

Experiment 05

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

Theory :

In Flutter, navigation, routing, and gestures are essential concepts for creating dynamic and interactive user interfaces.

- Navigation:

Navigation refers to the movement between different screens or pages within an app.

In Flutter, navigation is managed by the Navigator class, which maintains a stack of routes representing the app's navigation history.

Navigation can be triggered by user interactions, such as tapping on a button or selecting an item from a list, or programmatically in response to events or user input.

Flutter provides various navigation methods, such as push, pushReplacement, pop, popUntil, etc., to navigate between routes and manipulate the navigation stack.

- Routing:

Routing is the process of defining and configuring routes within a Flutter app.

A route represents a distinct screen or page in the app's UI hierarchy.

In Flutter, routes are typically defined using the MaterialApp widget's routes parameter or by manually creating instances of MaterialPageRoute or CupertinoPageRoute.

Routes can have parameters or arguments that are passed during navigation, allowing data to be shared between screens.

- Gestures:

Gestures are user actions, such as tapping, dragging, pinching, etc., that are detected and handled by the app to trigger specific actions or interactions.

In Flutter, gestures are implemented using gesture recognizer classes like GestureDetector, InkWell, DragGestureRecognizer, etc.

GestureDetector: Detects various gestures, such as taps, drags, long presses, etc., and invokes corresponding callbacks.

InkWell: A material widget that responds to taps with a splash effect. It wraps its child widget and triggers the onTap callback when tapped.

GestureDetector and InkWell can be used to make UI elements interactive and responsive to user input, enhancing the user experience.

- Gesture Recognition:

Gesture recognition is the process of identifying and interpreting user gestures to perform specific actions or trigger events.

Flutter provides built-in gesture recognizers for common gestures like taps, drags, scrolls, etc.

Developers can also implement custom gesture recognizers by subclassing the `GestureRecognizer` class and overriding its methods to detect and handle custom gestures.

Code:

a) main_page.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:linkedin_mobile_ui/pages/main/create/create_page.dart';
import 'package:linkedin_mobile_ui/pages/main/home/home_page.dart';
import 'package:linkedin_mobile_ui/pages/main/jobs/jobs_page.dart';
import 'package:linkedin_mobile_ui/pages/main/main_page/widgets/drawer_widget.dart';
import 'package:linkedin_mobile_ui/pages/main/network/network_page.dart';
import 'package:linkedin_mobile_ui/pages/main/notifications/notifications_page.dart';
import 'package:linkedin_mobile_ui/theme/styles.dart';
import 'widgets/app_bar_widget.dart';
```

```
class MainPage extends StatefulWidget {
  const MainPage({super.key});

  @override
  State<MainPage> createState() => _MainPageState();
}
```

```
class _MainPageState extends State<MainPage> {

  final GlobalKey<ScaffoldState> _scaffoldState = GlobalKey<ScaffoldState>();

  int _currentPageIndex = 0;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      drawer: const DrawerWidget(),
      key: _scaffoldState,
      appBar: _currentPageIndex == 4? appBarWidget(
```

```

context,
title: "Search Jobs",
isJobsTab: true,
onLeadingTapClickListener: () {
  setState(() {
    _scaffoldState.currentState!.openDrawer();
  });
}
):appBarWidget(
context,
title: "Search",
isJobsTab: false,
onLeadingTapClickListener: () {
  setState(() {
    _scaffoldState.currentState!.openDrawer();
  });
}
),
bottomNavigationBar: BottomNavigationBar(
  currentIndex: _currentPageIndex,
  onTap: (index) {
    setState(() {
      _currentPageIndex = index;
    });
  },
  selectedItemColor: linkedInBlack000000,
  selectedLabelStyle: const TextStyle(color: linkedInBlack000000),
  unselectedItemColor: linkedInMediumGrey86888A,
  unselectedLabelStyle: const TextStyle(color: linkedInMediumGrey86888A),
  showUnselectedLabels: true,
  items: const [
    BottomNavigationBarItem(
      icon: Icon(CupertinoIcons.house_fill),
      label: "Home",
    ),
    BottomNavigationBarItem(
      icon: Icon(FontAwesomeIcons.userGroup),
      label: "Network",
    ),
    BottomNavigationBarItem(
      icon: Icon(
        Icons.add_box,
        size: 30,
      ),
    ),
  ],

```

```

        label: "Post",
      ),
      BottomNavigationBarItem(
        icon: Icon(
          Icons.notifications,
          size: 30,
        ),
        label: "Notifications",
      ),
      BottomNavigationBarItem(
        icon: Icon(FontAwesomeIcons.briefcase),
        label: "Jobs",
      ),
    ],
  ),
  body: _switchPages(_currentPageIndex)
);
}

```

```

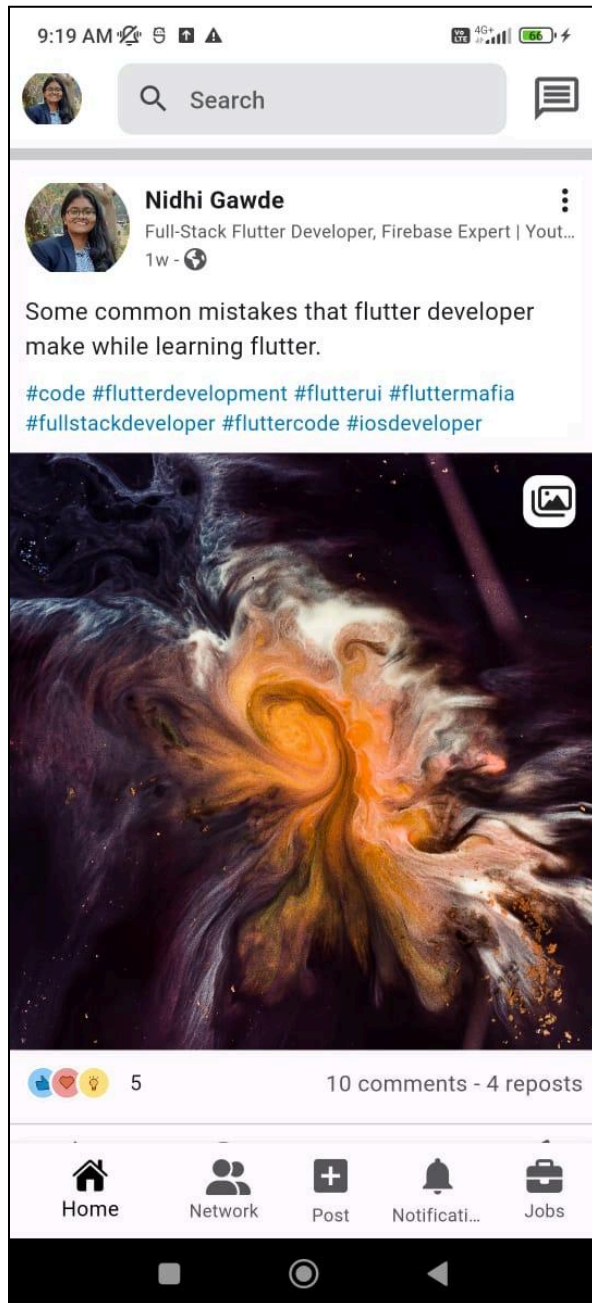
_switchPages(int index) {
  switch (index) {
    case 0:
      {
        return const HomePage();
      }
    case 1:
      {
        return const NetworkPage();
      }
    case 2:
      {
        return CreatePage(onCloneClickListener: () {
          Navigator.pop(context);
          setState(() {
            _currentPageIndex = 0;
          });
        },);
      }
    case 3:
      {
        return const NotificationsPage();
      }
    case 4:
      {

```

```

        return const JobsPage();
    }
}
}
}

```



b) Network_page.dart

```
import 'package:flutter/material.dart';
```

```

import 'package:linkedin_mobile_ui/data/network_entity.dart';
import
'package:linkedin_mobile_ui/pages/main/network/widgets/single_network_user_widget.d
art';
import 'package:linkedin_mobile_ui/theme/styles.dart';

class NetworkPage extends StatefulWidget {
  const NetworkPage({super.key});

  @override
  State<NetworkPage> createState() => _NetworkPageState();
}

class _NetworkPageState extends State<NetworkPage> {

  List<NetworkEntity> networkData = NetworkEntity.networkData;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        child: Column(
          children: [
            const SizedBox(
              height: 10,
            ),
            const Padding(
              padding: EdgeInsets.symmetric(horizontal: 10.0),
              child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                  Text(
                    "Manage my network",
                    style: TextStyle(
                      fontSize: 18,
                      fontWeight: FontWeight.bold,
                      color: linkedInBlue0077B5),
                  ),
                  Icon(
                    Icons.arrow_forward_ios,
                    color: linkedInMediumGrey86888A,
                  ),
                ],
              ),
            ),
          ],
        ),
      ),
    );
  }
}

```

```
),
const SizedBox(
  height: 15,
),
Container(
  width: double.infinity,
  height: 8,
  color: linkedInLightGreyCACCCCE,
),
const SizedBox(
  height: 15,
),
const Padding(
  padding: EdgeInsets.symmetric(horizontal: 10.0),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      Text(
        "Invitations",
        style: TextStyle(
          fontSize: 18,
          fontWeight: FontWeight.bold,
          color: linkedInBlue0077B5),
      ),
      Icon(
        Icons.arrow_forward_ios,
        color: linkedInMediumGrey86888A,
      ),
    ],
  ),
),
const SizedBox(
  height: 15,
),
Container(
  width: double.infinity,
  height: 8,
  color: linkedInLightGreyCACCCCE,
),
const SizedBox(
  height: 15,
),
```

```

GridView.builder(
  padding: const EdgeInsets.symmetric(horizontal: 10),
  shrinkWrap: true,
  physics: const ScrollPhysics(),
  itemCount: networkData.length,
  gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
    crossAxisCount: 2,
    mainAxisSpacing: 6,
    crossAxisSpacing: 6,
    childAspectRatio: 0.6),
  itemBuilder: (context, index) {
    final network = networkData[index];
    return SingleNetworkUserWidget(network: network);
  },
)
],
),
),
);
}
}

```

c) network_entity.dart

```

class NetworkEntity {

  final String? userBgImage;
  final String? userProfileImage;
  final String? username;
  final String? userBio;
  final num? mutualConnections;

  NetworkEntity(
    {this.userBgImage,
    this.userProfileImage,
    this.username,
    this.userBio,
    this.mutualConnections});

  static List<NetworkEntity> networkData = [

    NetworkEntity(

```



```
    userBgImage: "bg_image_1.jpeg",  
    userProfileImage: "profile_2.jpeg",  
    mutualConnections: 13,  
    userBio: "Flutter Developer & Advocate",  
    username: "ISTE",  
  ),
```

```
NetworkEntity(  
  userBgImage: "bg_image_2.png",  
  userProfileImage: "profile_1.jpeg",  
  mutualConnections: 22,  
  userBio: "Senior Software Engineer",  
  username: "",  
),
```

```
NetworkEntity(  
  userBgImage: "bg_image_1.jpeg",  
  userProfileImage: "profile_2.jpeg",  
  mutualConnections: 52,  
  userBio: "UX/UI Researcher & Designer",  
  username: "Diana Joe",  
),
```

```
NetworkEntity(  
  userBgImage: "bg_image_3.jpeg",  
  userProfileImage: "profile_1.jpeg",  
  mutualConnections: 13,  
  userBio: "Android Developer at Google",  
  username: "Stephan Covey",  
),
```

```
NetworkEntity(  
  userBgImage: "bg_image_1.jpeg",  
  userProfileImage: "profile_2.jpeg",  
  mutualConnections: 88,  
  userBio: "Flutter Developer & Advocate",  
  username: "Elon Musk",  
),
```

```
NetworkEntity(  
  userBgImage: "bg_image_2.png",  
  userProfileImage: "profile_1.jpeg",  
  mutualConnections: 11,  
  userBio: "Flutter Developer & Advocate",
```

```
    username: "Robert Frost",  
  ),
```

```
NetworkEntity(  
  userBgImage: "bg_image_3.jpeg",  
  userProfileImage: "profile_2.jpeg",  
  mutualConnections: 13,  
  userBio: "Flutter Developer & Advocate",  
  username: "Steve Wozniak",  
),
```

```
NetworkEntity(  
  userBgImage: "bg_image_3.jpeg",  
  userProfileImage: "profile_1.jpeg",  
  mutualConnections: 76,  
  userBio: "Flutter Developer & Advocate",  
  username: "Doug Stevenson",  
),  
];
```

```
}
```

