**BỘ CÔNG THƯƠNG**

**TRƯỜNG CAO ĐẲNG CÔNG THƯƠNG TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN THỰC TẬP TỐT NGHIỆP**

**XÂY DỰNG WEBSITE BÁN SẢN PHẨM THÚ CƯNG BẰNG REACTJS VÀ FASTIFY**

**Giảng viên hướng dẫn: Nguyễn Thái Khánh Quyên**

**Sinh viên thực hiện:**

1. Trần Lê Tiến MSSV: 2121110330 Lớp: CCQ2111J

TP. Hồ Chí Minh – Tháng 06/2024

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

--------------🙣🕮🙡--------------

|  |
| --- |
| *Thành phố Hồ Chí Minh, ngày …. tháng …. năm 2024* |
| **Giáo viên hướng dẫn** |
| (Ký tên, ghi rõ họ tên) |

# LỜI CẢM ƠN

Trong thời gian làm đồ án tốt nghiệp, chúng em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè. Trước hết, chúng em xin gửi lời cảm ơn sâu sắc đến cô **Nguyễn Thái Khánh Quyên**, người đã trực tiếp hướng dẫn, chỉ bảo chúng em trong suốt quá trình thực hiện đồ án. Cô đã dành nhiều thời gian quý báu để giúp đỡ chúng em từ khâu lên ý tưởng, thu thập tài liệu, cho đến khi hoàn thành sản phẩm cuối cùng. Những góp ý và chỉ dẫn của cô không chỉ giúp chúng em hoàn thiện đồ án mà còn mở rộng tầm nhìn, kiến thức chuyên môn. Bên cạnh đó, chúng em cũng xin cảm ơn gia đình đã luôn ủng hộ và động viên tinh thần, tạo điều kiện tốt nhất để chúng em có thể tập trung vào việc học.

Cuối cùng, chúng em muốn gửi lời cảm ơn đến tất cả các bạn bè, những người đã luôn bên cạnh, chia sẻ và giúp đỡ chúng em trong quá trình làm đồ án. Tất cả những sự giúp đỡ này đã góp phần quan trọng giúp chúng em hoàn thành đồ án tốt nghiệp một cách xuất sắc.

Em xin chân thành cảm ơn!

|  |
| --- |
| *Thành phố Hồ Chí Minh, ngày …. tháng …. năm 2024* |
| Sinh viên thực hiện |
| (Ký tên, ghi rõ họ tên) |
|  |
|  |

# LỜI MỞ ĐẦU

1. **Lý do chọn đề tài**

Ngày nay, công nghệ thông tin đã có những bước phát triển mạnh mẽ theo cả chiều rộng và sâu. Máy tính điện tử không còn là một phương tiện quý hiếm mà đang ngày càng trở thành một công cụ làm việc và giải trí thông dụng, không chỉ ở nơi làm việc mà còn ngay cả trong gia đình. Đặc biệt, công nghệ thông tin được áp dụng trên mọi lĩnh vực kinh tế, chính trị, xã hội. Ứng dụng công nghệ thông tin và tin học hóa được xem là một trong những yếu tố mang tính quyết định trong hoạt động của quốc gia, tổ chức và cả các cửa hàng. Nó đóng vai trò hết sức quan trọng và có thể tạo nên bước đột phá mạnh mẽ.

Mạng INTERNET là một trong những sản phẩm có giá trị hết sức lớn lao và ngày càng trở nên một công cụ không thể thiếu, là nền tảng để truyền tải, trao đổi thông tin trên toàn cầu. Bằng INTERNET, chúng ta có thể thực hiện những công việc với tốc độ nhanh hơn và chi phí thấp hơn nhiều so với cách thức truyền thống. Chính điều này đã thúc đẩy sự khai sinh và phát triển của thương mại điện tử trên khắp thế giới, làm biến đổi đáng kể bộ mặt văn hóa, nâng cao đời sống con người. Trong hoạt động sản xuất, kinh doanh, thương mại điện tử đã khẳng định được vai trò xúc tiến và thúc đẩy sự phát triển của doanh nghiệp. Đối với một cửa hàng, việc quảng bá và giới thiệu sản phẩm đến khách hàng để đáp ứng nhu cầu mua sắm ngày càng cao là cần thiết.

Vì vậy, em đã thực hiện đề tài “Xây dựng Website bán sản phẩm thú cưng”. Cửa hàng có thể đưa các sản phẩm lên website của mình và quản lý website đó. Khách hàng có thể đặt mua sản phẩm của cửa hàng mà không cần đến trực tiếp, cửa hàng sẽ gửi sản phẩm đến tận tay khách hàng. Website là nơi cửa hàng quảng bá tốt nhất tất cả các sản phẩm mình bán ra.

1. **Mục đích nghiên cứu (các kết quả cần đạt được)**

Mục đích của đề tài là mô tả một cách chi tiết về những chức năng của “Xây dựng web bán sản phẩm thú cưng”. Website sẽ minh họa chi tiết các chức năng và xây dựng một hệ thống hoàn chỉnh giúp người dùng mua sắm một cách trực tuyến và dễ dàng hơn. Đề tài sẽ giải thích rõ ràng về ràng buộc của hệ thống, trình bày tổng quan về giao diện và những tương tác với hệ thống.

Nâng cao trải nghiệm mua sắm của khách hàng: Trang web có thể nghiên cứu để hiểu rõ hơn về nhu cầu và mong muốn của khách hàng khi mua sắm các sản phẩm thú cưng. Kết quả từ nghiên cứu có thể giúp cải thiện giao diện, tính năng và dịch vụ để tối ưu hóa trải nghiệm mua sắm của khách hàng.

Tối ưu hóa chính sách định giá và khuyến mãi: Dữ liệu từ nghiên cứu có thể giúp trang web hiểu được mức độ nhạy cảm của khách hàng đối với giá cả và giá trị của các khuyến mãi. Dựa trên thông tin này, có thể điều chỉnh chính sách giá cả và khuyến mãi để tối ưu hóa doanh số bán hàng và lợi nhuận.

Phát triển chiến lược tiếp thị hiệu quả: Kết quả từ nghiên cứu có thể giúp trang web hiểu rõ hơn về phương tiện tiếp thị nào hiệu quả nhất để tiếp cận và thu hút khách hàng tiềm năng. Điều này giúp tối ưu hóa ngân sách tiếp thị và đạt được hiệu quả cao nhất.

1. **Đối tượng và phạm vi nghiên cứu**

Đối tượng nghiên cứu của trang web bán sản phẩm thú cưng thường là người tiêu dùng có nhu cầu mua các sản phẩm liên quan đến thú cưng. Đây có thể là những người đang tìm kiếm thông tin về các sản phẩm cụ thể để so sánh, đánh giá và chọn lựa trước khi mua. Ngoài ra, cũng có những người đang quan tâm đến các đánh giá, đánh giá từ người dùng khác và các bài viết so sánh giữa các thương hiệu và mẫu mã khác nhau.

Phạm vi của các trang web này thường tập trung vào việc cung cấp thông tin chi tiết và đáng tin cậy về các sản phẩm thú cưng để giúp người tiêu dùng quyết định mua hàng một cách thông thái và hợp lý.

Áp dụng các công nghệ: Reactjs, framework Fastify...

**MỤC LỤC**

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN i](#_Toc11966)

[LỜI CẢM ƠN ii](#_Toc5490)

[LỜI MỞ ĐẦU iii](#_Toc19236)

[1. Lý do chọn đề tài iii](#_Toc20754)

[2. Mục đích nghiên cứu (các kết quả cần đạt được) iii](#_Toc23785)

[3. Đối tượng và phạm vi nghiên cứu iv](#_Toc26737)

[MỤC LỤC v](#_Toc24805)

[DANH MỤC CÁC BẢNG vii](#_Toc28119)

[DANH MỤC HÌNH ẢNH viii](#_Toc19907)

[Chương 1: CƠ SỞ LÝ THUYẾT 1](#_Toc7330)

[1.1 Giới thiệu tổng quan về framework Fastify 1](#_Toc6986)

[1.1.1 Ưu điểm 1](#_Toc30713)

[1.1.2 Các chức năng 1](#_Toc18606)

[1.1.3 Các dịch vụ của fastify 1](#_Toc4149)

[1.2 Cơ sở dữ liệu MySQL 2](#_Toc730)

[1.3 Form và các đối tượng của Form 2](#_Toc11440)

[1.3.1 Giới thiệu 2](#_Toc20876)

[1.3.2 GET 2](#_Toc17614)

[1.3.3 POST 3](#_Toc10930)

[1.3.4 DELETE 3](#_Toc19066)

[1.4 RESTful API 4](#_Toc1399)

[1.4.1 Khái niệm 4](#_Toc6515)

[1.4.2 Các thành phần 4](#_Toc4822)

[1.4.3 RESTful API hoạt động 4](#_Toc14875)

[1.4.4 Cách gọi API bằng thư viện Axios 5](#_Toc12210)

[*Hình 1.2: Cách sử dụng Axios để gọi API* 5](#_Toc23464)

[1.5 REACT JS 5](#_Toc1221)

[1.5.1 Tổng quan về React JS 5](#_Toc18658)

[1.5.2 Các thành phần cơ bản của React JS 6](#_Toc10975)

[Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 9](#_Toc29656)

[2.1 Khảo sát hiện trạng và yêu cầu 9](#_Toc7760)

[2.1.1 Khảo sát hiện trạng 9](#_Toc22575)

[2.2 Phân tích hệ thống 9](#_Toc25202)

[2.2.1 Xác định các Actor 9](#_Toc12535)

[Khách hàng: 9](#_Toc32427)

[Quản trị viên: 9](#_Toc9778)

[2.2.2 Xác định các Use Case 9](#_Toc2149)

[2.2.3 Sơ đồ Use Case 10](#_Toc30566)

[*Hình 2.2: Biểu đồ use case tác nhân khách hàng* 11](#_Toc18885)

[2.2.4 Phân tích 11](#_Toc5109)

[2.3 Thiết kế 13](#_Toc21294)

[2.3.1 Biểu đồ hoạt động tổng quát 13](#_Toc22295)

[2.3.1.1 Tác nhân khách hành 13](#_Toc30165)

[2.3.1.2 Tác nhân quản trị viên 14](#_Toc3229)

[2.3.2 Biểu đồ tuần tự 15](#_Toc22510)

[Chương 3: GIAO DIỆN HỆ THỐNG 19](#_Toc19697)

[3.1 Giao diện trang quản trị 20](#_Toc5941)

[3.1.1 Login trang quản trị 20](#_Toc29165)

[3.1.2 Trang quản lý sản phẩm 21](#_Toc21739)

[3.1.3 Trang quản lý danh mục 23](#_Toc13174)

[3.1.4 Trang quản lý đơn đặt hàng của khách hàng 24](#_Toc7796)

[3.1.5 Trang quản lý tài khoản 25](#_Toc12811)

[3.1.7 Trang Slider 2](#_Toc64)

[3.1.8 Trang quản lý trang đơn 3](#_Toc9891)

[3.2 Giao diện người dùng 4](#_Toc15527)

[3.2.1 Trang chủ 4](#_Toc27271)

[3.2.2 Trang tất cả sản phẩm 6](#_Toc11987)

[3.2.3 Trang sản phẩm theo loại 6](#_Toc24343)

[3.2.4 Trang chi tiết sản phẩm 7](#_Toc25087)

[3.2.5 Trang giỏ hàng 8](#_Toc15513)

[3.2.6 Trang tìm kiếm 9](#_Toc26504)

[3.2.6 Trang bài viết 10](#_Toc18552)

[3.2.7 Trang thanh toán 11](#_Toc2721)

[3.2.8 Trang lịch sử đơn hàng của khách hàng 11](#_Toc13994)

[3.2.9 Trang thay đổi thông tin khách hàng 13](#_Toc14004)

[3.2.10 Trang yêu cầu đổi mật khẩu 13](#_Toc22291)

[3.2.11 Trang đăng kí 13](#_Toc19889)

[3.2.12 Trang đăng nhập 14](#_Toc30486)

[KẾT LUẬN 15](#_Toc21973)

[TÀI LIỆU THAM KHẢO 16](#_Toc24770)

[ Tiếng Việt 16](#_Toc25264)

[ Trang web 16](#_Toc27697)

DANH MỤC HÌNH ẢNH

[Hình 1. 1 Ví dụ về phương thức GET 4](#_Toc106308889)

[Hình 1. 2 Ví dụ về phương thức POST 4](#_Toc106308890)

[Hình 1. 3 Ví dụ về phương thức DELETE 5](#_Toc106308891)

[Hình 1. 4 MongoDB 6](#_Toc106308892)

[Hình 1. 5 Mô tả cách thức hoạt động của MongoDB 7](#_Toc106308893)

[Hình 1. 6 Ví dụ về truy vấn 7](#_Toc106308894)

[Hình 1. 7 Ví dụ về truy vấn 12](#_Toc106308895)

[Hình 1. 8 Ví dụ về Cookie 13](#_Toc106308896)

[Hình 1. 9 Sessions trong NodeJS 15](#_Toc106308897)

[Hình 1. 10 Một số tính năng của VSC 18](#_Toc106308898)

[Hình 1. 11 REST API Design 19](#_Toc106308899)

[Hình 1. 12 Cách hoạt động của RESTful API 20](#_Toc106308900)

# Chương 1: CƠ SỞ LÝ THUYẾT

## **Giới thiệu tổng quan về framework Fastify**

Fastify là một framework Node.js nhẹ và hiệu suất cao được thiết kế để xử lý các ứng dụng web và API. Nó được xây dựng nhằm cung cấp hiệu suất vượt trội và tiêu thụ ít tài nguyên hơn so với các framework khác như Express.

### **Ưu điểm**

Fastify có nhiều ưu điểm đáng chú ý, bao gồm:

* **Hiệu suất cao**: Fastify được thiết kế để đạt hiệu suất cao, sử dụng kiến trúc tối ưu và áp dụng các kỹ thuật tối ưu hóa để xử lý các yêu cầu nhanh chóng và hiệu quả.
* **Schema-Based Validation**: Fastify tích hợp sẵn hỗ trợ cho validation dựa trên schema, đảm bảo rằng dữ liệu gửi và nhận là hợp lệ.
* **Modular và Plugin System**: Fastify có kiến trúc linh hoạt và modulized, cho phép sử dụng các plugin để mở rộng chức năng của ứng dụng một cách dễ dàng.
* **Hỗ trợ TypeScript**: Fastify cung cấp hỗ trợ tích hợp cho TypeScript, cho phép viết mã bằng TypeScript và tận dụng các tính năng kiểm tra kiểu tĩnh và tự động hoàn thiện mã.
* **Xử lý lỗi tốt**: Fastify cung cấp cách tiếp cận mạnh mẽ và linh hoạt cho việc xử lý lỗi trong ứng dụng.
* **Hỗ trợ thư viện middleware**: Fastify hỗ trợ các thư viện middleware phổ biến như Helmet, CORS và nhiều hơn nữa, giúp dễ dàng bảo mật và mở rộng ứng dụng.
* **Thiết kế dễ dàng sử dụng**: Fastify có API sạch sẽ và dễ sử dụng, giúp việc xây dựng và duy trì ứng dụng web trở nên đơn giản hơn.

### **1.1.2 Các chức năng và tính năng**

Fastify cung cấp một loạt các chức năng và tính năng mạnh mẽ để phát triển ứng dụng web hiệu suất cao. Dưới đây là một số chức năng chính của Fastify:

* **Routing**: Fastify cung cấp một hệ thống định tuyến mạnh mẽ, cho phép bạn định nghĩa các tuyến đường (routes) và xử lý yêu cầu HTTP tương ứng.
* **Middleware Support**: Fastify hỗ trợ sử dụng middleware để xử lý các yêu cầu HTTP trước khi chúng đến các xử lý chính. Điều này cho phép bạn thực hiện các chức năng như xác thực, ghi log, nén và nhiều hơn nữa.
* **Compression**: Fastify hỗ trợ nén dữ liệu để giảm bớt kích thước của phản hồi HTTP, giúp tối ưu hóa băng thông và tăng tốc độ tải trang.
* **Lifecycle Hooks**: Fastify cho phép bạn sử dụng các hooks như onRequest, preValidation, preHandler, onSend, và onResponse để thực hiện các thao tác khác nhau trong vòng đời của mỗi yêu cầu HTTP.
* **Logging**: Fastify hỗ trợ ghi log một cách dễ dàng, giúp bạn theo dõi và gỡ lỗi ứng dụng của mình.

### **1.1.3 Các dịch vụ của fastify**

Fastify cung cấp một số dịch vụ tích hợp và bổ sung mà bạn có thể sử dụng để mở rộng chức năng của ứng dụng của mình. Dưới đây là một số dịch vụ quan trọng của Fastify:

* **fastify-cors**: Dịch vụ này cung cấp hỗ trợ cho Cross-Origin Resource Sharing (CORS), cho phép bạn cấu hình các chính sách CORS để xử lý yêu cầu từ các miền khác.
* **fastify-helmet**: Đây là một dịch vụ bảo mật được tích hợp sẵn, cung cấp các cài đặt bảo mật HTTP header mặc định, giúp bảo vệ ứng dụng của bạn khỏi các cuộc tấn công như Cross-Site Scripting (XSS), clickjacking và nhiều loại tấn công khác.
* **fastify-rate-limit**: Dịch vụ này giúp bạn áp đặt giới hạn tốc độ truy cập cho các yêu cầu đến ứng dụng của mình, giúp ngăn chặn tấn công từ phía máy khách hoặc giảm bớt áp lực lên máy chủ.
* **fastify-auth**: Dịch vụ này cung cấp các công cụ để triển khai xác thực và quản lý phiên cho ứng dụng của bạn, bao gồm việc xác định và xác thực người dùng, quản lý phiên và hỗ trợ đăng nhập qua các phương thức khác nhau như OAuth.
* **fastify-compress**: Dịch vụ này cho phép nén dữ liệu trước khi gửi phản hồi HTTP để giảm dung lượng của phản hồi và tăng tốc độ tải trang.
* **fastify-multipart**: Dịch vụ này cung cấp hỗ trợ xử lý các yêu cầu gửi dưới dạng multipart/form-data, thường được sử dụng cho việc tải lên file từ máy khách.
* **fastify-sqlite3**: Đây là một dịch vụ tích hợp để làm việc với cơ sở dữ liệu SQLite, cho phép bạn thực hiện các thao tác cơ bản như truy vấn, chèn, cập nhật và xóa dữ liệu.
* **fastify-jwt**: Dịch vụ này cung cấp hỗ trợ cho JSON Web Tokens (JWT), giúp bạn tạo và xác thực token để bảo vệ các API endpoints của ứng dụng.

### **1.1.4 Các thành phần cơ bản**

1. Ứ*ng dụng (Application)*

*-* Đại diện cho ứng dụng web được xây dựng bằng Fastify.

- Là điểm khởi đầu cho việc định nghĩa và cấu hình các tuyến đường, middleware, và các plugin.

1. *Tuyến đường (Route):*

*-* Định nghĩa cách xử lý yêu cầu HTTP cho một đường dẫn cụ thể.

- Bao gồm một phương thức HTTP (GET, POST, PUT, DELETE, v.v.), một đường dẫn và một hàm xử lý để thực hiện khi yêu cầu được nhận.

1. *Middleware*

*-* Là các hàm xử lý trung gian được thực thi trước hoặc sau khi xử lý yêu cầu trong một tuyến đường.

- Dùng để thực hiện các tác vụ như xác thực, ghi log, nén dữ liệu, xử lý lỗi, và nhiều tác vụ khác.

1. *Plugin*

*-* Cho phép mở rộng chức năng của Fastify bằng cách cung cấp các tính năng mới mà không cần sửa đổi mã nguồn của Fastify.

- ó thể thêm các plugin có sẵn từ Fastify hoặc tạo các plugin tùy chỉnh.

1. *Schema Validation*

*-* Fastify tích hợp hỗ trợ cho validation dựa trên schema, cho phép xác thực dữ liệu đầu vào và đầu ra dễ dàng.

- Sử dụng JSON Schema để định nghĩa các quy tắc validation.

1. *Request và Reply Object*

*-* Fastify cung cấp các đối tượng request và reply để truy cập thông tin của yêu cầu và gửi phản hồi.

- Bao gồm các phương thức và thuộc tính để xử lý yêu cầu và phản hồi.

1. *Logger*

*-* Fastify có tích hợp logging mạnh mẽ, cho phép ghi log các yêu cầu HTTP, lỗi và thông tin khác một cách dễ dàng.

1. *Hooks*

*-* Cho phép thực thi mã trước hoặc sau các sự kiện cụ thể như khi khởi động ứng dụng, xử lý yêu cầu, hoặc gửi phản hồi.

## **Cơ sở dữ liệu MySQL**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, được sử dụng rộng rãi trên toàn thế giới. Nó cung cấp một hệ thống quản lý dữ liệu mạnh mẽ, hỗ trợ các thao tác CRUD (Create, Read, Update, Delete), truy vấn SQL phức tạp, và có khả năng xử lý một lượng lớn dữ liệu một cách hiệu quả. MySQL thường được sử dụng cho các ứng dụng web và được ưa chuộng bởi tính ổn định và hiệu năng cao.

## **Form và các đối tượng của Form**

### **Giới thiệu**

Trong lĩnh vực công nghệ thông tin và phát triển phần mềm, "Form" thường được hiểu là một giao diện người dùng (UI) mà người dùng có thể sử dụng để nhập dữ liệu hoặc tương tác với một ứng dụng hoặc trang web. Form thường bao gồm các thành phần như các trường nhập liệu (text box, drop-down menu, radio button, checkbox), nút điều khiển (button), và các phần tử khác như vùng văn bản (text area), hộp kiểm (check box),

### **GET**

Phương thức GET được sử dụng để yêu cầu dữ liệu từ một nguồn đã biết, thường là từ một trang web hoặc một tài nguyên trên máy chủ web. Dưới đây là một số điểm chính về phương thức GET:

- **Yêu cầu dữ liệu**: Phương thức GET được sử dụng khi trình duyệt muốn yêu cầu một trang web hoặc một tài nguyên từ máy chủ web. Các thông tin yêu cầu (như tham số truy vấn) được gửi kèm theo URL.

**- Dễ nhìn thấy và hiểu được**: Do dữ liệu được truyền trực tiếp qua URL, nên các thông tin gửi đi dễ dàng nhìn thấy và hiểu được. Tuy nhiên, điều này cũng có thể gây ra một số vấn đề bảo mật nếu dữ liệu quan trọng được truyền qua phương thức GET.

- **Thích hợp cho các yêu cầu đơn giản**: Phương thức GET thích hợp cho các yêu cầu đơn giản như lấy dữ liệu từ cơ sở dữ liệu, hiển thị trang web, ...

- **Có hạn chế về dung lượng**: Do dữ liệu được truyền qua URL, nên phương thức GET có hạn chế về dung lượng dữ liệu có thể truyền đi, vì URL có một giới hạn kích thước nhất định.

### **POST**

Phương thức POST là một trong những phương thức chính trong giao thức HTTP (Hypertext Transfer Protocol), cùng với phương thức GET. Phương thức POST được sử dụng để gửi dữ liệu từ trình duyệt web đến máy chủ web để xử lý, thường được sử dụng khi người dùng muốn gửi dữ liệu biểu mẫu (form data) hoặc thực hiện các thao tác như tạo, cập nhật hoặc xóa dữ liệu. Dưới đây là một số điểm chính về phương thức POST:

- **Gửi dữ liệu ẩn**: Dữ liệu được gửi từ trình duyệt đến máy chủ thông qua phần body của yêu cầu HTTP, không hiển thị trực tiếp trên URL như phương thức GET. Điều này làm cho phương thức POST thích hợp cho việc gửi dữ liệu nhạy cảm hoặc lớn.

- **Dữ liệu không giới hạn**: So với phương thức GET, phương thức POST không có giới hạn về kích thước dữ liệu có thể gửi đi, do dữ liệu được gửi qua phần body của yêu cầu HTTP.

- **Bảo mật hơn**: Do dữ liệu không được hiển thị trên URL, nên phương thức POST thường được coi là bảo mật hơn so với phương thức GET. Điều này làm cho nó thích hợp cho việc gửi dữ liệu như mật khẩu hay thông tin tài khoản người dùng.

- **Thích hợp cho các yêu cầu phức tạp**: Phương thức POST thường được sử dụng cho các yêu cầu phức tạp như gửi dữ liệu biểu mẫu có nhiều trường hoặc yêu cầu xử lý phức tạp từ phía máy chủ.

- **Thích hợp cho các thao tác thay đổi dữ liệu**: Phương thức POST thường được sử dụng cho các thao tác như tạo mới, cập nhật hoặc xóa dữ liệu trên máy chủ.

### **DELETE**

Phương thức DELETE là một trong các phương thức HTTP chuẩn được sử dụng để yêu cầu xóa một tài nguyên (resource) cụ thể từ máy chủ web. Phương thức này thường được sử dụng trong các ứng dụng web hoặc API để thực hiện các thao tác xóa dữ liệu từ máy chủ. Dưới đây là một số điểm chính về phương thức DELETE:

- **Xóa tài nguyên**: Phương thức DELETE được sử dụng để gửi yêu cầu từ máy khách (client) đến máy chủ (server) để yêu cầu xóa một tài nguyên cụ thể trên máy chủ. Tài nguyên này có thể là một bản ghi trong cơ sở dữ liệu, một tệp hoặc bất kỳ nguồn tài nguyên nào khác trên máy chủ.

- **Bảo mật và xác thực**: Phương thức DELETE thường yêu cầu xác thực để đảm bảo rằng người dùng chỉ có thể xóa tài nguyên mà họ có quyền truy cập. Điều này giúp bảo vệ dữ liệu và ngăn chặn việc xóa các tài nguyên không được phép.

- **Sử dụng trong RESTful API**: Phương thức DELETE thường được sử dụng trong kiến trúc RESTful API để thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên tài nguyên.

## **RESTful API**

### **Khái niệm**

RESTful API (Representational State Transfer API) là một kiểu thiết kế API cho các ứng dụng web, dựa trên các nguyên tắc của kiến trúc REST. RESTful API giúp quản lý, truy xuất và thay đổi dữ liệu thông qua giao thức HTTP một cách hiệu quả và linh hoạt.

### **Các thành phần**

**Resource (Tài nguyên)**: Là các đối tượng hoặc dữ liệu mà bạn muốn truy xuất hoặc thao tác trong ứng dụng của mình. Mỗi resource được đại diện bằng một URI (Uniform Resource Identifier), ví dụ: /users, /products, /articles.

**Stateless (Không lưu trạng thái)**: RESTful API không lưu trạng thái của client giữa các yêu cầu. Mỗi yêu cầu từ client chứa đủ thông tin để server có thể hiểu và xử lý yêu cầu một cách độc lập.

**Hypermedia (Hypermedia)**: Một phần mở rộng của khái niệm liên kết, cho phép các liên kết được trả về không chỉ mô tả mối quan hệ giữa các resource mà còn cung cấp thông tin về cách thực hiện các hành động tiếp theo.

**HTTP Methods (Phương thức HTTP):** RESTful API sử dụng các phương thức HTTP như GET, POST, PUT, PATCH, DELETE để thực hiện các hoạt động trên resource:

- GET: Truy vấn dữ liệu từ resource.

- POST: Tạo mới resource.

- PUT: Cập nhật hoặc thay đổi toàn bộ resource.

- PATCH: Cập nhật một phần của resource.

- DELETE: Xóa resource.

**Representation (Biểu diễn)**: Resource có thể được biểu diễn dưới nhiều dạng khác nhau như JSON, XML, HTML, hoặc các định dạng khác. Mỗi loại biểu diễn đều có thể được yêu cầu bằng cách sử dụng các HTTP headers như "Content-Type" và "Accept".

**Stateless (Không lưu trạng thái)**: RESTful API không lưu trạng thái của client giữa các yêu cầu. Mỗi yêu cầu từ client chứa đủ thông tin để server có thể hiểu và xử lý yêu cầu một cách độc lập.

### RESTful API hoạt động



*Hình 1.1 : Nguyên lý hoạt động restful api*

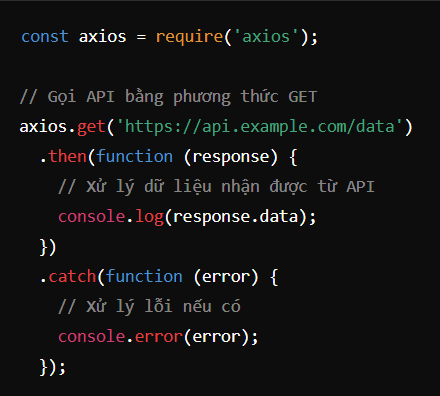
Cách thức hoạt động cơ bản của RESTful API dựa trên giao thức HTTP. Mỗi hoạt động khác nhau cơ bản sẽ có những phương thức HTTP riêng.

* **GET (SELECT)**: Trả về một Resource hoặc một danh sách Resource.
* **POST (CREATE)**: Tạo mới một Resource.
* **PUT (UPDATE)**: Cập nhật thông tin cho Resource.
* **DELETE (DELETE):** Xóa một Resource.

CRUD là cụm từ để chỉ những phương thức hay hoạt động tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

### **Cách gọi API bằng thư viện Axios**

Axios là một thư viện JavaScript dùng để thực hiện các yêu cầu HTTP. Dưới đây là ví dụ cách sử dụng Axios để gọi API:



*Hình 1.2: Cách sử dụng Axios để gọi API*

## **REACT JS**

### **Tổng quan về React JS**

React.js là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Nếu như AngularJS là một Framework cho phép nhúng code javasscript trong code html thông qua các attribute như ng-model, ng-repeat...thì với react là một library cho phép nhúng code html trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS.Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn. React cho phép bạn tạo những giao diện (UI) phức tạp từ những đoạn code nhỏ và độc lập. Những đoạn code này được gọi là “components”.

### **Các thành phần cơ bản của React JS**

Trước khi đến cài đặt và cấu hình, chúng ta sẽ đi đến một số khái niệm cơ bản trong Reactjs:

1. *Giới thiệu về JSX*

JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. Đặc điểm: Faster: Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javacsript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript Safer an toàn hơn. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt Easier Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng.

1. *Virtual DOM:*

* Công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này. Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM,

khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM tree thật. React sử dụng cơ chế one-way data binding – luồng dữ liệu 1 chiều. Dữ liệu được truyền từ parent đến child thông qua props. Luồng dữ liệu đơn giản giúp chúng ta dễ dàng kiểm soát cũng như sửa lỗi.

* Với các đặc điểm ở trên, React dùng để xây dựng các ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện. Ví dụ như Facebook: trên Newsfeed của bạn cùng lúc sẽ có các status khác nhau và mỗi status lại có số like, share, comment liên tục thay đổi. Khi đó React sẽ rất hữu ích để sử dụng.

1. *Giới thiệu về Components*

React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác. Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component. Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo.

1. *Props và State*:

* Props: giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến.
* State: thể hiện trạng thái của ứng dụng, khi state thay đồi thì component đồng thời render lại để cập nhật UI.

# Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## **2.1 Khảo sát hiện trạng và yêu cầu**

### **2.1.1 Khảo sát hiện trạng**

**Về khách hàng:** Đầu tiên tính thẩm mỹ của website có vai trò tạo ấn tượng và thu hút khách hàng, nhưng để giữ chân khách hàng và khiến họ trở lại với website những lần sau được quyết định bởi nội dung trình diễn trên website. Khách hàng sẽ cảm thấy thỏa mãn khi cửa hàng trực tuyến có khả năng đáp ứng các nhu cầu sau đây: tìm sản phẩm nhanh, sản phẩm có đủ thông tin để lựa chọn, mua hàng nhanh chóng.

**Về người quản trị:**

* **Tạo nội dung, cập nhật dữ liệu phù hợp:** người quản trị website sau khi tiếp nhận phải là người tạo nội dung hay chỉnh sửa, cập nhật những nội dung cho hệ thống.
* **Kiểm tra tình trạng đơn hàng và quản lý người dùng: Người quản trị có thể kiểm tra được tình trạng đơn hàng để có thể xử lý các yêu cầu người dùng về đơn hàng.**
* **Thống kê doanh thu: Doanh thu thể hiện sự phát triển của doanh nghiệp. Từ đó tập trung phát triển các mặt hàng được khách hàng tiêu dùng yêu cầu hoặc sử dụng nhiều nhất.**

## **2.2 Phân tích hệ thống**

### **2.2.1 Xác định các Actor**

**Khách hàng:**

* Xem thông tin và mua sắm các sản phẩm được đăng bán trên web.
* Đăng nhập, đăng ký thành viên.
* Quản lý thông tin cá nhân.
* Đánh giá về các sản phẩm, dịch vụ, ...tương tác với người bán.

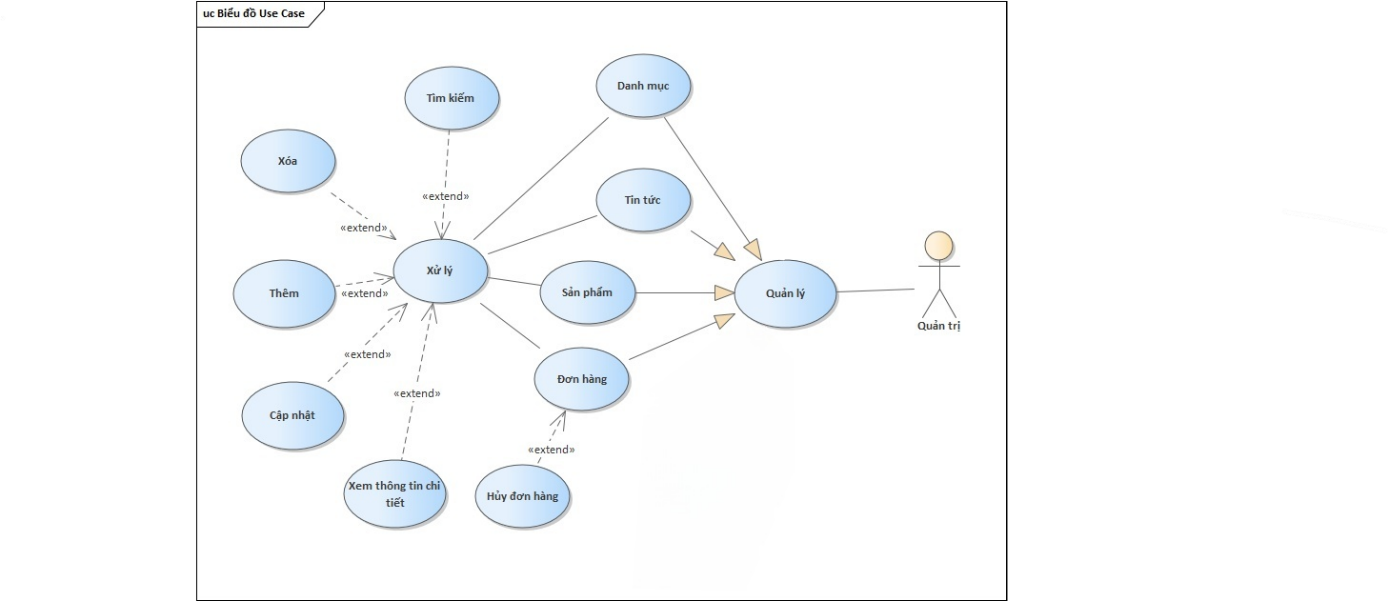
**Quản trị viên:**

* Quản lý thông tin sản phẩm, danh mục, tin tức, ...
* Cập nhật giao diện website
* Thêm, xóa, cập nhật dữ liệu.
* Quản lý nội dung website.

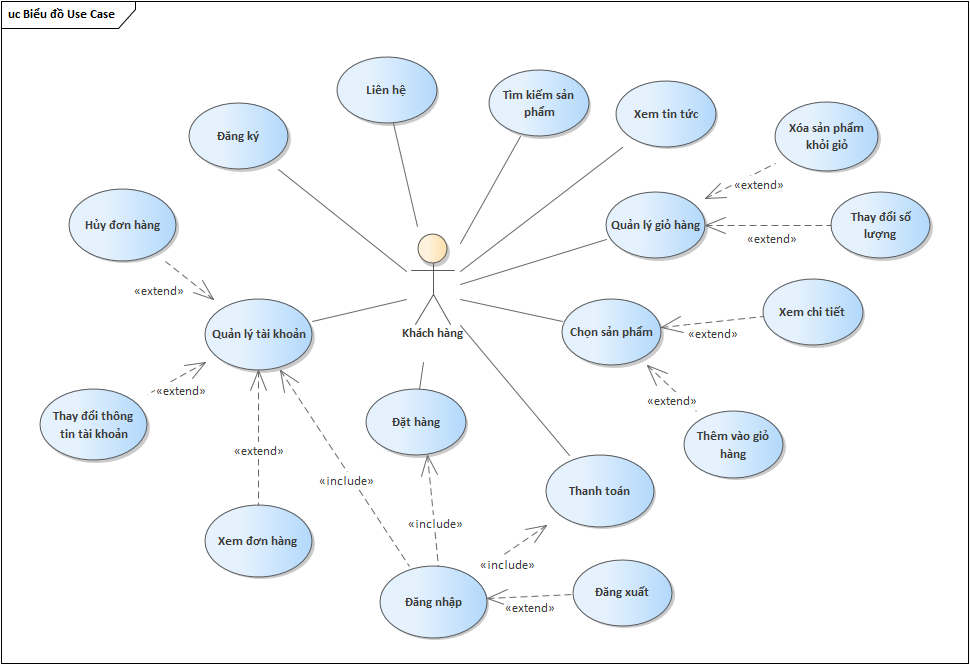
### **2.2.2 Xác định các Use Case**

* **Tìm kiếm sản phẩm**: Người dùng có thể tìm kiếm các sản phẩm thú cưng bằng cách nhập từ khóa vào thanh tìm kiếm. Họ cũng có thể sử dụng bộ lọc để hạn chế kết quả tìm kiếm dựa trên các tiêu chí như thương hiệu, danh mục, v.v.
* **Xem chi tiết sản phẩm**: Người dùng có thể xem thông tin chi tiết của một sản phẩm bao gồm hình ảnh, mô tả, thông số kỹ thuật, giá cả, v.v.
* **Thêm vào giỏ hàng**: Người dùng có thể thêm sản phẩm vào giỏ hàng để sau đó tiến hành thanh toán.
* **Quản lý giỏ hàng**: Người dùng có thể xem, cập nhật hoặc xóa các sản phẩm trong giỏ hàng trước khi tiến hành thanh toán.
* **Thanh toán**: Người dùng có thể chọn phương thức thanh toán và nhập thông tin thanh toán để hoàn tất giao dịch mua hàng.
* **Quản lý tài khoản**: Người dùng có thể đăng ký, đăng nhập và quản lý thông tin cá nhân như địa chỉ giao hàng, thông tin thanh toán, lịch sử đơn hàng, v.v.
* **Quản lý đơn hàng**: Người dùng có thể xem trạng thái của các đơn hàng của mình, theo dõi giao hàng và cập nhật thông tin đơn hàng.
* **Quản trị sản phẩm**: Quản trị viên có thể thêm, sửa đổi hoặc xóa sản phẩm từ cửa hàng trực tuyến, cũng như quản lý thông tin sản phẩm.
* **Quản lý đơn hàng (admin)**: Quản trị viên có thể xem và quản lý các đơn hàng được đặt từ trang web, bao gồm xác nhận đơn hàng, cập nhật trạng thái giao hàng, v.v.
* **Xem sản phẩm liên quan**: Khi xem chi tiết một sản phẩm, người dùng có thể thấy các sản phẩm liên quan hoặc gợi ý để tăng khả năng tìm kiếm và mua hàng.
* **Xem sản phẩm nổi bật**: Người dùng có thể xem danh sách các sản phẩm nổi bật, giảm giá hoặc mới nhất trên trang web.
* **Xem tin tức**: Người dùng có thể xem danh sách các tin tức mới nhất, chi tiết bài viết hoặc các bài viết liên quan trên trang web.

### **2.2.3 Sơ đồ Use Case**



*Hinh 2.1: Biểu đồ use case tác nhân quản trị viên*



*Hình 2.2: Biểu đồ use case tác nhân khách hàng*

### **2.2.4 Phân tích**

Bên trang người dùng:

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả** |
| 1 | Đăng ký | Khách hàng đăng ký thành viên để thực hiện thao tác giao dịch. |
| 2 | Đăng nhập | Khách hàng đăng nhập vào website bằng tài khoản mật khẩu đã đăng ký trước đó. |
| 3 | Tìm kiếm sản phẩm | Khách hàng nhập thông tin cần tìm vào thanh tìm kiếm hoặc tìm kiếm theo danh mục, thương hiệu. |
| 4 | Xem chi tiết sản phẩm | Khách hàng có thể xem chi tiết sản phẩm bao gồm như: Tên sản phẩm, giá, thương hiệu, mô tả, tình trạng… |
| 5 | Giỏ hàng | Cho phép người dùng xem các sản phẩm đã chọn, thay đổi số lượng mua hoặc xóa sản phẩm khỏi giỏ hàng. Bao gồm tính tổng giá sản phẩm khách đã chọn. |
| 6 | Đặt hàng | Khách hàng click đặt hàng các sản phẩm nằm trong giỏ hàng, hệ thống báo đặt hàng thành công và gửi thông báo xác nhận đặt hàng. |
| 7 | Tin tức | Người xem có thể xem các tin tức mới về hệ thống bán hàng. |
| 8 | Quản lý tài khoản | Khách hàng có thể xem thông tin tài khoản, thay đổi thông tin tài khoản và xem các đơn hàng đã đặt. |
| 9 | Xem sản phẩm | Hiển thị các sản phẩm mới, nỗi bật, giảm giá,. . . lên website. |

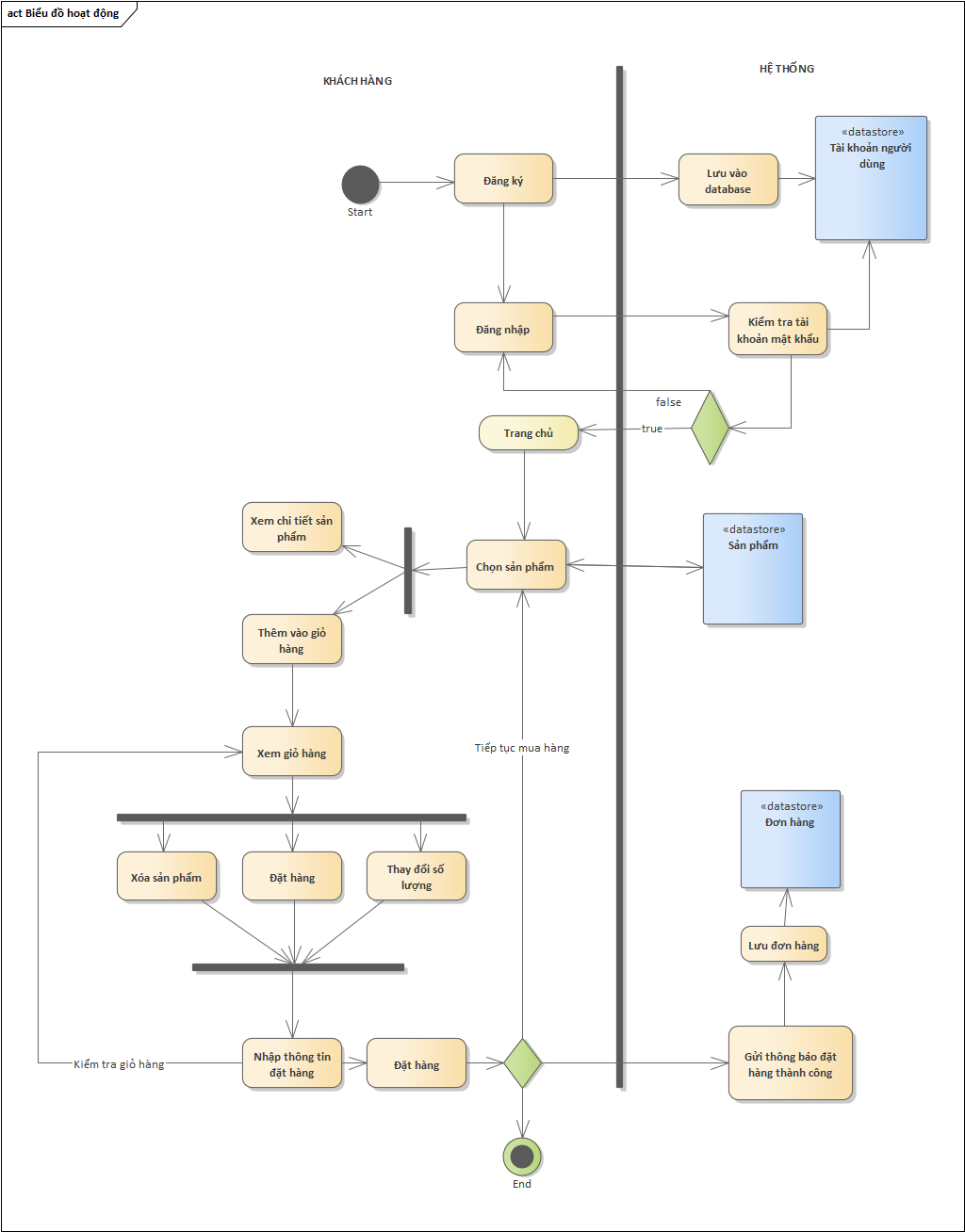
Bên trang quản trị

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả** |
| 1 | Đăng nhập | Đăng nhập vào hệ thống bằng tài khoản mặt khẩu đã thiết lập quản trị để thao tác các chức năng quản lý |
| 2 | Quản lý sản phẩm | Quản lý thông tin sản phẩm bao gồm: Thêm, xóa, cập nhật các thông tin sản phẩm |
| 3 | Quản lý danh mục sản phẩm | Quản lý danh mục bao gồm các chức năng: Thêm, xóa, sửa, cập nhập |
| 4 | Quản lý đơn hàng | Thống kê các đơn hàng, xem tình trạng (Đã giao/ chưa giao) và chi tiết đơn hàng |
| 5 | Chi tiết đơn hàng | Hiển thị chi tiết đơn hàng bao gồm thông tin khách hàng, mã đơn hàng, các sản phẩm đặt và giá. |
| 6 | Quản lý người dùng | Xem chi tiết, khóa tài khoản thông tin người dùng |
| 7 | Quản lý tin tức | Quản lý tin tức bao gồm các chức năng: Thêm, xóa, sửa, cập nhập tin tức |
| 8 | Quản lý sản phẩm giảm giá | Quản lý thông tin sản phẩm giảm giá bao gồm: Thêm, xóa, cập nhật các thông tin sản phẩm giảm giá. |
| 9 | Quản lý thương hiệu | Quản lý thông tin thương hiệu bao gồm: Thêm, xóa, cập nhật các thông tin của thương hiệu |
| 10 | Quản lý chủ đề bài viết | Quản lý thông tin chủ đề bài viết bao gồm: Thêm, xóa, cập nhật các thông tin của chủ đề bài viết |

## **2.3 Thiết kế**

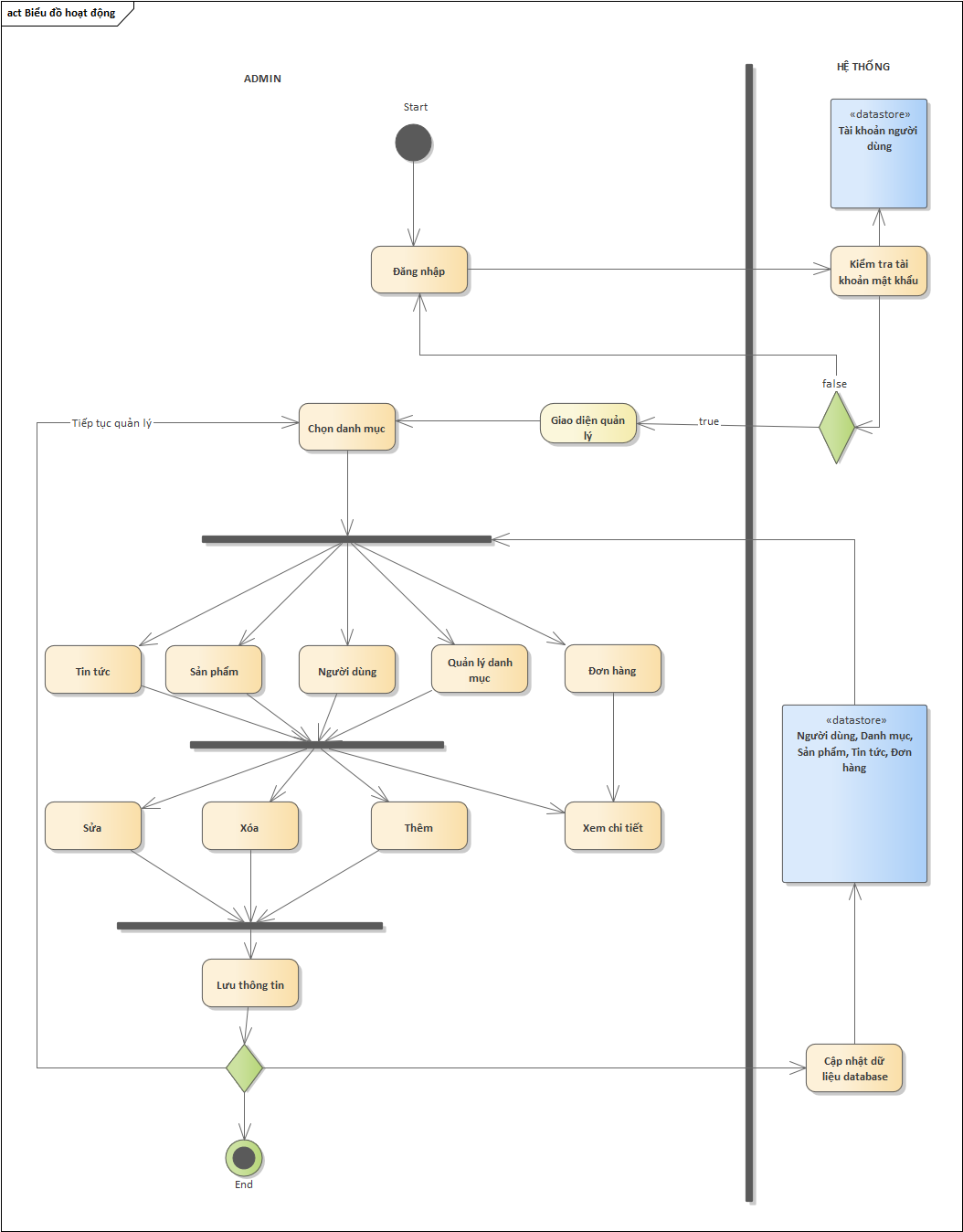
### **2.3.1 Biểu đồ hoạt động tổng quát**

2.3.1.1 Tác nhân khách hành



*Hình 2.1. Biểu đồ hoạt động tổng quát tác nhân khách hàng*

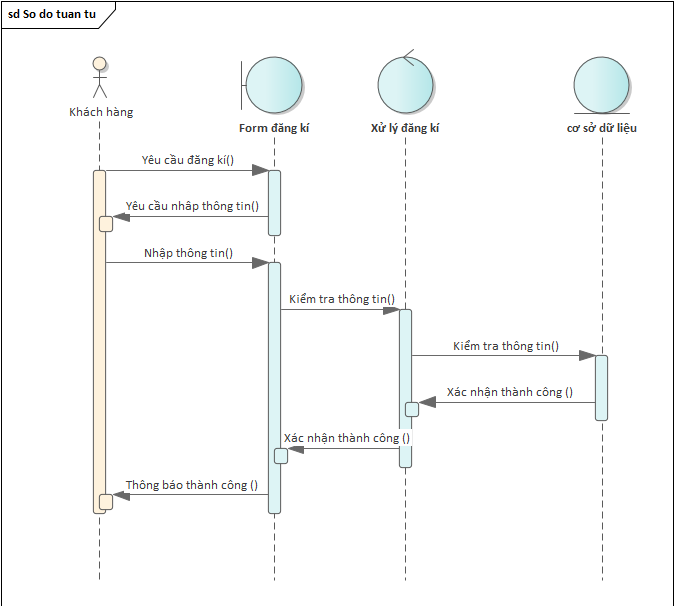
#### 2.3.1.2 Tác nhân quản trị viên



*Hình 2.2. Biểu đồ hoạt động tổng quát tác nhân quản trị viên*

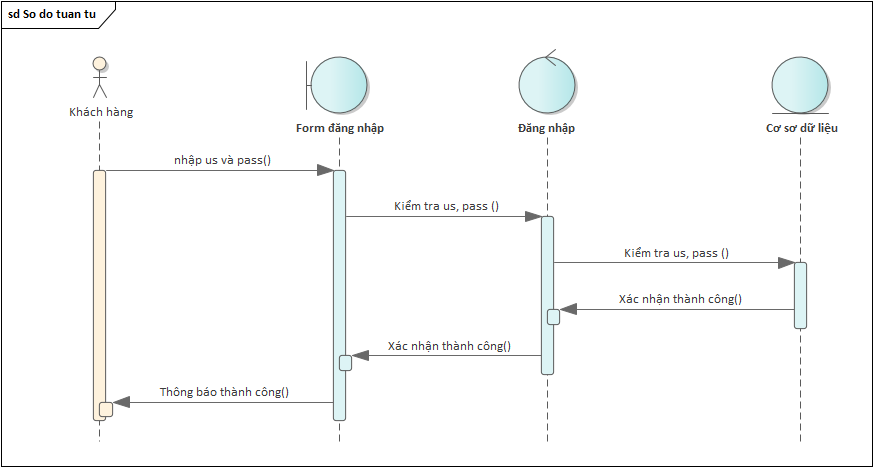
### **2.3.2 Biểu đồ tuần tự**

#### 2.3.2.1 Biểu đồ tuần tự chức năng đăng ký



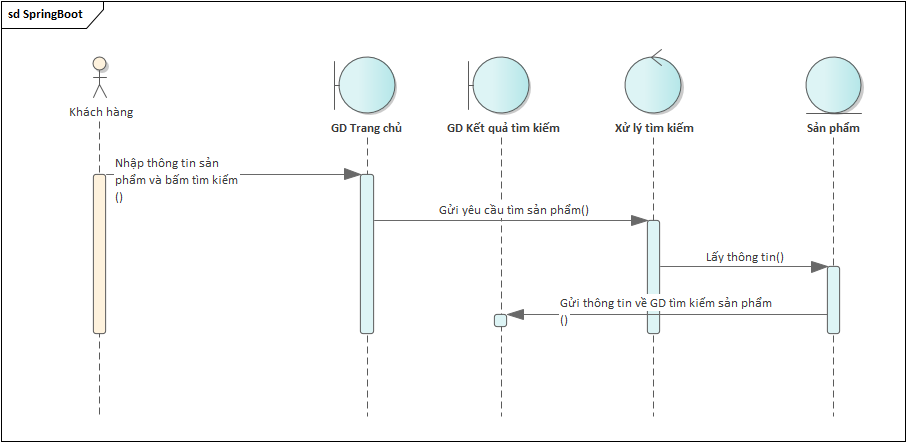
*Hình 2.3. Biểu đồ tuần tự chức năng đăng đăng ký*

#### 2.3.2.2 Biểu đồ tuần tự chức năng đăng nhập



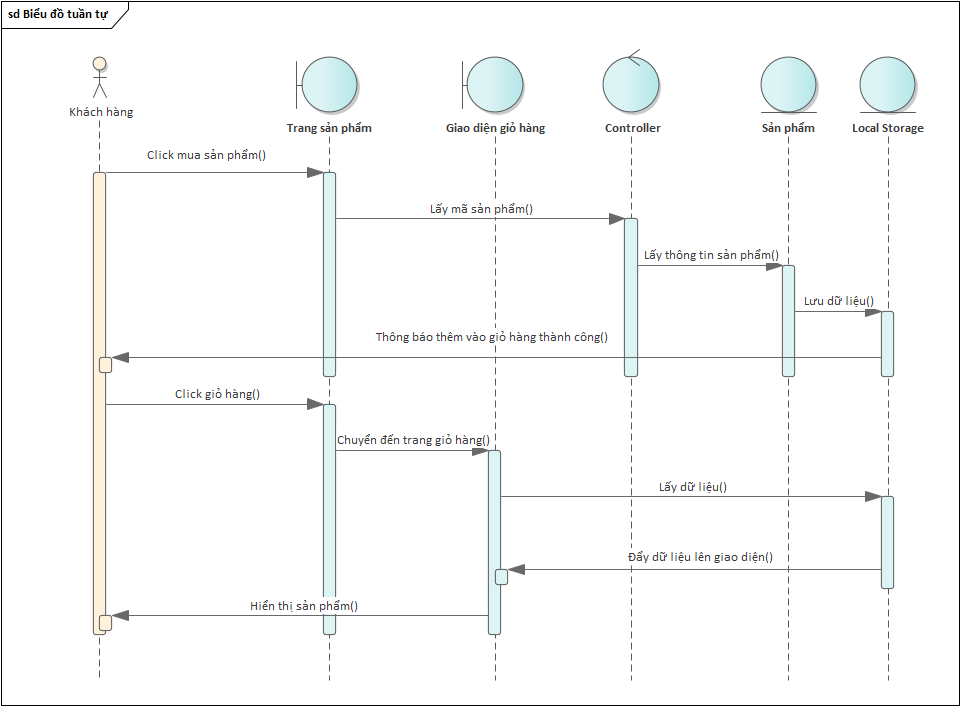
*Hình 2.4. Biểu đồ tuần tự chức năng đăng nhập*

#### 2.3.2.3 Biểu đồ tuần tự chức năng tìm kiếm sản phẩm



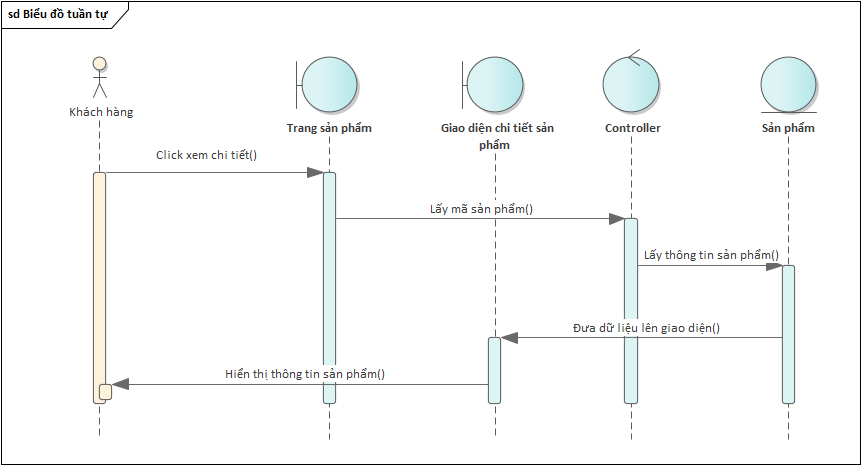
*Hình 2.5. Biểu đồ tuần tự chức năng tìm kiếm sản phẩm*

#### 2.3.2.4 Biểu đồ tuần tự chức năng thêm sản phẩm vào giỏ hàng



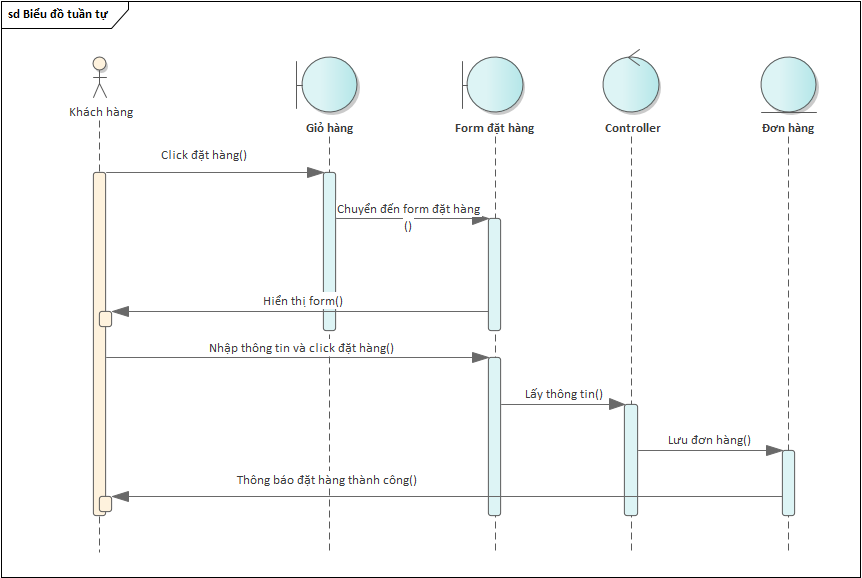
*Hình 2.6. Biểu đồ tuần tự chức năng thêm sản phẩm vào giỏ hàng*

#### 2.3.2.5 Biểu đồ tuần tự chức năng xem chi tiết sản phẩm



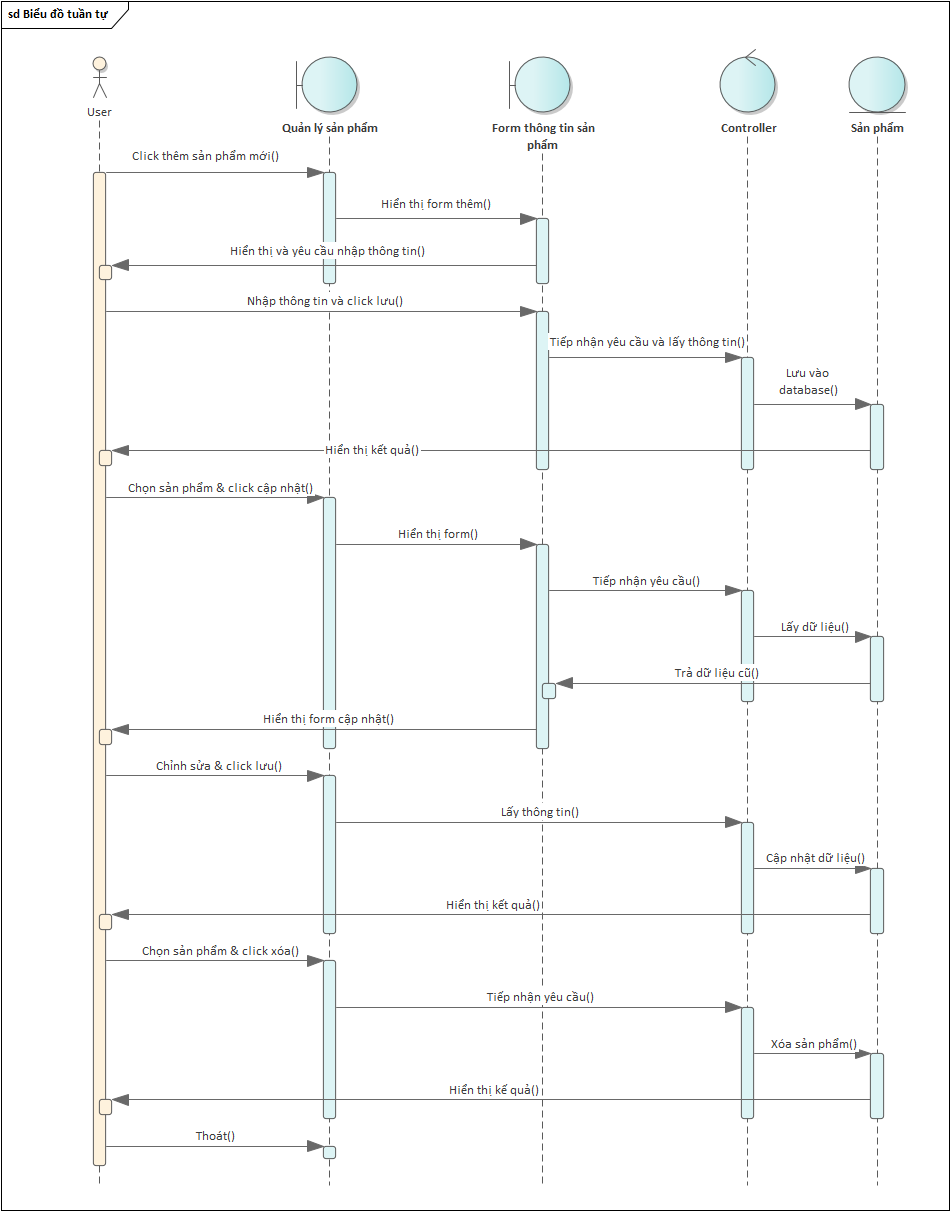
*Hình 2.7. Biểu đồ tuần tự chức năng xem chi tiết sản phẩm*

#### 2.3.2.6 Biểu đồ tuần tự chức năng đặt hàng



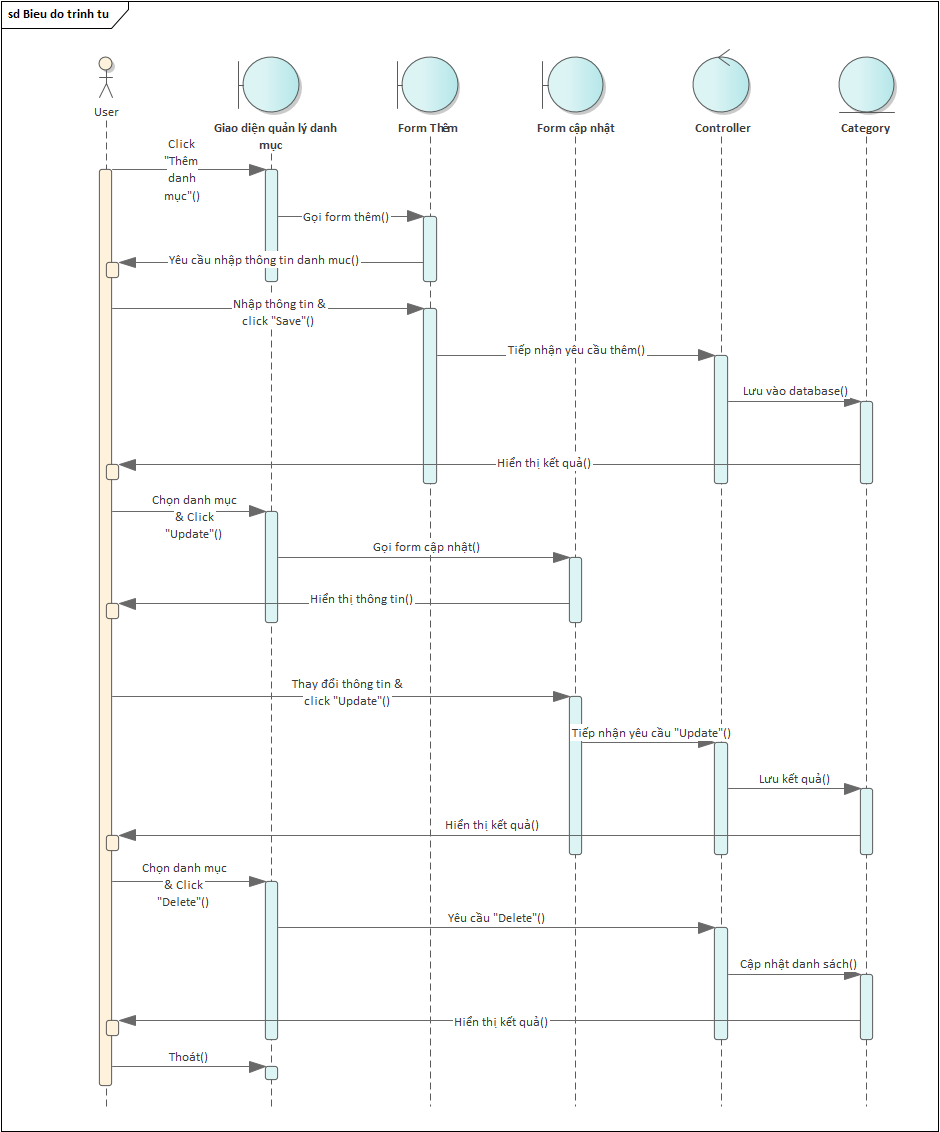
*Hình 2.8. Biểu đồ tuần tự chức năng đặt hàng*

#### 2.3.2.7 Biểu đồ tuần tự chức năng quản lý sản phẩm



*Hình 2.9. Biểu đồ tuần tự chức năng quản lý sản phẩm*

#### 2.3.2.7 Biểu đô tuần tự chức năng quản lý danh mục



*Hình 2.10. Biểu đồ tuần tự chức năng quản lý danh mục*

**Chương 3: GIAO DIỆN HỆ THỐNG**

Trong chương trình, giao diện chia ra làm 2 phần: giao diện trang quản trị và giao diện trang người dùng.

## 3.1 Giao diện trang quản trị



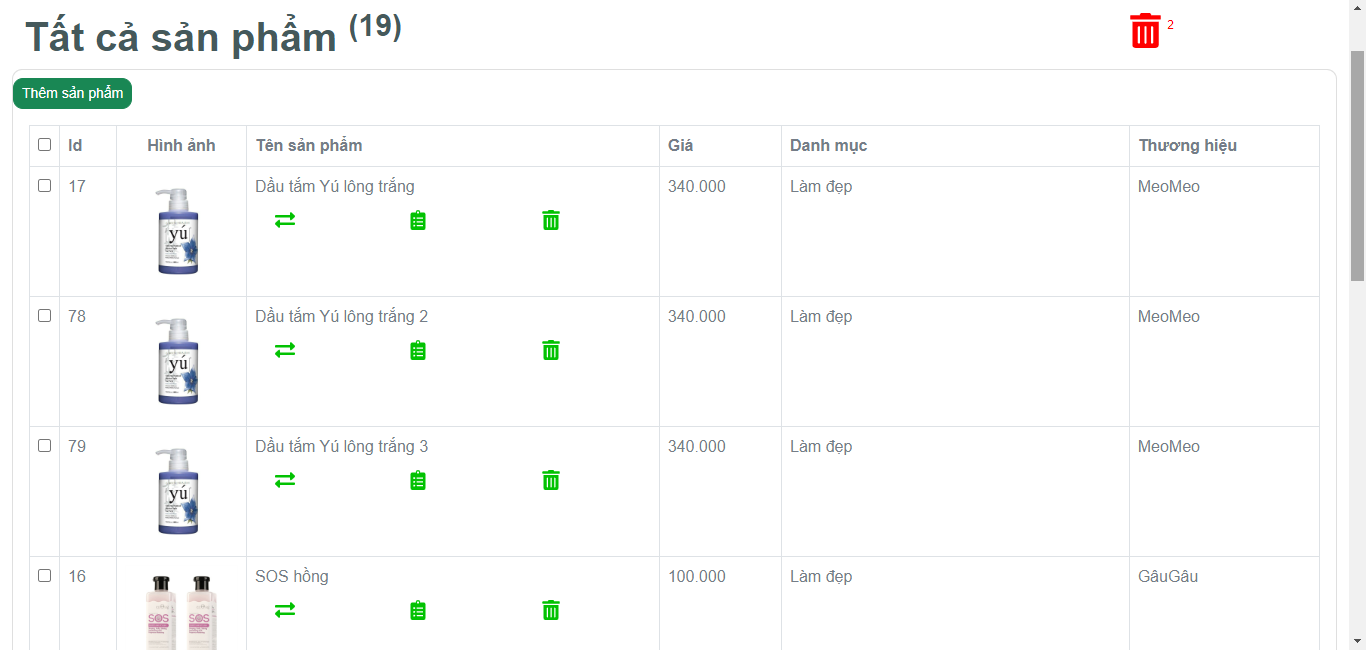
**3.1.1 Trang quản lý sản phẩm**

* Đây là trang giao diện quảng trị

*Hình 3. 2 Đăng nhập thành công*

**3.1.2 Trang quản lý sản phẩm**

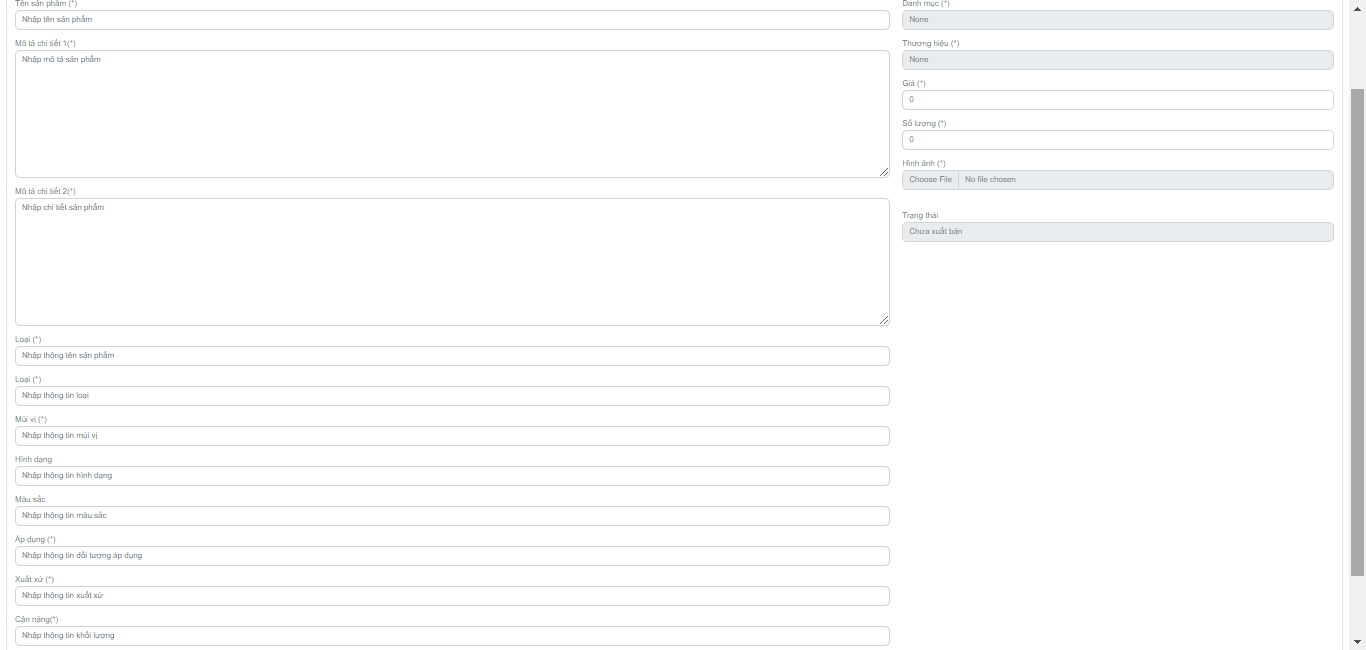
Trong phần quản lý sản phẩm, danh sách các sản phẩm sẽ được hiển thị theo từng dòng nhằm tối ưu hóa màn hình hiển thị. Khi sản phẩm quá nhiều sẽ được phân trang.



*Hình 3. 3 Trang quản lý sản phẩm*

Trên màn hình danh sách sản phẩm (hình 3.3), khi người dùng cần thêm mới, nhấn nút “**Thêm sản phẩm”**, màn hình sẽ chuyển sang trang thêm mới sản phẩm với các trường thông tin cần nhập như hình 3.4.

* **Trang thêm mới sản phẩm**



*Hình 3. 4 Thêm mới sản phẩm*

* **Trang chỉnh sửa sản phẩm**

Trên màn hình danh sách sản phẩm (hình 3.3), khi người dùng cần xem lại thông tin hoặc chỉnh sửa sản phẩm, ở cột chức năng, người dùng có thể nhấn vào nút **“chỉnh sửa”** từ đó trang theo chức năng cần sử dụng.



*Hình 3. 5 Chỉnh sửa sản phẩm*

* **Thùng rác sản phẩm**

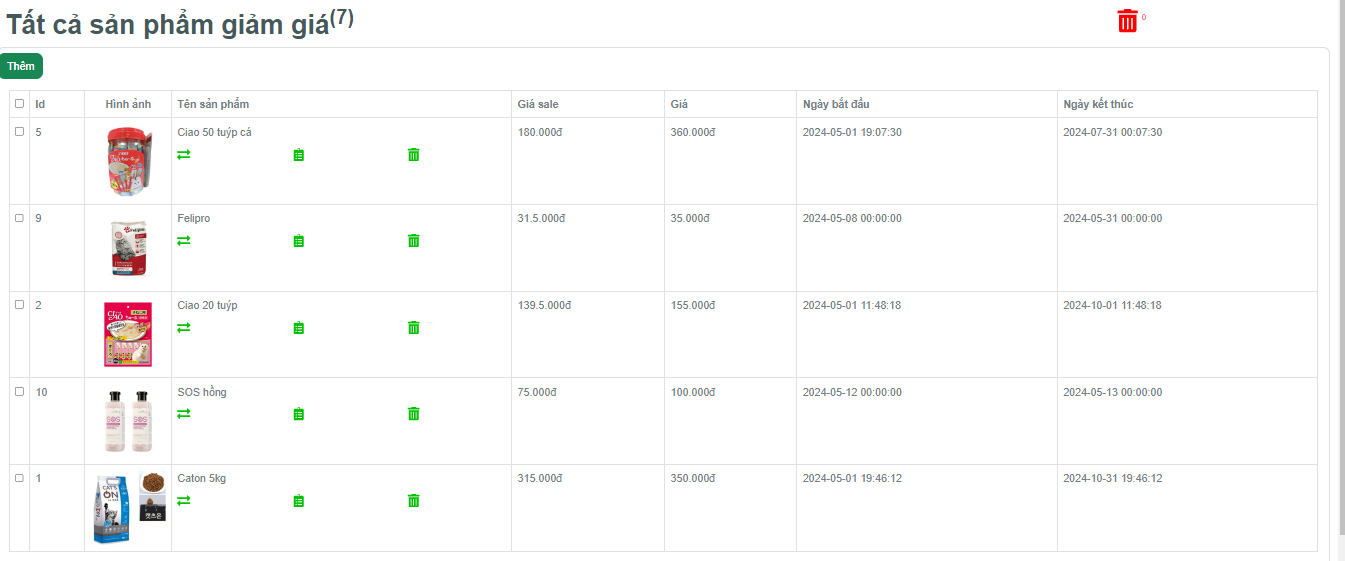
Trên màn hình danh sách sản phẩm (hình 3.3), khi người dùng cần xem lại danh sách sản phẩm đã xoá trước đó, người dùng có thể nhấn nút **“Thùng rác”**, màn hình sẽ chuyển sang trang thùng rác sản phẩm như hình 3.6.



*Hình 3. 6 Thùng rác sản phẩm*

**3.1.3 Trang quản lý sản phẩm giảm giá**

Trong phần quản lý sản phẩm giảm giá, danh sách các sản phẩm sẽ được hiển thị theo từng dòng nhằm tối ưu hóa màn hình hiển thị. Khi sản phẩm quá nhiều sẽ được phân trang.



*Hình 3. 7 Trang quản lý sản phẩm*

Trên màn hình danh sách sản phẩm giảm giá (hình 3.7), khi người dùng cần thêm mới, nhấn nút “**Thêm sản phẩm”**, màn hình sẽ chuyển sang trang thêm mới sản phẩm giảm giá với các trường thông tin cần nhập như hình 3.8 và người dùng có thể chọn sản phẩm trong danh sách sản phẩm để giảm giá.

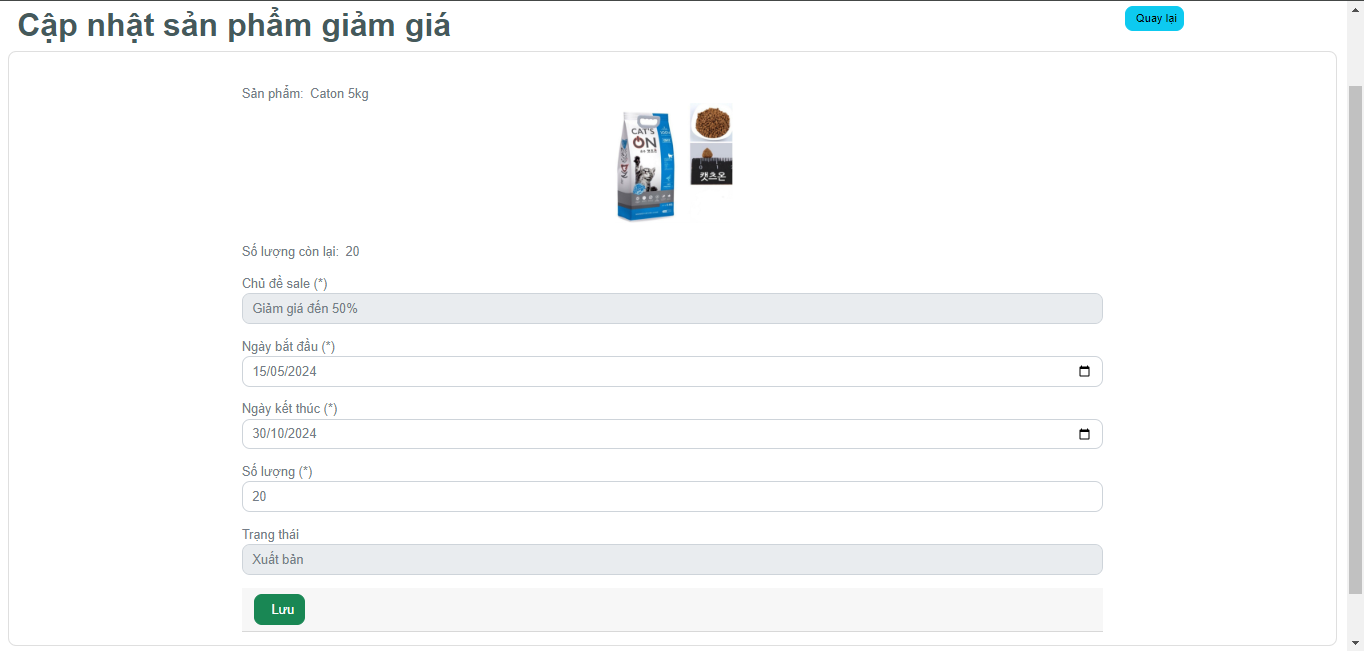
* **Trang thêm mới sản phẩm**



*Hình 3. 8 Thêm mới sản phẩm*

* **Trang chỉnh sửa sản phẩm**

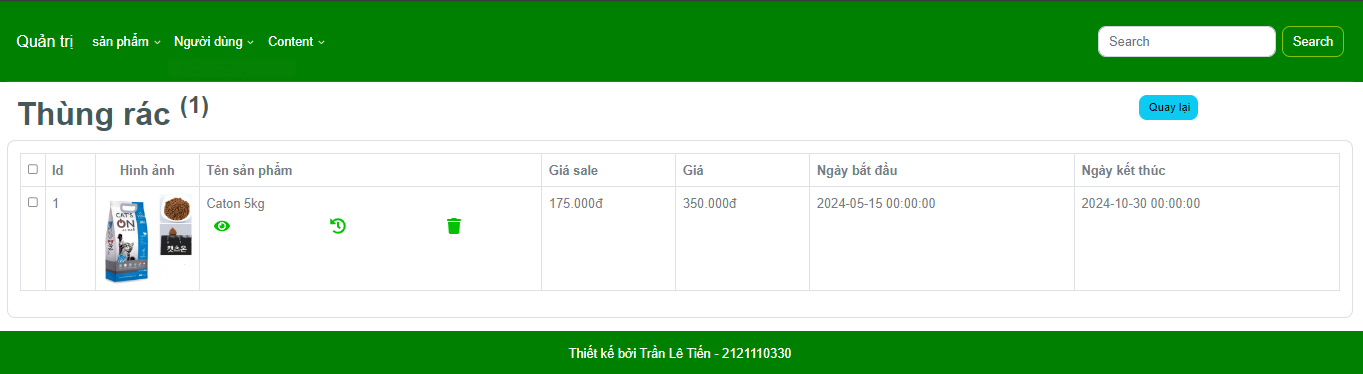
Trên màn hình danh sách sản phẩm giảm giá (hình 3.7), khi người dùng cần xem lại thông tin hoặc chỉnh sửa sản phẩm giảm giá, ở cột chức năng, người dùng có thể nhấn vào nút **“chỉnh sửa”** từ đó trang theo chức năng cần sử dụng.



*Hình 3. 9 Chỉnh sửa sản phẩm*

* **Thùng rác sản phẩm**

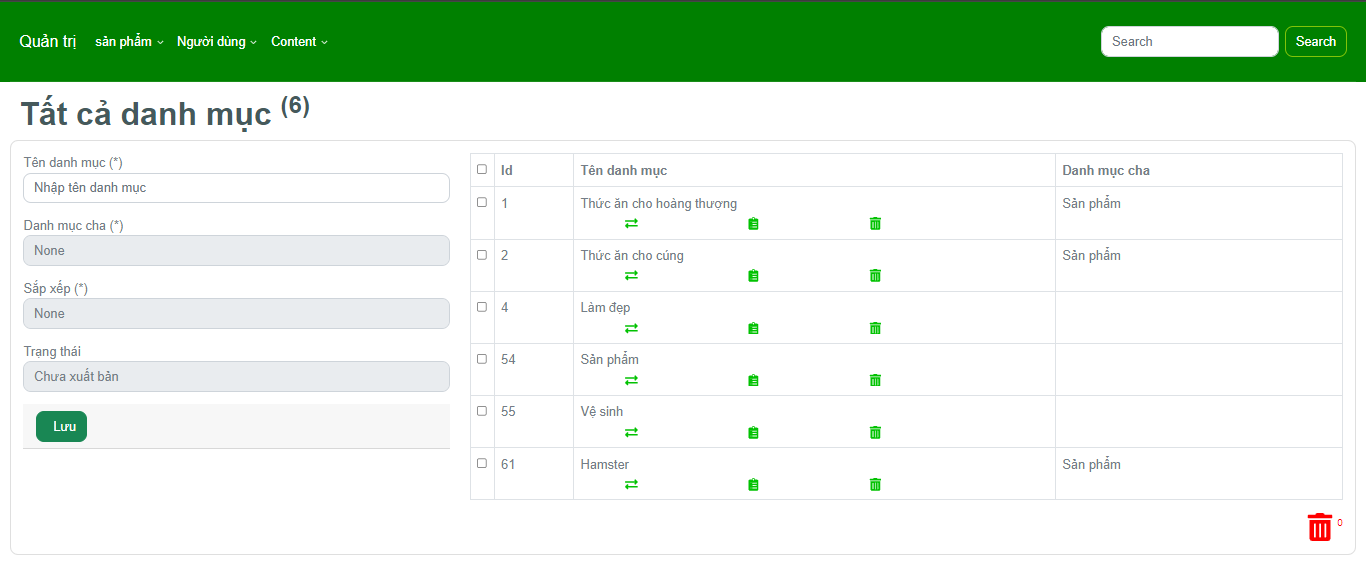
Trên màn hình danh sách sản phẩm giảm giá (hình 3.7), khi người dùng cần xem lại danh sách sản phẩm giảm giá đã xoá trước đó, người dùng có thể nhấn nút **“Thùng rác”**, màn hình sẽ chuyển sang trang thùng rác sản phẩm giảm giá như hình 3.10.



*Hình 3. 10 Thùng rác sản phẩm*

**3.1.4 Trang quản lý danh mục**

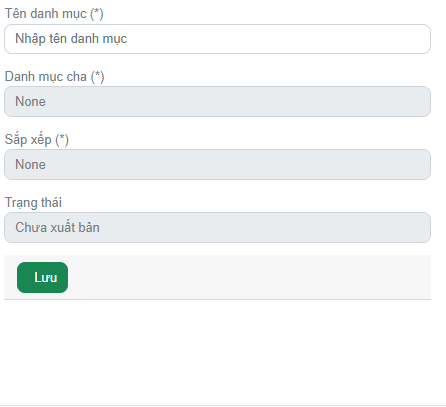
Trang quản danh mục sản phẩm, danh sách loại sản phẩm hiển thị theo từng dòng nhằm tối ưu hóa màn hình hiển thị như hình 3.11.



*Hình 3.11 Quản lý danh mục sản phẩm*

* **Trang thêm mới danh mục sản phẩm**

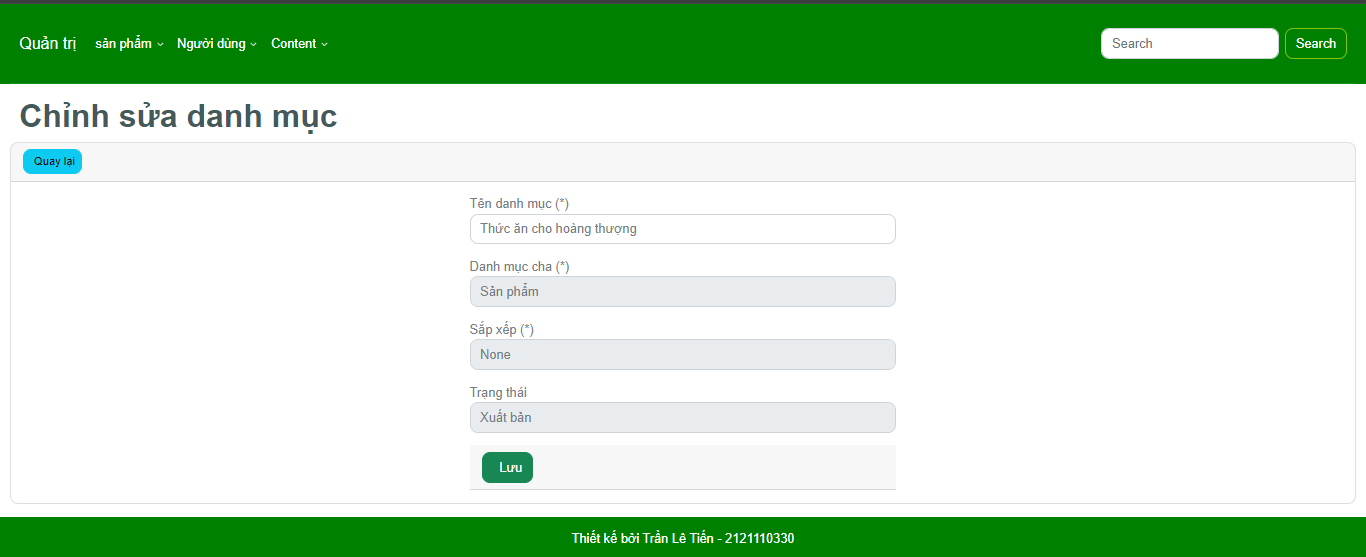
Trên màn hình danh sách sản phẩm (hình 3.11), khi người dùng cần thêm mới, nhấn nút “**Thêm Danh Mục** ” màn hình sẽ xuất hiện bảng thêm mới loại sản phẩm với các trường thông tin cần nhập như hình 3.12, sau đó nhấn nút thêm để thêm mới danh mục sau khi thêm mới màn hình sẽ được trả về trang quản lý danh mục như hình 3.11.



*Hình 3.12 Thêm mới danh mục sản phẩm*

* **Trang chỉnh sửa danh mục**

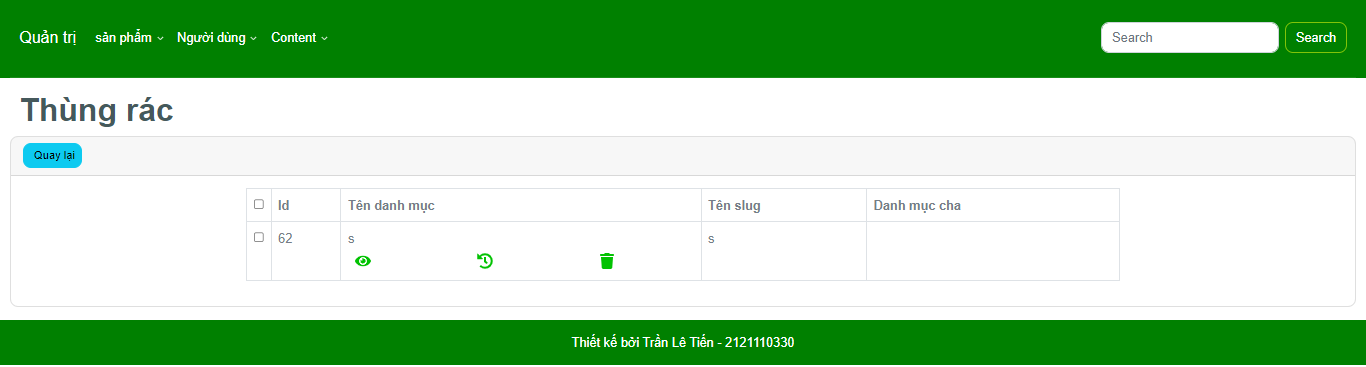
Trên màn hình danh sách sản phẩm (hình 3.11). Khi người dùng cần chỉnh sửa thông tin của danh mục, ở cột chức năng, người dùng có thể nhấn vào nút **“Chỉnh sửa”** để chọn chức năng chỉnh sửa, từ đó sẽ chuyển sang trang chỉnh sửa danh mục như hình 3.13.



*Hình 3. 13 Chỉnh sửa danh mục*

* **Thùng rác danh mục**

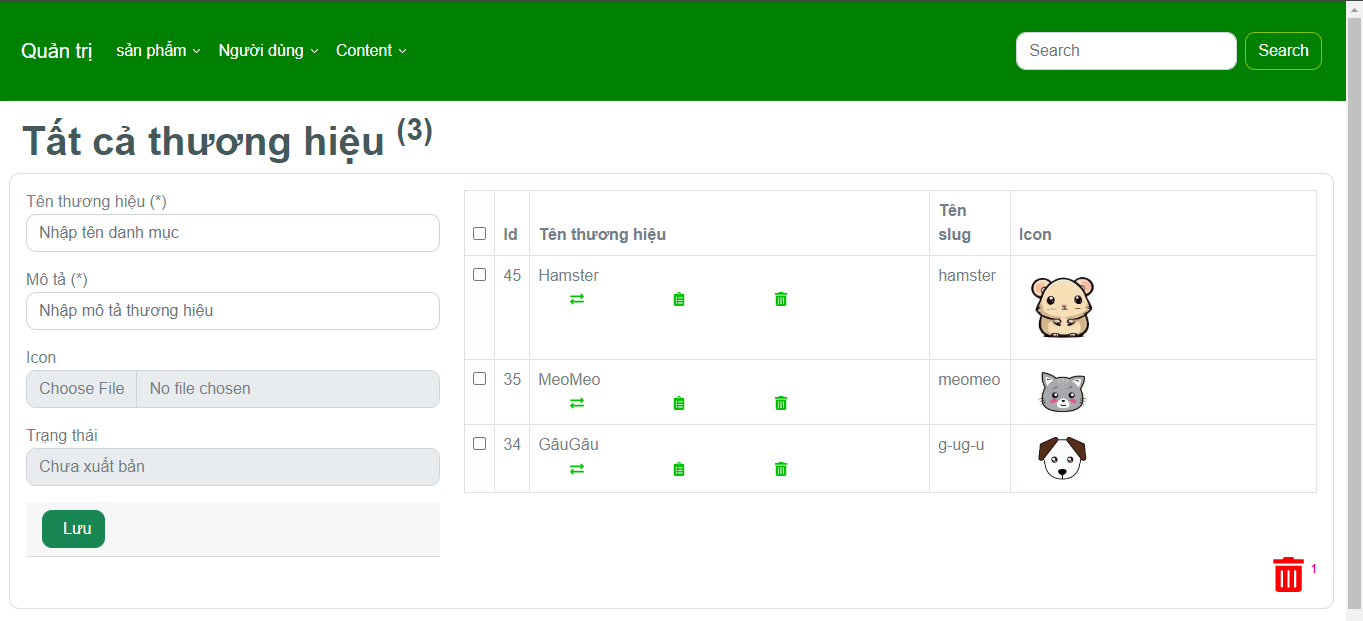
Trên màn hình danh sách danh mục (hình 3.11), khi người dùng cần xem lại danh sách danh mục đã xoá trước đó, người dùng có thể nhấn nút **“Thùng rác”**, màn hình sẽ chuyển sang trang thùng rác danh mục như hình 3.14.



*Hình 3. 14 Danh sách thùng rác*

**3.1.5 Trang quản lý thương hiệu**

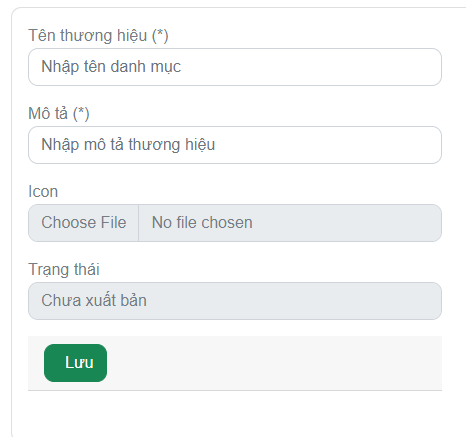
Trang quản thương hiệu sản phẩm, danh sách thương hiệu hiển thị theo từng dòng nhằm tối ưu hóa màn hình hiển thị như hình 3.15.



*Hình 3. 15 Quản lý thương hiệu sản phẩm*

* **Trang thêm mới thương hiệu sản phẩm**

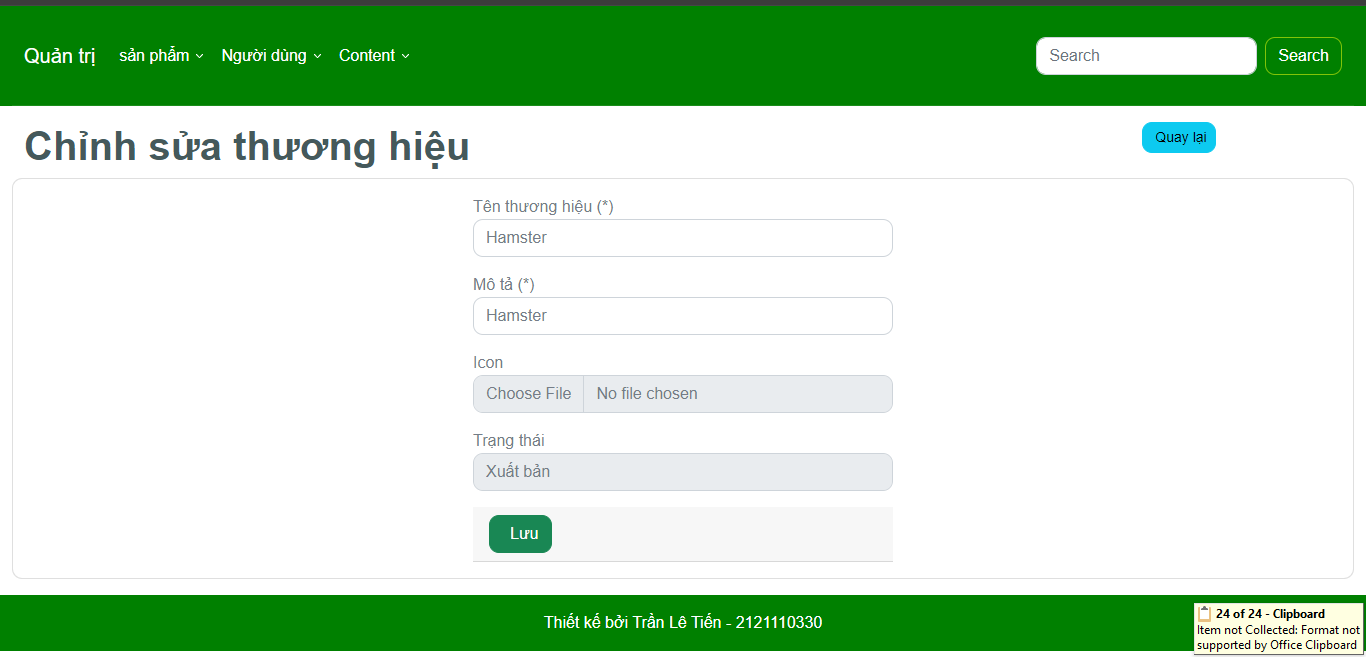
Trên màn hình danh sách thương hiêu (hình 3.15), khi người dùng cần thêm mới, nhấn nút “**Thêm thương hiệu** ” màn hình sẽ xuất hiện bảng thêm mới thương hiệu với các trường thông tin cần nhập như hình 3.16.



*Hình 3. 17 Thêm mới thương hiệu sản phẩm*

* **Trang chỉnh sửa thương hiệu**

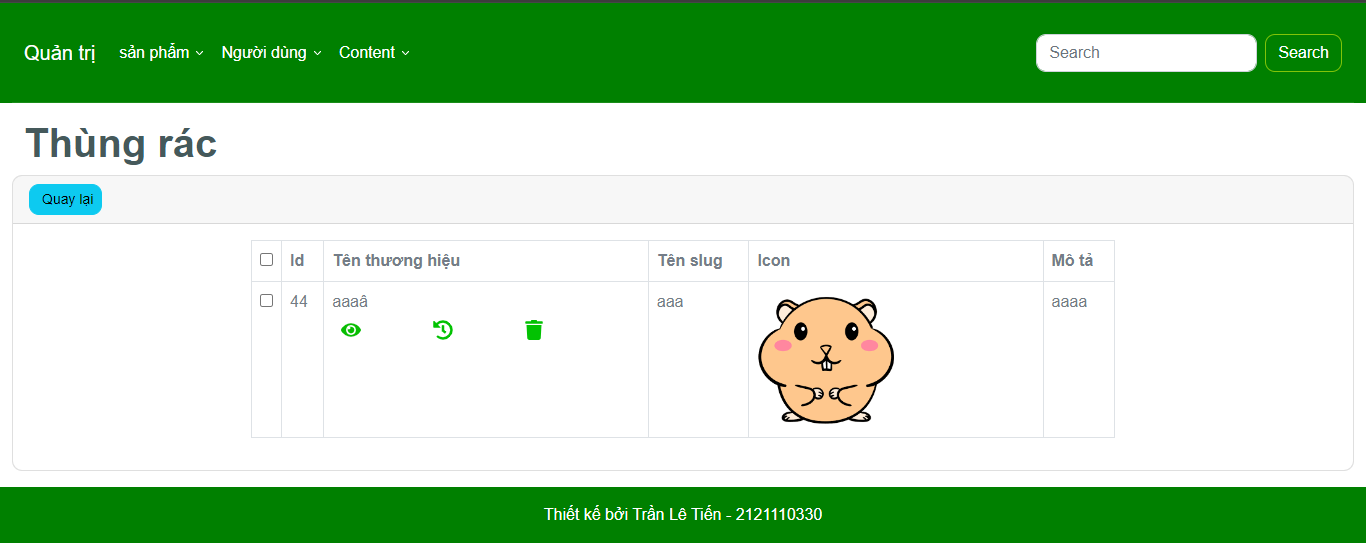
Trên màn hình danh sách thương hiệu (hình 3.15). Khi người dùng cần chỉnh sửa thông tin của thương hiệu, ở cột chức năng, người dùng có thể nhấn vào nút **“Chỉnh sửa”** để chọn chức năng chỉnh sửa, từ đó sẽ chuyển sang trang chỉnh sửa thương hiệu như hình 3.18.



*Hình 3. 18 Chỉnh sửa thương hiệu*

* **Thùng rác thương hiệu**

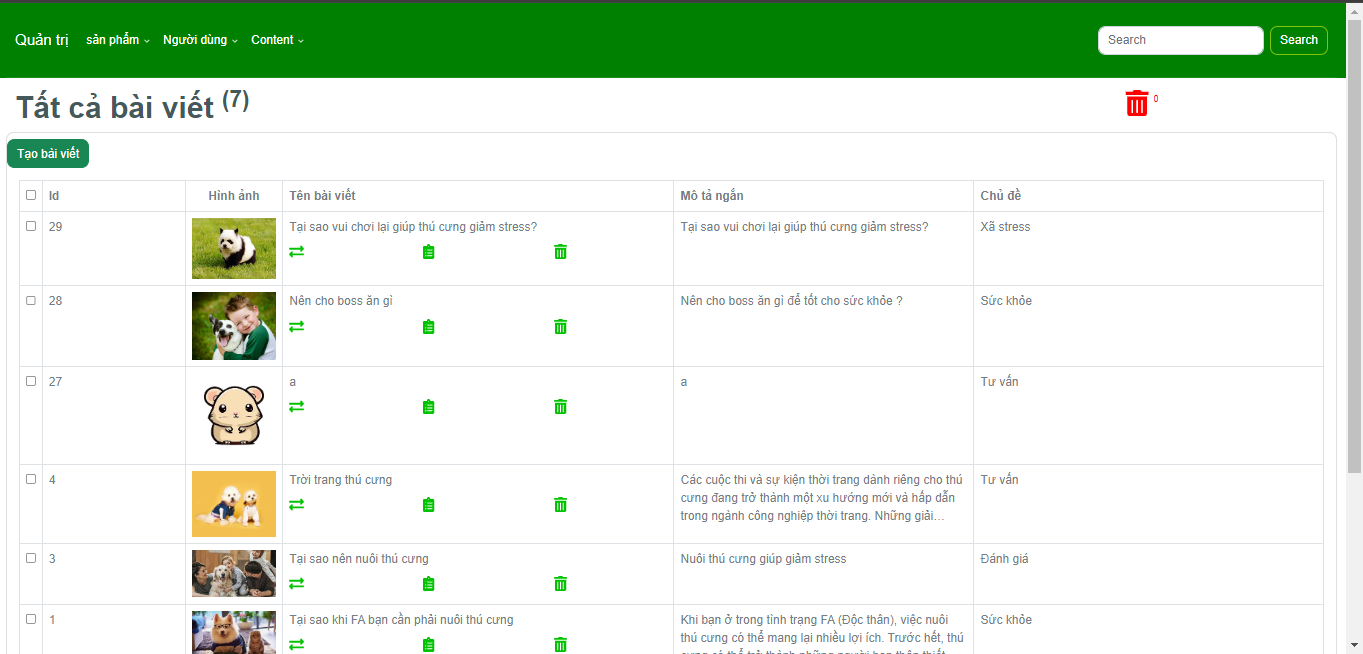
Trên màn hình danh sách thương hiệu (hình 3.15), khi người dùng cần xem lại danh sách thương hiệu đã xoá trước đó, người dùng có thể nhấn nút **“Thùng rác”**, màn hình sẽ chuyển sang trang thùng rác thương hiệu như hình 3.19.



*Hình 3. 19 Danh sách thùng rác*

**3.1.6 Trang quản lý bài viết**

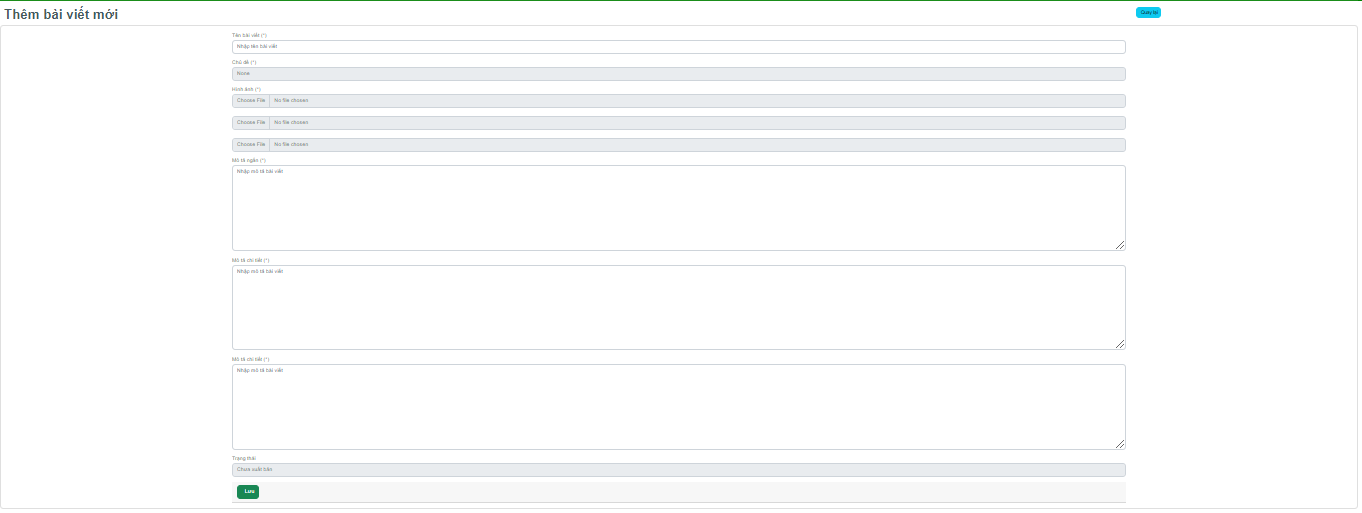
Trang quản lý bài viết, danh sách bài viết hiển thị theo từng dòng nhằm tối ưu hóa màn hình hiển thị như hình 3.20.



*Hình 3. 20 Quản lý bài viết*

* **Trang thêm mới bài viết**

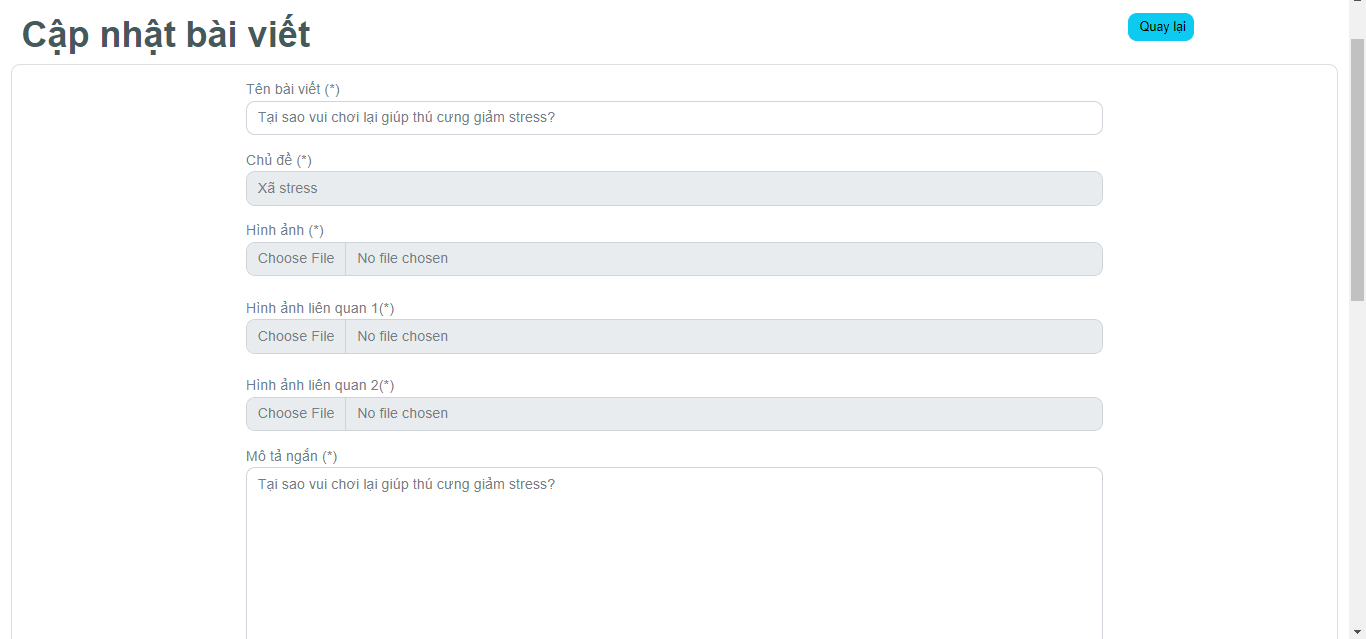
Trên màn hình danh sách bài viết (hình 3.20), khi người dùng cần thêm mới, nhấn nút “**Thêm bài viết** ” màn hình sẽ xuất hiện bảng thêm mới bài viết với các trường thông tin cần nhập như hình 3.22, sau đó nhấn nút thêm để thêm mới danh mục sau khi thêm mới màn hình sẽ được trả về trang quản lý bài viết như hình 3.20.



*Hình 3. 22 Thêm mới bài viết*

* **Trang chỉnh sửa bài viết**

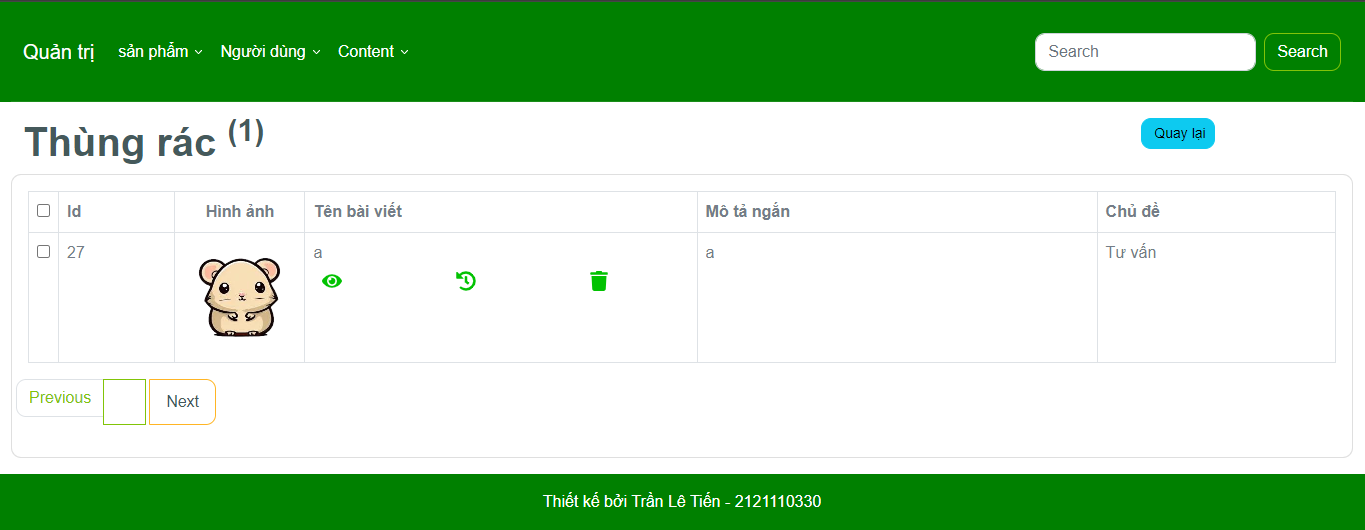
Trên màn hình danh sách bài viết (hình 3.20). Khi người dùng cần chỉnh sửa thông tin của bài viết, ở cột chức năng, người dùng có thể nhấn vào nút **“Chỉnh sửa”** để chọn chức năng chỉnh sửa, từ đó sẽ chuyển sang trang chỉnh sửa bài viết như hình 3.23.



*Hình 3. 23 Chỉnh sửa bài viết*

* **Thùng rác bài viết**

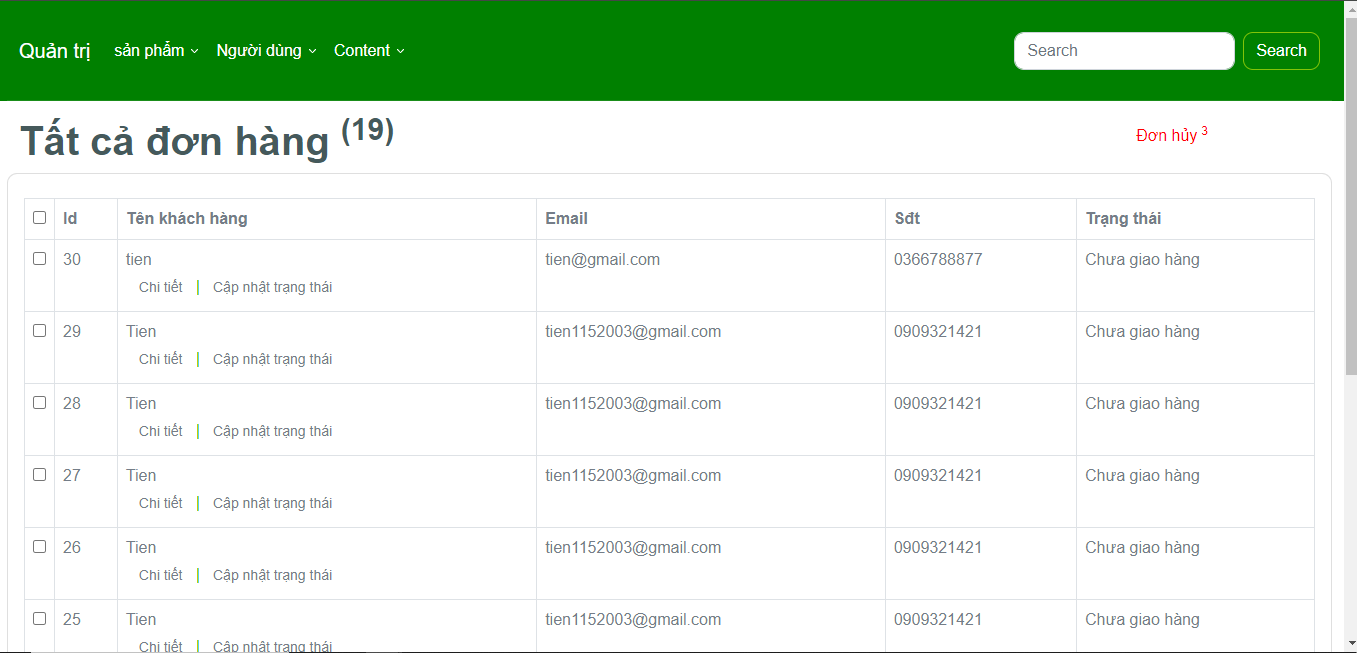
Trên màn hình danh sách bài viết (hình 3.20), khi người dùng cần xem lại danh sách bài viết đã xoá trước đó, người dùng có thể nhấn nút **“Thùng rác”**, màn hình sẽ chuyển sang trang thùng rác bài viết như hình 3.24.



*Hình 3. 24 Danh sách thùng rác*

**3.1.7 Trang quản lý đơn đặt hàng của khách hàng**

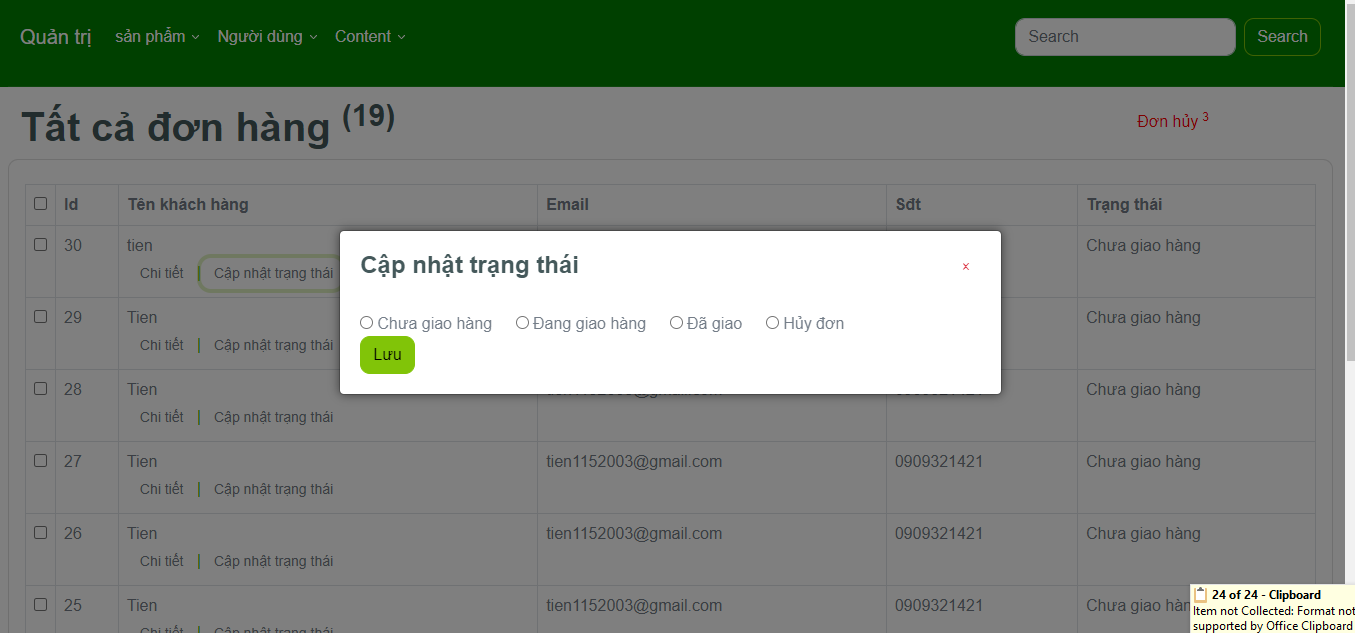
Trang này hiển thị ra danh sách các khách hàng đã đặt hàng online tại shop, từ đó shop sẽ chuẩn bị đơn hàng và gửi đến cho khách hàng.



*Hình 3. 25 Danh sách đơn hàng*

* **Cập nhật trạng thái đơn hàng**

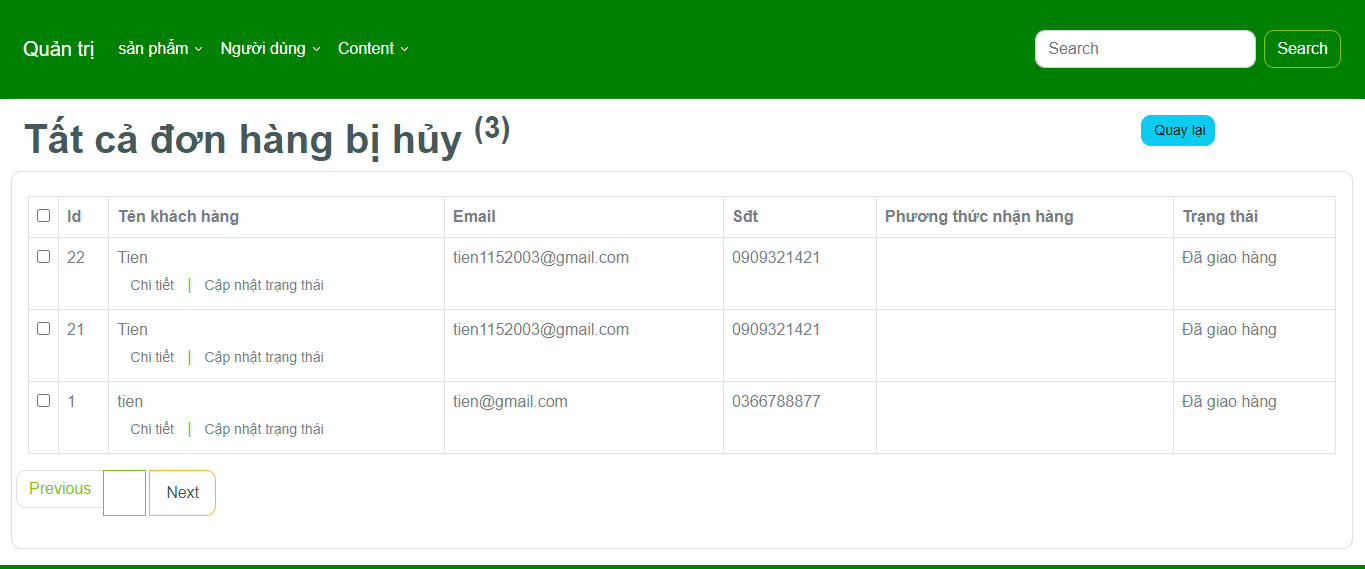
Trên màn hình danh sách đơn hàng (hình 3.25), khi người dùng cần thay đổi trạng thái đơn hàng , nhấn chọn vào nút **“Cập nhật trạng thái”** câu lệnh sẽ được thực thi và màn hình thay đổi trạng thái sẽ được xuất hiện ra như hình 3.27.



*Hình 3. 27 Cập nhật trạng thái đơn hàng*

* **Đơn hủy**

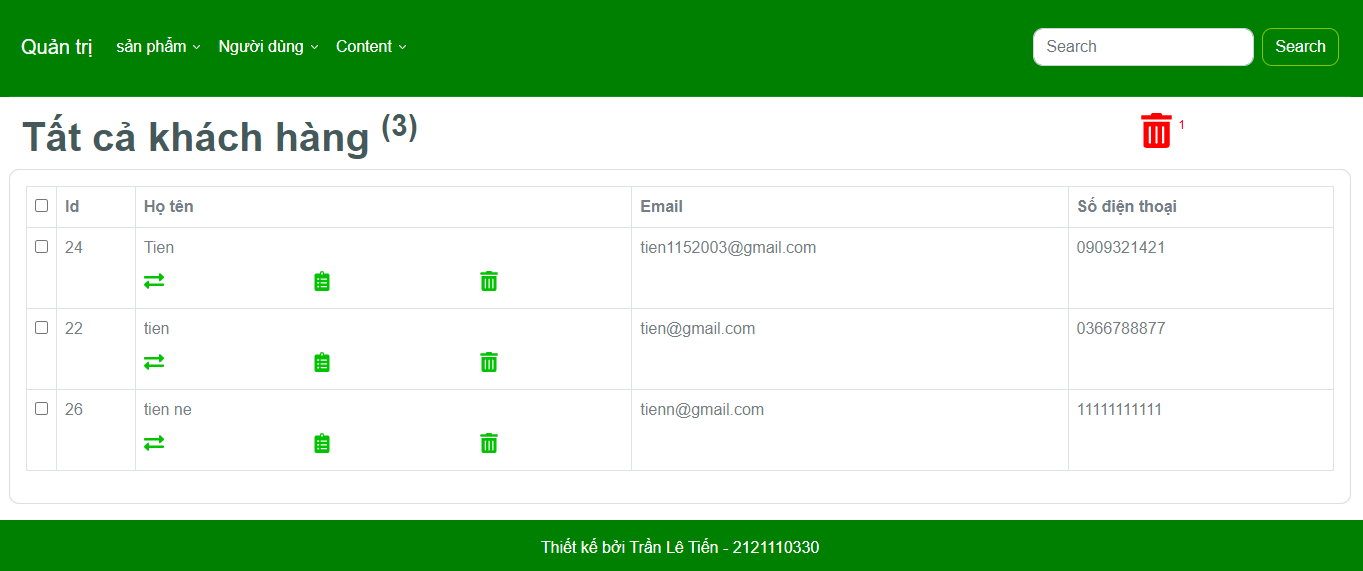
Trên màn hình danh sách bài viết (hình 3.27), khi người dùng cần xem lại danh sách đơn hàng đã hủy trước đó, người dùng có thể nhấn nút **“Đơn hủy”**, màn hình sẽ chuyển sang trang thùng rác bài viết như hình 3.28.



**3.1.8 Trang quản lý tài khoản khách hàng**

Trang này hiển thị thông tin tài khoản khách hàng thống theo từng dòng nhằm tối ưu hóa màn hình hiển thị và người dùng có thể khóa tài khoản khách hàng như hình 3.29.

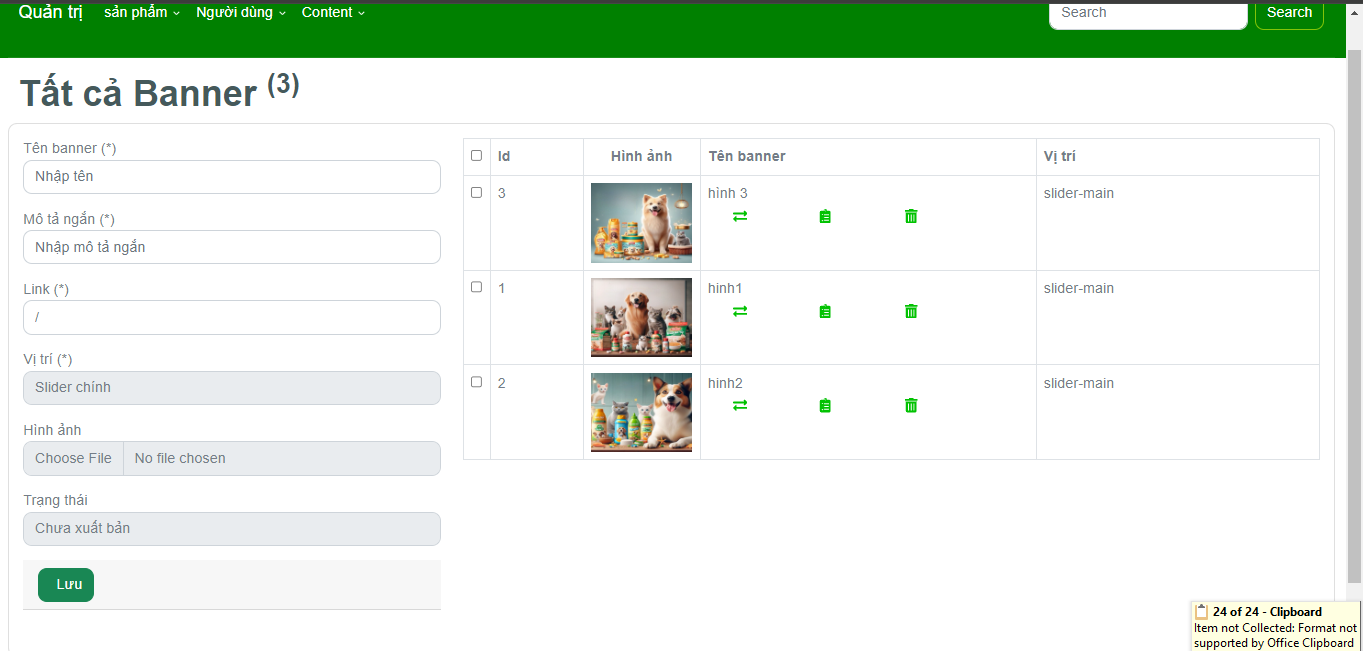
* **Danh sách quản lý tài khoản**



*Hình 3. 29 Danh sách tài khoản*

**3.1.9 Trang quản lý banner**

Trang quản lý banner, danh sách banner hiển thị theo từng dòng nhằm tối ưu hóa màn hình hiển thị như hình 3.30



*Hình 3. 30 Quản lý banner*

* **Trang thêm mới banner**

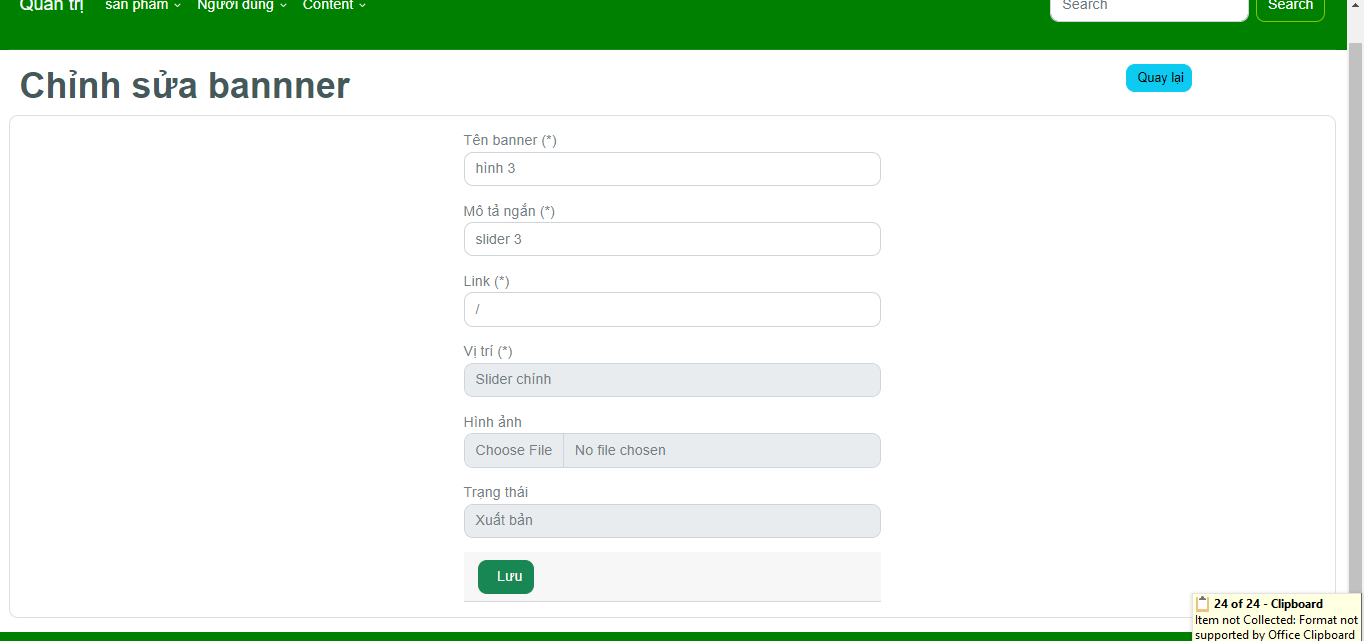
Trên màn hình danh sách banner (hình 3.30), khi người dùng cần thêm mới, nhấn nút “**Thêm banner** ” màn hình sẽ chuyển sang trang thêm mới banner với các trường thông tin cần nhập như hình 3.31, sau đó nhấn nút thêm để thêm mới banner sau khi thêm mới màn hình sẽ được trả về trang quản lý banner như hình 3.30.



*Hình 3. 31 Thêm mới banner*

* **Trang chỉnh sửa banner**

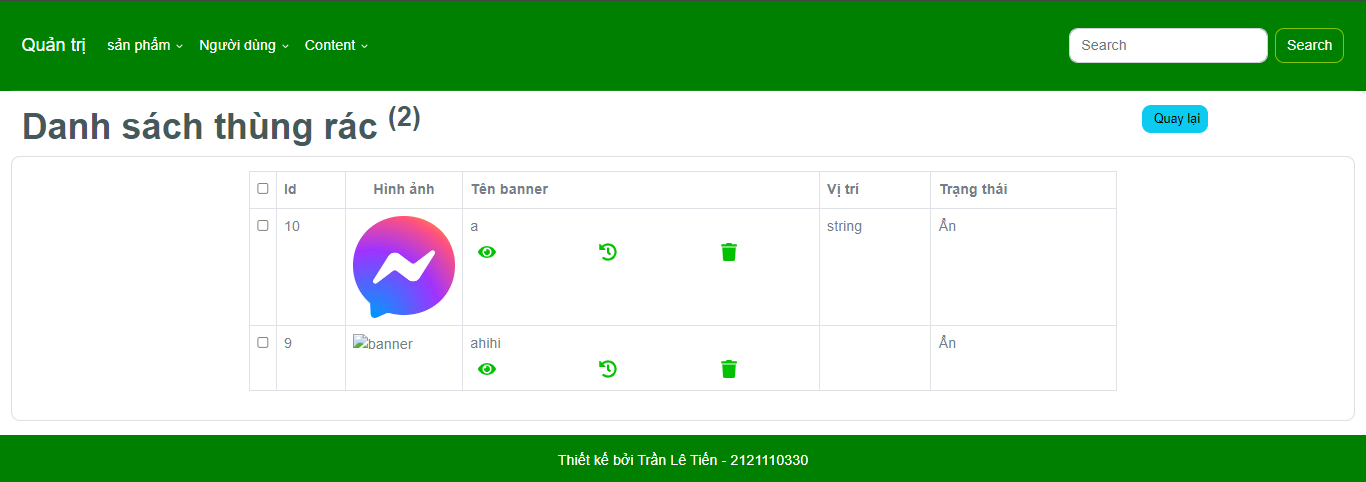
Trên màn hình danh sách banner (hình 3.30). Khi người dùng cần chỉnh sửa thông tin của banner, ở cột chức năng, người dùng có thể nhấn vào nút **“Chỉnh sửa”** để chọn chức năng chỉnh sửa, từ đó sẽ chuyển sang trang chỉnh sửa banner như hình 3.32.



*Hình 3. 32 Chỉnh sửa banner*

* **Thùng rác banner**

Trên màn hình danh sách banner (hình 3.30), khi người dùng cần xem lại danh sách banner đã xoá trước đó, người dùng có thể nhấn nút **“Thùng rác”**, màn hình sẽ chuyển sang trang thùng rác banner như hình 3.33.



*Hình 3. 33 Danh sách thùng rác*

## **3.2 Giao diện người dùng**

**3.2.1 Trang chủ**

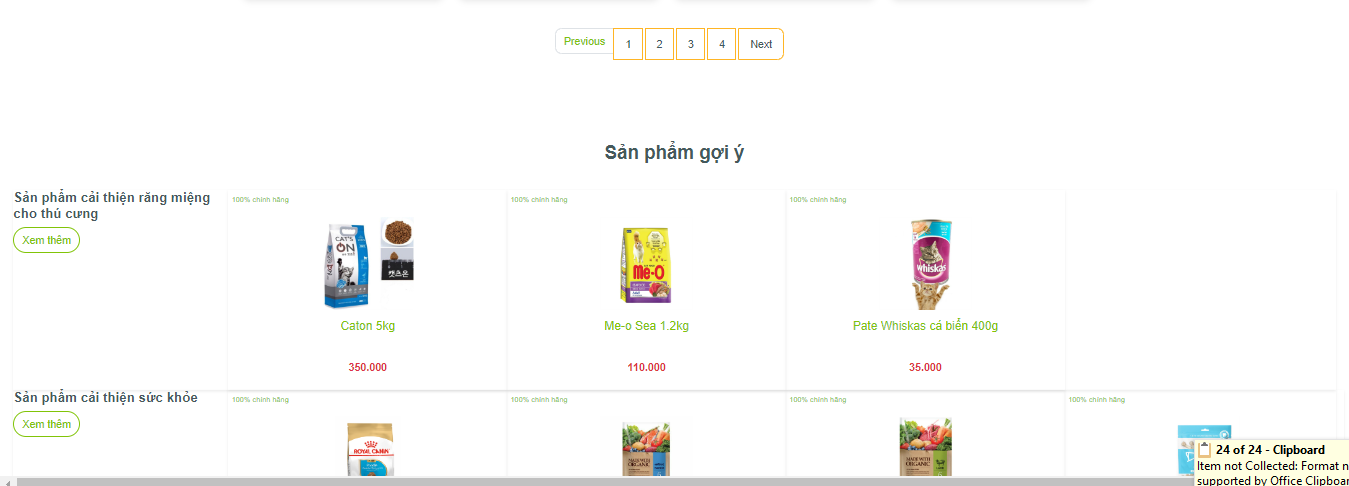
Phần trang chủ của giao diện khách hàng được thiết kế trực quan, các thông tin chính về sản phẩm như tên, hình ảnh, giá tiền, giá khuyến mãi (nếu có) được hiển thị giúp người đọc nắm bắt được thông tin. Trên trang chủ được bố trí theo thứ tự cụ thể header -> menu -> sản phẩm giảm gia -> sản phẩm mới -> sản phẩm bán chạy ->bài viết ->footer



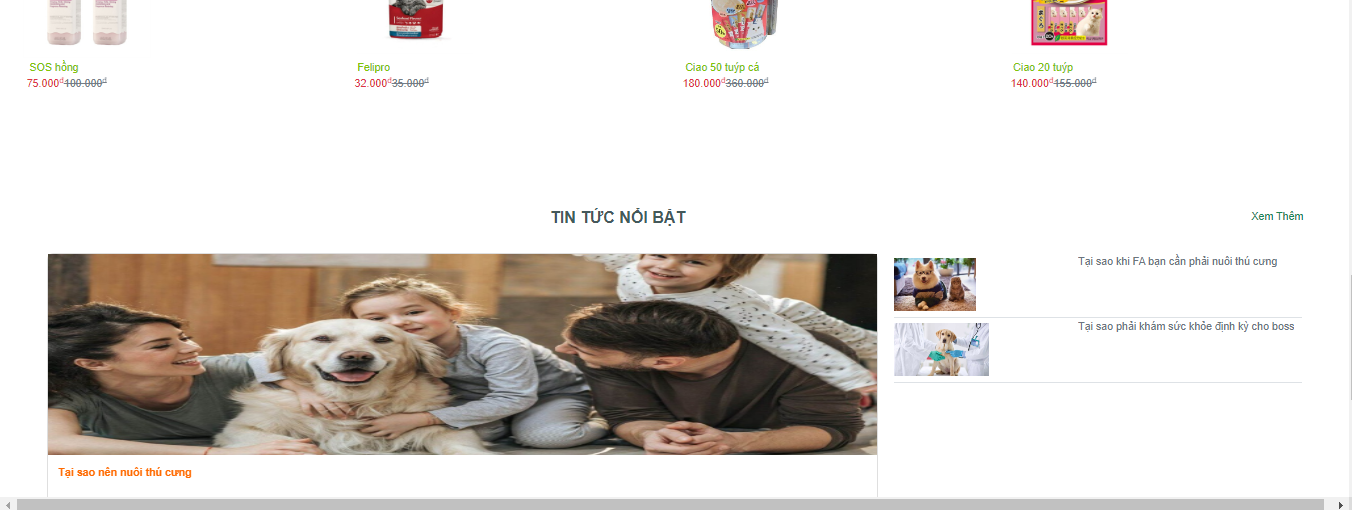
*Hình 3. 34 Giao diện trang chủ*



*Hình 3. 35 Giao diện trang chủ*

**

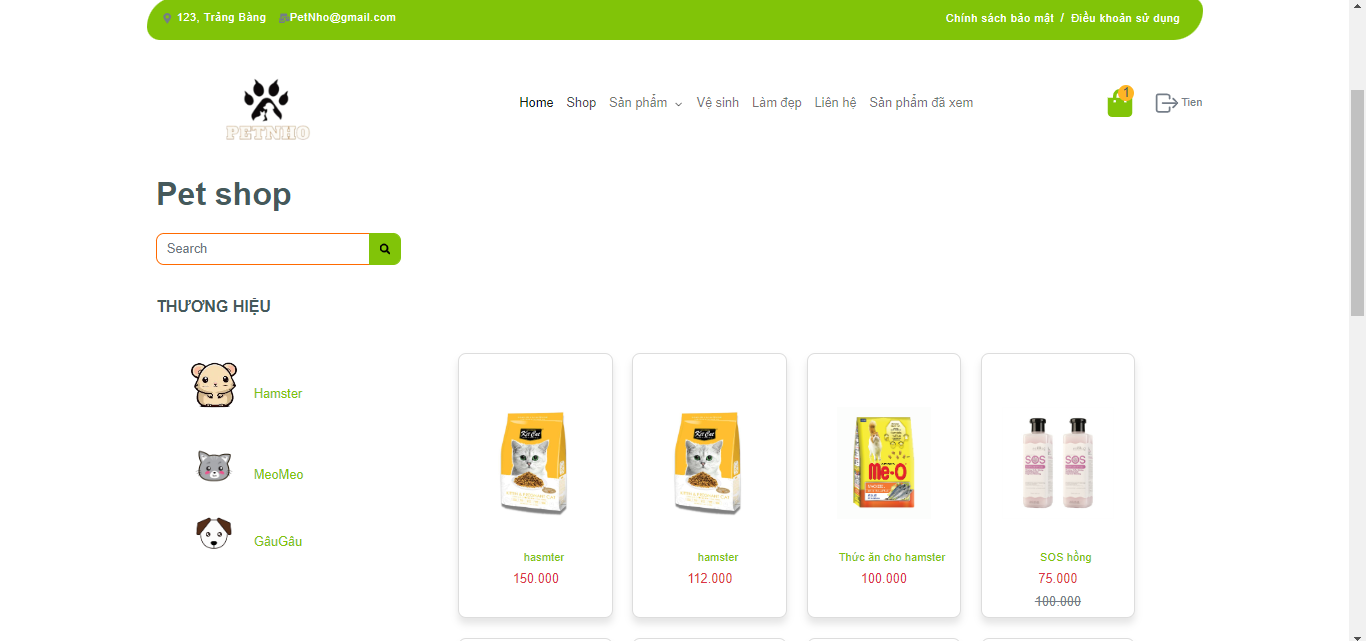
*Hình 3. 35 Giao diện trang chủ*

**

*Hình 3. 35 Giao diện trang chủ*

**3.2.2 Trang tất cả sản phẩm**

Trang này sẽ hiển ra tất cả sản phẩm mà shop hiện tại shop có, người dùng có thể tùy ý xem sản phẩm từ đó người dùng có thể lựa chọn sản phẩm mà cảm thấy vừa ý, phù hợp với túi tiền của mình.

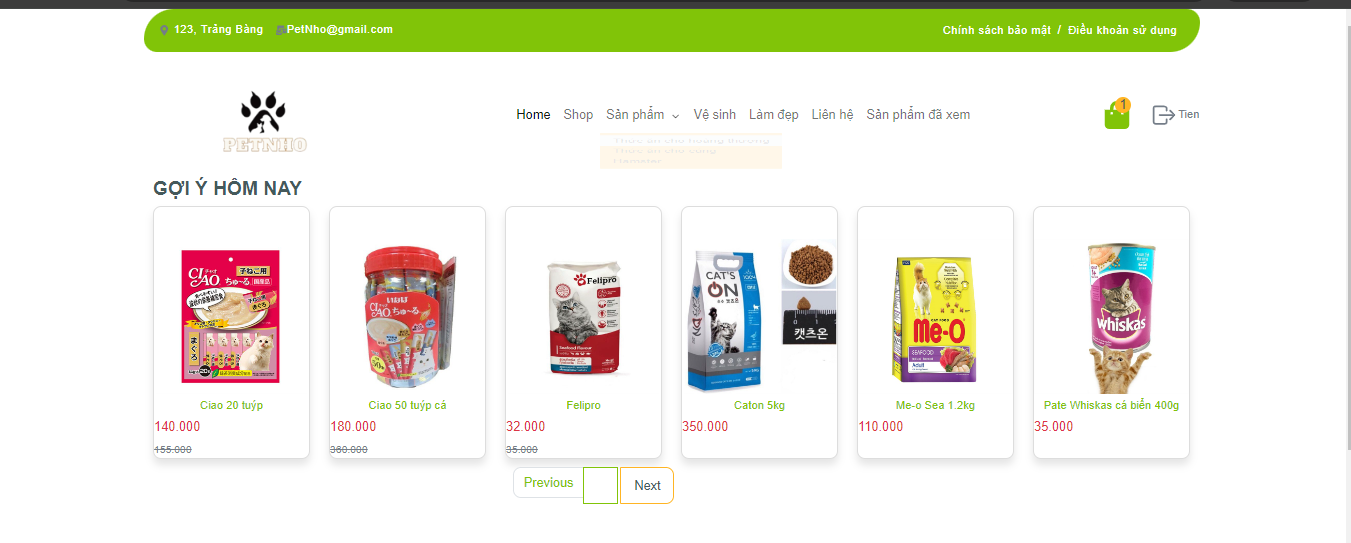


*Hình 3. 36 Tất cả sản phẩm*

**3.2.3 Trang sản phẩm theo danh mục**

Trang này hiển thị ra sản phẩm theo từng loại nhằm đáp ứng nhu cầu của người dùng một cách dễ dàng hơn. Người dùng có thể chọn sản phẩm mình thích mà không tốn quá nhiều thời gian của khách hàng.

* **Sản phẩm theo “Thức ăn cho cún”**

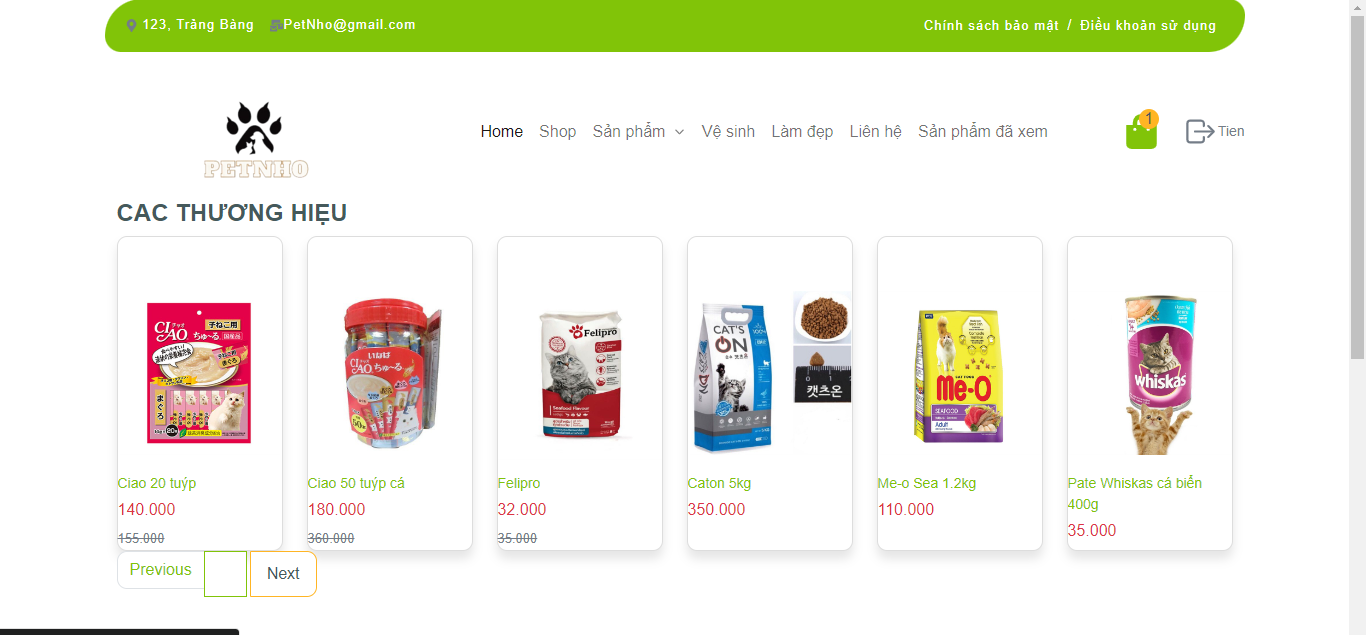


*Hình 3. 37 Giao diện sản phẩm theo thức ăn cho cún*

**3.2.4 Trang sản phẩm theo thương hiệu**

Trang này hiển thị ra sản phẩm theo từng thương hiệu nhằm đáp ứng nhu cầu của người dùng một cách dễ dàng hơn. Người dùng có thể chọn sản phẩm mình thích mà không tốn quá nhiều thời gian của khách hàng.

* **Sản phẩm theo MeoMeo**

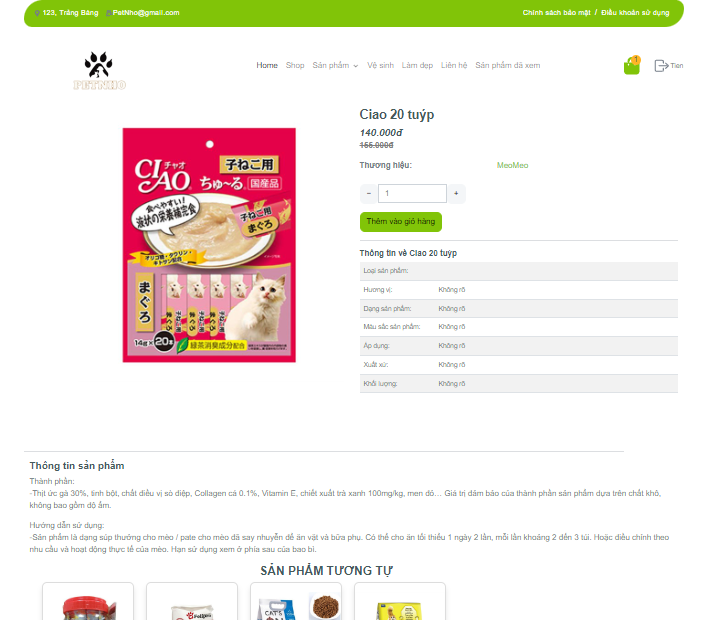


*Hình 3. 39 Giao diện sản phẩm theo robotime*

**3.2.5 Trang chi tiết sản phẩm**

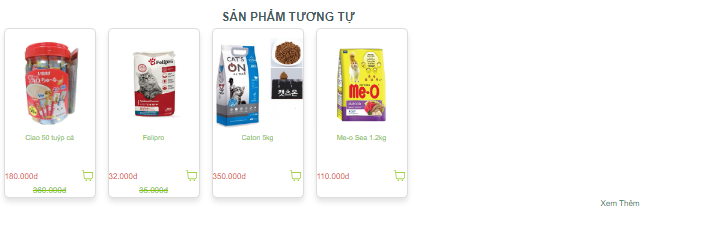
Trang chi tiết sản phẩm có mục đích hiển thị các thông tin về sản phẩm người dùng quan tâm như: thành phần, xuất xứ, áp dụng cho độ tuổi …v.v, sản phẩm liên quan

* **Chi tiết sản phẩm**



*Hình 3. 40 Chi tiết sản phẩm*

* **Sản phẩm liên quan**

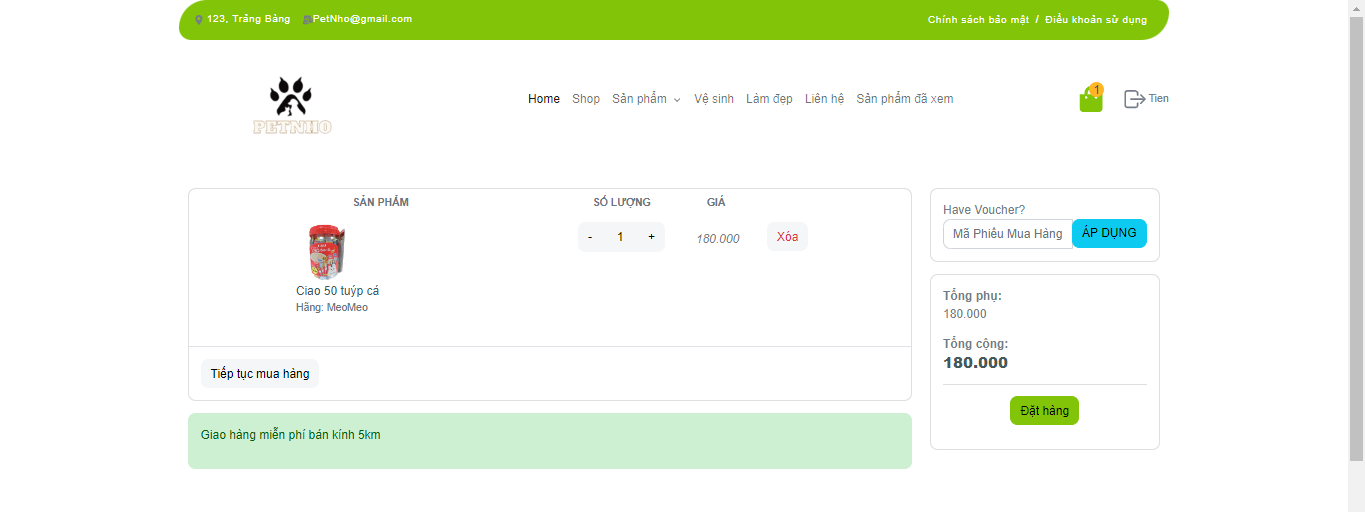


*Hình 3. 41 Mô tả sản phẩm*

**3.2.5 Trang giỏ hàng**

Trên mỗi sản phẩm đều có nút **[Thêm vào giỏ hàng]** để đưa sản phẩm đó vào trong giỏ hàng của người dùng. Sau khi đã chọn xong người dùng bấm vào biểu tượng giỏ hàng để xem sản phẩm mình đã chọn. Khách hàng có thể chỉnh sửa số lượng, xóa bỏ sản phẩm trước khi gửi thông tin đặt hàng.

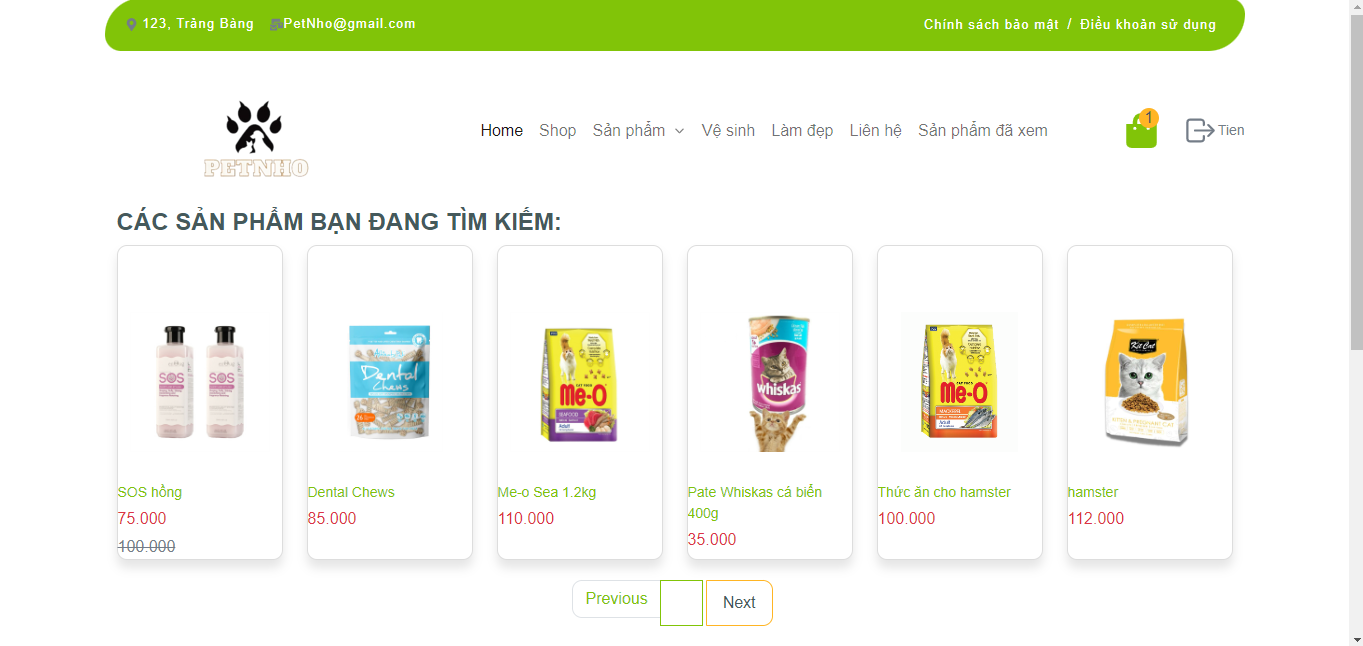
* **Giao diện giỏ hàng**



*Hình 3. 42 Giao diện giỏ hàng*

**3.2.6 Trang tìm kiếm**

Người dùng gõ tên sản phẩm cần tìm vào ô tìm kiếm trên đầu trang, hệ thống sẽ đưa ra gợi ý cho người dùng. Sau khi tìm kiếm màn hình hiển thị các sản phẩm tìm thấy.

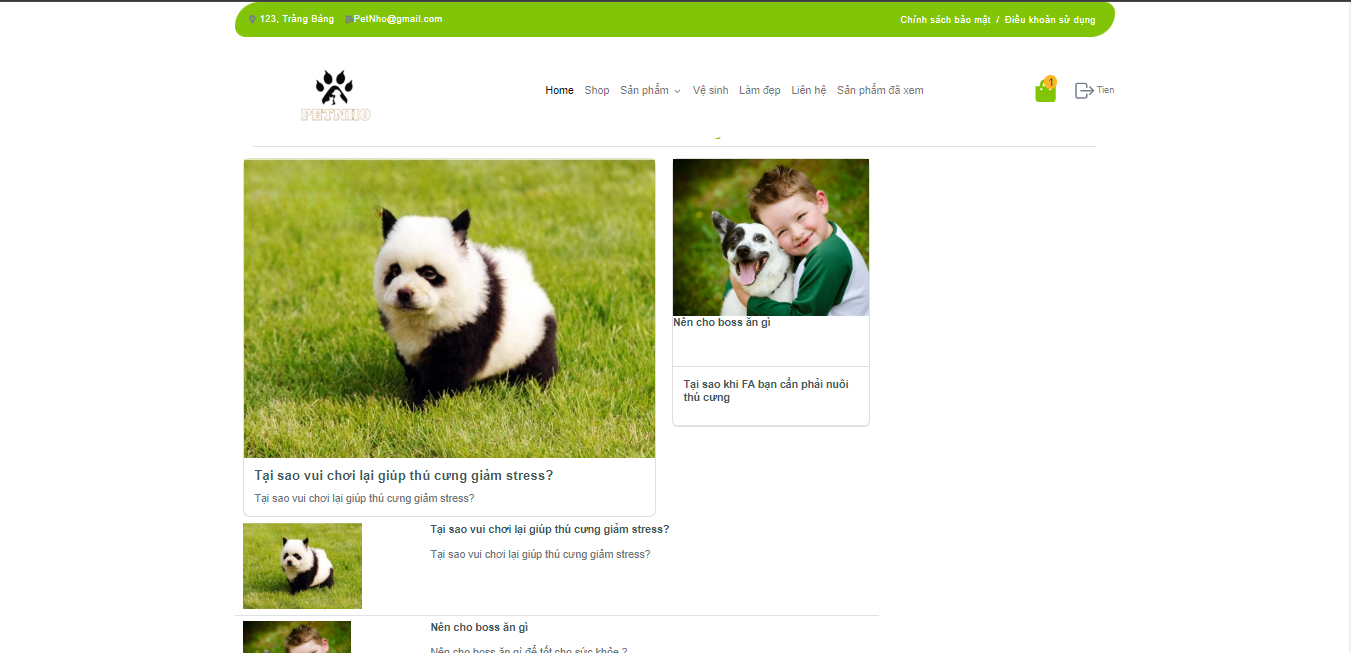


*Hình 3. 43 Giao diện tìm kiếm*

**3.2.7 Trang bài viết**

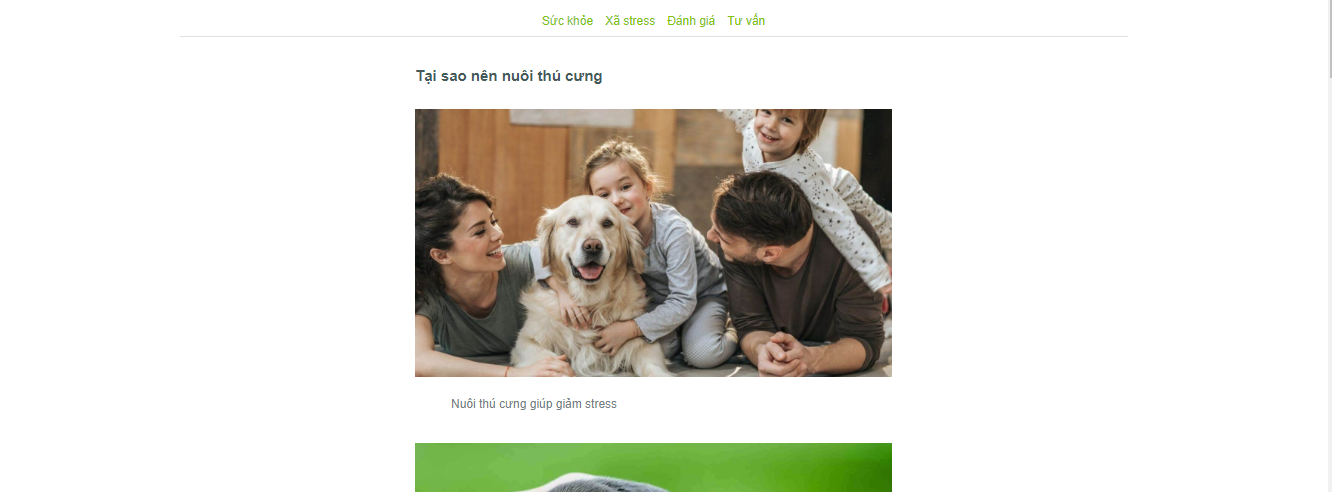
Người dùng tra cứu tin tức mới nhất trên thị trường mua bán hàng hoá may mặc trong và ngoài nước ở phần Tin Tức

* **Giao diện tất cả tin tức**



*Hình 3. 44 Giao diện trang bài viết*

* **Giao diện chi tiết tin tức**



*Hình 3. 45 Giao diện chi tiết bài viết*

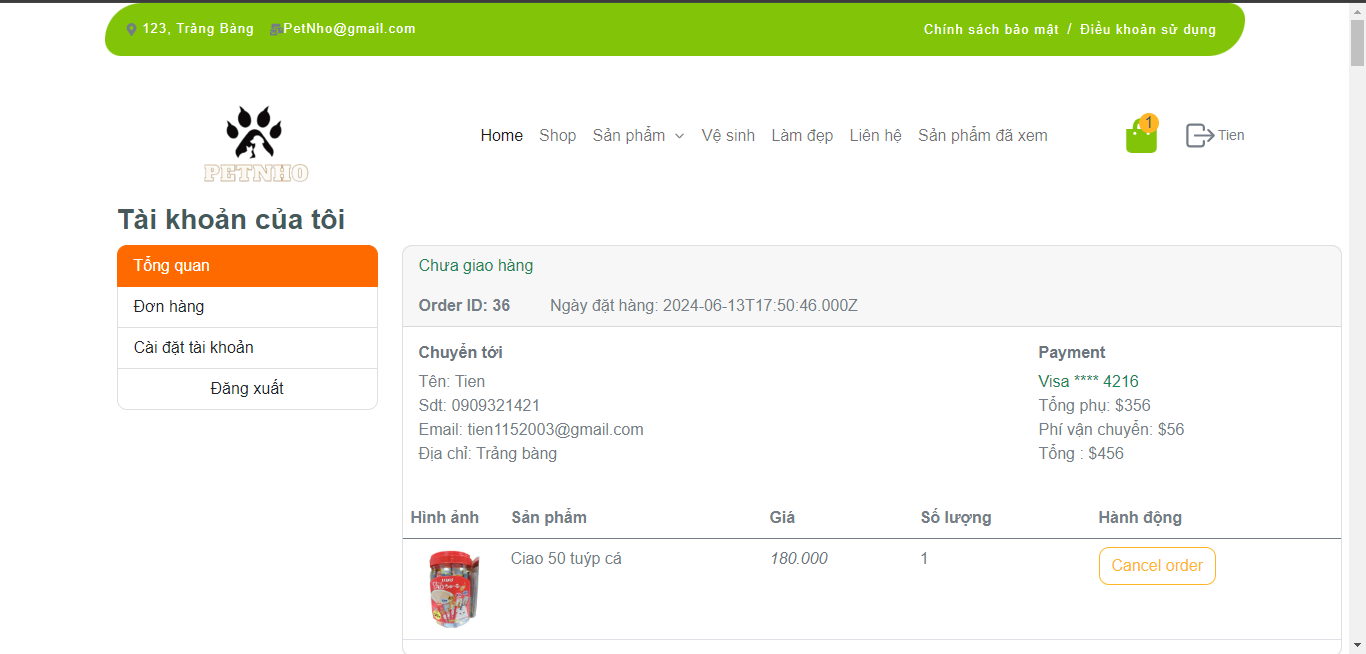
**3.2.9 Trang thanh toán**

Sau khi đã hoàn tất chỉnh sửa giỏ hàng, người dùng sẽ đặt hàng. Nếu người dùng chưa đăng nhập hệ thống sẽ chuyển sang trang đăng nhập. Sau khi đăng nhập xong người dùng mới có thể đặt hàng. Khi đặt hàng, các thông tin cá nhân của người dùng sẽ được tự động được điền vào các thông tin xác nhận đặt hàng. Có thể sửa nếu cần thay đổi và người dùng có thể chọn địa chỉ đã được thêm trong tài khoản.

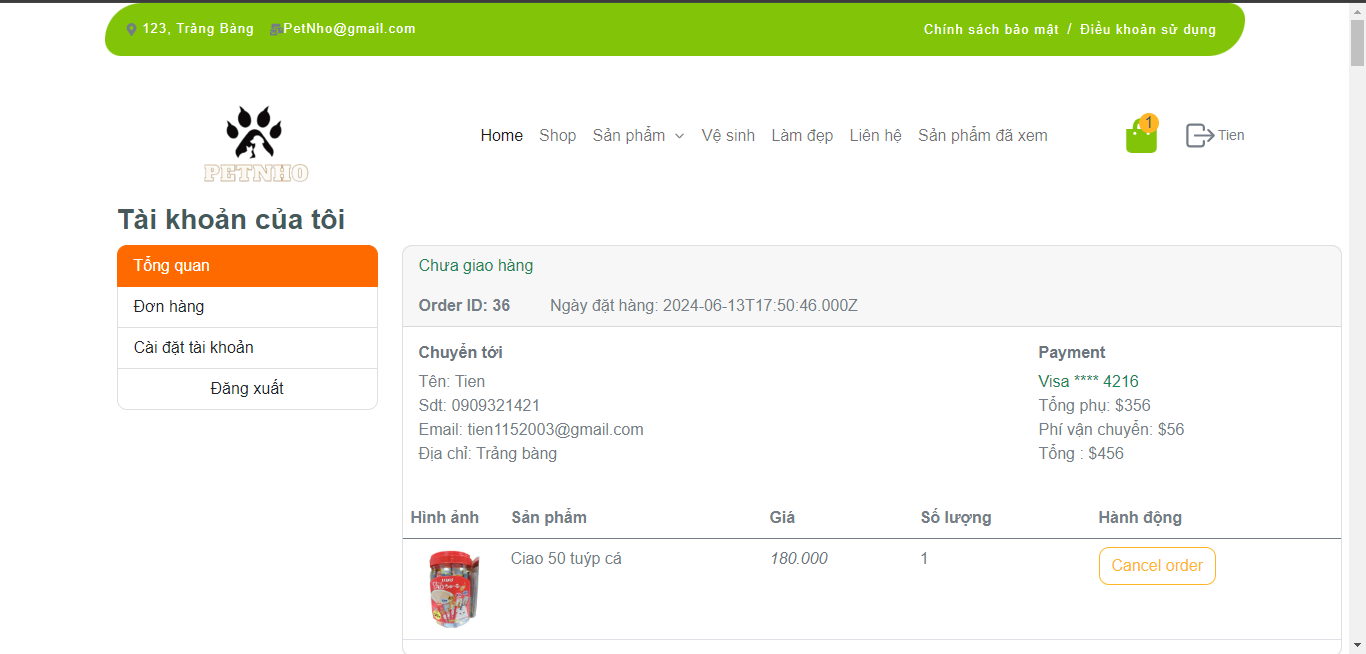
*Hình 3. 47 Trang thanh toán*

**3.2.10 Trang lịch sử đơn hàng của khách hàng**

Khách hàng có thể tra cứu lịch sử mua hàng của bản thân trong phần Lịch sử đơn hàng của website, ở đây sẽ lưu trữ chi tiết tất cả thông tin đơn hàng đầy đủ nhất cho khách hàng kiểm tra.



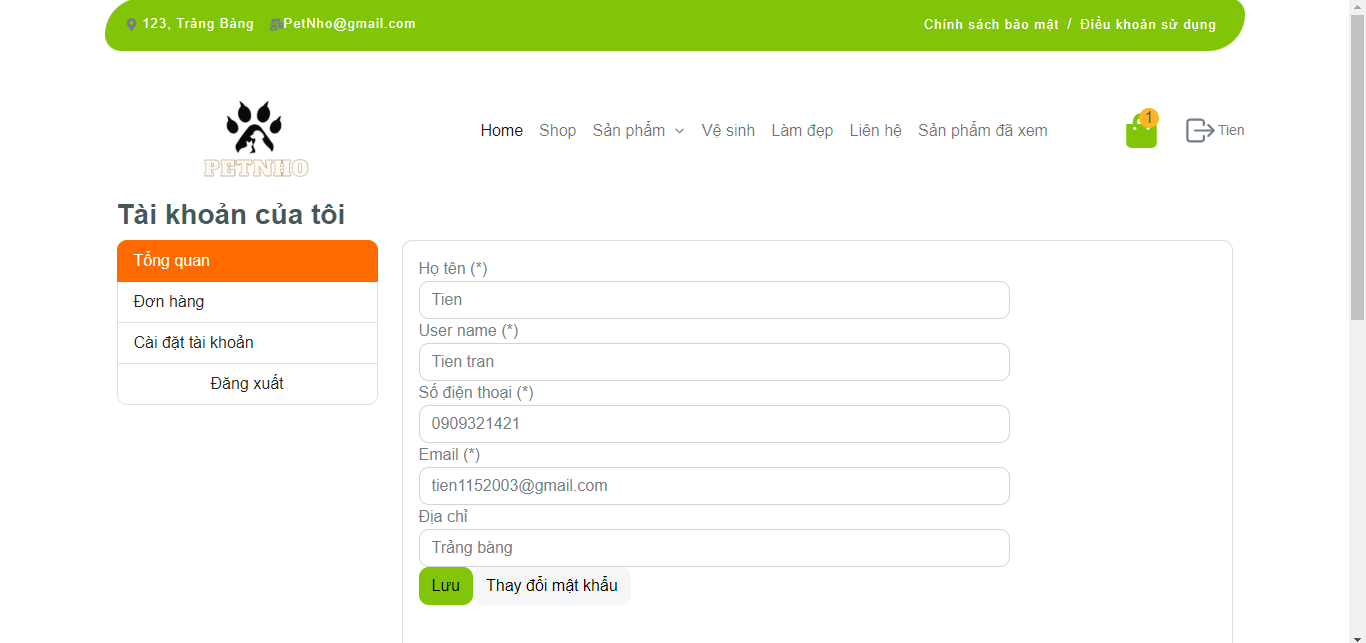
*Hình 3. 48 Trang lịch sử đơn hàng*



*Hình 3. 49 Chi tiết lịch sử đơn hàng*

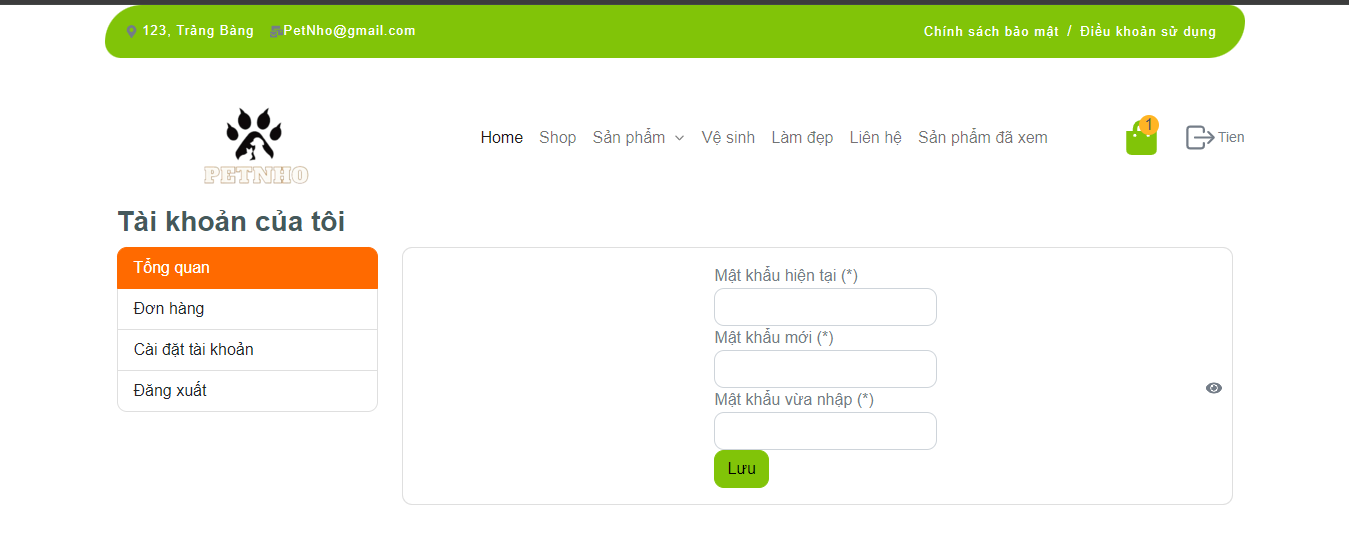
### **3.2.11 Trang thay đổi thông tin khách hàng**

Tại đây người dùng có thể thay đổi một số thông tin cơ bản để phù hợp hơn và dễ dàng sử dụng trong việc mua và kiểm tra hàng hoá.



*Hình 3. 50 Trang thay đổi thông tin khách hàng*

**3.2.12 Trang yêu cầu đổi mật khẩu**



*Hình 3. 51 Trang yêu cầu đổi mật khẩu*

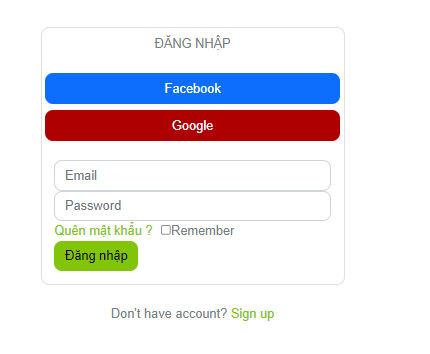
Khi người dùng yêu cầu đổi mật khẩu cần nhập mật khẩu cũ đã đăng ký và hệ thống sẽ kiểm tra nếu mật đúng sẽ có thể đổi mật khẩu.

**3.2.13 Trang đăng kí**



*Hình 3. 52 Giao diện trang đăng kí tài khoản*

**3.2.14 Trang đăng nhập**



*Hình 3. 54 Giao diện trang đăng nhập tài khoản*

# KẾT LUẬN

* **Kết luận**

- Trong bối cảnh phát triển ứng dụng web ngày nay, việc lựa chọn công nghệ phù hợp có ảnh hưởng rất lớn đến hiệu suất, tính bảo mật và sự linh hoạt của sản phẩm cuối cùng. Luận án này đã tập trung vào việc nghiên cứu và áp dụng Fastify và React.js, hai công nghệ tiên tiến, để xây dựng một ứng dụng web hiệu suất cao và tương tác.

- Qua quá trình nghiên cứu và thực nghiệm, nhận thấy rằng Fastify là một framework web Node.js mạnh mẽ, với các tính năng như hiệu suất cao, hỗ trợ validation dựa trên schema, plugin system linh hoạt và hỗ trợ TypeScript. Fastify đã giúp xây dựng backend của ứng dụng một cách dễ dàng và hiệu quả, đồng thời đảm bảo tính ổn định và an toàn của hệ thống.

- Một trong những điểm nổi bật của Fastify là hệ thống plugin mạnh mẽ, cho phép bạn mở rộng chức năng của ứng dụng một cách dễ dàng. Điều này tạo ra tính linh hoạt cao và giúp giảm bớt sự phức tạp trong quá trình phát triển.

- Không chỉ tập trung vào hiệu suất, Fastify còn chú trọng đến việc cung cấp một cách tiếp cận an toàn và ổn định cho việc xử lý lỗi, giúp giảm thiểu các vấn đề liên quan đến lỗi trong ứng dụng.

- Mặt khác, React.js đã cung cấp một cách tiếp cận hiệu quả để xây dựng giao diện người dùng tương tác, với việc sử dụng các thành phần (components) tái sử dụng, quản lý trạng thái (state) và routing linh hoạt. React.js đã giúp tạo ra một giao diện người dùng linh hoạt, dễ bảo trì và có trải nghiệm người dùng tốt.

- Bằng cách kết hợp Fastify và React.js, đã tạo ra một ứng dụng web mạnh mẽ, với backend hiệu suất cao và giao diện người dùng tương tác. Sự kết hợp giữa Fastify và React.js đã mở ra một loạt các cơ hội để phát triển ứng dụng web có thể mở rộng và linh hoạt. Điều này không chỉ tăng cường trải nghiệm người dùng mà còn giúp tăng cường năng suất phát triển và giảm thời gian phát triển ứng dụng.

Trong tương lai, việc tiếp tục sử dụng Fastify và React.js sẽ là một lựa chọn thông minh cho việc phát triển ứng dụng web, giúp cung cấp các sản phẩm chất lượng cao và đáp ứng nhu cầu ngày càng tăng của thị trường.

# TÀI LIỆU THAM KHẢO

* **Tiếng Việt**

[1] Thạc Bình Cường, Phân tích thiết kế hệ thống thông tin, Nhà xuất bản Khoa học Kỹ thuật, 1999.

[2] Đào Kiến Quốc, Phân tích thiết kế hệ thống thông tin học hóa, Đại học Quốc gia Hà Nội, 1999.

[3] Phạm Minh Tuấn, Giáo trình Phân tích và thiết kế hệ thống thông tin quản lý, Nhà xuất bản Hà Nội, 2005.

* **Trang web**

[4] [Get started with Bootstrap · Bootstrap v5.2 (getbootstrap.com)](https://getbootstrap.com/docs/5.2/getting-started/introduction/) (truy cập 5/2024)

[5] [Documentation | Fastify (fastify.dev)](https://nodejs.org/en/docs/) (truy cập 4/2024)

[6] [Documentation | MySQL](https://www.mongodb.com/) (truy cập 5/2024)

[7] [Dashboard - CRM | Sneat - Bootstrap 5 HTML Admin Template - Pro (themeselection.com)](https://demos.themeselection.com/sneat-bootstrap-html-admin-template/html/vertical-menu-template-semi-dark/index.html) (truy cập 5/2024)

[8] [Forgot password & Password reset flow in node.js - DEV Community](https://dev.to/cyberwolve/how-to-implement-password-reset-via-email-in-node-js-132m) (truy cập 5/2024)

[9] [(31) How to send email using Gmail API (uses OAuth2 and nodemailer) - YouTube](https://www.youtube.com/watch?v=-rcRf7yswfM) (truy cập 6/2024)

[10] [Export NoSql(MongoDB) Data to MS-Excel In Node.js. - YouTube](https://www.youtube.com/watch?v=zL_PcpvRk-M) (truy cập 6/2024)