TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH

**KHOA CÔNG NGHỆ THÔNG TIN**

---🙡 🕮 🙣---



**BÁO CÁO ĐỀ TÀI MÔN HỌC**

**KIỂM THỬ PHẦN MỀM**

ĐỀ TÀI: **KIỂM THỬ WEBSITE MSISTORE**

|  |  |
| --- | --- |
| **Sinh viên thực hiện:** | **Nguyễn Thế Anh** |
| **Mã số sinh viên:** | **2151013002** |
| **Lớp:** | **DH21CS02** |
| **Giảng viên hướng dẫn:** | **Th.s Nguyễn Thị Ngọc Thanh** |

**Tháng 01 năm 2024**

Mục Lục

[LỜI MỞ ĐẦU 4](#_Toc156206279)

[CHƯƠNG 1. KIỂM THỬ TỰ ĐỘNG 5](#_Toc156206280)

[**1.1 TỔNG QUAN KIỂM THỬ TỰ ĐỘNG** 5](#_Toc156206281)

[**1.2. SELENIUM WEBDRIVER** 5](#_Toc156206282)

[**1.2.1 Giới thiệu Selenium webdriver** 5](#_Toc156206283)

[**1.2.2 Quy trình khi làm việc với Selenium Web** 6](#_Toc156206284)

[**1.2.3 Cài đặt Selenium WebDriver với ngôn ngữ C#** 6](#_Toc156206285)

[**1.3 POSTMAN** 8](#_Toc156206286)

[**1.3.1 Giới thiệt postman** 8](#_Toc156206287)

[**1.3.2 Làm quen với công cụ postman** 8](#_Toc156206288)

[Cửa sổ postman có 3 phần chính gồm: Header bar, Slide bar, Builder (Hình 1.5) 8](#_Toc156206289)

[CHƯƠNG 2. ĐẶC TẢ PHẦN MỀM 9](#_Toc156206290)

[**2.1 GIỚI THIỆU PHẦN MỀM** 9](#_Toc156206291)

[**2.2 ĐẶC TẢ PHẦN MỀM** 10](#_Toc156206292)

[**2.3 ĐẶC TẢ USE CASE** 10](#_Toc156206293)

[**2.3.1 Chức năng đăng nhập** 10](#_Toc156206294)

[**2.3.2 Chức năng đăng ký** 11](#_Toc156206295)

[**2.3.3 Chức năng thay đổi mật khẩu** 12](#_Toc156206296)

[CHƯƠNG 3. TRIỂN KHAI VIẾT CÁC TEST CASE VỚI SELENIUM WEBDRIVER 13](#_Toc156206297)

[**3.1 CHUẨN BỊ** 13](#_Toc156206298)

[**3.2 TRIỂN KHAI VIẾT TEST CASE** 13](#_Toc156206299)

[**3.2.1 Test case 01: Login** 13](#_Toc156206300)

[**3.2.2 Test case 02: Register** 15](#_Toc156206301)

[**3.2.3 Test case 03: Change Password** 17](#_Toc156206302)

[CHƯƠNG 4. TRIỂN KHAI VIẾT CACS TEST CASE VỚI POSTMAN 20](#_Toc156206303)

[**4.1 CHUẨN BỊ** 20](#_Toc156206304)

[**4.2 TRIỂN KHAI VIẾT TEST CASE** 22](#_Toc156206305)

[**4.2.1 TestCase 01: Login** 22](#_Toc156206306)

[**4.2.2 Test case 02: Register** 24](#_Toc156206307)

[**4.2.3 Test case 03: Pagination** 26](#_Toc156206308)

[**4.2.4 Test case 04: Filter by category** 28](#_Toc156206309)

[**4.2.5 Test case 05: Filter by price** 29](#_Toc156206310)

[**4.2.6 Test case 06: Performance** 31](#_Toc156206311)

[**4.2.7 Test case 07: Update user info** 32](#_Toc156206312)

[**4.2.7 Test case 07: Delete User** 33](#_Toc156206313)

[KẾT LUẬN 35](#_Toc156206314)

# **LỜI MỞ ĐẦU**

Ngày nay công nghệ thông tin ngày càng phát triển nhanh chóng, kéo theo

đó là hệ thống mạng và các phần mềm cũng gia tăng cả về số lượng theo quy mô

rộng và cả về chất lượng phần mềm theo chiều sâu. Nhưng cũng từ đó đã nảy

sinh ra nhiều vấn đề về lỗi hỏng hóc phần mềm không đáng có gây ra ảnh hưởng

nghiêm trọng đến xã hội, kinh tế,.... Những lỗi này có thể do tự bản thân phần

mềm bị hỏng do không được kiểm duyệt kỹ lưỡng trước khi đưa vào cho người

dùng cuối hay cũng có thể do có người cố tình phá hoại nhằm đánh cắp thông tin

cá nhân như mã số tài khoản ngân hàng, số điện thoại, danh bạ, tin nhắn,...

Những vấn đề nan giải và cấp thiết này càng có xu hướng mở rộng trong các

năm gần đây, điển hình như sự cố máy tính Y2K năm 2000 làm tê liệt nhiều hệ

thống máy tính lớn hay như càng có nhiều loại virus phá hoại mới xuất hiện, tấn

công vào các lỗ hổng bảo mật phần mềm làm tê liệt nhiều hệ thống phần mềm

và phần cứng. Từ đó, có thể dễ dàng nhận ra là mặc dù phần mềm phát triền

ngày càng phức tạp nhưng vấn đề về chất lượng vẫn là một dấu hỏi lớn cần xem

xét cẩn thận.

Do đó yêu cầu đặt ra là cần có công tác kiểm thử phần mềm thật kỹ lưỡng

nhằm ngăn chặn các lỗi hay hỏng hóc còn tiềm tàng bên trong phần mềm mà ta

chưa kịp nhận ra. Tuy nhiên vì phần mềm ngày càng lớn, hàng nghìn module, có

thể do cả một công ty hàng nghìn người phát triền vì vậy để kiểm thử được một

phần mềm lớn như vậy sẽ tốn rất nhiều công sức và thời gian nếu làm thủ công,

chưa kể đến chất lượng kiểm thử sẽ không cao và thật chính xác phù hợp cho

yêu cầu. Theo nhiều tính toán thì công việc kiểm thử đóng vai trò hết sức quan

trọng trong quy trình phát triền phần mềm, nó đóng góp tới 40% tổng toàn bộ

chi phí cho việc sản xuất phần mềm. Vì vậy cần có các hệ thống kiểm thử phần

mềm một cách tự động cho phép ta thực hiện được các công việc một cách

nhanh chóng và độ an toàn, chính xác cao nhất có thể. Và đó chính là lý do em lựa chọn website “Msistore – website bán máy tính online” để nghiên cứu, tìm hiểu và đề ra các gỉải pháp mới để cải tiến các quy trình kiểm thử như hiện nay sao cho có năng xuất cao nhất.

# **CHƯƠNG 1. KIỂM THỬ TỰ ĐỘNG**

## **1.1 TỔNG QUAN KIỂM THỬ TỰ ĐỘNG**

Kiểm thử tự động (Automation Test ) là kỹ thuật sử dụng các công cụ, phần mềm để thực thi các test case một cách tự động sau đó đưa ra kết quả kiểm thử một các hiệu quả và nhanh chóng.

Mục đích chính của kiểm thử tự động giúp giảm thời gian, chi phí, tăng tính tin cậy, nâng cao hiệu suất công việc, đồng thời giảm sự nhàm chán của kiểm thử viên.

Để xây dựng được một chương trình kiểm thử tự động, ngoài việc yêu cầu có kiến thức về kiểm thử phần mềm, các kỹ thuật trong kiểm thử phần mềm, kiểm thử viên cần nắm ít nhất một ngôn ngữ lập trình như Java, Python, Javascript, biết ít nhất một framework kiểm thử tự động như Selenium, JUnit, TestNG, v.v., nắm rõ các kiến thức HTML, CSS selector, XPath để tương tác với các thành phần giao diện.

## **1.2. SELENIUM WEBDRIVER**

### **1.2.1 Giới thiệu Selenium webdriver**

Selenium WebDriver ( gọi tắt là WebDriver) là một automation framework dành cho web, với WebDriver bạn có thể thực hiện các phiên kiểm thử của mình trên các trình duyệt khác nhau mà không cần chỉ giới hạn trên Firefox hay Chrome

WebDriver hỗ trợ nhiều ngôn ngữ lập trình khác nhau như Java, .Net, PHP, Python, Perl, Ruby. Không nhất thiết bạn phải am hiểu tất cả các ngôn ngữ trên nhưng để sử dụng WebDriver 1 cách hiệu quả hơn, bạn nên biết ít nhất một trong số những ngôn ngữ trên.

### **1.2.2 Quy trình khi làm việc với Selenium Web**

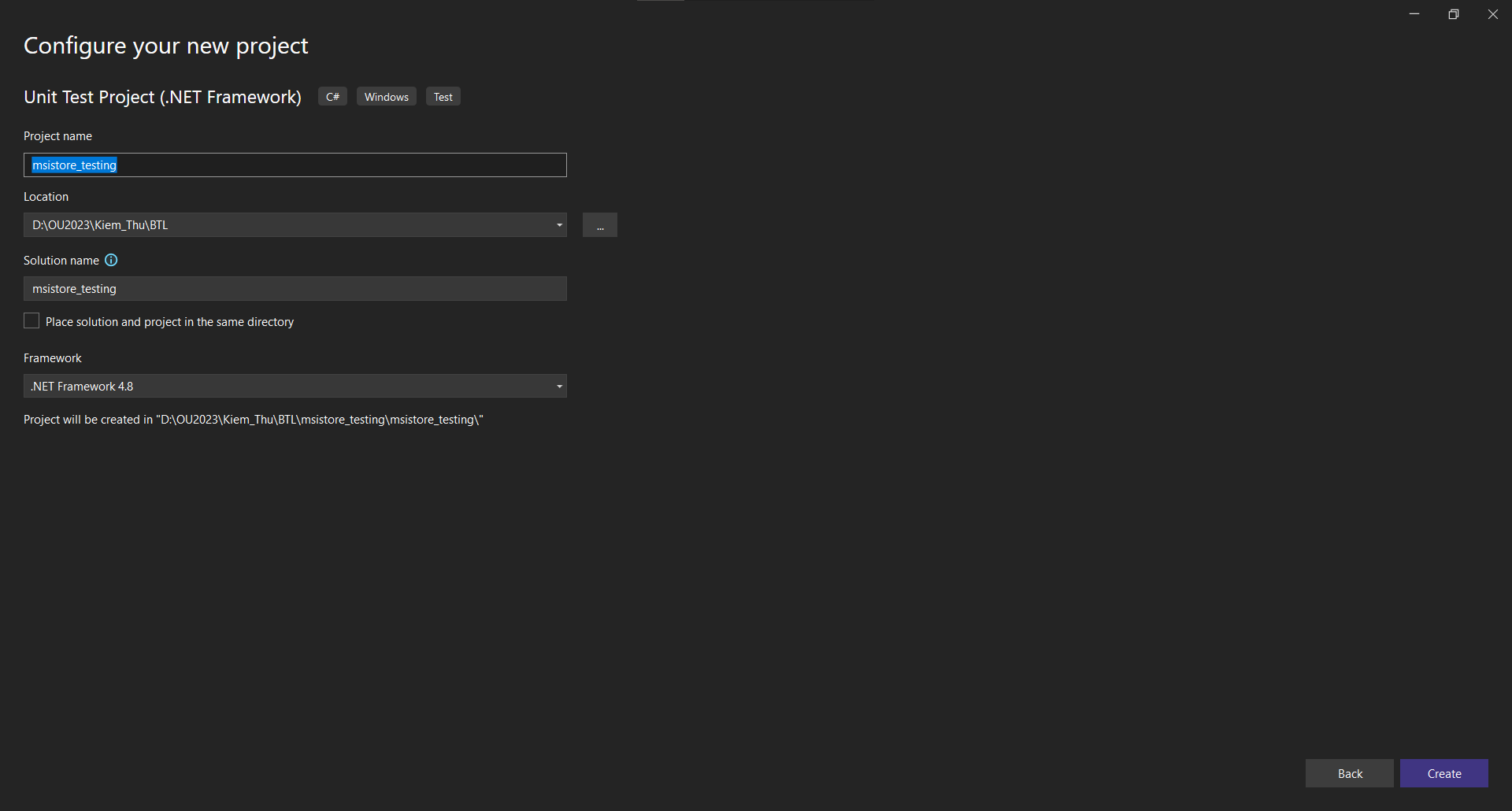
Kiểm thử viên viết các kịch bản kiểm thử và sử dụng các API của Selenium Webdriver trên các ngôn ngữ lập trình tương ứng. Những kịch bản này sẽ chỉ định các sự tương tác với trình duyệt như Chrome, Firefox, Safari để tương tác với ứng dụng web.

Sau khi thực thi xong, Webdriver sẽ trả về kết quả tương ứng.

### **1.2.3 Cài đặt Selenium WebDriver với ngôn ngữ C#**

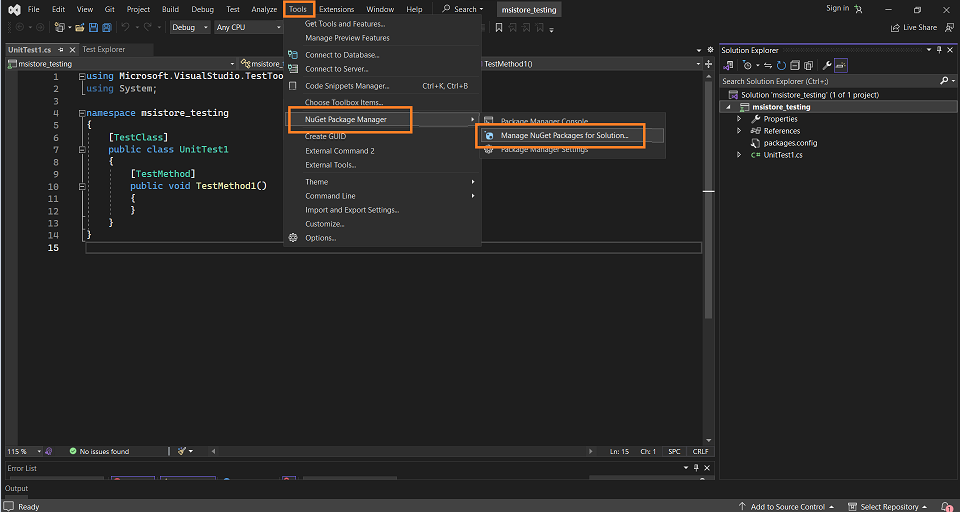
**‣** Bước 1. Tạo một project mới trong Visual Studio 2022

Tải Visual Studio 2022 tại: https://visualstudio.microsoft.com/downloads/



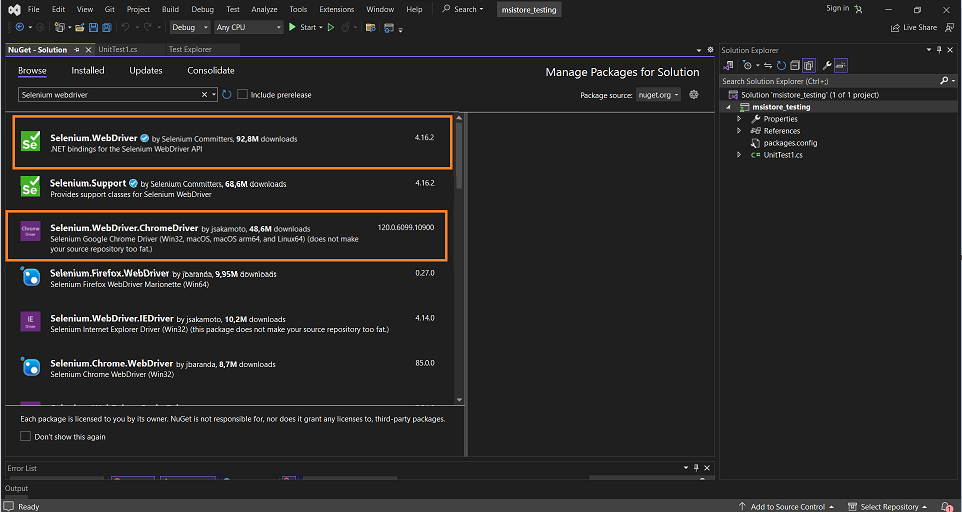
*Hình 1.1 Tạo project trong visual studio 2022*

‣ Bước 2: Tại giao diện chính chọn Tool – Nuget Package Manage – Manage Nuget package solution



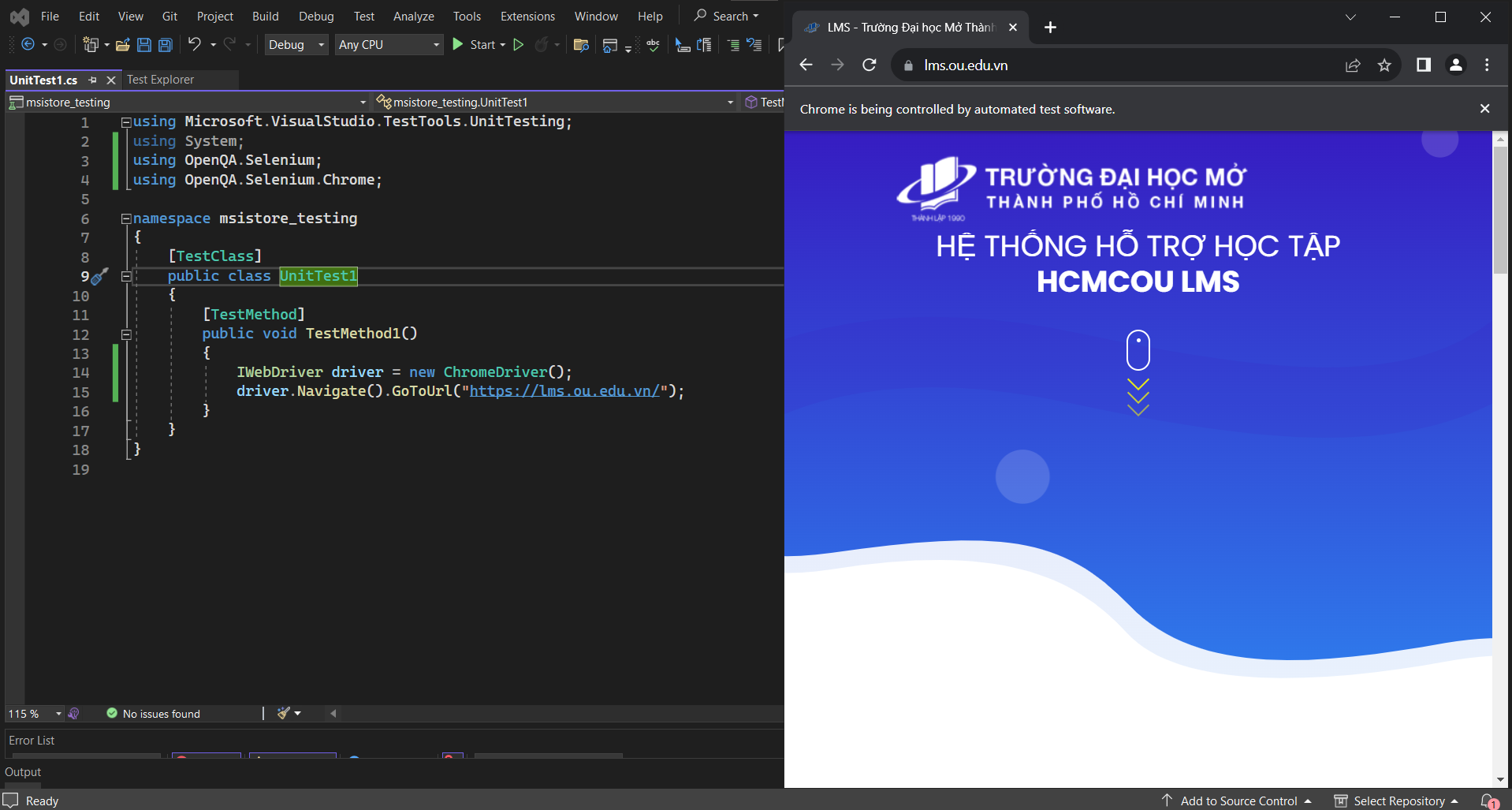
*Hình1.2 Cách vào Nuget package solution*

‣ Bước 3: Tại mục Browse nhập vào ô tìm kiếm với từ khoá “selenium webdriver” và tải hai package có tên “Selenium.Webdiver” và “Selenium.Webdriver.ChromeDriver”



*Hình1.3 Những package cần tải để làm việc với Selenium webdriver*

‣ Bước 4: Chạy thử



*Hình1.4: Chạy đoạn code đầu tiên với Selenium webdriver*

## **1.3 POSTMAN**

### **1.3.1 Giới thiệt postman**

Postman là một công cụ phổ biến được sử dụng rộng rãi trong trong kiểm thử API. Postman hỗ trợ trên nhiều nền tảng hệ điều hành khác nhau như Mac, Windows (32-bit/64-bit), Linux (32-bit/64-bit).

Tải xuống Postman tại sau: <https://www.postman.com/downloads/>

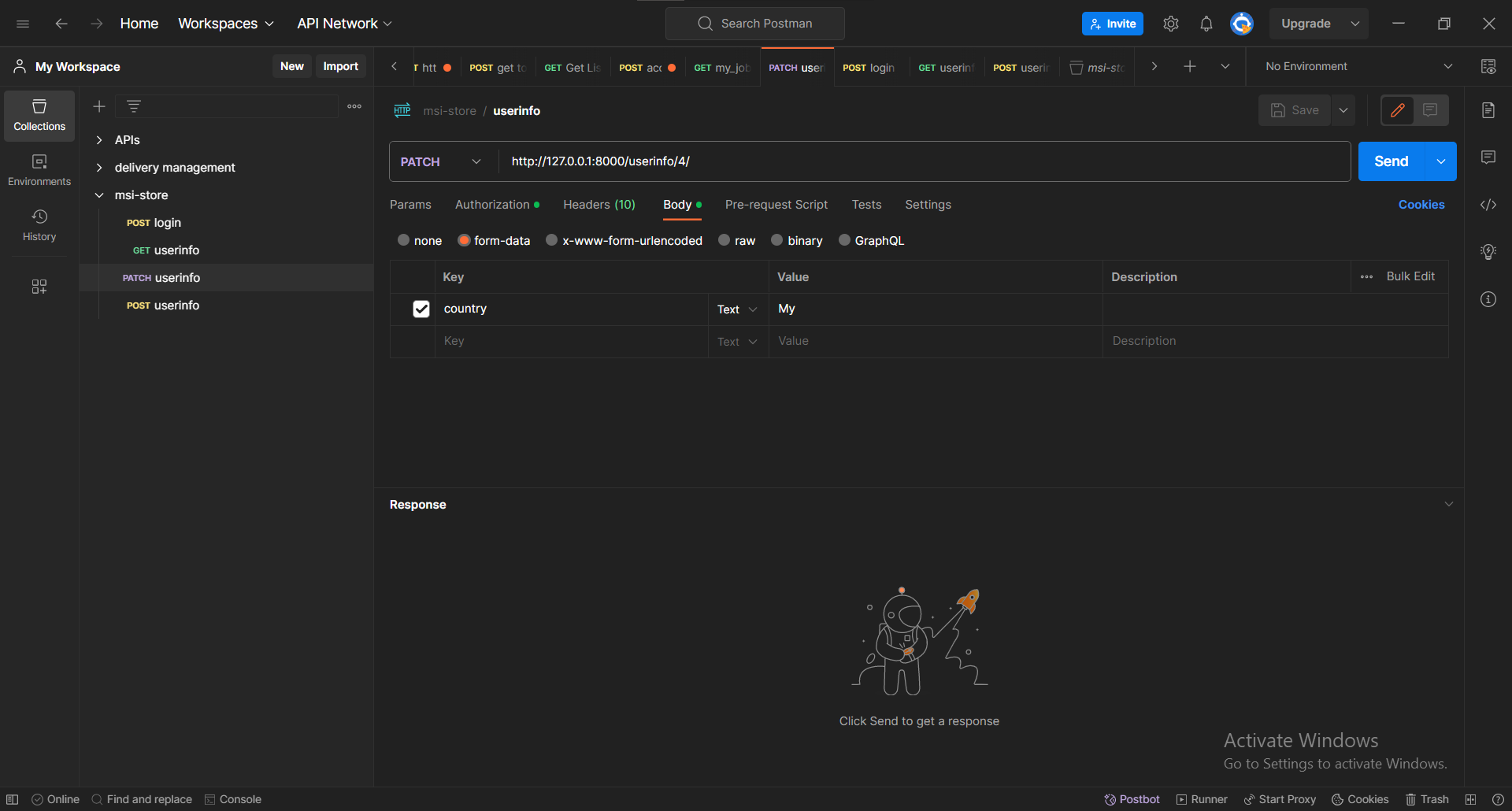
### **1.3.2 Làm quen với công cụ postman**

### Cửa sổ postman có 3 phần chính gồm: Header bar, Slide bar, Builder (Hình 1.5)

**Header bar** chứa các nút chức năng để tạo và nhập các request (yêu cầu).

**Slider bar** có thể truy cập vào các mục History, Collections và APIS. Trong đó history lưu lại những request đã được thực hiện, collections chứa các collection – một danh sách request được gom nhóm lại.

**Builder** xây dựng và làm việc với các request, gồm một số thành phần quan trọng như: HTTP request, Request, Params, Headers, Body.



*Hình 1.5 Giao diện công cụ postman*

# **CHƯƠNG 2. ĐẶC TẢ PHẦN MỀM**

## **2.1 GIỚI THIỆU PHẦN MỀM**

Tên phần mềm: Msistore – website bán máy tính online.

Ngôn ngữ viết:

FrontEnd: ReactJs

Đã được host tại: https://msistore.vercel.app/

Backend: Django

Đã được host tại: https://ntheanh.pythonanywhere.com/

Database: Mysql server

Chức năng chính của phần mềm:

Đăng ký: Cho phép người dùng đăng ký tài khoản cá nhân bao gồm các thông tin cá nhân, địa chỉ, ảnh đại diện …

Đăng nhập: đăng nhập vào tài khoản đã đăng ký để tiến hành đặt mua hàng.

Quản lý: thêm, sửa, xoá người dùng, sản phẩm, đơn hàng.

## **2.2 ĐẶC TẢ PHẦN MỀM**

Khi khách hàng quyết định tham gia vào trải nghiệm mua sắm trực tuyến, họ sẽ được cấp một giỏ hàng cá nhân để tự do thêm vào đó các sản phẩm mong muốn. Sử dụng chức năng tìm kiếm trên trang web giúp khách hàng nhanh chóng định vị và thêm vào giỏ hàng những sản phẩm ưa thích.

Khách hàng có khả năng linh hoạt cập nhật giỏ hàng bằng cách loại bỏ các mặt hàng không mong muốn. Khi quyết định thanh toán, họ sẽ được yêu cầu nhập chi tiết cá nhân như tên, địa chỉ email, số điện thoại, và nếu chưa đăng ký, họ có thể tạo một tài khoản mới với username và password. Đối với những khách hàng đã đăng ký, việc đăng nhập sẽ giúp quá trình này trở nên thuận tiện hơn.

Sau khi hoàn tất quá trình đặt hàng, khách hàng có thể lựa chọn các phương thức thanh toán trực tuyến trên trang web. Họ cũng có thể thêm sản phẩm mới vào đơn hàng hoặc theo dõi tình trạng đơn hàng hiện tại.

Nếu có bất kỳ vấn đề nào, khách hàng có thể gửi đơn khiếu nại hoặc phản hồi thông qua chức năng liên hệ khách hàng. Điều này giúp tạo ra một trải nghiệm mua sắm trực tuyến toàn diện và linh hoạt, đồng thời nâng cao sự hài lòng của khách hàng với sản phẩm và dịch vụ.

## **2.3 ĐẶC TẢ USE CASE**

### **2.3.1 Chức năng đăng nhập**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã use case** | UC001 | **Nội dung** | Đăng nhập |
| **Tác nhân** | Khách hàng | | |
| **Mô tả** | Tác nhân đăng nhập vào hệ thống để sử dụng các chức năng hệ thống. | | |
| **Tiền điều kiện** | Tác nhân chưa đăng nhập vào hệ thống. | | |
| **Hậu điều kiện** | Tác nhân được chuyển đến trang phù hợp. | | |
| **Luồng hoạt động chính** | 1. Khách vào trang đăng nhập. 2. Khách nhập username, password. 3. Nhấn nút đăng nhập. 4. Hệ thống kiểm tra thông tin đăng nhập. 5. Nếu thông tin hợp lệ hệ thống sẽ chuyển đến trang chủ, ngược lại sẽ ở lại trang đăng nhập. | | |
| **Luồng hoạt động thay thế** | 4. a) Hệ thống yêu cầu nhập đầy đủ thông tin.  4. b) Hệ thống thông báo thông tin nhập sai định dạng.  4. c) Hệ thống thông báo không tìm thấy tài khoản. | | |

**Bảng 2.1 Đặc tả chức năng đăng nhập**

\* Dữ liệu đầu vào gồm các các trường dữ liệu sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Mô tả | Bắt buộc ? | Điều kiện hợp lệ | Ví dụ |
| 1 | Username | Tài khoản người dùng | Có | Từ 3-16 kí tự và không chứa kí tự đặc biệt | Theanhmgt28 |
| 2 | Password | Mật khẩu của tương ứng với tài khoản | Có | Từ 8-20 kí tự và chứa ít nhất một chữ số | Passw0rd |

**Bảng 2.2 Bảng dữ liệu đầu vào cho chức năng đăng nhập**

### **2.3.2 Chức năng đăng ký**

|  |  |  |  |
| --- | --- | --- | --- |
| Mã use case | UC002 | Nôi dung | Đăng ký |
| Tác nhân | Khách hàng | | |
| Mô tả | Tác nhân đăng ký tài khoản để có thể sử dụng các chức năng hệ thống. | | |
| Tiền điều kiện | Tác nhân đã đăng xuất khỏi hệ thống trước đó. | | |
| Hậu điều kiện | Tác nhân được chuyển đến trang phù hợp. | | |
| Luồng hoạt động chính | 1. Khách vào trang đăng ký 2. Nhập các thông tin cá nhân như họ tên, email, username, password. 3. Nhấn vào “chọn tệp” để tải ảnh đại diện. 4. Hệ thống kiểm tra thông tin đăng ký. 5. Nếu thông tin hợp lệ hệ thống sẽ chuyển đến trang chủ, ngược lại sẽ ở lại trang đăng ký. | | |
| Luồng hoạt động thay thế | 5. a) Hệ thống yêu cầu nhập đầy đủ thông tin.  5. b) Hệ thống thông báo thông tin nhập sai định dạng. | | |

**Bảng 2.3 Đặc tả chức năng đăng ký**

\* Dữ liệu đầu vào gồm các các trường dữ liệu sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Mô tả | Bắt buộc? | Điều kiện hợp lệ | Ví dụ |
| 1 | First name | Tên | Có | Từ 3-20 kí tự | Thế Anh |
| 2 | Last name | Họ | Có | Từ 3-20 kí tự | Nguyễn |
| 3 | Username | Tài khoản | Có | Từ 3-16 kí tự và không chứa kí tự đặc biệt | Theanh28 |
| 4 | Password | Mật khẩu | Có | Từ 8-20 kí tự và chứa ít nhất một chữ số | Passw0rd |
| 5 | Email | Email | Có | Đúng định dạng email | Abc@gmail.com |
| 6 | Avatar | Ảnh đại diện | Có | Thuộc một trong cái định dạng file:  .jpg .jpeg .png .gif | Photo.jpg  Photo.png |

### **2.3.3 Chức năng thay đổi mật khẩu**

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC001 | Nội dung | Đổi mật khẩu |
| Tác nhân | Khách hàng | | |
| Mô tả | Tác nhân thay cung cấp mật khẩu cũ và mới để tiến hành thay đổi mật khẩu | | |
| Tiền điều kiện | Tác nhân đăng nhập thành công vào hệ thống | | |
| Hậu điều kiện | Tác nhân nhận thông báo tương ứng | | |
| Luồng hoạt động chính | 1. Khách vào trang đăng nhập và tiến hành đăng nhập 2. Truy cập vào mục thông tin cá nhân 3. Lựa chọn chức năng thay đổi mật khẩu 4. Nhập mật khẩu cũ và mật khẩu mới vào ô input 5. Hệ thống tiến hành kiểm tra thông tin được nhập 6. Nếu mật khẩu cũ chính xác và mật khẩu mới được nhập đúng định dạng, hệ thống thay đổi mật khẩu và thông báo thay đổi thành công. | | |
| Luồng hoạt động thay thế | 5. a) Hệ thống yêu cầu nhập đầy đủ thông tin.  5. b) Hệ thống thông báo thông tin nhập sai định dạng. | | |

\* Dữ liệu đầu vào gồm các các trường dữ liệu sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Mô tả | Bắt buộc? | Điều kiện hợp lệ | Ví dụ |
| 1 | Current Password | Mật khẩu cũ | Có | Từ 8-20 kí tự và chứa ít nhất một chữ số | Passw0rd |
| 2 | New Password | Mật khẩu mới | Có | Từ 8-20 kí tự và chứa ít nhất một chữ số | Passw0rd |

# **CHƯƠNG 3. TRIỂN KHAI VIẾT CÁC TEST CASE VỚI SELENIUM WEBDRIVER**

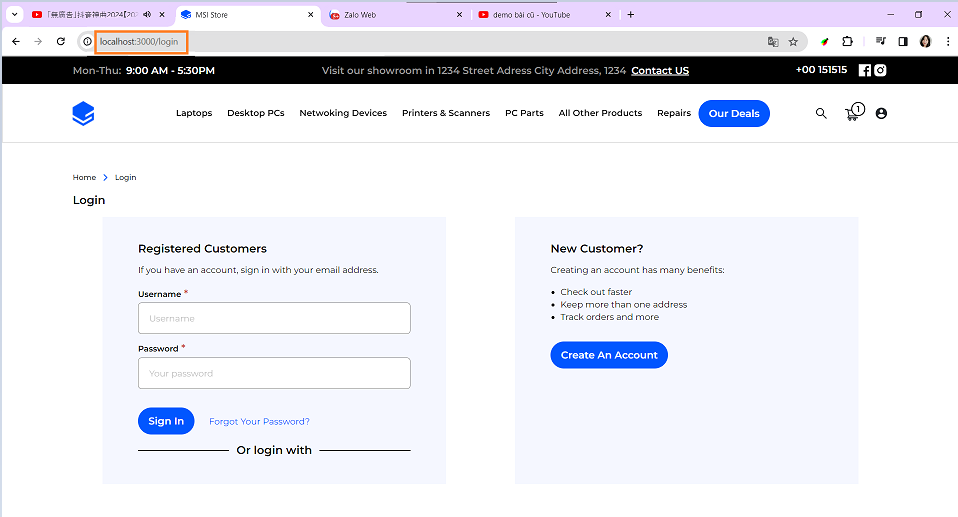
## **3.1 CHUẨN BỊ**

Tạo một project có tên msistore\_testing với những cách đã được trình bày ở mục **1.2.3**

## **3.2 TRIỂN KHAI VIẾT TEST CASE**

### **3.2.1 Test case 01: Login**

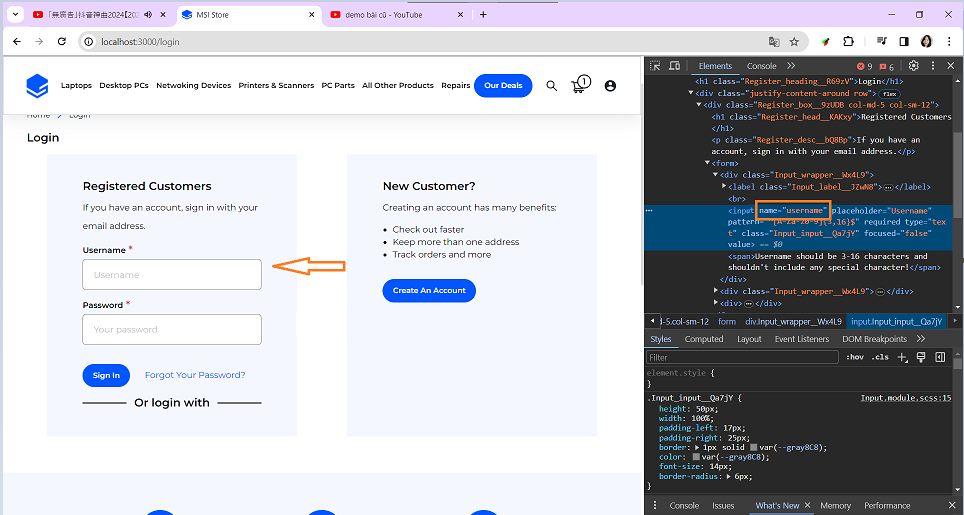
‣ Bước 1: Lấy Url của trang login



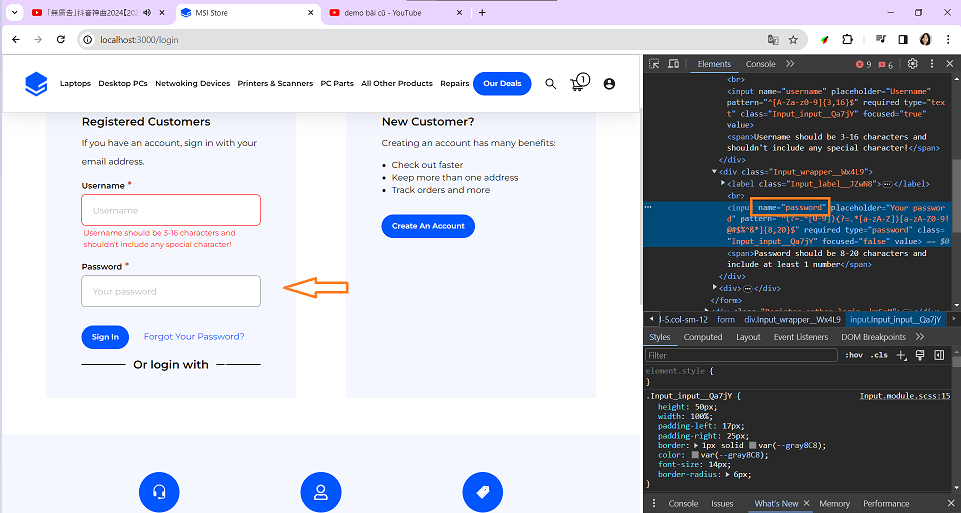
*Hình3.1 Đường dẫn của trang đăng nhập*

‣ Bước 2: Nhập username và password

▪ Cần lấy thuộc tính “name” của hai input này để thực hiện sendKey gửi dữ liệu vào hai ô input.



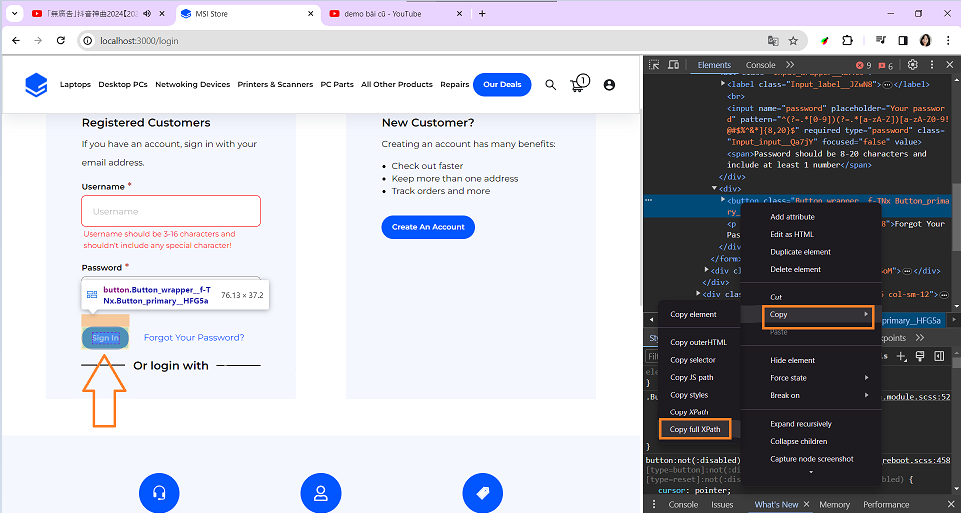
*Hình3.2 Thuộc tính name của ô input username*



*Hình3.3 Thuộc tính name của thẻ ô input password*

‣ Bước 3: Click vào nút Sign In

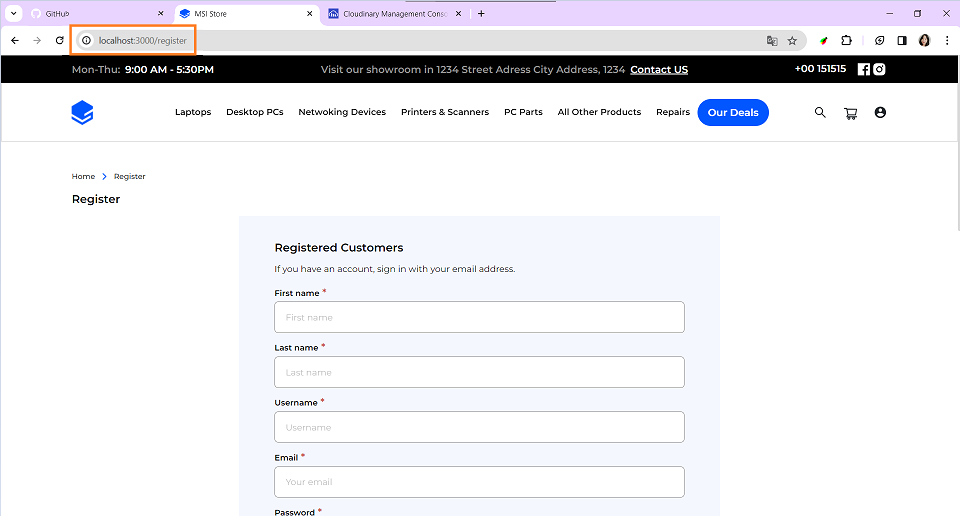
▪ Cần bắt Xpath của button Sign in để thực hiện click.



*Hình3.4 Xpath của button Sign In*

### **3.2.2 Test case 02: Register**

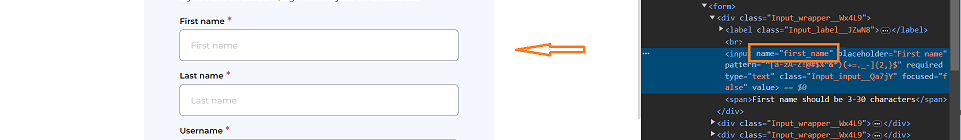
‣ Bước 1: Lấy Url của trang register



*Hình 3.5 Đường dẫn của trang đăng ký*

‣ Bước 2: Nhập thông tin tài khoản muốn đăng ký.

▪ Cần lấy thuộc tính “name” của những ô input: first name, last name, email, username, password, confirm password để thực hiện sendKey gửi dữ liệu vào các ô input này.



*Hình 3.6 Thuộc tính name của ô first name*



*Hình 3.7 Thuộc tính name của ô last name*



*Hình 3.8 Thuộc tính name của ô username*



*Hình 3.9 Thuộc tính name của ô email*



*Hình 3.10 Thuộc tính name của ô password*



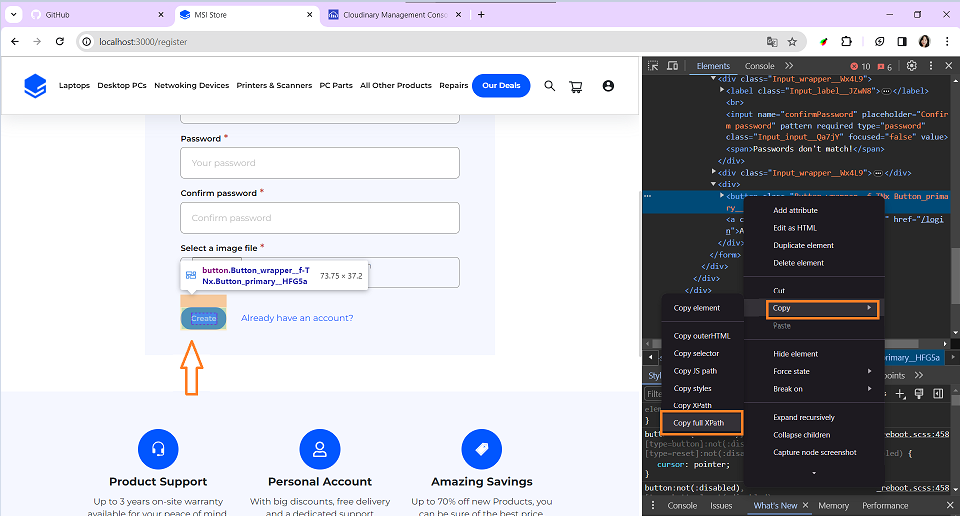
*Hình 3.11 Thuộc tính name của ô confirm password*



*Hình 3.12 Thuộc tính name của ô select file*

‣ Bước 3: Click vào nút Register

▪ Cần bắt Xpath của button Create để thực hiện click.

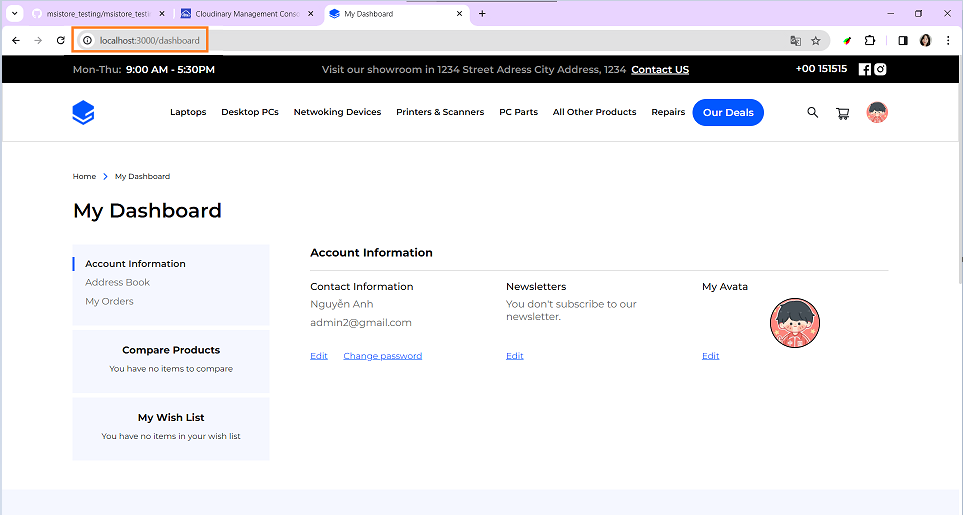


*Hình3.12 Xpath của nút Create*

### **3.2.3 Test case 03: Change Password**

‣ Bước 1: Lấy url của trang web

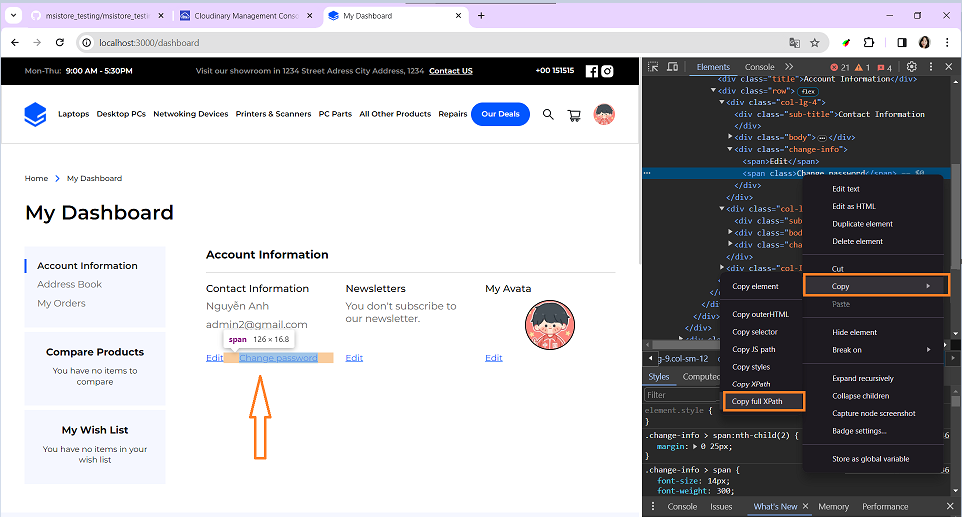
▪ Cần đăng nhập vào hệ thống trước, rồi đến trang thông tin cá nhân.



*Hình 3.13 Đường dẫn của trang thông tin cá nhân*

‣ Bước 2: Click vào Change password

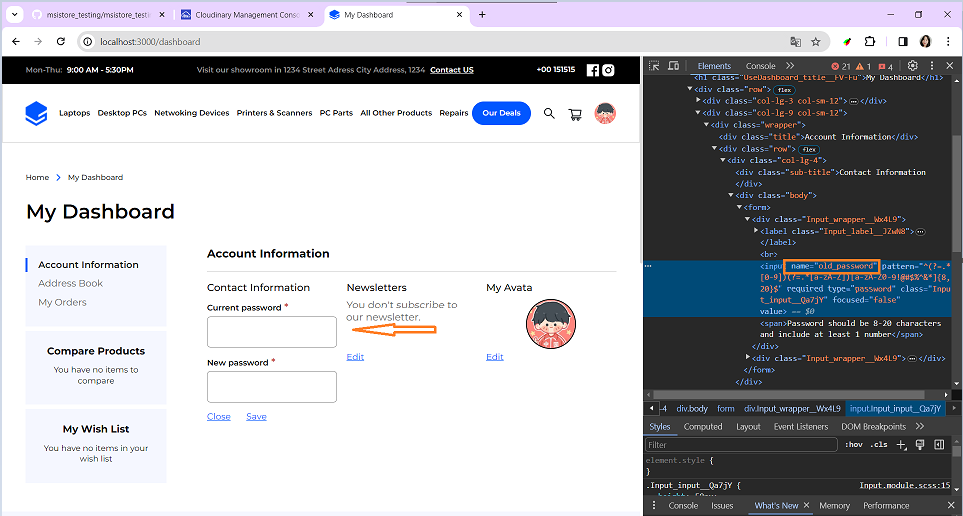
▪ Cần bắt xpath vào thẻ span chứa chữ “Change password” để thực hiện hành động click



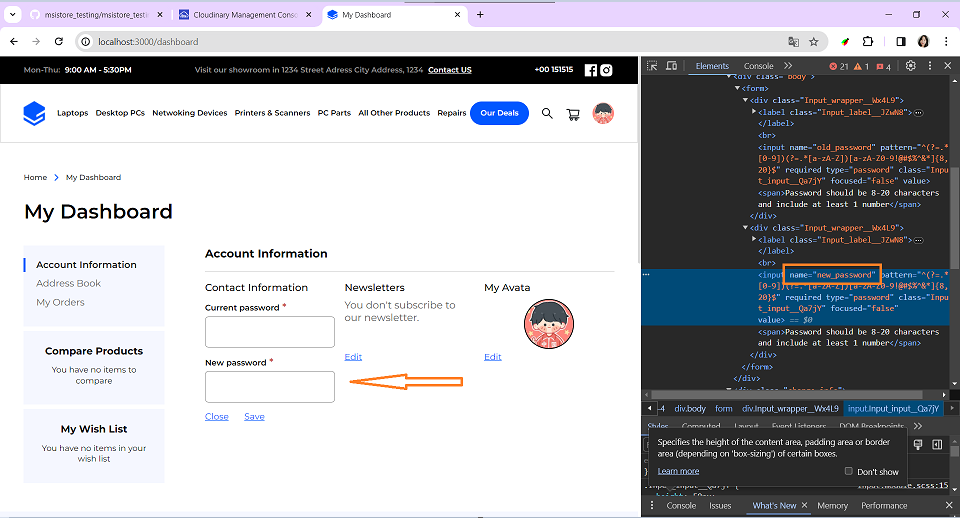
*Hình3.14 Xpath của thẻ chứa nội dung “Change password”*

‣ Bước 3: Điền current password và new password

▪ Cần bắt thẻ input có thuộc tính name để thực hiện hành động sendkey.



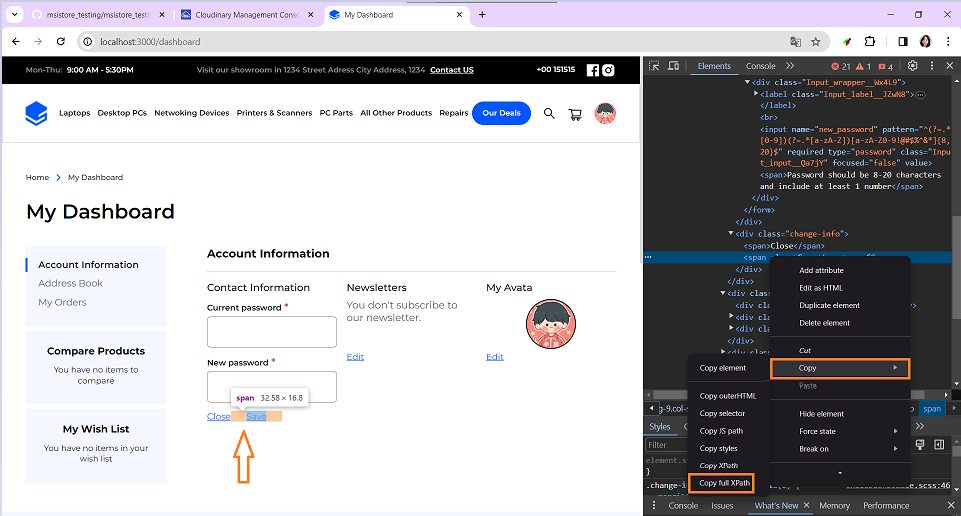
*Hình 3.15 Thuộc tính name của ô Current password*



*Hình 3.16 Thuộc tính name của ô New password*

‣ Bước 4: Click vào Save

▪ Cần bắt xpath vào thẻ span chứa chữ “Change password” để thực hiện hành động click

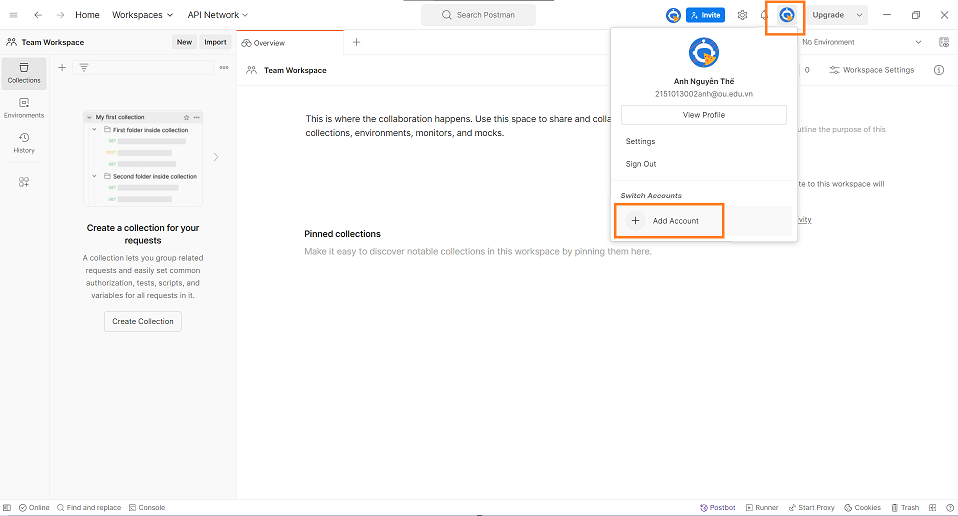


*Hình3.17 Xpath của thẻ chứa nội dung “Save”*

# **CHƯƠNG 4. TRIỂN KHAI VIẾT CACS TEST CASE VỚI POSTMAN**

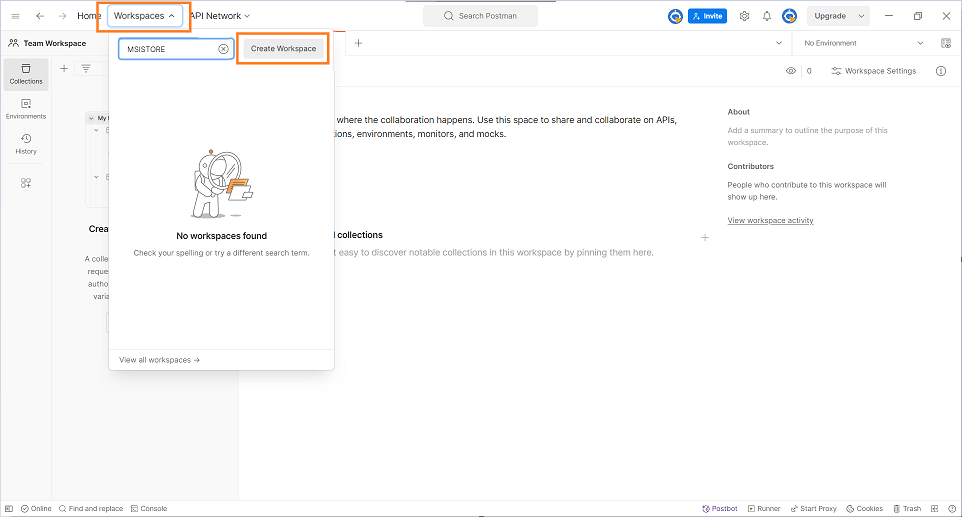
## **4.1 CHUẨN BỊ**

‣ Bước 1: Đăng nhập vào postman



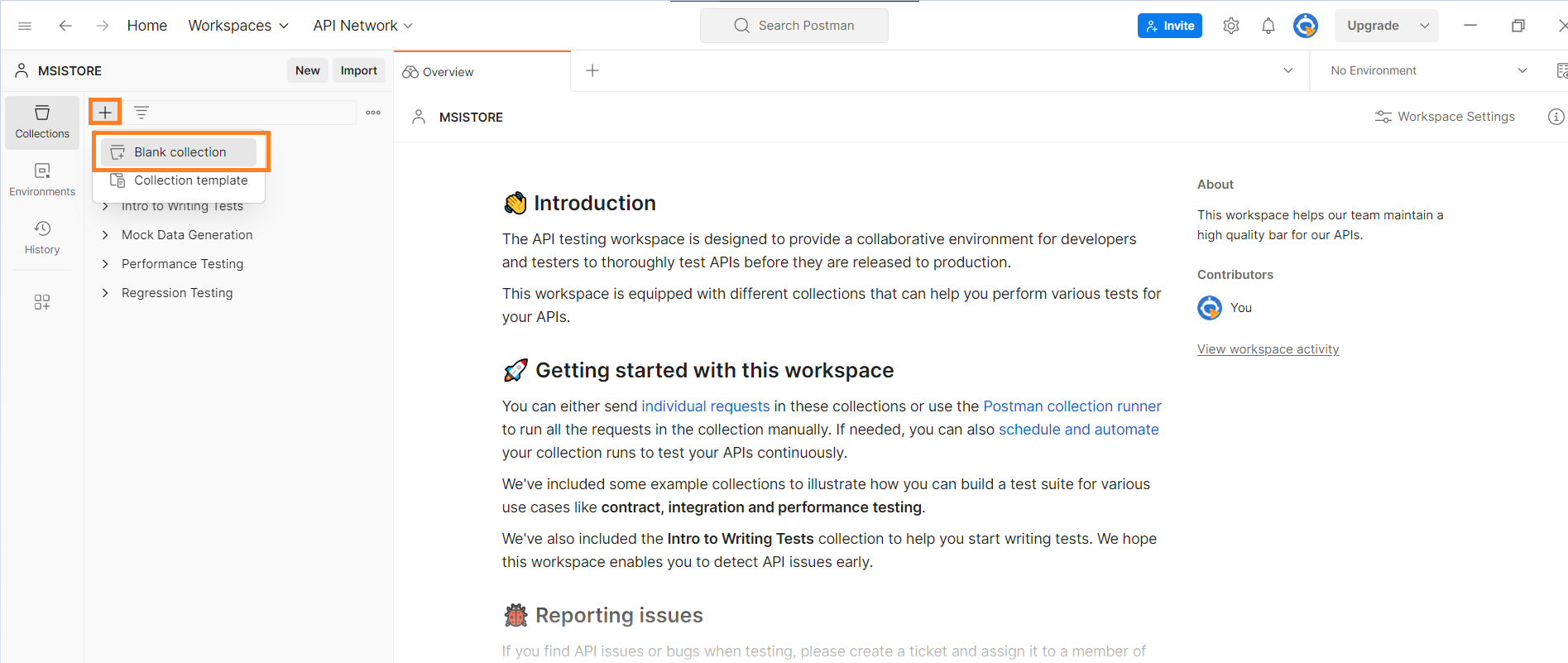
*Hình 4.1 Màn hình chính ứng dụng Postman*

‣ Bước 3: Tạo một workspace mới



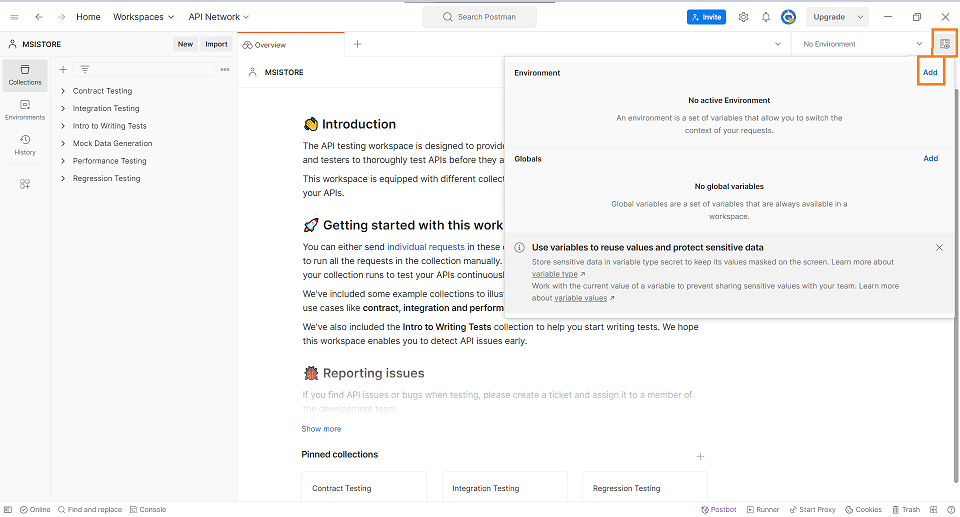
*Hình 4.2 Tạo workspace mới*

‣ Bước 4: Tạo một collection mới

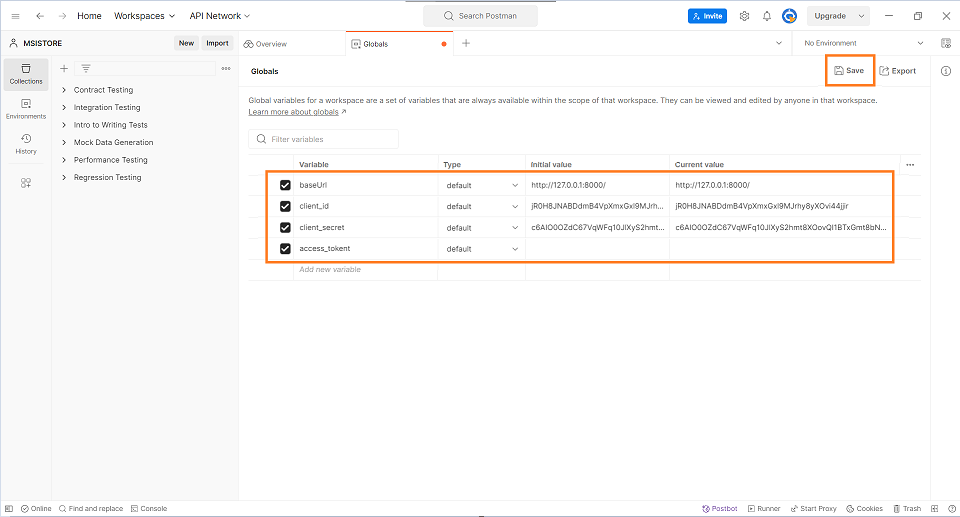


*Hình 4.3 Tạo collection mới*

‣ Bước 5: Thiết lập biến môi trường



*Hình 4.4 Thiết lập biến môi trường*



*Hình 4.5 Tạo và ngán giá trị cho các biến môi trường*

## **4.2 TRIỂN KHAI VIẾT TEST CASE**

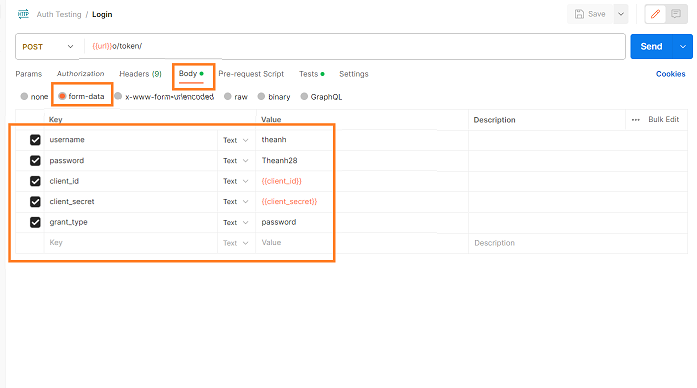
### **4.2.1 TestCase 01: Login**

‣ Bước 1: Nhập url có sử dụng biến môi trường đã thêm trước đó và chọn phương thức là POST



*Hình 4.6 Nhập địa chỉ API và chọn phương thức gửi lên*

‣ Bước 2: Chọn kiểu dữ liệu đính kèm và điều đầy đủ những trường cần thiết ở phần **Body**



*Hình 4.7 Nhập giá trị gửi kèm trong phần body*

‣ Bước 3: Viết các test case trong phần **Tests**



*Hình 4.8 Viết các test script trong phần Tests*

▪ Test case có tên “TC-4.2.1.1 Login Successfully” để xem liệu có login thành công hay không bằng cách kiểm tra:

* status có là 200 không?
* header có thuộc tính ‘Content-Type’ là ‘application/json’ không?

▪ Test case có tên “TC-2.2.1.2 An access token is returned” để kiểm tra có access token được trả về sau khi đã đăng nhập hay không bằng cách kiểm tra xem:

* có thuộc tính ‘access\_token’ trong response không?
* thuộc tính ‘access\_token’ có kiểu dữ liệu là string không?
* thuôc tính ‘access\_token’ không được là chuỗi rỗng?

Sau đó set giá trị token vào biến ‘access\_token’ để dùng cho những test case sau.

‣ Bước 4: Bấm Send và xem kết quả



*Hình 4.9 Kết quả sau khi bấm nút Send*

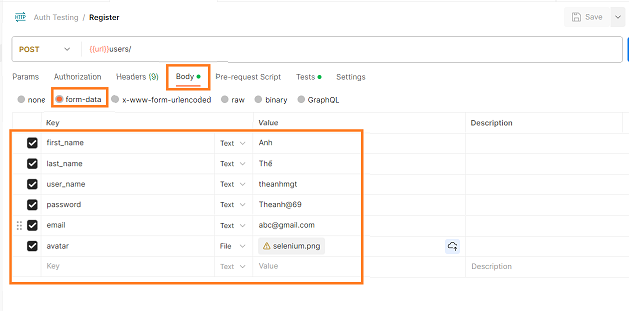
### **4.2.2 Test case 02: Register**

‣ Bước 1: Nhập url có sử dụng biến môi trường đã thêm trước đó và chọn phương thức là POST



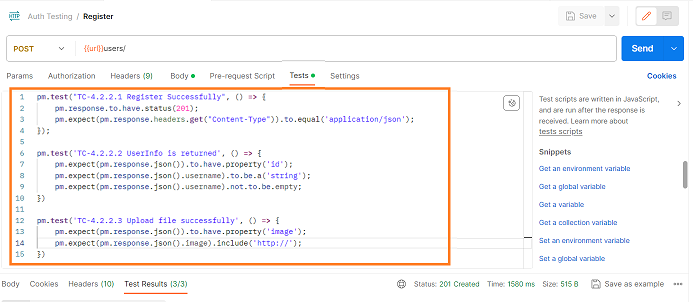
*Hình 4.10 Nhập địa chỉ API và chọn phương thức gửi lên*

‣ Bước 2: Chọn kiểu dữ liệu đính kèm và điều đầy đủ những trường cần thiết ở phần Body



*Hình 4.11 Nhập giá trị gửi kèm trong phần body*

‣ Bước 3: Viết các test case trong phần Tests



*Hình 4.12 Viết các test script trong phần Tests*

▪ Test case có tên “TC-4.2.2.1 Register Successfully” để xem liệu đăng ký có thành công hay không bằng cách kiểm tra:

* status có là 201 không?
* header có thuộc tính ‘Content-Type’ là ‘application/json’ không?

▪ Test case có tên “TC-4.2.2.2 UserInfo is returned” để xem có thông tin được trả về sau khi đã đăng ký bằng cách kiểm tra:

* có thuộc tính ‘id’ trong response hay không?
* thuộc tính ‘username’ có phải kiểu dữ liệu là string không?
* thuộc tính ‘username’ không phải chuỗi rỗng đúng không?

▪ Test case có tên “TC-4.2.2.3 Upload file successfully” để xem liệu có upload avatar thành công hay không bằng cách kiểm tra

* có thuộc tính ‘image’ trong response hay không?
* thuộc tính ‘image’ có chữa chuỗi ‘http://’ hay không?

‣ Bước 4: Bấm Send và xem kết quả



*Hình 4.13 Kết quả sau khi bấm nút Send*

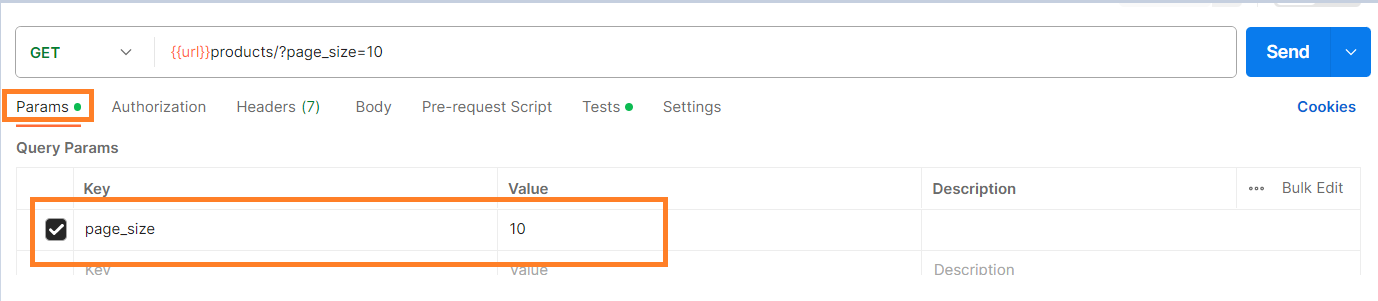
### **4.2.3 Test case 03: Pagination**

‣ Bước 1: Nhập url có sử dụng biến môi trường đã thêm trước đó và chọn phương thức là GET



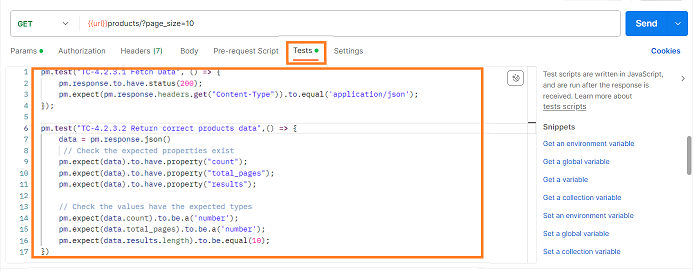
*Hình 4.14 Nhập địa chỉ API và chọn phương thức gửi lên*

‣ Bước 2: Ở phần Params chọn dữ liệu muốn gửi lên và nhập giá trị



*Hình 4.15 Nhập giá trị params gửi kèm trong phần Params*

‣ Bước 3: Viết các test case trong phần Tests



*Hình 4.16 Viết các test script trong phần Tests*

▪ Test case có tên “TC-4.2.3.1 Fetch data” để xem liệu có lấy dữ liệu thành công hay không bằng cách kiểm tra:

- status có là 200 không?

- header có thuộc tính ‘Content-Type’ là ‘application/json’ không?

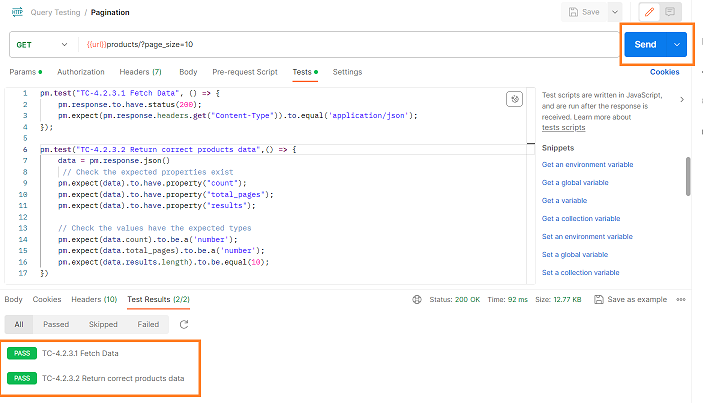
▪ Test case có tên “TC-4.2.2.2 Return correct products data” để xem có thông tin được trả về có đúng hay không bằng các kiểm tra:

- có các thuộc tính ‘count, total\_page, results’ trong response hay không?

- thuộc tính ‘total\_pages có phải kiểu dữ liệu là số không?

- độ dài thuộc tính ‘results’ có bằng với params gửi lên là 10 không?

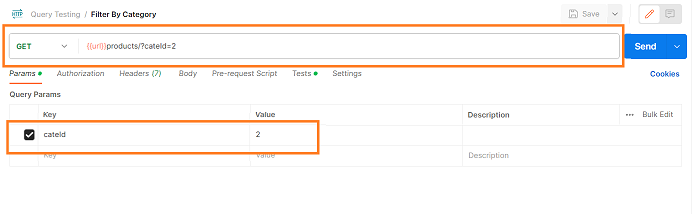
‣ Bước 4: Bấm Send và xem kết quả



*Hình 4.17 Kết quả sau khi bấm nút Send*

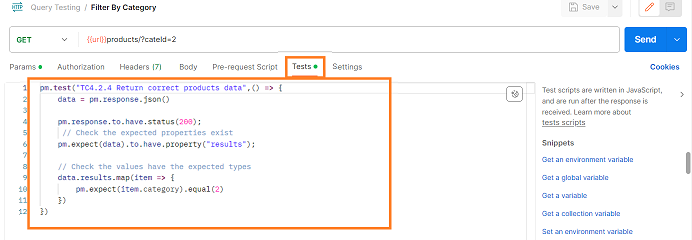
### **4.2.4 Test case 04: Filter by category**

‣ Bước 1: Nhập url có sử dụng biến môi trường đã thêm trước đó, chọn phương thức là GET và gửi kèm giá trị cateId = 2 trong phần Params



*Hình 4.18 Nhập địa chỉ API, chọn phương thức và Params muốn gửi lên*

‣ Bước 2: Viết các test case trong phần Tests



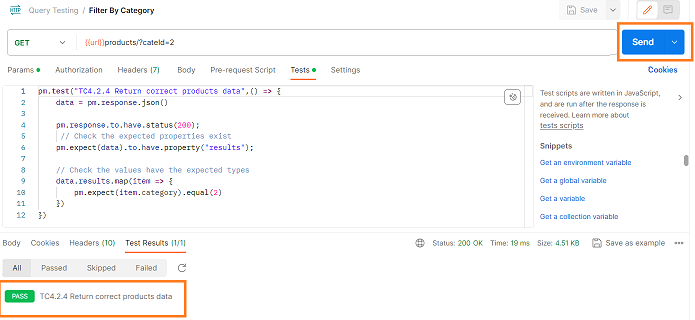
*Hình 4.19 Viết các test script trong phần Tests*

▪ Test case có tên “TC-4.2.4 Return correct products data” để xem thông tin được trả về có đúng hay không bằng các kiểm tra:

- status có là 200 không?

- lặp qua từng dữ liệu products trả về và kiểm tra xem thuộc tính category có bằng giá trị params đã gửi lên là 2 hay không?

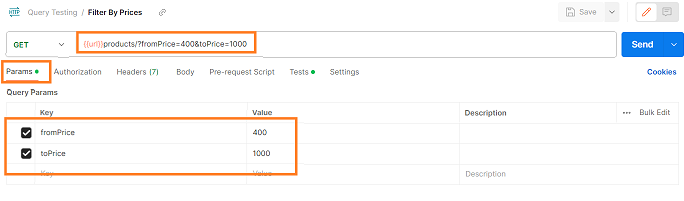
‣ Bước 3: Bấm Send và xem kết quả



*Hình 4.20 Kết quả sau khi bấm nút Send*

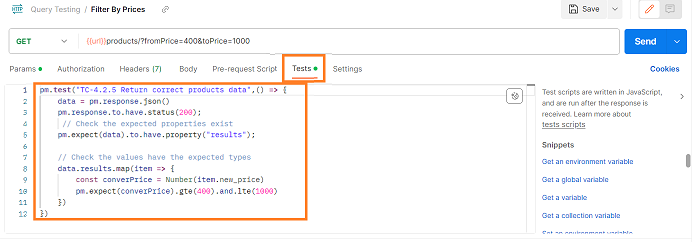
### **4.2.5 Test case 05: Filter by price**

‣ Bước 1: Nhập url có sử dụng biến môi trường đã thêm trước đó, chọn phương thức là GET và gửi kèm giá trị fromPrice = 400, toPrice = 1000 trong phần Params



*Hình 4.21 Nhập địa chỉ API, chọn phương thức và Params muốn gửi lên*

‣ Bước 2: Viết các test case trong phần Tests



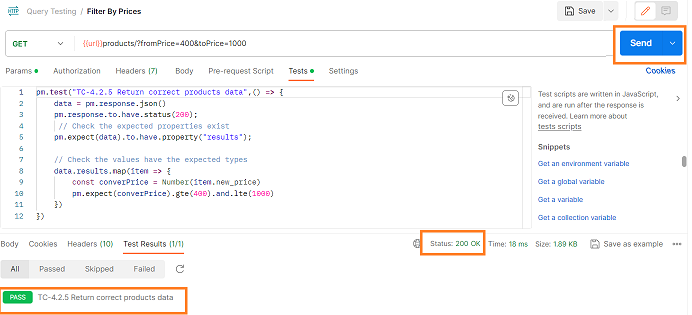
*Hình 4.22 Viết các test script trong phần Tests*

▪ Test case có tên “TC-4.2.5 Return correct products data” để xem thông tin được trả về có đúng hay không bằng các kiểm tra:

- status có là 200 không?

- lặp qua từng dữ liệu products trả về và kiểm tra xem thuộc tính new\_price có lớn hơn 400 và nhỏ hơn 1000 (2 giá trị params đã gửi lên) hay không?

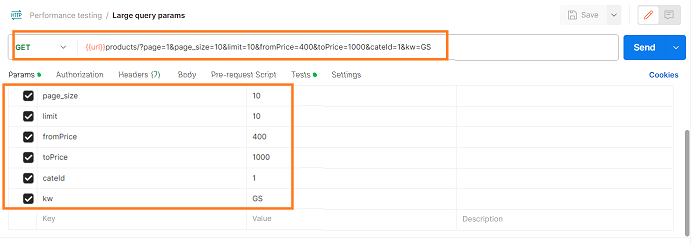
‣ Bước 3: Bấm Send và xem kết quả



*Hình 4.23 Kết quả sau khi bấm nút Send*

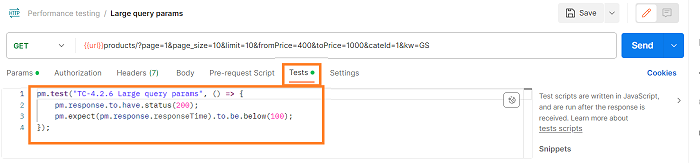
### **4.2.6 Test case 06: Performance**

‣ Bước 1: Nhập url có sử dụng biến môi trường đã thêm trước đó, chọn phương thức là GET và gửi kèm một loạt params như hình



*Hình 4.24 Nhập địa chỉ API, chọn phương thức và Params muốn gửi lên*

‣ Bước 2: Viết các test case trong phần Tests



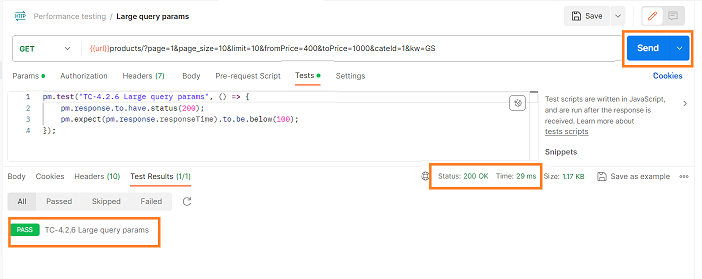
*Hình 4.25 Viết các test script trong phần Tests*

▪ Test case có tên “TC-4.2.6 Return correct products data” để xem thông tin được trả về có đúng hay không bằng các kiểm tra:

- status có là 200 không?

- thời gian trả về dữ liệu có nhỏ hơn 200ms không?

‣ Bước 3: Bấm Send và xem kết quả



*Hình 4.26 Kết quả sau khi bấm nút Send*

### **4.2.7 Test case 07: Update user info**

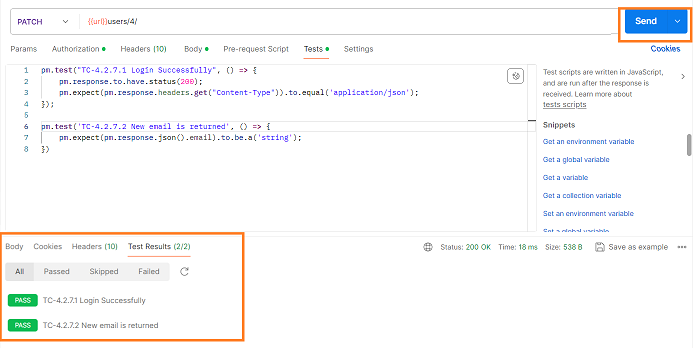
‣ Bước 1: Nhập url, và chọn phương thức là PATH



‣ Bước 2: Viết các test case trong phần Tests

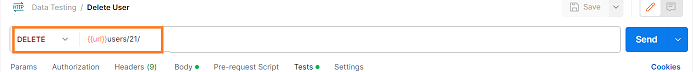


‣ Bước 3: Bấm Send và xem kết quả

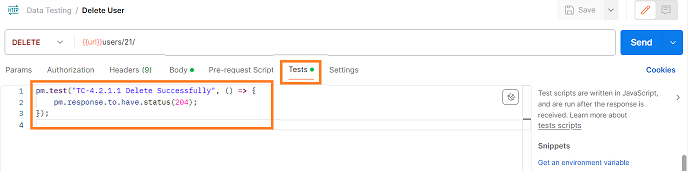


### **4.2.7 Test case 07: Delete User**

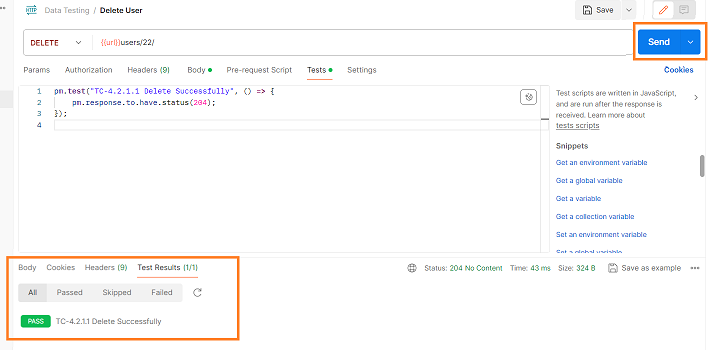
‣ Bước 1: Nhập url, gửi Id user muốn xoá, và chọn phương thức là DELETE



‣ Bước 2: Viết các test case trong phần Tests



‣ Bước 3: Bấm Send và xem kết quả



# **KẾT LUẬN**

Kiểm thử phần mềm hiện nay vẫn là vấn đề hết sức quan trọng với các tổ

chức phát triển phần mềm. Trong quá trình thực hiện báo cáo của mình do thời

gian nghiên cứu và kinh nghiệm bản thân còn hạn chế nên còn một só phần của

báo cáo nghiên cứu chưa được sâu.

Sau một thời gian thực hiện báo cáo dưới sự hướng dẫn của cô Nguyễn Thị Ngọc Thanh báo cáo của em đã thực hiện tốt được các mục tiêu đề ra và đạt được những kết quả sau:

Kết quả đạt được:

Trình bày đầy đủ và chính xác các vấn đề tổng quan về phần mềm, công

nghệ phần mềm, lỗi phần mềm, các vấn đề liên quan đến kiểm thử phần mềm:

▪ Giới thiệu công cụ kiểm thử Selenium Webdriver

▪ Giới thiệu công cụ kiểm thử Postman và các thao tác để sử dụng công cụ này

Áp dụng các kiến thức đã được dạy để thực hiện kiểm thử những chức năng của ứng dụng web đó là chức năng Đăng nhập, Đăng ký, Thay đổ mật khẩu. Và kiểm thử các API của ứng dụng web đó.

Hướng phát triển:

Trong thời gian tới em sẽ tiếp tục nghiên cứu sâu hơn về các vấn đề của

kiểm thử phần mềm, và đặc biệt là bộ công cụ kiểm thử Postman, để có thể vận

dụng vào kiểm thử các ứng dụng lớn hơn trong thực tế công việc trong tương lai

nhằm góp một phần nhỏ bé vào công cuộc chuyên nghiệp hoá kiểm thử phần

mềm ở Việt Nam.