

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

Theory

1. Navigation in Flutter

Navigation refers to the mechanism that allows users to move between different screens or pages within a Flutter app. Flutter uses the Navigator widget to manage a stack of screens, providing a way to push and pop screens.

Key Concepts:

- **Navigator: The core widget for managing a stack of routes.**
- **Routes: Represent different screens in an app.**

Push and Pop Navigation:

Push: Navigates to a new screen.

```
Navigator.push(  
  context,  
  MaterialPageRoute(builder: (context) =>  
    SecondScreen()),  
);
```

Pop: Returns to the previous screen.

```
Navigator.pop(context);
```

Named Routes:

Using named routes improves code readability:

```
Navigator.pushNamed(context, '/second');
```

2. Routing in Flutter

Routing refers to how screens or pages are mapped and displayed in an app. In Flutter, routing is managed by the MaterialApp or

CupertinoApp widget, which uses routes to determine which screen to show.

Types of Routes:

Static Routes: Defined at app launch.

```
MaterialApp(  
  routes: {  
    '/': (context) => HomeScreen(),  
    '/second': (context) => SecondScreen(),  
  },  
);
```

Dynamic Routes: Allow passing parameters when navigating

```
Navigator.push(  
  context,  
    MaterialPageRoute(builder: (context) =>  
      DetailScreen(itemId: 42)),  
);
```

3. Gestures in Flutter

Gestures are actions performed by the user on the screen, like tapping, swiping, or pinching. Flutter provides the **GestureDetector** widget to detect these actions.

Common Gestures:

Tap Gesture: Detects taps.

```
GestureDetector(onTap: () => print("Tapped"), child:  
  Container());
```

Long Press Gesture: Detects long presses.

```
GestureDetector(onLongPress: () => print("Long  
Pressed"), child: Container());
```

Swipe Gesture: Detects swipes (horizontal or vertical).

```
GestureDetector(  
  child: Container(),
```

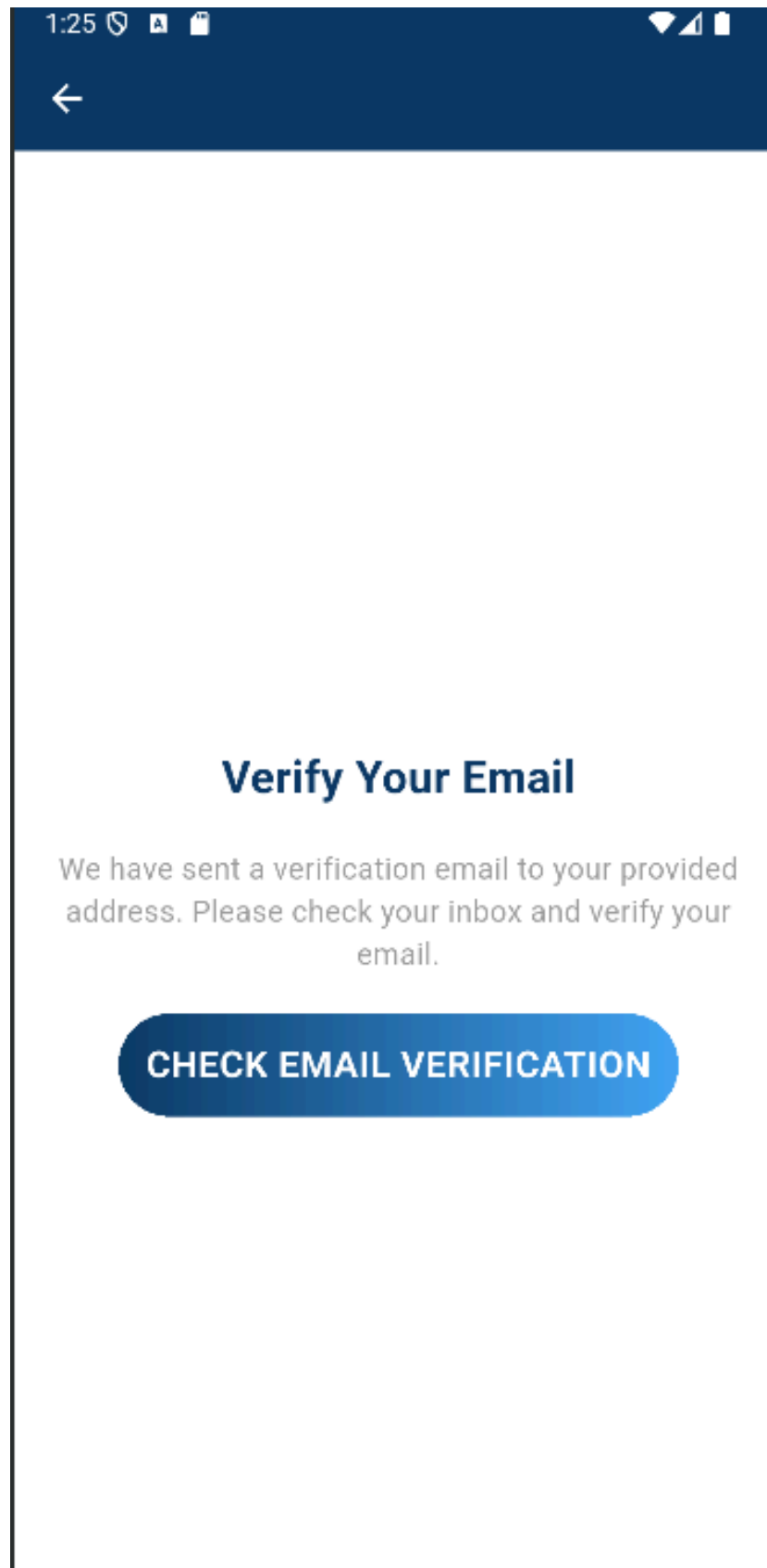
```
    onHorizontalDragEnd: (details) => print("Swiped"),  
    child: Container(),  
  );
```

Combining Navigation and Gestures:

Gestures can also trigger navigation between screens.

```
GestureDetector(  
    onHorizontalDragEnd:      (details)      =>  
Navigator.pushNamed(context, '/nextScreen'),  
    child: Container(),  
);
```

Output:



12:52



Edit Report

Incident Title

Car Theft In Andheri

Crime Type

Theft

Location

Andheri East

Incident Date

2025-03-13

Incident Description

A luxury car was stolen from a residential parking area in Andheri, leading to a citywide search. The car was tracked using the vehicle's GPS system, and it was recovered within 48 hours in the nearby suburb of Vile Parle. The suspects have been linked to a network involved in car thefts across the city.

Update Report

Conclusion

In Flutter, navigation, routing, and gestures are integral for creating interactive and user-friendly applications.

- **Navigation enables seamless transitions between screens, using methods like push and pop.**
- **Routing allows for both static and dynamic screen management, enhancing flexibility and modularity in app design.**
- **Gestures offer a way for users to interact with the app, making the experience more engaging.**

By mastering these concepts, developers can build well-structured, responsive, and intuitive applications in Flutter. These techniques are essential for handling user interactions, creating dynamic navigation flows, and optimizing the overall user experience in any mobile app.