

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH



BÁO CÁO MÔN HỌC

Môn học: Kiểm thử phần mềm

Ngành: Khoa học máy tính

Lớp: DH22CS02

SINH VIÊN THỰC HIỆN:

2251012095 - Trần Nguyễn Đức Minh

TP. HỒ CHÍ MINH, 2024

MỤC LỤC

CHƯƠNG 1. KIỂM THỬ UNIT TEST VỚI NUNIT CHO HỆ THỐNG TRUNG TÂM GIA SƯ	1
1.1. Xác định yêu cầu chức năng	1
1.1.1. Đặc tả chức năng Nhập danh sách học viên:	1
1.1.2. Đặc tả chức năng Xác định danh sách học viên có thẻ nhận học bổng:	2
1.2. Thiết lập dự án	2
1.2.1. Tạo project Windows Forms App (.NET Framework):	2
1.2.2. Thiết kế chức năng Nhập danh sách học viên và Xác định danh sách học viên có thẻ nhận học bổng:	4
a. Tạo giao diện nhập và hiển thị danh sách học viên:	4
b. Tạo một lớp để thiết lập đối tượng học viên:	7
1.2.3. Thiết kế test case cho từng chức năng:	10
1.2.4. Kiểm thử chức năng:	11
a. Tạo project Unit Test kiểm thử các chức năng:	11
b. Tạo file .csv kiểm thử:	16
c. Thêm dự án cần test vào file kiểm thử:	20
d. Kết quả thực hiện:	22
CHƯƠNG 2. KIỂM THỬ SELENIUM VỚI TRANG WEBSITE PINTEREST	24
2.1. Tổng quan	24
2.1.1. Giới thiệu:	24
2.1.2. Đặc tả yêu cầu:	24
a. Chức năng đăng nhập:	24
b. Chức năng tải hình ảnh về:	25
c. Chức năng chia sẻ hình ảnh (Copy link):	26
2.1.3. Thiết kế test case:	27
2.2. Thiết lập dự án	27
2.2.1. Tạo một project Unit Test Project (.NET Framework):	27

2.2.2. Thêm thư viện Selenium vào project:	29
2.3. Kiểm thử chức năng.....	31
2.3.1. Chuẩn bị website trong trình duyệt:	31
2.3.2. Chức năng đăng nhập:.....	33
a. Đăng nhập thành công:	33
b. Đăng nhập với mật khẩu sai:	39
c. Đăng nhập với email không tồn tại:	43
d. Đăng nhập với email không hợp lệ:	47
e. Đăng nhập khi để trống thông tin:	51
2.3.3. Chức năng tải hình ảnh về:.....	55
a. Đăng nhập và tải hình ảnh về thành công:.....	55
b. Tải hình ảnh về khi không đăng nhập:	61
2.3.4. Chức năng chia sẻ hình ảnh (Copy link):.....	64
a. Đăng nhập và chia sẻ hình ảnh (Copy link):	64
b. Chia sẻ hình ảnh (Copy link) khi không đăng nhập:	70
CHƯƠNG 3. TEST API BẰNG POSTMAN	73
3.1. Test API có sẵn.....	73
3.1.1. Đăng nhập website The Movie Database (TMDB) và xin cấp quyền truy cập API:	73
3.1.2. Tạo request để test API:	79
3.1.3. Viết các API Test Case:	89
3.2. TẠO API BẰNG NODE.JS	94
3.2.1. Tạo API bằng Node.js:	94
3.2.2. Tạo request để test API:	103
3.2.3. Viết các API Test Case:	107

DANH MỤC HÌNH VẼ

Hình 1.1. Các yêu cầu và chức năng trong hệ thống của đề tài (31_Minhh)	1
Hình 1.2. Các bước tạo một project Windows Form App (.NET Framework) (31_Minhh)	4
Hình 1.3. Giao diện của hệ thống (31_Minhh)	7
Hình 1.4. Các bước thiết lập đối tượng học viên (31_Minhh)	10
Hình 1.5. Test case của chức năng Nhập danh sách học viên (31_Minhh)	10
Hình 1.6. Test case của chức năng Xác định danh sách học viên có thể nhận học bổng (31_Minhh)	10
Hình 1.7. Các bước tạo một project Unit Test (31_Minhh)	16
Hình 1.8. Các bước tạo file .csv kiểm thử (31_Minhh)	20
Hình 1.9. Các bước thực hiện thêm dự án (31_Minhh)	22
Hình 1.10. Kết quả kiểm thử chức năng của hệ thống (31_Minhh)	23
Hình 2.1. Các test case của các chức năng kiểm thử website Pinterest (31_Minhh)	27
Hình 2.2. Các bước tạo một project Unit Test (31_Minhh)	28
Hình 2.3. Các bước thêm thư viện Selenium vào project (31_Minhh)	30
Hình 2.4. Mở trình duyệt kiểm thử bằng cách run test (31_Minhh)	32
Hình 2.5. Giao diện của website trong trình duyệt kiểm thử (31_Minhh)	32
Hình 2.6. Các bước lấy các thuộc tính liên quan đến chức năng đăng nhập (31_Minhh) ..	35
Hình 2.7. Các bước lấy thuộc tính của tên người dùng (31_Minhh)	37
Hình 2.8. Kết quả sau khi chạy TC1_LoginAccountSuccess_31_Minhh (31_Minhh)	39
Hình 2.9. Lấy thuộc tính của nội dung thông báo lỗi khi đăng nhập với mật khẩu sai (31_Minhh)	40
Hình 2.10. Kết quả sau khi chạy TC2_LoginAccountWrongPassword_31_Minhh (31_Minhh)	43
Hình 2.11. Lấy thuộc tính của nội dung thông báo lỗi khi đăng nhập với email không tồn tại (31_Minhh)	44

Hình 2.12. Kết quả sau khi chạy TC3_LoginAccountNonExistentEmail_31_Minh (31_Minh)	47
Hình 2.13. Lấy thuộc tính của nội dung thông báo lỗi khi đăng nhập với email không hợp lệ (31_Minh)	48
Hình 2.14. Kết quả sau khi chạy TC4_LoginAccountInvalidEmail_31_Minh (31_Minh)	51
Hình 2.15. Lấy thuộc tính của nội dung thông báo lỗi khi để trống thông tin đăng nhập (31_Minh)	52
Hình 2.16. Kết quả sau khi chạy TC5_LoginAccountEmptyFields_31_Minh (31_Minh)	55
Hình 2.17. Các bước lấy các thuộc tính liên quan đến chức năng tải hình ảnh về khi đăng nhập (31_Minh)	58
Hình 2.18. Kết quả sau khi chạy TC6_LoginAndSearchImageAndDownload_31_Minh (31_Minh)	61
Hình 2.19. Lấy thuộc tính liên quan đến chức năng tải hình ảnh về khi không đăng nhập (31_Minh)	62
Hình 2.20. Kết quả sau khi chạy TC7_SearchImageAndDownload_31_Minh (31_Minh)	64
Hình 2.21. Các bước lấy thuộc tính liên quan đến chức năng chia sẻ hình ảnh (Copy link) khi đăng nhập (31_Minh)	66
Hình 2.22. Cấp quyền được xem văn bản đã sao chép trong trình duyệt (31_Minh)	69
Hình 2.23. Kết quả sau khi chạy TC8_LoginAndShareLinkImage_31_Minh (31_Minh)	69
Hình 2.24. Kết quả sau khi chạy TC9_ShareLinkImage_31_Minh (31_Minh)	72
Hình 3.1. Các bước đăng nhập tài khoản và xin cấp quyền truy cập API (31_Minh)	78
Hình 3.2. Các bước tạo workspaces (31_Minh)	80
Hình 3.3. Các bước tạo một Blank collection và các request API (31_Minh)	83
Hình 3.4. Các bước lấy token để xác thực người dùng (31_Minh)	84
Hình 3.5. Các bước lấy API Reference và kết quả sau khi test API (31_Minh)	89
Hình 3.6. Các bước thực hiện API Test Case (31_Minh)	94

Hình 3.7. Các bước tạo API (31_Minh)	103
Hình 3.8. Các bước tạo request script của API vừa tạo (31_Minh)	106
Hình 3.9. Kết quả sau khi gửi request (31_Minh).....	107
Hình 3.10. Thực hiện Run collection các test case và kết quả sau khi chạy (31_Minh). 110	

CHƯƠNG 1. KIỂM THỬ UNIT TEST VỚI NUNIT CHO HỆ THỐNG TRUNG TÂM GIA SƯ

Một trung tâm gia sư cần quản lý thông tin học viên, một học viên bao gồm thông tin: m. số học viên, họ tên, quê quán, điểm của ba môn học chính. Vào cuối khoá học, trung tâm muốn tìm ra một số học viên có thành tích học tập tốt để trao học bổng khuyến khích. Một học viên được đánh giá là tốt nếu điểm trung bình ba môn học chính từ 8.0 trở lên và không có môn nào trong ba môn chính điểm dưới 5.

- a) Viết chương trình cho phép nhập danh sách học viên và xác định danh sách học viên có thể nhận học bổng.
- b) Viết các Unit Test để kiểm thử các chức năng của chương trình trên.

Hình 1.1. Các yêu cầu và chức năng trong hệ thống của đề tài (31_Minh)

1.1. Xác định yêu cầu chức năng

1.1.1. **Đặc tả chức năng Nhập danh sách học viên:**

- **Tên chức năng:** Nhập danh sách học viên.
- **Mô tả:** Cho phép người dùng nhập các thông tin học viên.
- **Luồng hoạt động:**
 - Bước 1: Người dùng nhập danh sách học viên (có thể nhập thủ công hoặc từ tệp dữ liệu).
 - Bước 2: Người dùng thực hiện nhấn nút thêm học viên.
 - Bước 3: Hệ thống kiểm tra tính hợp lệ của dữ liệu đầu vào hợp lệ sẽ thêm học viên vào danh sách.
- **Luồng thay thế:**
 - Người dùng để trống thông tin thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập.
 - Người dùng nhập điểm ngoài khoảng hợp lệ (dưới 0 hoặc trên 10) thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại.
 - Người dùng nhập mã học viên bị trùng với một học viên đã có trong danh sách thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại.

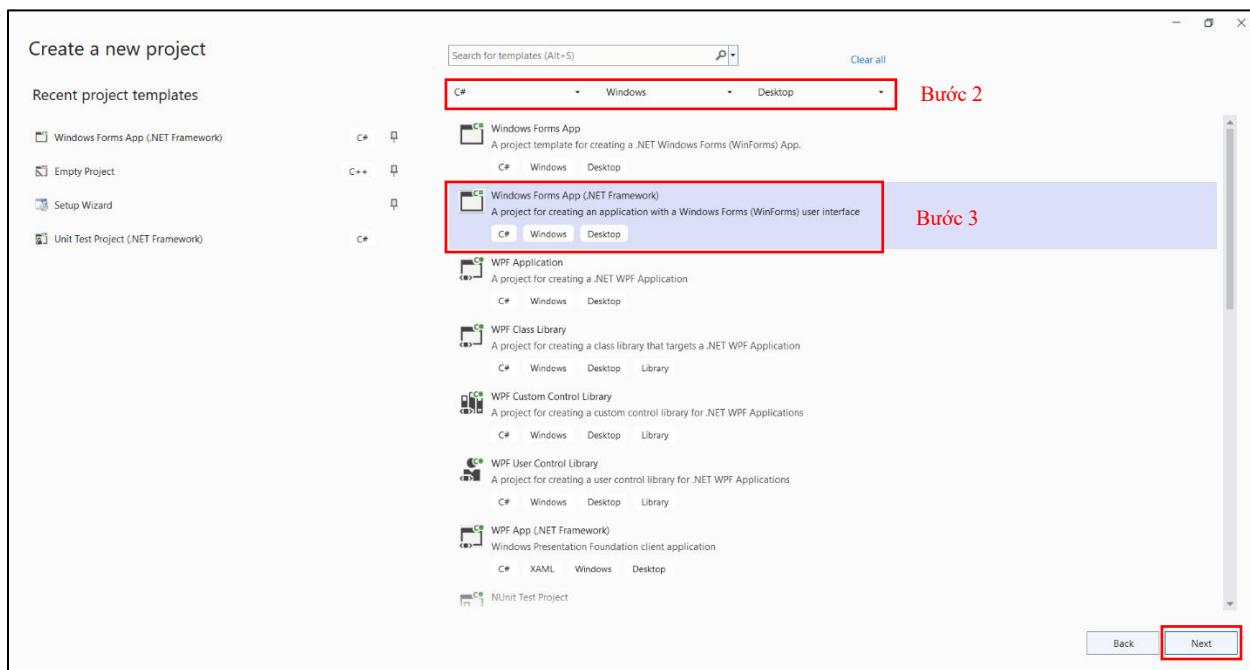
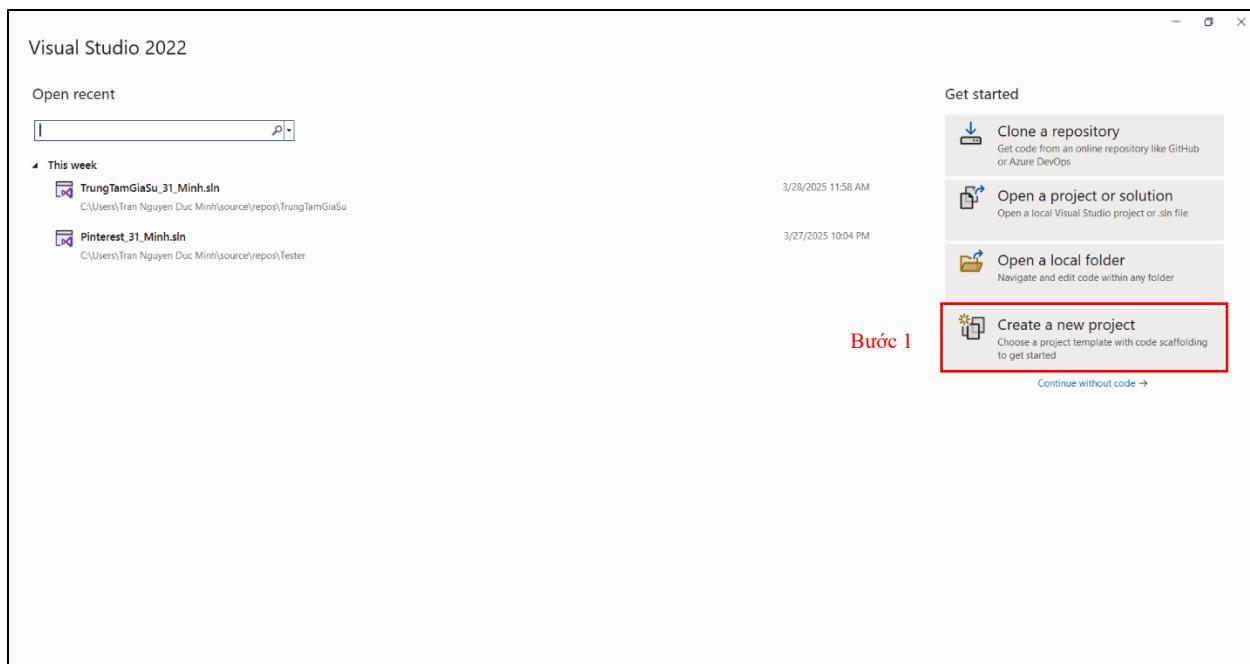
1.1.2. *Đặc tả chức năng Xác định danh sách học viên có thể nhận học bổng:*

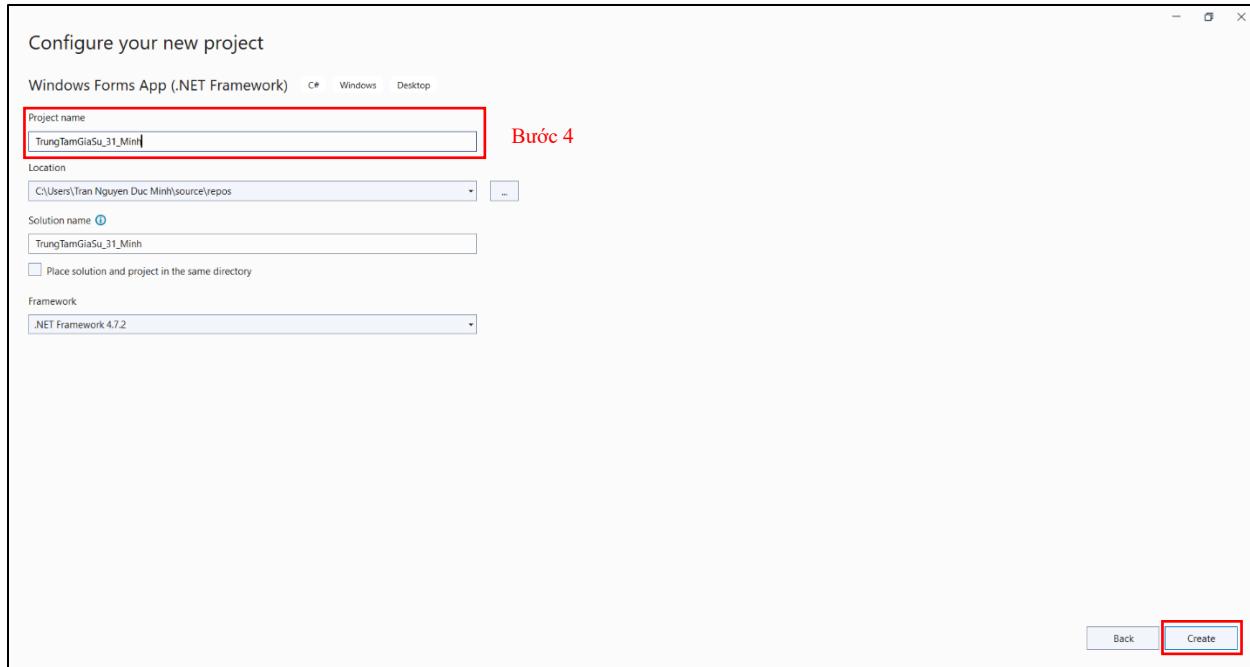
- **Tên chức năng:** Xác định danh sách học viên có thể nhận học bổng.
- **Mô tả:** Duyệt danh sách học viên, tính điểm trung bình, kiểm tra điều kiện học bổng của học viên sau đó hiển thị danh sách ra màn hình.
- **Luồng hoạt động:**
 - Bước 1: Người dùng nhấn vào nút lấy danh sách học bổng.
 - Bước 2: Hệ thống duyệt qua từng học viên trong danh sách.
 - Bước 3: Hệ thống tính điểm trung bình ba môn học chính.
 - Bước 4: Kiểm tra điều kiện để học viên nhận được học bổng.
 - Bước 5: Hiển thị danh sách học viên đủ điều kiện nhận học bổng.
- **Luồng thay thế:** Không có.

1.2. Thiết lập dự án

1.2.1. *Tạo project Windows Forms App (.NET Framework):*

- Bước 1: Mở **Visual Studio 2022** > Chọn **Create a new project**
- Bước 2: Tìm kiếm project với các thông tin **C#, Windows, Desktop**
- Bước 3: Chọn **Windows Forms App (.NET Framework)** > Next
- Bước 4: Đặt tên project là **TrungTamGiaSu_31_Minh** (Có thể thay thế vị trí source code bằng cách thay đổi **Location**) > Create





Hình 1.2. Các bước tạo một project Windows Form App (.NET Framework) (31_Minh)

1.2.2. Thiết kế chức năng Nhập danh sách học viên và Xác định danh sách học viên có thẻ nhận học bỗng:

a. Tạo giao diện nhập và hiển thị danh sách học viên:

- Viết code giao diện:

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Windows.Forms;

namespace TrungTamGiaSu_31_Minh
{
    public partial class Form1 : Form
    {
        //Tạo danh sách đối tượng học viên
        public List<Student_31_Minh> listStudents_31_Minh = new
List<Student_31_Minh>();
        public Form1()
        {
            InitializeComponent();

            //Gọi hàm load dữ liệu vào DataGridView
            LoadData_31_Minh();

            //Gọi hàm hiển thị danh sách học viên
            showList_31_Minh();
        }
    }
}

```

```

}

private void addStudent_31_Minh_Click(object sender, EventArgs e)
{
    try
    {
        //Lấy Student ID từ textbox
        String ma_31_Minh = maHocVien_31_Minh.Text;

        //Lấy Fullname từ textbox
        String ten_31_Minh = tenHocVien_31_Minh.Text;

        //Lấy Hometown từ textbox
        String que_31_Minh = queQuan_31_Minh.Text;

        //Lấy Math Score từ textbox
        double toan_31_Minh = double.Parse(diemToan_31_Minh.Text);

        //Lấy Literature Score từ textbox
        double van_31_Minh = double.Parse(diemVan_31_Minh.Text);

        //Lấy English Score từ textbox
        double tiengAnh_31_Minh =
double.Parse(diemTiengAnh_31_Minh.Text);

        //Tạo đối tượng học viên
        Student_31_Minh std_31_Minh = new Student_31_Minh(ma_31_Minh,
ten_31_Minh, que_31_Minh, toan_31_Minh, van_31_Minh, tiengAnh_31_Minh);

        //Kiểm tra thông tin học viên
        if (std_31_Minh.isNotEmptyInfo_31_Minh() == true)
        {
            //Kiểm tra tính hợp lệ của điểm
            if (std_31_Minh.isValidScore_31_Minh() == true)
            {
                //Thêm vào danh sách học viên
                listStudents_31_Minh.Add(std_31_Minh);

                //Hiển thị danh sách học viên
                showList_31_Minh();
            }
            else
            {
                MessageBox.Show("Điểm số không hợp lệ");
            }
        }
        else
        {
            MessageBox.Show("Chưa điền đủ thông tin");
        }
    }
    catch
    {
        MessageBox.Show("Điểm số không để trống");
    }
}

private void listScholarship_31_Minh_Click(object sender, EventArgs e)

```

```

{
    //Lọc các học viên thỏa điều kiện nhận học bổng
    var scholarshipStudents_31_Minh = listStudents_31_Minh.Where(s =>
(s.isScholarship_31_Minh() == true)).ToList();
    dataGridView_31_Minh.DataSource = null;
    dataGridView_31_Minh.DataSource = scholarshipStudents_31_Minh;
}

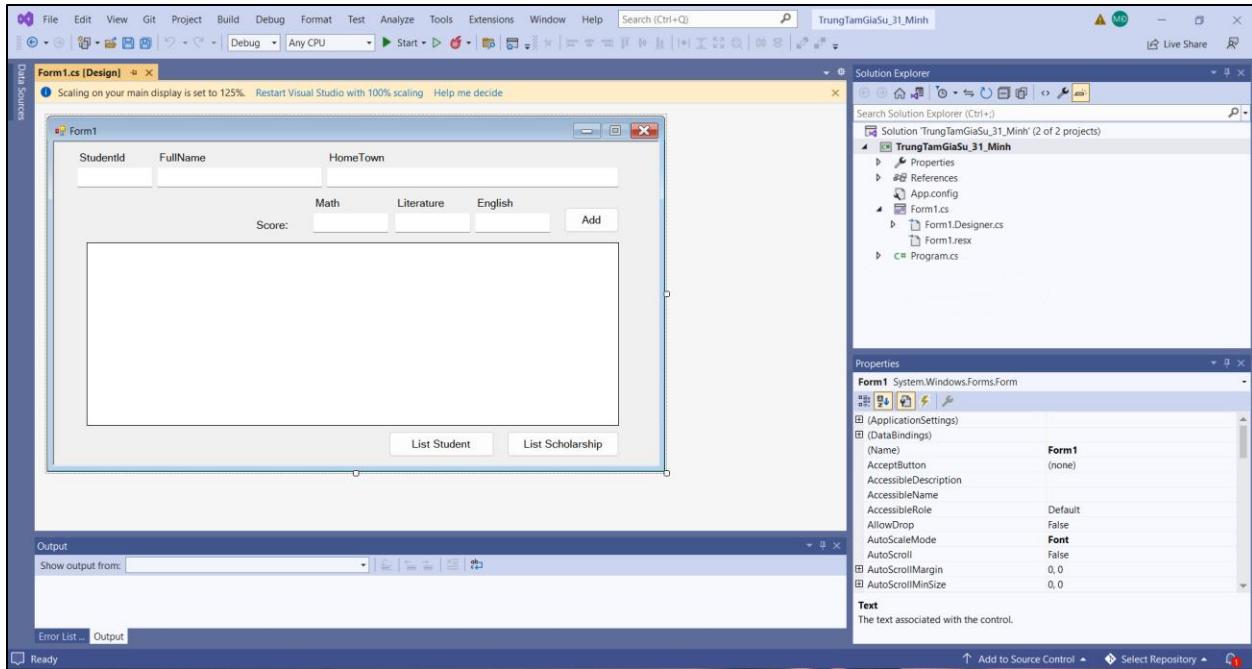
private void listStudent_31_Minh_Click(object sender, EventArgs e)
{
    showList_31_Minh();
}

//Hiển thị danh sách học viên
public void showList_31_Minh()
{
    dataGridView_31_Minh.DataSource = null;
    dataGridView_31_Minh.DataSource = listStudents_31_Minh;
}

//Kiểm tra trùng mã học viên
public bool isStudentExists_31_Minh(string studentId)
{
    return listStudents_31_Minh.Any(s => s.StudentId_31_Minh == studentId);
}

//Dữ liệu mẫu
private void LoadData_31_Minh()
{
    listStudents_31_Minh.Add(new Student_31_Minh("HV01", "Trần Nguyễn
Đức Minh", "Hà Nội", 9.1, 8.5, 8.4));
    listStudents_31_Minh.Add(new Student_31_Minh("HV02", "Nguyễn Thị
Thúy Hạnh", "TP. HCM", 7.8, 6.3, 8.7));
    listStudents_31_Minh.Add(new Student_31_Minh("HV03", "Lê Văn Hiền",
"Đà Nẵng", 5.9, 5.2, 6.1));
    listStudents_31_Minh.Add(new Student_31_Minh("HV04", "Trần Xuân
Hưng", "Cần Thơ", 8.6, 8.5, 9.7));
    listStudents_31_Minh.Add(new Student_31_Minh("HV05", "Ngô Tấn Tài",
"Bình Dương", 3.6, 7.8, 6.6));
    listStudents_31_Minh.Add(new Student_31_Minh("HV06", "Huỳnh Quốc
Ánh", "Nha Trang", 7.9, 7.0, 10));
    listStudents_31_Minh.Add(new Student_31_Minh("HV07", "Hứa Quang
Sáng", "Nghệ An", 8.3, 9.4, 8.8));
    listStudents_31_Minh.Add(new Student_31_Minh("HV08", "Cao Anh Thơ",
"Huế", 10, 8.0, 9.2));
}
}
}

```



Hình 1.3. Giao diện của hệ thống (31_Minh)

b. Tạo một lớp để thiết lập đối tượng học viên:

- Bước 1: Nhấn chuột phải vào project **TrungTamGiaSu_31_Minh**
- Bước 2: Add > New Item...
- Bước 3: Đặt tên lớp là **Student_31_Minh** > Add
- Bước 4: Viết code lớp **Student_31_Minh**:

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

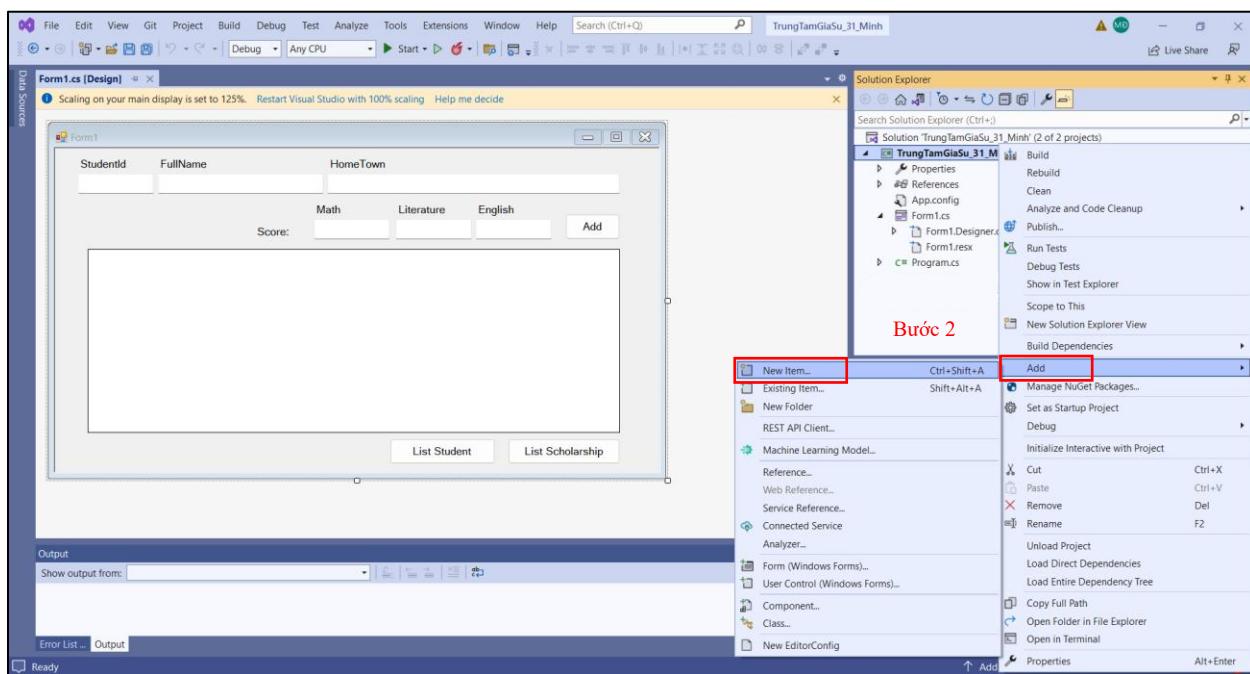
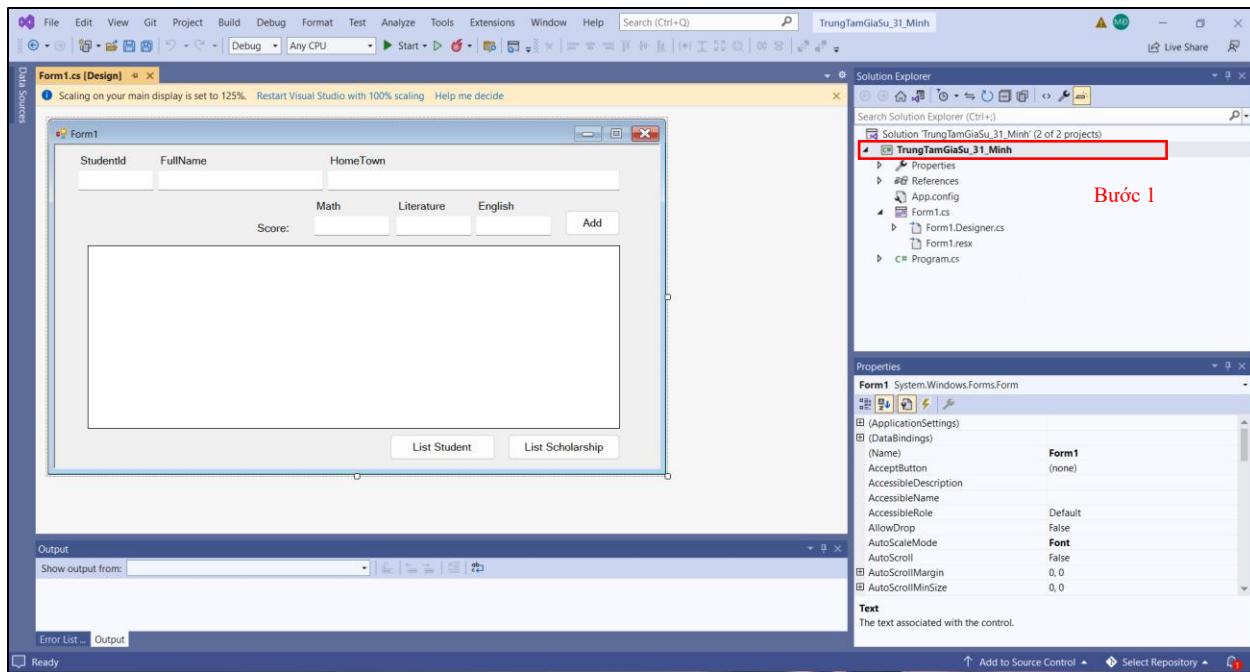
namespace TrungTamGiaSu_31_Minh
{
    public class Student_31_Minh
    {
        public string StudentId_31_Minh { get; set; }
        public string FullName_31_Minh { get; set; }
        public string HomeTown_31_Minh { get; set; }
        public double MathScore_31_Minh { get; set; }
        public double LiteratureScore_31_Minh { get; set; }
        public double EnglishScore_31_Minh { get; set; }

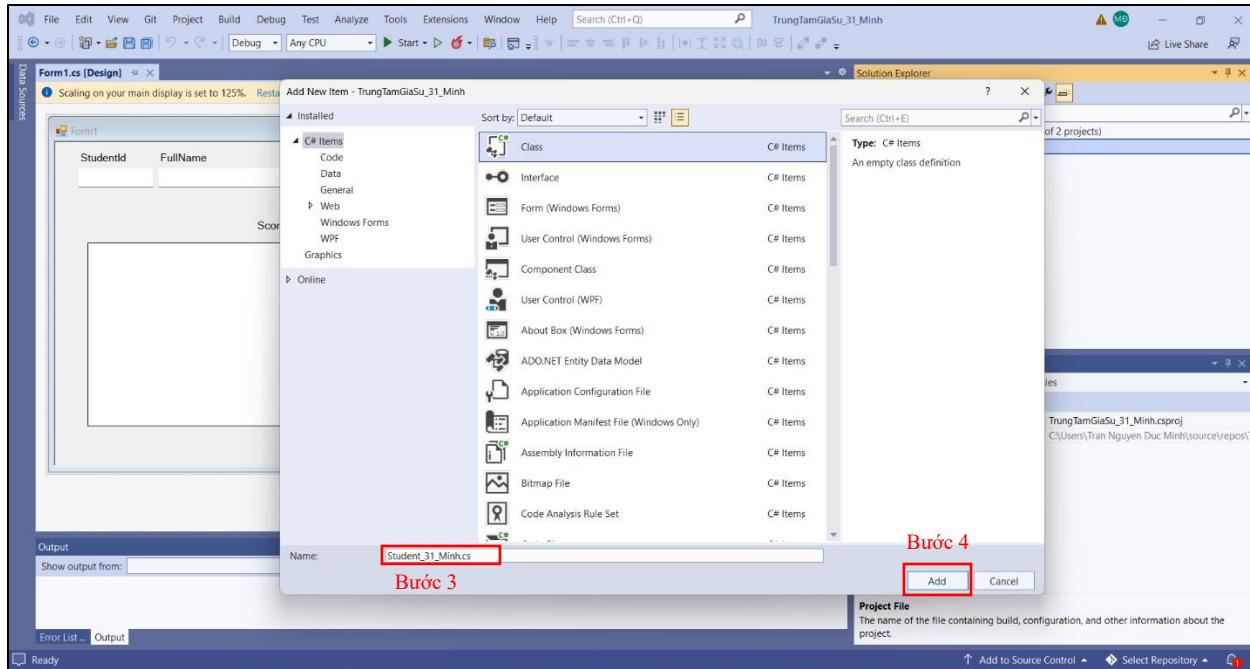
        //Phương thức khởi tạo
    }
}
```

```
public Student_31_Minh(string studentId_31_Minh, string
fullName_31_Minh, string homeTown_31_Minh, double mathScore_31_Minh, double
literatureScore_31_Minh, double englishScore_31_Minh)
{
    StudentId_31_Minh = studentId_31_Minh;
    FullName_31_Minh = fullName_31_Minh;
    HomeTown_31_Minh = homeTown_31_Minh;
    MathScore_31_Minh = mathScore_31_Minh;
    LiteratureScore_31_Minh = literatureScore_31_Minh;
    EnglishScore_31_Minh = englishScore_31_Minh;
}

//Kiểm tra học viên có được nhận học bổng hay không
public bool isScholarship_31_Minh()
{
    double avg_31_Minh = (MathScore_31_Minh + LiteratureScore_31_Minh +
EnglishScore_31_Minh) / 3.0;
    return avg_31_Minh >= 8.0 && MathScore_31_Minh >= 5.0 &&
LiteratureScore_31_Minh >= 5.0 && EnglishScore_31_Minh >= 5.0;
}

//Kiểm tra thông tin hợp lệ của học viên
public bool isNotEmptyInfo_31_Minh()
{
    return StudentId_31_Minh != String.Empty && FullName_31_Minh !=
String.Empty && HomeTown_31_Minh != String.Empty;
}
//Kiểm tra tính hợp lệ của điểm số
public bool isValidScore_31_Minh()
{
    return MathScore_31_Minh >= 0 && MathScore_31_Minh <= 10 &&
LiteratureScore_31_Minh >= 0 && LiteratureScore_31_Minh <= 10 &&
EnglishScore_31_Minh >= 0 && EnglishScore_31_Minh <= 10;
}
}
```





Hình 1.4. Các bước thiết lập đối tượng học viên (31_Minh)

1.2.3. Thiết kế test case cho từng chức năng:

- Thiết kế test case dựa trên đặc tả chức năng Nhập danh sách học viên và Xác định danh sách học viên có thẻ nhận học bổng. Ta được form test case:

STT_31_Minh	TestCaseID_31_Minh	Input_31_Minh							KetQuaMongDoi_31_Minh	KetQuaThucTe_31_Minh	KetQua_31_Minh
1	TC01_31_Minh	HV01	Tran Nguyen Duc Minh	Ha Noi	9.1	8.5	8.4	FALSE	FALSE	Pass	
2	TC02_31_Minh	HV09	Nguyen Thi Thuy Hanh	TP.HCM	7.8	-3	8.7	FALSE	FALSE	Pass	
3	TC03_31_Minh	HV03	Le Van Hien	Da Nang	5.9	5.2	6.1	FALSE	FALSE	Pass	
4	TC04_31_Minh	HV10	Tran Xuan Hung	Can Tho	8.6	8.5	11	FALSE	FALSE	Pass	
5	TC05_31_Minh	HV05	Ngo Tan Tai	Binh Duong	3.6	7.8	6.6	FALSE	FALSE	Pass	
6	TC06_31_Minh	HV11	Huynh Quoc Anh	Nha Trang	7.9	7	10	1	1	Pass	
7	TC07_31_Minh	HV12	Hua Quang Sang	Nghia An	8.3	9.4	8.8	1	1	Pass	
8	TC08_31_Minh	HV13	Cao Anh Tho	Hue	10	8	9.2	1	1	Pass	

Hình 1.5. Test case của chức năng Nhập danh sách học viên (31_Minh)

STT_31_Minh	TestCaseID_31_Minh	StudentId_31_Minh	FullName_31_Minh	HomeTown_31_Minh	MathScore_31_Minh	LiteratureScore_31_Minh	englishScore_31_Minh	KetQuaMongDoi_31_Minh	KetQuaThucTe_31_Minh	KetQua_31_Minh
1	TC01_31_Minh	HV01	Tran Nguyen Duc Minh	Ha Noi	9.1	8.5	8.4	TRUE	TRUE	Pass
2	TC02_31_Minh	HV09	Nguyen Thi Thuy Hanh	TP.HCM	7.8	4.2	8.7	FALSE	FALSE	Pass
3	TC03_31_Minh	HV03	Le Van Hien	Da Nang	5.9	5.2	6.1	FALSE	FALSE	Pass
4	TC04_31_Minh	HV10	Tran Xuan Hung	Can Tho	8.6	8.5	9.6	TRUE	TRUE	Pass
5	TC05_31_Minh	HV05	Ngo Tan Tai	Binh Duong	3.6	7.8	6.6	FALSE	FALSE	Pass
6	TC06_31_Minh	HV11	Huynh Quoc Anh	Nha Trang	7.9	7	10	TRUE	TRUE	Pass
7	TC07_31_Minh	HV12	Hua Quang Sang	Nghia An	8.3	9.4	8.8	TRUE	TRUE	Pass
8	TC08_31_Minh	HV13	Cao Anh Tho	Hue	10	8	9.2	TRUE	TRUE	Pass

Hình 1.6. Test case của chức năng Xác định danh sách học viên có thẻ nhận học bổng (31_Minh)

1.2.4. Kiểm thử chức năng:

a. Tạo project Unit Test kiểm thử các chức năng:

- Bước 1: Nhấn chuột phải vào **Solution ‘TrungTamGiaSu_31_Minh’** > Add > New Project...
- Bước 2: Tìm kiếm theo **C#, Windows, Test** > Chọn **Unit Test Project (.NET Framework)** > Next
- Bước 3: Đặt tên project là **TrungTamGiaSuUnitTest_31_Minh** > Create
- Viết code thực hiện kiểm thử:

```

using Microsoft.VisualStudio.TestTools.UnitTesting;
using System;
using System.Collections.Generic;
using TrungTamGiaSu_31_Minh;

namespace TrungTamGiaSuUnitTest_31_Minh
{
    [TestClass]
    public class UnitTest_31_Minh
    {
        //Tạo đối tượng TestContext cho lớp UnitTest
        public TestContext TestContext { get; set; }

        private string studentId_31_Minh;
        private string fullName_31_Minh;
        private string homeTown_31_Minh;
        private double mathScore_31_Minh;
        private double literatureScore_31_Minh;
        private double englishScore_31_Minh;
        private bool isSchls_31_Minh;
        private string errorMessage = "";

        Form1 form = new Form1();

        //Kiểm thử với 7 cột và 8 hàng dữ liệu có sẵn
        //Hàng 1: studentId_31_Minh = 'HV01', fullName_31_Minh = 'Tran Nguyen
        Duc Minh', homeTown_31_Minh = 'Ha Noi',
        //mathScore_31_Minh = 9.1, literatureScore_31_Minh = 8.5,
        englishScore_31_Minh = 8.4, isSchls_31_Minh = true

        //Hàng 2: studentId_31_Minh = 'HV02', fullName_31_Minh = 'Nguyen Thi
        Thuy Hanh', homeTown_31_Minh = 'TP.HCM',
        //mathScore_31_Minh = 7.8, literatureScore_31_Minh = 6.3,
        englishScore_31_Minh = 8.7, isSchls_31_Minh = false

        //Hàng 3: studentId_31_Minh = 'HV03', fullName_31_Minh = 'Le Van Hien',
        homeTown_31_Minh = 'Da Nang',
        //mathScore_31_Minh = 5.9, literatureScore_31_Minh = 5.2,
        englishScore_31_Minh = 6.1, isSchls_31_Minh = false
    }
}

```

```

        //Hàng 4: studentId_31_Minh = 'HV04', fullName_31_Minh = 'Tran Xuan
Hung', homeTown_31_Minh = 'Can Tho',
        //mathScore_31_Minh = 8.6, literatureScore_31_Minh = 8.5,
englishScore_31_Minh = 9.7, isSchls_31_Minh = true

        //Hàng 5: studentId_31_Minh = 'HV05', fullName_31_Minh = 'Ngo Tan Tai',
homeTown_31_Minh = 'Binh Duong',
        //mathScore_31_Minh = 3.6, literatureScore_31_Minh = 7.8,
englishScore_31_Minh = 6.6, isSchls_31_Minh = false

        //Hàng 6: studentId_31_Minh = 'HV06', fullName_31_Minh = 'Huynh Quoc
Anh', homeTown_31_Minh = 'Nha Trang',
        //mathScore_31_Minh = 7.9, literatureScore_31_Minh = 7.0,
englishScore_31_Minh = 10, isSchls_31_Minh = true

        //Hàng 7: studentId_31_Minh = 'HV07', fullName_31_Minh = 'Hua Quang
Sang', homeTown_31_Minh = 'Nghe An',
        //mathScore_31_Minh = 8.3, literatureScore_31_Minh = 9.4,
englishScore_31_Minh = 8.8, isSchls_31_Minh = true

        //Hàng 8: studentId_31_Minh = 'HV08', fullName_31_Minh = 'Cao Anh Tho',
homeTown_31_Minh = 'Hue',
        //mathScore_31_Minh = 10, literatureScore_31_Minh = 8.0,
englishScore_31_Minh = 9.2, isSchls_31_Minh = true

    [TestInitialize]
    public void SetUp_31_Minh()
    {
        //Cột dữ liệu đầu tiên
        studentId_31_Minh = TestContext.DataRow[0].ToString();

        //Cột dữ liệu thứ hai
        fullName_31_Minh = TestContext.DataRow[1].ToString();

        //Cột dữ liệu thứ ba
        homeTown_31_Minh = TestContext.DataRow[2].ToString();

        //Cột dữ liệu thứ tư
        mathScore_31_Minh = double.Parse(TestContext.DataRow[3].ToString());

        //Cột dữ liệu thứ năm
        literatureScore_31_Minh =
double.Parse(TestContext.DataRow[4].ToString());

        //Cột dữ liệu thứ sáu
        englishScore_31_Minh =
double.Parse(TestContext.DataRow[5].ToString());

    }

    //Test case 1: Thực hiện thêm học viên vào danh sách (3 Pass, 5 Fail)
    //Hàng 1, 3, 5: "Mã sinh viên bị trùng" (Fail)
    //Hàng 2, 4: "Điểm không hợp lệ" (Fail)
    //Hàng 6, 7, 8: (Pass)
    [DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV",
@".\Data_31_Minh\StudentsData_31_Minh.csv", "StudentsData_31_Minh#csv",
DataAccessMethod.Sequential)]
    [TestMethod]

```

```

public void TC1_AddStudent_31_Minh()
{
    //Đếm số lượng học viên ban đầu có trong danh sách
    int init_31_Minh = form.listStudents_31_Minh.Count;

    //Tạo đối tượng học viên
    Student_31_Minh student_31_Minh = new
    Student_31_Minh(studentId_31_Minh, fullName_31_Minh, homeTown_31_Minh,
    mathScore_31_Minh, literatureScore_31_Minh, englishScore_31_Minh);

    //Điều kiện điền đủ thông tin
    if (student_31_Minh.isNotEmptyInfo_31_Minh())
    {
        //Điều kiện không có mã sinh viên bị trùng
        if (!form.isStudentExists_31_Minh(studentId_31_Minh))
        {
            //Điều kiện điểm số hợp lệ
            if (student_31_Minh.isValidScore_31_Minh())
            {
                // Thêm vào danh sách học viên
                form.listStudents_31_Minh.Add(student_31_Minh);
            }
            else
            {
                errorMessage = "Điểm không hợp lệ";
            }
        }
        else
        {
            errorMessage = "Mã học viên đã tồn tại";
        }
    }
    else
    {
        errorMessage = "Chưa điền đủ thông tin";
    }

    Console.WriteLine(init_31_Minh); //Hiển thị số lượng học viên ban
đầu có trong danh sách
    Console.WriteLine("=====");
    Console.WriteLine(form.listStudents_31_Minh.Count); //Hiển thị ra số
lượng học viên sau khi thêm học viên

    //So sánh kết quả mong muốn (1) và kết quả thực tế
    (form.listStudents_31_Minh.Count - init_31_Minh)
    Assert.AreEqual(1, form.listStudents_31_Minh.Count - init_31_Minh,
    errorMessage);
}

//Test case 2: Thực hiện lọc thành công danh sách học viên nhận học bổng
(8 Pass, 0 Fail)
[DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV",
@".\Data_31_Minh\StudentsData2_31_Minh.csv", "StudentsData2_31_Minh#csv",
DataAccessMethod.Sequential)]
[TestMethod]
public void TC2_ListScholarshipStudents_31_Minh()
{
    //Cột dữ liệu thứ bảy
}

```

```

        isSchls_31_Minh = bool.Parse(TestContext.DataRow[6].ToString());

        //Khai báo danh sách kết quả mong muốn chứa đối tượng học viên nhận
        //học bổng
        List<Student_31_Minh> expected_31_Minh = new
        List<Student_31_Minh>();
        //Khai báo danh sách kết quả thực tế chứa đối tượng học viên nhận
        //học bổng
        List<Student_31_Minh> actual_31_Minh = new List<Student_31_Minh>();

        //Tạo đối tượng học viên
        Student_31_Minh student_31_Minh = new
        Student_31_Minh(studentId_31_Minh, fullName_31_Minh, homeTown_31_Minh,
        mathScore_31_Minh, literatureScore_31_Minh, englishScore_31_Minh);

        //Điều kiện để là một học viên
        if (student_31_Minh.isNotEmptyInfo_31_Minh() == true &&
        student_31_Minh.isValidScore_31_Minh() == true)
        {
            if (isSchls_31_Minh == true)
            {
                //Thêm vào danh sách kết quả mong muốn
                expected_31_Minh.Add(student_31_Minh);
            }
            //Kiểm tra điều kiện để nhận học bổng
            if (student_31_Minh.isScholarship_31_Minh() == true)
            {
                //Thêm vào danh sách kết quả thực tế
                actual_31_Minh.Add(student_31_Minh);
            }
        }

        //Hiển thị thông tin học viên có trong danh sách kết quả mong muốn
        foreach (Student_31_Minh s_31_Minh in expected_31_Minh)
        {
            Console.WriteLine($"Student ID: {s_31_Minh.StudentId_31_Minh}, 
Fullname: {s_31_Minh.FullName_31_Minh}, Hometown: {s_31_Minh.HomeTown_31_Minh}, 
" +
                $"Math: {s_31_Minh.MathScore_31_Minh}, Literature: 
{s_31_Minh.LiteratureScore_31_Minh}, English: {s_31_Minh.EnglishScore_31_Minh}, 
" +
                $"Scholarship: {s_31_Minh.isScholarship_31_Minh()}");
        }
        Console.WriteLine("=====");

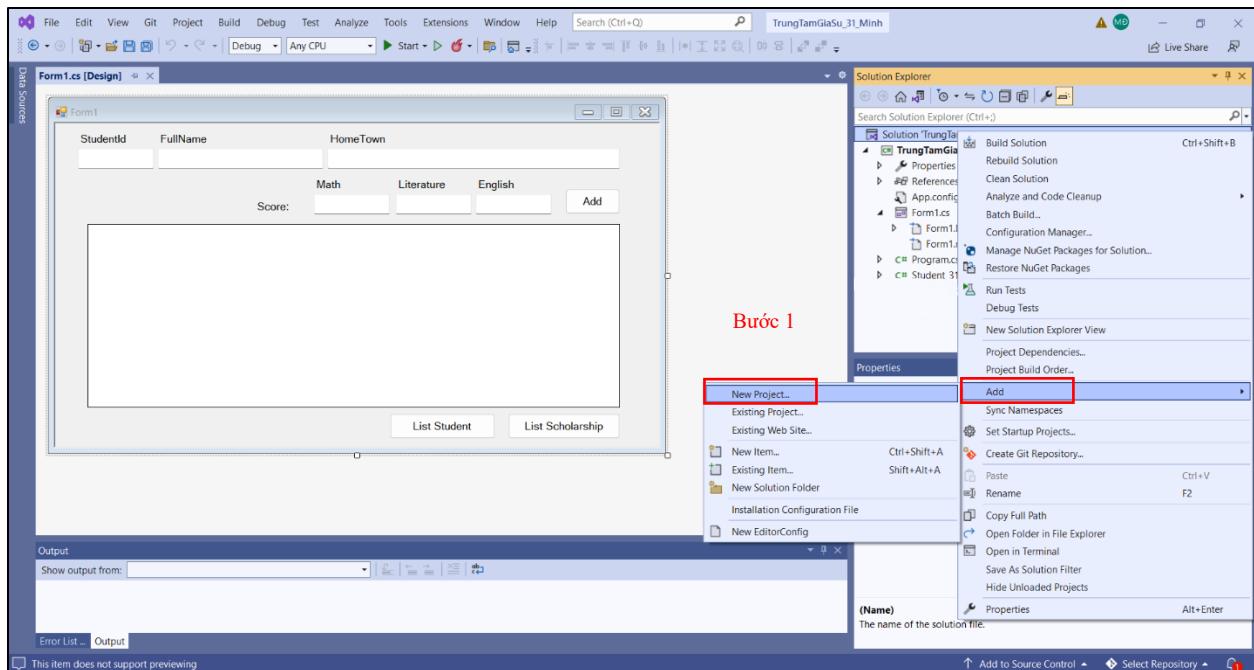
        //Hiển thị thông tin học viên có trong danh sách kết quả thực tế
        foreach (Student_31_Minh s_31_Minh in actual_31_Minh)
        {
            Console.WriteLine($"Student ID: {s_31_Minh.StudentId_31_Minh}, 
Fullname: {s_31_Minh.FullName_31_Minh}, Hometown: {s_31_Minh.HomeTown_31_Minh}, 
" +
                $"Math: {s_31_Minh.MathScore_31_Minh}, Literature: 
{s_31_Minh.LiteratureScore_31_Minh}, English: {s_31_Minh.EnglishScore_31_Minh}, 
" +
                $"Scholarship: {s_31_Minh.isScholarship_31_Minh()}");
        }
    }
}

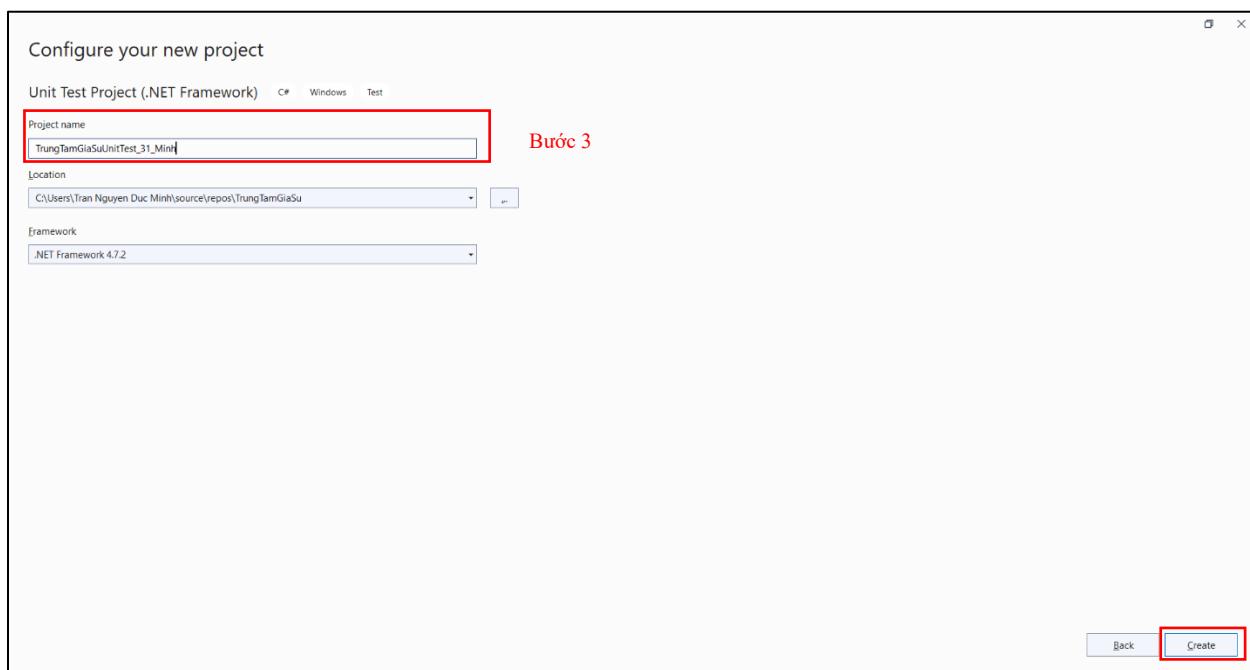
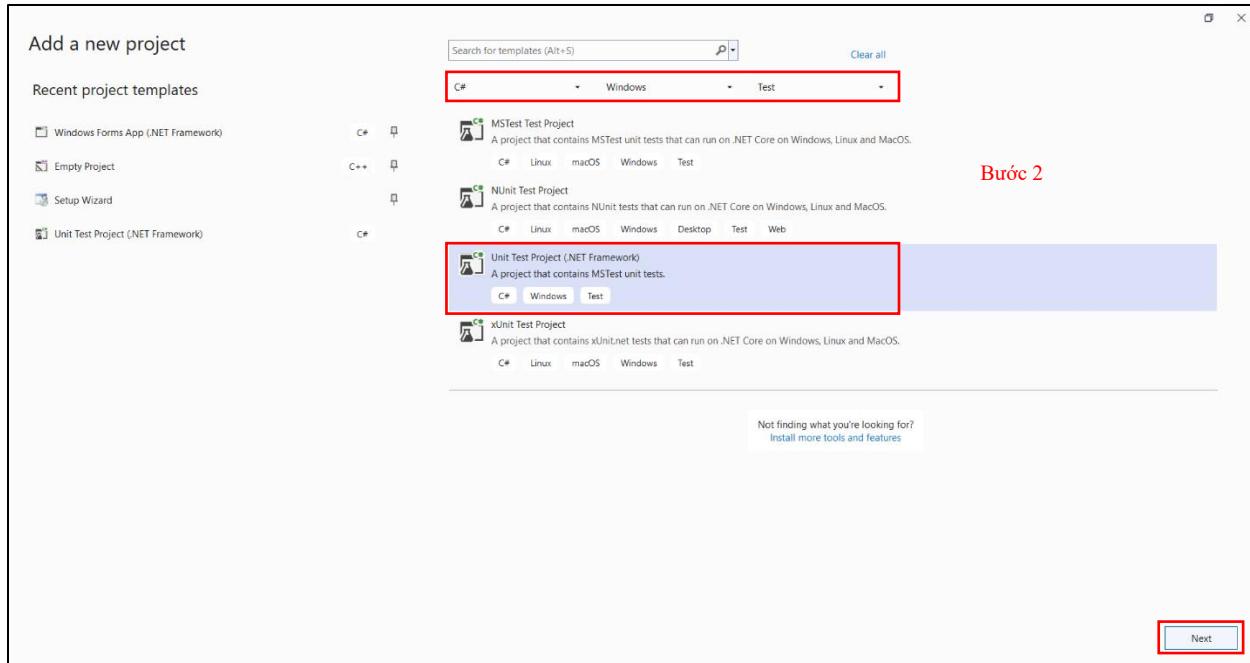
```

```

        //So sánh kết quả mong muốn (expected_31_Minh) và kết quả thực tế
        (actual_31_Minh)
        CollectionAssert.AreEqual(expected_31_Minh, actual_31_Minh, "Danh
        sách học viên nhận học bổng không giống nhau");
    }
}
}

```





Hình 1.7. Các bước tạo một project Unit Test (31_Minh)

b. Tạo file .csv kiểm thử:

- Bước 4: Nhấn chuột phải vào project Unit Test vừa tạo > Add > New Folder
- Bước 5: Đặt tên cho Folder là **Data_31_Minh**

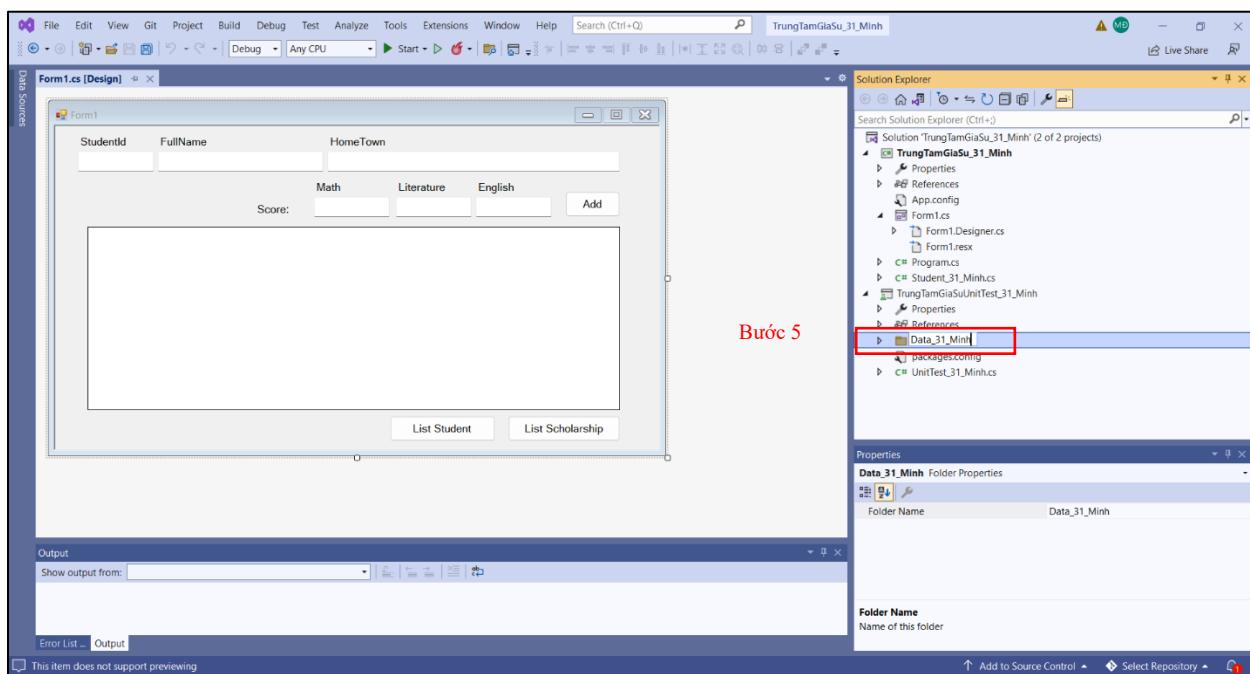
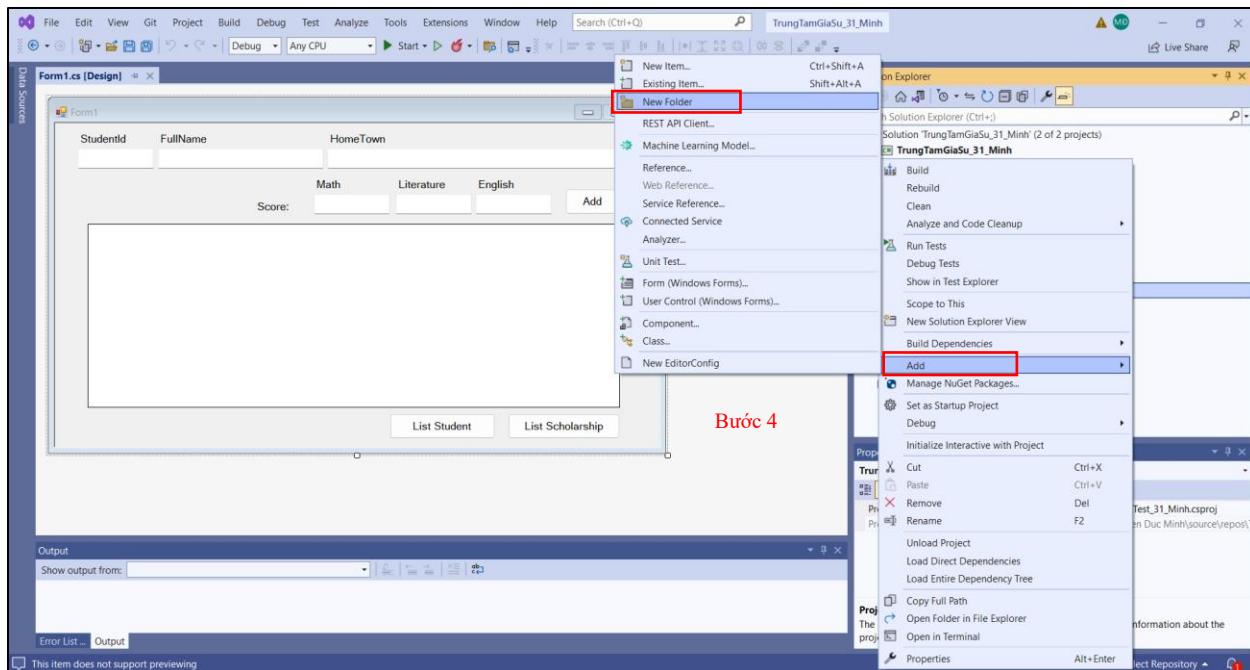
- Bước 6: Nhấn chuột phải vào Folder vừa tạo > Add > New Item...
- Bước 7: Chọn Text File > Đặt tên **StudentsData_31_Minh.csv** > Add (Làm tương tự với file **StudentsData2_31_Minh.csv**)
- Bước 8: Chuột phải vào file .csv vừa tạo > Properties
- Bước 9: Sửa thuộc tính Copy to Output Directory thành **Copy always** (Làm tương tự với file **StudentsData2_31_Minh.csv**)
- Nhập dữ liệu 2 file .csv **StudentsData_31_Minh.csv** và **StudentsData2_31_Minh.csv**:

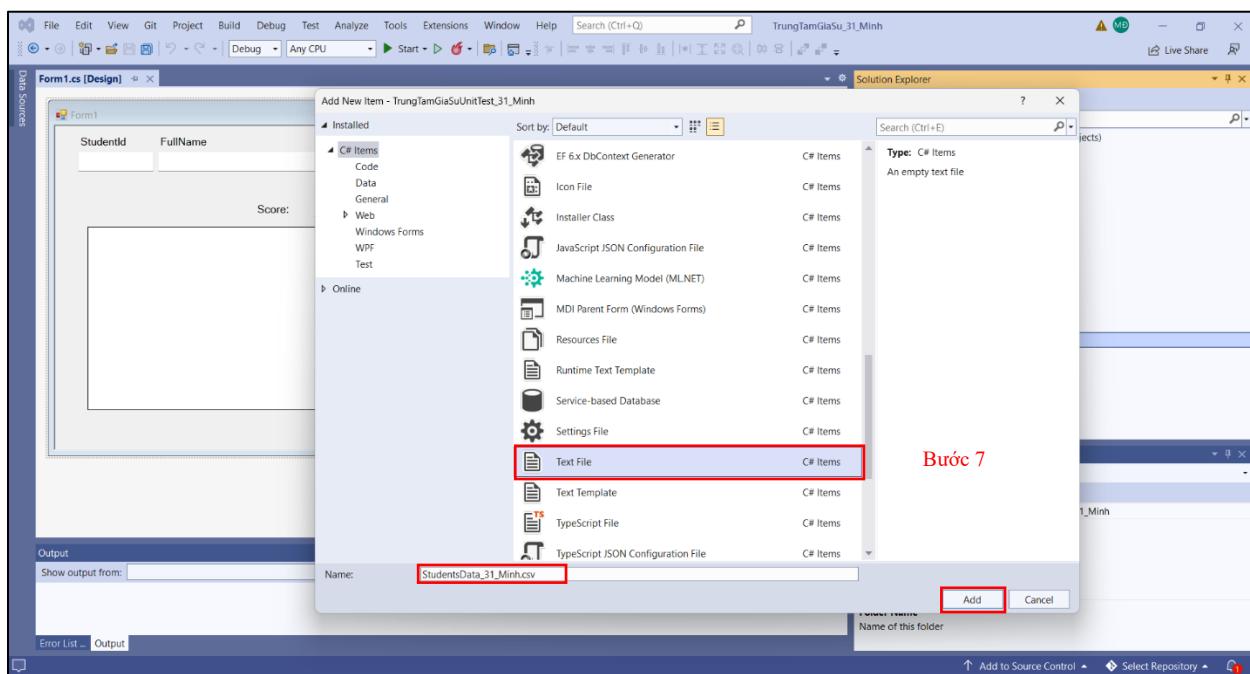
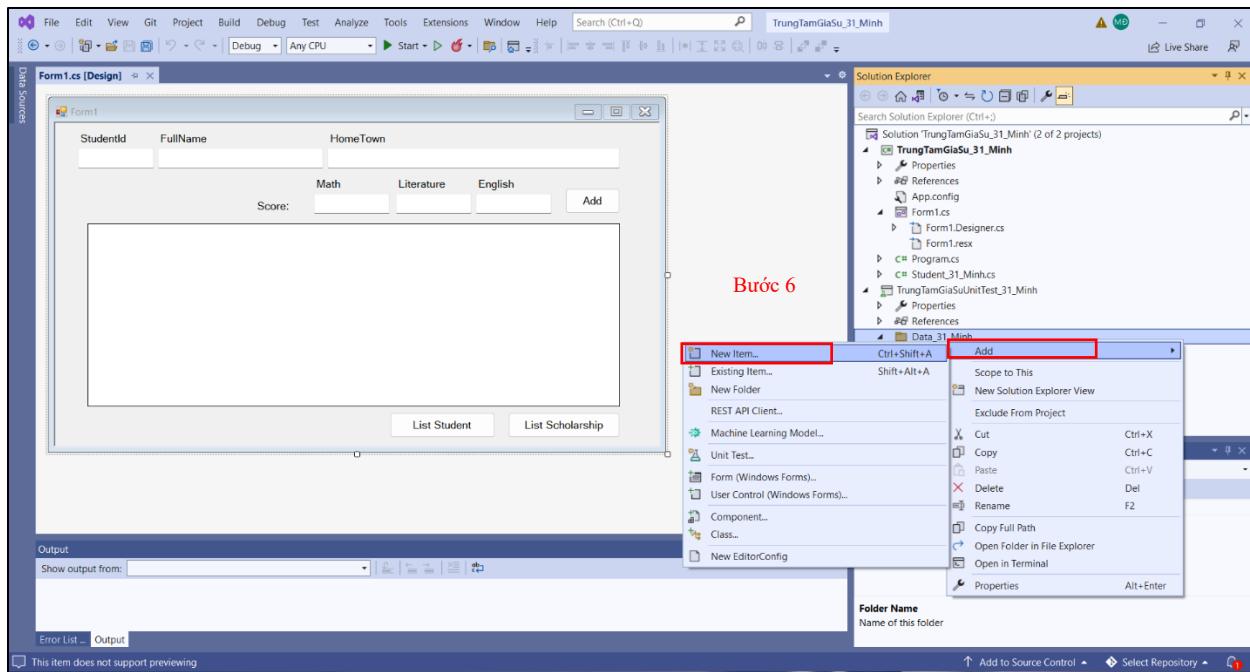
StudentsData_31_Minh.csv:

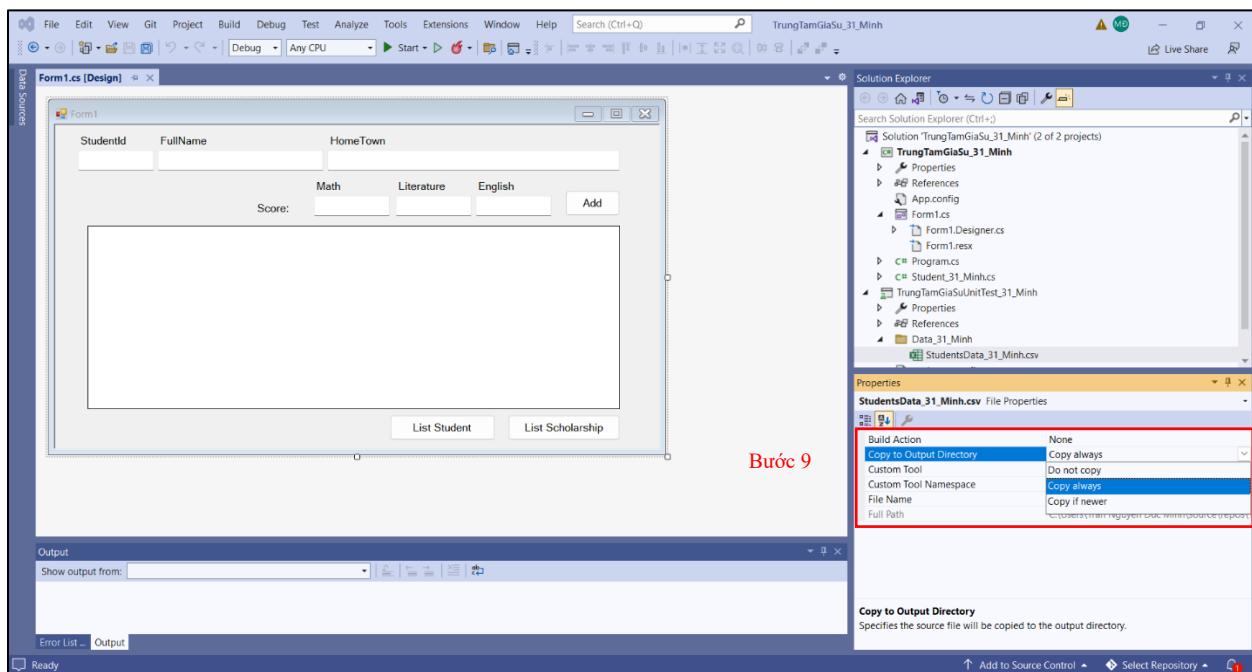
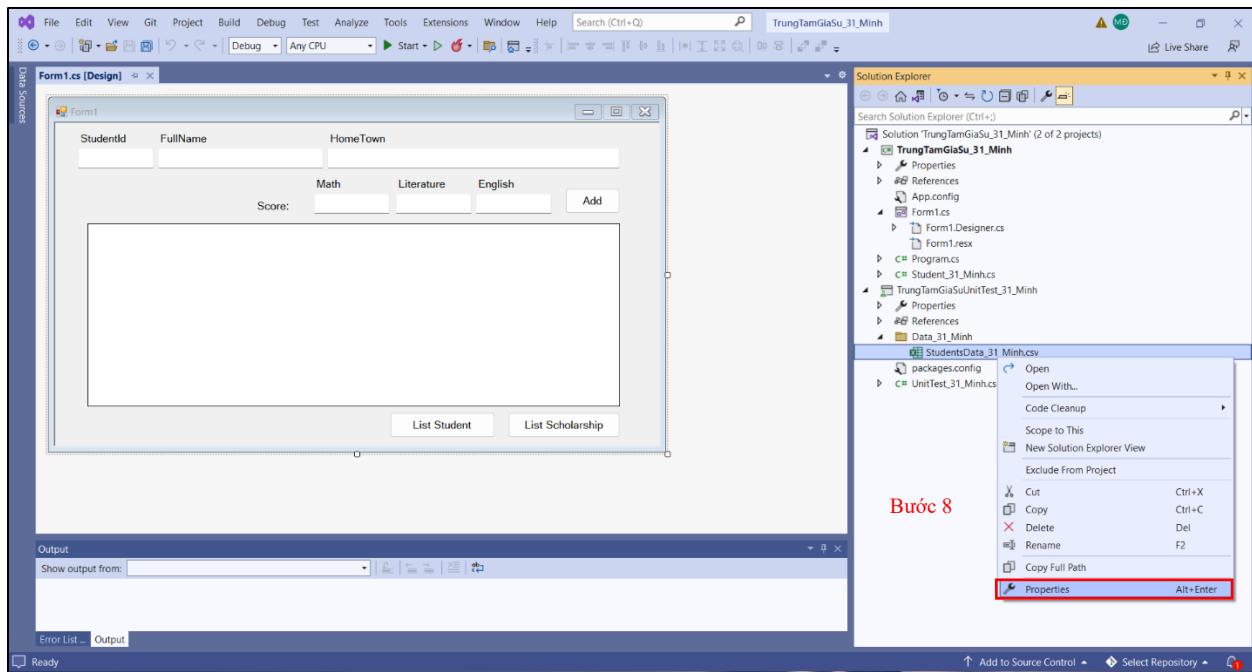
```
studentId_31_Minh,fullName_31_Minh,homeTown_31_Minh,mathScore_31_Minh,literatureScore_31_Minh,englishScore_31_Minh
HV01,Tran Nguyen Duc Minh,Ha Noi,9.1,8.5,8.4
HV09,Nguyen Thi Thuy Hanh,TP.HCM,7.8,-3,8.7
HV03,Le Van Hien,Da Nang,5.9,5.2,6.1
HV10,Tran Xuan Hung,Can Tho,8.6,8.5,11
HV05,Ngo Tan Tai,Binh Duong,3.6,7.8,6.6
HV11,Huynh Quoc Anh,Nha Trang,7.9,7.0,10
HV12,Hua Quang Sang,Nghe An,8.3,9.4,8.8
HV13,Cao Anh Tho,Hue,10,8.0,9.2
```

StudentsData2_31_Minh.csv:

```
studentId_31_Minh,fullName_31_Minh,homeTown_31_Minh,mathScore_31_Minh,literatureScore_31_Minh,englishScore_31_Minh,isScholarship_31_Minh
HV01,Tran Nguyen Duc Minh,Ha Noi,9.1,8.5,8.4,true
HV09,Nguyen Thi Thuy Hanh,TP.HCM,7.8,4.2,8.7,false
HV03,Le Van Hien,Da Nang,5.9,5.2,6.1,false
HV10,Tran Xuan Hung,Can Tho,8.6,8.5,9.6,true
HV05,Ngo Tan Tai,Binh Duong,3.6,7.8,6.6,false
HV11,Huynh Quoc Anh,Nha Trang,7.9,7.0,10,true
HV12,Hua Quang Sang,Nghe An,8.3,9.4,8.8,true
HV13,Cao Anh Tho,Hue,10,8.0,9.2,true
```





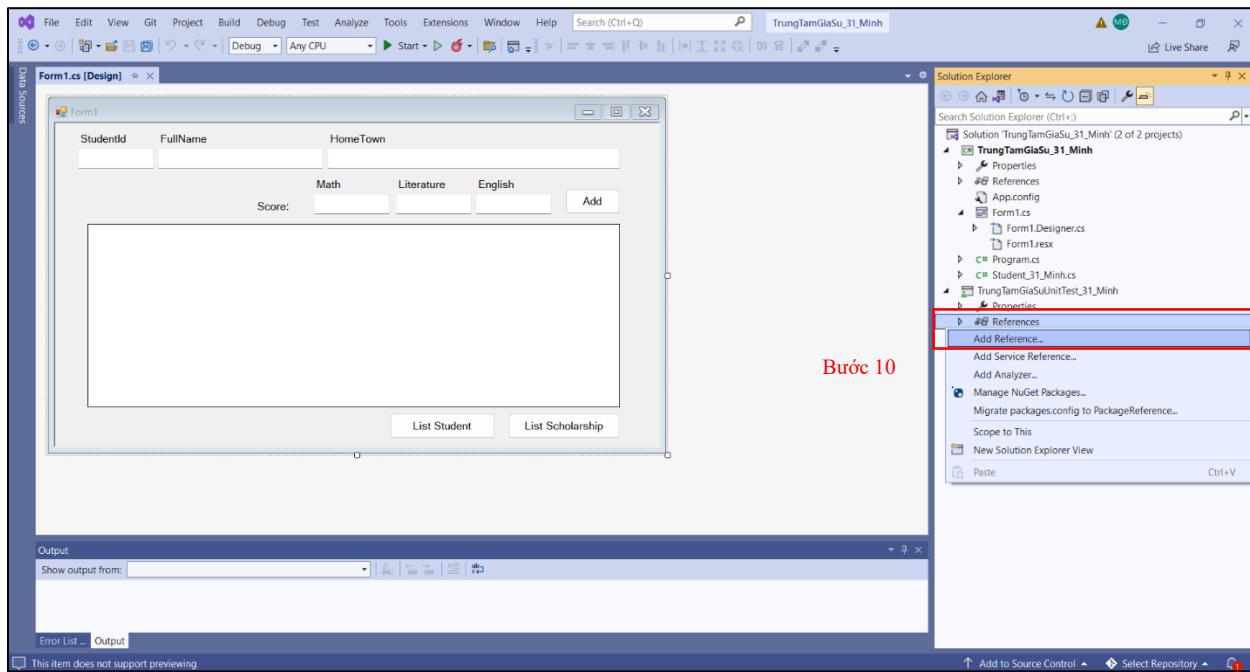


Hình 1.8. Các bước tạo file .csv kiểm thử (31_Minh)

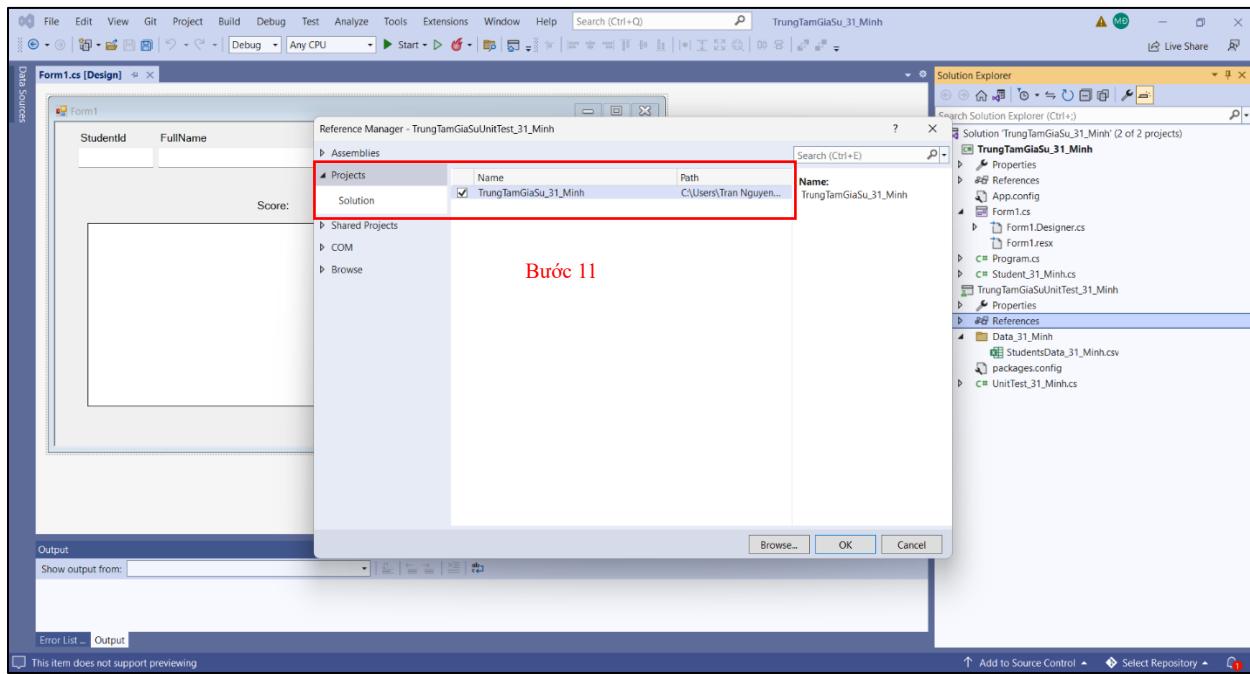
c. Thêm dự án cần test vào file kiểm thử:

- Bước 10: Nhấn chuột phải vào References > Add Reference...

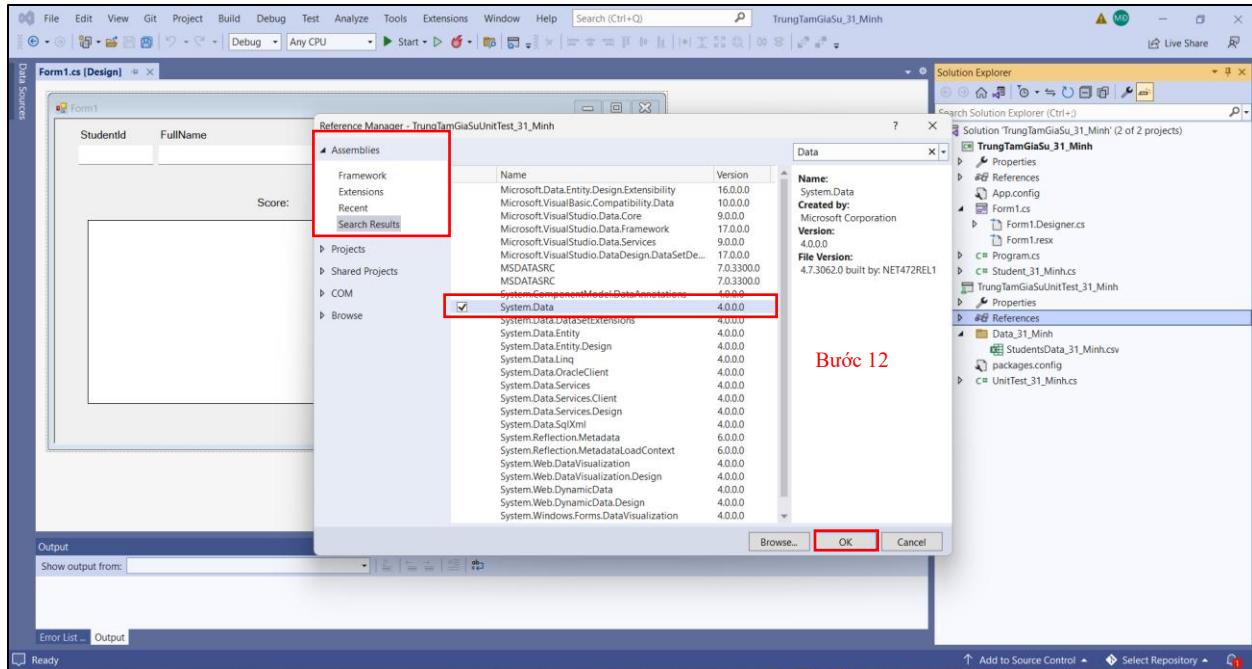
- Bước 11: Chọn Projects > Chọn Solution > Tick vào ô TrungTamGiaSu_31_Minh
- Bước 12: Chọn Assemblies > Chọn Search Results > Tick vào ô System.Data > OK



Bước 10



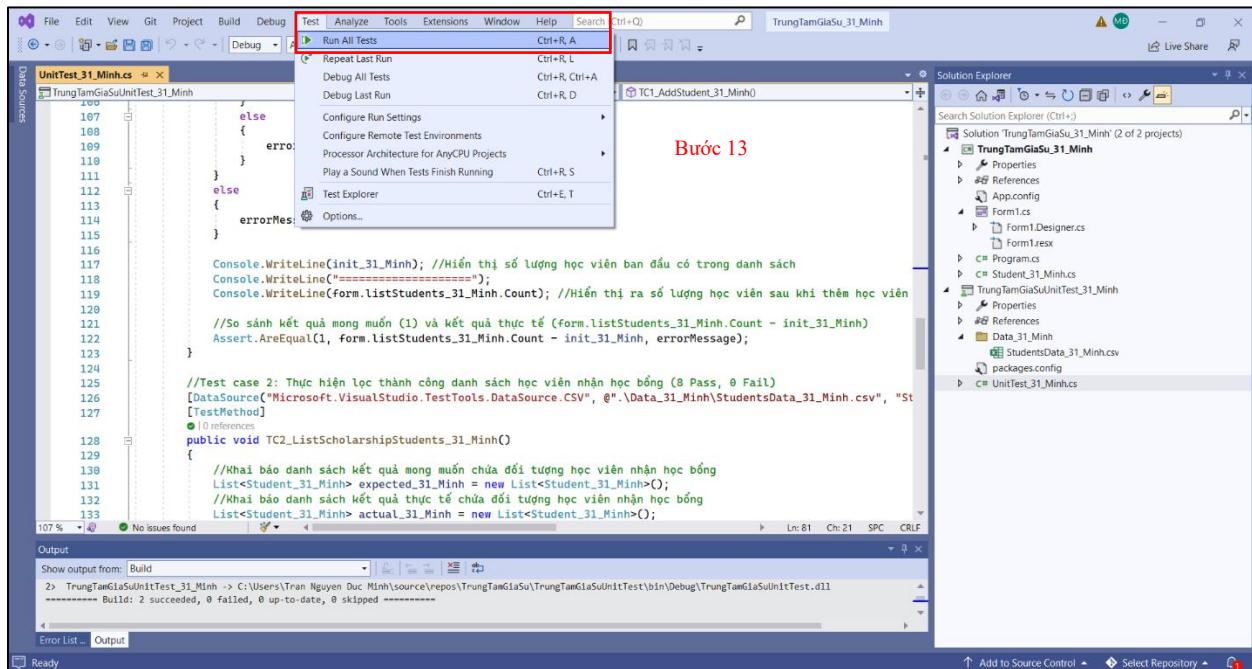
Bước 11

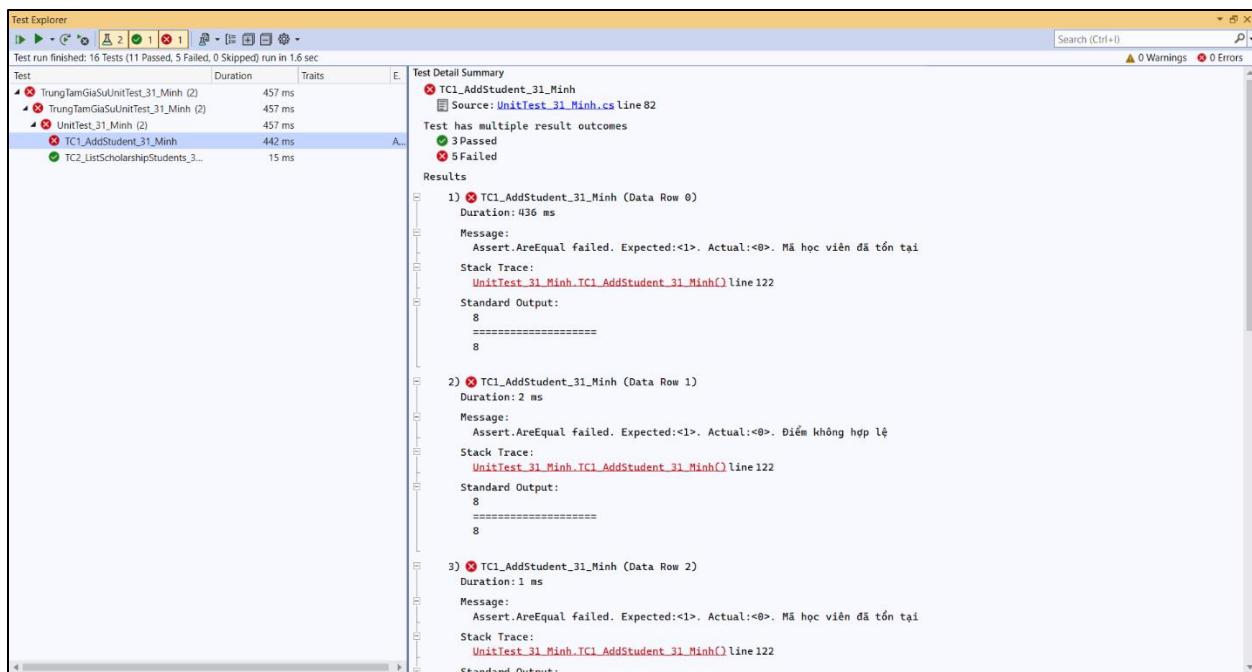


Hình 1.9. Các bước thực hiện thêm dự án (31_Minh)

d. Kết quả thực hiện:

- Bước 13: Chọn Test > Run All Tests





Hình 1.10. Kết quả kiểm thử chức năng của hệ thống (31_Minh)

CHƯƠNG 2. KIỂM THỬ SELENIUM VỚI TRANG WEBSITE PINTEREST

2.1. Tổng quan

2.1.1. *Giới thiệu:*

Pinterest là một nền tảng mạng xã hội và công cụ khám phá trực quan, cho phép người dùng tìm kiếm, lưu trữ và chia sẻ ý tưởng dưới dạng hình ảnh hoặc video. Được ra mắt vào năm 2010, Pinterest nhanh chóng trở thành một trong những trang web phổ biến nhất dành cho những ai yêu thích sáng tạo, thiết kế, thời trang, ẩm thực, du lịch và nhiều lĩnh vực khác.

Trên Pinterest, người dùng có thể tạo các boards để sắp xếp và lưu trữ nội dung yêu thích của mình dưới dạng pins. Các ghim này có thể là hình ảnh, video hoặc liên kết đến các trang web bên ngoài. Nhờ vào thuật toán gợi ý thông minh, Pinterest giúp người dùng khám phá những ý tưởng mới dựa trên sở thích và hành vi tìm kiếm của họ.

Pinterest không chỉ phục vụ cá nhân mà còn là một công cụ mạnh mẽ cho doanh nghiệp, giúp họ tiếp cận khách hàng tiềm năng thông qua các chiến lược tiếp thị hình ảnh. Với giao diện thân thiện và kho nội dung phong phú, Pinterest đã trở thành một nền tảng lý tưởng để tìm kiếm cảm hứng và hiện thực hóa ý tưởng.

2.1.2. *Đặc tả yêu cầu:*

a. *Chức năng đăng nhập:*

- **Tên chức năng:** Đăng nhập tài khoản.
- **Mô tả:** Chức năng đăng nhập cho phép người dùng truy cập vào tài khoản Pinterest bằng email để sử dụng các chức năng trong hệ thống.
- **Tiền điều kiện:** Người dùng đã có tài khoản.
- **Luồng hoạt động:**

- Bước 1: Người dùng truy cập vào trang website <https://www.pinterest.com/>

- Bước 2: Người dùng nhấn vào nút “Log in”
 - Bước 3: Người dùng cần nhập chính xác email và mật khẩu
 - Bước 4: Người dùng nhấn chọn nút submit “Log in”
 - Bước 5: Hệ thống hiển thị màn hình trang chủ khi người dùng đăng nhập thành công
- **Luồng thay thế:**
- Nếu người dùng nhập sai email hoặc mật khẩu thì sẽ hiển thị thông báo lỗi.
 - Nếu tài khoản của người dùng bị khóa hoặc bị vô hiệu hóa thì sẽ hiển thị thông báo và hướng dẫn cách khôi phục tài khoản.
 - Nếu người dùng nhập sai định dạng email hoặc để trống thì sẽ hiển thị thông báo lỗi.
 - Nếu người dùng quên mật khẩu thì hệ thống cung cấp tùy chọn “Forgot your password?” để đặt lại mật khẩu qua email.

b. Chức năng tải hình ảnh về:

- **Tên chức năng:** Tải hình ảnh xuống.
- **Mô tả:** Chức năng tải hình ảnh xuống cho phép người dùng lưu hình ảnh từ Pinterest về thiết bị cá nhân, giúp người dùng có thể xem và sử dụng hình ảnh ngoại tuyến.
- **Tiền điều kiện:** Người dùng đã được xác thực qua chức năng đăng nhập.
- **Luồng hoạt động:**
 - Bước 1: Người dùng thực hiện đăng nhập vào hệ thống
 - Bước 2: Người dùng nhấn vào một hình ảnh bất kỳ hoặc tìm kiếm hình ảnh mà mình muốn tải về
 - Bước 3: Hệ thống hiển thị giao diện xem chi tiết hình ảnh
 - Bước 4: Người dùng chọn vào biểu tượng nút ba chấm
 - Bước 5: Người dùng chọn vào nút “Tải hình ảnh xuống”
 - Bước 6: Hệ thống tải ảnh về thiết bị của người dùng theo định dạng mặc định

- **Luồng thay thế:**

- Nếu người dùng muốn tải hình ảnh xuống nhưng chưa đăng nhập thì hiển thị thông báo yêu cầu người dùng phải đăng nhập.
- Một số hình ảnh có thể bị tắt tính năng tải xuống do quyền riêng tư của người đăng thì hệ thống hiển thị thông báo và không cho phép tải.
- Mạng Internet bị gián đoạn trong khi tải hình ảnh thì hệ thống hiển thị thông báo lỗi và yêu cầu thử lại khi có kết nối Internet.
- Không đủ dung lượng lưu trữ trên thiết bị của người dùng thì hệ thống hiển thị cảnh báo và hủy quá trình tải.

c. Chức năng chia sẻ hình ảnh (Copy link):

- **Tên chức năng:** Chia sẻ hình ảnh (Copy link).
- **Mô tả:** Chức năng Copy link cho phép người dùng sao chép đường dẫn (URL) của hình ảnh trên Pinterest để chia sẻ với người khác qua tin nhắn, email hoặc mạng xã hội.
- **Tiền điều kiện:** Không có.
- **Luồng hoạt động:**

- Bước 1: Người dùng nhấn vào một hình ảnh bất kỳ hoặc tìm kiếm hình ảnh mà mình muốn chia sẻ
- Bước 2: Hệ thống hiển thị giao diện xem chi tiết hình ảnh
- Bước 3: Người dùng nhấn vào biểu tượng “Chia sẻ”
- Bước 4: Chọn tùy chọn “Sao chép liên kết” (Copy link)
- Bước 5: Hệ thống tự động sao chép đường dẫn của hình ảnh vào clipboard
- Bước 6: Người dùng có thể dán liên kết vào nơi mong muốn

- **Luồng thay thế:**

- Nếu hình ảnh bị đặt ở chế độ riêng tư hoặc người đăng không cho phép chia sẻ thì hệ thống hiển thị thông báo.
- Lỗi hệ thống hoặc trình duyệt không cho phép sao chép link thì hiển thị thông báo lỗi và hướng dẫn người dùng cấp quyền sao chép.

2.1.3. Thiết kế test case:

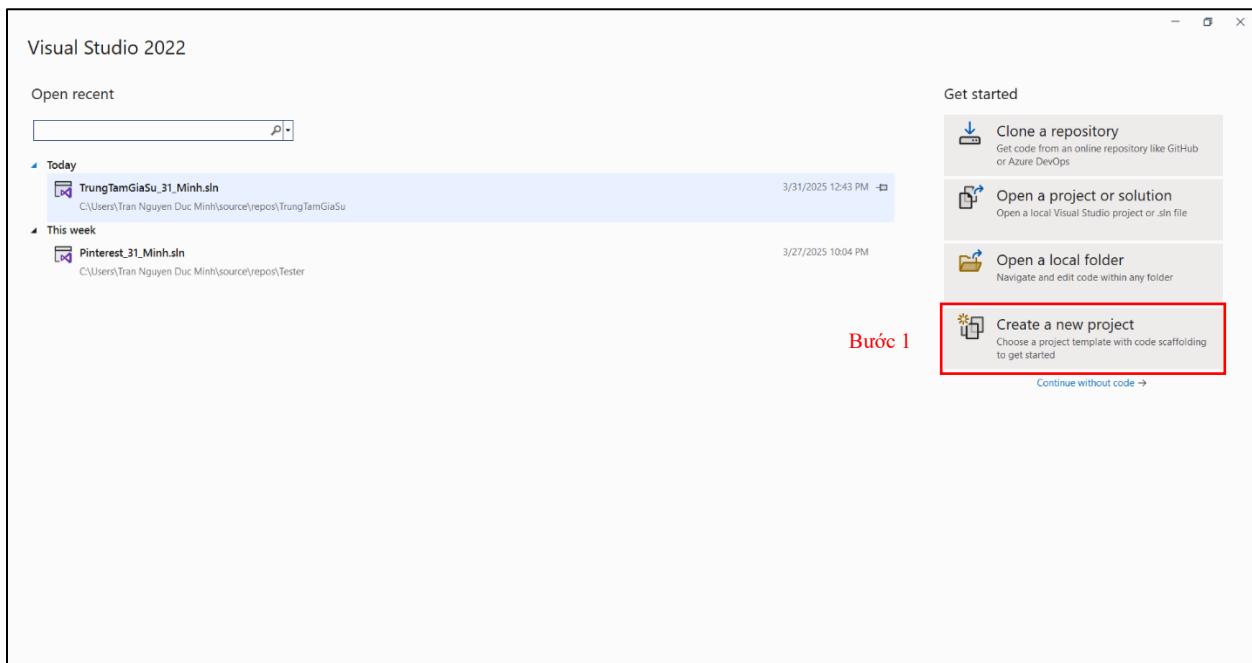
STT_31_Minh	TestCaseID_31_Minh	Input_31_Minh			ChucNang_31_Minh	KetQuaMongDoi_31_Minh	KetQuaThucTe_31_Minh	KetQua_31_Minh
		Email_31_Minh	Password_31_Minh	KeySearch_31_Minh				
1	TC1_LoginAccountSuccess_31_Minh	ducmnh2004@gmail.com	minhpro123		Login	Đăng nhập thành công	Đăng nhập thành công	Pass
2	TC2_LoginAccountWrongPassword_31_Minh	ducmnh2004@gmail.com	minhpro321		Login	Bảo lối sai mật khẩu	Bảo lối sai mật khẩu	Pass
3	TC3_LoginAccountNonexistentEmail_31_Minh	minhzzzzzz@gmail.com			Login	Bảo lối email không tồn tại	Bảo lối email không tồn tại	Pass
4	TC4_LoginAccountInvalidEmail_31_Minh	ducmnh2004@gmail.com	minhpro123		Login	Bảo lối email không hợp lệ	Bảo lối email không hợp lệ	Pass
5	TC5_LoginAccountEmptyFields_31_Minh				Login	Bảo lối không nhập thông tin tài khoản	Bảo lối không nhập thông tin tài khoản	Pass
6	TC6_LoginAndSearchImageAndDownload_31_Minh	ducmnh2004@gmail.com	minhpro123	Cat	Download Image	Tải hình ảnh xuống thành công	Tải hình ảnh xuống thành công	Pass
7	TC7_SearchImageAndDownload_31_Minh			Cat	Download Image	Yêu cầu người dùng đăng nhập	Yêu cầu người dùng đăng nhập	Pass
8	TC8_LoginAndShareLinkImage_31_Minh	ducmnh2004@gmail.com	minhpro123	Cat	Share Link	Sao chép đường dẫn chia sẻ ảnh thành công	Sao chép đường dẫn chia sẻ ảnh thành công	Pass
9	TC9_ShareLinkImage_31_Minh			Cat	Share Link	Sao chép đường dẫn chia sẻ ảnh thành công	Sao chép đường dẫn chia sẻ ảnh thành công	Pass

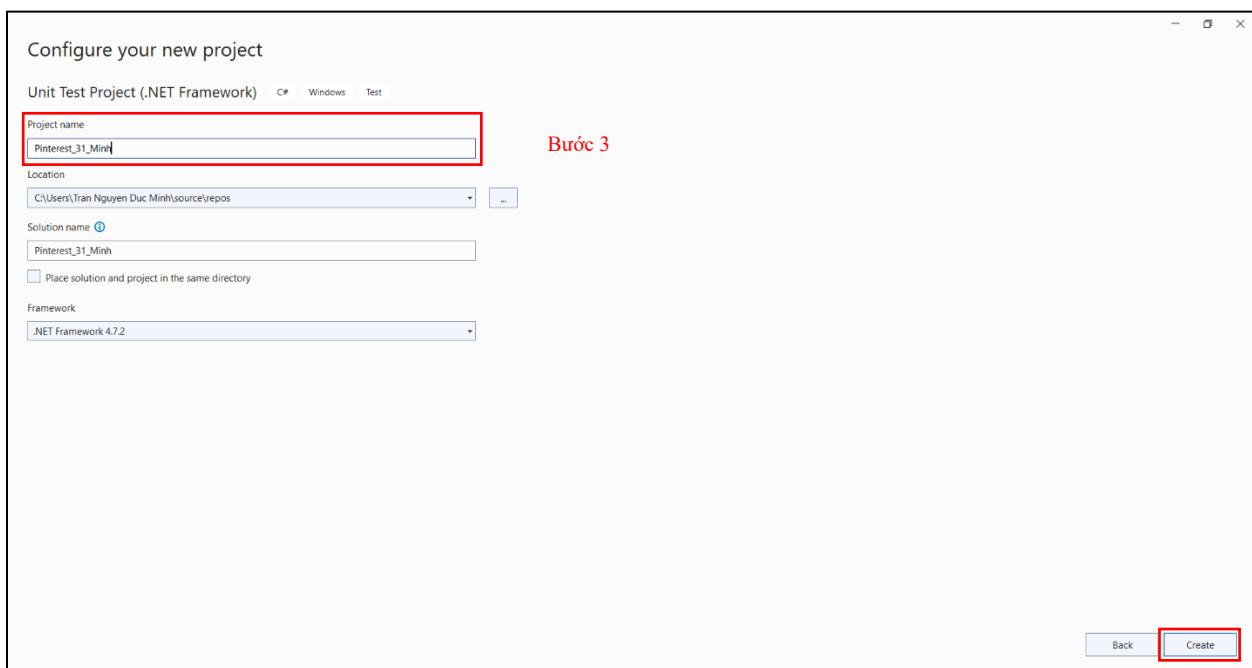
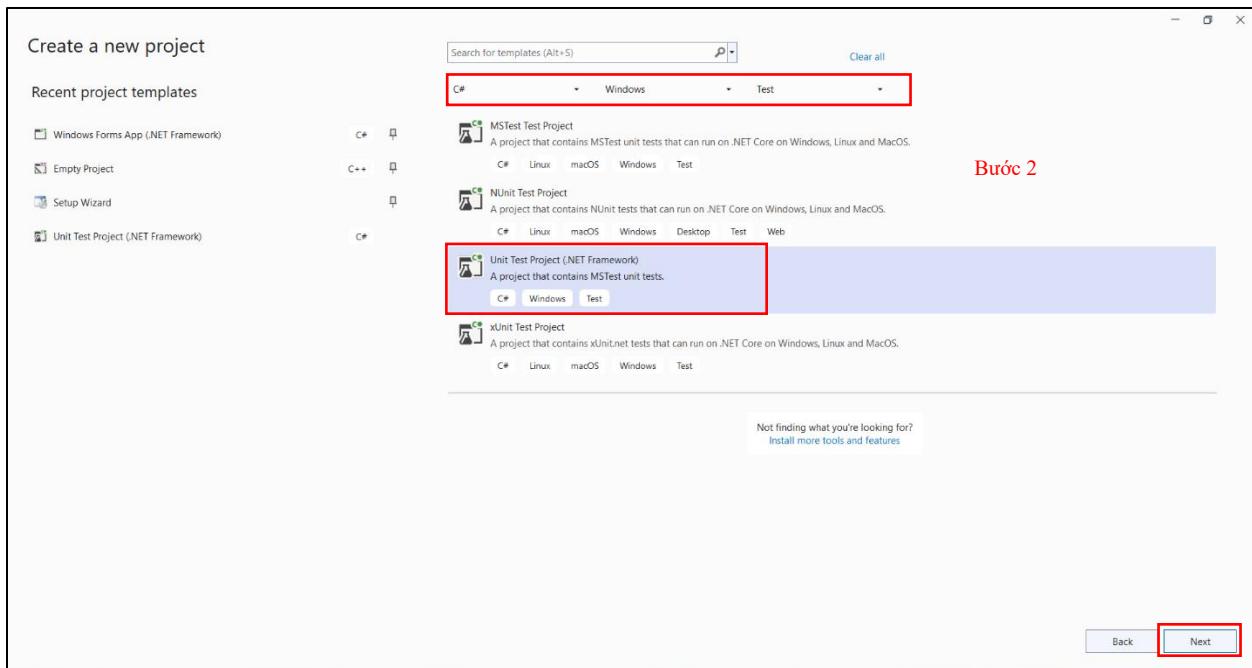
Hình 2.1. Các test case của các chức năng kiểm thử website Pinterest (31_Minh)

2.2. Thiết lập dự án

2.2.1. Tạo một project Unit Test Project (.NET Framework):

- Bước 1: Mở **Visual Studio 2022** > Chọn **Create a new project**
- Bước 2: Tìm kiếm project với các thông tin **C#, Windows, Test** > Chọn **Unit Test Project (.NET Framework)** > Next
- Bước 3: Đặt tên project là **Pinterest_31_Minh** > Create

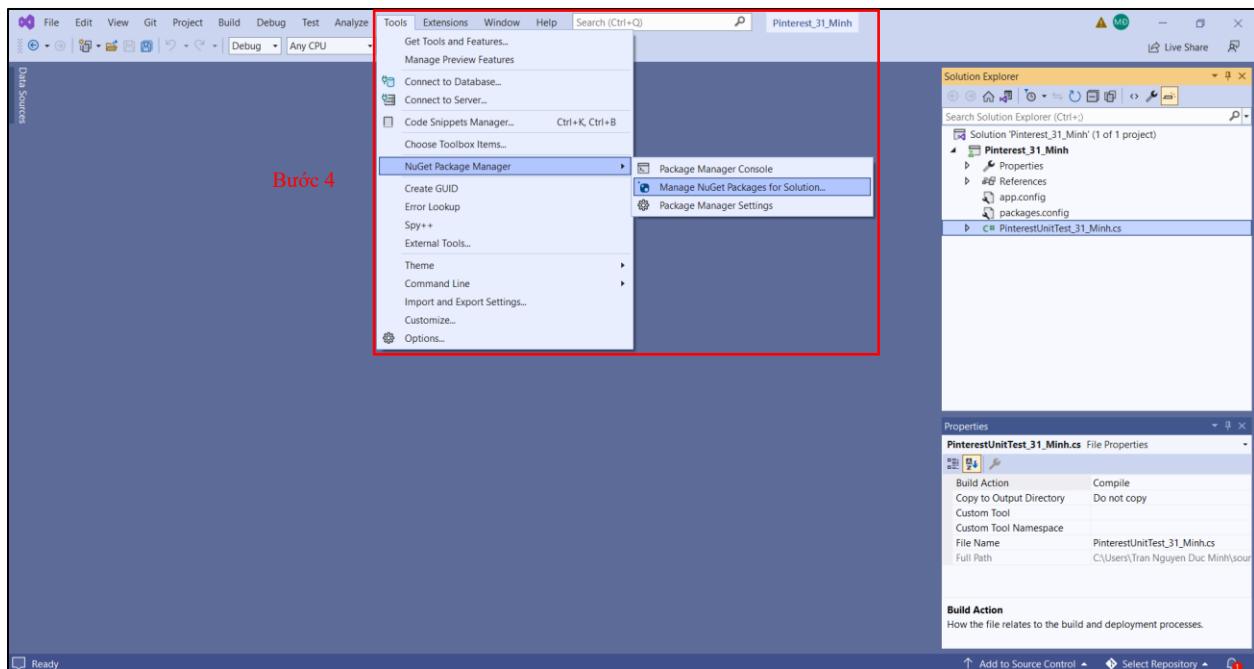


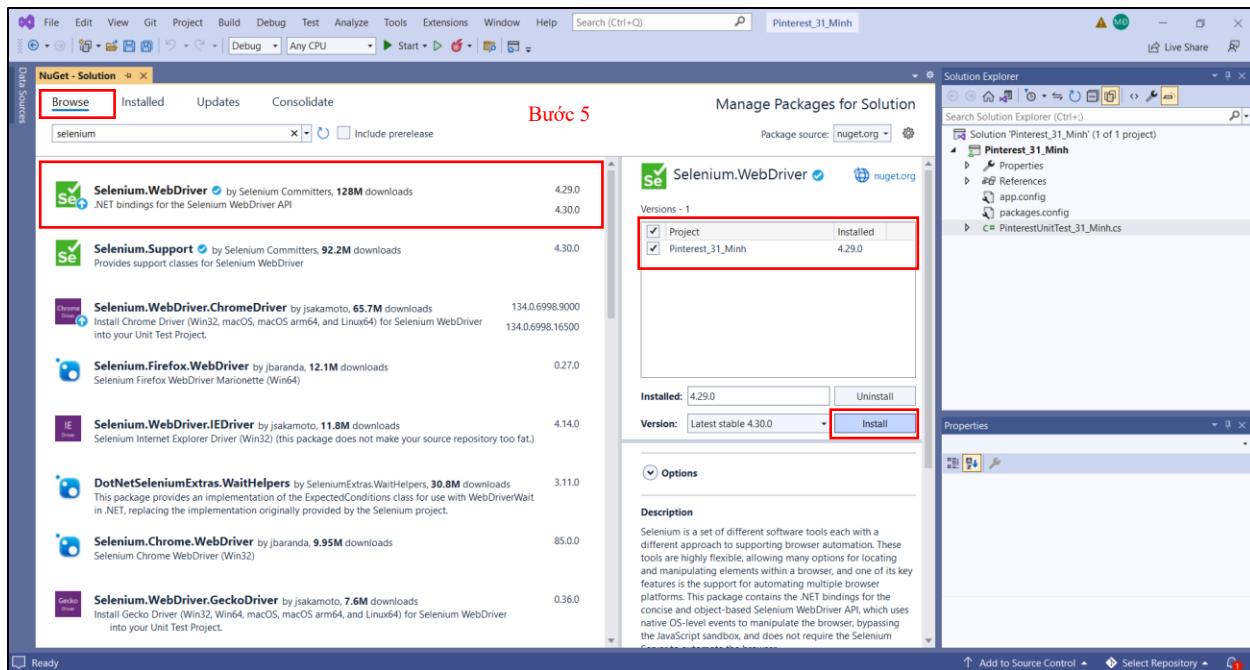


Hình 2.2. Các bước tạo một project Unit Test (31_Minh)

2.2.2. Thêm thư viện Selenium vào project:

- Bước 4: Chọn Tools > Chọn NuGet Package Manager > Chọn Manage NuGet Packages for Solution...
- Bước 5: Chọn Browse > Tìm kiếm **Selenium.WebDriver** > Install
- Bước 6: Tìm kiếm **Selenium.WebDriver.ChromeDriver** > Install





Hình 2.3. Các bước thêm thư viện Selenium vào project (31_Minh)

2.3. Kiểm thử chức năng

2.3.1. Chuẩn bị website trong trình duyệt:

- Thực hiện viết code như sau và run để mở website trong trình duyệt kiểm thử:

```

using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using System.Threading;
using OpenQA.Selenium.Interactions;
using System.IO;

namespace PinterestUnitTest_31_Minh
{
    [TestClass]
    public class PinterestUnitTest_31_Minh
    {
        private IWebDriver driver_31_Minh;

        //Địa chỉ trang web
        private string url_31_Minh = "https://www.pinterest.com/";

        [TestInitialize]
        public void SetUp_31_Minh()
        {
            //Khởi tạo đối tượng ChromeDriver
            driver_31_Minh = new ChromeDriver();

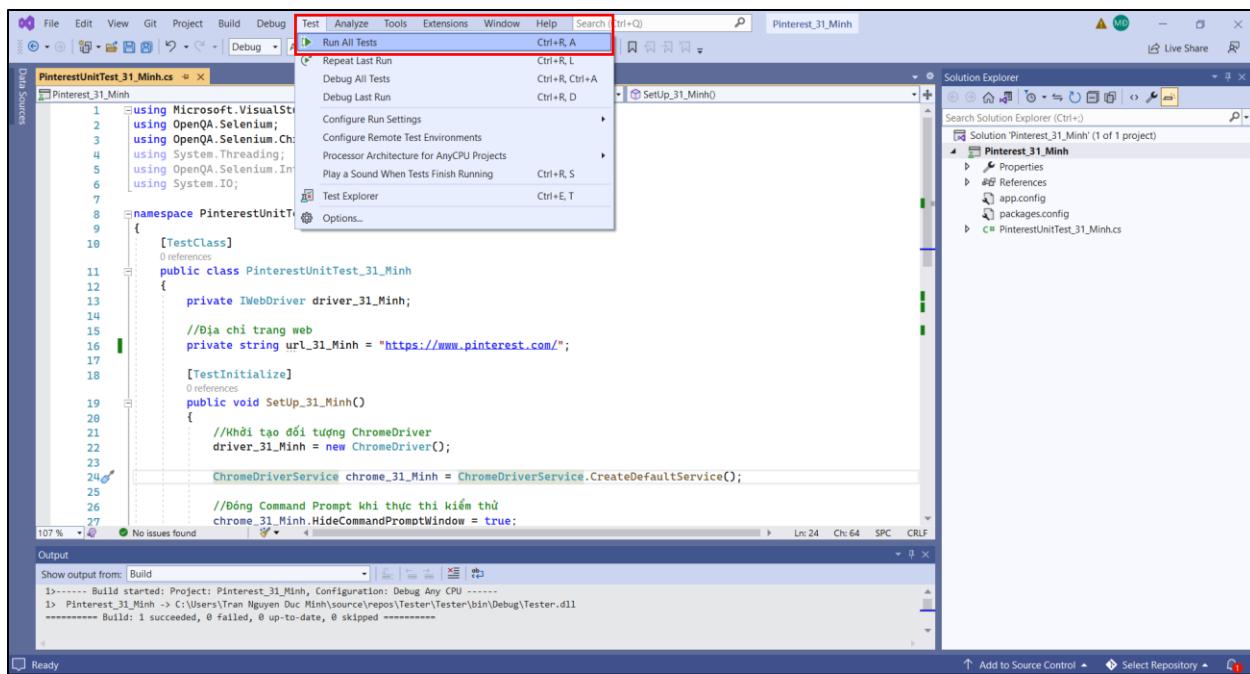
            ChromeDriverService chrome_31_Minh =
            ChromeDriverService.CreateDefaultService();

            //Đóng Command Prompt khi thực thi kiểm thử
            chrome_31_Minh.HideCommandPromptWindow = true;

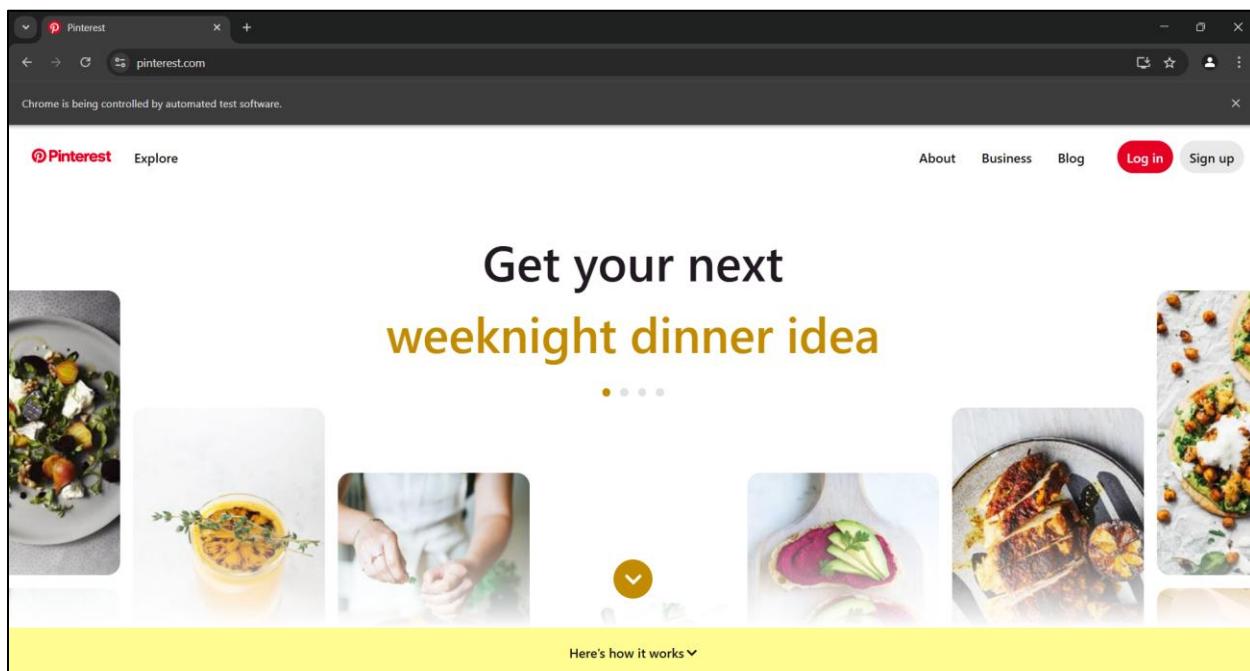
            //Mở trình duyệt ở chế độ toàn màn hình
            driver_31_Minh.Manage().Window.Maximize();
        }

        [TestMethod]
        public void ChromeDriver_31_Minh()
        {
            //Điều hướng trình duyệt tới trang web
            driver_31_Minh.Navigate().GoToUrl(url_31_Minh);
        }
    }
}

```



Hình 2.4. Mở trình duyệt kiểm thử bằng cách run test (31_Minh)

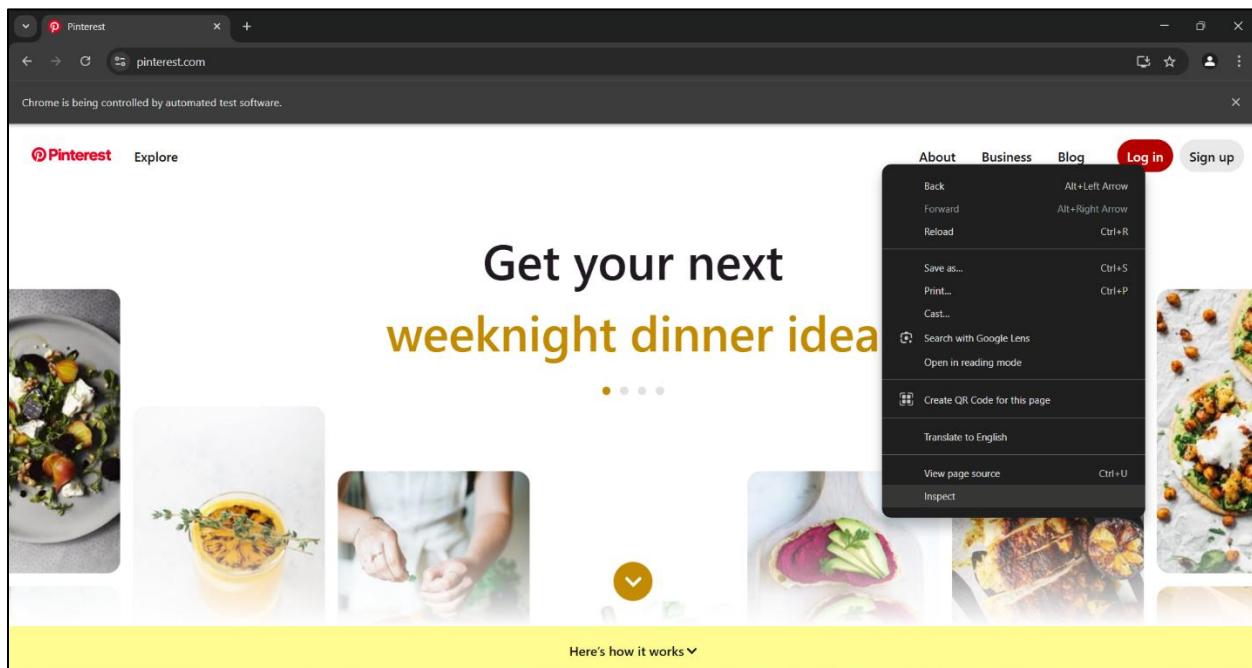


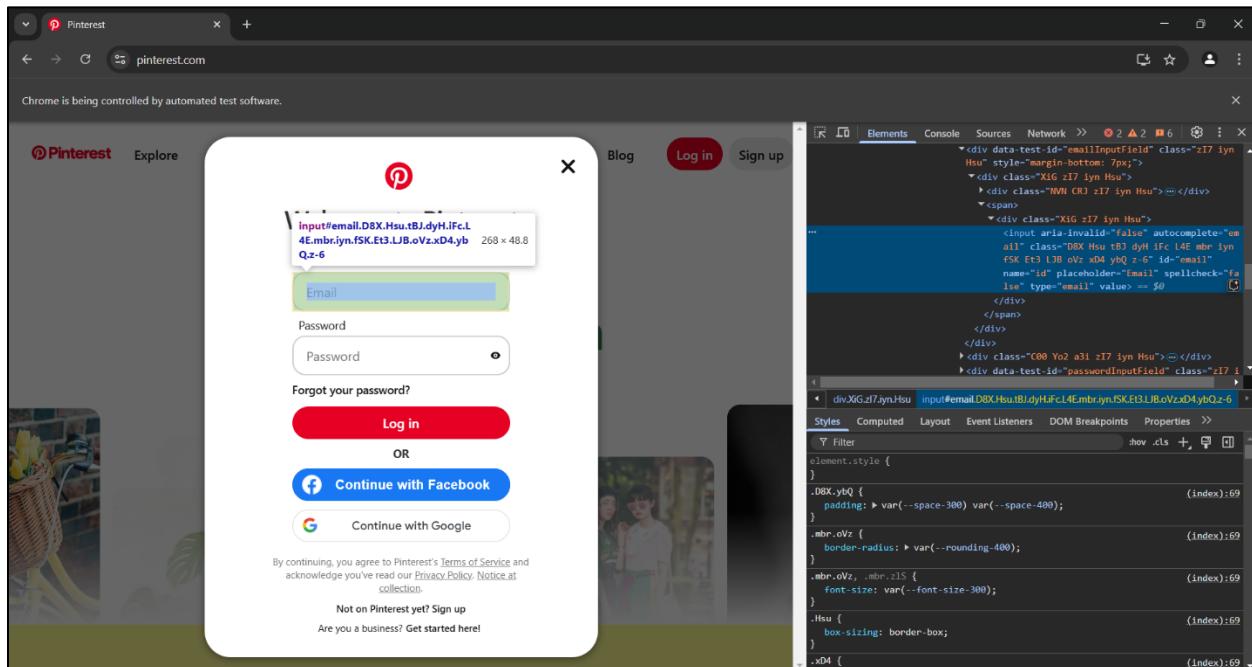
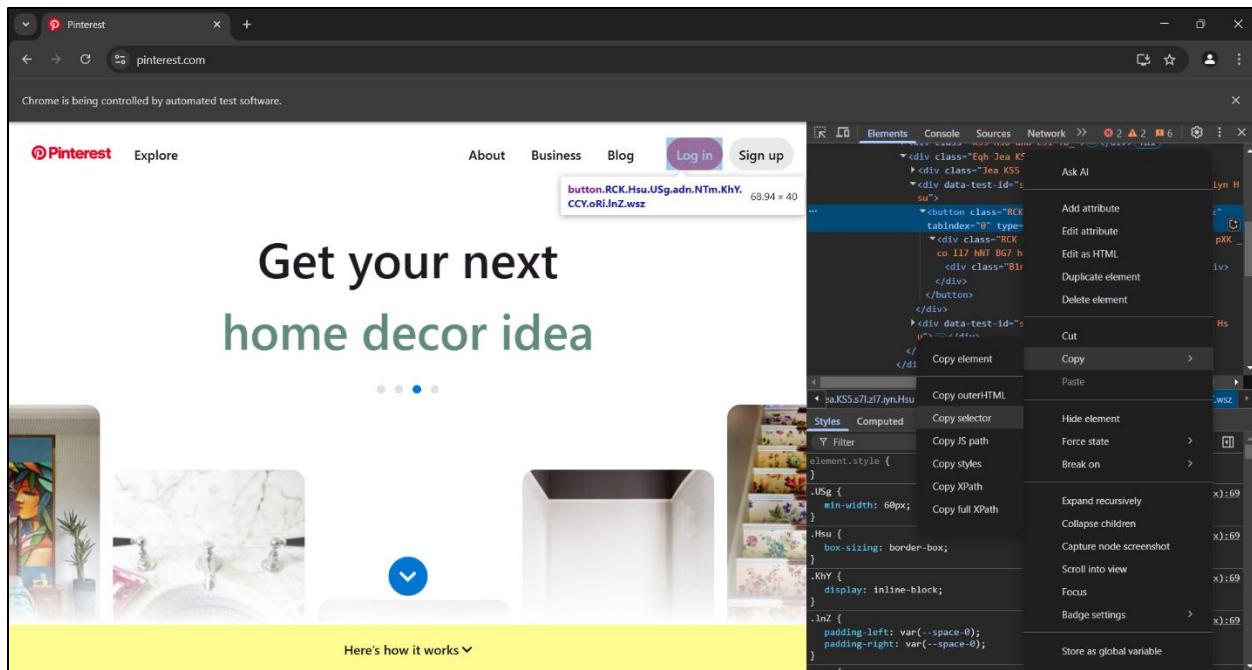
Hình 2.5. Giao diện của website trong trình duyệt kiểm thử (31_Minh)

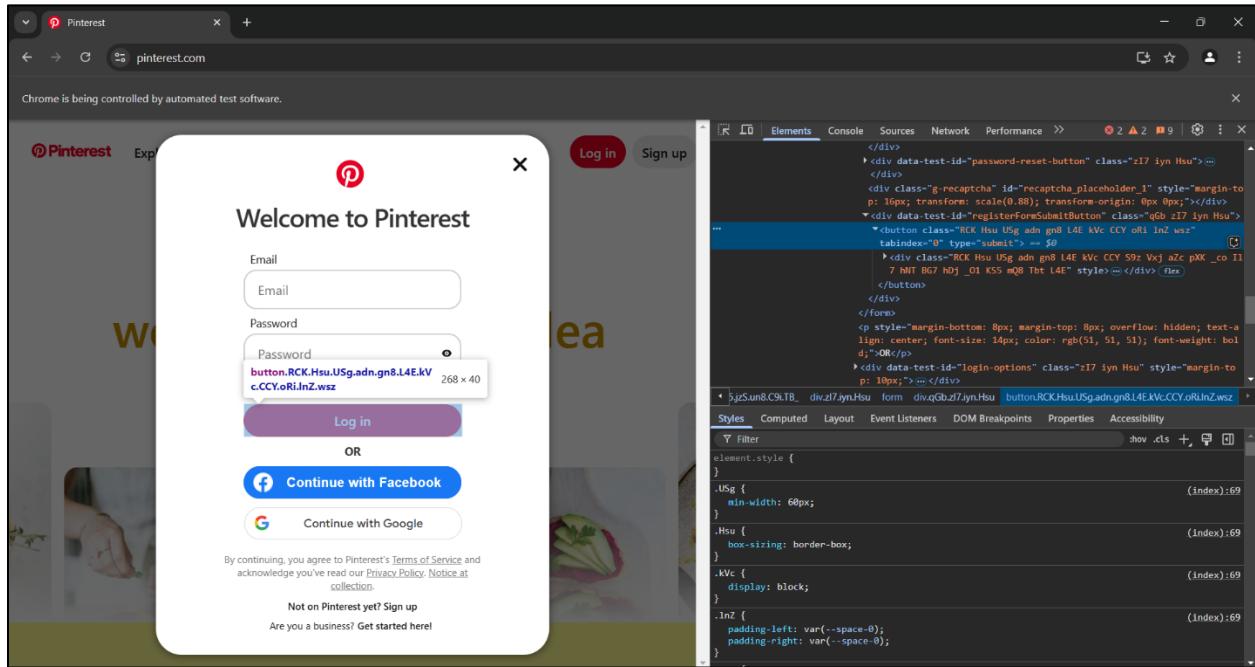
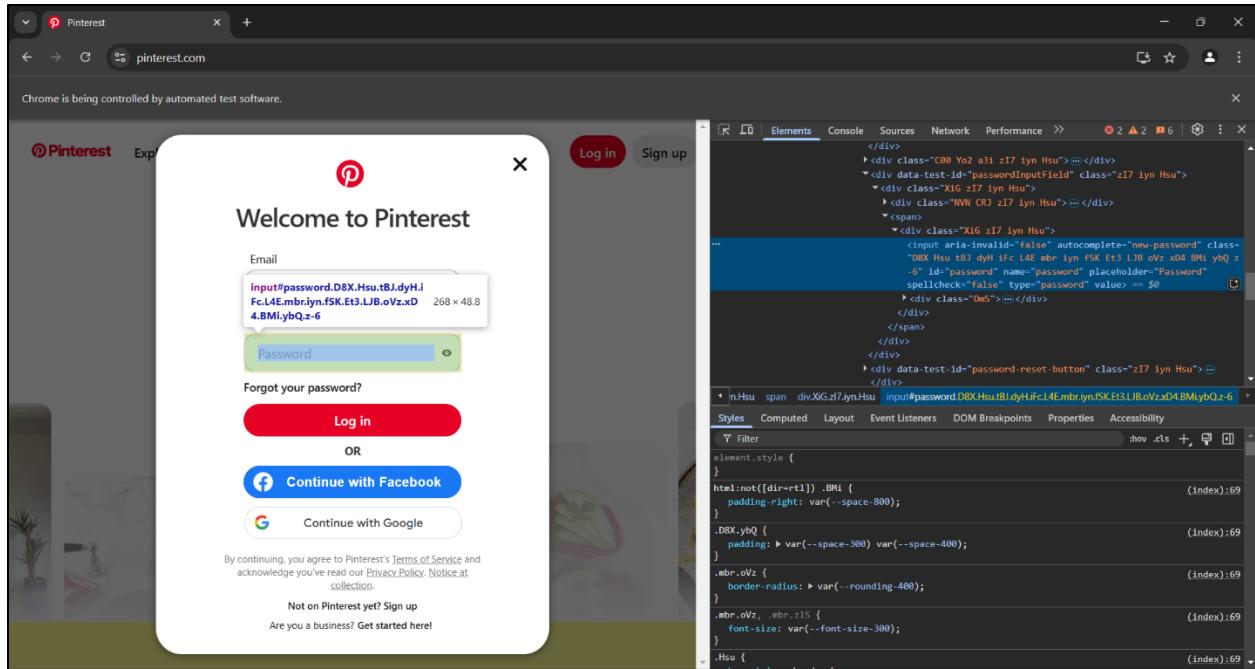
2.3.2. *Chức năng đăng nhập:*

a. *Đăng nhập thành công:*

- Bước 1: Thực hiện lấy thuộc tính của các nút điều hướng liên quan đến chức năng:
 - Lấy thuộc tính nút “Log in”: Nhấn chuột phải vào nút “Log in” > Inspect (Kiểm tra) > Tìm đến thẻ chứa nút > Nhấn chuột phải > Copy > Copy selector
 - Lấy thuộc tính ô nhập “Email”: Nhấn chuột phải vào ô nhập “Email” > Inspect (Kiểm tra) > Tìm đến thẻ chứa ô nhập > Lấy thuộc tính id=“email”
 - Lấy thuộc tính ô nhập “Password”: Nhấn chuột phải vào ô nhập “Password” > Inspect (Kiểm tra) > Tìm đến thẻ chứa ô nhập > Lấy thuộc tính id=“password”
 - Lấy thuộc tính nút submit “Log in”: Nhấn chuột phải vào nút submit “Log in” > Inspect (Kiểm tra) > Tìm đến thẻ chứa nút > Nhấn chuột phải > Copy > Copy selector



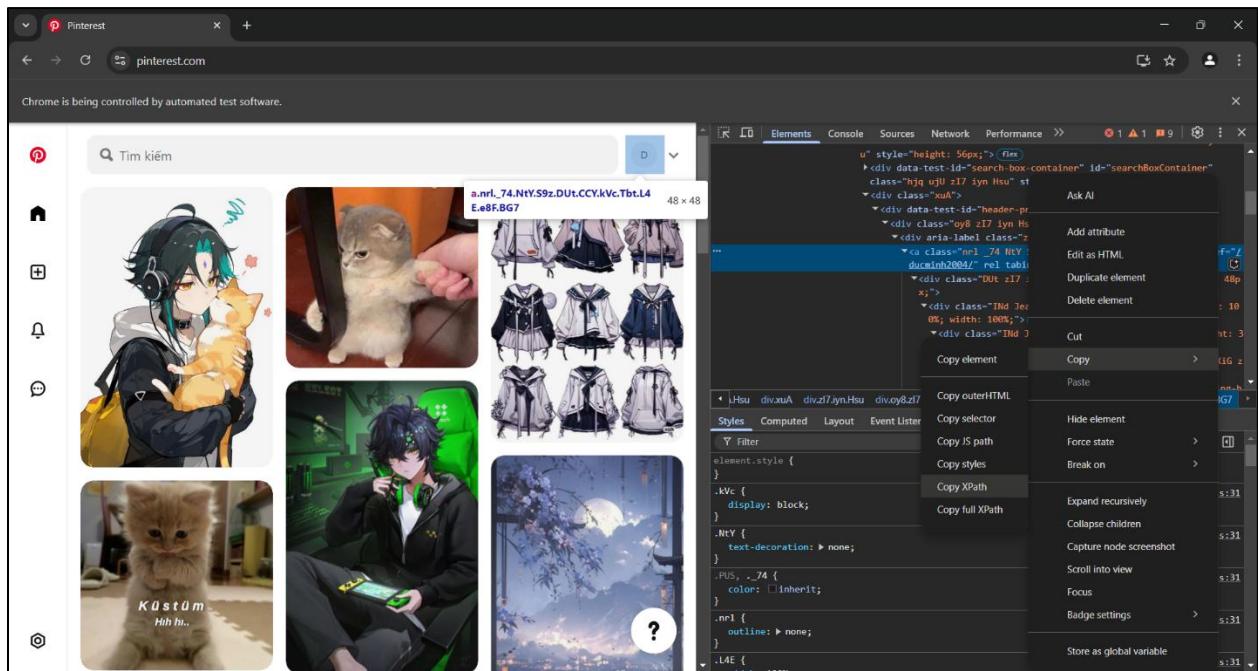


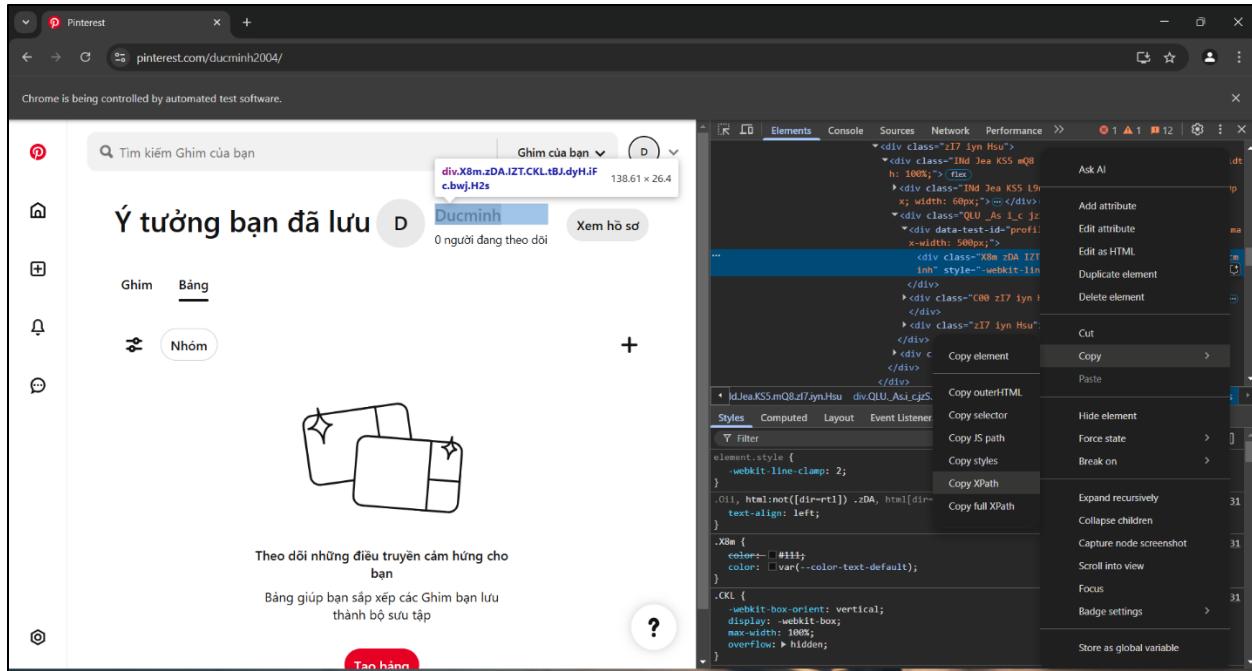


Hình 2.6. Các bước lấy các thuộc tính liên quan đến chức năng đăng nhập (31_Minh)

- Bước 2: Đăng nhập thành công với tài khoản cần kiểm thử:

- Lấy thuộc tính nút “Hồ sơ của bạn”: Nhấn chuột phải vào nút “Hồ sơ của bạn” > Inspect (Kiểm tra) > Tìm đến thẻ chứa nút > Nhấn chuột phải > Copy > Copy XPath
- Lấy thuộc tính tên người dùng: Bôi tên người dùng > Nhấn chuột phải > Inspect (Kiểm tra) > Tìm đến thẻ chức nội dung > Nhấn chuột phải > Copy > Copy XPath





Hình 2.7. Các bước lấy thuộc tính của tên người dùng (31_Minh)

- Bước 3: Viết code và chạy test case:

```

using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using System.Threading;
using OpenQA.Selenium.Interactions;
using System.IO;

namespace PinterestUnitTest_31_Minh
{
    [TestClass]
    public class PinterestUnitTest_31_Minh
    {
        private IWebDriver driver_31_Minh;

        //Địa chỉ trang web
        private string url_31_Minh = "https://www.pinterest.com/";

        //Tên tài khoản
        private string email_31_Minh = "ducminh2004@gmail.com";

        //Mật khẩu
        private string password_31_Minh = "minhpro123";

        //Tên người dùng
        private string username_31_Minh = "Ducminh";

        [TestInitialize]
        public void SetUp_31_Minh()
    }
}

```

```

{
    //Khởi tạo đối tượng ChromeDriver
    driver_31_Minh = new ChromeDriver();

    ChromeDriverService chrome_31_Minh =
    ChromeDriverService.CreateDefaultService();

    //Đóng Command Prompt khi thực thi kiểm thử
    chrome_31_Minh.HideCommandPromptWindow = true;

    //Mở trình duyệt ở chế độ toàn màn hình
    driver_31_Minh.Manage().Window.Maximize();
}

//Test case 1: Đăng nhập với thông tin hợp lệ
//email_31_Minh = 'ducminh2004@gmail.com'
//password_31_Minh = 'minhpro123'
[TestMethod]
public void TC1_LoginAccountSuccess_31_Minh()
{
    //Điều hướng trình duyệt tới trang web
    driver_31_Minh.Navigate().GoToUrl(url_31_Minh);
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút "Log in" bằng cách sử dụng FindElement CSS
    Selector
    driver_31_Minh.FindElement(By.CssSelector("#__PWS_ROOT__ > div > div >
    div:nth-child(1) > div:nth-child(2) > div.QLY._he.zI7.iyn.Hsu > div >
    div.Eqh.Jea.KS5.s7I.zI7.iyn.Hsu > div.H-G.zI7.iyn.Hsu > button")).Click();
    Thread.Sleep(10000);

    //Điền tên đăng nhập vào ô "Email" bằng cách sử dụng FindElement Id
    driver_31_Minh.FindElement(By.Id("email")).SendKeys(email_31_Minh);

    //Điền mật khẩu vào ô "Password" bằng cách sử dụng FindElement Id
    driver_31_Minh.FindElement(By.Id("password")).SendKeys(password_31_Minh);
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút submit "Log in" bằng cách sử dụng
    FindElement CSS Selector

    driver_31_Minh.FindElement(By.CssSelector("button[type='submit']")).Click();
    Thread.Sleep(10000);

    //Tìm kiếm và click vào "Hồ sơ của bạn" bằng cách sử dụng FindElement
    XPath
    driver_31_Minh.FindElement(By.XPath("//*[@id=__HeaderContent__]/div/div/div[2]/div/d
    iv/div/div[2]/div/div/div/a")).Click();
    Thread.Sleep(10000);

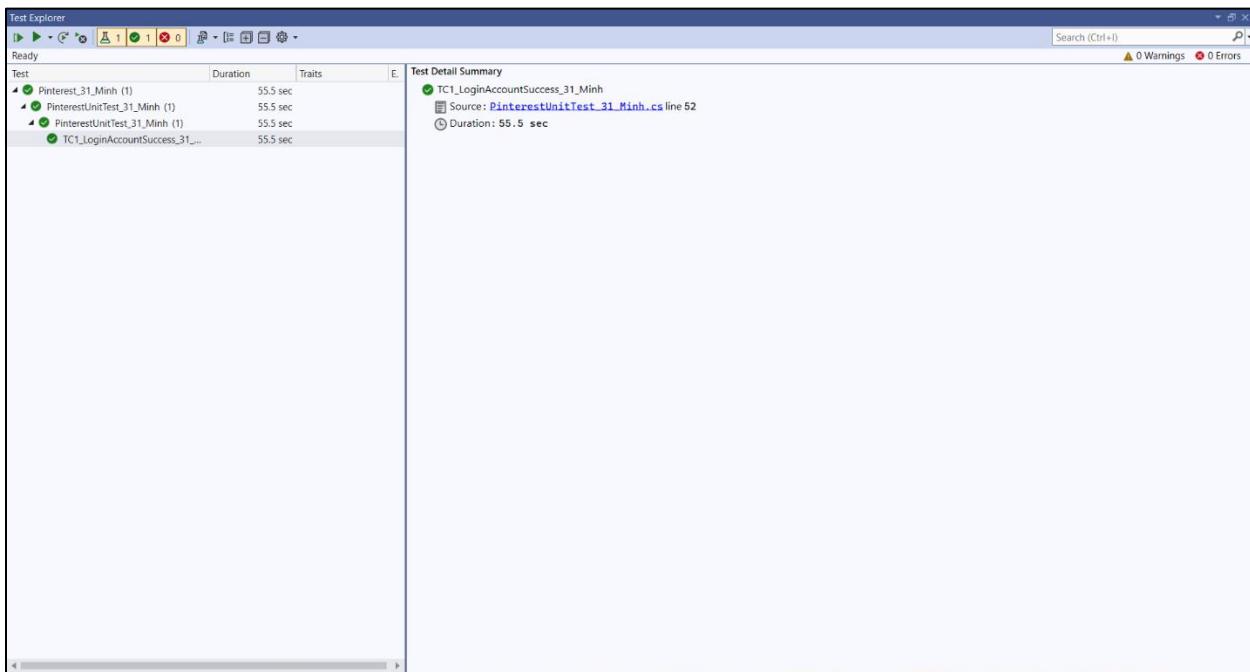
    //Tìm kiếm và lấy tên người dùng hiện tại bằng cách sử dụng FindElement
    XPath
    IWebElement currentUser_31_Minh =
    driver_31_Minh.FindElement(By.XPath("//*[@id=__PWS_ROOT__]/div/div/div[1]/div/di
    v[2]/div/div/div/div[1]/div/div[1]/div[2]/div/div[1]/div"));
}

```

```
//Chuyển kiểu dữ liệu của currentUser_31_Minhh thành string
string actualUserName_31_Minhh = currentUser_31_Minhh.Text;

//So sánh kết quả mong muốn (username_31_Minhh) và kết quả thực tế
(actualUserName_31_Minhh)
Assert.AreEqual(username_31_Minhh, actualUserName_31_Minhh, "Đăng nhập
thất bại");
}

[TestCleanup]
public void Quit()
{
    //Đóng tất cả cửa sổ liên kết
    driver_31_Minhh.Quit();
}
```

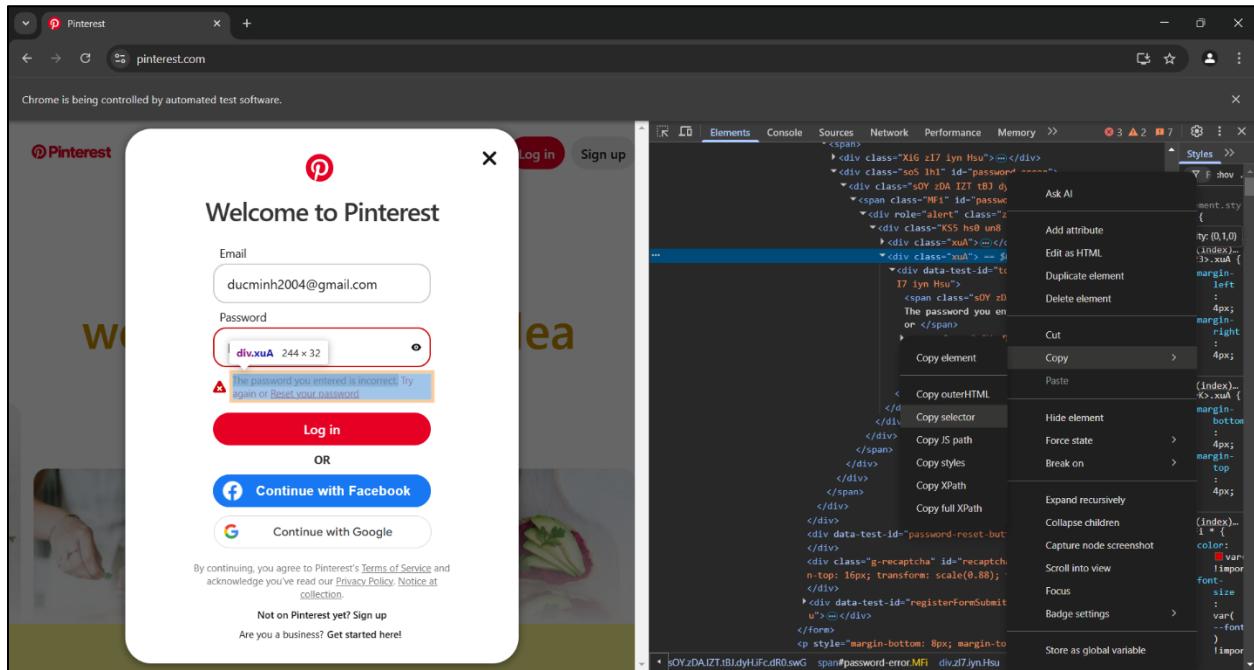


Hình 2.8. Kết quả sau khi chạy TC1_LoginAccountSuccess_31_Minh (31_Minh)

b. Đăng nhập với mật khẩu sai:

- Bước 1: Thực hiện lấy các thuộc tính liên quan đến chức năng đăng nhập giống như trên
 - Bước 2: Thử đăng nhập với mật khẩu sai:

- Lấy thuộc tính thông báo lỗi (màu đỏ): Bôi dòng thông báo lỗi > Nhấn chuột phải > Inspect (Kiểm tra) > Tìm đến thẻ chức nội dung > Nhấn chuột phải > Copy > Copy selector



Hình 2.9. Lấy thuộc tính của nội dung thông báo lỗi khi đăng nhập với mật khẩu sai
(31_Minh)

- Bước 3: Viết code và chạy test case:

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using System.Threading;
using OpenQA.Selenium.Interactions;
using System.IO;

namespace PinterestUnitTest_31_Minh
{
    [TestClass]
    public class PinterestUnitTest_31_Minh
    {
        private IWebDriver driver_31_Minh;

        //Địa chỉ trang web
        private string url_31_Minh = "https://www.pinterest.com/";

        //Tên tài khoản
        private string email_31_Minh = "ducminh2004@gmail.com";
    }
}
```

```

//Mật khẩu
private string password_31_Minh = "minhpro321";

[TestInitialize]
public void SetUp_31_Minh()
{
    //Khởi tạo đối tượng ChromeDriver
    driver_31_Minh = new ChromeDriver();

    ChromeDriverService chrome_31_Minh =
    ChromeDriverService.CreateDefaultService();

    //Đóng Command Prompt khi thực thi kiểm thử
    chrome_31_Minh.HideCommandPromptWindow = true;

    //Mở trình duyệt ở chế độ toàn màn hình
    driver_31_Minh.Manage().Window.Maximize();
}

//Test case 2: Đăng nhập với mật khẩu sai
//email_31_Minh = 'ducminh2004@gmail.com'
//password_31_Minh = 'minhpro321'
[TestMethod]
public void TC2_LoginAccountWrongPassword_31_Minh()
{
    //Điều hướng trình duyệt tới trang web
    driver_31_Minh.Navigate().GoToUrl(url_31_Minh);
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút "Log in" bằng cách sử dụng FindElement CSS
    Selector
    driver_31_Minh.FindElement(By.CssSelector("#__PWS_ROOT__ > div > div >
    div:nth-child(1) > div:nth-child(2) > div.QLY._he.zI7.iyn.Hsu > div >
    div.Eqh.Jea.KS5.s7I.zI7.iyn.Hsu > div.H-G.zI7.iyn.Hsu > button")).Click();
    Thread.Sleep(10000);

    //Điền tên đăng nhập vào ô "Email" bằng cách sử dụng FindElement Id
    driver_31_Minh.FindElement(By.Id("email")).SendKeys(email_31_Minh);

    //Điền mật khẩu vào ô "Password" bằng cách sử dụng FindElement Id
    driver_31_Minh.FindElement(By.Id("password")).SendKeys(password_31_Minh);
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút submit "Log in" bằng cách sử dụng
    FindElement CSS Selector
    driver_31_Minh.FindElement(By.CssSelector("button[type='submit']")).Click();
    Thread.Sleep(10000);

    //Tìm kiếm và lấy thông báo mật khẩu không đúng bằng cách sử dụng
    FindElement CSS Selector
    IWebElement error_31_Minh =
    driver_31_Minh.FindElement(By.CssSelector("#password-error > div > div > div:nth-
    child(2)"));

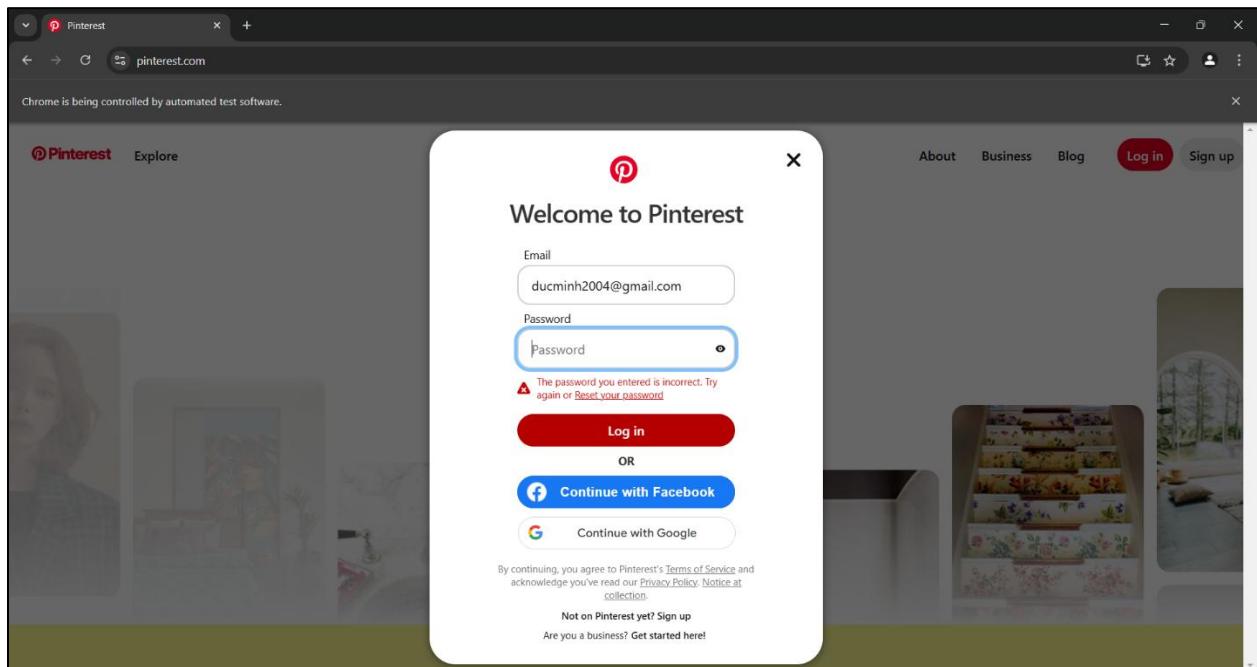
    //Chuyển kiểu dữ liệu của error_31_Minh thành string
}

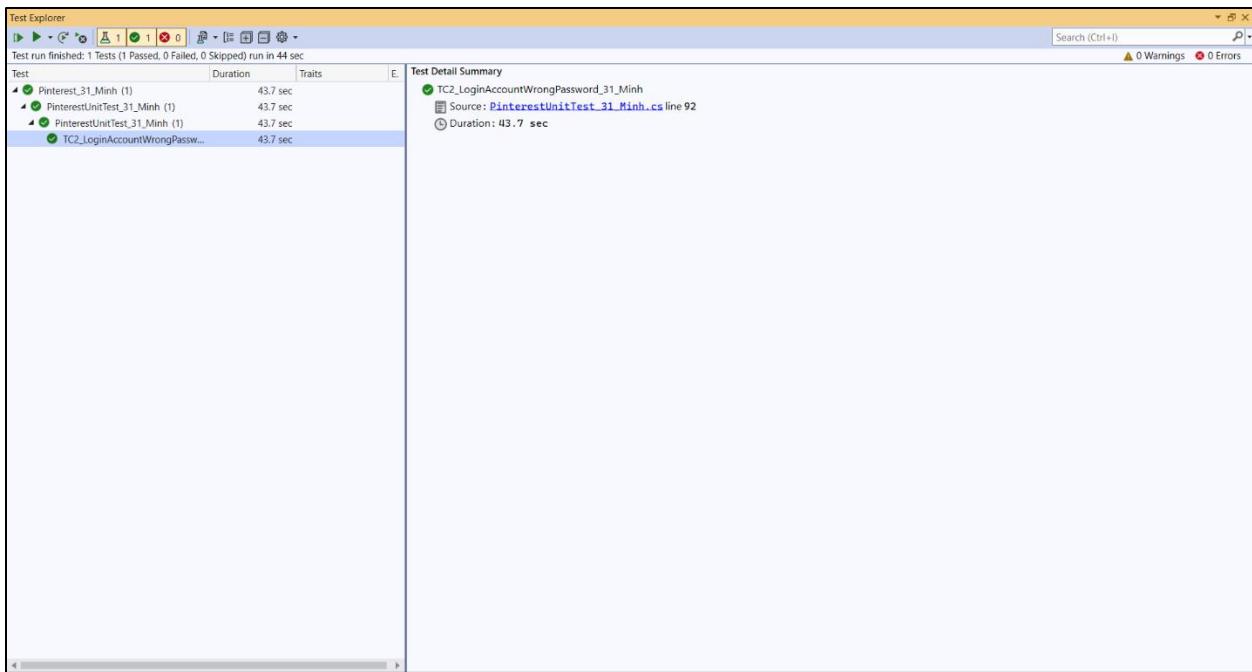
```

```
string errorMessage_31_Minh = error_31_Minh.Text;

    //So sánh kết quả thực tế (errorMessage_31_Minh) giống kết quả mong muốn
    ("The password you entered is incorrect.")
    Assert.IsTrue(errorMessage_31_Minh.Contains("The password you entered is
incorrect."));
}

[TestCleanup]
public void Quit()
{
    //Đóng tất cả cửa sổ liên kết
    driver_31_Minh.Quit();
}
}
```

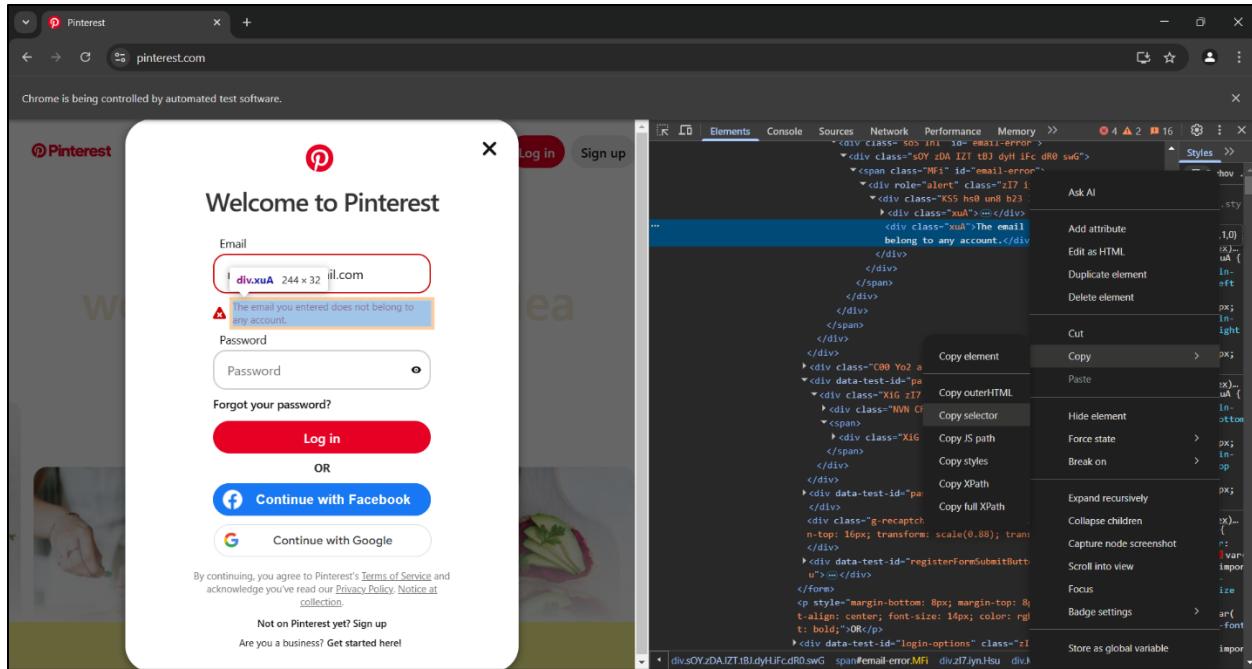




Hình 2.10. Kết quả sau khi chạy TC2_LoginAccountWrongPassword_31_Minh (31_Minh)

c. Đăng nhập với email không tồn tại:

- Bước 1: Thực hiện lấy các thuộc tính liên quan đến chức năng đăng nhập giống như trên
- Bước 2: Thủ đăng nhập với email không tồn tại:
 - Lấy thuộc tính thông báo lỗi (màu đỏ): Bôi dòng thông báo lỗi > Nhấn chuột phải > Inspect (Kiểm tra) > Tìm đến thẻ chức nội dung > Nhấn chuột phải > Copy > Copy selector



Hình 2.11. Lấy thuộc tính của nội dung thông báo lỗi khi đăng nhập với email không tồn tại (31_Minh)

- Bước 3: Viết code và chạy test case:

```

using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using System.Threading;
using OpenQA.Selenium.Interactions;
using System.IO;

namespace PinterestUnitTest_31_Minh
{
    [TestClass]
    public class PinterestUnitTest_31_Minh
    {
        private IWebDriver driver_31_Minh;

        //Địa chỉ trang web
        private string url_31_Minh = "https://www.pinterest.com/";

        //Tên tài khoản
        private string email_31_Minh = "minhzzzzzz@gmail.com";

        //Mật khẩu
        private string password_31_Minh = "";

        [TestInitialize]
        public void SetUp_31_Minh()
        {
    
```

```

//Khởi tạo đối tượng ChromeDriver
driver_31_Minh = new ChromeDriver();

ChromeDriverService chrome_31_Minh =
ChromeDriverService.CreateDefaultService();

//Đóng Command Prompt khi thực thi kiểm thử
chrome_31_Minh.HideCommandPromptWindow = true;

//Mở trình duyệt ở chế độ toàn màn hình
driver_31_Minh.Manage().Window.Maximize();
}

//Test case 3: Đăng nhập với Email không tồn tại
//email_31_Minh = 'minhzzzzz@gmail.com'
//password_31_Minh = ''
[TestMethod]
public void TC3_LoginAccountNonExistentEmail_31_Minh()
{
    //Điều hướng trình duyệt tới trang web
    driver_31_Minh.Navigate().GoToUrl(url_31_Minh);
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút "Log in" bằng cách sử dụng FindElement CSS
    Selector
    driver_31_Minh.FindElement(By.CssSelector("#__PWS_ROOT__ > div > div >
div:nth-child(1) > div:nth-child(2) > div.QLY._he.zI7.iyn.Hsu > div >
div.Eqh.Jea.KS5.s7I.zI7.iyn.Hsu > div.H-G.zI7.iyn.Hsu > button")).Click();
    Thread.Sleep(10000);

    //Điền tên đăng nhập vào ô "Email" bằng cách sử dụng FindElement Id
    driver_31_Minh.FindElement(By.Id("email")).SendKeys(email_31_Minh);

    //Điền mật khẩu vào ô "Password" bằng cách sử dụng FindElement Id
    driver_31_Minh.FindElement(By.Id("password")).SendKeys(password_31_Minh);
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút submit "Log in" bằng cách sử dụng
    FindElement CSS Selector
    driver_31_Minh.FindElement(By.CssSelector("button[type='submit']")).Click();
    Thread.Sleep(10000);

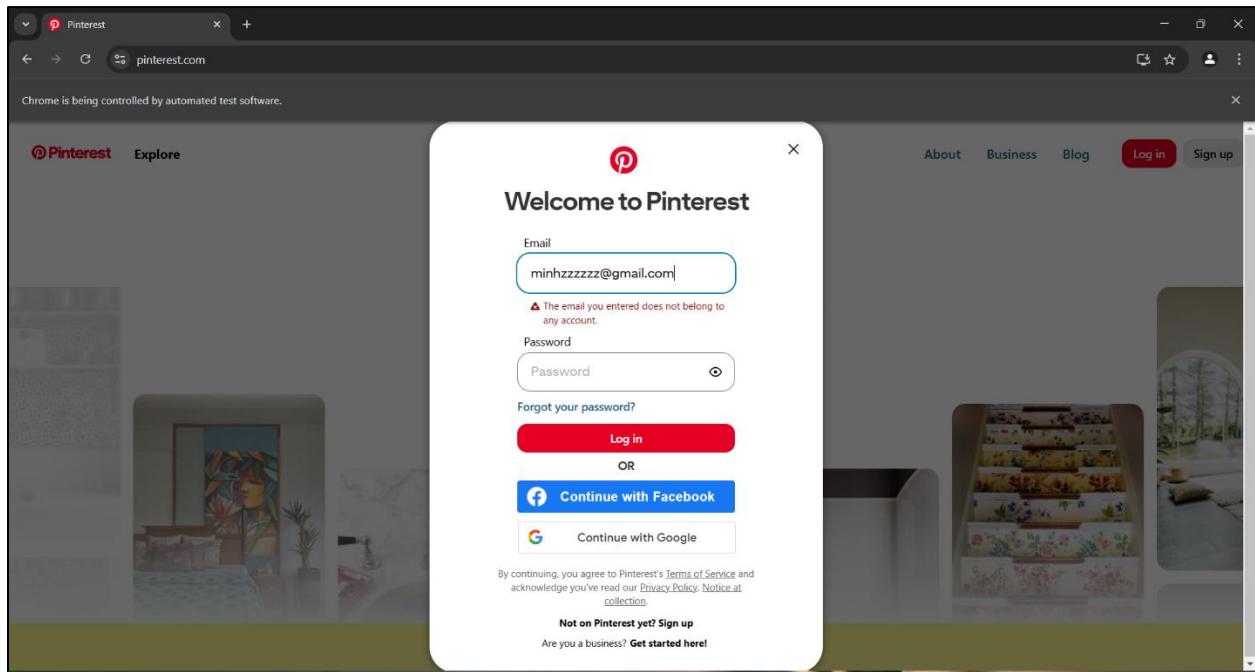
    //Tìm kiếm và lấy thông báo Email không tồn tại bằng cách sử dụng
    FindElement CSS Selector
    IWebElement error_31_Minh =
    driver_31_Minh.FindElement(By.CssSelector("#email-error > div > div > div:nth-
    child(2)"));

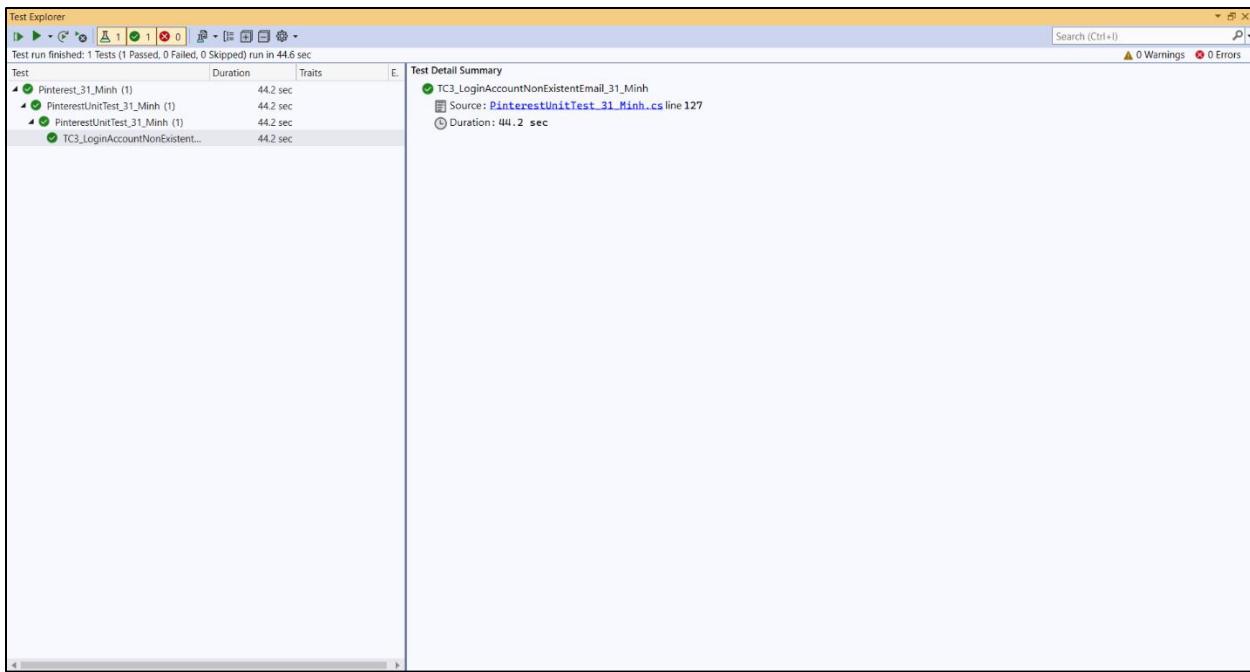
    //Chuyển kiểu dữ liệu của error_31_Minh thành string
    string errorMessage_31_Minh = error_31_Minh.Text;

    //So sánh kết quả thực tế (errorMessage_31_Minh) giống kết quả mong muốn
    ("The email you entered does not belong to any account.")
    Assert.IsTrue(errorMessage_31_Minh.Contains("The email you entered does
    not belong to any account."));
}

```

```
[TestCleanup]
public void Quit()
{
    //Đóng tất cả cửa sổ liên kết
    driver_31_Minh.Quit();
}
```

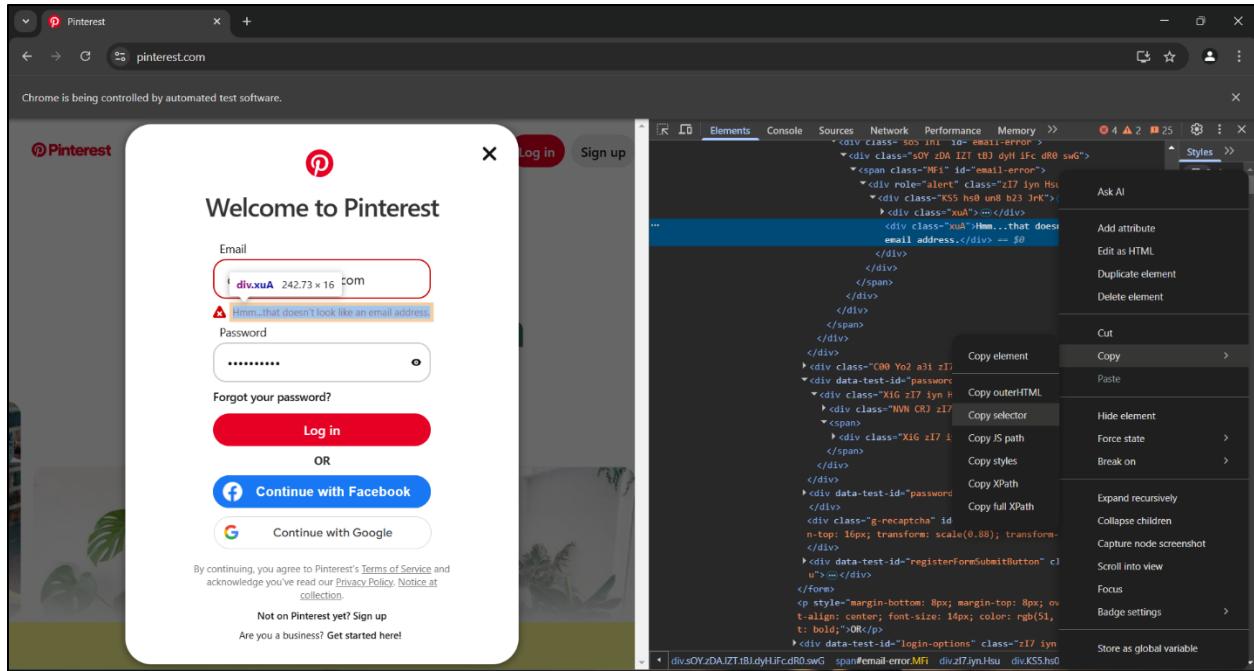




Hình 2.12. Kết quả sau khi chạy TC3_LoginAccountNonExistentEmail_31_Minh (31_Minh)

d. Đăng nhập với email không hợp lệ:

- Bước 1: Thực hiện lấy các thuộc tính liên quan đến chức năng đăng nhập giống như trên
- Bước 2: Thủ đăng nhập với email không hợp lệ:
 - Lấy thuộc tính thông báo lỗi (màu đỏ): Bôi dòng thông báo lỗi > Nhấn chuột phải > Inspect (Kiểm tra) > Tìm đến thẻ chức nội dung > Nhấn chuột phải > Copy > Copy selector



Hình 2.13. Lấy thuộc tính của nội dung thông báo lỗi khi đăng nhập với email không hợp lệ (31_Minh)

- Bước 3: Viết code và chạy test case:

```

using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using System.Threading;
using OpenQA.Selenium.Interactions;
using System.IO;

namespace PinterestUnitTest_31_Minh
{
    [TestClass]
    public class PinterestUnitTest_31_Minh
    {
        private IWebDriver driver_31_Minh;

        //Địa chỉ trang web
        private string url_31_Minh = "https://www.pinterest.com/";

        //Tên tài khoản
        private string email_31_Minh = "ducminh2004@gmail.com";

        //Mật khẩu
        private string password_31_Minh = "minhpro123";

        [TestInitialize]
        public void SetUp_31_Minh()
        {
    
```

```

//Khởi tạo đối tượng ChromeDriver
driver_31_Minh = new ChromeDriver();

ChromeDriverService chrome_31_Minh =
ChromeDriverService.CreateDefaultService();

//Đóng Command Prompt khi thực thi kiểm thử
chrome_31_Minh.HideCommandPromptWindow = true;

//Mở trình duyệt ở chế độ toàn màn hình
driver_31_Minh.Manage().Window.Maximize();
}

//Test case 4: Đăng nhập với Email không hợp lệ
//email_31_Minh = 'ducminh2004@gmail.com'
//password_31_Minh = 'minhpro123'
[TestMethod]
public void TC4_LoginAccountInvalidEmail_31_Minh()
{
    //Điều hướng trình duyệt tới trang web
    driver_31_Minh.Navigate().GoToUrl(url_31_Minh);
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút "Log in" bằng cách sử dụng FindElement CSS
    Selector
    driver_31_Minh.FindElement(By.CssSelector("#__PWS_ROOT__ > div > div >
div:nth-child(1) > div:nth-child(2) > div.QLY._he.zI7.iyn.Hsu > div >
div.Eqh.Jea.KS5.s7I.zI7.iyn.Hsu > div.H-G.zI7.iyn.Hsu > button")).Click();
    Thread.Sleep(10000);

    //Điền tên đăng nhập vào ô "Email" bằng cách sử dụng FindElement Id
    driver_31_Minh.FindElement(By.Id("email")).SendKeys(email_31_Minh);

    //Điền mật khẩu vào ô "Password" bằng cách sử dụng FindElement Id
    driver_31_Minh.FindElement(By.Id("password")).SendKeys(password_31_Minh);
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút submit "Log in" bằng cách sử dụng
    FindElement CSS Selector
    driver_31_Minh.FindElement(By.CssSelector("button[type='submit']")).Click();
    Thread.Sleep(10000);

    //Tìm kiếm và lấy thông báo Email không hợp lệ bằng cách sử dụng
    FindElement CSS Selector
    IWebElement error_31_Minh =
    driver_31_Minh.FindElement(By.CssSelector("#email-error > div > div > div:nth-
    child(2)"));

    //Chuyển kiểu dữ liệu của error_31_Minh thành string
    string errorMessage_31_Minh = error_31_Minh.Text;

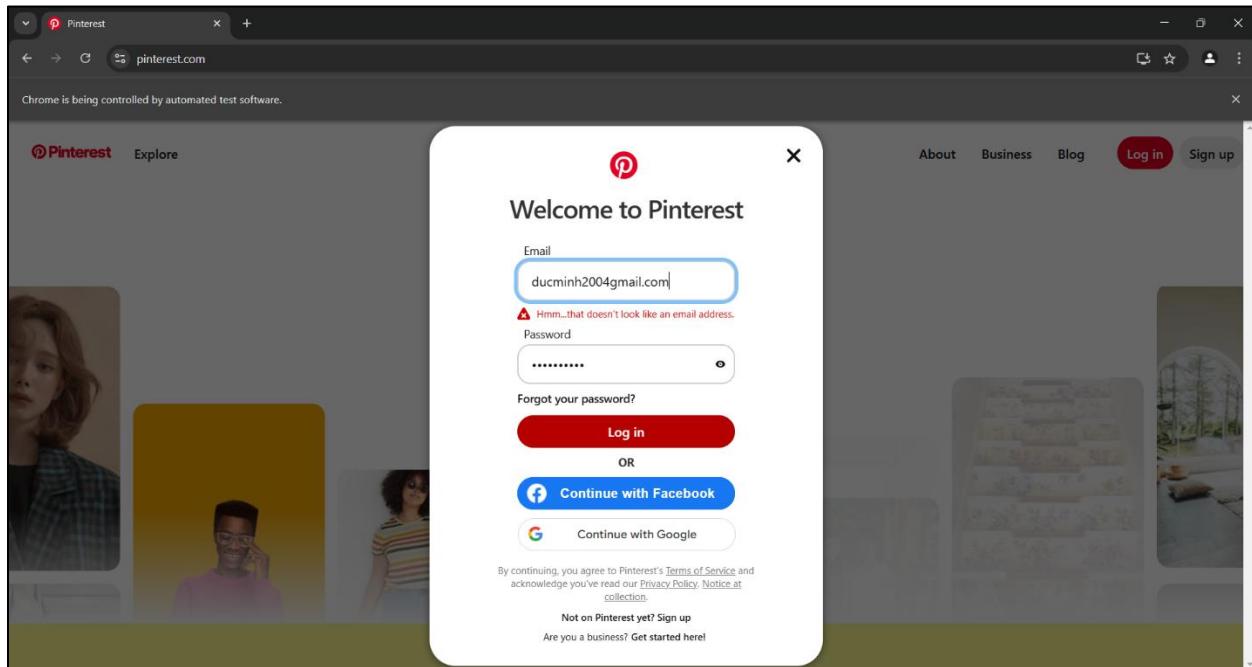
    //So sánh kết quả thực tế (errorMessage_31_Minh) giống kết quả mong muốn
    ("Hmm...that doesn't look like an email address.")
    Assert.IsTrue(errorMessage_31_Minh.Contains("Hmm...that doesn't look
    like an email address."));
}

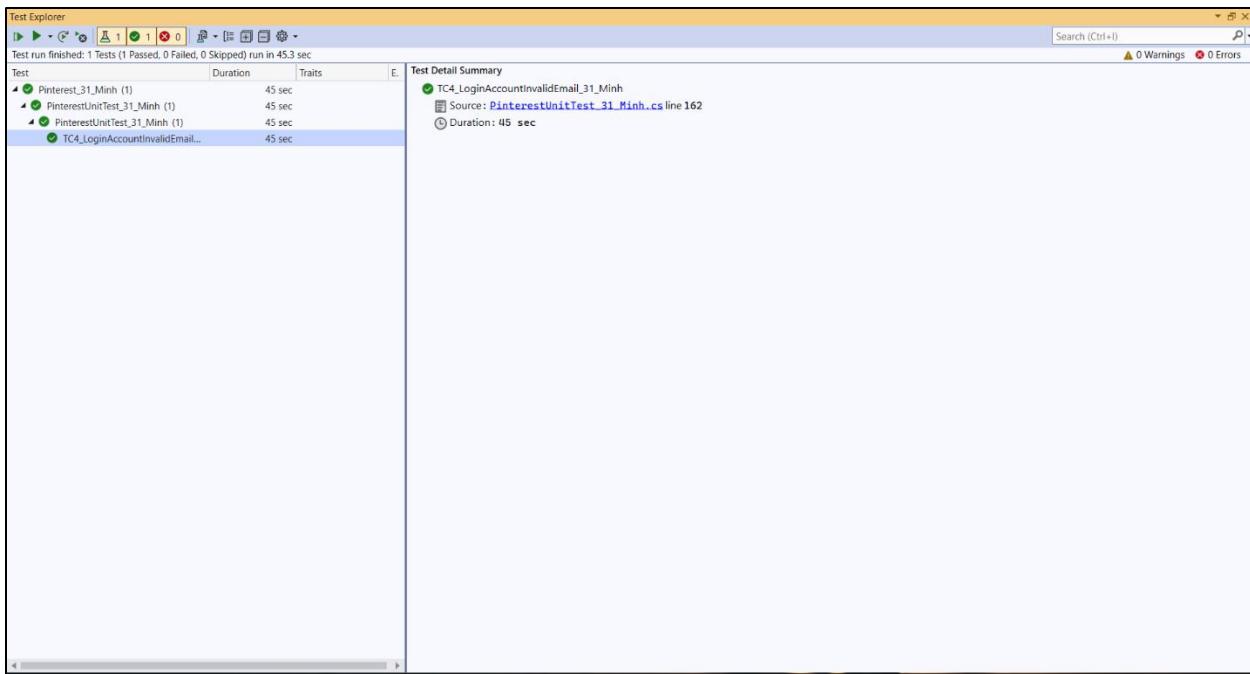
```

```
}

[TestMethod]
public void Quit()
{
    //Đóng tất cả cửa sổ liên kết
    driver_31_Minh.Quit();
}

}
```

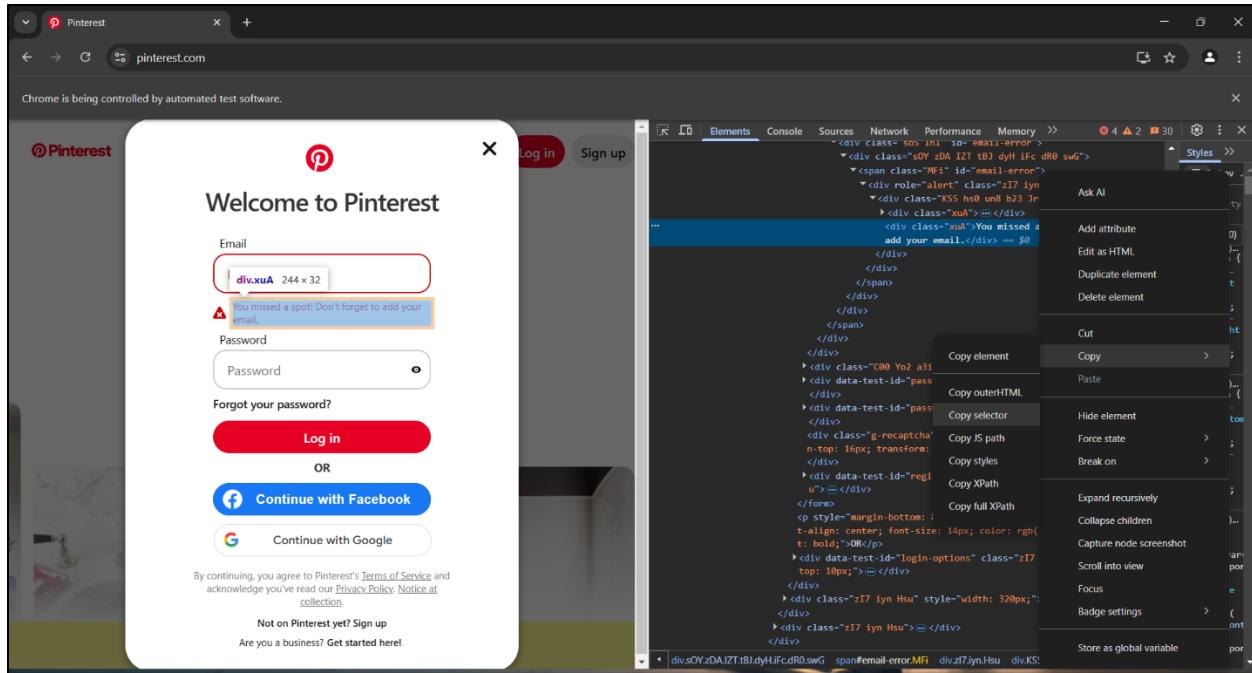




Hình 2.14. Kết quả sau khi chạy TC4_LoginAccountInvalidEmail_31_Minh (31_Minh)

e. Đăng nhập khi để trống thông tin:

- Bước 1: Thực hiện lấy các thuộc tính liên quan đến chức năng đăng nhập giống như trên
- Bước 2: Thủ để trống thông tin đăng nhập rồi đăng nhập:
 - Lấy thuộc tính thông báo lỗi (màu đỏ): Bôi dòng thông báo lỗi > Nhấn chuột phải > Inspect (Kiểm tra) > Tìm đến thẻ chức nội dung > Nhấn chuột phải > Copy > Copy selector



Hình 2.15. Lấy thuộc tính của nội dung thông báo lỗi khi để trống thông tin đăng nhập (31_Minh)

- Bước 3: Viết code và chạy test case:

```

using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using System.Threading;
using OpenQA.Selenium.Interactions;
using System.IO;

namespace PinterestUnitTest_31_Minh
{
    [TestClass]
    public class PinterestUnitTest_31_Minh
    {
        private IWebDriver driver_31_Minh;

        //Địa chỉ trang web
        private string url_31_Minh = "https://www.pinterest.com/";

        //Tên tài khoản
        private string email_31_Minh = "";

        //Mật khẩu
        private string password_31_Minh = "";

        [TestInitialize]
        public void SetUp_31_Minh()
        {
    
```

```

//Khởi tạo đối tượng ChromeDriver
driver_31_Minh = new ChromeDriver();

ChromeDriverService chrome_31_Minh =
ChromeDriverService.CreateDefaultService();

//Đóng Command Prompt khi thực thi kiểm thử
chrome_31_Minh.HideCommandPromptWindow = true;

//Mở trình duyệt ở chế độ toàn màn hình
driver_31_Minh.Manage().Window.Maximize();
}

//Test case 5: Để trống thông tin đăng nhập
//email_31_Minh = ''
//password_31_Minh = ''
[TestMethod]
public void TC5_LoginAccountEmptyFields_31_Minh()
{
    //Điều hướng trình duyệt tới trang web
    driver_31_Minh.Navigate().GoToUrl(url_31_Minh);
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút "Log in" bằng cách sử dụng FindElement CSS
    Selector
    driver_31_Minh.FindElement(By.CssSelector("#__PWS_ROOT__ > div > div >
div:nth-child(1) > div:nth-child(2) > div.QLY._he.zI7.iyn.Hsu > div >
div.Eqh.Jea.KS5.s7I.zI7.iyn.Hsu > div.H-G.zI7.iyn.Hsu > button")).Click();
    Thread.Sleep(10000);

    //Điền tên đăng nhập vào ô "Email" bằng cách sử dụng FindElement Id
    driver_31_Minh.FindElement(By.Id("email")).SendKeys(email_31_Minh);

    //Điền mật khẩu vào ô "Password" bằng cách sử dụng FindElement Id
    driver_31_Minh.FindElement(By.Id("password")).SendKeys(password_31_Minh);
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút submit "Log in" bằng cách sử dụng
    FindElement CSS Selector
    driver_31_Minh.FindElement(By.CssSelector("button[type='submit']")).Click();
    Thread.Sleep(10000);

    //Tìm kiếm và lấy thông báo không hợp lệ bằng cách sử dụng FindElement
    CSS Selector
    IWebElement error_31_Minh =
    driver_31_Minh.FindElement(By.CssSelector("#email-error > div > div > div:nth-
    child(2)"));

    //Chuyển kiểu dữ liệu của error_31_Minh thành string
    string errorMessage_31_Minh = error_31_Minh.Text;

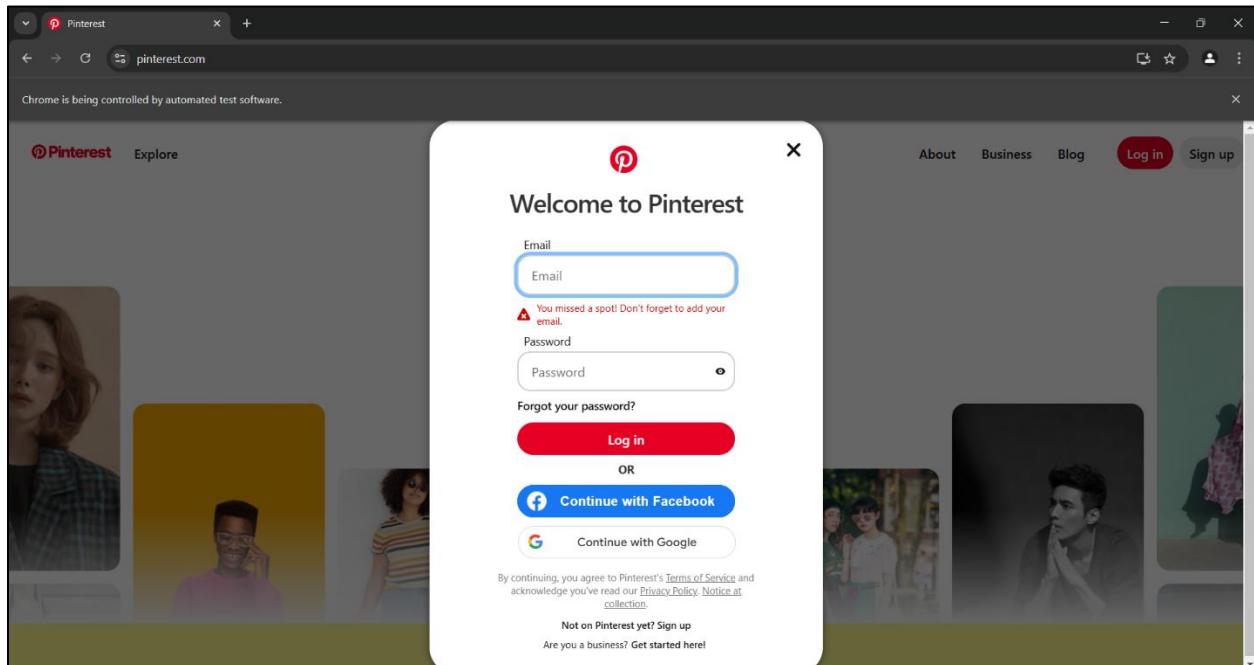
    //So sánh kết quả thực tế (errorMessage_31_Minh) giống kết quả mong muốn
    ("You missed a spot! Don't forget to add your email.")
    Assert.IsTrue(errorMessage_31_Minh.Contains("You missed a spot! Don't
    forget to add your email."));
}

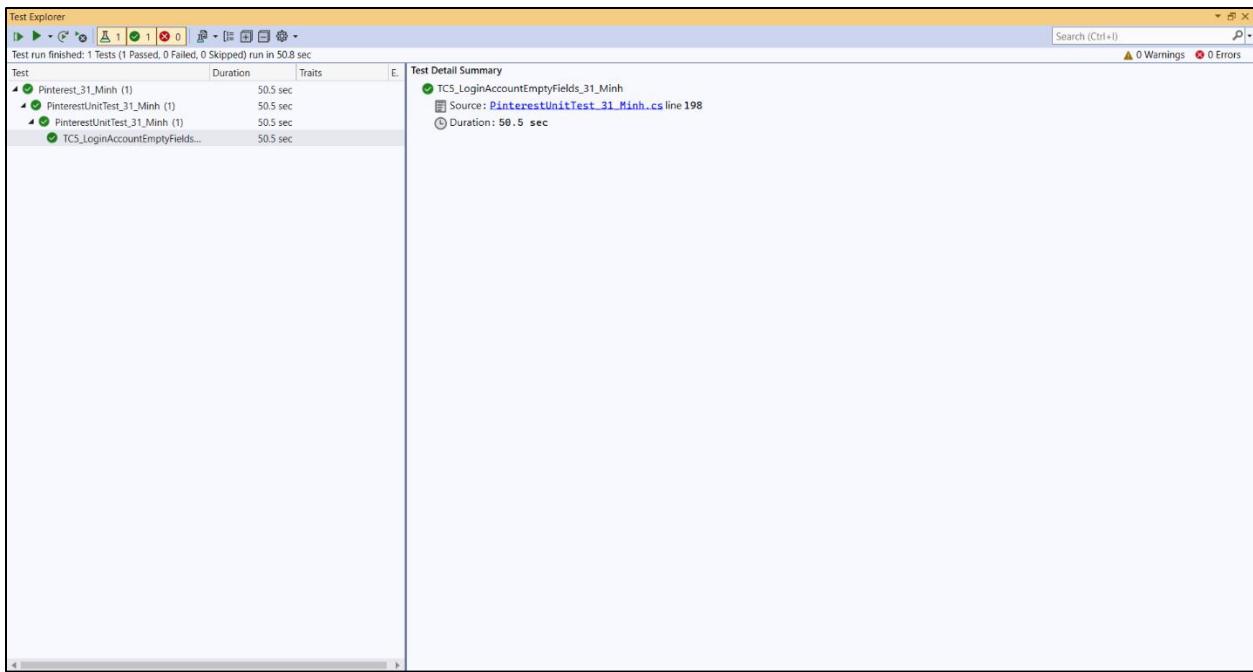
```

```
}

[TestMethod]
public void Quit()
{
    //Đóng tất cả cửa sổ liên kết
    driver_31_Minh.Quit();
}

}
```





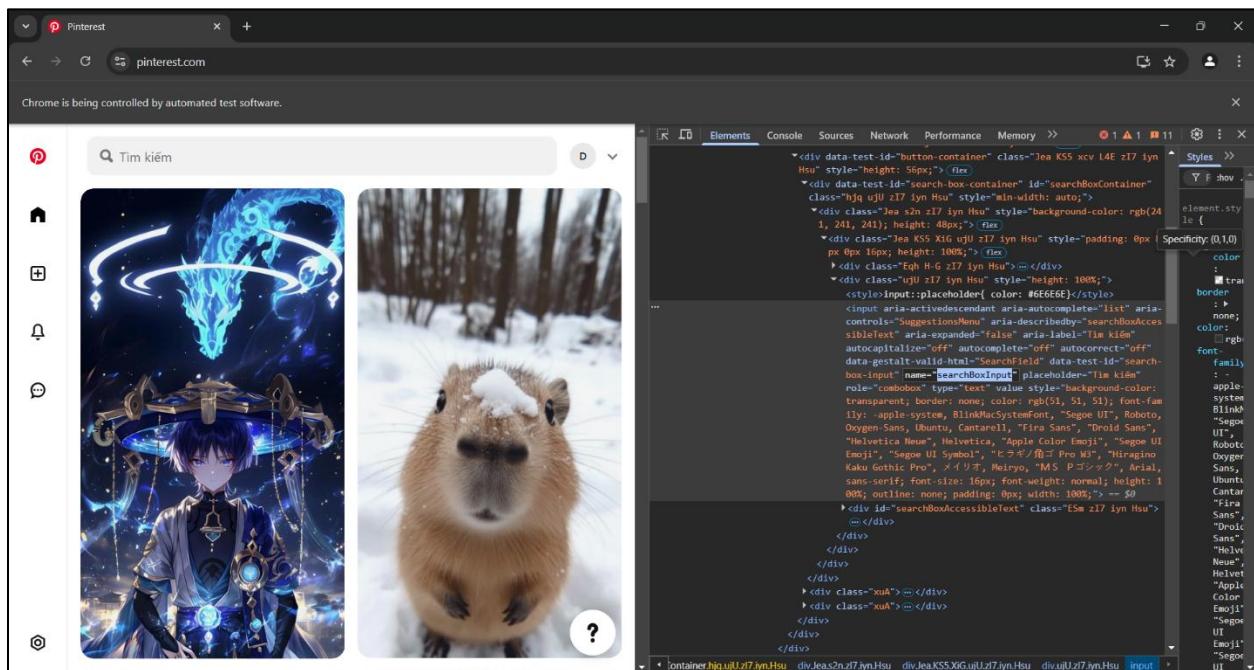
Hình 2.16. Kết quả sau khi chạy TC5_LoginAccountEmptyFields_31_Minh (31_Minh)

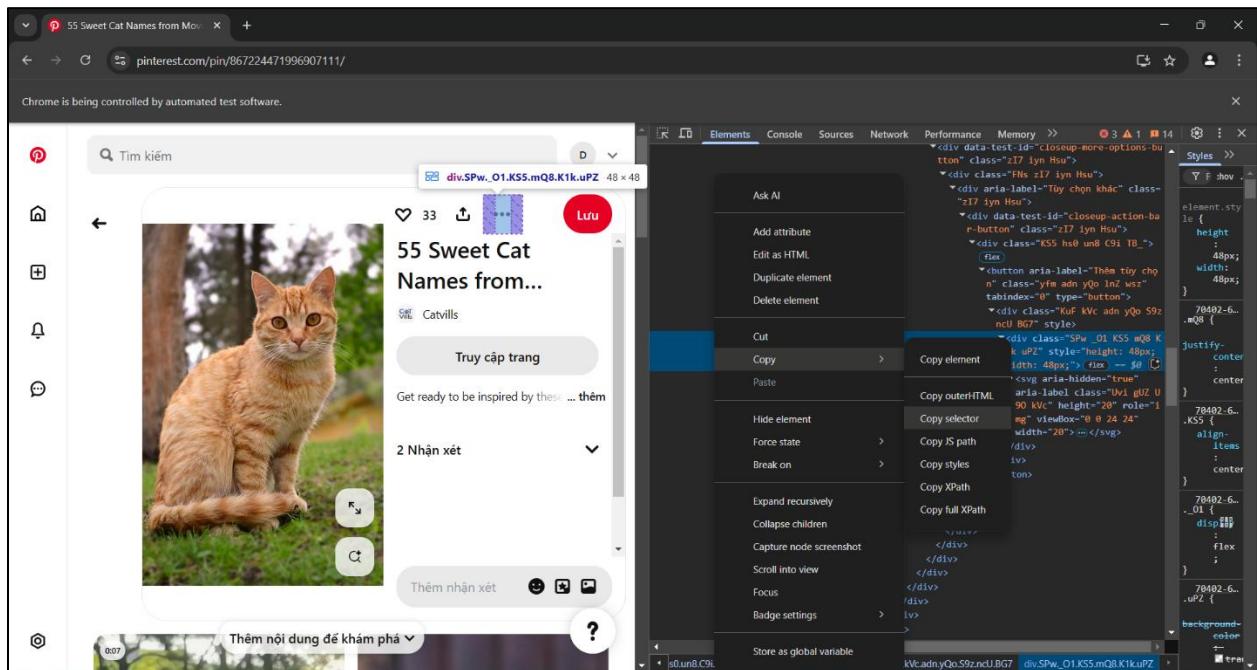
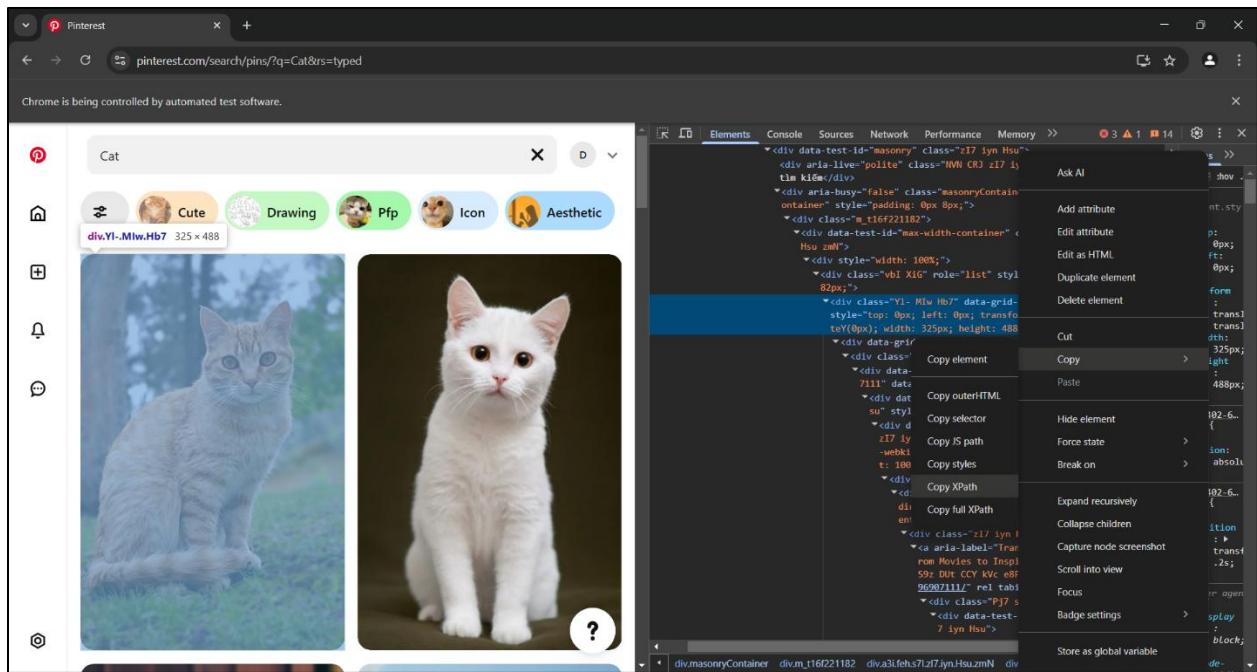
2.3.3. *Chức năng tải hình ảnh về*:

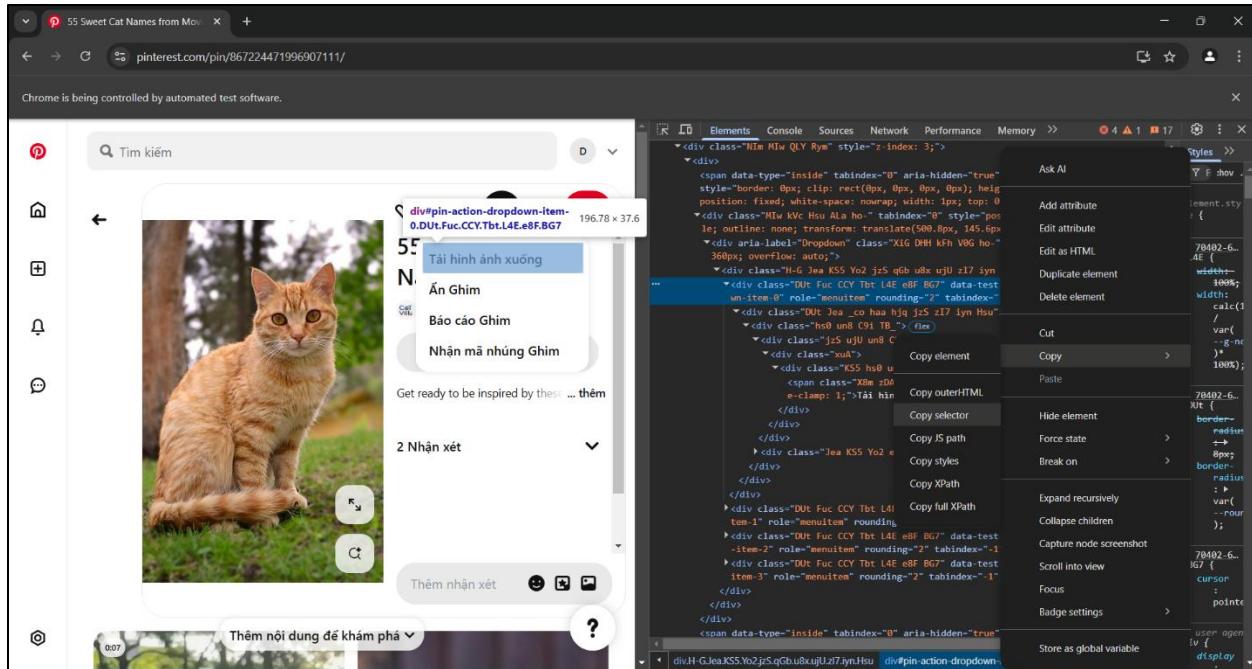
a. Đăng nhập và tải hình ảnh về thành công:

- Bước 1: Thực hiện lấy các thuộc tính liên quan đến chức năng đăng nhập giống như trên
- Bước 2: Tải hình ảnh xuống khi đăng nhập thành công:
 - Lấy thuộc tính thanh công cụ “Tìm kiếm”: Nhấn chuột phải vào thanh công cụ “Tìm kiếm” > Inspect (Kiểm tra) > Tìm đến thẻ chứa thanh công cụ > Lấy thuộc tính name=“searchBoxInput”
 - Nhập từ khóa để tìm kiếm một chủ đề hình ảnh
 - Lấy thuộc tính của hình ảnh đầu tiên: Nhấn chuột phải vào hình ảnh đầu tiên > Inspect (Kiểm tra) > Tìm đến thẻ chứa hình ảnh > Nhấn chuột phải > Copy > Copy Xpath
 - Nhấn vào hình ảnh đầu tiên

- Lấy thuộc tính của nút ba chấm: Nhấn chuột phải vào nút ba chấm > Inspect (Kiểm tra) > Tìm đến thẻ chứa nút > Nhấn chuột phải > Copy > Copy selector
- Nhấn vào nút ba chấm
- Lấy thuộc tính của nút “Tải hình ảnh xuống”: Nhấn chuột phải vào nút “Tải hình ảnh xuống” > Inspect (Kiểm tra) > Tìm đến thẻ chứa nút > Nhấn chuột phải > Copy > selector







Hình 2.17. Các bước lấy các thuộc tính liên quan đến chức năng tải hình ảnh về khi đăng nhập (31_Minh)

- Bước 3: Viết code và chạy test case:

```

using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using System.Threading;
using OpenQA.Selenium.Interactions;
using System.IO;

namespace PinterestUnitTest_31_Minh
{
    [TestClass]
    public class PinterestUnitTest_31_Minh
    {
        private IWebDriver driver_31_Minh;

        //Địa chỉ trang web
        private string url_31_Minh = "https://www.pinterest.com/";

        //Tên tài khoản
        private string email_31_Minh = "ducminh2004@gmail.com";

        //Mật khẩu
        private string password_31_Minh = "minhpro123";

        //Từ khóa
        private string keySearch_31_Minh = "Cat";

        [TestInitialize]
    }
}

```

```

public void SetUp_31_Minh()
{
    //Khởi tạo đối tượng ChromeDriver
    driver_31_Minh = new ChromeDriver();

    ChromeDriverService chrome_31_Minh =
    ChromeDriverService.CreateDefaultService();

    //Đóng Command Prompt khi thực thi kiểm thử
    chrome_31_Minh.HideCommandPromptWindow = true;

    //Mở trình duyệt ở chế độ toàn màn hình
    driver_31_Minh.Manage().Window.Maximize();
}

//Test case 6: Đăng nhập và tải hình ảnh về thành công
//keySearch_31_Minh = 'Cat'
//email_31_Minh = 'ducminh2004@gmail.com'
//password_31_Minh = 'minhpro123'
[TestMethod]
public void TC6_LoginAndSearchImageAndDownload_31_Minh()
{
    //Điều hướng trình duyệt tới trang web
    driver_31_Minh.Navigate().GoToUrl(url_31_Minh);
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút "Log in" bằng cách sử dụng FindElement CSS
    Selector
    driver_31_Minh.FindElement(By.CssSelector("#__PWS_ROOT__ > div > div >
div:nth-child(1) > div:nth-child(2) > div.QLY._he.zI7.iyn.Hsu > div >
div.Eqh.Jea.KS5.s7I.zI7.iyn.Hsu > div.H-G.zI7.iyn.Hsu > button")).Click();
    Thread.Sleep(10000);

    //Điền tên đăng nhập vào ô "Email" bằng cách sử dụng FindElement Id
    driver_31_Minh.FindElement(By.Id("email")).SendKeys(email_31_Minh);

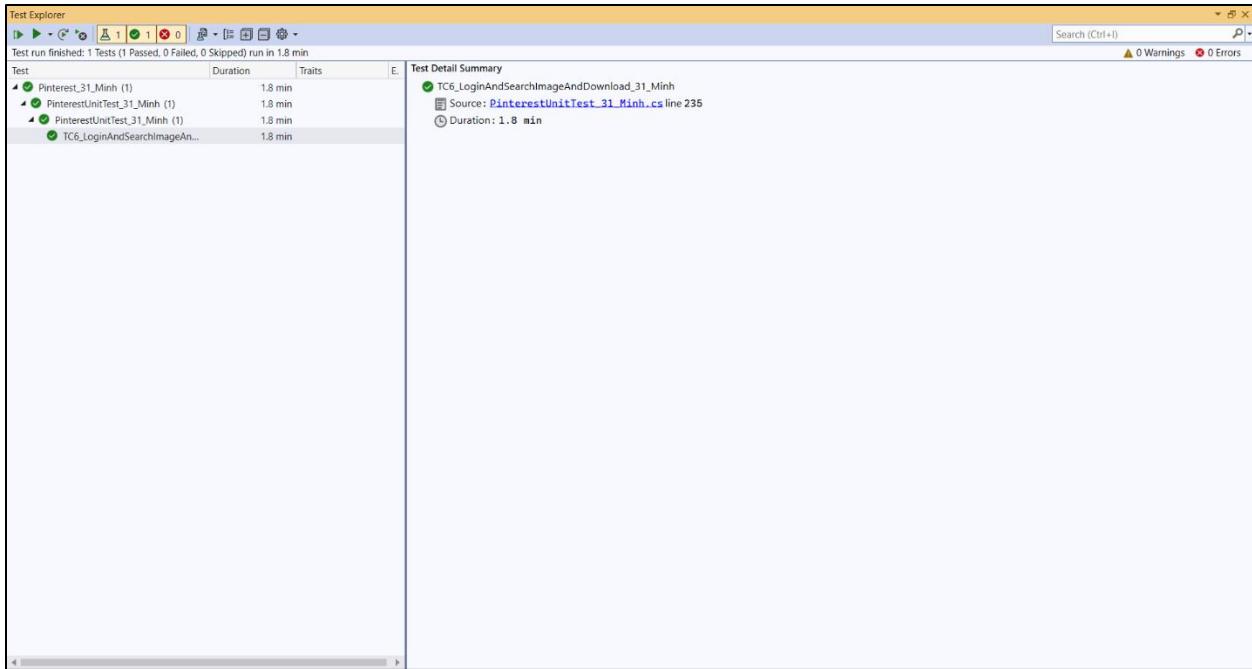
    //Điền mật khẩu vào ô "Password" bằng cách sử dụng FindElement Id
    driver_31_Minh.FindElement(By.Id("password")).SendKeys(password_31_Minh);
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút submit "Log in" bằng cách sử dụng
    FindElement CSS Selector
    driver_31_Minh.FindElement(By.CssSelector("button[type='submit']")).Click();
    Thread.Sleep(10000);

    //Tìm kiếm ô "Tìm kiếm" bằng cách sử dụng FindElement Name
    IWebElement searchImage_31_Minh =
    driver_31_Minh.FindElement(By.Name("searchBoxInput"));
    //Điền tên hình ảnh muốn tìm
    searchImage_31_Minh.SendKeys(keySearch_31_Minh);
    Thread.Sleep(10000);

    //Mô phỏng nhấn phím Enter trên bàn phím
    new Actions(driver_31_Minh).KeyDown(Keys.Enter).Build().Perform();
    Thread.Sleep(10000);
}

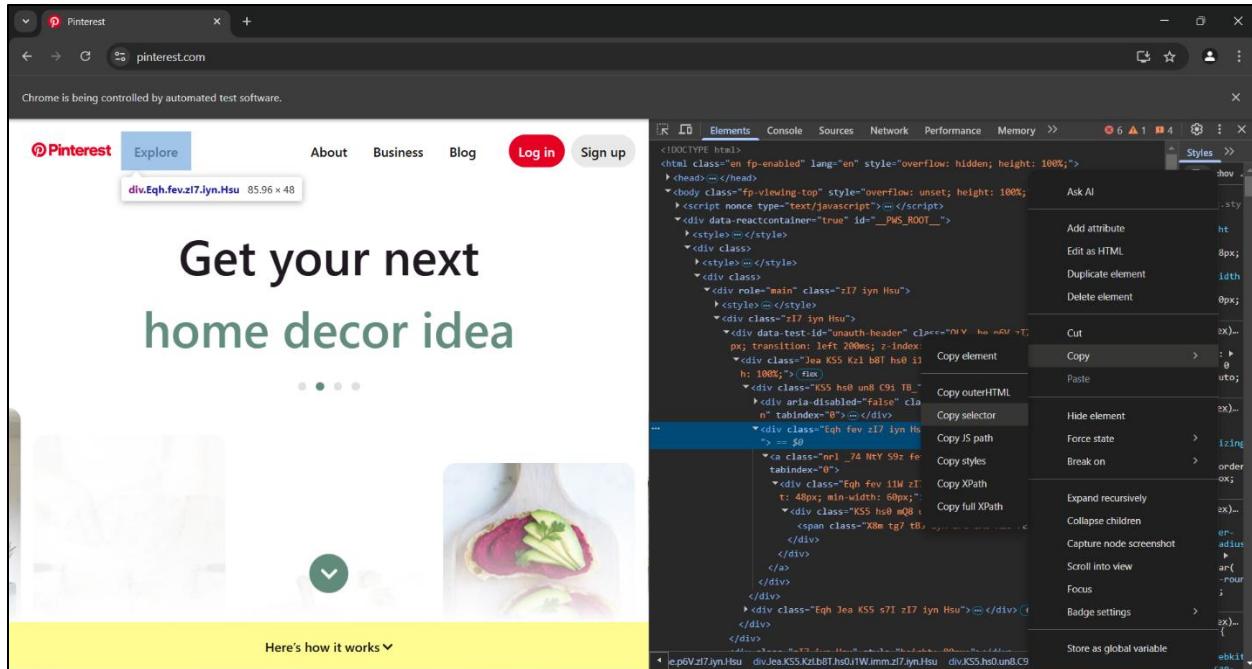
```

Hình 2.18. Kết quả sau khi chạy TC6_LoginAndSearchImageAndDownload_31_Minh (31_Minh)

b. Tải hình ảnh về khi không đăng nhập:

- Bước 1: Lấy thuộc tính nút “Explore”: Nhấn chuột phải vào nút “Explore” > Inspect (Kiểm tra) > Tìm đến thẻ chứa nút > Nhấn chuột phải > Copy > Copy selector
- Bước 2: Thực hiện lấy các thuộc tính liên quan đến chức năng tải hình ảnh xuống như trên



Hình 2.19. Lấy thuộc tính liên quan đến chức năng tải hình ảnh về khi không đăng nhập
(31_Minh)

- Bước 3: Viết code và chạy test case:

```

using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using System.Threading;
using OpenQA.Selenium.Interactions;
using System.IO;

namespace PinterestUnitTest_31_Minh
{
    [TestClass]
    public class PinterestUnitTest_31_Minh
    {
        private IWebDriver driver_31_Minh;

        //Địa chỉ trang web
        private string url_31_Minh = "https://www.pinterest.com/";

        //Từ khóa
        private string keySearch_31_Minh = "Cat";

        [TestInitialize]
        public void SetUp_31_Minh()
        {
            //Khởi tạo đối tượng ChromeDriver
            driver_31_Minh = new ChromeDriver();
        }
    }
}

```

```

ChromeDriverService chrome_31_Minh =
ChromeDriverService.CreateDefaultService();

//Đóng Command Prompt khi thực thi kiểm thử
chrome_31_Minh.HideCommandPromptWindow = true;

//Mở trình duyệt ở chế độ toàn màn hình
driver_31_Minh.Manage().Window.Maximize();
}

//Test case 7: Tải hình ảnh xuống khi không đăng nhập
//keySearch_31_Minh = 'Cat'
[TestMethod]
public void TC7_SearchImageAndDownload_31_Minh()
{
    //Điều hướng trình duyệt tới trang web
    driver_31_Minh.Navigate().GoToUrl(url_31_Minh);
    Thread.Sleep(10000);

    //Tìm kiếm và click vào "Explore" bằng cách sử dụng FindElement CSS
    Selector
    IWebElement explore_31_Minh =
driver_31_Minh.FindElement(By.CssSelector("#__PWS_ROOT__ > div > div > div:nth-child(1) > div:nth-child(2) > div.QLY._he.p6V.zI7.iyn.Hsu > div > div.KS5.hs0.un8.C9i.TB_ > div.Eqh.fev.zI7.iyn.Hsu"));
    explore_31_Minh.Click();

    //Tìm kiếm ô "Tìm kiếm" bằng cách sử dụng FindElement Name
    IWebElement searchImage_31_Minh =
driver_31_Minh.FindElement(By.Name("searchBoxInput"));
    //Điền tên hình ảnh muốn tìm
    searchImage_31_Minh.SendKeys(keySearch_31_Minh);
    Thread.Sleep(10000);

    //Mô phỏng nhấn phím Enter trên bàn phím
    new Actions(driver_31_Minh).KeyDown(Keys.Enter).Build().Perform();
    Thread.Sleep(10000);

    //Tìm kiếm và click vào bức ảnh đầu tiên bằng cách sử dụng FindElement
    CSS Selector
    IWebElement firstImage_31_Minh =
driver_31_Minh.FindElement(By.CssSelector("div[data-test-id=\"pin\"]"));
    firstImage_31_Minh.Click();
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút "..." bằng cách sử dụng FindElement CSS
    Selector
    IWebElement seeMore_31_Minh =
driver_31_Minh.FindElement(By.CssSelector("div[data-test-id=\"ellipsis-button\"]"));
    seeMore_31_Minh.Click();
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút "Download image" bằng cách sử dụng
    FindElement CSS Selector
    IWebElement btnDownload_31_Minh =
driver_31_Minh.FindElement(By.Id("desktop-context-menu-item-0"));
    btnDownload_31_Minh.Click();
    Thread.Sleep(10000);
}

```

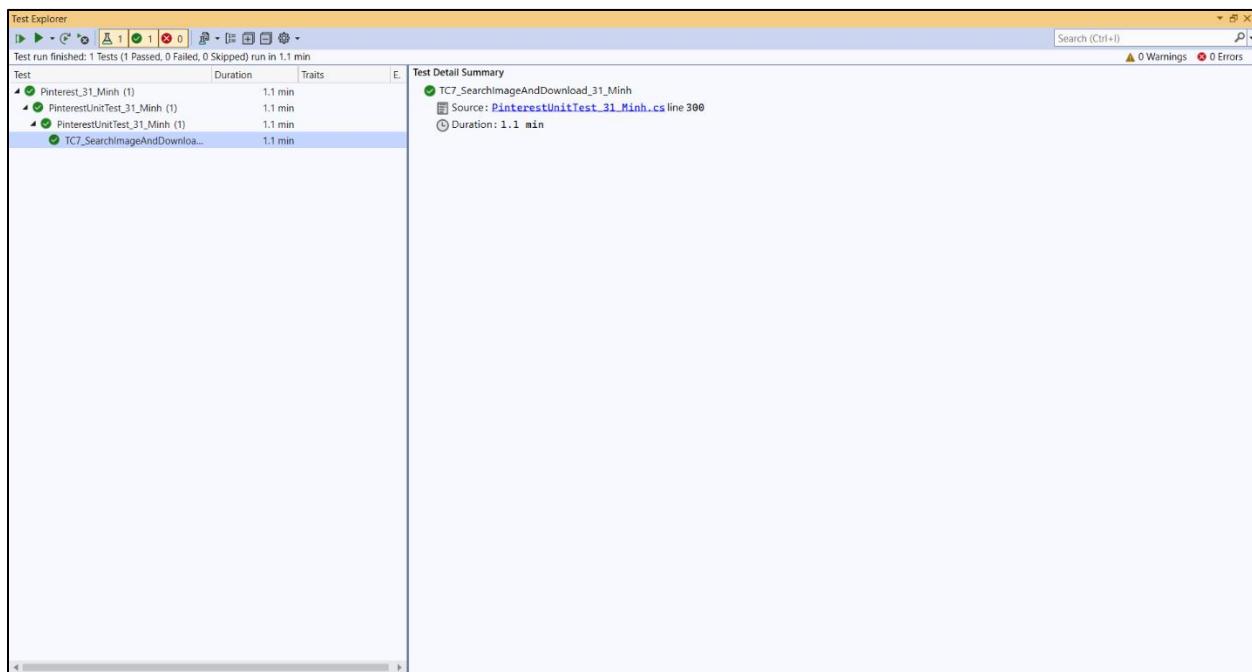
```

//Tim kiém thông báo yêu cầu đăng nhập để tải ảnh bằng cách sử dụng
.FindElement XPath
IWebElement request_31_Minh =
driver_31_Minh.FindElement(By.XPath("//*[@id=\"__PWS_ROOT__\"]/div/div/div[1]/div[4]
/div/div/div/div[1]/div[3]/h1"));
//Chuyển kiểu dữ liệu của request_31_Minh thành string
string requestLogin_31_Minh = request_31_Minh.Text;

Assert.AreEqual(requestLogin_31_Minh, "Sign up to download");
}

[TestCleanup]
public void Quit()
{
    //Đóng tất cả cửa sổ liên kết
    driver_31_Minh.Quit();
}
}
}

```



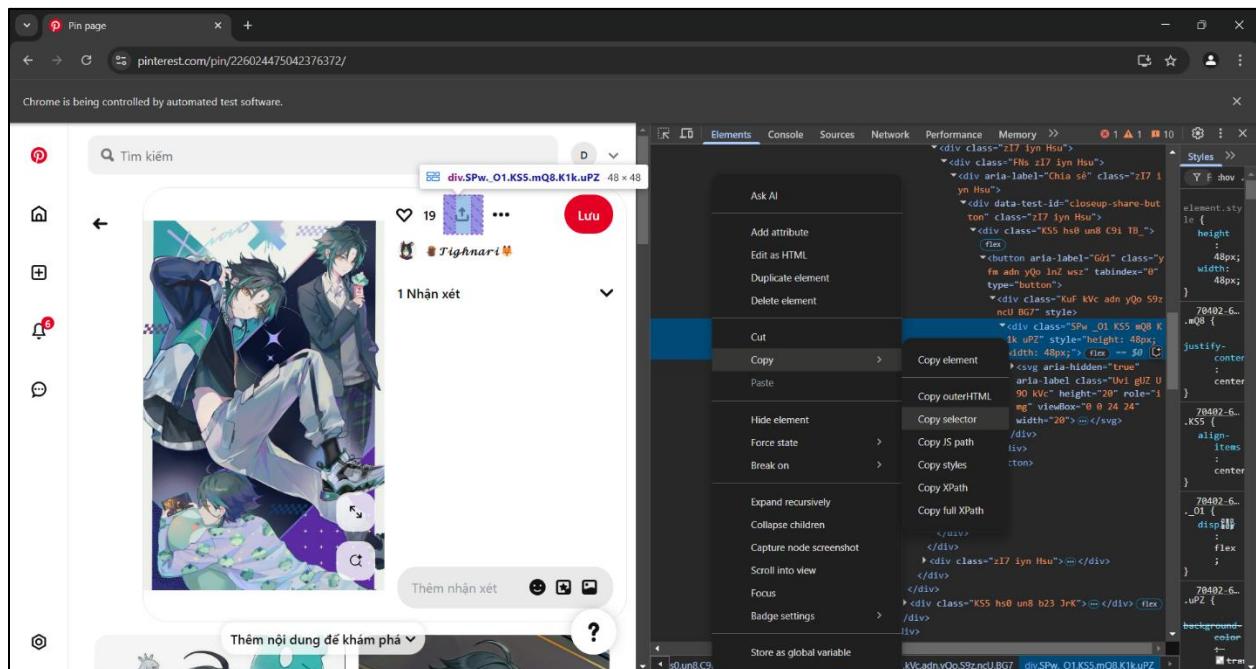
Hình 2.20. Kết quả sau khi chạy TC7_SearchImageAndDownload_31_Minh (31_Minh)

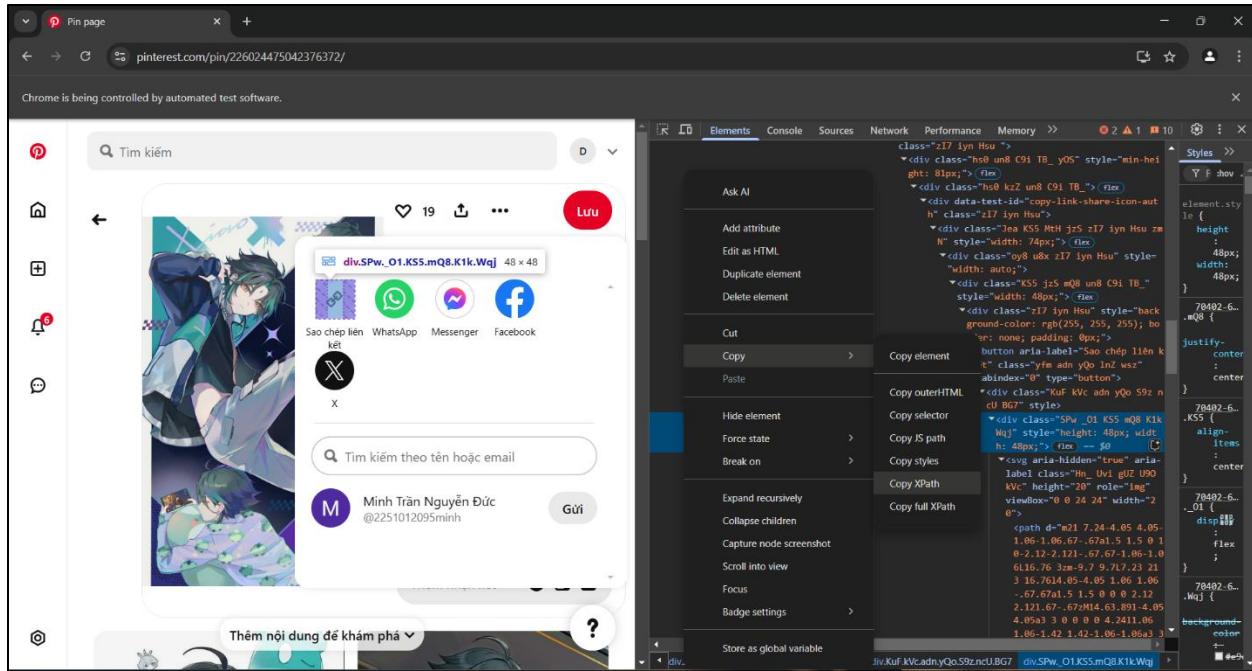
2.3.4. **Chức năng chia sẻ hình ảnh (Copy link):**

a. Đăng nhập và chia sẻ hình ảnh (Copy link):

- Bước 1: Thực hiện lấy các thuộc tính liên quan đến chức năng đăng nhập giống như trên

- Bước 2: Lấy các thuộc tính liên quan của chức năng chia sẻ hình ảnh (Copy link):
 - Lấy thuộc tính của nút có biểu tượng chia sẻ: Nhấn chuột phải vào nút > Inspect (Kiểm tra) > Tìm đến thẻ chứa nút > Nhấn chuột phải > Copy > Copy selector
 - Lấy thuộc tính của nút “Sao chép liên kết”: Nhấn chuột phải vào nút “Sao chép liên kết” > Inspect (Kiểm tra) > Tìm đến thẻ chứa nút > Nhấn chuột phải > Copy > Copy XPath





Hình 2.21. Các bước lấy thuộc tính liên quan đến chức năng chia sẻ hình ảnh (Copy link) khi đăng nhập (31_Minh)

- Bước 3: Viết code và chạy test case:

```

using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using System.Threading;
using OpenQA.Selenium.Interactions;
using System.IO;

namespace PinterestUnitTest_31_Minh
{
    [TestClass]
    public class PinterestUnitTest_31_Minh
    {
        private IWebDriver driver_31_Minh;

        //Địa chỉ trang web
        private string url_31_Minh = "https://www.pinterest.com/";

        //Tên tài khoản
        private string email_31_Minh = "ducminh2004@gmail.com";

        //Mật khẩu
        private string password_31_Minh = "minhpro123";

        //Từ khóa
        private string keySearch_31_Minh = "Cat";
    }
}

```

```

//Đường dẫn hình ảnh
private string link_31_Minh = "https://pin.it/";

[TestInitialize]
public void SetUp_31_Minh()
{
    //Khởi tạo đối tượng ChromeDriver
    driver_31_Minh = new ChromeDriver();

    ChromeDriverService chrome_31_Minh =
    ChromeDriverService.CreateDefaultService();

    //Đóng Command Prompt khi thực thi kiểm thử
    chrome_31_Minh.HideCommandPromptWindow = true;

    //Mở trình duyệt ở chế độ toàn màn hình
    driver_31_Minh.Manage().Window.Maximize();
}

//Test case 8: Kiểm tra link hợp lệ khi đăng nhập và chia sẻ hình ảnh bằng
cách copy link
//keySearch_31_Minh = 'Cat'
//email_31_Minh = 'ducminh2004@gmail.com'
//password_31_Minh = 'minhpro123'
[TestMethod]
public void TC8_LoginAndShareLinkImage_31_Minh()
{
    driver_31_Minh.Navigate().GoToUrl(url_31_Minh); //Điều hướng trình duyệt
    tới trang web
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút "Log in" bằng cách sử dụng FindElement CSS
    Selector
    driver_31_Minh.FindElement(By.CssSelector("#__PWS_ROOT__ > div > div >
    div:nth-child(1) > div:nth-child(2) > div.QLY._he.zI7.iyn.Hsu > div >
    div.Eqh.Jea.KS5.s7I.zI7.iyn.Hsu > div.H-G.zI7.iyn.Hsu > button")).Click();
    Thread.Sleep(10000);

    //Điền tên đăng nhập vào ô "Email" bằng cách sử dụng FindElement Id
    driver_31_Minh.FindElement(By.Id("email")).SendKeys(email_31_Minh);
    //Điền mật khẩu vào ô "Password" bằng cách sử dụng FindElement Id

    driver_31_Minh.FindElement(By.Id("password")).SendKeys(password_31_Minh);
    Thread.Sleep(10000);

    //Tìm kiếm và click vào nút submit "Log in" bằng cách sử dụng
    FindElement CSS Selector

    driver_31_Minh.FindElement(By.CssSelector("button[type='submit']")).Click();
    Thread.Sleep(10000);

    //Tìm kiếm ô "Tìm kiếm" bằng cách sử dụng FindElement Name
    IWebElement searchImage_31_Minh =
    driver_31_Minh.FindElement(By.Name("searchBoxInput"));
    //Điền tên hình ảnh muốn tìm
    searchImage_31_Minh.SendKeys(keySearch_31_Minh);
    Thread.Sleep(10000);
}

```

```

//Mô phỏng nhấn phím Enter trên bàn phím
new Actions(driver_31_Minh).KeyDown(Keys.Enter).Build().Perform();
Thread.Sleep(10000);

//Tìm kiếm và click vào bức ảnh đầu tiên bằng cách sử dụng FindElement
XPath
IWebElement firstImage_31_Minh =
driver_31_Minh.FindElement(By.XPath("//*[@id=\"__PWS_ROOT__\"]/div/div/div[1]/div/div[2]/div/div/div[4]/div/div/div/div[2]/div/div/div/div[1]/div[1]"));
firstImage_31_Minh.Click();
Thread.Sleep(10000);

//Tìm kiếm và click vào nút "Share" bằng cách sử dụng FindElement CSS
Selector
IWebElement btnShare_31_Minh =
driver_31_Minh.FindElement(By.CssSelector("#gradient > div > div > div.ujU.zI7.iyn.Hsu > div > " +
"div > div.Jea.jzS.zI7.iyn.Hsu > div.qiB > div > div > div > div > div.DUT.Jea.KS5.b8T.imm.zI7.iyn.Hsu > div > div:nth-child(2) > " +
"div > div > div > div > div > button > div > div > svg"));
btnShare_31_Minh.Click();
Thread.Sleep(10000);

//Tìm kiếm và click vào nút "Copy link" bằng cách sử dụng FindElement
XPath
IWebElement copy_31_Minh =
driver_31_Minh.FindElement(By.XPath("//*[@id=\"gradient\"]/div/div/div[2]/div/div/div/div/div/div/div/div[2]/div[1]/div/div/div[1]/div/div[2]/div/div/div/div[2]/div[1]/div/div/div[1]/div/div[1]/div/div/button/div/div"));
copy_31_Minh.Click();
Thread.Sleep(10000);

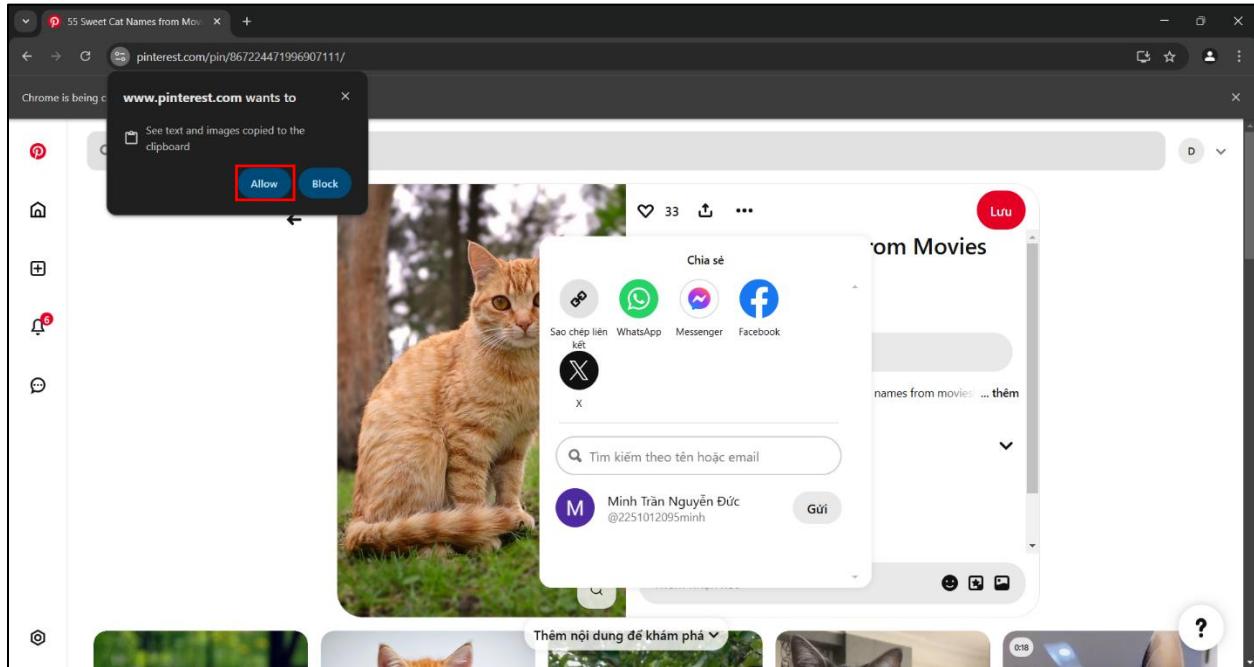
//Ép kiểu driver thành IJavaScriptExecutor (một interface của Selenium
cho phép thực thi JavaScript trong trình duyệt)
IJavaScriptExecutor drv_31_Minh = (IJavaScriptExecutor)driver_31_Minh;
//Đọc dữ liệu đã được sao chép từ clipboard (bộ nhớ tạm của hệ điều
hành) bằng API của JavaScript (return navigator.clipboard.readText();) rồi gán cho
copiedLink
string copiedLink_31_Minh = drv_31_Minh.ExecuteScript("return
navigator.clipboard.readText();").ToString();

//So sánh kết quả mong muốn (copiedLink_31_Minh) và kết quả thực tế
(link_31_Minh)
Assert.IsTrue(copiedLink_31_Minh.StartsWith(link_31_Minh), "Link không
khớp");
}

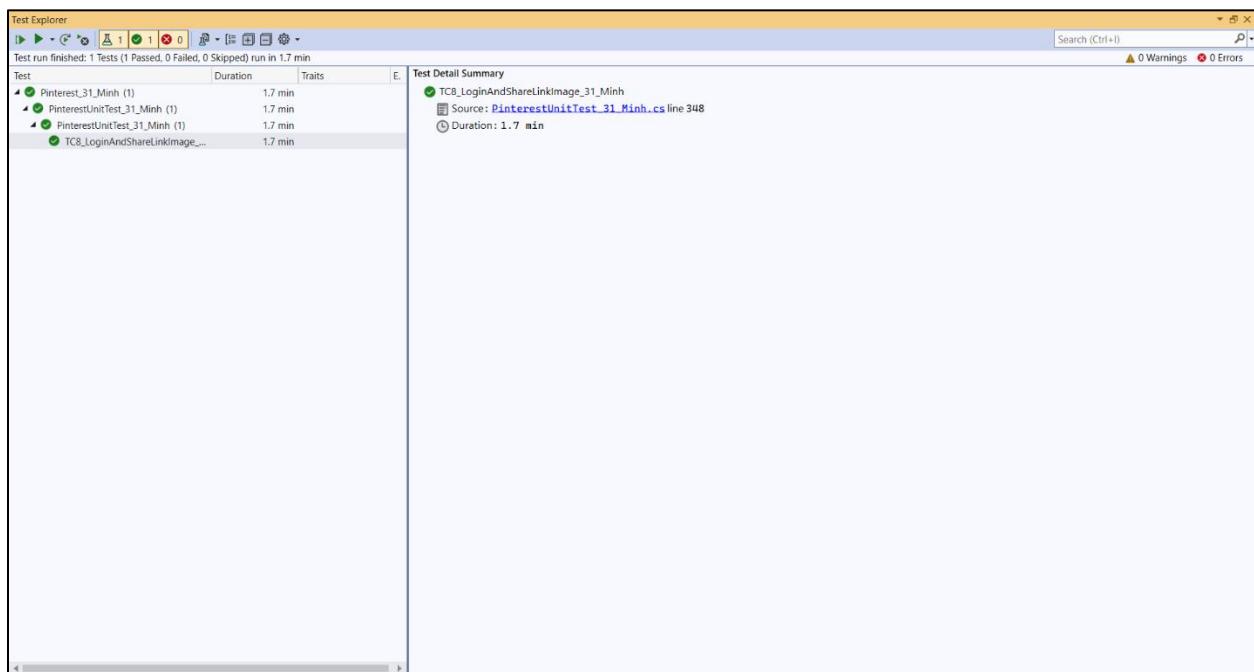
[TestCleanup]
public void Quit()
{
    //Đóng tất cả cửa sổ liên kết
    driver_31_Minh.Quit();
}
}
}

```

- **Lưu ý:** Phải cấp quyền xem văn bản đã được sao chép trong clipboard.



Hình 2.22. Cấp quyền được xem văn bản đã sao chép trong trình duyệt (31_Minh)



Hình 2.23. Kết quả sau khi chạy TC8_LoginAndShareLinkImage_31_Minh (31_Minh)

b. Chia sẻ hình ảnh (Copy link) khi không đăng nhập:

- Bước 1: Thực hiện lấy các thuộc tính liên quan đến chức năng chia sẻ hình ảnh (Copy link) như trên
- Bước 2: Viết code và chạy test case:

```

using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using System.Threading;
using OpenQA.Selenium.Interactions;
using System.IO;

namespace PinterestUnitTest_31_Minh
{
    [TestClass]
    public class PinterestUnitTest_31_Minh
    {
        private IWebDriver driver_31_Minh;

        //Địa chỉ trang web
        private string url_31_Minh = "https://www.pinterest.com/";

        //Từ khóa
        private string keySearch_31_Minh = "Cat";

        //Đường dẫn hình ảnh
        private string link_31_Minh = "https://pin.it/";

        [TestInitialize]
        public void SetUp_31_Minh()
        {
            //Khởi tạo đối tượng ChromeDriver
            driver_31_Minh = new ChromeDriver();

            ChromeDriverService chrome_31_Minh =
            ChromeDriverService.CreateDefaultService();

            //Đóng Command Prompt khi thực thi kiểm thử
            chrome_31_Minh.HideCommandPromptWindow = true;

            //Mở trình duyệt ở chế độ toàn màn hình
            driver_31_Minh.Manage().Window.Maximize();
        }

        //Test case 9: Kiểm tra link hợp lệ khi chia sẻ hình ảnh bằng cách copy link
        //mà không đăng nhập
        //keySearch_31_Minh = 'Cat'
        [TestMethod]
        public void TC9_ShareLinkImage_31_Minh()
        {
            driver_31_Minh.Navigate().GoToUrl(url_31_Minh); //Điều hướng trình duyệt
            // đến trang web
            Thread.Sleep(10000);
        }
    }
}

```

```

    //Tìm kiếm và click vào "Explore" bằng cách sử dụng FindElement CSS
    Selector
    IWebElement explore_31_Minh =
driver_31_Minh.FindElement(By.CssSelector("#__PWS_ROOT__ > div > div > div:nth-child(1) > div:nth-child(2) > div.QLY._he.p6V.zI7.iyn.Hsu > div >
div.KS5.hs0.un8.C9i.TB_ > div.Eqh.fev.zI7.iyn.Hsu"));
explore_31_Minh.Click();

    //Tìm kiếm ô "Tìm kiếm" bằng cách sử dụng FindElement Name
    IWebElement searchImage_31_Minh =
driver_31_Minh.FindElement(By.Name("searchBoxInput"));
    //Điền tên hình ảnh muốn tìm
    searchImage_31_Minh.SendKeys(keySearch_31_Minh);
Thread.Sleep(10000);

    //Mô phỏng nhấn phím Enter trên bàn phím
    new Actions(driver_31_Minh).KeyDown(Keys.Enter).Build().Perform();
Thread.Sleep(10000);

    //Tìm kiếm và click vào bức ảnh đầu tiên bằng cách sử dụng FindElement
    CSS Selector
    IWebElement firstImage_31_Minh =
driver_31_Minh.FindElement(By.CssSelector("div[data-test-id=\"pin\"]"));
firstImage_31_Minh.Click();
Thread.Sleep(10000);

    //Tìm kiếm và click vào nút "Share" bằng cách sử dụng FindElement CSS
    Selector
    IWebElement btnShare_31_Minh =
driver_31_Minh.FindElement(By.CssSelector("div[data-test-id=\"share-button\"]"));
btnShare_31_Minh.Click();
Thread.Sleep(10000);

    //Tìm kiếm và click vào nút "Copy link" bằng cách sử dụng FindElement
    CSS Selector
    IWebElement copy_31_Minh =
driver_31_Minh.FindElement(By.CssSelector("div[data-test-id=\"copy-link-share-
icon\"]"));
copy_31_Minh.Click();
Thread.Sleep(10000);

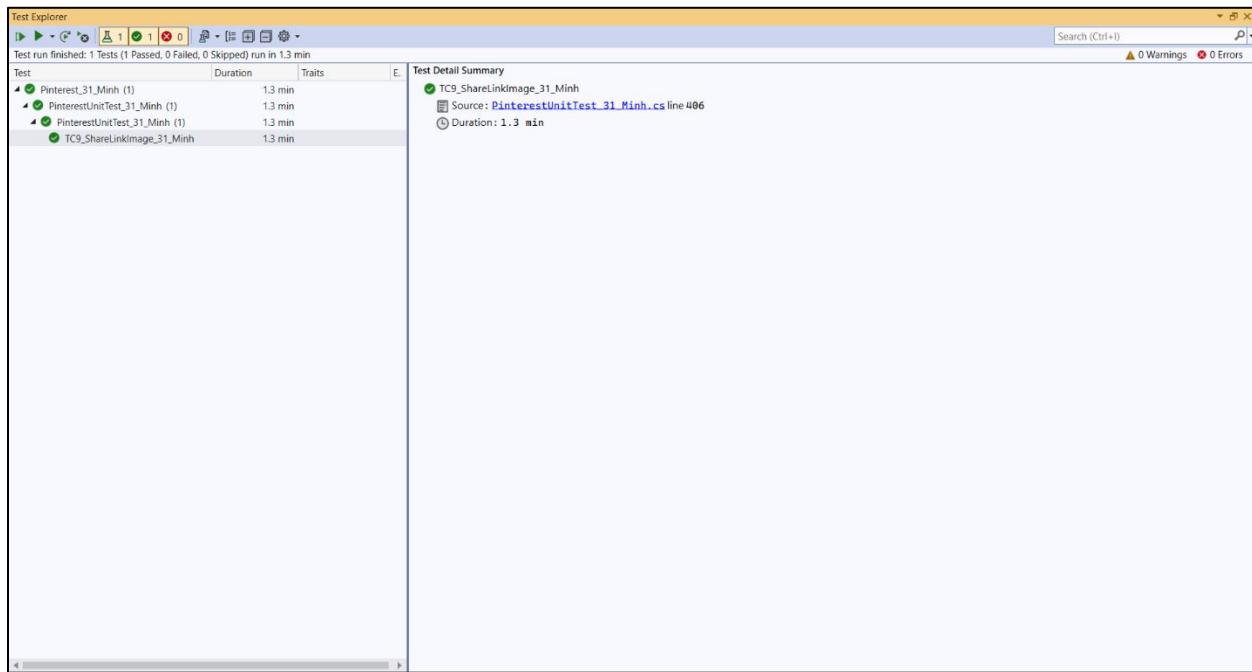
    //Ép kiểu driver thành IJavaScriptExecutor (một interface của Selenium
    cho phép thực thi JavaScript trong trình duyệt)
    IJavaScriptExecutor drv_31_Minh = (IJavaScriptExecutor)driver_31_Minh;
    //Đọc dữ liệu đã được sao chép từ clipboard (bộ nhớ tạm của hệ điều
    hành) bằng API của JavaScript (return navigator.clipboard.readText();) rồi gán cho
    copiedLink
    string copiedLink_31_Minh = drv_31_Minh.ExecuteScript("return
navigator.clipboard.readText();").ToString();

    //So sánh kết quả mong muốn (copiedLink_31_Minh) và kết quả thực tế
    (link_31_Minh)
    Assert.IsTrue(copiedLink_31_Minh.StartsWith(link_31_Minh), "Link không
    khớp");
}

    [TestCleanup]
    public void Quit()

```

```
{  
    //Đóng tất cả cửa sổ liên kết  
    driver_31_Minh.Quit();  
}  
}
```



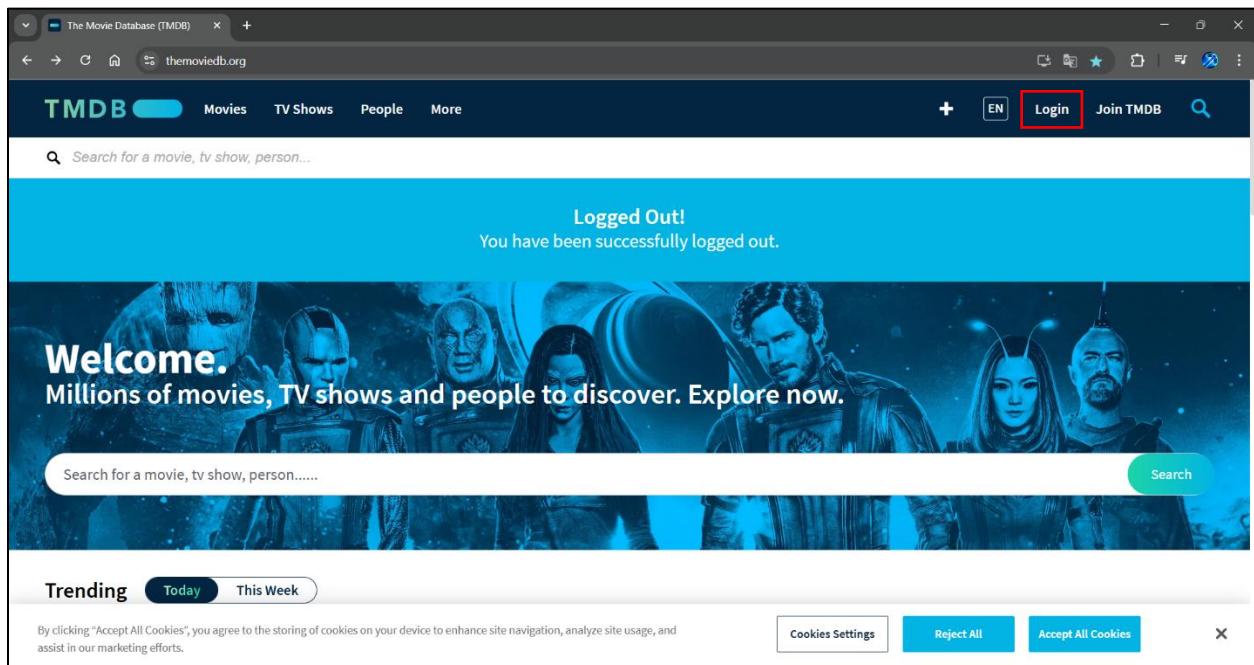
Hình 2.24. Kết quả sau khi chạy TC9_ShareLinkImage_31_Minh (31_Minh)

CHƯƠNG 3. TEST API BẰNG POSTMAN

3.1. Test API có sẵn

3.1.1. Đăng nhập website *The Movie Database (TMDB)* và xin cấp quyền truy cập API:

- Bước 1: Mở Google Chrome > Tìm kiếm website <https://www.themoviedb.org/> > Login > Điền Username và Password > Nhấn submit Login
- Bước 2: Nhấn vào hồ sơ của mình > Chọn Settings > Chọn API > Chọn “click here” > Yes > Điền đầy đủ thông tin > Chọn “Access your API key details here.”
- Bước 3: Chọn “clicking here” > API Reference



TMDB

Movies TV Shows People More

EN Login Join TMDB

Login to your account

In order to use the editing and rating capabilities of TMDB, as well as get personal recommendations you will need to login to your account. If you do not have an account, registering for an account is free and simple. [Click here](#) to get started.

If you signed up but didn't get your verification email, [click here](#) to have it resent.

Username

Password

Login [Reset password](#)

THE MOVIE THE BASICS GET INVOLVED COMMUNITY LEGAL

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

Cookies Settings Reject All Accept All Cookies

My Profile — The Movie Database

TMDB

Movies TV Shows People More

ducminh2004 Member since March 2025

0* Average Movie Score | 0* Average TV Score

Overview Discussions Lists Ratings Watchlist

Stats

Total Edits: 0 Total Ratings: 0

Rating Overview

1 2 3 4 5 6 7 8 9 10

Upcoming From Watchlist

There are no upcoming movies on your watchlist.

Go to Watchlist

ducminh2004

View profile

Discussions

Lists

Ratings

Watchlist

Edit Profile

Settings

Logout

Most Watched Genres

You haven't logged any movies or TV shows.

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

Cookies Settings Reject All Accept All Cookies

My Settings — The Movie Database

TMDB

Movies TV Shows People More

EN D

Edit Profile English (en-US)

Account Settings Fallback Language None (Don't Fallback)

Streaming Services Country Vietnam

Notification Settings Timezone - Auto detect? Asia - Kabul

Blocked Users

Import List

Sharing Settings Include Adult Items in Search? No

Sessions Filter Profanity? Yes

API

Delete Account Enable Keyboard Shortcuts? Yes

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and provide in-app personalization effects. <https://www.themoviedb.org/settings/api>

Cookies Settings Reject All Accept All Cookies

My API Settings — The Movie Database

TMDB

Movies TV Shows People More

D ducminh2004

API Overview Create

Settings

Edit Profile

Account Settings

Streaming Services

Notification Settings

Blocked Users

Import List

Sharing Settings

Sessions

TMDB offers a powerful API service that is free for personal use. Please ensure you attribute TMDB for any images or data you use. You can find the logos for attribution [here](#).

Documentation Our primary documentation is located at [developer.themoviedb.org](#).

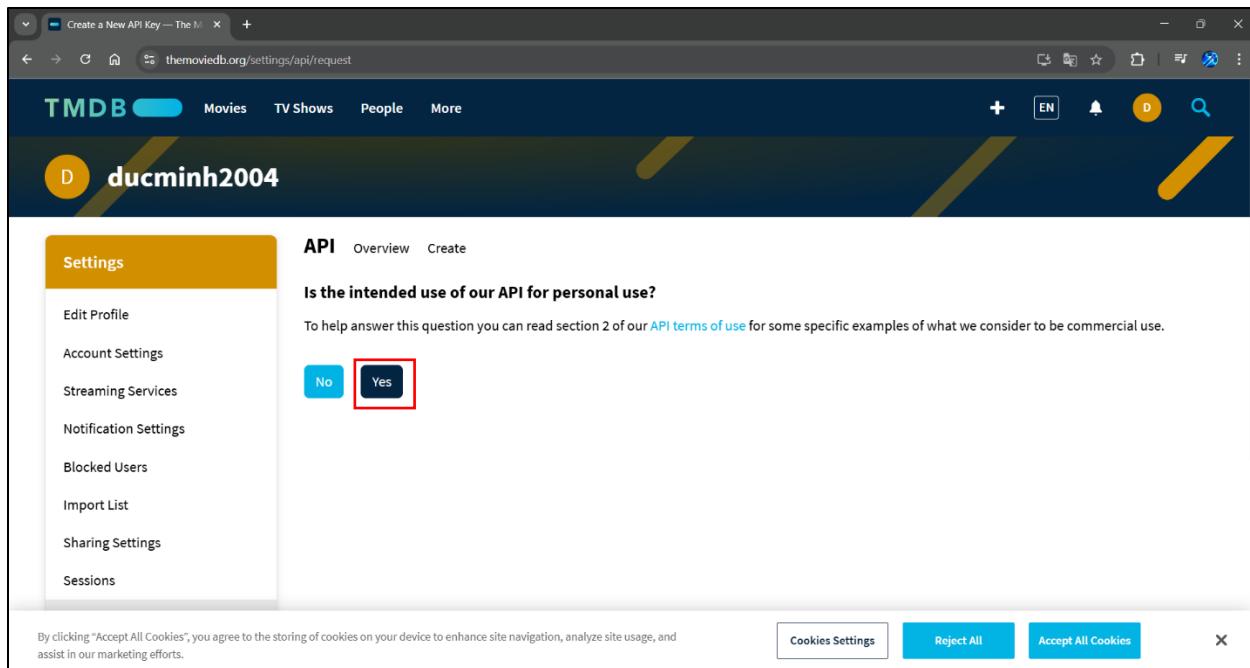
Support If you have questions or comments about the information covered here, please create a post on our [support forums](#).

Request an API Key

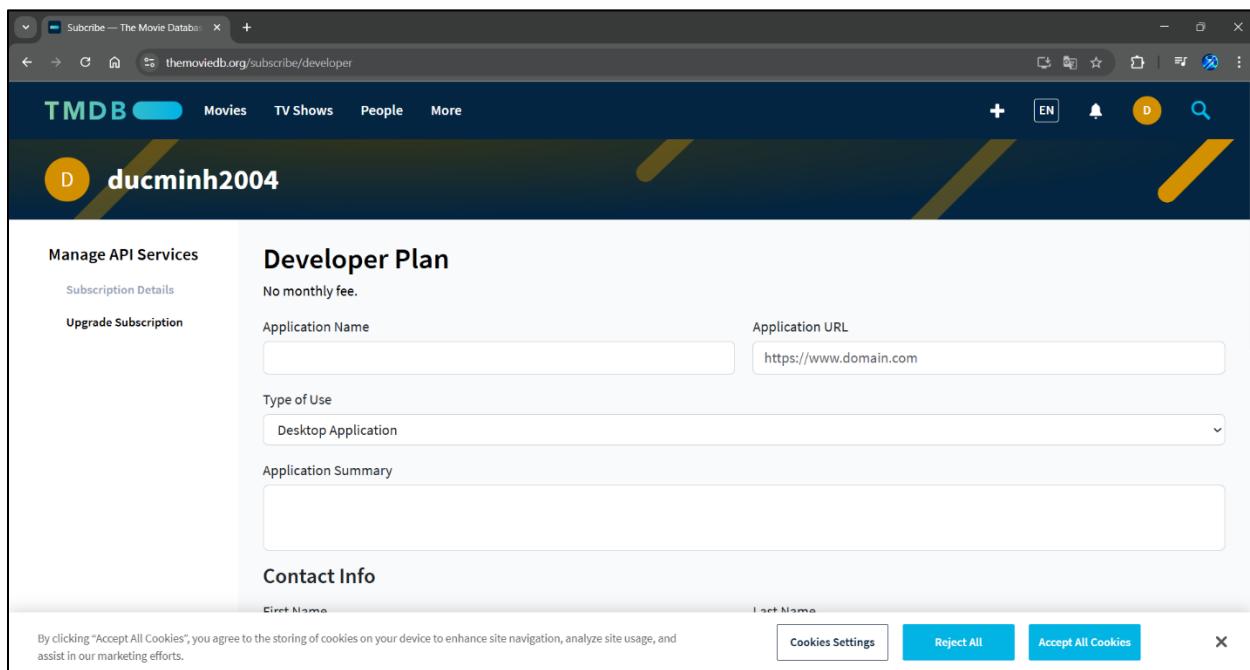
To generate a new API key, [click here](#).

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and provide in-app personalization effects. <https://www.themoviedb.org/settings/api/request>

Cookies Settings Reject All Accept All Cookies



The screenshot shows the TMDB API settings page. On the left, a sidebar titled 'Settings' contains links for Edit Profile, Account Settings, Streaming Services, Notification Settings, Blocked Users, Import List, Sharing Settings, and Sessions. The main content area is titled 'API' with sub-links for Overview and Create. A question 'Is the intended use of our API for personal use?' is displayed, with two buttons: 'No' (blue) and 'Yes' (black). The 'Yes' button is highlighted with a red box. Below the question, a note states: 'To help answer this question you can read section 2 of our [API terms of use](#) for some specific examples of what we consider to be commercial use.' At the bottom, a cookie consent banner asks: 'By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.' It includes buttons for 'Cookies Settings', 'Reject All', 'Accept All Cookies', and a close 'X' button.



The screenshot shows the TMDB developer plan page. On the left, a sidebar titled 'Manage API Services' has links for Subscription Details and Upgrade Subscription. The main content area is titled 'Developer Plan' and states 'No monthly fee.' It includes fields for 'Application Name' (input field), 'Application URL' (input field with value 'https://www.domain.com'), 'Type of Use' (dropdown menu with 'Desktop Application' selected), and 'Application Summary' (input field). Below this is a 'Contact Info' section with 'First Name' and 'Last Name' fields. At the bottom, a cookie consent banner is identical to the one in the previous screenshot.

Manage API Services

Subscription Details

You are currently on the **Free Developer plan**.

[Access your API key details here.](#)

Subscription Details

Upgrade Subscription

Support

THE MOVIE DB

THE BASICS

- About TMDB
- Contact Us
- Support Forums
- API Documentation
- System Status

GET INVOLVED

- Contribution Bible
- Add New Movie
- Add New TV Show

COMMUNITY

- Guidelines
- Discussions
- Leaderboard

LEGAL

- Terms of Use
- API Terms of Use
- Privacy Policy
- DMCA Policy

Hi ducminh2004!

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

Cookies Settings

Reject All

Accept All Cookies

API New

Overview Upgrade Details Sessions Stats Regenerate Key

TMDB offers a powerful API service that is free for personal use. Please ensure you attribute TMDB for any images or data you use. You can find the logos for attribution [here](#).

Documentation

Our primary documentation is located at [developer.themoviedb.org](#).

Support

If you have questions or comments about the information covered here, please create a post on our [support forums](#).

API Details

Test your API credentials by [clicking here](#). You can also edit the details of your app by [clicking here](#).

API Read Access Token

```
eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyZTNlYTQxNjY2NzC4NGI4MzUxYjYwZDgxMDM2ZmNjMlsIm5iZlI6MTc0MzI1NDA2NC44ODMwMDAxLCJzdWIiOiZ2N2U3ZjZMDhmmMzAzZDImOTVkZGEzZGEiLCJzY29wZXMiOlsYXBpX3JlYWQXswidmVyc2lvbli6MX0.mZdyfYcrU4plgNFKGINGKwJHwGIUUTGM5g1uhB4w
```

API Key

2e3ea416637784b8351b60d81036cc2

TMDB

Getting Started

API Reference

Changelog

THE BASICS

Getting Started

FAQ

Popularity & Trending

Roadmap

Wrappers & Libraries

AUTHENTICATION

Application

User

Guest Sessions

USING THE API

Append To Response

Daily ID Exports

Errors

Finding Data

JSON & JSONP

Languages

Getting Started

Get started with the basics of the TMDB API.

Welcome to version 3 of The Movie Database (TMDB) API. This is where you will find the definitive list of currently available methods for our movie, tv, actor and image API. If you need help or support, please head over to our [API support forum](#).

To register for an API key, click the [API link](#) from within your account settings page.

Please note that the API registration process is *not* optimized for mobile devices so you should access these pages on a desktop computer and browser.

Before being issued an API key you will have to agree to our terms of use. You can read that [here](#).

A few useful tips...

- The [configuration](#) methods are useful to get the static lists of data we use throughout the database.
 - You can find things like the languages, countries, timezones and translations that we use. The [configuration](#) method also holds useful image information.
- Understanding the basics of our authentication is useful. You can read about this [here](#).
- We enforce some amount of rate limiting on the API. You can read about that [here](#).

TMDB

Getting Started

API Reference

Changelog

JUMP TO

DEVELOPER DASHBOARD

Getting Started

Authentication

My Requests

ACCOUNT

Details

Add Favorite

Add To Watchlist

Favorite Movies

Favorite TV

Lists

Rated Movies

Rated TV

Rated TV Episodes

Watchlist Movies

Watchlist TV

Getting Started

Welcome to version 3 of The Movie Database (TMDB) API. This is where you will find the definitive list of currently available methods for our movie, tv, actor and image API.

1 Pick a language

Shell Node Ruby PHP Python Java C# :

2 Credentials

HEADER

eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyZTNlYTQxNjYzNzc4NGI	Access Token Auth
--	-------------------

MY TOP ENDPOINTS LAST 30 DAYS

MY REQUESTS LAST 30 DAYS

API Requests

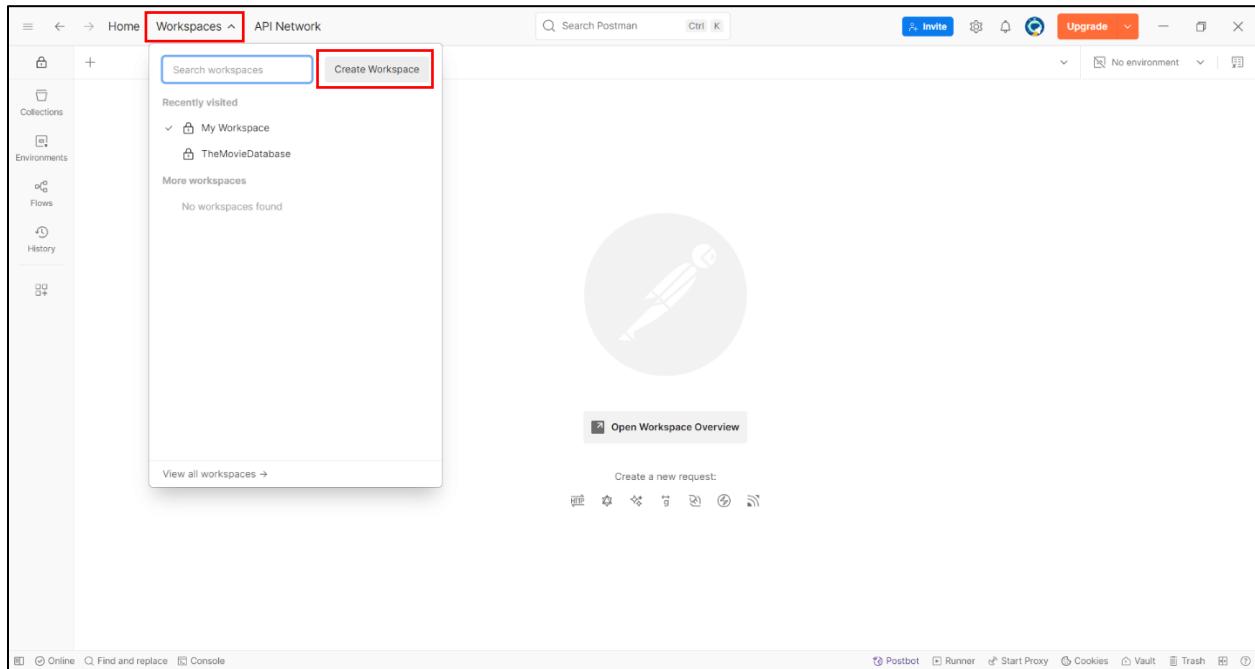
Success Error

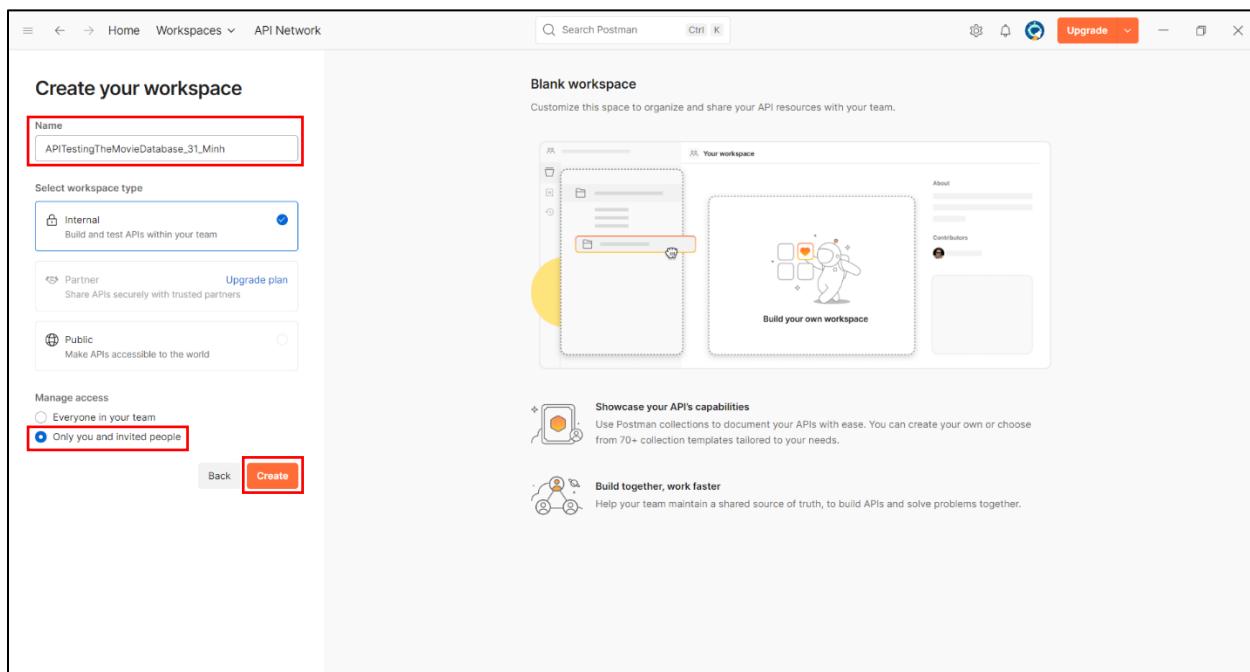
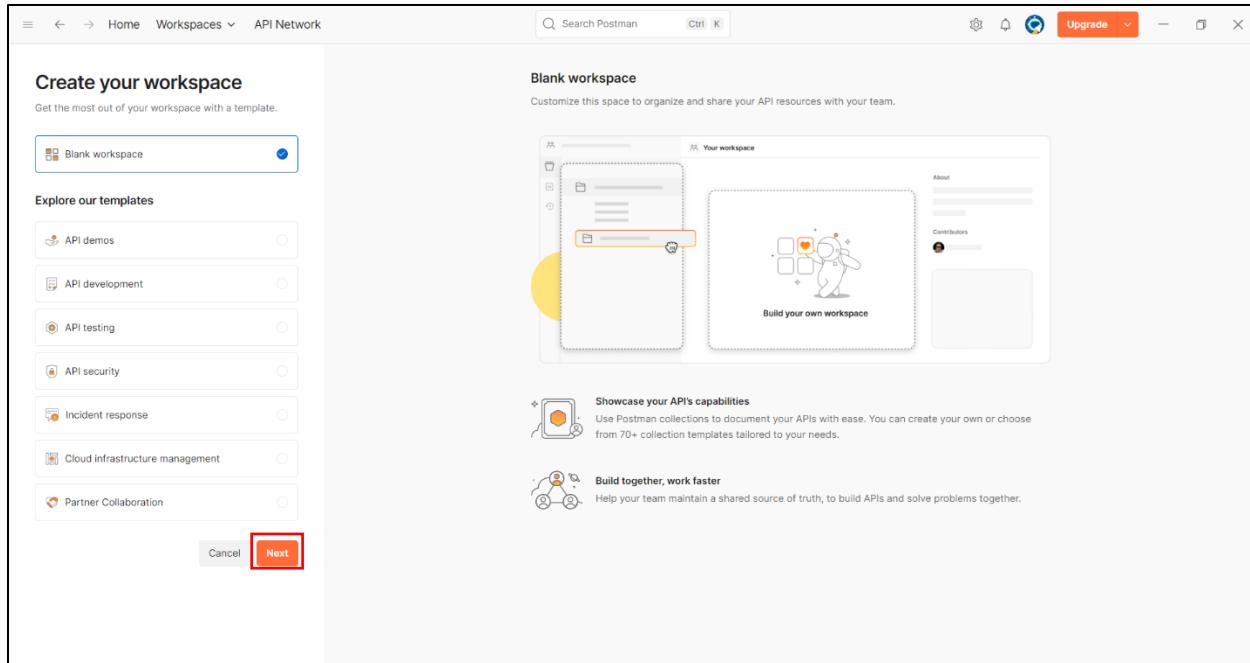
ALL REQUESTS

Hình 3.1. Các bước đăng nhập tài khoản và xin cấp quyền truy cập API (31_Minh)

3.1.2. Tao request để test API:

- Mở Postman > Workspaces > Create Workspaces > Next > Đặt tên là **APITestingTheMovieDatabase_31_Minh** > Tick vào ô “Only you and invited people” > Create



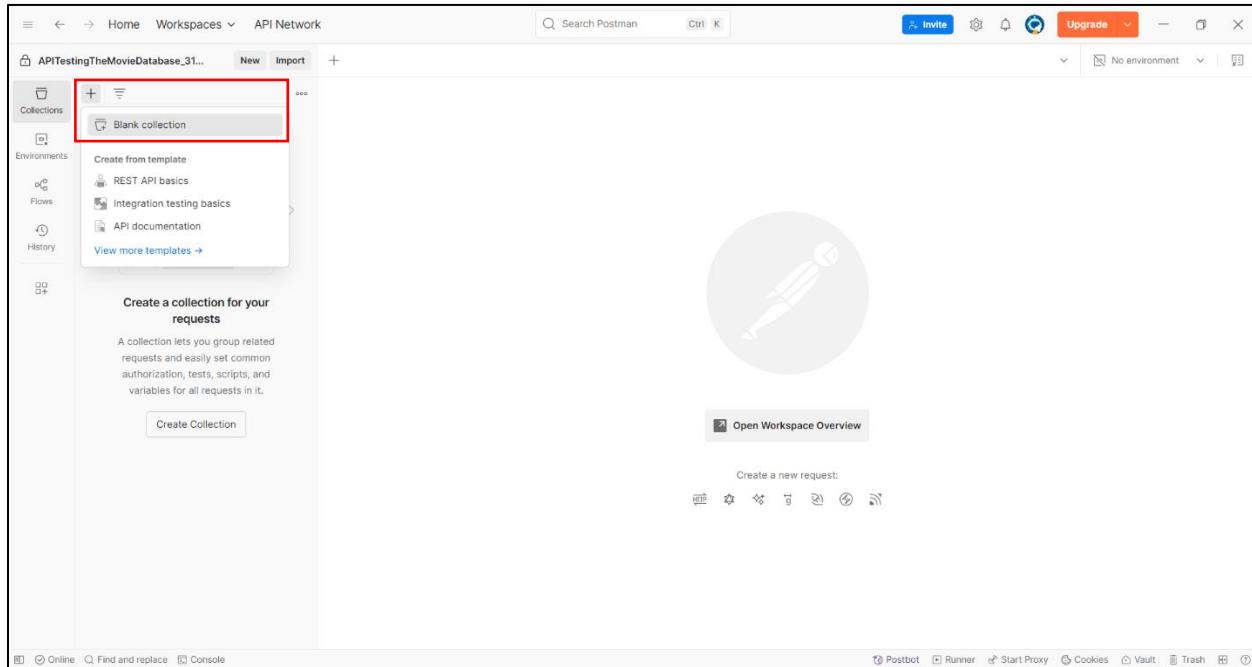


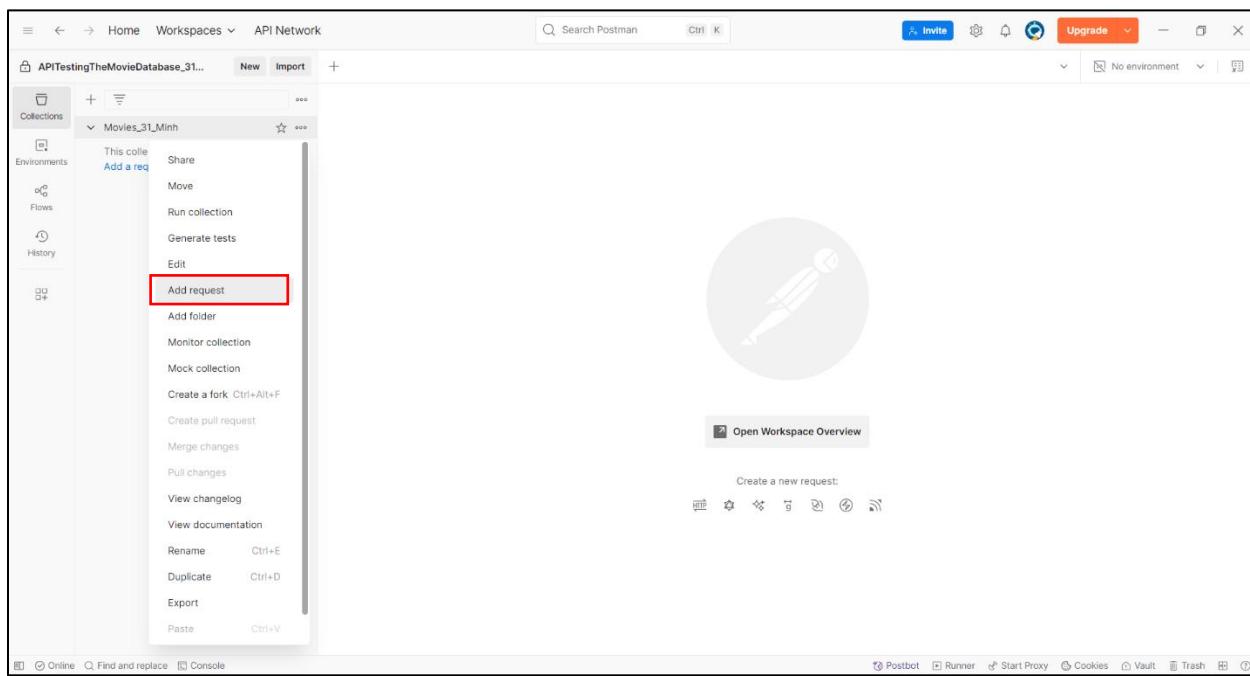
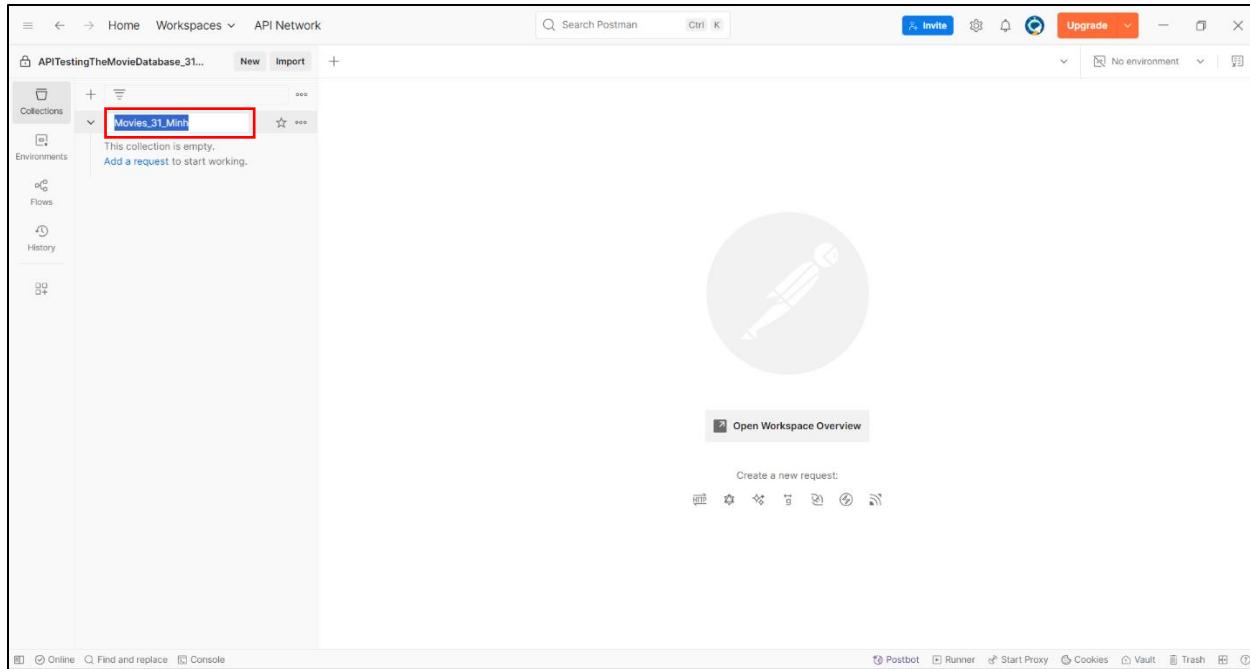
Hình 3.2. Các bước tạo workspaces (31_Minh)

- Bước 1: Tạo Blank collection > Đặt tên là **Movies_31_Minh** > Nhấn chuột phải vào **Movies_31_Minh** > Add request > Đặt tên request là

TC1_31_Minh_Get_ListMovie > Chính sang phương thức Get (Tạo các request còn lại tương tự **TC2_31_Minh_Post_AddFavoriteMovie** phương thức Post và **TC3_31_Minh_Delete_RatingMovie** phương thức Delete)

- Bước 2: Sang trang web TMDB để lấy “API Read Access Token” > Quay trở lại Postman > Chọn Headers > Thêm một Key: Authorization với Value: Bearer + “API Read Access Token” (Làm tương tự với các request còn lại)
- Bước 3: Copy các API lấy được từ TMDB như hình bên dưới > Dán vào request trong Postman > Send
 - **TC1_31_Minh_Get_ListMovie:**
<https://api.themoviedb.org/3/discover/movie>
 - **TC2_31_Minh_Post_AddFavoriteMovie:**
<https://api.themoviedb.org/3/account/21913469/favorite>
 - **TC3_31_Minh_Delete_RatingMovie:**
<https://api.themoviedb.org/3/movie/1165067/rating>





The screenshot shows the Postman interface with the following details:

- Collection:** APITestingTheMovieDatabase_31...
- Collection Name:** Movies_31_Minh
- Request:**
 - Method: GET
 - URL: http://Movies_31_Minh / TC1_31_Minh_Get_ListMovie
 - Description: TC1_31_Minh_Get_ListMovie
- Headers:** Headers tab is visible but empty.
- Body:** Body tab is visible but empty.
- Scripts:** Scripts tab is visible but empty.
- Settings:** Settings tab is visible but empty.
- Query Params:** A table with columns Key, Value, and Description. It has one row with 'Key' and 'Value' columns.
- Response:** Placeholder text: "Enter the URL and click Send to get a response".
- Bottom Bar:** Includes buttons for Online, Find and replace, Console, Postbot, Runner, Start Proxy, Cookies, Vault, Trash, and Help.

The screenshot shows the Postman interface with the following details:

- Collection:** APITestingTheMovieDatabase_31...
- Collection Name:** Movies_31_Minh
- Requests:**
 - GET TC1_31_Minh_Get_ListMovie
 - POST TC2_31_Minh_Post_AddFavoriteMovie
 - DEL TC3_31_Minh_Delete_RatingMovie
- Headers:** Headers tab is visible but empty.
- Body:** Body tab is visible but empty.
- Scripts:** Scripts tab is visible but empty.
- Settings:** Settings tab is visible but empty.
- Query Params:** A table with columns Key, Value, and Description. It has one row with 'Key' and 'Value' columns.
- Response:** Placeholder text: "Enter the URL and click Send to get a response".
- Bottom Bar:** Includes buttons for Online, Find and replace, Console, Postbot, Runner, Start Proxy, Cookies, Vault, Trash, and Help.

Hình 3.3. Các bước tạo một Blank collection và các request API (31_Minh)

The screenshot shows the TMDB API Settings page. On the left, there is a sidebar with the following menu items:

- Edit Profile
- Account Settings
- Streaming Services
- Notification Settings
- Blocked Users
- Import List
- Sharing Settings
- Sessions
- API** (highlighted with a red box)
- Delete Account

The main content area has the following sections:

- API Overview**: A brief introduction to the API service.
- Documentation**: A link to the developer documentation.
- Support**: A link to the support forums.
- API Details**: A link to test API credentials.
- API Read Access Token**: A text box containing a long API token, which is highlighted with a red box.
- API Key**: A text box containing a short API key.

The screenshot shows the Postman API client interface. On the left, the sidebar shows the following environments and flows:

- Collections
- Environments
- Flows
- History

The main workspace shows a collection named "API Testing The Movie Database_31...". Under this collection, there are three requests:

- GET TC1_31_Minh_Get_ListMovie**: The request highlighted with a red box.
- POST TC2_31_Minh_Post_AddFavoriteMovie**
- DEL TC3_31_Minh_Delete_RatingMovie**

The "Authorization" tab is selected for the "GET TC1_31_Minh_Get_ListMovie" request. The "Key" field, which contains a Bearer token, is highlighted with a red box.

Hình 3.4. Các bước lấy token để xác thực người dùng (31_Minh)

TMDB API Reference

Discover

Movie

https://api.themoviedb.org/3/discover/movie

Find movies using over 30 filters and sort options.

RECENT REQUESTS

TIME STATUS USER AGENT

Make a request to see history.

0 Requests This Month SEE ALL REQUESTS →

Advanced Filtering

If you specify the region parameter, the regional release date will be used instead of the primary release date. The date returned will be the first date based on your query (i.e. if a `with_release_type` is specified). It's important to note the order of the release types that are used. Specifying `2|3` would return the limited theatrical release date as opposed to `3|2` which would return the theatrical date.

AND / OR Logic

Also note that a number of filters support being comma (,) or pipe (|) separated.

REQUEST

```

1 CURL *hnd = curl_easy_init();
2
3 curl_easy_setopt(hnd, CURLOPT_CUSTOMREQUEST, "GET");
4 curl_easy_setopt(hnd, CURLOPT_WRITEFUNCTION, stdout);
5 curl_easy_setopt(hnd, CURLOPT_URL, "https://api.themoviedb.org/3/discover/movie");
6
7 struct curl_slist *headers = NULL;
8 headers = curl_slist_append(headers, "Accept: application/json");
9 headers = curl_slist_append(headers, "Authorization: Bearer");
10 curl_easy_setopt(hnd, CURLOPT_HTTPHEADER, headers);
11
12 CURLcode ret = curl_easy_perform(hnd);

```

Try It!

POSTMAN

GET TC1_31_Minh_Get_ListMovie

https://api.themoviedb.org/3/discover/movie

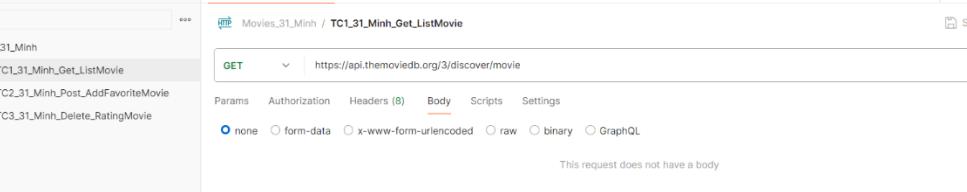
Send

Params Authorization Headers (8) Body Scripts Settings

none

Response History

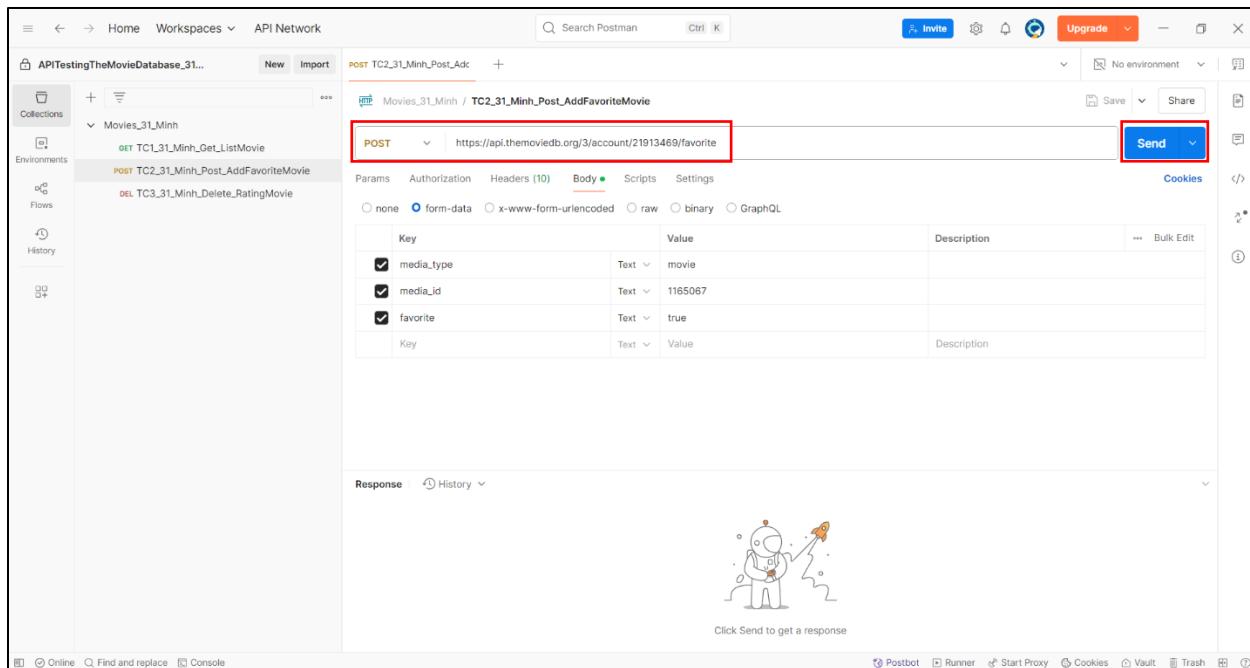
Click Send to get a response



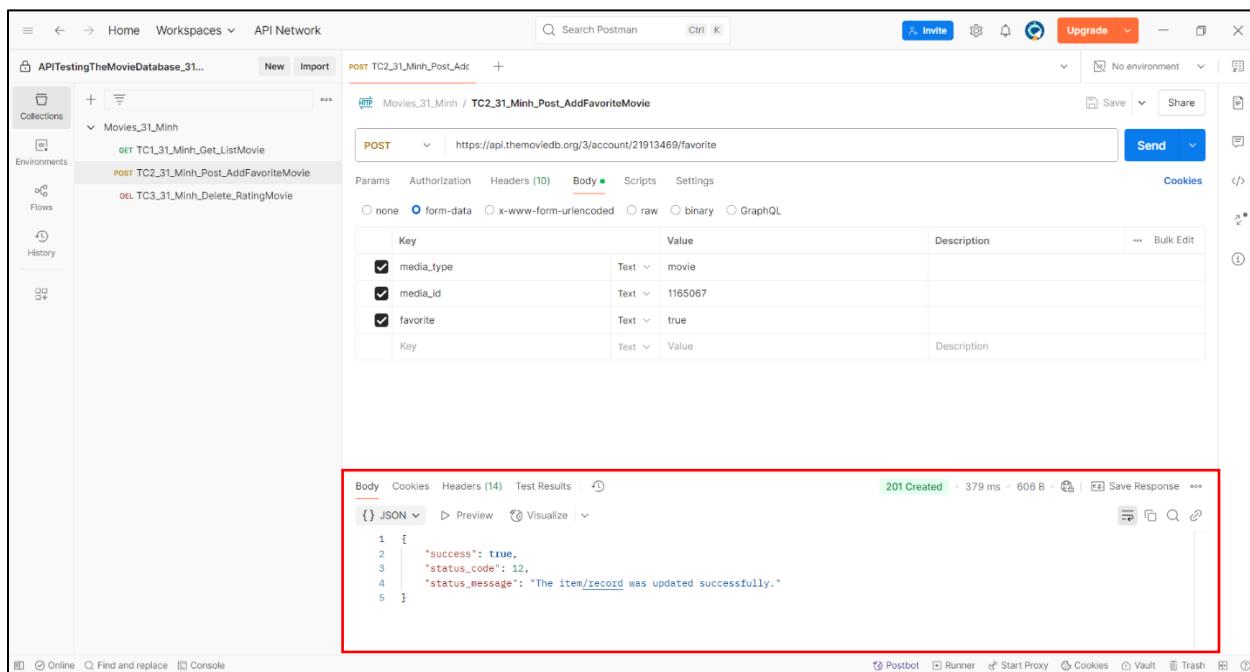
The screenshot shows the Postman application interface. The left sidebar contains 'Collections' (APITestingTheMovieDatabase_31...), 'Environments' (Movies_31_Minh), and 'Flows' (History). The main area shows a collection named 'Movies_31_Minh' with three items: 'GET TC1_31_Minh_Get_ListMovie', 'POST TC2_31_Minh_Post_AddFavoriteMovie', and 'DEL TC3_31_Minh_Delete_RatingMovie'. The 'GET TC1_31_Minh_Get_ListMovie' item is selected. The request details show a GET request to <https://api.themoviedb.org/3/discover/movie>. The 'Body' tab is selected, showing 'none' as the content type. The response body is displayed in JSON format, showing a list of movies. The response status is 200 OK with a duration of 142 ms and a size of 5.74 KB. The response body content is as follows:

```
1  [
2    {
3      "page": 1,
4      "results": [
5        {
6          "adult": false,
7          "backdrop_path": "/m2e2lsJjE3UAqeUB5fLUkpg4Ie.jpg",
8          "genre_ids": [
9            53,
10           878
11         ],
12         "id": 1165667,
13         "original_language": "en",
14         "original_title": "Cosmic Chaos",
15         "overview": "Battles in virtual reality, survival in a post-apocalyptic wasteland, a Soviet spaceship giving a distress signal - Fantastic stories created with advanced special effects and passion.",
16         "popularity": 639.5047,
17         "poster_path": "/mC1zqv7eBngff7Xp49Envoyx1v.jpg"
18       }
19     ]
20   ]
```

The screenshot shows the TMDb API Reference for the 'Add Favorite' endpoint. The URL is https://api.themoviedb.org/3/account/{account_id}/favorite. The page includes a 'RECENT REQUESTS' table, 'PATH PARAMS' (account_id: 21913469), 'QUERY PARAMS' (session_id: string), and 'BODY PARAMS'. On the right, there are sections for 'LANGUAGE' (Shell, Node.js, Ruby, PHP, C++) and 'CREDENTIALS' (Header, Access Token). A 'REQUEST' code block shows curl examples, and a 'Try It!' button is at the bottom right.



The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (Movies_31_Minh), 'Environments', 'Flows', and 'History'. The main area shows a 'POST' request to 'TC2_31_Minh_Post_AddFavoriteMovie' with the URL 'https://api.themoviedb.org/3/account/21913469/favorite'. The 'Body' tab is selected, showing a form-data structure with three fields: 'media_type' (value: 'movie'), 'media_id' (value: '1165067'), and 'favorite' (value: 'true'). A red box highlights the 'Send' button. Below the request, there's a 'Response' section with a small illustration of an astronaut and a red box around the text 'Click Send to get a response'.



This screenshot shows the same Postman interface after the request has been sent. The 'Body' tab is highlighted, showing the JSON response:

```
{ "success": true, "status_code": 12, "status_message": "The item/record was updated successfully." }
```

 A red box highlights this JSON response. The status bar at the bottom indicates '201 Created'.

TMDB API Reference

Delete Rating

DELETE https://api.themoviedb.org/3/movie/{movie_id}/rating

Delete a user rating.

RECENT REQUESTS

TIME STATUS USER AGENT

Make a request to see history.

0 Requests This Month SEE ALL REQUESTS →

PATH PARAMS

movie_id int32 required

QUERY PARAMS

guest_session_id string

session_id string

LANGUAGE: Shell, Node, Ruby, PHP, C++, ...

CREDENTIALS: Header eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyZT...

REQUEST:

```

1 CURL *hnd = curl_easy_init();
2
3 curl_easy_setopt(hnd, CURLOPT_CUSTOMREQUEST, "DELETE");
4 curl_easy_setopt(hnd, CURLOPT_WRITEOUT, stdout);
5 curl_easy_setopt(hnd, CURLOPT_URL, "https://api.themoviedb.org/3/movie/1165067/rating");
6
7 struct curl_slist *headers = NULL;
8 headers = curl_slist_append(headers, "accept: application/json");
9 headers = curl_slist_append(headers, "Content-type: application/json");
10 headers = curl_slist_append(headers, "Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyZTNIYQ...");
```

Try It!

Postman

API Testing The Movie Database 31...

DELETE TC3_31_Minh_Delete_RatingMovie

https://api.themoviedb.org/3/movie/1165067/rating

Send

Params Authorization Headers (8) Body Scripts Settings

Key	Value	Description
Cache-Control	no-cache	
Postman-Token	<calculated when request is sent>	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.43.3	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Authorization	Bearer eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyZTNIYQ...;	

Response History

Click Send to get a response

Hình 3.5. Các bước lấy API Reference và kết quả sau khi test API (31_Minh)

3.1.3. Viết các API Test Case:

- Chọn Scripts > Post-response > Viết code test case cho API request > Nhấn chuột phải vào **Movies_31_Min** > Run collection > Run **Movies_31_Min**
 - **TC1_31_Min_Get_ListMovie:**

```
//TC1_31_Min: Kiểm tra trạng thái mã HTTP trả về là 200 OK
pm.test("TC1_31_Min_ResponseStatusCodeIs200OK", function () {
    pm.response.to.have.status(200);
});

//TC2_31_Min: Kiểm tra trả về có đúng là JSON không, sử dụng utf-8 để mã hóa ký tự
pm.test("TC2_31_Min_ResponseIsInJSONFormat", function () {
    pm.response.to.have.header("Content-Type", "application/json; charset=utf-8");
});

//TC3_31_Min: Kiểm tra đường dẫn ảnh có đúng định dạng .jpg
pm.test("TC3_31_Min_ImageURLsHaveTheCorrectExtension", function () {
    var jsonData_31_Minh = pm.response.json();
    jsonData_31_Minh.results.forEach(function (movie_31_Minh) {
        var posterPath_31_Minh = movie_31_Minh.poster_path;
        pm.expect(posterPath_31_Minh).to.match(/\.\jpg$/i);
    });
});
```

```

        });
    });

//TC4_31_Minh: Kiểm tra có trả về các thông tin, bao gồm: id, title, overview,
release_date, vote_average, vote_count.
pm.test("TC4_31_Minh_ResponseContainsMovieInformation", function () {
    var jsonData_31_Minh = pm.response.json();

    pm.expect(jsonData_31_Minh.results).to.be.an("array");
    pm.expect(jsonData_31_Minh).to.have.property("page");
    pm.expect(jsonData_31_Minh).to.have.property("results");
    pm.expect(jsonData_31_Minh).to.have.property("total_pages");
    pm.expect(jsonData_31_Minh).to.have.property("total_results");
});

//TC5_31_Minh: Kiểm tra mỗi bộ phim có các thuộc tính: id, title, overview,
release_date, vote_average, vote_count.
pm.test("TC5_31_Minh_CheckProperty", function () {
    var jsonData_31_Minh = pm.response.json();
    var propertiesToCheck_31_Minh = ["id", "title", "overview",
"release_date", "vote_average", "vote_count"];

    jsonData_31_Minh.results.map(function (movie) {
        propertiesToCheck_31_Minh.forEach(function (property) {
            pm.expect(movie).to.have.property(property);
        });
    });
});

//TC6_31_Minh: Kiểm tra thuộc tính adult của bộ phim đầu tiên trong danh sách
có giá trị là false.
pm.test("TC6_31_Minh_CheckValueOfAdultForTheFirstMovie", function () {
    var jsonData_31_Minh = pm.response.json();
    pm.expect(jsonData_31_Minh.results[0].adult).to.be.false;
})

```

- **TC2_31_Minh_Post_AddFavoriteMovie:**

```

//TC7_31_Minh: Kiểm tra trạng thái mã HTTP trả về là 201 Created
pm.test("TC7_31_Minh_ResponseStatusCodeIs201OK", function () {
    pm.response.to.have.status(201);
});

//TC8_31_Minh: Kiểm tra xem dữ liệu gửi lên có đúng không?
pm.test("TC8_31_Minh_CheckFormData", function(){
    var formData_31_Minh = request.data;
    pm.expect(formData_31_Minh.hasOwnProperty('media_type')).to.be.true;
    pm.expect(formData_31_Minh.hasOwnProperty('media_id')).to.be.true;
})

```

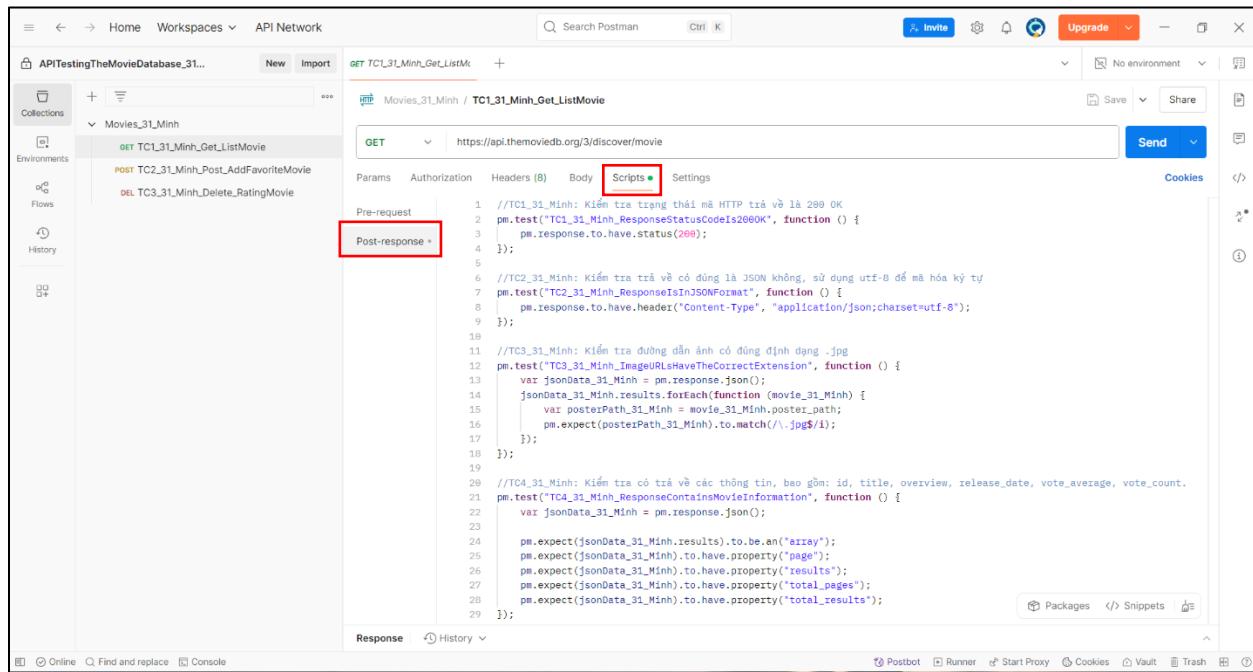
```
pm.expect(formData_31_Minh.hasOwnProperty('favorite')).to.be.true;
});
```

- **TC3_31_Minh_Delete_RatingMovie:**

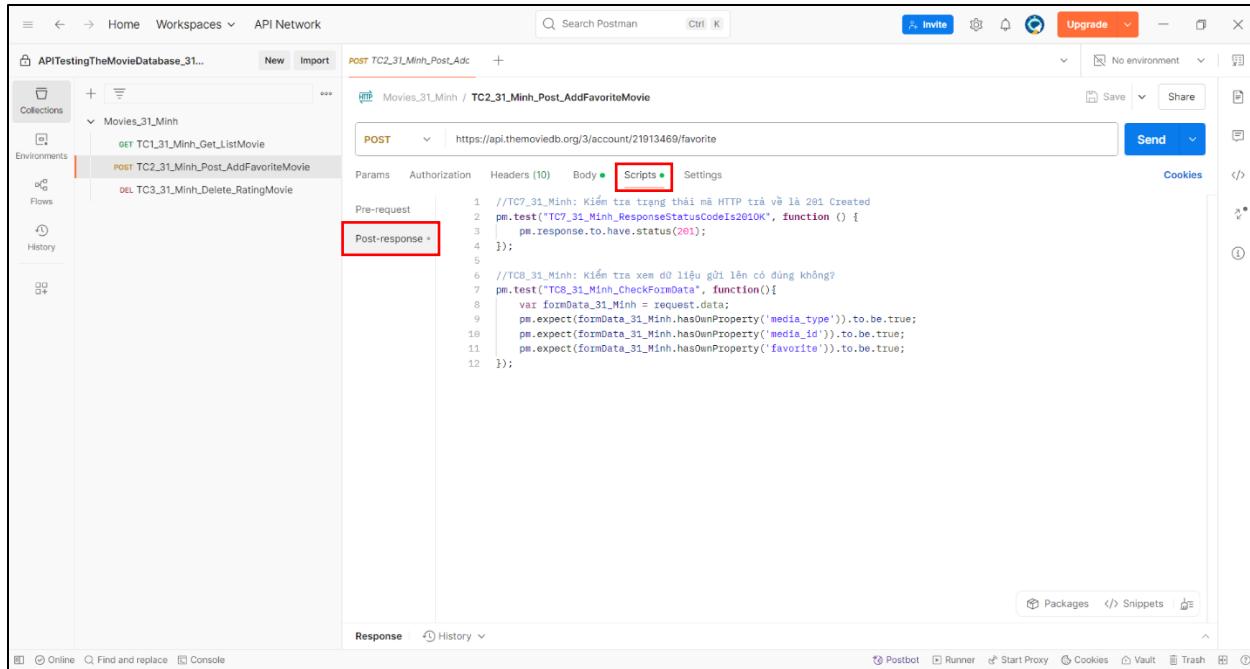
```
//TC9_31_Minh: Kiểm tra trạng thái mã HTTP trả về là 200 OK
pm.test("TC9_31_Minh_StatusCodeIs200", function () {
    pm.response.to.have.status(200);
});

//TC10_31_Minh: Kiểm tra trạng thái 'success' trong response body trả về là
boolean và bằng true
pm.test("TC10_31_Minh_CheckSuccessIsBoolean", function () {
    const resData_31_Minh = pm.response.json();
    pm.expect(resData_31_Minh.success).to.be.a("boolean");
    pm.expect(resData_31_Minh.success).to.be.true;
});

//TC11_31_Minh: Kiểm tra 'status_message' trong response body trả về có chứa chuỗi
"The item/record was deleted successfully." không?
pm.test("TC11_31_Minh_Check'status_message'ContainsExpectedMessage", function () {
    var jsonData_31_Minh = pm.response.json();
    pm.expect(jsonData_31_Minh.status_message).to.include("The item/record was
deleted successfully.");
});
```



```
1 //TC1_31_Minh: Kiểm tra trạng thái mã HTTP trả về là 200 OK
2 pm.test("TC1_31_Minh_ResponseStatusCodesIs200OK", function () {
3     pm.response.to.have.status(200);
4 });
5
6 //TC2_31_Minh: Kiểm tra trả về có đúng là JSON không, sử dụng utf-8 để mã hóa ký tự
7 pm.test("TC2_31_Minh_ResponseIsInJSONFormat", function () {
8     pm.response.to.have.header("Content-Type", "application/json;charset=utf-8");
9 });
10
11 //TC3_31_Minh: Kiểm tra đường dẫn ảnh có đúng định dạng .jpg
12 pm.test("TC3_31_Minh_ImageURLsHaveTheCorrectExtension", function () {
13     var jsonData_31_Minh = pm.response.json();
14     jsonData_31_Minh.results.forEach(function (movie_31_Minh) {
15         var posterPath_31_Minh = movie_31_Minh.poster_path;
16         pm.expect(posterPath_31_Minh).to.match(/\.\.jpg$/i);
17     });
18 });
19
20 //TC4_31_Minh: Kiểm tra có trả về các thông tin, bao gồm: id, title, overview, release_date, vote_average, vote_count.
21 pm.test("TC4_31_Minh_ResponseContainsMovieInformation", function () {
22     var jsonData_31_Minh = pm.response.json();
23
24     pm.expect(jsonData_31_Minh.results).to.be.an("array");
25     pm.expect(jsonData_31_Minh).to.have.property("page");
26     pm.expect(jsonData_31_Minh).to.have.property("results");
27     pm.expect(jsonData_31_Minh).to.have.property("total_pages");
28     pm.expect(jsonData_31_Minh).to.have.property("total_results");
29 });
```



POST TC2_31_Minh_Post_Adv

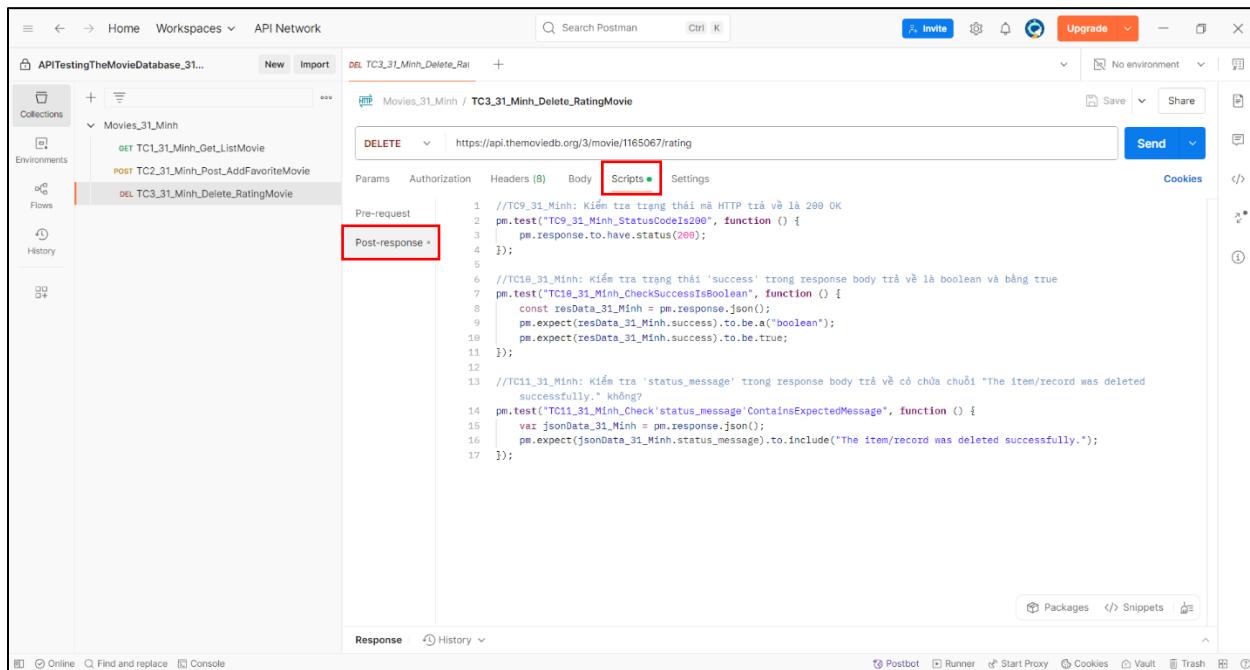
POST https://api.themoviedb.org/3/account/21913469/favorite

Params Authorization Headers (10) Body Scripts Settings

```

1 //TC7_31_Minh: Kiểm tra trạng thái mã HTTP trả về là 201 Created
2 pm.test("TC7_31_Minh_ResponseStatusCodeIs201OK", function () {
3   pm.response.to.have.status(201);
4 });
5
6 //TC8_31_Minh: Kiểm tra xem dữ liệu gửi lên có đúng không?
7 pm.test("TC8_31_Minh_CheckFormData", function(){
8   var formData_31_Minh = request.data;
9   pm.expect(formData_31_Minh.hasOwnProperty('media_type')).to.be.true;
10  pm.expect(formData_31_Minh.hasOwnProperty('media_id')).to.be.true;
11  pm.expect(formData_31_Minh.hasOwnProperty('favorite')).to.be.true;
12 });

```



DELETE TC3_31_Minh_Delete_Rai

DELETE https://api.themoviedb.org/3/movie/1165067/rating

Params Authorization Headers (8) Body Scripts Settings

```

1 //TC9_31_Minh: Kiểm tra trạng thái mã HTTP trả về là 200 OK
2 pm.test("TC9_31_Minh_StatusCodeIs200", function () {
3   pm.response.to.have.status(200);
4 });
5
6 //TC10_31_Minh: Kiểm tra trang thái 'success' trong response body trả về là boolean và bằng true
7 pm.test("TC10_31_Minh_CheckSuccessIsBoolean", function () {
8   const resData_31_Minh = pm.response.json();
9   pm.expect(resData_31_Minh.success).to.be.a("boolean");
10  pm.expect(resData_31_Minh.success).to.be.true;
11 });
12
13 //TC11_31_Minh: Kiểm tra 'status_message' trong response body trả về có chứa chuỗi "The item/record was deleted successfully." không?
14 pm.test("TC11_31_Minh_CheckStatusMessageContainsExpectedMessage", function () {
15   var jsonData_31_Minh = pm.response.json();
16   pm.expect(jsonData_31_Minh.status_message).to.include("The item/record was deleted successfully.");
17 });

```

API Testing The Movie Database_31...

GET TC1_31 | Share | Move | Run collection | Generate tests | Edit | Add request | Add folder | Monitor collection | Mock collection | Create a fork | Create pull request | Merge changes | Pull changes | View changelog | View documentation | Rename | Duplicate | Export | Paste |

Send all requests in this collection

```

//TC1_31_Minh: Kiểm tra trạng thái mã HTTP trả về là 200 OK
pm.test("TC1_31_Minh_ResponseStatusIs200OK", function () {
  pm.response.to.have.status(200);
});

//TC2_31_Minh: Kiểm tra trả về có đúng là JSON không, sử dụng utf-8 để mã hóa ký tự
pm.test("TC2_31_Minh_ResponseIsInJSONFormat", function () {
  pm.response.to.have.header("Content-Type", "application/json;charset=utf-8");
});

//TC3_31_Minh: Kiểm tra đường dẫn ảnh có đúng định dạng .jpg
pm.test("TC3_31_Minh_ImageURLHaveTheCorrectExtension", function () {
  var jsonData_31_Minh = pm.response.json();
  jsonData_31_Minh.results.forEach(function (movie_31_Minh) {
    var posterPath_31_Minh = movie_31_Minh.poster_path;
    pm.expect(posterPath_31_Minh).to.match(/\.\jpg$/i);
  });
});

//TC4_31_Minh: Kiểm tra có trả về các thông tin, bao gồm: id, title, overview, release_date, vote_average, vote_count.
pm.test("TC4_31_Minh_ResponseContainsMovieInformation", function () {
  var jsonData_31_Minh = pm.response.json();
  pm.expect(jsonData_31_Minh.results).to.be.an("array");
  pm.expect(jsonData_31_Minh).to.have.property("page");
  pm.expect(jsonData_31_Minh).to.have.property("results");
  pm.expect(jsonData_31_Minh).to.have.property("total_pages");
  pm.expect(jsonData_31_Minh).to.have.property("total_results");
});

```

Response | History | Postbot | Runner | Start Proxy | Cookies | Vault | Trash | ↻

API Testing The Movie Database_31...

GET TC1_31_Minh_Get_ListMovie | POST TC2_31_Minh_Post_AddFavoriteMovie | DEL TC3_31_Minh_Delete_RatingMovie

Run Sequence | Deselect All | Select All | Reset | Functional | Performance

Choose how to run your collection

- Run manually
Run this collection in the Collection Runner.
- Schedule runs
Periodically run collection at a specified time on the Postman Cloud.
- Automate runs via CLI
Configure CLI command to run on your build pipeline.

Run configuration

Iterations: 1

Delay: 0 ms

Test data file: Only JSON and CSV files are accepted.
Select File

Advanced settings

Run Movies_31_Minh

25/25 runs used. Get unlimited runs with add-on.
Learn more ↗

Online | Find and replace | Console | Postbot | Runner | Start Proxy | Cookies | Vault | Trash | ↻

Test Case	Request Type	URL	Status	Duration	Response Size
TC1_31_Minh_Get_ListMovie	GET	https://api.themoviedb.org/3/discover/movie	200 OK	133 ms	5.873 KB
TC2_31_Minh_Post_AddFavoriteMovie	POST	https://api.themoviedb.org/3/account/21913469/favorite	201 Created	447 ms	606 B
TC3_31_Minh_Delete_RatingMovie	DELETE	https://api.themoviedb.org/3/movie/1165087/rating	200 OK	297 ms	635 B

Hình 3.6. Các bước thực hiện API Test Case (31_Minh)

3.2. TẠO API BẰNG NODE.JS

3.2.1. Tạo API bằng Node.js:

- Bước 1: Tạo Folder **MyScript** > Mở Terminal > Kiểm tra phiên bản “node -v” và “npm -v” > “npm init” > Nhấn enter > yes
- Bước 2: Mở Visual Studio Code > File > Open Folder... > Select Folder **MyScript**
- Bước 3: Tạo file .json có tên là **data_31_Minh.json** và thêm dữ liệu sau

```
{
  "users": [
    {
      "id": 1,
      "name": "Nguyễn Văn An",
      "email": "nguyenvanan@gmail.com",
      "phone": "+84901234567",
      "address": {
        "street": "12 Đường Quang Trung",
        "city": "Hồ Chí Minh",
        "country": "Việt Nam"
      },
      "skills": ["Java", "React Native", "C#", "Cloud Computing"],
      "projects": [
        {
          "name": "Project A"
        }
      ]
    }
  ]
}
```

```

        "name": "Social Media App",
        "description": "Ứng dụng mạng xã hội sử dụng React Native và Firebase.",
        "year": 2025
    },
    {
        "name": "Website Testing Automation",
        "description": "Dự án kiểm thử tự động cho một website bằng C#.",
        "year": 2024
    }
]
},
{
    "id": 2,
    "name": "Trần Thị Bình",
    "email": "tranthibinh@example.com",
    "phone": "+84987654321",
    "address": {
        "street": "37 Đường Lê Thị Liền",
        "city": "Hà Nội",
        "country": "Việt Nam"
    },
    "skills": ["Python", "Django", "Machine Learning"],
    "projects": [
        {
            "name": "AI Chatbot",
            "description": "Chatbot AI sử dụng Python và TensorFlow.",
            "year": 2024
        },
        {
            "name": "E-commerce Website",
            "description": "Trang web thương mại điện tử sử dụng Django.",
            "year": 2023
        }
    ]
},
{
    "id": 3,
    "name": "Lê Văn Cường",
    "email": "levancuong@example.com",
    "phone": "+84906543210",
    "address": {
        "street": "51 Đường Lê Văn Lương",
        "city": "Đà Nẵng",
        "country": "Việt Nam"
    },
    "skills": ["C++", "Embedded Systems", "IoT"],
    "projects": [
        {
            "name": "Smart Home System",
            "description": "Hệ thống nhà thông minh sử dụng IoT và Arduino.",
            "year": 2022
        },
        {
            "name": "Robot Control System",
            "description": "Hệ thống điều khiển robot dùng C++.",
            "year": 2025
        }
    ]
}

```

```

},
{
  "id": 4,
  "name": "Phạm Thị Dương",
  "email": "phamthiduong@example.com",
  "phone": "+84911223344",
  "address": {
    "street": "159 Đường Thống Nhất",
    "city": "Cần Thơ",
    "country": "Việt Nam"
  },
  "skills": ["HTML", "CSS", "JavaScript", "React"],
  "projects": [
    {
      "name": "Portfolio Website",
      "description": "Trang web cá nhân giới thiệu sản phẩm và kỹ năng.",
      "year": 2019
    },
    {
      "name": "Real-time Chat App",
      "description": "Ứng dụng chat thời gian thực với WebSocket.",
      "year": 2023
    }
  ]
},
{
  "id": 5,
  "name": "Hoàng Văn Hậu",
  "email": "hoangvanhau@example.com",
  "phone": "+84955667788",
  "address": {
    "street": "632 Đường Cách mạng Tháng 8",
    "city": "Nha Trang",
    "country": "Việt Nam"
  },
  "skills": ["Swift", "iOS Development", "UI/UX Design"],
  "projects": [
    {
      "name": "iOS Fitness App",
      "description": "Ứng dụng theo dõi sức khỏe và luyện tập trên iOS.",
      "year": 2021
    },
    {
      "name": "Task Manager App",
      "description": "Ứng dụng quản lý công việc trên iPhone.",
      "year": 2023
    }
  ]
},
{
  "id": 6,
  "name": "Võ Minh Phương",
  "email": "vominhphuong@example.com",
  "phone": "+84966778899",
  "address": {
    "street": "42 Đường Lê Thị Kính",
    "city": "Huế",
    "country": "Việt Nam"
  }
}

```

```

},
"skills": ["PHP", "Laravel", "MySQL"],
"projects": [
  {
    "name": "Job Portal",
    "description": "Website tuyển dụng sử dụng Laravel.",
    "year": 2018
  },
  {
    "name": "Online Learning Platform",
    "description": "Nền tảng học trực tuyến với PHP.",
    "year": 2021
  }
]
},
{
  "id": 7,
  "name": "Đinh Thảo Giang",
  "email": "dinhthaogiang@example.com",
  "phone": "+84977889900",
  "address": {
    "street": "71 Đường Huyền Trần Công Chúa",
    "city": "Vũng Tàu",
    "country": "Việt Nam"
  },
  "skills": ["Kotlin", "Android Development", "Firebase"],
  "projects": [
    {
      "name": "Android Shopping App",
      "description": "Ứng dụng mua sắm trên Android sử dụng Kotlin.",
      "year": 2022
    },
    {
      "name": "News Aggregator",
      "description": "Ứng dụng tổng hợp tin tức trên Android.",
      "year": 2025
    }
  ]
},
{
  "id": 8,
  "name": "Bùi Thành Hương",
  "email": "buithanhhuong@example.com",
  "phone": "+84988990011",
  "address": {
    "street": "369 Đường Nguyễn Văn Linh",
    "city": "Hải Phòng",
    "country": "Việt Nam"
  },
  "skills": ["Go", "Microservices", "Docker"],
  "projects": [
    {
      "name": "Cloud Storage Service",
      "description": "Dịch vụ lưu trữ đám mây sử dụng Go.",
      "year": 2020
    },
    {
      "name": "Log Monitoring System",
      "description": "Hệ thống giám sát log."
    }
  ]
}

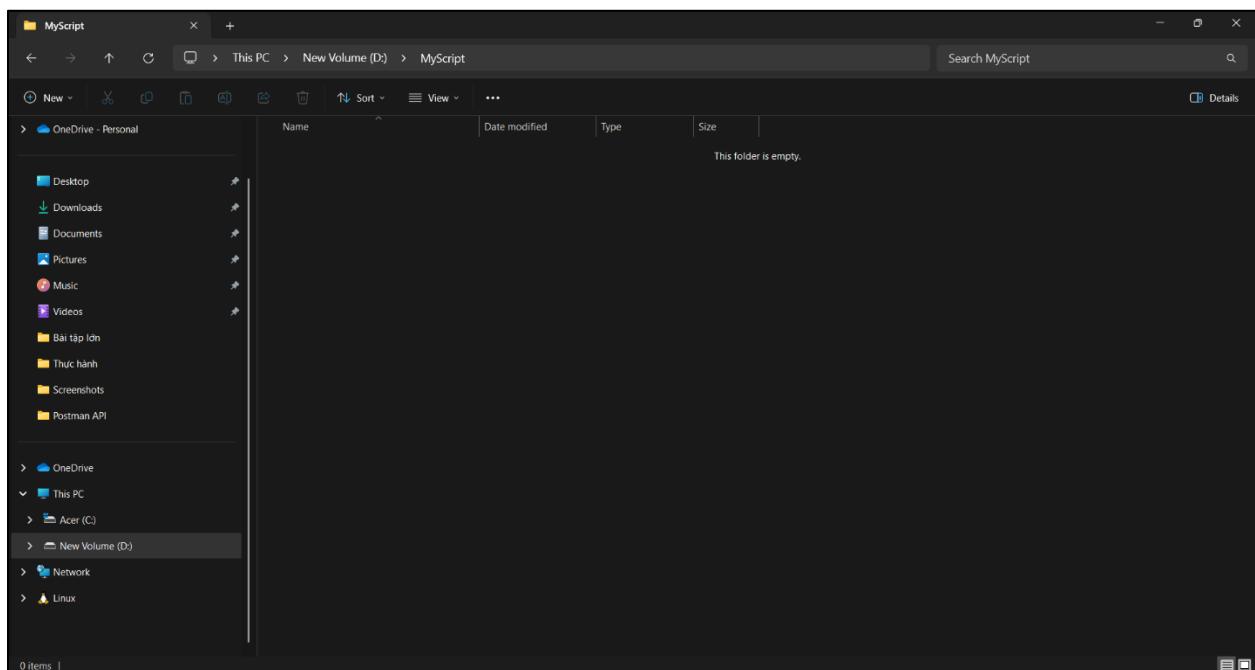
```

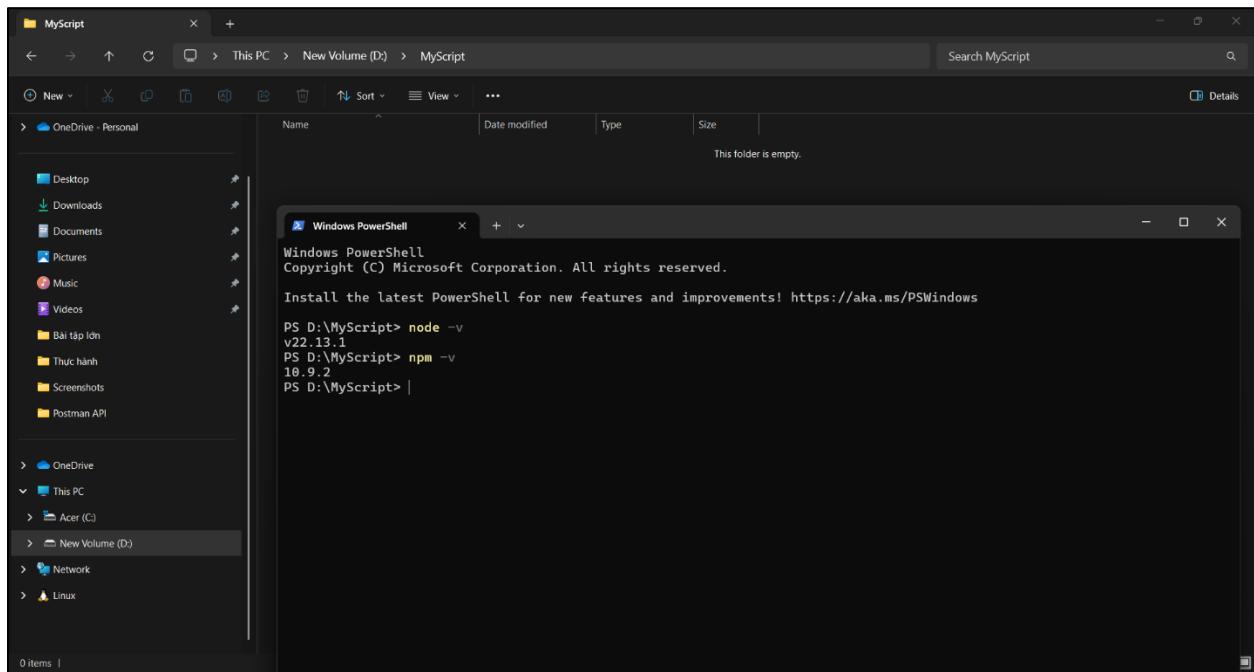
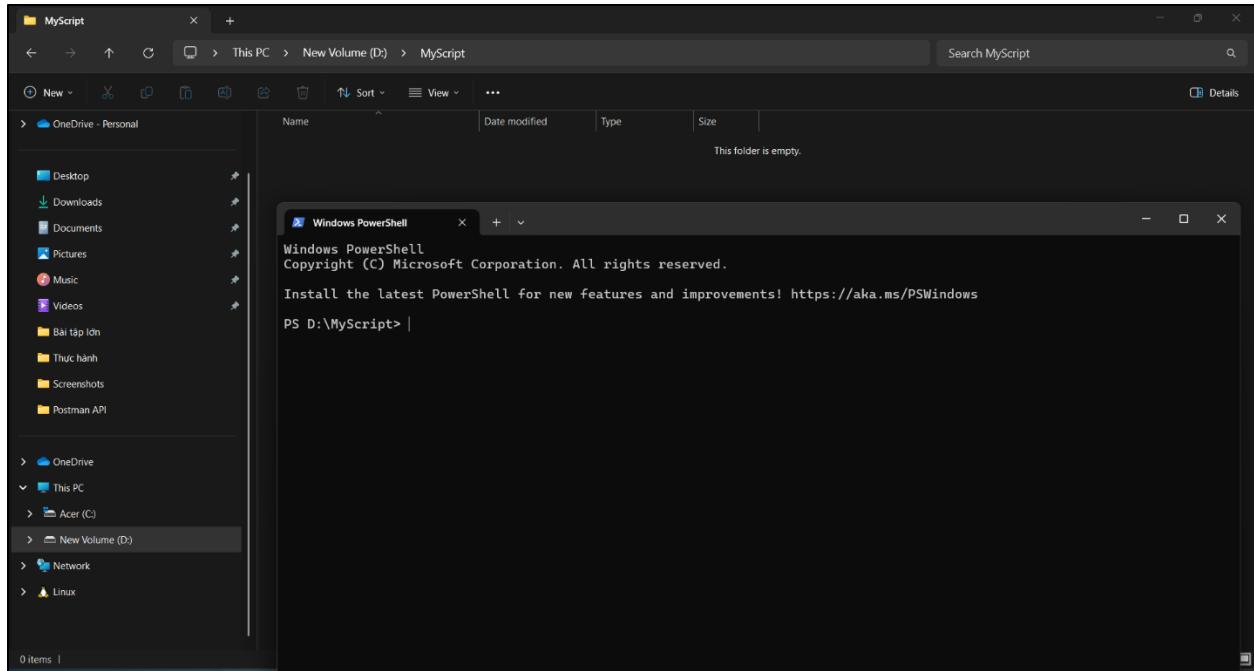
```
        "description": "Hệ thống giám sát log với Microservices.",  
        "year": 2023  
    }  
}  
}  
}
```

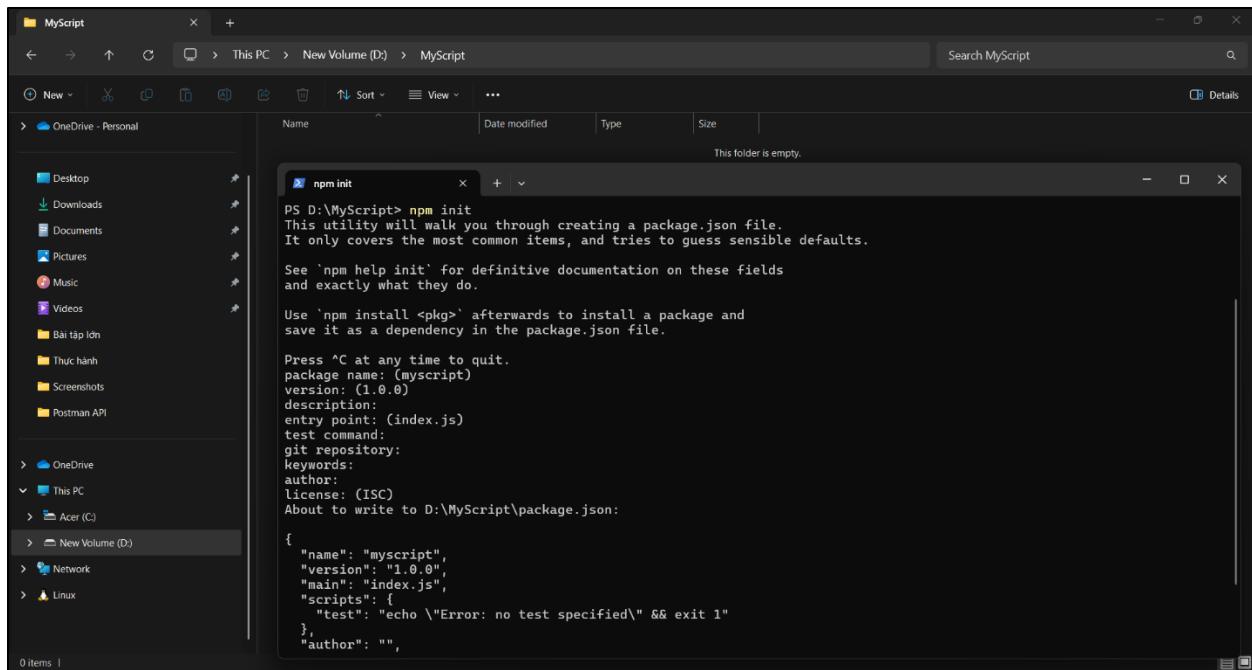
- Bước 4: Mở file **package.json** và thêm dòng sau vào

```
"start": "json-server --watch data_31_Minh.json",
```

- Bước 5: Mở Terminal > Gõ “npm start” ta được API <http://localhost:3000/users/>





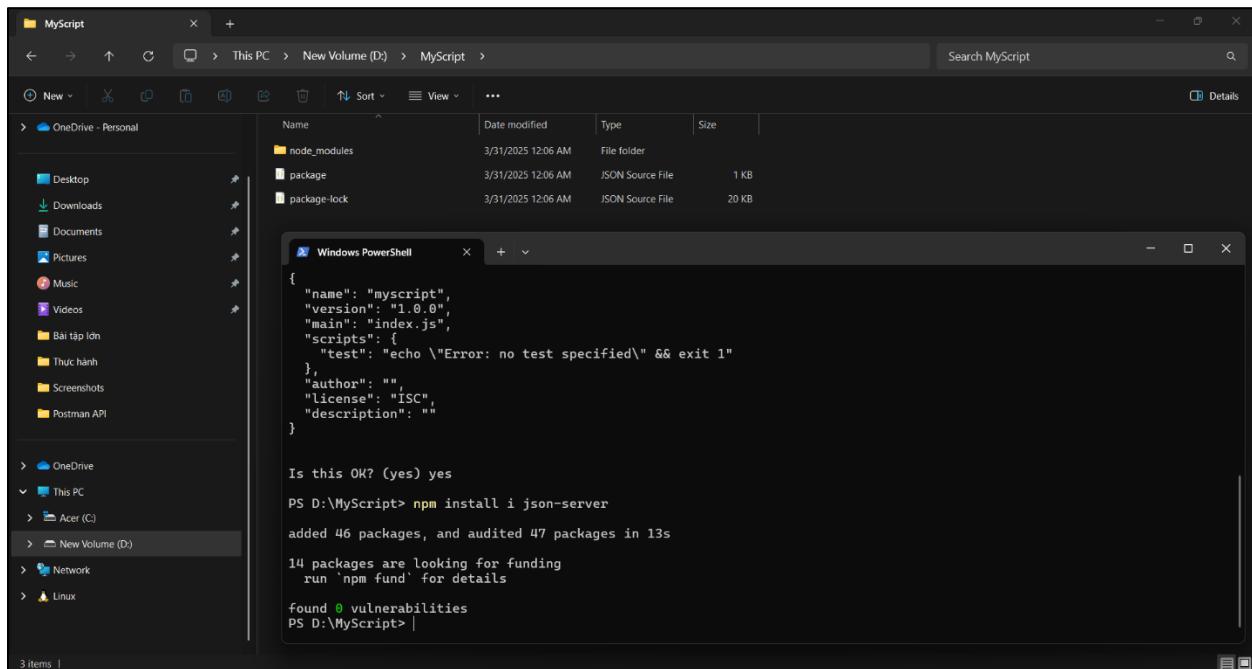


```
PS D:\MyScript> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help init' for definitive documentation on these fields
and exactly what they do.

Press 'C' at any time to quit.
package name: (myscript)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to D:\MyScript\package.json:

{
  "name": "myscript",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "",
```

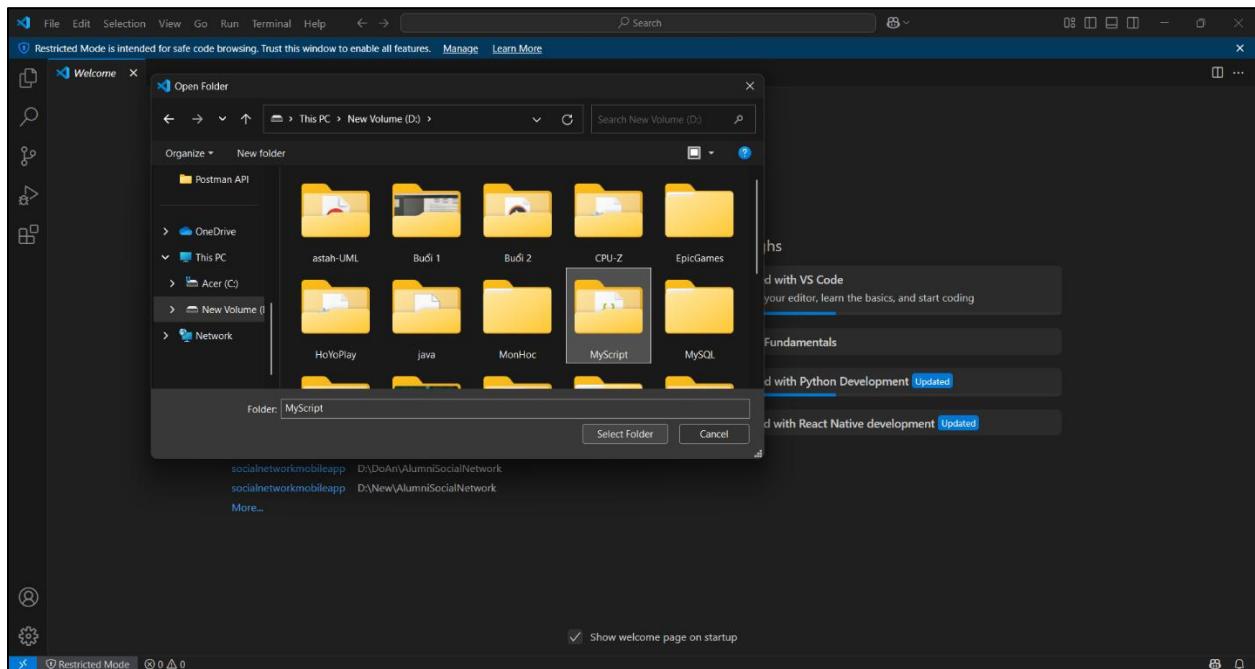
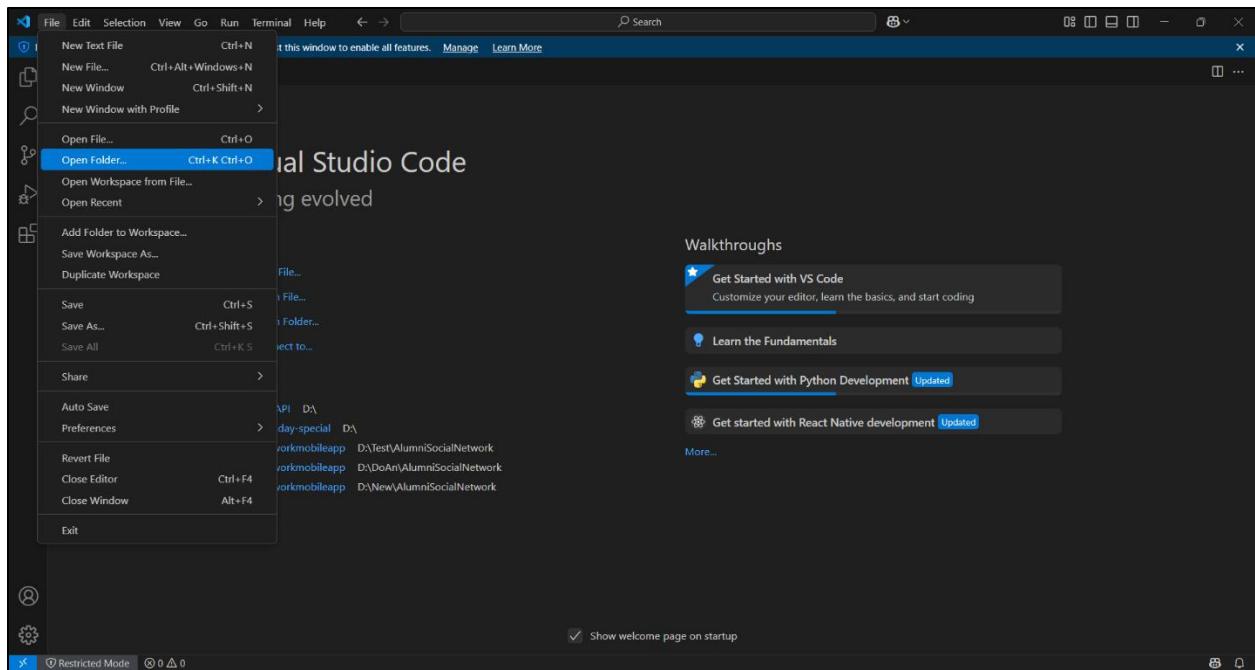


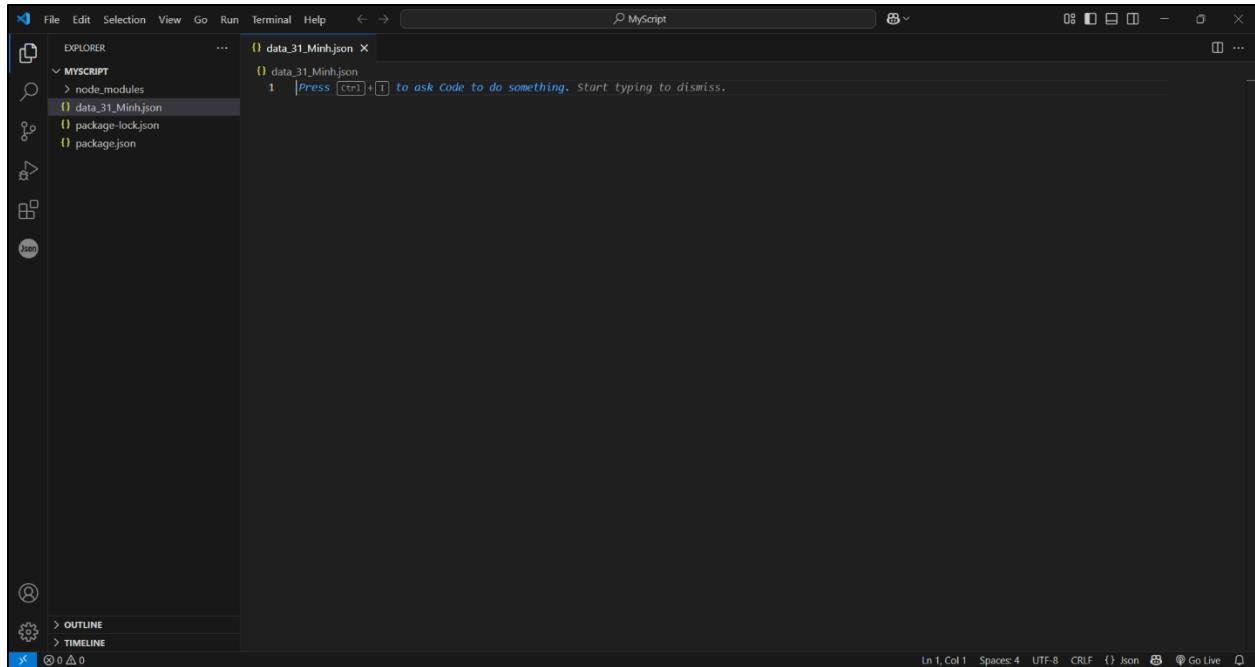
```
PS D:\MyScript> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help init' for definitive documentation on these fields
and exactly what they do.

Press 'C' at any time to quit.
package name: (myscript)
version: (1.0.0)
main: (index.js)
scripts: {
  "test": "echo \\\"Error: no test specified\\\" && exit 1"
},
author: "",
license: (ISC)
description: ""

Is this OK? (yes) yes
PS D:\MyScript> npm install i json-server
added 46 packages, and audited 47 packages in 13s
14 packages are looking for funding
  run 'npm fund' for details
found 0 vulnerabilities
PS D:\MyScript> |
```

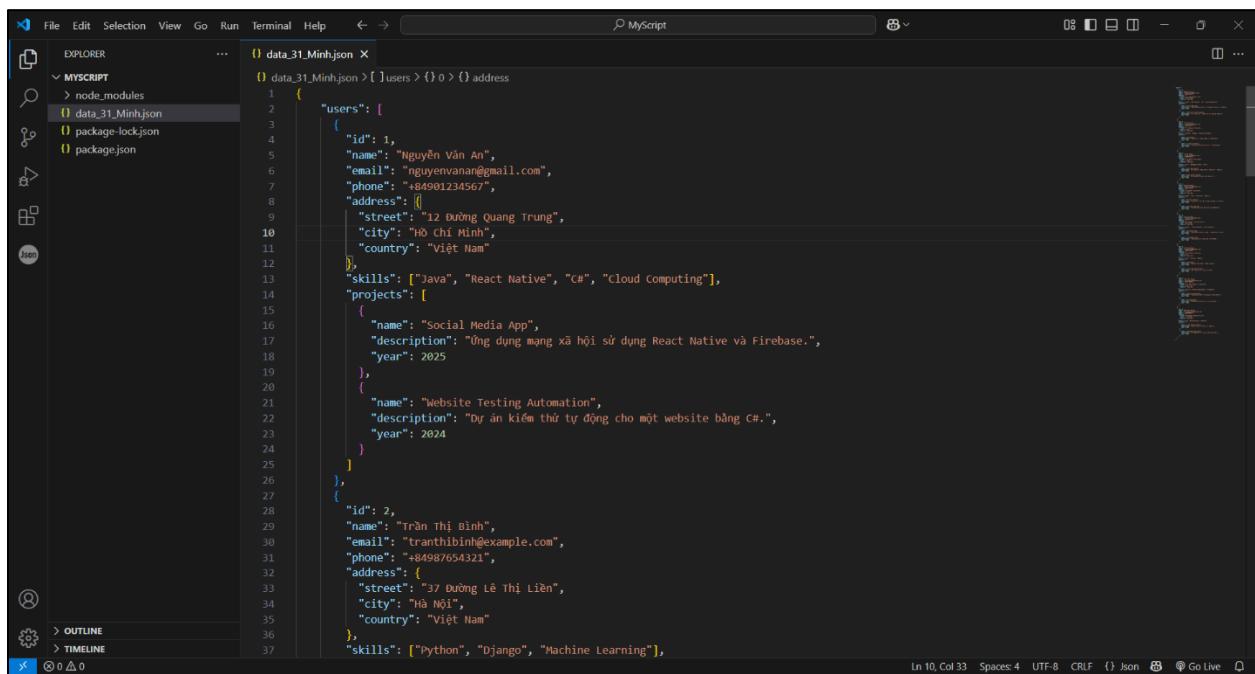




```

{
  "users": [
    {
      "id": 1,
      "name": "Nguyễn Văn An",
      "email": "nguyenvanan@gmail.com",
      "phone": "+84901234567",
      "address": {
        "street": "12 Đường Quang Trung",
        "city": "Hồ Chí Minh",
        "country": "Việt Nam"
      },
      "skills": ["Java", "React Native", "C#", "Cloud Computing"],
      "projects": [
        {
          "name": "Social Media App",
          "description": "Đã áp dụng mạng xã hội sử dụng React Native và Firebase.",
          "year": 2025
        },
        {
          "name": "Website Testing Automation",
          "description": "Đã áp dụng thử tự động cho một website bằng C#.",
          "year": 2024
        }
      ]
    },
    {
      "id": 2,
      "name": "Trần Thị Bình",
      "email": "tranthibinh@example.com",
      "phone": "+84987654321",
      "address": {
        "street": "37 Đường Lê Thị Liền",
        "city": "Hà Nội",
        "country": "Việt Nam"
      },
      "skills": ["Python", "Django", "Machine Learning"]
    }
  ]
}

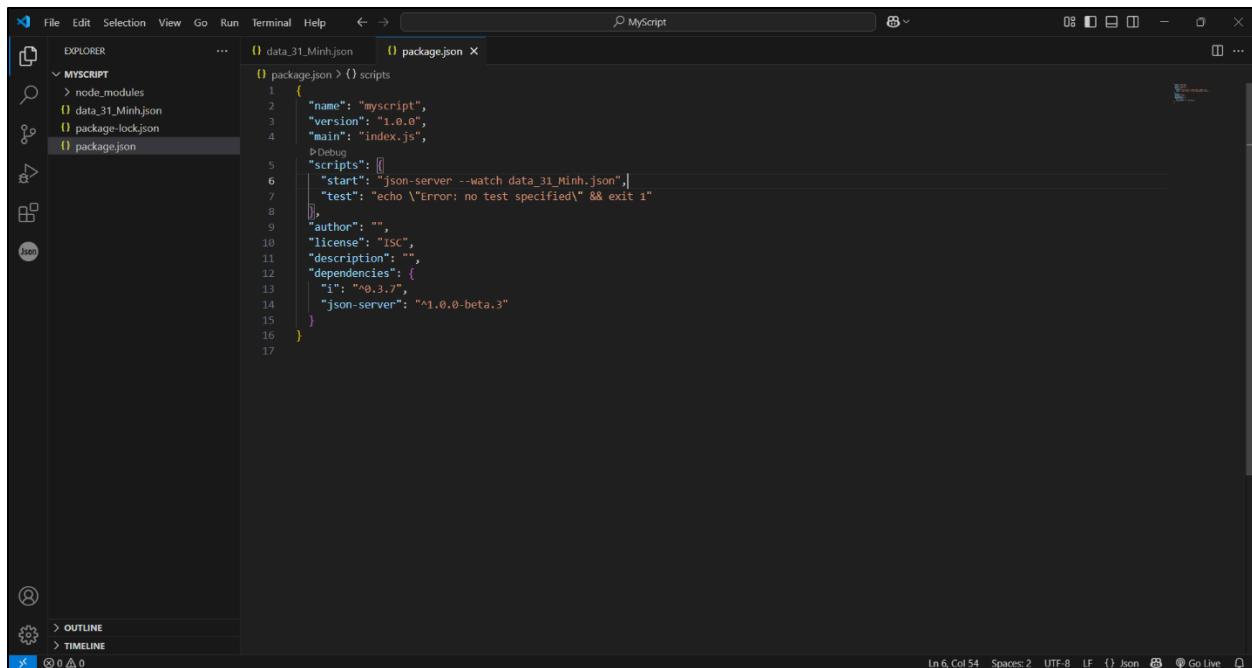
```



```

{
  "users": [
    {
      "id": 1,
      "name": "Nguyễn Văn An",
      "email": "nguyenvanan@gmail.com",
      "phone": "+84901234567",
      "address": {
        "street": "12 Đường Quang Trung",
        "city": "Hồ Chí Minh",
        "country": "Việt Nam"
      },
      "skills": ["Java", "React Native", "C#", "Cloud Computing"],
      "projects": [
        {
          "name": "Social Media App",
          "description": "Đã áp dụng mạng xã hội sử dụng React Native và Firebase.",
          "year": 2025
        },
        {
          "name": "Website Testing Automation",
          "description": "Đã áp dụng thử tự động cho một website bằng C#.",
          "year": 2024
        }
      ]
    },
    {
      "id": 2,
      "name": "Trần Thị Bình",
      "email": "tranthibinh@example.com",
      "phone": "+84987654321",
      "address": {
        "street": "37 Đường Lê Thị Liền",
        "city": "Hà Nội",
        "country": "Việt Nam"
      },
      "skills": ["Python", "Django", "Machine Learning"]
    }
  ]
}

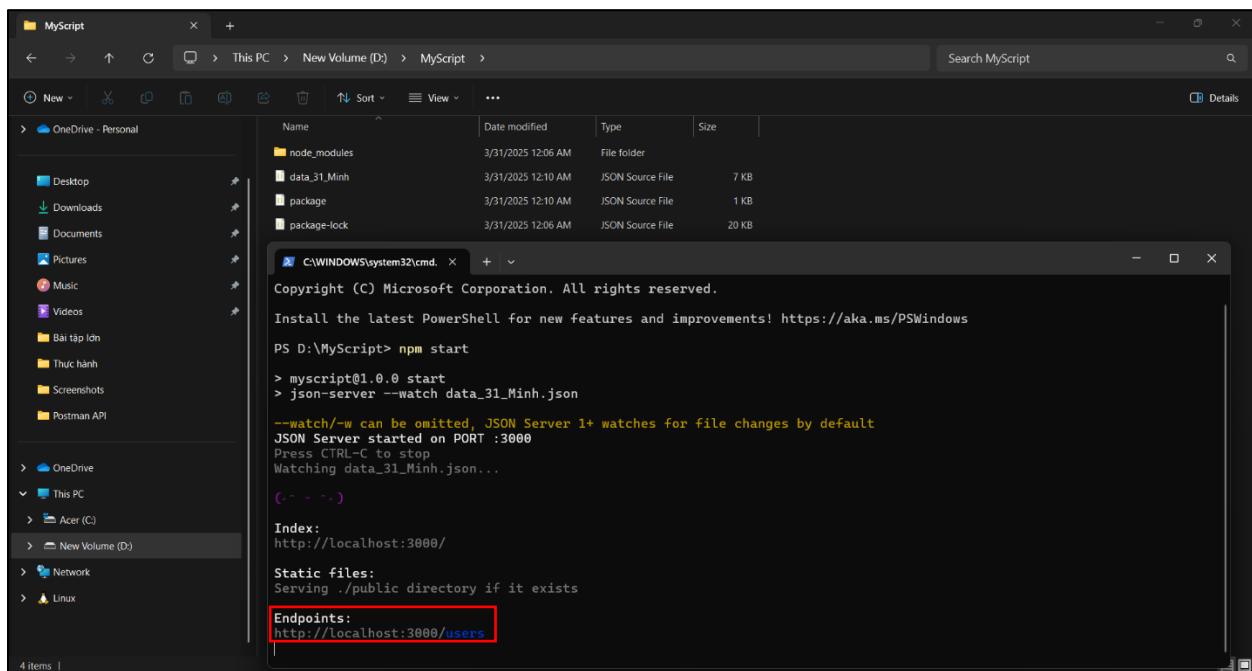
```



```

1  {
2   "name": "myscript",
3   "version": "1.0.0",
4   "main": "index.js",
5   "scripts": [
6     "start": "json-server --watch data_31_Minh.json",
7     "test": "echo \\\"Error: no test specified\\\" && exit 1"
8   ],
9   "author": "",
10  "license": "ISC",
11  "description": "",
12  "dependencies": {
13    "i": "^0.3.7",
14    "json-server": "^1.0.0-beta.3"
15  }
16 }
17

```



Name	Date modified	Type	Size
node_modules	3/31/2025 12:06 AM	File folder	
data_31_Minh	3/31/2025 12:10 AM	JSON Source File	7 KB
package	3/31/2025 12:10 AM	JSON Source File	1 KB
package-lock	3/31/2025 12:06 AM	JSON Source File	20 KB

```

Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS D:\MyScript> npm start
> myscript@1.0.0 start
> json-server --watch data_31_Minh.json
--watch/-w can be omitted, JSON Server 1+ watches for file changes by default
JSON Server started on PORT :3000
Press CTRL-C to stop
Watching data_31_Minh.json...
(.~. ~.~)
Index:
http://localhost:3000/
Static files:
Serving ./public directory if it exists
Endpoints:
http://localhost:3000/users

```

Hình 3.7. Các bước tạo API (31_Minh)

3.2.2. Tạo request để test API:

- Bước 1: Tạo Blank collection > Đặt tên là **Users_31_Minh** > tạo một request > Đặt tên là **TC4_31_Minh_Get_ListUser** với phương thức Get

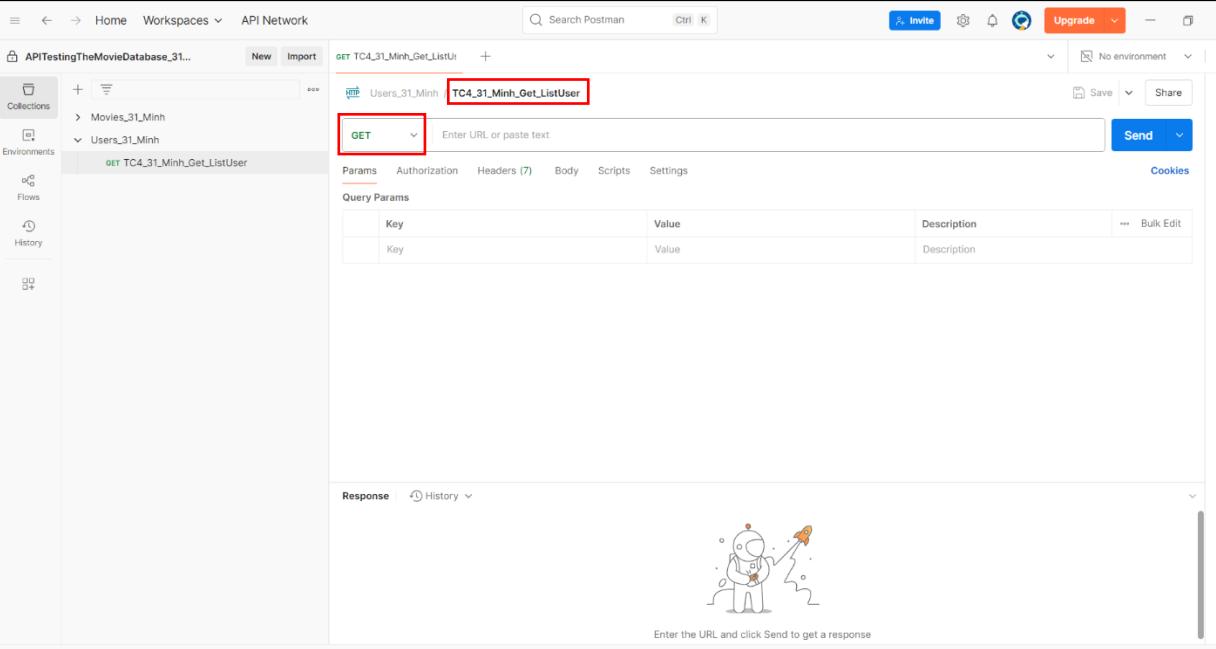
- Bước 2: Chọn biểu tượng “Variables in this request” > Globals > Add > Variable: users > Initial value: <http://localhost:3000/>
- Bước 3: Điền API request {{users}}/users > Send

Request	Test	URL	Status	Time	Body
GET	TC1_31_Minh_Get_ListMovie	https://api.themoviedb.org/3/discover/movie	200 OK	133 ms	5.873 KB
	PASS	TC1_31_Minh_ResponseStatusCodes200OK			
	PASS	TC2_31_Minh_ResponseIsInJSONFormat			
	PASS	TC3_31_Minh_ImageURLsHaveTheCorrectExtension			
	PASS	TC4_31_Minh_ResponseContainsMovieInformation			
	PASS	TC5_31_Minh_CheckProperty			
POST	TC2_31_Minh_Post_AddFavoriteMovie	https://api.themoviedb.org/3/account/21913469/favorite	201 Created	447 ms	606 B
	PASS	TC7_31_Minh_ResponseStatusCodes201OK			
	PASS	TC8_31_Minh_CheckFormData			
DELETE	TC3_31_Minh_Delete_RatingMovie	https://api.themoviedb.org/3/movie/1165057/rating	200 OK	297 ms	635 B
	PASS	TC9_31_Minh_StatusCodes200			
	PASS	TC10_31_Minh_CheckSuccessIsBoolean			
	TC11_31_Minh_CheckStatus_messageContainsExpectedMessage				

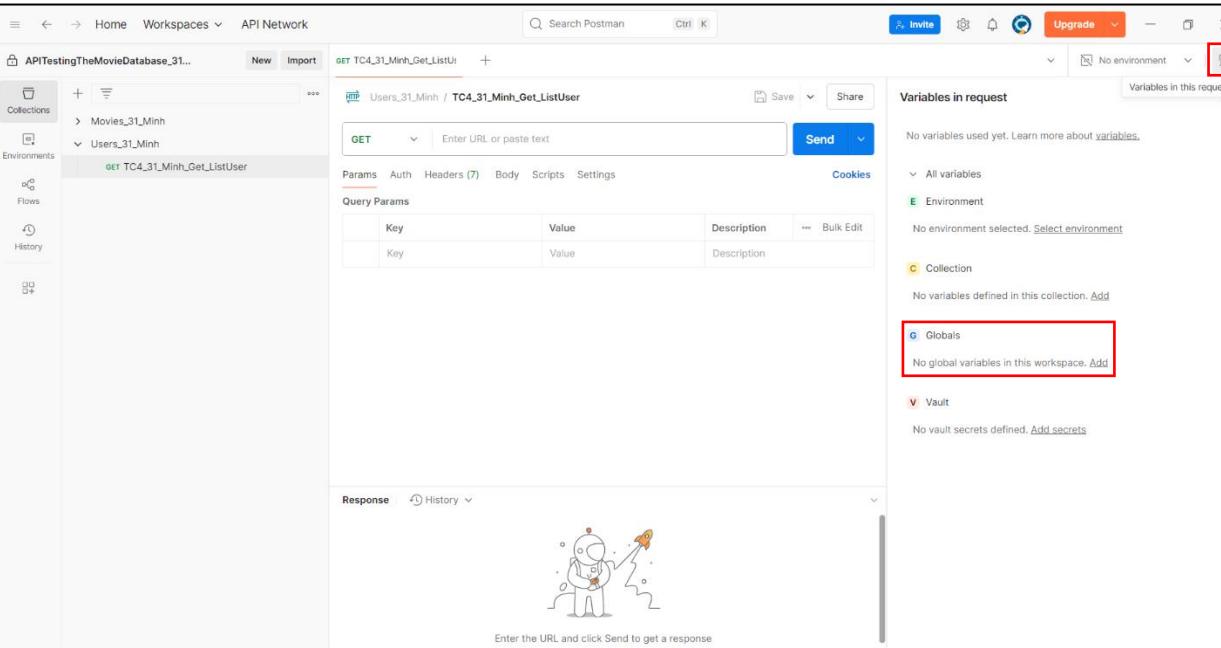
New Collection

Make things easier for your teammates with a complete collection description.

View complete documentation →



The screenshot shows the Postman application interface. On the left, the sidebar includes 'Collections' (Movies_31_Minh, Users_31_Minh), 'Environments' (Users_31_Minh), 'Flows', and 'History'. The main workspace displays a collection named 'APITestingTheMovieDatabase_31...'. A specific request is selected: 'GET TC4_31_Minh_Get_ListUser'. The 'Method' dropdown is highlighted with a red box and set to 'GET'. The 'URL' field contains 'TC4_31_Minh_Get_ListUser'. Below the URL, the 'Params', 'Authorization', 'Headers (7)', 'Body', 'Scripts', and 'Settings' tabs are visible. The 'Headers' tab is selected. The 'Query Params' table has one row with 'Key' and 'Value' columns, both empty. On the right, there are 'Save', 'Share', 'Send', and 'Cookies' buttons. The 'Response' tab is open, showing a small cartoon character with a rocket ship and the placeholder text 'Enter the URL and click Send to get a response'. The bottom navigation bar includes 'Online', 'Find and replace', 'Console', 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and 'Help'.



The screenshot shows the Postman application interface. On the left, the sidebar includes 'Collections' (selected), 'Environments', 'Flows', and 'History'. The main workspace shows a collection named 'APITestingTheMovieDatabase_31...'. A specific test case 'TC4_31_Minh_Get_ListUser' is selected, showing a 'GET' request to 'http://Users_31_Minh / TC4_31_Minh_Get_ListUser'. The 'Variables in request' section indicates no variables are used. The 'Global' section (highlighted with a red box) states 'No global variables in this workspace. Add'. The 'Response' section at the bottom features a cartoon character of a spaceman launching a rocket from a planet, with the text 'Enter the URL and click Send to get a response'.

The screenshot shows the Postman interface. In the top navigation bar, 'Home' and 'Workspaces' are visible. The workspace 'APITestingTheMovieDatabase_31...' is selected. In the left sidebar, 'Collections' and 'Environments' are listed. Under 'Collections', 'Movies_31_Minh' and 'Users_31_Minh' are shown. Under 'Environments', 'GET TC4_31_Minh_Get_ListUser' is selected. The main area is titled 'Globals' with a sub-section 'Global variables for a workspace are a set of variables that are always available within the scope of that workspace. They can be viewed and edited by anyone in that workspace.' Below this is a table with a red border:

Variable	Type	Initial value	Current value
<input checked="" type="checkbox"/> users	default	http://localhost:3000/	http://localhost:3000/

At the bottom of the table, there is a link 'Add new variable'.

The screenshot shows the Postman interface with the same workspace and collection as the previous screenshot. The request URL is 'http://localhost:3000/users' and the method is 'GET'. The 'Send' button is highlighted with a red border. Below the URL, the 'Params' tab is selected, showing a table:

Key	Value	Description	Bulk Edit
Key	Value	Description	

Below the table, there is a 'Response' section with a small illustration of an astronaut and the text 'Click Send to get a response'.

Hình 3.8. Các bước tạo request script của API vừa tạo (31_Minh)

The screenshot shows the Postman application interface. In the left sidebar, there are sections for Collections, Environments, Flows, and History. The main workspace shows a collection named 'APITestingTheMovieDatabase_31...' with a sub-collection 'Users_31_Minh'. A specific request named 'TC4_31_Minh_Get_ListUser' is selected, which is a GET request to the URL 'http://Users_31_Minh / TC4_31_Minh_Get_ListUser'. The 'Body' tab is selected, showing a JSON response with a status of 200 OK. The response body is a JSON array containing one user object, which is highlighted with a red box. The user object has fields: id, name, email, phone, address, skills, and projects. The 'Skills' array contains Java, React Native, C#, and Cloud Computing. The 'Projects' array contains a single project named 'Social Media App'.

Hình 3.9. Kết quả sau khi gửi request (31_Minh)

3.2.3. Viết các API Test Case:

- Thực hiện viết các test case và Run collection như trên.

```
//TC12_31_Minh: Kiểm tra trạng thái mã HTTP trả về là 200 OK
pm.test("TC12_31_Minh_StatusCodeIs200OK", function () {
    pm.response.to.have.status(200);
});

//TC13_31_Minh: Kiểm tra trả về của API có phải là JSON không?
pm.test("TC13_31_Minh_ResponseShouldBeJSON", function () {
    pm.response.to.be.json();
});

//TC14_31_Minh: Kiểm tra trả về của API có phải là một mảng danh sách không?
pm.test("TC14_31_Minh_ResponseShouldBeAnArray", function () {
    var jsonData_31_Minh = pm.response.json();
    pm.expect(jsonData_31_Minh).to.be.an('array');
});
```

```

//TC15_31_Minh: Kiểm tra danh sách người dùng có đúng 8
user không?
pm.test("TC15_31_Minh_UserListShouldHave8Users", function
() {
    var jsonData_31_Minh = pm.response.json();
    pm.expect(jsonData_31_Minh.length).to.equal(8);
});

//TC16_31_Minh: Kiểm tra mỗi người dùng có đủ thông tin
không?
pm.test("TC16_31_Minh_EachUserShouldHaveEnoughInfomation",
function () {
    var jsonData_31_Minh = pm.response.json();
    jsonData_31_Minh.forEach(user_31_Minh => {
        pm.expect(user_31_Minh.skills).to.be.an('array');

        pm.expect(user_31_Minh).to.have.property('id');
        pm.expect(user_31_Minh).to.have.property('name');
        pm.expect(user_31_Minh).to.have.property('email');
        pm.expect(user_31_Minh).to.have.property('phone');
        pm.expect(user_31_Minh).to.have.property('skills');

    });
});
}
);

```

Postman interface showing the 'Users_31_Minh' collection selected. A red box highlights the 'Run collection' option in the context menu. The main panel displays a GET request for '/users' with a JSON pre-request script.

```

    //TC12_31_Minh: Kiểm tra trạng thái mã HTTP trả về là 200 OK
    pm.test("TC12_31_Minh_StatusCodeIs200OK", function () {
        pm.response.to.have.status(200);
    });

    //TC13_31_Minh: Kiểm tra trả về của API có phải là JSON không?
    pm.test("TC13_31_Minh_ResponseShouldBeJSON", function () {
        pm.response.to.be.json();
    });

    //TC14_31_Minh: Kiểm tra trả về của API có phải là một mảng danh sách không?
    pm.test("TC14_31_Minh_ResponseShouldBeAnArray", function () {
        var jsonData_31_Minh = pm.response.json();
        pm.expect(jsonData_31_Minh).to.be.an('array');
    });

    //TC15_31_Minh: Kiểm tra danh sách người dùng có đúng 8 user không?
    pm.test("TC15_31_Minh_UserListShouldHave8Users", function () {
        var jsonData_31_Minh = pm.response.json();
        pm.expect(jsonData_31_Minh.length).to.equal(8);
    });

    //TC16_31_Minh: Kiểm tra mỗi người dùng có đủ thông tin không?
    pm.test("TC16_31_Minh_EachUserShouldHaveEnoughInformation", function () {
        var jsonData_31_Minh = pm.response.json();
        jsonData_31_Minh.forEach(user_31_Minh => {
            pm.expect(user_31_Minh.skills).to.be.an('array');
            pm.expect(user_31_Minh).to.have.property('id');
        });
    });

```

Postman interface showing the 'Runner' tab selected. A red box highlights the 'Run Users_31_Minh' button. The 'Run Sequence' panel shows a single step: '1 GET TC4_31_Minh_Get_ListUser'.

Choose how to run your collection

- Run manually

Run this collection in the Collection Runner.
- Schedule runs

Periodically run collection at a specified time on the Postman Cloud.
- Automate runs via CLI

Configure CLI command to run on your build pipeline.

Run configuration

Iterations: 1

Delay: 0 ms

Test data file: (Only JSON and CSV files are accepted.)

Run Users_31_Minh

25/25 runs used. Get unlimited runs with add-on.

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1	461ms	5	6 ms

All Tests Passed (5) Failed (0) Skipped (0) [View Summary](#)

Iteration 1

GET TC4_31_Minh_GetListUser
http://localhost:3000/users

PASS	TC12_31_Minh_StatusCodeIs200OK
PASS	TC13_31_Minh_ResponseShouldBeJSON
PASS	TC14_31_Minh_ResponseShouldBeAnArray
PASS	TC15_31_Minh_UserListShouldHave8Users
PASS	TC16_31_Minh_EachUserShouldHaveEnoughInformation

Hình 3.10. Thực hiện Run collection các test case và kết quả sau khi chạy (31_Minh)

--- Hết ---