**Experiment No:04**

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

**Theory:**

**Enhancing Visual Appeal and Functionality in Flutter Apps with Icons, Images, and Fonts**

In Flutter, the Form widget is an essential component for creating interactive user input forms. It enables input validation, data submission, and error handling. Here's an overview of how to utilize the Form widget effectively:

1. **Understanding the Form Widget**

* **Definition:** The Form widget acts as a container that houses multiple form fields, enabling users to input data.
* **State Management:** It manages the state of the form and offers methods for validation and submission.

1. **Building a Form::**

* **Creation:** Wrap your form fields within a Form widget to create a form.
* **Global Key:** Utilize the GlobalKey<FormState> to uniquely identify the form and access its state.

1. **Utilizing Form Fields:**

* **Types:** Various form fields like TextFormField, DropdownButtonFormField, etc., collect user input.
* **Configuration:** Each form field should be associated with a controller (for controlled input) and a validator function to ensure data integrity.

1. **Validation Process:**

* **Purpose:** Validation ensures that user input meets predefined criteria before submission.
* **Implementation:** Specify validation logic using the validator property of form fields. Validators are functions that return an error message if validation fails, or null if the input is valid.

1. **Handling Form Submission:**

* **Triggering Submission:** Submission occurs when the user interacts with a submit button or similar action.
* **Submission Process:** Inside the submission handler, validate the form using the validate method of the FormState. If the form is valid, proceed with the submission logic, such as saving data to a database.

1. **Error Management:**

* **Guiding Users:** If form validation fails, display error messages to guide users in correcting their input.
* **Error Display:** Errors can be shown below each form field or as a general error message at the top of the form.

1. **Cleanup Procedures:**

* **Resource Disposal:** Dispose of form controllers and other resources in the dispose method of the State object to prevent memory leaks.

1. **Additional Features:**

* **Enhancements:** Flutter offers various widgets and utilities to augment forms, such as InputDecoration for customizing form field appearance, FocusNode for managing focus between fields, and SnackBar for displaying feedback messages.

**Code:**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Interactive Form Example',

theme: ThemeData(

primaryColor: Colors.blue, // Change primary color of the app

appBarTheme: AppBarTheme(

titleTextStyle: TextStyle(

fontSize: 20.0,

fontWeight: FontWeight.bold,

color: Colors.white, // Change text color of the AppBar title

),

),

),

home: MyForm(),

);

}

}

class MyForm extends StatefulWidget {

@override

\_MyFormState createState() => \_MyFormState();

}

class \_MyFormState extends State<MyForm> {

final GlobalKey<FormState> \_formKey = GlobalKey<FormState>();

TextEditingController \_nameController = TextEditingController();

TextEditingController \_emailController = TextEditingController();

TextEditingController \_mobileController = TextEditingController();

String? \_gender;

@override

void dispose() {

\_nameController.dispose();

\_emailController.dispose();

\_mobileController.dispose();

super.dispose();

}

void \_submitForm() {

if (\_formKey.currentState != null && \_formKey.currentState!.validate()) {

// If the form is valid, perform actions like saving to a database or API

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(content: Text('Form is validated and submitted!')),

);

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Expt-4 Interactive Form'),

// Customize background color of the AppBar

backgroundColor: Colors.green,

),

body: Padding(

padding: EdgeInsets.all(16.0),

child: Form(

key: \_formKey,

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: <Widget>[

TextFormField(

controller: \_nameController,

decoration: InputDecoration(

labelText: 'Name',

),

validator: (value) {

if (value == null || value.isEmpty) {

return 'Please enter your name';

}

return null;

},

),

TextFormField(

controller: \_emailController,

decoration: InputDecoration(

labelText: 'Email',

),

validator: (value) {

if (value == null || value.isEmpty) {

return 'Please enter your email';

}

if (!value.contains('@')) {

return 'Please enter a valid email';

}

return null;

},

),

TextFormField(

controller: \_mobileController,

decoration: InputDecoration(

labelText: 'Mobile Number',

),

keyboardType: TextInputType.phone,

validator: (value) {

if (value == null || value.isEmpty) {

return 'Please enter your mobile number';

}

if (value.length != 10) {

return 'Please enter a valid 10-digit mobile number';

}

return null;

},

),

DropdownButtonFormField<String>(

decoration: InputDecoration(

labelText: 'Gender',

),

value: \_gender,

items: ['Male', 'Female', 'Other']

.map((gender) => DropdownMenuItem<String>(

value: gender,

child: Text(gender),

))

.toList(),

onChanged: (value) {

setState(() {

\_gender = value;

});

},

validator: (value) {

if (value == null) {

return 'Please select your gender';

}

return null;

},

),

SizedBox(height: 20),

ElevatedButton(

onPressed: \_submitForm,

child: Text('Submit'),

),

],

),

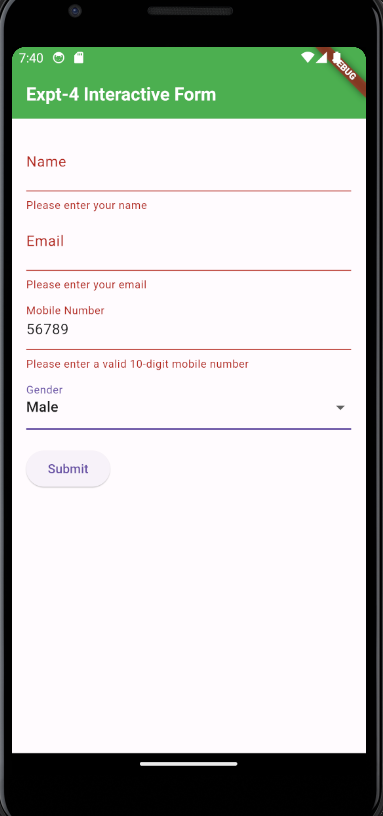
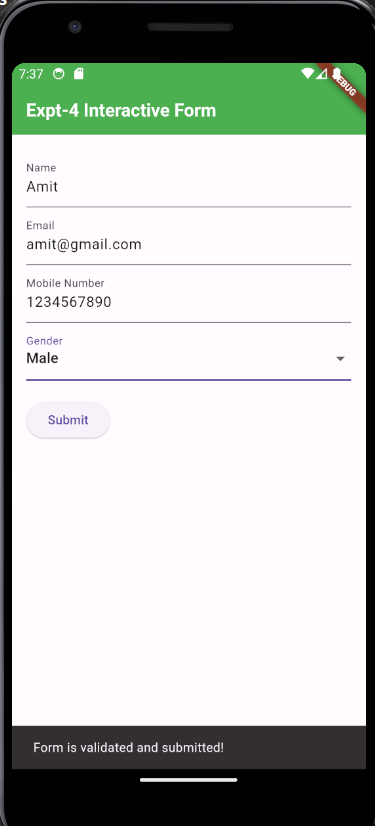
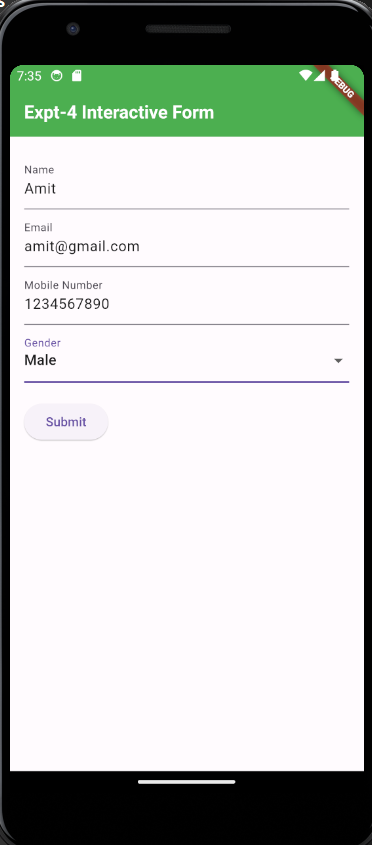
),

),

);

}

}

****

**2.Login Form:**

**Code:**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Login Form',

theme: ThemeData(

primaryColor: Colors.blue, // Change primary color of the app

appBarTheme: AppBarTheme(

titleTextStyle: TextStyle(

fontSize: 20.0,

fontWeight: FontWeight.bold,

color: Colors.white, // Change text color of the AppBar title

),

),

),

home: LoginForm(),

);

}

}

class LoginForm extends StatefulWidget {

@override

\_LoginFormState createState() => \_LoginFormState();

}

class \_LoginFormState extends State<LoginForm> {

final GlobalKey<FormState> \_formKey = GlobalKey<FormState>();

TextEditingController \_emailController = TextEditingController();

TextEditingController \_passwordController = TextEditingController();

@override

void dispose() {

\_emailController.dispose();

\_passwordController.dispose();

super.dispose();

}

void \_submitForm() {

if (\_formKey.currentState != null && \_formKey.currentState!.validate()) {

// If the form is valid, perform actions like login authentication

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(content: Text('Login Successful!')),

);

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Login Form'),

),

body: Padding(

padding: EdgeInsets.all(16.0),

child: Form(

key: \_formKey,

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.stretch,

children: <Widget>[

ElevatedButton(

onPressed: () {

// Action for Login using Google

},

child: Text('Login using Google'),

),

SizedBox(height: 20),

Text(

'Login Options',

style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),

textAlign: TextAlign.center,

),

SizedBox(height: 20),

TextFormField(

controller: \_emailController,

decoration: InputDecoration(

labelText: 'Email',

border: OutlineInputBorder(),

),

validator: (value) {

if (value == null || value.isEmpty) {

return 'Please enter your email';

}

if (!value.contains('@')) {

return 'Please enter a valid email';

}

return null;

},

),

SizedBox(height: 10),

TextFormField(

controller: \_passwordController,

obscureText: true,

decoration: InputDecoration(

labelText: 'Password',

border: OutlineInputBorder(),

),

validator: (value) {

if (value == null || value.isEmpty) {

return 'Please enter your password';

}

if (value.length < 6) {

return 'Password must be at least 6 characters long';

}

return null;

},

),

SizedBox(height: 20),

ElevatedButton(

onPressed: \_submitForm,

child: Text('Submit'),

),

SizedBox(height: 10),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text('Don\'t have an account?'),

TextButton(

onPressed: () {

// Action for Register Button

},

child: Text('Register'),

),

],

),

],

),

),

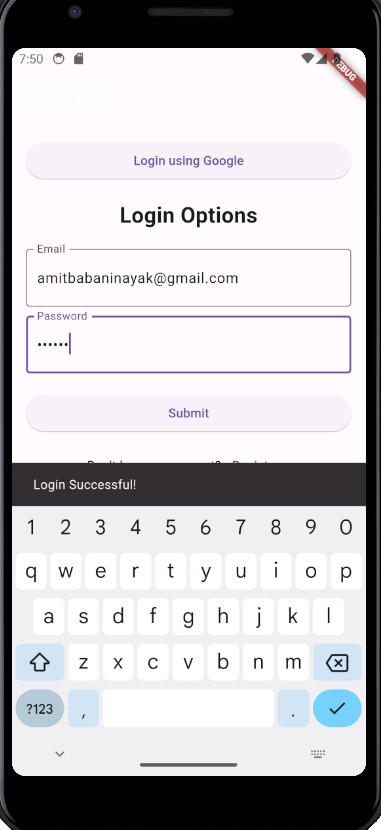
),

);

}

}

**Output:**



**Conclusion:** I have successfully created an interactive Form using form widget in Flutter.