

MPL Experiment 4

Name: Eesha Chavan

Class: D15A

Roll no: 8

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:

register.dart:

```
import 'package:flutter/material.dart';
import 'login.dart'; // Importing login page for navigation

class RegisterPage extends StatefulWidget {
```

```

@override
_RegisterPageState createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {
  DateTime? selectedDate;
  String? selectedGender;

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

  // Controllers for text fields
  final TextEditingController _firstNameController =
TextEditingController();
  final TextEditingController _lastNameController =
TextEditingController();
  final TextEditingController _phoneController = TextEditingController();
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController =
TextEditingController();

  @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.pushReplacement(
            context,
            MaterialPageRoute(builder: (context) => LoginPage()),
          ),
        ),
    ),
  ),

```

```

    ),
    body: SingleChildScrollView(
      padding: EdgeInsets.symmetric(horizontal: 20),
      child: Form(
        key: _formKey, // Assign the form key
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Center(
              child: Column(
                children: [
                  Text(
                    "Looks like you are new",
                    style: TextStyle(fontSize: 18, fontWeight:
FontWeight.bold, color: Colors.red),
                  ),
                  SizedBox(height: 5),
                  Text(
                    "Get 10% additional cashback as TSS points on every
purchase.",
                    textAlign: TextAlign.center,
                    style: TextStyle(fontSize: 14, color: Colors.grey),
                  ),
                ],
              ),
            ),
            SizedBox(height: 20),

            // First Name & Last Name
            Row(
              children: [
                Expanded(
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [

```

```
Text("First Name", style: TextStyle(fontSize: 14,  
color: Colors.black)),  
  
    TextFormField(  
      controller: _firstNameController,  
      decoration: InputDecoration(  
        hintText: "First Name *",  
        hintStyle: TextStyle(color: Colors.grey),  
        border: OutlineInputBorder(),  
      ),  
      validator: (value) {  
        if (value == null || value.isEmpty) {  
          return 'First Name is required';  
        }  
        return null;  
      },  
    ),  
  ],  
),  
),  
SizedBox(width: 10),  
Expanded(  
  child: Column(  
    crossAxisAlignment: CrossAxisAlignment.start,  
    children: [  
      Text("Last Name", style: TextStyle(fontSize: 14,  
color: Colors.black)),  
  
      TextFormField(  
        controller: _lastNameController,  
        decoration: InputDecoration(  
          hintText: "Last Name *",  
          hintStyle: TextStyle(color: Colors.grey),  
          border: OutlineInputBorder(),  
        ),  
        validator: (value) {  
          if (value == null || value.isEmpty) {
```

[illegible]

```

        TextFormField(
          controller: _emailController,
          keyboardType: TextInputType.emailAddress,
          decoration: InputDecoration(
            hintText: "Email ID *",
            hintStyle: TextStyle(color: Colors.grey),
            border: OutlineInputBorder(),
          ),
          validator: (value) {
            if (value == null || value.isEmpty) {
              return 'Email is required';
            } else if
(!RegExp(r'^[\w-\.] + @ ([\w- ]+ \. )+ [\w- ] {2,4} $').hasMatch(value)) {
              return 'Enter a valid email address';
            }
            return null;
          },
        ),
        SizedBox(height: 15),

        // Password
        Text("Password", style: TextStyle(fontSize: 14, color:
Colors.black)),
        TextFormField(
          controller: _passwordController,
          obscureText: true,
          decoration: InputDecoration(
            hintText: "Password *",
            hintStyle: TextStyle(color: Colors.grey),
            border: OutlineInputBorder(),
            suffixIcon: Icon(Icons.visibility_off, color:
Colors.grey),
          ),
          validator: (value) {
            if (value == null || value.isEmpty) {

```

```

        return 'Password is required';
    }
    return null;
},
),
 SizedBox(height: 15),

 // Birthdate
 Text("Birthdate", style: TextStyle(fontSize: 14, color:
Colors.black)),
 TextFormField(
  readOnly: true,
  decoration: InputDecoration(
    hintText: "DD-MM-YYYY *",
    hintStyle: TextStyle(color: Colors.grey),
    border: OutlineInputBorder(),
    suffixIcon: Icon(Icons.calendar_today, color:
Colors.grey),
  ),
  onTap: () async {
    DateTime? pickedDate = await showDatePicker(
      context: context,
      initialDate: DateTime.now(),
      firstDate: DateTime(1900),
      lastDate: DateTime.now(),
      builder: (context, child) {
        return Theme(
          data: ThemeData.light().copyWith(
            primaryColor: Colors.grey,
            colorScheme: ColorScheme.light(primary:
Colors.grey),
            buttonTheme: ButtonThemeData(textTheme:
ButtonTextTheme.primary),
          ),
          child: child!,
        );
      }
    );
  }
);

```

```

        },
      );

      if (pickedDate != null) {
        setState(() {
          selectedDate = pickedDate;
        });
      }
    },
    validator: (value) {
      if (selectedDate == null) {
        return 'Birthdate is required';
      }
      return null;
    },
  ),
  SizedBox(height: 5),

  // Birthday Discount Note
  Text(
    "Avail 10% birthday discount as a member",
    style: TextStyle(fontSize: 12, color: Colors.grey),
  ),
  SizedBox(height: 20),

  // Gender Selection
  Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      _genderOption("Male"),
      _genderOption("Female"),
      _genderOption("Other"),
    ],
  ),
  SizedBox(height: 20),

```



```

        // Register Button
        SizedBox(
          width: double.infinity,
          child: ElevatedButton(
            style: ElevatedButton.styleFrom(
              backgroundColor: Color(0xFF31827C), // New Darker Teal
Color
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(8),
              ),
            ),
            onPressed: () {
              // Validate the form
              if (_formKey.currentState!.validate()) {
                // If the form is valid, proceed with registration
                print("Registration Successful");
                // Add your registration logic here
              }
            },
            child: Padding(
              padding: EdgeInsets.symmetric(vertical: 14),
              child: Text("REGISTER", style: TextStyle(fontSize: 16,
color: Colors.white)),
            ),
          ),
        ),
        SizedBox(height: 10),

        // Already Registered? LOG IN
        Center(
          child: GestureDetector(
            onTap: () {
              Navigator.pushReplacement(
                context,


```

[illegible]

```
        activeColor: Color(0xFF31827C), // New Teal Color for Selected
Option
        value: text,
        groupValue: selectedGender,
        onChanged: (value) {
            setState(() {
                selectedGender = value.toString();
            });
        },
    ),
    Text(text, style: TextStyle(fontSize: 14)),
],
);
}
}
```

Register Page:

01:25

 100%

←

First Name

First Name *

First Name is required

Last Name

Last Name *

Last Name is required

9986548649469

Phone number must be 10 digits


Email ID

eesha

Enter a valid email address

Password


Password *



Password is required

Birthdate

DD-MM-YYYY *



Birthdate is required

Avail 10% birthday discount as a member

☐ Male

☐ Female

☐ Other

REGISTER

Already Registered? [LOG IN](#)