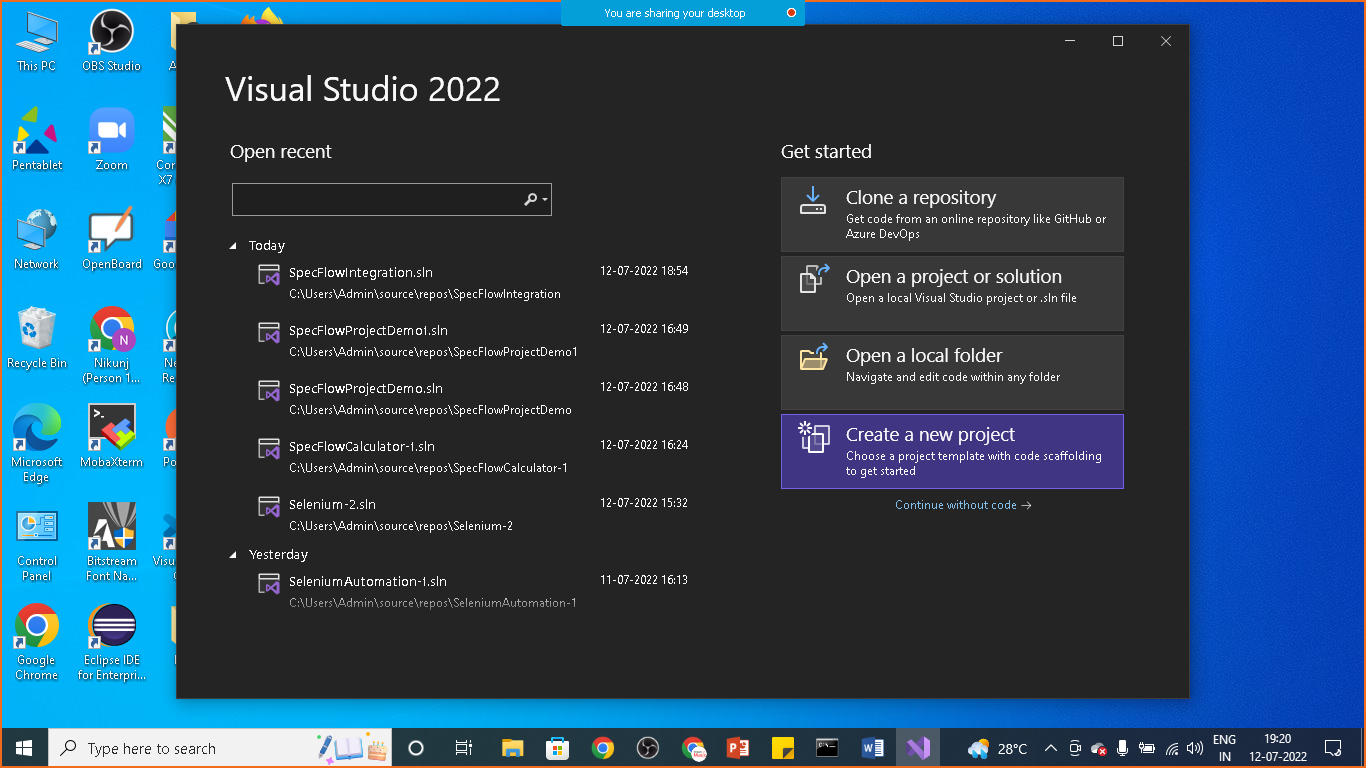
Creating Calculator Project

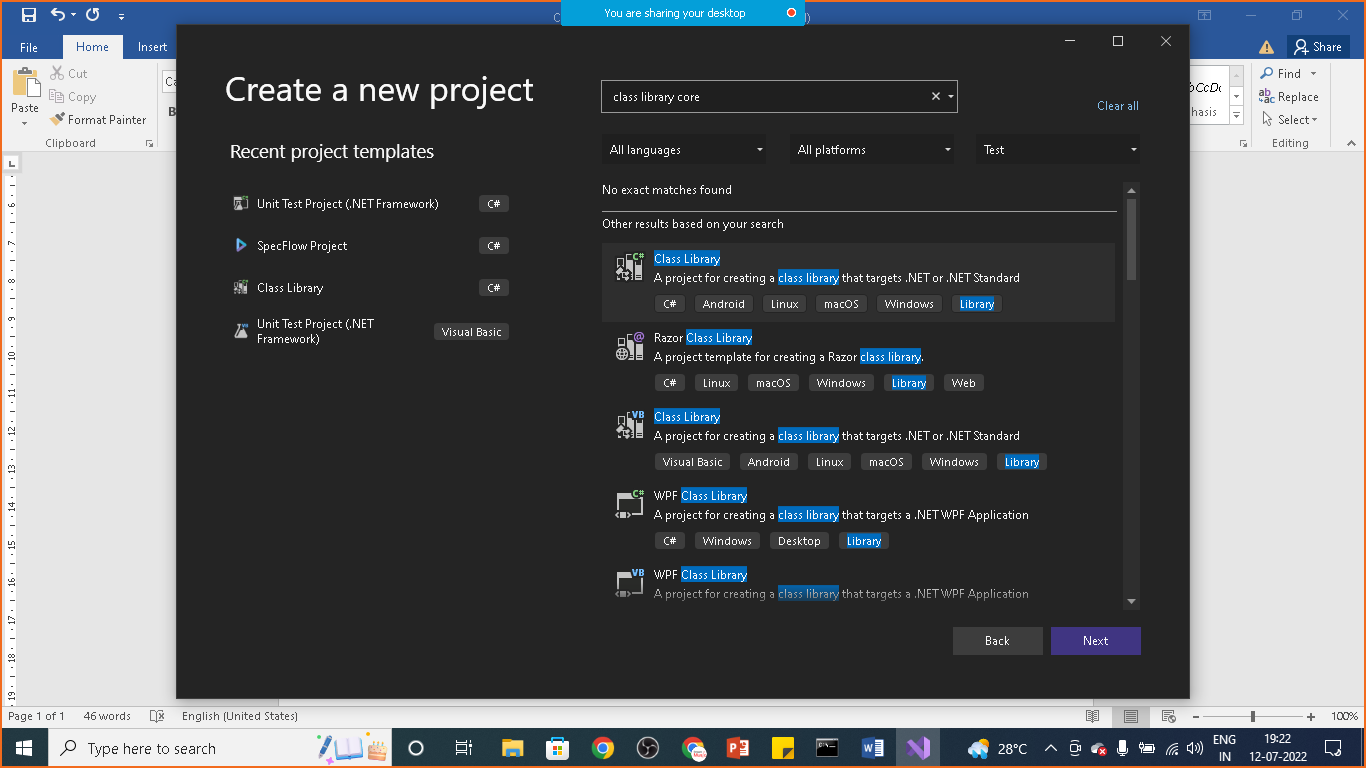
* In this project we are going to create the application that will be tested, also called as System Under Test(SUT) .
* This application will be simple calculator in C#

Step:1 open visual studio 2022

* Click on **Create a New Project**

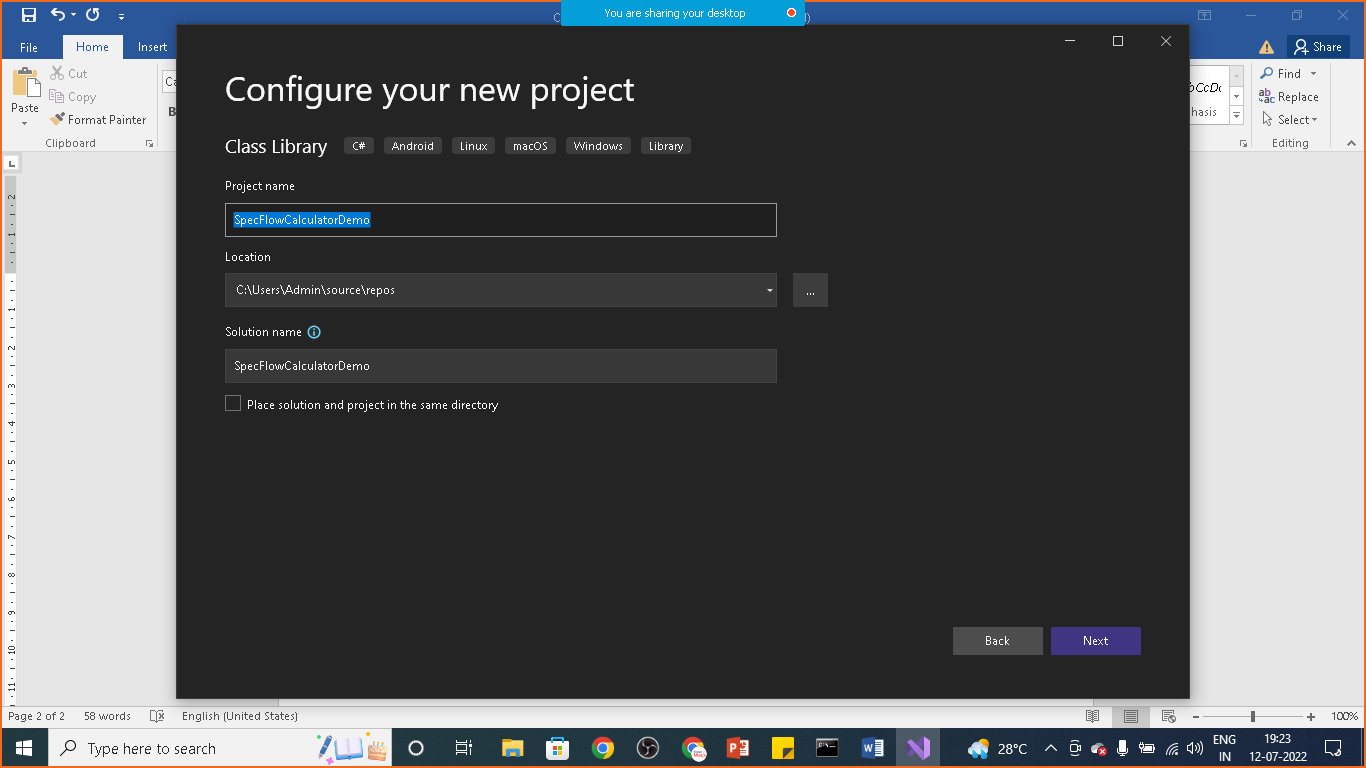


Step: 2

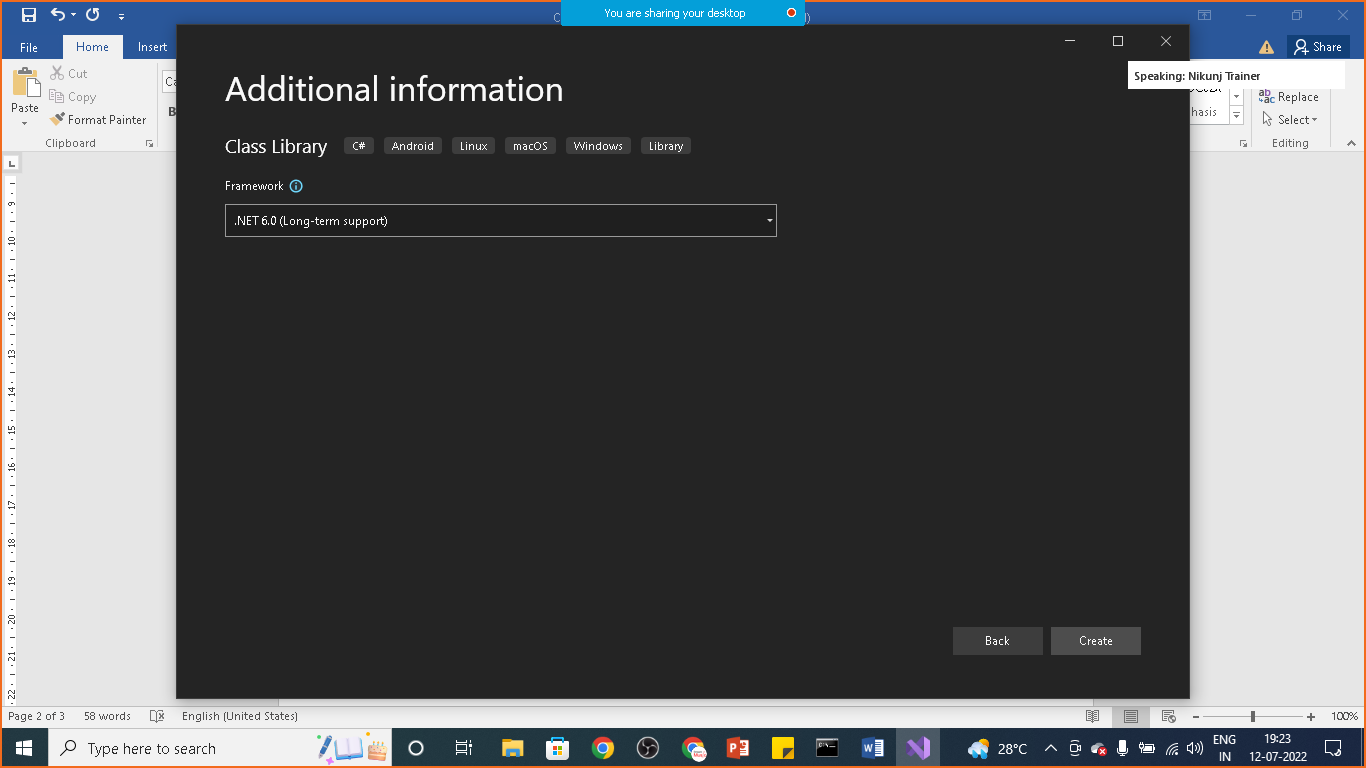


Search for : class library core and select as shown

Step:3

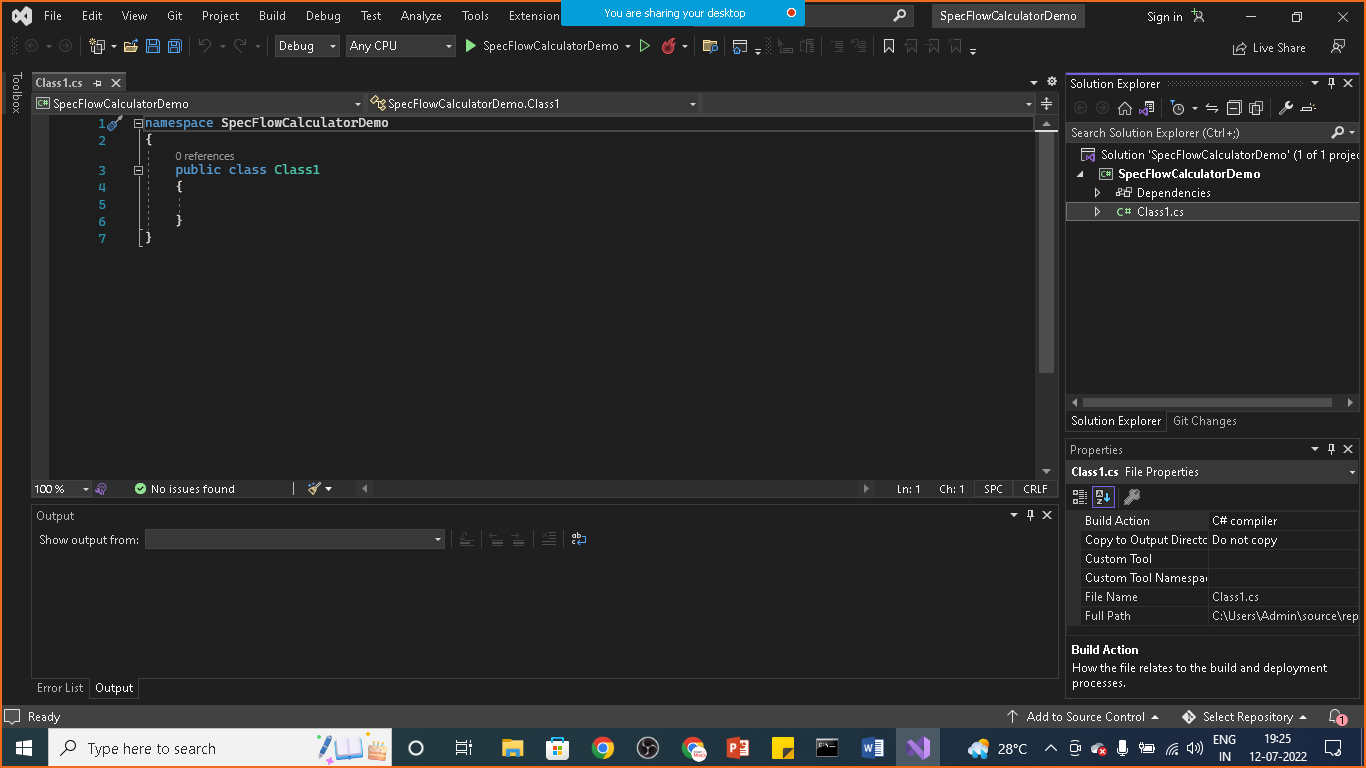


Step:4



Click on create

Step: 5: rename **class1.cs** to **calculator.cs**



Copy the below code in the **Calculator.cs** file

namespace SpecFlowCalculatorDemo

{

public class Calculator

{

public int FirstNumber { get; set; }

public int SecondNumber { get; set; }

public int Add() {

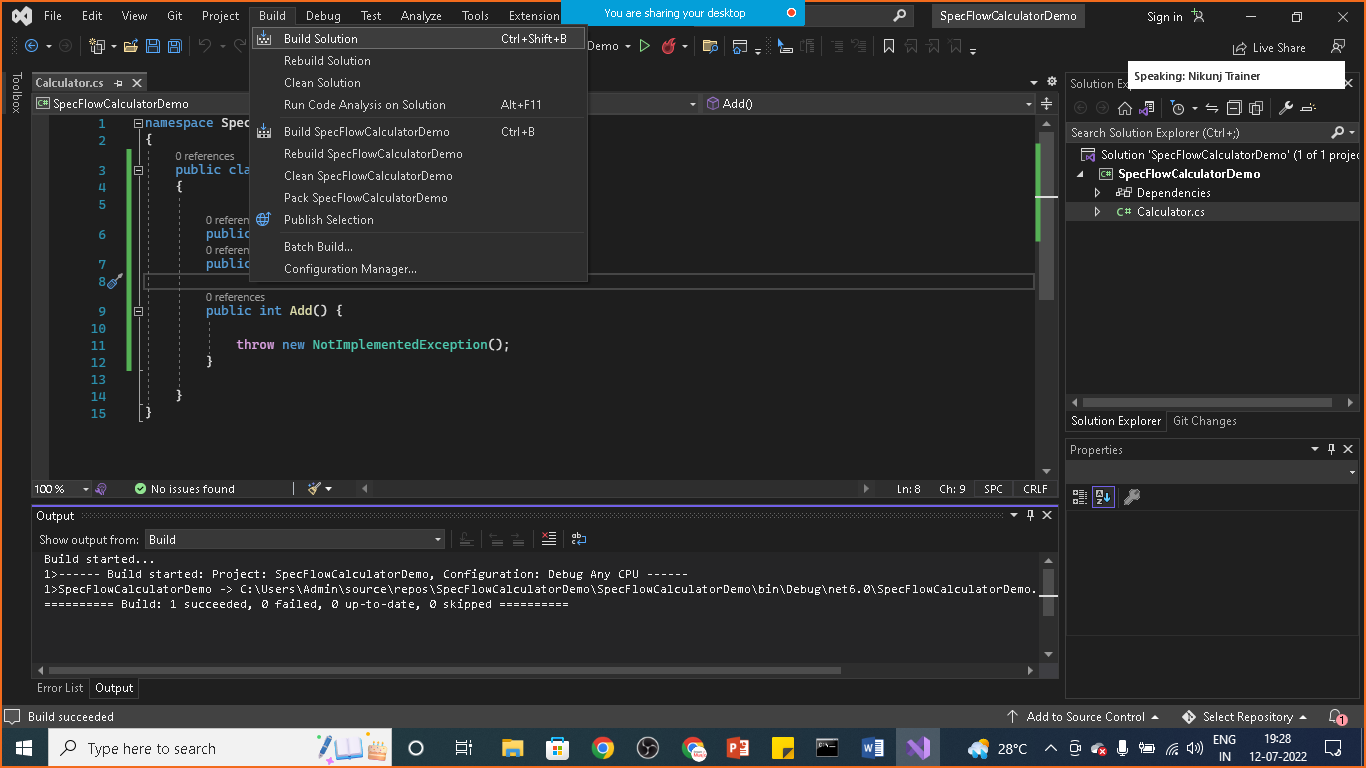
throw new NotImplementedException();

}

}

}

Step:6 click on Build and build the project

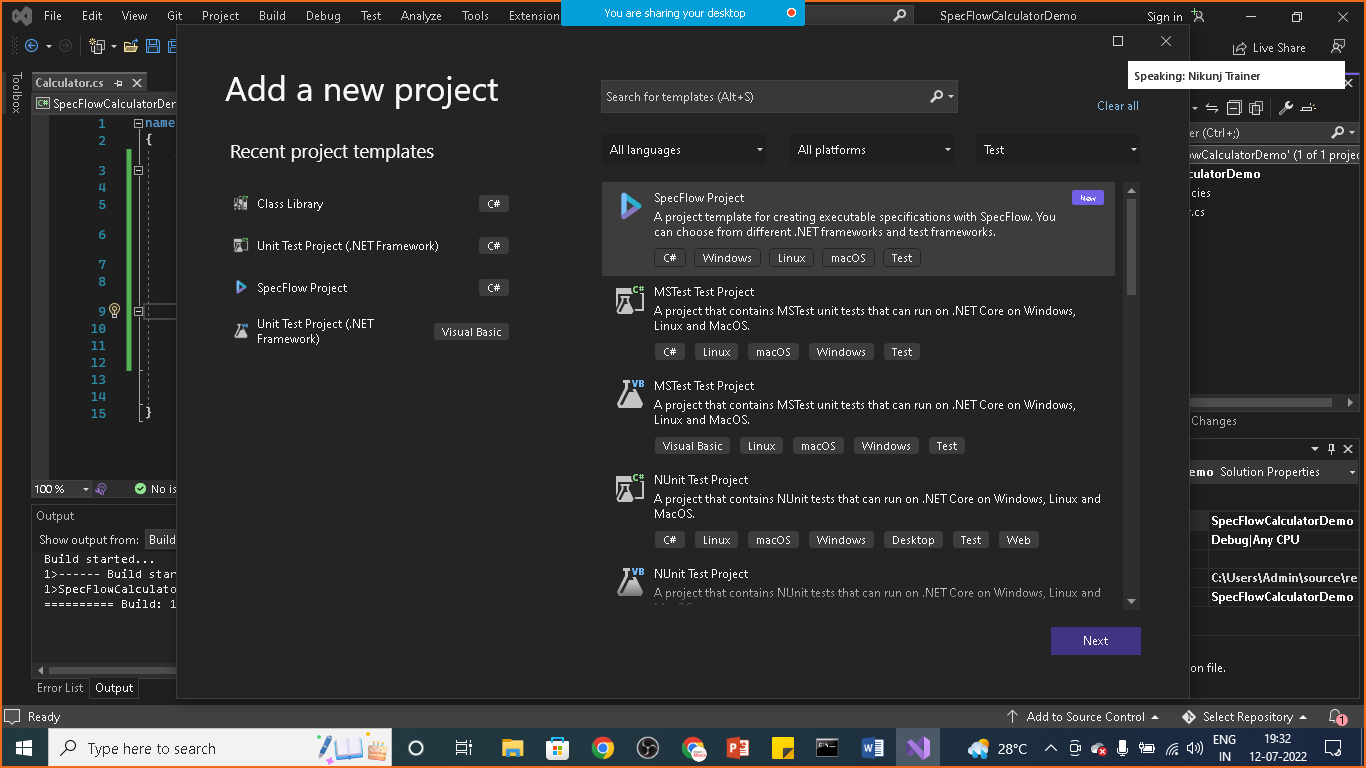


You will see that buil:1 succeeded message in output window

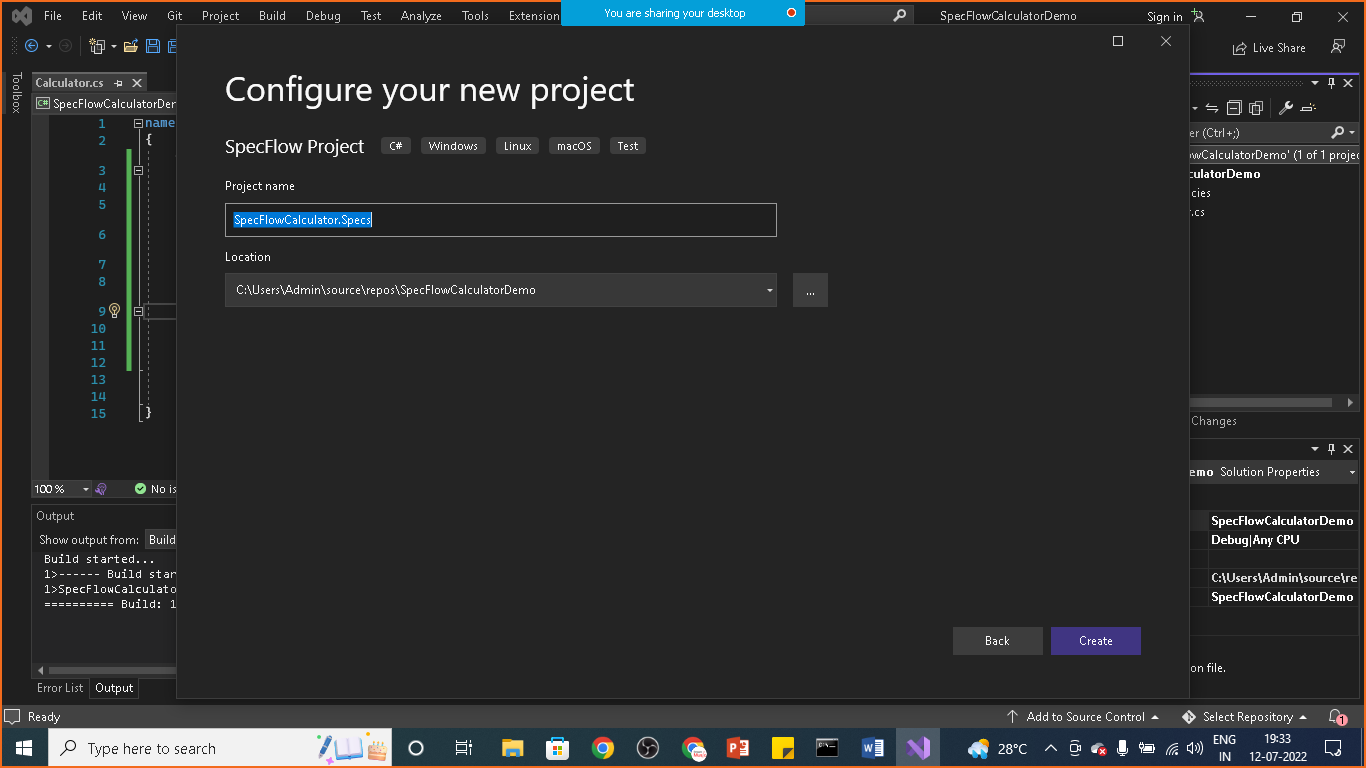
Creating SpecFlowProject

In this step you’ll create a SpecFlow Project and Add it to the existing calculator

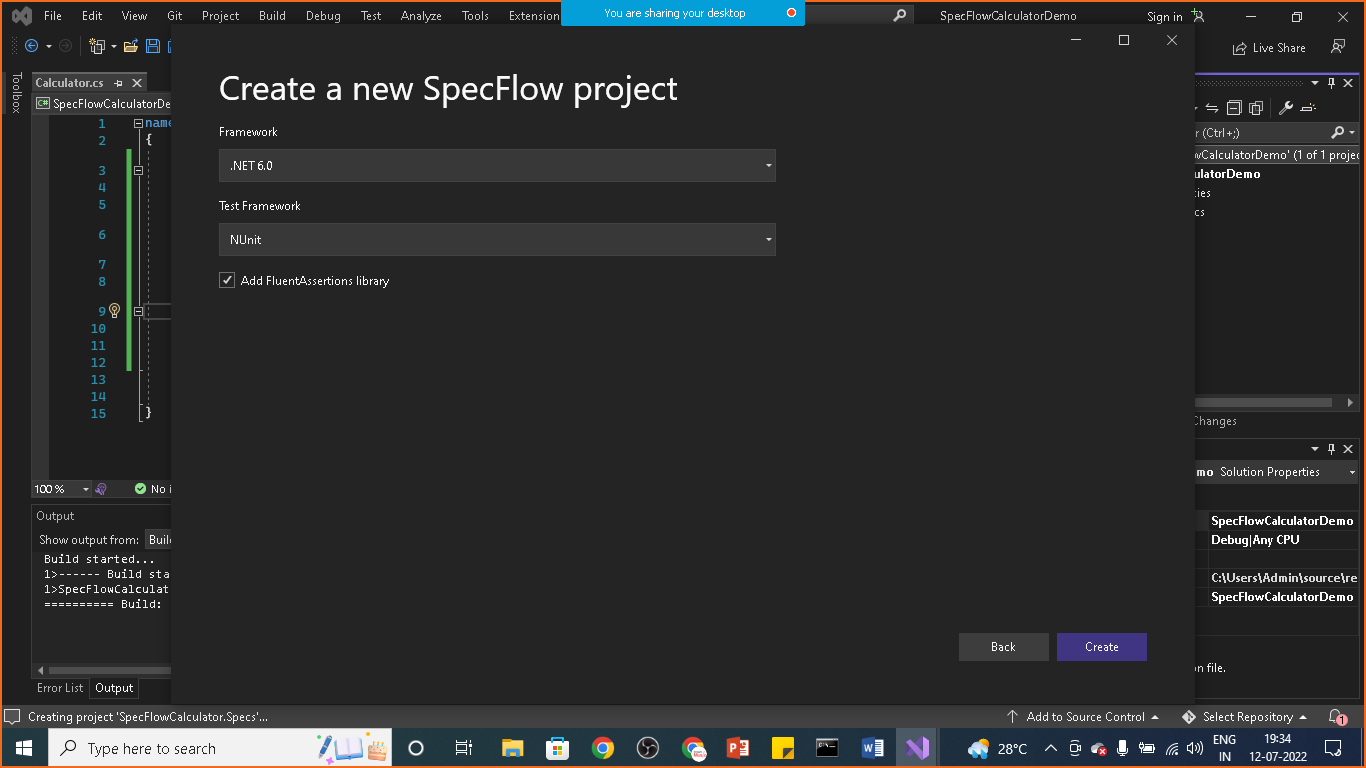
Goto “Solution ‘SpecFlowCalculator’ (1 of 1 project) “ under the solution explorer and select Add🡪 New Project….> and select specflow project



Step: 8 give project name

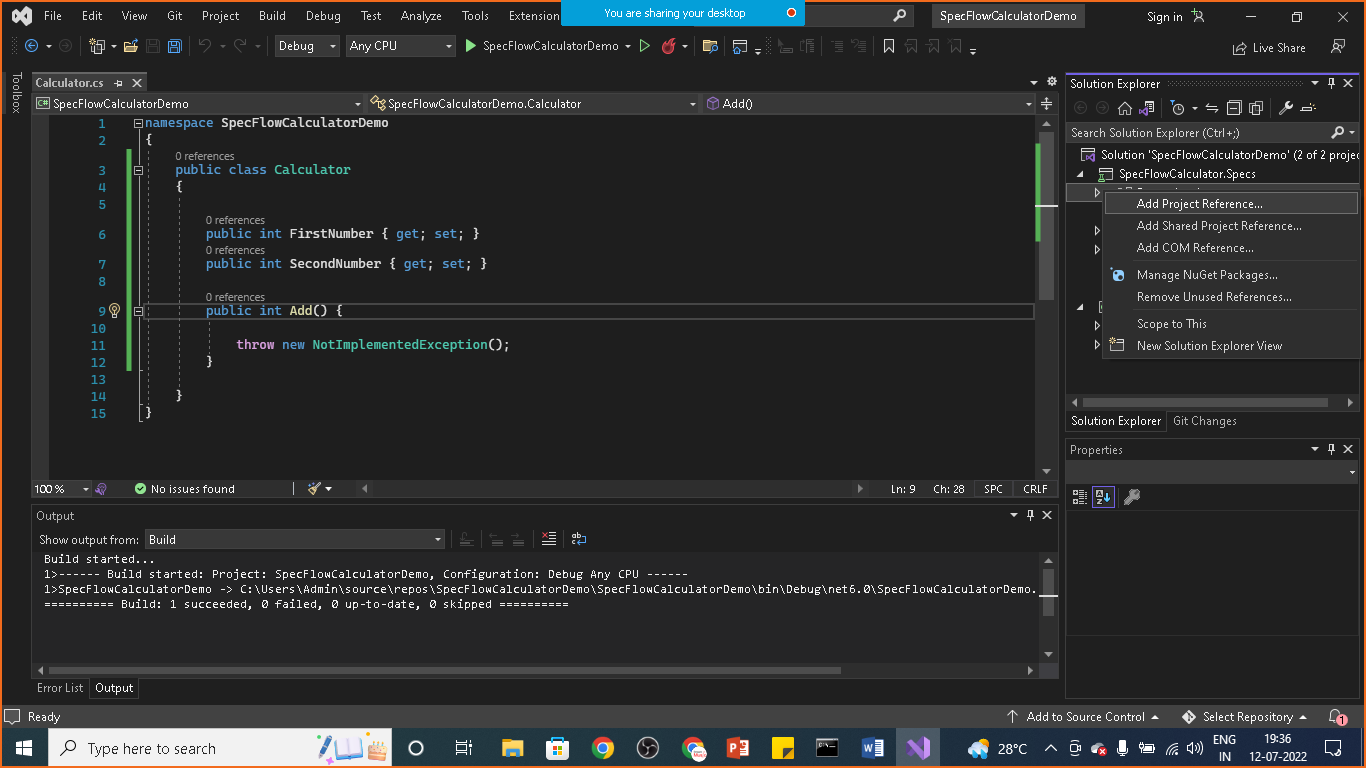


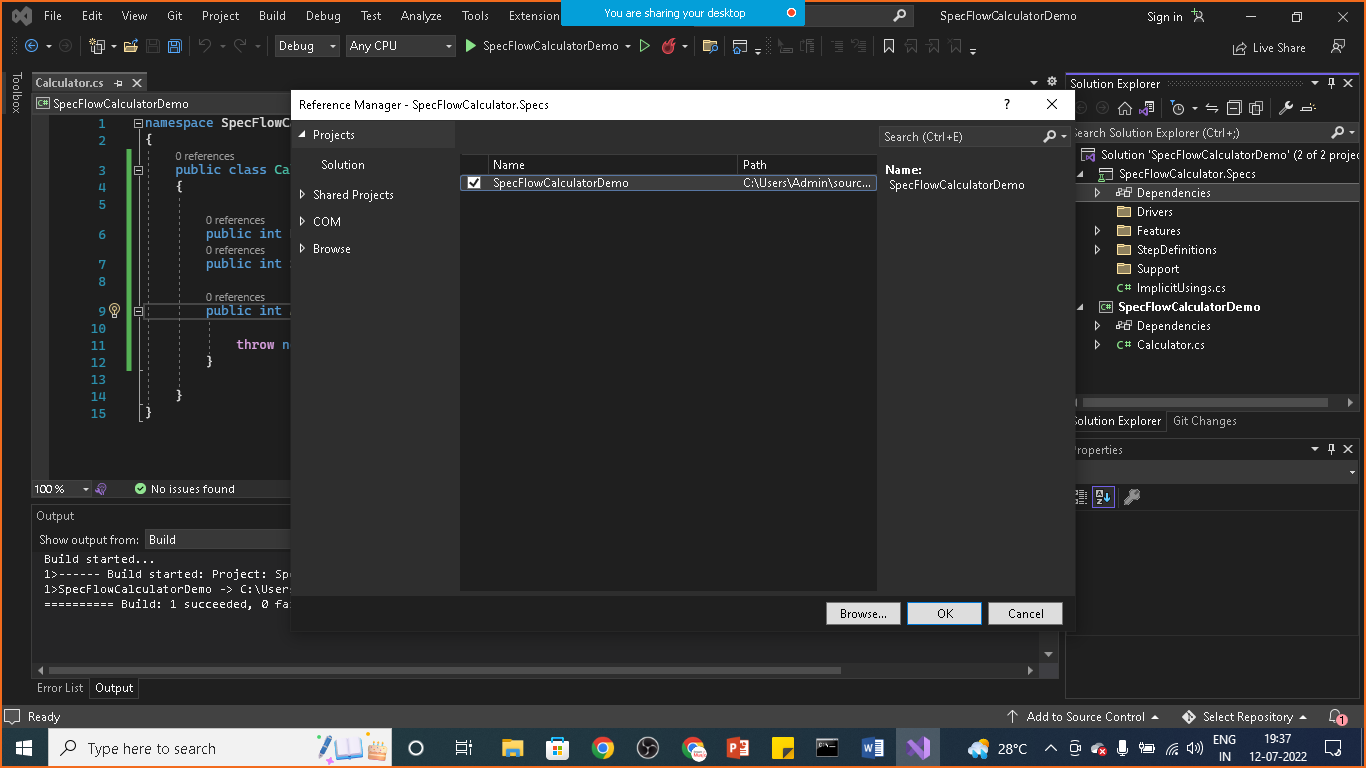
Step: 9 select framework



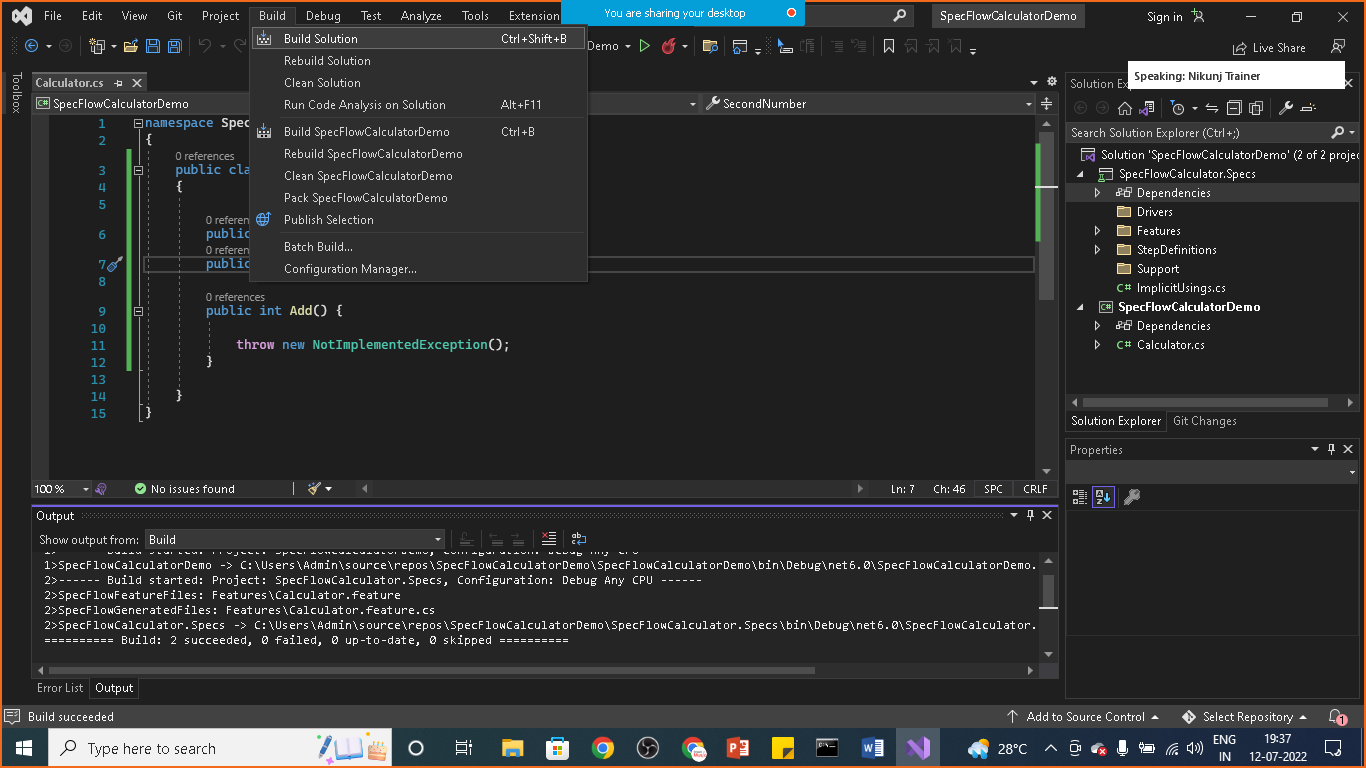
Click on create button

Step: 10 : right click on Dependency > click on Add project Reference and select SpecflowCalculator



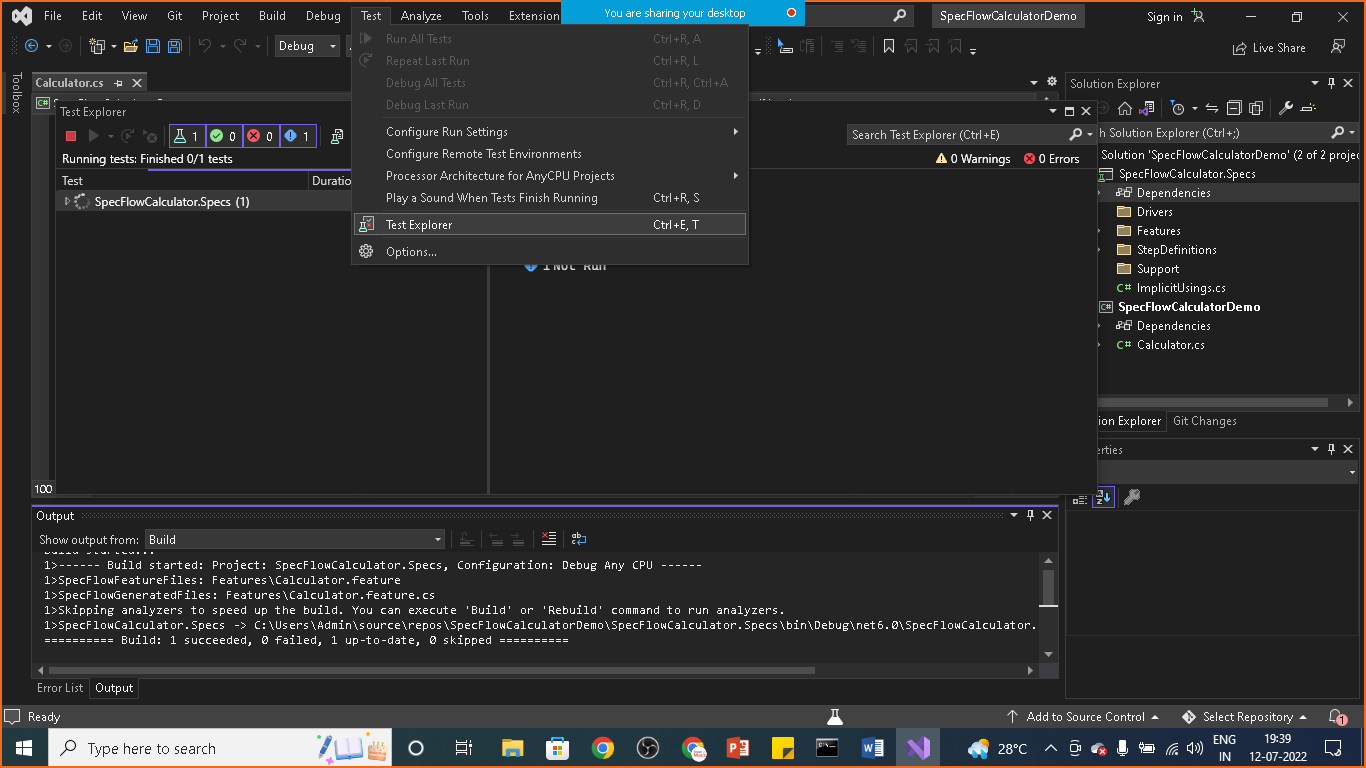


Step:11 build the solution again

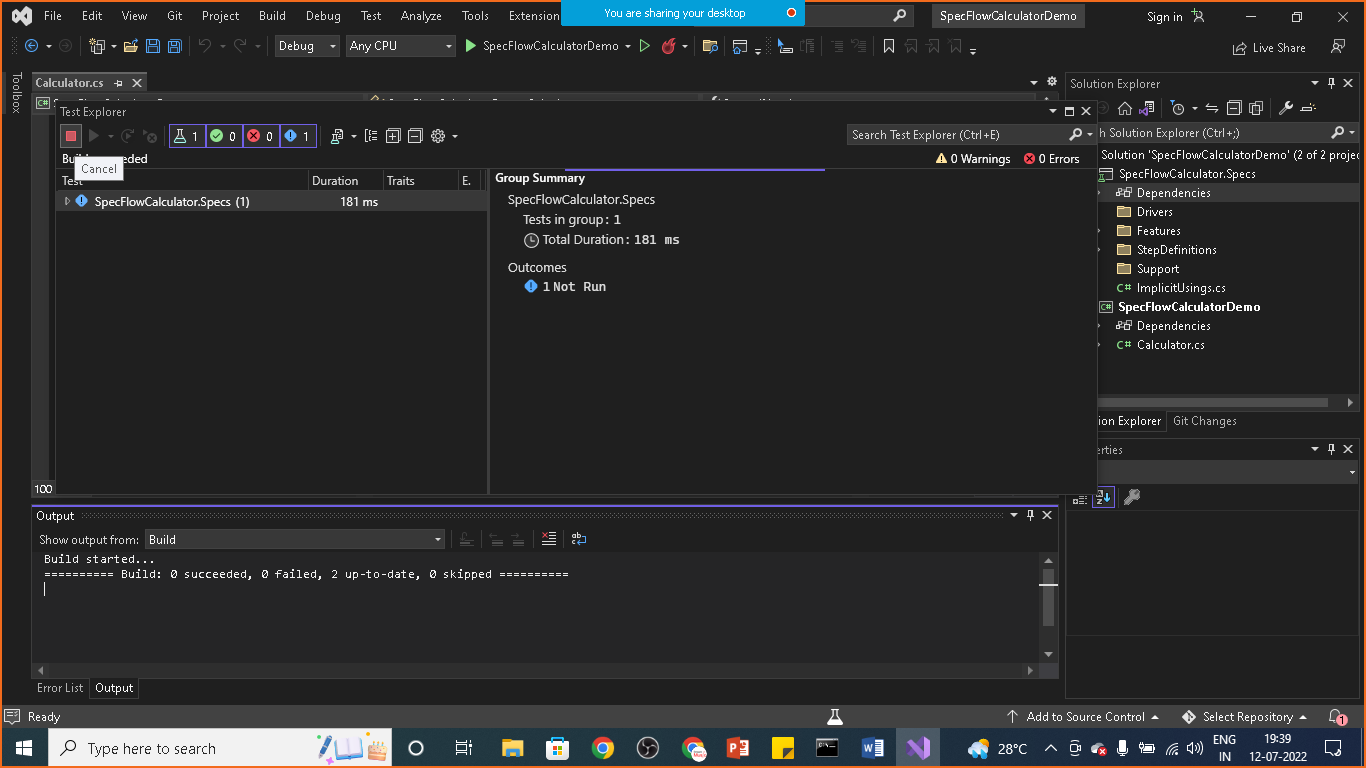


You will see that Build: succeeded message in output

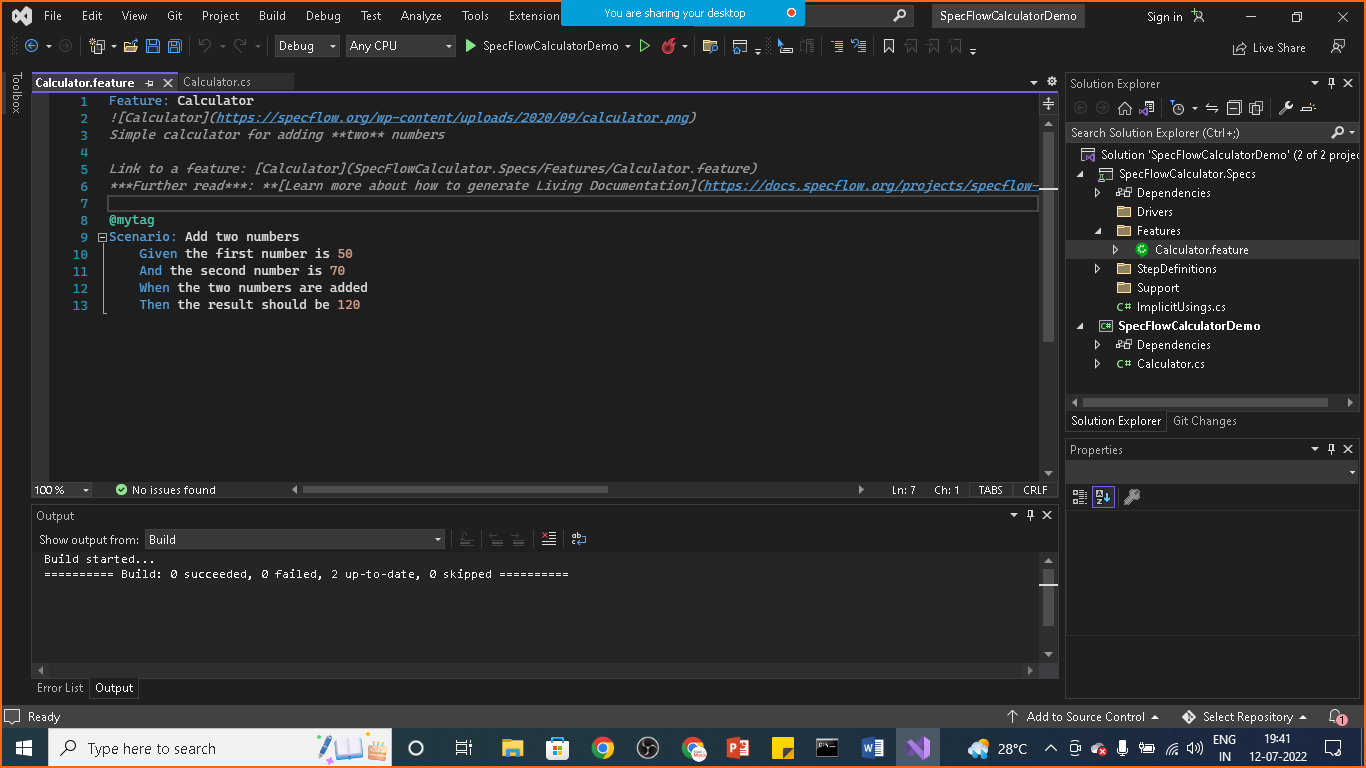
Step: 11 go to test



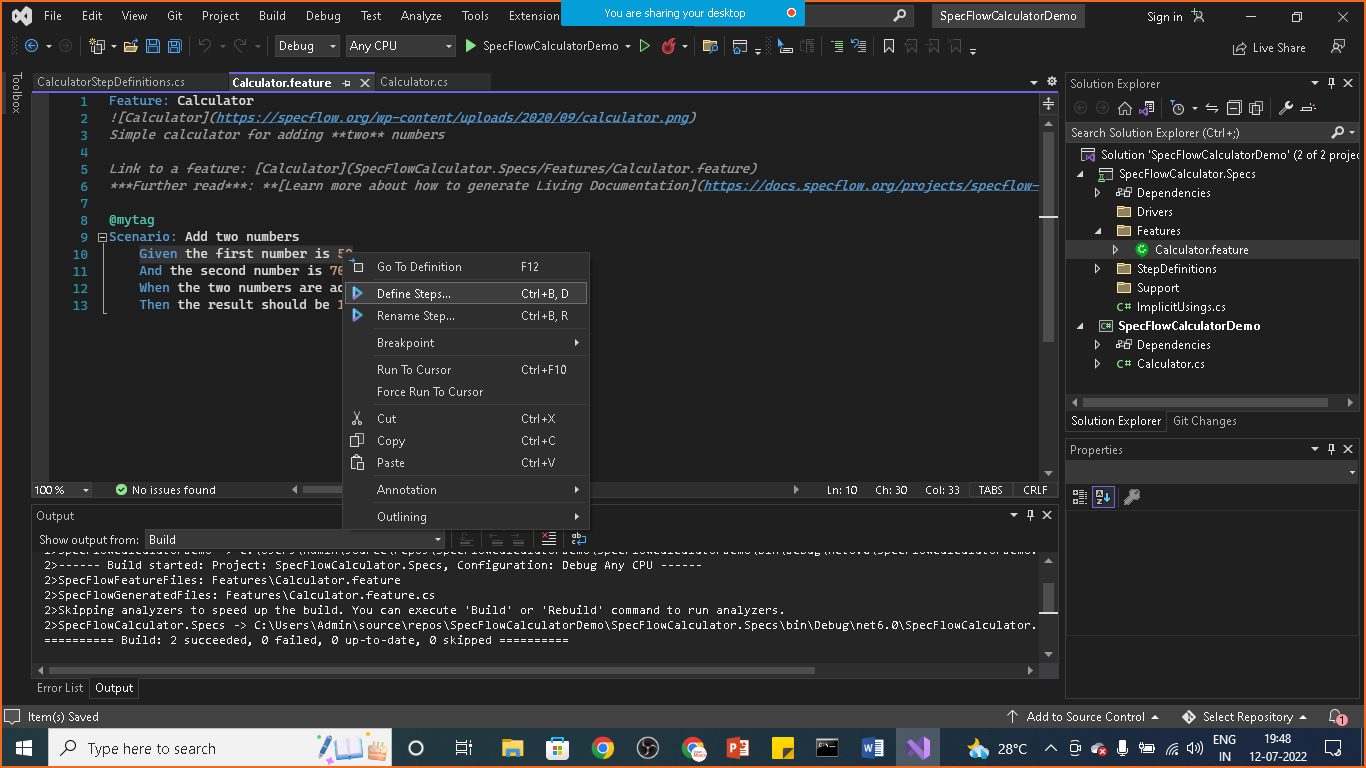
Open test explorer



**Step: Bind the first step:**



**right click on the scenario for which you want to generate step definition**



**It will open CalculatorStepDefination.cs file**

**Change the code as below**

using SpecFlowCalculatorDemo;

namespace SpecFlowCalculator.Specs.StepDefinitions

{

[Binding]

public sealed class CalculatorStepDefinitions

{

// For additional details on SpecFlow step definitions see https://go.specflow.org/doc-stepdef

private readonly Calculator \_calculator = new Calculator();

private int \_result;

private readonly ScenarioContext \_scenarioContext;

public CalculatorStepDefinitions(ScenarioContext scenarioContext)

{

\_scenarioContext = scenarioContext;

}

[Given("the first number is (.\*)")]

public void GivenTheFirstNumberIs(int number)

{

\_calculator.FirstNumber = number;

}

[Given("the second number is (.\*)")]

public void GivenTheSecondNumberIs(int number)

{

\_calculator.SecondNumber = number;

}

[When("the two numbers are added")]

public void WhenTheTwoNumbersAreAdded()

{

\_result = \_calculator.Add();

}

[Then("the result should be (.\*)")]

public void ThenTheResultShouldBe(int result)

{

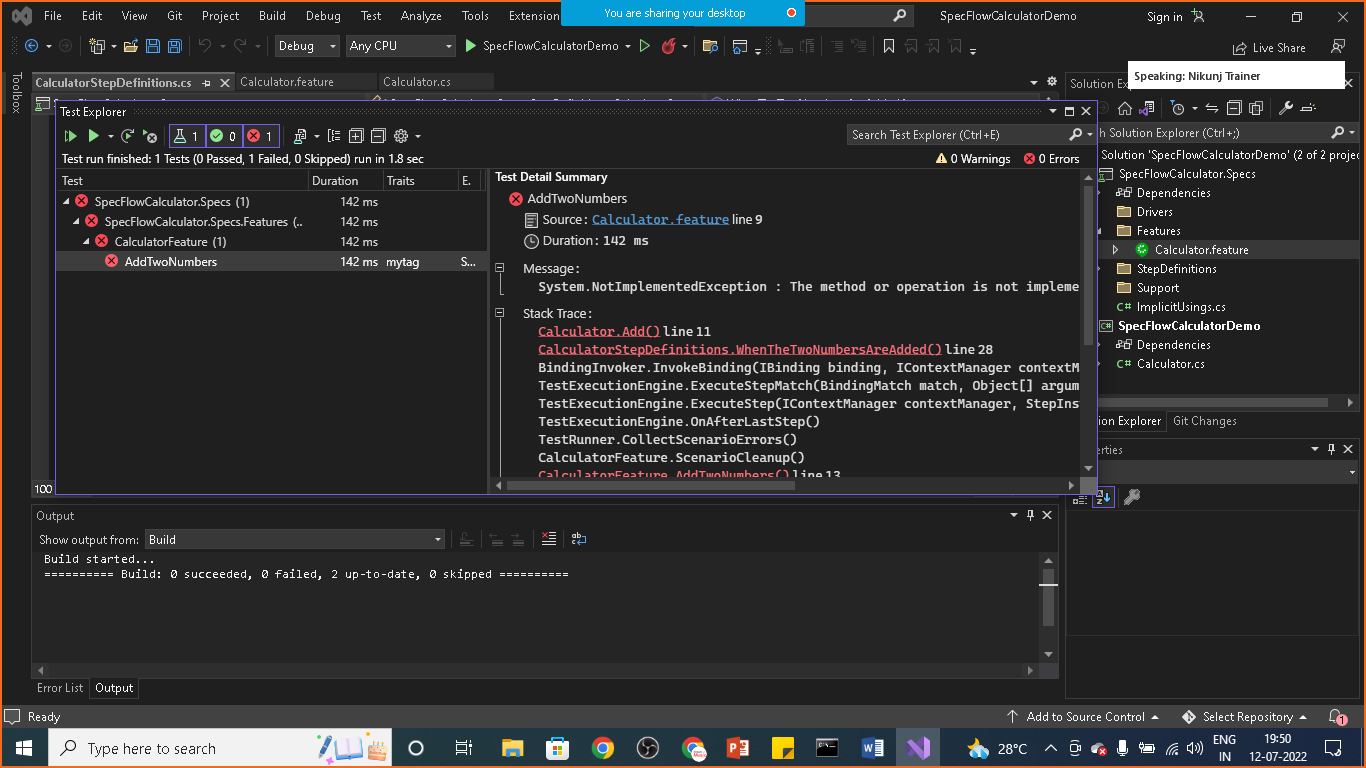
\_result.Should().Be(result);

}

}

}

**Now click on build> build solution**



**This time build succeeded**

**Goto> test> run all test**

**So the test will become failed**

**As add method is throwing :NotImplemented Exception**

**Goto Calculator.cs File and change the code of Add() method as mentioned below**

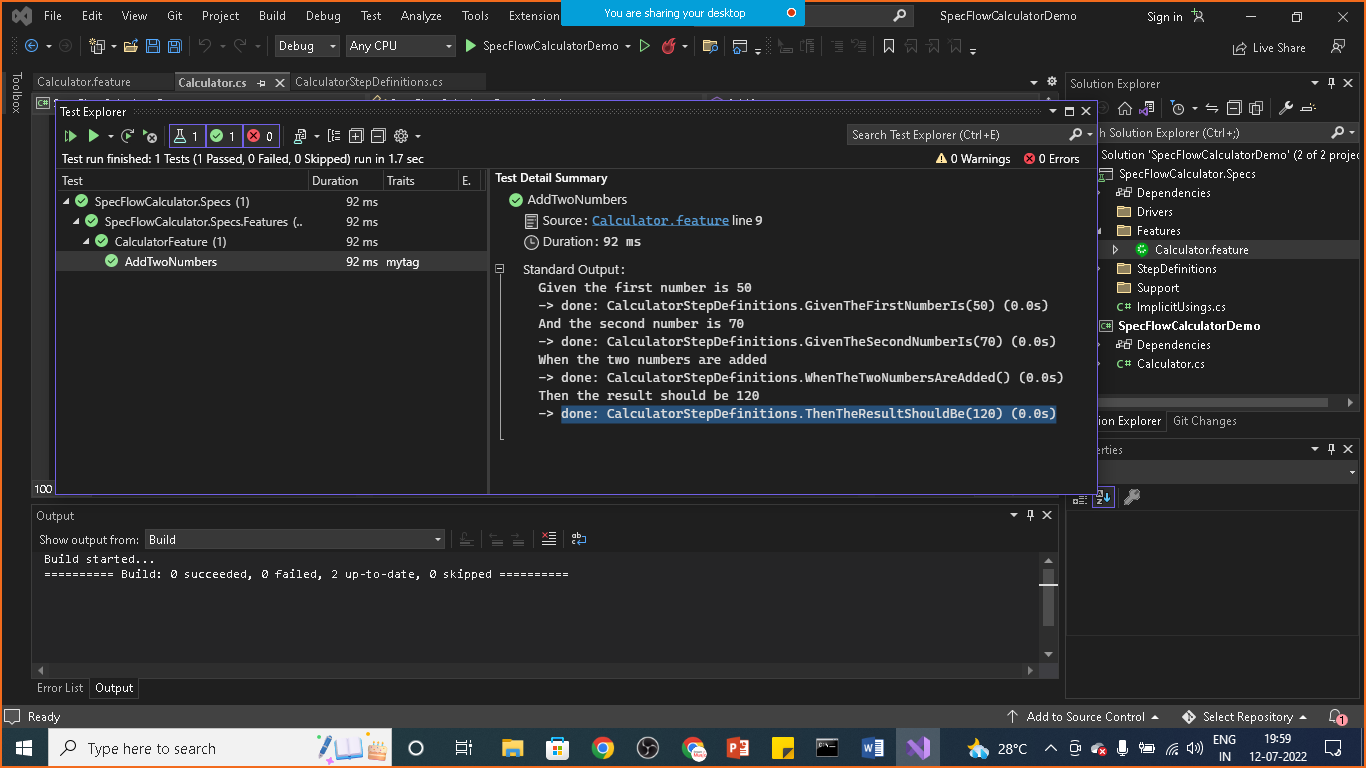
public int Add() {

return FirstNumber + SecondNumber;

//throw new NotImplementedException();

}

**Now if you will build the solution it will succeeded and test case too**

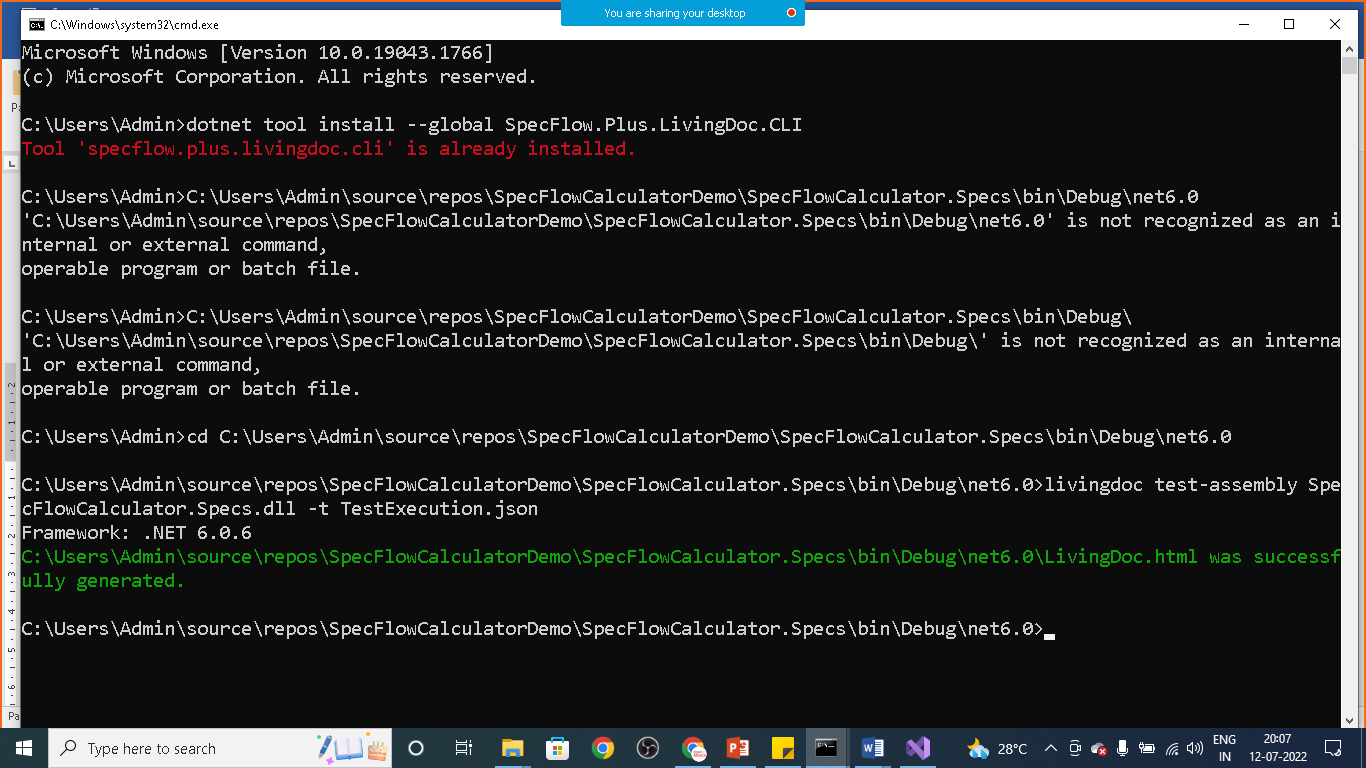


**Step: Add living Documentations**

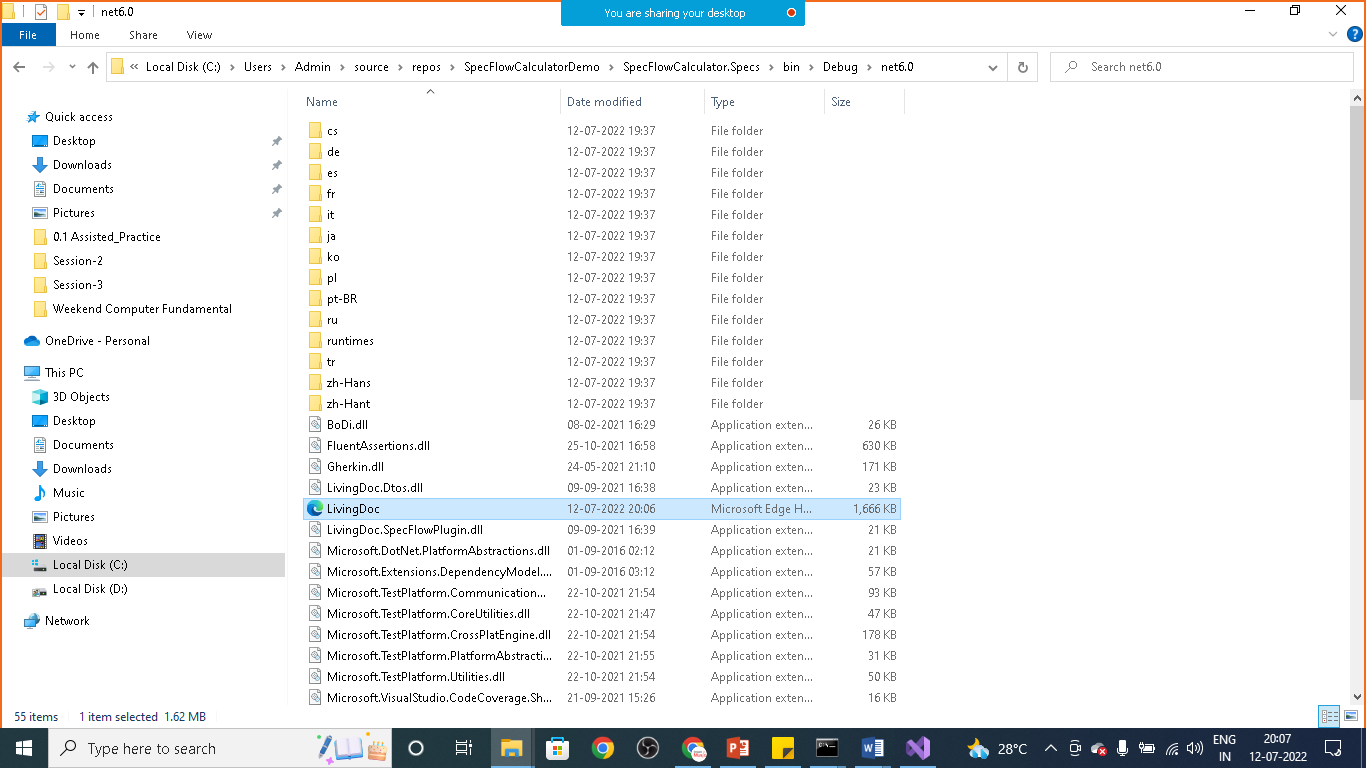
**Goto> cmd> dotnet tool install --global SpecFlow.Plus.LivingDoc.CLI**

**Once it is done move to the root directory of project and run below code**

**livingdoc test-assembly SpecFlowCalculator.Specs.dll -t TestExecution.json**



**Open the folder**



**You will see living document.html file**

**Open it with your browser**

