Experiment no. 4
Name: Soham A. Phalke
Roll no.: 55
Div: D15B

Aim: To create an interactive Form using a form widget

Theory:
In Flutter, forms are a fundamental part of building user interfaces, allowing users to input data and interact with your application. The TextFormField widget is commonly used to create text input fields within a form. Here's an overview of each component you mentioned:

1. Form: A Form widget is a container for a group of form fields (such as TextFormField, Checkbox, Radio, etc.). It manages the form's state, validation, and submission.

2. TextFormField: A TextFormField widget is used to create a text input field within a form. It provides several properties for customizing the appearance and behavior of the input field, such as decoration (for styling), validator (for input validation), onSaved (for saving the input value), and more.

3. Button: In the context of forms, buttons are used to submit or reset the form. You can use the ElevatedButton, TextButton, OutlinedButton, or IconButton widgets to create buttons in Flutter. Typically, a button's onPressed callback is used to handle form submission or other actions.

4. Validator: A validator is a function that determines whether the input provided by the user is valid. It is commonly used with form fields like TextFormField to perform validation before submitting the form. The validator function takes the input value as a parameter and returns an error message if the input is invalid, or null if the input is valid.

5. FormKey: The FormKey is a unique key that identifies a Form widget in Flutter. It's used to access and interact with the form's state, including validating and submitting the form. You typically create a GlobalKey<FormState> and assign it to the key property of the Form widget.

Code:

Main.dart:

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



void main()=> runApp(MaterialApp(
 home: Home(),
 theme: ThemeData(
   fontFamily: 'Teko',
 ),
));



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

 @override
 State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {
```

```
int selected_index=0;
void _onTap(int index) {
  setState(() {
    selected_index = index;
  });
}
@override
Widget build(BuildContext context) {
  return Scaffold(

      appBar: AppBar(
        title:Text("Footy",
          style: TextStyle(
              fontSize: 25
          ),
        ),
        centerTitle: true,
        backgroundColor: Colors.teal[400],

      ),
      body:
```

```
Container(
  padding: EdgeInsets.all(25),
  child: _getBodyWidget(selected_index),),



bottomNavigationBar: BottomNavigationBar(
  items: [
    BottomNavigationBarItem(


      icon: Icon(Icons.home),
      label: 'Home',
      backgroundColor: Colors.deepOrange,
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.sports_soccer),
      label: 'Score',
      backgroundColor: Colors.green,
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.sensor_occupied),
      label: 'Profile',
```

```
            backgroundColor: Colors.green,
          ),




        ],
        currentIndex: selected_index,
        selectedItemColor: Colors.teal[600],


        onTap: _onTap,
      ),
    floatingActionButton: selected_index == 0 ? Float() : null,



  );
}
Widget _getBodyWidget(int index) {
  switch (index) {
    case 0:
      return Home_tab();
    case 1:
```

```
      return Score_tab();
    case 2:
      return Profile_tab();
    default:
      return SizedBox.shrink();
  }
 }
}


class Home_tab extends StatelessWidget {
 const Home_tab({super.key});

 @override
 Widget build(BuildContext context) {
  return  Container(
    child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceAround,


    children: [
```

```
ElevatedButton(// for player stats


  onPressed: () {
    // Add your onPressed callback here
  },
  child:Column(
    mainAxisSize: MainAxisSize.min, // Ensure that the column only occupies the space required by its children
    children: [
      Icon(Icons.person,color: Colors.teal,), // Icon widget
      SizedBox(height: 8), // Spacer between icon and text
      Text('Player',
          style:TextStyle(
            color: Colors.black,
          )),
      // Text widget
    ],
  ),// Specify the label
),


ElevatedButton(//for squad stats
```

```
        onPressed: () {
          // Add your onPressed callback here
        },
        child:Column(
          mainAxisSize: MainAxisSize.min, // Ensure that the column only occupies the space required by its children
          children: [
            Icon(Icons.people,color: Colors.teal,), // Icon widget
            SizedBox(height: 8), // Spacer between icon and text
            Text('Squad',
                style:TextStyle(
                  color: Colors.black,
                )), // Text widget
          ],
        ),// Specify the label
      ),

      ElevatedButton(//for points table
        onPressed: () {
          // Add your onPressed callback here
        },
        child:Column(
```

```
          mainAxisSize: MainAxisSize.min, // Ensure that the column only occupies the space required by its children
          children: [
            Icon(Icons.table_rows_outlined,color: Colors.teal,), // Icon widget
            SizedBox(height: 8), // Spacer between icon and text
            Text('Points table',
                style:TextStyle(
                  color: Colors.black,
                )), // Text widget
          ],
        ),// Specify the label
      ),
    ],
    ),


  );
 }
}


class Score_tab extends StatelessWidget {
 const Score_tab({super.key});
```

```
 @override
 Widget build(BuildContext context) {
   return Text(
       'Score',
       style: TextStyle(
         fontSize: 40,
       )


   );
 }
}

class Profile_tab extends StatelessWidget {
 const Profile_tab({super.key});

 @override
 Widget build(BuildContext context) {
    return  Login_form();
 }
}
```

```
class Float extends StatelessWidget {
 const Float({super.key});


 @override
 Widget build(BuildContext context) {
  return FloatingActionButton(
    onPressed: () {},
    child:Text("Add team",
      textAlign: TextAlign.center,
      style: TextStyle(
        color: Colors.black,
        fontSize: 15,
        fontFamily: 'Teko',
      ),),
    backgroundColor: Colors.teal[400],
```

```
    );
  }
}
```

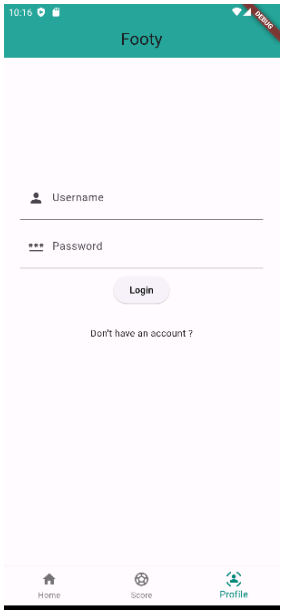
Login_form.dart:

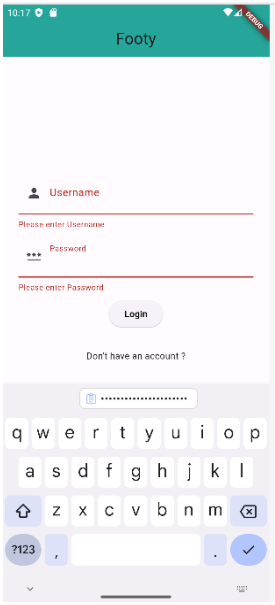
```
import 'package:flutter/material.dart';


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


  @override
  State<Login_form> createState() => _Login_formState();
}


class _Login_formState extends State<Login_form> {
  final _formKey = GlobalKey<FormState>();
  @override
  Widget build(BuildContext context) {
    return Center(
      child:Form(
      key: _formKey,
      child: Column(
```

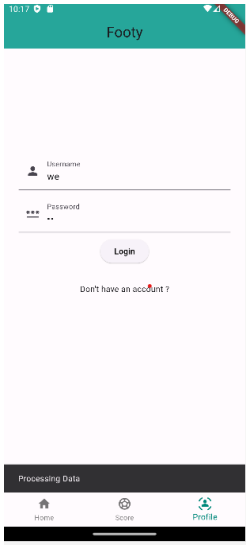
```
children: [
  SizedBox(height:150),
  TextFormField(

    decoration: InputDecoration(
      labelText: 'Username',
      prefixIcon: Icon(Icons.person),

    ),
    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Please enter Username';
      }
      return null;
    },

  ),
  SizedBox(height: 8,),
  TextFormField(
    obscureText: true,
```

```
  decoration: InputDecoration(
    labelText: 'Password',
    prefixIcon: Icon(Icons.password),

  ),
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Please enter Password';
    }
    return null;
  },

),
SizedBox(height: 8,),

ElevatedButton(
  onPressed: (){
    if (_formKey.currentState!.validate()) {
      // If the form is valid, display a snackbar. In the real world,
      // you'd often call a server or save the information in a database.
      ScaffoldMessenger.of(context).showSnackBar(
```

```
                const SnackBar(content: Text('Processing Data')),
              );
            }
          },
          child: Text("Login",
            style: TextStyle(
              color: Colors.black,
            ),
          ),
        ),
        SizedBox(height: 30,),

        Text("Don't have an account ?")
      ],
    ),
  ),
  );
 }
}
```







Conclusion: Hence, we have successfully created a login form where validation is working with the help of validator and formkey.