Name : Niddhi Rijhwani
Class : D15B Roll No : 47
Subject : MPL Exp 05

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

Theory: 1. Navigation in Flutter : Navigation allows users to move from one screen (called a route in Flutter) to another.
Flutter uses a stack-based navigation model :

- Screens are pushed onto the navigation stack when navigated to.
- They are popped off when the user goes back.

a. **Basic navigation :**
```
Navigator.push(
  context,
  MaterialPageRoute(builder: (context) => SecondPage()),
);
```

b. **Pop (go back) :**
```
Navigator.pop(context);
```

2. Routing in Flutter : Routing defines how screens are connected and how users navigate between them.
Types of Routing:

a. **Named Routing :** Routes are defined in a central place (usually in MaterialApp).
```
MaterialApp(
  initialRoute: '/',
  routes: {
    '/': (context) => HomePage(),
    '/second': (context) => SecondPage(),
  },
);
Navigator.pushNamed(context, '/second');
```

b. **Anonymous Routing :** Routes are directly passed without naming them.
```
Navigator.push(
  context,
  MaterialPageRoute(builder: (context) => SecondPage()),
);
```

c. **Generated Routing :** Useful for complex apps, routes are generated dynamically.
```
onGenerateRoute: (settings) {
  if (settings.name == '/second') {
    return MaterialPageRoute(builder: (context) => SecondPage());
  }
  return null;
}
```

Gestures in Flutter : Gestures are used to detect user interactions like tap, drag, swipe, long press, etc.

GestureDetector Widget : Wraps any widget to detect gestures.

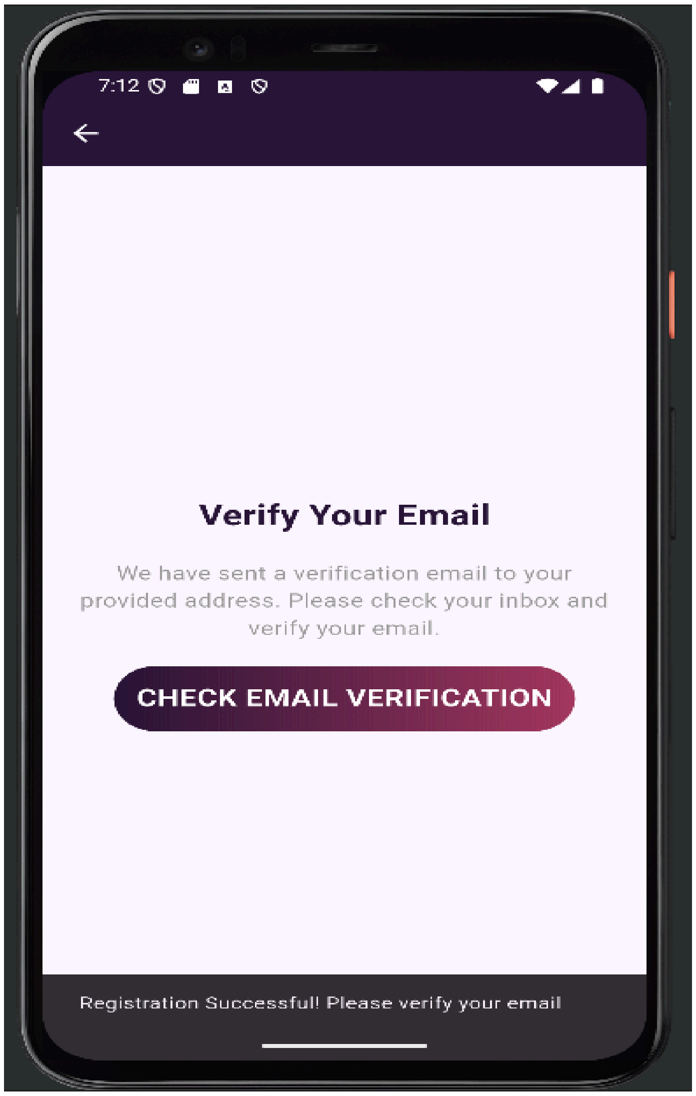
```
GestureDetector(
  onTap: () {
    print("Tapped!");
  },
  child: Container(
    color: Colors.blue,
    padding: EdgeInsets.all(16.0),
    child: Text("Tap Me"),
  ),
);
```

Common Gestures:

- onTap
- onDoubleTap
- onLongPress
- onPanUpdate (drag)
- onVerticalDragUpdate / onHorizontalDragUpdate

Output : Using route and onTap that redirect

```
Navigator.pushReplacement(
 context,
 MaterialPageRoute(builder: (context) => VerifyEmailScreen()),
);
```



7:12

Verify Your Email

We have sent a verification email to your provided address. Please check your inbox and verify your email.

CHECK EMAIL VERIFICATION

Registration Successful! Please verify your email

Conclusion :

In conclusion, navigation, routing, and gesture detection are fundamental aspects of building dynamic and user-friendly Flutter applications. Navigation allows users to move between different screens within the app, enhancing the overall user experience. Routing helps define and manage how screens are connected, whether through named routes, anonymous routes, or generated routes, making the code more organized and scalable. Additionally, gestures enable the app to respond to user interactions like taps, swipes, and drags, adding interactivity and improving usability. Mastering these features is essential for developing responsive and engaging mobile applications in Flutter.