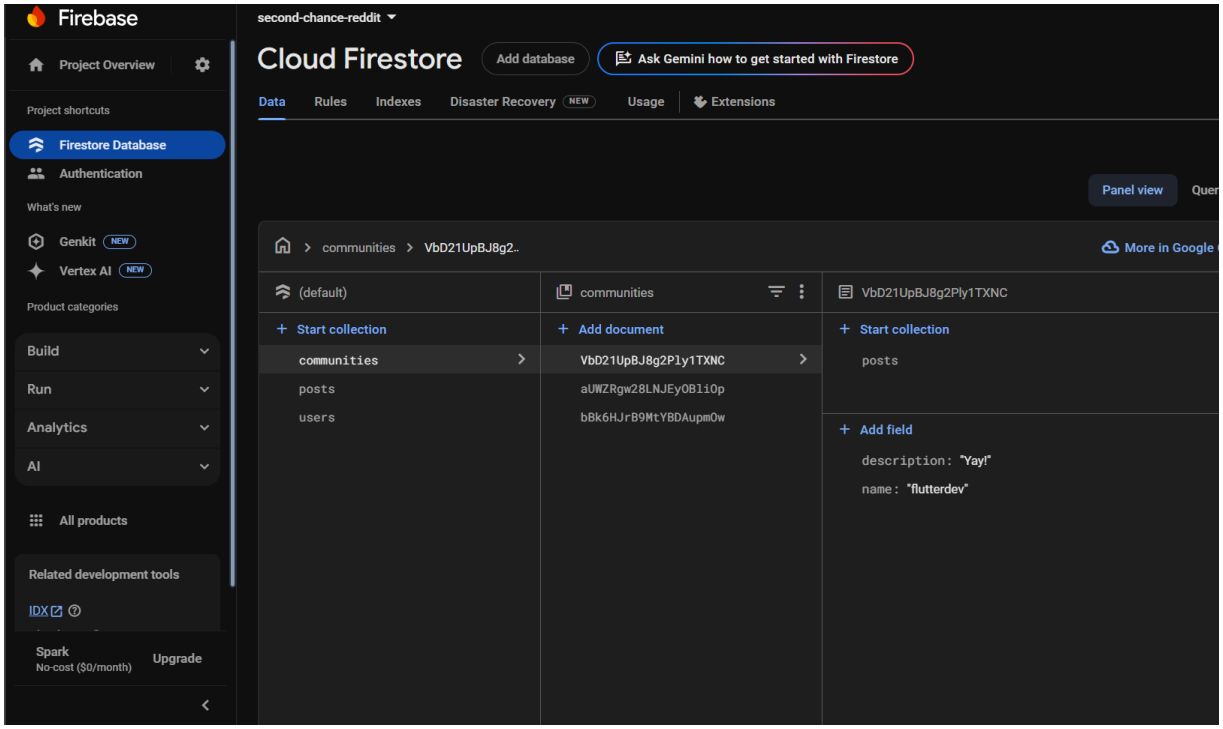
```
class PostList extends
StatelessWidget {
  final List<String>
followedCommunities;

  const PostList({super.key,
required this.followedCommunities});

  @override
  Widget build(BuildContext context)
{
    return
followedCommunities.isEmpty
        ? const Center(
            child: Text(
              "Follow communities to
see posts!",
              style:
TextStyle(color: Colors.white),
            ),
          )
        : StreamBuilder(
            stream:
FirebaseFirestore.instance

.collection('communities')
                .where('name',
whereIn: followedCommunities)
                .snapshots(),
            builder: (context,
snapshot) {
              if (!snapshot.hasData
|| snapshot.data!.docs.isEmpty) {
                return const
Center(child:
CircularProgressIndicator());
              }

              final communities =
snapshot.data!.docs;

              return FutureBuilder(
                future: Future.wait(

communities.map((communityDoc) async
{
                    final
postsSnapshot =
                        await
communityDoc.reference.collection('p
osts').get();

                    return
postsSnapshot.docs.map((postDoc) {
                      return {
                        'post':
postDoc,

'communityId': communityDoc.id,
                      };
                    }).toList();
                  }),
                ),
                builder: (context,
futureSnapshot) {
                  if
(!futureSnapshot.hasData) {
                    return const
Center(child:
CircularProgressIndicator());
                  }

                  final allPosts =

futureSnapshot.data!.expand((posts)
=> posts).toList();

                  if
(allPosts.isEmpty) {
                    return const
Center(
                      child: Text(
                        "No posts
found in your communities!",
```

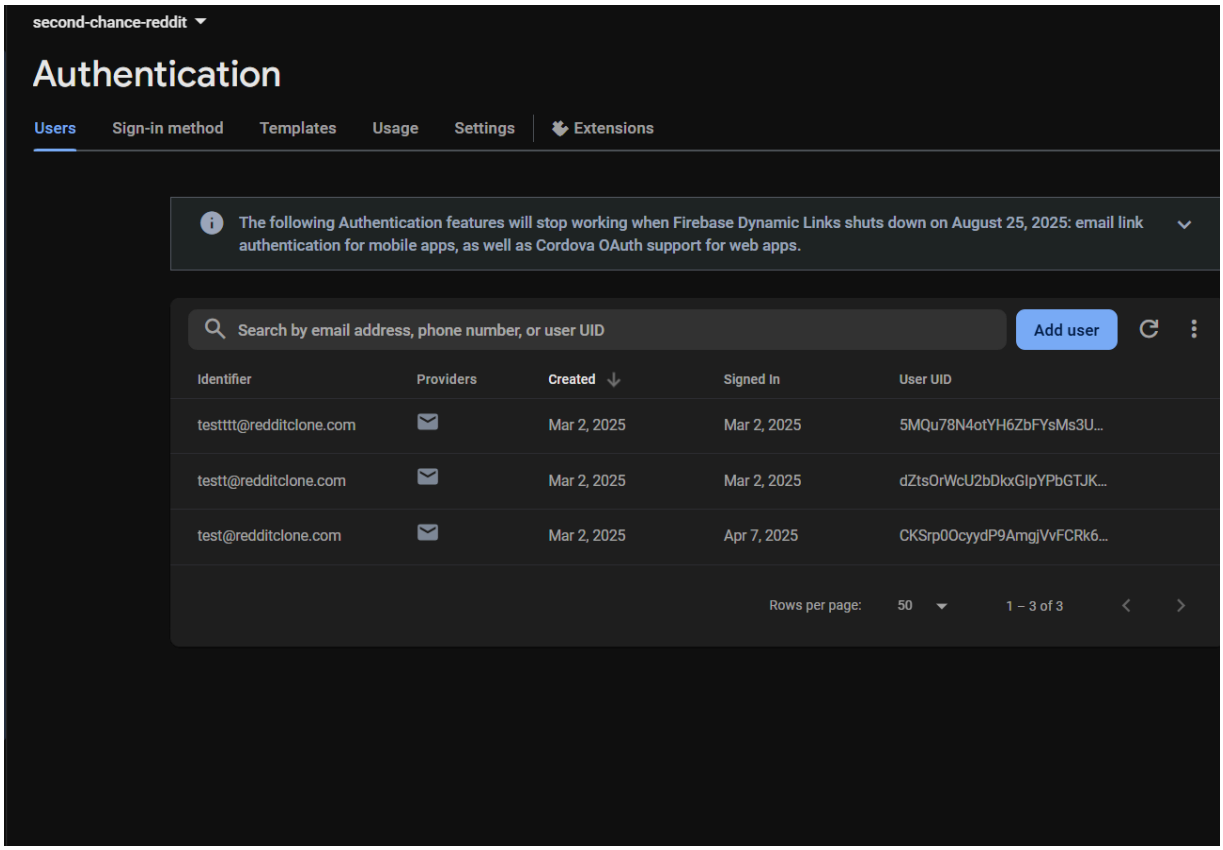
```
                      style:
TextStyle(color: Colors.white),
                    ),
                  );
                }

                return
ListView.builder(
                  itemCount:
allPosts.length,
                  itemBuilder:
(context, index) {
                    final postData
= allPosts[index];
                    final post =
postData['post'] as
DocumentSnapshot?;
                    final
communityId =
postData['communityId'] as String?;

                    if (post ==
null || communityId == null) {
                      return const
SizedBox.shrink(); // Skip invalid
posts
                    }

                    return
PostCard(
                      postId:
post.id ?? '',
                      subredditId:
communityId,
                      title:
post['title'] ?? 'Untitled Post',
                      upvotes:
(post['upvotes'] ?? 0) as int,
                      downvotes:
(post['downvotes'] ?? 0) as int,
                    );
                  },
                          );
                        },
                      );
                    },
                  );
                }
              }
```

```
build.gradle:
plugins {
    id "com.android.application"
    id "kotlin-android"
    id
'com.google.gms.google-services'  //
Firebase Plugin
    id
"dev.flutter.flutter-gradle-plugin"
// Keep this last
}

android {
    namespace =
"com.android.reddit_clone"
    compileSdkVersion 34
    ndkVersion = flutter.ndkVersion

    compileOptions {
        sourceCompatibility =
JavaVersion.VERSION_1_8
        targetCompatibility =
JavaVersion.VERSION_1_8
    }

    kotlinOptions {
        jvmTarget =
JavaVersion.VERSION_1_8
    }

    defaultConfig {
        // TODO: Specify your own
unique Application ID (htt
```

```
ps://developer.android.com/studio/bu
ild/application-id.html).
        applicationId =
"com.android.reddit_clone"
        // You can update the
following values to match your
application needs.
        // For more information,
see:
https://flutter.dev/to/review-gradle
-config.
        minSdkVersion 23
        targetSdkVersion 33
        multiDexEnabled true

        versionCode =
flutter.versionCode
        versionName =
flutter.versionName
    }

    buildTypes {
        release {
            // TODO: Add your own
signing config for the release
build.
            // Signing with the
debug keys for now, so `flutter run
--release` works.
            signingConfig =
signingConfigs.debug
        }
    }
}

flutter {
    source = "../.."
}

dependencies {
    // Import Firebase BoM (Bill of
Materials)
    implementation
platform('com.google.firebase:fireba
se-bom:33.9.0')

    // Add the Firebase SDKs you
need
    implementation
'com.google.firebase:firebase-auth'
    implementation
'com.google.firebase:firebase-firest
ore'
    implementation
'com.google.firebase:firebase-storag
e'
}
```

OUTPUT:


Firebase
second-chance-reddit
Project Overview
Project shortcuts
Firestore Database
Authentication
What's new
Genkit NEW
Vertex AI NEW
Product categories
Build
Run
Analytics
AI
All products
Related development tools
IDX
Spark
No-cost ($0/month)
Upgrade
Cloud Firestore
Add database
Ask Gemini how to get started with Firestore
Data
Rules
Indexes
Disaster Recovery NEW
Usage
Extensions
Panel view
communities > VbD21UpBJ8g2..
More in Google
(default)
communities
VbD21UpBJ8g2Ply1TXNC
Start collection
communities
posts
users
Add document
VbD21UpBJ8g2Ply1TXNC
aUWZRgw28LNJEyOBliOp
bBk6HJrB9MtYBDAupmOw
Start collection
posts
Add field
description: "Yay!"
name: "flutterdev"


second-chance-reddit
Authentication
Users
Sign-in method
Templates
Usage
Settings
Extensions
The following Authentication features will stop working when Firebase Dynamic Links shuts down on August 25, 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.
Search by email address, phone number, or user UID
Add user
Identifier
Providers
Created
Signed In
User UID
testttt@redditclone.com
Mar 2, 2025
Mar 2, 2025
5MQu78N4otYH6ZbFYsMs3U...
testt@redditclone.com
Mar 2, 2025
Mar 2, 2025
dZtsOrWcU2bDkxGlpYPbGTJK...
test@redditclone.com
Mar 2, 2025
Apr 7, 2025
CKSrp0OcyydP9AmgjVvFCRk6...
Rows per page:
50
1 – 3 of 3