EXPERIMENT NO.2

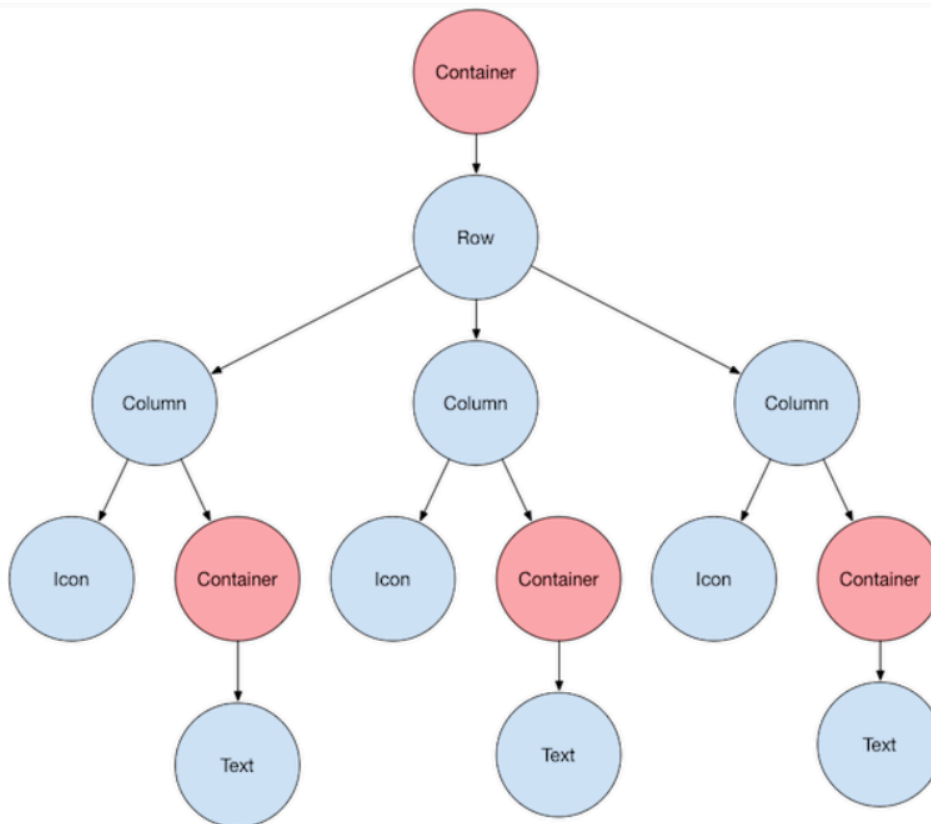|	
Experiment No 2: **To design Flutter UI by including common widgets.**	
ROLL NO	**28**
NAME	**Anushka Karhadkar**
CLASS	**D15-B**
SUBJECT	**MAD & PWA Lab**
LO-MAPPED	

Aim : To design Flutter UI by including common widgets.

Theory:
The core of Flutter's layout mechanism is widgets. In Flutter, almost everything is a widget—even layout models are widgets. The images, icons, and text that you see in a Flutter app are all widgets. But things you don't see are also widgets, such as the rows, columns, and grids that arrange, constrain, and align the visible widgets.

Widget Tree Diagram :

Common Widgets in Flutter:

1. Scaffold:

    The Scaffold widget is a basic skeletal structure that provides the visual structure of the screen.
    It typically includes an AppBar for the top bar, a Drawer for a side menu, a BottomNavigationBar for bottom navigation, and a body for the main content.

2. Container:

   The Container widget is a box model that allows you to contain and style other widgets.
   It's often used to provide padding, margins, or a background color to its child widgets.

3. Column:

   The Column widget is a flexible layout widget that arranges its children in a vertical array.
   It's commonly used to stack widgets vertically. You can control the alignment and spacing of its children.

4. ListView:

   The ListView widget is used to create a scrollable list of widgets.
   It can be used with a finite list of children, or you can use a builder function to create children dynamically based on the data source.

5. Text:

   The Text widget displays a string of text with a given style.
   You can customize the font size, color, and style of the text.

6. ElevatedButton:

   The ElevatedButton widget is a material design raised button.
   It responds to touches by elevating when pressed and provides a callback when pressed.

7. Row:

   The Row widget is similar to the Column but arranges its children horizontally in a horizontal array.It allows you to create a horizontal layout for your widgets.
   You can control the alignment and spacing of its children similar to Column.

Syntax:
1. Column:

```
Column(
  mainAxisAlignment: MainAxisAlignment.start, // or .center, .end, .spaceAround,
.spaceBetween, .spaceEvenly
  crossAxisAlignment: CrossAxisAlignment.start, // or .center, .end
  mainAxisSize: MainAxisSize.max, // or .min
  children: <Widget>[
    // Widgets to be arranged vertically
    Widget1(),
    Widget2(),
    Widget3(),
  ],
)
```

2. Row:

```
Row(
  mainAxisAlignment: MainAxisAlignment.start, // or .center, .end, .spaceAround,
.spaceBetween, .spaceEvenly
  crossAxisAlignment: CrossAxisAlignment.start, // or .center, .end
  mainAxisSize: MainAxisSize.max, // or .min
  children: <Widget>[
    // Widgets to be arranged horizontally
    Widget1(),
    Widget2(),
    Widget3(),
  ],
)
```

3. Container:

```
Container(
  alignment: Alignment.center, // Alignment for the child within the container
  margin: EdgeInsets.all(16.0), // Empty space to be placed outside the container
  padding: EdgeInsets.all(16.0), // Empty space to be placed inside the container
  color: Colors.blue, // Background color of the container
  width: 200.0, // Width of the container
  height: 100.0, // Height of the container
```

```
  child: YourChildWidget(), // Widget to be placed inside the container
)
```

4. Scaffold:

```
Scaffold(
  appBar: AppBar(
    title: Text('Your App Title'),
    // Other app bar properties like actions, leading, etc.
  ),
  body: YourBodyWidget(), // Main content of the scaffold
  floatingActionButton: FloatingActionButton(
    onPressed: () {
      // Add functionality for the floating action button press
    },
    child: Icon(Icons.add),
  ),
)
```

5. Text:

```
Text(
  'Your Text Here',
  style: TextStyle(
    fontSize: 16.0, // Font size of the text
    fontWeight: FontWeight.bold, // Font weight of the text
    color: Colors.black, // Color of the text

  ),
)
```

6. Elevated Button:

```
ElevatedButton(
  onPressed: () {
    // Add functionality for button press
  },
  style: ElevatedButton.styleFrom(
    primary: Colors.blue, // Background color of the button
    onPrimary: Colors.white, // Text color of the button
```

```
      elevation: 4.0, // Elevation (shadow) of the button
      // Other styling properties like padding, shape, textStyle, etc.
    ),
    child: Text(
      'Your Button Label',
      style: TextStyle(
        fontSize: 16.0, // Font size of the button text
      ),
    ),
)
```

7. ListView:

```
ListView(
  scrollDirection: Axis.vertical, // Axis.horizontal for a horizontal list
  padding: EdgeInsets.all(8.0), // Padding around the entire list
  children: <Widget>[
    // List of widgets goes here
    ListTile(
      title: Text('Item 1'),
    ),
    ListTile(
      title: Text('Item 2'),
    ),
    // Add more ListTiles or other widgets as needed
  ],
)
```

Code:

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

void main() {
  runApp(ProfilePage());
}

class ProfilePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Profile Page'),
          backgroundColor: Colors.deepPurple,
        ),
        body: buildProfilePage(),
      ),
    );
  }

  Widget buildProfilePage() {
    return ListView(
      children: [
        Container(
          padding: const EdgeInsets.all(16.0),
          child: const Column(
            crossAxisAlignment: CrossAxisAlignment.center,
            children: [
              SizedBox(height: 16.0),
              Text(
                'Anushka Karhadkar',
                style: TextStyle(
                  fontSize: 24.0,
                  fontWeight: FontWeight.bold,
                ),
              ),
              Text(
                'Student',
```

```dart
            style: TextStyle(
              fontSize: 16.0,
              color: Colors.grey,
            ),
          ),
        ],
      ),
    ),
    Container(
      padding: const EdgeInsets.all(16.0),
      child: const Text(
        'A coding enthusiast who enjoys coding and tries to find out different solutions to
a coding problem.',
        style: TextStyle(
          fontSize: 16.0,
          color: Colors.black,
        ),
      ),
    ),
    Container(
      padding: const EdgeInsets.all(16.0),
      child: const Text(
        'Projects:',
        style: TextStyle(
          fontSize: 18.0,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
    buildProjectItem('Project 1', 'Crop Predictor App'),
    buildProjectItem('Project 2.', 'Clothing Webapp'),
    buildProjectItem('Project 3', 'PreSchool Tutor App'),
    // Add more project items as needed
    Container(
      padding: const EdgeInsets.all(16.0),
      child: ElevatedButton(
        onPressed: null, // No action assigned
        child: Text(
          'Contact',
          style: TextStyle(
```

```dart
          color: Colors.white, // Set the text color to white
        ),
      ),
      style: ButtonStyle(
        backgroundColor: MaterialStateProperty.all(Colors.deepPurple),
      ),
    ),
  ],
  );
}

Widget buildProjectItem(String projectName, String projectDescription) {
  return Container(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(
          projectName,
          style: const TextStyle(
            fontSize: 18.0,
            fontWeight: FontWeight.bold,
          ),
        ),
        const SizedBox(height: 8.0),
        Text(
          projectDescription,
          style: const TextStyle(
            fontSize: 16.0,
            color: Colors.grey,
          ),
        ),
      ],
    ),
  );
}
}
```

Output:

Profile Page

Anushka Karhadkar
Student

A coding enthusiast who enjoys coding and tries to find out different solutions to a coding problem.

Projects:

Project 1
Crop Predictor App

Project 2.
Clothing Webapp

Project 3
PreSchool Tutor App

Contact