

MPL Experiment 4

Name: Rohan Lalchandani

Class: D15A Roll no: 25

AIM-To apply navigation, routing and gestures in Flutter App

THEORY-

Navigation in Flutter allows users to move between different screens (or pages) in the app. Flutter uses the Navigator widget to handle navigation between routes (screens).

Types of Navigation

- Push Navigation (Forward Navigation) → Moves to a new screen.
- Pop Navigation (Backward Navigation) → Moves back to the previous screen.
- PushReplacement → Replaces the current screen with a new one.
- PushAndRemoveUntil → Moves to a new screen and removes previous screens from the stack.

Routing in Flutter manages different screens in the app. It helps organize and structure navigation efficiently.

Types of Routing

1. Direct Route Navigation (MaterialPageRoute)-Used for simple page-topage navigation.
2. Named Routes (Predefined Routes in main.dart)-Defined in the MaterialApp widget and used throughout the app.

Flutter uses the GestureDetector widget to detect user interactions like taps, swipes, pinches, and long presses. This is essential for making an app interactive.

Common Gestures & Their Uses:

- Tap → Detects simple taps on a widget.
- Double Tap → Recognizes double-clicking.
- Long Press → Triggers an action when the user presses and holds.
- Swipe (Drag) → Detects horizontal or vertical dragging.
- Pinch (Zoom In/Out) → Detects two-finger pinch for zooming.

SYNTAX

Navigator

```
Navigator.push(
  context,
  MaterialPageRoute(builder: (context) => SecondPage()),
);
Navigator.pushReplacement(
  context,
  MaterialPageRoute(builder: (context) => NewPage()),
);
```

```
Routing void main() {
  runApp(MaterialApp(
    initialRoute: '/',
    routes: {
      '/': (context) => HomePage(),
      '/profile': (context) => ProfilePage(),
    },
  ));
}
```

Gestures

GestureDetector(onTap:

```
() { print("Widget
```

```
Tapped!");
```

```
},
```

```
child: Container(
```

```
width: 100, height:
```

```
100, color:
```

```
Colors.blue,
```

```
),
```

```
);
```

Widget Properties Navigator context → The current

build context for navigation.

MaterialPageRoute → Creates a transition animation between pages.

builder → Defines the widget to navigate to.

Navigator.push() → Pushes a new screen on top of the stack.

Navigator.pop() → Removes the top screen and goes back.

Navigator.pushReplacement() → Replaces the current screen with a new one.

Routing initialRoute → Sets the first screen when the

app starts.

routes → Defines a map of route names and corresponding widgets.

Navigator.pushNamed() → Navigates using a predefined route.

Navigator.pop() → Closes the current screen and returns to the previous one.

Gestures onDoubleTap → Detects a double tap.

onLongPress → Detects when the user presses and holds.

onHorizontalDragStart → Detects when a horizontal drag begins.

onHorizontalDragUpdate → Detects movement during a horizontal drag.

onHorizontalDragEnd → Detects when a horizontal drag stops.

CODE

```
// Navigate to Home Page
Navigator.pushReplacement(
  context,
  MaterialPageRoute(builder: (context) => const TwitterHomePage()),
);
}
```


OUTPUT

CODE

To go the bench press exercise:


```
onTap:() {
  Navigator.push(
context,
  MaterialPageRoute(builder: (context) => const UserProfilePage()),
  );
},
```

OUTPUT


 **Chest**

Select Level ▾


Bench press






Incline press



Decline Press



 **Bench Press**



Bench Press



- 1) Lie back on a flat bench. Using a medium width grip, lift the bar from the rack and hold it straight over you with your arms locked. This will be your starting position.
- 2) From the starting position, breathe in and begin coming down slowly until the bar touches your middle chest.
- 3) After a brief pause, push the bar back to the starting position as you breathe out.

Equipment Required

Barbell, Bench , Plate, Lock

Target Muscle

Chest, Shoulder, Triceps



When clicked on introduction of widget

CODE

```
onTap: () {  
    Navigator.pushReplacement(  
        context,  
        MaterialPageRoute(builder: (context) => const TwitterLoginPage())  
    );  
},
```

OUTPUT:

