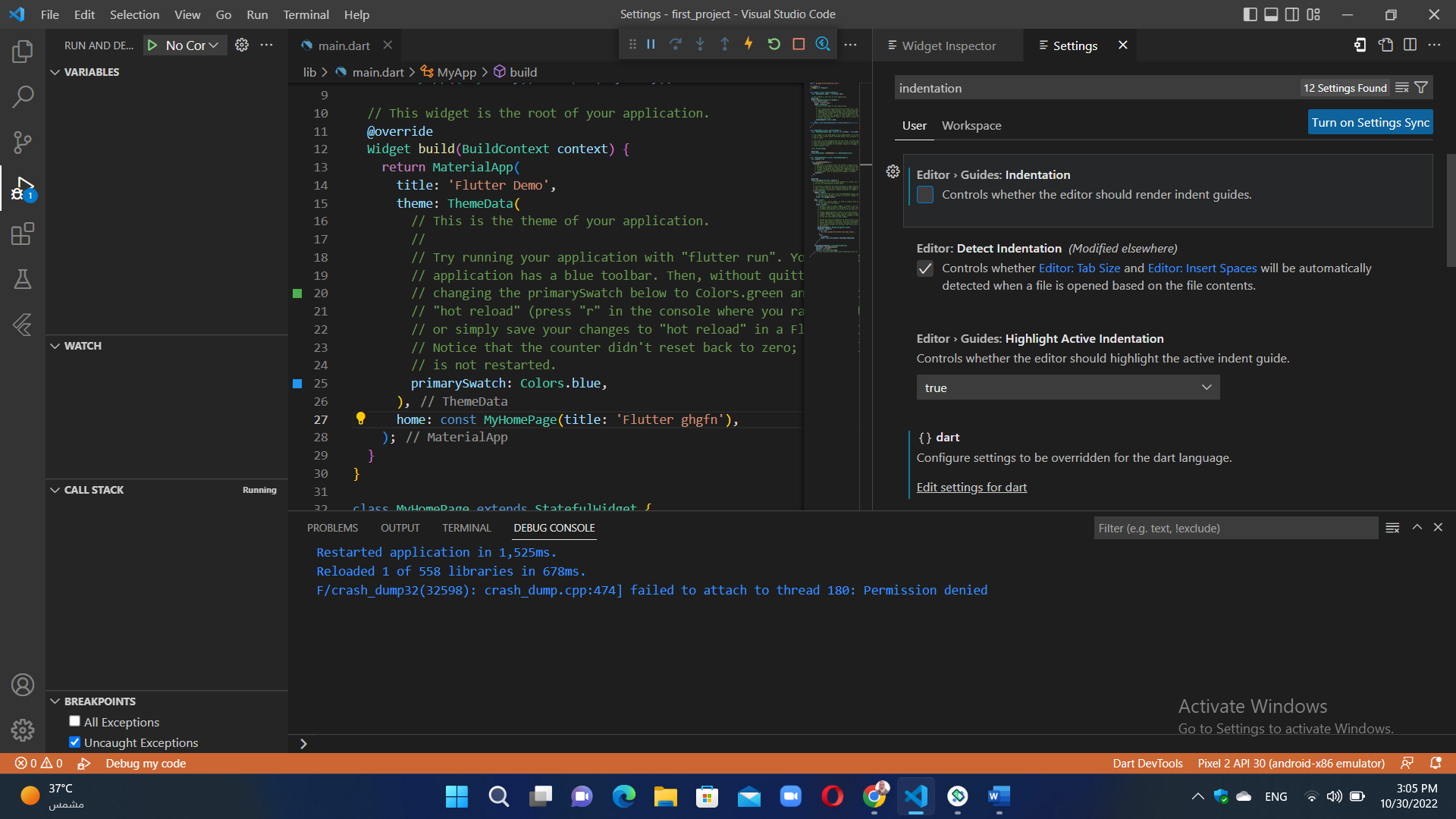
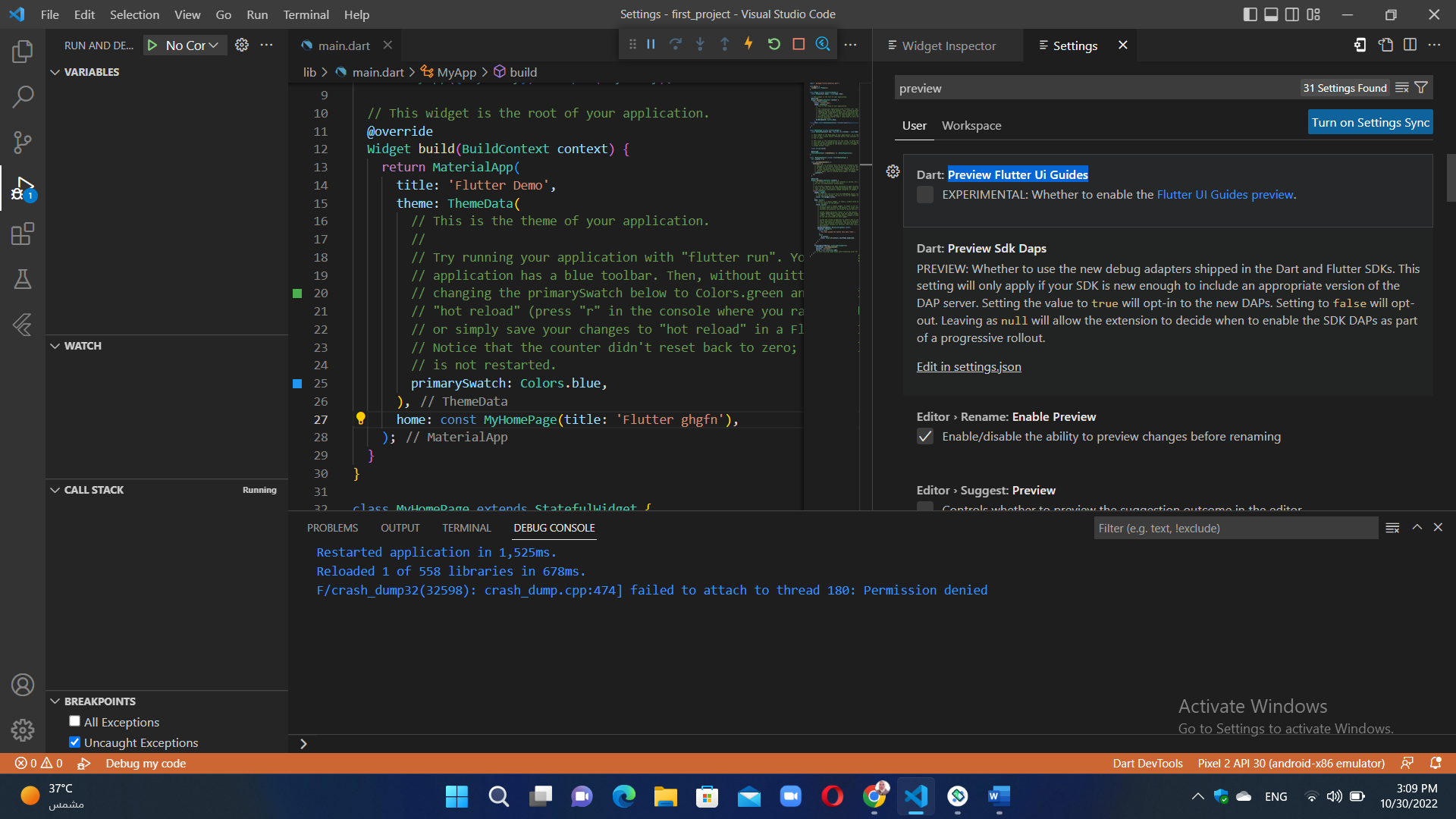


الخيار manual يعني ان لاتغيرات تحدث الا عند ضغط حفظ اي كونترول مع s

اما الخيار all في **Flutter Hot Reload On Save فيعني ان تتغير بدون ان نضغط حفظ انما اوتوماتيكيا**

ازاله الصح عن guide indentation يعمل على ازالة الخطوط من الكود

يستخدم لاظهار الخطوط المقطعه في الكود

البرنامج الاول

import 'package:flutter/material.dart';

void main() {

  runApp(

    const Center(

      child: Text(

        'Hello,!kkldjd',

        textDirection: TextDirection.ltr,

      ),

    ),

  );

}

البرنامج الثاني

import 'package:flutter/material.dart';

void main() {

runApp(const GFGapp());

}

class GFGapp extends StatelessWidget {

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

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'GeeksforGeeks',

theme: ThemeData(primarySwatch: Colors.green),

darkTheme: ThemeData(primarySwatch: Colors.grey),

color: Colors.amberAccent,

supportedLocales: {const Locale('en', ' ')},

debugShowCheckedModeBanner: false,

home: Scaffold(

appBar: AppBar(title: const Text('GeeksforGeeks')),

),

);

}

}

**Output explanation:**

* **import statement:** The *import* statement is used to import the libraries that are provided by the flutter SDK. Here we have imported the ‘material.dart’ file. We can use all the flutter widgets that implement the material design by importing this file.
* **main() function:** Like many other programming languages, we also have main function in which we have to write the statements those are to be executed when the app starts. The return type of main function is *‘void’*.
* **runApp(Widget widget) function:** The void runApp(Widget widget) takes a widget as an argument and sets it on a screen. It gives the constraints to the widget to fit into the screen. It makes the given widget the root widget of the app and other widgets as the child of it. Here we have used the *MaterialApp* as a root widget in which we have defined the other widgets.
* **MaterialApp() widget:** I have discussed MaterialApp in the beginning. Let us have a look at the different properties of the MaterialApp widget.
* **title:** This property is used to provide a short description of the application to the user. When the user press the *recent apps* button on mobile the text proceeded in *title* is displayed.
* **theme:** This property is used to provide the default theme to the application like the theme-color of the application.  
  For this, we use the inbuilt class/widget named *ThemeData()*. In *Themedata()* widget we have to write the different properties related to the theme. Here we have used the *primarySwatch* which is used to define the default themecolor of the application. To choose the color we have used *Colors* class from the material library. In ThemeData() we can also define some other properties like TextTheme, Brightness(Can enable dark theme by this), AppBarTheme, and many more.
* **home:** It is used for the default route of the app means the widget defined in it is displayed when the application starts normally. Here we have defined the *Scaffold* widget inside the home property. Inside the Scaffold we define various properties like appBar, body, floatingActionButton, backgroundColor, etc.   
  For example in the *appBar* property we have used the AppBar() widget in which as a title we have passed *‘GeeksforGeeks’* which will be displayed at the top of the application in appbar.
* The other properties in MaterialApp() are *debugShowCheckedModeBanner* (used to remove the debug tag at top corner), *darkTheme* (To request dark mode in application), *color* (For the primary color of application), *routes* (For routing table of application), *ThemeMode* (To determine which theme to be used) and *supportedLocales*contains a list of languages the app supports, etc.

البرنامج الثالث

import 'package:flutter/material.dart';

void main() => runApp(const MyApp());

class MyApp extends StatelessWidget {

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

@override

Widget build(BuildContext context) {

return MaterialApp(

home: Scaffold(

appBar: AppBar(

title: const Text("Container example"),

),

body: Container(

child:const Text("Hello! i am inside a container!",

style: TextStyle(fontSize: 20)),

),

),

);

}

}

**Container**class in flutter is a convenience widget that combines common painting, positioning, and sizing of widgets. A Container class can be used to store one or more widgets and position them on the screen according to our convenience. Basically, a container is like a box to store contents. A basic container element that stores a widget has a **margin**, which separates the present container from other contents. The total container can be given a **border**of different shapes, for example, rounded rectangles, etc. A container surrounds its child with **padding**and then applies additional constraints to the padded extent (incorporating the width and height as constraints, if either is non-null).

البرنامج الرابع

**Scaffold** is a class in [**flutter**](https://www.geeksforgeeks.org/flutter-an-introduction-to-the-open-source-sdk-by-google/) which provides many widgets or we can say [APIs](https://www.geeksforgeeks.org/introduction-to-apis/) like Drawer, SnackBar, BottomNavigationBar, FloatingActionButton, AppBar, etc. **Scaffold** will expand or occupy the whole device screen. It will occupy the available space. Scaffold will provide a [framework](https://www.geeksforgeeks.org/software-framework-vs-library/) to implement the basic material design layout of the application.

The class Hierarchy is as follows:

Object

↳ Diagnosticable

↳ Diagnosticable Tree

↳ Widget

↳ StateFul Widget

↳ Scaffold

Example

import 'package:flutter/material.dart';

void main() {

   runApp(MyApp());

}

class MyApp extends StatelessWidget{

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

       home: HomePage(),

    );

  }

  }

class HomePage extends StatelessWidget{

  @override

  Widget build(BuildContext context) {

     return Scaffold(

        appBar: AppBar( //appbar widget on Scaffold

          title:Text("iiiiiiir")

        ),

        floatingActionButton:FloatingActionButton( //Floating action button on Scaffold

          onPressed: (){},

          child: Icon(Icons.send),

        ),

        drawer: Drawer(), //drawer on scaffold, open with left menu icon

        endDrawer: Drawer(), //end drawer on scaffold, open with right menu icon

        bottomNavigationBar: BottomNavigationBar( //bottom navigation bar on scaffold

         items: [

             BottomNavigationBarItem(

               icon: Icon(Icons.add),

               label: "Button 1",

             ),

              BottomNavigationBarItem(

               icon: Icon(Icons.search),

               label: "Button 2",

             ),

             BottomNavigationBarItem(

               icon: Icon(Icons.camera),

               label: "Button 3",

             ),

        ],),

        body: Center( //content body on scaffold

           child: Text("IIIIIIIII")

        )

     );

  }

}