

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

**BÁO CÁO BÀI TẬP LỚN KIỂM THỬ PHẦN MỀM**

***ĐỀ TÀI:* KIỂM THỬ WEBSITE DICTIONARY.COM**

**GVHD:** ThS.Nguyễn Thị Ngọc Thanh

**SINH VIÊN THỰC HIỆN:** Nguyễn Huỳnh Ngọc Tuyền

**NHÓM:** 1

**STT:** 72

**MSSV:** 2151050515

**LỚP:** DH21IT2101

**KHOA:** CÔNG NGHỆ THÔNG TIN

**Thành phố Hồ Chí Minh, 2024**

**LỜI NÓI ĐẦU**

Dictionary.com là một trang web từ điển trực tuyến cung cấp định nghĩa, từ đồng nghĩa, trái nghĩa, ví dụ sử dụng và các nguồn ngữ pháp cho nhiều từ tiếng Anh. Trang web này được cung cấp bởi [https://www.dictionary.com/](https://www.dictionary.com/" \t "_blank), một công ty công nghệ giáo dục.

Dictionary.com có nhiều tính năng hữu ích cho người học tiếng Anh, bao gồm:

* **Tra cứu từ điển:** Bạn có thể tra cứu từ tiếng Anh bằng cách nhập từ đó vào thanh tìm kiếm. Dictionary.com sẽ cung cấp cho bạn định nghĩa, từ đồng nghĩa, trái nghĩa, ví dụ sử dụng và các nguồn ngữ pháp cho từ đó.
* **Từ điển chuyên ngành:** Dictionary.com cung cấp nhiều từ điển chuyên ngành cho các lĩnh vực khác nhau, chẳng hạn như kinh doanh, y học, luật pháp và kỹ thuật.
* **Ngữ pháp:** Dictionary.com cung cấp các nguồn ngữ pháp toàn diện, bao gồm các bài giải thích về các quy tắc ngữ pháp, ví dụ và bài tập thực hành.
* **Học tập:** Dictionary.com cung cấp các công cụ học tập giúp bạn cải thiện vốn từ vựng và kỹ năng tiếng Anh của mình, chẳng hạn như thẻ ghi nhớ và trò chơi.
* **Ứng dụng:** Dictionary.com có ứng dụng dành cho thiết bị di động cho phép bạn tra cứu từ và học tiếng Anh mọi lúc mọi nơi.

Dictionary.com là một nguồn tài nguyên có giá trị cho người học tiếng Anh ở mọi trình độ. Trang web này cung cấp thông tin chính xác và cập nhật, đồng thời có nhiều tính năng hữu ích giúp bạn học tiếng Anh một cách hiệu quả.

Ngoài ra, Dictionary.com còn có một số tính năng độc đáo khác, chẳng hạn như:

* **Word of the Day:** Mỗi ngày, Dictionary.com sẽ chọn một từ tiếng Anh mới và cung cấp định nghĩa, từ đồng nghĩa, trái nghĩa, ví dụ sử dụng và các nguồn ngữ pháp cho từ đó.
* **Thesaurus:** Dictionary.com cung cấp từ điển đồng nghĩa giúp bạn tìm các từ có nghĩa tương tự hoặc liên quan đến từ bạn đang tra cứu.
* **Bảng phát âm:** Dictionary.com cung cấp bảng phát âm cho các từ tiếng Anh, giúp bạn học cách phát âm các từ đó một cách chính xác.
* **Ví dụ sử dụng:** Dictionary.com cung cấp ví dụ sử dụng cho các từ tiếng Anh, giúp bạn hiểu cách sử dụng các từ đó trong ngữ cảnh.

Dictionary.com là một trang web toàn diện và dễ sử dụng, cung cấp nhiều thông tin và công cụ hữu ích cho người học tiếng Anh.

**MỤC LỤC**

**[1. TỔNG QUAN VỀ SELENIUM: 1](#_Toc5367)**

[1.1. Selenium là gì? 1](#_Toc29222)

[1.2. Giới thiệu công cụ Selenium WebDriver: 1](#_Toc22488)

**[2. CÀI ĐẶT SELENIUM WEBDRIVER: 1](#_Toc19272)**

[2.1. Tạo một Project Windows Form C# 1](#_Toc21835)

[2.2. Tạo một Project kiểm thử Unit Test 2](#_Toc3683)

[2.3. Cài đặt Selenium Web Server 4](#_Toc30051)

[2.4. Tạo bộ Data Test 5](#_Toc632)

**[3. KIỂM THỬ CHỨC NĂNG LOGIN: 7](#_Toc27182)**

[3.1. Mô tả chức năng Login: 7](#_Toc18076)

[3.2. Thực hiện test chức năng Login của trang Dictionary.com: 8](#_Toc30034)

[3.2.1. Bước 1: Bắt ClassName object Login để vào trang Login: Click phải chọn inspect 8](#_Toc24864)

[3.2.2. Bước 2: Bắt Id object Email, Password: 9](#_Toc10776)

[3.2.3. Bước 3: Bắt XPath button Login 10](#_Toc3450)

[3.3. Testcase01: TestSuccessfulLogin\_72\_Tuyen (Đăng nhập thành công): 10](#_Toc15910)

[Hình ảnh website khi đăng nhập thành công: 10](#_Toc6809)

[3.3.1. Bước 1: Viết code cho testcase TestSuccessfulLogin\_72\_Tuyen dựa trên các trường Login, Email, Password, button Login đã bắt ở trên: 10](#_Toc32012)

[3.3.2. Bước 2: Thực thi Testcase và xem kết quả: 12](#_Toc25792)

[3.4. Testcase02: TestFailedLogin\_NoEmail\_72\_Tuyen (Không điền trường email): 12](#_Toc23846)

[Hình ảnh website đăng nhập không thành công vì không nhập trường email: 12](#_Toc3936)

[3.4.1. Bước 1: Viết code cho testcase TestFailedLogin\_NoEmail\_72\_Tuyen dựa trên các trường Login, Email, Password, button Login đã bắt ở trên: 12](#_Toc26784)

[3.4.2. Bước 2: Thực thi Testcase và xem kết quả: 14](#_Toc1666)

[3.5. Testcase03: TestFailedLogin\_NoPassword\_72\_Tuyen (Không điền trường Password): 14](#_Toc937)

[Hình ảnh website đăng nhập không thành công vì không nhập trường password: 14](#_Toc19647)

[3.5.1. Bước 1: Viết code cho testcase 14](#_Toc8460)

[3.5.2. Bước 2: Thực thi Testcase và xem kết quả: 16](#_Toc485)

[3.6. Testcase04: TestFailedLogin\_WrongPasswordorLogin\_72\_Tuyen 16](#_Toc16589)

[Hình ảnh website đăng nhập không thành công vì không nhập trường password: 16](#_Toc12202)

[3.6.1. Bước 1: Viết code cho testcase 16](#_Toc12417)

[3.6.2. Bước 2: Thực thi Testcase và xem kết quả: 17](#_Toc5414)

[3.7. Kết quả sau khi chạy hết 4 Testcase: 18](#_Toc876)

**[4. KIỂM THỬ CHỨC NĂNG TÌM KIẾM (SEARCH): 18](#_Toc12008)**

[4.1. Mô tả chức năng tìm kiếm: 18](#_Toc10428)

[4.2. Thực hiện test chức năng tìm kiếm của trang Dictionary.com 18](#_Toc26846)

[4.3. Testcase01: TestSuccessfulSearch\_72\_Tuyen(Tìm kiếm thành công): 21](#_Toc30796)

[Hình ảnh website tìm kiếm thành công: 21](#_Toc4211)

[4.3.1. Bước 1: Viết code cho testcase TestSuccessfulSearch\_72\_Tuyen: 21](#_Toc4826)

[4.3.2. Bước 2: Thực hiện testcase và xem kết quả: 22](#_Toc4228)

[4.4. Testcase02: TestFailSearch\_72\_Tuyen (Tìm kiếm không thành công): 22](#_Toc6294)

[Hình ảnh website sau khi tìm kiếm không thành công: 23](#_Toc3616)

[4.4.1. Bước 1: Viết code cho testcase TestFailSearch\_72\_Tuyen: 23](#_Toc4462)

[4.4.2. Bước 2: Thực hiện testcase và xem kết quả: 24](#_Toc29747)

[4.5. Kết quả sau khi chạy 2 testcase: 24](#_Toc31882)

**[5. KẾT QUẢ SAU KHI CHẠY 6 TESTCASE: 24](#_Toc11650)**

**[6. API TESTING: 25](#_Toc26833)**

[6.1. API là gì? 25](#_Toc1965)

[6.2. Vì sao phải Test API? 25](#_Toc18032)

[6.3. Các tool cho kiểm thử API: 25](#_Toc25017)

[6.4. Các phương thức cơ bản: 25](#_Toc29345)

Version Chrome: 124.0.6367.92

Version Selenium.WebDriver.ChromeDrive: 124.0.6367.9100

**DANH MỤC HÌNH ẢNH**

[Hình 1 . Tạo Project Windows Form C# 6](#_Toc14437)

[Hình 2 . Đặt tên và tạo Project 6](#_Toc32062)

[Hình 3 . Tạo Project kiểm thử 7](#_Toc30740)

[Hình 4 . Tạo Unit Test Project (.NET Framework) 7](#_Toc12107)

[Hình 5 . Mở hộp thoại Manage NuGet Packages.. 8](#_Toc29638)

[Hình 6 . Cài đặt Selenium.WebDriver 8](#_Toc4253)

[Hình 7 . Cài đặt Selenium.WebDriver.ChromeDriver 9](#_Toc21858)

[Hình 8 . Tạo tập tin TestLogin.csv 9](#_Toc14198)

[Hình 9 . Thiết lập thuộc tính “Copy Always” 10](#_Toc27471)

[Hình 10 . Nhập dữ liệu vào tệp tin 10](#_Toc7320)

[Hình 11 . Thêm reference System.Data 10](#_Toc25738)

[Hình 12 . Hình ảnh website Dictionary.com 11](#_Toc22525)

[Hình 13 . Bắt Id object Login 11](#_Toc4854)

[Hình 14 . Kết quả sau khi bắt ClassName object Login 11](#_Toc5575)

[Hình 15 . Bắt Id object Email 12](#_Toc10228)

[Hình 16 . Kết quả sau khi bắt Id object Login 12](#_Toc22394)

[Hình 17 . Bắt Id object Password 12](#_Toc7988)

[Hình 18 . Kết quả sau khi bắt Id object Password 13](#_Toc21865)

[Hình 19 . Bắt XPath button Login 13](#_Toc19403)

[Hình 20 . Kết quả sau khi bắt Id object Password 13](#_Toc5230)

[Hình 21 . Ảnh code của testcase TestSuccessfulLogin\_72\_Tuyen 15](#_Toc15237)

[Hình 22 . Kết quả sau khi chạy Testcase TestSuccessfulLogin\_72\_Tuyen 15](#_Toc990)

[Hình 23 . Ảnh code của testcase TestFailedLogin\_NoEmail\_72\_Tuyen 16](#_Toc25724)

[Hình 24 . Kết quả sau khi chạy testcase TestFailedLogin\_NoEmail\_72\_Tuyen 17](#_Toc14268)

[Hình 25 . Ảnh code của testcase TestFailedLogin\_NoPassword\_72\_Tuyen 18](#_Toc20713)

[Hình 26 . Kết quả của testcase TestFailedLogin\_NoPassword\_72\_Tuyen 19](#_Toc4909)

[Hình 27 .Ảnh code của testcase TestFailedLogin\_NoPassword\_72\_Tuyen 20](#_Toc15675)

[Hình 28 . Kết quả của testcase TestFailedLogin\_NoPassword\_72\_Tuyen 21](#_Toc30508)

[Hình 29 . Kết quả sau khi thực thi hết 4 Testcase 21](#_Toc23654)

[Hình 30 . Tạo một project Unit Test mới 22](#_Toc18815)

[Hình 31 . Bắt CssSelector object thanh tìm kiếm 22](#_Toc13950)

[Hình 32 . Kết quả sau khi bắt CssSelector object thanh tìm kiếm 22](#_Toc26602)

[Hình 33 . Bắt ClassName button tìm kiếm 23](#_Toc21892)

[Hình 34 . Kết quả sau khi bắt ClassName button tìm kiếm 23](#_Toc23458)

[Hình 35 . Bắt sự kiện thông báo tìm kiếm fail 23](#_Toc13440)

[Hình 36 . Kết quả sau khi bắt sự kiện tìm kiếm fail 23](#_Toc24836)

[Hình 37 . Bắt sự kiện thông báo tìm kiếm thành công 23](#_Toc9423)

[Hình 38 . Kết quả sau khi bắt sự kiện tìm kiếm thành công 24](#_Toc6023)

[Hình 39 . Ảnh code của testcase TestSuccessfulSearch\_72\_Tuyen 25](#_Toc7615)

[Hình 40 . Kết quả sau khi chạy Testcase 25](#_Toc28144)

[Hình 41 . Hình ảnh website khi tìm kiếm không thành công 26](#_Toc10841)

[Hình 42 . Ảnh code của Testcasse TestFailSearch\_72\_Tuyen 27](#_Toc7877)

[Hình 43 . Kết quả sau khi chạy Testcase 27](#_Toc27006)

[Hình 44 . Kết quả sau khi chạy 2 Testcase tìm kiếm 27](#_Toc20739)

[Hình 45 . Kết quả sau khi chạy hết 6 Testcase 28](#_Toc21558)

# TỔNG QUAN VỀ SELENIUM:

## Selenium là gì?

Selenium là công cụ mã nguồn mở dùng cho kiểm thử tự động các ứng dụng web được phát triển bởi ThoughWorks năm 2004. Selenium chỉ tập trung vào việc tự động hóa các ứng dụng dựa trên web. Việc kiểm thử được sử dụng bằng công cụ Selenium thường được gọi là Selenium Testing. Selenium có thể hoạt động trên nhiều trình duyệt và hệ điều hành khác nhau. Các công cụ của Selenium:

* Selenium IDE
* Selenium RC
* Selenium WebDriver
* Selenium Grid

## Giới thiệu công cụ Selenium WebDriver:

Selenium WebDriver là một khuôn khổ tự động hóa web cho phép bạn thực hiện các kiểm thử của mình trên các trình duyệt khác nhau. Nó nằm trong bộ kiểm thử tự động Selenium. Selenium WebDriver cho phép gửi các lệnh trực tiếp lên trình duyệt và tìm kiếm kết quả.

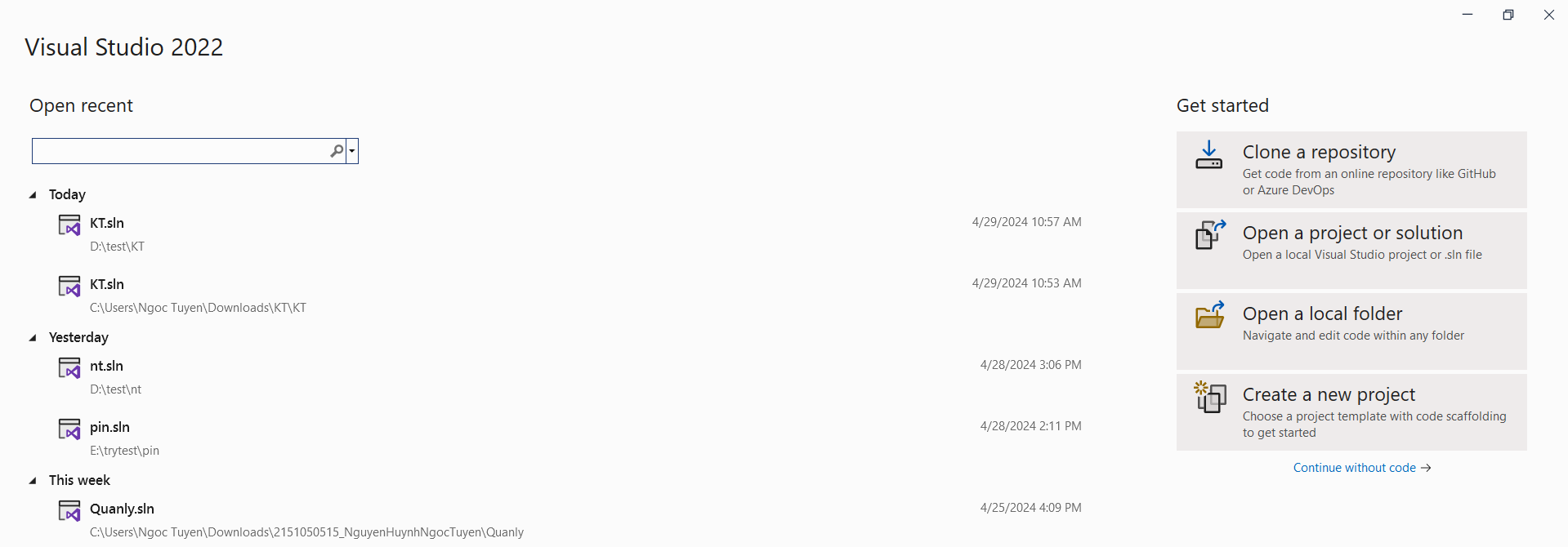
**Lý do chọn Selenium WebDriver:**

* Hỗ trợ nhiều ngôn ngữ lập trình như: Java, Python, C#, Ruby, JavaScript,...
* Hỗ trợ nhiều trình duyệt như: Chrome, Firefox, Safari, Edge và nhiều hệ điều hành khác nhau như Windows, macOS, Linux,...
* Tự động hóa các thao tác trình duyệt: Selenium cho phép tự động hóa các thao tác trình duyệt như điều hướng, click, nhập liệu, và kiểm tra các phần tử trên trang web.

# CÀI ĐẶT SELENIUM WEBDRIVER:

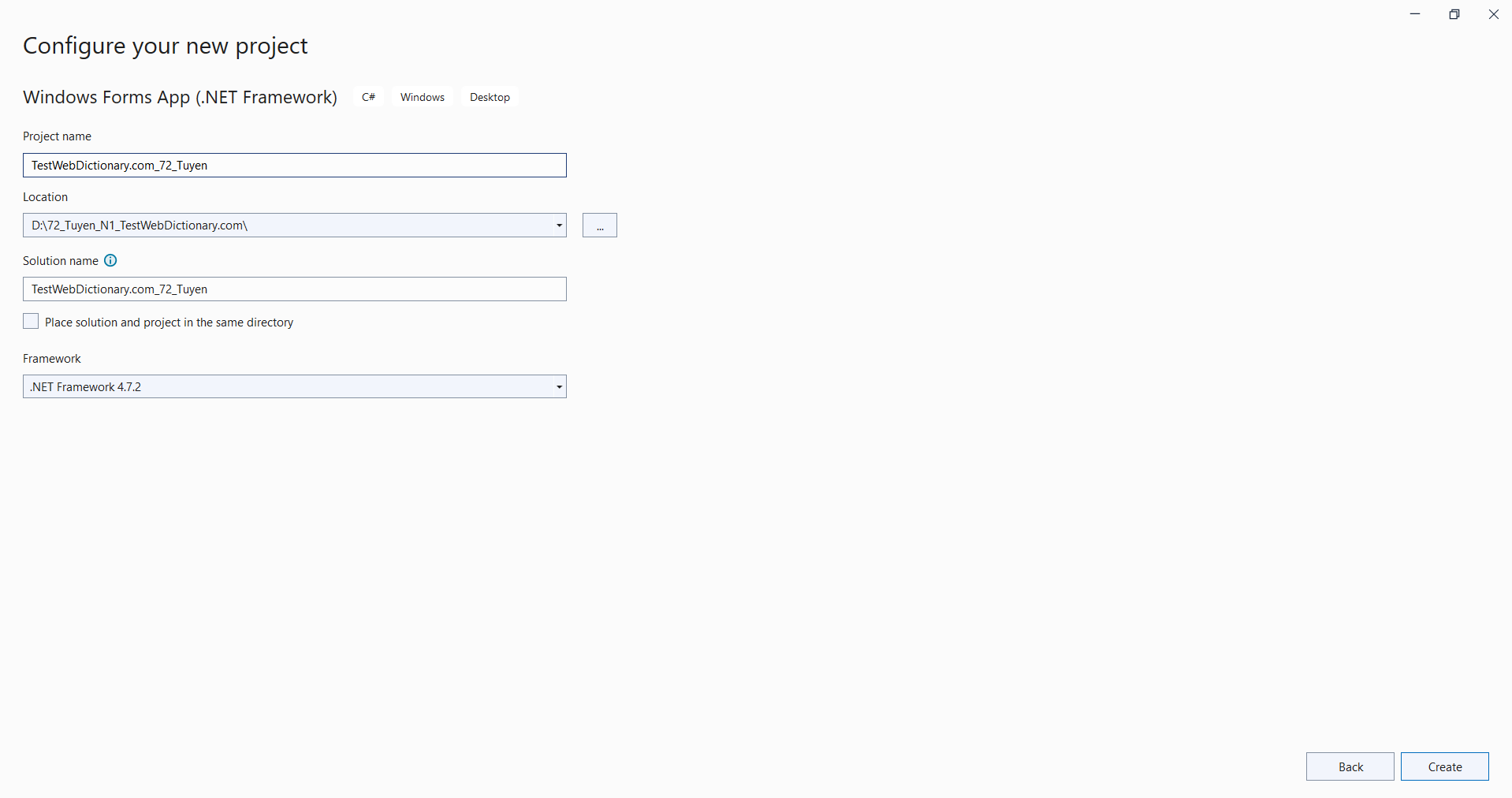
## Tạo một Project Windows Form C#

**Bước 1:** Mở Microsoft Visual Studio 2022 -> Chọn Create a new project -> Chọn Windows Forms App(.NET Framework) -> Next



Hình 1. Tạo Project Windows Form C#

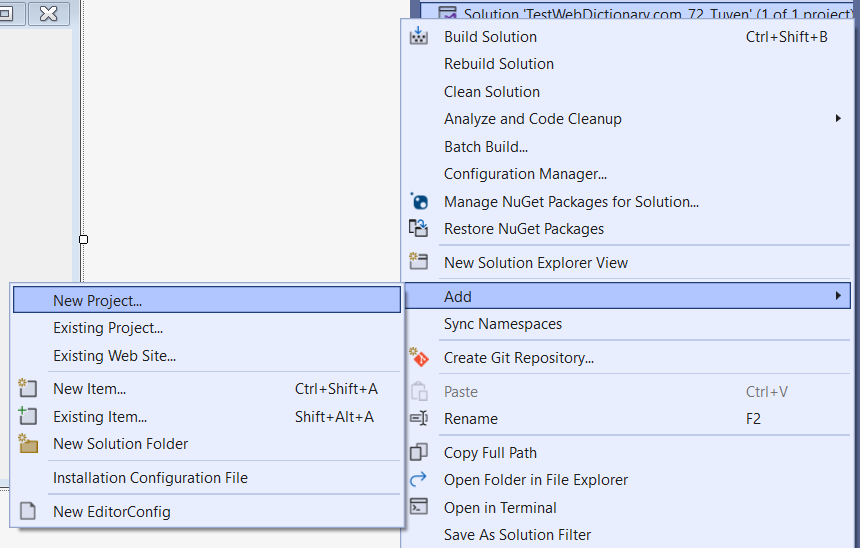
**Bước 2:** Đặt tên cho Project



Hình 2. Đặt tên và tạo Project

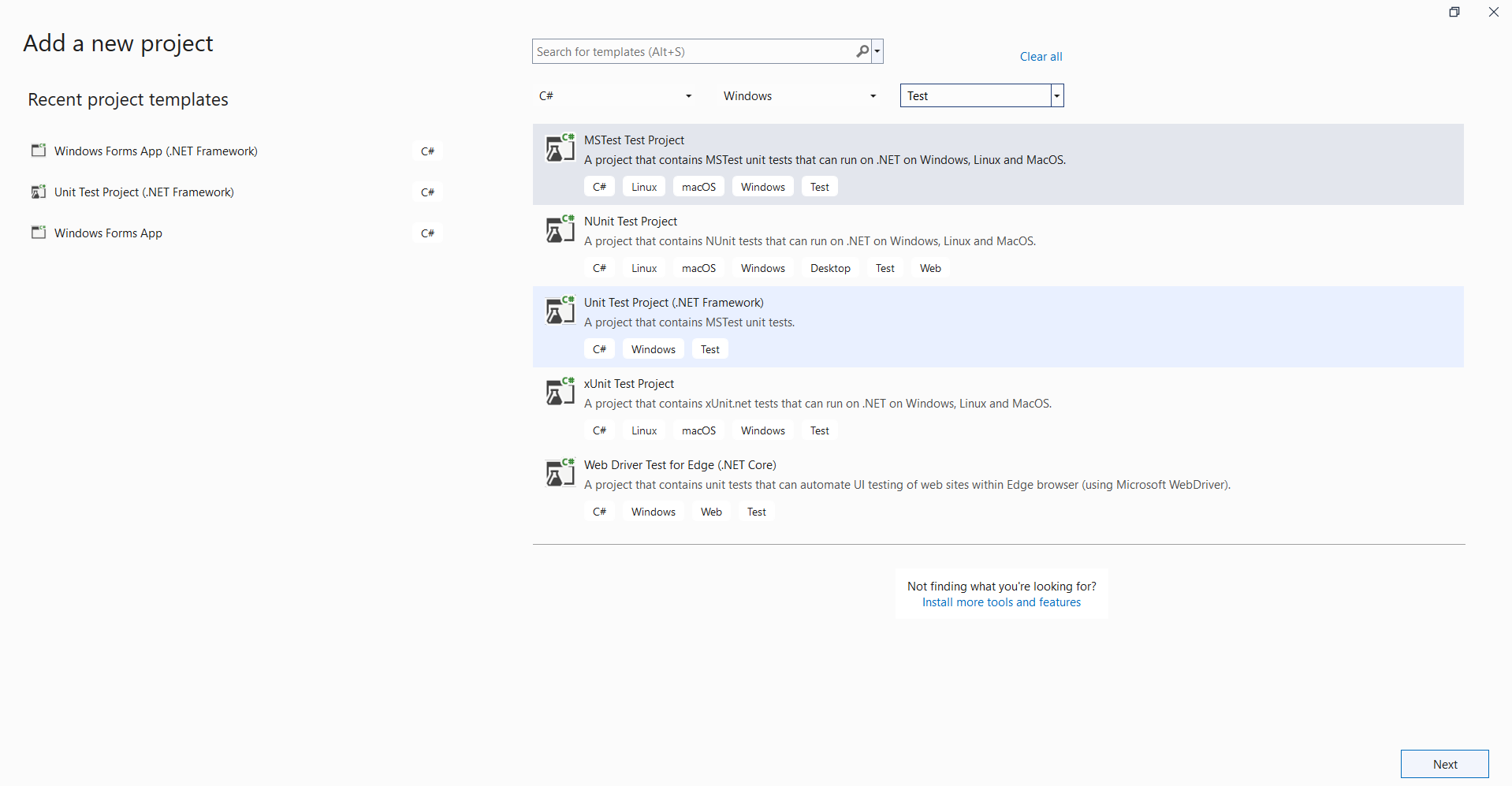
## Tạo một Project kiểm thử Unit Test

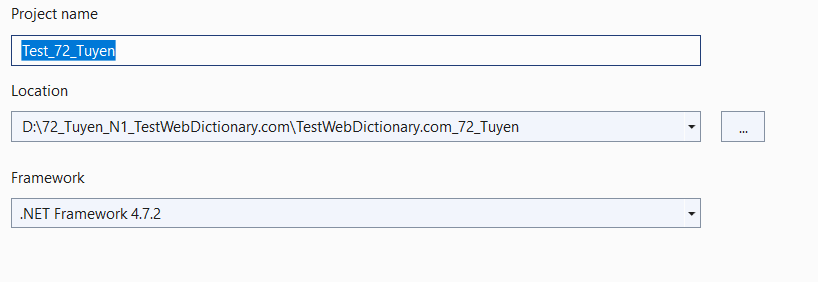
**Bước 1:** Click phải vào Solution chọn Add => New project



Hình 3. Tạo Project kiểm thử

**Bước 2:** Chọn ngôn ngữ C# -> Chọn kiểu Test -> Chọn Unit Test Project (.NET Framework) -> Next -> Đặt tên cho project test là (Test\_72\_Tuyen).

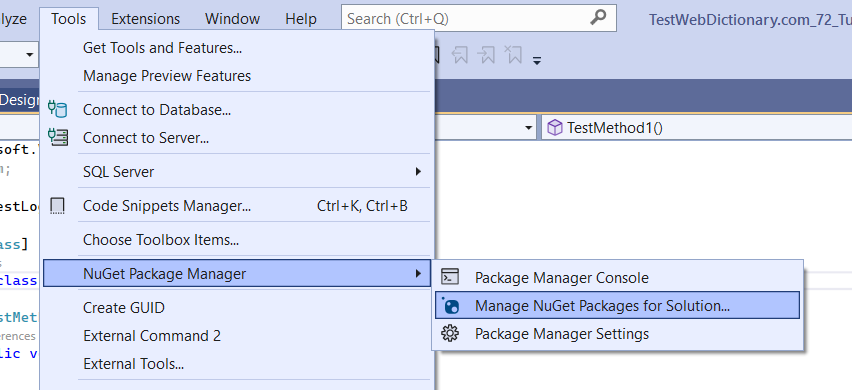




Hình 4. Tạo Unit Test Project (.NET Framework)

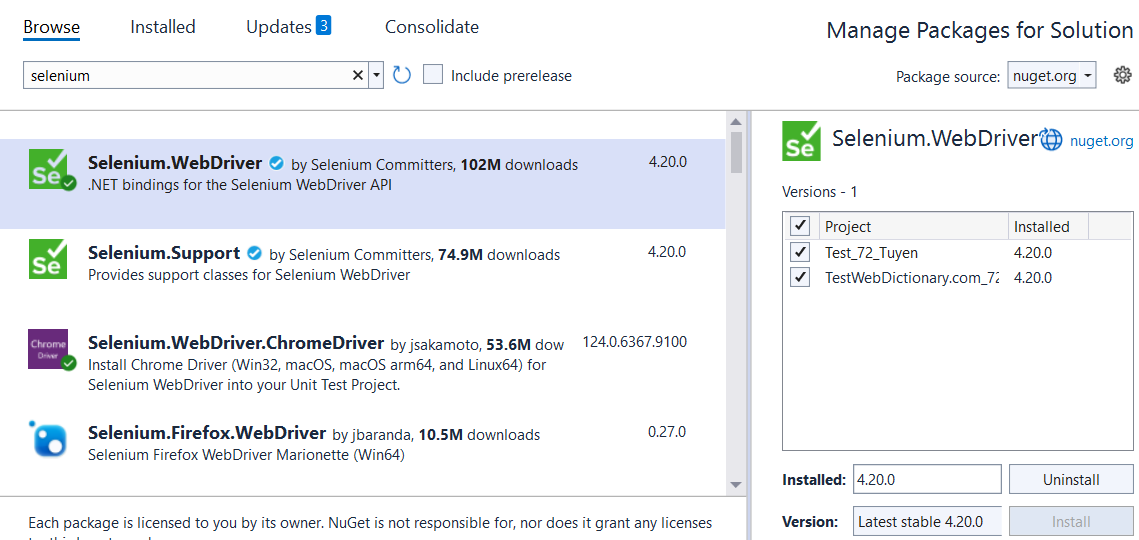
## Cài đặt Selenium Web Server

**Bước 1:** Tools => Nuget Pakage Manager => Manage Nuget Pakages for Solution…



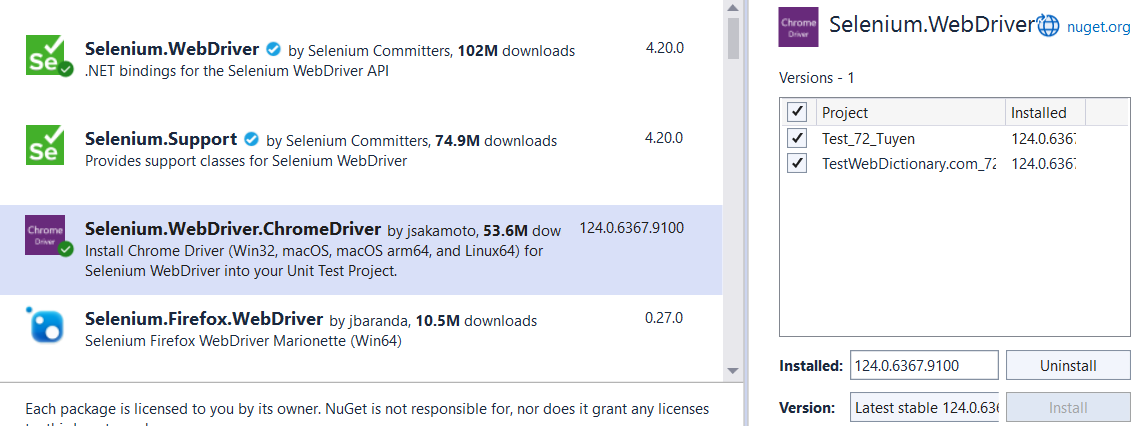
Hình 5. Mở hộp thoại Manage NuGet Packages..

**Bước 2:** Chọn Browse -> Tìm kiếm từ khóa: Selenium -> Chọn và cài đặt Selenium.WebDriver -> Chọn Project -> Install



Hình 6. Cài đặt Selenium.WebDriver

**Bước 3:** Chọn Browse -> Tìm kiếm từ khóa: Selenium -> Chọn và cài đặt Selenium.WebDriver -> Chọn Project -> Install

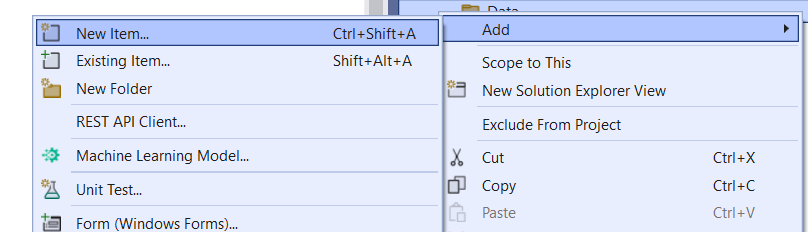


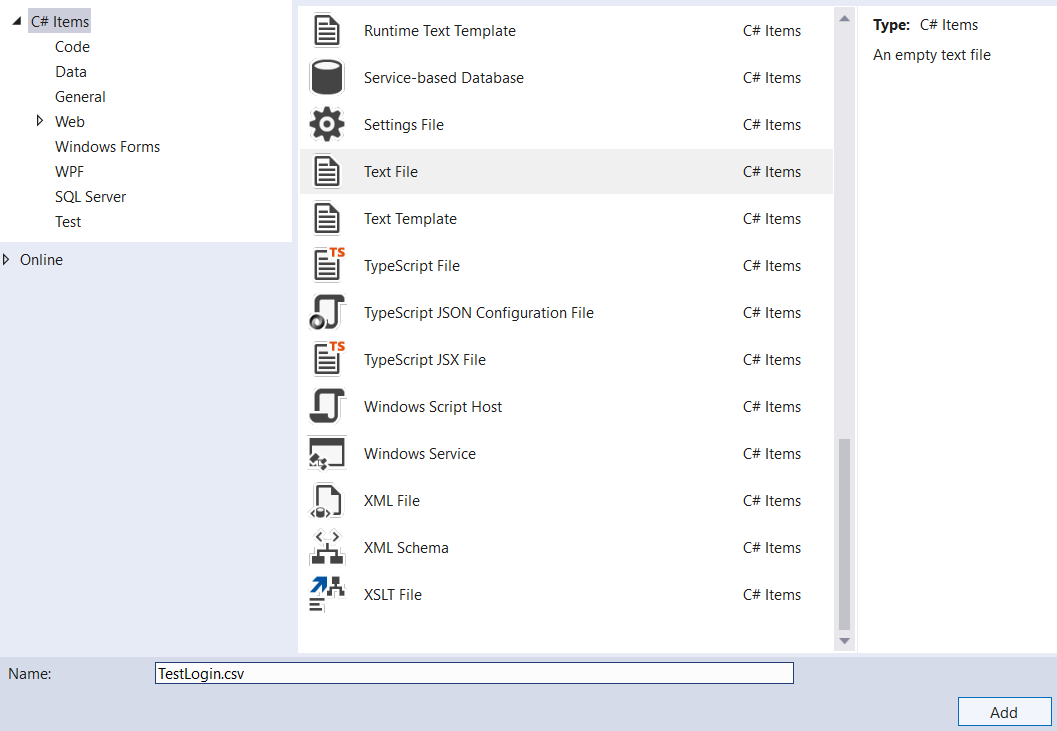
Hình 7. Cài đặt Selenium.WebDriver.ChromeDriver

## Tạo bộ Data Test

**Bước 1:** Tạo thư mục Data trong project Test

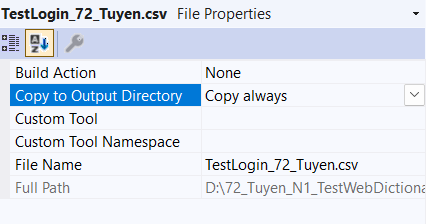
**Bước 2:** Click chuột phải vào thư mục Data => New item => tạo tập tin TestLogin.csv





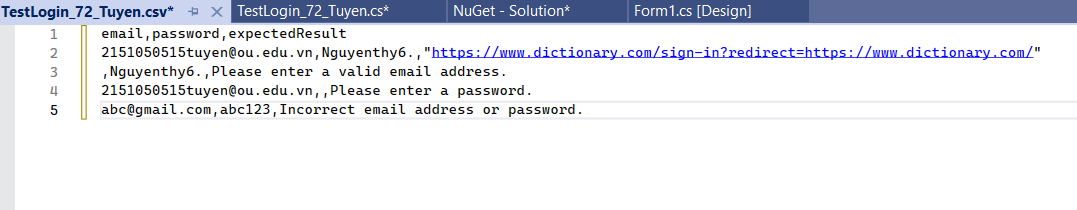
Hình 8. Tạo tập tin TestLogin.csv

**Bước 3:** Chuột phải vào tập tin DataTestLogin.csv chọn Properties, thiết lập thuộc tính “Copy to Output Directory” thành “Copy Always” để tập tin này sẽ được sao chép vào thư mục bin khi build project



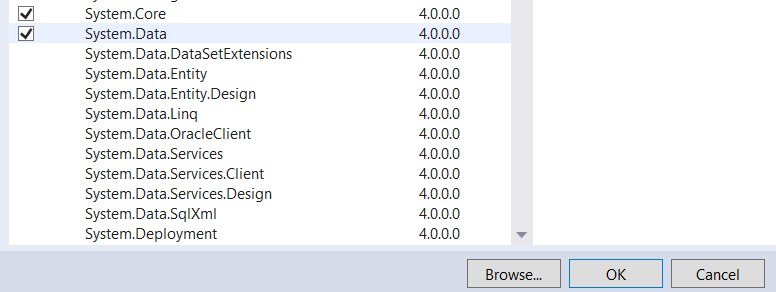
Hình 9. Thiết lập thuộc tính “Copy Always”

**Bước 4:** Nhập dữ liệu vào tệp tin TestLogin\_72\_Tuyen.csv:



Hình 10. Nhập dữ liệu vào tệp tin

**Bước 5:** Thêm reference System.Data vào project Test\_72\_Tuyen



Hình 11. Thêm reference System.Data

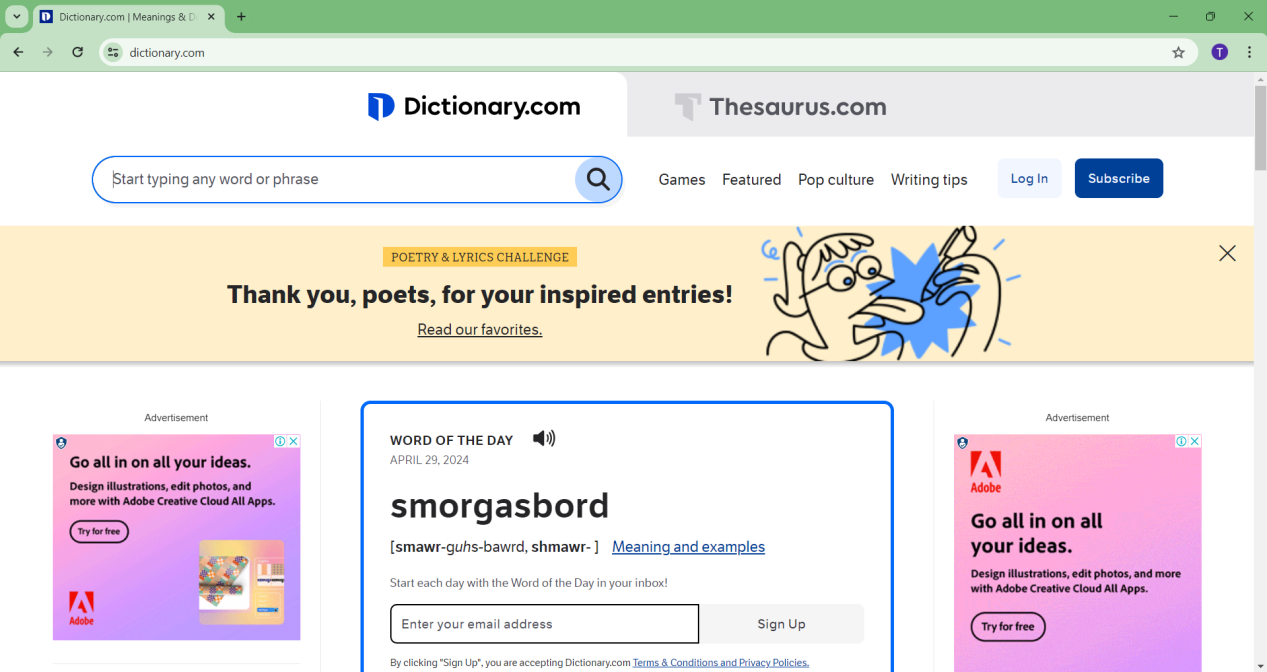
# KIỂM THỬ CHỨC NĂNG LOGIN:

## Mô tả chức năng Login:

Chức năng đăng nhập cho phép người dùng xác thực danh tính và truy cập vào hệ thống hoặc ứng dụng. Người dùng cung cấp thông tin đăng nhập, bao gồm tên người dùng và mật khẩu. Hệ thống xác thực thông tin và cấp quyền truy cập tương ứng. Việc duy trì phiên làm việc giúp người dùng tiếp tục truy cập mà không cần đăng nhập lại. Chức năng này đảm bảo tính bảo mật và quản lý truy cập hiệu quả.

Ở đây em sẽ kiểm tra chức năng Login của trang **Dictionary.com**. Dictionary.com là một trang web từ điển và ngôn ngữ trực tuyến được cung cấp bởi Lexico Publishing Group, LLC, một công ty con thuộc sở hữu của IAC. Nó cung cấp các định nghĩa, từ đồng nghĩa, trái nghĩa, ví dụ và nhiều thông tin ngôn ngữ khác.

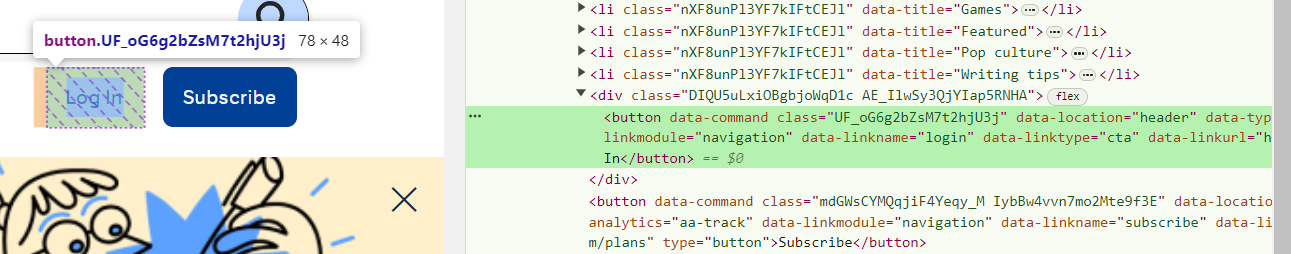
Đây là hình ảnh của website:



Hình 12. Hình ảnh website Dictionary.com

## Thực hiện test chức năng Login của trang Dictionary.com:

### **Bước 1:** Bắt ClassName object Login để vào trang Login: Click phải chọn inspect



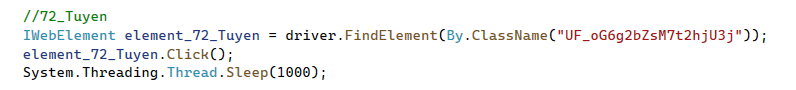
Hình 13. Bắt Id object Login

Sau khi bắt chúng ta sẽ thu được đoạn code sau:

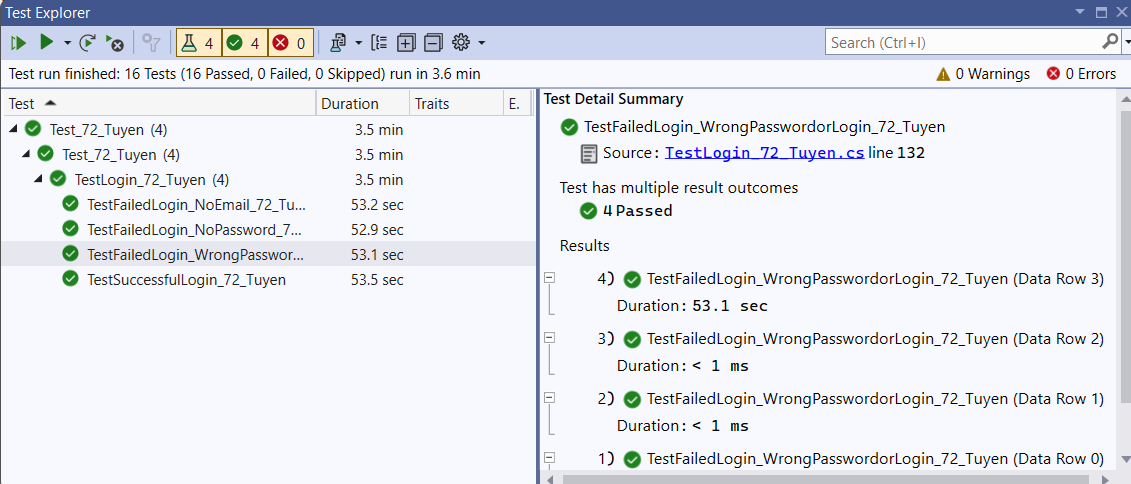
IWebElement element\_72\_Tuyen = driver.FindElement(By.ClassName("UF\_oG6g2bZsM7t2hjU3j"));

element\_72\_Tuyen.Click();

System.Threading.Thread.Sleep(1000);



Hình 14. Kết quả sau khi bắt ClassName object Login



### Bước 2: Bắt Id object Email, Password:



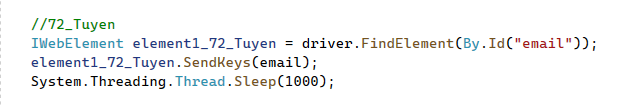
Hình 15. Bắt Id object Email

Sau khi bắt xong chúng ta sẽ thu được đoạn code sau:

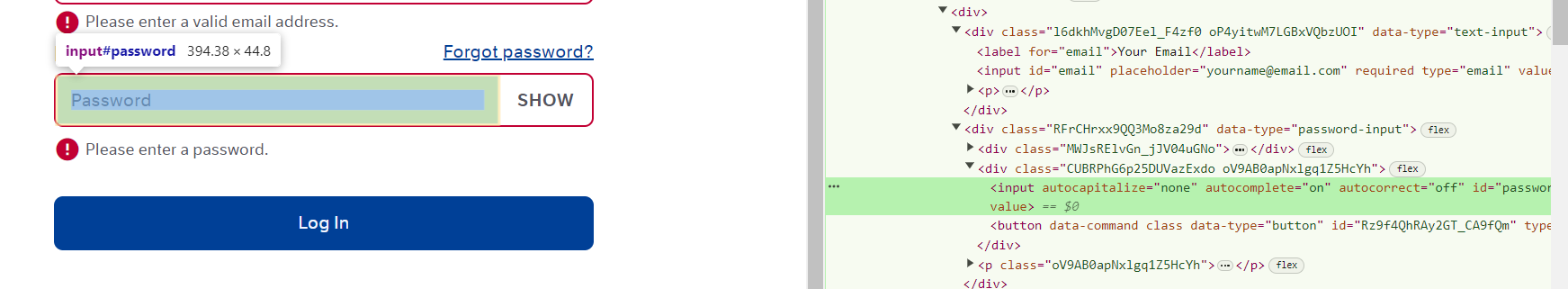
IWebElement element1\_72\_Tuyen = driver.FindElement(By.Id("email"));

element1\_72\_Tuyen.SendKeys(email);

System.Threading.Thread.Sleep(1000);



Hình 16. Kết quả sau khi bắt Id object Login



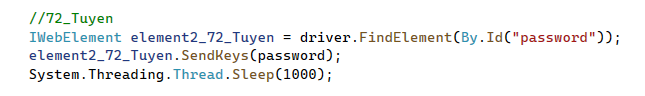
Hình 17. Bắt Id object Password

Sau khi bắt xong chúng ta sẽ thu được đoạn code sau:

IWebElement element2\_72\_Tuyen = driver.FindElement(By.Id("password"));

element2\_72\_Tuyen.SendKeys(password);

System.Threading.Thread.Sleep(1000);



Hình 18. Kết quả sau khi bắt Id object Password

### Bước 3: Bắt XPath button Login



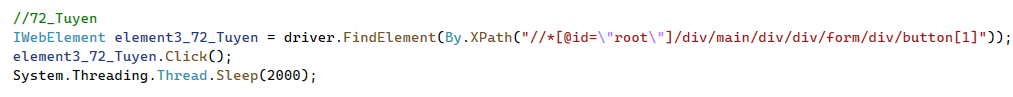
Hình 19. Bắt XPath button Login

Sau khi bắt xong chúng ta sẽ thu được đoạn code sau:

IWebElement element3\_72\_Tuyen = driver.FindElement(By.XPath("//\*[@id=\"root\"]/div/main/div/div/form/div/button[1]"));

element3\_72\_Tuyen.Click();

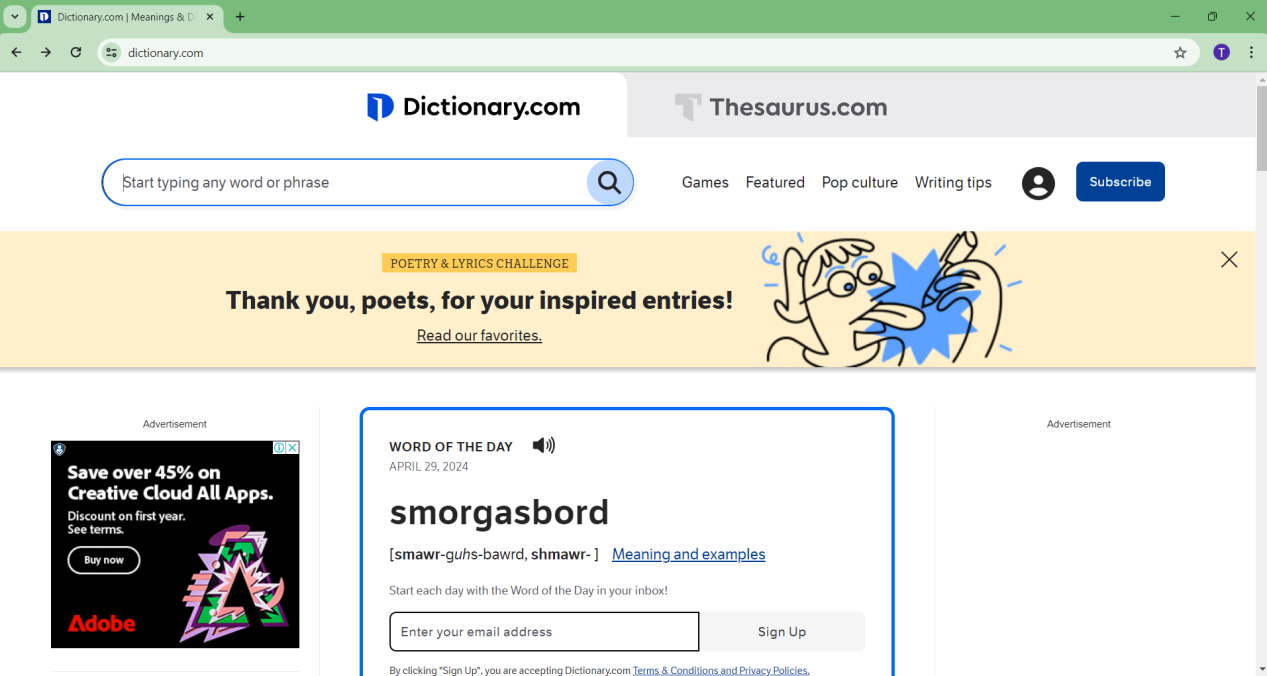
System.Threading.Thread.Sleep(2000);



Hình 20. Kết quả sau khi bắt Id object Password

## Testcase01: TestSuccessfulLogin\_72\_Tuyen (Đăng nhập thành công):

### Hình ảnh website khi đăng nhập thành công:



### **Bước 1:** Viết code cho testcase **TestSuccessfulLogin\_72\_Tuyen** dựa trên các trường Login, Email, Password, button Login đã bắt ở trên:

[TestMethod]

[DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV", @".\Data\TestLogin\_72\_Tuyen.csv", "TestLogin\_72\_Tuyen#csv", DataAccessMethod.Sequential)]

public void TestSuccessfulLogin\_72\_Tuyen()//Đăng nhập thành công

{

email = TestContext.DataRow[0].ToString();

password = TestContext.DataRow[1].ToString();

expectedResult = TestContext.DataRow[2].ToString();

if (expectedResult == "https://www.dictionary.com/sign-in?redirect=https://www.dictionary.com/")

{

driver = new ChromeDriver();

driver.Navigate().GoToUrl("https://www.dictionary.com/");

System.Threading.Thread.Sleep(1000);

IWebElement element\_72\_Tuyen = driver.FindElement(By.ClassName("UF\_oG6g2bZsM7t2hjU3j"));

element\_72\_Tuyen.Click();

System.Threading.Thread.Sleep(1000);

IWebElement element1\_72\_Tuyen = driver.FindElement(By.Id("email"));

element1\_72\_Tuyen.SendKeys(email);

System.Threading.Thread.Sleep(1000);

IWebElement element2\_72\_Tuyen = driver.FindElement(By.Id("password"));

element2\_72\_Tuyen.SendKeys(password);

System.Threading.Thread.Sleep(1000);

IWebElement element3\_72\_Tuyen = driver.FindElement(By.XPath("//\*[@id=\"root\"]/div/main/div/div/form/div/button[1]"));

element3\_72\_Tuyen.Click();

System.Threading.Thread.Sleep(2000);

Assert.AreEqual(expectedResult, driver.Url);

driver.Quit();

}

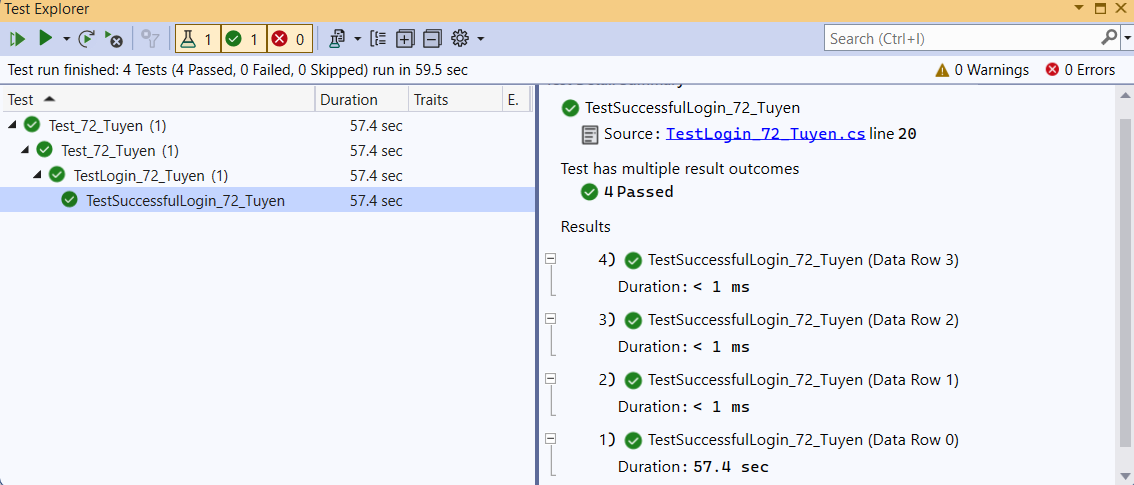
}



Hình 21. Ảnh code của testcase TestSuccessfulLogin\_72\_Tuyen

(Đăng nhập thành công)

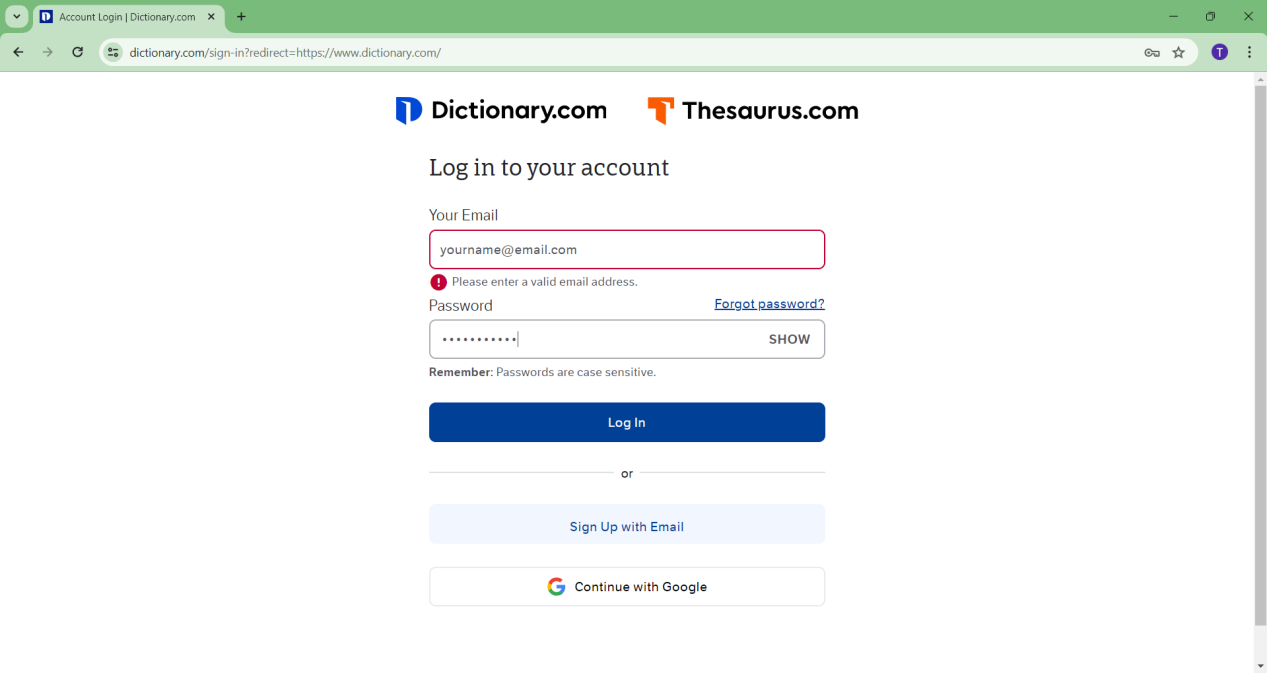
### **Bước 2:** Thực thi Testcase và xem kết quả:



Hình 22. Kết quả sau khi chạy Testcase TestSuccessfulLogin\_72\_Tuyen

## Testcase02: TestFailedLogin\_NoEmail\_72\_Tuyen (Không điền trường email):

### Hình ảnh website đăng nhập không thành công vì không nhập trường email:



### **Bước 1:** Viết code cho testcase **TestFailedLogin\_NoEmail\_72\_Tuyen** dựa trên các trường Login, Email, Password, button Login đã bắt ở trên:

[TestMethod] //Không điền trường Email

[DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV", @".\Data\TestLogin\_72\_Tuyen.csv", "TestLogin\_72\_Tuyen#csv", DataAccessMethod.Sequential)]

public void TestFailedLogin\_NoEmail\_72\_Tuyen()

{

email = TestContext.DataRow[0].ToString();

password = TestContext.DataRow[1].ToString();

expectedResult = TestContext.DataRow[2].ToString();

if (email == "")

{

driver = new ChromeDriver();

driver.Navigate().GoToUrl("https://www.dictionary.com/");

System.Threading.Thread.Sleep(1000);

IWebElement element\_72\_Tuyen = driver.FindElement(By.ClassName("UF\_oG6g2bZsM7t2hjU3j"));

element\_72\_Tuyen.Click();

System.Threading.Thread.Sleep(1000);

IWebElement element1\_72\_Tuyen = driver.FindElement(By.Id("email"));

element1\_72\_Tuyen.SendKeys(email);

System.Threading.Thread.Sleep(1000);

IWebElement element2\_72\_Tuyen = driver.FindElement(By.Id("password"));

element2\_72\_Tuyen.SendKeys(password);

System.Threading.Thread.Sleep(1000);

IWebElement element3\_72\_Tuyen = driver.FindElement(By.XPath("//\*[@id=\"root\"]/div/main/div/div/form/div/button[1]"));

element3\_72\_Tuyen.Click();

IWebElement element4\_72\_Tuyen = driver.FindElement(By.XPath("//\*[@id=\"root\"]/div/main/div/div/form/div/div[1]/p"));

System.Threading.Thread.Sleep(2000);

Assert.AreEqual(expectedResult, element4\_72\_Tuyen.Text);

driver.Quit();

}

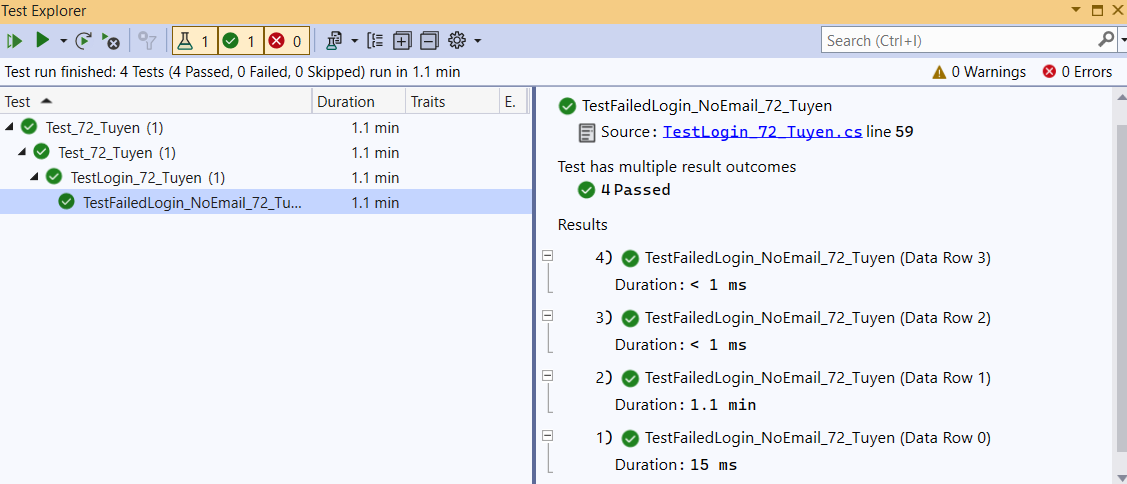
}



Hình 23. Ảnh code của testcase TestFailedLogin\_NoEmail\_72\_Tuyen

(Đăng nhập không thành công vì không điền email)

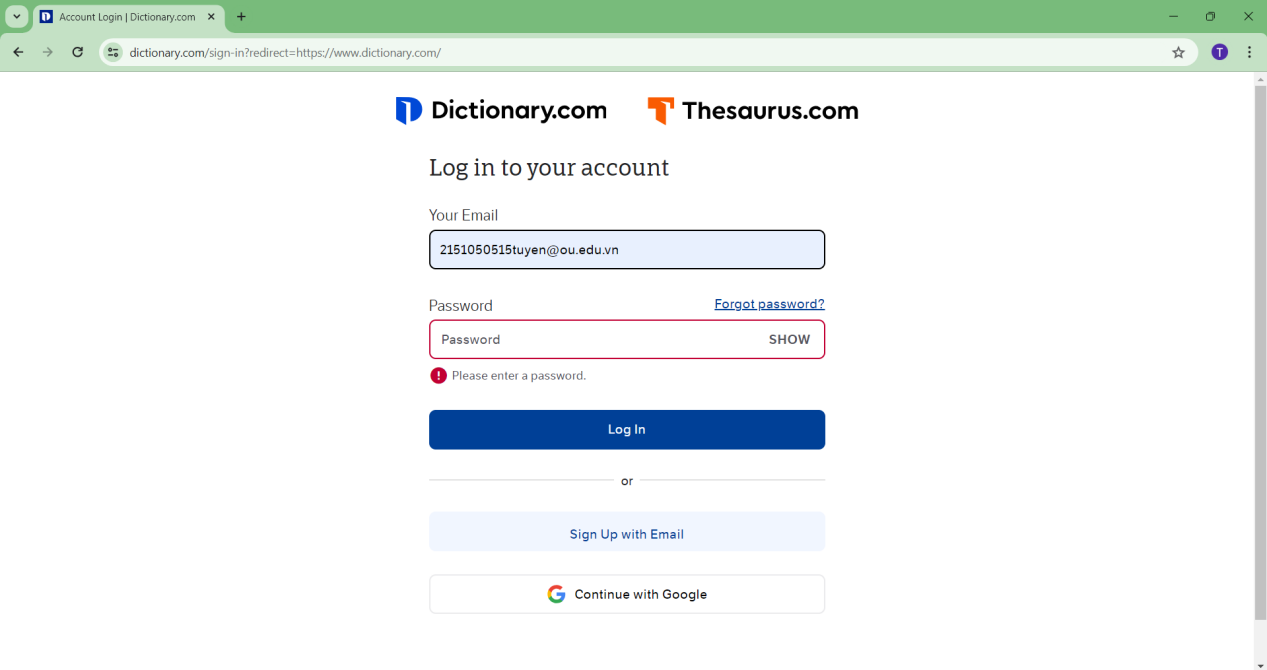
### **Bước 2:** Thực thi Testcase và xem kết quả:



Hình 24. Kết quả sau khi chạy testcase TestFailedLogin\_NoEmail\_72\_Tuyen

## Testcase03: TestFailedLogin\_NoPassword\_72\_Tuyen (Không điền trường Password):

### Hình ảnh website đăng nhập không thành công vì không nhập trường password:



### **Bước 1:** Viết code cho testcase

TestFailedLogin\_NoPassword\_72\_Tuyen dựa trên các trường Login, Email, Password, button Login đã bắt ở trên:

[TestMethod]//Không điền trường Password

[DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV", @".\Data\TestLogin\_72\_Tuyen.csv", "TestLogin\_72\_Tuyen#csv", DataAccessMethod.Sequential)]

public void TestFailedLogin\_NoPassword\_72\_Tuyen()

{

email = TestContext.DataRow[0].ToString();

password = TestContext.DataRow[1].ToString();

expectedResult = TestContext.DataRow[2].ToString();

if (expectedResult == "Please enter a password.")

{

driver = new ChromeDriver();

driver.Navigate().GoToUrl("https://www.dictionary.com/");

System.Threading.Thread.Sleep(1000);

IWebElement element\_72\_Tuyen = driver.FindElement(By.ClassName("UF\_oG6g2bZsM7t2hjU3j"));

element\_72\_Tuyen.Click();

System.Threading.Thread.Sleep(1000);

IWebElement element1\_72\_Tuyen = driver.FindElement(By.Id("email"));

element1\_72\_Tuyen.SendKeys(email);

System.Threading.Thread.Sleep(1000);

IWebElement element2\_72\_Tuyen = driver.FindElement(By.Id("password"));

element2\_72\_Tuyen.SendKeys(password);

System.Threading.Thread.Sleep(1000);

IWebElement element3\_72\_Tuyen = driver.FindElement(By.XPath("//\*[@id=\"root\"]/div/main/div/div/form/div/button[1]"));

element3\_72\_Tuyen.Click();

System.Threading.Thread.Sleep(2000);

IWebElement element4\_72\_Tuyen = driver.FindElement(By.XPath("//\*[@id=\"root\"]/div/main/div/div/form/div/div[2]/p"));

Assert.AreEqual(expectedResult, element4\_72\_Tuyen.Text);

driver.Quit();

}

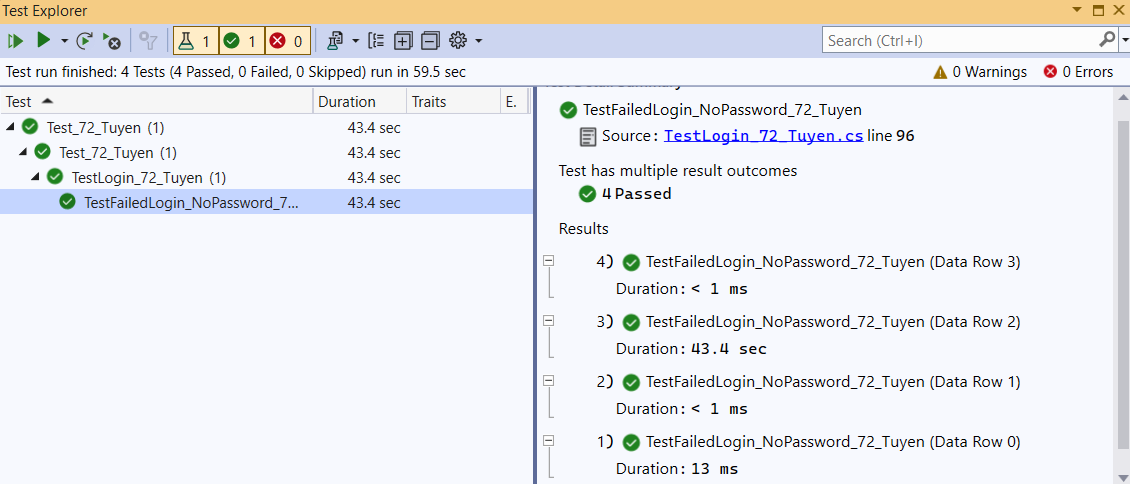
}



Hình 25. Ảnh code của testcase TestFailedLogin\_NoPassword\_72\_Tuyen

(Đăng nhập không thành công vì không điền password)

### **Bước 2:** Thực thi Testcase và xem kết quả:

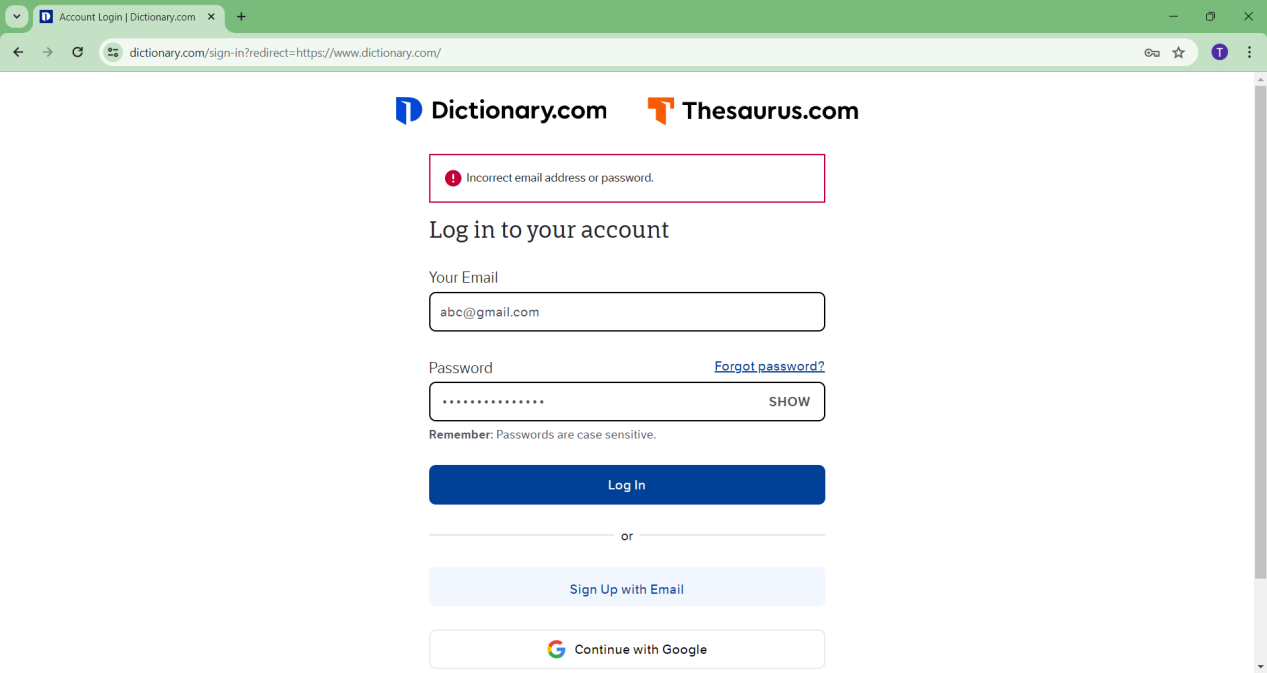


Hình 26. Kết quả của testcase TestFailedLogin\_NoPassword\_72\_Tuyen

## Testcase04: TestFailedLogin\_WrongPasswordorLogin\_72\_Tuyen

**(Tài khoản không tồn tại, email và password đều sai):**

### Hình ảnh website đăng nhập không thành công vì không nhập trường password:



### **Bước 1:** Viết code cho testcase

TestFailedLogin\_NoPassword\_72\_Tuyen dựa trên các trường Login, Email, Password, button Login đã bắt ở trên:

[TestMethod]//email và password không tồn tại

[DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV", @".\Data\TestLogin\_72\_Tuyen.csv", "TestLogin\_72\_Tuyen#csv", DataAccessMethod.Sequential)]

public void TestFailedLogin\_WrongPasswordorLogin\_72\_Tuyen()

{

email = TestContext.DataRow[0].ToString();

password = TestContext.DataRow[1].ToString();

expectedResult = TestContext.DataRow[2].ToString();

if (expectedResult == "Incorrect email address or password.")

{

driver = new ChromeDriver();

driver.Navigate().GoToUrl("https://www.dictionary.com/");

System.Threading.Thread.Sleep(1000);

IWebElement element\_72\_Tuyen = driver.FindElement(By.ClassName("UF\_oG6g2bZsM7t2hjU3j"));

element\_72\_Tuyen.Click();

System.Threading.Thread.Sleep(1000);

IWebElement element1\_72\_Tuyen = driver.FindElement(By.Id("email"));

element1\_72\_Tuyen.SendKeys(email);

System.Threading.Thread.Sleep(1000);

IWebElement element2\_72\_Tuyen = driver.FindElement(By.Id("password"));

element2\_72\_Tuyen.SendKeys(password);

System.Threading.Thread.Sleep(1000);

IWebElement element3\_72\_Tuyen = driver.FindElement(By.XPath("//\*[@id=\"root\"]/div/main/div/div/form/div/button[1]"));

element3\_72\_Tuyen.Click();

System.Threading.Thread.Sleep(2000);

IWebElement element4\_72\_Tuyen = driver.FindElement(By.ClassName("BYHIXdj8Bf9CJAeu3iRG"));

Assert.AreEqual(expectedResult, element4\_72\_Tuyen.Text);

driver.Quit();

}

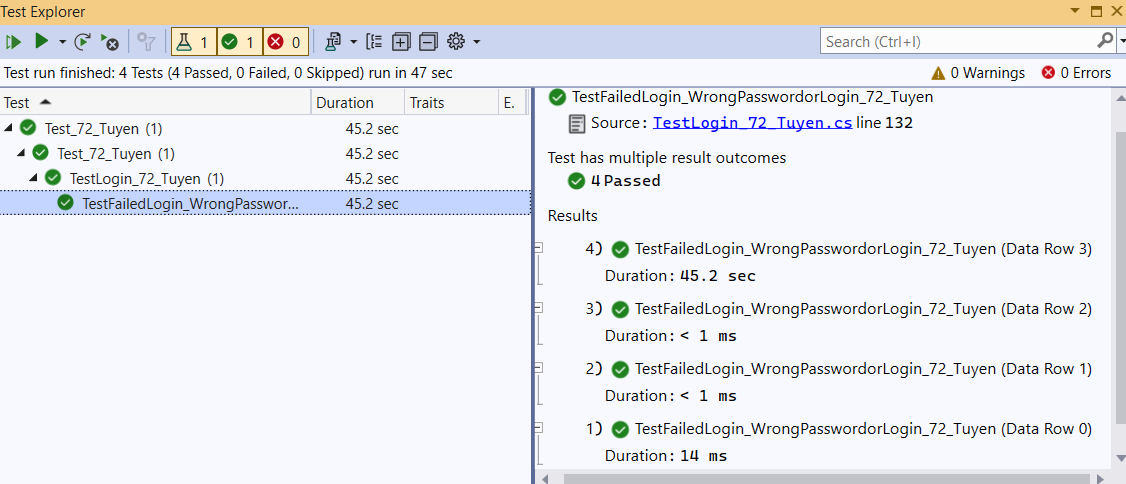
}



Hình 27.Ảnh code của testcase TestFailedLogin\_NoPassword\_72\_Tuyen

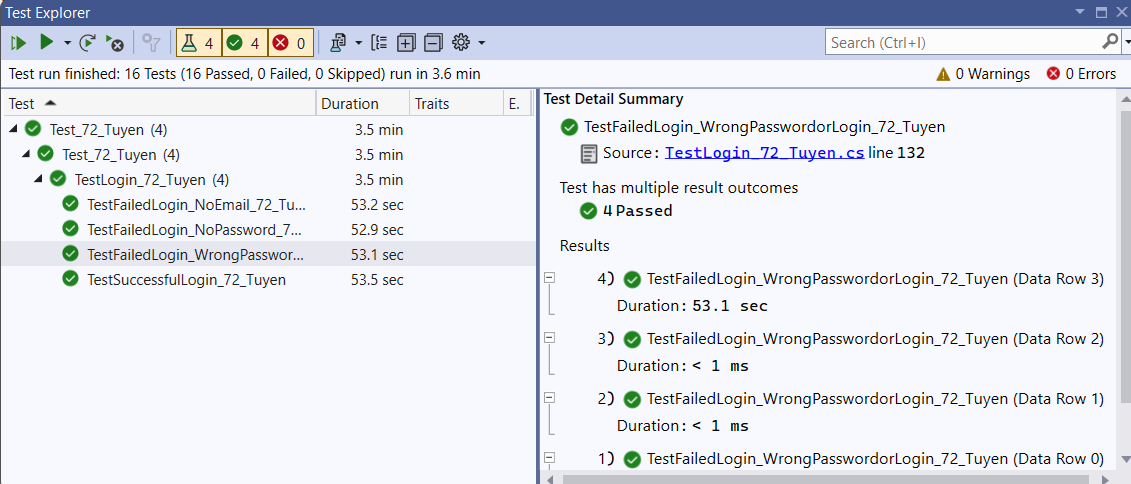
(Đăng nhập không thành công vì email và password đều sai)

### **Bước 2:** Thực thi Testcase và xem kết quả:



Hình 28. Kết quả của testcase TestFailedLogin\_NoPassword\_72\_Tuyen

## Kết quả sau khi chạy hết 4 Testcase:



Hình 29. Kết quả sau khi thực thi hết 4 Testcase

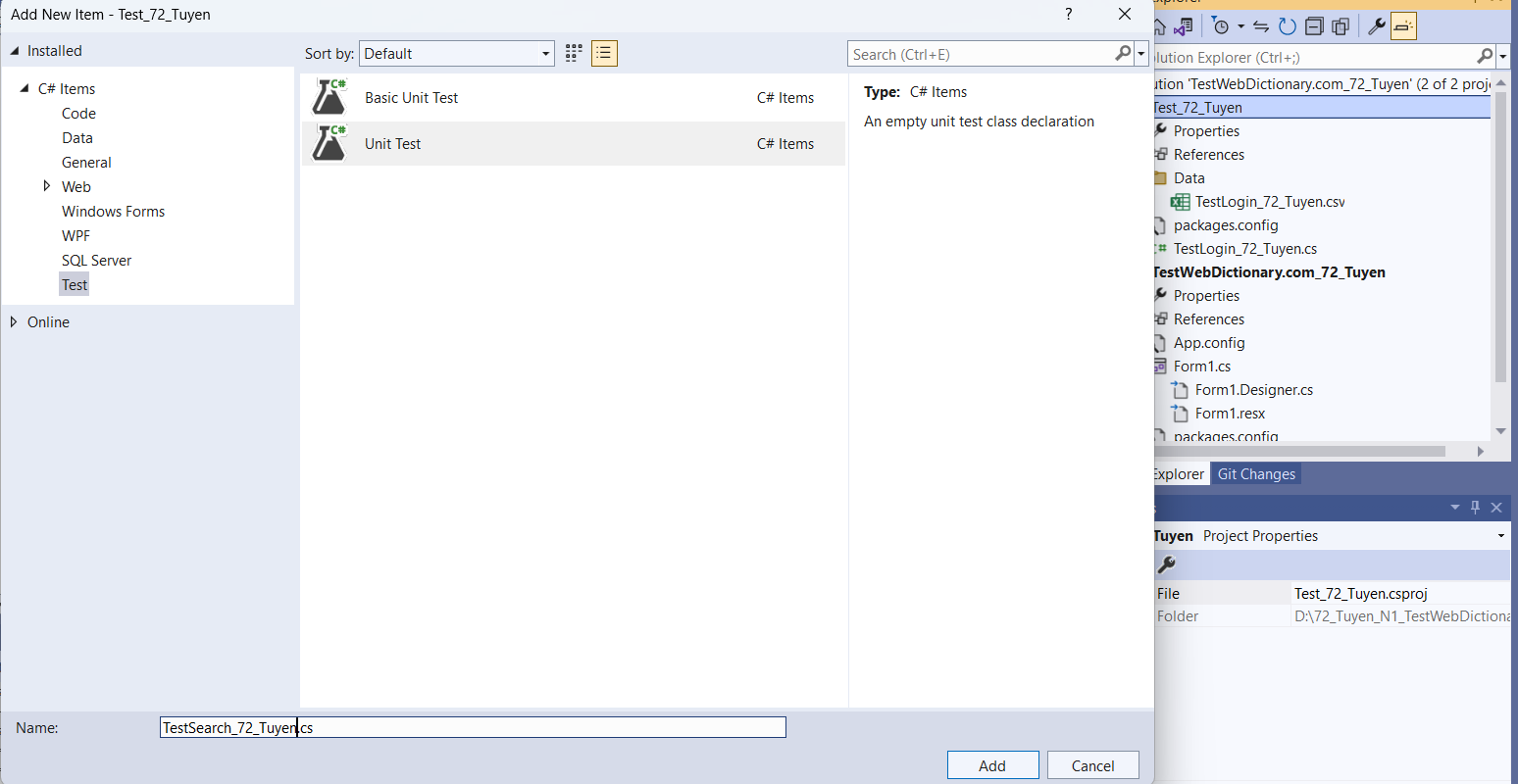
# KIỂM THỬ CHỨC NĂNG TÌM KIẾM (SEARCH):

## Mô tả chức năng tìm kiếm:

Chức năng sẽ tìm kiếm những từ liên quan đến từ khóa mà bạn nhập trên thanh tìm kiếm sau đó nó sẽ trả về tất cả các từ có liên đến từ khóa mà bạn tìm kiếm được.

## Thực hiện test chức năng tìm kiếm của trang Dictionary.com

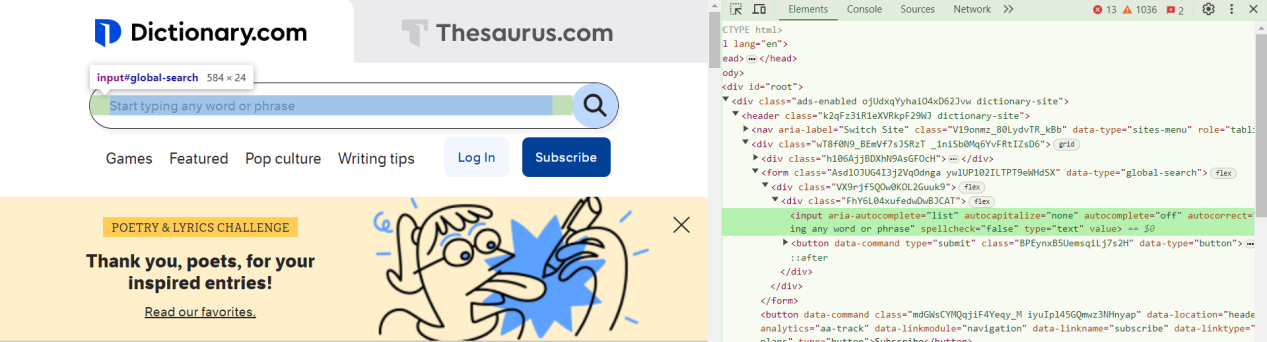
Bước 1: Tạo thêm 1 project Unit Test đặt tên là TestSearch\_72\_Tuyen



Hình 30. Tạo một project Unit Test mới

Bước 2: Cài đặt Selenium.Web Driver và Selenium.WebDriver.ChromeDriver

Bước 3: Bắt CssSelector, ClassName, XPath cho các đối tượng cần test



Hình 31. Bắt CssSelector object thanh tìm kiếm

Sau khi bắt chúng ta thu được đoạn code sau:

IWebElement element\_72\_Tuyen = driver.FindElement(By.CssSelector("input[type='text']"));

element\_72\_Tuyen.SendKeys(search);

System.Threading.Thread.Sleep(1000);



Hình 32. Kết quả sau khi bắt CssSelector object thanh tìm kiếm



Hình 33. Bắt ClassName button tìm kiếm

Sau khi bắt chúng ta sẽ thu được đoạn code sau:

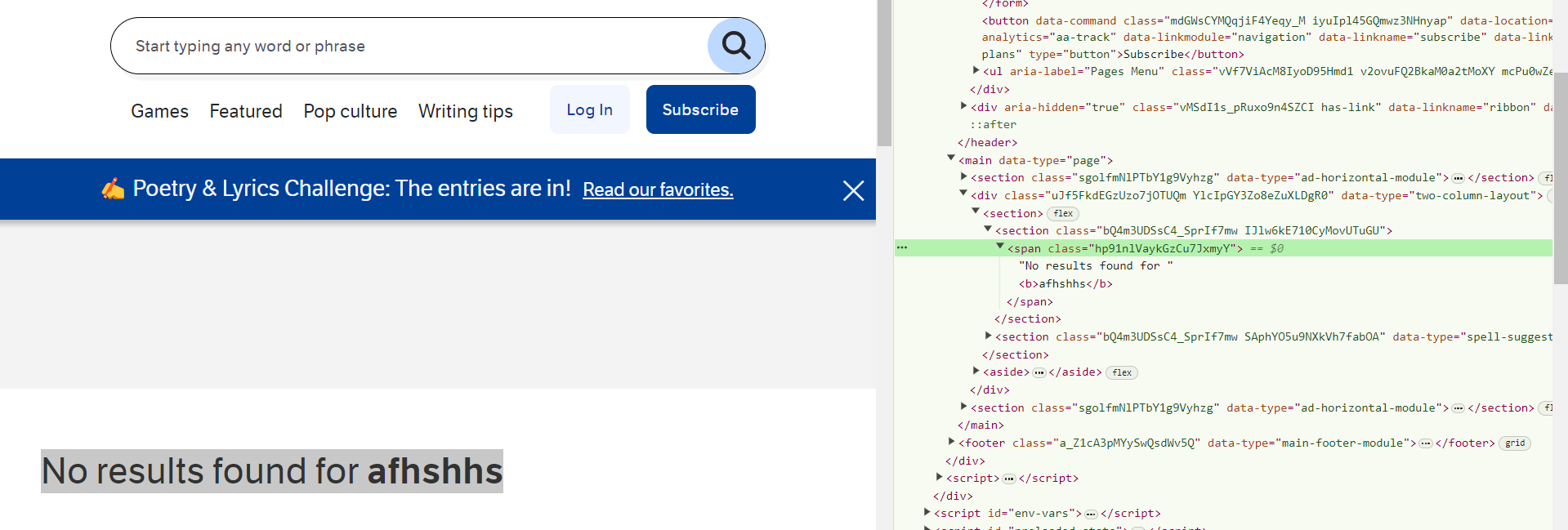
IWebElement element1\_72\_Tuyen = driver.FindElement(By.ClassName("BPEynxB5UemsqiLj7s2H"));

element1\_72\_Tuyen.Click();

System.Threading.Thread.Sleep(1000);



Hình 34. Kết quả sau khi bắt ClassName button tìm kiếm



Hình 35. Bắt sự kiện thông báo tìm kiếm fail

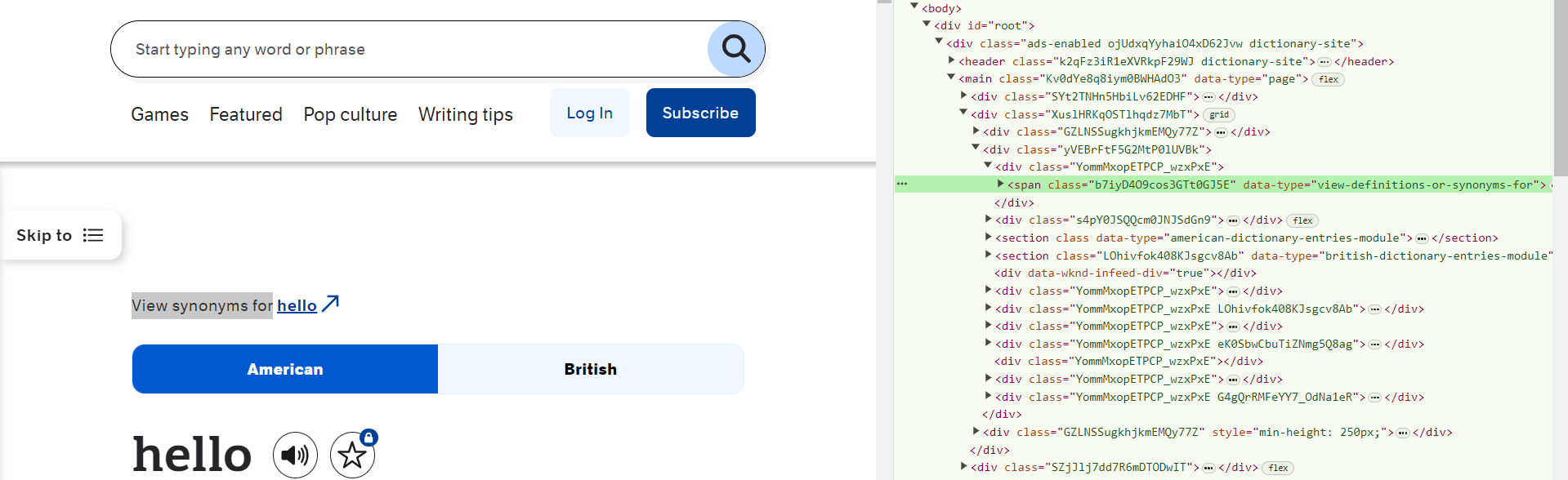
Sau khi bắt chúng ta sẽ thu được đoạn code sau:

IWebElement element3\_72\_Tuyen = driver.FindElement(By.XPath("//\*[@id=\"root\"]/div/main/div/section/section[1]/span"));

element3\_72\_Tuyen.Click();



Hình 36. Kết quả sau khi bắt sự kiện tìm kiếm fail

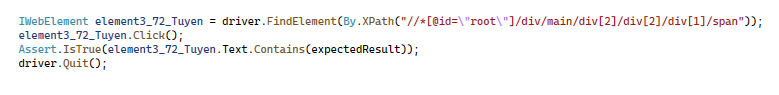


Hình 37. Bắt sự kiện thông báo tìm kiếm thành công

Sau khi bắt chúng ta thu được đoạn code sau:

IWebElement element3\_72\_Tuyen = driver.FindElement(By.XPath("//\*[@id=\"root\"]/div/main/div[2]/div[2]/div[1]/span"));

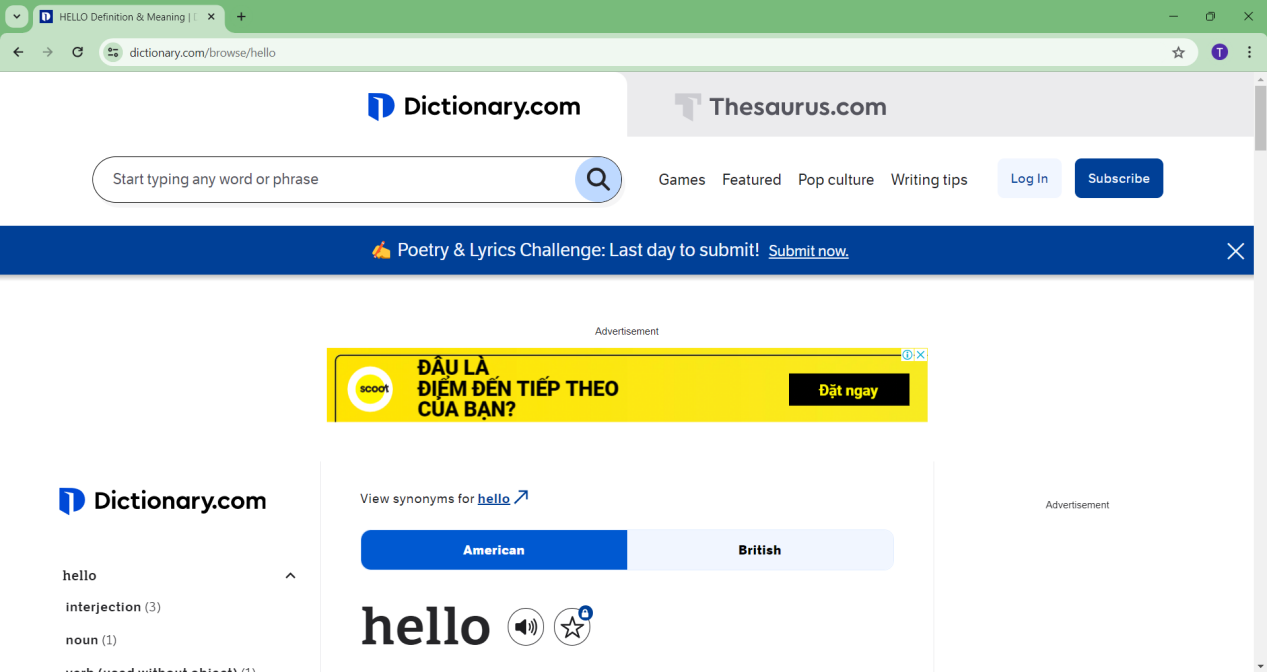
element3\_72\_Tuyen.Click();



Hình 38. Kết quả sau khi bắt sự kiện tìm kiếm thành công

## Testcase01: TestSuccessfulSearch\_72\_Tuyen(Tìm kiếm thành công):

### Hình ảnh website tìm kiếm thành công:



### Bước 1: Viết code cho testcase TestSuccessfulSearch\_72\_Tuyen:

[DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV", @".\Data\TestSearch\_72\_Tuyen.csv", "TestSearch\_72\_Tuyen#csv", DataAccessMethod.Sequential)]

[TestMethod]

public void TestSuccessfulSearch\_72\_Tuyen()

{

search = TestContext.DataRow[0].ToString();

expectedResult = TestContext.DataRow[1].ToString();

if (expectedResult == "View synonyms for")

{

driver = new ChromeDriver();

driver.Navigate().GoToUrl("https://www.dictionary.com/");

System.Threading.Thread.Sleep(1000);

IWebElement element\_72\_Tuyen = driver.FindElement(By.CssSelector("input[type='text']"));

element\_72\_Tuyen.SendKeys(search);

System.Threading.Thread.Sleep(1000);

IWebElement element1\_72\_Tuyen = driver.FindElement(By.ClassName("BPEynxB5UemsqiLj7s2H"));

element1\_72\_Tuyen.Click();

System.Threading.Thread.Sleep(1000);

IWebElement element3\_72\_Tuyen = driver.FindElement(By.XPath("//\*[@id=\"root\"]/div/main/div[2]/div[2]/div[1]/span"));

element3\_72\_Tuyen.Click();

Assert.IsTrue(element3\_72\_Tuyen.Text.Contains(expectedResult));

driver.Quit();

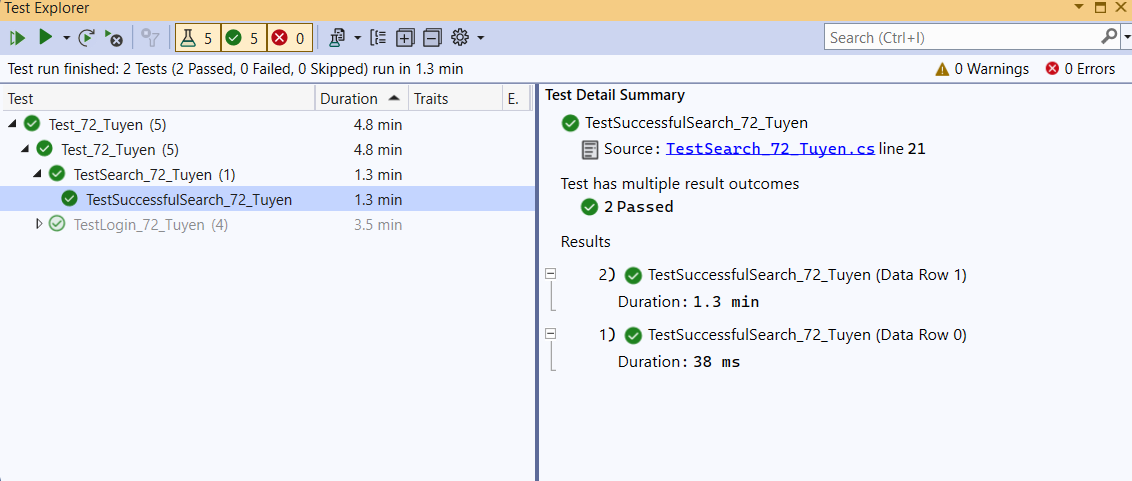
}

}



Hình 39. Ảnh code của testcase TestSuccessfulSearch\_72\_Tuyen

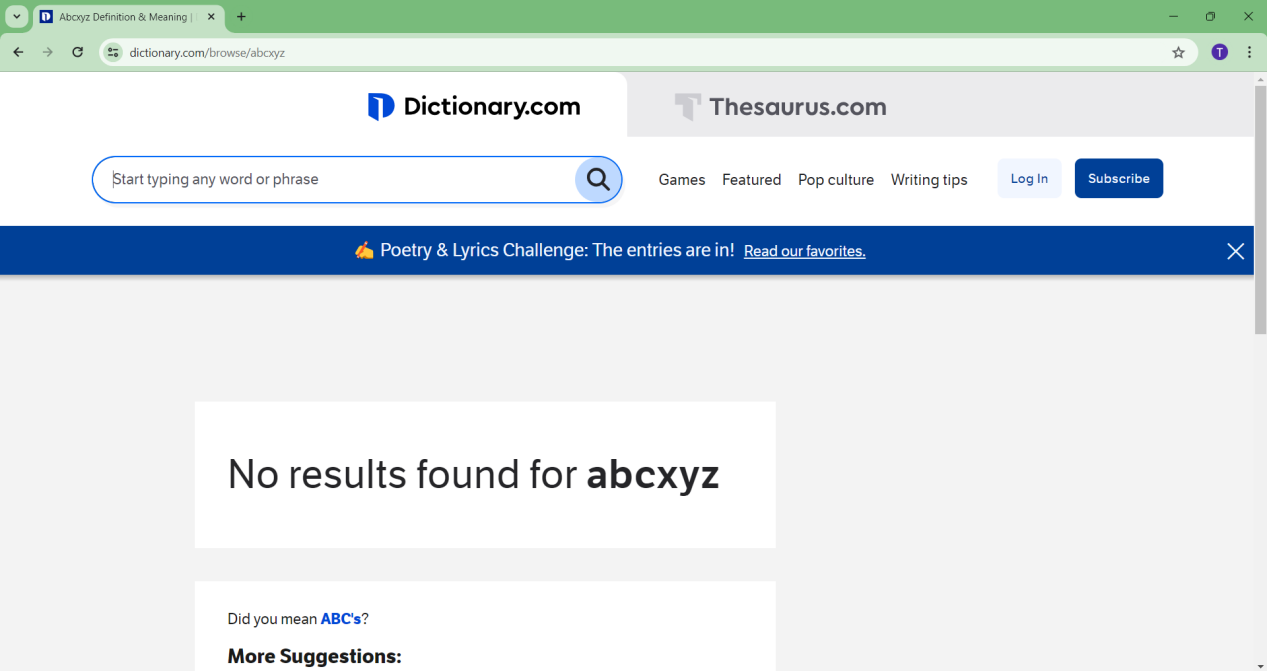
### Bước 2: Thực hiện testcase và xem kết quả:



Hình 40. Kết quả sau khi chạy Testcase

## Testcase02: TestFailSearch\_72\_Tuyen (Tìm kiếm không thành công):

### Hình ảnh website sau khi tìm kiếm không thành công:



Hình 41. Hình ảnh website khi tìm kiếm không thành công

### **Bước 1:** Viết code cho testcase TestFailSearch\_72\_Tuyen:

[TestMethod]

[DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV", @".\Data\TestSearch\_72\_Tuyen.csv", "TestSearch\_72\_Tuyen#csv", DataAccessMethod.Sequential)]

public void TestFailSearch\_72\_Tuyen()

{

search = TestContext.DataRow[0].ToString();

expectedResult = TestContext.DataRow[1].ToString();

if (expectedResult == "No results found for")

{

driver = new ChromeDriver();

driver.Navigate().GoToUrl("https://www.dictionary.com/");

System.Threading.Thread.Sleep(1000);

IWebElement element\_72\_Tuyen = driver.FindElement(By.CssSelector("input[type='text']"));

element\_72\_Tuyen.SendKeys(search);

System.Threading.Thread.Sleep(1000);

IWebElement element1\_72\_Tuyen = driver.FindElement(By.ClassName("BPEynxB5UemsqiLj7s2H"));

element1\_72\_Tuyen.Click();

System.Threading.Thread.Sleep(1000);

IWebElement element3\_72\_Tuyen = driver.FindElement(By.XPath("//\*[@id=\"root\"]/div/main/div/section/section[1]/span"));

element3\_72\_Tuyen.Click();

Assert.IsTrue(element3\_72\_Tuyen.Text.Contains(expectedResult));

driver.Quit();

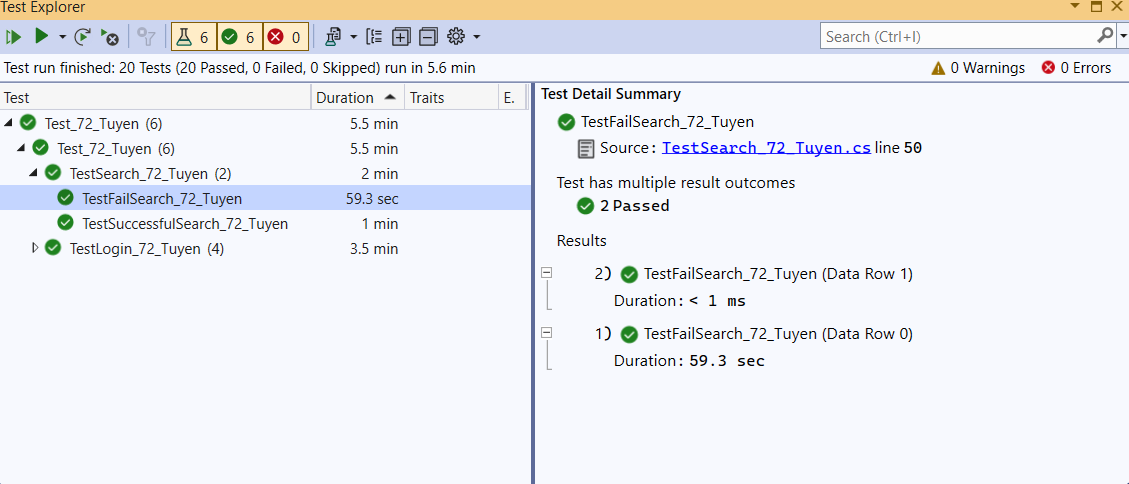
}

}



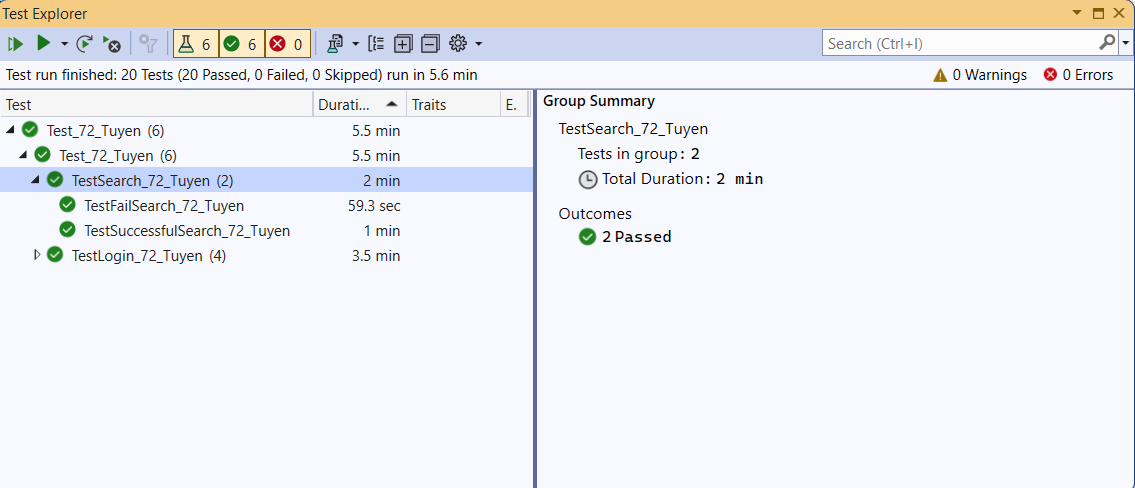
Hình 42. Ảnh code của Testcasse TestFailSearch\_72\_Tuyen

### Bước 2: Thực hiện testcase và xem kết quả:



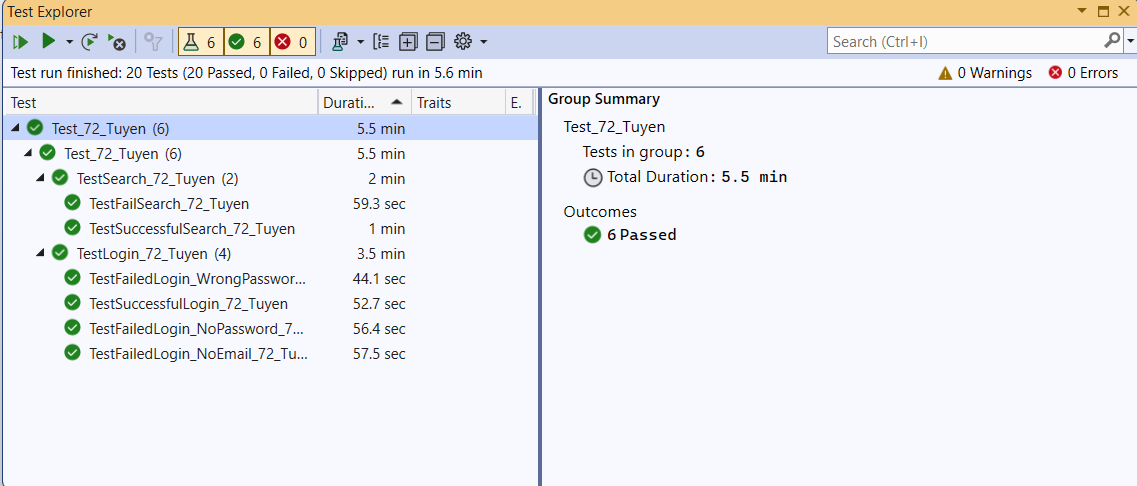
Hình 43. Kết quả sau khi chạy Testcase

## **Kết quả sau khi chạy 2 testcase:**



Hình 44. Kết quả sau khi chạy 2 Testcase tìm kiếm

# KẾT QUẢ SAU KHI CHẠY 6 TESTCASE:



Hình 45. Kết quả sau khi chạy hết 6 Testcase

# API TESTING:

## API là gì?

API là cầu nối giữa Client và Server. Client có thể là máy tính, điện thoại, sử dụng hệ điều hành khác nhau và được viết bằng những ngôn ngữ khác nhau. Server cũng được viết bằng những ngôn ngữ khác nhau. API là ngôn ngữ để Client và Server có thể nói chuyện được với nhau.

## Vì sao phải Test API?

## Trong quá trình triển khai dự án, phần Server và Client làm độc lập với nhau nên có nhiều chỗ bên Client chưa làm xong, ta không thể chờ Client làm xong để test được dữ liệu. Do đó ta cần kiểm tra API bằng công cụ khác.

## Các tool cho kiểm thử API:

Có rất nhiều công cụ dùng để kiểm thử API như:

* SOAPUI
* Runscope
* Postman
* Curl
* Cfix
* Check
* CTESK

## Các phương thức cơ bản:

Có 4 phương thức cơ bản:

* GET: Sử dụng để lấy thông tin từ server theo URI đã cung cấp.
* PUT: Ghi đè tất cả thông tin của đối tượng với những gì được gửi lên
* POST: Gửi thông tin tới server thông qua các parameters HTTP
* DELETE: Xóa resource trên server