**Name:** Kaklotar Gaurav Amarshibhai

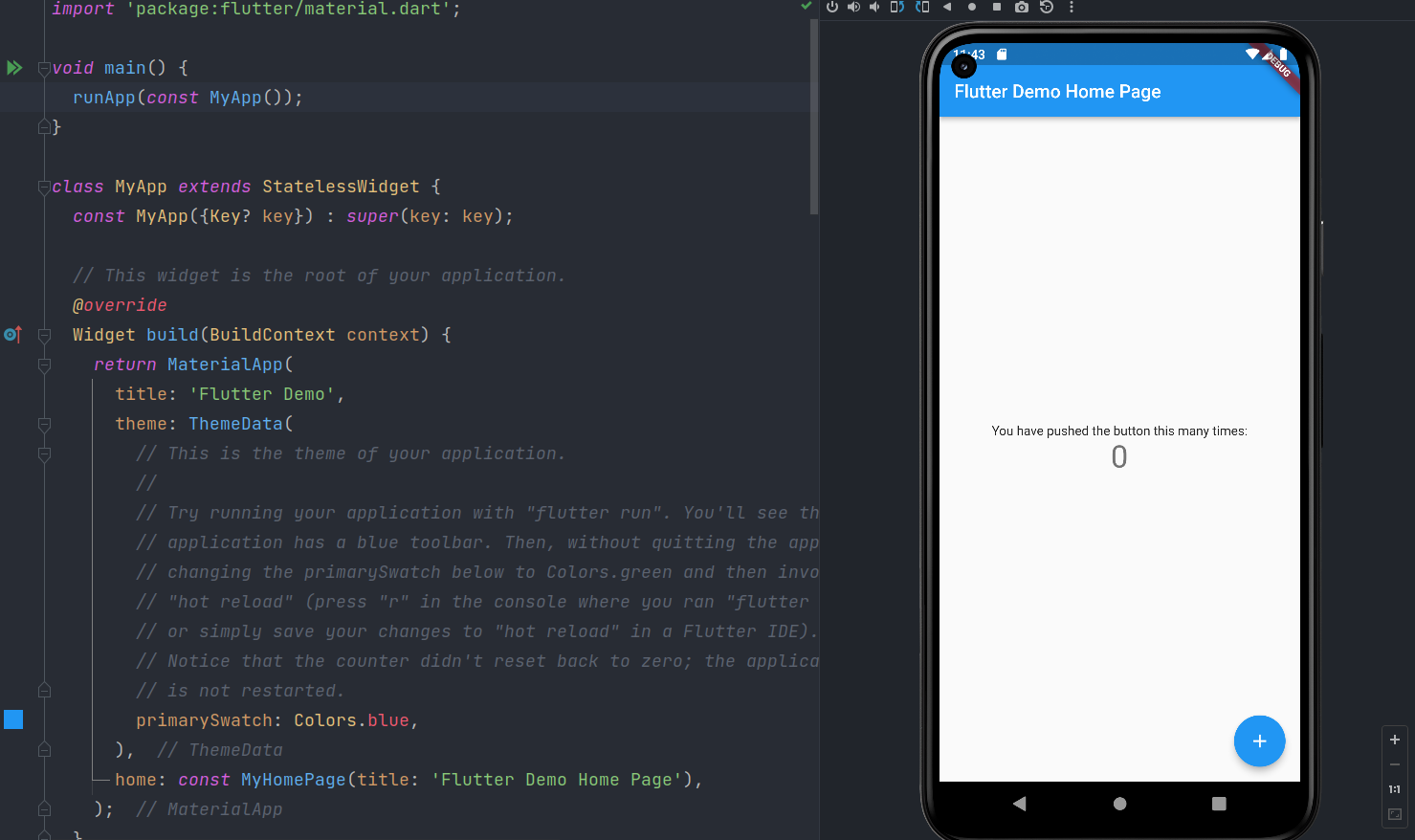
**Subject:** SDP

**Batch:** A3

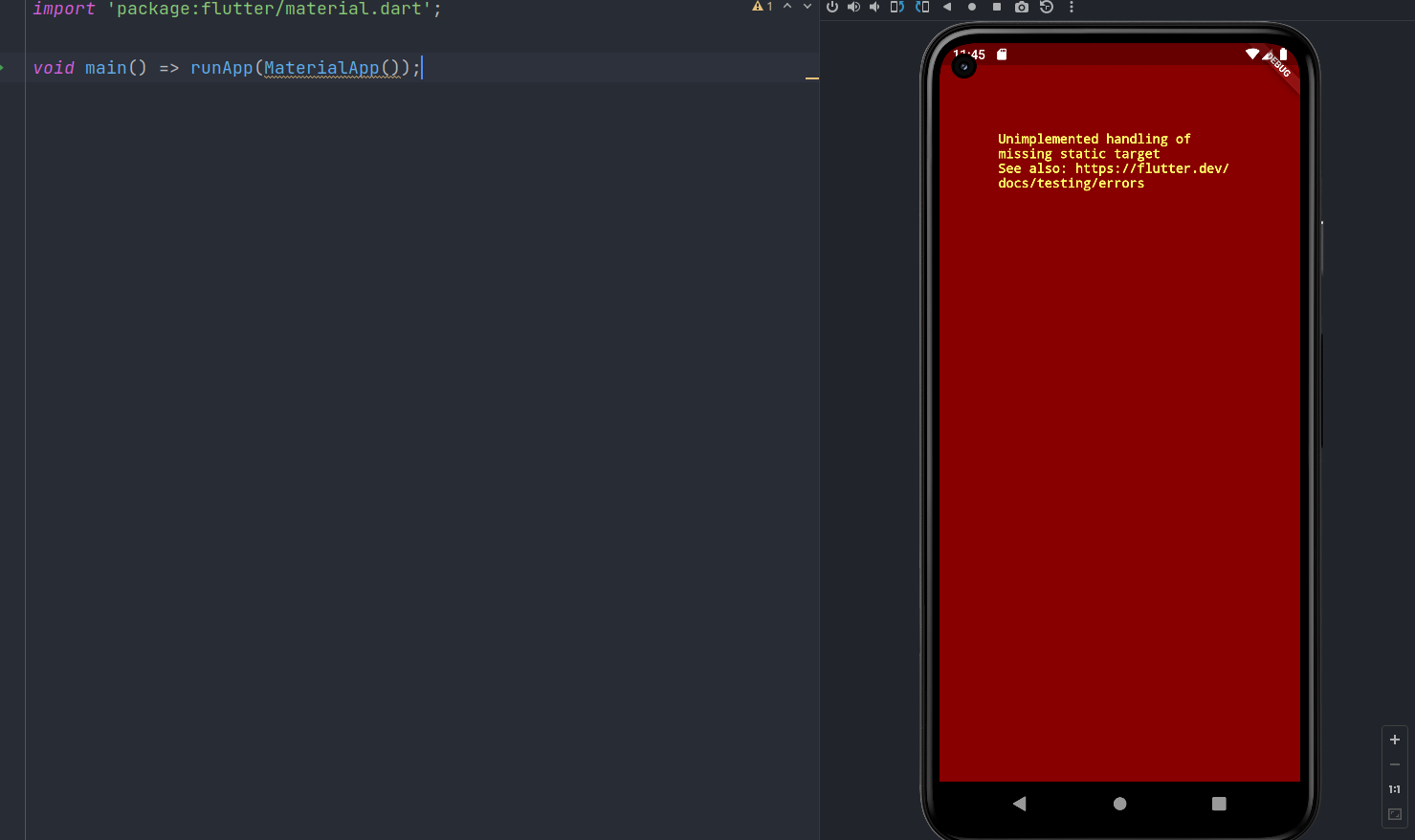
**Roll No:** CE053

**ID No:** 20CEUBG084

**Lab06 Tutorial-2**



Code of newly created flutter project.

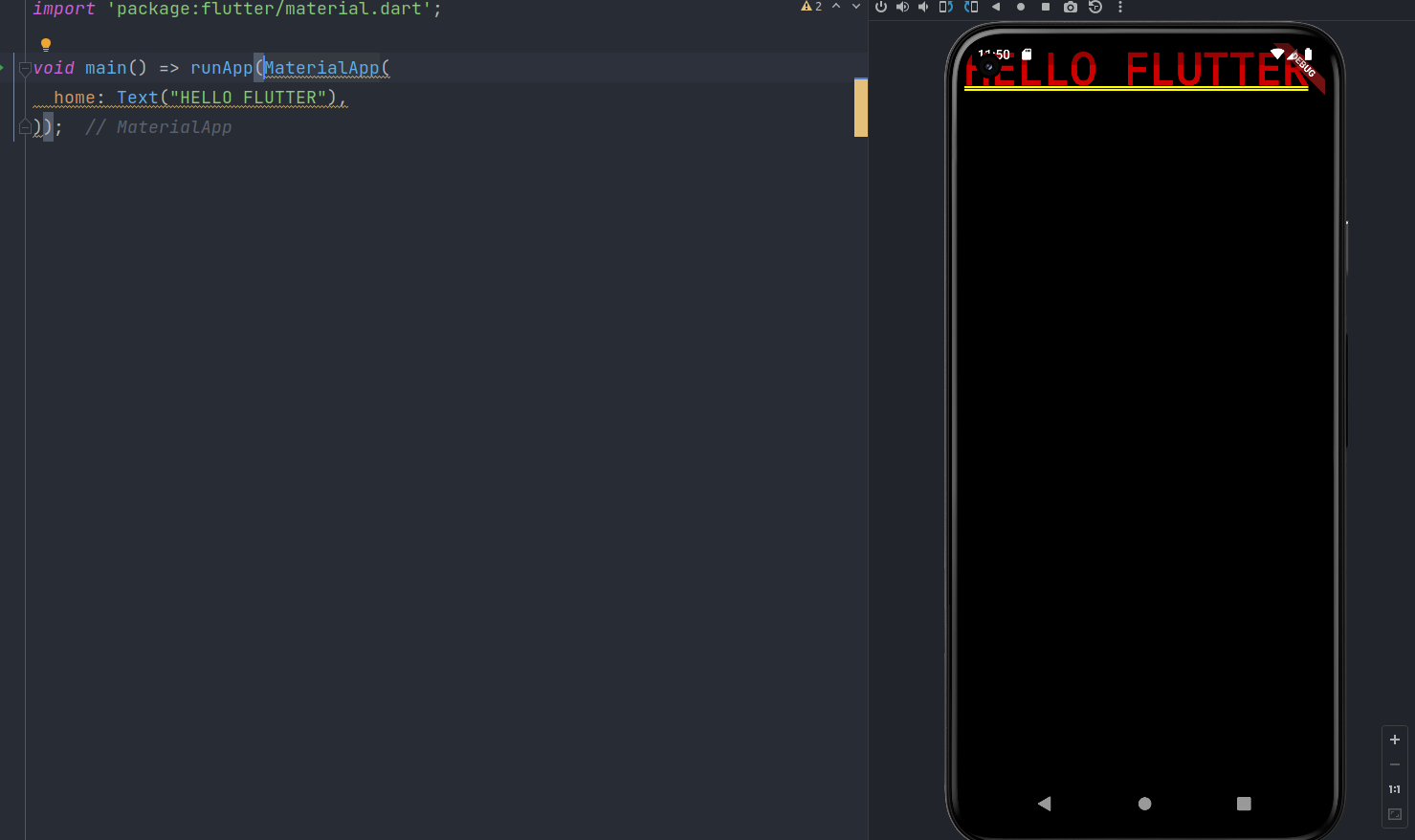


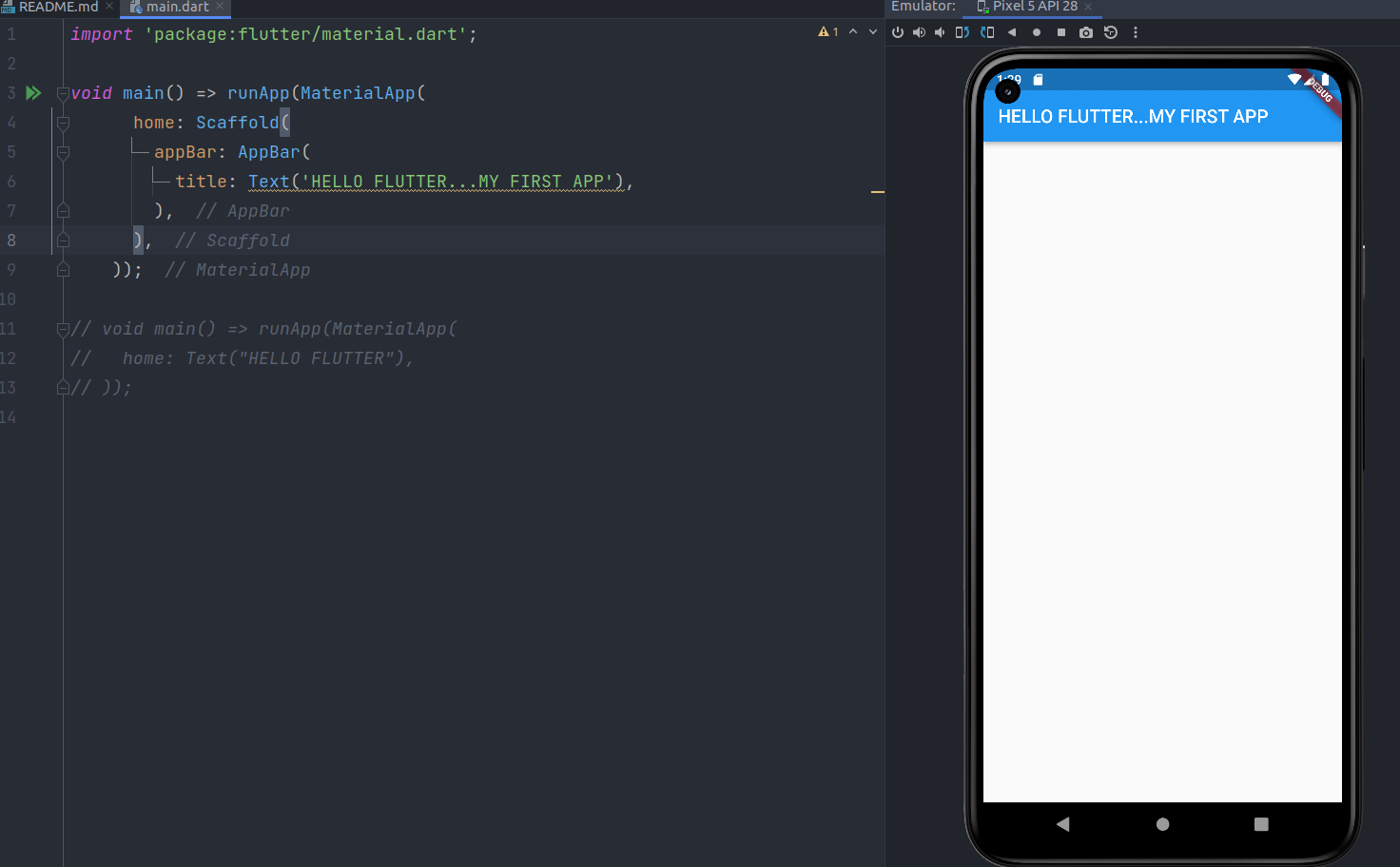
Containing class: MaterialApp

Creates a MaterialApp.

At least one of home, routes, onGenerateRoute, or builder must be non-null. If only routes is given, it must include an entry for the Navigator.defaultRouteName (/), since that is the route used when the application is launched with an intent that specifies an otherwise unsupported route.

This class creates an instance of WidgetsApp.





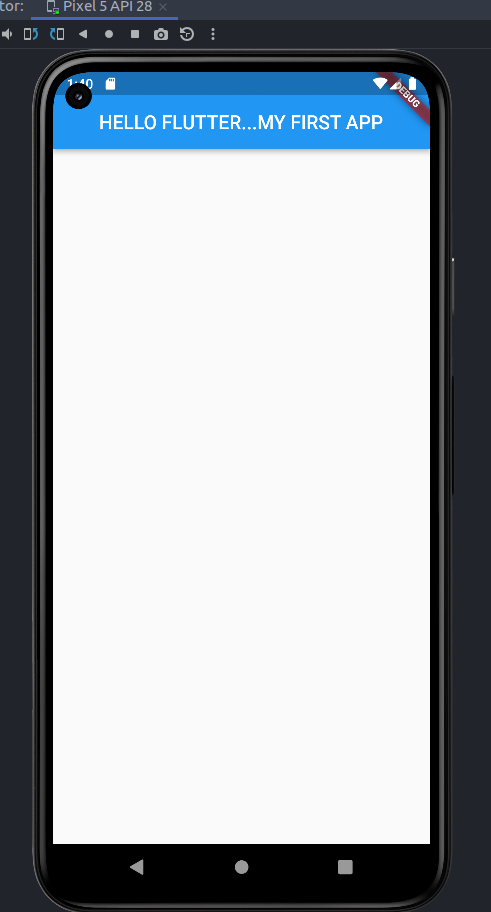
Containing class: Scaffold

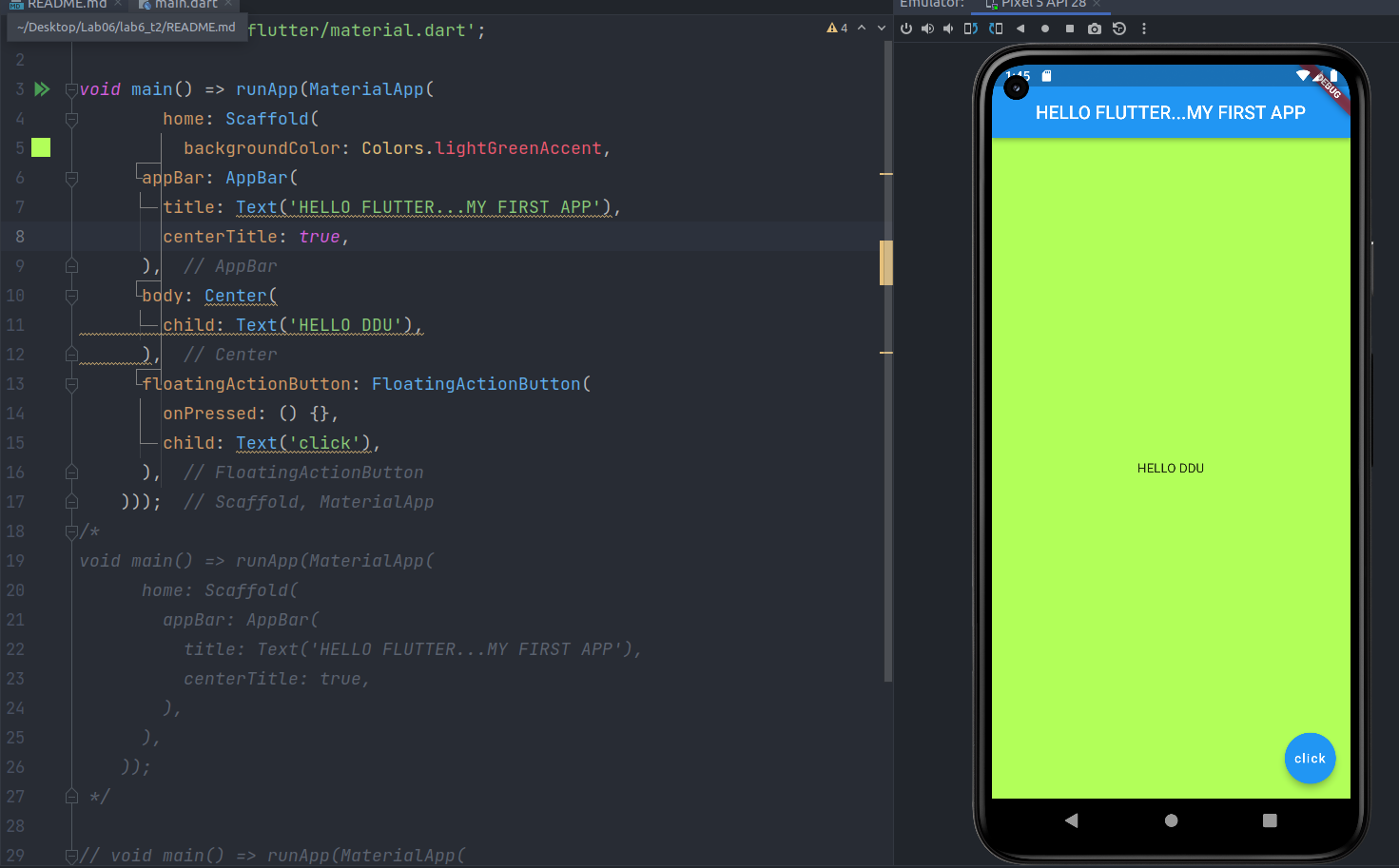
Creates a visual scaffold for material design widgets.

Scaffold is widgets built in flutter sdk..Scaffold is one type of

layout manager.

CenterTitle: true,



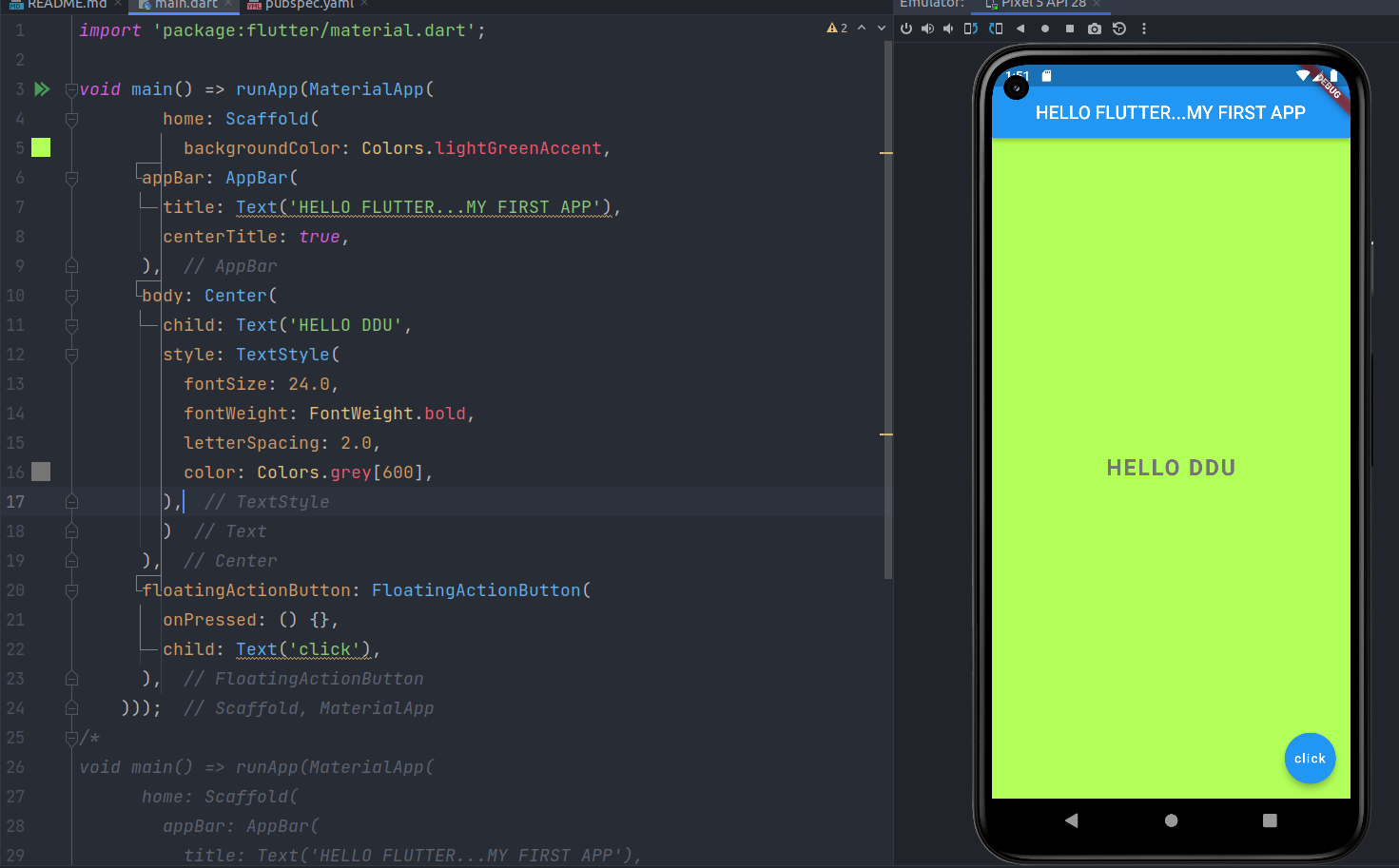


**Description:**

Containing class: FloatingActionButton

Creates a circular floating action button.

The mini and clipBehavior arguments must not be null. Additionally, elevation, highlightElevation, and disabledElevation (if specified) must be non-negative.



Type: TextStyle?

If non-null, the style to use for this text.

If the style's "inherit" property is true, the style will be merged with the closest enclosing DefaultTextStyle. Otherwise, the style will replace the closest enclosing DefaultTextStyle.

We Can add font style in pubspec.yaml file and get those dependency.

Hot reloading through stateless widget

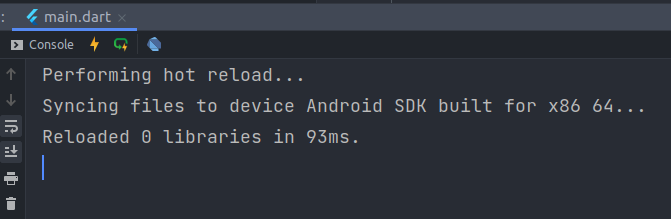
Instead of hot restart your code every time after saving work, you need to use Hot reload functionality to auto reload your work on vm.

Stateless widgets are immutable, meaning that their properties can't change—all values are final.

Stateful widgets maintain state that might change during the lifetime of the widget.

Implementing a stateful widget requires at least two classes, a StatefulWidget that creates an instance of a State class.

The StatefulWidget object is, itself, immutable and can be thrown away and regenerated, but the State object persists over the lifetime of the widget.



**Key Differences**

**Hot Reload**

It performs very as compared to hot restart or default restart of flutter.

If we are using the state in our app then hot reload will not change the

state of the app.

**Hot Restart**

It is slower than hot reload but faster than the default restart.

It doesn’t preserve the state of our it starts from the initial state of our app.

**Final Code:**

**Main.dart**

*import* 'package:flutter/material.dart';  
  
void main() => runApp(MaterialApp(  
 home: Scaffold(  
 backgroundColor: Colors.lightGreenAccent,  
 appBar: AppBar(  
 title: Text('HELLO FLUTTER...MY FIRST APP'),  
 centerTitle: *true*,  
 ),  
 body: Center(  
 child: Text('HELLO DDU',  
 style: TextStyle(  
 fontSize: 24.0,  
 fontWeight: FontWeight.bold,  
 letterSpacing: 2.0,  
 color: Colors.redAccent,  
 ),  
 )  
 ),  
 floatingActionButton: FloatingActionButton(  
 onPressed: () {},  
 child: Text('click'),  
 ),  
 )));  
/\*  
void main() => runApp(MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text('HELLO FLUTTER...MY FIRST APP'),  
 centerTitle: true,  
 ),  
 ),  
 ));  
 \*/  
  
// void main() => runApp(MaterialApp(  
// home: Text("HELLO FLUTTER"),  
// ));

**Github Link:**

<https://github.com/GauravKaklotar/SDP/tree/master/Lab06>