

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ  
KHOA CÔNG NGHỆ THÔNG TIN



CHUYÊN ĐỀ ASP.NET  
HỌC KỲ 6, NĂM HỌC 2025

**Xây dựng website bán Laptop**

*Giáo viên hướng dẫn:*  
Họ tên: ThS. Đoàn Phước Miền

*Sinh viên thực hiện:*  
Họ tên: Nông Hồng Hải  
MSSV: 170123694  
Lớp: DK23TTG12

*Vĩnh Long, tháng 07 năm 2025.*

## **NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Vĩnh Long, ngày ..... tháng ..... năm .....*  
**Giáo viên hướng dẫn**  
(Ký tên và ghi rõ họ tên)

## **NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

Vĩnh Long, ngày ..... tháng ..... năm .....  
**Thành viên hội đồng**  
(Ký tên và ghi rõ họ tên)

## LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn chân thành đến tất cả những người đã đồng hành và hỗ trợ em trong suốt quá trình thực hiện đề tài này.

Em xin được tri ân Trường Đại học Trà Vinh cùng quý thầy cô Trường Kỹ Thuật Và Công Nghệ khoa Công Nghệ Thông Tin đã tạo điều kiện học tập, cung cấp kiến thức nền tảng quý giá, giúp em có đủ khả năng triển khai và hoàn thành đề tài với niềm đam mê và sự tự tin.

Em xin gửi lời cảm ơn chân thành đến thầy Th.S Đoàn Phước Miền đã tận tình hướng dẫn, chỉ bảo em trong suốt quá trình học tập và thực hiện đề tài.

Em cũng xin gửi lời cảm ơn đến QLHT Trần Thị Khánh Linh, người đã hỗ trợ và tạo điều kiện tốt nhất cho em trong suốt quá trình học.

Em cũng xin bày tỏ lòng biết ơn đến gia đình và bạn bè đã luôn động viên, khích lệ em không ngừng, đặc biệt trong những lúc khó khăn và mệt mỏi nhất.

Đề tài này là kết quả của một quá trình học hỏi không ngừng và em biết rằng những thành quả đạt được là nhờ vào sự góp phần của rất nhiều người. Tuy nhiên, em cũng ý thức rõ rằng đề tài vẫn còn những thiếu sót nhất định và em rất mong nhận được những góp ý chân thành từ quý thầy cô và các bạn để hoàn thiện hơn trong tương lai.

Một lần nữa, em xin chân thành cảm ơn và kính chúc mọi người sức khỏe, hạnh phúc và thành công.

## MỤC LỤC

|   |           |
|---|-----------|
| <b>DANH MỤC HÌNH ẢNH .....</b>                                  | <b>7</b>  |
| <b>DANH MỤC BẢNG BIỂU .....</b>                                 | <b>8</b>  |
| <b>MỞ ĐẦU.....</b>  | <b>9</b>  |
| <b>CHƯƠNG 1: TỔNG QUAN.....</b>                                 | <b>10</b> |
| 1.1. Lý do chọn đề tài .....                                    | 10        |
| 1.2. Mục tiêu đề tài .....                                      | 10        |
| 1.3. Phạm vi đề tài.....  | 10        |
| <b>CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT.....</b>                      | <b>12</b> |
| 2.1. Mô hình MVC .....  | 12        |
| 2.1.1. Giới thiệu tổng quan .....                               | 12        |
| 2.1.2. Các thành phần chính của mô hình MVC .....               | 12        |
| 2.1.3. Cách thức hoạt động của mô hình MVC .....                | 13        |
| 2.1.4. Ưu điểm và nhược điểm của mô hình MVC.....               | 13        |
| 2.1.5. Kết Luận .....   | 14        |
| 2.2. C# .....   | 14        |
| 2.2.1. Giới thiệu tổng quan .....                               | 14        |
| 2.2.2. Đặc điểm nổi bật .....                                   | 15        |
| 2.2.3. Một số nhược điểm .....                                  | 15        |
| 2.2.4. Ứng dụng thực tế .....                                   | 16        |
| 2.3. ASP.NET .....  | 16        |
| 2.3.1. Giới thiệu tổng quan .....                               | 16        |
| 2.3.2. Đặc điểm nổi bật .....                                   | 17        |
| 2.3.3. Các mô hình phát triển trong ASP.NET .....               | 17        |
| 2.3.4. Cấu trúc của một ứng dụng ASP.Net Core MVC.....          | 18        |
| 2.4. Structured Query Language (SQL).....                       | 18        |
| 2.4.1. Giới thiệu tổng quan .....                               | 18        |
| 2.4.2. Các thành phần trong một hệ thống SQL .....              | 19        |
| 2.4.3. SQL hoạt động như thế nào? .....                         | 20        |
| 2.5. Microsoft SQL Server .....                                 | 21        |
| 2.5.1. Giới thiệu tổng quan .....                               | 21        |
| 2.5.2. Các thành phần chính của SQL Server .....                | 21        |
| 2.5.3. Ứng dụng của SQL Server.....                             | 22        |
| 2.6. HTML, CSS và JavaScript .....                              | 22        |
| 2.6.1. Giới thiệu tổng quan .....                               | 22        |
| 2.6.2. Vai trò của HTML, CSS và JavaScript trong trang web..... | 23        |
| 2.7. Responsive Web Design và Bootstrap framework .....         | 24        |
| 2.7.1. Giới thiệu tổng quan .....                               | 24        |
| 2.7.2. Tính năng nổi bật và ứng dụng thực tế .....              | 25        |
| <b>CHƯƠNG 3: ĐÁNH GIÁ KẾT QUẢ.....</b>                          | <b>27</b> |
| 3.1. Mô tả bài toán.....  | 27        |
| 3.2. Phân tích yêu cầu của hệ thống .....                       | 27        |
| 3.2.1. Người dùng.....  | 27        |

|  |           |
|--|-----------|
| 3.2.2. Người Quản trị.....                         | 28        |
| 3.2.3. Nhân viên quản lý .....                     | 28        |
| 3.2.4. Nhân viên tư vấn, chăm sóc khách hàng ..... | 28        |
| 3.3. Quy trình nghiệp vụ.....                      | 28        |
| 3.3.1. Đặt hàng và xử lý đơn hàng .....            | 28        |
| 3.3.2. Quản lý sản phẩm .....                      | 29        |
| 3.3.3. Quản lý người dùng.....                     | 29        |
| 3.3.4. Giao tiếp với khách hàng .....              | 29        |
| 3.4. Use Case .....                                | 29        |
| 3.4.1. Sơ đồ tổng quát.....                        | 29        |
| 3.4.2. Đặc tả Use Case .....                       | 30        |
| 3.5. Sơ đồ Activity .....                          | 47        |
| 3.6. Giao diện hệ thống.....                       | 54        |
| <b>CHƯƠNG 4: KẾT LUẬN.....</b>                     | <b>64</b> |
| <b>CHƯƠNG 5: HƯỚNG PHÁT TRIỂN.....</b>             | <b>65</b> |
| <b>DANH MỤC TÀI LIỆU THAM KHẢO .....</b>           | <b>66</b> |
| <b>PHỤ LỤC .....</b>                               | <b>67</b> |

## **DANH MỤC HÌNH ẢNH**

|   |    |
|---|----|
| Hình 3.1. Sơ đồ tổng quát .....                             | 29 |
| Hình 3.2. Sơ đồ Cơ sở dữ liệu .....                         | 30 |
| Hình 3.3. Sơ đồ Activity – Xem khuyến mại .....             | 47 |
| Hình 3.4. Sơ đồ Activity – Xem bài viết .....               | 48 |
| Hình 3.5. Sơ đồ Activity – Duyệt đơn hàng .....             | 48 |
| Hình 3.6. Sơ đồ Activity – Cập nhật đơn hàng .....          | 49 |
| Hình 3.7. Sơ đồ Activity – Hủy đơn hàng .....               | 49 |
| Hình 3.8. Sơ đồ Activity – Tìm kiếm sản phẩm .....          | 50 |
| Hình 3.9. Sơ đồ Activity – Đăng ký .....                    | 50 |
| Hình 3.10. Sơ đồ Activity – Đăng nhập .....                 | 51 |
| Hình 3.11. Sơ đồ Activity – Quên mật khẩu .....             | 51 |
| Hình 3.12. Sơ đồ Activity – lập báo cáo .....               | 52 |
| Hình 3.13. Sơ đồ Activity – thêm bài viết .....             | 52 |
| Hình 3.14. Sơ đồ Activity – thêm sản phẩm .....             | 53 |
| Hình 3.15. Sơ đồ Activity – thêm nhân viên .....            | 53 |
| Hình 3.16. Sơ đồ Activity – Sửa thông tin nhân viên .....   | 54 |
| Hình 3.17. Giao diện trang chủ .....                        | 54 |
| Hình 3.18. Giao diện trang tin tức .....                    | 55 |
| Hình 3.19. Giao diện phần danh mục sản phẩm .....           | 55 |
| Hình 3.20. Giao diện trang chi tiết sản phẩm .....          | 56 |
| Hình 3.21. Giao diện trang mô tả sản phẩm .....             | 56 |
| Hình 3.22. Giao diện màn hình đăng nhập khách hàng .....    | 57 |
| Hình 3.23. Giao diện giỏ hàng .....                         | 57 |
| Hình 3.24. Giao diện thông tin đặt hàng .....               | 58 |
| Hình 3.25. Giao diện chi tiết đơn hàng .....                | 58 |
| Hình 3.26. Giao diện màn hình đăng nhập cho nhân viên ..... | 59 |
| Hình 3.27. Giao diện trang chủ trang quản trị .....         | 59 |
| Hình 3.28. Giao diện quản lý sản phẩm .....                 | 60 |
| Hình 3.29. Giao diện thêm mới sản phẩm .....                | 60 |
| Hình 3.30. Giao diện cập nhật thông tin sản phẩm .....      | 61 |
| Hình 3.31. Giao diện quản lý tài khoản nhân viên .....      | 61 |
| Hình 3.32. Giao diện thêm mới tài khoản nhân viên .....     | 62 |
| Hình 3.33. Giao diện quản lý phiếu nhập/ xuất kho .....     | 62 |
| Hình 3.34. Giao diện tạo phiếu nhập/ xuất hàng hóa .....    | 63 |

**DANH MỤC BẢNG BIỂU**

|   |    |
|---|----|
| Bảng 2.1. Các nhóm lệnh SQL và mục đích .....                         | 19 |
| Bảng 2.2. Một số thẻ phổ biến và chức năng của chúng trong HTML ..... | 23 |
| Bảng 3.1. Đặc tả Use case: “Livechat”.....                            | 30 |
| Bảng 3.2. Đặc tả Use case: “Xem sản phẩm” .....                       | 31 |
| Bảng 3.3. Đặc tả Use case “Xem khuyến mãi” .....                      | 32 |
| Bảng 3.4. Đặc tả Use case “Xem bài viết” .....                        | 32 |
| Bảng 3.5. Đặc tả Use case “Hủy đơn đặt hàng”.....                     | 33 |
| Bảng 3.6. Đặc tả Use case “Tìm kiếm sản phẩm” .....                   | 34 |
| Bảng 3.7. Đặc tả Use case “Xóa sản phẩm trong giỏ hàng” .....         | 35 |
| Bảng 3.8. Đặc tả Use case “Duyệt đơn hàng” .....                      | 36 |
| Bảng 3.9. Đặc tả Use case “Thanh toán” .....                          | 37 |
| Bảng 3.10. Đặc tả Use case “Đăng ký” .....                            | 38 |
| Bảng 3.11. Đặc tả Use case “Đăng nhập” .....                          | 39 |
| Bảng 3.12. Đặc tả Use case “Lập báo cáo thống kê” .....               | 40 |
| Bảng 3.13. Đặc tả Use case “Tạo bài viết” .....                       | 41 |
| Bảng 3.14. Đặc tả Use case “Sửa bài viết” .....                       | 42 |
| Bảng 3.15. Đặc tả Use case “Xóa bài viết” .....                       | 43 |
| Bảng 3.16. Đặc tả Use case “Thêm vào giỏ hàng” .....                  | 44 |
| Bảng 3.17. Đặc tả Use case “Xóa khỏi giỏ hàng” .....                  | 45 |
| Bảng 3.18. Đặc tả Use case “Thêm sản phẩm” .....                      | 46 |

## MỞ ĐẦU

Trong bối cảnh công nghệ thông tin phát triển vượt bậc, hành vi tiêu dùng của khách hàng đang từng bước chuyển dịch từ mua hàng trực tiếp tại cửa hàng sang hình thức mua sắm trực tuyến. Sự thay đổi này được thúc đẩy bởi nhiều yếu tố như tốc độ kết nối Internet ngày càng nhanh, sự phổ biến của smartphone và đặc biệt là nhu cầu tiết kiệm thời gian trong cuộc sống hiện đại.

Mua sắm trực tuyến mang đến nhiều tiện ích thiết thực: người tiêu dùng có thể dễ dàng tra cứu thông tin sản phẩm, so sánh giá cả, xem đánh giá từ những người mua trước và thực hiện giao dịch mọi lúc, mọi nơi. Đối với các sản phẩm công nghệ như laptop – vốn yêu cầu nhiều thông tin kỹ thuật và mức độ tương thích cao – việc mua sắm online càng trở nên phù hợp nhờ khả năng trình bày chi tiết, lọc theo nhu cầu và hỗ trợ tư vấn trực tuyến.

Đối với doanh nghiệp việc xây dựng website bán hàng không chỉ là một công cụ kinh doanh hiện đại mà còn tạo ra các tác động sâu rộng đến hoạt động vận hành và chiến lược kinh doanh.

Từ đó có thể thấy rằng xu hướng chuyển đổi sang mua hàng trực tuyến không chỉ là lựa chọn cá nhân mà đang trở thành một hướng đi tất yếu trong chiến lược phát triển của các doanh nghiệp. Việc xây dựng một website bán laptop chuyên nghiệp là đáp ứng kịp thời nhu cầu của thị trường, đồng thời mở rộng cơ hội tiếp cận khách hàng trên quy mô rộng lớn hơn.

## CHƯƠNG 1: TỔNG QUAN

### 1.1. Lý do chọn đề tài

Trong nhiều năm qua, mua hàng trực tiếp tại cửa hàng đã là thói quen phổ biến của người tiêu dùng Việt Nam. Tuy nhiên, với sự phát triển nhanh chóng của công nghệ số và sự bùng nổ của Internet, xu hướng mua sắm đang dần thay đổi. Người tiêu dùng hiện nay có xu hướng ưu tiên hình thức mua hàng trực tuyến bởi tính tiện lợi, đa dạng sản phẩm và khả năng so sánh giá cả nhanh chóng. Sự chuyển đổi này không chỉ giúp tiết kiệm thời gian, công sức mà còn mở ra cơ hội tiếp cận hàng hóa từ khắp nơi mà không cần phải di chuyển. Đây chính là nền tảng thúc đẩy sự ra đời của các website thương mại điện tử, đặc biệt là trong lĩnh vực kinh doanh laptop – sản phẩm công nghệ cao có nhu cầu ngày càng lớn.

Đối với doanh nghiệp việc xây dựng một website bán laptop không chỉ giúp doanh nghiệp tiếp cận rộng rãi hơn với khách hàng mà còn tạo ra một kênh phân phối hiệu quả, hiện đại, tối ưu chi phí vận hành.

Năm bắt được nhu cầu đó, em quyết định nghiên cứu và phát triển đề tài “Xây dựng website bán Laptop”. Website không chỉ mang đến trải nghiệm mua sắm tiện lợi và hiện đại cho khách hàng, mà còn đem lại nhiều lợi ích thiết thực cho doanh nghiệp.

### 1.2. Mục tiêu đề tài

- Đề tài "Xây dựng website bán laptop" hướng đến việc nghiên cứu và triển khai một hệ thống thương mại điện tử chuyên nghiệp, thân thiện với người dùng, tích hợp các tính năng cần thiết nhằm phục vụ nhu cầu tìm kiếm, mua bán và quản lý sản phẩm một cách nhanh chóng và tiện lợi..

- Xây dựng giao diện thân thiện, tích hợp chức năng quản lý sản phẩm, giờ hàng, thanh toán và quản lý đơn hàng

- Nghiên cứu công nghệ mới QR- Code và áp dụng vào Website

- Hỗ trợ nhiều nền tảng: máy tính, điện thoại di động, máy tính bảng...

### 1.3. Phạm vi đề tài

Đề tài tập trung nghiên cứu quá trình xây dựng và triển khai website thương mại điện tử phục vụ hoạt động kinh doanh laptop tại thị trường Việt Nam. Phạm vi nghiên

cứu bao gồm các khía cạnh: thiết kế giao diện người dùng, tích hợp chức năng bán hàng trực tuyến, phương thức thanh toán điện tử, quản lý tồn kho và hỗ trợ khách hàng.

Ngoài ra, hệ thống cũng hỗ trợ những chức năng cơ bản khác như: xem sản phẩm, tìm kiếm sản phẩm, xem bài viết, quản lý bài viết, thêm vào giỏ hàng, xóa khỏi giỏ hàng,... Cho phép khách hàng đăng ký tài khoản, đăng nhập trên hệ thống. Những tính năng ưu việt này của hệ thống được tin tưởng và sử dụng.

## CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

### 2.1. Mô hình MVC

#### 2.1.1. Giới thiệu tổng quan

Mô hình MVC (Model-View-Controller) là một mẫu kiến trúc phần mềm được sử dụng phổ biến trong lập trình để phân chia ứng dụng thành ba thành phần chính: Model (Mô hình), View (Giao diện) và Controller (Bộ điều khiển). Mỗi thành phần có một vai trò riêng biệt và tương tác với nhau để tạo nên một ứng dụng hoàn chỉnh.

Kiến trúc MVC được thảo luận lần đầu vào năm 1979 bởi Trygve Reenskaug. Mô hình MVC được giới thiệu lần đầu tiên vào năm 1987 bằng ngôn ngữ lập trình Smalltalk. Vào năm 1988, MVC lần đầu tiên được chấp nhận như một khái niệm chung trong một bài báo. Trong thời gian gần đây, MVC pattern được sử dụng rộng rãi trong các ứng dụng web hiện đại.

#### 2.1.2. Các thành phần chính của mô hình MVC

##### Model (Mô hình dữ liệu)

Đại diện cho **dữ liệu** và **logic nghiệp vụ** của ứng dụng.

Quản lý truy xuất, xử lý và lưu trữ dữ liệu (thường tương tác với cơ sở dữ liệu).

Không biết gì về giao diện người dùng hay cách dữ liệu được hiển thị.

*Ví dụ:* Trong ứng dụng quản lý sinh viên, Student là một model chứa thông tin như tên, tuổi, lớp học và có các phương thức như save(), update()...

##### View (Giao diện hiển thị)

Chịu trách nhiệm hiển thị dữ liệu cho người dùng.

Nhận dữ liệu từ Controller và hiển thị theo định dạng mong muốn.

Không chứa logic xử lý dữ liệu.

*Ví dụ:* Một trang HTML hiển thị danh sách sinh viên hoặc một form nhập liệu.

##### Controller (Bộ điều khiển)

Là cầu nối giữa Model và View.

Nhận yêu cầu từ người dùng, xử lý logic, tương tác với Model và chọn View để hiển thị kết quả.

Điều phối luồng dữ liệu và hành vi của ứng dụng.

Ví dụ: Khi người dùng nhấn nút “Lưu”, Controller sẽ nhận yêu cầu, gọi phương thức save() của Model, rồi chuyển đến View hiển thị thông báo “Lưu thành công”.

### **2.1.3. Cách thức hoạt động của mô hình MVC**

#### **Luồng xử lý trong của mô hình MVC:**

- Khi một yêu cầu của từ máy khách (Client) gửi đến Server. Thì bị Controller trong MVC chặn lại để xem đó là URL request hay sự kiện.
- Sau đó, Controller xử lý input của user rồi giao tiếp với Model trong MVC.
- Model chuẩn bị data và gửi lại cho Controller.
- Cuối cùng, khi xử lý xong yêu cầu thì Controller gửi dữ liệu trả lại View và hiển thị cho người dùng trên trình duyệt.

### **2.1.4. Ưu điểm và nhược điểm của mô hình MVC**

#### **\* Ưu điểm của mô hình MVC**

##### **- Tách biệt rõ ràng giữa các thành phần:**

Giúp phân chia công việc: nhóm thiết kế có thể làm việc với View, nhóm backend xử lý Model và Controller.

Dễ bảo trì và mở rộng ứng dụng.

- **Tái sử dụng mã nguồn:** Model và Controller có thể dùng lại cho nhiều View khác nhau (ví dụ: web, mobile, API).

- **Dễ kiểm thử (Testing):** Do các thành phần độc lập, việc viết unit test cho từng phần trở nên dễ dàng hơn.

- **Hỗ trợ phát triển theo nhóm:** Nhóm làm frontend không cần chờ backend hoàn thành và ngược lại.

- **Tăng hiệu suất phát triển:** Khi đã quen với mô hình, việc phát triển ứng dụng sẽ nhanh và có tổ chức hơn.

### \* Nhược điểm của mô hình MVC

- Độ phức tạp ban đầu: Với những ứng dụng nhỏ hoặc đơn giản, MVC có thể gây “quá tải” vì phải tạo nhiều file và cấu trúc.
- Yêu cầu hiểu biết rõ về kiến trúc: Lập trình viên cần nắm vững cách các thành phần tương tác để tránh sai sót.
- Tốn thời gian thiết kế ban đầu: Việc phân chia rõ ràng các lớp và chức năng đòi hỏi thời gian và công sức.
- Khó khăn khi debug: Do luồng xử lý đi qua nhiều lớp (Controller → Model → View), việc lần theo lỗi có thể phức tạp hơn.

### 2.1.5. Kết Luận

- MVC là một mẫu kiến trúc phân tách ứng dụng thành 1) Model, 2) View và 3) Controller.
  - Model: Nó bao gồm tất cả dữ liệu và logic liên quan của nó.
  - View: Trình bày dữ liệu cho người dùng hoặc xử lý tương tác của người dùng.
  - Controller: Là nơi kết nối giữa các thành phần Model và View.
- Kiến trúc MVC được thảo luận lần đầu vào năm 1979 bởi Trygve Reenskaug.
- MVC là một mô hình dễ dàng mở rộng và bảo trì code.

## 2.2. C#

### 2.2.1. Giới thiệu tổng quan

C# hay C-Sharp là một ngôn ngữ lập trình đa mục đích, được phát triển bởi Microsoft và ra mắt lần đầu tiên vào năm 2000. Nó là một phần của Framework .NET, mang lại khả năng linh hoạt và tính tương tác cao. C# được thiết kế để hỗ trợ việc phát triển các ứng dụng Windows, website và game một cách dễ dàng.

C# được thiết kế bởi Anders Hejlsberg, một kỹ sư phần mềm nổi tiếng người Đan Mạch của Microsoft, người cũng đã tạo ra Turbo Pascal và Delphi. Ý tưởng ban đầu của C# là tạo ra một ngôn ngữ lập trình mới, dựa trên C++ nhưng đơn giản và an toàn hơn. C# cũng được ảnh hưởng bởi các ngôn ngữ khác như Java, Python và Visual Basic.

C# được công bố lần đầu tiên vào tháng 6 năm 2000 và phiên bản chính thức đầu tiên được phát hành vào tháng 1 năm 2002. Với sự hỗ trợ mạnh mẽ từ Microsoft, C# đã nhanh chóng được công nhận là một trong những ngôn ngữ lập trình hàng đầu trên thế giới.

### 2.2.2. Đặc điểm nổi bật

- Ngôn ngữ hiện đại và mạnh mẽ: C# được thiết kế bởi Microsoft với mục tiêu đơn giản hóa việc phát triển phần mềm, kết hợp các tính năng tốt nhất của Java, C++ và Visual Basic.

- Hướng đối tượng thuận túy: C# hỗ trợ đầy đủ 4 tính chất của lập trình hướng đối tượng: kế thừa, đóng gói, trừu tượng và đa hình.

- LINQ – Truy vấn dữ liệu tích hợp: LINQ cho phép truy vấn dữ liệu trực tiếp trong mã C#, giúp viết code ngắn gọn và dễ hiểu hơn.

- Lập trình bất đồng bộ (Asynchronous): Hỗ trợ async/await, giúp cải thiện hiệu suất ứng dụng mà không làm nghẽn luồng chính.

- Extension Methods: Cho phép mở rộng các lớp hiện có mà không cần sửa đổi mã gốc – rất tiện lợi trong việc tái sử dụng mã.

- Pattern Matching: Giúp xử lý kiểu dữ liệu một cách linh hoạt và dễ dàng hơn.

- Nullable Types: Hỗ trợ kiểu dữ liệu có thể null, giúp xử lý các giá trị không xác định một cách an toàn.

- Đa nền tảng: Với .NET Core và .NET 6+, C# có thể chạy trên Windows, Linux, macOS và cả thiết bị di động.

- Mã nguồn mở và cộng đồng mạnh: C# là ngôn ngữ mã nguồn mở, được hỗ trợ bởi cộng đồng lớn và tài liệu phong phú.

### 2.2.3. Một số nhược điểm

- Phụ thuộc vào nền tảng .NET: C# chủ yếu chạy trên nền tảng .NET Framework hoặc .NET Core, nên nếu hệ thống không hỗ trợ .NET, việc triển khai sẽ gặp khó khăn.

- Tối ưu phần cứng không cao: So với các ngôn ngữ như C hoặc C++, C# không tối ưu tốt cho các tác vụ cấp thấp như điều khiển phần cứng hoặc lập trình hệ thống.

- Hiệu suất thấp hơn ngôn ngữ biên dịch truyền thống: Dù C# có hiệu suất khá tốt, nhưng vẫn thua kém C/C++ trong các ứng dụng yêu cầu tốc độ xử lý cao như game engine hoặc xử lý đồ họa chuyên sâu.

- Không hỗ trợ đa kế thừa: C# loại bỏ đa kế thừa như trong C++, thay vào đó sử dụng interface. Điều này giúp tránh xung đột nhưng cũng hạn chế một số mô hình thiết kế.

- Không lý tưởng cho lập trình hệ thống: Nếu bạn cần viết driver, hệ điều hành hoặc phần mềm nhúng, C# không phải lựa chọn tối ưu do thiếu khả năng thao tác trực tiếp với bộ nhớ và phần cứng.

#### 2.2.4. Ứng dụng thực tế

- **Web:** ASP.NET Core giúp xây dựng website hiện đại, bảo mật cao.

- **Desktop:** Windows Forms, WPF cho ứng dụng giao diện người dùng.

- **Mobile:** Xamarin hỗ trợ phát triển ứng dụng Android/iOS.

- **Game:** Unity – engine game phổ biến nhất hiện nay – dùng C# làm ngôn ngữ chính.

- **AI & Machine Learning:** ML.NET cho phép xây dựng mô hình học máy bằng C#.

- **Cloud & IoT:** Tích hợp tốt với Azure và các nền tảng đám mây.

### 2.3. ASP.NET

#### 2.3.1. Giới thiệu tổng quan

ASP.Net là một công nghệ web của Microsoft, cho phép xử lý hoàn toàn trên server và nó hoàn toàn độc lập với mọi trình duyệt. Nghĩa là trình duyệt không cần cài đặt bất kỳ công cụ hỗ trợ nào khi duyệt trang web định dạng .aspx.

ASP.NET là phiên bản kế tiếp của Active Server Page (ASP), nó là một nền phát triển ứng dụng Web hợp nhất, cung cấp nhiều dịch vụ cần thiết cho lập trình viên xây dựng các lớp ứng dụng chuyên nghiệp. Cú pháp ASP.NET tương thích với cú pháp ASP, ngoài ra ASP.NET còn cung cấp một mô hình lập trình mới, nền tảng an toàn (secure), linh hoạt (scalable), và ổn định (stable). Sẽ cảm thấy dễ chịu khi nâng cấp ứng dụng ASP hiện có, bằng cách đưa vào các hàm ASP.NET cho chúng (lời của MS).

Khi tạo một ứng dụng Web lập trình viên có thể chọn Web Forms hoặc Web Services, ngoài ra cũng có thể kết hợp hai loại này với nhau theo bất kỳ cách nào. Cả

hai loại này có một nền cơ bản, cho phép sử dụng authentication schemes, cache frequently used data hoặc chỉnh sửa cấu hình ứng dụng (customize application's configuration). Tuy nhiên mỗi loại có một vài khả năng riêng.

Web Forms cho phép xây dựng các form có nội dung phong phú (powerful forms) trên trang Web (Web Page). Khi xây dựng các form có thể sử dụng các server control để tạo các thành phần UI thông dụng và lập trình cho chúng một vài chức năng thông dụng, những control này cho phép xây dựng nhanh chóng các form chỉ dùng một lần (out of reusable built-in) hoặc các custom component, làm đơn giản mã của trang web (simplifying the code of a page).

ASP.Net được thiết kế tương thích với các phiên bản ASP (Active Server Page) trước đó. Cụ thể triển khai ứng dụng ASP.Net và ASP trên cùng một server mà không cần thay đổi cấu hình ứng dụng ASP. Tuy chạy trên cùng một Server nhưng chúng hoạt động độc lập với nhau.

Công nghệ ASP.Net là một phần công nghệ của .Net Framework, các ý tưởng công nghệ và các thành phần đều nằm trong công nghệ .Net.

### 2.3.2. Đặc điểm nổi bật

- Độc lập ngôn ngữ: Có thể dùng VB, C#, thậm chí C++ để xây dựng trang ASP.Net. Điều này đã thực hiện tối ưu việc kết hợp các ngôn ngữ.

- Dễ phát triển: Khai báo và mã hóa đơn giản. Việc phát triển một ứng dụng Web được tiến hành một cách chuyên nghiệp hơn. Vẫn đề quan tâm không phải là việc viết mã mà quan trọng hơn đó là ghép các thành phần ứng dụng, việc sử dụng các dịch vụ.

#### - Dễ dàng thay đổi và nâng cấp

- Hỗ trợ nhiều trình duyệt: người lập trình không phải quan tâm đến nhận dạng trình duyệt phía client, các điều khiển ASP.Net có thể tự động nhận dạng trình duyệt khách để hiển thị cho phù hợp.

#### - Xử lý trên Server

#### - Cung cấp dịch vụ mới hiệu quả: Webservice.

### 2.3.3. Các mô hình phát triển trong ASP.NET

#### - Web Forms: Giao diện kéo thả, dễ dùng cho người mới.

- ASP.NET MVC: Kiến trúc Model-View-Controller, tách biệt logic và giao diện.
- ASP.NET Web Pages: Sử dụng Razor syntax, viết HTML và C# trong cùng một trang.
- ASP.NET Core: Phiên bản hiện đại, hiệu suất cao, chạy đa nền tảng.

#### **2.3.4. Cấu trúc của một ứng dụng ASP.Net Core MVC**

- Web Form và Razor Pages: Cung cấp diện rộng cho ứng dụng web với phần mở rộng .cshtml, cho phép xây dựng các nội dung phong phú và tương tác phức tạp.
- Trang Code-behind: Liên kết với Razor Pages, chứa mã xử lý sự kiện từ giao diện, tăng cường trách nhiệm phân biệt rõ ràng giữa logic và giao diện.
- File cấu hình: Định dạng XML, định nghĩa các thiết lập chung cho ứng dụng như khai báo kết nối cơ sở dữ liệu. Mỗi ứng dụng web đều có một file cấu hình Web.config.
- File Global.asax: Chứa các code cần thiết cho việc đáp ứng lại các sự kiện ứng dụng, file này được sinh tự động bởi ASP.Net.
- Web Service và API: Hỗ trợ giao tiếp và trao đổi dữ liệu qua mạng bằng cách sử dụng Web API, cho phép xây dựng các ứng dụng và dịch vụ mạng linh hoạt và mở rộng.
- Kết nối cơ sở dữ liệu: Kết nối cơ sở dữ liệu cho phép ứng dụng cơ thể chuyển dữ liệu từ một nguồn ứng dụng hoặc đọc ra từ đó.
- Các định nghĩa không gian: Tên miền, lớp do người dùng định nghĩa được sử dụng trong toàn ứng dụng với file định dạng .vb (nếu sử dụng VB.Net) và .cs (nếu sử dụng C#).

### **2.4. Structured Query Language (SQL)**

#### **2.4.1. Giới thiệu tổng quan**

Structured Query Language (SQL) là ngôn ngữ truy vấn có cấu trúc dùng để tương tác với cơ sở dữ liệu quan hệ (RDBMS) như MySQL, SQL Server, Oracle, PostgreSQL... Nó là công cụ chính để lưu trữ, truy xuất, cập nhật và quản lý dữ liệu trong các hệ thống thông tin.

SQL là một ngôn ngữ truy vấn phổ biến thường được sử dụng trong tất cả các loại ứng dụng. Các nhà phân tích và phát triển dữ liệu tìm hiểu và sử dụng SQL do ngôn ngữ

này tích hợp hiệu quả với nhiều ngôn ngữ lập trình khác nhau. Ví dụ: họ có thể nhúng truy vấn SQL với ngôn ngữ lập trình Java để xây dựng ứng dụng xử lý dữ liệu hiệu năng cao có các hệ thống cơ sở dữ liệu SQL chính như Oracle hoặc MS SQL Server. SQL cũng khá dễ tìm hiểu do nó sử dụng những từ khóa tiếng Anh phổ biến trong các câu lệnh.

SQL được dùng để:

- Truy vấn dữ liệu: Lấy thông tin từ các bảng trong cơ sở dữ liệu.
- Thêm, sửa, xóa dữ liệu: Quản lý nội dung trong bảng.
- Tạo và thay đổi cấu trúc bảng: Xây dựng hệ thống lưu trữ dữ liệu.
- Phân quyền truy cập: Kiểm soát ai được phép làm gì với dữ liệu.

#### 2.4.2. Các thành phần trong một hệ thống SQL

Hệ thống quản lý cơ sở dữ liệu quan hệ sử dụng ngôn ngữ truy vấn có cấu trúc (SQL) để lưu trữ và quản lý dữ liệu. Hệ thống này lưu trữ nhiều bảng cơ sở dữ liệu có liên quan đến nhau. MS SQL Server, MySQL hoặc MS Access là những ví dụ về hệ thống quản lý cơ sở dữ liệu quan hệ. Một hệ thống như vậy có các thành phần sau:

**- Bảng SQL:** Bảng SQL là phần tử cơ bản của cơ sở dữ liệu quan hệ. Bảng cơ sở dữ liệu SQL bao gồm các hàng và cột. Kỹ sư cơ sở dữ liệu tạo ra các mối quan hệ giữa nhiều bảng cơ sở dữ liệu để tối ưu hóa không gian lưu trữ dữ liệu.

**- Câu lệnh SQL:** Câu lệnh SQL hoặc truy vấn SQL là các lệnh hướng dẫn hợp lệ mà hệ thống quản lý cơ sở dữ liệu quan hệ hiểu được. Nhà phát triển phần mềm xây dựng các câu lệnh SQL bằng nhiều phần tử ngôn ngữ SQL khác nhau. Phần tử ngôn ngữ SQL là các thành phần như mã định danh, biến và điều kiện tìm kiếm tạo thành một câu lệnh SQL đúng.

Bảng 2.1. Các nhóm lệnh SQL và mục đích

| Các nhóm lệnh SQL                       | Mục đích                               |
|---|--|
| <b>DDL</b> (Data Definition Language)   | Tạo, sửa, xóa bảng và cấu trúc dữ liệu |
| <b>DQL</b> (Data Query Language)        | Truy vấn dữ liệu từ cơ sở dữ liệu      |
| <b>DML</b> (Data Manipulation Language) | Thêm, sửa, xóa dữ liệu trong bảng      |

| Các nhóm lệnh SQL                         | Mục đích                                     |
|---|--|
| <b>DCL</b> (Data Control Language)        | Phân quyền truy cập dữ liệu                  |
| <b>TCL</b> (Transaction Control Language) | Quản lý giao dịch dữ liệu (commit, rollback) |

- **Quy trình được lưu trữ:** Quy trình được lưu trữ là tập hợp bao gồm một hoặc nhiều câu lệnh SQL được lưu trữ trong cơ sở dữ liệu quan hệ. Nhà phát triển phần mềm sử dụng các quy trình được lưu trữ để cải thiện hiệu quả và hiệu suất. Ví dụ: họ có thể tạo một quy trình được lưu trữ để cập nhật bảng bán hàng thay vì viết cùng một câu lệnh SQL trong nhiều ứng dụng khác nhau.

#### 2.4.3. SQL hoạt động như thế nào?

Việc triển khai ngôn ngữ truy vấn có cấu trúc (SQL) liên quan đến một máy chủ xử lý truy vấn cơ sở dữ liệu và trả về kết quả. Quá trình SQL đi qua một số thành phần phần mềm, bao gồm những thành phần sau:

\* **Trình phân tích cú pháp:** Trình phân tích cú pháp bắt đầu bằng cách token hóa hoặc thay thế một số từ trong câu lệnh SQL bằng các ký hiệu đặc biệt. Sau đó, nó sẽ kiểm tra câu lệnh để tìm kiếm những yếu tố sau:

- **Tính đúng đắn:** Trình phân tích cú pháp xác minh rằng câu lệnh SQL tuân theo ngữ nghĩa SQL, hay các quy tắc, đảm bảo tính đúng đắn của câu lệnh truy vấn. Ví dụ: trình phân tích cú pháp kiểm tra xem lệnh SQL có kết thúc bằng dấu chấm phẩy hay không. Nếu thiếu dấu chấm phẩy, trình phân tích cú pháp sẽ trả về lỗi.

- **Quyền hạn:** Trình phân tích cú pháp cũng xác thực rằng người dùng đang chạy truy vấn có quyền cần thiết để thao tác với dữ liệu tương ứng. Ví dụ: chỉ người dùng quản trị mới có quyền xóa dữ liệu.

\* **Công cụ quan hệ:** Công cụ quan hệ, hay bộ xử lý truy vấn, tạo kế hoạch truy xuất, ghi hoặc cập nhật dữ liệu tương ứng theo cách hiệu quả nhất. Ví dụ: công cụ này kiểm tra các truy vấn tương tự, sử dụng lại các phương pháp thao tác dữ liệu trước đó hoặc tạo một phương pháp mới. Công cụ quan hệ viết kế hoạch trong mã byte, một dạng biểu diễn trung cấp của câu lệnh SQL. Cơ sở dữ liệu quan hệ sử dụng mã byte để thực hiện tìm kiếm và điều chỉnh cơ sở dữ liệu một cách hiệu quả.

\* **Công cụ lưu trữ:** Công cụ lưu trữ hoặc công cụ cơ sở dữ liệu, là thành phần phần mềm xử lý mã byte và chạy câu lệnh SQL dự định. Công cụ này đọc và lưu trữ dữ liệu trong các tệp cơ sở dữ liệu trên ổ đĩa lưu trữ vật lý. Sau khi hoàn tất, công cụ lưu trữ trả về kết quả cho ứng dụng yêu cầu.

## 2.5. Microsoft SQL Server

### 2.5.1. Giới thiệu tổng quan

**Microsoft SQL Server** hay còn gọi tắt là **SQL Server**, là một **hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS)** do Microsoft phát triển từ năm 1989. Nó được thiết kế để lưu trữ, quản lý, truy vấn và phân tích dữ liệu một cách hiệu quả và an toàn.

#### SQL Server dùng để:

- Tạo và duy trì cơ sở dữ liệu cho các ứng dụng web, desktop, mobile.
- Truy vấn và thao tác dữ liệu bằng ngôn ngữ SQL hoặc T-SQL (Transact-SQL).
- Phân tích dữ liệu và tạo báo cáo thông qua các công cụ như SSRS (Reporting Services).
- Tự động hóa tác vụ như sao lưu, phục hồi, xử lý định kỳ bằng SQL Server Agent.

#### Các phiên bản phổ biến:

- **Express:** Miễn phí, giới hạn tính năng – phù hợp cho học tập.
- **Standard:** Đầy đủ tính năng cho doanh nghiệp vừa.
- **Enterprise:** Tối đa hiệu suất, bảo mật – dành cho doanh nghiệp lớn.
- **Developer:** Miễn phí, đầy đủ tính năng – dùng cho phát triển và thử nghiệm.

### 2.5.2. Các thành phần chính của SQL Server

- **Database Engine:** Là trái tim của SQL Server. Chịu trách nhiệm lưu trữ, xử lý, truy vấn và bảo mật dữ liệu. Hỗ trợ các thao tác như tạo bảng, chèn dữ liệu, truy vấn dữ liệu và quản lý giao dịch.

- **SQL Server Management Studio (SSMS):** Giao diện đồ họa giúp quản trị cơ sở dữ liệu. Cho phép người dùng viết truy vấn, quản lý bảng, backup dữ liệu và theo dõi hiệu suất.

- **Transact-SQL (T-SQL)**: Là phần mở rộng của SQL do Microsoft phát triển. Hỗ trợ các tính năng nâng cao như khai báo biến, vòng lặp, xử lý ngoại lệ, thủ tục lưu trữ (stored procedures).

- **SQL Server Agent**: Công cụ tự động hóa các tác vụ như sao lưu, phục hồi, chạy truy vấn định kỳ. Hỗ trợ lập lịch và giám sát các quy trình trong hệ thống.

- **Integration Services (SSIS)**: Dùng để nhập, xuất và chuyển đổi dữ liệu giữa các hệ thống khác nhau. Hỗ trợ ETL (Extract, Transform, Load) trong các dự án dữ liệu lớn.

- **Reporting Services (SSRS)**: Tạo và quản lý các báo cáo động từ dữ liệu trong SQL Server. Hỗ trợ xuất báo cáo ra nhiều định dạng như PDF, Excel, Word.

- **Analysis Services (SSAS)**: Dùng để phân tích dữ liệu, xây dựng mô hình OLAP và khai phá dữ liệu. Hỗ trợ các ứng dụng BI (Business Intelligence).

### 2.5.3. Ứng dụng của SQL Server

SQL Server là một công cụ mạnh mẽ để quản lý và làm việc với dữ liệu, được sử dụng rộng rãi trong nhiều ngành công nghiệp khác nhau, bao gồm:

- **Thương mại điện tử**: Quản lý kho hàng, xử lý giao dịch và phân tích hành vi người dùng.

- **Ngân hàng**: Quản lý tài khoản, giao dịch và các hoạt động tài chính khác.

- **Doanh nghiệp**: Quản lý dữ liệu khách hàng, dữ liệu nhân sự và các dữ liệu kinh doanh khác.

- **Ứng dụng web và di động**: Cung cấp nền tảng lưu trữ dữ liệu cho các ứng dụng web và di động.

## 2.6. HTML, CSS và JavaScript

### 2.6.1. Giới thiệu tổng quan

**HTML** (viết tắt của **HyperText Markup Language**) là **ngôn ngữ đánh dấu siêu văn bản** dùng để tạo và cấu trúc nội dung trên các trang web. Nó không phải là ngôn ngữ lập trình, mà là ngôn ngữ giúp trình duyệt hiểu và hiển thị nội dung web một cách có tổ chức.

**CSS** (viết tắt của **Cascading Style Sheets**) là một **ngôn ngữ định kiểu** dùng để thiết kế và trình bày giao diện cho các tài liệu HTML hoặc XML. Nói cách khác, nếu HTML là phần **nội dung và cấu trúc** của trang web, thì CSS là phần **hình thức và thẩm mỹ** – giúp trang web trở nên đẹp mắt, dễ nhìn và chuyên nghiệp hơn.

**JavaScript** (viết tắt là **JS**) là một **ngôn ngữ lập trình bậc cao**, được sử dụng phổ biến nhất trong phát triển web để tạo ra các **trang web tương tác, sống động và linh hoạt**.

**HTML, CSS và JavaScript** là ba công nghệ cốt lõi trong phát triển web. Chúng hoạt động cùng nhau để tạo ra các trang web hoàn chỉnh – từ cấu trúc, giao diện đến tương tác. Khi kết hợp cả ba, bạn có thể xây dựng một trang web hoàn chỉnh: có nội dung rõ ràng, giao diện đẹp và trải nghiệm người dùng mượt mà.

### 2.6.2. Vai trò của HTML, CSS và JavaScript trong trang web

**HTML – Xây dựng cấu trúc trang web:** HTML Là nền tảng cơ bản nhất trong phát triển web, được dùng để tạo cấu trúc cho trang web như: tiêu đề, đoạn văn, hình ảnh, liên kết, bảng,..

Bảng 2.2. Một số thẻ phổ biến và chức năng của chúng trong HTML

| Thẻ     | Chức năng                           |
|---------|-------------------------------------|
| <h1>    | Định nghĩa 1 Tiêu đề lớn            |
| <p>     | Định nghĩa 1 Đoạn văn bản           |
| <a>     | Định nghĩa 1 Liên kết               |
| <img>   | Định nghĩa 1 Hình ảnh               |
| <ul>    | Định nghĩa 1 Danh sách không thứ tự |
| <table> | Định nghĩa 1 Bảng dữ liệu           |
| ...     | ...                                 |

## CSS – Trang trí và định dạng giao diện:

CSS được dùng để:

- **Tạo kiểu dáng cho phần tử HTML:** màu sắc, font chữ, kích thước, khoảng cách, bố cục,...
- **Thiết kế giao diện linh hoạt:** thay đổi cách hiển thị tùy theo thiết bị (responsive design).
- **Tách biệt nội dung và trình bày:** giúp mã HTML gọn gàng, dễ bảo trì.

CSS có thể được viết trực tiếp trong thẻ HTML (**Inline CSS**), viết trong cặp thẻ `<style></style>` trong phần `<head>` (**Internal CSS**) hoặc viết trong file `.css` riêng và liên kết qua thẻ `<link>` (**External CSS**)

## JavaScript – Tạo tương tác và chức năng động

JavaScript được dùng để xử lý hành vi và tương tác trên trang web:

- Tạo hiệu ứng động: như slideshow, pop-up, menu thả, hiệu ứng cuộn,...
- Xử lý sự kiện người dùng: như click chuột, nhập dữ liệu, di chuyển chuột,...
- Giao tiếp với máy chủ: thông qua AJAX, giúp tải dữ liệu mà không cần reload trang.
- Thay đổi nội dung HTML và CSS: giúp cập nhật giao diện theo thời gian thực.
- Xây dựng ứng dụng web: như game, ứng dụng di động, phần mềm chạy trên trình duyệt.

## 2.7. Responsive Web Design và Bootstrap framework

### 2.7.1. Giới thiệu tổng quan

**Responsive Web Design** (RWD), hay còn gọi là **thiết kế web đáp ứng**, là phương pháp thiết kế website sao cho **giao diện và nội dung tự động điều chỉnh** để hiển thị tốt trên mọi thiết bị: từ máy tính để bàn, laptop, tablet đến điện thoại di động

**Responsive Web Design** giúp trang web:

- Tương thích đa thiết bị: Một website duy nhất có thể tự động điều chỉnh kích thước để hiển thị vừa vặn trên tất cả các thiết bị truy cập.

- Tiết kiệm chi phí: Không cần tạo nhiều phiên bản web riêng cho từng thiết bị.
- Tối ưu SEO: Google ưu tiên các trang web có thiết kế responsive trong xếp hạng tìm kiếm.
- Trải nghiệm người dùng tốt hơn: Giao diện dễ nhìn, dễ thao tác, không bị vỡ bố cục hay chữ quá nhỏ

**Bootstrap** là một framework (mã nguồn mở) **bao gồm các HTML, CSS và JavaScript**. Trong bootstrap có các thành tố có sẵn như typography (kiểu chữ), forms (biểu mẫu), buttons (nút), tables (bảng), grids (bản mô tả ngắn), navigation (điều hướng trang web), image (ảnh),... tóm lại là các template (mẫu có sẵn) dùng để thiết kế và tối ưu website chuẩn responsive.

### 2.7.2. Tính năng nổi bật và ứng dụng thực tế

**Responsive Web Design** là tư duy thiết kế giúp website hiển thị tốt trên mọi thiết bị và **Bootstrap** là một công cụ giúp bạn thực hiện RWD dễ dàng hơn nhờ các thành phần dựng sẵn và hệ thống lưới mạnh mẽ.

#### Tính năng nổi bật của Bootstrap

- Thiết kế responsive (đa thiết bị) Bootstrap được xây dựng theo triết lý mobile-first, giúp giao diện tự động điều chỉnh theo kích thước màn hình từ điện thoại đến desktop.
- Hệ thống lưới 12 cột (Grid System) Cho phép chia bố cục linh hoạt, dễ dàng căn chỉnh và sắp xếp nội dung theo hàng và cột.
- Thành phần dựng sẵn (Prebuilt Components) Bao gồm các phần tử như: nút (button), biểu mẫu (form), menu điều hướng (navbar), hộp thoại (modal), trình chiếu ảnh (carousel), v.v.
- Tùy biến dễ dàng Có thể ghi đè (override) các class CSS để tùy chỉnh giao diện theo ý muốn mà không cần viết lại từ đầu.
- Tài liệu đầy đủ và cộng đồng lớn Bootstrap có tài liệu hướng dẫn chi tiết, ví dụ minh họa rõ ràng và cộng đồng hỗ trợ mạnh mẽ.
- Tính nhất quán giao diện Giúp các lập trình viên làm việc nhóm dễ dàng duy trì phong cách thiết kế thống nhất trên toàn bộ website.

- Dễ tích hợp và sử dụng Chỉ cần nhúng một dòng CDN là có thể sử dụng ngay các tính năng của Bootstrap mà không cần cài đặt phức tạp.

### **Ứng dụng thực tế của Bootstrap**

- Xây dựng website cá nhân hoặc doanh nghiệp Tạo giao diện đẹp, chuyên nghiệp cho blog, portfolio, trang giới thiệu công ty...

- Thiết kế giao diện admin dashboard Bootstrap rất phù hợp để tạo các bảng điều khiển quản trị với bố cục rõ ràng và responsive.

- Tạo landing page cho sản phẩm hoặc dịch vụ Giúp bạn nhanh chóng triển khai trang giới thiệu sản phẩm với hiệu ứng đẹp mắt.

- Phát triển giao diện cho ứng dụng web Bootstrap hỗ trợ tốt cho các ứng dụng như hệ thống quản lý, CRM, hệ thống đặt hàng...

- Tích hợp vào CMS như WordPress, Joomla Nhiều theme và plugin sử dụng Bootstrap để đảm bảo giao diện đẹp và dễ tùy chỉnh.

## CHƯƠNG 3: ĐÁNH GIÁ KẾT QUẢ

### 3.1. Mô tả bài toán

Website bán Laptop là một website chuyên cung cấp các sản phẩm laptop chất lượng với nhiều mức giá và cấu hình đa dạng. Website thường xuyên cập nhật các sản phẩm mới, combo hấp dẫn như laptop + chuột không dây, laptop + balo chống sốc hoặc combo học tập + văn phòng, kèm theo các mã giảm giá và dịch vụ ưu đãi để khách hàng có thể dễ dàng lựa chọn theo nhu cầu của mình.

Với mỗi sản phẩm laptop, website sẽ hiển thị đầy đủ thông tin như: tên sản phẩm, thông số kỹ thuật chi tiết (CPU, RAM, ổ cứng, VGA, kích thước màn hình, v.v.), mô tả, giá bán và các ưu đãi đi kèm. Khi khách hàng nhấn chọn vào một sản phẩm cụ thể, hệ thống sẽ hiện ra đầy đủ thông tin chi tiết để khách hàng tham khảo.

Khách hàng có thể tìm kiếm laptop theo từ khóa (ví dụ: "i5", "16GB RAM", "ASUS") hoặc tìm theo loại máy như: laptop văn phòng, laptop gaming, laptop đồ họa, laptop sinh viên, thông qua thanh tìm kiếm hoặc danh mục trên giao diện chính. Website còn có tính năng chat trực tiếp, hỗ trợ khách hàng tư vấn nhanh chóng với nhân viên khi cần giải đáp thắc mắc.

Để sử dụng các tính năng như giỏ hàng, đặt hàng... khách hàng cần đăng ký và đăng nhập tài khoản. Việc đăng ký có thể thực hiện nhanh chóng qua liên kết Facebook, Zalo, Google hoặc điền đầy đủ thông tin như số điện thoại và mật khẩu vào form đăng ký.

Sau khi tìm được sản phẩm ưng ý, khách hàng có thể mua laptop trực tuyến bằng cách nhấn chọn nút “Mua Laptop”, chọn số lượng rồi xác nhận thêm vào giỏ hàng. Nếu muốn bỏ sản phẩm nào, có thể xóa khỏi giỏ hàng. Khi hoàn tất việc chọn mua, khách hàng xác nhận đơn hàng trong giỏ và nhấn “Đặt hàng”. Hệ thống sẽ gửi thông báo tới nhân viên kiểm tra hàng tồn kho. Nếu còn sản phẩm, khách sẽ được mời chọn hình thức thanh toán, nếu không có, hệ thống sẽ thông báo “hết hàng” và gợi ý sản phẩm tương tự.

### 3.2. Phân tích yêu cầu của hệ thống

#### 3.2.1. Người dùng

**- Đăng ký và Đăng nhập:** Người dùng có thể đăng ký tài khoản mới hoặc đăng nhập vào hệ thống.

- **Quản lý Tài khoản:** Người dùng có thể chỉnh sửa thông tin cá nhân sau khi đăng nhập, có thể đăng xuất khỏi tài khoản.

- **Tìm kiếm sản phẩm:** Người dùng có thể tìm kiếm sản phẩm theo tên, mô tả, nhãn hàng hoặc nhãn.

- **Mua hàng:** Chọn sản phẩm và thêm vào giỏ hàng, xem và quản lý giỏ hàng, thực hiện thanh toán, đặt hàng.

- **Quản lý đơn hàng:** Xem lịch sử giao dịch và đơn hàng đã mua.

- **Liên hệ nhân viên tư vấn:** Người dùng có thể nhắn tin với nhân viên qua live chat.

### 3.2.2. Người Quản trị

- **Phân quyền:** Người quản trị có quyền quản lý và phân quyền cho các tài khoản trong hệ thống.

- **Quản lý người dùng:** Xem, sửa, xóa thông tin của các tài khoản trong hệ thống.

- **Thống kê và quản lý tài chính:** Xem và thống kê doanh thu, lợi nhuận thông qua biểu đồ.

### 3.2.3. Nhân viên quản lý

- **Quản lý sản phẩm:** Xem sản phẩm, chi tiết sản phẩm, thêm, xóa, sửa sản phẩm.

- **Quản lý đơn hàng:** Xem, sửa, xóa đơn hàng.

### 3.2.4. Nhân viên tư vấn, chăm sóc khách hàng

**Giao tiếp với khách hàng:** Nhân viên tư vấn có thể nhắn tin hoặc gọi điện với khách hàng để tư vấn thông qua livechat, mạng xã hội, điện thoại...

## 3.3. Quy trình nghiệp vụ

### 3.3.1. Đặt hàng và xử lý đơn hàng

- Khách hàng thực hiện tìm kiếm sản phẩm và thêm vào giỏ hàng.

- Khách hàng xem lại giỏ hàng, điều chỉnh số lượng và tiến hành thanh toán.

- Hệ thống nhận đơn hàng và xác nhận thanh toán (nếu có).

- Nhân viên quản lý kiểm tra và xác nhận đơn hàng.

- Hệ thống cập nhật trạng thái đơn hàng và thông báo cho khách hàng.

- Nhân viên chuẩn bị hàng và tiến hành giao cho đơn vị vận chuyển.
- Khách hàng nhận hàng, kiểm tra hàng hóa và xác nhận.
- Hệ thống cập nhật trạng thái đơn hàng và hoàn tất giao dịch
- Chăm sóc khách hàng liên hệ khách hàng để xác nhận hàng hóa, nhận thông tin đánh giá từ khách hàng.

### 3.3.2. Quản lý sản phẩm

- Nhân viên quản lý thêm, sửa, xóa sản phẩm.
- Hệ thống cập nhật danh sách sản phẩm và chi tiết sản phẩm.
- Cập nhật và hiển thị sản phẩm lên Website.

### 3.3.3. Quản lý người dùng

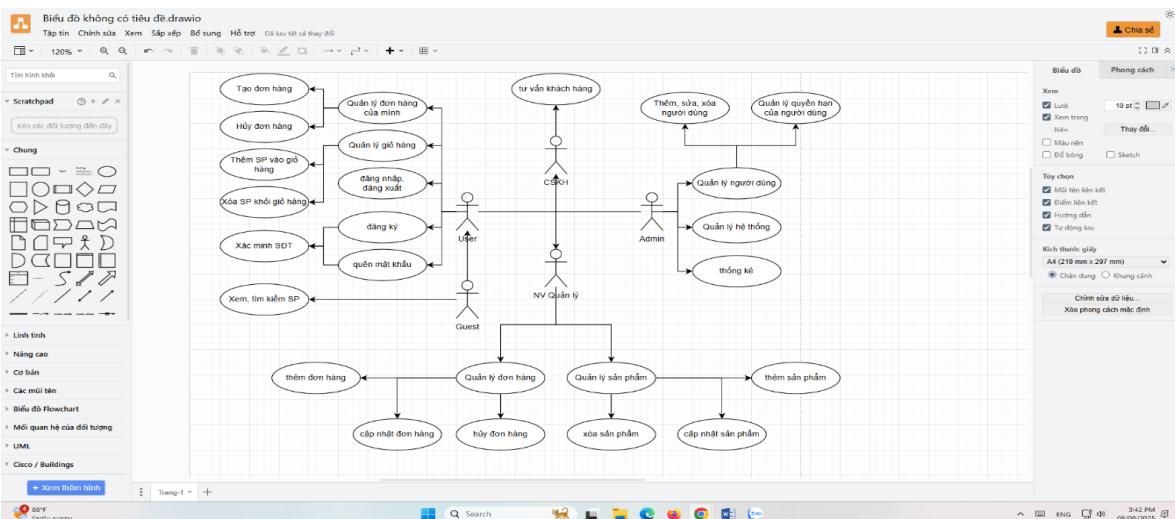
- Người quản trị xem thông tin và hoạt động của người dùng trong hệ thống.
- Thực hiện phân quyền, thêm, sửa, xóa tài khoản người dùng.
- Hệ thống cập nhật trạng thái của tài khoản người dùng sau khi thay đổi.

### 3.3.4. Giao tiếp với khách hàng

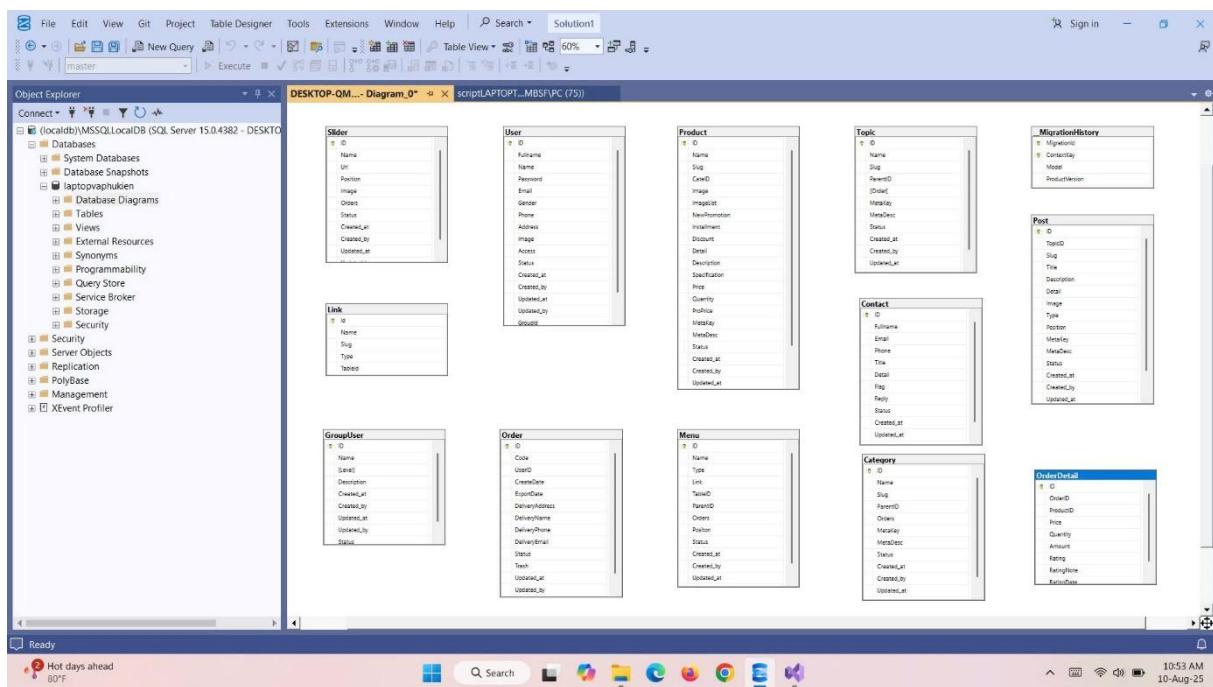
- Khách hàng gửi yêu cầu hỗ trợ hoặc gửi tin nhắn đến hệ thống
- Nhân viên tư vấn tiếp nhận và xử lý yêu cầu của khách hàng.

## 3.4. Use Case

### 3.4.1. Sơ đồ tổng quát



Hình 3.1. Sơ đồ tổng quát



Hình 3.2. Sơ đồ Cơ sở dữ liệu

### 3.4.2. ĐẶC TẢ USE CASE

#### - Use case: “Livechat”:

Bảng 3.1. Đặc tả Use case: “Livechat”

|   |                                 |
|---|---------------------------------|
| <b>- Tên use case: Livechat</b>   |                                 |
| <b>- Mô tả sơ lược:</b> Chức năng livechat giúp khách hàng có thể trao đổi trực tuyến với nhân viên trên hệ thống.      |                                 |
| <b>- Actor chính:</b> Khách hàng  |                                 |
| <b>- Actor phụ:</b> Không   |                                 |
| <b>- Tiền điều kiện (Pre-condition):</b> Đăng nhập thành công   |                                 |
| <b>- Hậu điều kiện (Post-condition):</b> Nếu khách hàng livechat với cửa hàng thì hệ thống sẽ lưu lại lịch sử trao đổi. |                                 |
| <b>- Luồng sự kiện chính (main flow):</b>   |                                 |
| <b>Khách hàng</b>   | <b>Hệ thống</b>                 |
| 1. Người dùng chọn chức năng livechat tại giao diện   | 2. Hệ thống hiển thị khung chat |

|  |  |
|--|--|
| 3. Người dùng nhập nội dung vào khung chat và nhấn nút gửi |  |
| <b>- Luồng sự kiện thay thế (alternate flow):</b>          |  |
| Không  |  |
| <b>- Luồng sự kiện ngoại lệ (exception flow):</b>          |  |
| Không  |  |

**- Use case “Xem sản phẩm”:**

Bảng 3.2. Đặc tả Use case: “Xem sản phẩm”

|  |  |
|--|--|
| <b>- Tên use case:</b> Xem sản phẩm  |  |
| <b>- Mô tả sơ lược:</b> Chức năng giúp khách hàng có thể xem sản phẩm trên Website dễ dàng |  |
| <b>- Actor chính:</b> Khách hàng   |  |
| <b>- Actor phụ:</b> Không  |  |
| <b>- Tiền điều kiện (Pre-condition):</b> Kết nối Internet                                  |  |
| <b>- Hậu điều kiện (Post-condition):</b> Không   |  |
| <b>- Luồng sự kiện chính (main flow):</b>  |  |
| <b>Khách hàng</b>  | <b>Hệ thống</b>  |
| 1. Khách hàng chọn sản phẩm muốn xem   | 2. Hệ thống hiển thị thông tin sản phẩm mà khách hàng chọn |
| <b>- Luồng sự kiện thay thế (alternate flow):</b>  |  |
| Không  |  |
| <b>- Luồng sự kiện ngoại lệ (exception flow):</b>  |  |
| Không  |  |

**- Use case “Xem khuyến mãi”:**

Bảng 3.3. Đặc tả Use case “Xem khuyến mãi”

| <b>- Tên use case:</b> Xem khuyến mãi   |   |                 |   |   |  |   |
|---|---|-----------------|---|---|--|---|
| <b>- Mô tả sơ lược:</b> Chức năng giúp khách hàng biết được các thông tin hoặc sản phẩm nào đang khuyến mãi để đặt mua nếu có nhu cầu.  |   |                 |   |   |  |   |
| <b>- Actor chính:</b> Khách hàng  |   |                 |   |   |  |   |
| <b>- Actor phụ:</b> Không   |   |                 |   |   |  |   |
| <b>- Tiền điều kiện (Pre-condition):</b> Kết nối Internet   |   |                 |   |   |  |   |
| <b>- Hậu điều kiện (Post-condition):</b> Không  |   |                 |   |   |  |   |
| <b>- Luồng sự kiện chính (main flow):</b>   |   |                 |   |   |  |   |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;"><b>Khách hàng</b></th> <th style="text-align: center; padding: 5px;"><b>Hệ thống</b></th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">1. Khách hàng chọn vào mục “Khuyến mãi” trên giao diện Website.</td> <td style="padding: 5px;">2. Hệ thống hiển thị danh sách khuyến mãi</td> </tr> <tr> <td style="padding: 5px;">3. Khách hàng chọn một sự kiện khuyến mãi muốn xem</td> <td style="padding: 5px;">4. Hệ thống hiển thị thông tin chi tiết sự kiện khuyến mãi mà khách hàng chọn</td> </tr> </tbody> </table> | <b>Khách hàng</b>   | <b>Hệ thống</b> | 1. Khách hàng chọn vào mục “Khuyến mãi” trên giao diện Website. | 2. Hệ thống hiển thị danh sách khuyến mãi | 3. Khách hàng chọn một sự kiện khuyến mãi muốn xem | 4. Hệ thống hiển thị thông tin chi tiết sự kiện khuyến mãi mà khách hàng chọn |
| <b>Khách hàng</b>   | <b>Hệ thống</b>   |                 |   |   |  |   |
| 1. Khách hàng chọn vào mục “Khuyến mãi” trên giao diện Website.   | 2. Hệ thống hiển thị danh sách khuyến mãi                                     |                 |   |   |  |   |
| 3. Khách hàng chọn một sự kiện khuyến mãi muốn xem  | 4. Hệ thống hiển thị thông tin chi tiết sự kiện khuyến mãi mà khách hàng chọn |                 |   |   |  |   |
| <b>- Luồng sự kiện thay thế (alternate flow):</b>   |   |                 |   |   |  |   |
| 2.1. Hệ thống hiển thị “Hiện tại chưa có sự kiện khuyến mãi nào”  |   |                 |   |   |  |   |
| <b>- Luồng sự kiện ngoại lệ (exception flow):</b>   |   |                 |   |   |  |   |
| Không   |   |                 |   |   |  |   |

**- Use case “Xem bài viết”:**

Bảng 3.4. Đặc tả Use case “Xem bài viết”

|   |
|---|
| <b>- Tên use case:</b> Xem bài viết   |
| <b>- Mô tả sơ lược:</b> Chức năng giúp khách hàng xem các bài viết trên Website |
| <b>- Actor chính:</b> Khách hàng  |

| - <b>Actor phụ:</b> Không   |   |                 |   |   |  |   |
|---|---|-----------------|---|---|--|---|
| - <b>Tiền điều kiện (Pre-condition):</b> Kết nối Internet   |   |                 |   |   |  |   |
| - <b>Hậu điều kiện (Post-condition):</b> Không  |   |                 |   |   |  |   |
| - <b>Luồng sự kiện chính (main flow):</b>   |   |                 |   |   |  |   |
| <table border="1"> <thead> <tr> <th><b>Khách hàng</b></th><th><b>Hệ thống</b></th></tr> </thead> <tbody> <tr> <td>1. Khách hàng chọn vào mục “Bài viết” trên giao diện Website.</td><td>2. Hệ thống hiển thị danh sách bài viết</td></tr> <tr> <td>3. Khách hàng chọn một bài viết muốn xem</td><td>4. Hệ thống hiển thị thông tin chi tiết của bài viết đó</td></tr> </tbody> </table> | <b>Khách hàng</b>                                       | <b>Hệ thống</b> | 1. Khách hàng chọn vào mục “Bài viết” trên giao diện Website. | 2. Hệ thống hiển thị danh sách bài viết | 3. Khách hàng chọn một bài viết muốn xem | 4. Hệ thống hiển thị thông tin chi tiết của bài viết đó |
| <b>Khách hàng</b>   | <b>Hệ thống</b>   |                 |   |   |  |   |
| 1. Khách hàng chọn vào mục “Bài viết” trên giao diện Website.   | 2. Hệ thống hiển thị danh sách bài viết                 |                 |   |   |  |   |
| 3. Khách hàng chọn một bài viết muốn xem  | 4. Hệ thống hiển thị thông tin chi tiết của bài viết đó |                 |   |   |  |   |
| - <b>Luồng sự kiện thay thế (alternate flow):</b>   |   |                 |   |   |  |   |
| 2.1. Hệ thống hiển thị “Hiện tại chưa có bài viết nào”  |   |                 |   |   |  |   |
| - <b>Luồng sự kiện ngoại lệ (exception flow):</b>   |   |                 |   |   |  |   |
| Không   |   |                 |   |   |  |   |

- **Use case “Hủy đơn đặt hàng”:**

Bảng 3.5. Đặc tả Use case “Hủy đơn đặt hàng”

|   |
|---|
| - <b>Tên use case:</b> Hủy đơn đặt hàng   |
| - <b>Mô tả sơ lược:</b> Chức năng giúp khách hàng có thể hủy đơn đặt hàng trong điều kiện cho phép          |
| - <b>Actor chính:</b> Khách hàng  |
| - <b>Actor phụ:</b> Không   |
| - <b>Tiền điều kiện (Pre-condition):</b> Đăng nhập thành công, đã có đơn hàng, kiểm tra trạng thái đơn hàng |

|  |  |
|--|--|
| <b>- Hậu điều kiện (Post-condition):</b> Không                                   |  |
| <b>- Luồng sự kiện chính (main flow):</b>  |  |
| <b>Khách hàng</b>  | <b>Hệ thống</b>  |
| 1. Khách hàng chọn vào đơn hàng muốn hủy và chọn nút “Hủy đơn hàng”.             | 2. Hệ thống kiểm tra trạng thái đơn hàng.                            |
|  | 3. Hệ thống hiển thị thông báo “Bạn có muốn chắc chắn hủy đơn hàng?” |
| 4. Khách hàng chọn “Đồng ý” để xác nhận  | 5. Hệ thống thông báo “Hủy đơn hàng thành công”                      |
| <b>- Luồng sự kiện thay thế (alternate flow):</b>                                |  |
| 3.1. Hệ thống hiển thị “Hiện tại đơn hàng đã được vận chuyển. Không thể hủy đơn” |  |
| 3.2. Khách hàng nhấn nút “Ok” để xác nhận  |  |
| <b>- Luồng sự kiện ngoại lệ (exception flow):</b>                                |  |
| Không  |  |

**- Use case “Tìm kiếm sản phẩm”:**

Bảng 3.6. Đặc tả Use case “Tìm kiếm sản phẩm”

|   |
|---|
| <b>- Tên use case:</b> Tìm kiếm sản phẩm  |
| <b>- Mô tả sơ lược:</b> Chức năng tìm kiếm giúp khách hàng có thể tìm kiếm sản phẩm phù hợp với sở thích. |
| <b>- Actor chính:</b> Khách hàng.   |
| <b>- Actor phụ:</b> Không   |
| <b>- Tiền điều kiện (Pre-condition):</b> Thiết bị của khách hàng kết nối Internet thành công              |

|  |   |
|--|---|
| <p><b>- Hậu điều kiện (Post-condition):</b> Sau khi khách hàng nhập từ khóa tìm kiếm, giao diện sẽ cho ra kết quả các sản phẩm phù hợp với từ khóa đã được nhập.</p>   |   |
| <p><b>- Luồng sự kiện chính (main flow):</b></p>   |   |
| <b>Actor</b>   | <b>System</b>   |
| 1. Khách hàng click chọn và nhập từ khóa vào ô tìm kiếm.   | 2. Hệ thống kiểm tra các sản phẩm hiện có dựa trên dữ liệu nhập                             |
|  | 3. Hệ thống hiển thị các sản phẩm tương ứng.  |
| 4. Khách hàng lựa chọn 1 sản phẩm.   | 4. Hệ thống hiển thị thông tin chi tiết bao gồm: tên sản phẩm, thành phần, hình ảnh, giá... |
| <p><b>- Luồng sự kiện thay thế (alternate flow):</b></p>   |   |
| <p>1.1. Hệ thống hiển thị thông báo từ khóa không phù hợp hoặc sản phẩm không tìm thấy.</p> <ol style="list-style-type: none"> <li>1. Khách hàng click nút trở lại.</li> <li>2. Hệ thống quay lại bước 1.</li> </ol> |   |

#### - Use case “Xóa sản phẩm trong giỏ hàng”:

Bảng 3.7. Đặc tả Use case “Xóa sản phẩm trong giỏ hàng”

|  |
|--|
| <p><b>- Tên use case:</b> Xóa sản phẩm trong giỏ hàng</p>  |
| <p><b>- Mô tả sơ lược:</b> Chức năng “Xóa sản phẩm trong giỏ hàng” giúp khách hàng có thể xóa sản phẩm mình không thích trong giỏ hàng hiện tại trên hệ thống</p>  |
| <p><b>- Actor chính:</b> Khách hàng</p>  |
| <p><b>- Actor phụ:</b> Không</p>   |
| <p><b>- Tiền điều kiện (Pre-condition):</b> Đang ở trang giỏ hàng</p>  |
| <p><b>- Hậu điều kiện (Post-condition):</b> Nếu xóa sản phẩm thành công thì hệ thống hiển thị thông báo xóa thành công và lưu vào hệ thống, số lượng sản phẩm đó lên 1 hoặc nhiều tùy vào lúc khách hàng đã thêm vào giỏ hàng sản phẩm đó với số lượng một hoặc nhiều.</p> |

|   |   |
|---|---|
| <b>- Luồng sự kiện chính (main flow):</b>   |   |
| <b>Khách hàng</b>   | <b>Hệ thống</b>   |
| 1. Chọn 1 sản phẩm muốn xóa trong giỏ hàng và nhấn vào chức năng <b>Xóa sản phẩm</b>  | 2. Kiểm tra xem sản phẩm đó đã được thêm vào giỏ hàng cách đây trong bao lâu. |
|   | 3. Hiển thị thông báo xóa sản phẩm khỏi giỏ hàng thành công.                  |
| 4. Click chọn nút xác nhận.   |   |
| <b>- Luồng sự kiện thay thế (alternate flow):</b>   |   |
| 1.1. Hệ thống hiển thị thông báo “Xóa sản phẩm không thành công. Sản phẩm này đã được hệ thống tự động xóa khỏi giỏ hàng trước đó do quá thời gian quy định mà sản phẩm này vẫn chưa được đặt.” |   |
| 1. Khách hàng xác nhận.   |   |
| 2. Hệ thống quay lại bước 1   |   |
| <b>- Luồng sự kiện ngoại lệ (exception flow): Không.</b>  |   |

**- Use case “Duyệt đơn hàng”:**

Bảng 3.8. Đặc tả Use case “Duyệt đơn hàng”

|  |
|--|
| <b>Use case:</b> Duyệt đơn hàng  |
| <b>Mục đích:</b> Đơn hàng của khách hàng được nhân viên quản lý xác nhận việc đặt hàng.  |
| <b>Mô tả:</b> chức năng duyệt đơn hàng cho phép nhân viên quản lý duyệt đơn đặt hàng của khách hàng sau khi kiểm tra đơn đặt hàng và đơn đặt hàng của khách hàng được xác nhận thành công. |
| <b>Tác nhân:</b> khách hàng, nhân viên quản lý.  |
| <b>Tiền điều kiện (Pre-condition):</b> đăng nhập thành công.   |

|   |   |
|---|---|
| <b>Hậu điều kiện (Post-condition):</b> nếu nhân viên quản lý duyệt đơn hàng thành công thì sẽ có thông báo tiếp tục thanh toán đến khách hàng và được lưu vào hệ thống. |   |
| <b>Luồng sự kiện chính (main flows):</b>  |   |
| <b>Actor</b>  | <b>System</b>   |
|   | 1. Hệ thống gửi yêu cầu duyệt đơn đến giao diện của nhân viên quản lý.  |
| 2. Nhân viên chọn chức năng duyệt đơn hàng.   | 3. Hệ thống hiển thị danh sách đơn đặt hàng đang chờ duyệt.             |
| 4. Nhân viên chọn một đơn hàng muốn duyệt.  | 5. Hệ thống hiển thị đơn đặt hàng gồm các sản phẩm khách hàng đã chọn.  |
| 6. Nhân viên click chọn nút duyệt đơn.  | 7. Hệ thống hiển thị thông báo duyệt đơn thành công.                    |
|   | 8. Hệ thống gửi thông báo mời chọn hình thức thanh toán cho khách hàng. |
| <b>Luồng sự kiện thay thế (alternate flows):</b>  |   |
| 8.1. Nếu số lượng sản phẩm nhiều, suy ra giá tổng đơn nếu trên 500k thì nhân viên chọn gửi yêu cầu thanh toán online 50% cho khách hàng.                                |   |
| 8.2. Hệ thống gửi yêu cầu thanh toán online cho khách hàng.   |   |
| <b>Luồng sự kiện ngoại lệ (exception flows):</b>  |   |
| Không có  |   |

- Use case “Thanh toán”:

Bảng 3.9. Đặc tả Use case “Thanh toán”

|   |
|---|
| <b>Use case:</b> Thanh toán   |
| <b>Mục đích:</b> Đơn đặt hàng của khách hàng được thanh toán và hóa đơn được tạo.             |
| <b>Mô tả:</b> Chức năng thanh toán cho phép khách hàng thanh toán đơn đặt hàng và in hóa đơn. |

| <b>Tác nhân:</b> Khách hàng.  |   |
|---|---|
| <b>Tiền điều kiện (Pre-condition):</b> Đăng nhập thành công   |   |
| <b>Hậu điều kiện (Post-condition):</b> Nếu khách hàng thanh toán thành công thì hóa đơn sẽ được in và đơn đặt hàng được tạo thành công, đơn hàng được lưu vào hệ thống. |   |
| <b>Luồng sự kiện chính (main flows):</b>  |   |
| Actor   | System  |
| 1. Khách hàng chọn hình thức thanh toán.  | 2. Hệ thống hiển thị 2 hình thức thanh toán: thanh toán online và thanh toán trực tiếp. |
| 3. Khách hàng chọn thanh toán trực tiếp cho shipper.  | 4. Hệ thống hiển thị thông báo thanh toán thành công và in hóa đơn cho khách hàng.      |
| <b>Luồng sự kiện thay thế (alternate flows):</b>  |   |
| 3.1. Khách hàng chọn thanh toán online.<br>1. Hệ thống chuyển sang cổng thanh toán của ngân hàng.<br>2. Hệ thống hiển thị thanh toán thành công, in hóa đơn.            |   |

- **Use case “Đăng ký”:**

Bảng 3.10. Đặc tả Use case “Đăng ký”

|   |
|---|
| <b>Use case:</b> Đăng ký  |
| <b>Mục đích:</b> Tạo tài khoản riêng cho user.                            |
| <b>Mô tả:</b> chức năng đăng ký giúp user đăng ký tài khoản trên website. |
| <b>Tác nhân:</b> Khách hàng, quản lý cửa hàng.                            |
| <b>Tiền điều kiện (Pre-condition):</b> Không có                           |

**Hậu điều kiện (Post-condition):** Nếu khách hàng đăng ký thành công thì có thẻ đăng nhập để Bán Laptop, quản lý chuỗi cửa hàng đăng ký tài khoản cho nhân viên và có thẻ phân quyền cho nhân viên.

### Luồng sự kiện chính (main flows):

| Actor   | System   |
|---|--|
| 1. User chọn chức năng đăng ký                  | 2. Hệ thống hiển thị form đăng ký tài khoản.       |
| 3. User điền đầy đủ thông tin vào form đăng ký. |  |
| 4. User chọn nút đăng ký.                       | 5. Hệ thống hiển thị thông báo đăng ký thành công. |

### Luồng sự kiện thay thế (alternate flows):

- 3.1. Hệ thống hiển thị thông báo thông tin cần điền chưa đầy đủ và mời user thêm vào.
  - 1. User điền đầy đủ thông tin theo yêu cầu.
  - 2. Hệ thống hiển thị đăng ký thành công.

#### - Use case “Đăng nhập”:

Bảng 3.11. Đặc tả Use case “Đăng nhập”

|   |
|---|
| <b>Use case:</b> Đăng nhập  |
| <b>Mục đích:</b> User đăng nhập vào hệ thống.   |
| <b>Mô tả:</b> Chức năng này giúp các user đăng nhập vào hệ thống để có thể thực hiện các nghiệp vụ. |
| <b>Tác nhân:</b> Khách hàng, quản lý cửa hàng, nhân viên quản trị, nhân viên giao hàng.             |
| <b>Tiền điều kiện (Pre-condition):</b> User đã có tài khoản được lưu trong hệ thống                 |

**Hậu điều kiện (Post-condition):** Nếu user đăng nhập thành công thì có thể thực hiện các nghiệp vụ trên hệ thống.

#### Luồng sự kiện chính (main flows):

| Actor                               | System   |
|-------------------------------------|--|
| 1. User chọn chức năng đăng nhập.   | 2. Hệ thống hiển thị form đăng nhập.                 |
| 3. User điền tài khoản và password. |  |
| 4. User chọn nút đăng nhập.         | 5. Hệ thống hiển thị thông báo đăng nhập thành công. |

#### Luồng sự kiện thay thế (alternate flows):

- 3.1. Hệ thống hiển thị thông báo tài khoản hoặc pass sai.
- 3.2. User điền lại thông tin sai.
- 3.3. Hệ thống hiển thị đăng nhập thành công.

#### Luồng sự kiện ngoại lệ (exception flows):

- 3.4. Hệ thống hiển thị thông báo thông tin cần nhập bị thiếu.
- 3.5. User điền đầy đủ thông tin cần nhập.
- 3.6. Hệ thống hiển thị đăng nhập thành công.

#### - Use case “Lập báo cáo thống kê”:

Bảng 3.12. Đặc tả Use case “Lập báo cáo thống kê”

|   |
|---|
| <b>Use case:</b> Lập báo cáo thống kê   |
| <b>Mục đích:</b> Thống kê và báo cáo mỗi ngày.  |
| <b>Mô tả:</b> Chức năng giúp nhân viên giao hàng thống kê số đơn hàng đã giao được và doanh thu mỗi ngày để báo cáo cho quản lý cửa hàng. |
| <b>Tác nhân:</b> Nhân viên giao hàng.   |
| <b>Tiền điều kiện (Pre-condition):</b> Đăng nhập thành công   |

**Hậu điều kiện (Post-condition):** Nhân viên giao hàng báo cáo cho quản lý cửa hàng thành công.

#### Luồng sự kiện chính (main flows):

| Actor   | System   |
|---|--|
| 1. Nhân viên giao hàng chọn chức năng lập thống kê báo cáo. | 2. Hệ thống hiển thị hai thông tin cần thống kê: đơn hàng và doanh thu.  |
| 3. Nhân viên chọn lập báo cáo.                              | 4. Hệ thống hiển thị mẫu báo cáo bao gồm thông tin chi tiết về các đơn hàng: Tên khách hàng, ngày giờ, sản phẩm, thành tiền. |
| 5. Nhân viên kiểm tra và chọn nộp báo cáo.                  | 6. Hệ thống gửi báo cáo đến user chủ cửa hàng kèm theo lưu báo cáo vào hệ thống và hiển thị báo cáo thành công.              |

#### Luồng sự kiện thay thế (alternate flows):

- 5.1. Nhân viên giao hàng muốn ghi chú vào báo cáo, chọn đơn hàng muốn ghi chú.
- 5.2. Hệ thống hiển thị thông tin chi tiết đơn hàng và ô ghi chú.
- 5.3. Nhân viên giao hàng điền vào ô ghi chú sau đó chọn submit.
- 5.4. Hệ thống hiển thị thông báo ghi chú thành công và quay về trang danh sách báo cáo.

#### - Use case “Tạo bài viết”:

Bảng 3.13. Đặc tả Use case “Tạo bài viết”

|   |
|---|
| <b>Use case:</b> Tạo bài viết.  |
| <b>Mục đích:</b> Tạo bài viết trên website.   |
| <b>Mô tả:</b> Chức năng giúp nhân viên quản lý cửa hàng tạo bài viết về sản phẩm của cửa hàng trên website. |
| <b>Tác nhân:</b> Nhân viên quản lý cửa hàng.  |
| <b>Tiền điều kiện (Pre-condition):</b> Đăng nhập thành công   |

**Hậu điều kiện (Post-condition):** Nhân viên bán hàng quản lý cửa hàng tạo được bài viết để hiển thị trên website.

#### Luồng sự kiện chính (main flows):

| Actor  | System  |
|--|---|
| 1. Nhân viên quản lý chọn chức năng tạo bài viết.            | 2. Hệ thống hiển thị form tạo bài viết.   |
| 3. Nhân viên điền thông tin muốn tạo trên bài viết lên form. |   |
| 4. Nhân viên chọn nút tạo bài viết.                          | 5. Hệ thống thông báo tạo bài viết thành công và hiển thị bài viết lên trang của website. |

#### - Use case “Sửa bài viết”:

Bảng 3.14. Đặc tả Use case “Sửa bài viết”

**Use case:** Sửa bài viết.

**Mục đích:** Sửa bài viết trên website.

**Mô tả:** Chức năng giúp nhân viên quản lý cửa hàng sửa bài viết về sản phẩm của cửa hàng trên website.

**Tác nhân:** Nhân viên quản lý cửa hàng.

**Tiền điều kiện (Pre-condition):** Đăng nhập thành công

**Hậu điều kiện (Post-condition):** Nhân viên bán hàng quản lý cửa hàng sửa được bài viết để hiển thị trên website.

#### Luồng sự kiện chính (main flows):

| Actor   | System                                   |
|---|--|
| 1. Nhân viên quản lý chọn chức năng bài viết. | 2. Hệ thống hiển thị danh sách bài viết. |

|  |   |
|--|---|
| 3. Nhân viên chọn một bài viết muốn sửa. | 4. Hệ thống hiển thị form bài viết đã tạo.  |
| 5. Nhân viên sửa thông tin trên form.    |   |
| 6. Nhân viên chọn submit.                | 7. Hệ thống thông báo sửa bài viết thành công và hiển thị bài viết lên trang của website. |

**- Use case “Xóa bài viết”:**

Bảng 3.15. Đặc tả Use case “Xóa bài viết”

| <b>Use case:</b> Xóa bài viết.   |  |
|--|--|
| <b>Mục đích:</b> Xóa bài viết trên website.  |  |
| <b>Mô tả:</b> Chức năng giúp nhân viên quản lý cửa hàng xóa bài viết về sản phẩm của cửa hàng trên website.            |  |
| <b>Tác nhân:</b> Nhân viên quản lý cửa hàng.   |  |
| <b>Tiền điều kiện (Pre-condition):</b> Đăng nhập thành công  |  |
| <b>Hậu điều kiện (Post-condition):</b> Nhân viên bán hàng quản lý cửa hàng xóa được bài viết để hiển thị trên website. |  |
| <b>Luồng sự kiện chính (main flows):</b>   |  |
| Actor  | System   |
| 1. Nhân viên quản lý chọn chức năng bài viết.  | 2. Hệ thống hiển thị danh sách bài viết.   |
| 3. Nhân viên chọn một bài viết muốn xóa.   | 4. Hệ thống hiển thị thông báo xác nhận.   |
| 5. Nhân viên xác nhận xóa bài viết.  | 6. Hệ thống thông báo xóa bài viết thành công và xóa hiển thị bài viết trên trang của website. |

**Luồng sự kiện thay thế (alternate flows):**

- 5.1. Nhân viên không muốn xóa bài viết, chọn cancel.
- 5.2. Hệ thống quay lại danh sách các bài viết.

**- Use case “Thêm vào giỏ hàng”:***Bảng 3.16. Đặc tả Use case “Thêm vào giỏ hàng”*

| <b>Use case:</b> Thêm vào giỏ hàng.   |   |
|---|---|
| <b>Mục đích:</b> Thêm sản phẩm vào giỏ hàng.  |   |
| <b>Mô tả:</b> Chức năng giúp khách hàng thêm sản phẩm muốn mua vào giỏ hàng.  |   |
| <b>Tác nhân:</b> Khách hàng.  |   |
| <b>Tiền điều kiện (Pre-condition):</b> Đăng nhập thành công   |   |
| <b>Hậu điều kiện (Post-condition):</b> Sản phẩm được thêm vào giỏ hàng của khách hàng, sau đó hệ thống cập nhật số lượng tồn trong thông kê của cửa hàng. |   |
| <b>Luồng sự kiện chính (main flows):</b>  |   |
| Actor   | System  |
| 1. Khách hàng chọn sản phẩm muốn mua.   | 2. Hệ thống hiển thị thông tin chi tiết sản phẩm.                                       |
| 3. Khách hàng click chọn thêm vào giỏ hàng.   | 4. Hệ thống hiển thị giao diện chọn số lượng sản phẩm.                                  |
| 5. Khách hàng chọn số lượng sản phẩm.   |   |
| 6. Khách hàng chọn submit.  | 7. Hệ thống hiển thị thông báo thêm vào giỏ hàng thành công, cập nhật lại số lượng tồn. |

### Luồng sự kiện thay thế (alternate flows):

- 6.1. Hệ thống hiển thị thông báo khách hàng chưa chọn số lượng và mời chọn số lượng.
- 6.2. Khách hàng chọn số lượng sản phẩm và ấn chọn submit.
- 6.3. Hệ thống hiển thị thông báo thêm vào giỏ hàng thành công.

#### - Use case “Xóa khỏi giỏ hàng”:

Bảng 3.17. Đặc tả Use case “Xóa khỏi giỏ hàng”

**Use case:** Xóa khỏi giỏ hàng.

**Mục đích:** Xóa sản phẩm khỏi giỏ hàng.

**Mô tả:** Chức năng giúp khách hàng xóa sản phẩm không còn muốn mua khỏi giỏ hàng.

**Tác nhân:** khách hàng.

**Tiền điều kiện (Pre-condition):** User chọn sản phẩm muốn xóa ra khỏi giỏ hàng.

**Hậu điều kiện (Post-condition):** sản phẩm được xóa khỏi giỏ hàng của khách hàng.

### Luồng sự kiện chính (main flows):

| Actor   | System  |
|---|---|
| 1. Khách hàng chọn chức năng giỏ hàng.                    | 2. Hệ thống hiển thị thông tin giỏ hàng.                              |
| 3. Khách hàng click chọn sản phẩm muốn xóa khỏi giỏ hàng. |   |
| 4. Khách hàng click chọn nút xóa sản phẩm.                | 5. Hệ thống hiển thị thông báo xác nhận.                              |
| 6. Khách hàng xác nhận.                                   | 7. Hệ thống hiển thị thông báo xóa sản phẩm khỏi giỏ hàng thành công. |

**Luồng sự kiện thay thế (alternate flows):**

- 5.1. Nhân viên không muốn xóa sản phẩm, chọn cancel.
- 5.2. Hệ thống quay lại giỏ hàng.

**- Use case “Thêm sản phẩm”:***Bảng 3.18. ĐẶC TẢ USE CASE “THÊM SẢN PHẨM”*

| <b>Tên use case:</b> Thêm sản phẩm   |   |
|--|---|
| <b>Mô tả sơ lược:</b> Cho phép người dùng được cấp quyền thêm các thông tin, hình ảnh sản phẩm                                   |   |
| <b>Actor chính:</b> Quản lý cửa hàng   |   |
| <b>Actor phụ:</b> Không  |   |
| <b>Tiền điều kiện (Pre-condition):</b> Đăng nhập thành công  |   |
| <b>Hậu điều kiện (Post-condition):</b> Nếu thêm sản phẩm thành công thì sẽ được lưu vào hệ thống và cập nhật trên trang thực đơn |   |
| <b>Luồng sự kiện chính (main flow):</b>  |   |
| Quản lý chuỗi cửa hàng   | Hệ thống  |
| 1. Chọn chức năng <b>Thêm sản phẩm</b>   | 2. Hiển thị form điền thông tin sản phẩm          |
| 3. Điền thông tin sản phẩm vào form  |   |
| 4. Click vào nút thêm hình ảnh   | 5. Hiển thị các ảnh đã lưu sẵn trong máy          |
| 6. Chọn ảnh và click tải ảnh lên   | 7. Hiển thị form thông báo tải ảnh lên thành công |
| 8. Click vào nút Thêm  | 9. Hệ thống thông báo thêm sản phẩm thành công    |

### - Luồng sự kiện thay thế (alternate flow):

1.1. Hệ thống thông báo tải sản phẩm lên thất bại do chưa điền đầy đủ thông tin

1. Quản lý bấm xác nhận

2. Hệ thống quay lại bước 3

1.2. Hệ thống thông báo tải sản phẩm thất bại do chưa thêm hình ảnh

1. Quản lý bấm xác nhận

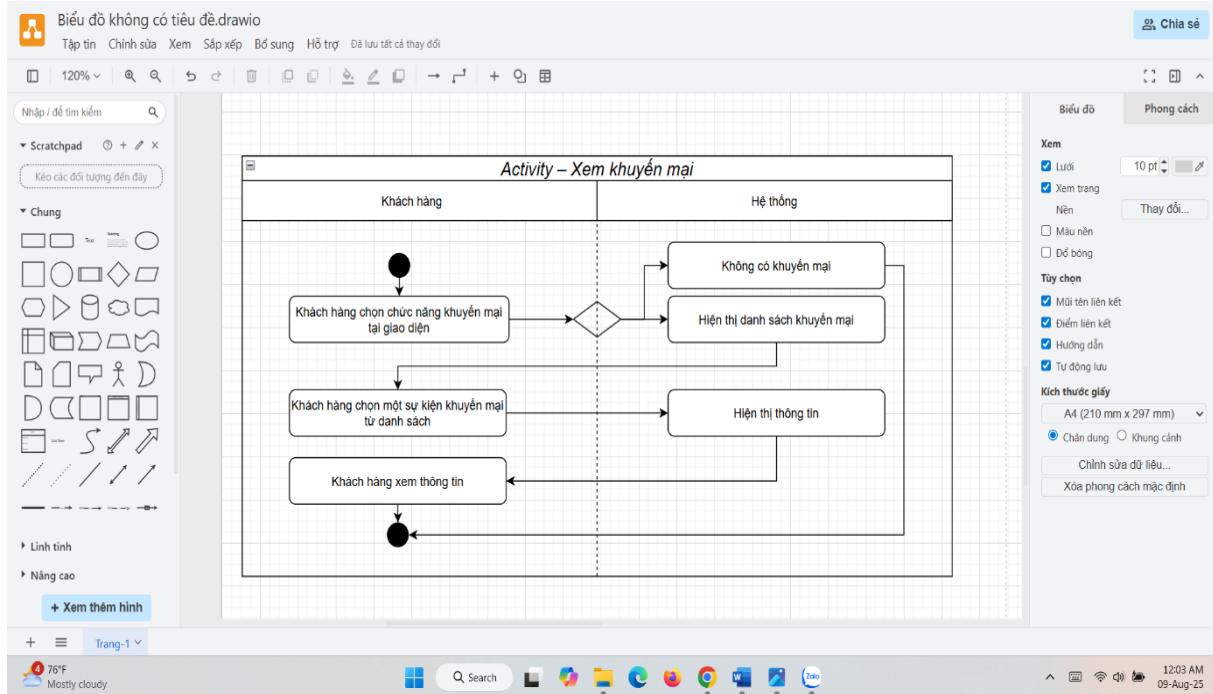
2. Hệ thống quay lại bước 4

### - Luồng sự kiện ngoại lệ (exception flow):

Không

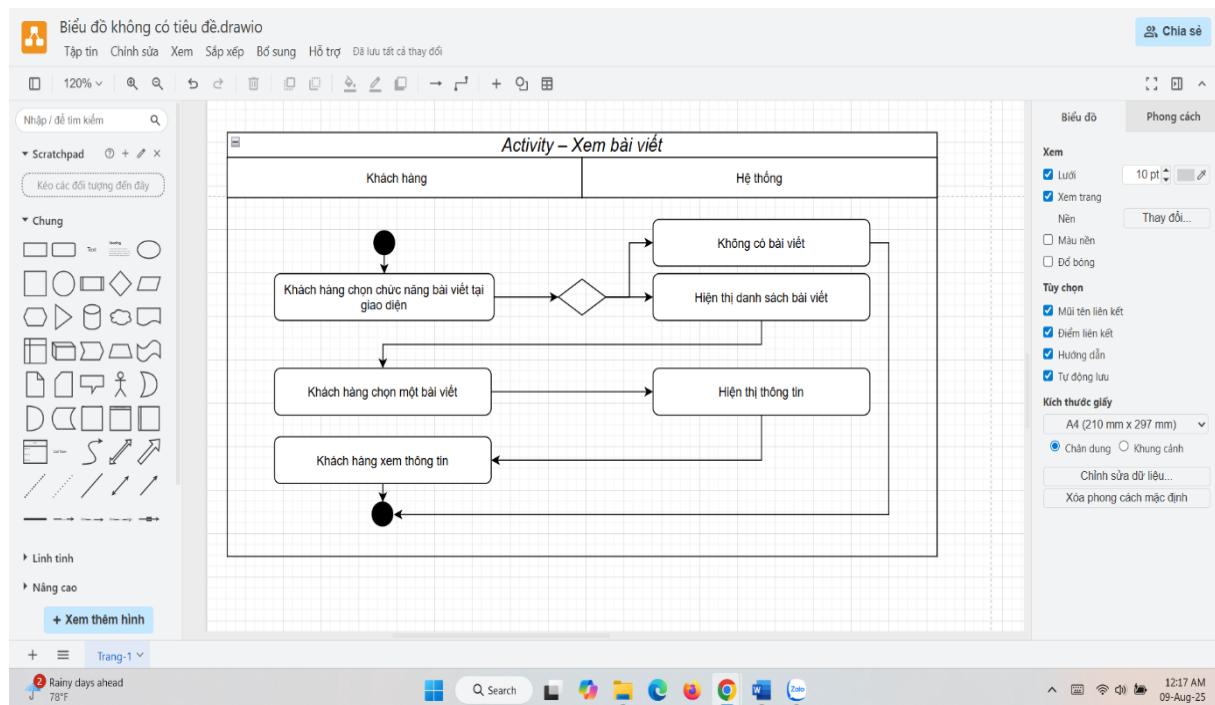
### 3.5. Sơ đồ Activity

**- Activity - Xem khuyến mại:** Khách hàng chọn vào 1 bài viết khuyến mại và xem thông tin khuyến mại.



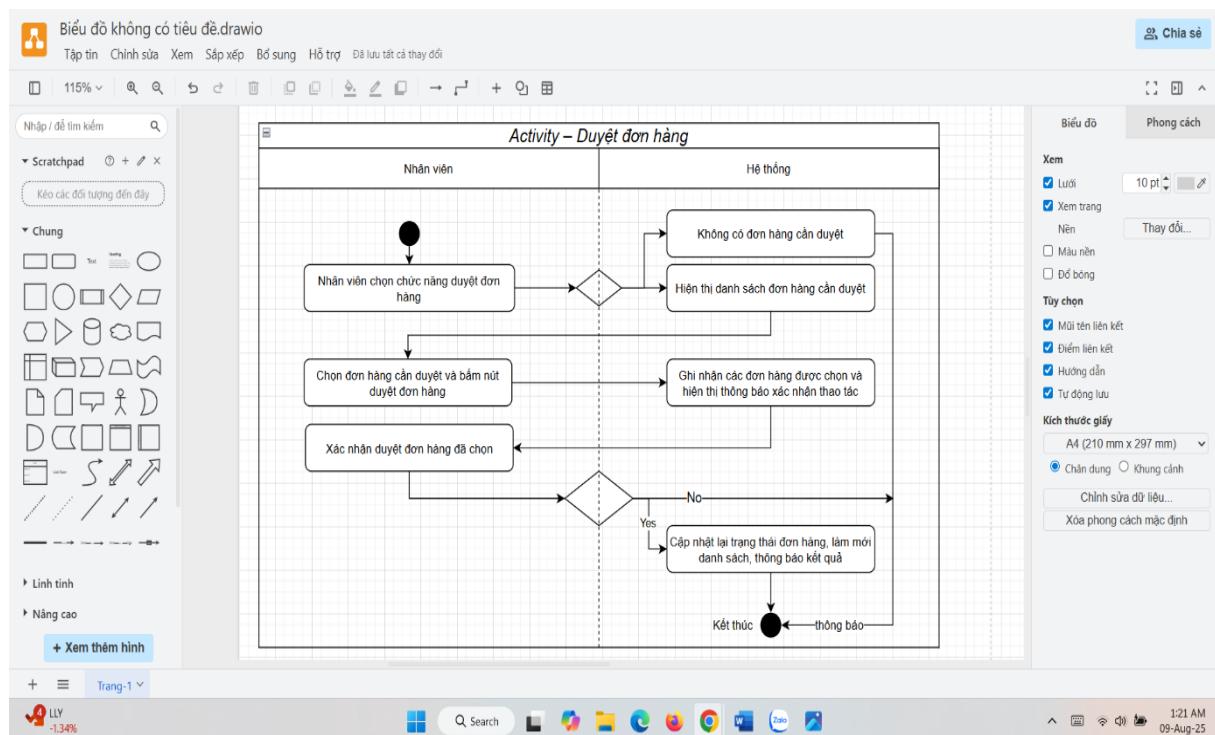
Hình 3.3. Sơ đồ Activity – Xem khuyến mại

**- Activity - Xem bài viết:** Khách hàng chọn vào 1 bài viết bất kỳ trong danh mục “Bài viết” của trang web



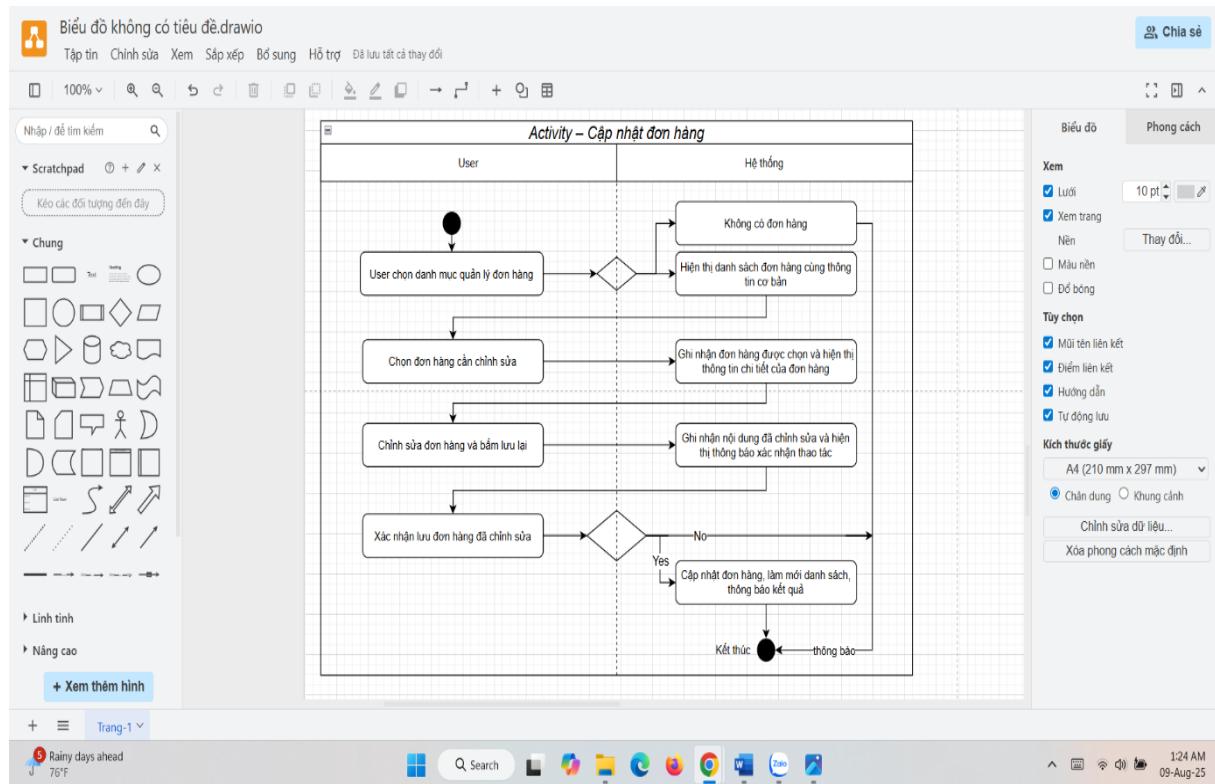
Hình 3.4. Sơ đồ Activity – Xem bài viết

**- Activity – Duyệt đơn hàng:** Nhân viên quản lý chọn đơn hàng cần duyệt và tiến hành duyệt đơn hàng



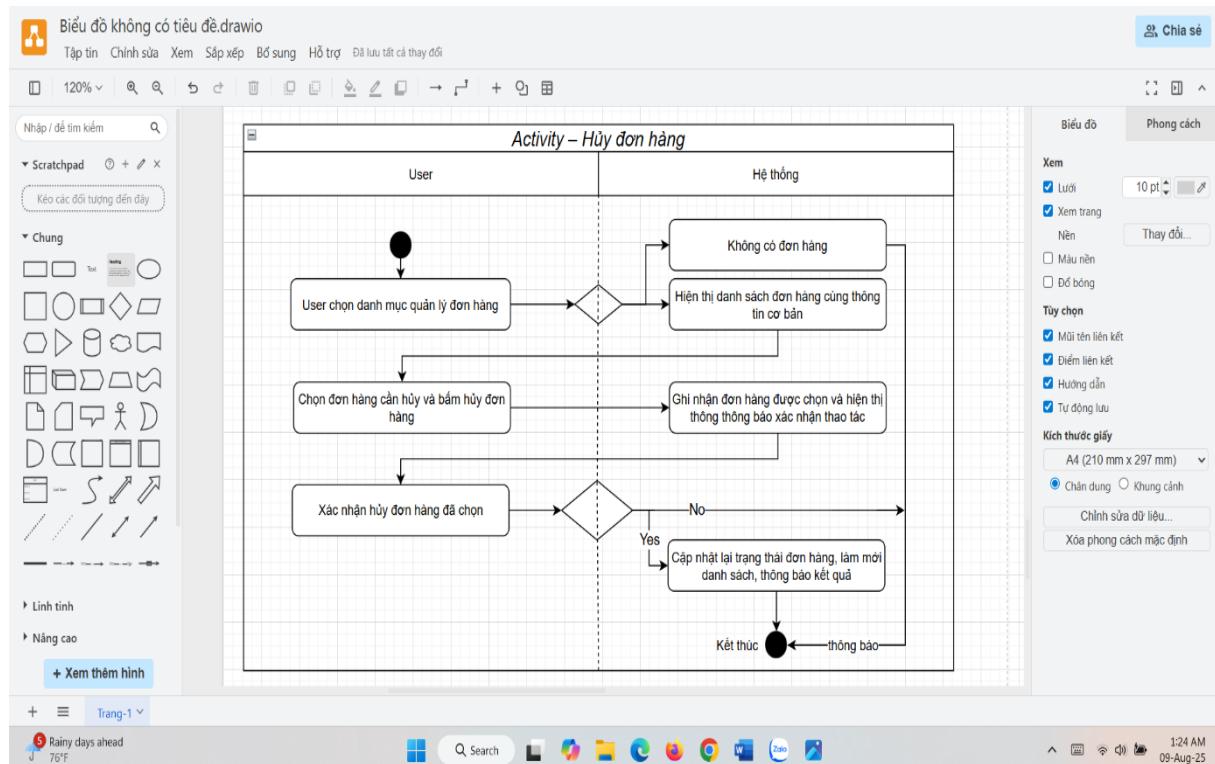
Hình 3.5. Sơ đồ Activity – Duyệt đơn hàng

**- Activity – Cập nhật đơn hàng:** Nhân viên quản lý hoặc khách hàng đã đăng nhập chọn đơn hàng cần chỉnh sửa và tiến hành cập nhật đơn hàng



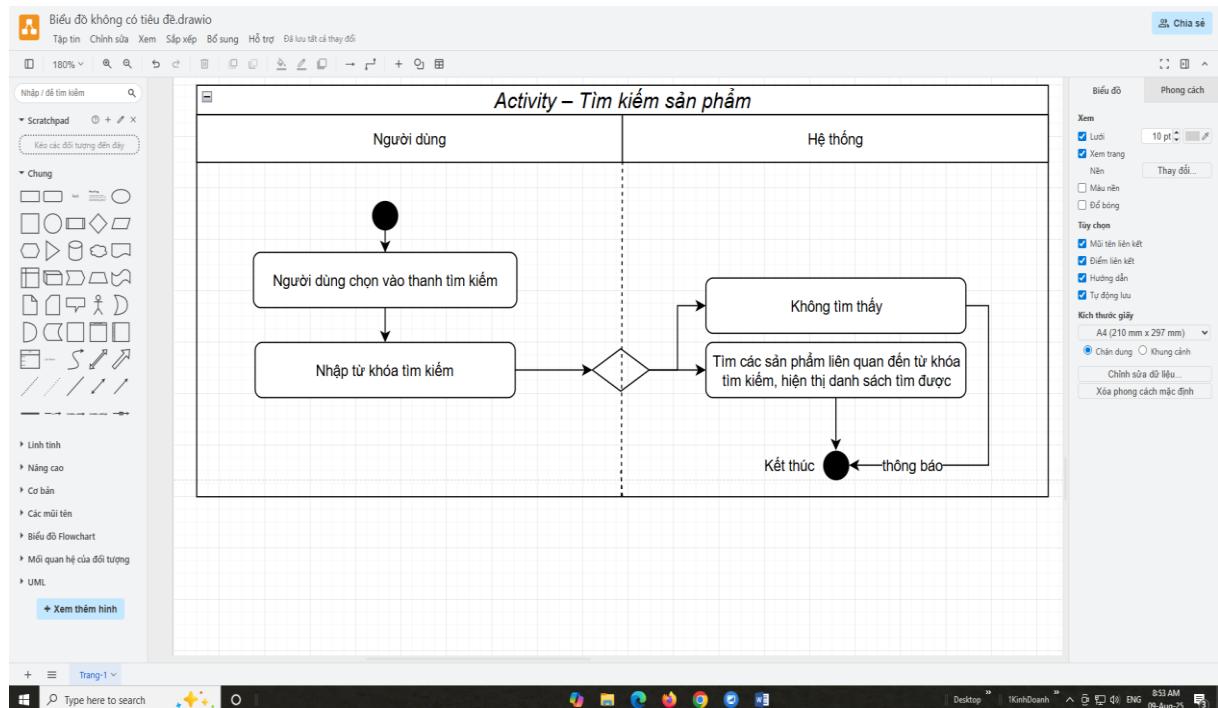
Hình 3.6. Sơ đồ Activity – Cập nhật đơn hàng

**- Activity – Hủy đơn hàng:** Nhân viên quản lý hoặc khách hàng đã đăng nhập chọn đơn hàng cần hủy và tiến hành hủy đơn hàng



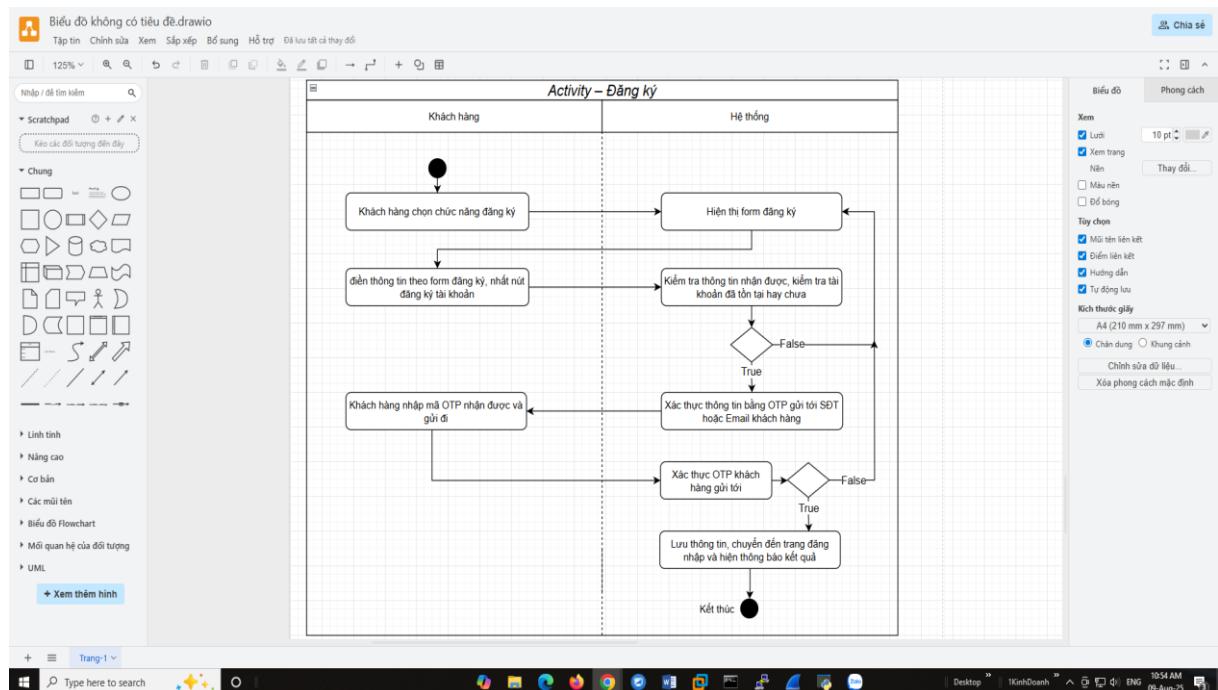
Hình 3.7. Sơ đồ Activity – Hủy đơn hàng

**- Activity – Tìm kiếm sản phẩm:** Khách hàng chọn thanh tìm kiếm và nhập từ khóa tìm kiếm, sau đó xem kết quả trả ra từ hệ thống.



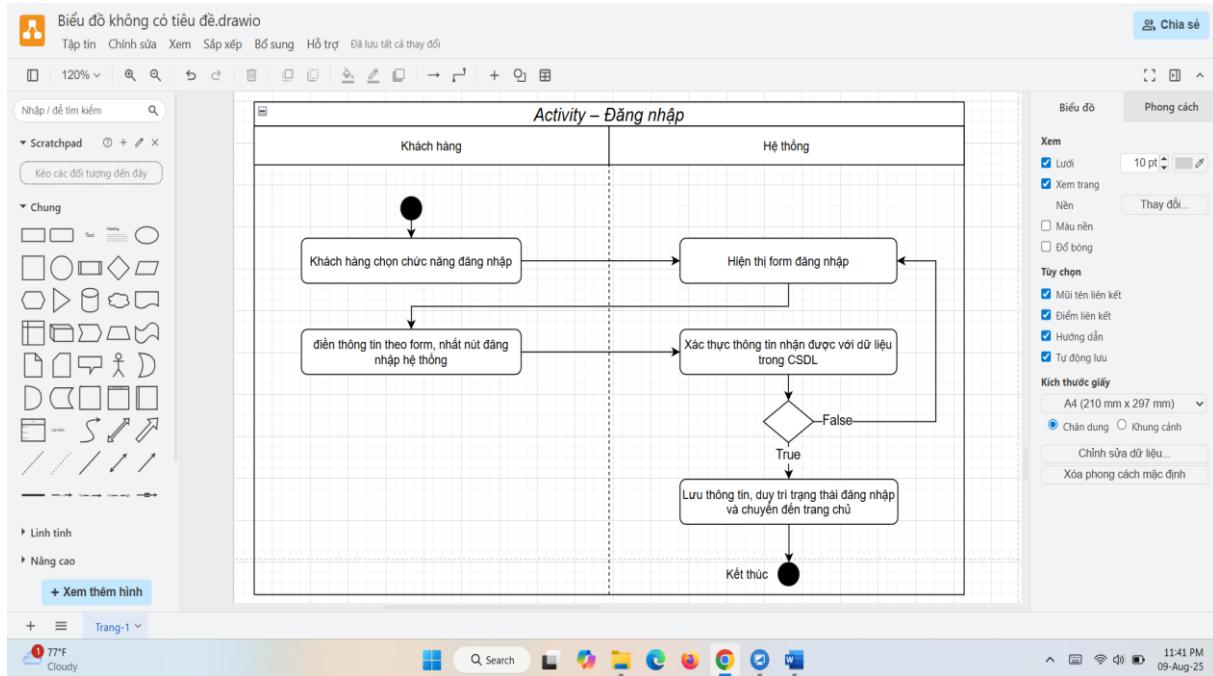
Hình 3.8. Sơ đồ Activity – Tìm kiếm sản phẩm

**- Activity – Đăng ký:** Khách hàng chọn trang đăng ký tài khoản khi chưa có tài khoản, khách hàng điền thông tin vào form đăng ký, hệ thống tiến hành kiểm tra thông tin được nhập và lưu lại nếu thỏa các điều kiện.



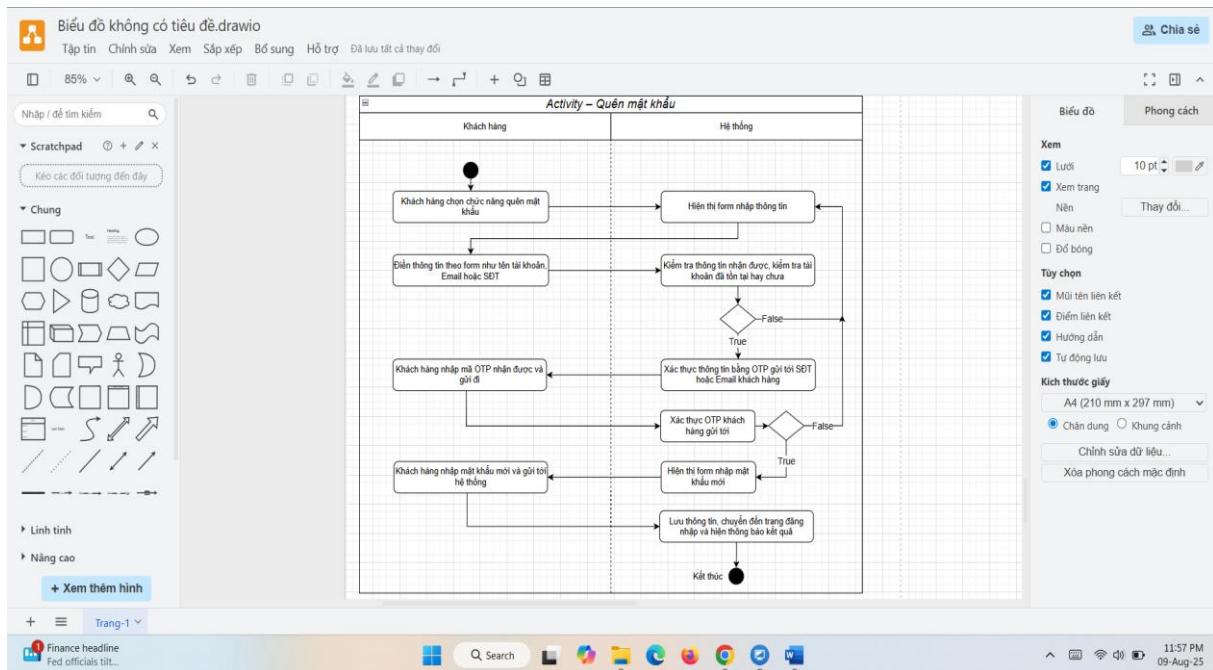
Hình 3.9. Sơ đồ Activity – Đăng ký

**- Activity – Đăng nhập:** Khách hàng chọn trang đăng nhập, nhập thông tin đăng nhập theo form đăng nhập, nếu thông tin đăng nhập không hợp lệ thì quay lại bước nhập thông tin, nếu thông tin hợp lệ thì hệ thống lưu thông tin và duy trì trạng thái đăng nhập.



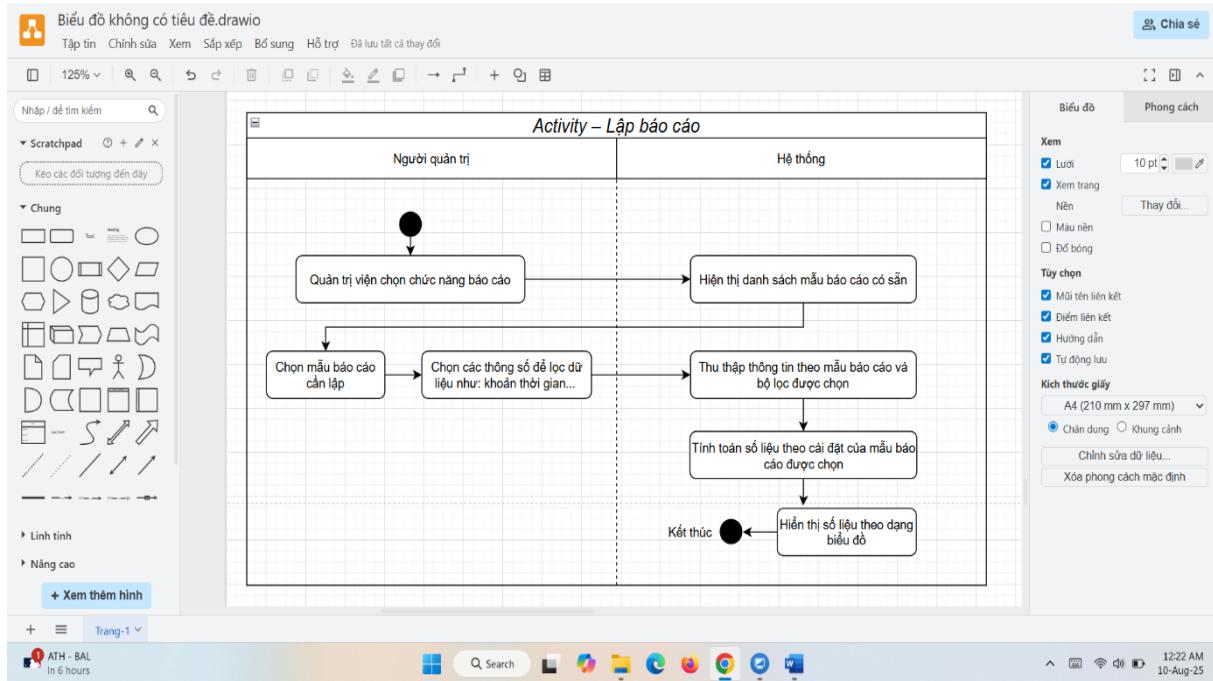
Hình 3.10. Sơ đồ Activity – Đăng nhập

**- Activity – Quên mật khẩu:** Khách hàng chọn chức năng quên mật khẩu, nhập thông tin theo form, nếu thông tin nhập không hợp lệ thì quay lại bước nhập thông tin, nếu thông tin hợp lệ thì hệ thống lưu thông tin và chuyển đến trang đăng nhập.



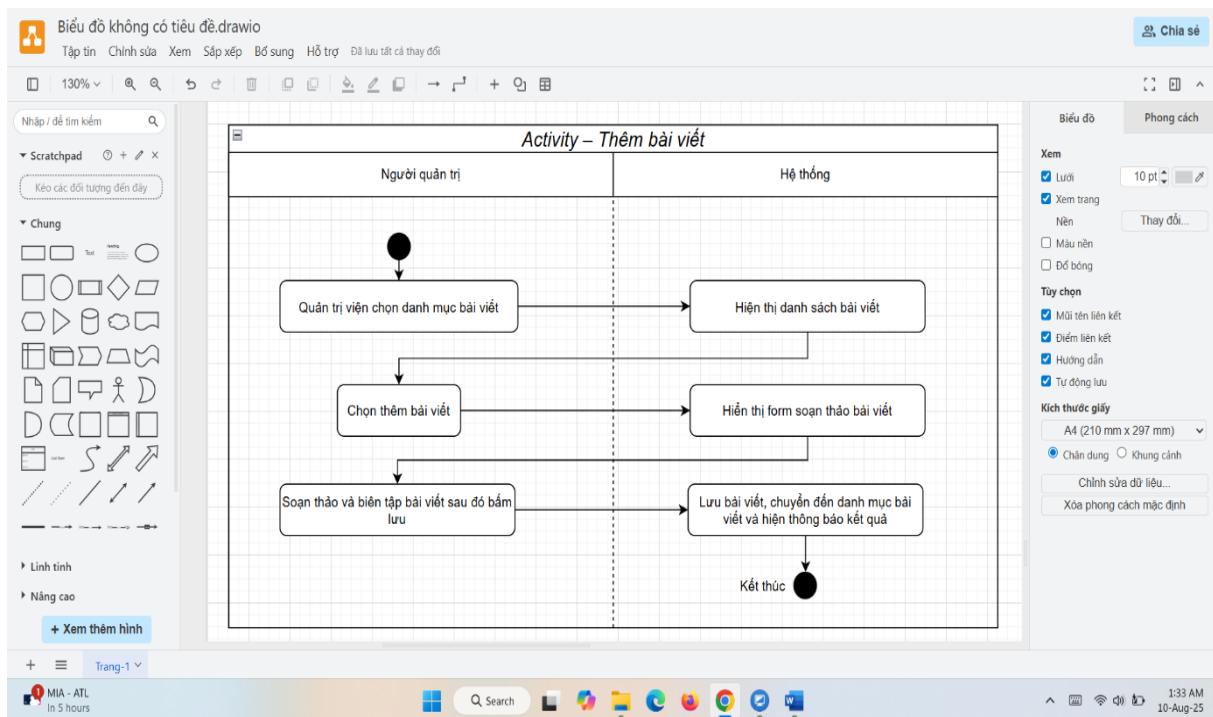
Hình 3.11. Sơ đồ Activity – Quên mật khẩu

**- Activity – Lập báo cáo, thống kê:** Quản trị viên chọn chức năng lập báo cáo, thống kê và chọn mẫu báo cáo cần lập. Hệ thống hiển thị số liệu theo cài đặt của mẫu báo cáo được chọn.



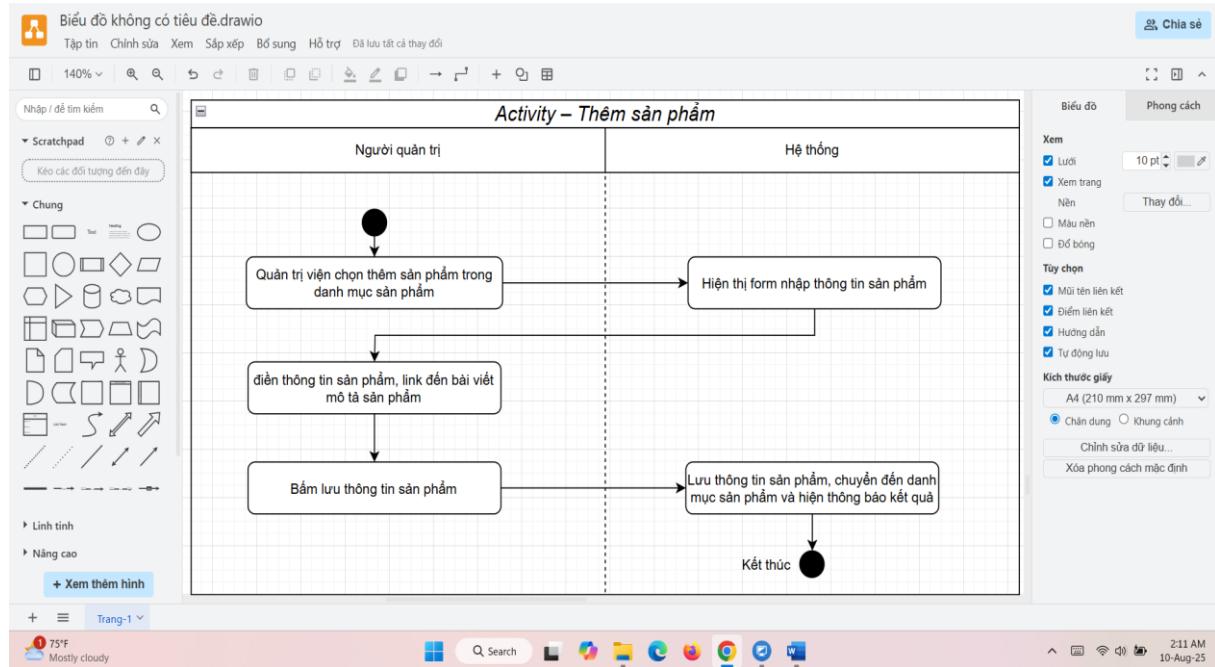
Hình 3.12. Sơ đồ Activity – lập báo cáo

**- Activity – Thêm bài viết:** Quản trị viên chọn thêm bài viết ở danh mục bài viết, tiến hành soạn thảo và biên tập bài viết. Hệ thống sẽ lưu trữ và hiển thị bài viết mới.



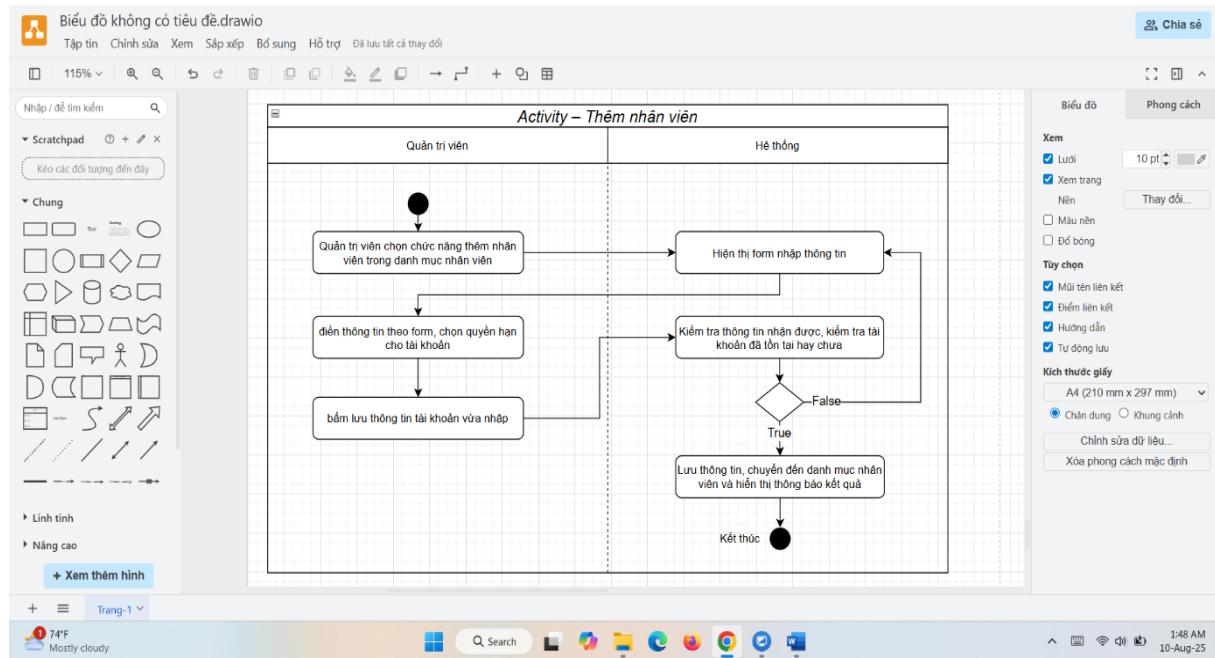
Hình 3.13. Sơ đồ Activity – thêm bài viết

**- Activity – Thêm sản phẩm:** Quản trị viên chọn chức năng thêm sản phẩm ở danh mục sản phẩm, tiến hành nhập thông tin của sản phẩm. Hệ thống sẽ lưu trữ và hiển thị sản phẩm.



Hình 3.14. Sơ đồ Activity – thêm sản phẩm

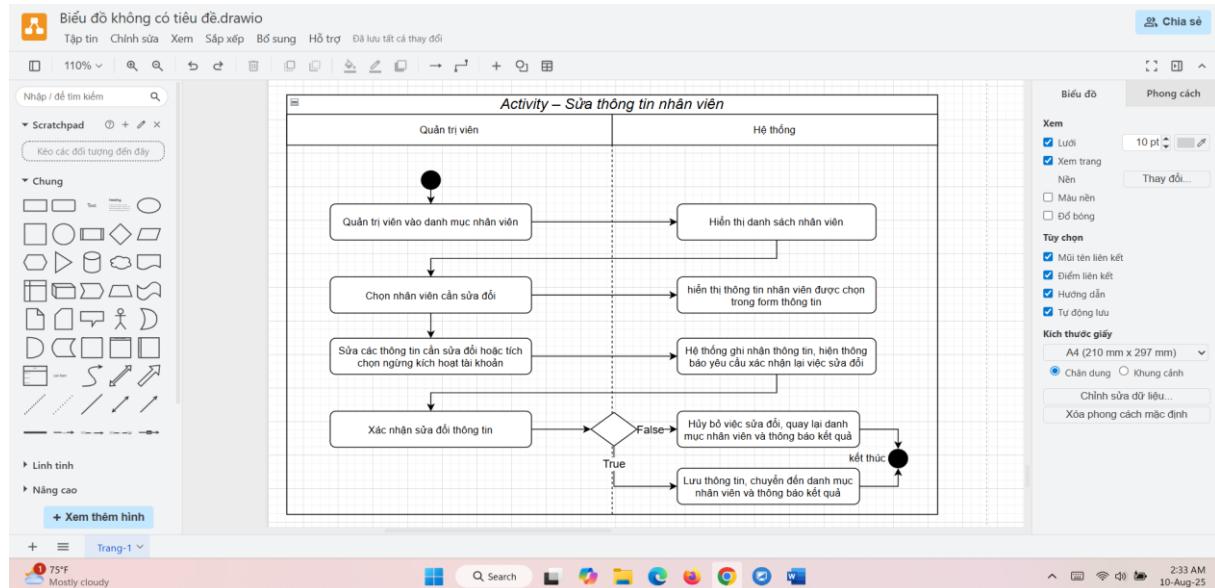
**- Activity – Thêm nhân viên:** Quản trị viên chọn thêm nhân viên ở danh mục nhân viên, tiến hành nhập thông tin nhân viên và phân quyền tài khoản. Hệ thống sẽ lưu trữ kích hoạt tài khoản vừa thêm.



Hình 3.15. Sơ đồ Activity – thêm nhân viên

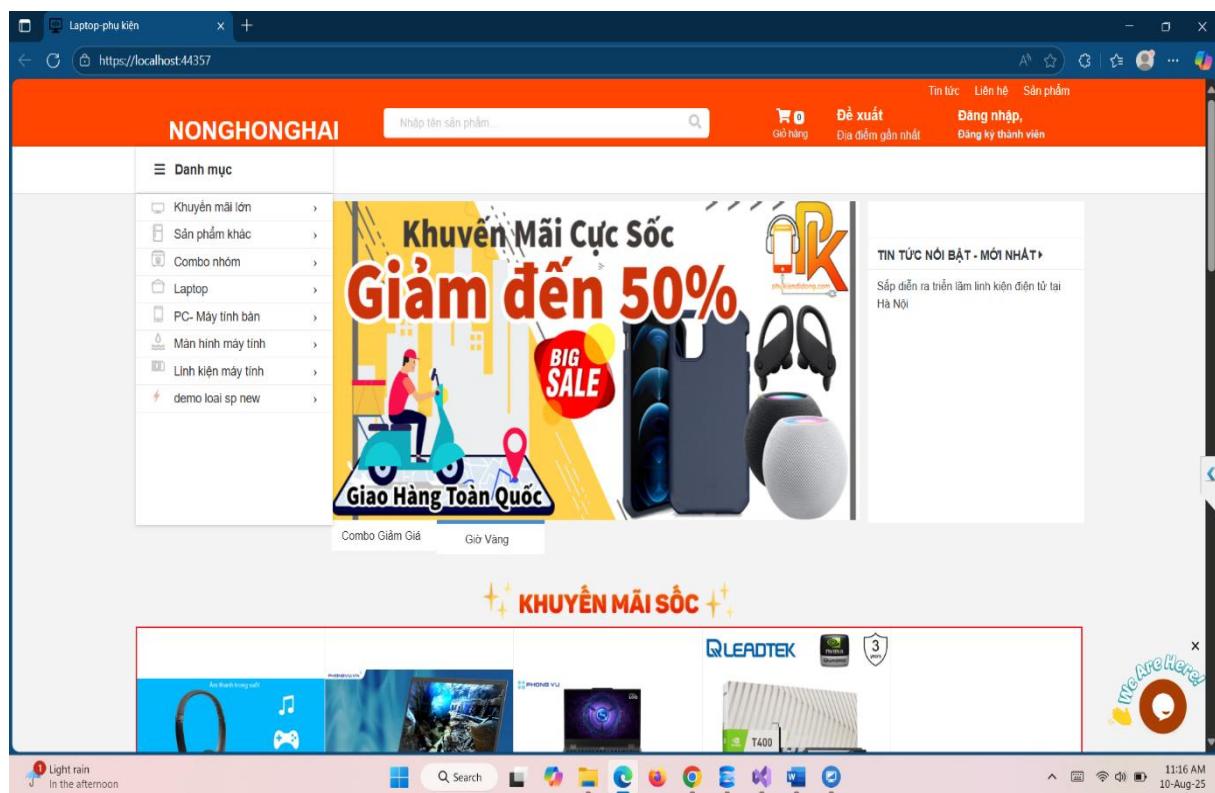
### - Activity – Sửa thông tin nhân viên, ngừng kích hoạt tài khoản nhân viên:

Quản trị viên chọn sửa thông tin nhân viên ở danh mục nhân viên, tiến hành cập nhập thông tin nhân viên, cập nhật quyền hạn tài khoản hoặc ngừng kích hoạt tài khoản. Hệ thống sẽ lưu lại thông tin tài khoản sau khi sửa đổi.

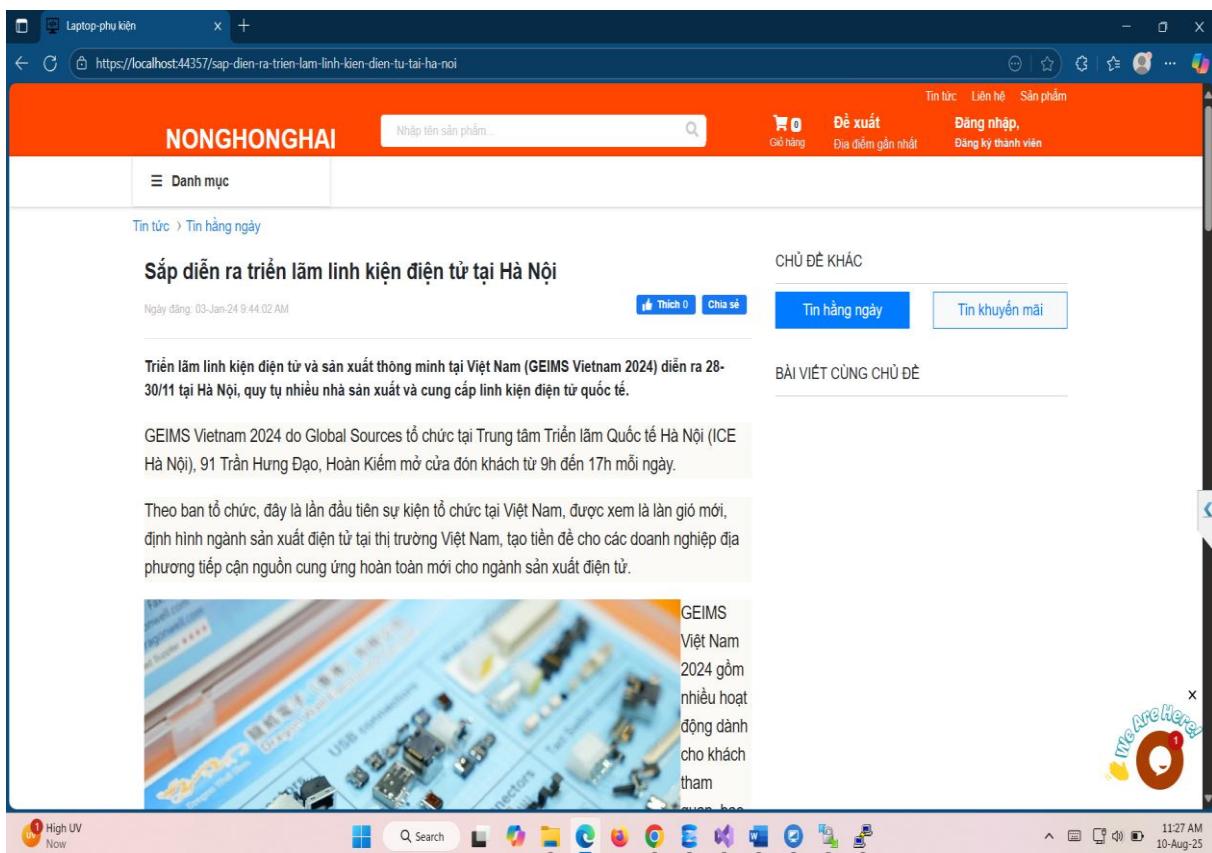


Hình 3.16. Sơ đồ Activity – Sửa thông tin nhân viên

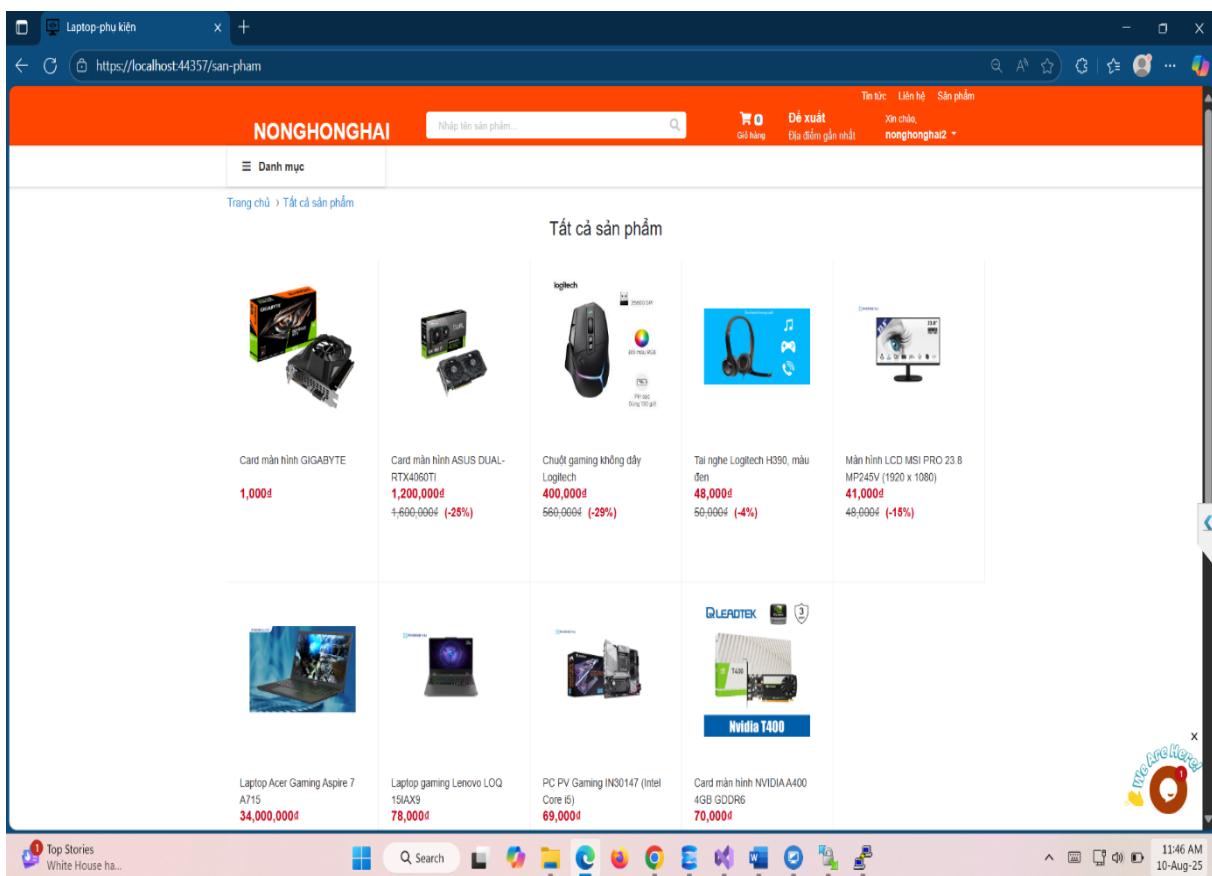
### 3.6. Giao diện hệ thống



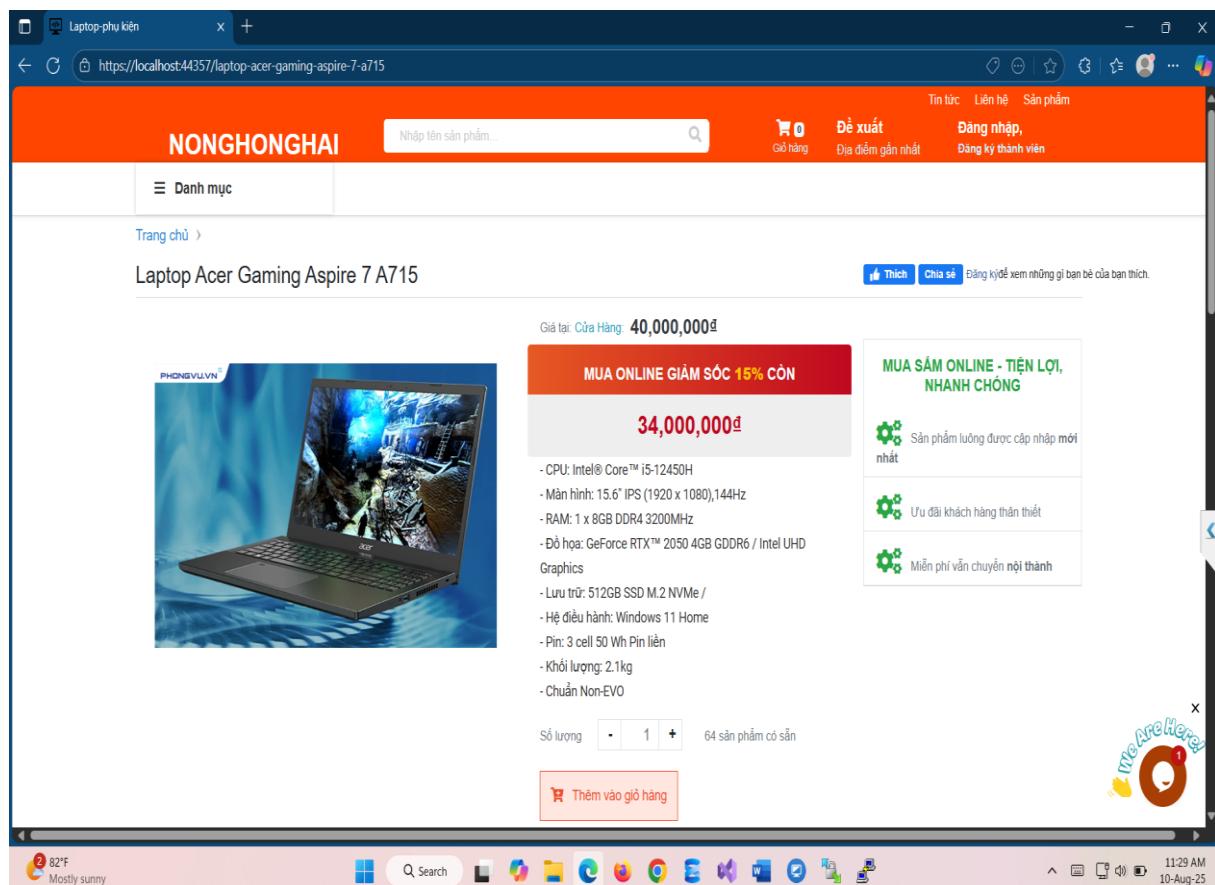
Hình 3.17. Giao diện trang chủ



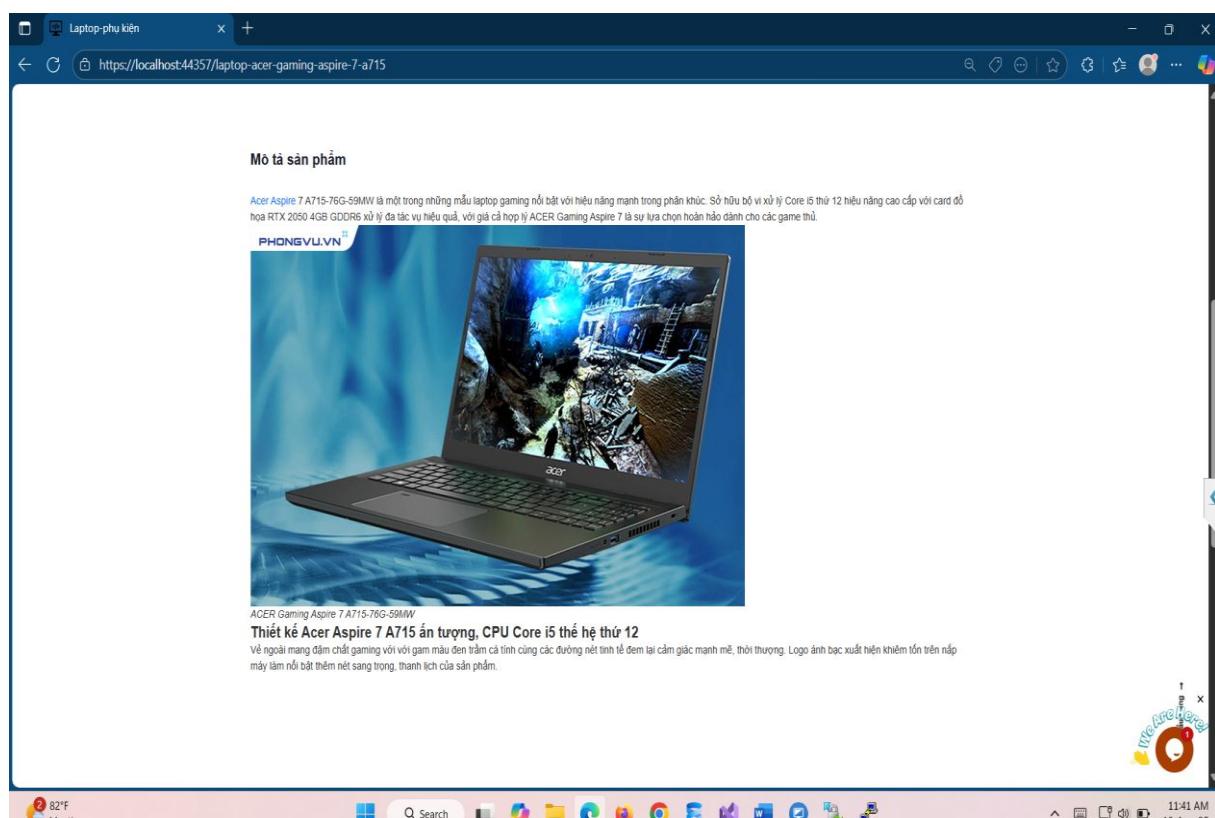
Hình 3.18. Giao diện trang tin tức



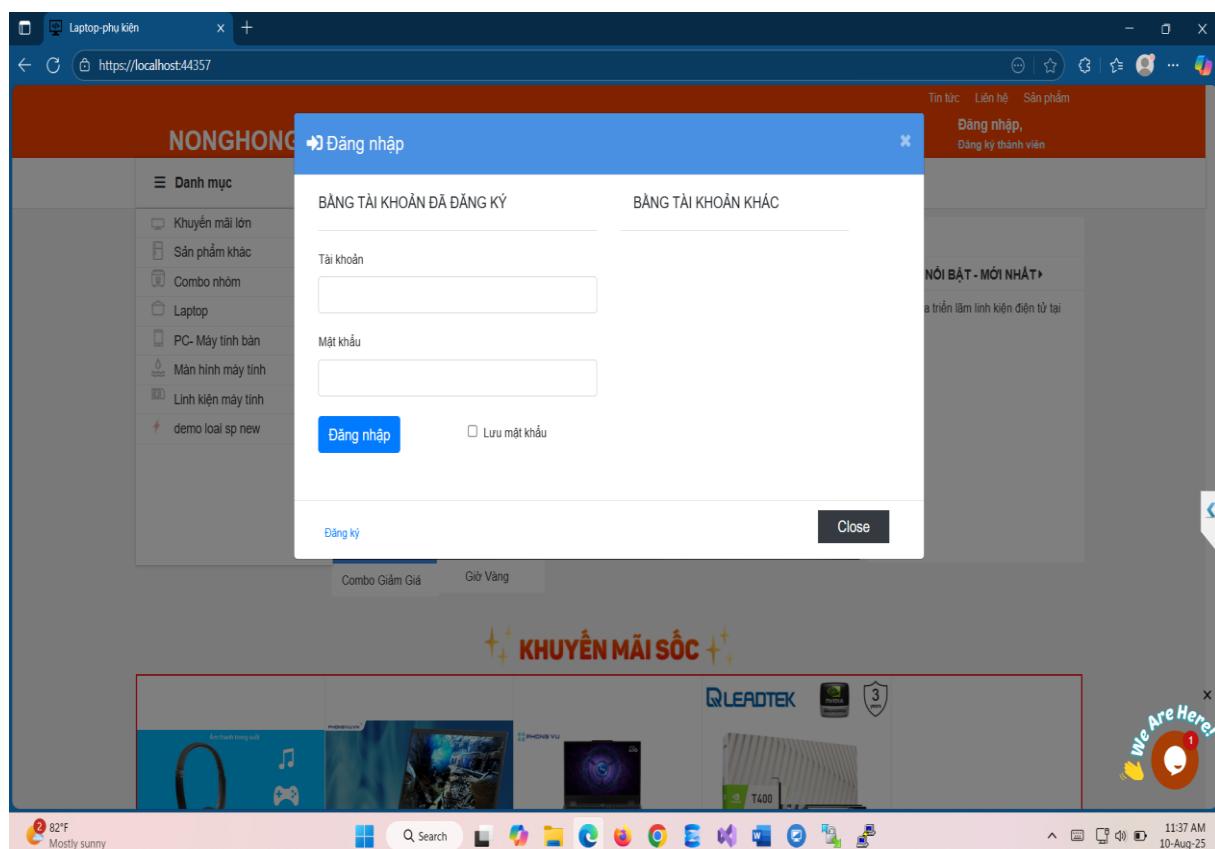
Hình 3.19. Giao diện phân danh mục sản phẩm



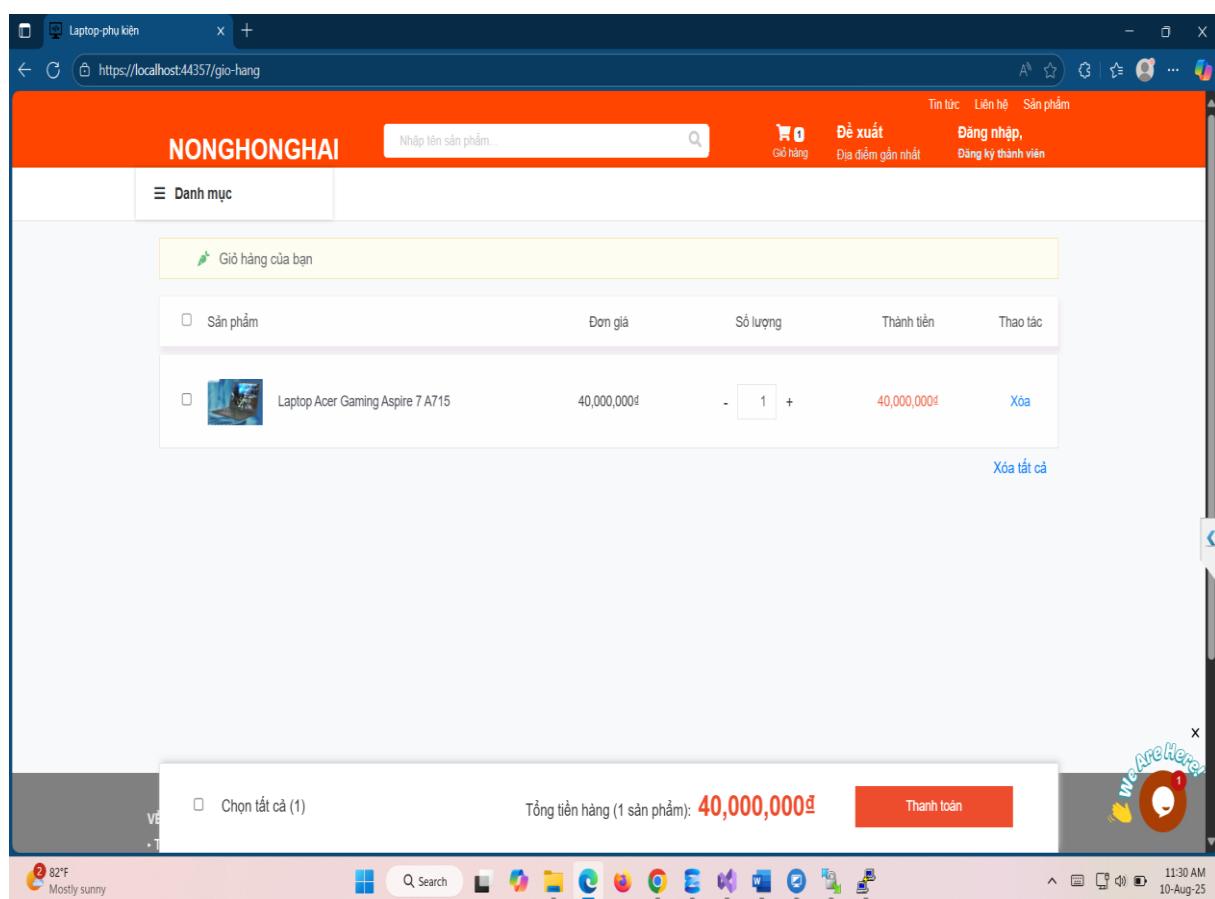
Hình 3.20. Giao diện trang chi tiết sản phẩm



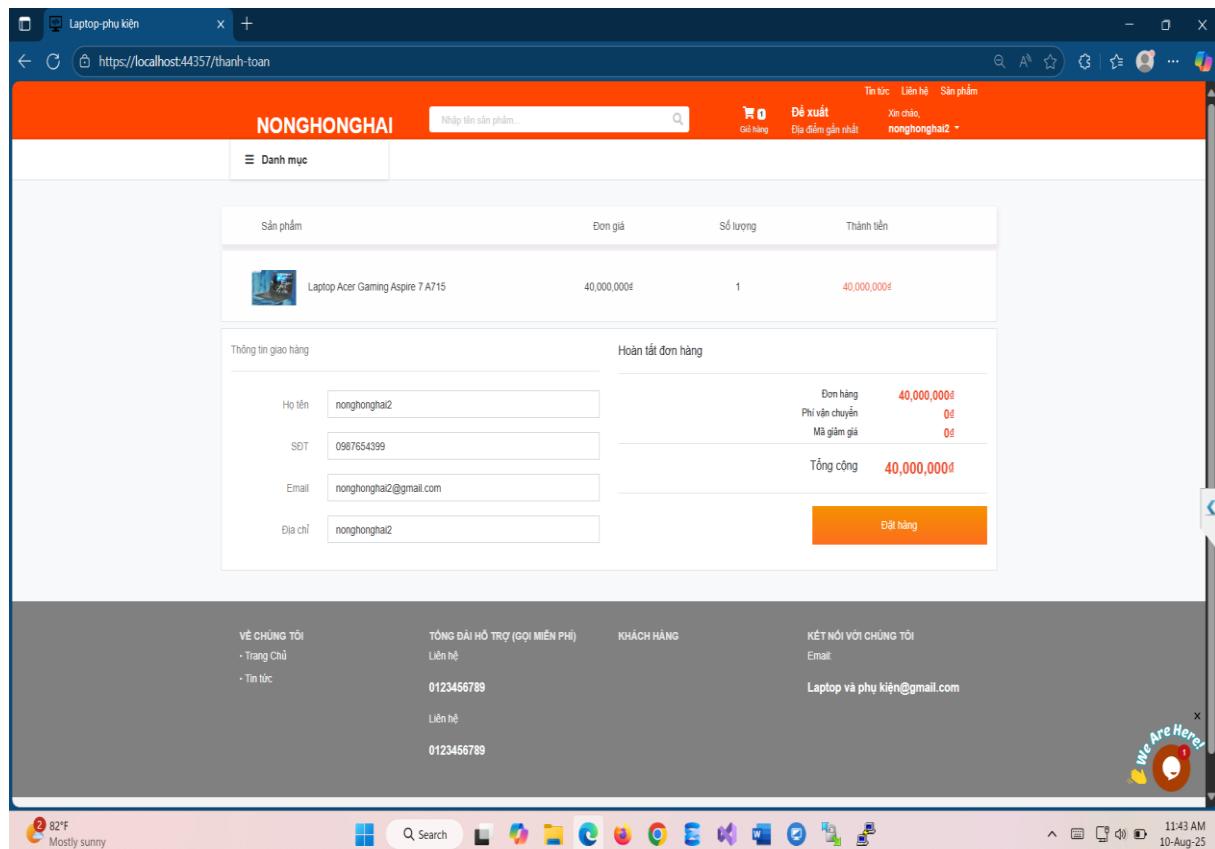
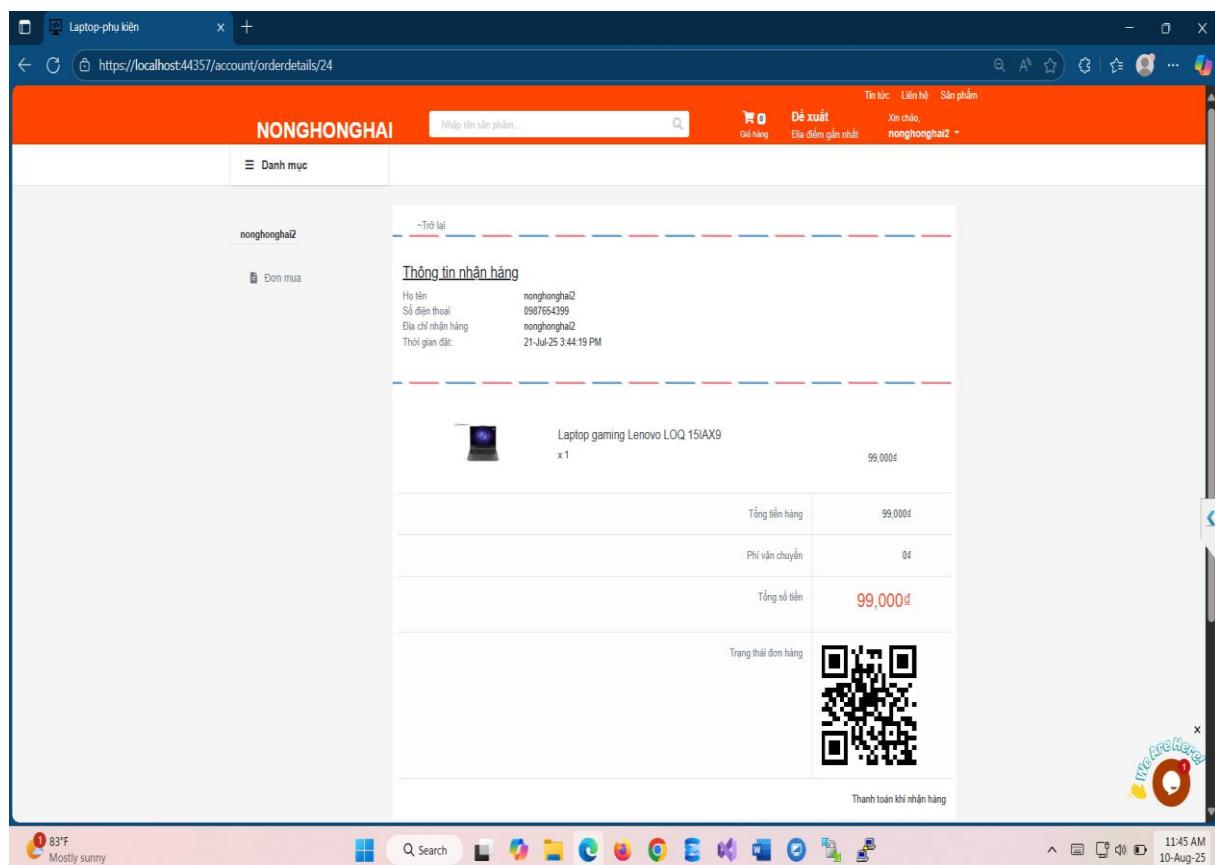
Hình 3.21. Giao diện trang mô tả sản phẩm

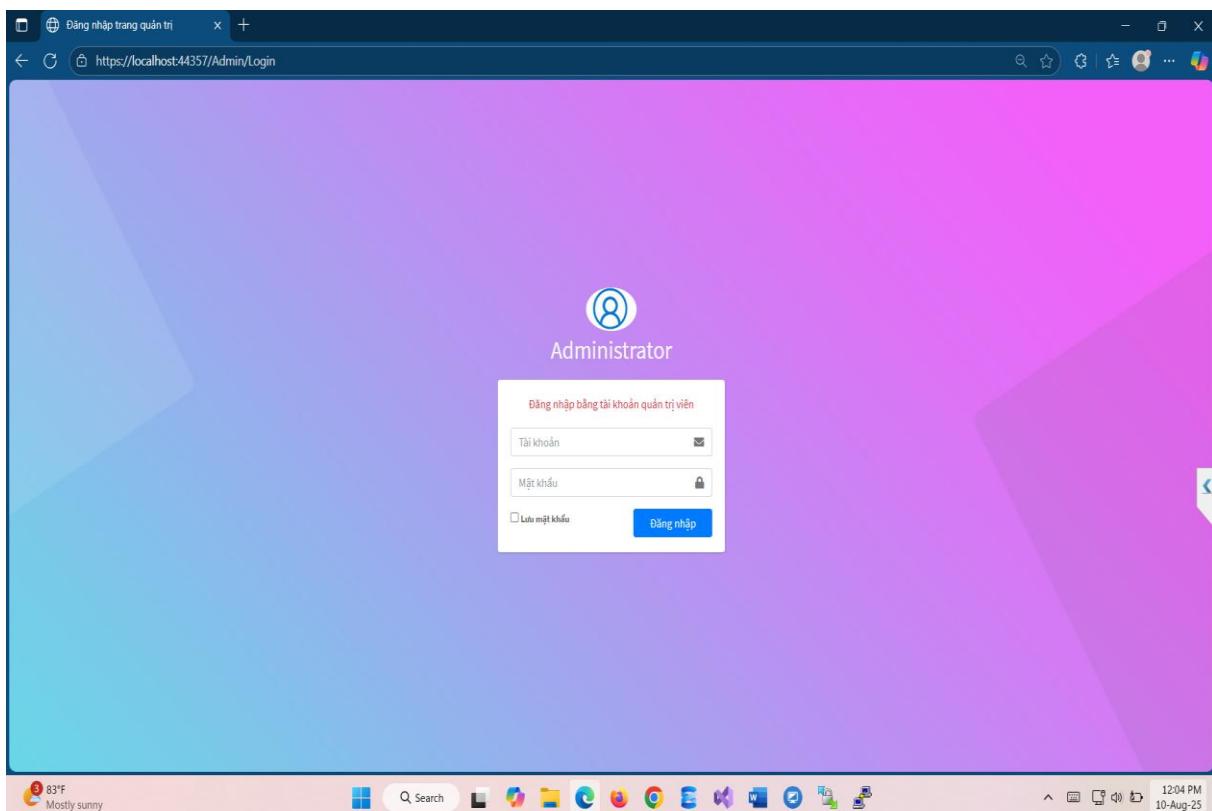


Hình 3.22. Giao diện màn hình đăng nhập khách hàng

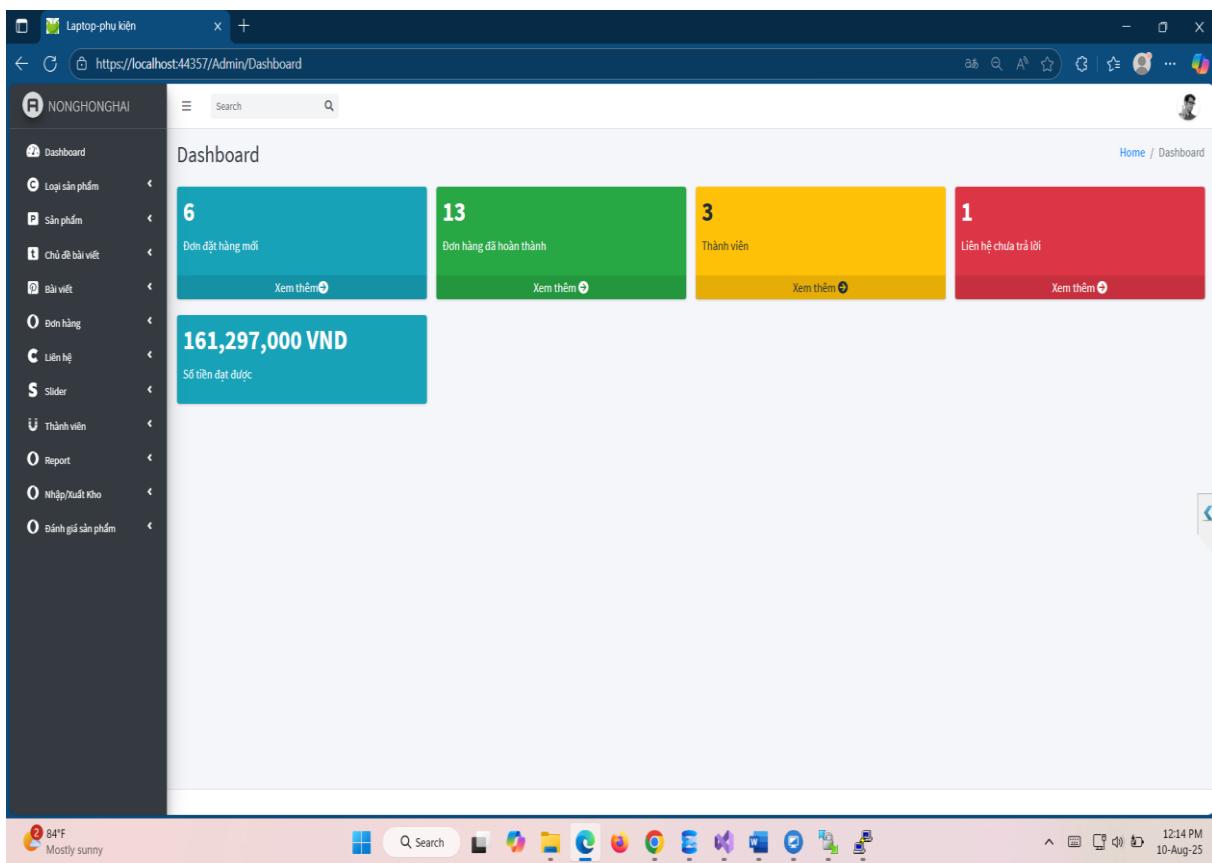


Hình 3.23. Giao diện gio hàng

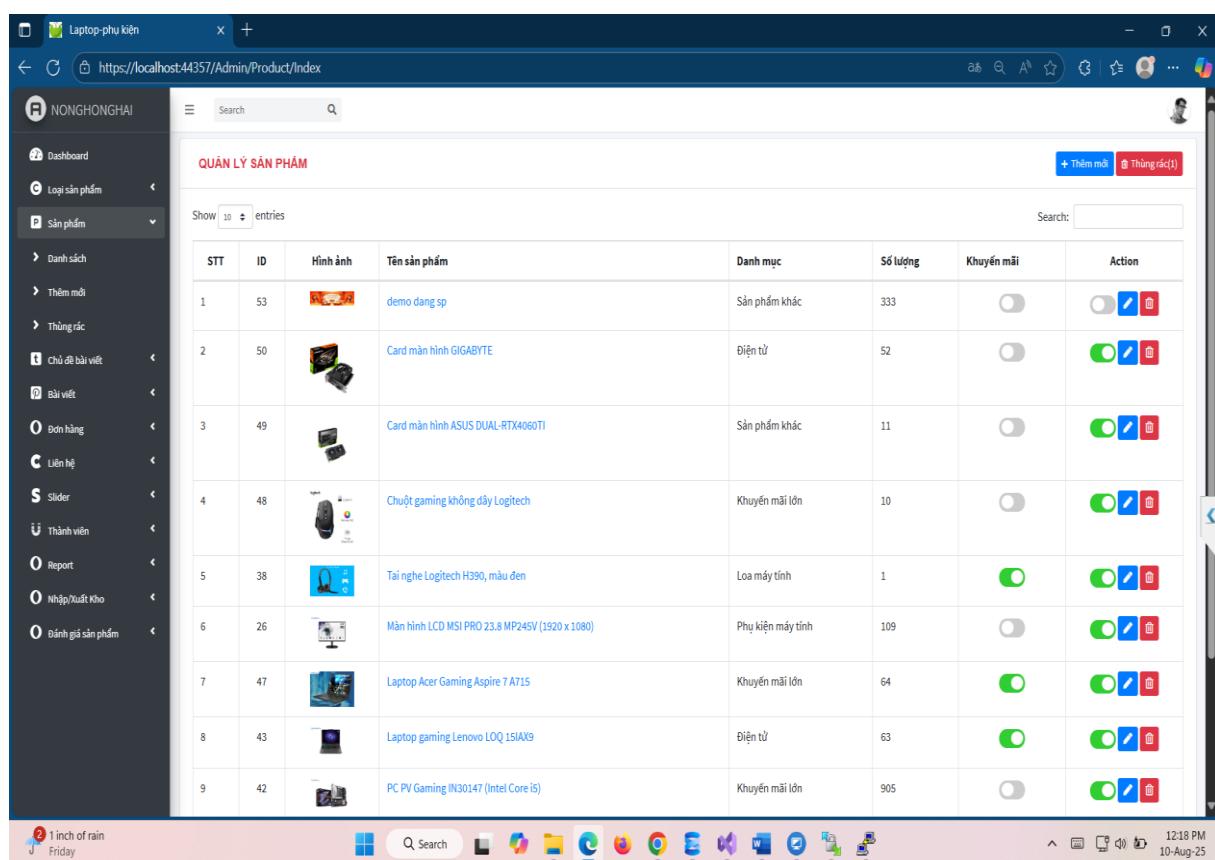
*Hình 3.24. Giao diện thông tin đặt hàng**Hình 3.25. Giao diện chi tiết đơn hàng*



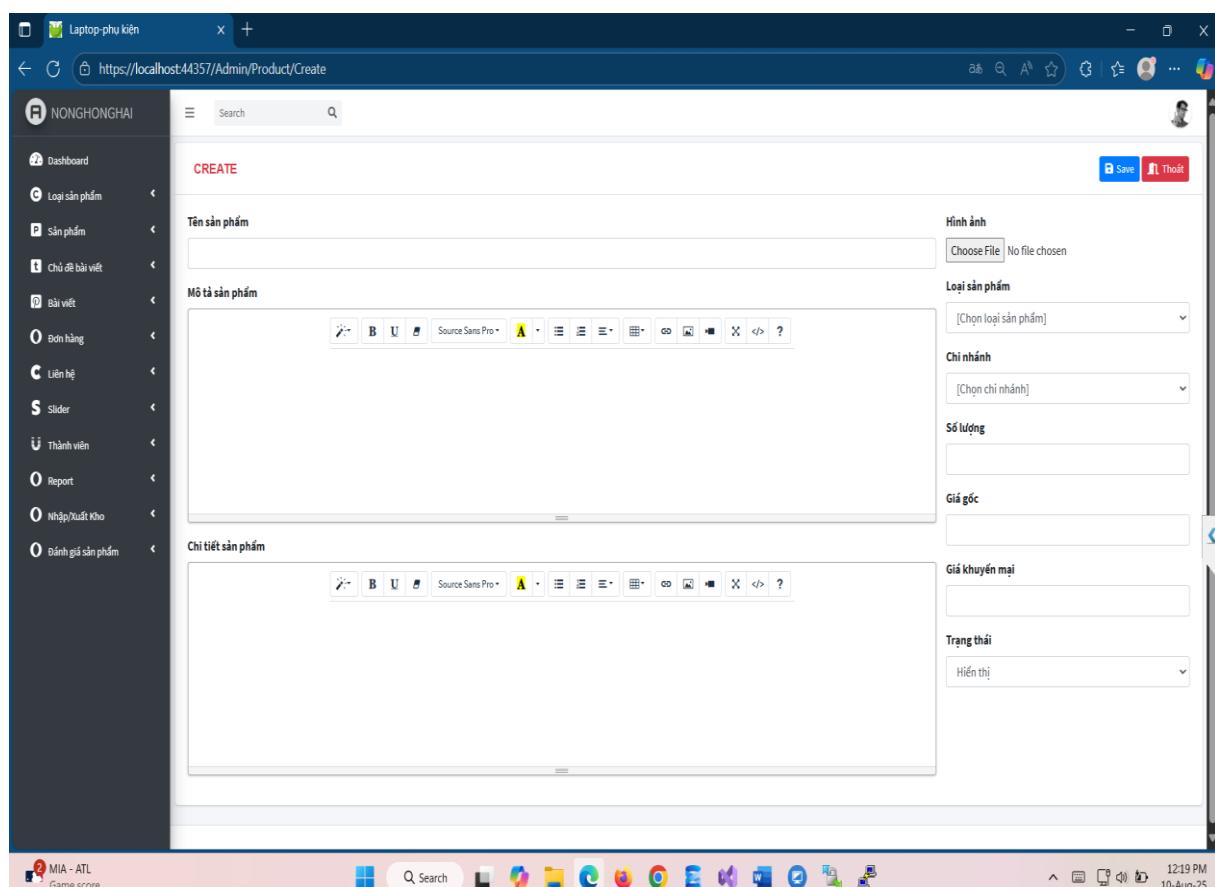
*Hình 3.26. Giao diện màn hình đăng nhập cho nhân viên*



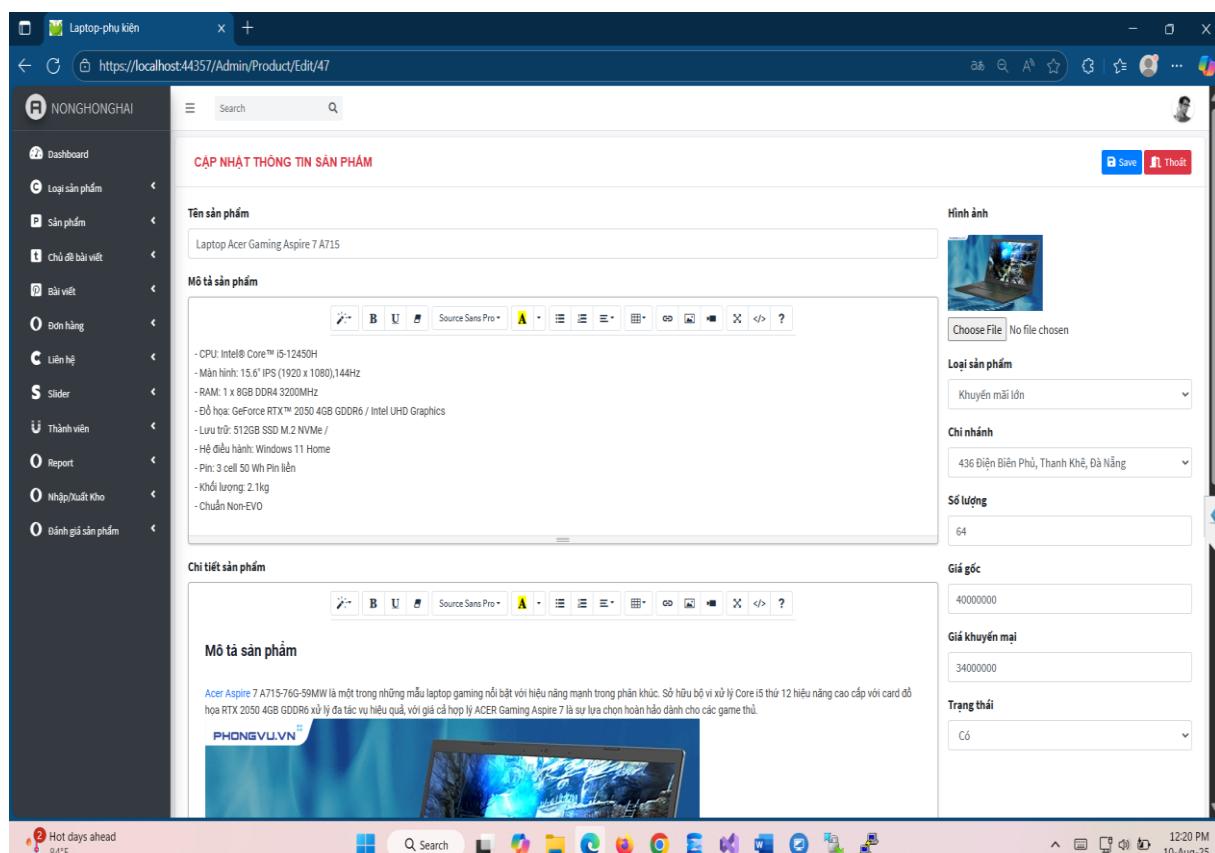
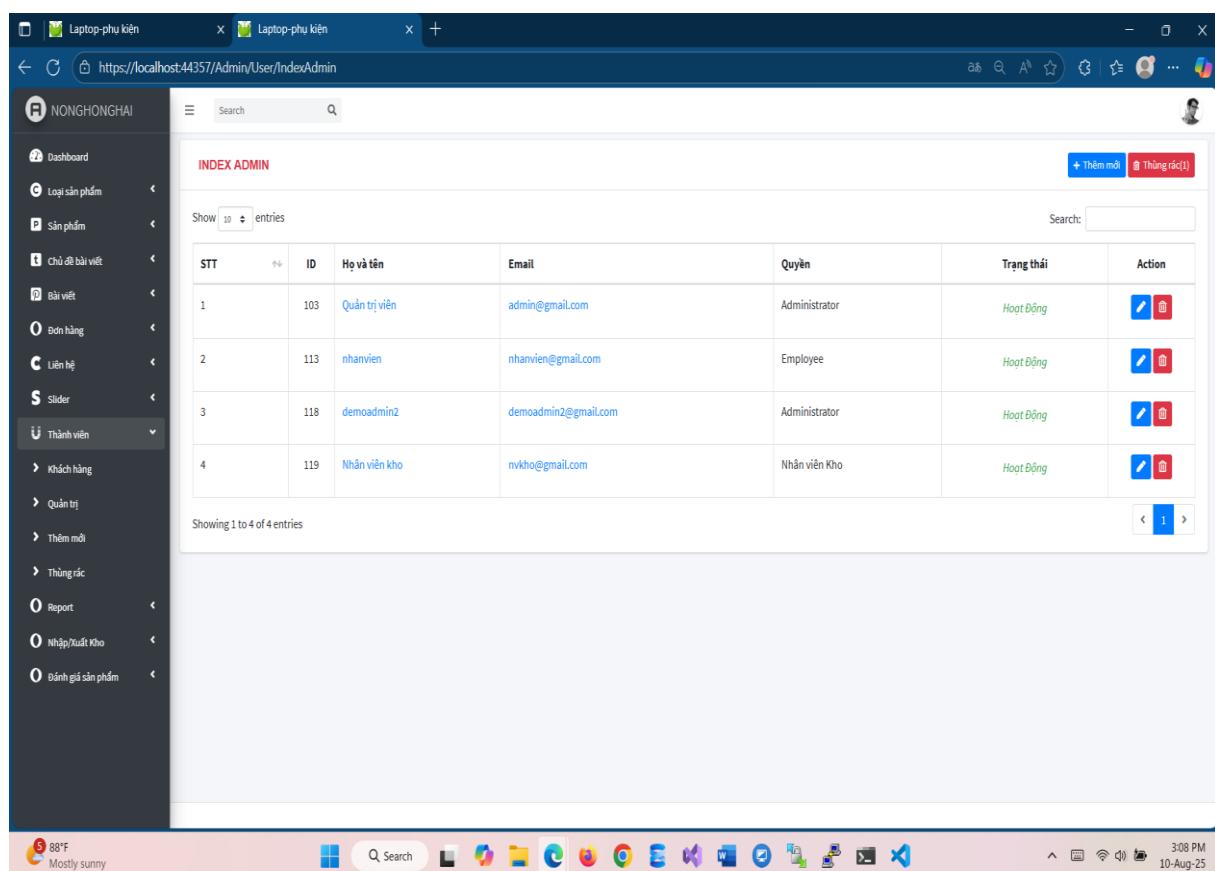
*Hình 3.27. Giao diện trang chủ trang quản trị*

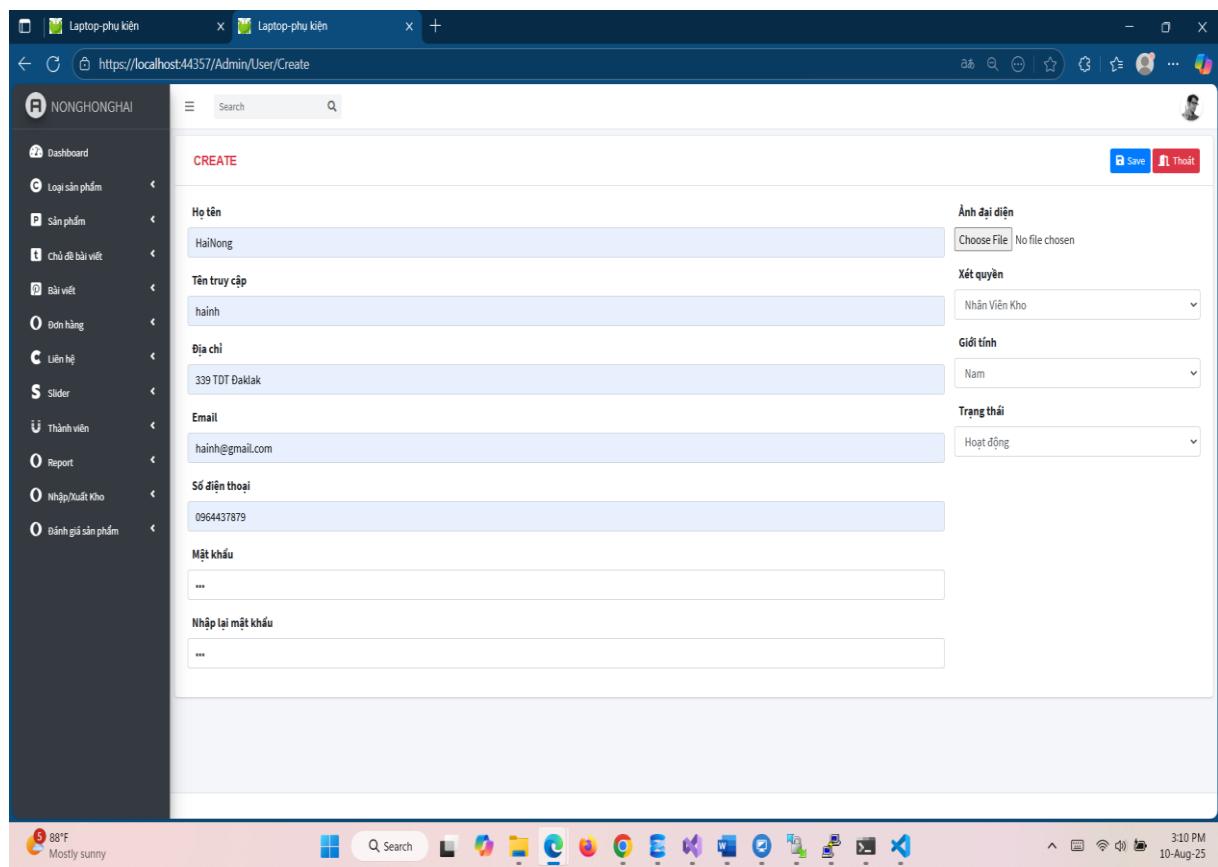


Hình 3.28. Giao diện quản lý sản phẩm

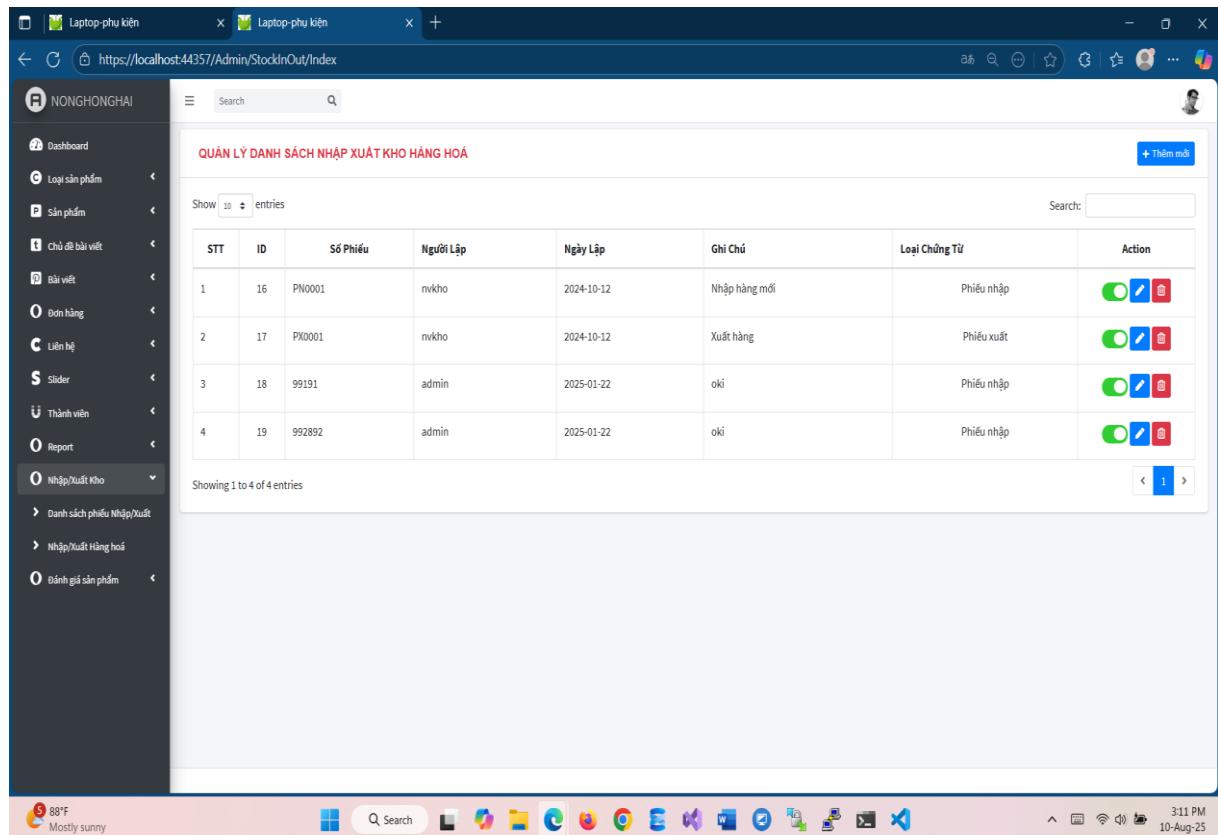


Hình 3.29. Giao diện thêm mới sản phẩm

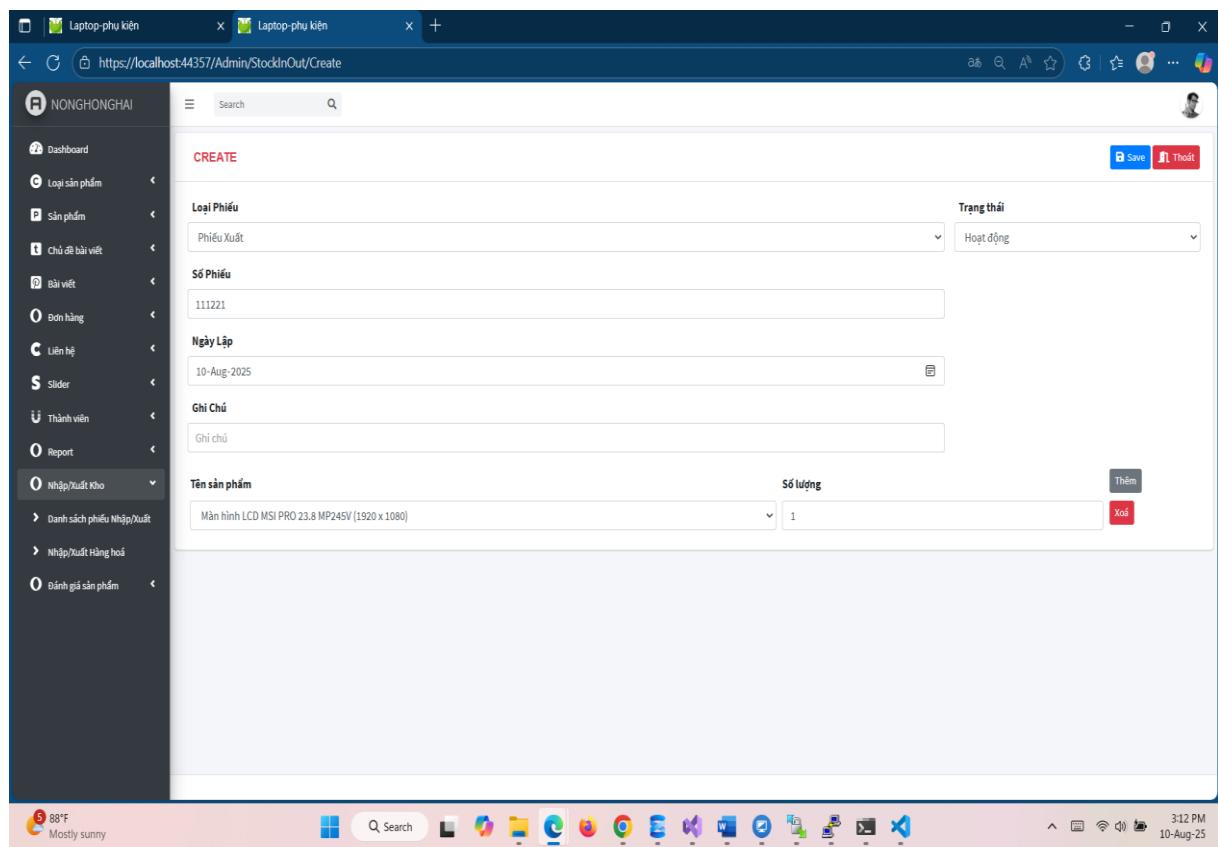
*Hình 3.30. Giao diện cập nhật thông tin sản phẩm**Hình 3.31. Giao diện quản lý tài khoản nhân viên*



Hình 3.32. Giao diện thêm mới tài khoản nhân viên



Hình 3.33. Giao diện quản lý phiếu nhập/xuất kho



Hình 3.34. Giao diện tạo phiếu nhập/ xuất hàng hóa

## CHƯƠNG 4: KẾT LUẬN

Trong quá trình thực hiện đề tài "Xây dựng website bán laptop", em đã có gắng tìm hiểu, thiết kế và triển khai các chức năng cần thiết nhằm đáp ứng yêu cầu cơ bản của một hệ thống bán hàng trực tuyến. Tuy nhiên, do thời gian thực hiện có hạn và kiến thức còn hạn chế, em chưa thể hoàn thiện đầy đủ các tính năng nâng cao như tích hợp thanh toán trực tuyến, tối ưu bảo mật và trải nghiệm người dùng.

### \* **Những kết quả đạt được:**

- Đã vận dụng lý thuyết cơ sở dữ liệu và phân tích hệ thống vào đề tài.
- Hiểu được nhiều hơn về cách quản lý bán hàng trực tuyến, có cơ hội để vận dụng kiến thức lý thuyết trong nhà trường áp dụng vào thực tiễn.
- Biết được cách thiết kế Web động cũng như cách tổ chức cơ sở dữ liệu.
- Hiểu sâu hơn về ASP.NET MVC, ngôn ngữ lập trình C#.
- Thiết kế giao diện được cho các giao diện liên quan đến đề tài.
- Cho phép tra cứu mặt hàng khi khách hàng có nhu cầu tìm hàng.

### \* **Những hạn chế còn tồn đọng:**

- Cách tổ chức dữ liệu và kỹ thuật lập trình chưa được tốt.
- Bắt lỗi người dùng chưa được toàn diện.
- Các chức năng của chương trình chưa linh động.
- Chưa bảo mật được toàn bộ dữ liệu.

## CHƯƠNG 5: HƯỚNG PHÁT TRIỂN

Mặc dù hệ thống đã đáp ứng được một số yêu cầu cơ bản về chức năng và giao diện, nhưng trong quá trình triển khai vẫn bộc lộ một số hạn chế. Để nâng cao chất lượng sản phẩm và hướng tới một hệ thống hoàn chỉnh hơn, đề tài cần được tiếp tục phát triển theo các hướng sau:

- Tích hợp cổng thanh toán trực tuyến như MoMo, VNPay hoặc thẻ ngân hàng để hỗ trợ giao dịch thực tế.
- Tăng cường bảo mật thông qua mã hóa dữ liệu, xác thực hai bước và kiểm tra lõi hỏng bảo mật.
- Tối ưu hiệu năng website, cải thiện tốc độ tải trang và khả năng xử lý khi có nhiều người dùng truy cập đồng thời.
- Phát triển phiên bản di động hoặc ứng dụng di động để mở rộng khả năng tiếp cận người dùng.
- Nghiên cứu cải tiến thêm cho hệ thống quản lý kho và đơn hàng chi tiết, hỗ trợ người quản trị theo dõi tình trạng hàng hóa và xử lý đơn hàng hiệu quả hơn.
- Ứng dụng trí tuệ nhân tạo để gợi ý sản phẩm theo hành vi mua sắm của người dùng.

## **DANH MỤC TÀI LIỆU THAM KHẢO**

1. <https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet>
2. <https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet-core>
3. <https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-8.0>
4. <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-8.0&tabs=visual-studio>
5. <https://aws.amazon.com/vi/what-is/net/>
6. <https://aws.amazon.com/vi/what-is/sql/>
7. <https://aws.amazon.com/vi/what-is/sql-database/>
8. <https://aws.amazon.com/vi/what-is/javascript/>
9. <https://www.w3schools.com/html/default.asp>
10. <https://www.w3schools.com/cs/index.php>
11. <https://www.w3schools.com/css/default.asp>
12. <https://www.w3schools.com/js/default.asp>
13. <https://www.w3schools.com/sql/default.asp>

## PHỤ LỤC

### DANH MỤC CÁC THUẬT NGỮ VIẾT TẮT

| Tù viết tắt | Tù đầy đủ                               | Nghĩa                                 |
|-------------|---|---------------------------------------|
| <b>IoT</b>  | Internet of Things                      | Vạn vật kết nối Internet              |
| <b>HTML</b> | HyperText Markup Language               | Ngôn ngữ siêu văn bản                 |
| <b>CSS</b>  | Cascading Style Sheets                  | Ngôn ngữ tạo phong cách cho trang web |
| <b>JS</b>   | JavaScript                              | Là một ngôn ngữ lập trình thông dịch  |
| <b>PHP</b>  | Personal Home Page                      | Ngôn ngữ lập trình kịch bản           |
| <b>GPS</b>  | Global Positioning System               | Hệ thống Định vị Toàn cầu             |
| <b>GSM</b>  | Global System for Mobile Communications | Hệ thống thông tin di động toàn cầu   |
| <b>GPRS</b> | General Packet Radio Service            | Dịch vụ vô tuyến gói tổng hợp         |
| <b>IC</b>   | Integrated Circuit                      | Mạch tích hợp                         |
| <b>API</b>  | Application Programming Interface       | Giao diện lập trình ứng dụng          |