

MPL EXPERIMENT 4

NAME-ARYAN PATANKAR

CLASS-D15A

ROLL NO-33

AIM: To create an interactive Form using form widget.

Theory:

1. Introduction

Forms are a crucial component in mobile applications for collecting user input. Flutter provides the **Form** widget, which helps manage multiple input fields efficiently and allows validation before submission.

The **Form** widget works in combination with **TextFormField** and a **GlobalKey<FormState>** to track the form state, validate inputs, and process the collected data.

2. Form Widget Overview

The **Form** widget in Flutter acts as a container for form fields and manages their state. It provides built-in validation and submission handling.

Key Components of a Form:

1. Form Widget:

- Wraps multiple input fields (TextFormFields) together.
- Requires a *GlobalKey<FormState>* to track and validate form fields.

2. TextFormField Widget:

- A special type of text field used in forms.
- Supports validation and user input.

3. Validator Function:

- Ensures correct user input and prevents invalid data.
- Example: Ensuring an email field contains a valid email format.

4. Submit Button:

- Triggers form validation and submission.

3. Working Principle of Form widget

The **Form** widget allows developers to:

- Group multiple **TextFormField** widgets together.
- Validate input before proceeding.
- Save user input after validation.

The **GlobalKey<FormState>** is used to control the form, providing methods like:

- *formKey.currentState!.validate()* → Checks all validators in the form fields.
- *formKey.currentState!.save()* → Saves the form input values.
- *formKey.currentState!.reset()* → Clears the form fields.

CODE:

Sign_in_screen.dart

```
// lib/features/auth/screens/sign_in_screen.dart

import 'package:flutter/material.dart';

import 'package:spotify_clone/features/auth/widgets/social_login_button.dart';

class SignInScreen extends StatefulWidget {

  const SignInScreen({super.key});

  @override

  State<SignInScreen> createState() => _SignInScreenState();

}
```

```
class _SignInScreenState extends State<SignInScreen> {  
  
  bool _isPasswordVisible = false;  
  
  final TextEditingController _emailController = TextEditingController();  
  
  final TextEditingController _passwordController = TextEditingController();
```

```
@override
```

```
void dispose() {  
  
  _emailController.dispose();  
  
  _passwordController.dispose();  
  
  super.dispose();  
  
}
```

```
@override
```

```
Widget build(BuildContext context) {  
  
  return Scaffold(  
  
    backgroundColor: Colors.white,  
  
    body: SafeArea(  
  
      child: SingleChildScrollView(  
  
        padding: const EdgeInsets.symmetric(horizontal: 24.0),  
  
        child: Column(  
  
          crossAxisAlignment: CrossAxisAlignment.start,  
  
          children: [  
  

```

```
const SizedBox(height: 20),

IconButton(

  icon: const Icon(Icons.arrow_back),

  onPressed: () {

    Navigator.pop(context);

  },

),

const SizedBox(height: 20),

Center(

  child: Image.asset(

    'assets/images/spotify_logo.png',

    height: 200,

  ),

),

const SizedBox(height: 40),

const Center(

  child: Text(

    'Sign In',

    style: TextStyle(

      fontSize: 28,

      fontWeight: FontWeight.bold,

    ),
```

```
    ),  
  ),  
  const SizedBox(height: 16),  
  Center(  
    child: RichText(  
      text: TextSpan(  
        text: 'If You Need Any Support ',  
        style: const TextStyle(color: Colors.black),  
        children: [  
          TextSpan(  
            text: 'Click Here',  
            style: TextStyle(color: Theme.of(context).primaryColor),  
          ),  
        ],  
      ),  
    ),  
  ),  
  const SizedBox(height: 40),  
  TextField(  
    controller: _emailController,  
    decoration: InputDecoration(  
      hintText: 'Enter Username Or Email',
```

```
        filled: true,
        fillColor: Colors.grey[100],
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(8),
          borderSide: BorderSide.none,
        ),
        contentPadding: const EdgeInsets.all(16),
      ),
    ),
    const SizedBox(height: 16),
    TextField(
      controller: _passwordController,
      obscureText: !_passwordVisible,
      decoration: InputDecoration(
        hintText: 'Password',
        filled: true,
        fillColor: Colors.grey[100],
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(8),
          borderSide: BorderSide.none,
        ),
        contentPadding: const EdgeInsets.all(16),
```

```

suffixIcon: IconButton(
  icon: Icon(
    _isPasswordVisible
      ? Icons.visibility
      : Icons.visibility_off,
    color: Colors.grey,
  ),
  onPressed: () {
    setState(() {
      _isPasswordVisible = !_isPasswordVisible;
    });
  },
),
),
),
const SizedBox(height: 8),
TextButton(
  onPressed: () {},
  child: const Text(
    'Recovery Password',
    style: TextStyle(color: Colors.black),
  ),
),

```

```
),  
  
const SizedBox(height: 24),  
  
ElevatedButton(  
  
  onPressed: () {},  
  
  style: ElevatedButton.styleFrom(  
  
    backgroundColor: const Color(0xFF1DB954),  
  
    minimumSize: const Size(double.infinity, 50),  
  
    shape: RoundedRectangleBorder(  
  
      borderRadius: BorderRadius.circular(25),  
  
    ),  
  
  ),  
  
  child: const Text(  
  
    'Sign In',  
  
    style: TextStyle(  
  
      color: Colors.white,  
  
      fontSize: 16,  
  
    ),  
  
  ),  
  
),  
  
const SizedBox(height: 24),  
  
const Row(  
  
  children: [
```







```
Expanded(child: Divider()),
Padding(
  padding: EdgeInsets.symmetric(horizontal: 16),
  child: Text('Or'),
),
Expanded(child: Divider()),
],
),
const SizedBox(height: 24),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    SocialLoginButton(
      assetPath: 'assets/images/google_logo.png',
      onTap: () {},
    ),
    SocialLoginButton(
      assetPath: 'assets/images/apple_logo.png',
      onTap: () {},
    ),
  ],
),
```

```
const SizedBox(height: 24),
Center(
  child: RichText(
    text: TextSpan(
      text: 'Not A Member ? ',
      style: const TextStyle(color: Colors.black),
      children: [
        TextSpan(
          text: 'Register Now',
          style: TextStyle(color: Theme.of(context).primaryColor),
        ),
      ],
    ),
  ),
),
),
);
}
```

OUTPUT


21:10

   12%




Sign In


If You Need Any Support [Click Here](#)



Recovery Password

Or





Not A Member ? [Register Now](#)

