Name: Atif Ansari Roll no: 04 Class: D15B

MPL Experiment - 4

* Aim: To create an Interactive form using Form Widget.

* Theory: Forms are essential for user interactions in mobile applications, allowing users to enter and submit information. In Flutter, the Form widget provides a structured way to handle user input with validation. The key concepts include:

1. GlobalKey <formstate> : A global key to manage form validation and submission.
2. TextFormField : A field where users input text, with built-in validation.
3. DropdownButtonFormField : A dropdown menu for selecting options.
4. Checkbox : A selection box for Boolean values.
5. ElevatedButton : A button to validate and submit the form.

The form validates user input, displays error messages for invalid input, and processes data upon successful submission.

* Conclusion: This experiment successfully demonstrates how to create and validate a form in Flutter. The form accepts user input, ensures correct data entry through validation, and processes the data when submitted.

1. Interactive Form

Code:

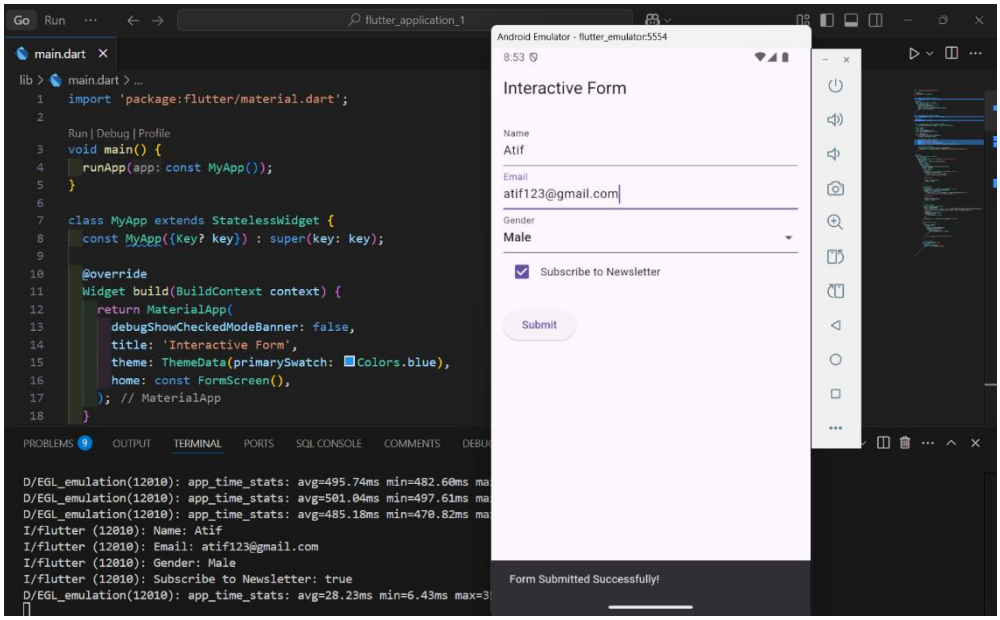
```
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
 const MyApp({Key? key}) : super(key: key);

 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   debugShowCheckedModeBanner: false,
   title: 'Interactive Form',
   theme: ThemeData(primarySwatch: Colors.blue),
   home: const FormScreen(),
  );
 }
}

class FormScreen extends StatefulWidget {
 const FormScreen({Key? key}) : super(key: key);

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

class _FormScreenState extends State<FormScreen> {
 final _formKey = GlobalKey<FormState>();

 String _name = '';
 String _email = '';
 String _selectedGender = 'Male';
 bool _subscribeToNewsletter = false;

 void _submitForm() {
  if (_formKey.currentState!.validate()) {
   _formKey.currentState!.save();

   // Display user input in console
   print('Name: $_name');
   print('Email: $_email');
   print('Gender: $_selectedGender');
```

```
    print('Subscribe to Newsletter: $_subscribeToNewsletter');

    // Show success message
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('Form Submitted Successfully!')),
    );
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: const Text('Interactive Form')),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Form(
        key: _formKey,
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            // Name Input
            TextFormField(
              decoration: const InputDecoration(labelText: 'Name'),
              validator: (value) {
                if (value == null || value.isEmpty) {
                  return 'Please enter your name';
                }
                return null;
              },
              onSaved: (value) {
                _name = value!;
              },
            ),

            // Email Input
            TextFormField(
              decoration: const InputDecoration(labelText: 'Email'),
              keyboardType: TextInputType.emailAddress,
              validator: (value) {
                if (value == null || value.isEmpty || !value.contains('@')) {
                  return 'Please enter a valid email address';
                }
                return null;
              },
              onSaved: (value) {
```

```
            _email = value!;
          },
        ),

        // Gender Dropdown
        DropdownButtonFormField<String>(
          value: _selectedGender,
          items: ['Male', 'Female', 'Other'].map((gender) {
            return DropdownMenuItem(value: gender, child: Text(gender));
          }).toList(),
          onChanged: (value) {
            setState(() {
              _selectedGender = value!;
            });
          },
          decoration: const InputDecoration(labelText: 'Gender'),
        ),

        // Newsletter Checkbox
        Row(
          children: [
            Checkbox(
              value: _subscribeToNewsletter,
              onChanged: (value) {
                setState(() {
                  _subscribeToNewsletter = value!;
                });
              },
            ),
            const Text('Subscribe to Newsletter'),
          ],
        ),

        // Submit Button
        const SizedBox(height: 20),
        ElevatedButton(
          onPressed: _submitForm,
          child: const Text('Submit'),
        ),
      ],
    ),
  ),
 );
}
```

}

Output:



2. Login Form with Email and Password Fields

Code:

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

void main() => runApp(const MyApp());

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  static const String _title = 'Sample App';

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: _title,
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          elevation: 0,
          backgroundColor: Colors.blue[300],
          centerTitle: true,
          title: const Text(
            'Login',
            style: TextStyle(
              color: Colors.white,
              fontFamily: 'Roboto',
              fontSize: 24,
```

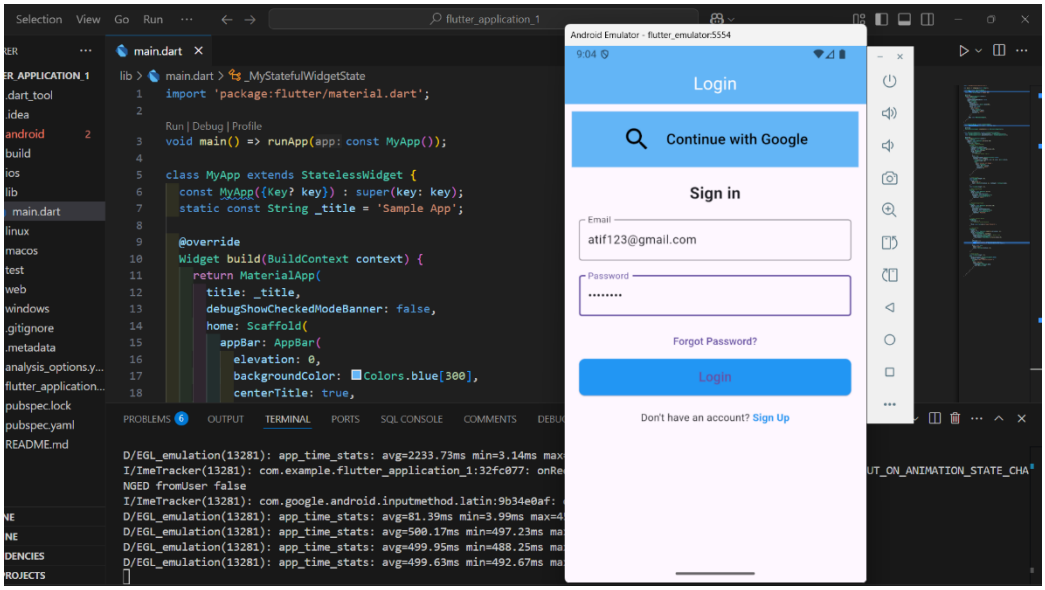
```
          ),
        ),
      ),
      body: const MyStatefulWidget(),
    ),
  );
 }
}

class MyStatefulWidget extends StatefulWidget {
 const MyStatefulWidget({Key? key}) : super(key: key);

 @override
 State<MyStatefulWidget> createState() => _MyStatefulWidgetState();
}

class _MyStatefulWidgetState extends State<MyStatefulWidget> {
 final TextEditingController emailController = TextEditingController();
 final TextEditingController passwordController = TextEditingController();

 @override
 Widget build(BuildContext context) {
  return Padding(
   padding: const EdgeInsets.all(10),
   child: ListView(
    children: <Widget>[
     // Google Sign-In Button
     Container(
      width: double.infinity,
      alignment: Alignment.center,
      padding: const EdgeInsets.all(10),
      color: Colors.blue[300],
      child: TextButton(
       onPressed: () {
        // Implement Google Sign-in
       },
       child: Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: const [
         Icon(Icons.search, size: 40, color: Colors.black),
         SizedBox(width: 20),
         Text(
          'Continue with Google',
          style: TextStyle(
           color: Colors.black,
```

```
          fontFamily: 'Roboto',
          fontSize: 20,
        ),
      ),
    ],
   ),
  ),
),

// Sign-in Text
const SizedBox(height: 20),
const Center(
  child: Text(
    'Sign in',
    style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold),
  ),
),
const SizedBox(height: 10),

// Email Input Field
Padding(
  padding: const EdgeInsets.all(10),
  child: TextField(
    controller: emailController,
    decoration: const InputDecoration(
      border: OutlineInputBorder(),
      labelText: 'Email',
    ),
  ),
),

// Password Input Field
Padding(
  padding: const EdgeInsets.all(10),
  child: TextField(
    obscureText: true,
    controller: passwordController,
    decoration: const InputDecoration(
      border: OutlineInputBorder(),
      labelText: 'Password',
    ),
  ),
),

// Forgot Password Button
```

```
TextButton(
  onPressed: () {
    // Implement Forgot Password
  },
  child: const Text('Forgot Password?'),
),

// Login Button
Container(
  height: 50,
  padding: const EdgeInsets.symmetric(horizontal: 10),
  child: ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.blue, // Button color
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(8),
      ),
    ),
    onPressed: () {
      print('Email: ${emailController.text}');
      print('Password: ${passwordController.text}');
    },
    child: const Text(
      'Login',
      style: TextStyle(fontSize: 18),
    ),
  ),
),

// Sign Up Text
const SizedBox(height: 20),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: const [
    Text("Don't have an account? "),
    Text(
      'Sign Up',
      style: TextStyle(
        fontWeight: FontWeight.bold,
        color: Colors.blue,
      ),
    ),
  ],
),
],
```

```
    ),
   );
  }
}
```

Output:



3. Form Validation

Code:

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

void main() => runApp(const MyApp());

class MyApp extends StatelessWidget {
 const MyApp({Key? key}) : super(key: key);

 @override
 Widget build(BuildContext context) {
  const appTitle = 'Form Validation Demo';
  return MaterialApp(
   title: appTitle,
   debugShowCheckedModeBanner: false,
   home: Scaffold(
    appBar: AppBar(
     title: const Text(appTitle),
    ),
    body: const Padding(
     padding: EdgeInsets.all(16.0),
     child: MyCustomForm(),
    ),
```

```
    ),
   );
  }
}

// Create a Form widget.
class MyCustomForm extends StatefulWidget {
  const MyCustomForm({Key? key}) : super(key: key);

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

// Create a corresponding State class.
// This class holds data related to the form.
class MyCustomFormState extends State<MyCustomForm> {
  // Create a global key that uniquely identifies the Form widget
  // and allows validation of the form.
  final _formKey = GlobalKey<FormState>();
  final TextEditingController textController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return SingleChildScrollView(
      child: Form(
        key: _formKey,
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            // Input Field
            TextFormField(
              controller: textController,
              decoration: const InputDecoration(
                border: OutlineInputBorder(),
                labelText: 'Enter text',
              ),
              // The validator receives the text that the user has entered.
              validator: (value) {
                if (value == null || value.isEmpty) {
                  return 'Please enter some text';
                }
                return null;
              },
            ),
```

```
        const SizedBox(height: 20),

        // Submit Button
        Center(
          child: ElevatedButton(
            onPressed: () {
              // Validate returns true if the form is valid, or false otherwise.
              if (_formKey.currentState!.validate()) {
                // If the form is valid, display a snackbar.
                ScaffoldMessenger.of(context).showSnackBar(
                  const SnackBar(content: Text('Processing Data')),
                );
              }
            },
            child: const Text('Submit'),
          ),
        ),
      ],
    ),
  ),
 );
 }
}
```

Output: