# Lab 12

Instructor: Sidra Khatoon E-mail: skhatoon[@uit.edu](mailto:ad@uit.edu)

# Objective

# The objective of lab is exploring Drawer and tab Navigation.

**Student Information**

|  |  |
| --- | --- |
| **Student Name** |  |
| **Student ID** |  |
| **Date** |  |

**Assessment**

|  |  |
| --- | --- |
| **Marks Obtained** |  |
| **Remarks** |  |
| **Signature** |  |

# Objective

# The objective of lab is exploring Drawer and tab Navigation.

# Instructions

You have to perform the following tasks yourselves. Raise your hand if you face any difficulty in understanding and solving these tasks. **Plagiarism** is an abhorrent practice and you should not engage in it.

# How to Submit?

Submit lab work using Teams.

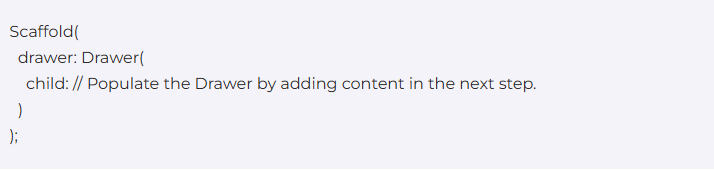
**Drawer Navigation:**

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.

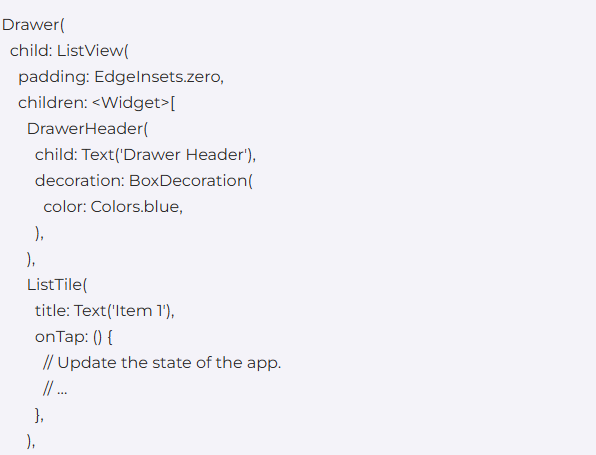
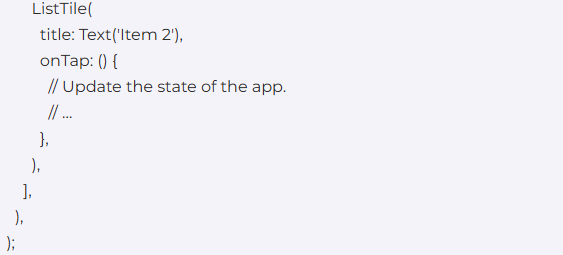
A drawer is an invisible side screen. It is a sliding left menu that generally contains important links in the application and occupies half of the screen when displayed.

Let us see how the drawer works in a Flutter. 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.

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.

Finally, close the drawer. We can do this by using the **navigator**.

**Tab Navigation**

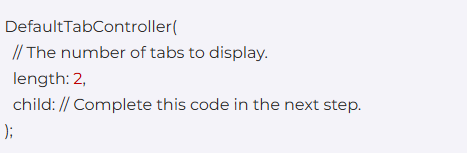
The tabs are mainly used for mobile navigation. Tab are placed on bottom and Top as per designer requirement.

Working with tabs is a common pattern in Android and iOS apps that follow the Material Design guidelines. Flutter provides a convenient way to create a tab layout. To add tabs to the app, we need to create a TabBar and TabBarView and attach them with the TabController. The controller will sync both so that we can have the behavior which we need.

Step 1: First, you need to create a Flutter project in your IDE.

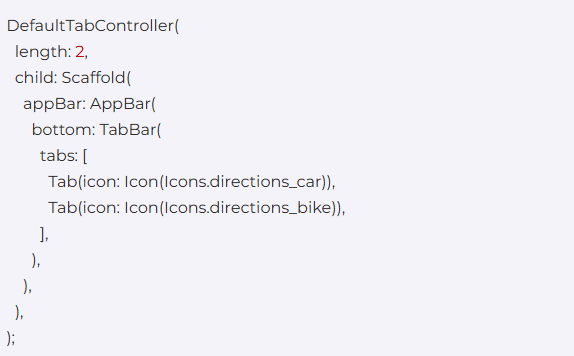
Step 2: Open the app in IDE and navigate to the lib folder. Inside the lib folder, create two dart files and named it as per your requirement.

Step 3: Next, we need to create a DefaultTabController. The DefaultTabController creates a TabController and makes it available to all widgets.

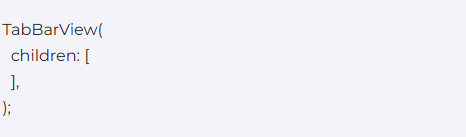


In the above code, the length property tells about the number of tabs used in the app.

Step 4: Create the tab. We can create tabs by using the TabBar widget as below code.



Step 5: Create content for each tab so that when a tab is selected, it displays the content. For this purpose, we have to use the TabBarView widget as:



**Assessments:**

Q1: What is the difference between Tab Navigator and Drawer Navigator in Flutter, and when would you use each for designing app navigation?

Q2: Explain the advantages of using a Drawer Navigator in apps with complex navigation requirements.

Q3: What are the key differences in user experience between Tab Navigator and Drawer Navigator?

Q4: In what scenarios would you prefer a Tab Navigator over a Drawer Navigator, and why?