

Experiment 4

Aim: To create an interactive Form using a form widget.

Theory:

Form Widget:

- In Flutter, a Form widget is used to group multiple form fields together.
- It provides functionality for managing form state, validation, and submission.
- The Form widget keeps track of the form's state using a `FormState` object.

Form Fields:

- Form fields represent input elements within the form, such as text fields, checkboxes, radio buttons, dropdowns, etc.
- Each form field typically has properties for input validation and handling user input changes.

Form State Management:

- The `FormState` object is responsible for managing the state of a form.
- It provides methods for validating the form, saving form data, resetting the form, and more.

User Interaction:

- User interaction with the form involves entering data into form fields, interacting with input elements, and submitting the form.

Code:

login_page.dart:

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import
'package:flutter_firebase/features/user_auth/presentation/pages/sign_up_page.dart';
import
'package:flutter_firebase/features/user_auth/presentation/widgets/form_container_wid
get.dart';
import 'package:flutter_firebase/global/common/toast.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:google_sign_in/google_sign_in.dart';
import 'package:firebase_app_check/firebase_app_check.dart';
// import 'package:firebase_app_check_debug/firebase_app_check_debug.dart';
```

```

import '../firebase_auth_implementation/firebase_auth_services.dart';

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

  @override
  State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  bool _isSigning = false;
  final FirebaseAuthService _auth = FirebaseAuthService();
  final FirebaseAuth _firebaseAuth = FirebaseAuth.instance;
  TextEditingController _emailController = TextEditingController();
  TextEditingController _passwordController = TextEditingController();

  // @override
  // void initState() {
  //   super.initState();
  //   // Initialize Firebase App Check
  //   FirebaseAppCheck.instance.installAppCheckProviderFactory(DebugAppCheckProviderFactory());
  // }

  @override
  void dispose() {
    _emailController.dispose();
    _passwordController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.black,
        automaticallyImplyLeading: false,
      ),
      body: Container(
        color: Colors.black, // Set the background color to white
        padding: EdgeInsets.symmetric(horizontal: 15),

```

```

child: Column(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Text(
      "Sign In",
      style: TextStyle(
        fontSize: 27,
        fontWeight: FontWeight.bold,
        color: Colors.white),
    ),
    SizedBox(height: 10), // Add SizedBox for spacing
    Padding(
      padding: EdgeInsets.symmetric(horizontal: 32), // Adjust the left padding as
needed
      child: Center(
        child: Text(
          "By using our services you are agreeing to our Terms and Privacy Statement",
          textAlign: TextAlign.center,
          style: TextStyle(
            fontSize: 16,
            color: Colors.white,
          ),
        ),
      ),
    ),
    ),
    SizedBox(height: 70),
    FormContainerWidget(
      controller: _emailController,
      hintText: "Email",
      isPasswordField: false,
    ),
    SizedBox(height: 10),
    FormContainerWidget(
      controller: _passwordController,
      hintText: "Password",
      isPasswordField: true,
    ),
    SizedBox(height: 30),
    GestureDetector(
      onTap: () {
        _signIn();
      },
      child: Container(
        width: double.infinity,

```

```
height: 45,  
decoration: BoxDecoration(  
  color: Colors.black,  
  borderRadius: BorderRadius.circular(2),  
  border: Border.all(color: Colors.white),  
) ,  
child: Center(  
  child: _isSigning  
    ? CircularProgressIndicator(color: Colors.white)  
    : Text(  
      "Sign In",  
      style: TextStyle(  
        color: Colors.white,  
        fontSize: 18.0,  
        fontWeight: FontWeight.bold,  
      ),  
    ),  
  ),  
) ,  
) ,  
) ,  
) ,  
) ,  
),  
SizedBox(height: 15),  
GestureDetector(  
  onTap: () {  
    _signInWithGoogle();  
  },  
  child: Container(  
    width: double.infinity,  
    height: 45,  
    decoration: BoxDecoration(  
      color: Colors.black,  
      borderRadius: BorderRadius.circular(2),  
      border: Border.all(color: Colors.white),  
    ),  
    child: Center(  
      child: Row(  
        mainAxisAlignment: MainAxisAlignment.center,  
        children: [  
          Icon(FontAwesomeIcons.google, color: Colors.white),  
          SizedBox(width: 10),  
          Text(  
            "Sign in with Google",  
            style: TextStyle(  
              color: Colors.white,  
              fontSize: 18.0,
```

[illegible]

```

}

void _signIn() async {
  setState() {
    _isSigning = true;
  });

  String email = _emailController.text;
  String password = _passwordController.text;

  User? user = await _auth.signInWithEmailAndPassword(email, password);

  setState() {
    _isSigning = false;
  });

  if (user != null) {
    showToast(message: "User is successfully signed in");
    Navigator.pushNamed(context, "/home");
  } else {
    showToast(message: "Some error occurred");
  }
}

_signInWithGoogle() async {
  final GoogleSignIn _googleSignIn = GoogleSignIn();

  try {
    final GoogleSignInAccount? googleSignInAccount =
      await _googleSignIn.signIn();

    if (googleSignInAccount != null) {
      final GoogleSignInAuthentication googleSignInAuthentication =
        await googleSignInAccount.authentication;

      final AuthCredential credential = GoogleAuthProvider.credential(
        idToken: googleSignInAuthentication.idToken,
        accessToken: googleSignInAuthentication.accessToken,
      );

      await _firebaseAuth.signInWithCredential(credential);
      Navigator.pushNamed(context, "/home");
    }
  } catch (e) {

```

```

        showToast(message: "Some error occurred $e");
    }
}

```

```

sign_up_page.dart:
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import
'package:flutter_firebase/features/user_auth/firebase_auth_implementation/firebase_a
uth_services.dart';
import
'package:flutter_firebase/features/user_auth/presentation/pages/login_page.dart';
import
'package:flutter_firebase/features/user_auth/presentation/widgets/form_container_wid
get.dart';
import 'package:flutter_firebase/global/common/toast.dart';

```

```

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

  @override
  State<SignUpPage> createState() => _SignUpPageState();
}

```

```

class _SignUpPageState extends State<SignUpPage> {
  final FirebaseAuthService _auth = FirebaseAuthService();

  TextEditingController _usernameController = TextEditingController();
  TextEditingController _emailController = TextEditingController();
  TextEditingController _passwordController = TextEditingController();

```

```

  bool isSigningUp = false;

```

```

  @override
  void dispose() {
    _usernameController.dispose();
    _emailController.dispose();
    _passwordController.dispose();
    super.dispose();
  }

```

```

  @override

```

```

Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      automaticallyImplyLeading: false,
      backgroundColor: Colors.black,
    ),
    backgroundColor: Colors.black,
    body: Center(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 15),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              "Create an account",
              style: TextStyle(
                fontSize: 27,
                fontWeight: FontWeight.bold,
                color: Colors.white),
            ),
            SizedBox(
              height: 20,
            ),
            Text(
              "By clicking on 'Create account' you are agreeing to our Terms of Use and Privacy Policy",
              textAlign: TextAlign.center,
              style: TextStyle(fontSize: 16, color: Colors.white),
            ),
            SizedBox(
              height: 80,
            ),
            FormContainerWidget(
              controller: _usernameController,
              hintText: "Username",
              isPasswordField: false,
            ),
            SizedBox(
              height: 10,
            ),
            FormContainerWidget(
              controller: _emailController,
              hintText: "Email",
              isPasswordField: false,

```



```

),
  SizedBox(
    height: 10,
  ),
  FormContainerWidget(
    controller: _passwordController,
    hintText: "Password",
    isPasswordField: true,
  ),
  SizedBox(
    height: 30,
  ),
  GestureDetector(
    onTap: () {
      _signUp();
    },
    child: Container(
      width: double.infinity,
      height: 45,
      decoration: BoxDecoration(
        color: Colors.white,
        borderRadius: BorderRadius.circular(10),
      ),
      child: Center(
        child: isSigningUp
          ? CircularProgressIndicator(
              color: Colors.white,
            )
          : Text(
              "Create account",
              style: TextStyle(
                color: Colors.black,
                fontWeight: FontWeight.bold,
                fontSize: 18.0,
              ),
            ),
      ),
    ),
  ),
  SizedBox(
    height: 80,
  ),
  Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Text(

```

```

        "Have an account?",
        style: TextStyle(
          color: Colors.white,
          fontSize: 18.0),
      ),
      SizedBox(
        width: 5,
      ),
      GestureDetector(
        onTap: () {
          Navigator.pushAndRemoveUntil(
            context,
            MaterialPageRoute(
              builder: (context) => LoginPage(),
              (route) => false);
        },
        child: Text(
          "    Sign In",
          style: TextStyle(
            color: Colors.purpleAccent, fontWeight: FontWeight.bold, fontSize: 18.0),
        ))
      ),
    ),
  ),
),
),
);
}

```

```

void _signUp() async {
  setState() {
    isSigningUp = true;
  });
}

```

```

String username = _usernameController.text;
String email = _emailController.text;
String password = _passwordController.text;

```

```

User? user = await _auth.signUpWithEmailAndPassword(email, password);

```

```

setState() {
  isSigningUp = false;
});

```

```
if (user != null) {  
  showToast(message: "User is successfully created");  
  Navigator.pushNamed(context, "/home");  
} else {  
  showToast(message: "Some error happend");  
}  
}  
}
```

Output:

