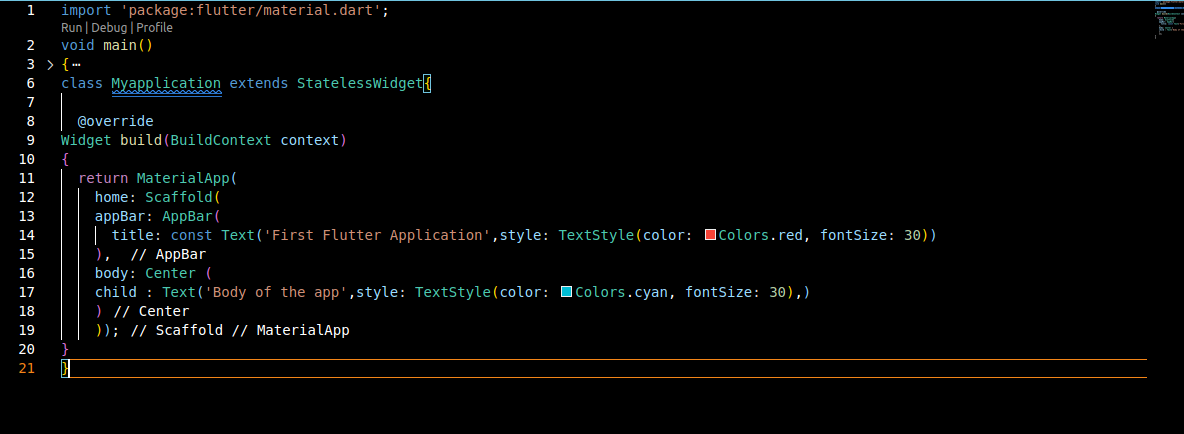
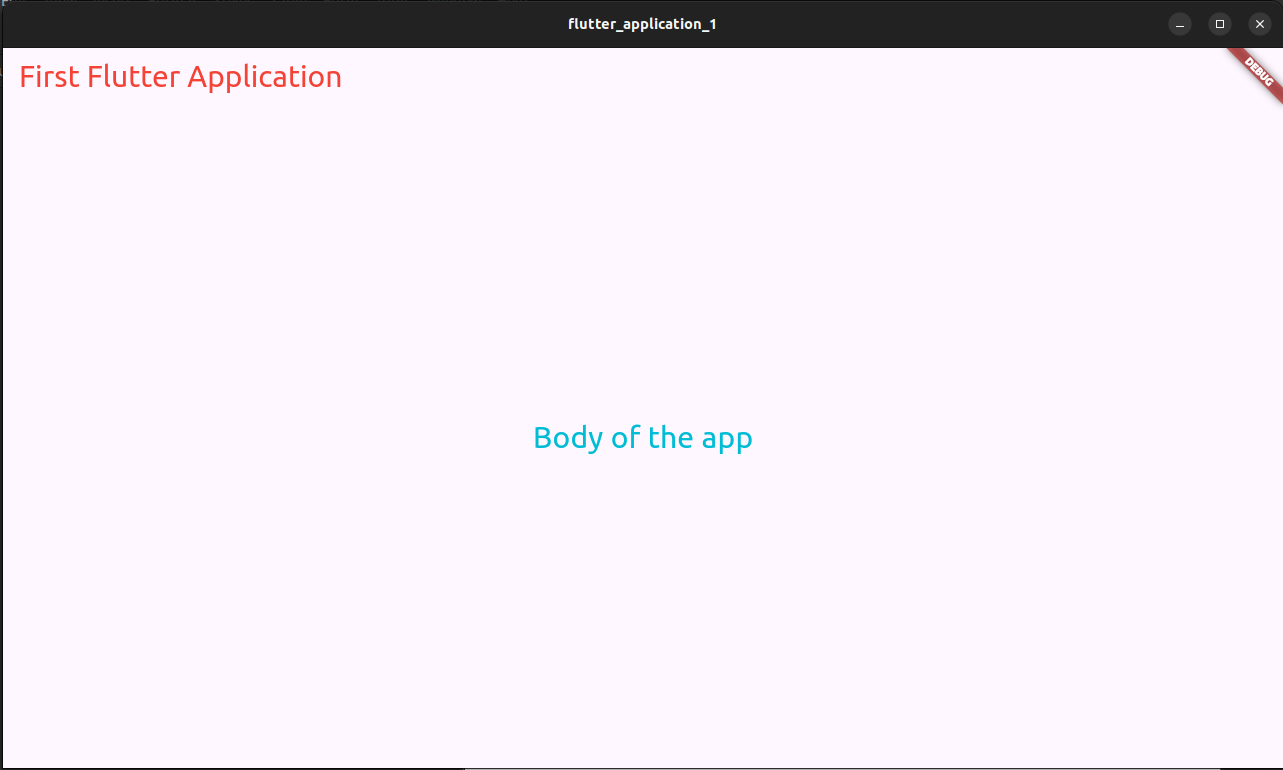
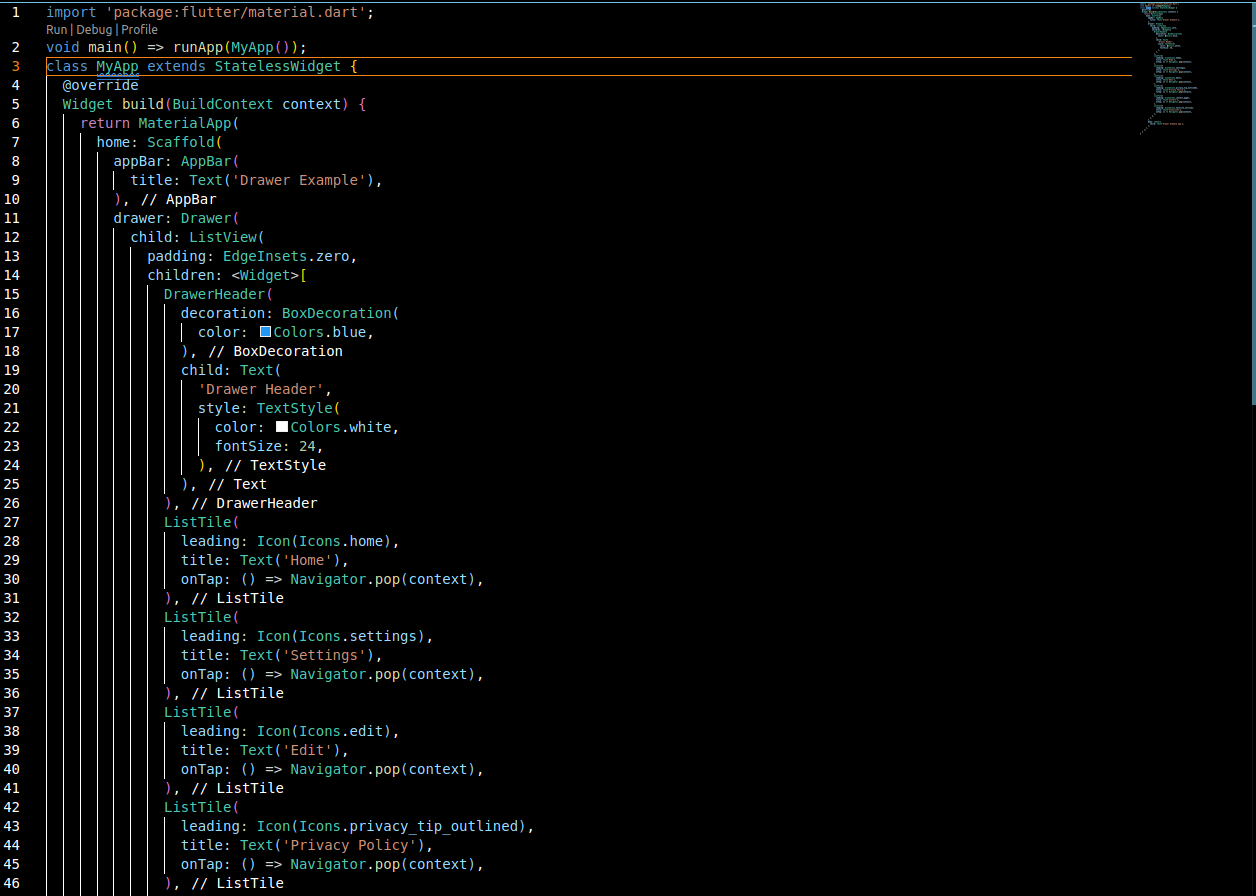
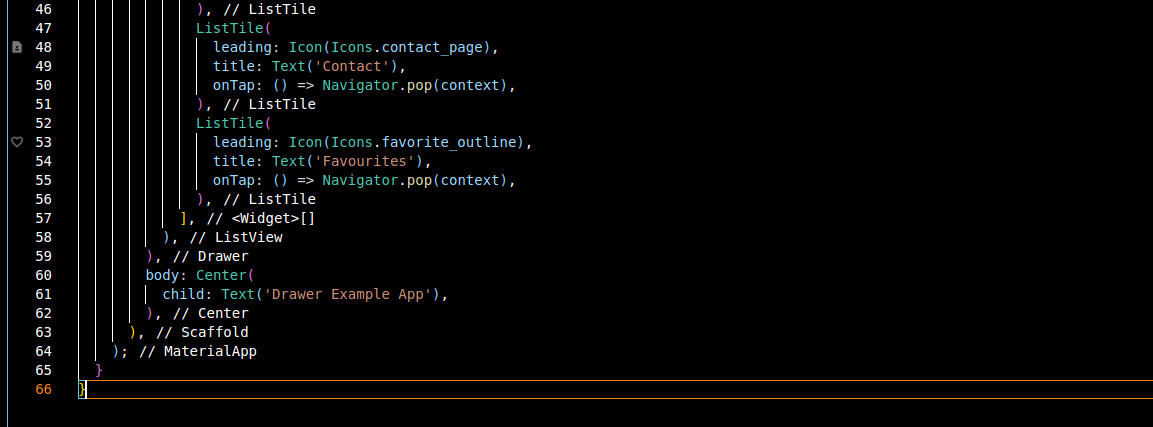
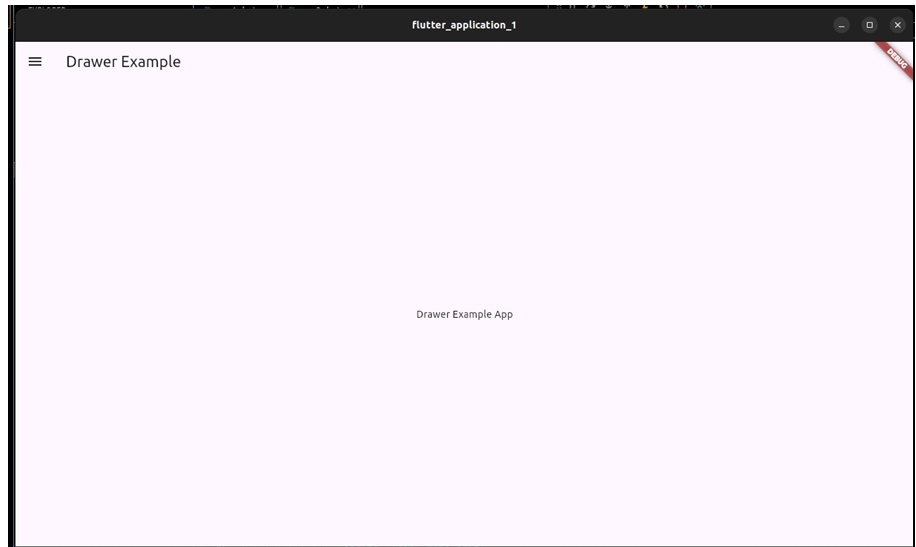
INTERNSHIP DAY-3

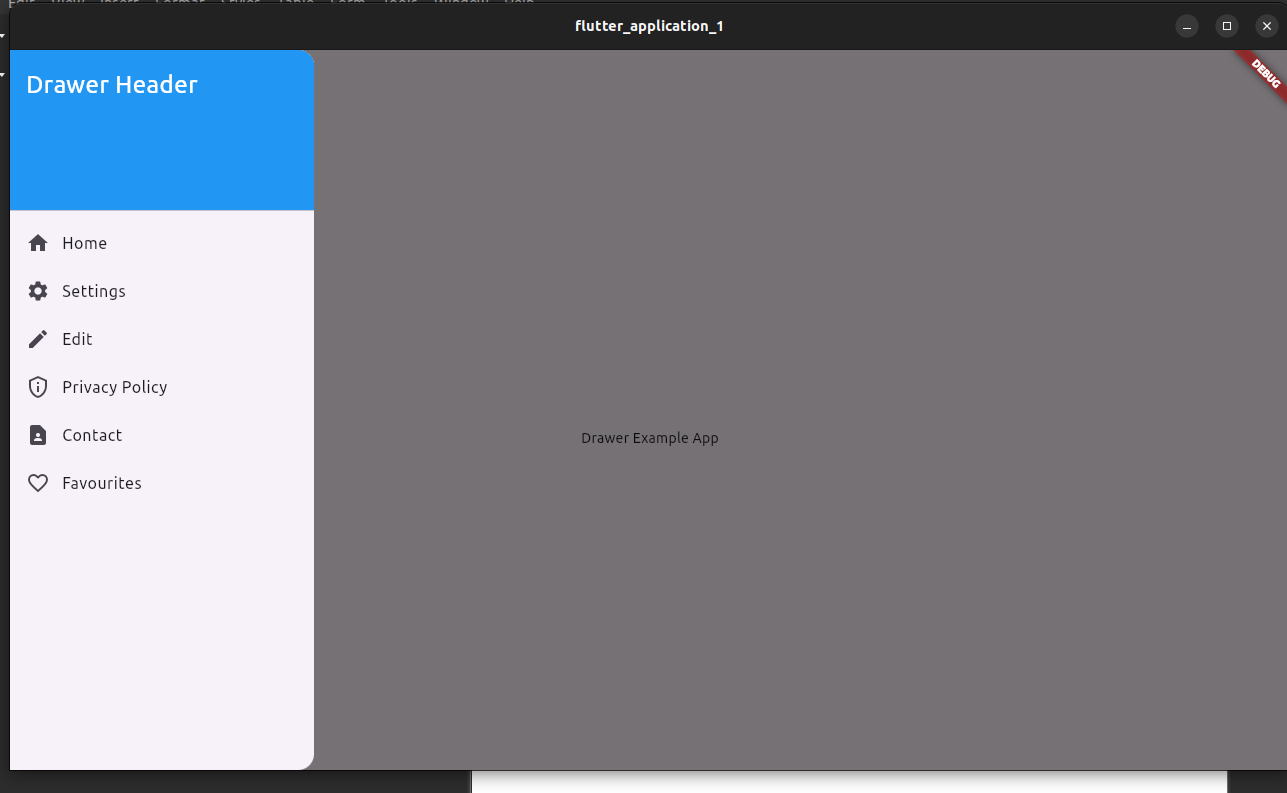
1.



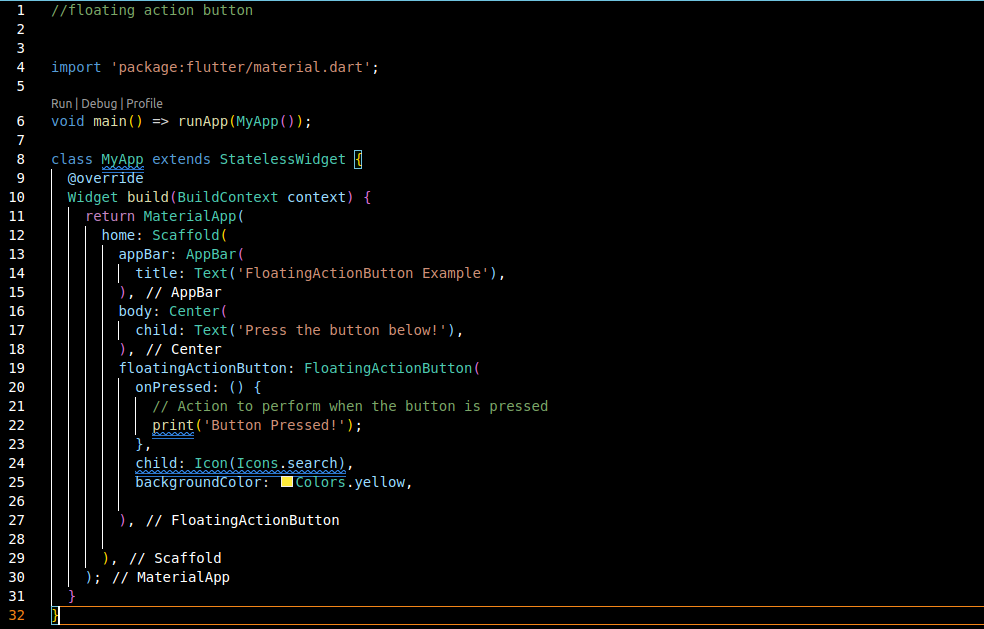
2.

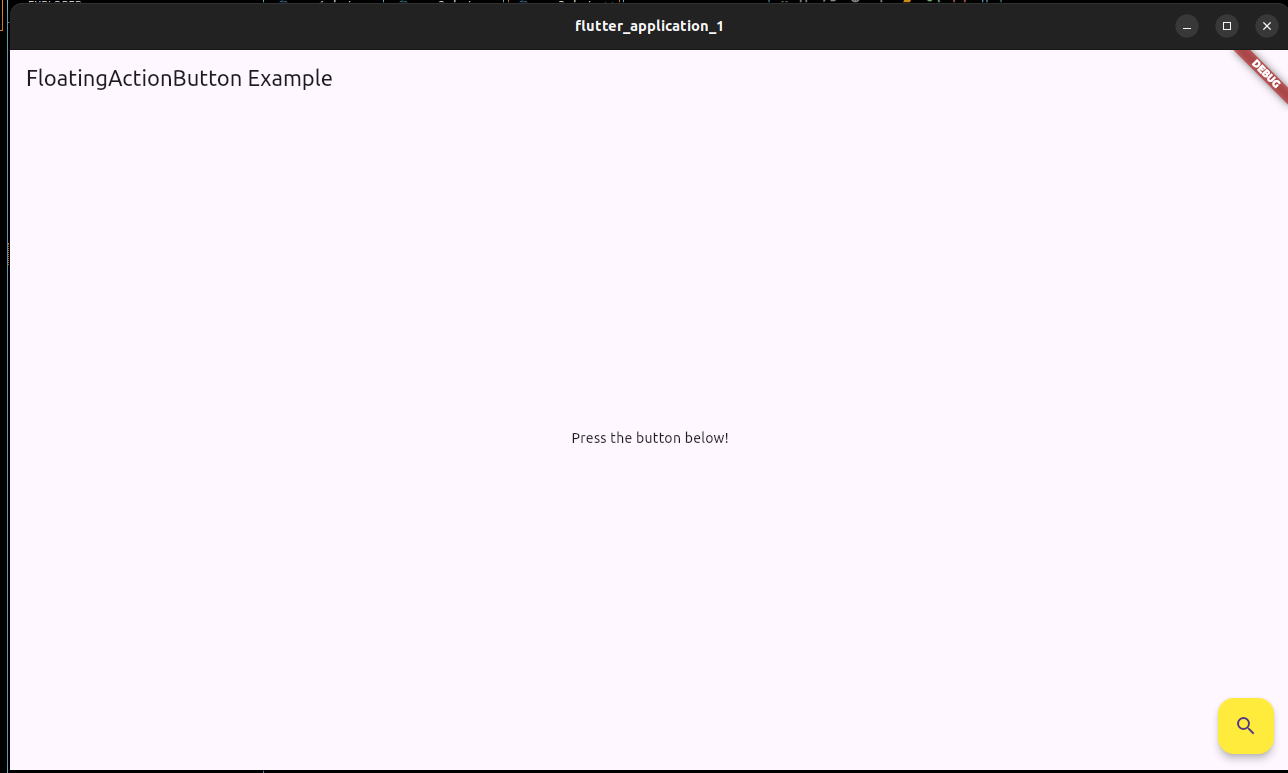


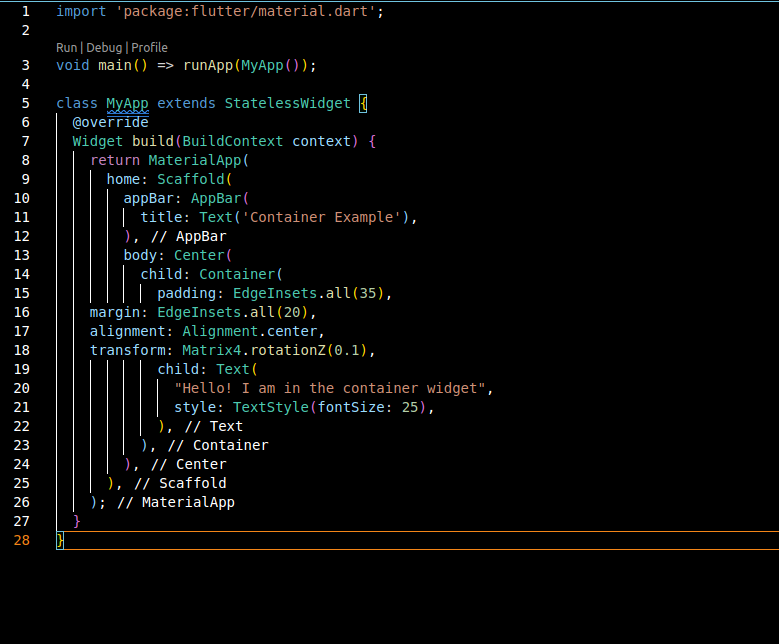


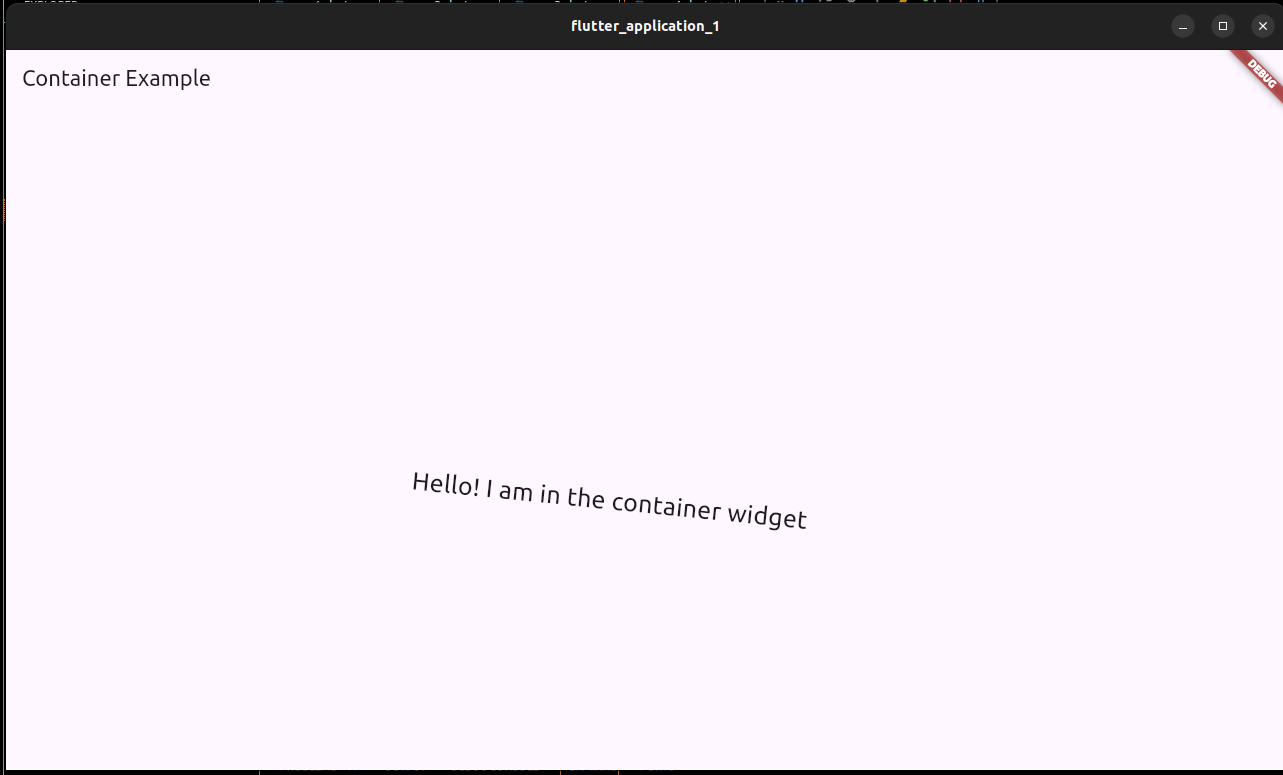


3.

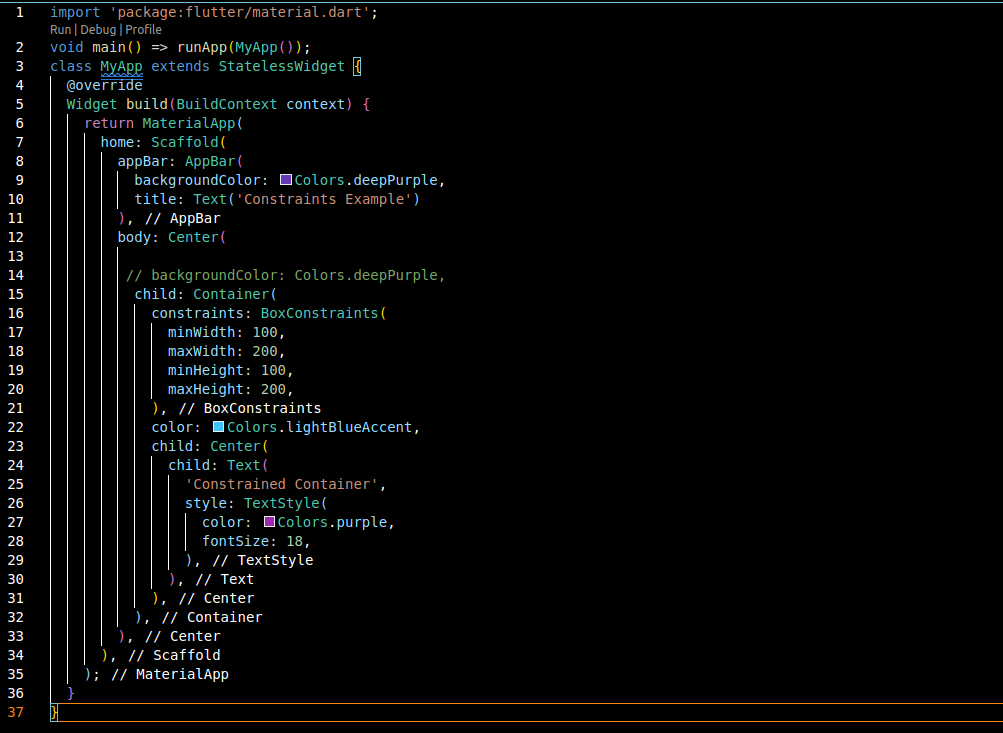


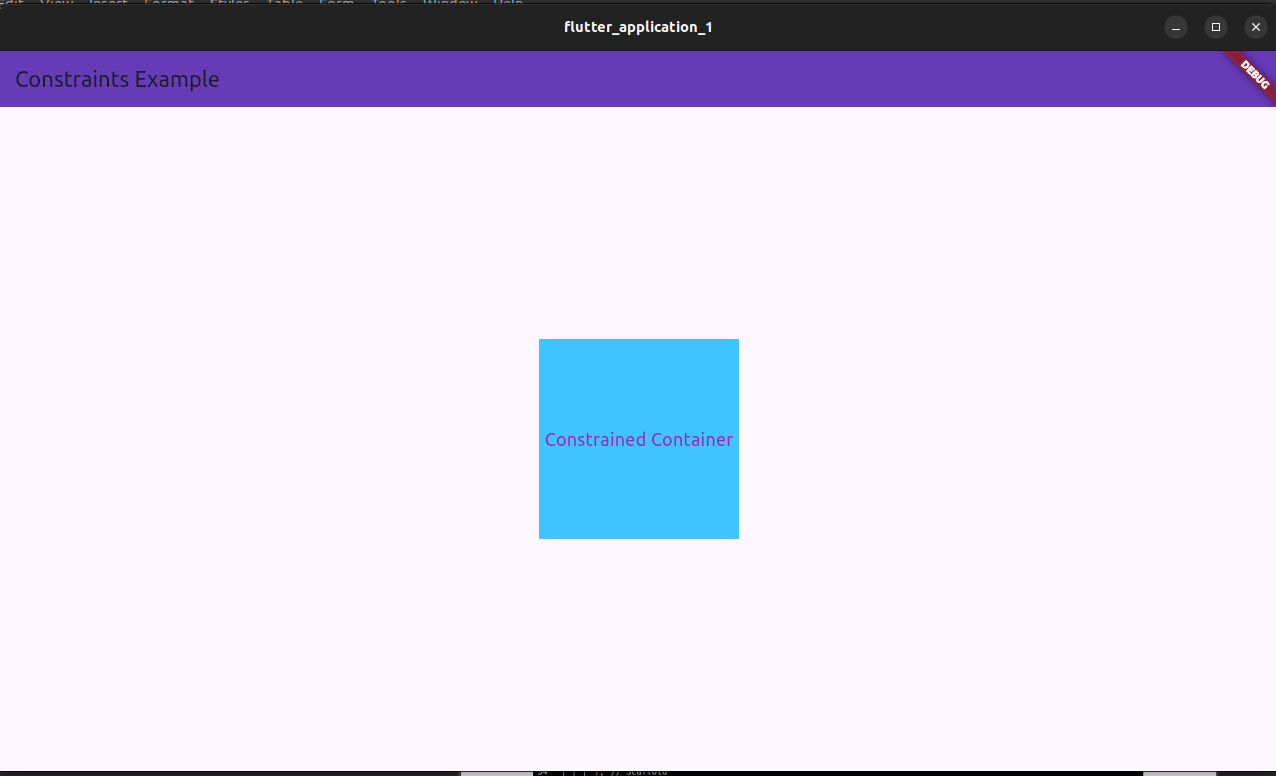


4.

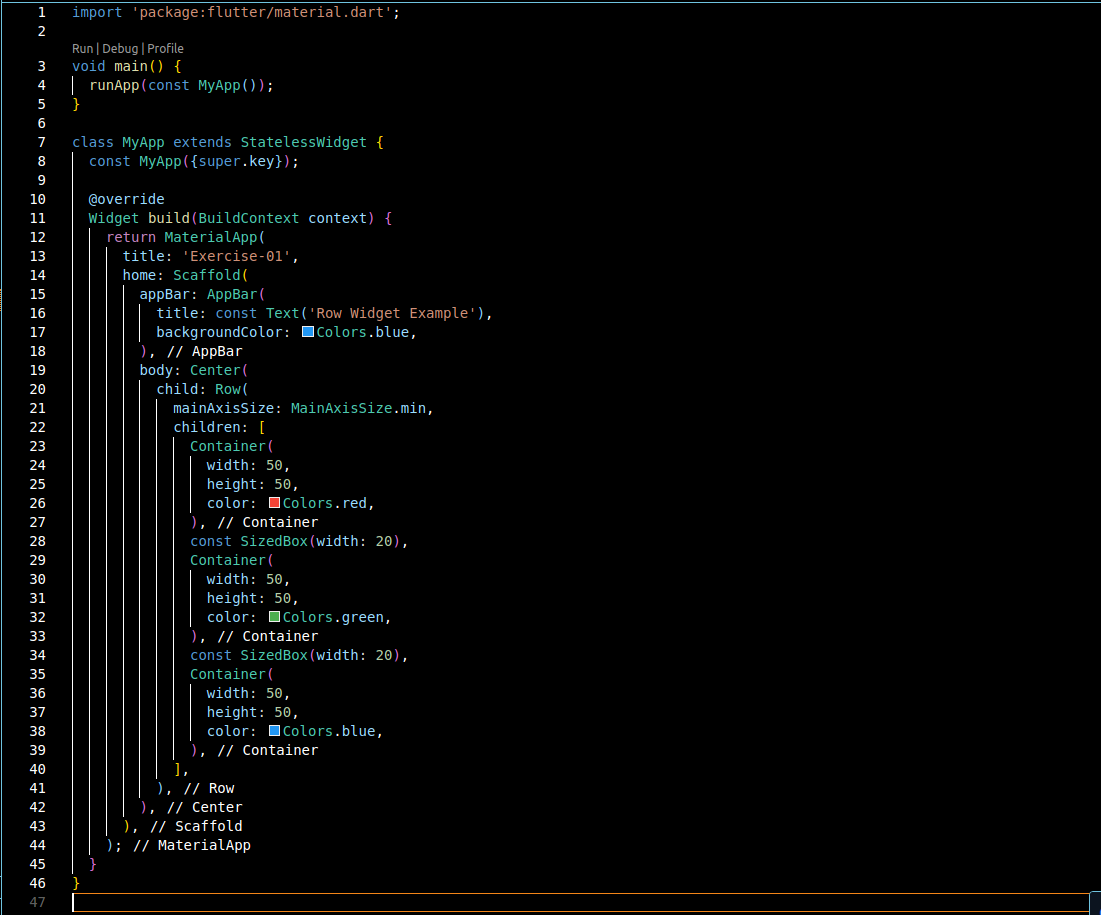


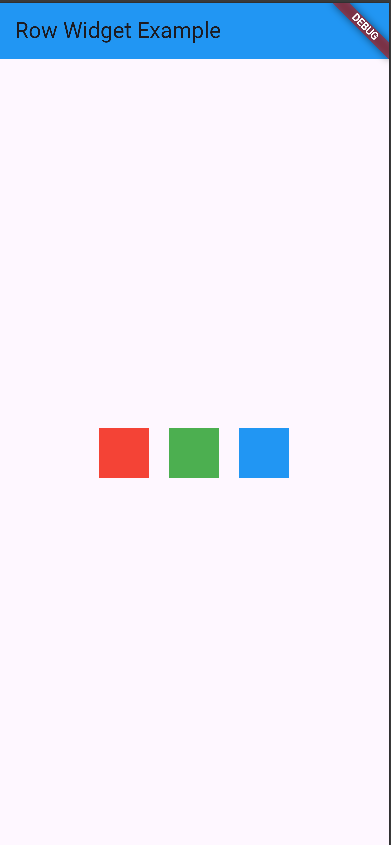
5.

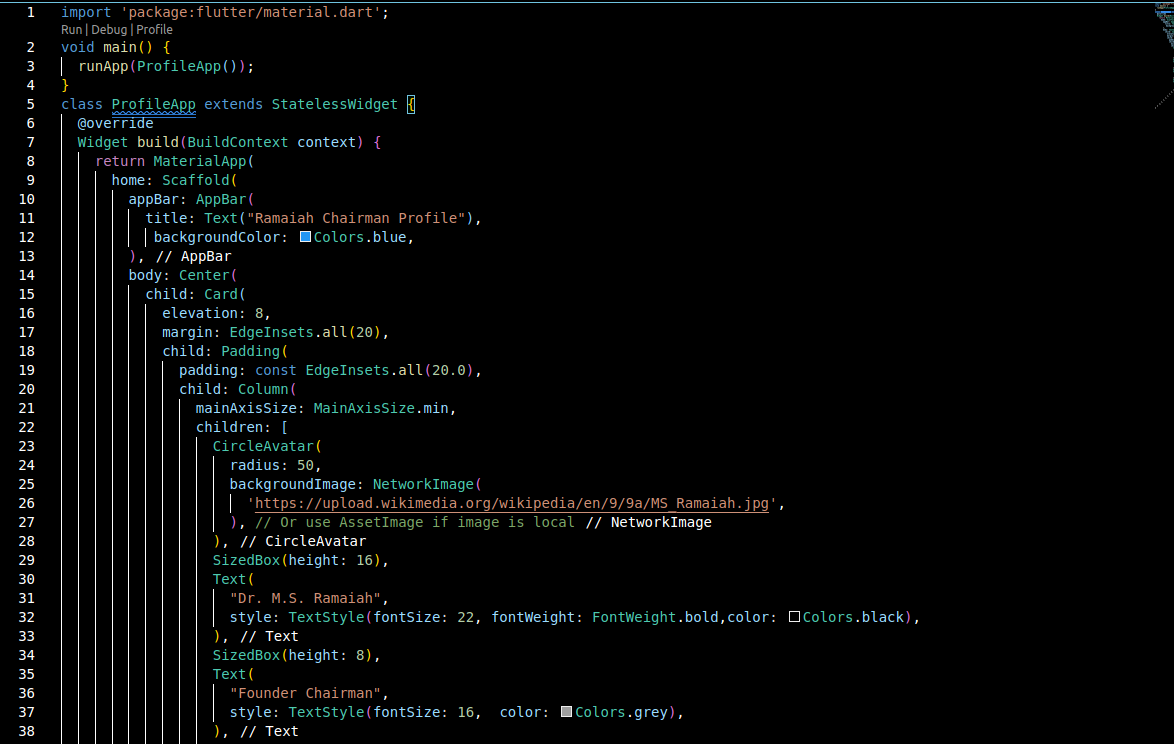


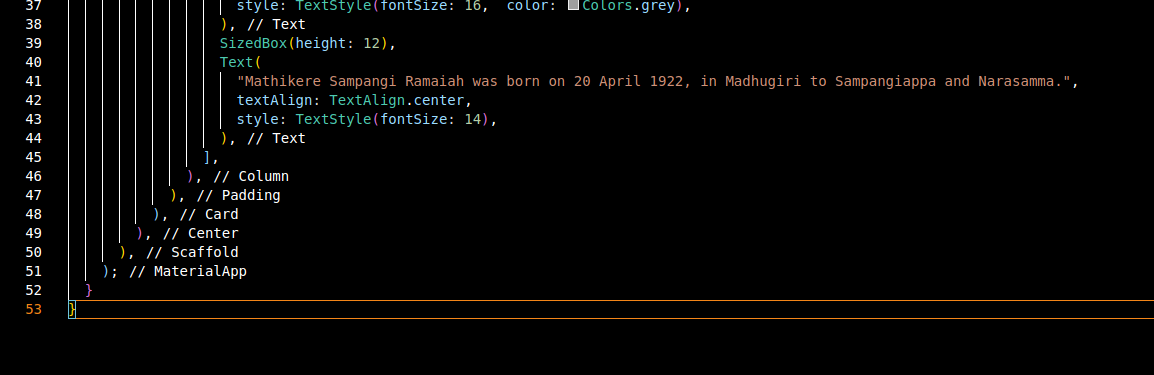


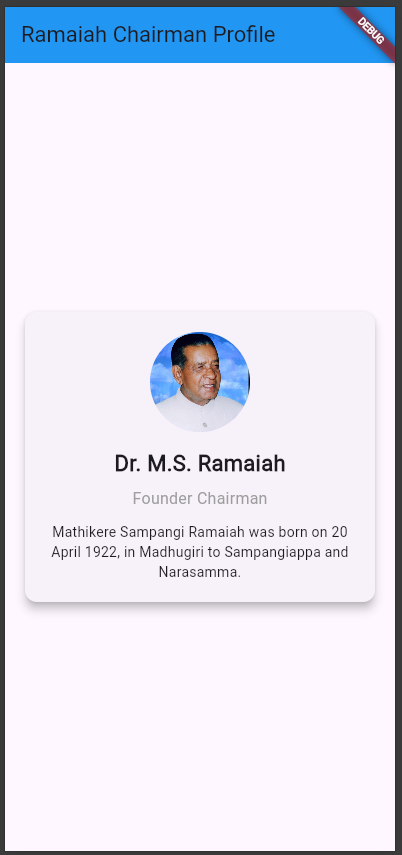
ASSIGNMENT EXERCISES-

1.

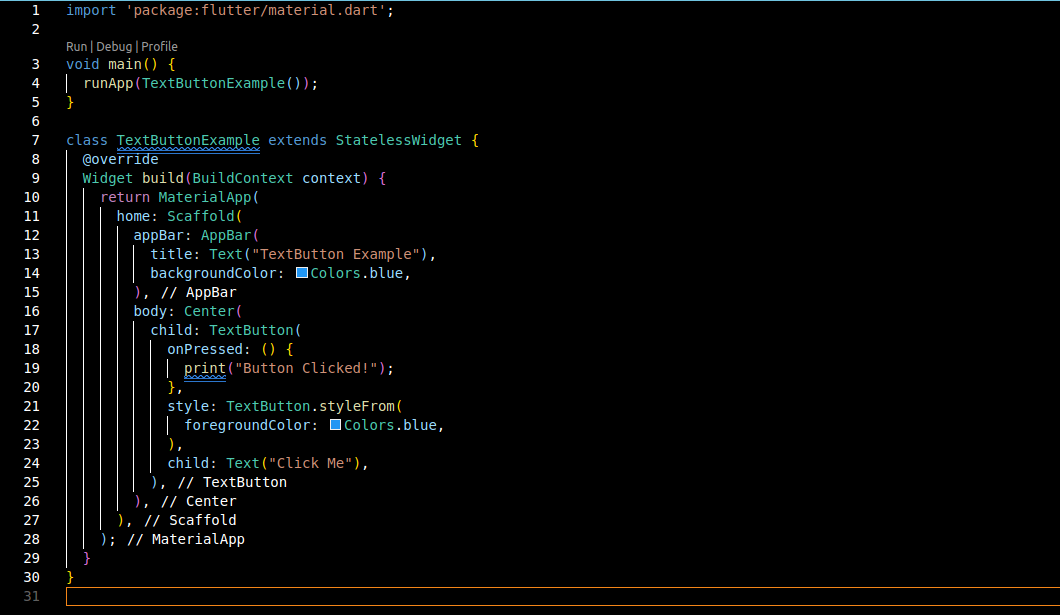


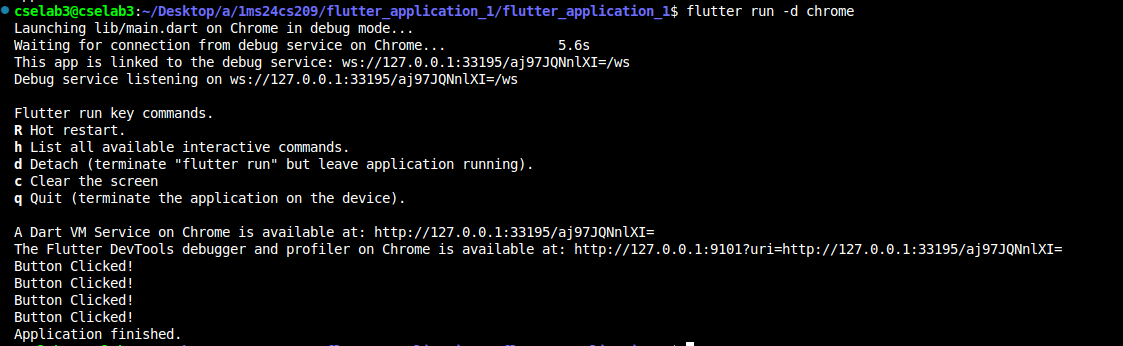
2.

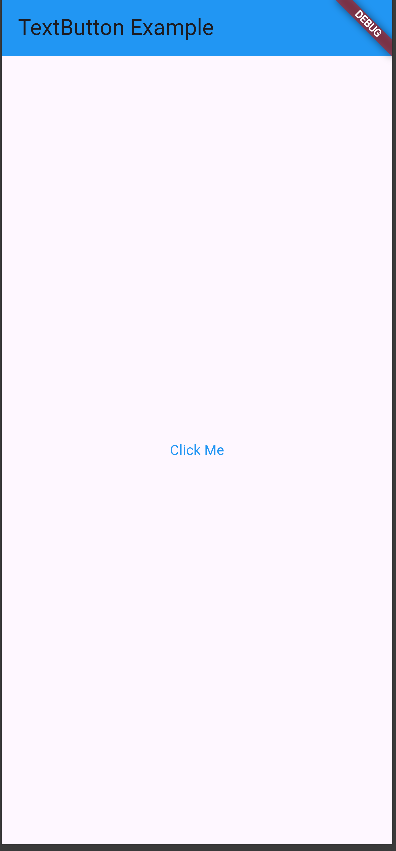




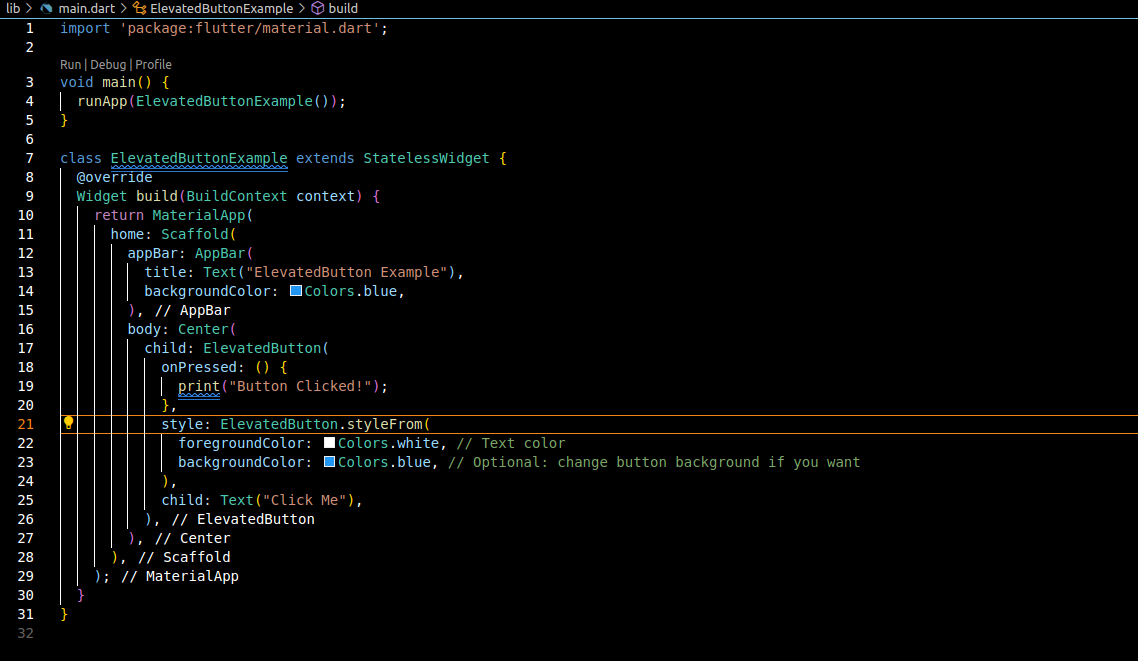
3.

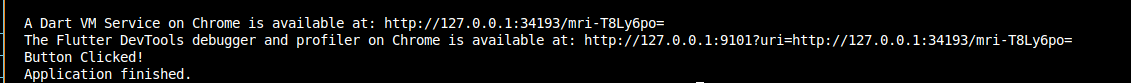


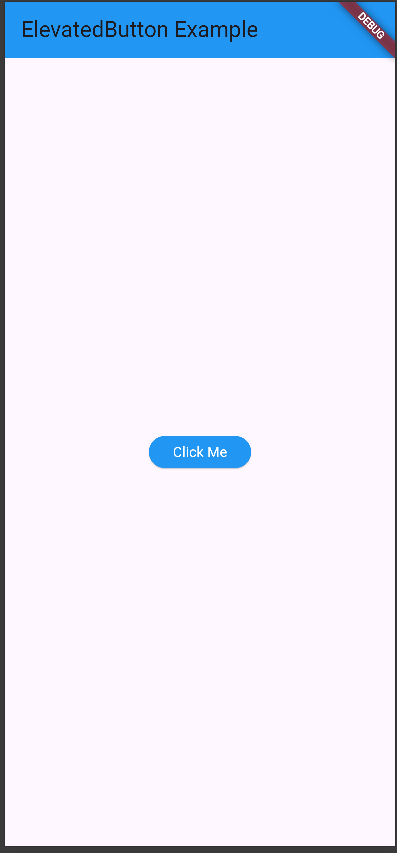


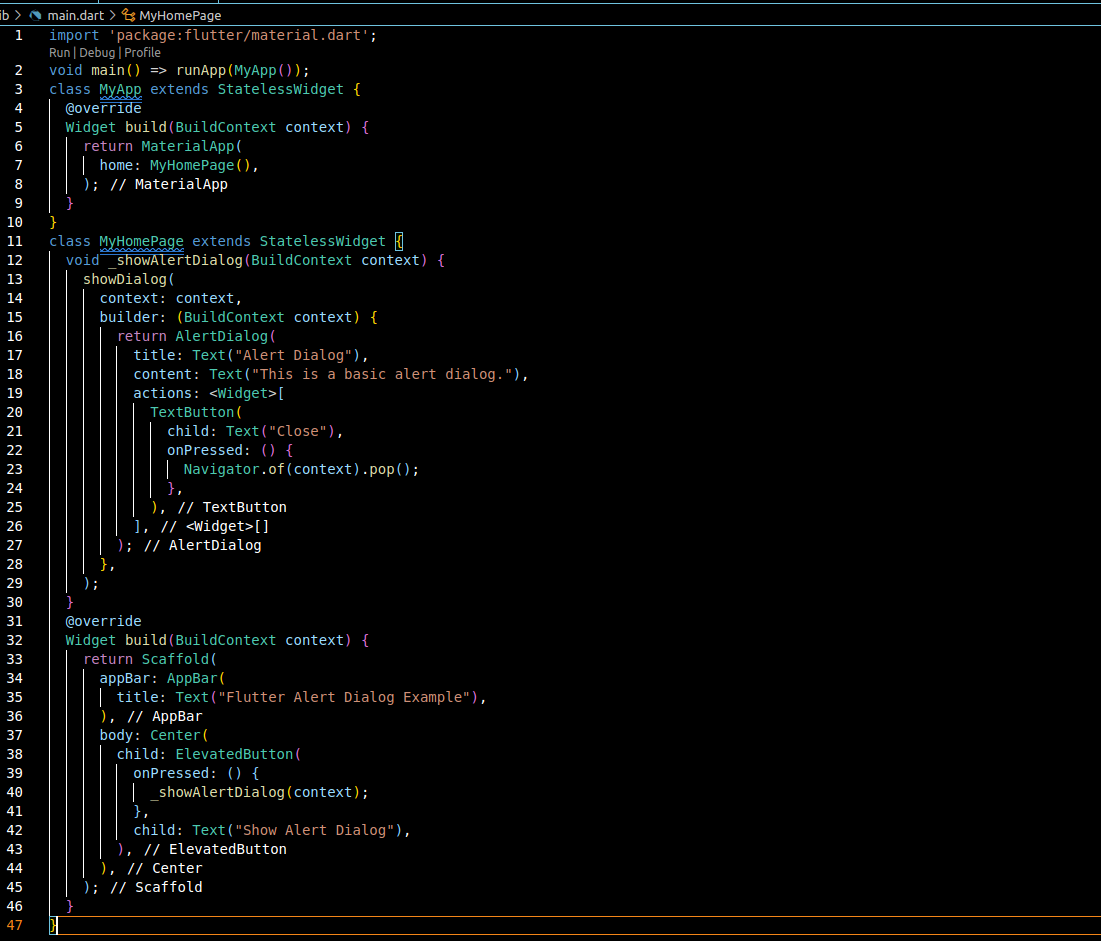


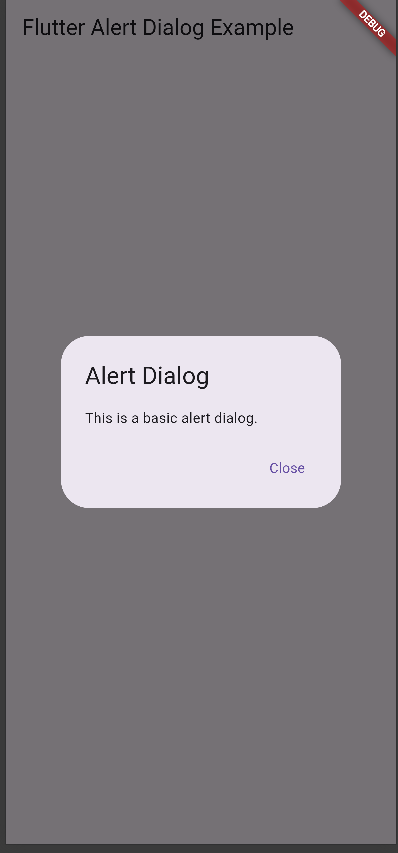
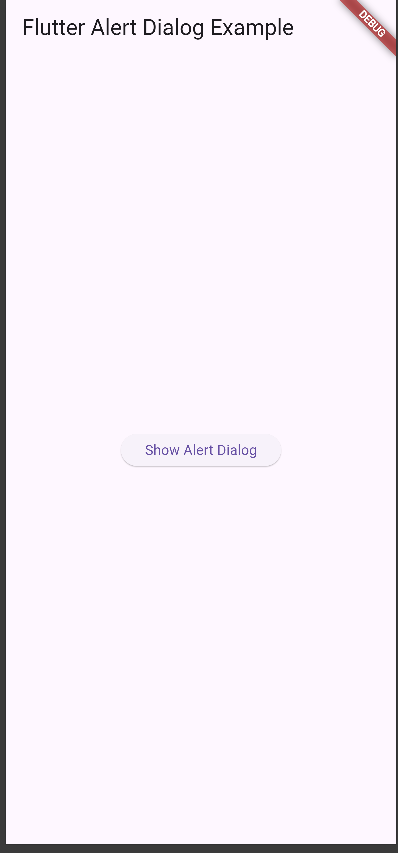
4.

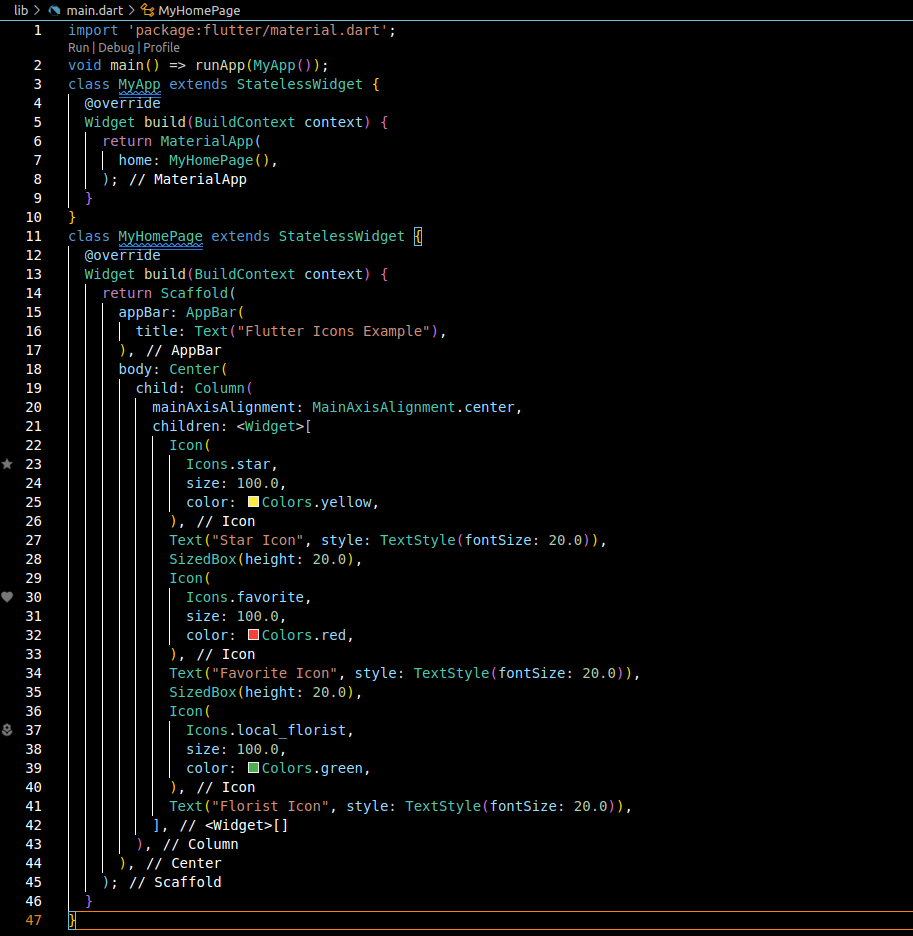


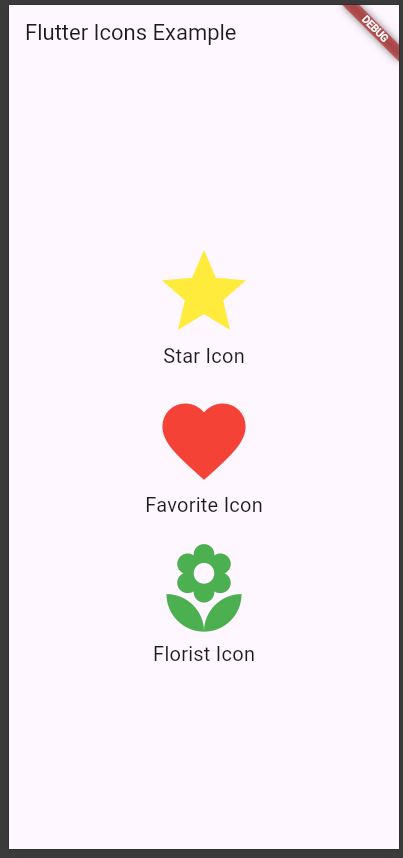


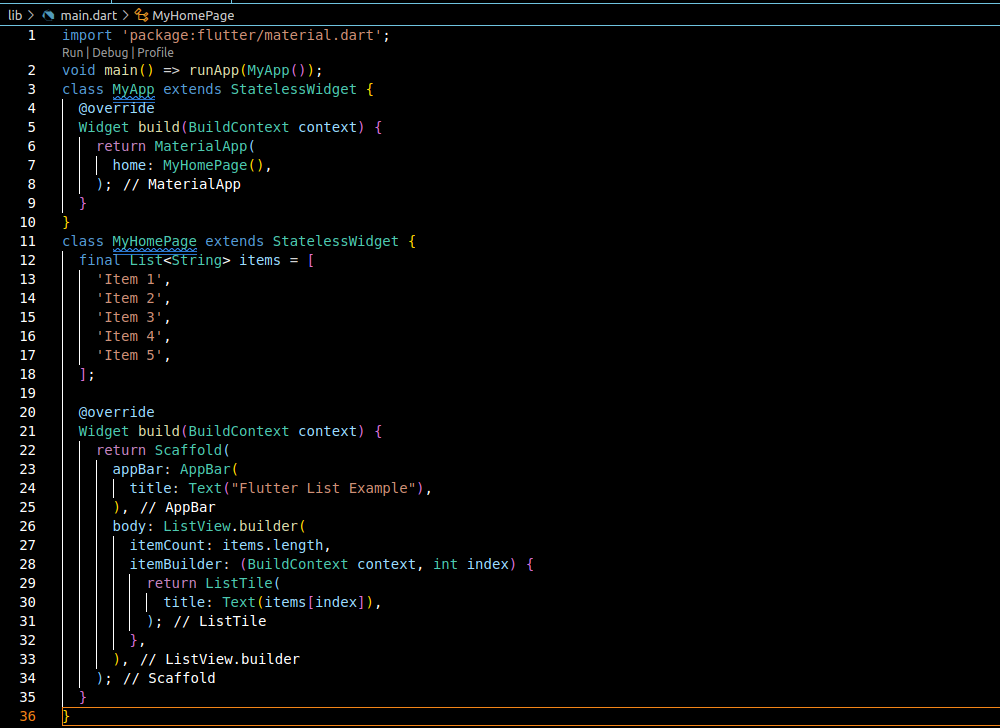


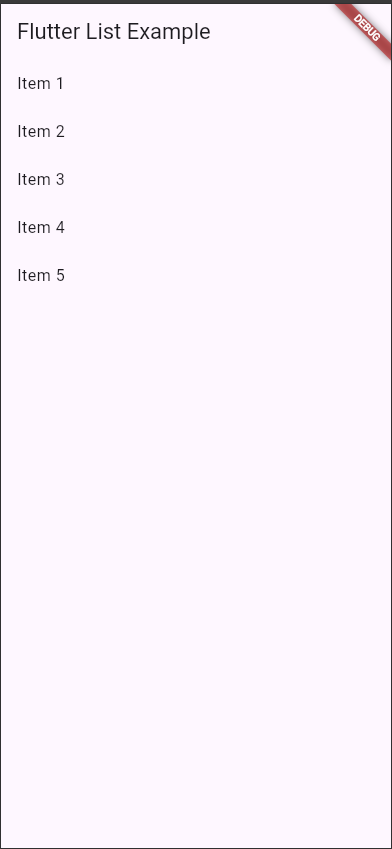
5.

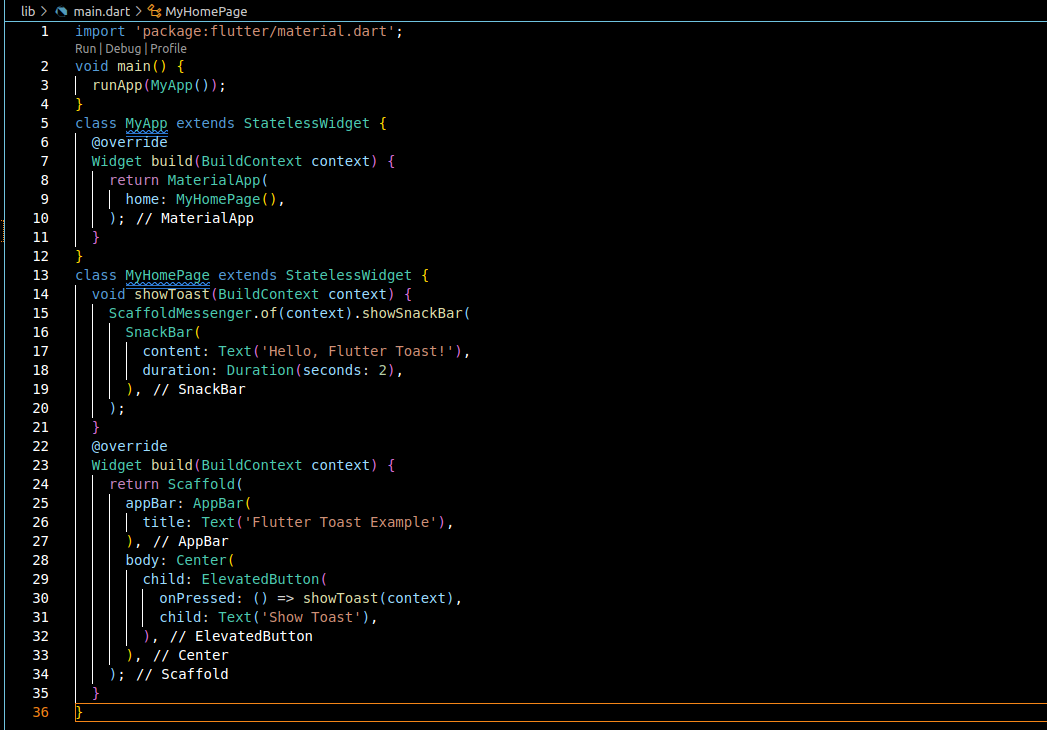


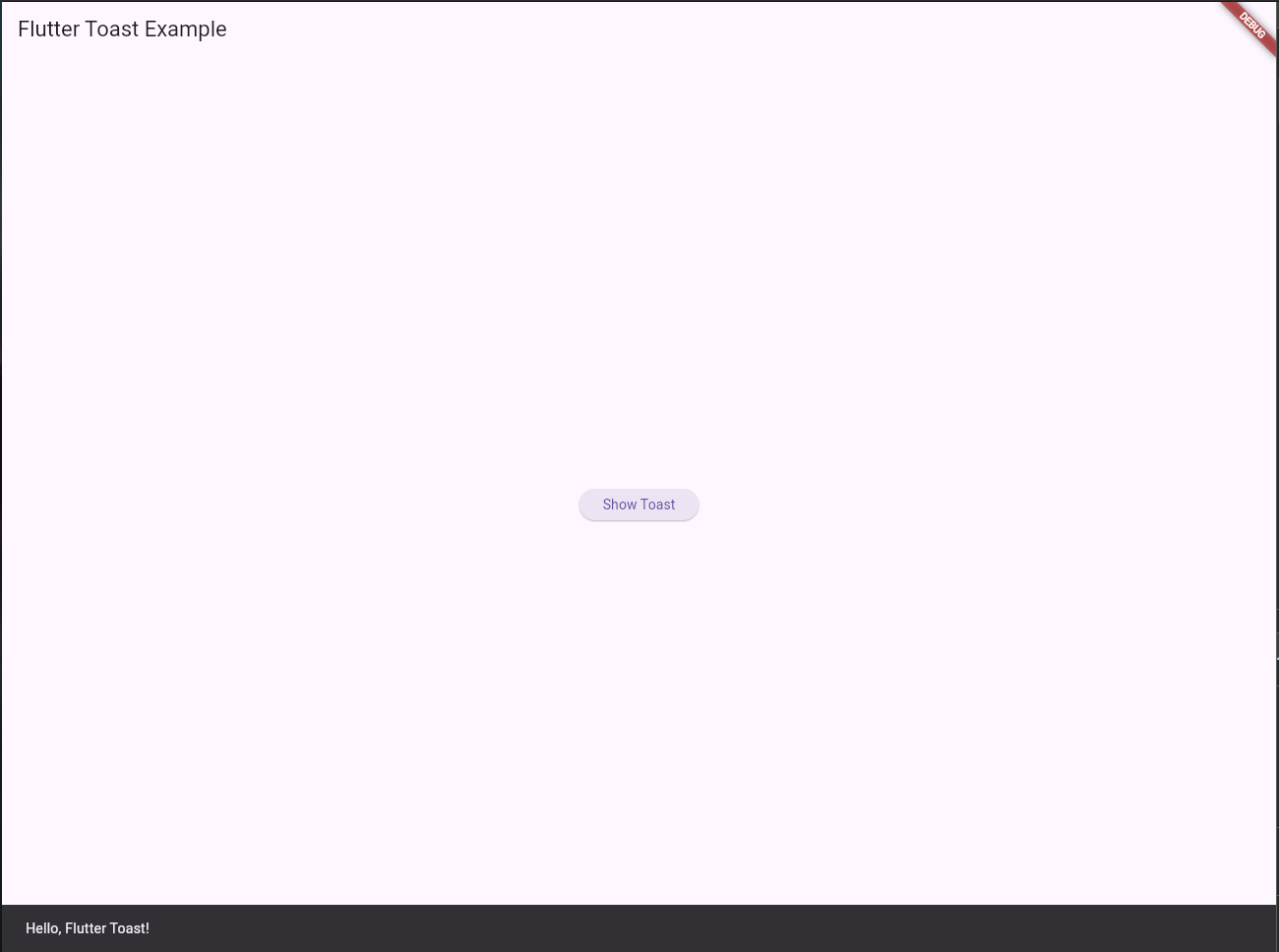
6.



7.



8.



9.

Code:

import 'package:flutter/material.dart';

import 'package:fl\_chart/fl\_chart.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

home: MyHomePage(),

);

}

}

class MyHomePage extends StatelessWidget {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Flutter Pie Chart Demo'),

),

body: Center(

child: PieChart(

PieChartData(

sections: [

PieChartSectionData(

color: Colors.blue,

value: 30,

title: 'Category A',

),

PieChartSectionData(

color: Colors.green,

value: 40,

title: 'Category B',

),

PieChartSectionData(

color: Colors.orange,

value: 20,

title: 'Category C',

),

PieChartSectionData(

color: Colors.red,

value: 10,

title: 'Category D',

),

],

),

),

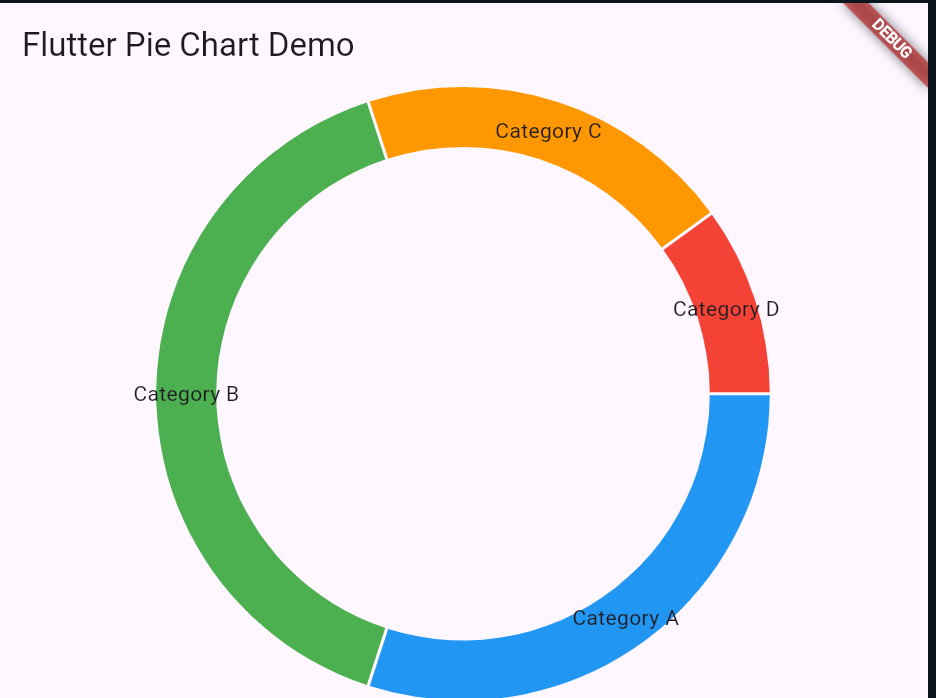
),

);

}

}

Output:



10. Code:

import 'package:flutter/material.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatefulWidget {

const MyApp({super.key});

@override

State<MyApp> createState() => \_MyAppState();

}

class \_MyAppState extends State<MyApp> {

ThemeMode \_themeMode = ThemeMode.light;

void \_toggleTheme() {

setState(() {

\_themeMode =

\_themeMode == ThemeMode.light ? ThemeMode.dark : ThemeMode.light;

});

}

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Sliding Theme Toggle',

theme: ThemeData(

brightness: Brightness.light,

primarySwatch: Colors.blue,

),

darkTheme: ThemeData(

brightness: Brightness.dark,

primarySwatch: Colors.deepPurple,

),

themeMode: \_themeMode,

home: HomeScreen(

onToggleTheme: \_toggleTheme,

isDarkMode: \_themeMode == ThemeMode.dark,

),

);

}

}

class HomeScreen extends StatelessWidget {

final VoidCallback onToggleTheme;

final bool isDarkMode;

const HomeScreen({

super.key,

required this.onToggleTheme,

required this.isDarkMode,

});

@override

Widget build(BuildContext context) {

return Scaffold(

body: Center(

child: GestureDetector(

onTap: onToggleTheme,

child: Container(

width: 120,

height: 50,

padding: const EdgeInsets.all(5),

decoration: BoxDecoration(

color: isDarkMode ? Colors.grey[800] : Colors.grey[300],

borderRadius: BorderRadius.circular(30),

),

child: Stack(

children: [

AnimatedAlign(

duration: const Duration(milliseconds: 250),

curve: Curves.easeInOut,

alignment:

isDarkMode ? Alignment.centerRight : Alignment.centerLeft,

child: Container(

width: 40,

height: 40,

decoration: BoxDecoration(

color: isDarkMode ? Colors.yellow[700] : Colors.black87,

shape: BoxShape.circle,

),

child: Icon(

isDarkMode ? Icons.light\_mode : Icons.dark\_mode,

color: Colors.white,

size: 20,

),

),

),

],

),

),

),

),

);

}

}

Output:

