

Experiment : 06

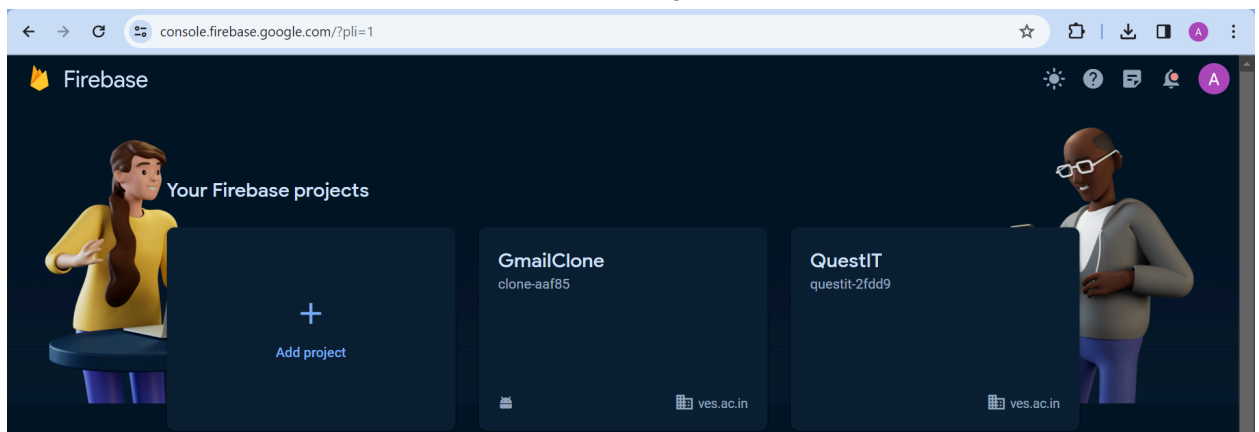
AIM : To Connect Flutter UI with fireBase database

THEORY:

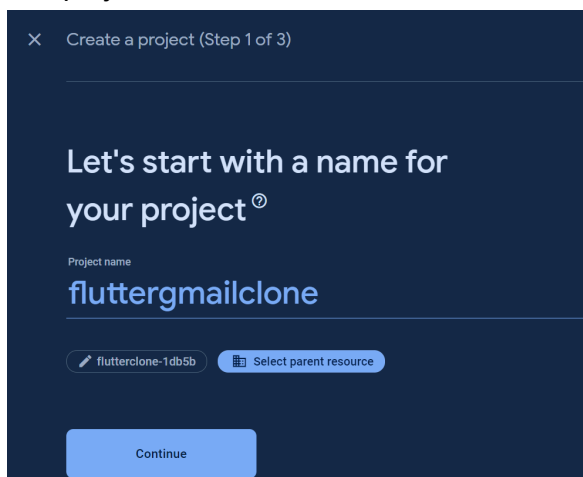
Integrating Firebase with a Flutter application involves several steps, and Firebase offers a variety of backend services that can enhance the functionality of your app. Here's a general process for integrating Firebase with a Flutter application:

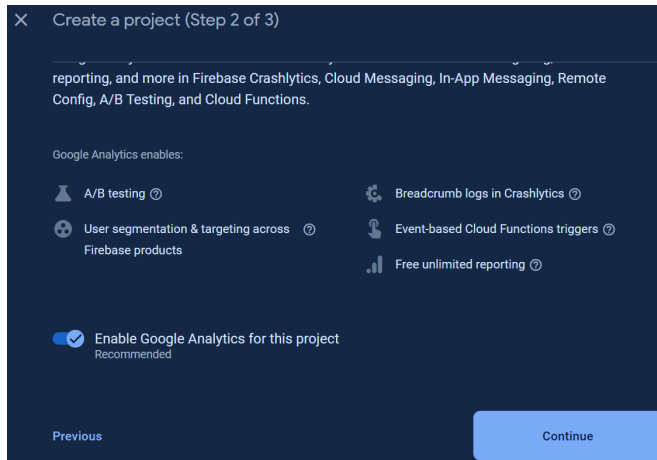
Steps for Firebase Integration in Flutter:

- Go to the **Firestore Console** and click on **"Add Project."**

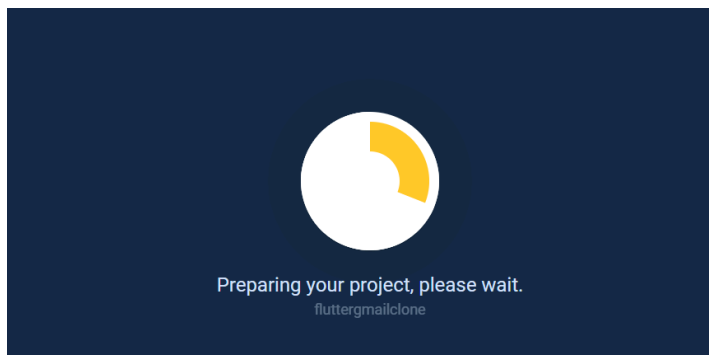


- Add project name:

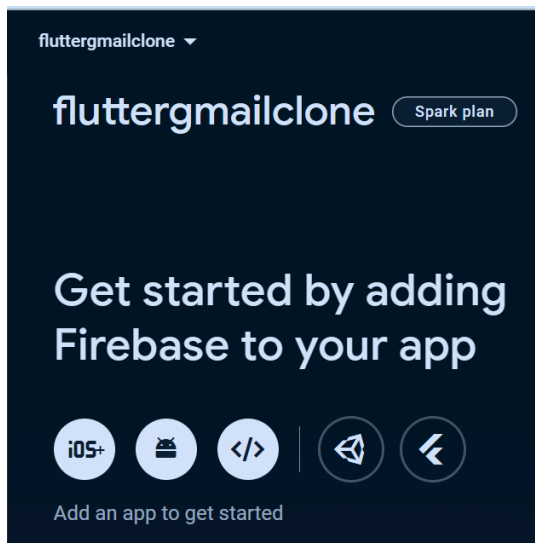




- Create project



- Create app ,click on Android icon



- Register app

The screenshot shows the 'Add Firebase to your Android app' dialog with the title '1 Register app'. It contains three input fields: 'Android package name' with the value 'com.company.appname', 'App nickname (optional)' with the value 'myapp', and 'Debug signing certificate SHA-1 (optional)' with a long hexadecimal string. A 'Register app' button is at the bottom. A note at the bottom states: 'Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.'

- Download the file and place them in the respective directories
app>src>google-services.json

The screenshot shows the '2 Download and then add config file' step. It features a 'Download google-services.json' button, instructions to switch to the Project view in Android Studio, and a diagram showing the file being moved to the 'app' directory. A 'Next' button is at the bottom. On the right, a snippet of the Android Studio Project view shows the file tree with 'google-services.json' highlighted under the 'app' directory. Blue arrows point from the text instructions to the corresponding parts of the Project view.

- Make necessary changes in following files:

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` 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` plugin 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  
    // https://firebase.google.com/docs/android/setup#available-libraries  
}
```

- Continue to console:

4 Next steps

You're all set!

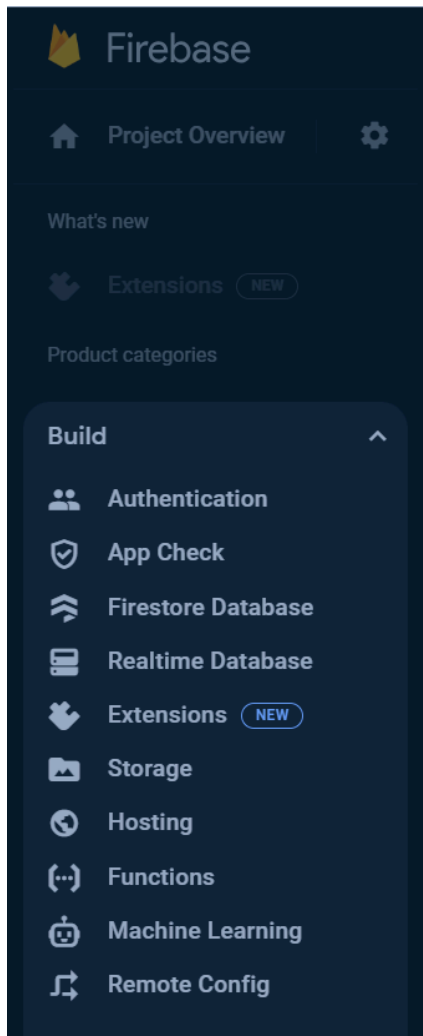
Make sure to check out the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

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

Or, continue to the console to explore Firebase.

[Previous](#) [Continue to console](#)

- Firebase Services like Authentication,Firestore Database,Realtime Database can be used,



- Add Firebase dependencies to your pubspec.yaml file:

pubspec.yaml

dependencies:

firebase_core: ^latest_version

firebase_auth: ^latest_version

cloud_firestore: ^latest_version

Run **flutter pub get** to install the dependencies.

- **Initialize Firebase in your Flutter app**

main.dart file:

```
import 'package:firebase_core/firebase_core.dart';
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(MyApp());
}
```

In models>user.dart:

```
import 'package:gmail_clone/variables.dart';
class User{
  storeuser(email,username,password,profilepic){
    usercollection.doc(email).set({
      'email':email,
      'username':username,
      'password':password,
      'profilepic':profilepic
    });
  }
}
```

Variables.dart:

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

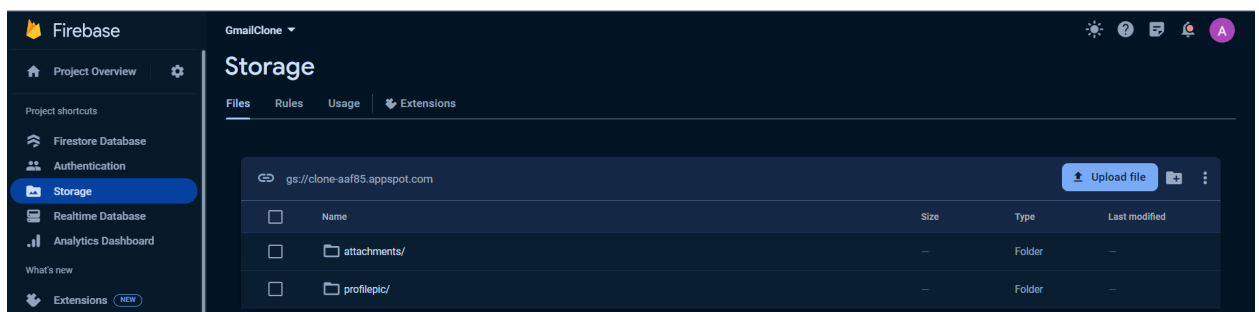
```
TextStyle mystyle(double size, [Color? color, FontWeight? fw]) {
  return GoogleFonts.montserrat(fontSize: size, fontWeight: fw, color: color);
}
```

```
CollectionReference usercollection =
```

```
  FirebaseFirestore.instance.collection('users');
```

```
Reference profilepics = FirebaseStorage.instance.ref().child('profilepic');
```

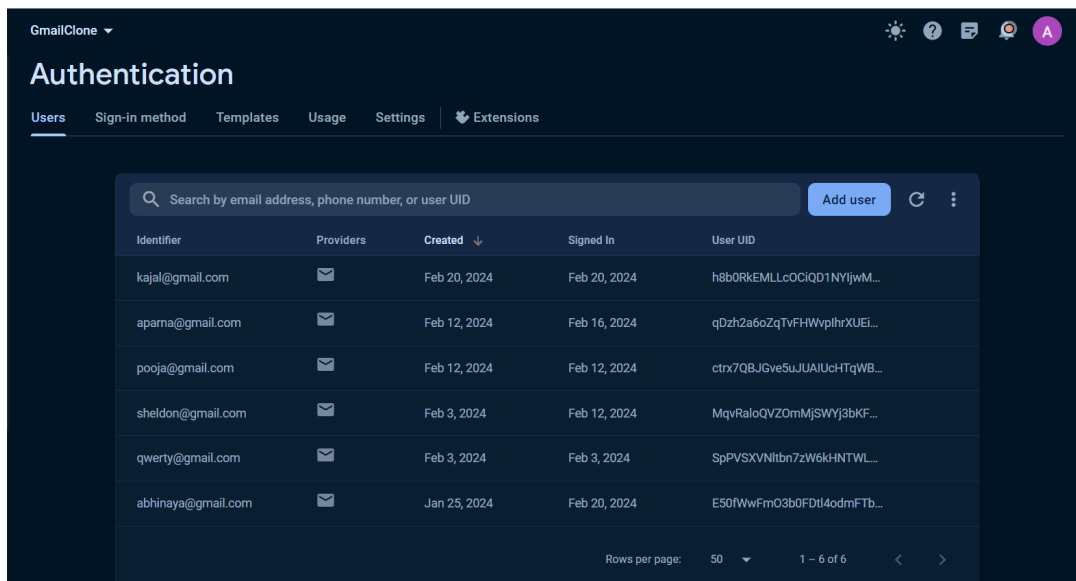
```
Reference attachments = FirebaseStorage.instance.ref().child('attachments');
```









SignUp.dart:

```
registeruser() async {  
  try {  
    String downloadpic = imagepath == null  
      ? 'https://fdlc.org/157-1578186_user-profile-default-image-png-clipart-png/'  
      : await uploadimage();  
    FirebaseAuth.instance  
      .createUserWithEmailAndPassword(  
        email: usernamecontroller.text + '@gmail.com',  
        password: passwordcontroller.text)  
      .then((signeduser) {  
        CustomUser.User().storeuser(usernamecontroller.text + '@gmail.com',  
          usernamecontroller.text, passwordcontroller.text, downloadpic);  
      });  
  }  
}
```

❖ Authenticated Users



Identifier	Providers	Created	Signed In	User UID
kajal@gmail.com		Feb 20, 2024	Feb 20, 2024	h8b0RkEMLLcOCiQD1NYijwM...
aparna@gmail.com		Feb 12, 2024	Feb 16, 2024	qDzh2a6oZqTvFHWvpIhrXUEI...
pooja@gmail.com		Feb 12, 2024	Feb 12, 2024	ctrx7QBjGve5uJUAIUcHTqWB...
sheldon@gmail.com		Feb 3, 2024	Feb 12, 2024	MqvRaloQVZOmMjSWYj3bkKF...
qwerty@gmail.com		Feb 3, 2024	Feb 3, 2024	SpPVsXVNIItbn7zW6kHINTWL...
abhinaya@gmail.com		Jan 25, 2024	Feb 20, 2024	E50fWwFmO3b0FDil4odmFTb...

❖ Mails of each User

Mails.dart:

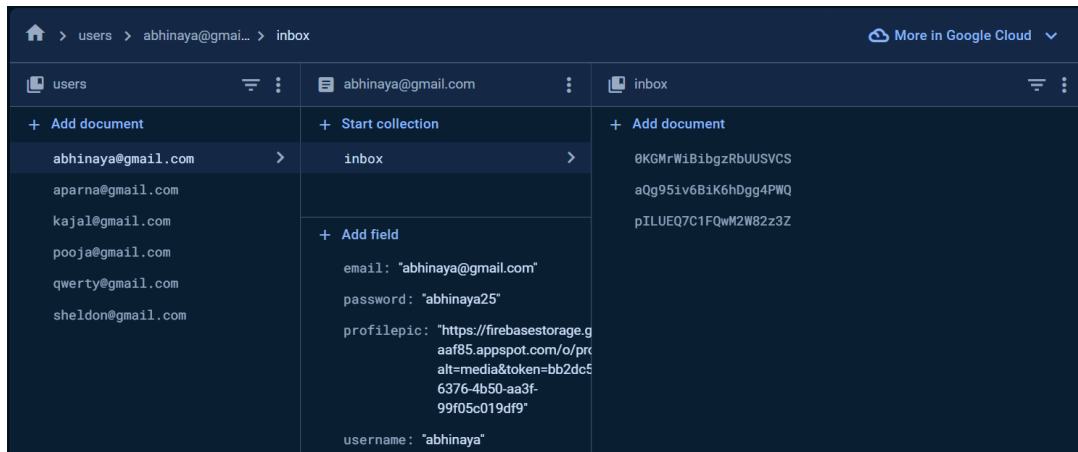
```
class _MailsState extends State<Mails> {
  String? usermail;
  Stream? mystream;
  int _currentIndex = 0;
  @override
  initState() {
    super.initState();
    getuserdata();
    getstream();
  }

  getstream() {
    setState(() {
      mystream = usercollection.doc(usermail).collection('inbox').snapshots();
    });
  }

  getuserdata() async {
    var firebaseuser = FirebaseAuth.instance.currentUser;
    setState(() {
      usermail = firebaseuser!.email;
    });
  }

  starmessage(String id) async {
    DocumentSnapshot document =
      await usercollection.doc(usermail).collection('inbox').doc(id).get();

    if (document['stared'] == false) {
      usercollection
        .doc(usermail)
        .collection('inbox')
        .doc(id)
        .update({'stared': true});
    } else {
      usercollection
        .doc(usermail)
        .collection('inbox')
        .doc(id)
        .update({'stared': false});
    }
  }
}
```

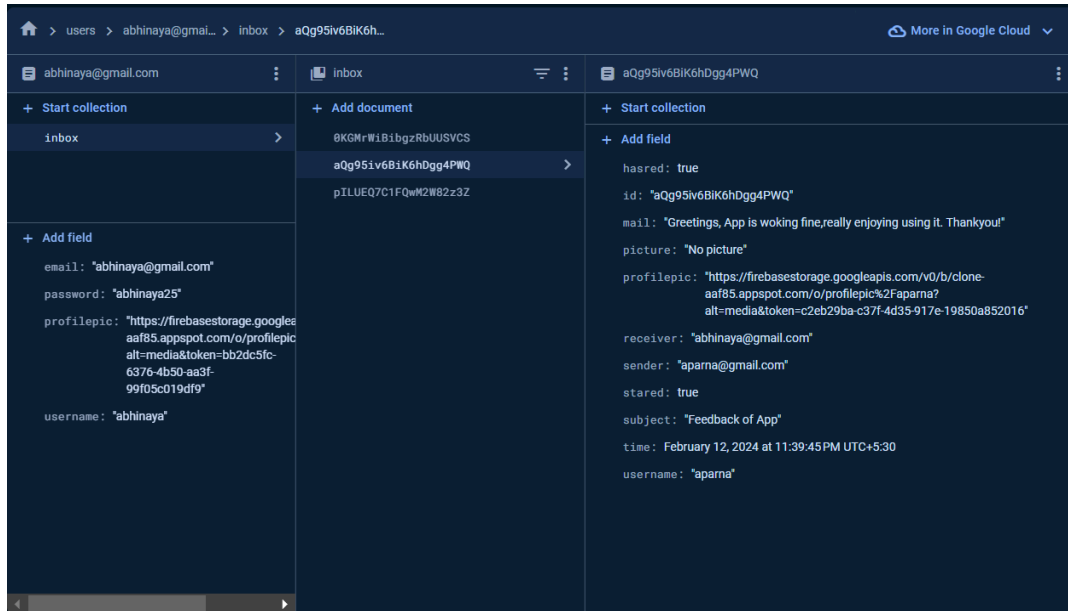
❖ Mail Inbox stored

addMail.dart:

```

sendmail() async {
  try {
    setState(() {
      uploading = true;
    });
    //made own change
    var firebaseuser = FirebaseAuth.instance.currentUser!;
    DocumentSnapshot userdocument =
      await usercollection.doc(firebaseuser.email).get();
    String picture = imagepath == null ? 'No picture' : await uploadimage();
    final id = usercollection.doc(receiver.text).collection('inbox').doc().id;
    usercollection.doc(receiver.text).collection('inbox').doc(id).set({
      'sender': firebaseuser.email,
      'receiver': receiver.text,
      'subject': subject.text,
      'mail': mail.text,
      'hasred': false,
      'stared': false,
      'id': id,
      'picture': picture,
      'time': DateTime.now(),
      'profilepic': userdocument['profilepic'],
      'username': userdocument['username'],
    });
    Navigator.pop(context);
  } catch (e) {
    print(e);
  }
}

```



Conclusion: Connected Flutter UI with fireBase database,where users are authenticated,mail inboxes are stored,attachment and profile pic of each user is stored.
