TRƯỜNG ĐẠI HỌC TÂY NGUYÊN

**KHOA KHOA HỌC TỰ NHIÊN VÀ CÔNG NGHỆ**

Logo

Description automatically generated

**CHUYÊN ĐỀ TỐT NGHIỆP**

**XÂY DỰNG CÁC TRƯỜNG HỢP KIỂM THỬ TỰ ĐỘNG CHO WEBSITE SỬ DỤNG SELENIUM VỚI PYTHON**

**Sinh viên : Đinh Ngọc Hương Giang**

**Chuyên ngành : Công nghệ thông tin**

**Khóa học : 2020 - 2024**

**Đắk Lắk, tháng 05 năm 2024**

TRƯỜNG ĐẠI HỌC TÂY NGUYÊN

**KHOA KHOA HỌC TỰ NHIÊN VÀ CÔNG NGHỆ**

Logo

Description automatically generated

**CHUYÊN ĐỀ TỐT NGHIỆP**

**XÂY DỰNG CÁC TRƯỜNG HỢP KIỂM THỬ TỰ ĐỘNG CHO WEBSITE SỬ DỤNG SELENIUM VỚI PYTHON**

**Sinh viên: Đinh Ngọc Hương Giang**

Chuyên ngành: Công nghệ thông tin

**Người hướng dẫn**

ThS. Trương Hải

**Đắk Lắk, tháng 05 năm 2024**

# Lời cảm ơn

Em xin phép gửi lời cảm ơn sâu sắc tới Khoa Khoa học tự nhiên và Công nghệ trường Đại học Tây Nguyên đã đồng hành cùng em trong suốt 4 năm học tập tại trường, cùng với bộ môn Công nghệ thông tin đã trực tiếp giảng dạy và hướng dẫn em đi suốt chặng đường là sinh viên của em.

Chính thầy cô đã mang lại cho em những nền tảng vững chắc, cung cấp kiến thức và kỹ năng mềm cho em có thể thực hiện đề tài cách dễ dàng hơn. Cùng với giáo viên hướng dẫn, thầy ThS. Trương Hải đã tạo điều kiện và tận tâm hỗ trợ cho em trong suốt quá trình làm đề tài chuyên đề tốt nghiệp. Sự giúp đỡ của thầy đã giúp em hoàn thành tốt đề tài. Em cũng xin gia đình và bạn bè đã và đang tiếp sức, động viên trong suốt khoản thời gian học tập tại trường.

Mặc dù đã rất cố gắng để thực hiện tốt đề tài nhưng không thể tránh thiết sót những sai lầm, mong quý thầy cô có thể góp ý giúp em hoàn thành tốt hơn nữa.

Em xin chân thành cảm ơn!

Đắk Lắk, tháng 05 năm 2024

|  |  |
| --- | --- |
|  | **SINH VIÊN**  **Đinh Ngọc Hương Giang** |

# Mục lục

[LỜI CẢM ƠN i](#_Toc109621396)

[MỤC LỤC ii](#_Toc109621397)

[CÁC CỤM TỪ VIẾT TẮT, KÝ HIỆU iv](#_Toc109621398)

[DANH MỤC BẢNG BIỂU v](#_Toc109621399)

[DANH MỤC HÌNH ẢNH vi](#_Toc109621400)

[ĐẶT VẤN ĐỀ 1](#_Toc109621401)

[1. Tính cấp thiết 1](#_Toc109621402)

[2. Mục tiêu nghiên cứu 1](#_Toc109621403)

[CƠ SỞ LÝ LUẬN 2](#_Toc109621404)

[NỘI DUNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU 3](#_Toc109621405)

[1. Đối tượng nghiên cứu 3](#_Toc109621406)

[2. Phạm vi nghiên cứu 3](#_Toc109621407)

[3. Nội dung nghiên cứu 3](#_Toc109621408)

[4. Phương pháp nghiên cứu 3](#_Toc109621409)

[5. Ý nghĩa khoa học và thực tiễn 3](#_Toc109621409)

[CHƯƠNG 1. TỔNG QUAN 5](#_Toc109621410)

[1.1 Tổng quan tình hình nghiên cứu 5](#_Toc109621411)

[1.2 Giới thiệu chung về kiểm thử phần mềm 5](#_Toc109621412)

[1.3 Nguyên tắc kiểm thử phần mềm 8](#_Toc109621413)

[CHƯƠNG 2. TỔNG QUAN VỀ CÔNG CỤ KIỂM THỬ 10](#_Toc109621414)

[2.1 Công cụ kiểm thử tự động Selenium 10](#_Toc109621415)

[2.2 Ngôn ngữ lập trình Python 14](#_Toc109621416)

[2.3 Unittest Framework 14](#_Toc109621417)

[Chương 3. XÂY DỰNG TRƯỜNG HỢP KIỂM THỬ 17](#_Toc109621418)

[3.1 Trường hợp kiểm thử 17](#_Toc109621419)

[3.2 Viết trường hợp kiểm thử giao diện 19](#_Toc109621420)

[3.3 Viết trường hợp kiểm thử chức năng website 23](#_Toc109621421)

[CHƯƠNG 4. THỰC HIỆN VÀ CHẠY TỰ ĐỘNG KIỂM THỬ 32](#_Toc109621424)

[4.1 Kiểm thử tự động 32](#_Toc109621425)

[4.2 Thực hiện kiểm thử giao diện tự động 33](#_Toc109621426)

[4.3 Thực hiện kiểm thử chức năng tự động 37](#_Toc109621427)

[PHẦN KẾT LUẬN 41](#_Toc109621434)

[1. Kết quả đạt được 41](#_Toc109621435)

[2. Ưu điểm 41](#_Toc109621436)

[3. Nhược điểm 41](#_Toc109621435)

[2. Hướng phát triển cho tương lai 41](#_Toc109621436)

[TÀI LIỆU THAM KHẢO 43](#_Toc109621437)

[PHỤ LỤC 44](#_Toc109621438)

# Các cụm từ viết tắt, ký hiệu

|  |  |  |
| --- | --- | --- |
| Từ viết tắt | Từ đầy đủ | Nghĩa của từ |
| API | Application Programming Interface | Giao diện lập trình cung cấp cho ứng dụng |
| AI | Artificial Intelligence | Trí tuệ nhân tạo |
| CI/CD | Continuous Integration và Continuous Delivery | Tích hợp liên tục và triển khai liên tục |
| TC | Test Case | Trường hợp kiểm thử |

# Danh mục Bảng biểu

[Bảng 2.1.3 Trình điều kiển web và nguồn cài đặt 12](#_Toc109621439)

[Bảng 3.1.3 Cấu trúc Test Case 19](#_Toc109621440)

[Bảng 3.2.1 Test Case giao diện Trang chủ 21](#_Toc109621441)

[Bảng 3.2.2 Test Case giao diện Trang Demo Site 22](#_Toc109621442)

[Bảng 3.2.3 Test Case Thêm vào giỏ hàng 22](#_Toc109621443)

[Bảng 3.3.1 Test Case chức năng Đăng ký 23](#_Toc109621444)

[Bảng 3.3.2 Test Case chức năng Đăng nhập 25](#_Toc109621445)

[Bảng 3.3.3 Test Case chức năng Đăng xuất 25](#_Toc109621446)

[Bảng 3.3.4 Test Case chức năng Cập nhật tài khoản 27](#_Toc109621447)

[Bảng 3.3.5 Test Case chức năng Thêm địa chỉ giao hàng 28](#_Toc109621448)

[Bảng 3.3.6 Test Case chức năng Đặt hàng mua sắm 30](#_Toc109621443)

[Bảng 4.1.2 Quy trình kiểm thử tự động 32](#_Toc109621444)

[Bảng 4.2.1 Các bước thực hiện giao diện Trang chủ 34](#_Toc109621445)

[Bảng 4.2.2 Các bước thực hiện giao diện Trang Demo Site 35](#_Toc109621446)

[Bảng 4.2.3 Các bước thực hiện giao diện Thêm vào giỏ hàng 36](#_Toc109621447)

# Danh mục hình ảnh

[Hình 1.3 Nguyên tắc kiểm thử 8](#_Toc109621449)

[Hình 2.1.1 Phân loại Selenium 10](#_Toc109621450)

[Hình 2.1.2 Kiến trúc WebDriver của Selenium 11](#_Toc109621451)

[Hình 2.3.1 Sơ đồ thể hiện Uniitest 15](#_Toc109621452)

[Hình 2.3.2 Kết quả hiển thị sau khi chạy 16](#_Toc109621453)

[Hình 3.1.3 Giải thích vòng đời lỗi 17](#_Toc109621454)

[Hình 3.2.1 Giao diện Trang chủ 21](#_Toc109621455)

[Hình 3.2.2 Giao diện Trang Demo Site 22](#_Toc109621456)

[Hình 3.2.3 Giao diện Thêm vào giỏ hàng 23](#_Toc109621457)

[Hình 3.3.1 Giao diện trang Đăng ký 24](#_Toc109621458)

[Hình 3.3.2 Giao diện trang Đăng nhập 25](#_Toc109621459)

[Hình 3.3.3 Giao diện trang Đăng xuất 26](#_Toc109621460)

[Hình 3.3.4 Giao diện trang Cập nhật tài khoản 27](#_Toc109621461)

[Hình 3.3.5 Giao diện trang Thêm địa chỉ giao hàng 29](#_Toc109621462)

[Hình 3.3.6 Giao diện trang Đặt hàng mua sắm 31](#_Toc109621463)

[Hình 4.2.1 Kết quả giao diện Trang chủ 34](#_Toc109621464)

[Hình 4.2.2 Kết quả giao diện Trang Demo Site 36](#_Toc109621465)

[Hình 4.2.3 Kết quả giao diện Thêm sản phẩm vào giỏ hàng 37](#_Toc109621466)

[Hình 4.3.1 Kết quả chức năng Đăng ký 37](#_Toc109621467)

[Hình 4.3.2 Kết quả chức năng Đăng nhập 38](#_Toc109621468)

[Hình 4.3.3 Kết quả chức năng Đăng xuất 39](#_Toc109621469)

[Hình 4.3.5 Kết quả chức năng Đặt hàng mua sắm và Thêm địa chỉ 40](#_Toc109621470)

# Đặt vấn đề

## Tính cấp thiết

Nền kinh tế số tạo điều kiện cho sự phát triển mạnh mẽ, việc sử dụng website không chỉ trong giáo dục, thương mại, giải trí mà còn là cả một trang thông tin cũng đang trở nên ngày càng phổ biến hơn. Website đóng vai trò quan trọng trong việc kết nối tương tác với khách hàng, giao lưu cộng đồng.

Cùng với sự phát triển của công nghệ website, các doanh nghiệp và nhà phát triển website đang đối mặt với thách thức trong việc kiểm soát lỗi và đảm bảo chất lượng sản phẩm. Lỗi trong website không chỉ gây ra thiệt hại về bảo mật mà còn ảnh hưởng đến chức năng của sản phẩm, gây ra thiệt hại về kinh tế, giảm mất độ tin cậy của khách hàng.

Kiểm thử phần mềm là một bước không thể thiếu trong quá trình phát triển sản phẩm công nghệ thông tin, bao gồm cả phần mềm và website. Qua việc áp dụng các phương pháp kiểm thử chặt chẽ ngay từ những giai đoạn đầu tiên cho đến khi hoàn thiện, các lỗi kỹ thuật có thể được sàng lọc và giải quyết một cách nhanh chóng.Vì vậy, việc sử dụng kiểm thử tự động trở thành một giải pháp hiệu quả, giúp nâng cao hiệu suất và chất lượng của các website.

Với lý do đó, em đã chọn đề tài: **“Xây dựng các trường hợp kiểm thử tự động cho Website sử dụng Selenium với Python”.**

## Mục tiêu nghiên cứu

Trong quá trình hoàn thiện đề tài chuyên đề, em đặt ra những mục tiêu sau:

* Xây dựng kế hoạch kiểm thử Website trực tuyến
* Xây dựng được 1 kịch bản kiểm thử có thể tái sử dụng
* Sử dụng để tăng cường tính hoàn thiện, chất lượng tới tận tay khách hàng

# Cơ sở lý luận

Website: Website là tập hợp một trang chứa các thông tin bao gồm hình ảnh, video, văn bản... trên một domain, được lưu trữ trên máy chủ web. Website có thể được người dùng truy cập từ xa thông qua mạng internet.Website đóng vai trò quan trọng trong đời sống hiện đại, phục vụ cho mục đích truyền tải thông tin, quảng bá thương hiệu, kinh doanh trực tuyến, v.v.

Selenium: Selenium là một bộ công cụ kiểm thử tự động web mã nguồn mở, hỗ trợ kiểm thử miễn phí trên nhiều trình duyệt web phổ biến như Chrome, Firefox, Safari, Edge, v.v. Selenium cung cấp các API và thư viện cho phép lập trình viên viết các trường hợp kiểm thử tự động mô phỏng hành vi người dùng trên website.

Python: Python là ngôn ngữ lập trình bậc cao, đa năng, dễ sử dụng và có cộng đồng người dùng lớn. Sử dụng Python để viết các trường hợp kiểm thử tự động giúp đơn giản hóa việc phát triển và bảo trì các trường hợp kiểm thử.

Unittest Framework: Được thống kê trong những framework hàng đầu năm 2023. Đây là phiên bản ngôn ngữ Python của JUnit, cũng là một phần của Thư viện chuẩn Python kể từ phiên bản Python 2.1. Đây là khung thử nghiệm mặc định mà bạn có thể sử dụng ngay với gói Python, vì vậy hầu hết các nhà phát triển Python đều bắt đầu chạy thử nghiệm với nó trước khi sử dụng các khung khác, cho phép tự động hóa kiểm thử, chia sẻ thiết lập và tắt mã cho các kiểm thử, đối chiếu các kiểm thử thành các bộ sưu tập và giữ các kiểm thử độc lập với hệ thống phân cấp báo cáo. Tạo báo cáo nhanh chóng nhưng chi tiết bằng XML cũng như báo cáo unittest-sml

Testcase (trường hợp kiểm thử): Testcase là một tập các bước được thiết kế để kiểm tra một chức năng cụ thể của phần mềm. Các testcase kiểm thử tự động có thể được tái sử dụng và dễ dàng bảo trì khi có thay đổi trong phần mềm.

# Nội dung và phương pháp nghiên cứu

## Đối tượng nghiên cứu

* Công cụ kiểm thử tự động Selenium, ngôn ngữ lập trình Python, Unittest.
* Thiết kế trường hợp kiểm thử tự động Website.

## Phạm vi nghiên cứu

* Thiết kế trường hợp kiểm thử: Thiết kế các trường hợp kiểm thử chi tiết, bao gồm các bước kiểm thử, dữ liệu đầu vào, dữ liệu đầu ra mong đợi và kết quả kiểm thử thực tế.
* Thực hiện kiểm thử: Chạy các trường hợp kiểm thử tự động trên website và ghi nhận kết quả.

## Nội dung nghiên cứu

Chương 1: Tổng quan về Kiểm thử phần mềm.

Chương 2: Giới thiệu tổng quan về công cụ Selenium và ngôn ngữ lập trình Python.

Chương 3: Xây dựng các trường hợp kiểm thử.

Chương 4: Thực hiện và chạy tự động kiểm thử cho Website.

## Phương pháp nghiên cứu

* Phương pháp nghiên cứu tài liệu:
* Nghiên cứu về Selenium, Python.
* Tìm hiểu về Unittest Framework.
* Tìm kiếm tài liệu và tham khảo: xây dựng các trường hợp kiểm thử tự động với Selenium và Python.
* Phương pháp thực nghiệm và đánh giá:
* Triển khai và thực hiện kiểm thử: Chạy các kịch bản và ghi nhận kết quả.
* Đánh giá kết quả của các kịch bản kiểm thử và xác định các lỗi hoặc vấn đề phát sinh. Điều chỉnh các kịch bản và thực hiện lại kiểm thử nếu cần thiết.

## Ý nghĩa khoa học và thực tiễn

* Ý nghĩa khoa học:
* Nghiên cứu và phát triển công nghệ: Việc phát triển các trường hợp kiểm thử tự động là một phần của quá trình nghiên cứu và phát triển công nghệ trong lĩnh vực tự động hóa kiểm thử phần mềm.
* Ứng dụng các nguyên lý và phương pháp: Xây dựng các trường hợp kiểm thử tự động yêu cầu hiểu biết sâu rộng về các nguyên lý kiểm thử phần mềm và cách áp dụng chúng vào thực tế.
* Ý nghĩa thực tiễn:
* Đảm bảo chất lượng sản phẩm: Tự động hóa kiểm thử giúp đảm bảo chất lượng của sản phẩm bằng cách phát hiện và báo cáo lỗi một cách nhanh chóng và hiệu quả.
* Tiết kiệm thời gian và chi phí: Tự động hóa kiểm thử giúp tiết kiệm thời gian và chi phí so với việc thực hiện kiểm thử thủ công, đặc biệt là khi áp dụng trong các dự án lớn và phức tạp.
* Tăng tính ổn định và tin cậy: Các trường hợp kiểm thử tự động có thể chạy định kỳ hoặc sau mỗi lần thay đổi mã nguồn, giúp đảm bảo rằng ứng dụng luôn ổn định và tin cậy.

# Tổng quan

## Tổng quan tình hình nghiên cứu

### **Tổng quan tình hình nghiên cứu trong nước**

Nhu cầu kiểm thử phần mềm ngày càng tăng:

* Do sự phát triển mạnh mẽ của ngành công nghiệp phần mềm Việt Nam, nhu cầu kiểm thử phần mềm cũng ngày càng tăng cao.
* Các doanh nghiệp ngày càng quan tâm đến chất lượng phần mềm và coi trọng việc kiểm thử để đảm bảo sản phẩm đáp ứng yêu cầu của khách hàng.

Nghiên cứu khoa học về xây dựng các trường hợp kiểm thử tự động cho website sử dụng Selenium với Python đang ngày càng phát triển. Các nghiên cứu này góp phần nâng cao hiệu quả và chất lượng của việc kiểm thử tự động web, giúp các tổ chức và doanh nghiệp tiết kiệm thời gian và chi phí.

### **Tổng quan tình hình nghiên cứu nước ngoài**

Kiểm thử phần mềm đang ngày càng phát triển và trở thành yếu tố không thể thiếu trong quy trình phát triển phần mềm. Sự tiến bộ của công nghệ, kết hợp với nhu cầu cao về chất lượng phần mềm, đang thúc đẩy ngành này tiến lên mạnh mẽ trên toàn cầu. Với sự áp dụng những kỹ thuật mới như:

* Kiểm thử liên tục (Continuous Testing): kiểm thử liên tục là yếu tố then chốt, giúp đảm bảo chất lượng phần mềm trong quá trình phát triển liên tục.
* Kiểm thử đa nền tảng giúp sản phẩm trên nhiều thiết bị, hệ điều hành và trình duyệt đảm bảo khả năng tương tính và tính đồng bộ hóa.
* Tích hợp CI/CD và AI vào kiểm thử.

## Giới thiệu chung về kiểm thử phần mềm

### **Kiểm thử phần mềm**

* Kiểm thử phần mềm (software testing) là quá trình thực thi một chương trình với mục đích tìm lỗi. Là hoạt động nhằm tìm kiếm và phát hiện ra các lỗi của phần mềm, đảm bảo phần mềm chính xác, đúng và đầy đủ theo yêu cầu của khách hàng, yêu cầu của sản phẩm đã đặt ra. Software testing cũng cung cấp mục tiêu, cái nhìn độc lập về phần mềm điều này cho phép đánh giá và hiểu rõ các rủi ro khi thực thi phần mềm.
* Mục tiêu của kiểm thử phần mềm
* Phát hiện lỗi, đảm bảo chất lượng, Đảm bảo khả năng tương thích của phần mềm với các hệ thống khác.
* Kiểm thử phần mềm không chỉ cần tìm lỗi phần mềm, mà còn là quá trình kiểm
* tra và xác minh một phần mềm đã đáp ứng được yêu cầu và mong đợi của khách hàng.

### **Tại sao phải kiểm thử phần mềm**

Kiểm thử phần mềm rất cần thiết bởi vì nó giúp tìm ra những sai sót, khiếm khuyết trong giai đoạn phát triển, đảm bảo chất lượng sản phẩm, thỏa mãn nhu cầu và tạo niềm tin với khách hàng.

* Giúp tiết kiệm chi phí

Kiểm thử phần mềm mang lại nhiều lợi ích, trong đó, tiết kiệm chi phí là lợi ích nổi bật. Thực tế, vấn đề ưu tiên của các công ty sử dụng dịch vụ kiểm thử phần mềm thường là chi phí.

Đối với bất cứ hệ thống nào, dù đơn giản hay phức tạp, lỗi về thiết kế cũng khó có thể được loại trừ hoàn toàn. Đó không hẳn là bất cẩn của lập trình viên, đôi khi, lỗi xuất hiện là do sự phức tạp của hệ thống. Do đó, kiểm thử phần mềm là vô cùng cần thiết.

Kiểm thử phần mềm bao gồm rất nhiều giai đoạn. Nếu phát hiện và sửa lỗi ở giai đoạn đầu, chắc chắn sẽ tiết kiệm được một khoản chi phí đáng kể. Nếu các vấn đề về thiết kế không được phát hiện sớm, việc chỉnh sửa nó sẽ trở nên khó khăn, tốn kém hơn nhiều.

Đôi khi, trong khi sửa một lỗi, bạn sẽ vô tình phát hiện ra nằm ở các mô-đun khác. Điều đó chứng tỏ rằng, kiểm thử phần mềm ở giai đoạn đầu sẽ giúp tiết kiệm đáng kể về chi phí.

* Chất lượng của sản phẩm

Kiểm thử phần mềm là một phần quan trọng trong quá trình phát triển phần mềm. Kiểm thử góp phần tạo nên một phần mềm chất lượng, đúng thời hạn, xây dựng niềm tin đối với khách hàng và khẳng định danh tiếng của tổ chức thử nghiệm trên thị trường.

Kiểm thử phần mềm góp phần tạo nên một sản phẩm chất lượng

Mang đến cho khách hàng sản phẩm chất lượng sẽ giúp họ có được niềm tin từ người dùng và thu về lợi nhuận không hề nhỏ. Ngược lại, nếu gửi đến khách hàng sản phẩm kém chất lượng, không đáng tin cậy thì khả năng cao họ sẽ lựa chọn tổ chức thử nghiệm khác.

* Sự hài lòng của khách hàng

Mục tiêu cuối cùng của bất cứ hoạt động kinh doanh nào cũng là mang đến cho khách hàng trải nghiệm tuyệt vời. Trong quá trình hợp tác về sau, sự hài lòng của khách hàng thực sự quan trọng. Tuy nhiên, có được sự hài lòng của khách hàng không phải việc dễ dàng.

Kiểm thử phần mềm giúp mang đến cho khách hàng những trải nghiệm tuyệt vời.

Kiểm thử phần mềm góp phần cải thiện trải nghiệm của người dùng. Người dùng hài lòng về sản phẩm phần mềm đồng nghĩa với việc khách hàng hay chính tổ chức kiểm thử có thêm doanh thu và khẳng định danh tiếng.

* Xác định hiệu suất của phần mềm

Kiểm thử phần mềm là lựa chọn đúng đắn và dễ dàng để có thể xác định hiệu suất của phần mềm. Nếu phần mềm hay ứng dụng có hiệu suất thấp, nó sẽ làm giảm đáng kể uy tín của tổ chức kiểm thử.

### **Phân loại về kiểm thử phần mềm**

* Manual Testing

Kiểm thử viên thực hiện thao tác thủ công trên phần mềm.

Phù hợp cho các trường hợp kiểm thử đơn giản, ít lặp lại.

* Automation Testing

Kiểm thử tự động là quá trình thực hiện một cách tự động các bước trong một kịch bản kiểm thử. Kiểm thử tự động bằng một công cụ nhằm rút ngắn thời gian kiểm thử.

Sử dụng công cụ tự động để thực hiện các thao tác kiểm thử.

Phù hợp cho các trường hợp kiểm thử phức tạp, lặp lại nhiều lần.

Tiết kiệm thời gian và công sức.

* Security Testing

Đánh giá mức độ an toàn của phần mềm trước các tấn công.

Phát hiện và khắc phục lỗ hổng bảo mật.

Đảm bảo an toàn cho dữ liệu và hệ thống.

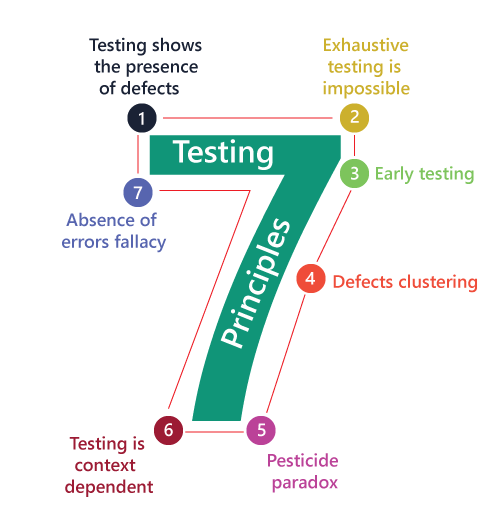
* API Testing

Kiểm tra giao diện lập trình ứng dụng (API) để đảm bảo hoạt động chính xác và hiệu quả.

Phù hợp cho các phần mềm có tích hợp API.

Đảm bảo tính tương thích và khả năng kết nối giữa các ứng dụng

## Nguyên tắc kiểm thử phần mềm



Hình 1: 7 nguyên tắc kiểm thử

1. **Kiểm thử chứng minh sự hiện diện của lỗi**

* Mục đích của kiểm thử là tìm kiếm lỗi
* Chỉ có thể khẳng định rằng phần mềm có lỗi, chứ không thể khẳng định rằng phần mềm không có lỗi.
* Luôn có khả năng tồn tại lỗi mà các trường hợp kiểm thử chưa phát hiện ra.

1. **Kiểm thử toàn bộ là không thể**

* Không thể thực hiện kiểm thử tất cả các trường hợp đầu vào và đầu ra có thể xảy ra.
* Cần tập trung vào việc kiểm thử các trường hợp quan trọng và có khả năng xảy ra lỗi cao.
* Có thể sử dụng các kỹ thuật như phân tích rủi ro để xác định các trường hợp kiểm thử cần thiết.

1. **Kiểm thử sớm**

* Việc phát hiện lỗi sớm giúp tiết kiệm chi phí và thời gian sửa lỗi.
* Nên thực hiện kiểm thử ngay từ giai đoạn đầu của vòng đời phát triển phần mềm.
* Có thể sử dụng các kỹ thuật như kiểm thử đơn vị và kiểm thử tích hợp để kiểm thử sớm.

1. **Gom nhóm lỗi (lỗi thường được phân bố tập chung)**

* Một số phần nhỏ của phần mềm thường chứa nhiều lỗi hơn các phần khác.
* Nên tập trung kiểm thử các phần này để phát hiện nhiều lỗi hơn.
* Có thể sử dụng các kỹ thuật như phân tích mã và theo dõi lỗi để xác định các phần có nhiều lỗi.

1. **Nghịch lý thuốc trừ sâu**

* Nghịch lý thuốc trừ sâu trong testing đề cập đến việc sử dụng bộ test case cũ lặp đi lặp lại có thể dẫn đến việc không phát hiện ra lỗi mới. Giống như côn trùng có thể trở nên kháng thuốc trừ sâu sau một thời gian sử dụng, phần mềm cũng có thể "thích nghi" với bộ test case quen thuộc và che giấu các lỗi mới.

1. **Kiểm thử phụ thuộc vào ngữ cảnh.**

* Quá trình kiểm thử sẽ được thực hiện khác nhau trong các ngữ cảnh khác nhau. Ví dụ, testing web cho ngân hàng chú trọng test về bảo mật, testing web mua hàng thì chú trọng test giao diện

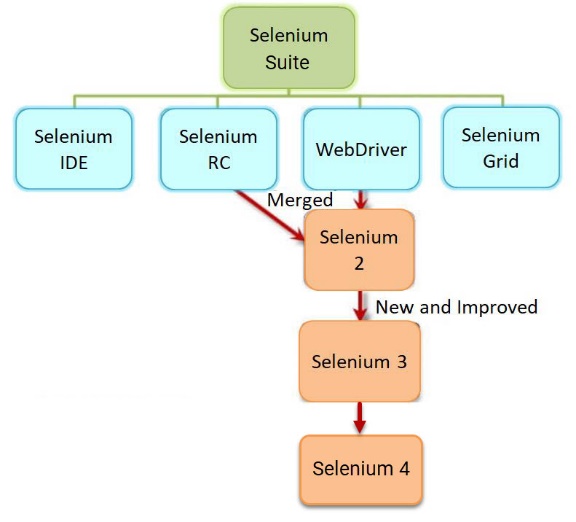
1. **Quan niệm sai lầm về việc không có lỗi.**

* Không có phần mềm nào hoàn toàn không có lỗi.
* Luôn có khả năng tồn tại lỗi mà các trường hợp kiểm thử chưa phát hiện ra.
* Cần thực hiện bảo trì phần mềm thường xuyên để sửa lỗi và cập nhật phần mềm.

# Tổng quan về công cụ kiểm thử

## Công cụ kiểm thử tự động Selenium

### **Phân loại Selenium**



Hình 2: Phân loại Selenium

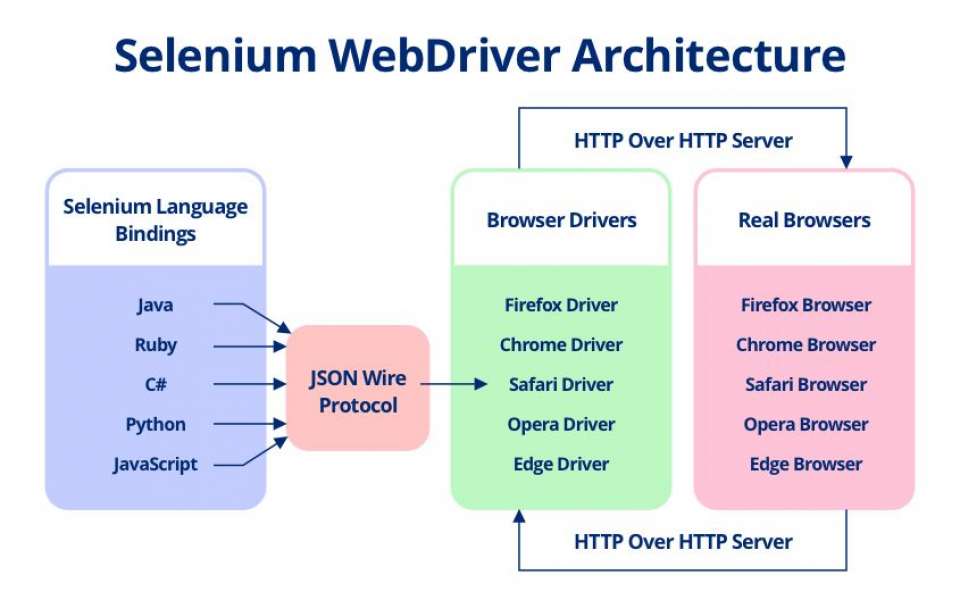
Selenium là một bộ kiểm thử tự động (mã nguồn mở) miễn phí cho các ứng dụng web trên các trình duyệt và nền tảng khác nhau. Selenium chỉ tập trung vào việc tự động hóa các ứng dụng dựa trên web. Việc kiểm thử được sử dụng bằng công cụ Selenium thường được gọi là Automation Selenium Testing.

Selenium ra đời vào năm 2004 tại ThoughtWorks để thử nghiệm một ứng dụng web có tên Time and Expenses của Jason Huggins. Công cụ này được phát triển để kiểm tra hành vi front-end của một ứng dụng trong các trình duyệt khác nhau.

Selenium không chỉ là một công cụ duy nhất mà nó là một bộ phần mềm, mỗi bộ nó cung cấp các nhu cầu thử nghiệm khác nhau của một tổ chức. Nó có 4 thành phần:

* Selenium Integrated Development Environment (Selenium IDE)
* Selenium Remote Control (Selenium RC)
* Selenium WebDriver
* Selenium Grid

### **Selenium Webdriver**



Hình 3: Kiến trúc WebDriver của Selenium

* Selenium WebDriver là gì?

Cốt lõi của Selenium sử dụng API tự động hóa trình duyệt do nhà cung cấp trình duyệt cung cấp để xử lý trình duyệt và chạy thử nghiệm. Nó giống như thể một người dùng thực sự đang vận hành trình duyệt và vì nó không cần API của nó được biên dịch cùng với mã ứng dụng nên nó không xâm phạm. Selenium WebDriver cho phép kiểm tra khả năng tương thích giữa nhiều trình duyệt và kiểm soát các trình duyệt bằng cách giao tiếp trực tiếp với chúng.

* Nó hỗ trợ hầu hết tất cả các ngôn ngữ lập trình bao gồm Java, Python, C#, Perl, Ruby và PHP.
* Selenium WebDriver hỗ trợ các hệ điều hành: Windows, Mac OS, Linux.
* Nó hỗ trợ các trình duyệt Firefox, Chrome, Safari, Internet Explorer, Microsoft Edge.
* Nó cho phép bạn thực hiện các bài kiểm tra trên nhiều trình duyệt .
* Selenium WebDriver hoạt động như thế nào?

Selenium WebDriver hoạt động như một trung gian hòa giải và cho phép mã giao tiếp với các trình điều khiển trình duyệt khác nhau. Khi Selenium được sử dụng để chạy thử nghiệm, các trường hợp thử nghiệm sẽ được tạo bằng cách sử dụng bộ định vị phần tử được định vị bằng kỹ thuật định vị phần tử Selenium. Sau đó, việc sử dụng các hành động Selenium WebDriver có thể được thực hiện trên các phần tử đó. Tóm lại Selenium WebDriver hoạt động theo ba bước:

* Các lệnh kiểm tra được chuyển đổi thành yêu cầu HTTP bằng giao thức dây JSON.
* Trước khi thực hiện bất kỳ trường hợp thử nghiệm nào, mỗi trình duyệt đều có trình điều khiển riêng để khởi chạy máy chủ.
* Trình duyệt sau đó bắt đầu nhận yêu cầu thông qua trình điều khiển của nó.

### **Cài đặt và sử dụng Selenium**

* Cài đặt Selenium

Khi Python được cài đặt trên hệ thống của bạn, bạn có thể sử dụng pip để tải xuống gói Selenium. Lệnh để cài đặt Selenium bằng pip là

pip install selenium

Cú pháp tương tự có thể được sử dụng cho các cài đặt gói khác bằng cách thay thế selenium bằng một tên gói khác, được thảo luận trong một chương khác của cuốn sách này. Phiên bản Selenium được sử dụng là 4.0. Nếu phiên bản thấp hơn của thư viện Selenium đã được cài đặt hoặc nếu bạn cần nâng cấp phiên bản hiện tại, hãy sử dụng lệnh sau.

pip install --upgrade selenium

* Cài đặt Drivers

Cần có trình điều khiển để tương tác với trình duyệt được chỉ định. Mỗi trình duyệt có một Selenium WebDriver cụ thể được liên kết với nó, vì vậy bạn cần cài đặt bất kỳ trình điều khiển nào được cung cấp.

Ví dụ: geckodriver là một Selenium WebDriver chỉ hoạt động với Mozilla Firefox.

|  |  |
| --- | --- |
| Browser Driver | Nguồn cài đặt |
| Google Chrome | https://sites.google.com/a/chromium.org/chromedriver/ |
| Mozilla Firefox | https://github.com/mozilla/geckodriver/releases |
| Internet Explorer | https://www.selenium.dev/downloads/ |
| Microsoft Edge | https://developer.microsoft.com/en-us/microsoft-edge/tools/ webdriver / |

Trình điều khiển web và nguồn cài đặt

* Ưu điểm Selenium Webdriver
* Tích hợp với nhiều ngôn ngữ lập trình: Selenium WebDriver hỗ trợ nhiều ngôn ngữ lập trình phổ biến như Java, Python, C#, Ruby, và JavaScript, cho phép nhà phát triển sử dụng ngôn ngữ ưa thích của mình.
* Hỗ trợ nhiều trình duyệt: Selenium WebDriver có thể kiểm soát nhiều trình duyệt phổ biến như Chrome, Firefox, IE, Edge, Safari, Opera, v.v.
* Tính linh hoạt: Selenium WebDriver cho phép bạn tạo và chạy các kịch bản kiểm thử trên nhiều nền tảng và môi trường khác nhau.
* Cộng đồng lớn và hỗ trợ mạnh mẽ: Selenium có một cộng đồng lớn của các nhà phát triển và kiểm thử viên, cung cấp nhiều tài liệu, hướng dẫn và hỗ trợ trực tuyến.
* Open-source: Selenium WebDriver là một dự án mã nguồn mở, điều này có nghĩa là nó miễn phí và có thể tùy chỉnh theo nhu cầu của bạn.
* Nhược điểm
* Không hỗ trợ kiểm thử hình ảnh và video: Selenium WebDriver không có tính năng tích hợp cho việc kiểm thử hình ảnh và video, điều này có nghĩa là bạn cần sử dụng các công cụ bên ngoài để kiểm tra hiển thị và hoạt động của ứng dụng web.
* Khả năng ổn định phụ thuộc vào trình duyệt và môi trường: Sự ổn định của kịch bản kiểm thử Selenium có thể bị ảnh hưởng bởi cập nhật của trình duyệt hoặc thay đổi trong môi trường kiểm thử.
* Khó khăn trong việc xử lý các trường hợp ngoại lệ: Selenium không cung cấp một cách tiếp cận dễ dàng để xử lý các trường hợp ngoại lệ hoặc thông báo động mà không cần phải sử dụng mã nguồn.
* Tốc độ chậm hơn so với giải pháp khác: Do Selenium WebDriver hoạt động trực tiếp với trình duyệt, nó có thể chậm hơn so với các công cụ kiểm thử không đồng bộ khác.
* Yêu cầu kỹ năng lập trình: Sử dụng Selenium WebDriver đòi hỏi kiến thức về lập trình, điều này có thể là một thách thức đối với những người không có kinh nghiệm lập trình.

## Ngôn ngữ lập trình Python

### **Cài đặt ngôn ngữ lập trình**

Python là ngôn ngữ lập trình máy tính bậc cao thường được sử dụng để xây dựng trang web và phần mềm, tự động hóa các tác vụ và tiến hành phân tích dữ liệu. Python là ngôn ngữ có mục đích chung, nghĩa là nó có thể được sử dụng để tạo nhiều chương trình khác nhau và không chuyên biệt cho bất kỳ vấn đề cụ thể nào.

Bạn sẽ thấy Python được cài đặt mặc định trên hầu hết các bản phân phối Linux, Mac OS X và các máy Unix khác. Trên Windows, bạn sẽ cần phải cài đặt riêng. Bạn có thể tìm thấy trình cài đặt cho các nền tảng khác nhau tại

http://python.org/download/.

### **Cài đặt môi trường**

Bây giờ em đã thiết lập Python và Selenium WebDriver, chúng ta sẽ cần một trình soạn thảo hoặc Môi trường phát triển tích hợp (IDE) để viết các tập lệnh tự động hóa. Một trình soạn thảo hoặc IDE tốt làm tăng năng suất và giúp thực hiện nhiều việc khác giúp trải nghiệm mã hóa trở nên đơn giản và dễ dàng. Mặc dù chúng ta có thể viết mã Python trong các trình soạn thảo đơn giản như Emacs, Vim hoặc Notepad, nhưng sử dụng IDE sẽ giúp cuộc sống dễ dàng hơn rất nhiều.

* Cài đặt Visual Studio Code:
* Truy cập trang chính thức của VSCode: https://code.visualstudio.com/.
* Tải và cài đặt phiên bản mới nhất cho hệ điều hành của bạn.

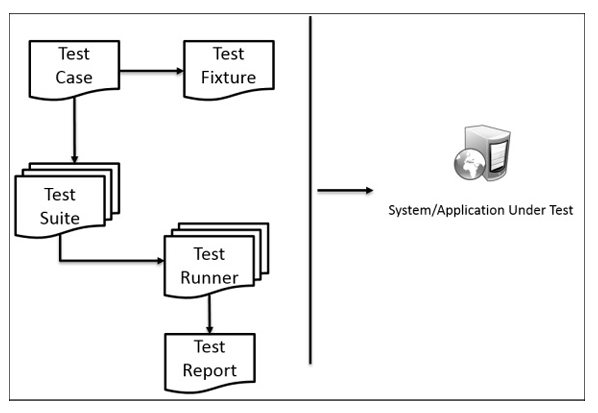
## Unittest Framework

### **Tổng quan về Unittest**

Thư viện unittest (ban đầu được đặt tên là PyUnit) được lấy cảm hứng từ thư viện JUnit được sử dụng rộng rãi trong phát triển ứng dụng Java. Chúng tôi có thể sử dụng unittest để tạo ra một bộ kiểm thử toàn diện cho bất kỳ dự án nào. Mô-đun unittest được sử dụng trong dự án Python để kiểm tra các mô-đun thư viện tiêu chuẩn khác nhau bao gồm cả unittest. Bạn có thể tìm thấy tài liệu unittest tại

http://docs.python.org/2/library/unittest.html.

Thư viện unittest cung cấp cho chúng ta khả năng tạo các trường hợp kiểm thử, bộ kiểm thử và đồ đạc kiểm thử. Hãy hiểu từng thành phần này như thể hiện trong sơ đồ sau:



Hình 25: Sơ đồ thể hiện Unittest

* Test Fixture: Bằng cách sử dụng vật cố thử nghiệm, chúng tôi có thể xác định sự chuẩn bị cần thiết để thực hiện một hoặc nhiều thử nghiệm và bất kỳ hành động dọn dẹp nào liên quan.
* Test Case: Test case là đơn vị kiểm thử nhỏ nhất trong unittest. Nó kiểm tra một phản hồi cụ thể cho một tập hợp các hành động và đầu vào cụ thể bằng cách sử dụng các phương thức assert khác nhau được cung cấp bởi thư viện unittest. Thư viện unittest cung cấp một lớp cơ sở được gọi là TestCase có thể được sử dụng để tạo các trường hợp kiểm thử mới.
* Test Suite: Test Suite là tập hợp nhiều test hoặc test case để tạo các nhóm test đại diện cho chức năng hoặc module cụ thể của ứng dụng đang test, sẽ được thực thi cùng nhau.
* Test Runner: Người chạy thử nghiệm sắp xếp thực hiện các bài kiểm tra và cung cấp kết quả cho người dùng. Người chạy có thể sử dụng giao diện đồ họa, giao diện văn bản hoặc trả về một giá trị đặc biệt để cho biết kết quả thực hiện các bài kiểm tra.
* Test Report: Báo cáo kiểm thử hiển thị tóm tắt kết quả kiểm thử hiển thị trạng thái đạt hoặc không đạt của các trường hợp kiểm thử được thực hiện, kết quả dự kiến so với kết quả thực tế cho các bước không thành công và tóm tắt thông tin thời gian và chạy tổng thể.

### **Chạy kết quả**

Chạy thử nghiệm Để chạy test từ dòng lệnh, chúng ta có thể thêm một lệnh gọi vào phương thức chính của test case. Chúng ta sẽ truyền đối số verbosity được sử dụng để hiển thị số lượng chi tiết kết quả kiểm tra trên bảng điều khiển:

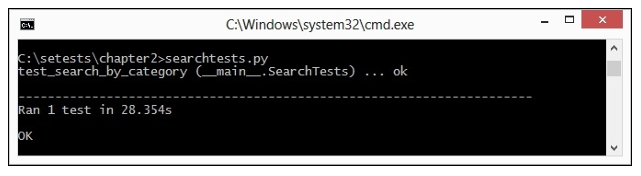
if \_\_name\_\_ == '\_\_main\_\_':

unittest.main(verbosity=2)

Chúng ta có thể lưu trữ các bài kiểm tra dưới dạng tập lệnh Python bình thường.

Trong ví dụ này, hãy lưu thử nghiệm mẫu dưới dạng login\_test.py. Sau khi lưu tệp, chúng ta có thể thực thi nó thông qua dòng lệnh bằng cách sử dụng lệnh sau:

python login\_test.py



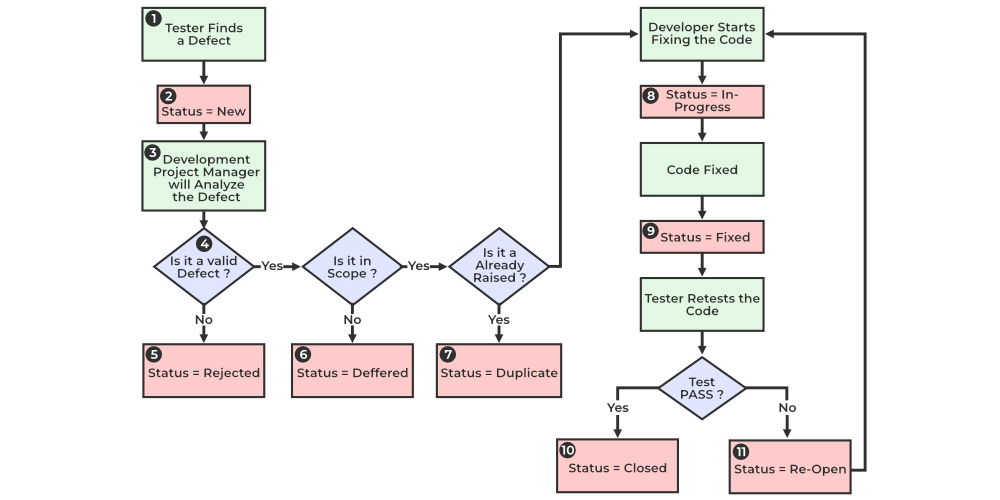
Hình 26: Kết quả hiển thị sau khi chạy

# XÂY DỰNG TRƯỜNG HỢP KIỂM THỬ

## Trường hợp kiểm thử

### **BUG LIFE CYCLE (vòng đời lỗi)**

Giải thích vòng đời lỗi



Hình 4: Giải thích vòng đời lỗi

Vòng đời lỗi trong kiểm thử phần mềm là tập hợp các trạng thái cụ thể mà lỗi hoặc lỗi phải trải qua trong suốt vòng đời của nó. Mục đích của vòng đời Khiếm khuyết là để dễ dàng điều phối và truyền đạt trạng thái hiện tại của khiếm khuyết thay đổi đối với những người được giao khác nhau và làm cho quá trình sửa lỗi có hệ thống và hiệu quả.

1. Người kiểm tra tìm thấy lỗi
2. Trạng thái được gán cho lỗi - New
3. Một lỗi được chuyển tiếp đến Người quản lý dự án để phân tích
4. Người quản lý dự án quyết định xem lỗi có hợp lệ hay không
5. Ở đây lỗi không hợp lệ - trạng thái được đưa ra là Rejected (bị từ chối)
6. Vì vậy, người quản lý dự án gán trạng thái bị từ chối . Nếu lỗi không bị từ chối thì bước tiếp theo là kiểm tra xem nó có nằm trong phạm vi hay không. Giả sử chúng ta có một chức năng khác - chức năng email cho cùng một ứng dụng và bạn thấy có vấn đề với chức năng đó. Nhưng nó không phải là một phần của bản phát hành hiện tại khi những lỗi như vậy được chỉ định là trạng thái Deffered (trì hoãn).
7. Tiếp theo, người quản lý xác minh xem lỗi tương tự có xuất hiện trước đó hay không. Nếu có lỗi sẽ được gán trạng thái Duplicate (trùng lặp).
8. Nếu không, lỗi sẽ được giao cho nhà phát triển, người bắt đầu sửa mã. Trong giai đoạn này, lỗi được gán trạng thái In- progress (đang tiến hành).
9. Sau khi mã được sửa. Một lỗi được gán trạng thái cố định
10. Tiếp theo, người kiểm tra sẽ kiểm tra lại mã. Trong trường hợp Test Case vượt qua lỗi được Closed (đã đóng). Nếu các trường hợp thử nghiệm lại thất bại, lỗi sẽ được mở lại và gán cho nhà phát triển.
11. Hãy xem xét tình huống trong lần phát hành đầu tiên của Đặt chỗ chuyến bay, người ta đã tìm thấy một lỗi trong đơn hàng Fax đã được khắc phục và được gán trạng thái đã đóng. Trong lần nâng cấp thứ hai, lỗi tương tự lại xuất hiện. Trong những trường hợp như vậy, một lỗi đã đóng sẽ Re-opened (được mở lại).

### **Kỹ thuật viết trường hợp kiểm thử**

Test Case hay còn được gọi là trường hợp kiểm thử được sử dụng trong ngành kiểm thử phần để kiểm tra ứng dụng, phần mềm cụ thể nào đó có hoạt động hay không. Kết thúc quá trình test case sẽ cung cấp cho doanh nghiệp những thông tin về kết quả, chức năng hoạt động của ứng dụng.

Một bộ Test Case sẽ mô tả những yếu tố sau:

* Dữ liệu đầu vào (Input).
* Hành động (Active) – Sự kiện (Event).
* Kết quả đầu ra (Expected response).

Các trường hợp cần viết Test Case như sau:

* **Trước khi phát triển:** Ở giai đoạn này, bạn cần xác định được yêu cầu của sản phẩm, phần mềm và tiến hành kiểm thử sau khi các yếu tố này được phát triển.
* **Trong quá trình phát triển:** Khi một phần của mô-đun, phần mềm được phát triển, bạn sẽ cần Test Case đảm bảo các ứng dụng đang đáp ứng đúng yêu cầu của khách hàng.
* **Sau khi phát triển:**Phần mềm sau khi phát triển hoàn thiện tính năng cần được viết Test Case để kiểm tra hoạt động và tính năng cụ thể trước khi gửi cho khách hàng.

### **Cấu trúc Test Cases**

|  |  |
| --- | --- |
| Thành phần | Chức năng |
| Test Case ID | Mã định danh duy nhất cho mỗi trường hợp kiểm thử |
| TestCase Description | Phần này mô tả mục tiêu cụ thể của Test case, tức là chức năng đang được kiểm tra. |
| Pre-condition / Data | Đây là dữ liệu cần thiết để thực hiện test. |
| Step | Phần này mô tả chi tiết các bước thực hiện trong quá trình kiểm thử. |
| Expected Output | Kết quả mong đợi sau khi thực hiện kiểm thử |
| Actual Output | Kết quả thực tế thu được sau khi thực hiện kiểm thử |
| Status | Trạng thái của Test Case sau khi kiểm thử, Passed (Đạt) hoặc Failed (Không đạt) |
| Test Date | Ngày thực hiện kiểm thử |
| Tester | Tên người thực hiện kiểm thử |

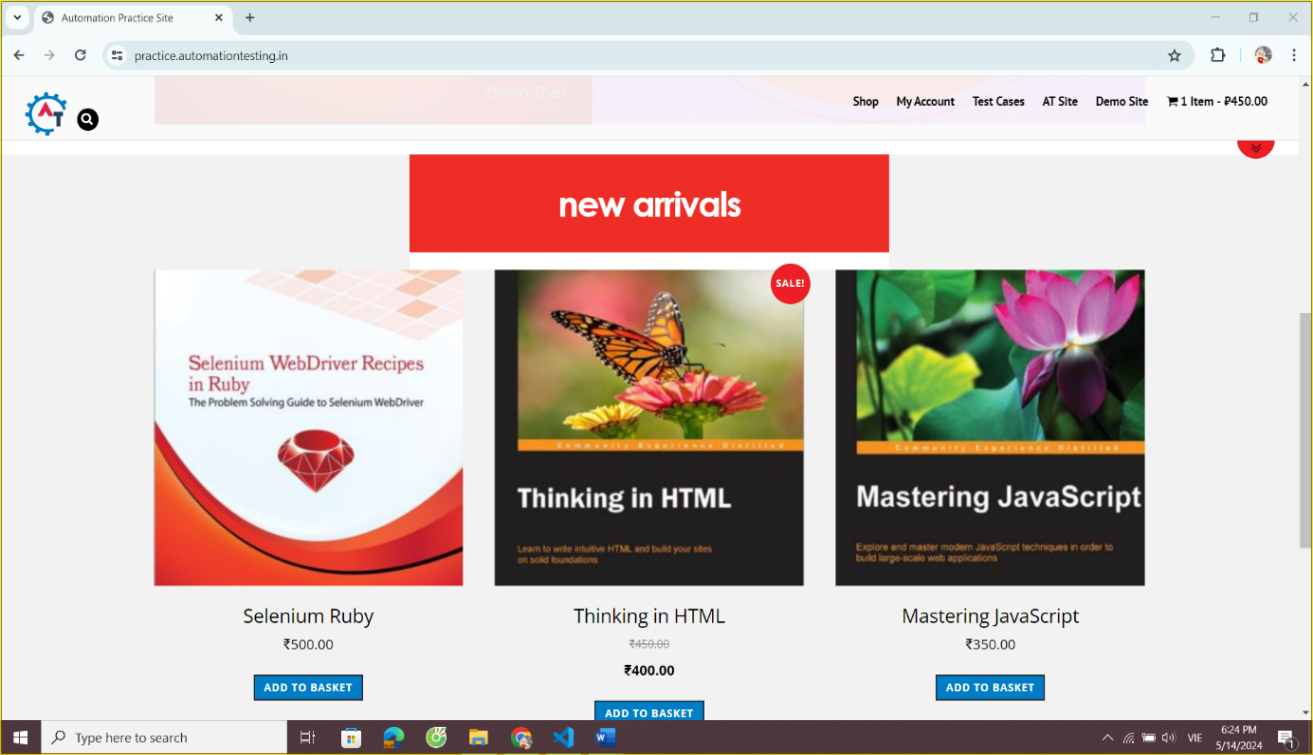
Bảng 3.1.3: Cấu trúc Test Case

## Viết trường hợp kiểm thử giao diện

### **Test Case giao diện Trang chủ**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test case Description** | **Actual Output** |
|  |
|  |
| **TC\_01** | Home screen | Show website homepage |  |
| **TC\_02** | Button Search | 1. Display the Search button 2. Navigate to the Search Enter page 3. "https://practice.automationtesting.in/?s=selenium" |  |
| Seach button: search icon, colors black, located in Header |  |
| **TC\_03** | Button Shop | 1. Display the Shop button 2. Navigate to the Shop page 3. "https://practice.automationtesting.in/shop/" 4. Shop button: colors red, located in Header |  |
| Shop button: colors black, located in Header |  |
| **TC\_04** | Button My Account | 1. Display the Shop button 2. Navigate to the My Accout page 3."https://practice.automationtesting.in/my-account/" 4. My Accout button: colors red, located in Header |  |
| My Accout button: colors black, located in Header |  |
| **TC\_05** | Button Test Cases | 1. Display the Test Cases button 2. Navigate to the Test Cases page 3."https://practice.automationtesting.in/test-cases/" 4. Test Cases button: colors red, located in Header |  |
| Test Cases button: colors black, located in Header |  |
| **TC\_06** | Button AT Site | 1. Display the AT Site button 2. Navigate to the AT Site page 3. "https://automationtesting.in/" |  |
| AT Site button: colors black, located in Header |  |
| **TC\_07** | Button Demo Site | 1. Display the Demo Site button 2. Navigate to the Demo Site page 3. "https://demo.automationtesting.in/Register.html" |  |
| Demo Site button: colors black, located in Header |  |
| **TC\_08** | Button Shopping | 1. Display the Shopping button 2. Navigate to the Shopping page 3. "https://practice.automationtesting.in/shop/" |  |
| Shopping button: shopping-cart icon, colors black, located in Header |  |

Bảng 3.2.1: TestCase giao diện Trang chủ

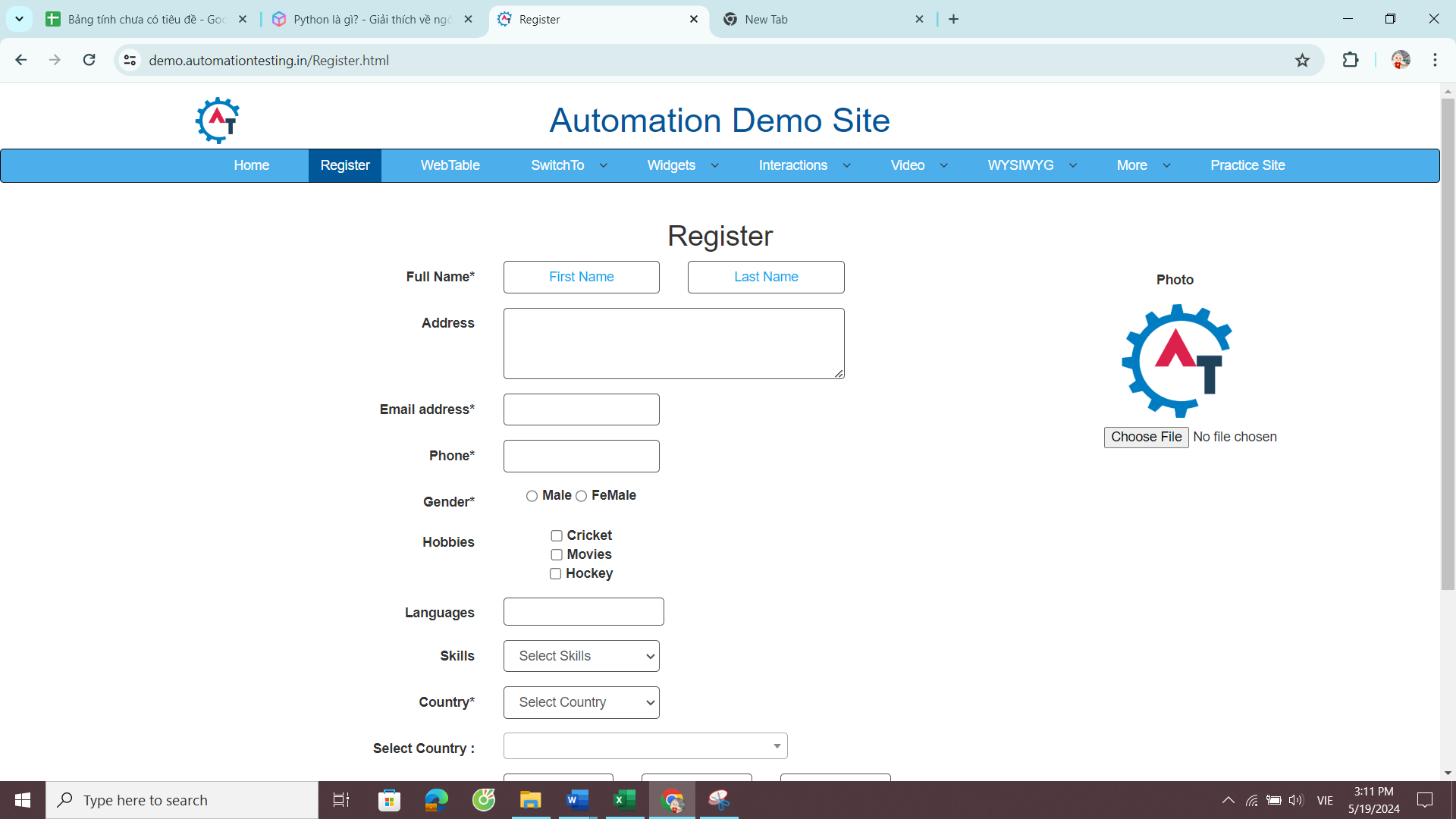
****

Hình 3.2.1: Giao diện Trang chủ

### **Test Case giao diện Trang Demo Site**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test case Description** | **Actual Output** |
| **TC\_01** | Button Register with Photo Upload Images | Successfully implemented, done all the steps |
| **TC\_02** | Button Interactions with Drag and Drop | Successfully implemented, done all the steps |
| **TC\_03** | Button WYSIWYG with SummerNote | Successfully implemented, done all the steps |
| **TC\_04** | Button More with Dynamic Data | Successfully implemented, done all the steps |
| **TC\_05** | Button More with File Download | Successfully implemented, done all the steps |
| **TC\_06** | Button More with File Upload | Successfully implemented, done all the steps |

Bảng 3.2.2: TestCase giao diện Trang Demo Site

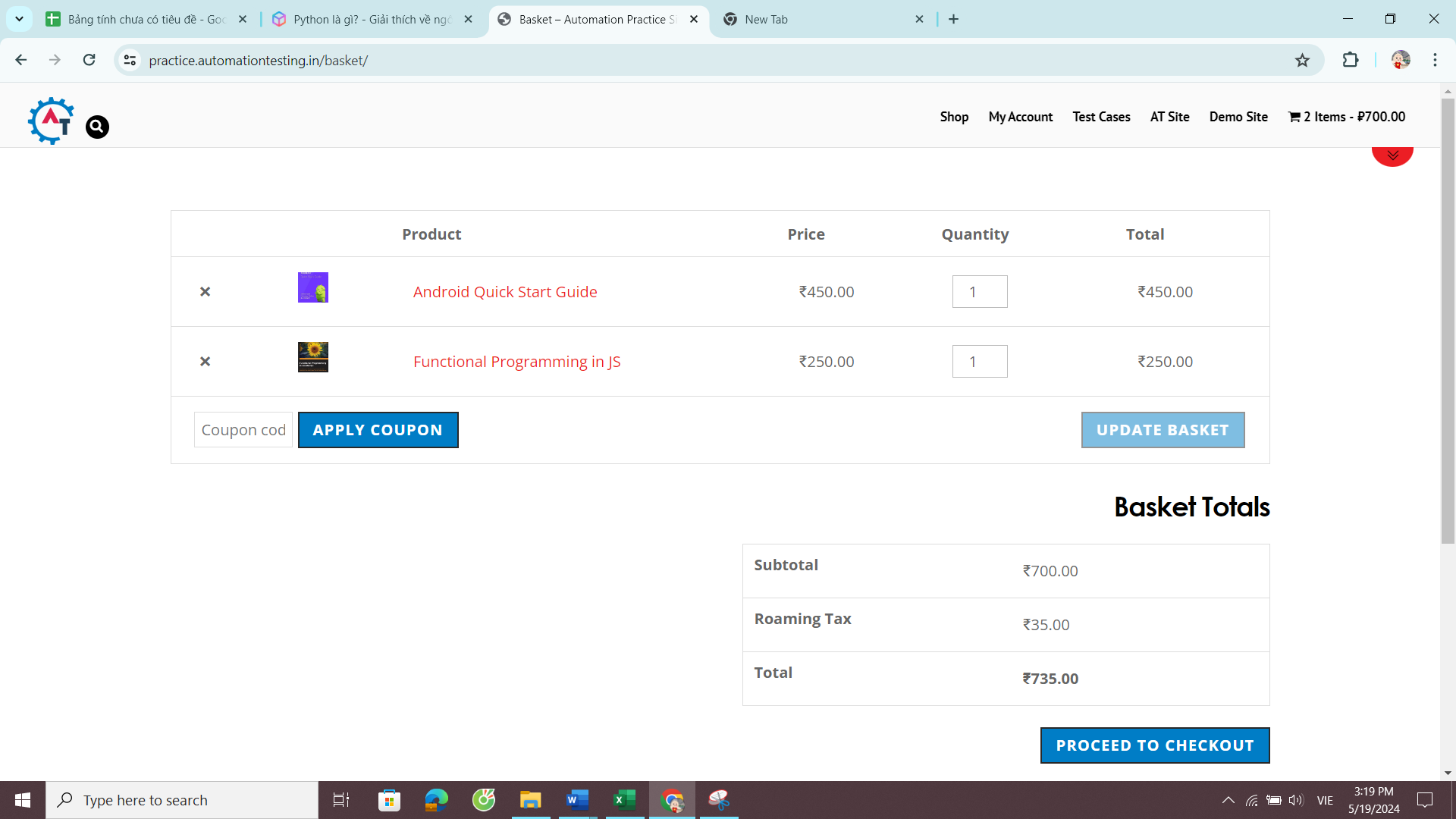


Hình 3.2.2: Giao diện Trang Demo Site

### **Test Case giao diện Thêm sản phẩm vào giỏ hàng**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test case Description** | **Actual Output** |
| **TC\_01** | Add products to cart | Successfully implemented, done all the steps |
| **TC\_02** | Update products in the cart | Successfully implemented, done all the steps |
| **TC\_03** | Remove products in the cart | Successfully implemented, done all the steps |

Bảng 3.2.3: Test Case Thêm vào giỏ hàng



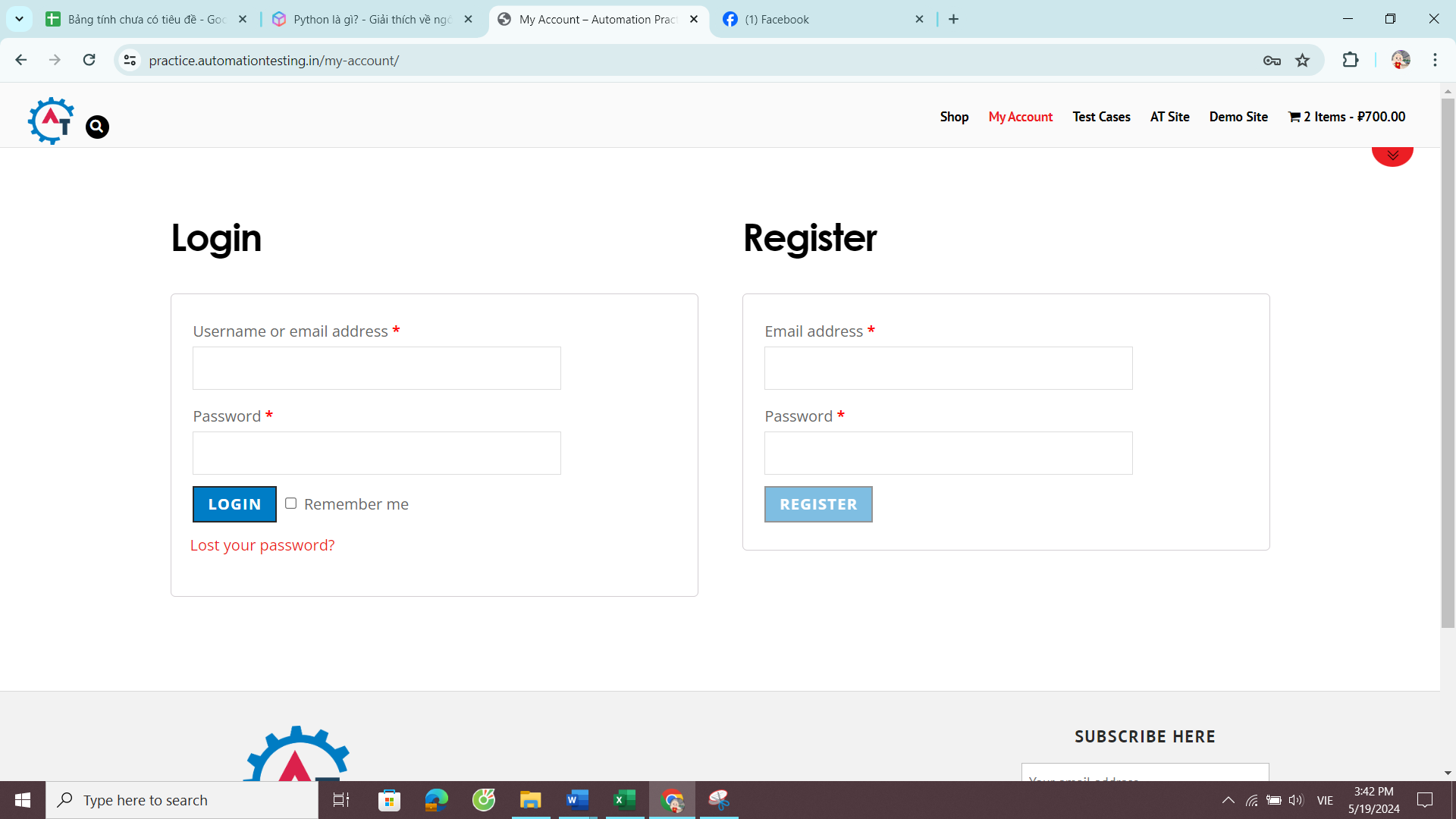
Hình 3.2.3: Giao diện Thêm vào giỏ hàng

## Viết trường hợp kiểm thử chức năng website

### **Test Case chức năng Đăng ký**

|  |  |  |
| --- | --- | --- |
| **TC ID** | **Test case Description** | **Actual Output** |
| **TC\_01** | Successful registration With the correct email and password | Successful registration, navigated to the Home page |
| **TC\_02** | Registration failed With empty email and empty password | Successful registration, navigated to the Home page |
| **TC\_03** | Registration failed With invalid Email | Registration must fail, Display the message "Error: Please provide a valid email address." |
| **TC\_04** | Registration failed With a weak password | Registration must fail, Display the message "Very weak - Please enter a stronger password." |
| **TC\_05** | Registration failed With invalid email and weak password | Registration must fail, Display the message "Error: Please provide a valid email address." |

Bảng 3.3.1: Test Case chức năng Đăng ký

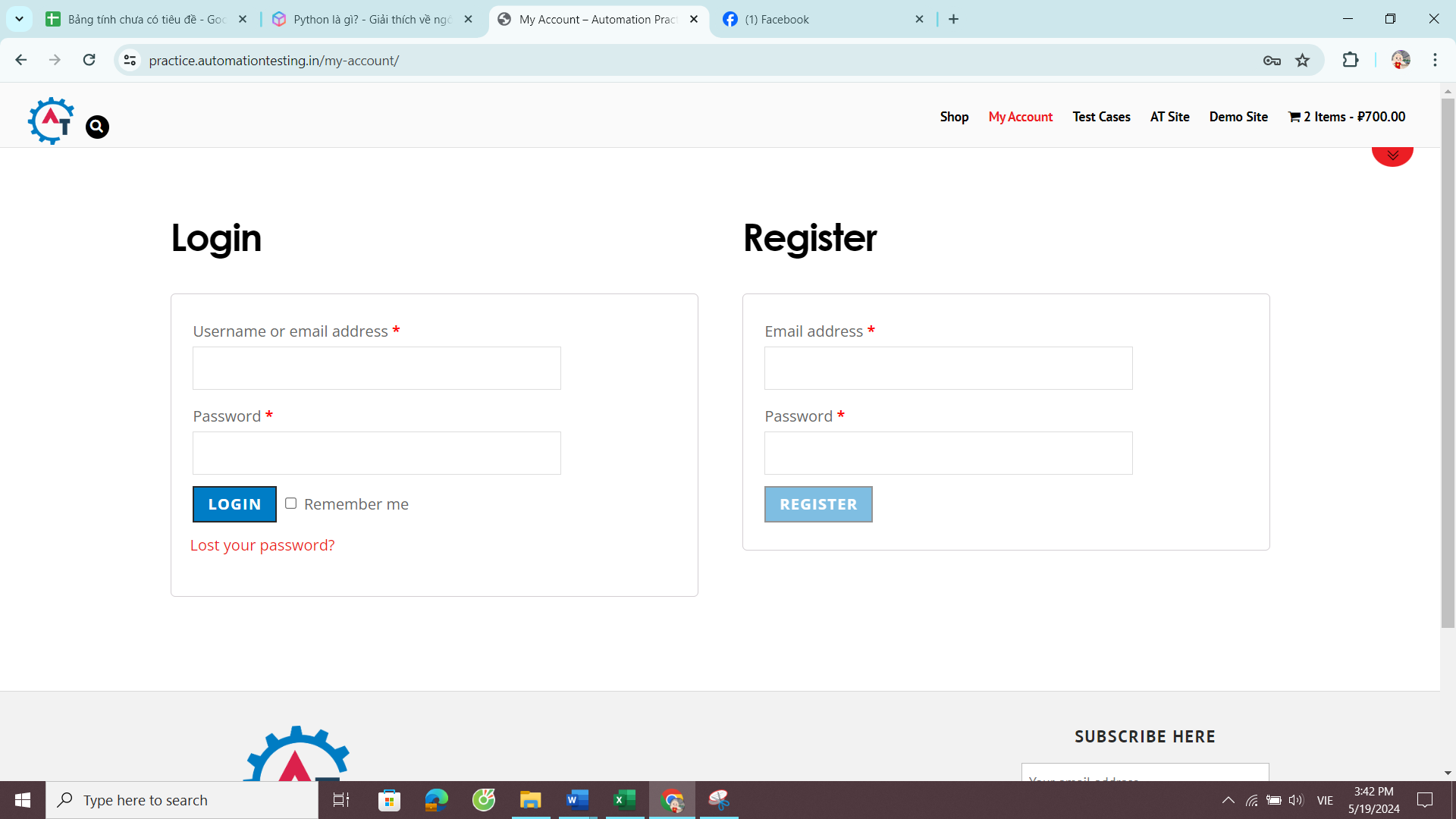


Hình 3.3.1: Giao diện trang Đăng ký

### **Test Case chức năng Đăng nhập**

|  |  |  |
| --- | --- | --- |
| **TC ID** | **Test case Description** | **Actual Output** |
| **TC\_01** | Successful Login With the correct email address and password | Login Successful, switch to user page (Hello dhuonggiang566) |
| **TC\_02** | Login failed With empty email and empty password | Login must fail, Display the message "Error: Username is required." |
| **TC\_03** | Login failed With empty email and correct password | Login must fail, Display the message "Error: Username is required." |
| **TC\_04** | Login failed With correct email and empty password | Login must fail, Display the message "Error: Password is required." |
| **TC\_05** | Login failed With incorrect email and incorrect password | Login must fail, Display the message "Error: A user could not be found with this email address." |
| **TC\_06** | Login failed With incorrect email and correct password | Login must fail, Display the message "Error: A user could not be found with this email address." |
| **TC\_07** | Login failed With correct email and incorrect password | Login must fail, Display the message "Error: The password you entered for the username dhuonggiang566@gmail.com is incorrect. Lost your password?" |

Bảng 3.3.2: Test Case chức năng Đăng nhập

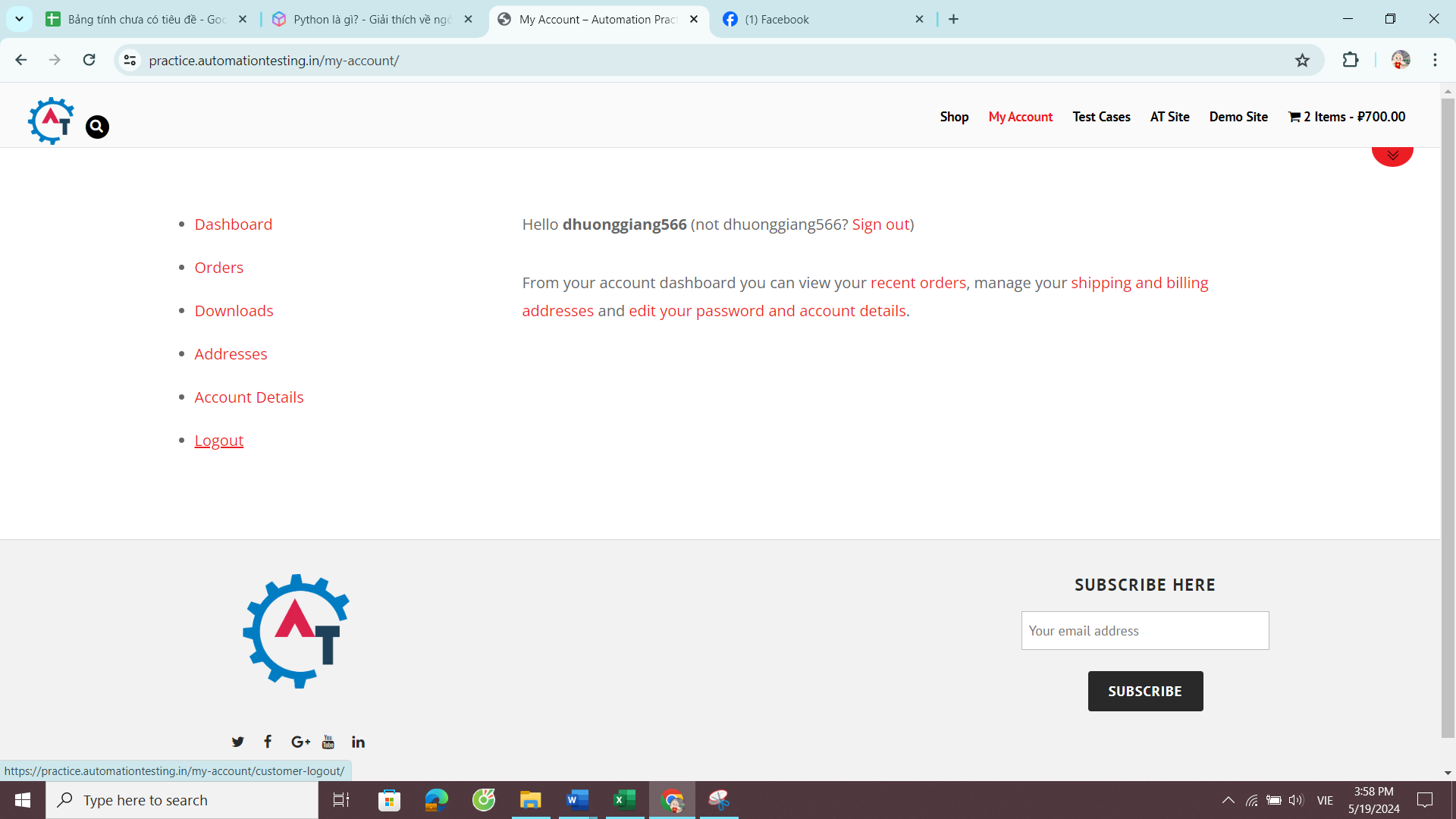


Hình 3.3.2: Giao diện trang Đăng nhập

### **Test Case chức năng Đăng xuất**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test case Description** | **Actual Output** |
| **TC\_01** | Log out successfully | Exit the user page. Display to url "https://practice.automationtesting.in/my-account/ |

Bảng 3.3.3: Test Case chức năng Đăng xuất

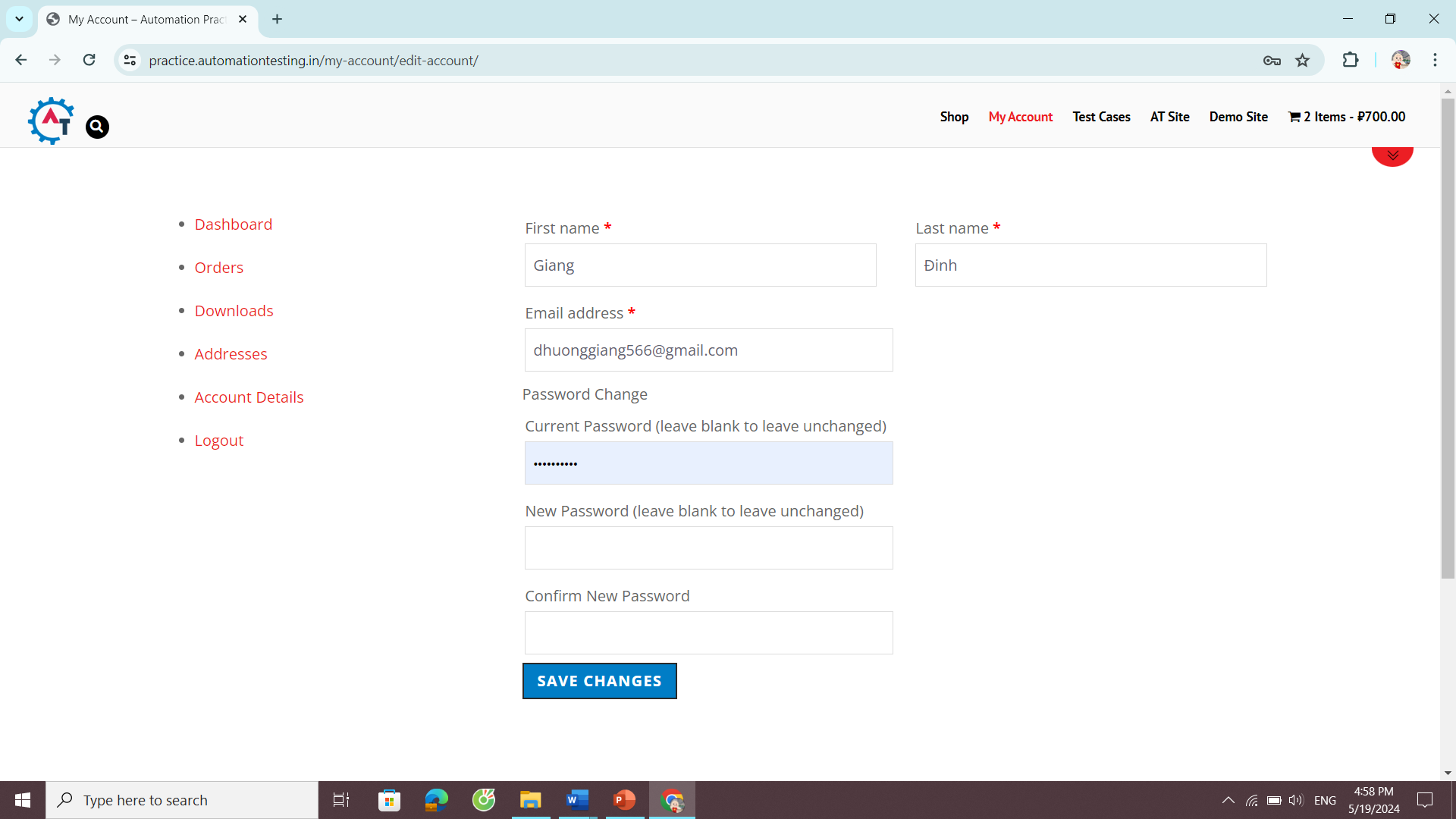


Hình 3.3.3: Giao diện trang Đăng xuất

### **Test Case chức năng Cập nhật tài khoản**

|  |  |  |
| --- | --- | --- |
| **TC ID** | **Test case Description** | **Actual Output** |
| **TC\_01** | Successful Change With name change and password change | Successful Change, display the message "Account details changed successfully." |
| **TC\_02** | Successful Change With name change | Successful Change, display the message "Account details changed successfully." |
| **TC\_03** | Change failed with empty First name, empty Last name | Display the message "First Name is a required field. Last Name is a required field." |
| **TC\_04** | Change failed with entering the wrong current password | Registration must fail, Display the message "Your current password is incorrect." |
| **TC\_05** | Change failed with a weak New Password | Registration must fail, Display the message "Weak - Please enter a stronger password." |
| **TC\_06** | Change failed with Confirm New Password does not match New Password | Registration must fail, Display the message "New passwords do not match." |
| **TC\_07** | Change failed with no change | Account Details page |

Bảng 3.3.4: Test Case chức năng Cập nhật tài khoản

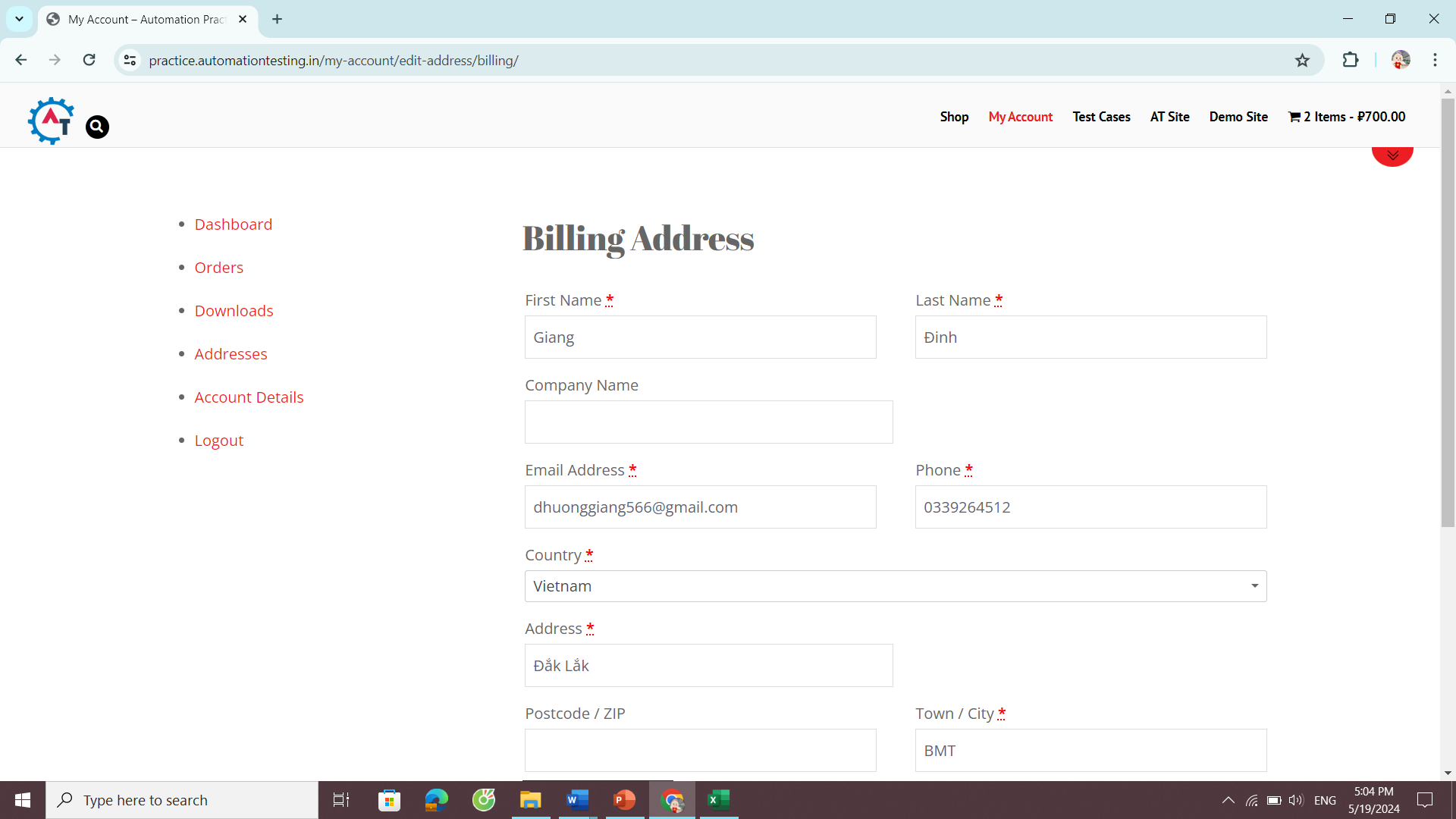


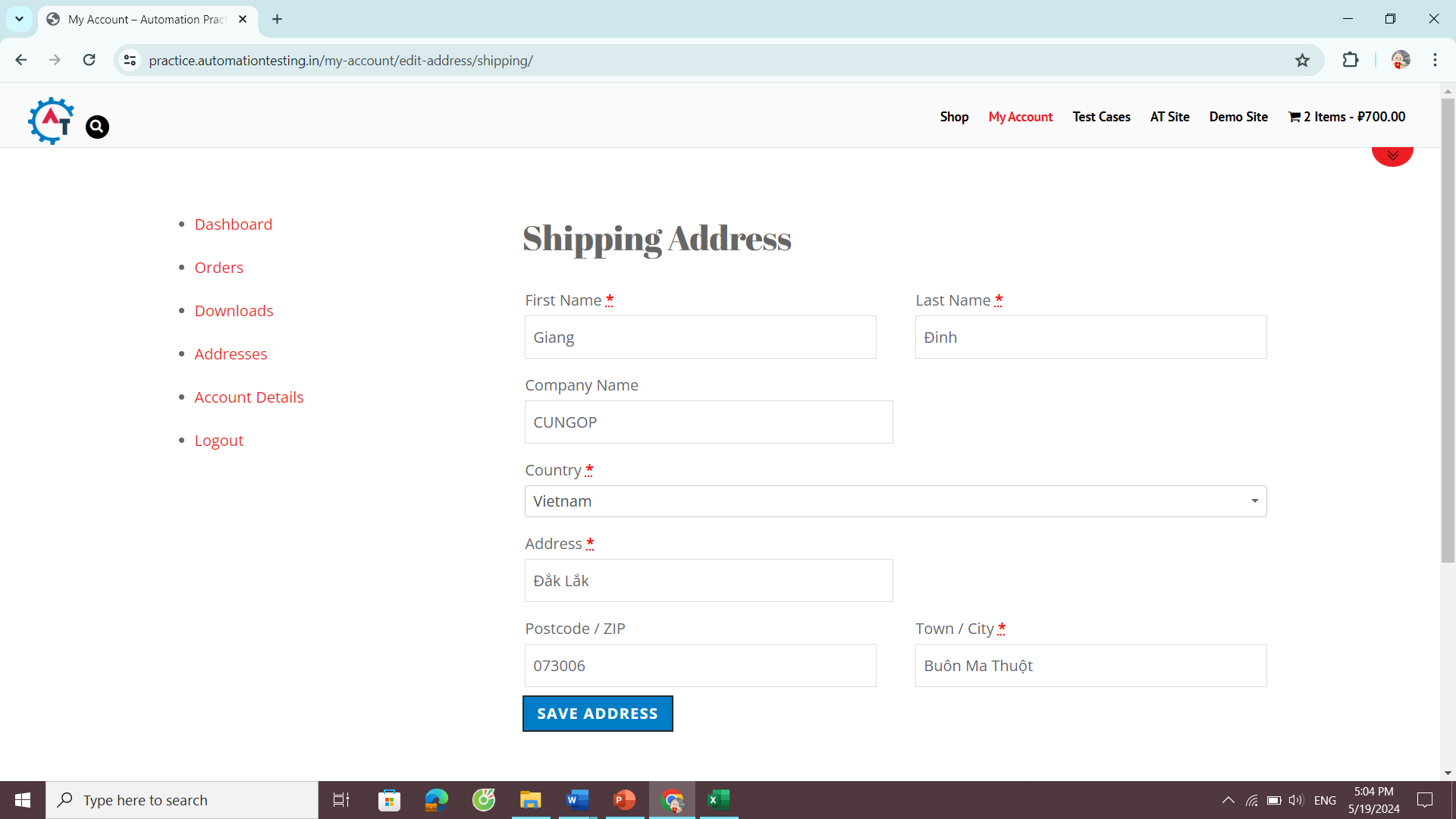
Hình 3.3.4: Giao diện trang Cập nhật tài khoản

### **Test Case chức năng Thêm địa chỉ giao hàng**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test case Description** | **Actual Output** |
| **TC\_01** | Successful Add Billing Address with the correct full enter | Successful Add, display the message "Address changed successfully …" |
| **TC\_02** | Successful Add Billing Address with leave the item blank optional | Successful Add, display the message "Address changed successfully ..." |
| **TC\_03** | Add Billing Address failed with empty First name, empty Last name | Add must fail, Display the message "First Name is a required field. Last Name is a required field." |
| **TC\_04** | Add Billing Address failed with empty Town/City, empty Phone | Add must fail, Display the message "Phone is a required field. Town / City is a required field." |
| **TC\_05** | Successful Add Shipping Address with the correct full enter | Successful Add, display the message "Address changed successfully ..." |
| **TC\_06** | Add Shipping Address failed with empty First name, empty Last name | Add must fail, Display the message "First Name is a required field. Last Name is a required field." |
| **TC\_07** | Add Shipping Address failed with empty Address, empty Town/City | Add must fail, Display the message "Address is a required field. Town / City is a required field." |
| **TC\_08** | Add Shipping Address failed with invalid Postcode/ZIP | Add must fail, Display the message "Please enter a valid postcode/ZIP ..." |

Bảng 3.3.5: Test Case chức năng Thêm địa chỉ giao hang



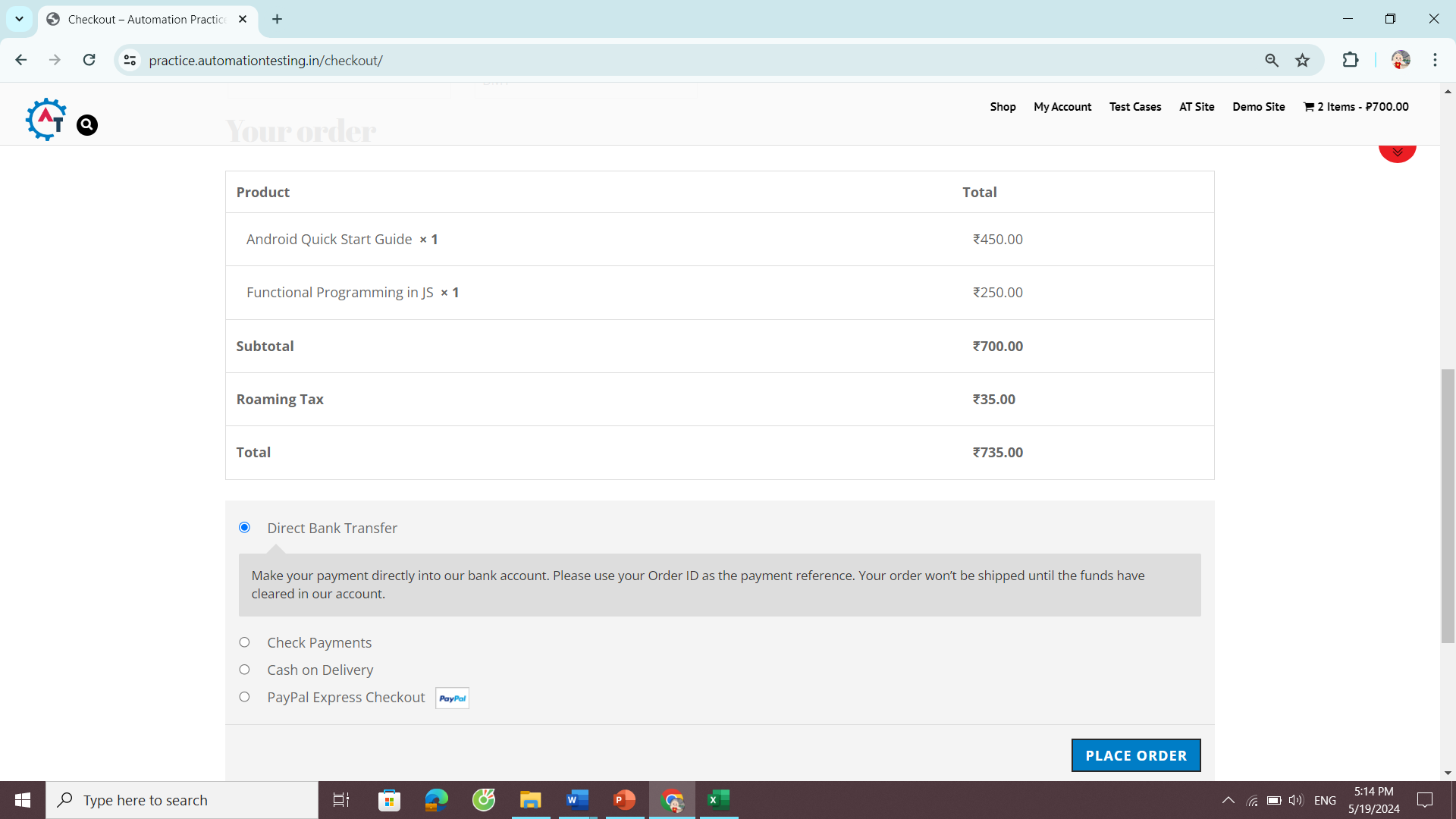


Hình 3.3.5: Giao diện trang Thêm địa chỉ giao hàng

### **Test Case chức năng Đặt hàng mua sắm**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test case Description** | **Actual Output** |
| **TC\_01** | Successful Shopping Order with the correct full enter | Successful Add, display the message "Thank you. Your order has been received." |
| **TC\_02** | Successful Shopping Order with leave the item blank optional | Successful Add, display the message "Thank you. Your order has been received." |
| **TC\_03** | Shopping Order failed with invalid Postcode/ZIP | Add must fail, Display the message "Please enter a valid postcode/ZIP." |
| **TC\_04** | Shopping Order failed with Empty First Name, Last Name | Add must fail, Display the message "Phone is a required field. Town / City is a required field." |
| **TC\_05** | Shopping Order failed with invalid email | Add must fail, Display the message "Email Address is not a valid email address." |
| **TC\_06** | Shopping Order failed with invalid Phone | Add must fail, Display the message "Phone is not a valid phone number. " |
| **TC\_07** | Shopping Order failed with Empty Address, City | Add must fail, Display the message "Billing Address is a required field. Billing Town / City is a required field." |

Bảng 3.3.6: Test Case chức năng Đặt hàng mua sắm



Hình 3.3.6: Giao diện trang Đặt hàng mua sắm

# Thực hiện và chạy tự động kiểm thử Website

## Kiểm thử tự động

### **Kiểm thử tự động là gì?**

Kiểm thử tự động là quá trình thực hiện một cách tự động các bước trong một kịch bản kiểm thử. Kiểm thử tự động bằng một công cụ nhằm rút ngắn thời gian kiểm thử. Kiểm thử tự động là một kỹ thuật tự động hóa, trong đó người kiểm thử tự viết các tập lệnh và sử dụng phần mềm phù hợp để kiểm thử phần mềm. Về cơ bản nó là một quá trình tự động hóa của một quy trình thủ công. Giống như kiểm thử hồi quy, kiểm thử tự động cũng được sử dụng để kiểm tra ứng dụng theo quan điểm load, performance và stress test.

Mục tiêu chính của kiểm thử tự động là tăng hiệu quả kiểm thử và phát triển giá trị phần mềm.

### **Quy trình kiểm thử tự động**

|  |  |  |
| --- | --- | --- |
| STT | Bước thực hiện | Mô tả |
| 1 | Lên kế hoạch | Xác định mục tiêu của việc kiểm thử: công cụ kiểm thử, chắc năng cần được kiểm thử |
| 2 | Kiểm thử thủ công | Thực hiện kiểm thử bằng tay để kiểm tra tính năng, giao diện và các trường hợp sử dụng thường gặp. |
| 3 | Thiết kế trường hợp kiểm thử | Viết chi tiết về các bước kiểm thử, dữ liệu đầu vào và kết quả mong đợi. |
| 4 | Tạo kịch bản kiểm thử | Tạo các kịch bản kiểm thử tự động (test scripts) dựa trên trường hợp kiểm thử đã viết |
| 5 | Chạy kịch bản kiểm thử | thực hiện kiểm thử tự động bằng cách chạy các kịch bản kiểm thử. |
| 6 | Phân tích, đánh giá kết quả | Đánh giá kết quả sau khi chạy kịch bản kiểm thử |

Bảng 4.1.2: Quy trình kiểm thử tự động

## Thực hiện kiểm thử giao diện tự động

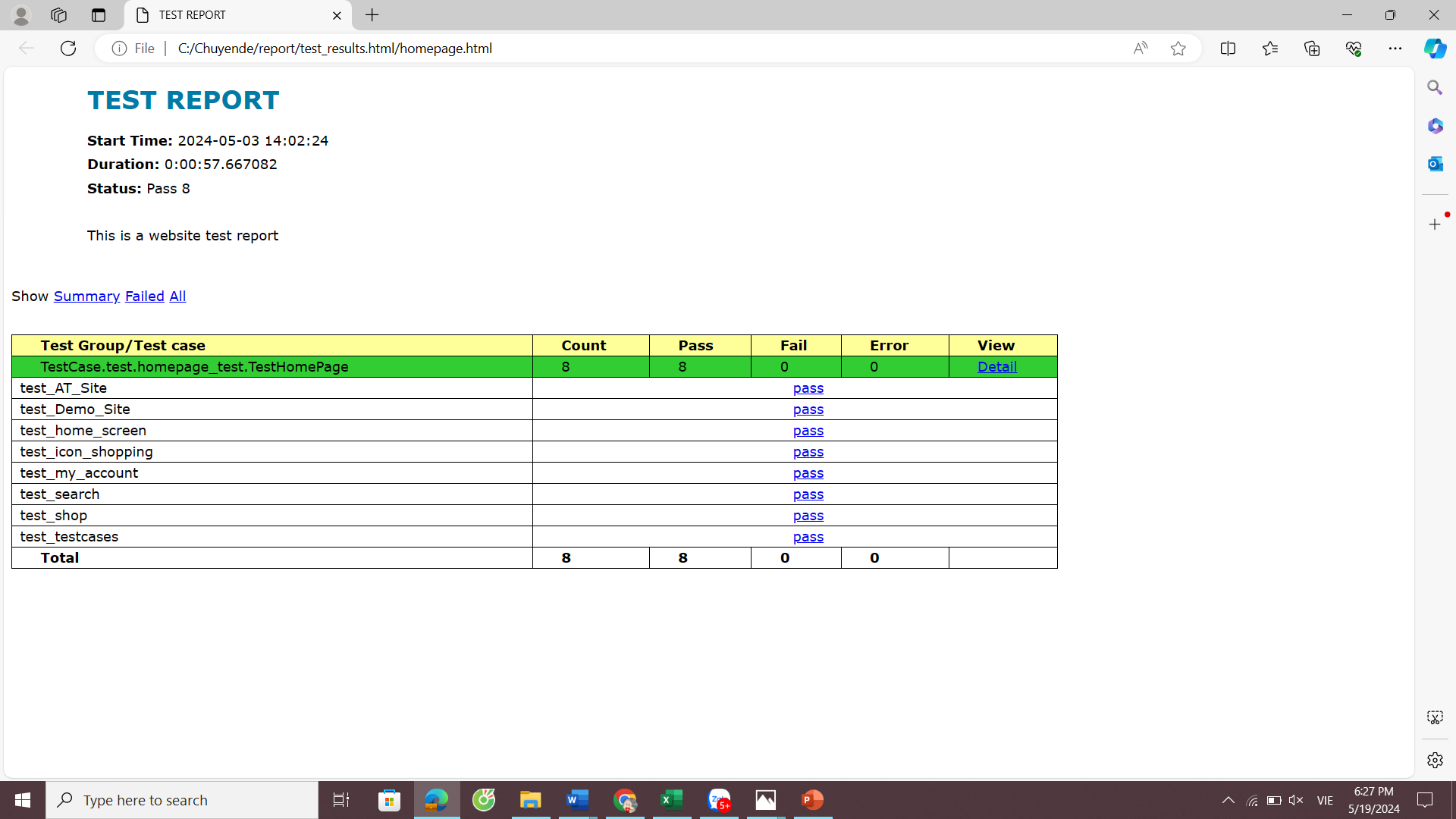
### **Thực hiện kiểm thử giao diện Trang chủ**

Điều kiện tiên quyết/ dữ liệu: mở trình duyệt, vào liên kết trang website

Các bước thực hiện:

|  |  |  |
| --- | --- | --- |
| **ID** | **Description** | **Step** |
|  |
|  |
| **TC\_01** | Home screen | 1) Mở trình duyệt  2) URL "http://practice.automationtesting.in/"  3) Cuộn xuống cuối trang |  |
| **TC\_02** | Button Search | 1) Mở trình duyệt  2) URL "http://practice.automationtesting.in/"  3) Nhấp vào Seach Menu  4) Nhập |  |
|  |
| **TC\_03** | Button Shop | 1) Mở trình duyệt  2)URL "http://practice.automationtesting.in/"  3) Nhấp vào Menu cửa hàng |  |
|  |
| **TC\_04** | Button My Account | 1) Mở trình duyệt  2) URL "http://practice.automationtesting.in/"  3) Nhấp vào Menu Tài khoản của tôi |  |
|  |
| **TC\_05** | Button Test Cases | 1) Mở trình duyệt  2) URL "http://practice.automationtesting.in/"  3) Nhấp vào menu Test Cases |  |
|  |
| **TC\_06** | Button AT Site | 1) Mở trình duyệt  2) URL "http://practice.automationtesting.in/"  3) Nhấp vào AT Site Menu |  |
|  |
| **TC\_07** | Button Demo Site | 1) Mở trình duyệt  2) URL "http://practice.automationtesting.in/"  3) Nhấp vào Menu trang web demo |  |
|  |
| **TC\_08** | Button Shopping | 1) Mở trình duyệt  2) URL "http://practice.automationtesting.in/"  3) Nhấp vào Menu mua sắm |  |
|  |

Bảng 4.2.1: Các bước thực hiện giao diện Trang chủ



Hình 4.2.1: Kết quả giao diện Trang chủ

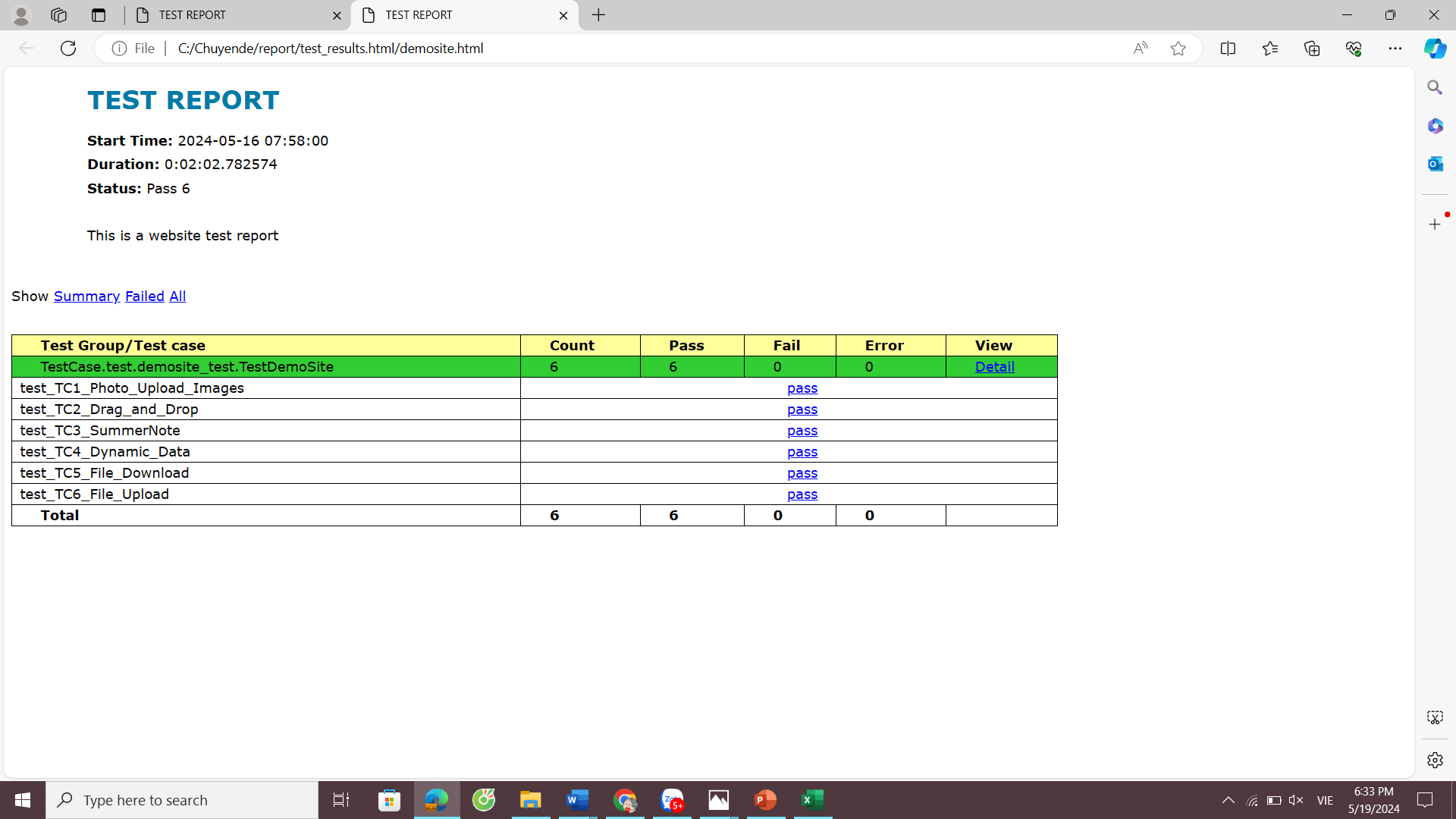
### **Thực hiện kiểm thử giao diện Trang Demo Site**

Điều kiện tiên quyết/ dữ liệu: mở trình duyệt, vào liên kết trang website

Các bước thực hiện:

|  |  |  |
| --- | --- | --- |
| **ID** | **Description** | **Step** |
|  |
|  |
| **TC\_01** | Button Register with Photo Upload Images | 1) Đến URL http://practice.automationtesting.in/  2) Nhấp vào nút Trang web Demo  3) Nhập Đăng ký trong hộp văn bản  4) Nhập ảnh với Chọn tập tin |  |
| **TC\_02** | Button Interactions with Drag and Drop | 1) Đến URL http://practice.automationtesting.in/  2) Nhấp vào nút Trang web Demo  3) Nhấp vào nút tương tác  4) Nhấp chuột Kéo và thả  5) Nhấp Static |  |
| **TC\_03** | Button WYSIWYG with SummerNote | 1) Đến URL http://practice.automationtesting.in/  2) Nhấp vào nút Trang web Demo  3) Nhấp vào nút WYSIWYG  4) Nhấp vào SummerNote |  |
| **TC04** | Button More with Dynamic Data | 1) Đến URL http://practice.automationtesting.in/ 2) Nhấp Demo Site button 3) Nhấp Button More 4) Nhấp Dynamic Data |  |
| **TC\_05** | Button More with File Download | 1) Đến URL http://practice.automationtesting.in/ 2) Nhấp Demo Site button 3) Nhấp Button More 4) Nhấp File Download |  |
| **TC 06** | Button More with File Upload | 1) Đến URL http://practice.automationtesting.in/ 2) Nhấp Demo Site button 3) Nhấp Button More 4) Nhấp File Upload |  |

Bảng 4.2.2: Các bước thực hiện giao diện Trang Demo Site



Hình 4.2.2: Kết quả giao diện Trang Demo Site

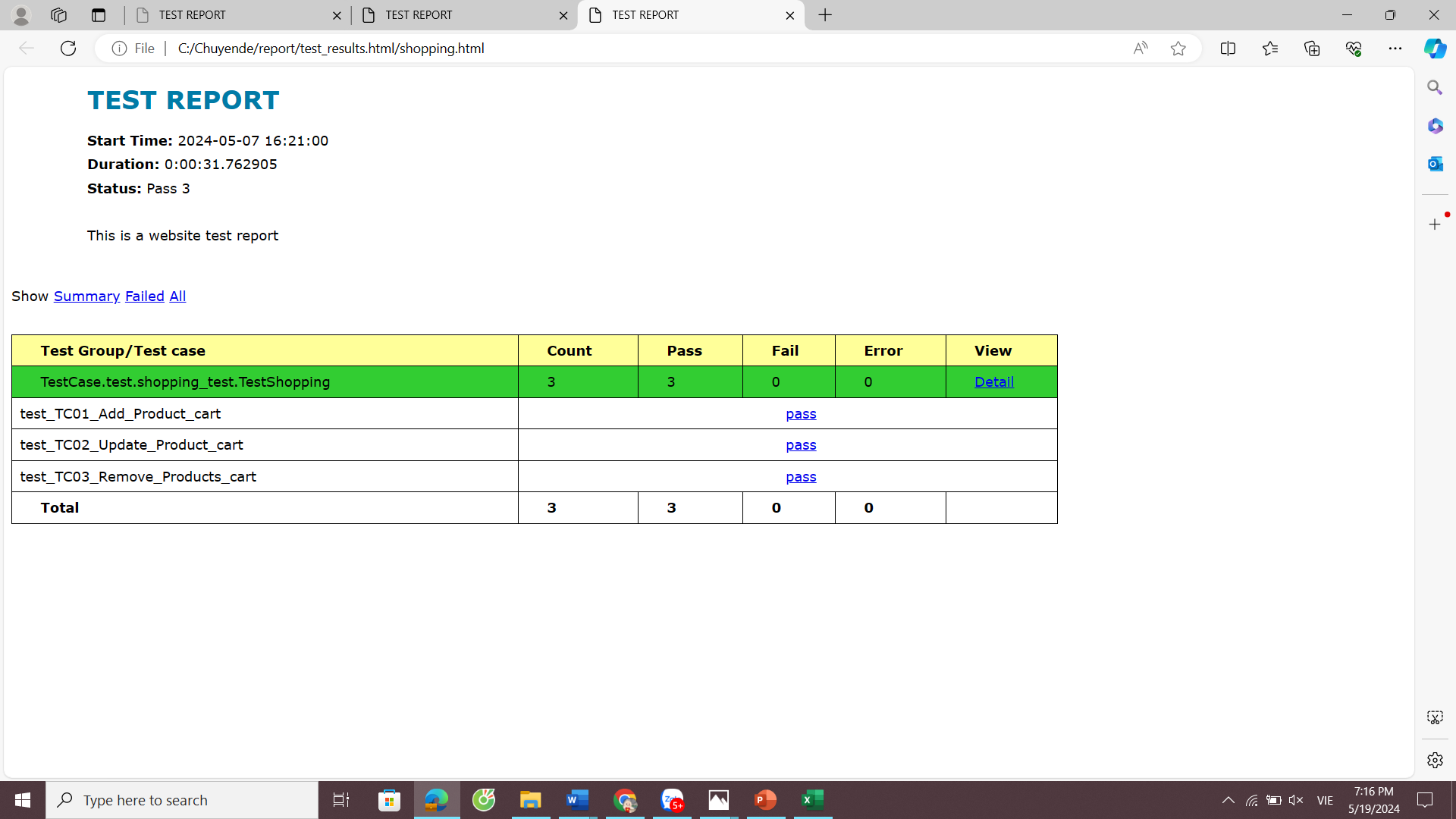
### **Thực hiện kiểm thử giao diện Thêm sản phẩm vào giỏ hàng**

Điều kiện tiên quyết/ dữ liệu: Đăng nhập vào tài khoản người dùng cá nhân. Sản phẩm được thêm vào giỏ hàng

Các bước thực hiện:

|  |  |  |
| --- | --- | --- |
| **ID** | **Description** | **Step** |
| **TC\_01** | Thêm sản phẩm vào giỏ hàng | 1) Đến URL http://practice.automationtesting.in/  2) Nhấp vào nút Cửa hàng  3) Nhấp vào sách bạn muốn thêm  4) Nhấp vào Thêm vào giỏ hàng  5) Nhấp vào Xem giỏ hàng |
| **TC\_02** | Cập nhật sản phẩm trong giỏ hàng | 1) Đến URL http://practice.automationtesting.in/  2) Nhấp vào nút Cửa hàng  3) Nhấp vào Thực hiện cập nhật sản phẩm |
| **TC\_03** | Xóa sản phẩm trong giỏ hàng | 1) Đến URL http://practice.automationtesting.in/  2) Nhấp vào nút Cửa hàng  3) Nhấp vào nút X để xóa sản phẩm |

Bảng 4.2.3: Các bước thực hiện giao diện Thêm sản phẩm vào giỏ hàng



Hình 4.2.3: Kết quả giao diện Thêm sản phẩm vào giỏ hàng

## Thực hiện kiểm thử chức năng tự động

### **Thực hiện kiểm thử chức năng Đăng ký**

Điều kiện tiên quyết/ dữ liệu: mở trình duyệt

Các bước thực hiện:

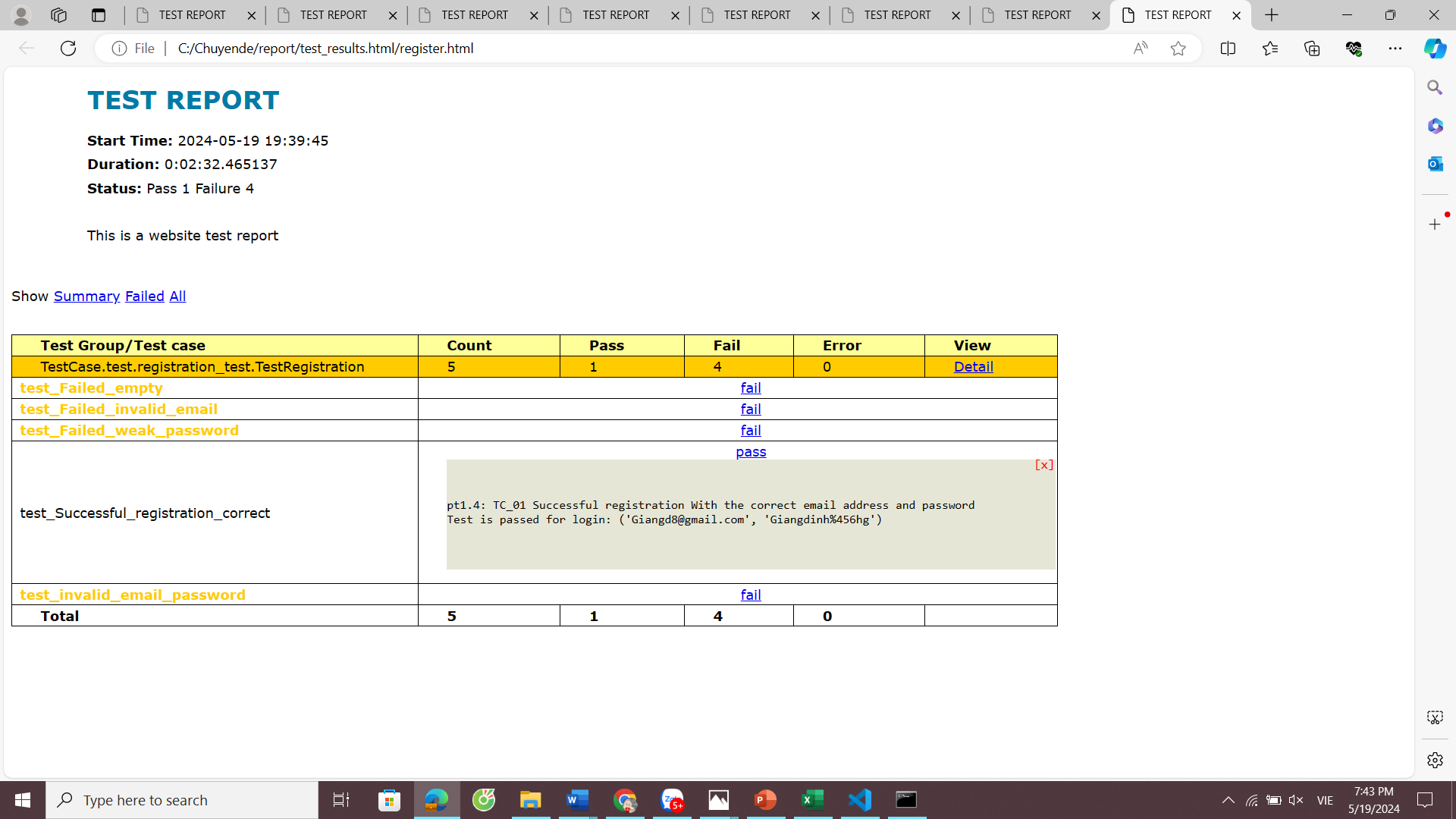
1) Nhập URL "http://practice.automationtesting.in/"

2) Nhấp vào menu tài khoản của tôi

3) Nhập Địa chỉ Email đã đăng ký vào hộp văn bản

4) Nhập mật khẩu vào hộp văn bản

5) Nhấp vào nút Đăng ký



Hình 4.3.1: Kết quả chức năng Đăng ký

### **Thực hiện kiểm thử chức năng Đăng nhập**

Điều kiện tiên quyết/ dữ liệu: mở trình duyệt, đã có tài khoản người dùng

Các bước thực hiện:

1) Nhập URL "http://practice.automationtesting.in/"

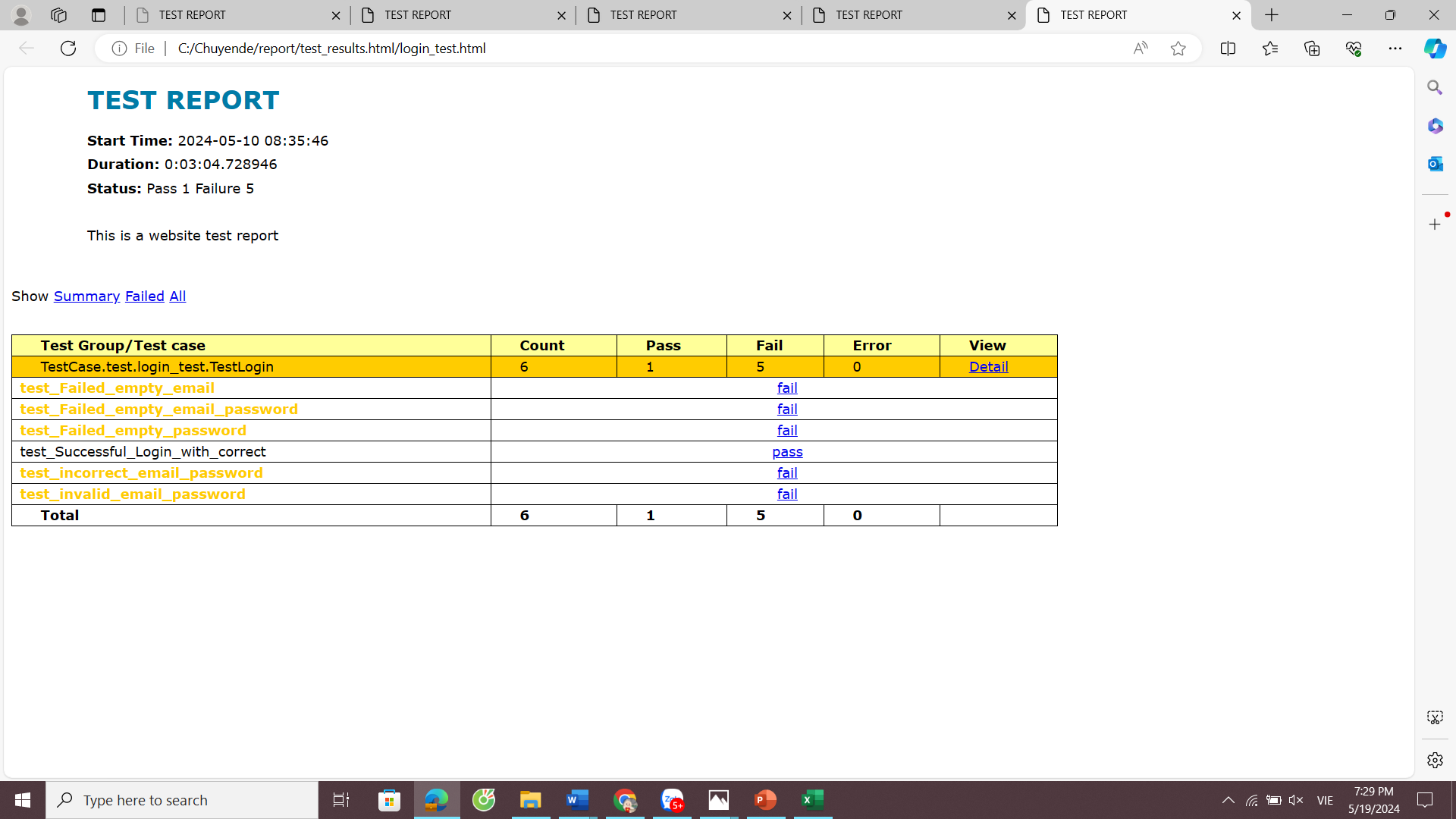
2) Nhấp vào menu tài khoản của tôi

3) Nhập Tên người dùng hoặc địa chỉ email vào hộp văn bản

4) Nhập mật khẩu vào hộp văn bản

5) Chọn hộp cho Ghi nhớ tôi

6) Nhấp vào nút đăng nhập



Hình 4.3.2: Kết quả chức năng Đăng nhập

### **Thực hiện kiểm thử chức năng Đăng xuất**

Điều kiện tiên quyết/ dữ liệu: đăng nhập vào tài khoản người dùng cá nhân

Các bước thực hiện: 1. Nhấp vào nút Đăng xuất, văn bản màu đỏ ở góc trái của trang



Hình 4.3.3: Kết quả chức năng Đăng xuất

### **Thực hiện kiểm thử chức năng Cập nhật tài khoản**

Điều kiện tiên quyết/ dữ liệu: đăng nhập vào tài khoản người dùng cá nhân

Các bước thực hiện:

1) Đăng nhập vào tài khoản người dùng cá nhân

http://practice.automationtesting.in/

2) Nhấp vào Chi tiết tài khoản

3) Nhập Tên và Họ

4) Nhập địa chỉ Email

5) Nhập mật khẩu hiện tại

6) Nhấp vào Mật khẩu mới, Xác nhận mật khẩu mới

7) Nhấp vào LƯU THAY ĐỔI

### **Thực hiện kiểm thử chức năng Thêm địa chỉ giao hàng**

Điều kiện tiên quyết/ dữ liệu: đăng nhập vào tài khoản người dùng cá nhân

Các bước thực hiện:

1) Đăng nhập vào tài khoản người dùng cá nhân

http://practice.automationtesting.in/

2) Nhấp vào Địa chỉ

3) Nhấp vào Địa chỉ thanh toán, nút chỉnh sửa

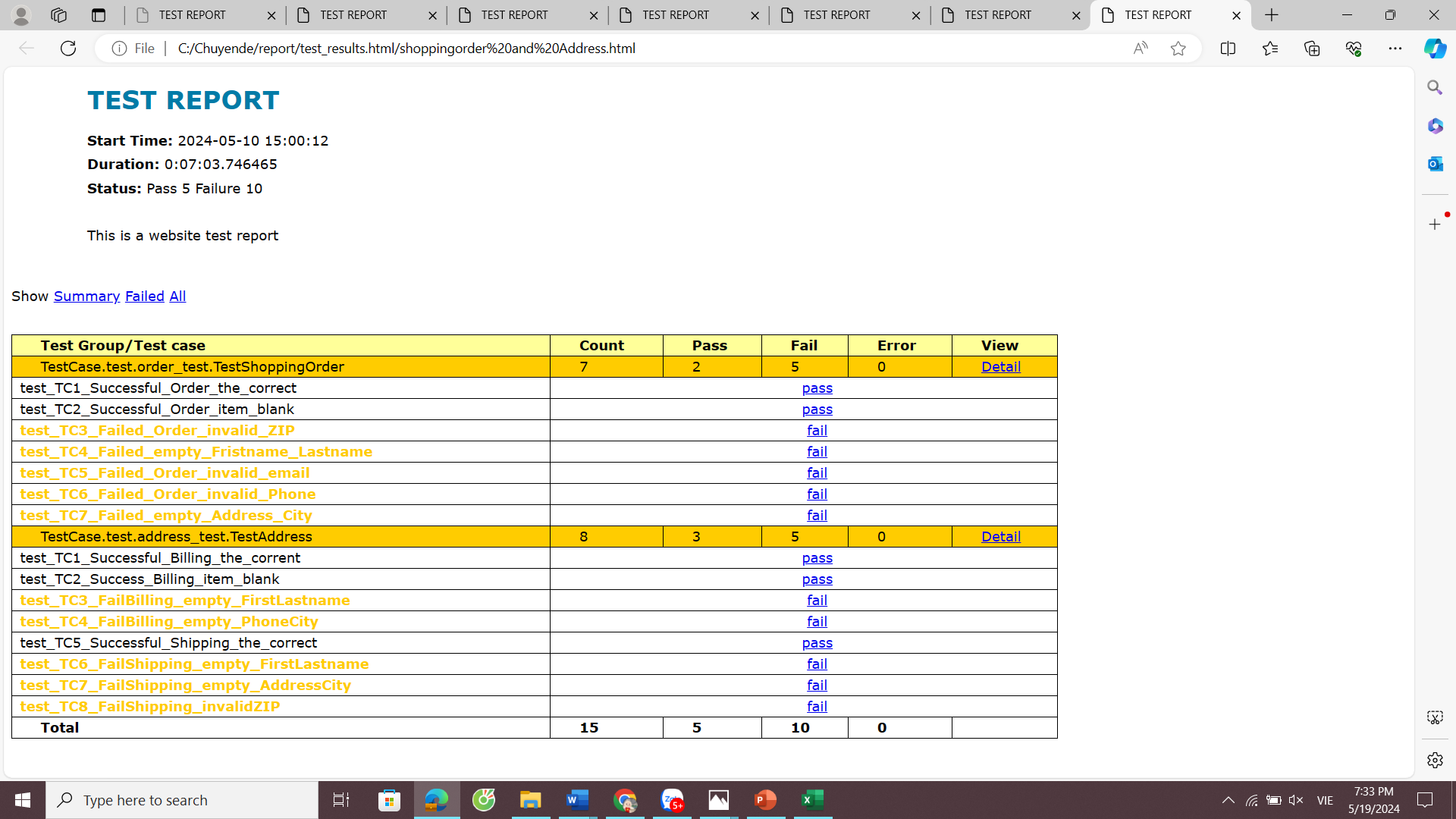
4) Nhập tên, họ

5) Nhập địa chỉ email, điện thoại

6) Nhập Quốc gia, Địa chỉ, Thị trấn / Thành phố

7) Nhập các mục tùy chọn

8) Nhấp vào LƯU ĐỊA CHỈ



Hình 4.3.5: Kết quả chức năng Đặt hàng mua sắm và Thêm địa chỉ

### **Thực hiện kiểm thử chức năng Đặt hàng mua sắm**

Điều kiện tiên quyết/ dữ liệu: đăng nhập vào tài khoản người dùng cá nhân

Các bước thực hiện:

1) Đăng nhập vào tài khoản người dùng cá nhân

http://practice.automationtesting.in/

2) Thanh toán giỏ hàng

3) Nhấp vào giỏ hàng

4) Nhập thông tin đầy đủ

5) Nhập Chọn phương thức thanh toán

6) Nhấp vào Đặt hàng

# PHẦN KẾT LUẬN

## 1. Kết quả đạt được

Sau một thời gian nghiên cứu và hoàn thiện đề tài, em đã có đủ kiến thức cũng như kỹ năng để thực hiện kiểm thử Website

* Về lý thuyết và kỹ năng:

Nắm được kiến thức về các công nghệ mới và phổ biến hiện nay: Python, Selenium, Unittest ...

Rèn luyện được kỹ năng làm việc độc lập, làm việc nhóm và giải quyết vấn đề

Tạo được tác phong riêng về làm việc cũng như cách ứng sử.

* Về kiểm thử:

Bao quát hết trường hợp kiểm thử các chức năng trang Website

Thực hiện chạy kiểm thử tự động đúng với mục tiêu đề tài đề ra

## 2. Ưu điểm

Kiểm thử Website có những ưu điểm sau:

* Thực hiện hầu hết các chức năng
* Kiểm tra trực tiếp trên Website: kiểm soát lỗi, tránh sai sót
* Vì tính tự động nên thực hiện lại một cách dễ dàng hơn

## 3. Nhược điểm

Kiểm thử cũng có một số nhược điểm sau:

* Chi phí thực hiện cao
* Vẫn chưa thực hết tất cả các chức năng

## 4. Hướng phát triển cho tương lai

* Mở rộng phạm vi kiểm thử: Đầu tiên, mở rộng phạm vi của các trường hợp kiểm thử bằng cách tạo ra nhiều kịch bản kiểm thử hơn. Điều này có thể bao gồm kiểm tra tính đúng đắn của các chức năng chính, kiểm tra tích hợp với các dịch vụ khác như API hoặc cơ sở dữ liệu, và kiểm tra hiệu suất của trang web.
* Tối ưu hóa và mở rộng tự động hóa: Cải thiện tự động hóa bằng cách sử dụng các kỹ thuật tối ưu hóa như parallelization để chạy các kịch bản kiểm thử đồng thời. Tạo các trường hợp kiểm thử đa nền tảng, đảm bảo tính nhất quán trên nhiều trình duyệt và thiết bị.
* Tích hợp với công cụ và hệ thống khác: Tích hợp tự động hóa kiểm thử vào các quy trình liên tục như CI/CD để đảm bảo rằng mọi thay đổi trong mã nguồn đều được kiểm tra tự động. Bạn cũng có thể tích hợp với các công cụ quản lý dự án như Jira để quản lý và theo dõi các vấn đề kiểm thử.
* Xây dựng báo cáo và phân tích kết quả: Phát triển các công cụ và quy trình để tự động tạo báo cáo sau mỗi lần chạy kiểm thử. Báo cáo này nên cung cấp thông tin chi tiết về kết quả kiểm thử, bao gồm cả các lỗi phát hiện và thời gian chạy. Bạn cũng có thể phân tích kết quả để xác định các xu hướng và vấn đề phổ biến.

# Tài liệu tham khảo

Tiếng Anh:

[1]. Kalilur Rahman (2019). Science of Selenium: Master Web UI Automation and Create Your Own Test Automation Framework.

[2]. Sujay Raghavendra (2020). Python Testing with Selenium: Learn to Implement Different Testing Techniques Using the Selenium WebDriver.

[3]. Unmesh Gundecha (2014). Learning Selenium Testing Tools with Python.

[4]. Selenium Webdriver:

<https://www.selenium.dev/documentation/webdriver/>

[5]. Selenium Python Tutorial For Automated Testing

<https://saucelabs.com/resources/blog/selenium-with-python-for-automated-testing>

Tiếng Việt:

[1]. Hướng dẫn cơ bản Selenium trong Python:

<https://co-well.vn/nhat-ky-cong-nghe/test-automation-huong-dan-co-ban-selenium-trong-python/>

# Phụ lục

Link github testing:

* Automation testing: <https://github.com/Gianghdn/test>

Link demo testing:

* <https://www.youtube.com/watch?v=jJph2f4gEcY>

Link Website thực hiện kiểm thử:

* <https://practice.automationtesting.in/>

# Ý KIẾN CỦA NGƯỜI HƯỚNG DẪN

***Nhận xét*** :

**Đối với khóa luận tốt nghiệp** (Đánh dấu  và ký tên vào ý kiến chọn lựa sau):

|  |  |
| --- | --- |
|  | Ký tên |
| Đồng ý thông qua báo cáo |  |
| Không đồng ý thông qua báo cáo |  |

|  |  |
| --- | --- |
|  | Đăk Lắk, *ngày ..... tháng ... năm 2024*  **NGƯỜI HƯỚNG DẪN**  (*Ký và ghi rõ họ tên*) |

Họ và tên sinh viên: Đinh Ngọc Hương Giang

Cán bộ hướng dẫn (ghi rõ học hàm, học vị): Thạc sĩ Trương Hải

Tên chuyên đề tốt nghiệp: “Xây dựng các trường hợp kiểm thử tự động cho Website sử dụng Selenium với Python”

Điểm:

*Đăk Lăk, ngày ..... tháng … năm 2024*

**SINH VIÊN**

*(Ký và ghi rõ họ tên)*

Đinh Ngọc Hương Giang

**XÁC NHẬN**

**CÁN BỘ HƯỚNG DẪN CHỦ TỊCH HỘI ĐỒNG**

*(Ký và ghi rõ họ tên) (Ký và ghi rõ họ tên)*