Experiment 6

Name: Sonali Makhijani
Div: D15A
Roll no: 34

Aim: To Connect Flutter UI with Firebase

Theory:

FlutterFire is a set of Flutter plugins that enable Flutter developers to integrate their applications with various Firebase services. Firebase is a comprehensive mobile and web application development platform provided by Google. FlutterFire is specifically designed to provide Flutter developers with a seamless way to interact with Firebase services.

Key features of FlutterFire include:

1. Firebase Authentication: FlutterFire provides plugins to easily integrate Firebase Authentication, allowing developers to implement user sign-up, sign-in, and password recovery features in their Flutter applications. Firebase supports various authentication methods, including email/password, Google Sign-In, Facebook Sign-In, and more.

2. Cloud Firestore and Realtime Database: FlutterFire supports both Cloud Firestore and Firebase Realtime Database, enabling developers to store and retrieve data in real-time. Firestore is a NoSQL document database, while Realtime Database is a JSON-based database.

3. Cloud Functions: Developers can deploy serverless functions using Cloud Functions for Firebase, and FlutterFire allows Flutter apps to trigger and interact with these functions.

4. Cloud Storage: FlutterFire supports Firebase Cloud Storage, allowing developers to upload, download, and manage files in the cloud. This is useful for handling user-generated content, such as images or videos.

5. Firebase Cloud Messaging (FCM): FCM enables developers to send push notifications to their Flutter applications. FlutterFire provides plugins for integrating FCM and handling push notifications.

6. Firebase Performance Monitoring: Developers can monitor the performance of their Flutter applications using Firebase Performance Monitoring. This includes measuring app startup time, screen rendering, and network performance.

7. Firebase Analytics: FlutterFire includes plugins for integrating Firebase Analytics, enabling developers to gain insights into user behavior and app usage.

8. Firebase Remote Config: FlutterFire supports Firebase Remote Config, allowing developers to remotely configure app behavior without publishing updates. This is useful for A/B testing and feature toggling.

9. Firebase Crashlytics: FlutterFire includes support for Firebase Crashlytics, providing real-time crash reporting to help developers identify and fix issues quickly.

10. Firebase AdMob: FlutterFire includes AdMob plugins for integrating advertisements into Flutter applications using Firebase AdMob.

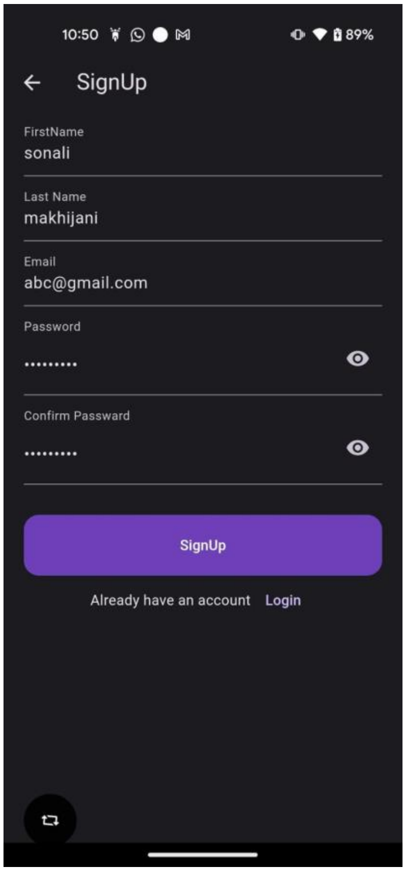
Firebase_options.dart

```
// File generated by FlutterFire CLI.
// ignore_for_file: lines_longer_than_80_chars, avoid_classes_with_only_static_members
import 'package:firebase_core/firebase_core.dart' show FirebaseOptions;
import 'package:flutter/foundation.dart'
    show defaultTargetPlatform, kIsWeb, TargetPlatform;

/// Default [FirebaseOptions] for use with your Firebase apps.
///
/// Example:
/// dart
/// import 'firebase_options.dart';
/// // ...
/// await Firebase.initializeApp(
///   options: DefaultFirebaseOptions.currentPlatform,
/// );
///
class DefaultFirebaseOptions {
  static FirebaseOptions get currentPlatform {
    if (kIsWeb) {
      throw UnsupportedError(
        'DefaultFirebaseOptions have not been configured for web - '
        'you can reconfigure this by running the FlutterFire CLI again.',
      );
    }
    switch (defaultTargetPlatform) {
      case TargetPlatform.android:
        return android;
      case TargetPlatform.iOS:
        return ios;
      case TargetPlatform.macOS:
        throw UnsupportedError(
          'DefaultFirebaseOptions have not been configured for macos - '
          'you can reconfigure this by running the FlutterFire CLI again.',
        );
      case TargetPlatform.windows:
        throw UnsupportedError(
          'DefaultFirebaseOptions have not been configured for windows - '
          'you can reconfigure this by running the FlutterFire CLI again.',
        );
      case TargetPlatform.linux:
        throw UnsupportedError(
          'DefaultFirebaseOptions have not been configured for linux - '
          'you can reconfigure this by running the FlutterFire CLI again.',
        );
      default:
        throw UnsupportedError(
          'DefaultFirebaseOptions are not supported for this platform.',
        );
    }
  }

  static const FirebaseOptions android = FirebaseOptions(
    apiKey: 'AIzaSyA2rMjc46X3ETR4LXPWIIdJOhyxjfe8zsg',
    appId: '1:949515072411:android:af622adb566f70a93ce73e',
    messagingSenderId: '949515072411',
    projectId: 'expensio-et',
    storageBucket: 'expensio-et.appspot.com',
  );

  static const FirebaseOptions ios = FirebaseOptions(
    apiKey: 'AIzaSyDacETYa3tmL17bK4QuFbT26XN3Al-lIu4',
    appId: '1:949515072411:ios:c7d0f53e74e787393ce73e',
    messagingSenderId: '949515072411',
    projectId: 'expensio-et',
    storageBucket: 'expensio-et.appspot.com',
    iosBundleId: 'com.example.expensio',
  );
}
```



SignUp

FirstName
sonali

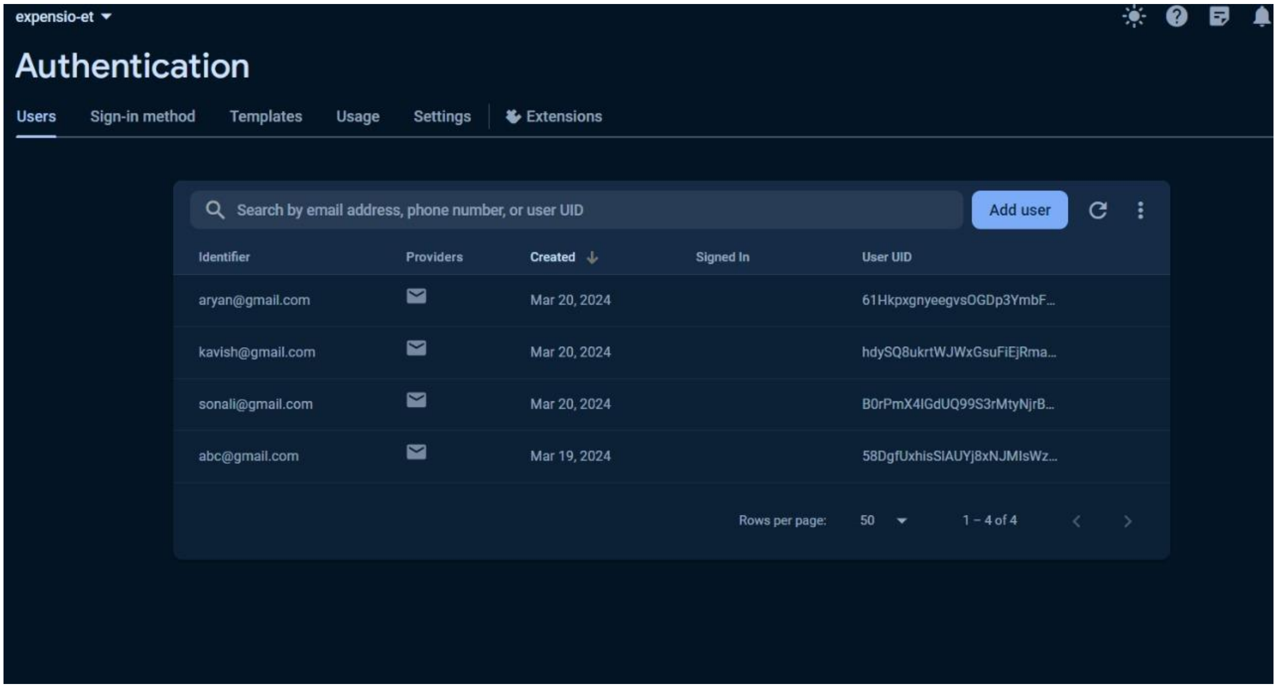
Last Name
makhijani

Email
abc@gmail.com

Password
•••••••••

Confirm Passward
•••••••••

Already have an account Login



Authentication

Identifier	Providers	Created ↓	Signed In	User UID
aryan@gmail.com	✉	Mar 20, 2024		61HkpxgnyeegvsOGDp3YmbF...
kavish@gmail.com	✉	Mar 20, 2024		hdySQ8ukrtWJWxGsuFiEjRma...
sonali@gmail.com	✉	Mar 20, 2024		B0rPmX4lGdUQ99S3rMtyNjrB...
abc@gmail.com	✉	Mar 19, 2024		58DgfUxhisSlAUYj8xNJMlsWz...

Rows per page: 50 1 – 4 of 4

Conclusion: Therefore, understood firebase database and successfully connected it to our project. We enabled creation of new users and authenticated all users.