Name: Aastha Jajoo **Roll no.: 23** **Class: D15B**

MAD Experiment 5

Aim : To apply navigation, routing and gestures in Flutter App.

Theory :

Flutter Navigation and Routing :

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 used to interact with an application. It is generally used in touch-based devices to physically interact with the application. It can be as simple as a single tap on the screen to a more complex physical interaction like swiping in a specific direction to scrolling down an application. It is heavily used in gaming and more or less every application requires it to function as devices turn more touch-based than ever. In this article, we will discuss them in detail.

Some widely used gestures are mentioned here :

- Tap: Touching the surface of the device with the fingertip for a small duration of time period and finally releasing the fingertip.
- Double Tap: Tapping twice in a short time.
- Drag: Touching the surface of the device with the fingertip and then moving the fingertip in a steadily and finally releasing the fingertip.

- Flick: Similar to dragging, but doing it in a speedier way.
- Pinch: Pinching the surface of the device using two fingers.
- Zoom: Opposite of pinching.
- Panning: Touching the device surface with the fingertip and moving it in the desired direction without releasing the fingertip.

The GestureDetector widget in flutter is used to detect physical interaction with the application on the UI. If a widget is supposed to experience a gesture, it is kept inside the GestureDetector widget. The same widget catches the gesture and returns the appropriate action or response

.

Below is the list of gestures and their corresponding events :

Tap

- onTapDown
- onTapUp
- onTap
- onTapCan

cel Double tap

- onDoubleT

ap Long press

- onLongPr

ess Vertical drag

- onVerticalDragStart
- onVerticalDragUpdate
- onVerticalDragE

nd Horizontal drag

- onHorizontalDragStart
- onHorizontalDragUpdate
- onHorizontalDragE

nd Pan

- onPanStart
- onPanUpdate

Code:

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyFlashcardApp());
}

class MyFlashcardApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flashcard App',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: LoginPage(),
    );
  }
}

class LoginPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Login'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            TextField(
              decoration: InputDecoration(
                hintText: 'Enter your username',
              ),
```

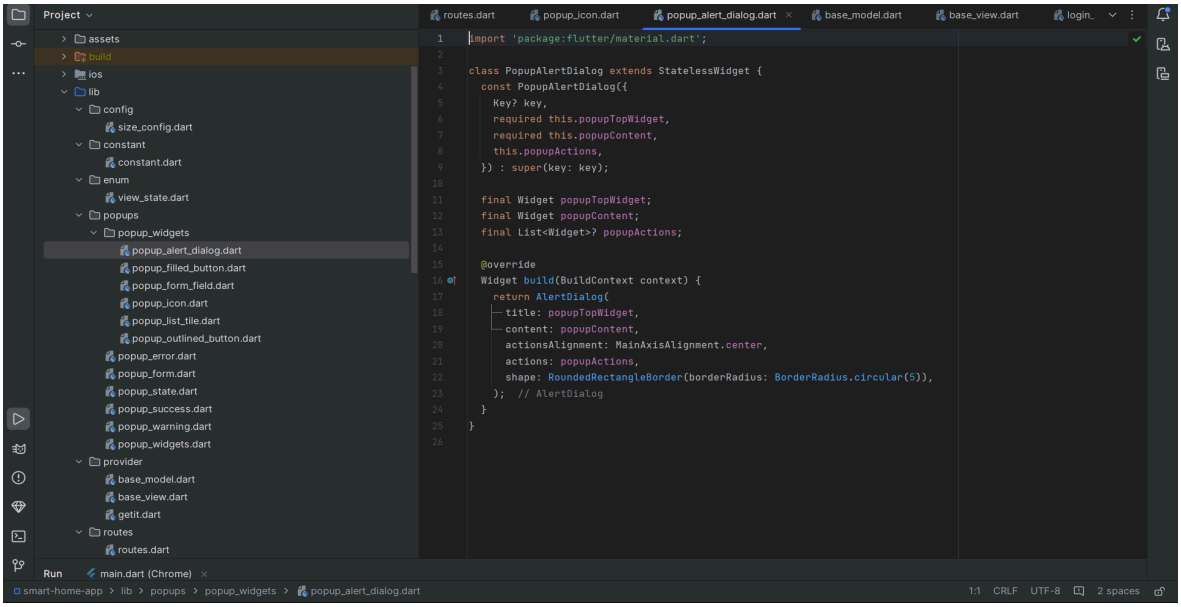
```
            ),
            SizedBox(height: 20),
            TextField(
              decoration: InputDecoration(
                hintText: 'Enter your password',
              ),
              obscureText: true,
            ),
            SizedBox(height: 20),
            ElevatedButton(
              onPressed: () {
                // Add login functionality here
                Navigator.push(
                  context,
                  MaterialPageRoute(builder: (context) => HomePage()),
                );
              },
              child: Text('Login'),
            ),
          ],
        ),
      ),
    );
  }
}

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Flashcard App'),
      ),
      body: Center(
        child: Text(
          'Welcome to the Flashcard App!',
```

```
          style: TextStyle(fontSize: 24),
        ),
      ),
    );
  }
}
```

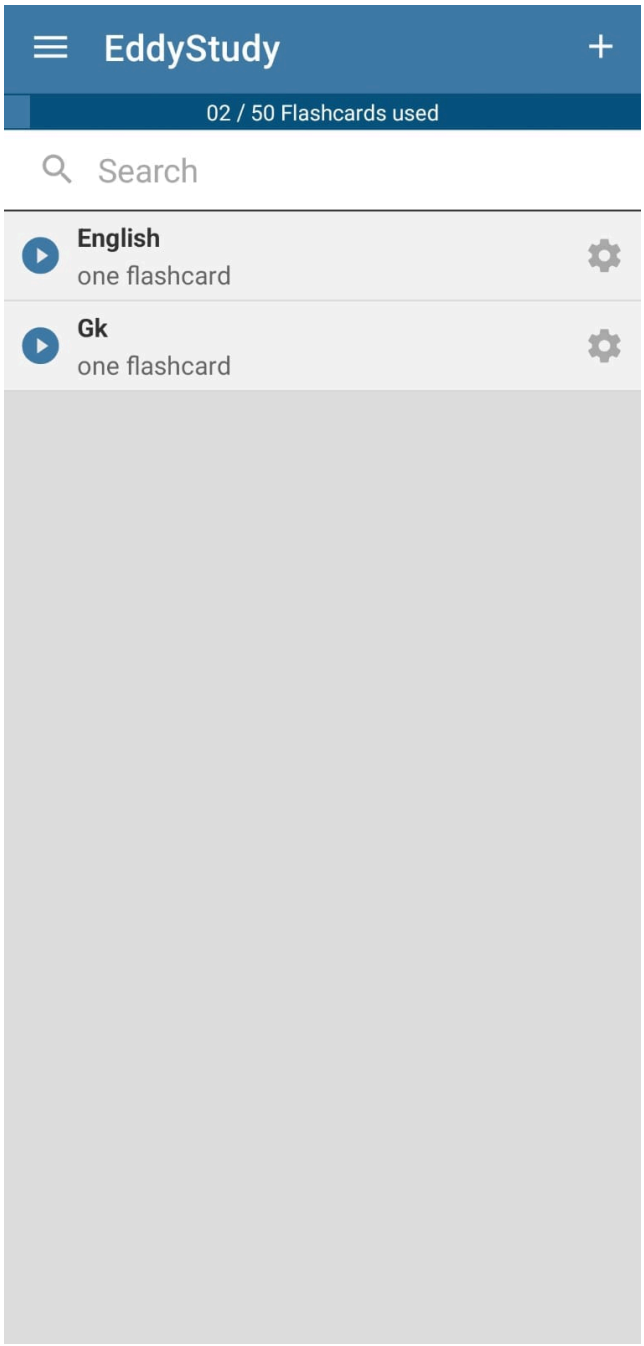
routes.dart

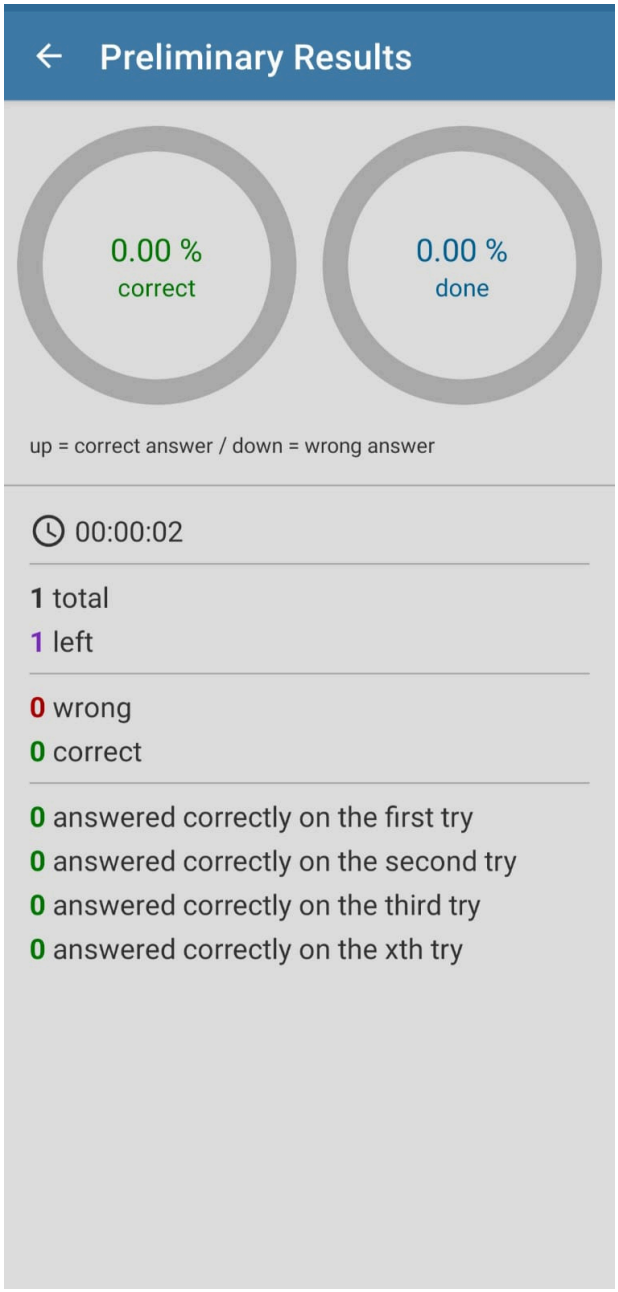
```
import 'package:domus/src/screens/set_event_screen/set_event_screen.dart';
import 'package:domus/src/screens/edit_profile/edit_profile.dart';
import 'package:domus/src/screens/login_screen/login_screen.dart';
import 'package:domus/src/screens/settings_screen/settings_screen.dart';
import 'package:domus/src/screens/smart_ac/smart_ac.dart';
import 'package:domus/src/screens/smart_light/smart_light.dart';
import 'package:domus/src/screens/smart_speaker/smart_speaker.dart';
import 'package:domus/src/screens/smart_fan/smart_fan.dart';
import 'package:domus/src/screens/splash_screen/splash_screen.dart';
import 'package:domus/src/screens/stats_screen/stats_screen.dart';
import 'package:flutter/cupertino.dart';
import 'package:domus/src/screens/home_screen/home_screen.dart';
import 'package:domus/src/screens/my_list_screen/my_list_screen.dart';
import 'package:domus/src/screens/savings_screen/savings_screen.dart';
import 'package:domus/src/screens/smart_tv/smart_tv.dart';

// Routes arranged in ascending order

final Map<String, WidgetBuilder> routes = {
  EditProfile.routeName: (context) => const EditProfile(),
  HomeScreen.routeName: (context) => const HomeScreen(),
  LoginScreen.routeName: (context) => const LoginScreen(),
  SavingsScreen.routeName: (context) => const SavingsScreen(),
  SetEventScreen.routeName: (context) => const SetEventScreen(),
  SettingScreen.routeName: (context) => const SettingScreen(),
  SmartAC.routeName: (context) => const SmartAC(),
  SmartFan.routeName: (context) => const SmartFan(),
```

```
  SmartTV.routeName: (context) => const SmartTV(),
  SmartLight.routeName: (context) => const SmartLight(),
  SmartSpeaker.routeName: (context) => const SmartSpeaker(),
  SplashScreen.routeName: (context) => const SplashScreen(),
  StatsScreen.routeName: (context) => const StatsScreen(),
  MyListScreen.routeName: (context) => const MyListScreen()
};
```



Output :


EddyStudy
02 / 50 Flashcards used
Search
English
one flashcard
Gk
one flashcard


Preliminary Results
0.00 %
correct
0.00 %
done
up = correct answer / down = wrong answer
00:00:02
1 total
1 left
0 wrong
0 correct
0 answered correctly on the first try
0 answered correctly on the second try
0 answered correctly on the third try
0 answered correctly on the xth try

Here as we can see a routing is provided between the home screen and the meeting screen, this is done so as the user can easily navigate in between the meeting and perform any other tasks(for eg. Scheduling a new meeting while a meeting is going on.

Conclusion : We have understood the concept of gestures, their use and implemented it in our flutter app as a search bar. Also, we created two pages and routed them in our app and further enabled navigation. The flow is smooth for our app.