Name :- Raghav Mundhara
Class:-D15B Roll No:- 41

Experiment No :- 4

AIM :- To create an interactive Form using form widget

THEORY :-
n Flutter involves providing real-time feedback to users as they input data into form fields. This feedback typically includes informing users whether the data they've entered is valid or invalid, and possibly providing additional information on how to correct any errors.

A general approach to implementing interactive form validation in Flutter:

- **Text Input Fields:** Use text input widgets like TextFormField or TextField to capture user input. These widgets provide properties like validator and autovalidateMode which are essential for form validation.

- **Validator Function:** Implement a validator function that checks the validity of the input data. This function should be passed to the validator property of the text input field. It returns a string error message if the input is invalid, or null if the input is valid.

- **Real-time Feedback:** Use the autovalidateMode property of the form widget to enable real-time feedback. This property determines when the validator function is called. You can set it to AutovalidateMode.always to trigger validation on every change, or AutovalidateMode.onUserInteraction to trigger validation only when the user interacts with the field.

- **Displaying Error Messages:** Display error messages to users when input data is invalid. You can use the errorText property of the text input field to show error messages directly below the field.

- **Submit Button:** Disable the submit button when the form is invalid to prevent users from submitting incomplete or incorrect data.

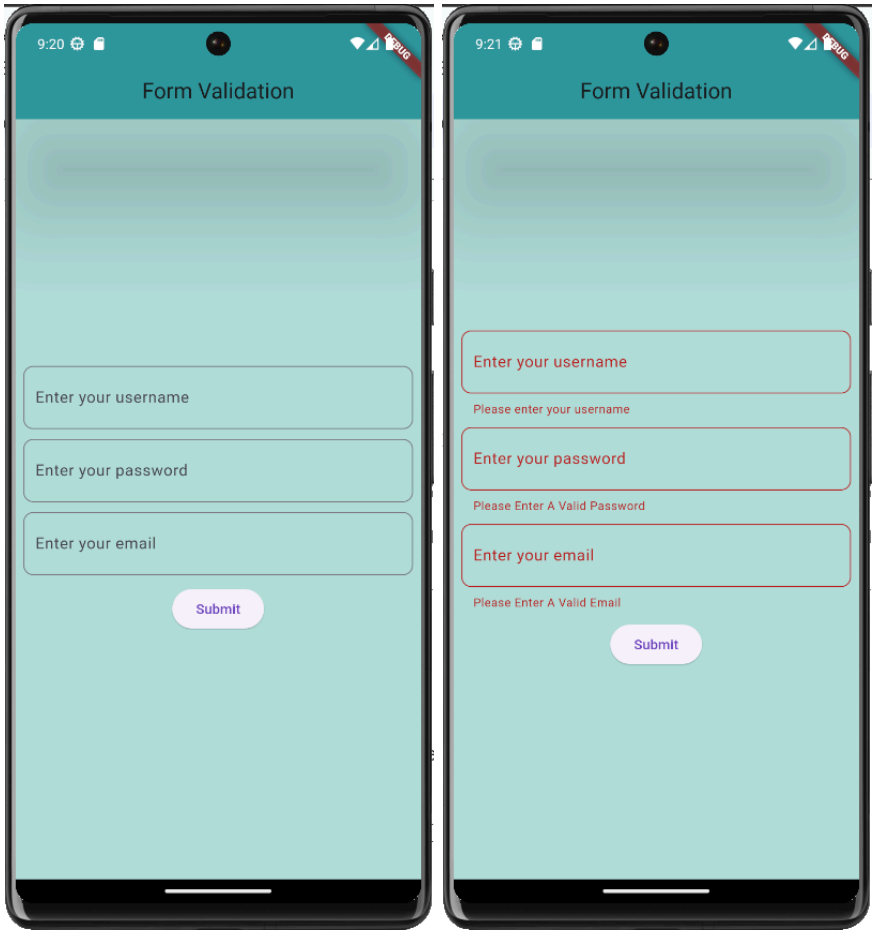
- **Handling Form Submission:** When the user submits the form, validate all form fields again to ensure data integrity. If any field is invalid, prevent form submission and display error messages to prompt the user to correct the errors.

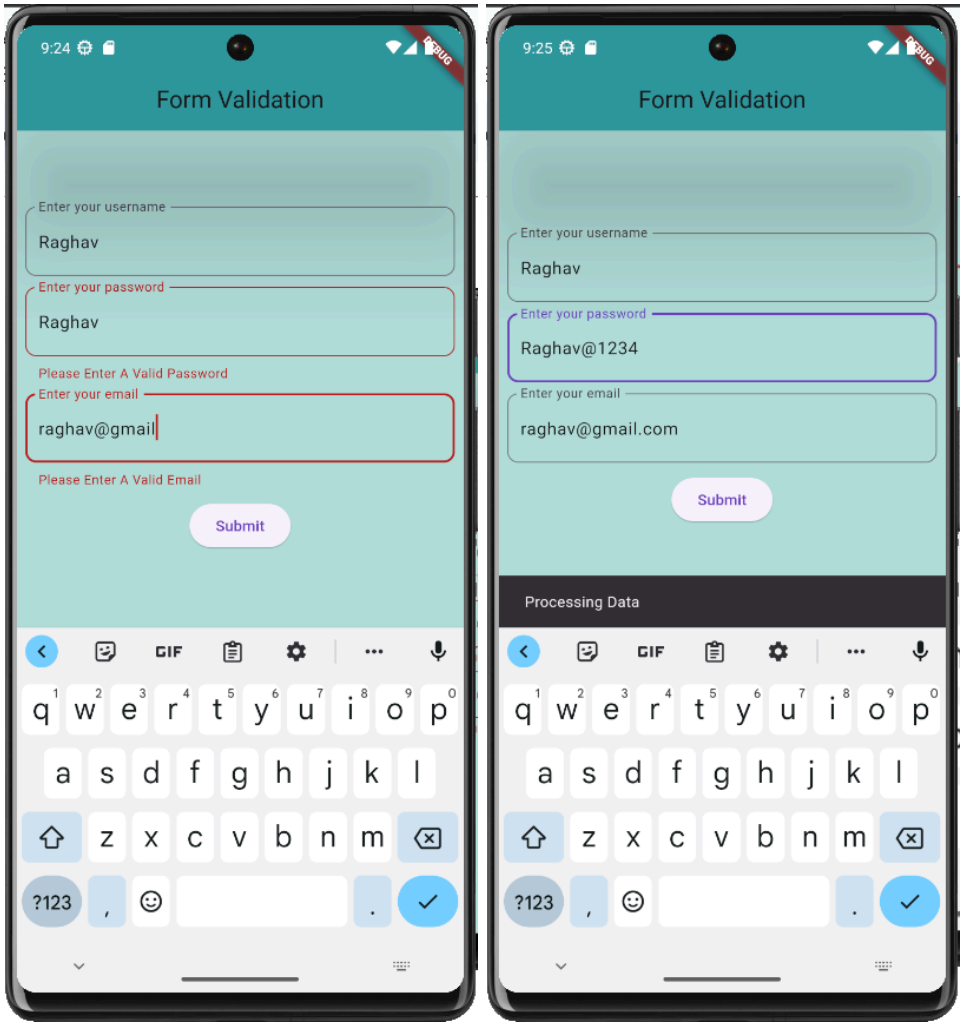
CODE & OUTPUT :-

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

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

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key});

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  final _formKey = GlobalKey<FormState>();
  RegExp passwordRegex =
RegExp(r'^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$');
  RegExp emailRegex = RegExp(r'^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$');
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.teal[100],
      appBar: AppBar(
        title: const Text('Form Validation'),
        centerTitle: true,
        backgroundColor: Colors.teal[400],
        elevation: 100,
        shadowColor: Colors.black,
```

```
),
body: Padding(
  padding: const EdgeInsets.all(8.0),
  child: Center(
      child: Form(
    key: _formKey,
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        TextFormField(
          decoration: InputDecoration(
              labelText: 'Enter your username',
              border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(10.0))),
          validator: (value) {
            if (value!.isEmpty) {
              return 'Please enter your username';
            }
            return null;
          },
        ),
        const SizedBox(height: 10),
        TextFormField(
          decoration: InputDecoration(
              labelText: 'Enter your password',
              border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(10.0))),
          validator: (value) {
            if (!passwordRegex.hasMatch(value!)) {
              return 'Please Enter A Valid Password';
            }
            return null;
          },
        ),
        const SizedBox(height: 10),
        TextFormField(
          decoration: InputDecoration(
              labelText: 'Enter your email',
              border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(10.0))),
          validator: (value) {
            if (!emailRegex.hasMatch(value!)) {
              return 'Please Enter A Valid Email';
            }
```

```
              return null;
            },
          ),
          const SizedBox(height: 10),
          ElevatedButton(
            onPressed: () {
              if (_formKey.currentState!.validate()) {
                ScaffoldMessenger.of(context).showSnackBar(
                    const SnackBar(content: Text('Processing Data')));
              }
            },
            child: const Text('Submit'),
          ),
        ],
      ),
    )),
  ),
 );
 }
}
```


9:20
Form Validation
Enter your username
Enter your password
Enter your email
Submit
9:21
Form Validation
Enter your username
Please enter your username
Enter your password
Please Enter A Valid Password
Enter your email
Please Enter A Valid Email
Submit



CONCLUSION:-In this experiment , we created an interactive form using “Form” & “TextFormField” widgets available in Flutter and validate using Regular Expressions(RegEx).