

TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH  
KHOA CÔNG NGHỆ THÔNG TIN

------



**BÁO CÁO ĐỀ TÀI MÔN HỌC**

**MÔN HỌC:**  
**KIỂM THỬ PHẦN MỀM**

**Giảng viên hướng dẫn:** Th. S Nguyễn Thị Ngọc Thanh

**Sinh viên thực hiện:** Nguyễn Lư Hồng Phương - 2251010077

**Khoa:** Công nghệ thông tin

**Lớp:** DH22CS01

**Khóa:** 2022-2026

**Thành phố Hồ Chí Minh tháng 3 năm 2025**



<b>I. KIỂM THỬ UNIT TEST VỚI NUNIT TÍNH GIÁ TRỊ ĐA THÚC TẠI MỘT GIÁ TRỊ X.</b>	<b>5</b>
<b>1. XÁC ĐỊNH YÊU CẦU CHỨC NĂNG.....</b>	<b>5</b>
1.1. Đặc tả chức năng “Tính giá trị đa thức tại một giá trị x”.....	5
<b>2. FORM CHỨC NĂNG.....</b>	<b>5</b>
2.1. Tao project “Tính giá trị đa thức tại một giá trị x”.....	5
2.2. Thực hiện chức năng “Tính giá trị đa thức tại một giá trị x”.....	7
<b>3. THIẾT KẾ TEST CASE.....</b>	<b>10</b>
<b>4. KIỂM THỬ CHỨC NĂNG.....</b>	<b>10</b>
4.1. TẠO UNIT TEST PROJECT.....	10
4.2. THÊM DỰ ÁN CẦN TEST VÀO FILE KIỂM THỬ.....	12
4.3. TẠO FILE EXCEL KIỂM THỬ.....	14
4.4. THÊM REFERENCE CHO UNIT TEST ĐỂ ĐỌC FILE EXCEL.....	14
4.5. THỰC HIỆN TEST CASE.....	15
4.6. KẾT QUẢ THỰC HIỆN .....	17
<b>II. KIỂM THỬ SELENIUM VỚI TRANG WEB KFC VIETNAM .....</b>	<b>17</b>
<b>1. SELENIUM.....</b>	<b>18</b>
1.1. Cài đặt trình duyệt FireFox.....	18
1.2. Cài đặt Selenium IDE trên trình duyệt FireFox.....	19
1.3. Cài đặt Selenium WebDriver.....	21
1.3.1. Tao project tên 11_phuong_webdriver trong solution ‘11_phuong_polynominal ‘ .....	21
1.3.2. Cài đặt Selenium .....	22
<b>2. GIỚI THIỆU TRANG WEB KFC VIETNAM.....</b>	<b>23</b>
<b>3. XÁC ĐỊNH YÊU CẦU CHỨC NĂNG.....</b>	<b>24</b>
3.1. Đặc tả chức năng “Đăng ký tài khoản”.....	24
3.2. Đặc tả chức năng “Đăng nhập tài khoản”.....	25
<b>4. THIẾT KẾ TEST CASE .....</b>	<b>25</b>
4.1. Thiết kế test case dựa trên đặc tả chức năng “Đăng ký tài khoản”.....	25
4.2. Thiết kế test case dựa trên đặc tả chức năng “Đăng nhập tài khoản”.....	26
<b>5. KIỂM THỬ CHỨC NĂNG.....</b>	<b>26</b>
5.1. Chuẩn bị trước khi kiểm thử .....	26
5.2. Kiểm thử chức năng “Đăng ký tài khoản” .....	27
5.2.1. Lấy các trường thông tin của chức năng đăng ký tài khoản.....	27
5.2.2. Thực hiện TestCase_DangKy_11_phuong.cs.....	29
5.3. Kiểm thử chức năng “Đăng nhập tài khoản”.....	34
5.3.1. Lấy các trường thông tin của chức năng đăng nhập tài khoản.....	34
5.3.2. Thực hiện TestCase_DangNhap_11_phuong.cs .....	35
<b>III. KIỂM THỬ API POSTMAN VỚI JSONPLACEHOLDER .....</b>	<b>38</b>
<b>1. THIẾT LẬP POSTMAN TRÊN WEB.....</b>	<b>38</b>
1.1. Tao WorkSpace trên Postman .....	38
1.2. Tao một request mới trên Postman trong workspace phuong_11_TestAPI. ....	40
<b>2. GIỚI THIỆU VỀ JSONPLACEHOLDER .....</b>	<b>40</b>
<b>3. KIỂM THỬ API POSTMAN VỚI JSONPLACEHOLDER.....</b>	<b>41</b>
3.1. Tao và lưu một request: get.....	41
3.2. Tao và lưu một request: post.....	44
3.3. Tao và lưu một request: put.....	46
3.4. Tao và lưu một request: delete. ....	48



## I. Kiểm thử Unit Test với NUnit tính giá trị đa thức tại một giá trị x.

### 1. Xác định yêu cầu chức năng

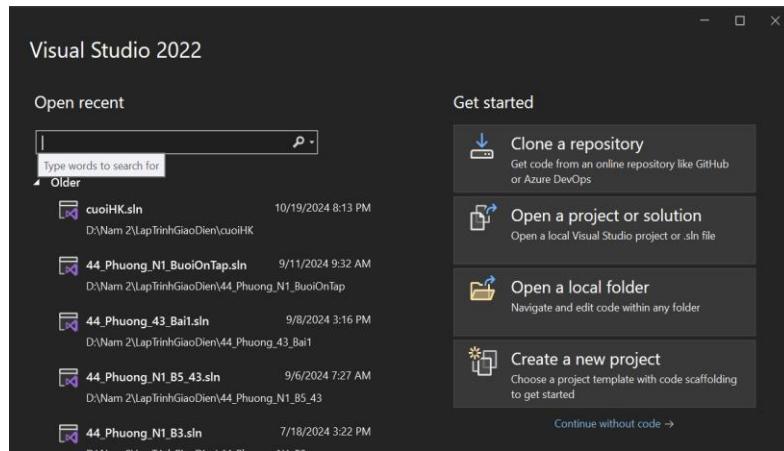
#### 1.1. Đặc tả chức năng “Tính giá trị đa thức tại một giá trị x”.

- Tên chức năng: Tính giá trị đa thức tại một giá trị x.
  - Mô tả: Chức năng cho phép người dùng nhập n là giá trị của bậc đa thức, các hệ số của đa thức và giá trị x, để tính giá trị đa thức tại một giá trị x.
  - Luồng hoạt động:
    - + Bước 1: Người dùng nhập vào một số nguyên n là bậc của đa thức với điều kiện  $n > 0$ .
    - + Bước 2: Người dùng nhập vào  $n + 1$  số nguyên  $a_i$ , trong đó  $a_i$  là hệ số của  $x^i$  với i chạy từ 0 đến n, các hệ số được nhập cách nhau bằng dấu phẩy.
    - + Bước 3: Người dùng nhập một số nguyên x là giá trị cần tính đa thức.
    - + Bước 4: Người dùng nhấn nút tính giá trị đa thức.
    - + Bước 5: Hệ thống tính giá trị đa thức theo công thức:
$$P(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$
  - + Bước 6: Hệ thống hiển thị kết quả giá trị của đa thức tại x ở ô kết quả.
- Luồng thay thế:
    - + Bước 4.1: Nếu người dùng nhập giá trị n, mang hệ số, hoặc giá trị x rỗng thì hệ thống hiển thị cảnh báo “Không được để trống dữ liệu”.
    - + Bước 4.2: Nếu giá trị bậc n không phải là số nguyên hoặc là số nguyên nhỏ hơn 0, hệ thống hiển thị cảnh báo “Bậc đa thức phải là số nguyên dương”.
    - + Bước 4.3: Nếu hệ số phân tách khác ngoài dấu phẩy, hệ thống hiển thị cảnh báo “Hệ số phải được nhập cách nhau bằng dấu phẩy và là số nguyên”.
    - + Bước 4.4: Nếu danh sách hệ số chứa hệ số rỗng hoặc định dạng sai, hệ thống hiển thị cảnh báo “Danh sách có hệ số rỗng hoặc định dạng sai”.
    - + Bước 4.5: Nếu danh sách hệ số nhập nhỏ hơn  $n + 1$ , hệ thống hiển thị cảnh báo “Số lượng hệ số không hợp lệ”.
    - + Bước 4.5: Nếu danh sách hệ số chứa hệ số rỗng hoặc định dạng sai, hệ thống hiển thị cảnh báo “Danh sách có hệ số rỗng hoặc định dạng sai”.
    - + Bước 4.6: Nếu x không phải là số nguyên, hệ thống hiển thị cảnh báo “Giá trị x phải là số nguyên”.

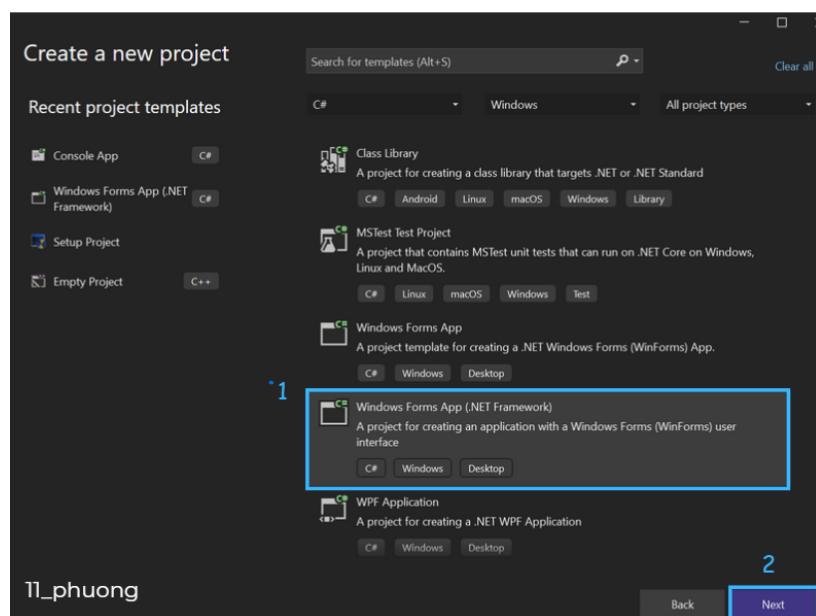
### 2. Form chức năng.

#### 2.1. Tạo project “Tính giá trị đa thức tại một giá trị x”.

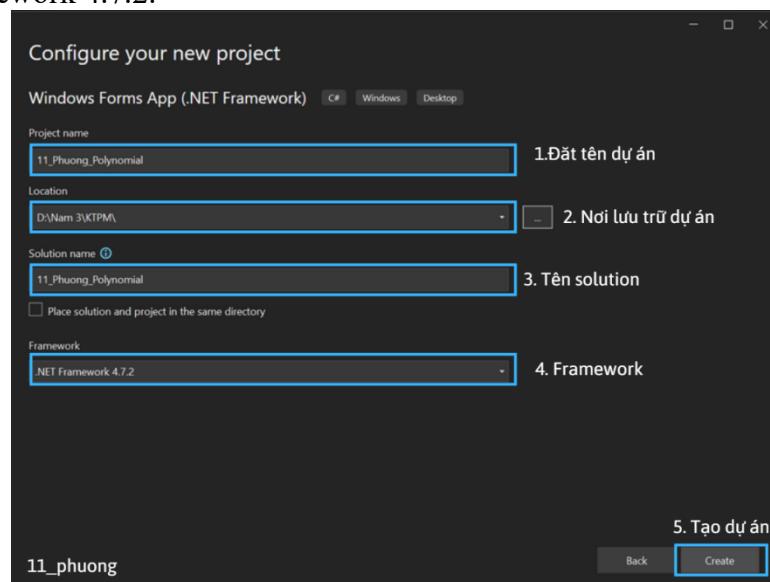
2.1.1. Bước 1: Mở Visual Studio 2022 -> chọn Create a new project.



### 2.1.2. Bước 2: Chọn Windows Forms App(.NET Framework) -> Next.

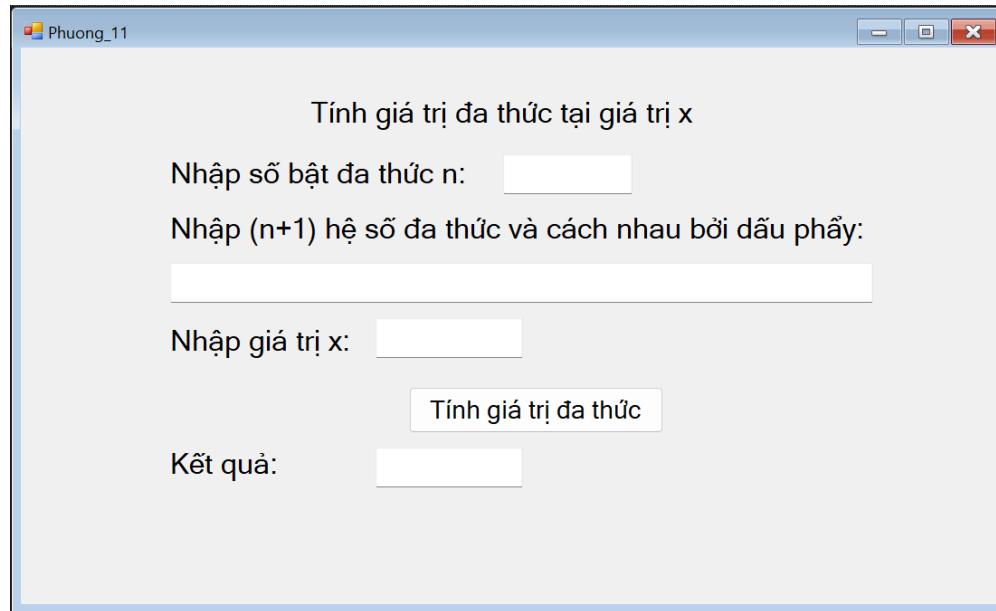


### 2.1.3. Bước 3: Đặt tên dự án “11\_Phuong\_Polynomial” tại nơi “D:\Nam 3\KTPM\” với phiên bản framework 4.7.2.



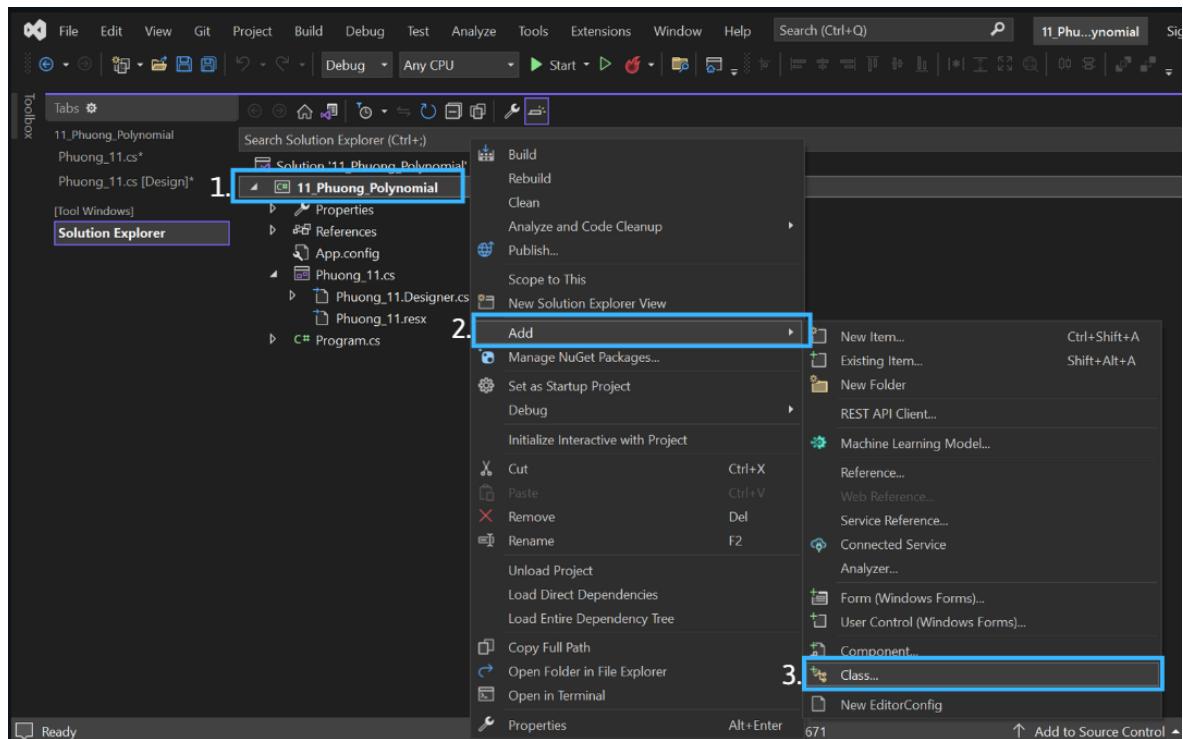
## 2.2. Thực hiện chức năng “Tính giá trị đa thức tại một giá trị x”.

### 2.2.1. Bước 1: Tạo giao diện “Tính giá trị đa thức tại một giá trị x”.

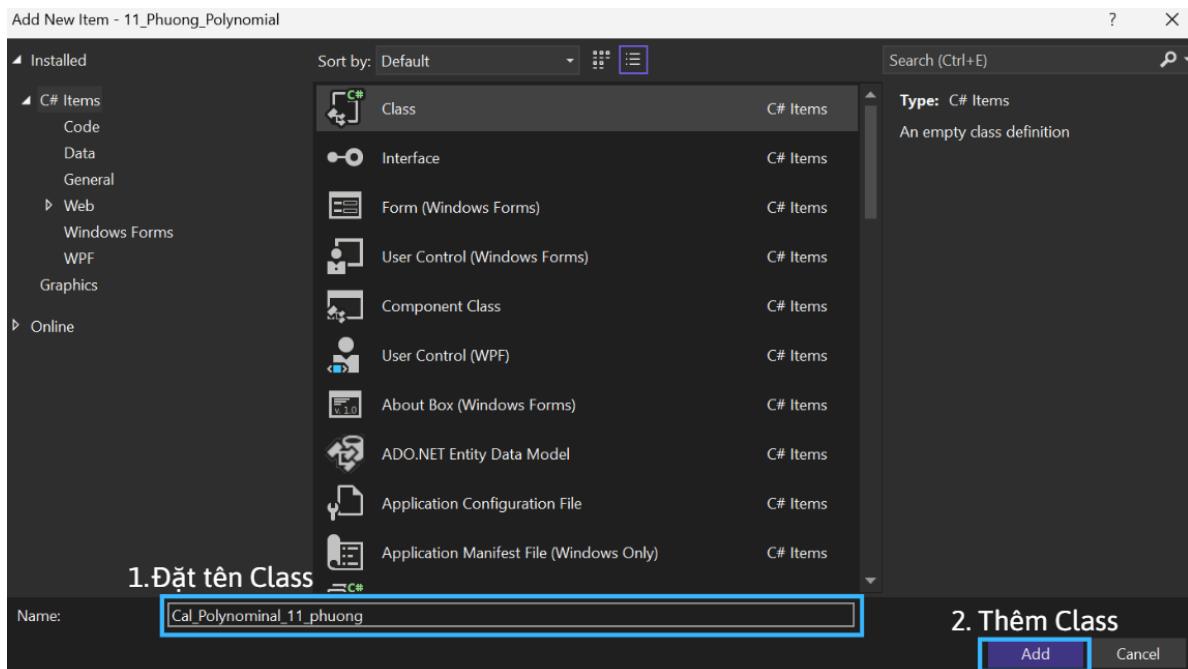


### 2.2.2. Bước 2: Tạo class tên “Cal\_Polynomial\_11\_phuong.cs” để tạo lớp tính giá trị đa thức.

#### 2.2.2.1. Ta vào Solution Explorer -> Nhấn chuột phải trên tệp “11\_Phuong\_Polynomial” -> Add -> Class...



#### 2.2.2.2. Đặt tên Class “Cal\_Polynomial\_11\_phuong.cs” -> Nhấn Add.



### 2.2.3. Bước 3: Souce code lớp Cal\_Polynomial\_11\_phuong.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace _11_Phương_Polynomial
{
    public class Cal_Polynomial_11_phuong
    {
        private int n_11_phuong;
        private List<int> a_11_phuong;
        public Cal_Polynomial_11_phuong(string num_11_phuong, string a_11_phuong)
        {
            //Bước 4.1:
            if (string.IsNullOrWhiteSpace(num_11_phuong) || string.IsNullOrWhiteSpace(a_11_phuong))
            {
                throw new ArgumentException("Không được để trống dữ liệu");
            }
            //Bước 4.2:
            if (!int.TryParse(num_11_phuong, out n_11_phuong) || n_11_phuong <= 0)
            {
                throw new ArgumentException("Bậc đa thức phải là số nguyên dương");
            }

            this.a_11_phuong = ParseCoeffs_11_phuong(a_11_phuong);
        }

        // Phương thức xử lý hệ số người dùng nhập
        private List<int> ParseCoeffs_11_phuong(string input_11_phuong)
        {
            input_11_phuong = input_11_phuong.Trim().Trim(',');
            // Bước 4.3:
            if (input_11_phuong.Any(c_11_phuong => !char.IsDigit(c_11_phuong) && c_11_phuong != ' ' && c_11_phuong != ',')) 
            {
                throw new ArgumentException("Hệ số phải được nhập cách nhau bằng dấu phẩy và là số nguyên");
            }
            string[] parts_11_phuong = input_11_phuong.Split(',');
            //Bước 4.4: Hệ số rỗng hoặc định dạng sai
            if (parts_11_phuong.Any(p =>
            {
                string trimmedPart = p.Trim();
                return string.IsNullOrEmpty(trimmedPart) || trimmedPart == " " || trimmedPart.EndsWith("-");
            })))
            {
                throw new ArgumentException("Danh sách có hệ số rỗng hoặc định dạng sai");
            }
        }
    }
}

```

```

//Bước 4.5:
if (parts_11_phuong.Length != n_11_phuong + 1)
{
    throw new ArgumentException("Số lượng hệ số không hợp lệ");
}
List<int> coeffList_11_phuong = new List<int>();
foreach (string part_11_phuong in parts_11_phuong)
{
    coeffList_11_phuong.Add(int.Parse(part_11_phuong.Trim()));
}
return coeffList_11_phuong;
}
public double Execute_11_phuong(string xInput_11_phuong)
{
    //Bước 4.1:
    if (string.IsNullOrWhiteSpace(xInput_11_phuong))
    {
        throw new ArgumentException("Không được để trống dữ liệu");
    }
    //Bước 4.6:
    if (!int.TryParse(xInput_11_phuong, out int x_11_phuong))
    {
        throw new ArgumentException("Giá trị x phải là số nguyên");
    }

    double result_11_phuong = 0;
    for (int i_11_phuong = 0; i_11_phuong <= n_11_phuong; i_11_phuong++)
    {
        result_11_phuong += a_11_phuong[i_11_phuong] * Math.Pow(x_11_phuong, i_11_phuong);
    }
    return result_11_phuong;
}
}
}
    
```

#### 2.2.4. Bước 4: Source code winform Phuong\_11.

```

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 _11_Phuong_Polynomial
{
    public partial class Phuong_11 : Form
    {
        public Phuong_11()
        {
            InitializeComponent();
        }

        private void Phuong_11_Load(object sender, EventArgs e)
        {

        }

        private void caculate_11_phuong_Click(object sender, EventArgs e)
        {
            try
            {
                string num_11_phuong = n_11_phuong.Text.ToString();
                string array_11_phuong = a_11_phuong.Text.ToString();
                Cal_Polynomial_11_phuong poly_11_phuong = new Cal_Polynomial_11_phuong(num_11_phuong, array_11_phuong);
                double r_11_phuong = poly_11_phuong.Execute_11_phuong(x_11_phuong.Text.ToString());
                result_11_phuong.Text = r_11_phuong.ToString();
            }
            catch (ArgumentException ex)
            {
                MessageBox.Show(ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
        }
    }
}
    
```

```

        }
        catch (Exception ex)
        {
            MessageBox.Show("Đã xảy ra lỗi không xác định: " + ex.Message, "Lỗi hệ thống", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        }
    }
}

```

### 3. Thiết kế test case.

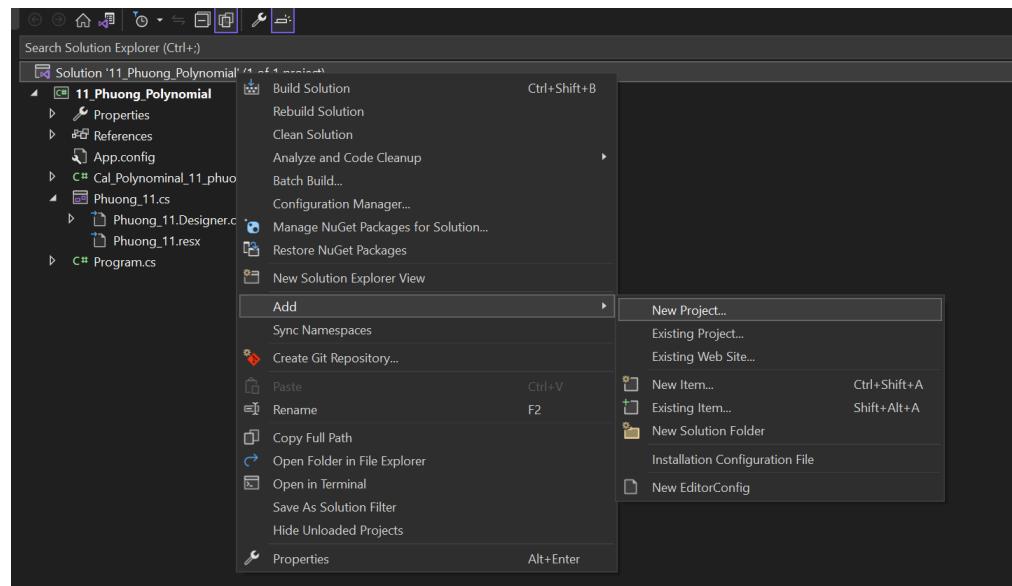
- Thiết kế test case dựa trên đặc tả chức năng “Tính giá trị đa thức tại giá trị x”. Ta được form test case.

STT	TestCase_11_phuong	n_11_phuong	a_11_phuong	x_11_phuong	KQ_MongDoi_11_phuong	KQ_ThucTe_11_phuong	Ảnh chụp màn hình	KQ_Test_11_phuong
1	TestCase1_NhapRong_11_Phương				Không được để trống dữ liệu	Không được để trống dữ liệu		pass
2	TestCase2_BacDaThucKhongHopLe_11_Phương	-4	1,2,3	1	Bậc đa thức phải là số nguyên dương	Bậc đa thức phải là số nguyên dương		pass
3	TestCase3_DinhDangHeSoSai_11_Phương	2	1,2,3	1	Hệ số phải được nhập cách nhau bằng dấu phẩy và là số nguyên	Hệ số phải được nhập cách nhau bằng dấu phẩy và là số nguyên		pass
4	TestCase4_HeSoRongHoacSaiDinhDang_11_Phương	2	1,,3	1	Danh sách có hệ số rỗng hoặc định dạng sai	Danh sách có hệ số rỗng hoặc định dạng sai		pass
5	TestCase5_SoLuongHeSoSai_11_Phương	2	1,2,3,4	1	Số lượng hệ số không hợp lệ	Số lượng hệ số không hợp lệ		pass
6	TestCase6_XKhongHopLe_11_Phương	2	1,2,3	1,2	Giá trị x phải là số nguyên	Giá trị x phải là số nguyên		pass
7	TestCase7_TinhDaThucHopLe_11_Phương	2	1,2,3	2	17	17		pass

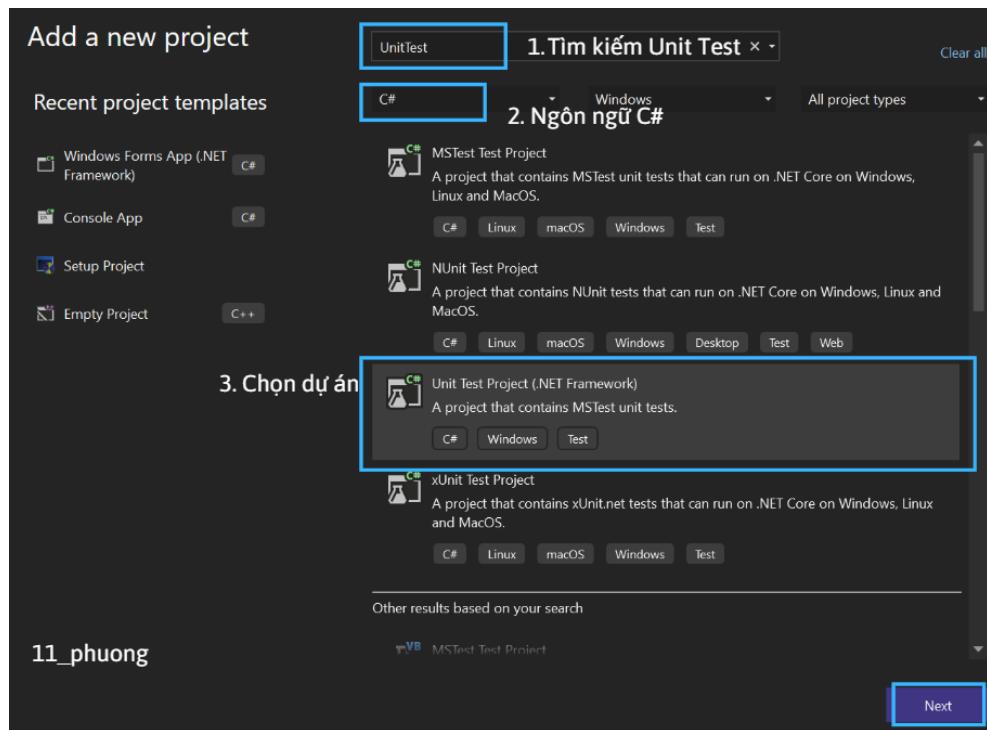
### 4. Kiểm thử chức năng.

#### 4.1. Tạo Unit Test project.

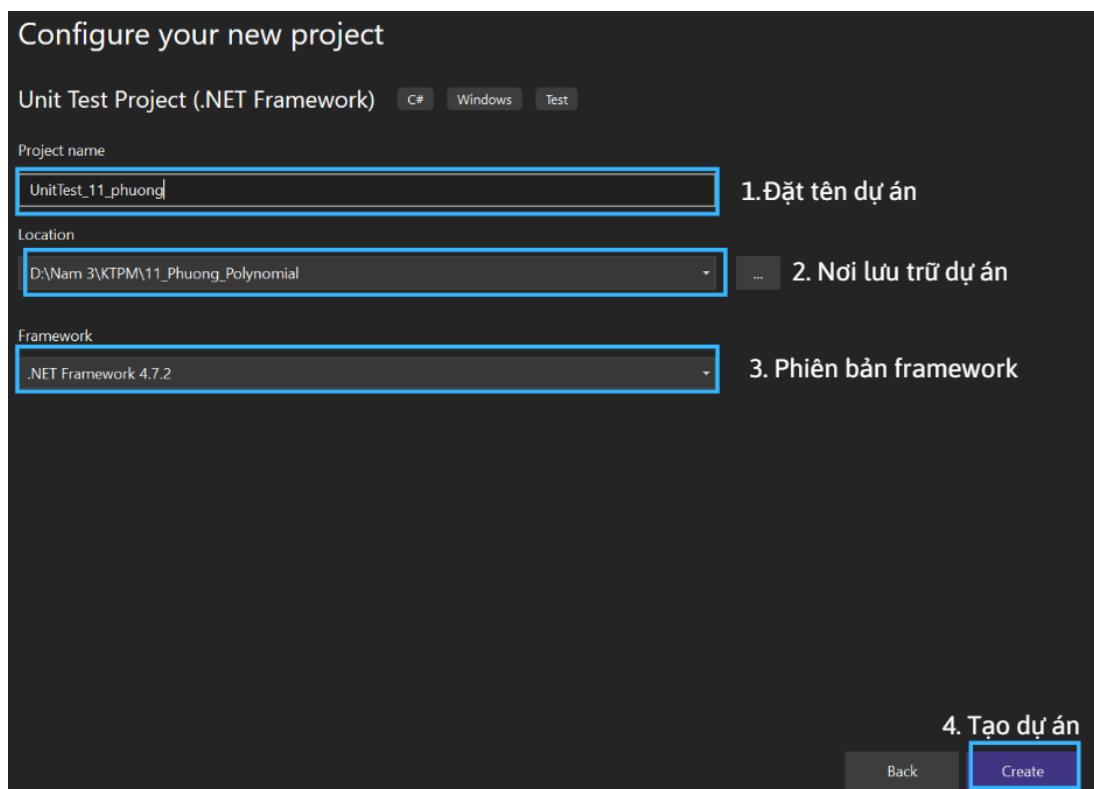
4.1.1. Bước 1: Ta vào Solution Explorer -> nhấn nút phải chuột Solution`11\_Phuong\_Polynomial` -> Add -> New Project...



#### 4.1.2. Bước 2: Tạo chọn Unit Test Project(.NET Framework) -> Next

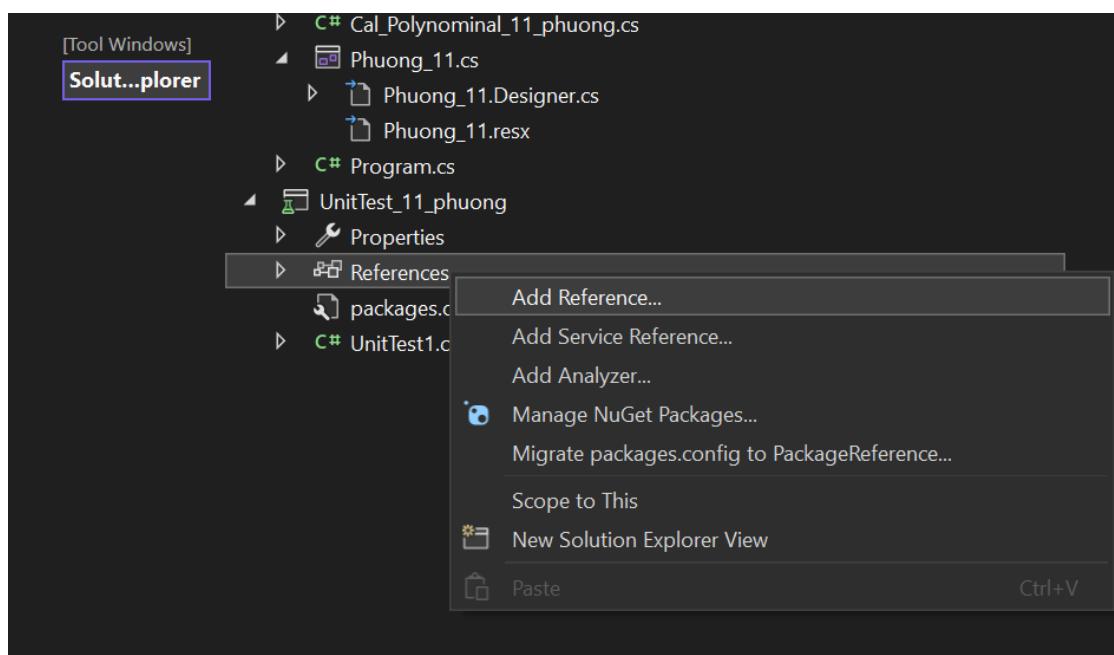


#### 4.1.3. Bước 3: Đặt tên dự án UnitTest\_11\_phuong -> create.

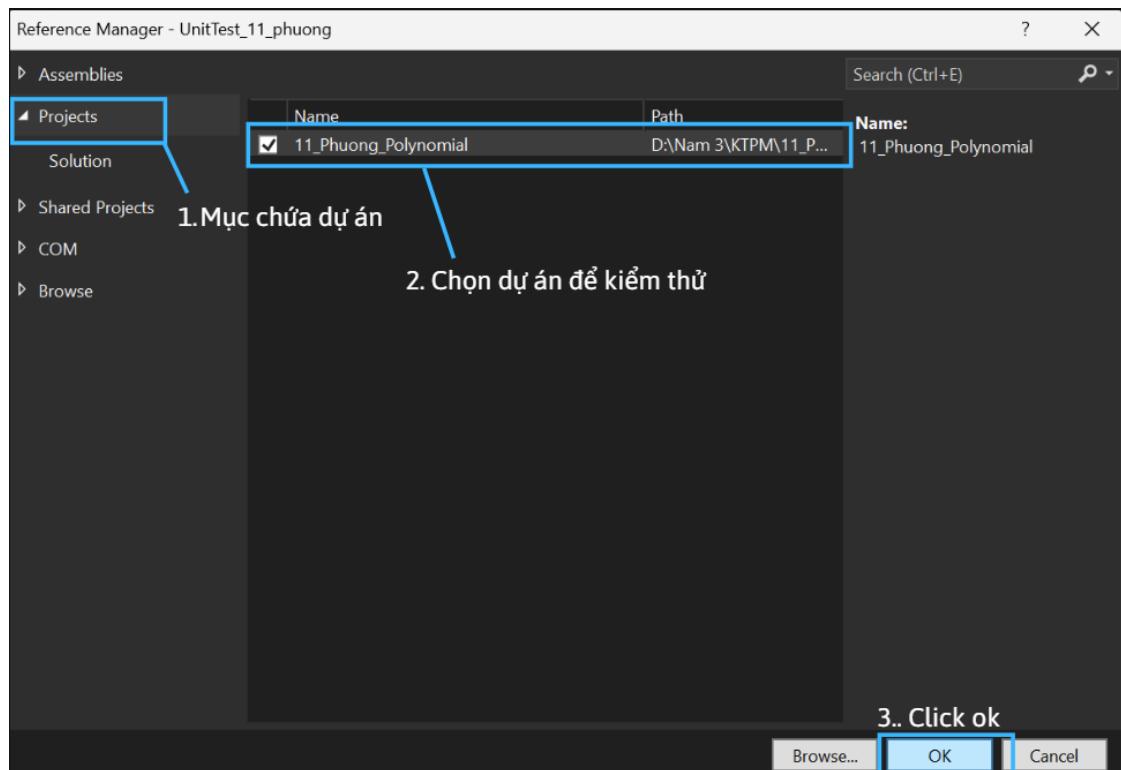


#### 4.2. Thêm dự án cần test vào file kiểm thử.

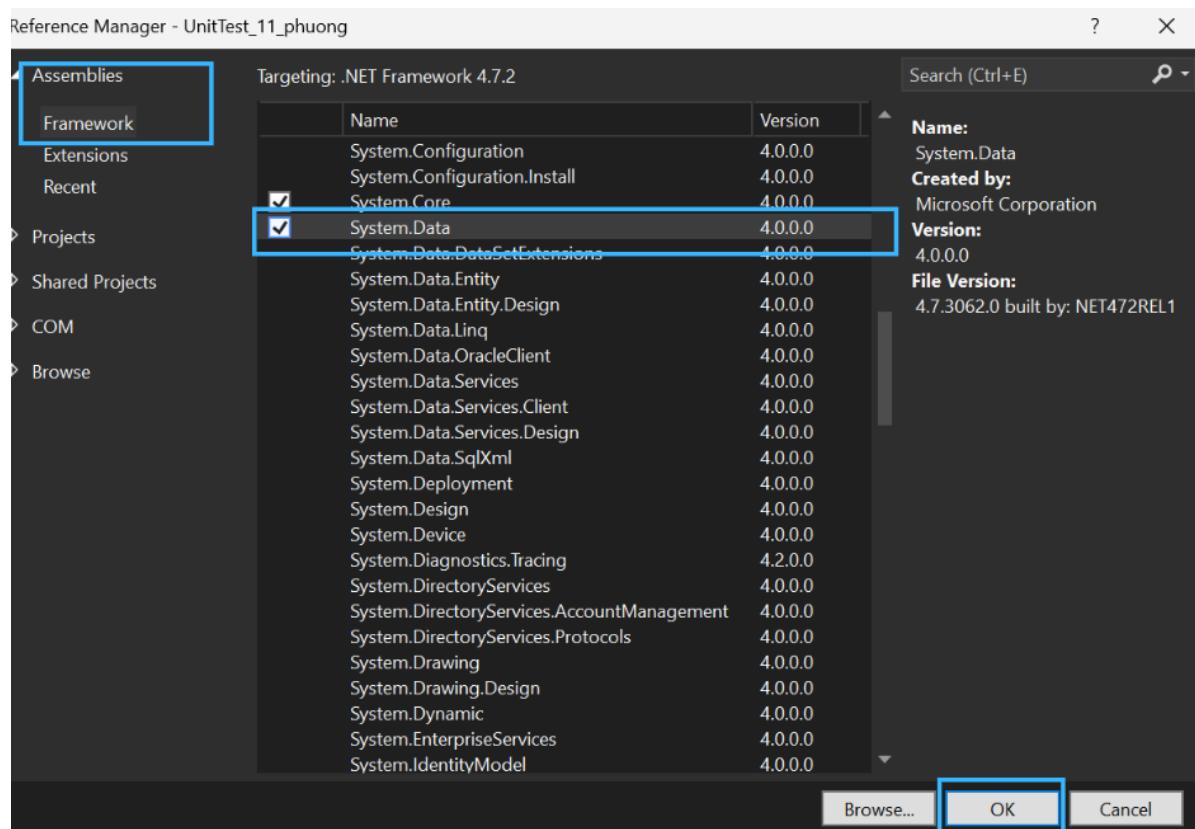
4.2.1. Bước 1: Ta vào References trong UnitTest\_11\_phuong -> nhấn chuột phải -> Add Reference...



4.2.2. Bước 2: Vào mục project -> click chọn dự án 11\_Phương\_Polynomial -> OK.

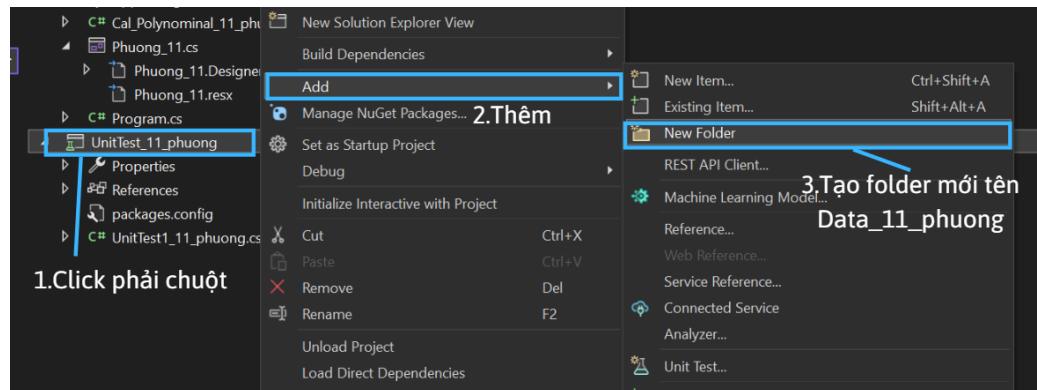


#### 4.2.3. Bước 3: Vào mục Assemblies -> Framework -> Thêm reference System.Data -> OK.

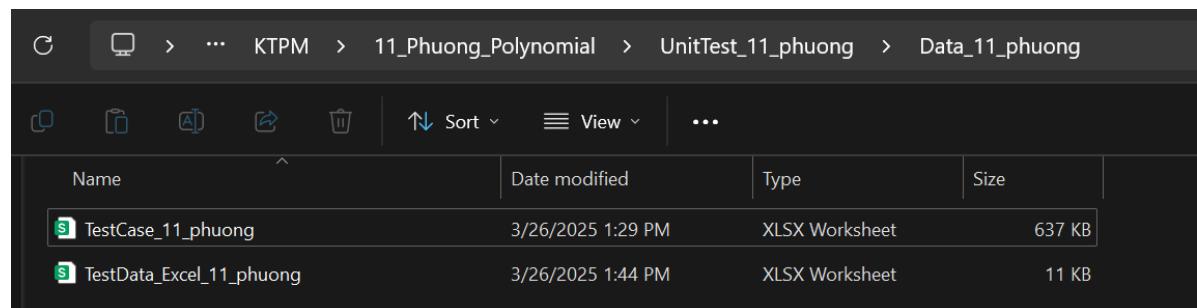


### 4.3. Tạo file excel kiểm thử.

#### 4.3.1. Bước 1: Tạo folder Data\_11\_phuong là nơi lưu trữ các tập dữ liệu test.

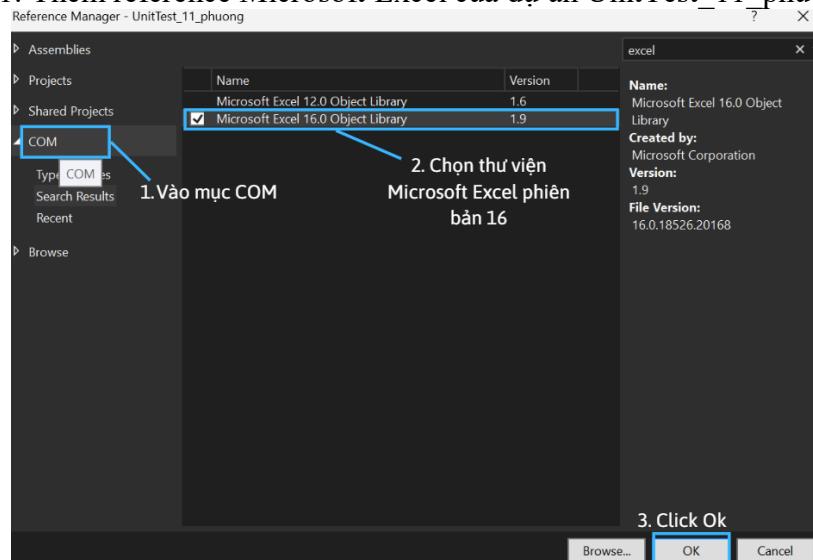


#### 4.3.2. Bước 2: Tạo file excel tên “TestData\_Excel\_11\_phuong” nằm trong folder Data.

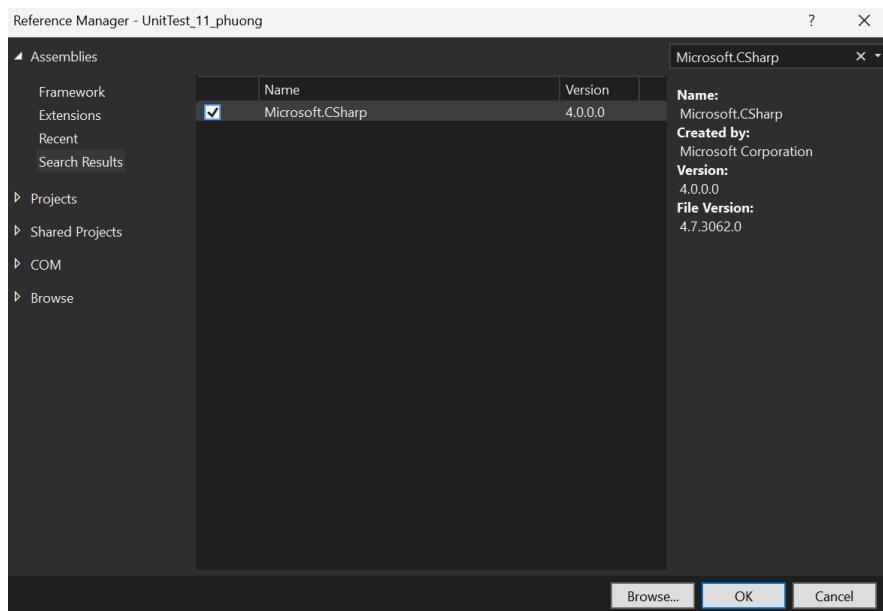


### 4.4. Thêm reference cho Unit Test để đọc file excel:

#### 4.4.1. Bước 1: Thêm reference Microsoft Excel của dự án UnitTest\_11\_phuong.



#### 4.4.2. Bước 2: Thêm thư viện Microsoft.Csharp -> ok



#### 4.5. Thực hiện test case.

- Source code thực hiện:

```

using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using _11_Phương_Polynomial;
using Microsoft.Office.Interop.Excel;
using System.IO;

namespace UnitTest_11_phuong
{
    [TestClass]
    public class UnitTest1_11_phuong
    {
        private Cal_Polynomial_11_phuong c_phuong_11;
        [TestInitialize]
        public void SetUp_11_phuong()
        {
            c_phuong_11 = new Cal_Polynomial_11_phuong("2", "1,2,3");
        }
        //TestCase1_NhapRong_11_Phương: n = "", a = "", x = "1" -> kq="Không được để trống dữ liệu"
        [TestMethod]
        public void TestCase1_NhapRong_11_Phuong()
        {
            string expected_11_phuong, actual_11_phuong;
            expected_11_phuong = "Không được để trống dữ liệu";
            try
            {
                c_phuong_11 = new Cal_Polynomial_11_phuong("", "");
                double result_11_phuong = c_phuong_11.Execute_11_phuong("1");
                actual_11_phuong = result_11_phuong.ToString();
            }
            catch (ArgumentException ex)
            {
                actual_11_phuong = ex.Message;
            }
            Assert.AreEqual(expected_11_phuong, actual_11_phuong);
        }
        //TestCase2_BacDaThucKhongHopLe_11_Phương: n = "a", a = "1,2,3", x = "1" -> kq="Bậc đa thức phải là số nguyên dương"
        [TestMethod]
        public void TestCase2_BacDaThucKhongHopLe_11_Phuong()
        {
            string expected_11_phuong, actual_11_phuong;
            expected_11_phuong = "Bậc đa thức phải là số nguyên dương";
            try
            {
                c_phuong_11 = new Cal_Polynomial_11_phuong("a", "1,2,3");
                double result_11_phuong = c_phuong_11.Execute_11_phuong("1");
                actual_11_phuong = result_11_phuong.ToString();
            }
            catch (ArgumentException ex)
            {
                actual_11_phuong = ex.Message;
            }
            Assert.AreEqual(expected_11_phuong, actual_11_phuong);
        }
    }
}

```

```

}

//TestCase3_DinhDangHeSoSai_11_Phuong n = "2", a = "1 2 ,3", x = "1" -> kq="Hệ số phải được nhập cách nhau bằng dấu phẩy và là số nguyên"
[TestMethod]
public void TestCase3_DinhDangHeSoSai_11_Phuong()
{
    string expected_11_phuong, actual_11_phuong;
    expected_11_phuong = "Hệ số phải được nhập cách nhau bằng dấu phẩy và là số nguyên";
    try
    {
        c_phuong_11 = new Cal_Polynomial_11_phuong("2", "1 2 ,3");
        double result_11_phuong = c_phuong_11.Execute_11_phuong("1");
        actual_11_phuong = result_11_phuong.ToString();
    }
    catch (ArgumentException ex)
    {
        actual_11_phuong = ex.Message;
    }
    Assert.AreEqual(expected_11_phuong, actual_11_phuong);
}

//TestCase4_HeSoRongHoacSaiDinhDang_11_Phuong n = "2", a = "1,3-,3", x = "1" -> kq = "Danh sách có hệ số rỗng hoặc định dạng sai"
[TestMethod]
public void TestCase4_HeSoRongHoacSaiDinhDang_11_Phuong()
{
    string expected_11_phuong, actual_11_phuong;
    expected_11_phuong = "Danh sách có hệ số rỗng hoặc định dạng sai";
    try
    {
        c_phuong_11 = new Cal_Polynomial_11_phuong("2", "1,3-,3");
        double result_11_phuong = c_phuong_11.Execute_11_phuong("1");
        actual_11_phuong = result_11_phuong.ToString();
    }
    catch (ArgumentException ex)
    {
        actual_11_phuong = ex.Message;
    }
    Assert.AreEqual(expected_11_phuong, actual_11_phuong);
}

//TestCase5_SoLuongHeSoSai_11_Phuong n = "2", a = "1,2", x = "1" -> kq="Số lượng hệ số không hợp lệ"
[TestMethod]
public void TestCase5_SoLuongHeSoSai_11_Phuong()
{
    string expected_11_phuong, actual_11_phuong;
    expected_11_phuong = "Số lượng hệ số không hợp lệ";
    try
    {
        c_phuong_11 = new Cal_Polynomial_11_phuong("2", "1,2");
        double result_11_phuong = c_phuong_11.Execute_11_phuong("1");
        actual_11_phuong = result_11_phuong.ToString();
    }
    catch (ArgumentException ex)
    {
        actual_11_phuong = ex.Message;
    }
    Assert.AreEqual(expected_11_phuong, actual_11_phuong);
}

//TestCase6_XKhongHopLe_11_Phuong n = "2", a = "1,2,3", x = "1.2" -> kq = "Giá trị x phải là số nguyên"
[TestMethod]
public void TestCase6_XKhongHopLe_11_Phuong()
{
    string expected_11_phuong, actual_11_phuong;
    expected_11_phuong = "Giá trị x phải là số nguyên";
    try
    {
        c_phuong_11 = new Cal_Polynomial_11_phuong("2", "1,2,3");
        double result_11_phuong = c_phuong_11.Execute_11_phuong("1.2");
        actual_11_phuong = result_11_phuong.ToString();
    }
    catch (ArgumentException ex)
    {
        actual_11_phuong = ex.Message;
    }
    Assert.AreEqual(expected_11_phuong, actual_11_phuong);
}

//TestCase7_TinhDaThucHopLe_11_Phuong n = "2", a = "3,4,5", x = "2" -> kq = "31"
[TestMethod]
public void TestCase7_TinhDaThucHopLe_11_Phuong()
{
    string expected_11_phuong, actual_11_phuong;
    expected_11_phuong = "31";
    try
    {
        c_phuong_11 = new Cal_Polynomial_11_phuong("2", "3,4,5");
        double result_11_phuong = c_phuong_11.Execute_11_phuong("2");
        actual_11_phuong = result_11_phuong.ToString();
    }
    catch (ArgumentException ex)
    {
}

```

```

        actual_11_phuong = ex.Message;
    }
    Assert.AreEqual(expected_11_phuong, actual_11_phuong);
}

//TestCase8_DocFileExcel_11_phuong: Đọc file excel để test tập dữ liệu với 5 cột cụ thể n,a,x,kq_mongdoi, kq_test
[TestMethod]
public void TestCase8_DocFileExcel_11_phuong()
{
    string file_11_phuong = @"D:\Nam 3\KTPM\11_Phương_Polynomial\UnitTest_11_phuong\Data_11_phuong\TestData_Excel_11_phuong.xlsx";
    Application excel_11_phuong = new Application();
    Workbook wb_11_phuong = excel_11_phuong.Workbooks.Open(file_11_phuong, ReadOnly: false);
    Worksheet ws_11_phuong = wb_11_phuong.Worksheets[1];

    int row_11_phuong = ws_11_phuong.UsedRange.Rows.Count;

    Console.WriteLine($"{{\"STT\",-5} {{\"n_11_phuong\",-5} {{\"a_11_phuong\",-10} {{\"x_11_phuong\",-5} {{\"KQ_ThucTe_11_phuong\",-70} {{\"KQ_Test\",-5}}}");
    Console.WriteLine(new string(' ', 150));

    int stt_11_phuong = 1;

    for (int i_11_phuong = 2; i_11_phuong <= row_11_phuong; i_11_phuong++)
    {
        string n_11_phuong = Convert.ToString(ws_11_phuong.Cells[i_11_phuong, 1].Value) ?? "";
        string a_11_phuong = Convert.ToString(ws_11_phuong.Cells[i_11_phuong, 2].Value) ?? "";
        string x_11_phuong = Convert.ToString(ws_11_phuong.Cells[i_11_phuong, 3].Value) ?? "";
        string expectedResult_11_phuong = Convert.ToString(ws_11_phuong.Cells[i_11_phuong, 4].Value) ?? "";
        string passFail_11_phuong = Convert.ToString(ws_11_phuong.Cells[i_11_phuong, 5].Value) ?? "";
        string actualResult_11_phuong = "";

        try
        {
            c_phuong_11 = new Cal_Polynomial_11_phuong(n_11_phuong, a_11_phuong);
            double result_11_phuong = c_phuong_11.Execute_11_phuong(x_11_phuong);
            actualResult_11_phuong = result_11_phuong.ToString();
        }
        catch (ArgumentException ex)
        {
            actualResult_11_phuong = ex.Message;
        }
        try
        {
            Assert.AreEqual(expectedResult_11_phuong, actualResult_11_phuong);
            passFail_11_phuong = passFail_11_phuong == "pass" ? passFail_11_phuong : "pass";
        }
        catch (AssertFailedException ex)
        {
            passFail_11_phuong = passFail_11_phuong == "fail" ? passFail_11_phuong : "fail";
        }
        Console.WriteLine($"{stt_11_phuong,-5} {n_11_phuong,-10} {a_11_phuong,-20} {x_11_phuong,-1} {actualResult_11_phuong,-63}
{passFail_11_phuong,-1}");
        stt_11_phuong++;
    }
    wb_11_phuong.Close(false);
    excel_11_phuong.Quit();
}

}

```

## 4.6. Kết quả thực hiện

Test run finished: 8 Tests (8 Passed, 0 Failed, 0 Skipped) run in 4 sec		
Test	Duration	Test Detail Summary
UnitTest_11_phuong (8)	3.4 sec	TestCase8_DocFileExcel_11_phuong
UnitTest_11_phuong (8)	3.4 sec	Source: UnitTest1_11_phuong.cs line 154
UnitTest1_11_phuong (8)	3.4 sec	Duration: 3.4 sec
TestCase1_NhapRong_11_Phuong	9 ms	Standard Output:
TestCase2_BacDaThucKhongHopLe_11_Phuong	< 1 ms	STT n_11_phuong a_11_phuong x_11_phuong KQ_ThucTe_11_phuong
TestCase3_DinhDangHeSoSai_11_Phuong	< 1 ms	-----
TestCase4_HeSoCongHoacSoDinhDang_11_Phuong	< 1 ms	1 1,2,3 2 Không được để trống dữ liệu
TestCase5_SoLuongHeSoSai_11_Phuong	< 1 ms	2 2 2 Không được để trống dữ liệu
TestCase6_XKhongHopLe_11_Phuong	< 1 ms	3 2 1,2,3 Không được để trống dữ liệu
TestCase7_TinhDaThucHopLe_11_Phuong	< 1 ms	4 2 Không được để trống dữ liệu
TestCase8_DocFileExcel_11_phuong	3.4 sec	5 1,2,3 Không được để trống dữ liệu
		6 2 Không được để trống dữ liệu

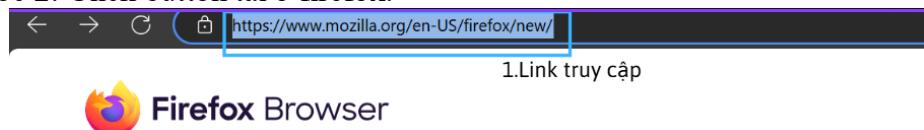
## II. Kiểm thử Selenium với trang web KFC VietNam.

## 1. Selenium.

### 1.1. Cài đặt trình duyệt FireFox.

1.1.1. Bước 1: Truy cập đường dẫn <https://www.mozilla.org/en-US/firefox/new/>

1.1.2. Bước 2: Click button tải ở firefox.



## Get the browser that protects what's important

Fast, reliable and private – for peace of mind online.

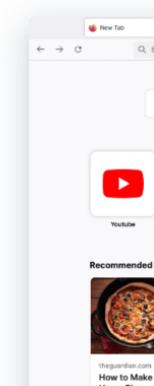
[Download Firefox](#) 2 Tải

Firefox Privacy Notice

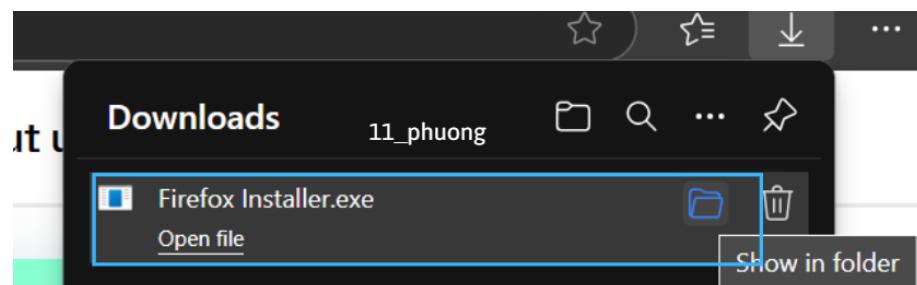
Download options and other languages

Firefox Browser support

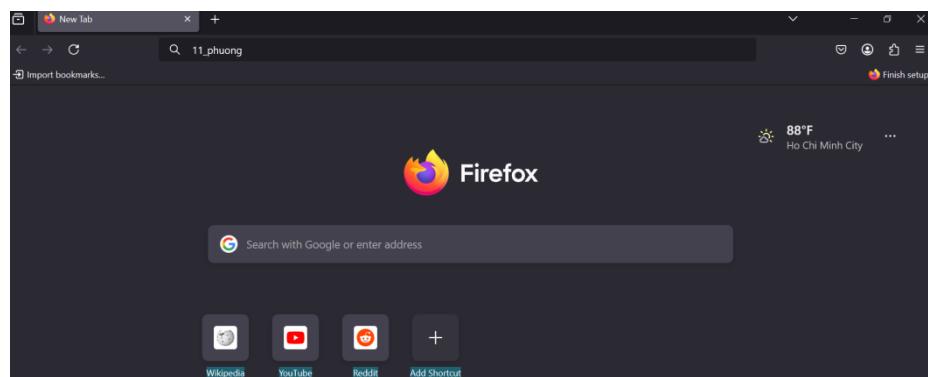
11\_phuong



1.1.3. Bước 3: Mở tệp file .exe đã download để cài đặt.



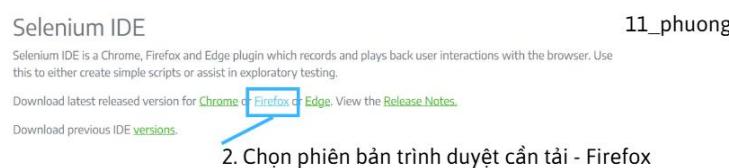
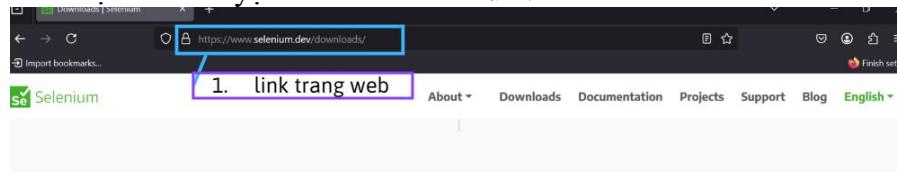
1.1.4. Bước 4: Chờ cài đặt và hoàn tất.



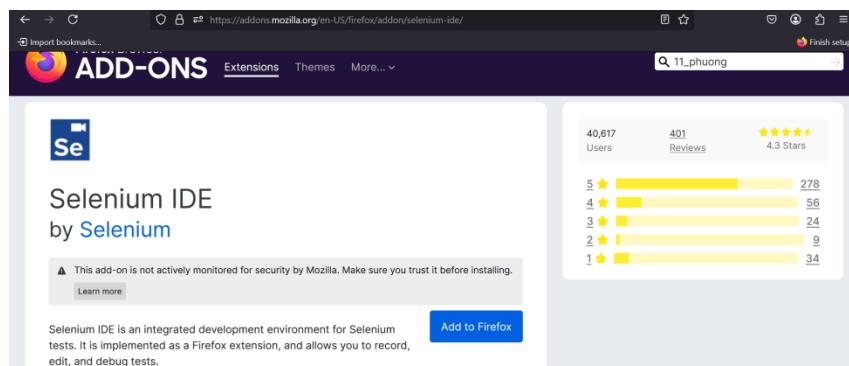
## 1.2. Cài đặt Selenium IDE trên trình duyệt FireFox.

1.2.1. Bước 1: Ở trình duyệt FireFox truy cập đường dẫn  
<https://www.selenium.dev/downloads/>

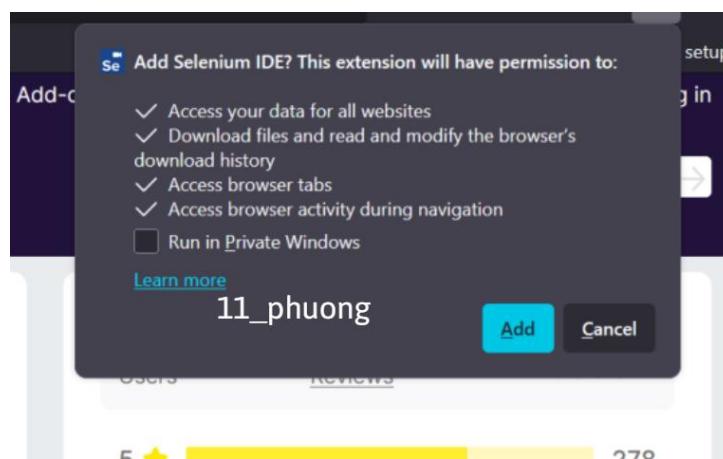
1.2.2. Bước 2: Chọn trình duyệt cần thêm Selenium.



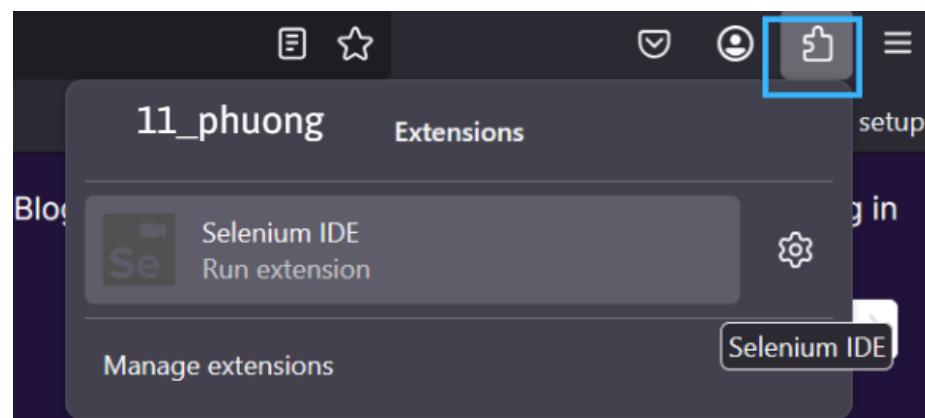
1.2.3. Bước 3: Nhấn Add to firefox.



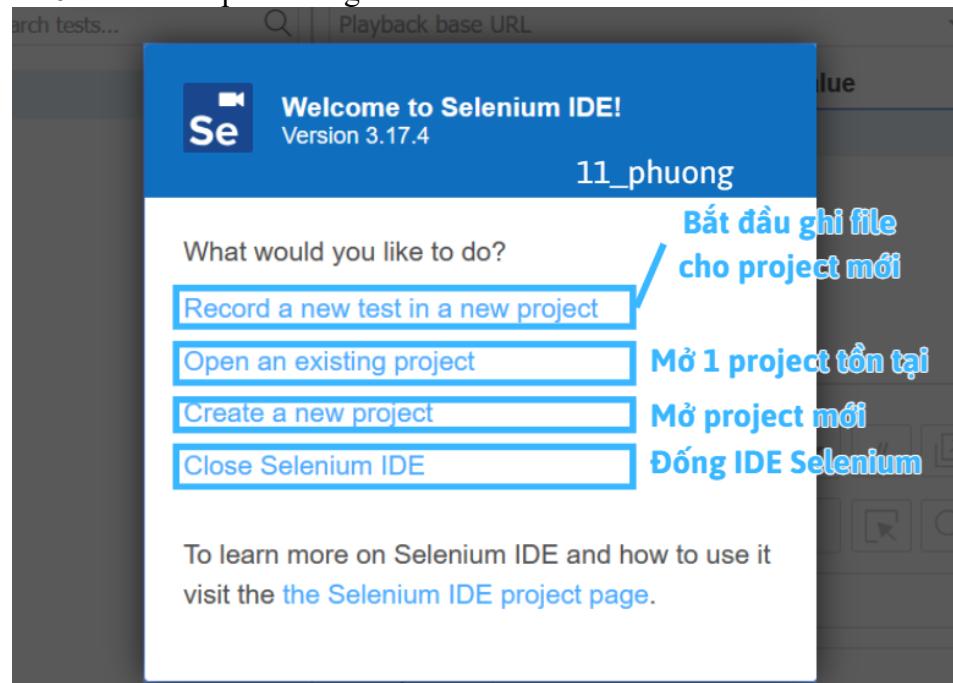
1.2.4. Bước 4: Trên trình duyệt hiển thị Popup -> Chọn Add.



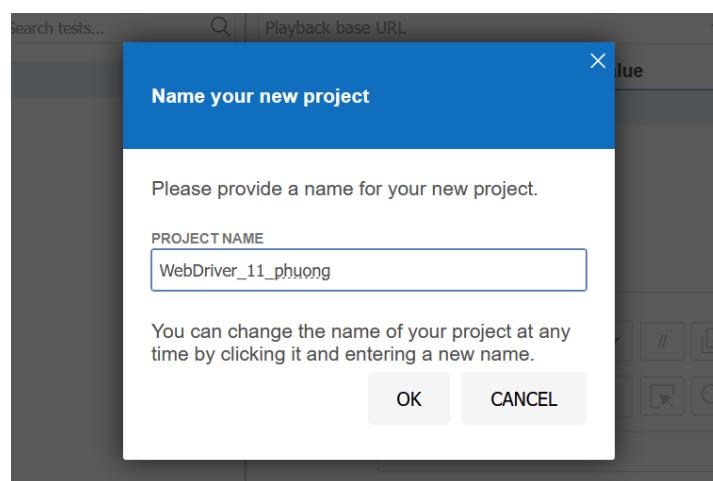
1.2.5. Bước 5: Sau khi thêm Selenium trên thanh công cụ sẽ hiển thị.

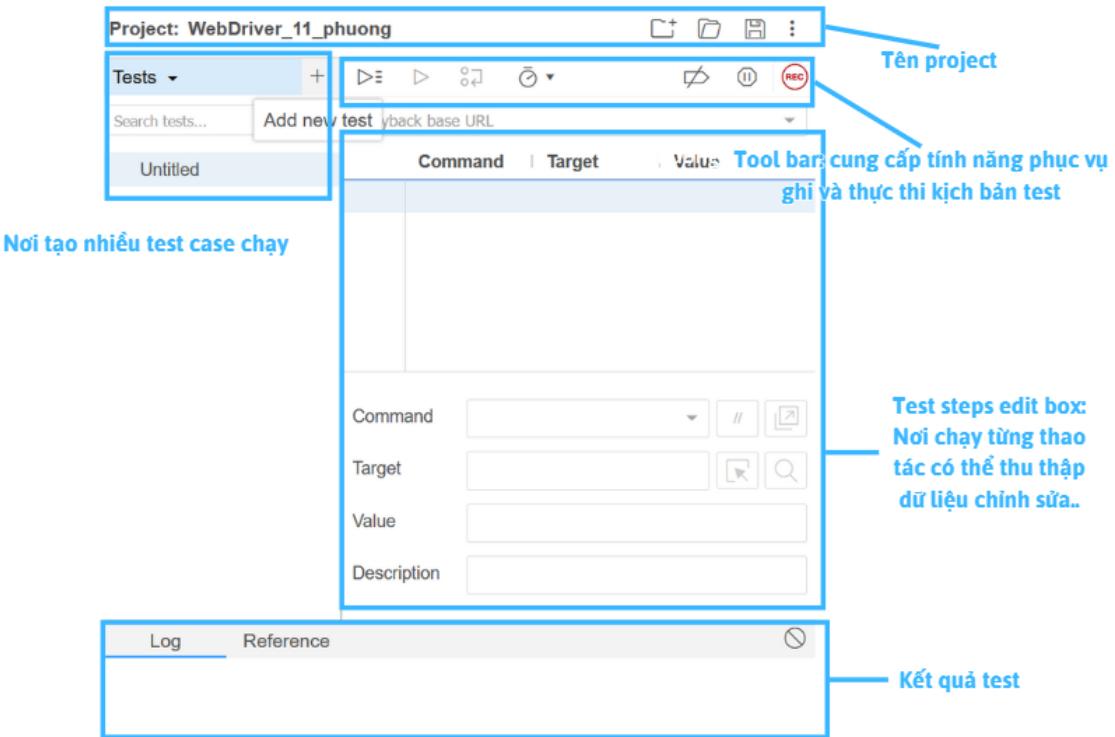


#### 1.2.6. Bước 6: Các thành phần trong Selenium IDE



#### 1.2.7. Bước 7: Tạo dự án mới tên WebDriver\_11\_phuong.

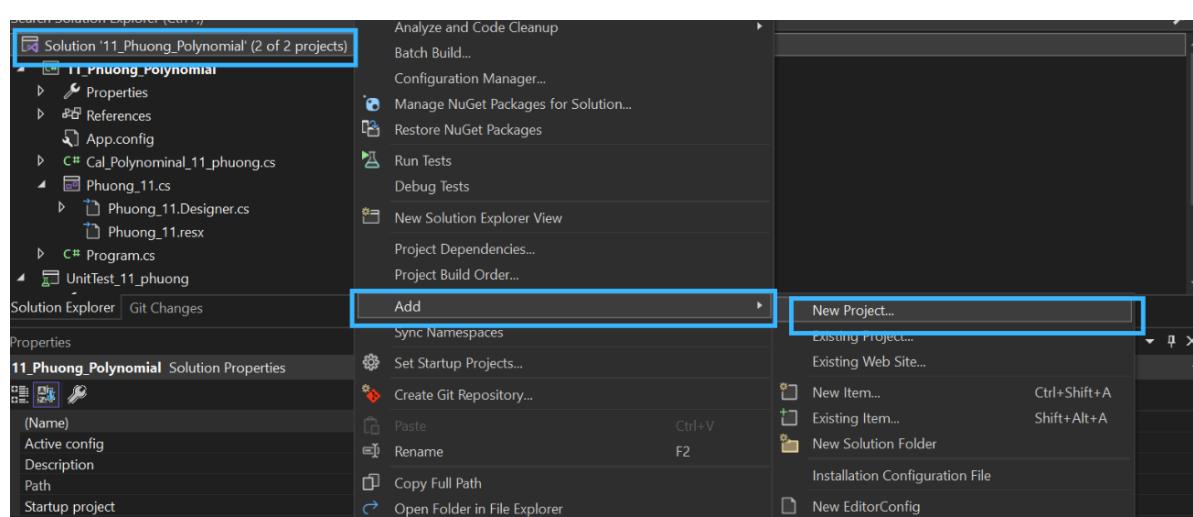




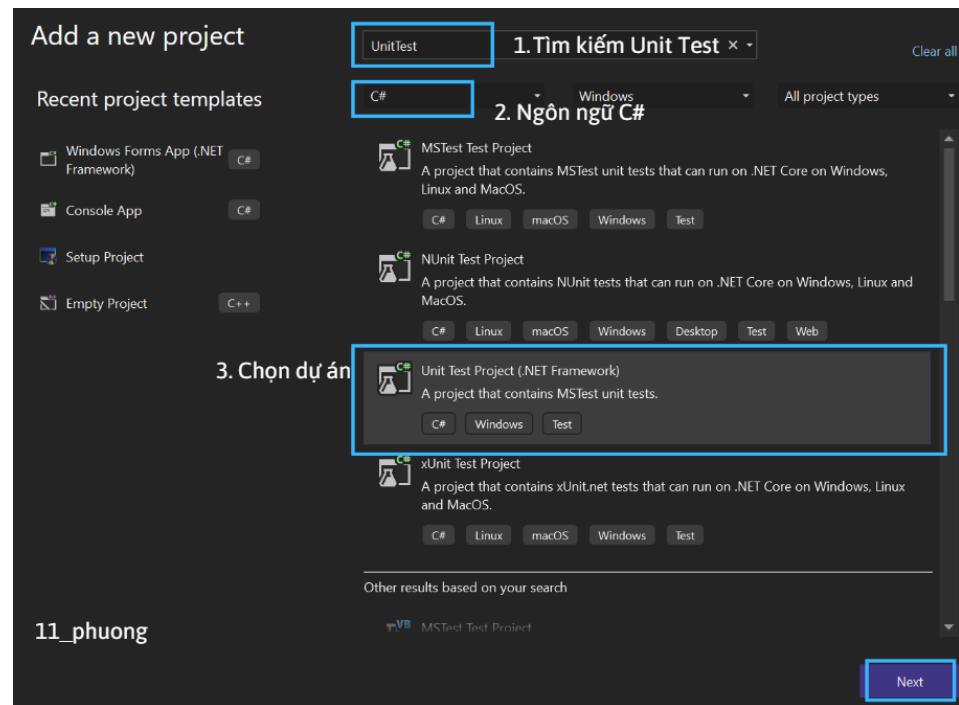
### 1.3. Cài đặt Selenium WebDriver.

#### 1.3.1. Tạo project tên 11\_phuong\_webdriver trong solution ‘11\_phuong\_polynomial’.

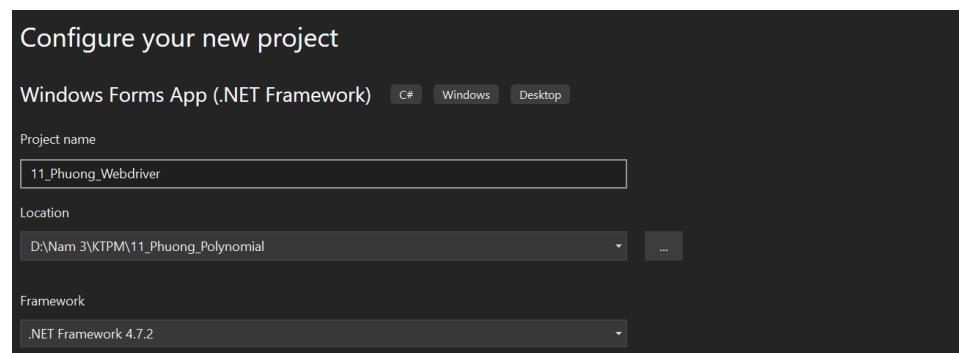
1.3.1.1. Bước 1: Nhấn chuột phải trên solution -> add -> new project....



1.3.1.2. Bước 2: Chọn dự án Unit Test Project(.NET)

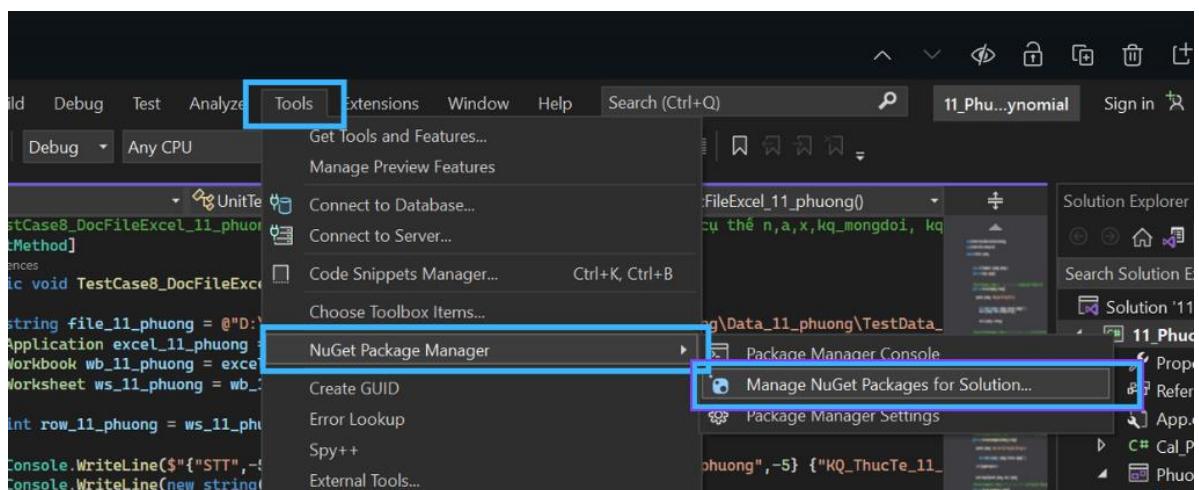


### 1.3.1.3. Bước 4: Đặt tên dự án 11\_Phuong\_Webdriver.

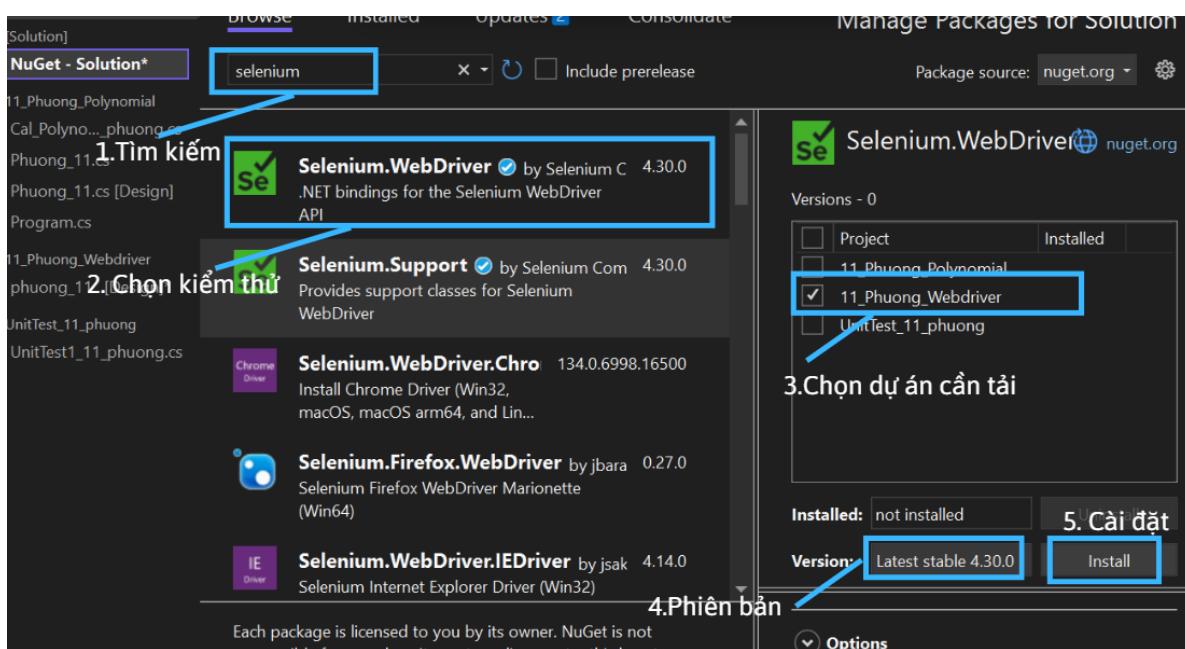


### 1.3.2. Cài đặt Selenium.

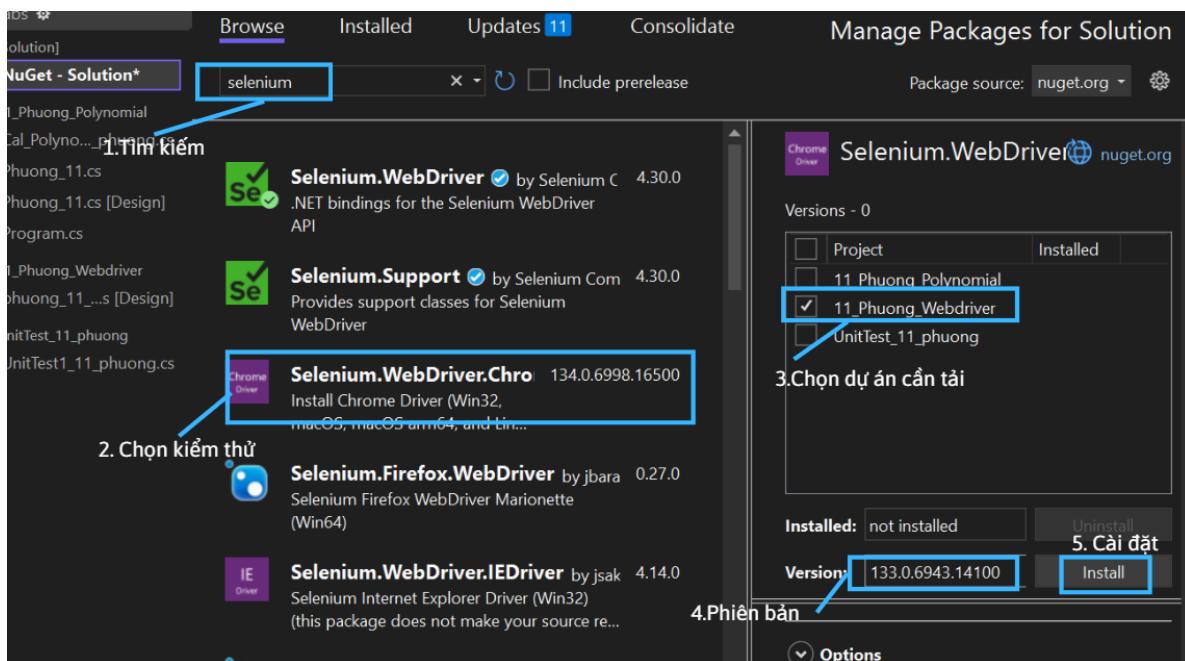
1.3.2.1. Bước 1: Tại project 11\_phuong\_polynomial ta vào tool -> NuGet Package Manager -> Manage NuGet Packages for Solution...



### 1.3.2.2. Bước 2: Chọn Selenium.WebDriver cần tải cho dự án 11\_Phuong\_Webdriver

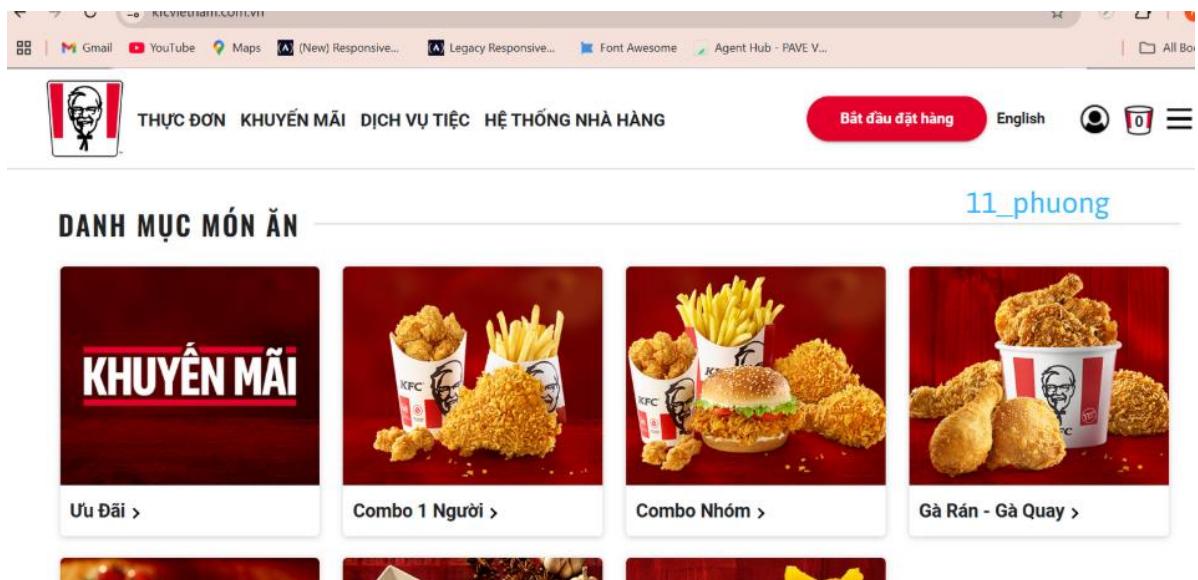


### 1.3.2.3. Bước 3: Chọn Selenium.WebDriver.Chrome trên trình duyệt Chrome cho phiên bản 133.0 của dự án.



## 2. Giới thiệu trang web KFC VietNam.

- KFC Vietnam là trang web chính thức của thương hiệu KFC tại Việt Nam, cung cấp dịch vụ đặt hàng trực tuyến, thông tin về thực đơn và chương trình khuyến mãi. Người dùng có thể dễ dàng tìm kiếm, đặt món và lựa chọn phương thức nhận hàng phù hợp như giao tận nơi hoặc nhận tại cửa hàng.
- Tổng quan về giao diện khi người dùng truy cập <https://kfcvietnam.com.vn/>.



### 3. Xác định yêu cầu chức năng.

#### 3.1. Đặc tả chức năng “Đăng ký tài khoản”.

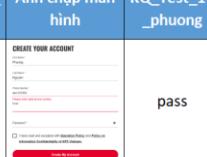
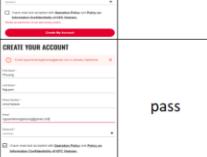
- Tên chức năng: Đăng ký tài khoản.
- Mô tả: Chức năng cho phép người dùng tạo mới tài khoản trên kfcvietnam.com.vn để có thể đăng nhập và dùng các dịch vụ khác của hệ thống.
- Tiền điều kiện: Không có.
- Luồng hoạt động:
  - + Bước 1: Người dùng truy cập vào trang <https://kfcvietnam.com.vn/en/account/register> đăng ký tài khoản.
  - + Bước 2: Người dùng nhập thông tin First Name và Last Name.
  - + Bước 3: Người dùng nhập Phone Number.
  - + Bước 4: Người dùng nhập email.
  - + Bước 5: Người dùng nhập password với điều kiện password phải có 1 ký tự hoa, 1 ký tự thường, 1 số và 1 ký tự đặc biệt.
  - + Bước 6: Người dùng chọn tích “I have read and accepted with Operation Policy and Policy on Information Confidentiality of KFC Vietnam”.
  - + Bước 7: Người dùng chọn nút “Create My Account”.
- Luồng thay thế:
  - + Bước 3.1: Nếu người dùng nhập nhỏ hơn 10 số hoặc nhập ký tự không hợp lệ, hệ thống hiển thị “Please enter valid phone number”.
  - + Bước 4.1: Nếu người dùng nhập sai định dạng email, hệ thống hiển thị “Please enter a valid email address”.
  - + Bước 5.1: Nếu người dùng nhập không đủ điều kiện đáp ứng ở bước 5, hệ thống hiển thị “Passwords must include an uppercase letter, a lowercase letter, a number, and one of the following symbols: !@#\$%&?\*”.
  - + Bước 7.1: Nếu người dùng bỏ sót hoặc nhập không đúng thông tin điện, hệ thống hiển thị thông báo yêu cầu người dùng thực hiện lại tại vị trí thiếu.
  - + Bước 7.2: Nếu email đã được đăng ký trước đó, hệ thống hiển thị “Email is already registered”.

### 3.2. Đặc tả chức năng “Đăng nhập tài khoản”.

- Tên chức năng: Đăng nhập tài khoản.
- Mô tả: Chức năng cho phép người dùng đăng nhập vào hệ thống bằng tài khoản đã đăng ký để sử dụng các dịch vụ như đặt hàng.
- Tiền điều kiện: Người dùng đã có tài khoản.
- Luồng hoạt động:
  - + Bước 1: Người dùng truy cập vào trang <https://kfcvietnam.com.vn/account/login> đăng nhập tài khoản.
  - + Bước 2: Người dùng nhập thông tin email.
  - + Bước 3: Người dùng nhập password.
  - + Bước 4: Người dùng chọn nút login.
  - + Bước 5: Hệ thống hiển thị trang tài khoản người dùng.
- Luồng thay thế:
  - + Bước 4.1: Nếu người dùng nhập thiếu thông tin, hệ thống thông báo tại vị trí thiếu cho người dùng nhập lại.
  - + Bước 4.2: Nếu tài khoản email chưa được đăng ký, email sai hoặc mật khẩu sai, hệ thống hiển thị lỗi “The email address or password you entered is incorrect. Please try again” và cho phép người dùng nhập lại.

## 4. Thiết kế test case.

### 4.1. Thiết kế test case dựa trên đặc tả chức năng “Đăng ký tài khoản”.

TestCase đăng ký tài khoản KFC											
STT	TestCase_11_phuong	firstname_11_phuong	lastname_11_phuong	phone_11_phuong	email_11_phuong	pass_11_phuong	check_11_phuong	KQ_MongMuon_11_phuong	KQ_ThucTe_11_phuong	Ảnh chụp màn hình	KQ_Test_11_phuong
1	TC1_SaiDinhDangSDT_11_phuong	Phuong	Nguyen	abc123456				Please enter valid phone number	Please enter valid phone number		pass
2	TC2_SaiDinhDangEmail_11_phuong	Phuong	Nguyen	0703792626	abc@	Phuong123*		Please enter a valid email address	Please enter a valid email address		pass
3	TC3_SaiDinhDangPass_11_phuong	Phuong	Nguyen	0703792626	nghongphuong@gmail.com	Phuong43		Passwords must include an uppercase letter, a lowercase	Passwords must include an uppercase letter, a lowercase		pass
4	TC4_NhapTrongThongTin_11_phuong	Phuong	Nguyen	0703792626	nghongphuong@gmail.com	Phuong123*		Please accept terms of user and privacy policy	Please accept terms of user and privacy policy		pass
5	TC5_EmailTonTai_11_Phuong	Phuong	Nguyen	0703792626	nguyenluongphuong@gmail.com	Phuong123*	click	Email is already registered	Email is already registered		pass

6	TC6_DangKyHopLe_11_phuong	Phuong	Nguyen	0703792626	<a href="#">nglkhongphuo ng@gmail.co m</a>	Phuong124*	click	firstname, lastname, email, phone, password hợp lệ	firstname, lastname, email, phone, password hợp lệ		pass
---	---------------------------	--------	--------	------------	--	------------	-------	--	--	---	------

#### 4.2. Thiết kế test case dựa trên đặc tả chức năng “Đăng nhập tài khoản”.

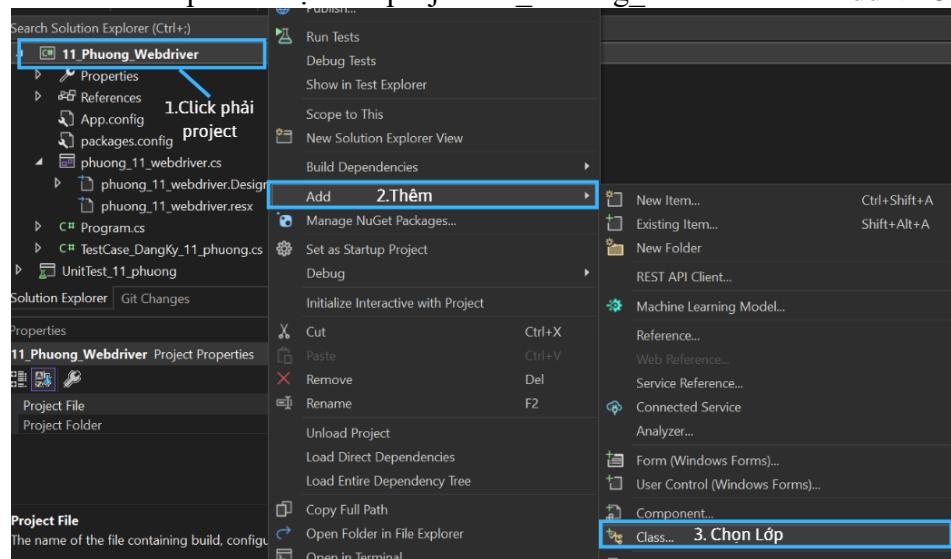
TestCase đăng nhập tài khoản KFC								
STT	TestCase_11_phuong	Email_11_phuong	password_11_phuong	KQ_MongMuon_11_phuong	KQ_ThucTe_11_phuong	Ảnh chụp màn hình	KQ_Test_11_phuong	
1	TC1_NhapTrongThongTin_11_phuong	<a href="#">nguyenluonghongphuong@gmail.com</a>		Please enter your password	Please enter your password		pass	
2	TC2_SaiMKHoacEmail_11_phuong	<a href="#">guiauuhu78@ou.edu.vn</a>	123	The email address or password you entered is incorrect. Please try again	The email address or password you entered is incorrect. Please try again		pass	
3	TC3_DangNhapHopLe_11_phuong	<a href="#">nguyenluonghongphuong@gmail.com</a>	Phuong123*	Email_11_phuong, password_11_phuong hợp lệ.	Email_11_phuong, password_12_phuong hợp lệ.		pass	

### 5. Kiểm thử chức năng.

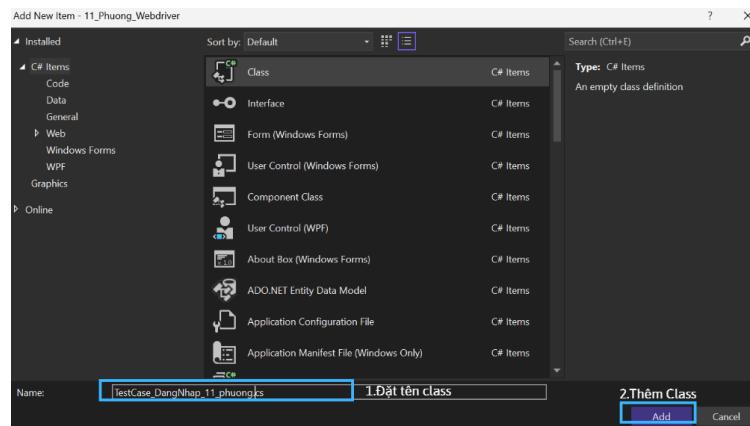
#### 5.1. Chuẩn bị trước khi kiểm thử.

- Tạo lớp TestCase\_DangKy\_11\_phuong.cs và TestCase\_DangNhap\_11\_phuong.cs là nơi lưu trữ các test case của 2 chức năng khác nhau.

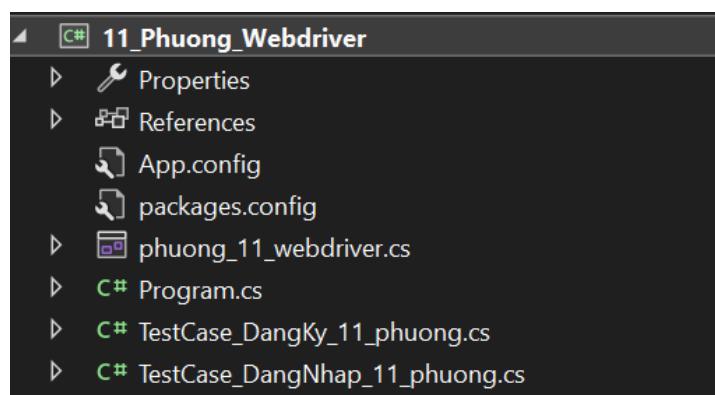
##### 5.1.1.1. Bước 1: Click phải chuột trên project 11\_Phuong\_Webdriver -> Add -> Class ...



##### 5.1.1.2. Bước 2: Đặt tên Class TestCase\_DangNhap\_11\_phuong -> Add.



5.1.1.3. Bước 3: Tương tự đối với tạo lớp TestCase\_DangKy\_11\_phuong ta được kết quả.



## 5.2. Kiểm thử chức năng “Đăng ký tài khoản”.

### 5.2.1. Lấy các trường thông tin của chức năng đăng ký tài khoản.

5.2.1.1. Bước 1: Ta truy cập trang <https://kfcvietnam.com.vn/en/account/register>, click phải page mở ché độ inspect -> để lấy được thông tin firstname ta cần bắt trường id.

```



```

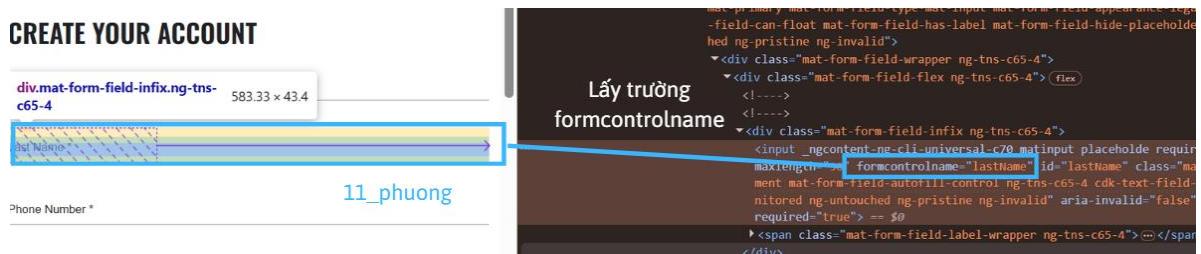
- Source code:

```

IWebDriver drive_11_phuong = new ChromeDriver();
drive_11_phuong.Url = "https://kfcvietnam.com.vn/en/account/register";
drive_11_phuong.Navigate();
IWebElement firstname_11_phuong = drive_11_phuong.FindElement(By.Id("firstName"));

```

### 5.2.1.2. Bước 2: Để lấy thông tin last name ta bắt trườòng formcontrolname='lastName'



- Source code:

```
IWebElement lastname_11_phuong =  
drive_11_phuong.FindElement(By.CssSelector("[formcontrolname='lastName']"));
```

### 5.2.1.3. Bước 3:Lấy trùòng id của phone number



- Source code:

```
IWebElement phone_11_phuong = drive_11_phuong.FindElement(By.Id("phoneNumber"));
```

### 5.2.1.4. Bước 4: Lấy trùòng id của email



- Source code:

```
IWebElement email_11_phuong = drive_11_phuong.FindElement(By.Id("email"));
```

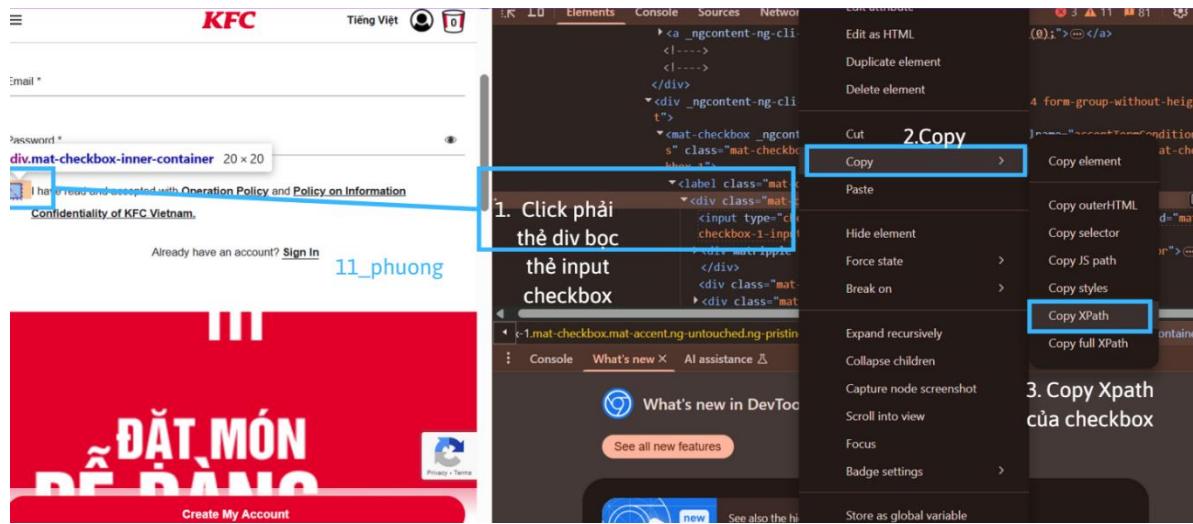
### 5.2.1.5. Bước 5: Bắt trùòng id của password.



- Source code:

```
IWebElement pass_11_phuong = drive_11_phuong.FindElement(By.Id("password"));
```

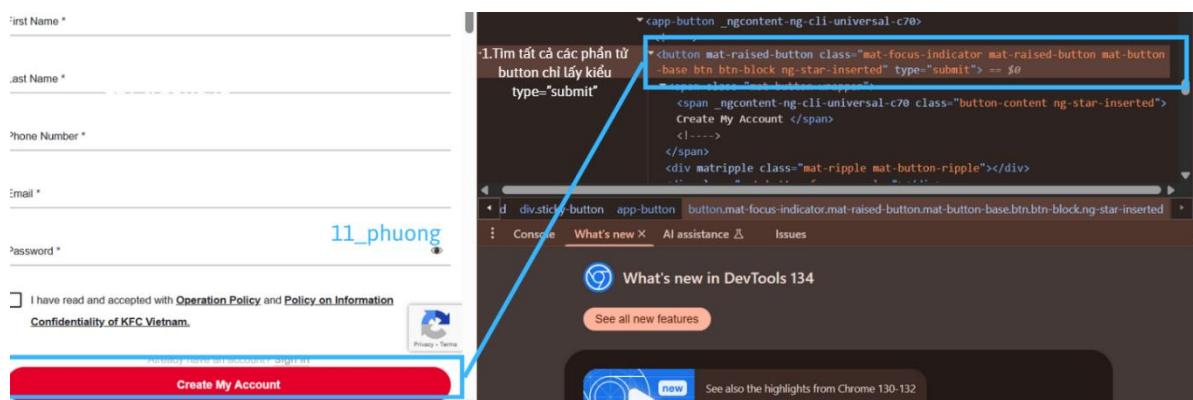
### 5.2.1.6. Bước 6: Bắt Xpath của checkbox



- Source code:

```
IWebElement checkbox_11_phuong = driver_11_phuong.FindElement(By.XPath("//*[@id='mat-checkbox-1']/label/div"));
```

### 5.2.1.7. Bước 7: Bắt trường Xpath của nút create my account



- Source code:

```
IWebElement button_11_phuong = driver_11_phuong.FindElement(By.XPath("//button[@type='submit']"));
```

## 5.2.2. Thực hiện TestCase\_DangKy\_11\_phuong.cs

5.2.2.1. Bước 1: Vào lớp TestCase\_DangKy\_11\_phuong.cs ta có lớp được thực hiện như sau:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```

using System.Threading.Tasks;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;

namespace _11_Phuong_Webdriver
{
    public class TestCase_DangKy_11_phuong
    {
        private string firstName_11_phuong;
        private string lastName_11_phuong;
        private string phone_11_phuong;
        private string email_11_phuong;
        private string password_11_phuong;
        private bool isChecked_11_phuong;

        public TestCase_DangKy_11_phuong(string firstName_11_phuong,
            string lastName_11_phuong,
            string phone_11_phuong,
            string email_11_phuong,
            string password_11_phuong,
            bool isChecked_11_phuong)
        {
            this.firstName_11_phuong = firstName_11_phuong;
            this.lastName_11_phuong = lastName_11_phuong;
            this.phone_11_phuong = phone_11_phuong;
            this.email_11_phuong = email_11_phuong;
            this.password_11_phuong = password_11_phuong;
            this.isChecked_11_phuong = isChecked_11_phuong;
        }

        public void Execute_11_phuong()
        {
            IWebDriver driver_11_phuong = new ChromeDriver();
            try
            {
                driver_11_phuong.Url = "https://kfcvietnam.com.vn/en/account/register";
                driver_11_phuong.Navigate();

                // Điền thông tin vào form đăng ký
                driver_11_phuong.FindElement(By.Id("firstName")).SendKeys(firstName_11_phuong);

                driver_11_phuong.FindElement(By.CssSelector("[formcontrolname='lastName']")).SendKeys(lastName_11_phuong);
                driver_11_phuong.FindElement(By.Id("phoneNumber")).SendKeys(phone_11_phuong);
                driver_11_phuong.FindElement(By.Id("email")).SendKeys(email_11_phuong);
                driver_11_phuong.FindElement(By.Id("password")).SendKeys(password_11_phuong);

                if (isChecked_11_phuong)
                {
                    // Tích vào checkbox
                    driver_11_phuong.FindElement(By.XPath("//*[@id='mat-checkbox-1']/label/div")).Click();
                }
                // Nhấn nút Đăng ký
                driver_11_phuong.FindElement(By.XPath("//button[@type='submit']")).Click();

                Console.WriteLine("Đăng ký tài khoản thành công!");
            }
            catch (Exception ex)
            {
                Console.WriteLine("Lỗi khi đăng ký tài khoản: " + ex.Message);
            }
            finally
            {

```

```

        // Đóng trình duyệt sau khi hoàn tất
        //driver_11_phuong.Quit();
    }
}

}

```

5.2.2.2. Bước 2: Thực hiện các test case của form đăng ký ở lớp `phuong_11_webdriver`.

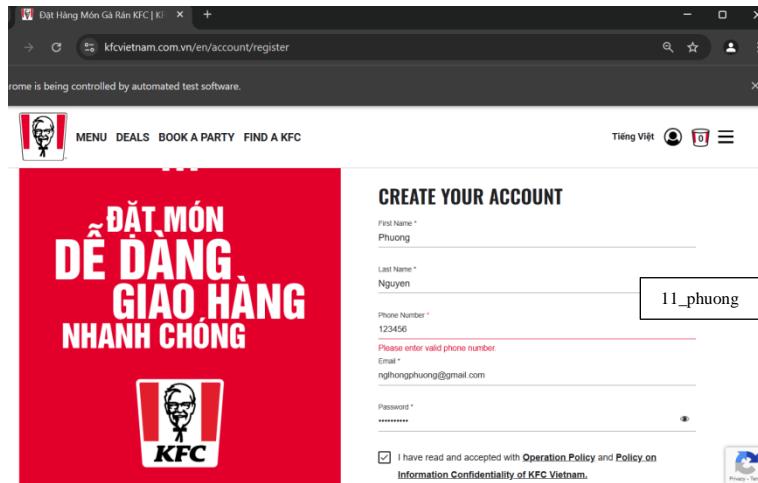
#### 5.2.2.2.1. TC1\_SaiDinhDangSDT\_11\_phuong

```

namespace _11_Phuong_Webdriver
{
    public partial class phuong_11_webdriver : Form
    {
        public phuong_11_webdriver()
        {
            InitializeComponent();
        }

        private void browser_11_phuong_Click(object sender, EventArgs e)
        {
            TestCase_DangKy_11_phuong TC1_SaiDinhDangSDT_11_phuong = new
            TestCase_DangKy_11_phuong(
                "Phuong",           // firstName
                "Nguyen",           // lastName
                "123abc456",       // phoneNumber (sai định dạng)
                "nglhongphuong@gmail.com", // email
                "Phuong123*",      // password
                true                // isChecked (tích checkbox)
            );
            TC1_SaiDinhDangSDT_11_phuong.Execute_11_phuong();
        }
    }
}

```



#### 5.2.2.2.2. TC2\_SaiDinhDangEmail\_11\_phuong

```

TestCase_DangKy_11_phuong TC2_SaiDinhDangEmail_11_phuong = new
TestCase_DangKy_11_phuong(
    "Phuong",           // firstName
    "Nguyen",           // lastName
    "12345678",         // phoneNumber

```

```
"nglkhongphuong", // email (sai định dạng)
"Phuong123*", // password
true // isChecked (tích checkbox)
);
TC2_SaiDinhDangEmail_11_phuong.Execute_11_phuong();
```

The screenshot shows a web browser on kfcvietnam.com.vn/en/account/register. The main banner says "ĐẶT MÓN DỄ ĐĂNG GIAO HÀNG NHANH CHÓNG". The "CREATE YOUR ACCOUNT" form has the following fields:

- First Name: Phuong
- Last Name: Nguyen
- Phone Number: 12345678
- Email: nglkhongphuong (highlighted in red)
- Password: 11\_phuong
- I have read and accepted with [Operation Policy](#) and [Policy on Information Confidentiality of KFC Vietnam](#).

An error message "Please enter valid phone number" is displayed above the phone number field, and "Please enter a valid email address." is displayed above the email field.

#### 5.2.2.2.3. TC3\_SaiDinhDangPass\_11\_phuong

```
TestCase_DangKy_11_phuong TC3_SaiDinhDangPass_11_phuong = new
TestCase_DangKy_11_phuong(
    "Phuong", // firstName
    "Nguyen", // lastName
    "1234567891", // phoneNumber
    "nglkhongphuong@gmail.com", // email
    "Phuong", // password (sai định dạng)
    true // isChecked (tích checkbox)
);
TC3_SaiDinhDangPass_11_phuong.Execute_11_phuong();
```

The screenshot shows a web browser on kfcvietnam.com.vn/en/account/register. The main banner says "ĐẶT MÓN DỄ ĐĂNG GIAO HÀNG NHANH CHÓNG". The "CREATE YOUR ACCOUNT" form has the following fields:

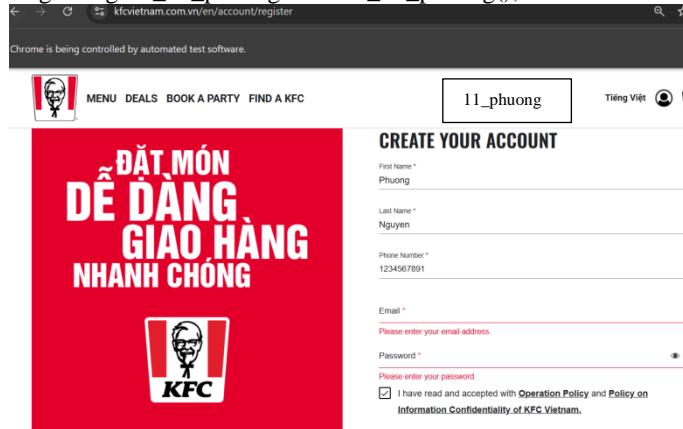
- First Name: Phuong
- Last Name: Nguyen
- Phone Number: 1234567891
- Email: nglkhongphuong@gmail.com
- Password: 11\_phuong
- I have read and accepted with [Operation Policy](#) and [Policy on Information Confidentiality of KFC Vietnam](#).

An error message "Passwords must include an uppercase letter, a lowercase letter, a number, and one of the following symbols: !@#\$%^&\*." is displayed below the password field.

#### 5.2.2.2.4. TC4\_NhapTrongThongTin\_11\_phuong

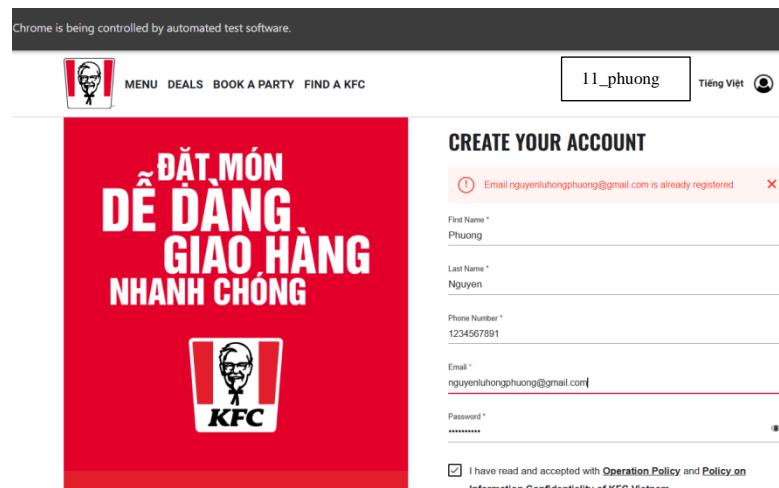
```
TestCase_DangKy_11_phuong TC4_NhapTrongThongTin_11_phuong = new
TestCase_DangKy_11_phuong(
    "Phuong", // firstName
    "Nguyen", // lastName
    "1234567891", // phoneNumber
```

```
""" // email - trống thông tin
""" // password - trống thông tin
true           // isChecked (tích checkbox)
);
TC4_NhapTrongThongTin_11_phuong.Execute_11_phuong();
```



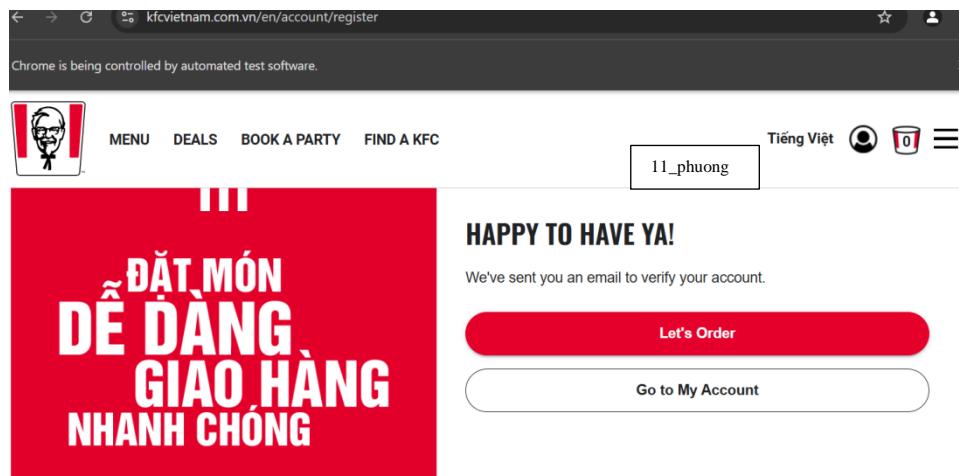
#### 5.2.2.2.5. TC5\_EmailTonTai\_11\_phuong

```
TestCase_DangKy_11_phuong TC5_EmailTonTai_11_phuong = new TestCase_DangKy_11_phuong(
"Phuong",           // firstName
"Nguyen",          // lastName
"1234567891",     // phoneNumber
"nguyenluongphuong@gmail.com", // email tồn tại
"Phuong123*",      // password
true               // isChecked (tích checkbox)
);
TC5_EmailTonTai_11_phuong.Execute_11_phuong();
```



#### 5.2.2.2.6. TC6\_DangKyHopLe\_11\_phuong

```
TestCase_DangKy_11_phuong TC6_DangKyHopLe_11_phuong = new TestCase_DangKy_11_phuong(
"Phuong",           // firstName
"Nguyen",          // lastName
"1234567891",     // phoneNumber
"nguyenluongphuong@gmail.com", // email
"Phuong123*",      // password
true               // isChecked (tích checkbox)
);
TC6_DangKyHopLe_11_phuong.Execute_11_phuong();
```



### 5.3. Kiểm thử chức năng “Đăng nhập tài khoản”.

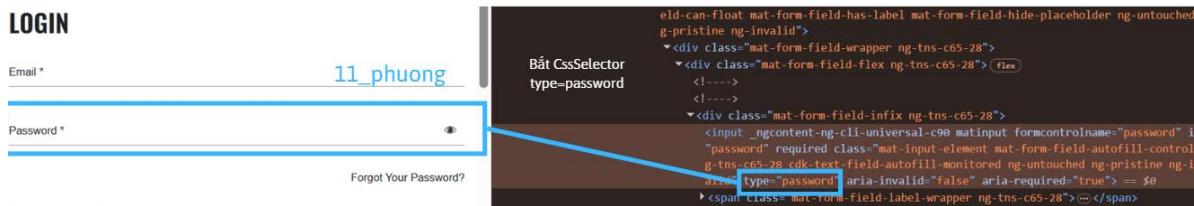
#### 5.3.1. Lấy các trường thông tin của chức năng đăng nhập tài khoản.

5.3.1.1. Bước 1: Ta truy cập trang <https://kfcvietnam.com.vn/account/login>, click phải page mở chế độ inspect -> để lấy được thông tin ta cần bắt trường formcontrol=email

- Source code:

```
IWebDriver drive_11_phuong = new ChromeDriver();
drive_11_phuong.Url = "https://kfcvietnam.com.vn/account/login";
drive_11_phuong.Navigate();
IWebElement email_11_phuong =
drive_11_phuong.FindElement(By.CssSelector("[formcontrolname='email']"));
```

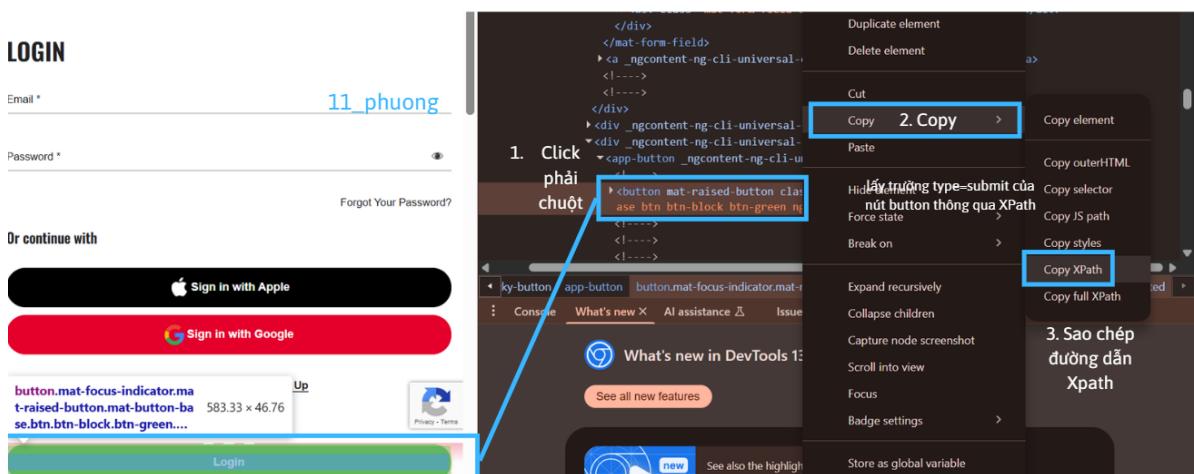
### 5.3.1.2. Bước 2: Bắt trỏng type=password của password



- Source code:

```
IWebElement pass_11_phuong = drive_11_phuong.FindElement(By.CssSelector("[type='password']"));
```

### 5.3.1.3. Bước 3: Bắt Xpath của nút login.



- Source code:

```
IWebElement button = drive_11_phuong.FindElement(By.XPath("//html/body/app-root/app-menu-layout/section/main/app-login/section/div/div[1]/div/form/div[4]/app-button/button"));
```

## 5.3.2. Thực hiện TestCase\_DangNhap\_11\_phuong.cs

### 5.3.2.1. Bước 1: Vào lớp TestCase\_DangNhap\_11\_phuong.cs ta có lớp được thực hiện như sau:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium;

namespace _11_Phuong_Webdriver
{
    public class TestCase_DangNhap_11_phuong
    {
```

```
private string email_11_phuong;
private string password_11_phuong;

public TestCase_DangNhap_11_phuong(string email_11_phuong, string password_11_phuong)
{
    this.email_11_phuong = email_11_phuong;
    this.password_11_phuong = password_11_phuong;
}

public void Execute_11_phuong()
{
    IWebDriver driver_11_phuong = new ChromeDriver();
    try
    {
        driver_11_phuong.Url = "https://kfcvietnam.com.vn/en/account/login";
        driver_11_phuong.Navigate();

        IWebElement emailField_11_phuong =
driver_11_phuong.FindElement(By.CssSelector("[formcontrolname='email']"));
        emailField_11_phuong.SendKeys(email_11_phuong);

        IWebElement passwordField_11_phuong =
driver_11_phuong.FindElement(By.CssSelector("[type='password']"));
        passwordField_11_phuong.SendKeys(password_11_phuong);

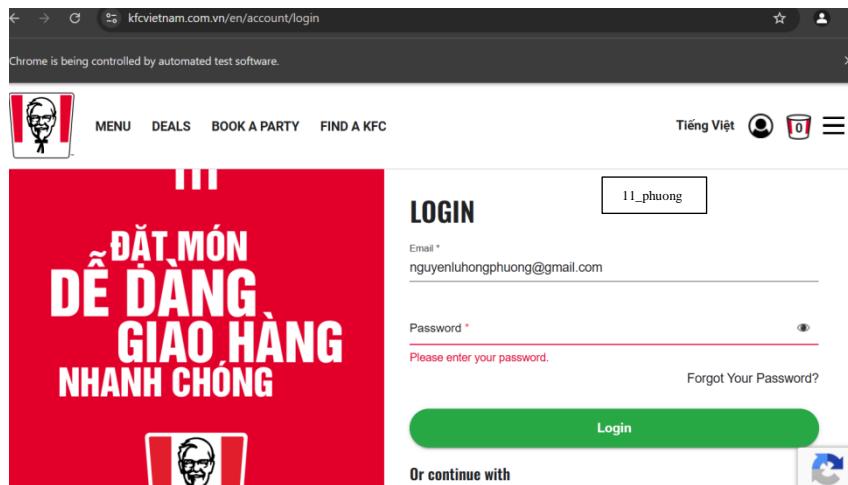
        IWebElement loginButton_11_phuong = driver_11_phuong.FindElement(By.XPath("//app-
login//form//app-button/button"));
        loginButton_11_phuong.Click();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Lỗi khi đăng nhập: " + ex.Message);
    }
    finally
    {
        // Đóng trình duyệt sau khi hoàn tất
        // driver_11_phuong.Quit();
    }
}

}
}
```

5.3.2.2. Bước 2: Thực hiện các test case của form đăng nhập ở lớp `phuong_11_webdriver`.

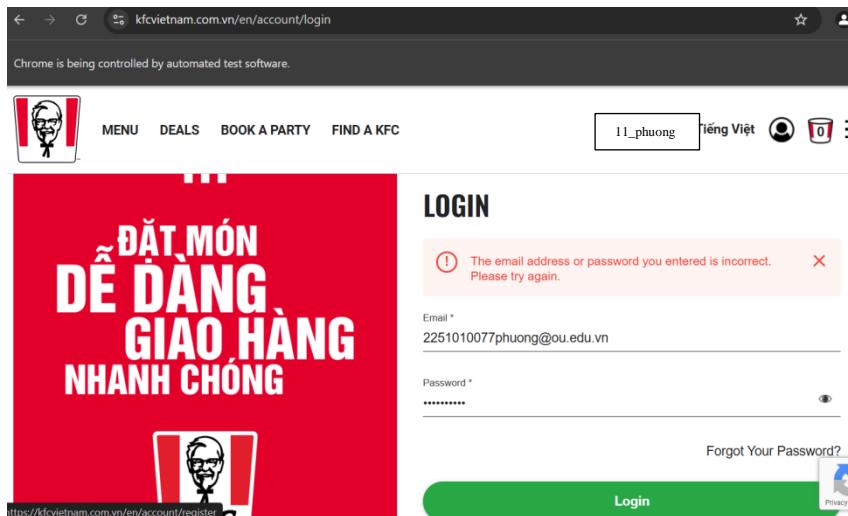
#### 5.3.2.2.1. TC1\_NhapTrongThongTin\_11\_phuong

```
TestCase_DangNhap_11_phuong TC1_NhapTrongThongTin_11_phuong = new
TestCase_DangNhap_11_phuong(
    "nguyenluongphuong@gmail.com",
    "" //Nhập trống thông tin
);
TC1_NhapTrongThongTin_11_phuong.Execute_11_phuong();
```



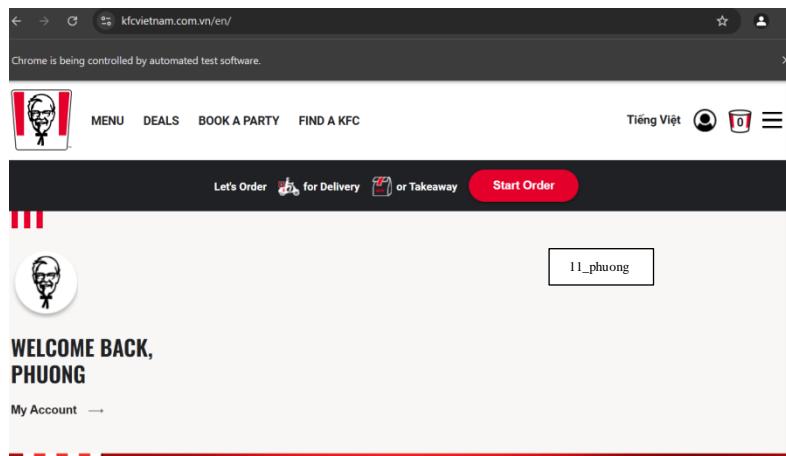
#### 5.3.2.2.2. TC2\_SaiMKHoacEmail\_11\_phuong

```
TestCase_DangNhap_11_phuong TC2_SaiMKHoacEmail_11_phuong = new TestCase_DangNhap_11_phuong(
    "2251010077phuong@ou.edu.vn", //Nhập email không tồn tại
    "Phuong123*"
);
TC2_SaiMKHoacEmail_11_phuong.Execute_11_phuong();
```



#### 5.3.2.2.3. TC3\_DangNhapHopLe\_11\_phuong

```
TestCase_DangNhap_11_phuong TC3_DangNhapHopLe_11_phuong = new
TestCase_DangNhap_11_phuong(
    "nguyenluongphuong@gmail.com",
    "Phuong123*"
);
TC3_DangNhapHopLe_11_phuong.Execute_11_phuong();
```

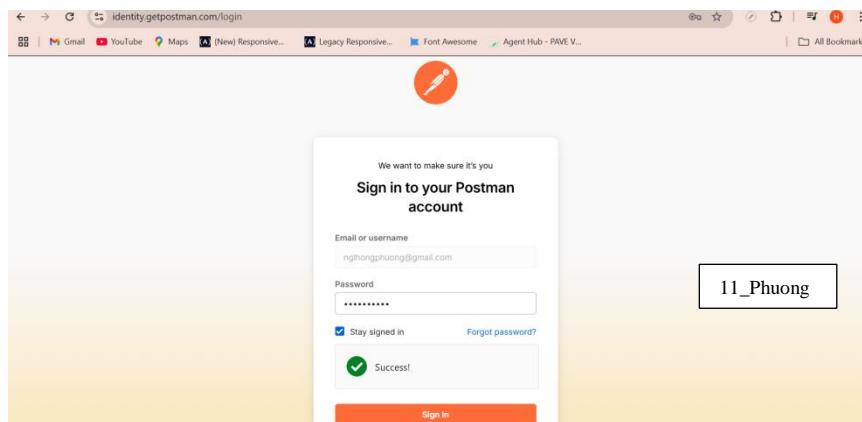


### III. Kiểm thử API Postman với JSONPlaceholder.

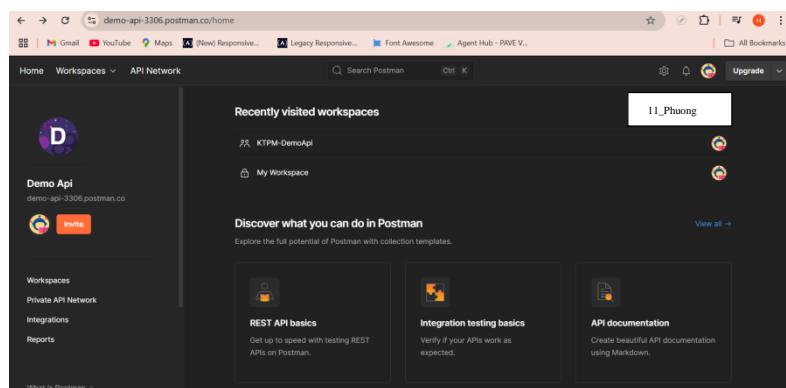
#### 1. Thiết lập Postman trên web.

##### 1.1. Tạo WorkSpace trên Postman

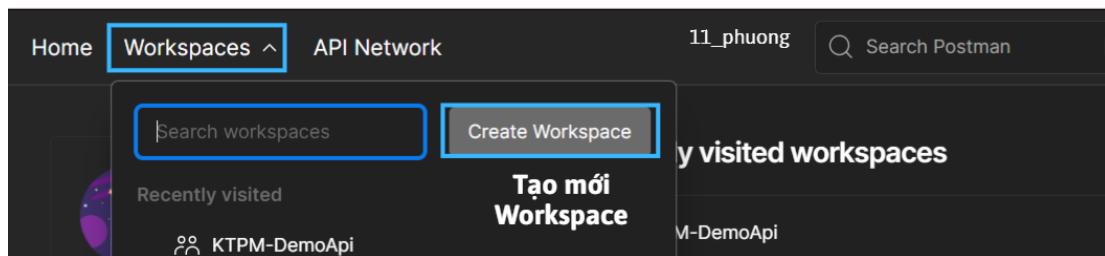
1.1.1. Bước 1: Ta truy cập <https://identity.getpostman.com/login> để đăng nhập tài khoản và sử dụng Postman trên nền tảng website của trình duyệt Chrome.



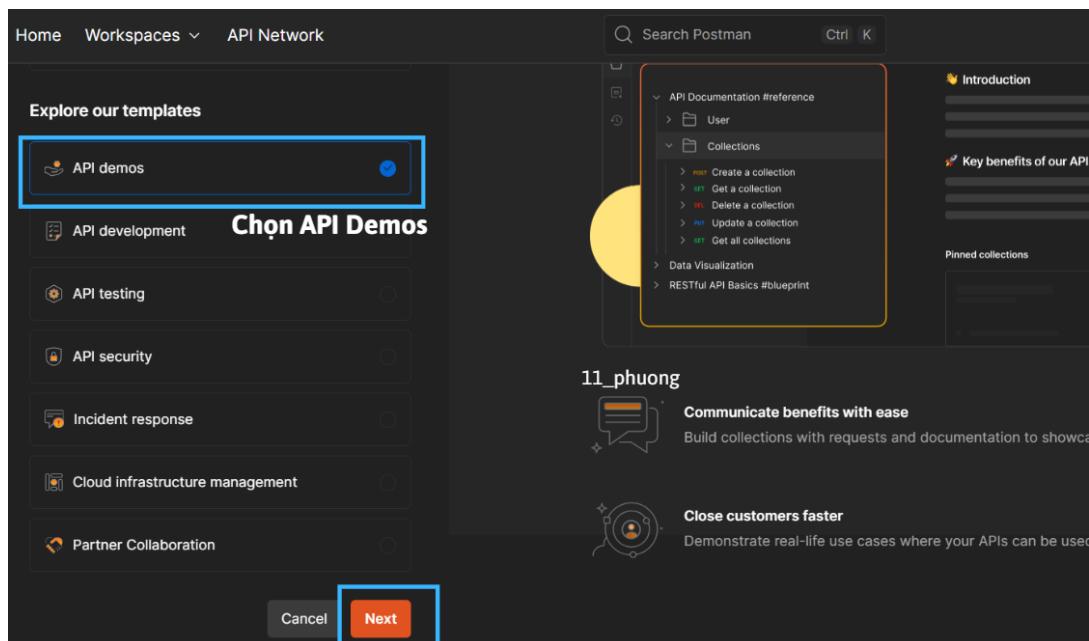
- Giao diện sau khi đăng nhập:



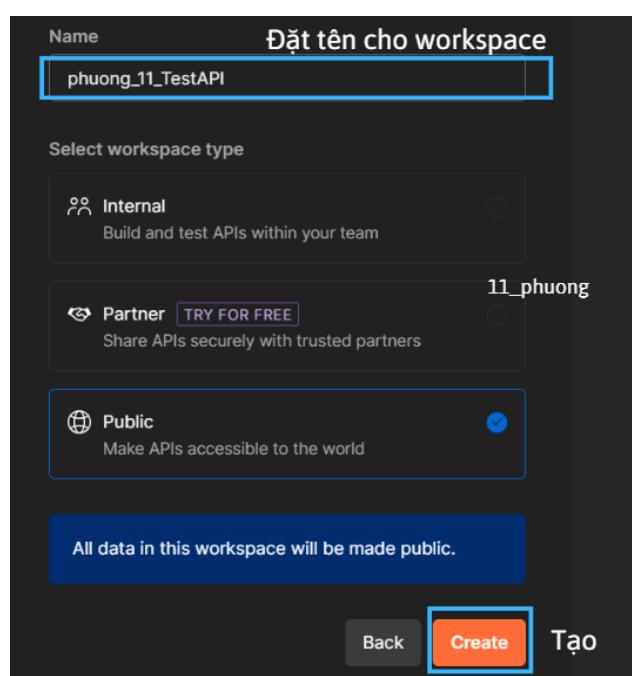
1.1.2. Bước 2: Click phải chuột trên Workspace ở thanh tool bar -> chọn create workspace



1.1.3. Bước 3: Chọn API Demos -> Next



1.1.4. Bước 4: Đặt tên cho workspace và để chế độ public -> Tạo



1.1.5. Bước 5: Sau khi tạo ta sẽ được giao diện dưới đây:

The screenshot shows the Postman interface with the workspace 'phuong\_11\_TestAPI' selected. The left sidebar shows collections like 'User', 'Collections', 'Data Visualization', and 'RESTful API Basics #blueprint'. The main area displays the 'API Documentation #reference' template, which includes sections for 'Overview', 'Authorization', 'Scripts', 'Variables', and 'Runs'. It also shows 'Popular requests' with options like 'Create a collection', 'Delete a collection', 'Get a collection', and 'Get all collections'.

1.2. Tạo một request mới trên Postman trong workspace phuong\_11\_TestAPI.

Ta chọn dấu cộng để thêm mới request.

The screenshot shows the Postman interface with the workspace 'phuong\_11\_TestAPI' selected. The top bar has 'Overview' highlighted. A large blue box highlights the '+ Thêm mới request' button in the top right corner of the main workspace area.

- Kết quả sau khi thêm:

The screenshot shows the Postman interface with the workspace 'phuong\_11\_TestAPI' selected. The main area shows an 'Untitled Request' configuration screen. Annotations explain various fields: 'Chọn phương thức HTTP' points to the method dropdown (GET, POST, PUT, PATCH, DELETE, HEAD, OPTIONS); 'Nhập URL để gửi request' points to the 'Enter URL or paste text' input field; 'Lưu request' points to the 'Save' button; and 'Gửi request' points to the 'Send' button. The left sidebar shows collections and environments.

## 2. Giới thiệu về JSONPlaceholder

- JSONPlaceholder là một API giả lập miễn phí dành cho thử nghiệm và phát triển ứng dụng. Nó cung cấp dữ liệu mẫu về người dùng, bài viết, bình luận, ảnh, v.v., giúp kiểm thử các phương thức HTTP như GET, POST, PUT, PATCH, DELETE một cách nhanh chóng mà không cần thiết lập server riêng.

- Các Endpoint Chính:

/posts – Bài viết mẫu

/comments – Bình luận mẫu

/albums – Album ảnh

/photos – Ảnh mẫu

/todos – Danh sách công việc

/users – Thông tin người dùng

- Ảnh tổng quan về trang web <https://jsonplaceholder.typicode.com/>

The screenshot shows the homepage of <https://jsonplaceholder.typicode.com/>. The page title is "When to use". It explains that JSONPlaceholder is a free online REST API for fake data. Below this, there's a section titled "Resources" which lists six common resources with their counts: posts (100), comments (500), albums (100), photos (5000), todos (200), and users (10). A box labeled "11\_phuong" highlights the "/posts" resource.

### 3. Kiểm thử API Postman với JSONPlaceholder.

- Docs: <https://learning.postman.com/docs/tests-and-scripts/write-scripts/test-examples/>

#### 3.1. Tạo và lưu một request: get.

3.1.1. Bước 1: Ta sao chép đường dẫn

<https://jsonplaceholder.typicode.com/posts/1/comments> để lấy danh sách bình luận thuộc bài viết mẫu có id là 1. vào đường dẫn request với phương thức GET.

The screenshot shows the Postman interface with a GET request for "https://jsonplaceholder.typicode.com/posts/1/comments". The "Dán đường link" (Paste URL) field contains the copied URL. The "Gửi request" (Send request) button is highlighted. The request is labeled "11\_phuong".

3.1.2. Bước 2: Sau khi nhấn send để gửi request ta được kết quả.

```

[{"postId": 1, "id": 1, "name": "id labore ex et quam laborum", "email": "Eliseo@gardner.biz", "body": "Iaudantium enim quasi est quidem magnam voluptate ipsam eos\\ntempora quo necessitatibus\\ndolor quam autem quasi\\nreiciendis et nam sapiente accusantium"}, {"postId": 1, "id": 2, "name": "id labore ex et quam laborum", "email": "Eliseo@gardner.biz", "body": "Iaudantium enim quasi est quidem magnam voluptate ipsam eos\\ntempora quo necessitatibus\\ndolor quam autem quasi\\nreiciendis et nam sapiente accusantium"}]
  
```

- Dữ liệu trả về có postId là 1 và id là mã comment thuộc post có id là 1.

3.1.3. Bước 3: Viết test script để kiểm tra phản hồi của postman, ta cần vào mục Post-response ở mục Scripts.

3.1.4. Bước 4: Ở nơi viết test script ta thực thi đoạn code sau để kiểm tra response trả về phía server.

```

// Kiểm tra status code là 200
pm.test("TC1_Status_11_phuong", function () {
    pm.response.to.have.status(200);
});

// Kiểm tra đúng định dạng email
pm.test("TC2_ValidateEmailFormat_11_phuong", function () {
    var jsonData_11_phuong = pm.response.json();
    jsonData_11_phuong.forEach((item_11_phuong) => {
        pm.expect(item_11_phuong.email).to.match(/[^@\s@]+\@\[^@\s@]+\.[^\s@]+\$/);
    });
});

// Test - fail vì email này không có trong response
pm.test("TC3_CheckEmailExist_11_phuong", function () {
    var jsonData_11_phuong = pm.response.json();
    var emailExists_11_phuong = jsonData_11_phuong.some((item_11_phuong) => item_11_phuong.email ===
"nglhongphuong@gmail.com");
    pm.expect(emailExists_11_phuong).to.be.true;
});
  
```

- Kết quả thực thi sau khi nhấn nút send.

The screenshot shows the Postman interface with a successful API request to `https://jsonplaceholder.typicode.com/posts/1/comments`. The 'Test Results' section displays three test cases: TC1\_Status\_11\_phuong (PASSED), TC2\_ValidateEmailFormat\_11\_phuong (PASSED), and TC3\_CheckEmailExist\_11\_phuong (FAILED). The failed test case includes an Assertion Error message: `AssertionError: expected false to be true`.

### 3.1.5. Bước 5: Lưu request hiện tại đang làm việc.

- Nhấn nút save.

The screenshot shows the Postman interface with the 'Save' button highlighted. The 'Lưu lại request đang thực hiện hiện tại' (Save current running request) button is also highlighted. The interface shows the same API request and test results as the previous screenshot.

- Đặt tên request và chọn thư mục lưu.

The screenshot shows the 'SAVE REQUEST' dialog in Postman. The 'Request name' field contains the value 'GET\_post/1/comments/\_11\_phuong'. The 'Save to' section shows a list of collections/folders, with 'RESTful API Basics #blueprint' selected and highlighted with a blue border. The 'Chọn thư mục lưu trữ' (Select storage folder) button is also highlighted. At the bottom right are 'Save' and 'Cancel' buttons.

### 3.1.6. Bước 6: Ta nhấn nút SAVE -> Kết quả sau khi lưu.

The screenshot shows the Postman interface with a collection named "phuong\_11\_TestAPI". A GET request is selected with the URL "https://jsonplaceholder.typicode.com/posts/1/comments/\_11\_phuong". The response pane displays a single comment object:

```

{
  "id": 1,
  "userId": 1,
  "body": "quia et suscipit\nrecusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnnostrum rerum est autem sunt rem",
  "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit"
}

```

### 3.2. Tạo và lưu một request: post.

3.2.1. Bước 1: Ta thêm request mới có phương thức post tương tự, ta dán link <https://jsonplaceholder.typicode.com/posts> trên url của request, được kết quả.

The screenshot shows a POST request to "https://jsonplaceholder.typicode.com/posts". The "Body" tab is selected, showing an empty form-data structure:

Key	Value	Description	Bulk Edit
Key	Text	Value	

3.2.2. Bước 2: Để post dữ liệu từ phía client lên server, ta cần xem nội dung của post gồm có những giá trị truyền nào.

The screenshot shows a browser window displaying the JSON response for "https://jsonplaceholder.typicode.com/posts". The response is a list of three posts:

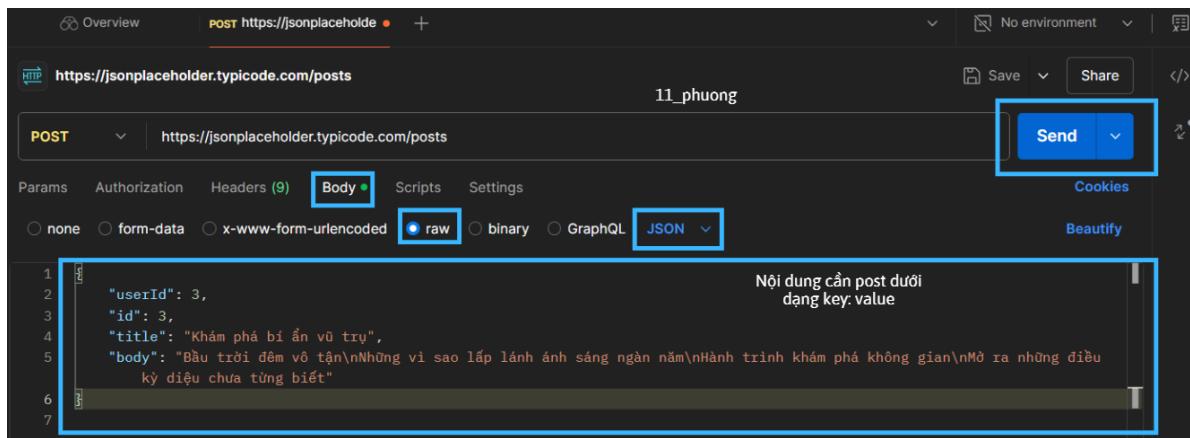
```

[
  {
    "userId": 1,
    "id": 1,
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
    "body": "quia et suscipit\nrecusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnnostrum rerum est autem sunt rem"
  },
  {
    "userId": 1,
    "id": 2,
    "title": "qui est esse",
    "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nnfugiat blanditiis voluptate porro vel nihil molestiae ut\naperiam non debitis possimus qui neque nisi nulla"
  },
  {
    "userId": 1,
    "id": 3,
    "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",
    "body": "et iusto sed quo iure\nvoluptatem occaecati omnis eligendi aut ad\nvoluptatem doloribus vel accusantium quis pariatur\nmolestiae porro eius odie"
  }
]

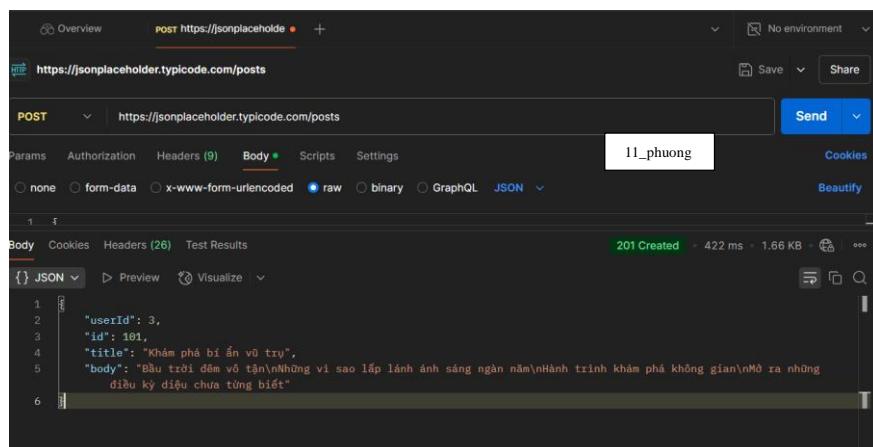
```

- Dựa vào thông tin trên ta thấy rằng, post lên một bài viết mẫu mới - post, cần truyền các key lần lượt là userId, id, title và body.

3.2.3. Bước 3: Thực hiện nội dung post cần truyền ở mục Body -> Raw -> JSON.



3.2.4. Bước 4: Sau khi nhấn nút send để post lên phía server ta nhận được kết quả response từ server mã trạng thái http 201 đã được tạo thành công.



3.2.5. Bước 5: Thực hiện viết test script ở mục Scripts phần post-response, kiểm tra trạng thái trả về và dữ liệu truyền có thuộc tính Description không.

- Source code:

```
pm.test("TC1_Status_11_phuong", function () {
    pm.expect(pm.response.code).to.be.oneOf([200, 201]);
});
// Kiểm tra dữ liệu truyền lên có thuộc tính Description không? -> Fail
pm.test("TC2_Response_Body_11_phuong", function () {
    let jsonData_11_phuong = pm.response.json();
    pm.expect(jsonData_11_phuong).to.have.property("Description");
});
```

- Kết quả sau khi nhấn nút send.

Nội viết test script

Kết quả trả về

```

1 pm.test("TC1_Status_11_phuong", function () {
2     pm.expect(pm.response.code).to.be.oneOf([200, 201]);
3 });
4 // Kiểm tra dữ liệu truyền lên có thuộc tính Description không? -> Fail
5 pm.test("TC2_Response_Body_11_phuong", function () {
6     let jsonData_11_phuong = pm.response.json();
7     pm.expect(jsonData_11_phuong).to.have.property("Description");
8 });

```

3.2.6. Bước 6: Lưu request post đang thực hiện, các bước thực hiện tương tự [3.1.5. Bước 5](#):  
**Lưu request hiện tại đang làm việc**. Ta nhấn nút save và đặt tên POST\_post\_11\_phuong  
- Kết quả thực hiện:

### 3.3. Tạo và lưu một request: put.

3.3.1. Bước 1: Ta chọn dấu cộng để thêm mới request có phương thức PUT - cập nhập toàn bộ thông tin, dán đường link <https://jsonplaceholder.typicode.com/posts/1> để cập nhập toàn bộ thông tin của post có id là 1.

Dán đường link

Gửi request

3.3.2. Bước 2: Thực hiện cập nhập toàn bộ nội dung của post có id=1 ở mục Body -> Raw -> JSON.

- Ta có thể thay đổi được nội dung bao gồm userId, id, title và body.

```

1
2
3
4
5
6
{
  "userId": 1,
  "id": 1,
  "title": "Tiêu đề mới được cập nhật",
  "body": "Nội dung bài viết đã được thay đổi hoàn toàn."
}

```

- Kết quả sau khi nhấn send, trạng thái http trả về 200 thành công.

```

1
2
3
4
5
6
{
  "userId": 1,
  "id": 1,
  "title": "Tiêu đề mới được cập nhật",
  "body": "Nội dung bài viết đã được thay đổi hoàn toàn."
}

```

3.3.3. Bước 3: Thực hiện viết test script để kiểm tra trạng thái trả về sau khi put, và có đúng id đã gửi lên không.

- Source code:

```

// Kiểm tra mã trạng thái trả về là 200 hoặc 204 (thành công)
pm.test("TC1_Status_11_phuong", function () {
    pm.expect(pm.response.code).to.be.oneOf([200, 204]);
});

// Kiểm tra phản hồi có đúng ID đã gửi lên không
pm.test("TC2_Correct_ID_11_phuong", function () {
    let jsonData_11_phuong = pm.response.json();
    pm.expect(jsonData_11_phuong.id).to.eq(1);
});

```

- Kết quả thực hiện:

The screenshot shows the Postman interface with a successful PUT request to <https://jsonplaceholder.typicode.com/posts/1>. The 'Scripts' tab is active, displaying the following test code:

```

1 // Kiểm tra mã trạng thái trả về là 200 hoặc 204 (thành công)
2 pm.test("TC1_Status_11_phuong", function () {
3     pm.expect(pm.response.code).to.be.oneOf([200, 204]);
4 });
5
6 // Kiểm tra phản hồi có đúng ID đã gửi lên không
7 pm.test("TC2_Correct_ID_11_phuong", function () {
8     let jsonData_11_phuong = pm.response.json();
9     pm.expect(jsonData_11_phuong.id).to.eql(1);
10 });

```

The 'Test Results' section shows two green 'PASSED' entries:

- TC1\_Status\_11\_phuong
- TC2\_Correct\_ID\_11\_phuong

A callout box highlights the 'Kết quả trả về' (Test results) area.

3.3.4. Bước 4: Lưu request put tương tự các bước thực hiện tương tự [3.1.5. Bước 5: Lưu request hiện tại đang làm việc](#). Ta nhấn nút save và đặt tên PUT\_post/1\_11\_phuong.  
- Kết quả thực hiện.

The screenshot shows the Postman interface with a saved request named 'PUT\_post/1\_11\_phuong' in the 'phuong\_11\_TestAPI' workspace. The request URL is <https://jsonplaceholder.typicode.com/posts/1>. The response status is 200 OK. The request body is a JSON object:

```

1 {
2     "userId": 1,
3     "id": 1,
4     "title": "Tiêu đề mới được cập nhật",
5     "body": "Nội dung bài viết đã được thay đổi hoàn toàn."

```

### 3.4. Tạo và lưu một request: delete.

3.4.1. Bước 1: Ta chọn dấu cộng để thêm mới request có phương thức DELETE và dán link ở đường URL <https://jsonplaceholder.typicode.com/posts/2> để xóa post có id=2.

Dán đường link

11\_phuong

Send

Gửi request

- 3.4.2. Bước 2: Ta nhấn nút send để kiểm tra đã xóa thành công posts có id=2 không.
- Kết quả: 200 thành công và thấy dữ liệu được trả về {} rỗng.

11\_phuong

Trạng thái thành công 200 OK

Dữ liệu từ phía server trả về

{}

- 3.4.3. Bước 3: Thực hiện viết test script để kiểm tra trả về là rỗng và có thuộc tính message không.

- Source code:

```
// Kiểm tra phản hồi không chứa nội dung (thường là empty object '{}') hoặc không có body
pm.test("TC2_Empty_Response_11_phuong", function () {
    pm.expect(pm.response.text()).to.be.oneOf(["", "{}", null]);
});

pm.test("TC2_Fail_Response_Not_Empty_11_phuong", function () {
    let jsonData_11_phuong = pm.response.json();
    pm.expect(jsonData_11_phuong).to.have.property("message");
});
```

- Kết quả thực hiện:

Chọn mục scripts

Nơi viết test script

Test Results (1/2)

PASSED TC1\_Empty\_Response\_11\_phuong

FAILED TC2\_Fail\_Response\_Not\_Empty\_11\_phuong | AssertionException: expected {} to have property 'message'

Kết quả trả về

3.4.4. Bước 4: Lưu request tương tự [3.1.5. Bước 5: Lưu request hiện tại đang làm việc.](#) Ta nhấn nút save và đặt tên DELETE\_posts/2/\_11\_phuong  
- Kết quả thực hiện:

The screenshot shows the Postman application interface. On the left, the sidebar displays a collection named "phuong\_11\_TestAPI" containing various API endpoints. The main workspace shows a single request for "DELETE\_posts/2/\_11\_phuong". The "Overview" tab is selected, showing the URL "https://jsonplaceholder.typicode.com/posts/2". The "Body" tab is active, showing a JSON response with a single object: { } (an empty object). The status bar at the bottom indicates a "200 OK" response. The overall interface is dark-themed.