**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

--🙢🕮🙠—

****

**BÁO CÁO MÔN KIỂM THỬ PHẦN MỀM**

**ĐỀ TÀI: UNIT TEST CHƯƠNG TRÌNH TÍNH TOÁN PHÂN SỐ, KIỂM THỬ WEBSITE PRACTICE.AUTOMATIONTESTING.IN**

**VÀ POSTMAN API TESTING**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn: | Ths. Nguyễn Thị Ngọc Thanh |
| Sinh viên thực hiện: | Dương Trần Ngọc Hiếu |
| Mã số sinh viên: | 2151053018 |
| Lớp: | DH21IT02 |
| Khóa: | 2021 – 2025 |

**Thành phố Hồ Chí Minh, ngày 13 tháng 05 năm 2024**

**LỜI CẢM ƠN**

Để hoàn thành chuyên đề báo cáo thực tập này trước tiên em xin gửi đến Thạc sĩ cô giáo Nguyễn Thị Ngọc Thanh lời cảm ơn chân thành và sâu sắc nhất - người đã tận tình hướng dẫn, tạo điều kiện cho em tìm hiểu, cung cấp các số liệu và tài liệu cho em trong quá trình học tập và làm chuyên đề, giúp đỡ em hoàn thành chuyên đề báo cáo môn học này,

Trong quá trình làm bài chắc chắn khó tránh khỏi những thiếu sót. Do đó, em kính mong nhận được những lời góp ý của cô để bài báo cáo của em ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

[**CHƯƠNG 1. TẠO PROJECT, CÀI ĐẶT CÔNG CỤ VÀ THƯ VIỆN KIỂM THỬ** 4](#_Toc166697893)

[1.Tạo project 4](#_Toc166697894)

[2.Cài đặt Selenium WebDriver 7](#_Toc166697895)

[3.Tích hợp System.Data vào Testing Project 10](#_Toc166697896)

[**CHƯƠNG 2. UNIT TEST** 12](#_Toc166697897)

[1.Giới thiệu chương trình Tính Toán Phân Số 12](#_Toc166697898)

[2.Thiết kế giao diện 12](#_Toc166697899)

[3.Code chương trình 13](#_Toc166697900)

[4.Unit Testing 17](#_Toc166697901)

[Thiết kế Test Case 17](#_Toc166697902)

[Chạy Unit Test 18](#_Toc166697903)

[Cập nhật TestCase 21](#_Toc166697904)

[5. Unit Testing nhiều testcase với file csv 22](#_Toc166697905)

[**CHƯƠNG 3. SELENIUM WEB DRIVER** 31](#_Toc166697906)

[1. Mô tả 31](#_Toc166697907)

[2. Chức năng đăng nhập 31](#_Toc166697908)

[Test Case 31](#_Toc166697909)

[Thực hiện 33](#_Toc166697910)

[3. Chức năng thêm sản phẩm vào giỏ hàng 39](#_Toc166697911)

[Chuẩn bị 39](#_Toc166697912)

[Thực hiện 45](#_Toc166697913)

[**CHƯƠNG 4. POSTMAN API TESTING** 47](#_Toc166697914)

[1. Tạo Workspace và Collection 47](#_Toc166697915)

[2. Test các phương thức và viết script 52](#_Toc166697916)

[Phương thức GET 52](#_Toc166697917)

[Phương thức POST 57](#_Toc166697918)

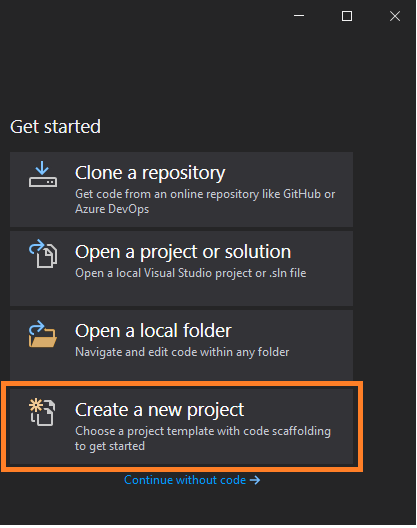
[Phương thức PUT 60](#_Toc166697919)

[Phương thức DELETE 63](#_Toc166697920)

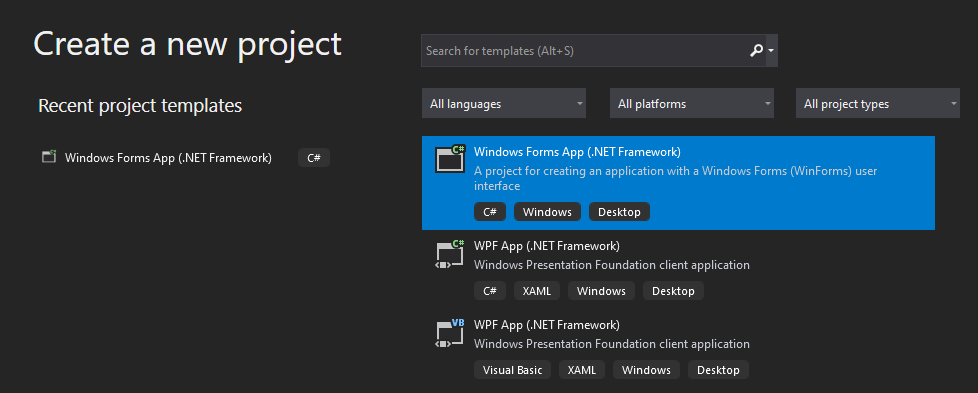
# **CHƯƠNG 1. TẠO PROJECT, CÀI ĐẶT CÔNG CỤ VÀ THƯ VIỆN KIỂM THỬ**

## 1.Tạo project

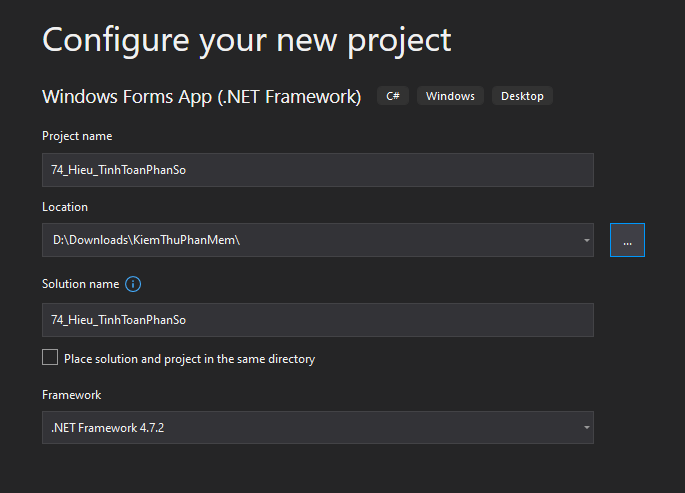
Bước 1: Mở Visual Studio, và chọn Create a new project



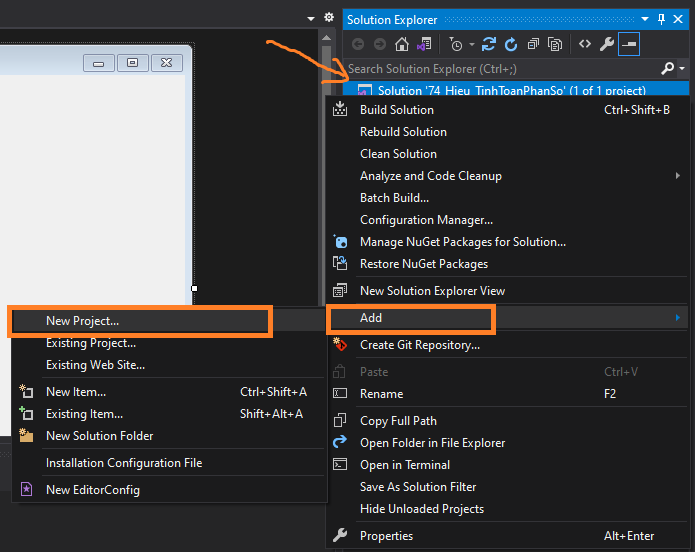
Bước 2: Chọn Windows Forms App (.NET Framework)



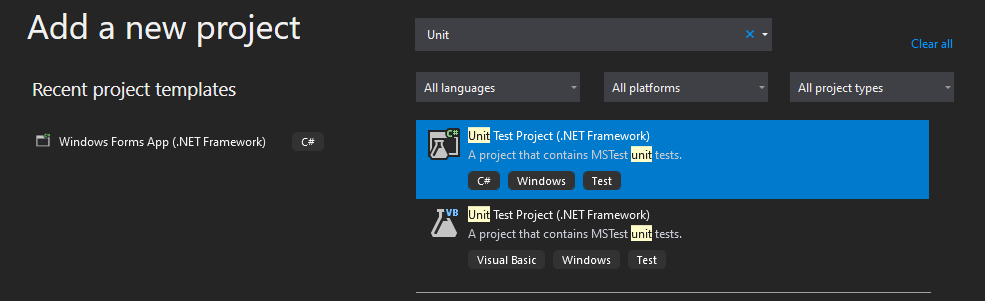
Bước 3: Đặt tên cho project là **74\_Hieu\_TinhToanPhanSo** và chọn Create



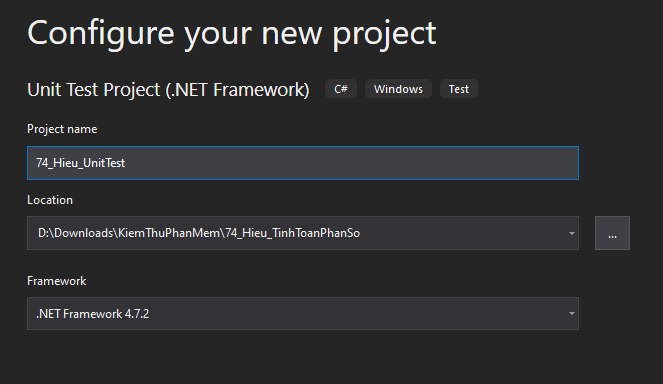
Bước 4: Thêm một project UnitTest bằng cách **right click** vào solution và **Add 🡪 New Project**



Bước 5: Tìm kiếm và chọn Unit Test Project (.NET Framework) với C#

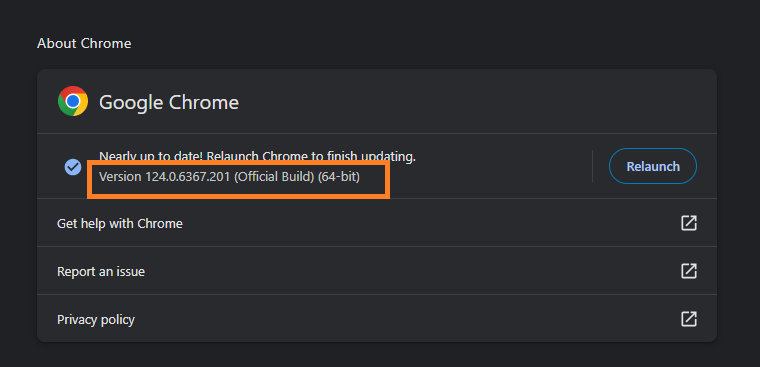


Bước 6: Đặt tên là **74\_Hieu\_UnitTest**, sau đó chọn Next

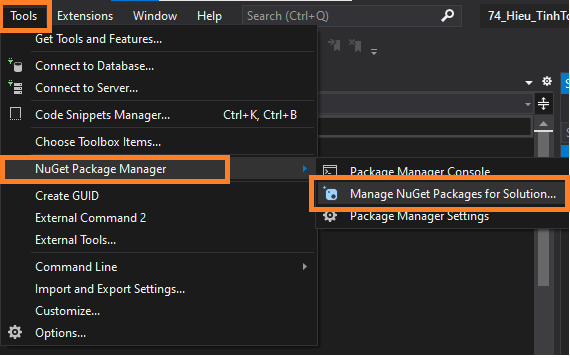


## 2.Cài đặt Selenium WebDriver

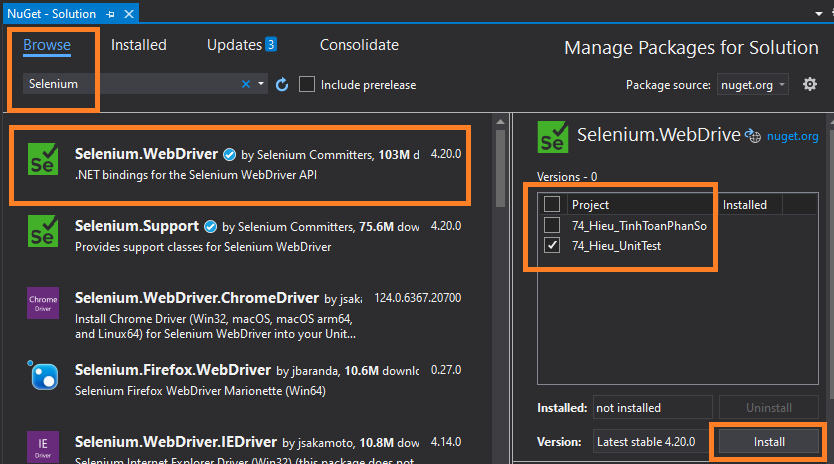
Bước 1: Kiểm tra phiên bản của Google Chrome bằng cách mở Chrome và truy cập link **chrome://settings/help**, ở đây ta có phiên bản 124.0.6367.201



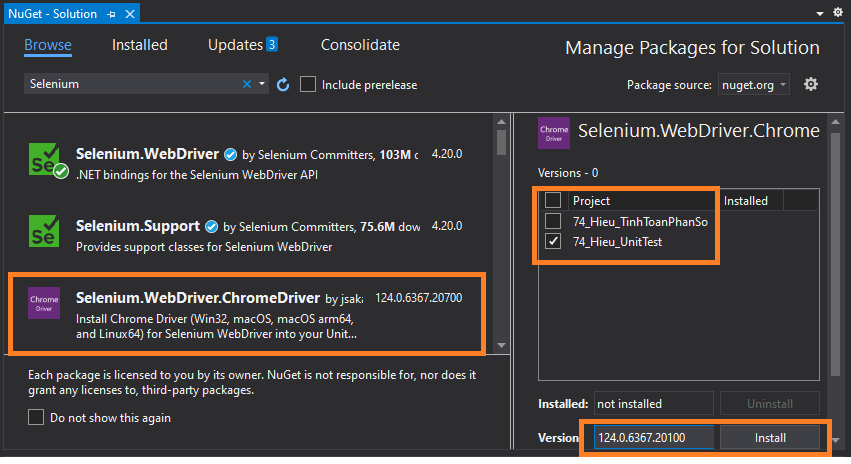
Bước 2: Vào **Tools 🡪 NuGet Package Manager 🡪 Manage NuGet Packages for Solution…**



Bước 3: Ở tab **Browse**, tìm kiếm từ khóa Selenium, chọn **Selenium.WebDriver**, sau đó chọn Project **74\_Hieu\_UnitTest** và **Install**

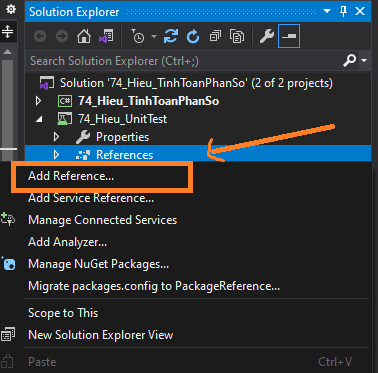


Bước 4: Tiếp tục cài đặt **Selenium.WebDriver.ChromeDriver**, chọn Project **74\_Hieu\_UnitTest**, sau đó chọn Version phù hợp với GoogleChrome là **124.0.6367.201,** cuối cùng **Install**

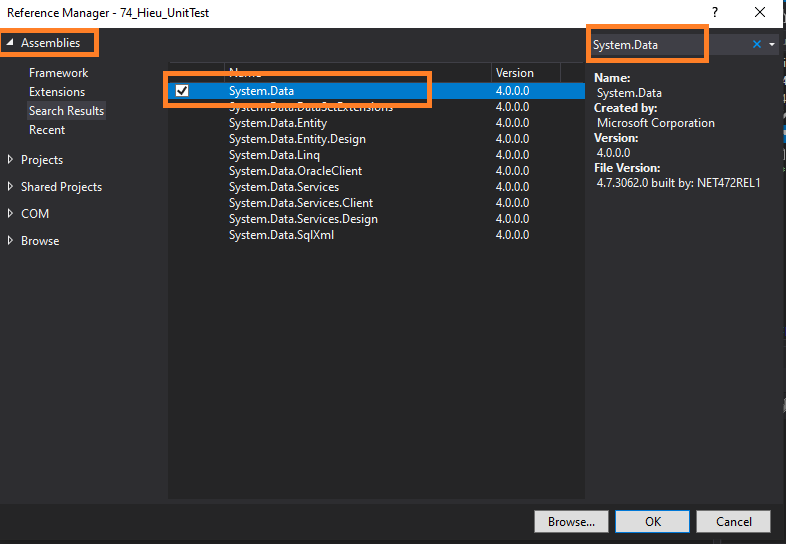


## 3.Tích hợp System.Data vào Testing Project

Bước 1: Tại Project **74\_Hieu\_UnitTest**, right click **References 🡪 Add References**



Bước 2: Tại **Assemblies**, tìm kiếm **System.Data**, chọn **System.Data**, sau đó **OK**



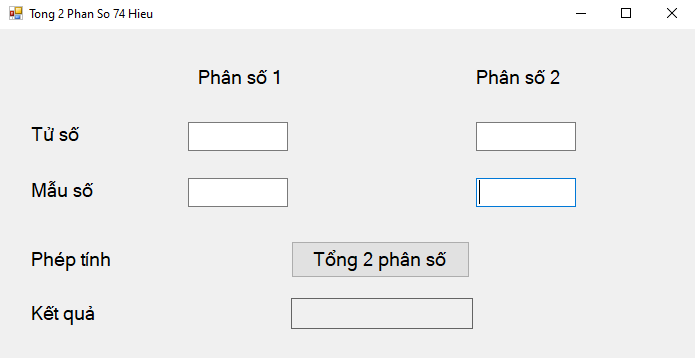
# **CHƯƠNG 2. UNIT TEST**

## 1.Giới thiệu chương trình Tính Toán Phân Số

Để minh họa cho phần Unit Test, em sẽ tạo một chương trình tính toán trên phân số, người dùng nhập vào tử số và mẫu s, và thực hiện phép tính cộng (với phép tính trừ, bản chất là cộng phân số có giá trị âm. Kết quả sẽ xuất ra là 1 phân số.

## 2.Thiết kế giao diện

Em thiết kế giao diện như sau:



Với các thành phần được đánh số, đặt thuộc tính (Name) lần lượt là:

(1) tuSo1\_74\_Hieu,

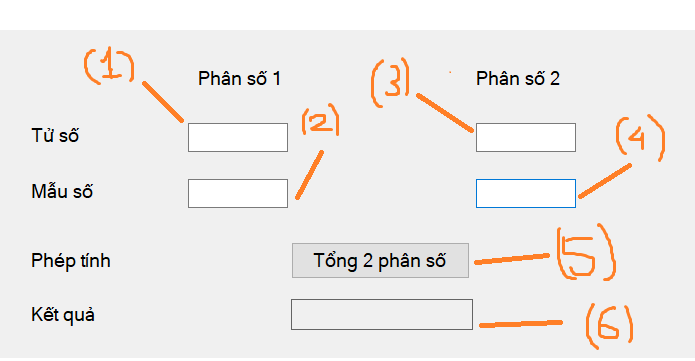
(2) mauSo1\_74\_Hieu,

(3) tuSo2\_74\_Hieu,

(4) mauSo2\_74\_Hieu

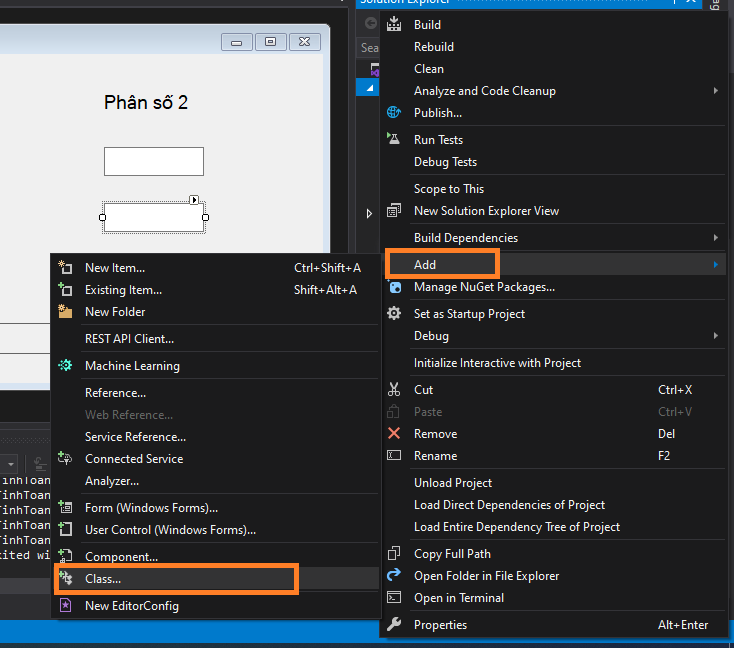
(5) buttonTong

(6) ketQua\_74\_Hieu

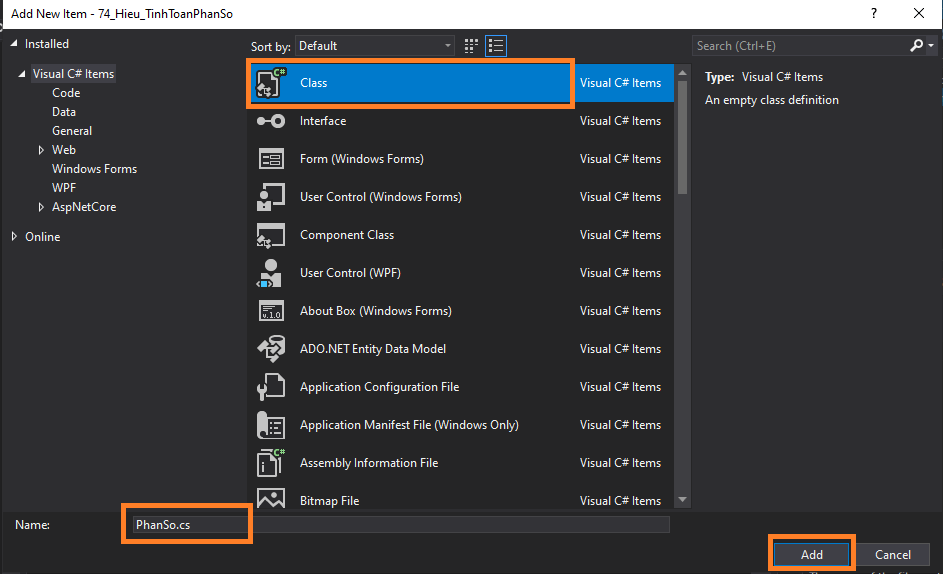


## 3.Code chương trình

Bước 1: Tạo Class PhanSo bằng cách right click vào Project **74\_Hieu\_TinhToanPhanSo 🡪 Add 🡪 Class…**



Bước 2: Đặt tên: **PhanSo.cs** và **Add**



Sau đó chép code:

public class PhanSo

{

private int tu\_74\_Hieu, mau\_74\_Hieu;

public PhanSo()

{

tu\_74\_Hieu = 0;

mau\_74\_Hieu = 1;

}

public PhanSo(int t, int m)

{

tu\_74\_Hieu = t;

mau\_74\_Hieu = m;

}

public int getTu()

{

return tu\_74\_Hieu;

}

public void setTu(int tu)

{

this.tu\_74\_Hieu = tu;

}

public int getMau()

{

return mau\_74\_Hieu;

}

public void setMau(int mau)

{

this.mau\_74\_Hieu = mau;

}

private int UCLN(int a, int b)

{

while (a != b)

if (a > b) a = a - b;

else b = b - a;

return a;

}

public void rutGon()

{

if (tu\_74\_Hieu \* mau\_74\_Hieu != 0)

{

int uoc = UCLN(Math.Abs(tu\_74\_Hieu), Math.Abs(mau\_74\_Hieu));

tu\_74\_Hieu /= uoc;

mau\_74\_Hieu /= uoc;

}

}

public PhanSo cong(PhanSo b)

{

PhanSo c = new PhanSo();

int kiemTraChiaCho0 = this.tu\_74\_Hieu / this.mau\_74\_Hieu;

kiemTraChiaCho0 = b.tu\_74\_Hieu / b.mau\_74\_Hieu;

//mục đích để kiểm tra có xuất hiện DivideByZeroException hay ko

c.tu\_74\_Hieu = this.tu\_74\_Hieu \* b.mau\_74\_Hieu + this.mau\_74\_Hieu \* b.tu\_74\_Hieu;

c.mau\_74\_Hieu = this.mau\_74\_Hieu \* b.mau\_74\_Hieu;

c.rutGon();

return c;

}

public String toString()

{

if(this.tu\_74\_Hieu == 0)

{

return "0";

}

if(this.mau\_74\_Hieu == 1)

{

return this.tu\_74\_Hieu.ToString();

}

return this.tu\_74\_Hieu + "/" + this.mau\_74\_Hieu;

}

}

Bước 3: Vào winform, click đúp vào button ‘**Tổng 2 phân số**’ để sinh ra hàm sự kiện click. Sau đó copy code:

private void buttonTong\_Click(object sender, EventArgs e)

{

int tu1\_74\_Hieu, mau1\_74\_Hieu;

tu1\_74\_Hieu = int.Parse(tuSo1\_74\_Hieu.Text);

mau1\_74\_Hieu = int.Parse(mauSo1\_74\_Hieu.Text);

PhanSo phanSo1\_74 = new PhanSo(tu1\_74\_Hieu, mau1\_74\_Hieu);

int tu2\_74\_Hieu, mau2\_74\_Hieu;

tu2\_74\_Hieu = int.Parse(tuSo2\_74\_Hieu.Text);

mau2\_74\_Hieu = int.Parse(mauSo2\_74\_Hieu.Text);

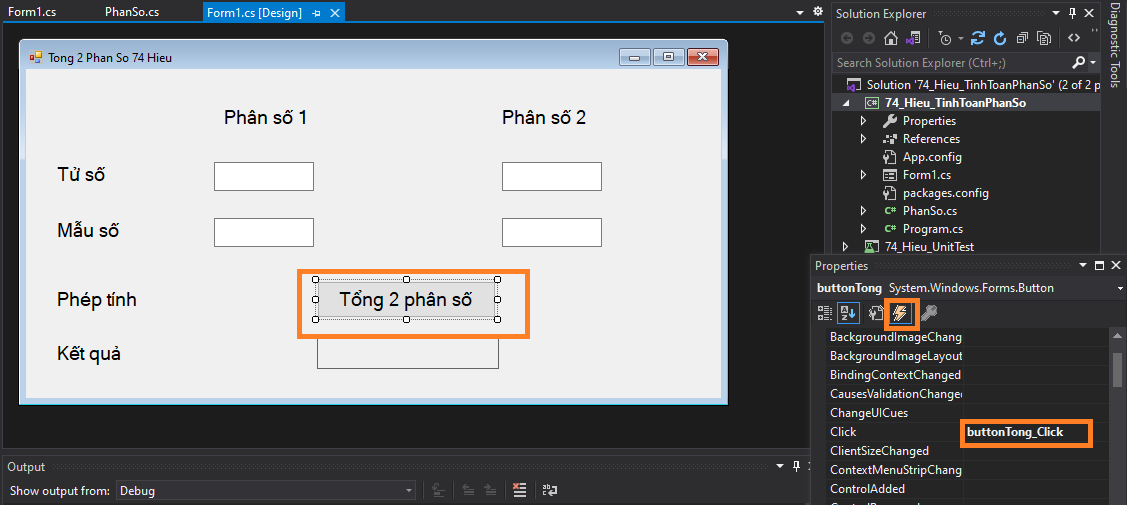
PhanSo phanSo2\_74 = new PhanSo(tu2\_74\_Hieu, mau2\_74\_Hieu);

PhanSo kq\_74\_Hieu = phanSo1\_74.cong(phanSo2\_74);

ketQua\_74\_Hieu.Text = kq\_74\_Hieu.toString();

}

Bước 4: Vào winform, chọn vào button 🡪 Chọn biểu tượng sấm sét🡪 Chọn hàm xử lý cho sự kiện Click



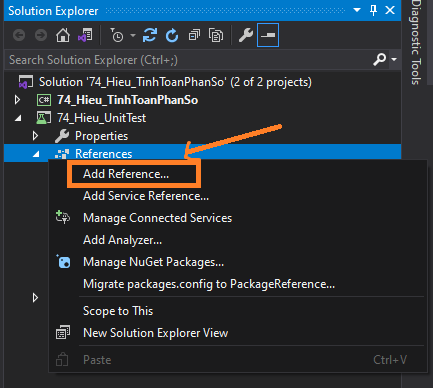
## 4.Unit Testing

### Thiết kế Test Case

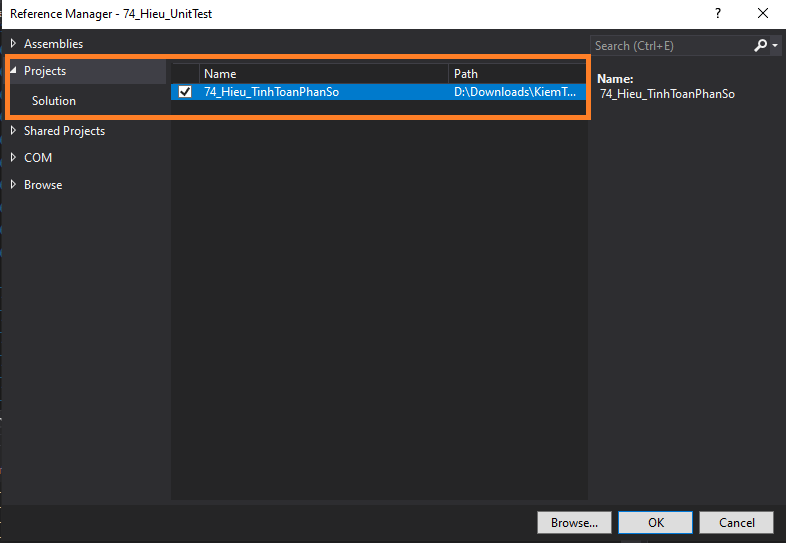
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Testcase  Description | Testcase  Procedure | Expected  Output | Test date | Result | Note |
| TC1\_TestCongPhanSo\_74\_Hieu | Kiểm tra kết quả tính toán của chương trình | 1. Cho phân số thứ 1 là ½  2. Cho phân số thứ 2 là ¾ | Tổng 2 phân số là 5/4. |  |  |  |
| TC2\_TestMauSoBang0\_74\_Hieu | Kiểm tra ngoại lệ phân số có mẫu số bằng 0 | 1. Cho phân số thứ 1 là ½  2. Cho phân số thứ 2 là 3/0 | Xuất hiện ngoại lệ DivideByZeroException |  |  |  |

### Chạy Unit Test

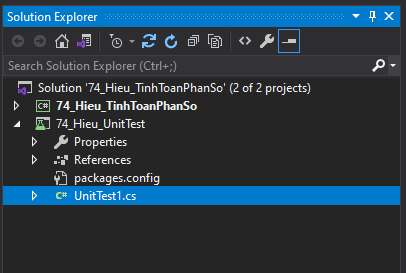
Bước 1: Ở project **74\_Hieu\_UnitTest 🡪 References 🡪Add Reference…**



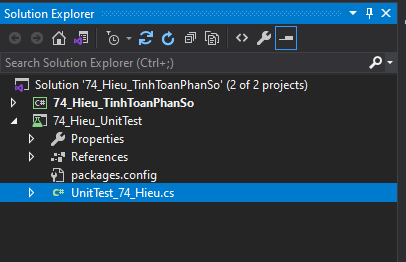
Sau đó chọn Project **74\_Hieu\_TinhToanPhanSo**, và **OK**



Bước 2: Khi tạo UnitTestProject, IDE đã tạo sẵn một file UnitTest1.cs



Em đổi tên file này thành UnitTest\_74\_Hieu.cs và sử dụng file này.



Chú ý: Khai báo dòng lệnh này trong file:

using \_74\_Hieu\_TinhToanPhanSo;

//do đặt tên có chữ số đứng đầu nên nó tự thêm kí tự \_ để hợp lệ

Code file **UnitTest\_74\_Hieu.cs**

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using \_74\_Hieu\_TinhToanPhanSo;

//do đặt tên có chữ số đứng đầu nên nó tự thêm kí tự \_ để hợp lệ

namespace \_74\_Hieu\_UnitTest

{

[TestClass]

public class UnitTest\_74\_Hieu

{

private PhanSo ps01\_74\_Hieu, ps02\_74\_Hieu;

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

public void SetUp()

{

ps01\_74\_Hieu = new PhanSo(1, 2);

ps02\_74\_Hieu = new PhanSo(3, 4);

}

[TestMethod]

public void TC1\_TestCongPhanSo\_74\_Hieu()

{

PhanSo actual;

PhanSo expected = new PhanSo(5, 4);

actual = ps01\_74\_Hieu.cong(ps02\_74\_Hieu);

Assert.AreEqual(expected.toString(), actual.toString());

}

[TestMethod]

[ExpectedException(typeof(DivideByZeroException))]

public void TC2\_TestMauSoBang0\_74\_Hieu()

{

ps02\_74\_Hieu = new PhanSo(3, 0);

ps01\_74\_Hieu.cong(ps02\_74\_Hieu);

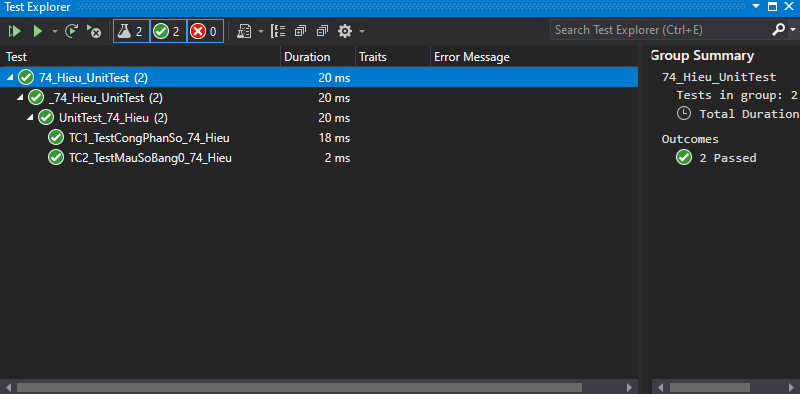
}

}

}

Bước 3: Nhấn tổ hợp phím Ctrl + R, A để Run All Tests.

Sau đây là kết quả:



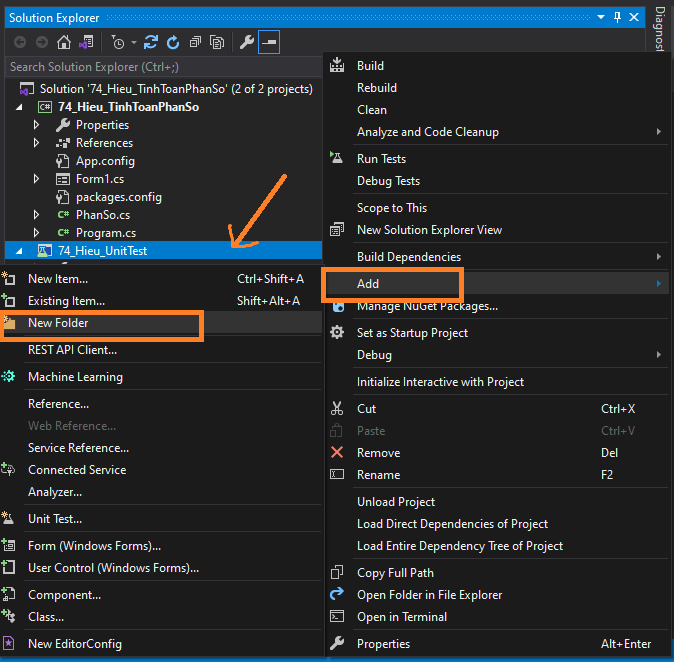
### Cập nhật TestCase

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Testcase  Description | Testcase  Procedure | Expected  Output | Test date | Result | Note |
| TC1\_TestCongPhanSo\_74\_Hieu | Kiểm tra kết quả tính toán của chương trình | 1. Cho phân số thứ 1 là ½  2. Cho phân số thứ 2 là ¾ | Tổng 2 phân số là 5/4. | 13/05/2024 | Pass |  |
| TC2\_TestMauSoBang0\_74\_Hieu | Kiểm tra ngoại lệ phân số có mẫu số bằng 0 | 1. Cho phân số thứ 1 là ½  2. Cho phân số thứ 2 là 3/0 | Xuất hiện ngoại lệ DivideByZeroException | 13/05/2024 | Pass |  |

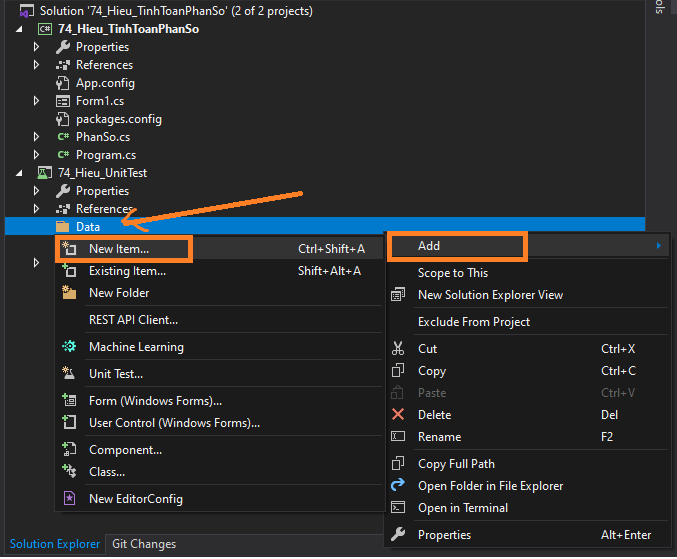
## 5. Unit Testing nhiều testcase với file csv

Chú thích: cài System.Data chính là để phục vụ cho công việc này

Bước 1: Tạo thư mục tên ‘Data’: right click vào Project **74\_Hieu\_UnitTest 🡪 Add 🡪 New Folder 🡪 Đặt tên ‘Data’**

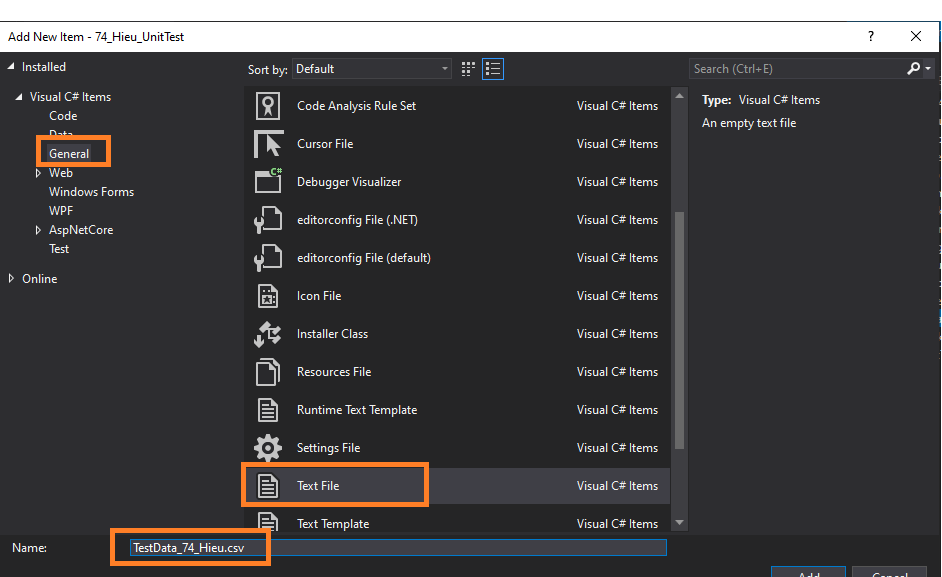


Add vào thư mục Data một Item mới



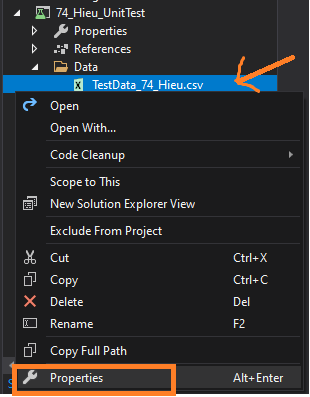
Chọn mục **General** 🡪 **Text File** 🡪 đặt tên là **TestData\_74\_Hieu.csv**

🡪 Add

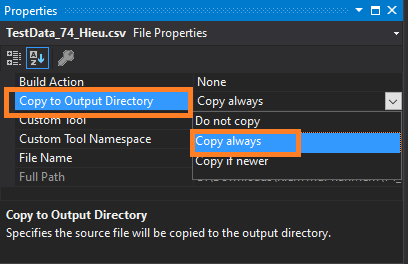


Bước 2:

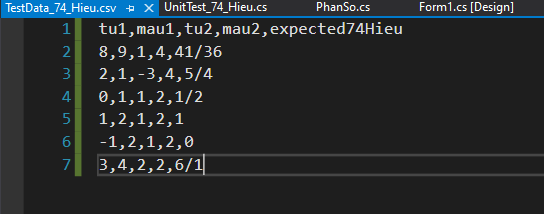
Xem **Properties** của file



Gán trường **Copy to Output Directory** là **Copy always**

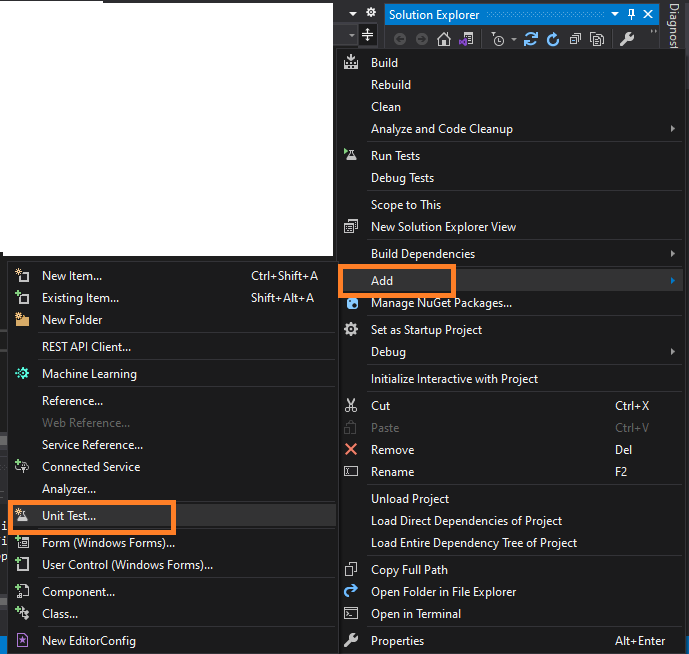


Bước 3: Ghi nội dung file TestData\_74\_Hieu.csv

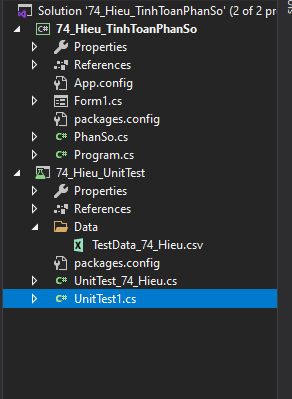


Lưu ý: ở đây có TestCase dòng 7 sẽ có kết quả fail

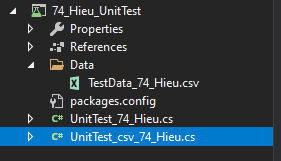
Bước 4: Tạo một file UnitTest mới ở Project **74\_Hieu\_UnitTest**



Ta được file UnitTest1.cs



Đổi tên là: UnitTest\_csv\_74\_Hieu.cs



Bước 5: Code của UnitTest\_csv\_74\_Hieu.cs

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using \_74\_Hieu\_TinhToanPhanSo;

namespace \_74\_Hieu\_UnitTest

{

[TestClass]

public class UnitTest\_csv\_74\_Hieu

{

public TestContext TestContext { get; set; }

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

@".\Data\TestData\_74\_Hieu.csv", "TestData\_74\_Hieu#csv",

DataAccessMethod.Sequential)]

[TestMethod]

public void TestWithDataSoure74Hieu()

{

int tu1\_74\_Hieu = int.Parse(TestContext.DataRow[0].ToString());

int mau1\_74\_Hieu = int.Parse(TestContext.DataRow[1].ToString());

int tu2\_74\_Hieu = int.Parse(TestContext.DataRow[2].ToString());

int mau2\_74\_Hieu = int.Parse(TestContext.DataRow[3].ToString());

string expected = TestContext.DataRow[4].ToString();

PhanSo ps01 = new PhanSo(tu1\_74\_Hieu, mau1\_74\_Hieu);

PhanSo ps02 = new PhanSo(tu2\_74\_Hieu, mau2\_74\_Hieu);

PhanSo actual = ps01.cong(ps02);

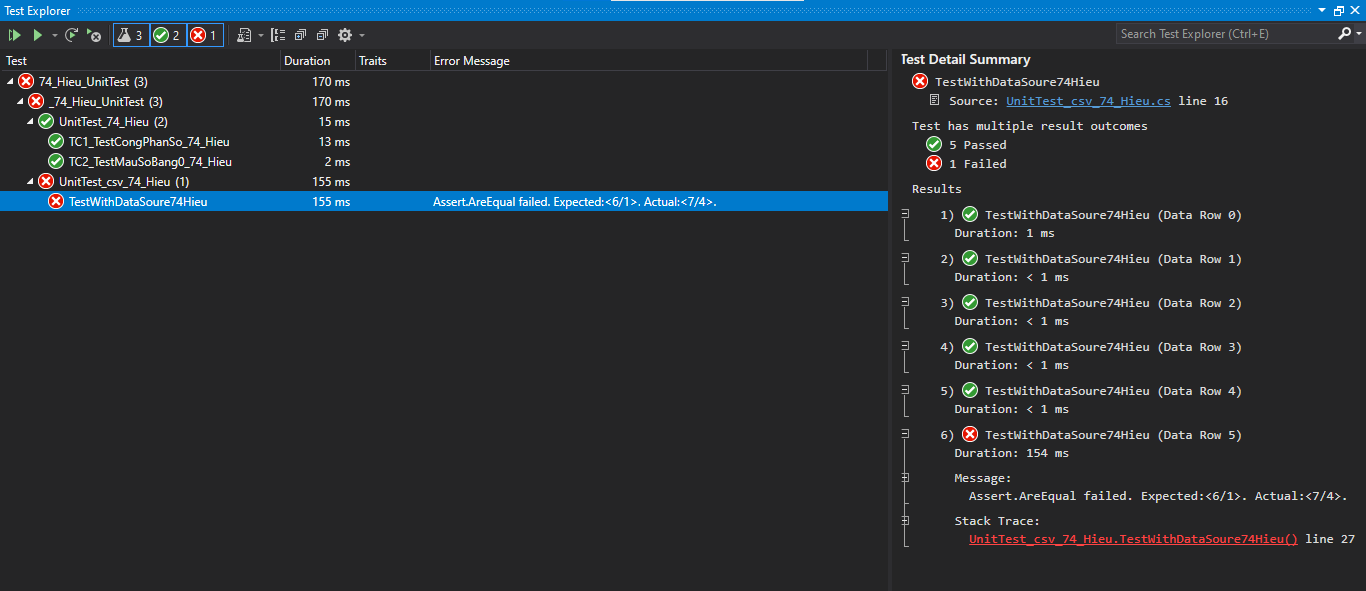
Assert.AreEqual(expected, actual.toString());

}

}

}

Kết quả: TestCase cuối cùng với giá trị expected khác với actual nên Fail



# **CHƯƠNG 3. SELENIUM WEB DRIVER**

## 1. Mô tả

- Chức năng đăng nhập:

+ Truy cập <https://practice.automationtesting.in/>

+ Click vào My Account

+ Nhập username và password, sau đó đăng nhập

- Chức năng tìm kiếm và thêm vào giỏ hàng:

+ Tìm kiếm từ khóa ‘javascript’ và enter

+ Chọn ‘Mastering JavaScript’

+ Chọn ‘ADD TO BASKET’

## 2. Chức năng đăng nhập

### Test Case

- Bảng quyết định (Decision table):

Chú thích:

B –Blank, V –Valid, I –Invalid

M1 : Username is required.

M2: Password is required.

M3: The username is not registered on this site. If you are unsure of your username, try your email address instead.

M4: The password you entered for the username is incorrect.

L : Trang Login, A: Trang tài khoản đã đăng nhập

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Các điều kiện | Username | B | B | B | I | I | I | V | V | V |
| Password | B | I | V | B | I | V | B | I | V |
| Các hành động | Thông điệp lỗi | M1 | M1 | M1 | M2 | M3 | M3 | M2 | M4 |  |
| Chuyển đến trang | L | L | L | L | L | L | L | L | A |

Rút gọn bảng quyết định:

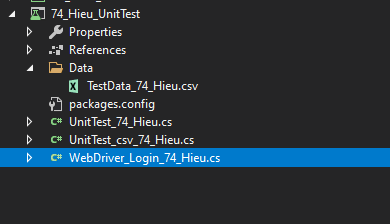
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Các điều kiện | Username | B | I/V | I | V | V |
| Password | - | B | I/V | I | V |
| Các hành động | Thông điệp lỗi | M1 | M2 | M3 | M4 |  |
| Chuyển đến trang | L | L | L | L | A |

Ta có **Bốn** testcase để kiểm tra việc đăng nhập thất bại. **Một** Test case còn lại là testcase đăng nhập thành công.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Testcase  Description | Testcase  Procedure | Expected  Output | Test date | Result | Note |
| LoginFail\_TC1\_74\_Hieu | Kiểm tra đăng nhập thất bại | Username trống, password trống | M1: Username is required. | 14/05/2024 |  |  |
| LoginFail\_TC2\_74\_Hieu | Username sai, password trống | M2: Password is required. | 14/05/2024 |  |  |
| LoginFail\_TC3\_74\_Hieu | Username sai, password sai | M3: The username is not registered on this site. If you are unsure of your username, try your email address instead. | 14/05/2024 |  |  |
| LoginFail\_TC4\_74\_Hieu | Username đúng = **2151053018hieu**, password sai | M4: The password you entered for the username is incorrect. | 14/05/2024 |  |  |
| TC5\_Success\_74\_Hieu | Kiểm tra đăng nhập thành công | Username đúng = **2151053018hieu**, password đúng = **H!eu123456#** | Chuyển đến trang tài khoản đã đăng nhập | 14/05/2024 |  |  |

### Thực hiện

Bước 1: Trước tiên em tạo một file **Unit Test** mới trong project **74\_Hieu\_UnitTest**, cách tạo như trên. Và đặt tên là **WebDriver\_Login\_74\_Hieu.cs**



Bước 2: Thêm thư viện vào file vừa tạo

Chú thích: cài Selenium chính là để phục vụ cho công việc này

using OpenQA.Selenium;

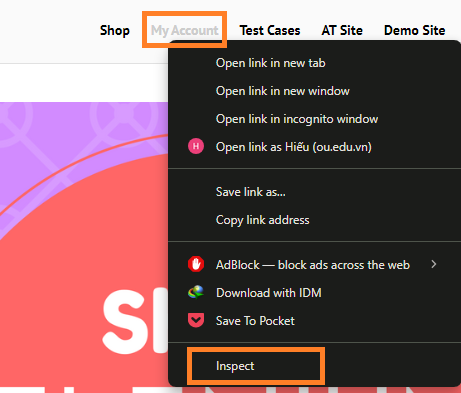
using OpenQA.Selenium.Chrome;

using System.Threading; // để sử dụng Thread.Sleep() dừng lại 1 chút để kịp theo dõi Running Test

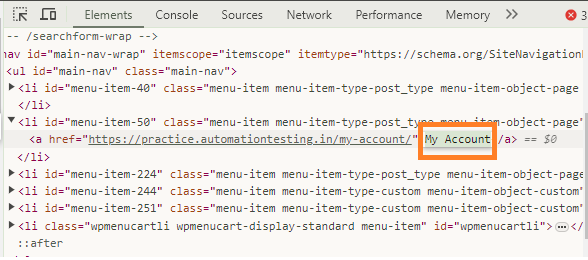
Bước 3: Truy cập <https://practice.automationtesting.in/>

* Lấy element **My Account**

Right Click vào ‘My Account’, chọn **Inspect**

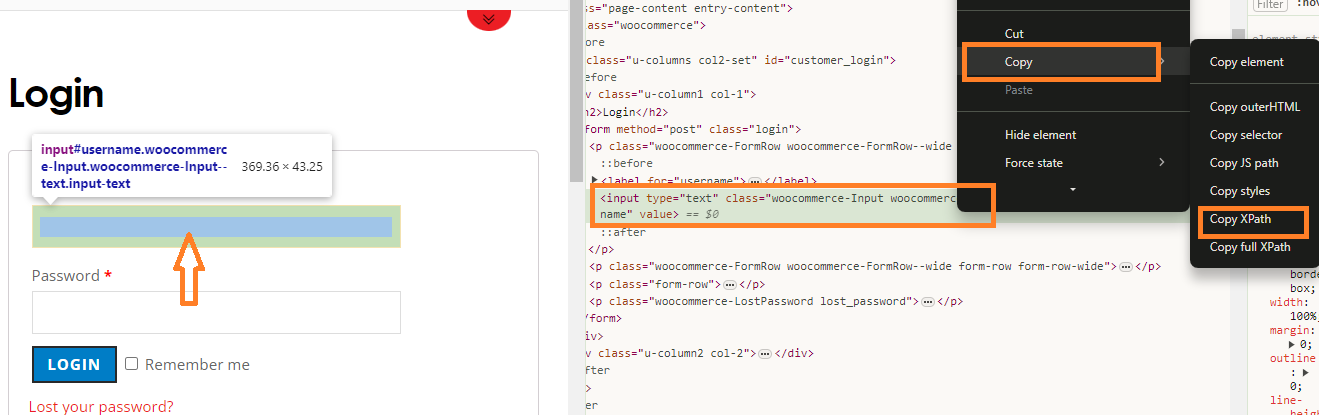


Lấy Link Text (minh họa cho lấy element bằng link text)



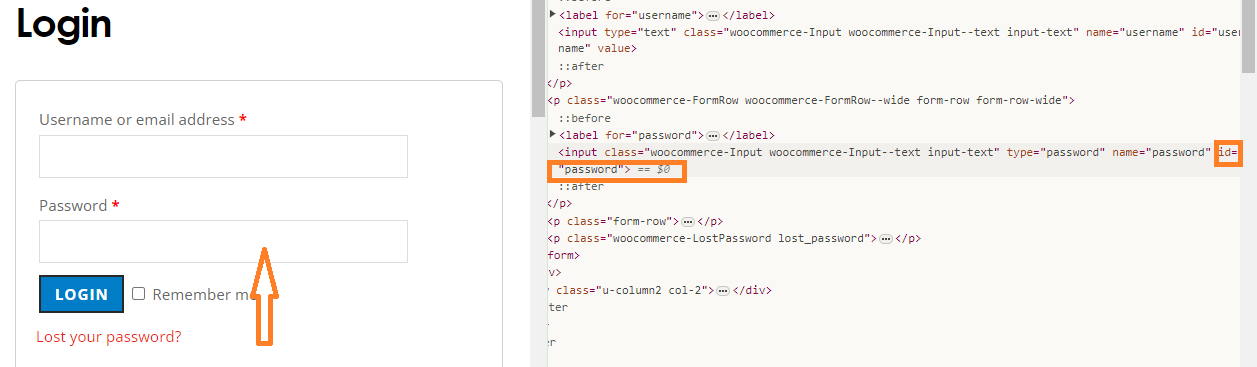
Ta có linkText: **My Account**

* Lấy element ô nhập **username**, tương tự như trên nhưng minh họa lấy bằng Xpath



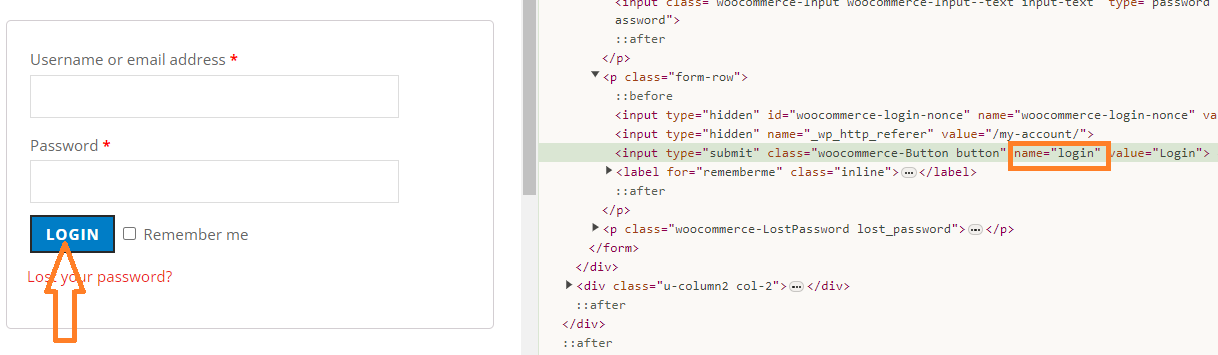
Ta có Xpath: //\*[@id="username"]

* Tương tự, lấy ô nhập **password**, minh họa lấy bằng ID



Ta có Id: **password**

* Lấy button **Login**, minh họa lấy bằng Name



Ta được trường name: **login**

Bước 4: Tạo các biến element trong code để sử dụng chung cho 5 Testcase

IWebDriver driver\_74\_Hieu;

IWebElement element\_MyAccount, element\_Username, element\_password, element\_Login;

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

public void SetUp()

{

// tạo driver

driver\_74\_Hieu = new ChromeDriver();

// gán trang cần navigate

driver\_74\_Hieu.Url = "https://practice.automationtesting.in/";

element\_MyAccount = driver\_74\_Hieu.FindElement(By.LinkText("My Account"));

}

Bước 5: Tạo các testcase trong code

[TestMethod]

public void LoginFail\_TC1\_74\_Hieu()

{

// truy cập trang web

//Cách 1: driver\_74\_Hieu.Navigate().GoToUrl("https://practice.automationtesting.in/");

driver\_74\_Hieu.Navigate();

// Click vào MyAccount

element\_MyAccount.Click();

Thread.Sleep(1500);

//Click đăng nhập (username và password trống)

//lấy element bằng name

element\_Login = driver\_74\_Hieu.FindElement(By.Name("login"));

element\_Login.Click();

Thread.Sleep(3000);

driver\_74\_Hieu.Close();

}

[TestMethod]

public void LoginFail\_TC2\_74\_Hieu()

{

driver\_74\_Hieu.Navigate();

// Click vào MyAccount

element\_MyAccount.Click();

Thread.Sleep(1500);

//lấy element bằng XPath

element\_Username = driver\_74\_Hieu.FindElement(By.XPath("//\*[@id=\"username\"]"));

//Nhập username sai, password trống

element\_Username.SendKeys("jhgga55k\_74\_Hieu");

Thread.Sleep(1500);

//lấy element bằng name

element\_Login = driver\_74\_Hieu.FindElement(By.Name("login"));

element\_Login.Click();

Thread.Sleep(3000);

driver\_74\_Hieu.Close();

}

[TestMethod]

public void LoginFail\_TC3\_74\_Hieu()

{

driver\_74\_Hieu.Navigate();

// Click vào MyAccount

element\_MyAccount.Click();

Thread.Sleep(1500);

//lấy element bằng XPath

element\_Username = driver\_74\_Hieu.FindElement(By.XPath("//\*[@id=\"username\"]"));

//lấy element bằng id

element\_password = driver\_74\_Hieu.FindElement(By.Id("password"));

//Nhập username sai, password sai

element\_Username.SendKeys("jhgga55k\_74\_Hieu");

element\_password.SendKeys("65af5g5g5\_74\_Hieu");

Thread.Sleep(1500);

//lấy element bằng name

element\_Login = driver\_74\_Hieu.FindElement(By.Name("login"));

element\_Login.Click();

Thread.Sleep(3000);

driver\_74\_Hieu.Close();

}

[TestMethod]

public void LoginFail\_TC4\_74\_Hieu()

{

driver\_74\_Hieu.Navigate();

// Click vào MyAccount

element\_MyAccount.Click();

Thread.Sleep(1500);

//lấy element bằng XPath

element\_Username = driver\_74\_Hieu.FindElement(By.XPath("//\*[@id=\"username\"]"));

//lấy element bằng id

element\_password = driver\_74\_Hieu.FindElement(By.Id("password"));

//Nhập username đúng, password sai

element\_Username.SendKeys("2151053018hieu");

element\_password.SendKeys("65af5g5g5\_74\_Hieu");

Thread.Sleep(1500);

//lấy element bằng name

element\_Login = driver\_74\_Hieu.FindElement(By.Name("login"));

element\_Login.Click();

Thread.Sleep(3000);

driver\_74\_Hieu.Close();

}

[TestMethod]

public void TC5\_Success\_74\_Hieu()

{

driver\_74\_Hieu.Navigate();

// Click vào MyAccount

element\_MyAccount.Click();

Thread.Sleep(1500);

//lấy element bằng XPath

element\_Username = driver\_74\_Hieu.FindElement(By.XPath("//\*[@id=\"username\"]"));

//lấy element bằng id

element\_password = driver\_74\_Hieu.FindElement(By.Id("password"));

//Nhập username đúng, password đúng

element\_Username.SendKeys("2151053018hieu");

element\_password.SendKeys("H!eu123456#");

Thread.Sleep(1500);

//lấy element bằng name

element\_Login = driver\_74\_Hieu.FindElement(By.Name("login"));

element\_Login.Click();

Thread.Sleep(3000);

driver\_74\_Hieu.Close();

}

Chạy bằng cách Nhấn tổ hợp phím Ctrl + R, T

Để cho chỉ chạy TestCase trong file này mà thôi

Kết quả



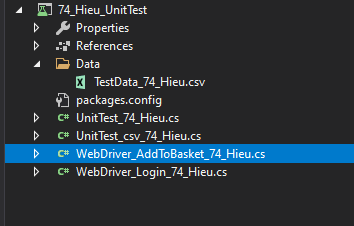
Cập nhật TestCase

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Testcase  Description | Testcase  Procedure | Expected  Output | Test date | Result | Note |
| LoginFail\_TC1\_74\_Hieu | Kiểm tra đăng nhập thất bại | Username trống, password trống | M1: Username is required. | 15/05/2024 | Pass |  |
| LoginFail\_TC2\_74\_Hieu | Username sai, password trống | M2: Password is required. | 15/05/2024 | Pass |  |
| LoginFail\_TC3\_74\_Hieu | Username sai, password sai | M3: The username is not registered on this site. If you are unsure of your username, try your email address instead. | 15/05/2024 | Pass |  |
| LoginFail\_TC4\_74\_Hieu | Username đúng = **2151053018hieu**, password sai | M4: The password you entered for the username is incorrect. | 15/05/2024 | Pass |  |
| TC5\_Success\_74\_Hieu | Kiểm tra đăng nhập thành công | Username đúng = **2151053018hieu**, password đúng = **H!eu123456#** | Chuyển đến trang tài khoản đã đăng nhập | 15/05/2024 | Pass |  |

## 3. Chức năng thêm sản phẩm vào giỏ hàng

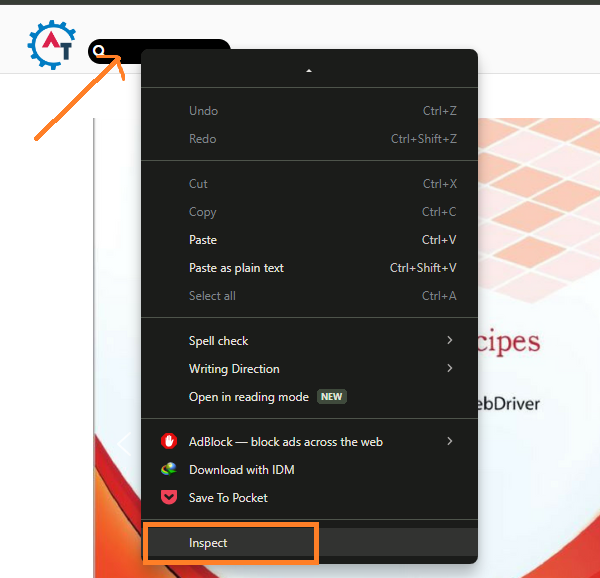
### Chuẩn bị

Bước 1: Tạo thêm 1 file UnitTest, tên là **WebDriver\_AddToBasket\_74\_Hieu.cs**



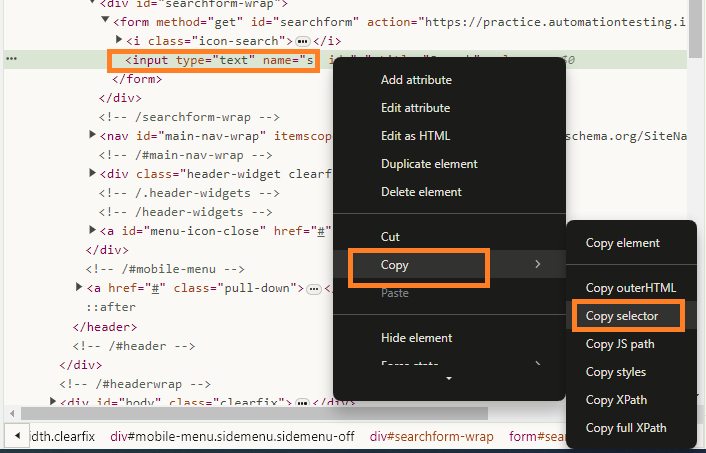
Bước 2: Lấy element **Search**

Right Click vào ô tìm kiếm, chọn **Inspect**



Right click vào thẻ 🡪 **Copy** 🡪 **Copy selector**

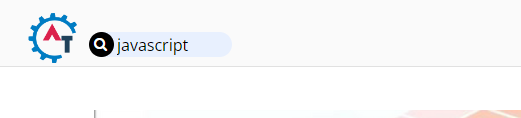
Minh họa lấy bằng Css Selector



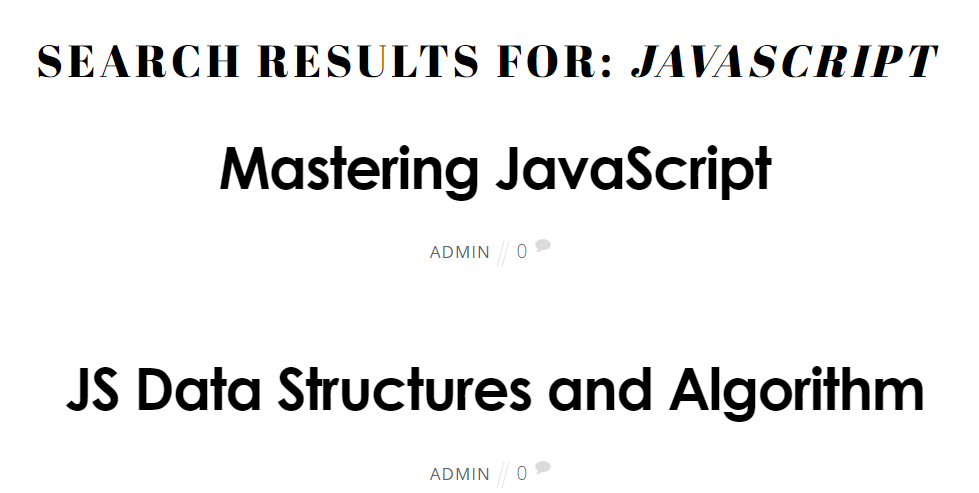
Ta được Selector: **#s**

Bước 3: Lấy element 1 kết quả tìm kiếm

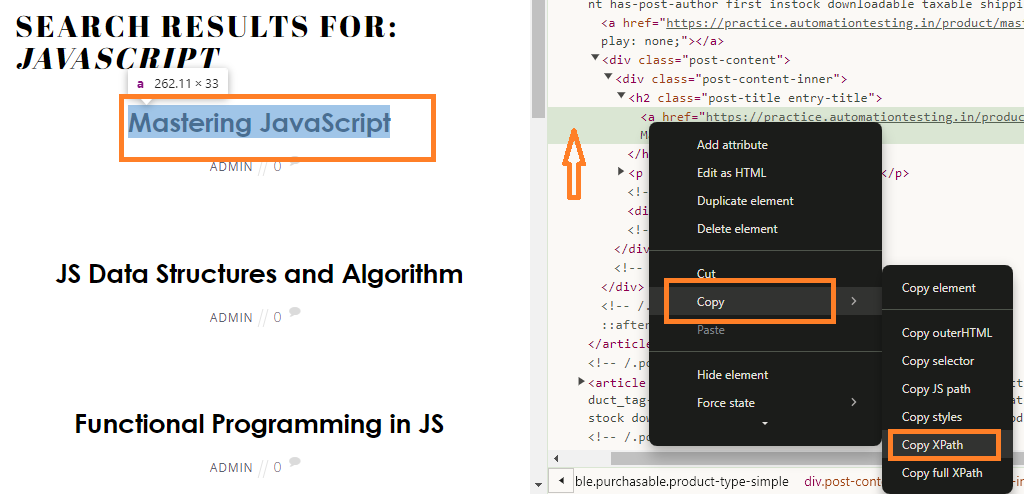
Trước tiên ta hãy nhập tìm kiếm từ khóa ‘javascript’ và enter



Ta được kết quả



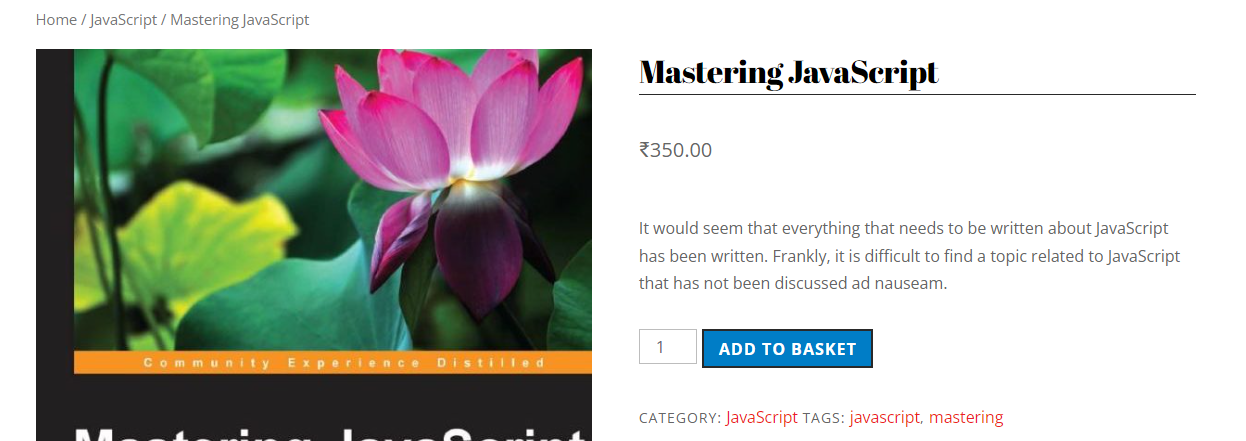
Lấy element ‘Mastering JavaScript’ tương tự như trên, em lấy bằng Xpath



Ta được: //\*[@id="post-165"]/div/div/h2/a

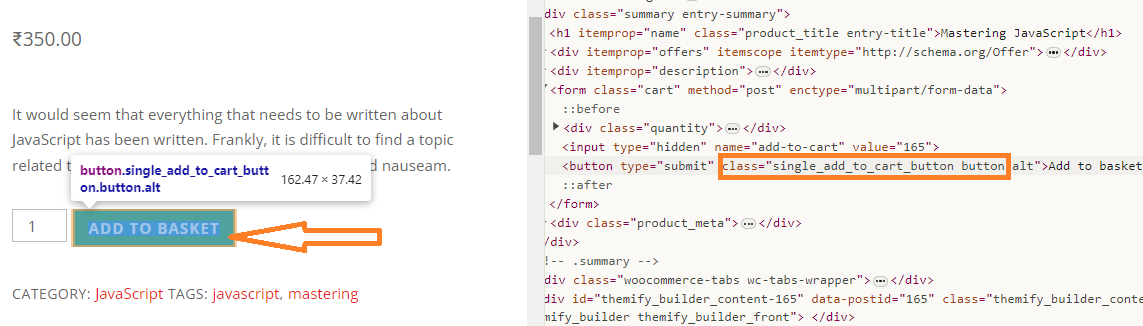
Bước 4: Lấy element button Add to Basket

Trước tiên phải click vào ‘Mastering JavaScript’, ta được giao diện



Tương tự, ta lấy element button ADD TO BASKET

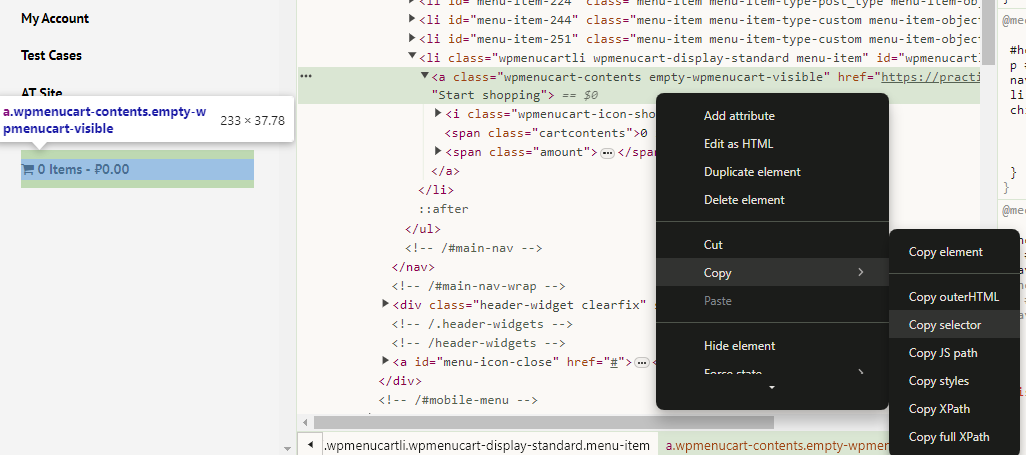
Minh họa lấy bằng **Class name**



Ta được class name: single\_add\_to\_cart\_button

Bước 5: Lấy element giỏ hàng

Tương tự như trên, lấy bằng CssSelector



Ta được Selector: **#wpmenucartli > a**

### Thực hiện

Code của file **WebDriver\_AddToBasket\_74\_Hieu.cs**

[TestMethod]

public void AddToBasket\_74\_Hieu()

{

IWebDriver driver\_74\_Hieu = new ChromeDriver();

driver\_74\_Hieu.Navigate().GoToUrl("https://practice.automationtesting.in/");

//lấy bằng Css Selector

IWebElement Search = driver\_74\_Hieu.FindElement(By.CssSelector("#s"));

// nhập từ khóa tìm kiếm

Search.SendKeys("javascript");

Thread.Sleep(2000);

// enter

Search.SendKeys(Keys.Enter);

Thread.Sleep(3000);

// element kết quả tìm kiếm

driver\_74\_Hieu.FindElement(By.XPath("//\*[@id=\"post-165\"]/div/div/h2/a")).Click();

Thread.Sleep(3000);

// thêm vào giỏ hàng (lấy bằng classname)

driver\_74\_Hieu.FindElement(By.ClassName("single\_add\_to\_cart\_button")).Click();

Thread.Sleep(3000);

//xem giỏ hàng

driver\_74\_Hieu.FindElement(By.CssSelector("#wpmenucartli > a")).Click();

Thread.Sleep(3000);

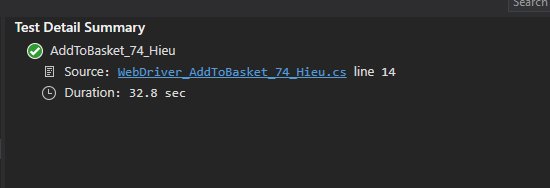
driver\_74\_Hieu.Close();

}

Chạy TestCase bằng cách nhấn tổ hợp phím Ctrl + R, T

Mục đích để chỉ run test trong file hiện tại mà thôi.

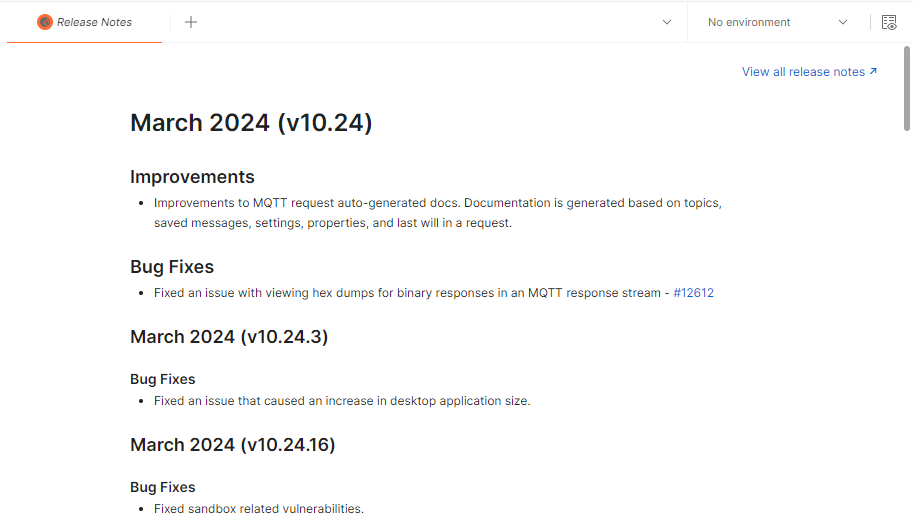
Kết quả:



# **CHƯƠNG 4. POSTMAN API TESTING**

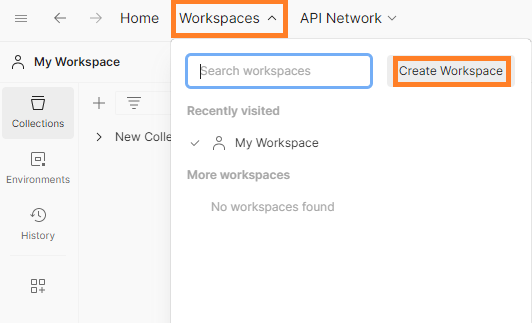
## 1. Tạo Workspace và Collection

Trong phần này, em sử dụng Postman phiên bản 10.24

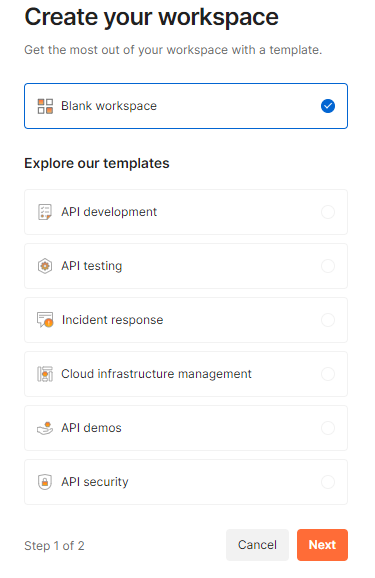


Chú ý: Để tạo được Workspace, trước tiên cần phải đăng nhập Postman.

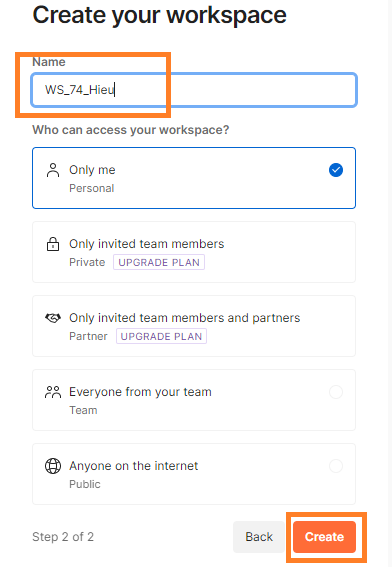
Trên thanh công cụ, chọn **Workspaces** 🡪 **Create Workspace**



Chọn **Next**

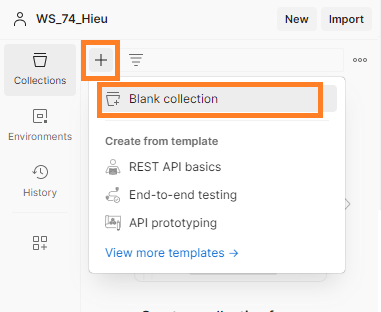


Đặt tên là WS\_74\_Hieu

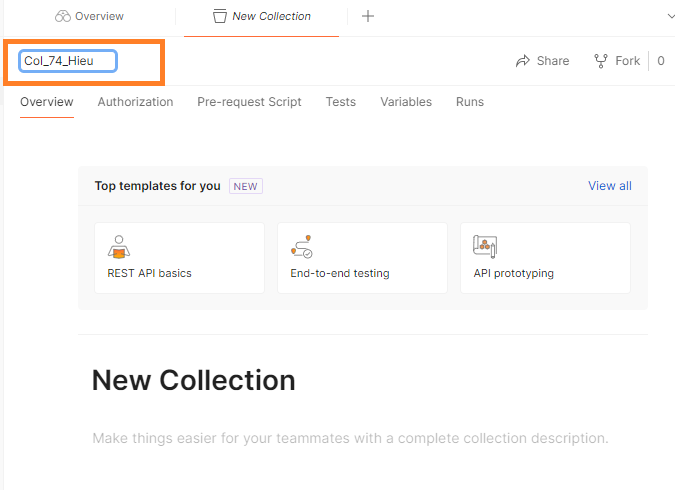


Sau đó tạo một collection, click vào biểu tượng +

Và chọn **Blank collection**

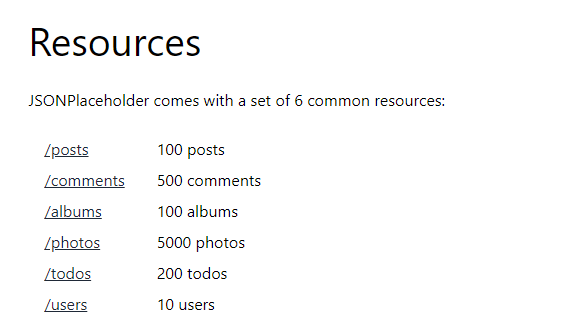


Đặt tên là Col\_74\_Hieu



## 2. Test các phương thức và viết script

Em sẽ sử dụng trang <https://jsonplaceholder.typicode.com/>



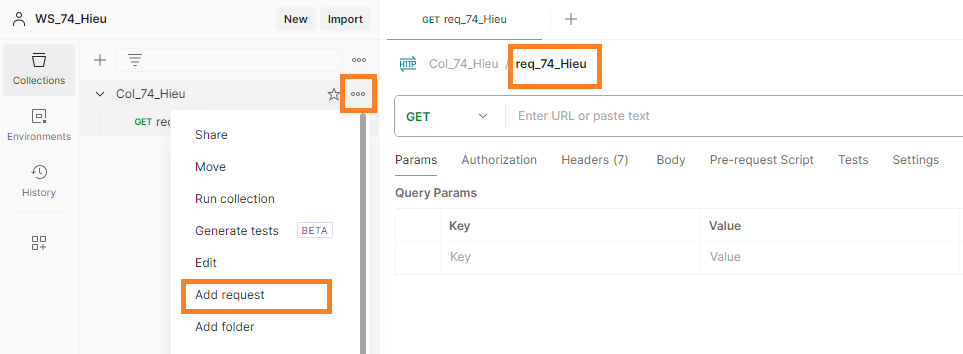
Em sử dụng Resource /posts

### Phương thức GET

API: /posts/{id}

Ví dụ: <https://jsonplaceholder.typicode.com/posts/1>

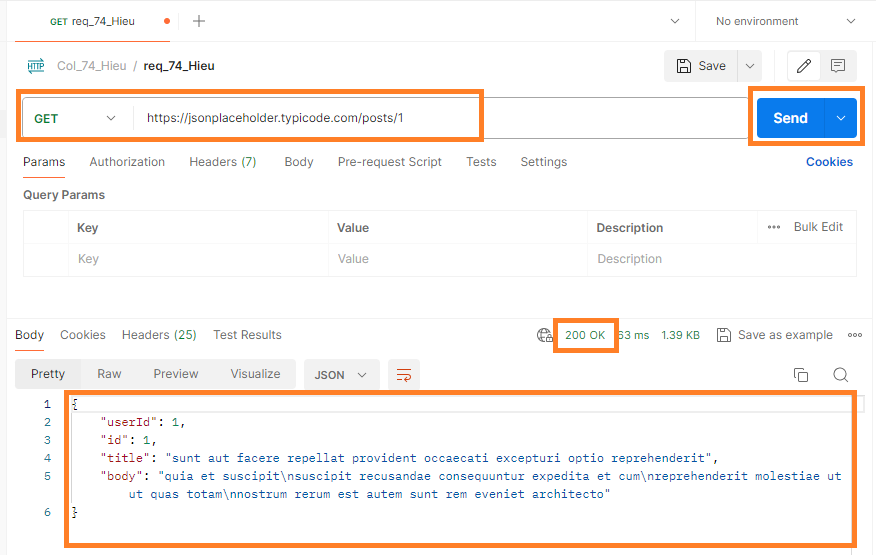
Tạo request có tên **req\_74\_Hieu**



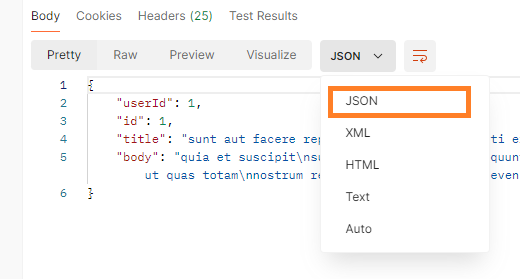
Dán URL ở trên, sau đó chọn phương thức GET 🡪 **Send**

Được trả về Reponse với status code 200 là thành công

Với dữ liệu đã được Serialize như hình



Lưu ý: phải chọn định dạng là JSON



Viết script để test

pm.test("74\_Hieu\_status200", **function** () {

  pm.response.to.have.status(200);

});

//kiểm tra response body có các thuộc tính userId, id, title và body không

pm.test("74\_Hieu", **function** () {

**var** responseJSON = pm.response.json();

    pm.expect(responseJSON).to.have.property('userId');

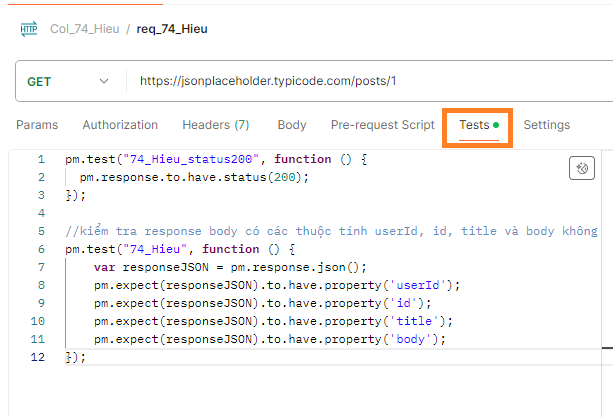
    pm.expect(responseJSON).to.have.property('id');

    pm.expect(responseJSON).to.have.property('title');

    pm.expect(responseJSON).to.have.property('body');

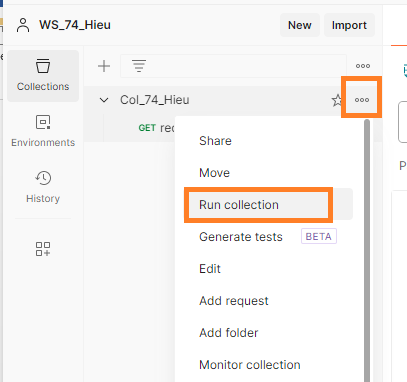
});

Dán code vào mục Tests

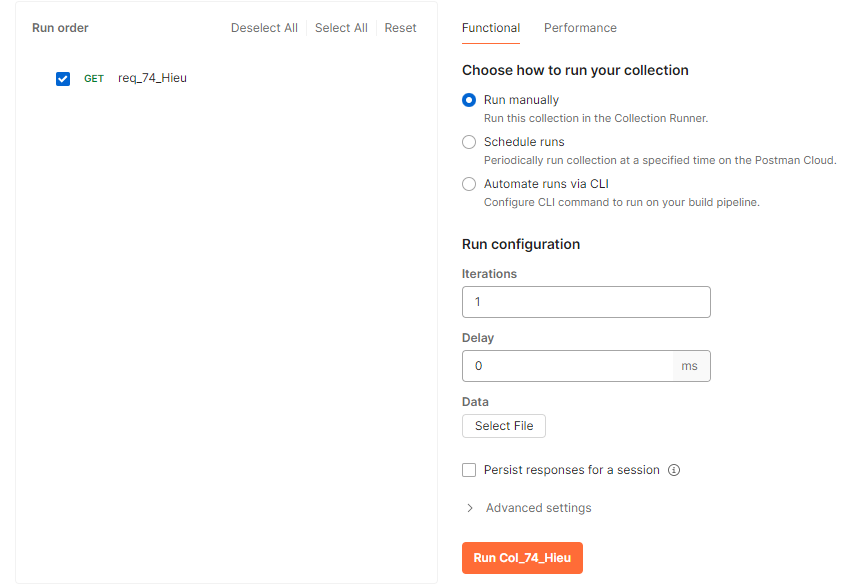


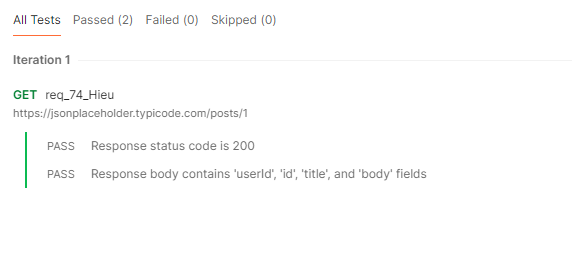
Sau đó lưu lại Ctrl + S

Để chạy Test, ta click biểu tượng 3 chấm bên cạnh Collection, và chọn **Run collection**



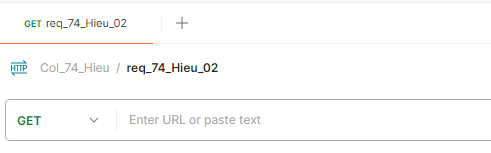
Chọn Run Col\_74\_Hieu



Kết quả 

### Phương thức POST

Tạo 1 request mới (cách làm bên trên), và đặt tên là **req\_74\_Hieu\_02**



**API:** <https://jsonplaceholder.typicode.com/users/>

Ta tạo một user mới bằng phương thức POST như sau:

Ta có Json của object User

{

    "name": "Duong Hieu",

    "username": "hieu74",

    "email": "hieu74@gmail.com",

    "address": {

      "street": "Kulas Light",

      "suite": "Apt. 556",

      "city": "Gwenborough",

      "zipcode": "92998-3874",

      "geo": {

        "lat": "-37.3159",

        "lng": "81.1496"

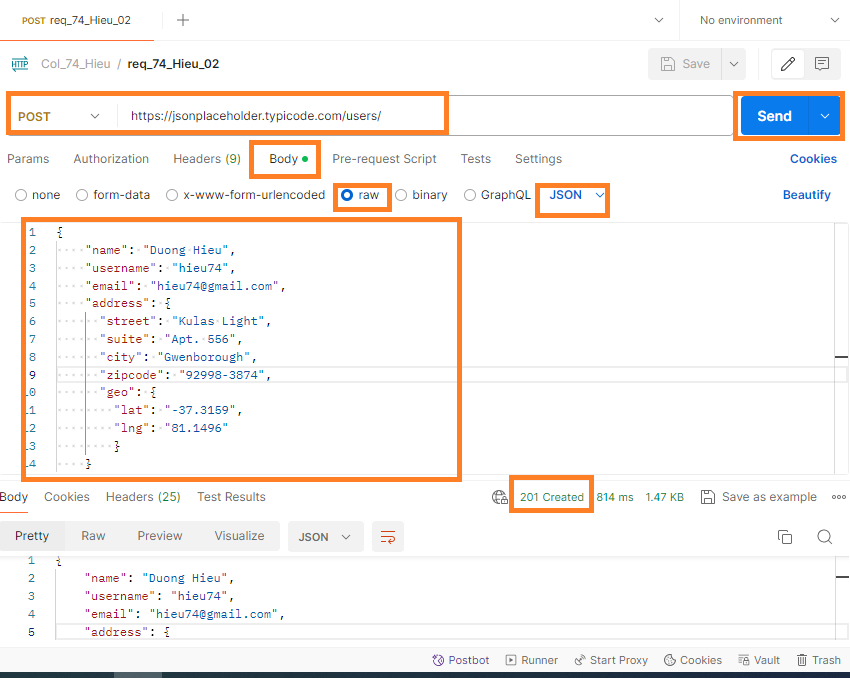
        }

    }

}

Dán url và chọn phương thức **POST**, sau đó qua mục **Body**, chọn **raw**, chọn kiểu dữ liệu **JSON**. Cuối cùng dán cung cấp thông tin user và nhấn **Send**

Ta thấy status là 201 là thành công



Viết script test: (Tương tự như trên)

pm.test("Created\_74\_Hieu", **function** () {

  pm.response.to.have.status(201);

});

// kiem tra thoi gian tra response < 0.3 giay

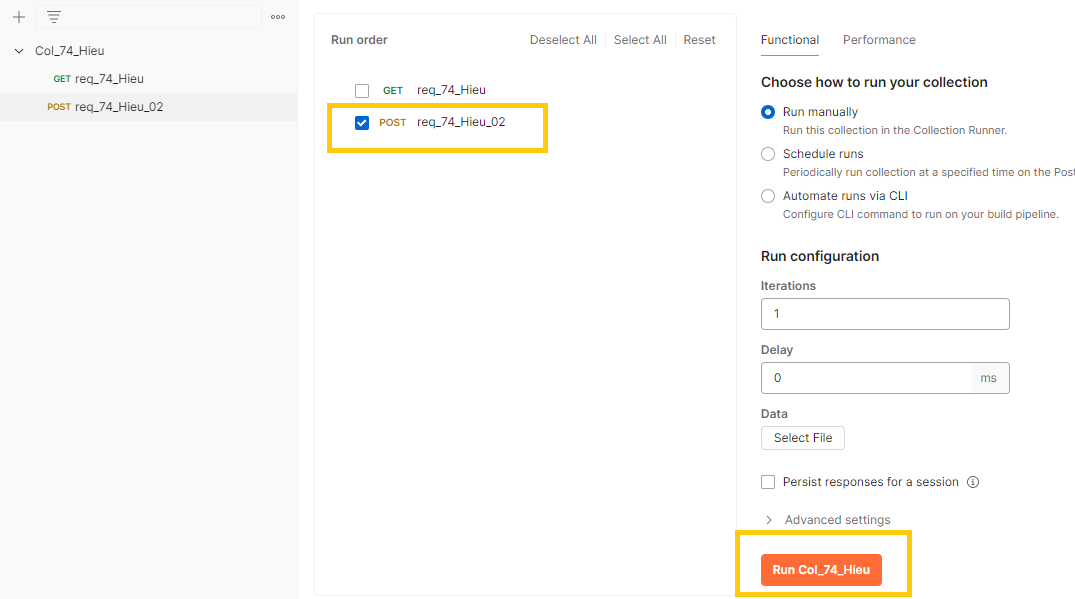
pm.test("TimeBelow0.3second\_74\_Hieu", **function** () {

  pm.expect(pm.response.responseTime).to.be.below(300);

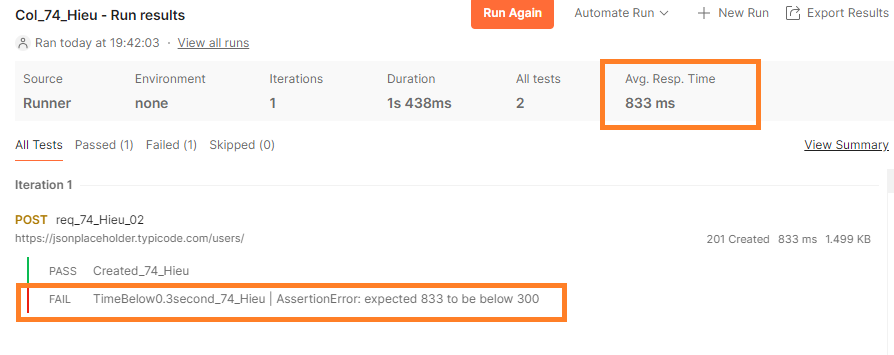
});

Lưu ý: nhớ Save trước khi Run collection

Chọn request là **req\_74\_Hieu\_02** và Run



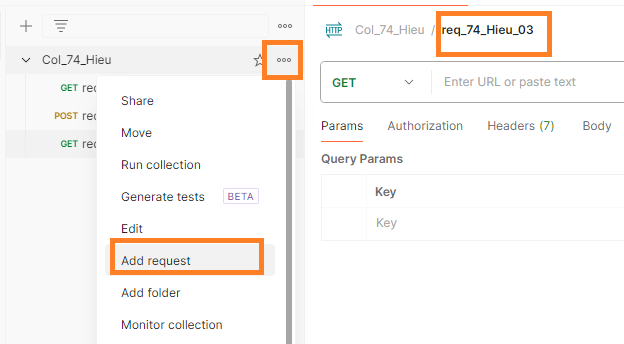
Ta được kết quả



1 test fail vì response time là 833 ms > 300 ms

### Phương thức PUT

Tương tự, thêm 1 request vào collection, và đặt tên là **req\_74\_Hieu\_03**



Ta cập nhật 1 record bằng phương thức PUT

API: /posts/1

Cụ thể ta cập nhật title và body của bài post có id là 1, với nội dung

{

    "userId": 1,

    "id": 1,

    "title": "test phuong thuc PUT - 74 Hieu",

    "body": "Day la noi dung duoc viet boi 74\_Hieu"

}

Dán URL 🡪 chọn phương thức PUT 🡪 Body 🡪 raw 🡪 JSON 🡪 Dán nội dung 🡪 Send

Ta nhận được status 200 là thành công



Viết script test:

//kiểm tra userID có phải là 1 ?

pm.test("Hieu 74", **function** () {

    pm.expect(pm.response.json().userId).to.equal(1);

});

// kiểm tra title có chứa "74 Hieu" ?

pm.test("title field containing '74 Hieu'", **function** () {

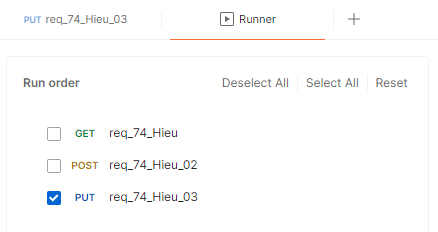
**var** response = pm.response.json();

  pm.expect(response).to.have.property('title').that.is.a('string').and.to.include('74 Hieu');

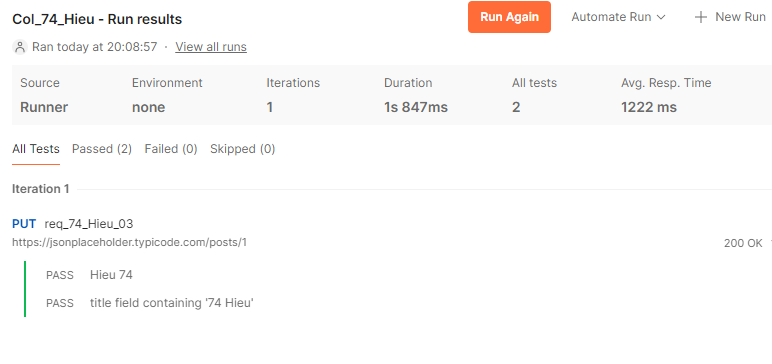
});



Run collection và chọn req\_74\_Hieu\_03



Ta được kết quả

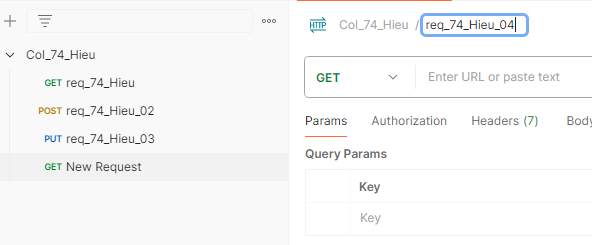


### Phương thức DELETE

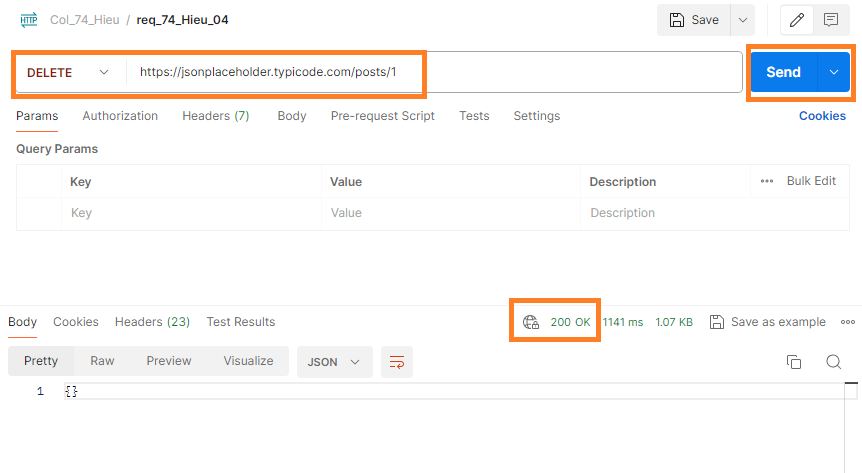
Ta xóa một post có id là 1

API: /posts/1

Add request đặt tên là **req\_74\_Hieu\_04**



Ta dán URL vào ô textbox và chọn phương thức DELETE, sau đó Send



Vì phương thức này không có body request nên sẽ không thấy body hiện dữ liệu.

Nhận được status 200 là thành công

Viết script test:

// kiểm tra xem response body tồn tại ?

pm.test("Response body exists 74 Hieu", **function** () {

    pm.expect(pm.response.json()).to.not.be.empty;

});

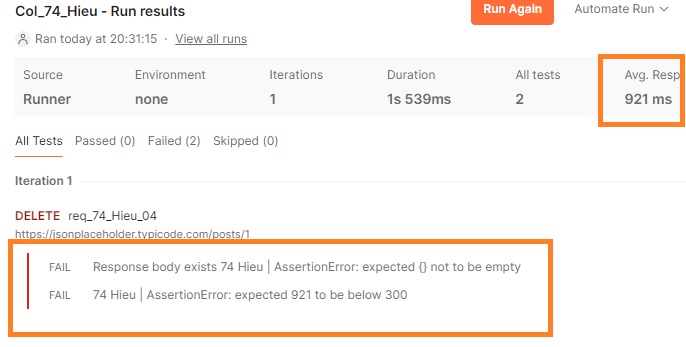
// Check if the response time is less than 300ms

pm.test("74 Hieu", **function** () {

  pm.expect(pm.response.responseTime).to.be.below(300);

});

Kết quả



Vì đây là phương thức DELETE nên sẽ không có response body, nên test này phải FAIL mới hợp lý

Và thời gian trả response trên 300 ms nên test cũng FAIL

-----------------------------------------HẾT-----------------------------------------