

Université des Mascareignes

Faculty of Information and Communication Technology

Licence Technologique en Informatique Appliquée

3ème année

Semestre 5

PMH

TP2

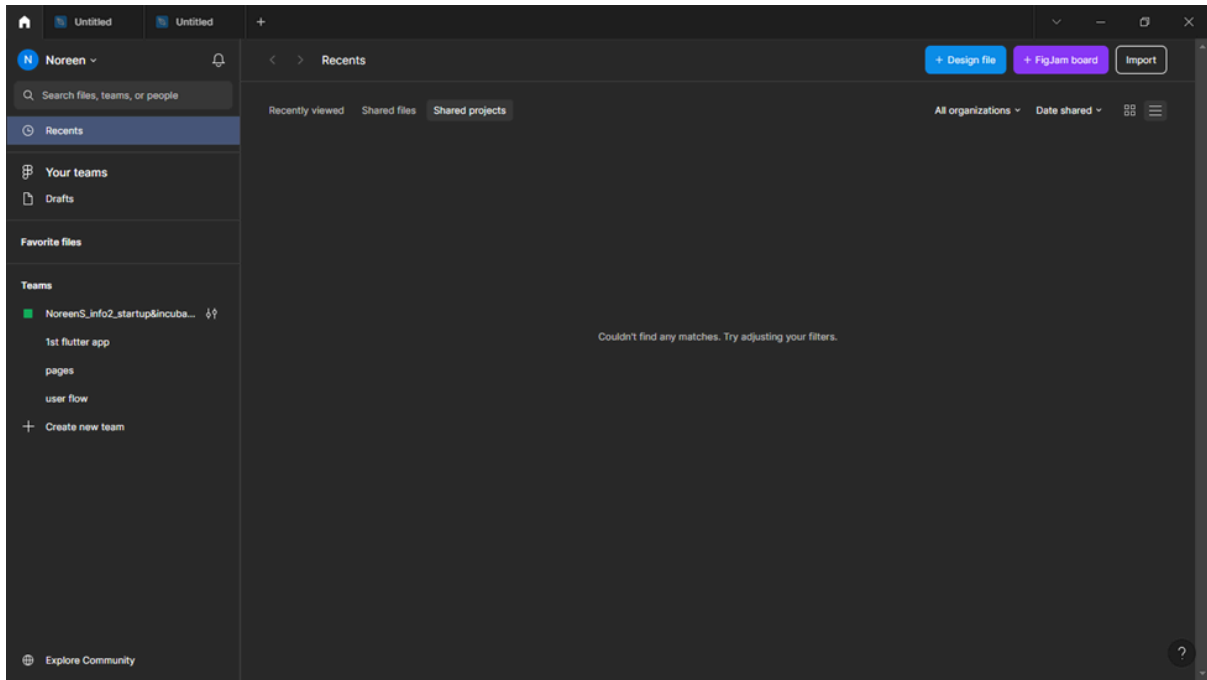
Par : Ken Addison Chan Yin Shin **THIBAUD**

Destinataire : MR. S. Beehary

Date: 03/02/2024

Task 1

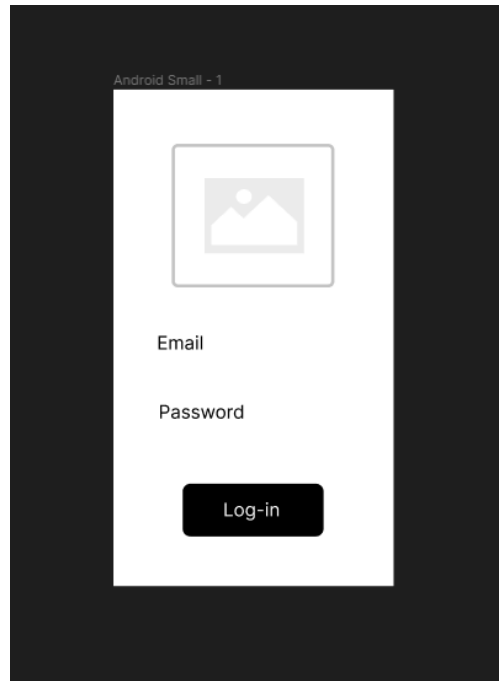
Open figma app on pc



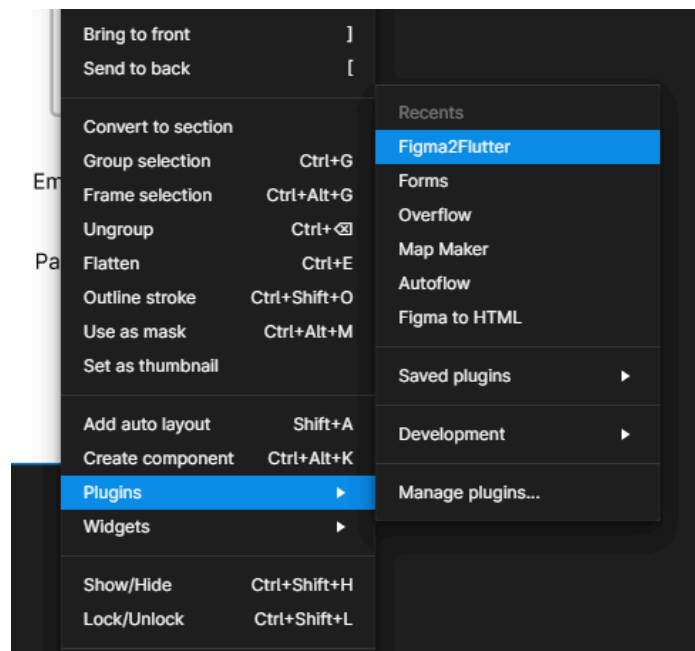
Download figma2flutter



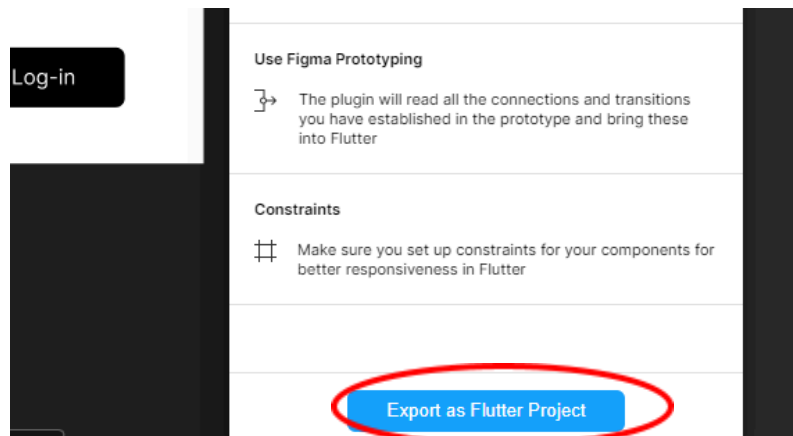
Open a design or create one



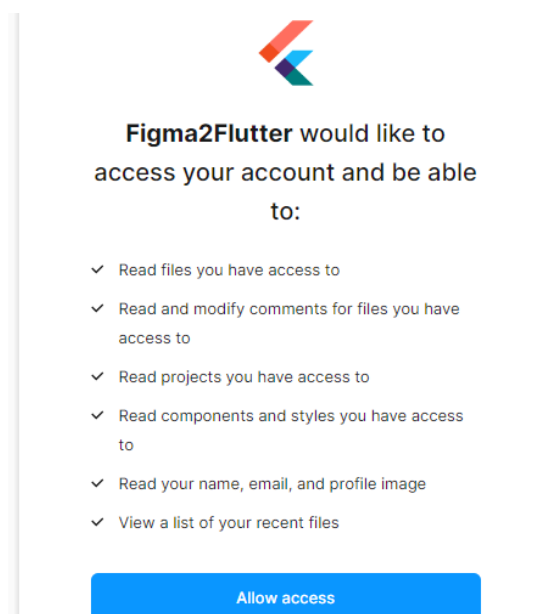
Go to plugins and chose the figma2flutter tool



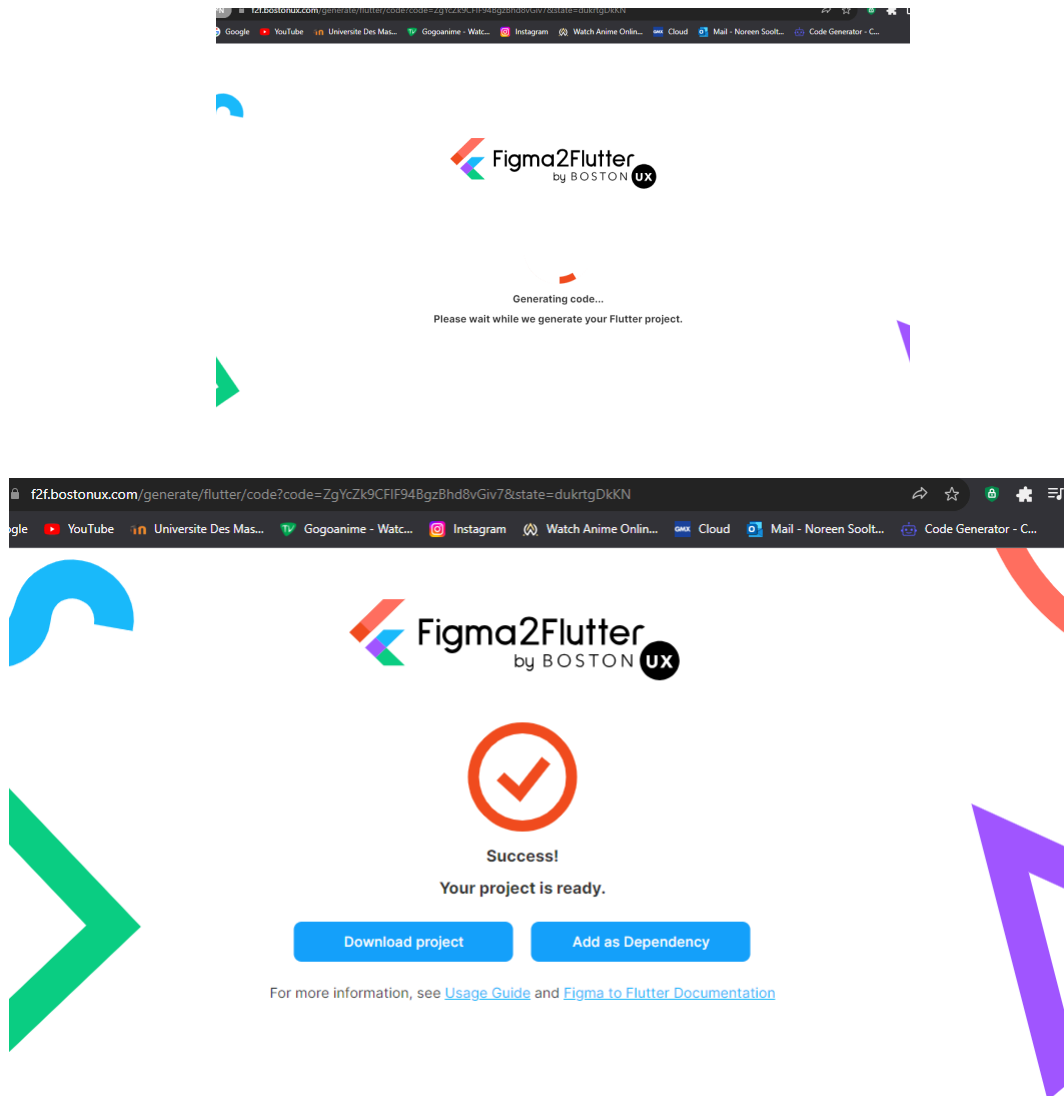
Select a screen and export it as flutter project



After copying and pasting the link of the document on the input url box, the web browser will open and we will have this message



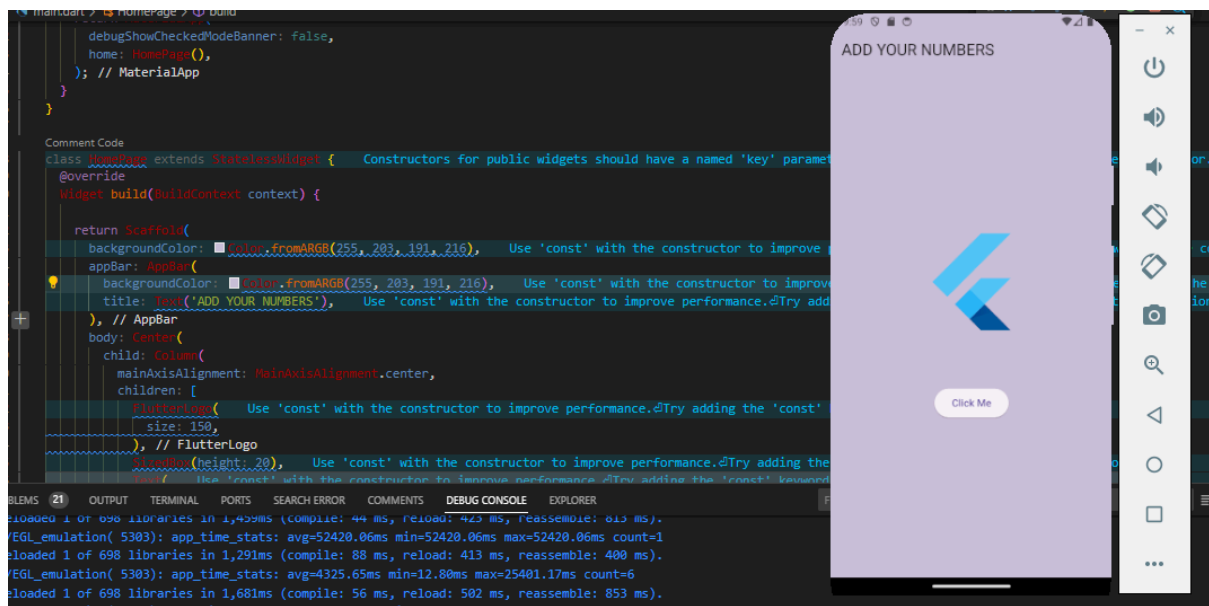
Select the blue button, “Allow access,” to allow the app to generate a Flutter project for you



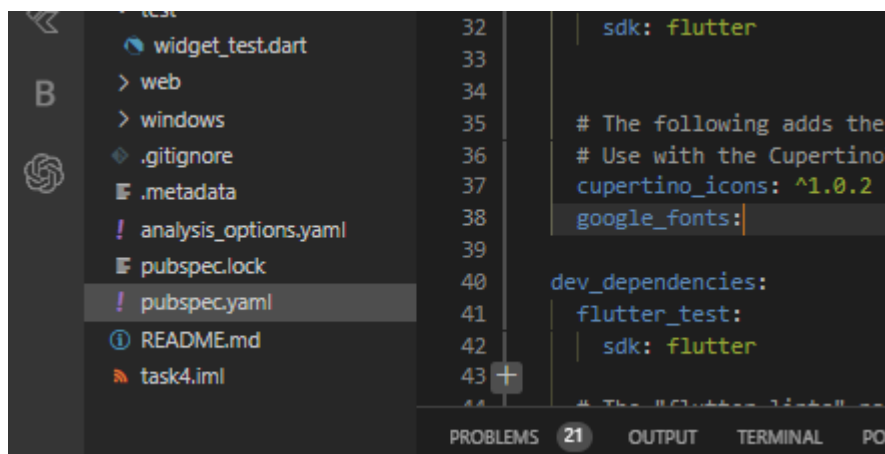
Task 2

Using the flutter documentation and some YouTube videos, we will modify the program making it more appealing

First, we will change the background color of the app

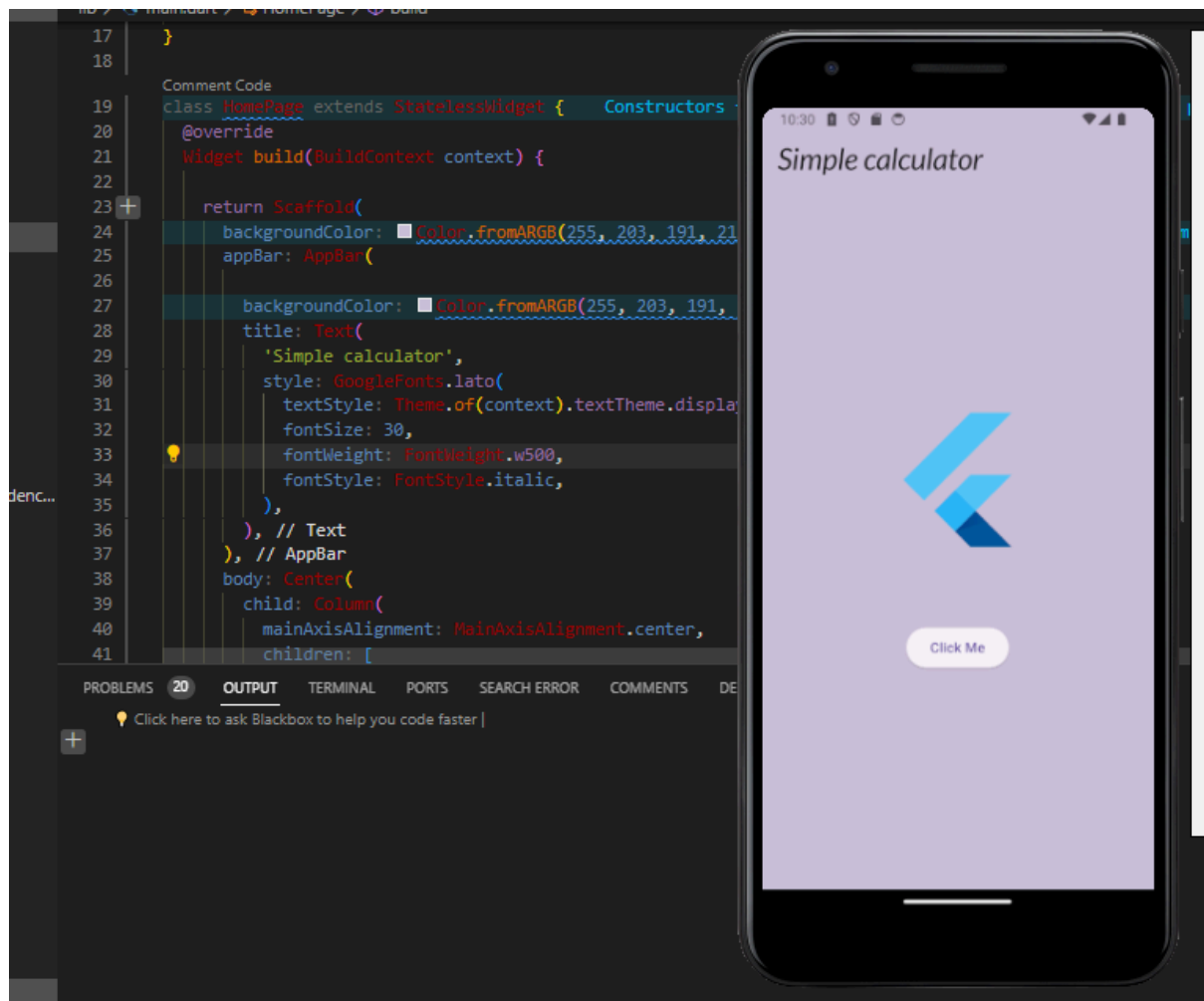


Then we can change the font, by using google fonts, we need to add the dependencies in the pubspec.yaml



Then import google fonts package

```
main.dart 1 • pubspec.yaml NewPage.dart widget_test.dart
lib > main.dart > ...
  Click here to ask Blackbox to help you code faster |
1 | import 'package:flutter/material.dart';
2 | import 'package:task4/NewPage.dart';
3 | import 'package:google_fonts/google_fonts.dart'; Unused import: 'package:
4 | +
  Run | Debug | Profile | Comment Code
5 | void main() {
6 |   runApp(MyApp());
7 | }
8 |
  Comment Code
9 | class MyApp extends StatelessWidget { Constructors for public widgets sh
10 |   @override
11 |   Widget build(BuildContext context) {
12 |     return MaterialApp(
13 |       debugShowCheckedModeBanner: false,
14 |       home: HomePage(),
15 |     ); // MaterialApp
```



Add some padding to make it more centered and change the button format, then we can also add some images, removing the flutter logo

```

body: Center(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      FlutterLogo(
        size: 150,
      ), // FlutterLogo
      SizedBox(height: 20),
      Text(
        'Click Me',
        style: TextStyle(
          color: Colors.white,
          fontSize: 20,

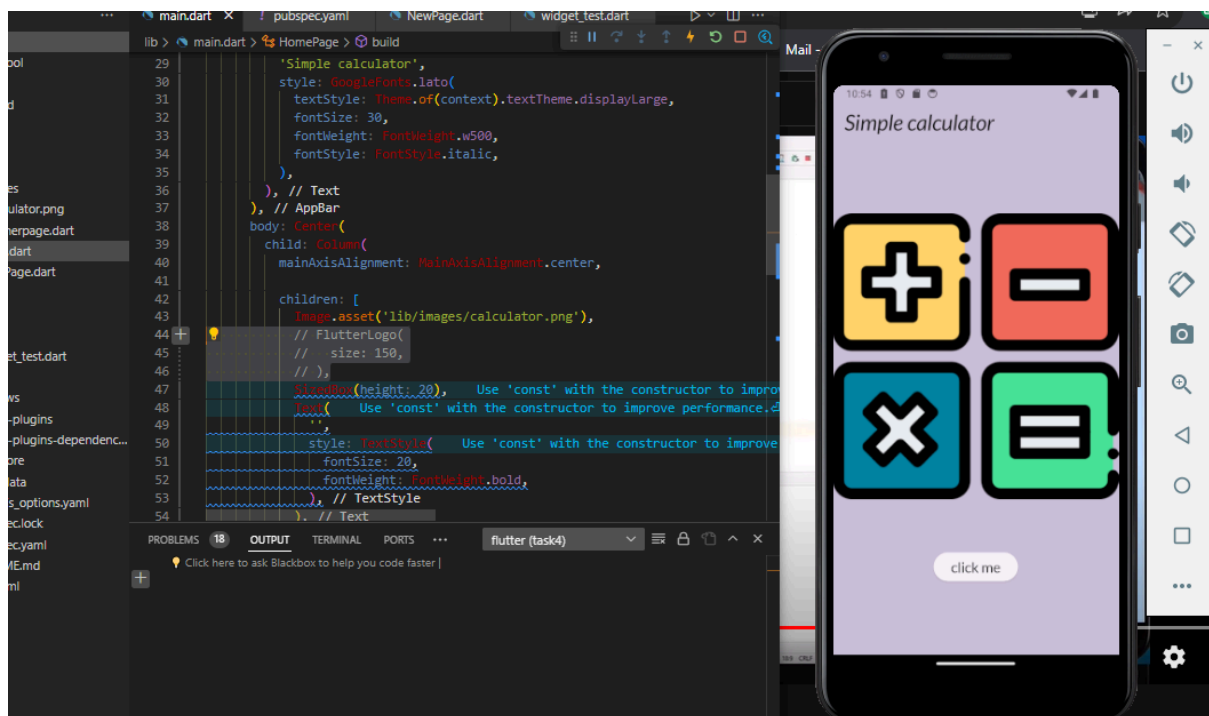
```

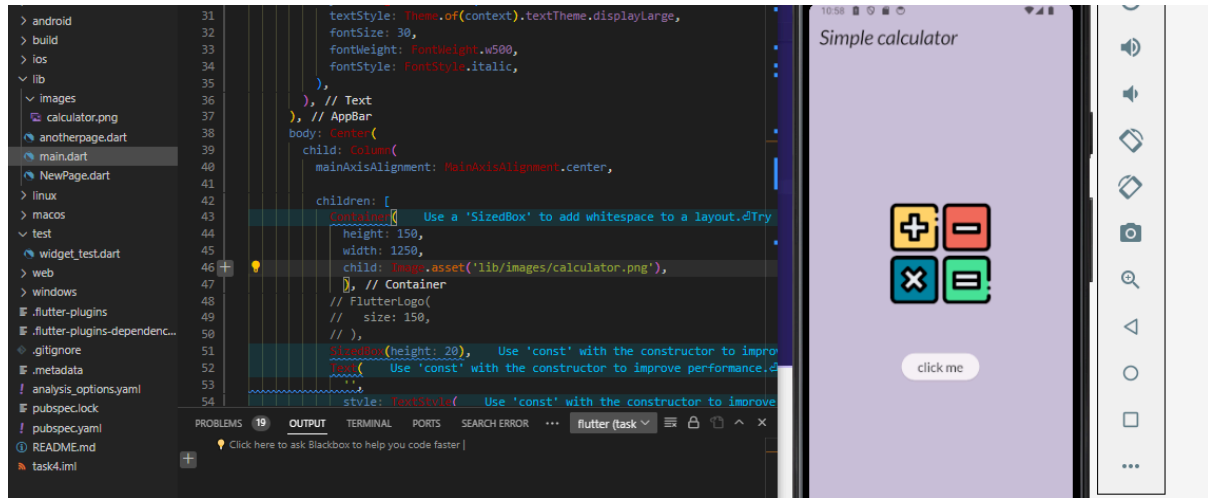

To add images we must also change the pubspec.yaml file to include the path where we will store the images

```

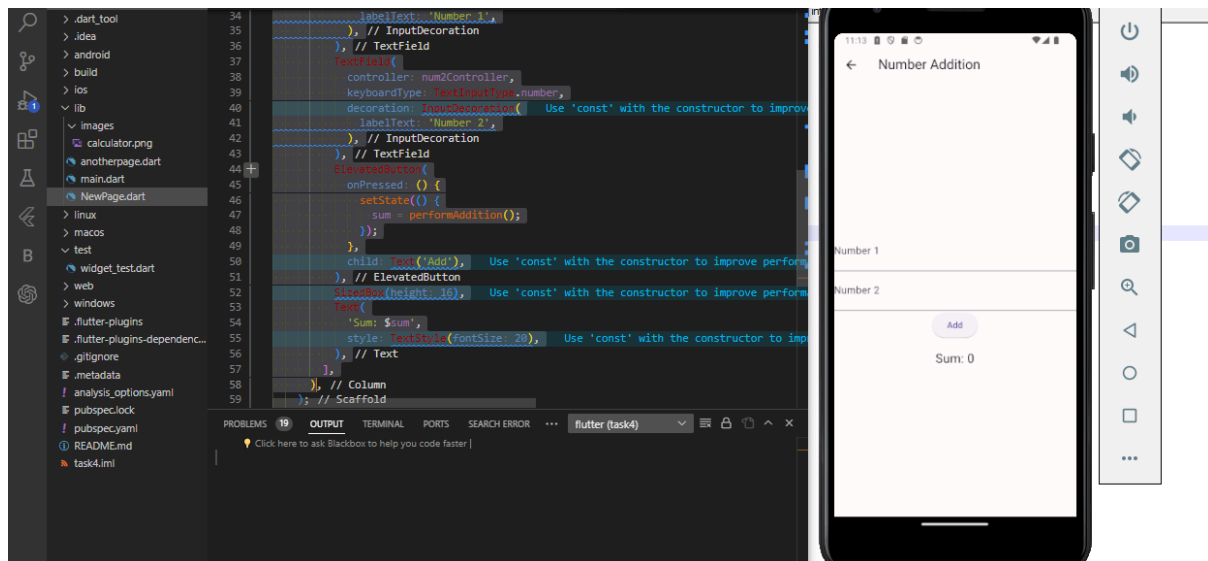
59 # the material icons class.
60 uses-material-design: true
61
62 # To add assets to your application, add an assets
63 + assets:
64 - lib/images/
65 # - images/a_dot_burr.jpeg
66 # - images/a_dot_ham.jpeg
67
68 # An image asset can refer to one or more resolutions
69 # https://flutter.dev/assets-and-images/#resolution

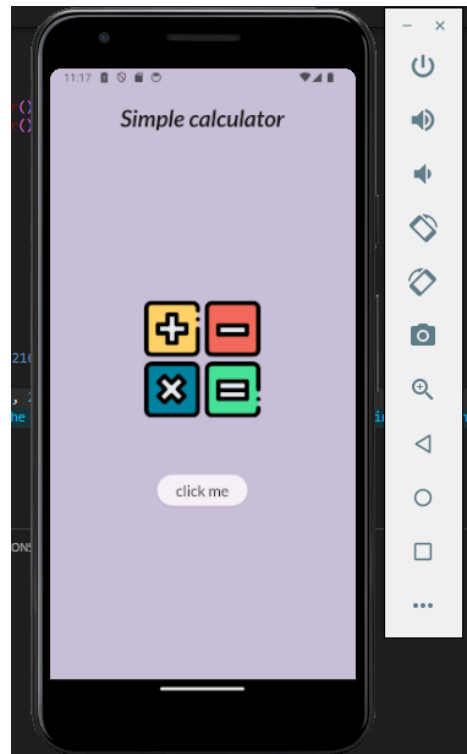
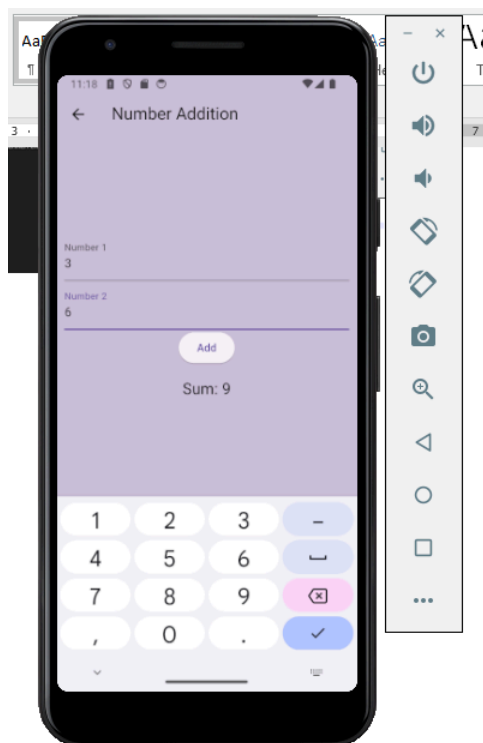
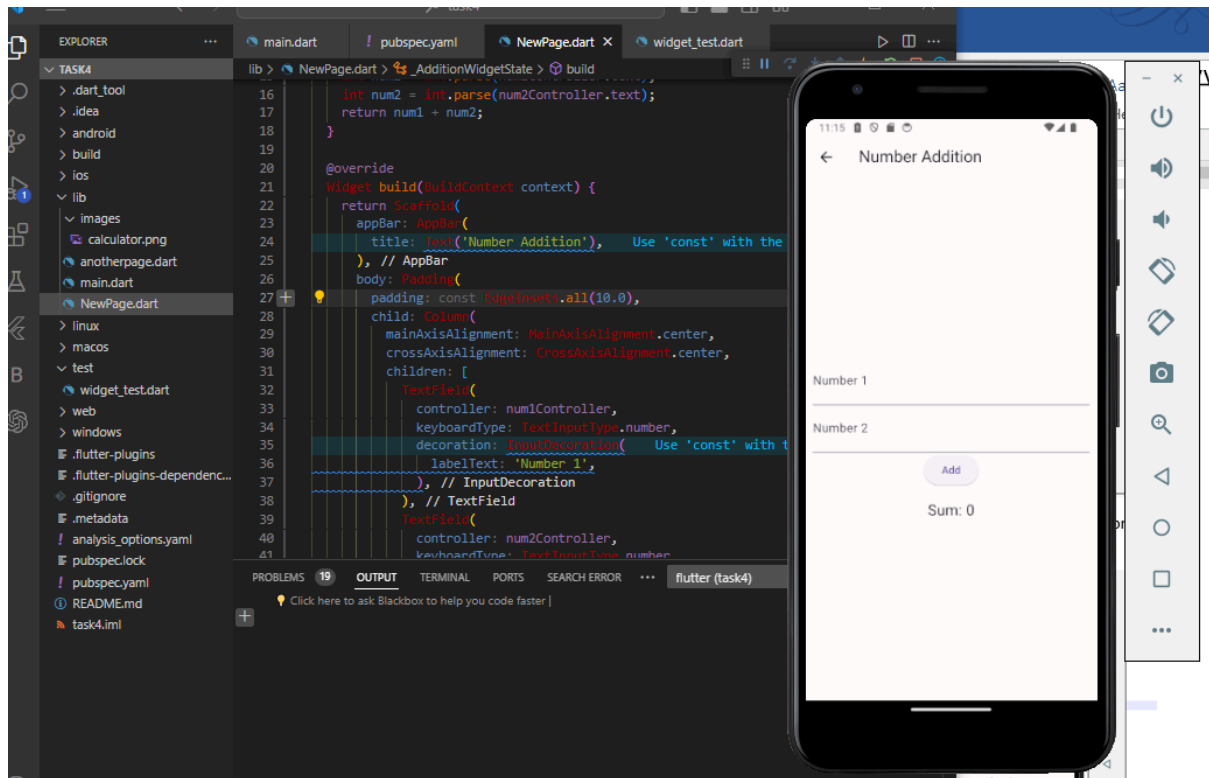
```





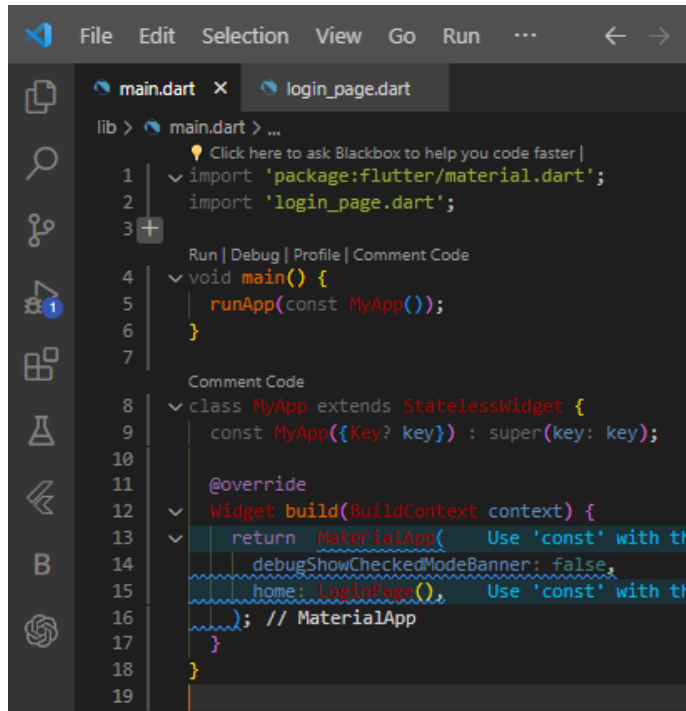
Add padding to make it more to the centre and match the background colour to the home page.





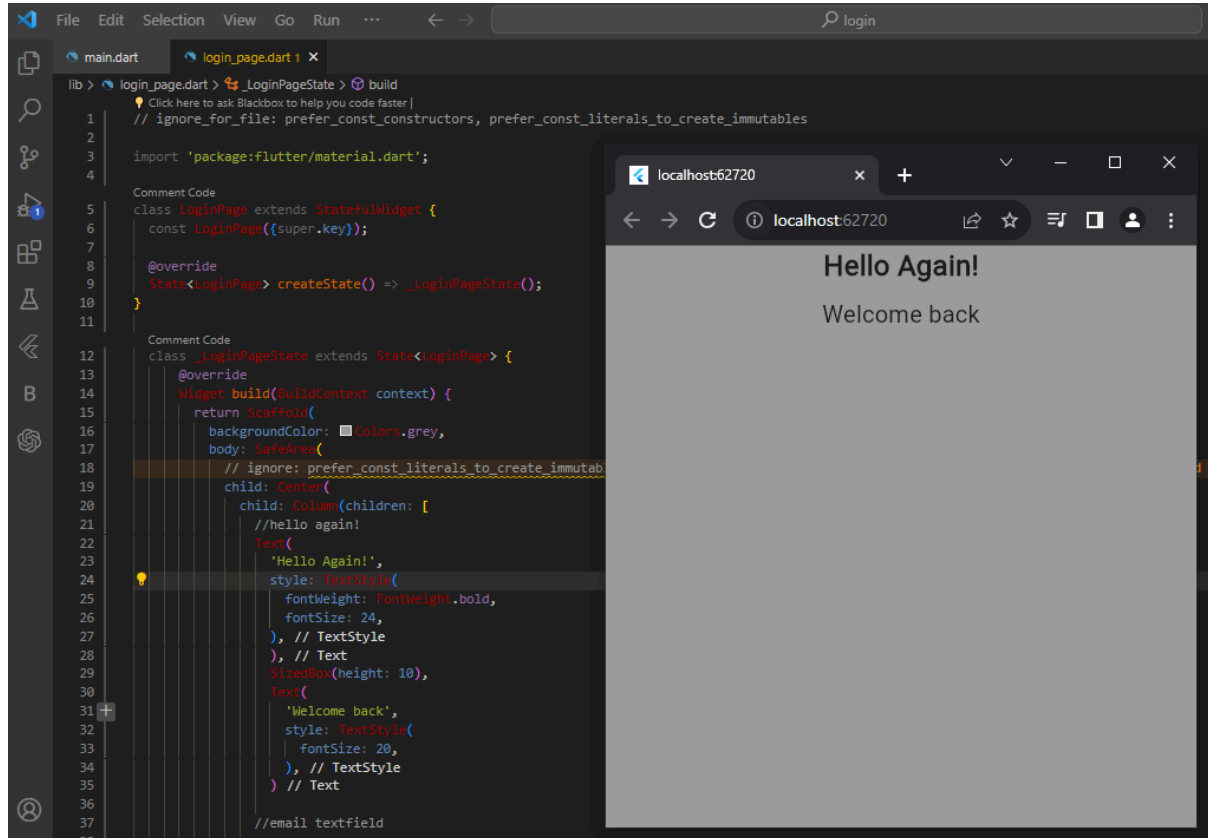
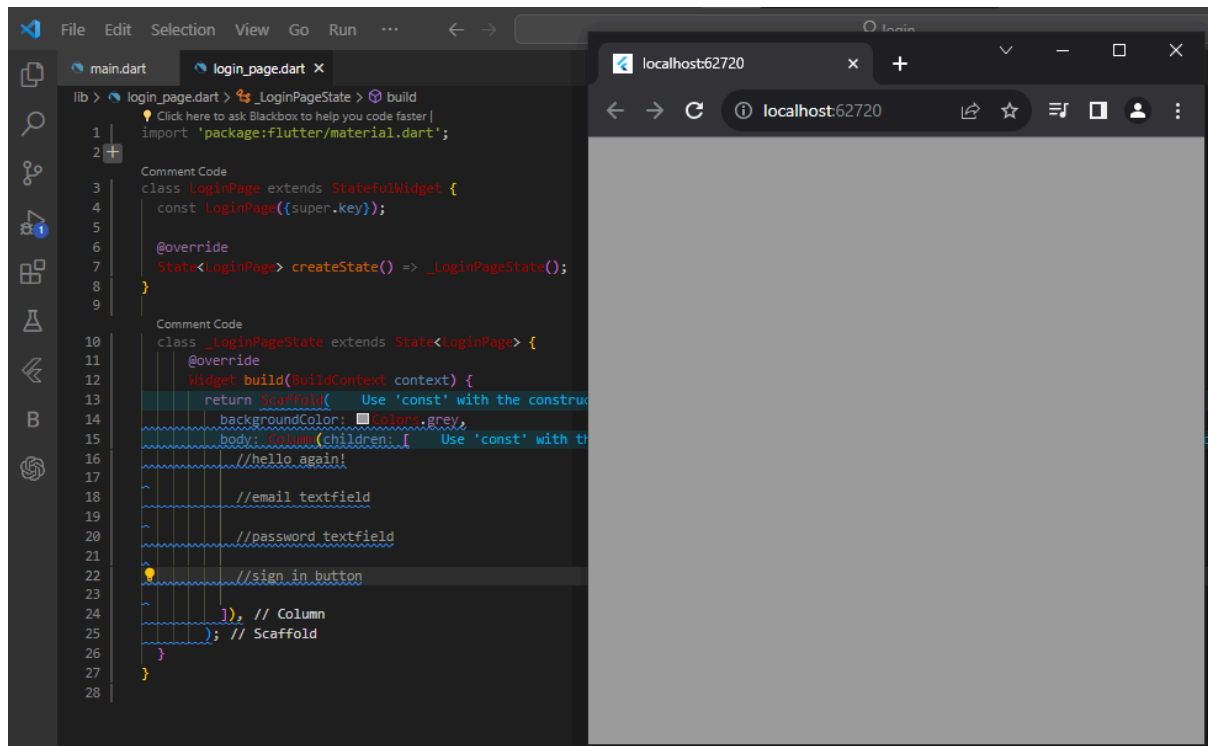
Task 3

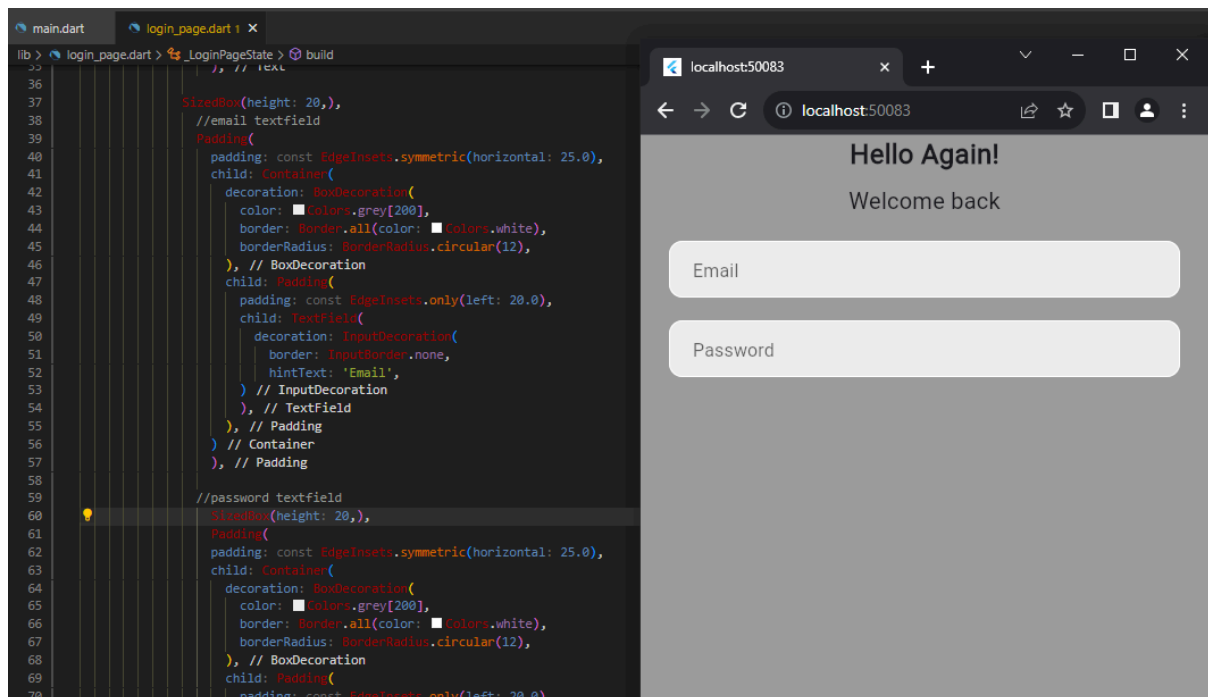
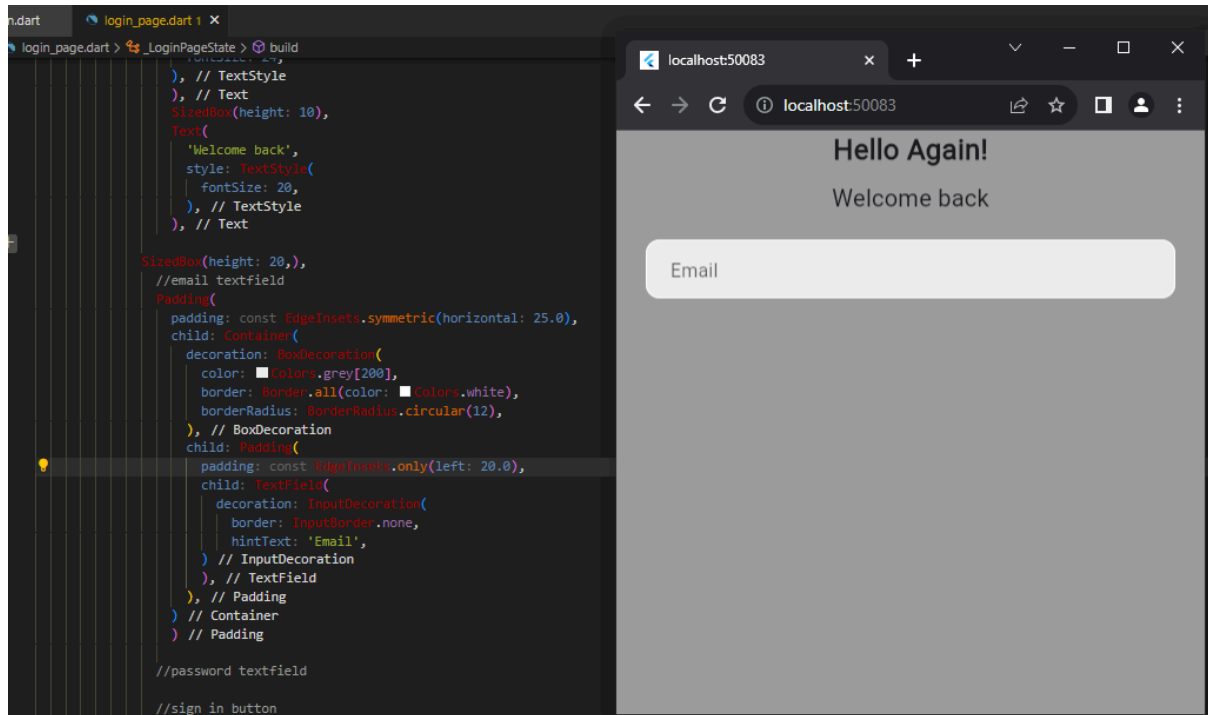
Flutter Authentication



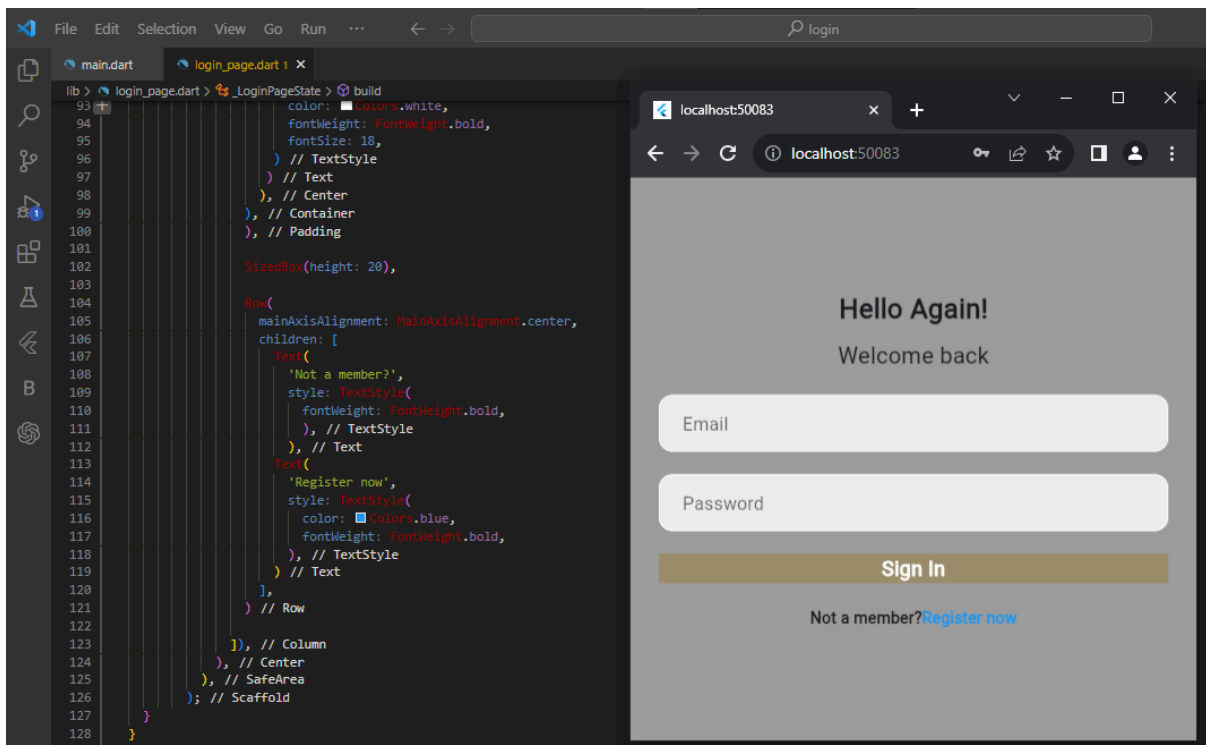
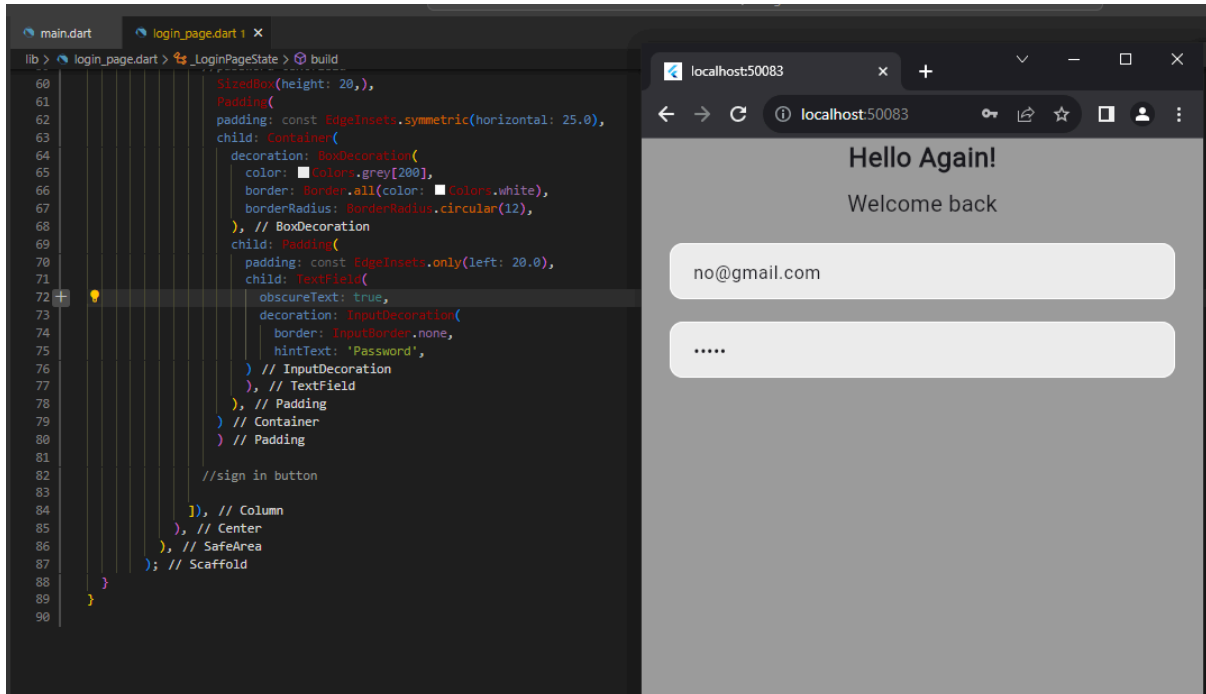
```
lib > main.dart > ...  
1  Click here to ask Blackbox to help you code faster |  
2  import 'package:flutter/material.dart';  
3  import 'login_page.dart';  
4  
5  Run | Debug | Profile | Comment Code  
6  void main() {  
7    runApp(const MyApp());  
8  }  
9  
10 Comment Code  
11 class MyApp extends StatelessWidget {  
12   const MyApp({Key? key}) : super(key: key);  
13  
14   @override  
15   Widget build(BuildContext context) {  
16     return MaterialApp(  
17       debugShowCheckedModeBanner: false,  
18       home: LoginPage(),  
19     ); // MaterialApp  
20   }  
21 }
```

Initially, we created a Flutter project consisting of two pages. The main.dart file will automatically navigate to the login page. We organise the content within a column, incorporating various elements such as a welcome message, an email text field, a password text field, and a sign-in message. Below are snippets of the code utilised for this implementation:



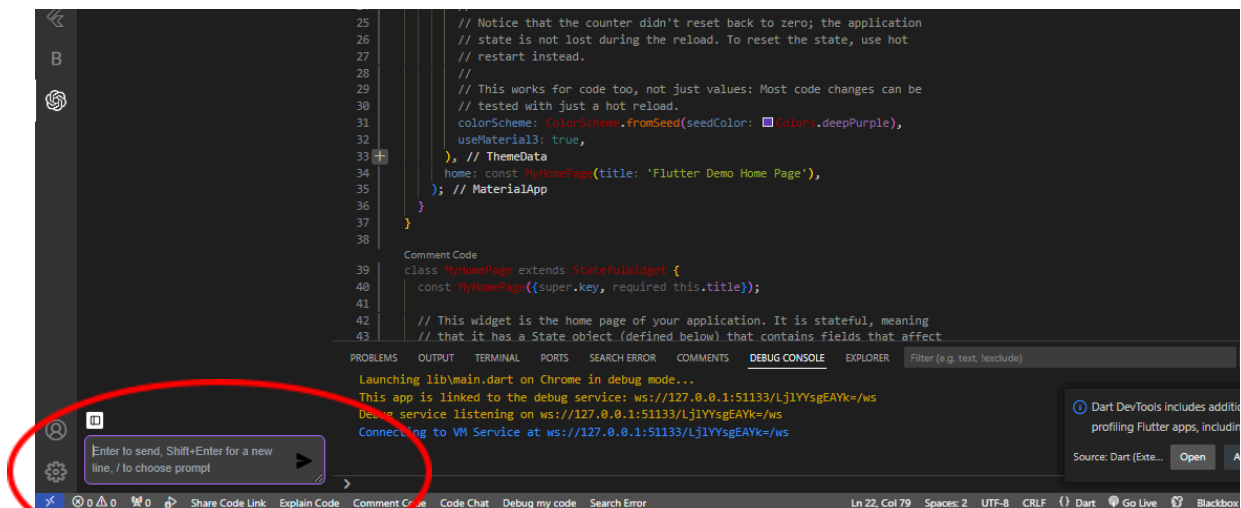
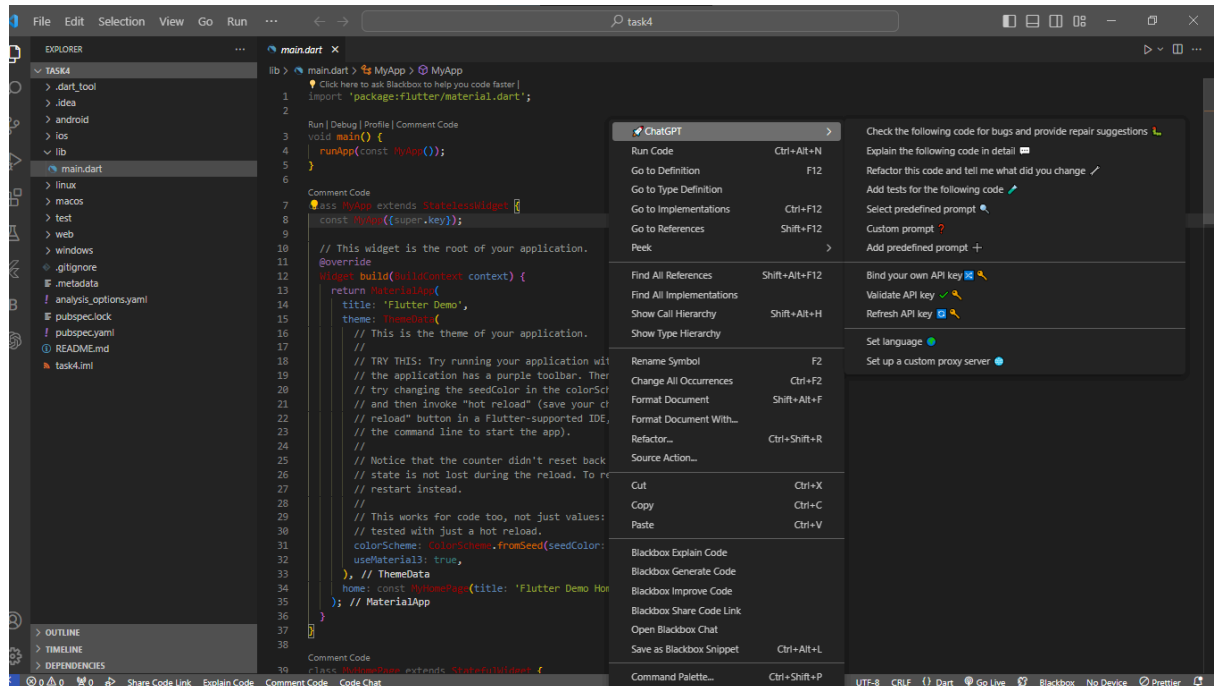


Adding 'obscure text' to make the password not visible



Task 4

Create a new flutter program, right click then select chatGPT plugin



All the necessary prompts can be presented in the input provided here. In this instance, I requested ChatGPT to enhance the appearance of the homepage, and it will furnish you with the complete code, along with instructions on incorporating it into your existing code.

You

code a pretty home page for flutter

ChatGPT

Sure! Here's an example of how you can code a basic home page in Flutter:

```
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(
        title: Text('Home Page'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment:
            MainAxisAlignment.center,
          children: [
```

CHATGPT

```
    ],
    child: Text('Click Me'),
  ),
],
),
);
}
}
```

This code will create a simple home page with an app bar, a Flutter logo, a welcome message, and a button. You can customize it further by adding more widgets or modifying the existing ones to suit your needs.

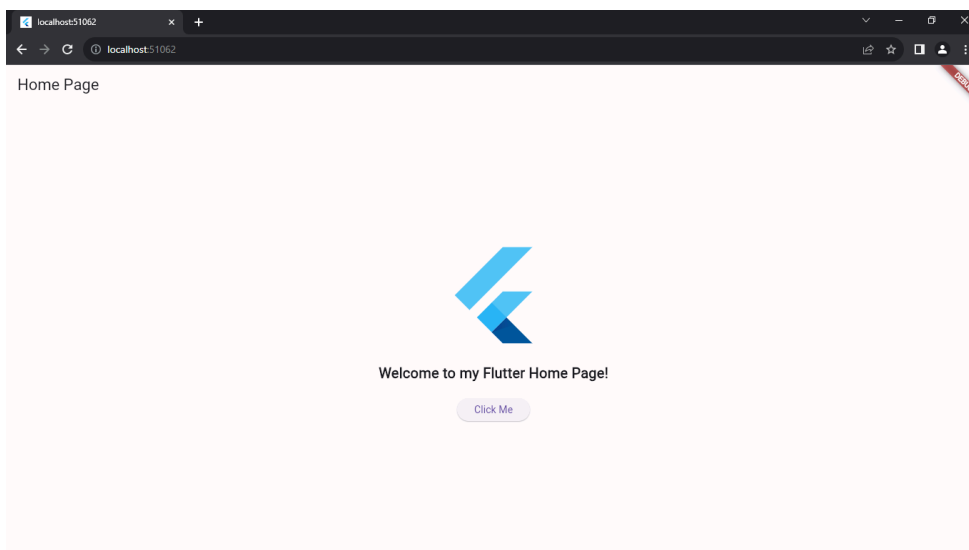
Here are some key points to note in the code:


- The main function initializes the app by running the `MyApp` widget.
- The `MyApp` widget is a `MaterialApp` widget that sets up the basic material design structure for the app and sets the `HomePage` as the initial route.
- The `HomePage` widget is a `StatelessWidget` that defines the structure of the home page.
- The `Scaffold` widget provides a basic layout structure for the page. It includes an app bar and a body.
- The `Center` widget is used to center its child widgets both horizontally and vertically.
- The `Column` widget is used to stack the child widgets vertically in a column.
- The `FlutterLogo` widget displays the Flutter logo.
- The `Text` widget displays the welcome message.
- The `ElevatedButton` widget creates a clickable button.

Feel free to modify and add more widgets as per your requirements to create a more complex and visually appealing home page.


Enter to send, Shift+Enter for a new line, / to choose prompt

This is the home page that we obtain



 You

go to a new page when clicking on click me on flutter

 ChatGPT

To navigate to a new page in Flutter when clicking on a widget, you can make use of the `Navigator` class. Here's how you can achieve it step by step:

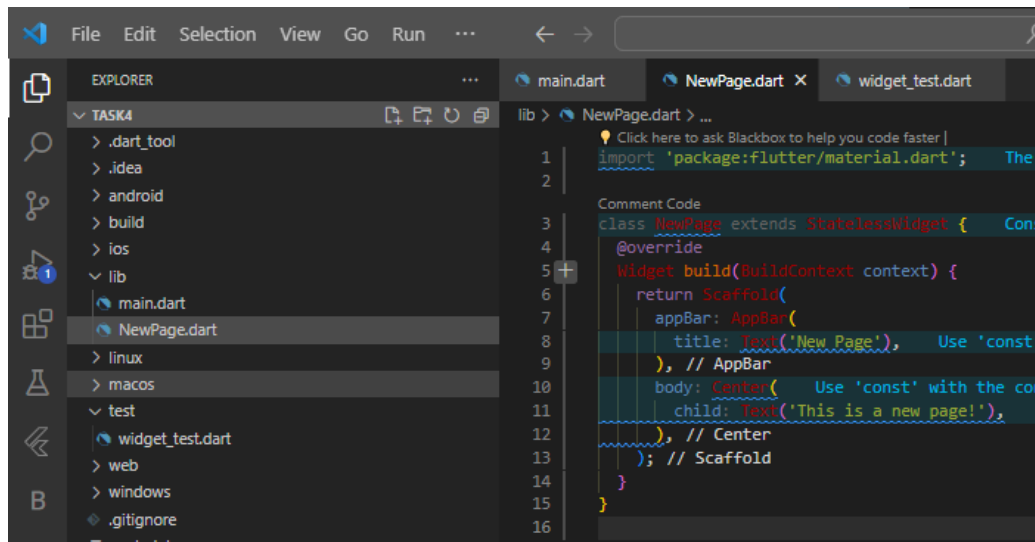
Define a new page widget that you want to navigate to. Let's call it `NewPage`. It can be defined as a separate Dart file or within the same file:

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

class NewPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('New Page'),
      ),
      body: Center(
        child: Text('This is a new page!'),
      ),
    );
  }
}
```

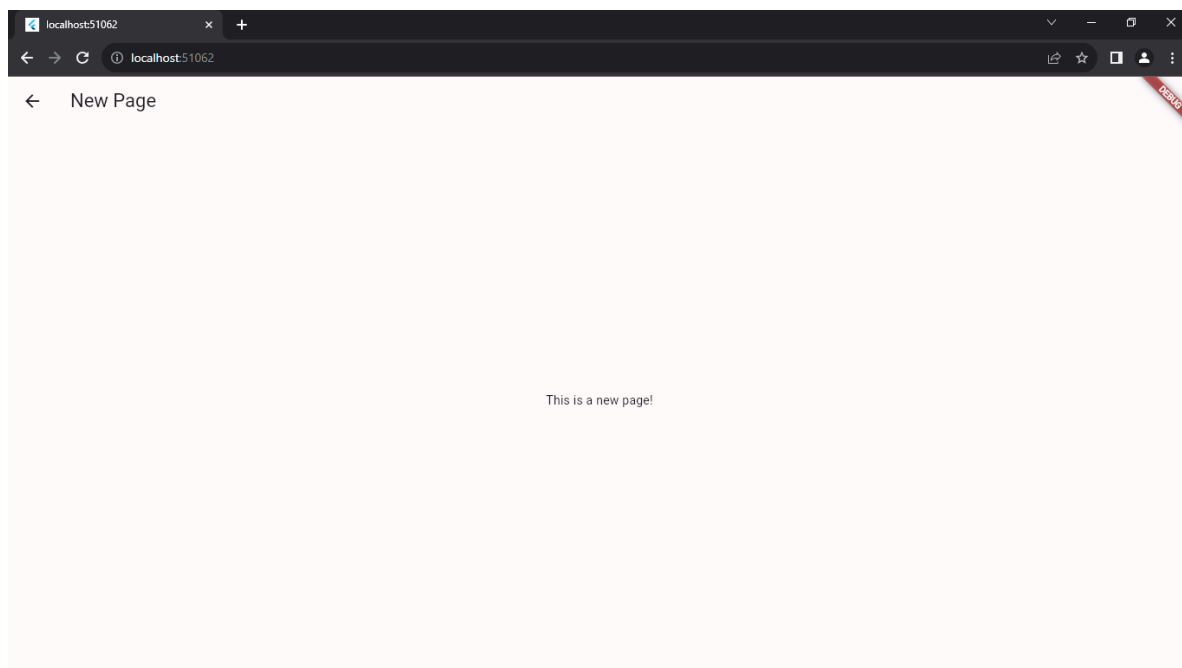
```
fontSize: 20,  
fontWeight: FontWeight.bold,  
), // TextStyle  
), // Text  
SizedBox(height: 20), Use 'const' with the constructor  
ElevatedButton(  
  onPressed: () {  
    // Add your desired action here  
  },  
  
  //added  
  child: GestureDetector(  
    onTap: () {  
      Navigator.push(  
        // Code to navigate to the new pageNavigator.push(  
          context,  
          MaterialPageRoute(builder: (context) => NewPage()),  
        );  
      },  
    ),  
  ),  
  child: Container( Unnecessary instance of 'Container'  
    child: Text('Click Me'), Use 'const' with the const  
  ), // Container  
), // GestureDetector
```

and follow to guide to make a new dart file containing the chatgpt code



```
File Edit Selection View Go Run ...  
EXPLORER  
TASK4  
  .dart_tool  
  .idea  
  android  
  build  
  ios  
  lib  
    main.dart  
    NewPage.dart  
  linux  
  macos  
  test  
    widget_test.dart  
  web  
  windows  
  .gitignore  
  ...  
lib > NewPage.dart > ...  
1 | import 'package:flutter/material.dart';  
2 |  
3 |  
4 | class NewPage extends StatelessWidget {  
5 |   @override  
6 |   Widget build(BuildContext context) {  
7 |     return Scaffold(  
8 |       appBar: AppBar(  
9 |         title: Text('New Page'),  
10 |       ),  
11 |       body: Center(  
12 |         child: Text('This is a new page!'),  
13 |       ),  
14 |     );  
15 |   }  
16 | }
```

And here is the result



Functionality can be incorporated into the app to enable user input and subsequent addition. I utilised a tool to generate the code for this, and here you'll find the prompt, the generated code, and the implementation details.

CHATGPT

in your Flutter app. For example, you can pass it to another page or use it to perform certain actions.

That's it! You have now created an input page in Flutter where you can take user inputs. Remember to navigate to this input page from your main page or wherever needed in your app using the `Navigator` class.

You
take 2 numbers and add in flutter

ChatGPT

To take two numbers as input and add them in Flutter, you can do the following:

- Create a `StatefulWidget` or add this code inside the `build` method of your existing widget.
- Add two `TextEditingController` objects to keep track of the user input.
- Create a method to perform the addition and return the sum.
- Use the `TextField` widget to allow the user to input two numbers.
- Add a button, such as `ElevatedButton`, to trigger the addition.
- Display the result using the `Text` widget.

Here's an example code snippet to help you understand the process:

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

class AdditionWidget extends StatefulWidget {
  @override
  _AdditionWidgetState createState() => _AdditionWidgetState();
}

class _AdditionWidgetState extends State<AdditionWidget> {
  TextEditingController num1Controller = TextEditingController();
  TextEditingController num2Controller = TextEditingController();
  int sum = 0;

  int performAddition() {
    int num1 = int.parse(num1Controller.text);
```

Enter to send, Shift+Enter for a new line, / to choose prompt

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

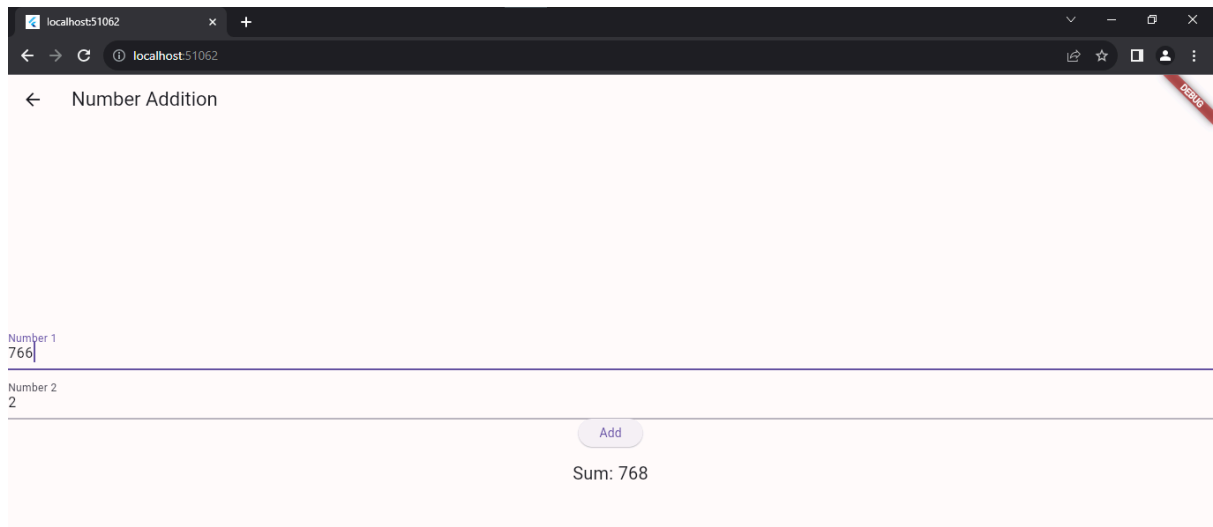
class NewPage extends StatefulWidget {
  @override
  _AdditionWidgetState createState() => _AdditionWidgetState();
}

class _AdditionWidgetState extends State<AdditionWidget> {
  TextEditingController num1Controller = TextEditingController();
  TextEditingController num2Controller = TextEditingController();
  int sum = 0;

  int performAddition() {
    int num1 = int.parse(num1Controller.text);
    int num2 = int.parse(num2Controller.text);

    return num1 + num2;
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Number Addition'),
      ),
      body: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
          TextField(
            controller: num1Controller,
            keyboardType: TextInputType.number,
            decoration: InputDecoration(
              labelText: 'Number 1',
            ),
          ),
          TextField(
            controller: num2Controller,
            keyboardType: TextInputType.number,
            decoration: InputDecoration(
              labelText: 'Number 2',
            ),
          ),
          ElevatedButton(
            onPressed: () {
              setState(() {
                sum = performAddition();
              });
            },
            child: Text('Add'),
          ),
        ],
      ),
    );
  }
}
```



Mobile version

