

**Name: Nishtha Joshi**

**D15B 25**

**MPL experiment 6**

**AIM:** To connect flutter UI with firebase.

Firebase is a Backend-as-a-Service (BaaS) platform provided by Google, offering a suite of tools and services to help developers build, improve, and grow their apps. It encompasses various services such as Realtime Database, Firestore, Authentication, Cloud Functions, Cloud Storage, Cloud Messaging (FCM), Analytics, and more. Firebase aims to streamline app development by offering features like real-time database synchronization, authentication, cloud storage, and hosting, allowing developers to focus more on the frontend development of their applications.

**Prerequisites for using Firebase with Flutter:**

1. Flutter Knowledge: You should have a basic understanding of Flutter framework and Dart programming language as Firebase integration will involve writing Dart code within your Flutter project.
2. Firebase Account: Sign up for a Firebase account at [firebase.google.com](https://firebase.google.com). Once you have an account, you can create a new Firebase project through the Firebase Console.
3. Flutter Project Setup: Set up a Flutter project using your preferred IDE or command-line tools. Ensure that your Flutter environment is correctly configured and up-to-date.
4. Firebase SDK Integration: Add the necessary Firebase SDK dependencies to your Flutter project. This typically involves modifying the `pubspec.yaml` file to include the required Firebase SDK packages.
5. Android/iOS Setup: For Android and iOS platforms, you'll need to configure your project to work with Firebase by adding configuration files provided by Firebase to your project. This involves adding the `google-services.json` file for Android and `GoogleService-Info.plist` file for iOS.

## How Firebase is used in Flutter:

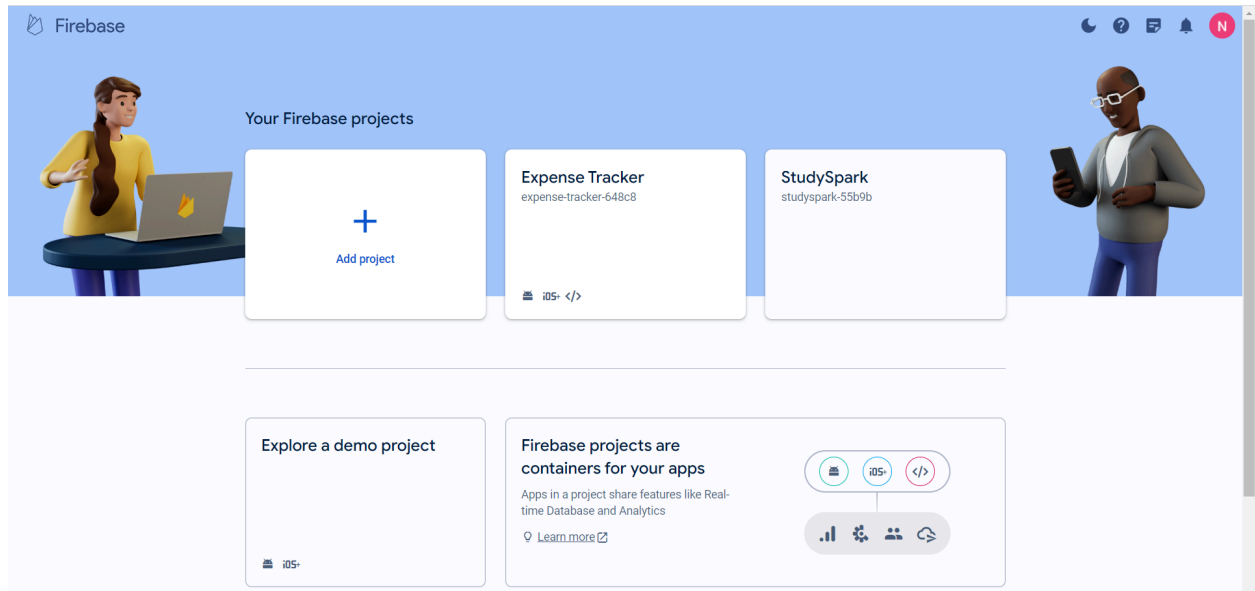
1. Authentication: Firebase Authentication provides easy-to-use SDKs and ready-made UI libraries to authenticate users to your app. It supports various authentication methods like email/password, phone number, Google sign-in, Facebook sign-in, etc.
2. Realtime Database and Firestore: Firebase Realtime Database and Firestore are NoSQL databases provided by Firebase. They allow you to store and sync data between your users in real-time. Firestore is a more scalable and feature-rich version of Realtime Database.
3. Cloud Functions: Firebase Cloud Functions allow you to run backend code in response to events triggered by Firebase features and HTTPS requests. This enables you to perform complex backend operations without managing your own server.
4. Cloud Storage: Firebase Cloud Storage provides a secure and scalable object storage solution for storing user-generated content such as images, videos, and files.
5. Cloud Messaging: Firebase Cloud Messaging (FCM) enables you to send notifications and messages to users across platforms, including Android, iOS, and web.

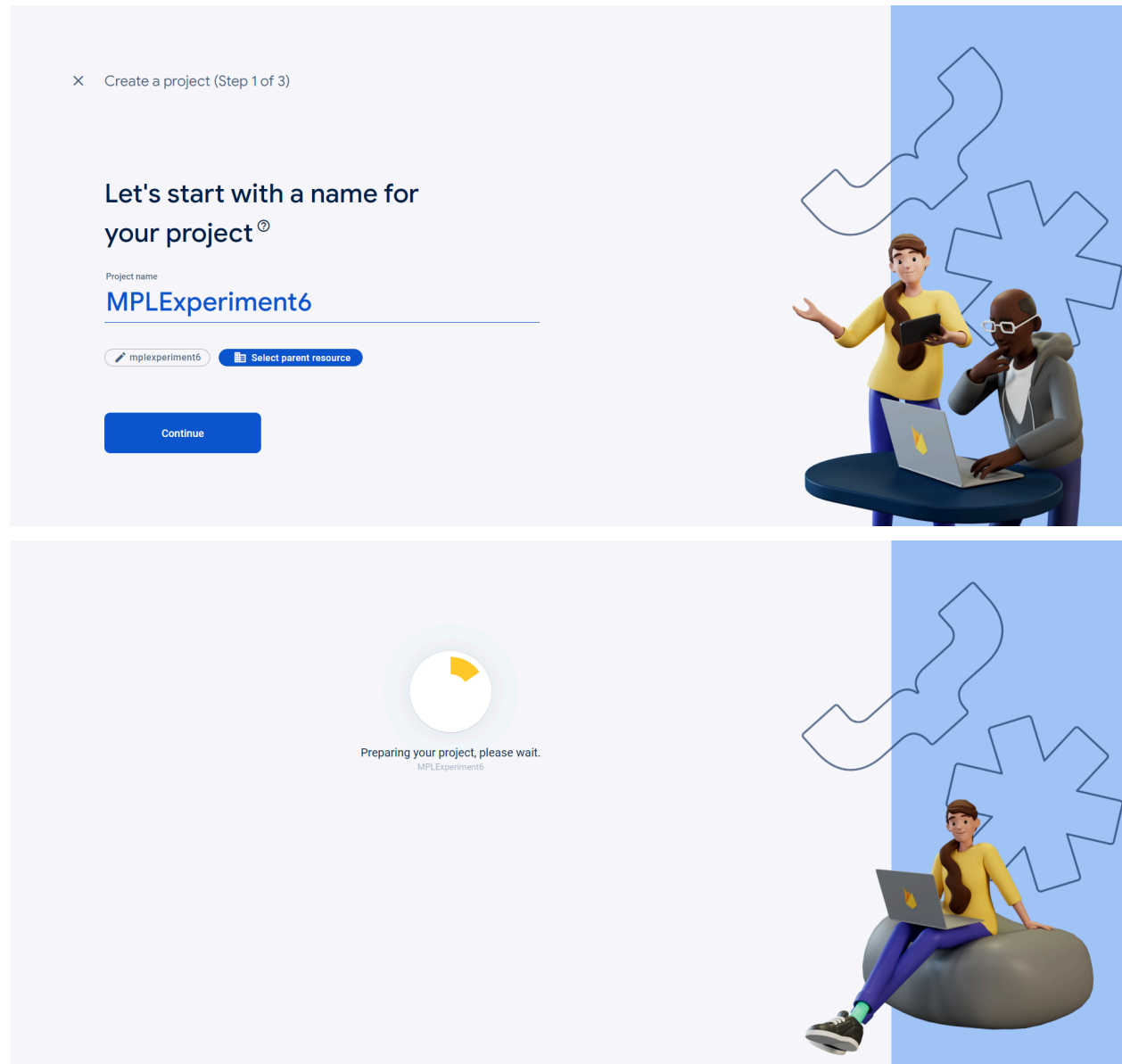
## Important commands for using Firebase in Flutter:

1. `firebase_core`: This is the core package required for any Firebase service integration in Flutter. Add it to your `pubspec.yaml` file:

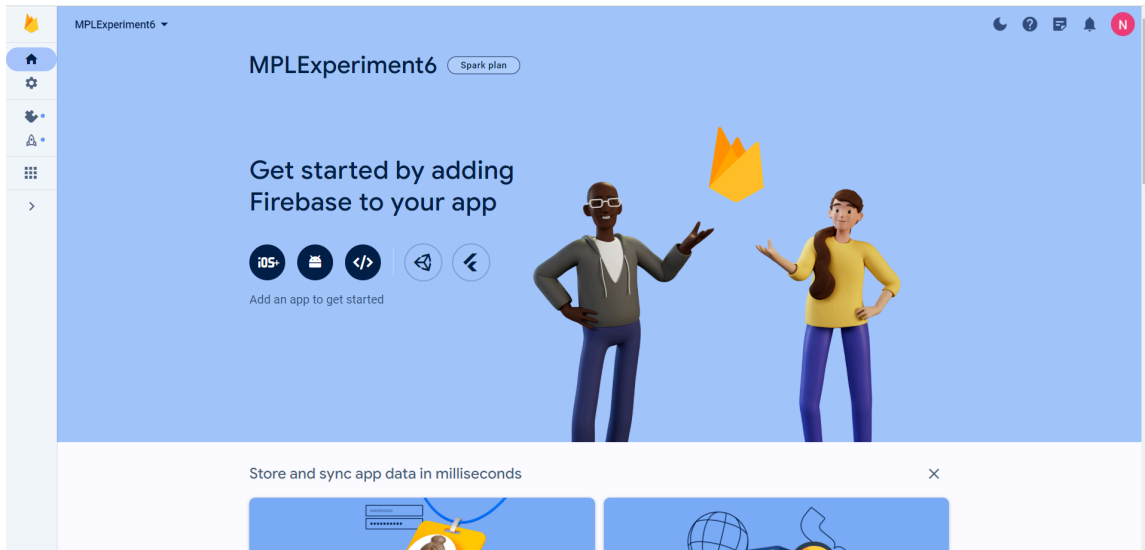
```
dependencies:  
  flutter:  
    sdk: flutter  
  firebase_core: ^1.10.0
```

## Step 1 : Go to firebase console and create a new project.





**Step 2:** Now click on flutter icon  
This will create a flutter firebase project



## × Add Firebase to your Flutter app

### 1 Prepare your workspace

The easiest way to get you started is to use the FlutterFire CLI.

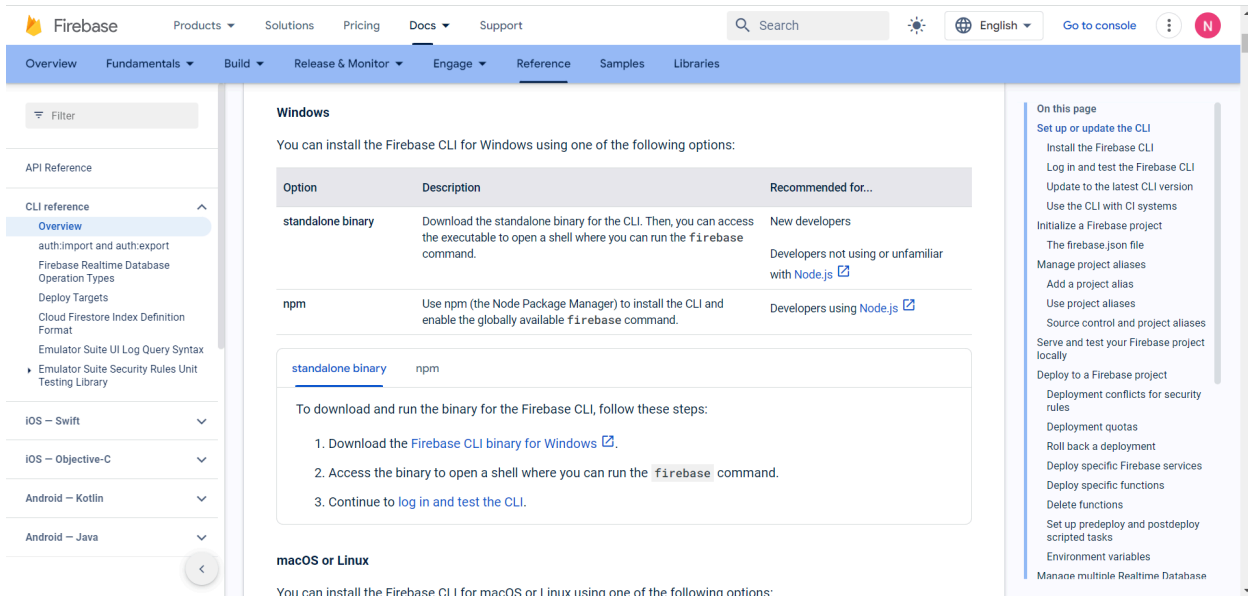
Before you continue, make sure to:

- Install the [Firebase CLI](#) and log in (run `firebase login`)
- Install the [Flutter SDK](#)
- Create a Flutter project (run `flutter create`)

Next

### 2 Install and run the FlutterFire CLI

### 3 Initialize Firebase and add plugins



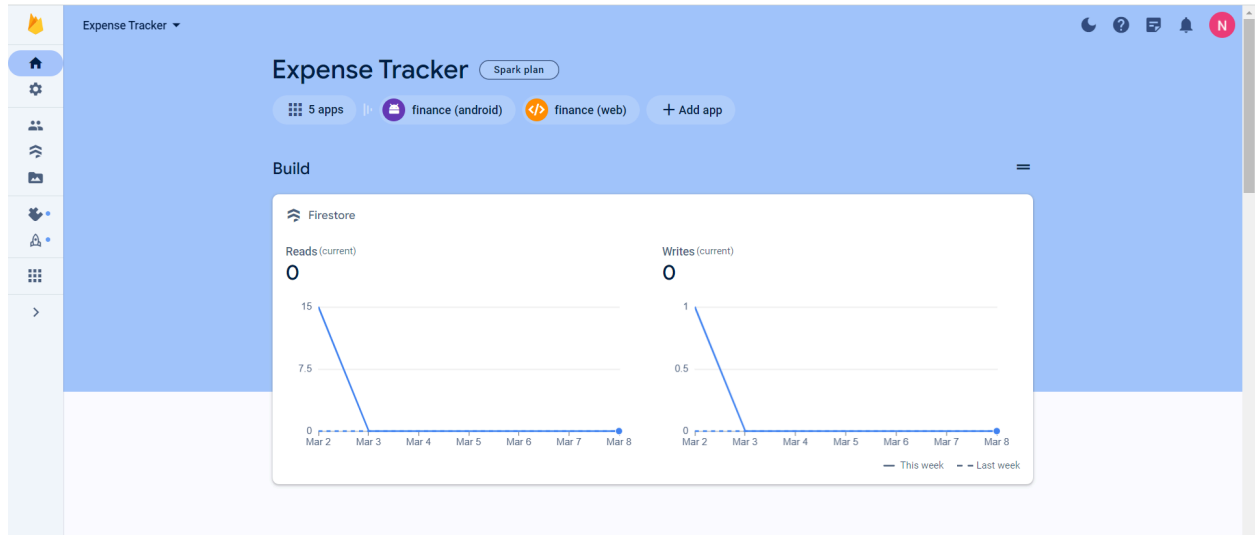
**Step 3:** Make a new flutter project in vscode and run the following commands

```
PS C:\Users\Joshi\OneDrive\Desktop\flutter_6\my_app> dart pub global activate flutterfire_cli
Package flutterfire_cli is currently active at version 0.2.7.
The package flutterfire_cli is already activated at newest available version.
To recompile executables, first run `dart pub global deactivate flutterfire_cli`.
Installed executable flutterfire.
Activated flutterfire_cli 0.2.7.
PS C:\Users\Joshi\OneDrive\Desktop\flutter_6\my_app>
```

```
PS C:\Users\Joshi\OneDrive\Desktop\flutter_6\my_app> flutterfire configure --project=mplexperiment6
i Found 3 Firebase projects. Selecting project mplexperiment6.
? Which platforms should your configuration support (use arrow keys & space to select)? >
✓ android
✓ ios
✓ macos
✓ web
```

Now the firebase is successfully added in your project

## Firestore connectivity in My App - Expense Tracker



Expense Tracker

### Authentication

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

Search by email address, phone number, or user UID

Add user

Identifier	Providers	Created ↓	Signed in	User UID
akshatg123@gmail.com	📧	Mar 2, 2024	Mar 2, 2024	hXLVg7S8ILf0ToZq4W2b5rEU...
joshinishtha99@gmail.com	📧	Feb 22, 2024	Mar 8, 2024	Cw8ttha0IzNZTcEc5v9U3oBtx...

Rows per page: 50 | 1 - 2 of 2

## Login.dart

```
import 'package:expense/Screens/home.dart';
import 'package:expense/Screens/register.dart';
import 'package:expense/widgets/bottomnavigationbar.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/foundation.dart';
```

```

import 'package:flutter/material.dart';
import 'package:expense/Screens/authentication.dart';

class Login extends StatefulWidget {
  const Login({Key? key});

  @override
  State<Login> createState() => _LoginState();
}

class _LoginState extends State<Login> {

  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController =
TextEditingController();

  @override
  void dispose() {
    super.dispose();
    emailController.dispose();
    passwordController.dispose();
  }

  void login() async {
    try {
      await FirebaseAuth.instance.signInWithEmailAndPassword(
        email: emailController.text, password: passwordController.text);
      Navigator.pop(context);
      Navigator.push(context, MaterialPageRoute(builder: (context) {
        return const Bottom();
      }));
    } on FirebaseAuthException catch (e) {
      if (e.code == 'user-not-found') {
        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(content: Text('No user found for that
email')));
      } else if (e.code == 'wrong-password') {
        ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
          content: Text('Wrong password provided for that user')));
      }
    }
  }
}

```



```

    } catch (e) {
      print(e);
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Color(0xFF8A2BE2),

      appBar: AppBar(
        backgroundColor: Colors.white,
        title: Container(
          padding: const EdgeInsets.all(10),
          child: const Row(
            mainAxisAlignment: MainAxisAlignment.min,
            children: [
              Text(
                'Expense Tracker',
                style: TextStyle(
                  color: Colors.black,
                  fontSize: 25,
                ),
              ),
              SizedBox(width: 10),
              Text(
                '$',
                style: TextStyle(
                  fontSize: 30,
                ),
              ),
            ],
          ),
        ),
        centerTitle: true,
        elevation: 2,
      ),
      body: SingleChildScrollView(
        child: Center(
          child: Column(

```

```
mainAxisAlignment: MainAxisAlignment.center,
children: [
  const SizedBox(height: 50),
  const Text(
    'Welcome to Expense Tracker!',
    style: TextStyle(
      color: Colors.white,
      fontSize: 25,
    ),
  ),
  const SizedBox(height: 20),
  Padding(
    padding: const EdgeInsets.all(8.0),
    child: TextField(
      controller: emailController,
      decoration: InputDecoration(
        fillColor: Colors.white,
        filled: true,
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(10),
        ),
        labelText: 'Email',
      ),
    ),
  ),
  const SizedBox(height: 20),
  Padding(
    padding: const EdgeInsets.all(8.0),
    child: TextField(
      obscureText: true,
      controller: passwordController,
      decoration: InputDecoration(
        fillColor: Colors.white,
        filled: true,
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(10),
        ),
        labelText: 'Password',
      ),
    ),
  ),
],
```

```

    ),
    const SizedBox(height: 20),
    ElevatedButton(
      onPressed: () {
        login();
      },
      style: ElevatedButton.styleFrom(
        backgroundColor: const Color(0xFFFFE5C1),
        padding:
          const EdgeInsets.symmetric(horizontal: 50, vertical:
10),

        textStyle: const TextStyle(
          fontSize: 20,
        ),
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(10),
        ),
      ),
      child: const Text('Login'),
    ),
    const SizedBox(height: 20),
    Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        const Text(
          'Don\'t have an account?',
          style: TextStyle(
            color: Colors.white,
            fontSize: 20,
          ),
        ),
        TextButton(
          onPressed: () {
            Navigator.pop(context);
            Navigator.push(context, MaterialPageRoute(
              builder: (context) {
                return const Register();
              },
            ));
          },
        ),
      ],
    ),
  ),
)

```

```

        child: const Text(
          'Sign Up',
          style: TextStyle(
            color: Colors.black,
            fontSize: 20,
          ),
        ),
      ),
    ],
  ),
  const SizedBox(height: 10),
  GestureDetector(
    onTap: () async {
      bool auth = await Authetication.authetication();
      print("can autheticate:$auth");
      if(auth){
        Navigator.push(context, MaterialPageRoute(builder:
(context)=>const Home()

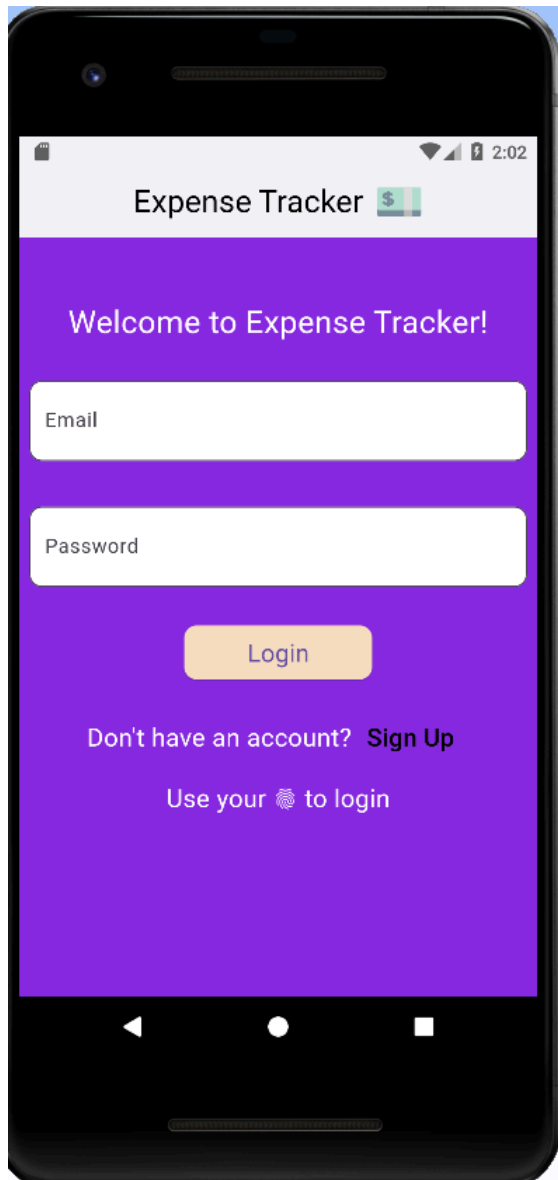
        ));
      }
    },
    child: const Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Text(
          'Use your ',
          style: TextStyle(
            color: Colors.white,
            fontSize: 20,
          ),
        ),
        Icon(
          Icons.fingerprint,
          size: 20,
          color: Colors.white,
        ),
        Text(
          ' to login',
          style: TextStyle(

```

```

        color: Colors.white,
        fontSize: 20,
    ),
),
),
1,
),
),
1,
),
),
),
),
),
),
);
}
}

```



## Register.dart

```
import 'package:expense/Screens/login.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

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

```

    @override
    State<Register> createState() => _RegisterState();
}

class _RegisterState extends State<Register> {
    Color newPurple = const Color(0xFFC6C1FF);
    Color newOrange = const Color(0xFFFFE5C1);
    final TextEditingController emailController = TextEditingController();
    final TextEditingController passwordController =
TextEditingController();
    final TextEditingController confirmPasswordController
=TextEditingController();
    final TextEditingController phoneController = TextEditingController();
    final TextEditingController nameController = TextEditingController();
    final TextEditingController pinController = TextEditingController();
    final FirebaseAuth _auth = FirebaseAuth.instance;
    final FirebaseFirestore _firestore = FirebaseFirestore.instance;
    @override
    void dispose() {
        super.dispose();
        emailController.dispose();
        passwordController.dispose();
        confirmPasswordController.dispose();
        phoneController.dispose();
    }

    void signUp() async {
        if (passwordController.text == confirmPasswordController.text) {
            try {
                await _auth.createUserWithEmailAndPassword(
                    email: emailController.text, password:
passwordController.text);
                debugPrint('User Registered ${_auth.currentUser}');
                await
_firestore.collection('users').doc(_auth.currentUser!.uid).set({
                    'name': nameController.text,
                    'email': emailController.text,
                    'phone': phoneController.text,

```

```

    ));
    _auth.currentUser!.updateDisplayName(nameController.text);
    debugPrint('User added to firestore');
    Navigator.pop(context);
    Navigator.push(context, MaterialPageRoute(builder: (context) {
      return const Login();
    }));
  } on FirebaseAuthException catch (e) {
    if (e.code == 'weak-password') {
      ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
        content: Text('The password provided is too weak')));
    } else if (e.code == 'email-already-in-use') {
      ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
        content: Text('The account already exists for that
email')));
    }
  } catch (e) {
    print(e);
  }
} else {
  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(content: Text('Passwords do not match')));
}
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Color(0xFF8A2BE2),

    appBar: AppBar(
      backgroundColor: Colors.white,
      title: const Text('Register ' ,
        style: TextStyle(
          color: Colors.black,
          fontSize: 25,
        )),
      foregroundColor: Colors.black,
      centerTitle: true,

```



```

        elevation: 2,
    ),
    body: SingleChildScrollView(
        child: Center(

            child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                    SizedBox(height: 20),
                    const Text(
                        'Welcome to Expense Tracker',
                        style: TextStyle(
                            color: Colors.white,
                            fontSize: 25,
                        ),
                    ),
                    const SizedBox(height: 20),
                    Padding(
                        padding: const EdgeInsets.all(8.0),
                        child: TextField(
                            controller: nameController,
                            decoration: InputDecoration(
                                fillColor: Colors.white,
                                filled: true,
                                border: OutlineInputBorder(
                                    borderRadius: BorderRadius.circular(10),
                                ),
                                labelText: 'Name',
                            ),
                        ),
                    ),
                    Padding(
                        padding: const EdgeInsets.all(8.0),
                        child: TextField(
                            controller: emailController,
                            decoration: InputDecoration(
                                fillColor: Colors.white,
                                filled: true,
                                border: OutlineInputBorder(
                                    borderRadius: BorderRadius.circular(10),
                                ),
                            ),
                        ),
                    ),
                ],
            ),
        ),
    ),

```

```

    ),
    labelText: 'Email',
  ),
),
),
),
Padding(
  padding: const EdgeInsets.all(8.0),
  child: TextField(
    controller: phoneController,
    decoration: InputDecoration(
      fillColor: Colors.white,
      filled: true,
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(10),
      ),
      labelText: 'Phone Number',
    ),
    keyboardType: TextInputType.phone,
  ),
),
Padding(
  padding: const EdgeInsets.all(8.0),
  child: TextField(
    controller: passwordController,
    decoration: InputDecoration(
      fillColor: Colors.white,
      filled: true,
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(10),
      ),
      labelText: 'Password',
    ),
  ),
),
),
Padding(
  padding: const EdgeInsets.all(8.0),
  child: TextField(
    obscureText: true,
    controller: confirmPasswordController,
    decoration: InputDecoration(

```

```

        fillColor: Colors.white,
        filled: true,
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(10),
        ),
        labelText: 'Confirm Password',
      ),
    ),
  ),
),

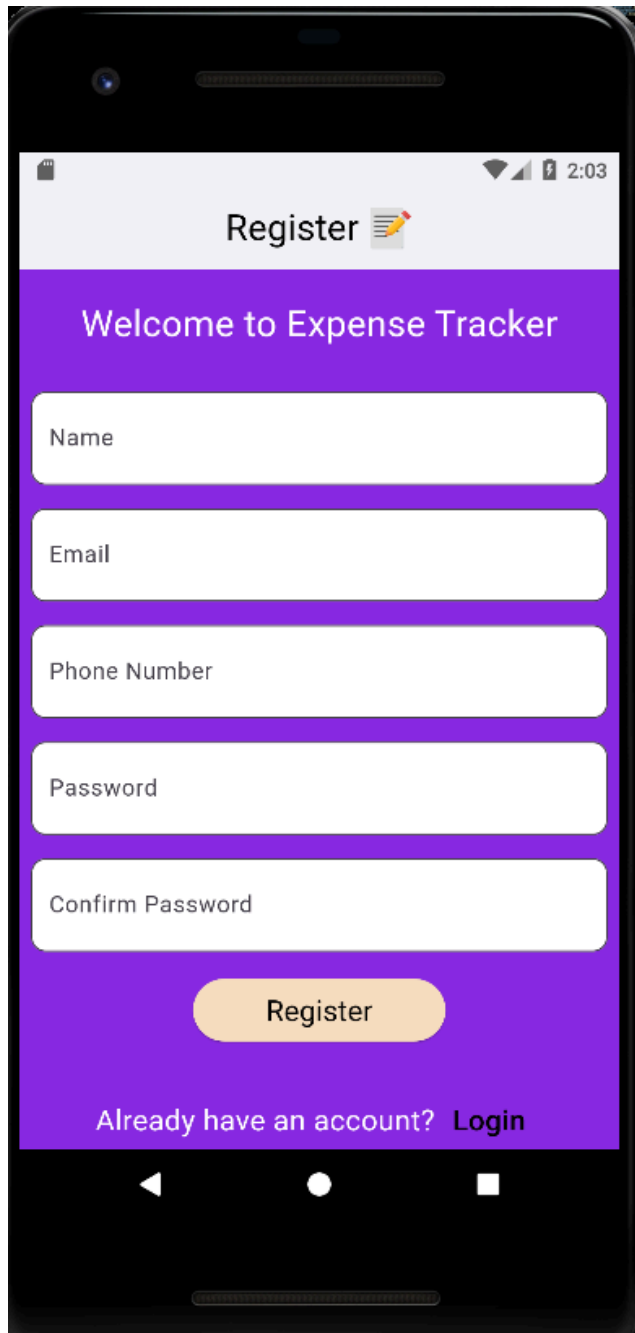
Padding(
  padding: const EdgeInsets.all(8.0),
  child: ElevatedButton(
    onPressed: () {
      signUp();
    },
    style: ElevatedButton.styleFrom(
      backgroundColor: newOrange,
      foregroundColor: Colors.black,
      padding:
        const EdgeInsets.symmetric(horizontal: 50, vertical:
10),

      textStyle: const TextStyle(
        fontSize: 20,
      ),
    ),
    child: const Text('Register'),
  ),
),
const SizedBox(height: 20),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    const Text('Already have an account?',
      style: TextStyle(
        color: Colors.white,
        fontSize: 20,
      )),
    TextButton(
      onPressed: () {

```

```
        Navigator.pop(context);
        Navigator.push(context, MaterialPageRoute(builder:
(context) {
            return const Login();
        }));
    },
    child: const Text('Login',

        style: TextStyle(
            color: Colors.black,
            fontSize: 20,
        )),
    ),
    ],
),
)),
),
);
}
```



**Conclusion :** Thus, we have successfully understood what firebase is and we have added it in our project by installing packages and adding necessary dependencies. We also created a login page and a signup page and successfully implemented authentication of users.