

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

**KHOA CÔNG NGHỆ THÔNG TIN**

**-------- --------**



**BÀI BÁO CÁO**

**MÔN: KIỂM THỬ PHẦN MỀM**

**Tên sinh viên: Ngô Thanh Lam**

**MSSV: 2151050220**

**STT: 32**

**Giảng viên hướng dẫn: Nguyễn Thị Ngọc Thanh**

**Tp.Hồ Chí Minh, 2024**

# Mục lục

# 

[**Mục lục 2**](#_heading=h.30j0zll)

[**Phần A: WINFORM 5**](#_heading=h.2et92p0)

[1. Chủ đề: Tính toán 7 hằng đẳng thức 5](#_heading=h.tyjcwt)

[2. Tạo project “NUnit\_32\_ThanhLam” 6](#_heading=h.3dy6vkm)

[3. Thiết kế giao diện 10](#_heading=h.1t3h5sf)

[4. Viết code cho tập tin Calculation\_32\_Lam.cs 14](#_heading=h.4d34og8)

[5. Add Reference 16](#_heading=h.2s8eyo1)

[6. Viết test case cho các hằng đẳng thức 17](#_heading=h.17dp8vu)

[6.1. Test case 1: tính bình phương của 1 tổng 17](#_heading=h.3rdcrjn)

[6.2. Test case 2: tính bình phương của 1 hiệu 18](#_heading=h.26in1rg)

[6.3. Test case 3: tính hiệu 2 bình phương 18](#_heading=h.lnxbz9)

[6.4. Test case 4: tính lập phương của 1 tổng 19](#_heading=h.35nkun2)

[6.5. Test case 5: tính lập phương của 1 hiệu 19](#_heading=h.1ksv4uv)

[6.6. Test case 6: tính tổng 2 lập phương 20](#_heading=h.44sinio)

[6.7. Test case 7: tính hiệu 2 lập phương 20](#_heading=h.2jxsxqh)

[6.8. Test case 8: test Data trên file csv 20](#_heading=h.z337ya)

[7. Chạy Test case 22](#_heading=h.3j2qqm3)

[**Phần B: SELENIUM WEBDRIVER 23**](#_heading=h.1y810tw)

[1. Giới thiệu website: https://whalishop.vn/ 23](#_heading=h.4i7ojhp)

[1.1. Đặc tả use case các chức năng thực hiện kiểm thử 23](#_heading=h.2xcytpi)

[a. Chức năng đăng nhập 23](#_heading=h.1ci93xb)

[b. Chức năng tìm kiếm 24](#_heading=h.i74io2ew41iu)

[c. Chức năng thêm sản phẩm vào giỏ hàng 25](#_heading=h.ir4hwq55dst)

[2. Các thành phần HTML 25](#_heading=h.1pxezwc)

[3. Thiết kế kiểm thử 27](#_heading=h.49x2ik5)

[3.1. Chức năng 1: Đăng nhập 27](#_heading=h.2p2csry)

[3.2. Chức năng 2: Tìm kiếm sản phẩm 28](#_heading=h.3o7alnk)

[3.3. Chức năng 3: Thêm sản phẩm vào giỏ hàng 29](#_heading=h.32hioqz)

[4. Viết các test case với Selenium WebDriver 30](#_heading=h.2grqrue)

[4.1. Tạo project: KTPM\_32\_Lam\_SeleniumWebdriver 30](#_heading=h.vx1227)

[4.2. Cài đặt package cho project 30](#_heading=h.3fwokq0)

[4.3. Tạo file riêng tương tác với các thành phần html 32](#_heading=h.1v1yuxt)

[a. Tương tác với trang đăng nhập: https://whalishop.vn/account/login 32](#_heading=h.4f1mdlm)

[b. Tương tác với trang chủ để tìm kiếm: 34](#_heading=h.2u6wntf)

[c. Tương tác với trang 1 sản phẩm bất kì để thêm vào giỏ hàng: chức năng này chúng ta vào 1 trang sản phẩm bất kì. Ví dụ: https://whalishop.vn/products/ao-thun-new-era-vien-den 35](#_heading=h.19c6y18)

[4.4. Chức năng 1: Đăng nhập 38](#_heading=h.3tbugp1)

[a. Test case 1: Nhập đúng email và password 39](#_heading=h.nmf14n)

[b. Test case 2: Nhập email nhưng thiếu password 39](#_heading=h.37m2jsg)

[c. Test case 3: Nhập password nhưng thiếu email 40](#_heading=h.1mrcu09)

[d. Test case 4: Không nhập email và password 41](#_heading=h.111kx3o)

[e. Test case 5: Nhập sai email (email chưa đăng ký) 41](#_heading=h.4k668n3)

[f. Test case 6: Nhập sai password 42](#_heading=h.1egqt2p)

[g. Test case 7: Nhập email không hợp lệ (email thiếu @) 43](#_heading=h.3ygebqi)

[4.5. Chức năng 2: Tìm kiếm sản phẩm 43](#_heading=h.2dlolyb)

[a. Test case 1: Nhập tìm kiếm 44](#_heading=h.sqyw64)

[b. Test case 2: Không nhập tìm kiếm 45](#_heading=h.3cqmetx)

[4.6. Chức năng 3: Thêm sản phẩm vào giỏ hàng 46](#_heading=h.1rvwp1q)

[a. Test case 1: thêm số lượng là 1 47](#_heading=h.4bvk7pj)

[b. Test case 2: thêm số lượng là 0 47](#_heading=h.2r0uhxc)

[c. Test case 3: thêm số lượng lớn hơn số lượng trong kho 48](#_heading=h.1664s55)

[d. Test case 4: thêm 1 sản phẩm vào giỏ hàng sau đó thêm số lượng là 0 49](#_heading=h.blyc81fkcpzn)

[e. Test case 5: thêm 1 sản phẩm vào giỏ hàng sau đó thêm số lượng là 1 50](#_heading=h.za4ft9lgurud)

[f. Test case 6: thêm 1 sản phẩm vào giỏ hàng sau đó thêm số lượng lớn hơn số lượng trong kho 51](#_heading=h.5sz731ypt4bs)

[5. Thực thi kiểm thử 52](#_heading=h.3q5sasy)

[**Phần C: API POST 53**](#_heading=h.25b2l0r)

[1. Postman. 53](#_heading=h.kgcv8k)

[2. Tạo Workspace có tên KTPM\_32\_Lam 54](#_heading=h.34g0dwd)

[3. Lấy quyền website để test API 56](#_heading=h.1jlao46)

[4. Thiết lập biến môi trường trong postman 58](#_heading=h.43ky6rz)

[5. Thực hiện test API 58](#_heading=h.2iq8gzs)

[5.1. Chức năng 1: Tạo một collection có tên là Customer\_32\_Lam 58](#_heading=h.xvir7l)

[5.1.1. Tạo request: GET GetCustomers\_32\_Lam 59](#_heading=h.3hv69ve)

[5.1.1.1. Test case 1: 59](#_heading=h.1x0gk37)

[5.1.1.2. Test case 2: 59](#_heading=h.4h042r0)

[5.1.2. Tạo request: POST AddCustomer\_32\_Lam 60](#_heading=h.2w5ecyt)

[5.1.2.1. Test case 1: 61](#_heading=h.1baon6m)

[5.1.2.2. Test case 2: 61](#_heading=h.3vac5uf)

[5.1.3. Tạo request: PUT UpdateCustomer\_32\_Lam 62](#_heading=h.pkwqa1)

[5.1.3.1. Test case 1: 63](#_heading=h.39kk8xu)

[5.1.3.2. Test case 2: 63](#_heading=h.1opuj5n)

[5.1.4. Tạo request: DELETE DeleteCustomer\_32\_Lam 64](#_heading=h.2nusc19)

[5.1.4.1. Test case 1: 64](#_heading=h.1302m92)

[5.2. Chức năng 2: Tạo một collection có tên là Products\_32\_Lam 65](#_heading=h.3mzq4wv)

[5.2.1. Tạo request: GET GetProducts\_32\_Lam 65](#_heading=h.2250f4o)

[5.2.1.1. Test case 1: 65](#_heading=h.haapch)

[5.2.1.2. Test case 2: 66](#_heading=h.319y80a)

[5.2.2. Tạo request: POST AddProductImage\_32\_Lam 67](#_heading=h.1gf8i83)

[5.2.2.1. Test case 1: 68](#_heading=h.40ew0vw)

[5.2.2.2. Test case 2: 68](#_heading=h.2fk6b3p)

[5.2.3. Tạo request: PUT UpdateProduct\_32\_Lam 69](#_heading=h.3ep43zb)

[5.2.3.1. Test case 1: 70](#_heading=h.1tuee74)

[5.2.3.2. Test case 2: 70](#_heading=h.4du1wux)

[5.2.4. Tạo request: DELETE DeleteProduct\_32\_Lam 71](#_heading=h.184mhaj)

[5.2.4.1. Test case 1: 71](#_heading=h.3s49zyc)

# 

# Phần A: WINFORM

## Chủ đề: Tính toán 7 hằng đẳng thức

7 hằng đẳng thức này bao gồm: bình phương của một tổng, bình phương của một hiệu, hiệu của hai bình phương, lập phương của một tổng, lập phương của một hiệu, tổng hai lập phương và cuối cùng là hiệu hai lập phương.

* Bình phương của một tổng

(a + b)² = a² + 2ab + b²

Bình phương của một tổng bằng với bình phương của số thứ nhất cộng với hai lần tích của số thứ nhất nhân với số thứ hai, cộng với bình phương số thứ hai.

* Bình phương của một hiệu

(a – b)² = a² – 2ab + b²

Bình phương một hiệu bằng với bình phương số thứ nhất trừ đi hai lần tích của số thứ nhất nhân số thứ hai sau đó cộng bình phương với số thứ hai.

* Hiệu hai bình phương

a² – b² = (a – b)(a + b)

Hiệu hai bình phương hai số sẽ bằng với tổng hai số đó nhân với hiệu hai số đó.

* Lập phương của một tổng

(a + b)³ = A³ + 3a²b + 3ab² + b³

Lập phương của một tổng hai số bằng với lập phương của số thứ nhất cộng với ba lần tích bình phương số thứ nhất nhân số thứ hai cộng với ba lần tích số thứ nhất nhân với bình phương số thứ hai cộng với lập phương số thứ hai.

* Lập phương của một hiệu

(a – b)³ = a³ – 3a²b + 3ab² – b³

Lập phương của một hiệu hai số sẽ bằng với lập phương của số thứ nhất trừ đi ba lần tích bình phương của số thứ nhất nhân với số thứ hai cộng với ba lần tích số thứ nhất nhân với bình phương số thứ hai trừ đi lập phương số thứ hai

* Tổng hai lập phương

a³ + b³ = (a + b)(a² – ab + b² )

Tổng hai lập phương hai số bằng tổng của hai số đó nhân với bình phương thiếu của hiệu hai số đó

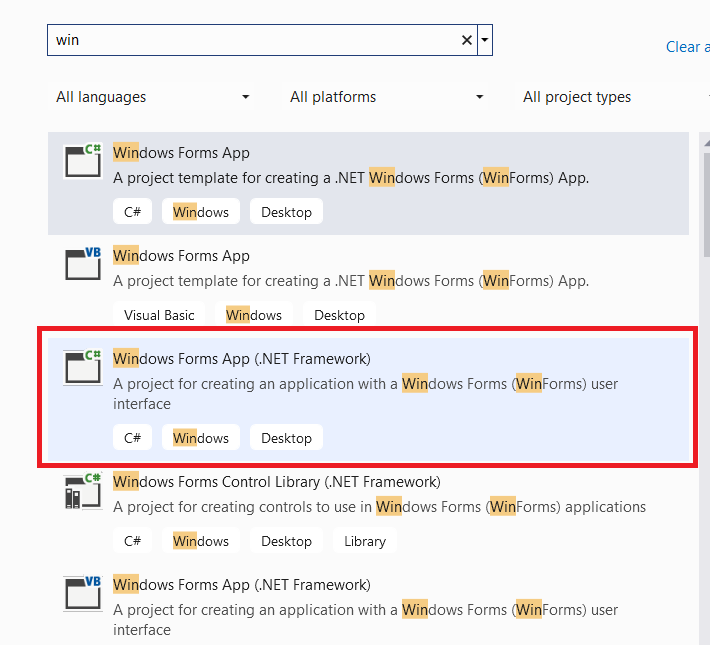
* Hiệu hai lập phương

a³ – b³ = (a – b)(a² + ab + b²)

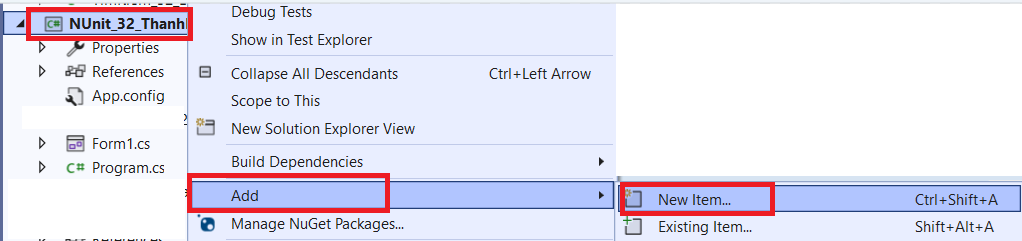
Hiệu hai lập phương của hai số bằng hiệu hai số đó nhân với bình phương thiếu của tổng của hai số đó.

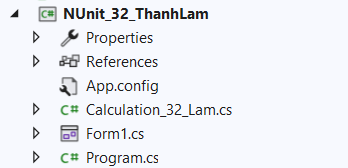
## Tạo project “NUnit\_32\_ThanhLam”

* Tạo 1 solution riêng cho project
* Tạo 2 project khác nhau
  + **Project “NUnit\_32\_ThanhLam”** chứa giao diện và 1 tập tin **Calculation\_32\_Lam.cs** chứa lớp public Calculation\_32\_Lam có phương thức Excute\_32\_Lam để thực hiện tính toán các công thức hằng đẳng thức.

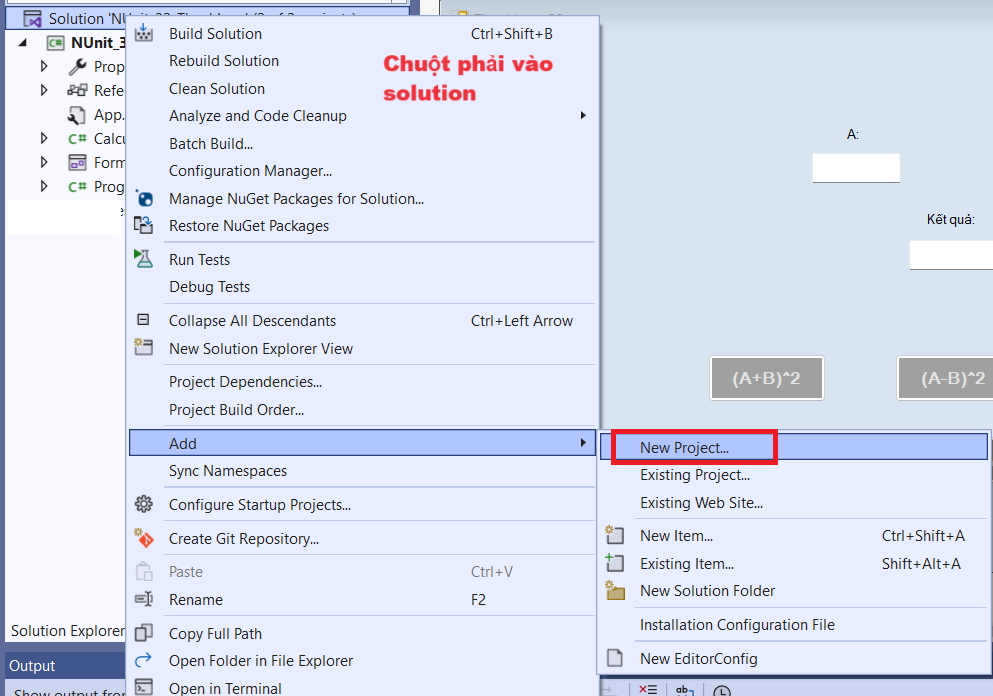


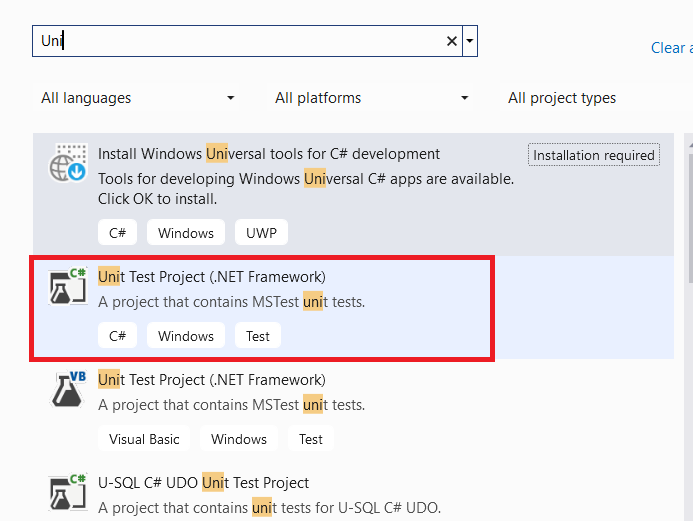
* + - Tạo file **Calculation\_32\_Lam.cs:** chuột phải project **NUnit\_32\_ThanhLam** => Add => New Item

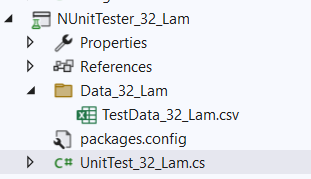
****



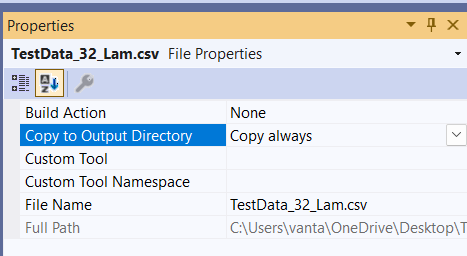
* + **Project kiểm thử “NUnitTester\_32\_Lam”** để kiểm thử các phép toán trong chương trình trên.



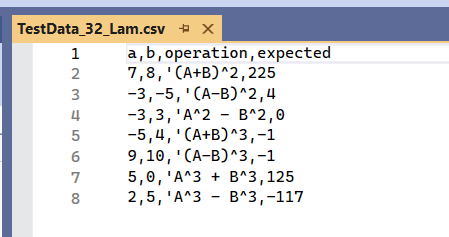




* Tạo thêm thư mục Data\_32\_Lam để chứa file TestData\_32\_Lam.csv
  + Click chuột phải vào thư mục Data\_32\_Lam => **New Item** để tạo file csv
  + Sau đó click chuột phải vào file csv. => Properties => thiết lập thuộc tính **Copy to Output Directory**: Copy always

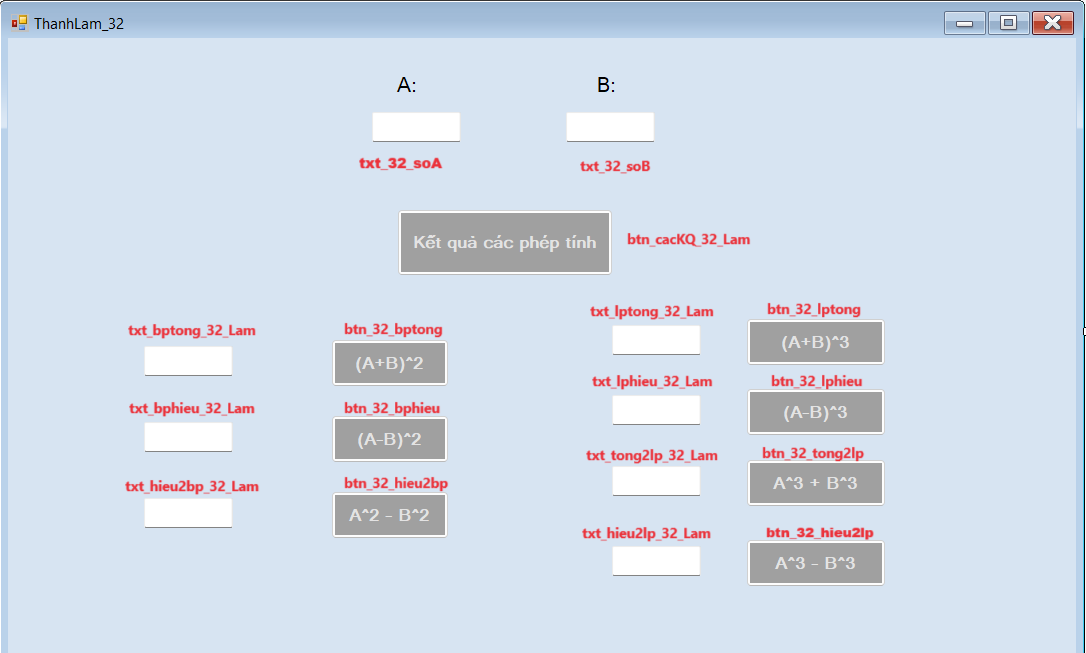


* + Nội dung trong file csv:

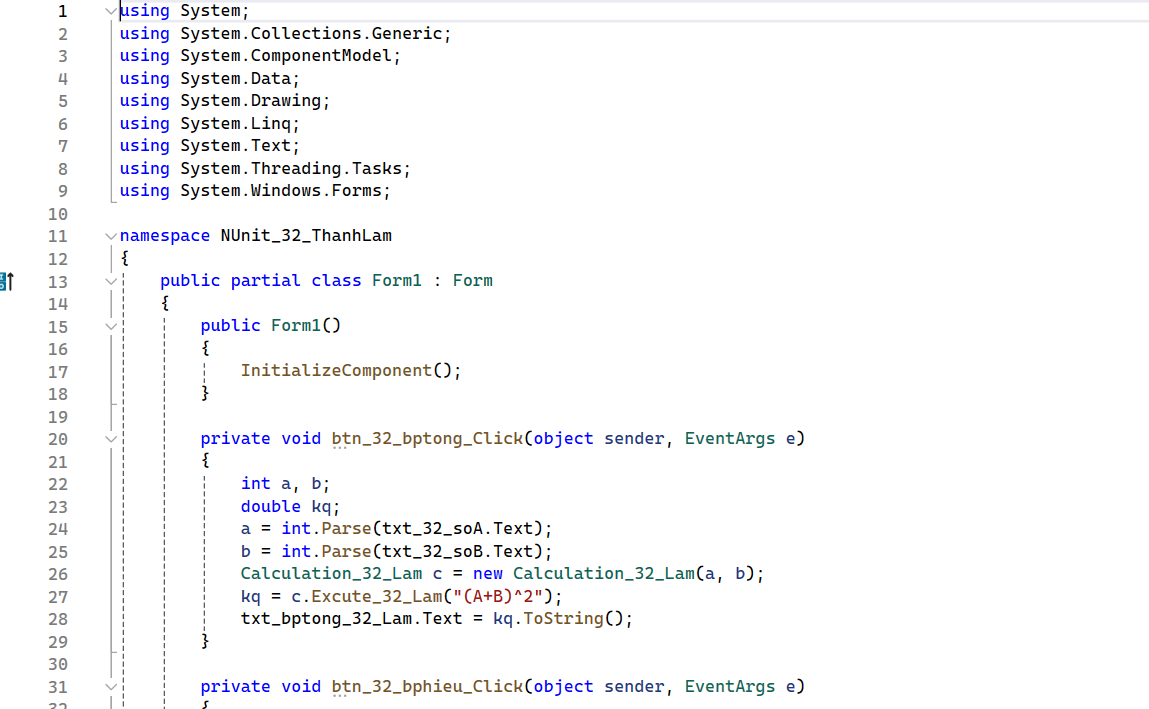


## Thiết kế giao diện

* Thiết kế giao diện Form1.cs trong project **NUnit\_32\_ThanhLam**



* Code Window Form:



| 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 NUnit\_32\_ThanhLam  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  private void btn\_32\_bptong\_Click(object sender, EventArgs e)  {  int a, b;  double kq;  a = int.Parse(txt\_32\_soA.Text);  b = int.Parse(txt\_32\_soB.Text);  Calculation\_32\_Lam c = new Calculation\_32\_Lam(a, b);  kq = c.Excute\_32\_Lam("(A+B)^2");  txt\_bptong\_32\_Lam.Text = kq.ToString();  }  private void btn\_32\_bphieu\_Click(object sender, EventArgs e)  {  int a, b;  double kq;  a = int.Parse(txt\_32\_soA.Text);  b = int.Parse(txt\_32\_soB.Text);  Calculation\_32\_Lam c = new Calculation\_32\_Lam(a, b);  kq = c.Excute\_32\_Lam("(A-B)^2");  txt\_bphieu\_32\_Lam.Text = kq.ToString();  }  private void btn\_32\_hieu2bp\_Click(object sender, EventArgs e)  {  int a, b;  double kq;  a = int.Parse(txt\_32\_soA.Text);  b = int.Parse(txt\_32\_soB.Text);  Calculation\_32\_Lam c = new Calculation\_32\_Lam(a, b);  kq = c.Excute\_32\_Lam("A^2 - B^2");  txt\_hieu2bp\_32\_Lam.Text = kq.ToString();  }  private void btn\_32\_lptong\_Click(object sender, EventArgs e)  {  int a, b;  double kq;  a = int.Parse(txt\_32\_soA.Text);  b = int.Parse(txt\_32\_soB.Text);  Calculation\_32\_Lam c = new Calculation\_32\_Lam(a, b);  kq = c.Excute\_32\_Lam("(A+B)^3");  txt\_lptong\_32\_Lam.Text = kq.ToString();  }  private void btn\_32\_lphieu\_Click(object sender, EventArgs e)  {  int a, b;  double kq;  a = int.Parse(txt\_32\_soA.Text);  b = int.Parse(txt\_32\_soB.Text);  Calculation\_32\_Lam c = new Calculation\_32\_Lam(a, b);  kq = c.Excute\_32\_Lam("(A-B)^3");  txt\_lphieu\_32\_Lam.Text = kq.ToString();  }    private void btn\_32\_tong2lp\_Click(object sender, EventArgs e)  {  int a, b;  double kq;  a = int.Parse(txt\_32\_soA.Text);  b = int.Parse(txt\_32\_soB.Text);  Calculation\_32\_Lam c = new Calculation\_32\_Lam(a, b);  kq = c.Excute\_32\_Lam("A^3 + B^3");  txt\_tong2lp\_32\_Lam.Text = kq.ToString();    }  private void btn\_32\_hieu2lp\_Click(object sender, EventArgs e)  {  int a, b;  double kq;  a = int.Parse(txt\_32\_soA.Text);  b = int.Parse(txt\_32\_soB.Text);  Calculation\_32\_Lam c = new Calculation\_32\_Lam(a, b);  kq = c.Excute\_32\_Lam("A^3 - B^3");  txt\_hieu2lp\_32\_Lam.Text = kq.ToString();  }  private void btn\_cacKQ\_32\_Lam\_Click(object sender, EventArgs e)  {  int a, b;  a = int.Parse(txt\_32\_soA.Text);  b = int.Parse(txt\_32\_soB.Text);  Calculation\_32\_Lam c = new Calculation\_32\_Lam(a, b);  //Bình phương tổng  double kq\_bptong\_32\_Lam;  kq\_bptong\_32\_Lam = c.Excute\_32\_Lam("(A+B)^2");  txt\_bptong\_32\_Lam.Text = kq\_bptong\_32\_Lam.ToString();  //Bình phương hiệu  double kq\_bphieu\_32\_Lam;  kq\_bphieu\_32\_Lam = c.Excute\_32\_Lam("(A-B)^2");  txt\_bphieu\_32\_Lam.Text = kq\_bphieu\_32\_Lam.ToString();  //Hiệu 2 bình phương  double kq\_hieu2bp\_32\_Lam;  kq\_hieu2bp\_32\_Lam = c.Excute\_32\_Lam("A^2 - B^2");  txt\_hieu2bp\_32\_Lam.Text = kq\_hieu2bp\_32\_Lam.ToString();  //Lập phương tổng  double kq\_lptong\_32\_Lam;  kq\_lptong\_32\_Lam = c.Excute\_32\_Lam("(A+B)^3");  txt\_lptong\_32\_Lam.Text = kq\_lptong\_32\_Lam.ToString();  //Lập phương hiệu  double kq\_lphieu\_32\_Lam;  kq\_lphieu\_32\_Lam = c.Excute\_32\_Lam("(A-B)^3");  txt\_lphieu\_32\_Lam.Text = kq\_lphieu\_32\_Lam.ToString();  //Tổng 2 lập phương  double kq\_tong2lp\_32\_Lam;  kq\_tong2lp\_32\_Lam = c.Excute\_32\_Lam("A^3 + B^3");  txt\_tong2lp\_32\_Lam.Text = kq\_tong2lp\_32\_Lam.ToString();  //Hiệu 2 lập phương  double kq\_hieu2lp\_32\_Lam;  kq\_hieu2lp\_32\_Lam = c.Excute\_32\_Lam("A^3 - B^3");  txt\_hieu2lp\_32\_Lam.Text = kq\_hieu2lp\_32\_Lam.ToString();  }    }  } |
| --- |

## Viết code cho tập tin Calculation\_32\_Lam.cs

* File thực hiện tính toán các công thức hằng đẳng thức.

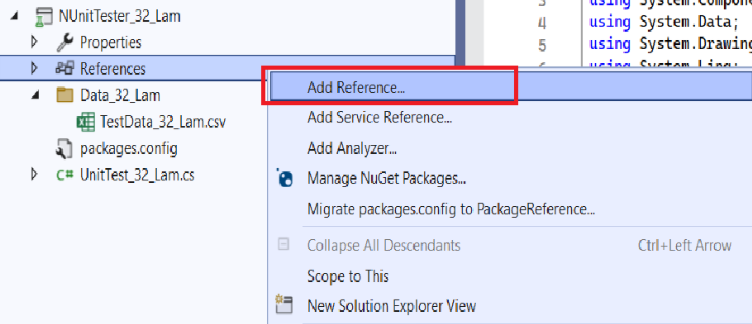
A screenshot of a computer

Description automatically generated

| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace NUnit\_32\_ThanhLam  {    public class Calculation\_32\_Lam  {  private int a, b;  public Calculation\_32\_Lam(int a, int b)  {  this.a = a;  this.b = b;  }  public double Excute\_32\_Lam(string Cal)  {  double result32 = 0;  switch (Cal)  {  case "(A+B)^2":  result32 = Math.Pow(this.a, 2) + 2 \* this.a \* this.b + Math.Pow(this.b, 2);  break;  case "(A-B)^2":  result32 = Math.Pow(this.a, 2) - 2 \* this.a \* this.b + Math.Pow(this.b, 2);  break;  case "A^2 - B^2":  result32 = (this.a - this.b) \* (this.a + this.b);  break;  case "(A+B)^3":  result32 = Math.Pow(this.a, 3) + 3 \* Math.Pow(this.a, 2) \* this.b + 3 \* this.a \* Math.Pow(this.b, 2) + Math.Pow(this.b, 3);  break;  case "(A-B)^3":  result32 = Math.Pow(this.a, 3) - 3 \* Math.Pow(this.a, 2) \* this.b + 3 \* this.a \* Math.Pow(this.b, 2) - Math.Pow(this.b, 3);  break;  case "A^3 + B^3":  result32 = (this.a + this.b) \* (Math.Pow(this.a, 2) - this.a \* this.b + Math.Pow(this.b, 2));  break;  case "A^3 - B^3":  result32 = (this.a - this.b) \* (Math.Pow(this.a, 2) + this.a \* this.b + Math.Pow(this.b, 2));  break;  }  return result32;  }  }  } |
| --- |

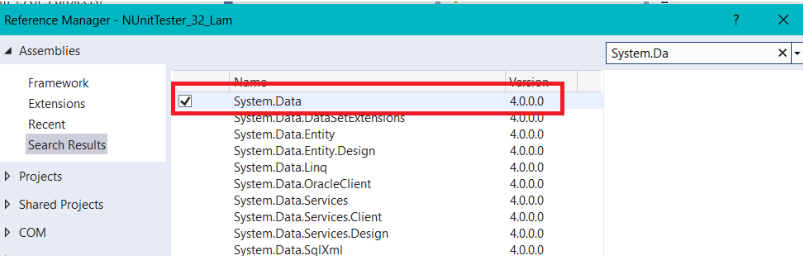
## Add Reference

* Ở project kiểm thử, thực hiện Add Reference để tham chiếu đến project cần thực hiện Unit Test.



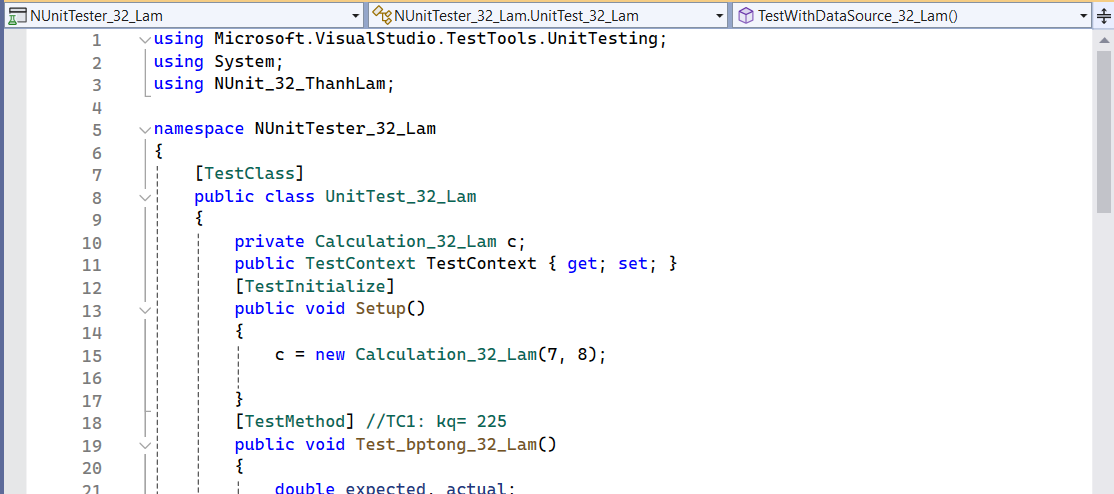


* Thêm Reference **System.Data** vào project kiểm thử



## Viết test case cho các hằng đẳng thức

* Tạo file **UnitTest\_32\_Lam.cs** trong project NUnitTester\_32\_Lam



### Test case 1: tính bình phương của 1 tổng

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập A | 7 | 225 | 225 | Pass |  |
| 2 | Nhập B | 8 |  |

| [TestMethod] //TC1: kq= 225  public void TC1\_bptong\_32\_Lam()  {  double expected, actual;  expected = 225;  actual = c.Excute\_32\_Lam("(A+B)^2");  Assert.AreEqual(expected, actual);  } |
| --- |

### Test case 2: tính bình phương của 1 hiệu

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập A | 7 | 1 | 1 | Pass |  |
| 2 | Nhập B | 8 |  |

| [TestMethod] //TC2: kq=1  public void TC2\_bphieu\_32\_Lam()  {  double expected, actual;  expected = 1;  actual = c.Excute\_32\_Lam("(A-B)^2");  Assert.AreEqual(expected, actual);  } |
| --- |

### Test case 3: tính hiệu 2 bình phương

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập A | 7 | -15 | -15 | Pass |  |
| 2 | Nhập B | 8 |  |

| [TestMethod] //TC3: kq=-15  public void TC3\_hieu2bp\_32\_Lam()  {  double expected, actual;  expected = -15;  actual = c.Excute\_32\_Lam("A^2 - B^2");  Assert.AreEqual(expected, actual);  } |
| --- |

### Test case 4: tính lập phương của 1 tổng

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập A | 7 | 3375 | 3375 | Pass |  |
| 2 | Nhập B | 8 |  |

| [TestMethod] //TC4: kq=3375  public void TC4\_lptong\_32\_Lam()  {  double expected, actual;  expected = 3375;  actual = c.Excute\_32\_Lam("(A+B)^3");  Assert.AreEqual(expected, actual);  } |
| --- |

### Test case 5: tính lập phương của 1 hiệu

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập A | 7 | -1 | -1 | Pass |  |
| 2 | Nhập B | 8 |  |

| [TestMethod] //TC5: kq=-1  public void TC5\_lphieu\_32\_Lam()  {  double expected, actual;  expected = -1;  actual = c.Excute\_32\_Lam("(A-B)^3");  Assert.AreEqual(expected, actual);  } |
| --- |

### Test case 6: tính tổng 2 lập phương

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập A | 7 | 855 | 855 | Pass |  |
| 2 | Nhập B | 8 |  |

| [TestMethod] //TC6: kq=855  public void TC6\_tong2lp\_32\_Lam()  {  double expected, actual;  expected = 855;  actual = c.Excute\_32\_Lam("A^3 + B^3");  Assert.AreEqual(expected, actual);  } |
| --- |

### Test case 7: tính hiệu 2 lập phương

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập A | 7 | -169 | -169 | Pass |  |
| 2 | Nhập B | 8 |  |

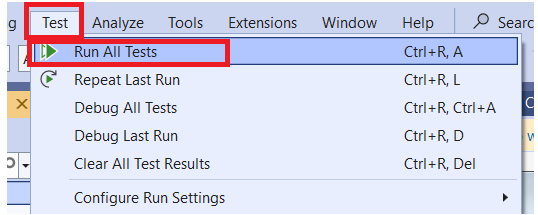
| [TestMethod] //TC7: kq=-169  public void TC7\_hieu2lp\_32\_Lam()  {  double expected, actual;  expected = -169;  actual = c.Excute\_32\_Lam("A^3 - B^3");  Assert.AreEqual(expected, actual);  } |
| --- |

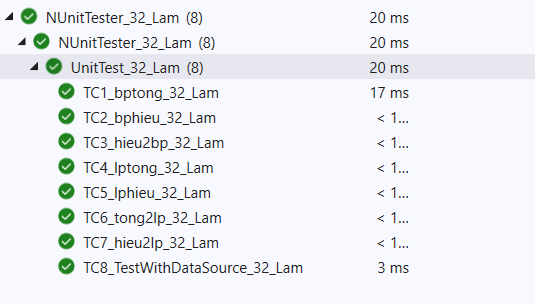
### Test case 8: test Data trên file csv

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Đọc A ở cột 1 file csv | cột 1 | cột 4 |  | Pass |  |
| 2 | Đọc B ở cột 2 file csv | cột 2 |  |

| [TestMethod]  [DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV",  @".\Data\_32\_Lam\TestData\_32\_Lam.csv", "TestData\_32\_Lam#csv", DataAccessMethod.Sequential)]  public void TC8\_TestWithDataSource\_32\_Lam()  {  int a, b ;  Double expected, actual;  string operation;  a = int.Parse(TestContext.DataRow[0].ToString());  b = int.Parse(TestContext.DataRow[1].ToString());  operation = TestContext.DataRow[2].ToString();  operation = operation.Remove(0, 1);  expected = Double.Parse(TestContext.DataRow[3].ToString());    c = new Calculation\_32\_Lam(a, b);  actual = c.Excute\_32\_Lam(operation);  Assert.AreEqual(expected, actual);  } |
| --- |

## Chạy Test case





# Phần B: SELENIUM WEBDRIVER

## Giới thiệu website: <https://whalishop.vn/>

* Đây là trang web bán quần áo cung cấp cho người dùng mua sắm quần áo online. Người dùng có thể tìm kiếm, xem chi tiết sản phẩm, thêm sản phẩm vào giỏ hàng và tiến hành thanh toán.

### Đặc tả use case các chức năng thực hiện kiểm thử

#### Chức năng đăng nhập

| **Mã use case** | UC001 | **Nội dung** | Đăng nhập |
| --- | --- | --- | --- |
| **Tác nhân** | Khách hàng | | |
| **Mô tả** | Tác nhân đăng nhập vào hệ thống để sử dụng 1 số chức năng hệ thống. | | |
| **Tiền điều kiện** | Trang web whalishop đã được tải và hiển thị. | | |
| **Hậu điều kiện** | Người dùng đã đăng nhập thành công và có thể tiếp tục mua sắm, quản lý thông tin cá nhân và thực hiện các thao tác khác trong tài khoản đã đăng nhập. | | |
| **Luồng hoạt động chính** | 1. Khách nhấn vào biểu tượng đăng nhập. 2. Khách nhập username, password. 3. Nhấn nút đăng nhập. 4. Hệ thống kiểm tra thông tin đăng nhập. 5. Nếu thông tin hợp lệ hệ thống sẽ chuyển đến trang chủ, ngược lại sẽ ở lại trang đăng nhập. | | |
| **Luồng hoạt động thay thế** | 2. a) Người dùng chọn đăng ký  2. b) Người dùng chọn quên mật khẩu  4. a) Hệ thống yêu cầu nhập đầy đủ thông tin.  4. b) Hệ thống thông báo thông tin nhập sai định dạng.  4. c) Hệ thống thông báo không tìm thấy tài khoản. | | |

#### 

#### Chức năng tìm kiếm

| **Mã use case** | UC002 | **Nội dung** | Tìm kiếm |
| --- | --- | --- | --- |
| **Tác nhân** | Khách hàng | | |
| **Mô tả** | Cho phép người dùng tìm kiếm sản phẩm dựa trên từ khóa | | |
| **Tiền điều kiện** | Trang web whalishop đã được tải và hiển thị. | | |
| **Hậu điều kiện** | Người dùng đã nhận được danh sách các sản phẩm phù hợp với từ khóa tìm kiếm | | |
| **Luồng hoạt động chính** | 1. Người dùng truy cập vào trang web bán quần áo. 2. Người dùng nhìn thấy ô tìm kiếm trên giao diện trang web. 3. Người dùng nhập từ khóa tìm kiếm vào ô tìm kiếm. 4. Người dùng nhấn Enter hoặc nhấp vào nút "Tìm kiếm". 5. Hệ thống xử lý yêu cầu tìm kiếm và hiển thị kết quả tìm kiếm. 6. Hệ thống hiển thị danh sách các sản phẩm phù hợp với từ khóa tìm kiếm. | | |
| **Luồng hoạt động thay thế** | Người dùng chọn tìm kiếm theo danh mục | | |

#### 

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

| **Mã use case** | UC003 | **Nội dung** | Thêm sản phẩm vào giỏ hàng |
| --- | --- | --- | --- |
| **Tác nhân** | Khách hàng | | |
| **Mô tả** | Cho phép người dùng chọn sản phẩm và thêm vào giỏ hàng | | |
| **Tiền điều kiện** | Trang web whalishop đã được tải và hiển thị. | | |
| **Hậu điều kiện** | Sản phẩm đã được thêm vào giỏ hàng của người dùng | | |
| **Luồng hoạt động chính** | 1. Người dùng đang xem thông tin chi tiết của một sản phẩm. 2. Người dùng nhấp vào nút "Thêm vào giỏ hàng". 3. Hệ thống kiểm tra số lượng của sản phẩm. 4. Nếu sản phẩm không còn, hệ thống hiển thị thông báo lỗi tương ứng và ngăn người dùng thêm sản phẩm vào giỏ hàng. 5. Nếu sản phẩm còn hàng, hệ thống thêm sản phẩm vào giỏ hàng của người dùng. | | |
| **Luồng hoạt động thay thế** | Người dùng chọn icon mua hàng nhanh | | |

## 

## Các thành phần HTML

|  | Tên | Mô tả |
| --- | --- | --- |
| Id | txtEmail\_32\_Lam | Lấy thuộc tính Id(“customer\_email”) của input để thực hiện gửi dữ liệu vào ô nhập email |
| btnThem\_32\_Lam | Tìm button dựa trên Id("add-to-cart") để Click vào nút Thêm vào giỏ |
| size\_32\_Lam | Click vào để đổi size |
| soLuong\_32\_Lam | Lấy dòng chữ để check số lượng trong ô giỏ hàng |
| Name | txtPass\_32\_Lam | Lấy thuộc tính Name(“customer[password]”) của input để gửi dữ liệu vào ô nhập password |
| ClassName | btnLoginClick\_32\_Lam | Lấy thuộc tính ClassName(“btn”) để thực hiện Click nút Đăng nhập |
| boxSearch\_32\_Lam | Lấy thuộc tính ClassName("search\_box") để tìm thanh tìm kiếm mới sau khi nhập tìm kiếm rỗng |
| XPath | txtSoLuong | Tìm ô nhập số lượng dựa trên XPath("//\*[@id=\"quantity\"]") |
| CssSelector | inputSearch\_32\_Lam | CssSelector("input[placeholder='Tìm kiếm...']") để gửi dữ liệu vào ô nhập tìm kiếm |
| divError\_32\_Lam | CssSelector("div[class = 'errors']"). Đây là thành phần div chứa thống báo lỗi sau khi nhập email chưa đăng kí. |
| formcart\_32\_Lam | CssSelector("form[id = 'cartform']"). Đây là form giỏ hàng xuất hiện sau khi nhấn thêm sản phẩm vào giỏ. |
| lbKQ\_32\_Lam | CssSelector("div[class='modal-dialog modal-lg']"). Thành phần div chứa cảnh báo sau khi thêm số lượng 0 sản phẩm vào giỏ hàng. |
| btnClose\_32\_Lam | CssSelector("span[class = 'hrv-close-modal']"). Dùng để đóng form giỏ hàng |
| sizeOption\_32\_Lam | CssSelector(" option[value='M']"). Dùng để đổi size M |

## Thiết kế kiểm thử

#### Chức năng 1: Đăng nhập

| **Test Case ID : DT1**  **Kiểm thử Đăng nhập** | | | | |
| --- | --- | --- | --- | --- |
| **Mục đích kiểm tra:** | Kiểm tra chức năng đăng nhập vào hệ thống | | | |
| **Chức năng kiểm tra:** | Kiểm tra dữ liệu để đăng nhập. | | | |
| **Test case:** | Mô tả | Email | Password | Trạng thái |
| 1. Đăng nhập thành công (Nhập đúng email và mật khẩu) | tan036075@gmail.com | TThanhlemmilk212 | Pass |
| 1. Đăng nhập không thành công (Nhập thiếu password) | tan036075@gmail.com |  | Pass |
| 1. Đăng nhập không thành công (Nhập thiếu email) |  | TThanhlemmilk212 | Pass |
| 1. Đăng nhập không thành công (Không nhập cả 2) |  |  | Pass |
| 1. Đăng nhập không thành công (Nhập email chưa đăng kí) | lam036075@gmail.com | TThanhlemmilk212 | Pass |
| 1. Đăng nhập không thành công (Nhập sai password) | tan036075@gmail.com | 123 | Pass |
| 1. Đăng nhập không thành công (Nhập email không hợp lệ) | tan036075gmail.com | TThanhlemmilk212 | Pass |
| **Dữ liệu dùng để kiểm tra:** | * Thông tin Email và Password | | | |
| **Các bước thực hiện:** | 1. Điền email: … 2. Điền mật khẩu: …. 3. Bấm nút "Đăng nhập". | | | |
| **Kết quả mong đợi:** | * Khi nhập sai hoặc nhập thiếu dữ liệu: Không thể đăng nhập vào website * Khi nhâp đúng thông tin: Truy cập thành công vào website | | | |

#### 

#### Chức năng 2: Tìm kiếm sản phẩm

#### 

| **Test Case ID : DT2**  **Kiểm thử Tìm kiếm sản phẩm** | | | |
| --- | --- | --- | --- |
| **Mục đích kiểm tra:** | Kiểm tra chức năng tìm kiếm sản phẩm | | |
| **Chức năng kiểm tra:** | Kiểm tra khả năng tìm kiếm thông tin liên quan đến sản phẩm. | | |
| **Test case:** | Mô tả | Từ khóa tìm kiếm | Trạng thái |
| 1. Hiển thị danh sách sản phẩm - theo từ khóa | “Ao” | Pass |
| 1. Không hiển thị danh sách - không nhập từ khóa | “” | Pass |
| **Dữ liệu dùng để kiểm tra:** | * Từ khóa | | |
| **Các bước thực hiện:** | 1. Nhập từ khóa 2. Nhấn Enter | | |
| **Kết quả mong đợi:** | * Khi nhập từ khóa không có trong danh sách sản phẩm: Không hiển thị danh sách sản phẩm lên trang web * Khi nhâp từ khóa có trong danh sách sản phẩm: Hiển thị danh sách sản phẩm | | |

#### 

#### Chức năng 3: Thêm sản phẩm vào giỏ hàng

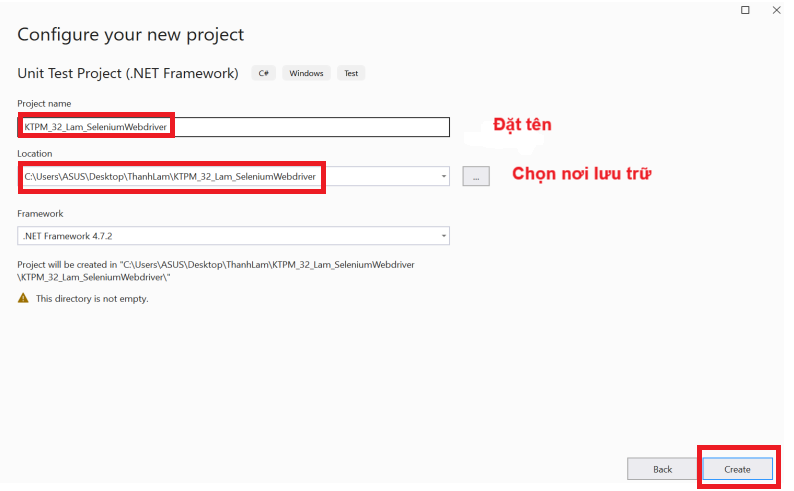
#### 

| **Test Case ID : DT3**  **Kiểm thử Thêm sản phẩm vào giỏ hàng** | | | |
| --- | --- | --- | --- |
| **Mục đích kiểm tra:** | Kiểm tra chức năng đăng nhập vào hệ thống | | |
| **Chức năng kiểm tra:** | Kiểm tra dữ liệu để đăng nhập. | | |
| **Test case:** | Mô tả | Số lượng | Trạng thái |
| 1. Thêm vào giỏ hàng thành công (click thêm vào giỏ hàng | 1 | Pass |
| 1. Thêm không thành thành công (Nhập số lượng là 0) | 0 | Pass |
| 1. Thêm không thành công (Nhập vào số lượng lớn hơn số lượng trong kho nên website cảnh báo | > số lượng trong kho | Pass |
| 1. Thêm thành công (Giỏ hàng có sẵn sản phẩm sau đó thêm sản phẩm với số lượng là 0) | 0 | Pass |
| 1. Thêm thành công (Giỏ hàng có sẵn sản phẩm sau đó thêm sản phẩm với số lượng là ) | 1 | Pass |
| 1. Thêm không thành công (Giỏ hàng có sẵn sản phẩm sau đó thêm sản phẩm với số lượng vượt quá số lượng trong kho ) | > số lượng trong kho | Pass |
| **Dữ liệu dùng để kiểm tra:** | * Số lượng | | |
| **Các bước thực hiện:** | Cách 1:  1. Nhấn “Thêm vào giỏ” với số lượng là 1.  Cách 2:   1. Điều chỉnh số lượng. 2. Nhấn “Thêm vào giỏ”. | | |
| **Kết quả mong đợi:** | * Khi thêm số lượng sản phẩm không vượt quá trong kho: thêm sản phẩm vào giỏ hàng thành công. * Khi thêm số lượng sản phẩm vượt quá trong kho: không thể thêm sản phẩm vào giỏ hàng. | | |

#### 

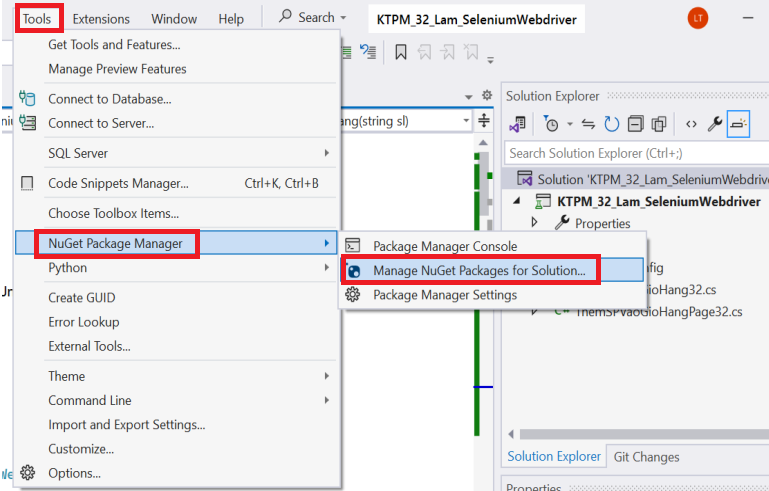
## Viết các test case với Selenium WebDriver

### Tạo project: KTPM\_32\_Lam\_SeleniumWebdriver

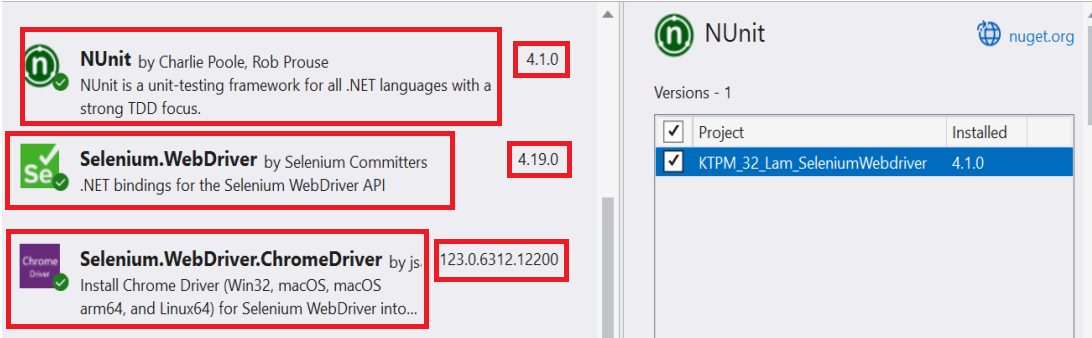


### Cài đặt package cho project

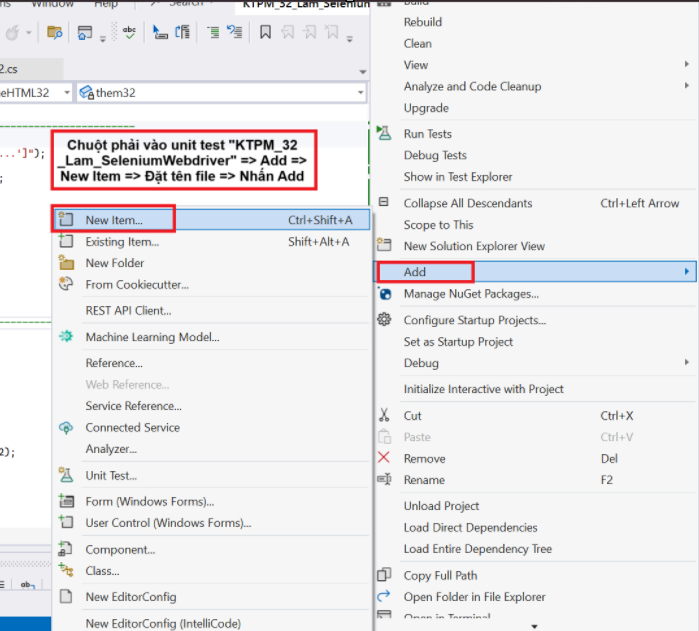
* Chọn Tool – Nuget Package Manage – Manage Nuget package solution



* Tại mục Browse nhập vào ô tìm kiếm với từ khoá “selenium webdriver” và tải 2 package có tên “**Selenium.Webdiver**” và “**Selenium.Webdriver.ChromeDriver**”
* Tiếp theo tìm kiếm “NUnit” và tải package có tên “**NUnit**”

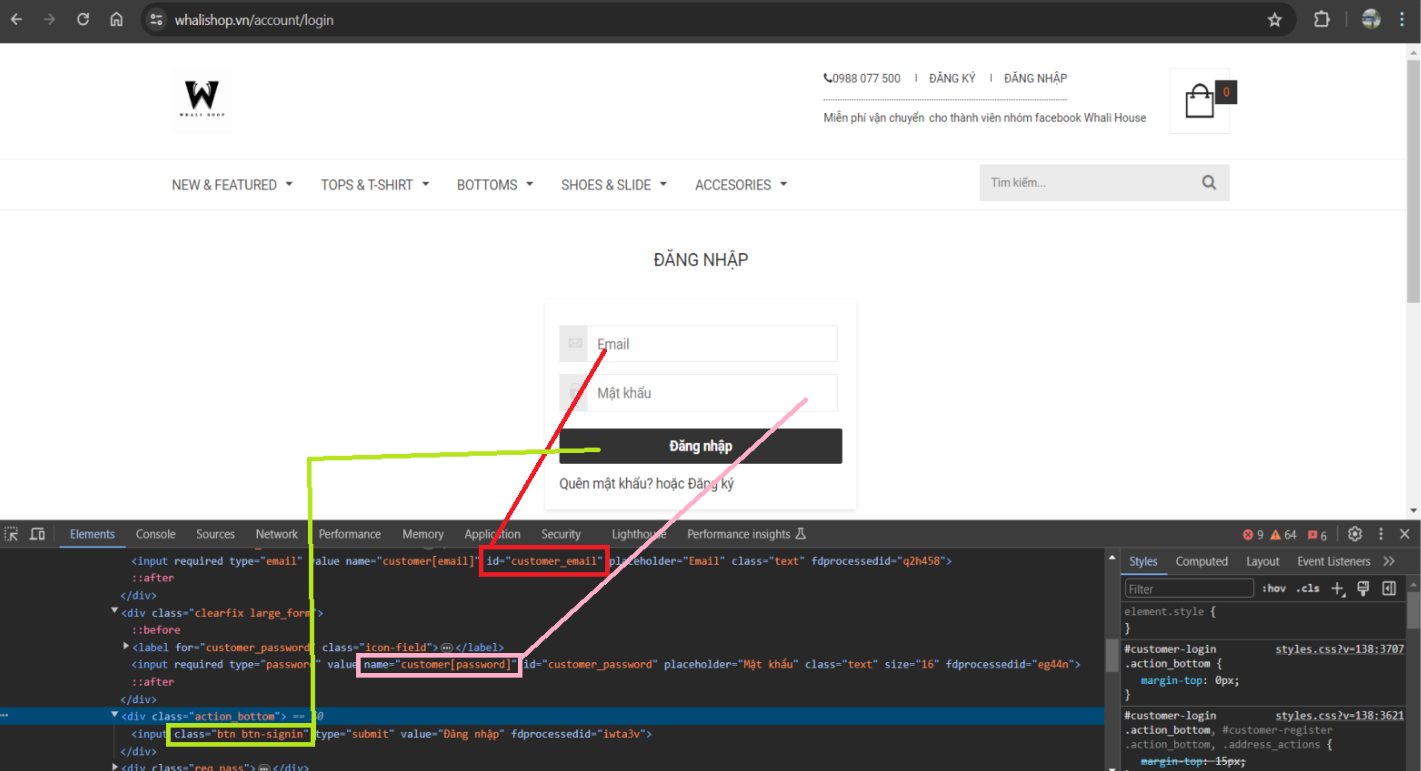


### Tạo file riêng tương tác với các thành phần html

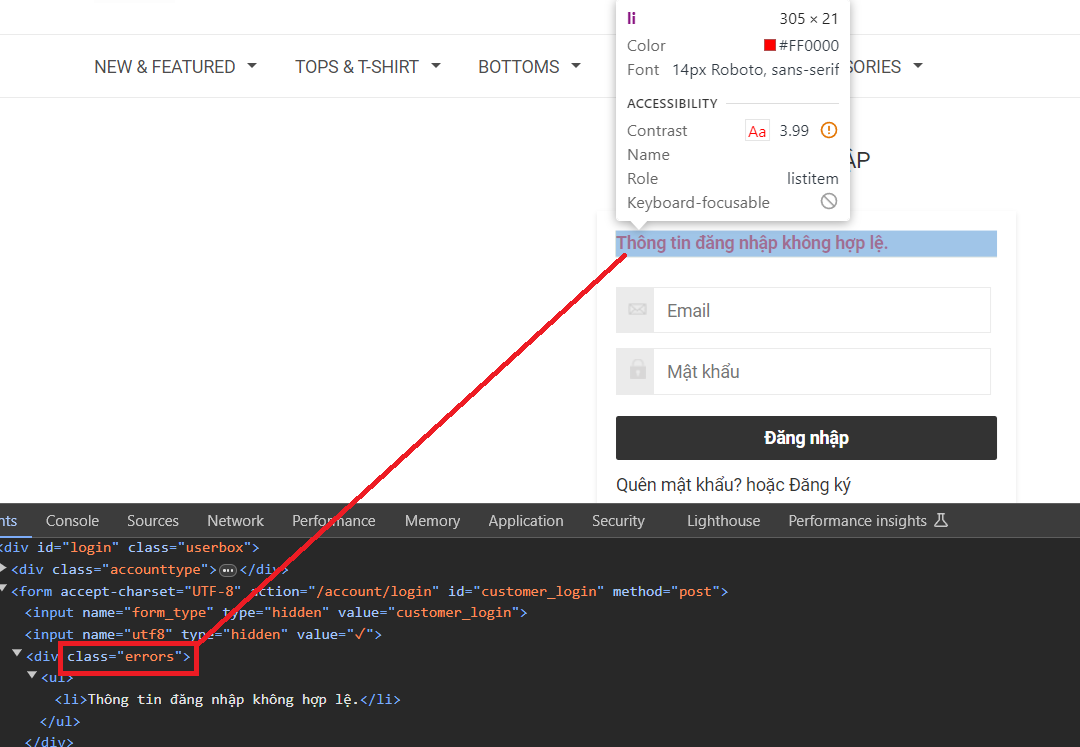


* Tạo file tên **PageHTML\_32\_Lam.cs**: đại diện chứa các phương thức để đi đến trang cần kiểm thử và tương tác với các thành phần HTML trên các trang (các ô nhập và các nút).

#### Tương tác với trang đăng nhập: <https://whalishop.vn/account/login>



* Sau khi nhập mật khẩu hoặc email sai thì sẽ xuất hiện lỗi “Thông tin đăng nhập không hợp lệ”



//ô nhập email

private IWebElement txtEmail\_32\_Lam => driver\_32\_Lam.FindElement(By.Id("customer\_email"));

//ô nhập password

private IWebElement txtPass\_32\_Lam => driver\_32\_Lam.FindElement(By.Name("customer[password]"));

//nút đăng nhập

private IWebElement btnLoginClick\_32\_Lam => driver\_32\_Lam.FindElement(By.ClassName("btn"));

public void InputDangNhap\_32\_Lam(string email, string pass)

{

//Gửi dữ liệu thông qua SendKeys

txtEmail\_32\_Lam.SendKeys(email);

txtPass\_32\_Lam.SendKeys(pass);

Thread.Sleep(2000);

btnLoginClick\_32\_Lam.Click();

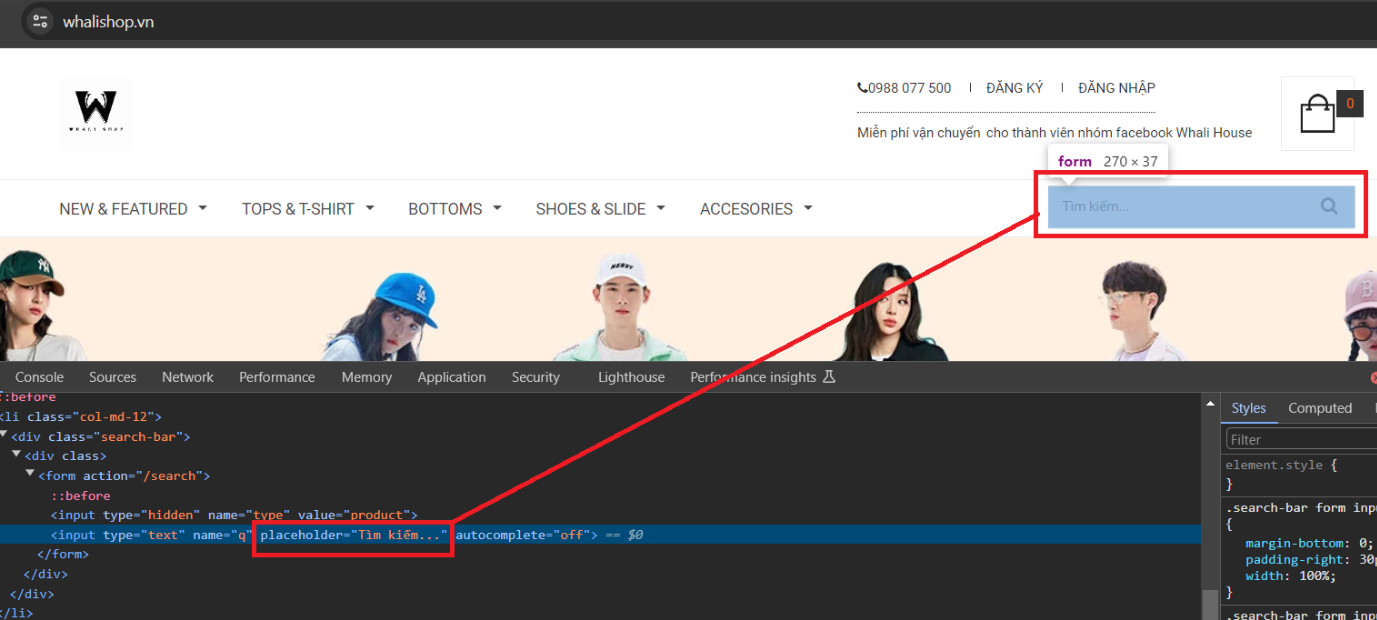
Thread.Sleep(5000);

}

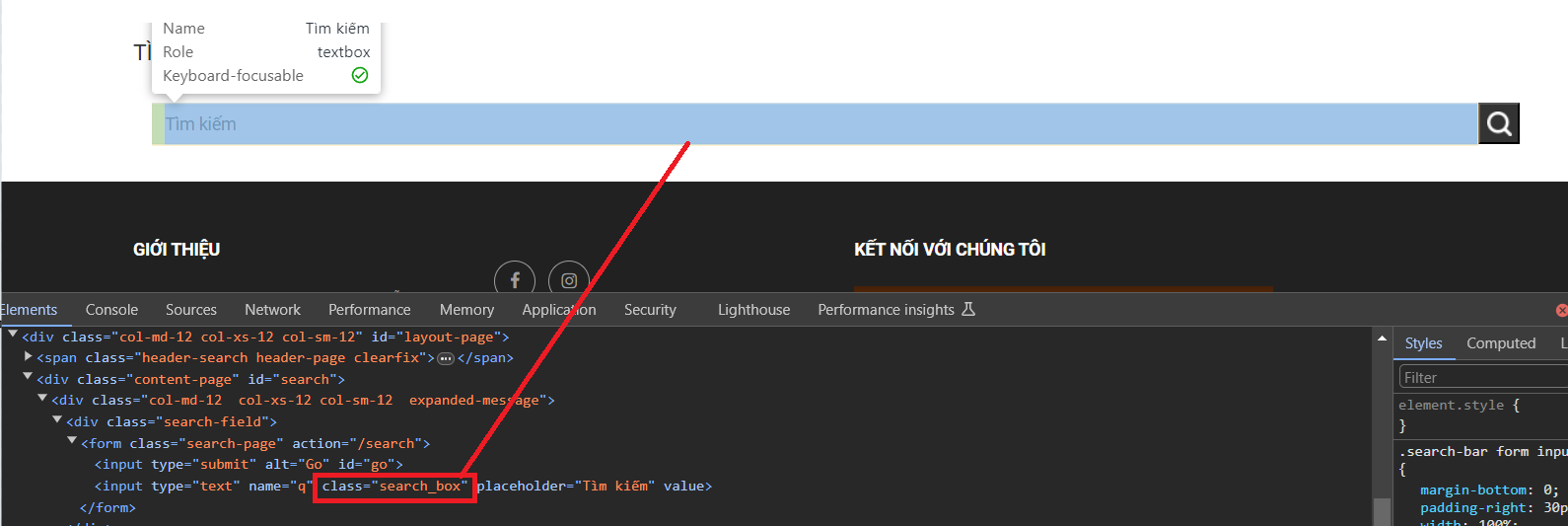
//Phần tử div xuất hiện lỗi sau khi nhập sai email hoặc password

public IWebElement divError\_32\_Lam => driver\_32\_Lam.FindElement(By.CssSelector("div[class = 'errors']"));

#### Tương tác với trang chủ để tìm kiếm:



* Sau khi không nhập từ khóa để tìm kiếm thì sẽ hiển thị thanh tìm kiếm lớn ở giữa website



//Thanh tìm kiếm

private IWebElement inputSearch\_32\_Lam => driver\_32\_Lam.FindElement(By.CssSelector("input[placeholder='Tìm kiếm...']"));

public void NhapTimKiem\_32\_Lam(string word\_32\_Lam)

{

inputSearch\_32\_Lam.SendKeys(word\_32\_Lam);

Thread.Sleep(2000);

inputSearch\_32\_Lam.SendKeys(Keys.Enter);

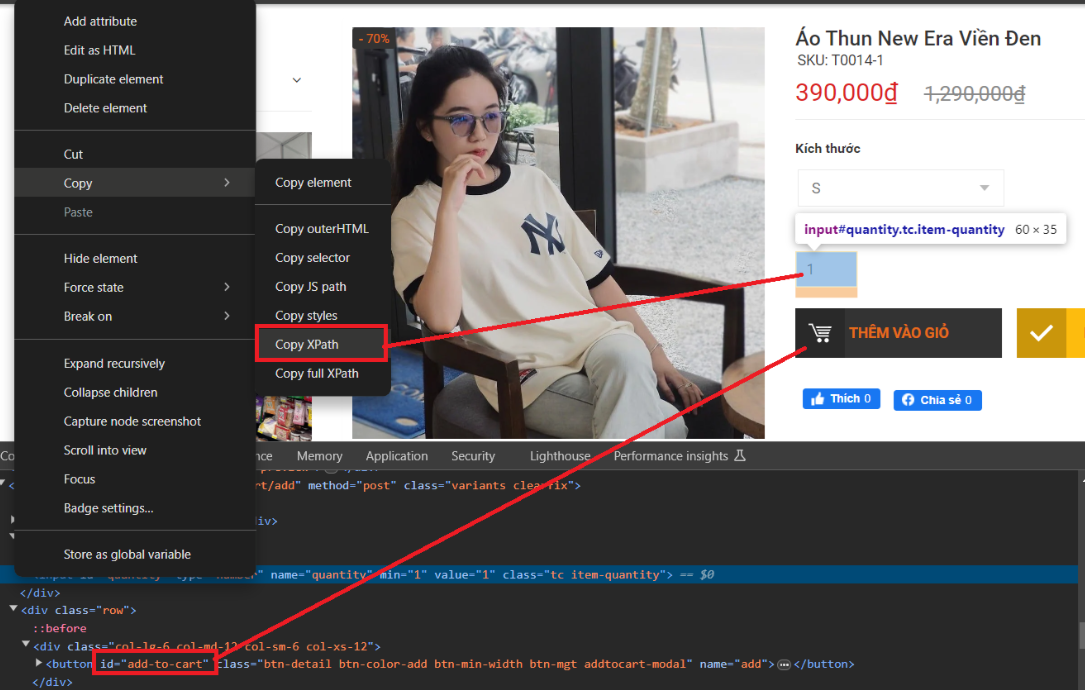
Thread.Sleep(5000);

}

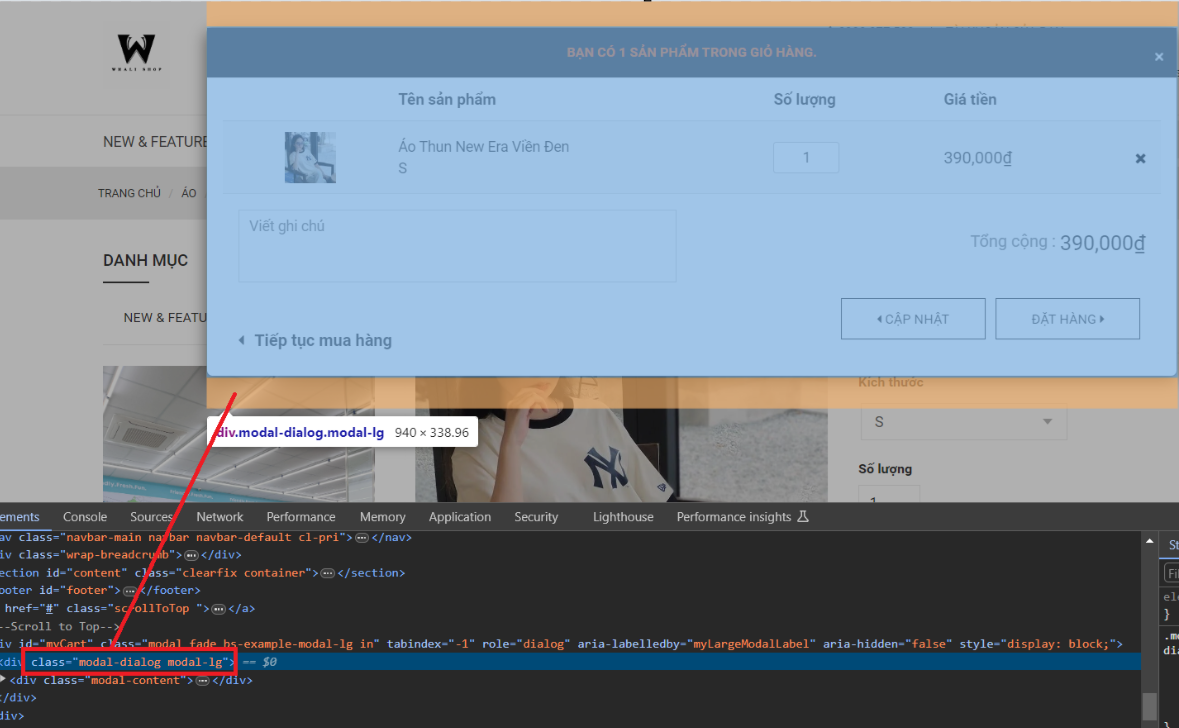
//thanh tìm kiếm lớn

public IWebElement boxSearch\_32\_Lam => driver\_32\_Lam.FindElement(By.ClassName("search\_box"));

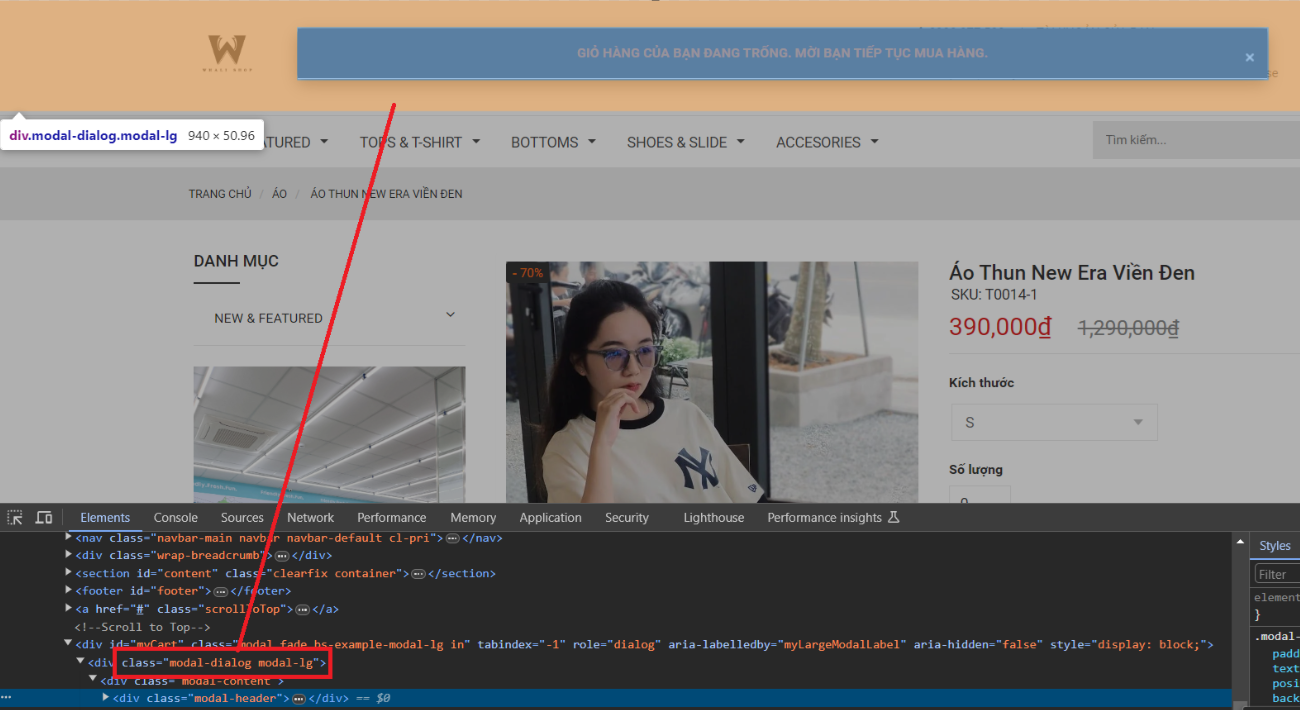
#### Tương tác với trang 1 sản phẩm bất kì để thêm vào giỏ hàng: chức năng này chúng ta vào 1 trang sản phẩm bất kì. Ví dụ: <https://whalishop.vn/products/ao-thun-new-era-vien-den>



* Sau khi nhấn nút thêm sản phẩm vào giỏ hàng với số lượng không vượt quá số lượng trong kho



* Khi trong giỏ hàng không chứa sản phẩm nào mà nhập số lượng là 0 và nhấn nút thêm vào giỏ hàng sẽ hiển thị ô thông báo “Giỏ hàng của bạn đang trống. Mời bạn tiếp tục mua hàng.”



//ô nhập số lượng

//ô nhập số lượng

private IWebElement txtSoLuong => driver\_32\_Lam.FindElement(By.XPath("//\*[@id=\"quantity\"]"));

//nút thêm vào giỏ hàng

private IWebElement btnThem\_32\_Lam => driver\_32\_Lam.FindElement(By.Id("add-to-cart"));

public void SoLuong1\_32\_Lam()

{

btnThem\_32\_Lam.Click();

Thread.Sleep(5000);

}

public void SoLuong\_32\_Lam(string sl\_32\_Lam)

{

//Xóa dữ liệu hiện có trong ô

txtSoLuong.Clear();

//Gửi dữ liệu mới vào ô

txtSoLuong.SendKeys(sl\_32\_Lam);

btnThem\_32\_Lam.Click();

Thread.Sleep(5000);

}

//form giỏ hàng

public IWebElement formcart\_32\_Lam => driver\_32\_Lam.FindElement(By.CssSelector("form[id = 'cartform']"));

//thành phần div chứa cảnh báo

public IWebElement lbKQ\_32\_Lam => driver\_32\_Lam.FindElement(By.CssSelector("div[class='modal-dialog modal-lg']"));

//Nút đóng form giỏ hàng

public IWebElement btnClose\_32\_Lam => driver\_32\_Lam.FindElement(By.CssSelector("span[class = 'hrv-close-modal']"));

//Thành phần div check số lượng

public IWebElement soLuong\_32\_Lam => driver\_32\_Lam.FindElement(By.Id("exampleModalLabel"));

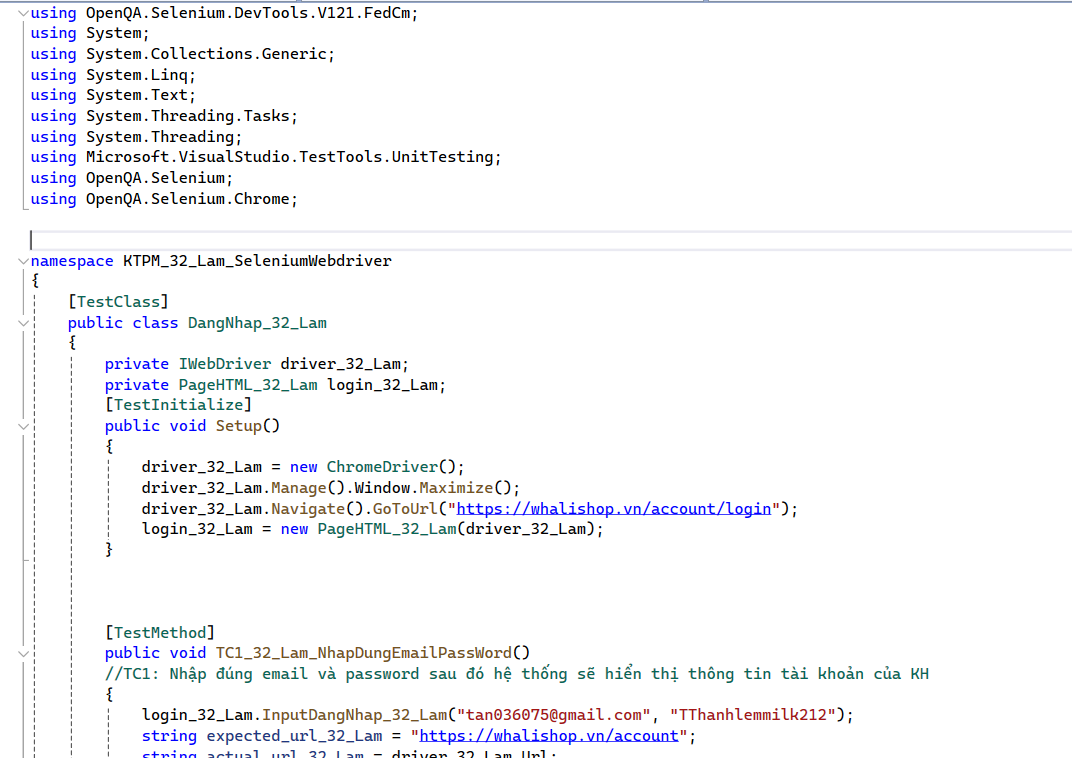
//Đổi size

public IWebElement size\_32\_Lam => driver\_32\_Lam.FindElement(By.Id("product-select-option-0"));

public IWebElement sizeOption\_32\_Lam => driver\_32\_Lam.FindElement(By.CssSelector(" option[value='M']"));

### Chức năng 1: Đăng nhập

* Tạo file **DangNhap\_32\_Lam.cs**: là lớp kiểm thử chứa các phương thức kiểm thử cho các kịch bản kiểm thử khác nhau liên quan đến chức năng đăng nhập.



private IWebDriver driver\_32\_Lam;

private PageHTML\_32\_Lam login\_32\_Lam;

[TestInitialize]

public void Setup()

{

driver\_32\_Lam = new ChromeDriver();

driver\_32\_Lam.Manage().Window.Maximize();

driver\_32\_Lam.Navigate().GoToUrl("https://whalishop.vn/account/login");

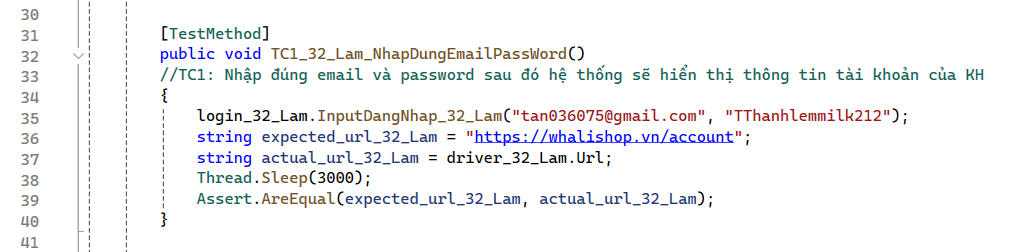
login\_32\_Lam = new PageHTML\_32\_Lam(driver\_32\_Lam);

}

#### 

#### Test case 1: Nhập đúng email và password

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập email | tan036075@gmail.com | https://whalishop.vn/account | https://whalishop.vn/account | Pass |  |
| 2 | Nhập password | TThanhlemmilk212 |  |



[TestMethod]

public void TC1\_32\_Lam\_NhapDungEmailPassWord()

//TC1: Nhập đúng email và password sau đó hệ thống sẽ hiển thị thông tin tài khoản của KH

{

login\_32\_Lam.InputDangNhap\_32\_Lam("tan036075@gmail.com", "TThanhlemmilk212");

string expected\_url\_32\_Lam = "https://whalishop.vn/account";

string actual\_url\_32\_Lam = driver\_32\_Lam.Url;

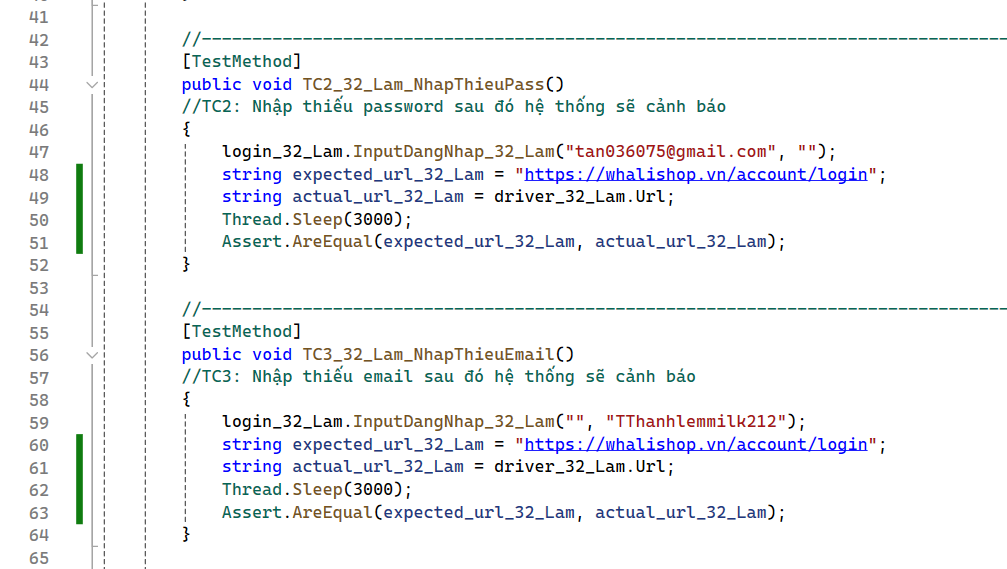
Thread.Sleep(3000);

Assert.AreEqual(expected\_url\_32\_Lam, actual\_url\_32\_Lam);

}

#### Test case 2: Nhập email nhưng thiếu password

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập email | tan036075@gmail.com | https://whalishop.vn/account/login | https://whalishop.vn/account/login | Pass |  |
| 2 | Nhập password |  |  |



[TestMethod]

public void TC2\_32\_Lam\_NhapThieuPass()

//TC2: Nhập thiếu password sau đó hệ thống sẽ cảnh báo

{

login\_32\_Lam.InputDangNhap\_32\_Lam("tan036075@gmail.com", "");

string expected\_url\_32\_Lam = "https://whalishop.vn/account/login";

string actual\_url\_32\_Lam = driver\_32\_Lam.Url;

Thread.Sleep(3000);

Assert.AreEqual(expected\_url\_32\_Lam, actual\_url\_32\_Lam);

}

#### Test case 3: Nhập password nhưng thiếu email

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập email |  | https://whalishop.vn/account/login | https://whalishop.vn/account/login | Pass |  |
| 2 | Nhập password | TThanhlemilk212 |  |

#### 

[TestMethod]

public void TC3\_32\_Lam\_NhapThieuEmail()

//TC3: Nhập thiếu email sau đó hệ thống sẽ cảnh báo

{

login\_32\_Lam.InputDangNhap\_32\_Lam("", "TThanhlemmilk212");

string expected\_url\_32\_Lam = "https://whalishop.vn/account/login";

string actual\_url\_32\_Lam = driver\_32\_Lam.Url;

Thread.Sleep(3000);

Assert.AreEqual(expected\_url\_32\_Lam, actual\_url\_32\_Lam);

}

#### 

#### Test case 4: Không nhập email và password

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập email |  | https://whalishop.vn/account/login | https://whalishop.vn/account/login | Pass |  |
| 2 | Nhập password |  |  |

#### 

[TestMethod]

public void TC4\_32\_Lam\_KhongNhapEmailPassword()

//TC4: Không nhập email và pass sau đó hệ thống sẽ cảnh báo

{

login\_32\_Lam.InputDangNhap\_32\_Lam("", "");

string expected\_url\_32\_Lam = "https://whalishop.vn/account/login";

string actual\_url\_32\_Lam = driver\_32\_Lam.Url;

Thread.Sleep(3000);

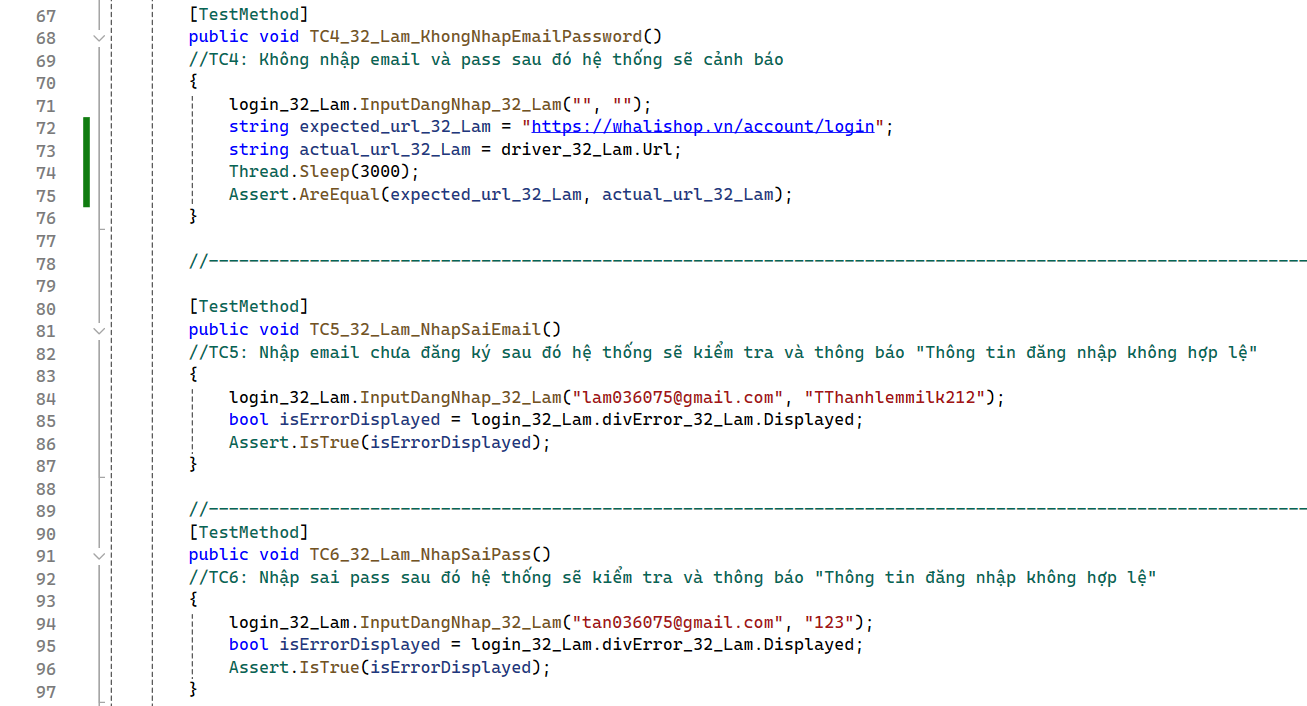
Assert.AreEqual(expected\_url\_32\_Lam, actual\_url\_32\_Lam);

}

#### 

#### Test case 5: Nhập sai email (email chưa đăng ký)

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập email | lam036075@gmail.com | Thông báo “thông tin đăng nhập không hợp lệ” | “Thông tin đăng nhập không hợp lệ” | Pass |  |
| 2 | Nhập password | TThanhlemmilk212 |  |



[TestMethod]

public void TC5\_32\_Lam\_NhapSaiEmail()

//TC5: Nhập email chưa đăng ký sau đó hệ thống sẽ kiểm tra và thông báo "Thông tin đăng nhập không hợp lệ"

{

login\_32\_Lam.InputDangNhap\_32\_Lam("lam036075@gmail.com", "TThanhlemmilk212");

bool isErrorDisplayed = login\_32\_Lam.divError\_32\_Lam.Displayed;

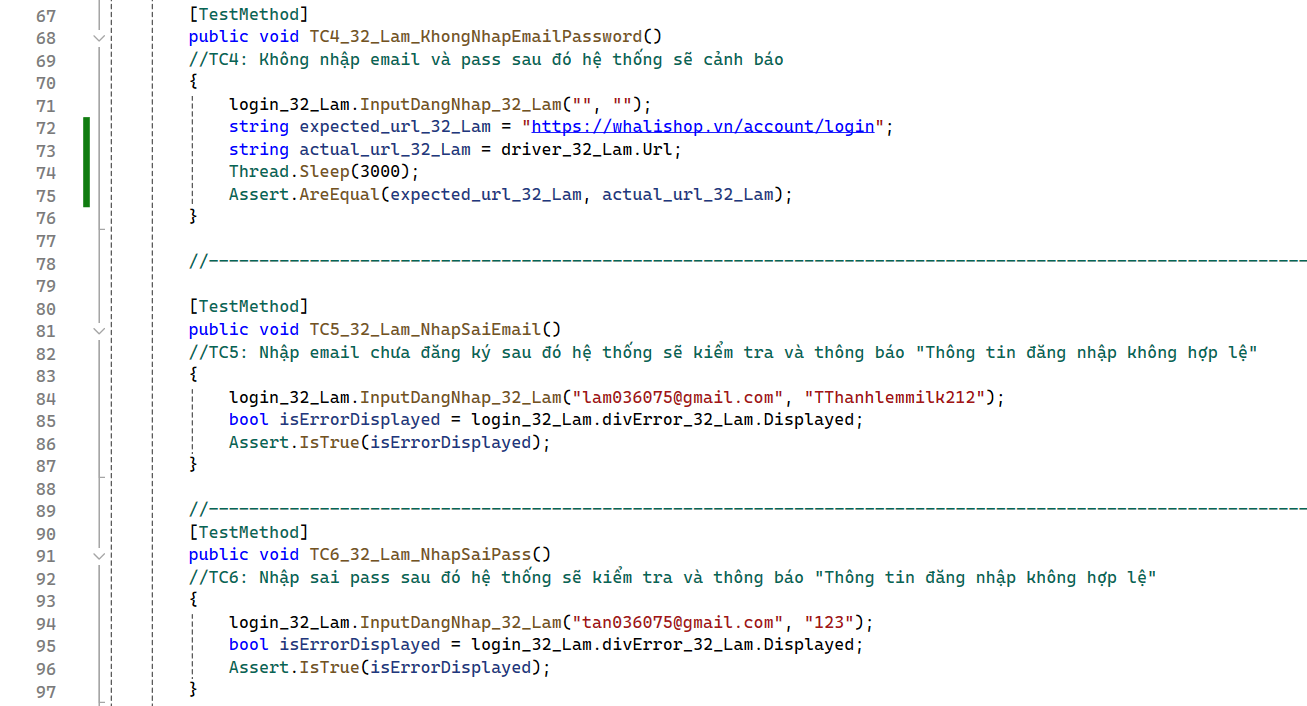
Assert.IsTrue(isErrorDisplayed);

}

#### 

#### Test case 6: Nhập sai password

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập email | tan036075@gmail.com | Thông báo “thông tin đăng nhập không hợp lệ” | “Thông tin đăng nhập không hợp lệ” | Pass |  |
| 2 | Nhập password | 123 |  |



[TestMethod]

public void TC6\_32\_Lam\_NhapSaiPass()

//TC6: Nhập sai pass sau đó hệ thống sẽ kiểm tra và thông báo "Thông tin đăng nhập không hợp lệ"

{

login\_32\_Lam.InputDangNhap\_32\_Lam("tan036075@gmail.com", "123");

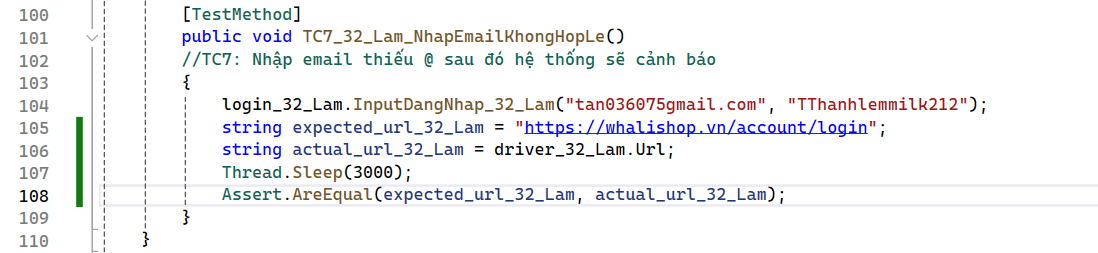
bool isErrorDisplayed = login\_32\_Lam.divError\_32\_Lam.Displayed;

Assert.IsTrue(isErrorDisplayed);

}

#### Test case 7: Nhập email không hợp lệ (email thiếu @)

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập email | lam036075gmail.com | https://whalishop.vn/account/login | https://whalishop.vn/account/login | Pass |  |
| 2 | Nhập password | TThanhlemmilk212 |  |



[TestMethod]

public void TC7\_32\_Lam\_NhapEmailKhongHopLe()

//TC7: Nhập email thiếu @ sau đó hệ thống sẽ cảnh báo

{

login\_32\_Lam.InputDangNhap\_32\_Lam("tan036075gmail.com", "TThanhlemmilk212");

string expected\_url\_32\_Lam = "https://whalishop.vn/account/login";

string actual\_url\_32\_Lam = driver\_32\_Lam.Url;

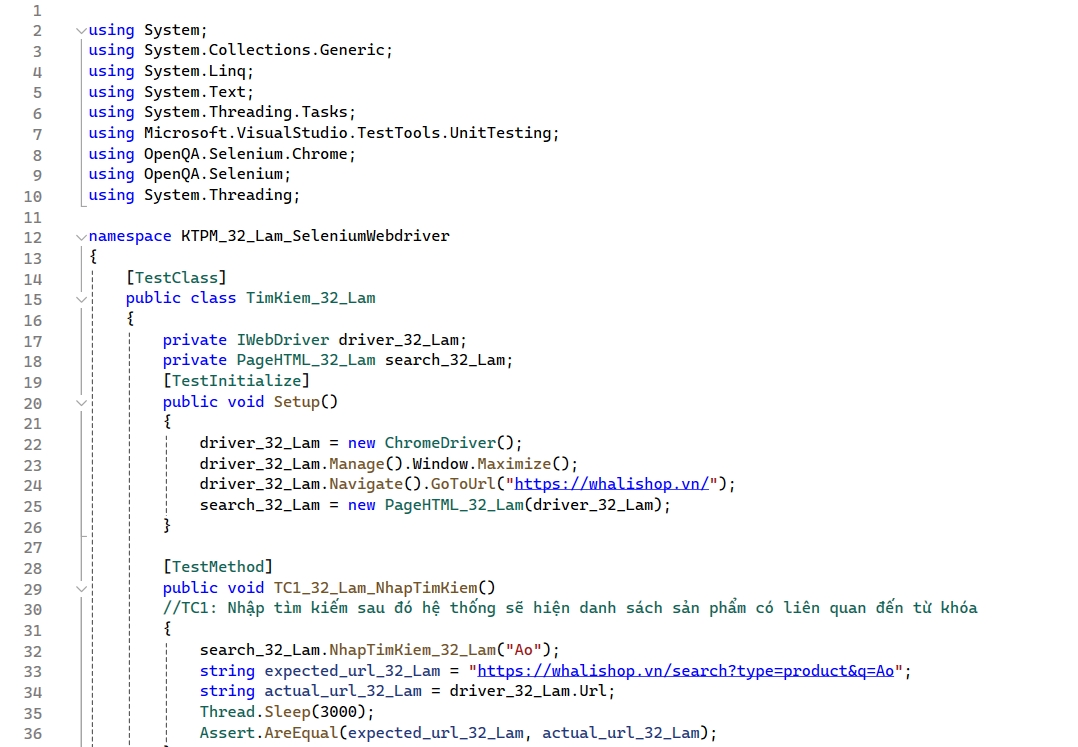
Thread.Sleep(3000);

Assert.AreEqual(expected\_url\_32\_Lam, actual\_url\_32\_Lam);

}

### Chức năng 2: Tìm kiếm sản phẩm

* Tạo file **TimKiem\_32\_Lam.cs**: là lớp kiểm thử chứa các phương thức kiểm thử cho các kịch bản kiểm thử khác nhau liên quan đến chức năng tìm kiếm.



private IWebDriver driver\_32\_Lam;

private PageHTML\_32\_Lam search\_32\_Lam;

[TestInitialize]

public void Setup()

{

driver\_32\_Lam = new ChromeDriver();

driver\_32\_Lam.Manage().Window.Maximize();

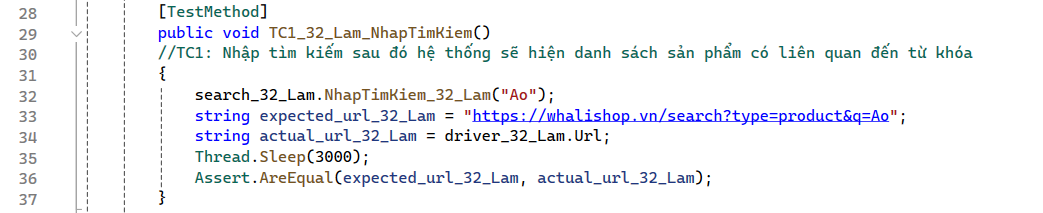
driver\_32\_Lam.Navigate().GoToUrl("https://whalishop.vn/");

search\_32\_Lam = new PageHTML\_32\_Lam(driver\_32\_Lam);

}

#### Test case 1: Nhập tìm kiếm

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập từ khoá | Ao | Chuyển tới trang xuất hiện danh sách sản phẩm phù hợp |  | Pass |  |



[TestMethod]

public void TC1\_32\_Lam\_NhapTimKiem()

//TC1: Nhập tìm kiếm sau đó hệ thống sẽ hiện danh sách sản phẩm có liên quan đến từ khóa

{

search\_32\_Lam.NhapTimKiem\_32\_Lam("Ao");

string expected\_url\_32\_Lam = "https://whalishop.vn/search?type=product&q=Ao";

string actual\_url\_32\_Lam = driver\_32\_Lam.Url;

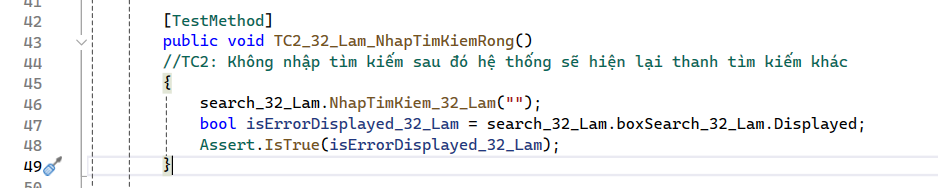
Thread.Sleep(3000);

Assert.AreEqual(expected\_url\_32\_Lam, actual\_url\_32\_Lam);

}

#### Test case 2: Không nhập tìm kiếm

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập từ khoá | “” | Xuất hiện thanh tìm kiếm khác ở giữa trang | Xuất hiện thanh tìm kiếm khác ở giữa trang | Pass |  |



[TestMethod]

public void TC2\_32\_Lam\_NhapTimKiemRong()

//TC2: Không nhập tìm kiếm sau đó hệ thống sẽ hiện lại thanh tìm kiếm khác

{

search\_32\_Lam.NhapTimKiem\_32\_Lam("");

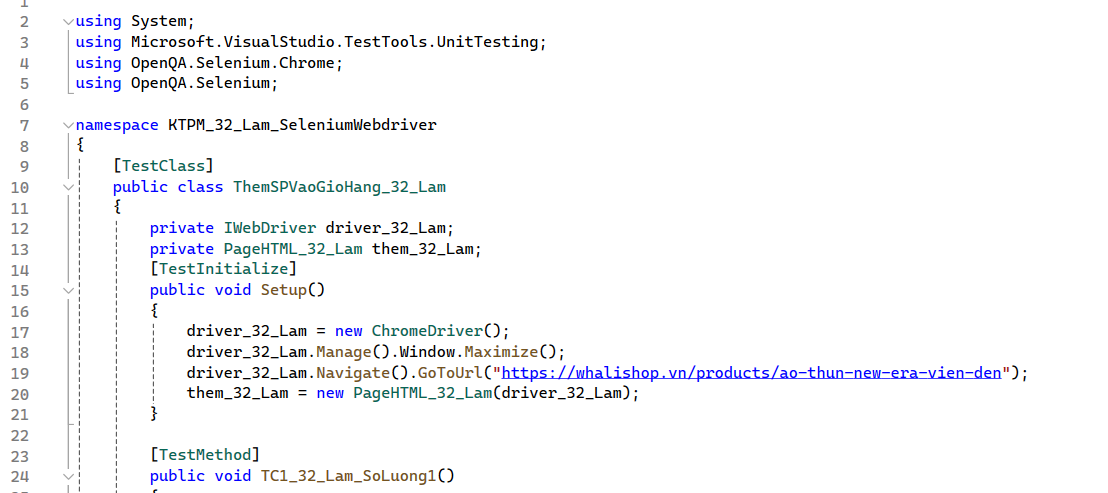
bool isErrorDisplayed\_32\_Lam = search\_32\_Lam.boxSearch\_32\_Lam.Displayed;

Assert.IsTrue(isErrorDisplayed\_32\_Lam);

}

### Chức năng 3: Thêm sản phẩm vào giỏ hàng

* Tạo file **ThemSPVaoGioHang\_32\_Lam.cs**: là lớp kiểm thử chứa các phương thức kiểm thử cho các kịch bản kiểm thử khác nhau liên quan đến chức năng thêm sản phẩm vào giỏ hàng.



private IWebDriver driver\_32\_Lam;

private PageHTML\_32\_Lam them\_32\_Lam;

[TestInitialize]

public void Setup()

{

driver\_32\_Lam = new ChromeDriver();

driver\_32\_Lam.Manage().Window.Maximize();

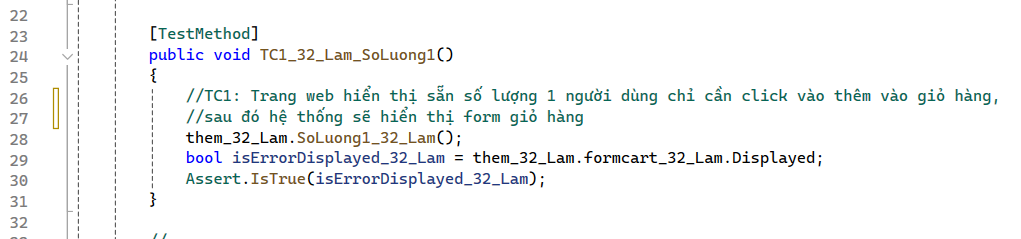
driver\_32\_Lam.Navigate().GoToUrl("https://whalishop.vn/products/ao-thun-new-era-vien-den");

them\_32\_Lam = new PageHTML\_32\_Lam(driver\_32\_Lam);

}

#### Test case 1: thêm số lượng là 1

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhấn nút “Thêm vào giỏ” | Số lượng = 1 | Xuất hiện form giỏ hàng giữa màn hình | Xuất hiện form giỏ hàng giữa màn hình | Pass |  |



[TestMethod]

public void TC1\_32\_Lam\_SoLuong1()

{

//TC1: Trang web hiển thị sẵn số lượng 1 người dùng chỉ cần click vào thêm vào giỏ hàng,

//sau đó hệ thống sẽ hiển thị form giỏ hàng

them\_32\_Lam.SoLuong1\_32\_Lam();

bool isErrorDisplayed\_32\_Lam = them\_32\_Lam.formcart\_32\_Lam.Displayed;

Assert.IsTrue(isErrorDisplayed\_32\_Lam);

}

#### Test case 2: thêm số lượng là 0

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập số lượng | số lượng = 0 | Xuất hiện ô cảnh báo của trang web | Xuất hiện ô cảnh báo của trang web | Pass |  |
| 2 | Nhấn nút “Thêm vào giỏ” |  |  |



[TestMethod]

public void TC2\_32\_Lam\_SoLuong0()

{

//TC2: Nhập số lượng là 0 sau đó click thêm vào giỏ sẽ hiện cảnh báo

them\_32\_Lam.SoLuong\_32\_Lam("0");

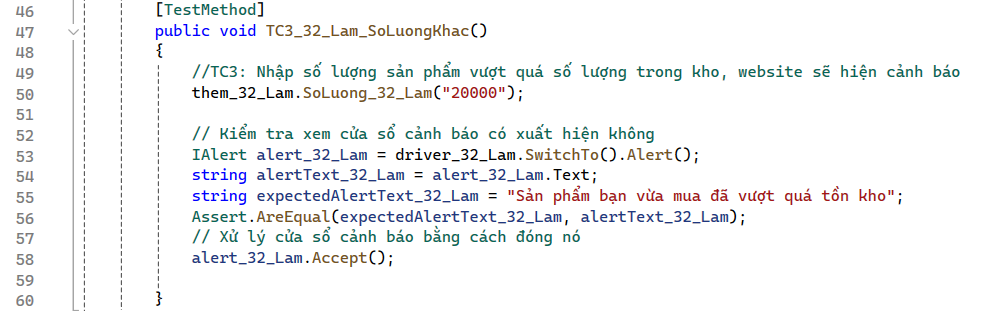
bool isErrorDisplayed\_32\_Lam = them\_32\_Lam.lbKQ\_32\_Lam.Displayed;

Assert.IsTrue(isErrorDisplayed\_32\_Lam);

}

#### Test case 3: thêm số lượng lớn hơn số lượng trong kho

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhập số lượng | số lượng = 20000 | Xuất hiện cảnh báo alert “Sản phẩm bạn vừa mua đã vượt quá tồn kho” | Xuất hiện cảnh báo alert “Sản phẩm bạn vừa mua đã vượt quá tồn kho” | Pass |  |
| 2 | Nhấn nút “Thêm vào giỏ” |  |  |



[TestMethod]

public void TC3\_32\_Lam\_SoLuongKhac()

{

//TC3: Nhập số lượng sản phẩm vượt quá số lượng trong kho, website sẽ hiện cảnh báo

them\_32\_Lam.SoLuong\_32\_Lam("20000");

// Kiểm tra xem cửa sổ cảnh báo có xuất hiện không

IAlert alert\_32\_Lam = driver\_32\_Lam.SwitchTo().Alert();

string alertText\_32\_Lam = alert\_32\_Lam.Text;

string expectedAlertText\_32\_Lam = "Sản phẩm bạn vừa mua đã vượt quá tồn kho";

Assert.AreEqual(expectedAlertText\_32\_Lam, alertText\_32\_Lam);

// Xử lý cửa sổ cảnh báo bằng cách đóng nó

alert\_32\_Lam.Accept();

}

#### Test case 4: thêm 1 sản phẩm vào giỏ hàng sau đó thêm số lượng là 0

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhấn nút “Thêm vào giỏ” | Số lượng = 1 | Xuất hiện form giỏ hàng giữa màn hình | Xuất hiện form giỏ hàng giữa màn hình | Pass |  |
| 3 | Đóng form giỏ hàng |  |  |
| 2 | Nhấn nút “Thêm vào giỏ” | Số lượng = 0 |  |

#### 

[TestMethod]

public void TC4\_32\_Lam\_GioHangCoSanSP\_SoLuong0()

{

//TC4: Thêm sản phẩm vào giỏ hàng với số lượng 1, sau đó lại thêm vào số lượng 0

them\_32\_Lam.SoLuong1\_32\_Lam();

them\_32\_Lam.btnClose\_32\_Lam.Click();

them\_32\_Lam.SoLuong\_32\_Lam("0");

bool isErrorDisplayed\_32\_Lam = them\_32\_Lam.formcart\_32\_Lam.Displayed;

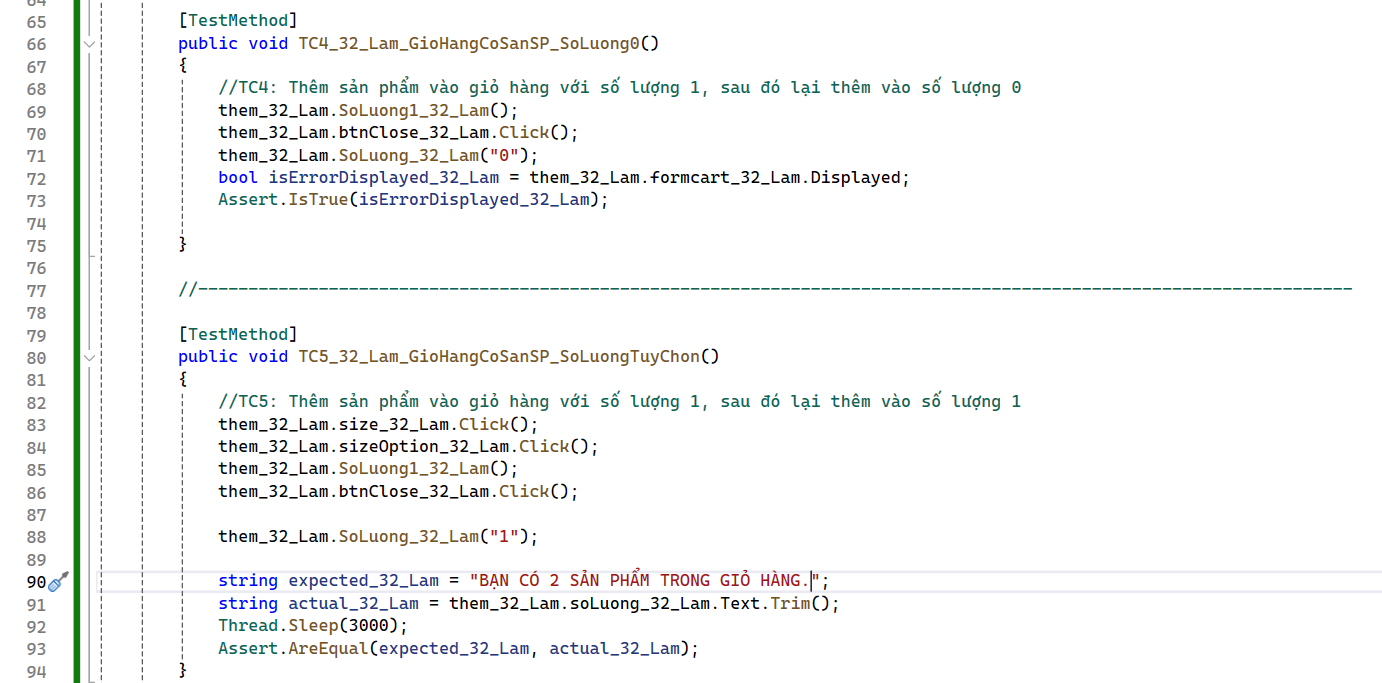
Assert.IsTrue(isErrorDisplayed\_32\_Lam);

}

#### Test case 5: thêm 1 sản phẩm vào giỏ hàng sau đó thêm số lượng là 1

* Phần này em đổi size khác để test

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhấn nút “Thêm vào giỏ” | Số lượng = 1 | Xuất hiện form giỏ hàng chứa câu “BẠN CÓ 2 SẢN PHẨM TRONG GIỎ HÀNG” | Xuất hiện form giỏ hàng chứa câu “BẠN CÓ 2 SẢN PHẨM TRONG GIỎ HÀNG” | Pass |  |
| 3 | Đóng form giỏ hàng |  |  |
| 2 | Nhấn nút “Thêm vào giỏ” | Số lượng = 1 |  |



[TestMethod]

public void TC5\_32\_Lam\_GioHangCoSanSP\_SoLuongTuyChon()

{

//TC5: Thêm sản phẩm vào giỏ hàng với số lượng 1, sau đó lại thêm vào số lượng 1

them\_32\_Lam.size\_32\_Lam.Click();

them\_32\_Lam.sizeOption\_32\_Lam.Click();

them\_32\_Lam.SoLuong1\_32\_Lam();

them\_32\_Lam.btnClose\_32\_Lam.Click();

them\_32\_Lam.SoLuong\_32\_Lam("1");

string expected\_32\_Lam = "BẠN CÓ 2 SẢN PHẨM TRONG GIỎ HÀNG.";

string actual\_32\_Lam = them\_32\_Lam.soLuong\_32\_Lam.Text.Trim();

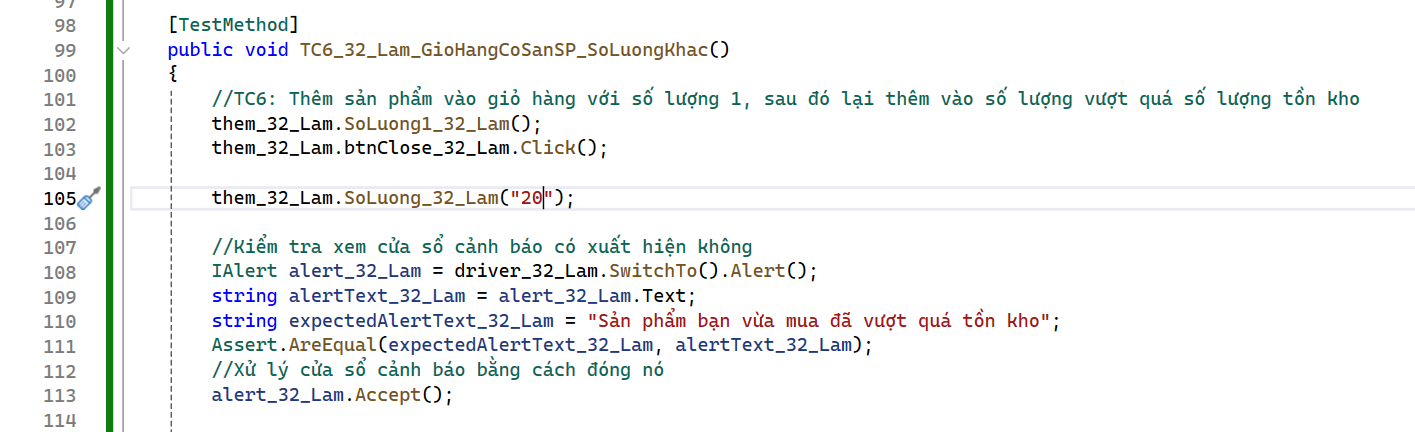
Thread.Sleep(3000);

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

}

#### Test case 6: thêm 1 sản phẩm vào giỏ hàng sau đó thêm số lượng lớn hơn số lượng trong kho

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status  (Pass/Fail) | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nhấn nút “Thêm vào giỏ” | Số lượng = 1 | Xuất hiện cảnh báo alert “Sản phẩm bạn vừa mua đã vượt quá tồn kho” | Xuất hiện cảnh báo alert “Sản phẩm bạn vừa mua đã vượt quá tồn kho” | Pass |  |
| 3 | Đóng form giỏ hàng |  |  |
| 2 | Nhấn nút “Thêm vào giỏ” | Số lượng = 20 |  |  |



[TestMethod]

public void TC6\_32\_Lam\_GioHangCoSanSP\_SoLuongKhac()

{

//TC6: Thêm sản phẩm vào giỏ hàng với số lượng 1, sau đó lại thêm vào số lượng vượt quá số lượng tồn kho

them\_32\_Lam.SoLuong1\_32\_Lam();

them\_32\_Lam.btnClose\_32\_Lam.Click();

them\_32\_Lam.SoLuong\_32\_Lam("20");

//Kiểm tra xem cửa sổ cảnh báo có xuất hiện không

IAlert alert\_32\_Lam = driver\_32\_Lam.SwitchTo().Alert();

string alertText\_32\_Lam = alert\_32\_Lam.Text;

string expectedAlertText\_32\_Lam = "Sản phẩm bạn vừa mua đã vượt quá tồn kho";

Assert.AreEqual(expectedAlertText\_32\_Lam, alertText\_32\_Lam);

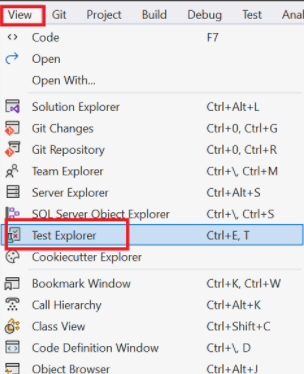
//Xử lý cửa sổ cảnh báo bằng cách đóng nó

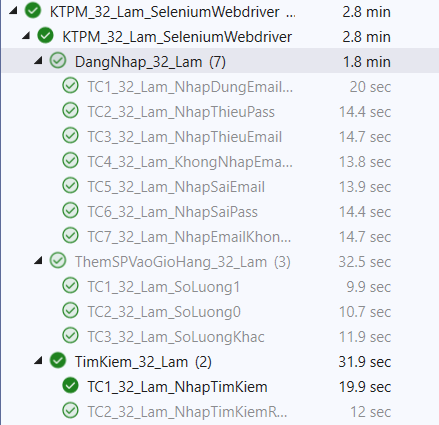
alert\_32\_Lam.Accept();

}

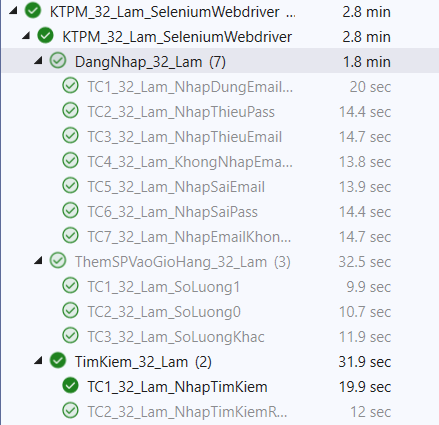
## Thực thi kiểm thử

* Mở cửa sổ Test Explorer để chạy test case: View => Test Explorer







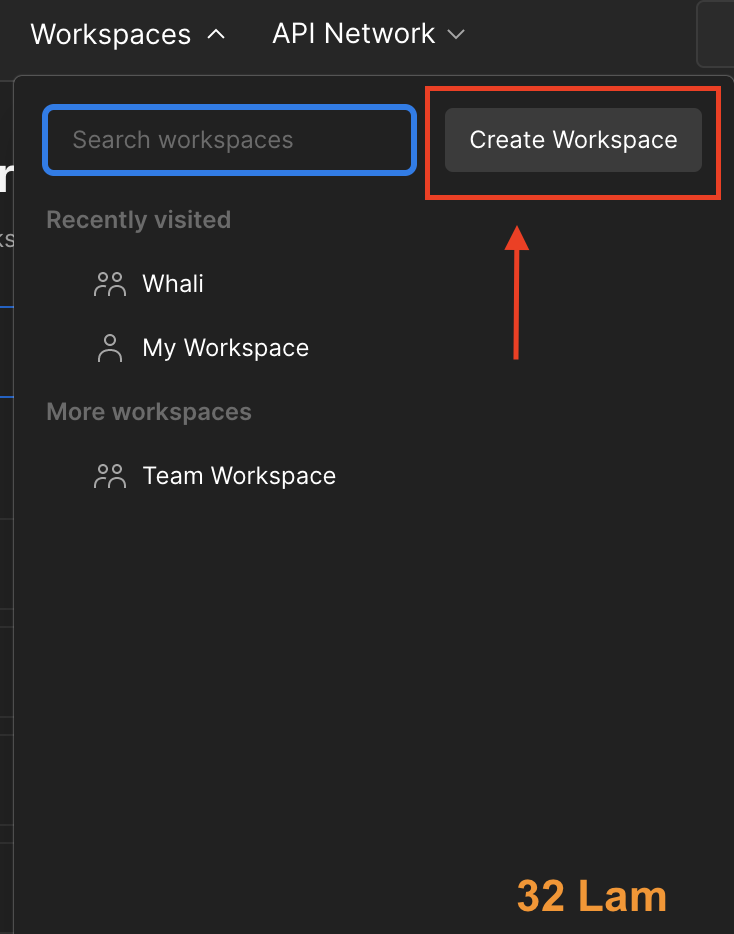


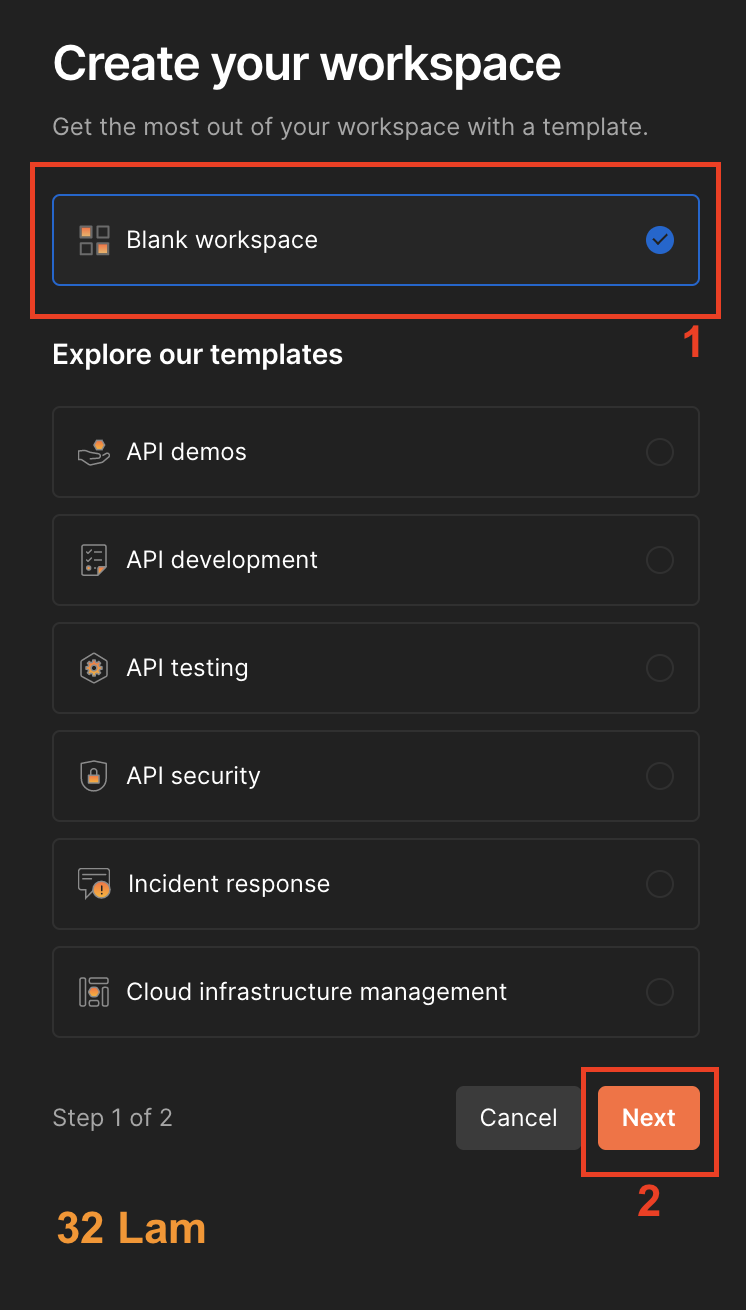
# Phần C: API POST

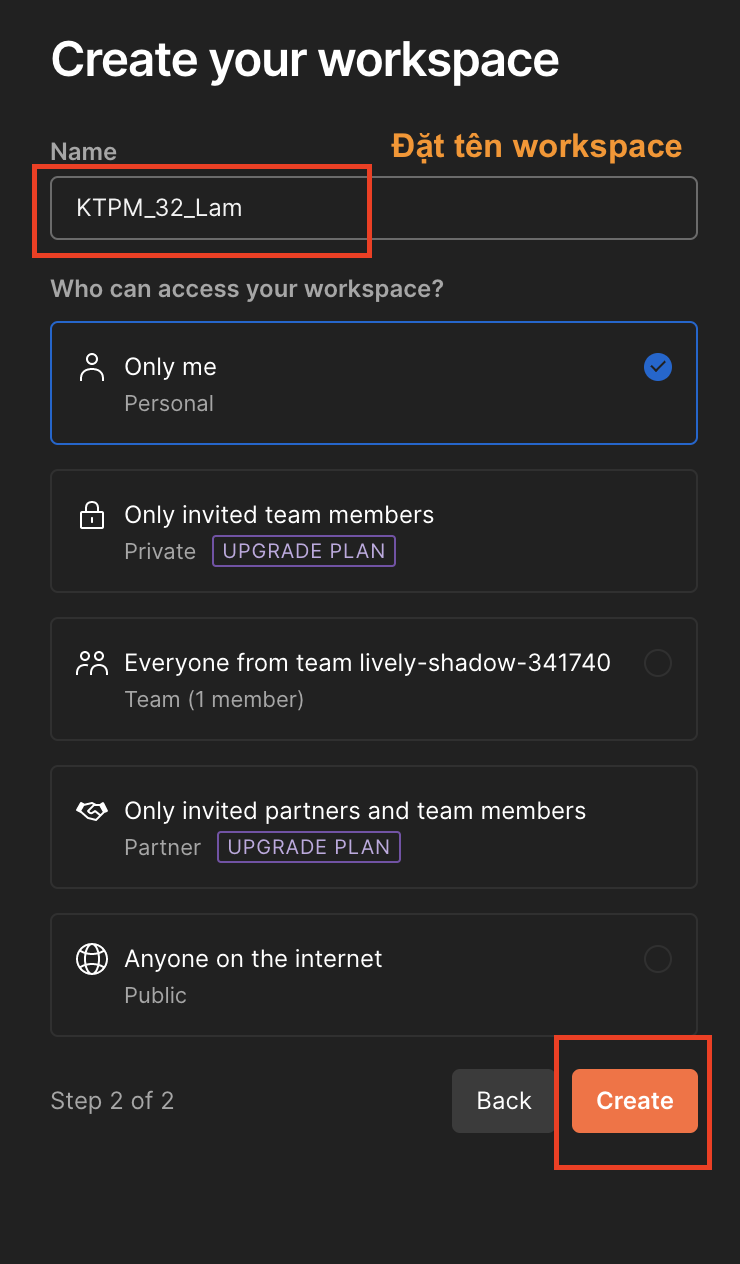
## Postman.

* Link download: <https://www.postman.com/downloads/>
* Em sẽ sử dụng trang web <https://www.haravan.com/> để làm API test.

## Tạo Workspace có tên KTPM\_32\_Lam

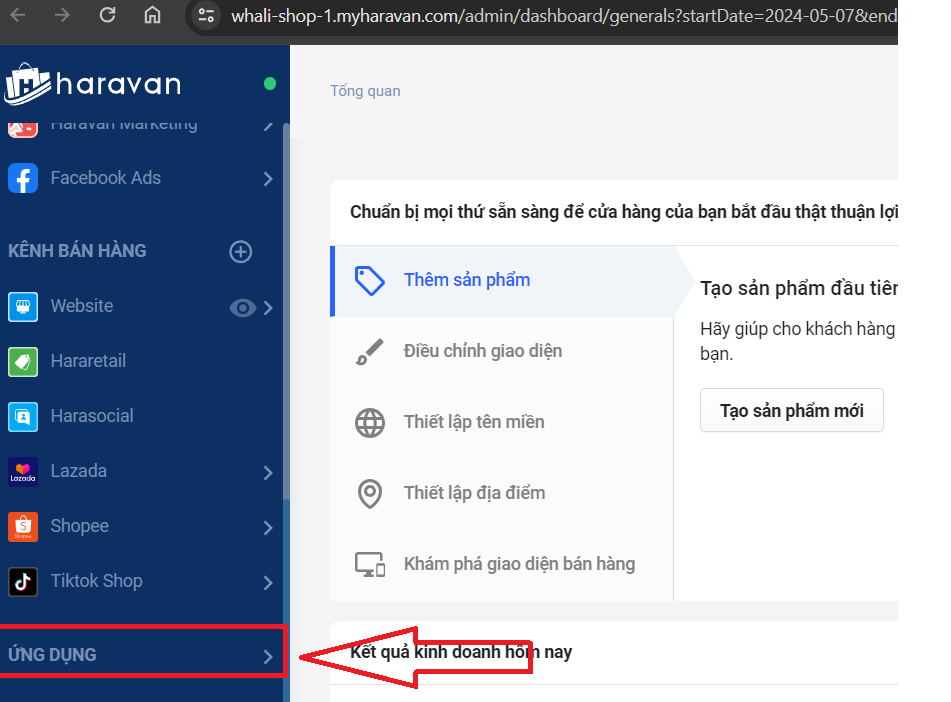


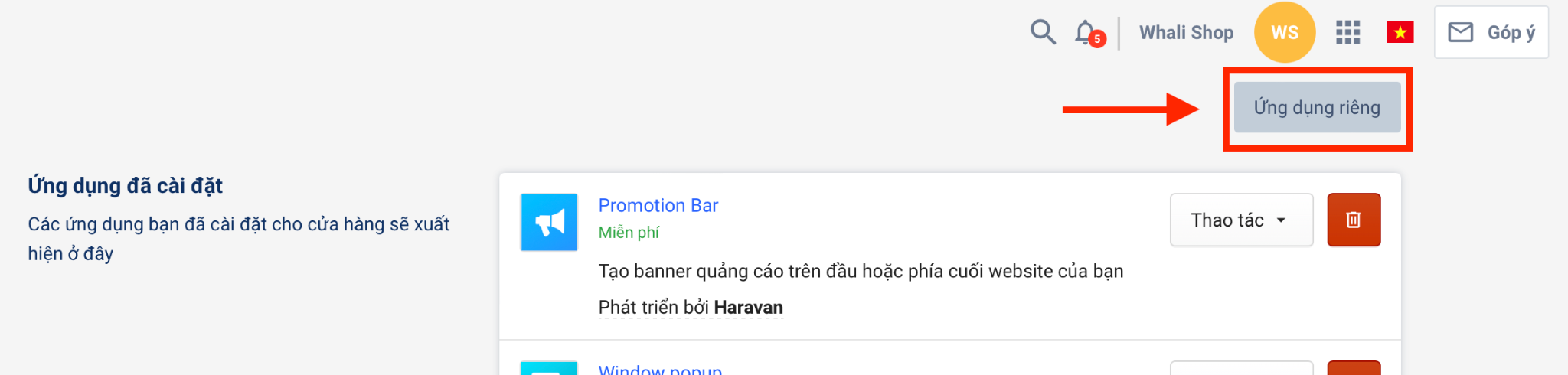




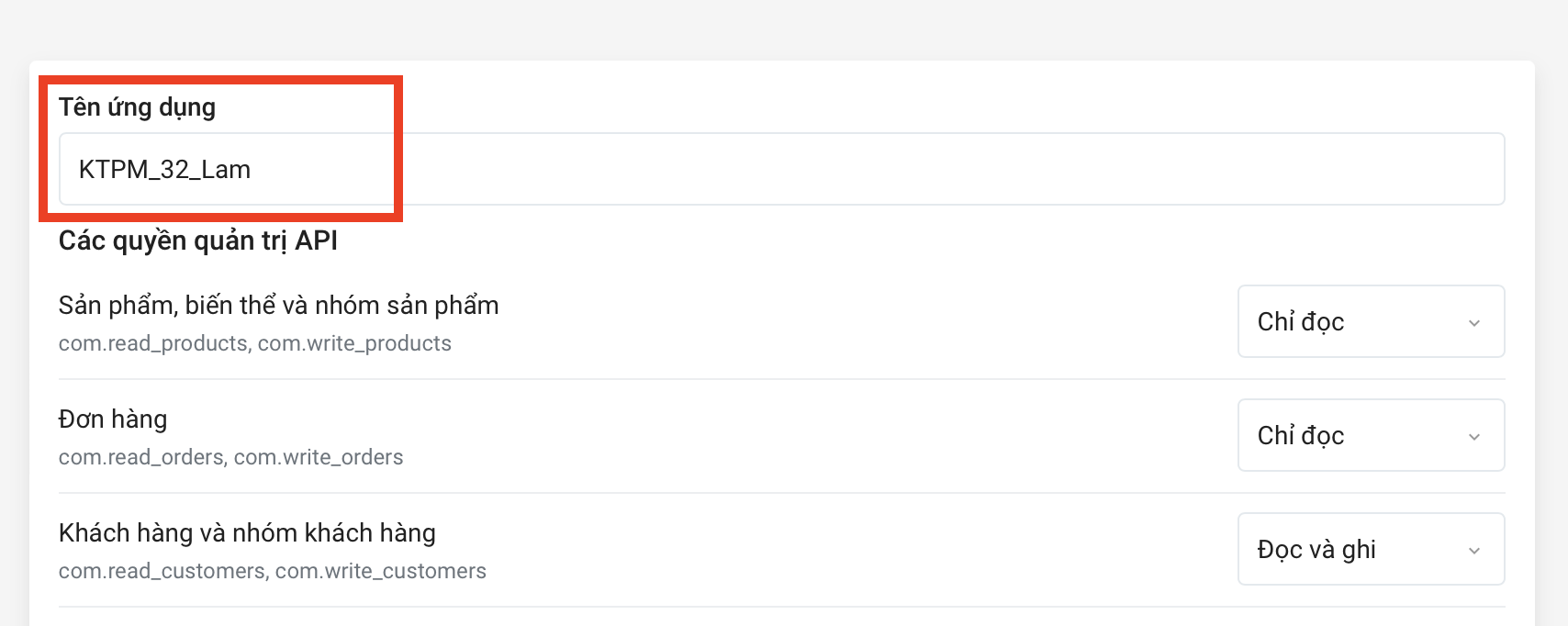
## Lấy quyền website để test API

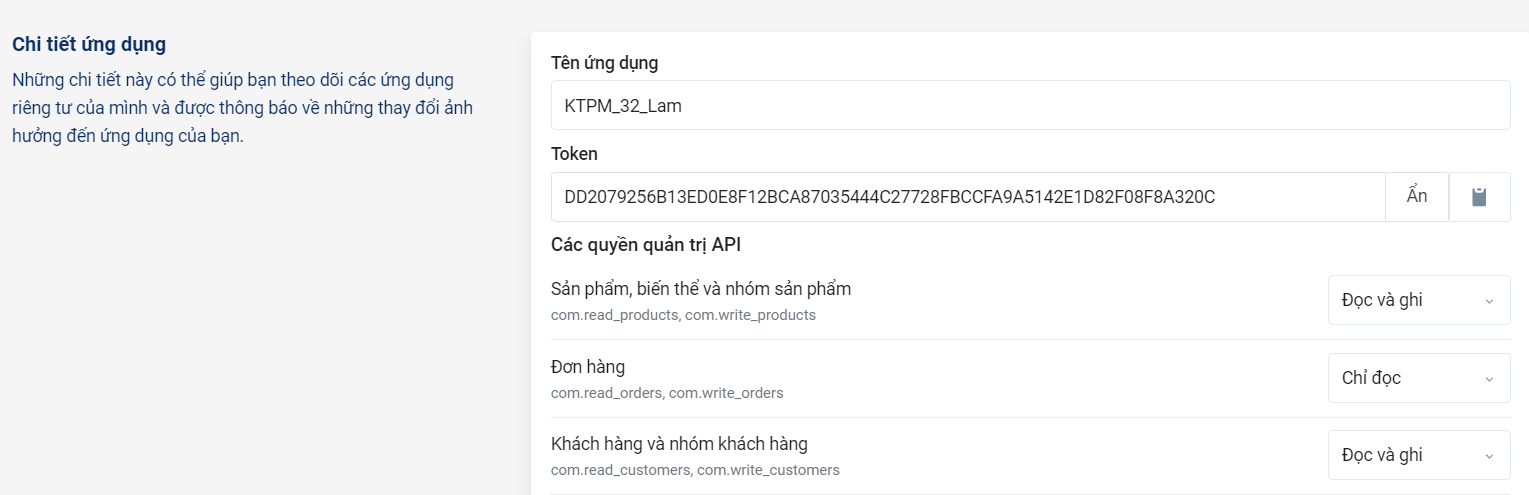
* Trước khi lấy quyền website thì chúng ta cần [đăng ký tài khoản](https://accounts.haravan.com/account/register?returnUrl=%2Fconnect%2Fauthorize%2Fcallback%3Fclient_id%3Domnipower.client%26redirect_uri%3Dhttps%253A%252F%252Fmyharavan.com%252Fadmin%252Fsignin-oidc%26response_type%3Dcode%2520id_token%26scope%3Dopenid%2520profile%2520org%2520userinfo%2520email%2520web_api%2520com_api%2520hac_api%2520lzd_api%2520tiki_api%2520adr_api%2520shopee_api%2520loy_api%2520zalo_api%2520billing_api%2520sendo_api%2520wh_api%2520partner_api%2520social_api%2520file_api%26response_mode%3Dform_post%26nonce%3D638506620013489926.YjNjYWM5ODctZTIzZS00MmVlLTk1NDMtZTljY2M0NjM0YjMxNGE1MTVmOTQtOTQ1My00OGVjLThlMDgtOGE2ZmY0MTk2ZjFh%26state%3DCfDJ8OhB9w14UplFo0ysyCS0-PIB4qeKUzFaHD4rdjHUnO6TjilGX9dEgGHGI2nsI0zWOu3L6tQcpaBAJJ_WMbdwLwlZH_NjJDQXCVuNeFh7C51sdEF2KAyDUF-CwZYvGfqkghC1tUiAHHfCXL726_zzYgUaHSElQhHm2lhdhLGNydF6vKAvBucrexxlvwBS98XHX6oQJVMUKzZVCrRw5A9N5FHLcDxFqX0deRYZCRYjLJqWAGmQbDPu5nreokt9N3qnexq4El2A9svQbMezZeoDBU6YEBUe-9CIy8dSRSkit4A1u6cTAV8QEiARlKtnMws-mI4q4jdoKulnDtOWsHLwn6BbBKwNv4OjZqcF7jRnXU1f7t9boxhMiIN4r8P6OCDO62qVb0EI9-DOz7HjRSugVTg%26x-client-SKU%3DID_NET8_0%26x-client-ver%3D7.0.3.0&locale=vi)
* Sau khi đăng ký xong thì bạn có thể thêm sản phẩm, khách hàng,... Trên trang quản trị bán hàng.
* Vào trang [Bảng điều khiển quản trị cửa hàng](https://myharavan.com/admin/apps) để tạo ứng dụng riêng hoặc click vào “ Ứng dụng” ở thanh bên trái.



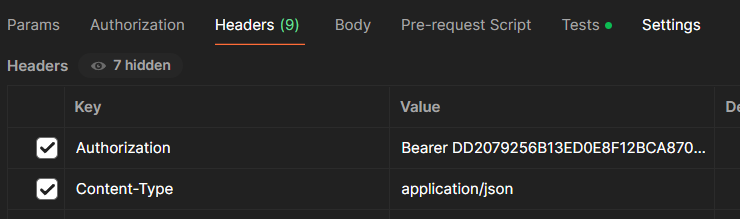






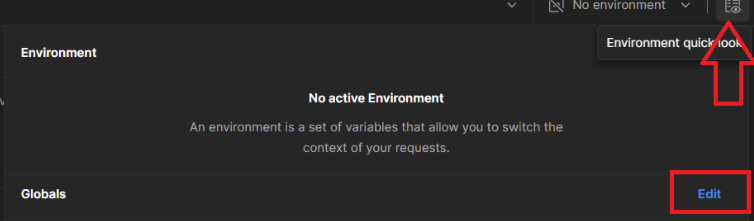


* Sau khi lấy được access\_token chúng ta sẽ vô POSTMAN-> chọn phần Headers -> nhập “Authorization” là “Bearer DD2079256B13ED0E8F12BCA87035444C27728FBCCFA9A5142E1D82F08F8A320C”
* Nhập “Content-type” là “application/json”

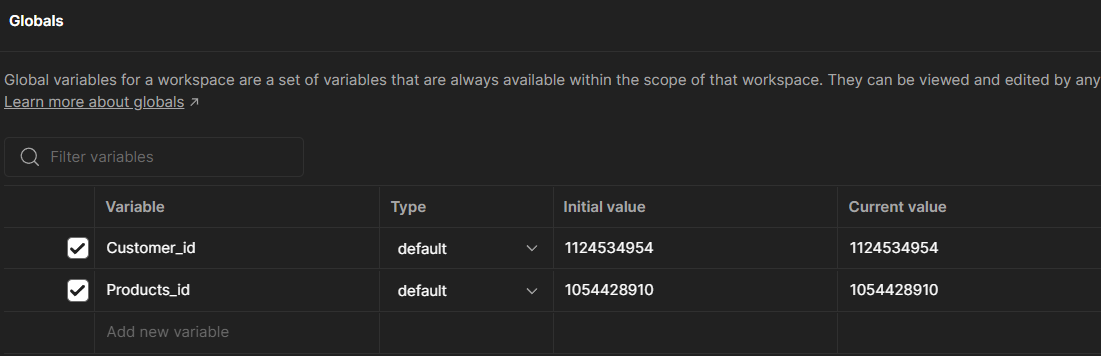


* **LƯU Ý**: Mỗi khi tạo request thì chúng ta luôn vào Headers của request đó để điền 2 giá trị Authorization và Content-type.

## Thiết lập biến môi trường trong postman



* Sau đó chúng ta tạo biến môi trường là những giá trị mà chúng ta hay sử dụng lặp đi lặp lại.



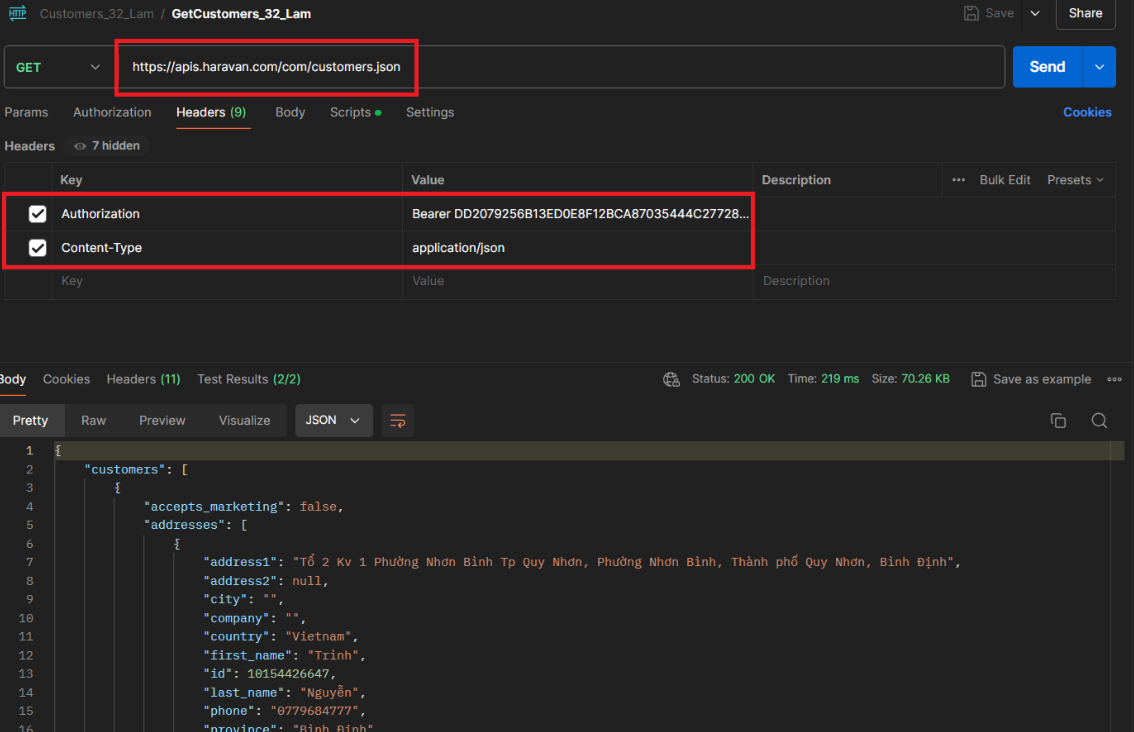
## Thực hiện test API

### Chức năng 1: Tạo một collection có tên là Customer\_32\_Lam



#### Tạo request: GET GetCustomers\_32\_Lam

* URL: <https://apis.haravan.com/com/customers.json>
* Sau khi đặt đường dẫn thì nhấn nút Send để lấy dữ liệu khách hàng về



##### Test case 1:

pm.test("TC1\_32\_Lam\_Status code is 200", function () {

pm.response.to.have.status(200);

});

##### Test case 2:

pm.test("TC2\_32\_Lam\_Kiểm tra các thuộc tính cần thiết của tất cả khách hàng", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.customers).to.be.an("array");

jsonData.customers.forEach(function (customer) {

// Kiểm tra xem mỗi khách hàng có thuộc tính "id" hay không

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

// Kiểm tra xem mỗi khách hàng có thuộc tính "addresses" hay không

pm.expect(customer).to.have.property("addresses");

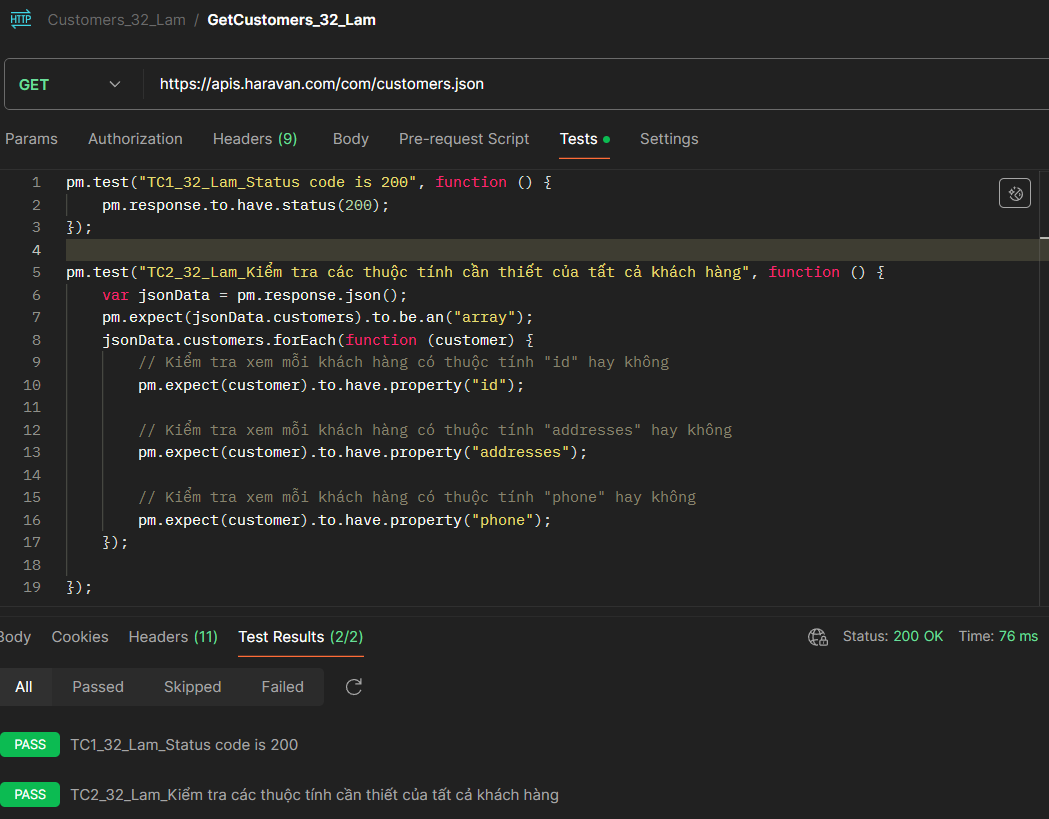
// Kiểm tra xem mỗi khách hàng có thuộc tính "phone" hay không

pm.expect(customer).to.have.property("phone");

});

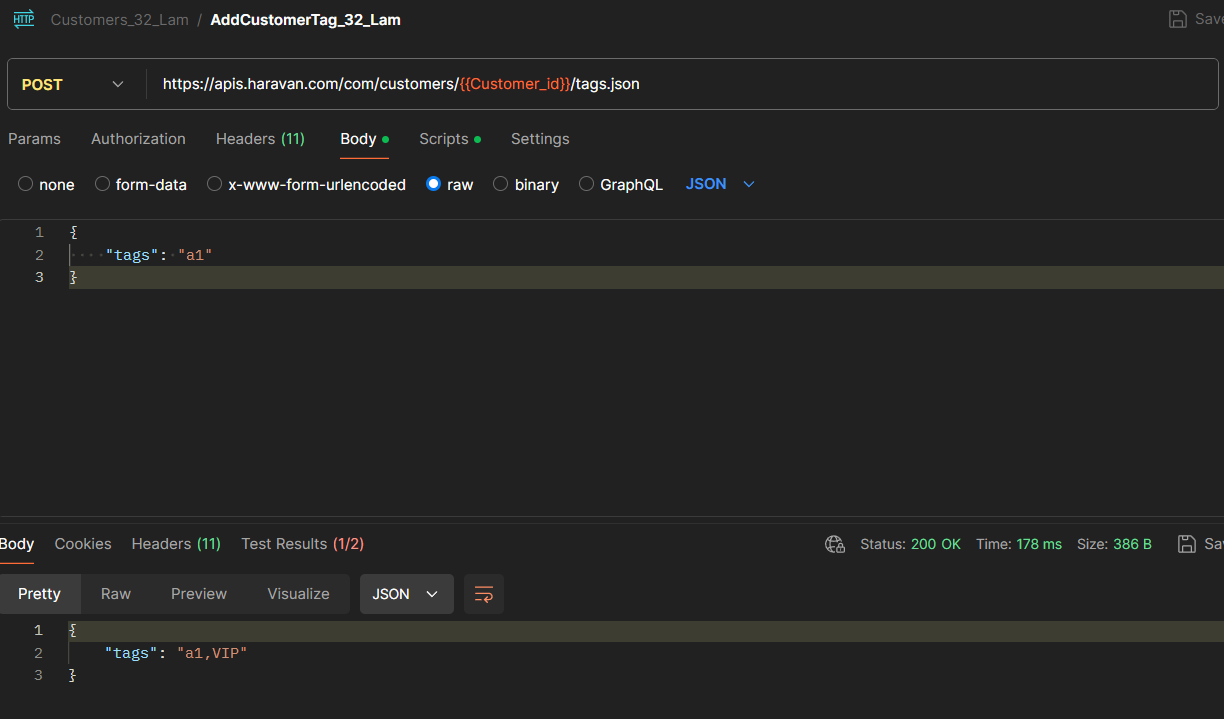
});

* Kết quả:



#### Tạo request: POST AddCustomer\_32\_Lam

* URL: [https://apis.haravan.com/com/customers/{{Customer\_id}}/tags.json](https://apis.haravan.com/com/customers/%7B%7BCustomer_id%7D%7D/tags.json)
* Sau khi tạo request POST, chúng ta vào phần Body -> raw -> điền thông tin tags muốn thêm vào khách hàng {{Customer\_id}} -> Send. Và thuộc tính tags đã được thêm vào 1 tag mới



##### Test case 1:

pm.test("TC1\_32\_Lam\_Kiểm tra trong JSON có chứa VIP,a1 không", function () {

pm.expect(pm.response.text()).to.include("a1,VIP");

});

##### Test case 2:

pm.test("TC2\_32\_Lam\_Successful POST request", function () {

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

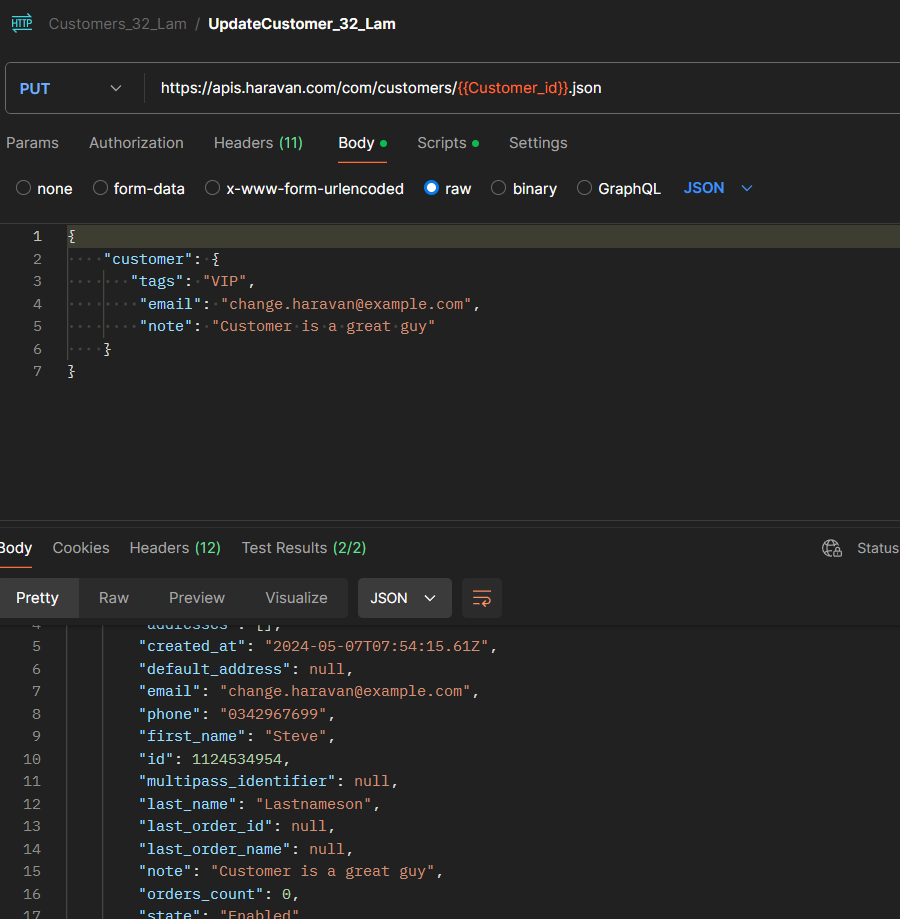
});

* Kết quả:

#### 

#### Tạo request: PUT UpdateCustomer\_32\_Lam

* [https://apis.haravan.com/com/customers/{{Customer\_id}}.json](https://apis.haravan.com/com/customers/%7B%7BCustomer_id%7D%7D.json)
* Vào phần Body -> raw -> nhập vào thuộc tính cần cập nhật của khách hàng {{Customer\_id}} như hình bên dưới



##### Test case 1:

pm.test("TC1\_32\_Lam\_checkEmail", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.customer.email).to.eql("change.haravan@example.com");

});

##### Test case 2:

pm.test("TC2\_32\_Lam\_checkNote", function () {

pm.expect(pm.response.text()).to.include("Customer is a great guy");

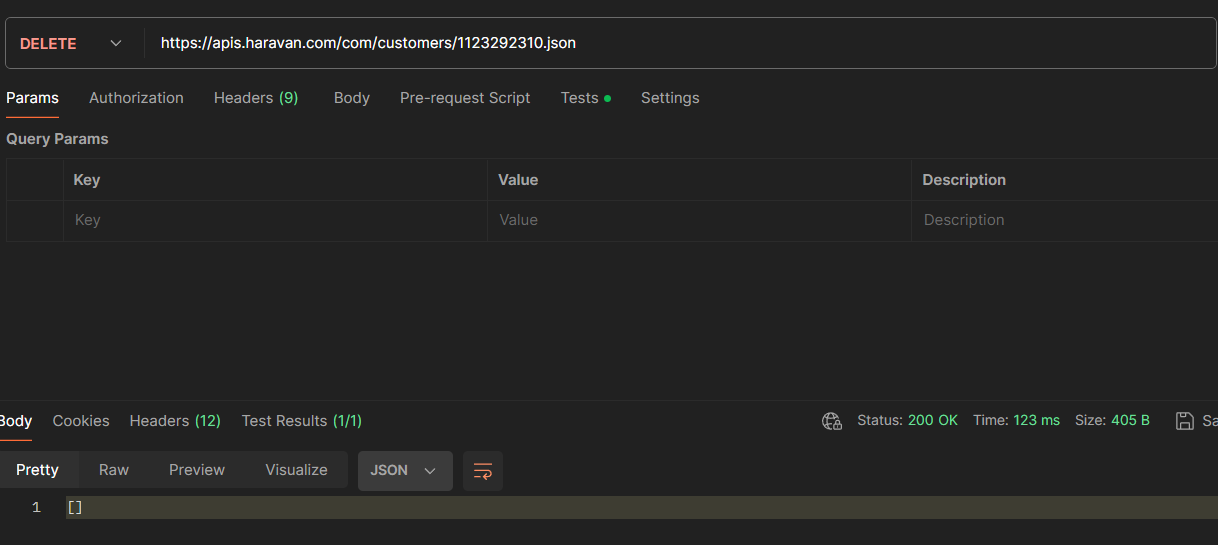
});

* Kết quả:

#### 

#### Tạo request: DELETE DeleteCustomer\_32\_Lam

* [https://apis.haravan.com/com/customers/{{Customer\_id}}.json](https://apis.haravan.com/com/customers/%7B%7BCustomer_id%7D%7D.json)
* Chỉ cần nhập url có {{Customer\_id}} và nhấn Send thì {{Customer\_id}} sẽ được xoá



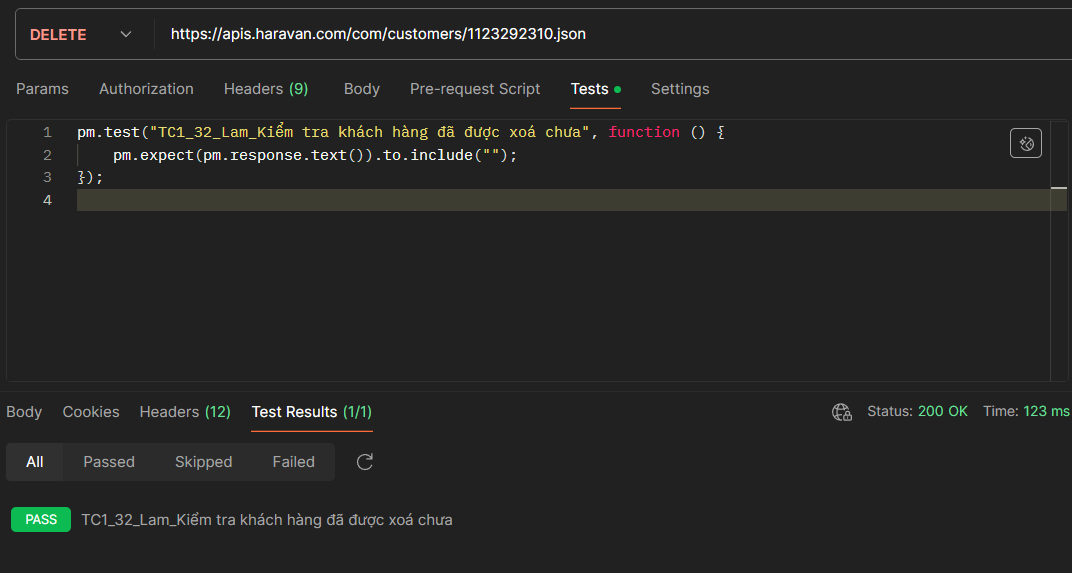
##### Test case 1:

pm.test("TC1\_32\_Lam\_Kiểm tra khách hàng đã được xoá chưa", function () {

pm.expect(pm.response.text()).to.include("");

});

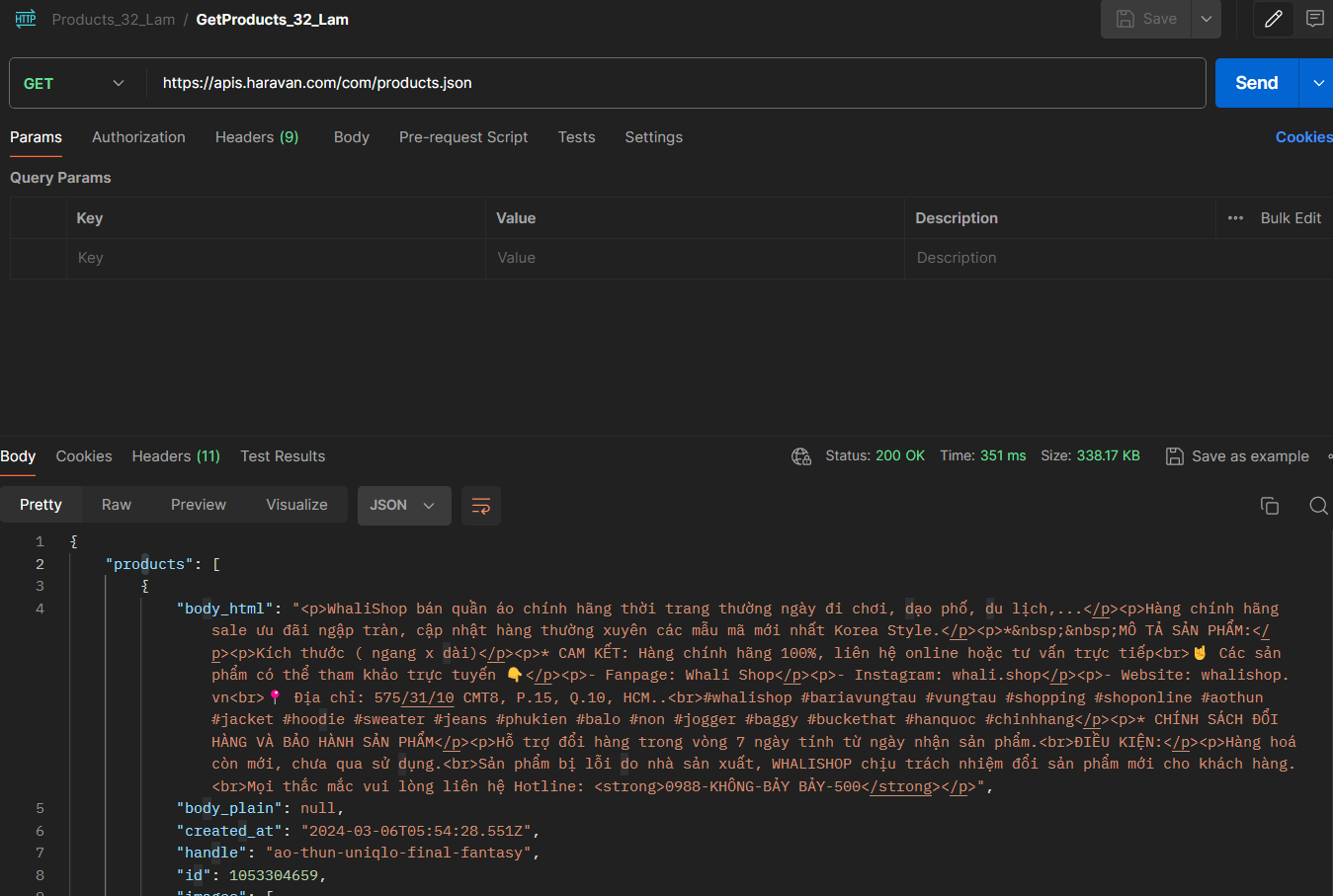
* Kết quả:



### Chức năng 2: Tạo một collection có tên là Products\_32\_Lam

#### Tạo request: GET GetProducts\_32\_Lam

* <https://apis.haravan.com/com/products.json>



##### Test case 1:

pm.test("TC1\_32\_Lam\_Kiểm tra số lượng sản phẩm trả về", function () {

var jsonData = pm.response.json();

// Kiểm tra xem có ít nhất 5 sản phẩm trong danh sách

pm.expect(jsonData.products.length).to.be.at.least(5);

});

##### Test case 2:

pm.test("TC2\_32\_Lam\_Kiểm tra các thuộc tính cần thiết của tất cả sản phẩm", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.products).to.be.an("array");

jsonData.products.forEach(function (product) {

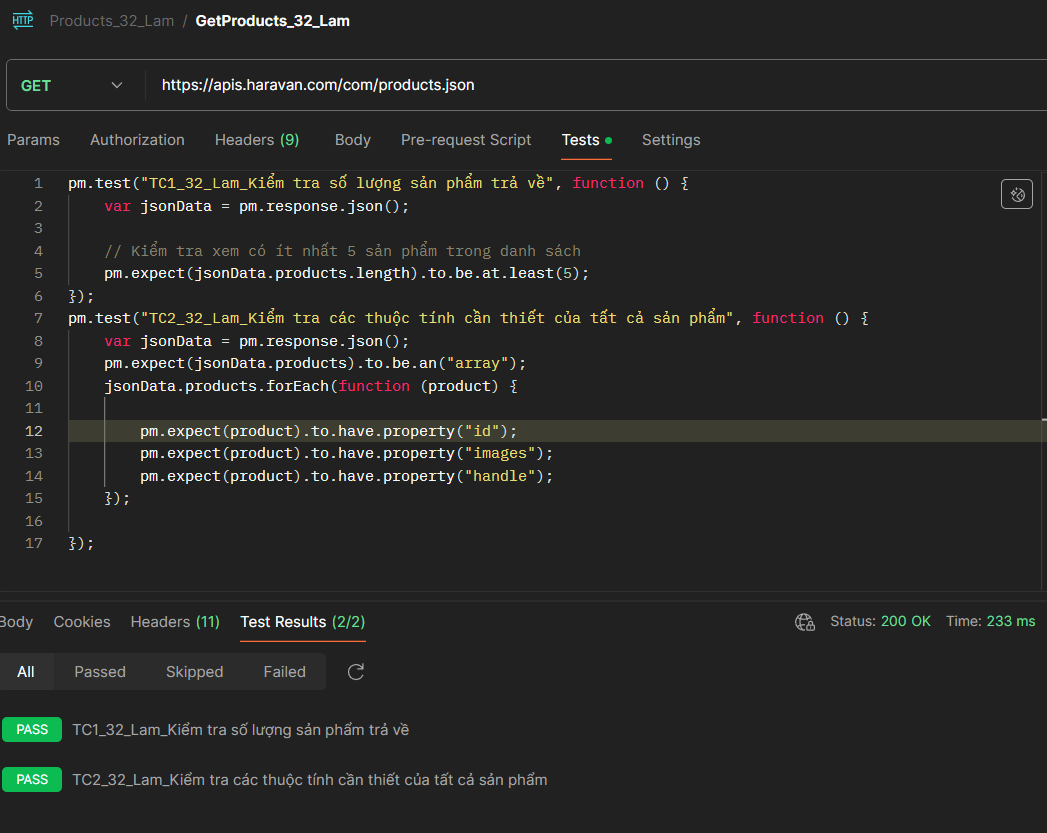
pm.expect(product).to.have.property("id");

pm.expect(product).to.have.property("images");

pm.expect(product).to.have.property("handle");

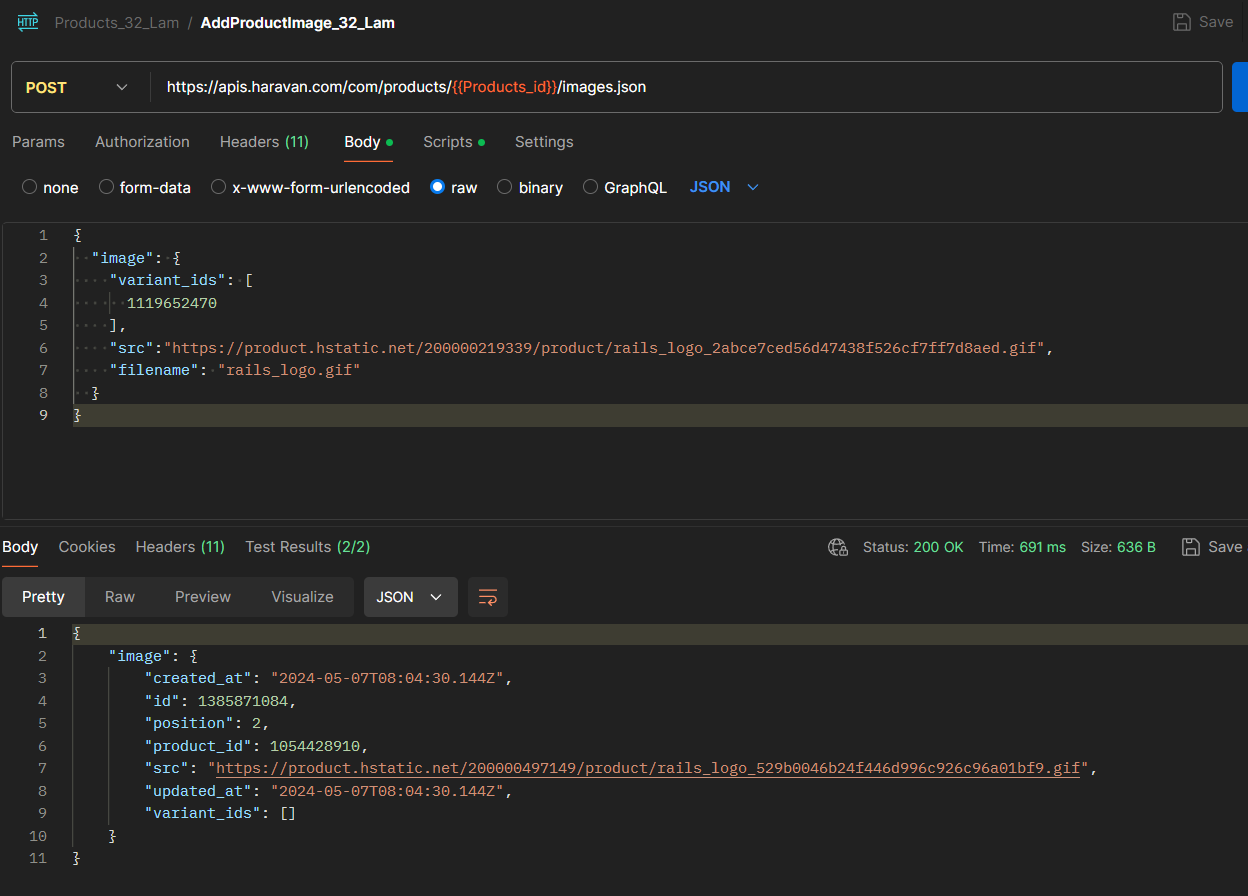
});

});

* Kết quả: 

#### Tạo request: POST AddProductImage\_32\_Lam

* [https://apis.haravan.com/com/products/{{Products\_id}}/images.json](https://apis.haravan.com/com/products/%7B%7BProducts_id%7D%7D/images.json)



##### Test case 1:

pm.test("TC1\_32\_Lam\_Kiểm tra trong JSON có chứa image không", function () {

pm.expect(pm.response.text()).to.include("image");

});

##### Test case 2:

pm.test("TC2\_32\_Lam\_checkIdProduct", *function* () {

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

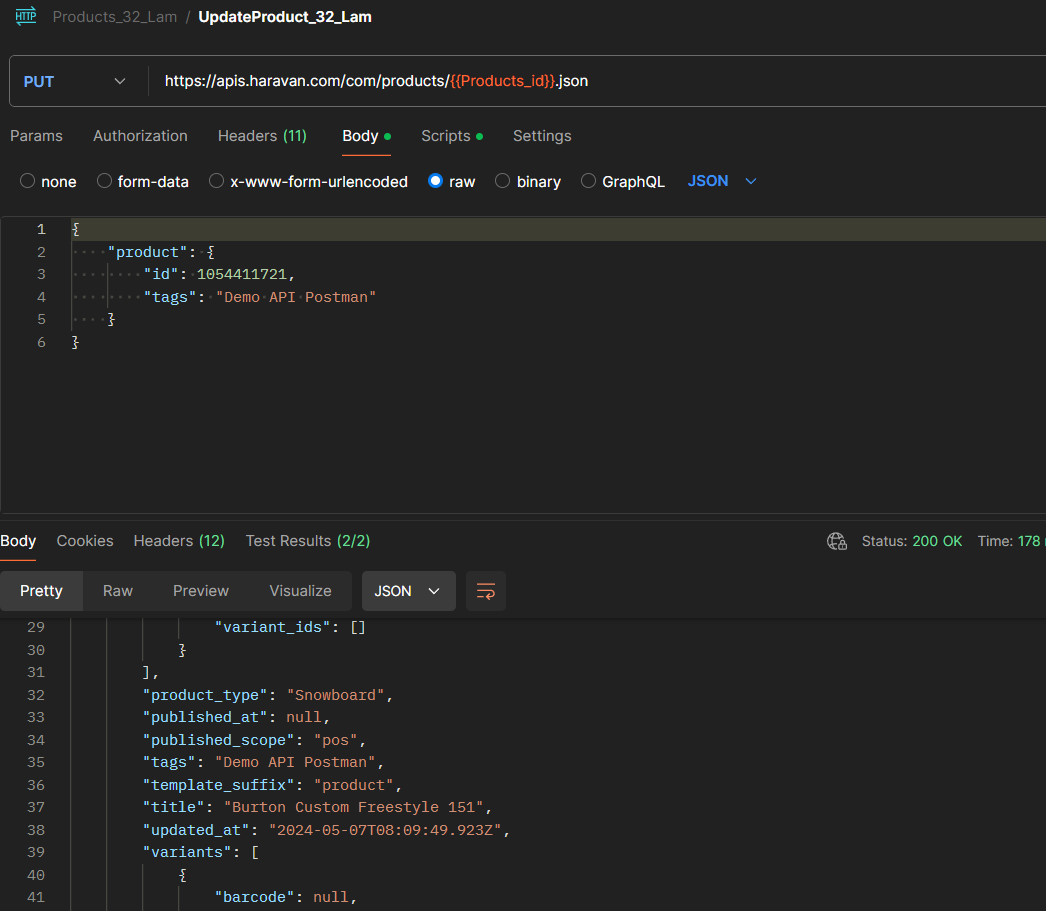
pm.expect(jsonData.image.product\_id).to.eql(1054428910 );

});

#### 

#### Tạo request: PUT UpdateProduct\_32\_Lam

* [https://apis.haravan.com/com/products/{{Products\_id}}.json](https://apis.haravan.com/com/products/%7B%7BProducts_id%7D%7D.json)



##### Test case 1:

pm.test("TC1\_32\_Lam\_checkId", *function* () {

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

pm.expect(jsonData.product.id).to.eql(1054428910);

});

##### Test case 2:

pm.test("TC2\_32\_Lam\_checkTag", function () {

pm.expect(pm.response.text()).to.include("Demo API Postman");

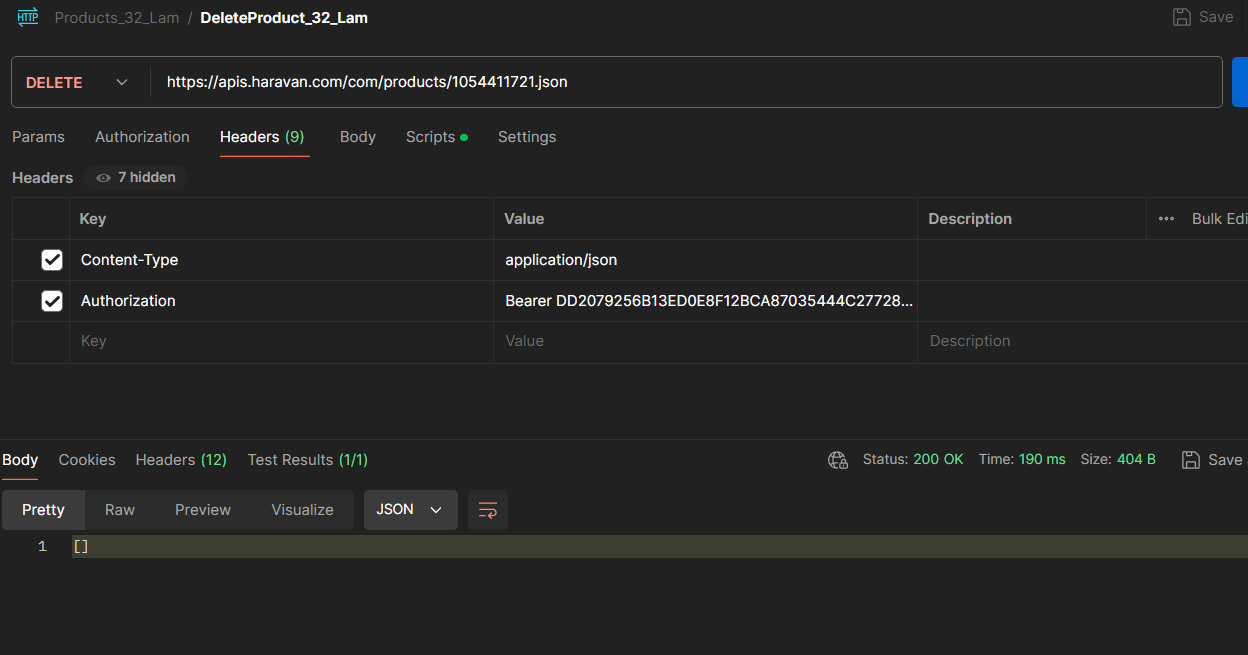
});

* Kết quả:

#### 

#### Tạo request: DELETE DeleteProduct\_32\_Lam

* [https://apis.haravan.com/com/products/{{Products\_id}}.json](https://apis.haravan.com/com/products/%7B%7BProducts_id%7D%7D.json)



##### Test case 1:

pm.test("TC1\_32\_Lam\_Kiểm tra sản phẩm đã được xoá chưa", *function* () {

pm.expect(pm.response.text()).to.include("");

});

* Kết quả:

