**BỘ** **GIÁO** **DỤC** **VÀ** **ĐÀO** **TẠO**

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



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

***ĐỀ*** ***TÀI:*** **KIỂM** **THỬ** **WEBSITE** **SPEAK LANGUAGE**

**GVHD:** Nguyễn Thị Ngọc Thanh

**SINH** **VIÊN** **THỰC** **HIỆN:** Nguyễn Ngọc Hậu

**STT:** 22

**MÃ** **SỐ** **SINH** **VIÊN** **:** 2151010329

**KHOA** **:** Công Nghệ Công Tin

**LỚP:** DH21IT02

**TPHCM,** **01/2024**

BÁO CÁO MÔN KIỂM THỬ PHẦN MỀM

[**1.** **CÀI ĐẶT NUNIT VÀ SELENIUM WEB DRIVER (Version: Latest stable 124.0.6367.7800)** 3](#_Toc165585320)

[**1.1.** **Tạo 1 project Window Form C#** 3](#_Toc165585321)

[**1.2.** **Tạo 1 project kiểm thử Unit Test – Nunit(#)** 3](#_Toc165585322)

[**1.3.** **Cài đặt Selenium Web Server** 5](#_Toc165585323)

[**2.** **TRIỂN KHAI TEST CASE** 7](#_Toc165585324)

[**2.1.** **Thực hiện test trang đăng nhập (TestLogin)** 7](#_Toc165585325)

[**2.1.1.** **Test case 1: TestSuccessfulLogin\_22Hau (Đăng nhập thành công)** 13](#_Toc165585326)

[**2.1.2.** **Test case 2: TestFailedLogin\_NoEmail\_22Hau (Đăng nhập không nhập email)** 14](#_Toc165585327)

[**2.1.3.** **Test case 3: TestFailedLogin\_NoPassword\_22Hau (Đăng nhập không nhập password)** 16](#_Toc165585328)

[**2.1.4.** **Test case 4: TestFailedLogin\_WrongPasswordorLogin\_22Hau(Đăng nhập sai email hoặc password)** 18](#_Toc165585329)

[**2.1.5.** **Tổng hợp code chạy cho 4 test case** 20](#_Toc165585330)

[**2.2.** **Thực hiên test button tìm kiếm (TestSearch)** 21](#_Toc165585331)

[**2.2.1.** **Test case 1: TestSuccessfullSearch\_22Hau()** 24](#_Toc165585332)

[**2.2.2.** **Test case 2: TestNoSearch\_22Hau()** 25](#_Toc165585333)

[**2.2.3.** **Tổng hợp code chạy 3 testcase** 27](#_Toc165585334)

[**3.** **API TEST** 28](#_Toc165585335)

[**3.1.** **Tạo Workspaces** 28](#_Toc165585336)

[**3.2.** **Test API của request Post (Trang web Speak Language)** 29](#_Toc165585337)

[**3.2.1.** **Tạo request phương Post** 29](#_Toc165585338)

[**3.2.2.** **Viết Test trên phương thức Post** 32](#_Toc165585339)

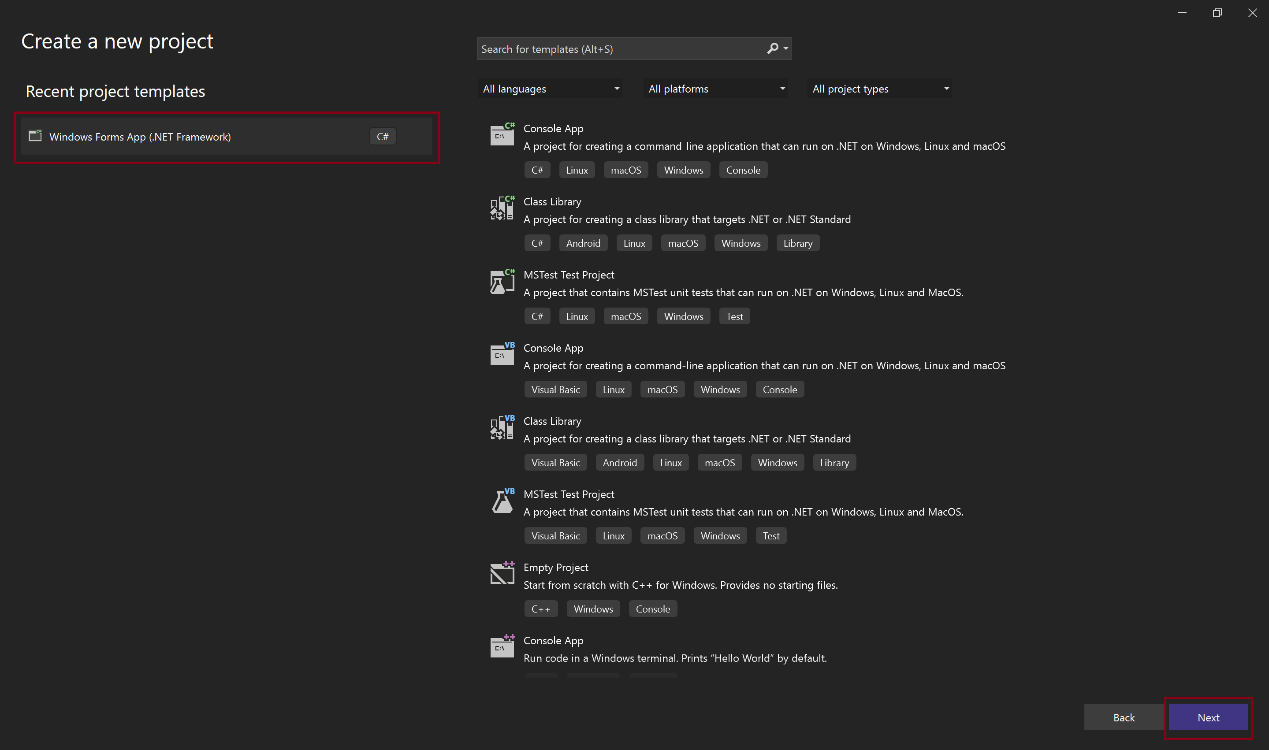
[**3.3.** **Test API của request Get (Trang web Viblo)** 33](#_Toc165585340)

[**3.3.1.** **Tạo request cho phương thức get** 34](#_Toc165585341)

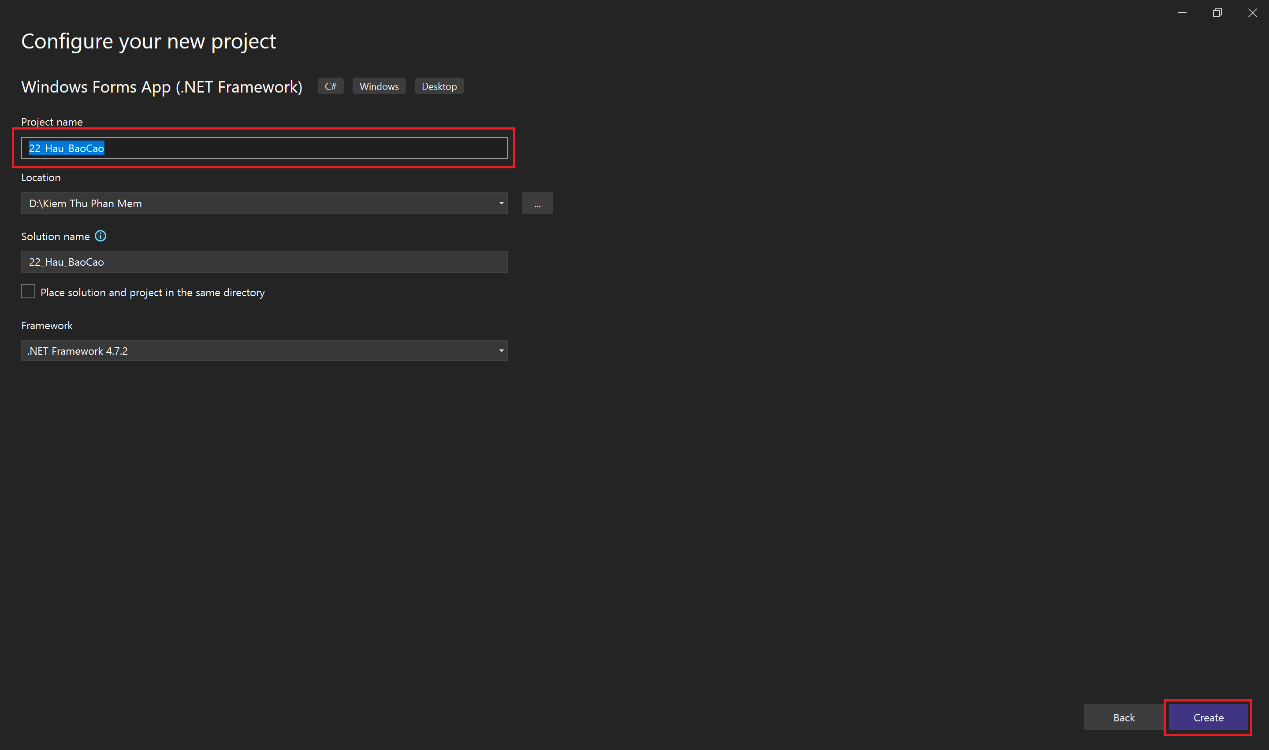
[**3.3.2.** **Viết test trên phương thức get** 35](#_Toc165585342)

# **CÀI ĐẶT UNIT VÀ SELENIUM WEB DRIVER (Version: Latest stable 124.0.6367.7800)**

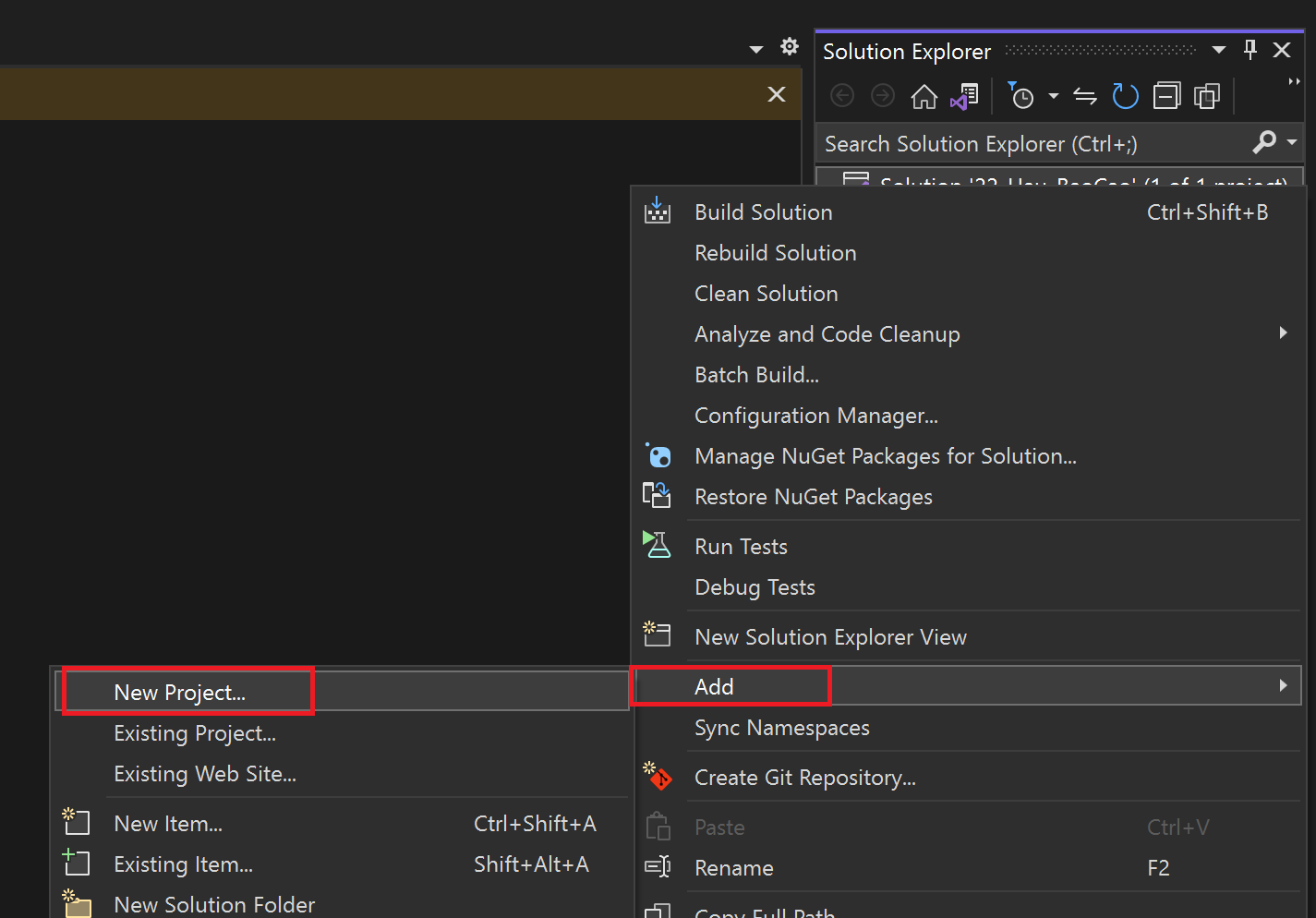
* 1. **Tạo 1 project Window Form C#**
* Bước 1: Chọn Windows Form App (.NET Framework) => next



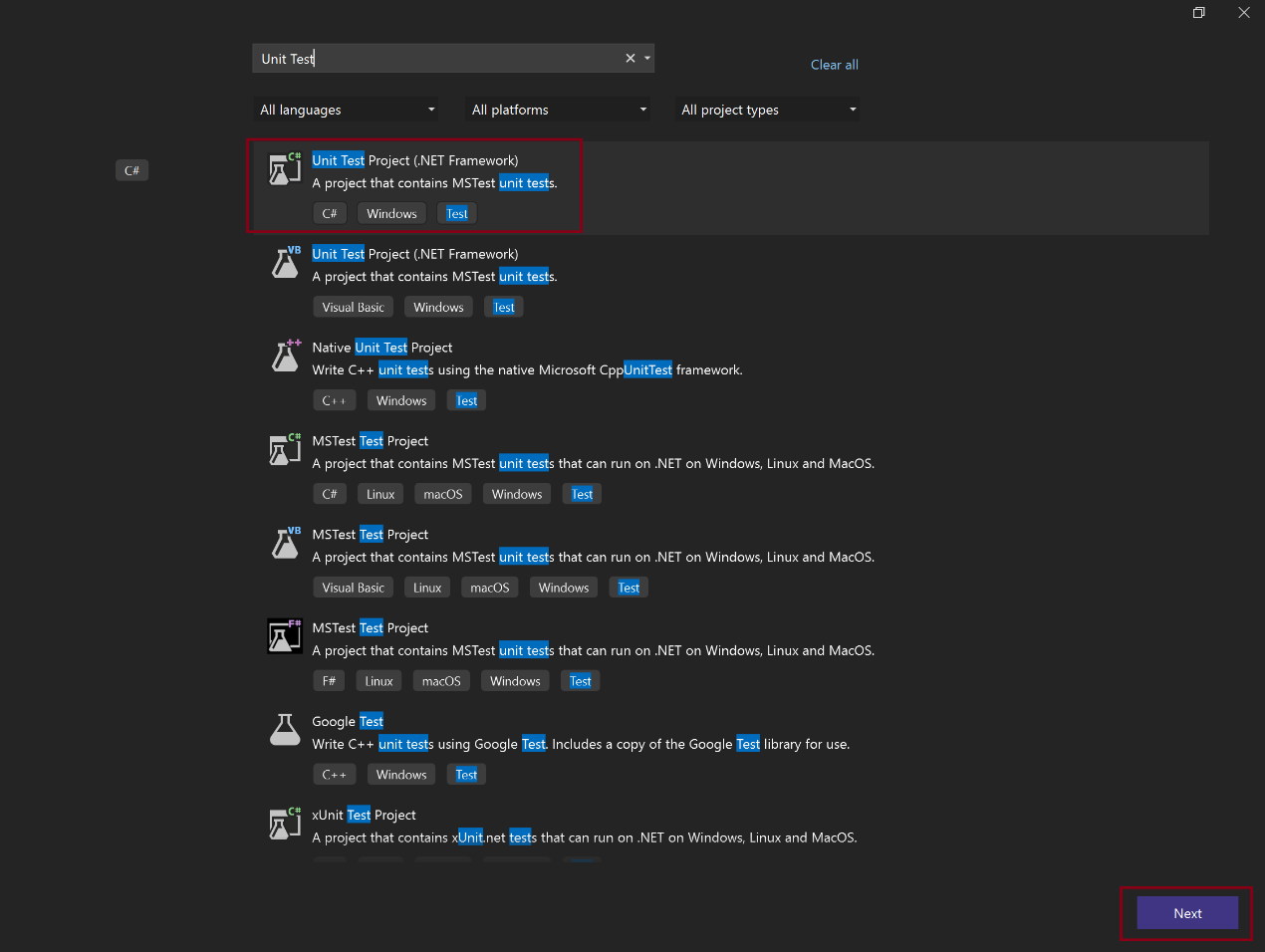
* Bước 2: Đặt tên project: 22\_Hau\_BaoCao => create



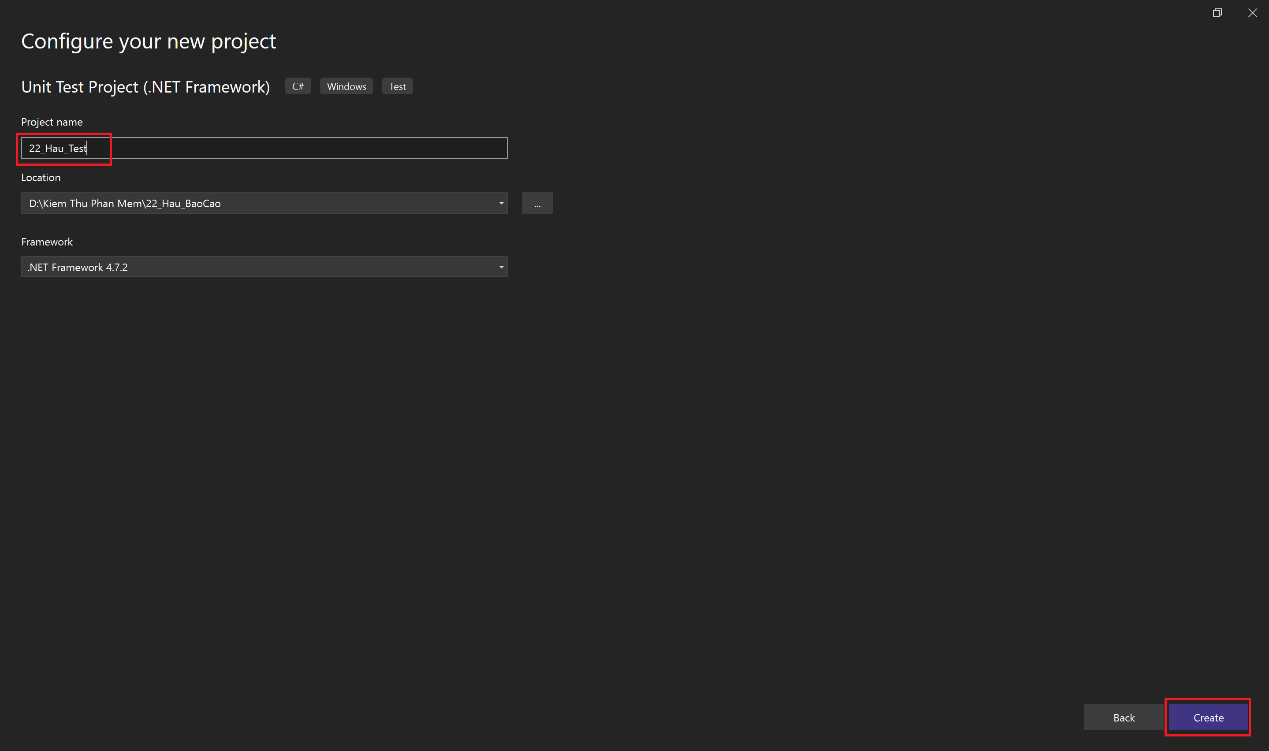
* 1. **Tạo 1 project kiểm thử Unit Test**
* Bước 1: Click phải vào Solution chọn Add => New project



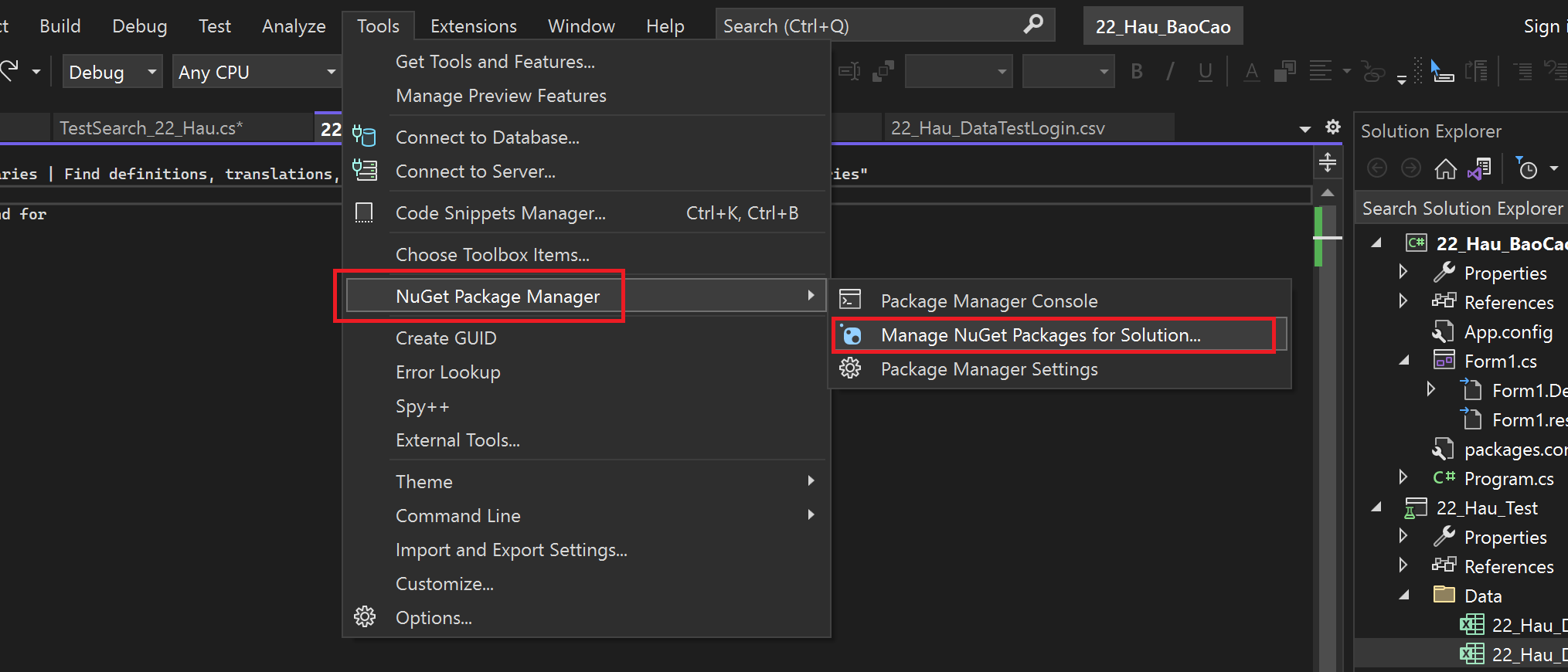
* Bước 2: Chọn Unit Test Project(.NET Framework) 🡪 next



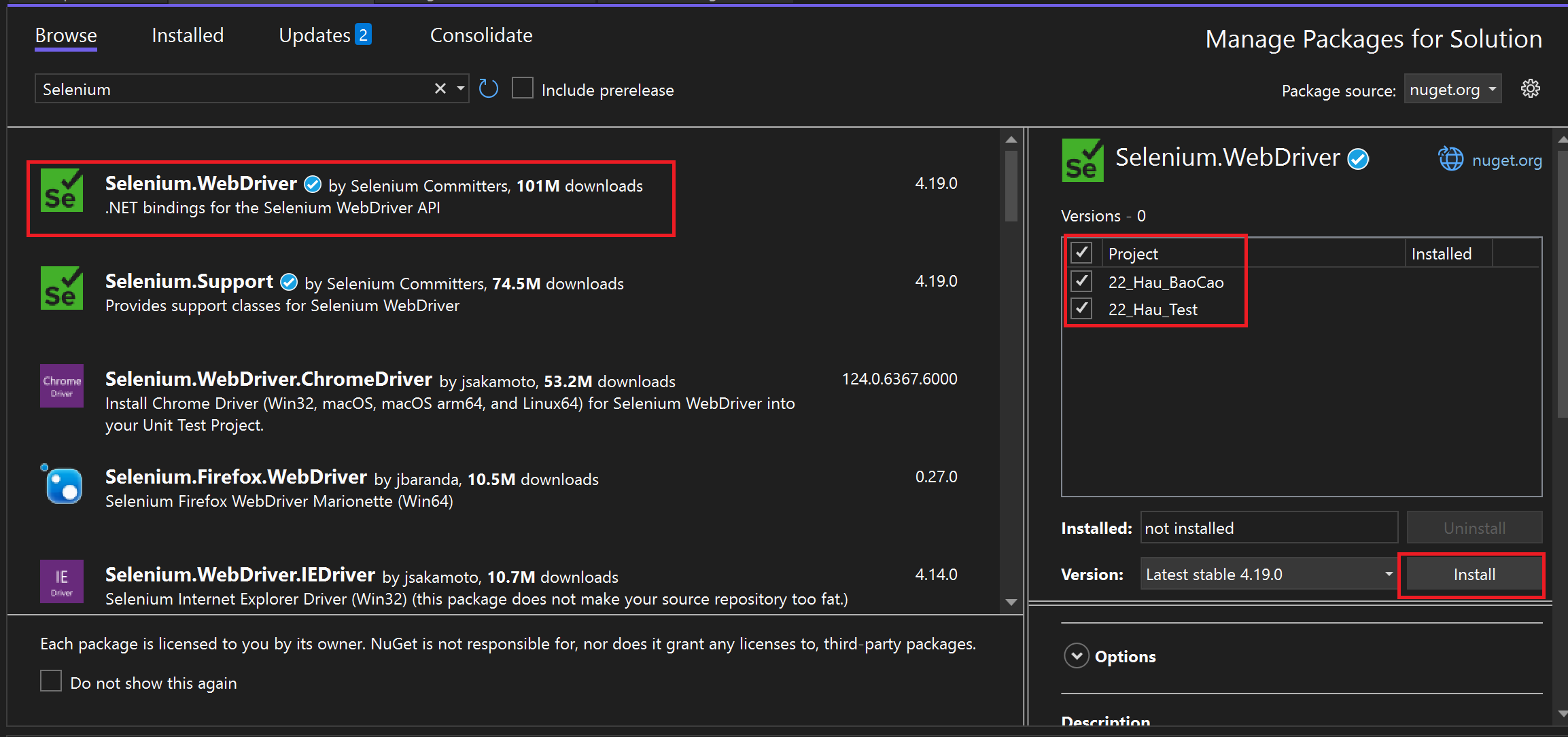
* Bước 3: Đặt tên Unit test: 22\_Hau\_Test => create



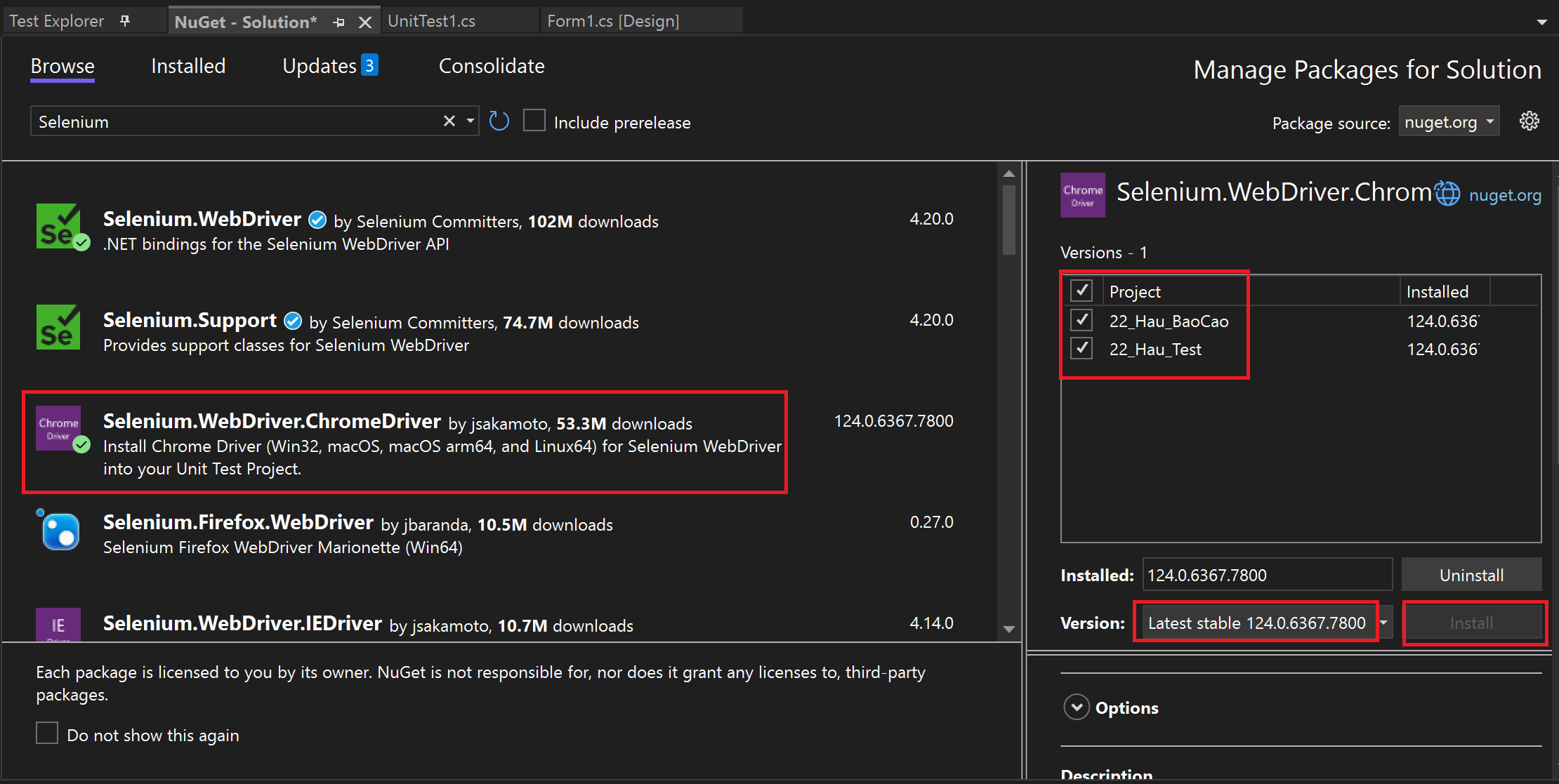
* 1. **Cài đặt Selenium Web Server**
* Bước 1: Tools => Nuget Pakage Manager => Manage Nuget Pakages for Solution…



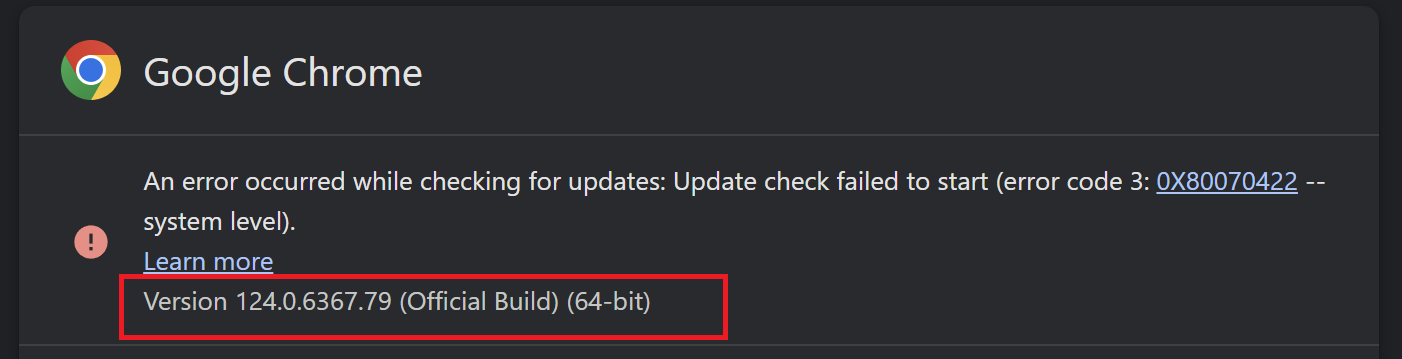
* Bước 2: Click vào Browse gõ từ khóa “Selenium” => Chọn Selenium.WebDriver => tick vào project và chọn install



* Bước 3: Chọn Selenium.WebDriver.ChromeDriver => Tích vào project => Chỉnh version: Latest stable 124.0.6367.7800 cho phù hợp với version google Chrome (124.0.6367.79) => Chọn install

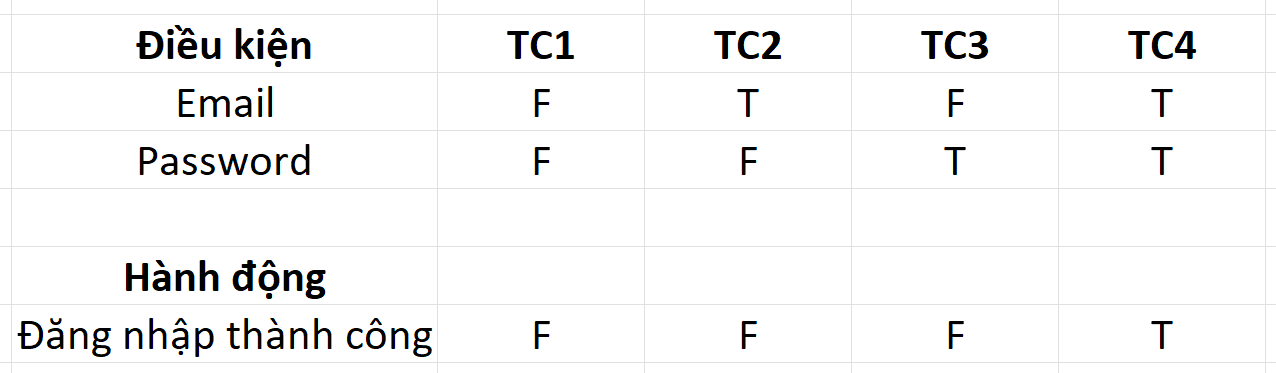


* Xem phiên bản của google chrome để cài đặt Selenium Web

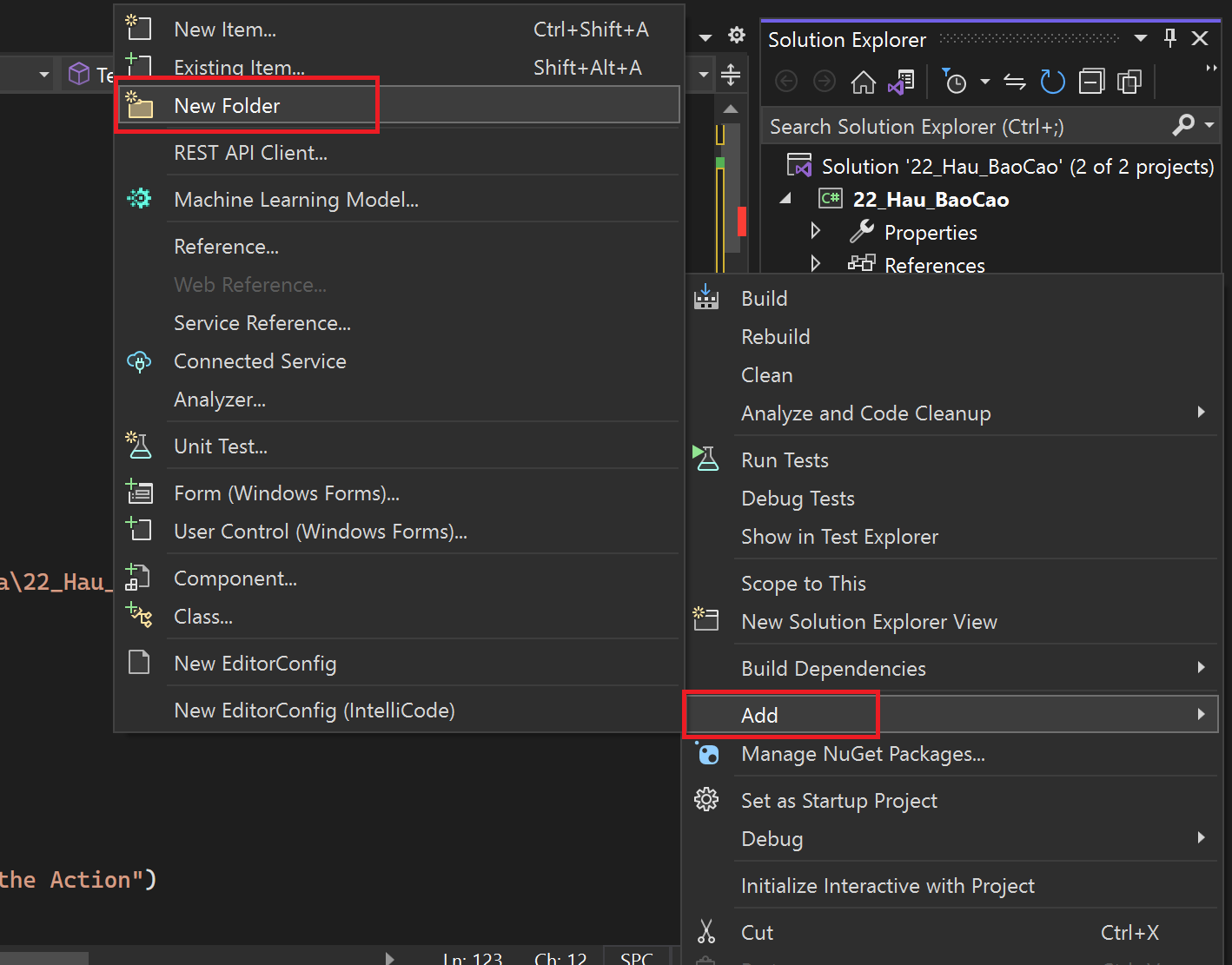


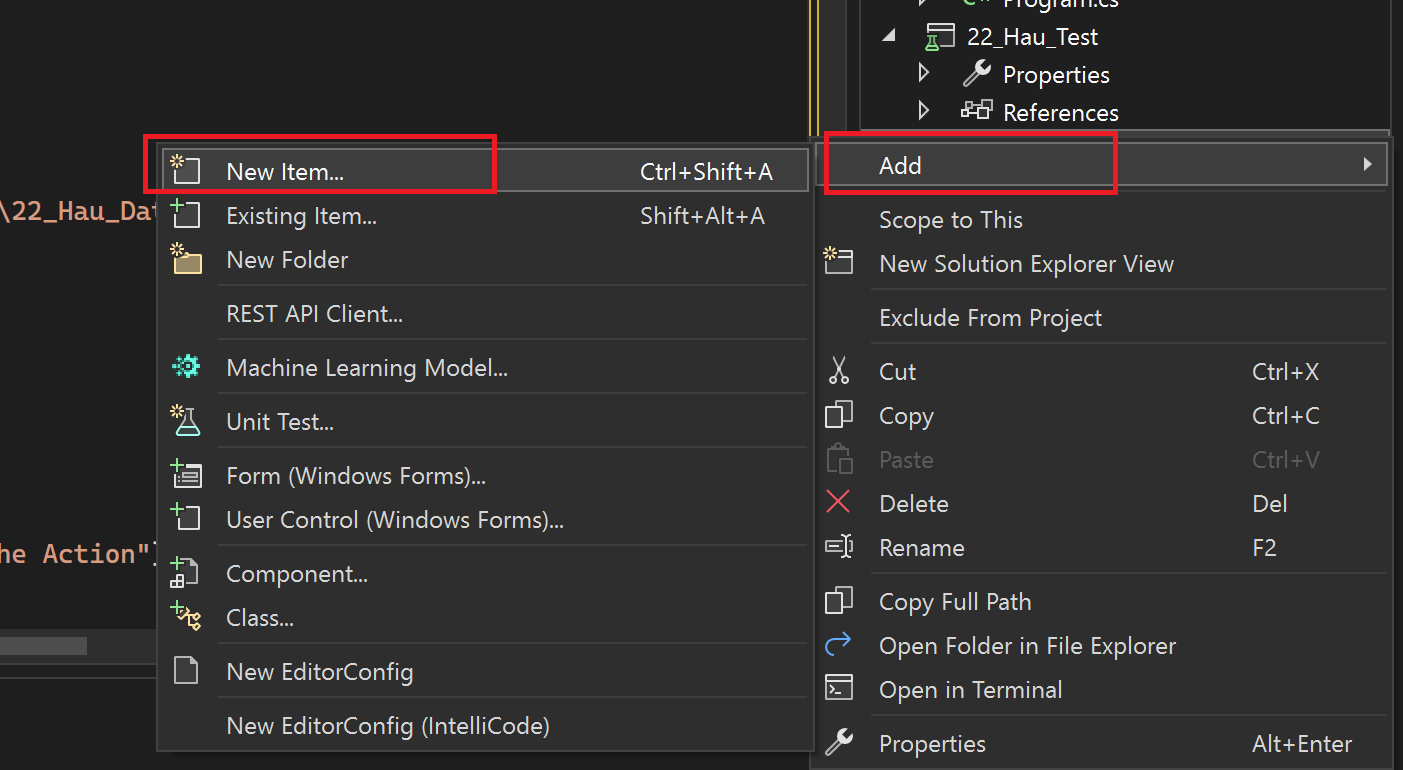
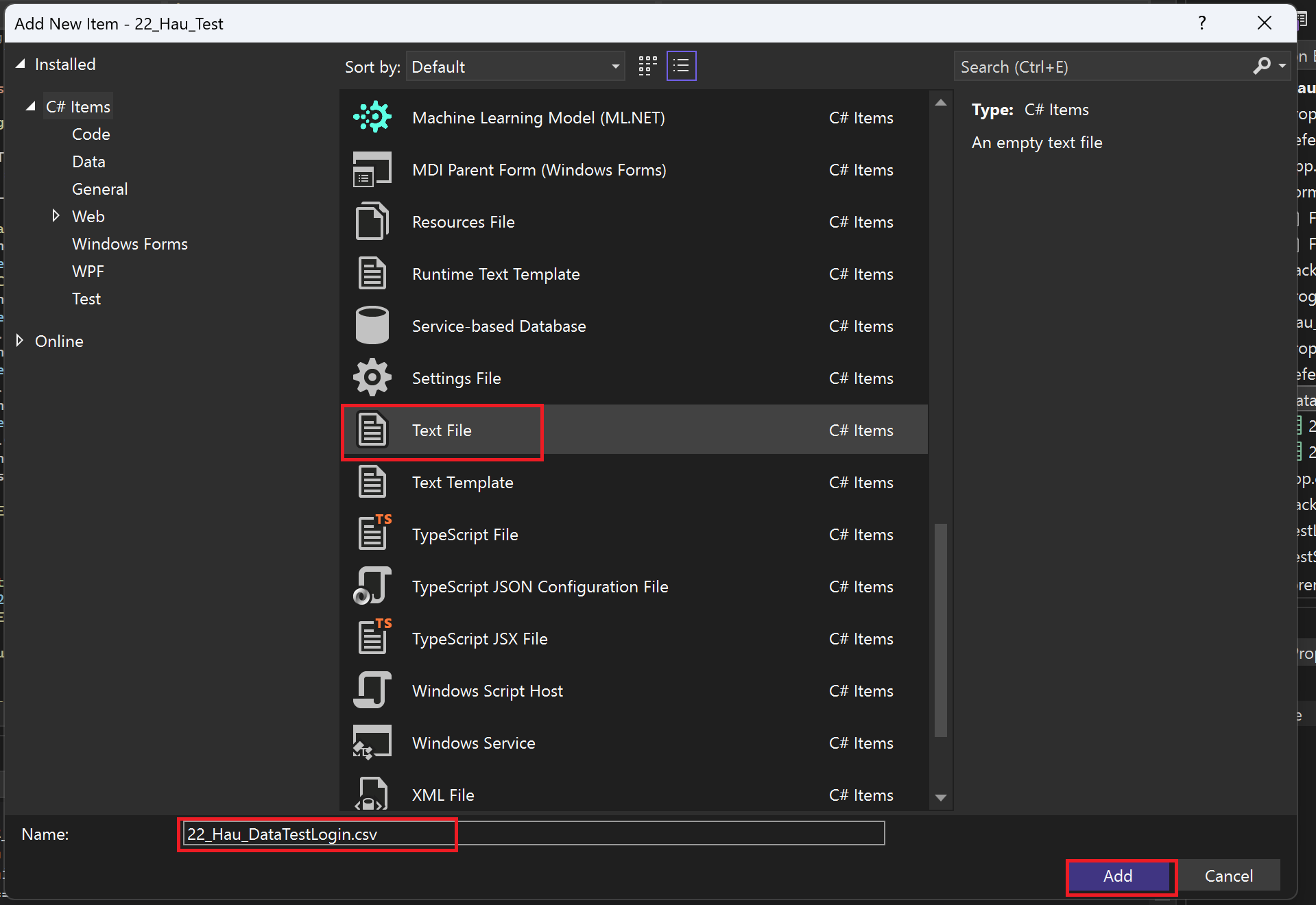
# **TRIỂN KHAI TEST CASE**

* 1. **Thực hiện test trang đăng nhập (TestLogin)**
* Bảng quyết định chức năng login

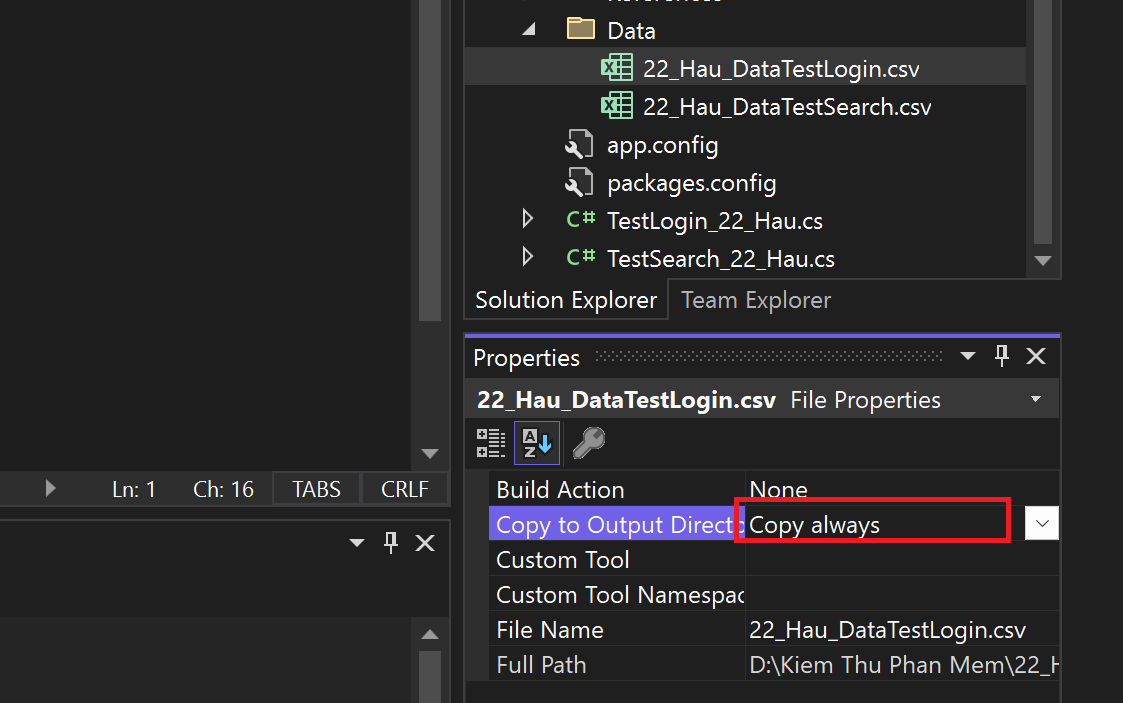


* Bước 1: Tạo thư mục Data trong project test, click chuột phải vào thư mục Data => New item => tạo tập tin 22\_Hau\_DataTestLogin.csv

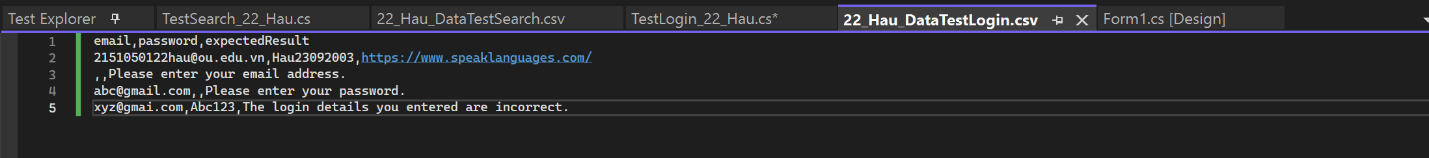


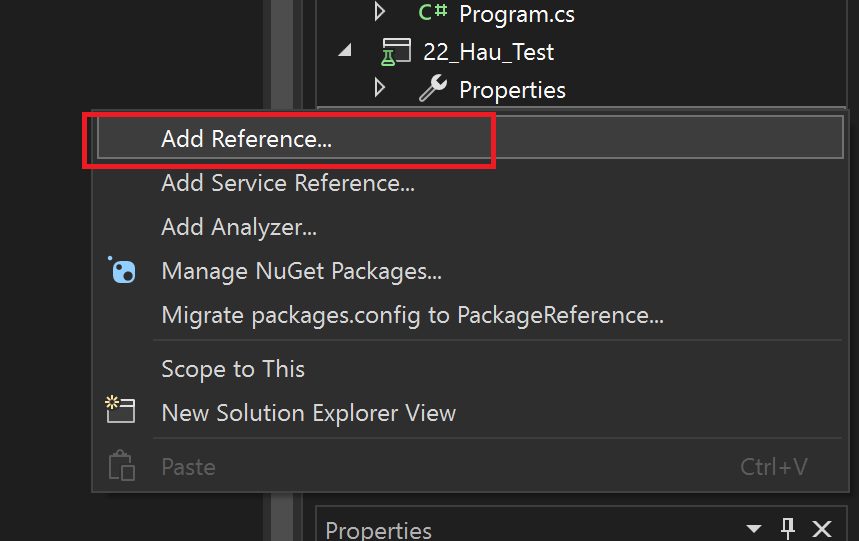
* Bước 2: Chuột phải vào tập tin 22\_Hau\_DataTestLogin.csv chọn Properties, thiết lập thuộc tính “Copy to Output Directory” thành “Copy Always” để tập tin này sẽ được sao chép khi build project

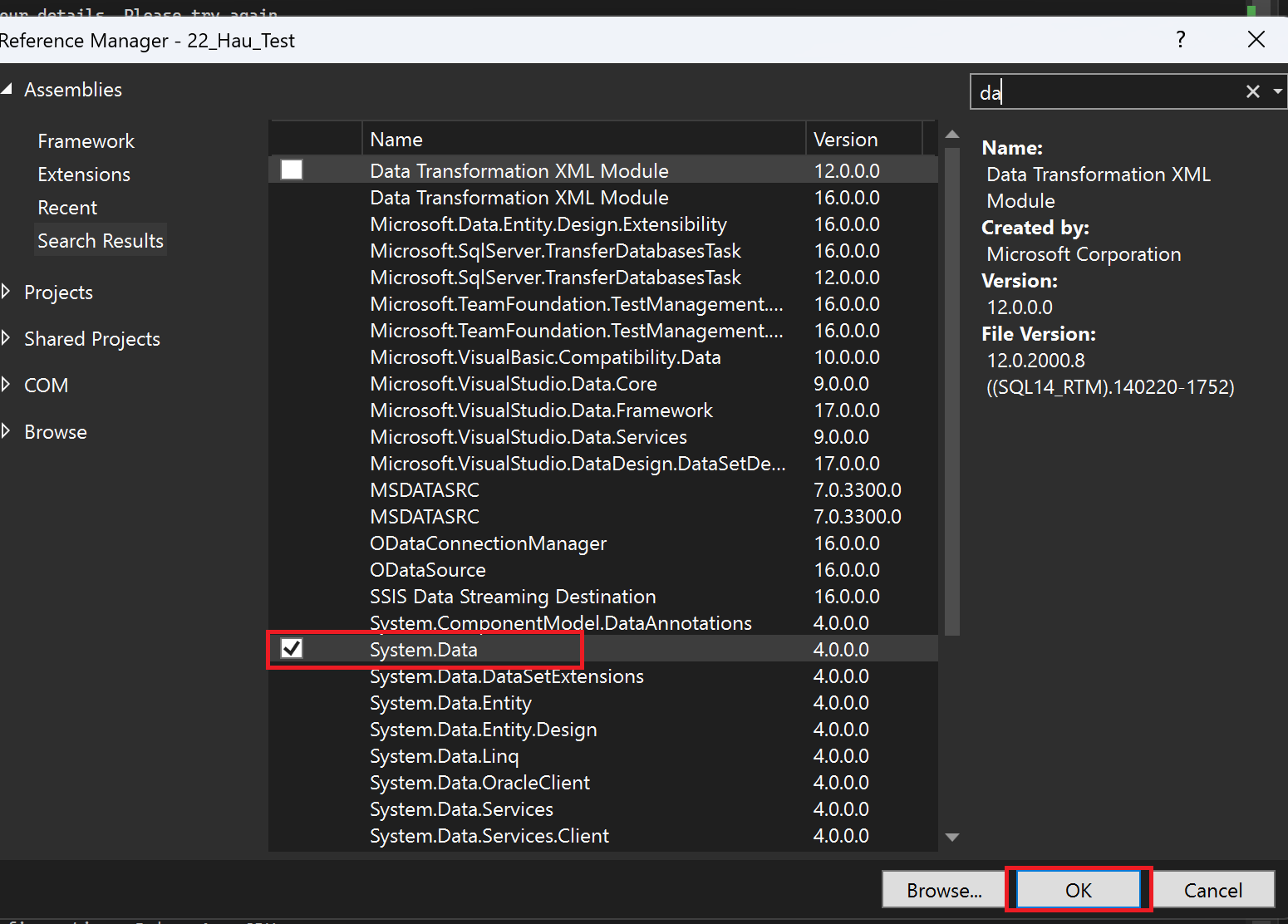


* Bước 3: Nhập dữ liệu vào tập tin DataTestLogin.csv như sau:



* Bước 4: Thêm reference System.Data vào project test

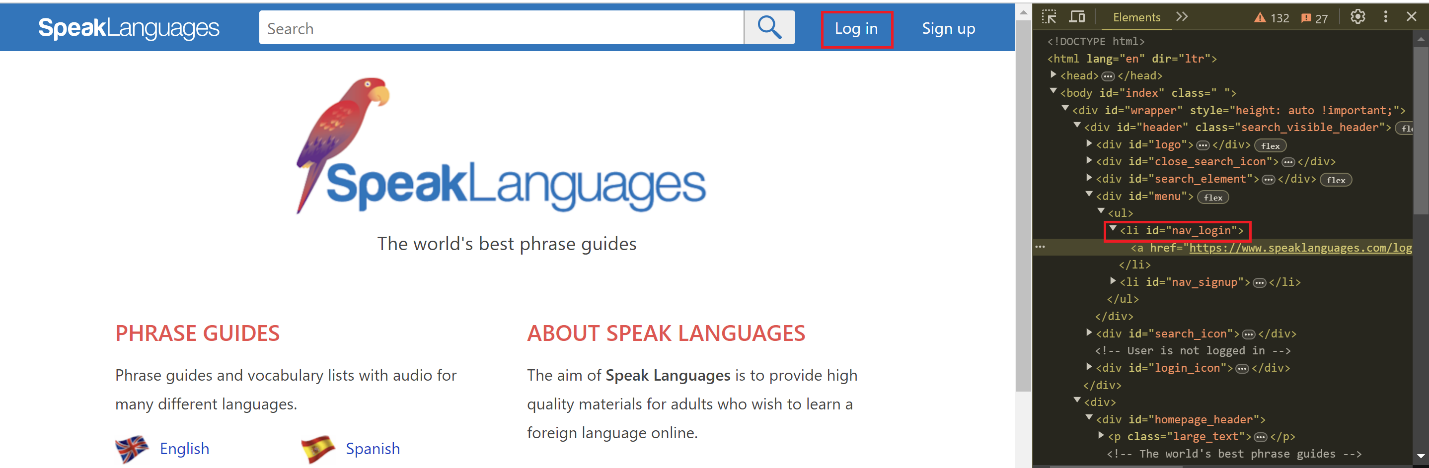




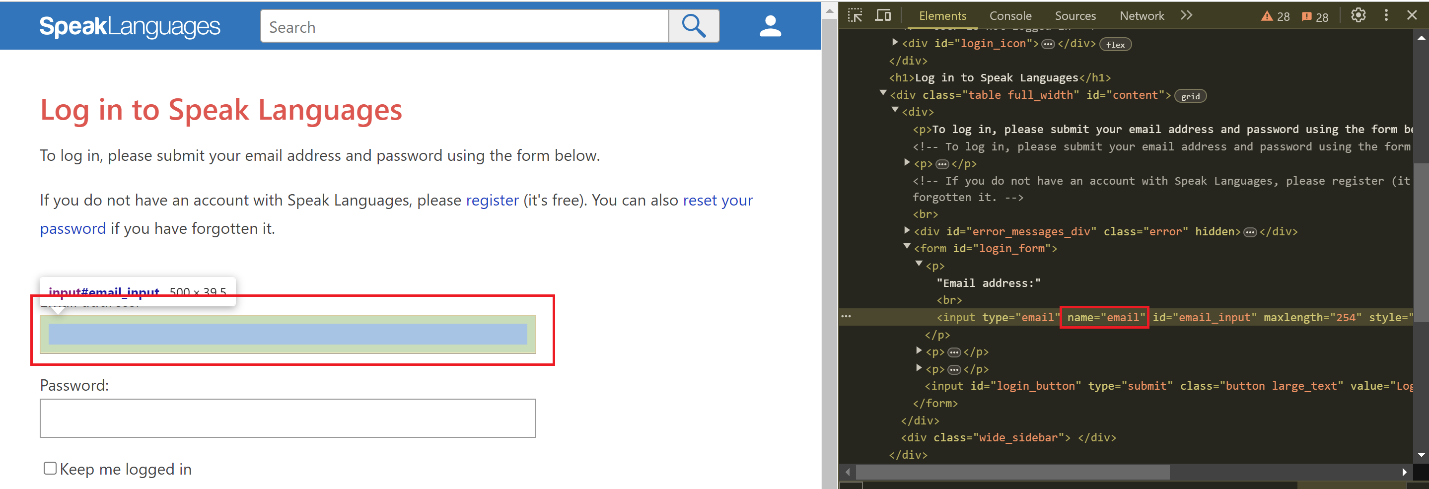
* Bước 5: Truy cập vào trang web <https://www.speaklanguages.com/>



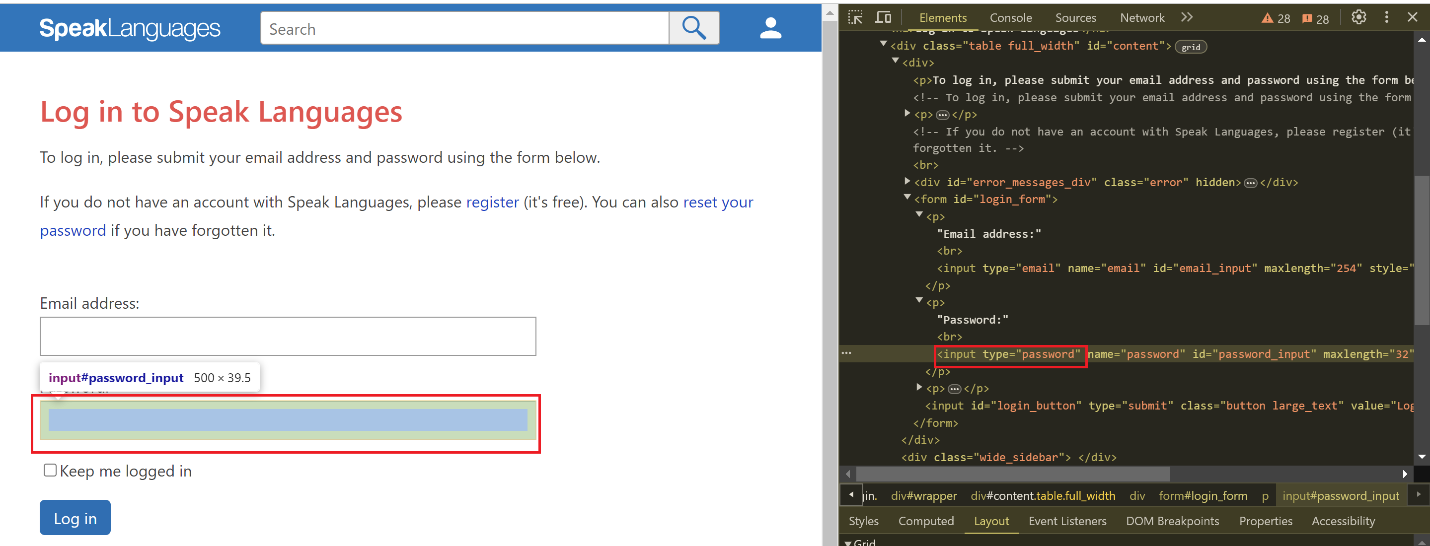
* Bước 6: Bắt Tagname mục Log in là li đầu tiên để vào trang đăng nhập



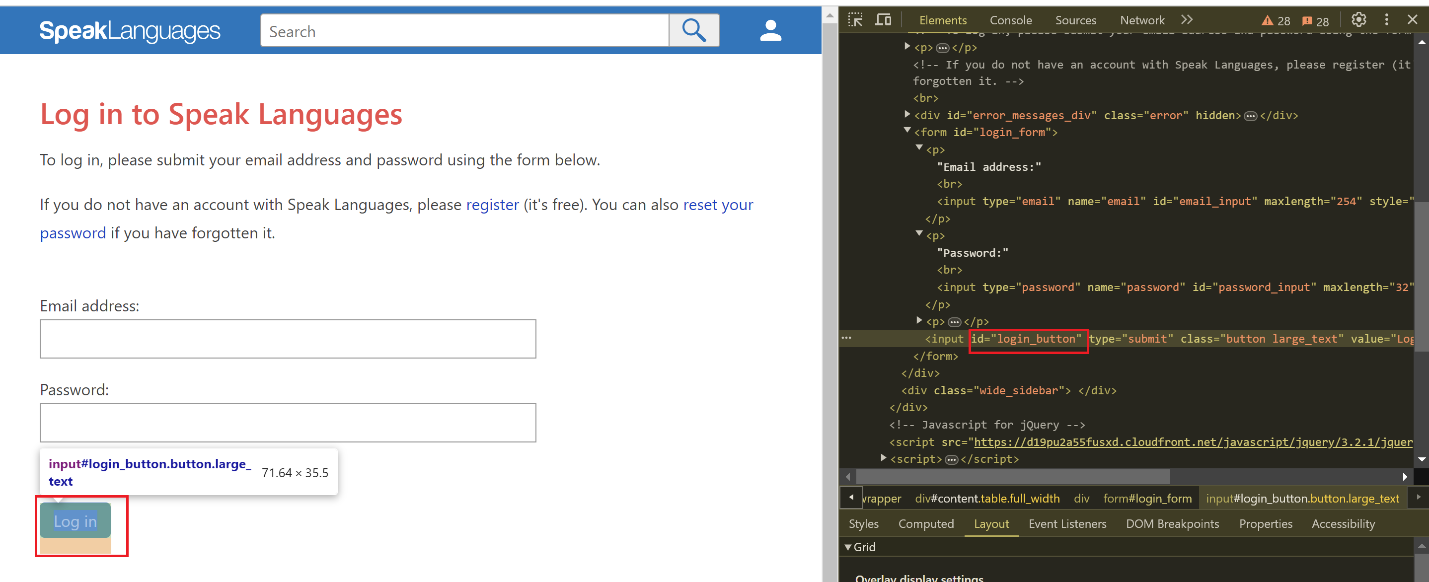
* Bước 7: Bắt name ô email



* Bước 8: Bắt CssSelector ô password



* Bước 9: Bắt id ô Sing in



* Bước 10: Khai báo driver và các biến để truyền dữ liệu từ file csv vào

public WebDriver driver\_22Hau;

public string email\_22Hau, password\_22Hau, expectedResult\_22Hau;

public TestContext TestContext { get; set; }

* Bước 11: Thiết code để đọc dữ liệu từ file csv

[DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV", @".\Data\22\_Hau\_DataTestLogin.csv", "22\_Hau\_DataTestLogin#csv", DataAccessMethod.Sequential)]

* + 1. **Test case 1: TestSuccessfulLogin\_22Hau (Đăng nhập thành công)**
* Bước 1: Thiết code cho testcase đăng nhập thành công

public void TestSuccessfulLogin\_22Hau()

{

email\_22Hau = TestContext.DataRow[0].ToString();

password\_22Hau = TestContext.DataRow[1].ToString();

expectedResult\_22Hau = TestContext.DataRow[2].ToString();

if (expectedResult\_22Hau == "https://www.speaklanguages.com/")

{

driver\_22Hau = new ChromeDriver();

driver\_22Hau.Navigate().GoToUrl("https://www.speaklanguages.com/");

System.Threading.Thread.Sleep(1000);

IWebElement element\_22Hau = driver\_22Hau.FindElement(By.TagName("li"));

element\_22Hau.Click();

System.Threading.Thread.Sleep(1000);

IWebElement element1\_22Hau = driver\_22Hau.FindElement(By.Name("email"));

element1\_22Hau.SendKeys(email\_22Hau);

IWebElement element2\_22Hau = driver\_22Hau.FindElement(By.CssSelector("input[type='password']"));

element2\_22Hau.SendKeys(password\_22Hau);

IWebElement element3\_22Hau = driver\_22Hau.FindElement(By.Id("login\_button"));

element3\_22Hau.Click();

System.Threading.Thread.Sleep(5000);

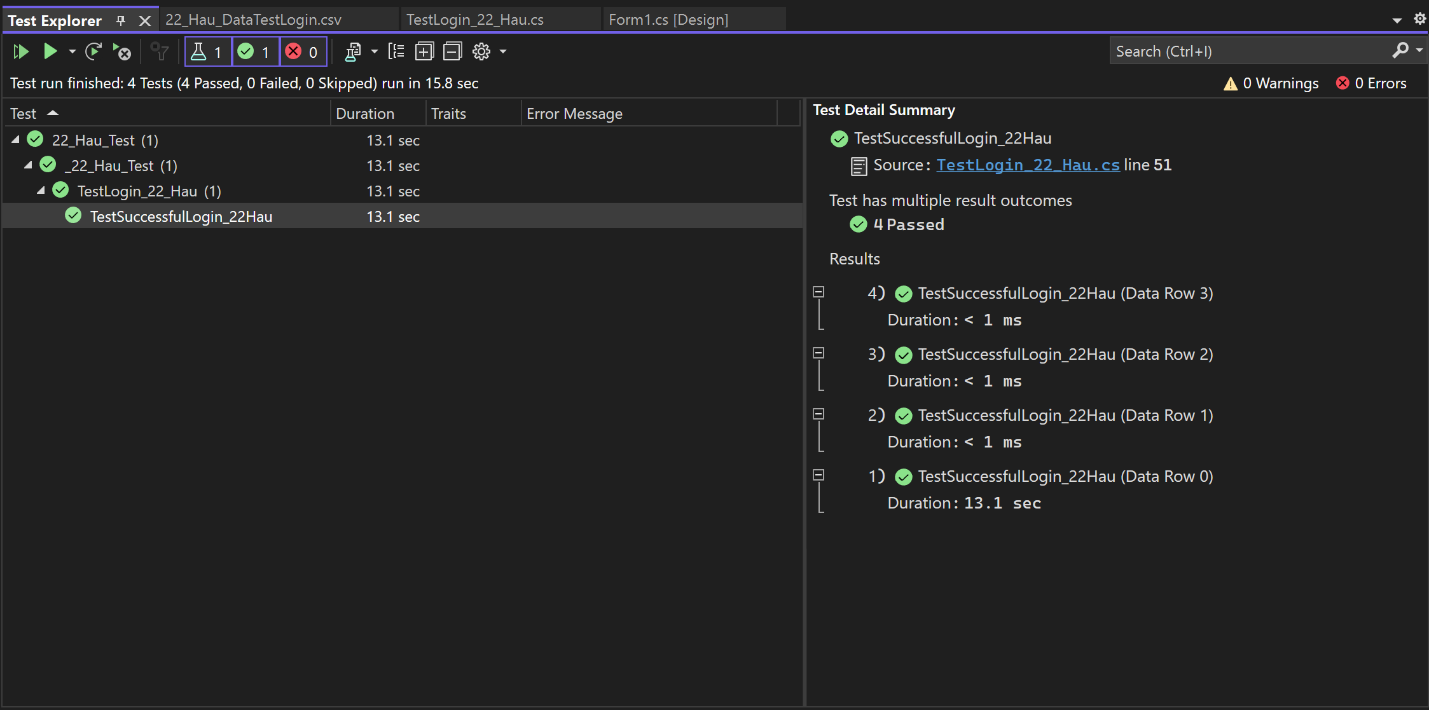
Assert.AreEqual(expectedResult\_22Hau, driver\_22Hau.Url);

driver\_22Hau.Quit();

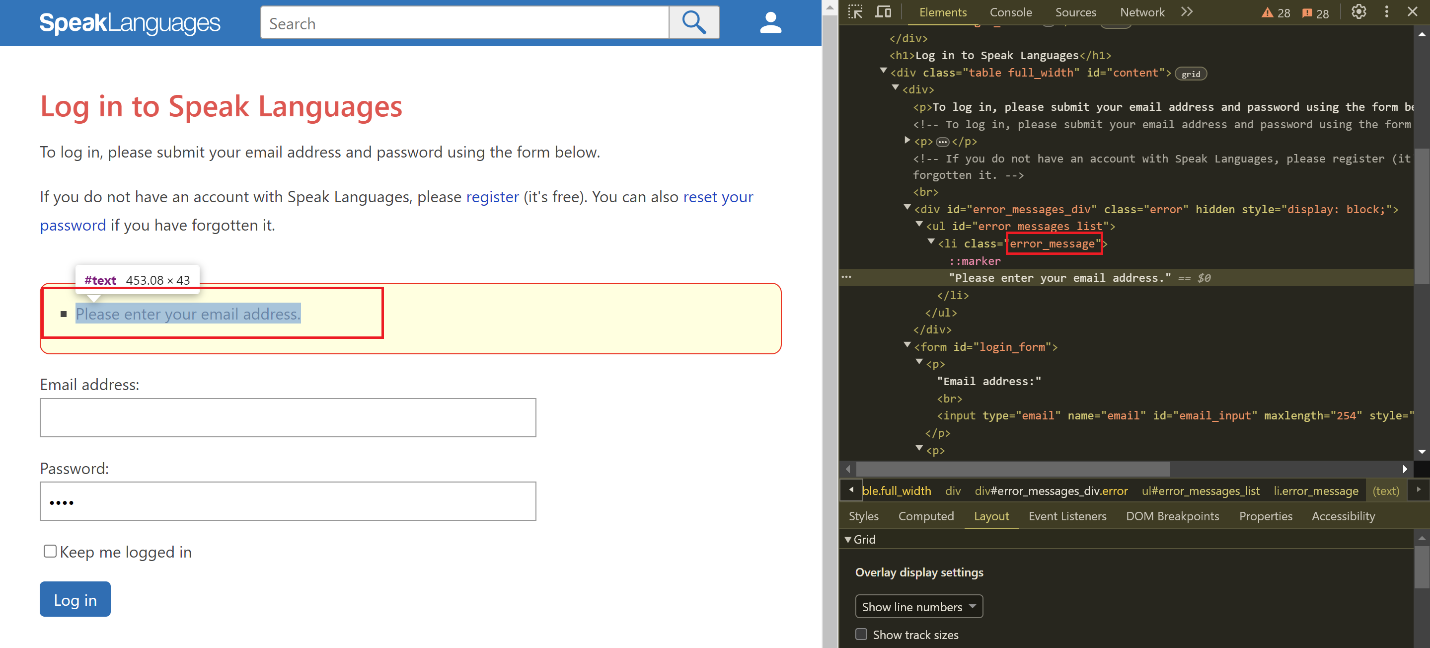
}

}

* Bước 2: Kết quả sau khi thực thi testcase



* + 1. **Test case 2: TestFailedLogin\_NoEmail\_22Hau (Đăng nhập không nhập email)**
* Bước 1: Bắt class của dòng thông báo không nhập email



* Bước 2: Thiết lập code cho testcase đăng nhập không nhập email

public void TestFailedLogin\_NoEmail\_22Hau()

{

email\_22Hau = TestContext.DataRow[0].ToString();

password\_22Hau = TestContext.DataRow[1].ToString();

expectedResult\_22Hau = TestContext.DataRow[2].ToString();

if (email\_22Hau == "")

{

driver\_22Hau = new ChromeDriver();

driver\_22Hau.Navigate().GoToUrl("https://www.speaklanguages.com/");

System.Threading.Thread.Sleep(1000);

IWebElement element\_22Hau = driver\_22Hau.FindElement(By.TagName("li"));

element\_22Hau.Click();

System.Threading.Thread.Sleep(1000);

IWebElement element1\_22Hau = driver\_22Hau.FindElement(By.Name("email"));

element1\_22Hau.SendKeys(email\_22Hau);

IWebElement element2\_22Hau = driver\_22Hau.FindElement(By.CssSelector("input[type='password']"));

element2\_22Hau.SendKeys(password\_22Hau);

IWebElement element3\_22Hau = driver\_22Hau.FindElement(By.Id("login\_button"));

element3\_22Hau.Click();

System.Threading.Thread.Sleep(5000);

IWebElement element4\_22Hau = driver\_22Hau.FindElement(By.ClassName("error\_message"));

element4\_22Hau.Click();

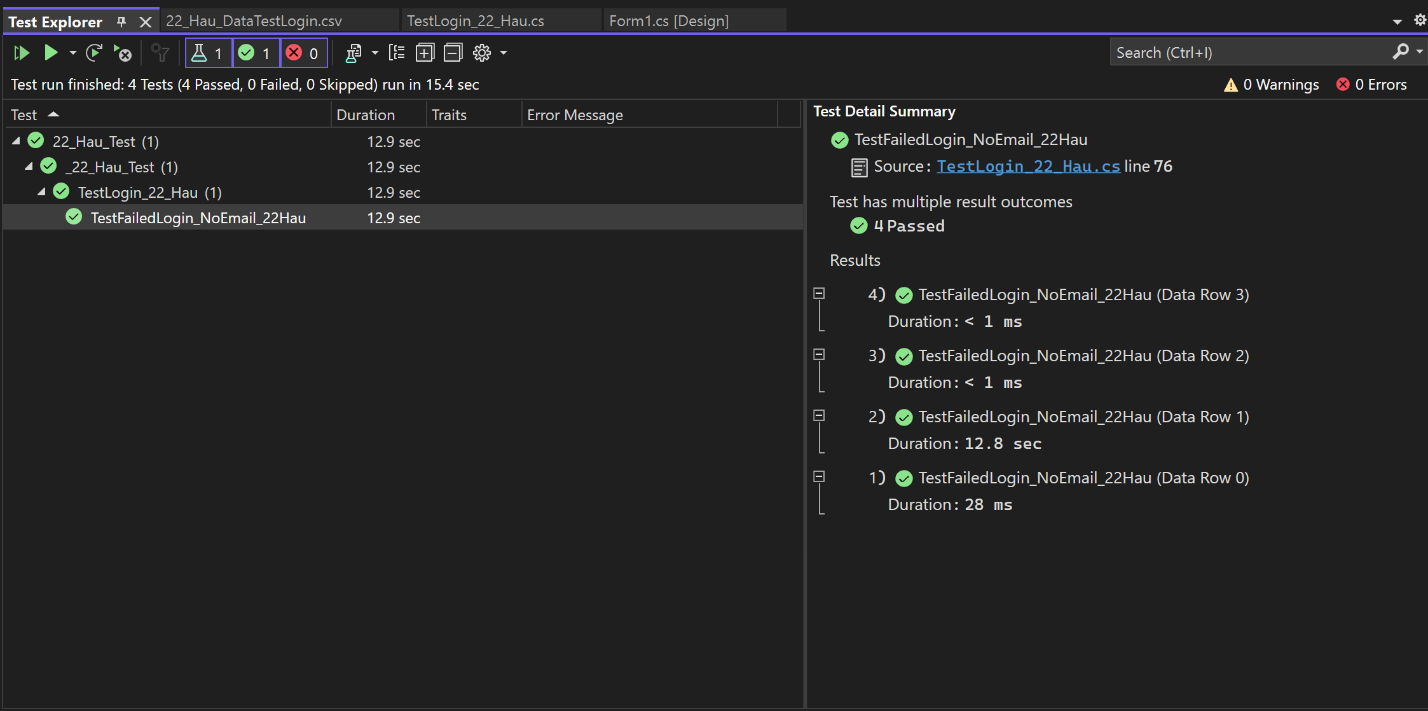
Assert.AreEqual(expectedResult\_22Hau, element4\_22Hau.Text);

driver\_22Hau.Quit();

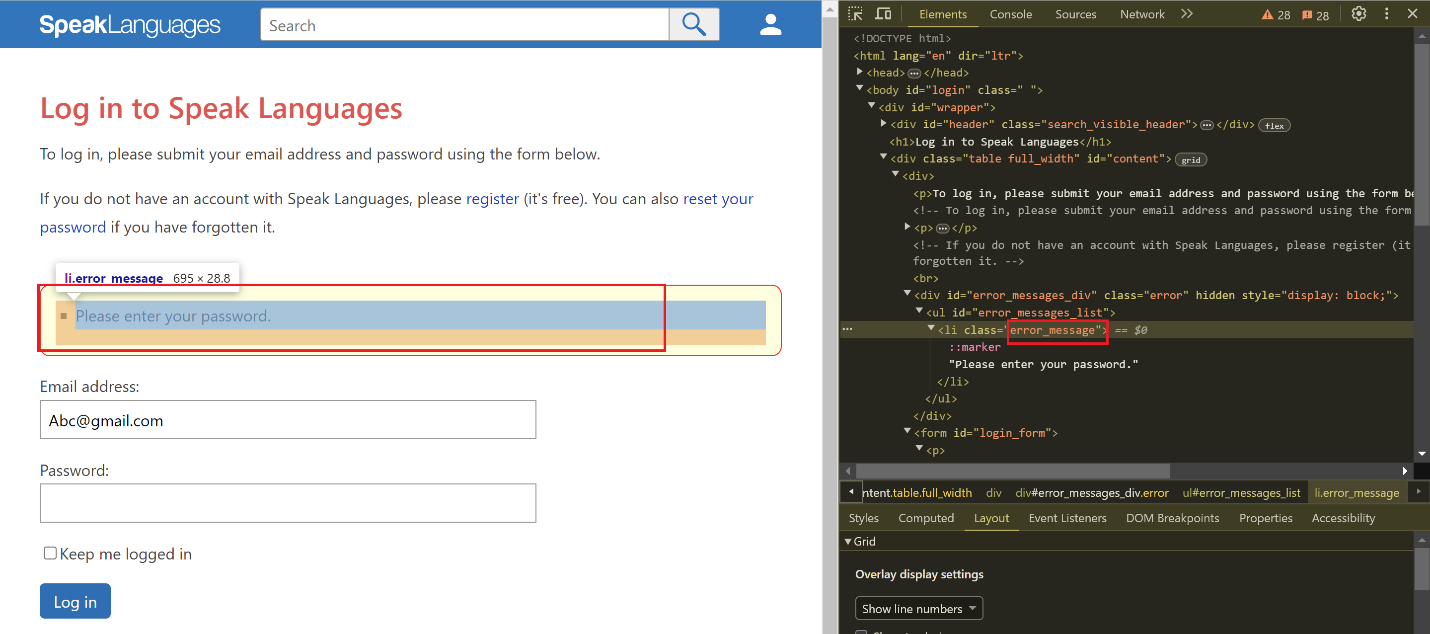
}

}

* Bước 3: Kết quả sau khi thực thi testcase



* + 1. **Test case 3: TestFailedLogin\_NoPassword\_22Hau (Đăng nhập không nhập password)**
* Bước 1: Bắt class của dòng thông báo không nhập password



* Bước 2: Thiết lập code cho testcase đăng nhập không nhập password

public void TestFailedLogin\_NoPassword\_22Hau()

{

email\_22Hau = TestContext.DataRow[0].ToString();

password\_22Hau = TestContext.DataRow[1].ToString();

expectedResult\_22Hau = TestContext.DataRow[2].ToString();

if (expectedResult\_22Hau == "Please enter your password.")

{

driver\_22Hau = new ChromeDriver();

driver\_22Hau.Navigate().GoToUrl("https://www.speaklanguages.com/");

System.Threading.Thread.Sleep(1000);

IWebElement element\_22Hau = driver\_22Hau.FindElement(By.TagName("li"));

element\_22Hau.Click();

System.Threading.Thread.Sleep(1000);

IWebElement element1\_22Hau = driver\_22Hau.FindElement(By.Name("email"));

element1\_22Hau.SendKeys(email\_22Hau);

IWebElement element2\_22Hau = driver\_22Hau.FindElement(By.CssSelector("input[type='password']"));

element2\_22Hau.SendKeys(password\_22Hau);

IWebElement element3\_22Hau = driver\_22Hau.FindElement(By.Id("login\_button"));

element3\_22Hau.Click();

System.Threading.Thread.Sleep(5000);

IWebElement element4\_22Hau = driver\_22Hau.FindElement(By.ClassName("error\_message"));

element4\_22Hau.Click();

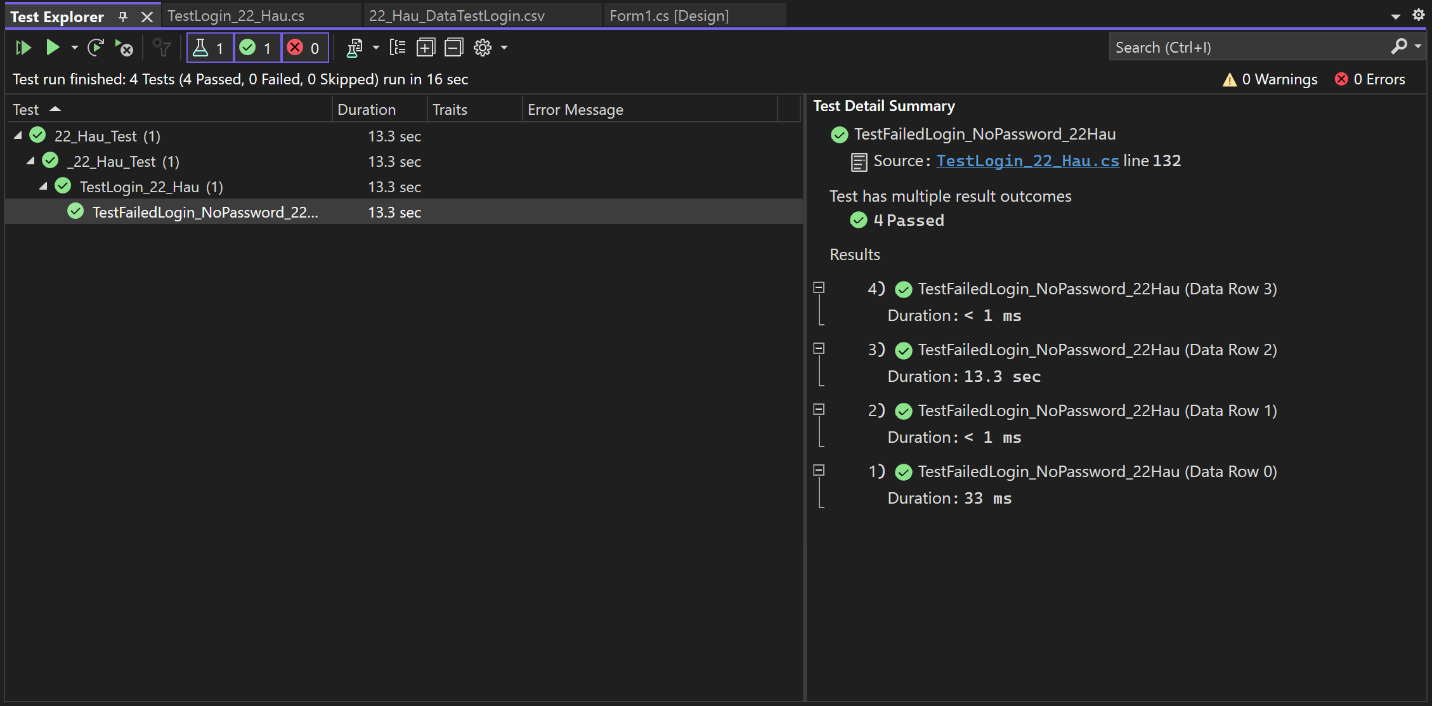
Assert.AreEqual(expectedResult\_22Hau, element4\_22Hau.Text);

driver\_22Hau.Quit();

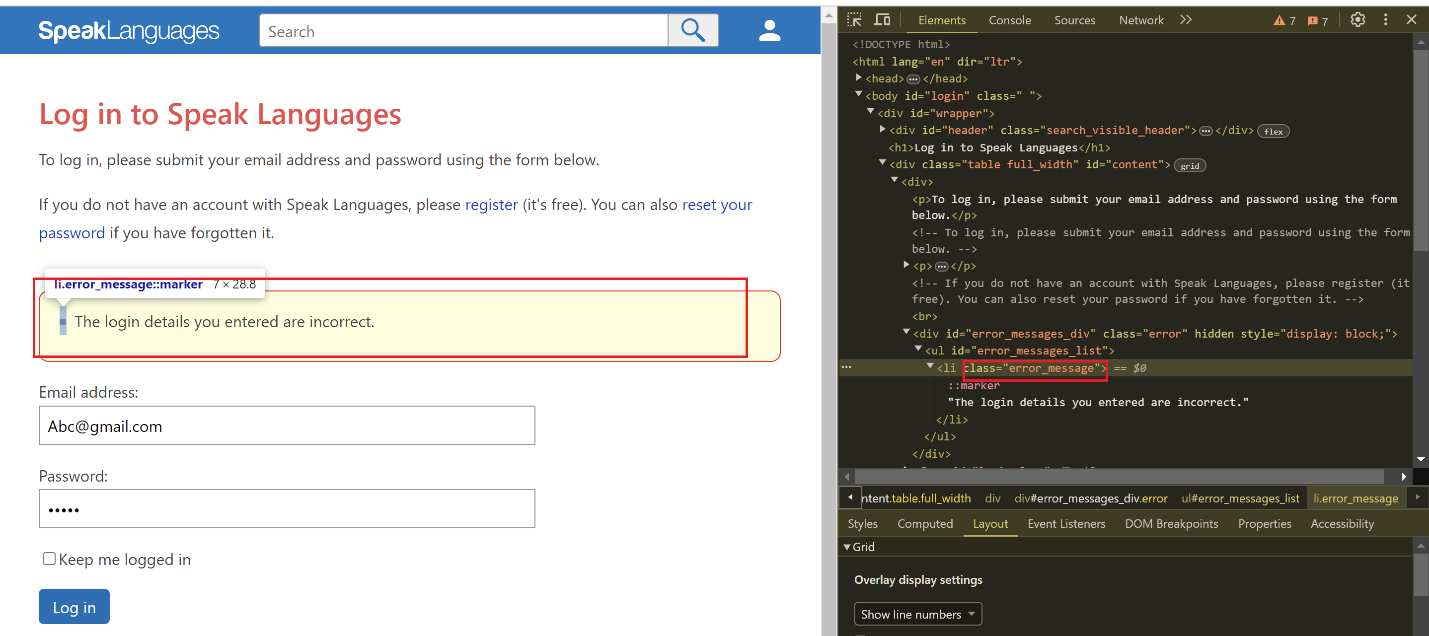
}

}

* Bước 3: Kết quả sau khi thực thi testcase



* + 1. **Test case 4: TestFailedLogin\_WrongPasswordorLogin\_22Hau(Đăng nhập sai email hoặc password)**
* Bước 1: Bắt id của dòng thông báo nhập sai



* Bước 2: Thiết lập code cho testcase đăng nhập sai

public void TestFailedLogin\_WrongPasswordorLogin\_22Hau()

{

email\_22Hau = TestContext.DataRow[0].ToString();

password\_22Hau = TestContext.DataRow[1].ToString();

expectedResult\_22Hau = TestContext.DataRow[2].ToString();

if (expectedResult\_22Hau == "The login details you entered are incorrect.")

{

driver\_22Hau = new ChromeDriver();

driver\_22Hau.Navigate().GoToUrl("https://www.speaklanguages.com/");

System.Threading.Thread.Sleep(1000);

IWebElement element\_22Hau = driver\_22Hau.FindElement(By.TagName("li"));

element\_22Hau.Click();

System.Threading.Thread.Sleep(1000);

IWebElement element1\_22Hau = driver\_22Hau.FindElement(By.Name("email"));

element1\_22Hau.SendKeys(email\_22Hau);

IWebElement element2\_22Hau = driver\_22Hau.FindElement(By.CssSelector("input[type='password']"));

element2\_22Hau.SendKeys(password\_22Hau);

IWebElement element3\_22Hau = driver\_22Hau.FindElement(By.Id("login\_button"));

element3\_22Hau.Click();

System.Threading.Thread.Sleep(5000);

IWebElement element4\_22Hau = driver\_22Hau.FindElement(By.ClassName("error\_message"));

element4\_22Hau.Click();

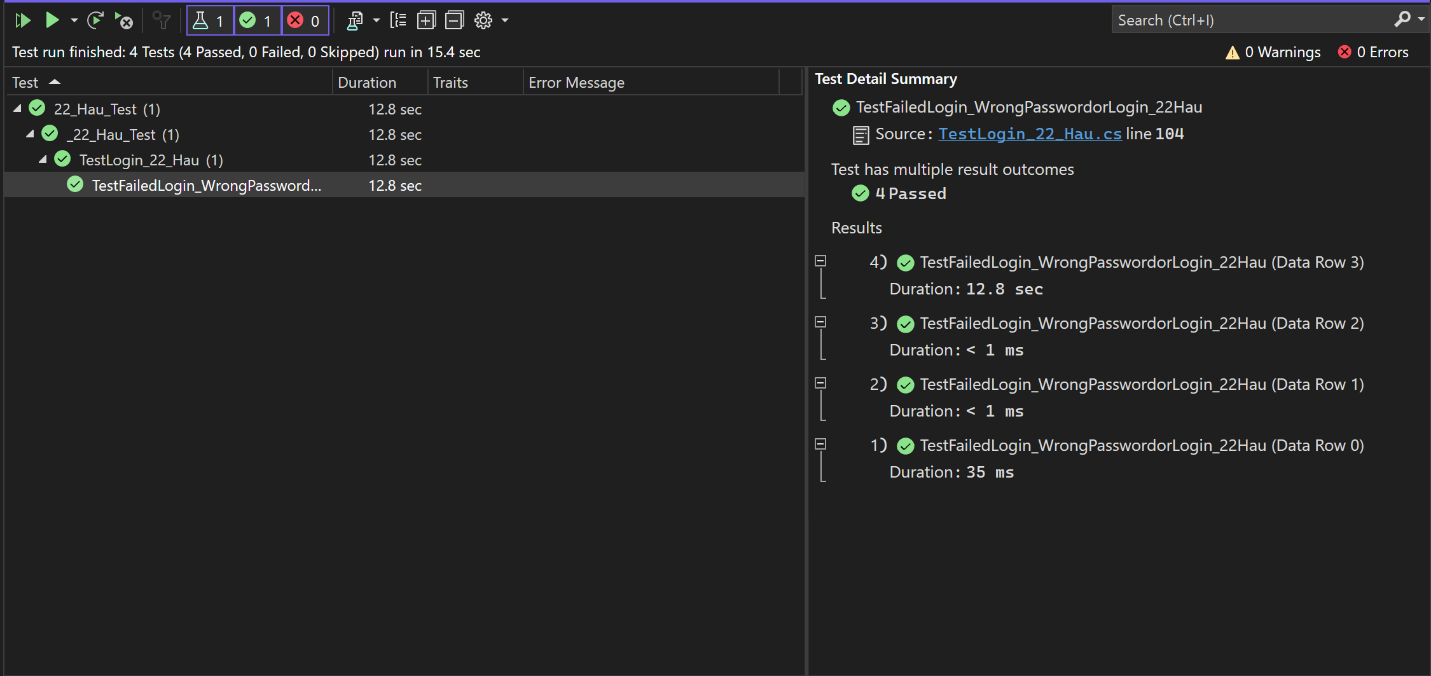
Assert.AreEqual(expectedResult\_22Hau, element4\_22Hau.Text);

driver\_22Hau.Quit();

}

}

* Bước 3: Kết quả sau khi thực thi testcase



* + 1. **Tổng hợp code chạy cho 4 test case**
* Bước 1: Viết code chạy hết 4 testcase

public void TestLogin\_22Hau()

{

email\_22Hau = TestContext.DataRow[0].ToString();

password\_22Hau = TestContext.DataRow[1].ToString();

expectedResult\_22Hau = TestContext.DataRow[2].ToString();

driver\_22Hau = new ChromeDriver();

driver\_22Hau.Navigate().GoToUrl("https://www.speaklanguages.com/");

System.Threading.Thread.Sleep(1000);

IWebElement element\_22Hau = driver\_22Hau.FindElement(By.TagName("li"));

element\_22Hau.Click();

System.Threading.Thread.Sleep(1000);

IWebElement element1\_22Hau = driver\_22Hau.FindElement(By.Name("email"));

element1\_22Hau.SendKeys(email\_22Hau);

System.Threading.Thread.Sleep(1000);

IWebElement element2\_22Hau = driver\_22Hau.FindElement(By.CssSelector("input[type='password']"));

element2\_22Hau.SendKeys(password\_22Hau);

System.Threading.Thread.Sleep(1000);

IWebElement element3\_22Hau = driver\_22Hau.FindElement(By.Id("login\_button"));

element3\_22Hau.Click();

System.Threading.Thread.Sleep(5000);

if (expectedResult\_22Hau == "https://www.speaklanguages.com/")

{

Assert.AreEqual(expectedResult\_22Hau, driver\_22Hau.Url);

}

else

{

IWebElement element4\_22Hau = driver\_22Hau.FindElement(By.ClassName("error\_message"));

element4\_22Hau.Click();

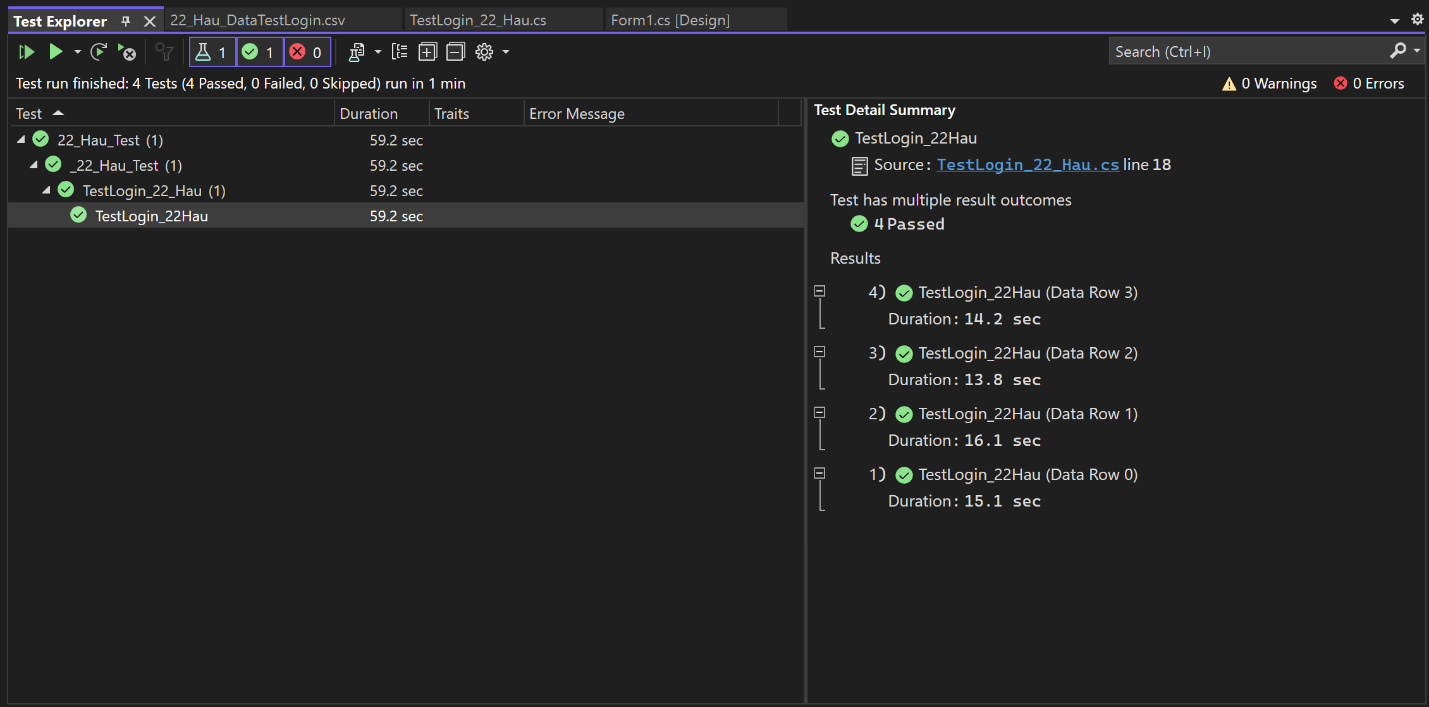
Assert.AreEqual(expectedResult\_22Hau, element4\_22Hau.Text);

}

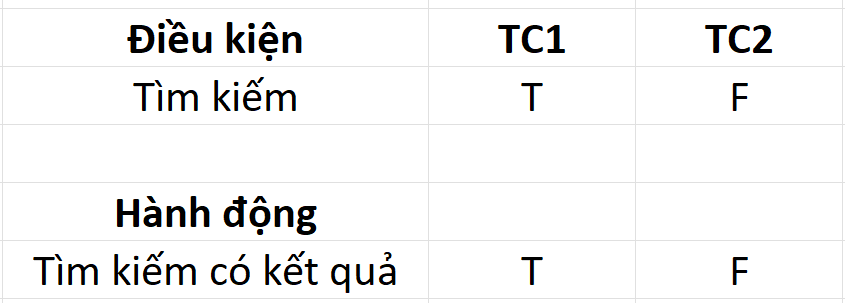
driver\_22Hau.Quit();

}

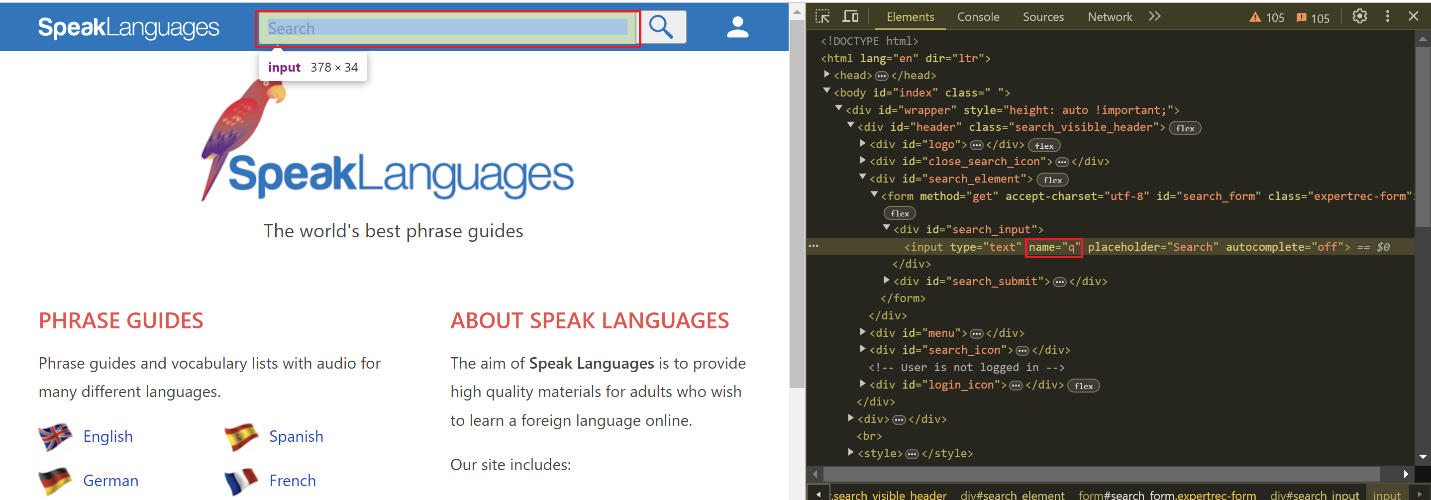
* Bước 2: Kết quả sau khi thự hiện testcase



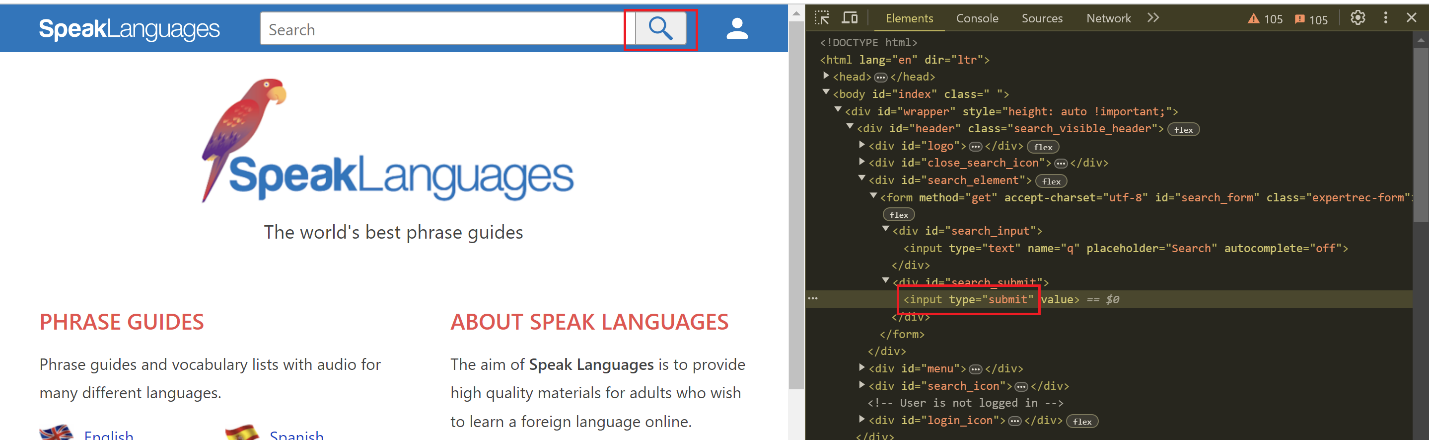
* 1. **Thực hiên test button tìm kiếm (TestSearch)**
* Bảng quyết định chức năng tìm kiếm



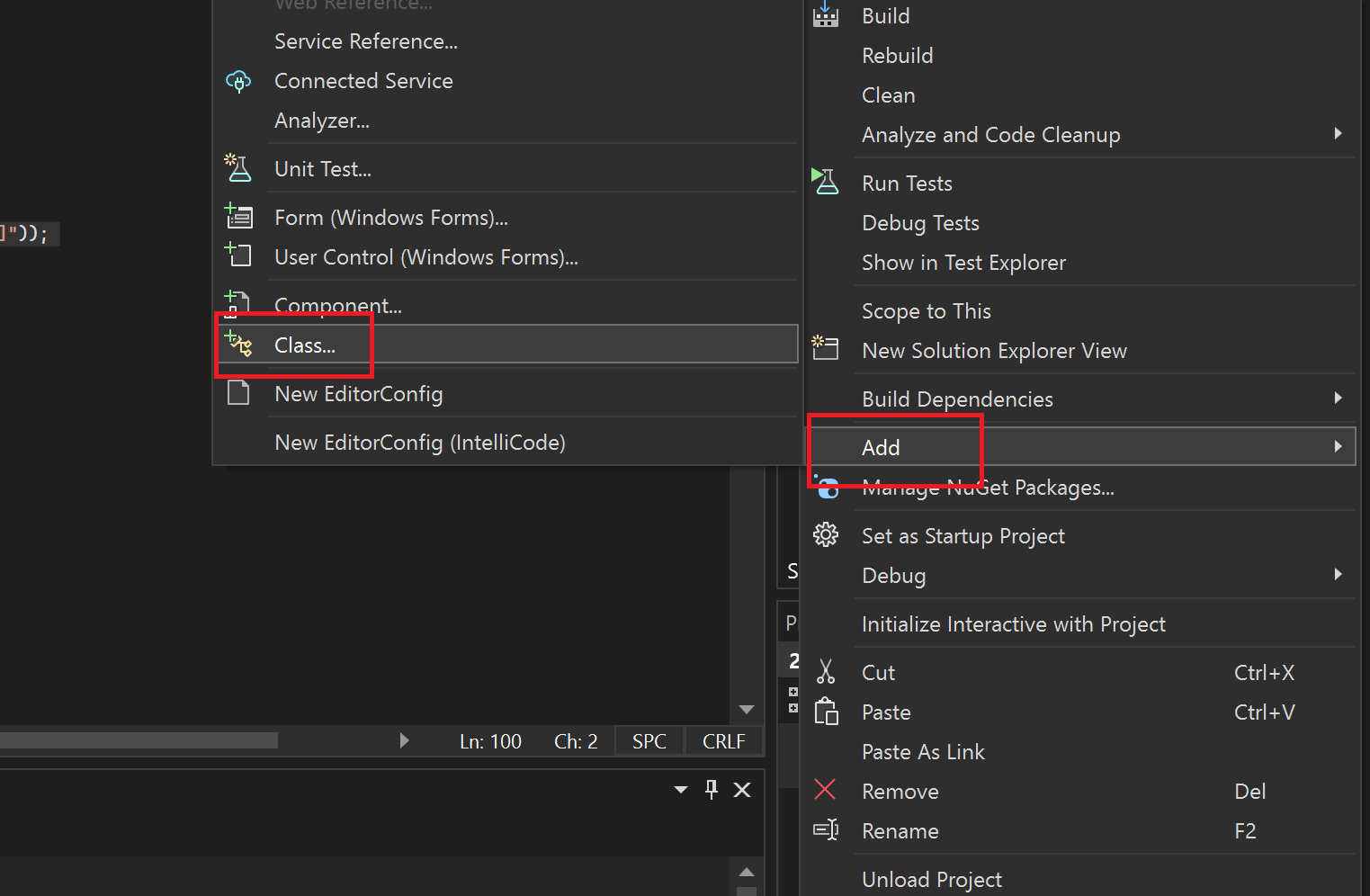
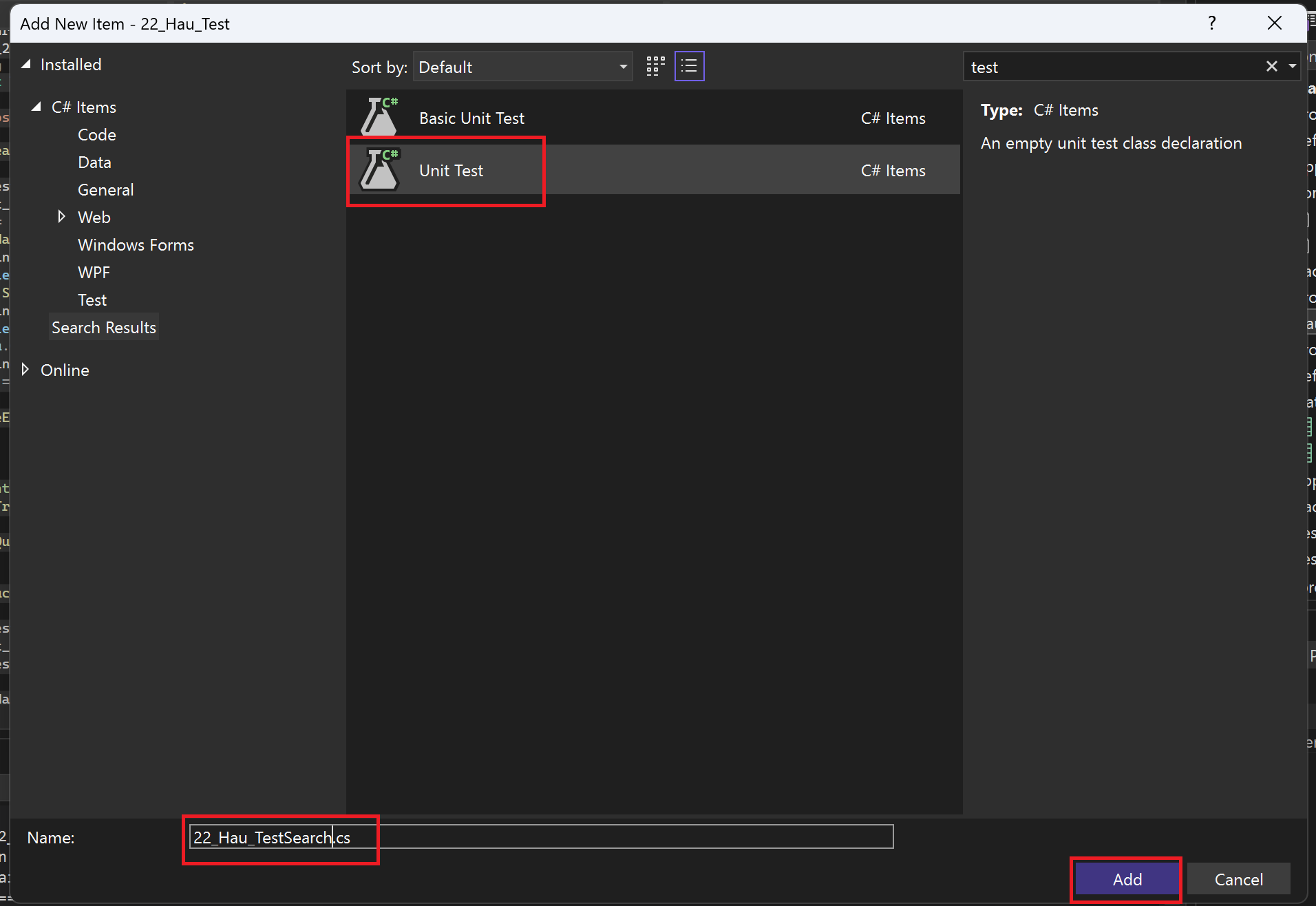
* Bước 1: Thực hiện bắt name của ô tìm kiếm



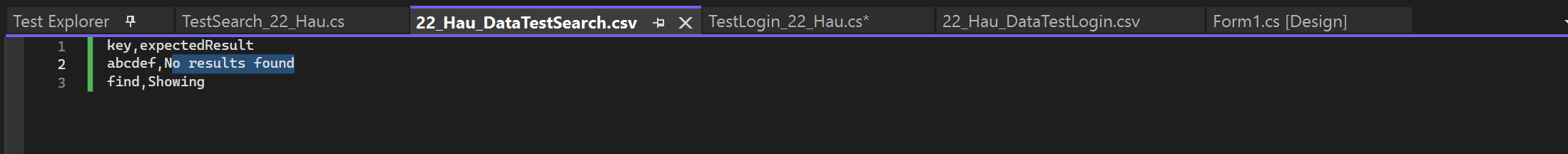
* Bước 2: Thực hiện bắt CssSelector của button tìm kiếm



* Bước 3: Tạo unit test mới chuột phải 22\_Hau\_Test => add => class chọn unit test và đăt tên: 22\_Hau\_TestSearch

* Bước 4: Tạo tập tin 22\_Hau\_DataTestSearch.csv, thiết lập thuộc tính Copy Always cho tập tin(tương tư như TestLogin)



* Bước 5: Khai báo driver, TestContext và các biến để truyền dữ liệu từ file csv vào

public WebDriver driver\_22Hau;

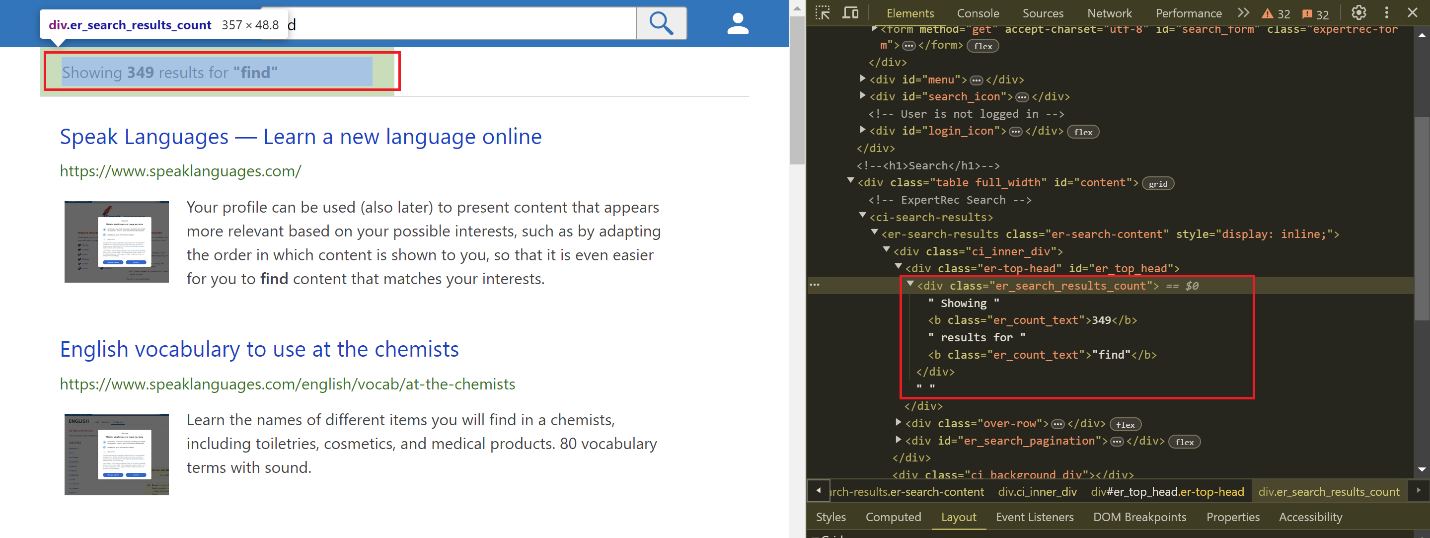
public string key\_22Hau, expectedResult\_22Hau;

public TestContext TestContext { get; set; }

* Bước 6: Thiết code để đọc dữ liệu từ file csv

[DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV", @".\Data\22\_Hau\_DataTestSearch.csv", "22\_Hau\_DataTestSearch#csv", DataAccessMethod.Sequential)]

* + 1. **Test case 1: TestSuccessfullSearch\_22Hau()**
* Bước 1: Bắt Xpath của dòng thông báo tìm kiếm thành công



* Bước 2: Thiết lập code cho testcase tìm kiếm thành công

public void TestSuccessfulSearch\_22Hau()

{

key\_22Hau = TestContext.DataRow[0].ToString();

expectedResult\_22Hau = TestContext.DataRow[1].ToString();

if (expectedResult\_22Hau == "Showing")

{

driver\_22Hau = new ChromeDriver();

driver\_22Hau.Navigate().GoToUrl("https://www.speaklanguages.com/");

System.Threading.Thread.Sleep(1000);

IWebElement element\_22Hau = driver\_22Hau.FindElement(By.Name("q"));

element\_22Hau.SendKeys(key\_22Hau);

IWebElement element1\_22Hau = driver\_22Hau.FindElement(By.CssSelector("input[type='submit']"));

element1\_22Hau.Click();

System.Threading.Thread.Sleep(2000);

IWebElement element2\_22Hau = driver\_22Hau.FindElement(By.XPath("//\*[@id=\"er\_top\_head\"]/div"));

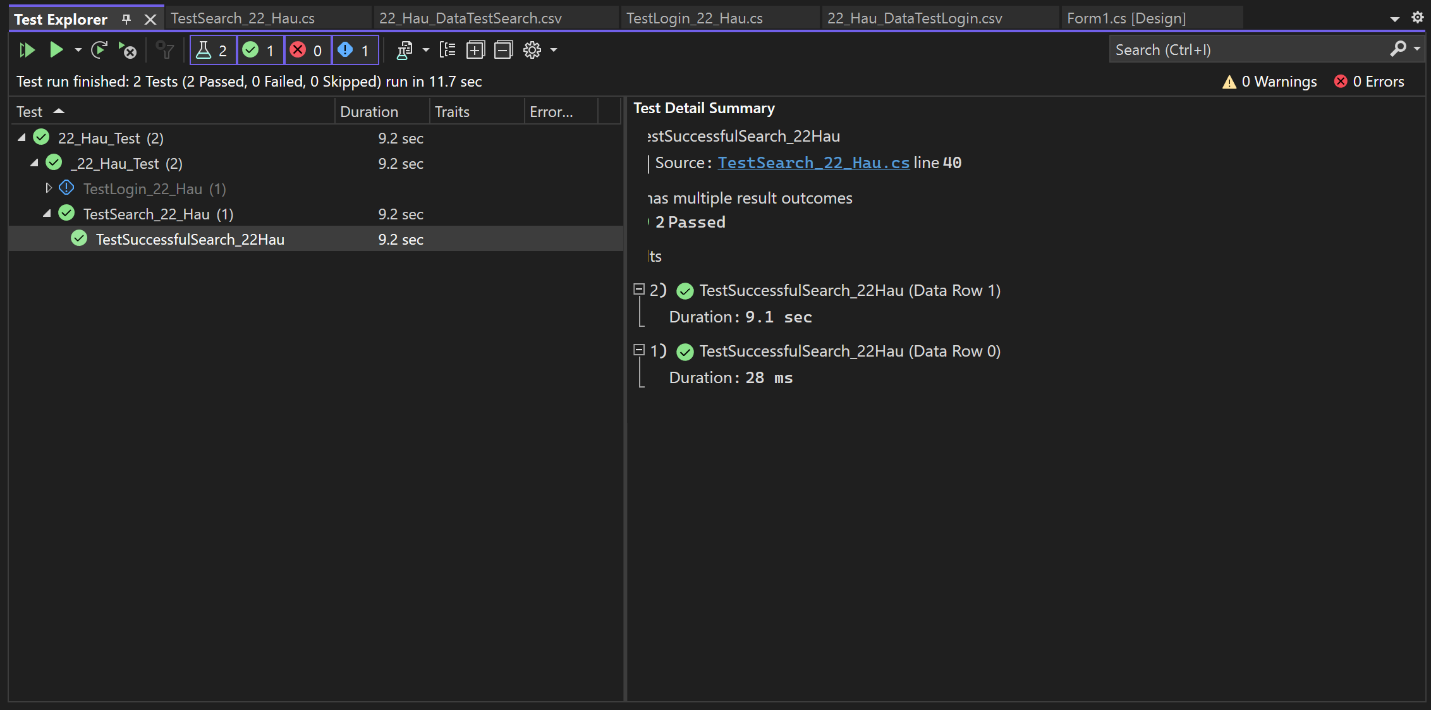
Assert.IsTrue(element2\_22Hau.Text.Contains(expectedResult\_22Hau));

driver\_22Hau.Quit();

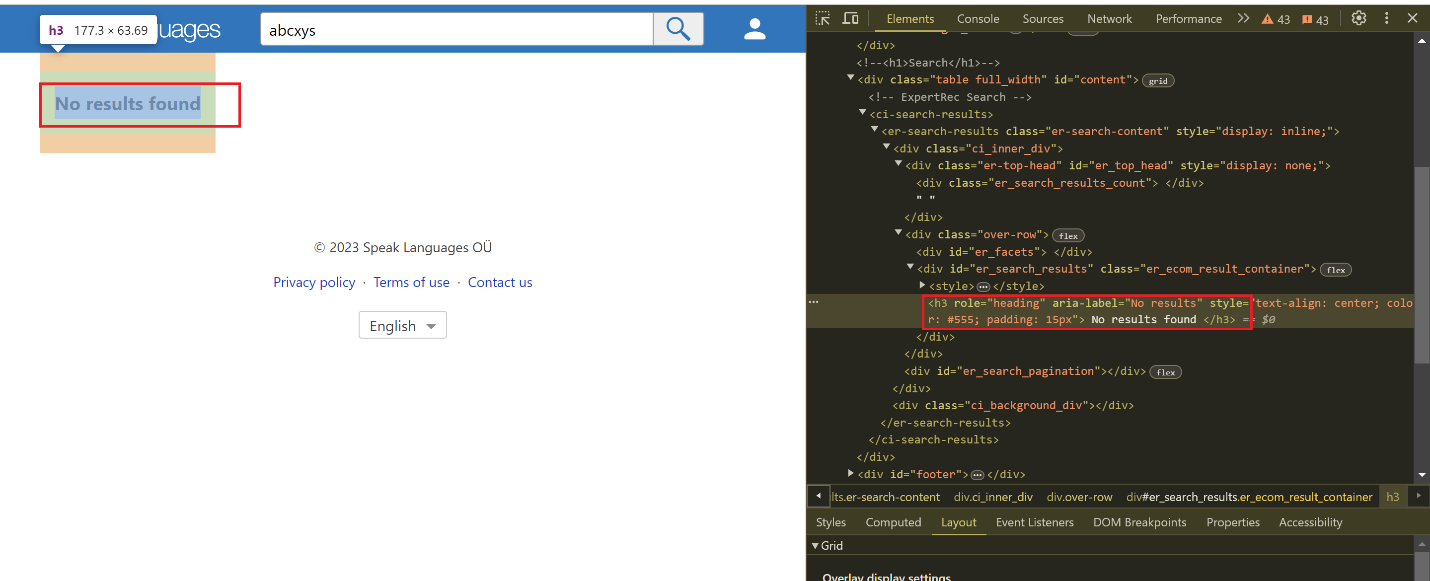
}

}

* Bước 3: Kết quả sau khi thực thi testcase



* + 1. **Test case 2: TestNoSearch\_22Hau()**
* Bước 1: Bắt Tagname (“h3”) cho dòng thông báo không tìm thấy



* Bước 2: Thiết lập code cho testcase tìm kiếm không thành công

public void TestNoSearch\_22Hau()

{

key\_22Hau = TestContext.DataRow[0].ToString();

expectedResult\_22Hau = TestContext.DataRow[1].ToString();

if (expectedResult\_22Hau == "No results found")

{

driver\_22Hau = new ChromeDriver();

driver\_22Hau.Navigate().GoToUrl("https://www.speaklanguages.com/");

System.Threading.Thread.Sleep(1000);

IWebElement element\_22Hau = driver\_22Hau.FindElement(By.Name("q"));

element\_22Hau.SendKeys(key\_22Hau);

IWebElement element1\_22Hau = driver\_22Hau.FindElement(By.CssSelector("input[type='submit']"));

element1\_22Hau.Click();

System.Threading.Thread.Sleep(2000);

IWebElement element2\_22Hau = driver\_22Hau.FindElement(By.TagName("h3"));

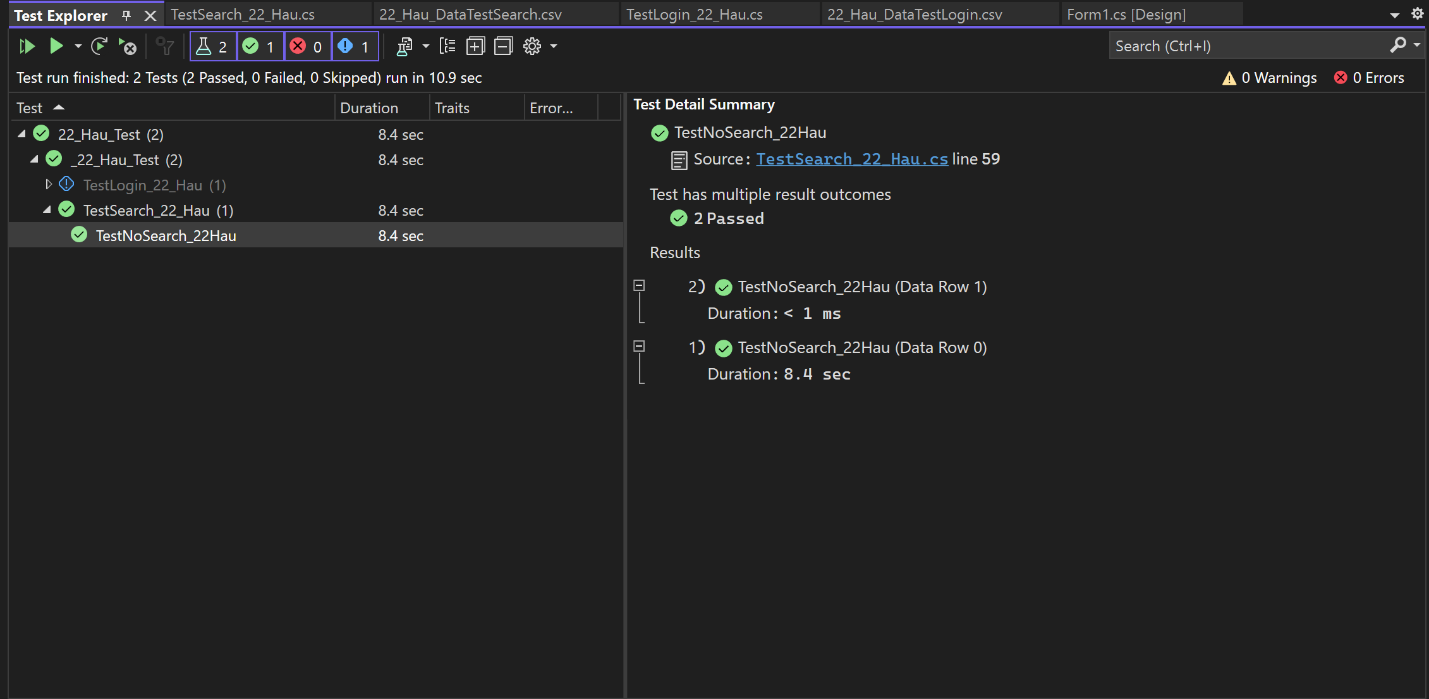
Assert.AreEqual(expectedResult\_22Hau, element2\_22Hau.Text);

driver\_22Hau.Quit();

}

}

* Bước 2: Kết quả sau khi thực thi code



* + 1. **Tổng hợp code chạy 3 testcase**
* Bước 1: Viết code chạy hết 4 testcase

public void TestSearch\_22Hau()

{

key\_22Hau = TestContext.DataRow[0].ToString();

expectedResult\_22Hau = TestContext.DataRow[1].ToString();

driver\_22Hau = new ChromeDriver();

driver\_22Hau.Navigate().GoToUrl("https://www.speaklanguages.com/");

System.Threading.Thread.Sleep(1000);

IWebElement element\_22Hau = driver\_22Hau.FindElement(By.Name("q"));

element\_22Hau.SendKeys(key\_22Hau);

IWebElement element1\_22Hau = driver\_22Hau.FindElement(By.CssSelector("input[type='submit']"));

element1\_22Hau.Click();

System.Threading.Thread.Sleep(2000);

if (expectedResult\_22Hau == "No results found")

{

IWebElement element2\_22Hau = driver\_22Hau.FindElement(By.TagName("h3"));

Assert.AreEqual(expectedResult\_22Hau, element2\_22Hau.Text);

}

else

{

IWebElement element3\_22Hau = driver\_22Hau.FindElement(By.XPath("//\*[@id=\"er\_top\_head\"]/div"));

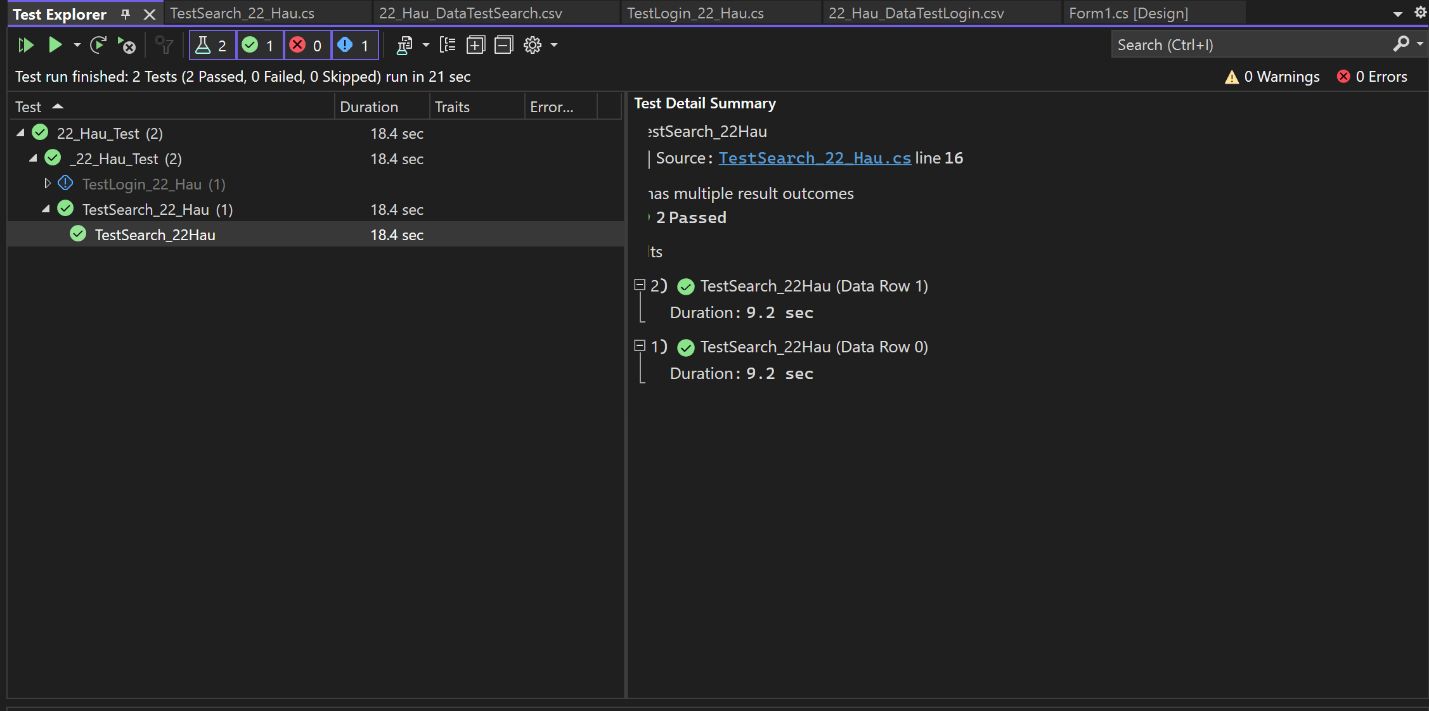
Assert.IsTrue(element3\_22Hau.Text.Contains(expectedResult\_22Hau));

}

driver\_22Hau.Quit();

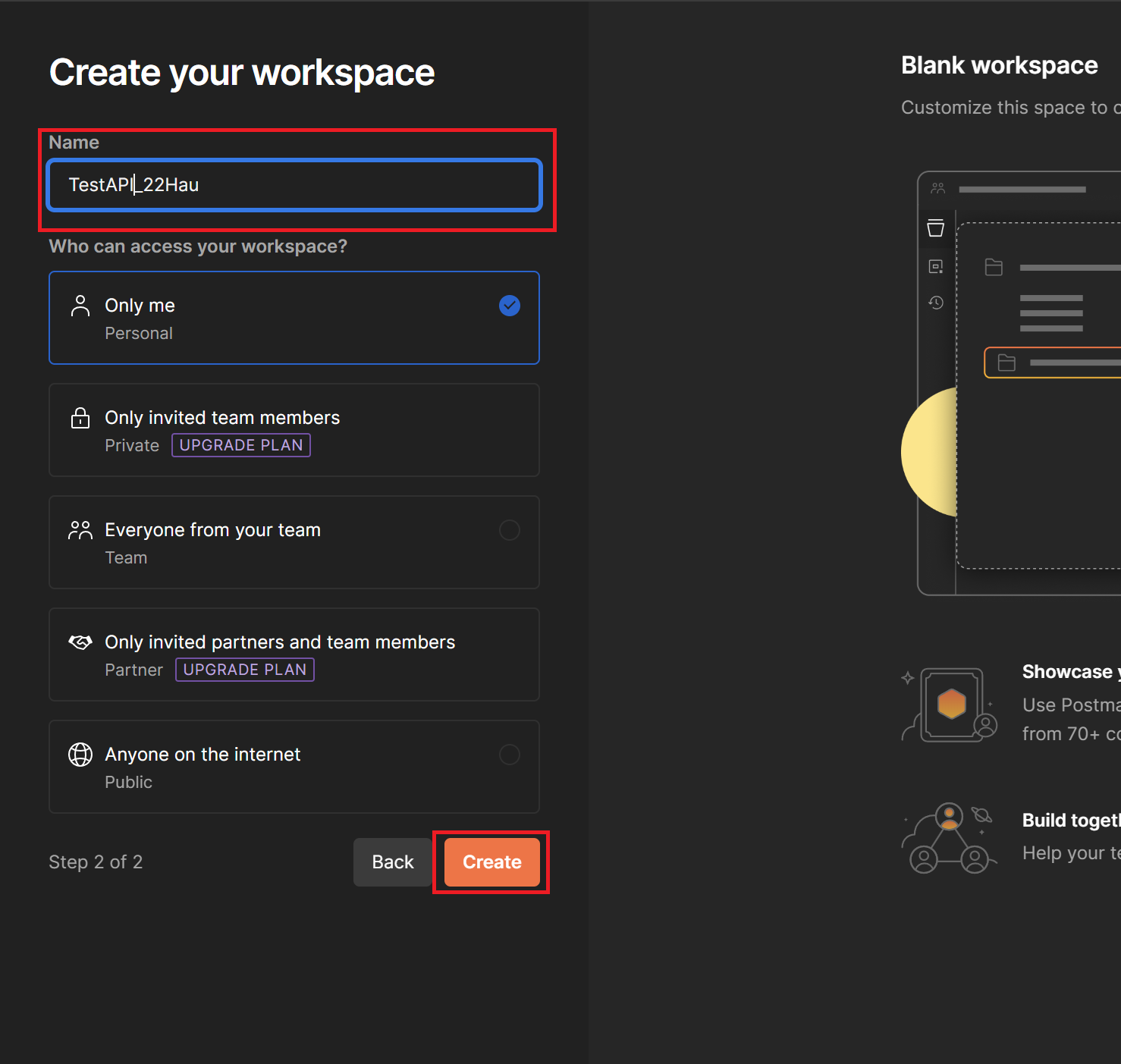
}

* Bước 2: Kết quả sau khi thực thi testcase

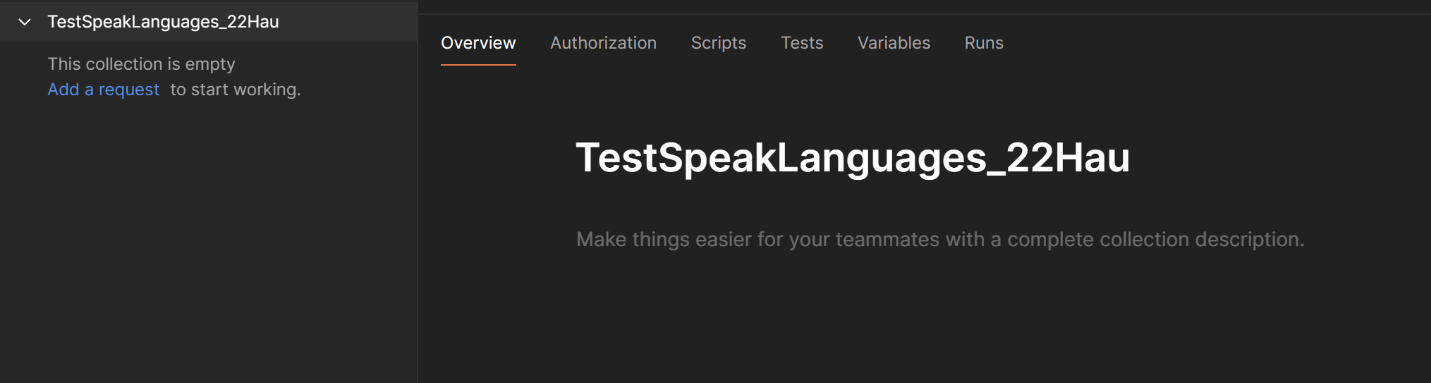


# **API TEST**

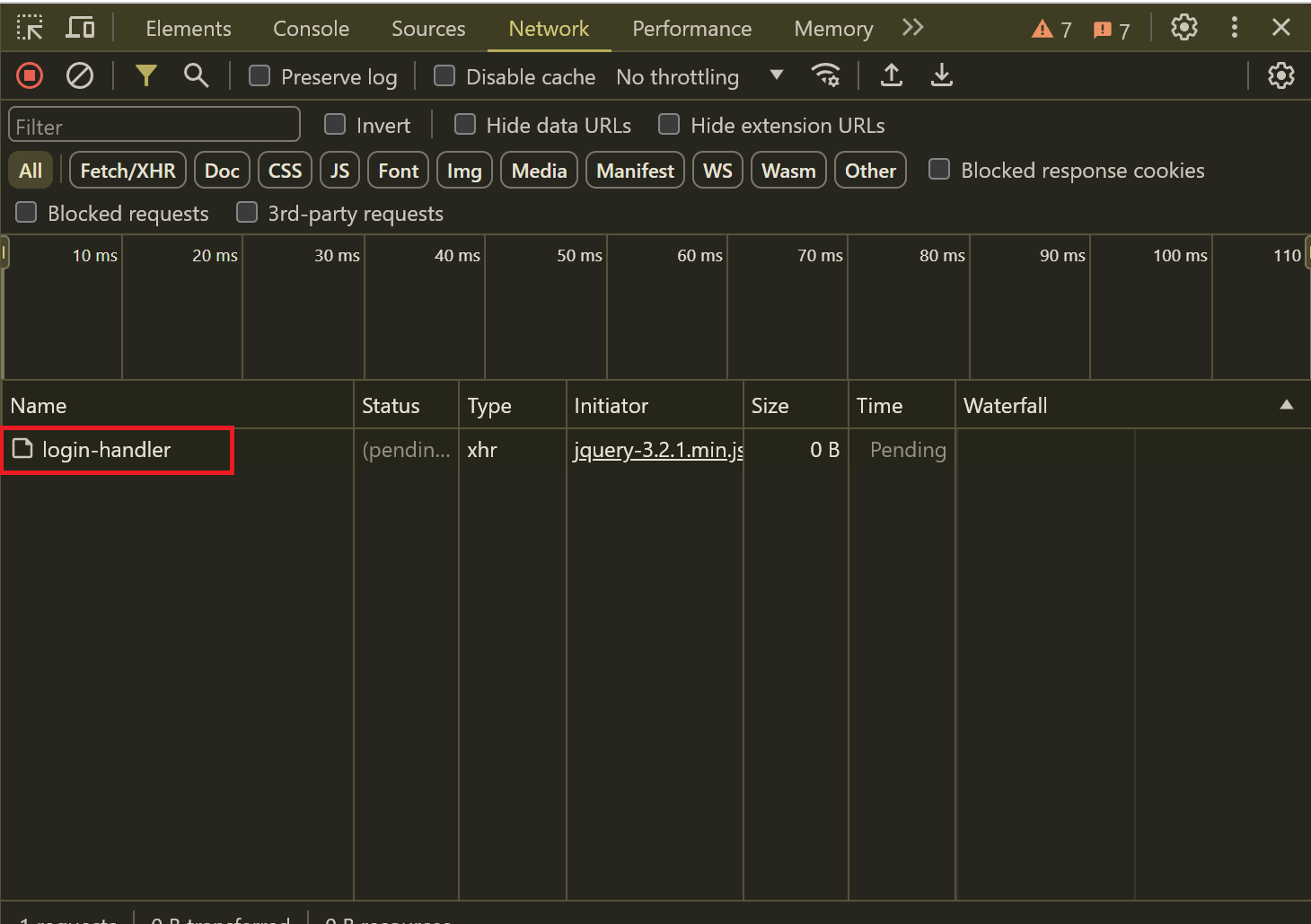
* 1. **Tạo Workspaces**
* Bước 1: Đăng nhập vào <https://www.postman.com/>
* Bước 2: Tạo Workspaces với tên: TestAPI\_22Hau => create

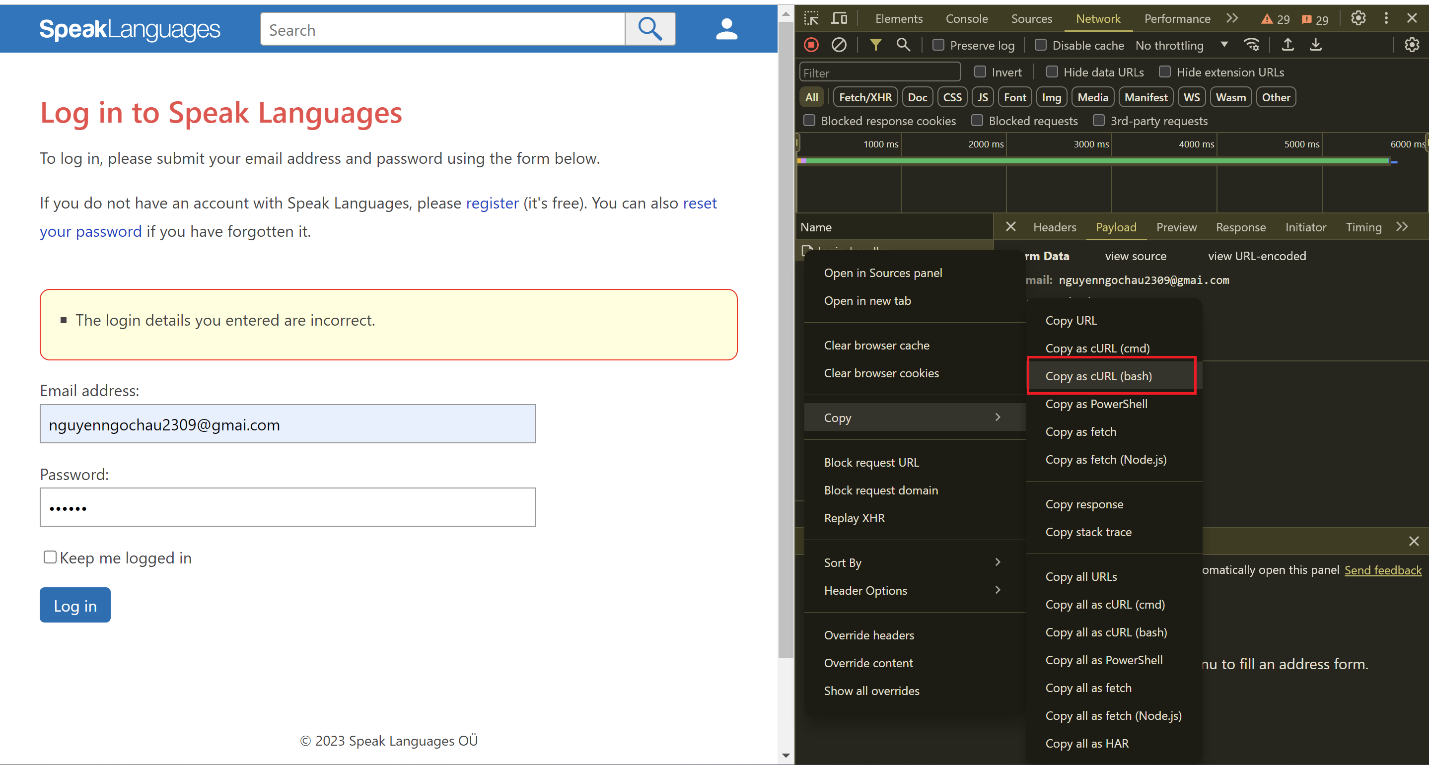


* 1. **Test API của request Post (Trang web Speak Language)**
     1. **Tạo request phương Post**
* Bước 1: Tạo collection cho trang web:

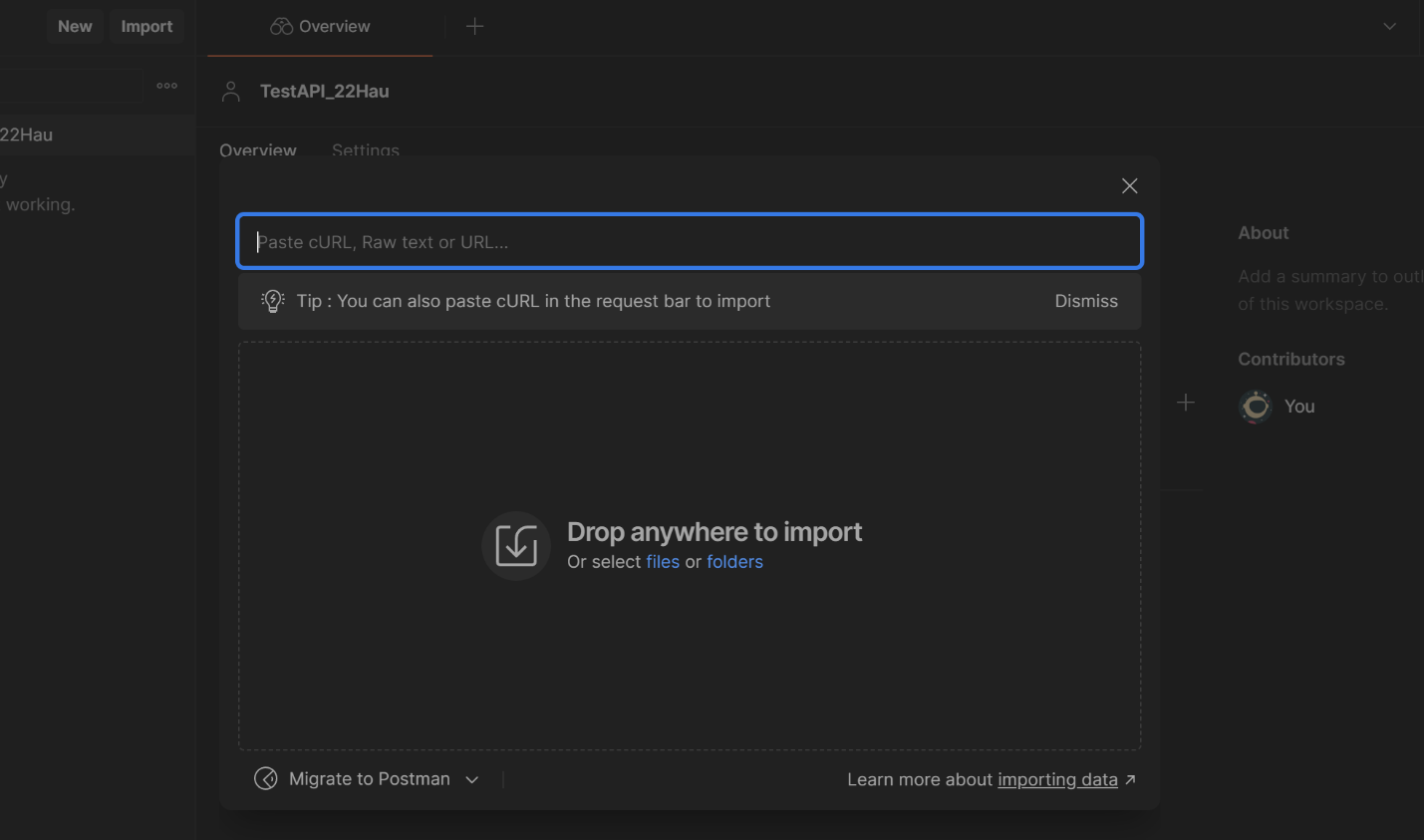


* Bước 2: Vào trang đăng nhập [**https://www.speaklanguages.com/login**](https://www.speaklanguages.com/login) => mở tab inspect vào network => chuột phải login-handler => chọn copy as cURL(bash)

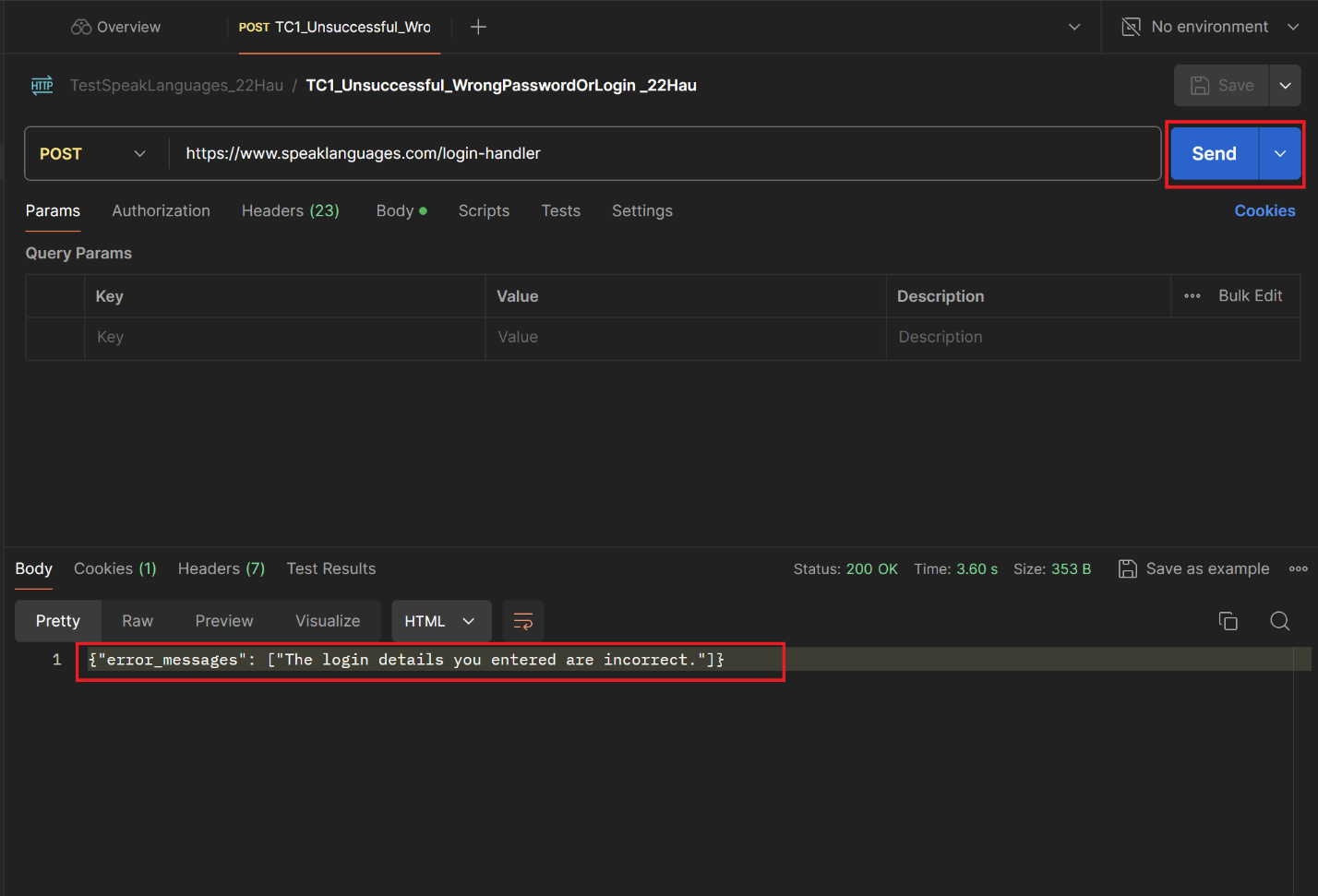




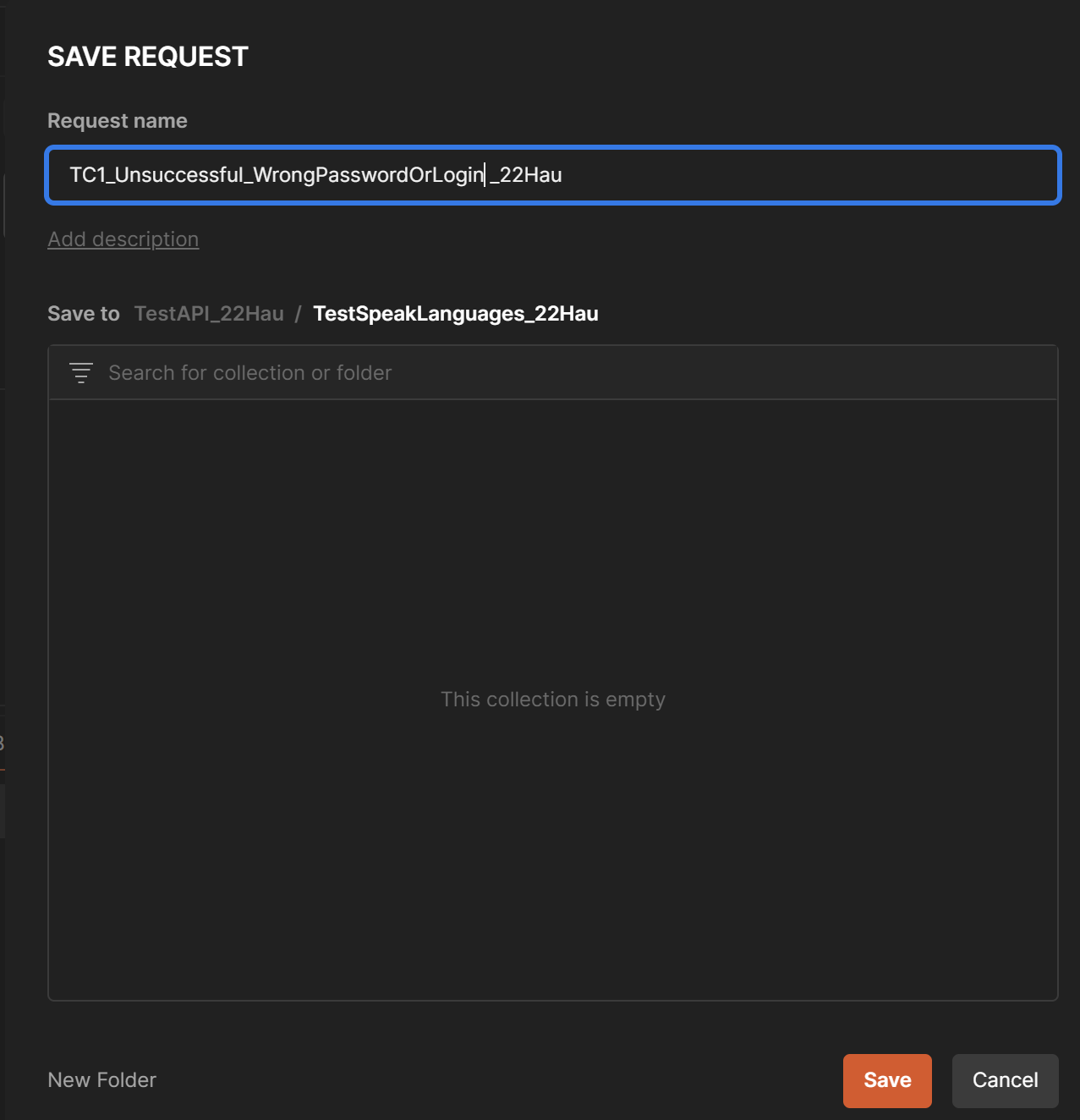
* Bước 3: Vào postman chọn import dán url



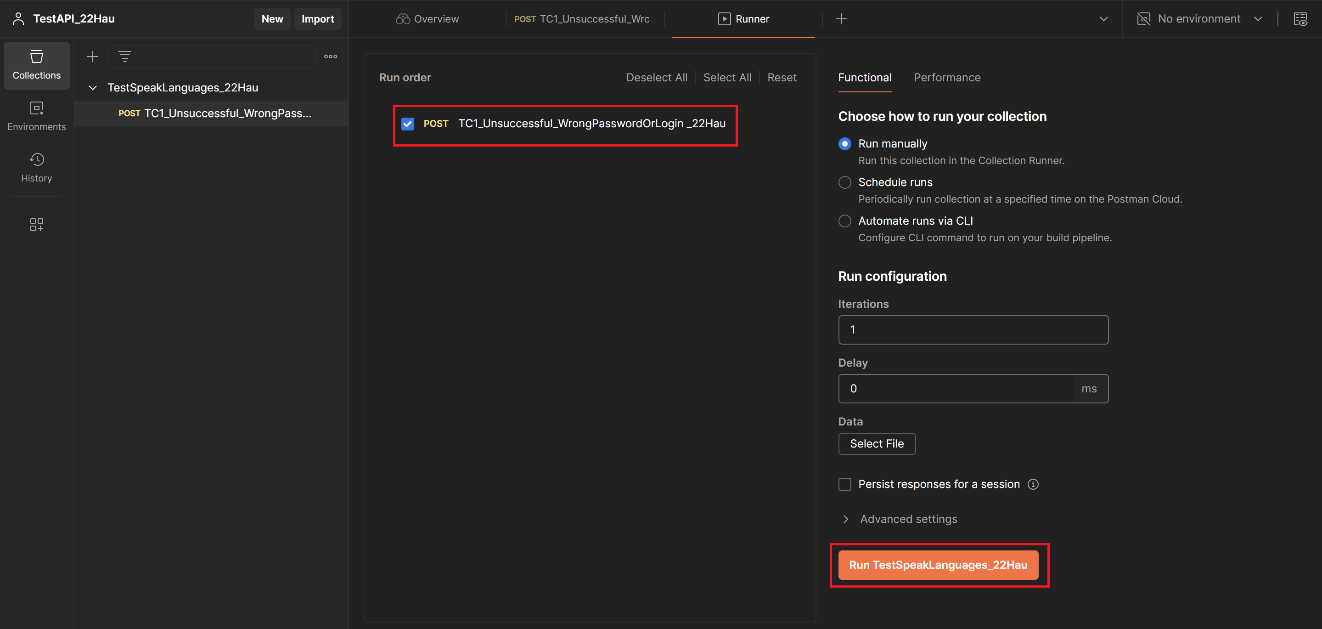
* Bước 4: Nhấn send và hiện kết quả

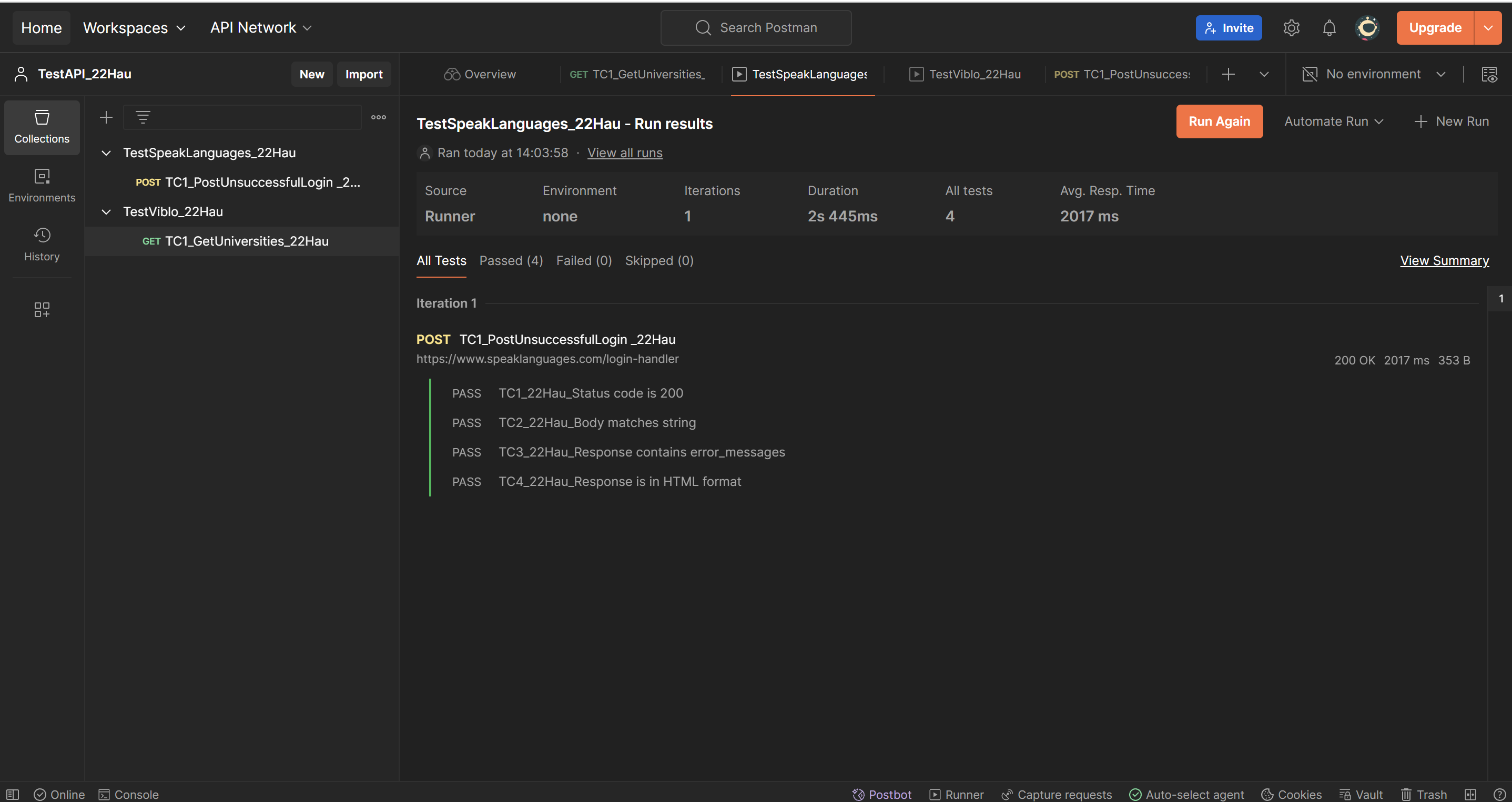


* Bước 5: Lưu request post trên collection TestSpeakLanguage\_22Hau

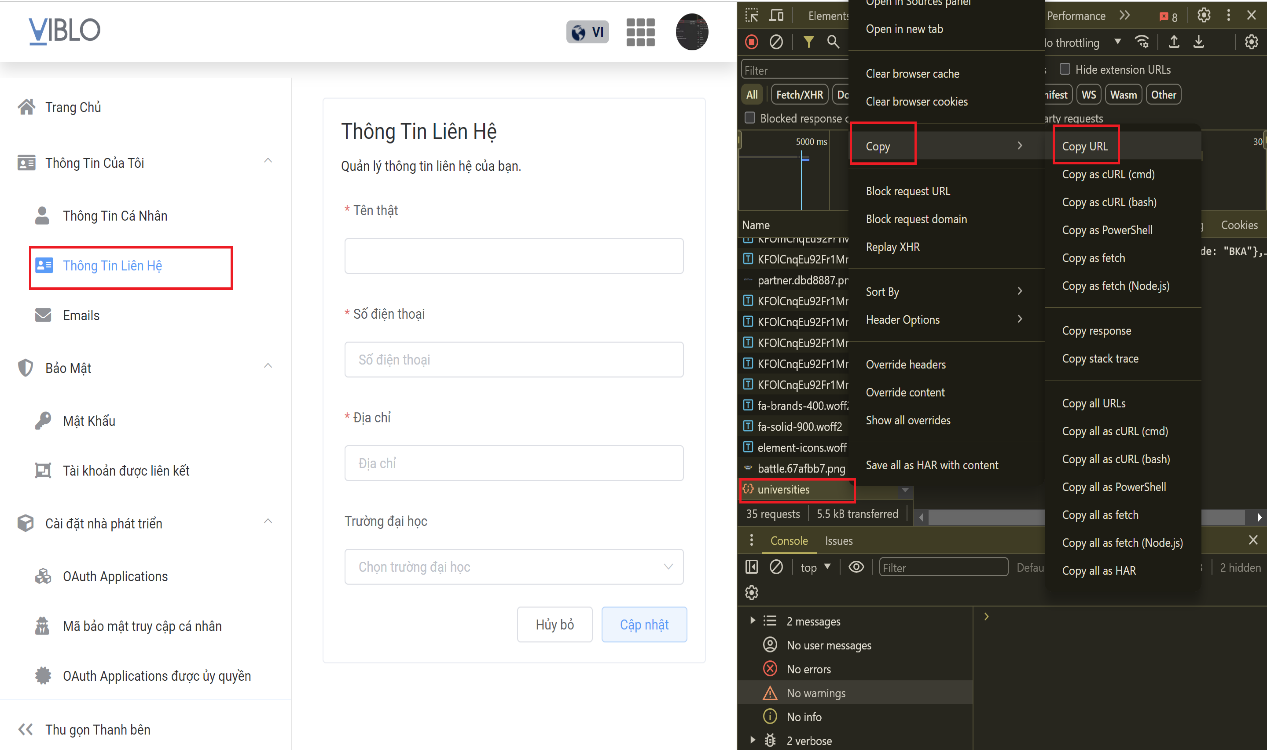


* + 1. **Viết Test trên phương thức Post**
* TC1­­\_Kiểm tra xem dữ liệu trả về có phải là 200 không:
* pm.test("TC1\_22Hau\_Status code is 200", *function* () {
* pm.response.to.have.status(200);
* });
* TC2\_Kiểm tra xem chữ trả về có khớp hay không
* pm.test("TC2\_22Hau\_Body matches string", *function* () {
* pm.expect(pm.response.text()).to.include("The login details you entered are incorrect.");
* });
* TC3\_Kiểm tra xem lúc trả về có dữ liệu ở đó không và kiểm tra xem có tồn tại trường error\_messages
* pm.test("TC3\_22Hau\_Response contains error\_messages", *function* () {
* pm.expect(pm.response.json()).to.be.an('object');
* pm.expect(pm.response.json()).to.have.property("error\_messages");
* });
* TC4\_Kiểm tra xem dữ liệu trả về có phải là text/html; charset=utf-8
* pm.test("TC4\_22Hau\_Response is in HTML format", *function* () {
* pm.response.to.have.header("Content-Type","text/html; charset=utf-8");
* });
* Chạy test trên postman: Chuột phải trên collection chọn run và chọn request muốn chạy => nhấn run

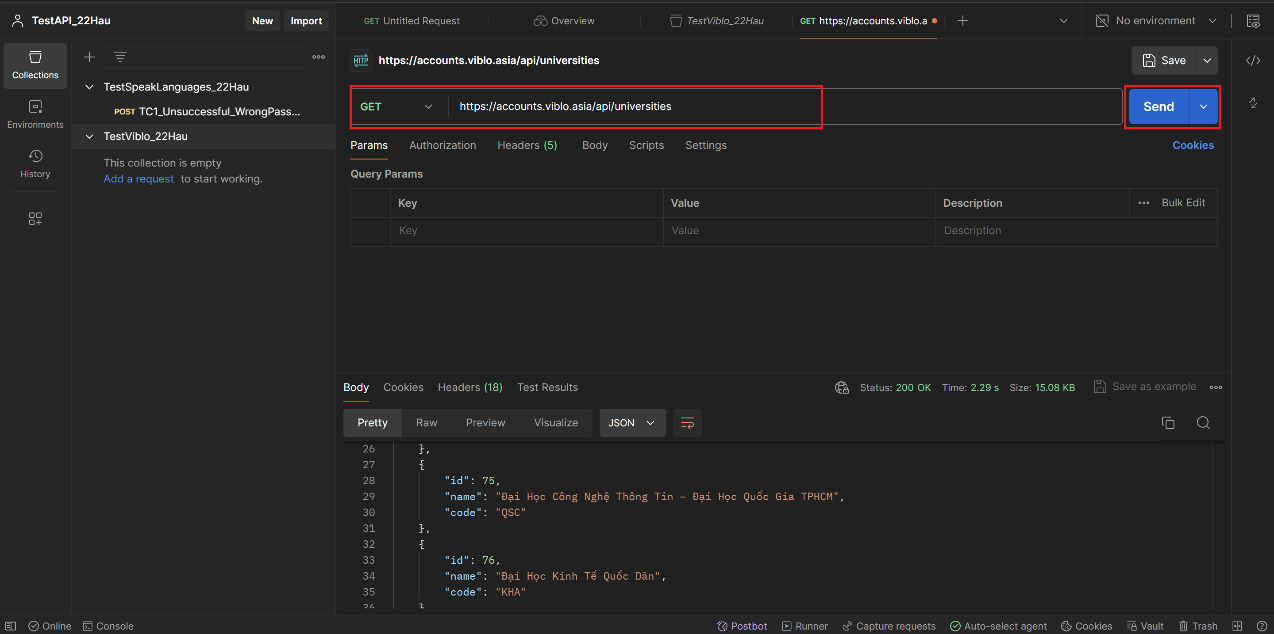




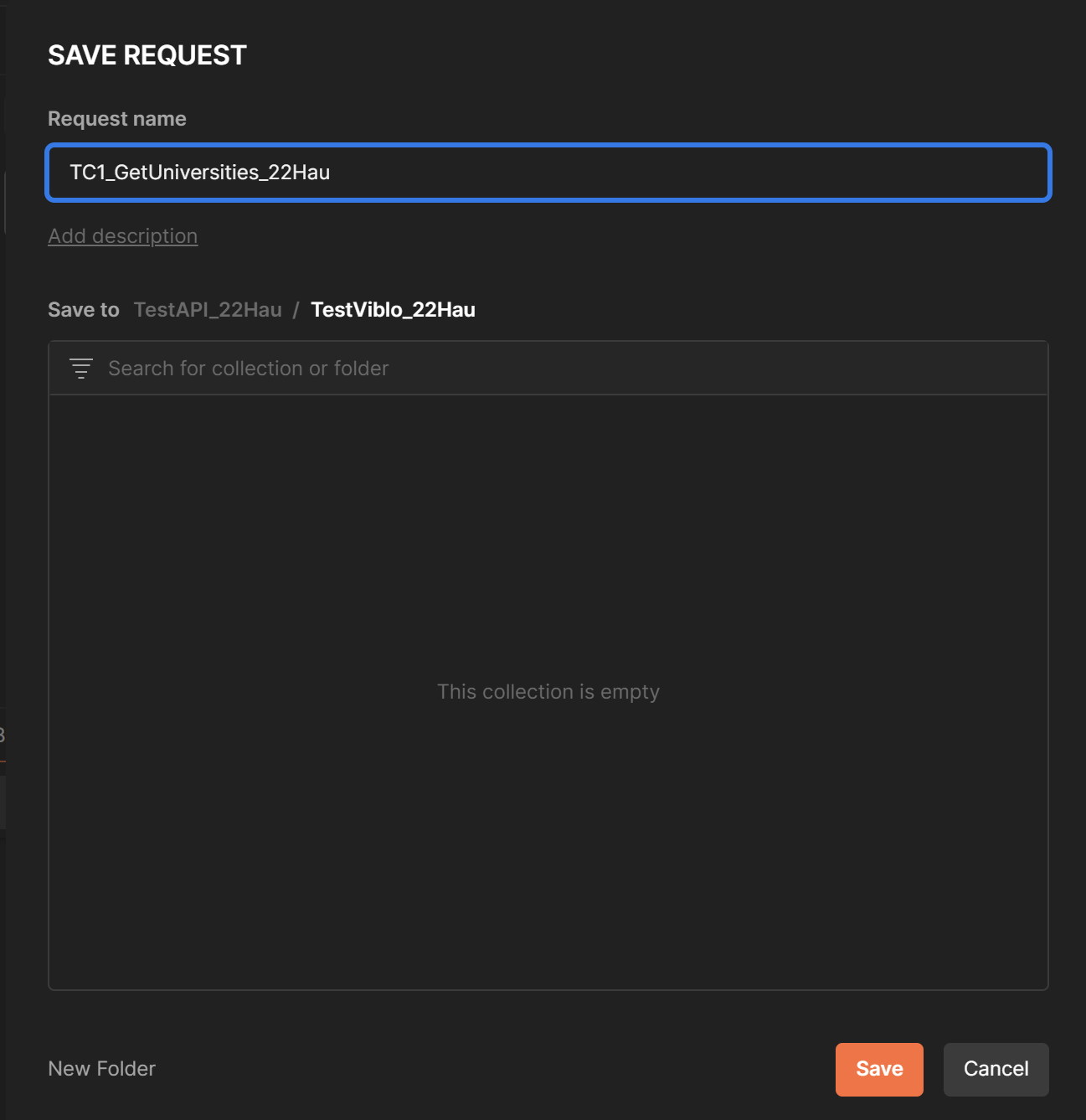
* 1. **Test API của request Get (Trang web Viblo)**
     1. **Tạo request cho phương thức get**
* Bước 1: Tạo collection cho trang web <https://accounts.viblo.asia/>
* Bước 2: Vào trang đăng nhập [**https://accounts.viblo.asia/login?continue=https%3A%2F%2Faccounts.viblo.asia**](https://accounts.viblo.asia/login?continue=https%3A%2F%2Faccounts.viblo.asia) => mở tab inspect vào network => chọn tab thông tin liên hệ => chọn universites copy as URL



* Bước 3: Vào postman chọn tạo request chọn phương thức get và past url vừa copy
* Bước 4: Nhấn send và nhận kết quả



* Bước 5: Lưu request get trong collection TestViblo\_22Hau



* + 1. **Viết test trên phương thức get**
* TC1\_Kiểm tra xem dữ liệu trả về có phải là 200 không
* pm.test("TC1\_22Hau\_Status code is 200", *function* () {
* pm.response.to.have.status(200);
* });
* TC2\_Kiểm tra xem dữ liệu trả về có phải là application/json không
* pm.test("TC2­\_22Hau\_Response is in json format", *function* () {
* pm.response.to.have.header("Content-Type", "application/json");
* });
* TC3\_Kiểm tra xem thời gian phản hồi có ít hơn 200ms không
* pm.test("TC3\_22Hau\_Response time is less than 200ms", *function* () {
* pm.expect(pm.response.responseTime).to.be.below(200);
* });
* Chạy test trên postman: Chuột phải trên collection chọn run và chọn request muốn chạy => nhấn run

