120A3051

Shreya Idate

Batch: E3

Experiment No: 5

AIM: To create an interactive Form using form widget

THEORY:

Flutter Form

Apps often require users to enter information into a text field. For example, you might require users to log in with an email address and password combination.

To make apps secure and easy to use, check whether the information the user has provided is valid. If the user has correctly filled out the form, process the information. If the user submits incorrect information, display a friendly error message letting them know what went wrong.

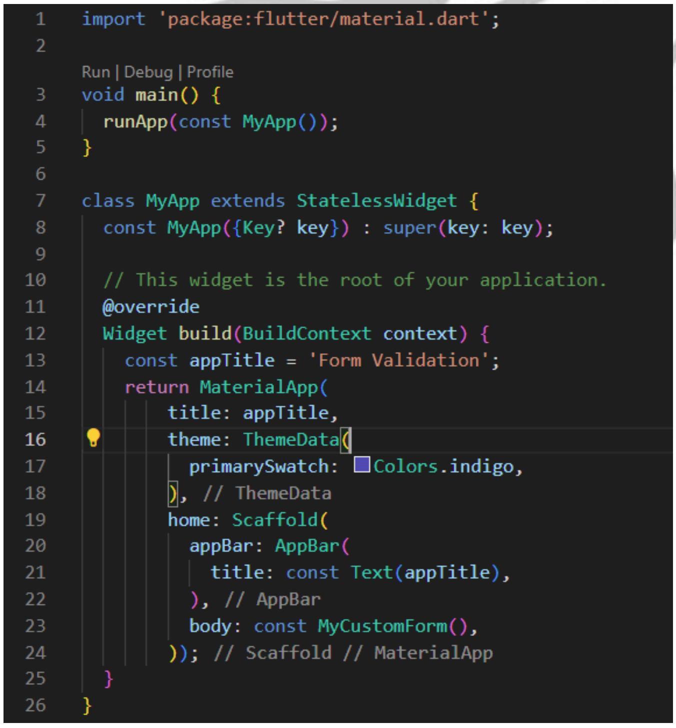
In this example, add validation to a form that has a single text field using the following steps:

1. Create a Form with a GlobalKey.
2. Add a TextFormField with validation logic.
3. Create a button to validate and submit the form.

1. Create a Form with a GlobalKey

First, create a Form. The Form widget acts as a container for grouping and validating multiple form fields.

When creating the form, provide a GlobalKey. This uniquely identifies the Form, and allows validation of the form in a later step.

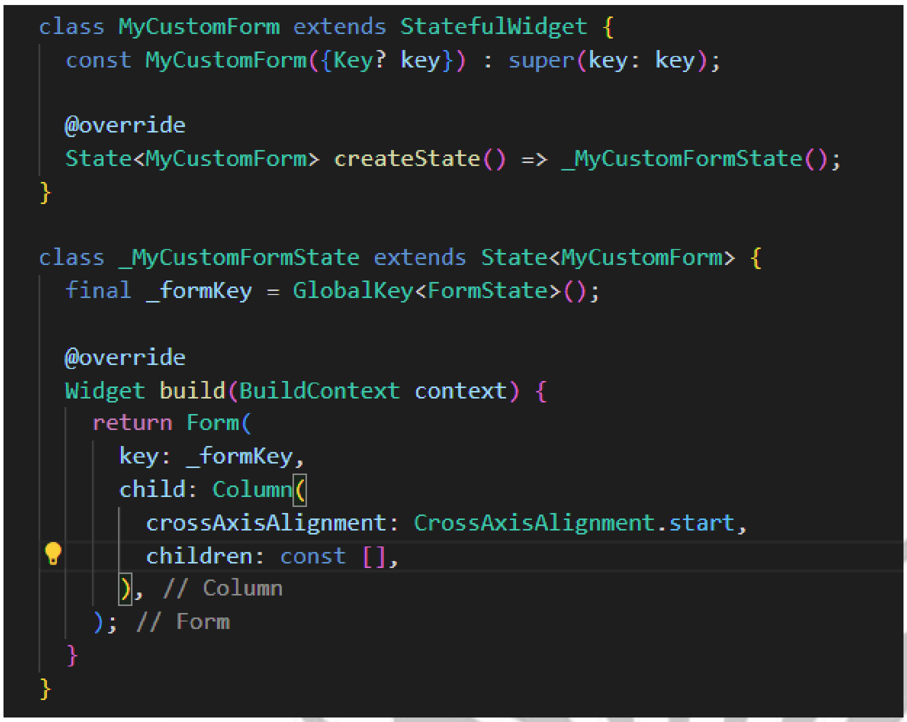


```dart
import 'package:flutter/material.dart';

Run | Debug | Profile
void main() {
  runApp(const MyApp());
}

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    const appTitle = 'Form Validation';
    return MaterialApp(
        title: appTitle,
        theme: ThemeData(
          primarySwatch: Colors.indigo,
        ), // ThemeData
        home: Scaffold(
          appBar: AppBar(
            title: const Text(appTitle),
          ), // AppBar
          body: const MyCustomForm(),
        )); // Scaffold // MaterialApp
  }
}
```

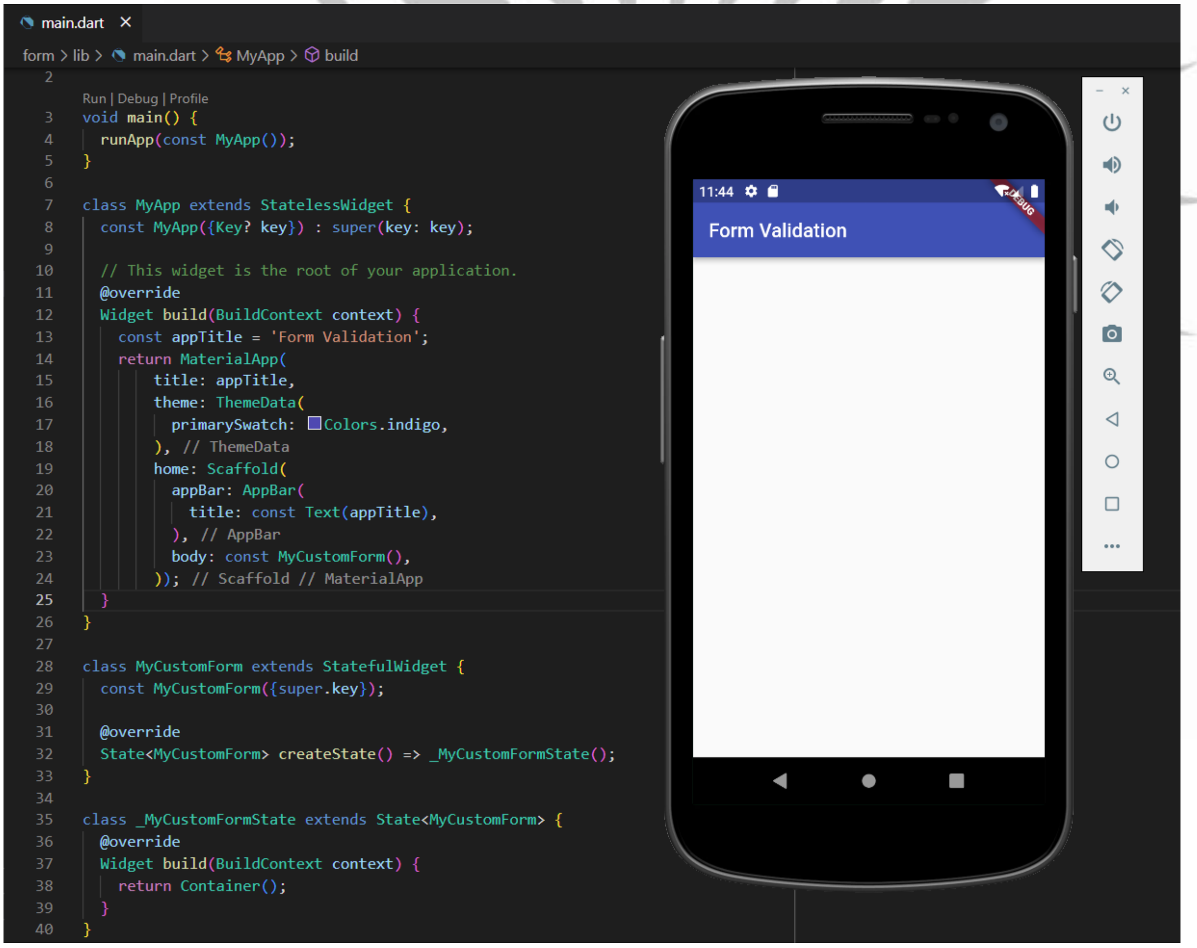


```dart
class MyCustomForm extends StatefulWidget {
  const MyCustomForm({Key? key}) : super(key: key);

  @override
  State<MyCustomForm> createState() => _MyCustomFormState();
}

class _MyCustomFormState extends State<MyCustomForm> {
  final _formKey = GlobalKey<FormState>();

  @override
  Widget build(BuildContext context) {
    return Form(
      key: _formKey,
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: const [],
      ), // Column
    ); // Form
  }
}
```



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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    const appTitle = 'Form Validation';
    return MaterialApp(
        title: appTitle,
        theme: ThemeData(
          primarySwatch: Colors.indigo,
        ), // ThemeData
        home: Scaffold(
          appBar: AppBar(
            title: const Text(appTitle),
          ), // AppBar
          body: const MyCustomForm(),
        )); // Scaffold // MaterialApp
  }
}

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

  @override
  State<MyCustomForm> createState() => _MyCustomFormState();
}

class _MyCustomFormState extends State<MyCustomForm> {
  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

2. Add a TextFormField with validation logic

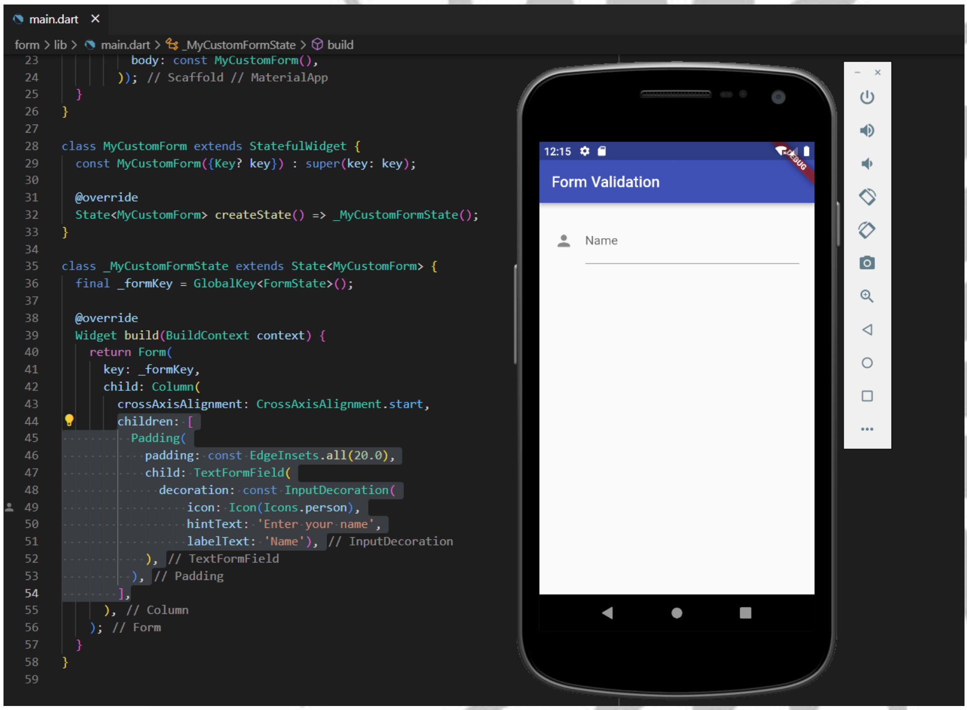
Although the Form is in place, it doesn't have a way for users to enter text. That's the job of a TextFormField. The **TextFormField** widget renders a material design text field and can display validation errors when they occur.

Validate the input by providing a **validator()** function to the TextFormField. If the user's input isn't valid, the validator function returns a String containing an error message. If there are no errors, the validator must return null.

For this example, create a validator that ensures the TextFormField isn't empty. If it is empty, return a friendly error message.

First decorate your TextField to display hint text , icon and label text.

```
child: TextFormField(
        decoration: const InputDecoration(
          border: UnderlineInputBorder(),
          labelText: 'Enter your username',
        ),
```



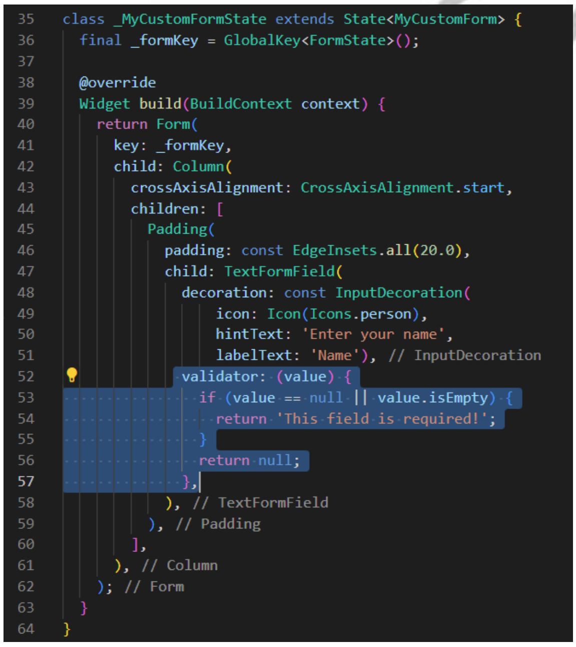
```
        body: const MyCustomForm(),
      )); // Scaffold // MaterialApp
  }
}

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

  @override
  State<MyCustomForm> createState() => _MyCustomFormState();
}

class _MyCustomFormState extends State<MyCustomForm> {
  final _formKey = GlobalKey<FormState>();

  @override
  Widget build(BuildContext context) {
    return Form(
      key: _formKey,
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Padding(
            padding: const EdgeInsets.all(20.0),
            child: TextFormField(
              decoration: const InputDecoration(
                  icon: Icon(Icons.person),
                  hintText: 'Enter your name',
                  labelText: 'Name'), // InputDecoration
            ), // TextFormField
          ), // Padding
        ],
      ), // Column
    ); // Form
  }
}
```



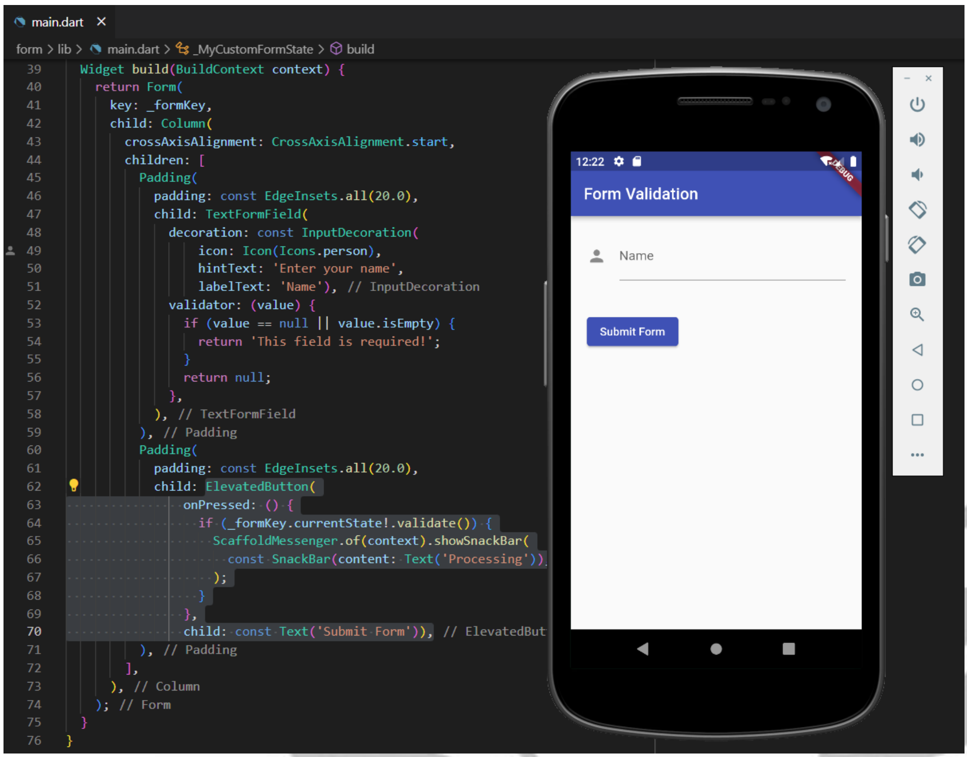
```
class _MyCustomFormState extends State<MyCustomForm> {
  final _formKey = GlobalKey<FormState>();

  @override
  Widget build(BuildContext context) {
    return Form(
      key: _formKey,
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Padding(
            padding: const EdgeInsets.all(20.0),
            child: TextFormField(
              decoration: const InputDecoration(
                  icon: Icon(Icons.person),
                  hintText: 'Enter your name',
                  labelText: 'Name'), // InputDecoration
              validator: (value) {
                if (value == null || value.isEmpty) {
                  return 'This field is required!';
                }
                return null;
              },
            ), // TextFormField
          ), // Padding
        ],
      ), // Column
    ); // Form
  }
}
```

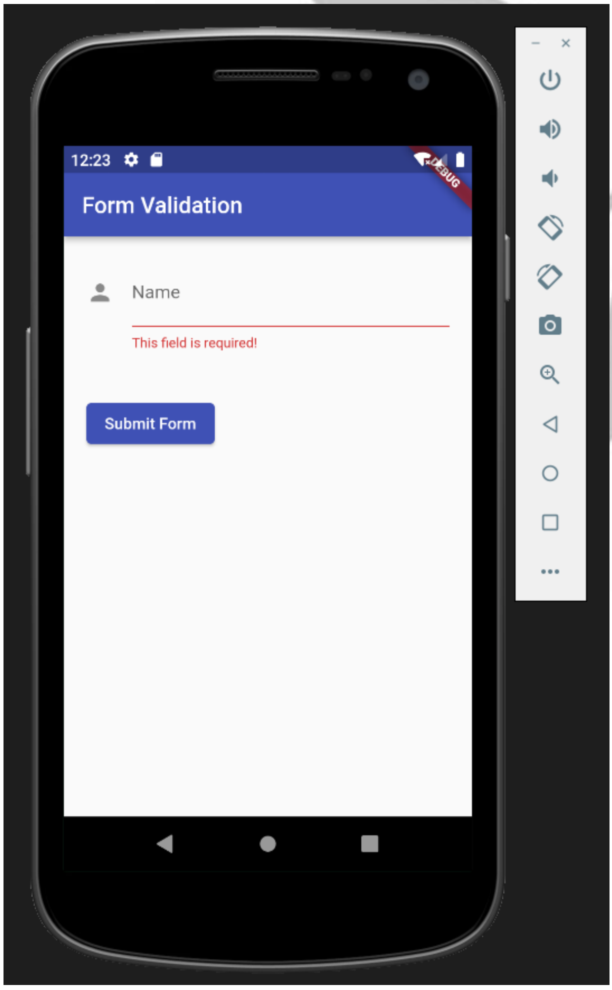
3. Create a button to validate and submit the form

Now that you have a form with a text field, provide a button that the user can tap to submit the information.

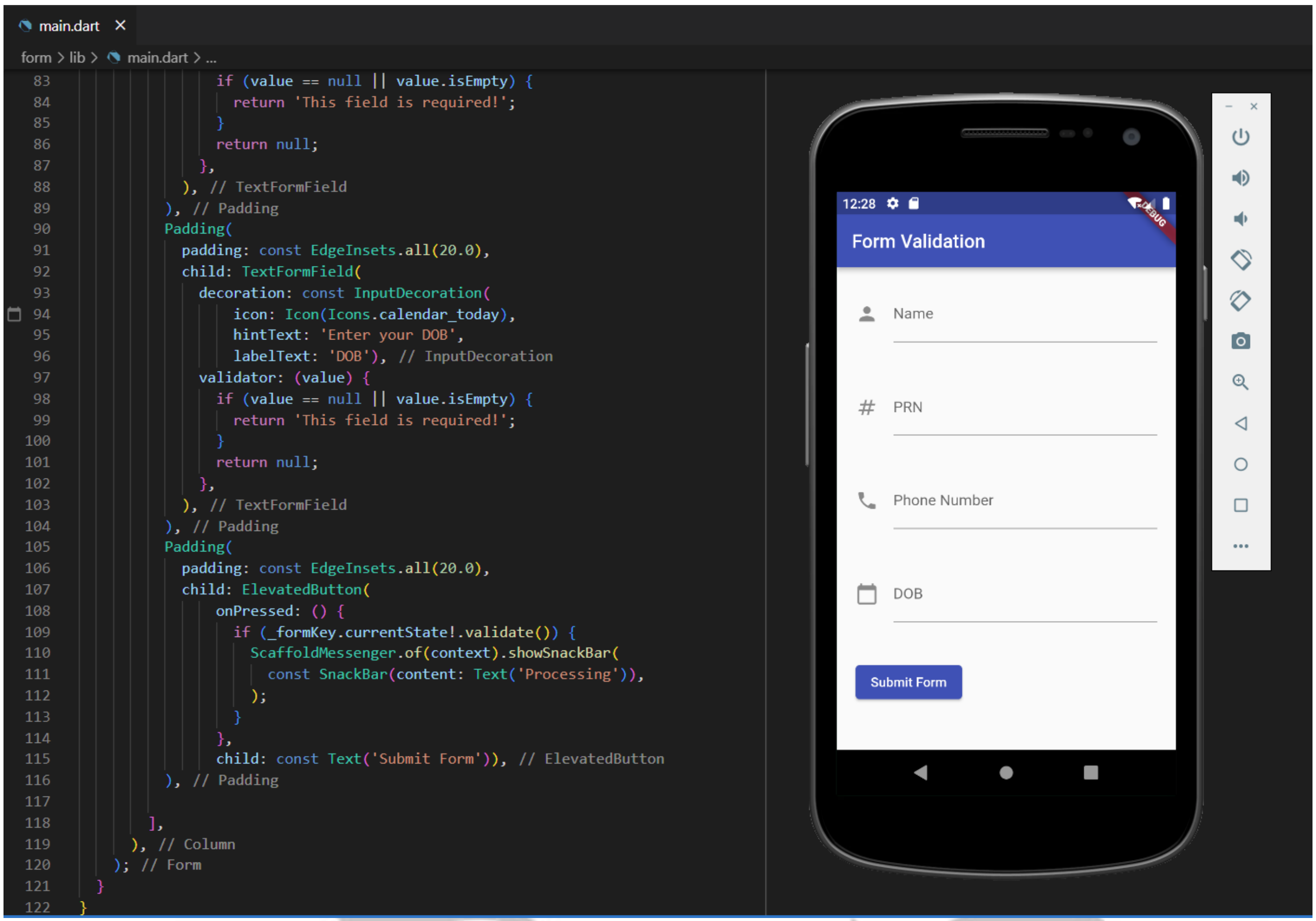
When the user attempts to submit the form, check if the form is valid. If it is, display a success message. If it isn’t (the text field has no content) display the error message.


main.dart
form > lib > main.dart > _MyCustomFormState > build

```dart
  Widget build(BuildContext context) {
    return Form(
      key: _formKey,
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Padding(
            padding: const EdgeInsets.all(20.0),
            child: TextFormField(
              decoration: const InputDecoration(
                  icon: Icon(Icons.person),
                  hintText: 'Enter your name',
                  labelText: 'Name'), // InputDecoration
              validator: (value) {
                if (value == null || value.isEmpty) {
                  return 'This field is required!';
                }
                return null;
              },
            ), // TextFormField
          ), // Padding
          Padding(
            padding: const EdgeInsets.all(20.0),
            child: ElevatedButton(
              onPressed: () {
                if (_formKey.currentState!.validate()) {
                  ScaffoldMessenger.of(context).showSnackBar(
                    const SnackBar(content: Text('Processing'))
                  );
                }
              },
              child: const Text('Submit Form')), // ElevatedBut
          ), // Padding
        ],
      ), // Column
    ); // Form
  }
}
```



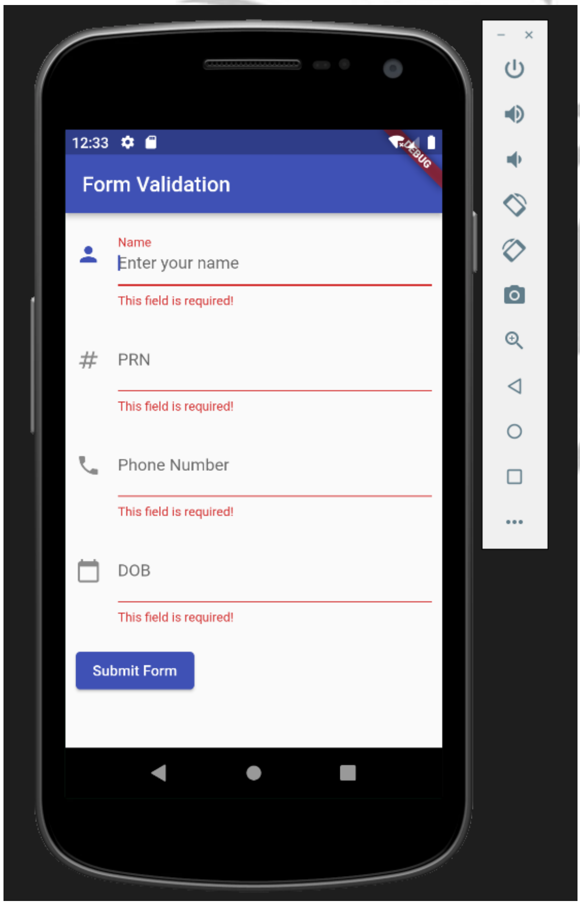
Add two more textfields to accept phone and dob respectively as shown below:


main.dart ✕

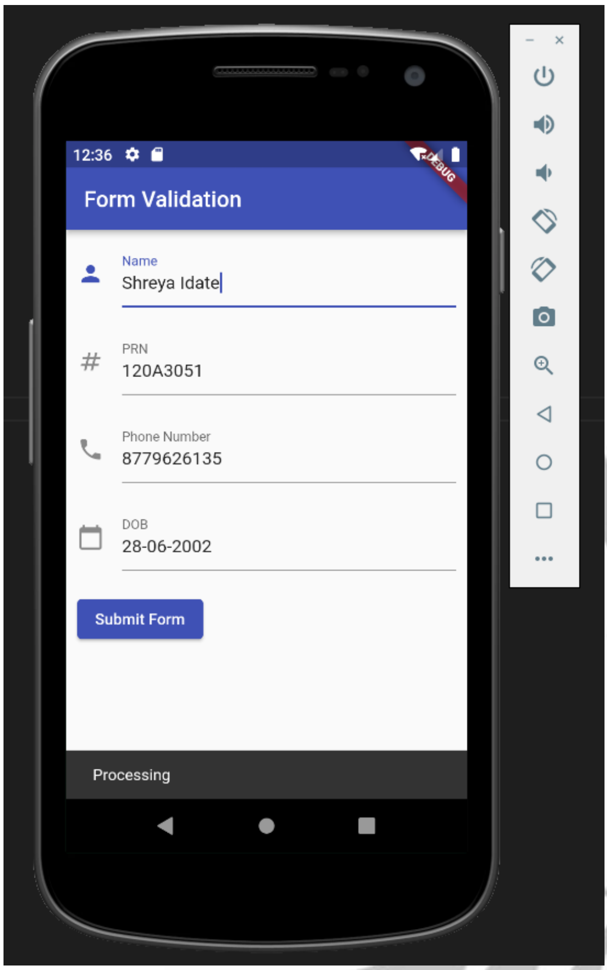
form > lib > main.dart > ...

```
                if (value == null || value.isEmpty) {
                  return 'This field is required!';
                }
                return null;
              },
            ), // TextFormField
          ), // Padding
          Padding(
            padding: const EdgeInsets.all(20.0),
            child: TextFormField(
              decoration: const InputDecoration(
                  icon: Icon(Icons.calendar_today),
                  hintText: 'Enter your DOB',
                  labelText: 'DOB'), // InputDecoration
              validator: (value) {
                if (value == null || value.isEmpty) {
                  return 'This field is required!';
                }
                return null;
              },
            ), // TextFormField
          ), // Padding
          Padding(
            padding: const EdgeInsets.all(20.0),
            child: ElevatedButton(
                onPressed: () {
                  if (_formKey.currentState!.validate()) {
                    ScaffoldMessenger.of(context).showSnackBar(
                      const SnackBar(content: Text('Processing')),
                    );
                  }
                },
                child: const Text('Submit Form')), // ElevatedButton
          ), // Padding

        ],
      ), // Column
    ); // Form
  }
}
```



Snackbar message text


12:36

Form Validation

Name
Shreya Idate

PRN
120A3051

Phone Number
8779626135

DOB
28-06-2002

Submit Form

Processing

CONCLUSION: Hence we have successfully designed an interactive Form using form widget.