

## MPL Experiment 4

**Name: Rohan Lalchandani**

**Class: D15A    Roll no: 25**

AIM- To create an interactive form using form widget

### THEORY-

Forms are essential components of web and mobile applications, allowing users to input and submit data. An interactive form enhances user experience by providing real-time validation, user-friendly input fields, and seamless data handling.

A Form Widget is a structured way to manage user input, validate data, and handle submissions efficiently. It provides an interactive interface for users to enter and modify information.

### Key Features of Interactive Forms

- User Input Fields: Text fields, dropdowns, checkboxes, radio buttons, and other input elements.
- Real-time Validation: Ensures correct data format before submission.
- Error Handling: Displays messages for invalid inputs.
- Data Submission: Sends user input to a backend or local storage for further processing.
- Dynamic Updates: Auto-fills or adjusts form fields based on user selections.

### Components of Form Widget

- Form Container: Wraps all input fields.
- Input Fields: Text fields, number fields, password inputs, email inputs, etc.
- Buttons: Submit and reset buttons to process or clear input.
- Validation Mechanisms: Ensures valid input before submission.

### SYNTAX

```
Form( key: formKey, // Unique key to manage form state
child: Column( children: [ TextFormField(
decoration: InputDecoration(labelText: "Enter your name"),
```

```

validator: (value) {      if (value == null || value.isEmpty) {
return "This field cannot be empty";

      }
return null;

    },

  ),

  SizedBox(height: 10),
ElevatedButton(      onPressed: () {
if (formKey.currentState!.validate()) {

      // Perform form submission action

    }

  },

  child: Text("Submit"),

),

],

),

)

```

## Widget Properties

### 1)key

- Used to uniquely identify the Form widget.
- Typically assigned a GlobalKey<FormState> to manage validation and submissions.

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

```

Form( key: _formKey, child:
Column(  children: [ /* Form fields go
here */ ],
),
);

```

## 2)child

- Defines the content inside the Form, usually containing form fields like TextFormField, DropdownButtonFormField, etc.

Example:

```
Form(  child:
Column(
children: [
  TextFormField(),
  ElevatedButton(onPressed: () {}, child: Text("Submit")),
],
),
);
```

## 3)onchanged

- A callback function that gets triggered when any field inside the form changes.
- Can be used to update state based on form input.

Example: Form( onChanged:

```
() {  print("Form data
changed!");
},
  child: TextFormField(),
);
```

## CODE

The introduction page

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

```
'package:font_awesome_flutter/font_awesome_flutter.dart';  
import 'package:myapp/pages/createaccount.dart'; import  
'package:myapp/pages/loginpage.dart';
```

```
class TwitterLoginPage extends StatelessWidget {  
  const TwitterLoginPage({super.key});  
  
  @override  
  Widget build(BuildContext context) {    return Scaffold(  
    body: Center(      child: Padding(      padding: const  
      EdgeInsets.symmetric(horizontal: 30),      child:  
      Column(        mainAxisAlignment:  
        MainAxisAlignment.center,  
        children: [  
          // X logo at the top center      const  
      FaIcon(FontAwesomeIcons.twitter, size: 50, color: Colors.blue),  
      const SizedBox(height: 50),  
  
      // Main Text  
      Text(  
        "See what's happening in the world right now",  
        textAlign: TextAlign.center,      style:  
      GoogleFonts.roboto(      fontSize: 24,  
      fontWeight: FontWeight.bold,  
      ),  
      ),  
      const SizedBox(height: 30),  
  
      // Continue with Google Button
```

```

        // Create Account Button

ElevatedButton(
  onPressed: () {
    Navigator.push(
      context,
        MaterialPageRoute(builder: (context) => const CreateAccount()),
      );
  },
  style: ElevatedButton.styleFrom(
    backgroundColor: Colors.blue,      foregroundColor:
Colors.white,      padding: const EdgeInsets.symmetric(vertical: 15,
horizontal: 50),      shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(30),
      ),
    ),
    child: const Text("Create account"),
  ),
  const SizedBox(height: 20),

  // Already have an account? Log
in      TextButton(
onPressed: () {
      Navigator.push(context,
        MaterialPageRoute(builder: (context) => const LoginPage()),
      );
    },
    child: const Text(
      "Have an account already? Log in",
      style: TextStyle(color: Colors.blue),

```

```

        ),
    ),
],
),
),
),
);
}
}

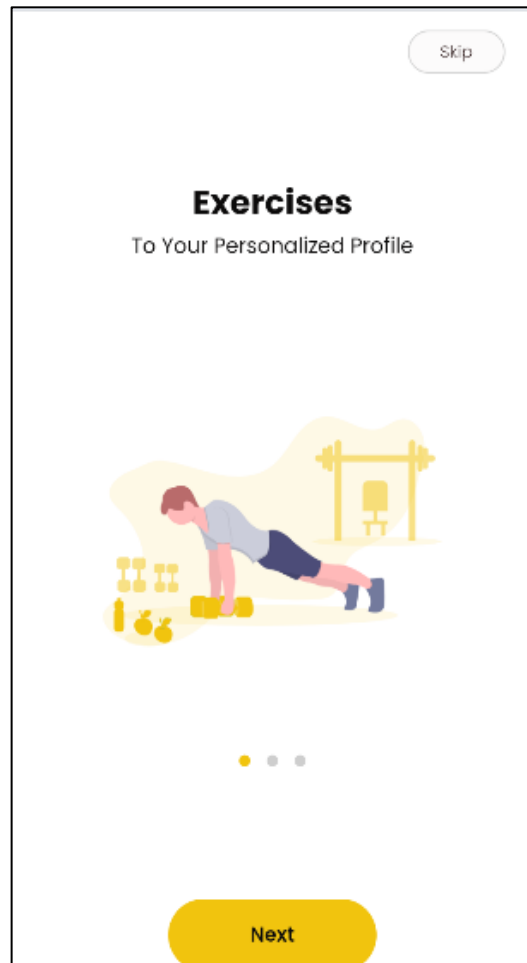
```

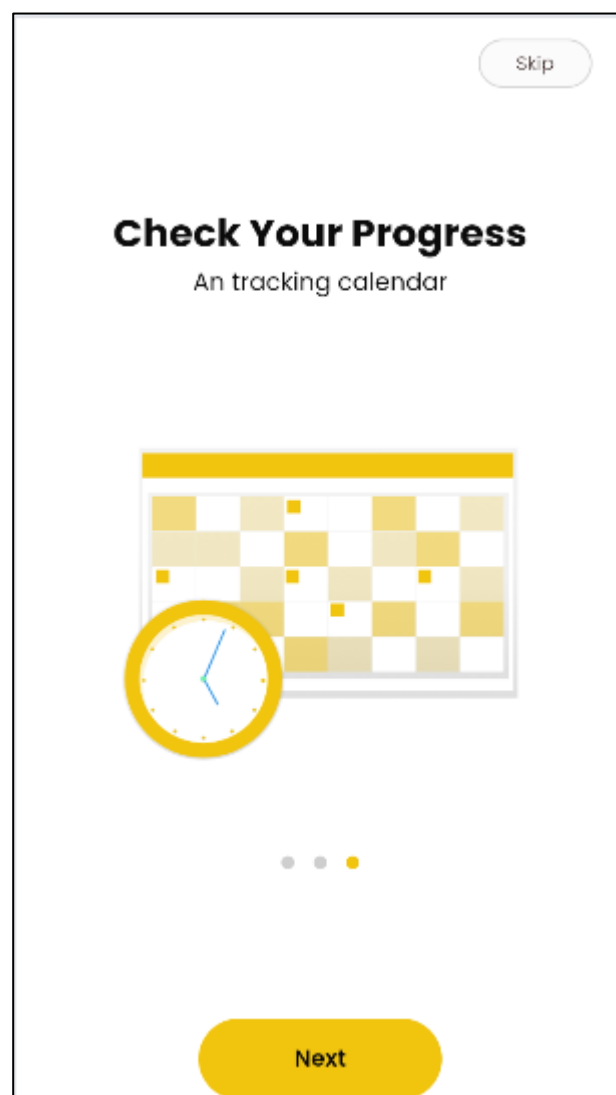
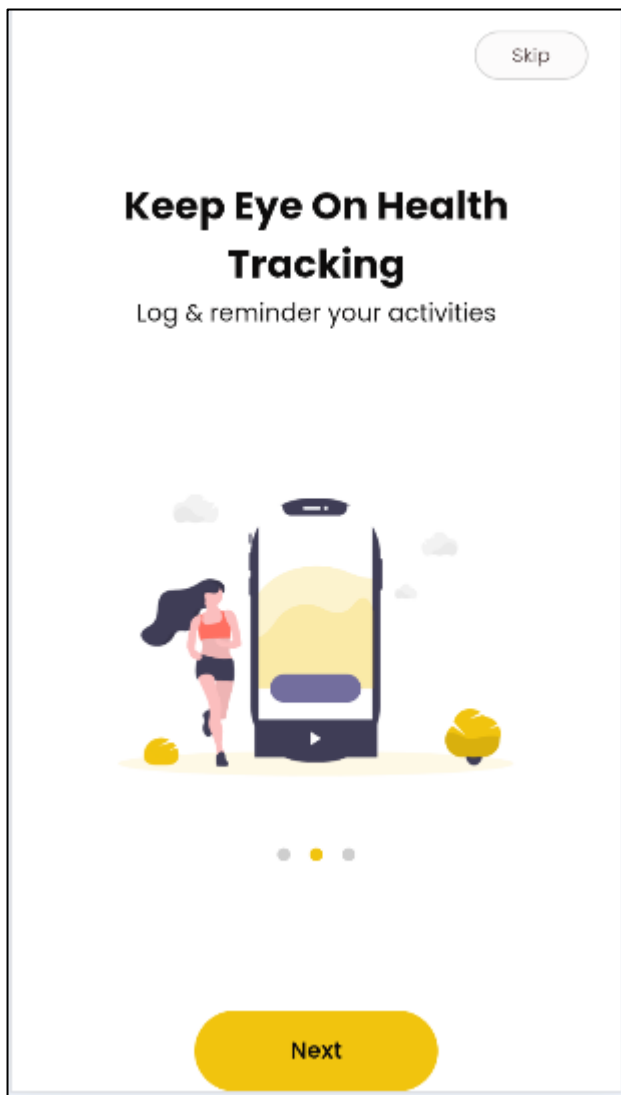
OUTPUT :

### Splash Screen



### Onboarding Screen





Create Account Page

CODE

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

```

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

  @override
  _CreateAccountState createState() => _CreateAccountState();
}

```

```

class _CreateAccountState extends State<CreateAccount>
{
  final _formKey = GlobalKey<FormState>();
  final _usernameController = TextEditingController();
  final _emailController = TextEditingController();
  final _dobController = TextEditingController();
  final _passwordController = TextEditingController();
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      leading: IconButton(
        icon: const Icon(Icons.arrow_back, color: Colors.black),
        onPressed: () {
          Navigator.pop(context);
        },
      ),
    centerTitle: true,
    title: const FaIcon(FontAwesomeIcons.twitter,
      color: Colors.blue, size: 30),
    backgroundColor: Colors.transparent,
    elevation: 0,
  ),
  body: Padding(
    padding: const EdgeInsets.all(20),
    child: Form(
      key:

```



```

    _formKey,      child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
      children: [      // Username      const Text(
"Username",      style: TextStyle(fontSize: 16, fontWeight:
FontWeight.bold),
      ),
      TextFormField(      controller: _usernameController,
decoration: InputDecoration(      hintText: "Enter your username",
border: OutlineInputBorder(borderRadius: BorderRadius.circular(10)),
      ),
      validator: (value) {      if
(value == null || value.isEmpty) {
return "Username cannot be empty";
      }
      return null;
      },
      ),
      const SizedBox(height: 15),

      // Email      const Text(      "Email",      style:
TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
      ),
      TextFormField(      controller: _emailController,
decoration: InputDecoration(      hintText: "Enter your email",
border: OutlineInputBorder(borderRadius: BorderRadius.circular(10)),
      ),
      validator: (value) {      if
(value == null || value.isEmpty) {
return "Email cannot be empty";      }

```

```

else if (!value.contains("@")) {
return "Enter a valid email with @";

}

return null;

},

),

const SizedBox(height: 15),

// Date of Birth
const Text(
    "Date of
    Birth",
    style:
    TextStyle(fontSize: 16,
    fontWeight:
    FontWeight.bold),
    ),
    TextFormField(
        controller: _dobController,
        decoration: InputDecoration(
            hintText: "DD/MM/YYYY",
            border: OutlineInputBorder(borderRadius: BorderRadius.circular(10)),
        ),
        keyboardType: TextInputType.datetime,
        validator: (value) {
final dobPattern = RegExp(r"^(0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[0-2])^\d{4}$");
if (value == null || value.isEmpty) {
return "Date of birth cannot be empty";
} else if (!dobPattern.hasMatch(value)) {
return "Enter a valid date in
DD/MM/YYYY format";
}
}

return null;

},

),

const SizedBox(height: 15),

```

```

        // Password
        const Text(
            "Password",
            style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
        ),
        TextFormField(
            controller: _passwordController,
            obscureText: true,
            decoration:
            InputDecoration(
                hintText:
                "Enter your password",
                border:
                OutlineInputBorder(borderRadius:
                BorderRadius.circular(10)),
            ),
            validator: (value) {
                if (value == null ||
                value.isEmpty) {
                    return "Password cannot be
                    empty";
                } else if (value.length < 6) {
                    return "Password must be at least 6 characters";
                }
            }
        ),
        return null;
    },
),
],
),
),
),
),

// Floating Action Button for "Next"
floatingActionButton: Padding(
padding: const EdgeInsets.only(bottom: 30), // Adjust to move the button up
child: FloatingActionButton(
    onPressed: () {
        if
        (_formKey.currentState!.validate()) {
            print("Username:
            ${_usernameController.text}");
            print("Email:

```

```

$_emailController.text}");          print("DOB: $_dobController.text}");
print("Password: $_passwordController.text}");


    }
  },
  backgroundColor: Colors.blue,      child: const
Icon(Icons.arrow_forward, color: Colors.white),
  ),
  ),
  floatingActionButtonLocation: FloatingActionButtonLocation.endFloat,
);}}


```


## OUTPUT


**Create Account**

**ealtho**

 Full Name

 Email

 Password

 Confirm Password

**SIGN UP**

Already have an account? [Login](#)

## Login Page

### CODE

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

```
class LoginPage extends StatefulWidget {  
  const LoginPage({super.key});  
  
  @override  
  _LoginPageState createState() => _LoginPageState();  
}
```

```
class _LoginPageState extends State<LoginPage> {  
  final _formKey = GlobalKey<FormState>(); final  
  _emailController = TextEditingController(); final  
  _passwordController = TextEditingController();
```

```
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        leading: IconButton(  
          icon: const  
            Icon(Icons.arrow_back),  
          onPressed:  
            () {  
              Navigator.pop(context);  
            },  
        ),  
      ),  
      body: Padding(  
        padding: const  
          EdgeInsets.symmetric(horizontal: 30),  
        child: Form(  

```

```

key: _formKey,      child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    // Twitter
    Logo      Center(
child: FaIcon(
  FontAwesomeIcons.twitter,
  size: 50,
color: Colors.blue,
  ),
  ),
  const SizedBox(height: 40),

  // Email TextField      const Text("Email",
style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
  ),
  TextFormField(      controller:
_emailController,      decoration:
InputDecoration(      border:
OutlineInputBorder(      borderRadius:
BorderRadius.circular(10),
  ),
  hintText: "Enter your email",
  ),
  validator: (value) {      if
(value == null || value.isEmpty) {
return "Email cannot be empty";      }
else if (!value.contains("@")) {
return "Enter a valid email with @";

```

```

    }
return null;

    },

),

const SizedBox(height: 20),

    // Password TextField    const Text(    "Password",
style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
    ),

    TextFormField(    controller:
_passwordController,    obscureText: true,
decoration: InputDecoration(    border:
OutlineInputBorder(    borderRadius:
BorderRadius.circular(10),
    ),
    hintText: "Enter your password",
    ),
    validator: (value) {    if
(value == null || value.isEmpty) {
return "Password cannot be empty";
} else if (value.length < 6) {    return
"Password must be at least 6 characters";
    }
return null;

    },

),

const SizedBox(height: 40),

    // Next Button aligned at bottom-right    Align(
alignment: Alignment.bottomRight,    child:

```

```


ElevatedButton(
  onPressed: () {
    if
    (_formKey.currentState!.validate()) {
      // Form is
      valid, proceed with login logic
      print("Email:
      ${_emailController.text}");
      print("Password:
      ${_passwordController.text}");
    }
  },
  style: ElevatedButton.styleFrom(
    backgroundColor:
    Colors.blue,
    foregroundColor: Colors.white,
    padding: const EdgeInsets.symmetric(vertical: 15, horizontal: 30),
    shape: RoundedRectangleBorder(
      borderRadius:
      BorderRadius.circular(30),
    ),
  ),
  child: const Text("Next"),
),
const SizedBox(height: 30),
],
),
),
),
);
}
}


```




## OUTPUT:

Login to your account

ealtho

 Email

 Password

[Forgot Password?](#)

LOGIN

Don't have an account? [Sign Up](#)