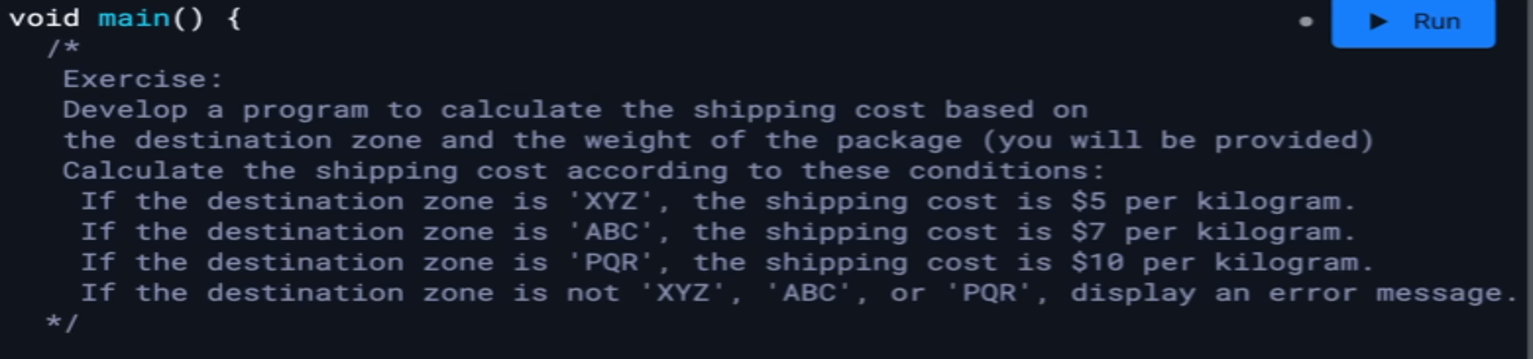
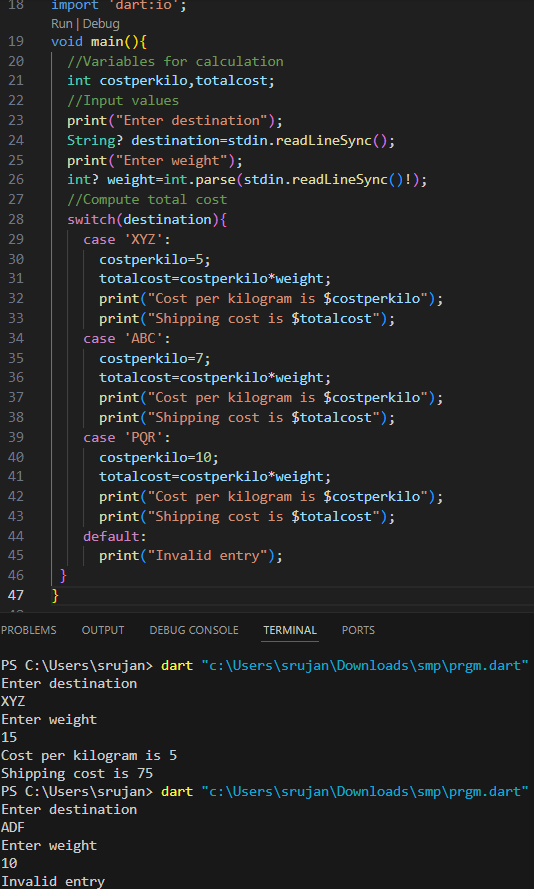
Day 1 Assignment

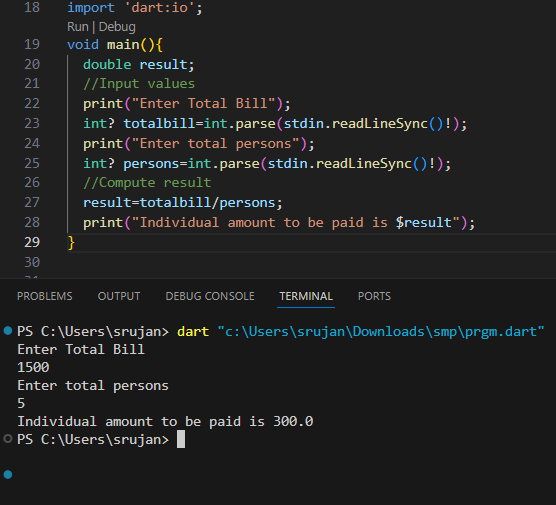
Code 1





Code 2

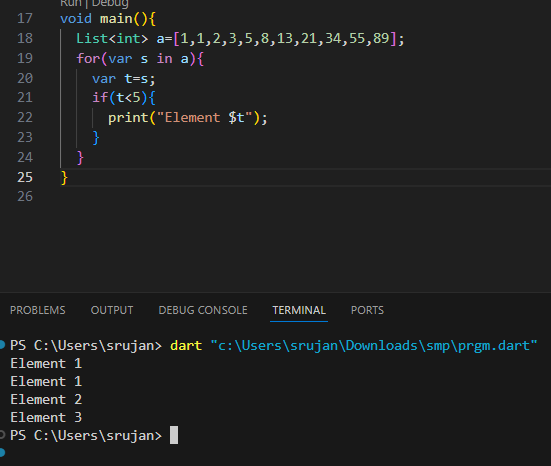
Suppose, you often go to restaurant with friends and you have to split amount of bill. Write a program to calculate split amount of bill. Formula= (total bill amount) / number of people



Code 3

Take a list, say for example this one:

a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] and write a program that prints out all the elements of the list that are less than 5.

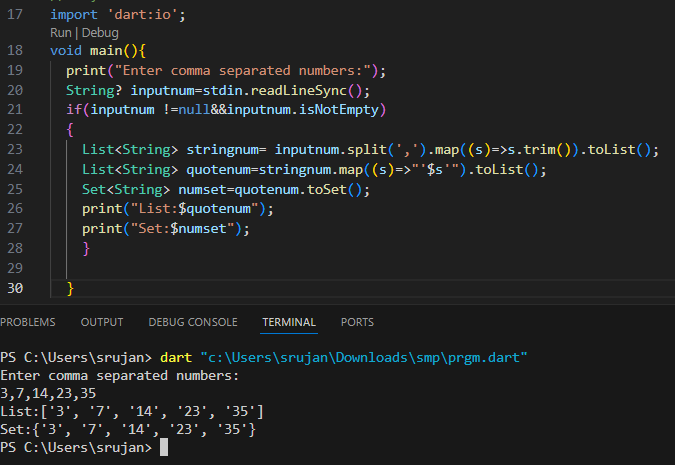


Code 4

Write a Dart program which accepts a sequence of comma-separated numbers from the user and generates a list and a set with those numbers.

Sample data : 3, 5, 7, 23

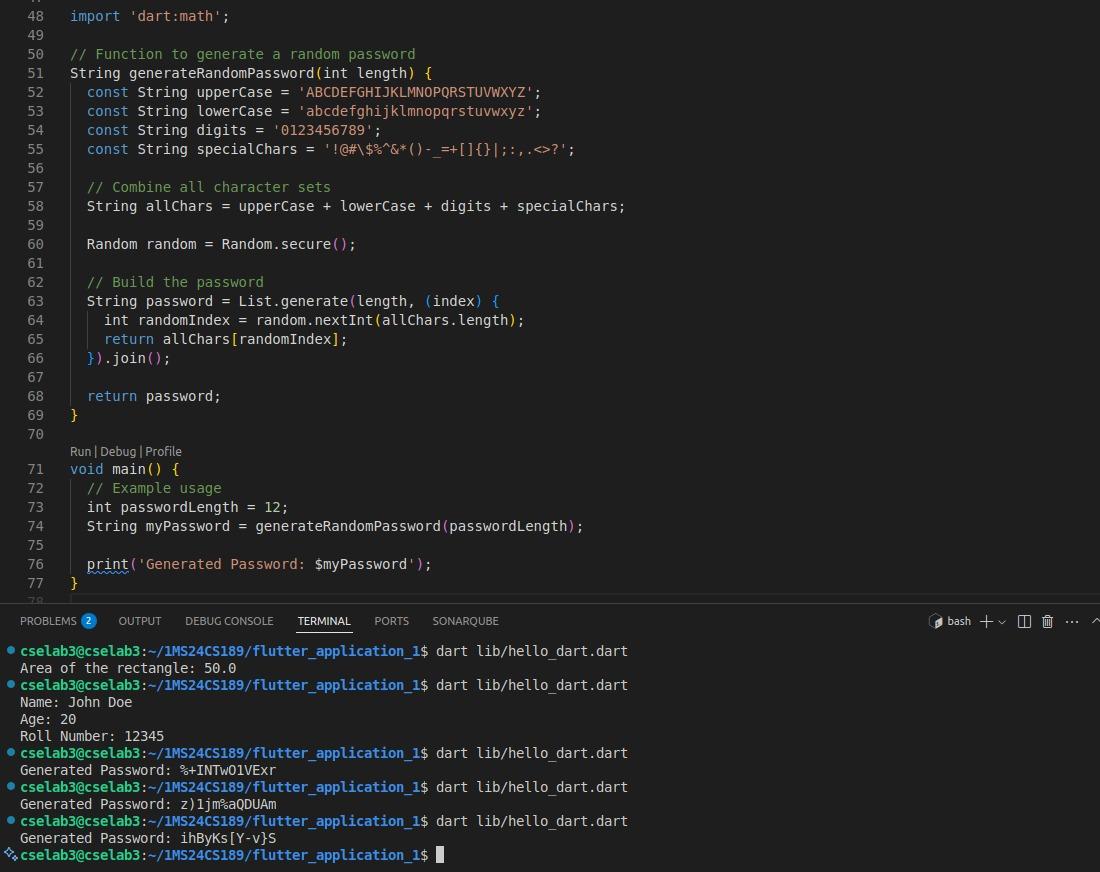
Output : List : [‘3’, ‘ 5’, ‘ 7’, ‘ 23’] Set : {‘3’, ‘ 5’, ‘ 7’, ‘ 23’}



Day 2

Code 1

Write a function in Dart that generates random password.



Code 2

You are given a list of integers. Write a Dart function that takes this list and returns the sum of all even numbers in the list. Implement this function using an anonymous function.

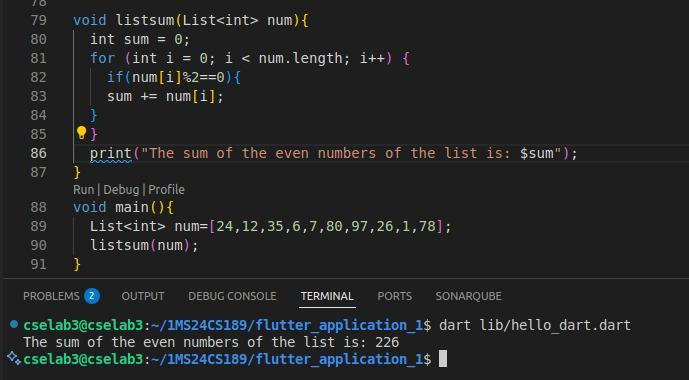
Input:

A list of integers, numbers, where 1 <= numbers.length <= 1000.

Each element of the list, num, where 1 <= num <= 10000.

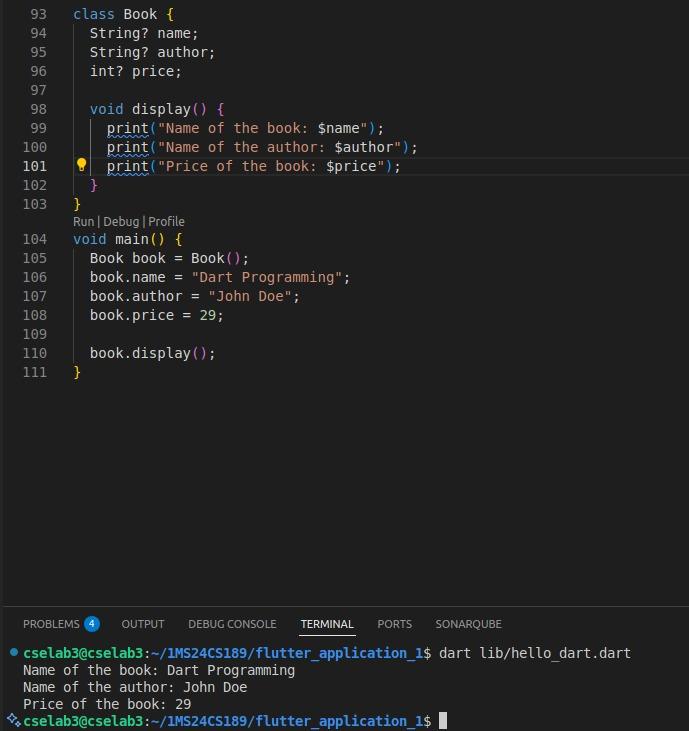
Output:

An integer representing the sum of all even numbers in the input list.



Code 3

Define a Dart class named "Book" that has three attributes: "name," "author," and "price." Additionally, implement a method named "display" within the class to print the values of these three attributes



Code 4

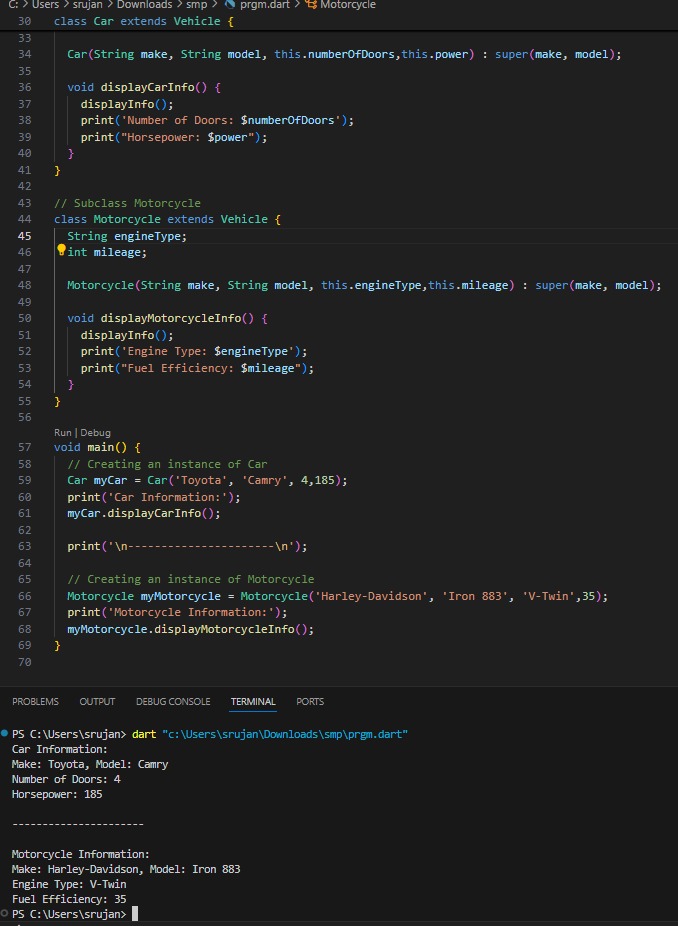
Create a program in Dart that models a simple hierarchy of vehicles. Start by defining a base class called Vehicle with properties like make, model, and a method called displayInfo that prints the make and model of the vehicle.

Then, create two subclasses: Car and Motorcycle, both inheriting from the Vehicle class. Each subclass should have additional properties and methods specific to cars and motorcycles. For example, Car might have a property for the number of doors and Motorcycle might have a property for the type of engine.

Demonstrate the use of inheritance by creating instances of Car and Motorcycle, setting their properties, and calling the displayInfo method to print the information.

Your code should showcase how inheritance allows you to reuse and extend the properties and methods of the base class in the derived classes.

Please provide the Dart code for this scenario.



Code 5

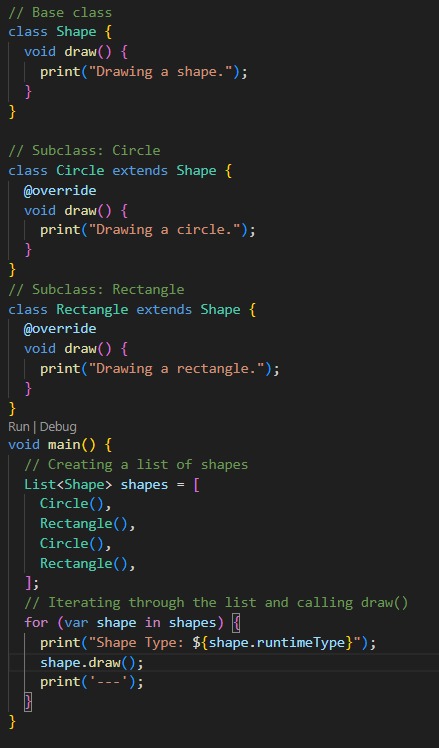
Imagine you are building a simple drawing application in Dart. You have a base class Shape with a method draw() that prints "Drawing a shape."

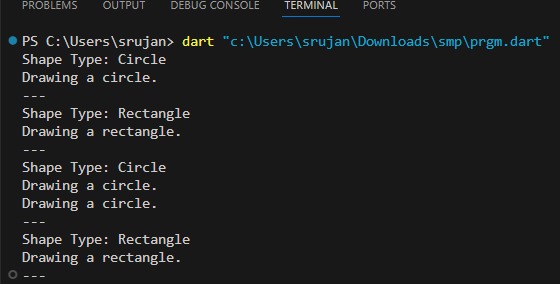
Create two subclasses: Circle and Rectangle, both inheriting from the Shape class. Implement the draw() method in each subclass to print "Drawing a circle" and "Drawing a rectangle," respectively.

Now, create a list of Shape objects that includes instances of both circles and rectangles. Iterate through the list and call the draw() method on each object. As you do so, print out the type of shape (circle or rectangle) and the message generated by the draw() method.

This exercise demonstrates polymorphism in a real-world context, where you can have a list of different shapes, and the correct draw() method is called based on the object's actual type.

Please provide the Dart code for this scenario.





Code 6

You are tasked with creating a Dart program that generates random passwords using a class-based approach. Design a class called PasswordGenerator with the following features:

A constructor that takes two parameters: the desired length of the password and a boolean flag to specify whether to include special characters.

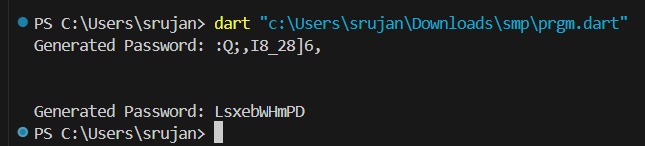
A method called generatePassword that generates a random password based on the length and special character settings. If special characters are included, the password should contain a mix of lowercase letters, uppercase letters, numbers, and special characters. If special characters are not included, the password should only contain letters and numbers.

A method called displayPassword that prints the generated password to the console.

Your task is to create the PasswordGenerator class and demonstrate its functionality. Write code to create an instance of the class, set the desired length and special character flag, generate a password, and display it.

Please provide the Dart code for this scenario.





Code 7

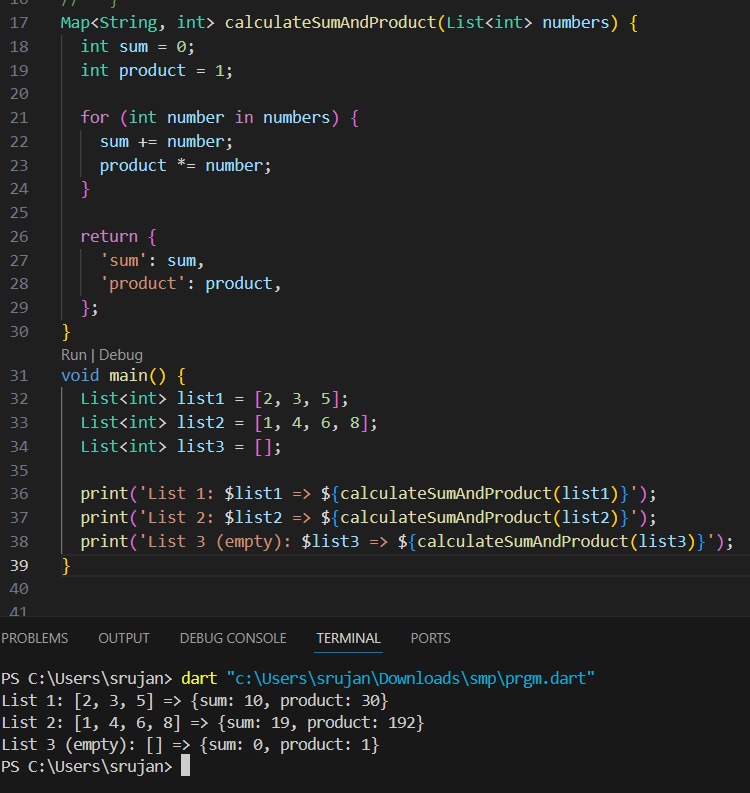
You are given a list of integers, and your task is to write a Dart function that finds the sum and product of all the elements in the list. Create a function called calculateSumAndProduct that takes a list of integers as its parameter and returns a map containing the sum and product of the list's elements.

The function should have the following signature:

Map<String, int> calculateSumAndProduct(List<int> numbers)

The calculateSumAndProduct function should return a map with two key-value pairs: "sum" and "product". For example, if the input list is [2, 3, 5], the function should return {"sum": 10, "product": 30}.

Write the Dart function and provide a usage example to demonstrate how it works with different input lists.



Code 8

Create a to-do list application in Dart. You need to design a class called TodoList to manage to-do items. The TodoList class should have the following features:

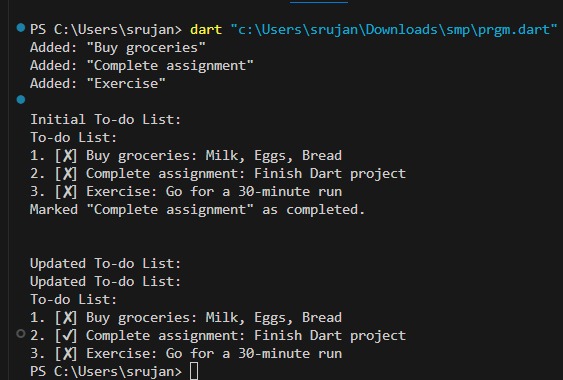
A method to add a new to-do item.A method to mark a to-do item as completed.

A method to display all to-do items, showing their status (completed or not).

Additionally, create a TodoItem class that represents individual to-do items. Each TodoItem should have properties such as a title, description, and a status flag indicating whether it is completed or not.Your task is to design the TodoList class and the TodoItem class with the specified functionality. Create an instance of the TodoList class, add some to-do items, mark some as completed, and display the list of to-do items with their statuses.

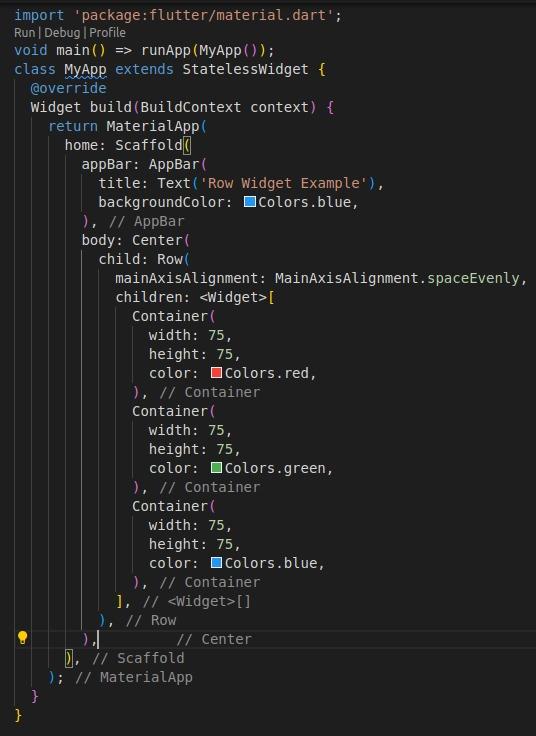


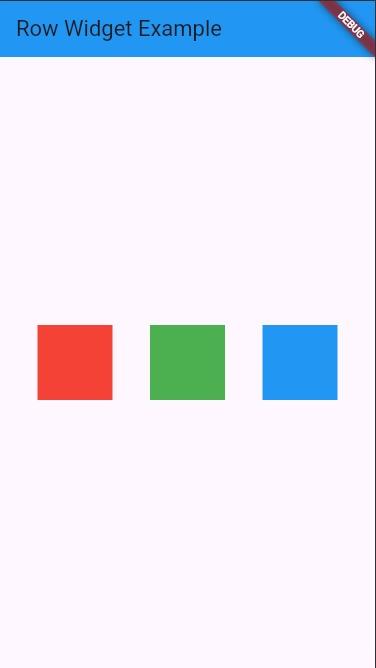




Day 3

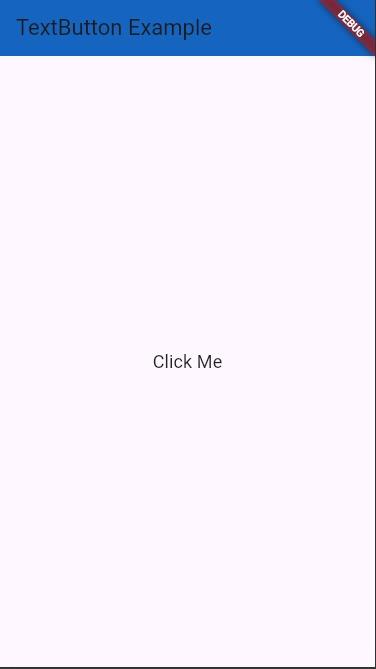
Code 1 : Row Widget



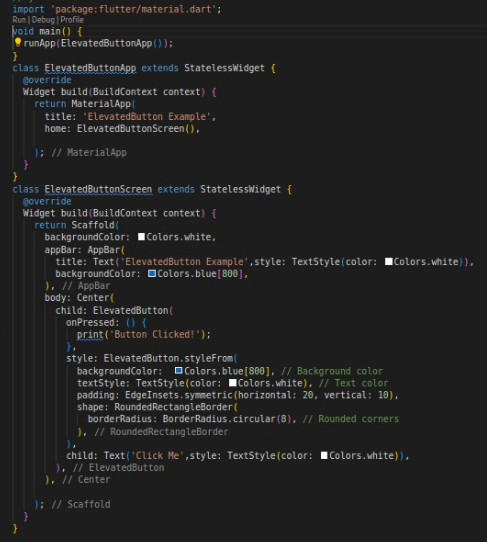


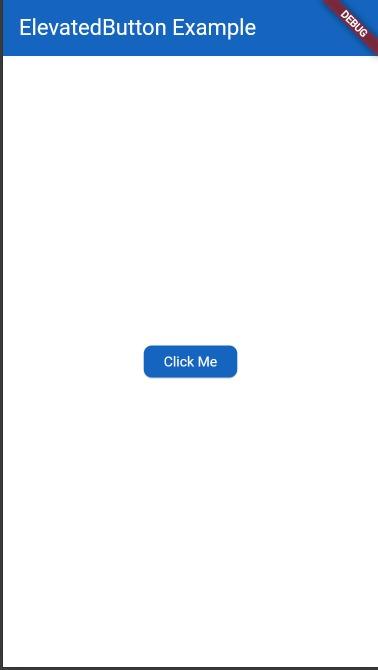
Code 2 : Text Button



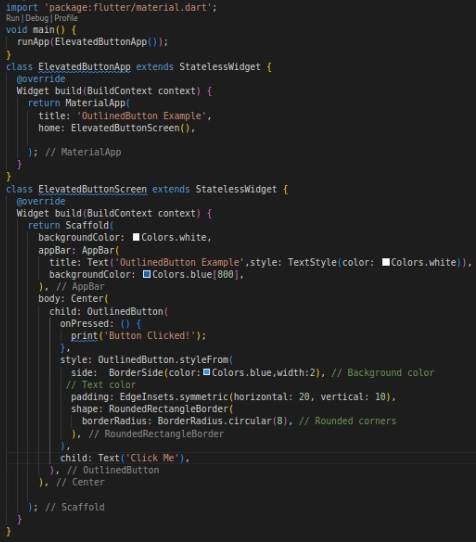


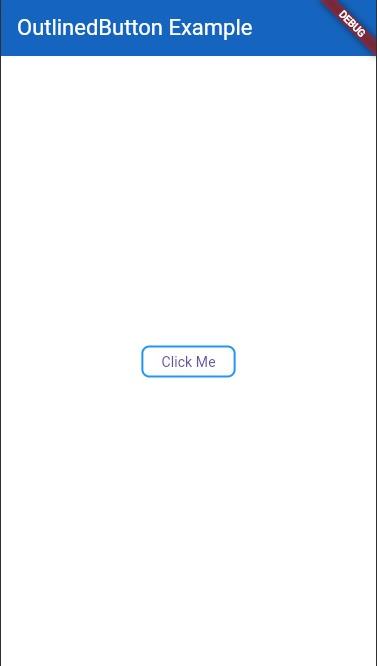
Code 3 : Elevated Button



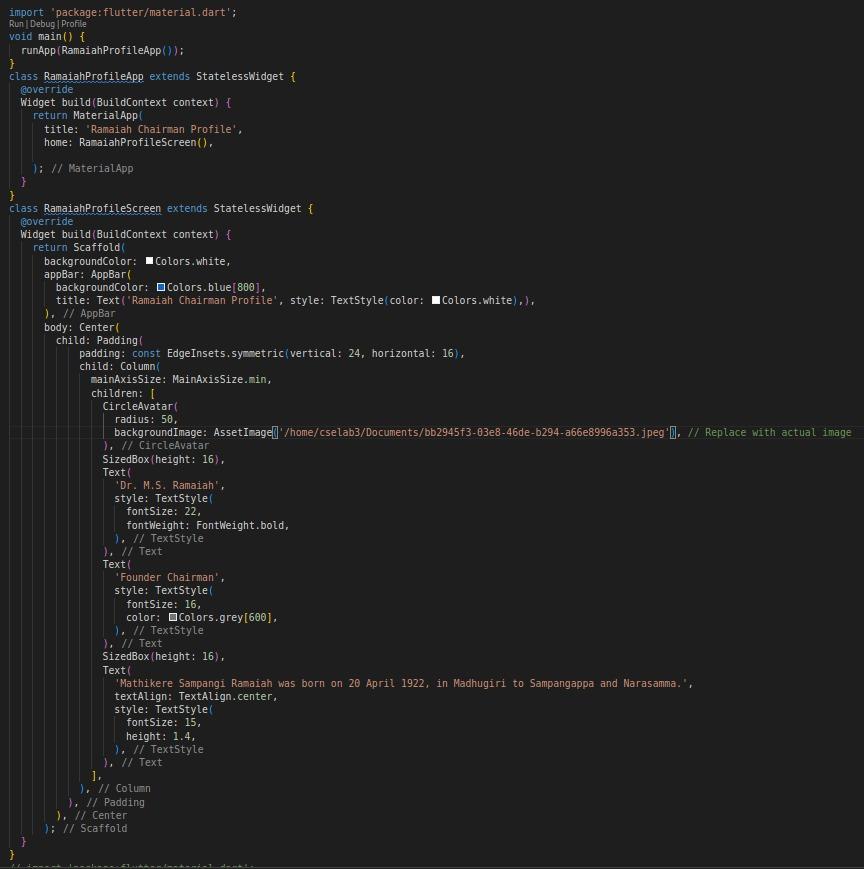


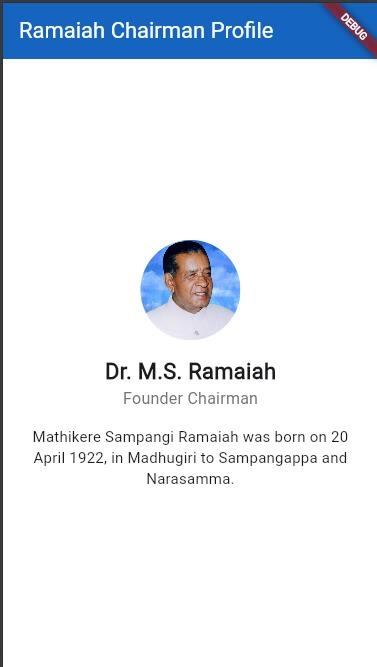
Code 4 : Outlined Button





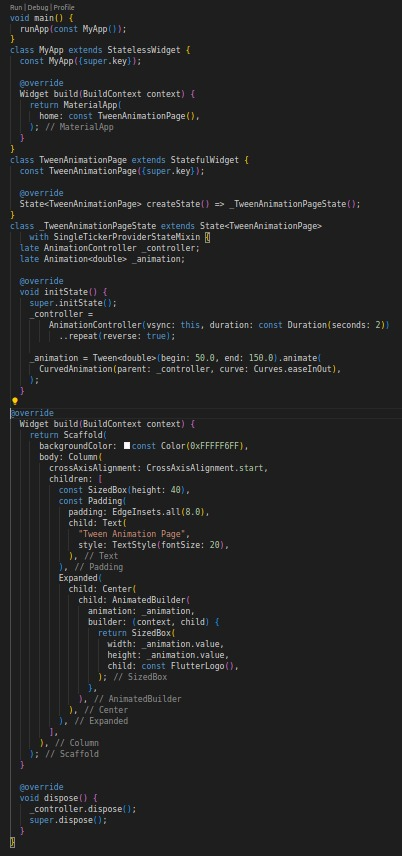
Code 5

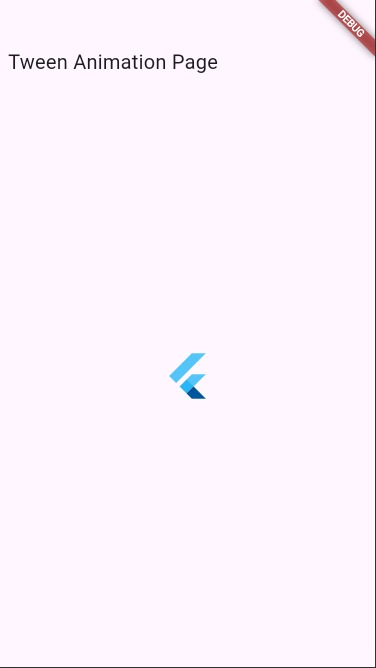




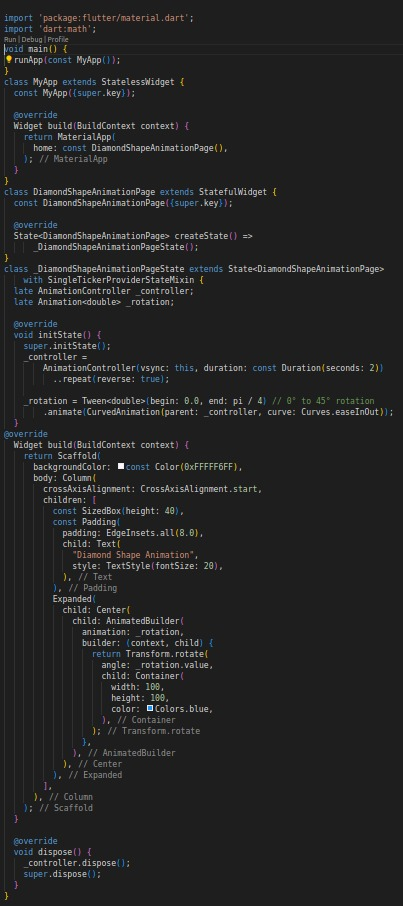
Day 4

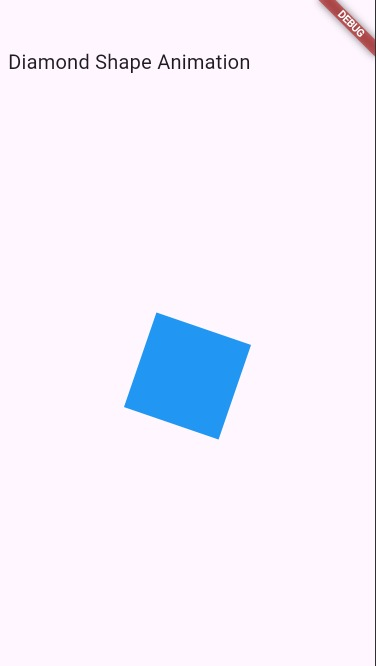
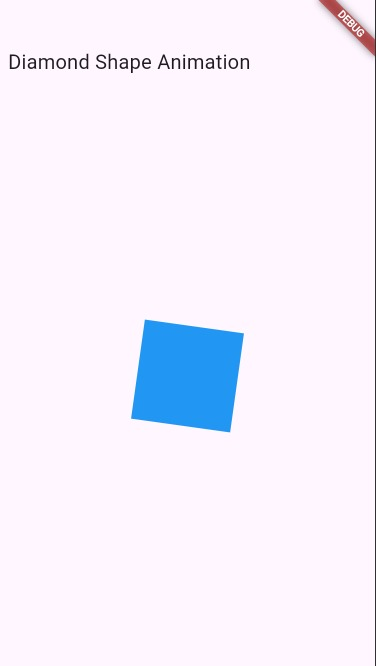
Code 1: Tween Animation



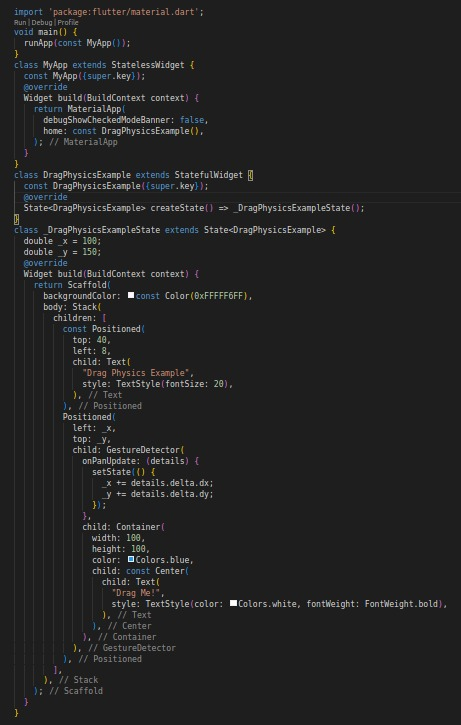
 

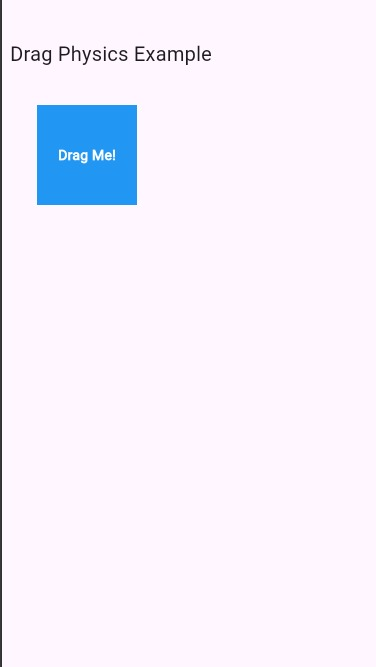
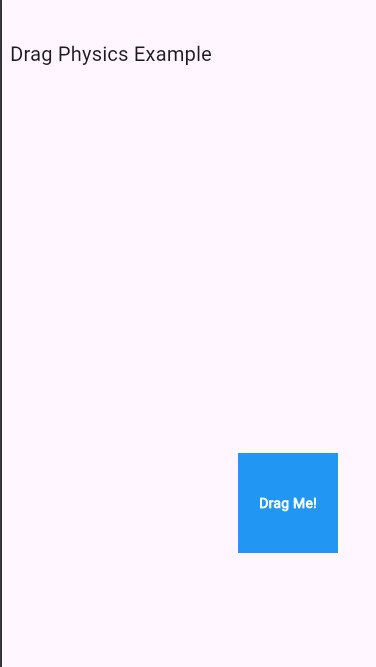
Code 2 : Diamond Shape Animation



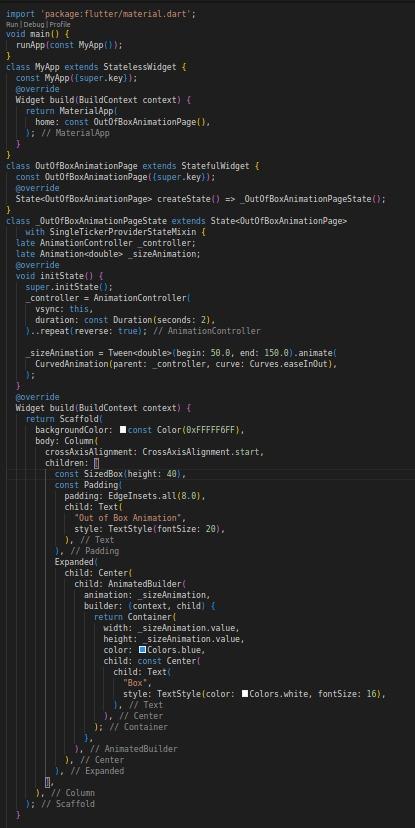
 

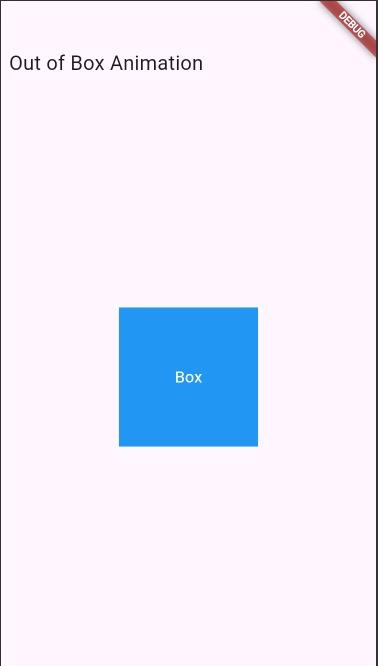
Code 3 : Drag Physics example

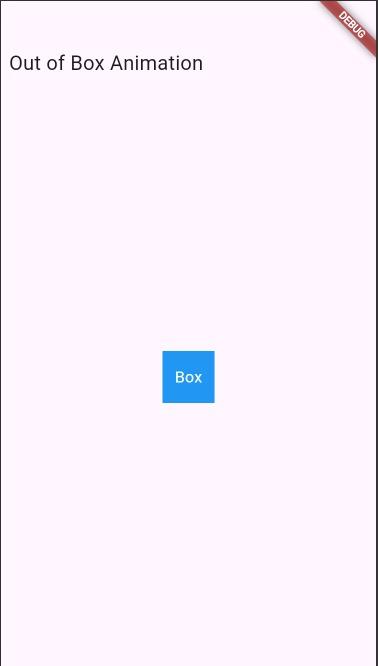


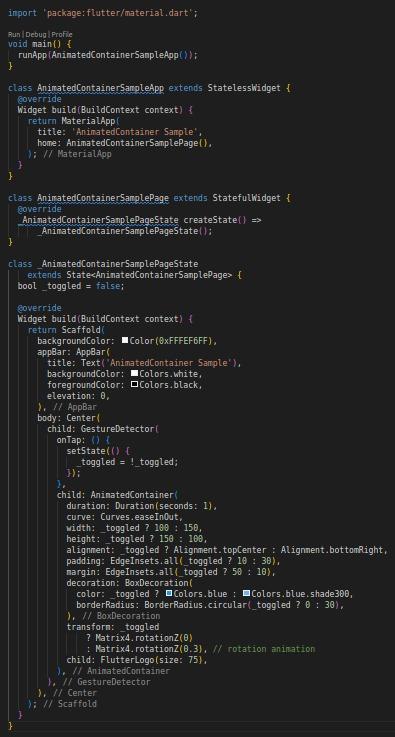
Code 4 : Out of box animation

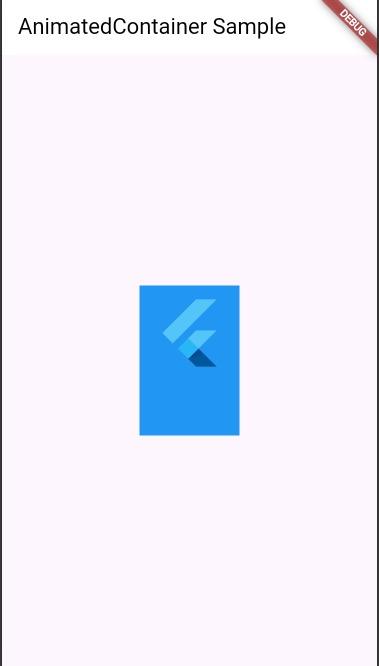
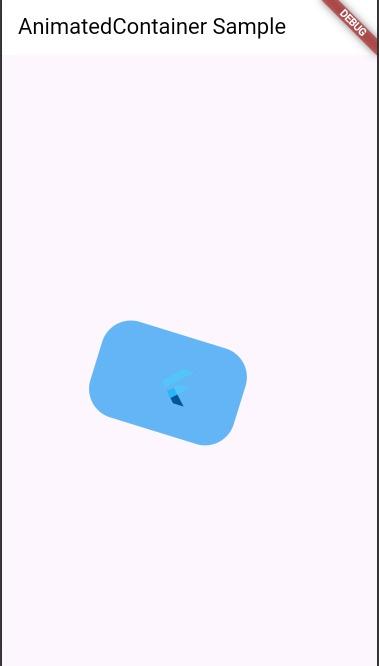




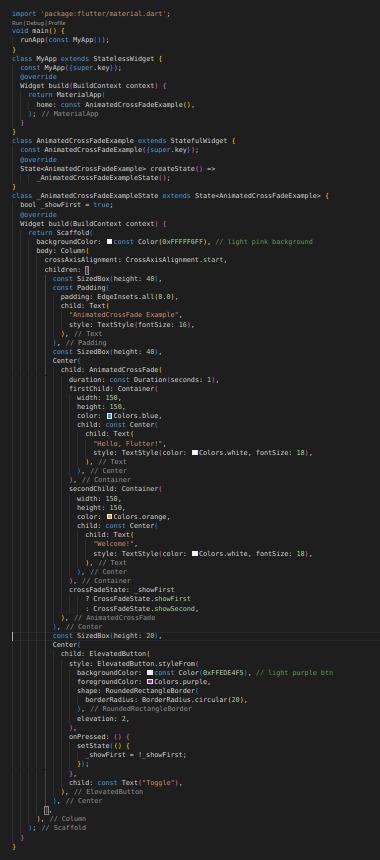


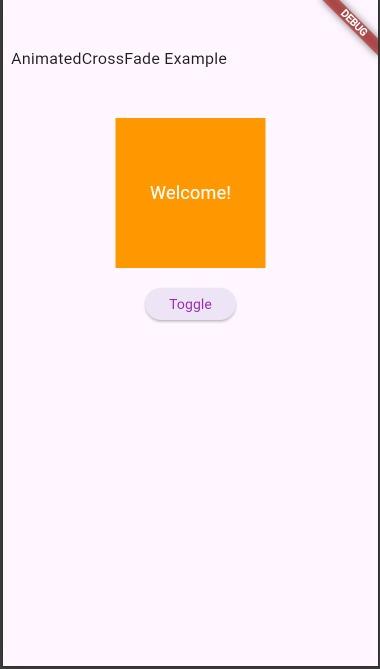
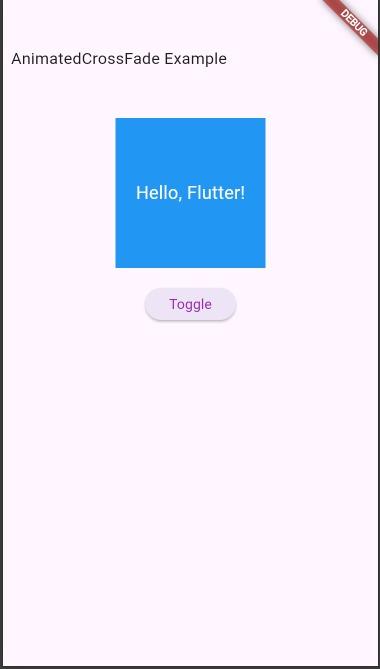
Code 5 : Animated Container Example



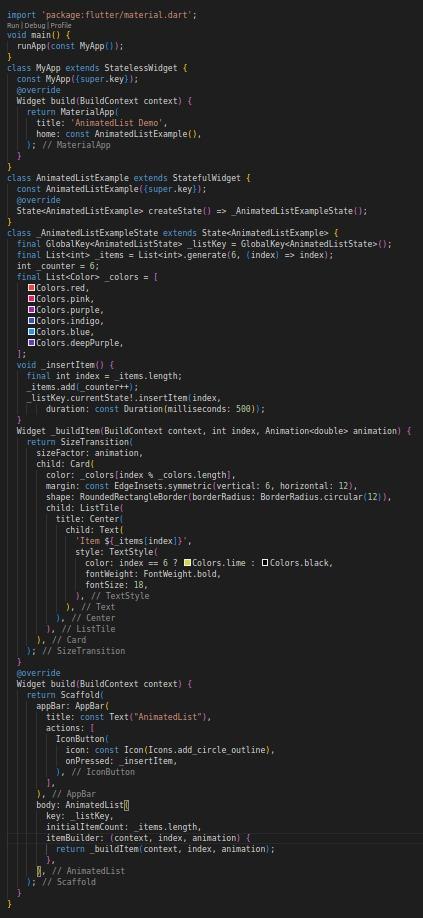
 

Code 6 : Animated Crossfade Example



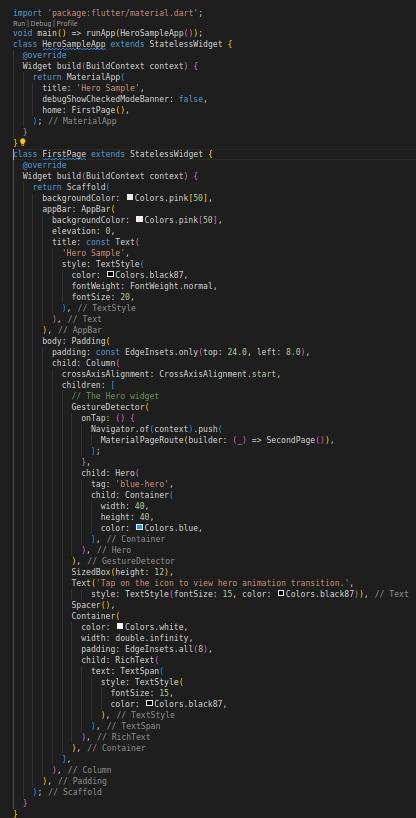
 

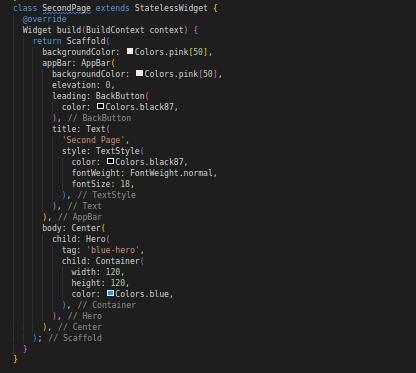
Code 7 : Animated List

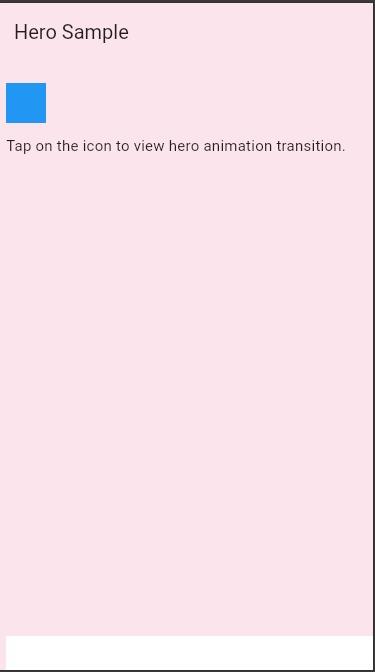
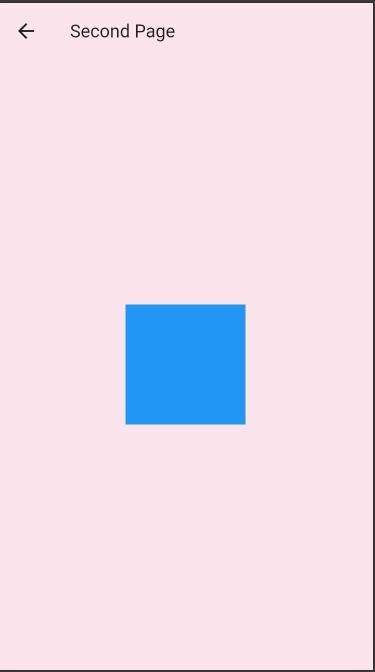


Code 8 : Hero Animation





Code 9: Resume using navigation

