https://blog.logrocket.com/how-to-build-a-bottom-navigation-bar-in-flutter/

Link:

What is BottomNavigationBar in Flutter?

BottomNavigationBar is a widget that displays a row of small widgets at the bottom of a Flutter app. Usually, it's used to show around three to five items. Each item must have a label and an icon. BottomNavigationBar allows you to select one item at a time and quickly navigate to a given page.

Now let's walk through the process of creating a simple BottomNavigationBar step by step. The finished product will look as follows:

```
 ),
```

Showing BottomNavigationBar

The BottomNavigationBar widget is given to the bottomNavigationBarproperty of Scaffold:

```
Scaffold(
  appBar: AppBar(
    title: const Text('BottomNavigationBar Demo'),
  ),
  bottomNavigationBar: BottomNavigationBar(
    items: const <BottomNavigationBarItem>[
      BottomNavigationBarItem(
        icon: Icon(Icons.call),
        label: 'Calls',
      ),
      BottomNavigationBarItem(
        icon: Icon(Icons.camera),
        label: 'Camera',
      ),
      BottomNavigationBarItem(
        icon: Icon(Icons.chat),
        label: 'Chats',
      ),
    ],
  ),
);
```

BottomNavigationBar has a required property called items. items accept a widget of a type BottomNavigationBarItem. BottomNavigationBarItem is simply used to show the actual item inside BottomNavigationBar.

The above code just displays the BottomNavigationBar with the first item selected as the default. It does not change the selection yet as we click on the other items:

Showing a selection of items

To show the selection of other items, we’ll use two properties: onTap and currentIndex.

```
int _selectedIndex = 0; //New

BottomNavigationBar(

  items: const <BottomNavigationBarItem>[

    ...

  currentIndex: _selectedIndex, //New

  onTap: _onItemTapped,        //New

)

//New

void _onItemTapped(int index) {

  setState(() {

    _selectedIndex = index;

  });

}
```

The _selectedIndex variable holds the value of the currently selected item. _selectedIndex is given to the currentIndex property.

The _onItemTapped() callback is assigned to onTap of BottomNavigationBar, which returns the index when the item is tapped. Simply assigning a currently selected item index to _selectedIndex and doing setState will show the item as selected in BottomNavigationBar.

Displaying the page of the selected item

As of now, we don’t have any page to show based on the selected item. So let’s go ahead and create it:

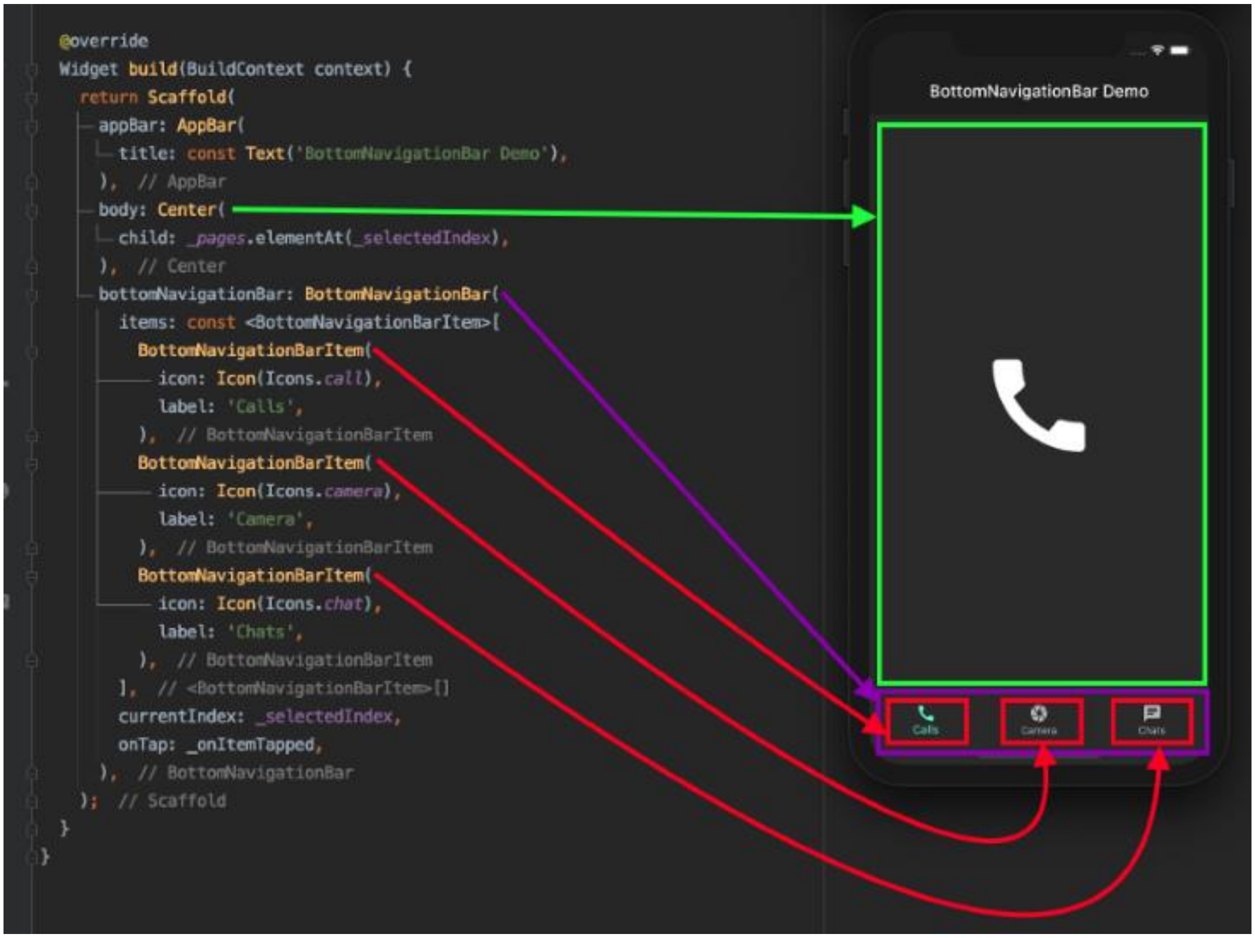
```
//New
static const List<Widget> _pages = <Widget>[
 Icon(
  Icons.call,
  size: 150,
 ),
 Icon(
  Icons.camera,
  size: 150,
 ),
 Icon(
  Icons.chat,
    size: 150,
  ),
];
Scaffold(
  appBar: AppBar(
    title: const Text('BottomNavigationBar Demo'),
  ),
  body: Center(
    child: _pages.elementAt(_selectedIndex), //New
  ),
  bottomNavigationBar: BottomNavigationBar(
    ...
  ),
);
```

_pages hold a list of widgets. For simplicity, we’re just showing a big icon of the item itself.

Showing one page in the center of the screen from _pages based on the _selectedIndex of the item will do the rest of the magic.

Now we have BottomNavigationBar up and running:

The illustration below shows how the code translates into the design:



```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('BottomNavigationBar Demo'),
    ), // AppBar
    body: Center(
      child: _pages.elementAt(_selectedIndex),
    ), // Center
    bottomNavigationBar: BottomNavigationBar(
      items: const <BottomNavigationBarItem>[
        BottomNavigationBarItem(
          icon: Icon(Icons.call),
          label: 'Calls',
        ), // BottomNavigationBarItem
        BottomNavigationBarItem(
          icon: Icon(Icons.camera),
          label: 'Camera',
        ), // BottomNavigationBarItem
        BottomNavigationBarItem(
          icon: Icon(Icons.chat),
          label: 'Chats',
        ), // BottomNavigationBarItem
      ], // <BottomNavigationBarItem>[]
      currentIndex: _selectedIndex,
      onTap: _onItemTapped,
    ), // BottomNavigationBar
  ); // Scaffold
}
}
```

Customizing the BottomNavigationBar

BottomNavigationBar has a lot of options to customize it per your need. Let's zoom in on some of the properties you can customize.

Background color

You may want to change the background color of the BottomNavigationBar to match your brand. You do that simply by using the backgroundColor property.

Mouse cursor

When running on the web, you can customize the mouse cursor when it hovers over an item on the BottomNavigationBar:

```
BottomNavigationBar(
  mouseCursor: SystemMouseCursors.grab,
  items: const <BottomNavigationBarItem>[
    ...
  ],
```