Experiment 4

Name:Mohit Tarachandani
Div: D15A
Roll no: 62

Aim: To create an interactive form using the form widget
Theory:

1. Form Widget:
    - The Form widget is a container used to group multiple form fields together.
    - It helps manage the state of the form, including validation, submission, and resetting.
    - The Form widget maintains a FormState object that holds the current state of the form fields.
    - Form widgets facilitate form submission, validation, and error handling.

2. FormField Widget:
    - A FormField widget represents a single form field within a Form.
    - Flutter provides various subclasses of the FormField widget for different types of input fields, such as TextFormField, CheckboxFormField, RadioFormField, DropdownButtonFormField, etc.
    - Each form field widget encapsulates the logic for validating user input and managing its state.
    - Form fields can be customized with properties to specify validation rules, error messages, initial values, input formatting, and more.
    - Form fields automatically register themselves with the Form widget and handle validation and state management transparently.

3. Validation:
    - Flutter's form widgets include built-in support for validation to ensure that user input meets specific criteria.
    - Form fields can be configured with validation functions or validators to check the correctness of user input.
    - Validators can be synchronous or asynchronous functions that return error messages if the input is invalid.
    - Flutter provides a FormFieldState class associated with each form field, which exposes methods to validate the field's value and retrieve validation errors if any.

4. Submission:
    - The Form widget provides a mechanism to submit the form data once it's been filled out by the user.
    - Developers can define an onSaved callback for each form field to specify how the field's value should be processed when the form is submitted.

- When the form is submitted, the onSaved callbacks for all form fields are invoked, allowing developers to collect, process, and submit the form data to a backend server or perform other actions.

5.
```dart
import 'package:flight_app_ui/screens/home_screen.dart';
import 'package:flutter/gestures.dart';
import 'package:flutter/material.dart';
import '../widgets/feild.dart';
import '../widgets/text.dart';
import '../auth.dart'; // Import your Auth class

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

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

class _LoginScreenState extends State<LoginScreen> {
  TextEditingController _emailController = TextEditingController();
  TextEditingController _passwordController = TextEditingController();
  final Auth _auth = Auth(); // Initialize your Auth class

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

  Future<void> _login() async {
    try {
      await _auth.signInWithEmailAndPassword(
        email: _emailController.text,
        password: _passwordController.text,
      );
      // Navigate to the home screen after successful login
      Navigator.of(context).pushReplacement(MaterialPageRoute(
        builder: (context) => const HomeScreen(),
      ));
    } catch (e) {
      // Handle login errors here
      print('Login failed: $e');
      // Optionally show a snackbar or dialog to the user
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Theme.of(context).primaryColor,
      body: SingleChildScrollView(
        child: Stack(
          children: [
            Container(
```

```
                height: MediaQuery.of(context).size.height,
                padding: const EdgeInsets.all(15),
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: [
                    Center(
                      child: Hero(
                        tag: "logo",
                        child: SizedBox(
                          height: 130,
                          width: 130,
                          child: Image.asset(
                            "assets/logo.png",
                            color: Theme.of(context).indicatorColor,
                          ),
                        ),
                      ),
                    ),
                    Field(
                      controller: _emailController,
                      hinttext: "E-MAIL",
                      icon: Icons.alternate_email,
                    ),
                    const SizedBox(height: 20,),
                    Field(
                      controller: _passwordController,
                      hinttext: "PASSWORD",
                      icon: Icons.lock,
                    ),
                    const SizedBox(height: 25,),
                    GestureDetector(
                      onTap: _login, // Call _login function when tapped
                      child: Container(
                        height: 50,
                        width: double.infinity,
                        decoration: BoxDecoration(
                            color: Theme.of(context).indicatorColor,
                            borderRadius: BorderRadius.circular(15)
                        ),
                        alignment: Alignment.center,
                        child: TextUtil(
                          text: "Log in",
                          weight: true,
                          color: Theme.of(context).primaryColor,
                          size: 16,
                        ),
                      ),
                    ),
                    const SizedBox(height: 25,),
                    RichText(
                      text: TextSpan(
                        text: 'Not a member? ',
                        style: TextStyle(color:
Theme.of(context).canvasColor,fontSize: 14),
                        children: <TextSpan>[
                          TextSpan(
                            text: 'Join now',
                            style: TextStyle(fontWeight:
```

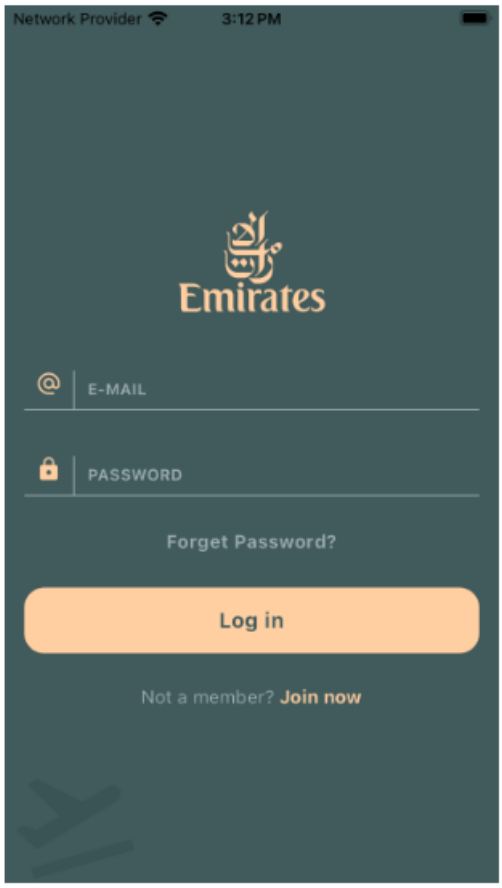
```
FontWeight.bold,color: Theme.of(context).indicatorColor),
                          recognizer: TapGestureRecognizer()..onTap=(){

                          },
                        ),
                      ],
                    ),
                  ),
                ],
              ),
            ),
            Positioned(
              bottom: 0,
              child: Transform.rotate(
                  angle: 6,
                  child:const Icon(Icons.flight_takeoff,size:
100,color: Color(0xff3a5455),)
              ),
            ),
          ],
        ),
      ),
    );
  }
}
import 'package:firebase_auth/firebase_auth.dart';

class Auth {
  final FirebaseAuth _firebaseAuth = FirebaseAuth.instance;

  User? get currentUser => _firebaseAuth.currentUser;

  Stream<User?> get authStateChanges =>
_firebaseAuth.authStateChanges();

  Future<void> signInWithEmailAndPassword({
    required String email,
    required String password,
  }) async {
    await _firebaseAuth.signInWithEmailAndPassword(
      email: email,
      password: password,
    );
  }

  Future<void> createUserWithEmailAndPassword({
    required String email,
    required String password,
  }) async {
    await _firebaseAuth.createUserWithEmailAndPassword(
      email: email,
      password: password,
    );
  }

  Future<void> signOut() async {
    await _firebaseAuth.signOut();
```

```
  }
}
```

```
      import 'dart:async';

import 'package:flutter/material.dart';

import 'login_screen.dart';
class SplashScreen extends StatefulWidget {
  const SplashScreen({super.key});

  @override
  State<SplashScreen> createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen>  with
SingleTickerProviderStateMixin{
/// ANIMATION CONTROLLER
  late AnimationController _controller;
  /// ANIMATION
  late Animation<double> _animation;

  @override
  void initState() {
    super.initState();
    /// INITIALING THE CONTROLLER
    _controller = AnimationController(vsync: this,duration: const
Duration(seconds: 2));
    /// INITIALING THE ANIMATION
    _animation= CurvedAnimation(parent: _controller, curve:
Curves.fastEaseInToSlowEaseOut);
    /// STARTING THE ANIMATION
    _controller.forward();
    /// TIMER FOR SPLASH DURATION
    Timer( const Duration(seconds: 3),(){
      /// NAVIAGTING TO LOGIN SCREEN
      Navigator.of(context).pushReplacement(MaterialPageRoute(builder:
(context)=>LoginScreen()));

    }

    );
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Theme.of(context).primaryColor,
      body: ScaleTransition(
        scale:  animation,
        child: Center(
          child: Hero(
            tag: "logo",
            child: SizedBox(
              height: 200,
                width: 200,
```

```
                  child: Image.asset("assets/logo.png",color:
Theme.of(context).indicatorColor,)),
            ),
          ),
        ),
      );
    }
}
```

App UI:



Widgets used: Form Widget, Form Widget Fields

Conclusion: Therefore understood the use of form widget in Flutter. Implemented signup and login page using form widget in my Flutter application.