

BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN
PHÂN HIỆU TRƯỜNG ĐẠI HỌC THỦY LỢI
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO MÔN CHUYÊN ĐỀ CNTT

Tên đề tài: Kiểm thử phần mềm bằng công cụ Selenium

Giảng viên hướng dẫn:	<i>ThS. Viên Thanh Nhã</i>
Sinh viên thực hiện:	Vũ Trí Minh
Mssv:	2051067806
Lớp:	S23-K62TH

TPHCM, ngày 22 tháng 10 năm 2023

LỜI NÓI ĐẦU

Với mong muốn có cái nhìn xác thực, rõ ràng hơn về kiểm thử phần mềm và tiếp cận được với công cụ kiểm thử tự động Selenium. Em lựa chọn đề tài “Kiểm thử website bằng công cụ Selenium” làm đề tài cho bài tập lớn môn Kiểm Thử Phần Mềm của mình. Sau một thời gian thì em đã hoàn thành báo cáo bài tập lớn của mình.

Trong khuôn khổ bài tập lớn, do thời gian và kinh nghiệm thực tế còn hạn chế nên có những phần thực hiện chưa được tốt, em rất mong nhận được sự góp ý của thầy và các bạn.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

VŨ TRÍ MINH

Mục Lục

Chương 1: XÁC ĐỊNH VÀ PHÂN TÍCH YÊU CẦU	4
1.1 Lý do chọn đề tài	4
1.2 Nhiệm vụ cần làm	5
1.3 Kết quả cần đạt	5
Chương 2 : GIỚI THIỆU VỀ CÁC CÔNG CỤ TESTING	5
2.1 Selenium là gì ?	5
2.2 Các loại selenium	6
2.2.1 Selenium IDE	6
2.2.2 Selenium Webdriver	7
2.3 Cách cài đặt Selenium	8
2.3.1 Selenium IDE	8
2.3.2 Selenium Webdriver	10
2.4 Các tính năng của Selenium IDE	12
2.4.1. Kiểm tra	12
2.4.2. Test case	12
2.4.3 Thêm một bộ	13
2.4.4 Xóa bài kiểm tra	13
2.4.5 Xóa hoặc đổi tên một bộ	13
2.4.6 Lưu công việc của bạn	14
2.4.7 Phát lại	14
2.5 Các câu lệnh thường gặp	14
2.5.1 Selenium IDE	14
2.5.2 Selenium Webdriver	16
Chương 3: CÁCH THỰC HIỆN TESTING TỰ ĐỘNG	20
3.1 Thực hiện trên Selenium IDE	20
3.2 Thực hiện trên Selenium Webdriver	31
Chương 4: KẾT QUẢ ĐẠT ĐƯỢC	37
4.1. Kết quả đạt được của đề tài	37
4.2. Hạn chế của đề tài	38
4.3. Hướng phát triển của đề tài	38
Chương 5: DEMO	38
5.1 Kịch bản demo	38

5.1.1 Selenium IDE	38
5.1.2 Selenium Webdriver	39

Chương 1: XÁC ĐỊNH VÀ PHÂN TÍCH YÊU CẦU

1.1 Lý do chọn đề tài

- Tăng hiệu quả: Kiểm thử thủ công có thể tốn thời gian và dễ xảy ra lỗi, đặc biệt đối với các dự án lớn hoặc phức tạp. Tự động hóa có thể giúp tăng tốc độ thử nghiệm và giảm nguy cơ lỗi của con người.
- Phạm vi bao phủ được cải thiện: Kiểm tra tự động có thể bao gồm nhiều trường hợp sử dụng hơn và tìm thấy nhiều lỗi hơn so với kiểm tra thủ công. Bằng cách tự động hóa quy trình thử nghiệm, bạn có thể chạy nhiều thử nghiệm hơn trong thời gian ngắn hơn và sớm phát hiện các vấn đề trong quá trình phát triển.
- Khả năng lặp lại: Kiểm tra tự động đảm bảo rằng các bài kiểm tra giống nhau được chạy mỗi lần và kết quả nhất quán. Điều này có thể giúp xác định bất kỳ vấn đề nào có thể xảy ra trong quá trình kiểm tra thủ công và giúp theo dõi các thay đổi theo thời gian dễ dàng hơn.
- Tiết kiệm chi phí: Kiểm thử tự động có thể tiết kiệm chi phí hơn so với kiểm thử thủ công, đặc biệt đối với các dự án lớn. Bằng cách giảm nhu cầu nỗ lực thủ công, bạn có thể tiết kiệm thời gian và tài nguyên, đồng thời sử dụng chúng cho các khía cạnh khác của dự án.

- Cải thiện trải nghiệm người dùng: Kiểm tra tự động có thể giúp đảm bảo trải nghiệm người dùng nhất quán và không có lỗi. Bằng cách kiểm tra ứng dụng web của bạn thường xuyên bằng một bộ kiểm tra tự động, bạn có thể tìm và khắc phục sự cố trước khi chúng ảnh hưởng đến người dùng của mình.

1.2 Nhiệm vụ cần làm

- Cài đặt và cấu hình Selenium WebDriver
- Viết tập lệnh để tương tác với ứng dụng web, chẳng hạn như nhấp vào nút, điền vào biểu mẫu và điều hướng đến các trang khác nhau
- Xử lý các trường hợp ngoại lệ và thông báo lỗi do Selenium tạo ra
- Triển khai các chức năng chờ để đảm bảo các phần tử có mặt trước khi tương tác với chúng
- Sử dụng bộ chọn XPath hoặc CSS để xác định các thành phần trên trang web

1.3 Kết quả cần đạt

- Nắm được lý thuyết kiểm thử phần mềm, kiểm thử tự động phần mềm.
- Nắm được lý thuyết về công cụ kiểm thử tự động Selenium.
- Ứng dụng được công cụ Selenim IDE, Selenium RC. - Selenium Grid, Selenium Webdriver vào kiểm thử một số trang web.

Chương 2 : GIỚI THIỆU VỀ CÁC CÔNG CỤ TESTING

2.1 Selenium là gì ?

- Selenium là một bộ công cụ kiểm thử tự động open source, dành cho các ứng dụng web, hỗ trợ hoạt động trên nhiều trình duyệt và nền tảng khác nhau như Windows, Mac, Linus... Với Selenium, bạn có thể viết các testscript bằng các ngôn ngữ lập trình khác nhau như Java, PHP, C#, Ruby hay Python hay thậm chí là Perl...
- Selenium được sử dụng để automate các thao tác với trình duyệt, hay dễ hiểu hơn là nó giúp giả lập lại các tương tác trên trình duyệt như một người dùng thực sự

2.2 Các loại selenium

2.2.1 Selenium IDE

- Selenium IDE là một công cụ phát triển web hoàn toàn miễn phí được sử dụng để thử nghiệm và phát triển các thử nghiệm tự động trên các web ứng dụng. Selenium IDE cho phép người dùng xác định các thành phần, hoạt động và tương tác với ứng dụng cũng như tạo các tập lệnh tự động. Tính năng nổi bật của Selenium IDE bao gồm khả năng phát lại, ghi và chỉnh sửa tập lệnh, hỗ trợ cho các trình duyệt khác nhau như Firefox, Chrome, Safari vv và hỗ trợ nhiều ngôn ngữ khác nhau.



Hình 2.1

2.2.2 *Selenium Webdriver*

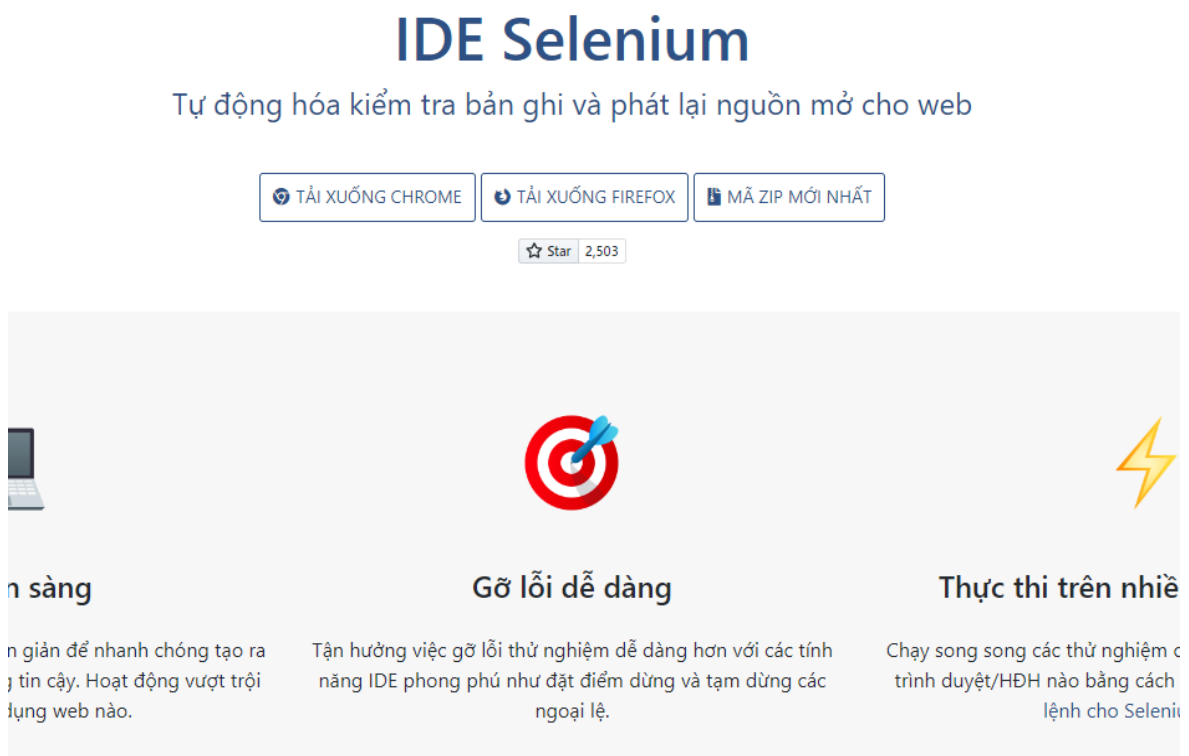
- Selenium WebDriver là một API khung, cho phép thành viên lập trình tự động tạo các bài kiểm tra để kiểm tra các tính năng của web ứng dụng và ứng dụng. Việc sử dụng WebDriver cho phép người dùng cài đặt chấm vào các thành phần của một trang web bất kỳ và gửi các lệnh, điều khiển ứng dụng và trang web. Selenium WebDriver cũng cho phép người dùng lập trình liên kết với nhiều trình duyệt và nền tảng khác nhau.



2.3 Cách cài đặt Selenium

2.3.1 Selenium IDE

- Điều kiện : Đã cài đặt Google Chrome
- Để dễ hiểu, tôi đã phân chia toàn bộ quá trình cài đặt IDE theo các bước sau:
- Bước 1: Tải xuống Selenium IDE: Mở trình duyệt Google Chrome và nhập URL <https://www.selenium.dev/selenium-ide/>
- Nó sẽ mở ra trang web trụ sở chính của Selenium. Click vào button "Tải xuống Chrome hoặc FireFox" tùy nhu cầu mỗi người, nó sẽ điều hướng đến trang Tải xuống; trang này bao gồm tất cả các bản phát hành mới nhất của tất cả các thành phần Selenium.
- Tham khảo hình sau:



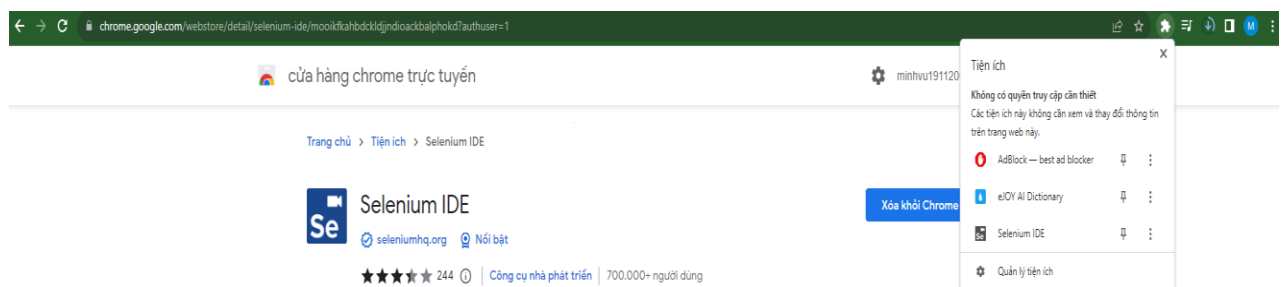
Hình 2.4

- Bước 3: Ngay sau khi chúng ta nhấp vào tải xuống , trang sẽ tự động chuyển hướng sang cửa hàng trực tuyến trên Chrome. Tại đó chúng ta chỉ cần nhấn vào nút cài đặt thì Chrome sẽ tự động tải xuống trình duyệt!



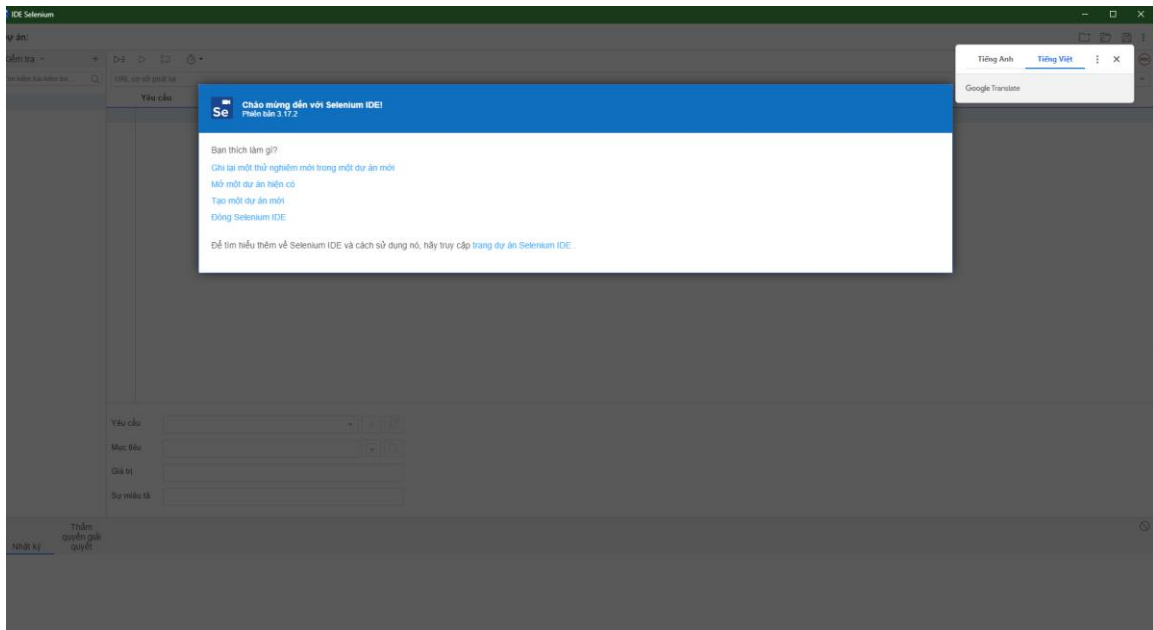
Hình 2.5

- Bước 4: Tại thanh tìm kiếm bên phải màn hình chúng ta nhấn vào icon "Tiện ích" và Selenium đã được cài đặt xong



Hình 2.6

- Bước 5: Ngay sau khi chúng ta mở Selenium IDE, cửa sổ Selenium IDE sẽ xuất hiện.



Hình 2.7

2.3.2 Selenium Webdriver

- Khác với cách cài đặt IDE vì Selenium Webdriver cần sử dụng công cụ hỗ trợ code để tương tác và kiểm tra web như Python, Java, CSharp...
- Cách cài đặt thư viện Webdriver như sau:
 - Điều kiện: phải có sẵn 1 trong các phần mềm hỗ trợ code trong máy như Python, JAVA, CSharp...
 - Có một số cách khác nhau để cài đặt Selenium:

Pip

```
pip install selenium
```

- Tải xuống:

- Ngoài ra, bạn có thể tải xuống <https://pypi.org/project/selenium/#files> (selenium-xxx.tar.gz) và cài đặt nó bằng setup.py :

```
python setup.py install
```

2.3.3 Selenium Grid

* Cách cài đặt Grid:

1. Điều kiện :

- Đã cài đặt Java 11 trở lên
- (Các) trình duyệt đã được cài đặt
- (Các) trình điều khiển trình duyệt
- Trình quản lý Selenium sẽ tự động định cấu hình trình điều khiển nếu bạn thêm `--selenium-manager true`
- Đã cài đặt và trên PATH
- Tải xuống tệp jar Selenium Server từ bản phát hành mới nhất

2. Bắt đầu lưới

```
java -jar selenium-server-<version>.jar standalone
```

3. Trở* các bài kiểm tra WebDriver của bạn tới \

<http://localhost:4444/> HYPERLINK "http://localhost:4444/"4444

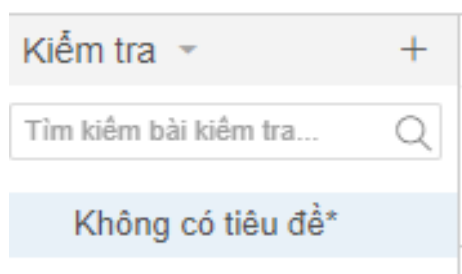
4. (Tùy chọn) Kiểm tra các bài kiểm tra đang chạy và các khả năng sẵn có bằng cách mở trình duyệt của bạn tại

<http://localhost:4444/> HYPERLINK "http://localhost:4444/"4444

2.4 Các tính năng của Selenium IDE

2.4.1. Kiểm tra

- Bạn có thể thêm bài kiểm tra mới bằng cách nhấp vào **+** biểu tượng ở đầu menu thanh bên trái (ở bên phải tiêu **Tests** đề), đặt tên cho nó và nhấp vào **ADD**.
- Sau khi thêm, bạn có thể nhập lệnh theo cách thủ công hoặc nhấp vào biểu tượng bản ghi ở phía trên bên phải của IDE.



Hình 2.8

2.4.2. Test case

- Các thử nghiệm có thể được nhóm lại với nhau thành các dãy.
- Khi tạo dự án, a **Default Suite** sẽ được tạo và thử nghiệm đầu tiên của bạn sẽ tự động được thêm vào đó.
- Để tạo và quản lý các bộ, hãy vào **Test Suites** bảng điều khiển. Bạn có thể đến đó bằng cách nhấp vào trình đơn thả xuống ở đầu menu thanh bên trái (ví dụ: nhấp vào từ **Tests**) và chọn **Test Suites**.

2.4.3 Thêm một bộ

- Để thêm một bộ, hãy nhấp vào **+** biểu tượng ở đầu menu thanh bên trái ở bên phải tiêu đề, cung cấp tên và nhấp vào **ADD**
- Thêm một bài kiểm tra
- Để thêm bài kiểm tra vào một bộ, hãy di chuột qua tên bộ, sau đó thực hiện như sau:
 - + Bấm vào biểu tượng xuất hiện bên phải tiêu đề
- đề
 - + Nhấp chuột
- + Chọn các bài kiểm tra bạn muốn thêm từ menu
- + Nhấp chuột

2.4.4 Xóa bài kiểm tra

- Để xóa bài kiểm tra, hãy di chuột qua bài kiểm tra và nhấp vào nút **X**
- xuất hiện ở bên phải tên.

2.4.5 Xóa hoặc đổi tên một bộ

- Để xóa một bộ, hãy nhấp vào biểu tượng xuất hiện ở bên phải tên của nó, nhấp vào **Delete** và nhấp **Delete** lại khi được nhắc.

- Để đổi tên một bộ, hãy di chuột qua tên bộ, nhấp vào biểu tượng xuất hiện ở bên phải tên, nhấp vào `Rename`, cập nhật tên và nhấp vào `RENAME`

2.4.6 Lưu công việc của bạn

- Để lưu mọi thứ bạn vừa thực hiện trong IDE, hãy nhấp vào biểu tượng lưu ở góc trên bên phải của IDE.
- Nó sẽ nhắc bạn về vị trí và tên nơi lưu dự án. Kết quả cuối cùng là một tệp có `.side` phần mở rộng.

2.4.7 Phát lại

- Trong trình duyệt
- Bạn có thể phát lại các bài kiểm tra trong IDE bằng cách chọn bài kiểm tra hoặc bộ bạn muốn phát và nhấp vào nút phát trên thanh menu phía trên trình chỉnh sửa bài kiểm tra.
- (Các) bài kiểm tra sẽ phát lại trong trình duyệt. Nếu một cửa sổ vẫn mở trong quá trình ghi, nó sẽ được sử dụng để phát lại. Nếu không, một cửa sổ mới sẽ được mở và sử dụng.

2.5 Các câu lệnh thường gặp

2.5.1 Selenium IDE

`assert`

- Kiểm tra xem một biến có phải là giá trị mong đợi hay không. Giá trị của biến sẽ được chuyển thành chuỗi để so sánh. Quá trình kiểm tra sẽ dừng nếu xác nhận không thành công.

assert alert

- Xác nhận rằng cảnh báo đã được hiển thị cùng với văn bản được cung cấp. Quá trình kiểm tra sẽ dừng nếu xác nhận không thành công.

assert title

- Xác nhận tiêu đề của trang hiện tại có chứa văn bản được cung cấp. Quá trình kiểm tra sẽ dừng nếu xác nhận không thành công.

click

- Nhấp chuột vào phần tử mục tiêu (ví dụ: liên kết, nút, hộp kiểm hoặc nút radio)

close

- Đóng cửa sổ hiện tại. Không cần phải đóng cửa sổ ban đầu, IDE sẽ sử dụng lại nó; việc đóng nó có thể gây ra hiệu suất kém trong bài kiểm tra.

double click

- Nhấp đúp chuột vào một phần tử (ví dụ: liên kết, nút, hộp kiểm hoặc nút radio).

open

- Mở một URL và đợi trang tải trước khi tiếp tục. Điều này chấp nhận cả URL tương đối và tuyệt đối.

pause

- Chờ trong khoảng thời gian quy định.

select

- Chọn một phần tử từ menu thả xuống bằng cách sử dụng công cụ định vị tùy chọn. Bộ định vị tùy chọn cung cấp các cách khác nhau để chỉ định phần tử được chọn (ví dụ: `nhãn=`, `value=`, `id=`, `index=`). Nếu không có tiền tố định vị tùy chọn nào được cung cấp thì kết quả khớp trên nhãn sẽ được thử..

type

- Đặt giá trị của trường đầu vào, như thể bạn đã nhập nó vào. Cũng có thể được sử dụng để đặt giá trị của hộp tổ hợp, hộp kiểm, v.v. Trong những trường hợp này, giá trị phải là giá trị của tùy chọn đã chọn, không phải văn bản hiển thị. Chỉ Chrome: Nếu đường dẫn tệp được cung cấp, đường dẫn đó sẽ được tải lên đầu vào (đối với `loại=tệp`), LƯU Ý: Trình định vị XPath không được hỗ trợ.

verify

- Khẳng định rằng một biến là một giá trị mong đợi. Giá trị của biến sẽ được chuyển thành chuỗi để so sánh. Quá trình kiểm tra sẽ tiếp tục ngay cả khi việc xác minh không thành công.

2.5.2 Selenium Webdriver

- Mọi thứ Selenium làm là gửi lệnh trình duyệt để thực hiện điều gì đó hoặc gửi yêu cầu cung cấp thông tin. Hầu hết những gì bạn sẽ làm với Selenium là sự kết hợp của các lệnh cơ bản sau. Ở đây là các câu lệnh thường gặp khi code bằng phần mềm Python.

* Các hàm cơ bản của WebElement

1) get()

- Để mở một URL trong trình duyệt hiện tại.
- Cú pháp:
- `driver.get("https://www.softwaretestinghelp.com");`
- Giải thích:
- Điều hướng đến URL <https://www.softwaretestinghelp.com>

2) *getCurrentUrl()*

- Được sử dụng URL hiện tại của trang đang được truy cập, thường được sử dụng khi muốn kiểm tra URL xem dữ liệu đã chính xác hay chưa.
- Cú pháp lấy URL - Cú pháp:
- `driver.getCurrentUrl();`
- Thường sử dụng phương pháp này trong các lệnh để kiểm tra xem điều chỉnh hướng đến trang đúng như mong đợi hay chưa. Để làm được điều đó, chúng ta phải sử dụng Assert:
- `Assert.assertEquals(expectedUrl, driver.getCurrentUrl());` mong đợiUrl : URL mong muốn

3) *findElement(By, by)* và *click()*

- `findElement(By, by)` và `click()` để Click vào bất kỳ phần tử nào của trang
- Tìm kiếm phương thức `findElement(By, by)` và định vị phần tử đầu tiên trên trang hiện tại, sao cho phù hợp với tiêu chí được đưa ra dưới dạng tham số. Phương pháp này thường được sử dụng trong các lệnh để mô phỏng hành động của người dùng như nhấp (nhấp), gửi (gửi), nhập (loại), vv..
- Ví dụ lệnh bên dưới tìm kiếm và định vị phần tử đầu tiên trong trang web có id "submit1" và nhấp vào nó nếu phần tử đó không được che:
- Cú pháp:

- `driver.findElement(By.id("submit1")).click();`
- Có thể định vị phần tử web theo ID , Name , Class Name , Tag Name , Link Text & Partial Link Text , CSS Selector hoặc X Path .
- Giải thích:
- Find node Submit được yêu cầu
- Click chuột vào nút đó
- Hoặc ví dụ về chọn 1 mục trong danh sách thả xuống của mục:
- `WebElement roleDropdown = driver.findElement(By.id("name1"));`
- `roleDropdown.click();`
- Tìm và định vị mục có id là “name1”.
- Bấm vào mục đó.

4) *isEnabled()*

- Kiểm tra phần web xem phần tử là Bật hay Tắt. Ví dụ:
- `boolean textBox =`
`driver.findElement(By.xpath("//input[@name='textbox1']")).isEnabled();`

5) *findElement(By, by) với sendKeys()*

- Use khi bạn muốn nhập giá trị vào trường biểu mẫu.
- Kiểm tra mẫu xác thực bằng cách nhập các đầu giá trị khác nhau của những người dùng thường được yêu cầu trong quá trình tự động hóa kiểm tra.
 Chúng tôi sử dụng `findElement(By, by)` để định vị các trường/trường và `sendKeys()` nhập một số nội dung vào trường có thể chỉnh sửa.
- Ví dụ dưới đây, sử dụng Tên định vị để tìm trường và nhập văn bản “Aaron” cho nó:

- `driver.findElement(By.name("name")).sendKeys("Aaron");`

6) *findElement(By, by) với getText()*

- Tìm kiếm phần tử web và lấy bản văn bản trong thẻ HTML tương ứng:
- `String dropDown =`
`driver.findElement(By.tagName("dropdown1")).getText();`
- Tìm kiếm phần tử có tagName “dropdown1”.
- Lấy văn bản bên trong thẻ HTML của nó.
- Lưu trữ văn bản này vào Chuỗi đối tượng 'dropDown'.

7) *Gửi()*

- Thực hiện nhấp vào nút gửi và gửi biểu mẫu. Nút gửi nằm bên trong thẻ HTML 'biểu mẫu' và nút loại là 'gửi' (không phải 'nút'). `Submit()` làm cho cuộc sống trở nên dễ dàng hơn bằng cách tự động tìm nút và có thể thêm vào bất kỳ trường nào khác như tên hoặc email địa chỉ trong phương thức để gửi biểu mẫu.
- Cú pháp:
- `driver.findElement(By.xpath("//input[@name='comments']")).submit();`
- Tìm kiếm phần tử web mà xpath có tên là 'comments'.
- Gửi biểu mẫu.

8) *findElements(By, by)*

- Tìm kiếm web các phần tử trong danh sách.
- Cú pháp:

- `List<WebElement> allChoices =`
`dropDown.findElements(By.xpath("//fruitoption"));`
- Lấy danh sách tất cả các phần tử web với xpath "`//fruitoption`". Tạo 1 List kê phần tử web có tên `allChoices` để lưu trữ danh sách nhận được điều này.

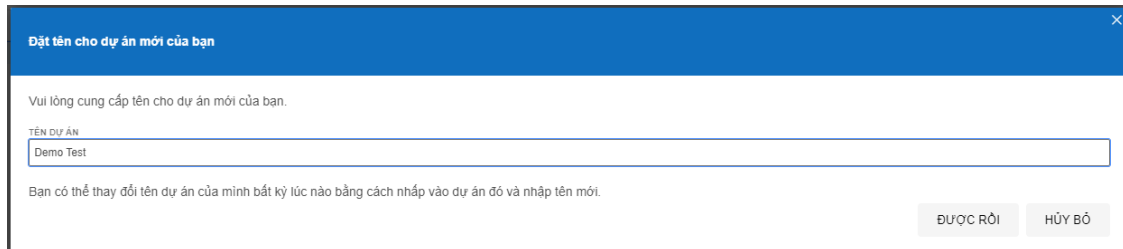
Chương 3: CÁCH THỰC HIỆN TESTING TỰ ĐỘNG

3.1 Thực hiện trên Selenium IDE

- Kịch bản kiểm thử:
 - B1. Mở trình duyệt Google Chrome và truy cập <https://www.chotot.com/>
 - B2. Nhấn vào thanh menu tài khoản
 - B3. Nhấn vào đăng nhập/đăng kí
 - B4. Nhập tài khoản
 - B5. Nhập mật khẩu
 - B6. Nhấn vào Đăng Nhập
- Sau đây là các bước để tiến hành kiểm thử trên trình duyệt Google Chrome bằng Selenium IDE:

B1. Mở trình duyệt Google Chrome và truy cập <https://www.chotot.com/>

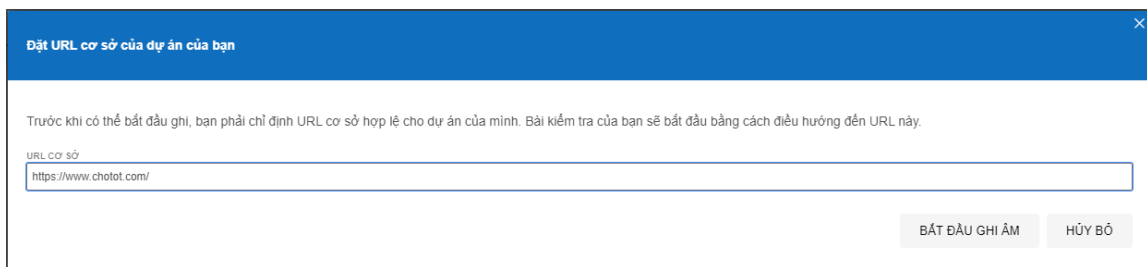
- Trước khi bắt đầu, ta tạo project mới tên Demo Test



The screenshot shows a dialog box titled "Đặt tên cho dự án mới của bạn" (Set name for your new project). It contains the instruction "Vui lòng cung cấp tên cho dự án mới của bạn." (Please provide a name for your new project.) and a text input field labeled "TÊN DỰ ÁN" (Project Name) with the value "Demo Test". Below the input field, it says "Bạn có thể thay đổi tên dự án của mình bất kỳ lúc nào bằng cách nhấp vào dự án đó và nhập tên mới." (You can change the name of your project at any time by clicking on the project and entering a new name.) At the bottom right, there are two buttons: "ĐƯỢC RỒI" (OK) and "HỦY BỎ" (Cancel).

Hình 3.1

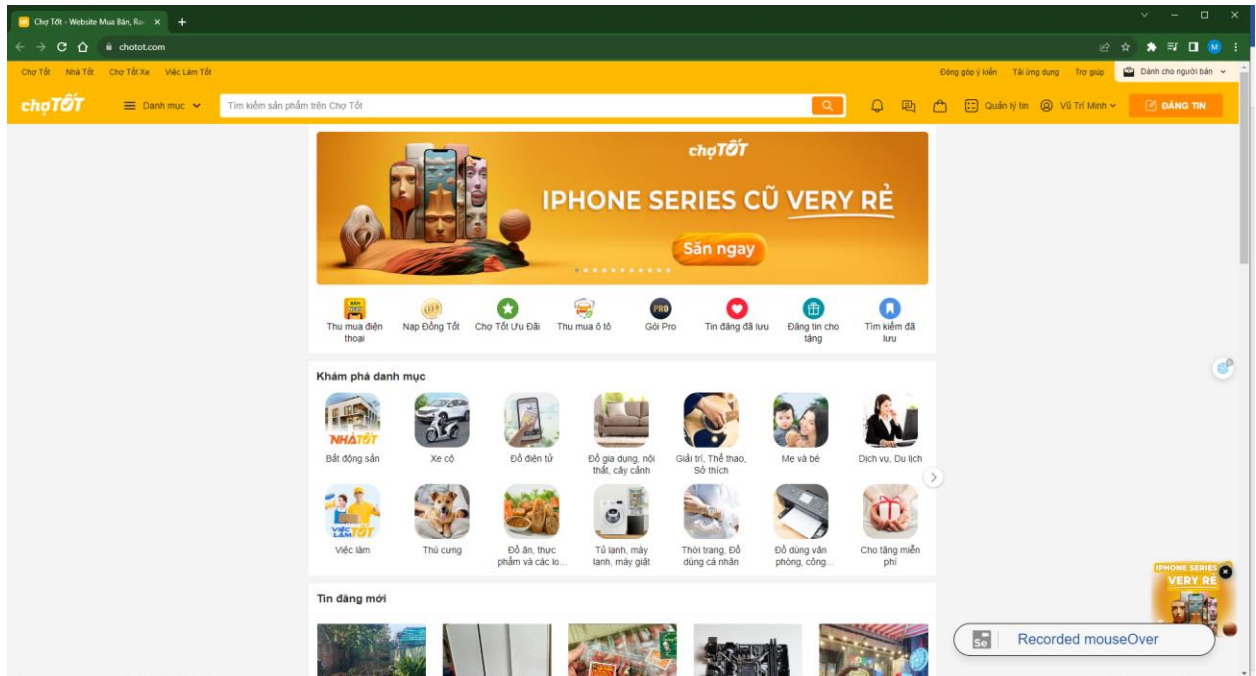
- Copy đường dẫn trang web



The screenshot shows a dialog box titled "Đặt URL cơ sở của dự án của bạn" (Set base URL for your project). It contains the instruction "Trước khi có thể bắt đầu ghi, bạn phải chỉ định URL cơ sở hợp lệ cho dự án của mình. Bài kiểm tra của bạn sẽ bắt đầu bằng cách điều hướng đến URL này." (Before you can start recording, you must specify a valid base URL for your project. Your test will start by navigating to this URL.) and a text input field labeled "URL CƠ SỞ" (Base URL) with the value "https://www.chotot.com/". At the bottom right, there are two buttons: "BẮT ĐẦU GHI ÂM" (Start Recording) and "HỦY BỎ" (Cancel).

Hình 3.2

- Cửa sổ của trang web hiện lên



Hình 3.3

- Hoặc ta có thể mở trang web bằng cách tạo lệnh “Open” để mở trang web tự động bằng lệnh
- Đầu tiên, ta nhấp vào khoảng trống dòng thứ nhất , ở ô “Yêu Cầu” phía dưới

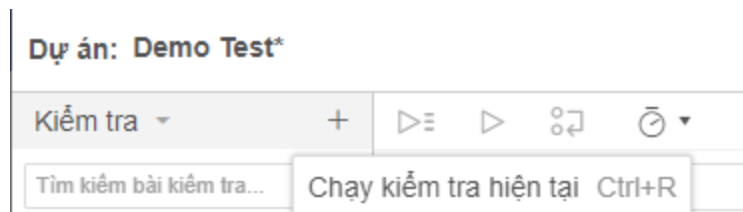
Yêu cầu	<input type="text"/>	//	
---------	----------------------	----	--

- Ta nhập hàm ”Open” vào ô trống
- Ở ô “Mục Tiêu” phía dưới, ta nhập vào địa chỉ của trang web

Yêu cầu	<input type="text" value="open"/>	//	
Mục tiêu	<input type="text" value="https://www.chotot.com/"/>		

Hình 3.4

- Sau khi nhập xong, ta nhấn vào nút”Chạy kiểm tra hiện tại”(Run Test)

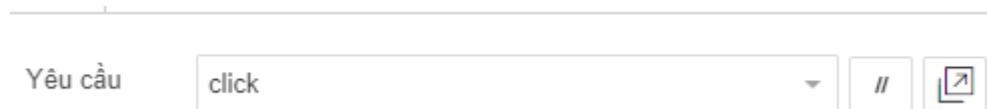


Hình 3.5

- Sau khi chạy thử thì phần mềm sẽ tự động mở trang web lên

B2. Nhấn vào thanh menu tài khoản

- Để nhấn vào một phần tử bất kỳ trên trang web ta dùng hàm sự kiện “Click”
- Quay lại IDE, ta tiếp tục nhấn vào ô trống thứ 2 để thực hành câu lệnh tiếp theo
 - Thao tác là :Nhấn vào nút “Yêu Cầu” nhập câu lệnh “Click”



Hình 3.6

- Ở ô Mục Tiêu phía dưới ta tiếp tục nhấn vào nút “Chọn mục tiêu trong trang” kế bên ô trống



Hình 3.7

- Sau đó IDE sẽ tự động đưa chúng ta đến trang web và nhấp vào ô được định

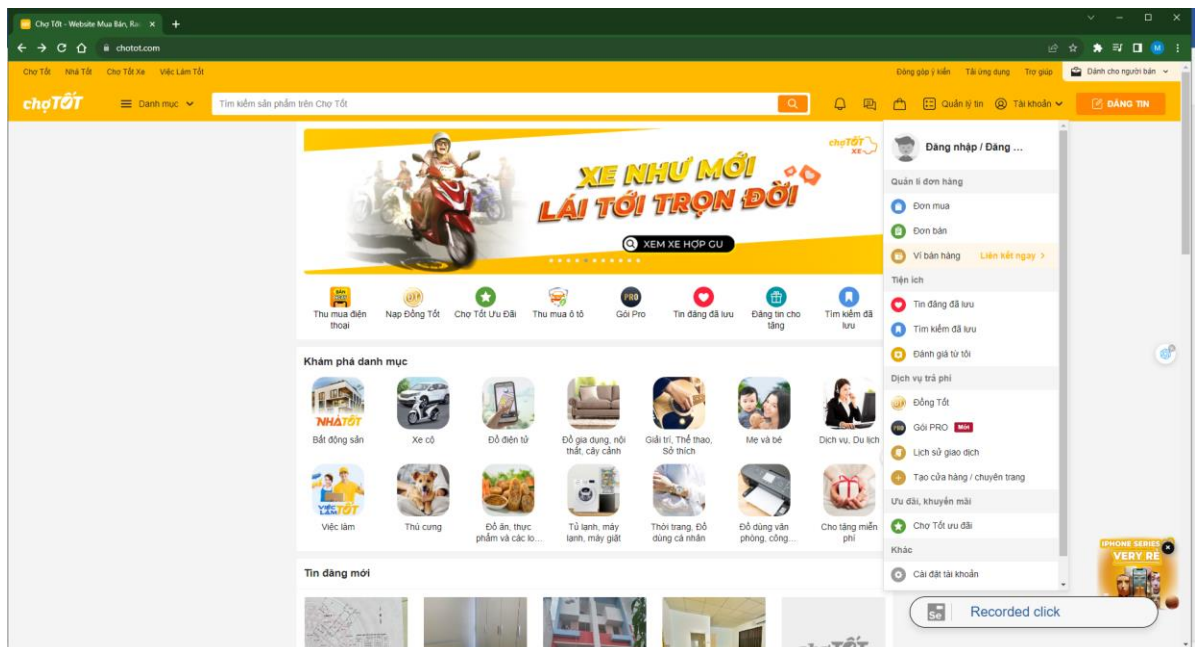
sẵn(nút menu) 

- Quay lại IDE, ta thấy địa chỉ của nút menu được nhập tự động

Yêu cầu	click
Mục tiêu	css=.aw__dvoj89e

Hình 3.8

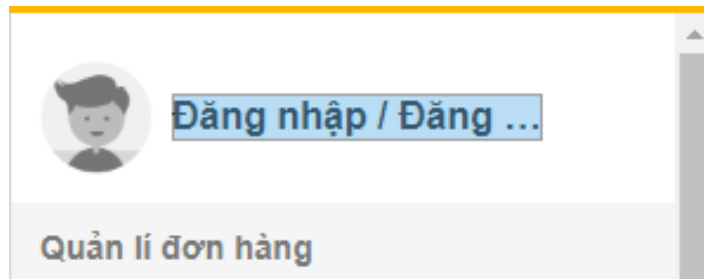
- Đây là hình ảnh sau khi chạy kiểm thử, thanh menu công cụ hiện ra.



Hình 3.9

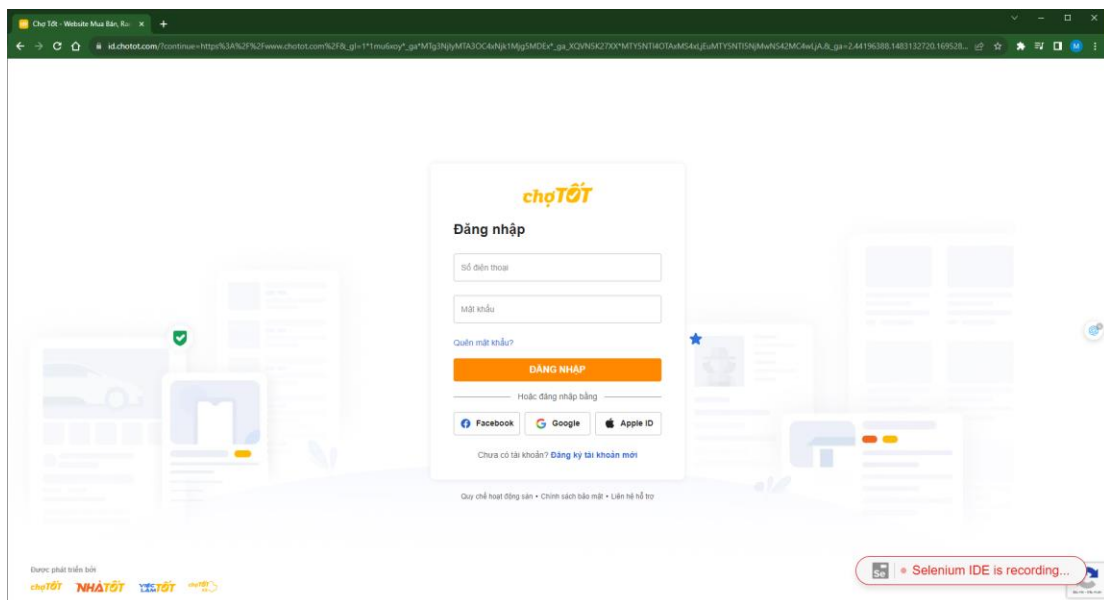
B3. Nhấn vào đăng nhập/dăng kí

- Ở B.3, ta cũng dùng sự kiện "Click" để nhấn vào ô "Đăng nhập/Đăng kí"
- Tương tự như B2, ta cũng tạo một câu lệnh mới và nhập "Click" vào ô Yêu Cầu
- Sau đó ta tiếp tục nhấn vào ô "Chọn mục tiêu trong trang". Chuyển hướng đến trang web ta nhấn vào ô "Đăng nhập/Đăng kí"



Hình 3.10

- Quay lại IDE, sau khi ta chạy kiểm thử thì web sẽ tự động nhấn vào ô "Đăng nhập/Đăng kí" trong trang và chuyển hướng đến trang đăng nhập



Hình 3.11

B4. Nhập tài khoản

- Để nhập giá trị của phần tử ta dùng hàm "Type"
- Thao tác như sau:
- Tạo một dòng lệnh mới, ấn vào ô "Yêu Cầu" nhập chữ "Type"
- Tiếp theo ta nhấn vào nút "Chọn mục tiêu trong trang" ở ô Mục Tiêu

Hình 3.12

- Quay lại IDE, ta thấy ô Mục Tiêu đã được nhập tự động





Hình 3.13

- Tiếp theo ta nhìn xuống ô “Giá Trị” ở dưới phần Mục Tiêu

Mục tiêu	<input type="text" value="name=phone"/>		
Giá trị	<input type="text"/>		

Hình 3.14

- Ta nhập giá trị của ô tài khoản (nhập số điện thoại)

Yêu cầu	<input type="text" value="type"/>		
Mục tiêu	<input type="text" value="name=phone"/>		
Giá trị	<input type="text" value="07*****"/>		

Hình 3.15

- Sau khi chạy kiểm thử ta được

chợTỐT


Đăng nhập


070000000000


[Quên mật khẩu?](#)

ĐĂNG NHẬP

Hoặc đăng nhập bằng

 Facebook

 Google

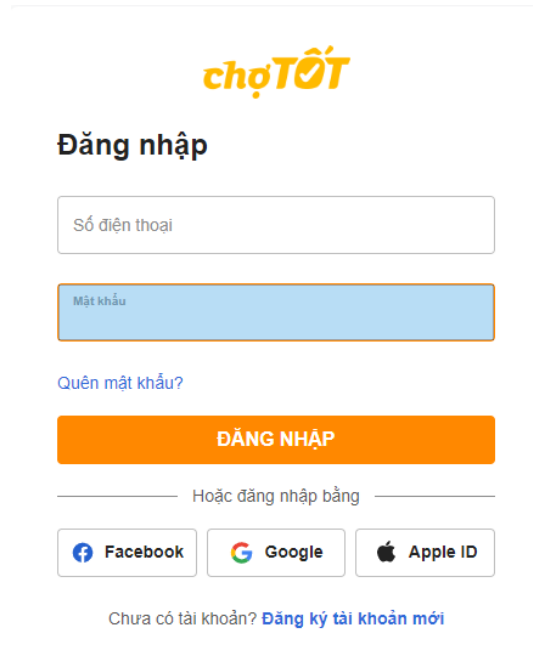
 Apple ID

Chưa có tài khoản? [Đăng ký tài khoản mới](#)

Hình 3.16

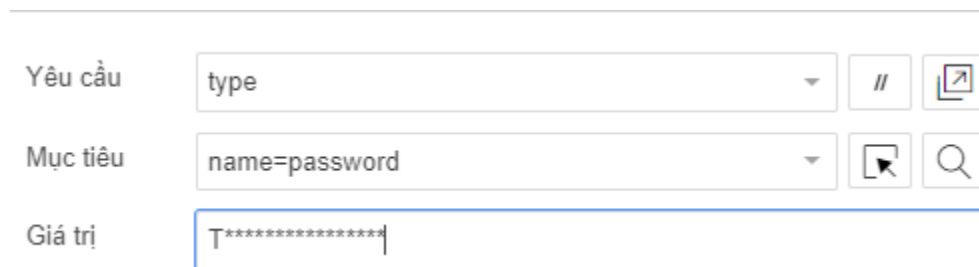
B5. Nhập mật khẩu

- Tương tự như ở B.4, ta cũng tạo dòng lệnh mới và sử dụng hàm “Type” để nhập giá trị cho ô mật khẩu
- Nhấn “Chọn mục tiêu trong trang” và nhấn vào ô mật khẩu



Hình 3.17

- Sau đó, ta tiếp tục nhấn vào ô “Giá Trị” và nhập mật khẩu của tài khoản chợ tốt



Hình 3.18

- Sau khi chạy kiểm thử

Hình 3.19

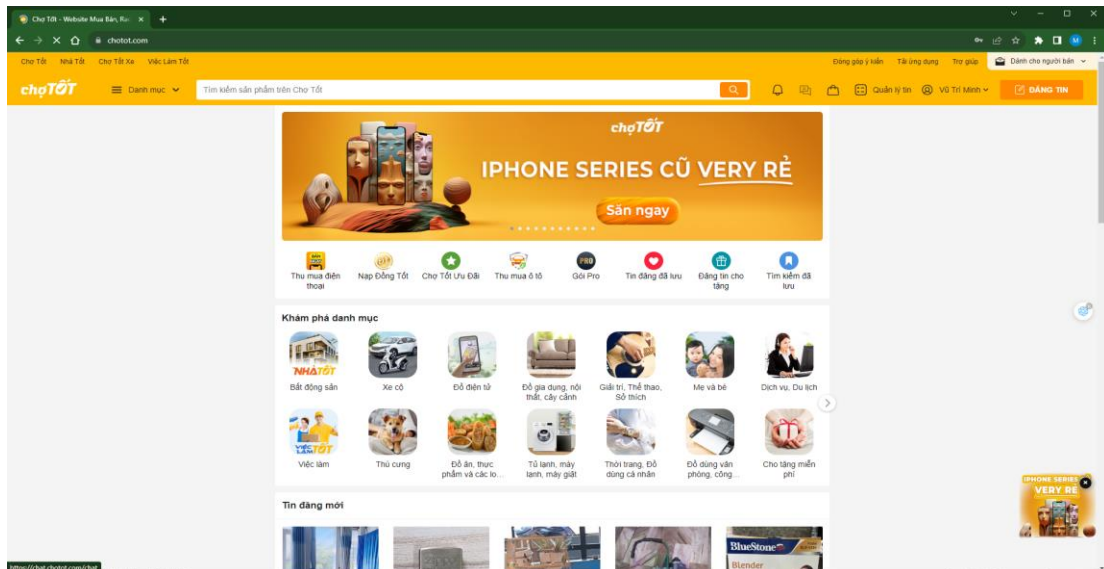
B6. Nhấn vào Đăng Nhập

- Ở bước cuối cùng, cũng giống như ở B.2 ta cũng dùng hàm sự kiện "Click" để nhấn vào ô "Đăng Nhập"
- Trước hết, ta tạo dòng lệnh mới và nhập lệnh "Click" vào ô Yêu Cầu
- Tiếp theo ta nhấp vào ô "Chọn mục tiêu trong trang" và nhấn vào nút "Đăng Nhập"

Hình 3.20

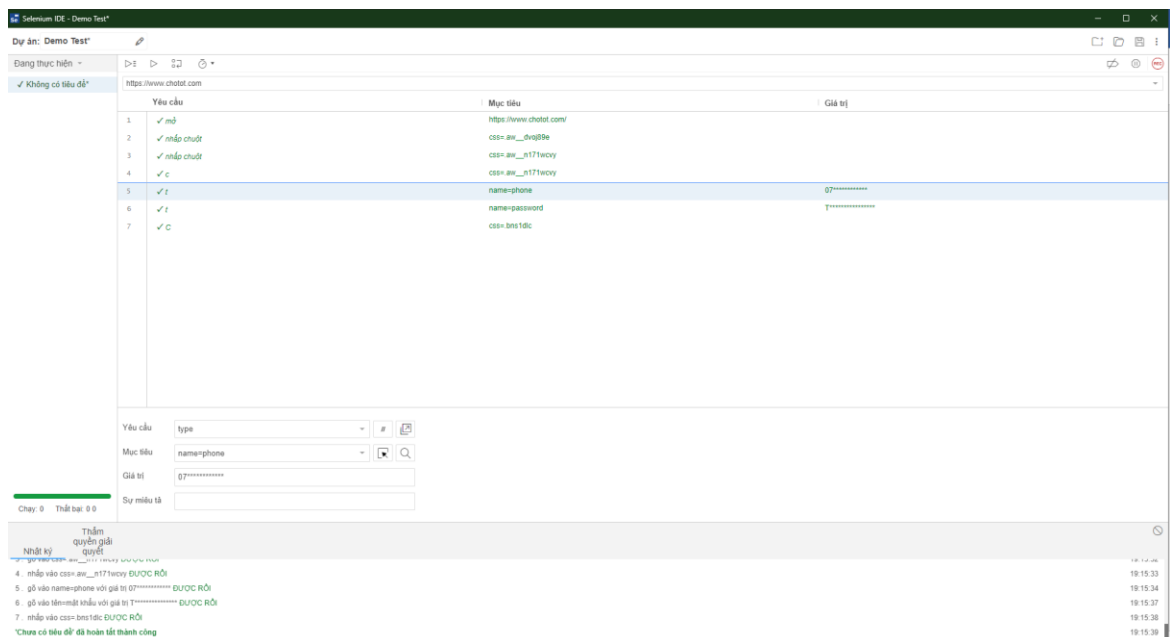
- Quay lại trình IDE, ta có địa chỉ được nhập sẵn ở ô Mục Tiêu

- Ta tiếp tục nhấn vào nút Kiểm Thử
- Sau khi chạy cả 6 Bước thì ta được chuyển hướng đến trang cá nhân trên chợ tốt



Hình 3.21

- Đây là kết quả của các bước trong Selenium IDE sau khi chạy thử thành công



Hình 3.22

- **Kết thúc kiểm thử**

3.2 Thực hiện trên Selenium Webdriver

- Kịch bản kiểm thử

B1. Mở trình duyệt Google Chrome

B2. Truy cập trang web <https://www.facebook.com/>

B3. Nhập tài khoản

B4. Nhập mật khẩu

B5. Nhấp vào nút đăng nhập

- Sau đây là các bước để tiến hành kiểm thử trên trình duyệt Google Chrome bằng Selenium Python:

B1:Mở trình duyệt Google Chrome

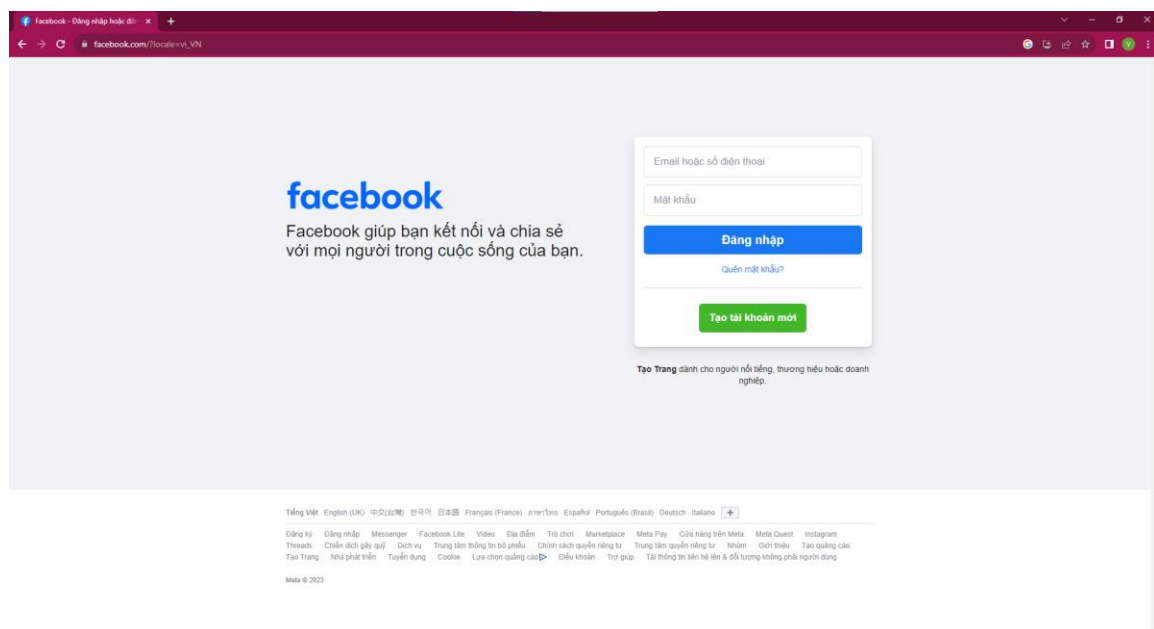
- Import thư viện cần thiết

```
from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.by import By
from selenium.common.exceptions import TimeoutException
```

B2: Truy cập trang web <https://www.facebook.com/>

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from time import sleep
driver = webdriver.Chrome()
driver.get("https://www.facebook.com/?locale=vi_VN")
```

- Gọi hàm driver.get để truy cập vào địa chỉ trang web facebook
- Sau khi chạy code ở trên



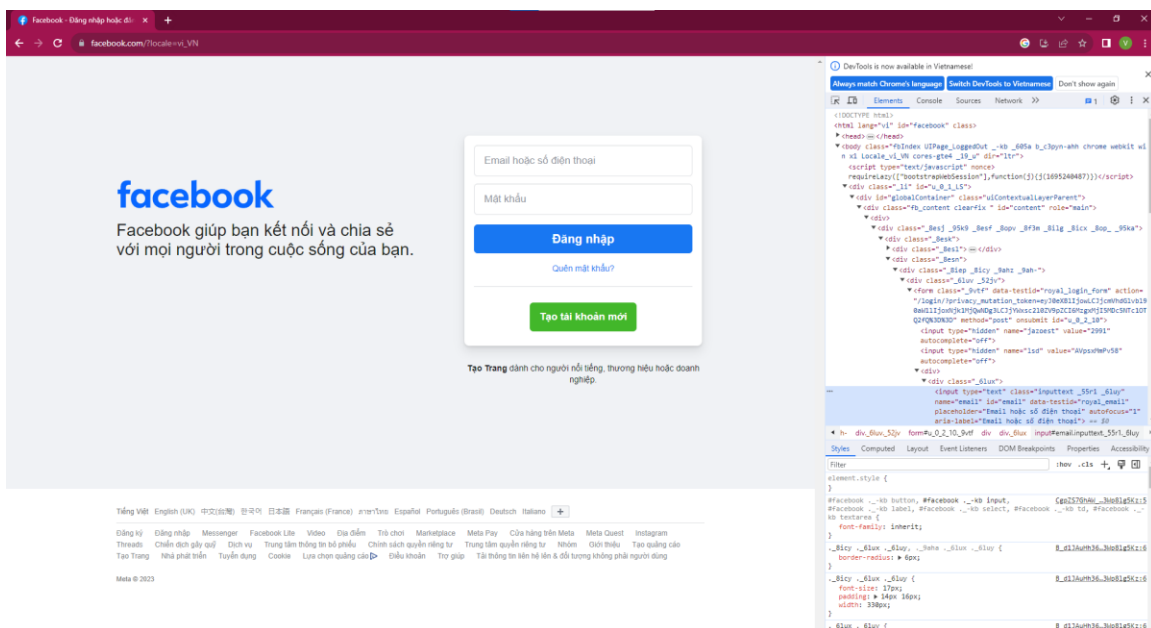
Hình 3.23

B3. Nhập tài khoản

- Đầu tiên chúng ta cần dùng câu lệnh `find_element_by_xpath` để thực hiện truy địa chỉ của nguyên tố chúng ta cần trên trang web(ở đây là ô tài khoản).

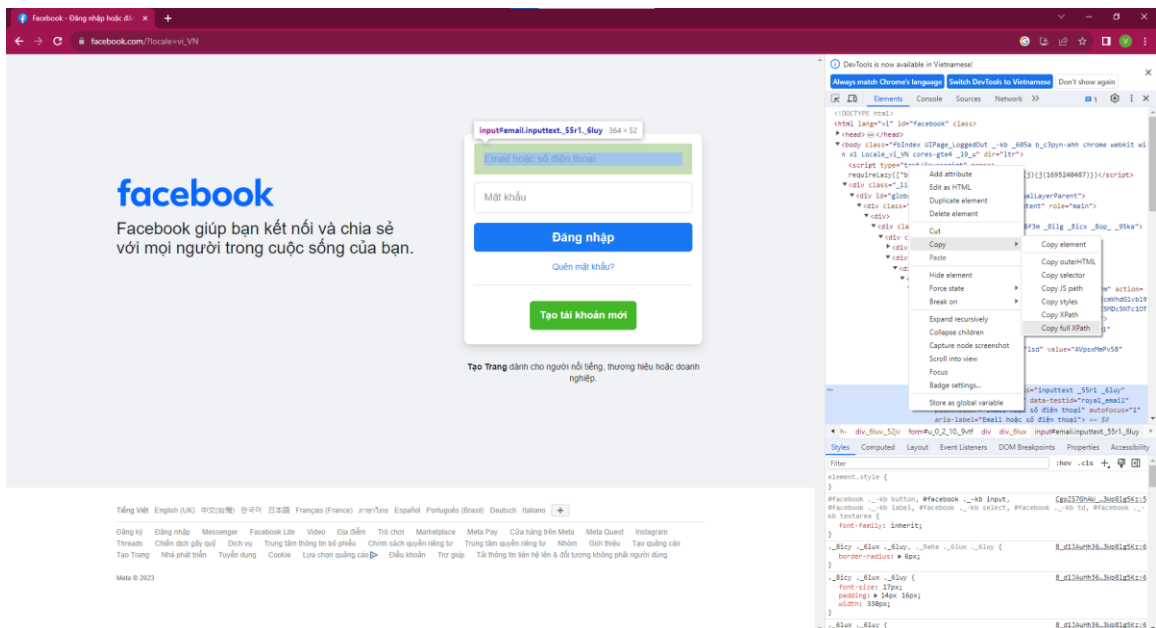
```
driver.find_element(By.XPATH('
```

- Nhấn chuột phải vào nguyên tố chúng ta cần để tìm, nhấn vào Inspect



Hình 3.24

- Đây là địa chỉ các nguyên tố trên trang web
 - Tiếp theo chúng ta nhấn vào vùng được in đậm màu xanh , nhấp chuột phải chọn Copy, chọn Copy full xpath



Hình 3.25

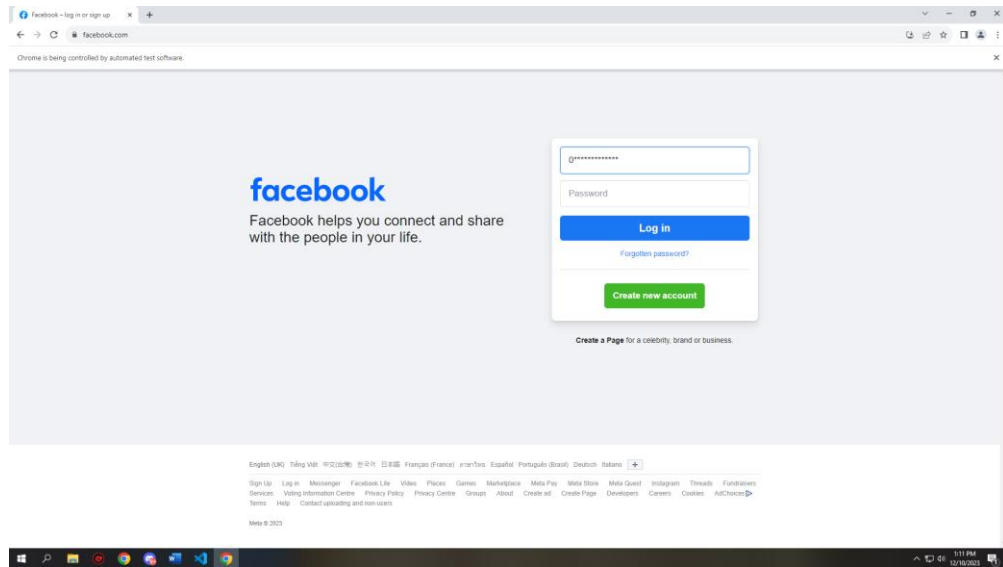
- Sau đó chúng ta vào lại phần code, nhập địa chỉ nguyên tố vào dòng lệnh

```
driver.find_element(By.XPATH('
```

- Tiếp theo chúng ta dùng hàm `send.keys` để nhập dữ liệu vào địa chỉ nguyên tố đó và nhập vào thông tin tài khoản của mình

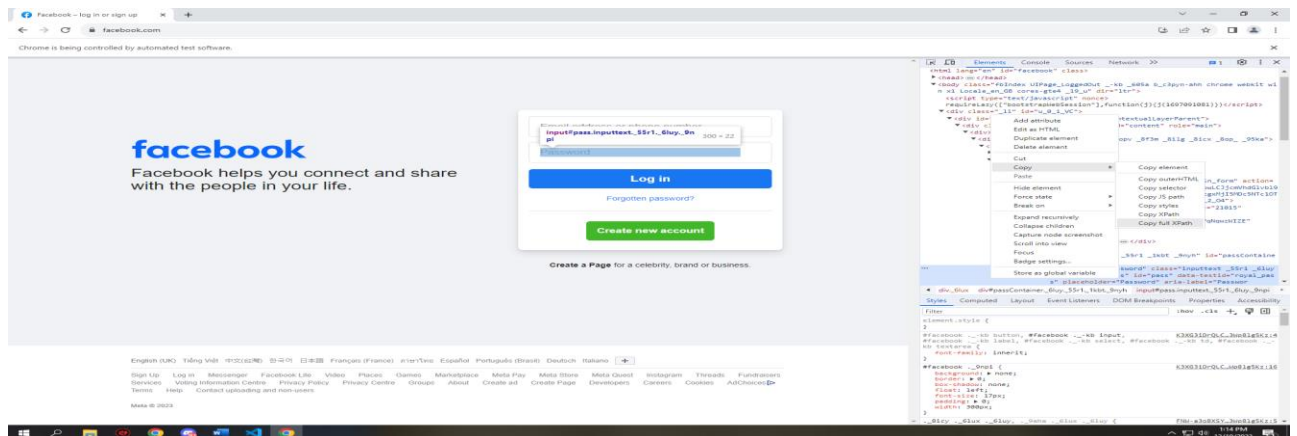
```
.send_keys('077*****')
```

- Sau khi chạy code:



Hình 3.26

- B4. Nhập mật khẩu
- Tương tự như ở B3 chúng ta vào trang web và tìm địa chỉ **XPATH** của ô mật khẩu:



Hình 3.27

- Tiếp theo ta vào phần code và nhập câu lệnh

```
driver.find_element(By.XPATH('
```

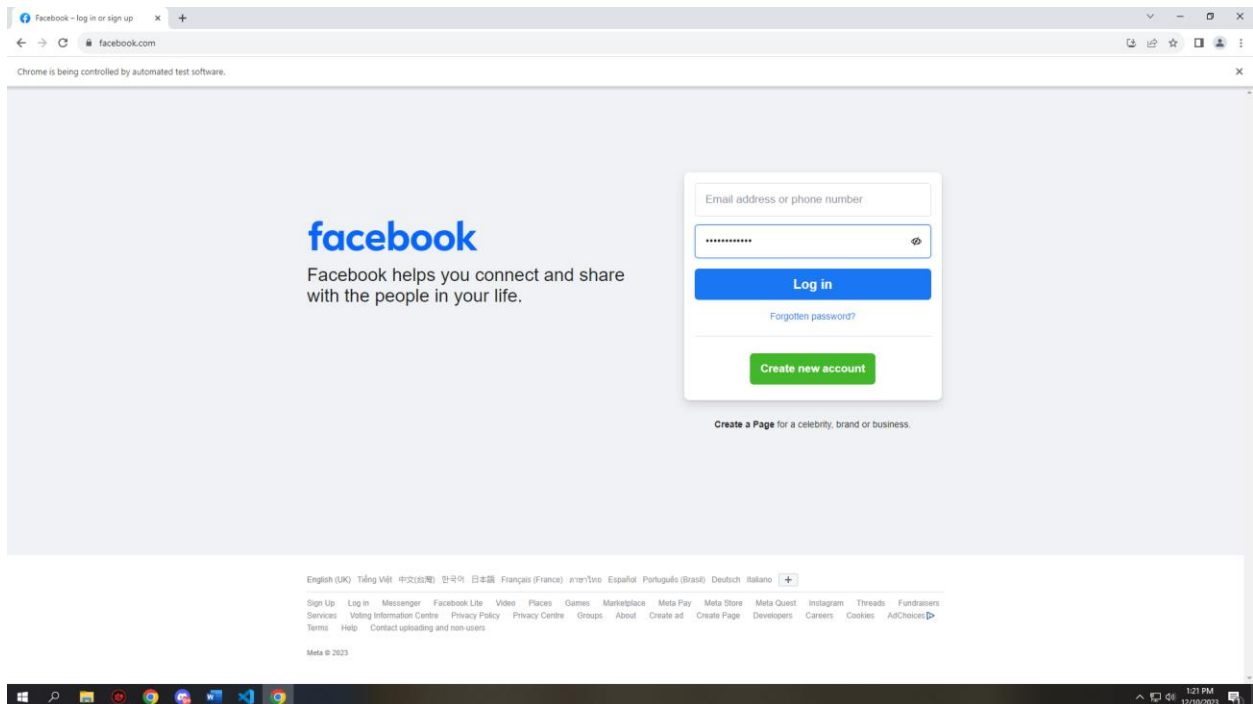
- Gán vào địa chỉ **XPATH** vừa copy được vào dòng lệnh trên ta được

```
driver.find_element(By.XPATH, '/html/body/div[1]/div[1]/div[1]/div/div/div/div[2]/div/div[1]/form/div[1]/div[2]/div/input')
```

- Sau đó ta dùng câu lệnh sendkeys để nhập mật khẩu vào ô trống

```
.send_keys('*****')
```

- Sau khi chạy code

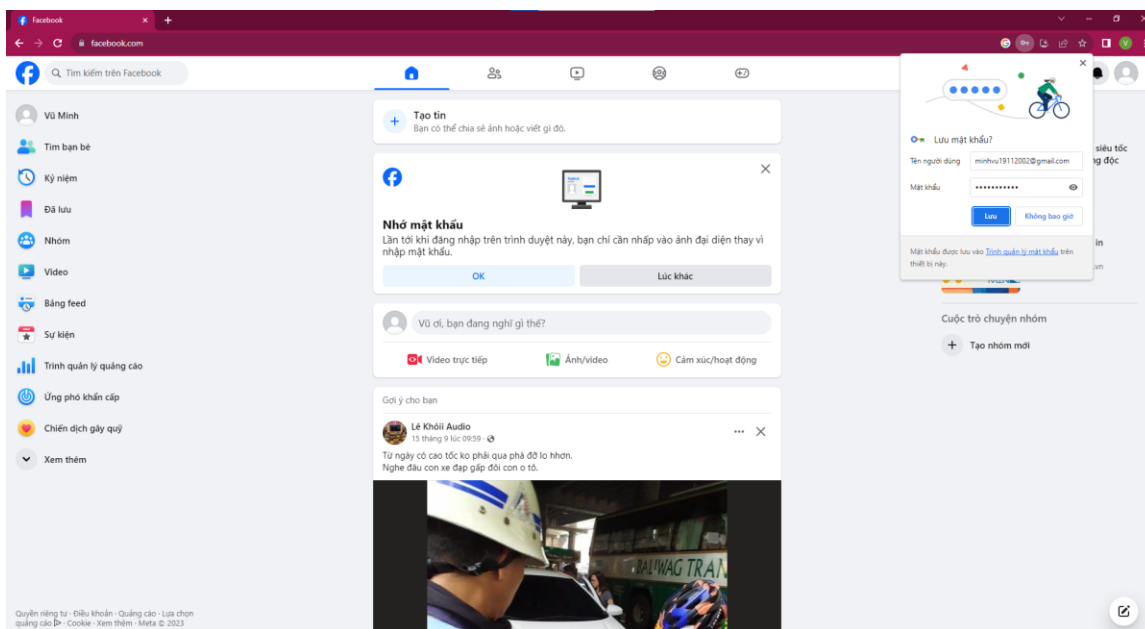


Hình 3.28

B5. Nhấp vào nút đăng nhập

- Ở bước cuối chúng ta dùng hàm "click" để áp dụng sự kiện nhấp vào 1 nguyên tố bất kì trên trang web (ở đây là nút đăng nhập)

- Tương tự như ở trên ta cũng dùng câu lệnh `driver.find_element(By.XPATH, để tìm địa chỉ nguyên tố và nhấn vào ô Đăng nhập để Copy địa chỉ Xpath dán vào dòng lệnh trên, tiếp tục ở câu lệnh chúng ta nhập hàm sự kiện click .click() để nhấp vào ô Đăng nhập.`
- Sau khi chạy code tất cả các bước trên thì ta sẽ vào được trang cá nhân facebook của mình



Hình 3.29

Chương 4: KẾT QUẢ ĐẠT ĐƯỢC

4.1. Kết quả đạt được của đề tài

- Nắm được lý thuyết tổng quan về kiểm thử tự động, kiểm thử ứng dụng web, công cụ kiểm thử tự động Selenium.

- Sử dụng công cụ Selenium WebDriver, Selenium IDE, kiểm tra các chức năng của website shopee, facebook, chotot, youtube...
- Tạo ra bộ testcase dựa trên bản đặc tả của website Shopee, Facebook, Youtube.. và đưa ra kết quả mong đợi và kết quả thực tế trong kiểm thử.
- Xây dựng kịch bản kiểm thử và bộ dữ liệu kiểm tra cho một số chức năng của website Shopee, Facebook, Youtube...

4.2. Hạn chế của đề tài

- Chưa kiểm thử hết chức năng của website.
- Chưa tìm hiểu hết bộ công cụ kiểm thử tự động Selenium.

4.3. Hướng phát triển của đề tài

- Tiếp tục tìm hiểu về kiểm thử tự động, kiểm thử ứng dụng website, đặc biệt là bộ công cụ Selenium. Trong tương lai, có thể vận dụng vào kiểm thử các ứng dụng thực tế trong công việc.
- Vận dụng Selenium để tối đa tự động hóa kiểm thử phần mềm.
- Tìm ra những hạn chế để khắc phục, hoàn thiện bộ công cụ khi sử dụng test hơn.

Chương 5: DEMO

5.1 Kịch bản demo

5.1.1 Selenium IDE

- Kịch bản kiểm thử

- B1. Đăng nhập trang web <https://shopee.vn/>
- B2. Xem thông báo
- B3. Tìm sản phẩm "Mô Hình"
- B4. Kiểm tra phần tử có mặt
- B5. Tăng số lượng sản phẩm
- B6. Thêm sản phẩm vào giỏ hàng
- B7. Xóa sản phẩm khỏi giỏ hàng
- B8. Đăng xuất

5.1.2 Selenium Webdriver

- Kịch bản kiểm thử
 - B1. Mở trình duyệt Google Chrome
 - B2. Truy cập trang web <https://www.youtube.com/>
 - B3. Mở một trang web khác trong tab mới
 - B4. Lấy danh sách id của tất cả các tab
 - B5. Chuyển đổi đến cửa sổ đầu tiên
 - B7. Điền giá trị vào ô tìm kiếm
 - B8. Nhấn vào icon "Tìm kiếm"
 - B9. Nhấp vào xem video "Hướng dẫn sử dụng Selenium IDE để làm Automation"
 - B10. Nhấp vào xem video tiếp theo