51\_Diễm & 02\_Đăng

MỤC LỤC

[CHƯƠNG 1: MS UNIT VÀ NUNIT 4](#_Toc195163721)

[1. Tạo Project C# để kiểm tra một tam giác có phải là tam giác cân không 4](#_Toc195163722)

[1.1. Tạo Windows Forms App (.NET Framework): 4](#_Toc195163723)

[1.2. Tạo giao diện trong form\_51\_Diem\_02\_Dang: 7](#_Toc195163724)

[1.3. Tạo file TriangleChecker\_51\_Diem\_02\_Dang.cs: 8](#_Toc195163725)

[1.4. Vào form\_51\_Diem\_02\_Dang code lệnh sau: 13](#_Toc195163726)

[2. Tạo project Unit Test để kiểm thử: 15](#_Toc195163727)

[2.1. Tạo project TriangleTester\_51\_Diem\_02\_Dang 15](#_Toc195163728)

[2.2. Viết code kiểm thử: 21](#_Toc195163729)

[2.2.1. Tạo class UnitTest\_51\_Diem\_02\_Dang trong project TriangleTester\_51\_Diem\_02\_Dang 21](#_Toc195163730)

[2.2.2. Code kiểm thử: 22](#_Toc195163731)

[2.3. Kết quả: 27](#_Toc195163732)

[2.4. Đặc tả Test Case: 28](#_Toc195163733)

[2.4.1. TC01\_51\_Diem\_02\_Dang: 28](#_Toc195163734)

[2.4.2. TC02\_51\_Diem\_02\_Dang: 29](#_Toc195163735)

[2.4.3. TC03\_51\_Diem\_02\_Dang: 30](#_Toc195163736)

[2.4.4. TC04\_51\_Diem\_02\_Dang: 31](#_Toc195163737)

[2.4.5. TC05\_51\_Diem\_02\_Dang: 32](#_Toc195163738)

[2.4.6. TC06\_51\_Diem\_02\_Dang: 33](#_Toc195163739)

[2.4.7. TC07\_51\_Diem\_02\_Dang: 34](#_Toc195163740)

[2.4.8. TC08\_51\_Diem\_02\_Dang: 35](#_Toc195163741)

[CHƯƠNG 2: WEBDRIVER SELENIUM 37](#_Toc195163742)

[Cài đặt Selenium 37](#_Toc195163743)

[1. Chức năng đăng nhập 39](#_Toc195163744)

[2. Chức năng đăng xuất 41](#_Toc195163745)

[3. Chức năng tìm bài đầu tiên theo Textbox 42](#_Toc195163746)

[4. Chức năng chọn bài top 100 45](#_Toc195163747)

[5. Chức năng tải nhạc 47](#_Toc195163748)

[6. Chức năng chia sẻ 48](#_Toc195163749)

[CHƯƠNG 3: API – POSTMAN 49](#_Toc195163750)

[1. Cài đặt JSON SERVER: 49](#_Toc195163751)

[1.1. Vào link: https://nodejs.org/en/download 49](#_Toc195163752)

[1.2. Mở cmd: Cài đặt gói sử dụng NODE package manager (NPM) 49](#_Toc195163753)

[1.3. Kết quả: 52](#_Toc195163754)

[2. Tạo file JSON thành API trên server ảo: 52](#_Toc195163755)

[2.1. Tạo file: Trong Visual Studio Code 52](#_Toc195163756)

[2.2. Mở file package.json, thêm dòng start: 54](#_Toc195163757)

[2.3. Mở cmd -> Chạy npm -> Tạo API server 55](#_Toc195163758)

[2.4. Mở Chrome -> Mở link http://localhost:3000/51\_Diem\_02\_Dang 56](#_Toc195163759)

[3. Truy cập vào Postman để test API vừa tạo: 57](#_Toc195163760)

[3.1. Tạo request GET: 57](#_Toc195163761)

[3.1.1. Tạo Workspace: 57](#_Toc195163762)

[3.1.2. Chọn GET -> Nhập link http://localhost:3000/51\_Diem\_02\_Dang -> Nhấn Send 60](#_Toc195163763)

[3.1.3. Thông tin của API: 60](#_Toc195163764)

[3.1.4. Đặt tên -> Nhấn Save -> Lưu vào Project 61](#_Toc195163765)

[3.1.5. Ta có thể lọc theo Name: 61](#_Toc195163766)

[3.1.6. Viết các test case trong test script: 61](#_Toc195163767)

[3.2. Tạo request POST: 63](#_Toc195163768)

[3.2.1. Sử dụng Workspace: 51\_Diem\_02\_Dang\_API 63](#_Toc195163769)

[3.2.2. Nhấn dấu “+” để thêm một request mới: 64](#_Toc195163770)

[3.2.3. Chọn POST -> Nhập link http://localhost:3000/51\_Diem\_02\_Dang 64](#_Toc195163771)

[3.2.4. Chọn Body -> Chọn raw -> Chọn JSON 64](#_Toc195163772)

[3.2.5. Nhập dữ liệu mới muốn gửi lên server -> Nhấn Send 64](#_Toc195163773)

[3.2.6. Dữ liệu được lưu vào server: 65](#_Toc195163774)

[3.2.7. Đặt tên -> Nhấn Save -> Lưu vào Project 66](#_Toc195163775)

[3.2.8. Viết các test case trong test scripts: 67](#_Toc195163776)

[3.3. Tạo request PUT: 69](#_Toc195163777)

[3.3.1. Sử dụng Workspace: 51\_Diem\_02\_Dang\_API 69](#_Toc195163778)

[3.3.2. Nhấn dấu “+” để thêm một request mới: 69](#_Toc195163779)

[3.3.3. Chọn PUT -> Nhập link http://localhost:3000/51\_Diem\_02\_Dang 70](#_Toc195163780)

[3.3.4. Chọn Body -> Chọn raw -> Chọn JSON 70](#_Toc195163781)

[3.3.5. Nhập dữ liệu cập nhật cho dữ liệu có id là “006” lên server 70](#_Toc195163782)

[3.3.6. Sửa link thành http://localhost:3000/51\_Diem\_02\_Dang/006 (có id là 006) -> Nhấn Send 71](#_Toc195163783)

[3.3.7. Dữ liệu được lưu vào server: 71](#_Toc195163784)

[3.3.8. Đặt tên -> Nhấn Save -> Lưu vào project 74](#_Toc195163785)

[3.3.9. Viết các test case trong test scripts: 74](#_Toc195163786)

[3.4. Tạo request DELETE: 76](#_Toc195163787)

[3.4.1. Sử dụng Workspace: 51\_Diem\_02\_Dang\_API 76](#_Toc195163788)

[3.4.2. Nhấn dấu “+” để thêm một request mới: 76](#_Toc195163789)

[3.4.3. Chọn DELETE -> Nhập link http://localhost:3000/51\_Diem\_02\_Dang 76](#_Toc195163790)

[3.4.4. Sửa link thành http://localhost:3000/51\_Diem\_02\_Dang/007 (có id là 007) -> Nhấn Send để xóa 77](#_Toc195163791)

[3.4.5. Dữ liệu được lưu vào server: 77](#_Toc195163792)

[3.4.6. Đặt tên -> Nhấn Save -> Lưu vào Project 80](#_Toc195163793)

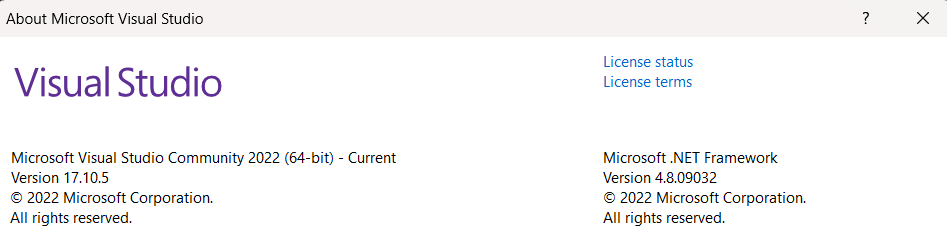
[3.4.7. Viết các test case trong test scripts: 80](#_Toc195163794)

# CHƯƠNG 1: MS UNIT VÀ NUNIT

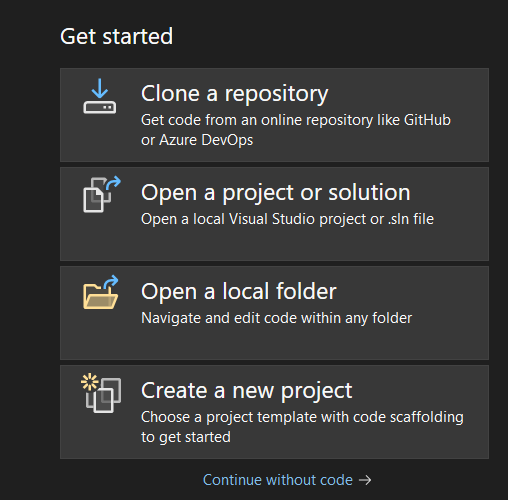
## Tạo Project C# để kiểm tra một tam giác có phải là tam giác cân không

### Tạo Windows Forms App (.NET Framework):

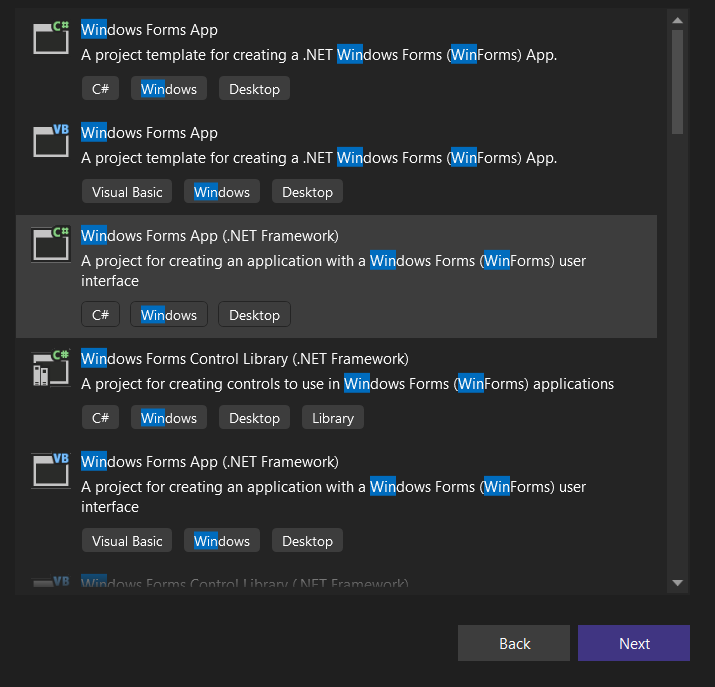
Vào Microsoft Visual Studio Community 2022 (64 bit) – Version 17.10.5



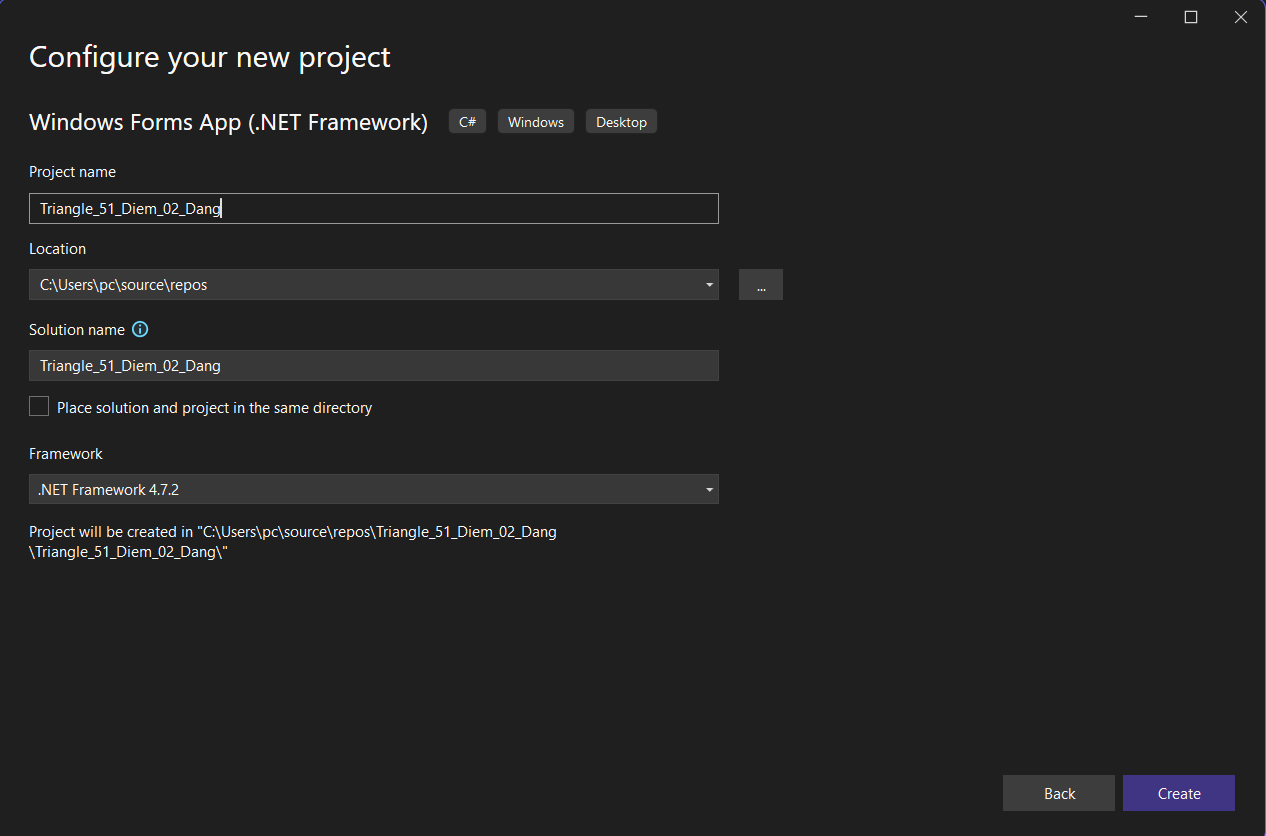
Chọn Create a new project



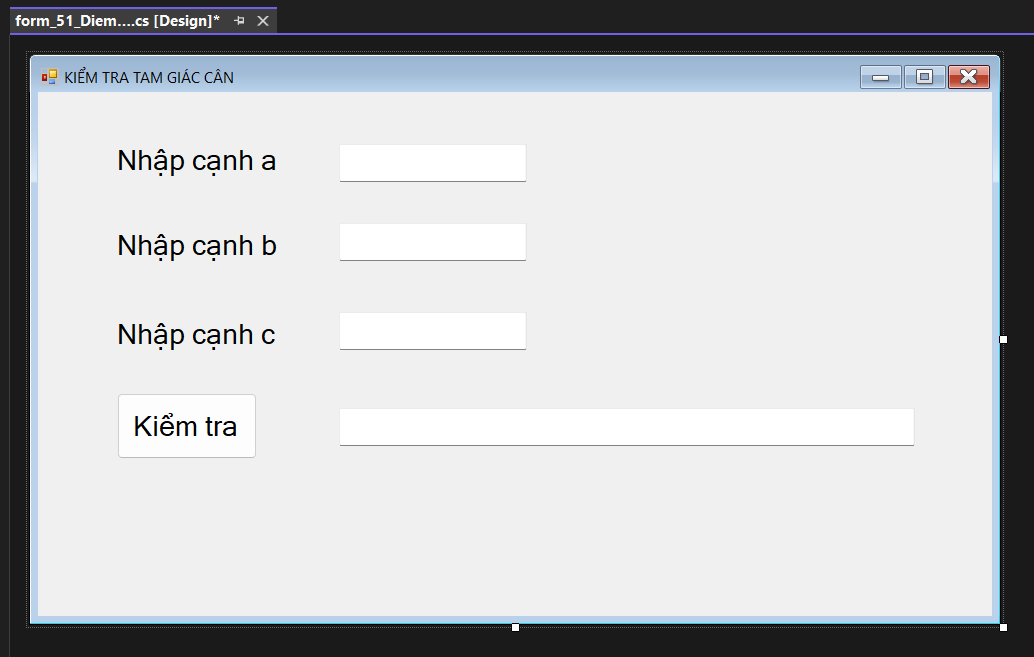
Tìm Windows Forms App (.NET Framework) -> Chọn Next



Đặt tên project là Triangle\_51\_Diem\_02\_Dang -> Chọn Create



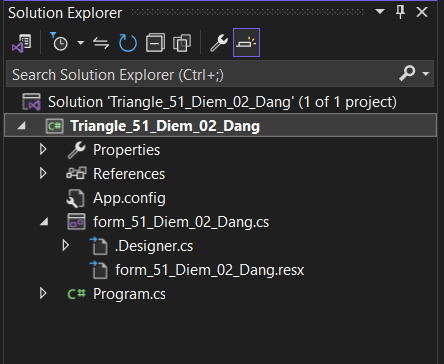
### Tạo giao diện trong form\_51\_Diem\_02\_Dang:



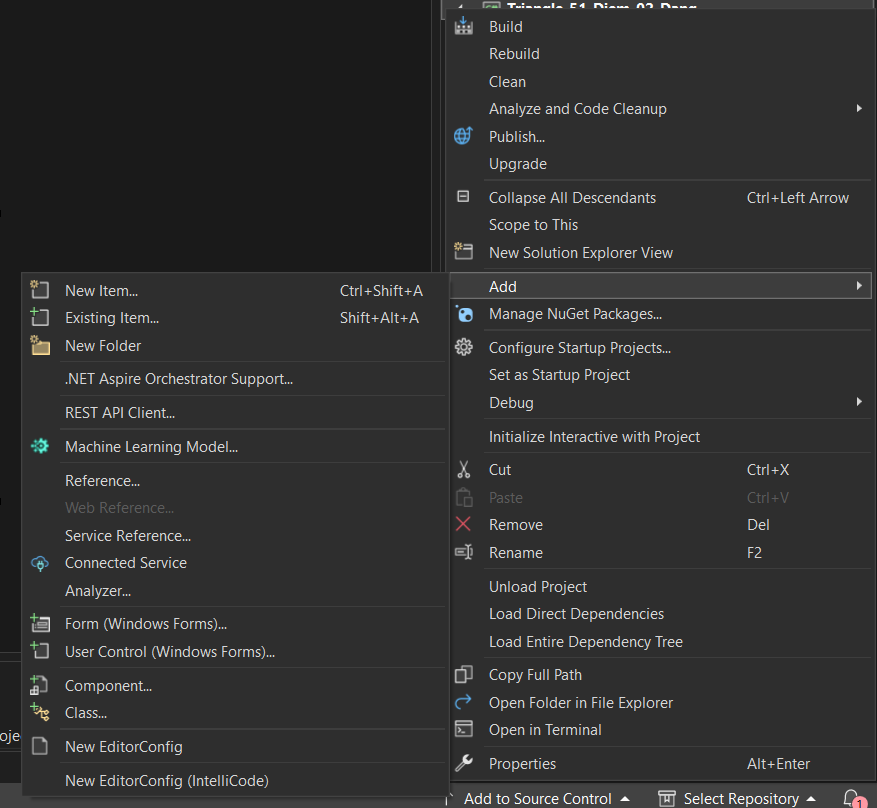
### Tạo file TriangleChecker\_51\_Diem\_02\_Dang.cs:

chứa lớp public TriangleChecker\_51\_Diem\_02\_Dang dùng phương thức Execute\_51\_Diem\_02\_Dang để thực hiện kiểm tra một tam giác có phải tam giác cân không

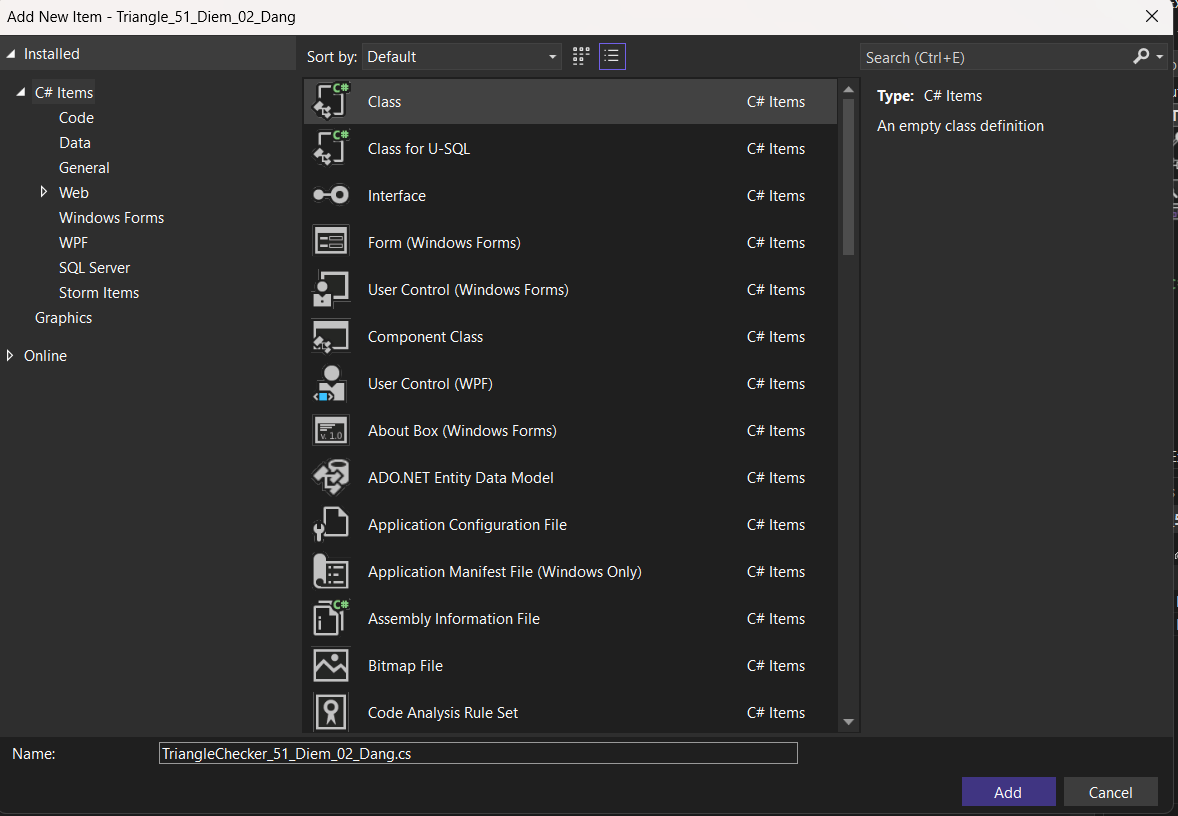
Chọn Triangle\_51\_Diem\_02\_Dang



Chuột phải vào Triangle\_51\_Diem\_02\_Dang -> Chọn Add -> Chọn Class…



Đặt tên Class là TriangleChecker\_51\_Diem\_02\_Dang.cs -> Chọn Add



Trong class TriangleChecker\_51\_Diem\_02\_Dang.cs code đoạn sau:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Triangle\_51\_Diem\_02\_Dang

{

// Định nghĩa một class tên là TriangleChecker\_51\_Diem\_02\_Dang

public class TriangleChecker\_51\_Diem\_02\_Dang

{

// Phương thức Execute nhận vào 3 số thực a, b, c đại diện cho 3 cạnh tam giác

public string Execute\_51\_Diem\_02\_Dang (double a, double b, double c)

{

// Kiểm tra nếu bất kỳ cạnh nào <= 0 thì không hợp lệ

if (a <= 0 || b <= 0 || c <= 0)

{

// Trả về thông báo lỗi

return "Các cạnh phải lớn hơn 0.";

}

// Kiểm tra điều kiện để 3 cạnh tạo thành tam giác: tổng hai cạnh phải lớn hơn cạnh còn lại

if (a + b > c && a + c > b && b + c > a)

{

// Nếu có ít nhất hai cạnh bằng nhau thì là tam giác cân

if (a == b || a == c || b == c)

{

// Trả về thông báo là tam giác cân

return "Đây là tam giác cân.";

}

else

{

// Không có cạnh nào bằng nhau

return "Đây không phải là tam giác cân.";

}

}

else

{

// Nếu không thoả điều kiện tam giác, trả về thông báo lỗi

return "Ba cạnh không tạo thành tam giác.";

}

}

}

}

### Vào form\_51\_Diem\_02\_Dang code lệnh sau:

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;

namespace Triangle\_51\_Diem\_02\_Dang

{

// Định nghĩa lớp Form\_51\_Diem\_02\_Dang kế thừa từ lớp Form của Windows Forms

public partial class form\_51\_Diem\_02\_Dang : Form

{

// Hàm khởi tạo của Form\_51\_Diem\_02\_Dang, sẽ được gọi khi form được tạo ra

public form\_51\_Diem\_02\_Dang()

{

// Hàm này tạo các control trên form (như button, textbox...)

InitializeComponent();

}

// Sự kiện khi người dùng click vào nút "Kiểm tra" (btnCheck\_51\_Diem\_02\_Dang)

private void btnCheck\_51\_Diem\_02\_Dang\_Click(object sender, EventArgs e)

{

// Thử chuyển đổi giá trị nhập vào từ các TextBox thành số thực (double)

if (double.TryParse(txtA\_51\_Diem\_02\_Dang.Text, out double a) &&

double.TryParse(txtB\_51\_Diem\_02\_Dang.Text, out double b) &&

double.TryParse(txtC\_51\_Diem\_02\_Dang.Text, out double c))

{

// Tạo một đối tượng của class TriangleChecker\_51\_Diem\_02\_Dang để gọi phương thức Execute\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

/\* Gọi phương thức Execute\_51\_Diem\_02\_Dang của TriangleChecker\_51\_Diem\_02\_Dang

hiển thị kết quả trên txtResult\_51\_Diem\_02\_Dang \*/

txtResult\_51\_Diem\_02\_Dang.Text = checker.Execute\_51\_Diem\_02\_Dang(a, b, c);

}

else

{

// Nếu không thể chuyển đổi giá trị nhập vào thành số thực, thông báo lỗi

txtResult\_51\_Diem\_02\_Dang.Text = "Vui lòng nhập số hợp lệ cho các cạnh.";

}

}

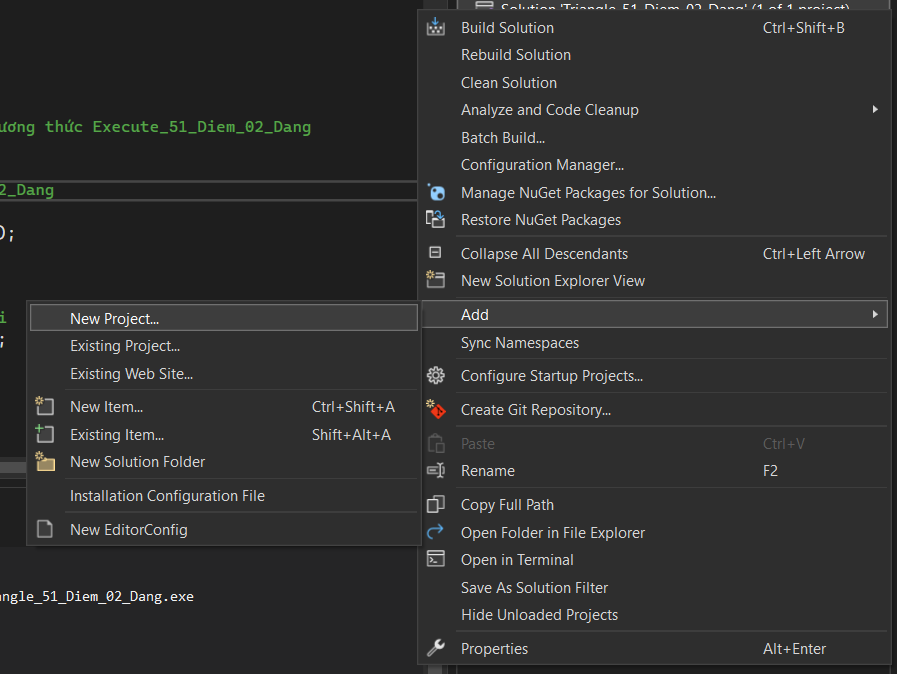
}

}

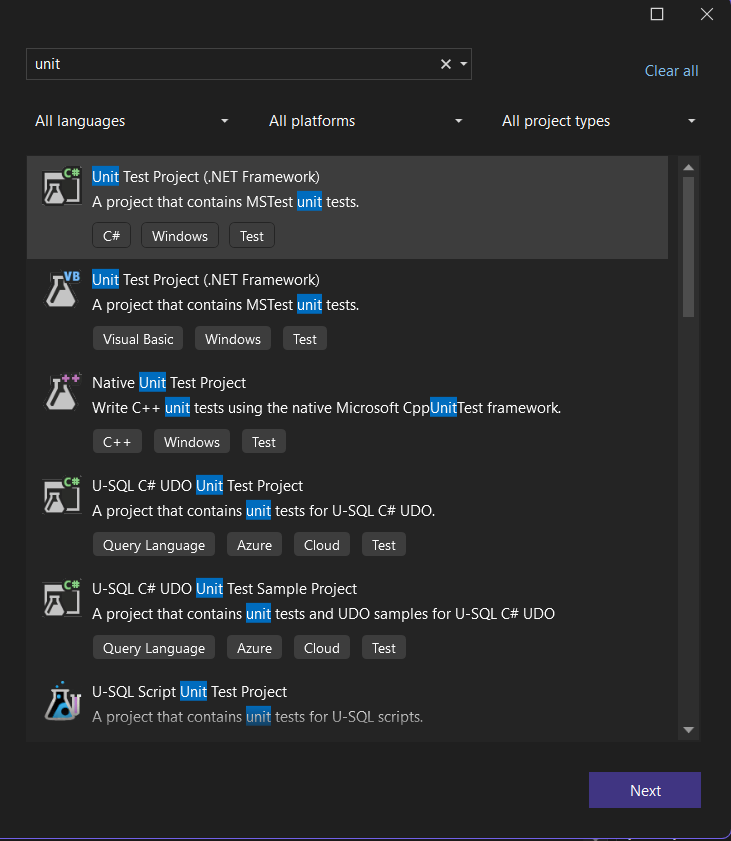
## Tạo project Unit Test để kiểm thử:

### Tạo project TriangleTester\_51\_Diem\_02\_Dang

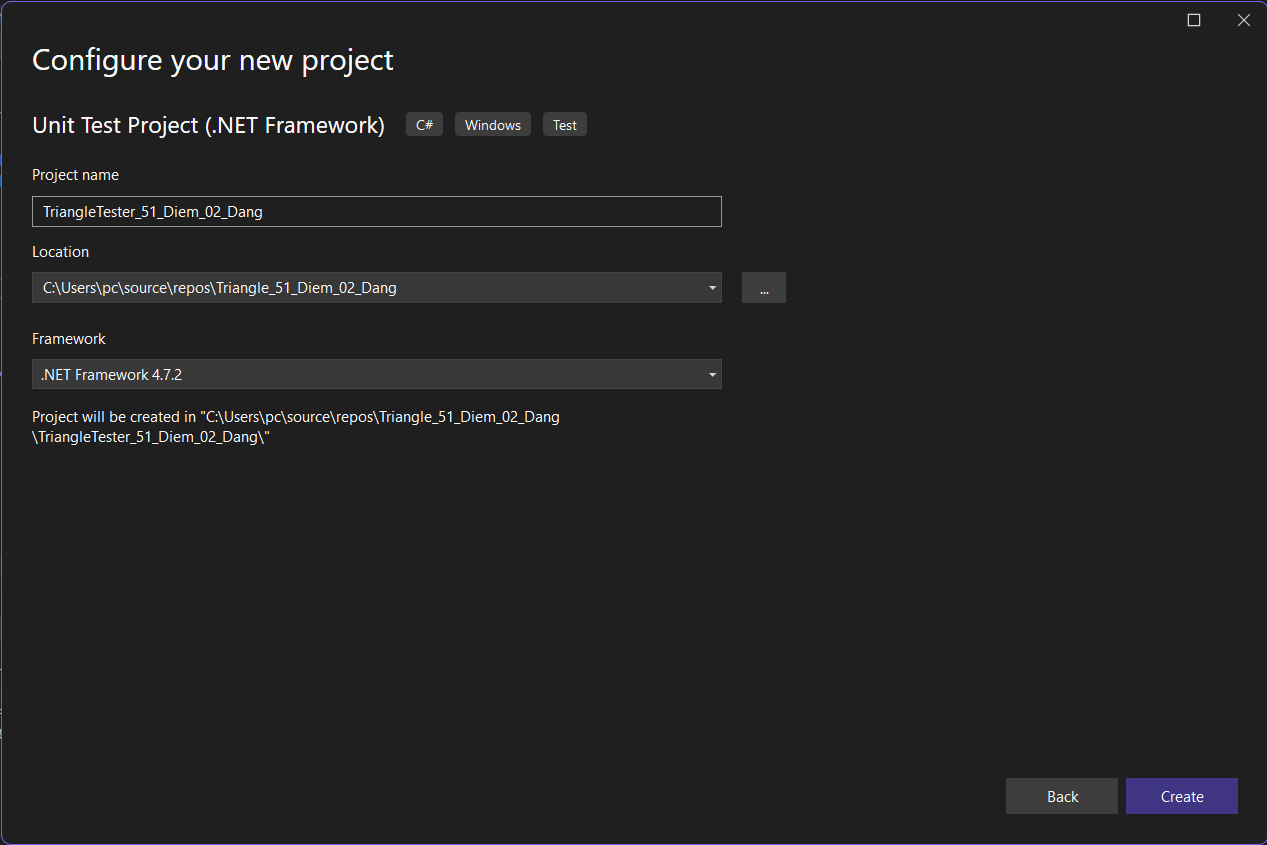
Click chuột phải vào Solution “Triangle\_51\_Diem\_02\_Dang” -> Chọn Add -> Chọn New Project…



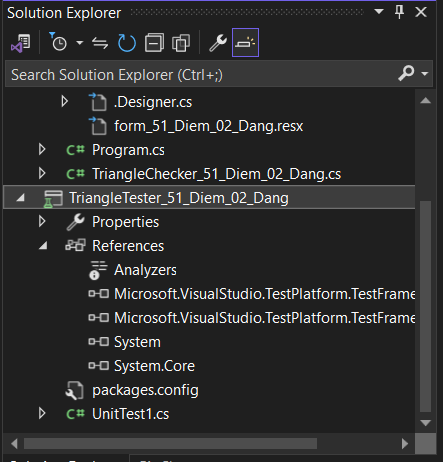
Tìm Unit Test Project (.NET Framework) -> Chọn Next



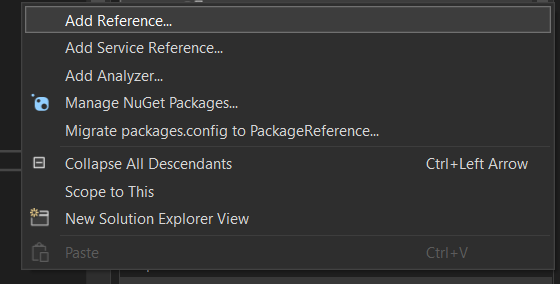
Đặt tên Project name là TriangleTester\_51\_Diem\_02\_Dang -> Chọn Create



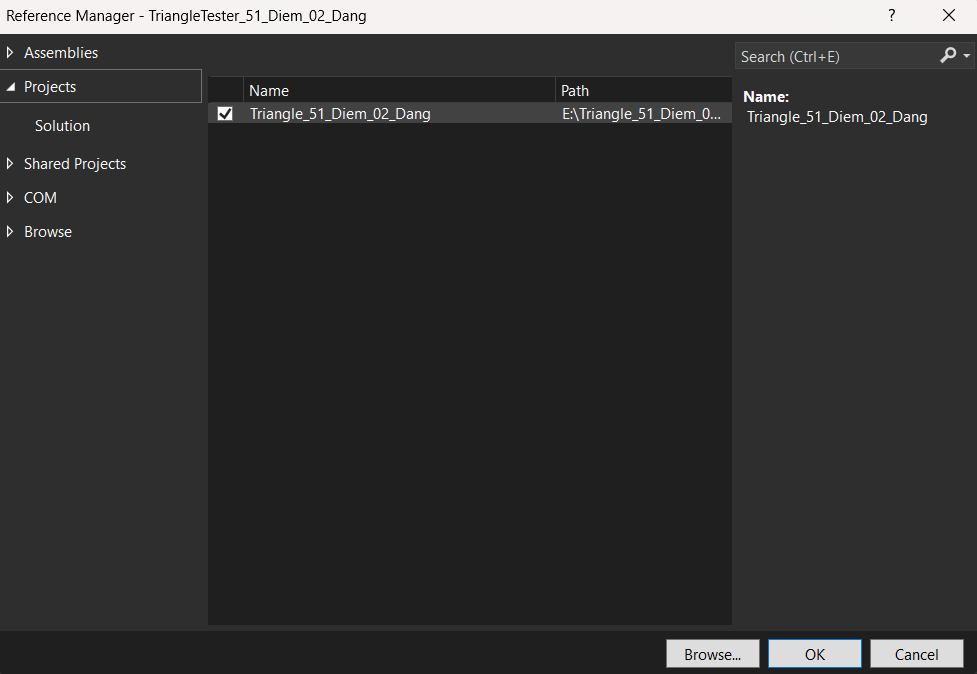
Tại project TriangleTester\_51\_Diem\_02\_Dang

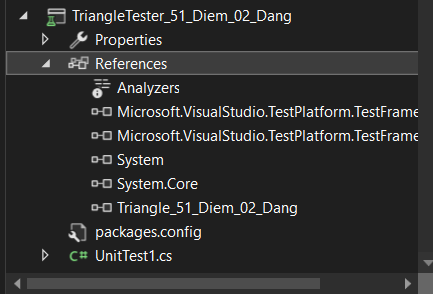


Click chuột phải lên References -> Chọn Add Reference…



Trong Projects -> Tích chọn Triangle\_51\_Diem\_02\_Dang -> Chọn OK



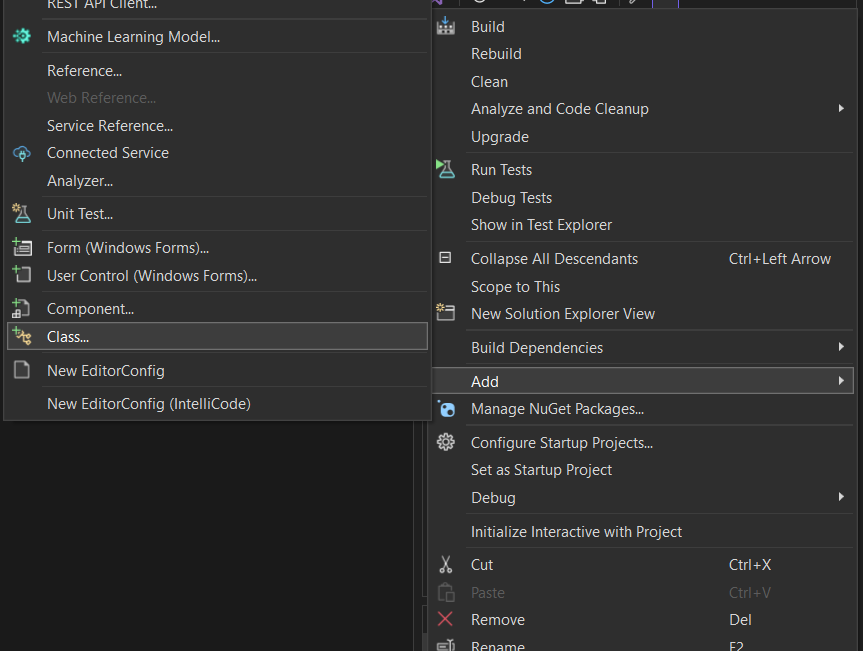


### Viết code kiểm thử:

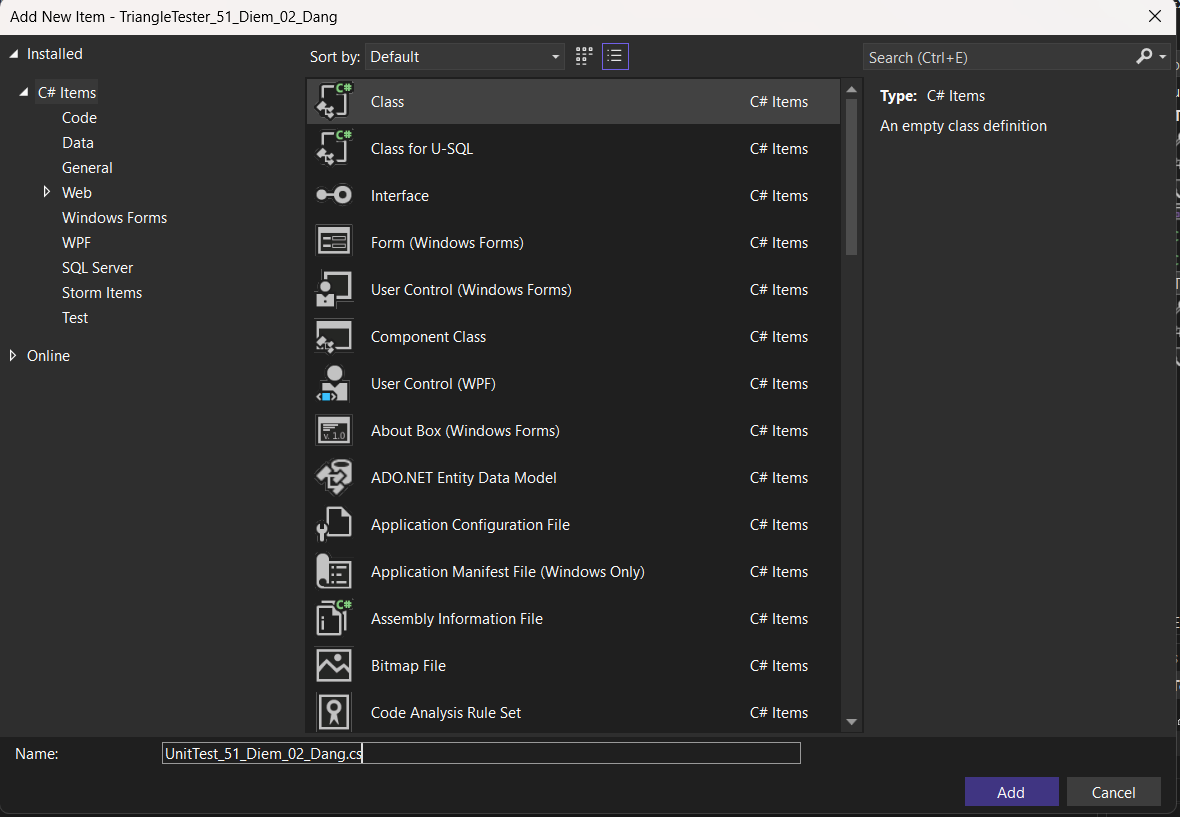
phương thức Execute\_51\_Diem\_02\_Dang trong lớp TriangleChecker\_51\_Diem\_02\_Dang

#### Tạo class UnitTest\_51\_Diem\_02\_Dang trong project TriangleTester\_51\_Diem\_02\_Dang

Chuột phải lên TriangleTester\_51\_Diem\_02\_Dang -> Chọn Add -> Chọn Class…



Chọn Class -> Đặt tên là UnitTest\_51\_Diem\_02\_Dang.cs -> Chọn Add



#### Code kiểm thử:

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Triangle\_51\_Diem\_02\_Dang;

namespace TriangleTester\_51\_Diem\_02\_Dang

{

[TestClass]

public class UnitTest\_51\_Diem\_02\_Dang

{

// Test trường hợp tam giác cân

[TestMethod]

public void TC01\_51\_Diem\_02\_Dang()

{

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với 2 cạnh bằng nhau

// Kết quả ra Pass

var result = checker.Execute\_51\_Diem\_02\_Dang(5, 5, 3);

// Kiểm tra kết quả trả về là "Đây là tam giác cân."

Assert.AreEqual("Đây là tam giác cân.", result);

}

// Test trường hợp fail của TC01\_51\_Diem\_02\_Dang

[TestMethod]

public void TC05\_51\_Diem\_02\_Dang()

{

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với 2 cạnh bằng nhau

// Kết quả ra Fail

var result = checker.Execute\_51\_Diem\_02\_Dang(5, 5, 3);

// Kiểm tra kết quả trả về là "Đây là tam giác cân."

Assert.AreEqual("Đây không phải là tam giác cân.", result);

}

// Test trường hợp tam giác không cân

[TestMethod]

public void TC02\_51\_Diem\_02\_Dang()

{

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với 3 cạnh khác nhau

// Kết quả ra Pass

var result = checker.Execute\_51\_Diem\_02\_Dang(5, 6, 7);

// Kiểm tra kết quả trả về là "Đây không phải là tam giác cân."

Assert.AreEqual("Đây không phải là tam giác cân.", result);

}

// Test trường hợp fail của TC02\_51\_Diem\_02\_Dang

[TestMethod]

public void TC06\_51\_Diem\_02\_Dang()

{

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với 3 cạnh khác nhau

// Kết quả ra Fail

var result = checker.Execute\_51\_Diem\_02\_Dang(5, 6, 7);

// Kiểm tra kết quả trả về là "Đây không phải là tam giác cân."

Assert.AreEqual("Đây là tam giác cân.", result);

}

// Test trường hợp không phải tam giác

[TestMethod]

public void TC03\_51\_Diem\_02\_Dang()

{

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với các cạnh không tạo thành tam giác (ví dụ: 1 + 2 <= 3)

// Kết quả ra Pass

var result = checker.Execute\_51\_Diem\_02\_Dang(1, 2, 3);

// Kiểm tra kết quả trả về là "Ba cạnh không tạo thành tam giác."

Assert.AreEqual("Ba cạnh không tạo thành tam giác.", result);

}

// Test trường hợp fail của TC03\_51\_Diem\_02\_Dang

[TestMethod]

public void TC07\_51\_Diem\_02\_Dang()

{

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với các cạnh không tạo thành tam giác (ví dụ: 1 + 2 <= 3)

// Kết quả ra Fail

var result = checker.Execute\_51\_Diem\_02\_Dang(1, 2, 3);

// Kiểm tra kết quả trả về là "Ba cạnh không tạo thành tam giác."

Assert.AreEqual("Đây là tam giác cân.", result);

}

// Test trường hợp cạnh không hợp lệ (<= 0)

[TestMethod]

public void TC04\_51\_Diem\_02\_Dang() {

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với các cạnh không hợp lệ

// Kết quả ra Pass

var result = checker.Execute\_51\_Diem\_02\_Dang(0, 5, 6);

// Kiểm tra kết quả trả về là "Các cạnh phải lớn hơn 0."

Assert.AreEqual("Các cạnh phải lớn hơn 0.", result);

}

// Test trường hợp cạnh không hợp lệ (<= 0)

[TestMethod]

public void TC08\_51\_Diem\_02\_Dang()

{

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với các cạnh không hợp lệ

// Kết quả ra Pass

var result = checker.Execute\_51\_Diem\_02\_Dang(0, 5, 6);

// Kiểm tra kết quả trả về là "Các cạnh phải lớn hơn 0."

Assert.AreEqual("Đây không phải là tam giác cân.", result);

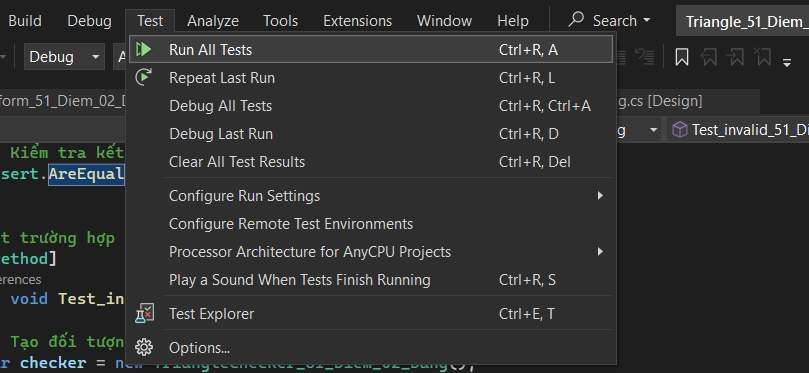
}

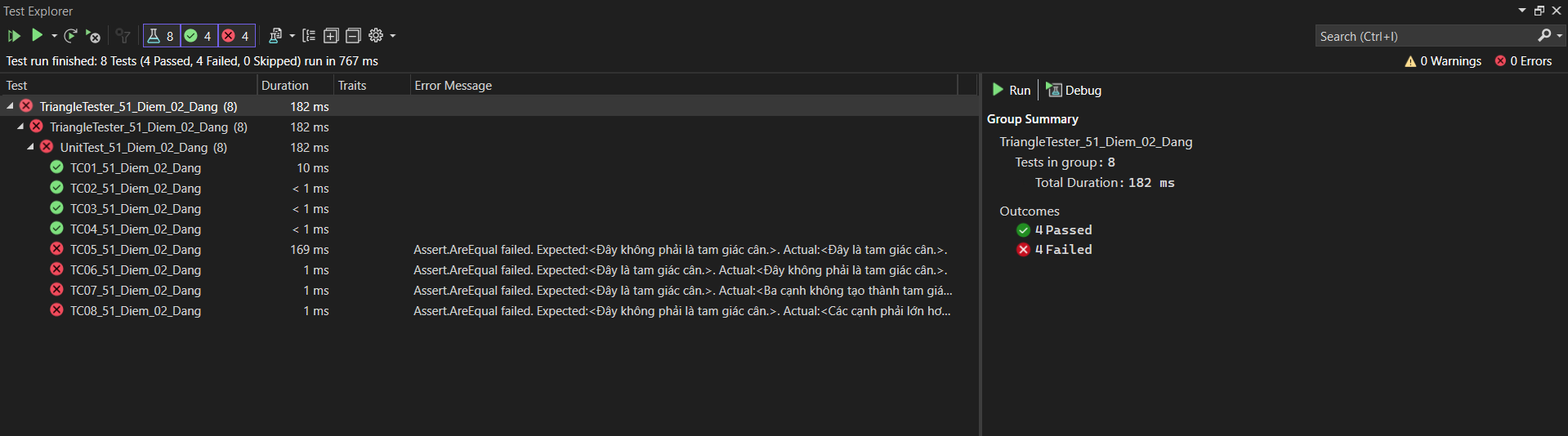
}

}

### Kết quả:

Vào Test -> Chọn Run All Tests



Sau khi chạy: Kết quả ra 4 Passed và 4 Failed

### Đặc tả Test Case:

#### TC01\_51\_Diem\_02\_Dang:

thực hiện kiểm tra một tam giác có phải tam giác cân không. Kết quả trả ra là tam giác cân.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Test Case ID** | **Input** | | | **Kết quả mong đợi** | **Kết quả thực sự** | **Kết quả cuối cùng** |
| a | b | c |
| 1 | TC01\_51\_Diem\_02\_Dang | 5 | 5 | 3 | “Đây là tam giác cân.” | “Đây là tam giác cân.” | Pass |

// Test trường hợp tam giác cân

[TestMethod]

public void TC01\_51\_Diem\_02\_Dang()

{

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với 2 cạnh bằng nhau

// Kết quả ra Pass

var result = checker.Execute\_51\_Diem\_02\_Dang(5, 5, 3);

// Kiểm tra kết quả trả về là "Đây là tam giác cân."

Assert.AreEqual("Đây là tam giác cân.", result);

}

#### TC02\_51\_Diem\_02\_Dang:

thực hiện kiểm tra một tam giác có phải tam giác cân không. Kết quả trả ra là tam giác không cân.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Test Case ID** | **Input** | | | **Kết quả mong đợi** | **Kết quả thực sự** | **Kết quả cuối cùng** |
| a | b | c |
| 2 | TC02\_51\_Diem\_02\_Dang | 5 | 6 | 7 | “Đây không phải là tam giác cân.” | “Đây không phải là tam giác cân.” | Pass |

[TestMethod]

public void TC02\_51\_Diem\_02\_Dang()

{

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với 3 cạnh khác nhau

// Kết quả ra Pass

var result = checker.Execute\_51\_Diem\_02\_Dang(5, 6, 7);

// Kiểm tra kết quả trả về là "Đây không phải là tam giác cân."

Assert.AreEqual("Đây không phải là tam giác cân.", result);

}

#### TC03\_51\_Diem\_02\_Dang:

thực hiện kiểm tra một tam giác có phải tam giác cân không. Kết quả trả ra là ba cạnh không tạo thành tam giác.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Test Case ID** | **Input** | | | **Kết quả mong đợi** | **Kết quả thực sự** | **Kết quả cuối cùng** |
| a | b | c |
| 3 | TC03\_51\_Diem\_02\_Dang | 1 | 2 | 3 | “Ba cạnh không tạo thành tam giác.” | “Ba cạnh không tạo thành tam giác.” | Pass |

// Test trường hợp không phải tam giác

[TestMethod]

public void TC03\_51\_Diem\_02\_Dang()

{

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với các cạnh không tạo thành tam giác (ví dụ: 1 + 2 <= 3)

// Kết quả ra Pass

var result = checker.Execute\_51\_Diem\_02\_Dang(1, 2, 3);

// Kiểm tra kết quả trả về là "Ba cạnh không tạo thành tam giác."

Assert.AreEqual("Ba cạnh không tạo thành tam giác.", result);

}

#### TC04\_51\_Diem\_02\_Dang:

thực hiện kiểm tra một tam giác có phải tam giác cân không. Kết quả trả ra là các cạnh phải lớn hơn 0.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Test Case ID** | **Input** | | | **Kết quả mong đợi** | **Kết quả thực sự** | **Kết quả cuối cùng** |
| a | b | c |
| 4 | TC04\_51\_Diem\_02\_Dang | 0 | 5 | 6 | “Các cạnh phải lớn hơn 0.” | “Các cạnh phải lớn hơn 0.” | Pass |

// Test trường hợp cạnh không hợp lệ (<= 0)

[TestMethod]

public void TC04\_51\_Diem\_02\_Dang() {

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với các cạnh không hợp lệ

// Kết quả ra Pass

var result = checker.Execute\_51\_Diem\_02\_Dang(0, 5, 6);

// Kiểm tra kết quả trả về là "Các cạnh phải lớn hơn 0."

Assert.AreEqual("Các cạnh phải lớn hơn 0.", result);

}

#### TC05\_51\_Diem\_02\_Dang:

thực hiện kiểm tra một tam giác có phải tam giác cân không. Kết quả trả ra là đây không phải là tam giác cân.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Test Case ID** | **Input** | | | **Kết quả mong đợi** | **Kết quả thực sự** | **Kết quả cuối cùng** |
| a | b | c |
| 5 | TC05\_51\_Diem\_02\_Dang | 5 | 5 | 3 | “Đây không phải là tam giác cân.” | “Đây là tam giác cân.” | Fail |

// Test trường hợp fail của TC01\_51\_Diem\_02\_Dang

[TestMethod]

public void TC05\_51\_Diem\_02\_Dang()

{

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với 2 cạnh bằng nhau

// Kết quả ra Fail

var result = checker.Execute\_51\_Diem\_02\_Dang(5, 5, 3);

// Kiểm tra kết quả trả về là "Đây là tam giác cân."

Assert.AreEqual("Đây không phải là tam giác cân.", result);

}

#### TC06\_51\_Diem\_02\_Dang:

thực hiện kiểm tra một tam giác có phải tam giác cân không. Kết quả trả ra là đây là tam giác cân.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Test Case ID** | **Input** | | | **Kết quả mong đợi** | **Kết quả thực sự** | **Kết quả cuối cùng** |
| a | b | c |
| 6 | TC06\_51\_Diem\_02\_Dang | 5 | 6 | 7 | “Đây là tam giác cân.” | “Đây không phải là tam giác cân.” | Fail |

// Test trường hợp fail của TC02\_51\_Diem\_02\_Dang

[TestMethod]

public void TC06\_51\_Diem\_02\_Dang()

{

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với 3 cạnh khác nhau

// Kết quả ra Fail

var result = checker.Execute\_51\_Diem\_02\_Dang(5, 6, 7);

// Kiểm tra kết quả trả về là "Đây không phải là tam giác cân."

Assert.AreEqual("Đây là tam giác cân.", result);

}

#### TC07\_51\_Diem\_02\_Dang:

thực hiện kiểm tra một tam giác có phải tam giác cân không. Kết quả trả ra là đây là tam giác cân.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Test Case ID** | **Input** | | | **Kết quả mong đợi** | **Kết quả thực sự** | **Kết quả cuối cùng** |
| a | b | c |
| 7 | TC07\_51\_Diem\_02\_Dang | 1 | 2 | 3 | “Đây là tam giác cân.” | “Ba cạnh không tạo thành tam giác.” | Fail |

// Test trường hợp fail của TC03\_51\_Diem\_02\_Dang

[TestMethod]

public void TC07\_51\_Diem\_02\_Dang()

{

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với các cạnh không tạo thành tam giác (ví dụ: 1 + 2 <= 3)

// Kết quả ra Fail

var result = checker.Execute\_51\_Diem\_02\_Dang(1, 2, 3);

// Kiểm tra kết quả trả về là "Ba cạnh không tạo thành tam giác."

Assert.AreEqual("Đây là tam giác cân.", result);

}

#### TC08\_51\_Diem\_02\_Dang:

thực hiện kiểm tra một tam giác có phải tam giác cân không. Kết quả trả ra là đây không phải là tam giác cân.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Test Case ID** | **Input** | | | **Kết quả mong đợi** | **Kết quả thực sự** | **Kết quả cuối cùng** |
| a | b | c |
| 8 | TC08\_51\_Diem\_02\_Dang | 0 | 5 | 6 | “Đây không phải là tam giác cân.” | “Các cạnh phải lớn hơn 0.” | Fail |

// Test trường hợp fail của TC04\_51\_Diem\_02\_Dang

[TestMethod]

public void TC08\_51\_Diem\_02\_Dang()

{

// Tạo đối tượng TriangleChecker\_51\_Diem\_02\_Dang

var checker = new TriangleChecker\_51\_Diem\_02\_Dang();

// Gọi phương thức Execute\_51\_Diem\_02\_Dang với các cạnh không hợp lệ

// Kết quả ra Fail

var result = checker.Execute\_51\_Diem\_02\_Dang(0, 5, 6);

// Kiểm tra kết quả trả về là "Các cạnh phải lớn hơn 0."

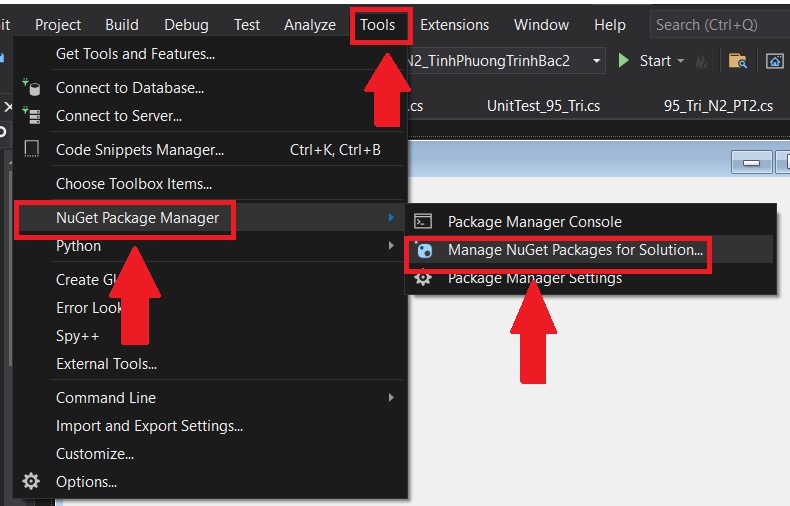
Assert.AreEqual("Đây không phải là tam giác cân.", result);

}

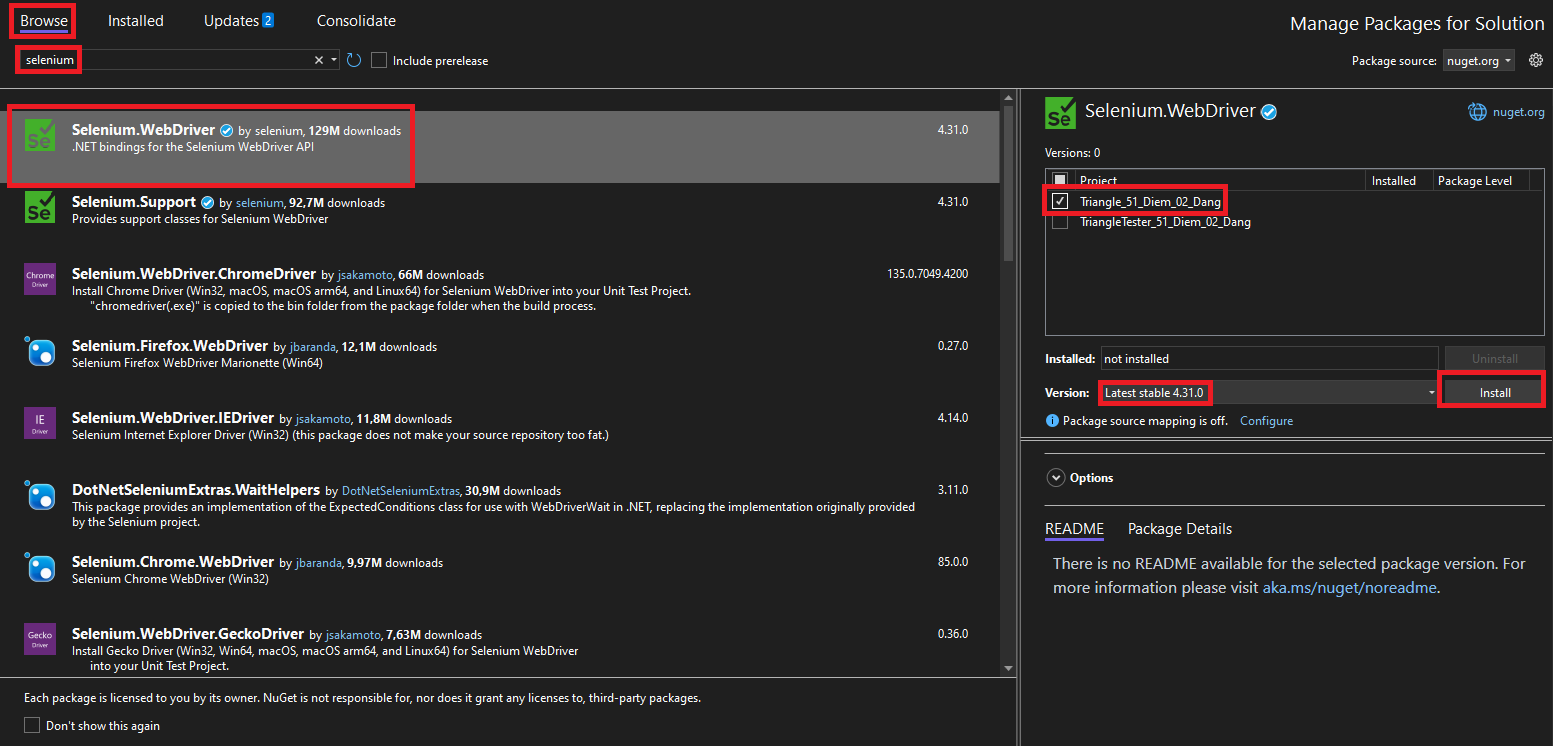
# CHƯƠNG 2: WEBDRIVER SELENIUM

## Cài đặt Selenium

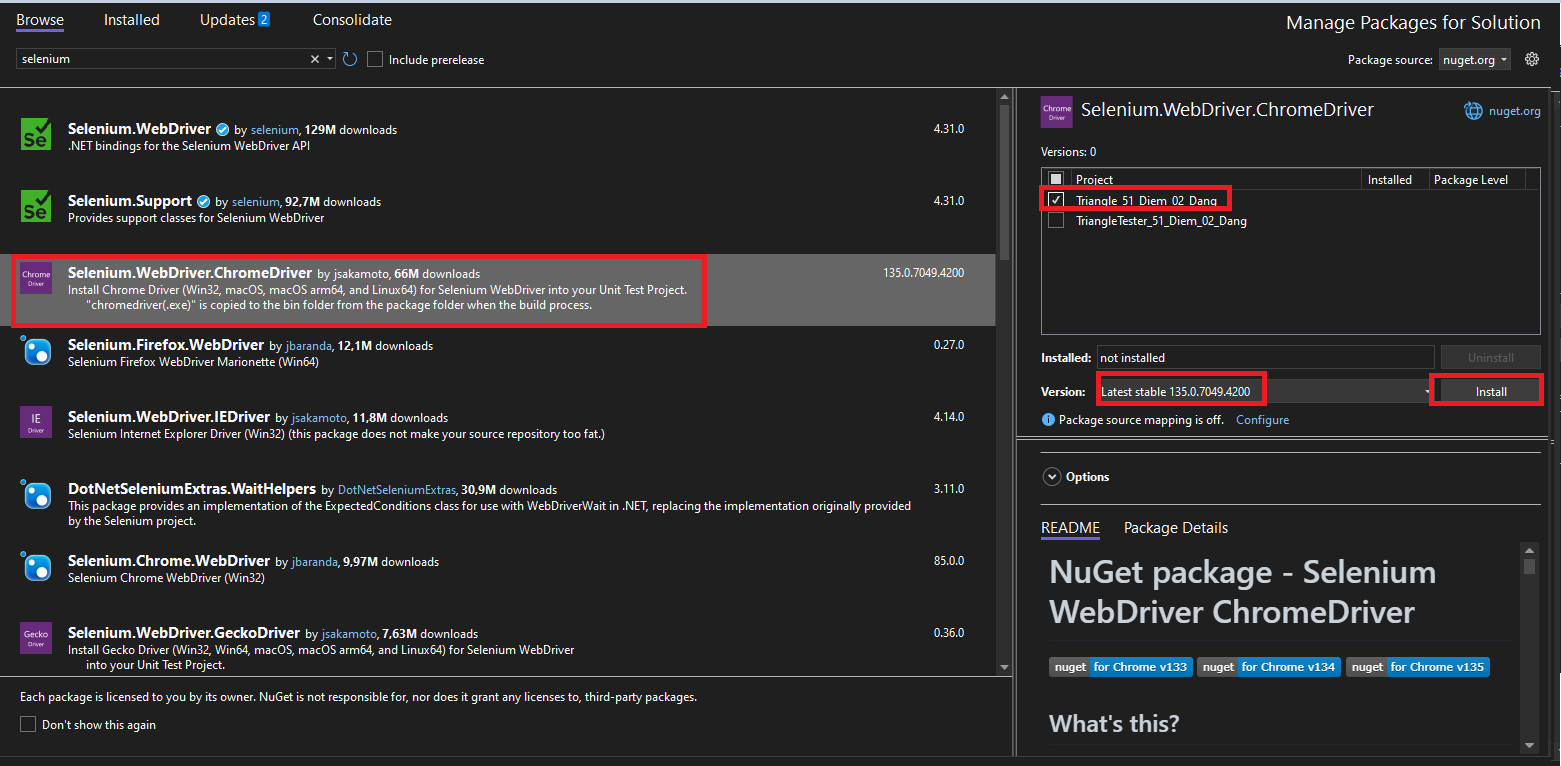
Bước 1: Vào Tool -> Chọn Nuget Package Manager -> Chọn Nuget Pack Manager for Solution



Bước 2: Chọn Browse -> Tìm kiếm từ khóa Selenium -> Cài đặt thư viện Selenium.WebDrvier. Sau đó chọn project cần cài đặt, chọn phiên bản rồi nhấn Install.



Bước 3: Cài đặt Selenium.WebDriver.ChromeDriver (trước khi cài đặt kiểm tra phiên bản của chrome để cài đặt cùng phiên bản). Sau đó chọn project cần cài đặt rồi nhấn Install.

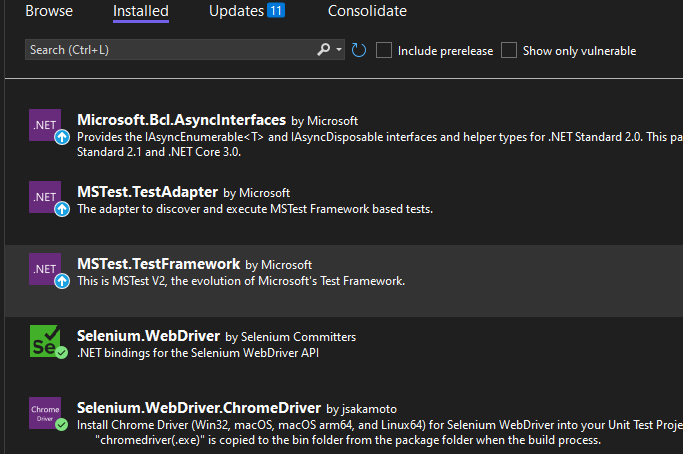


- Phiên bản của Selenium.WebDriver: 4.31.0

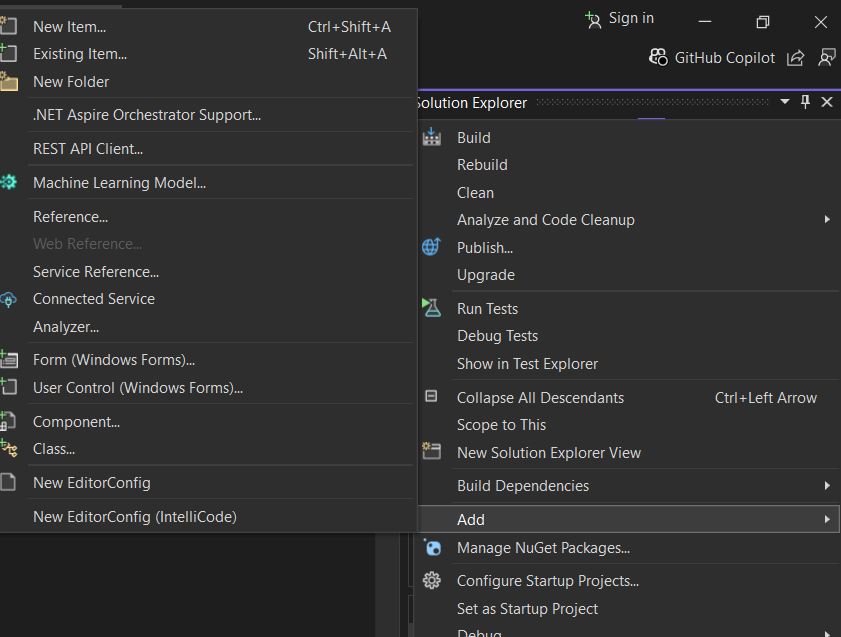
- Phiên bản của Chrome: 135.0.7049.4200

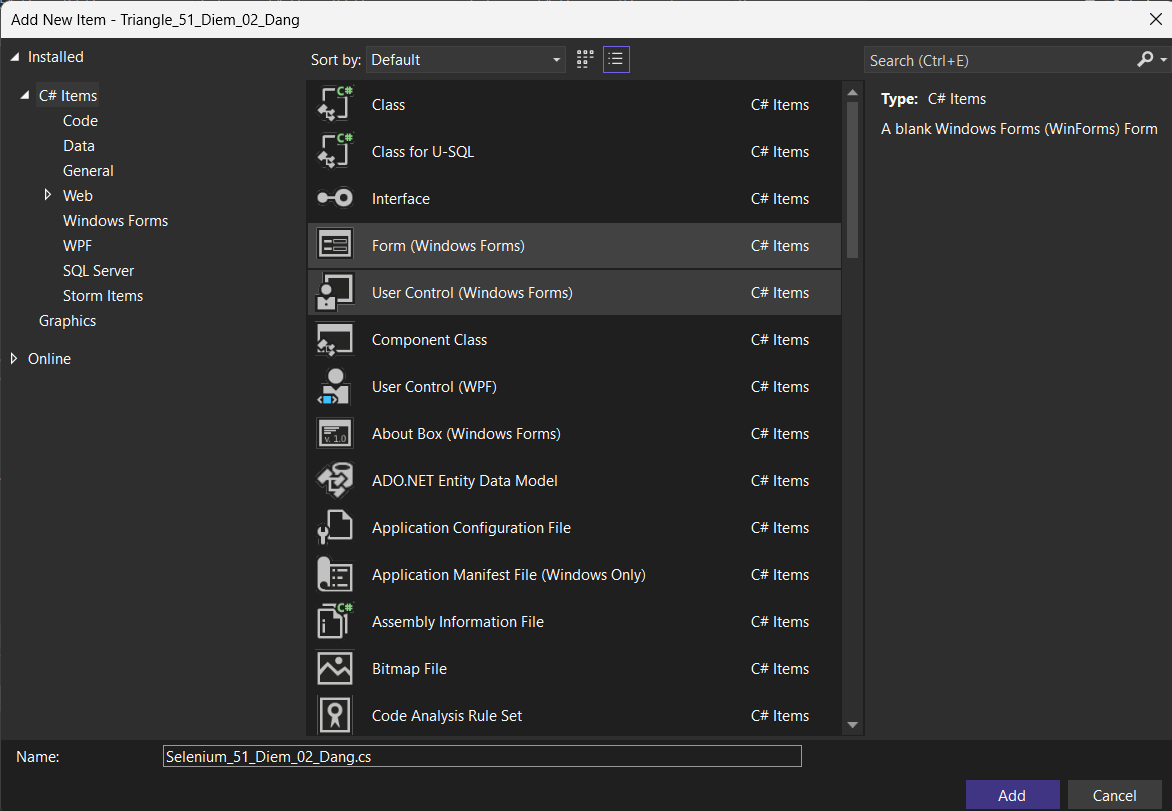
- Phiên bản của Selenium.WebDriver: 135.0.7049.4200

- Cài đặt thành công:



## Tạo form Selenium\_51\_Diem\_02\_Dang





## Chức năng đăng nhập

Tên hàm: login\_02\_51\_Click

Bước 1: Truy cập vào trang web <https://nhaccuatuicom/> và thực hiện kiếm thử

//Sử dụng thư viện

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

//Đóng màn hình đen khi chạy

ChromeDriverService chrome\_02\_51 = ChromeDriverService.CreateDefaultService();

chrome\_02\_51.HideCommandPromptWindow = true;

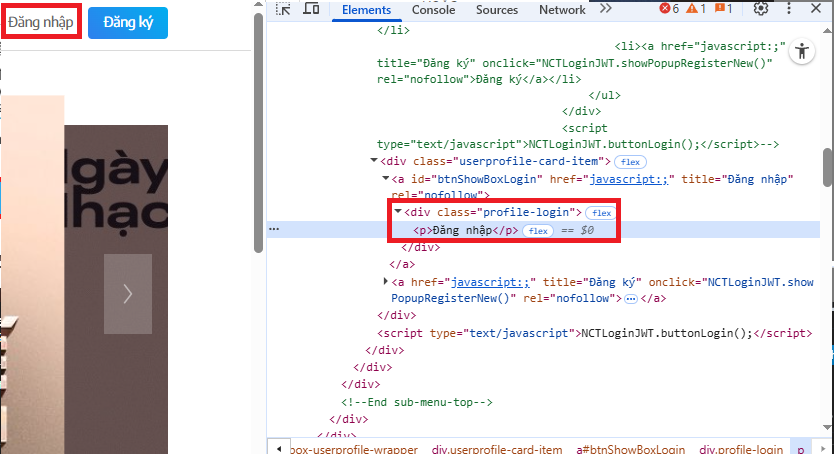
//Điều hướng trình duyệt đến link

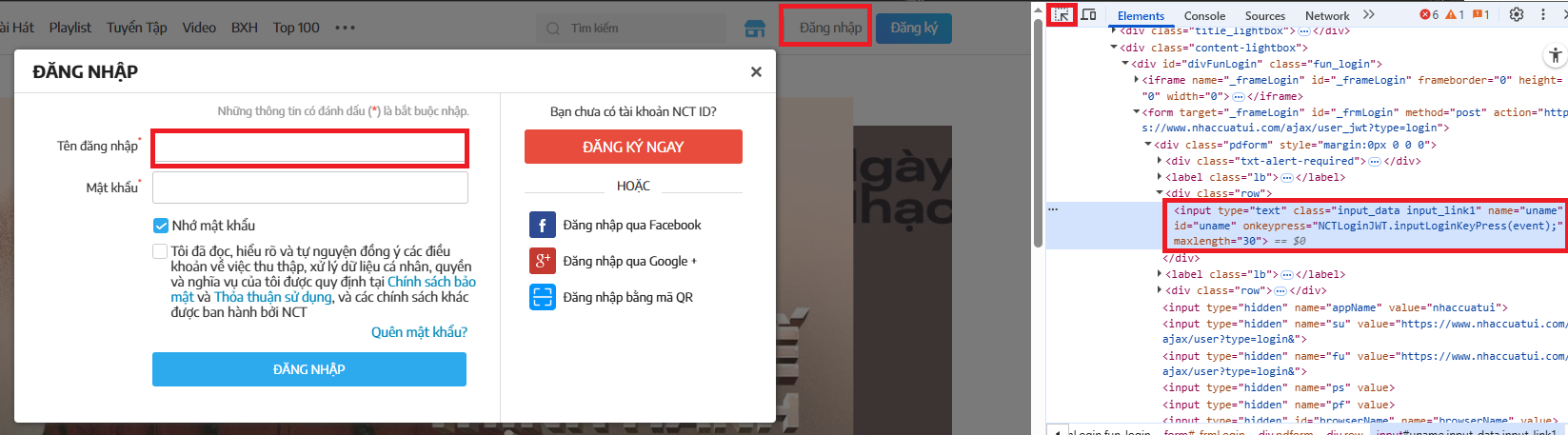
IWebDriver driver\_02\_51 = new ChromeDriver(chrome\_02\_51);

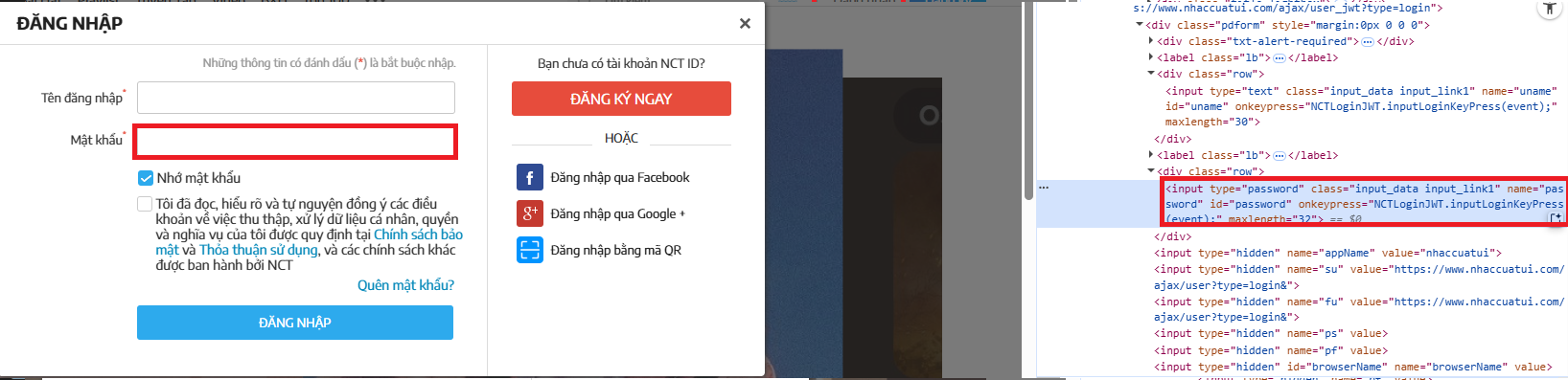
//Chuyển Link cho Selenium điều hướng

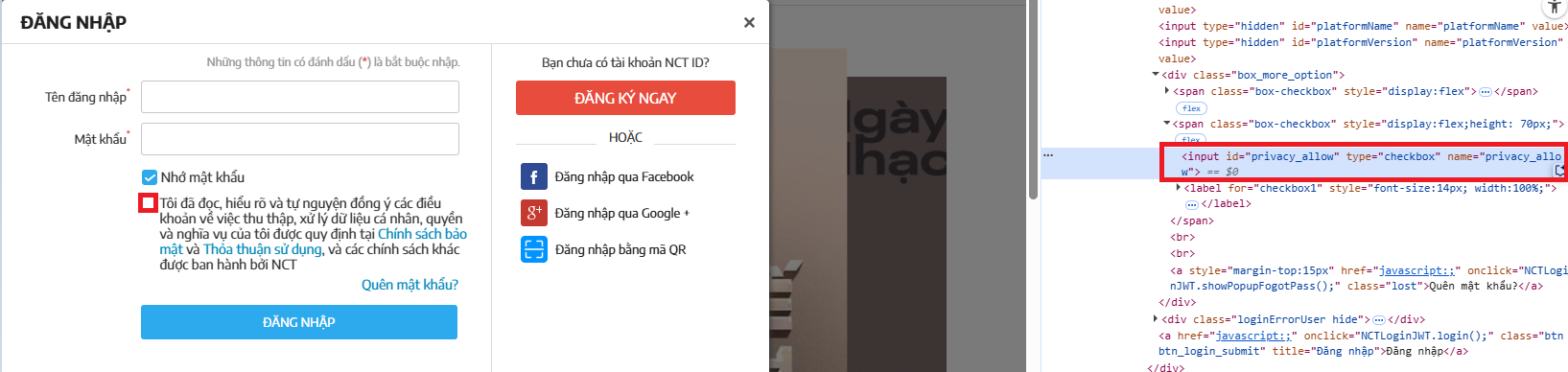
driver\_02\_51.Navigate().GoToUrl("https://www.nhaccuatui.com/");

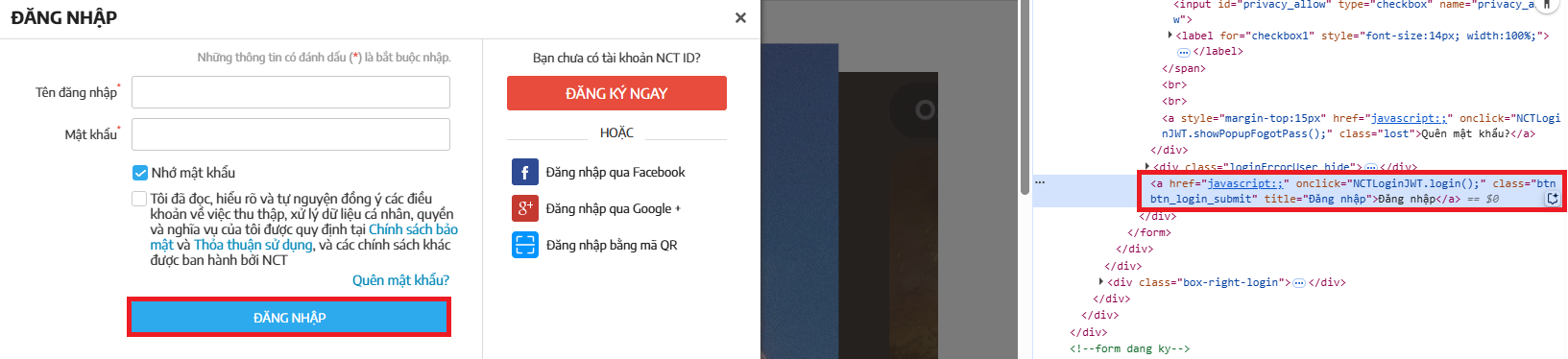
Bước 2: Lấy Element của các ô Login











Bước 3: Lấy Username và Pass từ các Textbox để kiểm thử

//Click nút đăng nhập

driver\_02\_51.FindElement(By.ClassName("profile-login")).Click();

//Nhập username

driver\_02\_51.FindElement(By.Name("uname")).SendKeys(uname\_02\_51.Text);

//Nhập password

driver\_02\_51.FindElement(By.Name("password")).SendKeys(pw\_02\_51.Text);

//Chấp nhận điều khoản

driver\_02\_51.FindElement(By.Id("privacy\_allow")).Click();

//Nhấp đăng nhập

driver\_02\_51.FindElement(By.XPath("//\*[@id=\"\_frmLogin\"]/div/a")).Click();

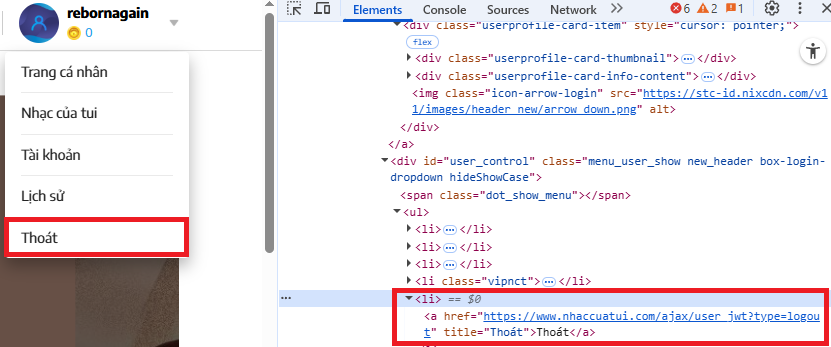
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Test Case ID** | **Input** | | **Kết quả mong đợi** | **Kết quả thực sự** | **Kết quả cuối cùng** |
| Username | Password |
| 9 | TC09\_51\_Diem\_02\_Dang | Rebornagain | Testing123! | Đăng nhập thành công. | Đăng nhập thành công | Pass |

## Chức năng đăng xuất

Tên hàm: logout\_02\_51\_Click

Bước 1: Giống như chức năng đăng nhập

Bước 2: Lấy element của nút đăng xuất



//Hover qua dropdown menu

Actions action\_02\_51 = new Actions(driver\_02\_51);

IWebElement ele\_02\_51 = driver\_02\_51.FindElement(By.ClassName("userprofile-card-item"));

action\_02\_51.MoveToElement(ele\_02\_51).Perform();

//Đăng xuất

driver\_02\_51.FindElement(By.XPath("//\*[@id=\"user\_control\"]/ul/li[5]/a")).Click();

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Test Case ID** | **Input** | | **Kết quả mong đợi** | **Kết quả thực sự** | **Kết quả cuối cùng** |
| Username | Password |
| 10 | TC10\_51\_Diem\_02\_Dang | Rebornagain | Testing123! | Đăng xuất thành công. | Đăng xuất thành công | Pass |

## Chức năng tìm bài đầu tiên theo Textbox

Tên hàm: search\_02\_51\_Click

Bước 1: Truy cập vào trang web <https://nhaccuatuicom/> và thực hiện kiếm thử

//Đóng màn hình đen khi chạy

ChromeDriverService chrome\_02\_51 = ChromeDriverService.CreateDefaultService();

chrome\_02\_51.HideCommandPromptWindow = true;

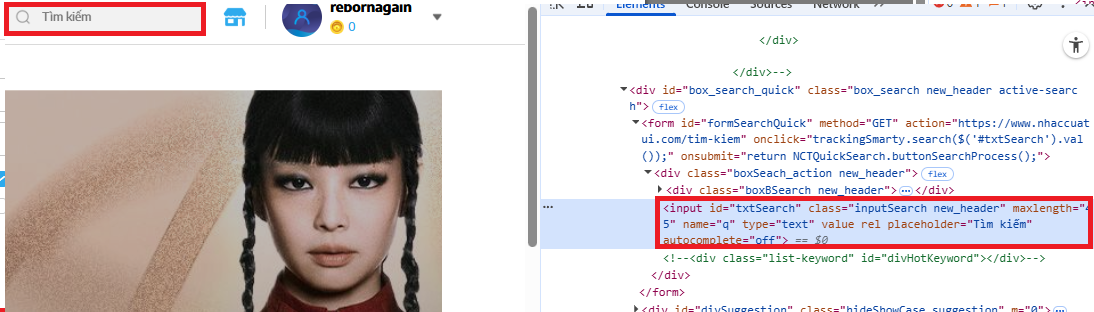
//Điều hướng trình duyệt đến link

IWebDriver driver\_02\_51 = new ChromeDriver(chrome\_02\_51);

//Chuyển Link cho Selenium điều hướng

driver\_02\_51.Navigate().GoToUrl("https://www.nhaccuatui.com/");

Bước 2: Lấy các Element của hộp tìm kiếm



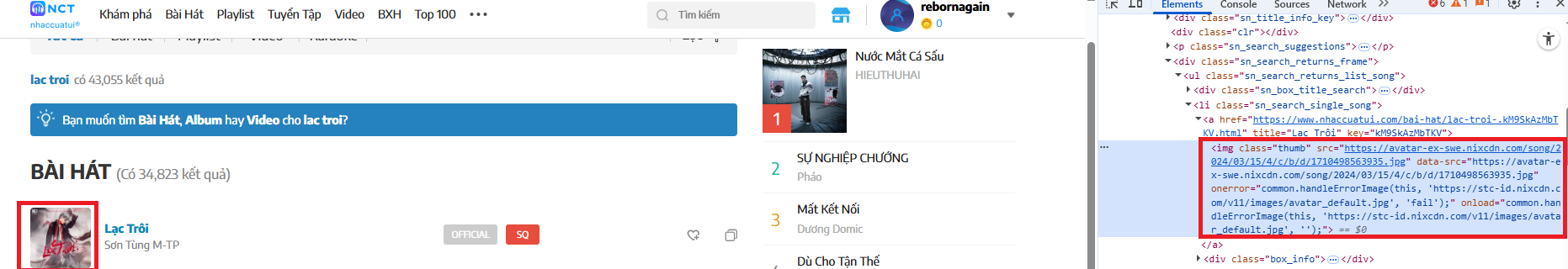
//Search nhạc

IWebElement search = driver\_02.FindElement(By.Id("txtSearch"));

search.SendKeys(search\_txtb\_02\_51.Text);

search.Submit();

Bước 3: Cho các element của hình ảnh vì các yếu tố khác không trùng



Bước 4: Thiết kế code và kiểm thử

//Search nhạc

IWebElement search = driver\_02\_51.FindElement(By.Id("txtSearch"));

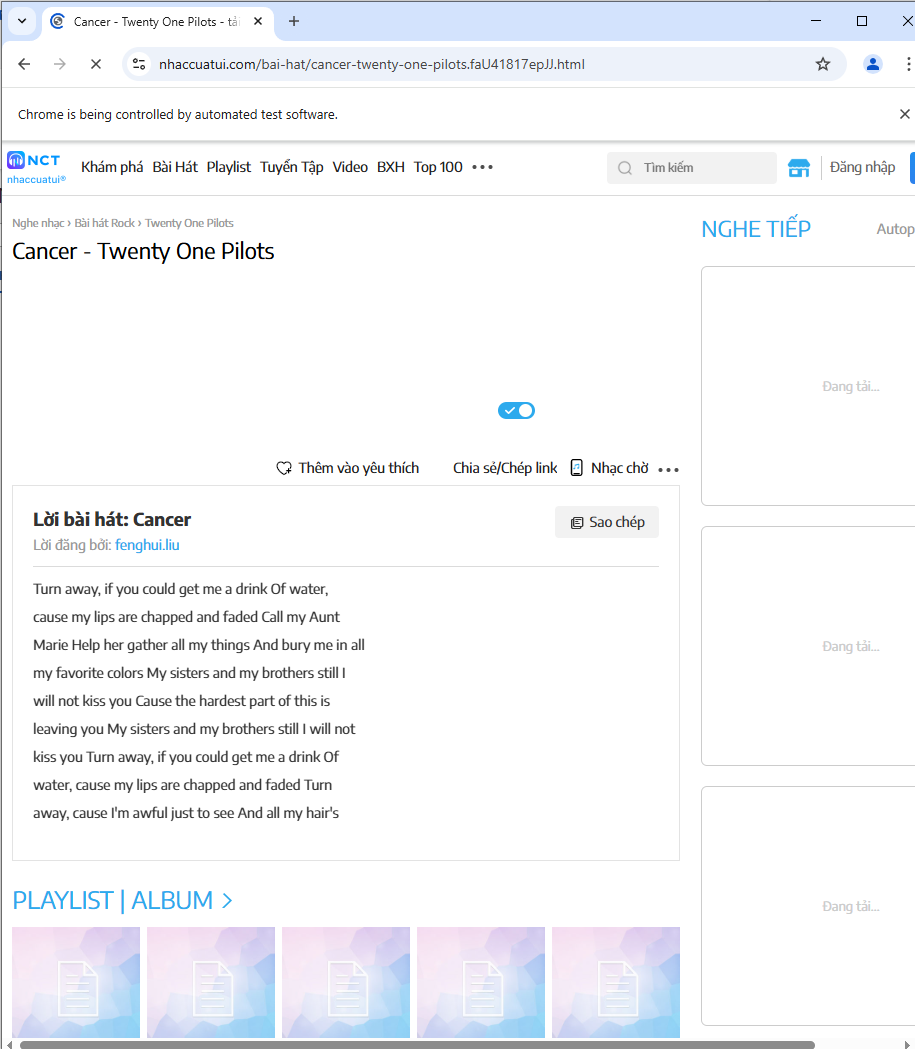
search.SendKeys(search\_txtb\_02\_51.Text);

search.Submit();

//Select First

IReadOnlyList<IWebElement> first\_02\_51 = driver\_02\_51.FindElements(By.ClassName("thumb"));

first\_02\_51.First().Click();



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Test Case ID** | **Input** | **Kết quả mong đợi** | **Kết quả thực sự** | **Kết quả cuối cùng** |
| Search |
| 11 | TC11\_51\_Diem\_02\_Dang | Cancer twentyone pilots | Tìm bài thành công | Tìm bài thành công | Pass |

## Chức năng chọn bài top 100

Tên hàm: top100\_02\_51\_Click

Bước 1: Truy cập vào trang web <https://nhaccuatuicom/> và thực hiện kiếm thử

//Đóng màn hình đen khi chạy

ChromeDriverService chrome\_02\_51 = ChromeDriverService.CreateDefaultService();

chrome\_02\_51.HideCommandPromptWindow = true;

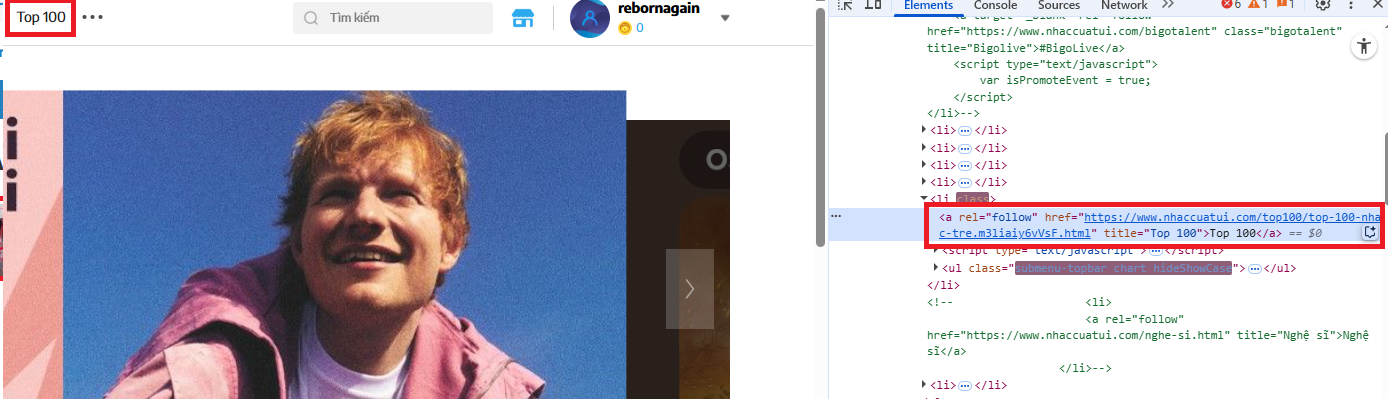
//Điều hướng trình duyệt đến link

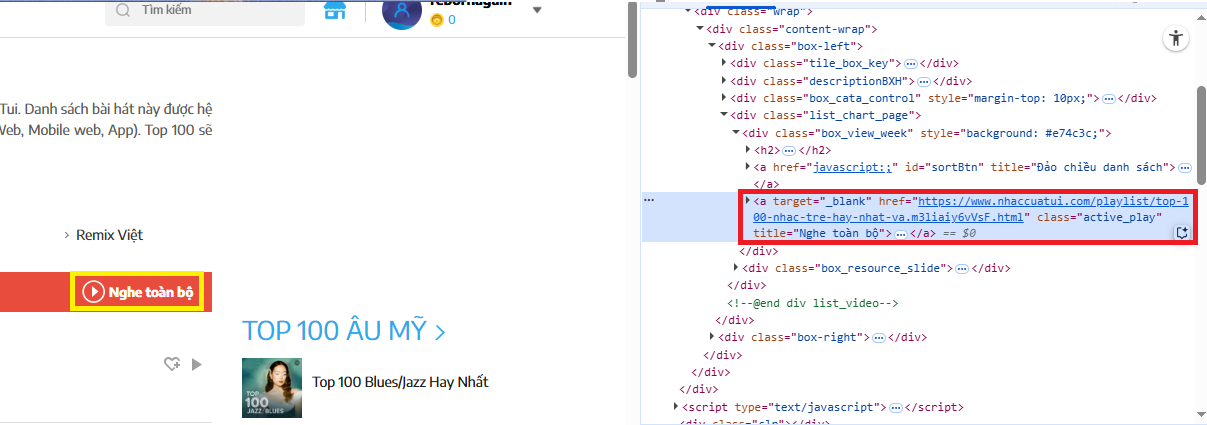
IWebDriver driver\_02\_51 = new ChromeDriver(chrome\_02\_51);

//Chuyển Link cho Selenium điều hướng

driver\_02\_51.Navigate().GoToUrl("https://www.nhaccuatui.com/");

Bước 2: Lấy các Element của hướng đi





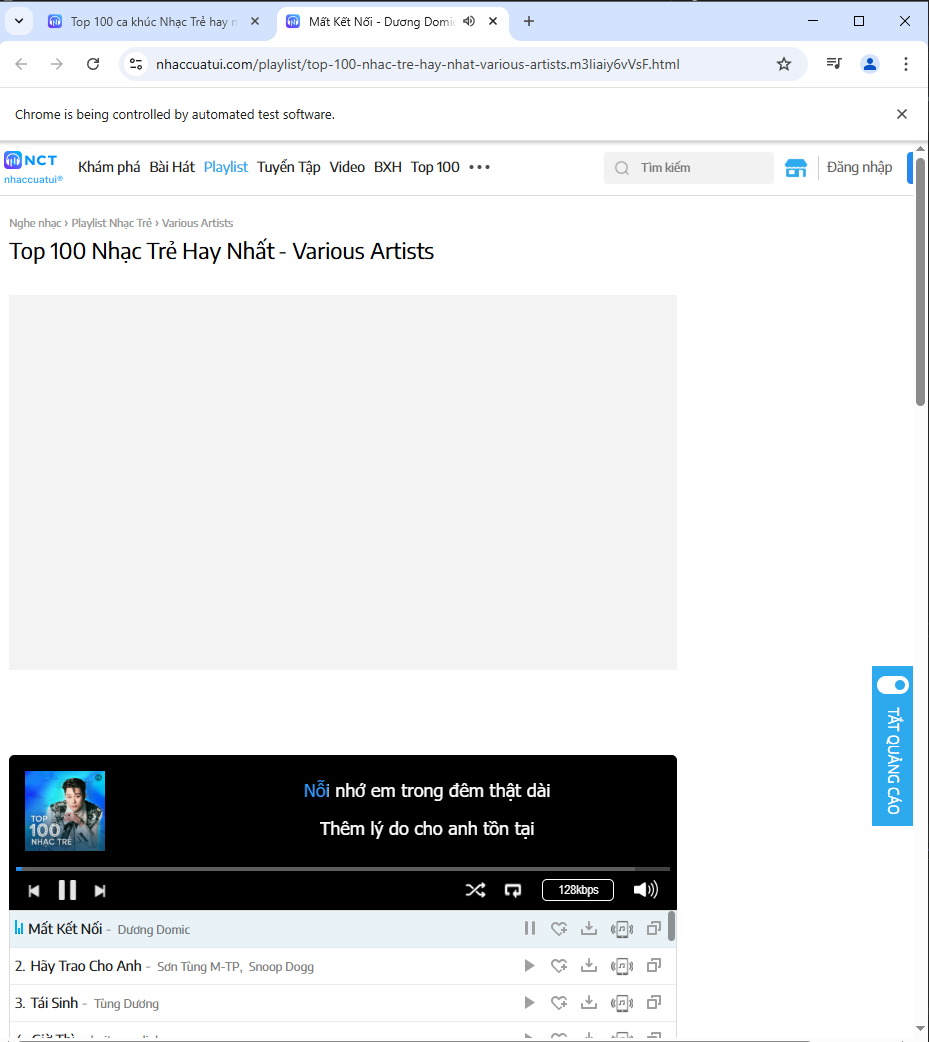
Bước 3: Thiết kế Code và kiểm thử

//Vào mục top 100

driver\_02\_51.FindElement(By.LinkText("Top 100")).Click();

//Nghe toàn bộ

driver\_02\_51.FindElement(By.ClassName("active\_play")).Click();



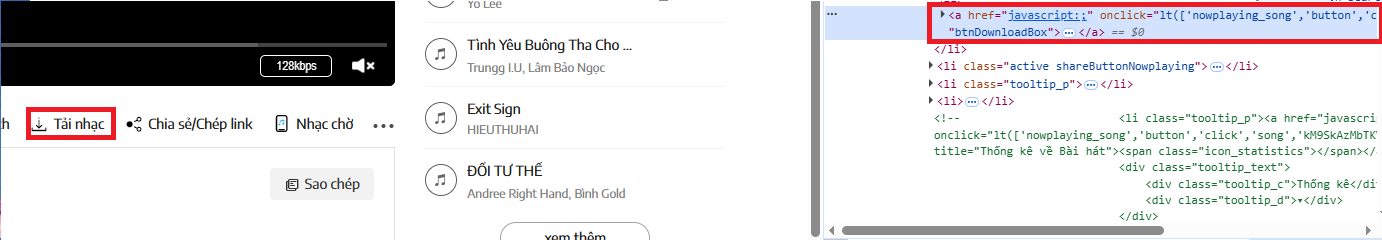
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Test Case ID** | **Kết quả mong đợi** | **Kết quả thực sự** | **Kết quả cuối cùng** |
| 12 | TC12\_51\_Diem\_02\_Dang | Tìm bài thành công | Tìm bài thành công | Pass |

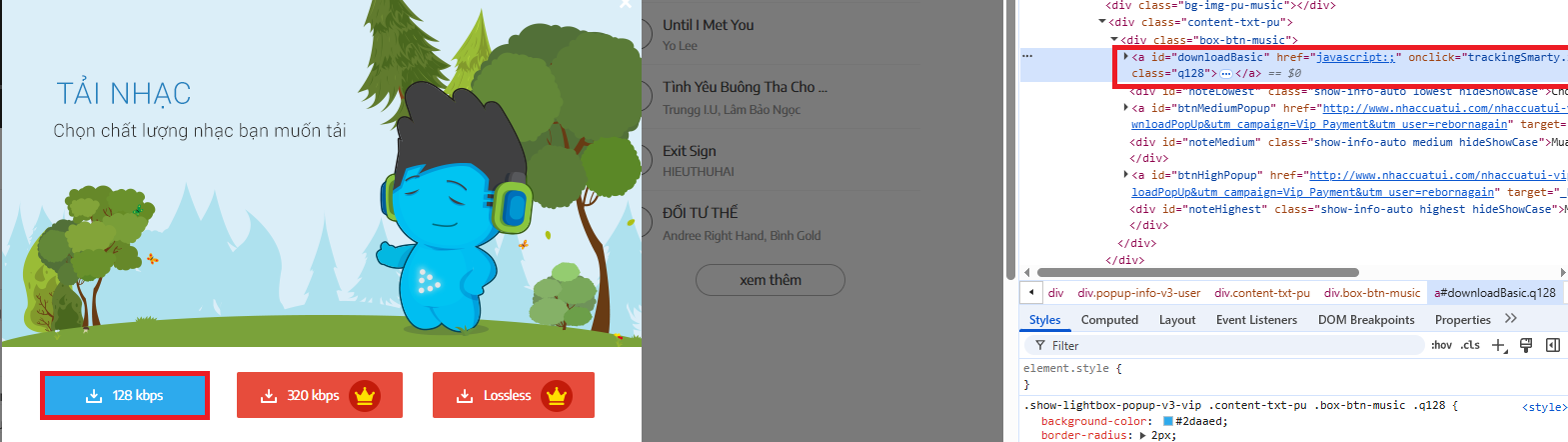
## Chức năng tải nhạc

Tên hàm: download\_02\_51\_Click

Bước 1: Như bước tìm bài đầu tiên theo Textbox

Bước 2: Lấy Element của tải bài hát





Bước 3: Thiết kế code và kiểm thử

//Tải nhạc

driver\_02\_51.FindElement(By.Id("btnDownloadBox")).Click();

Thread.Sleep(2000);

driver\_02\_51.FindElement(By.Id("downloadBasic")).Click();

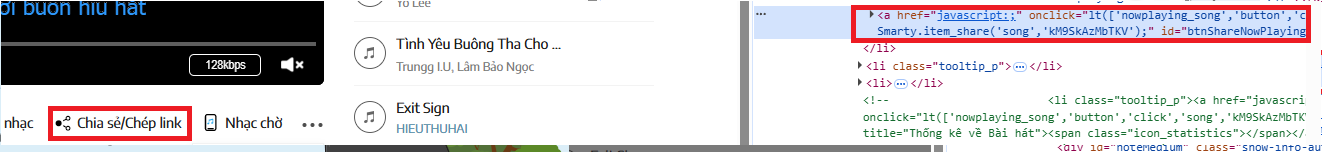
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Test Case ID** | **Input** | **Kết quả mong đợi** | **Kết quả thực sự** | **Kết quả cuối cùng** |
| Search |
| 13 | TC13\_51\_Diem\_02\_Dang | Cancer twentyone pilots | Tải nhạc thành công | Tải nhạc thành công | Pass |

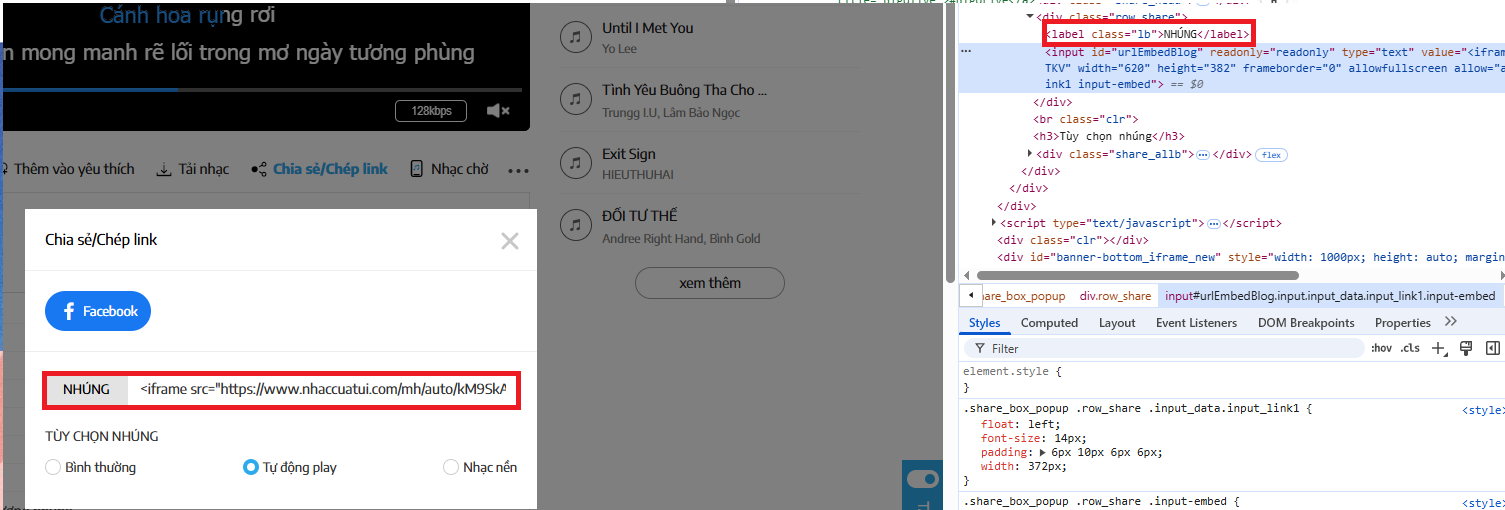
## Chức năng chia sẻ

Tên hàm: sharing\_02\_51\_Click

Bước 1: Như bước tìm bài đầu tiên theo Textbox

Bước 2: Lấy các element để chia sẻ



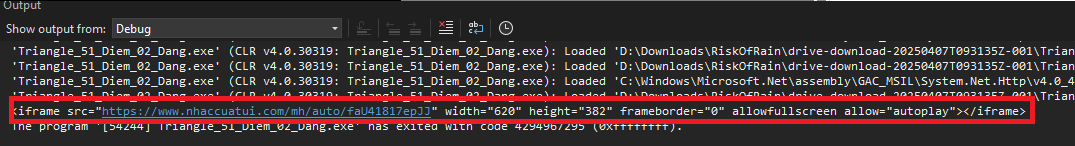


Bước 3: Thiết kế Code để lấy link chia sẻ và kiểm thử

//Click Share

driver\_02\_51.FindElement(By.Id("btnShareNowPlaying")).Click();

Console.WriteLine(driver\_02\_51.FindElement(By.Id("urlEmbedBlog")).GetAttribute("value"));

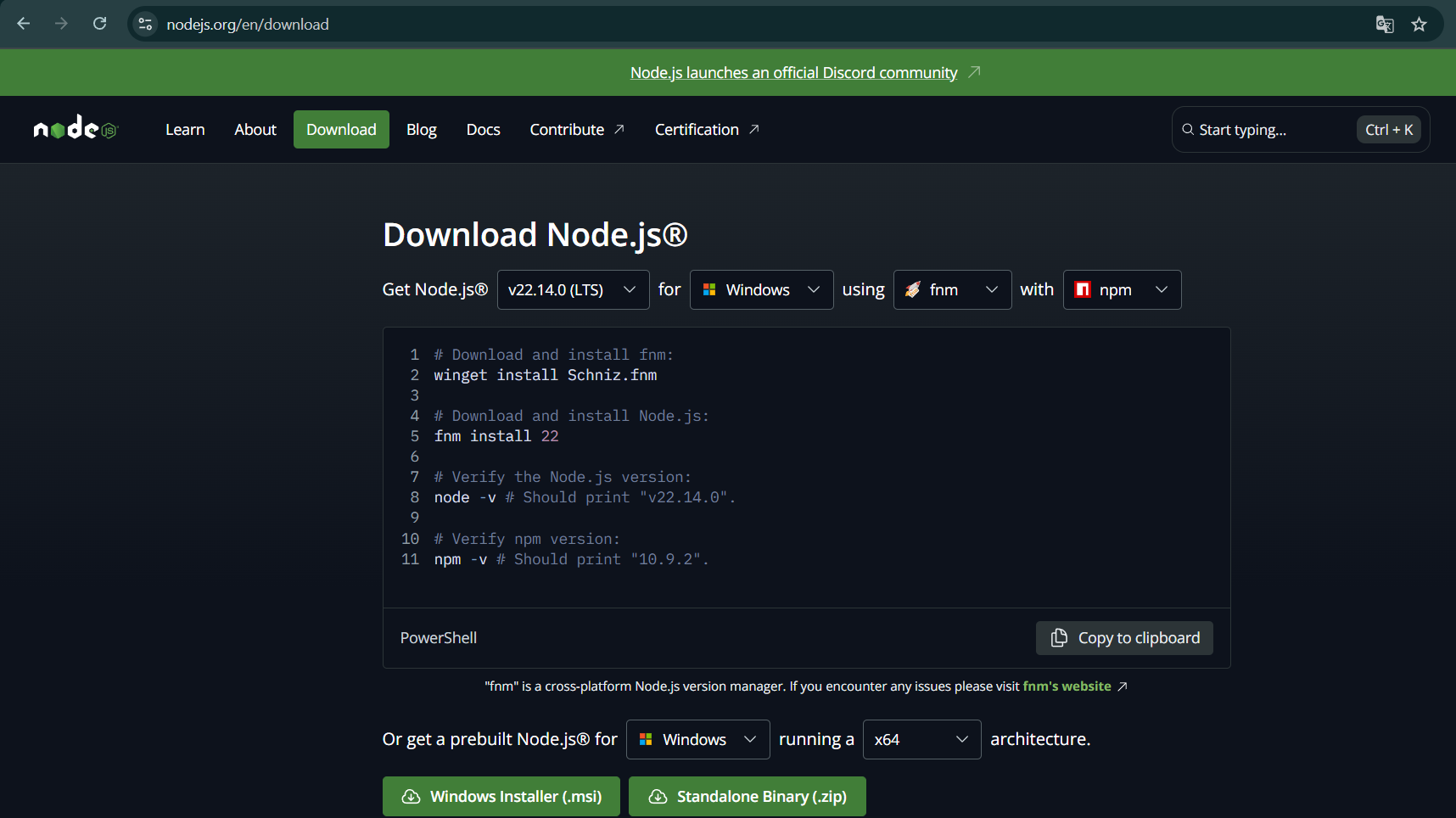


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Test Case ID** | **Input** | **Kết quả mong đợi** | **Kết quả thực sự** | **Kết quả cuối cùng** |
| Search |
| 14 | TC14\_51\_Diem\_02\_Dang | Cancer twentyone pilots | Xuất phần nhúng chia sẻ | Xuất phần nhúng chia sẻ | Pass |

# CHƯƠNG 3: API – POSTMAN

## Cài đặt JSON SERVER:

### Vào link: <https://nodejs.org/en/download>

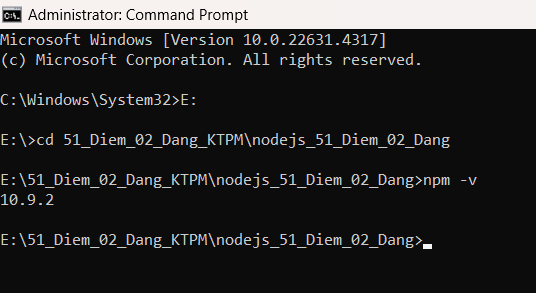


* Nhấn Windows Installer (.msi) để download
* Cài đặt: Nhấn Next…-> Finish

### Mở cmd: Cài đặt gói sử dụng NODE package manager (NPM)

* Vào thưc mục đã cài đặt: check version

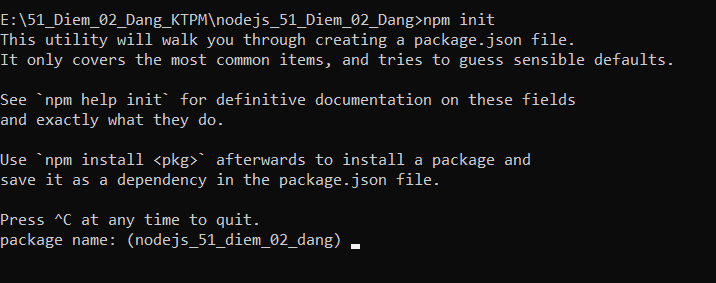
Sử dụng lệnh: npm –v



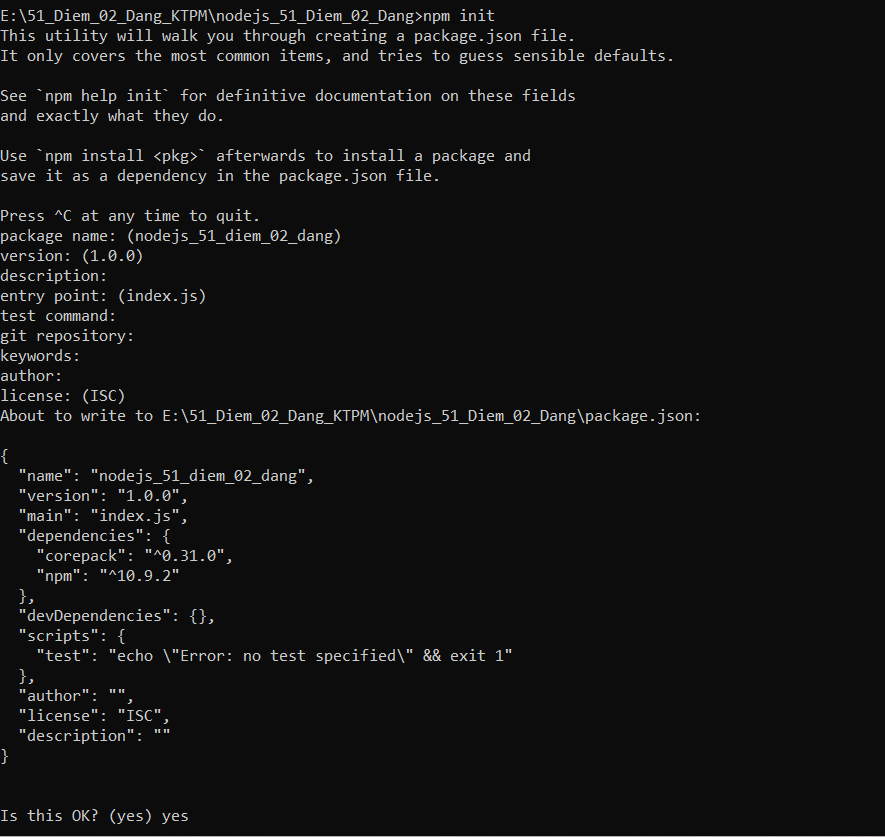
* Cài lên 1 folder: tạo folder E:\51\_Diem\_02\_Dang\_KTPM\nodejs\_51\_Diem\_02\_Dang



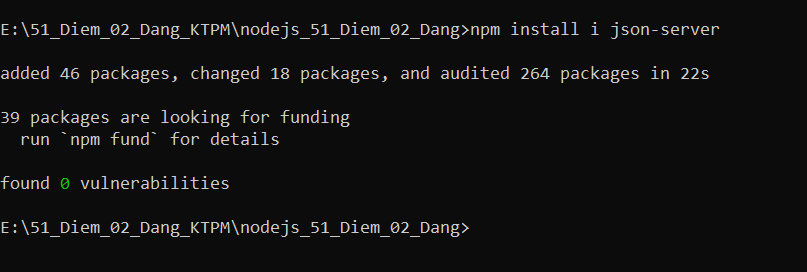
Sử dụng lệnh: npm init



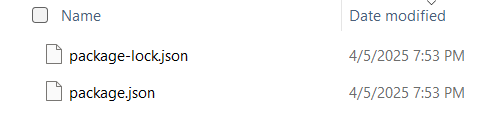
* Enter đến cuối -> Chọn yes



* Sau đó gõ : npm install i json-server



### Kết quả:



## Tạo file JSON thành API trên server ảo:

### Tạo file: Trong Visual Studio Code

51\_Diem\_02\_Dang.json trên visual stuido code (Trong thư mục E:\51\_Diem\_02\_Dang\_KTPM\nodejs\_51\_Diem\_02\_Dang)

Có dữ liệu là:

{

"51\_Diem\_02\_Dang": [

{

"id": "001",

"name": "Samsung A15 5G",

"price": 6000000,

"description": "RAM: 8GB, CPU: MediaTek Dimensity 6100+, ROM: 256GB, OS: Android 14"

},

{

"id": "002",

"name": "Xiaomi 11T 5G",

"price": 12000000,

"description": "RAM: 8GB, CPU: MediaTek Dimensity 1200, ROM: 256GB, OS: Android 11"

},

{

"id": "003",

"name": "Iphone 15 Plus",

"price": 20000000,

"description": "RAM: 6GB, CPU: Apple A16 Bionic, ROM: 128GB, OS: IOS 17"

},

{

"id": "004",

"name": "Oppo Reno12 5G",

"price": 13000000,

"description": "RAM: 12GB, CPU: MediaTek Dimensity 7300-Energy 5G 8 nhân, ROM: 512GB, OS: Android 14"

},

{

"id": "005",

"name": "Realme GT Neo 5",

"price": 10500000,

"description": "RAM: 16GB, CPU: Snapdragon 8+ Gen 1, ROM: 256GB, OS: Android 13"

},

{

"id": "006",

"name": "Vivo V30 Pro",

"price": 15000000,

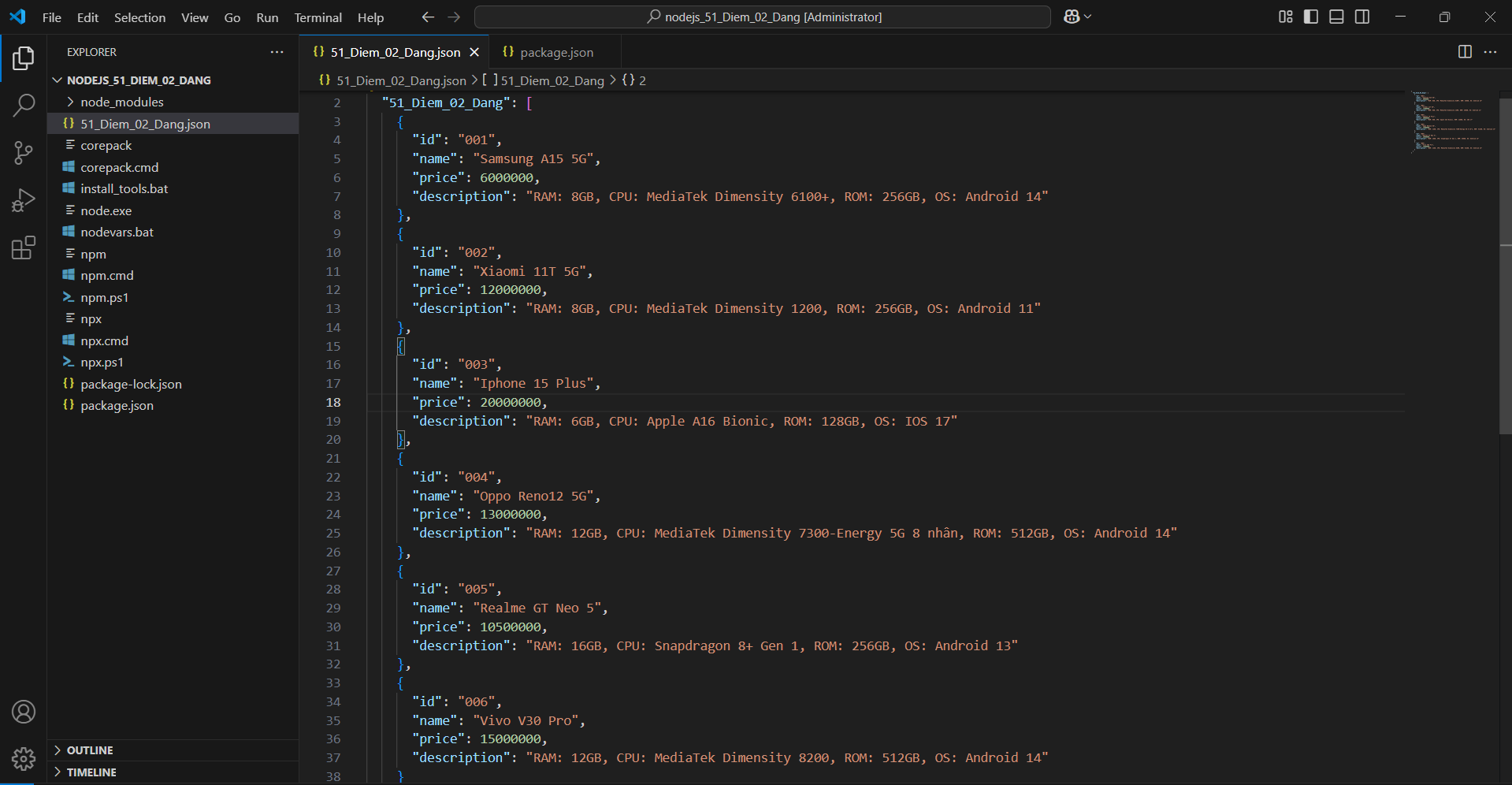
"description": "RAM: 12GB, CPU: MediaTek Dimensity 8200, ROM: 512GB, OS: Android 14"

}

]

}

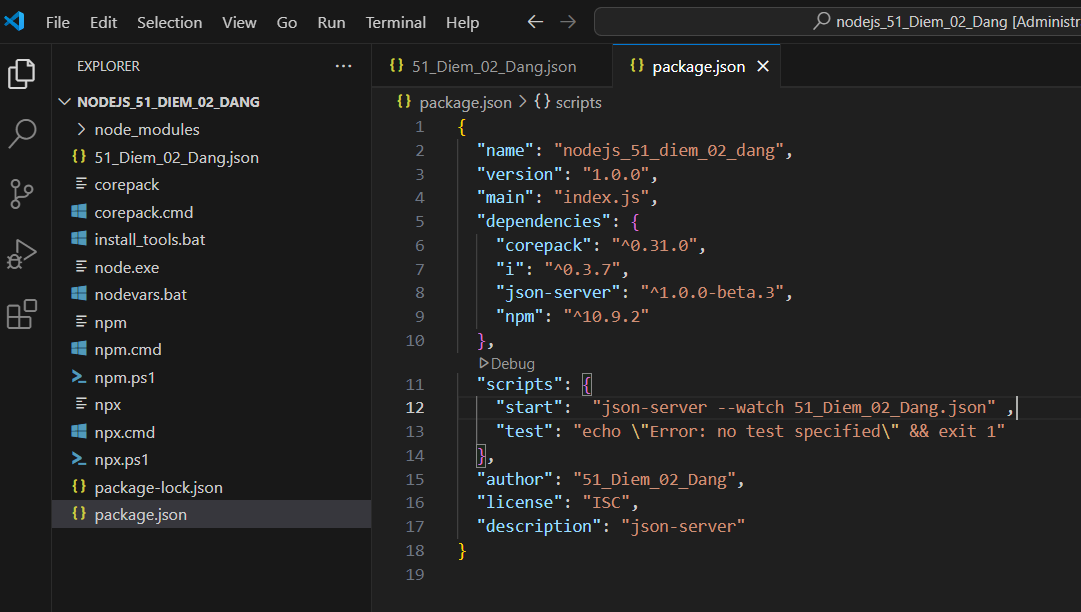
Kết quả:



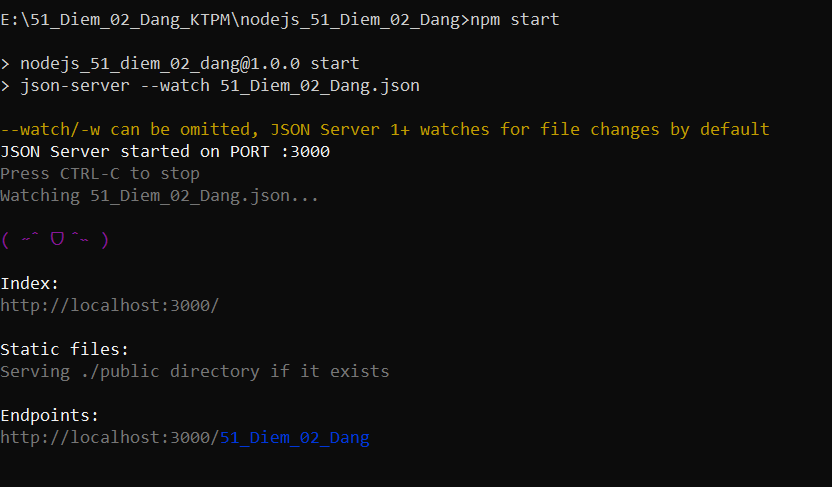
### Mở file package.json, thêm dòng start:

Thêm câu lệnh sau: "start": "json-server --watch 51\_Diem\_02\_Dang.json"

Kết quả:

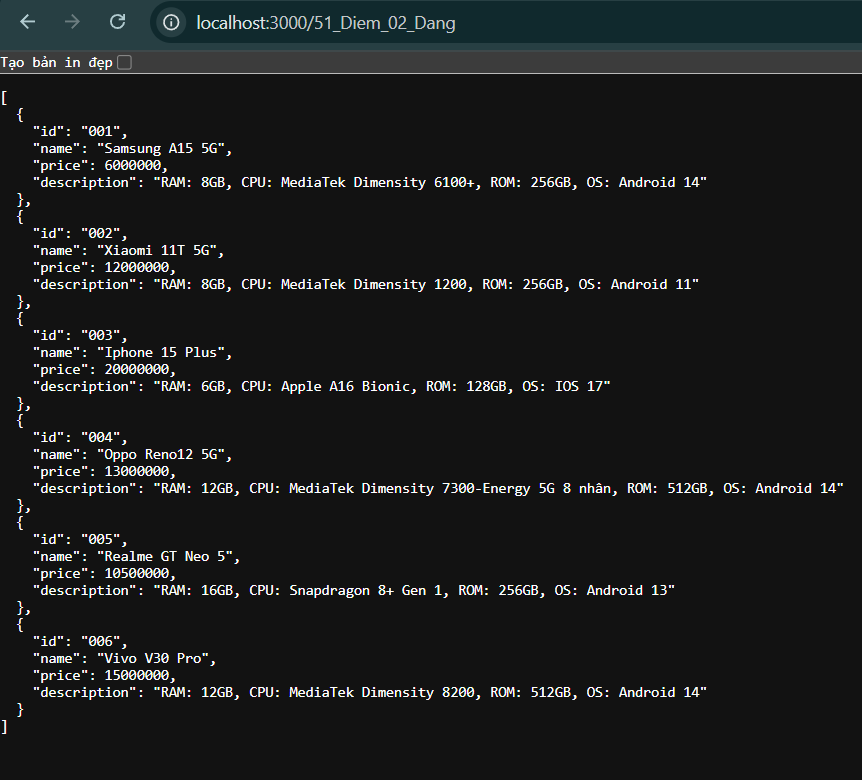


### Mở cmd -> Chạy npm -> Tạo API server



API của nó: <http://localhost:3000/51_Diem_02_Dang>

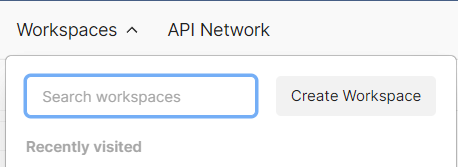
### Mở Chrome -> Mở link <http://localhost:3000/51_Diem_02_Dang>

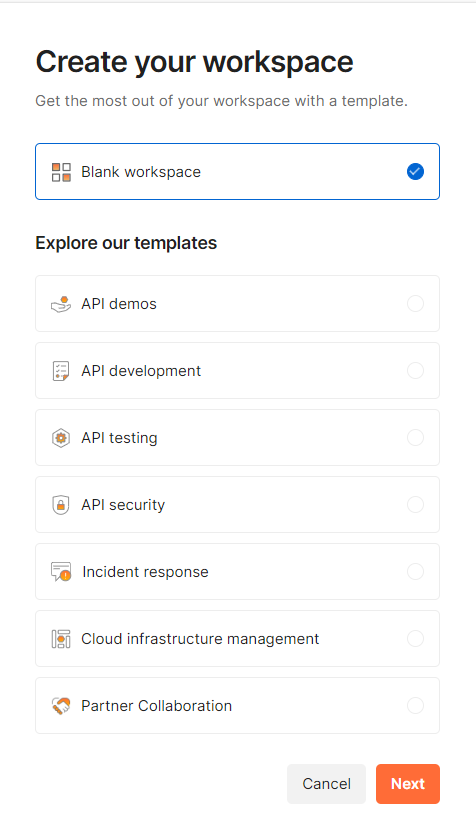


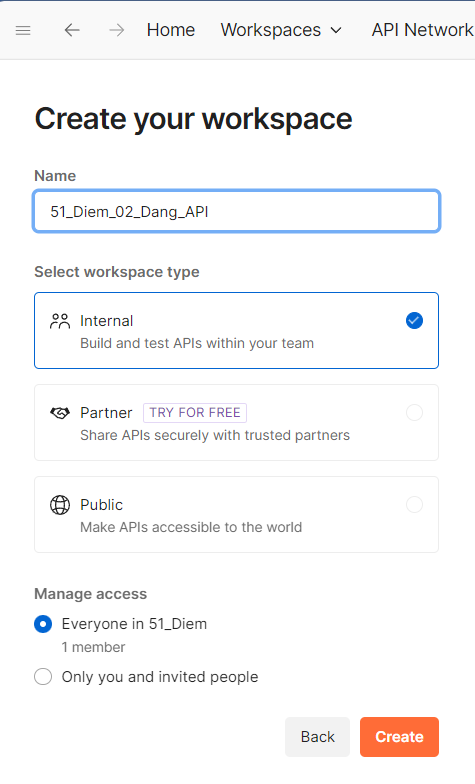
## Truy cập vào Postman để test API vừa tạo:

### Tạo request GET:

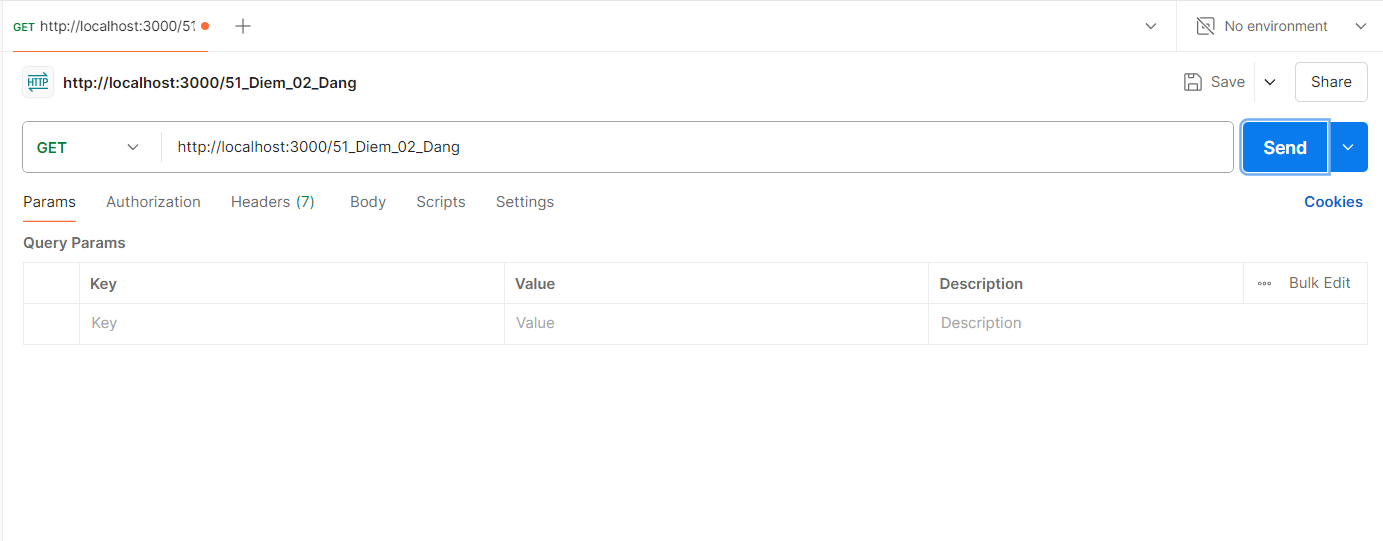
#### Tạo Workspace:



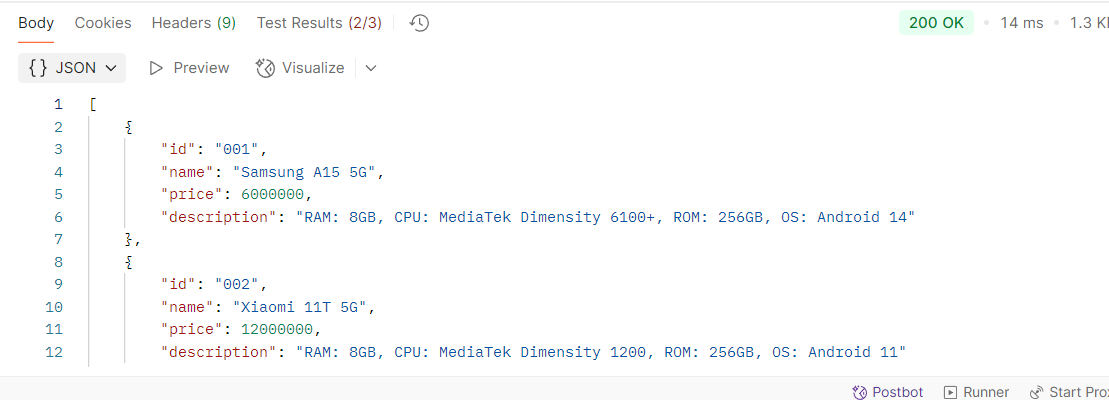




#### Chọn GET -> Nhập link <http://localhost:3000/51_Diem_02_Dang> -> Nhấn Send



#### Thông tin của API:



#### Đặt tên -> Nhấn Save -> Lưu vào Project



#### Ta có thể lọc theo Name:



#### Viết các test case trong test script:

* Code:

//Kiểm tra response body chứa chuỗi "Iphone 15 Plus"

pm.test ("TC1\_51\_Diem\_02\_Dang", function () {

pm.expect (pm.response.text()).to.include("Iphone 15 Plus");

})

//Kiểm tra response body chứa chuỗi "Samsung S24 Ultra"

pm.test ("TC2\_51\_Diem\_02\_Dang", function () {

pm.expect (pm.response.text()).to.include("Samsung S24 Ultra");

})

//Kiểm tra thời gian phản hồi dưới 200 ms

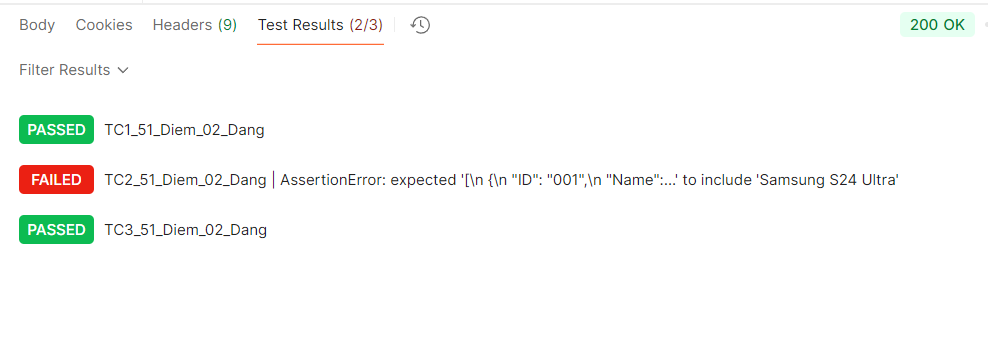
pm.test ("TC3\_51\_Diem\_02\_Dang", function () {

pm.expect(pm.response.responseTime).to.be.below(200);

} )

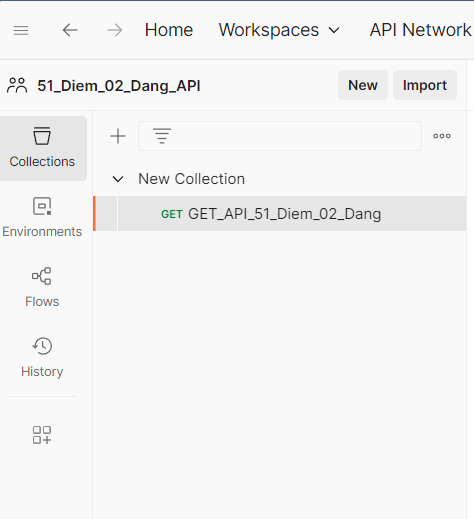


* Kết quả: Trong Test Results

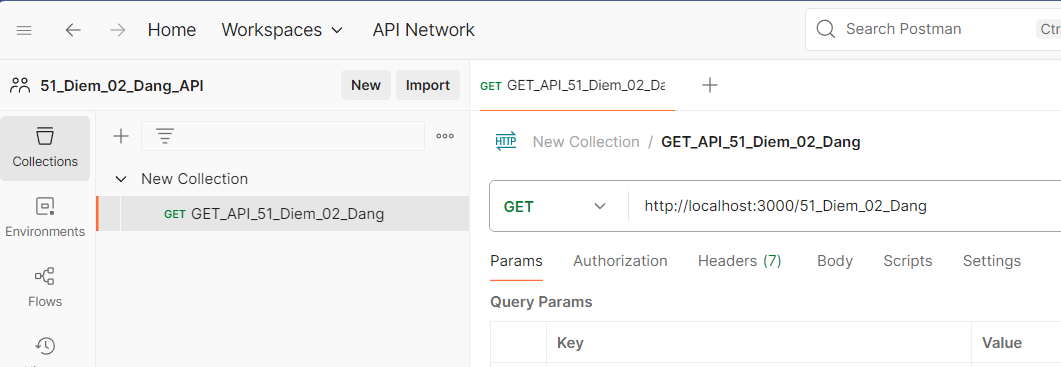


### Tạo request POST:

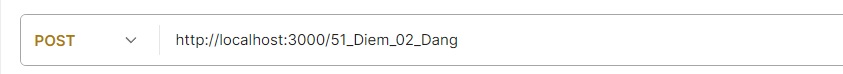
#### Sử dụng Workspace: 51\_Diem\_02\_Dang\_API



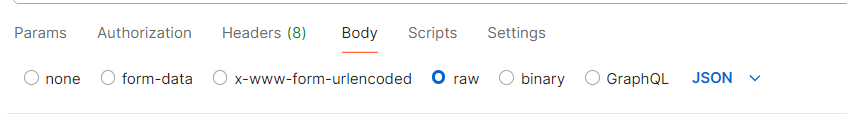
#### Nhấn dấu “+” để thêm một request mới:



#### Chọn POST -> Nhập link <http://localhost:3000/51_Diem_02_Dang>



#### Chọn Body -> Chọn raw -> Chọn JSON



#### Nhập dữ liệu mới muốn gửi lên server -> Nhấn Send

Code:

{

"id": "007",

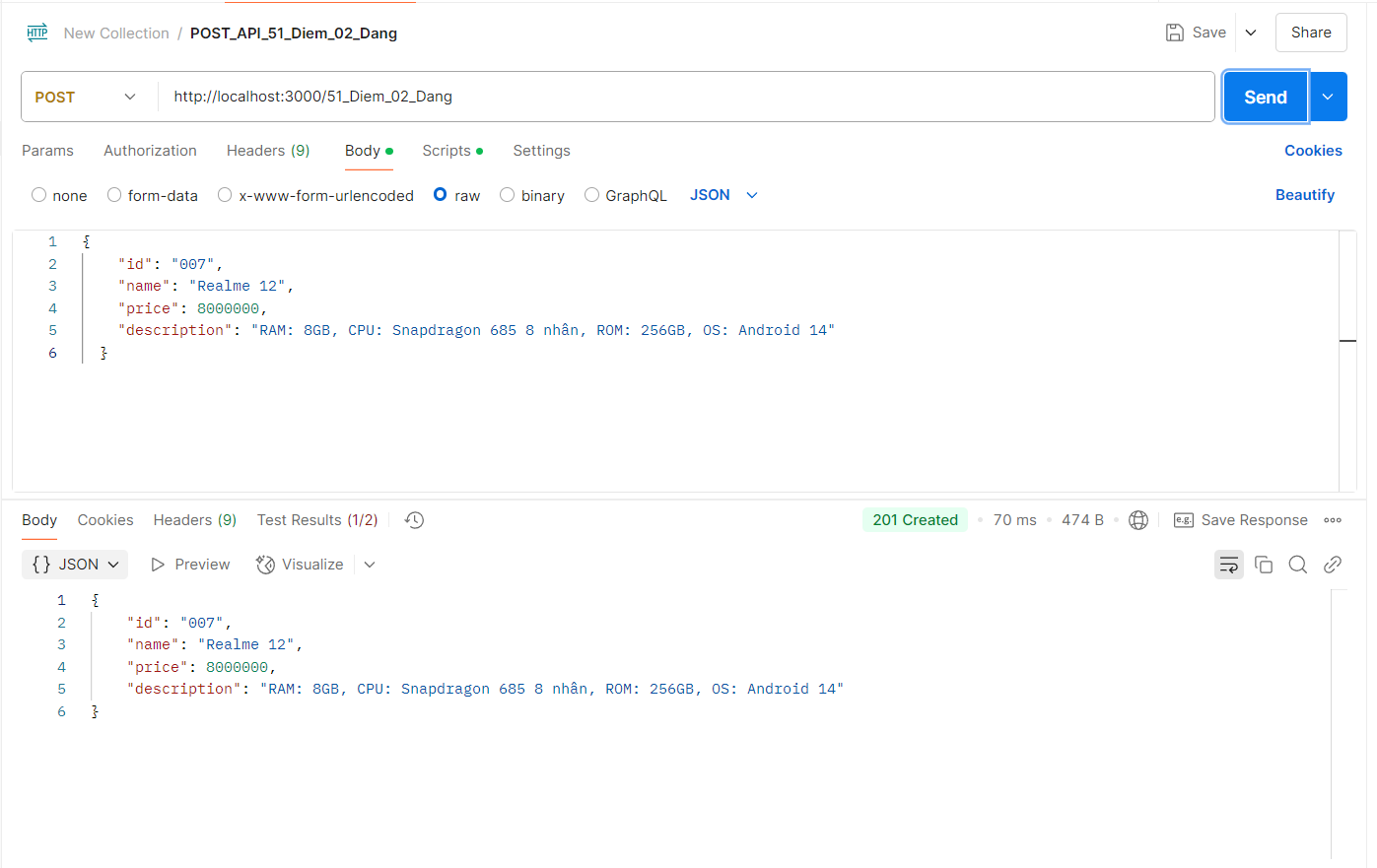
"name": "Realme 12",

"price": 8000000,

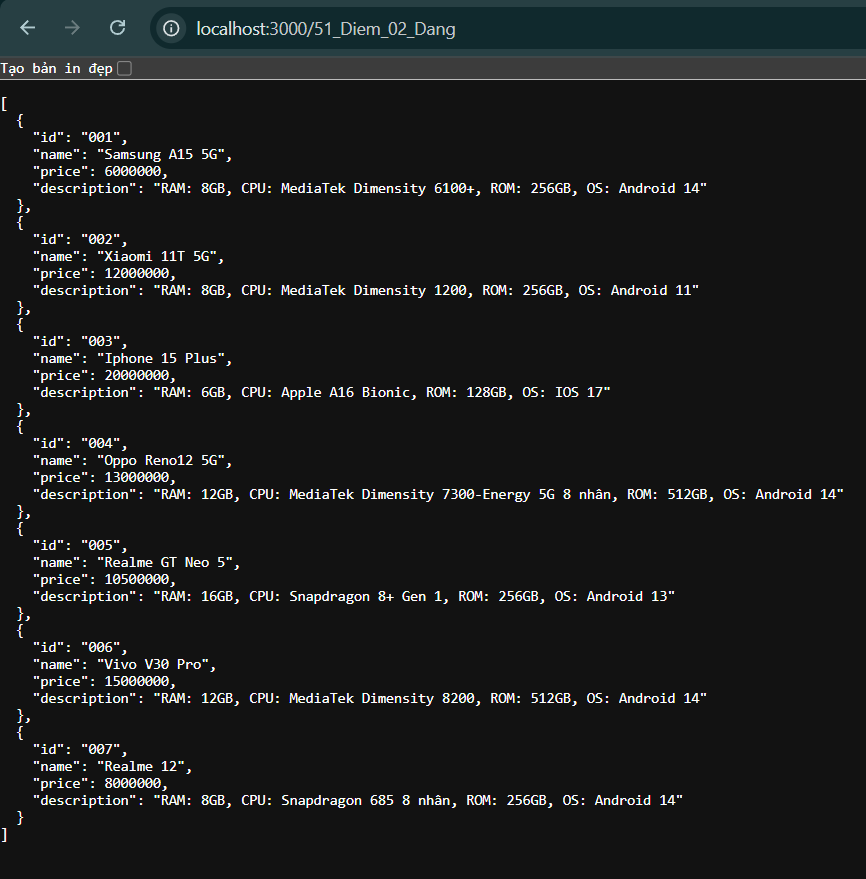
"description": "RAM: 8GB, CPU: Snapdragon 685 8 nhân, ROM: 256GB, OS: Android 14"

}

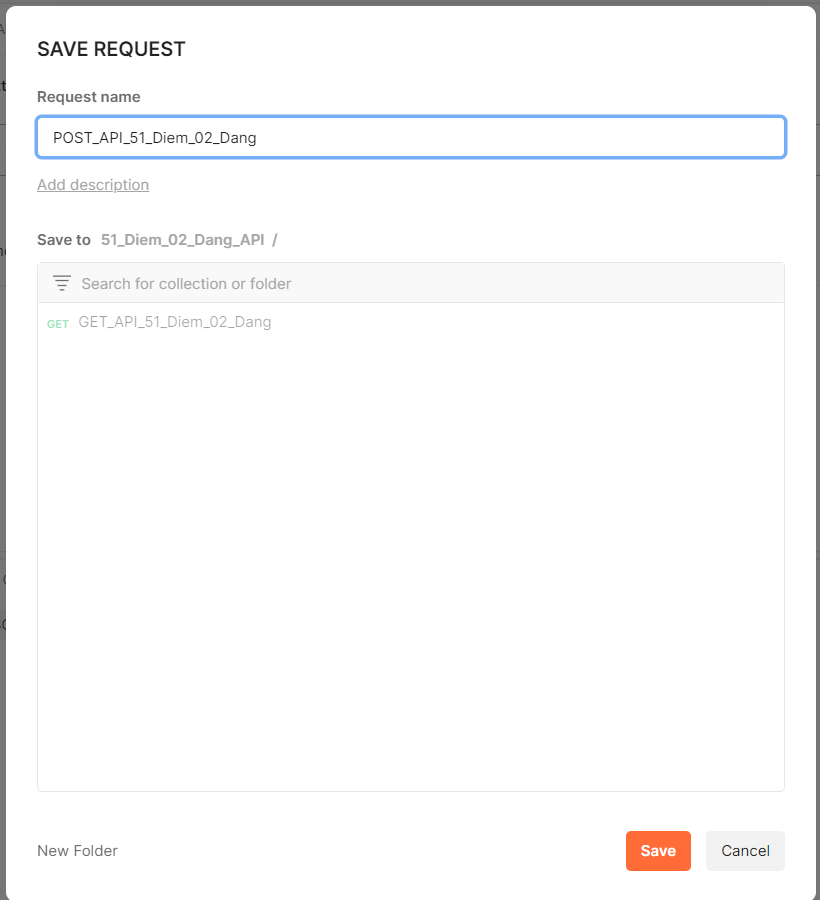
Kết quả:



#### Dữ liệu được lưu vào server:



#### Đặt tên -> Nhấn Save -> Lưu vào Project



#### Viết các test case trong test scripts:

* Code:

//Kiểm tra status code của POST request là thành công

pm.test("TC1\_51\_Diem\_02\_Dang", function () {

pm.expect(pm.response.code).to.be.oneOf([200, 201, 202]);

})

// Kiểm tra trường id trong response có phải là number hay không

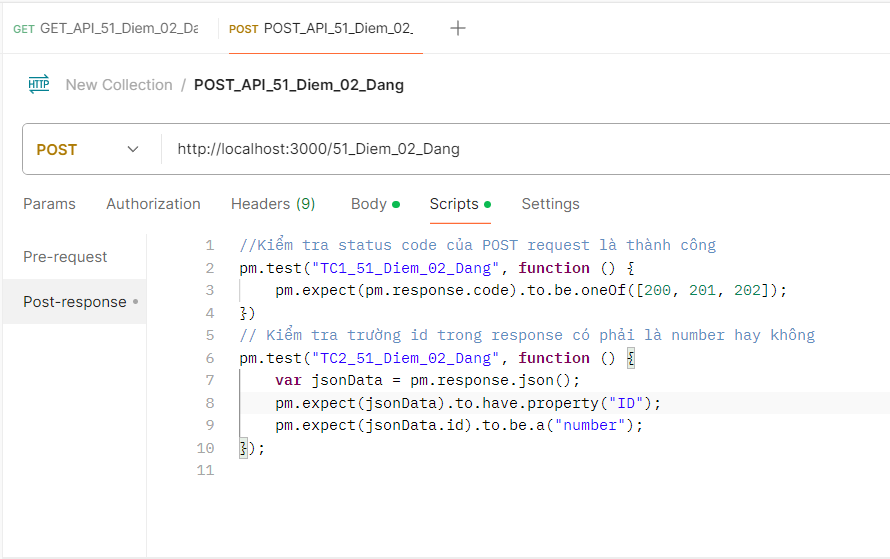
pm.test("TC2\_51\_Diem\_02\_Dang", function () {

var jsonData = pm.response.json();

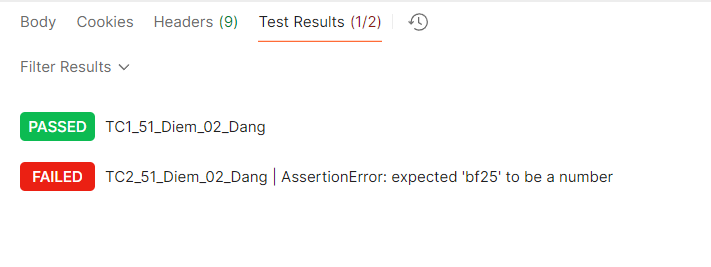
pm.expect(jsonData).to.have.property("id");

pm.expect(jsonData.id).to.be.a("number");

});

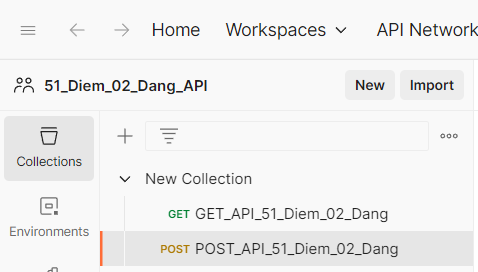


* Kết quả: Trong Test Results

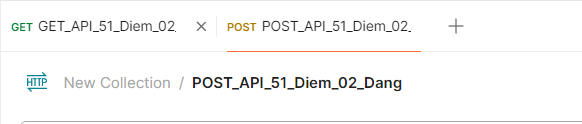


### Tạo request PUT:

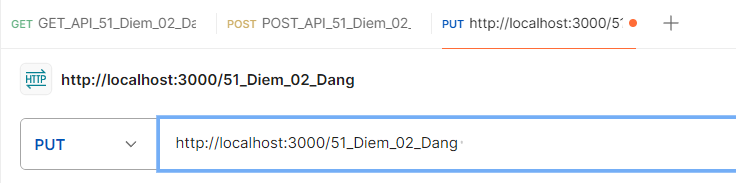
#### Sử dụng Workspace: 51\_Diem\_02\_Dang\_API



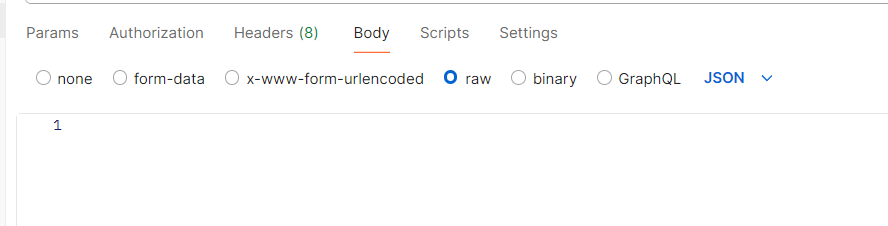
#### Nhấn dấu “+” để thêm một request mới:



#### Chọn PUT -> Nhập link <http://localhost:3000/51_Diem_02_Dang>



#### Chọn Body -> Chọn raw -> Chọn JSON



#### Nhập dữ liệu cập nhật cho dữ liệu có id là “006” lên server

{

"id": "006",

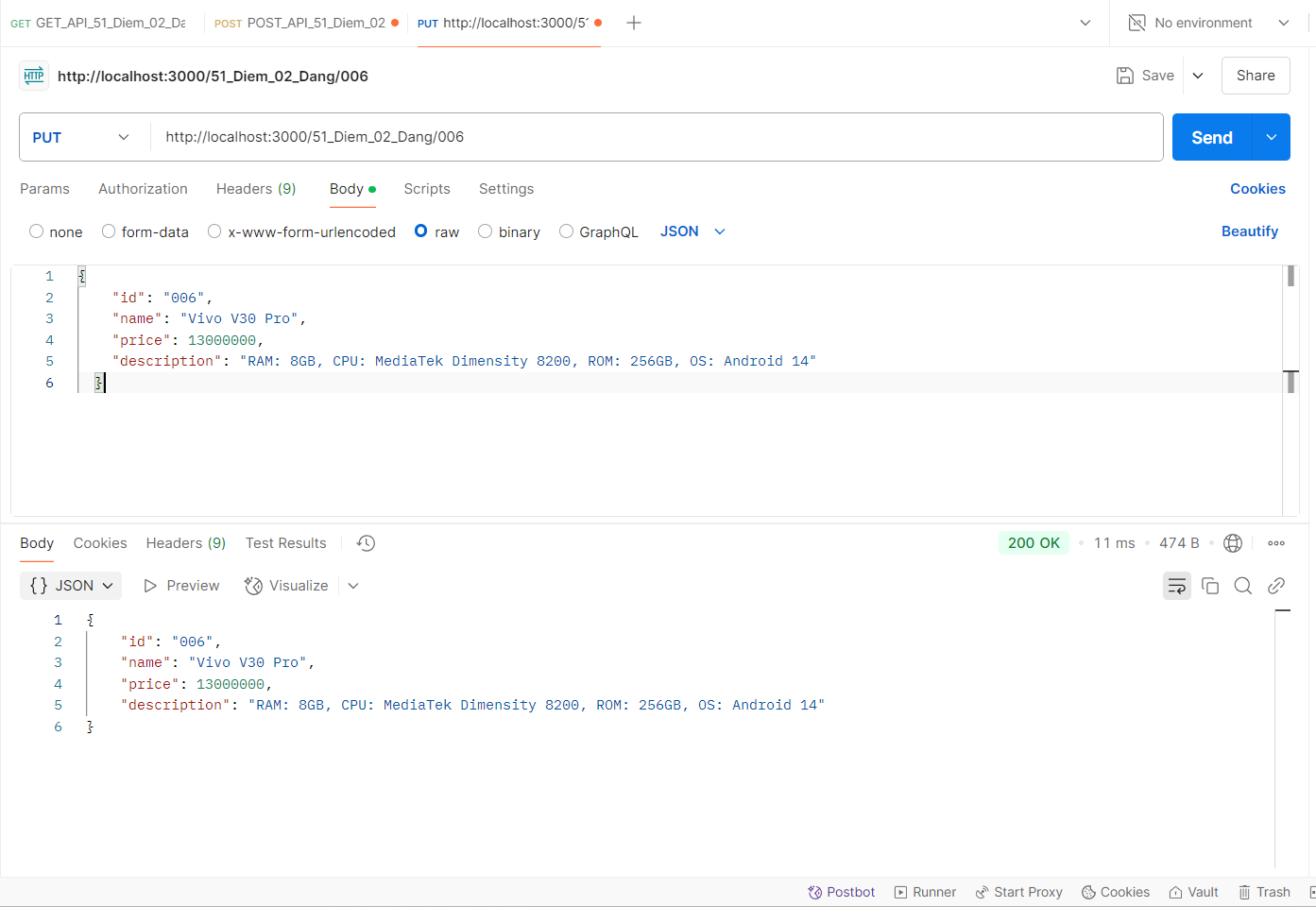
"name": "Vivo V30 Pro",

"price": 13000000,

"description": "RAM: 8GB, CPU: MediaTek Dimensity 8200, ROM: 256GB, OS: Android 14"

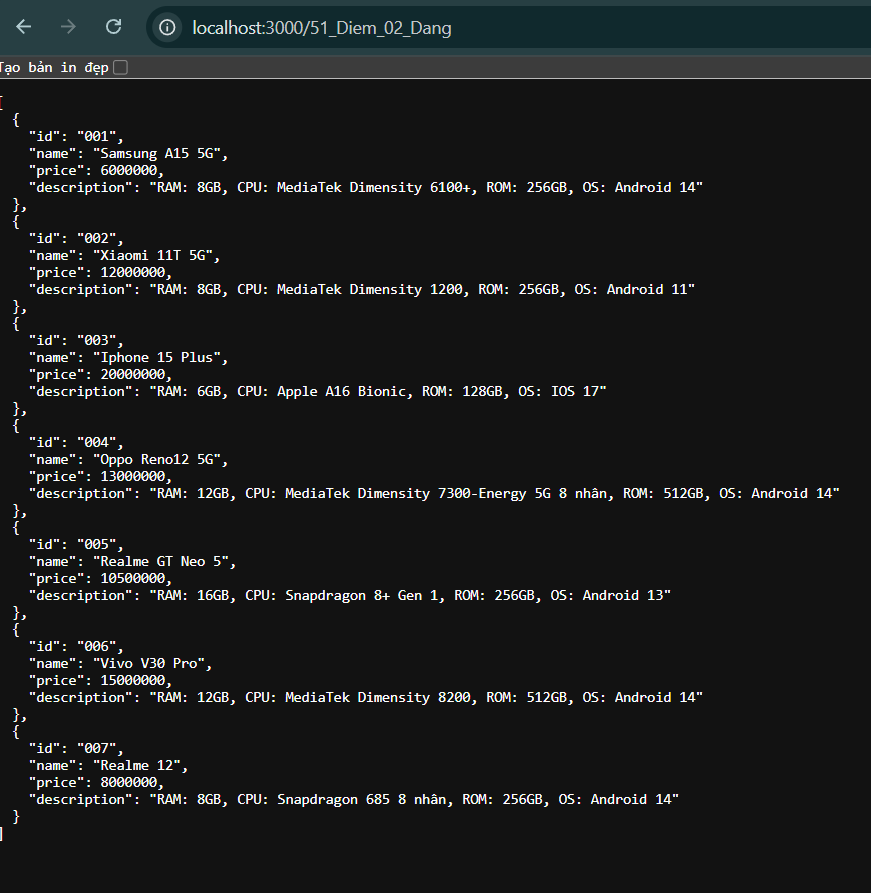
}

#### Sửa link thành <http://localhost:3000/51_Diem_02_Dang/006> (có id là 006) -> Nhấn Send

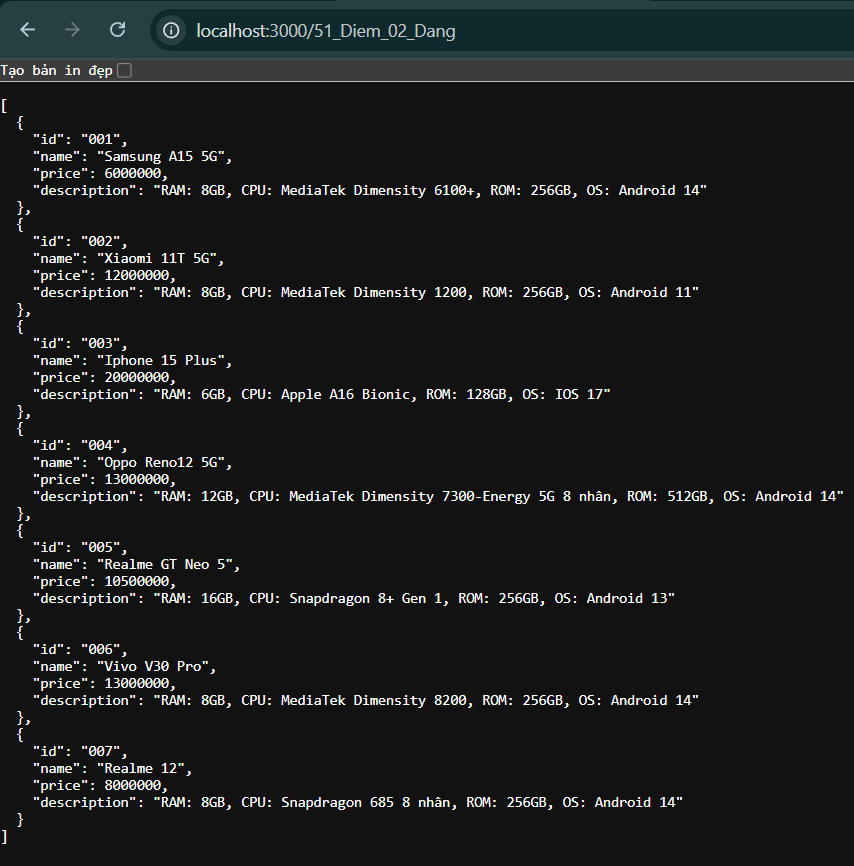


#### Dữ liệu được lưu vào server:

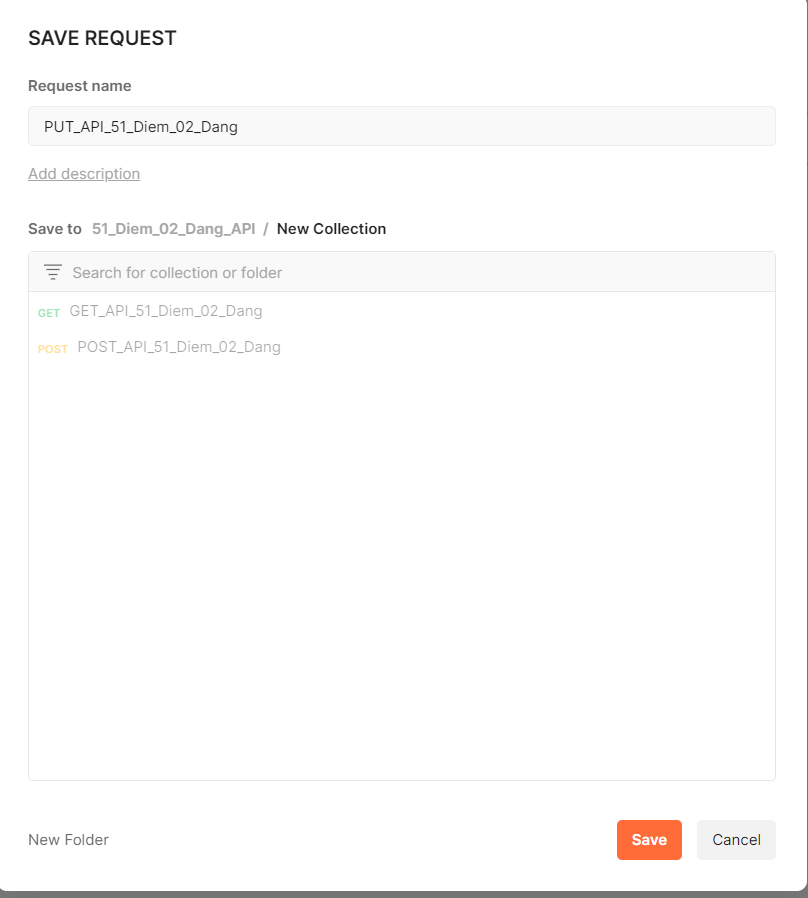
Dữ liệu cũ:



Dữ liệu mới:



#### Đặt tên -> Nhấn Save -> Lưu vào project



#### Viết các test case trong test scripts:

* Code:

// Kiểm tra status code là 200 OK hoặc 204 No Content

pm.test("TC1\_51\_Diem\_02\_Dang", function () {

pm.expect(pm.response.code).to.be.oneOf([200, 204]);

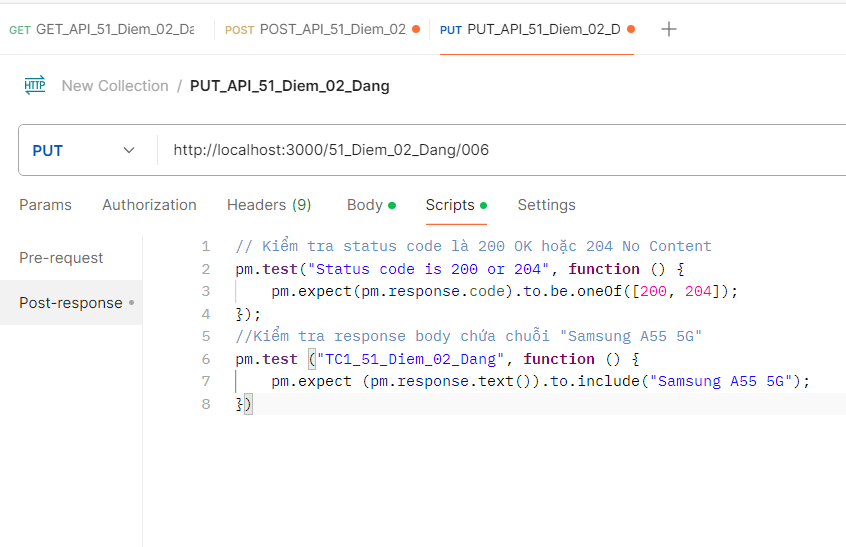
});

//Kiểm tra response body chứa chuỗi "Samsung A55 5G"

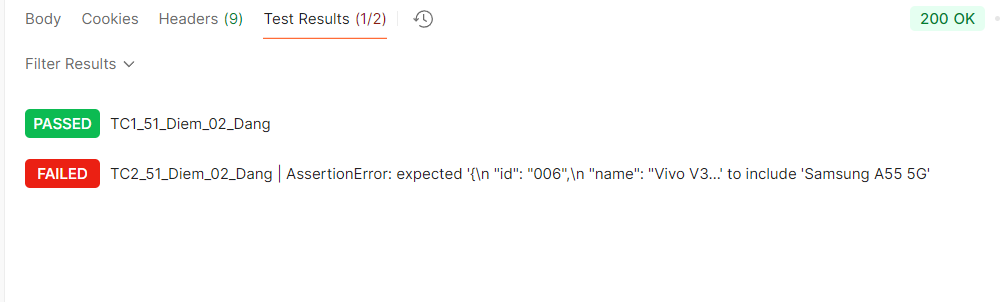
pm.test ("TC2\_51\_Diem\_02\_Dang", function () {

pm.expect (pm.response.text()).to.include("Samsung A55 5G");

})

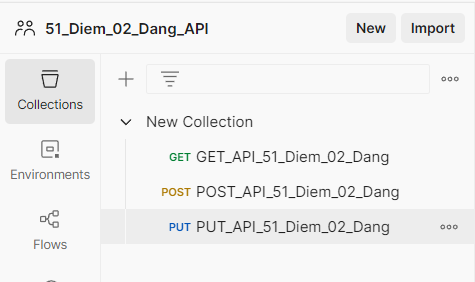


* Kết quả: Trong Test Results



### Tạo request DELETE:

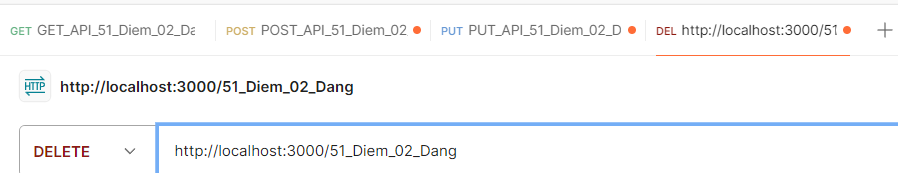
#### Sử dụng Workspace: 51\_Diem\_02\_Dang\_API



#### Nhấn dấu “+” để thêm một request mới:



#### Chọn DELETE -> Nhập link <http://localhost:3000/51_Diem_02_Dang>

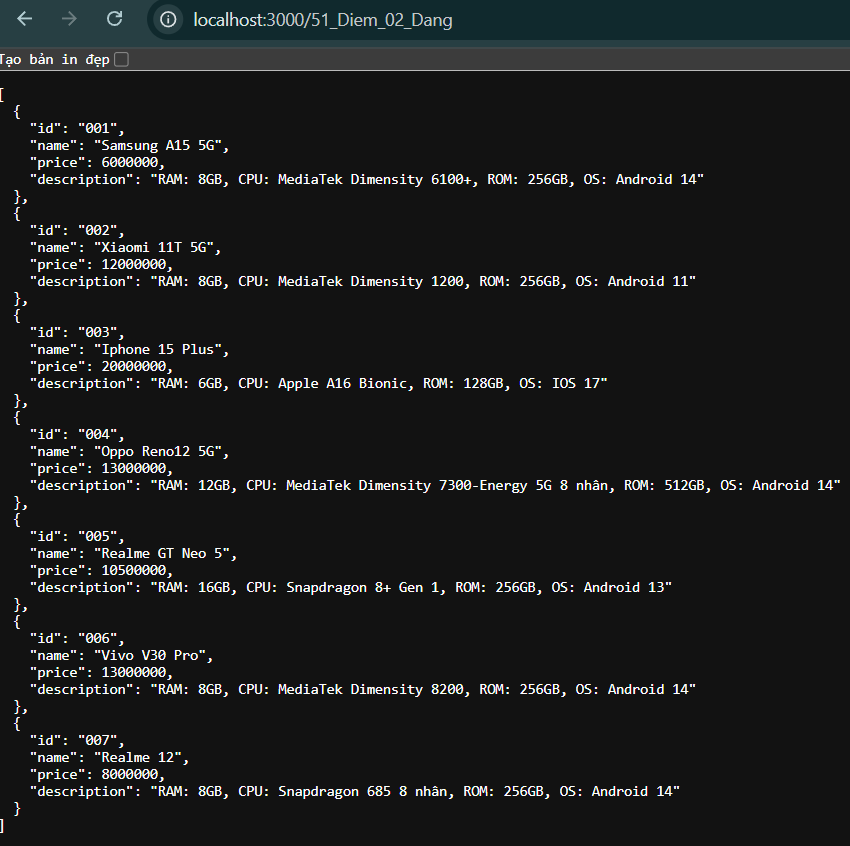


#### Sửa link thành <http://localhost:3000/51_Diem_02_Dang/007> (có id là 007) -> Nhấn Send để xóa

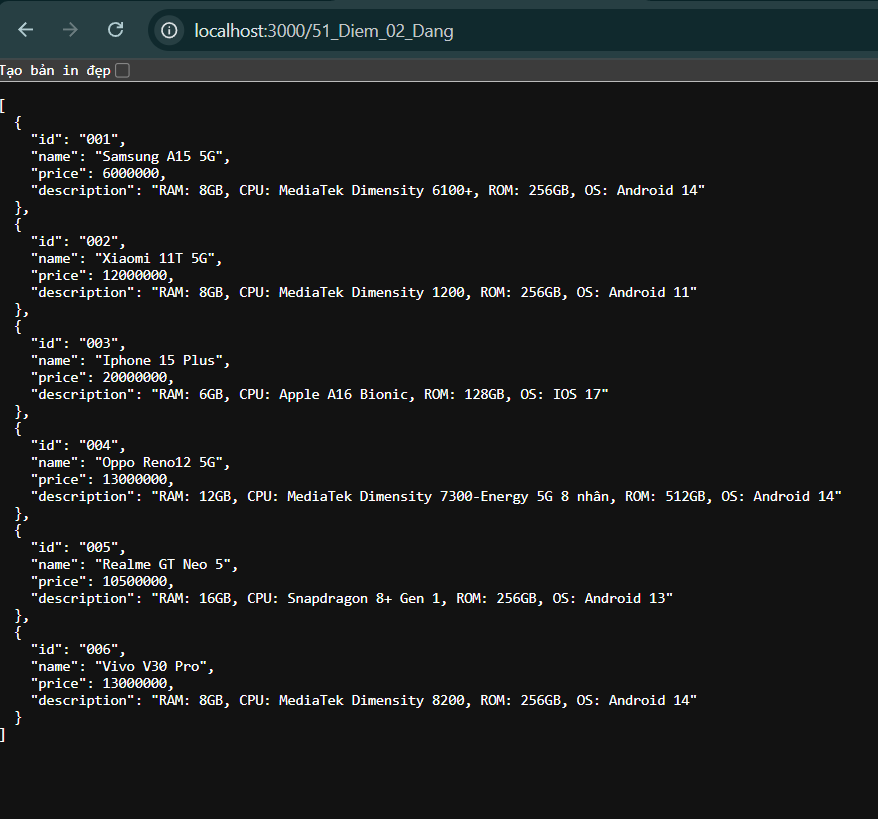


#### Dữ liệu được lưu vào server:

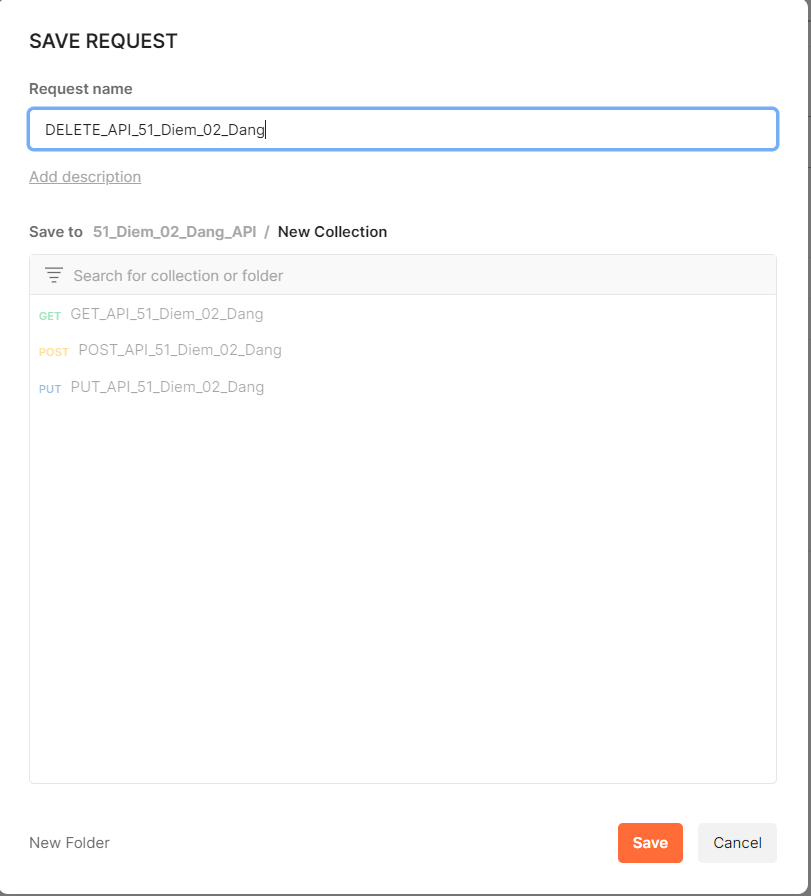
Trước khi xóa:



Sau khi xóa:



#### Đặt tên -> Nhấn Save -> Lưu vào Project



#### Viết các test case trong test scripts:

* Code:

//Kiểm tra thời gian phản hồi dưới 200 ms

pm.test ("TC1\_51\_Diem\_02\_Dang", function () {

pm.expect(pm.response.responseTime).to.be.below(200);

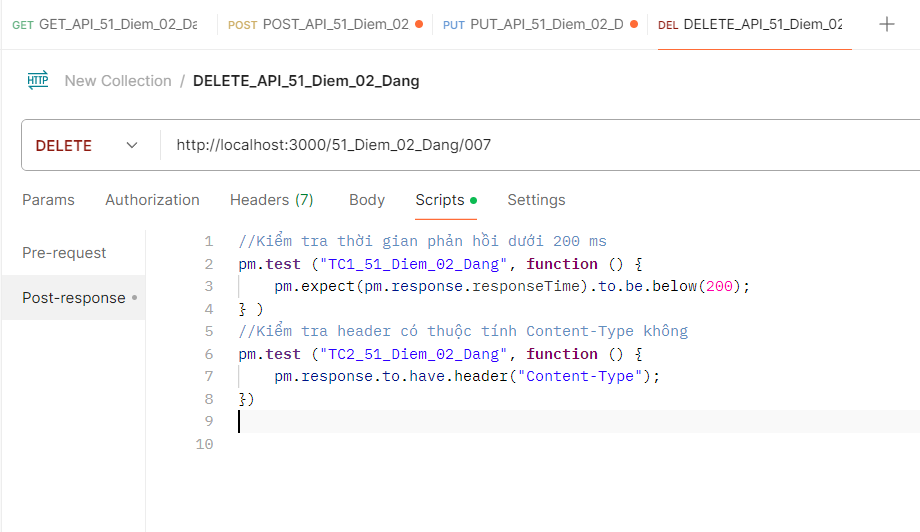
} )

//Kiểm tra header có thuộc tính Content-Type không

pm.test ("TC2\_51\_Diem\_02\_Dang", function () {

pm.response.to.have.header("Content-Type");

})



* Kết quả: Trong Test Results

