MAD AND PWA LAB

Name: Harshita Dubey Class: D15A

Roll no:14

Aim: To create an interactive Form using form widget.

THEORY:

In Flutter, a form is a collection of interactive widgets that allows users to input data and submit it for processing. The Form widget is a crucial component for handling user input and validation efficiently.

Form Widget:

The Form widget is the container for form elements in Flutter.

It provides a way to organize and manage form fields, as well as handle their validation and submission.

FormField Widget:

The FormField widget is a wrapper for individual form fields within a Form. It encapsulates the logic for managing the state of the form field, handling user input, and triggering validation.

TextFormField:

The TextFormField widget is a specialized form field for handling text input. It comes with built-in features for managing text editing, validation, and error handling.

FormState:

Each Form has an associated FormState that holds the current state of the form.

The FormState is automatically created by the framework and can be used to interact with and manage the form programmatically.

Validation:

Flutter provides a robust validation mechanism through the Form and FormField widgets.

You can define validation logic for each form field, and the framework takes care of displaying error messages when necessary.

Form Submission:

Form submission is often triggered by a button press, such as a "Submit" button.

You can use the onPressed callback of a button to trigger the form submission, and within that callback, you can access the FormState to check if the form is valid before proceeding with submission.

Handling Form Reset:

To reset the form to its initial state, you can use the reset method provided by the FormState.

CODE:

Steps:

1. Main.dart

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

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

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

  @override
  Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Flutter Demo',
    theme: ThemeData(),
    home:const LoginScreen(),
   );
  }
}
```

Step 2: In the Folder screens made previously, add new file login.dart

```
import 'package:flutter/material.dart';
import 'package:my app/screens/home.dart';
import 'signup.dart'; // Import the file where SignupScreen is defined
class LoginScreen extends StatefulWidget {
 const LoginScreen({Key? key}) : super(key: key);
 @override
 State<LoginScreen> createState() => LoginScreenState();
}
class LoginScreenState extends State<LoginScreen> {
 final emailController = TextEditingController();
 final passwordController = TextEditingController();
 @override
 Widget build(BuildContext context) {
  return Scaffold(
   body: SafeArea(
    child: Padding(
      padding: const EdgeInsets.all(20.0),
      child: Column(
       mainAxisAlignment: MainAxisAlignment.center,
       children: [
        Center(
          child: Image.asset(
           'assets/thread logo.png',
           width: 80,
         ),
        ),
        Padding(
          padding: const EdgeInsets.only(top: 20.0),
          child: TextFormField(
           controller: emailController,
           decoration: const InputDecoration(
            contentPadding: EdgeInsets.all(8),
            hintText: 'Enter your email',
            border: OutlineInputBorder(
             borderSide: BorderSide(
               color: Colors.grey,
             ),
            ),
            focusedBorder: OutlineInputBorder(
```

```
borderSide: BorderSide(
               color: Colors.grey,
              ),
            ),
          ),
        Padding(
          padding: const EdgeInsets.only(top: 20.0),
          child: TextFormField(
           controller: passwordController,
           decoration: const InputDecoration(
            contentPadding: EdgeInsets.all(8),
            hintText: 'Enter your password',
            border: OutlineInputBorder(
             borderSide: BorderSide(
               color: Colors.grey,
             ),
            ),
            focusedBorder: OutlineInputBorder(
              borderSide: BorderSide(
               color: Colors.grey,
              ),
            ),
           ),
          ),
         Row(
          mainAxisAlignment: MainAxisAlignment.end,
          children: [
           TextButton(
            onPressed: () {},
            child: const Text(
              'Forgot password',
              style: TextStyle(
               color: Colors.black,
              ),
            ),
         ],
         ),
         SizedBox(
          width: double.infinity,
          height: 42,
          child: ElevatedButton(
           onPressed: () {
            Navigator.push(context, MaterialPageRoute(builder: (context)=> const
Home()));
```

```
},
  style: ElevatedButton.styleFrom(backgroundColor: Colors.black),
  child: const Text("Login"),
 ),
),
const Divider(),
const Spacer(),
Row(
 mainAxisAlignment: MainAxisAlignment.center,
 children: [
  const Text("Don't have an account yet? "),
  TextButton(
   onPressed: () {
    // Use SignupScreen() constructor to create an instance
     Navigator.push(
      context,
      MaterialPageRoute(
       builder: (context) => SignupScreen(),
      ),
    );
   },
   child: const Text(
     "Sign up",
     style: TextStyle(
      fontWeight: FontWeight.bold,
      color: Colors.black,
    ),
  ),
```

Step 3: make a folder named assets and add the image threads_logo.png. Now add the path of the image file in the pubspec.yaml and then run it using flutter run.

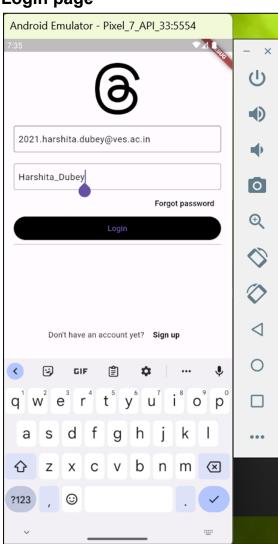
Step 4: make a file in the screens folder named signup.dart

```
import 'package:flutter/material.dart';
import 'package:my app/screens/login.dart';
// import 'signup_screen.dart'; // Import the file where SignupScreen is defined
class SignupScreen extends StatefulWidget {
 const SignupScreen({Key? key}) : super(key: key);
 @override
 State<SignupScreen> createState() => _SignupScreenState();
}
class SignupScreenState extends State<SignupScreen> {
 final emailController = TextEditingController();
 final passwordController = TextEditingController();
 final nameController = TextEditingController();
 final usernameController = TextEditingController();
 @override
 Widget build(BuildContext context) {
  return Scaffold(
   body: SafeArea(
    child: Padding(
      padding: const EdgeInsets.all(20.0),
      child: Column(
       mainAxisAlignment: MainAxisAlignment.center,
       children: [
        Center(
          child: Image.asset(
           'assets/thread logo.png',
           width: 80,
         ),
        ),
        Padding(
          padding: const EdgeInsets.only(top: 20.0),
          child: TextFormField(
           controller: emailController,
           decoration: const InputDecoration(
            contentPadding: EdgeInsets.all(8),
            hintText: 'Enter your email',
            border: OutlineInputBorder(
             borderSide: BorderSide(
```

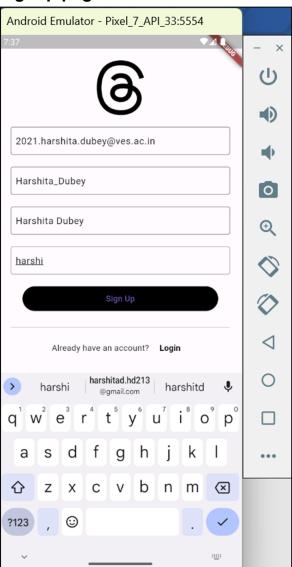
```
color: Colors.grey,
     ),
   ),
   focusedBorder: OutlineInputBorder(
     borderSide: BorderSide(
      color: Colors.grey,
     ),
   ),
Padding(
 padding: const EdgeInsets.only(top: 20.0),
 child: TextFormField(
  controller: passwordController,
  decoration: const InputDecoration(
   contentPadding: EdgeInsets.all(8),
   hintText: 'Enter your password',
   border: OutlineInputBorder(
    borderSide: BorderSide(
      color: Colors.grey,
    ),
   ),
   focusedBorder: OutlineInputBorder(
     borderSide: BorderSide(
      color: Colors.grey,
     ),
   ),
       Padding(
padding: const EdgeInsets.only(top: 20.0),
child: TextFormField(
 controller: nameController,
 decoration: const InputDecoration(
  contentPadding: EdgeInsets.all(8),
  hintText: 'Enter your full name',
  border: OutlineInputBorder(
   borderSide: BorderSide(
     color: Colors.grey,
   ),
  ),
  focusedBorder: OutlineInputBorder(
   borderSide: BorderSide(
     color: Colors.grey,
   ),
  ),
```

```
),
 ),
),
Padding(
 padding: const EdgeInsets.only(top: 20.0),
 child: TextFormField(
  controller: usernameController,
  decoration: const InputDecoration(
    contentPadding: EdgeInsets.all(8),
   hintText: 'Enter your username',
    border: OutlineInputBorder(
     borderSide: BorderSide(
      color: Colors.grey,
     ),
   ),
   focusedBorder: OutlineInputBorder(
     borderSide: BorderSide(
      color: Colors.grey,
     ),
Padding(
 padding: const EdgeInsets.all(20.0),
 child: SizedBox(
   width: double.infinity,
   height: 42,
    child: ElevatedButton(
     onPressed: () {},
     style: ElevatedButton.styleFrom(backgroundColor: Colors.black),
     child: const Text("Sign Up"),
   ),
  ),
),
 const Divider(),
 const Spacer(),
 Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    const Text("Already have an account? "),
    TextButton(
     onPressed: () {
      // Use SignupScreen() constructor to create an instance
      Navigator.push(
       context,
       MaterialPageRoute(
         builder: (context) => const LoginScreen(),
```

OUTPUT: Login page



Signup page



CONCLUSION:

I have completed with making signup and login page from which I gained useful insights to how to use form widgets like TextFormField, Onpressed etc. I have also used ElevatedButton.