

EXPERIMENT NO: - 04**Name:-** Swaraj Patil**Class:-** D15A**Roll:No: -** 39**AIM: -** To create an interactive Form using form widget.**Theory: -**

A form in Flutter is a structured container that collects user input through various fields like text fields, dropdowns, checkboxes, and buttons. It plays a crucial role in applications that require user data entry, such as login pages, registration forms, and feedback submissions. Flutter provides the **Form** widget, which works alongside **TextFormField** and other input elements to manage validation, state handling, and error messages efficiently. By using form validation techniques, developers can ensure data accuracy and enhance user experience.

When you create a form, it is necessary to provide the **GlobalKey**. This key uniquely identifies the form and allows you to do any validation in the form fields. The form widget uses child widget **TextFormField** to provide the users to enter the text field. This widget renders a material design text field and also allows us to display validation errors when they occur.

Creation of a Form

Swaraj Patil – D15A / 39

- While creating a form in Flutter, the **Form widget** is essential as it acts as a container for grouping multiple form fields and managing validation.
- A **GlobalKey<FormState>** is required to uniquely identify the form and enable validation or data retrieval from the form fields.
- The **TextFormField widget** is used to provide input fields where users can enter data such as names, phone numbers, or email addresses.
- To enhance the appearance and usability of input fields, **InputDecoration** is used, allowing customization of labels, icons, borders, and hint text.
- Validation plays a crucial role in forms, and the **validator property** within **TextFormField** ensures user input meets specific criteria before submission.
- Different types of input require appropriate **keyboard types**, such as **TextInputType.number** for numeric fields or **TextInputType.emailAddress** for email fields.
- Proper **state management** is needed to store and retrieve user input, ensuring the form data is processed correctly.
- A **submit button** is necessary to trigger form validation and submit the collected data for further processing.

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.

Some Methods of Form Widget

- **validate():** This method is used to trigger the validation of all the form fields within the Form. It returns true if all fields are valid, otherwise false. You can use it to check the overall validity of the form before submitting it.
- **save():** This method is used to save the current values of all form fields. It invokes the onSave callback for each field. Typically, this method is called after validation succeeds.
- **reset():** Resets the form to its initial state, clearing any user-entered data.
- **currentState:** A getter that returns the current FormState associated with the Form.

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

```
void main() {
```

```

runApp(MyApp());
    ),
    body: Padding(
        padding: const
        EdgeInsets.all(16.0),
class MyApp extends StatelessWidget
{
    child: InteractiveForm(),
    @override
    ),
    Widget build(BuildContext context)
    );
{
    }
    return MaterialApp(
    )
    title: 'Telegram Clone',
    theme: ThemeData(
        class InteractiveForm extends
        primarySwatch: Colors.blue, StatefulWidget {
    ),
    @override
    home: HomePage(),
    _InteractiveFormState
    createState() =>
    _InteractiveFormState();
    }
    }
}

class _InteractiveFormState extends
class HomePage extends
State<InteractiveForm> {
StatelessWidget {
    final _formKey =
    @override
    GlobalKey<FormState>(); // Key to
    identify the form
    Widget build(BuildContext context)
    final _nameController =
    {
    TextEditingController();
    return Scaffold(
    final _phoneController =
    appBar: AppBar(
    TextEditingController();
    title: Text('Telegram'),
    final _bioController =

```

```

TextEditingController();

    );

    }

    @override

    }

    void dispose() {

        // Clean up the controllers when
the widget is disposed

        _nameController.dispose();

        _phoneController.dispose();

        _bioController.dispose();

        super.dispose();

    }

    void _submitForm() {

        if

        (_formKey.currentState!.validate())

        {

            // If the form is valid,
display the data

            ScaffoldMessenger.of(context).showSn
ackBar(

                SnackBar(

                    content: Text(

                        'Form
Submitted!\nName:
${_nameController.text}\nPhone:
${_phoneController.text}\nBio:
${_bioController.text}'),

                        ),

                        if (value == null ||

```

```

value.isEmpty) {
    return 'Please enter your phone number';
}

return 'Please enter your name';

if
}
(!RegExp(r'^[0-9]{10}$').hasMatch(value)) {
    return null;
    return 'Please enter a valid 10-digit phone number';
},
},
return null;
},
// Phone Number Field
TextFormField(
    controller:
    _phoneController,
    decoration:
    InputDecoration(
        labelText: 'Phone Number',
        border:
        OutlineInputBorder(),
        prefixIcon:
        Icon(Icons.phone),
    ),
    keyboardType:
    TextInputType.phone,
    validator: (value) {
        if (value == null || value.isEmpty) {
            return 'Please enter
your phone number';
        }
        return null;
    },
),
// Bio Field
TextFormField(
    controller:
    _bioController,
    decoration:
    InputDecoration(
        labelText: 'Bio',
        border:
        OutlineInputBorder(),
        prefixIcon:
        Icon(Icons.edit),
    ),
    maxLines: 3,
    validator: (value) {
        if (value == null ||

```

```

value.isEmpty) {

        return 'Please enter
a bio';

    }

    return null;

},

),

    SizedBox(height: 24),

    // Submit Button

    Center(

        child: ElevatedButton(

            onPressed:
_submitForm,

            child: Text('Submit'),

        ),

    ),

],

),

);

}


}


```

OUTPUT:

Interactive Form

DEBUG

 Name

 Phone Number

 Bio

Submit