Logo

Description automatically generated

**TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**----🙠🕮🙢----**

**BÀI BÁO CÁO KẾT THÚC HỌC PHẦN**

**NĂM HỌC 2022-2023**

**Môn: ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM**

**Đề tài: AUTOMATION TESTING WEBSITE BÁN ĐIỆN THOẠI BẰNG SELENIUM**

**GV hướng dẫn: Đỗ Đức Bích Ngân**

|  |  |
| --- | --- |
| Nhóm : 11 | |
| Tên thành viên | **Mã số sinh viên** |
| Nguyễn Anh Nhật | 20DH110477 |
| Ngô Văn Phong | 20DH112142 |
| Hoàng Hải | 20DH111972 |
| Trần Thị Lan Nguyên | 20DH111117 |

**Thành phố Hồ Chí Minh**

# Nhận xét của giáo viên hướng dẫn

# LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập đến nay, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô, gia đình và bạn bè. Với lòng biết ơn sâu sắc nhất, chúng em xin gửi đến quý Thầy Cô ở Khoa Công nghệ Thông tin - Trường Đại học Ngoại ngữ - Tin học Tp. Hồ Chí Minh đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Và đặc biệt, trong học kỳ này, Khoa đã tổ chức cho chúng em được tiếp cận với môn học mà theo chúng em là rất hữu ích đối với sinh viên ngành Công nghệ phần mềm cũng như tất cả các sinh viên thuộc các chuyên ngành khác. Chúng em xin chân thành cảm ơn cô Đỗ Đức Bích Ngân đã tận tâm hướng dẫn chúng em qua từng buổi học trên lớp. Nếu không có những lời hướng dẫn, dạy bảo của cô thì bài báo cáo này của chúng em rất khó có thể hoàn thiện được. Một lần nữa, chúng em xin chân thành cảm ơn cô. Chúng em xin kính chúc quý Thầy Cô trong Khoa Công nghệ Thông tin thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau

# MỤC LỤC

Mục lục

[Nhận xét của giáo viên hướng dẫn 1](#_Toc153574239)

[LỜI CẢM ƠN 2](#_Toc153574240)

[MỤC LỤC 3](#_Toc153574241)

[BẢNG PHÂN CÔNG CÔNG VIỆC 5](#_Toc153574242)

[CHƯƠNG 1. GIỚI THIỆU CHUNG 7](#_Toc153574243)

[CHƯƠNG 2. LÝ THUYẾT 8](#_Toc153574244)

[2.1 Kiểm thử phần mềm 8](#_Toc153574245)

[2.2 Kiểm thử tự động 8](#_Toc153574246)

[2.3 Tầm quan trọng và lợi ích 8](#_Toc153574247)

[2.3.1 Tầm quan trọng: 8](#_Toc153574248)

[2.3.2 Lợi ích: 8](#_Toc153574249)

[2.4 Selenium 8](#_Toc153574250)

[2.5 Ưu điểm của Selenium: 9](#_Toc153574251)

[2.6 Hạn chế của Selenium: 10](#_Toc153574252)

[2.7 Lý do chọn Selenium cho việc kiểm thử phần mềm của nhóm: 10](#_Toc153574253)

[CHƯƠNG 3. KIỂM THỬ AUTOMATION HỆ THỐNG ABC 11](#_Toc153574254)

[3.1 Đặc tả các chức năng hệ thống (System Functions) 11](#_Toc153574255)

[3.2 Xác định mục tiêu, các chuẩn chất lượng cần đạt của hệ thống 12](#_Toc153574256)

[3.2.1 Xác định mục tiêu: 12](#_Toc153574257)

[3.3 Phạm vi kiểm thử của hệ thống (Scope) 14](#_Toc153574258)

[3.4 Thiết kế test scenario/ checklists 15](#_Toc153574259)

[3.4.1 Kiểm thử chức năng 15](#_Toc153574260)

[3.4.2 Kiểm thử phi chức năng 16](#_Toc153574261)

[3.5 Thực thi automations 16](#_Toc153574262)

[3.6 Hoàng Hải 16](#_Toc153574263)

[3.7 Văn Phong 27](#_Toc153574264)

[3.8 Anh Nhật 72](#_Toc153574265)

[3.9 Lan Nguyên 90](#_Toc153574266)

[CHƯƠNG 4. KẾT LUẬN 97](#_Toc153574267)

[4.1 Tổng kết 97](#_Toc153574268)

[4.2 Các thuận lợi khi test auto 97](#_Toc153574269)

[4.3 Các khó khăn 98](#_Toc153574270)

[4.4 Hướng phát triển nếu có thời gian 98](#_Toc153574271)

[CHƯƠNG 5. TÀI LIỆU THAM KHẢO 99](#_Toc153574272)

# BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |
| --- | --- | --- | --- |
| Họ tên | Công việc | Mức độ hoàn thành | % Công việc |
| Ngô Văn Phong | - Quản lý hệ điều hành PN  + Thêm  + Xóa  + Sửa  + Chi tiết sp  - Quản lý người dùng  + Xóa  + Chi tiết | 100% | 25% |
| Nguyễn Anh Nhật | - Quản lý sản phẩm  + Thêm  + Xóa  + Sửa  + Chi tiết sp  - Quản lý hãng sản xuất  + Thêm  + Xóa  + Sửa  + Chi tiết  Báo cáo | 100% | 25% |
| Trần Thị Lan Nguyên | - Đăng ký  - Đăng nhập  - Đăng xuất  - Chi tiết sản phẩm  - Giỏ hàng  - Đặt hàng ( có phương thức thanh toán)  - Đơn hàng | 100% | 25% |
| Hoàng Hải | Đăng nhập  - Đăng xuất  - Quản lý đơn hàng  + Chi tiết  - Thống kê ( Khi user đặt hàng thì số lương đơn hàng sẽ tăng) | 100% | 25% |
|  |  |  |  |

# GIỚI THIỆU CHUNG

* Trong những năm gần đây, công nghệ thông tin đã phát triển nhanh chóng, mạnh mẽ và được ứng dụng ở hầu hết các mặt đời sống, kinh tế, văn hóa, xã hội. Cùng với đó các ngành công nghệ phần mềm cũng phát triển theo, việc xây dựng các trang web hay các ứng dụng ngày càng phổ biến, gia tăng về số lượng theo quy mô mở rộng và cả về chất lượng phần mềm theo chiều sâu. Nhưng cũng từ đó cũng kèm theo lỗi hỏng hóc phần mềm không đáng có gây ra các ảnh hưởng đến việc sử dụng của sản phẩm của người dùng. Những lỗi này là do phầm mềm bị hỏng do không được kiểm định, kiểm duyệt kỹ lưỡng trước khi đưa cho người dùng cuối hoặc do những sự phá hoại nhằm đánh cắp thông tin cá nhân như số điện thoại, tin nhắn, danh bạ, mã số tài khoản ngân hàng,… Những vấn đề nan giải này ngày càng có xu hướng mở rộng trong những năm gần đây, cho thấy được là mặc dù phần mềm phát triển 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 phải có đội ngũ 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 chúng ta chưa kịp nhận ra. Tuy nhiên vì phần mềm ngày càng lớn, với hàng nghìn module, có thể do công ty cả công ty hàng nghìn người phát triển vì vậy để kiểm thử được một phần mềm lớn như vậy sẽ tốn nhiều công sức và thời gian nếu làm thủ công, chưa kể đến chất lượng kiểm đinh sẽ không cao và thật chính xác phù hợp cho yêu cầu. Theo nhiều tính toán thì việc kiểm thử đóng vai trò hết sức quan trọng trong quy trình phát triển 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 chống và độ an toàn chính xác cao nhất có thể. Chính vì thế nên các tool kiểm thử được sử dụng như một chiếc phao giúp cho đội ngũ kiểm định đỡ được phần nào sự mệt mõi và dễ dàng trong công việc kiểm định chất lượng của phần mềm.

# LÝ THUYẾT

## Kiểm thử phần mềm

Kiểm thử phần mềm là quá trình đánh giá và đảm bảo chất lượng của phần mềm thông qua việc kiểm tra, xác nhận tính đúng đắn, hoạt động và tuân thủ các yêu cầu.

## Kiểm thử tự động

Kiểm thử tự động là việc sử dụng công cụ và kịch bản để thực hiện các bước kiểm thử một cách tự động, giúp tăng tốc độ, hiệu suất và đồng nhất hóa quy trình kiểm thử.

## Tầm quan trọng và lợi ích

### Tầm quan trọng:

* Đảm bảo chất lượng: Giúp phát hiện lỗi và vấn đề trong phần mềm trước khi ra mắt, giúp đảm bảo tính ổn định và đáng tin cậy.
* Tiết kiệm thời gian và chi phí: Loại bỏ lỗi sớm giúp tránh chi phí sửa chữa lớn sau này và tối ưu hóa quá trình phát triển.
* Tăng cường uy tín: Phần mềm được kiểm thử kỹ lưỡng mang lại niềm tin từ người dùng.

### Lợi ích:

* Tăng tốc độ: Thực hiện kiểm thử nhanh chóng và lặp lại hàng loạt bằng cách sử dụng kịch bản tự động.
* Tăng độ chính xác: Loại bỏ yếu tố con người và giảm sai sót trong quá trình kiểm thử.
* Tiết kiệm chi phí: Giảm chi phí lao động và tăng hiệu suất.

## Selenium

Selenium là một công cụ phổ biến và mạnh mẽ được sử dụng rộng rãi cho việc kiểm thử tự động các ứng dụng web. Selenium cung cấp một bộ công cụ để tạo và thực thi các test case, gồm:

* **Selenium WebDriver:** Là một API cho phép giao tiếp trực tiếp với trình duyệt web để điều khiển hành vi của trang web. Nó hỗ trợ nhiều ngôn ngữ lập trình như Java, Python, C#, Ruby, JavaScript, v.v
* **Selenium IDE:** Là một công cụ mở rộng trình duyệt cho phép ghi lại và tái sử dụng các hành động trên trình duyệt web dưới dạng test case.
* **Selenium RC:** Selenium Remote Control, Selenium server sẽ khởi chạy và tương tác với các trình duyệt web.
* **Selenium Grid:** Selenium Hub được sử dụng để khởi chạy nhiều các test thông qua các máy cũng như các trình duyệt khác nhau tại cùng một thời điểm nhất định.

## Ưu điểm của Selenium:

1. **Tích hợp dễ dàng:** Selenium có thể tích hợp với các công cụ và framework kiểm thử khác nhau như TestNG, JUnit, và các công cụ CI/CD như Jenkins.
2. **Hỗ trợ đa nền tảng và đa trình duyệt:** Selenium hỗ trợ kiểm thử trên nhiều trình duyệt như Chrome, Firefox, Safari, Edge và Opera trên các nền tảng Windows, macOS và Linux.
3. **Ngôn ngữ lập trình linh hoạt:** Selenium hỗ trợ nhiều ngôn ngữ lập trình như Java, Python, C#, Ruby, JavaScript, PHP, v.v.
4. **Hỗ trợ kiểm thử ứng dụng web:** Selenium mạnh mẽ trong việc kiểm thử các ứng dụng web, cho phép tương tác với các phần tử trên trang web và thực hiện các test case một cách tự động.
5. **Quá trình cài đặt và sử dụng rất đơn giản**
6. **Yêu cầu cấu hình phần cứng thấp**
7. **Tạo ra bộ test lớn với nhiều bài test nhỏ để tiết kiệm thời gian**
8. **Hỗ trợ kiểm thử tự động thay cho người dùng thật giúp tiết kiệm nhiều nhân lực**

## Hạn chế của Selenium:

1. **Không hỗ trợ kiểm thử ứng dụng di động natively:** Selenium không thể kiểm thử ứng dụng di động (iOS, Android) trực tiếp mà cần sử dụng các công cụ bổ sung như Appium.
2. **Khả năng ổn định:** Có thể gặp phải một số vấn đề về ổn định và độ tin cậy trong việc thao tác với các phần tử trên trang web, đặc biệt khi có sự thay đổi trong cấu trúc của trang web.
3. **Khả năng tương thích:** Có thể xuất hiện các vấn đề tương thích khi Selenium tương tác với các phiên bản trình duyệt mới, có thể cần cập nhật hoặc chờ đợi các phiên bản mới của Selenium WebDriver để sử dụng được.
4. **Khả năng hiển thị thông tin không đồng nhất:** Có thể xảy ra tình huống mà các trình duyệt hiển thị thông tin một cách khác nhau, dẫn đến việc viết test script có thể không nhất quán trên các trình duyệt khác nhau.
5. **Khó khăn trong việc xử lý các yêu cầu đa phương tiện:** Selenium gặp khó khăn trong việc xử lý các yêu cầu liên quan đến đa phương tiện như video, âm thanh, hoặc đồ họa động.

## Lý do chọn Selenium cho việc kiểm thử phần mềm của nhóm:

Ở đồ án kiểm thử website bán điện thoại lần này thì chúng em sẽ tiến hành kiểm tra kĩ lưỡng về mặt giao diện bên ngoài của Website, phần nhập liệu thông tin, phần code bên trong và nhiều thứ khác vì vậy với những lợi ích của selenium và cũng đã được hướng dẫn trên lớp nên chúng em quyết định chọn selenium cho đồ án lần này.

# KIỂM THỬ AUTOMATION HỆ THỐNG ABC

## Đặc tả các chức năng hệ thống (System Functions)

Nhằm đáp ứng nhu cầu ứng dụng công nghệ thông tin cho các hoạt động kinh doanh và để giảm bớt thời gian tìm kiếm, theo dõi hoạt động website trong việc quản lý. Website ra đời nhằm đáp ứng vấn đề sau:

Khách hàng:

* Xem thông tin các sản phẩm
* Tìm kiếm thông tin các sản phẩm
* Xem thông tin giỏ hàng
* Chỉnh sửa thông tin giỏ hàng
* Xác nhận đặt hàng
* Xem thông tin chi tiết đơn hàng

Admin:

* Quản lý sản phẩm ( thêm, sửa, xóa thông tin các sản phẩm )
* Quản lý nhà sản xuất ( thêm, sửa, xóa thông tin các nhà sản xuất )
* Quản lý thể loại ( thêm, sửa, xóa thông tin thể loại )
* Quản lý khách hàng ( chỉnh sửa, xóa thông tin khách hàng )
* Quản lý tin tức ( Thêm, sửa, xóa tin tức )
* Quản lý đơn hàng ( Thêm, sửa, xóa đơn hàng )

## Xác định mục tiêu, các chuẩn chất lượng cần đạt của hệ thống

### Xác định mục tiêu:

* Website phải đảm bảo đúng chức năng với đặc tả và thiết kế hệ thống, đáp ứng yêu cầu của người dùng
* Các chức năng phải đầy đủ và hoạt động bình thường
* Kiểm thử giao diện của website
* Sau khi trải qua kiểm thử, website không được tồn tại lỗi quan trọng
* Hoàn thành các quy trình kiểm thử Requirenment analysis, Test planning, Test case development, …

**Yêu cầu về phía người sử dụng:**

+ Giao diện đẹp, thân thiện phù hợp với yêu cầu khách hàng

+ Dễ sử dụng với các đối tượng người dùng, thuận tiện trong quản trị, dễ bảo trì

+ Thông tin hiển thị chi tiết

+ Chạy ổn định trên các trình duyệt

+ Hoàn thành sản phẩm đúng thời gian quy định

**Yêu cầu về chức năng:**

+ Dễ dàng tùy chỉnh

+ Có tính hiệu quả cao

+ Có tính bảo mật cao

**Yêu cầu tính hữu dụng của phần mềm:**

+ Cập nhật, thêm, sửa, xóa, tra cứu các thông tin về khách hàng

+ Thống kê, báo cáo các hoạt động theo tuần, tháng/năm.

+ Giúp Khách hàng ở xa có thể đặt hàng dễ dàng.

Xác định chuẩn chất lượng đầu ra:

Website bán hàng của nhóm được xây dựng và phát triển dựa trên tiêu chuẩn của **ISO/IEC 9126,** tiêu chuẩn này trình bày một số tổ hợp chất lượng tượng trưng cho bất cứ phần mềm nào, bao gồm:

**Tính chức năng:** Khả năng của phần mềm cung cấp các chức năng đáp ứng được nhu cầu sử dụng khi phần mềm làm việc trong điều kiện cụ thể.

* Đảm bảo phân quyền tài khoản người dùng
* Tài khoản admin có thể quản lý toàn bộ thông tin của website: quản lý sản phẩm, quản lý người dùng, phân quyền người dùng,…
* Tài khoản người dùng có thể dùng để sử dụng các chức năng cơ bản của website theo nghiệp vụ đặt hàng: Xem sản phẩm, mua hàng, chỉnh sửa giỏ hàng,…

**Khả năng bảo hành, bảo trì:** Khả năng bảo hành, bảo trì: Khả năng của phần mềm có thể chỉnh sửa.

* Có thể thay đổi được: Website có khả năng nâng cấp lên các nền tảng hoặc công nghệ mới dựa theo quá trình phát triển của công nghệ.
* Có thể kiểm tra được: Website phải có khả năng cho phép tester thực hiện các quy trình kiểm thử website một cách định kỳ nhằm phát hiện ra bug sớm nhất có thể.

**Tính khả chuyển:**

* Có khả năng thích nghi: Website có khả năng sử dụng trên mọi trình duyệt và trên mọi hệ điều hành (Window, Linux, MacOS,…)

**Tính hiệu quả:**

* Đáp ứng thời gian: Các thao tác, tác vụ của người dùng như: xem sản phẩm, đặt hàng, giỏ hàng,… phải được thực hiện một cách nhanh chóng với mức độ delay thấp nhất.
* Tận dụng tài nguyên: Website phải tận dụng tài nguyên một cách triệt để, sử dụng ít Ram, sử dụng ít dung lượng Database.

**Tính tin cậy:** Là khả năng của phần mềm có thể hoạt động ổn định trong những điều kiện cụ thể.

* Khả năng chịu lỗi: khả năng của phần mềm hoạt động ổn định tại một mức độ cả trong trường hợp có lỗi xảy ra ở phần mềm hoặc có những vi phạm trong giao diện.
* Khả năng phục hồi: khả năng của phần mềm có thể tái thiết lại hoạt động tại một mức xác định và khôi phục lại những dữ liệu có liên quan trực tiếp đến lỗi.
* Tính tuân thủ tin cậy: phần mềm thoả mãn các chuẩn, quy ước, quy định.

**Tính khả dụng:** Là khả năng của phần mềm có thể hiểu được, học được, sử dụng được và hấp dẫn người sử dụng trong từng trường hợp sử dụng cụ thể.

* Có thể hiểu được: người sử dụng có thể hiểu được xem phần mềm có hợp với họ không và và sử dụng chúng thế nào cho những công việc cụ thể.
* Có thể học được: người sử dụng có thể học các ứng dụng của phần mềm.
* Có thể sử dụng được: khả năng của phần mềm cho phép người sử dụng sử dụng và điều khiển nó.
* Tính hấp dẫn: khả năng hấp dẫn người sử dụng của phần mềm.
* Tính tuân thủ khả dụng: phần mềm thoả mãn các chuẩn, quy ước, quy định.

## Phạm vi kiểm thử của hệ thống (Scope)

* Xác định Scope (phạm vi) của dự án cần test là một điều tối quan trọng trong quá trình kiểm thử một phần mềm hay một website nào đó. Việc này cung cấp cho bạn các thông tin về project mà bạn sẽ test trong tương lai. Điều đó giúp cho toàn bộ thành viên tham gia dự án của bạn đều sẽ biết trước được các function, các GUI hay các tác vụ cần test
* Ở đồ án lần này với website bán điện thoại thì nhóm đã đặt ra phạm vi kiểm thử như sau:
* Đăng nhập (Test các form nhập thông tin)
* Đăng ký (Test các form nhập thông tin)
* Giỏ hàng (Test các chức năng: thêm, sửa, xóa số lượng các sản phẩm hiện có trong giỏ hàng, xem thông tin giỏ hàng)
* Đơn hàng (Test các chức năng: xem thông tin các đơn hàng đã được đặt hàng bởi người dùng)
* Quản lý sản phẩm (Test các chức năng: thêm, sửa, xóa các sản phẩm mới trong cửa hàng, các form nhập liệu thông tin khi thêm hoặc sửa thông tin sản phẩm)
* Quản lý nhà sản xuất (Test các chức năng: thêm, sửa, xóa các hãng sản xuất của, các form nhập liệu thông tin khi thêm hoặc sửa thông tin hãng sản xuất)
* Quản lý thể loại (Test các chức năng: thêm, sửa, xóa các form nhập liệu thông tin khi thêm hoặc sửa thông tin hệ điều hành)
* Quản lý người dùng (Test các chức năng quản lý tài khoản người dùng, xóa người dùng hoặc chỉnh sửa thông tin người dùng, các form nhập liệu khi chỉnh sửa thông tin người dùng)

## Thiết kế test scenario/ checklists

### Kiểm thử chức năng

**User**

* Đăng ký
* Đăng nhập
* Đăng xuất
* Chi tiết sản phẩm
* Giỏ hàng
* Đặt hàng ( có phương thức thanh toán)
* Đơn hàng

**\* Admin**

* Quản lý sản phẩm
* Thêm
* Xóa
* Sửa
* Chi tiết sp
* Quản lý hãng sản xuất
* + Thêm
* + Xóa
* + Sửa
* + Chi tiết
* Quản lý hệ điều hành
* Thêm
* Xóa
* Sửa
* Chi tiết sp
* Quản lý người dùng
* Xóa
* Chi tiết
* Đăng nhập
* Đăng xuất
* Quản lý đơn hàng
* Chi tiết
* Thống kê ( Khi user đặt hàng thì số lương đơn hàng sẽ tăng)

### Kiểm thử phi chức năng

## Thực thi automations

## Hoàng Hải

using Microsoft.VisualStudio.TestTools.UnitTesting;

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

using OpenQA.Selenium.Support.UI;

using System;

using System.Threading;

using Excel = Microsoft.Office.Interop.Excel;

namespace UnitTestProject6

{

[TestClass]

public class UnitTest1

{

IWebDriver driver;

Excel.Application dataApp;

Excel.Workbook dataworkbook;

Excel.Worksheet dataworksheet;

Excel.Range xlrange;

[TestInitialize]

public void Init()

{

driver = new ChromeDriver();

driver.Url = "https://localhost:44322/";

driver.Navigate();

// Phóng to màn hình

driver.Manage().Window.Maximize();

Thread.Sleep(2000);

dataApp = new Excel.Application();

// Lấy file excel

dataworkbook = dataApp.Workbooks.Open(@"D:\\testcase.xlsx");

// Lấy sheet của excel

dataworksheet = dataworkbook.Sheets[3];

// Láy giữ liệu của cột, hàng

xlrange = dataworksheet.UsedRange;

}

//[TestMethod]

//public void DangNhapUser()

//{

// driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[3]/a[2]")).Click();

// String s1 = xlrange.Cells[6][4].value.ToString();

// String s2 = xlrange.Cells[6][5].value.ToString();

// driver.FindElement(By.Id("userMail")).SendKeys(s1);

// driver.FindElement(By.Id("password")).SendKeys(s2);

// driver.FindElement(By.XPath("/html/body/div[2]/form/div/div[3]/div/input")).Click();

// //Thread.Sleep(10000);

//}

//[TestMethod]

//public void DangXuatUser()

//{

// driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[3]/a[2]")).Click();

// String s1 = xlrange.Cells[6][3].value.ToString();

// String s2 = xlrange.Cells[6][4].value.ToString();

// driver.FindElement(By.Id("userMail")).SendKeys(s1);

// driver.FindElement(By.Id("password")).SendKeys(s2);

// driver.FindElement(By.XPath("/html/body/div[2]/form/div/div[3]/div/input")).Click();

// Thread.Sleep(3000);

// driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[3]/a[2]")).Click();

// Thread.Sleep(3000);

//}

[TestMethod]

public void DangNhapAdmin()

{

driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[3]/a[2]")).Click();

String s1 = xlrange.Cells[6][12].value.ToString();

String s2 = xlrange.Cells[6][13].value.ToString();

driver.FindElement(By.Id("userMail")).SendKeys(s1);

driver.FindElement(By.Id("password")).SendKeys(s2);

driver.FindElement(By.XPath("/html/body/div[2]/form/div/div[3]/div/input")).Click();

Thread.Sleep(10000);

}

[TestMethod]

public void DangNhapSaiAdmin()

{

driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[3]/a[2]")).Click();

String s1 = xlrange.Cells[6][9].value.ToString();

String s2 = xlrange.Cells[6][10].value.ToString();

driver.FindElement(By.Id("userMail")).SendKeys(s1);

driver.FindElement(By.Id("password")).SendKeys(s2);

driver.FindElement(By.XPath("/html/body/div[2]/form/div/div[3]/div/input")).Click();

Thread.Sleep(10000);

}

[TestMethod]

public void DonHangAdmin()

{

//Đăng nhập Admin

driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[3]/a[2]")).Click();

String s1 = xlrange.Cells[6][12].value.ToString();

String s2 = xlrange.Cells[6][13].value.ToString();

driver.FindElement(By.Id("userMail")).SendKeys(s1);

driver.FindElement(By.Id("password")).SendKeys(s2);

driver.FindElement(By.XPath("/html/body/div[2]/form/div/div[3]/div/input")).Click();

Thread.Sleep(3000);

// Đơn hàng

driver.FindElement(By.XPath("/html/body/div/div[1]/nav/div[1]/div/div[7]/a")).Click();

Thread.Sleep(3000);

driver.FindElement(By.XPath("/html/body/nav/ul/li/a")).Click();

Thread.Sleep(1000);

driver.FindElement(By.XPath("/html/body/nav/ul/li/div/a[2]")).Click();

Thread.Sleep(1000);

// Đăng nhập user

driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[3]/a[2]")).Click();

String s3 = xlrange.Cells[6][15].value.ToString();

String s4 = xlrange.Cells[6][16].value.ToString();

driver.FindElement(By.Id("userMail")).SendKeys(s3);

Thread.Sleep(1000);

driver.FindElement(By.Id("password")).SendKeys(s4);

Thread.Sleep(1000);

driver.FindElement(By.XPath("/html/body/div[2]/form/div/div[3]/div/input")).Click();

Thread.Sleep(1000);

// Mua hàng

driver.FindElement(By.Id("bgnGioHang")).Click();

Thread.Sleep(3000);

// Tra cứu giỏ hàng

driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[2]/a")).Click();

Thread.Sleep(3000);

// Đặt hàng

driver.FindElement(By.XPath("/html/body/div[2]/table/tbody/tr[3]/td[2]/form/input")).Click();

Thread.Sleep(3000);

// Nhập địa chỉ, đặt hàng

driver.FindElement(By.XPath("/html/body/div[2]/form/div/div[4]/div[2]/input")).SendKeys("CMT8");

Thread.Sleep(2000);

driver.FindElement(By.XPath("/html/body/div[2]/form/div/div[7]/div/input")).Click();

Thread.Sleep(2000);

driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[3]/a[2]")).Click();

Thread.Sleep(2000);

//Đăng nhập Admin

driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[3]/a[2]")).Click();

String s5 = xlrange.Cells[6][12].value.ToString();

String s6 = xlrange.Cells[6][13].value.ToString();

driver.FindElement(By.Id("userMail")).SendKeys(s5);

driver.FindElement(By.Id("password")).SendKeys(s6);

driver.FindElement(By.XPath("/html/body/div[2]/form/div/div[3]/div/input")).Click();

Thread.Sleep(3000);

// Đơn hàng

driver.FindElement(By.XPath("/html/body/div/div[1]/nav/div[1]/div/div[7]/a")).Click();

Thread.Sleep(3000);

// Xác nhận đơn hàng vừa đặt

driver.FindElement(By.XPath("/html/body/div/div[2]/main/div/table/tbody/tr[26]/td[6]/button/a")).Click();

Thread.Sleep(3000);

driver.FindElement(By.XPath("/html/body/div/div[2]/main/div/p/button/a")).Click();

Thread.Sleep(3000);

}

[TestMethod]

public void ThongKeAdmin()

{

// Đăng nhập Admin

driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[3]/a[2]")).Click();

String s3 = xlrange.Cells[6][12].value.ToString();

String s4 = xlrange.Cells[6][13].value.ToString();

driver.FindElement(By.Id("userMail")).SendKeys(s3);

driver.FindElement(By.Id("password")).SendKeys(s4);

driver.FindElement(By.XPath("/html/body/div[2]/form/div/div[3]/div/input")).Click();

// Thống kê

driver.FindElement(By.XPath("/html/body/div/div[1]/nav/div[1]/div/div[8]/a")).Click();

Thread.Sleep(5000);

//Đăng xuất

driver.FindElement(By.XPath("/html/body/nav/ul/li/a")).Click();

Thread.Sleep(1000);

driver.FindElement(By.XPath("/html/body/nav/ul/li/div/a[2]")).Click();

Thread.Sleep(1000);

// Đăng nhập user

driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[3]/a[2]")).Click();

String s1 = xlrange.Cells[6][15].value.ToString();

String s2 = xlrange.Cells[6][16].value.ToString();

driver.FindElement(By.Id("userMail")).SendKeys(s1);

Thread.Sleep(1000);

driver.FindElement(By.Id("password")).SendKeys(s2);

Thread.Sleep(1000);

driver.FindElement(By.XPath("/html/body/div[2]/form/div/div[3]/div/input")).Click();

Thread.Sleep(1000);

// Mua hàng

driver.FindElement(By.Id("bgnGioHang")).Click();

Thread.Sleep(3000);

// Tra cứu giỏ hàng

driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[2]/a")).Click();

Thread.Sleep(3000);

// Đặt hàng

driver.FindElement(By.XPath("/html/body/div[2]/table/tbody/tr[3]/td[2]/form/input")).Click();

Thread.Sleep(3000);

// Nhập địa chỉ, đặt hàng

driver.FindElement(By.XPath("/html/body/div[2]/form/div/div[4]/div[2]/input")).SendKeys("CMT8");

Thread.Sleep(2000);

driver.FindElement(By.XPath("/html/body/div[2]/form/div/div[7]/div/input")).Click();

Thread.Sleep(2000);

driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[3]/a[2]")).Click();

Thread.Sleep(2000);

// Đăng nhập lại Admin

driver.FindElement(By.XPath("/html/body/div[1]/div[1]/div/div[3]/a[2]")).Click();

String s6 = xlrange.Cells[6][12].value.ToString();

String s7 = xlrange.Cells[6][13].value.ToString();

driver.FindElement(By.Id("userMail")).SendKeys(s6);

driver.FindElement(By.Id("password")).SendKeys(s7);

driver.FindElement(By.XPath("/html/body/div[2]/form/div/div[3]/div/input")).Click();

Thread.Sleep(1000);

driver.FindElement(By.XPath("/html/body/div/div[1]/nav/div[1]/div/div[8]/a")).Click();

Thread.Sleep(3000);

}

[TestCleanup]

public void CleanUp()

{

dataworkbook.Save();

dataworkbook.Close();

dataApp.Quit();

driver.Quit();

}

}

}

## Văn Phong

﻿using OpenQA.Selenium;

using OpenQA.Selenium.DevTools.V118.Debugger;

using OpenQA.Selenium.Edge;

using OpenQA.Selenium.Support.UI;

using OpenQA.Selenium.Support;

using System.Collections.Generic;

using SeleniumExtras.WaitHelpers;

using NUnit.Framework.Internal;

using DocumentFormat.OpenXml.Spreadsheet;

using System.Reflection;

using DocumentFormat.OpenXml.Bibliography;

using System.Linq;

namespace TestScript

{

public class Tests

{

public IWebDriver driver;

private IWebElement btnDangNhap;

private IWebElement inGmail, inPass, btnLoginAdmin;

string filePath = "", nameSheet = "";

private ExcelOperations excel ;

string linkAdminafLogin, linkHeDieuHanh,linkNguoidung, linkAddHDH;

string linkEditHDH, linkDetailHDH, linkDeleteHDH;

[SetUp]

public void Setup()

{

var options = new EdgeOptions();

driver = new EdgeDriver(options);

driver.Manage().Window.Maximize();

driver.Navigate().GoToUrl("https://localhost:44322/Home/Index");

//setup link arrive

//link admin after login

linkAdminafLogin = "https://localhost:44322/Admin/Home";

linkHeDieuHanh = "https://localhost:44322/Admin/Hedieuhanhs";

linkNguoidung = "https://localhost:44322/Admin/Nguoidungs";

linkAddHDH = "https://localhost:44322/Admin/Hedieuhanhs/Create";

//setup arrive btn

btnDangNhap = driver.FindElement(By.CssSelector("a[href='/User/Dangnhap']"));

//set file excel

filePath = "D:\\\_doan\\DOAN\\đảm bảo chất lượng\\testScripts\\test.xlsx";

nameSheet = "Sheet1";

excel = new ExcelOperations(filePath, nameSheet);

}

[Test]

public void testGetFile()

{

//ReadFile file = new ReadFile();

//file.ReadExcelFileTest();

Assert.That(true);

}

private void arlogin(IWebDriver driver)

{

//click vào đnăg nhập

btnDangNhap.Click();

//setup element login

inGmail = driver.FindElement(By.CssSelector("#userMail"));

inPass = driver.FindElement(By.CssSelector("#password"));

btnLoginAdmin = driver.FindElement(By.CssSelector("input[value='Đăng nhập']"));

//input data

inGmail.SendKeys("Admin@gmail.com");

inPass.SendKeys("12345678");

btnLoginAdmin.Click();

}

private void userlogin(IWebDriver driver)

{

//click vào đnăg nhập

btnDangNhap.Click();

//setup element login

inGmail = driver.FindElement(By.CssSelector("#userMail"));

inPass = driver.FindElement(By.CssSelector("#password"));

btnLoginAdmin = driver.FindElement(By.CssSelector("input[value='Đăng nhập']"));

//input data

inGmail.SendKeys("test@gmail.com");

inPass.SendKeys("12345678");

btnLoginAdmin.Click();

}

private void adminArrive(IWebDriver driver, IWebElement linkArrive)

{

IWebElement btnMenu, linkClick;

btnMenu = driver.FindElement(By.CssSelector("#sidebarToggle"));

linkClick = linkArrive;

linkClick.Click();

}

[Test]

public void TestLoginAdmin()

{

arlogin(driver);

string currentUrl = driver.Url;

Assert.That(linkAdminafLogin, Is.EqualTo(currentUrl), "Admin không đăng nhập thành công. URL không chuyển đến trang Admin/Home");

Thread.Sleep(2000);

}

[Test]

public void arriveHeDieuHanh()

{

//login vào admin

arlogin(driver);

// link tới hệ điều hành

//thẻ a link Hệ điều hành

IWebElement arHedieihanh = driver.FindElement(By.CssSelector("a[href='/Admin/Hedieuhanhs']"));

adminArrive(driver, arHedieihanh);

// so sánh link hiện tại với link cần tới

string currentUrl = driver.Url;

Assert.That(linkHeDieuHanh, Is.EqualTo(currentUrl));

Thread.Sleep(2000);

}

public IList<IWebElement> getRowTableHDH(IWebDriver driver)

{

//login vào admin

arlogin(driver);

// tới giao diện hệ điều hành

IWebElement arHedieihanh = driver.FindElement(By.CssSelector("a[href='/Admin/Hedieuhanhs']"));

adminArrive(driver, arHedieihanh);

// code test từ đây

// get element

IList<IWebElement> rowTable;

IWebElement btnAdd, btnAddHDH;

IWebElement inNameHDH;

//set element

List<string> lsNameHDH = new List<string>();

rowTable = driver.FindElements(By.CssSelector("table tr"));

return rowTable;

}

public IWebElement getRow(IWebDriver driver, IList<IWebElement> rowTable, int index)

{

IWebElement collBtn = rowTable[index];

return collBtn;

}

public IWebElement getBtn(IWebDriver driver,IList<IWebElement> rowTable, int index,int btn)

{

int count = 0;

count = index;

var collBtn = rowTable[count].FindElements(By.CssSelector("td button"));

return collBtn[btn];

}

[Test]

public void testThemHDH()

{

//đợi xem nó xuất hiện k

WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

bool test = true;

string nameThods = "testThemHDH";

excel.EditDataAtCell(2,2, nameThods);

var datatest = excel.GetDataAtCell(2, 5).Trim();

Console.WriteLine("data test : " + datatest);

string nameHDHAdd = datatest;

// get element

IList<IWebElement> rowTable;

//set element

List<string> lsNameHDH = new List<string>();

rowTable = getRowTableHDH(driver);

for (int i = 1; i < rowTable.Count; i++)

{

var row = rowTable[i];

var cel = row.FindElements(By.CssSelector("td"));

lsNameHDH.Add(cel[1].Text);

Console.WriteLine(cel[0].Text + " " + cel[1].Text);

}

var lsSSold = new List<string>();

lsSSold = lsNameHDH.Where(w=>w.Contains(nameHDHAdd)).ToList();

Console.WriteLine("---------------------");

//bấm thêm HDH\

IWebElement btnAdd = driver.FindElement(By.CssSelector("a[href='/Admin/Hedieuhanhs/Create']"));

btnAdd.Click();

// đúng thì tiếp tục

wait.Until(driver => driver.Url != linkHeDieuHanh);

driver.Navigate().Refresh();

// đợi phần tử load xong

IWebElement inNameHDH = wait.Until(ExpectedConditions.ElementExists(By.CssSelector("#Tenhdh")));

IWebElement btnAddHDH = wait.Until(ExpectedConditions.ElementToBeClickable(By.CssSelector("input[value='Thêm mới']")));

inNameHDH.SendKeys(nameHDHAdd);

btnAddHDH.Click();

Console.WriteLine("đợi click");

try

{

Console.WriteLine("đã vào đây");

WebDriverWait wait1 = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

IWebElement notification = wait1.Until(ExpectedConditions.ElementExists(By.CssSelector(".field-validation-error.text-danger")));

Console.WriteLine("Thêm không thành công");

excel.EditDataAtCell(2, 8, notification.Text);

excel.EditDataAtCell(2, 9, "Failed\r\n");

}

catch (Exception)

{

wait.Until(driver => driver.Url == linkHeDieuHanh);

List<string> lsNameHDHnew = new List<string>();

var rowTableNew = driver.FindElements(By.CssSelector("table tr"));

for (int i = 1; i < rowTableNew.Count; i++)

{

var row = rowTableNew[i];

var cel = row.FindElements(By.CssSelector("td"));

lsNameHDHnew.Add(cel[1].Text);

Console.WriteLine(cel[0].Text + " " + cel[1].Text);

}

Console.WriteLine("---------------------");

var lsSSnew = new List<string>();

lsSSnew = lsNameHDHnew.Where(w => w.Contains(nameHDHAdd)).ToList();

//lsSSnew.Count - lsSSold.Count == 1 | kiểm tra list phần tử có tên vừa add thêm 1

//lsNameHDHnew.Contains(currentUrl) kiểm tra tên phần tử đã có trong list chưa

//lsNameHDHnew.Count - lsNameHDH.Count == 1: kiểm tra phần tử có sự thay đổi số lượng tăng 1

if (lsSSnew.Count - lsSSold.Count == 1 && lsNameHDHnew.Contains(nameHDHAdd) && lsNameHDHnew.Count - lsNameHDH.Count == 1)

{

Console.WriteLine("Thêm thành công");

excel.EditDataAtCell(2, 9, "Passed\r\n");

}

else

{

test = false;

}

Console.WriteLine("Kiểm tra những thằng liên quan");

var clickHome = driver.FindElement(By.CssSelector("a[href='/Admin/Home']"));

clickHome.Click();

wait.Until(driver => driver.Url == linkAdminafLogin);

IWebElement btnCreateDT = wait.Until(ExpectedConditions.ElementExists(By.CssSelector("a[href='/Admin/Home/Create']")));

btnCreateDT.Click();

wait.Until(driver => driver.Url != linkAdminafLogin);

((IJavaScriptExecutor)driver).ExecuteScript("window.scrollTo(0, document.body.scrollHeight);");

var lsHDH = driver.FindElements(By.CssSelector("#Mahdh option"));

var lsHDHinHD = new List<string>();

Console.WriteLine("-------------------------");

for (int i = 0; i < lsHDH.Count; i++)

{

lsHDHinHD.Add(lsHDH[i].Text);

Console.WriteLine(lsHDH[i].Text);

}

Console.WriteLine("Kiểm tra đã thật sự được update chưa");

if (lsHDHinHD.SequenceEqual(lsNameHDHnew))

{

Console.WriteLine("nó đã được update");

test = true;

}

Assert.That(test, Is.True);

}

}

[Test]

public void testEditHDH()

{

WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

string nameHDHAdd = "";

string valueChange = "hhhhh";

// get element

IList<IWebElement> rowTable;

//set element

List<string> lsNameHDH = new List<string>();

rowTable = getRowTableHDH(driver);

int count = 1;

count = rowTable.Count - 1;

var btnEdit = getBtn(driver, rowTable, count, 0);

var btnChiTiet = getBtn(driver, rowTable, count, 1);

var btnDelete = getBtn(driver, rowTable, count, 2);

//get name HDH được chọn

var item = getRow(driver, rowTable, count);

var nameItem = item.FindElements(By.CssSelector("td"))[1];

string stName = nameItem.Text;

Console.WriteLine("Hệ điều hành được chọn là: " + stName);

linkEditHDH = btnEdit.FindElement(By.CssSelector("a")).GetAttribute("href");

Console.WriteLine("Link : " + linkEditHDH);

btnEdit.Click();

wait.Until(driver => driver.Url != linkHeDieuHanh);

string url = driver.Url;

if (url.Equals(linkEditHDH))

{

Console.WriteLine("Link thành công");

// code xử lý khi chuyển thành công

IWebElement inputName = driver.FindElement(By.CssSelector("#Tenhdh"));

string value = inputName.GetAttribute("value");

if (value != null && value.Contains(stName))

{

Console.WriteLine("Tên trùng với nhau => đúng sản phẩm test");

valueChange = "Test 123";

Console.WriteLine("Tên thay đổi: " + valueChange);

//Nhập liệu lại value cần edit

inputName.Clear();

inputName.SendKeys(valueChange);

IWebElement btnEditChange = driver.FindElement(By.CssSelector("input[value='Save']"));

btnEditChange.Click();

WebDriverWait wait1 = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

wait1.Until(driver => driver.Url != linkEditHDH);

((IJavaScriptExecutor)driver).ExecuteScript("window.scrollTo(0, document.body.scrollHeight);");

// đợi chuyển trang

//wait.Until(driver => driver.Url != linkEditHDH);

string itemColl = "";

try

{

var tableRows = driver.FindElements(By.CssSelector("table tr"));

var table = tableRows[count-1];

var getColl = table.FindElements(By.CssSelector("td"))[1];

itemColl = (string)getColl.Text;

Console.WriteLine("Tên sản phẩm sau khi thay đổi: "+itemColl);

Console.WriteLine("Link thành công");

}

catch (Exception ex)

{

Console.WriteLine($"Error: {ex}");

}

Assert.That(itemColl,Is.EqualTo(valueChange));

}

}

else

{

Console.WriteLine("Link Không thành công");

Assert.Fail();

}

}

[Test]

public void testEditHDHFailt()

{

WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

string nameHDHAdd = "";

string valueChange = "HDH HyperOS";

// get element

IList<IWebElement> rowTable;

//set element

List<string> lsNameHDH = new List<string>();

rowTable = getRowTableHDH(driver);

int count = 1;

count = rowTable.Count - 1;

var btnEdit = getBtn(driver, rowTable, count, 0);

var btnChiTiet = getBtn(driver, rowTable, count, 1);

var btnDelete = getBtn(driver, rowTable, count, 2);

//get name HDH được chọn

var item = getRow(driver, rowTable, count);

var nameItem = item.FindElements(By.CssSelector("td"))[1];

string stName = nameItem.Text;

Console.WriteLine("Hệ điều hành được chọn là: " + stName);

linkEditHDH = btnEdit.FindElement(By.CssSelector("a")).GetAttribute("href");

Console.WriteLine("Link : " + linkEditHDH);

btnEdit.Click();

wait.Until(driver => driver.Url != linkHeDieuHanh);

string url = driver.Url;

if (url.Equals(linkEditHDH))

{

Console.WriteLine("Link thành công");

// code xử lý khi chuyển thành công

IWebElement inputName = driver.FindElement(By.CssSelector("#Tenhdh"));

string value = inputName.GetAttribute("value");

if (value != null && value.Contains(stName))

{

Console.WriteLine("Tên trùng với nhau => đúng sản phẩm test");

Console.WriteLine("Tên thay đổi: " + valueChange);

//Nhập liệu lại value cần edit

inputName.Clear();

inputName.SendKeys(valueChange);

IWebElement btnEditChange = driver.FindElement(By.CssSelector("input[value='Save']"));

btnEditChange.Click();

WebDriverWait wait1 = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

wait1.Until(driver => driver.Url != linkEditHDH);

((IJavaScriptExecutor)driver).ExecuteScript("window.scrollTo(0, document.body.scrollHeight);");

// đợi chuyển trang

//wait.Until(driver => driver.Url != linkEditHDH);

string itemColl = "";

try

{

var tableRows = driver.FindElements(By.CssSelector("table tr"));

var table = tableRows[count - 1];

var getColl = table.FindElements(By.CssSelector("td"))[1];

itemColl = (string)getColl.Text;

Console.WriteLine("Tên sản phẩm sau khi thay đổi: " + itemColl);

Console.WriteLine("Link thành công");

}

catch (Exception ex)

{

Console.WriteLine($"Error: {ex}");

}

Assert.That(itemColl, Is.EqualTo(valueChange));

}

}

else

{

Console.WriteLine("Link Không thành công");

Assert.Fail();

}

}

[Test]

public void testDetailHDH()

{

WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

string nameHDHAdd = "";

string valueChange = "";

// get element

IList<IWebElement> rowTable;

//set element

List<string> lsNameHDH = new List<string>();

rowTable = getRowTableHDH(driver);

int count = 1;

count = rowTable.Count - 1;

((IJavaScriptExecutor)driver).ExecuteScript("window.scrollTo(0, document.body.scrollHeight);");

//var btnEdit = getBtn(driver, rowTable, count, 0);

var btnChiTiet = getBtn(driver, rowTable, count, 1);

//var btnDelete = getBtn(driver, rowTable, count, 2);

//get name HDH được chọn

var item = getRow(driver, rowTable, count);

var nameItem = item.FindElements(By.CssSelector("td"))[1];

string stName = nameItem.Text;

Console.WriteLine("Hệ điều hành được chọn là: " + stName);

linkDetailHDH = btnChiTiet.FindElement(By.CssSelector("a")).GetAttribute("href");

Console.WriteLine("Link : " + linkDetailHDH);

btnChiTiet.Click();

wait.Until(driver => driver.Url != linkHeDieuHanh);

string url = driver.Url;

if (url.Equals(linkDetailHDH))

{

Console.WriteLine("Link thành công");

// code xử lý khi chuyển thành công

WebDriverWait wait1 = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

IWebElement testName = wait.Until(ExpectedConditions.ElementExists(By.CssSelector(".dl-horizontal dd")));

string value = testName.Text;

Assert.That(value, Is.EqualTo(stName));

}

else

{

Console.WriteLine("Link Không thành công");

Assert.Fail();

}

}

[Test]

public void testDeleteHDH()

{

WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

string nameHDHAdd = "";

string valueChange = "";

// get element

IList<IWebElement> rowTable;

//set element

List<string> lsNameHDH = new List<string>();

rowTable = getRowTableHDH(driver);

int count = 1;

count = rowTable.Count - 1;

((IJavaScriptExecutor)driver).ExecuteScript("window.scrollTo(0, document.body.scrollHeight);");

//var btnEdit = getBtn(driver, rowTable, count, 0);

//var btnChiTiet = getBtn(driver, rowTable, count, 1);

var btnDelete = getBtn(driver, rowTable, count, 2);

//get list HDH before delete one random element

for (int i = 1; i < rowTable.Count; i++)

{

var row = rowTable[i];

var cel = row.FindElements(By.CssSelector("td"));

lsNameHDH.Add(cel[0].Text.Trim() + " " + cel[1].Text.Trim());

Console.WriteLine(cel[0].Text.Trim() + " " + cel[1].Text.Trim());

}

Console.WriteLine("---------------------");

//get name HDH được chọn

var item = getRow(driver, rowTable, count);

var nameItem = item.FindElements(By.CssSelector("td"))[1];

string elementIdName = item.FindElements(By.CssSelector("td"))[0].Text.Trim() + " " + nameItem.Text.Trim();

string stName = nameItem.Text;

Console.WriteLine("Hệ điều hành được chọn là: " + stName);

Console.WriteLine("Hệ điều hành kèm id: " + elementIdName);

linkDeleteHDH = btnDelete.FindElement(By.CssSelector("a")).GetAttribute("href");

Console.WriteLine("Link : " + linkDeleteHDH);

btnDelete.Click();

wait.Until(driver => driver.Url != linkHeDieuHanh);

string url = driver.Url;

if (url.Equals(linkDeleteHDH))

{

Console.WriteLine("Link thành công");

// code xử lý khi chuyển thành công

IWebElement testName = wait.Until(ExpectedConditions.ElementExists(By.CssSelector(".dl-horizontal dd")));

string value = testName.Text;

if(value.Equals(stName)) { Console.WriteLine("tên trùng với nhau => đúng sản phẩm cần test"); }

else

{

Console.WriteLine($"Không phải cùng sản phẩm => chuyển hướng sai");

Assert.Fail();

}

IWebElement confirmDelete = wait.Until(ExpectedConditions.ElementExists(By.CssSelector("input[value='Delete']")));

confirmDelete.Click();

}

else

{

Console.WriteLine("Link Không thành công");

Assert.Fail();

}

WebDriverWait wait2 = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

wait2.Until(driver => driver.Url != linkDeleteHDH);

Thread.Sleep(2000);

((IJavaScriptExecutor)driver).ExecuteScript("window.scrollTo(0, document.body.scrollHeight);");

Console.WriteLine("--------get list HDH new----------");

// get list HDH after delete random element

List<string> lsNameHDHAfDelete = new List<string>();

var rowTableAfDelete = driver.FindElements(By.CssSelector("table tr"));

for (int i = 1; i < rowTableAfDelete.Count; i++)

{

var row = rowTableAfDelete[i];

var cel = row.FindElements(By.CssSelector("td"));

lsNameHDHAfDelete.Add(cel[0].Text.Trim()+" "+cel[1].Text.Trim());

Console.WriteLine(cel[0].Text.Trim() + " " + cel[1].Text.Trim());

}

Console.WriteLine("---------------------");

int lsEle = 0;

//list phần tử tổng sẽ giảm 1

lsEle = lsNameHDH.Count - lsNameHDHAfDelete.Count;

bool test = false;

if (lsEle == 1 && !lsNameHDHAfDelete.Any(a=>a.Contains(elementIdName)))

{

test = true;

}

Assert.That(test, Is.True);

}

[Test]

public void DatHang()

{

WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

userlogin(driver);

ThemSanPhamVaoGioHang(2, "https://localhost:44322/");

driver.Url = "https://localhost:44322/GioHang/GioHang";

var lsDT = driver.FindElements(By.CssSelector(".row col-md-3 col-sm-6"))[0];

var linkDT = lsDT.FindElement(By.CssSelector("img"));

string imageSrc = linkDT.GetAttribute("src");

linkDT.Click();

Thread.Sleep(2000);

var giohang = driver.FindElement(By.CssSelector(".giohang"));

var imgGH = driver.FindElement(By.CssSelector("img[src=\"/Images/files/ip3.jpg\"]"));

string imag = imgGH.GetAttribute("src");

if (imag.Contains(imageSrc))

{

Console.WriteLine("thêm thành công vào giỏ hàng");

}

var btnDatHang = driver.FindElement(By.CssSelector("input[value=\"Đặt hàng\"]"));

btnDatHang.Click();

Thread.Sleep(2000);

var inputDiachi = driver.FindElement(By.CssSelector("#Diachinhanhang"));

inputDiachi.SendKeys("282 Tạ Quang Bửu");

var phuongthuc = driver.FindElement(By.CssSelector("#MaTT"));

phuongthuc.Click();

var btnGioHang = driver.FindElement(By.CssSelector("input[value=\"Đặt hàng ngay\"]"));

btnGioHang.Click();

string linkgiohang = "https://localhost:44322/Donhangs";

wait.Until(driver => driver.Url != linkgiohang);

var dangxuat = driver.FindElement(By.CssSelector("a[href=\"/User/DangXuat\"]"));

dangxuat.Click();

arlogin(driver);

wait.Until(driver => driver.Url == linkAdminafLogin);

var btnDonhang = driver.FindElement(By.CssSelector("a[href=\"/Admin/Donhangs\"]"));

btnDonhang.Click();

var table = driver.FindElements(By.CssSelector("table tr"));

var clickCT = table[table.Count - 1].FindElement(By.CssSelector("btn btn-primary"));

var clickXacnhan = driver.FindElement(By.XPath("//a[contains(text(),\"Xác nhận đơn hàng\")]"));

clickXacnhan.Click();

wait.Until(driver => driver.Url == "https://localhost:44322/Admin/Donhangs");

Assert.That(true, Is.True);

}

public void ThemSanPhamVaoGioHang(int iMasp, string strURL)

{

// Navigate to the product page or use the existing page where the product is displayed

driver.Url = $"https://localhost:44322/GioHang/ThemGioHang?iMasp={iMasp}&strURL={Uri.EscapeDataString(strURL)}";

}

[Test]

public void testDeleteHDHFailt()

{

try

{

WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

string nameHDHAdd = "";

string valueChange = "";

// get element

IList<IWebElement> rowTable;

//set element

List<string> lsNameHDH = new List<string>();

rowTable = getRowTableHDH(driver);

int count = 1;

//count = rowTable.Count - 1;

((IJavaScriptExecutor)driver).ExecuteScript("window.scrollTo(0, document.body.scrollHeight);");

//var btnEdit = getBtn(driver, rowTable, count, 0);

//var btnChiTiet = getBtn(driver, rowTable, count, 1);

var btnDelete = getBtn(driver, rowTable, count, 2);

//get list HDH before delete one random element

for (int i = 1; i < rowTable.Count; i++)

{

var row = rowTable[i];

var cel = row.FindElements(By.CssSelector("td"));

lsNameHDH.Add(cel[0].Text.Trim() + " " + cel[1].Text.Trim());

Console.WriteLine(cel[0].Text.Trim() + " " + cel[1].Text.Trim());

}

Console.WriteLine("---------------------");

//get name HDH được chọn

var item = getRow(driver, rowTable, count);

var nameItem = item.FindElements(By.CssSelector("td"))[1];

string elementIdName = item.FindElements(By.CssSelector("td"))[0].Text.Trim() + " " + nameItem.Text.Trim();

string stName = nameItem.Text;

Console.WriteLine("Hệ điều hành được chọn là: " + stName);

Console.WriteLine("Hệ điều hành kèm id: " + elementIdName);

linkDeleteHDH = btnDelete.FindElement(By.CssSelector("a")).GetAttribute("href");

Console.WriteLine("Link : " + linkDeleteHDH);

btnDelete.Click();

wait.Until(driver => driver.Url != linkHeDieuHanh);

string url = driver.Url;

if (url.Equals(linkDeleteHDH))

{

Console.WriteLine("Link thành công");

// code xử lý khi chuyển thành công

IWebElement testName = wait.Until(ExpectedConditions.ElementExists(By.CssSelector(".dl-horizontal dd")));

string value = testName.Text;

if (value.Equals(stName)) { Console.WriteLine("tên trùng với nhau => đúng sản phẩm cần test"); }

else

{

Console.WriteLine($"Không phải cùng sản phẩm => chuyển hướng sai");

Assert.Fail();

}

IWebElement confirmDelete = wait.Until(ExpectedConditions.ElementExists(By.CssSelector("input[value='Delete']")));

confirmDelete.Click();

}

else

{

Console.WriteLine("Link Không thành công");

Assert.Fail();

}

WebDriverWait wait2 = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

wait2.Until(driver => driver.Url != linkDeleteHDH);

Thread.Sleep(2000);

((IJavaScriptExecutor)driver).ExecuteScript("window.scrollTo(0, document.body.scrollHeight);");

Console.WriteLine("--------get list HDH new----------");

// get list HDH after delete random element

List<string> lsNameHDHAfDelete = new List<string>();

var rowTableAfDelete = driver.FindElements(By.CssSelector("table tr"));

for (int i = 1; i < rowTableAfDelete.Count; i++)

{

var row = rowTableAfDelete[i];

var cel = row.FindElements(By.CssSelector("td"));

lsNameHDHAfDelete.Add(cel[0].Text.Trim() + " " + cel[1].Text.Trim());

Console.WriteLine(cel[0].Text.Trim() + " " + cel[1].Text.Trim());

}

Console.WriteLine("---------------------");

int lsEle = 0;

//list phần tử tổng sẽ giảm 1

lsEle = lsNameHDH.Count - lsNameHDHAfDelete.Count;

bool test = false;

if (lsEle == 1 && !lsNameHDHAfDelete.Any(a => a.Contains(elementIdName)))

{

test = true;

}

Assert.That(test, Is.True);

}

catch(WebDriverTimeoutException)

{

Console.WriteLine("Xóa không thành công");

Assert.Fail();

}

}

[Test]

public void arEditHDH()

{

string nameHDHAdd = "";

// get element

IList<IWebElement> rowTable;

//set element

List<string> lsNameHDH = new List<string>();

rowTable = getRowTableHDH(driver);

int count = 0;

count = rowTable.Count-1;

var btnEdit = getBtn(driver, rowTable, count, 0);

var btnChiTiet = getBtn(driver, rowTable, count, 1);

var btnDelete = getBtn(driver, rowTable, count, 2);

string linkEditHDH = btnEdit.FindElement(By.CssSelector("a")).GetAttribute("href");

btnEdit.Click();

string url = driver.Url;

Assert.That(url, Is.EqualTo(linkEditHDH));

if (url == linkEditHDH)

{

Assert.Pass();

Console.WriteLine("Link thành công");

// code xử lý khi chuyển thành công

}

else

{

Assert.Fail();

}

}

[Test]

public void arThemHDH()

{

//login vào admin

arlogin(driver);

// tới giao diện hệ điều hành

IWebElement arHedieihanh = driver.FindElement(By.CssSelector("a[href='/Admin/Hedieuhanhs']"));

adminArrive(driver, arHedieihanh);

// code test từ đây

// get element

IList<IWebElement> rowTable;

IWebElement btnAdd;

//set element

List<string> lsNameHDH = new List<string>();

btnAdd = driver.FindElement(By.CssSelector("a[href='/Admin/Hedieuhanhs/Create']"));

btnAdd.Click();

string currentUrl = driver.Url;

Assert.That(linkAddHDH, Is.EqualTo(currentUrl));

// so sánh link hiện tại với link cần tới

Thread.Sleep(2000);

}

[Test]

public void arriveNguoiDung()

{

//login vào admin

arlogin(driver);

//thẻ a link Người dùng

IWebElement arNguoiDung = driver.FindElement(By.CssSelector("a[href='/Admin/Nguoidungs']"));

//login vào admin

//link tới Người dùng

adminArrive(driver, arNguoiDung);

// so sánh link hiện tại với link cần tới

string currentUrl = driver.Url;

Assert.That(linkNguoidung, Is.EqualTo(currentUrl));

Thread.Sleep(2000);

}

public bool testChange()

{

return true;

}

[TearDown]

public void CloseBrowser()

{

driver.Quit();

}

}

}

## Anh Nhật

using NUnit.Framework;

using System;

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

using System.Collections.Generic;

using OpenQA.Selenium.Interactions;

using System.Linq;

using System.Threading;

namespace TestProject1

{

public class Tests

{

public IWebDriver driver;

IWebElement btnDangNhap;

IWebElement tk, mk, LoginAdmin;

ExcelOperations excel = new ExcelOperations("D:\\AUTOMATION\\test.xlsx", "Sheet1");

[SetUp]

public void Setup()

{

driver = new ChromeDriver();

driver.Navigate().GoToUrl("https://localhost:44322/Home/Index");

Thread.Sleep(1000);

btnDangNhap = driver.FindElement(By.CssSelector("a[href='/User/Dangnhap']"));

}

private void DN(IWebDriver driver)

{

//click vào đnăg nhập

btnDangNhap.Click();

//setup element login

tk = driver.FindElement(By.CssSelector("#userMail"));

mk = driver.FindElement(By.CssSelector("#password"));

LoginAdmin = driver.FindElement(By.CssSelector("input[value='Đăng nhập']"));

//input data

tk.SendKeys("Admin@gmail.com");

mk.SendKeys("12345678");

LoginAdmin.Click();

}

// Kiểm tra xem có trang kế tiếp hay không

public bool CoTrangKeTiep()

{

try

{

// Thay thế bằng đúng selector của phần tử nút "Next"

IWebElement nextButton = driver.FindElement(By.CssSelector("li[class='PagedList-skipToNext'] a"));

return nextButton.Enabled; // Kiểm tra xem nút "Next" có được kích hoạt hay không

}

catch (NoSuchElementException)

{

return false; // Nếu không tìm thấy nút "Next", có thể đã ở trang cuối cùng

}

}

// Chuyển đến trang kế tiếp (nếu có)

public void ChuyenDenTrangKeTiep()

{

if (CoTrangKeTiep())

{

IWebElement nextButton = driver.FindElement(By.CssSelector("li[class='PagedList-skipToNext'] a"));

nextButton.Click(); // Nhấp vào nút "Next" để chuyển đến trang kế tiếp

}

else

{

Console.WriteLine("Đã ở trang cuối cùng");

}

}

//done

[Test]

public void AddSP()

{

DN(driver);

List<string> lsSP = new List<string>();

List<string> lsSP2 = new List<string>();

bool hasMorePages = true;

bool hasMorePages2 = true;

while (hasMorePages)

{

// Lấy danh sách sản phẩm trên trang hiện tại

IList<IWebElement> rowTable = driver.FindElements(By.CssSelector("table tr"));

for (int i = 1; i < rowTable.Count; i++)

{

var row = rowTable[i];

var cel = row.FindElements(By.CssSelector("td"));

lsSP.Add(cel[0].Text); // Lấy dữ liệu từ cột đầu tiên

Console.WriteLine(cel[0].Text);

}

// Kiểm tra và chuyển đến trang kế tiếp (nếu có)

try

{

IWebElement nextButton = driver.FindElement(By.CssSelector("li[class='PagedList-skipToNext'] a"));

nextButton.Click(); // Nhấp vào nút "Next" để chuyển đến trang kế tiếp

}

catch (NoSuchElementException)

{

Console.WriteLine("Đã ở trang cuối cùng");

hasMorePages = false; // Không tìm thấy nút "Next", dừng vòng lặp

}

}

Console.WriteLine("Số Lượng:{0} ", lsSP.Count);

// them sp

driver.FindElement(By.LinkText("Thêm mới")).Click();

driver.FindElement(By.Id("Tensp")).Click();

driver.FindElement(By.Id("Tensp")).SendKeys("Iphone hihi");

driver.FindElement(By.Id("Giatien")).Click();

driver.FindElement(By.Id("Giatien")).SendKeys("2000000");

driver.FindElement(By.Id("Soluong")).Click();

driver.FindElement(By.Id("Soluong")).SendKeys("100");

driver.FindElement(By.Id("Mota")).Click();

driver.FindElement(By.Id("Mota")).SendKeys("Mới…...");

driver.FindElement(By.Id("Thesim")).Click();

driver.FindElement(By.Id("Thesim")).SendKeys("2");

driver.FindElement(By.Id("Bonhotrong")).Click();

driver.FindElement(By.Id("Bonhotrong")).SendKeys("1");

driver.FindElement(By.Id("Sanphammoi")).Click();

{

var dropdown = driver.FindElement(By.Id("Sanphammoi"));

dropdown.FindElement(By.XPath("//option[. = 'True']")).Click();

}

driver.FindElement(By.Id("Ram")).Click();

driver.FindElement(By.Id("Ram")).SendKeys("100");

driver.FindElement(By.Id("Anhbia")).Click();

driver.FindElement(By.Id("Anhbia")).SendKeys(". ko có");

driver.FindElement(By.Id("Mahang")).Click();

{

var dropdown = driver.FindElement(By.Id("Mahang"));

dropdown.FindElement(By.XPath("//option[normalize-space()='Apple']")).Click();

}

driver.FindElement(By.Id("Mahdh")).Click();

{

var dropdown = driver.FindElement(By.Id("Mahdh"));

dropdown.FindElement(By.XPath("//option[normalize-space()='IOS']")).Click();

}

driver.FindElement(By.CssSelector(".btn-success")).Click();

while (hasMorePages2)

{

// Lấy danh sách sản phẩm trên trang hiện tại

IList<IWebElement> rowTable = driver.FindElements(By.CssSelector("table tr"));

for (int i = 1; i < rowTable.Count; i++)

{

var row = rowTable[i];

var cel = row.FindElements(By.CssSelector("td"));

lsSP2.Add(cel[0].Text); // Lấy dữ liệu từ cột đầu tiên

Console.WriteLine(cel[0].Text);

}

// Kiểm tra và chuyển đến trang kế tiếp (nếu có)

try

{

IWebElement nextButton = driver.FindElement(By.CssSelector("li[class='PagedList-skipToNext'] a"));

nextButton.Click(); // Nhấp vào nút "Next" để chuyển đến trang kế tiếp

}

catch (NoSuchElementException)

{

Console.WriteLine("Đã ở trang cuối cùng");

hasMorePages2 = false; // Không tìm thấy nút "Next", dừng vòng lặp

}

}

Console.WriteLine("Số Lượng:{0} ", lsSP2.Count);

bool test = true;

if (lsSP2.Count - lsSP.Count == 1)

{

Console.WriteLine("Thêm thành công");

}

else

{

test = false;

}

Assert.That(test, Is.True);

}

//done

[Test]

public void EditSP()

{

DN(driver);

driver.FindElement(By.CssSelector("tr:nth-child(3) a:nth-child(1)")).Click();

driver.FindElement(By.Id("Tensp")).Click();

driver.FindElement(By.Id("Tensp")).SendKeys("Ximiheo");

driver.FindElement(By.Id("Ram")).Click();

driver.FindElement(By.Id("Ram")).SendKeys("8");

Thread.Sleep(4000);

driver.FindElement(By.CssSelector(".btn-success")).Click();

}

//done

[Test]

public void DetailSP()

{

DN(driver);

driver.FindElement(By.LinkText("Chi tiết")).Click();

Thread.Sleep(4000);

driver.FindElement(By.XPath("//a[contains(text(),'Về trang chủ')]")).Click();

}

//done

[Test]

public void XoaSP()

{

DN(driver);

driver.FindElement(By.CssSelector("a[href='/Admin/Home/Delete/34']")).Click();

Thread.Sleep(4000);

driver.FindElement(By.CssSelector("input[value='Delete']")).Click();

}

//Done

[Test]

public void DelHang()

{

DN(driver);

driver.FindElement(By.CssSelector("a[href='/Admin/Hangsanxuats']")).Click();

driver.FindElement(By.CssSelector("a[href='/Admin/Hangsanxuats/Delete/13']")).Click();

Thread.Sleep(4000);

driver.FindElement(By.CssSelector(".btn-default")).Click();

}

//Done

[Test]

public void EditHang()

{

DN(driver);

driver.FindElement(By.CssSelector("a[href='/Admin/Hangsanxuats']")).Click();

driver.FindElement(By.CssSelector("a[href='/Admin/Hangsanxuats/Edit/10']")).Click();

driver.FindElement(By.Id("Tenhang")).Click();

{

var element = driver.FindElement(By.Id("Tenhang"));

Actions builder = new Actions(driver);

builder.DoubleClick(element).Perform();

}

driver.FindElement(By.Id("Tenhang")).SendKeys("Xmi");

driver.FindElement(By.CssSelector(".btn-default")).Click();

}

//done

[Test]

public void DtHang()

{

DN(driver);

driver.FindElement(By.CssSelector("a[href='/Admin/Hangsanxuats']")).Click();

driver.FindElement(By.CssSelector("a[href='/Admin/Hangsanxuats/Details/1']")).Click();

Thread.Sleep(1000);

driver.FindElement(By.XPath("//a[normalize-space()='Back to List']")).Click();

}

//done

[Test]

public void AddHang()

{

string tenH = "HELOWORD";

DN(driver);

driver.FindElement(By.CssSelector("a[href='/Admin/Hangsanxuats']")).Click();

IList<IWebElement> rowTable;

// LAY DS HANG

List<string> lsH = new List<string>();

rowTable = driver.FindElements(By.CssSelector("table tr"));

List<string> lsHA = new List<string>();

Console.WriteLine("LIST HANG LUC BAN DAU");

for (int i = 1; i < rowTable.Count; i++)

{

var row = rowTable[i];

var cel = row.FindElements(By.CssSelector("td"));

lsHA.Add(cel[1].Text);

/\* Console.WriteLine(cel[1].Text);\*/

}

Console.WriteLine("Số Lượng:{0} ",lsHA.Count);

// THEM SP

driver.FindElement(By.CssSelector("a[href='/Admin/Hangsanxuats/Create']")).Click();

driver.FindElement(By.CssSelector("#Tenhang")).SendKeys(tenH);

Thread.Sleep(1000);

driver.FindElement(By.CssSelector("input[value='Create']")).Click();

List<string> lsHm = new List<string>();

rowTable = driver.FindElements(By.CssSelector("table tr"));

List<string> lsHAM = new List<string>();

Console.WriteLine("LIST HANG SAU KHI THEM");

for (int i = 1; i < rowTable.Count; i++)

{

var row = rowTable[i];

var cel = row.FindElements(By.CssSelector("td"));

lsHAM.Add(cel[1].Text);

/\* Console.WriteLine(cel[1].Text);\*/

}

Console.WriteLine("Số Lượng:{0} ", lsHAM.Count);

bool test = true;

if (lsHAM.Count- lsHA.Count == 1)

{

Console.WriteLine("Thêm thành công");

}

else

{

test = false;

}

Assert.That(test, Is.True);

driver.FindElement(By.XPath("//a[contains(text(),'Sản phẩm')]")).Click();

driver.FindElement(By.LinkText("Thêm mới")).Click();

driver.FindElement(By.Id("Tensp")).Click();

driver.FindElement(By.Id("Tensp")).SendKeys("Iphone hihi");

driver.FindElement(By.Id("Giatien")).Click();

driver.FindElement(By.Id("Giatien")).SendKeys("2000000");

driver.FindElement(By.Id("Soluong")).Click();

driver.FindElement(By.Id("Soluong")).SendKeys("100");

driver.FindElement(By.Id("Mota")).Click();

driver.FindElement(By.Id("Mota")).SendKeys("Mới…...");

driver.FindElement(By.Id("Thesim")).Click();

driver.FindElement(By.Id("Thesim")).SendKeys("2");

driver.FindElement(By.Id("Bonhotrong")).Click();

driver.FindElement(By.Id("Bonhotrong")).SendKeys("1");

driver.FindElement(By.Id("Sanphammoi")).Click();

{

var dropdown = driver.FindElement(By.Id("Sanphammoi"));

dropdown.FindElement(By.XPath("//option[. = 'True']")).Click();

}

driver.FindElement(By.Id("Ram")).Click();

driver.FindElement(By.Id("Ram")).SendKeys("100");

driver.FindElement(By.Id("Anhbia")).Click();

driver.FindElement(By.Id("Anhbia")).SendKeys(". ko có");

driver.FindElement(By.Id("Mahang")).Click();

{

var dropdown = driver.FindElement(By.Id("Mahang"));

dropdown.FindElement(By.XPath("//option[normalize-space()='HELOWORD']")).Click();

}

driver.FindElement(By.Id("Mahdh")).Click();

{

var dropdown = driver.FindElement(By.Id("Mahdh"));

dropdown.FindElement(By.XPath("//option[normalize-space()='IOS']")).Click();

}

driver.FindElement(By.CssSelector(".btn-success")).Click();

Console.WriteLine(" Thêm sản phẩm có hãng thành công ");

List<string> lsSPHANG = new List<string>();

bool hasMorePages = true;

while (hasMorePages)

{

// Lấy danh sách sản phẩm trên trang hiện tại

rowTable = driver.FindElements(By.CssSelector("table tr")); // Cập nhật lại rowTable khi chuyển trang

for (int i = 1; i < rowTable.Count; i++)

{

var row = rowTable[i];

var cel = row.FindElements(By.CssSelector("td"));

lsSPHANG.Add(cel[6].Text); // Lấy dữ liệu từ cột HANG

/\* Console.WriteLine(cel[6].Text);\*/

}

// Kiểm tra và chuyển đến trang kế tiếp (nếu có)

try

{

IWebElement nextButton = driver.FindElement(By.CssSelector("li[class='PagedList-skipToNext'] a"));

nextButton.Click(); // Nhấp vào nút "Next" để chuyển đến trang kế tiếp

}

catch (NoSuchElementException)

{

Console.WriteLine("Đã ở trang cuối cùng");

hasMorePages = false; // Không tìm thấy nút "Next", dừng vòng lặp

}

}

bool KTHANGSP = lsSPHANG.Contains(tenH); // Thay "XmeXme" bằng tên hãng bạn thêm

if (true)

{

Console.WriteLine("Hang có nằm trong sản phẩm");

}

else { Console.WriteLine("Hang ko nằm trong sản phẩm"); }

}

}

}

## Lan Nguyên

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

using System.Threading;

using OpenQA.Selenium.Support.UI;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

namespace BaoCao

{

[TestClass]

public class LoginTest

{

IWebDriver driver;

[TestInitialize]

public void Init()

{

driver = new ChromeDriver();

driver.Url = "https://localhost:44322/User/Dangnhap";

driver.Navigate();

Thread.Sleep(5000);

}

[TestMethod]

public void DatHang()

{

// Fill in email

var emailField = driver.FindElement(By.Id("userMail"));

emailField.SendKeys("test@gmail.com");

// Fill in password

var passwordField = driver.FindElement(By.Id("password"));

passwordField.SendKeys("12345678");

// Click login button

var loginButton = driver.FindElement(By.CssSelector(".btn.btn-primary"));

loginButton.Click();

// Verify successful login

Assert.AreEqual("https://localhost:44322/", driver.Url);

ThemSanPhamVaoGioHang(2, "https://localhost:44322/");

driver.Url = "https://localhost:44322/GioHang/GioHang";

// Click on "Chỉnh sửa giỏ hàng" link

var editCartLink = driver.FindElement(By.XPath("//a[contains(@href,'/GioHang/SuaGioHang')]"));

editCartLink.Click();

var quantityInput = driver.FindElement(By.Id("txtSoLuong"));

quantityInput.Clear();

quantityInput.SendKeys("3");

//Click on the "Cập nhật" button

var capNhatButton = driver.FindElement(By.CssSelector(".btn-secondary"));

capNhatButton.Click();

// Click on the "Xóa" button

// var xoaButton = driver.FindElement(By.CssSelector(".btn-success a"));

// xoaButton.Click();

// Navigate back to the main page

// driver.Navigate().GoToUrl("https://localhost:44322/");

// Click on the "Đặt hàng" button

var datHangButton = driver.FindElement(By.CssSelector("form[action='/GioHang/ThanhToanDonHang'] input[type='submit']"));

datHangButton.Click();

// Fill in information for "Diachinhanhang"

var diaChiNhanHangInput = driver.FindElement(By.Id("Diachinhanhang"));

diaChiNhanHangInput.SendKeys("HCM");

// Click on the "Đặt hàng ngay" button

var datHangNgayButton = driver.FindElement(By.CssSelector("form[action='/GioHang/ThanhToanDonHang'] input[type='submit']"));

datHangNgayButton.Click();

}

public void ThemSanPhamVaoGioHang(int iMasp, string strURL)

{

// Navigate to the product page or use the existing page where the product is displayed

driver.Url = $"https://localhost:44322/GioHang/ThemGioHang?iMasp={iMasp}&strURL={Uri.EscapeDataString(strURL)}";

}

[TestCleanup]

public void CleanUp()

{

Thread.Sleep(5000);

driver.Quit();

}

}

}

# KẾT LUẬN

## Tổng kết

Tổng kết dự án này, chúng em đã đặt nỗ lực lớn để hoàn thành công việc. Mặc dù có nhận thức về những hạn chế và thiếu sót, nhưng chúng em tin rằng chúng đã là một bước tiến quan trọng trong quá trình học tập và phát triển kỹ năng của chúng tôi.

Chúng em đã tập trung vào việc hoàn thành các chức năng cơ bản của đồ án. Tuy nhiên, nhận thức được rằng chỉ hoàn thành được khoảng 80% của tiêu chí đã đề ra là một thực tế. Quá trình này đã giúp chúng em nhận ra sức mạnh của kiên trì và sự quyết tâm để vượt qua những thách thức trong việc thực hiện dự án.

Chúng em cảm thấy hạnh phúc với những gì đã đạt được và xem những thiếu sót là cơ hội học hỏi. Chúng em sẽ tiếp tục cải thiện, học hỏi từ những sai lầm và áp dụng những kinh nghiệm này vào những dự án tương lai.

## Các thuận lợi khi test auto

* Tiết kiệm thời gian và chi phí: Loại bỏ lỗi sớm giúp tránh chi phí sửa chữa lớn sau này và tối ưu hóa quá trình phát triển.
* Tăng tốc độ: Thực hiện kiểm thử nhanh chóng và lặp lại hàng loạt bằng cách sử dụng kịch bản tự động.
* Tăng độ chính xác: Loại bỏ yếu tố con người và giảm sai sót trong quá trình kiểm thử.
* Đồng Thời Hóa: Bài kiểm thử tự động có thể chạy đồng thời trên các cấu hình, trình duyệt hoặc thiết bị khác nhau, giúp thực hiện song song và nhanh chóng quá trình kiểm thử.
* Độ chính xác cao: Bài kiểm thử tự động thực hiện các bước xác định một cách chính xác, giảm khả năng xảy ra lỗi do con người thường gặp trong kiểm thử thủ công.
* Có thể kiểm thử mở rộng: Kiểm thử tự động có thể mở rộng dễ dàng để xử lý bộ kiểm thử lớn, phù hợp cho các dự án có yêu cầu kiểm thử phức tạp và chi tiết.
* Báo Cáo Chi Tiết: Các công cụ kiểm thử tự động tạo ra báo cáo chi tiết về quá trình thực hiện kiểm thử, phủ sóng kiểm thử và vấn đề phát hiện, hỗ trợ quyết định và phân tích tốt hơn.

## Các khó khăn

* Cần có khả năng tư duy và lập trình tốt
* Có những kịch bản hoặc thử nghiệm cần sự can thiệp thủ công hoặc có tính phức tạp cao, và việc tự động hóa có thể khó khăn.

## Hướng phát triển nếu có thời gian

* Kiểm tra toàn bộ các trường hợp của đồ án
* Kiểm tra rộng và sâu hơn
* Kiểm tra hiệu suất và bảo mật
* Tối ưu hóa kịch bản

# TÀI LIỆU THAM KHẢO

[Link 1](https://www.youtube.com/watch?v=1wdUKOuDQxI&t=632s)

[Link 2](https://viblo.asia/p/automation-test-with-selenium-nwmGyMogGoW)

[Link 3](https://anhtester.com/lesson/selenium-java-bai-5-cach-xac-dinh-locators-cua-element-tren-website?fbclid=IwAR1asFRtVIZ8jsoTnjzol435giWA_xpf6yLxPSWtyhHNe2BkP1CnKZyGPPA)