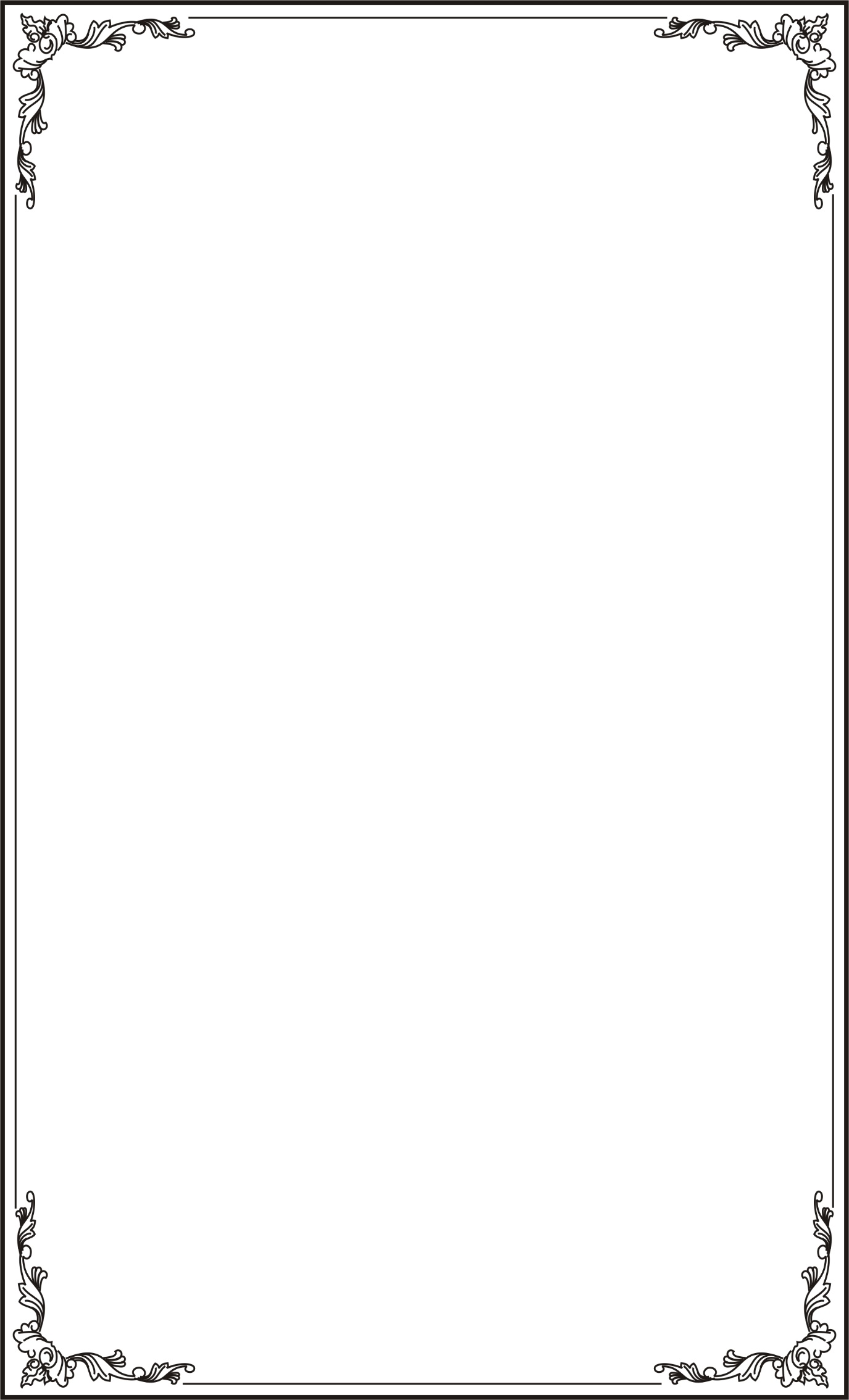
** BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

Trần Thị Thư MSSV: N18DCCN224 Tên Đề tài: Xây dựng công cụ kiểm thử chức năng cho một website 2021-2022 TP HCM 2022

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**---------------\*\*\*---------------**



**BÁO CÁO THỰC TẬP TỐT NGHIỆP ĐẠI HỌC**

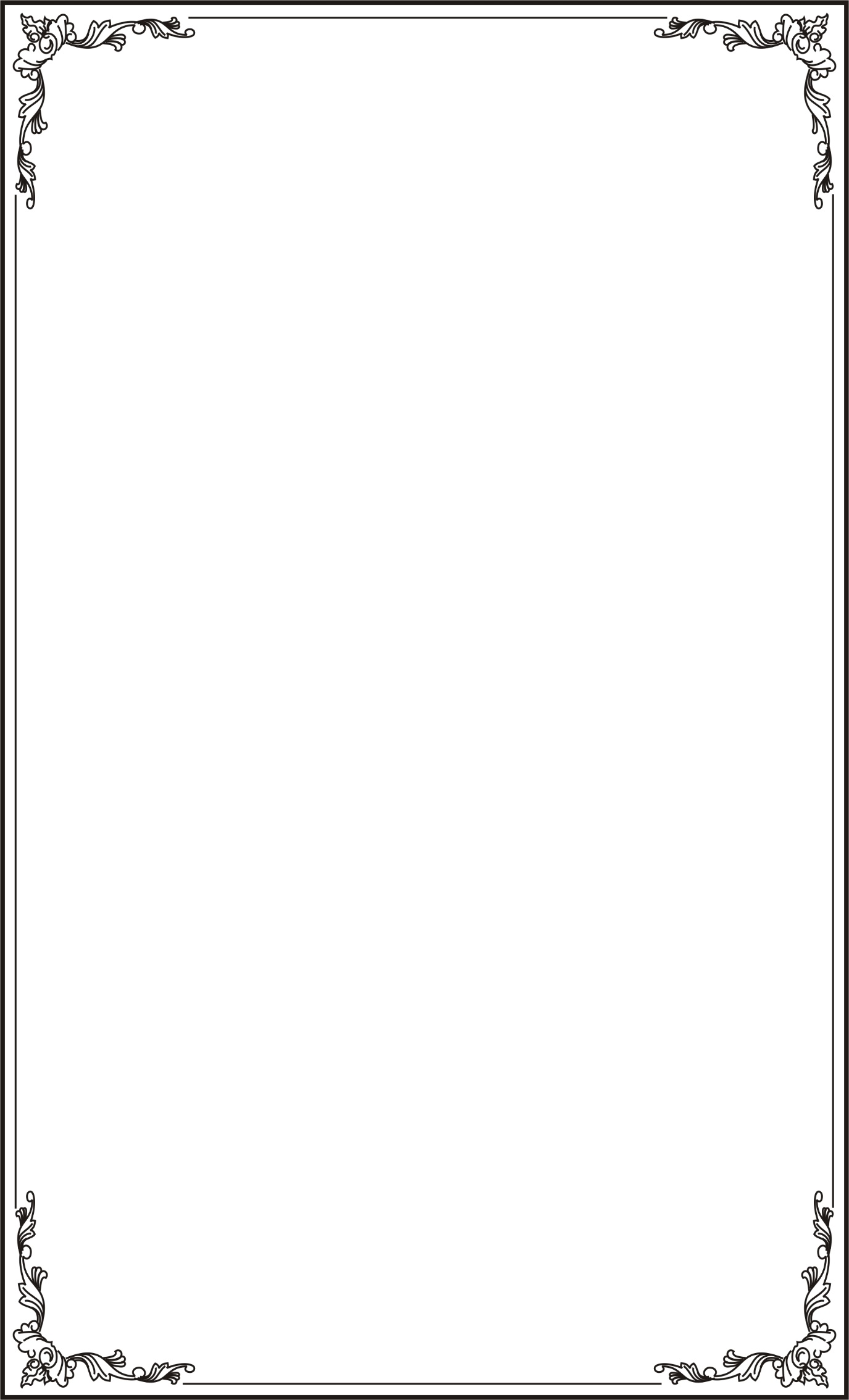
***Đề tài:***

**XÂY DỰNG CÔNG CỤ KIỂM THỬ CHỨC**

**NĂNG CHO MỘT WEBSITE**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn : | ThS. NGUYỄN ANH HÀO |
| Sinh viên thực hiện : | **TRẦN THỊ THƯ** |
| Mã số sinh viên : | **N18DCCN224** |
| Lớp : | **D18CQCP02-N** |
| Khoá : | **2018-2023** |
| Hệ : | **ĐẠI HỌC CHÍNH QUY** |

**TP.HCM 08/2022**

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**---------------\*\*\*---------------**



**BÁO CÁO THỰC TẬP TỐT NGHIỆP ĐẠI HỌC**

***Đề tài:***

**XÂY DỰNG CÔNG CỤ KIỂM THỬ CHỨC**

**NĂNG CHO MỘT WEBSITE**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn : | ThS. NGUYỄN ANH HÀO |
| Sinh viên thực hiện : | **TRẦN THỊ THƯ** |
| Mã số sinh viên : | **N18DCCN224** |
| Lớp : | **D18CQCP02-N** |
| Khoá : | **2018-2023** |
| Hệ : | **ĐẠI HỌC CHÍNH QUY** |

**TP.HCM 08/2022**

**LỜI CẢM ƠN**

Lời đầu tiên, em xin gửi lời tri ân sâu sắc đến các thầy cô trường Học Viện Công Nghệ Bưu Chính Viễn thông cơ sở tại TP.HCM đã tận tình dẫn dắt và truyền đạt cho em rất nhiều kiến thức quý báu trong các học kỳ vừa qua.

Đặc biệt, em xin gửi lời cảm ơn chân thành đến ThS.Lưu Nguyễn Anh Hào. Thầy đã hướng dẫn tận tình, truyền đạt kiến thức, chỉ bảo cho em trong suốt thời gian học tập và thực hiện đề tài. Kính chúc thầy và gia đình nhiều sức khoẻ và thành công trong cuộc sống.

Con xin gửi lòng biết ơn và sự kính trọng đến cha mẹ, gia đình đã luôn động viên, tạo điều kiện tốt nhất cho con học tập.

Cảm ơn công ty TNHH Enrich Management System đã tạo điều kiện cho em thực tập, truyền đạt kiến thức và giúp em làm quen với môi trường chuyên nghiệp hết sức tận tình chu đáo.

Và xin chân thành cảm ơn tất cả các bạn đã luôn sát cánh, giúp đỡ, động viên mình trong những thời điểm khó khăn, tiếp thêm động lực và ý chí giúp vượt qua khó khăn trong suốt quá trình học tập tại trường.

Tuy nhiên, vì thời gian và kiến thức chuyên môn còn hạn chế cùng nhiều yếu tố khách quan khác nên trong quá trình thực hiện đề tài không tránh khỏi những sai sót. Kính mong được lời nhận xét và góp ý của thầy cô, để em có thể hoàn thiện bản thân mình hơn.

Một lần nữa, em xin chân thành cảm ơn!

Hồ Chí Minh, tháng 08 năm 2022

Sinh viên thực hiện

Trần Thị Thư

**MỤC LỤC**

[**CHƯƠNG 1: TÌM HIỂU VỀ KIỂM THỬ PHẦN MỀM** 1](#_Toc112511831)

[**1.1** **Khái niệm kiểm thử phần mềm** 1](#_Toc112511832)

[**1.2** **Các loại kiểm thử phần mềm** 1](#_Toc112511833)

[**1.2.1 Kiểm thử chức năng:** 1](#_Toc112511834)

[**1.2.1.a Kiểm thử liên kết** 1](#_Toc112511835)

[**1.2.2** **Kiểm thử khả năng sử dụng** 2](#_Toc112511836)

[**1.2.3 Kiểm thử sự tương thích** 3](#_Toc112511837)

[**1.2.4 Kiểm thử cơ sở dữ liệu** 3](#_Toc112511838)

[**1.2.5 Kiểm thử giao diện** 4](#_Toc112511839)

[**1.2.6 Kiểm thử hiệu năng** 4](#_Toc112511840)

[**1.2.7 Kiểm thử bảo mật** 4](#_Toc112511841)

[**1.3 Kỹ thuật kiểm thử hộp đen** 5](#_Toc112511842)

[**1.3.1 Khái niệm** 5](#_Toc112511843)

[**1.3.2** **Quy trình làm việc của kiểm thử hộp đen** 5](#_Toc112511844)

[**1.3.3 Các ưu điểm và nhược điểm của kiểm thử hộp đen** 6](#_Toc112511845)

[**1.3.4 Các kỹ thuật kiểm thử hộp đen** 6](#_Toc112511846)

[**1.4 Tạo trường hợp kiểm thử (Test case)** 12](#_Toc112511847)

[**1.4.1 Khái niệm** 12](#_Toc112511848)

[**1.4.2 Cấu trúc của Test case** 12](#_Toc112511849)

[**1.4.3 Các bước xác định test case** 12](#_Toc112511850)

[**1.4.4 Xác định các trường hợp kiểm thử** 13](#_Toc112511851)

[**CHƯƠNG 2: GIỚI THIỆU VỀ KIỂM THỬ TỰ ĐỘNG** 14](#_Toc112511852)

[**2.1 Khái niệm:** 14](#_Toc112511853)

[**2.2 Quy trình kiểm thử tự động** 14](#_Toc112511854)

[**2.3 Mục đích của kiểm thử tự động** 14](#_Toc112511855)

[**2.4** **Lý do kiểm thử tự động** 14](#_Toc112511856)

[**1.5** **Rủi ro liên quan đến kiểm thử tự động** 15](#_Toc112511857)

[**CHƯƠNG 3. TÌM HIỂU VỀ SELENIUM WEBDRIVER** 16](#_Toc112511858)

[**3.1 Selenium Webdriver:** 16](#_Toc112511859)

[**3.2 Các tính năng của Selenium Webdriver:** 16](#_Toc112511860)

[**3.3.1** **Các câu lệnh trình duyệt** 16](#_Toc112511861)

[**3.3.2** **Các câu lệnh trong WebElement** 17](#_Toc112511862)

[**3.3.3 Các câu lệnh chuyển hướng trình duyệt** 17](#_Toc112511863)

[**3.3.4 Các câu lệnh wait (trên NodeJs)** 17](#_Toc112511864)

[**CHƯƠNG 4. WEBSITE BÁN HÀNG ONLINE** 20](#_Toc112511865)

[**4.1 Các chức năng chính của website** 20](#_Toc112511866)

[**4.2 Database** 21](#_Toc112511867)

[**4.2.1 Sơ đồ diagram:** 21](#_Toc112511868)

[**4.2.2 Các bảng thành phần** 21](#_Toc112511869)

[**4.3 Giao diện website** 24](#_Toc112511870)

[**4.3.1 Giao diện người dùng** 24](#_Toc112511871)

[**4.3.2 Giao diện admin** 30](#_Toc112511872)

[**CHƯƠNG 5. XÂY DỰNG CÔNG CỤ KIỂM THỬ TỰ ĐỘNG** 35](#_Toc112511873)

[**5.1 Xây dựng các test case để thực thi test script** 35](#_Toc112511874)

[**5.1.1 Đăng ký** 35](#_Toc112511875)

[**5.1.2 Đăng nhập** 36](#_Toc112511876)

[**5.1.3 Tài khoản người dùng** 37](#_Toc112511877)

[**5.1.4 Tìm kiếm** 39](#_Toc112511878)

[**5.1.5 Đặt hàng** 40](#_Toc112511879)

[**5.2 Cài đặt phần mềm** 45](#_Toc112511880)

[**5.3 Thực hành thiết kế** 45](#_Toc112511881)

[**5.3.1 Cấu trúc của project để định nghĩa các test scripts** 45](#_Toc112511882)

[**5.3.2 Cấu trúc của test scrips** 46](#_Toc112511883)

[**5.3.3 Ghi log cho từng test case vào file** 48](#_Toc112511884)

[**5.3.4 Ghi vết kết quả mỗi lần chạy test case nếu fail** 48](#_Toc112511885)

[**5.3.5 Báo cáo kết quả thực thi test case** 48](#_Toc112511886)

[**CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA PHẦN MỀM** 50](#_Toc112511887)

[**6.1 Kết luận** 50](#_Toc112511888)

[**6.1.1 Kết quả đạt được** 50](#_Toc112511889)

[**6.1.2 Hạn chế** 50](#_Toc112511890)

[**6.2 Hướng phát triển** 50](#_Toc112511891)

**DANH MỤC HÌNH ẢNH**

[Hình 1 Kiểm thử hộp đen 5](#_Toc112705614)

[Hình 2 Trạng thái chuyển đổi đăng nhập 9](#_Toc112705615)

[Hình 3 Sơ đồ diagram của database bán hàng. 21](#_Toc112705616)

[Hình 4 Màn hình Trang chủ 24](#_Toc112705617)

[Hình 5 Màn hình Đăng ký 24](#_Toc112705618)

[Hình 6 Màn hình Đăng ký khi bỏ trống thông tin 25](#_Toc112705619)

[Hình 7 Màn hình Đăng ký khi nhập thông tin không hợp lệ 25](#_Toc112705620)

[Hình 8 Màn hình Đăng ký thành công 26](#_Toc112705621)

[Hình 9 Màn hình đăng nhập 26](#_Toc112705622)

[Hình 10 Màn hình Đăng nhập khi bỏ trống tên đăng nhập 27](#_Toc112705623)

[Hình 11 Màn hình Đăng nhập khi bỏ trống mật khẩu 27](#_Toc112705624)

[Hình 12 Màn hình Đăng nhập khi nhập sai thông tin đăng nhập 28](#_Toc112705625)

[Hình 13 Màn hình thông tin tài khoản 28](#_Toc112705626)

[Hình 14 Màn hình giỏ hàng 29](#_Toc112705627)

[Hình 15 Màn hình Đặt hàng với giỏ hàng trống 29](#_Toc112705628)

[Hình 16 Màn hình Đặt hàng nếu bỏ trống số điện thoại và địa chỉ 30](#_Toc112705629)

[Hình 17 Màn hình Lịch sử mua hàng 30](#_Toc112705630)

[Hình 18 Màn hình Dashboard 31](#_Toc112705631)

[Hình 19 Màn hình Quản lý danh mục sản phẩm 31](#_Toc112705632)

[Hình 20 Màn hình thêm danh mục sản phẩm 32](#_Toc112705633)

[Hình 21 Màn hình Quản lý thương hiệu 32](#_Toc112705634)

[Hình 22 Màn hình thêm thương hiệu 33](#_Toc112705635)

[Hình 23 Màn hình Quản lý sản phẩm 33](#_Toc112705636)

[Hình 24 Màn hình Thêm sản phẩm 34](#_Toc112705637)

[Hình 25 Cấu trúc của project 46](#_Toc112705638)

[Hình 26 Cấu trúc của test script 47](#_Toc112705639)

[Hình 27 Màn hình file ghi log 48](#_Toc112705640)

[Hình 28 Hàm chụp ảnh màn hình nếu test case fail 48](#_Toc112705641)

[Hình 29 Màn hình báo cáo sau khi thực thi test case 49](#_Toc112705642)

[Hình 30 Màn hình báo cáo khi test case fail 49](#_Toc112705643)

**DANH MỤC CÁC BẢNG**

[Bảng 1 Bảng quyết định cho chức năng đăng nhập 7](#_Toc112629532)

[Bảng 2 Bảng chuyển trạng thái của lần đăng nhập đầu tiên không hợp lệ 9](#_Toc112629533)

[Bảng 3 Chuyển trạng thái nếu lần thử thứ ba là hợp lệ 10](#_Toc112629534)

[Bảng 4 Mô tả các giá trị đầu vào 11](#_Toc112629535)

[Bảng 5 Danh mục sản phẩm 21](#_Toc112629536)

[Bảng 6 Thương hiệu sản phẩm 21](#_Toc112629537)

[Bảng 7 Sản phẩm 22](#_Toc112629538)

[Bảng 8 Chi tiết sản phẩm 22](#_Toc112629539)

[Bảng 9 Đặt hàng 22](#_Toc112629540)

[Bảng 10 Chi tiết đặt hàng 23](#_Toc112629541)

[Bảng 11 Tài khoản 23](#_Toc112629542)

[Bảng 12 Phiên đăng nhập 23](#_Toc112629543)

[Bảng 13 Test case Đăng ký 36](#_Toc112629544)

[Bảng 14 Test case Đăng nhập 37](#_Toc112629545)

[Bảng 15 Test case Tài khoản người dùng 39](#_Toc112629546)

[Bảng 16 Test case Tìm kiếm 40](#_Toc112629547)

[Bảng 17 Test case Đặt hàng 42](#_Toc112629548)

[Bảng 18 Test case Quản lý danh mục sản phẩm 43](#_Toc112629549)

[Bảng 19 Test case Quản lý thương hiệu sản phẩm 44](#_Toc112629550)

[Bảng 20 Test case Quản lý sản phẩm 45](#_Toc112629551)

**MỞ ĐẦU**

Chúng ta đang ở thế kỷ XXI, thế kỷ mà mọi hoạt động diễn ra xung quanh chúng ta đều gắn liền với Internet. Với khả năng kết nối mở như vậy, Internet đã trở thành một mạng lớn nhất trên thế giới, mạng của các mạng, xuất hiện trong mọi lĩnh vực thương mại, chính trị, quân sự, nghiên cứu, giáo dục, văn hoá, xã hội... Cũng từ đó, các dịch vụ trên Internet không ngừng phát triển tạo ra cho nhân loại một thời kỳ mới. Hiện nay, các website đóng vai trò quan trọng và được ứng dụng rộng rãi trong nhiều lĩnh vực. Thông tin trên website rất đa dạng từ những thứ cơ bản nhất đến những thông tin mang tính hàn lâm, khoa học, cao hơn là những thông tin nhạy cảm như dữ liệu cá nhân người dùng, thông tin của tổ chức, doanh nghiệp.

Các ứng dụng hiện nay được thiết kế theo xu hướng “website hoá” có thể hoạt động trên nền tảng web, việc này giúp tiết kiệm thời gian và không cần phải thực hiện các bước cài đặt phần mềm. Trong website sẽ có nhiều chức năng, và sẽ được phát triển lâu dài. Trong quá trình xây dựng website, sẽ không chắc chắn chức năng làm ra đã hoạt động đúng hay có sai sót gì nghiêm trọng hay không. Mỗi lần cập nhật tính năng mới nhiều lúc sẽ phát sinh xung đột với chức năng cũ. Vì vậy cần phải kiểm thử các chức năng đó đã hoạt động đúng yêu cầu nhằm mang lại sự trải nghiệm tốt nhất cho người dùng.

Mục đích của đề tài “Xây dựng công cụ kiểm thử chức năng cho một website” nhằm phục vụ cho việc kiểm tra các chức năng của một website hoạt động đúng với yêu cầu đặt ra một cách nhanh chóng, giúp tiết kiệm thời gian và hiệu quả.

**1.** **Mục tiêu**

- Tìm hiểm về kiểm thử phần mềm.

- Tìm hiểu về kiểm thử tự động.

- Tìm hiểu về Cause-Effect Graph để thiết kế testcase.

- Tìm hiểu về MySQL để xây dựng database cho website bán hàng.

- Tìm hiểu về NodeJs để tiến hành xây dựng website bán hàng và xây dựng công cụ kiểm thử.

- Tìm hiểu về Selenium WebDriver để điều khiển browser.

**2. Phương pháp nghiên cứu**

- Về mặt lý thuyết: Thu thập, khảo sát, phân tích các tài liệu và thông tin có liên quan đến xây dựng website và kiểm thử tự động cho website.

- Về mặt thực nghiệm: Thực hiện xây dựng một website bán hàng đơn giản, xây dựng công cụ kiểm thử website bán hàng đã xây dựng ở trên.

# **CHƯƠNG 1: TÌM HIỂU VỀ KIỂM THỬ PHẦN MỀM**

* 1. **Khái niệm kiểm thử phần mềm**

**Kiểm thử phần mềm** (**Software testing**) là một cuộc kiểm tra được tiến hành để cung cấp cho các bên liên quan thông tin về chất lượng của [sản phẩm](https://vi.wikipedia.org/wiki/S%E1%BA%A3n_ph%E1%BA%A9m) hoặc [dịch vụ](https://vi.wikipedia.org/wiki/D%E1%BB%8Bch_v%E1%BB%A5) được kiểm thử. Kiểm thử có thể cung cấp cho doanh nghiệp một quan điểm, một cách nhìn độc lập về [phần mềm](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m) để từ đó cho phép đánh giá và thấu hiểu được những rủi ro trong quá trình [triển khai](https://vi.wikipedia.org/w/index.php?title=Tri%E1%BB%83n_khai&action=edit&redlink=1) phần mềm.

Bằng cách kiểm thử website, một tổ chức/ doanh nghiệp có thể đảm bảo rằng hệ thống website của mình đang hoạt động đúng và có thể được chấp nhận bởi người dùng thực của website.

Giao diện (UI) và chức năng (functionality) là những điều quan trọng nhất khi kiểm thử website.

* 1. **Các loại kiểm thử phần mềm**

### **1.2.1 Kiểm thử chức năng:**

Trong kiểm thử chức năng chúng ta cần kiểm tra từng thành phần hoạt động có như mong đợi hay không, vì vậy nó còn được gọi là “kiểm thử các thành phần”. Kiểm thử chức năng giúp kiểm tra các chức năng của thành phần ứng dung, về cơ bản là để kiểm tra các chức năng được đề cập trong tài liệu mô tả chức năng cũng như kiểm tra xem ứng dụng phần mềm có đáp ứng được kỳ vọng của người dùng hay không.

Các hoạt động kiểm thử này bao gồm:

### **1.2.1.a Kiểm thử liên kết**

Kiểm tra tất cả các liên kết hỏng trên website và tất cả các liên kết đang hoạt động chính xác, bạn có thể kiểm tra các [*liên kết*](https://mona.media/link-la-gi/) khác nhau trên website.

**1.2.1.b Kiểm thử web form**

Đây là phần thiết yếu của bất kỳ **kiểm thử website** nào, mục đích chính của kiểm thử web form là lấy thông tin từ người sử dụng và lưu trữ vào cơ sở dữ liệu đồng thời tương tác với lượng dữ liệu ấy. Dưới đây là các trường hợp kiểm thử được nhắc tới trong kiểm thử web form:

Điều đầu tiên là kiểm tra tính hợp lệ trên mỗi field của form, dưới đây là hai loại Validation cần được xem xét – “Client side” và “Server side” validations.

Điều đầu tiên là kiểm tra tính hợp lệ trên mỗi field của form, dưới đây là hai loại Validation cần được xem xét – “Client side” và “Server side” validations.

- Kiểm tra các giá trị mặc định.

- Kiểm tra tất cả các field bắt buộc.

- Kiểm tra nếu người dùng không nhập vào một field bắt buộc cần hiển thị một thông báo.

- Thêm và sửa thông tin bằng cách sử dụng form.

- Thứ tự các tab trên web form.

- Kiểm tra các giá trị mặc định của field.

- Form cần được định dạng tối ưu khả năng đọc.

- Kiểm tra số âm.

**1.2.1.c Kiểm thử cookie**

Cookie là tập tin chứa thông tin hệ thống của người dùng, các tệp này được lưu ở vị trí mong muốn và được sử dụng bởi các trình duyệt. Các session đăng nhập, thông tin được lưu lại trong cookie (như session) và có thể được truy xuất cho các trang web. Người dùng có thể kích hoạt hoặc vô hiệu Cookies trong các tùy chọn trình duyệt, kiểm thử để kiểm tra xem cookie có được lưu trữ trong máy của người dùng ở định dạng mã hóa hay không. Kiểm tra ứng dụng bằng cách vô hiệu cookies .

- Kiểm tra ứng dụng sau khi hỏng các cookies.

- Kiểm tra hành vi của ứng dụng sau khi xóa tất cả cookie trên website.

- Kiểm tra cookie có hoạt động trên nhiều duyệt khác nhau hay không.

- Kiểm tra cookie cho đăng nhập xác thực có hoạt động hay không.

- Kiểm tra hành vi của ứng dụng sau khi xoá cookie (session) bằng cách xoá bộ nhớ cache hoặc sau khi cookie hết hạn.

- Kiểm tra đăng nhập vào ứng dụng sau khi xóa cookie (session).

**1.2.1.d Kiểm thử HTML và CSS**

Kiểm thử này kiểm tra xem các công cụ tìm kiếm có thể thu thập dữ liệu trang web của bạn mà không xảy ra bất kỳ lỗi nào, bạn nên kiểm tra tất cả các lỗi cú pháp, màu sắc và tuân thử theo tiêu chuẩn như [*W3C*](https://www.w3.org/)***,* ISO**, ECMA, IETF, WS-I, OASIS.

Quy trình nghiệp vụ bao gồm:

- Kiểm tra luồng xử lý đảm bảo sự hoàn chỉnh của website.

- Kiểm tra các màn hình theo như tài liệu yêu cầu.

### **1.2.2** **Kiểm thử khả năng sử dụng**

Đóng một vai trò quan trọng trong bất kỳ ứng dụng web, ***Usability testing*** đảm bảo kiểm tra tất cả các test case xuất phát từ người dùng. Bao gồm:

**1.2.2.a Kiểm tra điều hướng website**

- Tất cả các tùy chọn như [*UI/UX*](https://mona.media/thiet-ke-ux-ui/), menu, liên kết hoặc các button trên website phải hiển thị và có thể truy cập.

- Điều hướng trang web dễ dàng sử dụng .

- Nội dung hướng dẫn phải rõ ràng và phải đáp ứng được mục đích.

- Tất cả tùy chọn trên header, footer và các điều hướng trái / phải phải nhất quán trên mỗi trang.

**1.2.2.b Kiểm tra nội dung website**

- Không có lỗi chính tả hoặc ngữ pháp lỗi trong nội dung.

- Tích hợp Alt trong hình ảnh.

- Không có ảnh hỏng.

- Xác nhận tính hợp lệ tất cả giao diện người dùng.

-Thực hiện theo một số tiêu chuẩn về xây dựng nội dung trên trang web.

- Tất cả nội dung phải rõ ràng và dễ hiểu.

- Màu tối gây bất bình cho người sử dụng, vì vậy tránh sử dụng theme màu tối.

- Kích thước hình ảnh kích nên phù hợp.

- Anchor text phải hoạt động bình thường.

### **1.2.3 Kiểm thử sự tương thích**

Đảm bảo làm thế nào ứng dụng làm việc trong các môi trường được hỗ trợ, sử dụng ứng dụng web trên các hệ điều hành khác nhau, khả năng tương thích của trình duyệt, khả năng tính toán của phần cứng, cơ sở dữ liệu và khả năng xử lý băng thông mạng. Kiểm thử tương thích đảm bảo rằng “ứng dụng web có hiển thị đúng trên các thiết bị khác nhau không?”.

**1.2.3.a Kiểm tra độ tương thích của trình duyệt**

Các ứng dụng web được hiển thị khác nhau trên các trình duyệt khác nhau, mục tiêu của kiểm tra khả năng tương thích của trình duyệt là đảm bảo rằng không có lỗi nào xảy ra trên các trình duyệt web khác nhau trong khi hiển thị website. Bạn cần đảm bảo rằng ứng dụng web của bạn đang được hiển thị đúng trên các trình duyệt khác nhau cũng như kiểm tra AJAX, [JavaScript](https://mona.media/thiet-ke-website-bang-ngon-ngu-javascript/) và xác thực hoạt động chính xác.

**1.2.3.b Khả năng tương thích hệ điều hành**

Công nghệ mới, sử dụng đồ họa mới hơn, các [API](https://mona.media/api-la-gi/) khác nhau được sử dụng có thể không hoạt động trên nhiều hệ điều hành, bên cạnh đó các text field, button có thể hiển thị khác nhau trên hệ điều hành khác nhau. Vì vậy, **kiểm thử website** cần được thực hiện trên các hệ điều hành khác nhau như Windows, MAC, Solaris, Unix, Linux.

**1.2.3.c Trình duyệt web di động**

Bạn cũng nên kiểm tra khả năng tương thích website trên điện thoại di động, đôi khi có thể xảy ra sự cố tương thích trên các trình duyệt điện thoại di động.

### **1.2.4 Kiểm thử cơ sở dữ liệu**

Độ tin cậy của dữ liệu là một phần quan trọng trong việc kiểm thử cơ sở dữ liệu. Vì vậy, đối với các ứng dụng web nên được kiểm tra một cách kỹ lưỡng. Các hoạt động kiểm tra bao gồm:

- Kiểm tra nếu các truy vấn được thực hiện mà không xảy ra lỗi.

- Thêm mới, cập nhật hoặc xoá dữ liệu trong cơ sở dữ liệu nên duy trì tính toàn vẹn của dữ liệu.

- Truy vấn dữ liệu không nên mất quá nhiều thời gian.

- Kiểm tra việc load dữ liệu và kết quả nhận được với các câu truy vấn dài.

- Dữ liệu nhận được trên cơ sở dữ liệu và hiển thị trên website có chính xác hay không.

### 

### **1.2.5 Kiểm thử giao diện**

Kiểm thử giao diện chủ yếu có ba lĩnh vực cần được kiểm tra: Web Server, Application server và Database server. Đảm bảo rằng tất cả các thông tin liên lạc giữa các server này phải được thực hiện đúng, xác minh kết nối giữa các máy chủ được thiết lập lại hoặc bị mất, kiểm tra xem có bất kỳ xung đột giữa lúc ứng dụng đang hoạt động, trả về bất kỳ lỗi từ web server hoặc database server đến application server sau đó được xử lý và cuối cùng là hiển thị kết quả tới người dùng.

- Web server: kiểm tra xem tất cả các yêu cầu web có đang được chấp nhận và không yêu cầu nào bị từ chối hoặc bị rò rỉ.

- Application server: kiểm tra xem yêu cầu có đang gửi đúng đến server, lỗi có được bắt và hiển thị cho người quản trị.

- Database server: kiểm tra kết quả truy vấn cơ sở dữ liệu.

### **1.2.6 Kiểm thử hiệu năng**

- Kiểm thử website làm việc dưới lượt tải nặng, được phân thành hai phần: kiểm tra tần suất, kiểm tra lượt tải. Bao gồm:

- Kiểm tra thời gian phản hồi của website với tốc độ kết nối khác nhau.

- Kiểm tra website có xử lý được nhiều yêu cầu người dùng vào cùng một thời điểm.

- Kiểm tra website có hoạt động tốt trong thời điểm lượt tải cao.

- Kiểm tra dữ liệu đầu vào lớn từ người dùng.

- Kiểm tra hành vi của website khi kết nối với cơ sở dữ liệu.

- Kiểm tra các phương pháp tối ưu hóa như giảm thời gian tải bằng cách bật [bộ nhớ cache trên trình duyệt](https://mona.media/huong-dan-xoa-cache-trinh-duyet/) và phía máy chủ, nén gzip…

### **1.2.7 Kiểm thử bảo mật**

Được thực hiện để đảm bảo rằng có bất kỳ rò rỉ thông tin nào về mã hoá dữ liệu hay không. Trong website thương mại điện tử, kiểm thử bảo mật đóng một vai trò rất quan trọng, nếu thông tin an toàn thì kiểm tra xem làm thế nào để lưu trữ các thông tin nhạy cảm như thẻ tín dụng, thanh toán hóa đơn…Các hoạt động kiểm tra sẽ bao gồm:

- Kiểm tra truy cập trái phép vào các trang an toàn, nếu người dùng thay đổi từ “https” sang “http” thì thông báo thích hợp sẽ được hiển thị và ngược lại.

- Kiểm tra việc truy cập các trang internal, nếu đăng nhập được yêu cầu thì người dùng nên được chuyển hướng đến trang đăng nhập hoặc thông báo thích hợp sẽ được hiển thị.

- Các thông tin liên quan đến giao dịch, thông báo lỗi, cố gắng đăng nhập nên được ghi vào file log.

- Kiểm tra các tệp tin có bị hạn chế tải xuống hay không.

- Kiểm tra các thư mục web hoặc tập tin web có thể truy cập được trừ khi không được cấu hình để tải xuống.

- Kiểm tra CAPTCHA đã được thêm vào và hoạt động bình thường cho đăng nhập để tự động ngăn chặn các đăng nhập hay chưa.

- Kiểm tra việc cố truy cập thông tin bằng cách thay đổi tham số trong chuỗi truy vấn. Ví dụ: nếu bạn đang chỉnh sửa thông tin và trên URL bạn thấy UserID = 123, hãy thử thay đổi các giá trị tham số này và kiểm xem ứng dụng có cung cấp thông tin người dùng khác không, nên từ chối hiển thị cho trường hợp này để ngăn chặn việc xem thông tin người dùng khác.

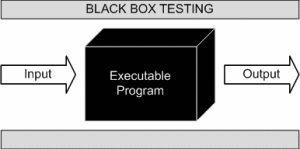
- Kiểm tra session hết hạn sau thời gian được xác định nếu người dùng không thao tác trên website.

- Kiểm tra user/password không hợp lệ.

## **1.3 Kỹ thuật kiểm thử hộp đen**

### **1.3.1 Khái niệm**

Kiểm thử hộp đen (Black Box Testing) là kỹ thuật kiểm thử trong đó chức năng của ứng dụng được kiểm tra mà không cần xem cấu trúc mã nội bộ, chi tiết triển khai và kiến thức về đường dẫn bên trong của phần mềm. Loại kiểm thử này hoàn toàn dựa trên các yêu cầu và thông số kỹ thuật phần mềm.



Hình 1 Kiểm thử hộp đen

Hộp đen ở trên có thể là bất kỳ hệ thống phần mềm nào muốn kiểm tra.

Ví dụ: một hệ điều hành như Windows, một trang web như Google, một cơ sở dữ liệu như Oracle hoặc thậm chí là ứng dụng tùy chỉnh của riêng mình. Trong Kiểm thử hộp đen, có thể kiểm tra các ứng dụng này bằng cách chỉ tập trung vào đầu vào và đầu ra mà không cần biết mã nội bộ của chúng.

### **1.3.2** **Quy trình làm việc của kiểm thử hộp đen**

Các bước giải thích cho quy trình làm việc của Kiểm thử hộp đen:

**Bước 1:** Đầu vào : Yêu cầu và đặc tả chức năng của hệ thống được kiểm tra. Các tài liệu thiết kế cấp cao và khối mã nguồn ứng dụng cũng được kiểm tra. Người tester lựa chọn đầu vào hợp lệ và loại bỏ những đầu vào không hợp lệ.

**Bước 2:** Đơn vị xử lý: Không quan tâm đến hoạt động nội bộ của hệ thống. Người tester xây dựng các test case với đầu vào được lựa chọn và thực thi chúng. Tester cũng

thực hiện load testing, stress testing, security testing. Nếu có bất kì lỗi nào được phát hiện, nó sẽ được sửa và kiểm tra lại.

**Bước 3:** Đầu ra: Sau khi kiểm thử, người tester có được đầu ra mong muốn và chuẩn bị bản báo cáo cuối cùng.

### **1.3.3 Các ưu điểm và nhược điểm của kiểm thử hộp đen**

**1.3.3.a Ưu điểm**

**-** Việc kiểm thử được thực hiện theo quan điểm và ý kiến của người dùng.

- Người test không cần phải biết các ngôn ngữ lập trình hoặc các phần đã được triển khai như thế nào.

- Cả người test và lập trình viên đều không cần can thiệp vào không gian làm việc của nhau.

- Thiết kế kịch bản kiểm thử khá nhanh, ngay khi mà các yêu cầu chức năng được xác định.

**1.3.3.b Nhược điểm**

- Cần một khối lượng mẫu khá lớn để làm dữ liệu đầu vào.

- Chỉ có thể kiểm tra một số lượng nhỏ đầu vào khả thi và nhiều đường dẫn chương trình sẽ bị bỏ qua.

- Tình huống nếu không có thông số kỹ thuật rõ ràng trong nhiều dự án, các trường hợp kiểm thử sẽ khó thiết kế.

- Có thể có sự lặp lại không cần thiết của đầu vào kiểm tra nếu người kiểm thử viên không có sự xác định chắc chắn.

### **1.3.4 Các kỹ thuật kiểm thử hộp đen**

**1.3.4.a Lập bảng quyết định**

Lập bảng quyết định là một trong những kỹ thuật thiết kế testcases được sử dụng rộng rãi để kiểm thử hộp đen. Kỹ thuật này được sử dụng để chọn các trường hợp kiểm thử một cách có hệ thống, nó tiết kiệm thời gian thử nghiệm và bao phủ tốt cho khu vực kiểm thử của phần mềm.

Lập bảng quyết định thích hợp cho các hàm có mối quan hệ logic giữa hai và nhiều hơn hai đầu vào, liên quan đến sự kết hợp chính xác của các đầu vào. Để thiết kế các trường hợp kiểm thử bằng kỹ thuật lập bảng quyết định, chúng ta cần coi các điều kiện là đầu vào và các hành động là đầu ra.  ***Ví dụ:***

Khi muốn sử dụng tài khoản email, cần nhập địa chỉ email và mật khẩu liên quan của nó.

Nếu cả email và mật khẩu đều khớp chính xác, người dùng sẽ được dẫn đến trang chủ của tài khoản email; nếu không, nó sẽ quay lại trang đăng nhập với thông báo lỗi được chỉ định bằng “Email không chính xác” hoặc “Mật khẩu không chính xác” .Ta có thể đăng nhập bằng email và mật khẩu. Cả email và mật khẩu đều là điều kiện tiên quyết và kết quả mong đợi là hành động.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Email**  **(condition 1)** | T | T | F | F |
| **Password**  **(condition 2)** | T | F | T | F |
| **Expected result (action)** | Logged in successfully | Incorrect password | Incorrect email | Incorrect email |

Bảng 1 Bảng quyết định cho chức năng đăng nhập

Trong khi sử dụng kỹ thuật lập bảng quyết định, người tester xác định đầu ra mong đợi, nếu hàm tạo ra đầu ra dự kiến thì nó được thông qua trong quá trình kiểm tra và nếu không thì nó không thành công. Phần mềm không thành công được gửi lại cho nhóm phát triển để sửa lỗi.

**1.3.4.b Phân vùng tương đương**

Phân vùng tương đương là một kỹ thuật kiểm thử phần mềm trong đó dữ liệu đầu vào được chia thành các phân vùng có giá trị hợp lệ và không hợp lệ, và bắt buộc tất cả các phân vùng phải thể hiện cùng một hành vi. Nếu điều kiện của một phân vùng là đúng, thì điều kiện của một phân vùng bằng khác cũng phải đúng, và nếu điều kiện của một phân vùng là sai, thì điều kiện của phân vùng bằng khác cũng phải sai. Nguyên tắc của phân vùng tương đương là, các trường hợp kiểm thử nên được thiết kế để bao phủ mỗi phân vùng ít nhất một lần.

Các phân vùng tương đương bắt nguồn từ các yêu cầu và thông số kỹ thuật của phần mềm. Ưu điểm của phương pháp này là giúp giảm thời gian kiểm thử do số lượng trường hợp kiểm thử từ vô hạn đến hữu hạn ít hơn. Nó có thể áp dụng ở tất cả các cấp của quá trình thử nghiệm.

***Ví dụ:*** Giả sử rằng có một chức năng của ứng dụng phần mềm chấp nhận một số chữ số cụ thể, không lớn hơn và không nhỏ hơn số cụ thể đó.

* Số OTP chỉ chứa sáu chữ số, ít hơn hoặc nhiều hơn sáu chữ số sẽ không được chấp nhận và ứng dụng sẽ chuyển hướng người dùng đến trang lỗi.

|  |  |  |
| --- | --- | --- |
| **INVALID** | **VALID** | **INVALID** |
| Digits <= 5 | Digits = 6 | Digits >= 7 |
| 12345 | 123456 | 123456789 |

*Bảng 1.3.2 Bảng phân vùng cho việc nhập OTP*

* Một chức năng của phần mềm chấp nhận một số điện thoại di động 10 chữ số.

|  |  |  |
| --- | --- | --- |
| **INVALID** | **VALID** | **INVALID** |
| Digits <= 9 | Digits = 10 | Digits >= 11 |
| 987456321 | ***1234567890*** | 01234567890 |

*Bảng 1.3.3 Bảng phân vùng cho việc nhập số điện thoại*

Trong cả hai ví dụ, chúng ta có thể thấy rằng có một phân vùng gồm hai phân vùng hợp lệ và không hợp lệ như nhau, khi áp dụng giá trị hợp lệ, chẳng hạn như OTP gồm sáu chữ số

trong ví dụ đầu tiên và số điện thoại di động gồm 10 chữ số trong ví dụ thứ hai, cả hai phân vùng hợp lệ đều hoạt động giống nhau, tức là được chuyển hướng đến trang tiếp theo.

Hai phân vùng khác chứa các giá trị không hợp lệ, chẳng hạn như 5 hoặc nhỏ hơn 5 và 7 hoặc nhiều hơn 7 chữ số trong ví dụ đầu tiên và 9 hoặc ít hơn 9 và 11 hoặc hơn 11 chữ số trong ví dụ thứ hai và khi áp dụng các giá trị không hợp lệ này, cả hai các phân vùng không hợp lệ hoạt động giống nhau, tức là được chuyển hướng đến trang lỗi.

Chúng ta có thể thấy trong ví dụ này, chỉ có ba trường hợp thử nghiệm cho mỗi ví dụ và đó cũng là nguyên tắc chính của phân vùng tương đương cho biết rằng phương pháp này nhằm giảm số lượng trường hợp thử nghiệm.

**1.3.4.c Phân tích giá trị biên**

Phân tích giá trị biên được sử dụng để kiểm tra các giá trị biên vì các giá trị đầu vào gần biên có khả năng bị lỗi cao hơn. Khi thực hiện phân tích giá trị biên, người tester sẽ tập trung vào trong khi nhập giá trị biên cho dù phần mềm có tạo đầu ra chính xác hay không.

Giá trị biên là những giá trị chứa giới hạn trên và giới hạn dưới của một

biến. Giả sử rằng, tuổi là một biến của bất kỳ hàm nào và giá trị nhỏ nhất của nó là 18 và giá trị lớn nhất là 30, cả 18 và 30 sẽ được coi là giá trị biên.Giả định cơ bản của phân tích giá trị biên là, các trường hợp kiểm thử được tạo bằng cách sử dụng các giá trị biên có nhiều khả năng gây ra lỗi nhất.

Có 18 và 30 là các giá trị biên, đó là lý do tại sao người tester chú ý nhiều hơn đến các giá trị này, nhưng điều này không có nghĩa là các giá trị ở giữa như 19, 20, 21, 27, 29 bị bỏ qua. Các trường hợp kiểm thử được phát triển cho mỗi và mọi giá trị của phạm vi. Kiểm tra các giá trị biên được thực hiện bằng cách tạo các phân vùng hợp lệ và không hợp lệ. Các phân vùng không hợp lệ được kiểm tra vì việc kiểm tra đầu ra trong điều kiện bất lợi cũng rất cần thiết.

***Ví dụ:*** Có một hàm chấp nhận một số từ 15 đến 33, trong đó 15 là giá trị nhỏ nhất và 33 là giá trị lớn nhất của phân vùng hợp lệ, các giá trị khác của phân vùng này là 16, 20, 21, 22, 23, 24, 25 , 26, 27, 28 ... 32. Phân vùng không hợp lệ bao gồm các số nhỏ hơn 15 chẳng hạn như 11, 12, 13, 14 và nhiều hơn 33 như 34, 36, 37 và 40. Người tester phát triển các trường hợp kiểm thử cho cả phân vùng hợp lệ và không hợp lệ để nắm bắt hành vi của hệ thống trên các điều kiện đầu vào khác nhau.

...12 14 15 16 17 18 20 ... 26 28 30 31 33 32 34 36 38 40 ..

Invalid Partition Valid Partition Invalid Partition

|  |  |  |
| --- | --- | --- |
| ***Invalid testcases*** | ***Valid testcases*** | ***Invalid testcases*** |
| *9,11,12,14* | *15,16,17,20,25,30,33* | *34,35,36,37,40* |

*Bảng 1.3.4 Bảng giá trị biên*

Hệ thống phần mềm sẽ được thông qua trong bài kiểm tra nếu nó chấp nhận một số hợp lệ và cho kết quả đầu ra mong muốn, nếu không, thì nó không thành công. Trong một trường hợp khác, hệ thống phần mềm không được chấp nhận các số không hợp lệ và nếu số đã nhập không hợp lệ, thì nó sẽ hiển thị lỗi.

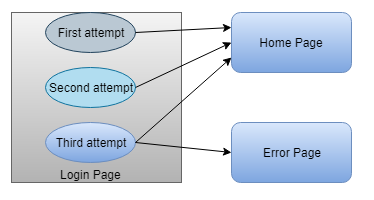
**1.3.4.d Kiểm tra chuyển đổi trạng thái**

Ý nghĩa chung của quá trình chuyển đổi trạng thái là, các hình thức khác nhau của cùng một tình huống, và theo ý nghĩa, phương thức chuyển đổi trạng thái cũng giống nhau. Nó được sử dụng để nắm bắt hành vi của ứng dụng phần mềm khi các giá trị đầu vào khác nhau được cấp cho cùng một chức năng.

Tất cả chúng ta đều sử dụng máy ATM, khi chúng ta rút tiền từ nó, cuối cùng nó sẽ hiển thị chi tiết tài khoản. Bây giờ chúng ta lại thực hiện một giao dịch khác, sau đó nó lại hiển thị chi tiết tài khoản, nhưng chi tiết hiển thị sau giao dịch thứ hai khác với giao dịch đầu tiên, nhưng cả hai chi tiết đều được hiển thị bằng cách sử dụng cùng một chức năng của máy ATM. Vì vậy, cùng một chức năng đã được sử dụng ở đây nhưng mỗi lần đầu ra lại khác nhau, điều này được gọi là chuyển trạng thái. Trong trường hợp kiểm tra một ứng dụng phần mềm, phương pháp này kiểm tra xem chức năng có tuân theo các thông số kỹ thuật chuyển đổi trạng thái khi nhập các đầu vào khác nhau hay không.

Điều này áp dụng cho những loại ứng dụng cung cấp số lần truy cập ứng dụng cụ thể, chẳng hạn như chức năng đăng nhập của ứng dụng bị khóa sau số lần thử không chính xác được chỉ định. Ví dụ, trong chức năng đăng nhập sử dụng email và mật khẩu, nó đưa ra một số lần truy cập ứng dụng cụ thể, sau khi vượt qua số lần thử tối đa, nó sẽ bị khóa với một thông báo lỗi.

Dưới đây là sơ đồ chức năng đăng nhập của một ứng dụng cung cấp số lần thử tối đa là 3 lần và sau khi vượt quá 3 lần thử, nó sẽ được chuyển đến trang lỗi.



Hình 2 Trạng thái chuyển đổi đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| ***State*** | ***Login*** | ***Validation*** | ***Redirected*** |
| S1 | First Attempt | Invalid | S2 |
| S2 | Second Attempt | Invalid | S3 |
| S3 | Third Attempt | Invalid | S5 |
| S4 | Home Page |  |  |
| S5 | Error Page |  |  |

Bảng 2 Bảng chuyển trạng thái của lần đăng nhập đầu tiên không hợp lệ

Trong bảng chuyển trạng thái ở trên, chúng ta thấy rằng trạng thái S1 biểu thị lần đăng nhập đầu tiên. Khi lần thử đầu tiên không hợp lệ, người dùng sẽ được chuyển đến lần thử thứ hai (trạng thái S2). Nếu lần thử thứ hai cũng không hợp lệ, thì người dùng sẽ được chuyển hướng đến lần thử thứ ba (trạng thái S3). Bây giờ nếu lần thử thứ ba và lần cuối cùng không hợp lệ, thì người dùng sẽ được chuyển đến trang lỗi (trạng thái S5).Nhưng nếu lần thử thứ ba là hợp lệ, thì nó sẽ được chuyển đến trang chủ (trạng thái S4).

|  |  |  |  |
| --- | --- | --- | --- |
| ***State*** | ***Login*** | ***Validation*** | ***Redirected*** |
| S1 | First Attempt | Invalid | S2 |
| S2 | Second Attempt | Invalid | S3 |
| S3 | Third Attempt | Valid | S4 |
| S4 | Home Page |  |  |
| S5 | Error Page |  |  |

Bảng 3 Chuyển trạng thái nếu lần thử thứ ba là hợp lệ

Bằng cách sử dụng bảng chuyển đổi trạng thái ở trên, chúng ta có thể thực hiện

kiểm tra bất kỳ ứng dụng phần mềm nào. Chúng ta có thể lập bảng chuyển đổi trạng thái bằng cách xác định đầu ra mong muốn, sau đó thực hiện hệ thống phần mềm để kiểm tra xem nó có đang cho đầu ra mong muốn hay không.

**1.3.4.e Kiểm thử dựa vào đồ thị nguyên nhân – kết quả**

Đồ thị nguyên nhân – kết quả nằm trong kỹ thuật kiểm tra hộp đen nhấn mạnh mối quan hệ giữa một kết quả nhất định và tất cả các yếu tố ảnh hưởng đến kết quả. Nó được sử dụng để viết các trường hợp kiểm thử động.   
 Các trường hợp kiểm tra động được sử dụng khi mã hoạt động động dựa trên đầu vào của người dùng. Ví dụ, trong khi sử dụng tài khoản email, khi nhập email hợp lệ, hệ thống chấp nhận nó, nhưng khi bạn nhập email không hợp lệ, hệ thống sẽ thông báo lỗi. Trong kỹ thuật này, các điều kiện đầu vào được gán với các nguyên nhân và kết quả của các điều kiện đầu vào này với các kết quả.  
 Kiểm thử dựa vào đồ thị nguyên nhân – kết quả dựa trên tập hợp các yêu cầu và đượcsử dụng để xác định các trường hợp kiểm thử tối thiểu có thể có, có thể bao phủ một vùng kiểm thử tối đa của phần mềm.

Ưu điểm chính của kỹ thuật này là làm giảm thời gian thực hiện và chi phí. Kỹ thuật này nhằm mục đích giảm số lượng các trường hợp thử nghiệm nhưng vẫn bao phủ tất cả các trường hợp thử nghiệm cần thiết với độ bao phủ tối đa để đạt được chất lượng ứng dụng mong muốn.

Kiểm thử dựa vào đồ thị nguyên nhân – kết quả chuyển đổi các đặc tả yêu cầu thành mối quan hệ logic giữa các điều kiện đầu vào và đầu ra bằng cách sử dụng các toán tử logic như AND, OR và NOT.

**1.3.4.f Đoán lỗi (Error Guessing Technique)**

Phạm vi của kỹ thuật đoán lỗi hoàn toàn phụ thuộc vào người tester và loại kinh nghiệm trong các lần kiểm thử trước đó vì nó không tuân theo bất kỳ phương pháp và hướng dẫn nào. Các trường hợp kiểm thử được chuẩn bị bởi nhà phân tích để xác định các điều kiện. Các điều kiện được chuẩn bị bằng cách xác định hầu hết các khu vực có thể xảy ra lỗi và sau đó các trường hợp kiểm thử được thiết kế cho chúng.

Mục đích chính của kỹ thuật này là xác định các lỗi phổ biến ở bất kỳ cấp độ kiểm tra nào bằng cách thực hiện các tác vụ sau:

- Nhập khoảng trống vào các trường văn bản.

- Ngoại lệ con trỏ rỗng.

- Nhập các thông số không hợp lệ.

- Chia cho số không.

-Sử dụng giới hạn tối đa của tệp được tải lên.

- Kiểm tra các nút mà không cần nhập giá trị.

***Ví dụ:***Giả sử chúng ta có một tài khoản ngân hàng và chúng ta phải gửi một số tiền vào đó, nhưng số tiền sẽ được chấp nhận trên một phạm vi cụ thể trong đó là 5000- 7000. Vì vậy, ở đây sẽ phải cung cấp giá trị của đầu vào khác nhau cho đến khi nó bao phủ phạm vi kiểm tra tối đa dựa trên kỹ thuật đoán lỗi và xem liệu nó có được chấp nhận hay đưa ra thông báo lỗi:

|  |  |
| --- | --- |
| **Value** | **Description** |
| 6000 | Accept |
| 5555 | Accept |
| 4000 | Error message |
| 8000 | Error message |
| Blank | Error message |
| 100$ | Error message |
| ..... | .... |
| Maximum test coverage |  |

Bảng 4 Mô tả các giá trị đầu vào

***Ghi chú:***

Điều kiện: nếu số tiền> 5000 và số tiền <7000

Và, nếu nhập 5000 → thông báo lỗi (không được chấp nhận dựa trên điều kiện)

7000 → thông báo lỗi (không được chấp nhận dựa trên điều kiện).

## **1.4 Tạo trường hợp kiểm thử (Test case)**

### **1.4.1 Khái niệm**

Test case là tập hợp các trường hợp điều kiện mà Tester dựa vào đó để xác định ứng dụng, hệ thống phần mềm hoặc là 1 trong các chức năng của nó có hoạt động như mong muốn hay không.

Quá trình phát triển test case có thể giúp tìm ra lỗi trong các yêu cầu hoặc thiết kế của ứng dụng, vì nó đòi hỏi phải tư duy hoàn toàn thông qua các hoạt động của ứng dụng. Vì lý do này, việc chuẩn bị test case sớm nhất có thể trong quy trình phát triển phần mềm là rất hữu ích.

Các trường hợp kiểm thử phải bao phủ được toàn bộ luồng xử lý chức năng mô tả trong tài liệu phân tích và thiết kế; các yêu cầu về bảo mật an toàn thông tin, yêu cầu hiệu năng của hệ thống.

### **1.4.2 Cấu trúc của Test case**

**- Test case ID:** Giá trị cần để xác định số lượng trường hợp cần để kiểm thử.

**- Test Case Description:** Mô tả cụ thể ngắn gọn về bước này. Điều kiện tiền đề nếu có.

**- Step:** Mô tả các bước thực thi.

**- Expected Result:** Kết quả mong đợi từ các bước thược hiện trên.

**- Test Data:** Những dữ liệu cần chuẩn bị để test.

**- Bug ID:** ID của bug đã report.

**- Note:** Note lại thông tin liên quan đến khi đã thực hiện test case.

**- Test Result:** Kết quả của việc thực hiện test case: Passed, Failed, N/A, Not Run, Blocked, Passed in previous build.

### **1.4.3 Các bước xác định test case**

- Xác định mục đích test: Cần hiểu rõ đặc tả yêu cầu của khách hàng

- Xác định chức năng cần testing: Cần phải biết làm thế nào phần mềm được sử dụng bao gồm các hoạt động, tổ chức chức năng khác nhau.

- Các bước thực hiện chỉ mô tả các bước thực hiện đứng từ phía người dùng cuối bao gồm nhập dữ liệu, nhấn button. Việc kiểm tra dữ liệu trong DB so với hiện thị trên màn hình nằm ở kết quả mong muốn. Thường được dùng cho các trường hợp kiểm thử kiểm tra lưu, cập nhật, xóa DB SELECT \* FROM … WHERE…

Ví dụ: Tạo 1 email đăng ký thành công +Test tạo 1 email đăng ký thành công. Đã đăng nhập thành công bằng email mới trên giao diện +Vào DB check xem email đó có được lưu vào DB hay không? ( tuy nhiên nhiều nơi ko yêu cầu tester vào CSDL để check)

- Xác định các yêu cầu phi chức năng: yêu cầu phần cứng, hệ điều hành, các khía cạnh an ninh.

- Xác định biểu mẫu cho TCs: bao gồm giao diện UI, chức năng, khả năng tương thích và hiệu suất...

- Xác định tính ảnh hưởng giữa các nguyên tắc mô-đun: TCs nên được thiết kế để có thể che phủ được sự ảnh hưởng của các mô-đun với nhau ở mức độ cao nhất.

### **1.4.4 Xác định các trường hợp kiểm thử**

**- Positive test case:** Thực hiện thử nghiệm trên hệ thống bằng cách cung cấp dữ liệu đầu vào hợp lệ. Nó sẽ kiểm tra xem ứng dụng có hoạt động như mong đợi với đầu vào đúng hay không. Thử nghiệm này được thực hiện để kiểm tra ứng dụng thực hiện những gì nó phải làm.

**- Negative test case** là một biến thể của testing, nó có thể được thực hiện trên hệ thống bằng cách cung cấp dữ liệu đầu vào không hợp lệ. Nó kiểm tra xem một ứng dụng có hoạt động như mong đợi với các đầu vào không hợp lệ hay không. Điều này là để kiểm tra ứng dụng không làm bất cứ điều gì mà nó không phải làm như vậy.

# **CHƯƠNG 2: GIỚI THIỆU VỀ KIỂM THỬ TỰ ĐỘNG**

**2.1 Khái niệm:**

**Kiểm thử tự động**: là một kỹ thuật tự động trong đó người kiểm thử tự viết các tập lệnh và sử dụng phần mềm phù hợp để kiểm thử phần mềm. Nó về cơ bản là một quá trình tự động hóa của một quy trình kiểm thử thủ công. Giống như kiểm thử hồi quy, kiểm thử tự động cũng được sử dụng để kiểm thử ứng dụng theo quan điểm tải, hiệu năng và ứng suất.

Kiểm thử tự động giúp giảm chi phí kiểm thử bằng cách hỗ trợ quá trình kiểm thử thông qua các công cụ phần mềm.

Kiểm thử tự động hay sử dụng phần mềm để kiểm thử với các ưu điểm:

- Có thể thực hiện các kiểm thử một cách liên tục, lặp lại và giảm chi phí cho nhân lực kiểm thử.

- Luôn đảm bảo hoạt động theo một kịch bản duy nhất – không bị ảnh hưởng như với kiểm thử viên.

**2.2 Quy trình kiểm thử tự động**

**Quy trình kiểm thử tự động bao gồm:** nhân viên kiểm thử (tester) sử dụng các kịch bản tự động (automation scripts) và thực thi các script để chạy ứng dụng với sự giúp sức của các công cụ kiểm thử tự động. Một khi script đã sẵn sàng thì việc thực thi kiểm thử có thể diễn ra nhanh chóng và hiệu quả.

Có rất nhiều công cụ hữu ích để viết các kịch bản kiểm thử tự động, trước khi sử dụng các công cụ đó, tốt hơn là hình dung ra quy trình có thể được sử dụng để tự động hóa kiểm thử.

Các hoạt động của kiểm thử tự động:

- Phân tích yêu cầu/Xác định môi trường/công cụ

- Xác định tiêu chí đầu ra

- Lên kế hoạch và kiểm soát

- Thiết lập môi trường kiểm thử

- Triển khai thiết kế kiểm thử

- Thực thi kiểm thử

- Phân tích, báo cáo

**2.3 Mục đích của kiểm thử tự động**

- Giảm bớt công sức và thời gian thực hiện quá trình kiểm thử

- Tăng độ tin cậy.

- Giảm sự nhàm chán cho con người

- Rèn luyện kỹ năng lập trình cho kiểm thử viên

- Giảm chi phí cho tổng quá trình kiểm thử

* 1. **Lý do kiểm thử tự động**

- Không đủ tài nguyên: Khi số lượng TestCase quá nhiều mà kiểm thử viên không thể hoàn tất trong thời gian cụ thể.

- Kiểm tra hồi quy: Nâng cấp phần mềm, kiểm tra lại các tính năng đã chạy tốt và những tính năng đã sửa. Tuy nhiên, việc này khó đảm bảo về mặt thời gian.

- Kiểm tra khả năng vận hành phần mềm trong môi trường đặc biệt (Đo tốc độ trung bình xử lý một yêu cầu của Web server, xác định cấu hình máy thấp nhất mà phần mềm vẫn có thể hoạt động tốt).

* 1. **Rủi ro liên quan đến kiểm thử tự động**

Trong một số trường hợp kiểm thử tự động chứng tỏ được sự quan trọng của nó. Nhưng nó cũng chứa một số rủi ro cần phải được xem xét trước khi áp dụng kiểm thử tự động:

**- Chi phí khởi đầu cho kiểm thử tự động rất cao**: Bất kỳ dự án lớn nào cũng cần một đội ngũ kiểm thử, việc thuê số lượng người thử nghiệm thủ công thực sự hiệu quả về chi phí. Chi phí tự động hóa là nhiều hơn cho thiết lập ban đầu như: chi phí mua công cụ tự động hóa, đào tạo và bảo trì các kịch bản kiểm thử chi phí rất cao.

**- Kiểm thử tự động không phải là 100%:** Kiểm thử tự động không thể là 100%. Kiểm thử tự động có thể được áp dụng cho việc kiểm thử hồi quy, hiểu suất (performance), tải (load/stress). Các lĩnh vực như giao diện người dùng, tài liệu, cài đặt, khả năng tương thích và phục hồi thì phải được thực hiện thủ công.

**- Không kiểm thử tự động với trường hợp UI không cố định**: Nếu giao diện người dùng luôn thay đổi, chi phí liên quan đến bảo trì tập lệnh sẽ rất cao.

**- Người kiểm thử nên có kiến thức lập trình tốt:** Người kiểm thử có kiến thức lập trình tốt có thể sử dụng công cụ tự động hóa tốt hơn và giải quyết vấn đề ứng dụng tốt hơn.

**CHƯƠNG 3. TÌM HIỂU VỀ SELENIUM WEBDRIVER**

## **3.1 Selenium Webdriver:**

Selenium Webdriver được coi là một trong những tool test auto mạnh mẽ và thông dụng nhất trong bộ Selenium. Webdriver được xem như là phiên bản mở rộng sau khi cải tiến các điểm không cần thiết cũng như hạn chế của Selenium RC, như không yêu cầu máy chủ Selenium phải bắt đầu trước khi thực thi các kịch bản test.

Cũng không giống như Selenium IDE, phiên bản WebDriver tương thích với nhiều trình duyệt và nền tảng hơn hẳn.

WebDriver trực tiếp gọi tới trình duyệt web và toàn bộ kịch bản kiểm thử được thực thi theo cách này. WebDriver sử dụng sự hỗ trợ, khả năng của trình duyệt để tự động hóa.

## **3.2 Các tính năng của Selenium Webdriver:**

- Tương thích với nhiều trình duyệt:

WebDriver hỗ trợ nhiều trình duyệt và phiên bản khác nhau .Từ những trình duyệt nổi nhất như Chrome, Firefox, Safari... đến những trình duyệt hiếm gặp như HtmlUnit. Tạo nên sự tiện dụng của nó.

-Hỗ trợ nhiều ngôn ngữ:

Web Driver hỗ trợ viết kịch bản kiểm thử bằng các ngôn ngữ khác nhau như Java, JavaScript, C#, PHP, Python, Perl, Ruby… và có thể sử dụng các điều kiện if else, các vòng lặp để tăng tính chính xác cho kịch bản kiểm thử.

-Tốc độ:

Khi so sánh với các công cụ khác của bộ Selenium, WebDriver là công cụ nhanh nhất trong số tất cả do tương tác trực tiếp từ hệ điều hành tới trình duyệt

Do đó, WebDriver tận dụng được khả năng tương thích với trình duyệt để tự động hóa.  
**3.3 Các câu lệnh thường dùng trong Selenium Webdriver:**

### **Các câu lệnh trình duyệt**

- Câu lệnh ***get***:

+ Mục đích : Dùng để mở một trang web mới

+ Cú pháp: driver.get(URL)

Trong đó: URL là url để tải trang , vì thế nên sử dụng một url đầy đủ.

- Câu lệnh lấy tiêu đề ***getTitle***:

+Mục đích: Câu lệnh này sử dụng để lấy tiêu đề của trang.

+ Cú pháp: driver.getTitle();

- Câu lệnh lấy URL hiện tại ***getCurrentUrl***:

+ Mục đích: Câu lệnh này sử dụng để lấy URL của trang hiện tại.

+ Cú pháp: driver.getCurrentUrl();

- Câu lệnh đóng trang web ***close***:

+ Mục đích: Câu lệnh này dùng để đóng cửa sổ hiện tại của trình duyệt.

+ Cú pháp: driver.close();

- Câu lệnh thoát trình duyệt ***quit***:

+ Mục đích: Câu lệnh này dùng để đóng cửa sổ hiện tại của trình duyệt.

+ Cú pháp: driver.quit();

### **Các câu lệnh trong WebElement**

- Tìm phần tử bằng ID (**By ID**):

+Mục đích: Tìm vị trí của phần tử bằng ID, nếu tìm được id phù hợp sẽ

trả về vị trí của phần tử, nếu không có phần từ phù hợp với id sẽ xuất hiện NoSuchElementException.

+ Cú pháp: driver.findElement (By.id(“”));

- Tìm phần tử bằng Name (**By Name**):

+ Mục đích: Tìm vị trí của phần tử bằng name, nếu tìm được name phù

hợp sẽ trả về vị trí của phần tử, nếu không có phần tử phù hợp với name sẽ xuất hiện NoSuchElementException.

+ Cú pháp: driver.findElement (By.name(“”));

- Tìm phần tử bằng Class Name (**By className**):

+ Mục đích: Tìm phần tử dựa trên giá trị của thuộc tính “class”.

+ Cú pháp: driver.findElement (By.className(“”));

- Tìm phần tử bằng tag names (**By tagName**):

+ Mục đích: Tìm phần tử dựa vào tag names.

+ Cú pháp: driver.findElement (By.tagName(“”));

- Tìm phần tử bằng link Text (**By linkText**):

+ Mục đích: Có thể tìm phần tử của thẻ a bằng tên của link.

+ Cú pháp: driver.findElement (By.linkText(“”));

### **3.3.3 Các câu lệnh chuyển hướng trình duyệt**

- Câu lệnh To:

+Mục đích: Lệnh này dùng để chuyển hướng đến trang web hoặc URL yêu cầu.

+ Cú pháp: driver.navigate.to(URL);

- Câu lệnh Forward:

+ Mục đích: Lệnh này dùng để đi đến trang tiếp theo , giống với nút

forward trên trình duyệt.

+ Cú pháp: driver.navigate.forward();

- Câu lệnh Back:

+ Mục đích: Lệnh này dùng để quay về trang trước, giống với nút back

trên trình duyệt.

+ Cú pháp: driver.navigate.back();

### **3.3.4 Các câu lệnh wait (trên NodeJs)**

- Câu lệnh ImplicitlyWait:

+ Mục đích: Đợi một thời gian nhất định trước khi ném một ngoại lệ khi

không thể tìm thấy các phần tử trên trang web. Chờ đợi ngầm luôn thực hiện trong suốt thời gian trình duyệt mở. Điều này có nghĩa tìm kiếm bất kỳ các phần tử nào trên trang có thể mất thời gian chờ đợi ngầm cho việc thiết lập câu lệnh này.

+ Cú pháp:

driver.manage().setTimeouts( { implicit: 10000 } );

- Câu lệnh sleep:

+ Mục đích: Câu lệnh này hiếm khi được sử dụng vì nó luôn luôn buộc

các trình duyệt chờ đợi một thời gian cụ thể. Nếu sử dụng câu lệnh

này có thể xác định giá trị thời gian chờ đợi cao hơn nhiều khi làm các

trường hợp kiểm thử.

+ Cú pháp: driver.sleep(1000);  
 Kết hợp những câu lệnh ở trên ta có thể thực thi một testcase hoàn chỉnh.

# 

# **CHƯƠNG 4. WEBSITE BÁN HÀNG ONLINE**

## **4.1 Các chức năng chính của website**

Website bán hàng online cho phép người dùng truy cập website của shop để tìm kiếm các sản phẩm cần mua, sau đó có thể tiến hành đặt hàng.

Website được xây dựng dựa trên hệ quản trị cơ sở dữ liệu MySQL để lưu trữ dự liệu và dùng ngôn ngữ Nodejs để tiền hành thành lập các chức năng.

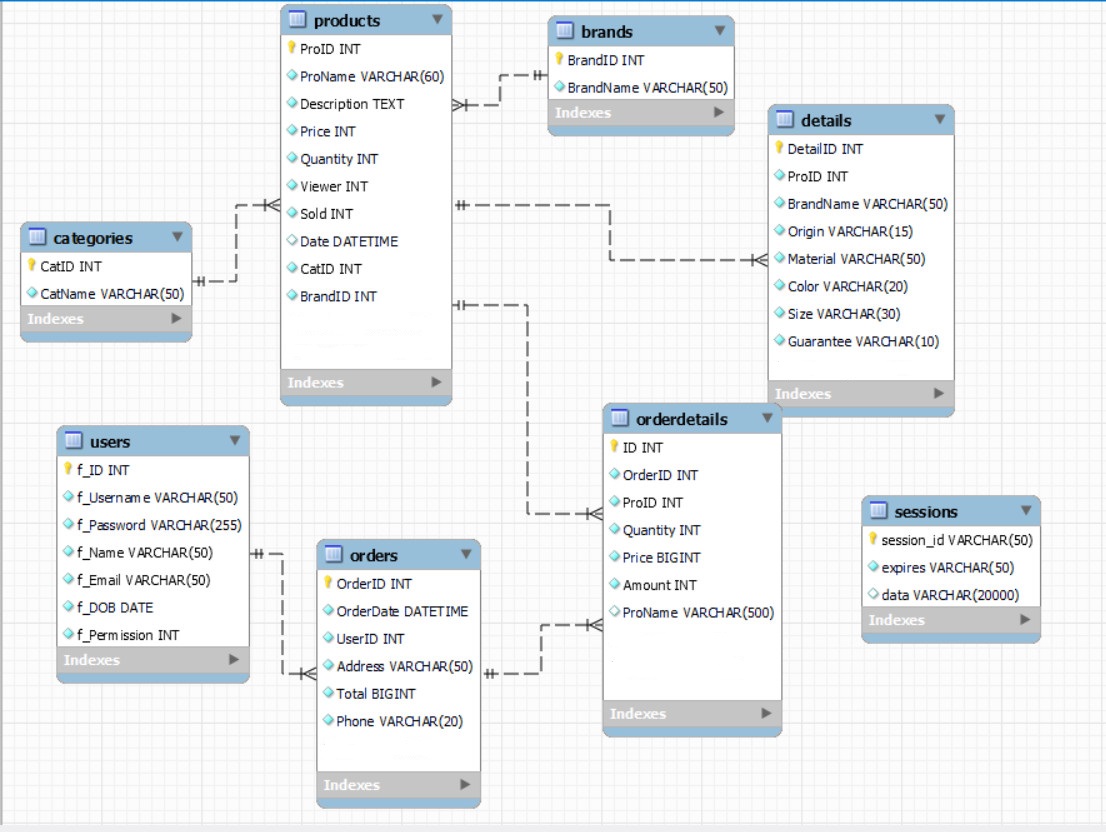
Mô tả các tác nhân:

* Khách hàng
* Đăng ký tài khoản mới.
* Đăng nhập
* Tìm kiếm sản phẩm
* Thêm sản phẩm vào giỏ hàng
* Đặt hàng online
* Xem lịch sử đặt hàng
* Admin
* Đăng nhập
* Quản lý danh mục sản phẩm
* Quản lý thương hiệu
* Quản lý sản phẩm

## **4.2 Database**

Sử dụng hệ quản trị cơ sở dữ liệu MySQL để lưu trữ databse cho website.

### **4.2.1 Sơ đồ diagram:**



Hình 3 Sơ đồ diagram của database bán hàng.

### **4.2.2 Các bảng thành phần**

**4.2.2.a Bảng Categories – Bảng danh mục sản phẩm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Loại | Kiểu | Ý nghĩa |
| 1 | CatID | Khóa chính | Int | Mã danh mục sản phẩm |
| 2 | CatName |  | Nvarchar(50) | Tên danh mục sản phẩm |

Bảng 5 Danh mục sản phẩm

**4.2.2b Bảng Brands – Bảng thương hiệu sản phẩm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Loại | Kiểu | Ý nghĩa |
| 1 | BrandID | Khóa chính | Int | Mã sản phẩm |
| 2 | BrandName |  | Nvarchar(50) | Tên sản phẩm |

Bảng 6 Thương hiệu sản phẩm

**4.2.2.c Bảng Products – Bảng sản phẩm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Loại | Kiểu | Ý nghĩa |
| 1 | ProID | Khóa chính | Int | Mã thương hiệu sản phẩm |
| 2 | ProName |  | Nvarchar(60) | Tên thương hiệu sản phẩm |
| 3 | Description |  | Text | Mô tả sản phẩm |
| 4 | Price |  | Int | Giá thành sản phẩm |
| 5 | Quantity |  | Int | Số lượng sản phẩm |
| 6 | Viewer |  | Int | Số lượt xem sản phẩm |
| 7 | Sold |  | Int | Số lượng sản phẩm đã bán |
| 8 | Date |  | Datetime | Ngày thêm sản phẩm |
| 9 | CatID |  | Int | Mã danh mục sản phẩm |
| 10 | BrandID |  | Int | Mã sản phẩm |

Bảng 7 Sản phẩm

**4.2.2.d Bảng Details – Bảng chi tiết sản phẩm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Loại | Kiểu | Ý nghĩa |
| 1 | DetailID | Khóa chính | Int | Mã chi tiết sản phẩm |
| 2 | ProID |  | Int | Mã sản phẩm |
| 3 | BrandName |  | Varchar(50) | Tên thương hiệu sản phẩm |
| 4 | Origin |  | Varchar(15) | Xuất sứ sản phẩm |
| 5 | Material |  | Varchar(50) | Chất liệu sản phẩm |
| 6 | Color |  | Varchar(20) | Màu sản phẩm |
| 7 | Size |  | Varchar(30) | Kích thước sản phẩm |
| 8 | Guarantee |  | Varchar(10) | Thời gian bảo hành |

Bảng 8 Chi tiết sản phẩm

**4.2.2.e Bảng Orders – Bảng đặt hàng**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Loại | Kiểu | Ý nghĩa |
| 1 | OrderID | Khóa chính | Int | Mã đơn đặt hàng |
| 2 | OrderDate |  | Datetime | Ngày đặt hàng |
| 3 | UserID |  | Int | Mã người dùng |
| 4 | Address |  | Varchar(50) | Địa chỉ giao hàng |
| 5 | Total |  | Bigint | Tổng tiền đơn hàng |
| 6 | Phone |  | Varchar(15) | Số điện thoại đặt hàng |

Bảng 9 Đặt hàng

**4.2.2.f Bảng OrderDetails – Bảng chi tiết đặt hàng**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Loại | Kiểu | Ý nghĩa |
| 1 | ID | Khóa chính | Int | Mã đơn hàng chi tiết |
| 2 | OrderID |  | Int | Mã đơn đặt hàng |
| 3 | ProID |  | Int | Mã sản phẩm |
| 4 | Quantity |  | Int | Số lượng sản phẩm |
| 5 | Price |  | Bigint | Giá từng sản phẩm |
| 6 | Amount |  | Int | Tổng tiền |
| 7 | ProName |  | Varchar(200) | Tên sản phẩm |

Bảng 10 Chi tiết đặt hàng

**4.2.2.g Bảng Users – Bảng tài khoản**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Loại | Kiểu | Ý nghĩa |
| 1 | F\_ID | Khóa chính | Int | Mã tài khoản |
| 2 | Username |  | Varchar(32) | Tên đăng nhập |
| 3 | Password |  | Varchar(255) | Mật khẩu đăng nhập |
| 4 | Name |  | Varchar(50) | Tên người dùng |
| 5 | Email |  | Varchar(50) | Email người dùng |
| 6 | DOB |  | Date | Ngày sinh |
| 7 | Permission |  | Int | Phân quyền tài khoản |

Bảng 11 Tài khoản

**4.2.2.h Bảng Sessions – Bảng phiên đăng nhập**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Loại | Kiểu | Ý nghĩa |
| 1 | SessionID | Khóa chính | Varchar(50) | Mã phiên đăng nhập |
| 2 | Expires |  | Varchar(50) |  |
| 3 | Data |  | Varchar(2000) |  |

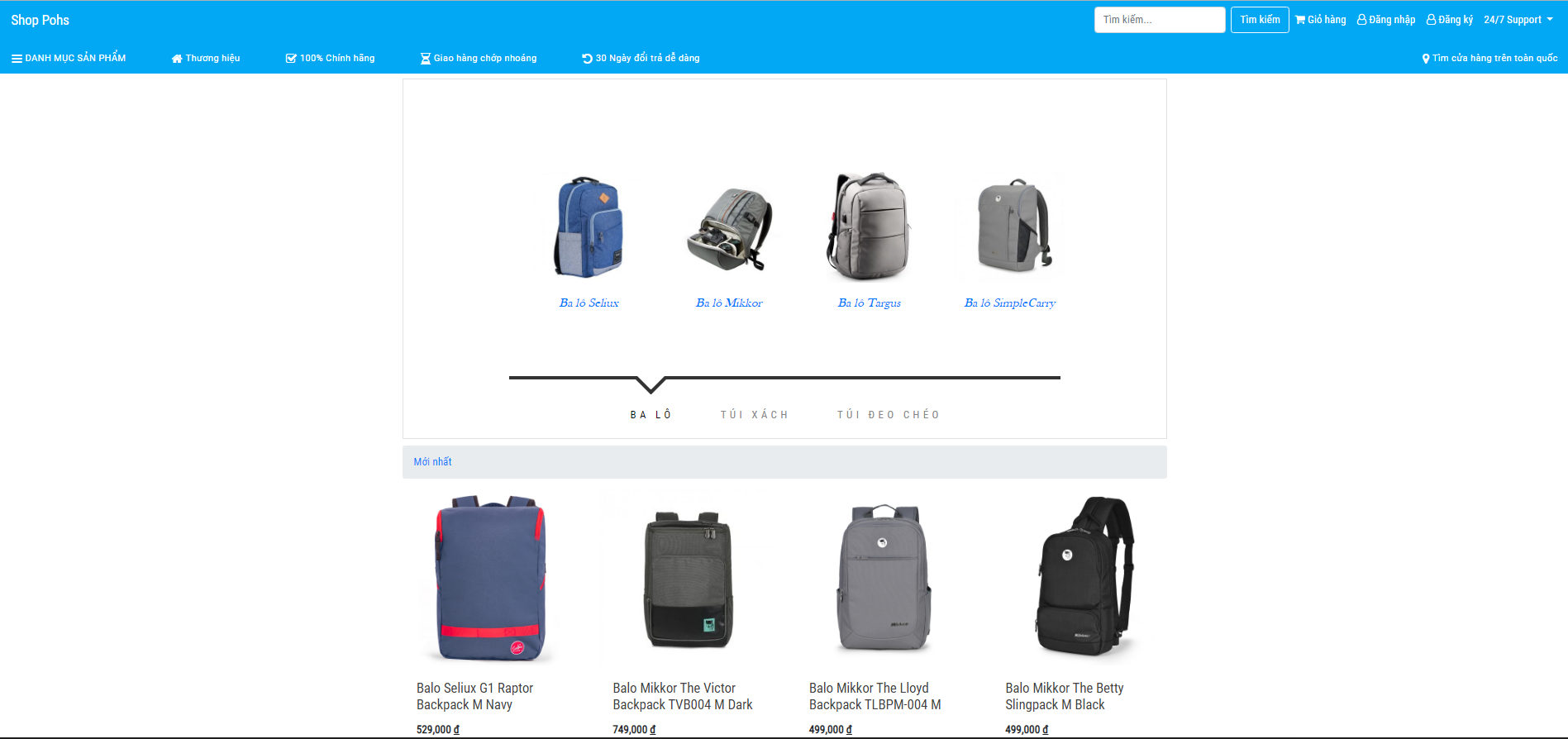
Bảng 12 Phiên đăng nhập

## **4.3 Giao diện website**

### **4.3.1 Giao diện người dùng**

**4.3.1a Trang chủ**

**-** Hiển thị các sản phẩm của trang web, có thể lọc sản phẩm theo danh mục sản phẩm, hoặc theo thương hiệu, cũng có thể tìm kiếm sản phẩm cần mua.

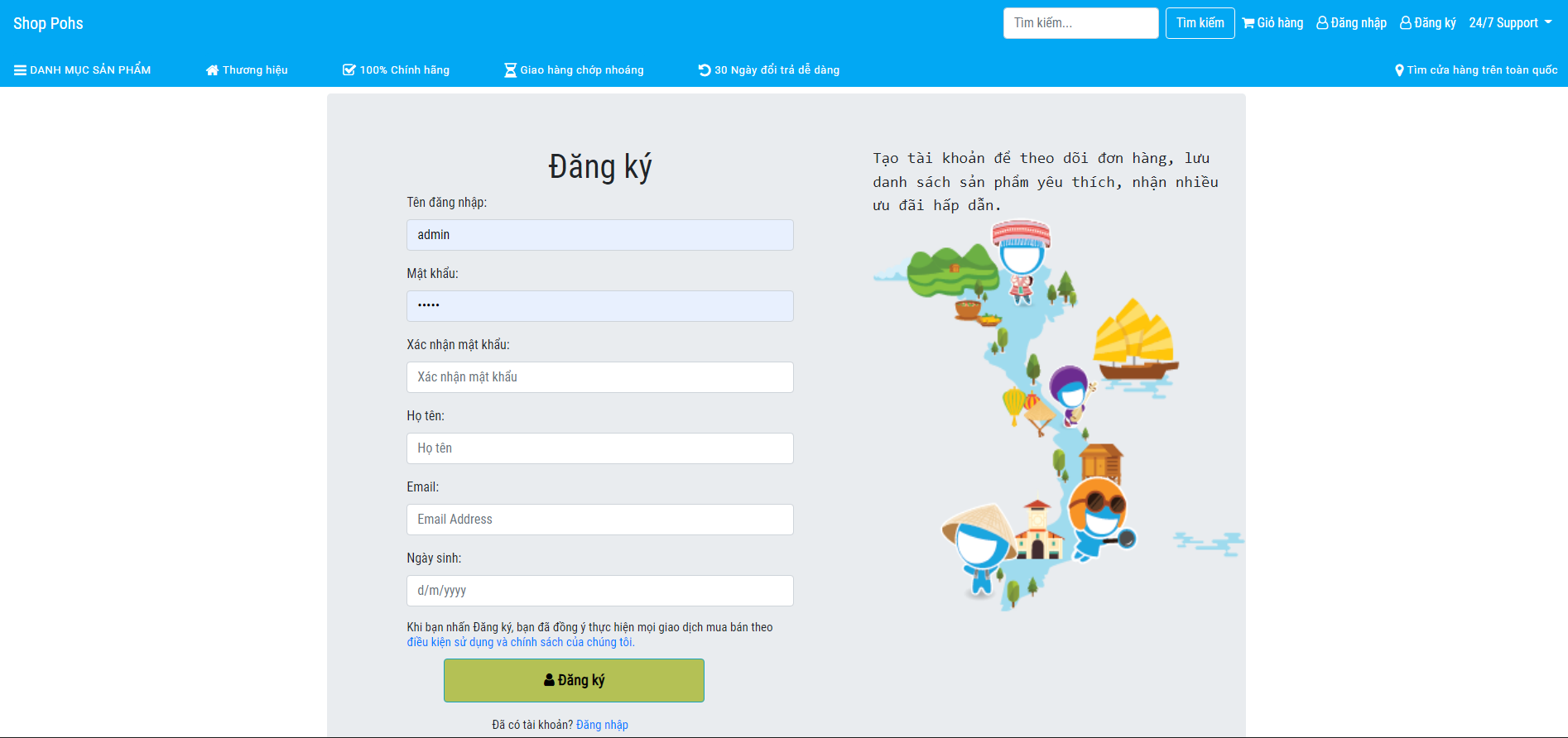
*Hình 4.Màn hình Trang chủ*

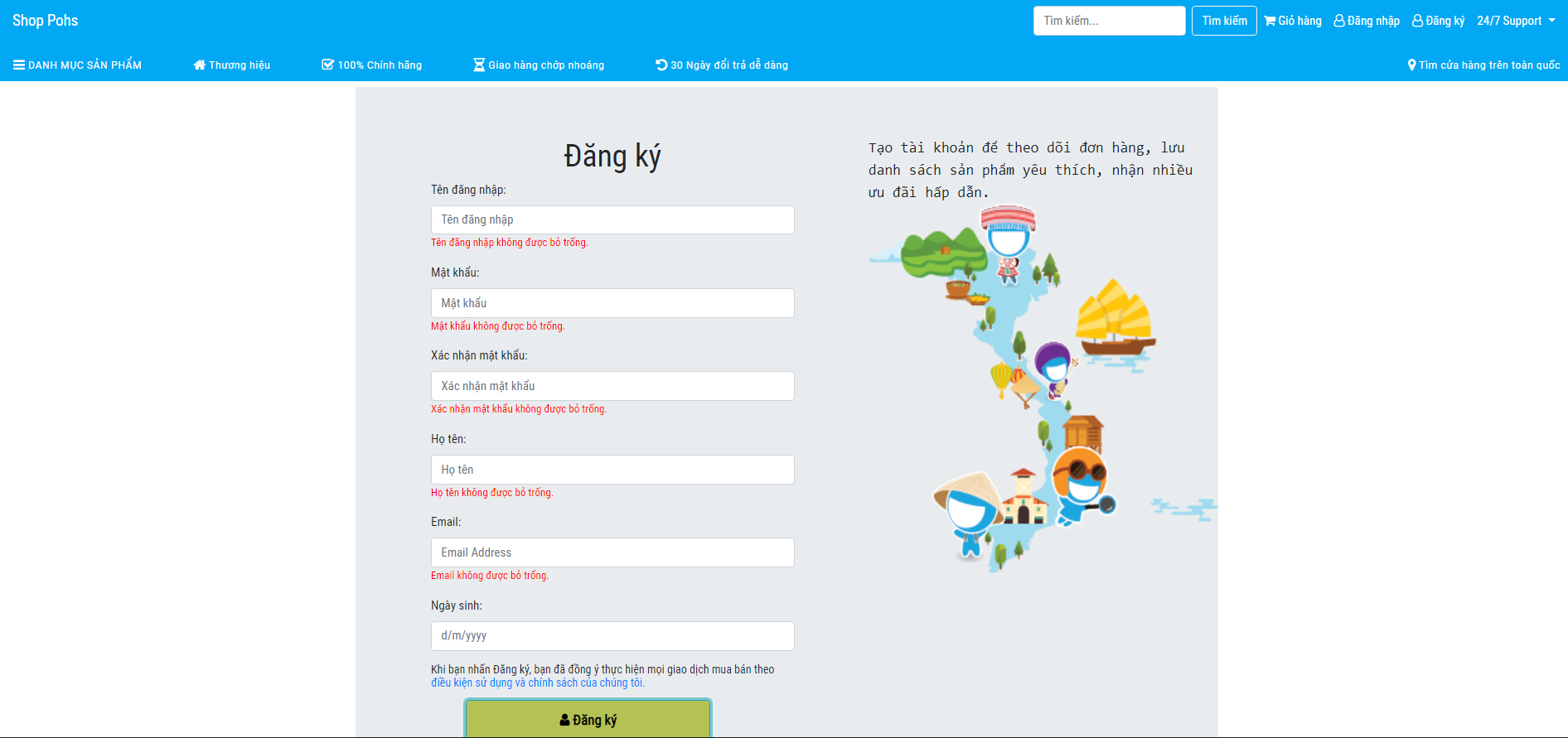
**4.3.1b Đăng ký**

- Khách hàng có thể đăng ký tài khoản để dễ dàng trong việc mua sắm, đặt hàng và xem lịch sử đơn hàng.

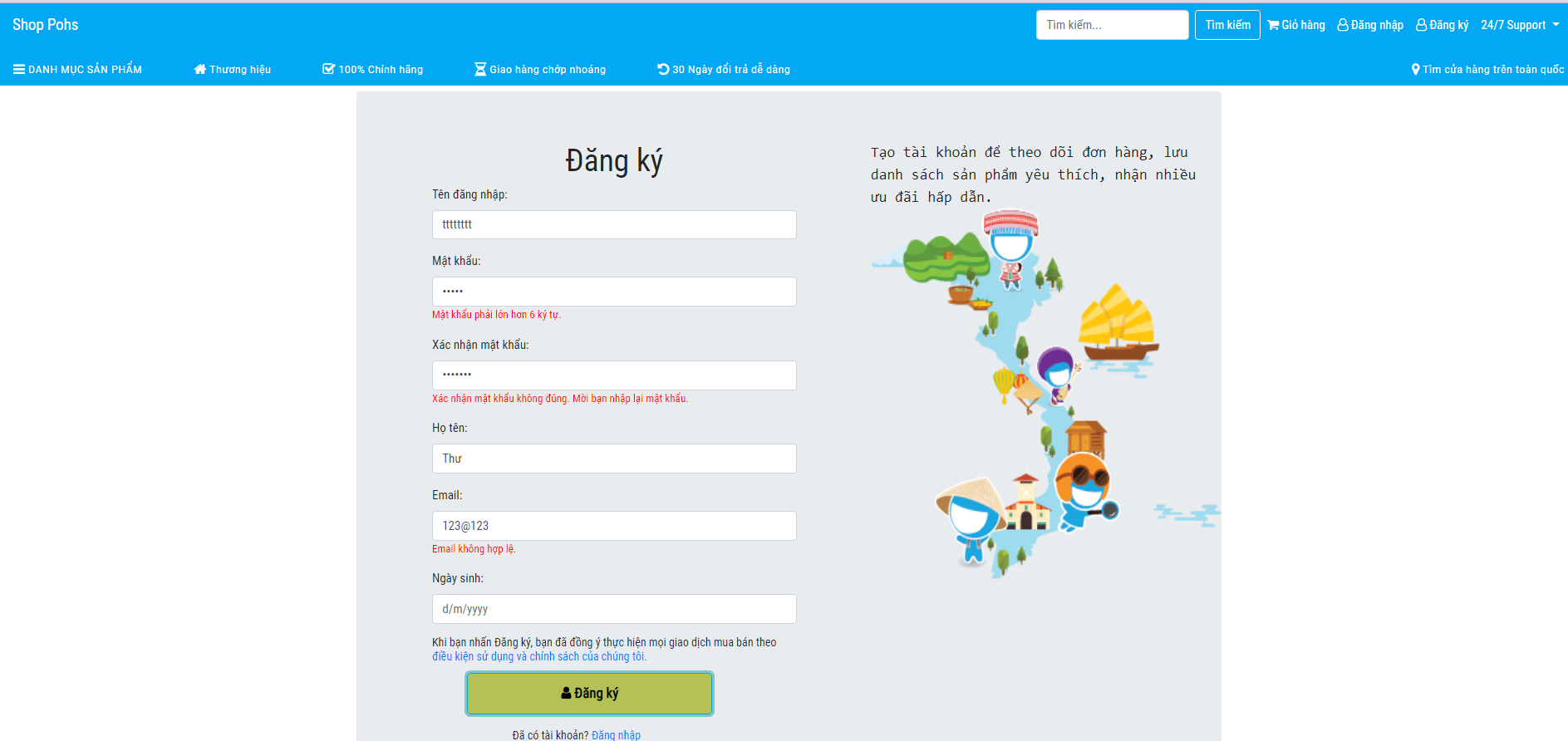
- Các thông tin bắt buộc nhập, nếu bỏ trống thông tin sẽ hiển thị thông báo lỗi.

- Mật khẩu phải ít nhất 6 ký tự, xác nhận mật khẩu phải giống với mật khẩu, email phải đúng định dạng. Nếu nhập sai, trang web sẽ xuất hiện các thông báo lỗi.

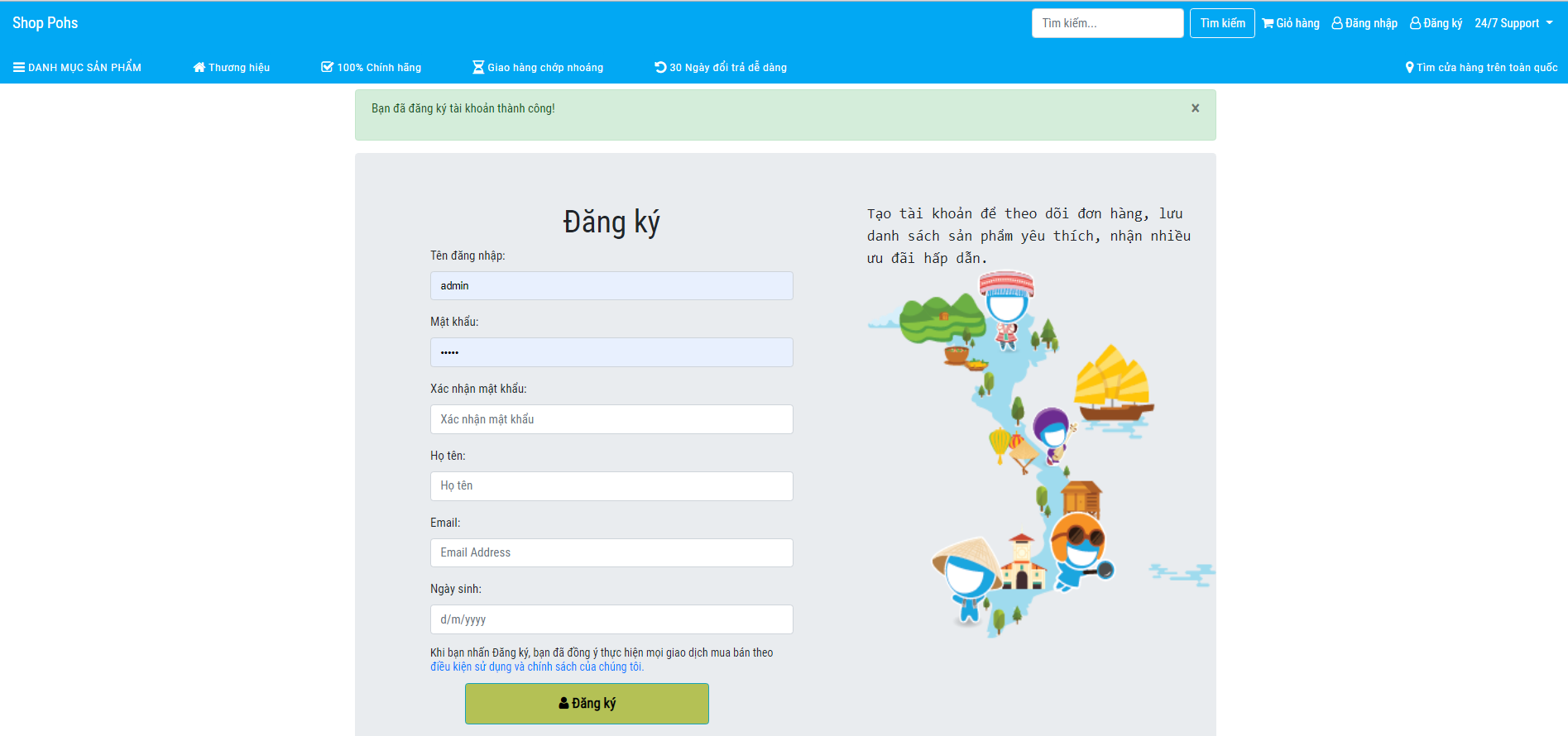
*Hình 5. Màn hình Đăng ký*



Hình 6 Màn hình Đăng ký khi bỏ trống thông tin



Hình 7 Màn hình Đăng ký khi nhập thông tin không hợp lệ



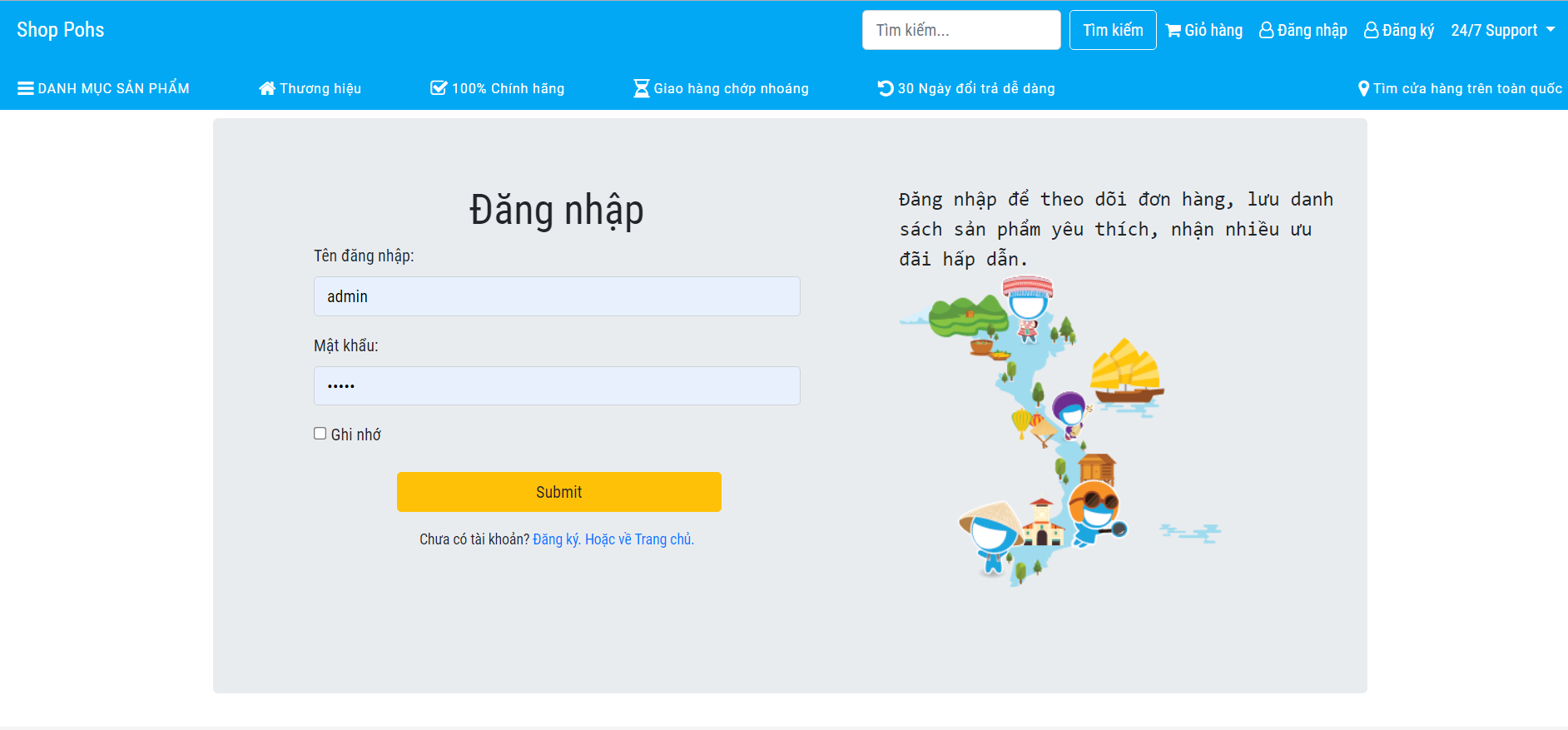
Hình 8 Màn hình Đăng ký thành công

**4.3.1c Đăng nhập**

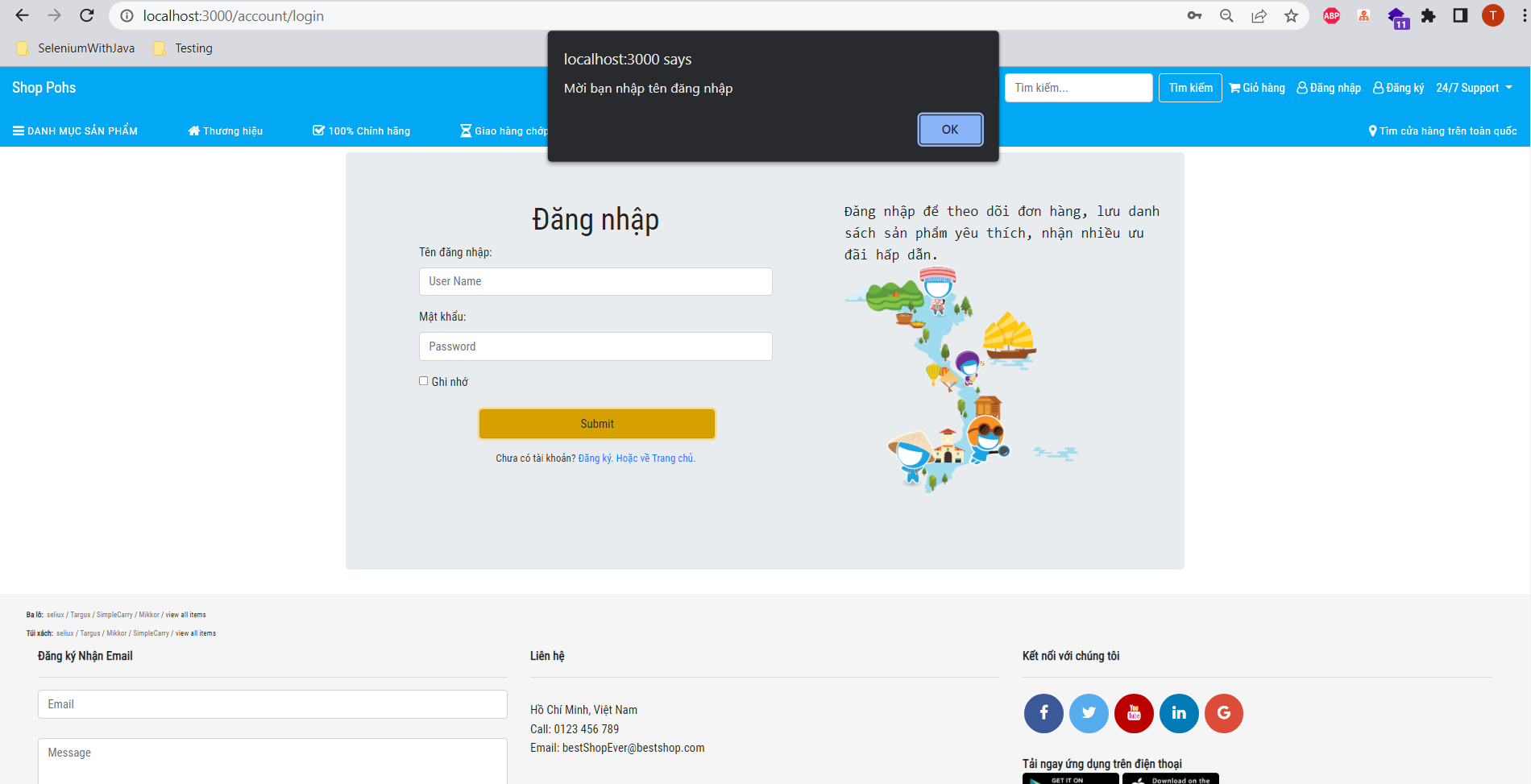
**-** Dùng để đăng nhập vào hệ thống với tài khoản và mật khẩu đã được đăng ký thành công.

- Hiển thị alert thông báo lỗi nếu bỏ trống tên đăng nhập hoặc mật khẩu.

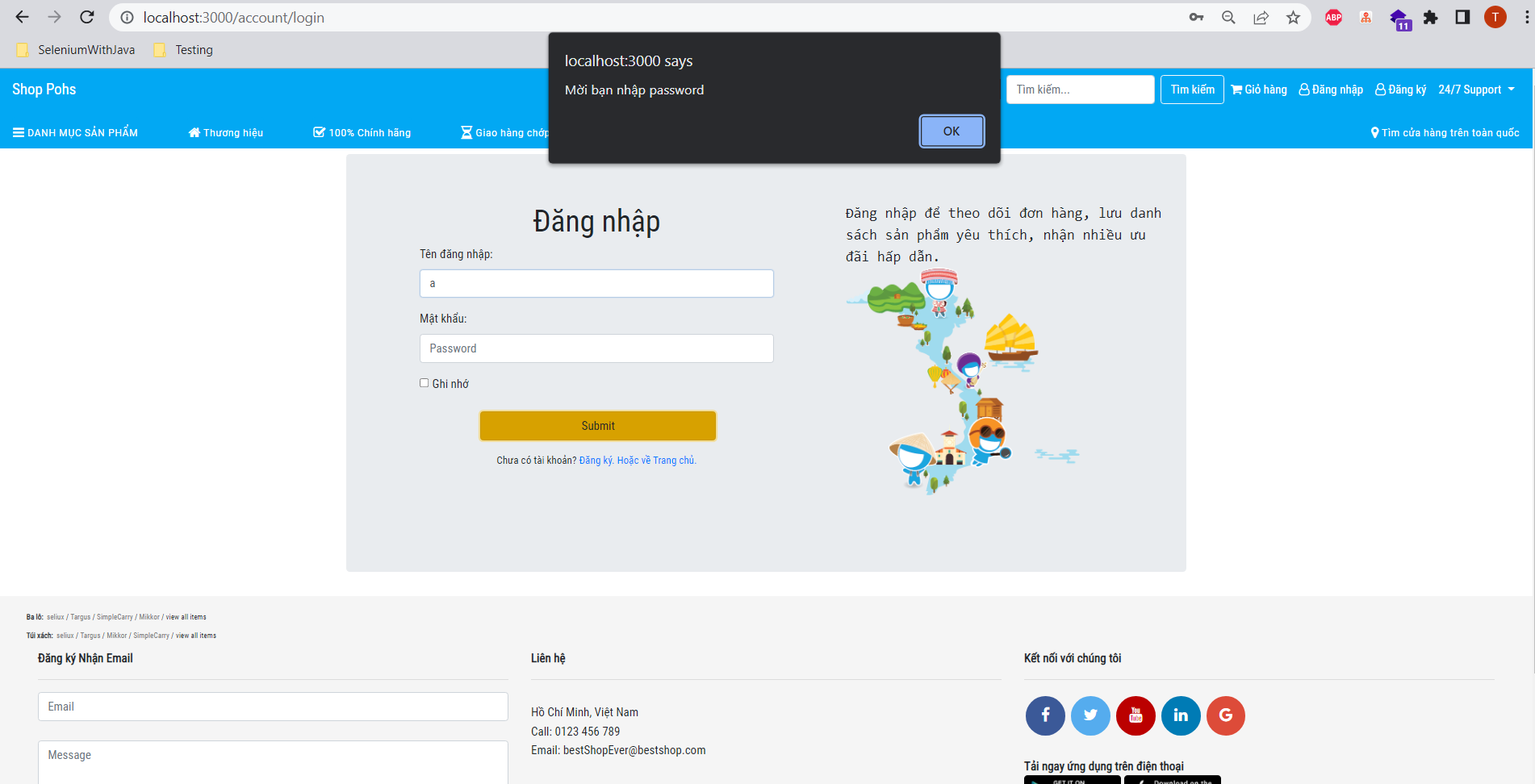
- Hiển thị thông báo lỗi nếu nhập sai tên đăng nhập hoặc mật khẩu.



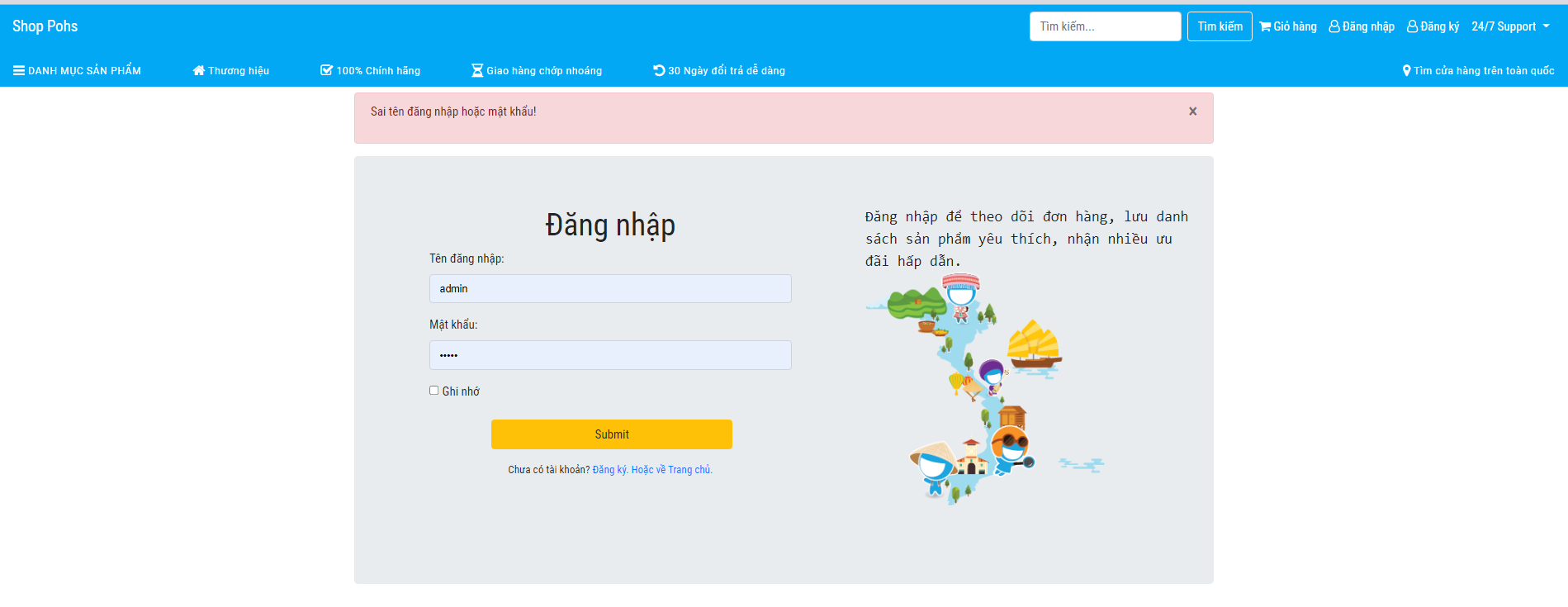
Hình 9 Màn hình đăng nhập



Hình 10 Màn hình Đăng nhập khi bỏ trống tên đăng nhập



Hình 11 Màn hình Đăng nhập khi bỏ trống mật khẩu



Hình 12 Màn hình Đăng nhập khi nhập sai thông tin đăng nhập

**4.3.1d Tài khoản người dùng**

- Quản lý thông tin tài khoản của người dùng: tên đăng nhập, email, họ tên, ngày sinh và mật khẩu. Trong đó:

+ Tên đăng nhập không được chỉnh sửa.

+ Các thông tin khác có thể cập nhật với giá trị hợp lệ.

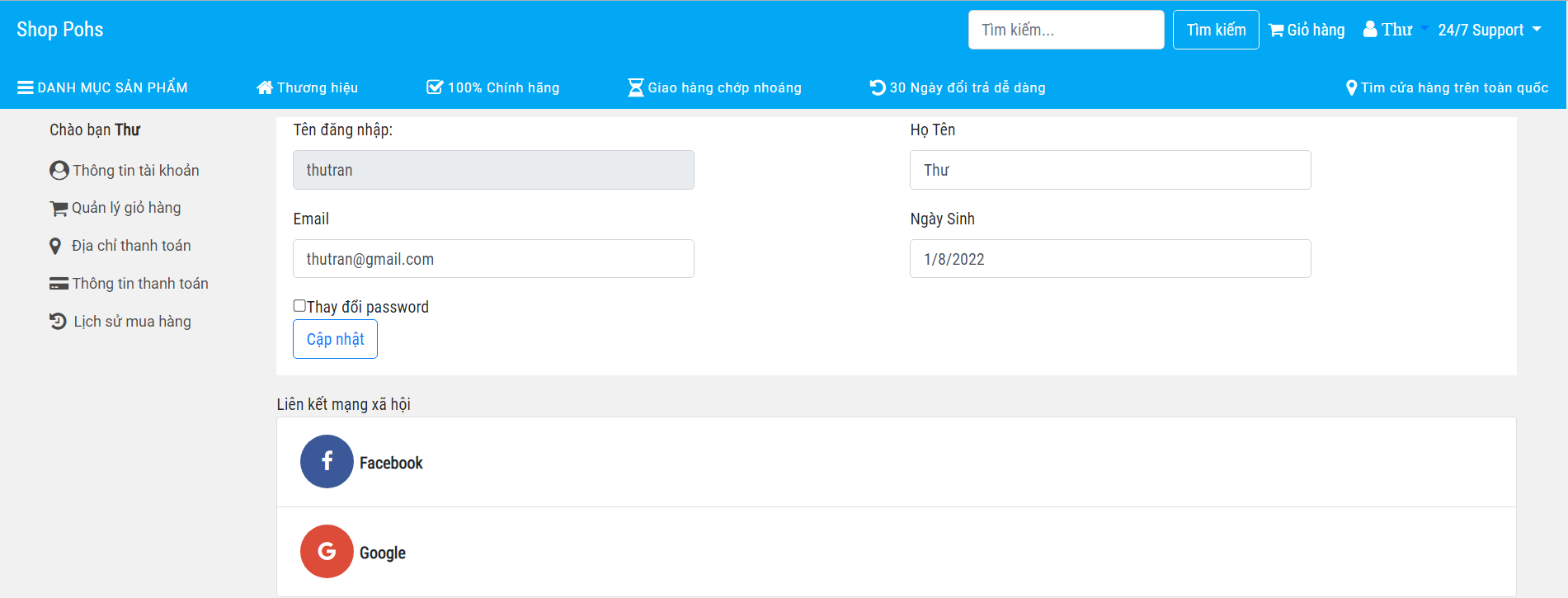
+ Thay đổi mật khẩu: mật khẩu mới phải có ít nhất 6 ký tự và xác nhận mật khẩu phải giống với mật khẩu đã nhập. Nếu thông tin nhập sai định dạng thì button “Cập nhật” bị vô hiệu hóa.

- Quản lý giỏ hàng: Các sản phẩm muốn mua phải được thêm vào giỏ hàng, sau đó có thể tiến hành đặt hàng với thông tin là số điện thoại và địa chỉ.

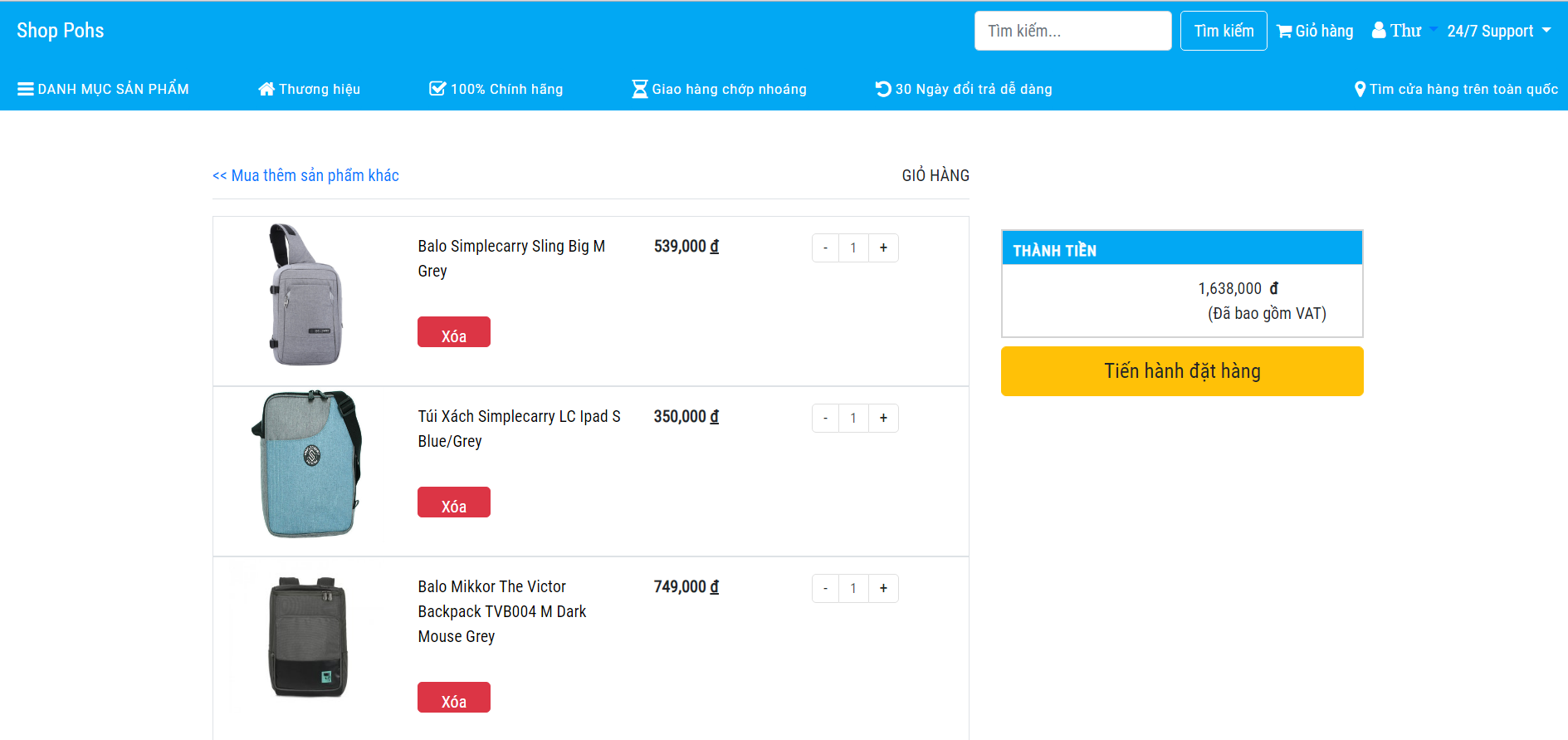
+ Nếu giỏ hàng trống thì không thể đặt hàng.

+ Nếu để trống số điện thoại hoặc địa chỉ thì không thể đặt hàng.

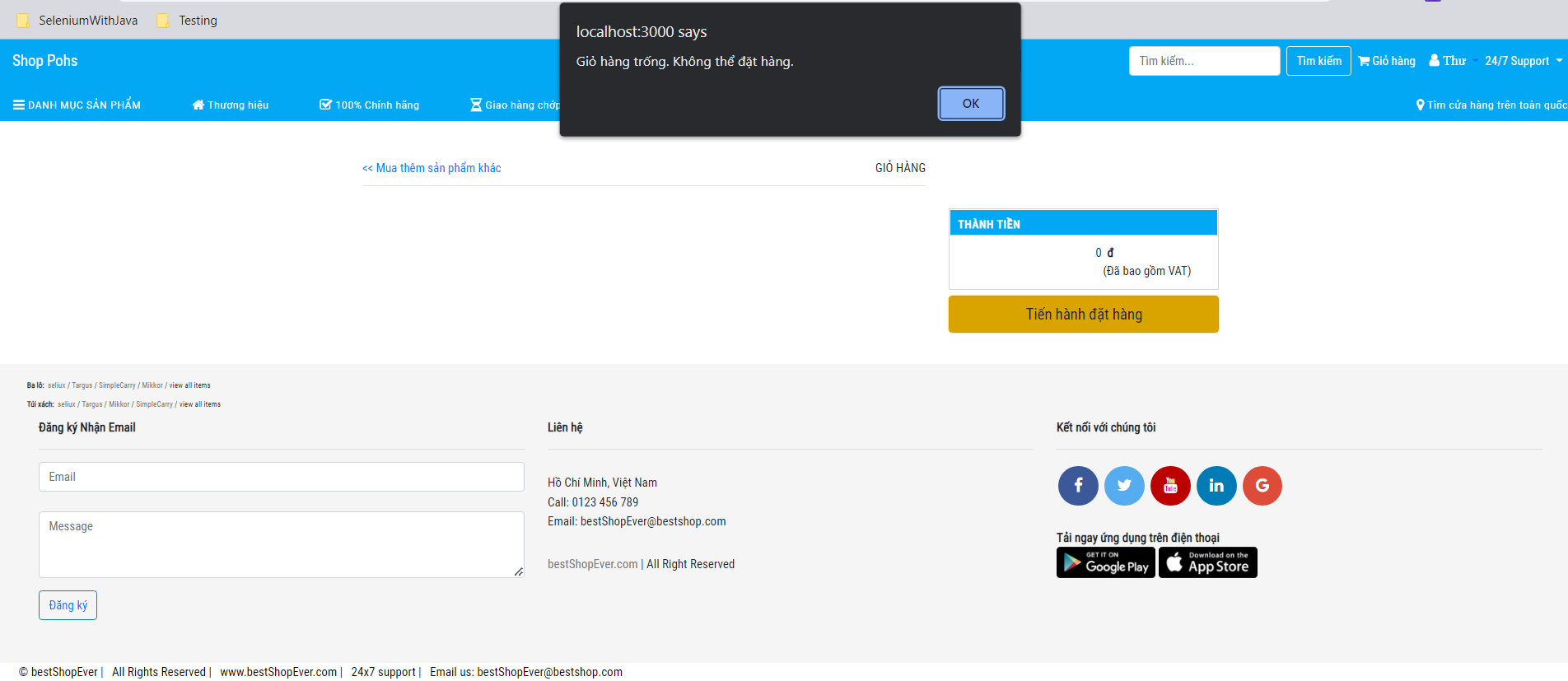
- Quản lý lịch sử đặt hàng: Các đơn đặt hàng thành công sẽ được lưu tại đây.



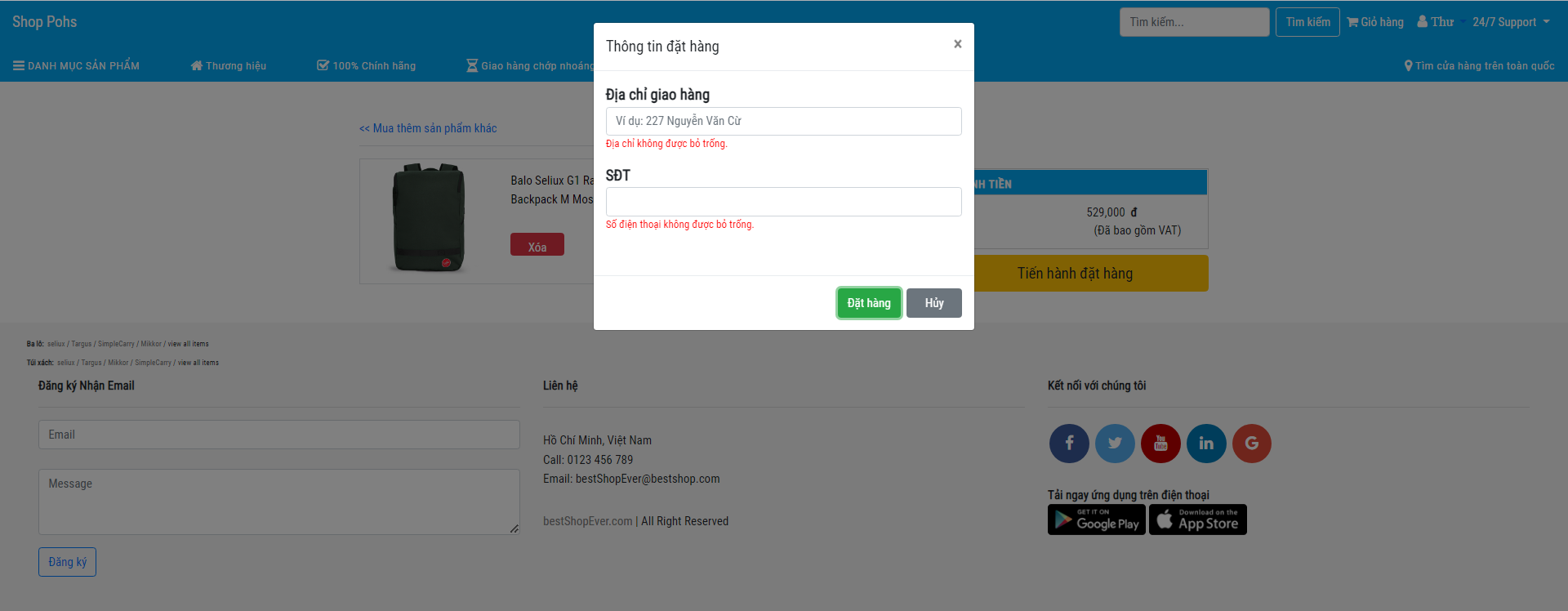
Hình 13 Màn hình thông tin tài khoản



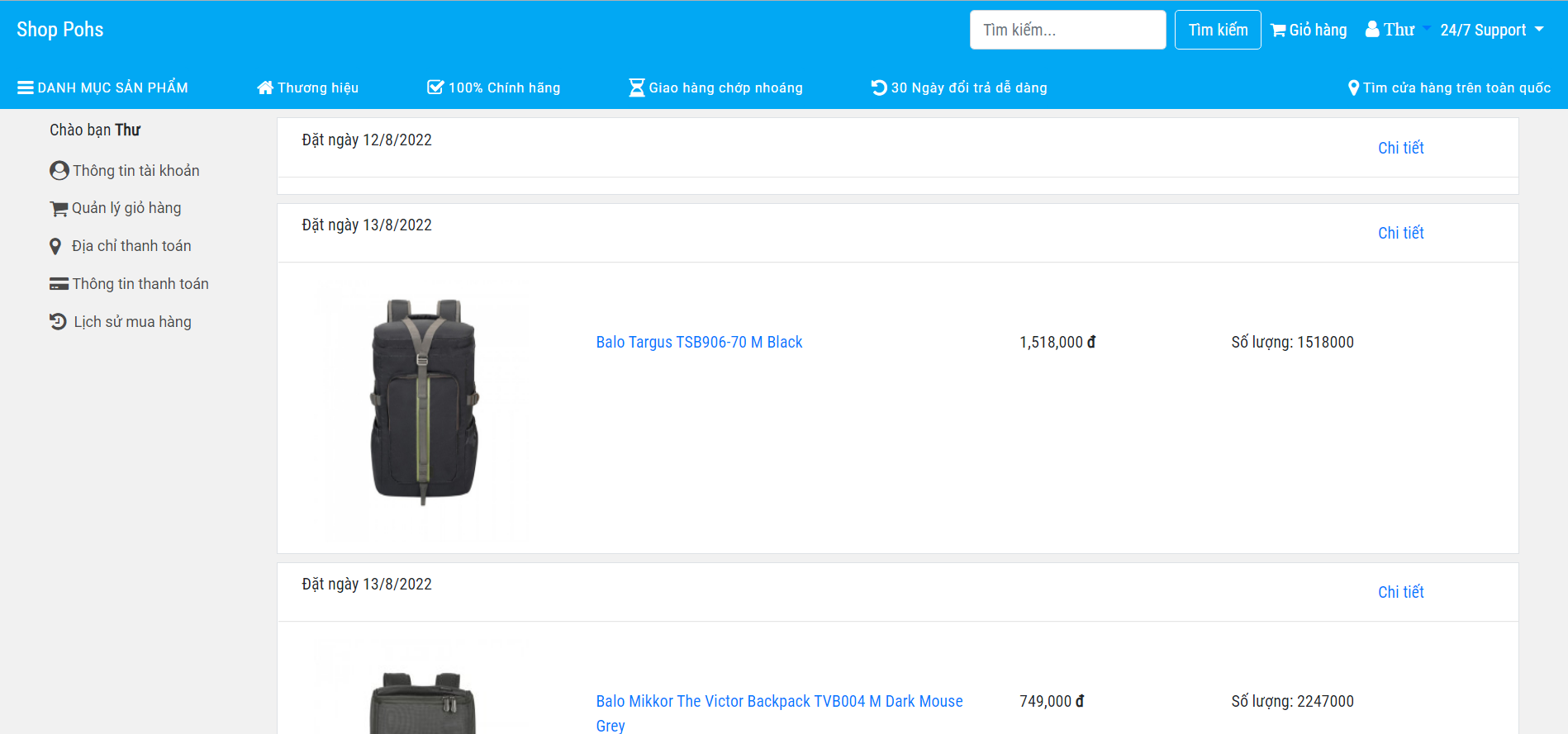
Hình 14 Màn hình giỏ hàng



Hình 15 Màn hình Đặt hàng với giỏ hàng trống



Hình 16 Màn hình Đặt hàng nếu bỏ trống số điện thoại và địa chỉ



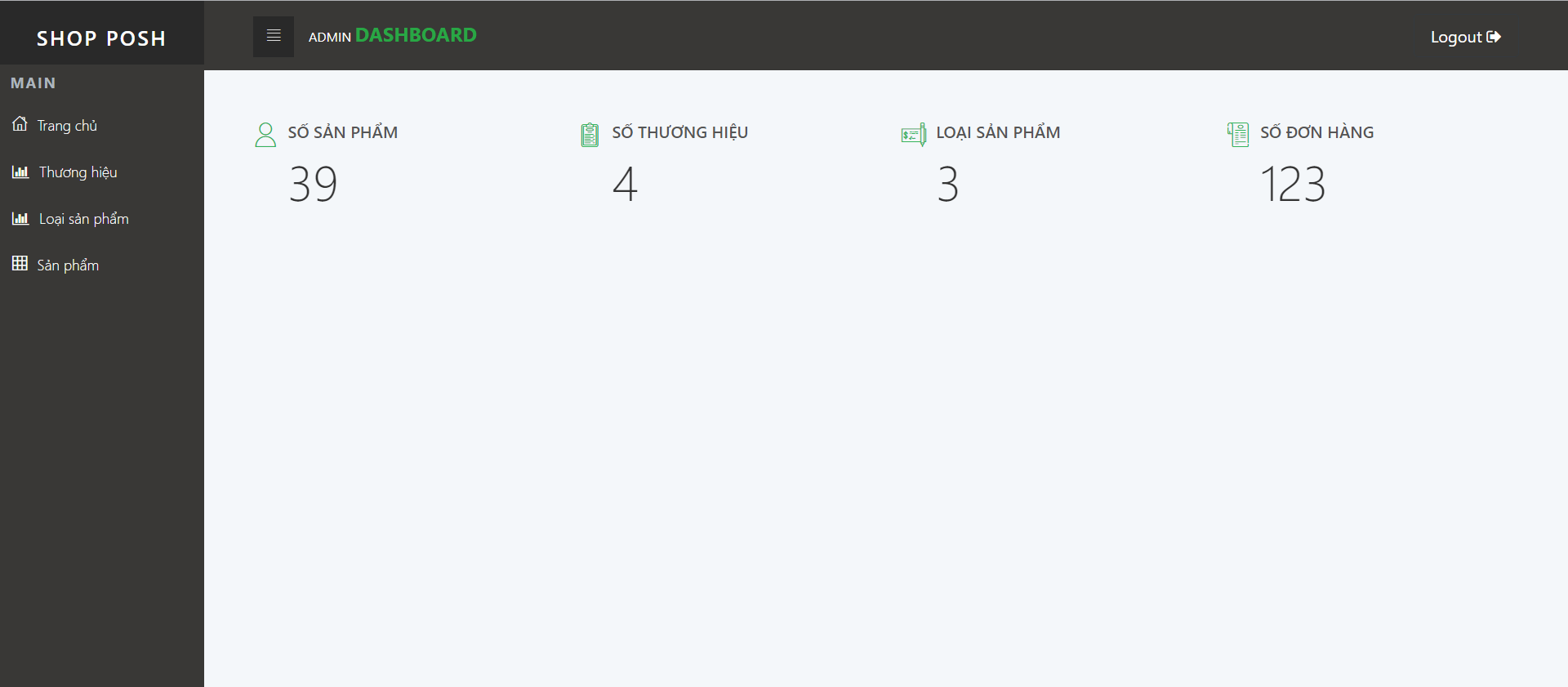
*Hình 17 Màn hình Lịch sử mua hàng*

### 

### **4.3.2 Giao diện admin**

**4.3.2a Dashboard**

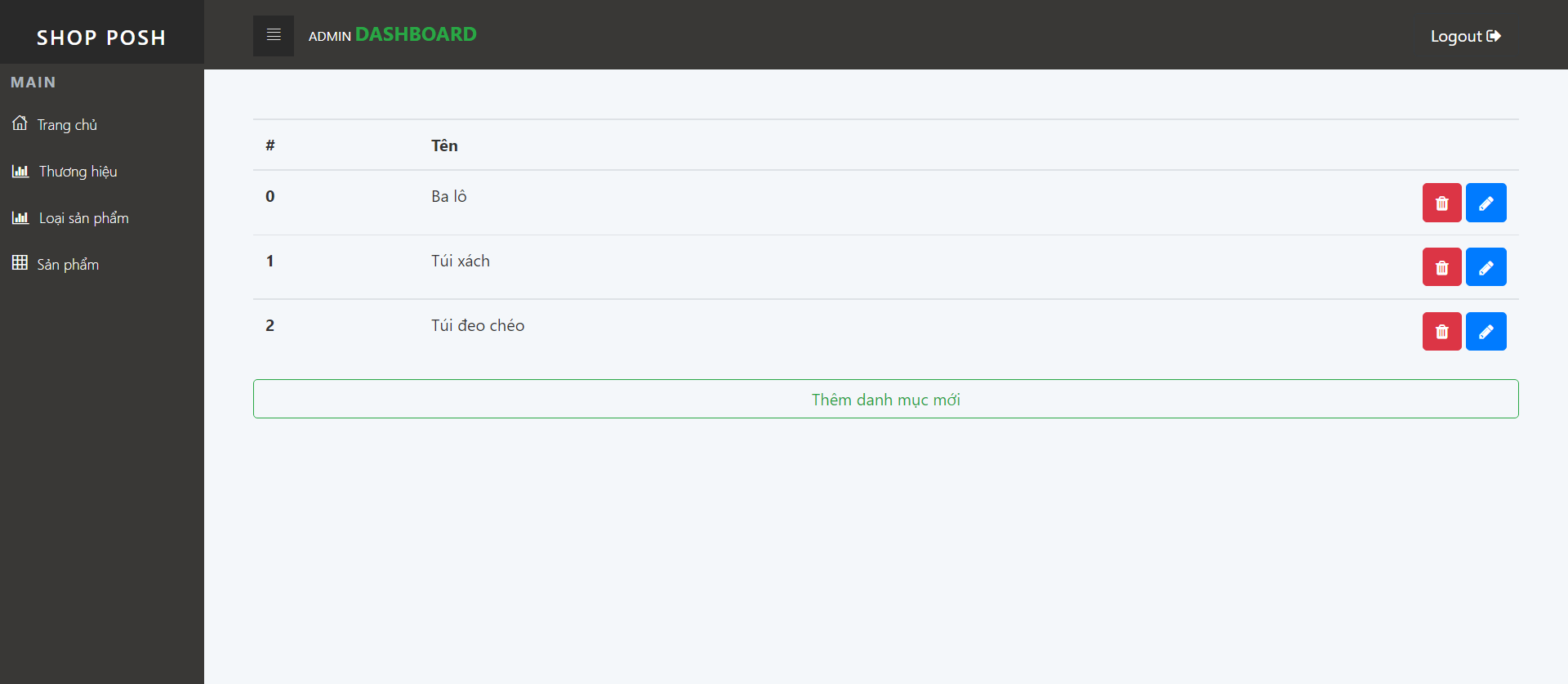
- Hiển thị số lượng danh mục sản phẩm, thương hiệu, sản phẩm và số đơn hàng.



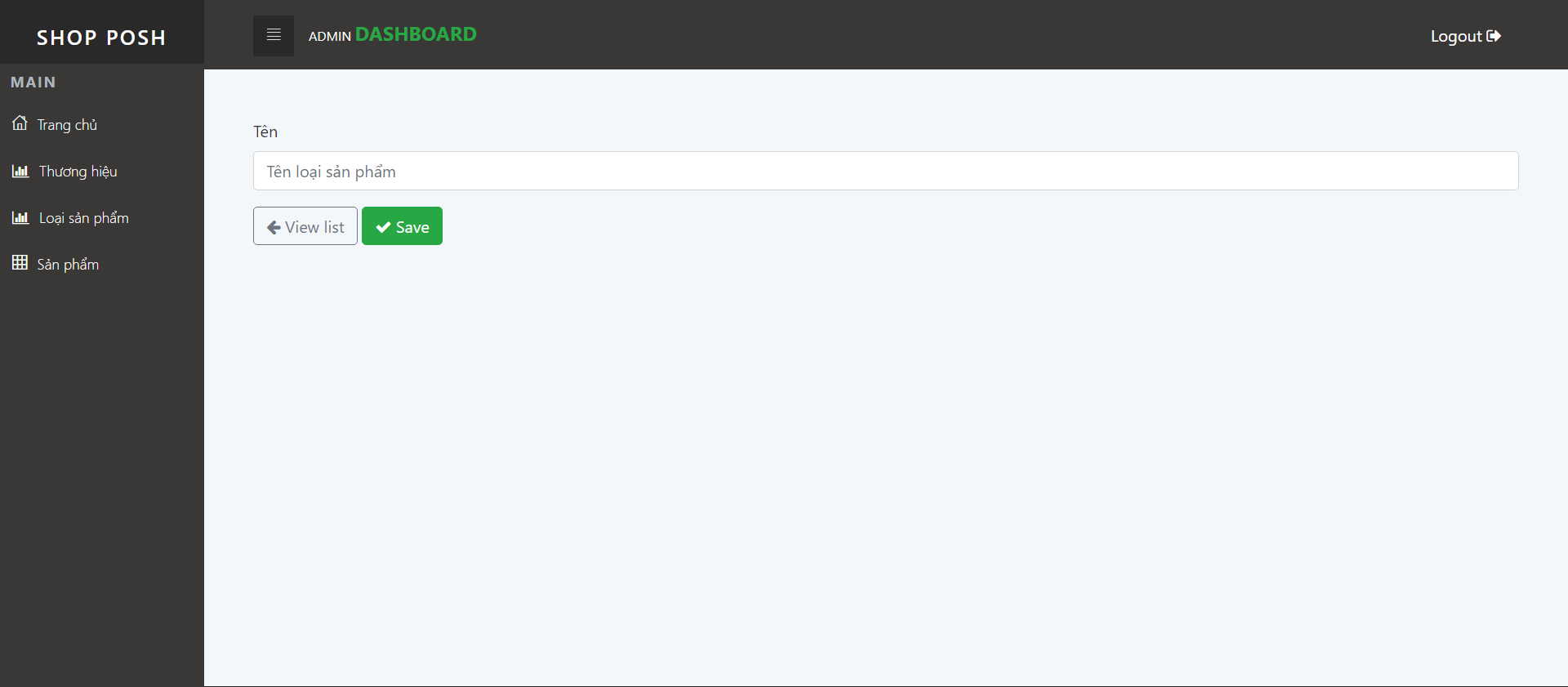
Hình 18 Màn hình Dashboard

**4.3.2b Quản lý danh mục sản phẩm**

- Quản lý danh mục sản phẩm: thêm, xóa và chỉnh sửa danh mục sản phẩm.



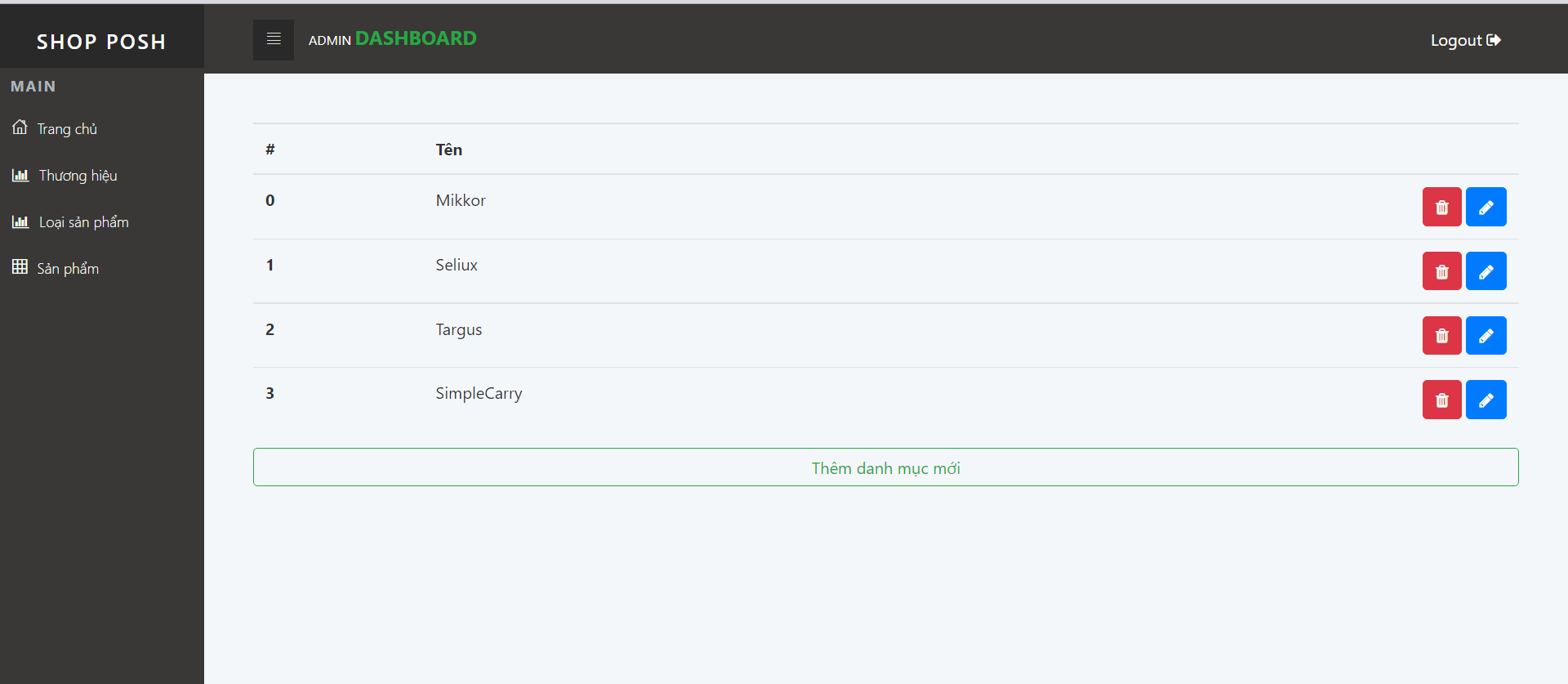
Hình 19 Màn hình Quản lý danh mục sản phẩm



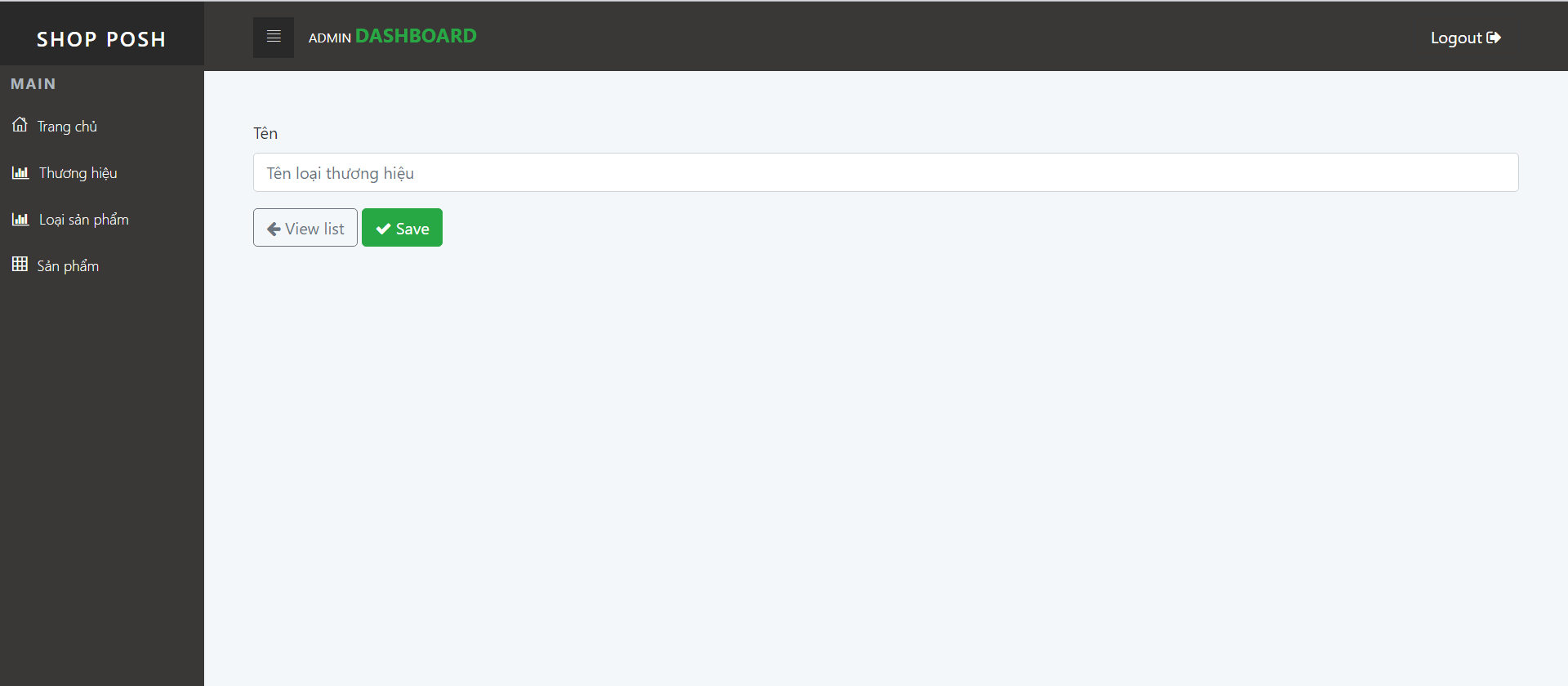
Hình 20 Màn hình thêm danh mục sản phẩm

**4.3.2c Quản lý thương hiệu**

- Quản lý thương hiệu: thêm, xóa và chỉnh sửa thương hiệu.



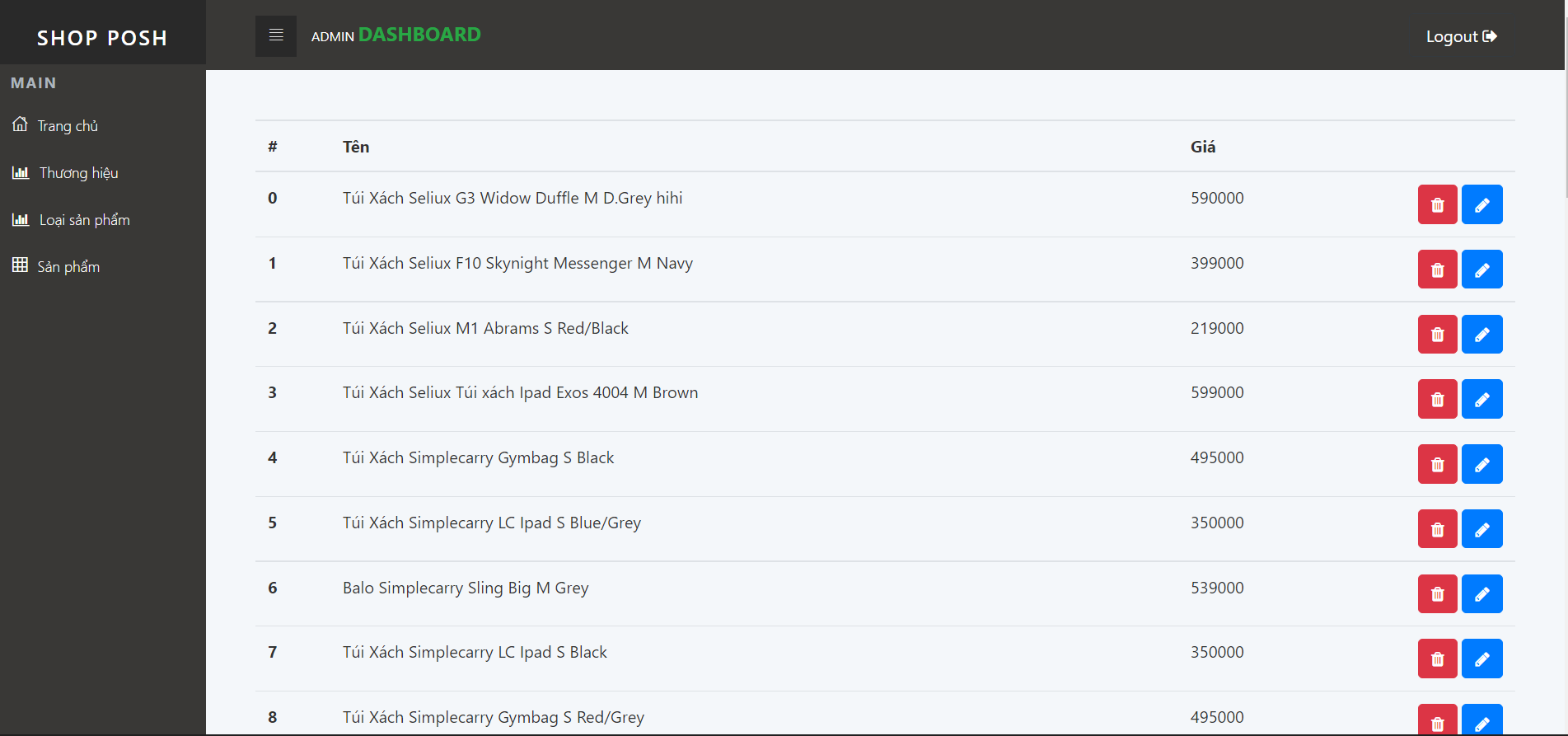
Hình 21 Màn hình Quản lý thương hiệu



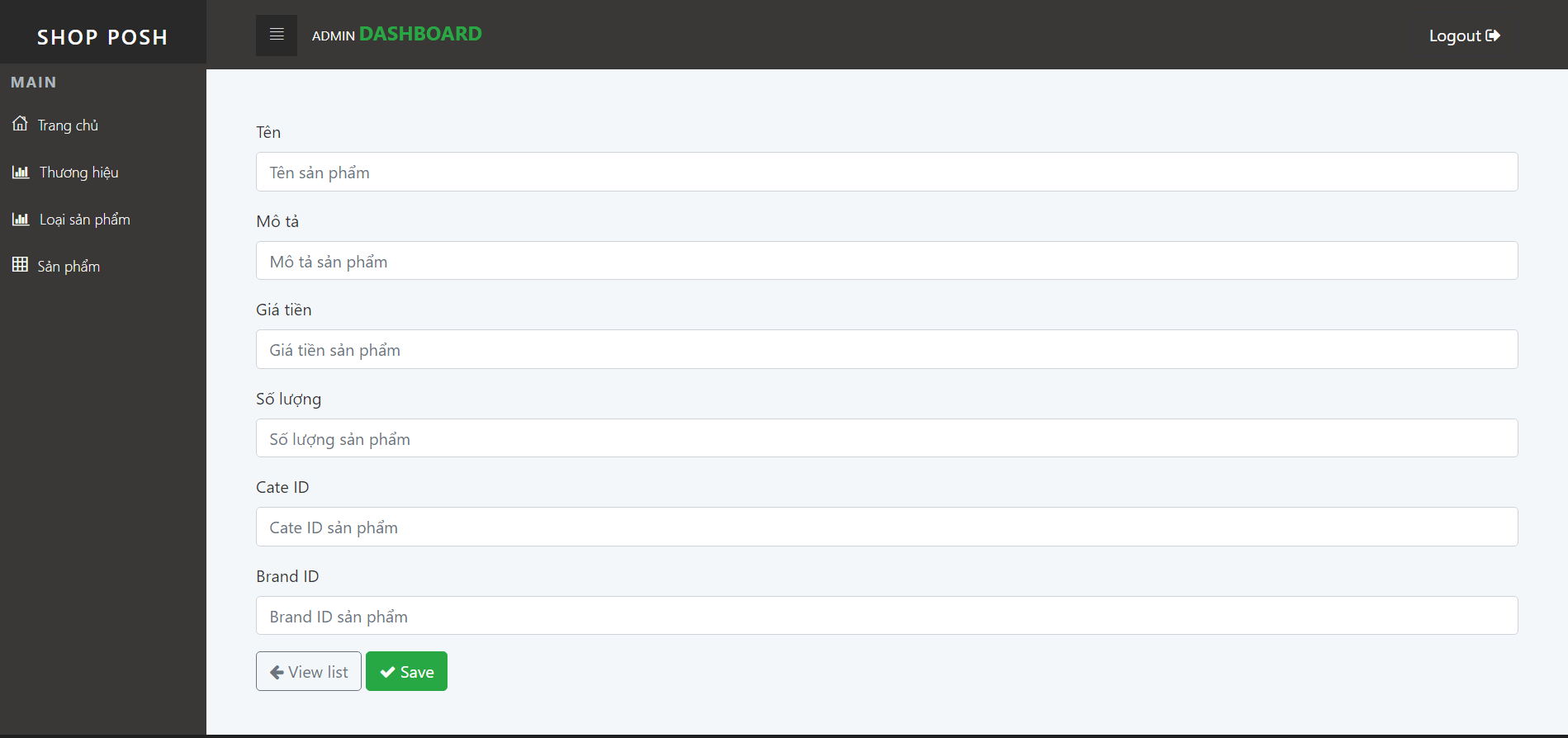
Hình 22 Màn hình thêm thương hiệu

**4.3.2d Quản lý sản phẩm**

- Quản lý sản phẩm: Thêm, xóa, chỉnh sửa sản phẩm



Hình 23 Màn hình Quản lý sản phẩm



Hình 24 Màn hình Thêm sản phẩm

# **CHƯƠNG 5. XÂY DỰNG CÔNG CỤ KIỂM THỬ TỰ ĐỘNG**

## **5.1 Xây dựng các test case để thực thi test script**

### **5.1.1 Đăng ký**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC ID | TC Description | Test Data | Test steps | Expected Result | Type |
| ĐK-01 | Xác minh không thể đăng ký tài khoản với giá trị rỗng. |  | 1. Mở trang web <http://localhost:3000/>  2. Chọn “Đăng ký”  3. Bỏ trống các trường giá trị.  4. Chọn nút “Đăng ký” | Hiển thị các thông báo lỗi “Không được bỏ trống” của từng field và không đăng ký tài khoản được. | Negative |
| ĐK-02 | Xác minh không thể đăng ký tài khoản với mật khẩu ít hơn 6 ký tự. | Mật khẩu: 12345 | 1. Mở trang web <http://localhost:3000/>  2. Chọn “Đăng ký”  3. Nhập vào ô “Mật khẩu” với giá trị ít hơn 6 ký tự. (Vd: 12345).  4. Điền các giá trị hợp lệ vào các ô còn lại.  5. Chọn nút “Đăng ký”. | Hiển thị thông báo lỗi “Mật khẩu phải lớn hơn 6 ký tự”. | Negative |
| ĐK-03 | Xác minh không thể đăng ký tài khoản với xác nhận mật khẩu khác mật khẩu đã đăng ký. | Mật khẩu: 123456  Xác nhận mật khẩu: 1234567 | 1. Mở trang web <http://localhost:3000/>  2. Chọn “Đăng ký”  3. Điền giá trị vào ô “Xác nhận mật khẩu” khác với giá trị mật khẩu.  4. Điền các giá trị hợp lệ vào các ô còn lại.  5. Chọn nút “Đăng ký”. | Hiển thị thông báo lỗi “Xác nhận mật khẩu không đúng. Mời bạn nhập lại mật khẩu.” | Negative |
| ĐK-04 | Xác minh rằng không thể đăng ký tài khoản với email không hợp lệ. | Email: 123@123 | 1. Mở trang web <http://localhost:3000/>  2. Chọn “Đăng ký”  3. Nhập email không hợp lệ vào ô “Email” (123@123).  4. Điền các giá trị hợp lệ vào các ô còn lại.  5. Chọn nút “Đăng ký”. | Hiển thị thông báo lỗi “Email không hợp lệ”. | Negative |
| ĐK-04 | Xác minh rằng đăng ký tài khoản thành công với các giá trị hợp lệ. | Tên đăng nhập: thuuuu046  Mật khẩu: 123456  Tên: Trần Thư  Email: thutran@  gmail.com | 1. Mở trang web <http://localhost:3000/>  2. Chọn “Đăng ký  3. Điền các giá trị hợp lệ vào các ô.  4. Chọn nút “Đăng ký”. | Hiển thị thông báo thành công: “Bạn đã đăng ký tài khoản thành công!”. | Positive |

Bảng 13 Test case Đăng ký

### **5.1.2 Đăng nhập**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC ID | TC Description | Pre-Conditions | Test steps | Expected Result | Type |
| ĐN-01 | Xác minh rằng không thể đăng nhập với tên đăng nhập trống |  | 1. Mở trang web <http://localhost:3000/>  2. Chọn “Đăng nhập”  3. Bỏ trống tên đăng nhập.  4. Nhập giá trị vào ô “Mật khẩu”. (123456)  5. Chọn nút “Đăng nhập” | Hiển thị alert với thông báo “Mời bạn nhập tên đăng nhập.” | Negative |
| ĐN-02 | Xác minh rằng không thể đăng nhập với mật khẩu trống. | Tên đăng nhập “thutran” đã được đăng ký. | 1. Mở trang web <http://localhost:3000/>  2. Chọn “Đăng nhập”  3. Nhập giá trị vào ô “Tên đăng nhập”. (thutran)  4. Bỏ trống mật khẩu.  5. Chọn nút “Đăng nhập” | Hiển thị alert với thông báo “Mời bạn nhập mật khẩu.” | Negative |
| ĐN-03 | Xác minh rằng không thể đăng nhập với thông tin không hợp lệ. |  | 1. Mở trang web <http://localhost:3000/>  2. Chọn “Đăng nhập”  3. Nhập giá trị không đúng vào ô “Tên đăng nhập” (abcd) và “Mật khẩu”.(12345)  4. Chọn nút “Đăng nhập” | Hiển thị thông báo “Sai tên đăng nhập hoặc mật khẩu.” | Negative |
| ĐN-04 | Xác minh rằng đăng nhập thành công với thông tin hợp lệ. | Đăng ký tài khoản thành công với tên đăng nhập (thutran) và mật khẩu (123456) | 1. Mở trang web <http://localhost:3000/>  2. Chọn “Đăng nhập”  3. Nhập giá trị đúng vào ô “Tên đăng nhập” và “Mật khẩu”.  4. Chọn nút “Đăng nhập” | Đăng nhập thành công. Hiển thị icon tên của khách hàng trên thanh menu. | Positive |

Bảng 14 Test case Đăng nhập

### **5.1.3 Tài khoản người dùng**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC ID | TC Description | Pre-Condition | Test steps | Expected Result | Type |
| TKND-01 | Xác minh rằng các thông tin hiển thị ở Profile trùng với thông tin đăng ký tài khoản. | 1. Mở trang web <http://localhost:3000/>  2. Đăng nhập thành công với tài khoản hợp lệ. (thutran,  123456) | 1. Chọn ‘Profile’ ở tên người dùng. | Các thông tin tên đăng nhập, họ tên, ngày sinh, email hiển thị ở ‘Thông tin tài khoản’ trùng với thông tin đăng ký. | Positive |
| TKND-02 | Xác minh không thể thay đổi mật khẩu nếu mật khẩu bé hơn 6 ký tự | 1. Mở trang web <http://localhost:3000/>  2. Đăng nhập thành công với tài khoản hợp lệ. (thutran,  123456) | 1. Chọn ‘Profile’ ở tên người dùng.  2. Chọn vào ô “Thay đổi mật khẩu” 3. Nhập mật khẩu bé hơn 6 ký tự vào ô “Mật khẩu mới” (123) | Không thể chọn nút “Cập nhật”. | Negative |
| TKND-03 | Xác minh không thể thay đổi mật khẩu nếu xác nhận mật khẩu khác với mật khẩu. | 1. Mở trang web <http://localhost:3000/>  2. Đăng nhập thành công với tài khoản hợp lệ. (thutran,  123456) | 1. Chọn ‘Profile’ ở tên người dùng.  2. Chọn vào ô “Thay đổi mật khẩu” 3. Nhập mật khẩu lớn hơn 6 ký tự vào ô “Mật khẩu mới” (1234567)  4. Nhập xác nhận mật khẩu khác với mật khẩu đã nhập ở trên. (12345677) | Không thể chọn nút “Cập nhật”. | Negative |
| TKND-04 | Xác minh có thể thay đổi mật khẩu thành công với giá trị hợp lệ. | 1. Mở trang web <http://localhost:3000/>  2. Đăng nhập thành công với tài khoản hợp lệ. (thutran,  123456) | 1. Chọn ‘Profile’ ở tên người dùng.  2. Chọn vào ô “Thay đổi mật khẩu” 3. Nhập mật khẩu hợp vào ô “Mật khẩu mới” và “Xác nhận mật khẩu”. (123456789) | Thay đổi mật khẩu thành công. Đăng nhập tài khoản này bằng mật khẩu mới thành công. | Negative |
| TKND-05 | Xác minh không thể thay đổi tên đăng nhập. | 1. Mở trang web <http://localhost:3000/>  2. Đăng nhập thành công với tài khoản hợp lệ. (thutran,  123456) | 1. Chọn ‘Profile’ ở tên người dùng. | Ô “Tên đăng nhập” bị khóa, không thể thêm ký tự nào vào ô này. | Negative |
| TKND-06 | Xác minh có thể thay đổi thông tin người dùng thành công với thông tin hợp lệ. | 1. Mở trang web <http://localhost:3000/>  2. Đăng nhập thành công với tài khoản hợp lệ. (thutran,  123456) | 1. Chọn ‘Profile’ ở tên người dùng.  2. Nhập các giá trị muốn chỉnh sửa vào ô tương ứng. | Cập nhật thông tin thành công. | Positive |

Bảng 15 Test case Tài khoản người dùng

### **5.1.4 Tìm kiếm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TC ID | TC Description | Test steps | Expected Result | Type |
| TK-01 | Xác minh rằng hiển thị thông báo lỗi khi tìm kiếm với thông tin trống. | 1. Mở trang web <http://localhost:3000/>  2. Chọn nút “Tìm kiếm” trên menu đầu trang web. | Hiển thị thông báo “Không có sản phẩm muốn tìm” | Negative |
| TK-02 | Xác minh rằng không hiển thị thông tin sản phẩm khi tìm kiếm với thông tin không có trong hệ thống. | 1. Mở trang web <http://localhost:3000/>  2. Nhập giá trị không có thông tin vào ô “Tìm kiếm” (Vd: aaaaaaaaaa”)  3. Chọn nút “Tìm kiếm”trên menu đầu trang web. | Không hiển thị bất cứ sản phẩm nào và hiển thị thông báo “Không có sản phẩm thỏa yêu cầu” | Negative |
| TK-03 | Xác minh rằng hiển thị tất cả các sản phẩm liên quan khi tìm kiếm với giá trị tương đối. | 1. Mở trang web <http://localhost:3000/>  2. Nhập giá trị tương đối vào ô “Tìm kiếm” (Vd: “túi xách”)  3. Chọn nút “Tìm kiếm”trên menu đầu trang web. | Hiển thị tất cả các sản phẩm liên quan đến túi xách. | Positive |
| TK-04 | Xác minh rằng hiển thị sản phẩm đúng khi tìm kiếm bằng giá trị tuyệt đối. | 1. Mở trang web <http://localhost:3000/> 2. Nhập giá trị tuyệt đối vào ô “Tìm kiếm” (Vd: “[Túi Xách Targus TBT268AP-72 M Black](http://localhost:3000/product/detail/20) ”).  3. Chọn nút “Tìm kiếm”trên menu đầu trang web. | Hiển thị sản phẩm “[Túi Xách Targus TBT268AP-72 M Black](http://localhost:3000/product/detail/20) ” | Positive |
| TK-05 | Xác minh rằng có thể lọc sản phẩm hiển thị theo danh mục | 1. Mở trang web <http://localhost:3000/>  2. Chọn “Danh mục”.  3. Chọn “Ba lô”. | Hiển thị tất cả sản phẩm thuộc danh mục “Ba lô” | Positive |
| TK-06 | Xác minh rằng có thể lọc sản phẩm hiển thị theo thương hiệu. | 1. Mở trang web <http://localhost:3000/>  2. Chọn “Thương hiệu”.  3. Chọn “Mikkor”. | Hiển thị tất cả sản phẩm thuộc thương hiệu “Mikkor” | Positive |

Bảng 16 Test case Tìm kiếm

### **5.1.5 Đặt hàng**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC ID | TC Description | Pre-Conditions | Test steps | Expected Result | Type |
| ĐH-01 | Xác minh rằng có thể thêm bất kỳ sản phẩm nào vào giỏ hàng. | 1. Mở trang web <http://localhost:3000/>  2. Đăng nhập thành công với tài khoản hợp lệ. (thutran và 123456) | 1. Chọn sản phẩm muốn thêm vào giỏ hàng. (“[Balo Targus TSB855-70 S Black"](http://localhost:3000/search?q=%22Balo%20Targus%20TSB855-70%20S%20Black%22))  2. Chọn nút “Mua ngay”. | Xác nhận sản phẩm được thêm vào giỏ hàng thành công với tên sản phẩm, giá tiền và số lượng hợp lệ. | Positive |
| ĐH-02 | Xác minh rằng có thể thêm hoặc giảm số lượng sản phẩm. | 1. Mở trang web <http://localhost:3000/>  2. Đăng nhập thành công với tài khoản hợp lệ. (thutran và 123456) | 1. Chọn sản phẩm muốn thêm vào giỏ hàng. (“[Balo Targus TSB855-70 S Black"](http://localhost:3000/search?q=%22Balo%20Targus%20TSB855-70%20S%20Black%22))  2. Chọn nút “Mua ngay”.  3. Chọn icon “+” nếu muốn tăng số lượng, icon “-” nếu muốn giảm số lượng. | Tăng giảm số lượng thành công, số tiền và số lượng được cập nhật đúng. | Positive |
| ĐH-03 | Xác minh rằng có thể xóa sản phẩm khỏi giỏ hàng. | 1. Mở trang web <http://localhost:3000/>  2. Đăng nhập thành công với tài khoản hợp lệ. (thutran và 123456) | 1. Thêm sản phẩm vào giỏ hàng. (“[Balo Targus TSB855-70 S Black"](http://localhost:3000/search?q=%22Balo%20Targus%20TSB855-70%20S%20Black%22))  2. Chọn nút “Xóa”. | Xóa sản phẩm ra khỏi giỏ hàng thành công. | Positive |
| ĐH-04 | Xác minh rằng có thể thêm nhiều sản phẩm vào giỏ hàng. | 1. Mở trang web <http://localhost:3000/>  2. Đăng nhập thành công với tài khoản hợp lệ. (thutran và 123456) | 1. Thêm sản phẩm vào giỏ hàng. (“[Balo Targus TSB855-70 S Black"](http://localhost:3000/search?q=%22Balo%20Targus%20TSB855-70%20S%20Black%22)) 2. Chọn sản phẩm khác thêm vào giỏ hàng. (Túi Xách Simplecarry Gymbag S Black) | Tất cả sản phẩm được chọn đều thêm vào giỏ hàng thành công. Hiển thị thông tin sản phẩm đúng. | Positive |
| ĐH-05 | Xác minh rằng không thể đặt hàng nếu giỏ hàng trống. | 1. Mở trang web <http://localhost:3000/>  2. Đăng nhập thành công với tài khoản hợp lệ. (thutran và 123456) | 1. Chọn “Giỏ hàng”.  2. Xóa tất cả sản phẩm nếu có trong giỏ hàng.  3. Chọn “Tiến hành đặt hàng”. | Hiện thị thông báo “Giỏ hàng trống. Không thể đặt hàng.” | Negative |
| ĐH-06 | Xác minh rằng không thể đặt hàng nếu để trống địa chỉ hoặc số điện thoại. | 1. Mở trang web <http://localhost:3000/>  2. Đăng nhập thành công với tài khoản hợp lệ.  (thutran và 123456) | 1.Thêm sản phẩm vào giỏ hàng. (“[Balo Targus TSB855-70 S Black"](http://localhost:3000/search?q=%22Balo%20Targus%20TSB855-70%20S%20Black%22))  2. Chọn “Tiến hành đặt hàng”.  3. Chọn “Đặt hàng” ở popup. | Hiển hị thông báo “Số điện thoại không được để trống.” và “Địa chỉ không được để trống.” | Negative |
| ĐH-07 | Xác minh rằng có thể đặt hàng thành công với thông tin hợp lệ. | 1. Mở trang web <http://localhost:3000/>  2. Đăng nhập thành công với tài khoản hợp lệ. | 1.Thêm sản phẩm vào giỏ hàng. (“[Balo Targus TSB855-70 S Black"](http://localhost:3000/search?q=%22Balo%20Targus%20TSB855-70%20S%20Black%22))  2. Chọn “Tiến hành đặt hàng”.  3. Nhập thông tin hợp lệ vào “Số điện thoại” (0987654321) và ô “Địa chỉ” (Man Thiện, Quận 9)  4. Chọn “Đặt hàng” ở popup. | Đặt hàng thành công. Đơn đặt hàng sẽ hiển thị trong lịch sử mua hàng của tài khoản người dùng. | Positive |

Bảng 17 Test case Đặt hàng

**5.4.6 Quản lý danh mục sản phẩm**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC ID | TC Description | Pre-Conditions | Test steps | Expected Result | Type |
| DM-01 | Xác minh rằng có thể thêm danh mục sản phẩm thành công với giá trị hợp lệ | Đăng nhập tài khoản admin thành công. (admin và admin) | 1. Mở trang web <http://localhost:3000/admin/>  2. Chọn “Loại sản phẩm” tại menu bar bên trái.  3. Chọn “Thêm loại sản phẩm”.  4. Nhập tên loại sản phẩm vào ô “Tên”. (“Túi đeo chéo”  5. Chọn “Save”. | Hiển thị thông báo thành công “Thêm thành công” sau khi thêm danh mục sản phẩm thành công. | Positive |
| DM-02 | Xác minh rằng có thể chỉnh sửa danh mục bất kỳ thành công. | Đăng nhập tài khoản admin thành công. (admin và admin) | 1. Mở trang web <http://localhost:3000/admin/>  2. Chọn “Loại sản phẩm” tại menu bar bên trái.  3. Chọn icon “Chỉnh sửa” đối với danh mục muốn chỉnh sửa. (“Túi đeo chéo”)  4. Chỉnh sửa thông tin cần thiết. (“Túi đeo chéo 1”)  5. Chọn “Update”. | Chỉnh sửa danh mục sản phẩm thành công. | Positive |
| DM-03 | Xác minh rằng có thể xóa danh mục sản phẩm thành công. | Đăng nhập tài khoản admin thành công. (admin và admin) | 1. Mở trang web <http://localhost:3000/admin/>  2. Chọn “Loại sản phẩm” tại menu bar bên trái.  3. Chọn icon “Xóa” đối với danh mục muốn xóa. (“Túi đeo chéo 1”).  4. Chọn “Delete”. | Xóa danh mục sản phẩm thành công. | Positive |

Bảng 18 Test case Quản lý danh mục sản phẩm

**5.4.7 Quản lý thương hiệu**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC ID | TC Description | Pre-Conditions | Test steps | Expected Result | Type |
| TH-01 | Xác minh rằng có thể thêm thương hiệu sản phẩm thành công với giá trị hợp lệ | Đăng nhập tài khoản admin thành công. (admin và admin) | 1. Mở trang web <http://localhost:3000/admin/>  2. Chọn “Thương hiệu” tại menu bar bên trái.  3. Chọn “Thêm thương hiệu”.  4. Nhập tên thương hiệu sản phẩm vào ô “Tên”. (“Gucci”)  5. Chọn “Save”. | Hiển thị thông báo thành công “Thêm thành công”. | Positive |
| TH-02 | Xác minh rằng có thể chỉnh sửa thương hiệu sản phẩm bất kỳ thành công. | Đăng nhập tài khoản admin thành công. (admin và admin) | 1. Mở trang web <http://localhost:3000/admin/>  2. Chọn “Thương hiệu” tại menu bar bên trái.  3. Chọn icon “Chỉnh sửa” đối với thương hiệu muốn chỉnh sửa. (“Gucci”)  4. Chỉnh sửa thông tin cần thiết. (“Channel”)  5. Chọn “Update”. | Chỉnh sửa thương hiệu sản phẩm thành công. | Positive |
| TH-03 | Xác minh rằng có thể xóa thương hiệu sản phẩm thành công. | Đăng nhập tài khoản admin thành công. (admin và admin) | 1. Mở trang web <http://localhost:3000/admin/>  2. Chọn “Thương hiệu” tại menu bar bên trái.  3. Chọn icon “Xóa” đối với thương hiệu muốn xóa. (“Channel”)  4. Chọn “Delete”. | Xóa thương hiệu sản phẩm thành công. | Positive |

Bảng 19 Test case Quản lý thương hiệu sản phẩm

**5.4.8 Quản lý sản phẩm**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC ID | TC Description | Pre-Conditions | Test steps | Expected Result | Type |
| TH-01 | Xác minh rằng có thể thêm sản phẩm thành công với giá trị hợp lệ | Đăng nhập tài khoản admin thành công. (admin và admin) | 1. Mở trang web <http://localhost:3000/admin/>  2. Chọn “Sản phẩm” tại menu bar bên trái.  3. Chọn “Thêm sản phẩm”.  4. Nhập các giá trị hợp lệ vào các ô tương ứng. (“Balo”)  5. Chọn “Save”. | Hiển thị thông báo thành công “Thêm thành công” sau khi thêm sản phẩm thành công. | Positive |
| TH-02 | Xác minh rằng có thể chỉnh sửa thương hiệu sản phẩm bất kỳ thành công. | Đăng nhập tài khoản admin thành công. (admin và admin) | 1. Mở trang web <http://localhost:3000/admin/>  2. Chọn “Sản phẩm” tại menu bar bên trái.  3. Chọn icon “Chỉnh sửa” đối với sản phẩm muốn chỉnh sửa. (“Balo”)  4. Chỉnh sửa thông tin cần thiết. (“Balo123”)  5. Chọn “Update”. | Chỉnh sửa sản phẩm thành công. | Positive |
| TH-03 | Xác minh rằng có thể xóa sản phẩm thành công. | Đăng nhập tài khoản admin thành công. (admin và admin) | 1. Mở trang web <http://localhost:3000/admin/>  2. Chọn “Sản phẩm” tại menu bar bên trái.  3. Chọn icon “Xóa” đối với sản phẩm muốn xóa. (“Balo123”)  4. Chỉnh sửa thông tin cần thiết.  5. Chọn “Delete”. | Xóa sản phẩm thành công. | Positive |

Bảng 20 Test case Quản lý sản phẩm

## **5.2 Cài đặt phần mềm**

Phần mềm sẽ được thực thi trên công cụ Visual Studio Code, cùng với ngôn ngữ Nodejs.

Để tạo được phần mềm cơ bản, ta phải tiến hành cài đặt các công cụ cần thiết:

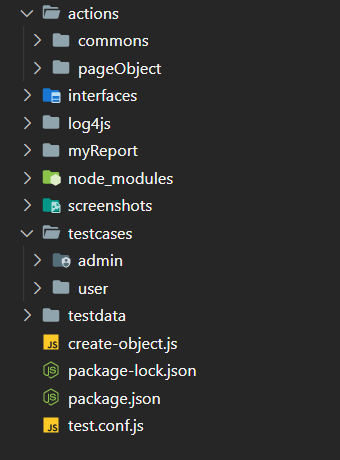
* Selenium Webdriver
* Chrome Driver
* Nodejs
* Mocha Testing Framework
* Chai Framework
* Mochawesome Reporter

## **5.3 Thực hành thiết kế**

### **5.3.1 Cấu trúc của project để định nghĩa các test scripts**

Project được xây dựng dựa trên cấu trúc Pattern Object Modal (POM):

* Actions: Thư mục chứa các hàm dùng chính của dự án.
* Commons: Thư mục chứa các hàm dùng chung cho cả dự án.
* PageObject: Thư mục chứa các hàm chức năng của từng module.
* Interfaces: Thư mục chứa tất cả locator của trang web.
* MyReport: Thư mục chứa báo cáo sau khi thực thi các test case.
* Screenshots: Thư mục chứa các hình ảnh chụp màn hình nếu chạy test case fail.
* Testcases: Thư mục chứa các test script của test case
* Testdata: Thư mục chứa dữ liệu khi thực thi test script.
* Package.json: File cài đặt dự án



Hình 25 Cấu trúc của project

### **5.3.2 Cấu trúc của test scrips**

Một test case cơ bản ngoài các bước để thực thi hành động, thì quan trọng nhất với người tester đó là phần xác minh. Đây là phần đảm bảo xem website đã được thực thi theo đúng tiêu chí mà tài liệu yêu cầu đưa ra hay chưa.

Cấu trúc một test case cơ bản được xây dựng dựa trên Mocha Testing Framework:

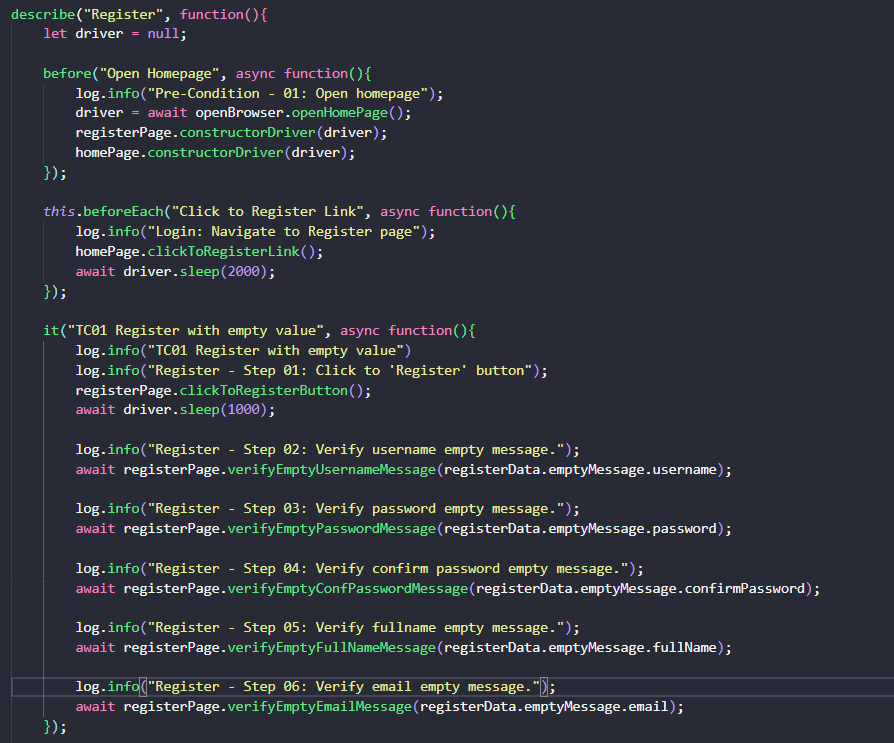
- describe()*:* Là một chức năng dùng để nhóm các test case có mối quan hệ mật thiết lại với nhau. Và khi gom các test case lại với nhau, thông thường ta gọi là Test Suite

- it(): Nội dung chính của test case sẽ được viết trong mục it(), mỗi it() là một test case và được xử lý tuần tự từ trên xuống nếu khai báo nhiều it() trong một describe()

- Hook: Mocha cung cấp một số interface với vai trò và thời điểm khởi chạy nhất định. Giúp phân bổ các bước trong kịch bản test phù hợp. Việc tuân thủ và áp dụng các hooks khi triển khai test case giúp cho chúng ta rất nhiều như dễ đọc, maintain dễ dàng và phát huy hiệu quả.

* before(): Chạy một lần duy nhất, trước khi test case (it) đầu tiên trong describe được khởi chạy. Thông thường hook này chứa những xử lý liên quan đến việc chuẩn bị một số điều kiện tiên quyết để các test case có thể chạy được. Ví dụ: Bật/tắt một số setting hệ thống, chuẩn bị các dữ liệu về người dùng…
* beforeEach(): Chạy trước khi mỗi khi test case (it) khởi chạy trong một describe.
* afterEach(): Tương tự beforeEach. Hook này chạy sau khi mỗi test case (it) được chạy trong một describe. Ví dụ: logout user ra khỏi hệ thống. Hook này thường sẽ chứa các bước để xóa dữ liệu phát sinh do test case.
* after(): Tương tự before. Hook này chạy một lần sau khi test case(it) cuối cùng trong describe được chạy. Đối với những dữ liệu phát sinh phục vụ chung cho toàn bộ các test case sẽ được làm sạch một lần bằng cách xử lý tại after hook.

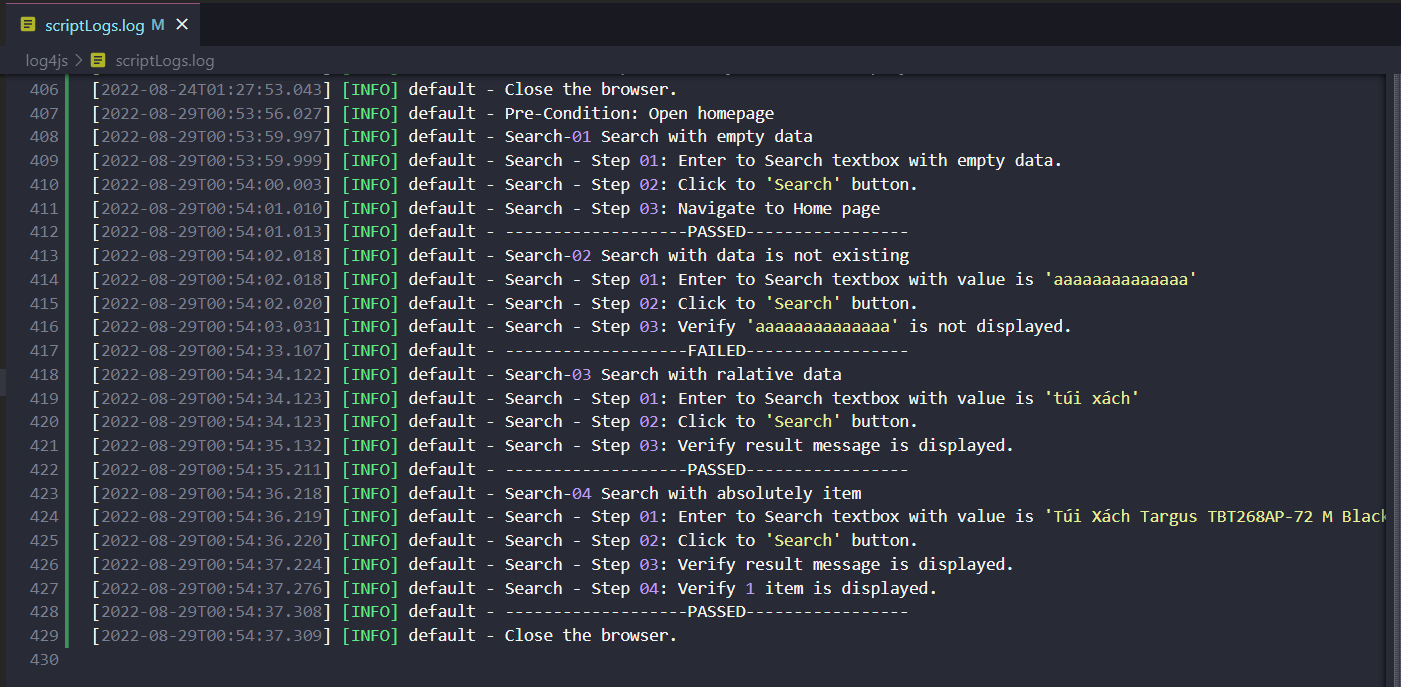
Mocha cũng cho phép cấu hình để có thể chạy nhiều test suite song song cùng một lúc.



Hình 26: Cấu trúc của test script

### **5.3.3 Ghi log cho từng test case vào file**

Sử dụng công cụ Log4js để ghi lại thời gian thực của kết quả test khi mỗi lần thực thi test case.



Hình 27 Màn hình file ghi log

### **5.3.4 Ghi vết kết quả mỗi lần chạy test case nếu fail**

Trong quá trình thực thi các test case, nếu testcase pass thì ta sẽ đóng quá trình thực thi. Nhưng với trường hợp test case bị fail, người kiểm thử phải biết testcase đó fail tại bước nào và cần có hình ảnh hỗ trợ giúp biết lỗi rõ ràng hơn.

Selenium Webdriver hỗ trợ hàm chức năng chụp màn hình, ta sẽ cấu hình mỗi lần test case bị fail thì chụp màn hình lại và hình ảnh sẽ được lưu vào thư mục trong dự án, đồng thời lưu vào báo cáo.

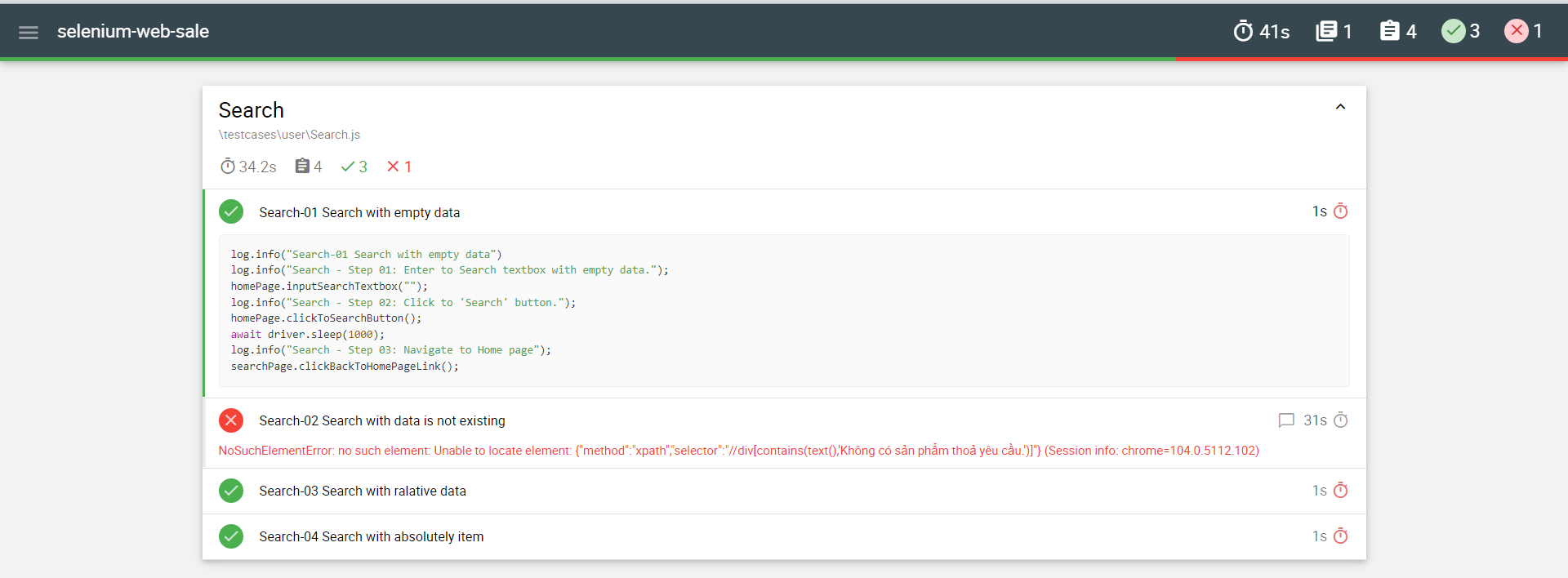


Hình 28 Hàm chụp ảnh màn hình nếu test case fail

### **5.3.5 Báo cáo kết quả thực thi test case**

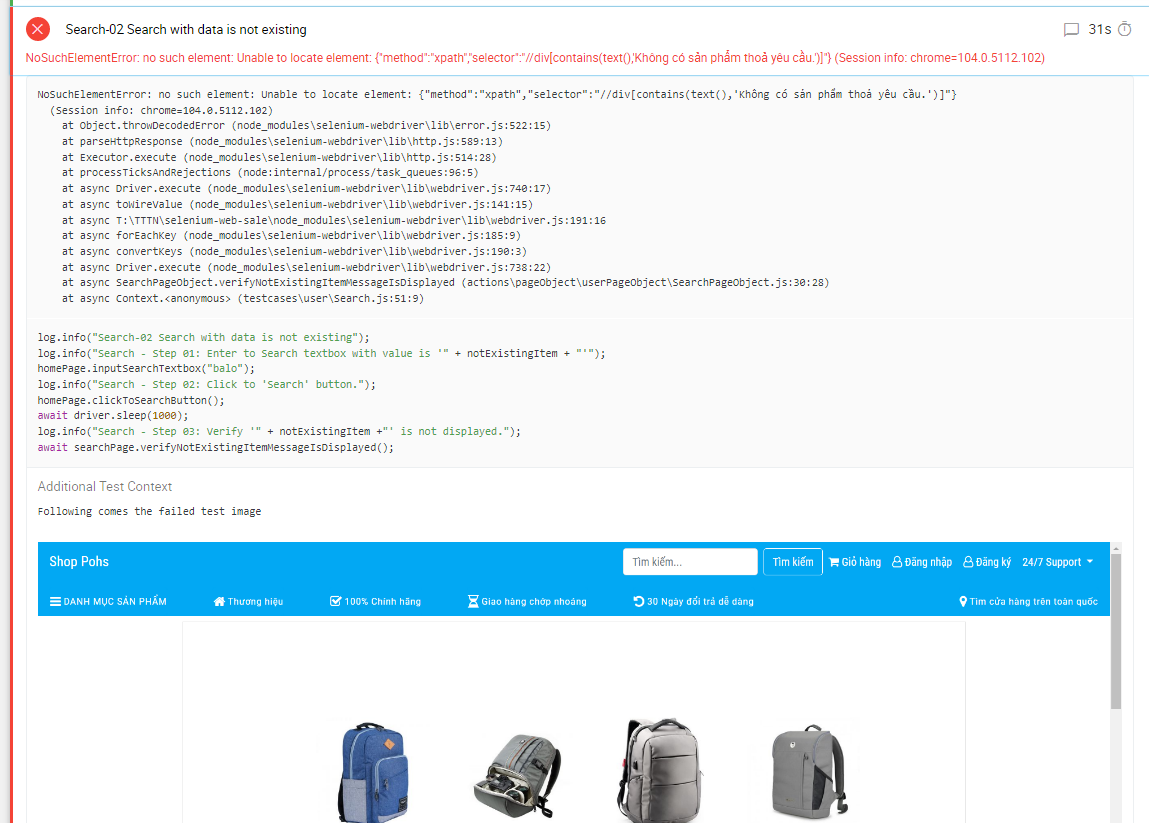
Mochawesome Report hỗ trợ tạo file báo cáo sau khi hoàn thành thực thi test case với 2 dạng file.json và file.html

Báo cáo ghi lại tên test suite, tổng số test case được thực thi, số test case pass và fail, cùng với tổng thời gian thực thi test suite này.



Hình 29 Màn hình báo cáo sau khi thực thi test case

Nếu test case bị fail, báo cáo sẽ chỉ ra lỗi bị fail cùng với hình ảnh chụp từ màn hình tại bước khiến tast case fail.



Hình 30 Màn hình báo cáo khi test case fail

# 

# **CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA PHẦN MỀM**

## **6.1 Kết luận**

### **6.1.1 Kết quả đạt được**

- Tìm hiểu lý thuyết về kiểm thử phần mềm thủ công và tự động.

- Tìm hiểu về My SQL và NodeJs để xây dựng website bán hàng có các chức năng cơ bản.

- Tìm hiểu về Selenium Webdriver và các công cụ cần thiết, từ đó xây dựng được công cụ kiểm thử tự động chức năng cho website.

- Tạo báo cáo kết quả thực thi testcase rõ ràng, dễ hiểu dành cho mọi người, kể cả những người không theo ngành kiểm thử phần mềm.

- Thực thi được nhiều test suite chạy song song cùng lúc.

### **6.1.2 Hạn chế**

- Website cò thiếu nhiều chức năng như quản lý đặt hàng, quản lý người dùng, theo dõi đơn hàng…

- Công cụ kiểm thử chỉ mới chạy được trên trình duyệt chrome, chưa chạy được trên các trình duyệt khác

- Chưa thể tích hợp CI/CD để đặt lịch chạy test case.

## **6.2 Hướng phát triển**

**-** Tích hợp CI/CD để đặt lịch chạy test case.

- Cấu hình project để có thể thực thi test case trên nhiều hệ điều hành, nhiều trình duyệt hay nhiều môi trường khác nhau.

- Sử dụng Selenium Grid để thực thi các test case theo chế độ phân tán.