

MAD & PWA Lab Journal

Experiment No.	06
Experiment Title.	To connect flutter UI with firebase database.
Roll No.	37
Name	Shravani Anil Patil
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 real-time, 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:

Signup.dart

```
import 'package:flutter/material.dart';
import 'home.dart'; // Import the HomePage file
import 'login.dart'; // Import LoginPage for navigation
import 'package:firebase_auth/firebase_auth.dart'; // Import Firebase Authentication

class SignupPage extends StatelessWidget {
  final TextEditingController nameController = TextEditingController();
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();

  void registerUser(BuildContext context) async {
    String name = nameController.text.trim();
    String email = emailController.text.trim();
    String password = passwordController.text.trim();
    if (name.isEmpty || email.isEmpty || password.isEmpty) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text("Please fill all fields")),
      );
      return;
    }
    try {
      // Create a new user with Firebase
      UserCredential userCredential = await FirebaseAuth.instance
        .createUserWithEmailAndPassword(email: email, password: password);
      // Optionally update the user's display name
      await userCredential.user?.updateDisplayName(name);
      // Navigate to HomePage after successful registration
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => HomePage()),
      );
    } catch (e) {
      // Handle errors and show an error message
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text("Registration failed: ${e.toString()}")),
      );
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Colors.white,
        elevation: 0,
        leading: IconButton(
          icon: Icon(Icons.arrow_back, color: Colors.black),
          onPressed: () {
            Navigator.pop(context);
          },
        ),
      ),
    );
  }
}
```

```

    ),
  ),
  body: Padding(
    padding: const EdgeInsets.symmetric(horizontal: 20),
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Image.asset('assets/logo.png', width: 120, height: 120),
        SizedBox(height: 20),
        Text(
          "Create an Account",
          style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold),
        ),
        SizedBox(height: 10),
        Text(
          "Sign up to get started",
          style: TextStyle(fontSize: 14, color: Colors.black54),
        ),
        SizedBox(height: 30),
        // Name Field
        TextField(
          controller: nameController,
          decoration: InputDecoration(
            labelText: "Full Name",
            border: OutlineInputBorder(),
          ),
        ),
        SizedBox(height: 15),
        // Email Field
        TextField(
          controller: emailController,
          decoration: InputDecoration(
            labelText: "Email",
            border: OutlineInputBorder(),
          ),
        ),
        SizedBox(height: 15),
        // Password Field
        TextField(
          controller: passwordController,
          obscureText: true,
          decoration: InputDecoration(
            labelText: "Password",
            border: OutlineInputBorder(),
          ),
        ),
        SizedBox(height: 20),
        // Signup Button
        ElevatedButton(
          onPressed: () {
            registerUser(context);
          },
          child: Text("Sign Up"),
        ),
      ],
    ),
  ),
),

```

```

style: ElevatedButton.styleFrom(
  backgroundColor: Colors.red,
  foregroundColor: Colors.white,
  minimumSize: Size(double.infinity, 50),
),
),
 SizedBox(height: 15),
// Login Redirect
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Text("Already have an account?"),
    TextButton(
      onPressed: () {
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(builder: (context) => LoginPage()),
        );
      },
      child: Text(
        "Login",
        style: TextStyle(color: Colors.red),
      ),
    ),
  ],
),
],
),
],
),
),
);
}
}

```

Login.dart

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'home.dart'; // Import the HomePage file
class LoginPage extends StatelessWidget {
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  final FirebaseAuth _auth = FirebaseAuth.instance;
  Future<void> _loginUser(BuildContext context) async {
    String email = emailController.text.trim();
    String password = passwordController.text.trim();
    if (email.isEmpty || password.isEmpty) {
      // Show error if fields are empty
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text("Please enter email and password")),
      );
      return;
    }
    try {
      // Attempt to sign in using email/password
      UserCredential userCredential = await _auth.signInWithEmailAndPassword(
        email: email,
        password: password,
      );
      // Check if the user is successfully authenticated
      if (userCredential.user != null) {
        // Navigate to HomePage if login is successful
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(builder: (context) => HomePage()),
        );
      }
    } catch (e) {
      // If error occurs (invalid credentials, network issue, etc.)
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text("Invalid email or password")),
      );
    }
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Colors.white,
        elevation: 0,
        leading: IconButton(
          icon: Icon(Icons.arrow_back, color: Colors.black),
          onPressed: () {
            Navigator.pop(context);
          }
        )
      )
    );
  }
}
```

```

    },
  ),
),
body: Padding(
  padding: const EdgeInsets.symmetric(horizontal: 20),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Image.asset('assets/logo.png', width: 120, height: 120),
      SizedBox(height: 20),
      Text(
        "Welcome Back!",
        style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold),
      ),
      SizedBox(height: 10),
      Text(
        "Login to continue",
        style: TextStyle(fontSize: 14, color: Colors.black54),
      ),
      SizedBox(height: 30),
      TextField(
        controller: emailController,
        decoration: InputDecoration(
          labelText: "Email",
          border: OutlineInputBorder(),
        ),
      ),
      SizedBox(height: 15),
      TextField(
        controller: passwordController,
        obscureText: true,
        decoration: InputDecoration(
          labelText: "Password",
          border: OutlineInputBorder(),
        ),
      ),
      SizedBox(height: 20),
      ElevatedButton(
        onPressed: () => _loginUser(context), // Firebase login
        child: Text("Login"),
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.red,
          foregroundColor: Colors.white,
          minimumSize: Size(double.infinity, 50),
        ),
      ),
      SizedBox(height: 15),
      TextButton(
        onPressed: () {
          // Navigate to forgot password page if you have one
        },
        child: Text(
          "Forgot Password?",

```

```

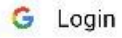
        style: TextStyle(color: Colors.red),
      ),
    ),
    SizedBox(height: 20),
    Text("Or login with"),
    SizedBox(height: 10),
    ElevatedButton.icon(
      onPressed: () {
        // Add Google login or other login methods here
      },
      icon: Image.asset('assets/Google.png', width: 24, height: 24),
      label: Text("Login with Google"),
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.white,
        foregroundColor: Colors.black,
        minimumSize: Size(double.infinity, 50),
        side: BorderSide(color: Colors.black26),
      ),
    ),
  ),
],
),
),
);
}
}

```


SKIP



Enjoy faster show booking through our
recommendations tailored for you.



I agree to the [Terms & Conditions](#) and [Privacy Policy](#)



Create an Account

Sign up to get started

Full Name

Shweta Patil

Email

sp@gmail.com

Password

.....

Sign Up

Already have an account? [Login](#)



book **my** show

Welcome Back!

Login to continue

Email

kp@gmail.com

Password

.....

Login

[Forgot Password?](#)

Or login with



Login with Google



