Experiment 4

Aim: To create an interactive Form using a form widget.

Theory:A Form Widget is a key user interface component used to gather input from users. It plays a critical role in applications and websites, enabling tasks such as logging in, registering, providing personal details, or submitting feedback. Interactive forms go a step further by improving usability through real-time validation, responsive design, and dynamic user interactions.

Core Concepts of Interactive Forms:

1. Form Widgets
   Include input components such as: ○
   - Text fields ○ Radio buttons
   - Checkboxes ○ Dropdowns
   - Action buttons (e.g., Submit, Reset)
2. Validation
   Ensures that user input follows defined rules (e.g., required fields, proper email format, password length).
3. State Management
   Maintains and updates form values dynamically as the user interacts with different fields.
4. Event Handling
   Detects user actions like typing, selection, or clicks to trigger responses (e.g., showing error messages or submitting data).
5. UI/UX Considerations Forms should be:
   - Visually appealing
   - Easy to navigate ○ Mobile-friendly
   - Clear in providing user feedback (e.g., error hints or success messages)

Implementation Strategy

- Select a Platform/Framework: Choose a suitable environment (e.g., HTML, React, Flutter).
- Build the Form Layout: Use input widgets (text fields, dropdowns, etc.) to structure the form.
- Add Input Validation: Implement logic to check and ensure data correctness.
- Integrate Real-Time Feedback: Provide instant visual cues (e.g., red borders, success icons).
- Ensure Responsiveness: Design the form to adapt seamlessly across devices and screen sizes.

Github Link: https://github.com/WadhwaShweta4/MPL-LAB/blob/main/MPL_experiment_2.pdf

Code:

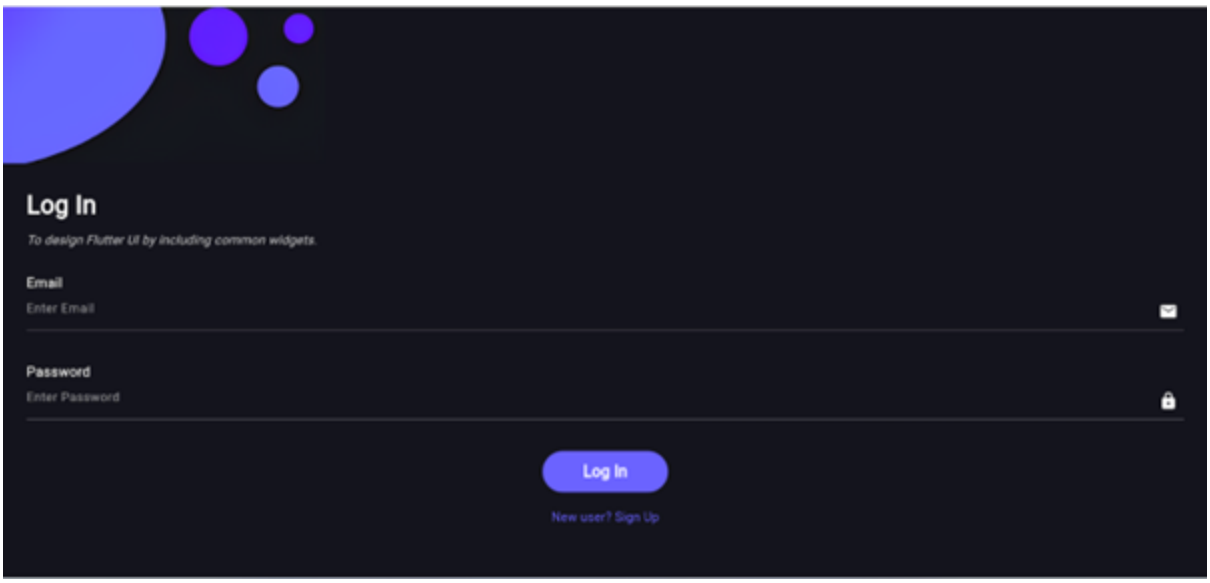
```
import 'package:flutter/material.dart'; class LoginPage extends

StatelessWidget {   final TextEditingController emailController =

TextEditingController();   final TextEditingController passwordController =

TextEditingController();



  @override

  Widget build(BuildContext context) {

    return Scaffold(       backgroundColor:

Colors.grey[100],       body: Padding(

padding: const EdgeInsets.all(24.0),
```

```
      child: Center(          child:
SingleChildScrollView(            child:
Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text("Sign in", style: TextStyle(fontSize: 28, fontWeight: FontWeight.bold)),
              SizedBox(height: 8),
Row(                  children: [
                Text("Don't have an account? "),
GestureDetector(
                  onTap: () {
                    Navigator.pushNamed(context, '/register');
                  },
                  child: Text("Register", style: TextStyle(color: Colors.indigo)),
                ),
              ],
            ),
            SizedBox(height: 24),
            TextField(
              controller: emailController,                decoration:
InputDecoration(labelText: 'Email'),
            ),
```

```
            SizedBox(height: 12),
TextField(
              controller: passwordController,              decoration:
InputDecoration(labelText: 'Password'),
              obscureText: true,
            ),
            SizedBox(height: 12),
            Align(
              alignment: Alignment.centerRight,              child: Text("Forgotten
password?", style: TextStyle(color: Colors.indigo)),
            ),
            SizedBox(height: 24),
ElevatedButton(              onPressed:
() {},
              style: ElevatedButton.styleFrom(              minimumSize: Size(double.infinity,
50),              backgroundColor: Colors.indigo,              shape:
RoundedRectangleBorder(borderRadius: BorderRadius.circular(25)),
              ),
              child: Text("Sign in"),
            ),
            SizedBox(height: 16),
Row(              children: [
```

```
                Expanded(child: Divider()),
Padding(
                  padding: const EdgeInsets.symmetric(horizontal: 8.0),
                  child: Text("or"),
                ),
                Expanded(child: Divider()),
              ],
            ),
            SizedBox(height: 16),
Center(
              child: Text("Sign in anonymously", style: TextStyle(color: Colors.indigo)),
            ),
          ],
        ),
      ),
    ),
  );
 }
}
```

Output:



-Register page can also be created in the similar way.

Conclusion: Interactive forms play a crucial role in modern applications by enhancing usability and data collection. Using form widgets with validation and event handling ensures a smooth user experience. Proper design and implementation of interactive forms lead to better engagement, accurate data submission, and improved overall functionality of an application.