Mục lục

[Unit test 2](#_Toc26560)

[I. Kiểm tra tam giác 2](#_Toc14521)

[1. Tạo form để kiểm tra tam giác 2](#_Toc3215)

[2. Khai báo ba biến kiểu số nguyên Danh\_18\_a, Danh\_18\_b, và Danh\_18\_c, để lưu giá trị của ba ô nhập liệu txtA, txtB, và txtC tương ứng, chuyển đổi từ chuỗi sang số nguyên, một đối tượng mới của lớp CheckTamGiac được tạo ra. Cuối cùng, giá trị của chuỗi Danh\_18\_kq được gán vào thuộc tính Text của ô nhập liệu txtKQ 2](#_Toc2176)

[3. Tạo lớp CheckTamGiac. Nó nhận ba tham số là x, y, và z để khởi tạo các biến thành viên Danh\_18\_a, Danh\_18\_b, và Danh\_18\_c của đối tượng. 3](#_Toc2886)

[4. Phương thức để kiểm tra loại tam giác dựa trên ba cạnh. Trong phần kiểm tra điều kiện, trước hết, nó kiểm tra xem các cạnh có âm không. Nếu không, nó tiếp tục kiểm tra các trường hợp cụ thể. Kết quả của phương thức được gán vào biến Danh\_18\_loaiTG và trả về 3](#_Toc18522)

[5. Cuối cùng tạo file csv và add references cái projects mình test vào. Thiết lập dữ liệu dùng chung tester, liên kết testdata với project 4](#_Toc1370)

[WebDriver 5](#_Toc25682)

[I. Chức năng “Đăng nhập”. 5](#_Toc28014)

[1) Mô tả chức năng 5](#_Toc26686)

[2) Kiểm thử hộp đen 5](#_Toc23764)

[II. Chức năng “Xóa”. 13](#_Toc6575)

[1) Mô tả chức năng 13](#_Toc30266)

[2) Kiểm thử hộp đen 13](#_Toc15990)

[Test API 22](#_Toc18078)

[I. Tạo JsonServer 22](#_Toc23614)

[1. Cài đặt Node, tạo JsonServer trên thư mục và chạy lệnh npm init 22](#_Toc24318)

[2. Tạo một file json trong thư mục 23](#_Toc25103)

[3. Mở file package.json trong thư mục 24](#_Toc29787)

[4. Cài đặt JsonServer, gõ lệnh npm install json-server và bắt đầu chạy gõ npm start 24](#_Toc29356)

[II. Thiệt lập Postman 24](#_Toc267)

[1. Tạo workspace 24](#_Toc32558)

[2. Tạo collection 25](#_Toc31088)

[3. Tạo biến global 25](#_Toc7000)

[4. Test API 25](#_Toc30603)

[5. Viết test case 26](#_Toc352)

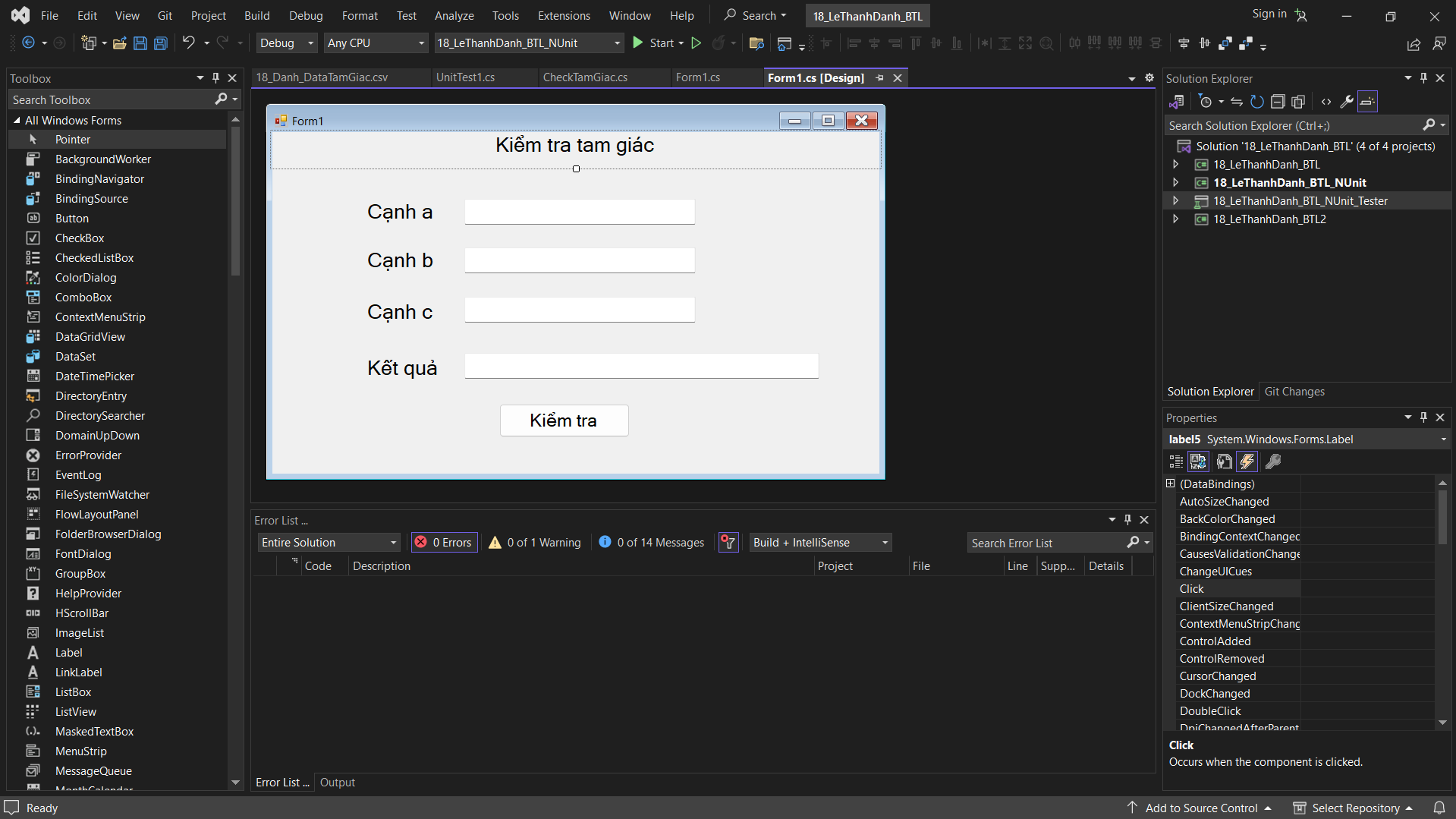
Tên: Lê Thanh Danh Kiểm thử phần mềm

MSSV: 2151050047

STT: 18

Unit test

1. Kiểm tra tam giác
2. Tạo form để kiểm tra tam giác



1. Khai báo ba biến kiểu số nguyên Danh\_18\_a, Danh\_18\_b, và Danh\_18\_c, để lưu giá trị của ba ô nhập liệu txtA, txtB, và txtC tương ứng, chuyển đổi từ chuỗi sang số nguyên, một đối tượng mới của lớp CheckTamGiac được tạo ra. Cuối cùng, giá trị của chuỗi Danh\_18\_kq được gán vào thuộc tính Text của ô nhập liệu txtKQ

private void btnKQ\_Click(object sender, EventArgs e)

{

int Danh\_18\_a, Danh\_18\_b, Danh\_18\_c;

//18\_Le Thanh Danh

Danh\_18\_a = int.Parse(txtA.Text);

Danh\_18\_b = int.Parse(txtB.Text);

Danh\_18\_c = int.Parse(txtC.Text);

CheckTamGiac Danh\_18 = new CheckTamGiac(Danh\_18\_a,Danh\_18\_b,Danh\_18\_c);

string Danh\_18\_kq = Danh\_18.funcTamGiac();

txtKQ.Text = Danh\_18\_kq.ToString();

}

1. Tạo lớp CheckTamGiac. Nó nhận ba tham số là x, y, và z để khởi tạo các biến thành viên Danh\_18\_a, Danh\_18\_b, và Danh\_18\_c của đối tượng.

public CheckTamGiac(int x,int y,int z) {

this.Danh\_18\_a = x; this.Danh\_18\_b = y; this.Danh\_18\_c = z;

}

1. Phương thức để kiểm tra loại tam giác dựa trên ba cạnh. Trong phần kiểm tra điều kiện, trước hết, nó kiểm tra xem các cạnh có âm không. Nếu không, nó tiếp tục kiểm tra các trường hợp cụ thể. Kết quả của phương thức được gán vào biến Danh\_18\_loaiTG và trả về

public string funcTamGiac()

{

string Danh\_18\_loaiTG;

if(Danh\_18\_a < 0 || Danh\_18\_b < 0 || Danh\_18\_c < 0) {

return Danh\_18\_loaiTG = "Tam giác không được cạnh < 0";

}

else

{

if (Danh\_18\_a == Danh\_18\_b && Danh\_18\_b == Danh\_18\_c)

return Danh\_18\_loaiTG = "Tam giac deu";

else if (Danh\_18\_a == Danh\_18\_b || Danh\_18\_a == Danh\_18\_c || Danh\_18\_b == Danh\_18\_c)

return Danh\_18\_loaiTG = "Tam giac can";

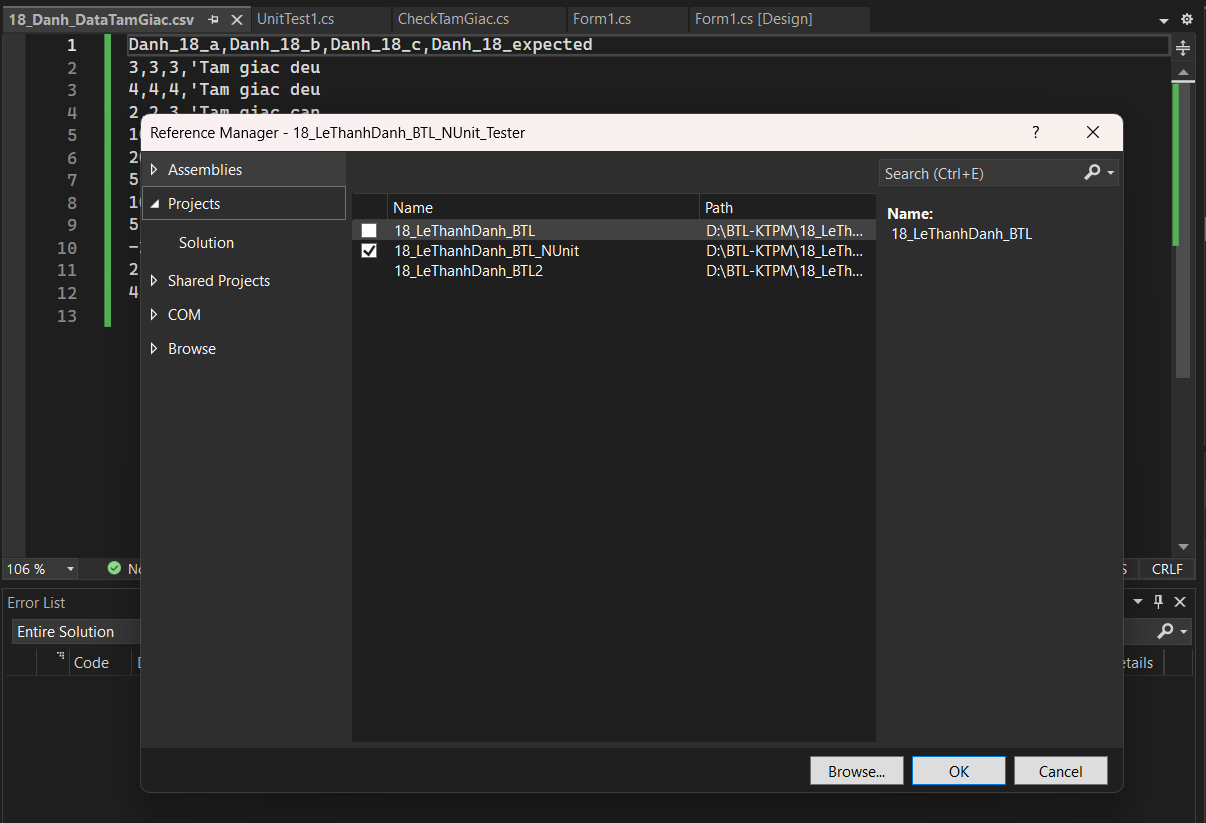
else

return Danh\_18\_loaiTG = "Tam giac thuong";

}

}

1. Cuối cùng tạo file csv và add references cái projects mình test vào. Thiết lập dữ liệu dùng chung tester, liên kết testdata với project



[TestMethod]

[DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV",

@".\18\_Danh\_Data\18\_Danh\_DataTamGiac.csv", "18\_Danh\_DataTamGiac#csv", DataAccessMethod.Sequential)]

public void Danh\_18\_TestWithDataSource()

{

int Danh\_18\_a, Danh\_18\_b, Danh\_18\_c;

string Danh\_18\_expected, Danh\_18\_actual;

Danh\_18\_a = int.Parse(TestContext.DataRow[0].ToString());

Danh\_18\_b = int.Parse(TestContext.DataRow[1].ToString());

Danh\_18\_c = int.Parse(TestContext.DataRow[2].ToString());

Danh\_18\_expected = TestContext.DataRow[3].ToString();

Danh\_18\_expected = Danh\_18\_expected.Remove(0, 1);

Danh\_18 = new CheckTamGiac(Danh\_18\_a, Danh\_18\_b, Danh\_18\_c);

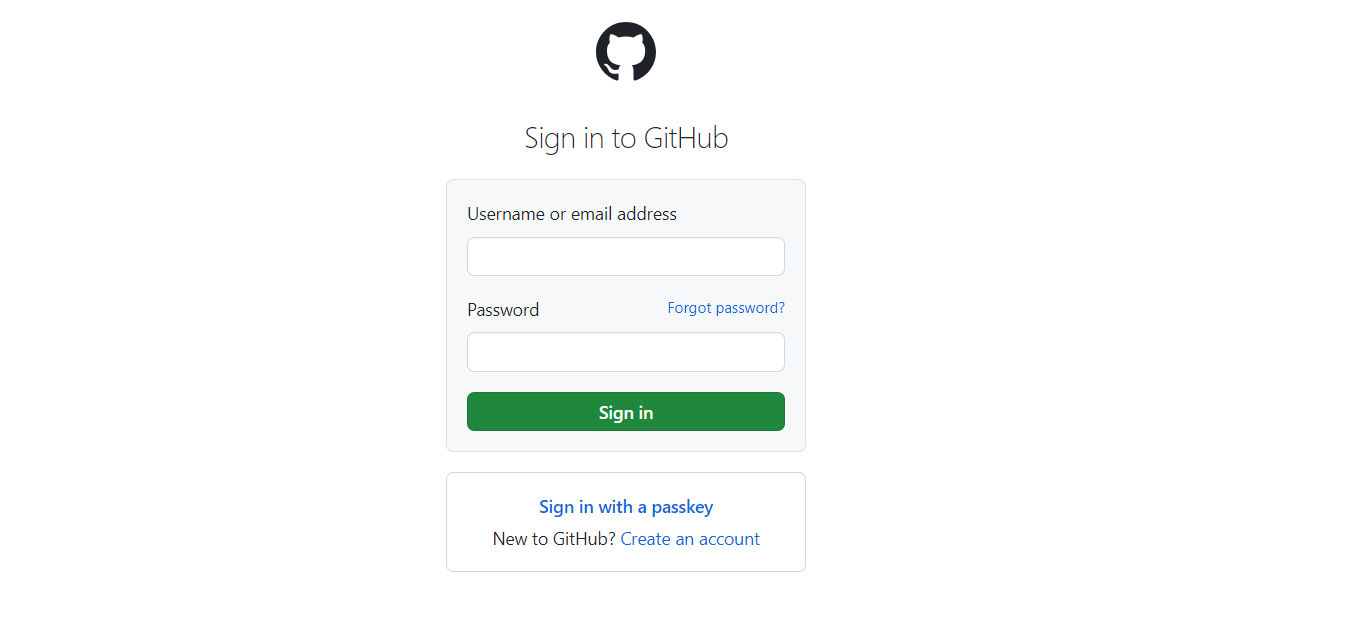
Danh\_18\_actual = Danh\_18.funcTamGiac();

Assert.AreEqual(Danh\_18\_expected, Danh\_18\_actual);

}

WebDriver

1. Chức năng “Đăng nhập”.
   1. Mô tả chức năng



* Chức năng “Đăng nhập”: cho phép người dùng xác thực danh tính của họ để truy cập vào hệ thống hoặc ứng dụng.
* Điều kiện để đăng nhập:
* Người dùng cần có tài khoản đã đăng ký trước đó
* Thông tin đăng nhập của người dùng phải đúng và phù hợp với thông tin đã đăng ký.
* Kiểm tra lỗi: nhập thiếu thông tin, nhập sai định dạng
  1. Kiểm thử hộp đen
     1. Bảng quyết định

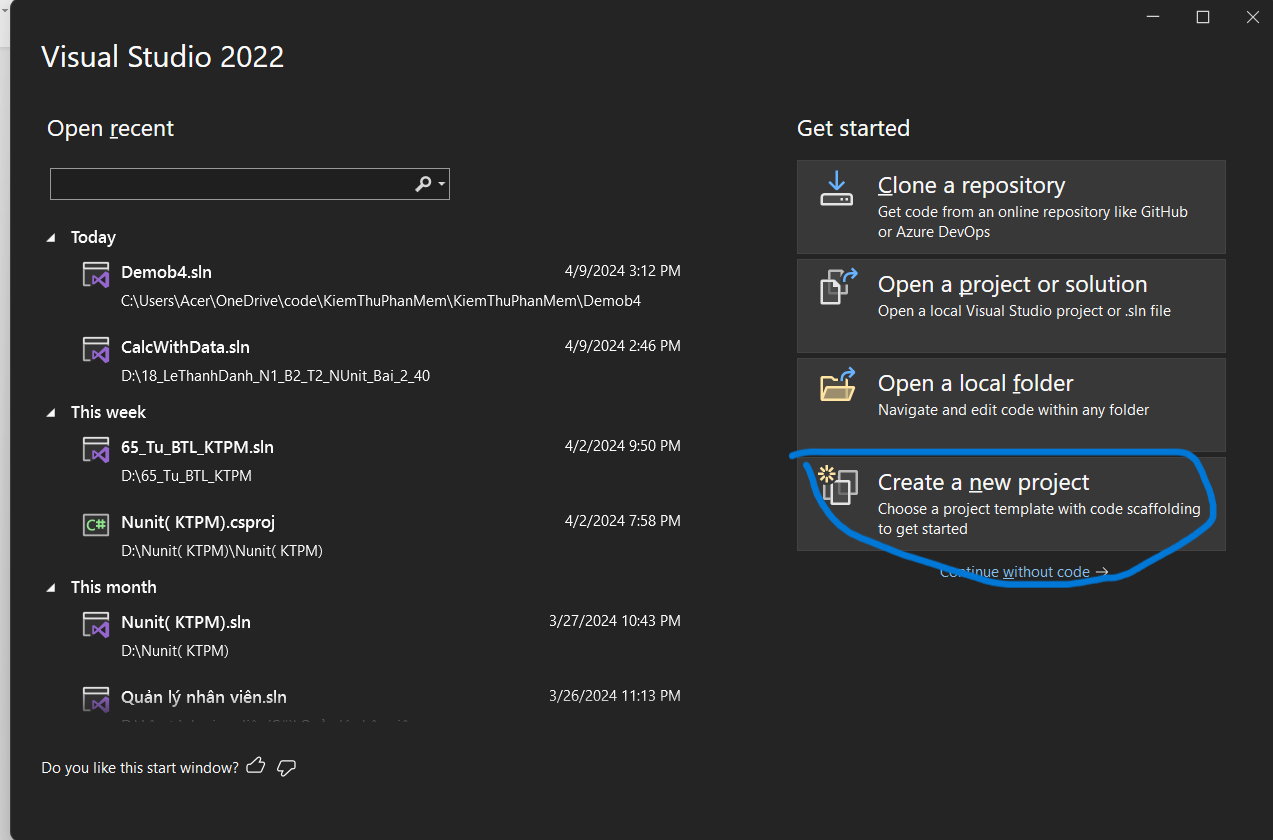
Liệt kê:

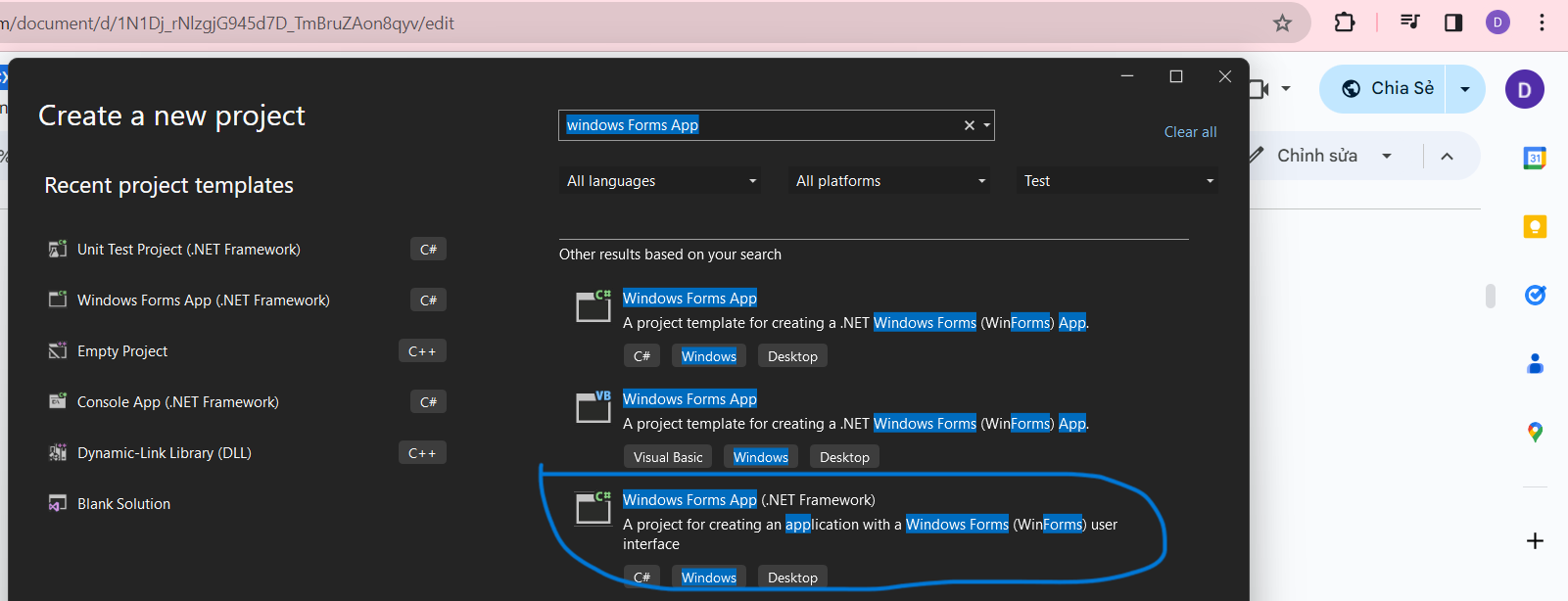
|  |  |
| --- | --- |
| Các điều kiện | * Tên người dùng và mật khẩu hợp lệ * Tên người dùng không hợp lệ * Mật khẩu không hợp lệ * Tên người dùng và mật khẩu không khớp |
| Các hành động | * Cho phép đăng nhập * Hiện thị thông báo lỗi |

Bảng:

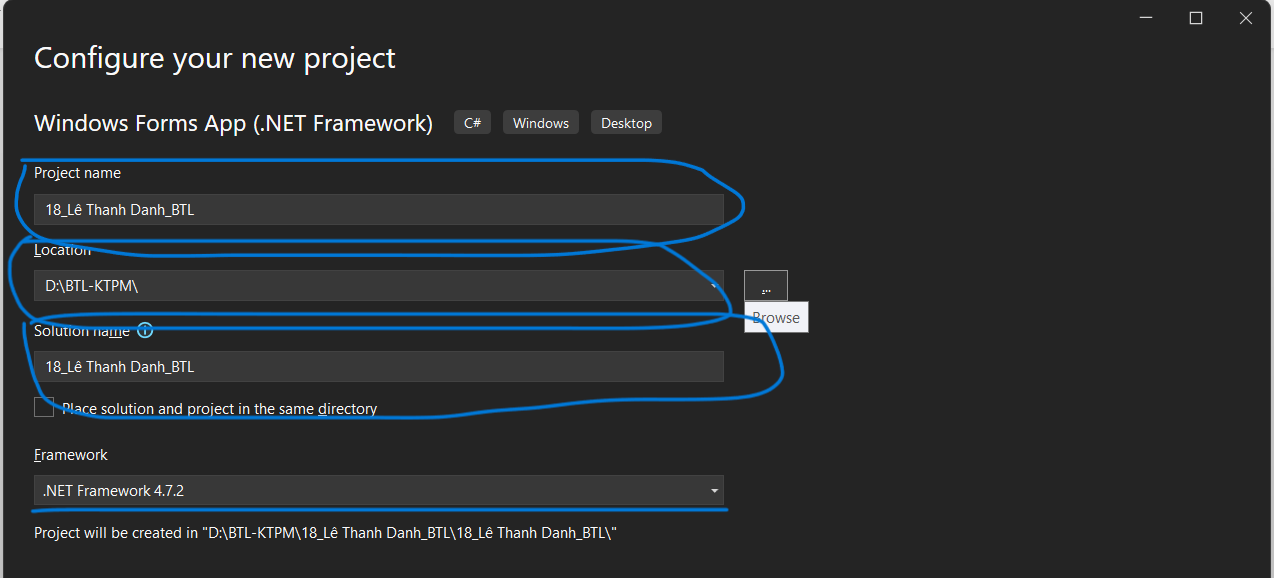
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case | Tên người dùng và mật khẩu hợp lệ | Tên người dùng không hợp lệ | Mật khẩu không hợp lệ | Tên người dùng và mật khẩu không khớp | Các hành động | Kết quả |
| 1 | True | False | False | False | Cho phép đăng nhập | Pass |
| 2 | False | True | False | False | Hiện thị thông báo lỗi | Fail |
| 3 | False | False | True | False | Hiện thị thông báo lỗi | Fail |
| 4 | False | False | False | True | Hiện thị thông báo lỗi | Fail |

* + 1. Giải thích code
       1. Chuẩn bị môi trường trong Visual Studio
          1. Tạo 1 project Window Form (.Net Framework) mới:

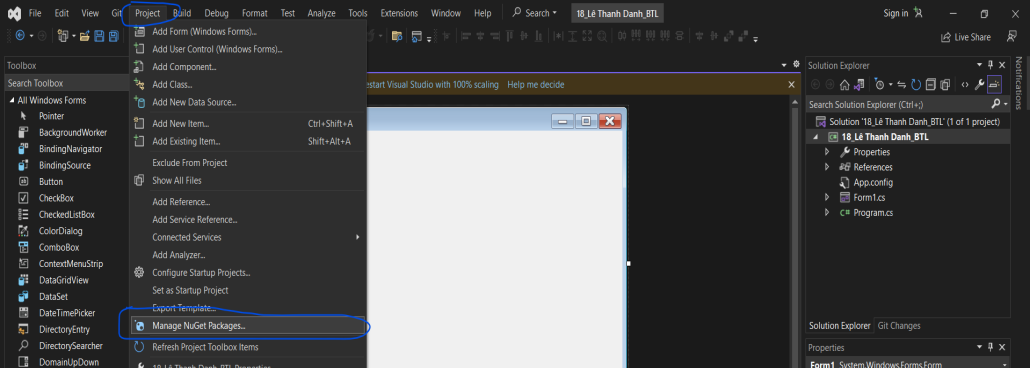
Chọn Create New Project

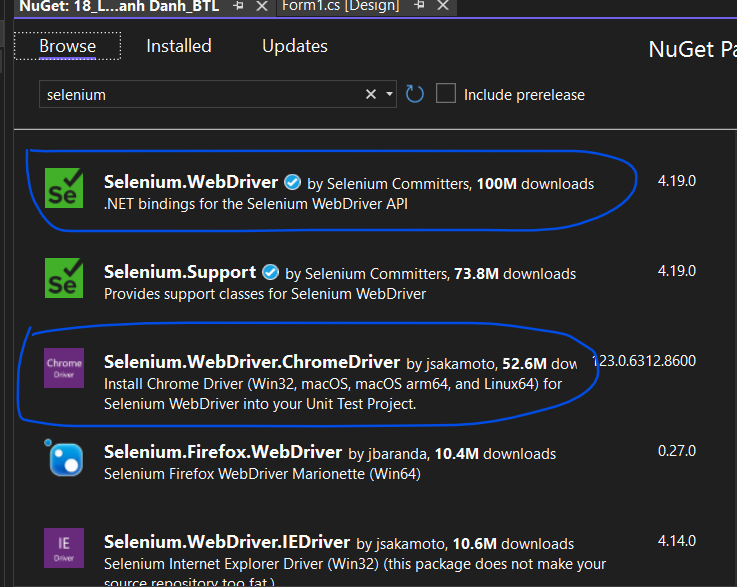
Search Window Form App (.Net Framework), chọn Window Form App (.Net Framework)

Nhập tên Project, location, framework của project, nhấn create:



* + - * 1. Cài Selenium cho 18\_Lê Thanh Danh\_BTL

Sau khi mở solution 18\_Lê Thanh Danh\_BTL, chọn Project, Manage NuGet Packages

Chọn Browse, search Selenium, cài Selenium.WebDriver, Selenium.WebDriver.ChromeDriver

* + - * 1. Khởi tạo các thuộc tính của class Form1

Khởi tạo driver, đóng cmd, gán url và điều hướng navigate để mở trang url đã nhập

//18\_Lê Thanh Danh

//Đóng cmd

ChromeDriverService TDanh\_18 = ChromeDriverService.CreateDefaultService();

TDanh\_18.HideCommandPromptWindow = true;

//18\_Lê Thanh Danh

IWebDriver Danh\_18 = new ChromeDriver(TDanh\_18);

Danh\_18.Url = "https://github.com/ThanhhDanh";//Gán URL

Danh\_18.Navigate(); //Điều hướng để mở trang url đã gán bằng navigate

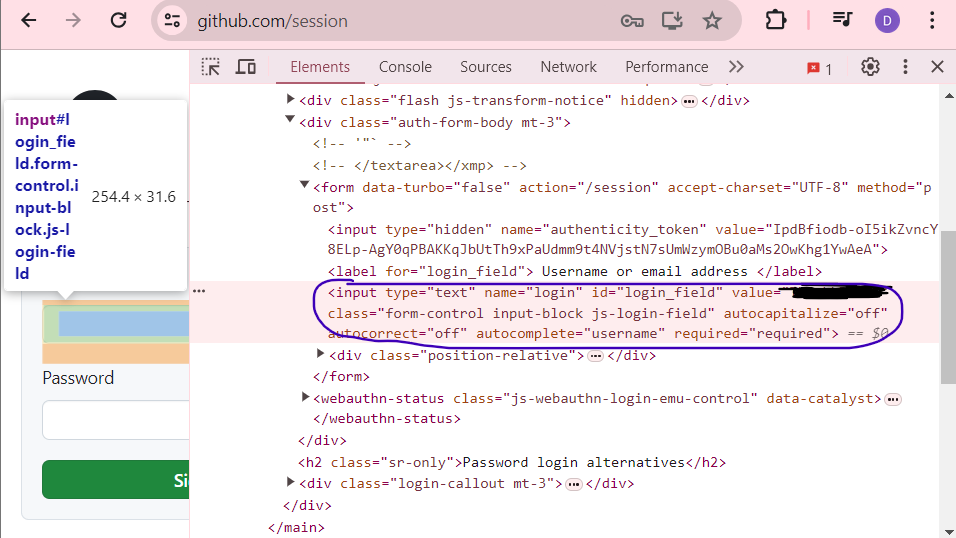
* + - 1. Các hàm hỗ trợ
* Tất cả các phương thức dưới đây đều làm về chức năng “Đăng nhập”
* Phương thức điều hướng tới trang github.com/login - Navigate

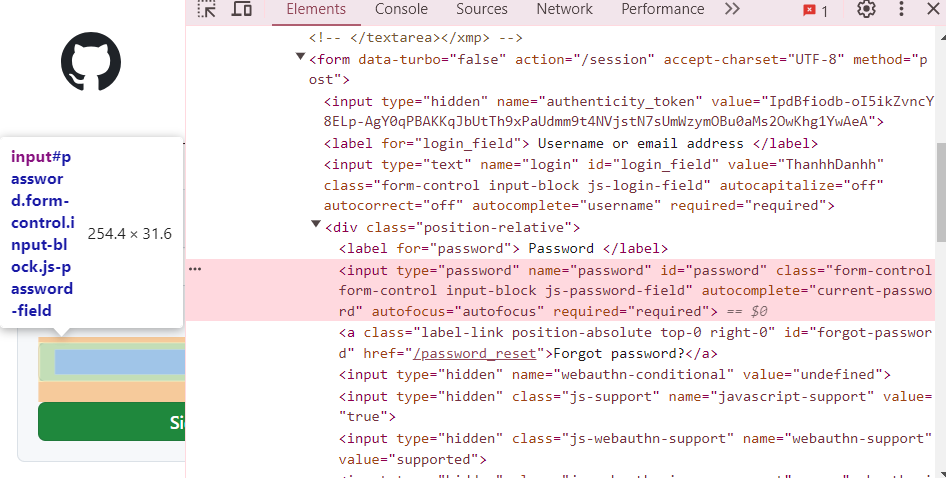
IWebDriver Danh\_18 = new ChromeDriver(TDanh\_18);

Danh\_18.Url = "https://github.com/ThanhhDanh";//Gán URL

Danh\_18.Navigate(); //Điều hướng để mở trang url đã gán bằng navigate

* Phương thức “Đăng nhập”
* Tìm class của input đăng nhập và mật khẩu





Test Case 1

* Sử dụng Name của 2 input, truyền vào phương thức FindElement của attribute driver để gọi đến 2 input và truyền giá trị qua SendKeys. Cuối cùng nhấn vào nút Sign In thông qua gọi Name và sửa dụng Click

//18\_Lê Thanh Danh

//Input của email để truyền giá trị từ SendKeys vào

IWebElement Danh\_18\_email = Danh\_18.FindElement(By.Name("login"));

Danh\_18\_email.SendKeys("ThanhhDanh");

Thread.Sleep(1000);

//18\_Lê Thanh Danh

//Input của password để truyền giá trị từ SendKeys vào

IWebElement Danh\_18\_password = Danh\_18.FindElement(By.Name("password"));

Danh\_18\_password.SendKeys("lethanhdanh2003");

Thread.Sleep(2000);

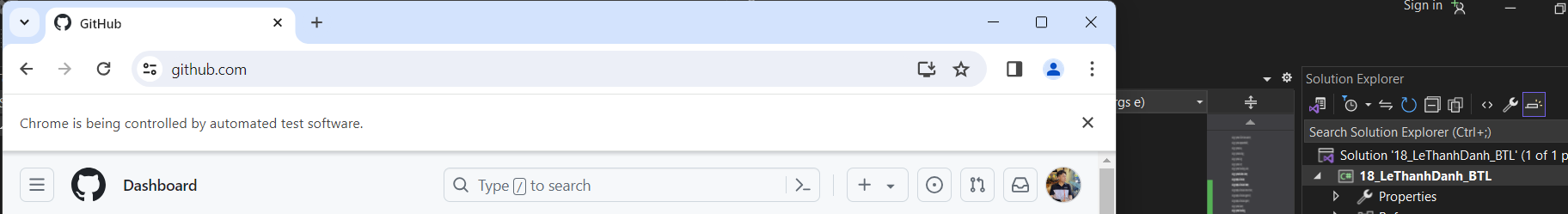
//18\_Lê Thanh Danh

//Click vào nút 'Log in'

IWebElement Danh\_18\_click = Danh\_18.FindElement(By.Name("commit"));

Danh\_18\_click.Click();

* Kết quả khi chạy code và “Thành công”



Test Case 2

* Sử dụng Name của 2 input, truyền vào phương thức FindElement của attribute driver để gọi đến 2 input và truyền giá trị qua SendKeys. Cuối cùng nhấn vào nút Sign In thông qua gọi Name và sửa dụng Click

//18\_Lê Thanh Danh

//Input của email để truyền giá trị từ SendKeys vào

IWebElement Danh\_18\_email = Danh\_18.FindElement(By.Name("login"));

Danh\_18\_email.SendKeys("Danh@gmail.com");

Thread.Sleep(1000);

//18\_Lê Thanh Danh

//Input của password để truyền giá trị từ SendKeys vào

IWebElement Danh\_18\_password = Danh\_18.FindElement(By.Name("password"));

Danh\_18\_password.SendKeys("lethanhdanh2003");

Thread.Sleep(2000);

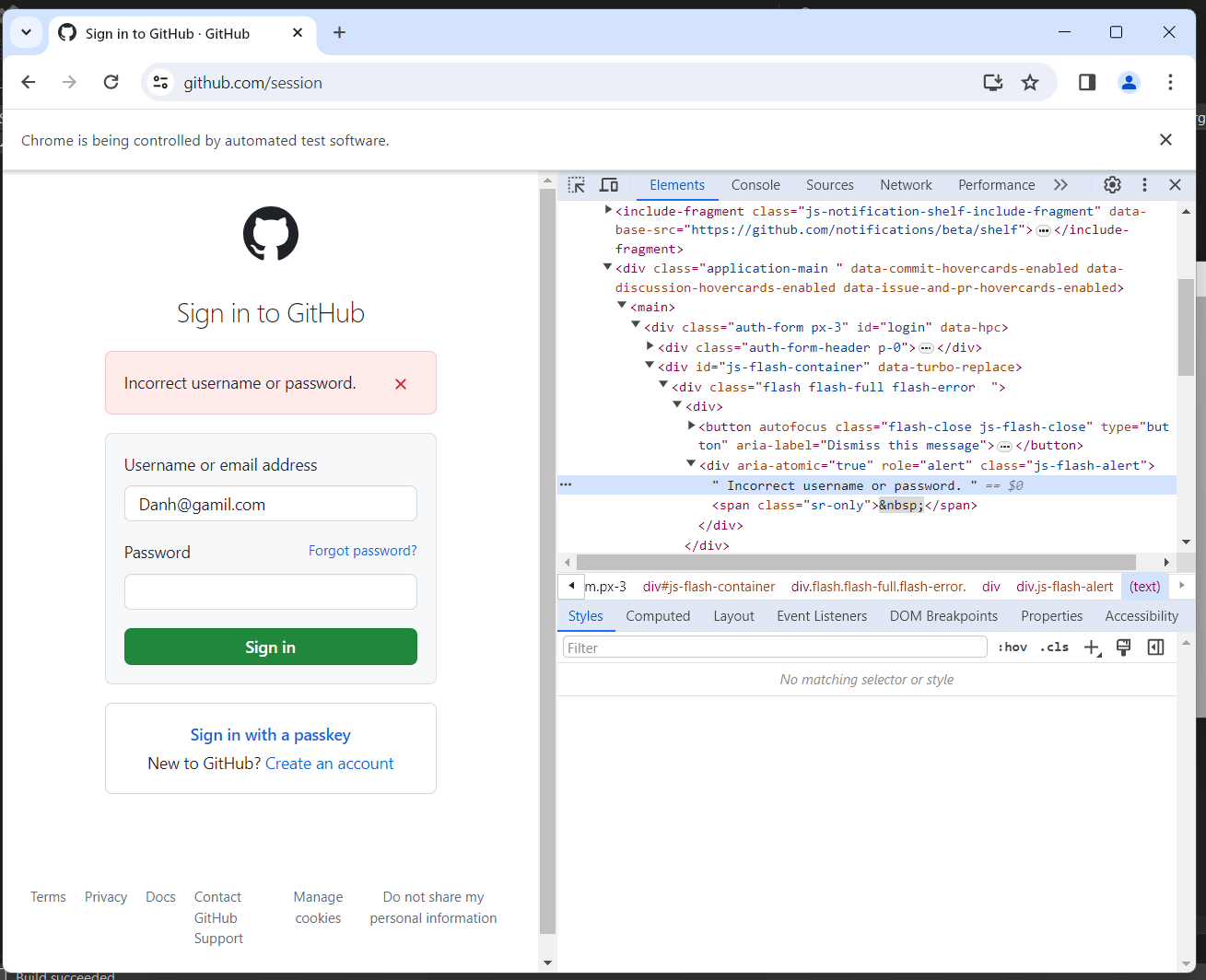
//18\_Lê Thanh Danh

//Click vào nút 'Log in'

IWebElement Danh\_18\_click = Danh\_18.FindElement(By.Name("commit"));

Danh\_18\_click.Click();

* Kết quả khi chạy code và “Hiện thị thông báo lỗi”



Test Case 4

* Sử dụng Name của 2 input, truyền vào phương thức FindElement của attribute driver để gọi đến 2 input và truyền giá trị qua SendKeys. Cuối cùng nhấn vào nút Sign In thông qua gọi Name và sửa dụng Click

//18\_Lê Thanh Danh

//Input của email để truyền giá trị từ SendKeys vào

IWebElement Danh\_18\_email = Danh\_18.FindElement(By.Name("login"));

Danh\_18\_email.SendKeys("Danh@gmail.com");

Thread.Sleep(1000);

//18\_Lê Thanh Danh

//Input của password để truyền giá trị từ SendKeys vào

IWebElement Danh\_18\_password = Danh\_18.FindElement(By.Name("password"));

Danh\_18\_password.SendKeys("123456");

Thread.Sleep(2000);

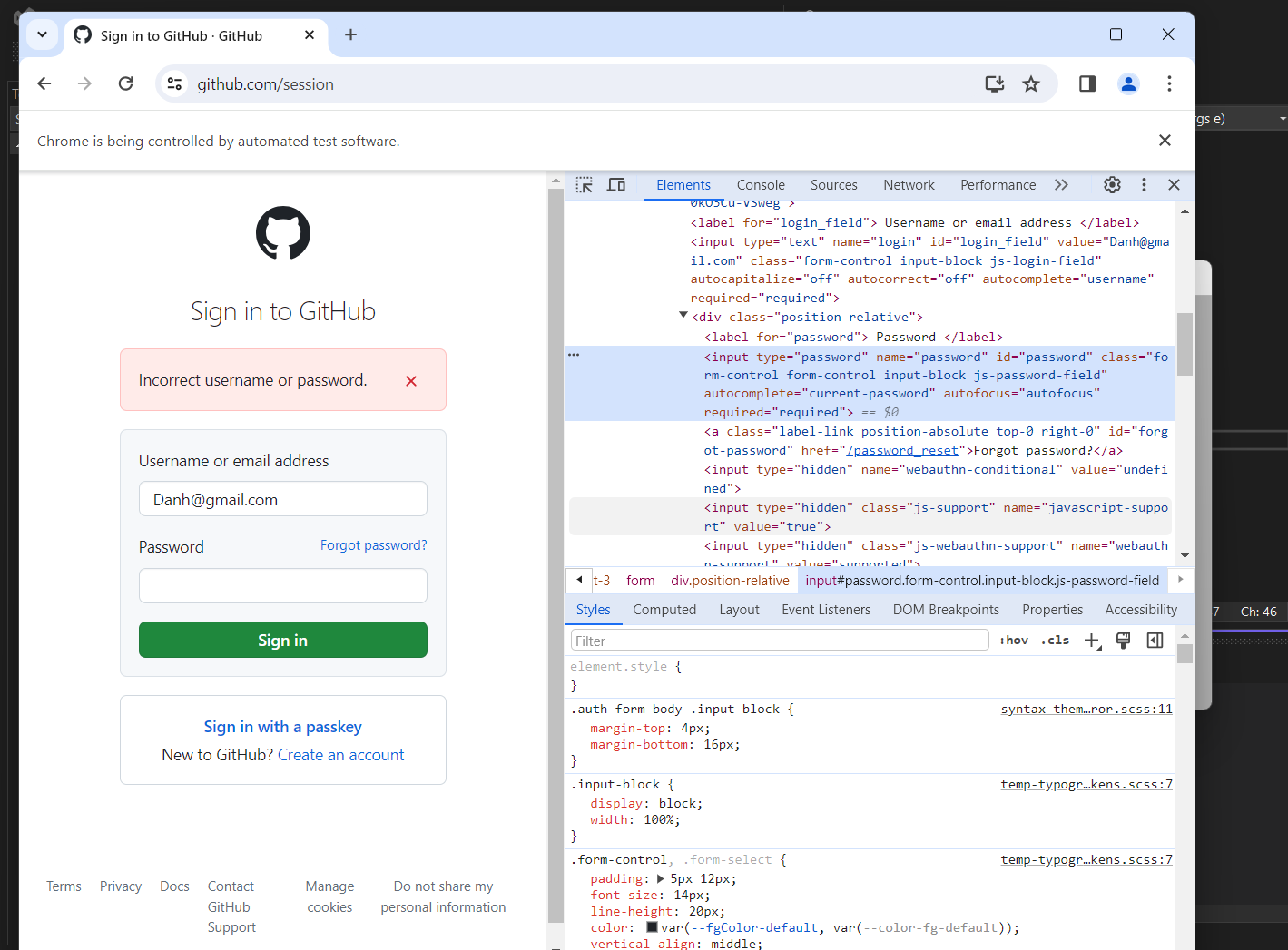
//18\_Lê Thanh Danh

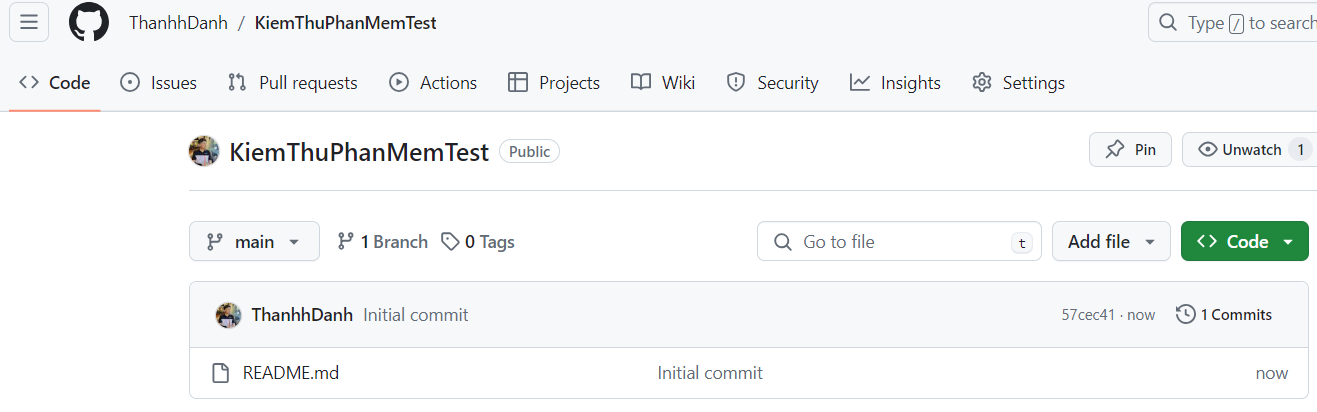
//Click vào nút 'Log in'

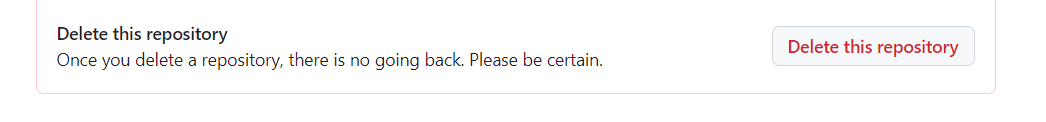
IWebElement Danh\_18\_click = Danh\_18.FindElement(By.Name("commit"));

Danh\_18\_click.Click();

* Kết quả khi chạy code và “Hiện thị thông báo lỗi”



1. Chức năng “Xóa”.
   1. Mô tả chức năng



* Chức năng “Xóa” repository: cho phép người dùng xóa một kho du liệu (repository) từ một dịch vụ quản lý mã nguồn.
* Điều kiện để “Xóa” repository:
* Kiểm tra quyền truy cập: Đảm bảo rằng người dùng đang thực hiện kiểm thử có quyền truy cập và quyền xóa repository
* Kiểm tra tồn tại: trước khi xóa, xem repository có tồn tại hay không .
* Xác nhận quyết định: Kiểm tra xem người dùng đã xác nhận quyết định xóa repository hay chưa.
* Kiểm tra lỗi: không nhập thiếu thông tin, không nhập sai định dạng trong.
  1. Kiểm thử hộp đen
     1. Bảng quyết định

Liệt kê:

|  |  |
| --- | --- |
| Các điều kiện | * Repository tồn tại * Quyền truy cập * Xác nhận quyết định * Kiểm tra lỗi |
| Các hàng động | * Xóa Repository * Không xóa được |

Bảng:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case | Repository tồn tại | Quyền truy cập | Xác nhận quyết định | Kiểm tra lỗi | Hành động | Kết quả |
| 1 | True | True | True | True | Xóa Repository | Pass |
| 2 | False | True | True | True | Không xóa được | Fail |
| 3 | True | False | True | True | Không xóa được | Fail |
| 4 | True | True | False | True | Không xóa được | Fail |
| 5 | True | True | True | False | Không xóa được | Fail |

* + 1. Giải thích code
       1. Chuẩn bị môi trường trong Visual Studio
* Ở chức năng “Đăng nhập” đã làm
  + - 1. Các hàm hỗ trợ
* Tất cả các phương thức dưới đây đều làm về chức năng “Xóa” repository
* Phương thức điều hướng tới trang https://github.com/ThanhhDanh/KiemThuPhanMemTest - Navigate

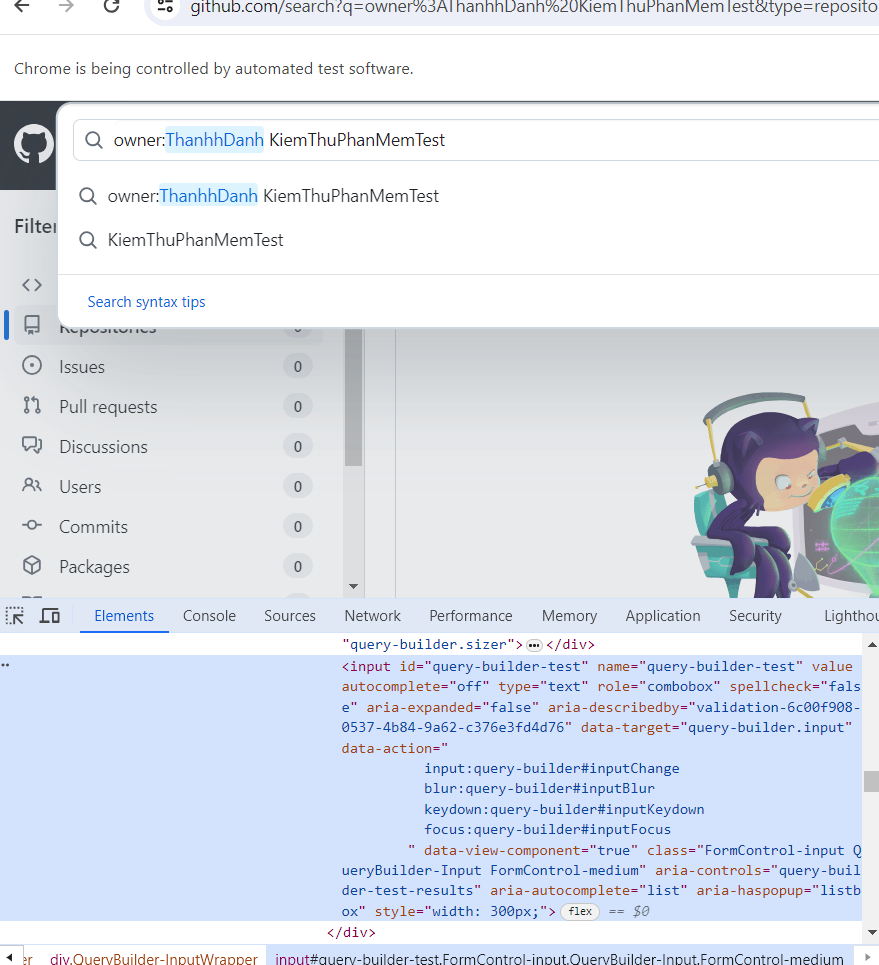
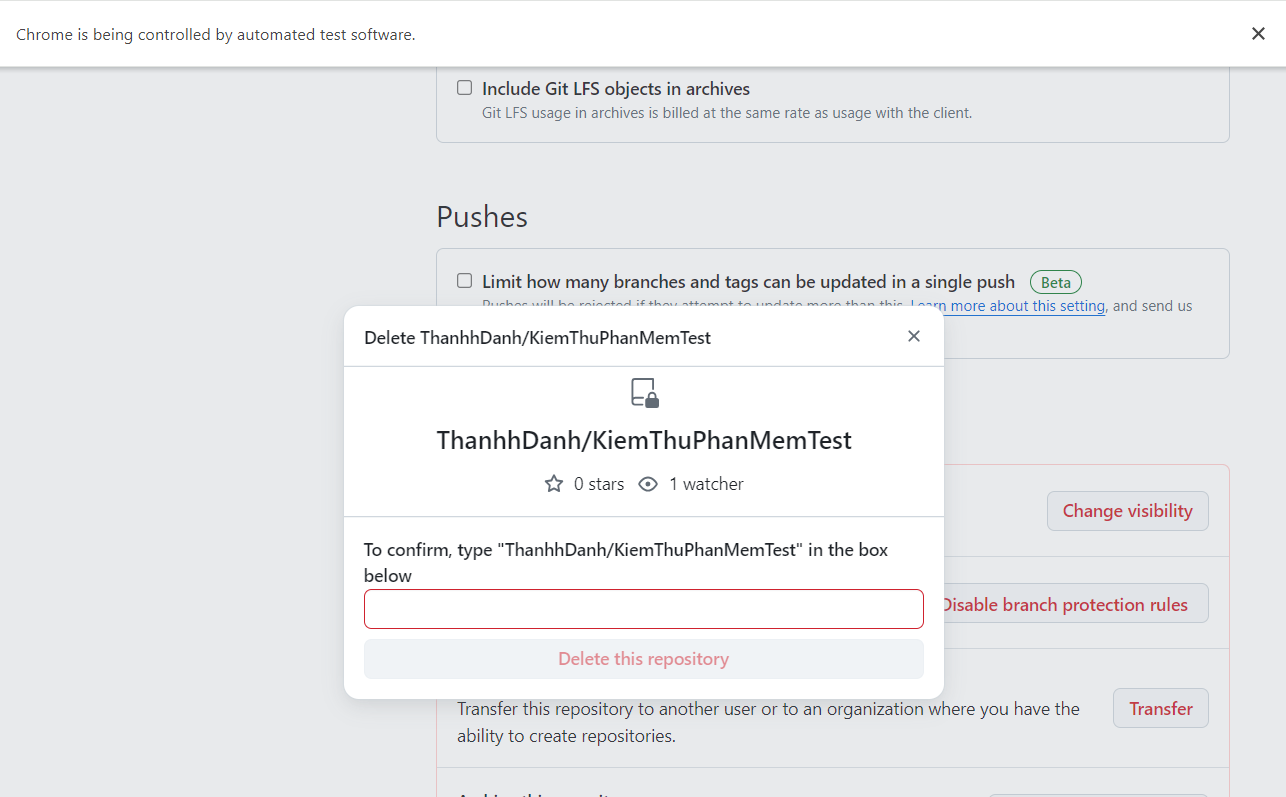
//18\_Lê Thanh Danh

IWebDriver Danh\_18 = new ChromeDriver(TDanh\_18);

Danh\_18.Url = "https://github.com/ThanhhDanh";//Gán URL

Danh\_18.Navigate(); //Điều hướng để mở trang url đã gán bằng navigate

* Phương thức “Xóa” repository
* Tìm Repository xóa và nhập tên Repository đó



Test Case 1

* Sử dụng Class Name để chọn ô tìm kiếm và Name để lấy input, truyền vào phương thức FindElement của attribute driver để gọi đến và truyền giá trị qua SendKeys. Cuối cùng sử dụng Keys.Enter để khi thông tin truyền vào và Enter

//18\_Lê Thanh Danh

//Click vào ô tìm kiếm bằng ClassName

Danh\_18.FindElement(By.ClassName("search-input-container")).Click();

Thread.Sleep(1000);

//18\_Lê Thanh Danh

//Sử dụng Name để truyền SendKeys cần tìm kiếm

IWebElement Danh\_18\_link = Danh\_18.FindElement(By.Name("query-builder-test"));

Danh\_18\_link.SendKeys("KiemThuPhanMemTest" + OpenQA.Selenium.Keys.Enter);

Thread.Sleep(2000);

* Sử dụng Name của 2 input, truyền vào phương thức FindElement của attribute driver để gọi đến 2 input và truyền giá trị qua SendKeys. Cuối cùng nhấn vào nút Sign In thông qua gọi Name và sửa dụng Click

//18\_Lê Thanh Danh

//Nếu muốn xóa thì phải đăng nhập nếu trước đó chưa đăng nhập

Danh\_18.FindElement(By.LinkText("Sign in")).Click();

Thread.Sleep(1000);

//18\_Lê Thanh Danh

//Input của email để truyền giá trị từ SendKeys vào

IWebElement Danh\_18\_email = Danh\_18.FindElement(By.Name("login"));

Danh\_18\_email.SendKeys("ThanhhDanh");

Thread.Sleep(1000);

//18\_Lê Thanh Danh

//Input của password để truyền giá trị từ SendKeys vào

IWebElement Danh\_18\_password = Danh\_18.FindElement(By.Name("password"));

Danh\_18\_password.SendKeys("lethanhdanh2003");

Thread.Sleep(2000);

//18\_Lê Thanh Danh

//Click vào nút 'Log in'

IWebElement Danh\_18\_click = Danh\_18.FindElement(By.Name("commit"));

Danh\_18\_click.Click();

Thread.Sleep(1000);

* Sử dụng Link Text để tới tới cái trang setting của Repository đó, dùng Id để chọn những input để xác thực và dùng Name để truyền giá trị qua SendKeys

//18\_Lê Thanh Danh

Danh\_18.FindElement(By.LinkText("Settings")).Click();

Thread.Sleep(1000);

//18\_Lê Thanh Danh

//Click vào "Delete this repository"

Danh\_18.FindElement(By.Id("dialog-show-repo-delete-menu-dialog")).Click();

Thread.Sleep(2000);

//18\_Lê Thanh Danh

//Sau đó Click vào "I want to delete this repository"

Danh\_18.FindElement(By.Id("repo-delete-proceed-button-container")).Click();

Thread.Sleep(2000);

//18\_Lê Thanh Danh

//Sau đó Click tiếp vào "I have read and understand these effects"

Danh\_18.FindElement(By.Id("repo-delete-proceed-button-container")).Click();

Thread.Sleep(2000);

//18\_Lê Thanh Danh

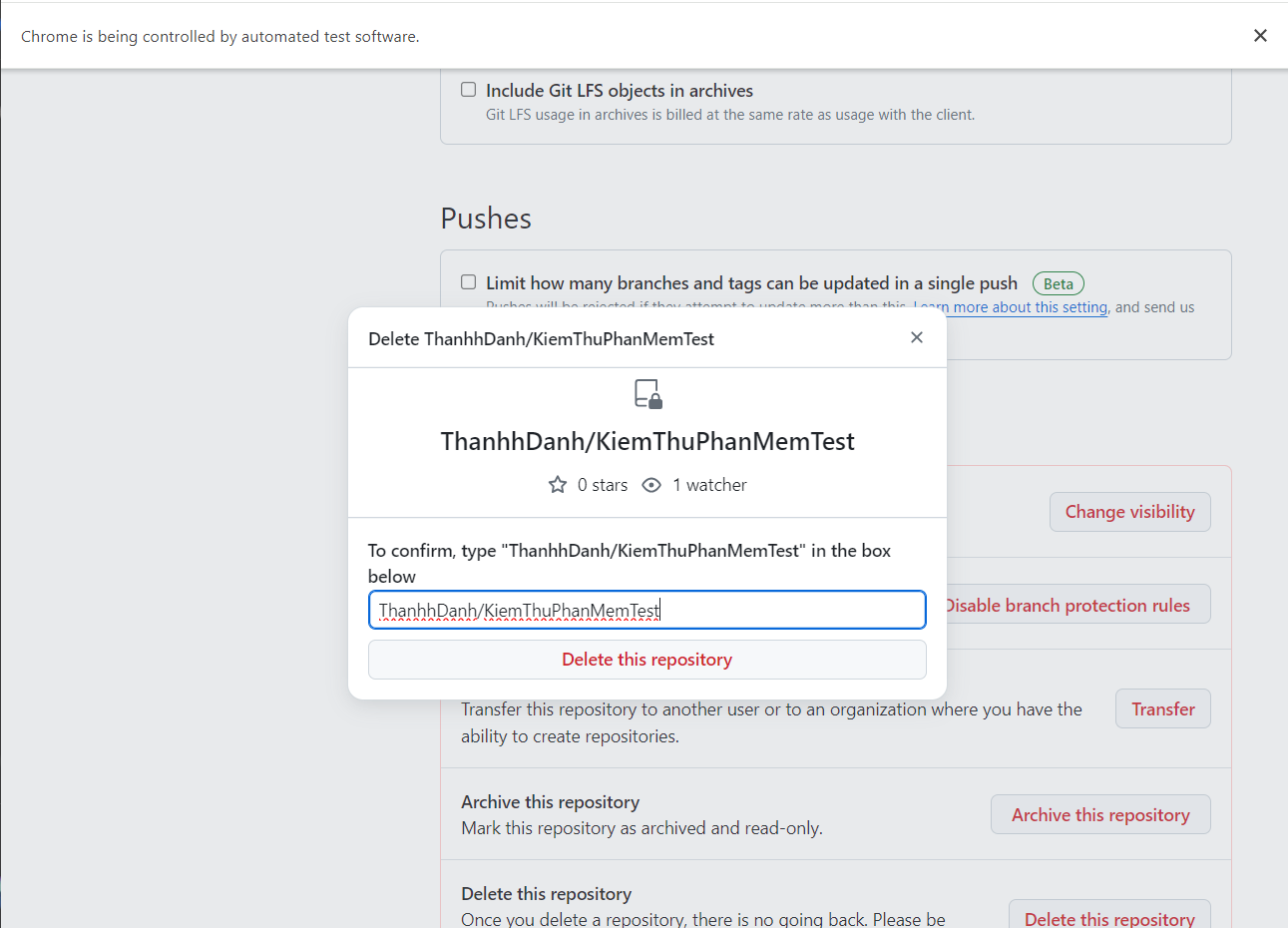
//Cuối cùng điền vào "To confirm, type "Repository" in the box below" muốn xóa qua SendKeys

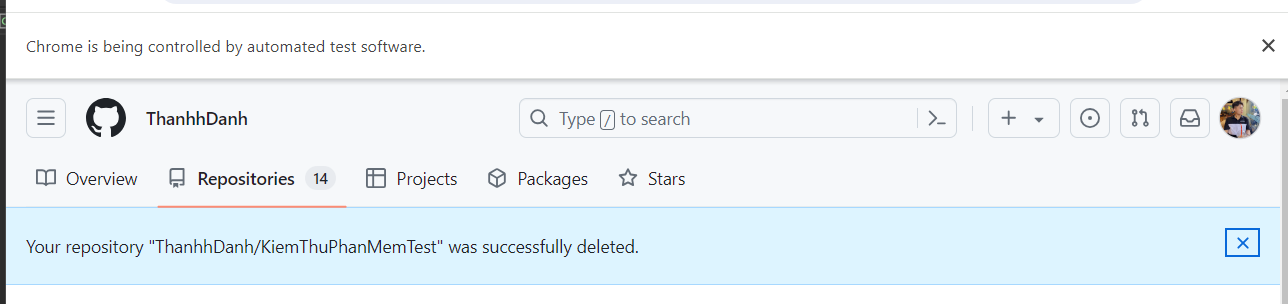
IWebElement Danh\_18\_search = Danh\_18.FindElement(By.Name("verification\_field"));

Danh\_18\_search.SendKeys("ThanhhDanh/KiemThuPhanMemTesttt");

Thread.Sleep(2000);

* Đây là kết quả khi chạy tới chỗ nhập Repository



* Kết quả cuối khi chạy code

Test Case 2

* Sử dụng Class Name để chọn ô tìm kiếm và Name để lấy input, truyền vào phương thức FindElement của attribute driver để gọi đến và truyền giá trị qua SendKeys. Cuối cùng sử dụng Keys.Enter để khi thông tin truyền vào và Enter

//18\_Lê Thanh Danh

//Click vào ô tìm kiếm bằng ClassName

Danh\_18.FindElement(By.ClassName("search-input-container")).Click();

Thread.Sleep(1000);

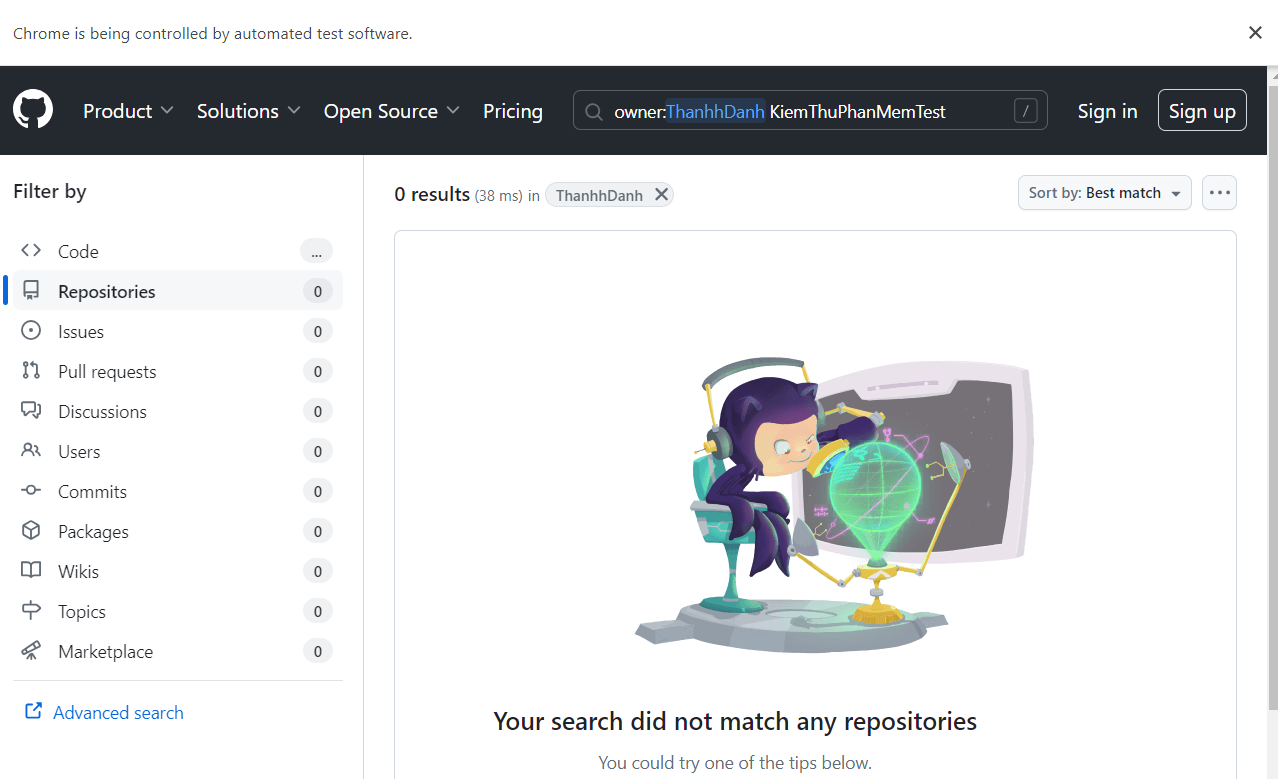
//18\_Lê Thanh Danh

//Sử dụng Name để truyền SendKeys cần tìm kiếm

IWebElement Danh\_18\_link = Danh\_18.FindElement(By.Name("query-builder-test"));

Danh\_18\_link.SendKeys("KiemThuPhanMemTest" + OpenQA.Selenium.Keys.Enter);

Thread.Sleep(2000);

* Kết quả khi chạy code

Test Case 3

* Sử dụng Class Name để chọn ô tìm kiếm và Name để lấy input, truyền vào phương thức FindElement của attribute driver để gọi đến và truyền giá trị qua SendKeys. Cuối cùng sử dụng Keys.Enter để khi thông tin truyền vào và Enter

//18\_Lê Thanh Danh

//Click vào ô tìm kiếm bằng ClassName

Danh\_18.FindElement(By.ClassName("search-input-container")).Click();

Thread.Sleep(1000);

//18\_Lê Thanh Danh

//Sử dụng Name để truyền SendKeys cần tìm kiếm

IWebElement Danh\_18\_link = Danh\_18.FindElement(By.Name("query-builder-test"));

Danh\_18\_link.SendKeys("KiemThuPhanMemTest" + OpenQA.Selenium.Keys.Enter);

Thread.Sleep(2000);

* Sử dụng Name của 2 input, truyền vào phương thức FindElement của attribute driver để gọi đến 2 input và truyền giá trị qua SendKeys. Cuối cùng nhấn vào nút Sign In thông qua gọi Name và sửa dụng Click. (Trong test case này là quyền truy cập nên sẽ làm sai mật khẩu hoặc không đăng nhập)

//18\_Lê Thanh Danh

//Nếu muốn xóa thì phải đăng nhập nếu trước đó chưa đăng nhập

Danh\_18.FindElement(By.LinkText("Sign in")).Click();

Thread.Sleep(1000);

//18\_Lê Thanh Danh

//Input của email để truyền giá trị từ SendKeys vào

IWebElement Danh\_18\_email = Danh\_18.FindElement(By.Name("login"));

Danh\_18\_email.SendKeys("ThanhhDanh");

Thread.Sleep(1000);

//18\_Lê Thanh Danh

//Input của password để truyền giá trị từ SendKeys vào

IWebElement Danh\_18\_password = Danh\_18.FindElement(By.Name("password"));

Danh\_18\_password.SendKeys("lethanhdanh2003");

Thread.Sleep(2000);

//18\_Lê Thanh Danh

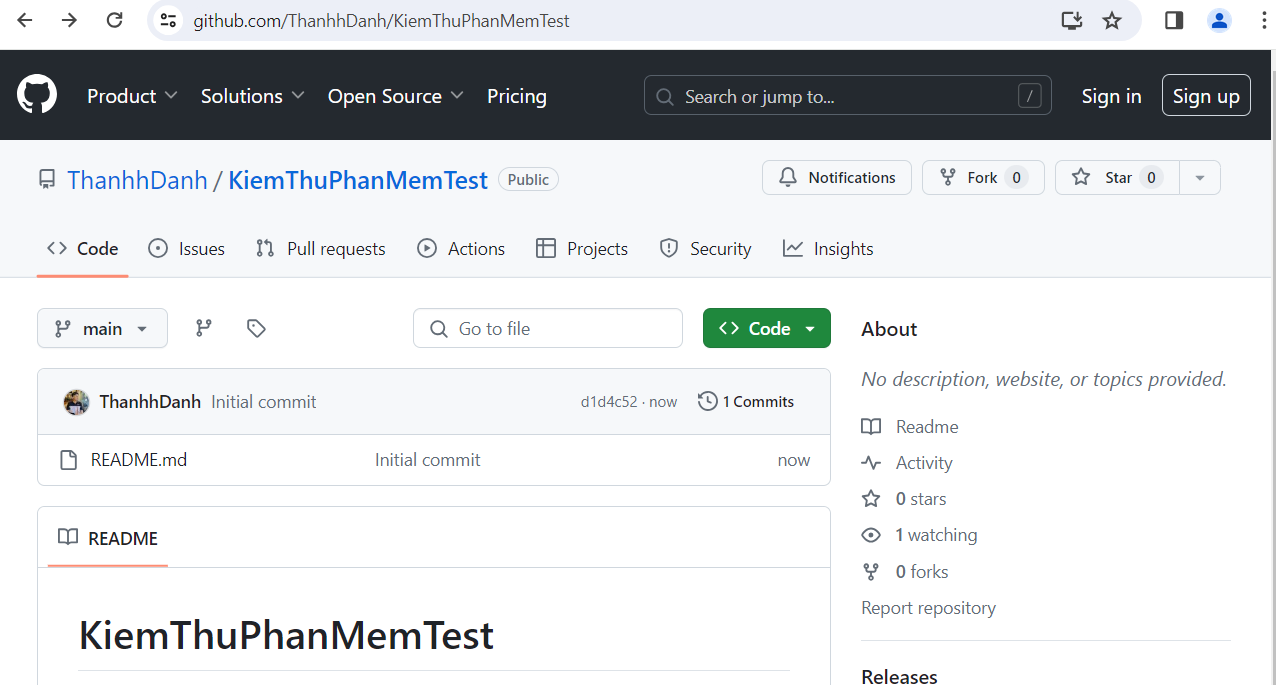
//Click vào nút 'Log in'

IWebElement Danh\_18\_click = Danh\_18.FindElement(By.Name("commit"));

Danh\_18\_click.Click();

Thread.Sleep(1000);

* Kết quả khi chạy code



Test Case 5

* Sử dụng Class Name để chọn ô tìm kiếm và Name để lấy input, truyền vào phương thức FindElement của attribute driver để gọi đến và truyền giá trị qua SendKeys. Cuối cùng sử dụng Keys.Enter để khi thông tin truyền vào và Enter

//18\_Lê Thanh Danh

//Click vào ô tìm kiếm bằng ClassName

Danh\_18.FindElement(By.ClassName("search-input-container")).Click();

Thread.Sleep(1000);

//18\_Lê Thanh Danh

//Sử dụng Name để truyền SendKeys cần tìm kiếm

IWebElement Danh\_18\_link = Danh\_18.FindElement(By.Name("query-builder-test"));

Danh\_18\_link.SendKeys("KiemThuPhanMemTest" + OpenQA.Selenium.Keys.Enter);

Thread.Sleep(2000);

* Sử dụng Name của 2 input, truyền vào phương thức FindElement của attribute driver để gọi đến 2 input và truyền giá trị qua SendKeys. Cuối cùng nhấn vào nút Sign In thông qua gọi Name và sửa dụng Click

//18\_Lê Thanh Danh

//Nếu muốn xóa thì phải đăng nhập nếu trước đó chưa đăng nhập

Danh\_18.FindElement(By.LinkText("Sign in")).Click();

Thread.Sleep(1000);

//18\_Lê Thanh Danh

//Input của email để truyền giá trị từ SendKeys vào

IWebElement Danh\_18\_email = Danh\_18.FindElement(By.Name("login"));

Danh\_18\_email.SendKeys("ThanhhDanh");

Thread.Sleep(1000);

//18\_Lê Thanh Danh

//Input của password để truyền giá trị từ SendKeys vào

IWebElement Danh\_18\_password = Danh\_18.FindElement(By.Name("password"));

Danh\_18\_password.SendKeys("lethanhdanh2003");

Thread.Sleep(2000);

//18\_Lê Thanh Danh

//Click vào nút 'Log in'

IWebElement Danh\_18\_click = Danh\_18.FindElement(By.Name("commit"));

Danh\_18\_click.Click();

Thread.Sleep(1000);

* Sử dụng Link Text để tới tới cái trang setting của Repository đó, dùng Id để chọn những input để xác thực và dùng Name để truyền giá trị qua SendKeys

//18\_Lê Thanh Danh

Danh\_18.FindElement(By.LinkText("Settings")).Click();

Thread.Sleep(1000);

//18\_Lê Thanh Danh

//Click vào "Delete this repository"

Danh\_18.FindElement(By.Id("dialog-show-repo-delete-menu-dialog")).Click();

Thread.Sleep(2000);

//18\_Lê Thanh Danh

//Sau đó Click vào "I want to delete this repository"

Danh\_18.FindElement(By.Id("repo-delete-proceed-button-container")).Click();

Thread.Sleep(2000);

//18\_Lê Thanh Danh

//Sau đó Click tiếp vào "I have read and understand these effects"

Danh\_18.FindElement(By.Id("repo-delete-proceed-button-container")).Click();

Thread.Sleep(2000);

//18\_Lê Thanh Danh

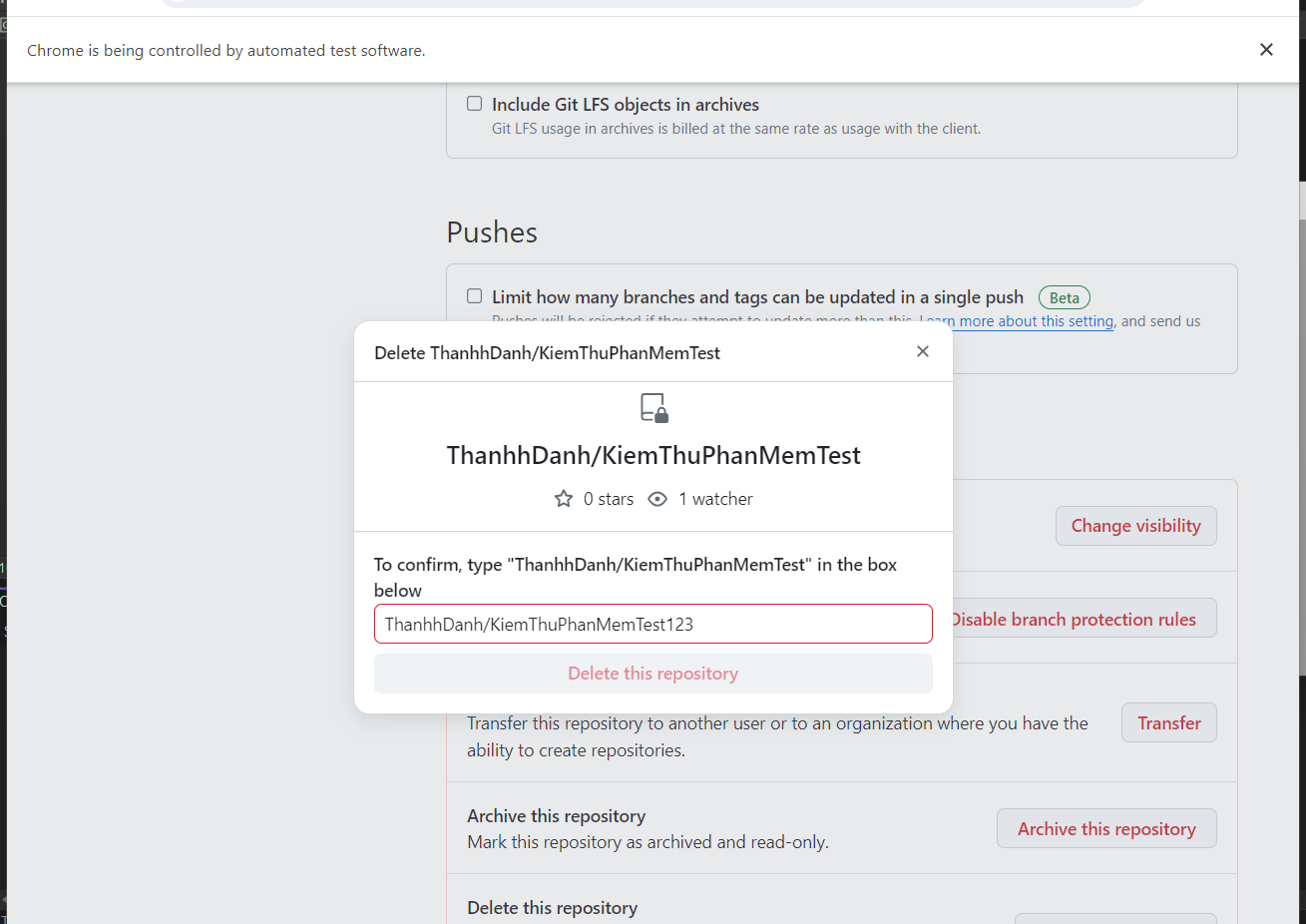
//Cuối cùng điền vào "To confirm, type "Repository" in the box below" muốn xóa qua SendKeys

IWebElement Danh\_18\_search = Danh\_18.FindElement(By.Name("verification\_field"));

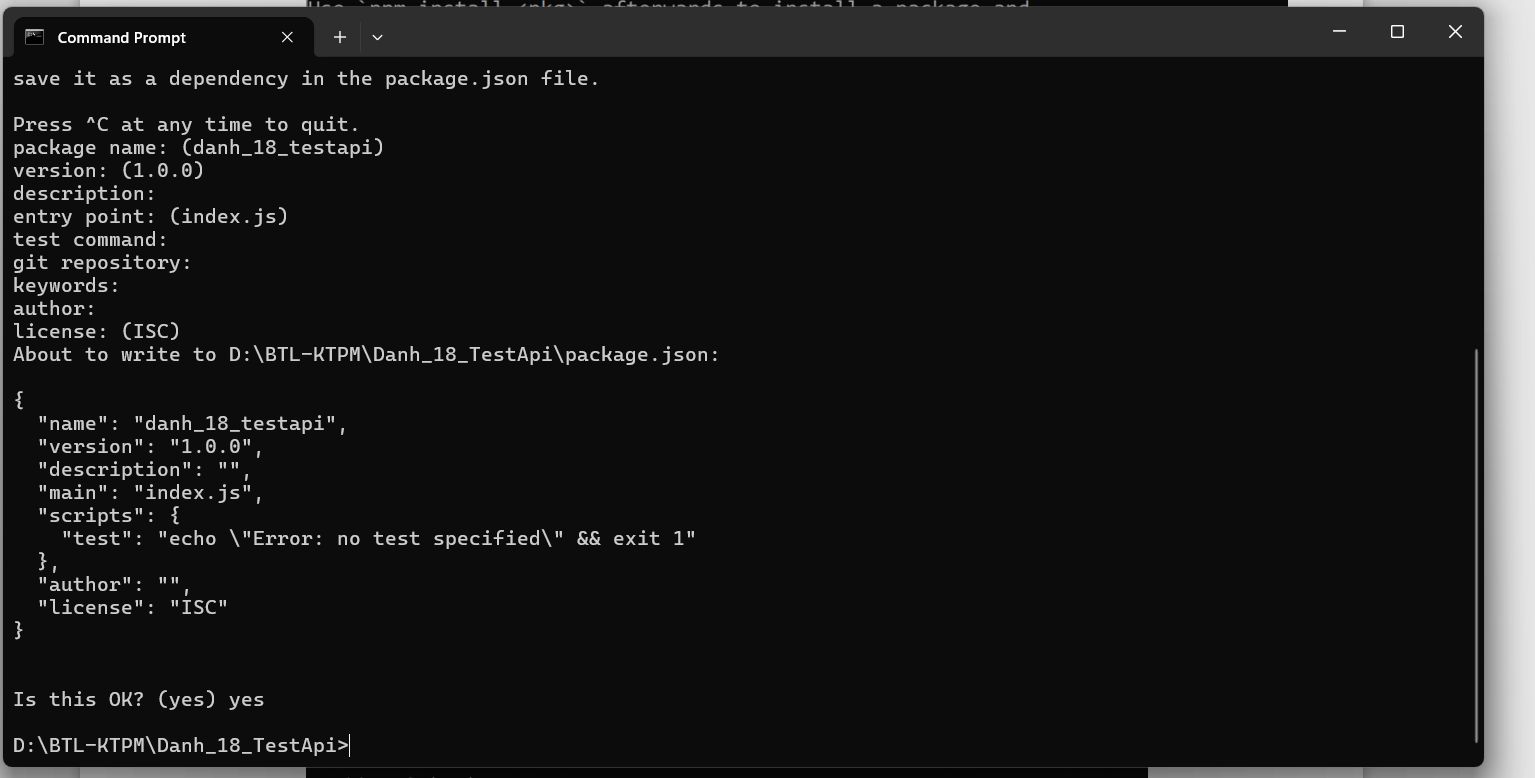
Danh\_18\_search.SendKeys("ThanhhDanh/KiemThuPhanMemTest123");

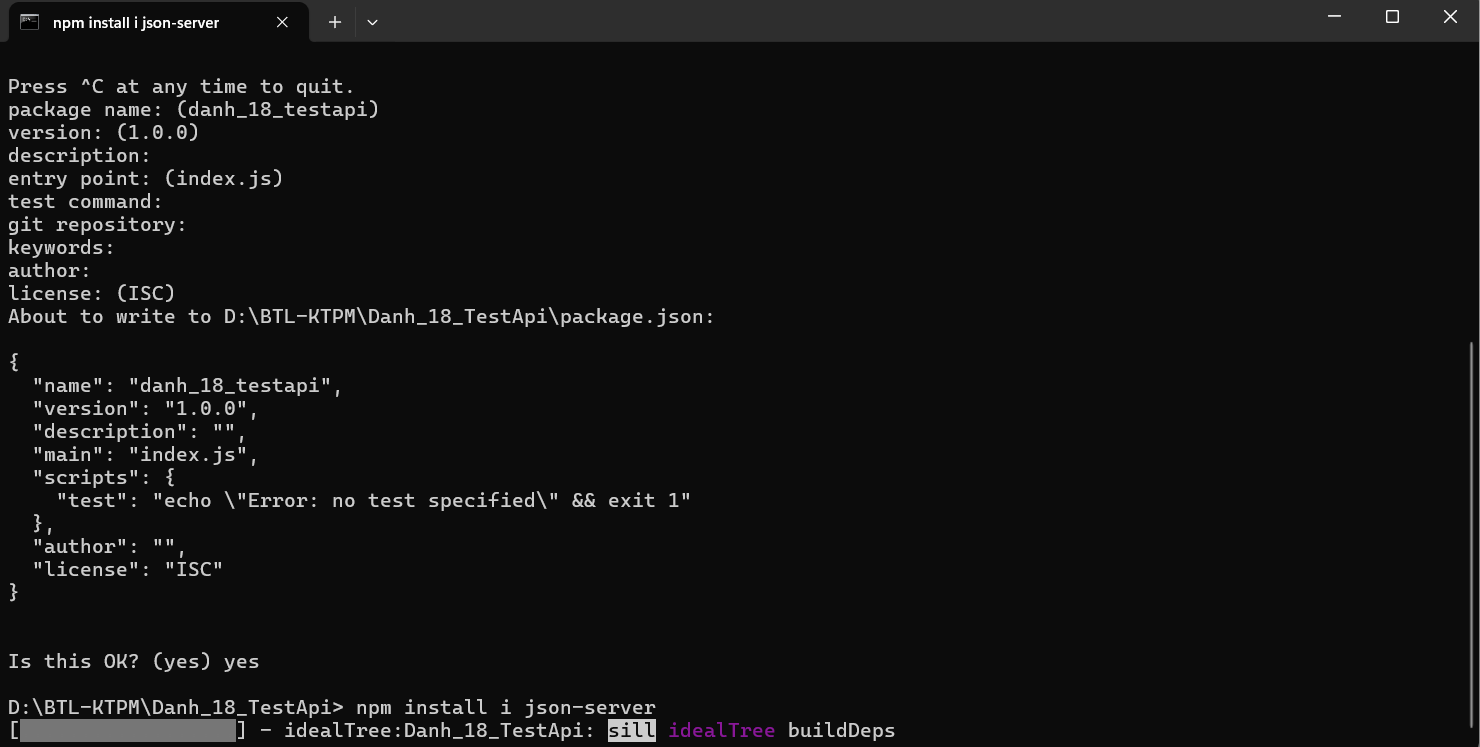
Thread.Sleep(2000);

* Kết quả khi chạy code

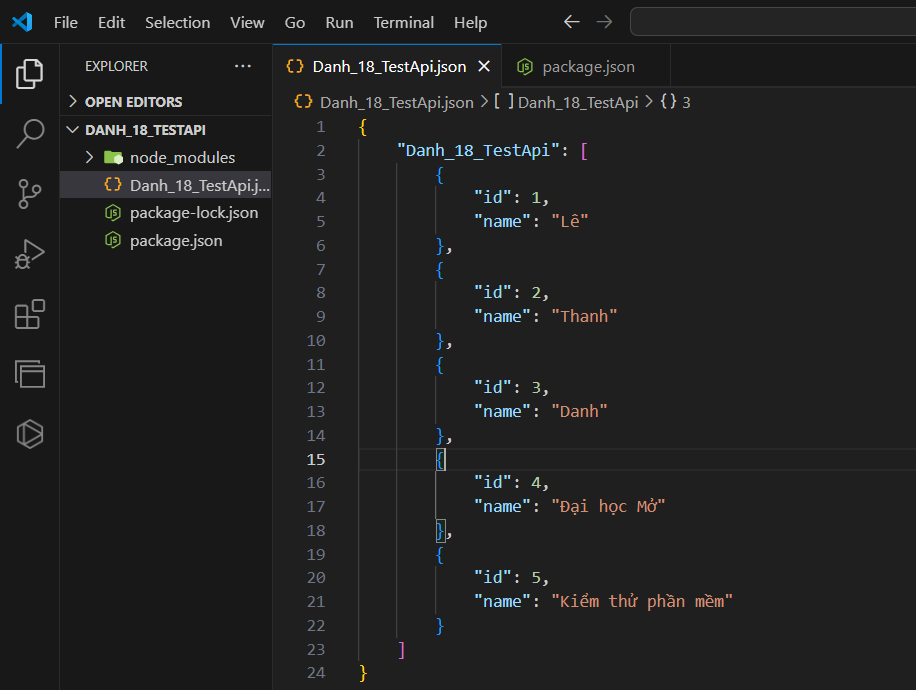


Test API

1. Tạo JsonServer
2. Cài đặt Node, tạo JsonServer trên thư mục và chạy lệnh npm init

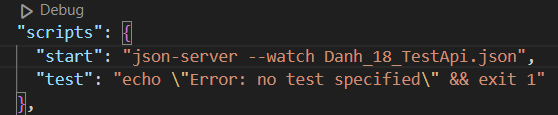


1. Tạo một file json trong thư mục

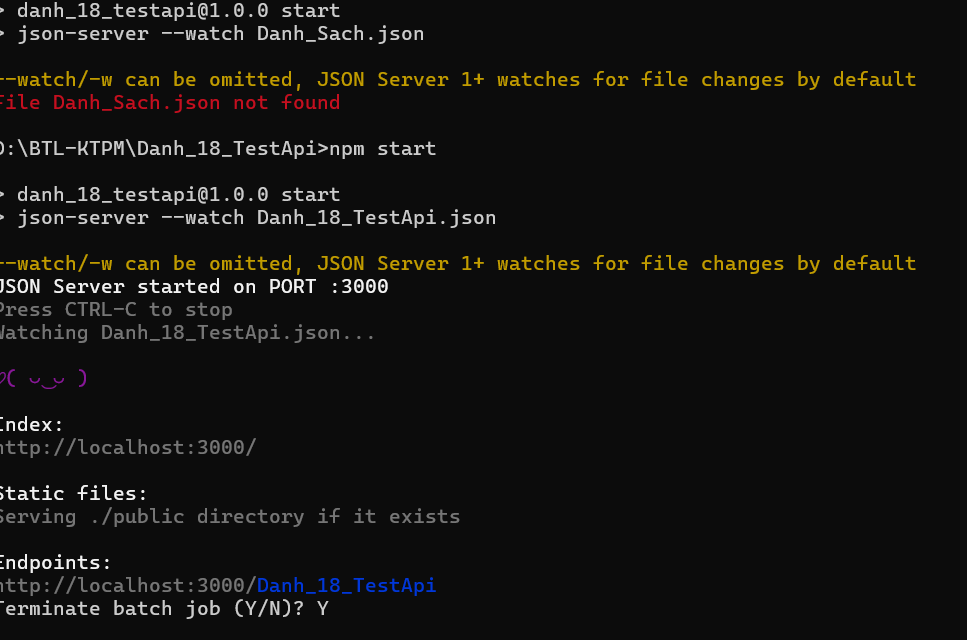


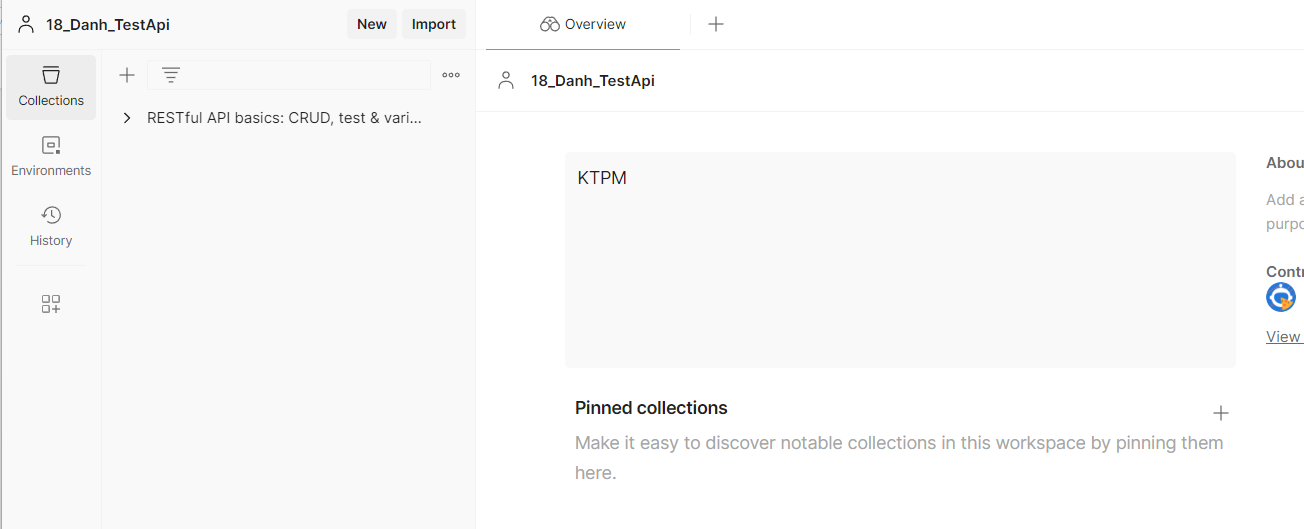
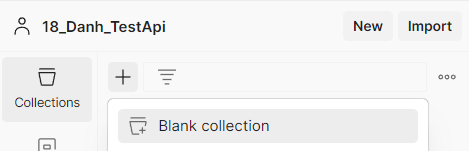
1. Mở file package.json trong thư mục

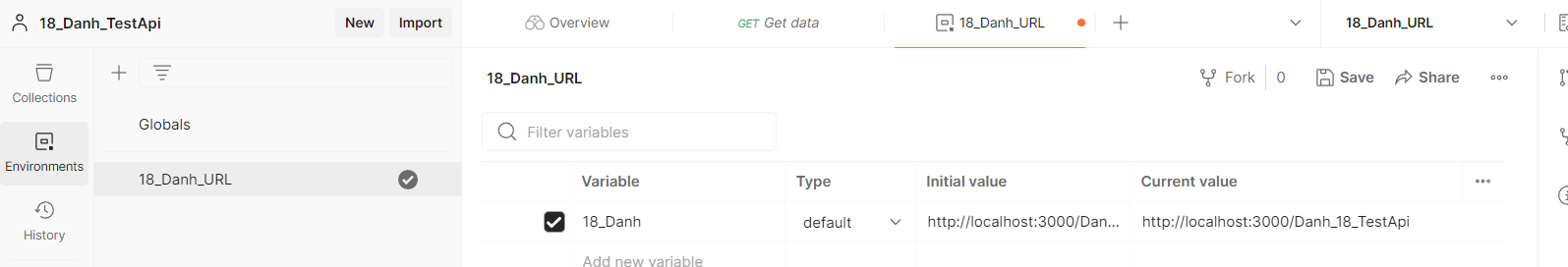
Thêm "start": "json-server --watch Danh\_18\_TestApi.json"



1. Cài đặt JsonServer, gõ lệnh npm install json-server và bắt đầu chạy gõ npm start

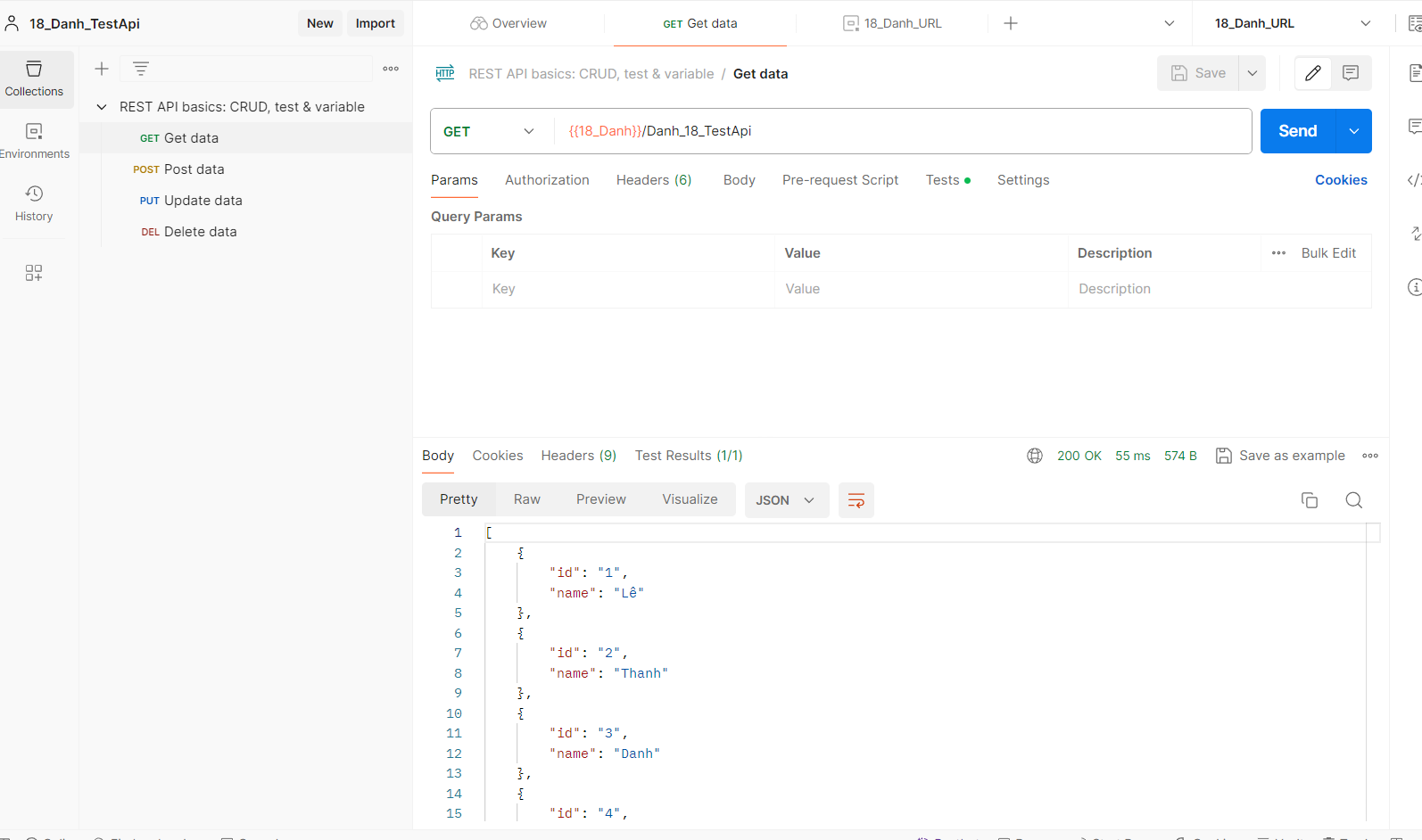


1. Thiệt lập Postman
2. Tạo workspace
3. Tạo collection
4. Tạo biến global

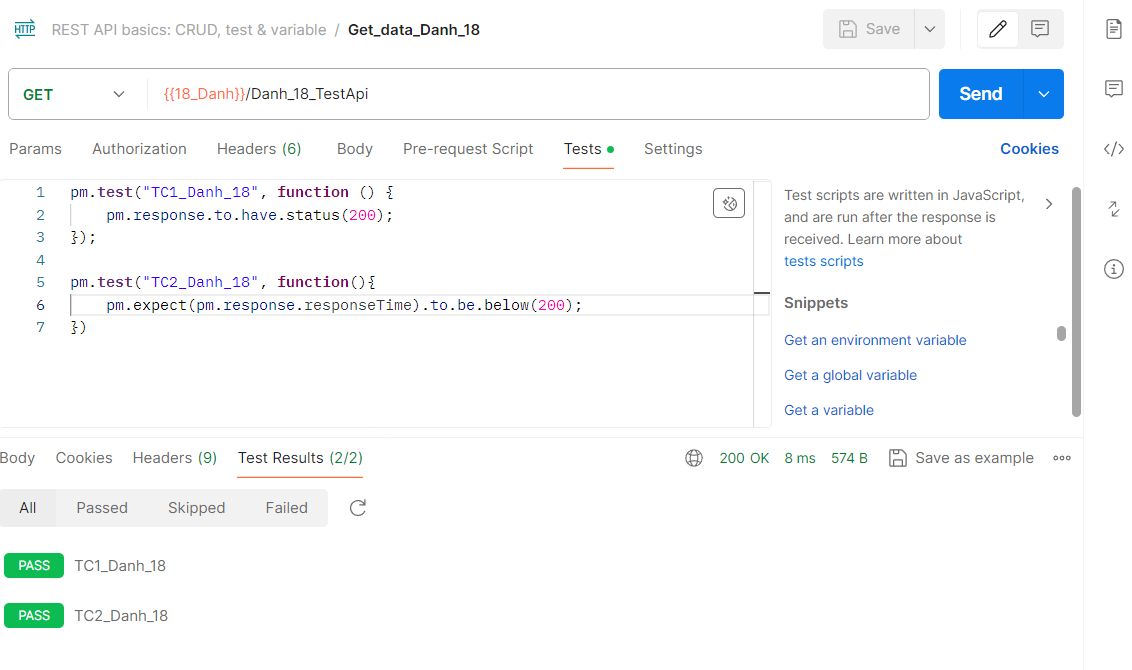


Sử dụng {{18\_Danh}}

1. Test API



1. Viết test case
2. Phương thức GET



//TC1 check status

pm.**test**("TC1\_Danh\_18", **function** () {

    pm.response.to.have.status(200);

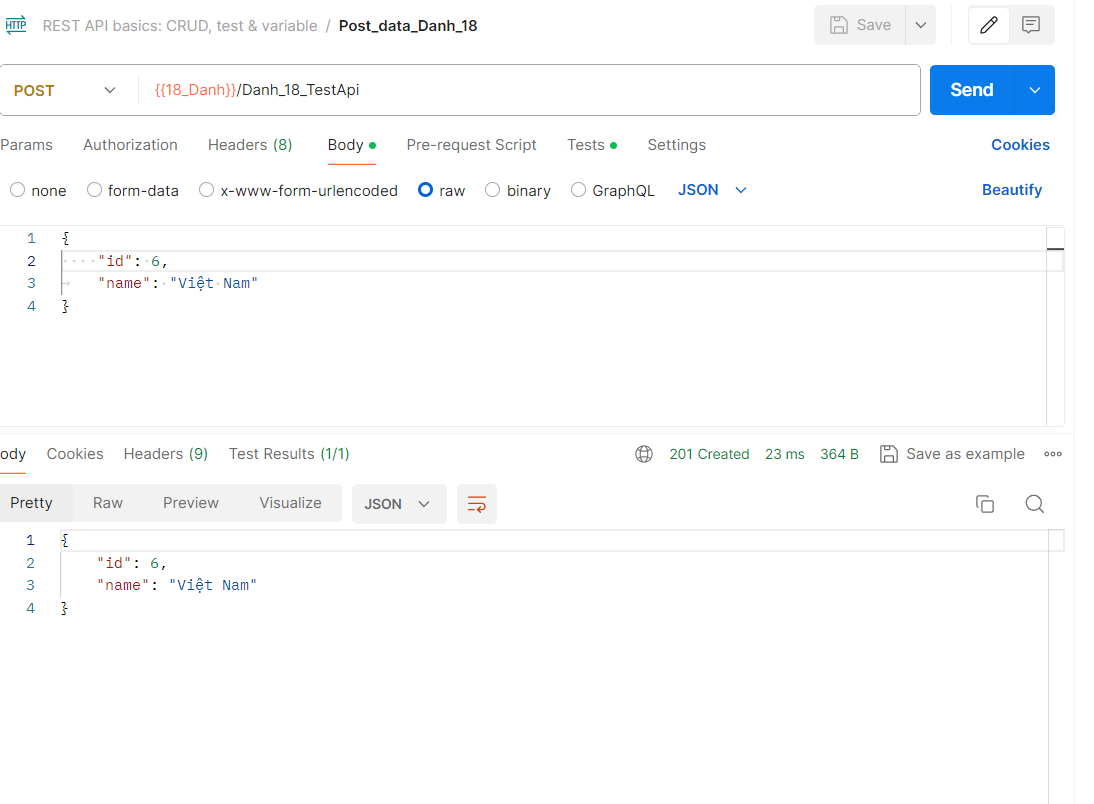
});

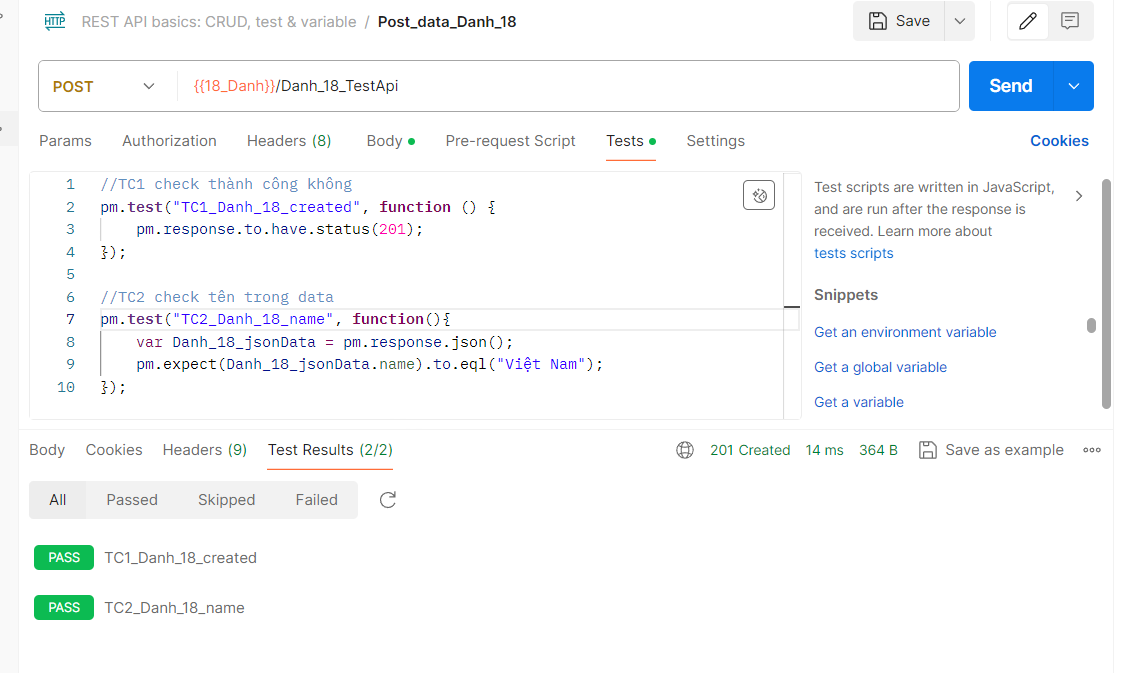
//TC2 check response

pm.**test**("TC2\_Danh\_18", **function**(){

    pm.expect(pm.response.responseTime).to.be.below(200);

})

1. Phương thức POST



//TC1 check thành công không

pm.**test**("TC1\_Danh\_18\_created", **function** () {

    pm.response.to.have.status(201);

});

//TC2 check tên trong data

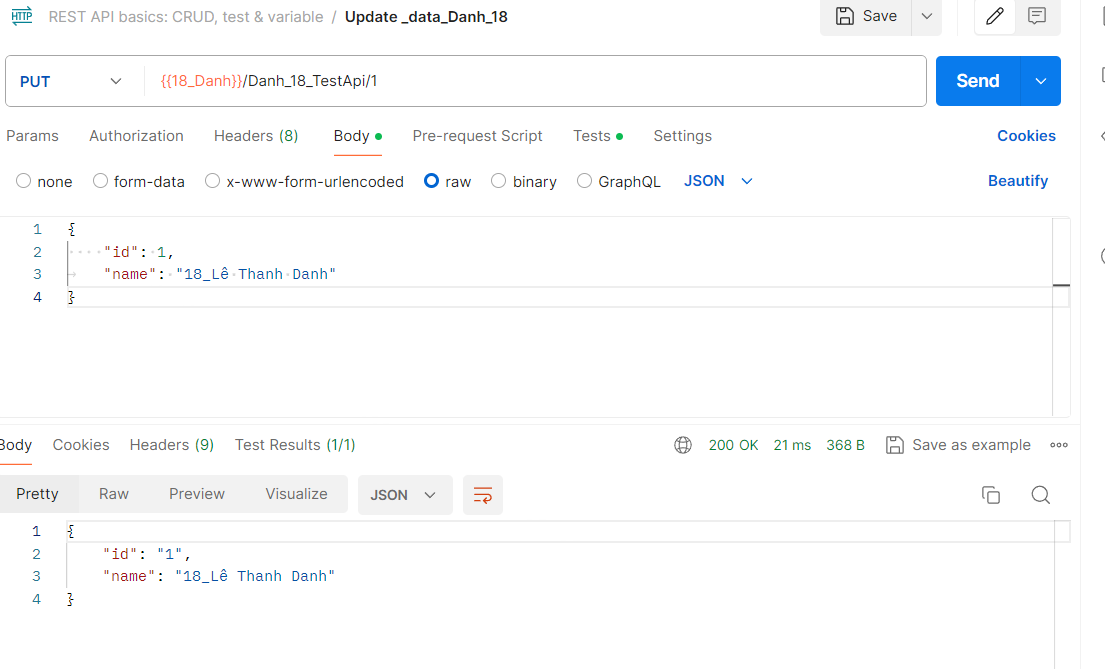
pm.**test**("TC2\_Danh\_18\_name", **function**(){

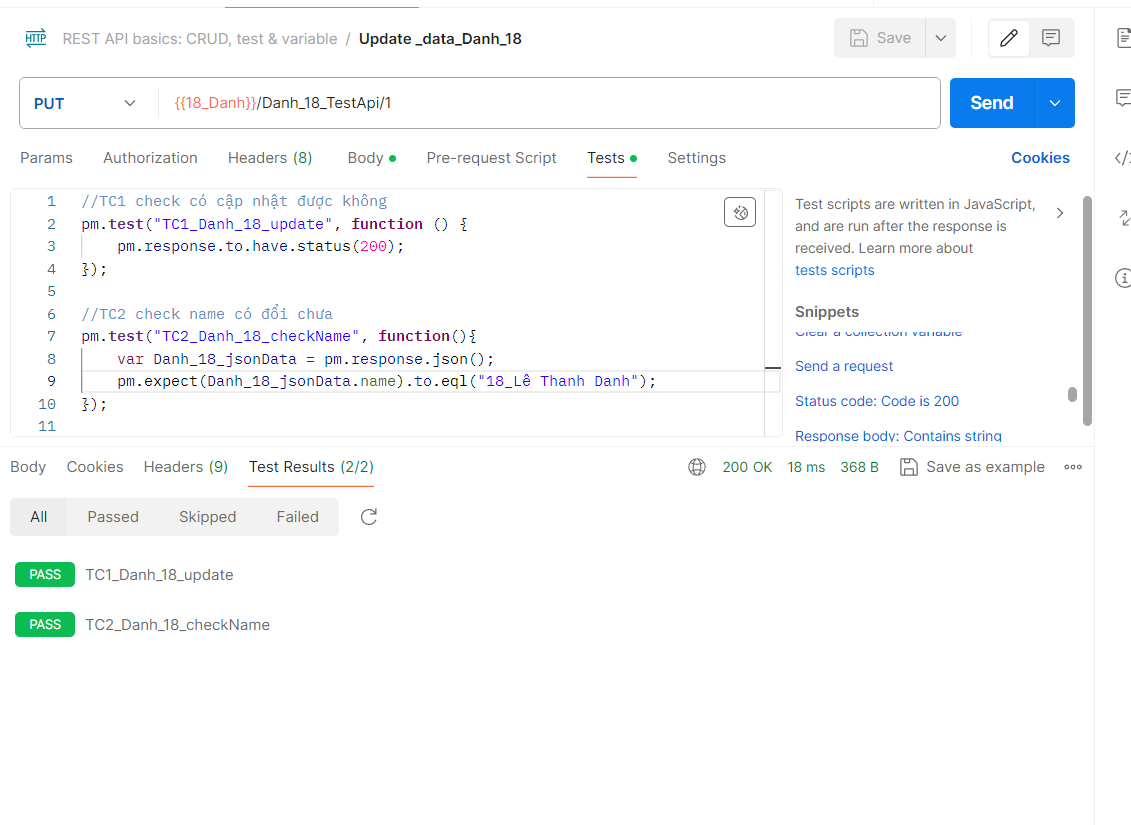
    var Danh\_18\_jsonData **=** pm.response.json();

    pm.expect(Danh\_18\_jsonData.name).to.eql("Việt Nam");

});

1. Phương thức PUT





//TC1 check có cập nhật được không

pm.**test**("TC1\_Danh\_18\_update", **function** () {

    pm.response.to.have.status(200);

});

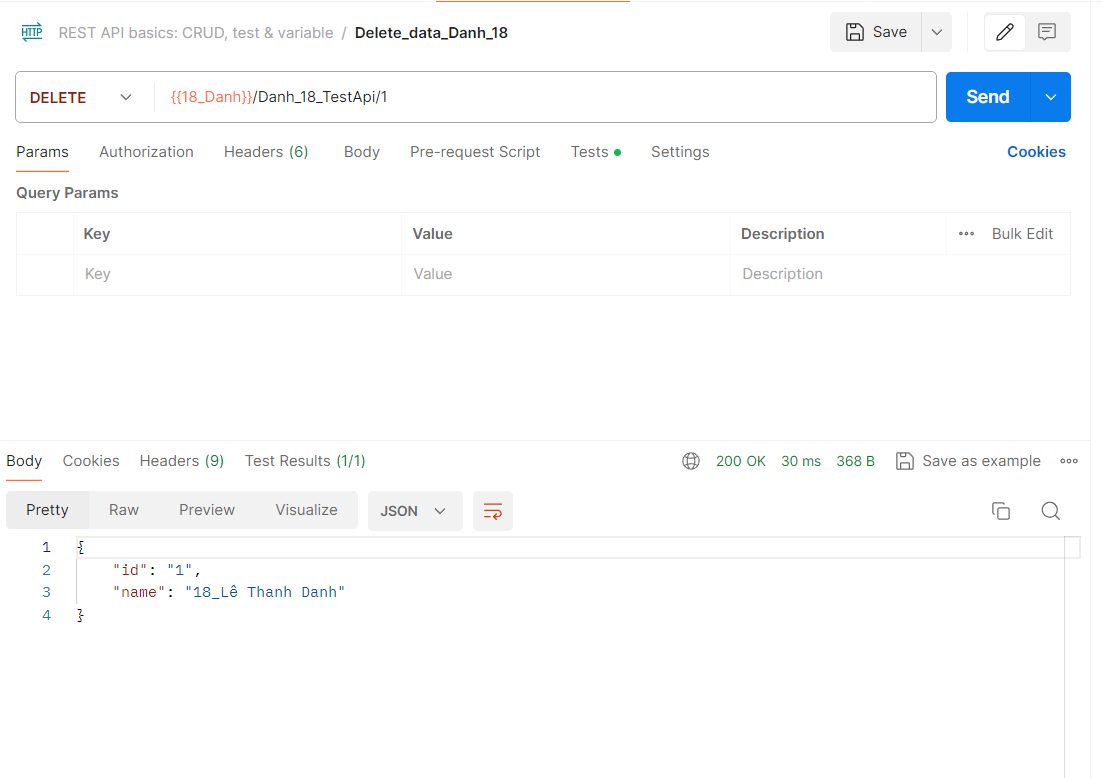
//TC2 check name có đổi chưa

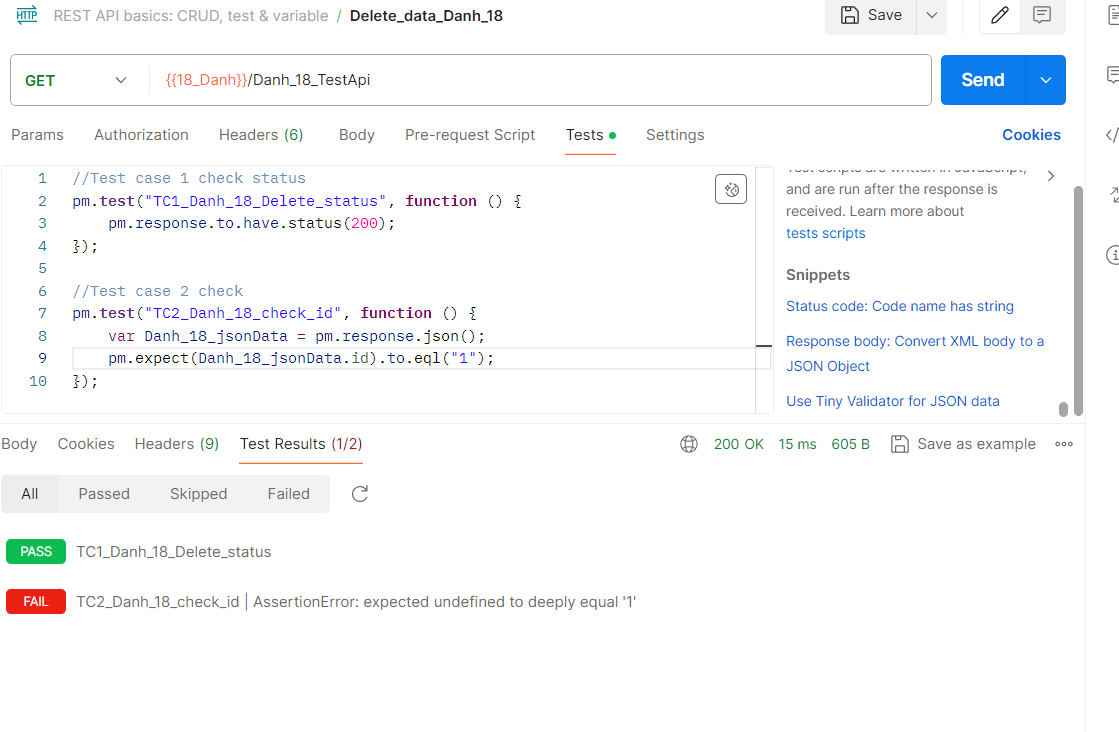
pm.**test**("TC2\_Danh\_18\_checkName", **function**(){

    var Danh\_18\_jsonData **=** pm.response.json();

    pm.expect(Danh\_18\_jsonData.name).to.eql("18\_Lê Thanh Danh");

});

1. Phương thức DELETE



//Test case 1 check status

pm.**test**("TC1\_Danh\_18\_Delete\_status", **function** () {

    pm.response.to.have.status(200);

});

//Test case 2 check

pm.**test**("TC2\_Danh\_18\_check\_id", **function** () {

    var Danh\_18\_jsonData **=** pm.response.json();

    pm.expect(Danh\_18\_jsonData.id).to.eql("1");

});