Báo cáo 32\_Nhi

Mục lục

**[Bài 1: Unit test 1](#_Toc12664)**

[Bước 1 : Tạo 1 project 1](#_Toc22750)

[Bước 3: Tạo một tập tin Calculation\_32\_Nhi.cs chứa lớp public Calculation\_32\_Nhi dùng có phương thức GCD\_32\_Nhi và LCM\_32\_Nhi để thực hiện phép tính tìm bội chung nhỏ nhất và ước chung lớn nhất với hai số nguyên. 2](#_Toc32347)

[Bước 4: Vào winform viết code để thực hiện các phép tính: 4](#_Toc24072)

[Bước 5: Tạo project kiểm thử để kiểm thử các phép toán trong chương trình trên 4](#_Toc9480)

[Bước 6: Tại project Unit Test, thực hiện Add Reference để tham chiếu đến project cần thực hiện Unit Test 5](#_Toc23202)

[Bước 7: Thực thi test case với các dữ liệu test có sẵn: 6](#_Toc9207)

[Bước 8: Viết code cho calculatorTest\_32\_Nhi 8](#_Toc11198)

[Bước 9: Chạy kiểm thử 9](#_Toc17027)

**[Bài 2:SELENIUM 9](#_Toc10135)**

[Cài đặt Selenium Web Driver 9](#_Toc8479)

[Giao diện 11](#_Toc2926)

[Lấy thông tin để code: 11](#_Toc5177)

[Code: đăng nhập google 13](#_Toc32100)

[Kết quả:đăng nhập google 14](#_Toc26416)

[Code: tìm từ 14](#_Toc27154)

[Kết quả tìm từ: 16](#_Toc20493)

**[Bài 3: 16](#_Toc1002)**

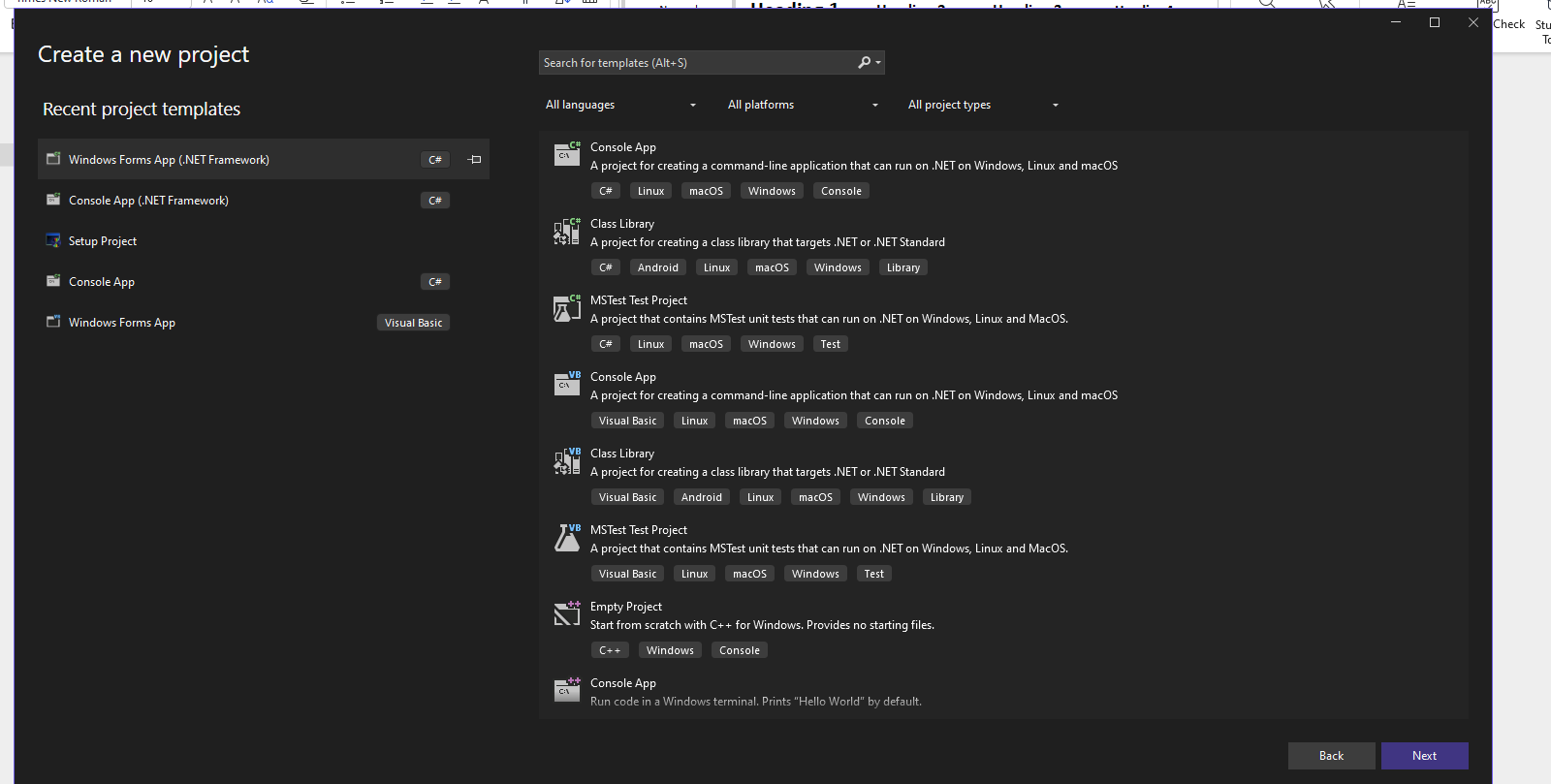
[+Bước 1:Download và cài đặt postman 16](#_Toc25047)

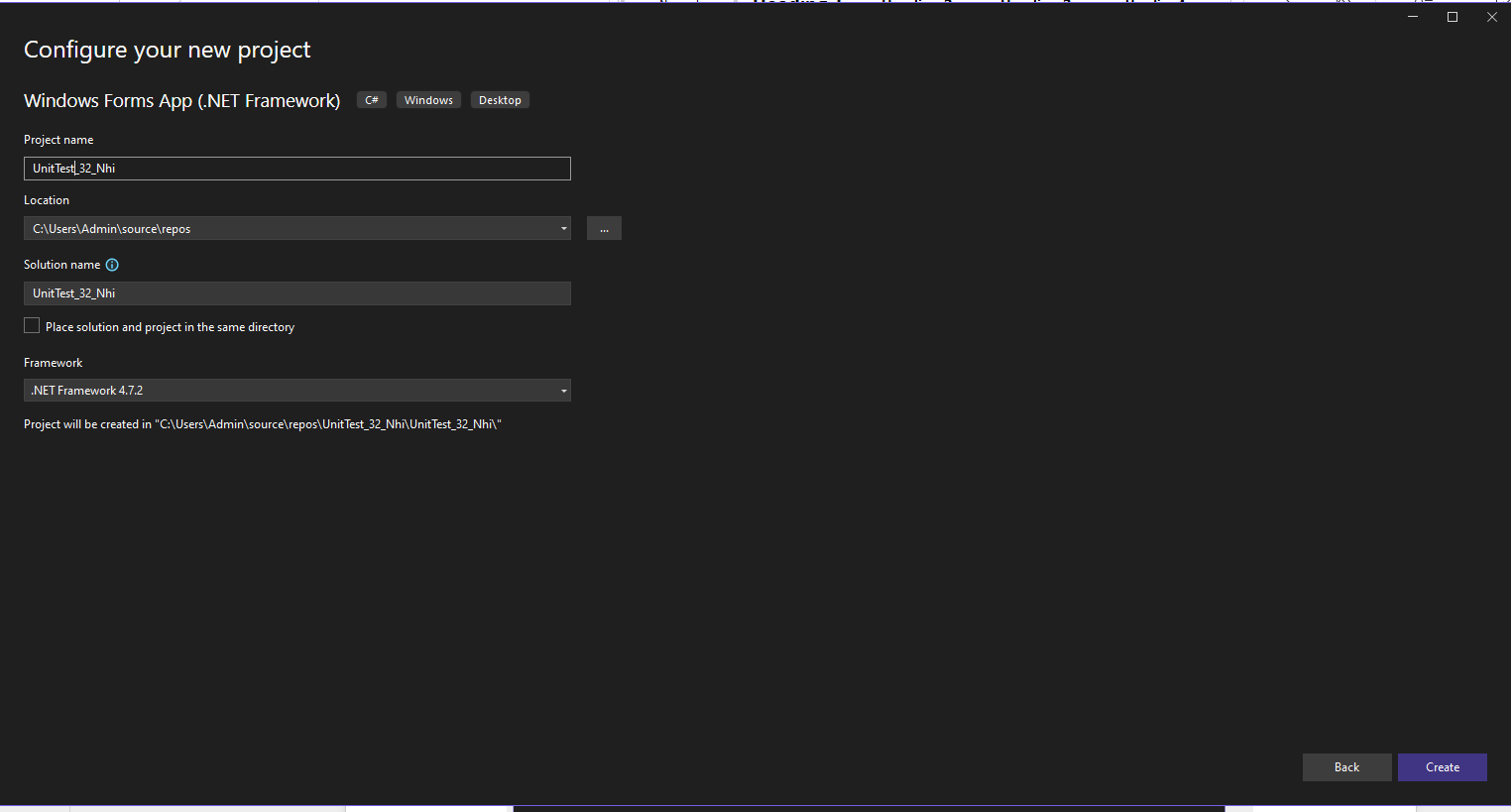
[+Bước 2: Tạo 1 workspace mới 17](#_Toc6830)

[+Bước 3 : Kiểm thử 18](#_Toc31424)

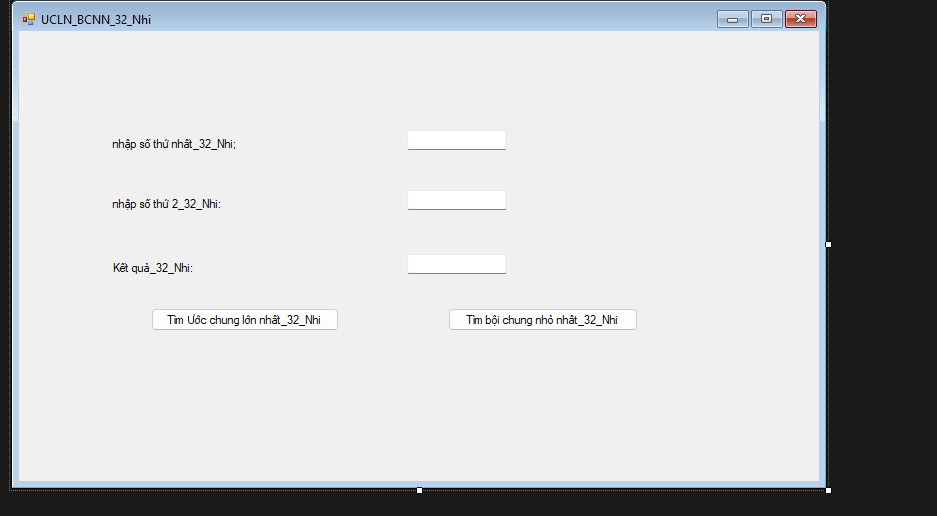
# Bài 1: Unit test

## Bước 1 : Tạo 1 project



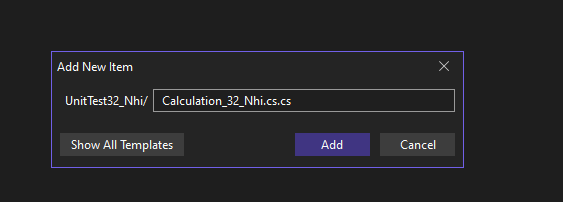


Bước 2: Thiết kế giao diện tính toán như bên dưới và việc xử lý các phép tính đều thông qua lớp Caculation\_32\_Nhi.



## Bước 3: Tạo một tập tin Calculation\_32\_Nhi.cs chứa lớp public Calculation\_32\_Nhi dùng có phương thức GCD\_32\_Nhi và LCM\_32\_Nhi để thực hiện phép tính tìm bội chung nhỏ nhất và ước chung lớn nhất với hai số nguyên.

Chuột trái UnitTest\_32\_Nhi->Add->New Item



using System;

namespace UnitTest\_32\_Nhi

{

public class Calculation\_32\_Nhi

{

private int a\_32\_Nhi, b\_32\_Nhi;

// Constructor để khởi tạo a và b

public Calculation\_32\_Nhi(int a\_32\_Nhi, int b\_32\_Nhi)

{

this.a\_32\_Nhi = a\_32\_Nhi;

this.b\_32\_Nhi = b\_32\_Nhi;

}

// Phương thức thực hiện tính toán GCD hoặc LCM tùy vào ký hiệu được truyền vào

public int Execute\_32\_Nhi(string CalSymbol\_32\_Nhi)

{

int result\_32\_Nhi = 0;

switch (CalSymbol\_32\_Nhi)

{

case "GCD":

result\_32\_Nhi = GCD\_32\_Nhi(this.a\_32\_Nhi, this.b\_32\_Nhi);

break;

case "LCM":

result\_32\_Nhi = LCM\_32\_Nhi(this.a\_32\_Nhi, this.b\_32\_Nhi);

break;

}

return result\_32\_Nhi;

}

// Tính GCD của hai số

private int GCD\_32\_Nhi(int a\_32\_Nhi, int b\_32\_Nhi)

{

while (b\_32\_Nhi != 0)

{

int temp\_32\_Nhi = b\_32\_Nhi;

b\_32\_Nhi = a\_32\_Nhi % b\_32\_Nhi;

a\_32\_Nhi = temp\_32\_Nhi;

}

return Math.Abs(a\_32\_Nhi); // Trả về GCD tuyệt đối

}

// Tính LCM của hai số

private int LCM\_32\_Nhi(int a\_32\_Nhi, int b\_32\_Nhi)

{

return Math.Abs(a\_32\_Nhi \* b\_32\_Nhi) / GCD\_32\_Nhi(a\_32\_Nhi, b\_32\_Nhi);

}

}

}

## **Bước 4: Vào winform viết code để thực hiện các phép tính:**

using System;

using System.Windows.Forms;

namespace UnitTest\_32\_Nhi

{

public partial class UCLN\_BCNN\_32\_Nhi : Form

{

public UCLN\_BCNN\_32\_Nhi()

{

InitializeComponent();

}

private void btnUCLN\_32\_Nhi\_Click(object sender, EventArgs e)

{

int a\_32\_Nhi = int.Parse(txtSoThuNhat\_32\_Nhi.Text);

int b\_32\_Nhi = int.Parse(txtSoThuHai\_32\_Nhi.Text);

Calculation\_32\_Nhi c\_32\_Nhi = new Calculation\_32\_Nhi(a\_32\_Nhi, b\_32\_Nhi);

int result\_32\_Nhi = c\_32\_Nhi.Execute\_32\_Nhi("GCD");

txtKetQua\_32\_Nhi.Text = $"ƯCLN của {a\_32\_Nhi} và {b\_32\_Nhi} là: {result\_32\_Nhi}";

}

private void btnBCNN\_32\_Nhi\_Click(object sender, EventArgs e)

{

int a\_32\_Nhi = int.Parse(txtSoThuNhat\_32\_Nhi.Text);

int b\_32\_Nhi = int.Parse(txtSoThuHai\_32\_Nhi.Text);

Calculation\_32\_Nhi c\_32\_Nhi = new Calculation\_32\_Nhi(a\_32\_Nhi, b\_32\_Nhi);

int result\_32\_Nhi = c\_32\_Nhi.Execute\_32\_Nhi("LCM");

txtKetQua\_32\_Nhi.Text = $"BCNN của {a\_32\_Nhi} và {b\_32\_Nhi} là: {result\_32\_Nhi}";

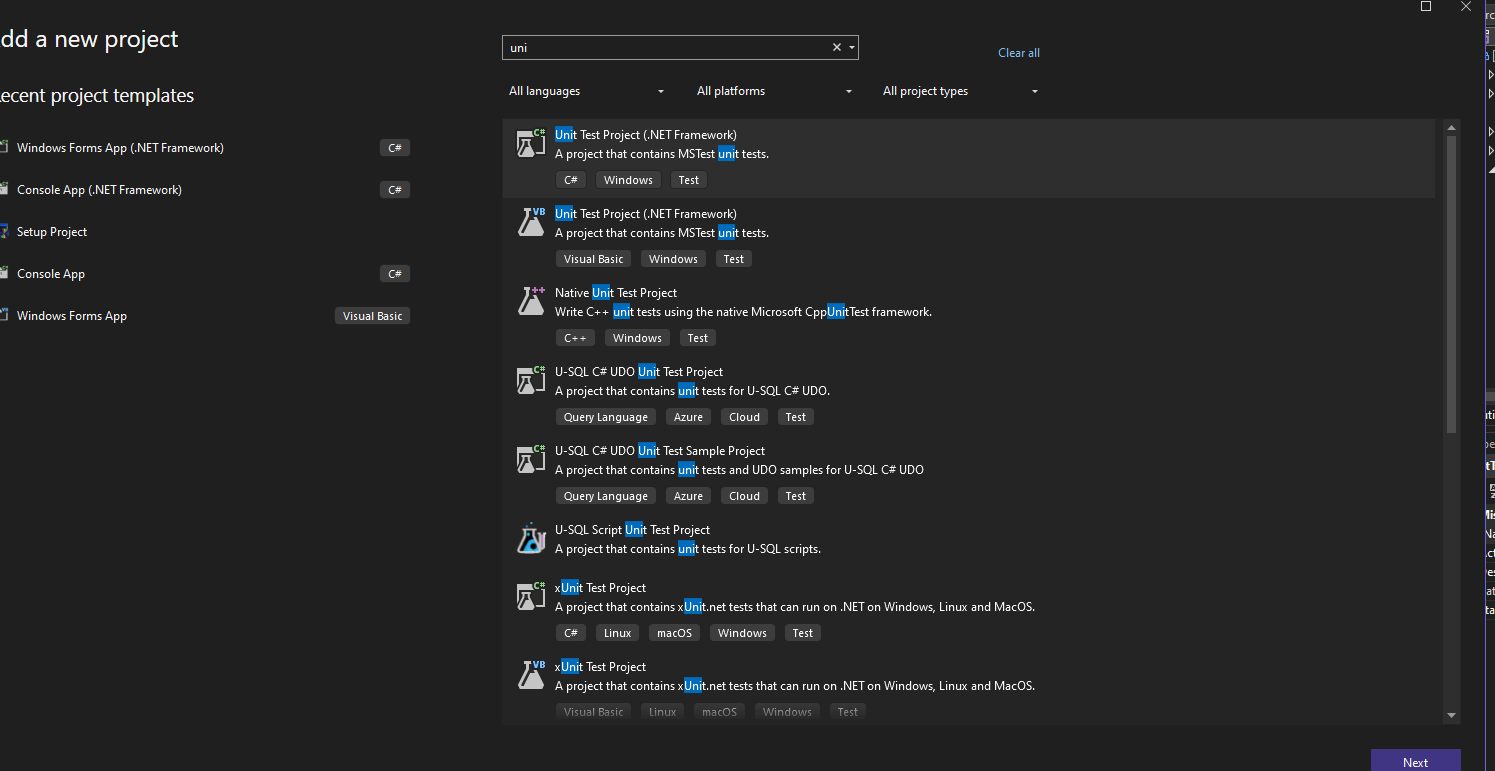
}

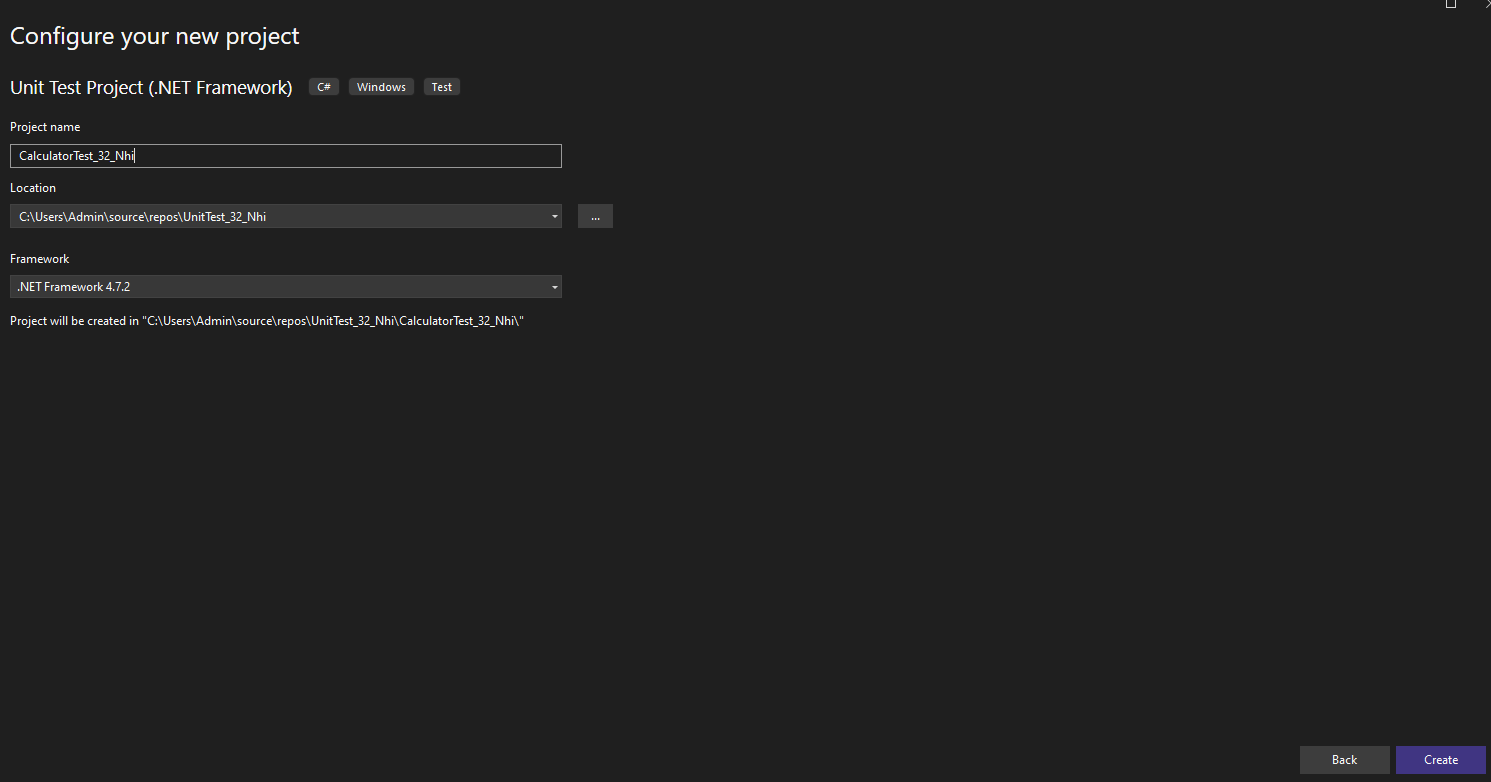
}

}

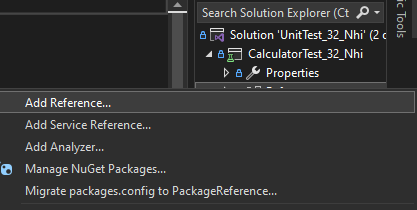
## Bước 5: Tạo project kiểm thử để kiểm thử các phép toán trong chương trình trên

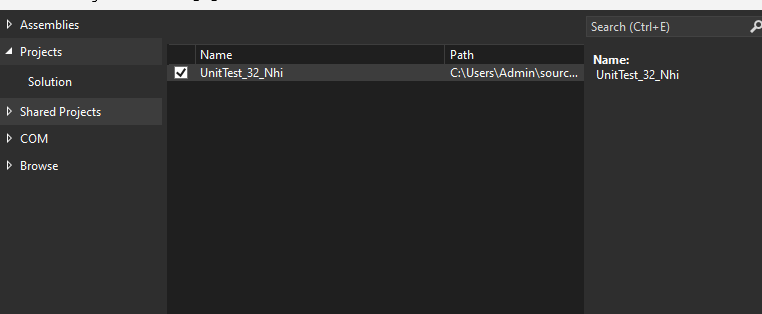
**Click chuột phải Solution > Add > New Projects... Sau đó chọn loại project là “Unit Test Project”**

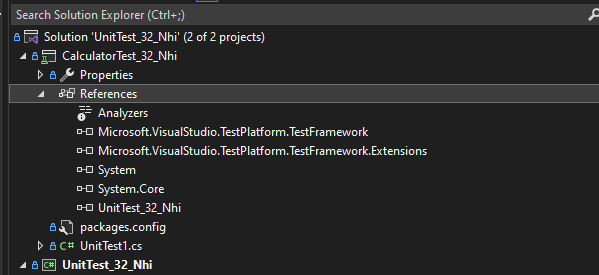




## Bước 6: Tại project Unit Test, thực hiện Add Reference để tham chiếu đến project cần thực hiện Unit Test

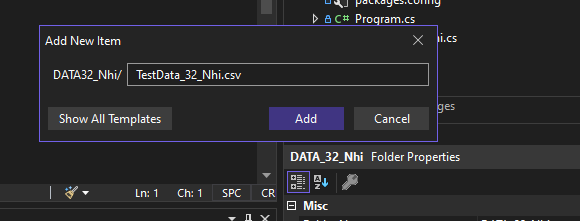




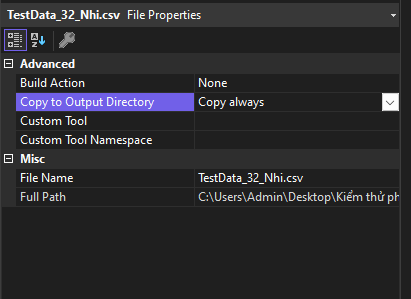


## Bước 7: Thực thi test case với các dữ liệu test có sẵn:

### +tạo thư mục Data\_32\_Nhi trong project test(click phải chuột vào thư mục Data\_32\_Nhi > New Item... tạo tập tin TestData\_32\_Nhi.csv



### +Chuột phải tập tin TestData\_32\_Nhi.csv chọn Properties, thiết lập thuộc tính “Copy to Output Directory” thành “Copy always” để tập tin này sẽ được sao chép vào thư mục bin khi build project.



### +Nhập dữ liệu vào tập tin TestData\_32\_Nhi.csv

a,b,expected

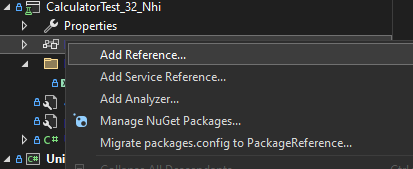
3,4,7

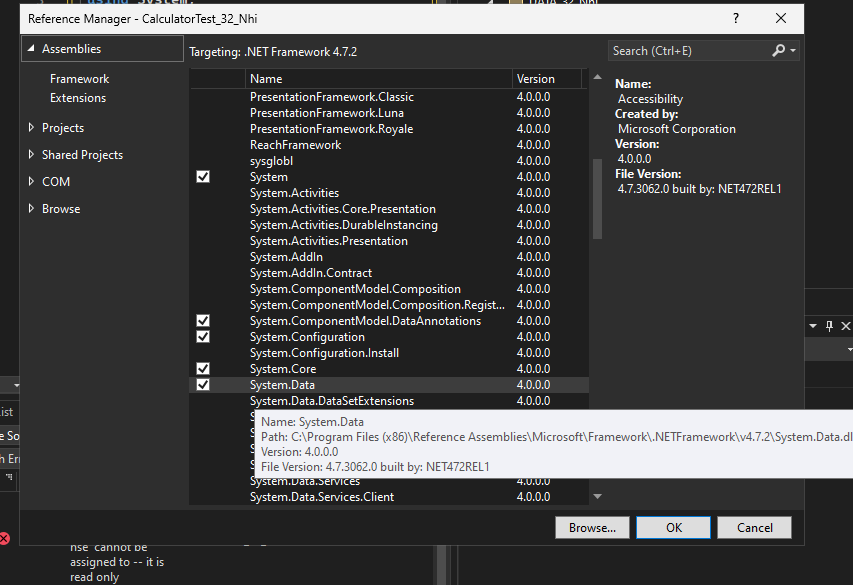
7,6,-3

3,-7,-10

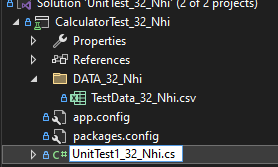
5,5,0

### +Thêm reference System.Data vào project test





Đổi tên



## Bước 8: Viết code cho calculatorTest\_32\_Nhi

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using UnitTest\_32\_Nhi;

namespace CalculatorTester\_32\_Nhi

{

[TestClass]

public class UnitTest1\_32\_Nhi

{

private Calculation\_32\_Nhi c\_32\_Nhi;

public TestContext TestContext { get; set; }

[TestInitialize] // thiet lap du lieu dung chung cho TC

public void SetUp\_32\_Nhi()

{

c\_32\_Nhi = new Calculation\_32\_Nhi(6, 3);

}

[TestMethod]**//TC1: a\_32\_Nhi =6, b\_32\_Nhi = 3, kq\_32\_Nhi= 3**

public void Test\_Cong\_32\_Nhi()

{

int expected\_32\_Nhi, actual\_32\_Nhi;

expected\_32\_Nhi = 9;//kết quả kì vọng

actual\_32\_Nhi = c\_32\_Nhi.Execute\_32\_Nhi("GCD");//kết quả thực tế

Assert.AreEqual(expected\_32\_Nhi, actual\_32\_Nhi);// vì expected\_32\_Nhi == actual\_32\_Nhi(9==3) nên kết quả khi chạy unit test sẽ là fail

}

[TestMethod]**//TC2: a\_32\_Nhi =6, b\_32\_Nhi = 3, kq\_32\_Nhi= 3**

public void Test\_Tru\_32\_Nhi()

{

int expected\_32\_Nhi, actual\_32\_Nhi;

expected\_32\_Nhi = 3;//kết quả kì vọng

actual\_32\_Nhi = c\_32\_Nhi.Execute\_32\_Nhi("GCD");//kết quả thực tế

Assert.AreEqual(expected\_32\_Nhi, actual\_32\_Nhi);// vì expected\_32\_Nhi == actual\_32\_Nhi(3==3) nên kết quả khi chạy unit test sẽ là pass

}

[TestMethod]**//TC3: a\_32\_Nhi =6, b\_32\_Nhi = 3, kq\_32\_Nhi= 6**

public void Test\_Cong\_32\_Nhi()

{

int expected\_32\_Nhi, actual\_32\_Nhi;

expected\_32\_Nhi = 9;//kết quả kì vọng

actual\_32\_Nhi = c\_32\_Nhi.Execute\_32\_Nhi("LCM");//kết quả thực tế

Assert.AreEqual(expected\_32\_Nhi, actual\_32\_Nhi);// vì expected\_32\_Nhi == actual\_32\_Nhi(9==6) nên kết quả khi chạy unit test sẽ là fail

}

[TestMethod]**//TC4: a\_32\_Nhi =6, b\_32\_Nhi = 3, kq\_32\_Nhi= 6**

public void Test\_Tru\_32\_Nhi()

{

int expected\_32\_Nhi, actual\_32\_Nhi;

expected\_32\_Nhi = 3;//kết quả kì vọng

actual\_32\_Nhi = c\_32\_Nhi.Execute\_32\_Nhi("LCM");//kết quả thực tế

Assert.AreEqual(expected\_32\_Nhi, actual\_32\_Nhi);// vì expected\_32\_Nhi == actual\_32\_Nhi(3==6) nên kết quả khi chạy unit test sẽ là pass

}

[TestMethod]

//Lien ket Testdata voi project

[DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV",

@".\DATA\_32\_Nhi\TestData\_32\_Nhi.csv", "TestData\_32\_Nhi#csv", DataAccessMethod.Sequential)]

public void TestWithDataSource\_32\_Nhi\_3Col\_3pass\_1failse()

{

int a\_32\_Nhi = int.Parse(TestContext.DataRow[0].ToString());

int b\_32\_Nhi = int.Parse(TestContext.DataRow[1].ToString());

int expected\_32\_Nhi = int.Parse(TestContext.DataRow[2].ToString());

Calculation\_32\_Nhi c\_32\_Nhi = new Calculation\_32\_Nhi(a\_32\_Nhi, b\_32\_Nhi);

int actual\_32\_Nhi = c\_32\_Nhi.Execute\_32\_Nhi("GCD");

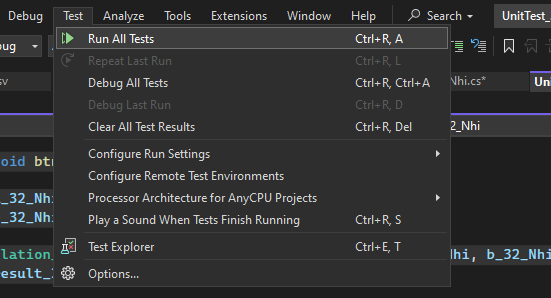
Assert.AreEqual(expected\_32\_Nhi, actual\_32\_Nhi);

}

}

}

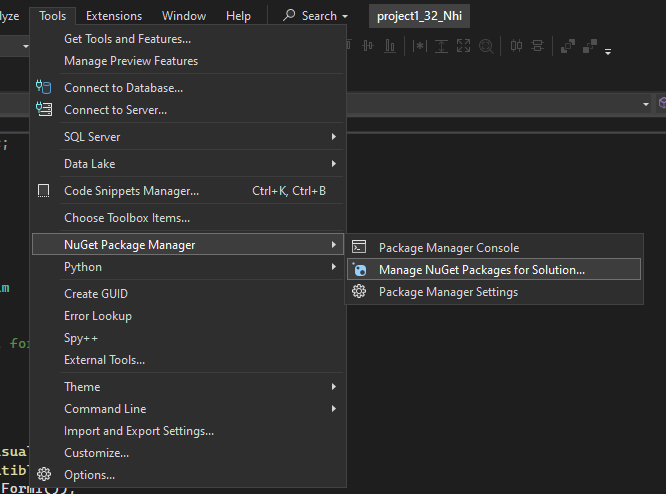
## Bước 9: Chạy kiểm thử



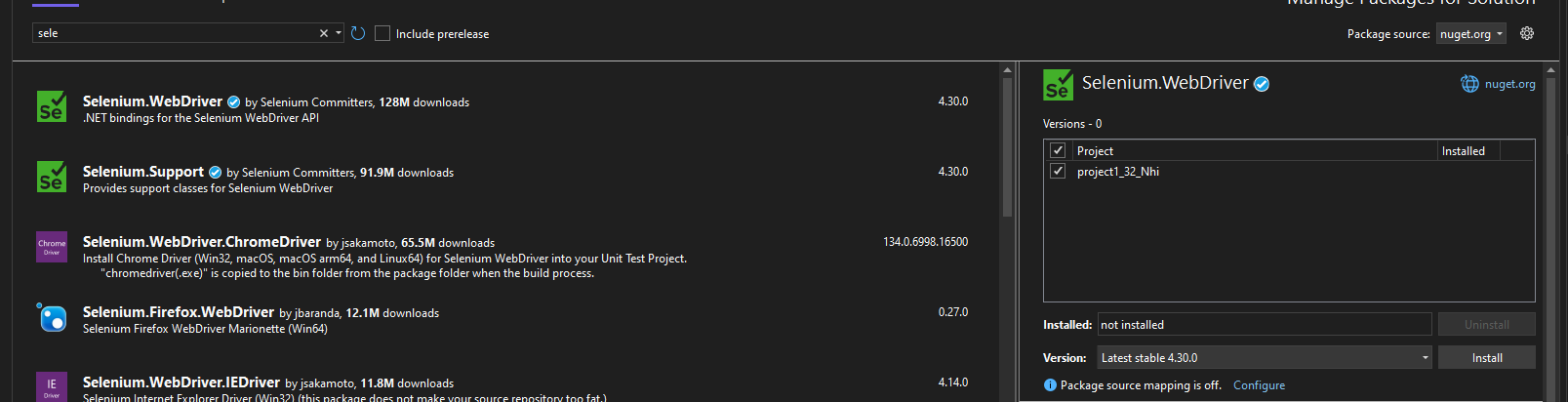
# Bài 2:SELENIUM

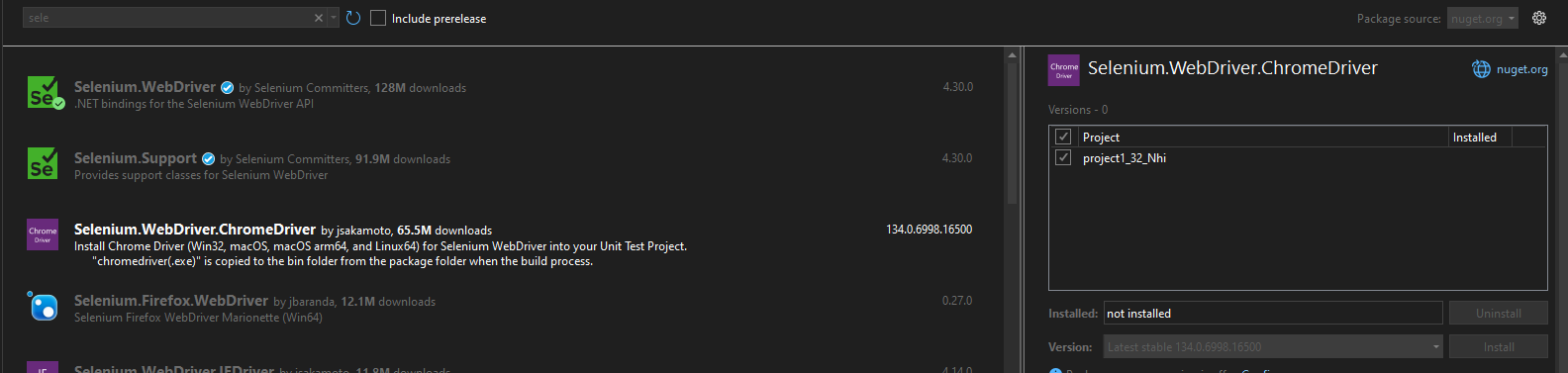
## Cài đặt Selenium Web Driver

▪ Tại project C#, chọn menu Tools > NuGet Package Manager > Manage NuGet Packages for Solution…

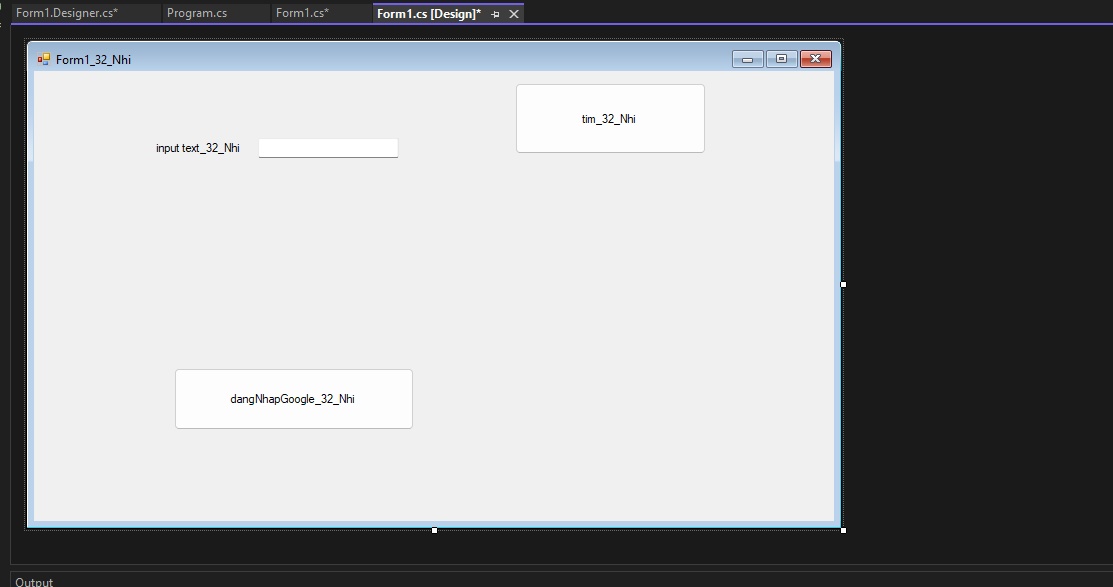


▪ Click tab Browse và tìm kiếm từ khóa Selenium và các web driver muốn kiểm thử để cài đặt



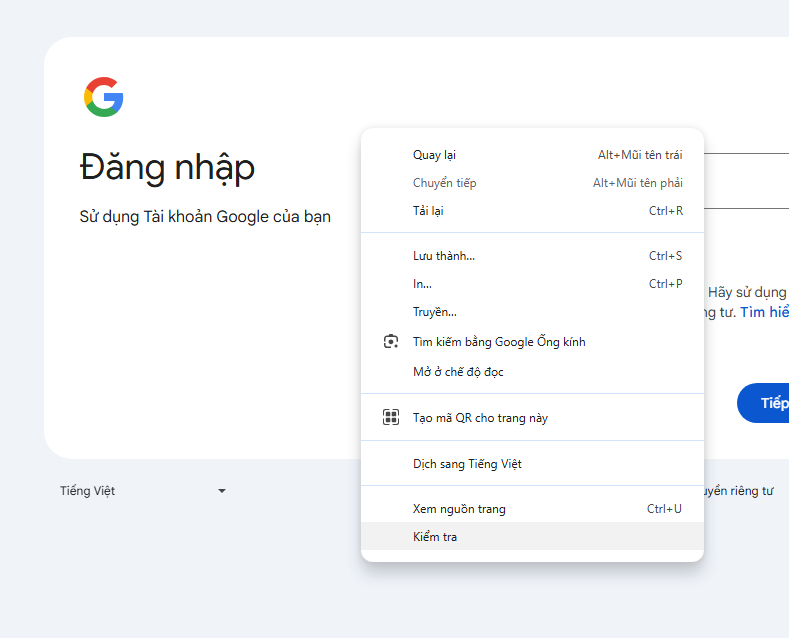


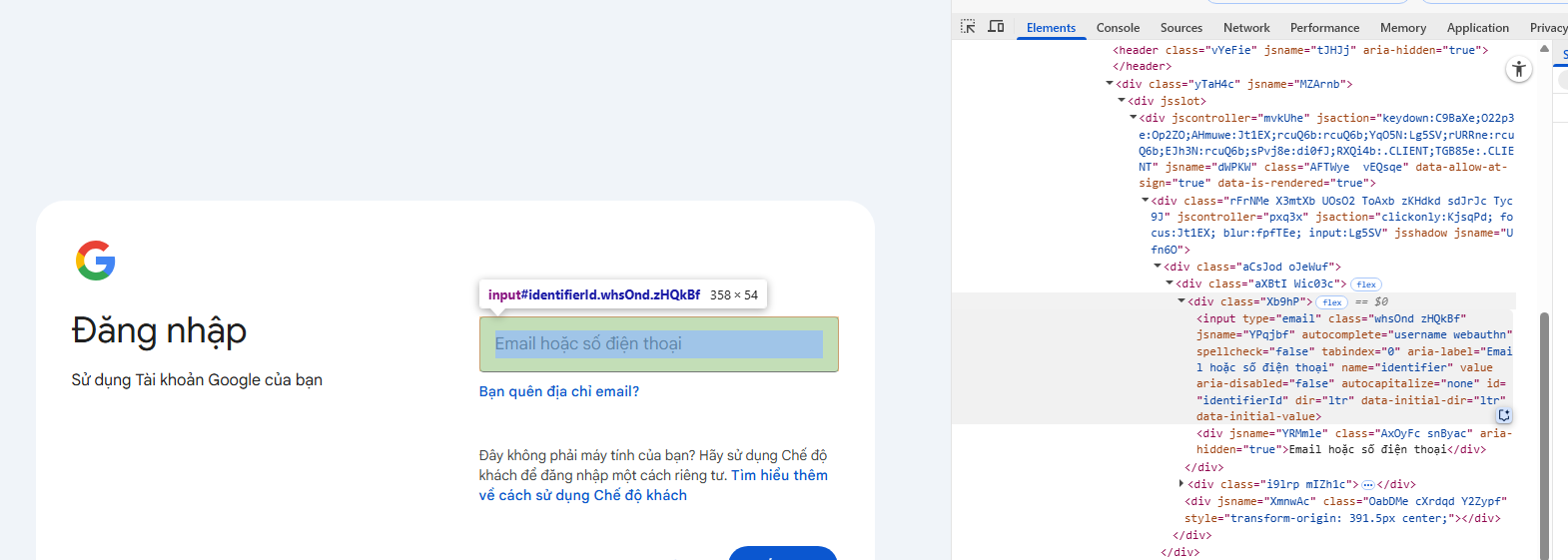
## Giao diện

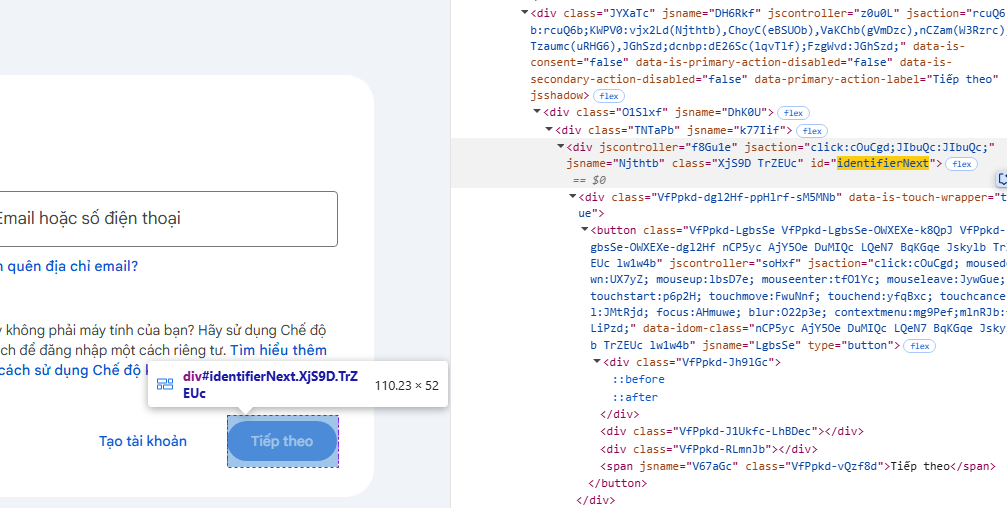


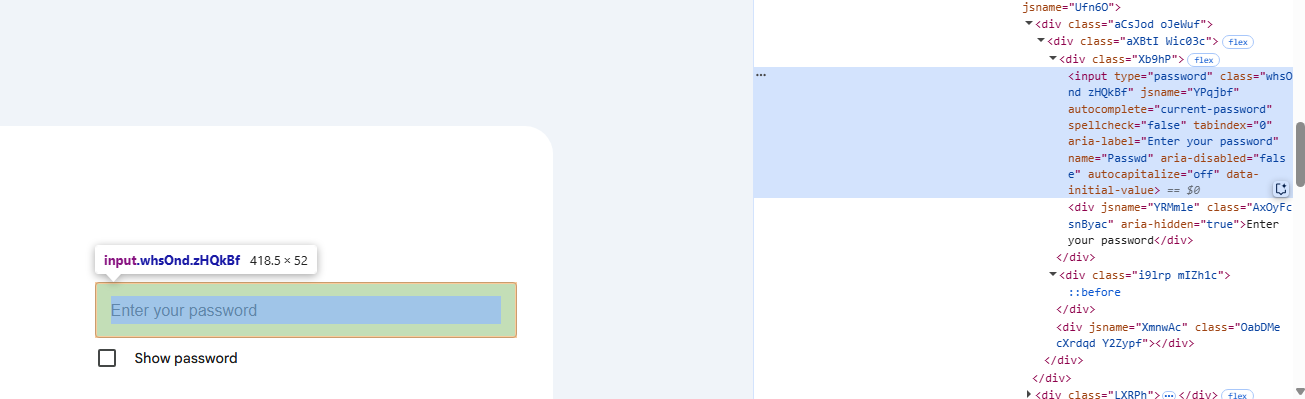
## Lấy thông tin để code:

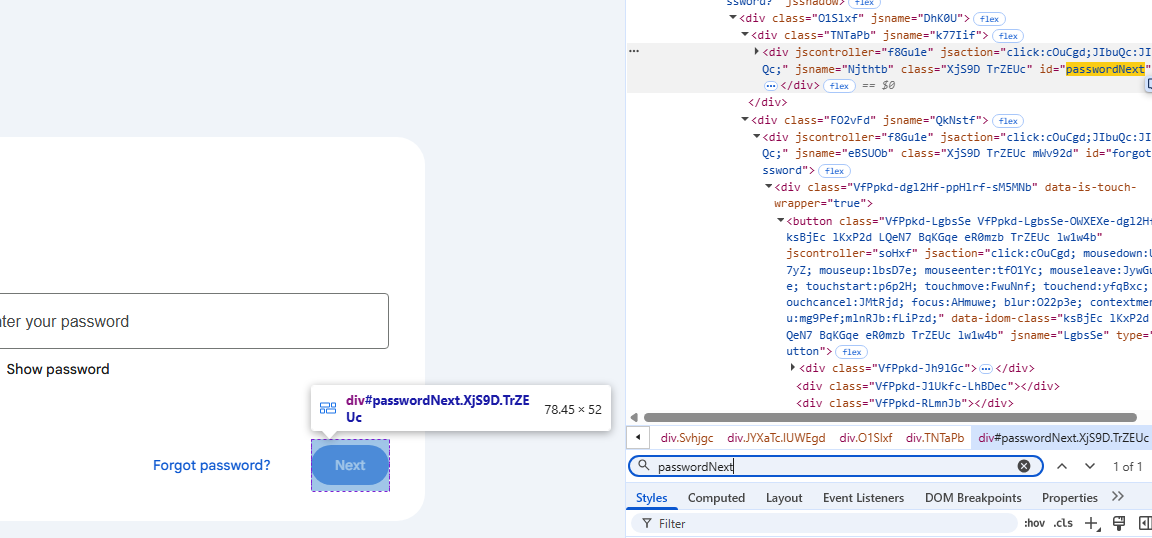
### Đăng nhập:



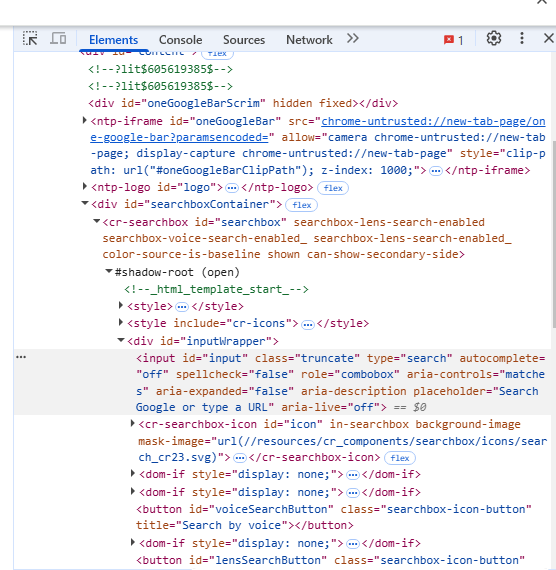








### Tự điền ô tìm kiếm



## Code: đăng nhập google

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

using OpenQA.Selenium.Support.UI;

namespace project1\_32\_Nhi

{

public partial class Form1\_32\_Nhi : Form

{

public Form1\_32\_Nhi()

{

InitializeComponent();

}

private void btnBrowse\_32\_Nhi\_Click(object sender, EventArgs e)

{

//dieu huong trinh duyet

IWebDriver driver\_32\_Nhi = new ChromeDriver();

driver\_32\_Nhi.Url = "https://accounts.google.com/";

driver\_32\_Nhi.Navigate();

// tự nhập email

IWebElement e\_email\_32\_Nhi = driver\_32\_Nhi.FindElement(By.Name("identifier"));

e\_email\_32\_Nhi.SendKeys("2251012104nhi@ou.edu.vn");

driver\_32\_Nhi.FindElement(By.Id("identifierNext")).Click();

Thread.Sleep(10000);

//tự nhập mật khẩu

IWebElement e\_pass\_32\_Nhi = driver\_32\_Nhi.FindElement(By.Name("Passwd"));

e\_pass\_32\_Nhi.SendKeys("074304001894");

driver\_32\_Nhi.FindElement(By.Id("passwordNext")).Click();

Thread.Sleep(10000);

// Đóng trình duyệt

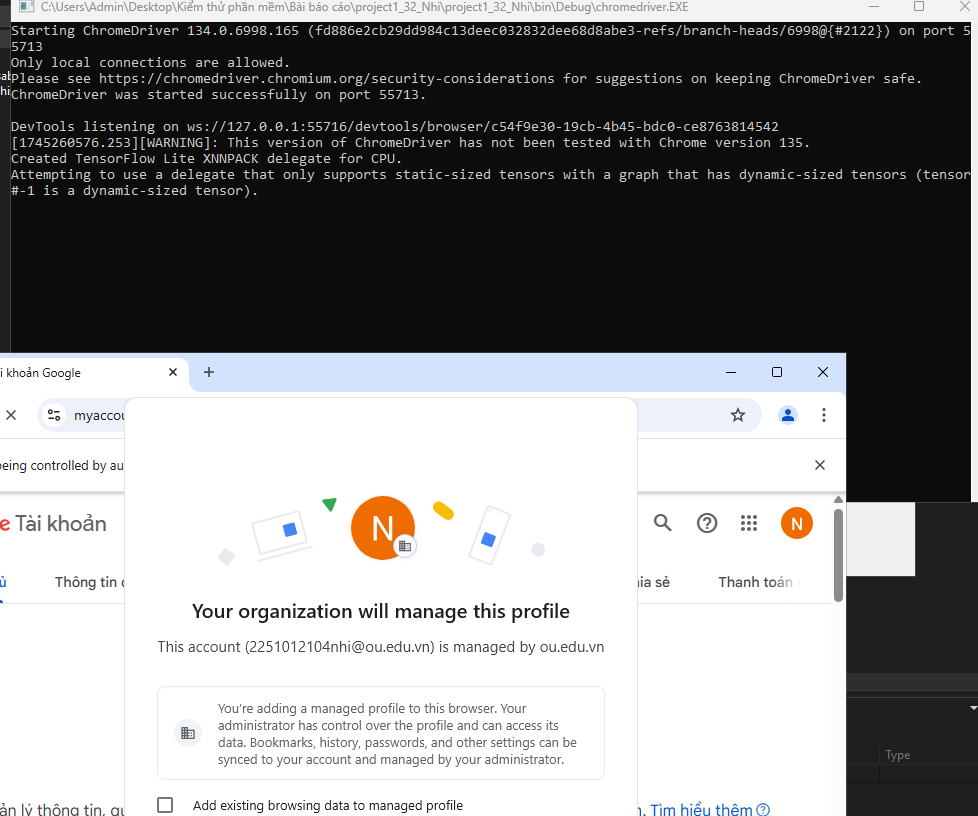
driver\_32\_Nhi.Quit();

}

}

}

## Kết quả: đăng nhập google



## Code: tìm từ

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

using OpenQA.Selenium.Support.UI;

namespace project1\_32\_Nhi

{

public partial class Form1\_32\_Nhi : Form

{

public Form1\_32\_Nhi()

{

InitializeComponent();

}

private void btnBrowse\_32\_Nhi\_Click(object sender, EventArgs e)

{

//dieu huong trinh duyet

IWebDriver driver\_32\_Nhi = new ChromeDriver();

driver\_32\_Nhi.Url = "https://accounts.google.com/";

driver\_32\_Nhi.Navigate();

// tự nhập email

IWebElement e\_email\_32\_Nhi = driver\_32\_Nhi.FindElement(By.Name("identifier"));

e\_email\_32\_Nhi.SendKeys("2251012104nhi@ou.edu.vn");

driver\_32\_Nhi.FindElement(By.Id("identifierNext")).Click();

Thread.Sleep(10000);

//tự nhập mật khẩu

IWebElement e\_pass\_32\_Nhi = driver\_32\_Nhi.FindElement(By.Name("Passwd"));

e\_pass\_32\_Nhi.SendKeys("074304001894");

driver\_32\_Nhi.FindElement(By.Id("passwordNext")).Click();

Thread.Sleep(10000);

// Đóng trình duyệt

driver\_32\_Nhi.Quit();

}

private void btnDich\_32\_Nhi\_Click(object sender, EventArgs e)

{

IWebDriver driver = new ChromeDriver();

driver.Navigate().GoToUrl("https://en.wikipedia.org/wiki/Selenium\_(software)");

string tim\_32\_Nhi=txtinput\_32\_Nhi.Text;

// Tìm phần tử chứa chữ "automation"

IWebElement element = driver.FindElement(By.XPath($"//\*[contains(text(),'{tim\_32\_Nhi}')]"));

// Tô đậm bằng cách thêm viền đỏ

IJavaScriptExecutor js = (IJavaScriptExecutor)driver;

js.ExecuteScript("arguments[0].style.border='3px solid red'", element);

Console.WriteLine("Đã tô đậm phần tử chứa 'automation'");

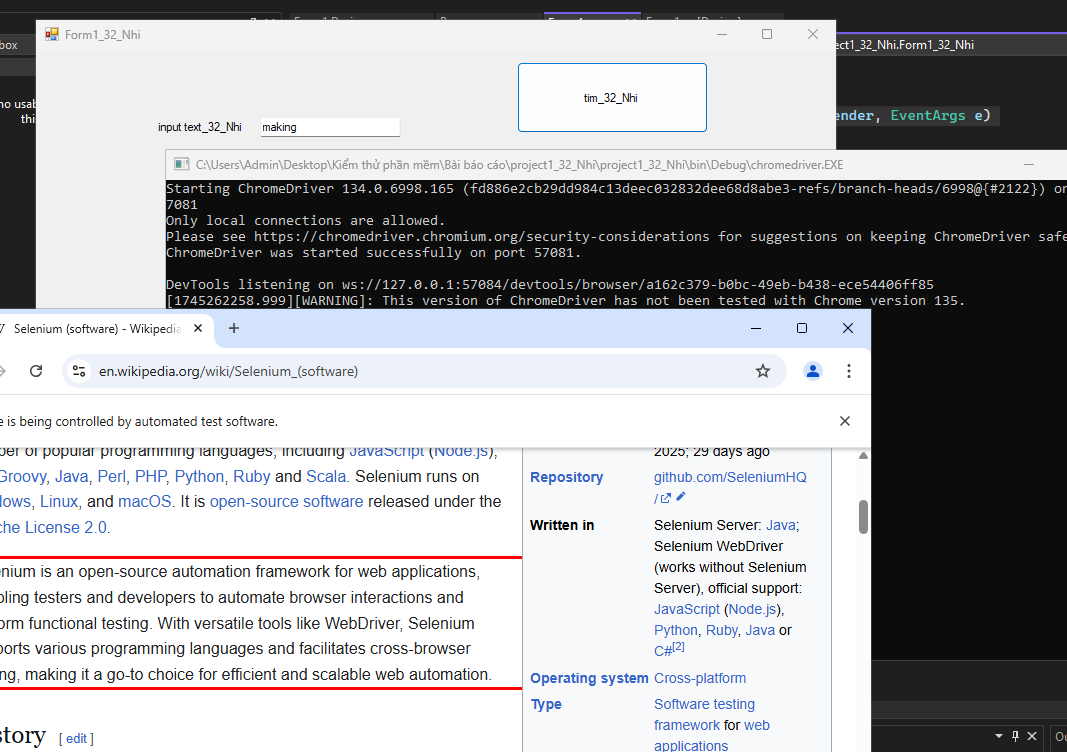
Console.ReadLine(); // giữ trình duyệt mở để xem kết quả

}

}

}

## Kết quả tìm từ:



## Code: Tự động điền vào ô tìm kiếm

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

namespace project1\_32\_Nhi

{

public partial class Form1\_32\_Nhi : Form

{

public Form1\_32\_Nhi()

{

InitializeComponent();

}

private void btnBrowse\_32\_Nhi\_Click(object sender, EventArgs e)

{

//dieu huong trinh duyet

IWebDriver driver\_32\_Nhi = new ChromeDriver();

string a\_32\_Nhi=txtinput\_32\_Nhi.Text;

driver\_32\_Nhi.Url = "https://google.com.vn";

driver\_32\_Nhi.Navigate();

IWebElement element\_32\_Nhi = driver\_32\_Nhi.FindElement(By.Name("q"));

element\_32\_Nhi.SendKeys(a\_32\_Nhi);

}

}

}

## Kết quả

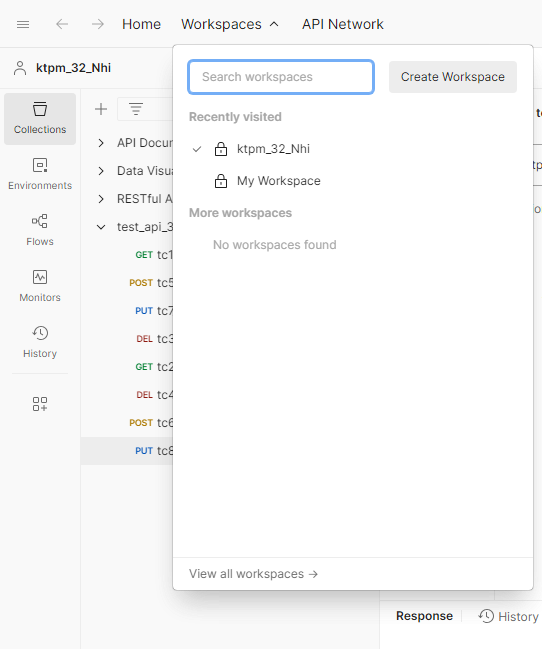
## :

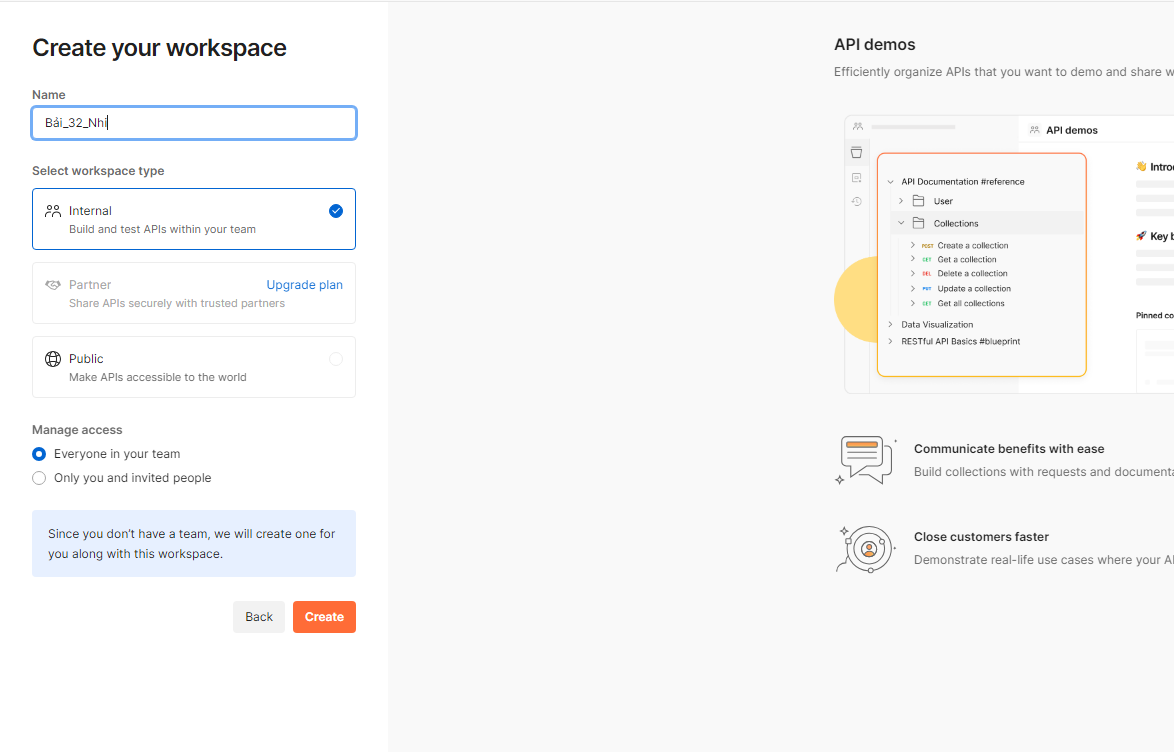
# Bài 3:API

## +Bước 1:Download và cài đặt postman

vào link <https://www.postman.com/downloads/> để tải postman về

## +Bước 2: Tạo 1 workspace mới



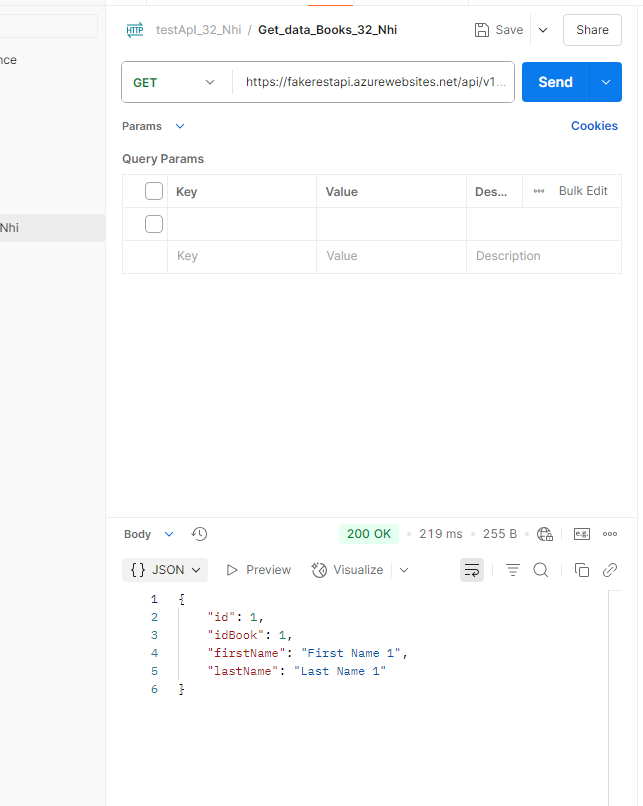


## +Bước 3 : Kiểm thử

### Với get:

lấy dữ liệu của author có id=1 qua <https://fakerestapi.azurewebsites.net/api/v1/Authors/1>

tạo request:Chọn get -> dán api -> bấm send



Vào scripts code như sau để kiểm thử kết quả trả về của thao tác trên có trả ra author với id=1 không

#### TC1:

pm.**test**("TC1\_check\_idBook\_PASS\_32\_Nhi", **function** () {

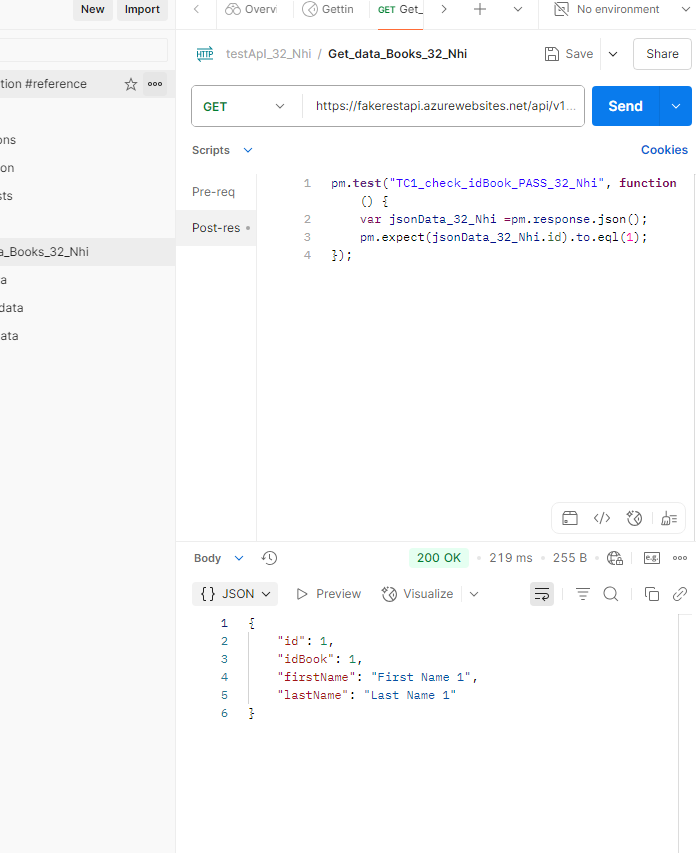
//kiểm tra xem kết quả phản hồi có giá trị id= 1 không

    var jsonData\_32\_Nhi **=**pm.response.json();

    pm.expect(jsonData\_32\_Nhi.id).to.eql(1);

});

//pass do kết quả mong đợi bằng kết quả thực tế ( đều có id=1 )



#### TC2:

pm.**test**("TC2\_check\_idBook\_FAIL\_32\_Nhi", **function** () {

//kiểm tra xem kết quả phản hồi có giá trị id= 2 không

    var jsonData\_32\_Nhi **=**pm.response.json();

    pm.expect(jsonData\_32\_Nhi.id).to.eql(2);

});

//Fail do kết quả mong đợi (id=2)khác kết quả thực tế ( id=1 )

### Với post:

Gửi dữ liệu user lên web:body ->raw->Json

{

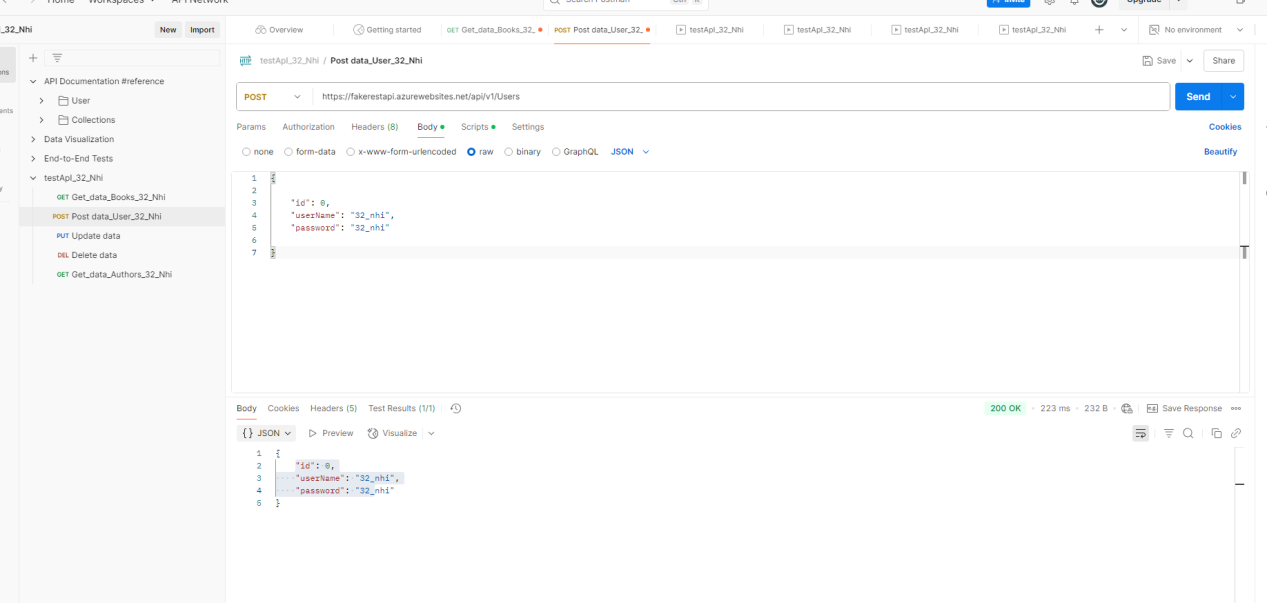
    "id": 0,

    "userName": "32\_nhi",

    "password": "32\_nhi"

}

Kết quả:



#### TC3:

pm.**test**("TC3\_check\_userName\_PASS\_32\_Nhi", **function** () {

//kiểm tra xem kết quả phản hồi có giá trị userName= “32\_nhi” không

    var jsonData\_32\_Nhi **=**pm.response.json();

    pm.expect(jsonData\_32\_Nhi.userName).to.eql("32\_nhi");

});

//pass do kết quả mong đợi bằng kết quả thực tế ( đều có userName="32\_nhi" )

#### TC4:

pm.**test**("TC4\_check\_userName\_FAIL\_32\_Nhi", **function** () {

//kiểm tra xem kết quả phản hồi có giá trị userName= “32\_Nhi\_67JGH” không

    var jsonData\_32\_Nhi **=**pm.response.json();

    pm.expect(jsonData\_32\_Nhi.userName).to.eql("32\_Nhi\_67JGH");

});

//Fail do kết quả mong đợi (userName="32\_nhi\_67JGH")khác kết quả thực tế (userName="32\_nhi" )

### Với Put

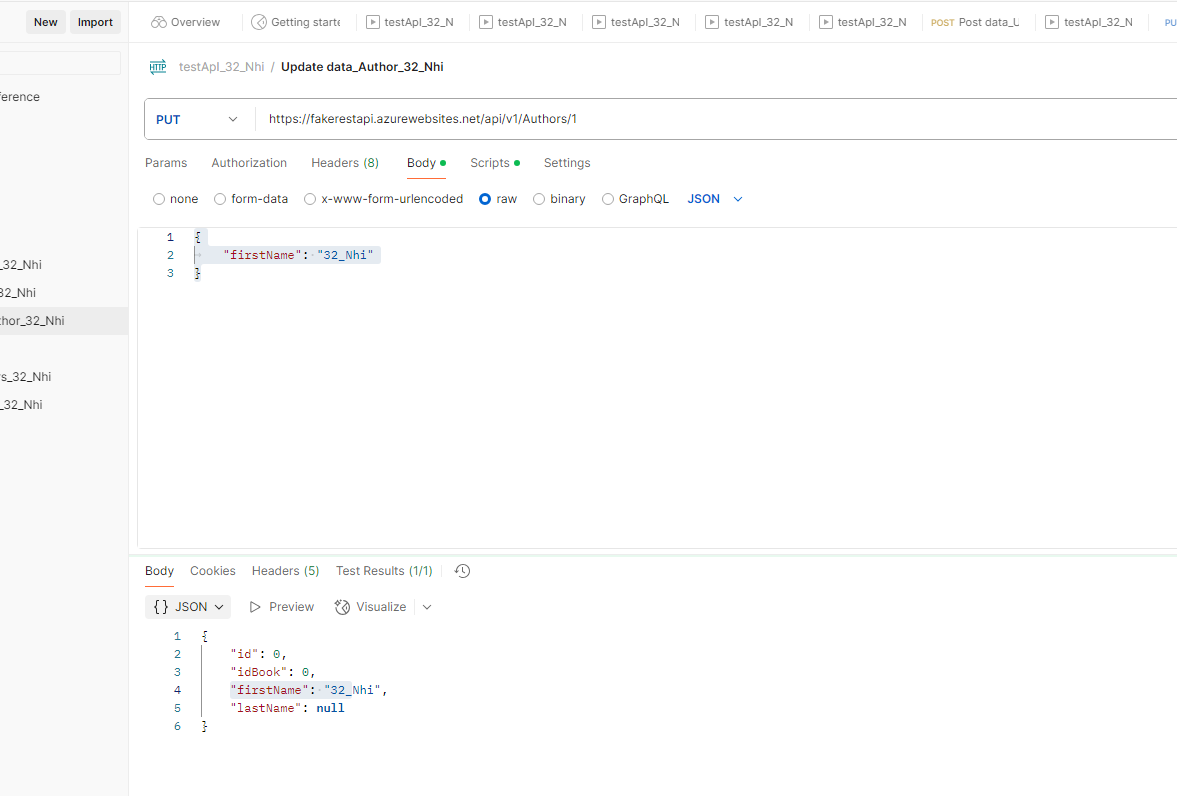
Cập nhật lại dữ liệu: body ->raw->Json

{

    "firstName": "32\_Nhi"

}

Kết quả:



#### TC5:

pm.**test**("TC5\_check\_updateAuthor\_PASS\_32\_Nhi", **function** () {

//kiểm tra xem kết quả phản hồi có giá trị firstName= “32\_nhi” không

    var jsonData\_32\_Nhi **=**pm.response.json();

    pm.expect(jsonData\_32\_Nhi.firstName).to.eql("32\_Nhi");

});

//pass do kết quả mong đợi bằng kết quả thực tế ( đều có firstName="32\_nhi" )

#### TC6:

pm.**test**("TC6\_check\_updateAuthor\_FAIL\_32\_Nhi", **function** () {

//kiểm tra xem kết quả phản hồi có giá trị firstName= “abc\_32\_nhi” không

var jsonData\_32\_Nhi **=**pm.response.json();

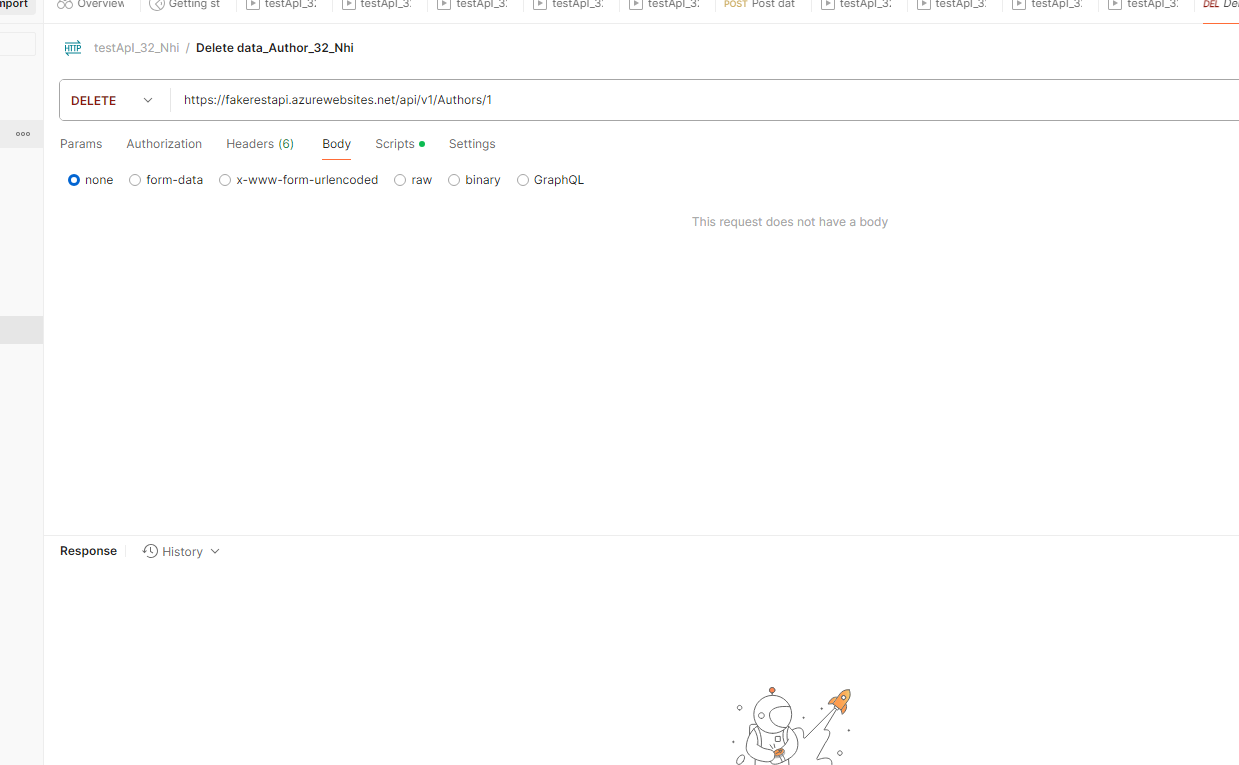
    pm.expect(jsonData\_32\_Nhi.firstName).to.eql("abc\_32\_Nhi");

});

//Fail do kết quả mong đợi (firstName="abc\_32\_nhi")khác kết quả thực tế (firstName="32\_nhi" )

### Với delete

Xoá 1 author:



#### TC7:

pm.**test**("TC7\_check\_deleteAuthor\_PASS\_32\_Nhi", **function** () {

    // Kiểm tra mã trạng thái phản hồi từ API là 204 (No Content)

    pm.response.to.have.status(204);

});

pass do kết quả mong đợi bằng kết quả thực tế ( phản hồi trạng thái 204)

#### TC8:

pm.**test**("TC8\_check\_deleteAuthor\_Fail\_32\_Nhi", **function** () {

// Kiểm tra mã trạng thái phản hồi từ API là 200(OK)

    pm.response.to.have.status(200);

});

//Fail do kết quả mong đợi (phản hồi trạng thái 200)khác kết quả thực tế (phản hồi trạng thái 204 )

Kết quả chạy test:

Chọn testApi\_32\_Nhi-> nhấn run->Nhấn run testApi\_32\_Nhi

