Experiment 4

Aim: To create an interactive Form using form widget

Theory:
The Form widget in Flutter is a fundamental widget for building forms. It provides a way to group multiple form fields together, perform validation on those fields, and manage their state.

State Management: Flutter provides various ways to manage the state of interactive forms. You can use StatefulWidget or state management solutions like Provider, Riverpod, or Bloc pattern to handle form data changes efficiently. The choice depends on the complexity and requirements of your application.

Form Widgets: Flutter offers a range of form widgets such as TextField, Checkbox, Radio, DropdownButton, etc., which are used to collect input from users. These widgets can be customized with various parameters to fit the design and functionality of your interactive form.

Validation: Validating user input is essential for ensuring data integrity. Flutter provides a built-in mechanism for form validation using the Form widget along with TextFormField widgets. You can define validation logic for each form field and display error messages accordingly.

Input Handling: Flutter offers various options for handling user input, including keyboard input, gestures, and voice input. You can use event listeners like onChanged, onSubmitted, and onTap to capture user input and update the form state accordingly.

Theming and Styling: Flutter's flexible styling and theming system allow you to customize the appearance of form elements to match your app's design language. You can use themes, colors, fonts, and custom widgets to create visually appealing and consistent interactive forms.

Some Properties of Form Widget

- key: A GlobalKey that uniquely identifies the Form. You can use this key to interact with the form, such as validating, resetting, or saving its state.
- child: The child widget that contains the form fields. Typically, this is a Column, ListView, or another widget that allows you to arrange the form fields vertically.
- autovalidateMode: An enum that specifies when the form should automatically validate its fields.

Code in main.dart:

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

void main() {
 runApp(MyApp());
}

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

class GoogleLoginPage extends StatefulWidget {
 @override
 _GoogleLoginPageState createState() => _GoogleLoginPageState();
}

class _GoogleLoginPageState extends State<GoogleLoginPage> {
 final GlobalKey<FormState> _formKey = GlobalKey<FormState>();

 @override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(
       title: Text('Google Login'),
     ),
     body: Padding(
       padding: const EdgeInsets.all(16.0),
       child: Form(
         key: _formKey,
         child: Column(
           mainAxisAlignment: MainAxisAlignment.center,
           children: [
             Image.asset(
               'assets/google_logo.png', // Replace with your Google logo asset
               height: 100,
             ),
             SizedBox(height: 20),
             Container(
               padding: EdgeInsets.symmetric(horizontal: 16, vertical: 8),
```

```
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(20),
    color: Colors.grey[300],
  ),
  child: Text(
    'example@gmail.com', // Dummy email address
    style: TextStyle(fontSize: 18),
  ),
),
SizedBox(height: 20),
TextFormField(
  decoration: InputDecoration(
    labelText: 'Password',
    border: OutlineInputBorder(),
  ),
  obscureText: true,
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Password is required';
    }
    return null;
  },
),
SizedBox(height: 20),
Row(
  children: [
    TextButton(
      onPressed: () {
        // Handle forgot password action
        print('Forgot password');
      },
      child: Text(
        'Forgot password?',
        style: TextStyle(color: Colors.blue),
      ),
    ),
  ],
),
SizedBox(height: 20),
ElevatedButton(
  onPressed: () {
    if (_formKey.currentState!.validate()) {
      // Handle form submission
      print('Next');
```

```
                  }
                },
                child: Text('Next'),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

Output: