

MPL EXPERIMENT : 6
D15A-30
Jai navani
(with firebase connection)

New_login and signin_page:

Theory:

The SignInPage is a stateful widget that allows users to log in using their email and password. It interacts with Firebase Authentication to validate user credentials and redirects users to the home page upon successful login.

The NewLoginPage is a stateful widget that allows users to register using their email and password. It interacts with Firebase Authentication to create a new account.

Widgets & Properties (Home Page)

StatefulWidget (SignInPage)

- Used because the UI changes dynamically based on user interactions (e.g., login success or failure).

TextEditingController (for Input Fields)

- `_emailController`: Captures user input for the email field.
- `_passwordController`: Captures the password input.
- These controllers help retrieve user input when needed.

FirebaseAuth (`_auth`)

- `FirebaseAuth.instance` → Used to handle user authentication.
- `_auth.signInWithEmailAndPassword(email, password)` → Validates credentials with Firebase.

Scaffold (Main UI Structure)

- `AppBar` → Displays the title "Sign In" with a teal-colored background.
- `body` → Contains a `Column` that aligns widgets vertically.

Column (Main Layout)

- TextField (Email Input)
 - controller: `_emailController` → Captures user email.
 - decoration: `InputDecoration(labelText: "Email", border: OutlineInputBorder())` → Provides a label and a border.
- TextField (Password Input)
 - controller: `_passwordController` → Captures password.
 - obscureText: `true` → Hides password characters for security.
- ElevatedButton (Login Button)
 - onPressed: `_loginUser` → Calls `_loginUser()` function.
 - child: `Text("Login")` → Displays button text.

`_loginUser()` (Handles Login)

- `signInWithEmailAndPassword(email, password)`
 - Tries to log in the user using Firebase.
- If login is successful → Displays success message and redirects to the home page (`Navigator.pushReplacementNamed(context, '/home')`).
- If login fails → Shows an error message using `SnackBar`.

NEWLOGIN_PAGE:

FirebaseAuth (`_auth`)

- `FirebaseAuth.instance` → Used to register new users.
- `_auth.createUserWithEmailAndPassword(email, password)` → Creates a new account.

Column (Main Layout)

- TextField (Email Input)
 - controller: `_emailController` → Captures email input.
 - decoration: `InputDecoration(labelText: "Email")` → Provides a label.
- TextField (Password Input)
 - controller: `_passwordController` → Captures password input.
 - obscureText: `true` → Hides password characters.
- ElevatedButton (Register Button)
 - onPressed: `_registerUser` → Calls `_registerUser()` function.
 - child: `Text("Register")` → Displays button text.

`_registerUser()` (Handles Registration)

- createUserWithEmailAndPassword(email, password)
 - Creates a new Firebase account.

Code:

SIGIN_PAGE:

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';

class SigninPage extends StatefulWidget {
  @override
  _SigninPageState createState() => _SigninPageState();
}

class _SigninPageState extends State<SigninPage> {
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final FirebaseAuth _auth = FirebaseAuth.instance;

  Future<void> _loginUser() async {
    try {
      UserCredential userCredential = await _auth.signInWithEmailAndPassword(
        email: _emailController.text.trim(),
        password: _passwordController.text.trim(),
      );

      if (userCredential.user != null) {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text('Login successful! Redirecting...')),
        );

        // Delay before navigation (Fixes async navigation issue)
        Future.delayed(Duration(seconds: 1), () {
          Navigator.pushReplacementNamed(context, '/home');
        });
      }
    } on FirebaseAuthException catch (e) {
```

```

String errorMessage = "Login failed: ${e.message}";
ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text(errorMessage)));
}
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text("Sign In"), backgroundColor: Colors.teal),
    body: Padding(
      padding: EdgeInsets.all(16.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          TextField(
            controller: _emailController,
            decoration: InputDecoration(labelText: "Email", border: OutlineInputBorder()),
          ),
          SizedBox(height: 10),
          TextField(
            controller: _passwordController,
            decoration: InputDecoration(labelText: "Password", border: OutlineInputBorder()),
            obscureText: true,
          ),
          SizedBox(height: 20),
          ElevatedButton(
            onPressed: _loginUser, // Now it logs in instead of signing up
            child: Text("Login"),
          ),
        ],
      ),
    ),
  );
}
}

```

NEWLOGIN_PAGE:

```

import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';

```

```

class NewLoginPage extends StatefulWidget {
  @override
  _NewLoginPageState createState() => _NewLoginPageState();
}

class _NewLoginPageState extends State<NewLoginPage> {
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final FirebaseAuth _auth = FirebaseAuth.instance;

  Future<void> _registerUser() async {
    try {
      await _auth.createUserWithEmailAndPassword(
        email: _emailController.text.trim(),
        password: _passwordController.text.trim(),
      );
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('User registered successfully!')),
      );
    } catch (e) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Registration failed: $e')),
      );
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("New Login")),
      body: Padding(
        padding: EdgeInsets.all(16.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            TextField(
              controller: _emailController,
              decoration: InputDecoration(labelText: "Email"),
            ),
            TextField(
              controller: _passwordController,
              decoration: InputDecoration(labelText: "Password"),
              obscureText: true,
            ),
          ],
        ),
      ),
    );
  }
}

```

```

        SizedBox(height: 20),
        ElevatedButton(
          onPressed: _registerUser,
          child: Text("Register"),
        ),
      ],
    ),
  ),
);
}
}

```

ACCOUNT_PAGE:

```

import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'signin_page.dart'; // Importing SignInPage

class AccountPage extends StatelessWidget {
  final FirebaseAuth _auth = FirebaseAuth.instance;

  @override
  Widget build(BuildContext context) {
    // Get the currently logged-in user
    User? user = _auth.currentUser;

    return Scaffold(
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Center(
              child: CircleAvatar(
                radius: 50,
                backgroundColor: Colors.teal[100],
                child: Icon(Icons.person, size: 50, color: Colors.teal),
              ),
            ),
            SizedBox(height: 16),
            Text('Name: John Doe', style: TextStyle(fontSize: 18)),
            SizedBox(height: 8),
            Text(

```

```

      'Email: ${user?.email ?? "Not Logged In"}', // Dynamically show email
      style: TextStyle(fontSize: 18),
    ),
    SizedBox(height: 8),
    Text('Phone: +91 9999999999', style: TextStyle(fontSize: 18)),
    SizedBox(height: 16),
    Divider(),
    ListTile(
      leading: Icon(Icons.settings, color: Colors.teal),
      title: Text('Settings'),
      onTap: () {},
    ),
    ListTile(
      leading: Icon(Icons.help, color: Colors.teal),
      title: Text('Help & Support'),
      onTap: () {},
    ),
    Spacer(),
    Center(
      child: ElevatedButton(
        onPressed: () async {
          await _auth.signOut(); // Logout user
          Navigator.pushReplacement(
            context,
            MaterialPageRoute(builder: (context) => SigninPage()),
          );
        },
        child: Text('Logout', style: TextStyle(fontSize: 18)),
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.red,
          padding: EdgeInsets.symmetric(horizontal: 32, vertical: 12),
        ),
      ),
    ),
  ],
),
);
}
}

```

OUTPUT:

Newlogin_page:

6:57

New Login

Email
def@gmail.com

Password
.....

Register

Search by email address, phone number, or user UID

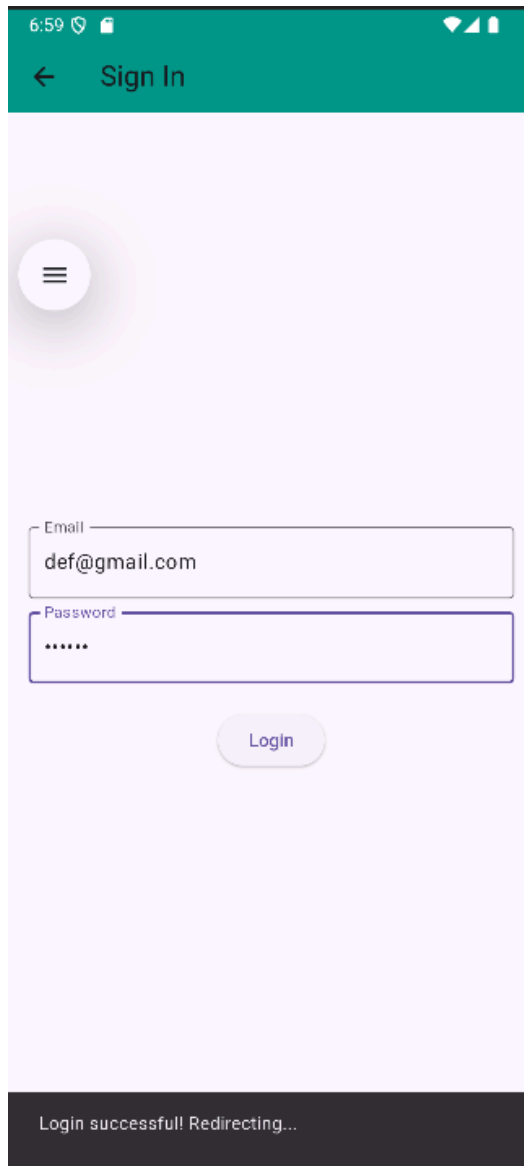
Add user

Identifier	Providers	Created ↓	Signed In	User UID
def@gmail.com		Mar 2, 2025	Mar 2, 2025	IMKSB0qlw4cyKXxeUweB2JP...
abc@gmail.com		Mar 2, 2025	Mar 2, 2025	DzDVpi5M7LcINQIS4uodONh...

Rows per page: 50 1 - 2 of 2

User registered


Now when tried to logged in:




Email we tried was def@gmail.com

Email: def@gmail.com

Phone: +91 9999999999

 Settings


 Help & Support


Logout

Then tried with other email:

Email: abc@gmail.com

Phone: +91 9999999999

 Settings

 Help & Support

Logout

