MPL Experiment 4

Name: Anuprita Mhapankar **Class:** D15A **Roll no:**28

Aim: To create an interactive Form using form widget

Theory:

Creating an interactive form in Flutter requires using form-related widgets to collect and validate user input efficiently. The `Form` widget, combined with `TextFormField`, provides a structured way to manage input fields. Various input widgets like `TextField`, `DropdownButton`, `Checkbox`, `Radio`, and `Switch` allow users to enter data in different formats.

Validation and state management can be handled using the `GlobalKey<FormState>` to validate inputs before submission. Wrapping the form in a `SingleChildScrollView` ensures smooth scrolling when multiple fields are present.

A `RaisedButton` (deprecated) or `ElevatedButton` can trigger validation and submission logic. To enhance usability and create a responsive experience, proper padding, spacing, and `InputDecoration` should be applied.

Steps:

Step 1: Create a new Flutter project or open an existing one.

Step 2: Define a `Form` widget inside a StatefulWidget to manage user input.

Step 3: Use `TextFormField` for text input fields with validation logic.

Step 4: Include other input widgets such as `DropdownButton`, `Checkbox`, `Radio`, and `Switch` for additional user selections.

Step 5: Wrap the form inside a `SingleChildScrollView` to ensure smooth scrolling.

Step 6: Implement an `ElevatedButton` to trigger form validation and submission.

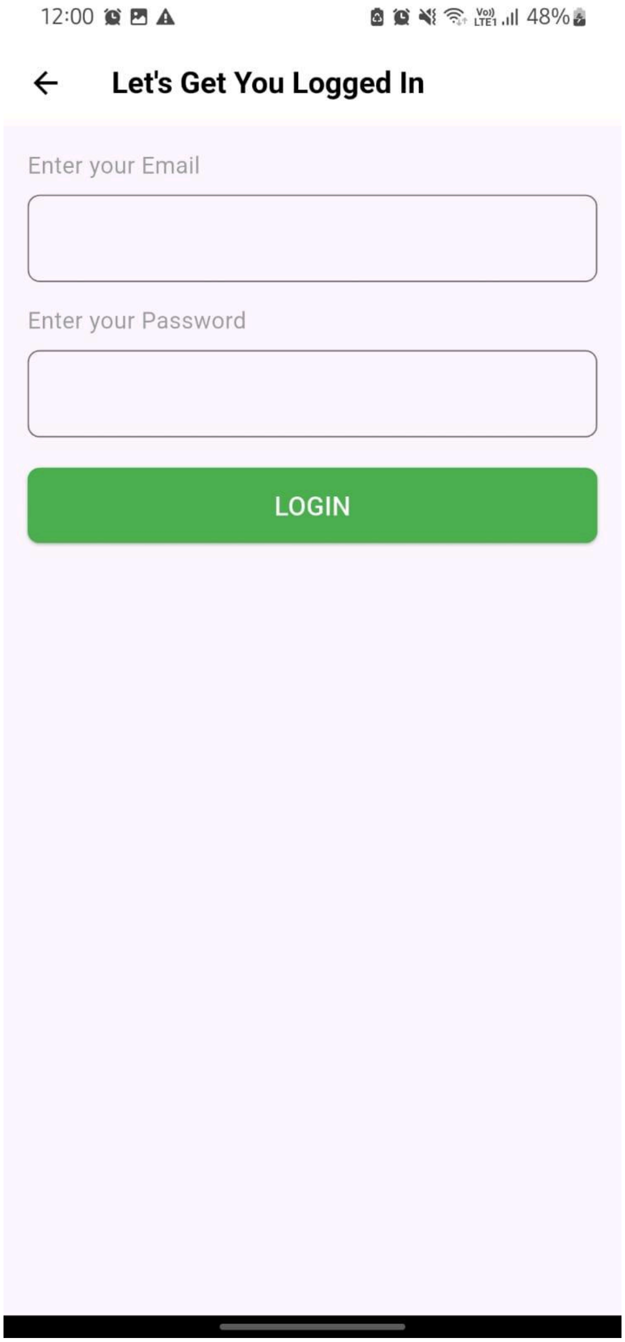
Step 7: Use `GlobalKey<FormState>` to manage form validation.

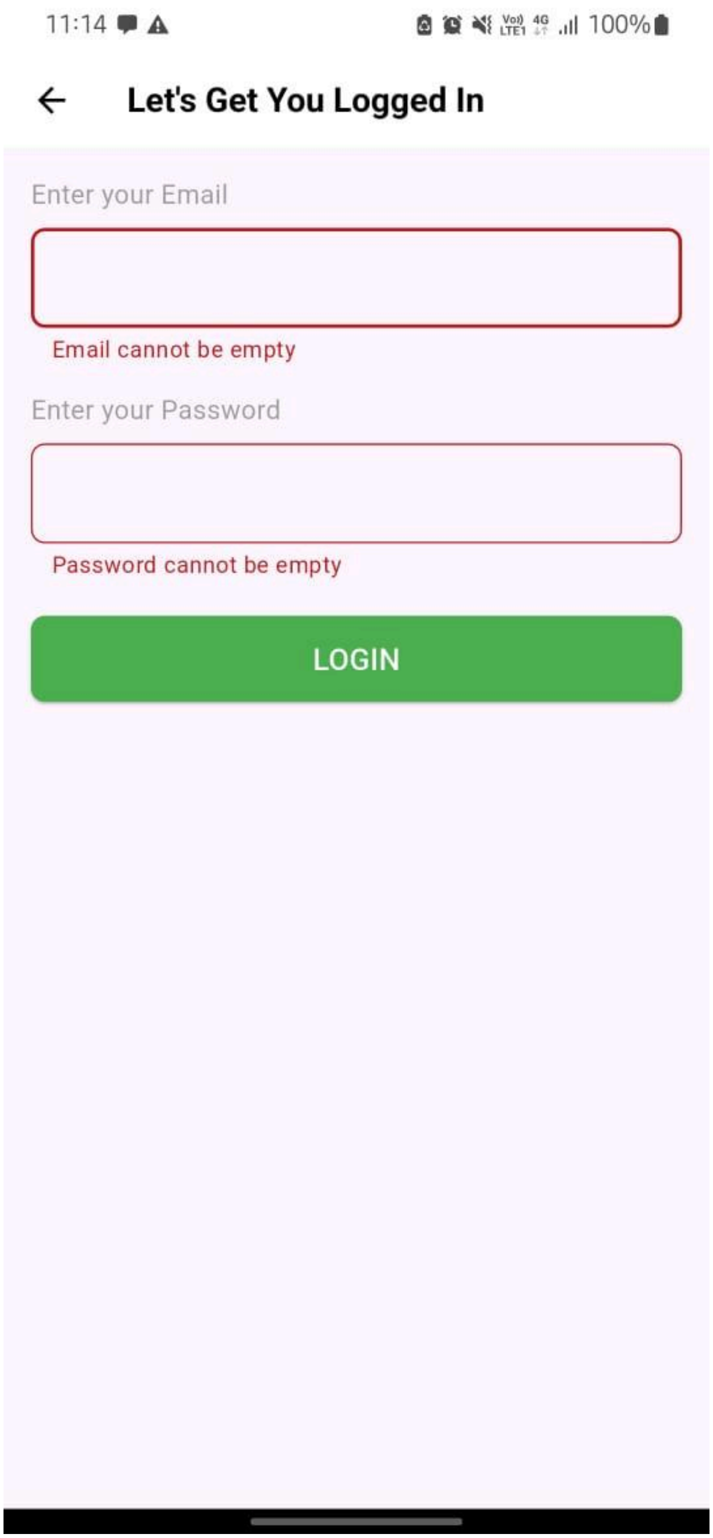
Code:

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:dmart/screens/home/home_page.dart';
import 'package:shared_preferences/shared_preferences.dart';
```

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

  @override
  State<LoginPage> createState() => _LoginPageState();
}

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

  final _formKey = GlobalKey<FormState>(); // Form Key for validation

  Future<void> _login() async {
    if (!_formKey.currentState!.validate()) {
      return; // Stop execution if validation fails
    }

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

      // Save login state
      SharedPreferences prefs = await SharedPreferences.getInstance();
      await prefs.setBool('isLoggedIn', true);

      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => const HomePage()),
      );
    } catch (e) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Login failed: ${e.toString()}')),
      );
    }
  }

  String? _validateEmail(String? value) {
    if (value == null || value.isEmpty) {
      return "Email cannot be empty";
    }
    String emailPattern =

r'^[a-zA-Z0-9.a-zA-Z0-9.!#$%&’*+/=?^_`{|}~-]+@[a-zA-Z0-9]+\.[a-zA-Z]+'
;
    RegExp regex = RegExp(emailPattern);
    if (!regex.hasMatch(value)) {
      return "Enter a valid email";
    }
    return null;
  }

  String? _validatePassword(String? value) {
    if (value == null || value.isEmpty) {
      return "Password cannot be empty";
    }
    if (value.length < 6) {
      return "Password must be at least 6 characters";
    }
    return null;
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        leading: IconButton(
          icon: const Icon(Icons.arrow_back, color: Colors.black),
          onPressed: () {
            Navigator.pop(context);
          },
        ),
        backgroundColor: Colors.white,
        elevation: 0,
        title: const Text(
          "Let's Get You Logged In",
          style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold,
color: Colors.black),
        ),
        centerTitle: false,
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Form(
          key: _formKey, // Wrap the form for validation
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              const Text(
                "Enter your Email",
                style: TextStyle(fontSize: 14, color: Colors.grey),
              ),
              const SizedBox(height: 8),
              TextFormField(
                controller: _emailController,
                decoration: InputDecoration(
                  border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(8),
                    borderSide: const BorderSide(color: Colors.grey),
                  ),
                  focusedBorder: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(8),
                    borderSide: const BorderSide(color: Colors.blue),
                  ),
                ),
                keyboardType: TextInputType.emailAddress,
                validator: _validateEmail,
              ),
              const SizedBox(height: 15),
              const Text(
                "Enter your Password",
                style: TextStyle(fontSize: 14, color: Colors.grey),
              ),
              const SizedBox(height: 8),
              TextFormField(
                controller: _passwordController,
                obscureText: true,
                decoration: InputDecoration(
                  border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(8),
                    borderSide: const BorderSide(color: Colors.grey),
                  ),
                  focusedBorder: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(8),
                    borderSide: const BorderSide(color: Colors.blue),
                  ),
                ),
                validator: _validatePassword,
              ),
              const SizedBox(height: 20),
              SizedBox(
                width: double.infinity,
                height: 50, // Match button height in image
                child: ElevatedButton(
                  onPressed: _login,
                  style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.green, // Match button
color
                    foregroundColor: Colors.white,
                    shape: RoundedRectangleBorder(
                      borderRadius: BorderRadius.circular(8), //
Rectangular button
                    ),
                  ),
                  child: const Text("LOGIN", style:
TextStyle(fontSize: 16)),
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

Output:
Login Page





Let's Get You Logged In

Enter your Email

Email cannot be empty

Enter your Password

Password cannot be empty