



Al-Mustaqbal University
College of Science
Intelligent Medical System Department



جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

College of Sciences
Intelligent Medical System Department

Lecture 9

Drawer Widget in Flutter

Subject: Application Development

Level: Third

Lecturer: Asst. Lect. Ali Saleem Haleem

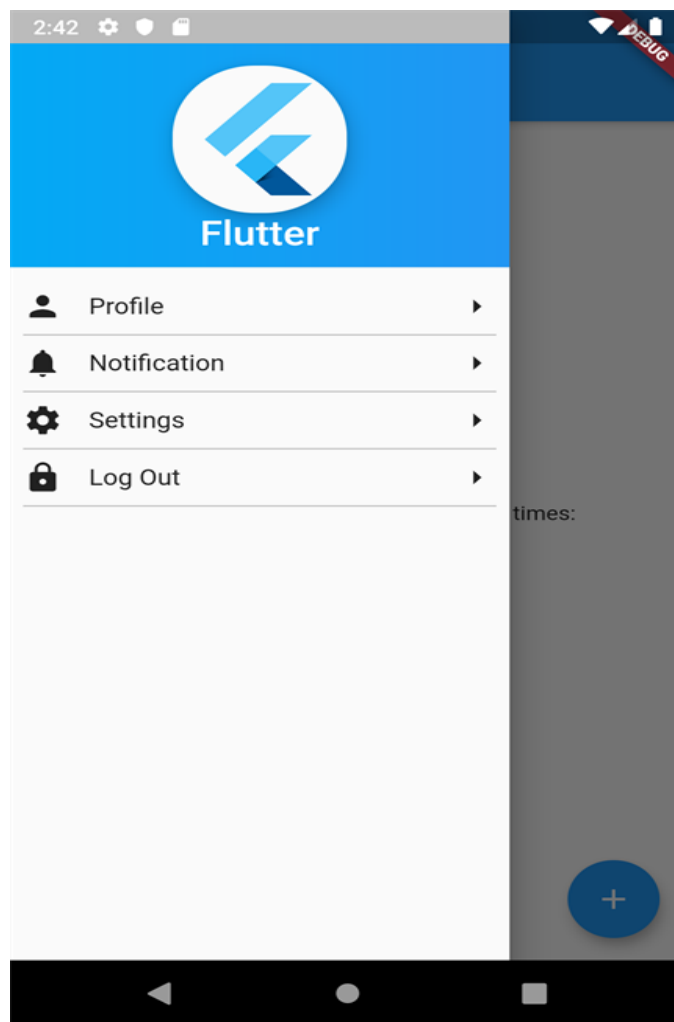


Introduction

Modern mobile applications rely on clear and intuitive navigation structures. Common navigation patterns include:

- **Bottom Navigation Bar**
- **TabBar Navigation**
- **Drawer (Side Navigation Menu)**

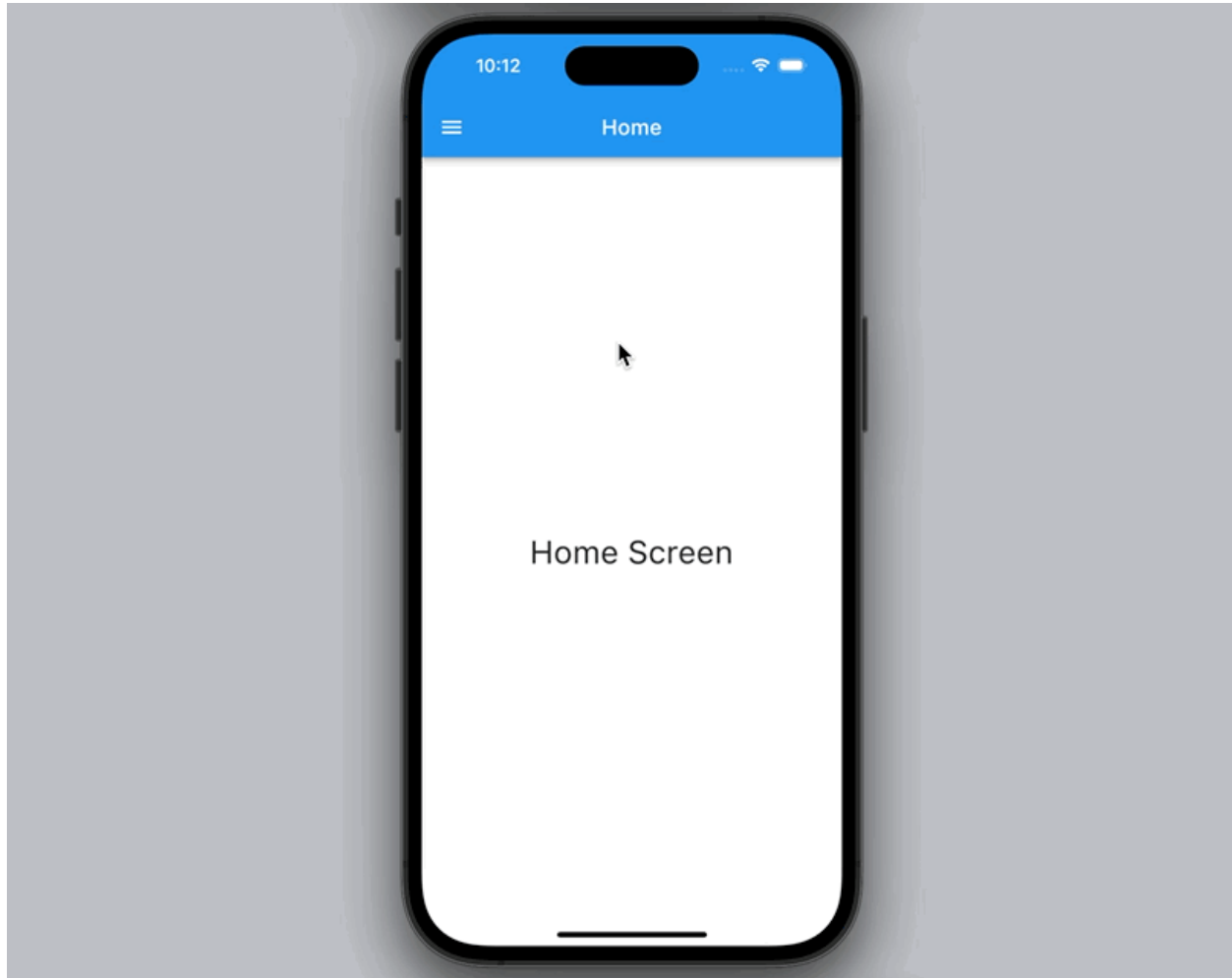
The **Drawer Widget** is used when an application contains multiple sections or settings that the user needs to access through a collapsible side panel.





What is the Drawer Widget?

The **Drawer** is a Flutter widget that provides a **side navigation menu** that slides from the left (or right) side of the screen.



It is typically placed inside the Scaffold widget using:

drawer: Drawer(...)

A Drawer can contain:

- *A header*
- *Navigation links*



- *Icons and menu items*
- *User account information*
- *Application settings*

Relationship Between Scaffold, AppBar, and Drawer

- **Scaffold** provides the basic layout structure for a Flutter screen.
- When a Drawer is added to the Scaffold:
 - The **AppBar** automatically displays the hamburger menu icon (☰).
 - Swiping from the left edge opens the Drawer.

Example:

```
Scaffold(  
  appBar: AppBar(title: Text('Drawer Example')),  
  drawer: Drawer(child: Text('Drawer Content')),  
  body: Center(child: Text('Main Page')),  
);
```

Building a Basic Drawer

A standard Drawer contains a scrollable list of items:

```
Drawer(  
  child: ListView(  
    padding: EdgeInsets.zero,  
    children: [  
      DrawerHeader(  
        decoration: BoxDecoration(color: Colors.indigo),  
        child: Text('Menu'),  
      ),  
    ],  
  ),  
);
```



```
ListTile(  
  leading: Icon(Icons.home),  
  title: Text('Home'),  
)  
ListTile(  
  leading: Icon(Icons.settings),  
  title: Text('Settings'),  
)  
],  
)
```

DrawerHeader – The Drawer’s Top Section

The DrawerHeader is used to display important information such as:

- *App name*
- *Logo*
- *Version number*
- *User details*

Example:

```
DrawerHeader(  
  decoration: BoxDecoration(color: Colors.indigo),  
  child: Column(  
    crossAxisAlignment: CrossAxisAlignment.start,  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: [  
      Text(  
        'Smart Medical App',  
        style: TextStyle(  
          color: Colors.white,  
          fontSize: 22,  
          fontWeight: FontWeight.bold,  
        ),  
      ),  
      SizedBox(height: 8),  
    ],  
  ),  
)
```



```
Text(  
  'Version 1.0.0',  
  style: TextStyle(color: Colors.white70),  
),  
],  
),  
)
```

UserAccountsDrawerHeader

This widget provides a professionally designed user account section following Material Design guidelines.

It allows you to display:

- *User's name*
- *Email*
- *Profile picture*
- *Optional multiple account pictures*

Example:

```
UserAccountsDrawerHeader(  
  accountName: Text('Student Name'),  
  accountEmail: Text('student@future.edu.iq'),  
  currentAccountPicture: CircleAvatar(  
    child: Text('S'),  
  ),  
  decoration: BoxDecoration(color: Colors.teal),  
)
```

ListTile for Menu Items

The **ListTile** widget is commonly used to create individual menu items inside the Drawer.



Example:

```
ListTile(  
  leading: Icon(Icons.dashboard),  
  title: Text('Dashboard'),  
  subtitle: Text('Overview of statistics'),  
  trailing: Icon(Icons.arrow_forward_ios, size: 16),  
  onTap: () {  
    // Action on tap  
  },  
)
```

Components of ListTile:

Property	Description
<i>leading</i>	Icon or widget at the start of the tile
<i>title</i>	Main text label
<i>subtitle</i>	Secondary descriptive text
<i>trailing</i>	Widget shown at the end of the tile (e.g., arrow)
<i>onTap</i>	Function executed when the tile is pressed

Navigating Between Screens from Drawer

To navigate from a Drawer item, we must:

1. **Close the Drawer**
2. **Navigate to the new screen**

```
onTap: () {  
  Navigator.pop(context); // Close Drawer  
  Navigator.push(  
    context,
```



```
MaterialPageRoute(builder: (context) => DashboardScreen()),  
);  
}
```

Best Practices When Using Drawer

1. Keep the menu concise and organized.
2. Use meaningful icons and clear labels.
3. Group related items together.
4. Always call `Navigator.pop(context)` before navigating.
5. Use `UserAccountsDrawerHeader` when account information is needed.

Full Practical Example

```
import 'package:flutter/material.dart';  
  
void main() {  
  runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Drawer Demo',  
      home: MainScreen(),  
    );  
  }  
}  
  
class MainScreen extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(title: Text('Drawer Demo')),  
      drawer: Drawer(  
        child: ListView(  
          padding: EdgeInsets.zero,  
          children: [  
            UserAccountsDrawerHeader(  
              accountName: Text('Student Name'),
```




Al-Mustaqbal University
College of Science
Intelligent Medical System Department

```
accountEmail: Text('student@future.edu.iq'),
currentAccountPicture: CircleAvatar(child: Text('S')),
),
ListTile(
  leading: Icon(Icons.home),
  title: Text('Home'),
  onTap: () {
    Navigator.pop(context);
  },
),
ListTile(
  leading: Icon(Icons.settings),
  title: Text('Settings'),
  onTap: () {
    Navigator.pop(context);
    Navigator.push(context,
      MaterialPageRoute(builder: (context) => SettingsScreen()));
  },
),
],
),
),
body: Center(child: Text('Main Screen Content')),
);
}
}

class SettingsScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Settings')),
      body: Center(child: Text('Settings Screen')),
    );
  }
}
```