**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

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

**KHOA CÔNG NGHỆ THÔNG TIN**

---🙡 🕮 🙣---

**BÁO CÁO**

**KIỂM THỬ PHẦN MỀM**

**Đề Tài**

**KIỂM THỬ WEBSITE MẠNG XÃ HỘI INSTAGRAM**

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

**Sinh viên thực hiện:** Trần Minh Bảo Long

**Mã số sinh viên:** 2151050235

**Lớp:** DH21IT02

**Khóa:** 2021-2025

**Tháng 4 năm 2024, TP. HỒ CHÍ MINH**

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Mục Lục

[Chương 1. Giao diện trang web 2](#_Toc165563919)

[1.1. Trang đăng nhập 2](#_Toc165563920)

[1.2. Trang thông tin cá nhân 3](#_Toc165563921)

[Chương 2. KIỂM THỬ SELENIUM 3](#_Toc165563922)

[Chương 3. THỰC HIỆN KIỂM THỬ 10](#_Toc165563923)

[3.1. Chức năng đăng nhập 10](#_Toc165563924)

[3.1.1. Viết test case 10](#_Toc165563925)

[3.2. Chức năng tìm kiếm 18](#_Toc165563926)

[3.2.1. TC1:tìm kiếm thành công 18](#_Toc165563927)

[3.2.2. TC2:Tìm kiếm thất bại 22](#_Toc165563928)

[Chương 4. Kiểm thử API 24](#_Toc165563929)

[4.1. Tạo JsonServer 24](#_Toc165563930)

[4.2. Thiết lập Postman 27](#_Toc165563931)

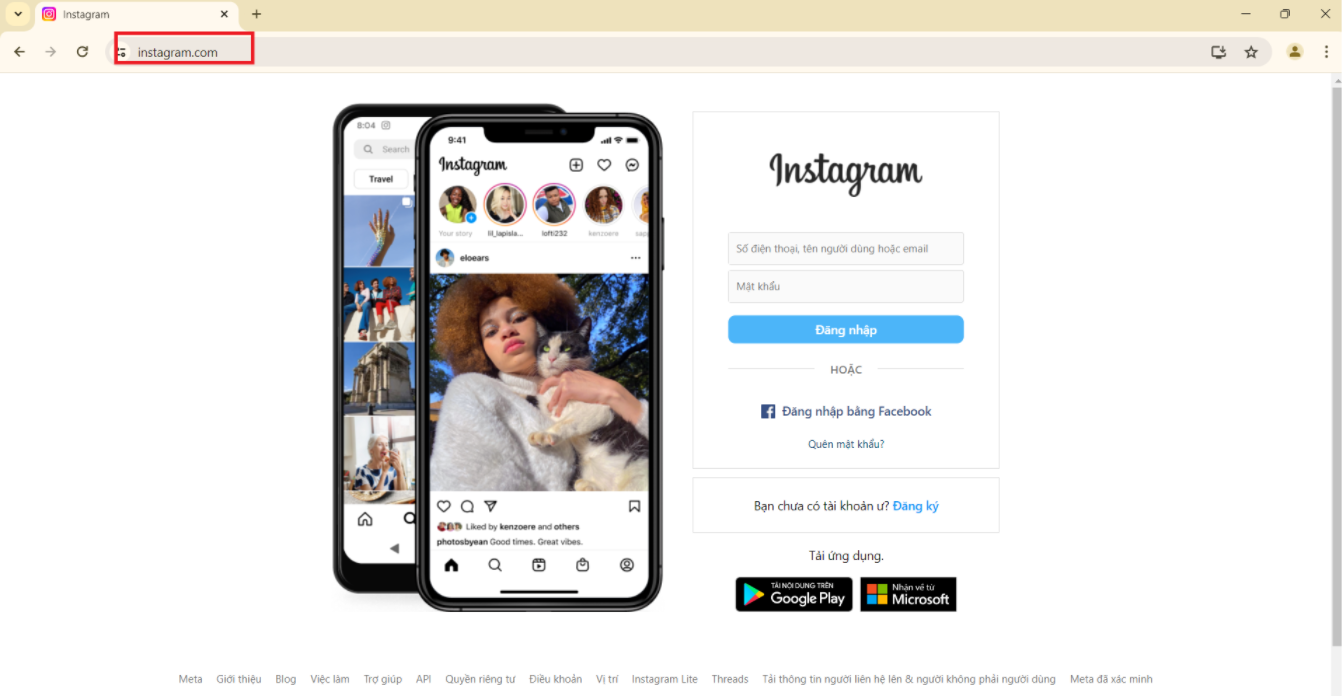
[4.3. Test Api và viết test case 30](#_Toc165563932)

MỞ ĐẦU

Instagram là một mạng xã hội chia sẻ ảnh và video lớn và phổ biến nhất hiện nay bên cạnh 2 mạng xã hội lớn là Facebook và Twitter . Đây cũng là một ứng dụng miễn phí trên nền tảng iOS và Android với tính năng cung cấp rất nhiều các chế độ chỉnh sửa ảnh và video khác nhau dựa trên sở thích của người dùng. Với Instagram, người dùng có thể chụp ảnh trên điện thoại của mình, thêm các bộ lọc hình ảnh, chỉnh sửa hình ảnh trực tiếp trên công cụ này và chia sẻ trên nhiều mạng xã hội khác nhau.

# Giao diện trang web

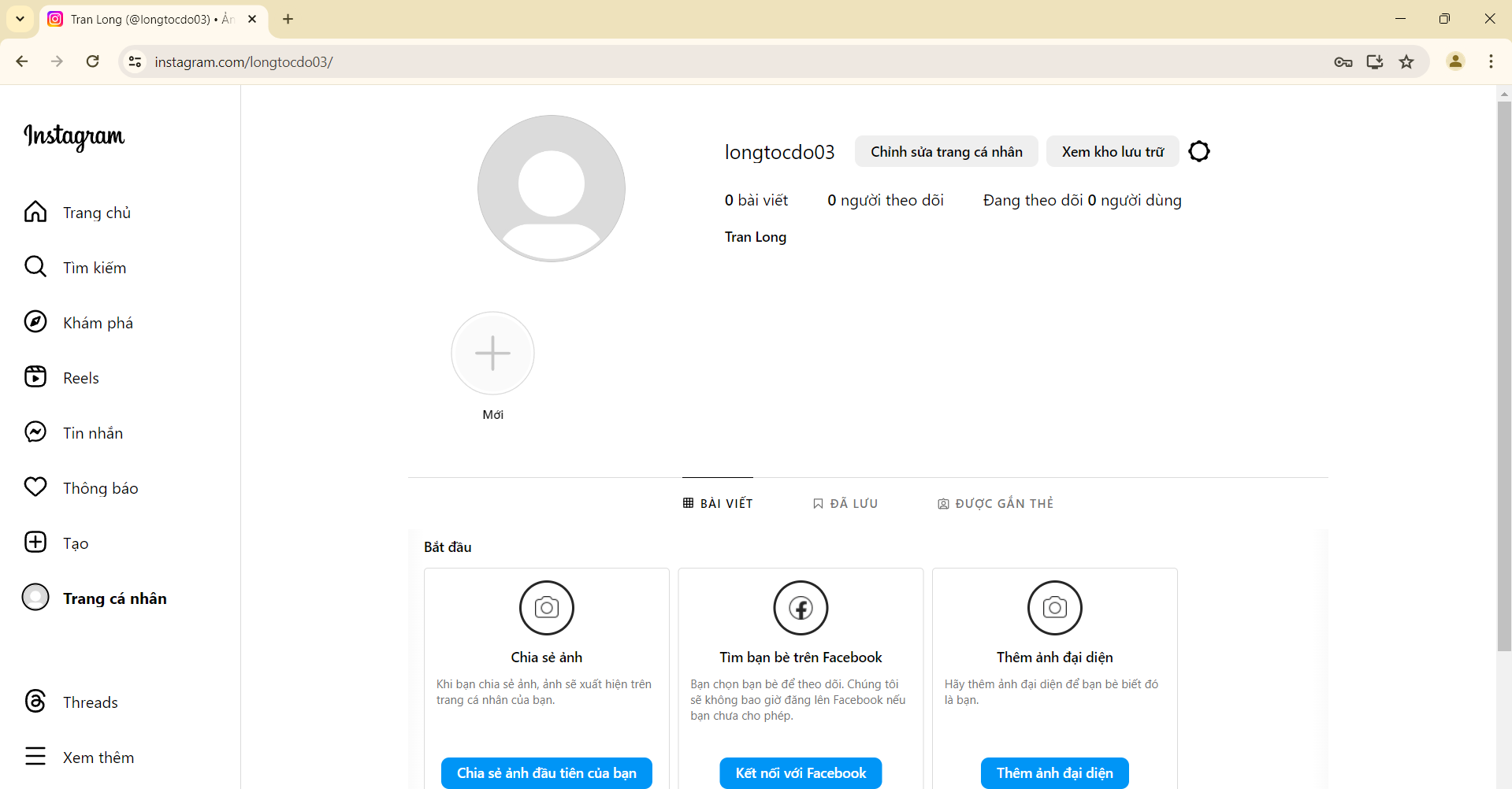
## Trang đăng nhập



**Trang đăng nhập chính**

Trang đăng nhập

## Trang thông tin cá nhân



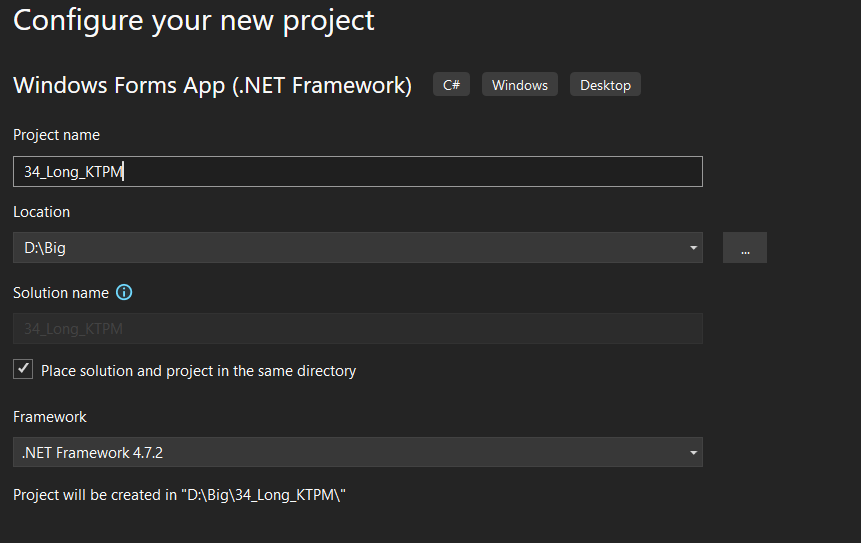
Trang thông tin cá nhân

# KIỂM THỬ SELENIUM

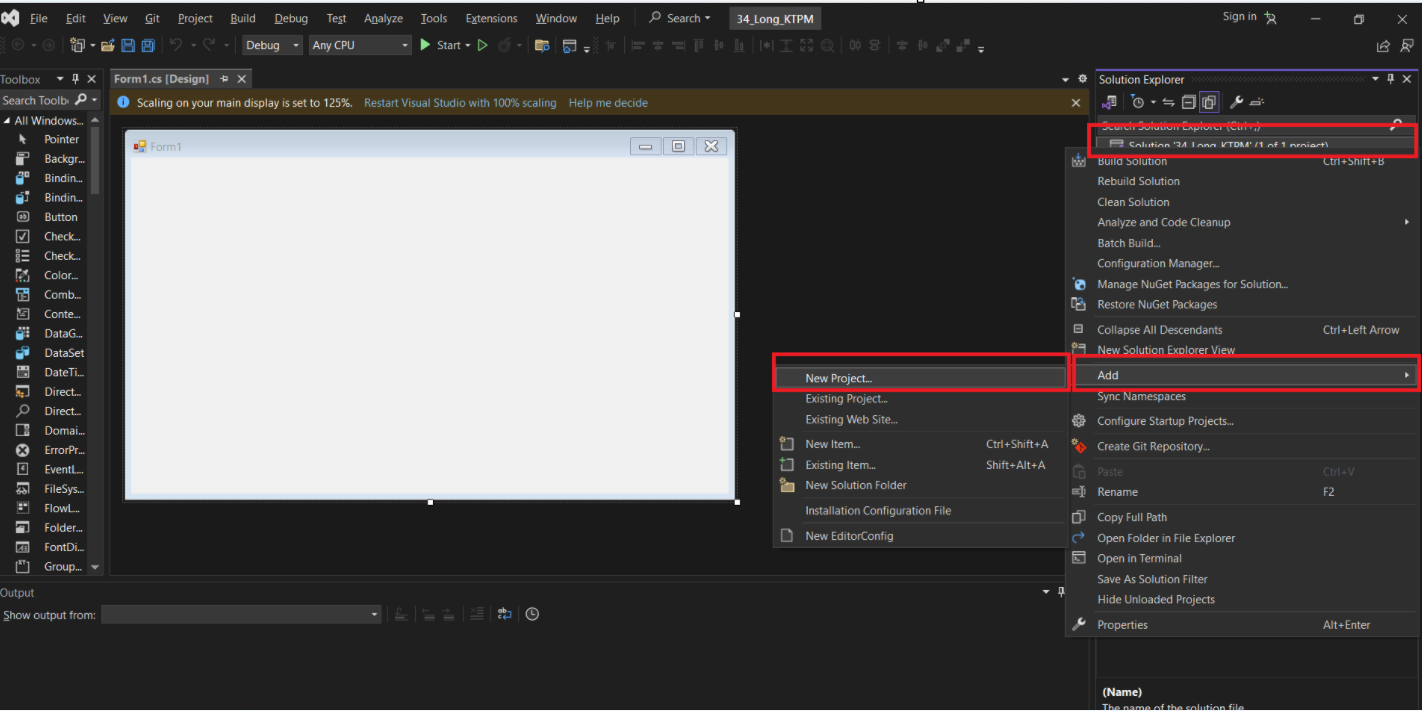
* Set up môi trường
* Tạo project và thêm thư viện kiểm thử
* Tạo project

Mở visual studio lên vào tạo project với window forms app

Đặt tên cho project và chọn phiên bản framework 4.7.2 sau đó create



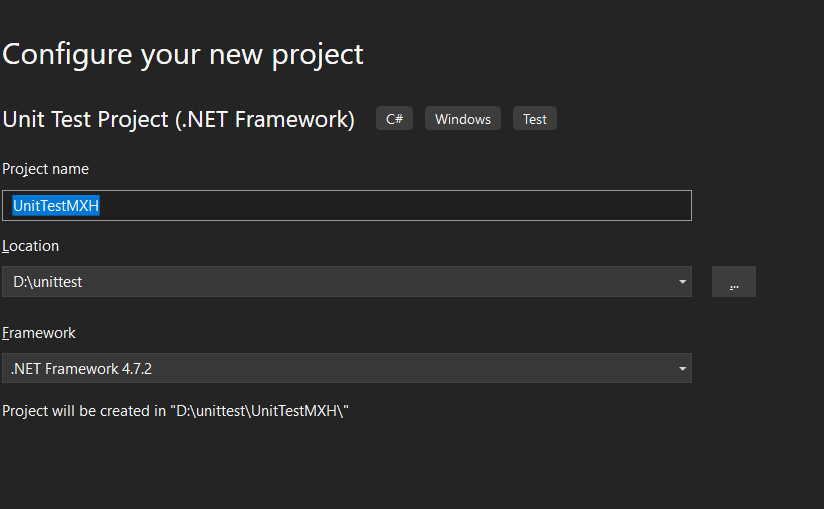
Để thêm môi trường kiểm thử vào project ta chuột phải vào solution và add new project



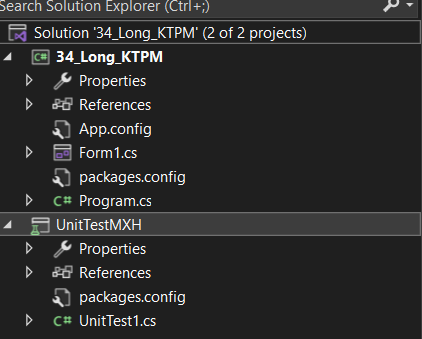
Tìm Unit Test Project



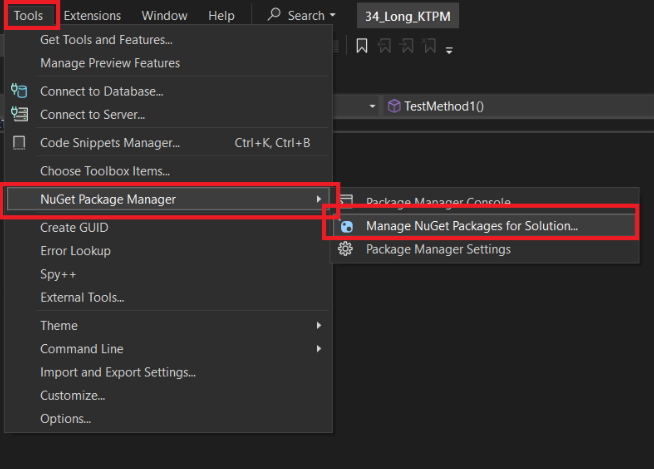
Đặt tên project và để phiên bản framework 4.7.2 và bấm tạo



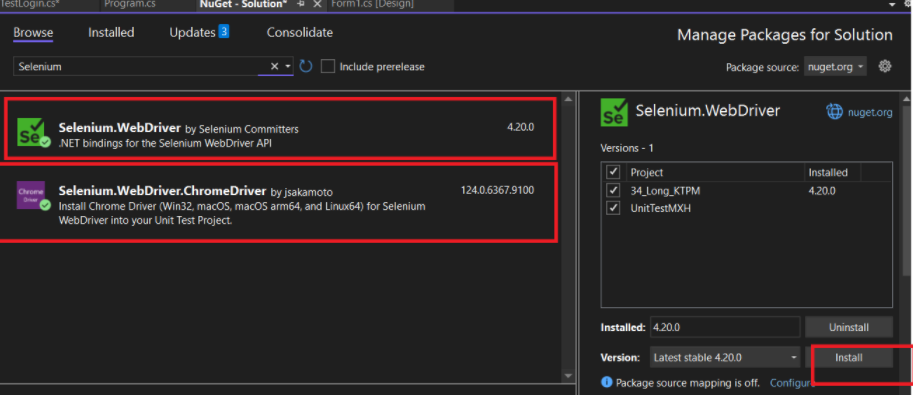
solution sẽ có 2 project là window forms và unit test



Thêm thư viện selenium vào project

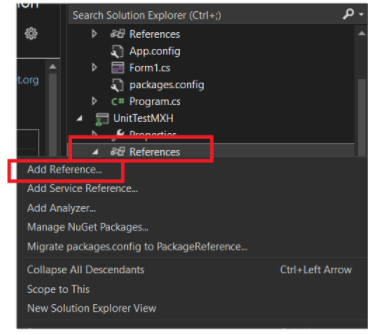


Chọn browse tìm selenium và thêm 2 thư viện Selenium.WebDriver.ChromeDriver và Selenium.WebDriver và cả 2 project đã tạo trước đó và bấm install

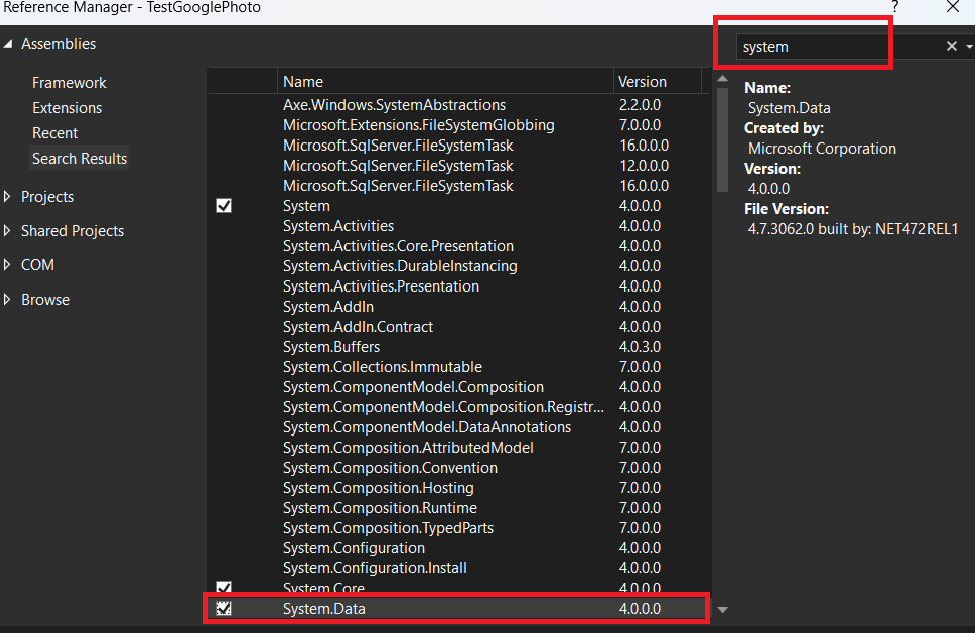


Thêm thư viện để đọc file csv lấy testcase

đầu tiên vào project unit test chọn references và add reference mới vào



tìm thư viện System data để thêm vào project



# THỰC HIỆN KIỂM THỬ

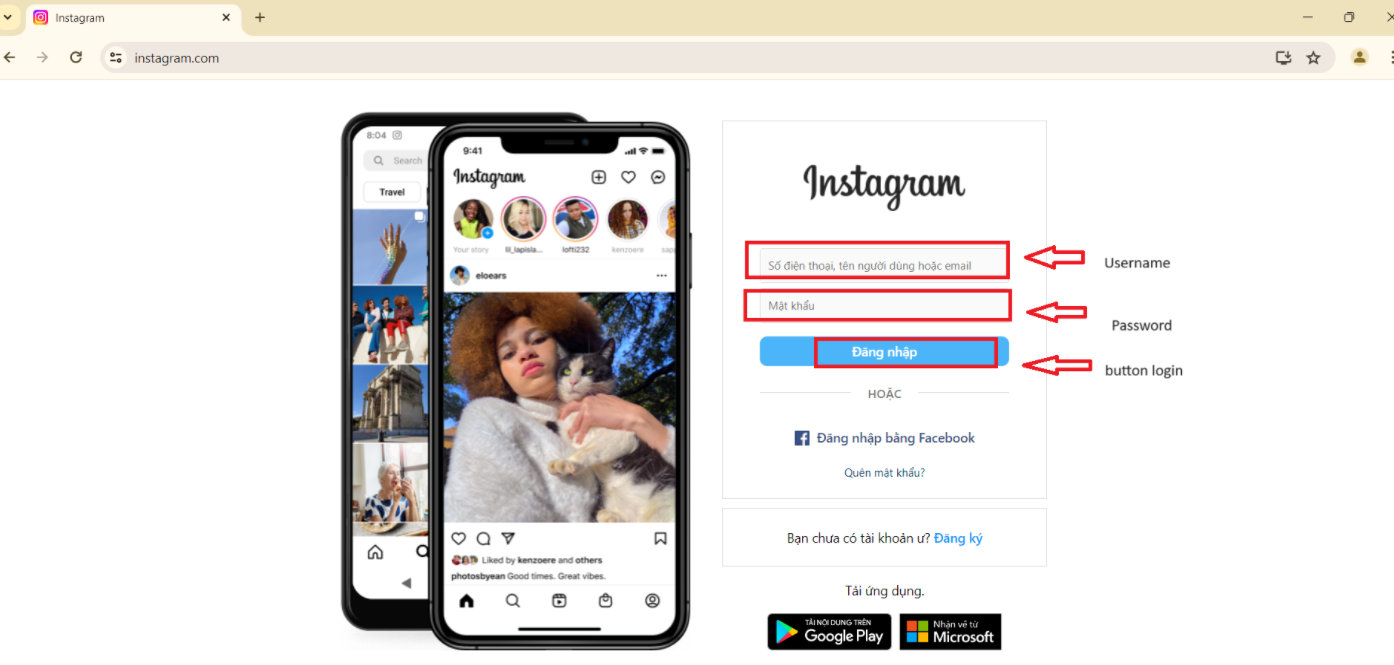
## Chức năng đăng nhập

### Viết test case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | TC1 | TC2 | TC3 |
| Điều kiện | Đúng tài khoản | F | T | T |
| Đúng mật khẩu | - | F | T |
| Hành Động | Đăng nhập thành công | F | F | T |
|

#### Các bước thực hiện

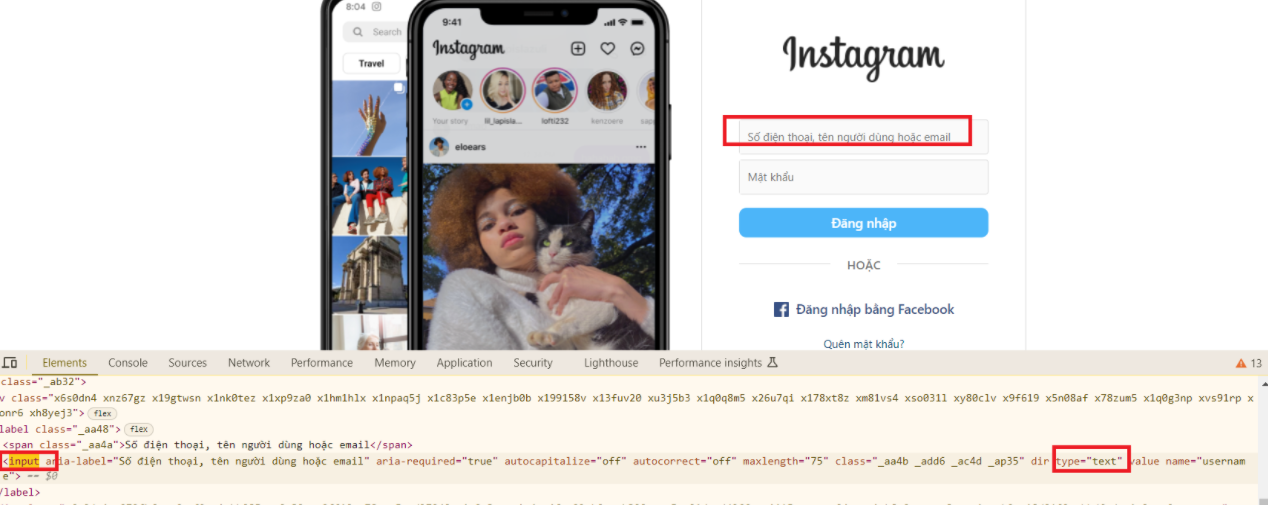
Tìm element



Các element cần thiết

Element username

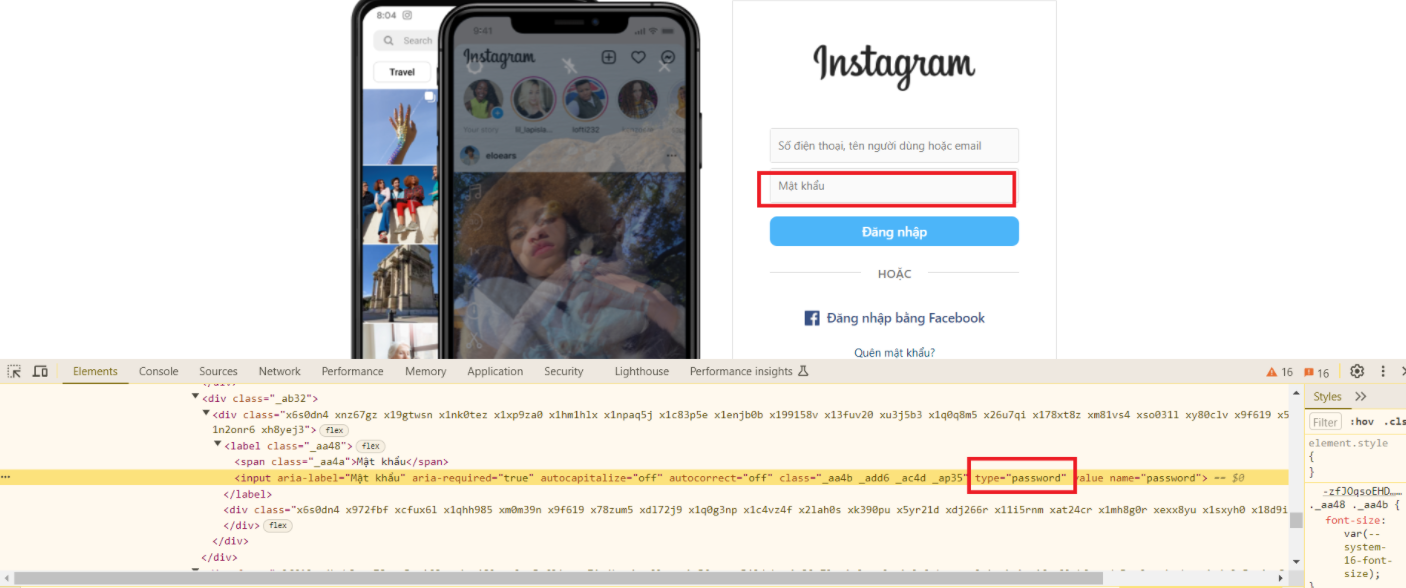
Sử dụng driver.FindElement(By.CssSelector("input[type=’text’]"));



Element username

Elemenet password

Sử dụng driver.FindElement(By.CssSelector("input[type='password']")



Element password

Element button login

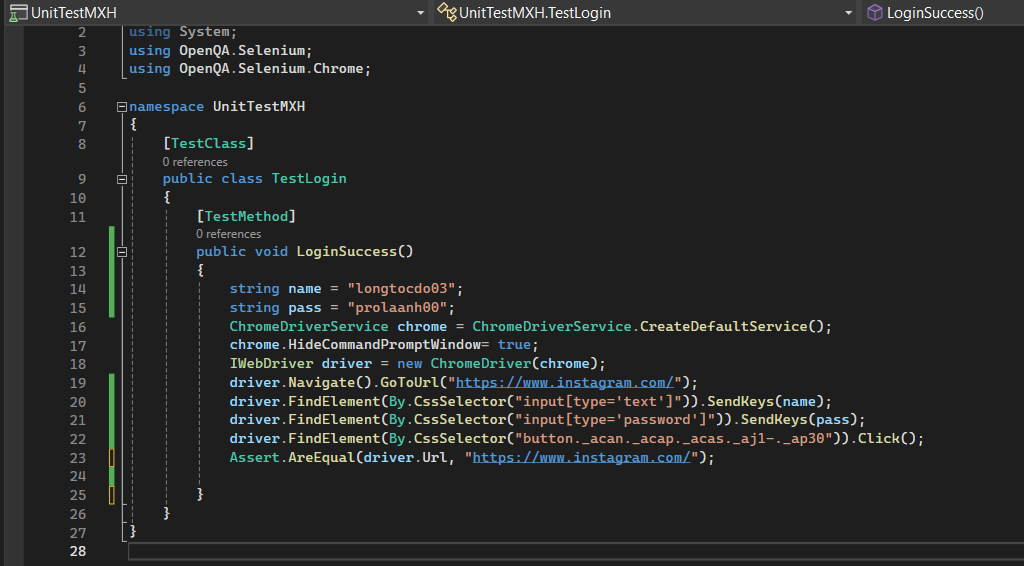
Sử dụng driver.find\_element(



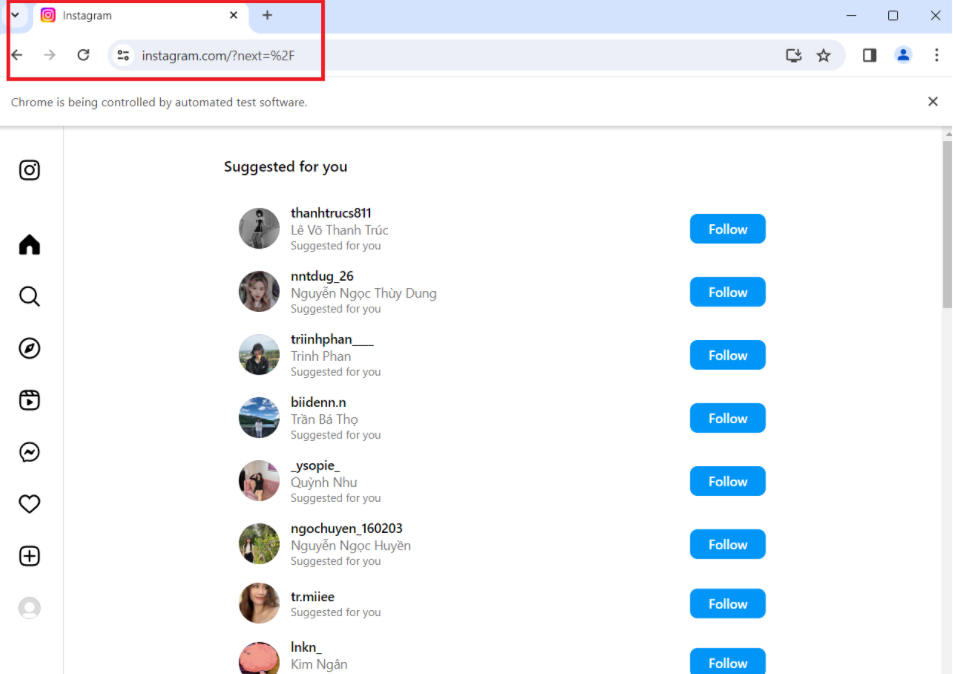
Element button login

#### TC1: Đăng nhập thành công

Code



Kết quả: Đăng nhập thành công

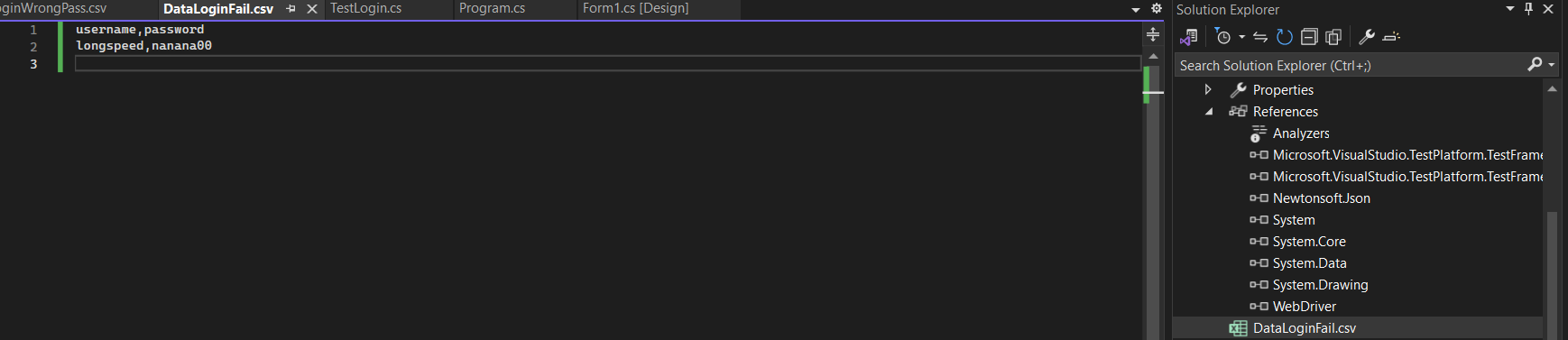


Kết quả tc1

vậy là 1 test case đã chạy thành công

#### TestCase 2: Sai username

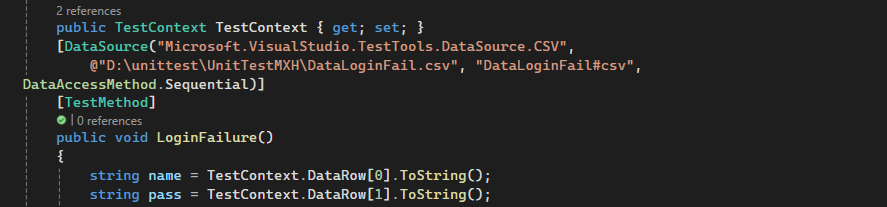
Để kiểm thử hàng loạt testcase ta sẽ tạo 1 file csv để kiểm thử hàng loạt



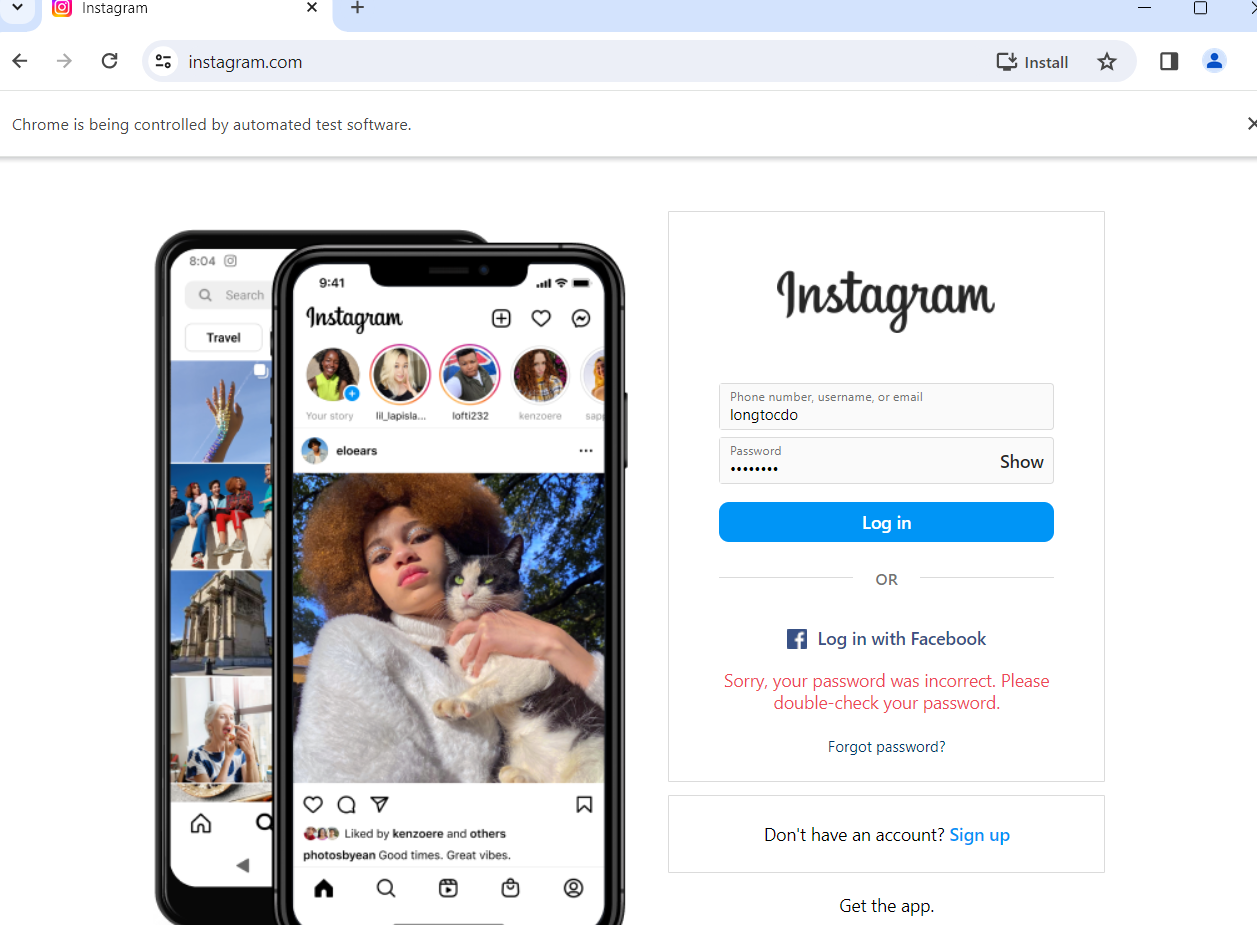
thêm lớp đọc file vào

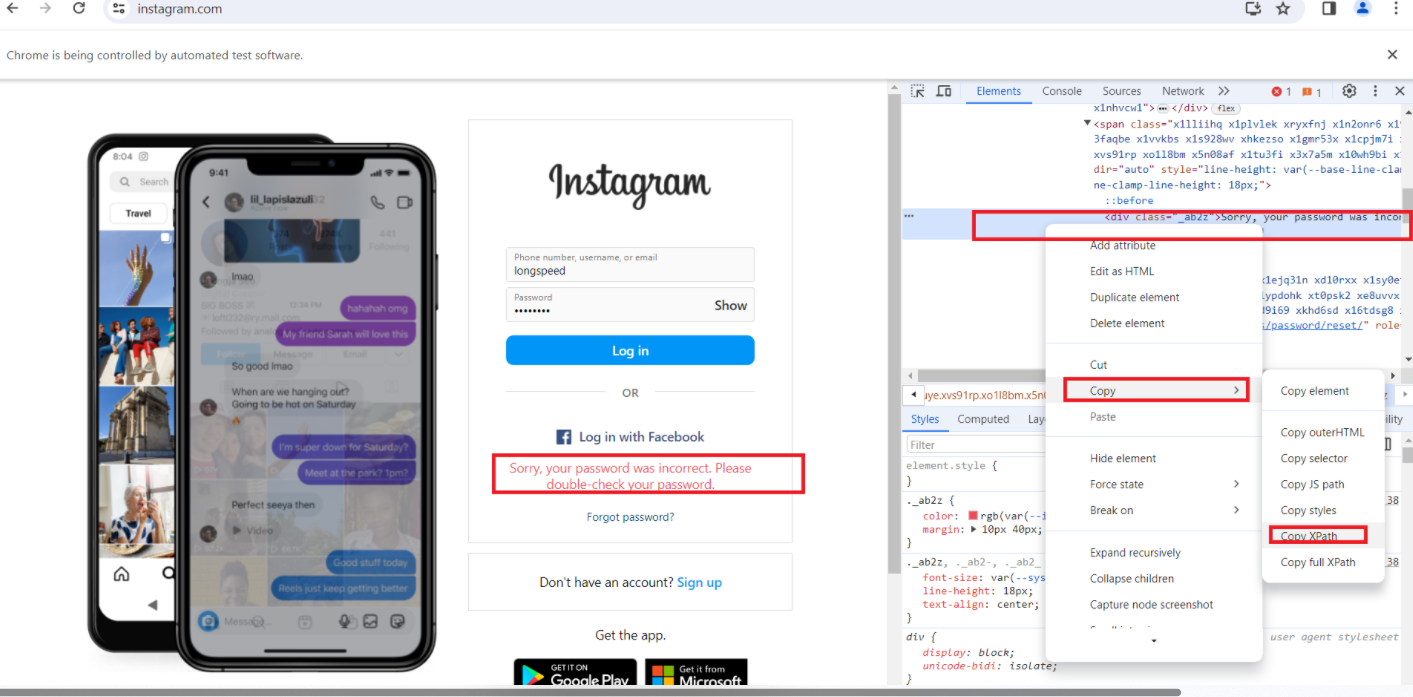


Thêm nguồn vào và gắn datarow tại 0 vào user và 1 cho password khi đọc

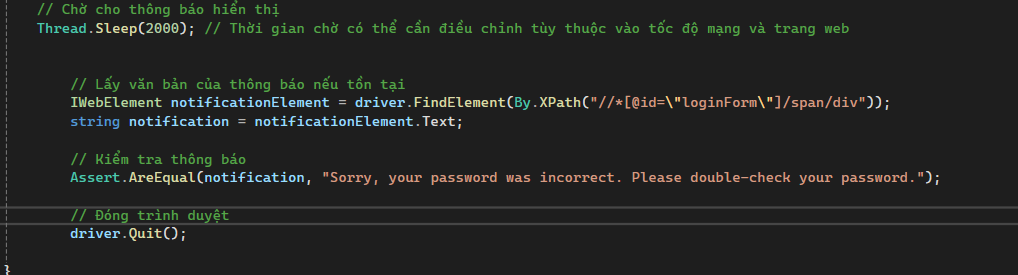


khi đăng nhập không thành công nó sẽ xuất hiện dòng màu đỏ

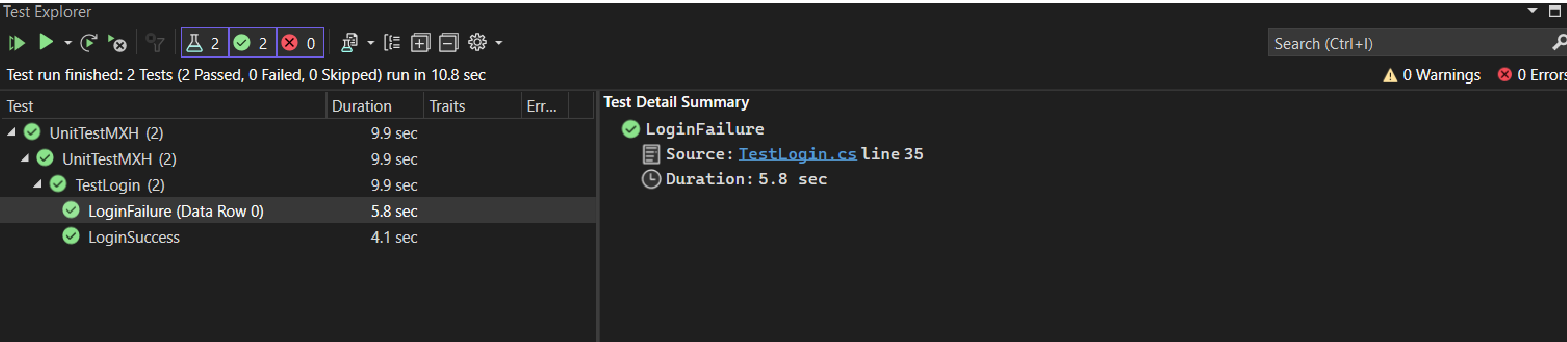




Sử dụng driver.FindElement(By.XPath("//\*[@id=\"loginForm\"]/span/div"));



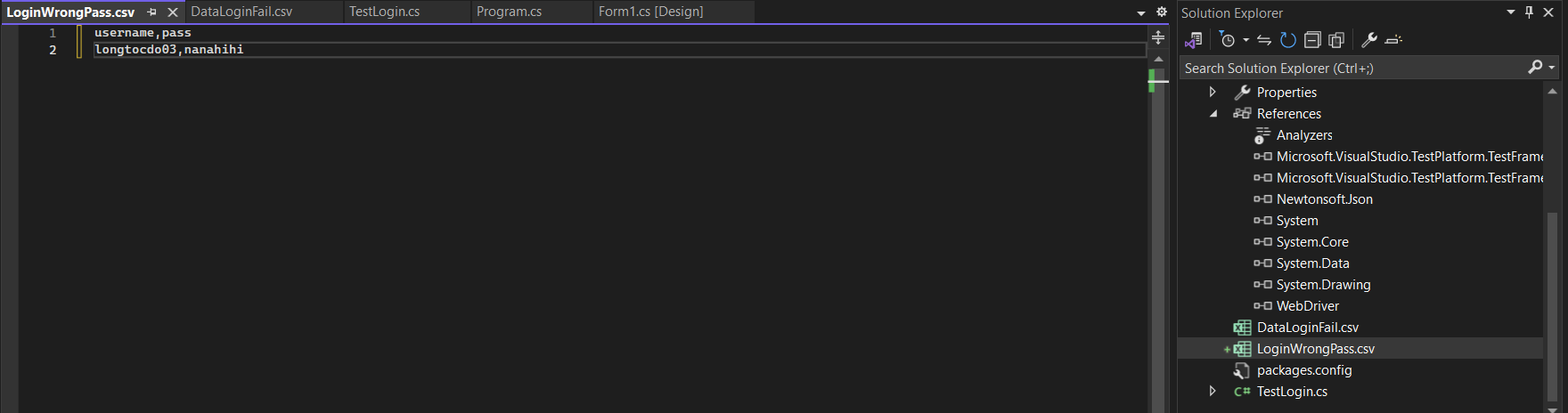
và chạy test case



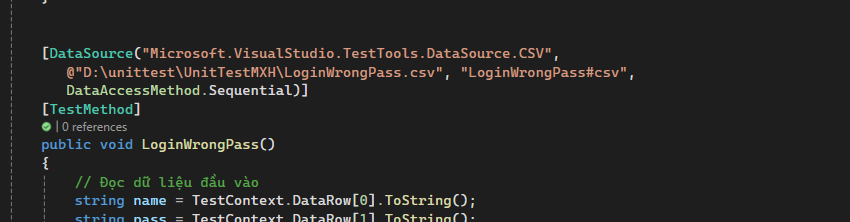
#### Testcase3 (Đăng nhập sai mật khẩu):

tương tự test case LoginFailure

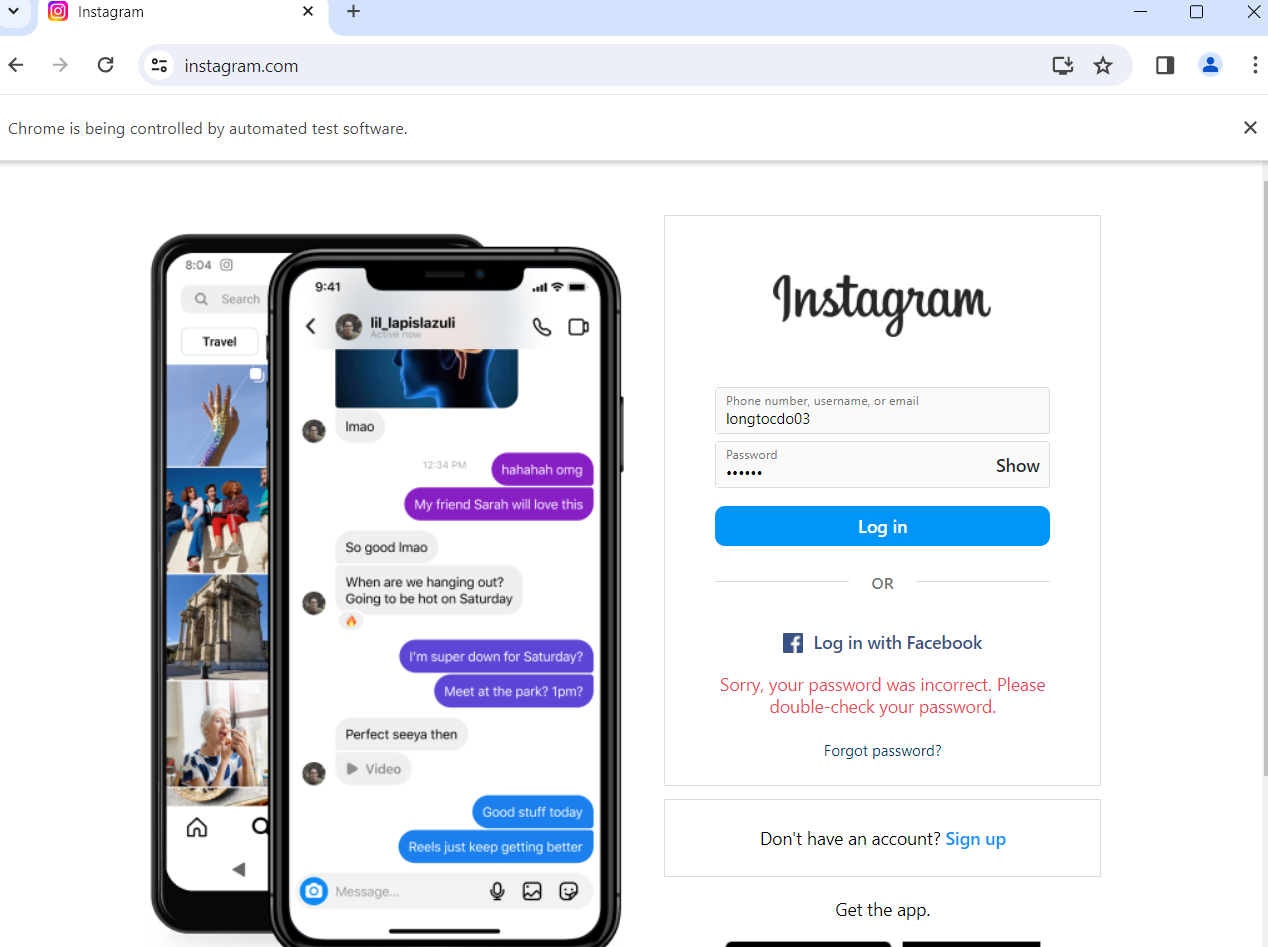
Thêm file csv

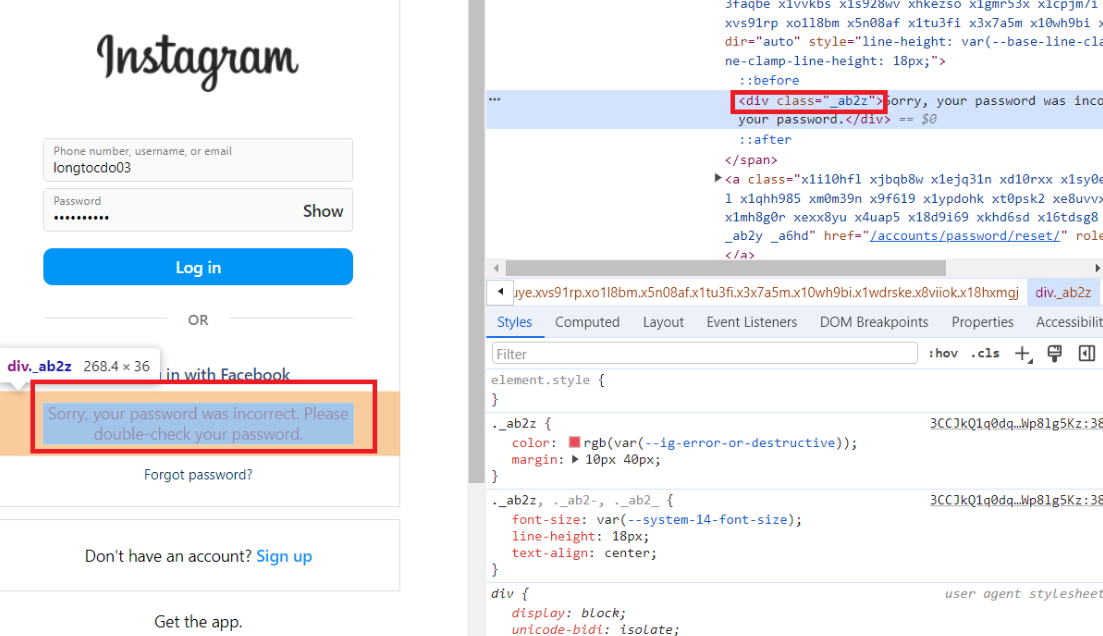


Thêm nguồn vào và gắn datarow tại 0 vào user và 1 cho password khi đọc

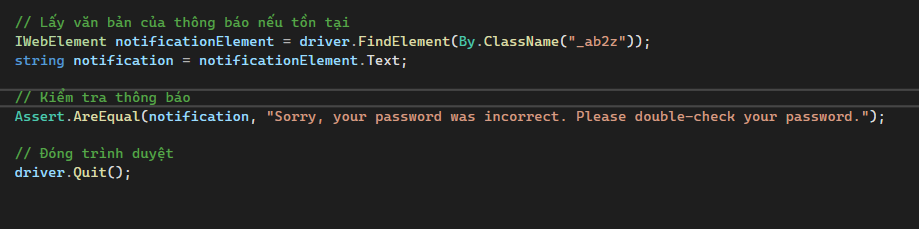


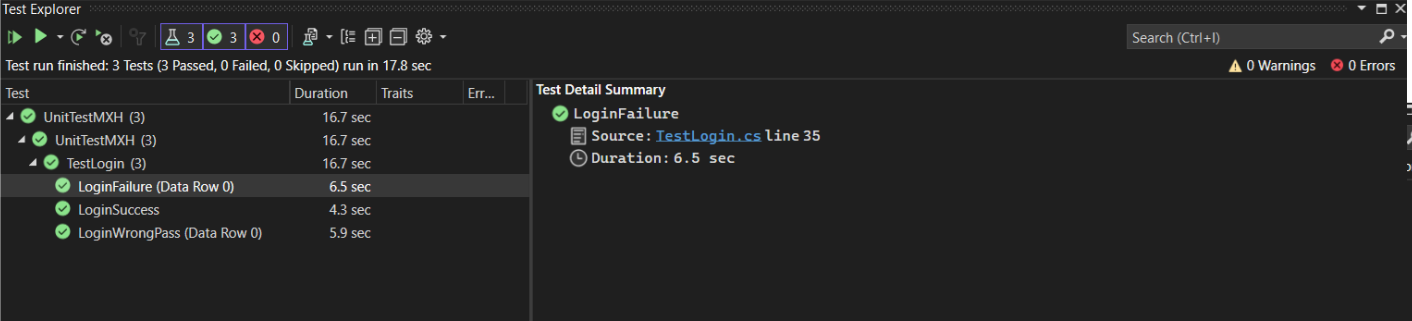
khi đăng nhập không thành công nó sẽ xuất hiện dòng màu đỏ





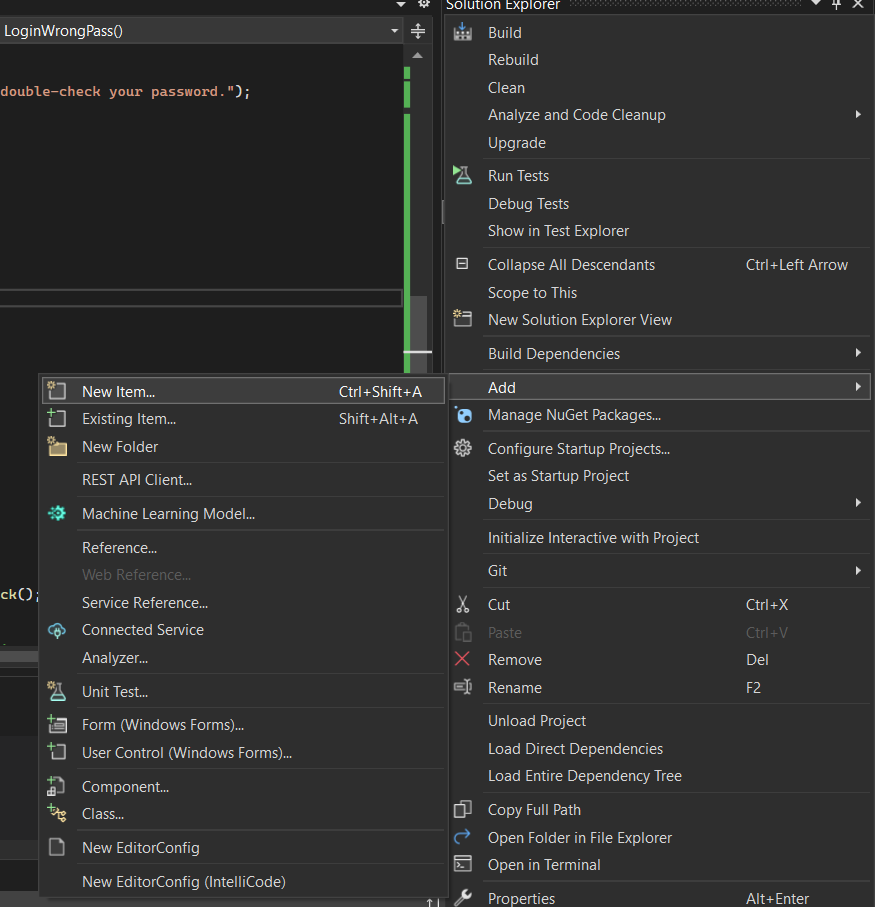
Lần này bắt driver.FindElement(By.ClassName("\_ab2z"))

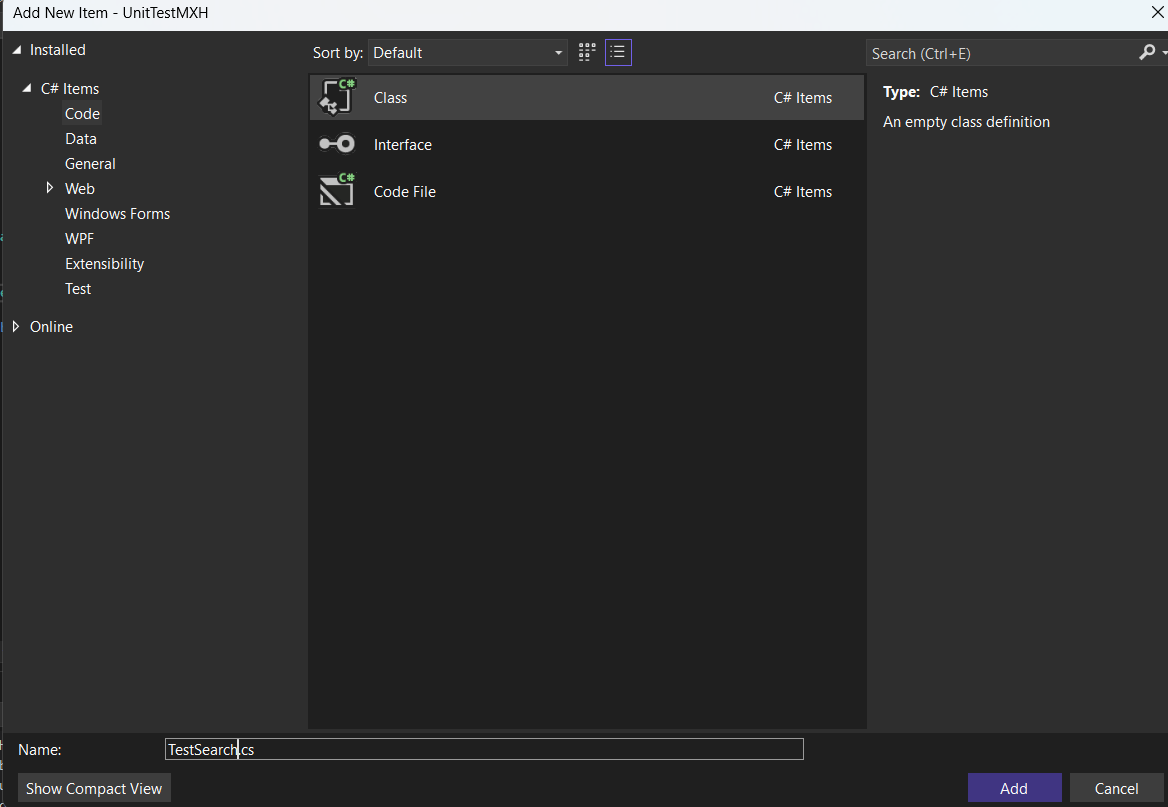


Kết quả:  


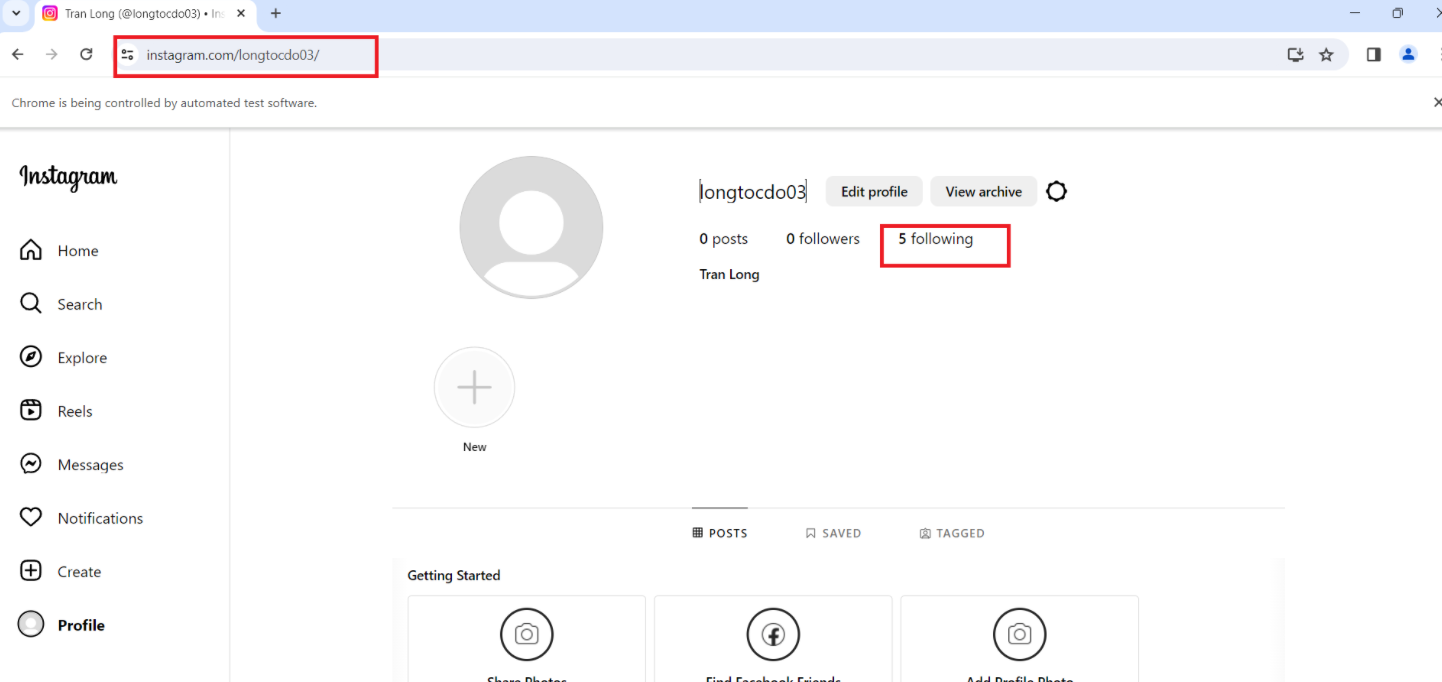
## Chức năng tìm kiếm

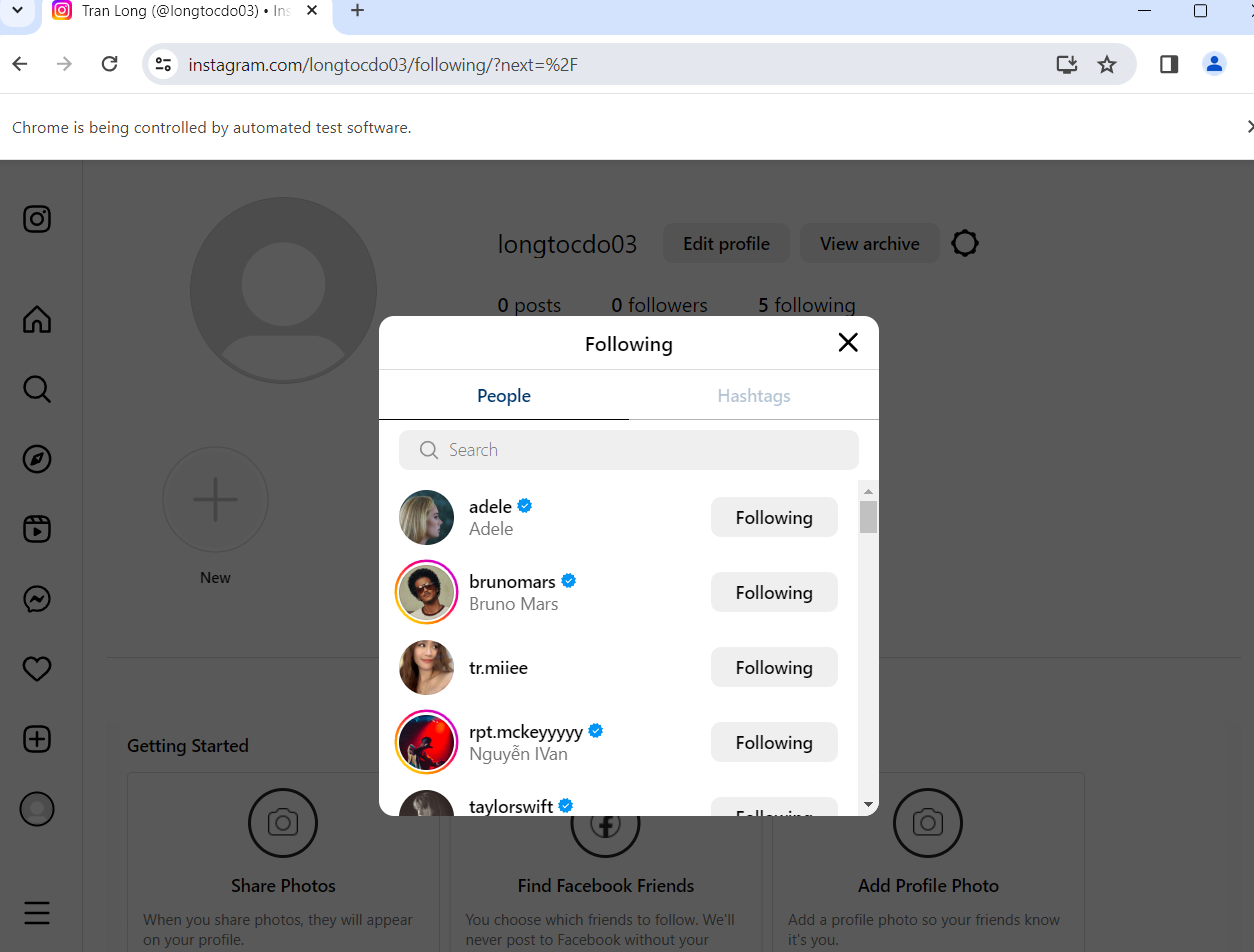
### TC1:tìm kiếm thành công



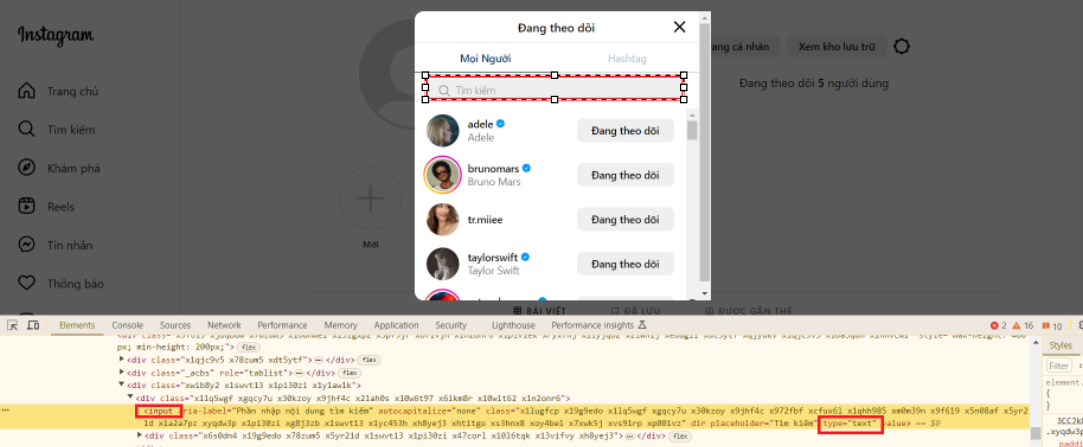


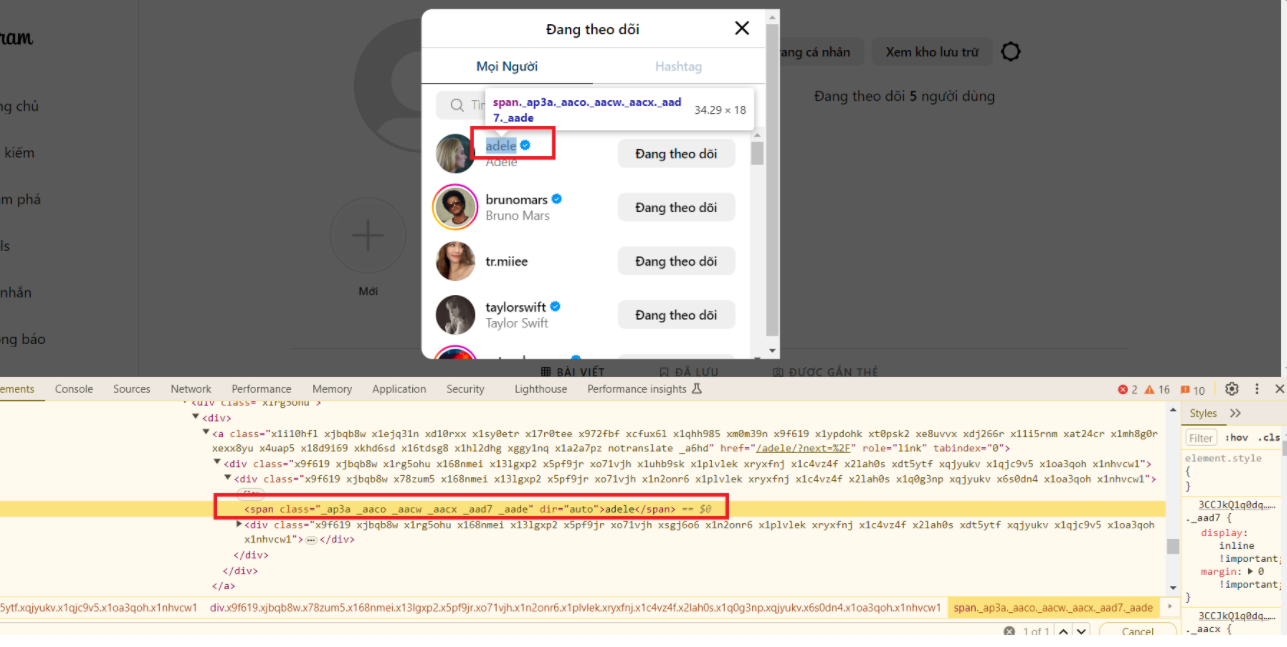
Giao diện trang cá nhân

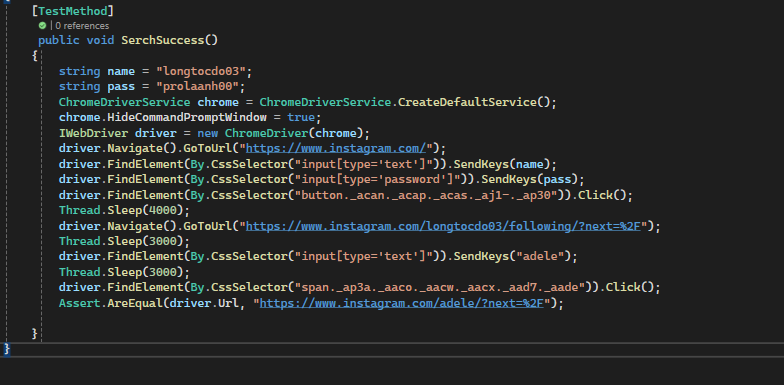
  
Ta truy cập vào danh sách following để tìm kiếm các accounts hiện đang following



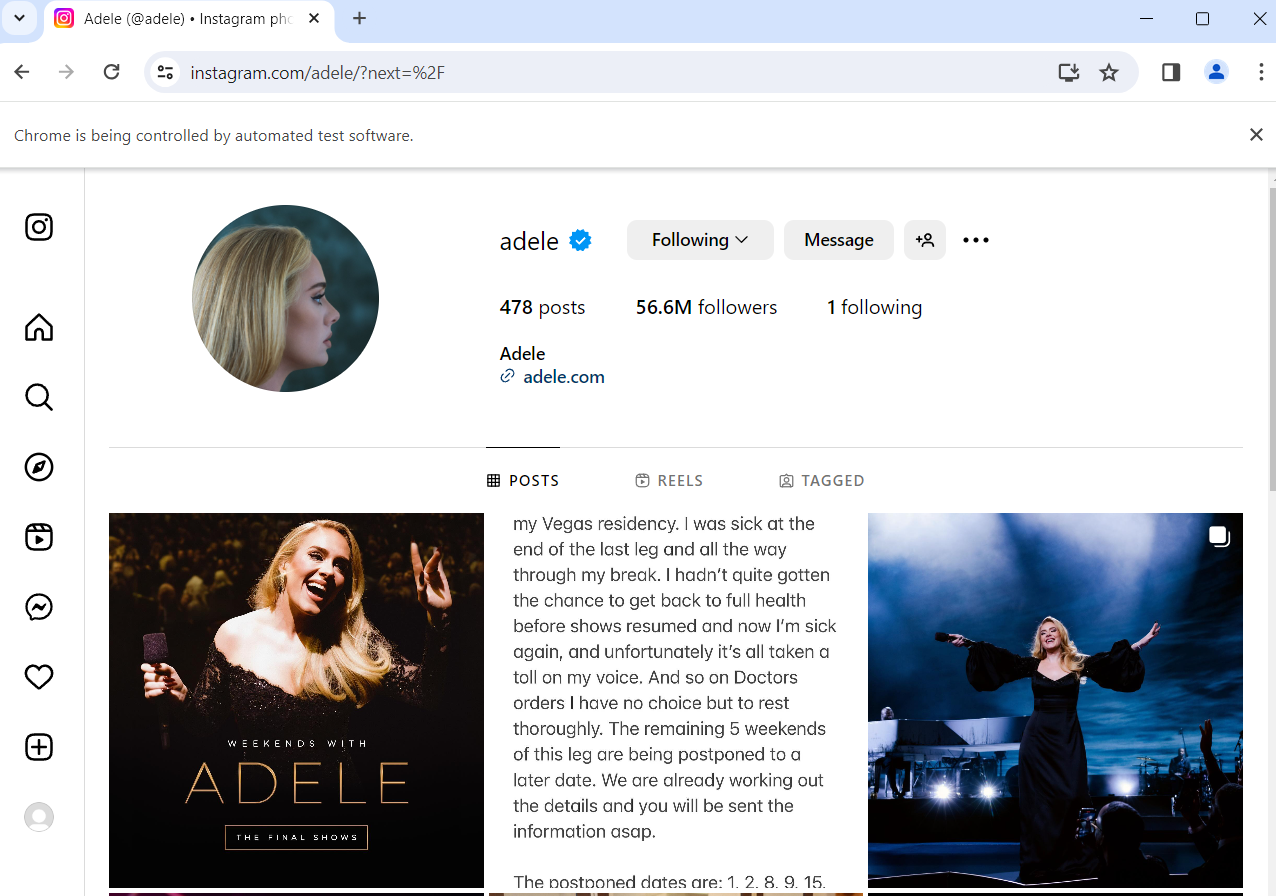
Ta kiểm tra thanh tìm kiếm:

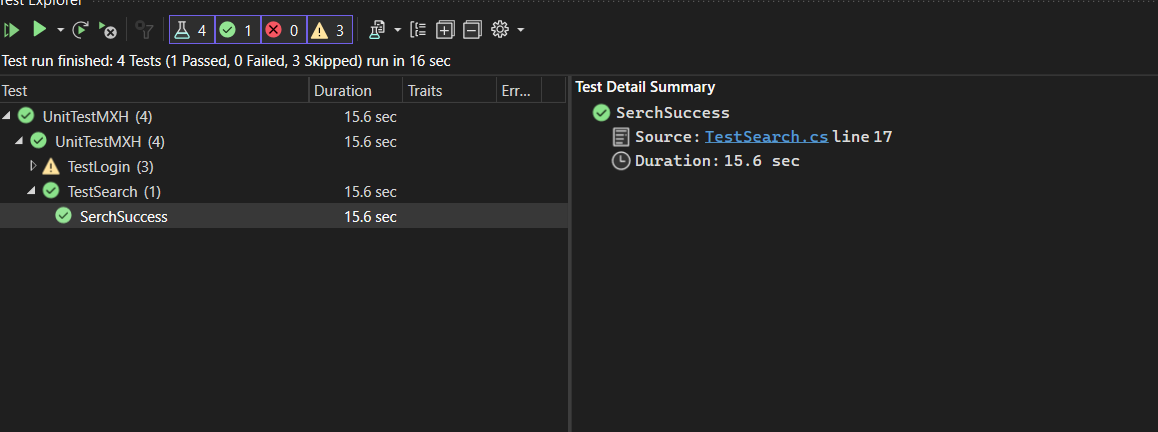
  
Lấy element truy vấn kết quả đầu tiên:



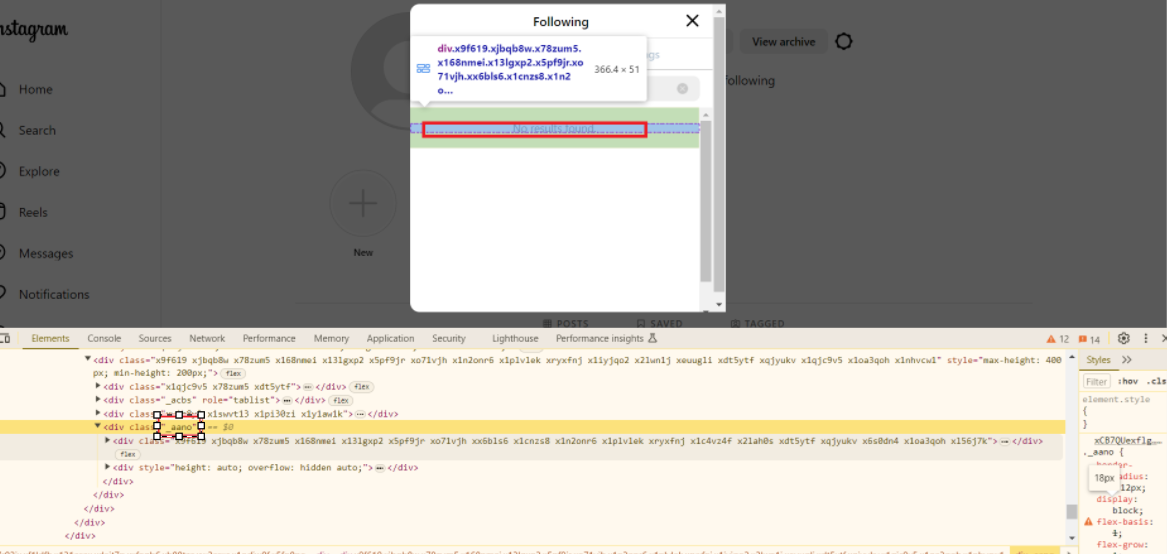
Code thực thi:  


Kết quả

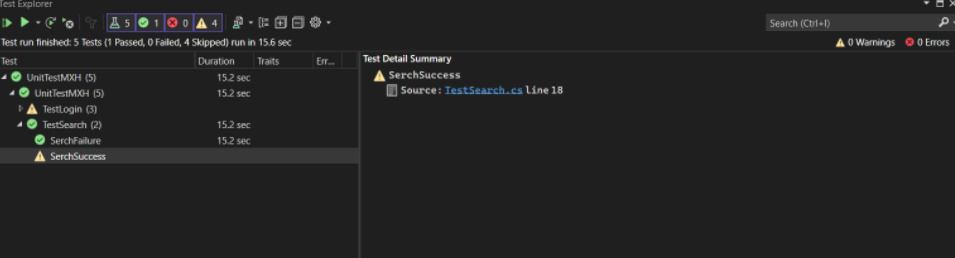




### TC2:Tìm kiếm thất bại

Tương tự như Tc1 nhưng khi truyền dữ liệu vào input ‘text’ thì truyền dữ liệu không đúng:  
 Sử dụng driver.FindElement(By.CssSelector("input[type='text']")).SendKeys(name);  
  
Tiếp đó ta bắt class hiển thị dòng chữ not found:  
driver.FindElement(By.ClassName("\_aano"));  


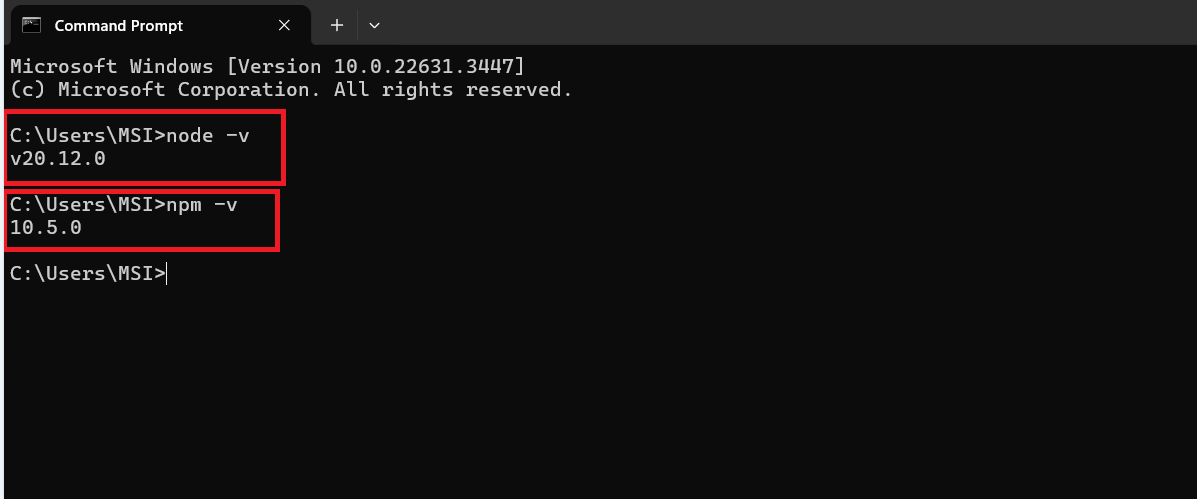
Code thực thi:  


Kết quả:  


# Kiểm thử API

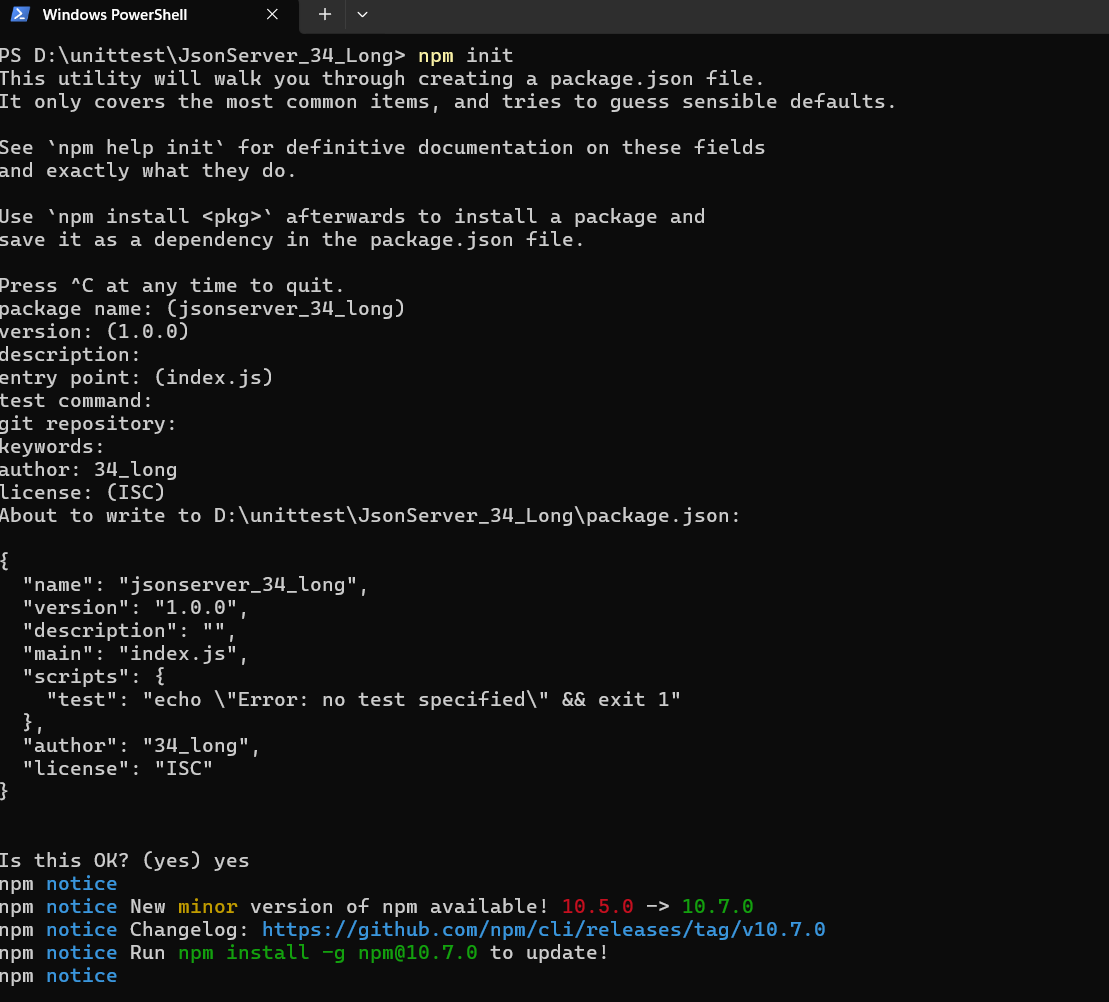
## Tạo JsonServer

Bước 1: Cài đặt node

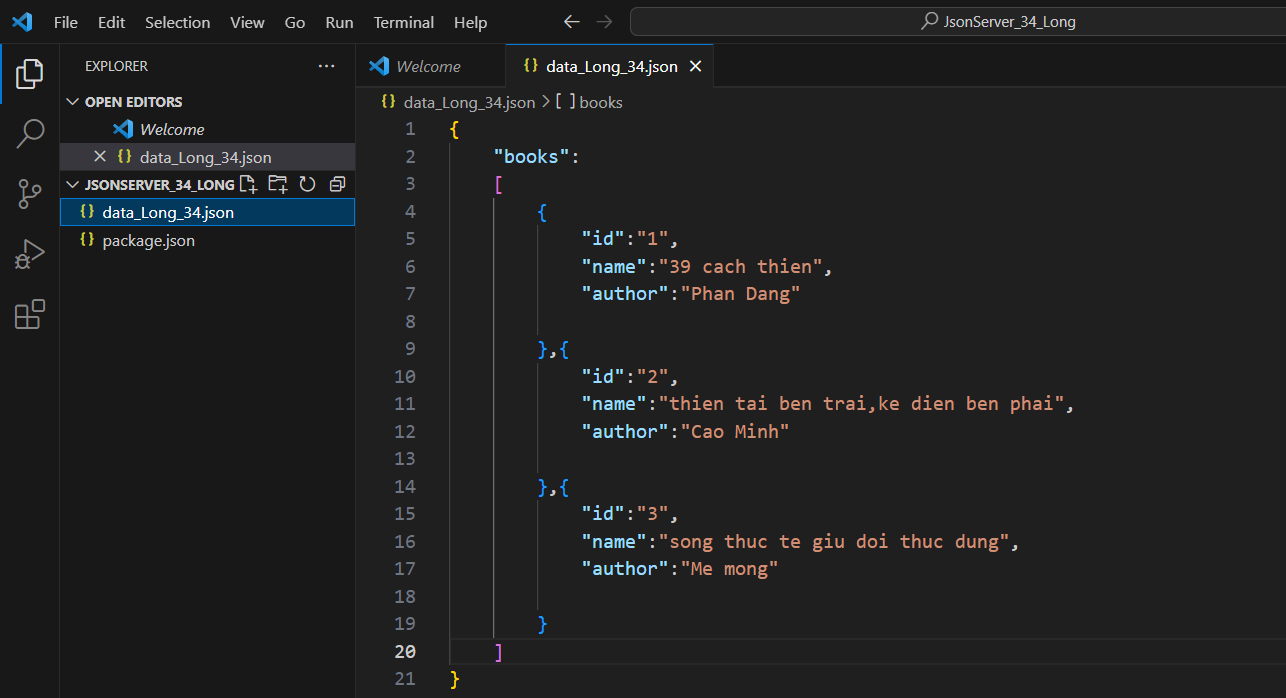


Bước 2: Tạo JsonServer trên thư mục

Tạo 1 thư mục JsonServer\_92\_Sang và chạy lệnh npm init và làm theo hướng dẫn.

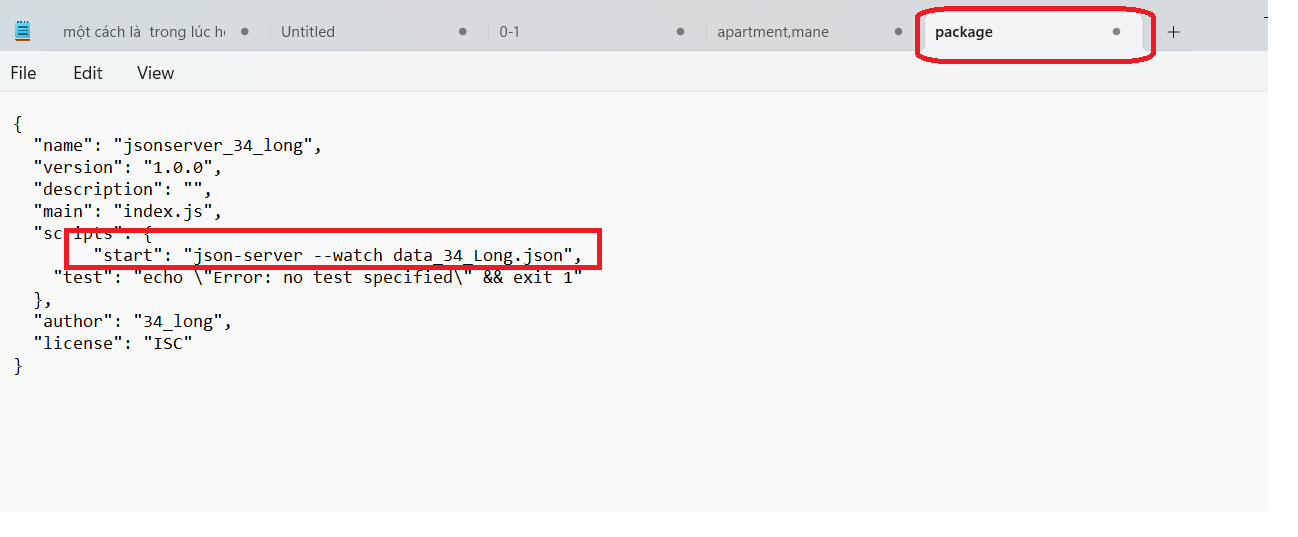


Bước 3: Tạo một file Json (data\_34\_Long.json ) trong thư mục JsonServer\_34\_Long



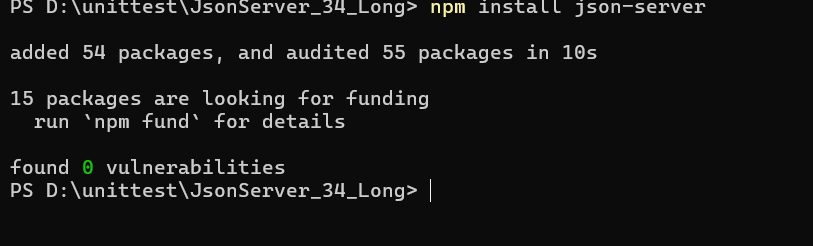
Bước 4: Mở file package.json trong thư mục JsonServer\_34\_Long

Thêm "start": "json-server --watch data\_34\_Long.json"



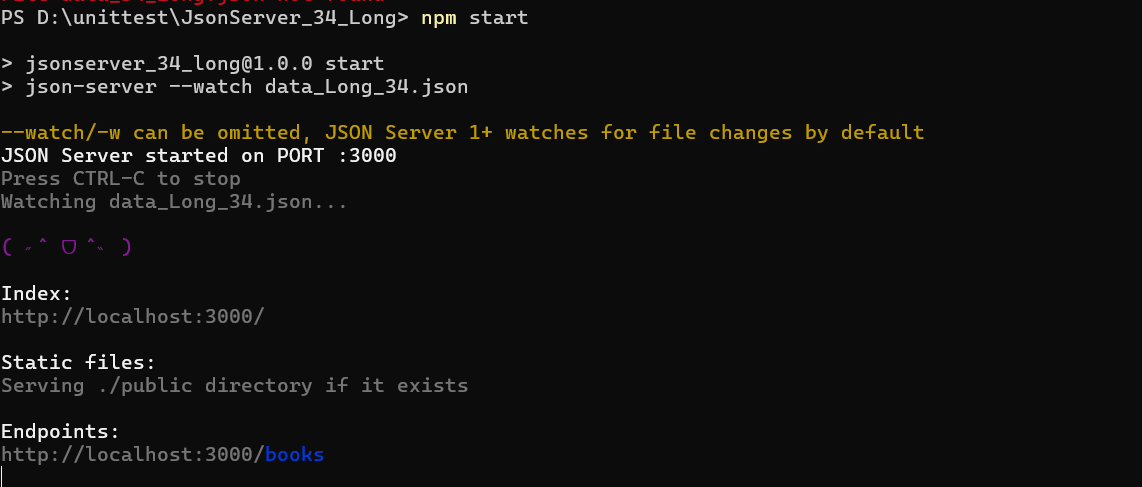
Bước 5: Cmd cài đặt json-server

Gõ lệnh npm install json-server



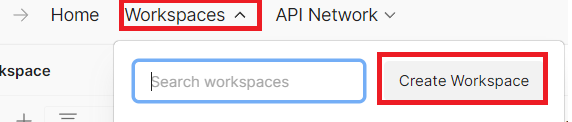
Bước 6: Chạy server

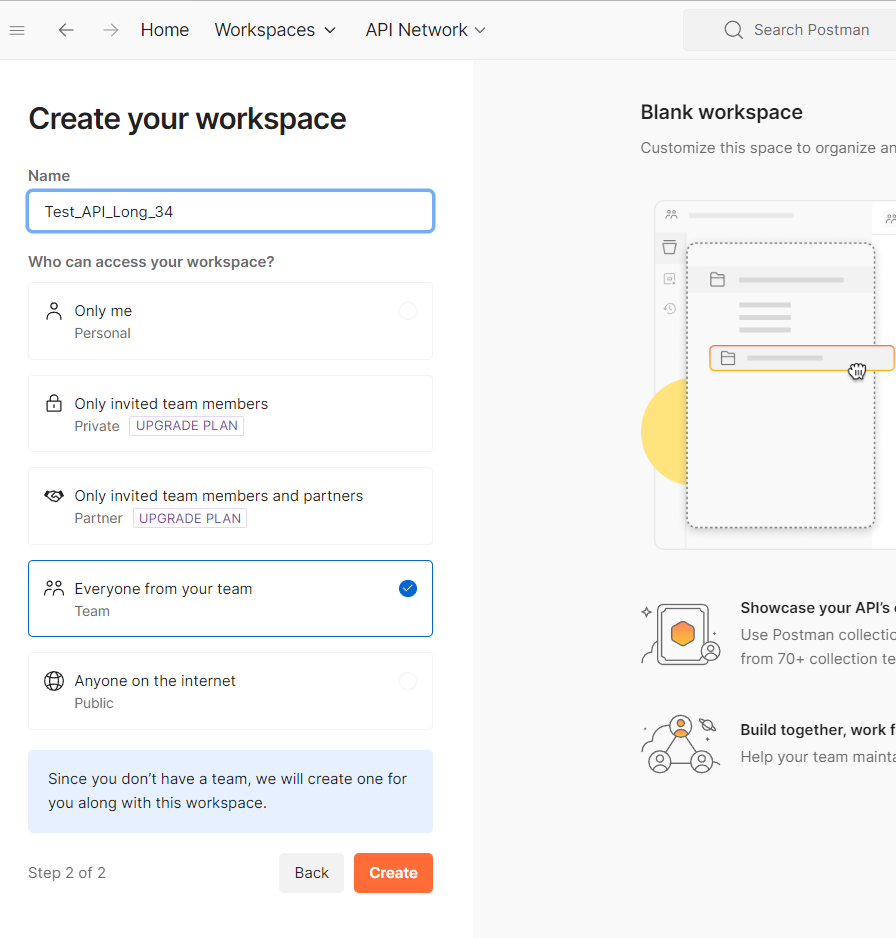
Gõ lệnh npm start



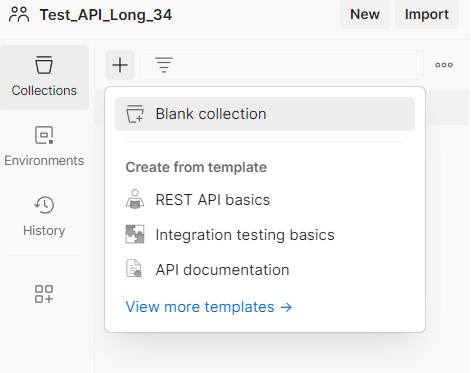
## Thiết lập Postman

bước 1: Tạo workspace

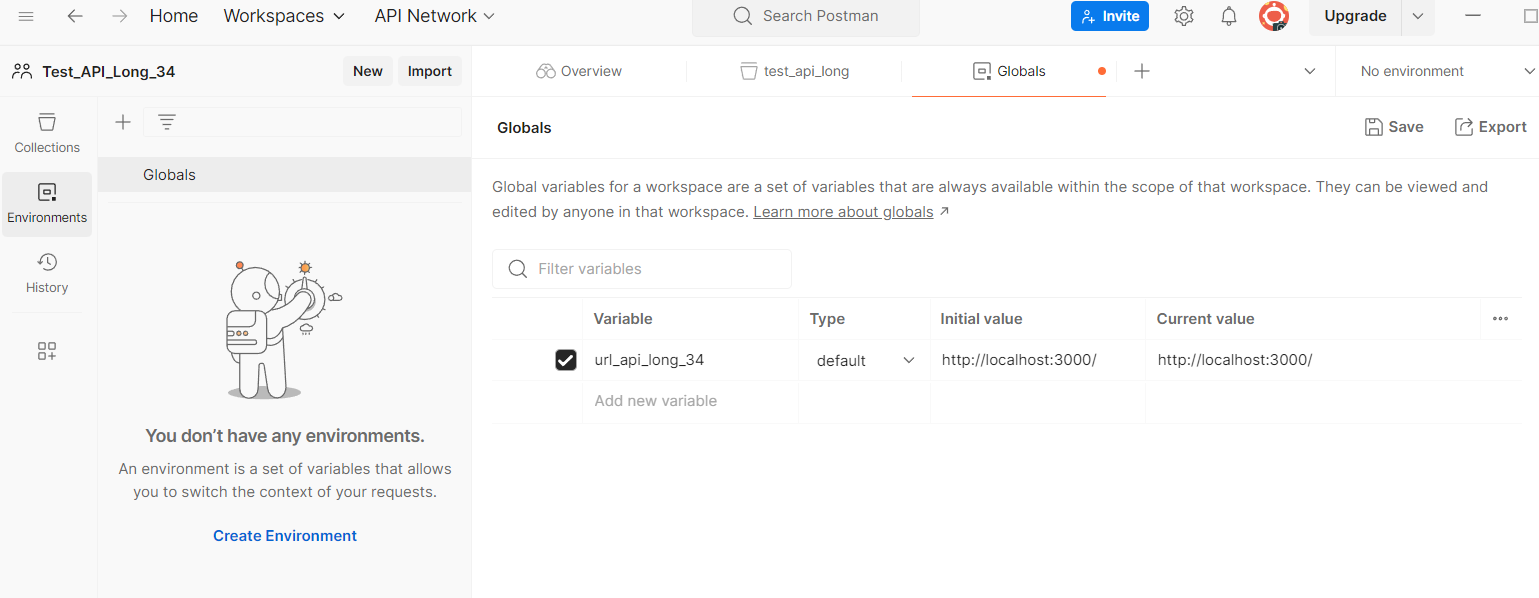




Bước 2: Tạo collection



Bước 3: Tạo biến global



Sử dụng bằng cách {{url\_api\_long\_34 }}

## Test Api và viết test case

4.3.1: Phương thức GET