Mobile Development :

8 : Flutter for Mobile Development : Part 3

State Management + Interactive MVP

Professor Imed Bouchrika

National School of Artificial Intelligence
imed.bouchrika@ensia.edu.dz

Outline :

- **Section 1: State Management**
  - ***Stateless & Stateful***
  - ***State Management (GetX, Provider..)***
- **Section 2 : Building Your MVP**
  - ***Project Structuring***
  - ***Flutter UIs for MVP Building***
    - ***PageView, TabView, Nav Widgets…***
- **Section 3 : Interactive MVP**
  - ***Connecting to Web***
  - ***ProgressBar***
  - ***FutureBuilder***

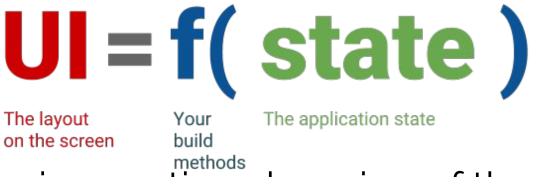
Section 1

State Management in Flutter

State Management in Flutter

- **Stateless & Stateful Widgets**
  - Stateful Widgets are used to build screens and Widgets (inside

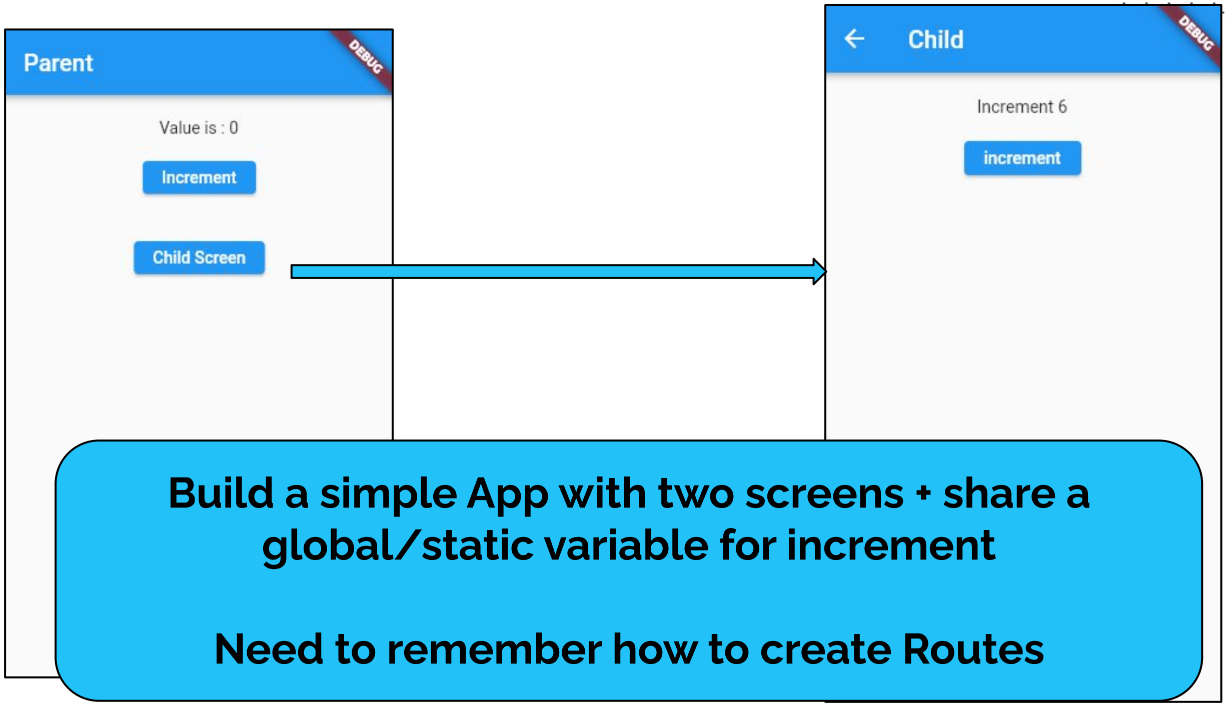
    that will be refreshed and changed.
    - The Widget is linked internally to state instance to build it on the screen.



    - There is no imperative changing of the UI itself (***like widget.setText***)—you change the state, and the UI rebuilds from scratch (Remember : of the relevant widget).
  - Stateless Widgets are used to build static components that will not change.

    But , They can include stateful widgets which can change.

State Management in Flutter

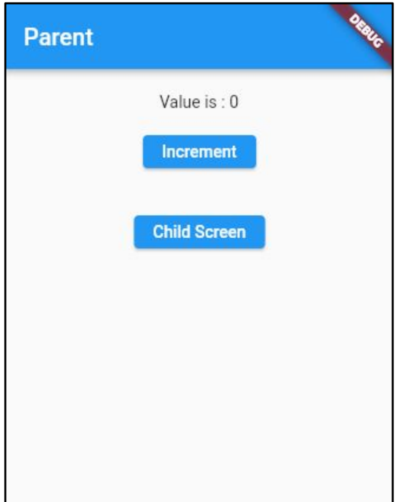


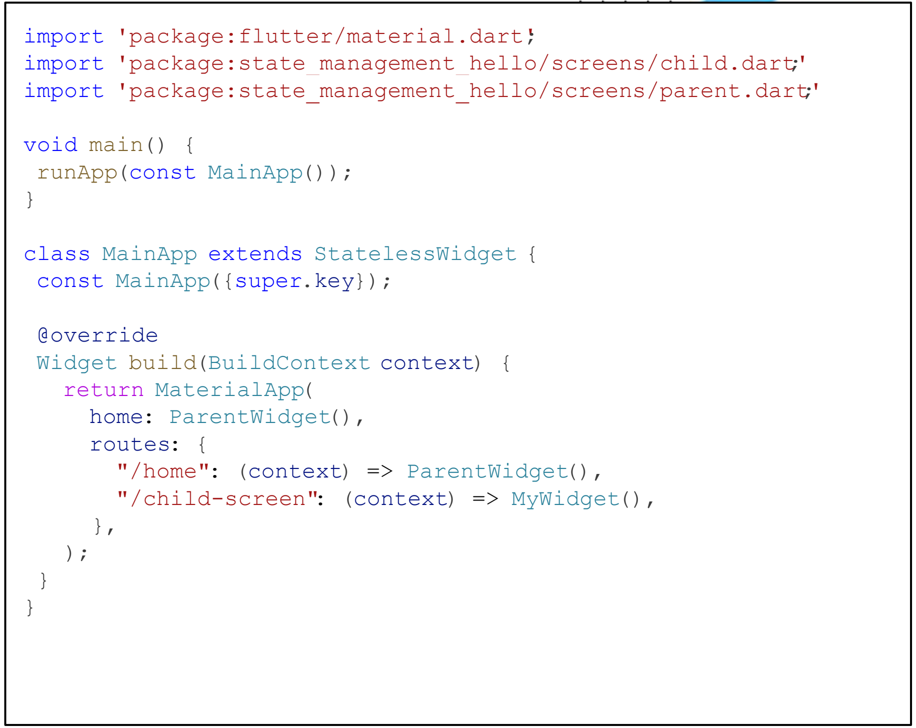
Build a simple App with two screens + share a global/static variable for increment

Need to remember how to create Routes

State Management in Flutter

- **Stateless & Stateful Widgets**

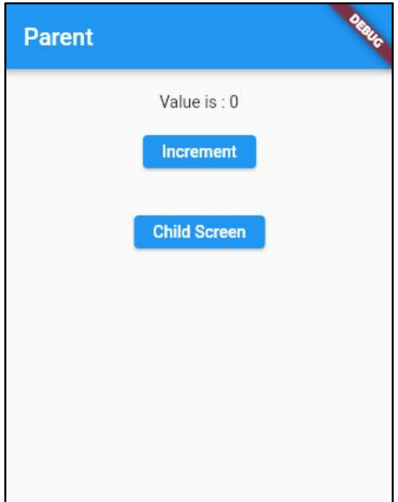


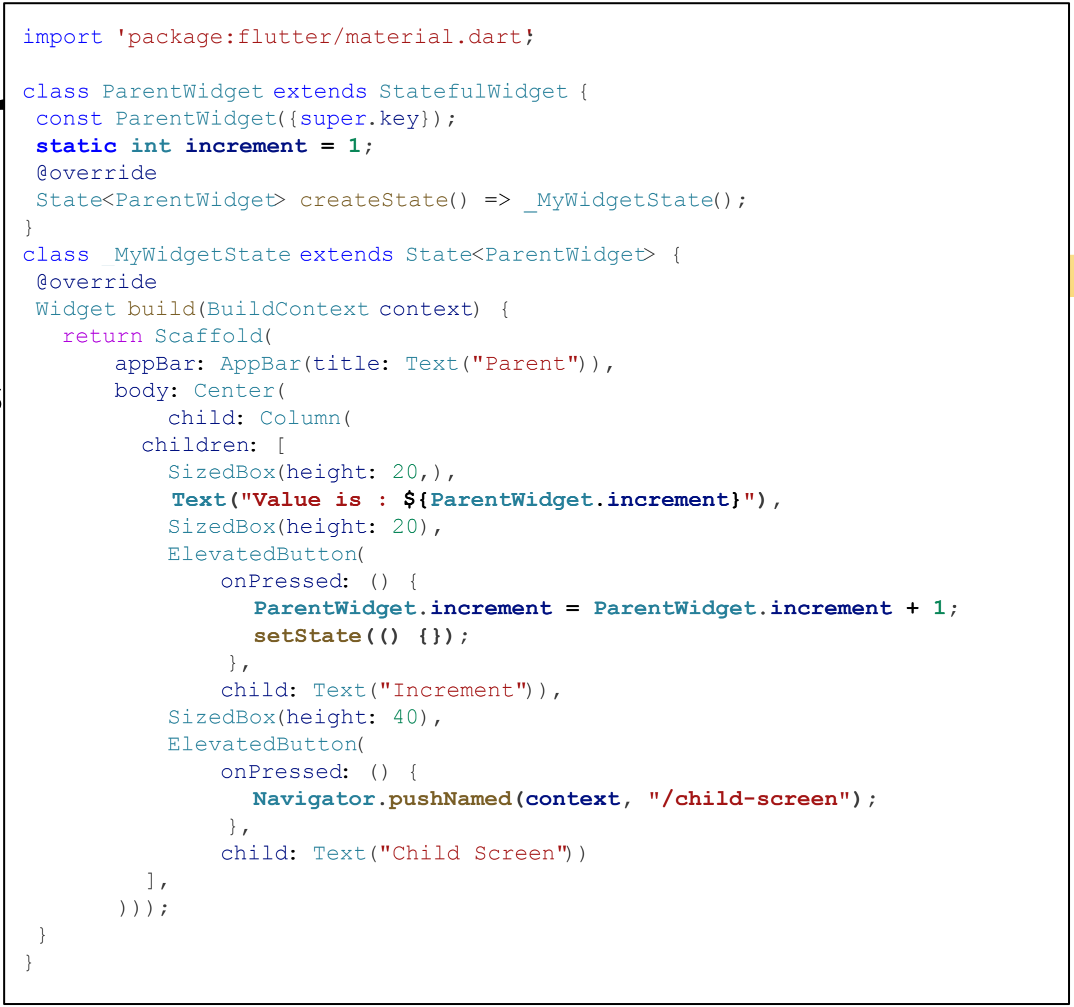


```
import 'package:flutter/material.dart';
import 'package:state_management_hello/screens/child.dart;'
import 'package:state_management_hello/screens/parent.dart;'

void main() {
 runApp(const MainApp());
}

class MainApp extends StatelessWidget {
 const MainApp({super.key});

 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     home: ParentWidget(),
     routes: {
       "/home": (context) => ParentWidget(),
       "/child-screen": (context) => MyWidget(),
     },
   );
 }
}
```

State Managem
Flutter

- **Stateless & Stateful Widgets**

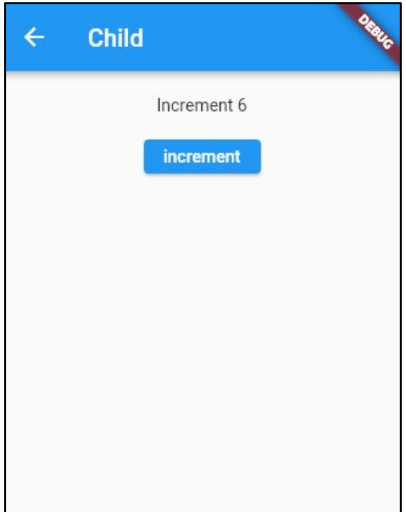


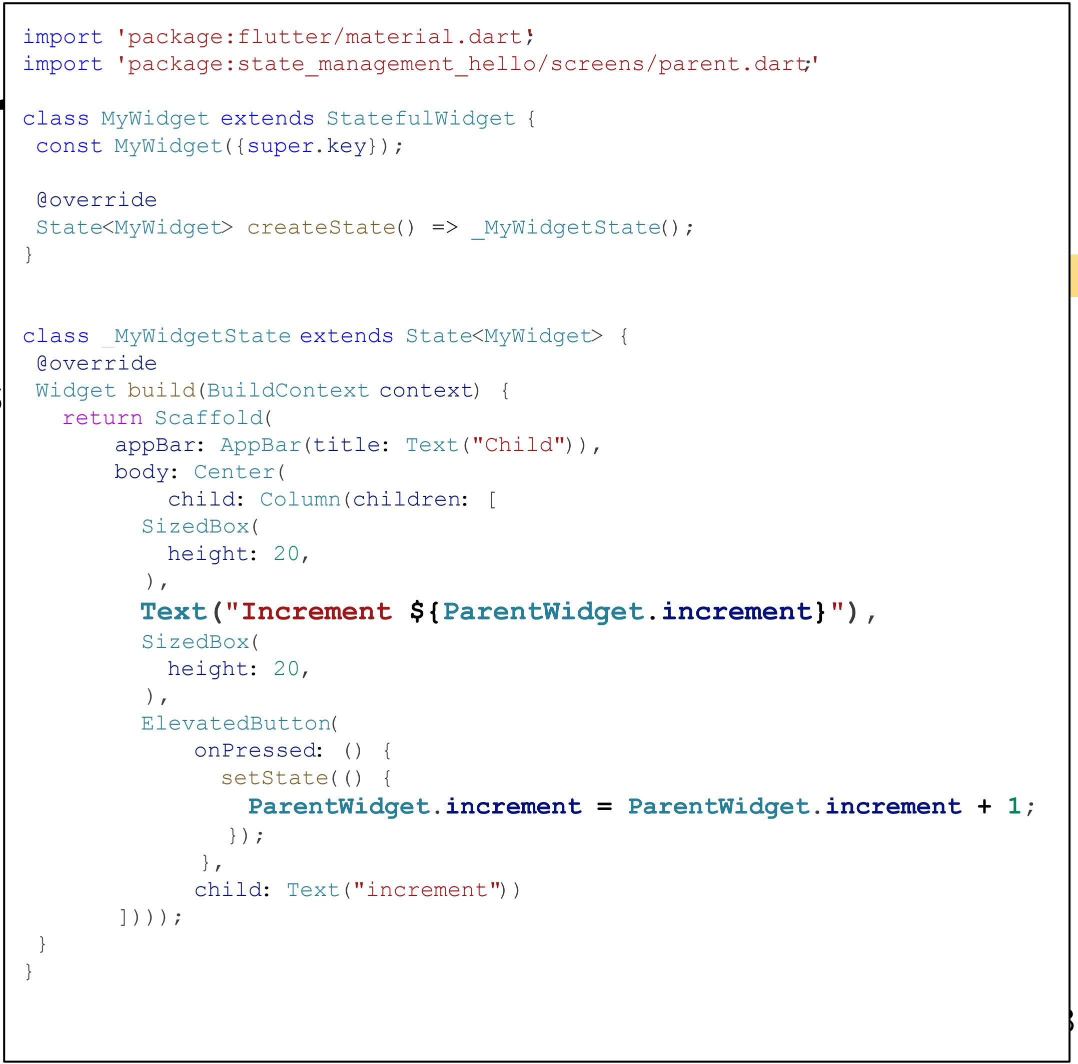


```
import 'package:flutter/material.dart';

class ParentWidget extends StatefulWidget {
 const ParentWidget({super.key});
 static int increment = 1;
 @override
 State<ParentWidget> createState() => _MyWidgetState();
}
class _MyWidgetState extends State<ParentWidget> {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
       appBar: AppBar(title: Text("Parent")),
       body: Center(
           child: Column(
         children: [
           SizedBox(height: 20,),
           Text("Value is : ${ParentWidget.increment}"),
           SizedBox(height: 20),
           ElevatedButton(
               onPressed: () {
                 ParentWidget.increment = ParentWidget.increment + 1;
                 setState(() {});
               },
               child: Text("Increment")),
           SizedBox(height: 40),
           ElevatedButton(
               onPressed: () {
                 Navigator.pushNamed(context, "/child-screen");
               },
               child: Text("Child Screen"))
         ],
       )));
 }
}
```

State Managem
Flutter

- **Stateless & Stateful Widgets**

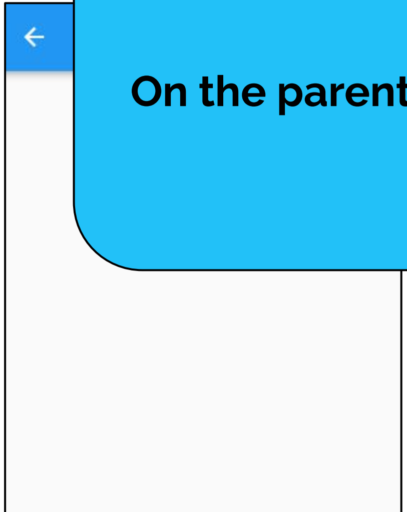


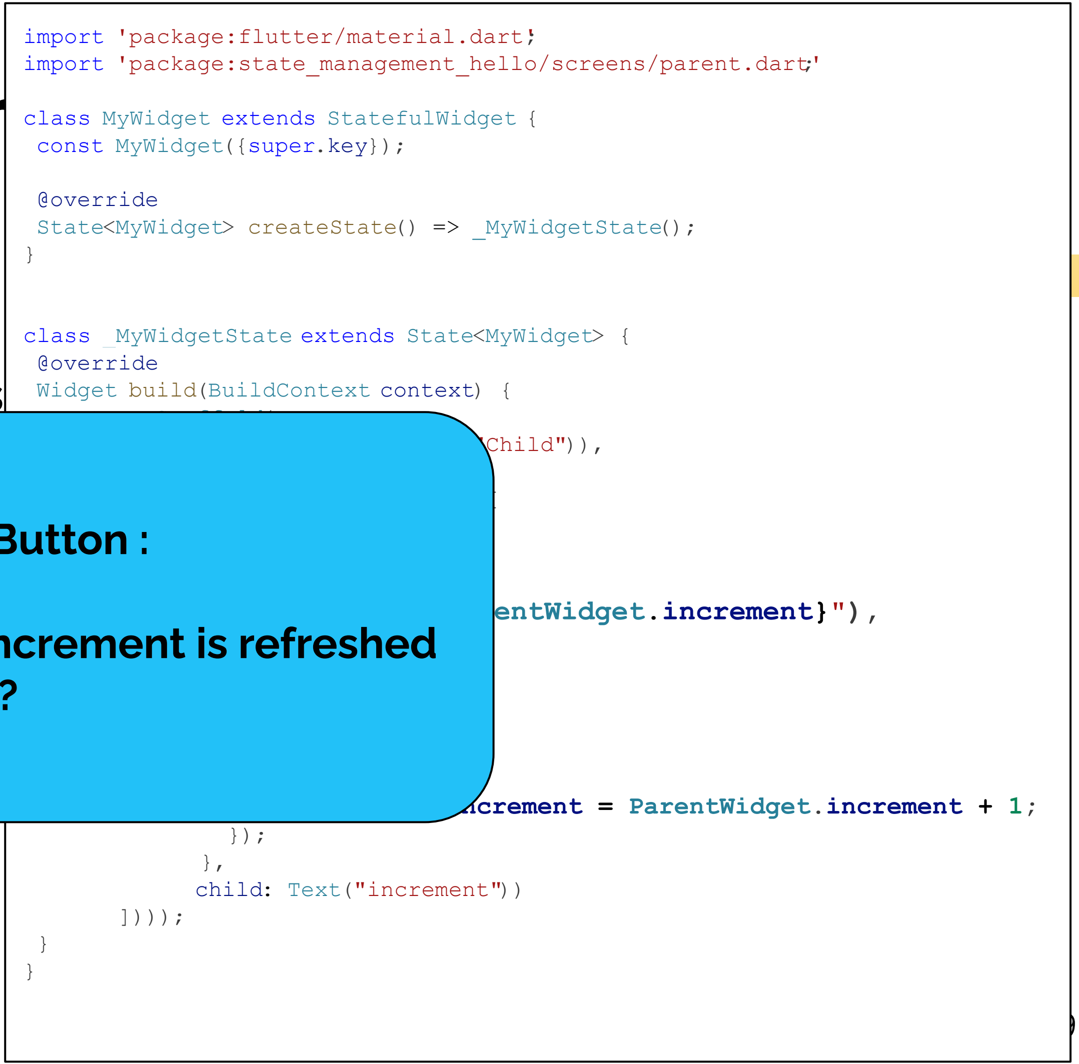


```
import 'package:flutter/material.dart';
import 'package:state_management_hello/screens/parent.dart;'

class MyWidget extends StatefulWidget {
 const MyWidget({super.key});

 @override
 State<MyWidget> createState() => _MyWidgetState();
}


class _MyWidgetState extends State<MyWidget> {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
       appBar: AppBar(title: Text("Child")),
       body: Center(
           child: Column(children: [
         SizedBox(
           height: 20,
         ),
         Text("Increment ${ParentWidget.increment}"),
         SizedBox(
           height: 20,
         ),
         ElevatedButton(
             onPressed: () {
               setState(() {
                 ParentWidget.increment = ParentWidget.increment + 1;
               });
             },
             child: Text("increment"))
       ])));
 }
}
```

State Managem
Flutter

- **Stateless & Stateful Widgets**





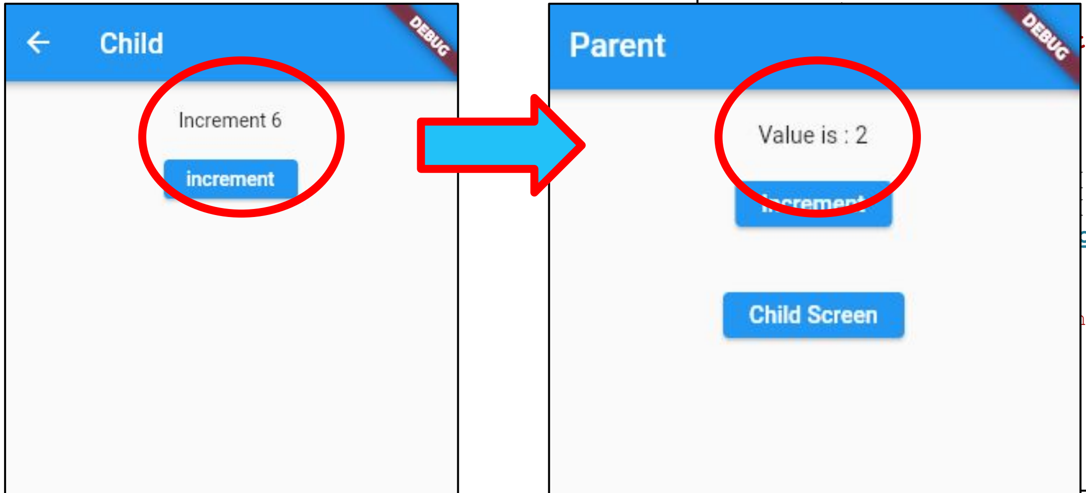
```
import 'package:flutter/material.dart';
import 'package:state_management_hello/screens/parent.dart;'

class MyWidget extends StatefulWidget {
 const MyWidget({super.key});

 @override
 State<MyWidget> createState() => _MyWidgetState();
}


class _MyWidgetState extends State<MyWidget> {
 @override
 Widget build(BuildContext context) {
                          Child")),

                          entWidget.increment}"),

                          crement = ParentWidget.increment + 1;
              });
            },
            child: Text("increment"))
       ])));
 }
}
```

Hit the Back Button :

On the parent Screen, the increment is refreshed or Not ?

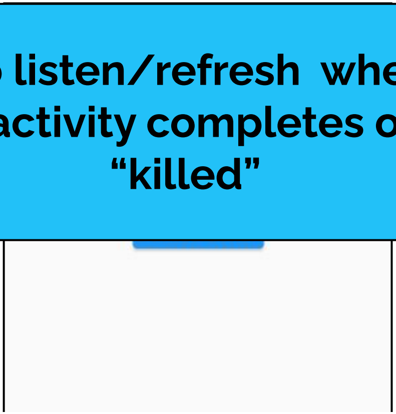
No, it is not updated .
Build method is not called for the case of “onResume”...

```
import 'package:flutter/material.dart';
import 'package:state_management_hello/screens/parent.dart';

                                     atefulWidget {
                                     y});

                                     ate() => _MyWidgetState();

                                     nds State<MyWidget> {

                                     t context) {

                                     le: Text("Child")),
      body: Center(
        child: Column(children: [
      SizedBox(
        height: 20,
                                     : ${ParentWidget.increment}"),

                                     get.increment = ParentWidget.increment + 1;

                                     ncrement"))
```

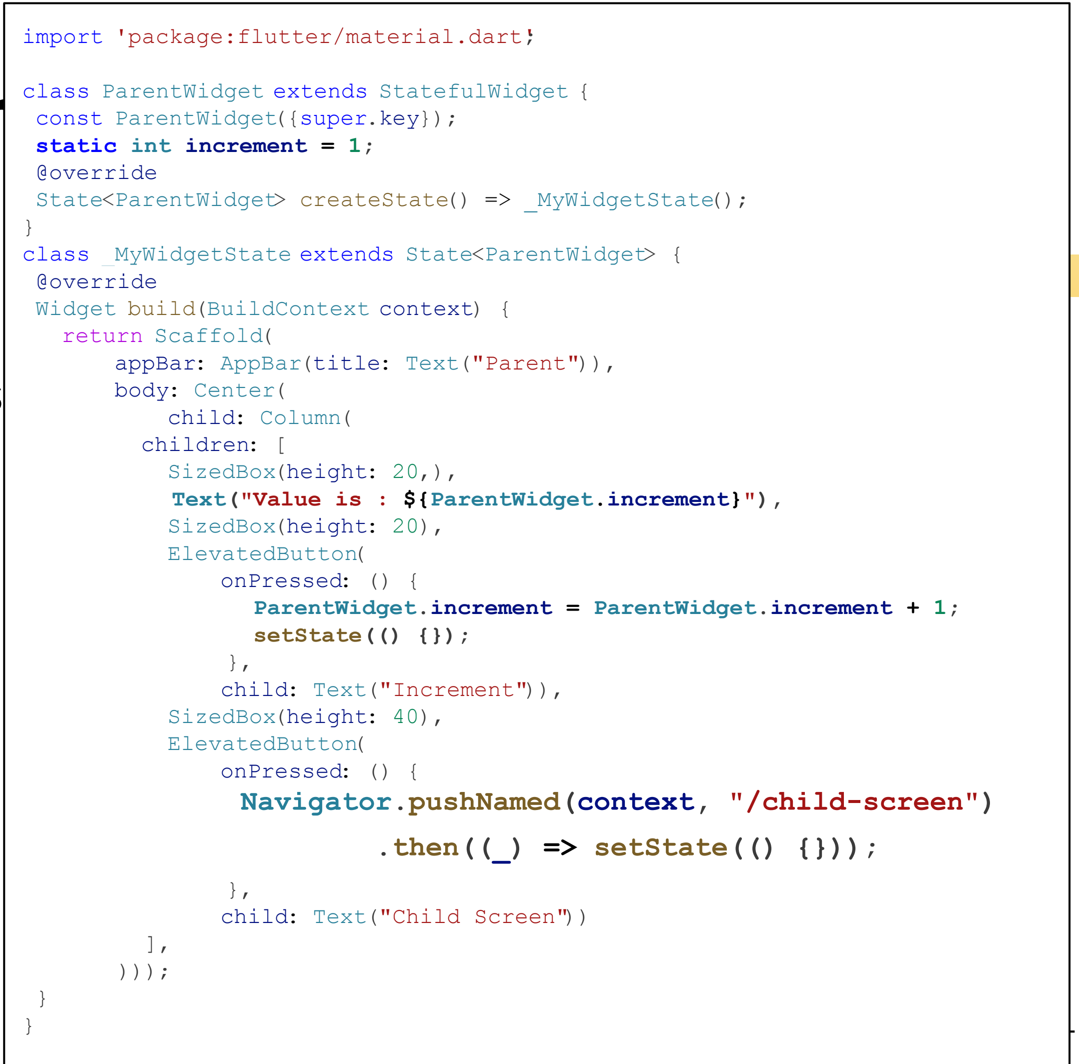


State Managem Flutter

- **Stateless & Stateful Widgets**



How to listen/refresh when the child activity completes or it is "killed"



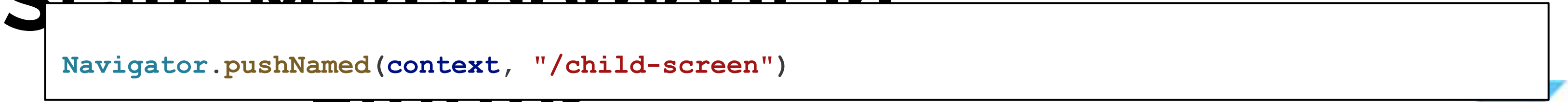
```
import 'package:flutter/material.dart';

class ParentWidget extends StatefulWidget {
 const ParentWidget({super.key});
 static int increment = 1;
 @override
 State<ParentWidget> createState() => _MyWidgetState();
}
class _MyWidgetState extends State<ParentWidget> {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
       appBar: AppBar(title: Text("Parent")),
       body: Center(
           child: Column(
         children: [
           SizedBox(height: 20,),
           Text("Value is : ${ParentWidget.increment}"),
           SizedBox(height: 20),
           ElevatedButton(
               onPressed: () {
                 ParentWidget.increment = ParentWidget.increment + 1;
                 setState(() {});
               },
               child: Text("Increment")),
           SizedBox(height: 40),
           ElevatedButton(
               onPressed: () {
                Navigator.pushNamed(context, "/child-screen")
                          .then((_) => setState(() {}));
               },
               child: Text("Child Screen"))
         ],
       )));
 }
}
```

then function : called on future objects. Its block will be executed when the future is returned/assigned

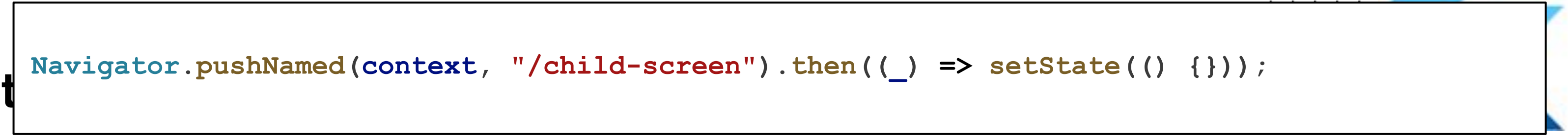
_ : a parameter not bothering to give it a name because it will not be used.

State Management in Flutter

- St

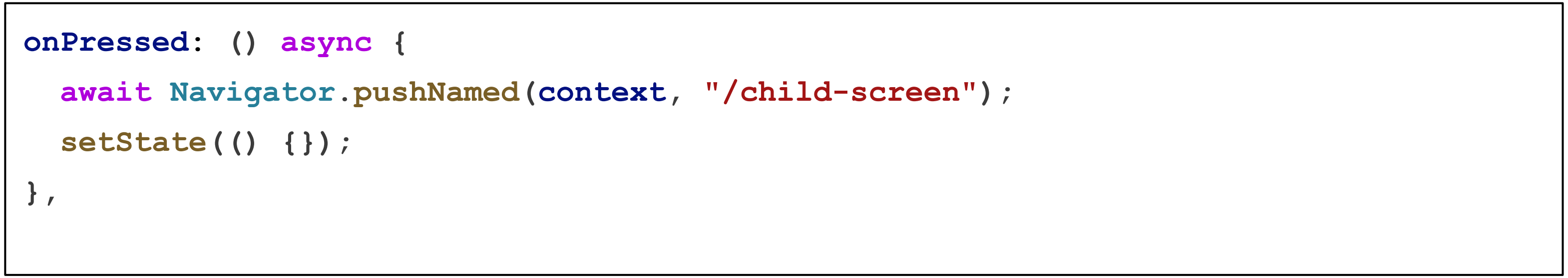


```
Navigator.pushNamed(context, "/child-screen")
```



```
Navigator.pushNamed(context, "/child-screen").then((_) => setState(() {}));
```

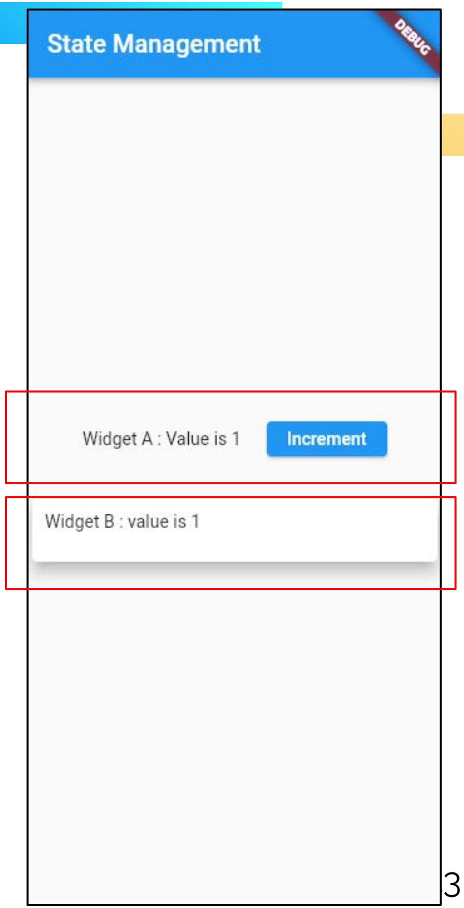
Or use async and await



```
onPressed: () async {
  await Navigator.pushNamed(context, "/child-screen");
  setState(() {});
},
```

State Management in Flutter

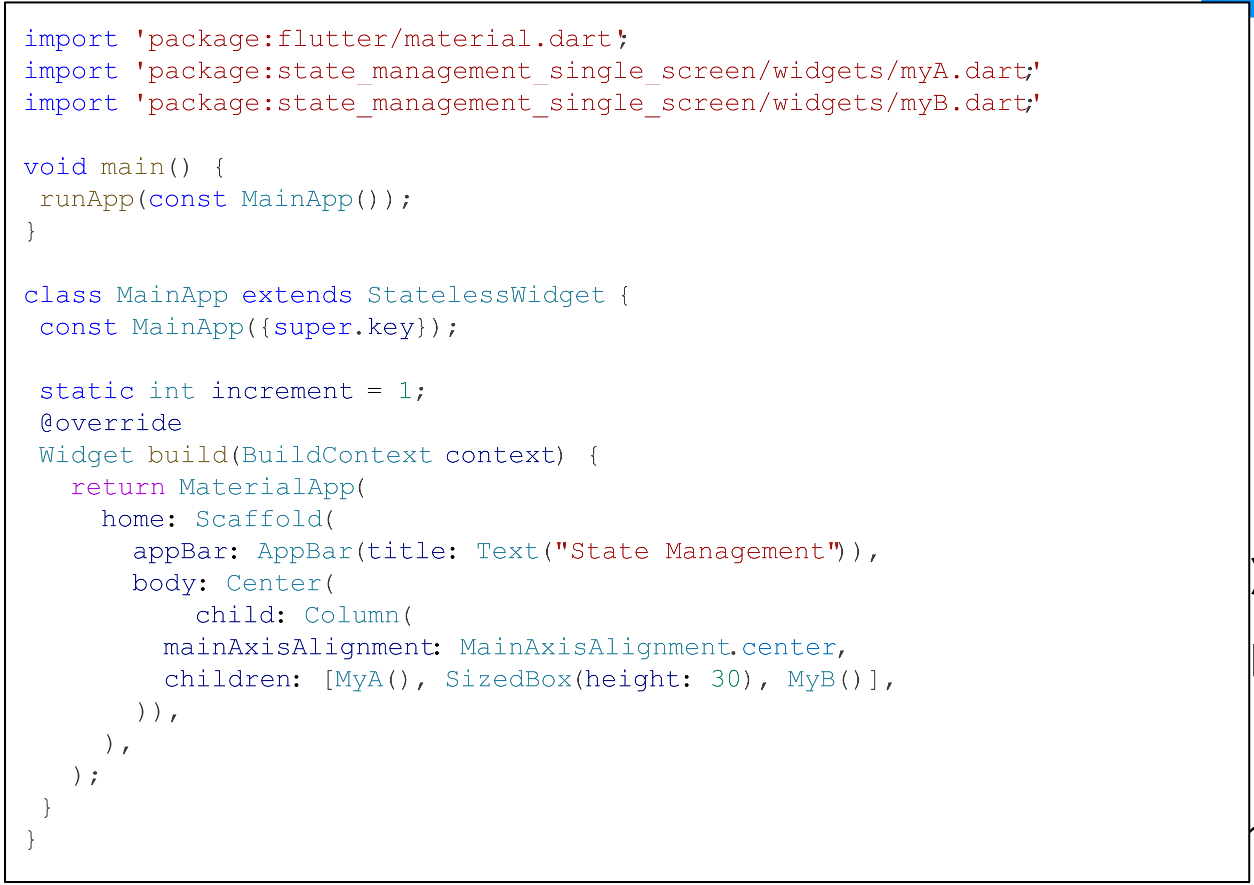
- **Stateless & Stateful Widgets**
  - Single Stateless Screen with :
    - Static/Global increment Variable
    - Stateful : Widget A
      - Showing Increment Value (Global var)
      - Button to increment the increment value
    - Stateful : Widget B
      - Show the increment value (Global var)



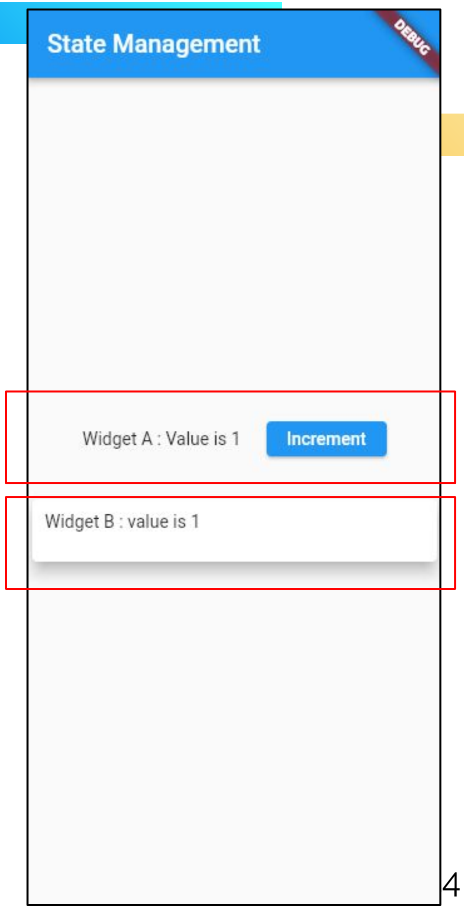
Widget A : Value is 1

Widget B : value is 1

State Management in



```
import 'package:flutter/material.dart';
import 'package:state_management_single_screen/widgets/myA.dart;'
import 'package:state_management_single_screen/widgets/myB.dart;'

void main()  {
 runApp(const MainApp());
}

class MainApp extends StatelessWidget {
 const MainApp({super.key});

 static int increment = 1;
 @override
 Widget build(BuildContext context)  {
   return MaterialApp(
     home: Scaffold(
       appBar: AppBar(title: Text("State Management")),
       body: Center(
           child: Column(
         mainAxisAlignment: MainAxisAlignment.center,
         children: [MyA(), SizedBox(height: 30), MyB()],
       )),
     ),
   );
 }
}
```

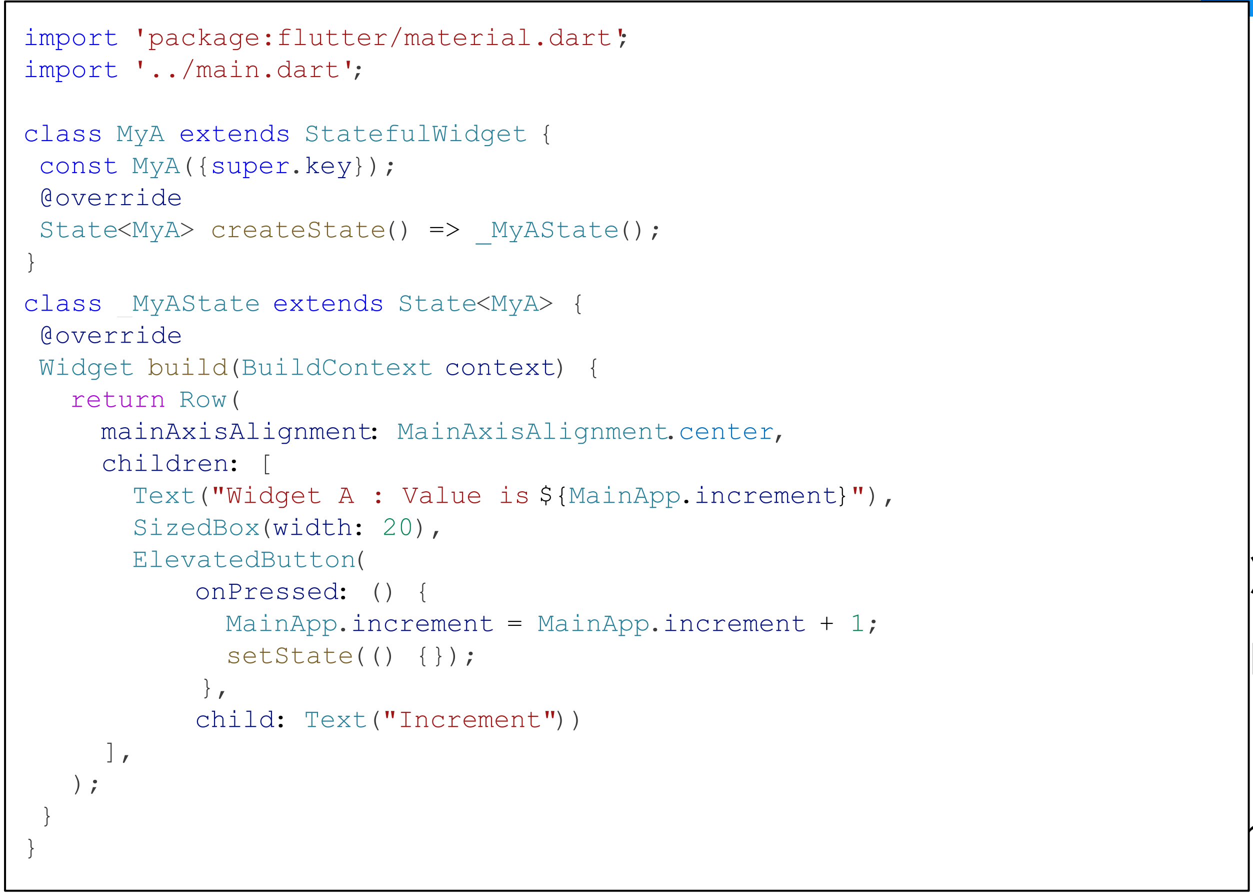


)

lue

r)

State Management in

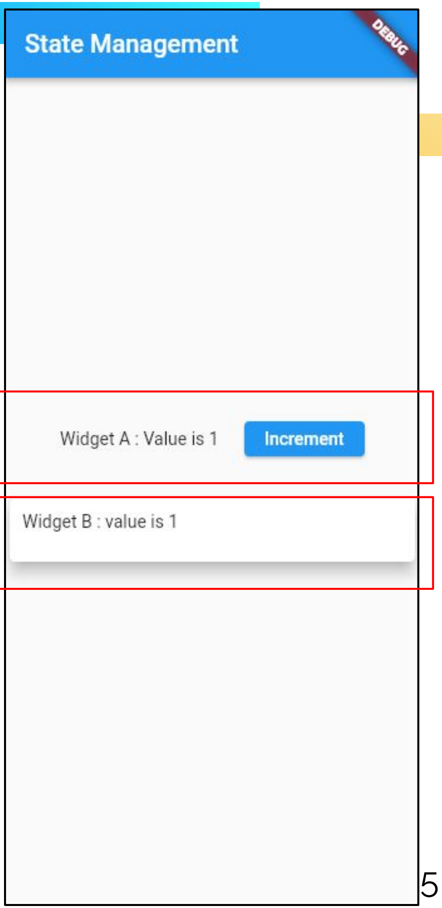


```
import 'package:flutter/material.dart';
import '../main.dart';

class MyA extends StatefulWidget {
 const MyA({super.key});
 @override
 State<MyA> createState() => _MyAState();
}

class _MyAState extends State<MyA> {
 @override
 Widget build(BuildContext context) {
   return Row(
     mainAxisAlignment: MainAxisAlignment.center,
     children: [
       Text("Widget A : Value is ${MainApp.increment}"),
       SizedBox(width: 20),
       ElevatedButton(
           onPressed: () {
             MainApp.increment = MainApp.increment + 1;
             setState(() {});
           },
           child: Text("Increment"))
     ],
   );
 }
}
```

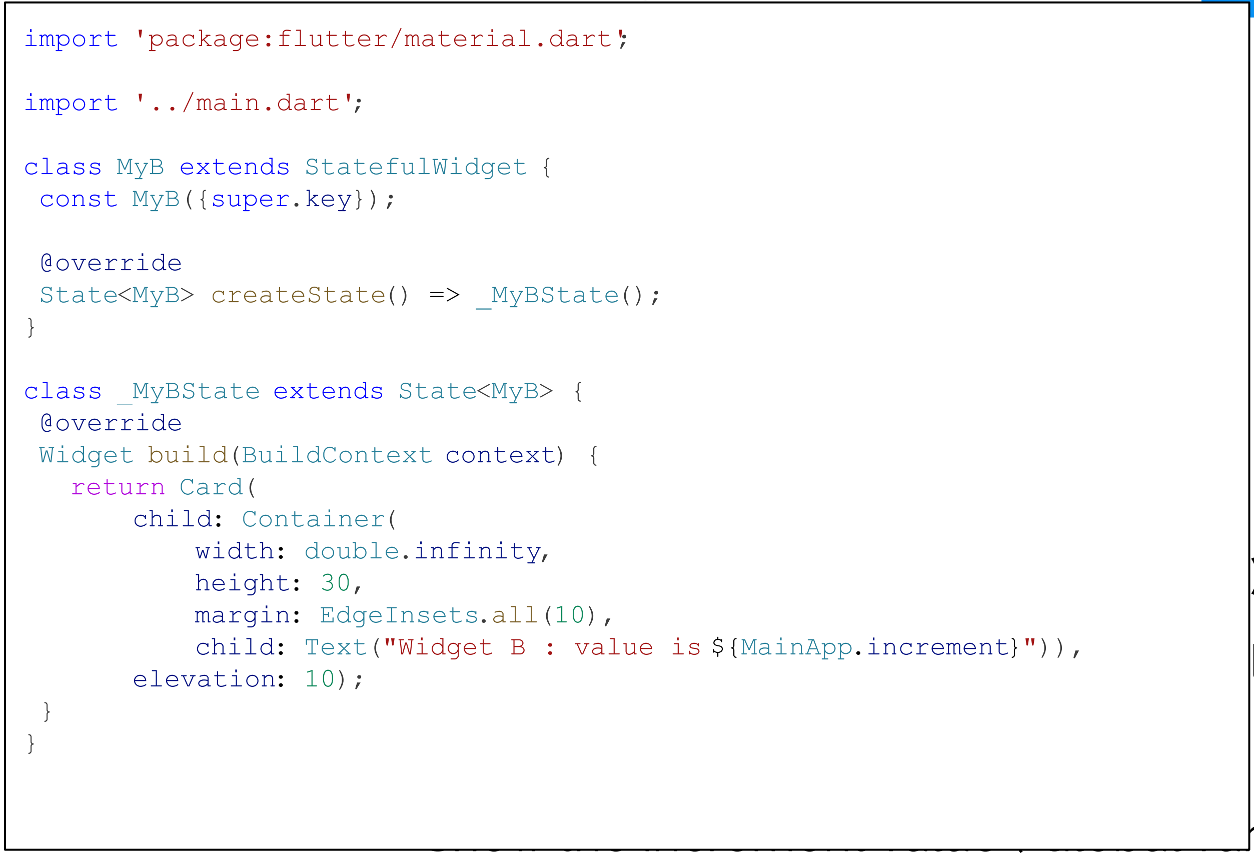
)

lue

r)



State Management in

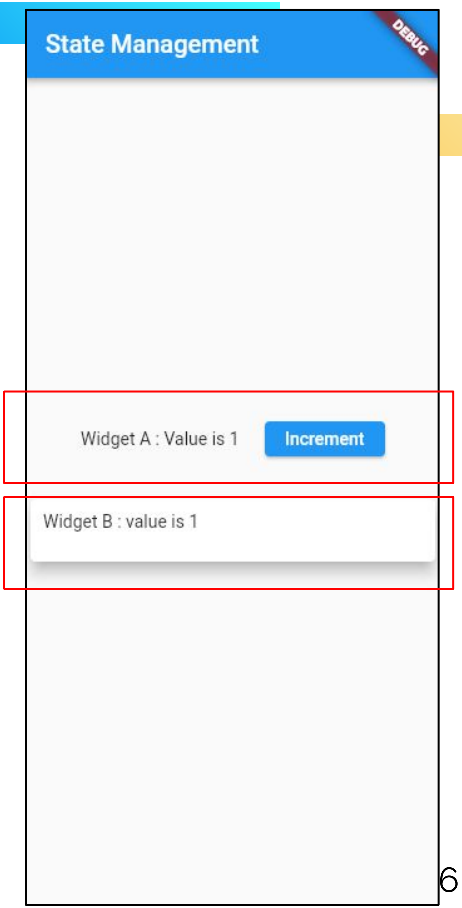


```
import 'package:flutter/material.dart';

import '../main.dart';

class MyB extends StatefulWidget {
 const MyB({super.key});

 @override
 State<MyB> createState() => _MyBState();
}

class _MyBState extends State<MyB> {
 @override
 Widget build(BuildContext context) {
   return Card(
       child: Container(
           width: double.infinity,
           height: 30,
           margin: EdgeInsets.all(10),
           child: Text("Widget B : value is ${MainApp.increment}")),
       elevation: 10);
 }
}
```

)

lue

r)



Widget A : Value is 1

Widget B : value is 1

State Management in Flutter

- **Stateless & Stateful Widgets**

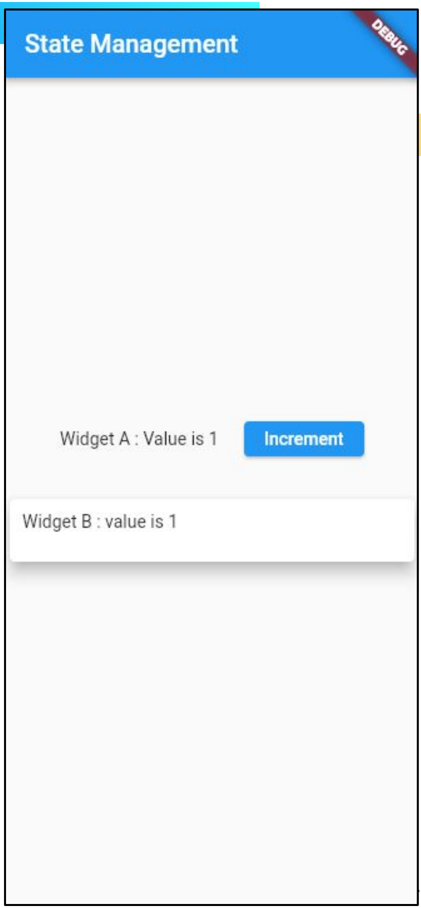
  ○

Will the second widget get updated too ?

- Button to increment the increment value

  ■ Stateful : Widget B

  - Show the increment value (Global var)



State Management

Widget A : Value is 1

Widget B : value is 1

State Management in Flutter

- **Stateless & Stateful Widgets**

No, calling the setState(()(}) will only update the calling widget, but not other Widgets.

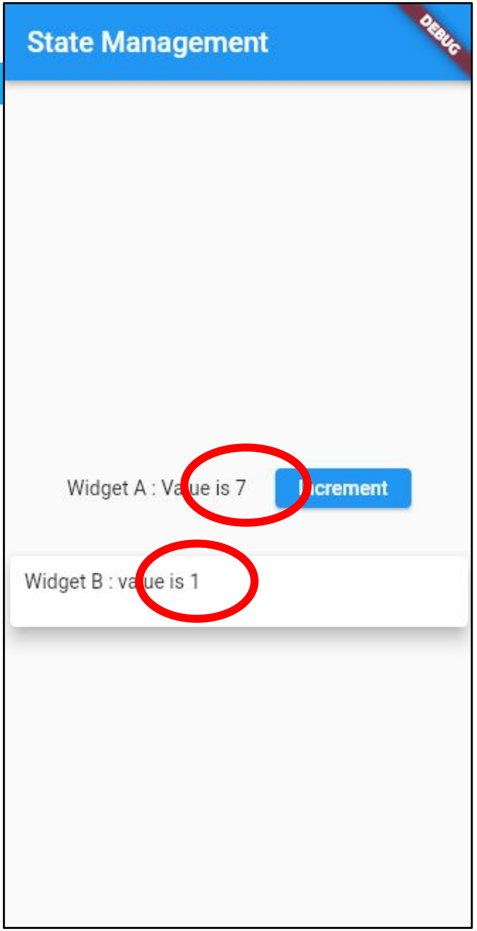
Any Solution ?

Don't think to put them into a single Widget.

- Button to increment the increment value

▪ Stateful : Widget B

- Show the increment value (Global var)



State Management in Flutter

- **State Management**
    - You have data (Variables or any data structure) that is usually used to draw the UI using Flutter.
        - State Management Techniques are used to ensure that data is fully synced with the UI.
            - If there is a change in the data, the UI is **automatically updated.** and vice versa.

State Management in Flutter

- **State Management Libraries**
  - setState (Low-Level)
  - Provider
  - GetX
  - Redux
  - BloC
  - …

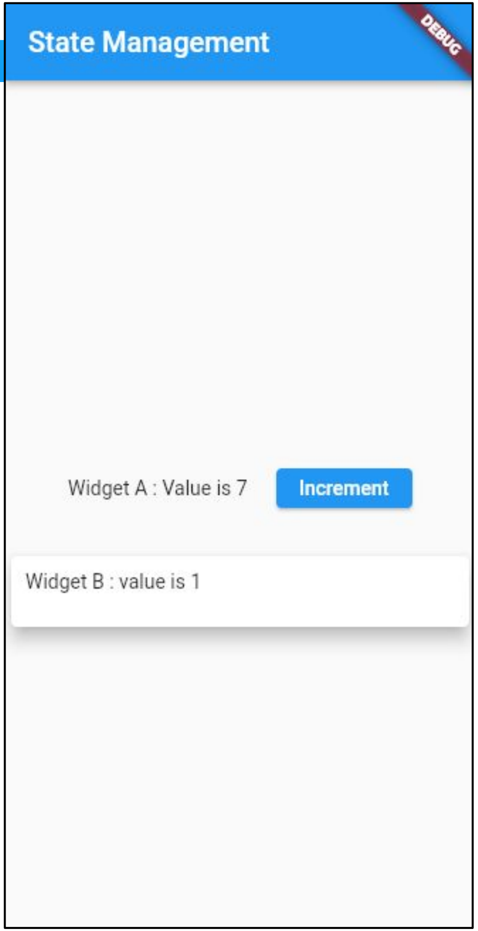
Which better one to use ?

State Management in Flutter

- **GetX**
  - Claiming to be the most popular library within the flutter pub.dev
  - Set of libraries offering various functionalities including:
    - State Management
    - Routing and Navigation
    - …
  - Why GetX : Performance + Coding Productivity + Decoupling

State Management in Flutter

- **GetX**
    - How to use GetX state management to update multiple widgets within the same screen ?



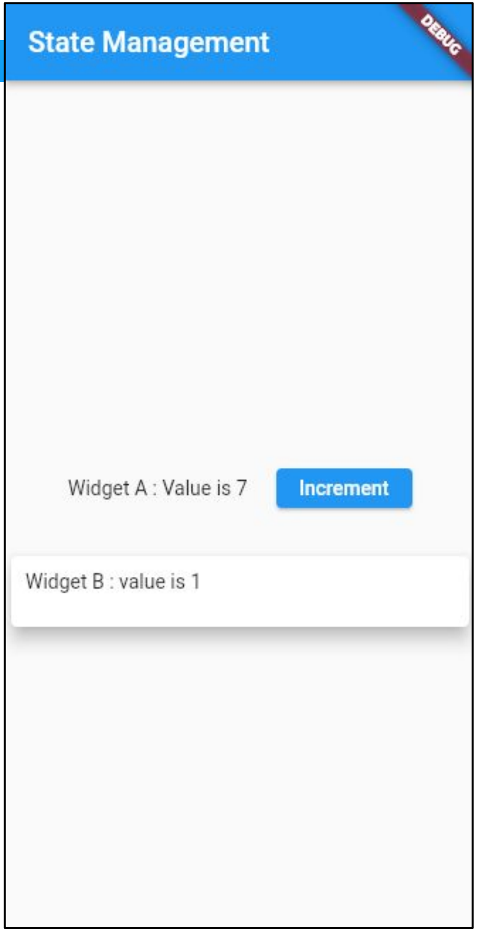
Widget A : Value is 7

Widget B : value is 1

State Management in Flutter

- **GetX**
  - From the command line, import the get Package

    flutter pub add get
  - Or, you can edit the pubspec.yaml file to add the relevant dependency



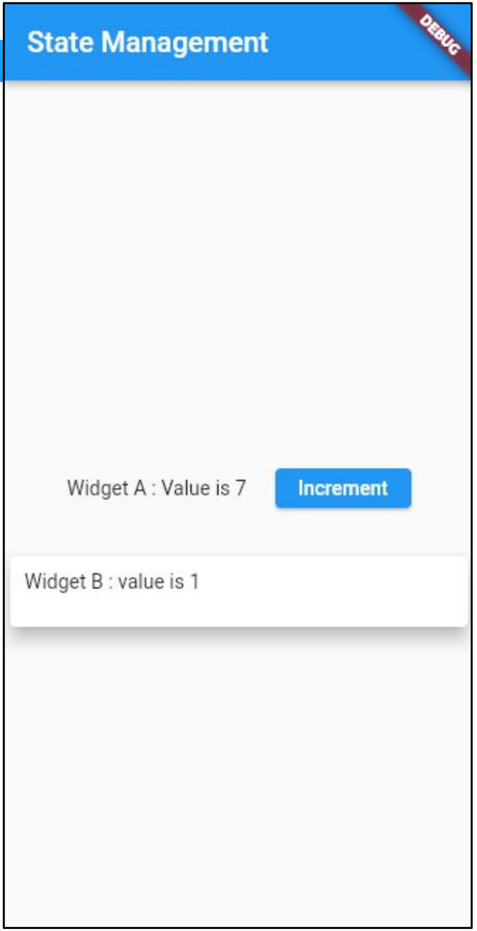
State Management

Widget A : Value is 7

Widget B : value is 1

State Management in Flutter

```dart
import 'package:get/get.dart';

class IncrementController extends GetxController {
 final increment = 0.obs;

 @override
 void onInit() {
   super.onInit();
 }

 @override
 void onReady() {}

 @override
 void onClose() {}

 void doIncrement() {
   increment.value = increment.value + 1;
 }
}
```

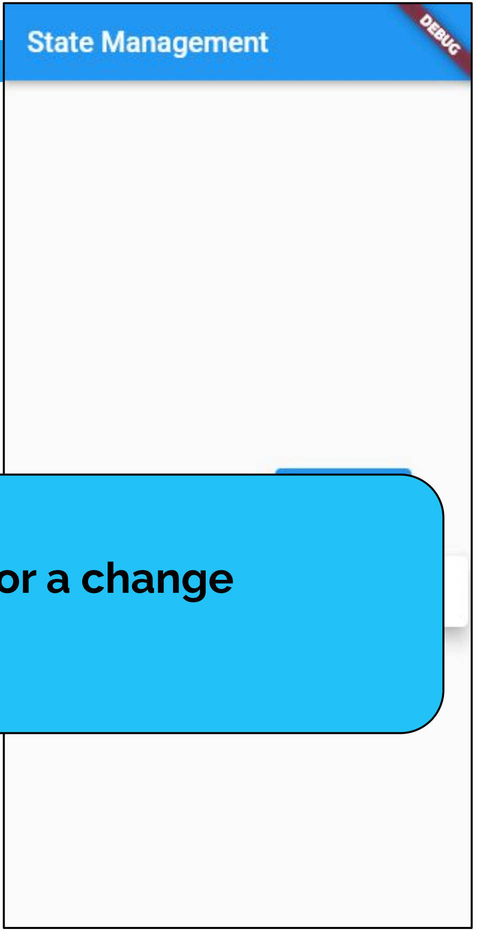


Widget A : Value is 7

Widget B : value is 1

State Management in Flutter

```
import 'package:get/get.dart';

class IncrementController extends GetxController {
 final increment = 0.obs;

 @override
 void onInit() {
   super.onInit();
 }

 @override
 void onReady() {}

 @override
 void onClose() {}

 void doIncrement() {
   increment.value = increment.value + 1;
 }
}
```

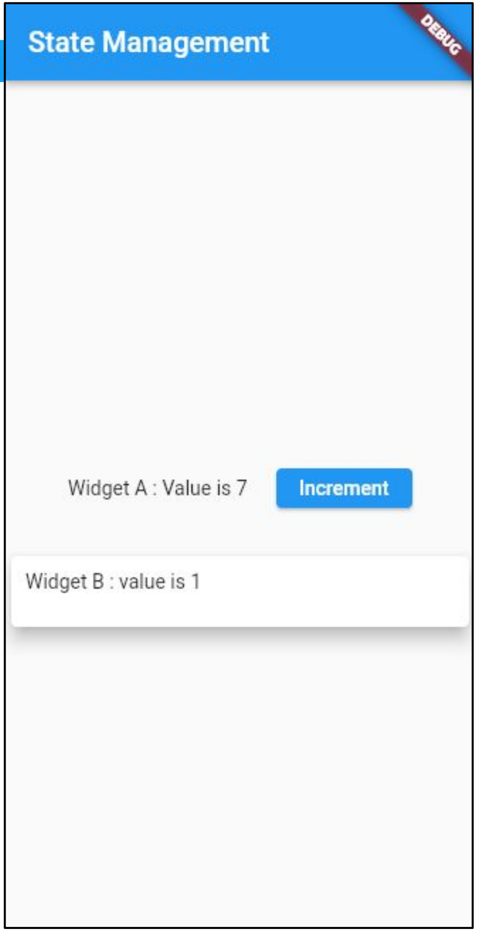


Obs : To be observed for a change

State Management in Flutter

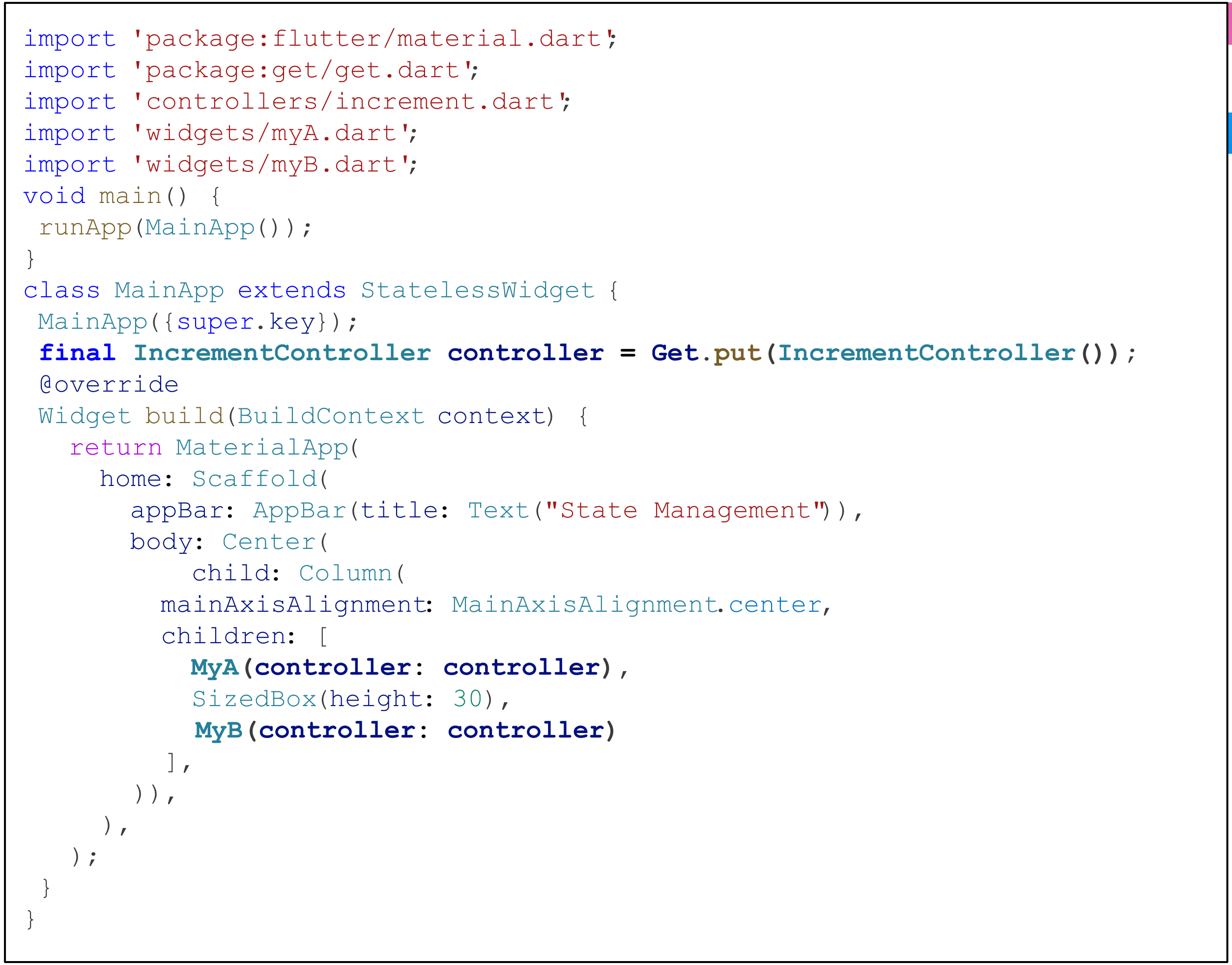
```
final increment = 0.obs;
final name = ‘Hello’.obs;
final myList = [].obs;
final myMap = {‘name’:’Amel’}.obs;
```

```
import 'package:get/get.dart';

class MyListController extends GetxController {
 final myList = [].obs;

 void add(String name) {
   myList.add(name);
 }
}
```

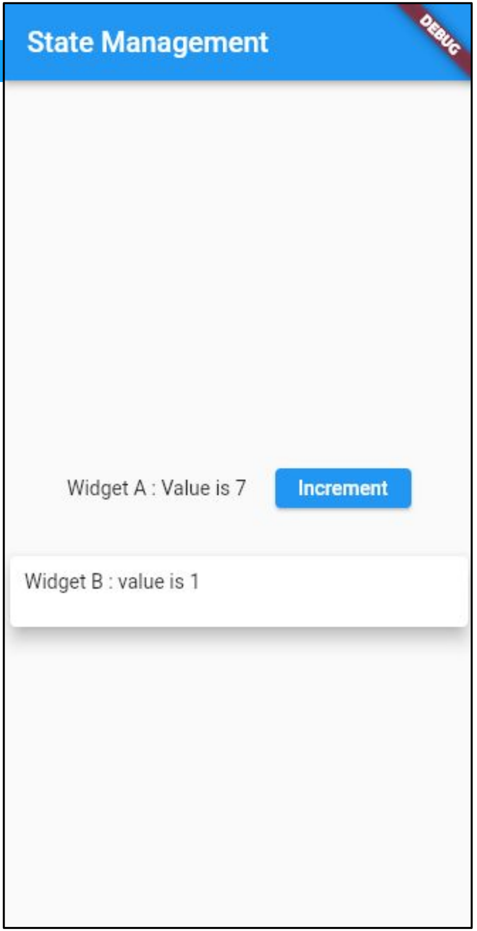


Widget A : Value is 7

Widget B : value is 1



```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'controllers/increment.dart';
import 'widgets/myA.dart';
import 'widgets/myB.dart';
void main() {
 runApp(MainApp());
}
class MainApp extends StatelessWidget {
 MainApp({super.key});
 final IncrementController controller = Get.put(IncrementController());
 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     home: Scaffold(
       appBar: AppBar(title: Text("State Management")),
       body: Center(
           child: Column(
         mainAxisAlignment: MainAxisAlignment.center,
         children: [
           MyA(controller: controller),
           SizedBox(height: 30),
           MyB(controller: controller)
         ],
       )),
     ),
   );
 }
}
```

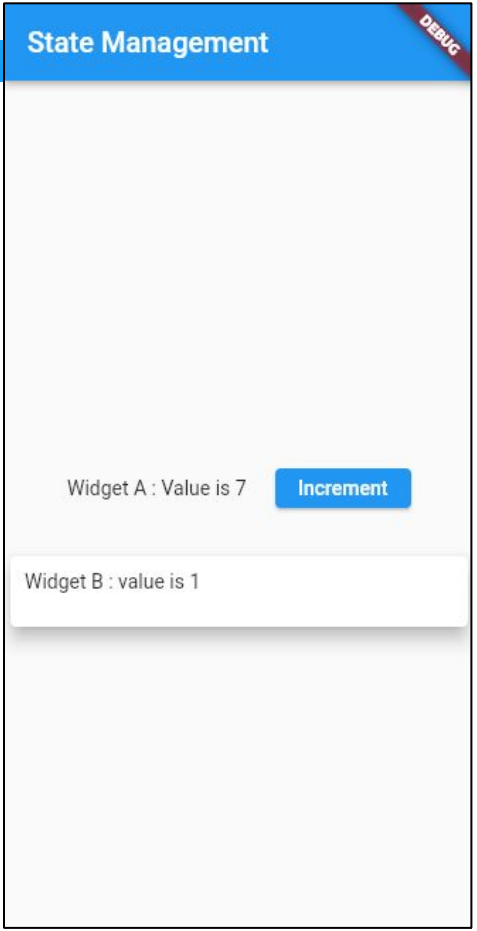


State Management

Widget A : Value is 7

Widget B : value is 1

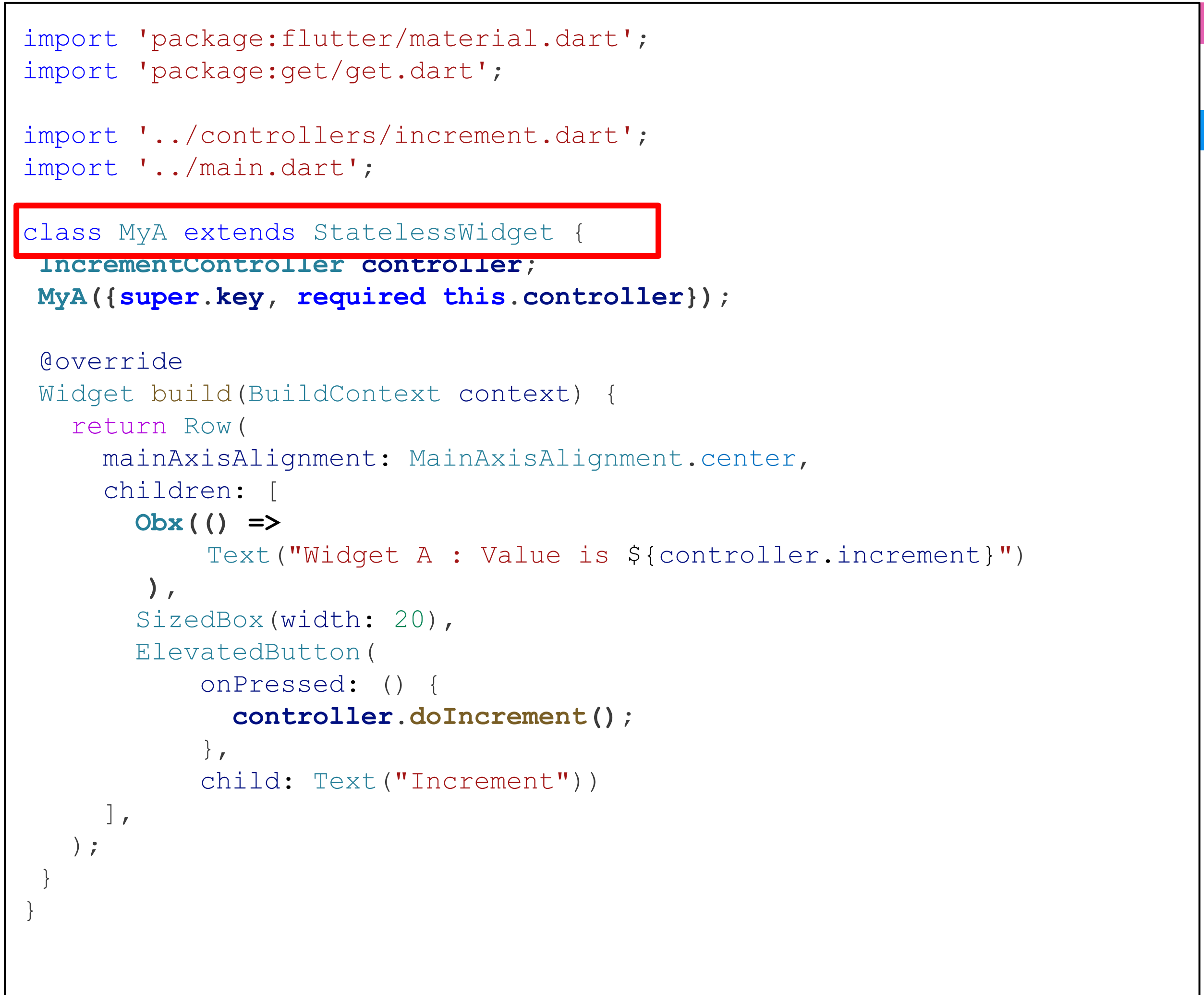
```
import 'package:flutter/material.dart';
import 'package:get/get.dart';

import '../controllers/increment.dart';
import '../main.dart';

class MyA extends StatelessWidget {
 IncrementController controller;
 MyA({super.key, required this.controller});

 @override
 Widget build(BuildContext context) {
   return Row(
     mainAxisAlignment: MainAxisAlignment.center,
     children: [
       Obx(() =>
           Text("Widget A : Value is ${controller.increment}")
        ),
       SizedBox(width: 20),
       ElevatedButton(
           onPressed: () {
             controller.doIncrement();
           },
           child: Text("Increment"))
     ],
   );
 }
}
```



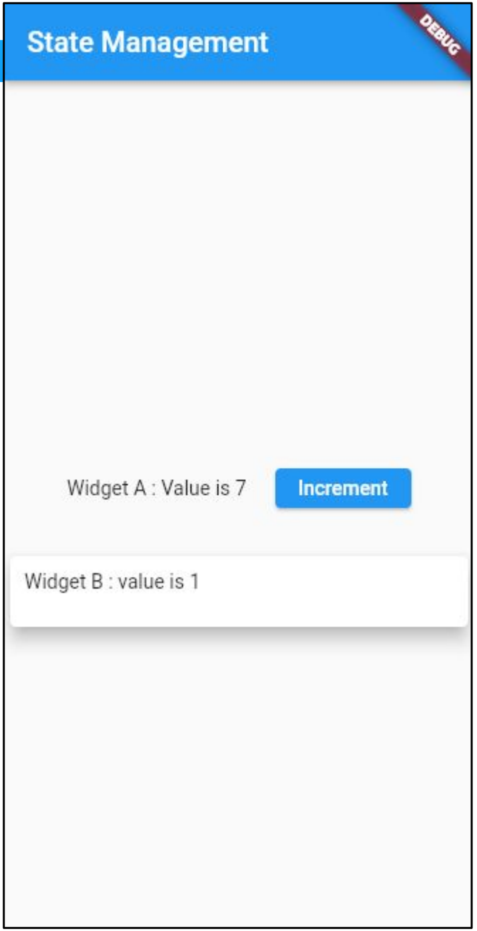
State Management

Widget A : Value is 7

Widget B : value is 1



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

import '../controllers/increment.dart';
import '../main.dart';

class MyA extends StatelessWidget {
 IncrementController controller;
 MyA({super.key, required this.controller});

 @override
 Widget build(BuildContext context) {
   return Row(
     mainAxisAlignment: MainAxisAlignment.center,
     children: [
       Obx(() =>
            Text("Widget A : Value is ${controller.increment}")
         ),
       SizedBox(width: 20),
       ElevatedButton(
           onPressed: () {
             controller.doIncrement();
           },
           child: Text("Increment"))
     ],
   );
 }
}
```

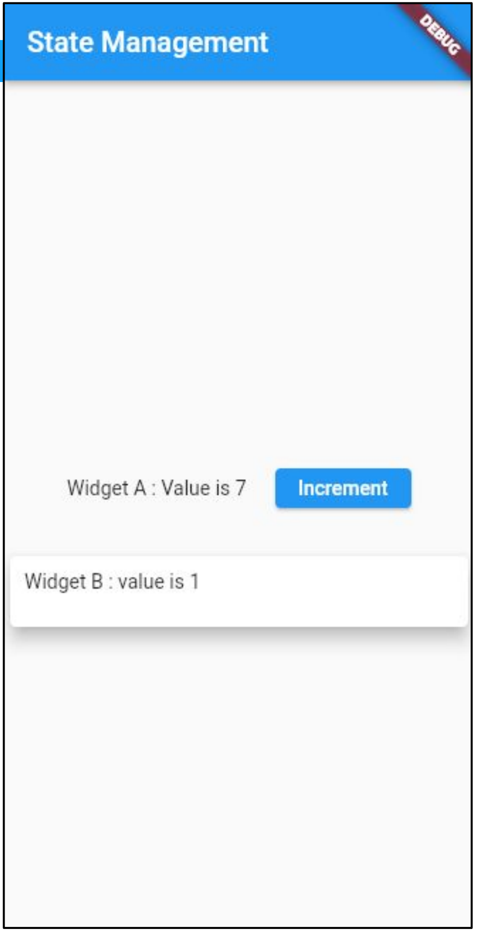


State Management

Widget A : Value is 7

Widget B : value is 1

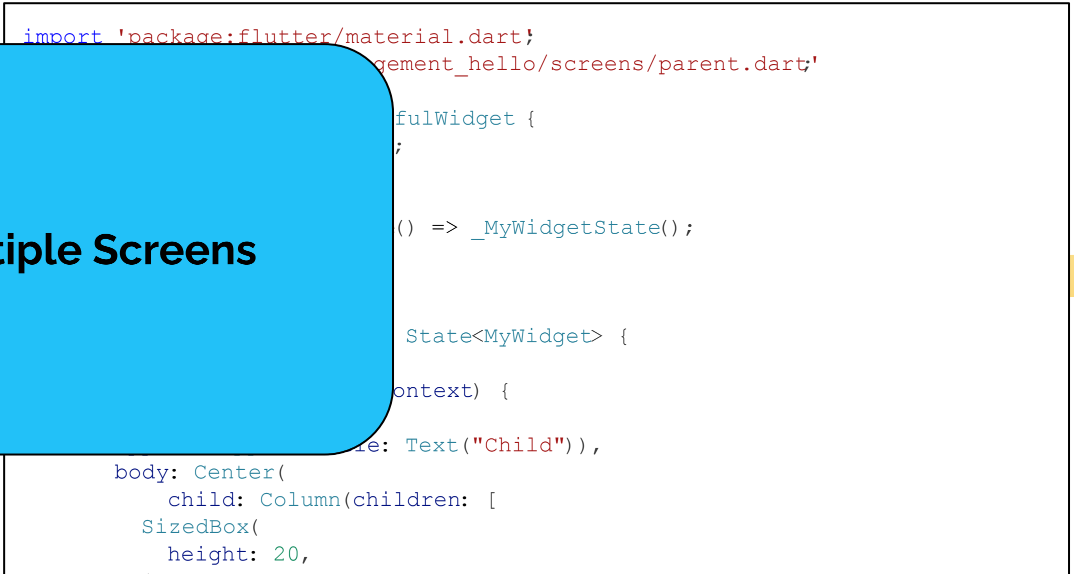
```
import 'package:flutter/material.dart';
import 'package:get/get.dart';

import '../controllers/increment.dart';
import '../main.dart';

class MyB extends StatelessWidget {
 IncrementController controller;
 MyB({super.key, required this.controller});

 @override
 Widget build(BuildContext context) {
   return Card(
       child: Container(
           width: double.infinity,
           height: 30,
           margin: EdgeInsets.all(10),
           child:
               Obx(() =>
                 Text("Widget B : value is ${controller.increment}"))
                 ),
       elevation: 10);
 }
}
```



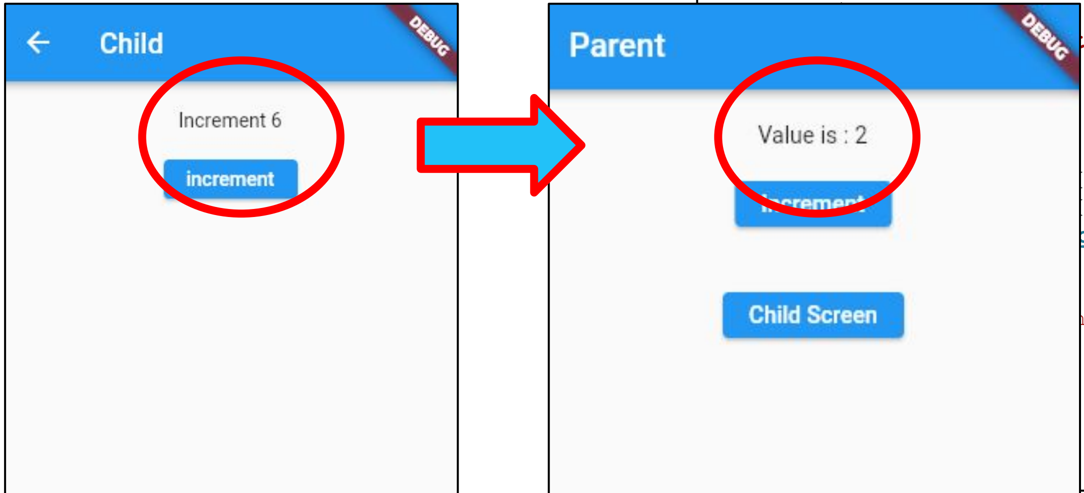
Widget A : Value is 7

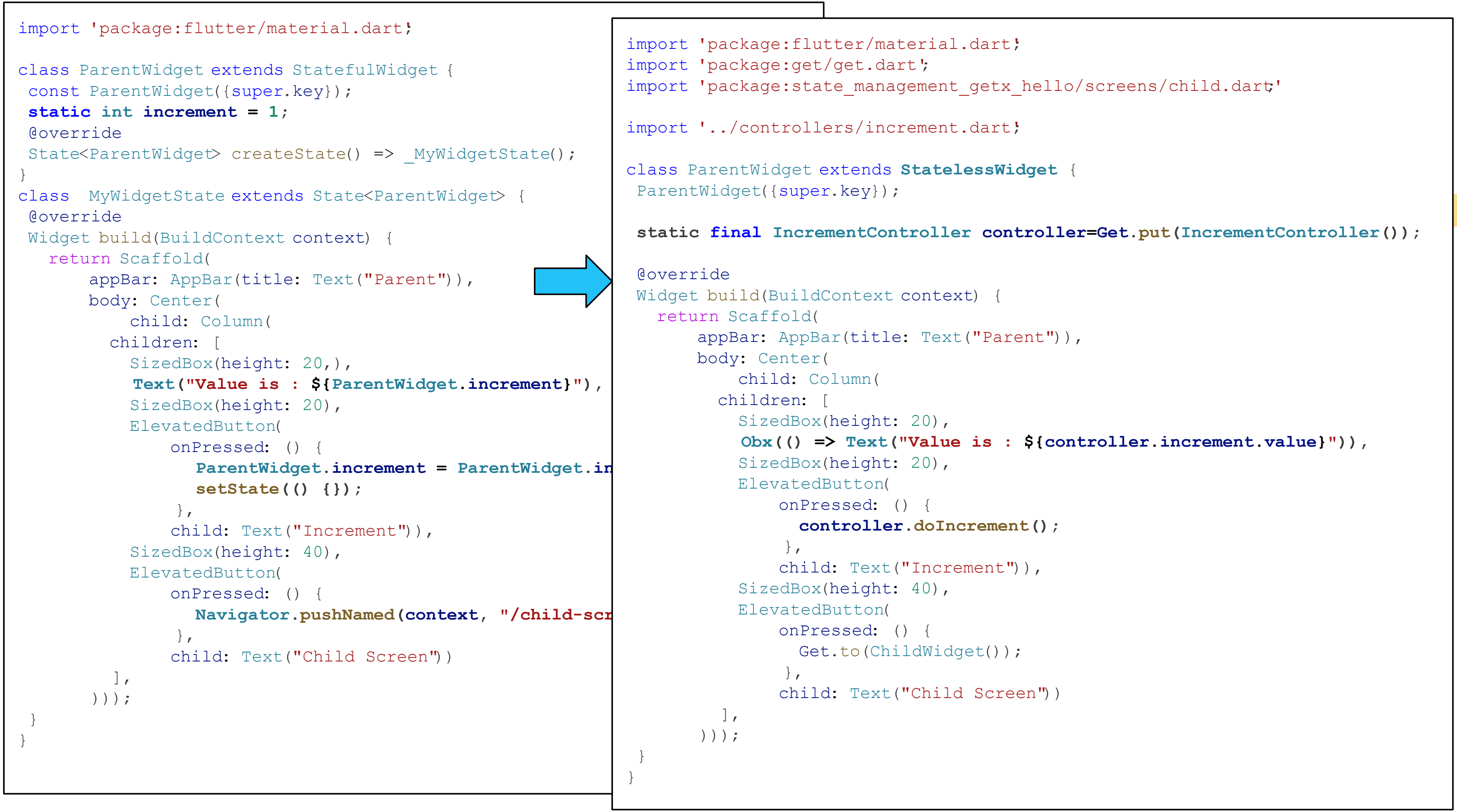
Widget B : value is 1

How to use GetX across Multiple Screens



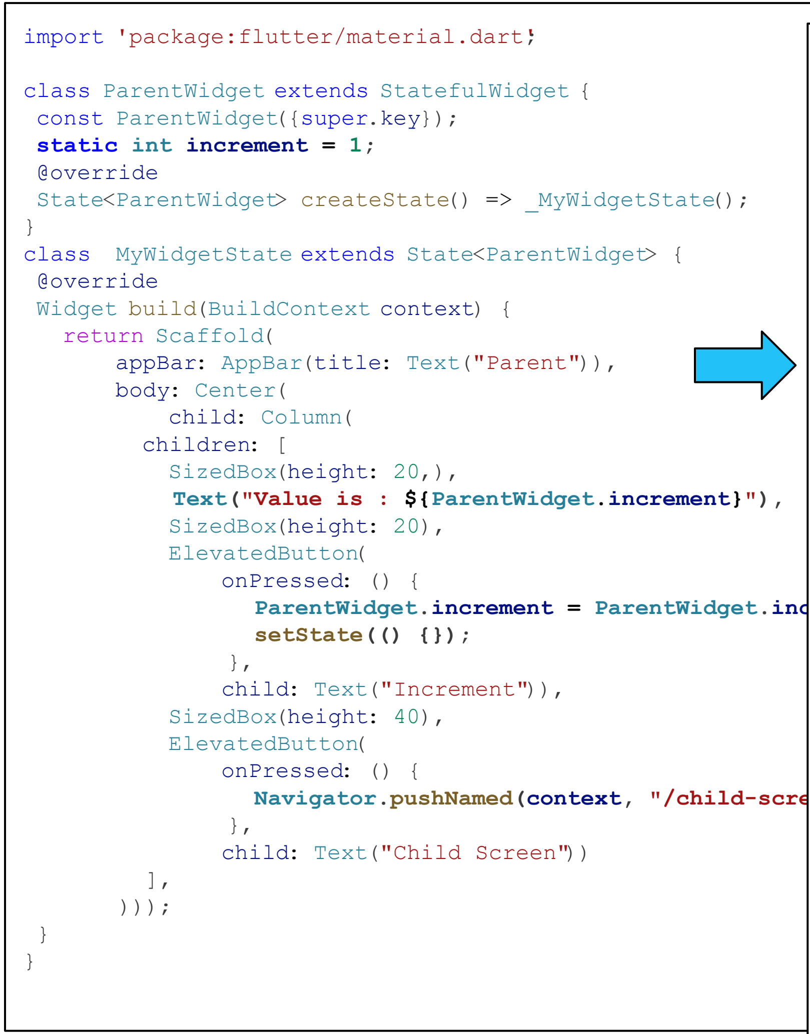
```
import 'package:flutter/material.dart';
                 gement_hello/screens/parent.dart';

              fulWidget {
              ;

              () => _MyWidgetState();

              State<MyWidget> {

             ontext) {

             e: Text("Child")),
   body: Center(
     child: Column(children: [
    SizedBox(
     height: 20,
              : ${ParentWidget.increment}"),

             get.increment = ParentWidget.increment + 1;

             ncrement"))
```



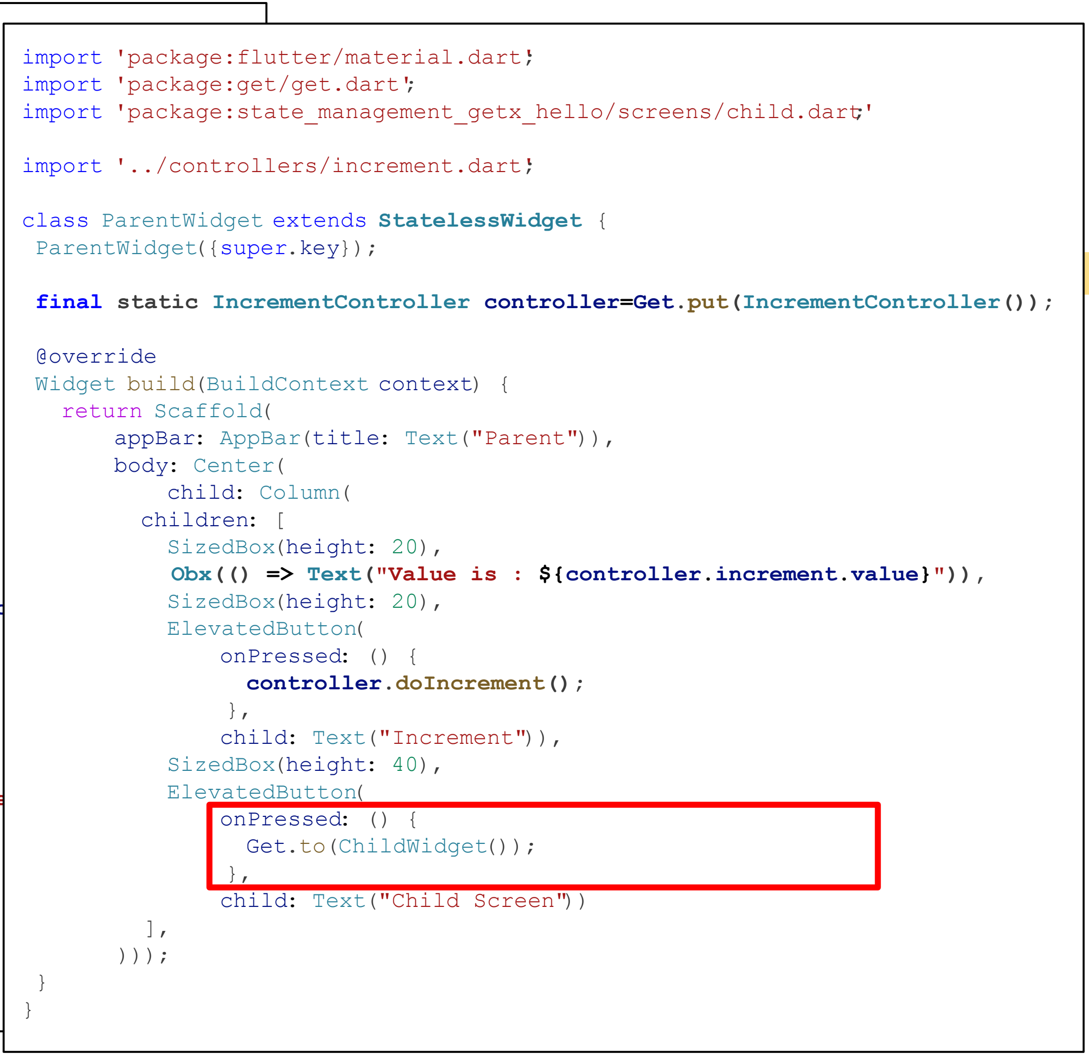


```
import 'package:flutter/material.dart';

class ParentWidget extends StatefulWidget {
 const ParentWidget({super.key});
 static int increment = 1;
 @override
 State<ParentWidget> createState() => _MyWidgetState();
}
class  MyWidgetState extends State<ParentWidget> {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
       appBar: AppBar(title: Text("Parent")),
       body: Center(
           child: Column(
         children: [
           SizedBox(height: 20,),
           Text("Value is : ${ParentWidget.increment}"),
           SizedBox(height: 20),
           ElevatedButton(
               onPressed: () {
                 ParentWidget.increment = ParentWidget.in
                 setState(() {});
               },
               child: Text("Increment")),
           SizedBox(height: 40),
           ElevatedButton(
               onPressed: () {
                 Navigator.pushNamed(context, "/child-scr
               },
               child: Text("Child Screen"))
         ],
       )));
 }
}
```

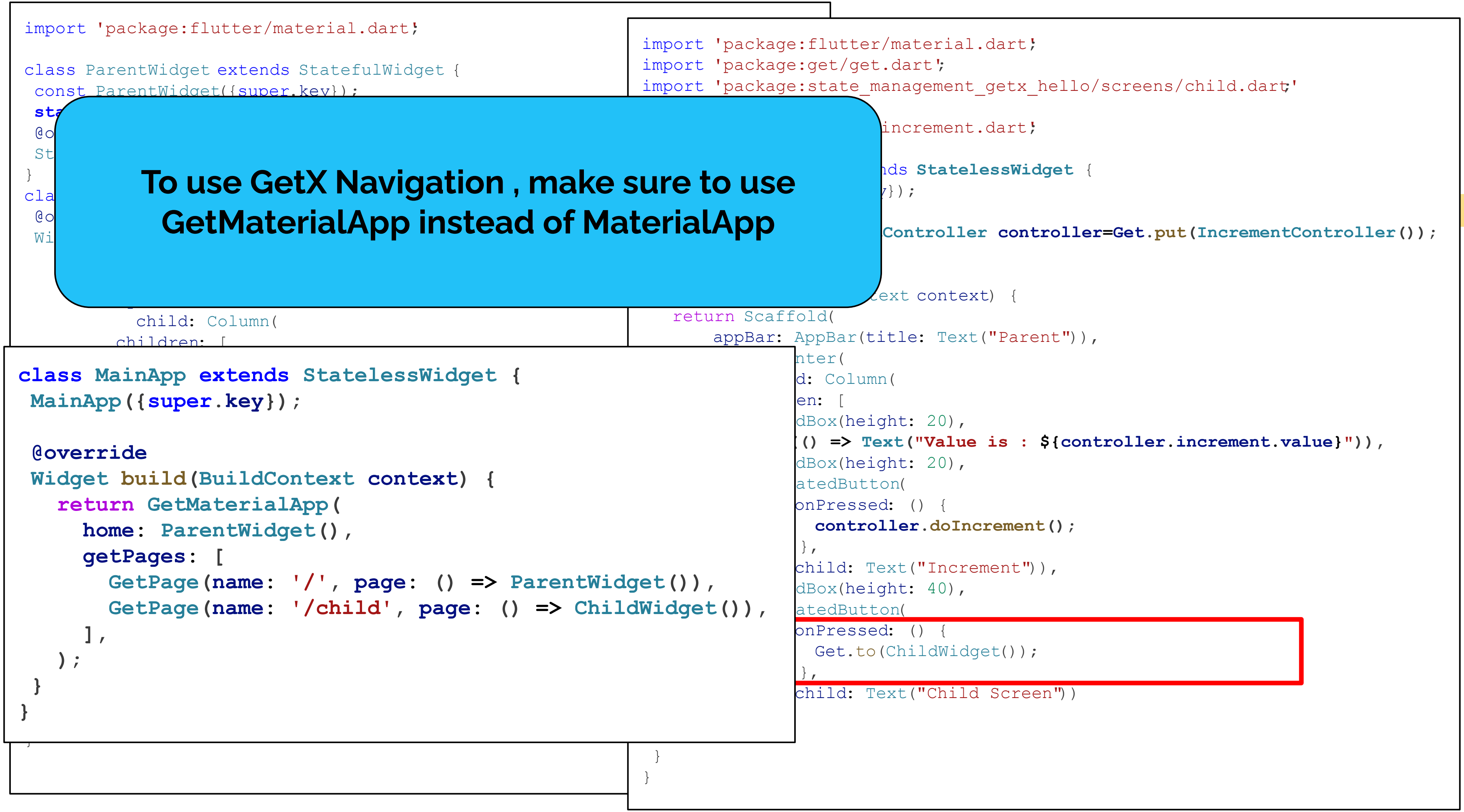
```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:state_management_getx_hello/screens/child.dart;'

import '../controllers/increment.dart';

class ParentWidget extends StatelessWidget {
 ParentWidget({super.key});

 static final IncrementController controller=Get.put(IncrementController());

 @override
 Widget build(BuildContext context) {
   return Scaffold(
       appBar: AppBar(title: Text("Parent")),
       body: Center(
           child: Column(
         children: [
           SizedBox(height: 20),
           Obx(() => Text("Value is : ${controller.increment.value}")),
           SizedBox(height: 20),
           ElevatedButton(
               onPressed: () {
                 controller.doIncrement();
               },
               child: Text("Increment")),
           SizedBox(height: 40),
           ElevatedButton(
               onPressed: () {
                 Get.to(ChildWidget());
               },
               child: Text("Child Screen"))
         ],
       )));
 }
}
```



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

class ParentWidget extends StatefulWidget {
 const ParentWidget({super.key});
 static int increment = 1;
 @override
 State<ParentWidget> createState() => _MyWidgetState();
}
class _MyWidgetState extends State<ParentWidget> {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
       appBar: AppBar(title: Text("Parent")),
       body: Center(
           child: Column(
         children: [
           SizedBox(height: 20,),
           Text("Value is : ${ParentWidget.increment}"),
           SizedBox(height: 20),
           ElevatedButton(
               onPressed: () {
                 ParentWidget.increment = ParentWidget.inc
                 setState(() {});
               },
               child: Text("Increment")),
           SizedBox(height: 40),
           ElevatedButton(
               onPressed: () {
                 Navigator.pushNamed(context, "/child-scre
               },
               child: Text("Child Screen"))
         ],
       )));
 }
}
```



```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:state_management_getx_hello/screens/child.dart';

import '../controllers/increment.dart';

class ParentWidget extends StatelessWidget {
 ParentWidget({super.key});

 final static IncrementController controller=Get.put(IncrementController());

 @override
 Widget build(BuildContext context) {
   return Scaffold(
       appBar: AppBar(title: Text("Parent")),
       body: Center(
           child: Column(
         children: [
           SizedBox(height: 20),
           Obx(() => Text("Value is : ${controller.increment.value}")),
           SizedBox(height: 20),
           ElevatedButton(
               onPressed: () {
                 controller.doIncrement();
               },
               child: Text("Increment")),
           SizedBox(height: 40),
           ElevatedButton(
               onPressed: () {
                 Get.to(ChildWidget());
               },
               child: Text("Child Screen"))
         ],
       )));
 }
}
```

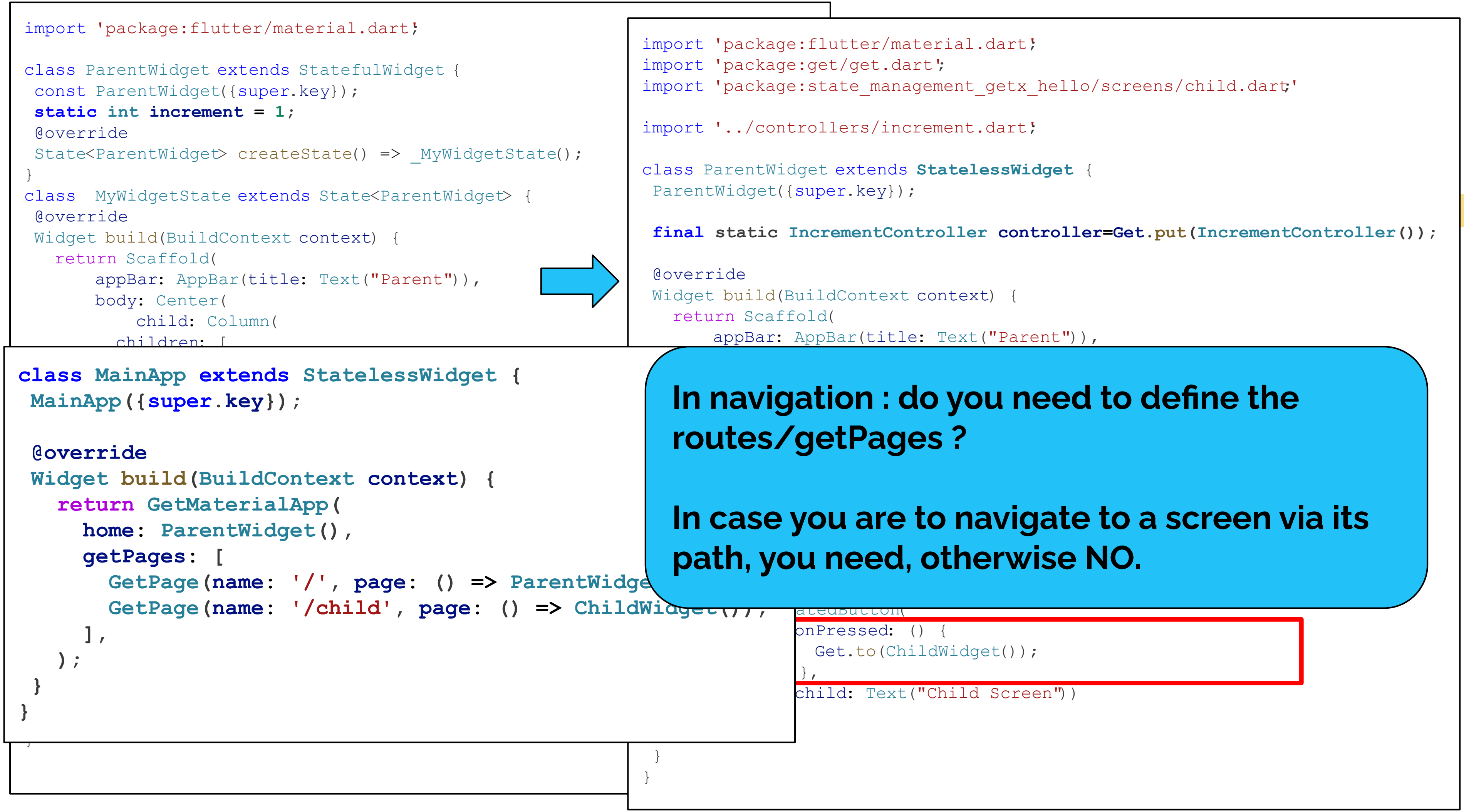


To use GetX Navigation , make sure to use GetMaterialApp instead of MaterialApp

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

class ParentWidget extends StatefulWidget {
 const ParentWidget({super.key});
 sta
 @o
 St
}
cla
 @o
 Wi

          child: Column(
        children: [
```

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:state_management_getx_hello/screens/child.dart;'

increment.dart';

nds StatelessWidget {
y});

Controller controller=Get.put(IncrementController());

ext context) {
   return Scaffold(
       appBar: AppBar(title: Text("Parent")),
nter(
d: Column(
en: [
dBox(height: 20),
() => Text("Value is : ${controller.increment.value}")),
dBox(height: 20),
atedButton(
onPressed: () {
  controller.doIncrement();
},
child: Text("Increment")),
dBox(height: 40),
atedButton(
onPressed: () {
  Get.to(ChildWidget());
},
child: Text("Child Screen"))

 }
}
```

```
class MainApp extends StatelessWidget {
 MainApp({super.key});

 @override
 Widget build(BuildContext context) {
   return GetMaterialApp(
     home: ParentWidget(),
     getPages: [
       GetPage(name: '/', page: () => ParentWidget()),
       GetPage(name: '/child', page: () => ChildWidget()),
     ],
   );
 }
}
```



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

class ParentWidget extends StatefulWidget {
 const ParentWidget({super.key});
 static int increment = 1;
 @override
 State<ParentWidget> createState() => _MyWidgetState();
}
class _MyWidgetState extends State<ParentWidget> {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
       appBar: AppBar(title: Text("Parent")),
       body: Center(
           child: Column(
         children: [
```

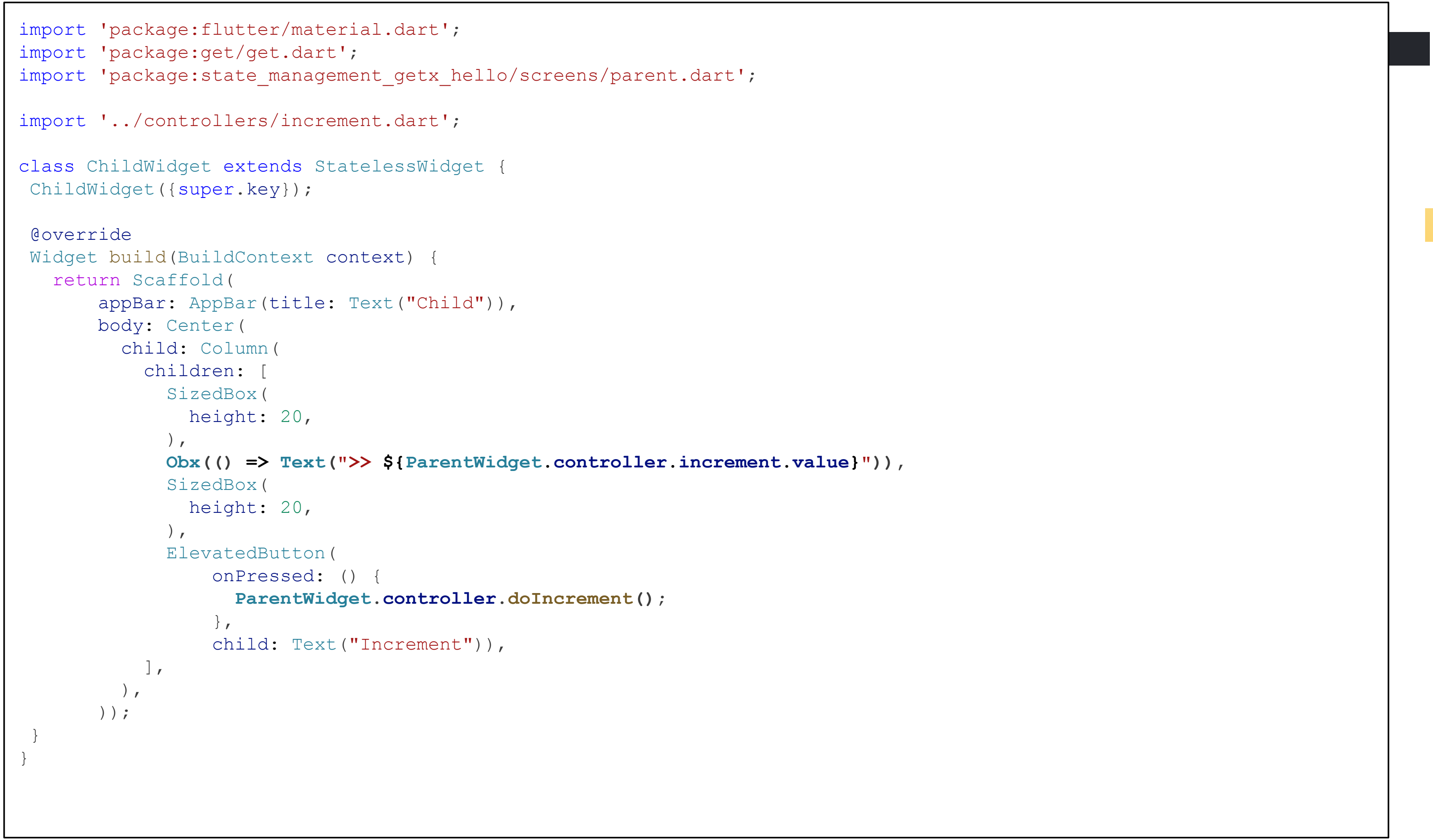
```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:state_management_getx_hello/screens/child.dart';

import '../controllers/increment.dart';

class ParentWidget extends StatelessWidget {
 ParentWidget({super.key});

 final static IncrementController controller=Get.put(IncrementController());

 @override
 Widget build(BuildContext context) {
   return Scaffold(
       appBar: AppBar(title: Text("Parent")),
```

```
                atedButton(
                onPressed: () {
                  Get.to(ChildWidget());
                },
                child: Text("Child Screen"))



   }
 }
```

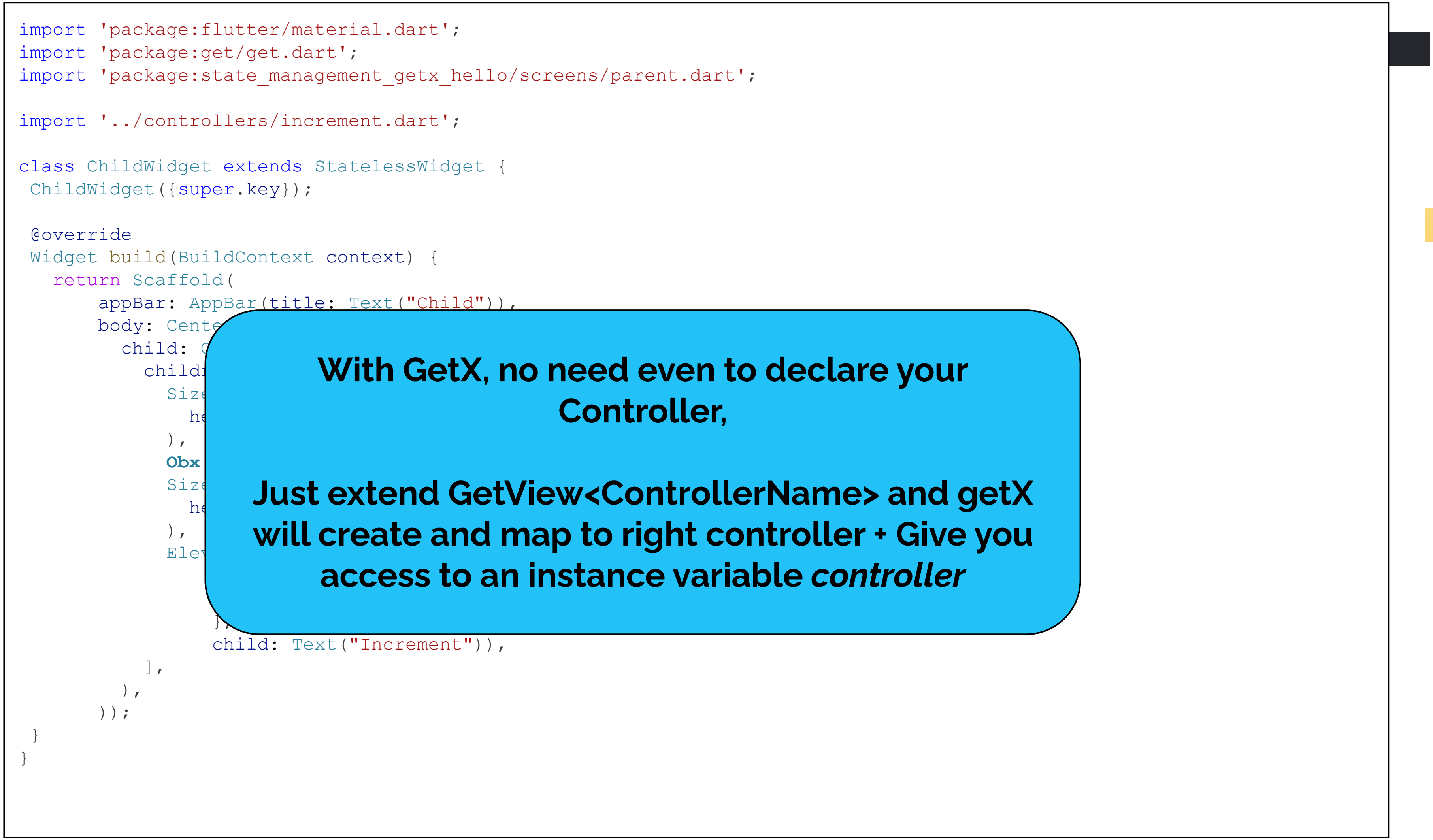
```
class MainApp extends StatelessWidget {
 MainApp({super.key});

 @override
 Widget build(BuildContext context) {
   return GetMaterialApp(
     home: ParentWidget(),
     getPages: [
       GetPage(name: '/', page: () => ParentWidge
       GetPage(name: '/child', page: () => ChildWidget()),
     ],
   );
 }
}
```

In navigation : do you need to define the routes/getPages ?

In case you are to navigate to a screen via its path, you need, otherwise NO.



```dart
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:state_management_getx_hello/screens/parent.dart';

import '../controllers/increment.dart';

class ChildWidget extends StatelessWidget {
 ChildWidget({super.key});

 @override
 Widget build(BuildContext context) {
   return Scaffold(
       appBar: AppBar(title: Text("Child")),
       body: Center(
         child: Column(
           children: [
             SizedBox(
               height: 20,
             ),
             Obx(() => Text(">> ${ParentWidget.controller.increment.value}")),
             SizedBox(
               height: 20,
             ),
             ElevatedButton(
                 onPressed: () {
                   ParentWidget.controller.doIncrement();
                 },
                 child: Text("Increment")),
           ],
         ),
       ));
 }
}
```



```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:state_management_getx_hello/screens/parent.dart';

import '../controllers/increment.dart';

class ChildWidget extends StatelessWidget {
 ChildWidget({super.key});

 @override
 Widget build(BuildContext context) {
   return Scaffold(
       appBar: AppBar(title: Text("Child")),
       body: Cente
         child: C
           child
             Size
               he
             ),
             Obx
             Size
               he
             ),
             Elev


                 })
                 child: Text("Increment")),
           ],
         ),
       ));
 }
}
```

With GetX, no need even to declare your Controller,

Just extend GetView<ControllerName> and getX will create and map to right controller + Give you access to an instance variable *controller*

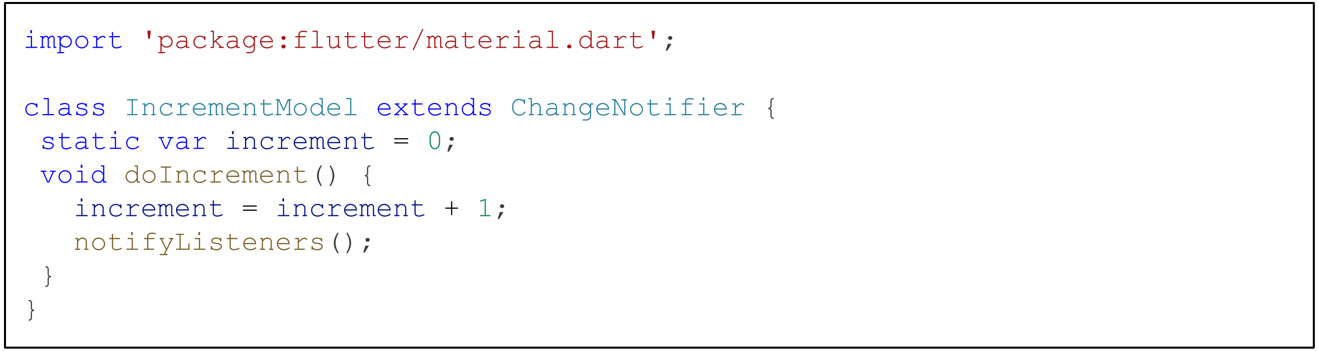
```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:state_management_getx_hello/screens/child.dart;'

import '../controllers/increment.dart';

class ParentWidget extends GetView<IncrementController> {


 ParentWidget({super.key});

 @override
 Widget build(BuildContext context) {
   return Scaffold(
       appBar: AppBar(title: Text("Parent")),
       body: Center(
           child: Column(
         children: [
           SizedBox(height: 20),
           Obx(() => Text("Value is : ${controller.increment.value}")),
           SizedBox(height: 20),
           ElevatedButton(
               onPressed: () {
                 controller.doIncrement();
                },
               child: Text("Increment")),
           SizedBox(height: 40),
           ElevatedButton(
               onPressed: () {
                 Get.to(ChildWidget());
                },
               child: Text("Child Screen"))
         ],
       )));
 }
}
```

controller : is inherited from GetView

```dart
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:state_management_getx_hello/screens/parent.dart;'

import '../controllers/increment.dart';

class ChildWidget extends GetView<IncrementController> {
 ChildWidget({super.key});

 @override
 Widget build(BuildContext context) {
   return Scaffold(
       appBar: AppBar(title: Text("Child")),
       body: Center(
         child: Column(
           children: [
             SizedBox(height: 20),
             Obx(() => Text(">> ${controller.increment.value}")),
             SizedBox(height: 20),
             ElevatedButton(
                 onPressed: () {
                   controller.doIncrement();
                 },
                 child: Text("Increment")),
           ],
         ),
       ));
 }
}
```

State Management in Flutter

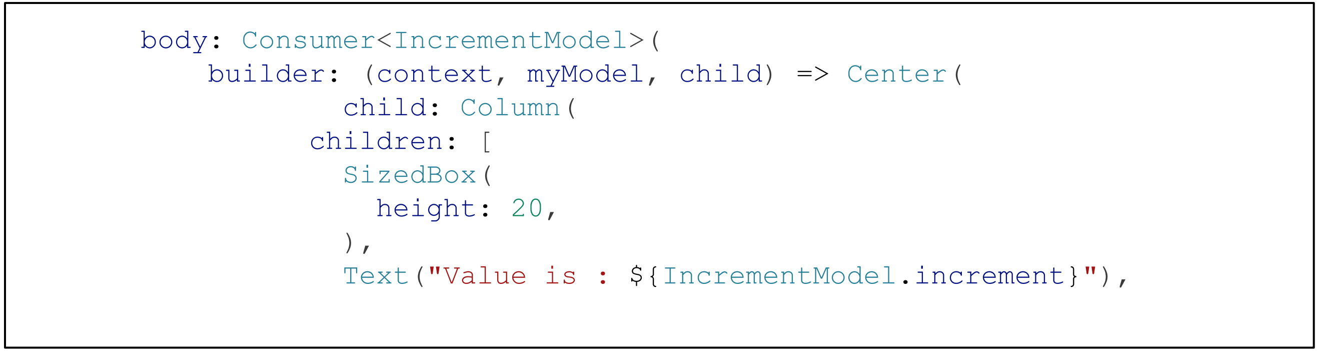
- **Provider** : Part of the Flutter Foundation Framework (Supported by Google)
  - The same way as getX:
    - Controller instead of extending GetxController,
      - It needs to extend : **ChangeNotifier**



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

class IncrementModel extends ChangeNotifier {
 static var increment = 0;
 void doIncrement() {
   increment = increment + 1;
   notifyListeners();
 }
}
```

State Management in Flutter

- **Provider** : Part of the Flutter Foundation Framework (Supported by Google)
  - The same way as getX:
    - Instead of using Obx (()=> Widget)
      - Use **Consumer<ControllerName>**

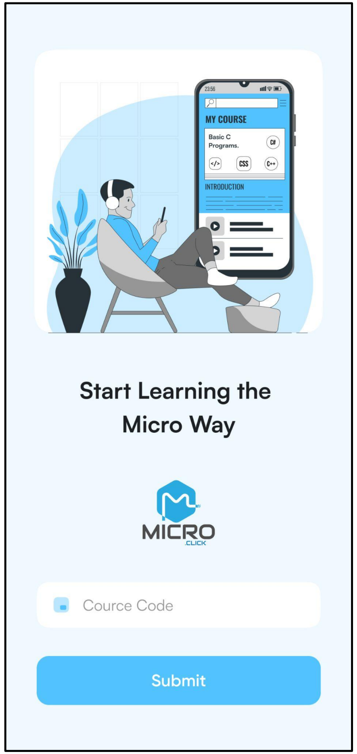


```
body: Consumer<IncrementModel>(
  builder: (context, myModel, child) => Center(
      child: Column(
     children: [
      SizedBox(
       height: 20,
      ),
      Text("Value is : ${IncrementModel.increment}"),
```

Section 2

Building your MVP using Flutter

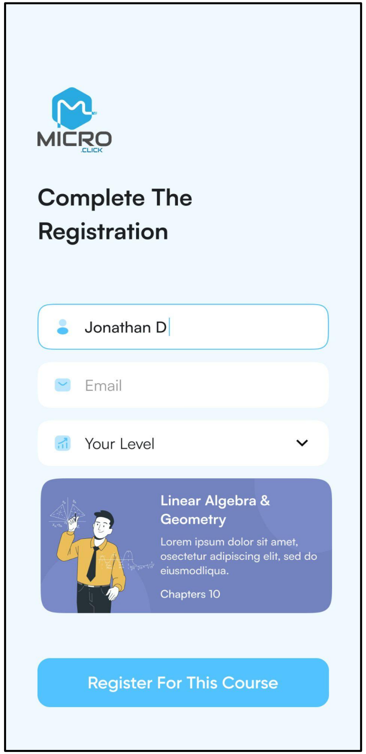
Building Your MVP using Flutter



Start Learning the Micro Way

MICRO .CLICK

Cource Code



MICRO .CLICK

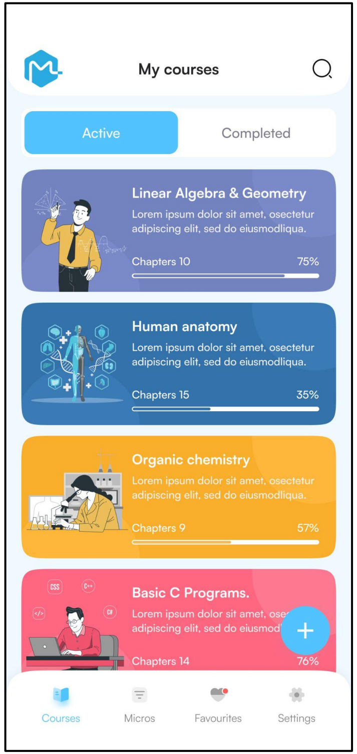
Complete The Registration

Jonathan D

Email

Your Level

Linear Algebra & Geometry
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 10



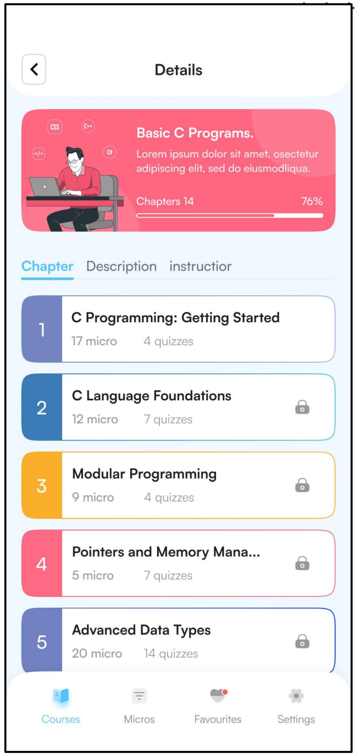
My courses

Linear Algebra & Geometry
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 10 75%

Human anatomy
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 15 35%

Organic chemistry
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 9 57%

Basic C Programs.
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 14 76%



Details

Basic C Programs.
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 14 76%

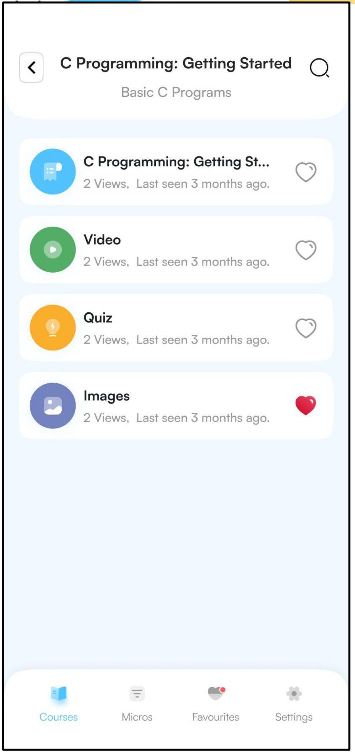
1 C Programming: Getting Started
17 micro 4 quizzes

2 C Language Foundations
12 micro 7 quizzes

3 Modular Programming
9 micro 4 quizzes

4 Pointers and Memory Mana...
5 micro 7 quizzes

5 Advanced Data Types
20 micro 14 quizzes



C Programming: Getting Started
Basic C Programs

C Programming: Getting St...
2 Views, Last seen 3 months ago.

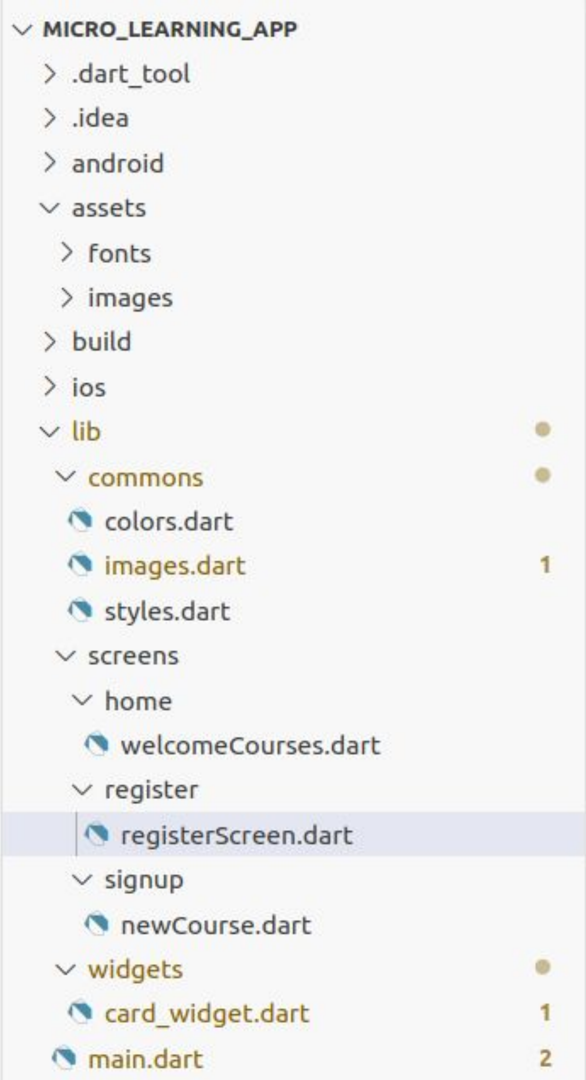
Video
2 Views, Last seen 3 months ago.

Quiz
2 Views, Last seen 3 months ago.

Images
2 Views, Last seen 3 months ago.

Building Your MVP using Flutter

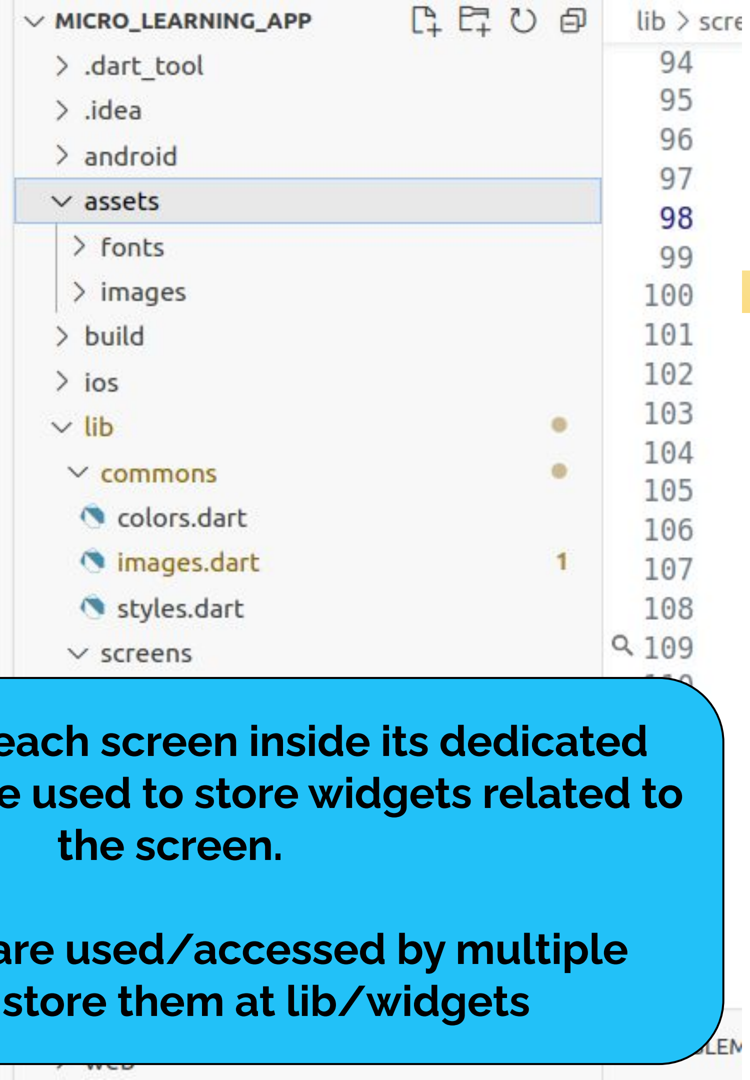
- **Structuring the Project**
  - **assets**
    - **fonts**
    - **images**
  - **lib**
    - **commons**
      - colors.dart
    - **screens**
      - **signup**
        - signup.dart
        - bottomsection.dart
      - **home**
      - **setting**
    - **widgets**
      - navbar.dart
      - drawer.dart
  - main.dart
  - pubspec.yaml



```
MICRO_LEARNING_APP
  > .dart_tool
  > .idea
  > android
  v assets
    > fonts
    > images
  > build
  > ios
  v lib
    v commons
        colors.dart
        images.dart                1
        styles.dart
    v screens
      v home
          welcomeCourses.dart
      v register
          registerScreen.dart
      v signup
          newCourse.dart
    v widgets
        card_widget.dart           1
      main.dart                    2
```

Building Your MVP using Flutter

- **Structuring the Project**
    - **assets**
        - **fonts**
        - **images**
    - **lib**
        - **commons**
            - colors.dart
        - **screens**
            - **signup**
                - signup.dart
                - bottomsection.d
            - **home**
            - **setting**
        - **widgets**
            - navbar.dart
            - drawer.dart
    - main.dart
    - pubspec.yaml



Better to have each screen inside its dedicated folder that can be used to store widgets related to the screen.

Widgets that are used/accessed by multiple screams, store them at lib/widgets

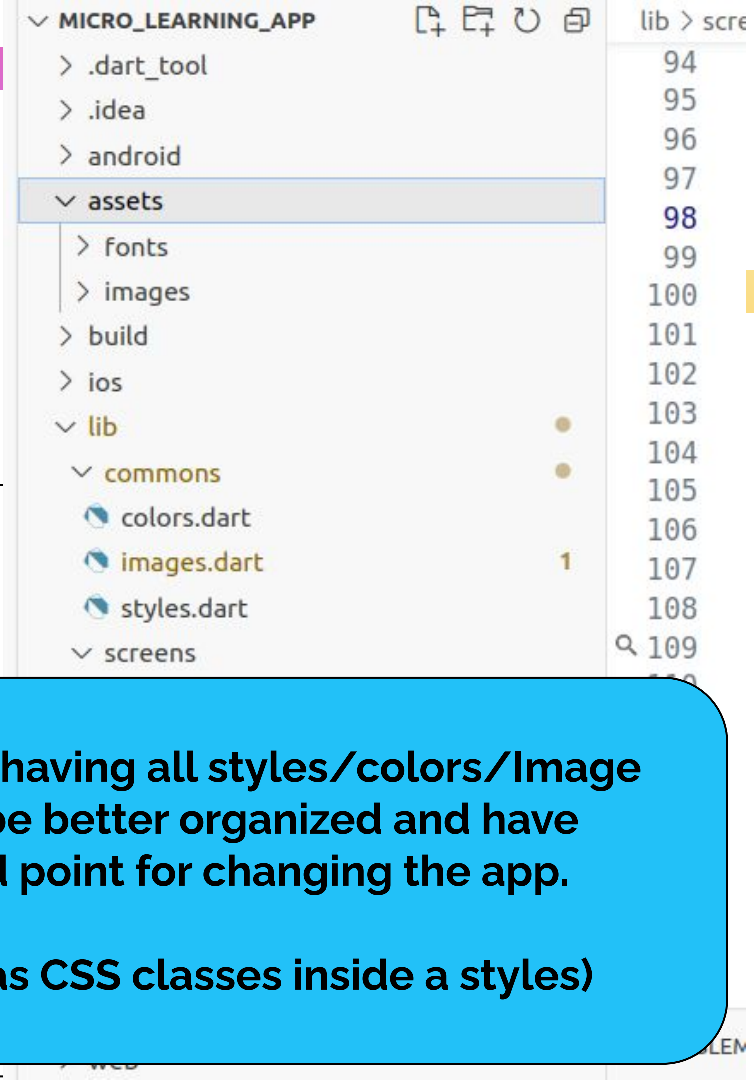
Building Your MVP using Flutter

- Structuring the Project
  - lib / commons / styles.dart

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

const startTextStyle = TextStyle(
 fontSize: 25.0,
 color: textColor,
 fontFamily: "Satoshi",
 fontWeight: FontWeight.w700,
 fontStyle: FontStyle.normal,
);
const buttonTextStyle = TextStyle(
 fontSize: 18.0,
 color: whiteColor,
 fontFamily: "Satoshi",
 fontWeight: FontWeight.w600,
 fontStyle: FontStyle.normal,
);
…
```

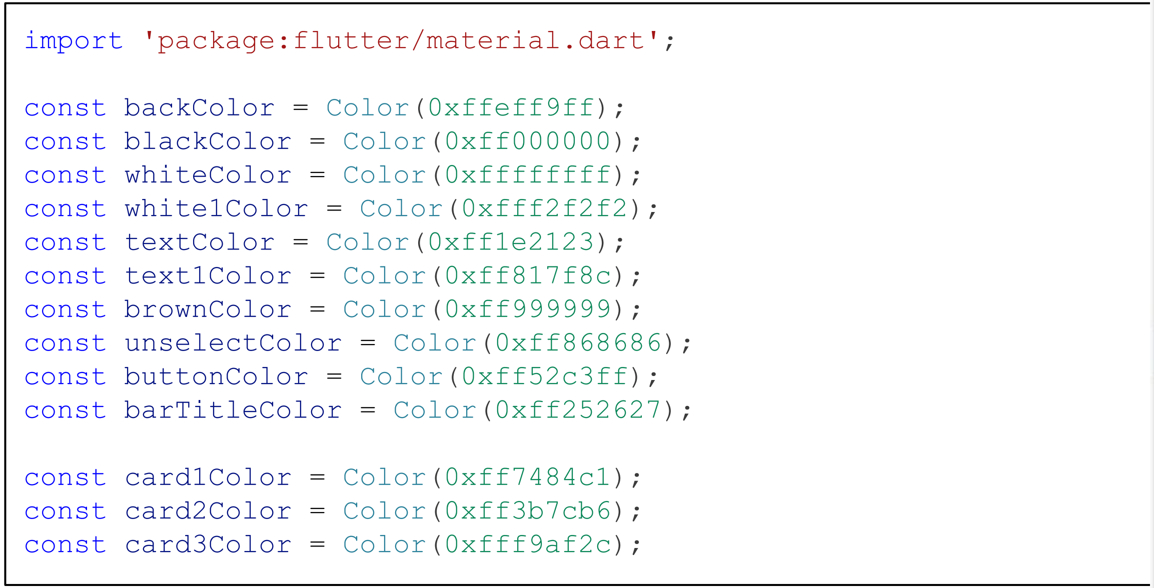
The benefit of having all styles/colors/Image URLs is to be better organized and have centralized point for changing the app.

(Same way as CSS classes inside a styles)

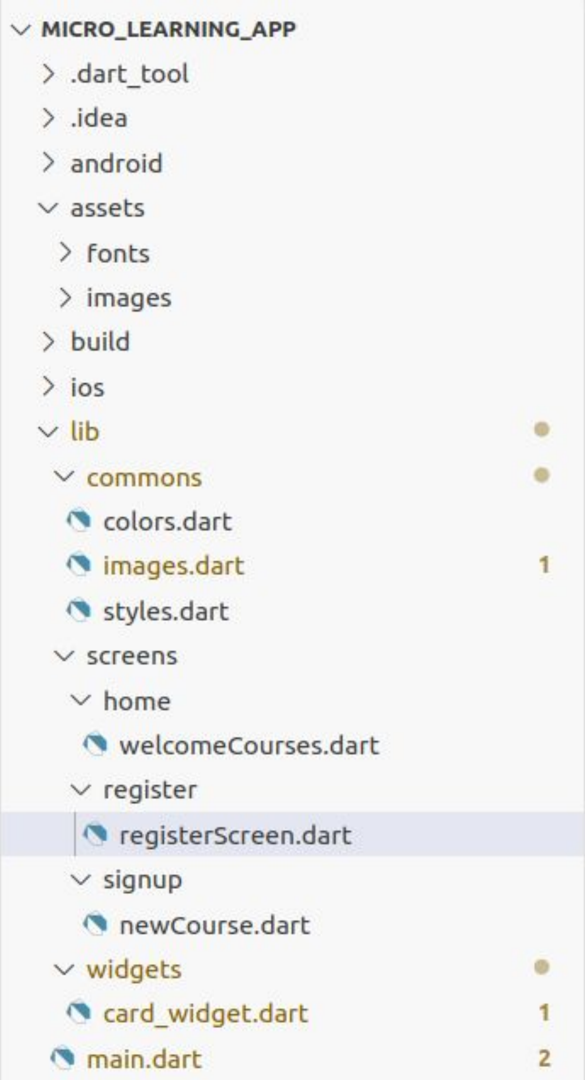


Building Your MVP using Flutter

- **Structuring the Project**
    - **lib / commons / colors.dart**



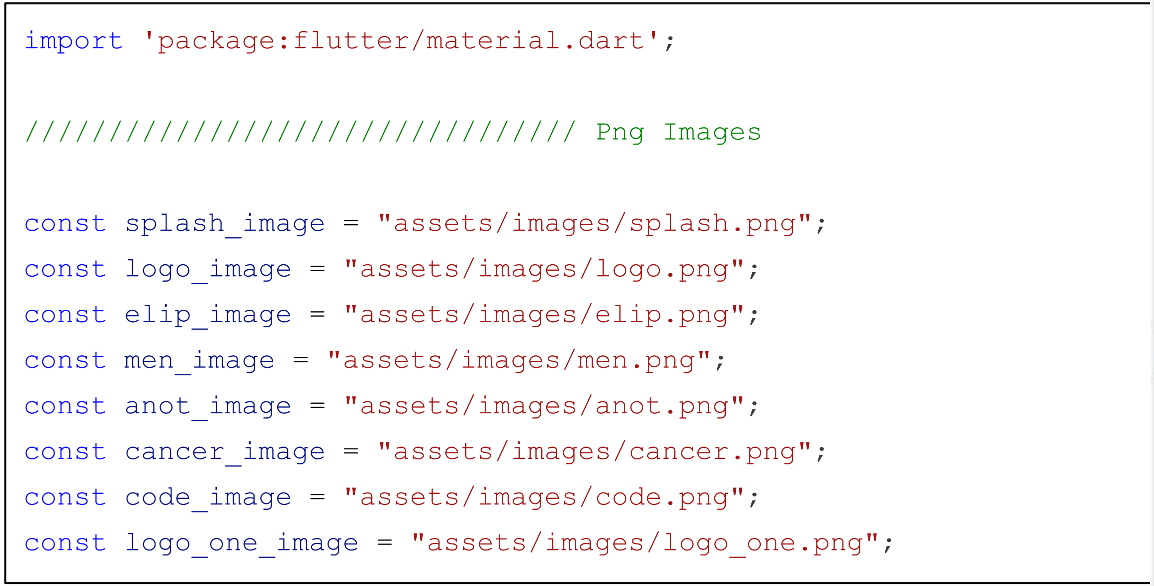
```dart
import 'package:flutter/material.dart';

const backColor = Color(0xffeff9ff);
const blackColor = Color(0xff000000);
const whiteColor = Color(0xffffffff);
const white1Color = Color(0xfff2f2f2);
const textColor = Color(0xff1e2123);
const text1Color = Color(0xff817f8c);
const brownColor = Color(0xff999999);
const unselectColor = Color(0xff868686);
const buttonColor = Color(0xff52c3ff);
const barTitleColor = Color(0xff252627);

const card1Color = Color(0xff7484c1);
const card2Color = Color(0xff3b7cb6);
const card3Color = Color(0xfff9af2c);
```



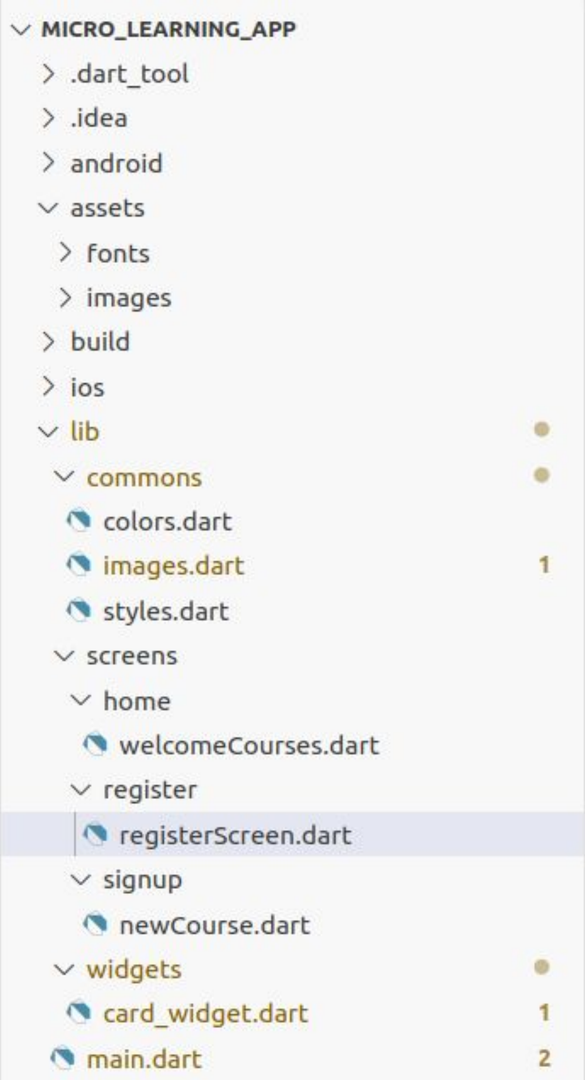
```
MICRO_LEARNING_APP
  .dart_tool
  .idea
  android
  assets
    fonts
    images
  build
  ios
  lib
    commons
      colors.dart
      images.dart                 1
      styles.dart
    screens
      home
        welcomeCourses.dart
      register
        registerScreen.dart
      signup
        newCourse.dart
    widgets
      card_widget.dart            1
    main.dart                     2
```

Building Your MVP using Flutter

- **Structuring the Project**
  - **lib / commons / images.dart**



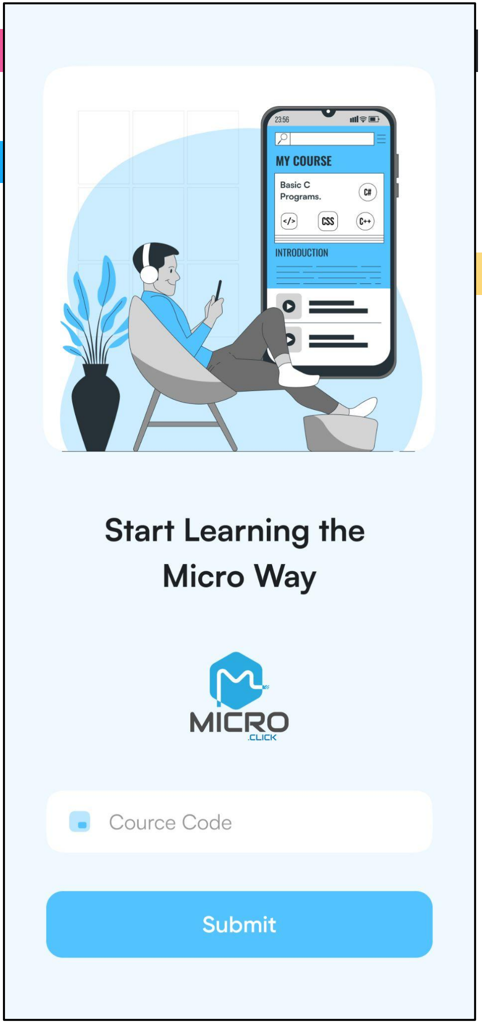
```
import 'package:flutter/material.dart';

/////////////////////////////////// Png Images

const splash_image = "assets/images/splash.png";
const logo_image = "assets/images/logo.png";
const elip_image = "assets/images/elip.png";
const men_image = "assets/images/men.png";
const anot_image = "assets/images/anot.png";
const cancer_image = "assets/images/cancer.png";
const code_image = "assets/images/code.png";
const logo_one_image = "assets/images/logo_one.png";
```



```
MICRO_LEARNING_APP
  .dart_tool
  .idea
  android
  assets
    fonts
    images
  build
  ios
  lib
    commons
      colors.dart
      images.dart                1
      styles.dart
    screens
      home
        welcomeCourses.dart
      register
        registerScreen.dart
      signup
        newCourse.dart
    widgets
      card_widget.dart           1
    main.dart                    2
```

Building Your MVP using Flutter

- **Tree of Widgets :**
    - MaterialApp
        - ?
            - ?
                - ?

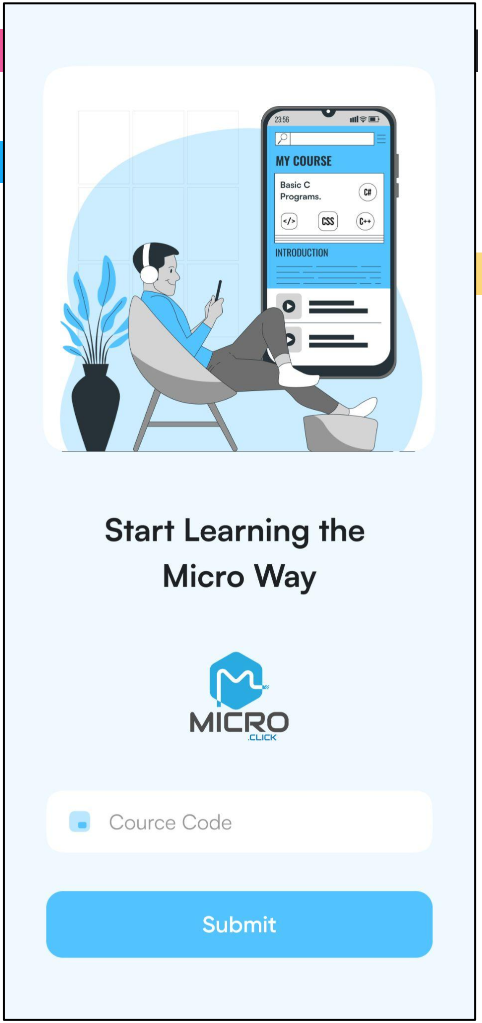


Start Learning the Micro Way

Cource Code

Building Your MVP using Flutter

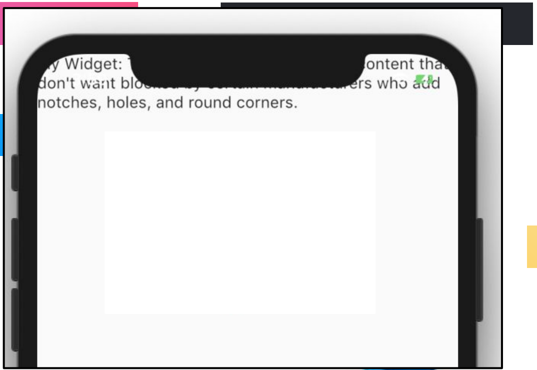
- **Tree of Widgets :**
    - MaterialApp
        - Scaffold
            - Column
                - Image
                - Text
                - Image
                - TextFormField
                - ElevatedButton



Start Learning the Micro Way

Cource Code

Building Your MVP using Flutter

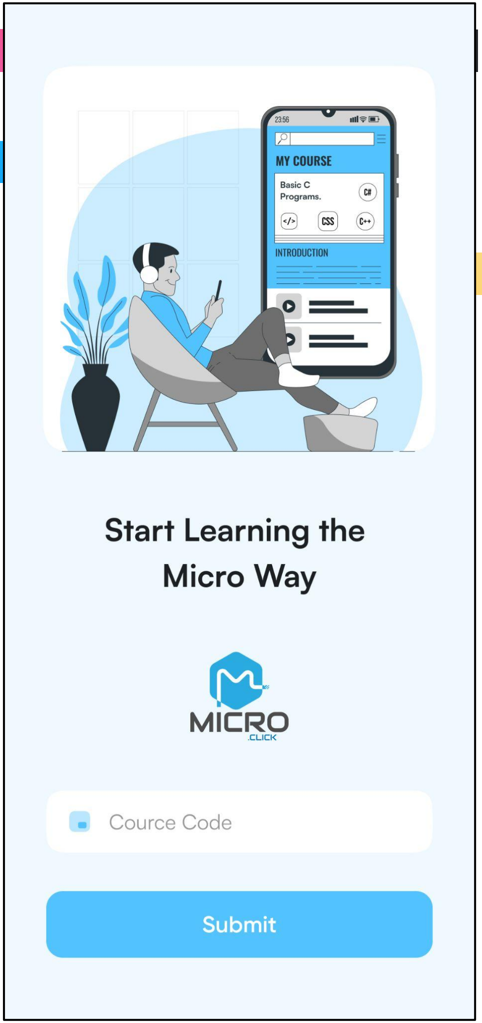


y Widget: ontent tha
don't want blo ers who add
notches, holes, and round corners.

- **Tree of Widgets**
  - **SafeArea** : A Container Widget that adds padding automatically to make sure that your widget is visible and hidden by the physical constraints.
  - **InkWell** : Like GestorDetector but less options. Used for detecting touch events (OnTap …)
  - **Ink** : A Convenience Widget like a Container mostly used for decorating
  - **SingleChildScrollView**
  - **InputDecoration** : Used to decorate and style form widgets

Building Your MVP using Flutter

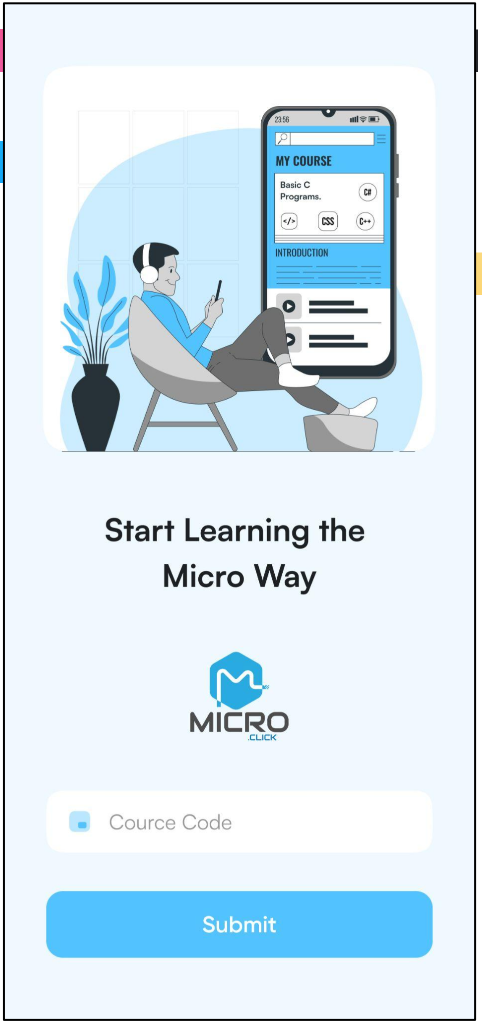
- **Tree of Widgets :**
  - MaterialApp
    - Scaffold
      - SafeArea
        - SingleChildScrollView
          - Column
            - Image
            - Text
            - Image
            - TextFormField
              - InputDecoration
            - ElevatedButton



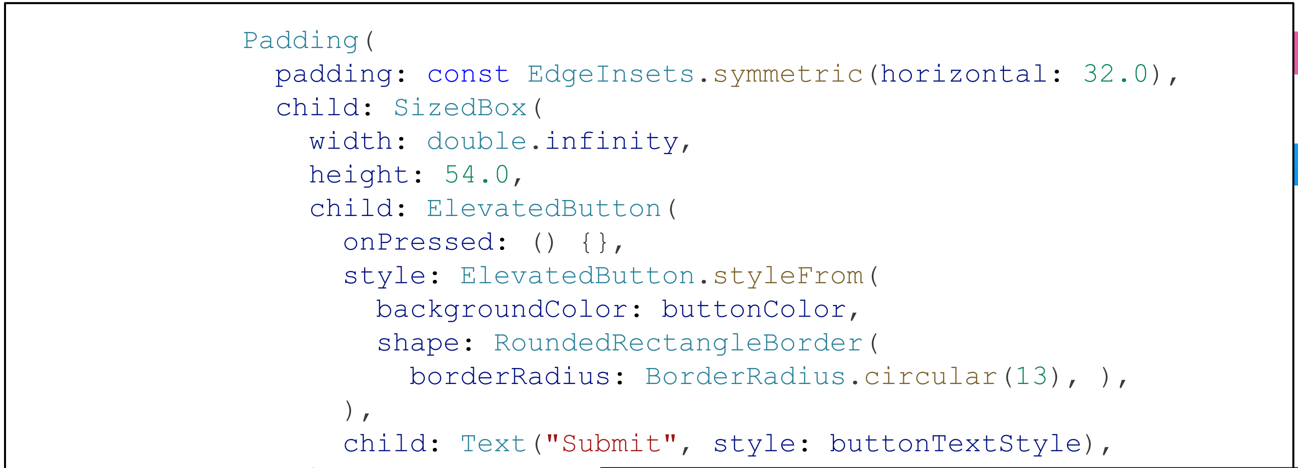
Start Learning the Micro Way

Building Your MVP using Flutter

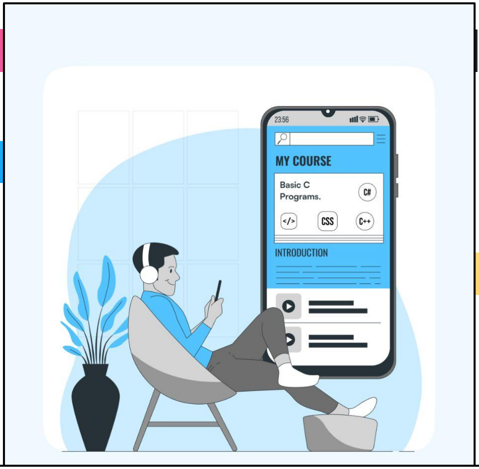
- **Tree of Widgets :**
  - MaterialApp
    - Scaffold
      - SafeArea
        - SingleChildScrollView
          - Column
            - Image
            - Text
            - Image
            - TextFormField
              - InputDecoration
            - **InkWell**
              - **Ink**
                - **DecorationBox**
                - **Text**

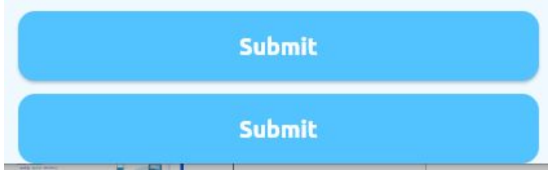


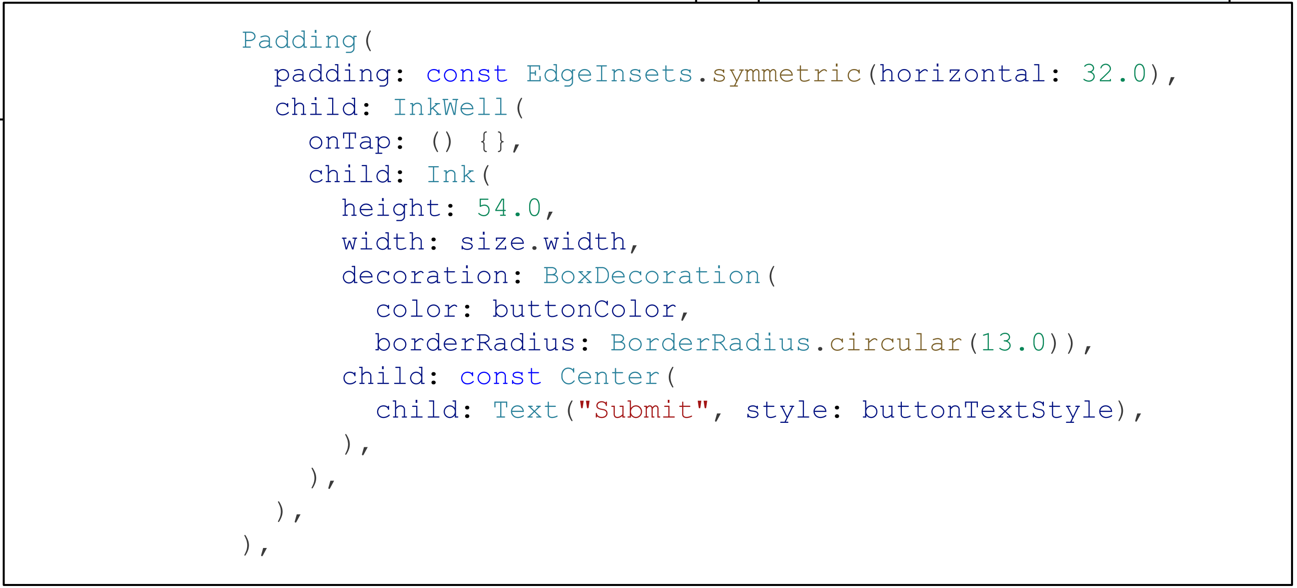
Start Learning the Micro Way



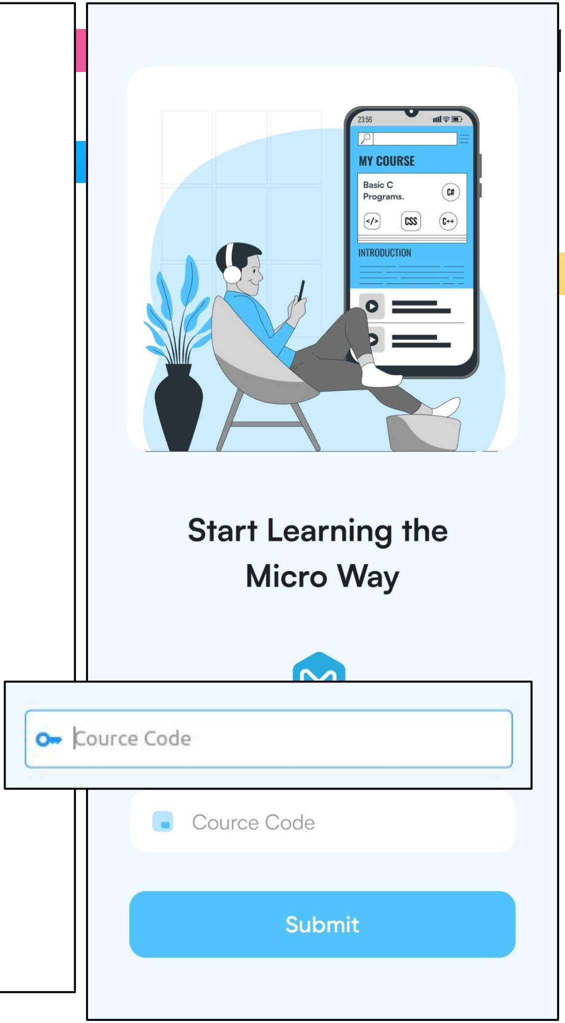
```
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 32.0),
  child: SizedBox(
    width: double.infinity,
    height: 54.0,
    child: ElevatedButton(
      onPressed: () {},
      style: ElevatedButton.styleFrom(
        backgroundColor: buttonColor,
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(13), ),
      ),
      child: Text("Submit", style: buttonTextStyle),
    ),
  ),
),
```







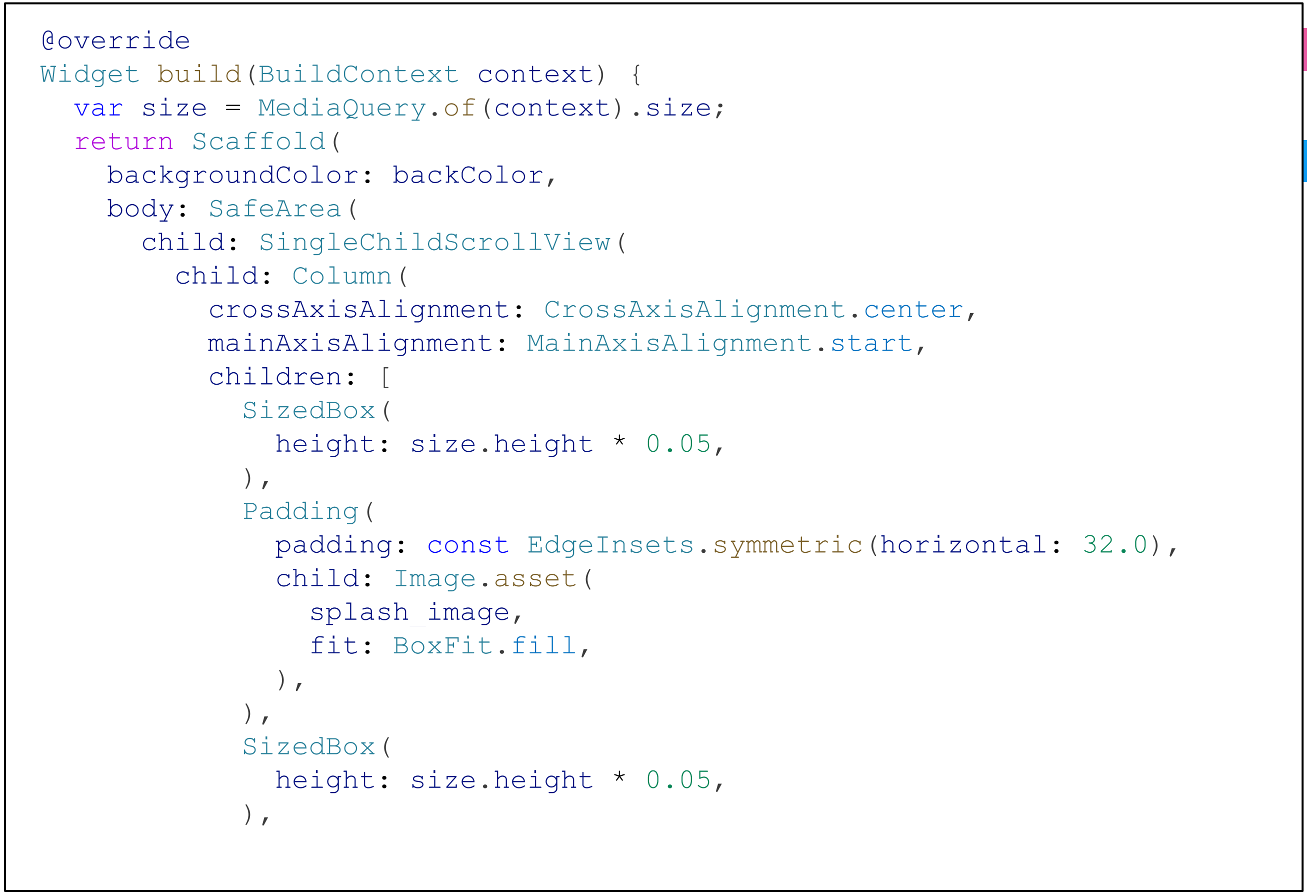
```
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 32.0),
  child: InkWell(
    onTap: () {},
    child: Ink(
      height: 54.0,
      width: size.width,
      decoration: BoxDecoration(
        color: buttonColor,
        borderRadius: BorderRadius.circular(13.0)),
      child: const Center(
        child: Text("Submit", style: buttonTextStyle),
      ),
    ),
  ),
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 32.0),
  child: Container(
    height: 50.0,
    width: size.width,
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(13.0),
      color: whiteColor,
    ),
    child: TextFormField(
      cursorColor: brownColor,
      style: fieldTextStyle,
      decoration: const InputDecoration(
        border: InputBorder.none,
        focusedBorder: OutlineInputBorder(
            borderRadius: BorderRadius.all(Radius.circular(13.0)),
            borderSide: BorderSide(color: Colors.blue)),
        prefixIcon: SizedBox(
          height: 16.0,
          width: 16.0,
          child: Icon(Icons.key),
        ),
        hintText: "Cource Code",
        hintStyle: codeTextStyle,
      ),
    ),
  ),
),
```



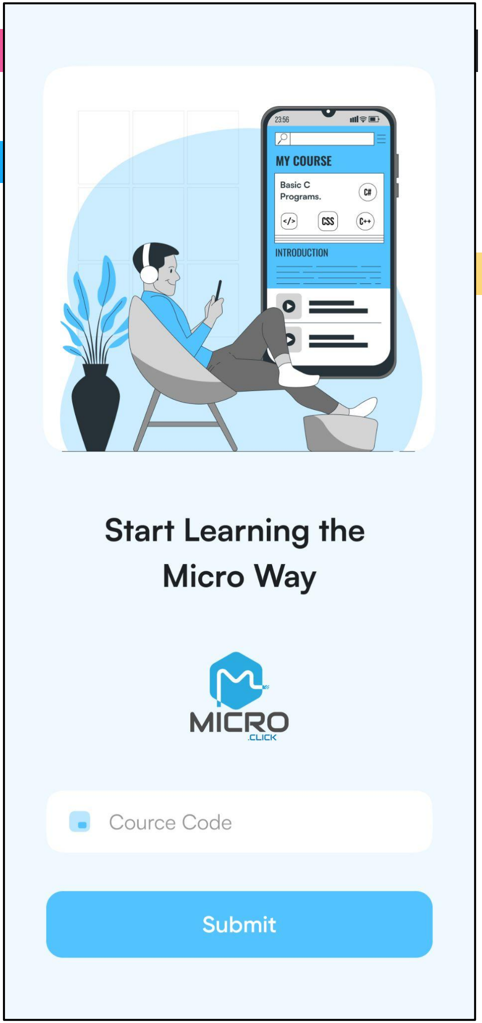
Start Learning the Micro Way

Cource Code

Cource Code



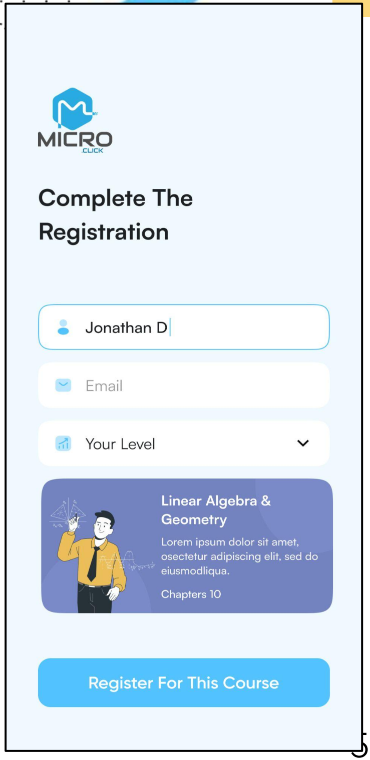
```
@override
Widget build(BuildContext context) {
  var size = MediaQuery.of(context).size;
  return Scaffold(
    backgroundColor: backColor,
    body: SafeArea(
      child: SingleChildScrollView(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.center,
          mainAxisAlignment: MainAxisAlignment.start,
          children: [
            SizedBox(
              height: size.height * 0.05,
            ),
            Padding(
              padding: const EdgeInsets.symmetric(horizontal: 32.0),
              child: Image.asset(
                splash_image,
                fit: BoxFit.fill,
              ),
            ),
            SizedBox(
              height: size.height * 0.05,
            ),
```



Start Learning the
Micro Way

Building Your MVP using Flutter

- **Tree of Widgets :**
  - MaterialApp
    - Scaffold
      - SafeArea
        - SingleChildScrollView
          - Column
            - SizedBox
              - Image
            - Text
            - TextFormField
              - InputDecoration
            - DropdownButtonFormField<String>
              - InputDecoration
            - ?
              - ?
              - ?
                - ?



Complete The Registration

Jonathan D

Email

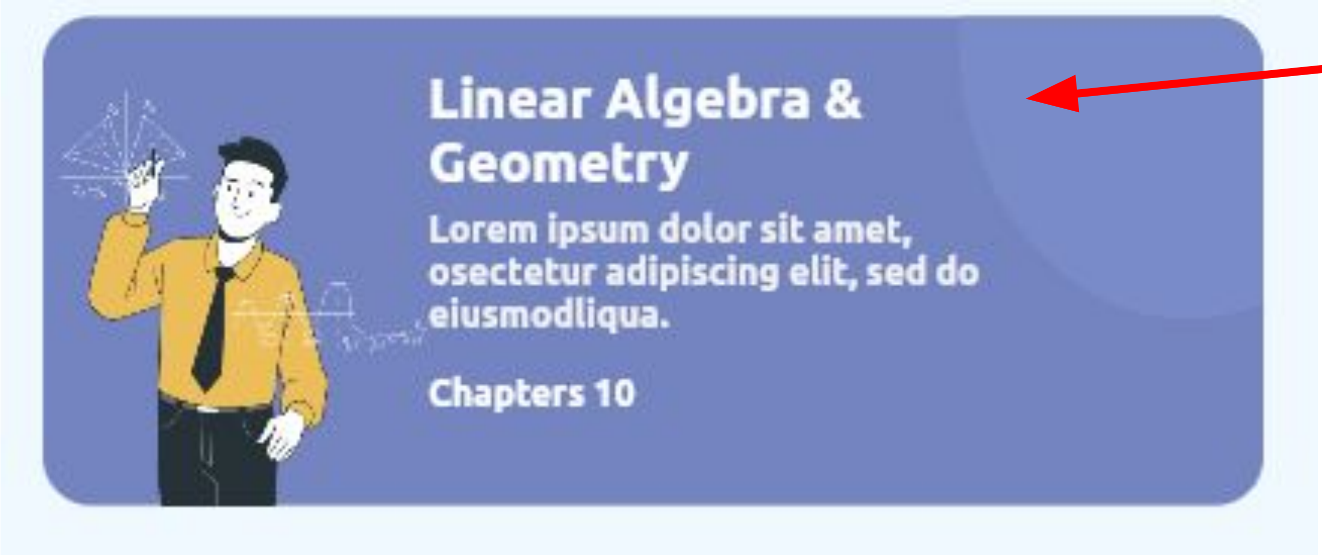
Your Level

Linear Algebra & Geometry

Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.

Chapters 10

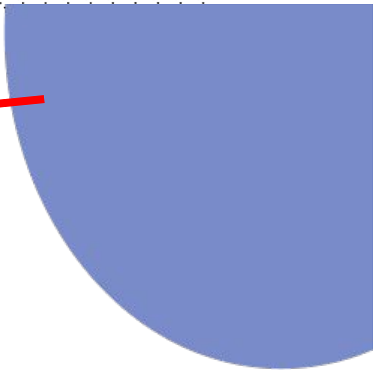
Building Your MVP using Flutter



Linear Algebra & Geometry

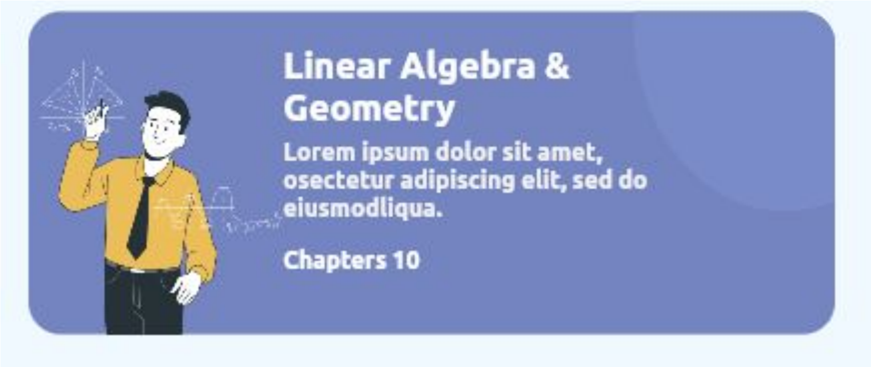
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.

Chapters 10



Building Your MVP using Flutter

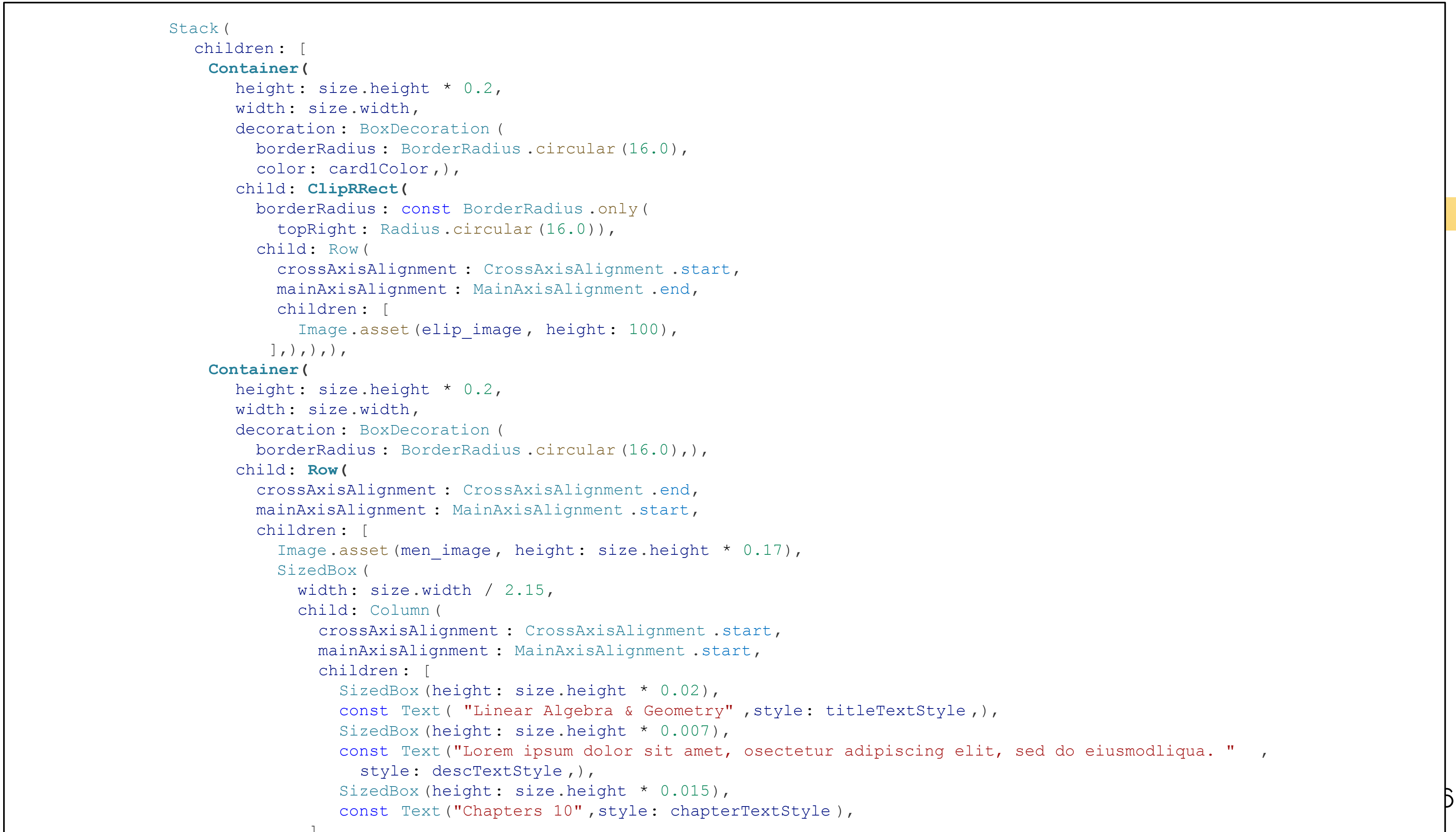
- **Tree of Widgets :**
    - Stack
        - Container
            - BoxDecoration
            - ClipRRect
                - Row
                    - Image.asset
        - Container
            - BoxDecoration
            - Row
                - Image.asset
                - Column
                    - Text
                    - Text
                    - Text



Linear Algebra & Geometry

Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.

Chapters 10

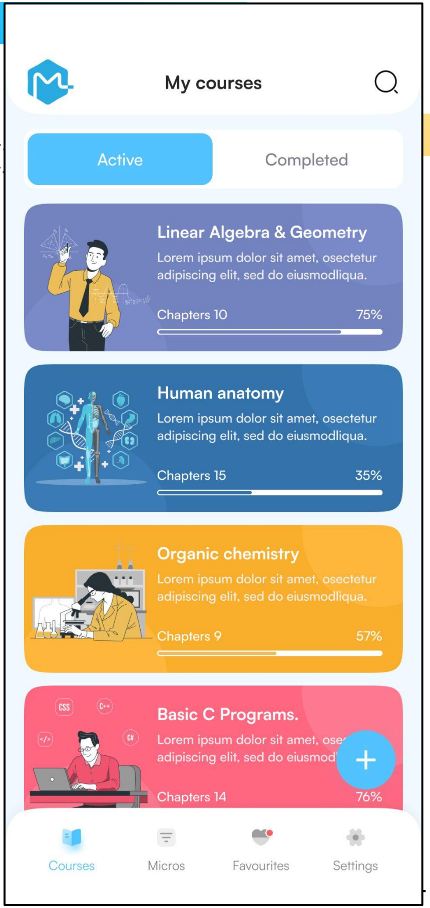


```
Stack(
  children: [
    Container(
      height: size.height * 0.2,
      width: size.width,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(16.0),
        color: card1Color,),
      child: ClipRRect(
        borderRadius: const BorderRadius.only(
          topRight: Radius.circular(16.0)),
        child: Row(
          crossAxisAlignment: CrossAxisAlignment.start,
          mainAxisAlignment: MainAxisAlignment.end,
          children: [
            Image.asset(elip_image, height: 100),
         ],),),),
    Container(
      height: size.height * 0.2,
      width: size.width,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(16.0),),
      child: Row(
        crossAxisAlignment: CrossAxisAlignment.end,
        mainAxisAlignment: MainAxisAlignment.start,
        children: [
          Image.asset(men_image, height: size.height * 0.17),
          SizedBox(
            width: size.width / 2.15,
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              mainAxisAlignment: MainAxisAlignment.start,
              children: [
                SizedBox(height: size.height * 0.02),
                const Text( "Linear Algebra & Geometry" ,style: titleTextStyle,),
                SizedBox(height: size.height * 0.007),
                const Text("Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua. "   ,
                  style: descTextStyle,),
                SizedBox(height: size.height * 0.015),
                const Text("Chapters 10",style: chapterTextStyle),
              ]
```

Building Your MVP using Flutter

- **Tree of Widgets :**
  - MaterialApp
    - Scaffold
      - ?

The user can slide from left to right between different pages



Linear Algebra & Geometry
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 10 75%

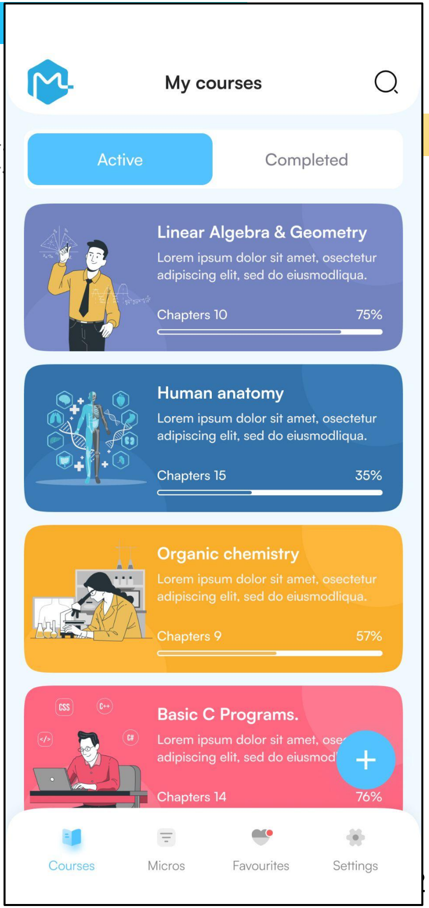
Human anatomy
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 15 35%

Organic chemistry
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 9 57%

Basic C Programs.
Lorem ipsum dolor sit amet, ose
adipiscing elit, sed do eiusmod
Chapters 14 76%

Building Your MVP using Flutter

- **Tree of Widgets :**
    - MaterialApp
        - Scaffold
            - GetTopBarWidget ()
            - PageView
                - AllCourses()
                    - TabBar
                    - TabBarView
                        - Column
                            - Card
                            - Card
                - MicroInformation()
                - Favorite()
                - Settings()
            - GetBottomNavBar()
            - FloatingButton

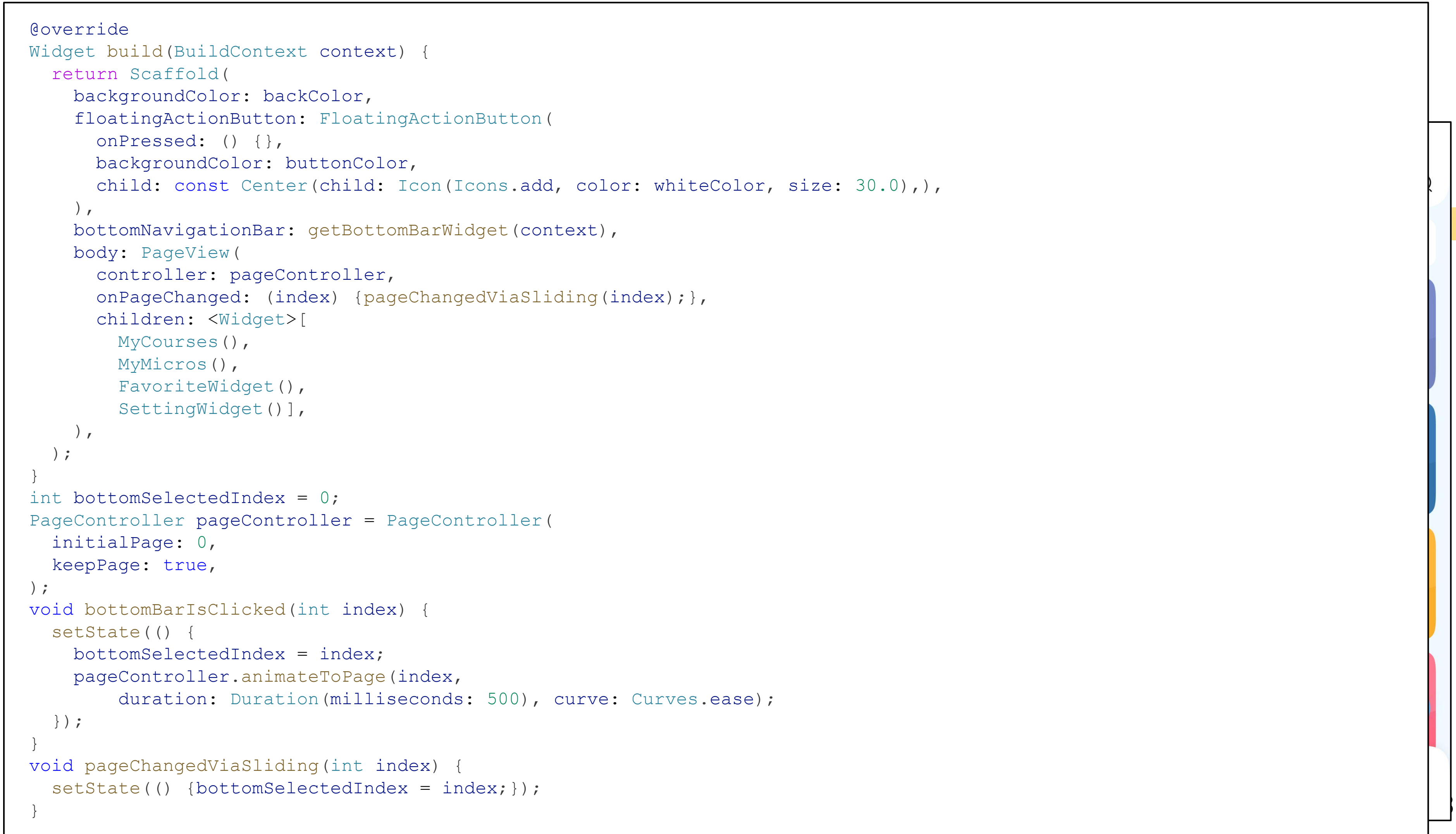


Linear Algebra & Geometry
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 10 75%

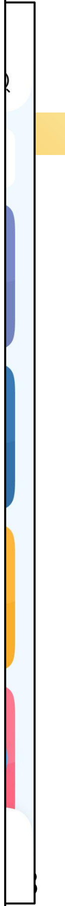
Human anatomy
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 15 35%

Organic chemistry
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 9 57%

Basic C Programs.
Lorem ipsum dolor sit amet, ose
adipiscing elit, sed do eiusmod
Chapters 14 76%

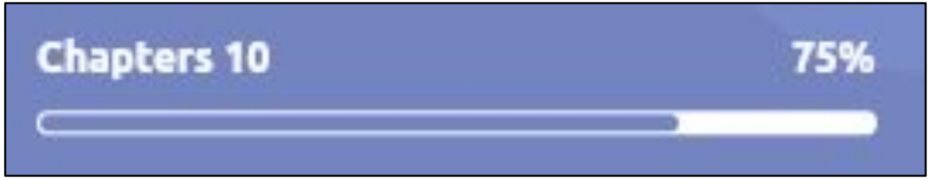


```
@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: backColor,
    floatingActionButton: FloatingActionButton(
      onPressed: () {},
      backgroundColor: buttonColor,
      child: const Center(child: Icon(Icons.add, color: whiteColor, size: 30.0),),
    ),
    bottomNavigationBar: getBottomBarWidget(context),
    body: PageView(
      controller: pageController,
      onPageChanged: (index) {pageChangedViaSliding(index);},
      children: <Widget>[
        MyCourses(),
        MyMicros(),
        FavoriteWidget(),
        SettingWidget()],
    ),
  );
}
int bottomSelectedIndex = 0;
PageController pageController = PageController(
  initialPage: 0,
  keepPage: true,
);
void bottomBarIsClicked(int index) {
  setState(() {
    bottomSelectedIndex = index;
    pageController.animateToPage(index,
        duration: Duration(milliseconds: 500), curve: Curves.ease);
  });
}
void pageChangedViaSliding(int index) {
  setState(() {bottomSelectedIndex = index;});
}
```

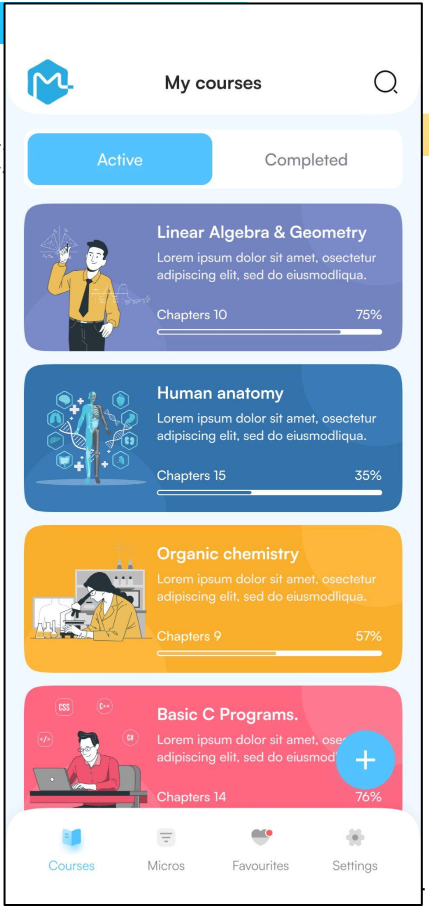


Building Your MVP using Flutter

- Widgets to Build using Flutter



Chapters 10 75%



My courses

Active Completed

Linear Algebra & Geometry

Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.

Chapters 10 75%

Human anatomy

Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.

Chapters 15 35%

Organic chemistry

Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.

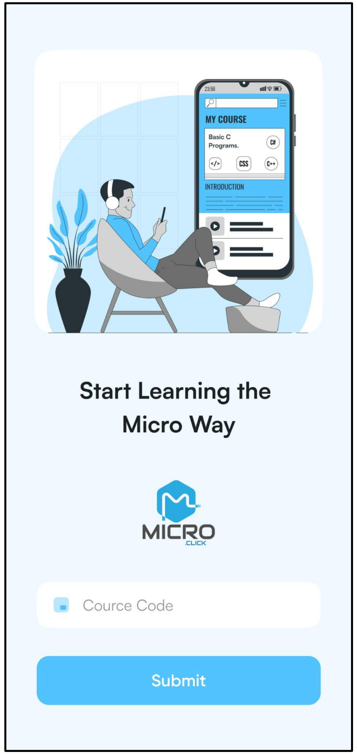
Chapters 9 57%

Basic C Programs.

Lorem ipsum dolor sit amet, ose adipiscing elit, sed do eiusmod

Chapters 14 76%

Building Your MVP using Flutter

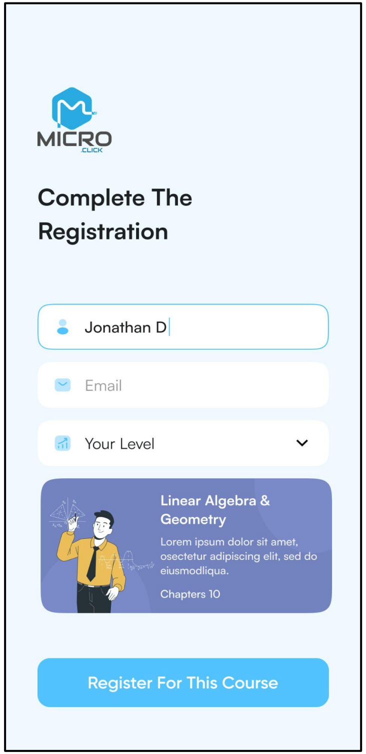


Start Learning the Micro Way

MICRO .CLICK

Cource Code

Submit


MICRO .CLICK

Complete The Registration

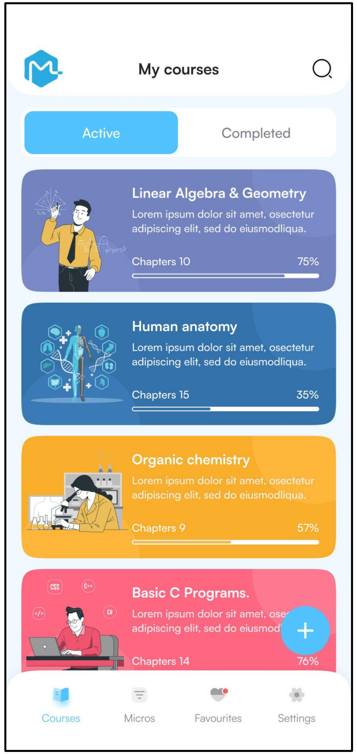
Jonathan D

Email

Your Level

Linear Algebra & Geometry
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 10

Register For This Course


My courses

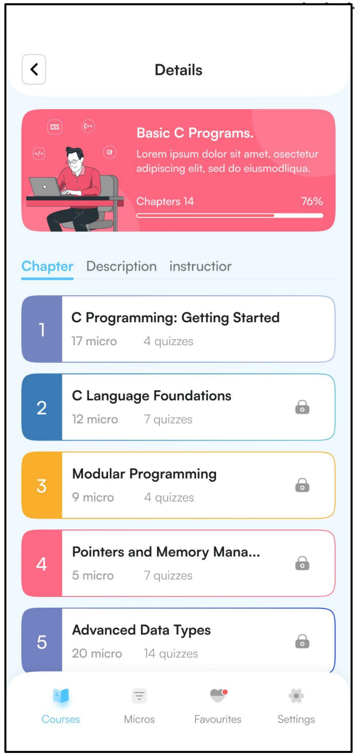
Active | Completed

Linear Algebra & Geometry
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 10 75%

Human anatomy
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 15 35%

Organic chemistry
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 9 57%

Basic C Programs.
Lorem ipsum dolor sit amet, ose adipiscing elit, sed do eiusmod
Chapters 14 76%


Details

Basic C Programs.
Lorem ipsum dolor sit amet, osectetur adipiscing elit, sed do eiusmodliqua.
Chapters 14 76%

Chapter | Description | instructior

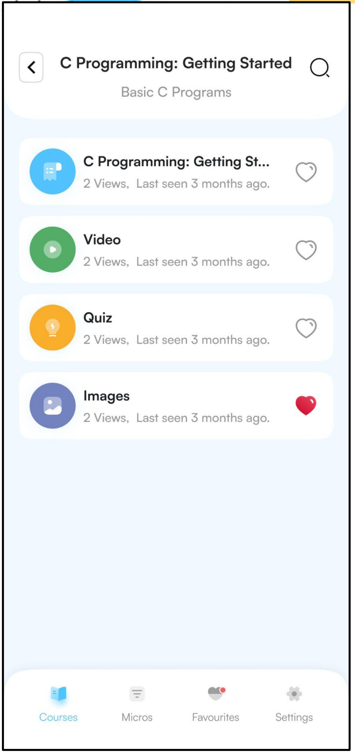
1 C Programming: Getting Started
17 micro 4 quizzes

2 C Language Foundations
12 micro 7 quizzes

3 Modular Programming
9 micro 4 quizzes

4 Pointers and Memory Mana...
5 micro 7 quizzes

5 Advanced Data Types
20 micro 14 quizzes


C Programming: Getting Started
Basic C Programs

C Programming: Getting St...
2 Views, Last seen 3 months ago.

Video
2 Views, Last seen 3 months ago.

Quiz
2 Views, Last seen 3 months ago.

Images
2 Views, Last seen 3 months ago.

Section 3

Adding Interactivity to MVP

Adding Interactivity to MVP

- **Connecting to the Network via HTTP**
  - You may need to connect to a web server via http to get data, verify, conduct further business logic.

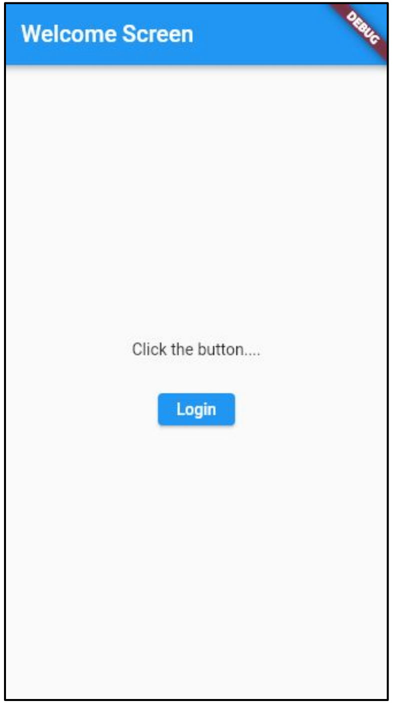
Adding Interactivity to MVP

- **Connecting to the Network via HTTP**

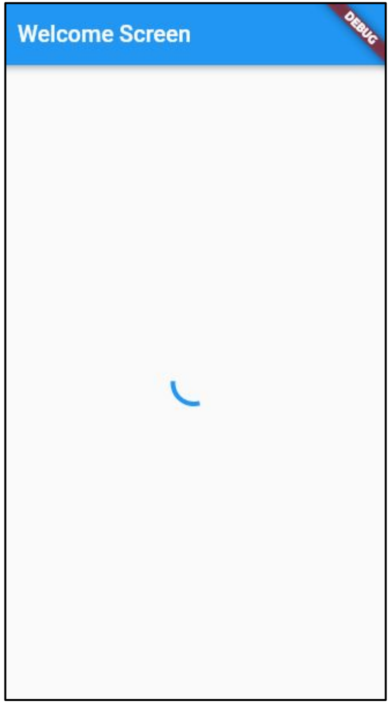
```
 void processRegister()  async {

  final response = await http.get(Uri.parse('http://api.micro.click/?action=register' ));

   if (response.statusCode == 200) {
     Map ret = jsonDecode(response.body);
     // do some processing here..

   } else {
     // If the server did not return a 200 OK response,
      // do some other actions
   }
}
```

Two Packages you need : http and convert

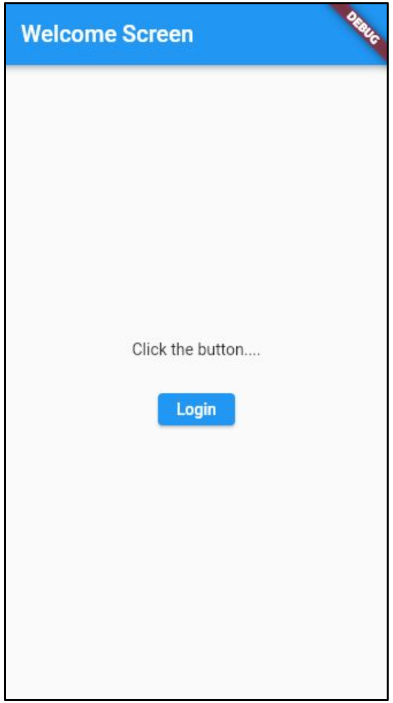
Adding Interactivity to MVP

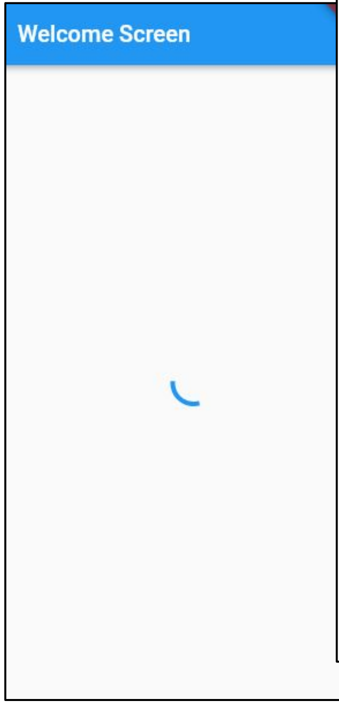


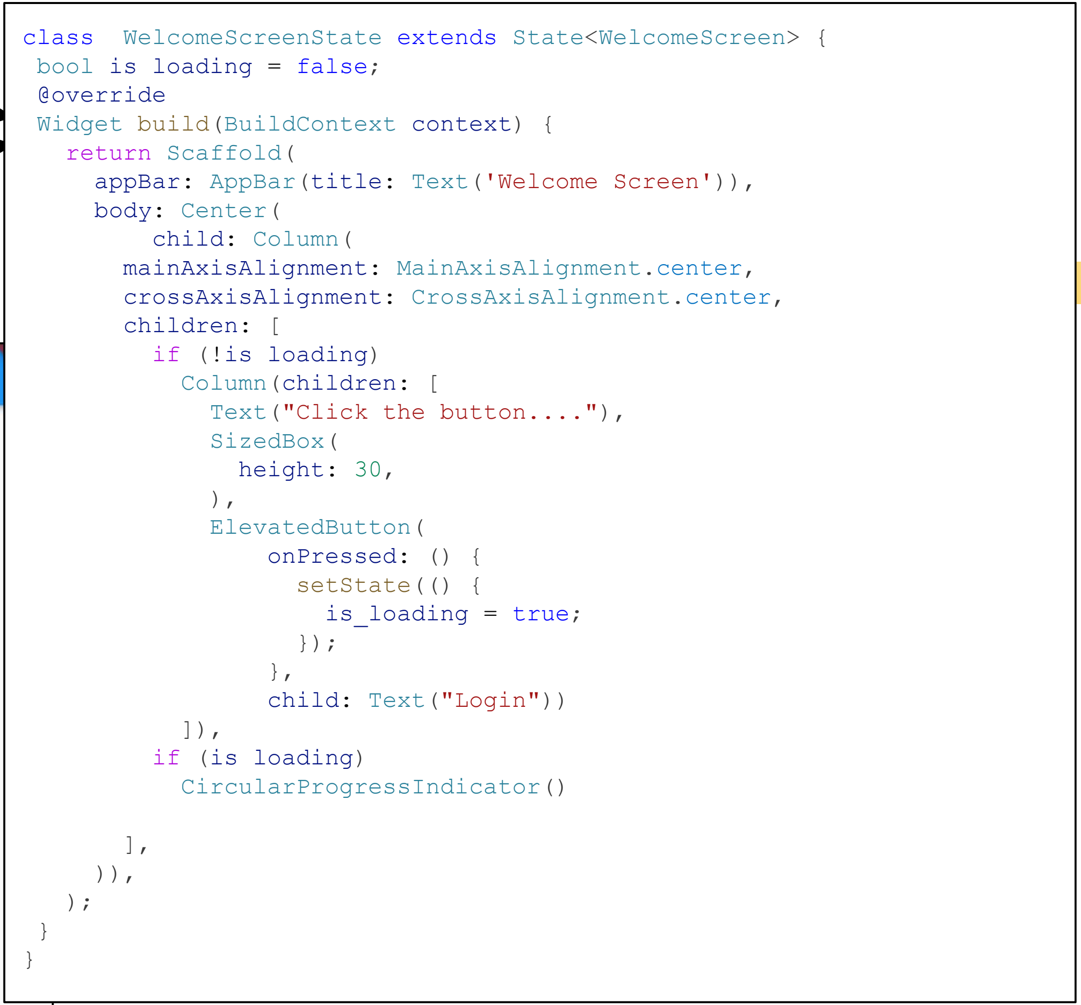
Click the button....



Adding Interac
MVP



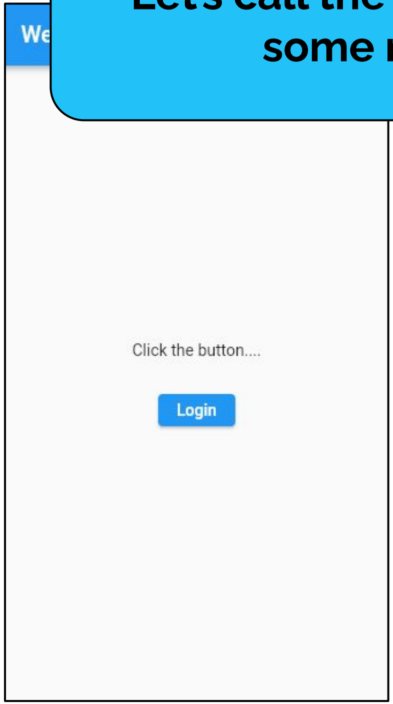


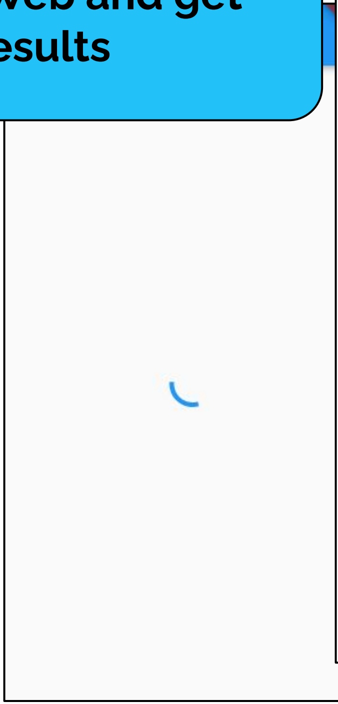


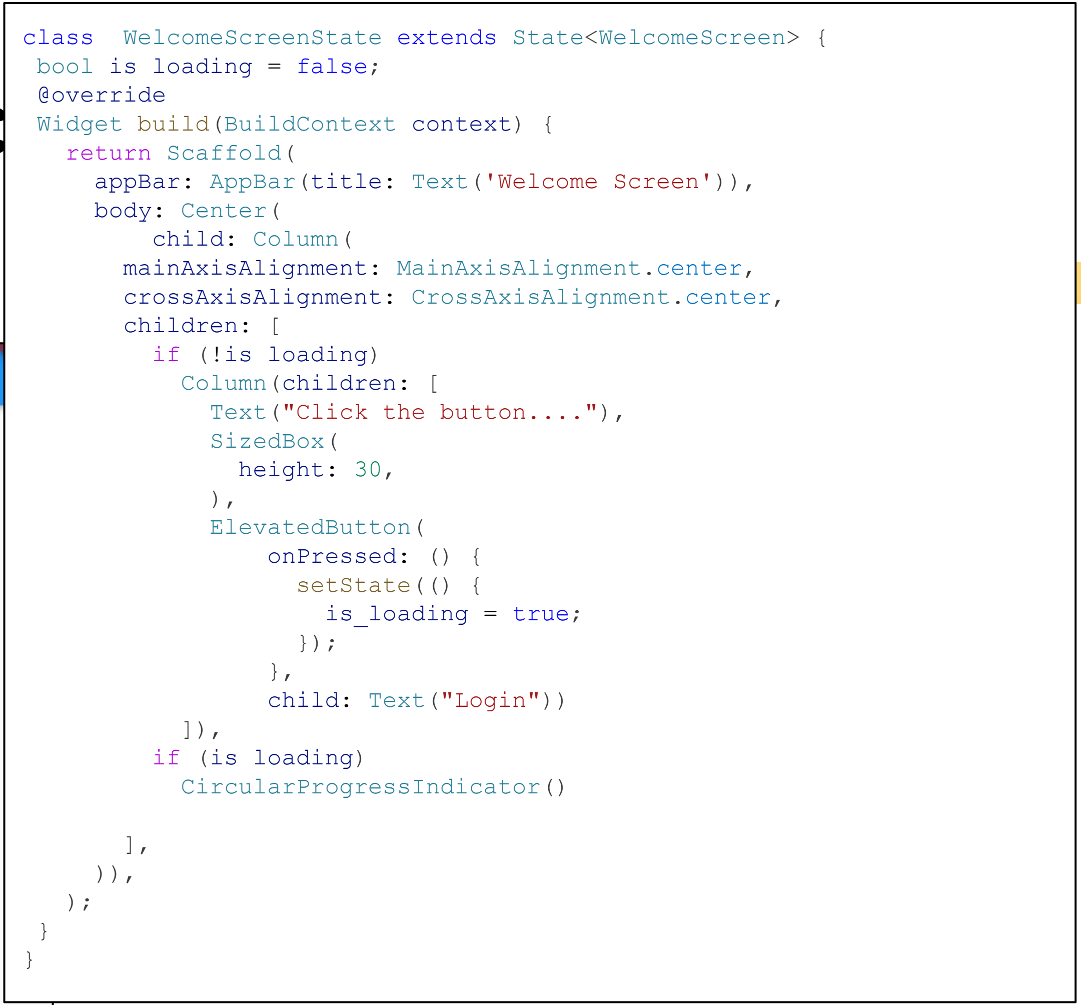
```
class  WelcomeScreenState extends State<WelcomeScreen> {
 bool is loading = false;
 @override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(title: Text('Welcome Screen')),
     body: Center(
         child: Column(
       mainAxisAlignment: MainAxisAlignment.center,
       crossAxisAlignment: CrossAxisAlignment.center,
       children: [
         if (!is loading)
           Column(children: [
             Text("Click the button...."),
             SizedBox(
               height: 30,
             ),
             ElevatedButton(
                 onPressed: () {
                   setState(() {
                     is_loading = true;
                   });
                 },
                 child: Text("Login"))
           ]),
         if (is loading)
           CircularProgressIndicator()

       ],
     )),
   );
 }
}
```

Adding Interac
MVP

Let's call the Web and get some results

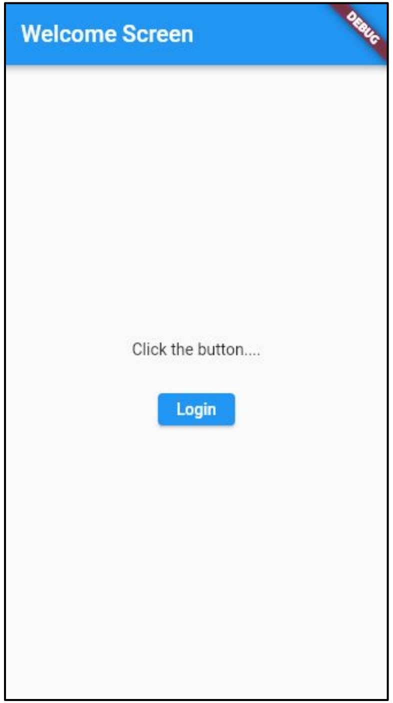


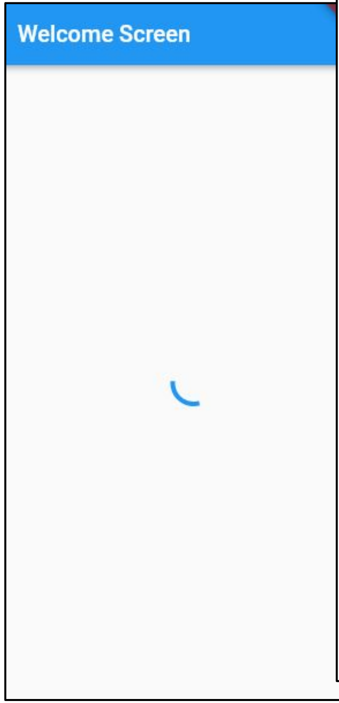


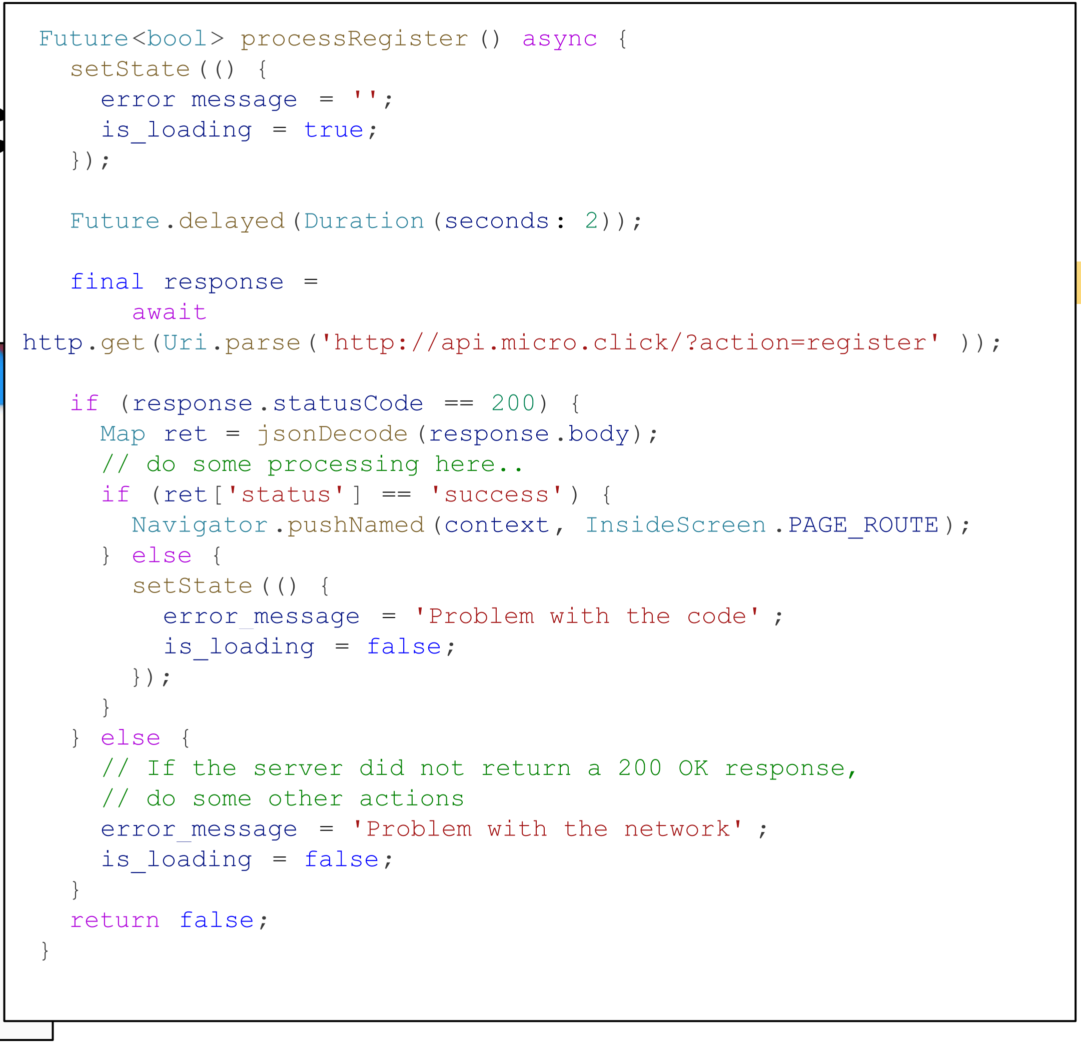


```
class  WelcomeScreenState extends State<WelcomeScreen> {
 bool is loading = false;
 @override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(title: Text('Welcome Screen')),
     body: Center(
         child: Column(
       mainAxisAlignment: MainAxisAlignment.center,
       crossAxisAlignment: CrossAxisAlignment.center,
       children: [
         if (!is loading)
           Column(children: [
             Text("Click the button...."),
             SizedBox(
               height: 30,
             ),
             ElevatedButton(
                 onPressed: () {
                   setState(() {
                     is_loading = true;
                   });
                 },
                 child: Text("Login"))
           ]),
         if (is loading)
           CircularProgressIndicator()

       ],
     )),
   );
 }
}
```

Adding Interac
MVP



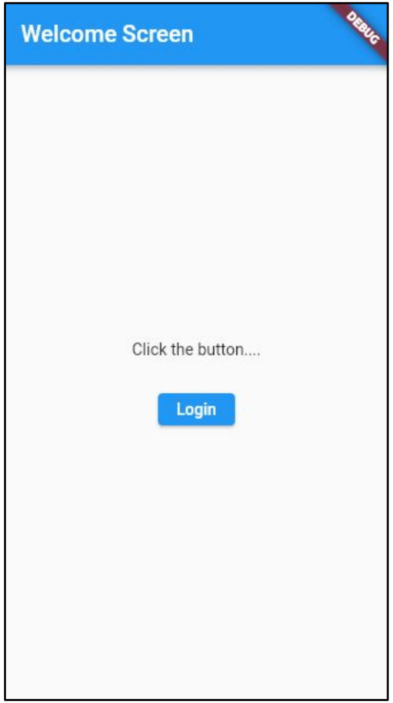




```
Future<bool> processRegister() async {
  setState(() {
    error message = '';
    is_loading = true;
  });

  Future.delayed(Duration(seconds: 2));

  final response =
      await
http.get(Uri.parse('http://api.micro.click/?action=register' ));

  if (response.statusCode == 200) {
    Map ret = jsonDecode(response.body);
    // do some processing here..
    if (ret['status'] == 'success') {
      Navigator.pushNamed(context, InsideScreen.PAGE_ROUTE);
    } else {
      setState(() {
        error_message = 'Problem with the code' ;
        is_loading = false;
      });
    }
  } else {
    // If the server did not return a 200 OK response,
    // do some other actions
    error_message = 'Problem with the network' ;
    is_loading = false;
  }
  return false;
}
```

Adding Interactivity to MVP

The data on the second screen is fetched from the web

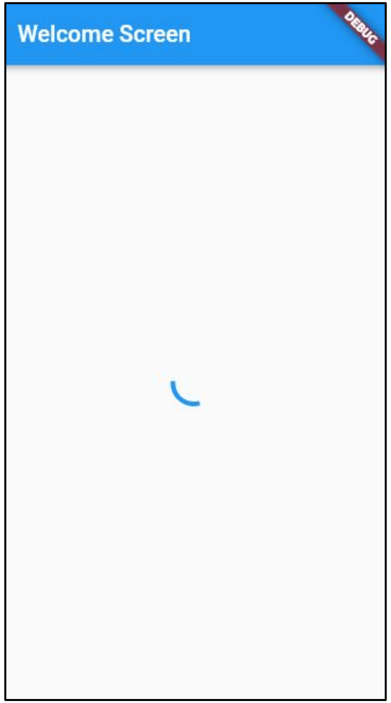


Welcome Screen

DEBUG

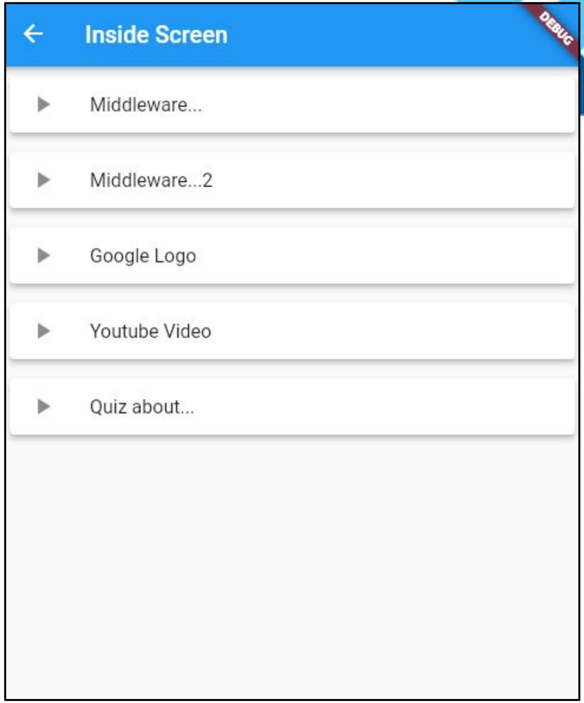
Click the button....

Login



Welcome Screen

DEBUG

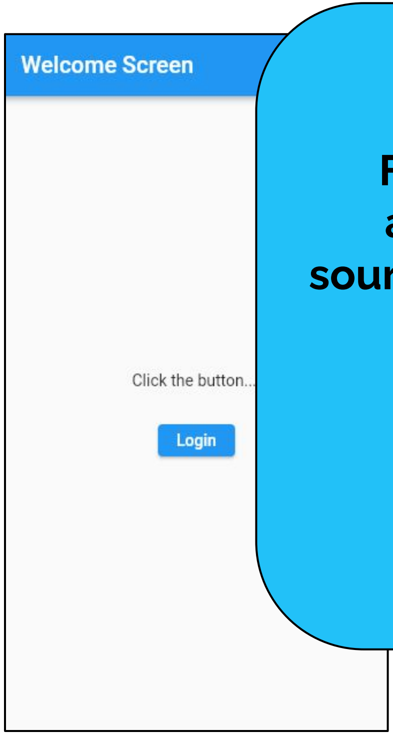


Inside Screen

DEBUG

- Middleware...
- Middleware...2
- Google Logo
- Youtube Video
- Quiz about...

Adding Interactivity to MVP

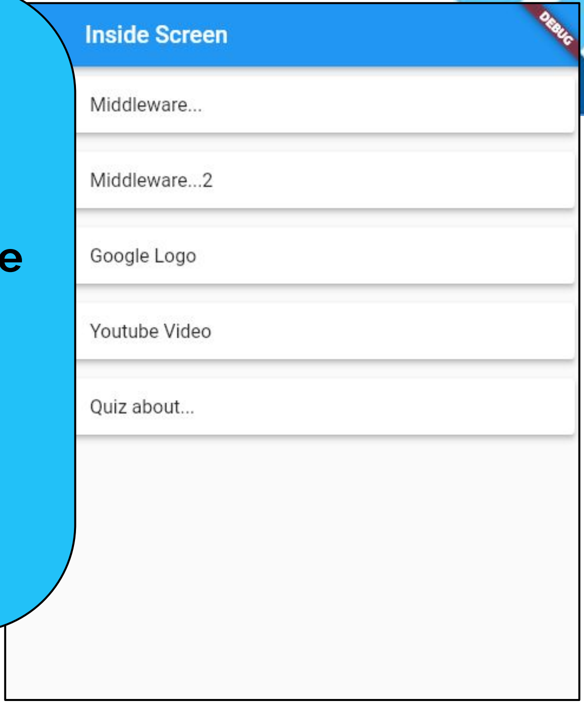


Welcome Screen

Click the button...

For your final App, Widgets will be always pulling data from “future” sources including databases, Files or the web.

Rarely, from variables that are “immediately” available.



Inside Screen

Middleware...

Middleware...2

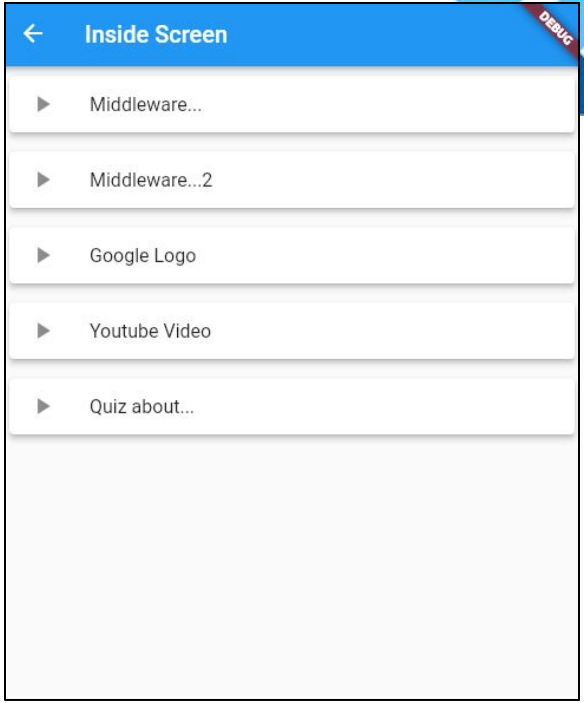
Google Logo

Youtube Video

Quiz about...

```dart
import 'package:flutter/material.dart';
import
'package:register_web_progress_bar/screens/insidescreen.dart'
;

import 'screens/welcome.dart';

void main() {
 runApp(const MainApp());
}

class MainApp extends StatelessWidget {
 const MainApp({super.key});

 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     home: WelcomeScreen(),
     routes: {
       WelcomeScreen.PAGE_ROUTE: (ctx) => WelcomeScreen(),
       InsideScreen.PAGE_ROUTE: (ctx) => InsideScreen(),
     },
   );
 }
}
```

The data on the second screen is fetched from the web



Inside Screen

▶ Middleware...

▶ Middleware...2

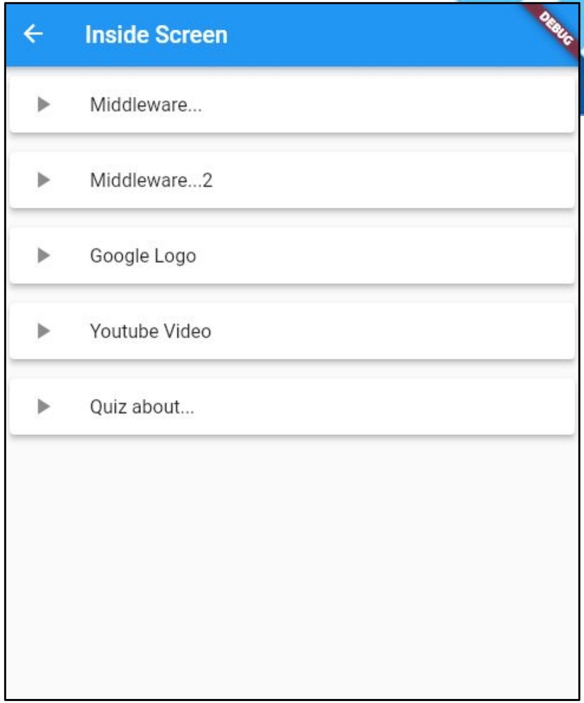
▶ Google Logo

▶ Youtube Video

▶ Quiz about...

```
import 'package:flutter/material.dart' ;
import 'package:http/http.dart'  as http;
import 'dart:convert' ;

class InsideScreen extends StatefulWidget {
 static const String PAGE_ROUTE = '/inside';
 const InsideScreen({super.key});
 @override
 State<InsideScreen> createState() => _InsideScreenState();
}

class _InsideScreenState extends State<InsideScreen> {
 Future<Map<String, dynamic>>? courseData;
 @override
 void initState() {
   super.initState();
   courseData = getCourseDataFromWeb();
 }

 Future<Map<String, dynamic>> getCourseDataFromWeb() async {
   await Future.delayed(Duration(seconds: 2));
   Map<String, dynamic> ret = {};
   final response = await http

.get(Uri.parse('http://micro.click/api/?action=get_information'));

   if (response.statusCode == 200) {
     ret = jsonDecode(response.body);
   }
   return ret;
 }
```

The data on the second screen is fetched from the web



← Inside Screen

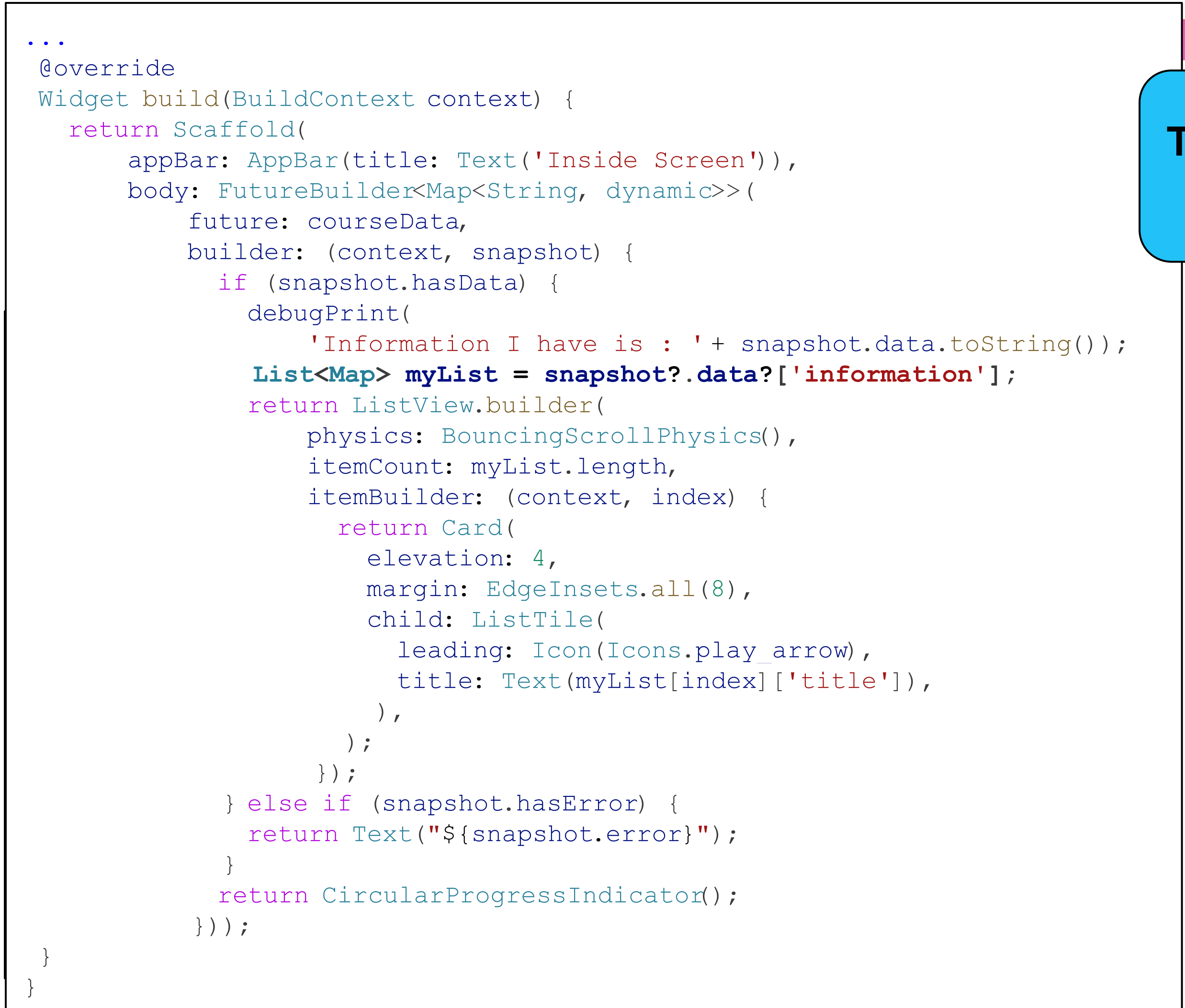
▶ Middleware...

▶ Middleware...2

▶ Google Logo

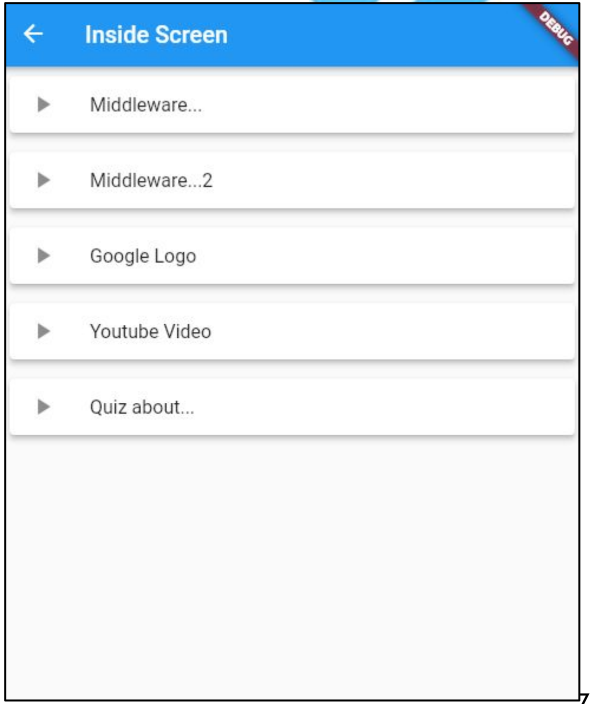
▶ Youtube Video

▶ Quiz about...



```
...
@override
Widget build(BuildContext context) {
  return Scaffold(
      appBar: AppBar(title: Text('Inside Screen')),
      body: FutureBuilder<Map<String, dynamic>>(
          future: courseData,
          builder: (context, snapshot) {
            if (snapshot.hasData) {
              debugPrint(
                  'Information I have is : ' + snapshot.data.toString());
              List<Map> myList = snapshot?.data?['information'];
              return ListView.builder(
                  physics: BouncingScrollPhysics(),
                  itemCount: myList.length,
                  itemBuilder: (context, index) {
                    return Card(
                      elevation: 4,
                      margin: EdgeInsets.all(8),
                      child: ListTile(
                        leading: Icon(Icons.play_arrow),
                        title: Text(myList[index]['title']),
                      ),
                    );
                  });
            } else if (snapshot.hasError) {
              return Text("${snapshot.error}");
            }
            return CircularProgressIndicator();
          }));
}
}
```

The data on the second screen is fetched from the web



Lecture Demo Apps

- All Demos : https://www.dropbox.com/scl/fo/tj08l6x4fwattumcyu5e5/h?rlkey=emt84ac8pps5n9dho2utsrayj&dl=0
- State Management using setState:
  - Multiple Screens : https://www.dropbox.com/scl/fo/1ha4ct3zn2731ratvlqy4/h?rlkey=h39vhbranvii40j44ne02mdr3&dl=0
  - Single Screen and Multiple Widgets https://www.dropbox.com/scl/fo/czrgwzxrhghth9yeyzsyn/h?rlkey=q5lrl7i6pzer3lz5fhubjy8tg&dl=0
  - Using Asyinc and Await : https://www.dropbox.com/scl/fo/6o11yvrald1msuo827l4w/h?rlkey=wt2uan99n89wwibax8in12916&dl=0
- GetX:
  - Single Screen and Multiple Widgets https://www.dropbox.com/scl/fo/88wxuexv8ffc6jcls5lsl/h?rlkey=j3s0ywimjzx69h5bvsokd915r&dl=0
  - Multiple Screens https://www.dropbox.com/scl/fo/kglaujuaruk0ber1u017t/h?rlkey=q0podfzqyeumtm7nvennx4fkk&dl=0
  - GetView https://www.dropbox.com/scl/fo/0qx2dsrg7lklccmv7sf0o/h?rlkey=qyehdbj59z8avuvqqm17vx4j6&dl=0
- Provider
  - https://www.dropbox.com/scl/fo/56orr5tnojgqouw2cvwql/h?rlkey=60ag6crnqp7umirljfmklmwf6&dl=0
- MVP
  - https://www.dropbox.com/scl/fo/oa01zhwif4cerdwvsnicr/h?rlkey=a5xkw9g0fyp5wemwv8cz4i8h1&dl=0
- FutureBuilder :
  - https://www.dropbox.com/scl/fo/upd88fqz3sfwn2renhtue/h?rlkey=6kjdvj1pt15h5fyo24nxjv2go&dl=0

Resources

- https://docs.flutter.dev/data-and-backend/state-mgmt/intro
- https://pub.dev/packages/get
- https://karthikponnam.medium.com/flutter-pageview-withbottomnavigationbar-fb4c87580f6a
- https://stackoverflow.com/questions/49991444/create-a-rounded-button-button-with-border-radius-in-flutter
- https://www.dbestech.com/tutorials/flutter-getx-app-tutorial