Project Title: BigBasket RollNo.27

MAD & PWA Lab Journal

Experiment No.	06
Experiment Title.	To connect flutter UI with firebase database.
Roll No.	27
Name	Rujuta Medhi
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO1: Understand cross platform mobile application development using Flutter framework
Grade:	

AIM : To connect flutter UI with firebase database.

THEORY:

Introduction to Firebase and Flutter Integration

Firebase is a comprehensive platform developed by Google, designed to help developers build high-quality applications for both mobile and web. It provides essential services such as real-time databases, authentication, cloud storage, hosting, and much more. One of the most widely used Firebase services is the Firebase Realtime Database, which is a NoSQL cloud database that allows data to be stored and synced in real-time across all connected devices. Flutter, on the other hand, is an open-source UI software development kit created by Google, which allows developers to build natively compiled applications for mobile, web, and desktop from a single codebase. Its rich set of pre-designed widgets and powerful tools makes Flutter an attractive option for developing visually appealing and performant applications. Integrating Firebase with Flutter allows developers to leverage the full potential of Firebase services in their applications. By using Firebase's Realtime Database, Flutter apps can achieve features such as real-time data synchronization, secure authentication, and cloud-based storage. This combination enables developers to create powerful, scalable, and feature-rich mobile and web applications.

Setting Up Firebase in Flutter :

To connect a Flutter app with Firebase, the following steps are typically followed:

1. Creating a Firebase Project: To start using Firebase with Flutter, the first step is to create a Firebase project in the Firebase Console. Once the project is created, developers can associate their Flutter app with the Firebase project by following the platform-specific instructions for Android or iOS. This usually involves configuring API keys, downloading configuration files, and adding them to the Flutter project.

2. Integrating Firebase SDK in Flutter: After the Firebase project is set up, developers need to integrate Firebase's SDK into the Flutter app. This involves adding the necessary dependencies to the Flutter project's pubspec.yaml file. For Firebase's Realtime Database, the package firebase_database is used. Additionally, Firebase's core SDK (firebase_core) must also be included to initialize Firebase services.

3. Initializing Firebase: Before any Firebase functionality can be used, it is essential to initialize Firebase in the Flutter app. This is done by calling Firebase.initializeApp() in the main entry point of the app (usually in the main.dart file). Firebase needs to be initialized before interacting with any Firebase services, such as the Realtime Database, Cloud Firestore, or Authentication. Connecting Firebase to a Flutter app enables developers to create robust, scalable, and real-time applications with ease. Firebase's Realtime Database offers a powerful, cloud-based solution for managing data in realtime, while Firebase Authentication ensures secure access control. By integrating Firebase with Flutter, developers can take advantage of

real-time data synchronization, offline support, and a wide range of other Firebase features, allowing them to build feature-rich apps that meet modern user expectations.

PROGRAM:

```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'home.dart';
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  try {
    await Firebase.initializeApp(
      options: DefaultFirebaseOptions.currentPlatform,
    );
    print("✅ Firebase Connection Successful");
  } catch (e) {
    print("❌ Firebase Initialization Failed: $e");
  }
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Login Screen',
      theme: ThemeData(primarySwatch: Colors.red),
      home: const LoginScreen(),
    );
  }
}



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

  @override
```

```
  _LoginScreenState createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final FirebaseAuth _auth = FirebaseAuth.instance;
  bool _isLoading = false;

  void _loginUser() async {
    setState(() {
      _isLoading = true;
    });

    try {
      UserCredential userCredential = await _auth.signInWithEmailAndPassword(
        email: _emailController.text.trim(),
        password: _passwordController.text.trim(),
      );

      User? user = userCredential.user;

      if (user != null) {
        // Check if user exists in Firestore
        DocumentSnapshot userDoc = await FirebaseFirestore.instance
            .collection("users")
            .doc(user.uid)
            .get();

        if (userDoc.exists) {
          // Navigate to HomeScreen after successful login
          Navigator.pushReplacement(
            context,
            MaterialPageRoute(builder: (context) => const HomeScreen()), // Ensure HomeScreen
is correct
          );
        } else {
          // User not found in Firestore (unlikely), show error
          ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(content: Text("User record not found.")),
          );
        }
      }
    } on FirebaseAuthException catch (e) {
```

```
    String errorMessage = "Login failed. Please try again.";

    if (e.code == 'user-not-found') {
      errorMessage = "No user found for this email.";
    } else if (e.code == 'wrong-password') {
      errorMessage = "Wrong password provided.";
    }

    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text(errorMessage)),
    );
  } catch (e) {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text("An error occurred: $e")),
    );
  } finally {
    setState(() {
      _isLoading = false;
    });
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: SingleChildScrollView(
      child: Column(
        children: [
          Container(
            height: 250,
            decoration: const BoxDecoration(
              image: DecorationImage(
                image: AssetImage('assets/loginnn.png'),
                fit: BoxFit.cover,
              ),
            ),
          ),
          Padding(
            padding: const EdgeInsets.all(24.0),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.stretch,
              children: [
                const Text('Login',
                    style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold)),
```

```
const SizedBox(height: 24),
Row(
  children: [
    Expanded(
      child: ElevatedButton.icon(
        onPressed: () {},
        icon: const Icon(Icons.facebook, color: Colors.white),
        label: const Text('FACEBOOK',
            style: TextStyle(color: Colors.white)),
        style: ElevatedButton.styleFrom(
          backgroundColor: const Color(0xFF3B5998),
        ),
      ),
    ),
    const SizedBox(width: 16),
    Expanded(
      child: OutlinedButton.icon(
        onPressed: () {},
        icon: const Icon(Icons.g_mobiledata, color: Colors.black87),
        label: const Text('GOOGLE',
            style: TextStyle(color: Colors.black87)),
      ),
    ),
  ],
),
const SizedBox(height: 24),
TextField(
  controller: _emailController,
  decoration: const InputDecoration(
    hintText: 'Email',
    border: OutlineInputBorder(),
  ),
),
const SizedBox(height: 16),
TextField(
  controller: _passwordController,
  obscureText: true,
  decoration: const InputDecoration(
    hintText: 'Password',
    border: OutlineInputBorder(),
  ),
),
const SizedBox(height: 24),
ElevatedButton(
```

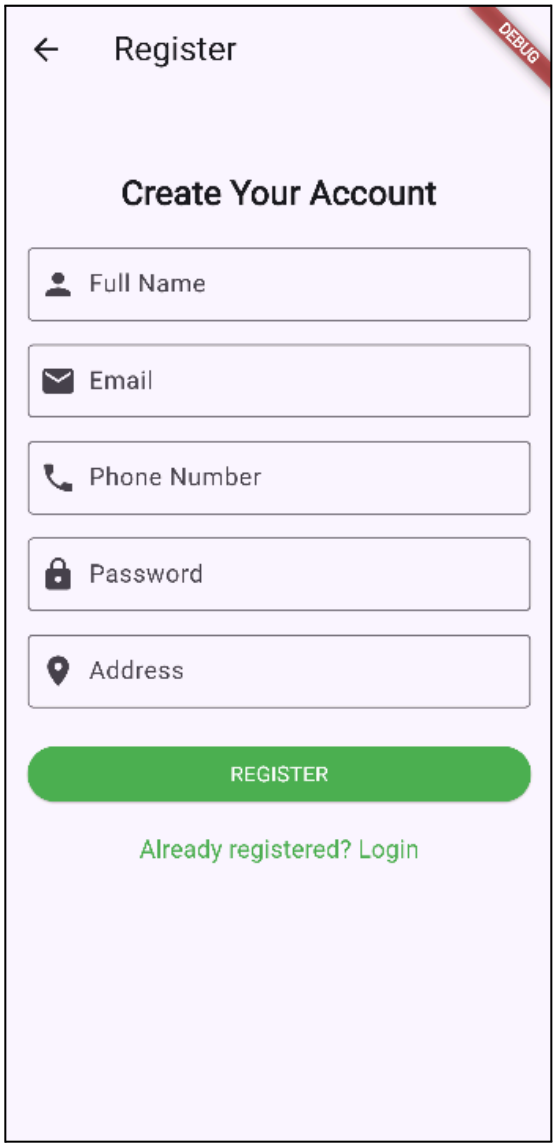
```
                onPressed: _isLoading ? null : _loginUser,
                style: ElevatedButton.styleFrom(
                  backgroundColor: Colors.green,
                ),
                child: _isLoading
                    ? const CircularProgressIndicator(color: Colors.white)
                    : const Text('LOGIN', style: TextStyle(color: Colors.white)),
              ),
              const SizedBox(height: 24),
              TextButton(
                onPressed: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(builder: (context) => const RegisterScreen()),
                  );
                },
                child: const Text('REGISTER HERE', style: TextStyle(color: Colors.green)),
              ),
            ],
          ),
        ),
      ),
    ],
  ),
 ),
 );
 }
}
```

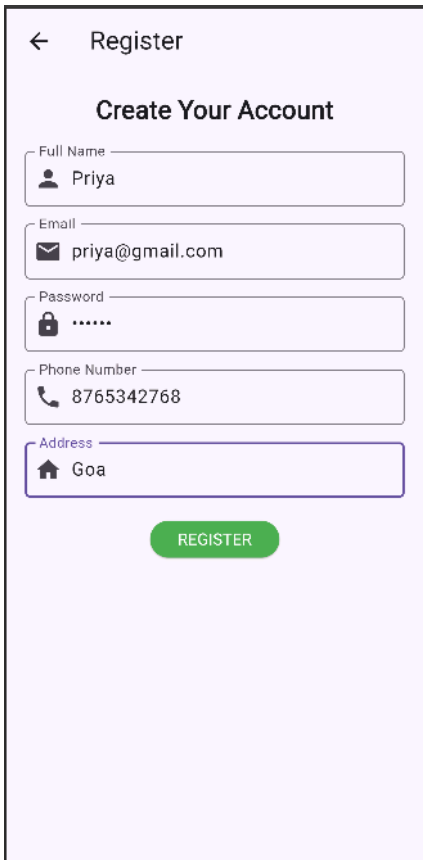
```
class RegisterScreen extends StatefulWidget {
 const RegisterScreen({super.key});

 @override
 _RegisterScreenState createState() => _RegisterScreenState();
}

class _RegisterScreenState extends State<RegisterScreen> {
 final _formKey = GlobalKey<FormState>();
 final TextEditingController _nameController = TextEditingController();
 final TextEditingController _emailController = TextEditingController();
 final TextEditingController _phoneController = TextEditingController();
 final TextEditingController _passwordController = TextEditingController();
```

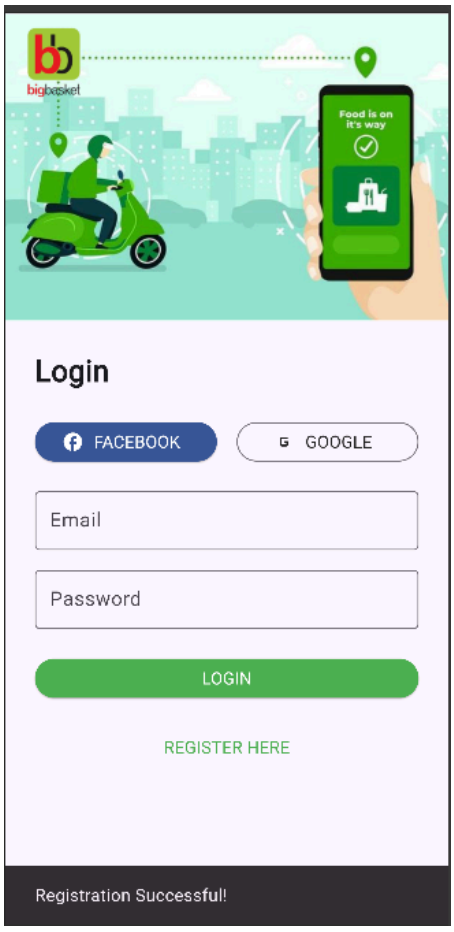
```
  final TextEditingController _addressController = TextEditingController();

  // Function to register user
  void registerUser() async {
    if (_formKey.currentState!.validate()) {
      try {
        // Register user in Firebase Authentication
        UserCredential userCredential = await
FirebaseAuth.instance.createUserWithEmailAndPassword(
          email: _emailController.text.trim(),
          password: _passwordController.text.trim(),
        );

        // Get the user ID
        String userId = userCredential.user!.uid;

        // Store user details in Firestore
        await FirebaseFirestore.instance.collection("users").doc(userId).set({
          "name": _nameController.text.trim(),
          "email": _emailController.text.trim(),
          "phone": _phoneController.text.trim(),
          "address": _addressController.text.trim(),
          "timestamp": FieldValue.serverTimestamp(),
        });

        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(content: Text('Registration Successful!')),
        );

        Navigator.pop(context); // Go back to Login Screen
      } catch (e) {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text('Registration Failed: $e')),
        );
      }
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Register')),
      body: SingleChildScrollView(
        child: Padding(
```

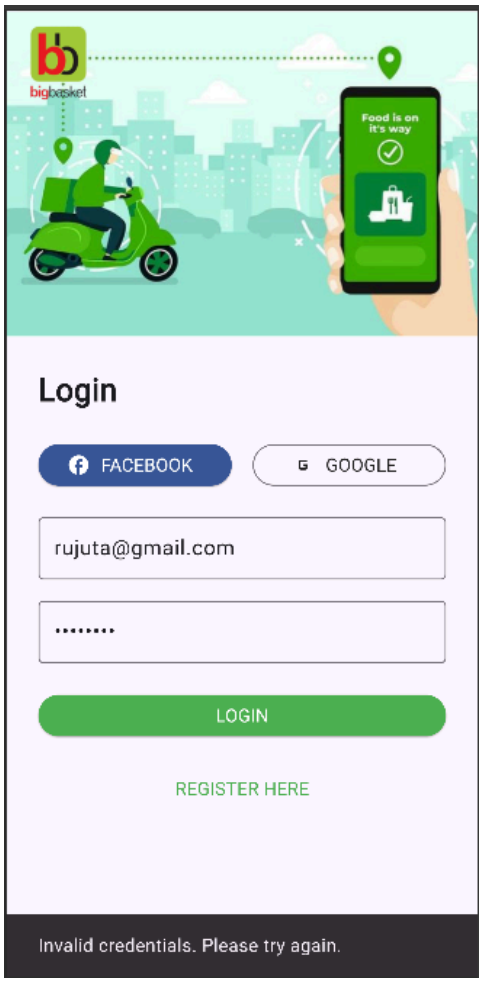
```
padding: const EdgeInsets.all(16.0),
child: Form(
  key: _formKey,
  child: Column(
    children: [
      const Text('Create Your Account',
          style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold)),
      const SizedBox(height: 20),
      TextFormField(
        controller: _nameController,
        decoration: const InputDecoration(
          labelText: 'Full Name',
          border: OutlineInputBorder(),
          prefixIcon: Icon(Icons.person),
        ),
        validator: (value) =>
        value!.isEmpty ? 'Please enter your name' : null,
      ),
      const SizedBox(height: 15),
      TextFormField(
        controller: _emailController,
        decoration: const InputDecoration(
          labelText: 'Email',
          border: OutlineInputBorder(),
          prefixIcon: Icon(Icons.email),
        ),
        validator: (value) =>
        !RegExp(r'^[^@]+@[^@]+\.[^@]+').hasMatch(value!)
            ? 'Enter a valid email'
            : null,
      ),
      const SizedBox(height: 15),
      TextFormField(
        controller: _passwordController,
        obscureText: true,
        decoration: const InputDecoration(
          labelText: 'Password',
          border: OutlineInputBorder(),
          prefixIcon: Icon(Icons.lock),
        ),
        validator: (value) =>
        value!.length < 6 ? 'Password must be at least 6 characters' : null,
      ),
      const SizedBox(height: 15),
```

```
                TextFormField(
                  controller: _phoneController,
                  decoration: const InputDecoration(
                    labelText: 'Phone Number',
                    border: OutlineInputBorder(),
                    prefixIcon: Icon(Icons.phone),
                  ),
                  validator: (value) =>
                  value!.isEmpty ? 'Please enter your phone number' : null,
                ),
                const SizedBox(height: 15),
                TextFormField(
                  controller: _addressController,
                  decoration: const InputDecoration(
                    labelText: 'Address',
                    border: OutlineInputBorder(),
                    prefixIcon: Icon(Icons.home),
                  ),
                  validator: (value) =>
                  value!.isEmpty ? 'Please enter your address' : null,
                ),
                const SizedBox(height: 20),
                ElevatedButton(
                  onPressed: registerUser, // Call registerUser function
                  style: ElevatedButton.styleFrom(backgroundColor: Colors.green),
                  child: const Text('REGISTER', style: TextStyle(color: Colors.white)),
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```


Register
DEBUG
Create Your Account
Full Name
Email
Phone Number
Password
Address
REGISTER
Already registered? Login


Register
Create Your Account
Full Name
Priya
Email
priya@gmail.com
Password
Phone Number
8765342768
Address
Goa
REGISTER


bigbasket
Food is on it's way
Login
FACEBOOK
GOOGLE
Email
Password
LOGIN
REGISTER HERE
Registration Successful!


bigbasket
Food is on it's way
Login
FACEBOOK
GOOGLE
rujuta@gmail.com
LOGIN
REGISTER HERE
Invalid credentials. Please try again.