

# EXPERIMENT NO: - 04

**Name:-** Arnav Sawant

**Class:-** D15A

**Roll:No: -** 52

**AIM: -** To create an interactive Form using form widget.

---

## **Forms in Flutter**

Forms in Flutter are essential components used to collect and manage user input efficiently. They are widely used in login screens, registration pages, and feedback forms. Flutter provides a **Form** widget that works alongside **TextFormField** and other input elements, offering features such as validation, error handling, and state management to improve user experience.

## **Key Components of a Form in Flutter**

### **1. Form Widget**

The **Form** widget acts as a container that groups multiple input fields and manages their validation.

- Requires a **GlobalKey<FormState>** to uniquely identify the form and interact with it.
- Helps in structuring form fields and handling user input efficiently.

### **2. Form Fields (TextFormField)**

The **TextFormField** widget is used for user input, such as entering names, emails, or phone numbers.

- It supports input validation using the **validator** property.
- Allows customization with **InputDecoration** (e.g., labels, icons, borders, hint text).
- Different **TextInputType** options can be set for appropriate keyboard input (e.g., **TextInputType.emailAddress** for emails).

### **3. Validation in Forms**

Ensuring valid user input is crucial. The **validator** property in **TextFormField** helps check whether the entered data meets specified criteria before submission.

- Validation can be triggered manually using `formKey.currentState!.validate()`.
- The **autovalidateMode** property can enable automatic validation during user input.

### **4. State Management in Forms**

To ensure data persistence and processing, proper state management is required.

- The **FormState** class provides methods like `validate()`, `save()`, and `reset()` to manage form behavior.
- The `save()` method stores user input when validation is successful.
- The `reset()` method clears the form fields and restores the initial state.

## 5. Submit Button

A **submit button** is necessary to trigger form validation and submit user data.

- When pressed, it checks validation using `formKey.currentState!.validate()`.
- If validation succeeds, the form data is saved and processed accordingly.

## Important Properties & Methods of Form Widget

### Properties

- **key** → A `GlobalKey<FormState>` that uniquely identifies the form.
- **child** → Contains form fields, typically wrapped in a `Column` or `ListView`.
- **autovalidateMode** → Defines when the form should auto-validate.

### Methods

- **validate()** → Checks if all form fields are valid and returns true or false.
- **save()** → Stores the current values of form fields after successful validation.
- **reset()** → Clears user input and resets the form to its initial state.
- **currentState** → Provides access to the form's state for validation, saving, or resetting.

Code:-

login.dart

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

class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() =>
    _LoginPageState();
}

class _LoginPageState extends
    State<LoginPage> {
  final _formKey = GlobalKey<FormState>();
  TextEditingController emailController =
    TextEditingController();
  TextEditingController passwordController =
    TextEditingController();
  TextEditingController usernameController =
    TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: Center(
        child: SingleChildScrollView(
          padding: const
            EdgeInsets.symmetric(horizontal: 32.0),
          child: Form(
            key: _formKey,
            child: Column(
              mainAxisAlignment:
                MainAxisAlignment.center,
              children: [
                // Logo
                Image.asset(
                  'assets/logo.jpg', // Replace with your
                    logo asset
                  height: 80,
                ),
                SizedBox(height: 40),

                // Username Field
                _buildTextField(controller:
                  usernameController, label: "Username"),
                SizedBox(height: 10),

                // Email Field
```

```

        _buildTextField(controller:
emailController, label: "Email", keyboardType:
TextInputType.emailAddress),
        SizedBox(height: 10),

```

*// Password Field*

```

        _buildTextField(controller:
passwordController, label: "Password",
obscureText: true),
        SizedBox(height: 20),

```

*// Login Button*

```

SizedBox(
  width: double.infinity,
  child: ElevatedButton(
    onPressed: () {
      if
(_formKey.currentState!.validate()) {
        // Handle login logic
      }
    },
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.blue,
      padding:
EdgeInsets.symmetric(vertical: 14),
    ),
    child: Text("Login", style:
TextStyle(fontSize: 16, color: Colors.white)),
  ),
  ),
  SizedBox(height: 20),

```

*// Social Login Buttons*

```

_buildLoginButton(
  color: Color(0xFF1877F2),
  text: "Sign in with Facebook",
  icon: Icons.facebook,
),
SizedBox(height: 10),
_buildLoginButton(
  color: Colors.white,
  text: "Sign in with Google",
  icon: Icons.g_mobiledata,
  borderColor: Colors.grey,
  textColor: Colors.black,
),
SizedBox(height: 10),
_buildLoginButton(
  color: Colors.blue,

```

```

        text: "Sign in with phone",
        icon: Icons.phone,
      ),
    ],
  ),
),
),
);
}

```

```

Widget _buildTextField({
  required TextEditingController controller,
  required String label,
  bool obscureText = false,
  TextInputType keyboardType =
TextInputType.text,
}) {
  return TextFormField(
    controller: controller,
    obscureText: obscureText,
    keyboardType: keyboardType,
    decoration: InputDecoration(
      labelText: label,
      border: OutlineInputBorder(),
    ),
    validator: (value) {
      if (value == null || value.isEmpty) {
        return "$label cannot be empty";
      }
      return null;
    },
  );
}

```

```

Widget _buildLoginButton({
  required Color color,
  required String text,
  required IconData icon,
  Color textColor = Colors.white,
  Color? borderColor,
}) {
  return Container(
    width: double.infinity,
    child: ElevatedButton.icon(
      icon: Icon(icon, color: textColor),
      label: Text(
        text,
        style: TextStyle(color: textColor),

```

```

),
onPressed: () {},
style: ElevatedButton.styleFrom(
  backgroundColor: color,
  padding: EdgeInsets.symmetric(vertical:
14),
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(8),
    side: borderColor != null ?

```

```

BorderSide(color: borderColor) :
BorderSide.none,
),
),
);
}
}

```

editprofile.dart

```

import 'package:flutter/material.dart';

class EditProfileScreen extends StatefulWidget {
  @override
  _EditProfileScreenState createState() =>
    _EditProfileScreenState();
}

class _EditProfileScreenState extends
State<EditProfileScreen> {
  TextEditingController nameController =
    TextEditingController(text: "Arnav");
  TextEditingController bioController =
    TextEditingController();

  int nameMaxLength = 20;
  int bioMaxLength = 120;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        leading: IconButton(
          icon: Icon(Icons.arrow_back, color:
Colors.black),
          onPressed: () => Navigator.pop(context),
        ),
        title: Text("Edit Profile", style:
TextStyle(color: Colors.black)),
        backgroundColor: Colors.white,
        elevation: 0,
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(

```

```

      crossAxisAlignment:
CrossAxisAlignment.start,
      children: [
        // Profile Picture
        Center(
          child: CircleAvatar(
            radius: 40,
            backgroundImage:
AssetImage('assets/profile.jpg'), // Replace with
your image
          ),
        ),
        SizedBox(height: 20),

        // Name Field
        _buildLabel("Name"),
        TextField(
          controller: nameController,
          maxLength: nameMaxLength,
          decoration: InputDecoration(
            border: OutlineInputBorder(),
          ),
        ),
        _buildCharacterLimit(nameController.te
xt.length, nameMaxLength),

        SizedBox(height: 16),

        // Bio Field
        _buildLabel("Your Bio"),
        TextField(
          controller: bioController,
          maxLength: bioMaxLength,
          maxLines: 3,
          decoration: InputDecoration(
            border: OutlineInputBorder(),

```

```

        hintText: "Enter your bio here",
      ),
    ),
    _buildCharacterLimit(bioController.text.
length, bioMaxLength),

    SizedBox(height: 20),

    // Submit Button
    SizedBox(
      width: double.infinity,
      child: ElevatedButton(
        onPressed: () {},
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.blue,
          padding:
EdgeInsets.symmetric(vertical: 14),
        ),
        child: Text("Submit", style:
TextStyle(fontSize: 16, color: Colors.white)),
      ),
    ),
  ],
),

```

```

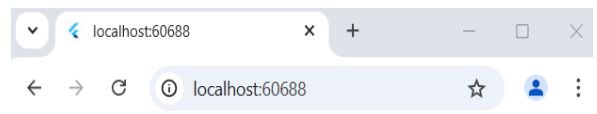
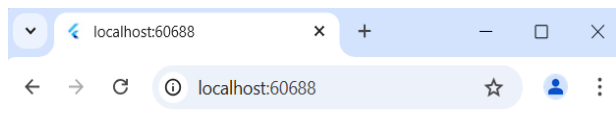
    ),
  );
}

// Helper to create labels
Widget _buildLabel(String text) {
  return Padding(
    padding: const EdgeInsets.only(bottom: 8),
    child: Text(text, style: TextStyle(fontSize:
16, fontWeight: FontWeight.bold)),
  );
}

// Character counter
Widget _buildCharacterLimit(int current, int
max) {
  return Align(
    alignment: Alignment.centerRight,
    child: Text("$current/$max", style:
TextStyle(color: Colors.grey, fontSize: 12)),
  );
}
}

```

Screenshots:-



## Edit Profile



Name

Arnav

5/20  
5/20

Your Bio

Enter your bio here

0/120  
0/120

Submit

