

**Name: Girish Chougule**

**Class: D15B**

**Roll: 10**

## **Lab 04**

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

### **Theory**

Form validation is an essential feature in mobile applications to ensure the correctness of user inputs before processing them. In Flutter, `Form` and `TextFormField` widgets are used to create forms with built-in validation capabilities.

### **Key Concepts:**

- 1. GlobalKey:** Used to uniquely identify the form and manage its state.
- 2. TextFormField:** A widget that allows users to enter input and supports validation.
- 3. Validation Logic:** Defines conditions to verify the correctness of input.
- 4. Submit Button:** Triggers form validation and processes the input if valid.

### **Code Implementation**

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

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: FormScreen(),
    );
  }
}
```

```
}
```

```
class FormScreen extends StatefulWidget {  
  @override  
  _FormScreenState createState() => _FormScreenState();  
}
```

```
class _FormScreenState extends State<FormScreen> {  
  final _formKey = GlobalKey<FormState>();  
  final TextEditingController _nameController = TextEditingController();
```

```
  void _submitForm() {  
    if (_formKey.currentState!.validate()) {  
      ScaffoldMessenger.of(context).showSnackBar(  
        SnackBar(content: Text('Form Submitted Successfully')),  
      );  
    }  
  }  
}
```

```
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(title: Text('Form Validation Demo')),  
    body: Padding(  
      padding: const EdgeInsets.all(16.0),  
      child: Form(  
        key: _formKey,  
        child: Column(  
          children: [  
            TextFormField(  
              controller: _nameController,  
              decoration: InputDecoration(labelText: 'Enter your name'),  
              validator: (value) {  
                if (value == null || value.isEmpty) {  
                  return 'Name cannot be empty';  
                }  
                return null;  
              }  
            ),  
          ],  
        ),  
      ),  
    ),  
  );  
}
```

```
    },  
  ),  
  SizedBox(height: 20),  
  ElevatedButton(  
    onPressed: _submitForm,  
    child: Text('Submit'),  
  ),  
],  
,  
,  
,  
,  
);  
}  
}
```

## Output

The screenshot shows a web browser window with the title 'Flutter Blog App' and the address bar displaying 'localhost:18791/#/myBlogs'. The browser interface includes back, forward, and refresh buttons, as well as icons for information, star, lightning bolt, user, and a menu. The main content area of the browser displays a 'Create Blog' form with a light pink background. The form includes a back arrow, a title 'Create Blog', and several input fields: 'Blog Title', 'Author Name', 'Select Category' (a dropdown menu), and 'Blog Description'. Below these fields is a 'Select Publish Date' label with a calendar icon. At the bottom of the form are two buttons: 'Upload Image' (with a small image icon) and 'Submit'.

## Conclusion

Form validation in Flutter can be efficiently managed using the `GlobalKey`, `TextFormField`, and validation functions. This ensures user inputs are validated before processing, improving app reliability.