Name: Dhairyash Jain
Class: D15B
Roll: 22

MAD Lab 5

Aim: Navigation, Routing, and Gestures in Flutter

Introduction

Navigation, routing, and gestures are essential for building interactive Flutter applications. Navigation allows users to move between screens, routing manages structured transitions, and gestures detect user interactions like taps, swipes, and long presses.

Routing and Navigation in Flutter

In Flutter, routing and navigation are essential for managing screen transitions in an application. Flutter uses a stack-based navigation system, where screens (also called routes) are managed using a Navigator widget.

Types of Navigation in Flutter

1. Imperative Navigation (Navigator API)

- Uses Navigator.push() and Navigator.pop()
- Follows a stack-based approach (LIFO - Last In, First Out)
- Best suited for simple applications

2. Declarative Navigation (Go Router, Auto Route)

- Uses URL-based navigation
- Ideal for large applications with deep linking
- Navigator and Routes in Flutter

The Navigator manages the stack of screens in a Flutter app. Each screen is called a Route, and the Navigator widget helps in transitioning between routes.

1. Navigation in Flutter

Flutter's Navigator widget manages a stack of screens. Below is an example of basic navigation between two screens.

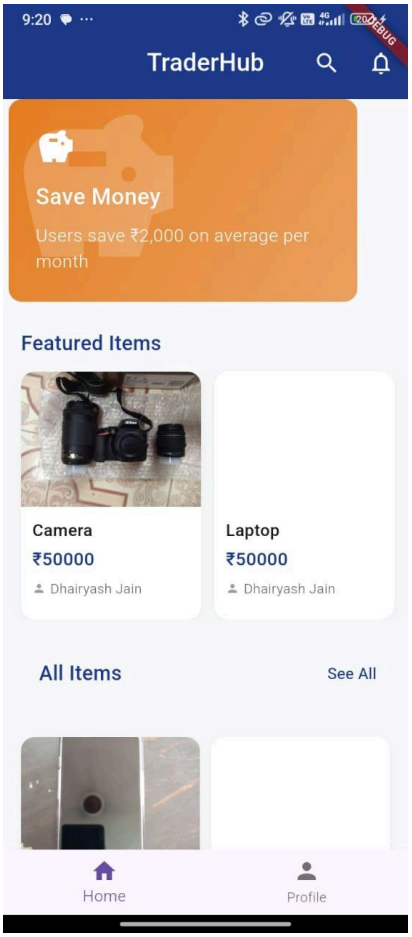
```
import 'package:flutter/material.dart';

void main() {
    runApp(MaterialApp(home: FirstScreen()));
}

class FirstScreen extends StatelessWidget { @override
    Widget build(BuildContext context) { return Scaffold(
            appBar: AppBar(title: Text('First Screen')), body: Center(
                child: ElevatedButton(
                    onPressed: () {
                        Navigator.push(
                            context, MaterialPageRoute(builder: (context) => SecondScreen()));
                    },
                    child: Text('Go to Second Screen'),
                ),
            ),
        );
    }
}

class SecondScreen extends StatelessWidget { @override
    Widget build(BuildContext context) { return Scaffold(
             appBar: AppBar(title: Text('Second Screen')), body: Center
            child: ElevatedButton( onPressed:
            () {
                                Navigator.pop(context);
                            },
                            child: Text('Go Back'),
                        ),
                    ),
                );
            }
```

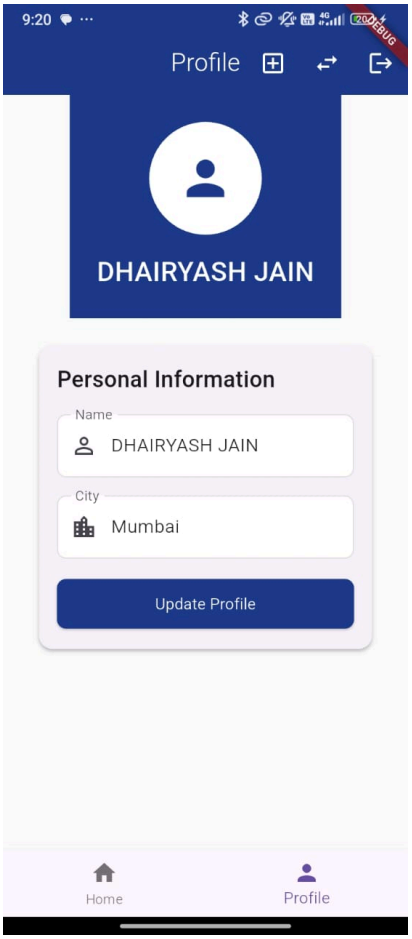
2. Gesture Detection in Flutter

Flutter's GestureDetector widget captures various gestures like taps, double taps, long presses, and swipes. Below is an example of detecting different gestures.

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

void main() {
    runApp(MaterialApp(home: GestureExample()));
}

class GestureExample extends StatefulWidget { @override
    _GestureExampleState createState() => _GestureExampleState();
}

class _GestureExampleState extends State<GestureExample> { String _message =
    "Tap or Swipe";

    @override
    Widget build(BuildContext context) { return Scaffold(
            appBar: AppBar(title: Text('Gesture Detector Example')), body:
            GestureDetector(
                onTap: () {
                    setState(() { _message = "Tapped!"; });
                },
                onDoubleTap: () {
                    setState(() { _message = "Double Tapped!"; });
                },
                onLongPress: () {
                    setState(() { _message = "Long Pressed!"; });
                },
                onHorizontalDragEnd: (details) { setState(() { _message =
                    "Swiped!"; });
                },
                child: Center(
                                    child: Text(_message, style: TextStyle(fontSize: 24, fontWeight:
    FontWeight.bold)),
                 ),
            ),
        );
    }
}
```

OUTPUT





Conclusion

Navigation allows movement between screens using Navigator. Routing helps structure navigation better using named routes. Gesture detection enables interactivity with user input. These features make Flutter apps more user-friendly.