Name: Abujaid Ansari **Batch: A**

Class: D15B **Roll No. 02**

Experiment No. 5

Aim: To Apply routing, navigation and gestures in Flutter.

Theory:

Routing and Navigation in Flutter:

Navigation and routing are some of the core concepts of all mobile application, which allows the user to move between different pages. We know that every mobile application contains several screens for displaying different types of information. **For example,** an app can have a screen that contains various products. When the user taps on that product, immediately it will display detailed information about that product.

In Flutter, the screens and pages are known as **routes,** and these routes are just a widget. In Android, a route is similar to an **Activity,** whereas, in iOS, it is equivalent to a **ViewController.**

In any mobile app, navigating to different pages defines the workflow of the application, and the way to handle the navigation is known as **routing.** Flutter provides a basic routing class **MaterialPageRoute** and two methods **Navigator.push()** and **Navigator.pop()** that shows how to navigate between two routes. The following steps are required to start navigation in your application.

Step 1: First, you need to create two routes.

Step 2: Then, navigate to one route from another route by using the Navigator.push() method.

Step 3: Finally, navigate to the first route by using the Navigator.pop() method.

Gestures:

Gestures are primarily a way for a user to interact with a mobile (or any touch based device) application. Gestures are generally defined as any physical action / movement of a user in the intention of activating a specific control of the mobile device. Gestures are as simple as tapping the screen of the mobile device to more complex actions used in gaming applications.

Some of the widely used gestures are mentioned here −

- **Tap** − Touching the surface of the device with fingertip for a short period and then releasing the fingertip.
- **Double Tap** − Tapping twice in a short time.
- **Drag** − Touching the surface of the device with fingertip and then moving the fingertip in a steady manner and then finally releasing the fingertip.
- **Flick** − Similar to dragging, but doing it in a speeder way.

Implementation:

login.dart

```
import 'package:bmi_calculator/signup.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';

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

  @override
  State<Login> createState() => _LoginState();
}

class _LoginState extends State<Login> {

  TextEditingController email=TextEditingController();
  TextEditingController password=TextEditingController();

  signIn() async {
    await FirebaseAuth.instance.signInWithEmailAndPassword(email: email.text,
password: password.text);
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Login"),),
      body:Padding(
        padding: const EdgeInsets.all(20.0),
        child: Column(
          children: [
            TextField(
              controller:email,
              decoration: InputDecoration(hintText: 'Enter email'),
            ),
            TextField(
              controller:password,
              decoration: InputDecoration(hintText: 'Enter password'),
```

```
          ),
          ElevatedButton(onPressed: (()=>signIn()), child: Text("Login")),
          SizedBox(height: 30,),
          ElevatedButton(onPressed: (()=>Get.to(SignUp())), child: Text("Register
Now"))
        ],),
    )
  );
 }
}
```

signup.dart

```
import 'package:bmi_calculator/wrapper.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';

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

 @override
 State<SignUp> createState() => _SignUpState();
}

class _SignUpState extends State<SignUp> {

 TextEditingController email=TextEditingController();
 TextEditingController password=TextEditingController();

 signUp() async {
  await FirebaseAuth.instance.createUserWithEmailAndPassword(email: email.text,
password: password.text);
  Get.offAll(()=>Wrapper());
 }
 @override
 Widget build(BuildContext context) {
return Scaffold(
    appBar: AppBar(title: Text("Sign Up"),),
    body:Padding(
      padding: const EdgeInsets.all(20.0),
      child: Column(
        children: [
          TextField(
            controller:email,
            decoration: InputDecoration(hintText: 'Enter email'),
          ),
          TextField(
```

```
            controller:password,
            decoration: InputDecoration(hintText: 'Enter password'),
          ),
          ElevatedButton(onPressed: (()=>signUp()), child: Text("Sign Up"))
        ],),
    )
  );
 }
}
```

home.dart

```
import "package:firebase_auth/firebase_auth.dart";
import "package:flutter/material.dart";

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

  @override
  State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {
  final user = FirebaseAuth.instance.currentUser;
  signout() async{
    await FirebaseAuth.instance.signOut();
  }
  TextEditingController _heightController = TextEditingController();
  TextEditingController _weightController = TextEditingController();
  TextEditingController _ageController = TextEditingController();
  String _selectedGender = 'Male';

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('My BMI'),
        centerTitle: true,
      ),
      body: Container(
        decoration: BoxDecoration(
          image: DecorationImage(
            image: AssetImage('assets/bg2.png'), // Set your background image
            fit: BoxFit.cover,
          ),
        ),
        child: Padding(
          padding: const EdgeInsets.all(16.0),
```

```
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: [
            Image.asset('assets/bmi_logo.png', height: 100), // Add your logo image
            SizedBox(height: 30),
            TextFormField(
              controller: _heightController,
              keyboardType: TextInputType.number,
              decoration: InputDecoration(
                labelText: 'Height (cm)',
                prefixIcon: Icon(Icons.height), // Add height icon
              ),
            ),
            SizedBox(height: 20),
            TextFormField(
              controller: _weightController,
              keyboardType: TextInputType.number,
              decoration: InputDecoration(
                labelText: 'Weight (kg)',
                prefixIcon: Icon(Icons.fitness_center), // Add weight icon
              ),
            ),
            SizedBox(height: 20),
            TextFormField(
              controller: _ageController,
              keyboardType: TextInputType.number,
              decoration: InputDecoration(
                labelText: 'Age (years)',
                prefixIcon: Icon(Icons.person), // Add person icon
              ),
            ),
            SizedBox(height: 20),
            DropdownButtonFormField<String>(
              value: _selectedGender,
              onChanged: (newValue) {
                setState(() {
                  _selectedGender = newValue!;
                });
              },
              items: <String>['Male', 'Female'].map((String value) {
                return DropdownMenuItem<String>(
                  value: value,
                  child: Text(value),
                );
              }).toList(),
              decoration: InputDecoration(
                labelText: 'Gender',
```

```
                prefixIcon: Icon(Icons.accessibility), // Add gender icon
              ),
            ),
            SizedBox(height: 30),
            ElevatedButton(
              onPressed: () {
                _calculateBMI();
              },
              child: Text('Calculate BMI'),
              style: ElevatedButton.styleFrom(
                primary: Colors.blue,
                padding: EdgeInsets.symmetric(vertical: 15),
              ),
            ),
          ],
        ),
      ),
    ),
    floatingActionButton: FloatingActionButton(onPressed: (()=>signout()),
    child: Icon(Icons.login_rounded),
    ),
  );
}

void _calculateBMI() {
  double height = double.tryParse(_heightController.text) ?? 0.0;
  double weight = double.tryParse(_weightController.text) ?? 0.0;
  int age = int.tryParse(_ageController.text) ?? 0;

  if (height <= 0 || weight <= 0 || age <= 0) {
    setState(() {
      var _bmiResult = 0.0;
      var _bmiFeedback = '';
    });
    return;
  }

  double bmi = weight / ((height / 100) * (height / 100));
  String feedback = '';

  if (_selectedGender == 'Male') {
    if (age >= 20 && age <= 39) {
      if (bmi < 8.5) {
        feedback = 'Low BMI';
      } else if (bmi >= 8.5 && bmi <= 20.7) {
        feedback = 'Normal BMI';
      } else if (bmi > 20.7) {
        feedback = 'High BMI';
```

```
      }
    } else if (age >= 40 && age <= 59) {
      if (bmi < 11) {
        feedback = 'Low BMI';
      } else if (bmi >= 11 && bmi <= 22.2) {
        feedback = 'Normal BMI';
      } else if (bmi > 22.2) {
        feedback = 'High BMI';
      }
    } else if (age >= 60) {
      if (bmi < 13) {
        feedback = 'Low BMI';
      } else if (bmi >= 13 && bmi <= 25.4) {
        feedback = 'Normal BMI';
      } else if (bmi > 25.4) {
        feedback = 'High BMI';
      }
    }
  } else if (_selectedGender == 'Female') {
    if (age >= 20 && age <= 39) {
      if (bmi < 7.8) {
        feedback = 'Low BMI';
      } else if (bmi >= 7.8 && bmi <= 19.1) {
        feedback = 'Normal BMI';
      } else if (bmi > 19.1) {
        feedback = 'High BMI';
      }
    } else if (age >= 40 && age <= 59) {
      if (bmi < 9.8) {
        feedback = 'Low BMI';
      } else if (bmi >= 9.8 && bmi <= 21.9) {
        feedback = 'Normal BMI';
      } else if (bmi > 21.9) {
        feedback = 'High BMI';
      }
    }
  }

  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) => ResultScreen(
        bmiResult: bmi,
        bmiFeedback: feedback,
      ),
    ),
  );
}
```

```
}
// showRecommendations(feedback);

class ResultScreen extends StatelessWidget {
  final double bmiResult;
  final String bmiFeedback;
  String recH=' ';
  String recN=' ';
  String recL=' ';

  ResultScreen({required this.bmiResult, required this.bmiFeedback});

  @override
  Widget build(BuildContext context) {
    Color bgColor;
    if (bmiFeedback == 'High BMI') {
      bgColor = Colors.red;
      recH='\n\tWorkout Plan: \n 1.Try doing physical activity like walking 30 mins a day.\n\t2.Go to gym at least thrice a week and do a cardio-vascular training.\n\t3.Include Aerobic exercises to burn calories\n\tDiet Plan:\n 1.Drink at least 4 litres water daily\n\t2.Try avoiding sugary and deep fried food items.\n\t 3.Look at the daily calorie count.';
    } else if (bmiFeedback == 'Normal BMI') {
      bgColor = Colors.green;
      recN='\n\tYou are doing well, Keep it up!.';
    } else {
      bgColor = Colors.yellow;
      recL='\n\tWorkout Plan: \n 1.Go to gym at least 3 times a week and do weighlifting exercises to increase muscle mass.\n\t2.Take proper rest.\n\tDiet Plan:\n 1.Drink at least 4 litres water daily\n\t2.Eat more proteins like egg, soy, paneer and chicken.\n\t 3.Intake more calories and nutrients.';
    }

    return Scaffold(
      appBar: AppBar(
        title: Text('BMI Result'),
        centerTitle: true,
      ),
      backgroundColor: bgColor,
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              'BMI: ${bmiResult.toStringAsFixed(2)}',
              style: TextStyle(fontSize: 40.0, fontWeight: FontWeight.bold),
            ),
            SizedBox(height: 10),
```

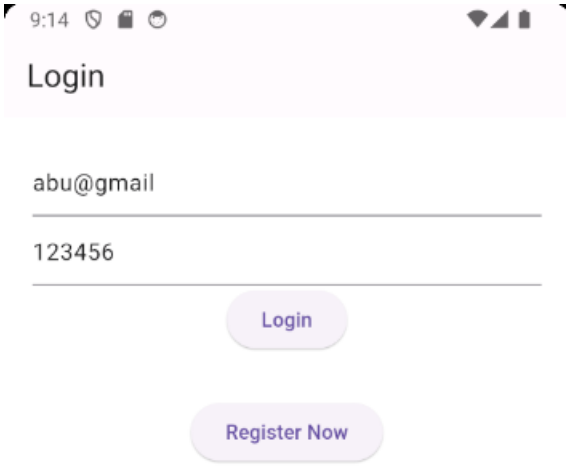
```
          Text(
            bmiFeedback,
            style: TextStyle(fontSize: 35.0, fontWeight: FontWeight.bold),
          ),
          Text(
            recH,
            style: TextStyle(fontSize: 20.0, fontWeight: FontWeight.bold, color:
Colors.black),
          ),
          Text(
            recN,
            style: TextStyle(fontSize: 20.0, fontWeight: FontWeight.bold, color:
Colors.black),
          ),
          Text(
            recL,
            style: TextStyle(fontSize: 20.0, fontWeight: FontWeight.bold, color:
Colors.black),
          ),
        ],
      ),
    ),
    floatingActionButton: FloatingActionButton(onPressed: (()=>signout()),
    child: Icon(Icons.login_rounded),
    ),

  );
 }

 signout() {}
}
```

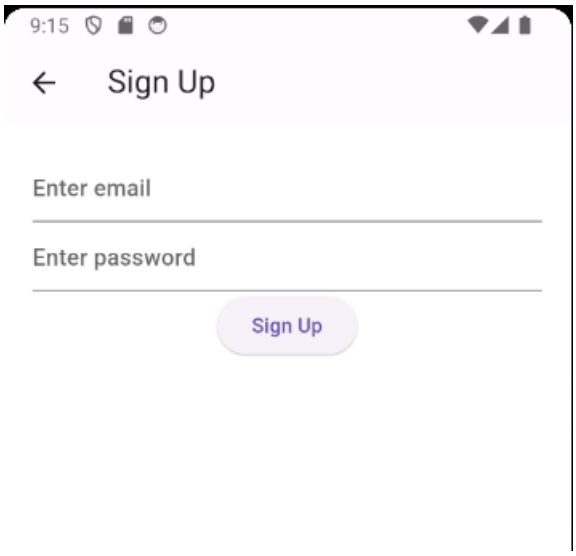
Output:

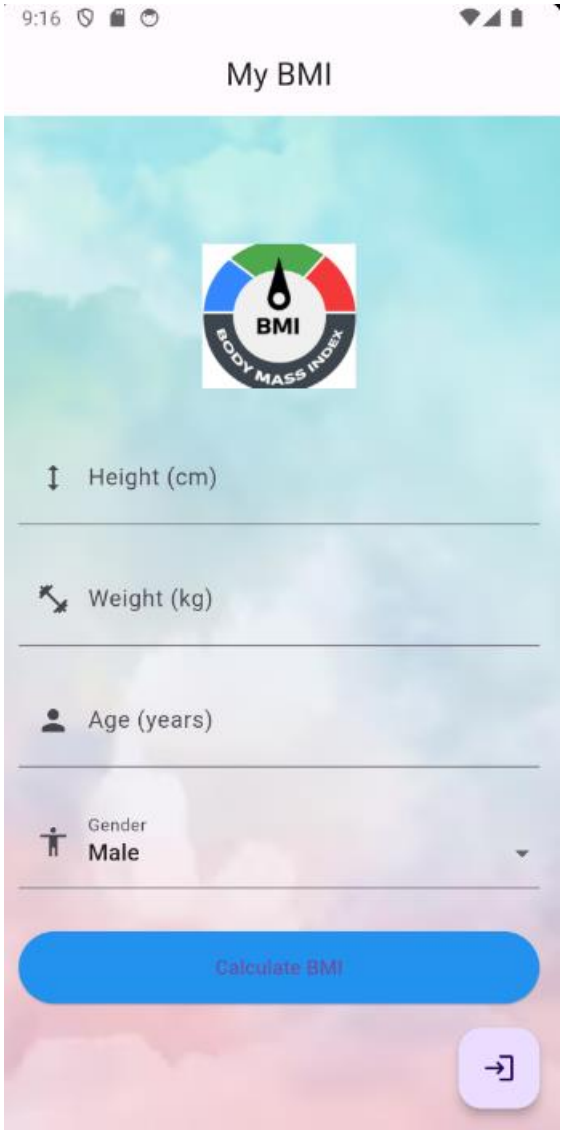
1. We have used navigator functions in our login and signup routes.

2. Also, we have used touch gestures in all the buttons like register and logout button, calculate BMI and logout button.

3. In login page, when the login button is clicked it navigates using Navigate.push() function to the home page.


9:14
Login
abu@gmail
123456
Login
Register Now


9:15
Sign Up
Enter email
Enter password
Sign Up



Conclusion:

Hence, we understood routing, navigation and gestures by implementing them in our flutter app. We used routing and navigation in the login page, signup page and home page. We used click gesture in all the buttons.