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

Theory:

Flutter provides tools to handle navigation, routing, and gestures, allowing users to move between screens and interact with the app smoothly. These features help create a user-friendly experience in mobile applications.

---

1. Navigation in Flutter

Navigation is the process of moving between different screens (or pages) in a Flutter app. Flutter uses a stack-based approach for navigation, where new screens are pushed onto the stack and removed when the user navigates back.

Types of Navigation:

- Push Navigation: Moves to a new screen and adds it to the stack.
- Pop Navigation: Removes the current screen and returns to the previous one.
- Named Routes: Uses pre-defined route names to navigate.
- Navigation with Data: Allows passing data between screens when navigating.

---

2. Routing in Flutter

Routing helps in managing different screens efficiently. Instead of manually handling each screen transition, Flutter allows defining routes in a structured way.

Types of Routing:

- Direct Routing: Navigates to a specific screen using explicit methods.
- Named Routing: Uses a predefined route name to navigate, making the app more organized.

Routing improves app maintainability, especially in apps with multiple screens.

---

3. Gestures in Flutter

Gestures enable user interaction in Flutter applications. Flutter provides built-in gesture detection capabilities for touch-based interactions.

Common Gestures:

- Tap: A single touch interaction.
- Double Tap: Two quick consecutive taps.

- Long Press: Holding a touch for a longer duration.
- Swipe: Moving a finger across the screen.
- Drag: Moving an object by pressing and holding it.

Gestures are essential for making apps interactive and responsive.

---

4. Combining Navigation and Gestures

Navigation and gestures can be combined to enhance user experience. For example:

- Tapping on a button can navigate to another screen.
- Swiping a card can delete an item or move to another page.
- Dragging an element can reposition items within the app.

Navigation, routing, and gestures are fundamental to creating an interactive Flutter application. Navigation allows movement between screens, routing helps manage screens efficiently, and gestures enable touch interactions. Mastering these concepts helps in developing dynamic and user-friendly Flutter applications.

Code:

login.dart file

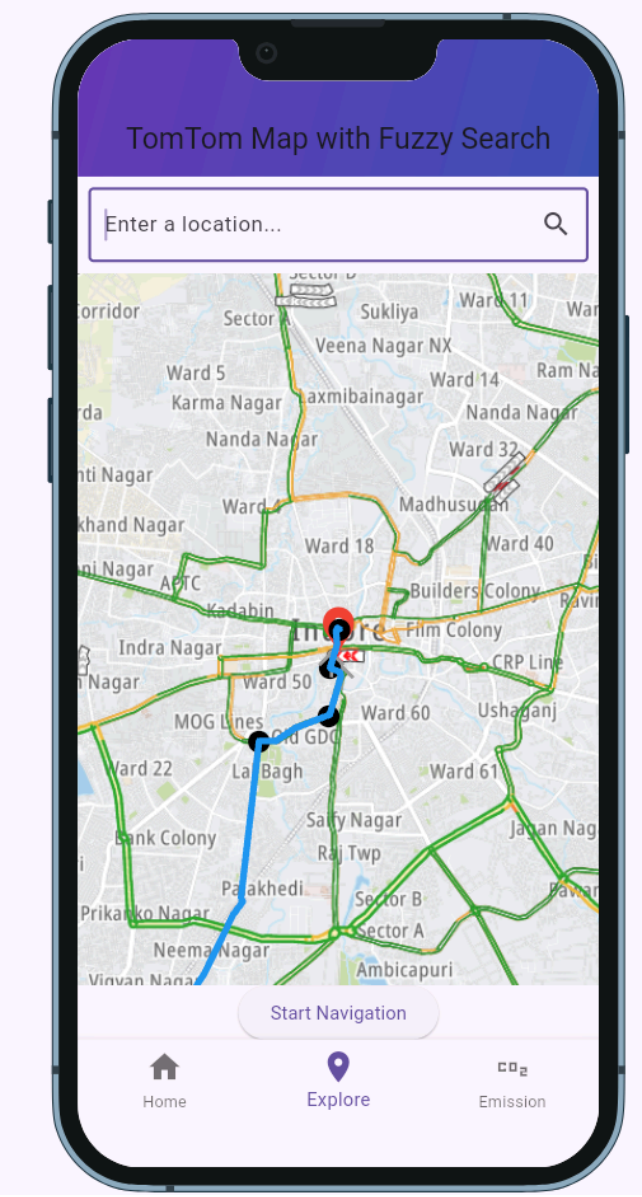
```
class _HomePageState extends
State<HomePage> {
  int currentPage = 0;
  List<Widget> pages = [
    HomeSectionPage(),
    MapPageVersion(),
    CarbonEmissionPage(),
  ];
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: IndexedStack(
        index: currentPage,
        children: pages,
      ),
      bottomNavigationBar:
BottomNavigationBar(
        type: BottomNavigationBarType.fixed,
        iconSize: 28,
        currentIndex: currentPage,
        onTap: (value) {
          setState(() {
            currentPage = value;
          });
        },
        items: [
          BottomNavigationBarItem(
            icon: Icon(Icons.home),
            label: 'Home',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.location_on),
            label: 'Explore',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.co2_sharp),
            label: 'Emission',
          ),
        ],
      ),
    );
  }
}
```

```
  Widget build(BuildContext context) {
    return Scaffold(
      body:
        Stack(
        children:
        [
          // Background Image
          Align(
            alignment:
            Alignment.topCenter, child:
            Container(
              height: 580, // Adjust the height as
needed
              decoration: BoxDecoration(
                image: DecorationImage(
                image:
AssetImage("assets/background.png
                  "), fit: BoxFit.scaleDown,
                ),
              ),
            ),
          )

Widget build(BuildContext context) {
  final List<Map<String, dynamic>> categories =
[
    {
      'title': 'Traffic Alerts',
      'icon': Icons.traffic,
      'color': Colors.red,
      'route': TrafficAlertScreen(),
    },
    {
      'title': 'Weather Updates',
      'icon': Icons.wb_sunny,
      'color': Colors.orange,
      'route': WeatherScreen(),
    },
    // {'title': 'Navigation','icon': Icons.map,'color':
Colors.blue},
    {
      'title': 'Fuel Stations',
      'icon': Icons.local_gas_station,
      'color': Colors.green,
      'route': FuelStationList(),
    },

    {
      'title': 'Emergency',
      'icon': Icons.local_hospital,
      'color': Colors.pink,
      'route': HospitalPage(),
    },
    {
      'title': 'ChatBot',
      'icon': Icons.local_hospital,
      'color': Colors.deepPurple,
      'route': ChatScreen(),
    },

  ];

  final List<Map<String, String>> videos = [
    {
      'title': 'Traffic Updates',
      'description': 'Stay updated with real-time
traffic alerts.',
      'videoUrl':

'https://flutter.github.io/assets-for-api-docs/assets
/videos/butterfly.mp4',
    },
    {
      'title': 'Weather Forecast',
      'description': 'Get accurate weather
predictions for your region.',
      'videoUrl':

'https://flutter.github.io/assets-for-api-docs/assets
/videos/bee.mp4',
    },
    {
      'title': 'Fuel Stations Nearby',
      'description': 'Find the nearest fuel stations
with ease.',
      'videoUrl':

'https://flutter.github.io/assets-for-api-docs/assets
/videos/butterfly.mp4',
    },
  ];

  bool isDarkMode =
Theme.of(context).brightness ==
Brightness.dark;

  return Scaffold(
    backgroundColor: const
Color.fromARGB(255, 187, 218, 234),
```

```
      appBar: AppBar(
        title: const Text(
          '🚛 Truck Map',
          style: TextStyle(
              fontSize: 22, fontWeight:
FontWeight.bold, color: Colors.white),
        ),
        flexibleSpace: Container(
          decoration: const BoxDecoration(
            gradient: LinearGradient(
              colors: [Colors.deepPurple,
Colors.indigo],
              begin: Alignment.topLeft,
              end: Alignment.bottomRight,
            ),
          ),
        ),
        elevation: 4,
        actions: [
          IconButton(
            onPressed: fetchTrafficAlerts,
            icon: const Icon(Icons.refresh, size: 28,
color: Colors.white),
          )
        ],
      ),
      body: ListView(
        padding: const EdgeInsets.all(12.0),
        children: [
          _buildLatestUpdateCard(isDarkMode),
          const SizedBox(height: 20),

          // Categories Section
          Wrap(
            spacing: 16.0,
            runSpacing: 16.0,
            children: categories
                .map(
                  (category) => GestureDetector(
                    onTap: () {
                      if (category.containsKey('route')) {
                        Navigator.push(
                          context,
                          MaterialPageRoute(
                            builder: (context) =>
category['route'],
                          ),
                        );
                      }
                    },
                    child: CategoryPage(
                      title: category['title'],
                      icon: category['icon'],
                      color: category['color'],
                    ),
                  ),
                )
                .toList(),
          ),
          // Suggested Videos
          const SizedBox(height: 20),
          Text(
            "Essential Driving Tutorials: Videos You
Must Watch for Safe Driving!",
            style: TextStyle(
              fontSize: 16,
              fontWeight: FontWeight.bold,
              color: Colors.black87,
            ),
            textAlign: TextAlign.center,
          ),
          Wrap(
            spacing: 16.0,
            runSpacing: 16.0,
            children: videos
                .map((video) => VideoCard(
                      title: video['title']!,
                      description: video['description']!,
                      videoUrl: video['videoUrl']!,
                    ))
                .toList(),
          ),

          SuggestedVideos(
            title: 'Rules and Regulations',
            description:
                'Drive safely with real-time traffic alerts
and navigation.',
            icon: Icons.local_police,
          ),
          const SizedBox(height: 20),
        ],
      ),
    );
  }

  Widget _buildLatestUpdateCard(bool
isDarkMode) {
```

```
    return Container(
      decoration: BoxDecoration(
        gradient: LinearGradient(
          colors: isDarkMode
              ? [Colors.deepPurple[700]!,
Colors.indigo[800]!]
              : [Colors.white, Colors.blueGrey[50]!],
          begin: Alignment.topLeft,
          end: Alignment.bottomRight,
        ),
        borderRadius: BorderRadius.circular(16),
        boxShadow: [
          BoxShadow(
            color: Colors.black.withOpacity(0.1),
            blurRadius: 10.0,
            spreadRadius: 2.0,
            offset: const Offset(0, 4),
          ),
        ],
      ),
      padding: const EdgeInsets.all(16.0),
      child: Column(
        crossAxisAlignment:
CrossAxisAlignment.start,
        children: [
          const Text(
            '📢 LATEST UPDATE',
            style: TextStyle(
                fontSize: 20,
                fontWeight: FontWeight.bold,
                color: Colors.black87),
          ),
          const SizedBox(height: 10),
          if (isLoading)
            _buildShimmerEffect()
          else if (latestUpdates.isEmpty)
            const Text(
              '🚦 No traffic alerts available',
              style: TextStyle(fontSize: 16, color:
Colors.redAccent),
            )
          else
            Column(
              children:
latestUpdates.take(3).map((update) {
                return Padding(
                  padding: const
EdgeInsets.symmetric(vertical: 6.0),
                  child: Row(
                    children: [
                      const Icon(Icons.warning,
                          color: Colors.orangeAccent, size:
24),
                      const SizedBox(width: 10),
                      Expanded(
                        child: Text(
                          update,
                          style: const TextStyle(
                              fontSize: 16, color:
Colors.black87),
                          overflow: TextOverflow.ellipsis,
                          maxLines: 2,
                        ),
                      ),
                    ],
                  ),
                );
              }).toList(),
            ),
        ],
      ),
    );
  }

  Widget _buildShimmerEffect() {
    return GestureDetector(
      onTap: () {},
      child: Column(
        children: List.generate(
          3,
          (index) => Padding(
            padding: const
EdgeInsets.symmetric(vertical: 6.0),
            child: Shimmer.fromColors(
              baseColor: Colors.grey[300]!,
              highlightColor: Colors.grey[100]!,
              child: Container(
                width: double.infinity,
                height: 20,
                decoration: BoxDecoration(
                  color: Colors.white,
                  borderRadius:
BorderRadius.circular(4),
                ),
              ),
            ),
          ),
        ),
      ),
```

Output :-


TomTom Map with Fuzzy Search
Enter a location...
Sukliya
Veena Nagar NX
Ward 5
Karma Nagar
Ward 14
Nanda Nagar
Ward 32
Madhusudan
Ward 18
Ward 40
APTC
Builders Colony
Kadabin
Film Colony
Indra Nagar
CRP Line
Ward 50
Ward 60
Ushaganj
MOG Lines
Ward 22
Ward 61
Sanjay Nagar
Raj Twp
Sector B
Sector A
Ambicapuri
Start Navigation
Home
Explore
Emission

Conclusion :-

Flutter simplifies app interaction through effective navigation, routing, and gesture handling. By using navigation and routing, developers can manage screen transitions and data flow between pages efficiently. Gesture detection enhances user experience by enabling interactive touch responses like taps and swipes. Together, these features help create smooth, intuitive, and user-friendly mobile applications.