**LAB SESSION 6**

**Lab Title: Exploring Container Widgets in Flutter**

**Objective:**

The objective of this lab is to introduce learners to container widgets in Flutter, understand their role in building user interfaces, and practice creating layouts using different container widgets.

**Introduction to Widgets:**

In Flutter, widgets are the building blocks of UI elements, and everything that you see on the screen is a widget. Widgets can be structural (like rows, columns, and containers) or stylistic (like text, buttons, and images). They are arranged in a hierarchical structure to create complex UIs.

**Single-child widgets:**

Single-child widgets in Flutter can have only one child widget. These widgets are used when you want to display a single piece of content or layout. Examples of single-child widgets include Container, Center, Align, SizedBox, AspectRatio, etc. These widgets are typically used to wrap other widgets to apply specific layout or styling properties.

**Single-child Widget Examples:**

Container: A widget that allows you to customize its properties like padding, margin, color, etc., and can contain a single child widget.

Center: A widget that centers its child widget within itself.

SizedBox: A widget that forces its child to have a specific width, height, or both. Align: A widget that aligns its child within itself according to a specified alignment.

**Multi-child widgets:**

Multi-child widgets in Flutter can have multiple child widgets.These widgets are used when you want to display multiple pieces of content or layout in a structured manner. Examples of multi child widgets include Row, Column, Stack, ListView, GridView, etc. These widgets allow you to arrange their child widgets horizontally, vertically, or in a customized layout.

**Multi-child Widget Examples:**

Row: A widget that arranges its children in a horizontal line.

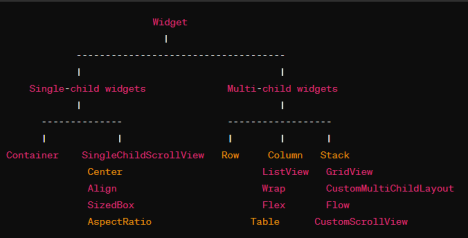
Column: A widget that arranges its children in a vertical column.

Stack: A widget that overlays its children on top of each other, allowing you to position them using coordinates.

ListView: A widget that displays a scrollable list of children in a vertical layout. GridView: A widget that displays a scrollable grid of children with a specified number of columns.

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATION DEVELOPMENT (SE-487)

These single-child and multi-child widgets form the building blocks of Flutter UIs, allowing developers to create complex and dynamic layouts for their applications.



**Container Widgets in Flutter:**

Container widgets are a fundamental part of building layouts in Flutter. They allow you to customize the visual appearance of your UI elements and control their layout properties such as alignment, padding, margin, and decoration. Commonly used container widgets in Flutter include Container, Row, Column, Stack, ListView, GridView, etc.it is a widget that combines positioning and sizing of the child widgets.it is also a class to store one or more widgets and position them on the screen according to our needs.generally it is similar to a box for storing contents.it allows many attributes to be used for decorating its child widget such as using margin which separates containers with other contents

**Ctrl+ space =see the properties that you can add**

**Ctrl + S =hot reload feature makes use JIT compiler and reflects the changes instantly.**

import 'package:flutter/material.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATION DEVELOPMENT (SE-487)

return MaterialApp(

title:'flutter demo',

theme:ThemeData(

primarySwatch:Colors.*blue*,

),

home:const MyHomePage(),

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({Key?key}):super(key:key);

@override

State<MyHomePage> createState() => \_MyHomePageState(); }

class \_MyHomePageState extends State<MyHomePage>

{

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

backgroundColor: Colors.*blue*,

title: Text('Flutter Container'),

),

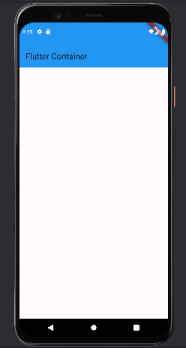
body: Container(),

);

}

}

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATION DEVELOPMENT (SE-487)

To remove the debug mode 

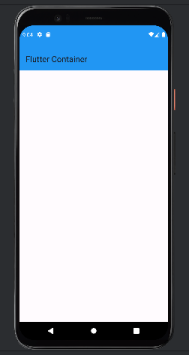
banner add the following

command in Materials App

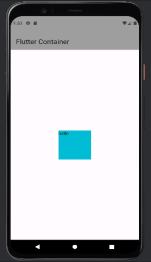
class :

debugShowCheckedModeB

anner: false,



DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATION DEVELOPMENT (SE-487)

import 'package:flutter/material.dart'; 

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title:'flutter demo',

debugShowCheckedModeBanner: false,

theme:ThemeData(

primarySwatch:Colors.*lightBlue*,

),

home:const MyHomePage(),

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({Key?key}):super(key:key);

@override

State<MyHomePage> createState() => \_MyHomePageState(); }

class \_MyHomePageState extends State<MyHomePage>

{

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

backgroundColor: Colors.*grey*,

title: Text('Flutter Container'),

),

body: Center(

child: Container(

width: 100,

height: 100,

color: Colors.*cyan*,

child: Text('hello'),

),

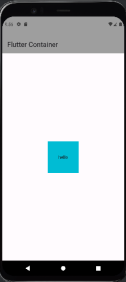
),

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATION DEVELOPMENT (SE-487)

);

}

}

import 'package:flutter/material.dart'; 

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title:'flutter demo',

debugShowCheckedModeBanner: false,

theme:ThemeData(

primarySwatch:Colors.*lightBlue*,

),

home:const MyHomePage(),

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({Key?key}):super(key:key);

@override

State<MyHomePage> createState() => \_MyHomePageState(); }

class \_MyHomePageState extends State<MyHomePage>

{

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

backgroundColor: Colors.*grey*,

title: Text('Flutter Container'),

),

body: Center(

child: Container(

width: 100,

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATION DEVELOPMENT (SE-487)

height: 100,

color: Colors.*cyan*,

child: Center(child: Text('hello')),

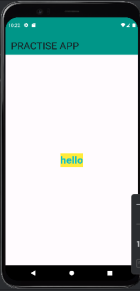
),

),

);

}

if you do not define the height and width of the container it will either take the entire screen (if no child exists) or it will take the size of its child widget.

import 'package:flutter/material.dart'; 

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title:'flutter demo',

debugShowCheckedModeBanner: false,

theme:ThemeData(

primarySwatch:Colors.*lightBlue*,

),

home:const MyHomePage(),

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({Key?key}):super(key:key);

@override

State<MyHomePage> createState() => \_MyHomePageState(); }

class \_MyHomePageState extends State<MyHomePage>

{

@override

Widget build(BuildContext context) {

return Scaffold(

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATION DEVELOPMENT (SE-487)

appBar: AppBar(

backgroundColor: Colors.*teal*,

title: Text('PRACTISE APP',style: TextStyle(fontSize: 30),),

),

body:

Center(child: Text('hello',style: TextStyle(fontSize: 30,fontWeight:

FontWeight.*bold*,color:Colors.*cyan*,backgroundColor:Colors.*yellow* ),)),

);

}

}

**Text Widget**

The Text widget in Flutter is used to display a piece of text within the user interface. It's one of the most commonly used widgets for showing static text in Flutter applications. Here's an overview of the Text widget:

**Properties of the Text Widget:**

data (String): The text to display.

style (TextStyle): Defines the style of the text, including font family, font size, font weight, color, etc.

textAlign (TextAlign): Specifies how the text should be aligned within its container. textDirection (TextDirection): Determines the direction in which the text flows, such as left-to right or right-to-left.

softWrap (bool): Indicates whether the text should wrap onto the next line if it exceeds the width of its container.

overflow (TextOverflow): Defines how the text should behave if it overflows its container, such as clipping, ellipsis, or fading.

maxLines (int): Specifies the maximum number of lines that the text should occupy before truncating.

import 'package:flutter/material.dart';

void main() {

runApp(const MyApp());

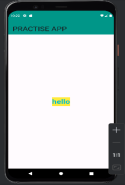
}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATION DEVELOPMENT (SE-487)

Widget build(BuildContext context) { 

return MaterialApp(

title:'flutter demo',

debugShowCheckedModeBanner: false,

theme:ThemeData(

primarySwatch:Colors.*lightBlue*,

),

home:const MyHomePage(),

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({Key?key}):super(key:key);

@override

State<MyHomePage> createState() => \_MyHomePageState(); }

class \_MyHomePageState extends State<MyHomePage>

{

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

backgroundColor: Colors.*teal*,

title: Text('PRACTISE APP',style: TextStyle(fontSize: 30),),

),

body:

Center(child: Text('hello',style: TextStyle(fontSize: 30,fontWeight:

FontWeight.*bold*,color:Colors.*cyan*,backgroundColor:Colors.*yellow* ),)),

);

}

}

import 'package:flutter/material.dart';

void main() {

runApp(const MyApp());

}

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATION DEVELOPMENT (SE-487)

class MyApp extends StatelessWidget { 

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title:'flutter demo',

debugShowCheckedModeBanner: false,

theme:ThemeData(

primarySwatch:Colors.*lightBlue*,

),

home:const MyHomePage(),

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({Key?key}):super(key:key);

@override

State<MyHomePage> createState() => \_MyHomePageState(); }

class \_MyHomePageState extends State<MyHomePage>

{

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

backgroundColor: Colors.*teal*,

title: Text('PRACTISE APP',style: TextStyle(fontSize: 30),),

),

body:

Container(

color: Colors.*grey*,

height: 400,

width: 400,

child: Text( 'This is a long text example that demonstrates the usage of all the properties of the Text widget in Flutter.',

style: TextStyle(fontSize: 30,fontWeight: FontWeight.*bold*,color:Colors.*cyan*, backgroundColor:Colors.*yellow* ),

//textAlign:TextAlign.left, //textDirection: TextDirection.rtl,

softWrap: true,

overflow: TextOverflow.ellipsis,

maxLines: 2,

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATION DEVELOPMENT (SE-487)

),

),

);

}

}

**Center Widget :**

The Center widget in Flutter is used to center its child widget both horizontally and vertically within itself. It's commonly used to position a single child widget at the center of its parent widget. The Center widget in Flutter does not have any properties of its own, as it only serves to center its child widget within itself. However, it does inherit properties from its parent widget. Here are some common properties that affect the behavior of the Center widget:

child (Widget): The child widget that is centered within the Center widget.

key (Key): An optional key that can be used to identify the Center widget.

widthFactor (double): The width factor applied to the width of the child widget. If non-null, the child's width will be this factor times the Center widget's width.

heightFactor (double): The height factor applied to the height of the child widget. If non-null, the child's height will be this factor times the Center widget's height.

These properties can be used to customize the behavior and appearance of the child widget within the Center widget. However, the main purpose of the Center widget is to position its child widget at the center of the available space both horizontally and vertically.

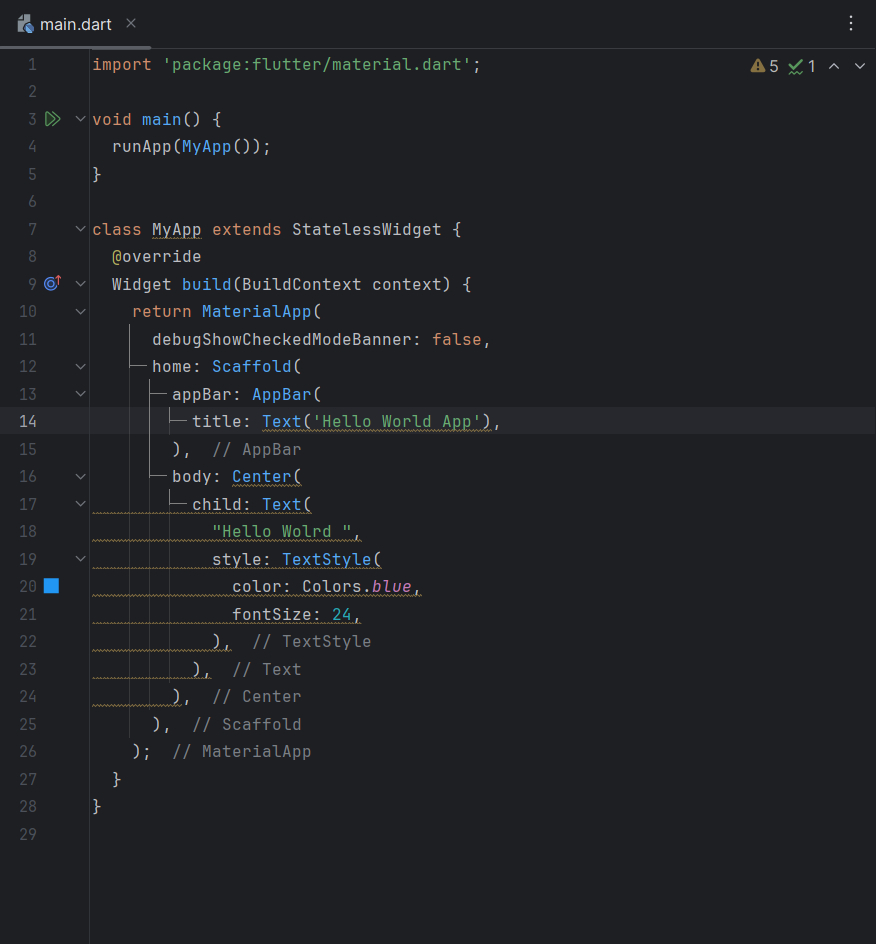
DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATION DEVELOPMENT (SE-487)

**Exercise**

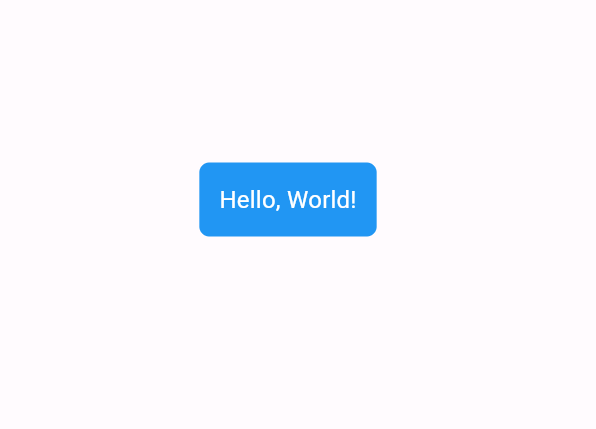
**Question 1:**

Create a Flutter widget that displays a text box in the center of the screen with the text "Hello, World!" in blue color and font size 24.



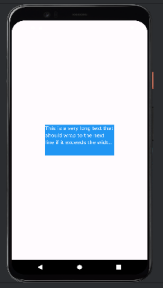
****

**Output**

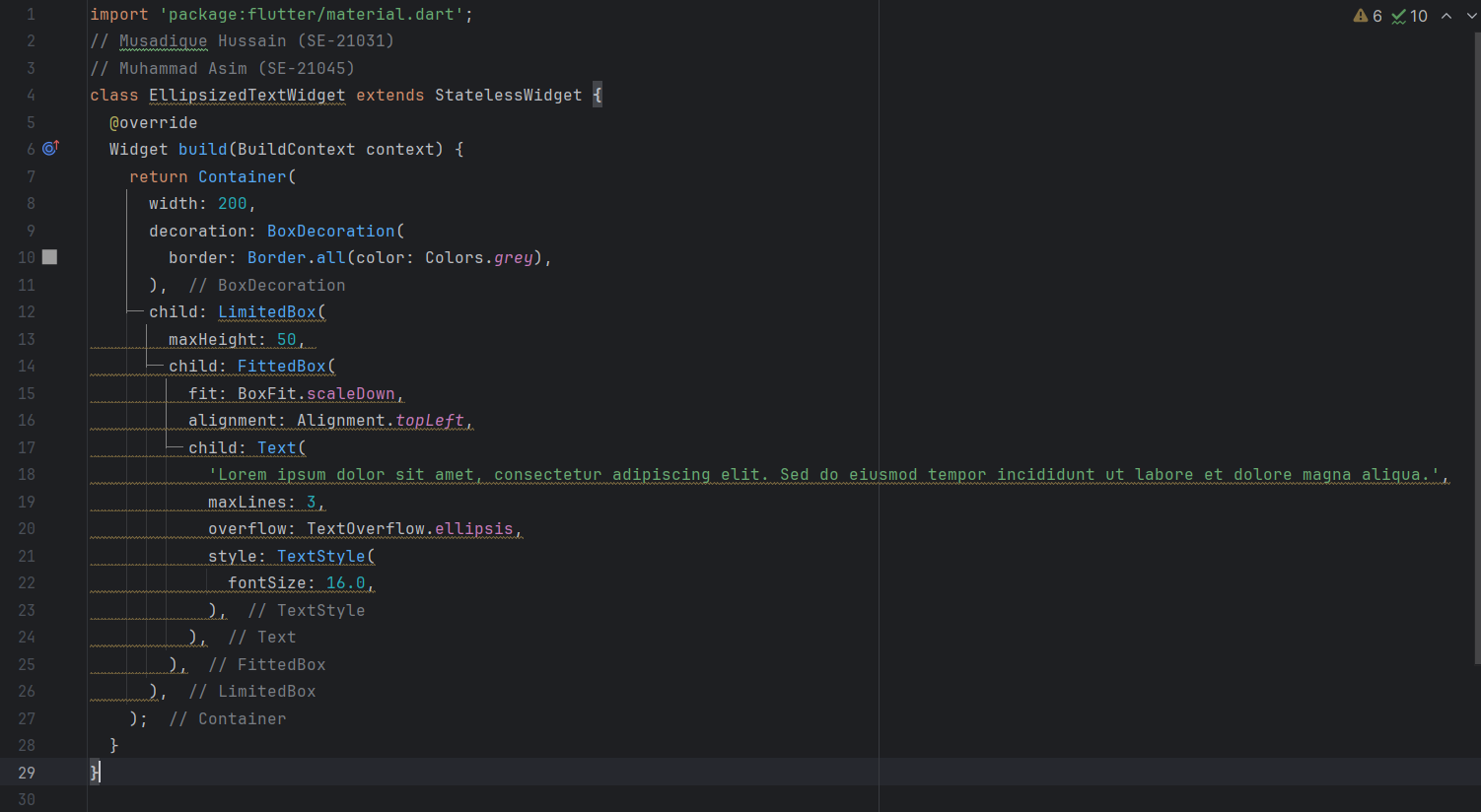


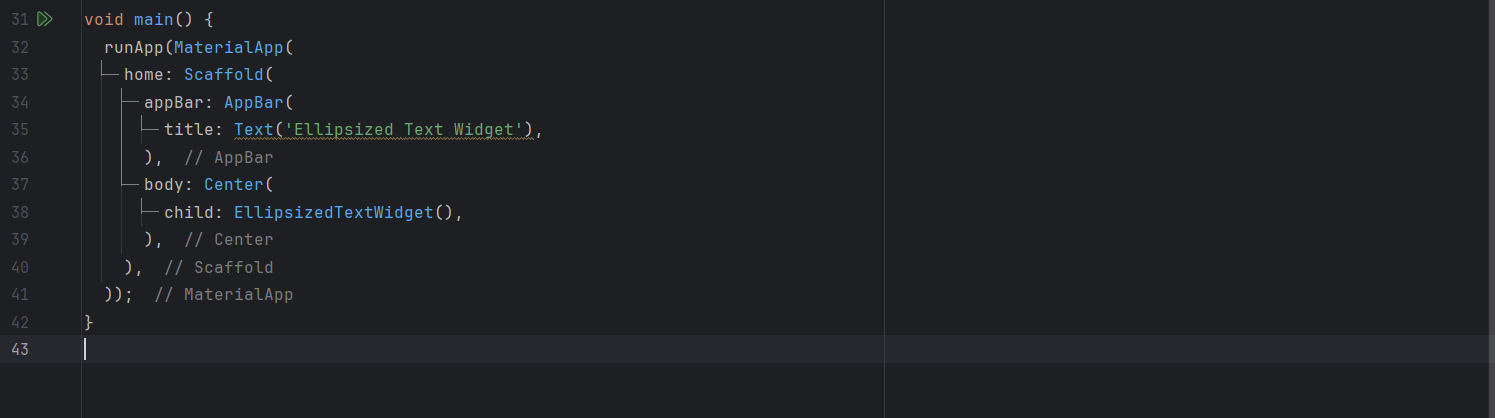
**Question 2:**

Create a Flutter widget that displays a long text inside a container. The text should wrap to the next line if it exceeds the width of the container and should be ellipsized with three dots (...) if it overflows vertically. Limit the text to a maximum of 3 lines.

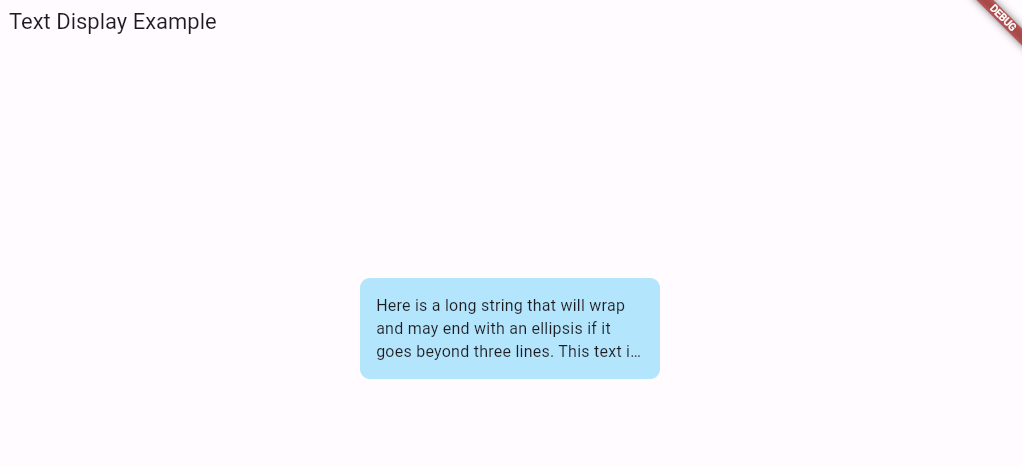


**Code**





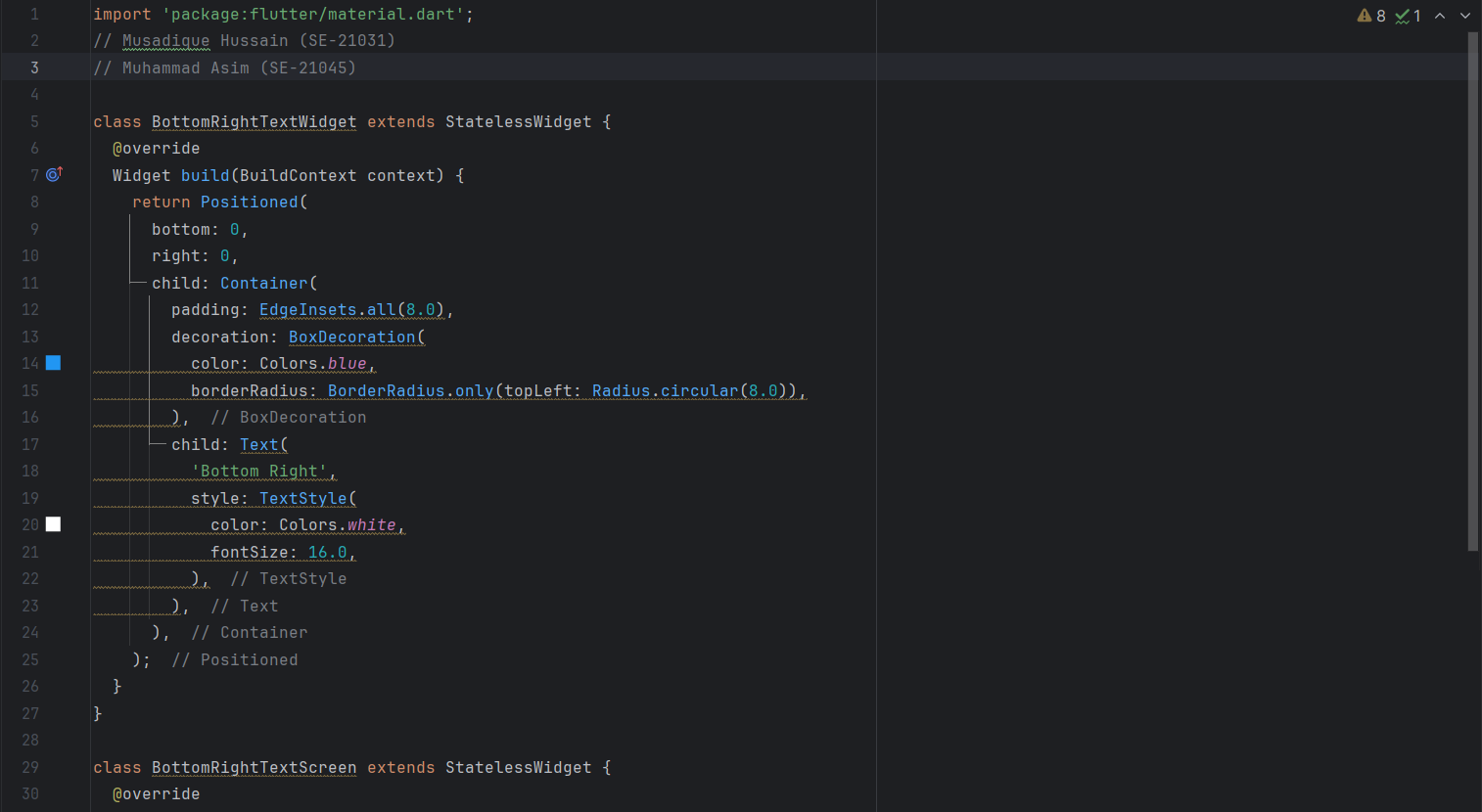
**Output**



Create a Flutter widget that displays a text widget in the bottom right corner of the screen using a container. The text should say "Bottom Right" and should be displayed in white color with a blue background.



Code





Ouput



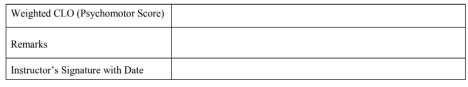
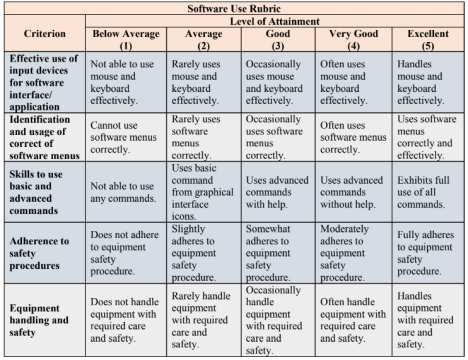
Code

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATION DEVELOPMENT (SE-487)

**NED University of Engineering & Technology** 

**Department of Software Engineering**

**Course Code and Title: MOBILE APPLICATION DEVELOPMENT (SE-487)**

**Laboratory Session No. 6 Date: 30-04-2024**

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATION DEVELOPMENT (SE-487)