

## **Exp No: 4**

**Aim:** To create an interactive Form using form widget.

### **Theory:**

In Flutter, forms are used to collect user input and interact with users. Flutter provides a set of widgets and classes to create forms efficiently.

#### **1) Form Widget:**

The Form widget is a container for form fields and manages the form's state. It's typically used as the root widget of a form and helps with validation and submission.

The Form widget doesn't display anything by itself but provides utilities to interact with its child form fields.

#### **2) FormField Widget:**

The FormField widget represents a single form field within a Form.

It's a generic class that's extended by various field-specific widgets like TextFormField, DropdownButtonFormField, CheckboxFormField, etc.

FormField widgets handle user input, validation, and error messages associated with the field.

#### **3) TextFormField:**

TextFormField is a commonly used form field widget for collecting text input from users.

It provides features like keyboard input, text editing, validation, error handling, and input formatting.

You can customize its appearance, input type, validator, controller, and more.

#### **4) Form Validation:**

Flutter provides built-in support for form validation, which ensures that user input meets specific criteria before submission.

You can define validation logic using validators like required, minLength, maxLength, email, numeric, etc., or create custom validators.

Form validation is typically performed within the validator parameter of form field widgets or by implementing the FormFieldValidator function.

### 5) Form Submission:

After validating user input, you can handle form submission using callbacks like `onSaved` or `onFieldSubmitted` provided by form field widgets.

Alternatively, you can use the `FormState` object to access and process the form data when the form is submitted.

Form submission involves processing the input data, performing additional actions, and updating the UI accordingly.

### 6) `GlobalKey<FormState>`:

To interact with a Form and its state, you typically use a `GlobalKey<FormState>` object.

The `FormState` object contains methods to validate, reset, and save form fields. It's essential for accessing and manipulating the state of the form, especially when performing actions like validation and submission.

### Code:

```
import 'package:flutter/material.dart';
import 'package:igclone/resources/auth_methods.dart';
import 'package:igclone/screens/login_screen.dart';

import 'package:igclone/utils/colors.dart';
import 'package:igclone/widgets/text_file_input.dart';

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

  @override
  State<SignupScreen> createState() => _SignupScreenState();
}

class _SignupScreenState extends State<SignupScreen> {
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final TextEditingController _bioController = TextEditingController();
  final TextEditingController _usernameController = TextEditingController();

  @override
  void dispose() {
```

```
// TODO: implement dispose
super.dispose();
_emailController.dispose();
_passwordController.dispose();
_bioController.dispose();
_usernameController.dispose();
}
```

@override

```
Widget build(BuildContext context) {
  return Scaffold(
    body: SafeArea(
      child: Container(
        padding: EdgeInsets.symmetric(horizontal: 32),
        width: double.infinity,
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            Flexible(
              child: Container(),
              flex: 2,
            ),
            //svg img
            Image.asset(
              'assets/igword.png',
              height: 64,
              color: primaryColor,
            ),
            const SizedBox(height: 64),

            //circular widget for displaying user card
            Stack(
              children: [
                const CircleAvatar(
                  radius: 64,
                  backgroundImage: NetworkImage(
```

```
'https://images.unsplash.com/photo-1633332755192-727a05c4013d?w=500&auto=
=format&fit=crop&q=60&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8Mnx
8dXNlcnxlbmwHwwfHx8MA%3D%3D'),
```

```
    ),
    Positioned(
      bottom: -10,
      left: 80,
      child: IconButton(
        onPressed: () {},
        icon: const Icon(Icons.add_a_photo),
      )),
  ],
),
const SizedBox(
  height: 25,
),
//text for username
TextFieldInput(
  hintText: 'Enter your username',
  TextInputType: TextInputType.text,
  textEditingController: _usernameController,
),
const SizedBox(
  height: 25,
),

//text for email
TextFieldInput(
  hintText: 'Enter your email',
  TextInputType: TextInputType.emailAddress,
  textEditingController: _emailController,
),

const SizedBox(
  height: 25,
),
```

```

//text for pass
TextFieldInput(
  hintText: 'Enter your password',
  TextInputType: TextInputType.text,
  textEditingController: _passwordController,
  isPass: true,
),

const SizedBox(
  height: 25,
),

//text for bio
TextFieldInput(
  hintText: 'Enter your bio',
  TextInputType: TextInputType.emailAddress,
  textEditingController: _bioController,
),
const SizedBox(
  height: 25,
),
//button for login
InkWell(
  onTap: () async {
    String res = await AuthMethods().signUpUser(
      email: _emailController.text,
      password: _passwordController.text,
      username: _usernameController.text,
      bio: _bioController.text,
    );

    print(res);
  },
  child: Container(
    child: const Text('Sign up'),
    width: double.infinity,
    alignment: Alignment.center,
  ),
)

```

```

padding: const EdgeInsets.symmetric(vertical: 12),
decoration: const BoxDecoration(
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.all(
      Radius.circular(4),
    ),
  ),
  color: blueColor,
),
),
const SizedBox(
  height: 12,
),
Flexible(
  child: Container(),
  flex: 2,
),

```

//transition to signup

```

Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Container(
      child: Text("Don't have an account?"),
      padding: const EdgeInsets.symmetric(vertical: 8),
    ),
    GestureDetector(
      onTap: () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => const LoginScreen(),
          );
      },
      child: Container(
        child: Text(

```

```

        "LogIn",
        style: TextStyle(fontWeight: FontWeight.bold),
      ),
      padding: const EdgeInsets.symmetric(vertical: 8),
    ),
  ),
],
)
],
),
)))
}
}

```

## Output:

