EXPERIMENT NO 4

NAME-kartik bhat

CLASS-D15A

ROLL NO-03

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(),

);

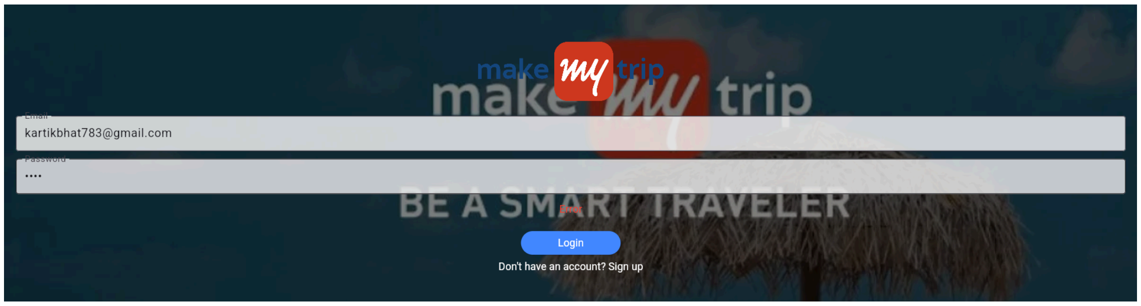
CODE

The introduction page

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
// ignore: unused_import
import 'home_screen.dart';
// ignore: unused_import
import 'signup_screen.dart';

class LoginScreen extends StatefulWidget {
  const LoginScreen({Key? key}) : super(key: key);

  @override
  State<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  final TextEditingController _emailController =
TextEditingController();
  final TextEditingController _passwordController =
TextEditingController();
  String? errorMessage;

  Future<void> _login() async {
    try {
```

```
    await FirebaseAuth.instance.signInWithEmailAndPassword(
      email: _emailController.text.trim(),
      password: _passwordController.text.trim(),
    );
    Navigator.pushReplacementNamed(context, '/home');
  } on FirebaseAuthException catch (e) {
    setState(() {
      errorMessage = e.message;
    });
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Stack(
      children: [
        // 🔹 Background Image
        Positioned.fill(
          child: Image.asset(
            "assets/mmt_background.jpg",
            fit: BoxFit.cover,
          ),
        ),

        // 🔹 Dark overlay
        Positioned.fill(
          child: Container(color: Colors.black.withOpacity(0.5)),
        ),

        // 🔹 Login Form
        Center(
          child: SingleChildScrollView(
            padding: const EdgeInsets.all(16.0),
            child: Column(
              mainAxisSize: MainAxisSize.min,
              children: [
                  Image.asset("assets/mmt_logo.png", height: 80),
                  const SizedBox(height: 20),

                  TextField(
                    controller: _emailController,
                    keyboardType: TextInputType.emailAddress,
                    decoration: InputDecoration(
                      labelText: 'Email',
                      filled: true,
                      fillColor: Colors.white.withOpacity(0.8),
                      border: const OutlineInputBorder(),
                    ),
                  ),
                  const SizedBox(height: 10),

                  TextField(
                    controller: _passwordController,
                    obscureText: true,
                    decoration: InputDecoration(
                      labelText: 'Password',
                      filled: true,
                      fillColor: Colors.white.withOpacity(0.8),
                      border: const OutlineInputBorder(),
                    ),
                  ),
                  const SizedBox(height: 10),

                  if (errorMessage != null)
                    Padding(
                      padding: const EdgeInsets.only(bottom: 10),
                      child: Text(
                        errorMessage!,
                        style: const TextStyle(color: Colors.red,
fontWeight: FontWeight.w600),
                      ),
                    ),
                  ElevatedButton(
                    onPressed: _login,
                    style: ElevatedButton.styleFrom(
                      backgroundColor: Colors.blueAccent,
                      foregroundColor: Colors.white,
                      padding: const
EdgeInsets.symmetric(horizontal: 50, vertical: 12),
                    ),
                    child: const Text("Login"),
                  ),
                  const SizedBox(height: 10),

                  TextButton(
                    onPressed: () {
                      Navigator.pushNamed(context, '/signup');
                    },
                    child: const Text(
                      "Don't have an account? Sign up",
                      style: TextStyle(color: Colors.white),
                    ),
                  ),
                ],
              ),
            ),
          ),
        ],
      ),
    );
  }
}
```

OUTPUT



kartikbhat783@gmail.com

••••

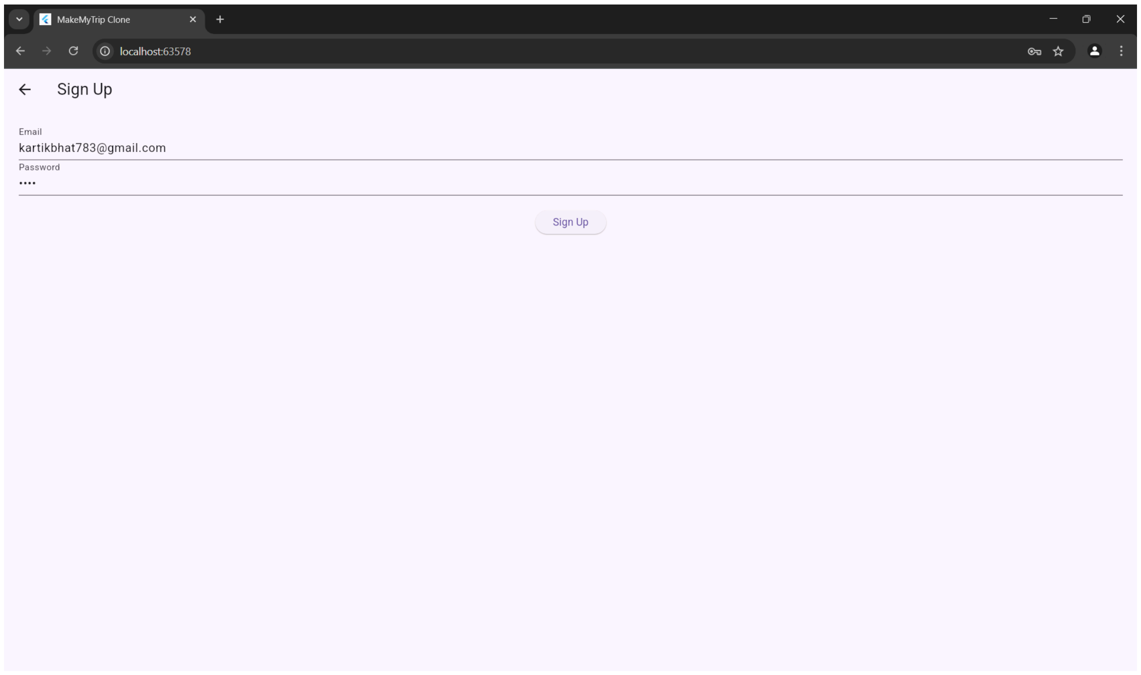
Create Account Page

CODE

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
// ignore: unused_import
import 'login_screen.dart';


class SignUpScreen extends StatefulWidget {
  const SignUpScreen({Key? key}) : super(key: key);


  @override
  State<SignUpScreen> createState() =>
_SignUpScreenState();
}


class _SignUpScreenState extends State<SignUpScreen> {
  final TextEditingController _emailController =
TextEditingController();
  final TextEditingController _passwordController =
TextEditingController();

  String? errorMessage;


  Future<void> _signup() async {

    try {

      final userCredential = await
FirebaseAuth.instance.createUserWithEmailAndPassword(

        email: _emailController.text.trim(),

        password: _passwordController.text.trim(),

      );


      // Store extra user info in Firestore (optional
but useful)

      await FirebaseFirestore.instance

          .collection('users')

          .doc(userCredential.user!.uid)

          .set({

        'email': _emailController.text.trim(),

        'name': '', // you can ask for this in form
later

        'photoUrl': '', // default empty

      });


      // After signup, go to login screen

      Navigator.pushReplacementNamed(context,
'/login');
```

```
    } on FirebaseAuthException catch (e) {
      setState(() {
        errorMessage = e.message;
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text("Sign Up")),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          children: [
            TextField(
              controller: _emailController,
              decoration: const
InputDecoration(labelText: 'Email'),
              keyboardType:
TextInputType.emailAddress,
            ),
            const SizedBox(height: 10),
            TextField(
              controller: _passwordController,
              decoration: const
InputDecoration(labelText: 'Password'),
              obscureText: true,
            ),
            const SizedBox(height: 20),
            if (errorMessage != null)
              Text(errorMessage!, style: const
TextStyle(color: Colors.red)),
            ElevatedButton(
              onPressed: _signup,
              child: const Text("Sign Up"),
            ),
          ],
        ),
      ),
    );
  }
}
```

OUTPUT



MakeMyTrip Clone

localhost:63578

Sign Up

Email
kartikbhat783@gmail.com

Password
••••

Sign Up

T