Experiment No 03

Aim: create application for Form and Form Validator

Code:

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

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

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    const appTitle = 'Aakash Dhotre';

    return MaterialApp(
      title: appTitle,
      home: Scaffold(
        appBar: AppBar(
          title: const Text(appTitle),
        ),
        body: const MyCustomForm(),
      ),
    );
  }
}

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

  @override
  MyCustomFormState createState() {
    return 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.
  //
  // Note: This is a GlobalKey<FormState>,
  // not a GlobalKey<MyCustomFormState>.
  final _formKey = GlobalKey<FormState>();
```

```
@override
Widget build(BuildContext context) {
 // Build a Form widget using the _formKey created above.
 return Form(
  key: _formKey,
  child: Column(
   crossAxisAlignment: CrossAxisAlignment.start,
   children: [
    TextFormField(
     decoration: InputDecoration(
      enabledBorder: OutlineInputBorder(
         borderSide: BorderSide(width: 3, color: Colors.blueAccent),
         borderRadius: BorderRadius.circular(50.0)),
      hintText: 'Enter your full name',
      labelText: 'Name',
     ),
     // The validator receives the text that the user has entered.
     validator: (value) {
      if (value == null || value.isEmpty) {
       return 'Please enter some text';
      }
      return null;
     },
    ),
    TextFormField(
     decoration: InputDecoration(
      enabledBorder: OutlineInputBorder(
         borderSide: BorderSide(width: 3, color: Colors.blueAccent),
         borderRadius: BorderRadius.circular(50.0)),
      hintText: 'Enter your email',
      labelText: 'Email',
     ),
     validator: (value) {
      if (value == null || value.isEmpty) {
       return 'Please enter correct email';
      }
      if (value == null ||
         value.isEmpty ||
         !RegExp(r"^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'*+-/=?^_`{|}~]+@[a-zA-Z0-9]+\.[a-zA-Z]+")
            .hasMatch(value)) {
       return 'Enter a valid email!';
      }
      return null;
     },
    ),
    TextFormField(
     decoration: InputDecoration(
      enabledBorder: OutlineInputBorder(
```

```
            borderSide: BorderSide(width: 3, color: Colors.blueAccent),
            borderRadius: BorderRadius.circular(50.0)),
          hintText: 'Enter a phone number',
          labelText: 'Phone',
        ),
        validator: (value) {
          if (value == null ||
              value.isEmpty ||
              !RegExp(r'(^(?:[+0]9)?[0-9]{10,12}$)').hasMatch(value)) {
            return 'Please enter valid phone number';
          }
          return null;
        },
      ),
      Padding(
        padding: const EdgeInsets.symmetric(vertical: 16.0),
        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. In the real world,
              // you'd often call a server or save the information in a database.
              ScaffoldMessenger.of(context).showSnackBar(
                const SnackBar(content: Text('Processing Data')),
              );
            }
          },
          child: const Text('Submit'),
        ),
      ),
    ],
  ),
 );
 }
}
```

Output: