Name:Aarya Gandhi
Roll No. 14
Class: D15B

MPL Experiment 02

Aim:To design Flutter UI by including common widgets.

Theory:

Flutter UI Design Using Common Widgets

Flutter provides a comprehensive set of widgets to create visually appealing and interactive user interfaces. These widgets are categorized into **stateless** and **stateful** widgets.

- **Stateless Widgets** remain constant once built and do not update dynamically.
- **Stateful Widgets** can change based on user interaction or internal state changes.

Common Flutter Widgets

1. Basic Structural Widgets

These widgets provide the foundational layout structure of the app.

- **Scaffold** – Serves as the main layout structure with built-in support for `AppBar`, `Body`, and `FloatingActionButton`.
- **AppBar** – A top navigation bar that contains a title and optional action buttons.
- **Container** – A box-like widget used for styling, including background color, padding, and margins.
- **Column & Row** – Used to arrange child widgets in a vertical or horizontal direction, respectively.

2. User Input Widgets

These widgets allow users to provide input and interact with the application.

- **TextField** – Accepts user input in the form of text.
- **DropdownButton** – Displays a dropdown menu to select from multiple options.
- **Checkbox & Switch** – Used for enabling/disabling settings or selecting multiple options.

3. Display Widgets

These widgets are used to display text, images, and content.

- **Text** – Displays static or dynamic text content.
- **Image** – Loads images from assets, the network, or memory.
- **Card** – A material design component used to display content in an organized manner.

4. Interactive Widgets

These widgets handle user interactions like button clicks and gestures.

- **ElevatedButton** – A raised button used to perform actions on tap.
- **IconButton** – A button with an icon instead of text.
- **GestureDetector** – Detects gestures such as taps, swipes, and long presses.

5. Lists & Scrolling Widgets

These widgets help manage scrollable content.

- **ListView** – Displays a scrollable list of items.
- **GridView** – Creates a grid layout, ideal for galleries and product listings.
- **SingleChildScrollView** – Allows scrolling for a single child widget when content overflows.

6. Navigation Widgets

These widgets enable seamless navigation between different screens in the app.

- **Navigator** – Manages screen transitions using push and pop methods.
- **BottomNavigationBar** – A bottom menu that allows switching between different sections of the app.

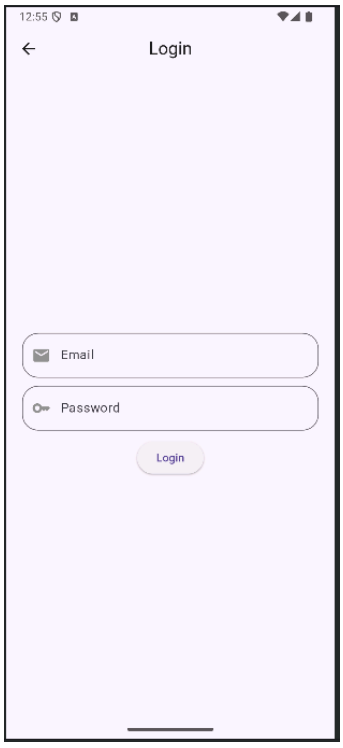
Implementation:

Login Page:

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

class SignupPage extends StatefulWidget {
  @override
  _SignupPageState createState() => _SignupPageState();
}

class _SignupPageState extends State<SignupPage> {
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  final FirebaseAuth _auth = FirebaseAuth.instance;

  void signUp() async {
    try {
      await _auth.createUserWithEmailAndPassword(
        email: emailController.text,
        password: passwordController.text,
      );
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text("Signup Successful!")),
      );
    } catch (e) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text("Error: ${e.toString()}")),
      );
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(" Sign Up Page",
          style: TextStyle(
            fontWeight: FontWeight.bold
```

```
  ),),
  centerTitle: true,
),
body: Center(
  child: Padding(
    padding: EdgeInsets.all(16),
    child: Column(
      mainAxisSize: MainAxisSize.min,
      children: [
        TextField(

          controller: emailController,
          decoration: InputDecoration(
            prefixIcon: Icon(Icons.person,color: Colors.grey,),
            labelText: "Enter Your Username",
            border: OutlineInputBorder(
              borderRadius: BorderRadius.circular(20),
            ),
          ),
        ),
        SizedBox(height: 16),
        TextField(
          controller: emailController,
          decoration: InputDecoration(
            prefixIcon: Icon(Icons.mail,color: Colors.grey,),
            labelText: "Enter Your Email",
            border: OutlineInputBorder(
              borderRadius: BorderRadius.circular(20),
            ),
          ),
        ),
        SizedBox(height: 16),
        TextField(
          controller: passwordController,
          obscureText: true,
          decoration: InputDecoration(
            prefixIcon: Icon(Icons.key,color: Colors.grey,),
            labelText: "Enter Your Password",
            border: OutlineInputBorder(
```

```
                borderRadius: BorderRadius.circular(20),
              ),
            ),
          ),
          SizedBox(height: 16),
          ElevatedButton(
            onPressed: signUp,
            child: Text("Sign Up"),
          ),
        ],
      ),
    ),
  ),
 );
 }
}
```



Signup Page:

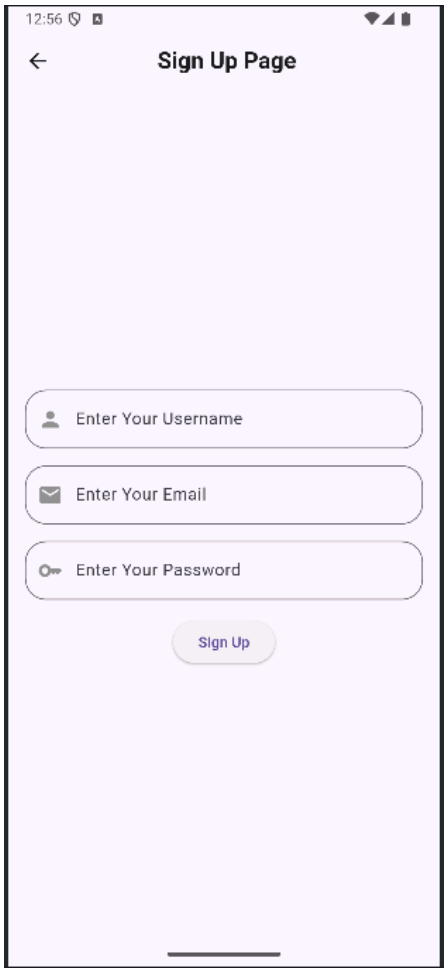
```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_auth/firebase_auth.dart';
```

```
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: LoginPage(),
    );
  }
}

class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  String errorMessage = '';

  Future<void> loginUser() async {
    try {
      await _auth.signInWithEmailAndPassword(
        email: emailController.text.trim(),
        password: passwordController.text.trim(),
      );
      // Navigate to home page after successful login
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => HomePage()),
      );
    } catch (error) {
```

```
    setState(() {
      errorMessage = error.toString();
    });
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Login"),
      centerTitle: true,
    ),
    body: Padding(
      padding: EdgeInsets.all(20.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          TextField(
            controller: emailController,
            decoration: InputDecoration(
              prefixIcon: Icon(Icons.mail,color: Colors.grey,),
              labelText: "Email",
              border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(20)
              ),
            ),
          ),
          SizedBox(height: 10),
          TextField(
            controller: passwordController,
            obscureText: true,
            decoration: InputDecoration(
              labelText: "Password",
              border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(20)
              ),
              prefixIcon: Icon(Icons.key,color: Colors.grey,)
            ),
```

```
            ),
            SizedBox(height: 10),
            ElevatedButton(
              onPressed: loginUser,
              child: Text("Login"),
            ),
            if (errorMessage.isNotEmpty)
              Text(
                errorMessage,
                style: TextStyle(color: Colors.red),
              ),
          ],
        ),
      ),
    );
  }
}

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

  @override
  Widget build(BuildContext context) {
    User? user = _auth.currentUser;

    return Scaffold(
      appBar: AppBar(
        title: Text("Home"),
        actions: [
          IconButton(
            icon: Icon(Icons.logout),
            onPressed: () async {
              await _auth.signOut();
              Navigator.pushReplacement(
                context,
                MaterialPageRoute(builder: (context) => LoginPage()),
              );
            },
          ),
        ],
```

```
      ),
      body: Center(
        child: Text(
          "Welcome, ${user?.email ?? 'User'}",
          style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
        ),
      ),
    );
  }
}
```



Conclusion

This app demonstrates Firebase authentication integration in Flutter with a well-structured UI. Using Flutter's Material Design widgets ensures a **responsive, smooth, and visually appealing** user experience.