

Name - Gaurang A Raorane  
Roll no - 49

Div - D15A  
Batch - C

## **Experiment - 4**

**Aim :-** To create an interactive Form using form widget

### **Theory:-**

Forms are essential components of user interaction in mobile and web applications, allowing users to input data, make selections, and submit information. In Flutter, the Form widget serves as the foundation for creating interactive and validated forms. Let's explore the theory behind creating an interactive form using the Form widget:

#### **Form Widget:**

- The Form widget in Flutter provides a container for form fields and facilitates form validation and submission.
- It maintains the state of form fields and manages their validation status.
- The Form widget should be placed as the parent widget of form fields within the widget tree.

#### **Form Fields:**

- Form fields represent input elements within the form, such as text fields, checkboxes, radio buttons, dropdowns, etc.
- Flutter provides various built-in form field widgets like TextFormField, Checkbox, Radio, DropdownButton, etc.
- Each form field widget must be wrapped inside a FormField widget to register with the Form and participate in form validation.

#### **Validation:**

- Form validation ensures that user input meets specified criteria or constraints before submitting the form.
- Flutter offers built-in validation mechanisms for form fields using validators.
- Validators are callback functions applied to form fields to validate input data. They return an error message if validation fails.
- Validators can be synchronous or asynchronous, depending on the complexity of validation logic.

#### **Submission:**

- Once the form is filled out and validated, it can be submitted to process the user input.
- Form submission involves handling the data collected from form fields and performing any necessary actions, such as saving data to a database, sending it to a server, etc.
- Submission can be triggered by a button press or any other user action.

## LoginScreen.dart

```
import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:osiris/Services/auth.dart';
import 'package:osiris/Services/consts.dart';
import 'package:provider/provider.dart';

import 'package:flutter_svg/flutter_svg.dart';

class LoginScreen extends StatelessWidget {
  const LoginScreen({super.key, Key? key1});

  @override
  Widget build(BuildContext context) {
    var size = MediaQuery.of(context).size;
    TextEditingController emailController = TextEditingController();
    TextEditingController passwordController = TextEditingController();

    return Scaffold(
      body: Container(
        color: background_primary,
        child: SafeArea(
          child: SingleChildScrollView(
            child: Container(
              color: background_primary,
              height: size.height,
              width: size.width,
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  Container(
                    height: size.height * 0.3,
                  ),
                  SvgPicture.asset(
                    "assets/netflix.svg",
                    width: size.width * 0.60,
                  ),
                  Padding(
                    padding: const EdgeInsets.all(24.0),
                    child: Column(
                      children: [
                        TextFormField(
                          controller: emailController,
```

```

decoration: const InputDecoration(
  labelText: 'Email',
  filled: true,
  fillColor: Colors.white,
),
),
const SizedBox(height: 20),
TextFormField(
  controller: passwordController,
  decoration: const InputDecoration(
    labelText: 'Password',
    filled: true,
    fillColor: Colors.white,
  ),
  obscureText: true,
),
const SizedBox(height: 20),
ElevatedButton(
  onPressed: () async {
    try {
      await FirebaseAuth.instance
        .signInWithEmailAndPassword(
          email: emailController.text.trim(),
          password: passwordController.text.trim(),
        );
      // Authentication successful, navigate to next screen
    } catch (e) {
      // Handle authentication failure
      print('Error: $e');
    }
  },
  child: const Text('Sign In with Email'),
),
const SizedBox(height: 20),
ElevatedButton.icon(
  onPressed: () {
    final provider = Provider.of<GoogleSignInProvider>(
      context,
      listen: false);
    provider.googleLogin();
  },
  icon: const Falcon(
    FontAwesomeIcons.google,
    color: Colors.white,
  ),
),

```

```

        ),
        label: const Text(
          'Continue with Google',
          style: TextStyle(color: Colors.white),
        ),
        style: ElevatedButton.styleFrom(
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(14)),
            backgroundColor: const Color(0xFF2A292F)),
      ),
    ],
  ),
  Container(
    margin: const EdgeInsets.fromLTRB(28, 0, 28, 24),
    child: Text(
      footer_text,
      textAlign: TextAlign.center,
      style: const TextStyle(color: Color(0xFF423E50)),
    ),
  ),
],
),
),
),
),
),
),
);
}
}

```

### SignUpScreen.dart

```

import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:osiris/Screens/LoginScreen.dart';
import 'package:osiris/Services/auth.dart';
import 'package:osiris/Services/consts.dart';
import 'package:provider/provider.dart';

import 'package:flutter_svg/flutter_svg.dart';

class SignUpScreen extends StatelessWidget {
  const SignUpScreen({super.key, Key? key1});

```

```

@override
Widget build(BuildContext context) {
  var size = MediaQuery.of(context).size;
  TextEditingController emailController = TextEditingController();
  TextEditingController passwordController = TextEditingController();
  TextEditingController confirmPasswordController = TextEditingController();

  return Scaffold(
    body: Container(
      color: background_primary,
      child: SafeArea(
        child: SingleChildScrollView(
          child: Container(
            color: background_primary,
            height: size.height,
            width: size.width,
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Container(
                  height: size.height * 0.2,
                ),
                SvgPicture.asset(
                  "assets/netflix.svg",
                  width: size.width * 0.40,
                ),
                Padding(
                  padding: const EdgeInsets.all(24.0),
                  child: Column(
                    children: [
                      TextFormField(
                        controller: emailController,
                        decoration: const InputDecoration(
                          labelText: 'Email',
                          filled: true,
                          fillColor: Colors.white,
                        ),
                      ),
                      const SizedBox(height: 20),
                      TextFormField(
                        controller: passwordController,
                        decoration: const InputDecoration(
                          labelText: 'Password',

```

```

        filled: true,
        fillColor: Colors.white,
      ),
      obscureText: true,
    ),
    const SizedBox(height: 20),
    TextFormField(
      controller: confirmPasswordController,
      decoration: const InputDecoration(
        labelText: 'Confirm Password',
        filled: true,
        fillColor: Colors.white,
      ),
      obscureText: true,
    ),
    const SizedBox(height: 20),
    ElevatedButton(
      onPressed: () async {
        if (passwordController.text.trim() == confirmPasswordController.text.trim()) {
          try {
            await FirebaseAuth.instance
              .createUserWithEmailAndPassword(
                email: emailController.text.trim(),
                password: passwordController.text.trim(),
              );
            // Sign up successful, navigate to login screen
            Navigator.pushReplacement(
              context,
              MaterialPageRoute(builder: (context) => const LoginScreen()),
            );
          } catch (e) {
            // Handle sign up failure
            print('Error: $e');
          }
        } else {
          // Handle password mismatch
          print('Error: Passwords do not match');
        }
      },
      child: const Text('Sign Up with Email'),
    ),
    const SizedBox(height: 20),
    ElevatedButton.icon(
      onPressed: () {

```

```

        final provider = Provider.of<GoogleSignInProvider>(
            context,
            listen: false);
        provider.googleLogin();
    },
    icon: const Falcon(
        FontAwesomeIcons.google,
        color: Colors.white,
    ),
    label: const Text(
        'Continue with Google',
        style: TextStyle(color: Colors.white),
    ),
    style: ElevatedButton.styleFrom(
        shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(14)),
        backgroundColor: const Color(0xFF2A292F)),
),
const SizedBox(height: 20),
TextButton(
    onPressed: () {
        Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => const LoginScreen()),
        );
    },
    child: const Text(
        'Already have an account? Sign In',
        style: TextStyle(color: Colors.white),
    ),
),
],
),
),
Container(
    margin: const EdgeInsets.fromLTRB(28, 0, 28, 24),
    child: Text(
        footer_text,
        textAlign: TextAlign.center,
        style: const TextStyle(color: Color(0xFF423E50)),
    ),
),
],
),
),

```

```
    },
    },
    },
    );
}
```



## Output:-

