

Experiment 2

Aim: To design Flutter UI by including common widgets.

Theory:

Widgets: Each element on a screen of the Flutter app is a widget. The view of the screen completely depends upon the choice and sequence of the widgets used to build the apps. The structure of the code of an app is a tree of widgets.

- **Flutter Scaffold:** Scaffold is a layout structure provided by Flutter that provides a framework for implementing the basic material design visual layout structure of the Flutter app.
- **Flutter Container:** Container is a widget that allows you to customize its appearance using properties like color, padding, margin, and more. It's a basic building block for creating layouts in Flutter.
- **Flutter Row & Column:** Row and Column are layout widgets used for arranging child widgets horizontally (Row) or vertically (Column).
- **Flutter Text:** Text widget is used to display a string of text with a single style.
- **Flutter Buttons:** Buttons are interactive widgets that trigger actions when tapped. Flutter provides various button widgets like RaisedButton, FlatButton, IconButton, etc.
- **Flutter Icons:** Icons widget is used to display Material icons.

- Flutter Images: Image widget is used to display images.
- Bottom Navigation Bar: BottomNavigationBar widget is used to provide navigation between top-level views in the app.

The code in main.dart:

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

import 'home_page.dart';

class SearchPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    List<String> boxTexts = [
      'Python',
      'Java',
      'Dart',
      'Flutter',
      'React',
      'Angular',
      'C++',
      'Ruby',
      'Swift',
      'Kotlin',
      'HTML',
      'CSS'
    ];

    List<String> categories = [
      'Development',
      'Business',
      'Design',
      'Technology',
      'Finance',
      'Education',
      'Health',
      'Food',
      'Sports',
      'Entertainment'
    ];

    return Scaffold(
```

```

appBar: PreferredSize(
  preferredSize: Size.fromHeight(10), // Adjust the height as needed
  child: AppBar(
    backgroundColor: Colors.black,
  ),
),
backgroundColor: Colors.black,
body: Column(
  crossAxisAlignment: CrossAxisAlignment.stretch,
  children: [
    Padding(
      padding: const EdgeInsets.all(16.0),
      child: Row(
        children: [
          Icon(
            Icons.search,
            color: Colors.white,
          ),
          SizedBox(width: 30),
          Text('Search', style: TextStyle(color: Colors.white, fontSize: 19.0)),
        ],
      ),
    ),
    Padding(
      padding: const EdgeInsets.symmetric(horizontal: 16.0),
      child: Text(
        'Top Searches',
        style: TextStyle(
          fontSize: 20,
          fontWeight: FontWeight.bold,
          color: Colors.white,
        ),
      ),
    ),
    SizedBox(height: 10),
    Expanded(
      child: GridView.count(
        crossAxisCount: 4,
        children: List.generate(
          12,
          (index) {
            String text = index < boxTexts.length ? boxTexts[index] : "";
            return Container(
              margin: EdgeInsets.all(8),
              width: 60, // Adjust the width here
              height: 20 + (text.length * 2.0), // Adjust the height based on text length
              decoration: BoxDecoration(

```

[illegible]

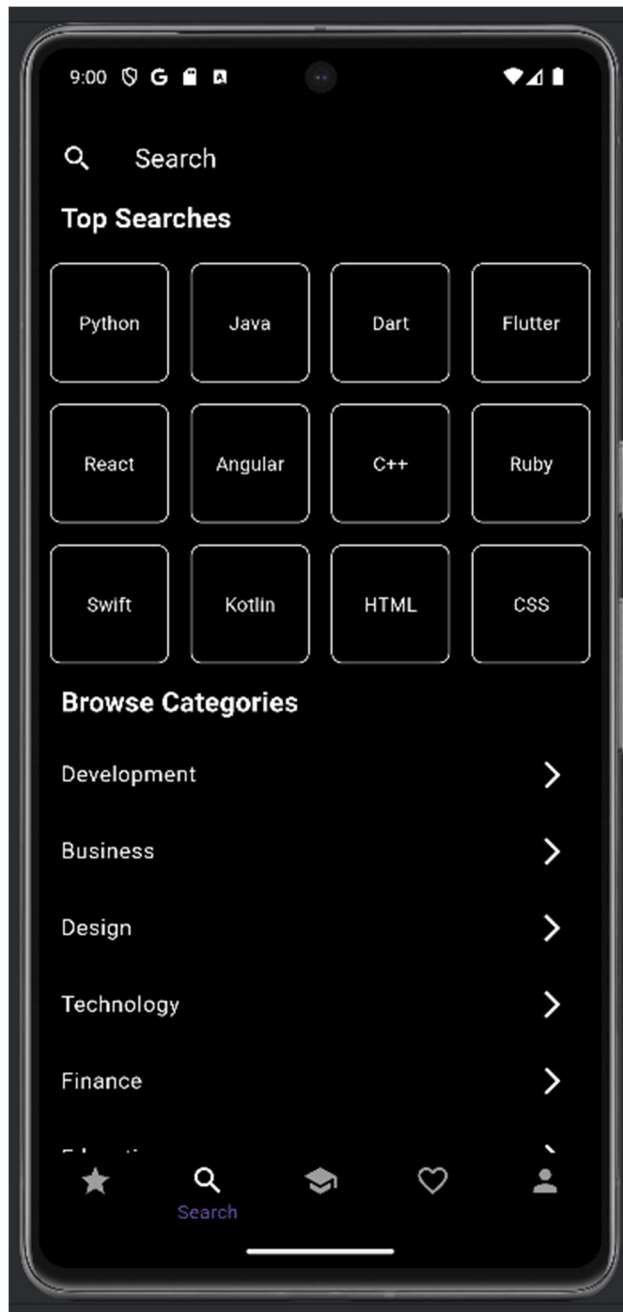
```

selectedLabelStyle: const TextStyle(color: Colors.white),
unselectedIconTheme: const IconThemeData(color: Colors.grey),
unselectedLabelStyle: const TextStyle(color: Colors.grey),
iconSize: 26.0,
selectedFontSize: 14.0,
unselectedFontSize: 12.0,
currentIndex: 1, // Set the index for the "Search" option
onTap: (index) {
  if (index == 0) {
    // If "Featured" option is tapped, navigate to the home page
    Navigator.pushReplacement(
      context,
      MaterialPageRoute(builder: (context) => HomePage()),
    );
  }
},
type: BottomNavigationBarType.fixed,
items: const [
  BottomNavigationBarItem(icon: Icon(Icons.star), label: 'Featured'),
  BottomNavigationBarItem(icon: Icon(Icons.search), label: 'Search'),
  BottomNavigationBarItem(
    icon: Icon(Icons.school), label: 'My Learning'),
  BottomNavigationBarItem(
    icon: Icon(Icons.favorite_border), label: 'Wishlist'),
  BottomNavigationBarItem(icon: Icon(Icons.person), label: 'Profile'),
],
),
);
}
}

void main() {
  runApp(MaterialApp(
    home: SearchPage(),
  ));
}

```

Output:



In the code, the following widgets are used:

1] Scaffold: Provides a standard layout structure for the app's pages. It includes functionality like app bars, drawers, and bottom navigation.

2] PreferredSize: Specifies the preferred size for a widget. In this code, it's used to set the preferred size for the AppBar.

3] Icon: Displays an icon. Used to display the search icon in the app bar.

4] Text: Displays a string of text. Used to display the "Search" text next to the search icon in the app bar.

5] Padding: Adds padding around its child widget. Used to add padding around the search icon and text.

6] Row: Arranges its children widgets in a horizontal row. Used to place the search icon and text horizontally.

7] SizedBox: A box with a specified size. Used to add spacing between the search icon/text and the "Top Searches" text.

8] GridView: Arranges its children widgets in a two-dimensional grid. Used to display the "Top Searches" items in a grid layout.

9] ListView.builder: Builds a list on demand. Used to display the "Browse Categories" list dynamically based on the categories list.

10] ListTile: Represents a single fixed-height row that contains one to three lines of text, along with an optional leading and trailing icon. Used to display each category item in the "Browse Categories" list.

11] BottomNavigationBar: Represents a material design bottom navigation bar. Used to display navigation options at the bottom of the screen.

12] MaterialApp: Represents the root widget of a Flutter application. Used to configure the top-level properties of the app and specify the home page.