# EXPERIMENT 5

**AIM:** To apply navigation, routing, and gestures in Flutter App

**THEORY:**

**Flutter Drawer**

The mobile apps that use Material Design have two primary options for navigation. These navigations are Tabs and Drawers. A drawer is an alternative option for tabs because sometimes the mobile apps do not have sufficient space to support tabs.

A drawer is an invisible side screen. It is a sliding left menu that generally contains important links in the application and occupies half of the screen when displayed.

Flutter uses a drawer widget to create a sliding left menu layout with a Material Design widget. The following steps are required to use a drawer in the app.

1. Create a Flutter Project.

2. Add drawer in scaffold widget

3. Populate the drawer by adding content

4. Close the drawer.

In the main.dart file, create a drawer in the scaffold widget as the code given below.

```
Scaffold(
 drawer: Drawer(
 child: // Populate the Drawer by adding content in the next step.
 )
);
```

Next, we need to add content in the drawer. In this example, we are going to use the Listview widget that

allows the users to scroll through the drawer if the content does not fit in the screen supports. The following

code explains it more clearly.

```
Drawer(
 child: ListView(
 padding: EdgeInsets.zero,
 children: <Widget>[
 DrawerHeader(
 child: Text('Drawer Header'),
 decoration: BoxDecoration(
 color: Colors.blue, ), ),
ListTile(
 title: Text('Item 1'),
 onTap: () {
 // Update the state of the app.
 // ...
 },
 ),
 ListTile(
 title: Text('Item 2'),
 onTap: () {
 // Update the state of the app.
 // ...
 },
 ),
 ],
 ),
);
```

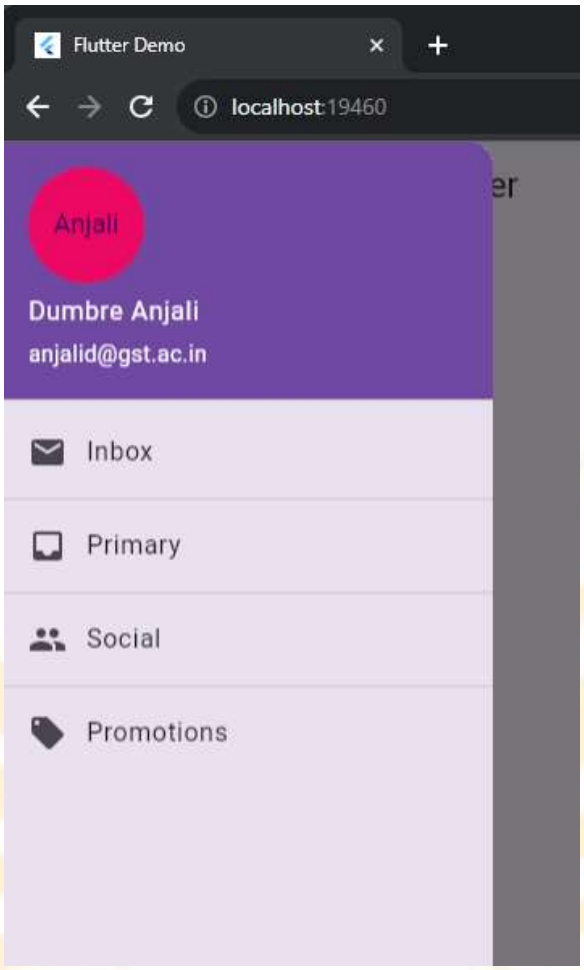
main.dart × widget_test.dart

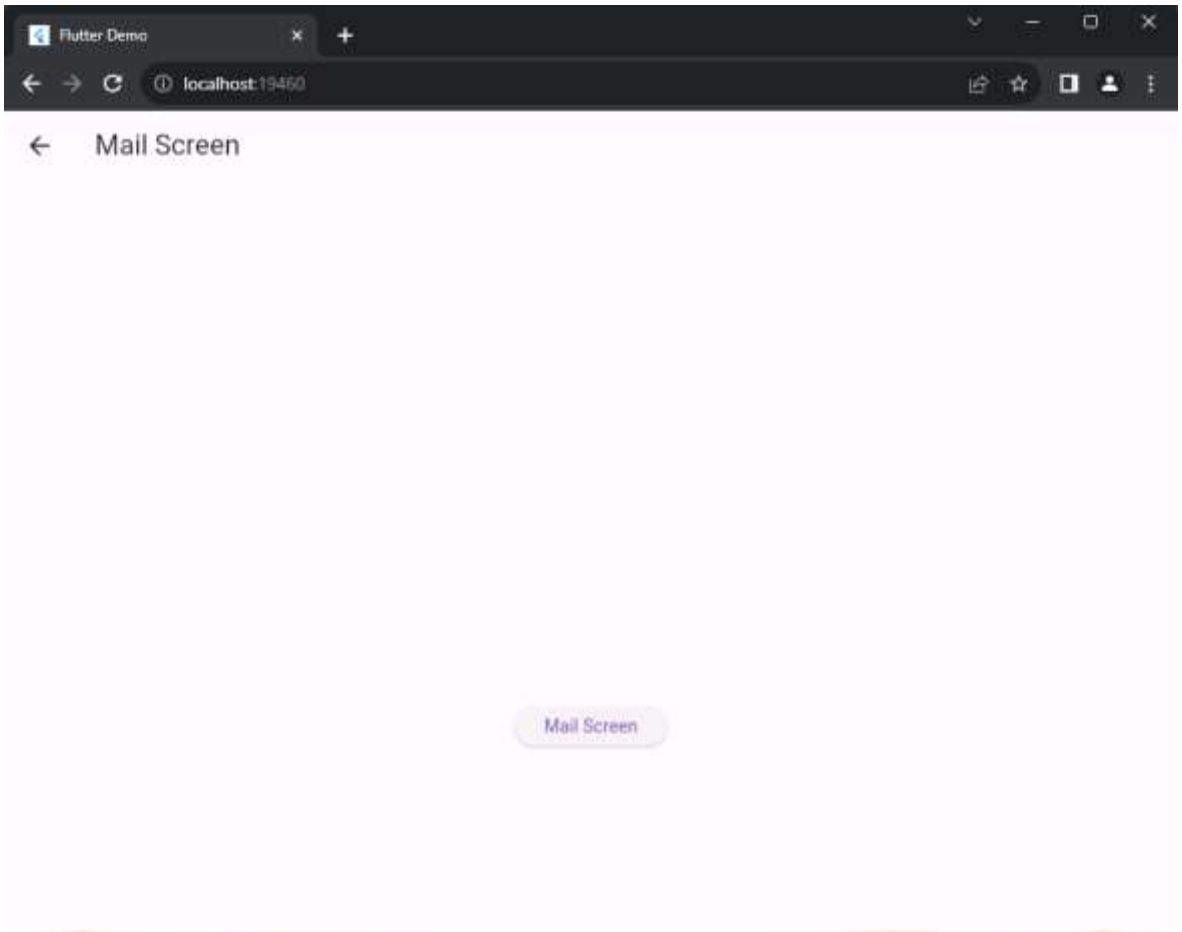
app > lib > main.dart > _MyStatefulWidgetState > build

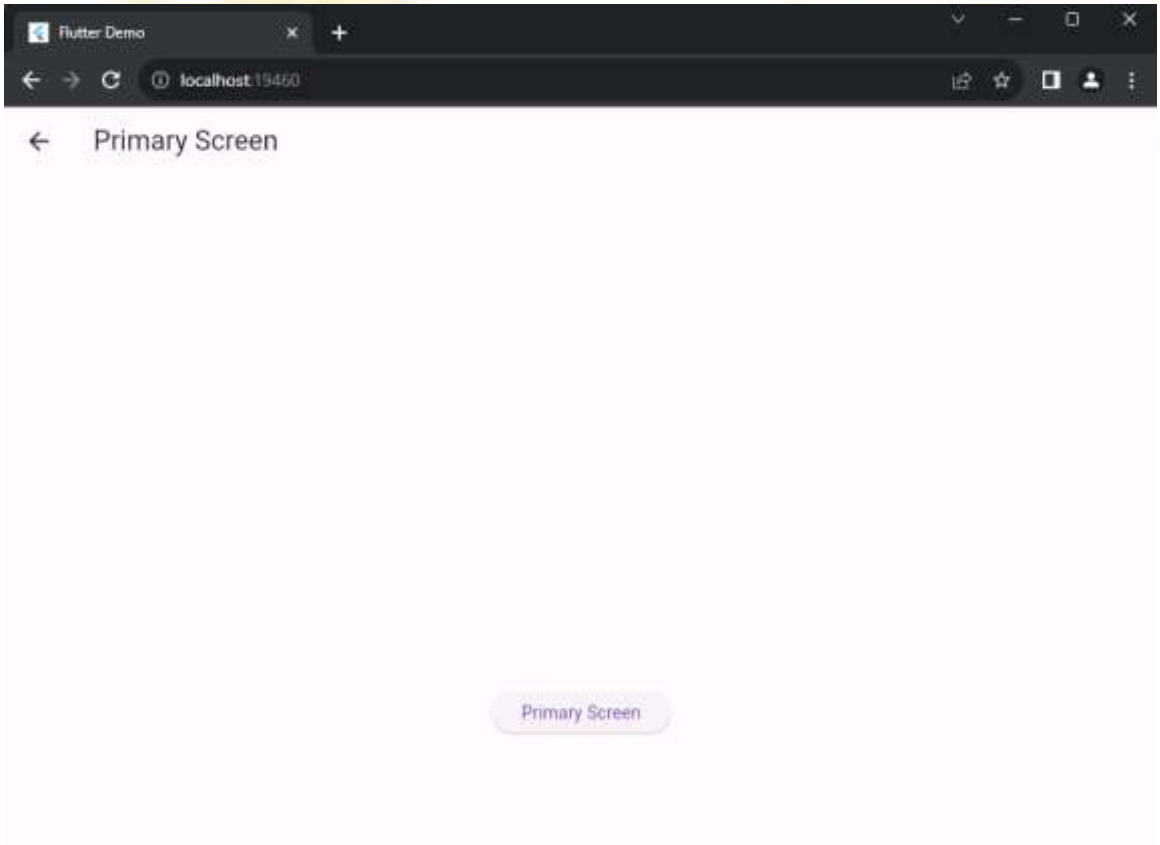
```dart
import 'package:flutter/material.dart';

Run | Debug | Profile
void main() {
  runApp( MyApp());
}

class MyApp extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: MyStatefulWidget(),
    ); // MaterialApp
  }
}

class MyStatefulWidget extends StatefulWidget {
  const MyStatefulWidget({Key? key}) : super(key: key);

  @override
  State<MyStatefulWidget> createState() => _MyStatefulWidgetState();
}

class _MyStatefulWidgetState extends State<MyStatefulWidget> {


  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Flutter NavigationDrawer Example'),
      ), // AppBar
```

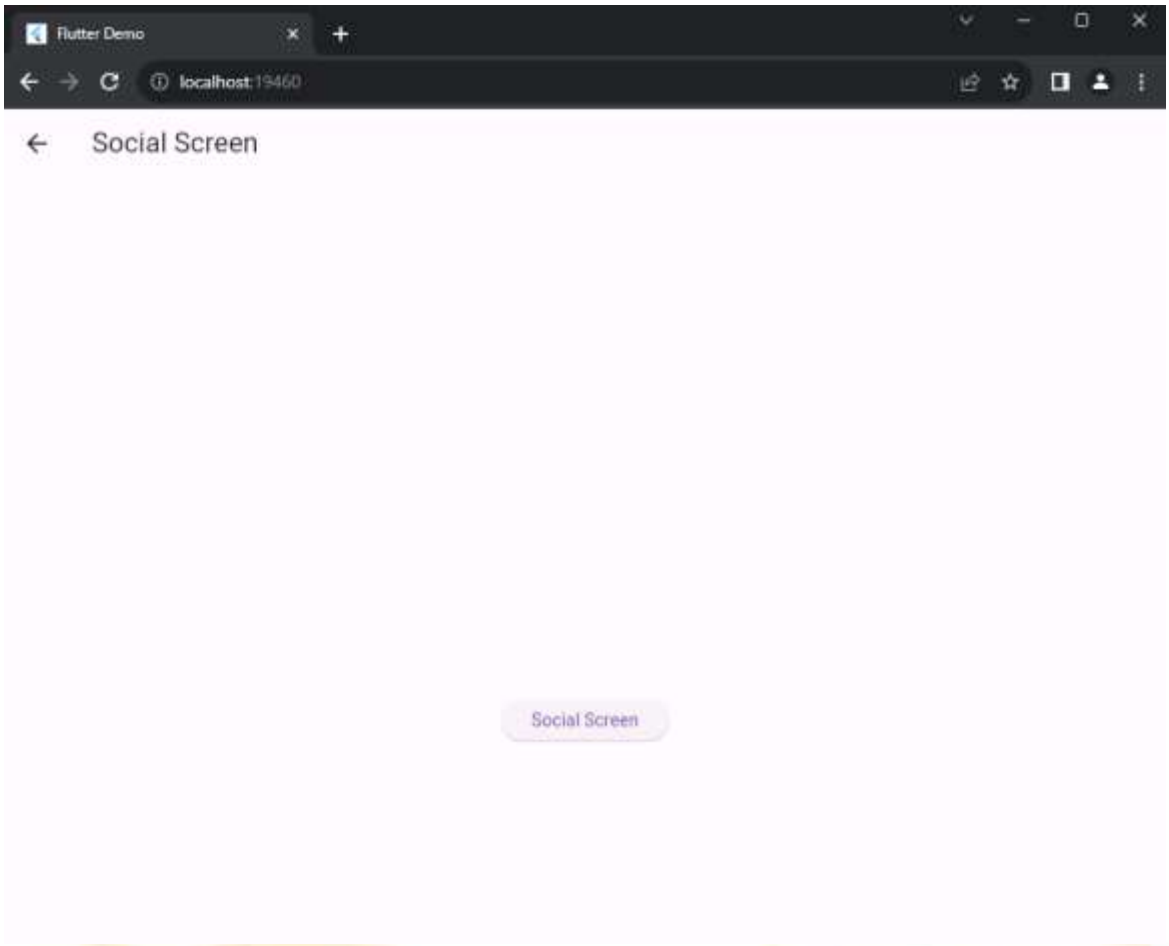
```
33        drawer: Drawer(
34          elevation: 20.0,
35          child: Column(
36            children: <Widget>[
37              const UserAccountsDrawerHeader(
38              accountName: Text("Shantanu Gobade"),
39              accountEmail: Text("shantanu.gobade@gmail.com"),
40              currentAccountPicture: CircleAvatar(
41                backgroundColor: Colors.pink,
42                child: Text("Shantanu"),
43              ), // CircleAvatar
44            ), // UserAccountsDrawerHeader
45            ListTile(
46              title: const Text("Inbox"),
47              leading: const Icon(Icons.inbox),
48              onTap: () {
49                Navigator.pop(context);
50                Navigator.push(
51                  context,
52                  MaterialPageRoute(builder: (context) => Mailpage()),
53                  );
54              },
55            ), // ListTile
56            const Divider(
57              height: 0.1,
58            ), // Divider
59            ListTile(
60              title: const Text("Primary"),
61              leading: const Icon(Icons.inbox),
62              onTap: () {
63                Navigator.pop(context);
64                Navigator.push(context,
65                MaterialPageRoute(builder: (context) => Primarypage()),
```

```
                );
              },
            ), // ListTile
            ListTile(
              title: const Text("Social"),
              leading: new Icon(Icons.people),
            ), // ListTile
              ListTile(
                title: const Text("Promotions"),
                leading: new Icon(Icons.local_offer),
              ), // ListTile
            ], // <Widget>[]
          ) // Column
        ), // Drawer
      ); // Scaffold
    }
  }

  class Mailpage extends StatelessWidget {

    @override
    Widget build(BuildContext context) {
      return Scaffold(
        appBar: AppBar(
          title: Text("Mail Screen"),
        ), // AppBar
        body: Center(
          child: ElevatedButton(onPressed: () {
            Navigator.pop(context);
          },
          child: Text('Mail Screen')
          ), // ElevatedButton
        ), // Center
      ); // Scaffold
    }
  }

```


Flutter Demo
localhost:19460
Anjali
Dumbre Anjali
anjalid@gst.ac.in
Inbox
Primary
Social
Promotions


Flutter Demo
localhost:19460
Mail Screen
Mail Screen


Flutter Demo
localhost:19460
Primary Screen
Primary Screen




**Bottom Navigation Bar:**

The bottom navigation bar in Flutter can contain multiple items such as text labels, icons, or both. It allows

the user to navigate between the top-level views of an app quickly. If we are using a larger screen, it is better

to use a side navigation bar.

In Flutter application, we usually set the bottom navigation bar in conjunction with the scaffold widget.

Scaffold widget provides a Scaffold.bottomNavigationBar argument to set the bottom navigation bar. It is to

note that only adding BottomNavigationBar will not display the navigation items. It is required to set the

BottomNavigationItems for Items property that accepts a list of BottomNavigationItems widgets.

 We can display only a small number of widgets in the bottom navigation that can be 2 to 5.

 It must have at least two bottom navigation items. Otherwise, we will get an error.

 It is required to have the icon and title properties, and we need to set relevant widgets for them.

Properties of the BottomNavigationBar Widget

The following are the properties used with the bottom navigation bar widget:

items: It defines the list to display within the bottom navigation bar. It uses argument

BottomNavigationBarItem that contains sup-properties given below:

```
const BottomNavigationBarItem({
 @required this.icon,
 this.title,
 Widget activeIcon,
 this.backgroundColor,
 })
```

currentIndex: It determines the current active bottom navigation bar item on the screen.

onTap: It is called when we tapped one of the items on the screen.

iconSize: It is used to specify the size of all bottom navigation item icons.

fixedColor: It is used to set the color of the selected item. If we have not set a color to the icon or title, it will

be shown.

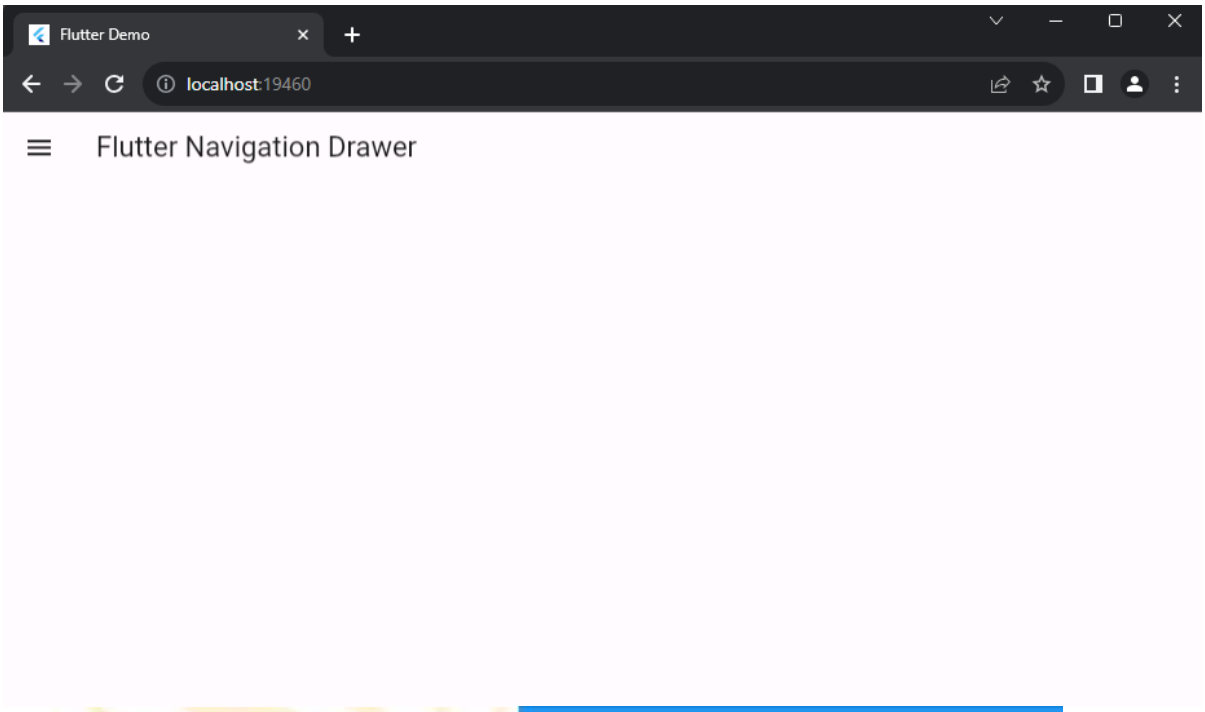
type: It determines the layout and behavior of a bottom navigation bar. It behaves in two different ways that

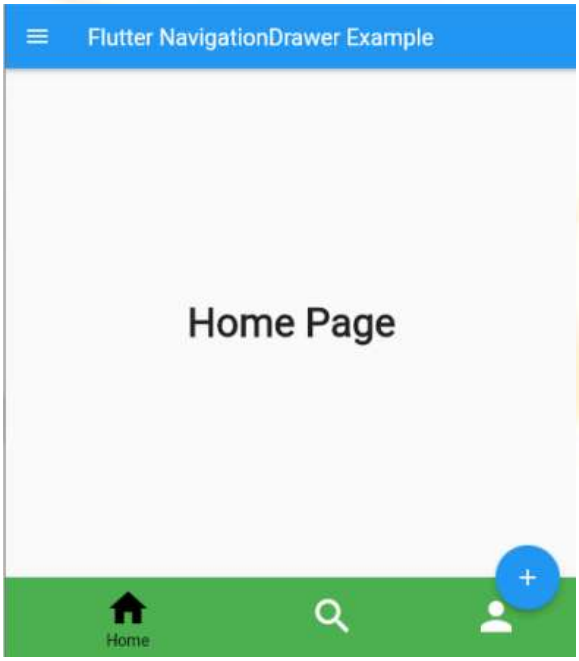
are: fixed and shifting. If it is null, it will use fixed. Otherwise, it will use shifting where we can see an

animation when we click a button.

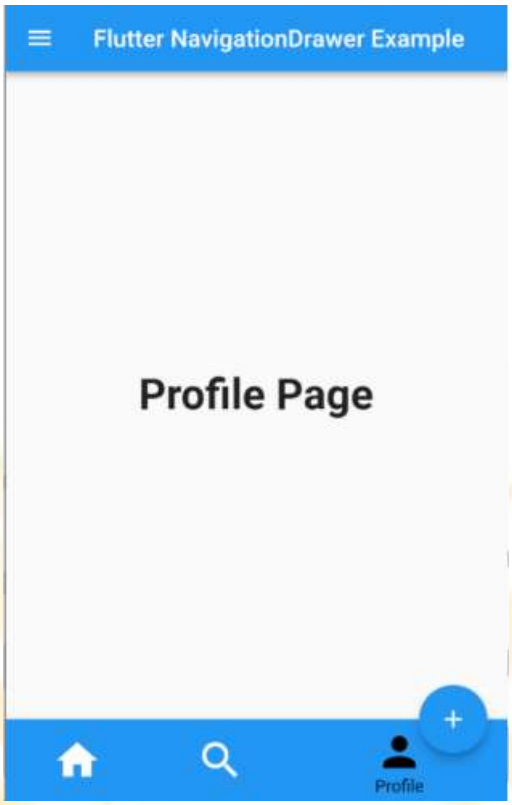
```dart
import 'package:flutter/material.dart';

Run | Debug | Profile
void main() {
  runApp(MyApp());
}

/// This Widget is the main application widget.
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: MyStatefulWidget(),
    ); // MaterialApp
  }
}

class MyStatefulWidget extends StatefulWidget {
  MyStatefulWidget({Key? key}) : super(key: key);

  @override
  _MyStatefulWidgetState createState() => _MyStatefulWidgetState();
}

class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  int _count = 0;
  int _selectedIndex = 0;
  static const List<Widget> _widgetOptions = <Widget>[
    Text('Home Page',        style: TextStyle(fontSize: 35, fontWeight: FontWeight.bold)),
    Text('Search Page',
        style: TextStyle(fontSize: 35, fontWeight: FontWeight.bold)), // Text
    Text('Profile Page',
        style: TextStyle(fontSize: 35, fontWeight: FontWeight.bold)), // Text
  ]; // <Widget>[]

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }
```

```
    body: Center(
      child: _widgetOptions.elementAt(_selectedIndex),
    ), // Center

    bottomNavigationBar: BottomNavigationBar(
        items: const <BottomNavigationBarItem>[
          BottomNavigationBarItem(
              icon: Icon(Icons.home),
              title: Text('Home'),
              backgroundColor: Colors.green), // BottomNavigationBarItem
          BottomNavigationBarItem(
              icon: Icon(Icons.search),
              title: Text('Search'),
              backgroundColor: Colors.yellow), // BottomNavigationBarItem
          BottomNavigationBarItem(
            icon: Icon(Icons.person),
            title: Text('Profile'),
            backgroundColor: Colors.blue,
          ), // BottomNavigationBarItem
        ], // <BottomNavigationBarItem>[]
        type: BottomNavigationBarType.shifting,
        currentIndex: _selectedIndex,
        selectedItemColor: Colors.black,
        iconSize: 40,
        onTap: _onItemTapped,
        elevation: 5), // BottomNavigationBar
    //);
    //floating button
    floatingActionButton: FloatingActionButton(
      onPressed: () => setState(() {
        _count++;
      }),
      tooltip: 'Increment Counter',
      child: const Icon(Icons.add),
    ), // FloatingActionButton

    floatingActionButtonLocation: FloatingActionButtonLocation.endDocked,
  ); // Scaffold
 }
}
```


Flutter Demo
localhost:19460
Flutter Navigation Drawer


Flutter NavigationDrawer Example
Home Page
Home


Flutter NavigationDrawer Example
Search Page
Search




**Conclusion:** Hence, we have successfully designed and applied navigation and routing in flutter.