MAD & PWA LAB Experiment 5

Name: Swaraj Patil Class: D15A Roll no: 39

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

Theory:

1. Navigation in Flutter

Navigation allows users to move between screens in an app. It is managed using the Navigator and a stack-based system. The most common navigation actions include:

- Push Navigation: Moves to a new screen.
- Pop Navigation: Goes back to the previous screen.
- Named Routes: Uses predefined names for navigation instead of directly creating new screen instances.

2. Routing in Flutter

Routing defines the path and logic for screen transitions. There are two main types:

- Static Routing: Predefined routes set in MaterialApp.
- Dynamic Routing: Routes created dynamically using onGenerateRoute, allowing flexible screen navigation.

3. Gestures in Flutter

Gestures enable user interactions such as tapping, swiping, and dragging. Flutter provides:

- GestureDetector: Detects custom gestures like tap, double-tap, and long press.
- InkWell: Adds ripple effects for a better touch response.

Code:

bottomNavigationBar: BottomNavigationBar(items: const <BottomNavigationBarItem>[BottomNavigationBarItem(icon: Icon(Icons.chat), label: 'Chats'), BottomNavigationBarItem(icon: Icon(Icons.person), label: 'Profile'), BottomNavigationBarItem(icon: Icon(Icons.settings), label: 'Settings',),], currentIndex: selectedIndex,

onTap: _onItemTapped,

)

Navigation to Detail Screen

```
onTap: () {
   Navigator.push(
   context,
   MaterialPageRoute(
   builder: (context) => DemoChatScreen(userName: 'User ${index + 1}'),
   ),
   );
}
```

Navigation with Replacement

```
Navigator.of(context).pushReplacement(
    MaterialPageRoute(
    builder: (context) => const DemoLoginScreen(),
    ),
);
```

Output:





