

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A blue logo with a ball and a swirly circle

Description automatically generated**

**MÔN HỌC: LẬP TRÌNH WEB**

**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**XÂY DỰNG WEBSITE BÁN HÀNG**

**UTE SHOP**

**SINH VIÊN THỰC HIỆN:**

**Tô Hữu Đức 22110311**

**Trần Trọng Nghĩa 22110380**

**Nguyễn Duy Đạt 22110307**

**GVHD: TS. Nguyễn Hữu Trung**

**Tp. Hồ Chí Minh, tháng 12 năm 2024**

**NHẬN XÉT CỦA GIẢNG VIÊN**

MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc182980300)

[LỜI MỞ ĐẦU 2](#_Toc182980301)

[PHẦN NỘI DUNG 3](#_Toc182980302)

[CHƯƠNG 1: GIỚI THIỆU 3](#_Toc182980303)

**[1.1 Lý do chọn đề tài 3](#_Toc182980305)**

**[1.2 Mục tiêu của hệ thống 3](#_Toc182980306)**

**[1.3 Phạm vi thực hiện 3](#_Toc182980307)**

**[1.4 Công nghệ sử dụng 4](#_Toc182980305)**

[CHƯƠNG 2: PHÂN TÍCH YÊU CẦU HỆ THỐNG 5](#_Toc182980304)

**[1. Giới thiệu một sô công nghệ chính sử dụng 5](#_Toc182980305)**

**[2. Tầng dữ liệu 14](#_Toc182980306)**

**[3. Tầng trung gian 14](#_Toc182980307)**

[4](#_Toc182980307)**[. Tầng trình bày 15](#_Toc182980307)**

[CHƯƠNG 3: PHÂN TÍCH CHỨC NĂNG 17](#_Toc182980304)

**[1. Mô tả chức năng 17](#_Toc182980305)**

**[2. Sơ đồ chức năng 18](#_Toc182980306)**

[CHƯƠNG 4: PHÂN TÍCH YÊU CẦU DỮ LIỆU 19](#_Toc182980304)

**[1. Mô tả thiết kế cơ sở dữ liệu 19](#_Toc182980305)**

**[2. Mô tả sơ đồ diagram 22](#_Toc182980306)**

[CHƯƠNG 5: KẾT QUẢ ĐẠT ĐƯỢC 19](#_Toc182980304)

**[1. Giao diện của trang khách hàng 23](#_Toc182980305)**

**[2. Giao diện của trang admin 29](#_Toc182980306)**

**[3. Giao diện của trang vendor 35](#_Toc182980305)**

# LỜI CẢM ƠN

Nhóm chúng em xin gửi lời cảm ơn sâu sắc đến thầy Nguyễn Hữu Trung, giảng viên môn Lập trình Web, người đã tận tình giảng dạy, hướng dẫn và luôn đồng hành cùng chúng em trong suốt quá trình học tập và thực hiện bài tiểu luận này.

Qua từng bài giảng của thầy, chúng em không chỉ được tiếp thu những kiến thức nền tảng về lập trình web mà còn học được cách áp dụng linh hoạt vào các tình huống thực tế. Những ví dụ minh họa, những lời giải thích chi tiết của thầy đã giúp chúng em hiểu rõ hơn về các công nghệ và phương pháp xây dựng website hiện đại. Thầy không chỉ là người truyền đạt kiến thức mà còn là người truyền cảm hứng, giúp chúng em có thêm động lực và sự tự tin để chinh phục những thử thách trong quá trình học tập.

Trong suốt thời gian thực hiện bài tiểu luận này, thầy đã luôn sẵn sàng lắng nghe, giải đáp những thắc mắc và đưa ra những ý kiến quý báu để nhóm chúng em có thể hoàn thiện bài làm tốt nhất có thể. Nhờ sự hướng dẫn tận tình và tâm huyết của thầy, chúng em đã học được cách làm việc nhóm hiệu quả, tư duy logic và kỹ năng giải quyết vấn đề, những điều vô cùng cần thiết cho sự phát triển nghề nghiệp trong tương lai.

Chúng em hiểu rằng bài tiểu luận này vẫn còn nhiều hạn chế và thiếu sót. Vì vậy, nhóm chúng em rất mong nhận được sự góp ý, đánh giá từ thầy để có thể cải thiện và hoàn thiện hơn trong những dự án sau này.

Một lần nữa, nhóm chúng em xin gửi lời cảm ơn chân thành đến thầy vì tất cả những gì thầy đã làm cho chúng em trong suốt thời gian qua.

# LỜI MỞ ĐẦU

Xin chào thầy và các bạn,

Trong thời đại công nghệ số phát triển vượt bậc, các nền tảng thương mại điện tử đã và đang trở thành xu hướng tất yếu, mang lại sự tiện lợi, nhanh chóng và hiệu quả cho cả người mua lẫn người bán. Việc xây dựng một website bán hàng không chỉ giúp doanh nghiệp mở rộng thị trường mà còn là cầu nối quan trọng giữa sản phẩm và khách hàng.

Xuất phát từ nhu cầu thực tế này, nhóm chúng em đã chọn đề tài "Xây dựng website bán hàng UTESHOP" để tìm hiểu và áp dụng các kiến thức đã học. Đây là dự án mang tính thực tiễn cao, đòi hỏi sự kết hợp nhuần nhuyễn giữa lý thuyết và thực hành, cũng như khả năng phối hợp các công nghệ hiện đại như Servlet, JSP/JSTL, Bootstrap, JDBC và các hệ quản trị cơ sở dữ liệu SQLServer, MySQL, PostgreSQL. Ngoài ra, việc tích hợp các công cụ như Decorator Sitemesh và JWT giúp dự án đạt được tính bảo mật và hiệu quả cao hơn.

Trong quá trình thực hiện, chúng em đã cố gắng vận dụng những kiến thức đã học để hoàn thiện hệ thống, từ khâu thiết kế giao diện, xây dựng cơ sở dữ liệu, xử lý nghiệp vụ đến kiểm thử và triển khai. Tuy nhiên, do thời gian và năng lực còn hạn chế, chắc chắn hệ thống vẫn còn những điểm chưa hoàn thiện. Chúng em mong nhận được sự góp ý từ thầy cô để có thể cải thiện và phát triển thêm trong tương lai.

# PHẦN NỘI DUNG

## GIỚI THIỆU

**1.1 Lý do chọn đề tài:**

Ngày nay, thương mại điện tử đang phát triển mạnh mẽ, thay đổi cách thức mua sắm và kinh doanh của con người. Các nền tảng trực tuyến không chỉ giúp tiết kiệm thời gian mà còn mở rộng phạm vi tiếp cận khách hàng một cách đáng kể.

Đề tài "Xây dựng website bán hàng UTESHOP" được chọn với mong muốn cung cấp một hệ thống giúp người dùng trải nghiệm việc mua sắm trực tuyến dễ dàng, tiện lợi và hiệu quả. Đồng thời, đây cũng là cơ hội để nhóm vận dụng các kiến thức đã học trong môn lập trình web và tiếp cận với các công nghệ thực tiễn, từ đó nâng cao kỹ năng lập trình cũng như khả năng làm việc nhóm.

**1.2 Mục tiêu của hệ thống:**

Hệ thống UTESHOP hướng tới các mục tiêu chính:

* Tạo môi trường mua sắm trực tuyến: Cung cấp một giao diện thân thiện, dễ sử dụng cho người dùng để tìm kiếm và mua sản phẩm.
* Quản lý hiệu quả: Hỗ trợ quản lý sản phẩm, danh mục, đơn hàng, và tài khoản một cách hiệu quả.
* Bảo mật cao: Tích hợp các giải pháp bảo mật hiện đại như JWT để bảo vệ thông tin người dùng và hệ thống.
* Khả năng mở rộng: Xây dựng hệ thống với kiến trúc linh hoạt, dễ dàng mở rộng và tích hợp các tính năng mới trong tương lai.

**1.3 Phạm vi thực hiện:**

Dự án tập trung xây dựng một hệ thống bán hàng trực tuyến cơ bản với các chức năng chính như:

* Trang chủ hiển thị sản phẩm nổi bật và danh mục.
* Chức năng tìm kiếm, phân loại sản phẩm theo danh mục, yêu thích, bán chạy, sản phẩm mời.
* Thêm sản phẩm vào giỏ hàng hoặc yêu thích, quản lý và thanh toán đơn hàng.
* Đăng ký, đăng nhập và quản lý tài khoản.
* Trang quản trị dành cho người bán hàng có thể quản lý sản phẩm, cửa hàng kinh doanh, khách hàng , đơn hàng, khuyến mãi, đánh giá về sản phẩm của khách hàng và tính toán doanh thu (ngày, tháng, quý và năm).
* Trang quản trị cho admin quản lý sản phẩm, doanh mục, quản lý tài khoản khách hàng và người bán hàng, đơn hàng và tính toán doanh thu (ngày, tháng, quý và năm).

**1.4 Công nghệ sử dụng:**

* Ngôn ngữ lập trình: Java (JDK 17), HTML, CSS, JAVASCRIPT
* Framework thực hiện:

+ Servlet, JSP/JSTL: Xử lý backend và hiển thị dữ liệu.

+ Bootstrap: Thiết kế giao diện web thân thiện và responsive.

+ Decorator Sitemesh: Quản lý và tái sử dụng các thành phần giao diện.

+ JWT (JSON Web Token): Bảo mật phiên đăng nhập và thông tin người dùng.

* Cơ sở dữ liệu: SQL Server để lưu trữ dữ liệu và sử dụng JDBC để tương tác cơ sở dữ liệu.
* Công cụ quản lý môi trường và phát triển:

+ IDE: Eclipse

+ Máy chủ ứng dụng: Apache Tomcat 10.1.28

+ Công cụ quản lý: Github

## CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

**1. Giới thiệu một số công nghệ chính sử dụng:**

**1.1 JSP**

**1.1.1 Định nghĩa:**

JavaServer Pages (JSP) là một công nghệ dựa trên Java dùng để phát triển các ứng dụng web động. JSP cho phép nhúng mã Java vào mã HTML thông qua các thẻ JSP, giúp tạo ra các trang web động một cách dễ dàng.

+ JSP được biên dịch thành các servlet Java bởi máy chủ ứng dụng, sau đó servlet này xử lý các yêu cầu từ client.

+ Nó thường được sử dụng để xây dựng giao diện phía server, kết nối với cơ sở dữ liệu, và tạo các nội dung động.

**1.1.2 Ưu điểm:**

* Tích hợp tốt với Java: Hỗ trợ đầy đủ các API và công nghệ Java.
* Độc lập nền tảng: Chạy trên JVM, hỗ trợ đa hệ điều hành.
* Phân tách giao diện và logic: Dễ bảo trì và phát triển.
* Tái sử dụng mã: Hỗ trợ Custom Tag và JavaBean.
* Hiệu suất cao: Biên dịch thành servlet để tăng tốc độ.

**1.1.3 Nhược điểm:**

* Phức tạp: Viết và bảo trì mã khó hơn các ngôn ngữ khác.
* Yêu cầu máy chủ hỗ trợ JSP: Cần server như Tomcat, GlassFish.
* Khó gỡ lỗi: Mã JSP được chuyển thành servlet.
* Tốn tài nguyên server: JVM cần nhiều tài nguyên.
* Hạn chế trong giao diện phức tạp: Khó quản lý HTML trực tiếp trong JSP.

**1.2 JSTL**

**1.2.1 Định nghĩa:**

JSTL (JavaServer Pages Standard Tag Library) là một thư viện các thẻ tiêu chuẩn dành cho JSP (JavaServer Pages). Nó cung cấp các thẻ sẵn có để thực hiện các tác vụ phổ biến như điều kiện, vòng lặp, định dạng, xử lý XML, và truy vấn cơ sở dữ liệu mà không cần phải viết mã Java trực tiếp trong JSP.

JSTL giúp mã JSP dễ đọc, dễ bảo trì hơn và giảm thiểu sự phụ thuộc vào mã Java phức tạp.

**1.2.2 Ưu điểm:**

* Giảm mã Java trong JSP: Giúp mã dễ đọc và tập trung hơn vào giao diện.
* Chuẩn hóa: Cung cấp các thẻ thống nhất cho các tác vụ phổ biến, giảm sự phụ thuộc vào mã tùy chỉnh.
* Tăng hiệu suất phát triển: Các thẻ đơn giản, dễ sử dụng, giúp giảm thời gian phát triển.
* Tái sử dụng và bảo trì dễ dàng: Cấu trúc mã gọn gàng, dễ chỉnh sửa và tái sử dụng.

**1.2.3 Nhược điểm:**

* Hạn chế về tính linh hoạt: Không thể thực hiện các tác vụ phức tạp như khi sử dụng mã Java trực tiếp.
* Phụ thuộc vào máy chủ hỗ trợ: Cần máy chủ hỗ trợ JSTL và cấu hình thư viện đầy đủ.
* Khả năng gỡ lỗi thấp: Nếu xảy ra lỗi trong thẻ JSTL, việc tìm nguyên nhân có thể phức tạp hơn so với lỗi Java thuần.
* Hiệu năng: Việc sử dụng nhiều thẻ JSTL có thể làm giảm hiệu suất so với mã Java tối ưu.

**1.3 CSS**

**1.3.1 Định nghĩa:**

CSS (Cascading Style Sheets) là một ngôn ngữ dùng để mô tả cách trình bày của tài liệu HTML. CSS quản lý các yếu tố như màu sắc, phông chữ, bố cục, khoảng cách, và cách các thành phần hiển thị trên trang web.

**1.3.2 Ưu điểm:**

* Tách biệt nội dung và giao diện: Giúp HTML tập trung vào cấu trúc, còn CSS đảm nhận giao diện.
* Tái sử dụng mã: Một file CSS có thể được áp dụng cho nhiều trang web, tiết kiệm công sức và thời gian.
* Dễ bảo trì: Thay đổi giao diện chỉ cần sửa trong file CSS mà không cần chỉnh sửa từng tệp HTML.
* Tối ưu hóa tốc độ tải trang: Sử dụng CSS ngoài (external CSS) giúp giảm kích thước mã HTML và tăng tốc độ tải trang.
* Hỗ trợ thiết kế đáp ứng (Responsive Design): Dễ dàng tạo các trang web hiển thị tốt trên nhiều thiết bị (máy tính, điện thoại, tablet).
* Hỗ trợ hiệu ứng và hoạt hình: CSS cung cấp các thuộc tính như transition, animation, và transform để tạo hiệu ứng giao diện bắt mắt.

**1.3.3 Nhược điểm:**

* Không kiểm soát chặt chẽ: Nếu không tổ chức tốt, mã CSS dễ trở nên phức tạp và khó bảo trì.
* Dễ xung đột: Khi có nhiều lớp (class) hoặc ID được áp dụng, việc ưu tiên của các quy tắc CSS (specificity) có thể gây khó hiểu.
* Không hỗ trợ logic phức tạp: CSS chỉ xử lý giao diện, không có khả năng logic hoặc tương tác như JavaScript.

**1.4 Javascript**

**1.4.1 Định nghĩa:**

JavaScript (JS) là một ngôn ngữ lập trình kịch bản (scripting language), chủ yếu được sử dụng để tạo nội dung động và tương tác trên các trang web. Là một phần của bộ ba công nghệ cốt lõi của web (HTML, CSS, và JavaScript), JS cho phép thêm các tính năng nâng cao như xử lý sự kiện, thay đổi DOM, tạo hiệu ứng, và kết nối với server qua AJAX hoặc APIs.

**1.4.2 Ưu điểm:**

* Chạy trên trình duyệt: Không cần cài đặt thêm phần mềm, JavaScript chạy trực tiếp trên các trình duyệt hiện đại.
* Tốc độ nhanh: JavaScript là ngôn ngữ biên dịch (interpreted) và chạy phía client, giúp giảm tải cho máy chủ.
* Đa nền tảng: Tương thích với mọi hệ điều hành và trình duyệt.
* Dễ học: Cú pháp đơn giản, dễ tiếp cận cho người mới bắt đầu.
* Động và tương tác: Giúp xây dựng giao diện động, tương tác cao cho website.

**1.4.3 Nhược điểm:**

* Vấn đề bảo mật: Vì chạy trên client, mã JavaScript dễ bị xem và sửa đổi, gây rủi ro về bảo mật.
* Khác biệt trình duyệt: Một số chức năng có thể hoạt động khác nhau trên các trình duyệt cũ hoặc không tuân thủ chuẩn.
* Lỗi runtime: Là ngôn ngữ động, lỗi trong JavaScript chỉ xuất hiện khi chạy, gây khó khăn trong việc kiểm tra lỗi sớm.
* Khó quản lý khi dự án lớn: Nếu không tổ chức tốt, mã JavaScript có thể trở nên lộn xộn và khó bảo trì.
* Hiệu suất hạn chế: JavaScript không phải lựa chọn tốt nhất cho các tác vụ tính toán phức tạp.

**1.5 Boostrap**

**1.5.1 Định nghĩa:**

Bootstrap là một framework CSS mã nguồn mở được phát triển bởi Twitter, giúp thiết kế giao diện web nhanh chóng và dễ dàng. Bootstrap cung cấp các component sẵn có như lưới (grid system), nút, biểu mẫu, điều hướng, và hỗ trợ thiết kế đáp ứng (responsive), giúp trang web hiển thị tốt trên mọi thiết bị**.**

**1.5.2 Ưu điểm:**

* Tiết kiệm thời gian: Cung cấp các mẫu giao diện sẵn, giảm thời gian thiết kế từ đầu.
* Responsive Design: Hệ thống lưới (grid system) mạnh mẽ giúp xây dựng giao diện phù hợp trên nhiều kích thước màn hình.
* Hỗ trợ đa trình duyệt: Hoạt động ổn định trên hầu hết các trình duyệt phổ biến.
* Tùy chỉnh dễ dàng: Cho phép ghi đè CSS hoặc tùy chỉnh trực tiếp với SCSS.
* Hỗ trợ JavaScript: Bao gồm các component tương tác như modal, dropdown, carousel mà không cần viết mã JavaScript phức tạp.

**1.5.3 Nhược điểm:**

* Tính đồng nhất giao diện: Nhiều website sử dụng Bootstrap có giao diện giống nhau nếu không tùy chỉnh.
* Tải trọng lớn: Sử dụng toàn bộ Bootstrap có thể gây lãng phí nếu chỉ dùng một phần nhỏ.
* Phụ thuộc vào class: Phải sử dụng nhiều class trong HTML, làm mã nguồn trở nên khó đọc hơn.
* Không tối ưu cho mọi dự án nhỏ: Với các dự án nhỏ, việc sử dụng Bootstrap có thể không cần thiết và làm tăng kích thước trang web.

**1.6 Servlet**

**1.6.1 Định nghĩa:**

Servlet là một lớp Java chạy trên máy chủ, được sử dụng để xử lý các yêu cầu từ phía client (chẳng hạn từ trình duyệt web), thực hiện xử lý logic và trả về phản hồi (response) dưới dạng HTML hoặc dữ liệu khác. Servlet hoạt động như một thành phần trung gian giữa giao diện người dùng và hệ thống backend trong các ứng dụng web dựa trên Java.

**1.6.2 Ưu điểm:**

* Hiệu suất cao: Servlet chạy trong môi trường máy chủ, được tối ưu hóa bởi JVM, giúp xử lý các yêu cầu nhanh hơn so với các ngôn ngữ thông dịch như CGI.
* Độc lập nền tảng: Servlet chạy trên bất kỳ máy chủ nào hỗ trợ Java (như Apache Tomcat, JBoss).
* Quản lý tài nguyên tốt: Servlet sử dụng các luồng (threads) thay vì tiến trình (processes), giúp tiết kiệm tài nguyên hệ thống.
* Tích hợp dễ dàng với Java: Tận dụng sức mạnh của các API Java để xử lý logic, kết nối cơ sở dữ liệu, và xử lý nâng cao.
* Bảo mật: Java hỗ trợ bảo mật mạnh mẽ, giúp Servlet an toàn hơn khi xử lý dữ liệu nhạy cảm.
* Khả năng mở rộng cao: Servlet dễ dàng tích hợp với các công nghệ Java khác như JSP, JDBC, Hibernate, và Spring.

**1.6.3 Nhược điểm:**

* Mã phức tạp: Servlet yêu cầu viết mã HTML trực tiếp trong Java, dẫn đến mã nguồn khó đọc và bảo trì.
* Thiếu tách biệt: Không tách biệt rõ ràng giữa logic xử lý (backend) và giao diện (frontend).
* Khó khăn trong thiết kế giao diện: Khi xây dựng giao diện phức tạp, việc nhúng HTML vào Servlet có thể gây rối.
* Phụ thuộc vào môi trường máy chủ: Servlet cần máy chủ ứng dụng hỗ trợ để chạy.

**1.7 JDBC**

**1.7.1 Định nghĩa:**

JDBC (Java Database Connectivity) là một API tiêu chuẩn trong Java, được sử dụng để kết nối và làm việc với các cơ sở dữ liệu (CSDL) quan hệ. JDBC cho phép ứng dụng Java thực hiện các tác vụ như gửi truy vấn SQL, cập nhật dữ liệu, và truy xuất dữ liệu từ các CSDL như MySQL, Oracle, SQL Server, PostgreSQL,..

**1.7.2 Ưu điểm:**

* Nền tảng độc lập: JDBC là một phần của Java, có thể làm việc với bất kỳ CSDL nào có driver hỗ trợ JDBC.
* Tiêu chuẩn hóa: Cung cấp API nhất quán, giúp nhà phát triển dễ dàng làm việc với các loại CSDL khác nhau mà không cần thay đổi nhiều mã nguồn.
* Tích hợp mạnh mẽ: Dễ dàng tích hợp với các công nghệ Java khác như Servlet, JSP, Spring, và Hibernate.
* Quản lý tài nguyên tự động: Các lớp như try-with-resources giúp quản lý kết nối, statement, và result set một cách tự động, giảm nguy cơ rò rỉ tài nguyên.
* Hỗ trợ xử lý giao dịch: Cho phép kiểm soát giao dịch (commit, rollback) để đảm bảo tính toàn vẹn dữ liệu.

**1.7.3 Nhược điểm:**

* Phụ thuộc vào CSDL cụ thể: Mặc dù JDBC chuẩn hóa API, nhưng driver JDBC có thể có những hành vi không đồng nhất giữa các CSDL.
* Quản lý kết nối thủ công: Yêu cầu nhà phát triển tự quản lý kết nối (connect, close), có thể gây rò rỉ tài nguyên nếu không cẩn thận.
* Viết mã phức tạp: Khi làm việc với các dự án lớn, mã JDBC có thể trở nên dài dòng và khó bảo trì.
* Thiếu trừu tượng hóa: Không cung cấp các tính năng trừu tượng hóa nâng cao như ORM (Object-Relational Mapping) của Hibernate hoặc JPA.

**1.8 SQL Server**

**1.8.1 Định nghĩa:**

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft. SQL Server sử dụng ngôn ngữ T-SQL (Transact-SQL) để quản lý và thao tác dữ liệu. Nó cung cấp các giải pháp quản lý cơ sở dữ liệu mạnh mẽ, hỗ trợ lưu trữ, truy vấn, phân tích, và bảo mật dữ liệu.

**1.8.2 Ưu điểm:**

* Hiệu suất cao: Tối ưu hóa xử lý truy vấn và khả năng quản lý khối lượng dữ liệu lớn.
* Tích hợp tốt với hệ sinh thái Microsoft: Hỗ trợ dễ dàng với các ứng dụng như .NET, Power BI, và Azure.
* Bảo mật mạnh mẽ: Cung cấp các tính năng như mã hóa dữ liệu, xác thực người dùng, và kiểm soát quyền truy cập.
* Hỗ trợ đa nền tảng: SQL Server có thể chạy trên cả Windows lẫn Linux.
* Khả năng sao lưu và phục hồi tốt: Hỗ trợ backup tự động, phục hồi dữ liệu theo thời gian.

**1.8.3 Nhược điểm:**

* Chi phí cao: Phiên bản đầy đủ của SQL Server (Enterprise Edition) có giá thành cao, phù hợp cho doanh nghiệp lớn.
* Yêu cầu tài nguyên hệ thống: SQL Server cần nhiều tài nguyên để hoạt động tối ưu, đặc biệt với các hệ thống lớn.
* Phức tạp khi xử lý đa nền tảng: Mặc dù hỗ trợ Linux, SQL Server vẫn tối ưu hóa hơn trên hệ điều hành Windows.
* Không phù hợp với cơ sở dữ liệu phi quan hệ: Thiếu hỗ trợ đầy đủ cho NoSQL hoặc các kiểu cơ sở dữ liệu không quan hệ.

**1.9 Decorator Sitemesh**

**1.9.1 Định nghĩa:**

Sitemesh là một thư viện Java được sử dụng để trang trí (decorator) các trang web trong các ứng dụng Java web, giúp quản lý giao diện người dùng (UI) một cách dễ dàng hơn. Sitemesh cho phép bạn tách riêng phần giao diện (header, footer, menu, v.v.) khỏi nội dung của trang, giúp tái sử dụng và duy trì các yếu tố UI mà không cần phải lặp lại mã HTML trên mỗi trang.

Decorator trong Sitemesh là một mô hình thiết kế cho phép bạn sử dụng một "layout" chung (mẫu bố cục) cho tất cả các trang của website, từ đó chỉ cần thay đổi nội dung chính của từng trang mà không ảnh hưởng đến cấu trúc chung.

**1.9.2 Ưu điểm:**

* Tách biệt giao diện và nội dung: Giúp phân tách phần giao diện (header, footer, sidebar, etc.) khỏi nội dung của từng trang, giúp dễ dàng bảo trì và thay đổi giao diện mà không cần thay đổi toàn bộ mã HTML.
* Tiết kiệm thời gian phát triển: Sử dụng layout chung cho tất cả các trang giúp giảm bớt công việc lặp lại khi xây dựng các trang web lớn.
* Dễ dàng tái sử dụng: Các layout có thể được tái sử dụng cho nhiều trang mà không cần phải sao chép lại mã HTML của header và footer trong từng trang.
* Khả năng mở rộng: Dễ dàng tích hợp với các ứng dụng Java web, đặc biệt là với các framework như Spring MVC.

**1.9.3 Nhược điểm:**

* Cấu hình phức tạp: Việc cấu hình Sitemesh trong một ứng dụng Java web có thể phức tạp, đặc biệt khi làm việc với các cấu trúc thư mục phức tạp.
* Không hỗ trợ mạnh mẽ các tính năng động: Nếu ứng dụng web của bạn cần tính năng động, chẳng hạn như Ajax hay cập nhật theo thời gian thực, Sitemesh có thể không phải là lựa chọn tốt nhất.
* Giới hạn về tính linh hoạt: Việc sử dụng các layout cố định có thể làm giảm tính linh hoạt trong việc thay đổi giao diện của từng trang trong một số tình huống.

**1.10. JWT**

**1.10.1 Định nghĩa:**

JWT (JSON Web Token) là một tiêu chuẩn mở (RFC 7519) để truyền tải thông tin an toàn dưới dạng JSON giữa các bên. JWT thường được sử dụng trong các hệ thống xác thực và cấp quyền (authentication and authorization). Token này có thể được mã hóa và/hoặc ký để bảo vệ tính toàn vẹn và bảo mật của dữ liệu.

JWT có ba phần chính:

* Header: Mô tả loại token và thuật toán mã hóa (thường là HMAC SHA256 hoặc RSA).
* Payload: Chứa các tuyên bố (claims) về đối tượng, ví dụ như thông tin người dùng và quyền truy cập. Payload không được mã hóa, vì vậy không nên chứa dữ liệu nhạy cảm.
* Signature: Được sử dụng để xác minh tính toàn vẹn của token và đảm bảo rằng payload không bị thay đổi. Được tạo ra bằng cách mã hóa header và payload với một khóa bí mật hoặc khóa công khai.

**1.10.2 Ưu điểm:**

* Tự chứa thông tin: JWT có thể lưu trữ thông tin người dùng và quyền truy cập trong token, giúp giảm tải cho máy chủ. Không cần lưu trạng thái trên máy chủ.
* Quản lý dễ dàng: Dễ dàng cấp và kiểm tra quyền truy cập bằng cách xác thực token trong mỗi yêu cầu.
* Không cần lưu trữ session trên server: JWT có thể được lưu trữ ở client (cookies hoặc localStorage), giúp giảm tải cho server.
* Dễ dàng truyền tải: JWT có thể được gửi qua HTTP header, URL parameters, hoặc trong body của POST request.

**1.10.3 Nhược điểm:**

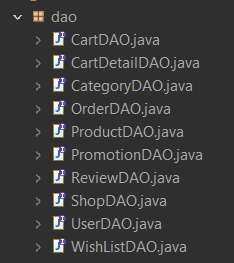
* Không thể thay đổi sau khi phát hành: Một khi token được cấp phát, nó không thể thay đổi. Nếu cần thay đổi quyền hoặc thông tin, phải cấp phát token mới.
* Lưu trữ trên client: Vì JWT thường được lưu trữ ở client (cookies, localStorage), nếu không bảo vệ tốt, nó có thể bị đánh cắp (cross-site scripting - XSS).
* Không có khả năng thu hồi (revocation): Khi token đã được cấp phát, không thể thu hồi nó ngay lập tức mà không có một hệ thống đặc biệt để theo dõi và quản lý trạng thái của token.
* Kích thước lớn: Vì JWT chứa thông tin (payload), kích thước của nó có thể lớn hơn so với các phương thức xác thực khác, ảnh hưởng đến băng thông.

**2. Tầng dữ liệu:**

Đây là tầng chịu trách nhiệm lưu trữ, truy xuất và xử lý dữ liệu từ cơ sở dữ liệu (MySQL). Bao gồm các thành phần chính:

**DAO:**

* Chịu trách nhiệm giao tiếp trực tiếp với cơ sở dữ liệu, thực hiện các thao tác CRUD như thêm, sửa, xóa, truy vấn.
* Tách biệt hoàn toàn logic truy cập dữ liệu khỏi các tầng khác, đảm bảo tính modular và dễ bảo trì.
* Sử dụng JDBI để đơn giản hóa truy vấn SQL, giúp tăng tính hiệu quả trong việc xử lý dữ liệu.

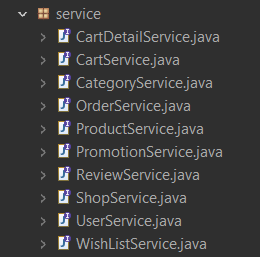


**3. Tầng trung gian:**

Tầng này xử lý logic nghiệp vụ và là cầu nối giữa tầng dữ liệu (DAO) và tầng trình bày (Servlet).

**Service**

* Thực hiện các quy trình nghiệp vụ của hệ thống như xác nhận đơn hàng, kiểm tra thông tin người dùng, và xử lý logic tính toán (nếu có).
* Đóng vai trò như một lớp trung gian để gọi đến các phương thức của DAO hoặc xử lý dữ liệu trước khi gửi đến tầng Servlet.
* Giúp tách biệt logic nghiệp vụ khỏi tầng giao diện, đảm bảo tính linh hoạt và dễ dàng mở rộng.



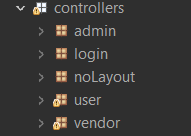
**4. Tầng trình bày:**

Đây là tầng giao tiếp với người dùng thông qua giao diện web. Tầng này bao gồm các thành phần điều khiển và hiển thị.

**Servlet**

* Các Servlet hoạt động như bộ điều khiển (controller), xử lý yêu cầu từ phía người dùng và trả về kết quả tương ứng.
* Được chia thành các nhánh cụ thể để dễ quản lý:
* Admin: Xử lý các chức năng liên quan đến quản trị như quản lý sản phẩm, danh mục, người dùng, đơn hàng.
* client: Đáp ứng các yêu cầu từ phía khách hàng như tìm kiếm sản phẩm, xem chi tiết sản phẩm, quản lý đánh giá sản phẩm.
* general: Chứa các chức năng dùng chung như xử lý đăng nhập, đăng ký, hoặc cấu hình chung của hệ thống.

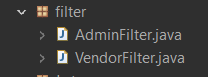
Ngoài ra còn có thể chia theo các role: admin, vendor, user và login



**Filter**

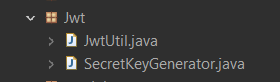
Thực hiện các tác vụ trước hoặc sau khi Servlet xử lý yêu cầu, ví dụ:

* Kiểm tra quyền truy cập (authentication và authorization).
* Ghi log các yêu cầu của người dùng (logging).
* Tối ưu hiệu suất thông qua caching hoặc xử lý bộ nhớ tạm.



**JWT**

* Xác thực người dùng: Được sử dụng để xác thực người dùng trong các hệ thống phân tán.
* Cung cấp quyền truy cập: Chứa thông tin về quyền và quyền truy cập của người dùng.
* Bảo mật thông tin: Thông qua việc ký kết token, JWT đảm bảo tính toàn vẹn và bảo mật của dữ liệu.



## CHƯƠNG 3: PHÂN TÍCH CHỨC NĂNG

**1. Mô tả chức năng:**

|  |  |
| --- | --- |
| **Chức năng** | **Mô tả** |
| Đăng nhập | Người dùng sử dụng tài khoản đã đăng ký để đăng nhập. |
| Đăng ký | Người dùng có thể đăng ký bằng cách các thông tin cơ bản như: họ tên, email, số điện thoại, địa chỉ, tên tài khoản và mật khẩu. |
| Quên mật khẩu | Sử dụng email để xác thực người dùng và sau đó đổi mật khẩu mới. |
| Quản lý thông tin cá nhân | Cập nhật thông tin cá nhân. |
| Quản lý tài khoản người dùng | Đăng ký tài khoản mới.  Chỉnh sửa thông tin tài khoản.  Quản lý quyền truy cập tài khoản (vendor và khách hàng). |
| Quản lý sản phẩm và doanh mục | Thêm, chỉnh sửa, xóa và tìm kiếm sản phẩm.  Thêm, chỉnh sửa, xóa và tìm kiếm doanh mục. |
| Quản lý giỏ hàng | Thêm, chỉnh sửa, xóa giỏ hàng.  Hiển thị thông tin giỏ hàng. |
| Thanh toán | Thanh toán và ghi nhận đơn hàng. |
| Quản lý đơn hàng | Hiển thị danh sách đơn hàng, tìm kiếm đơn hàng.  Cập nhật tình trạng đơn hàng (thanh toán, chưa thanh toán) |
| Sản phẩm yêu thích | Thêm, xóa sản phẩm yêu thích |
| Đánh giá sản phẩm | Đánh giá sản phẩm và cho điểm sản phẩm |
| Khuyến mãi | Áp dụng mã khuyến mãi để giảm giá đơn hàng |
| Quản lý cửa hàng | Cho phép thêm, xóa, sửa cửa hàng  Cập nhật trạng thái của cửa hàng (hoạt động, ngừng hoạt động) |
| Doanh thu | Tính toán doanh thu theo ngày, tháng, quý và năm. |

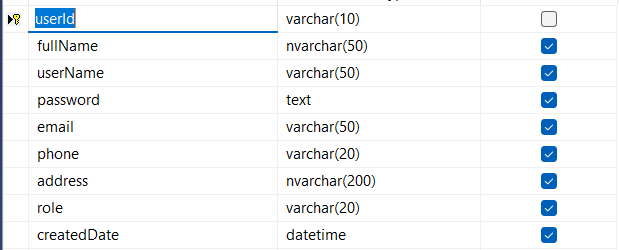
**2. Sơ đồ chức năng:**



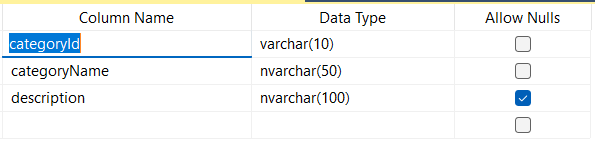
## CHƯƠNG 4: PHÂN TÍCH YÊU CẦU DỮ LIỆU

**1. Mô tả thiết kế cơ sở dữ liệu**

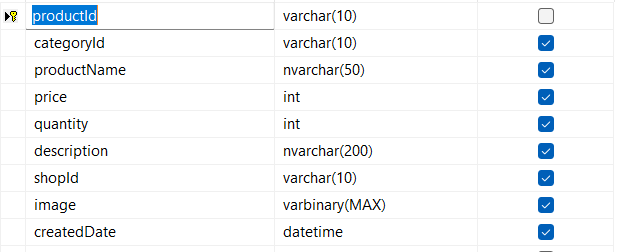
-Bảng User



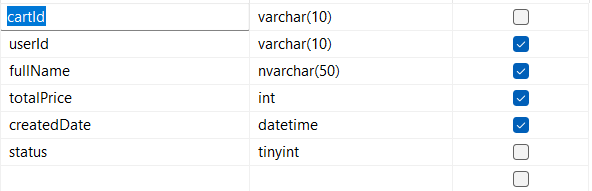
-Bảng Category



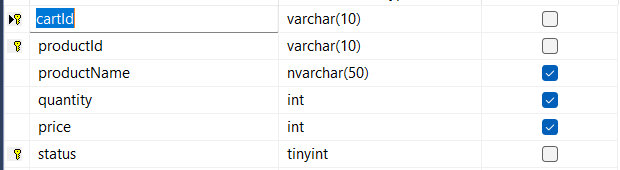
-Bảng Product



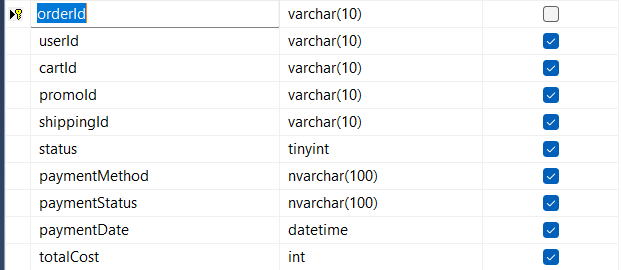
-Bảng cart



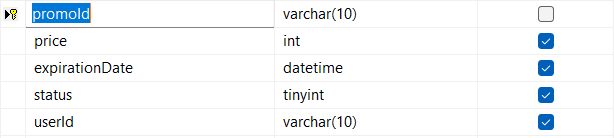
-Bảng cartDetail



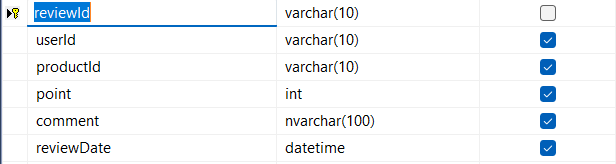
-Bảng order



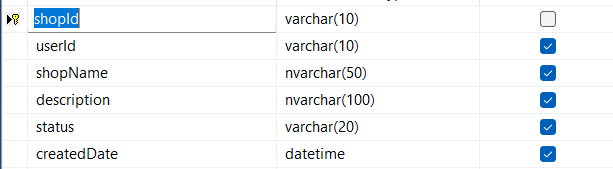
-Bảng promotion



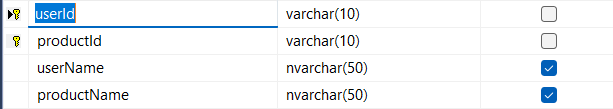
-Bảng review



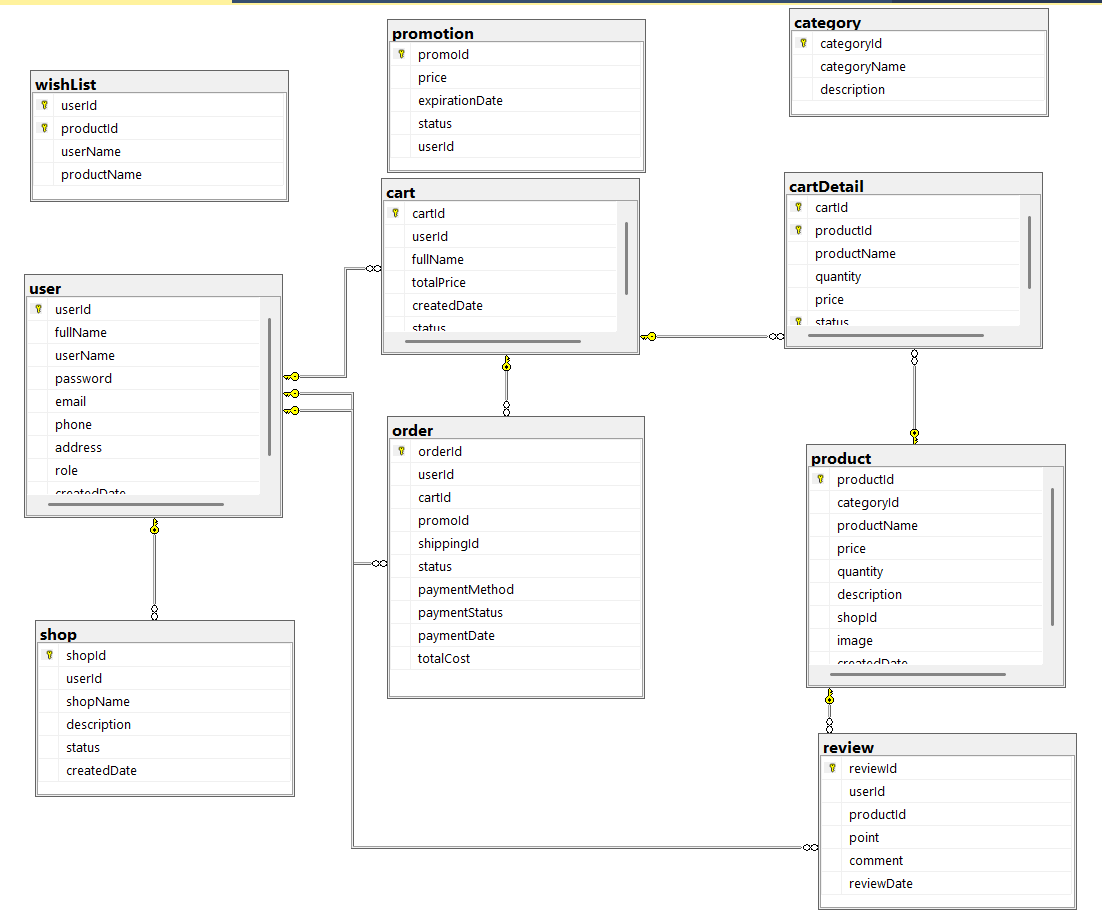
-Bảng shop



-Bảng wishList

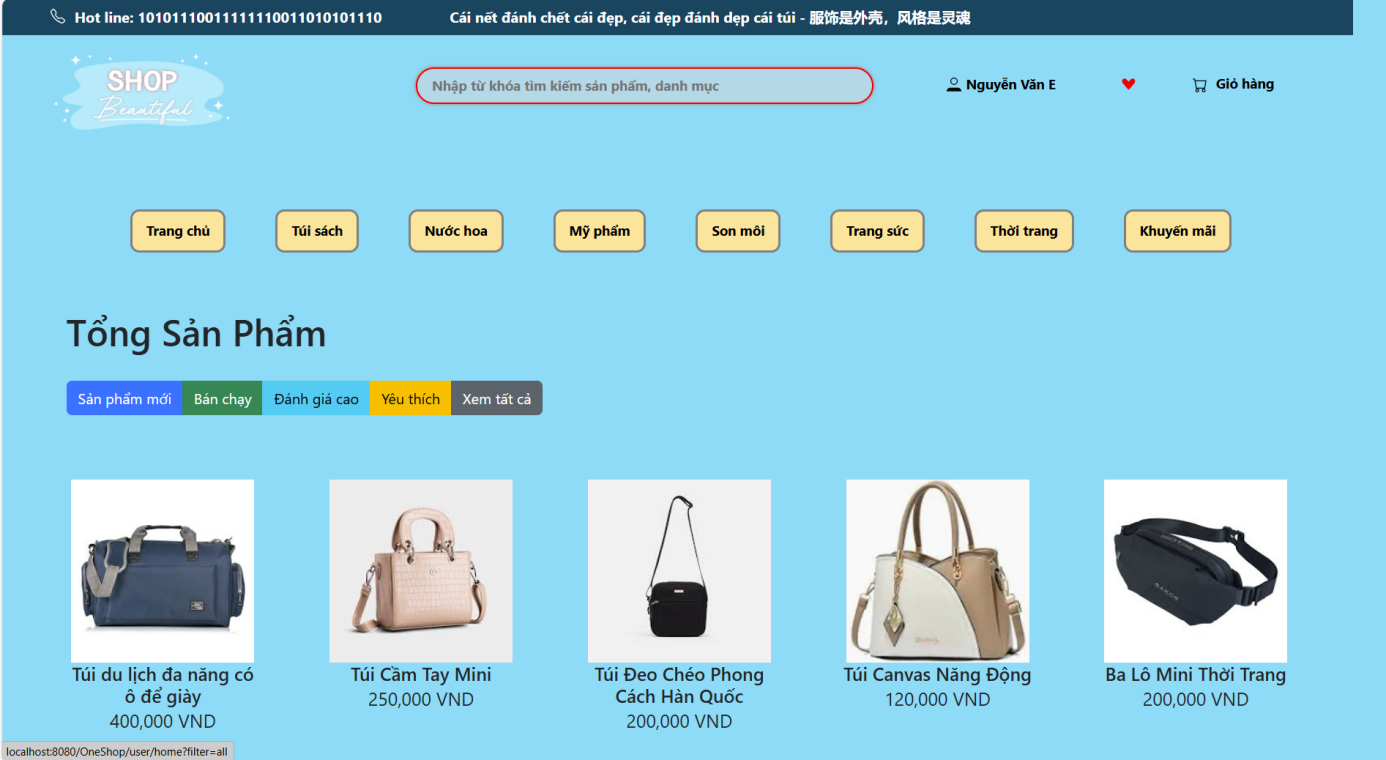


**2. Mô tả sơ đồ diagram**



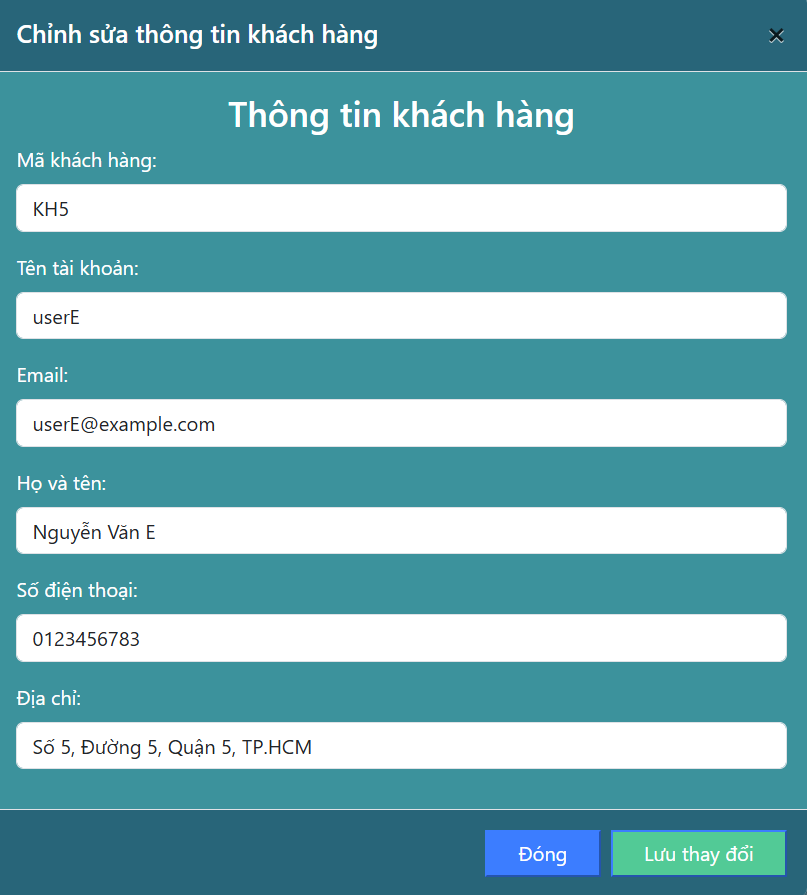
## CHƯƠNG 5: KẾT QUẢ ĐẠT ĐƯỢC

1. **Giao diện của khách hàng:**



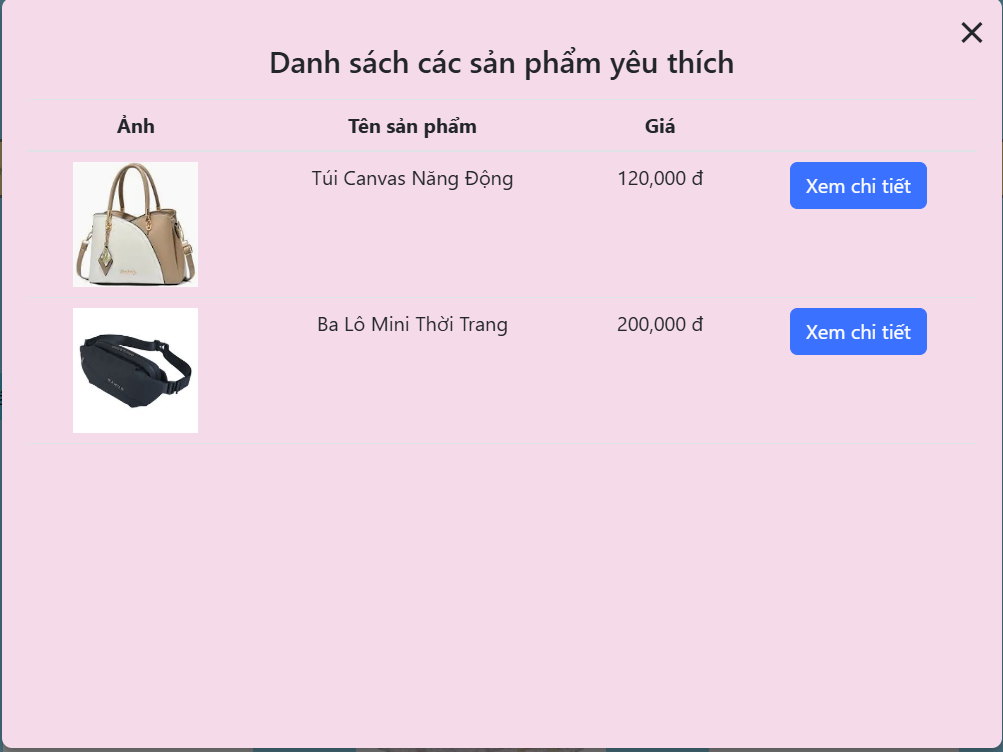
**- Mô tả chức năng:** Trang này cho phép người dùng xem sản phẩm bán chạy, sản phẩm được nhiều người yêu thích,… Menu ngang để tìm kiếm các sản phẩm theo sự lựa chọn của bản thân, hay tìm kiếm sản phẩm, xem danh sách các sản phẩm mà bản thân yêu thích, xem thông tin cá nhân và xem giỏ hàng của bản thân. Có thể xem những thông tin về UTE SHOP như là giới thiệu, tuyển dụng, liên hệ, tin tức và blog, chính sách và điều khoản, hổ trợ khách hàng.

Giao diện thông tin của bản thân



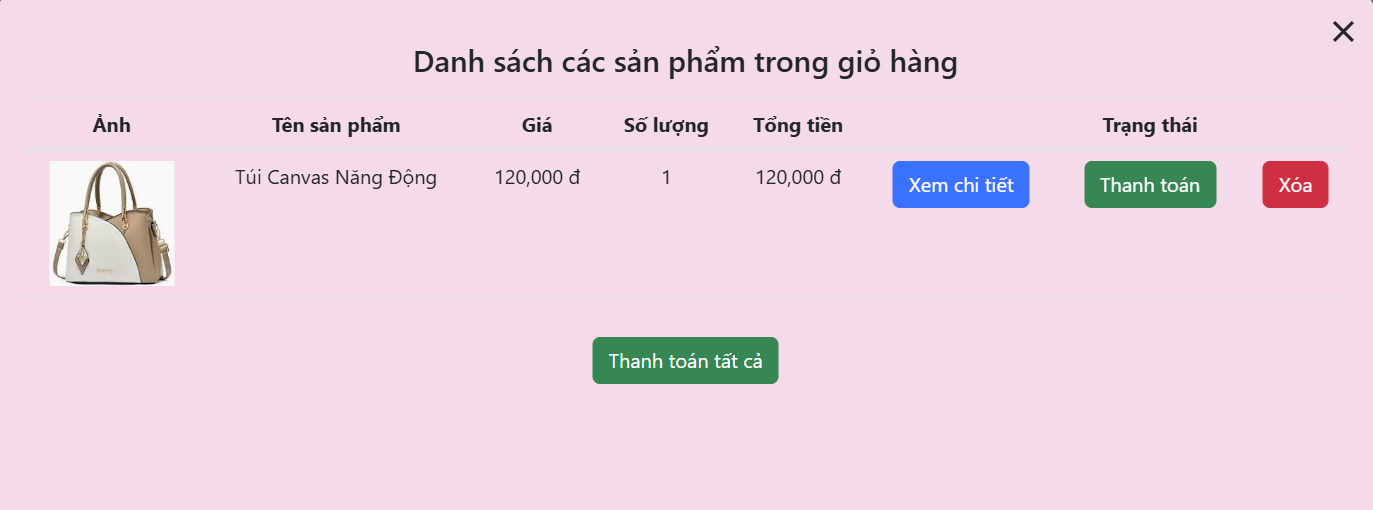
-Mô tả: xem thông tin cá nhân và có thể chỉnh sửa họ tên, email, số điện thoại và địa chỉ.

Giao diện các sản phẩm yêu thích khi ấn vào icon hình trái tim.



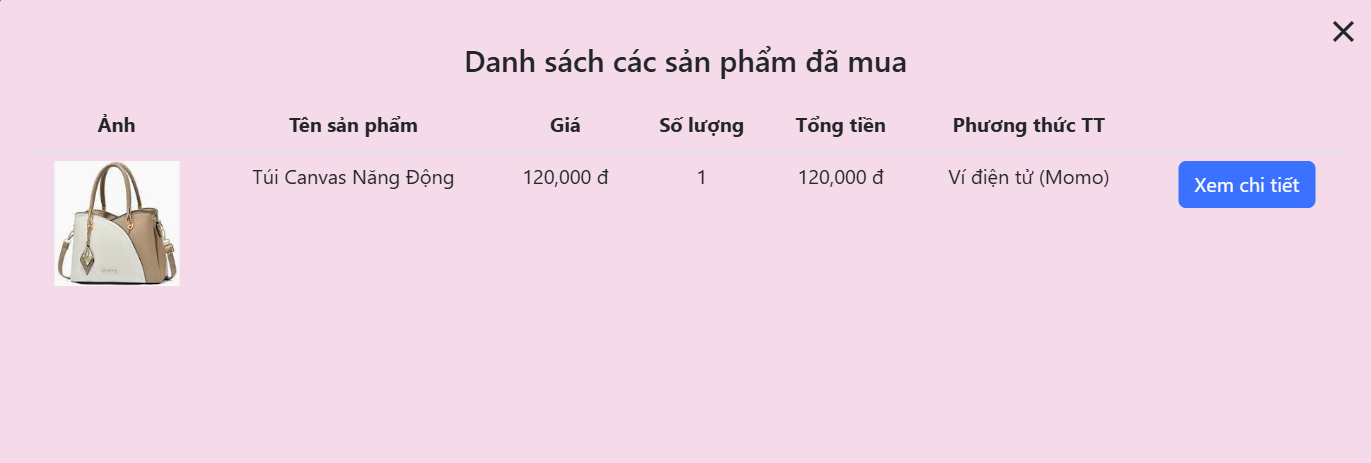
Ở phần giỏ hàng ta có hai lựa chọn chưa thanh toán và đơn hàng

Giao diện khi ấn vào chưa thanh thanh toán

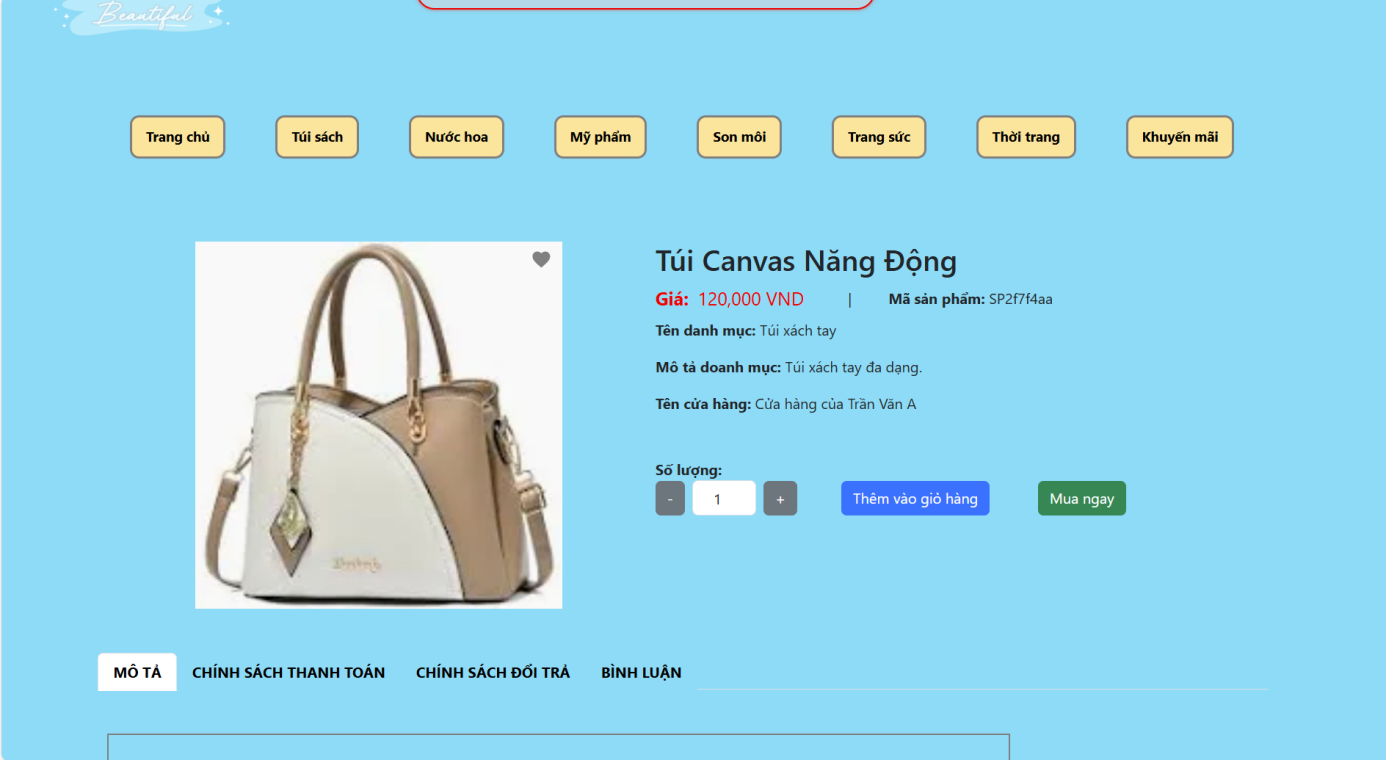


- Mô tả: xem được danh sách sản phẩm yêu thích có thể vào xem chi tiết để xem chi tiết sản phẩm, thanh toán và xóa sản phẩm, có thể thanh toán tất cả nếu có nhiều sản phẩm.

Giao diện khi ấn vào đơn hàng

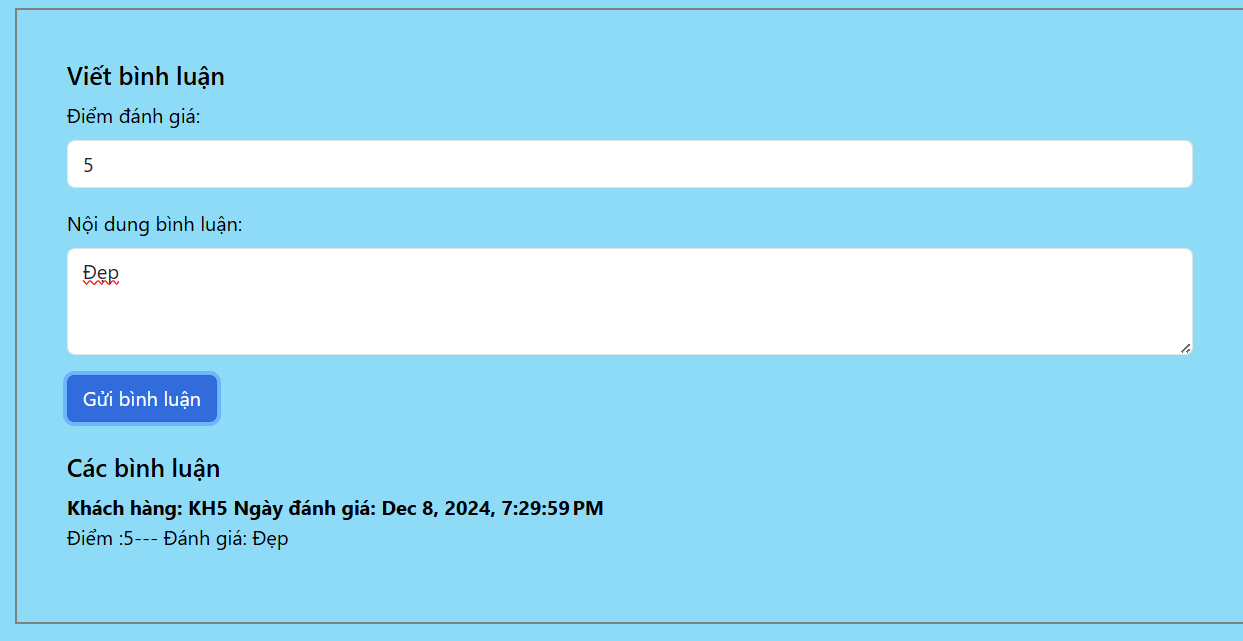
  
 - Hiên ra danh sách các đơn hàng đã được thanh toán và ấn xem chi tiết để xem thông tin đơn hàng đã được thanh toán.

Giao diện khi ấn vào sản phẩm bất kì hoặc khi ấn vào xem chi tiết sản phẩm:



- Sẽ có những thông tin như giá, mã sản phẩm và các mô tả sản phẩm, cũng như chính sách thanh toán, chính sách đổi trả và phần bình luận để lại đánh giá của khách hàng. Có nút mua ngay hoặc thêm vào giỏ hàng và thêm vào danh sách sản phẩm yêu thích ở nút hình trái tim góc phải hình ảnh.

Khung bình luận

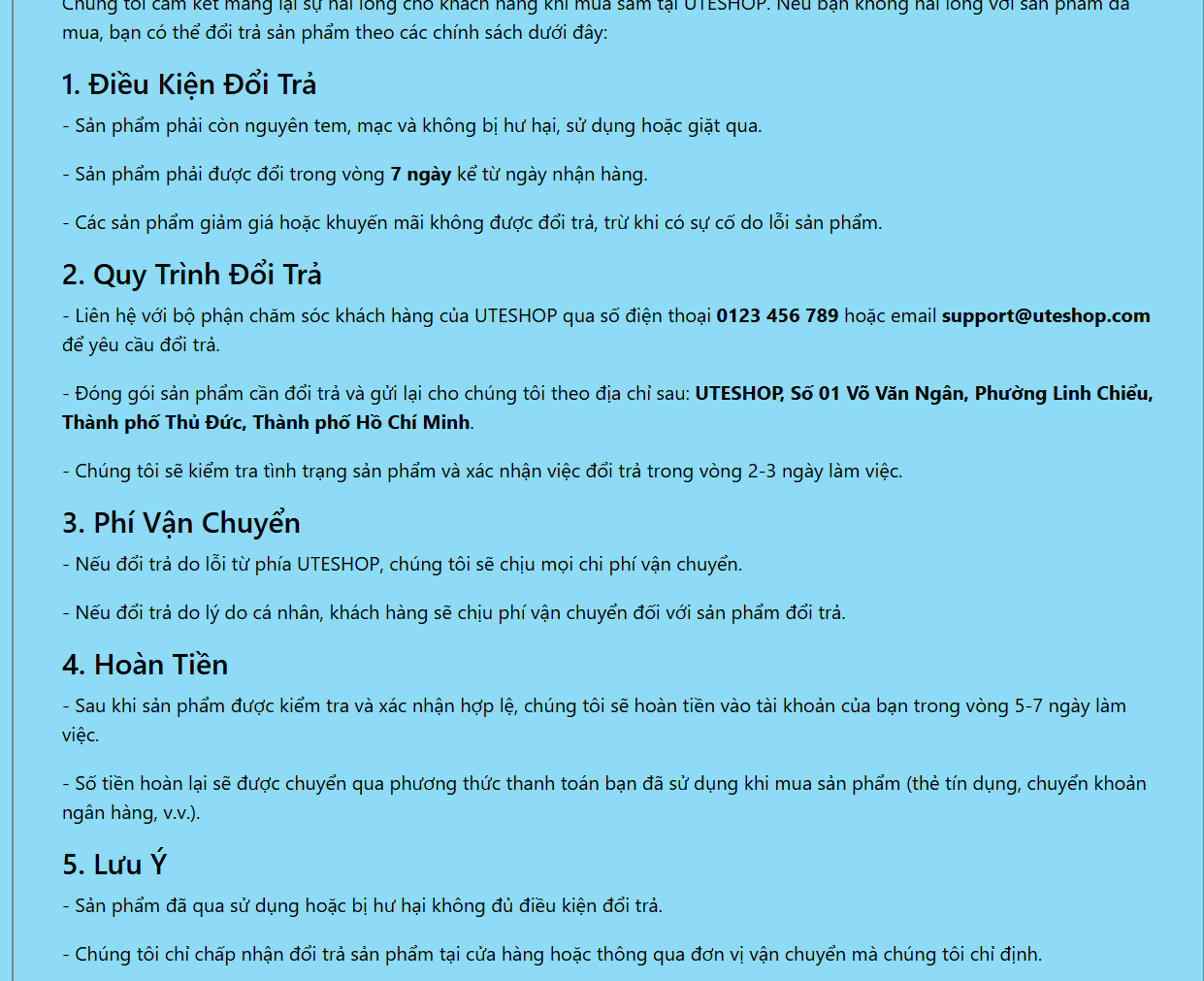


Sẽ có phần điểm đánh giá và nội dung bình luận đến từ khách hàng.

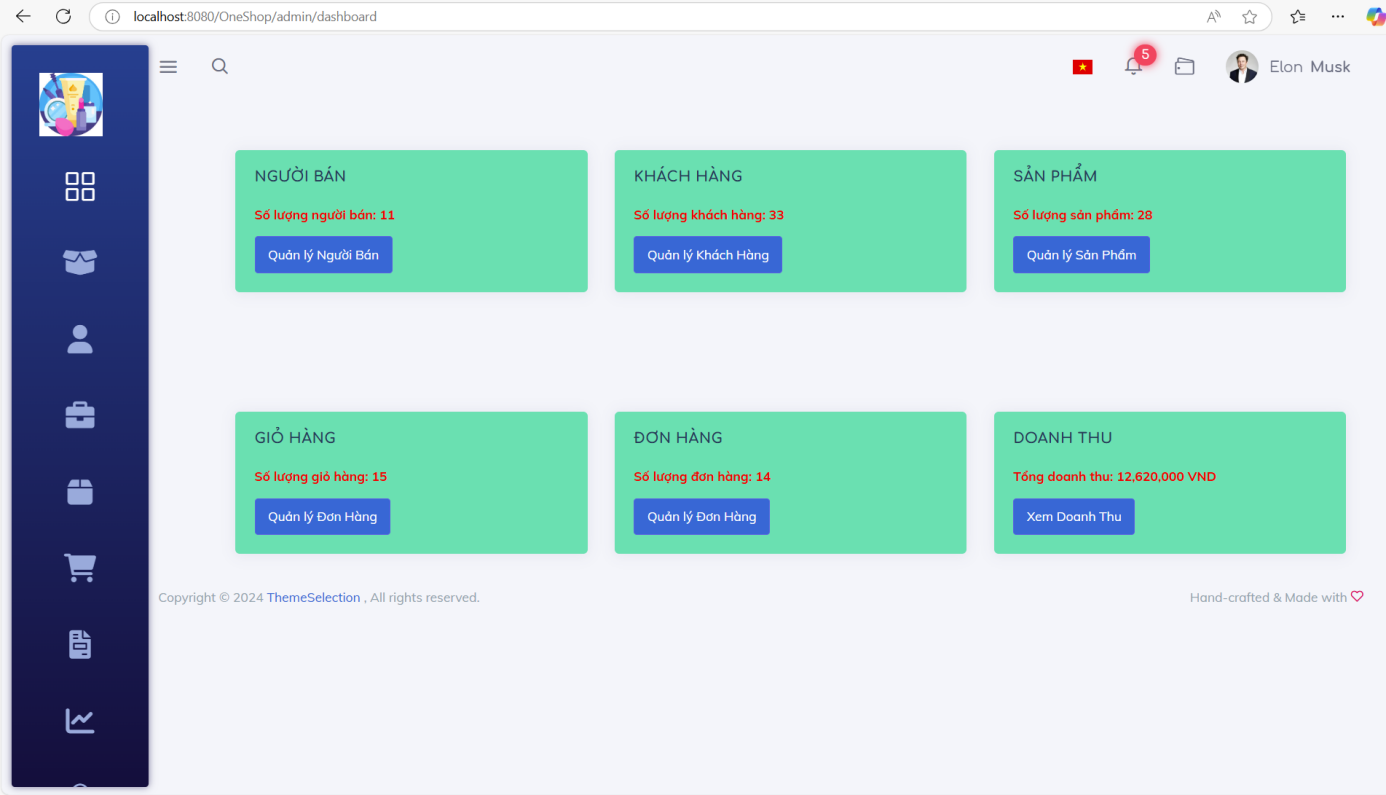
Chính sách thanh toán



Chính sách đổi trả

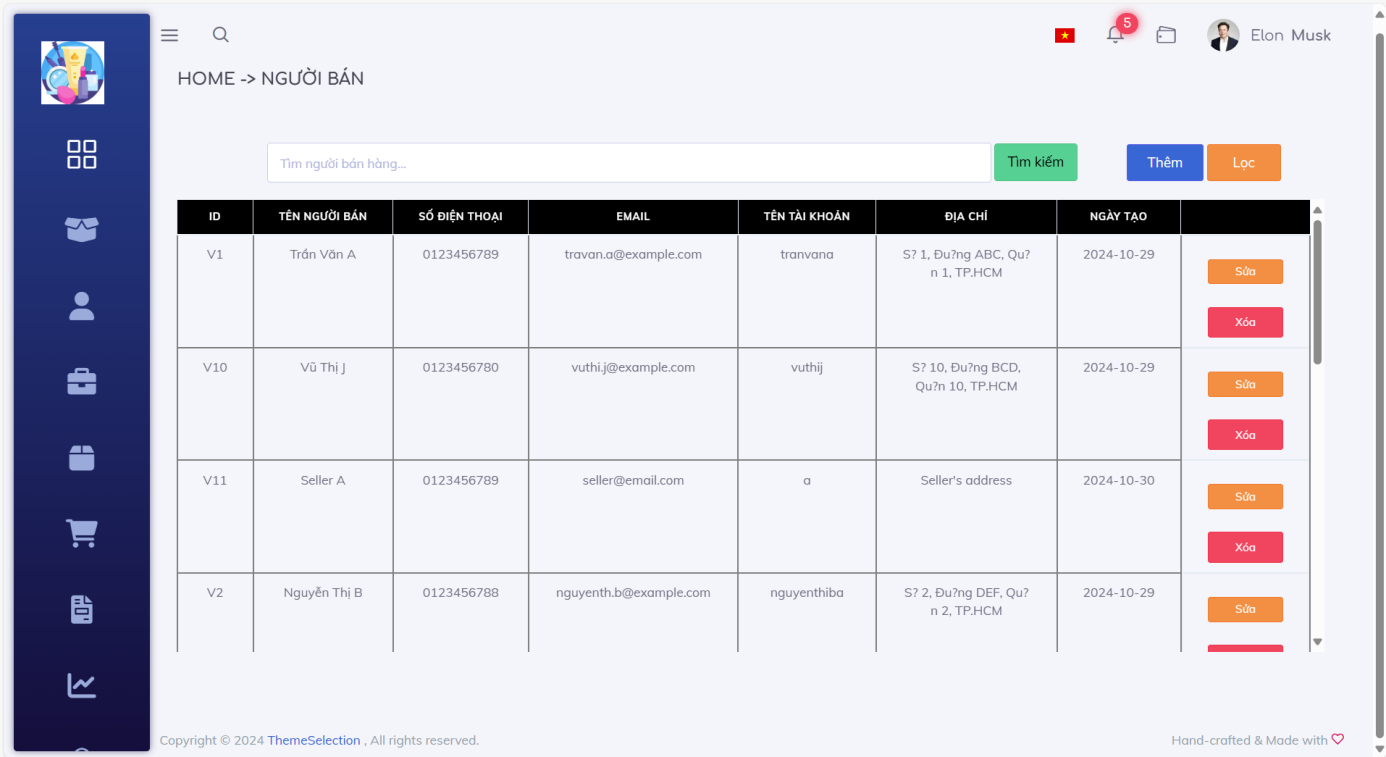


1. **Giao diện của trang Admin**

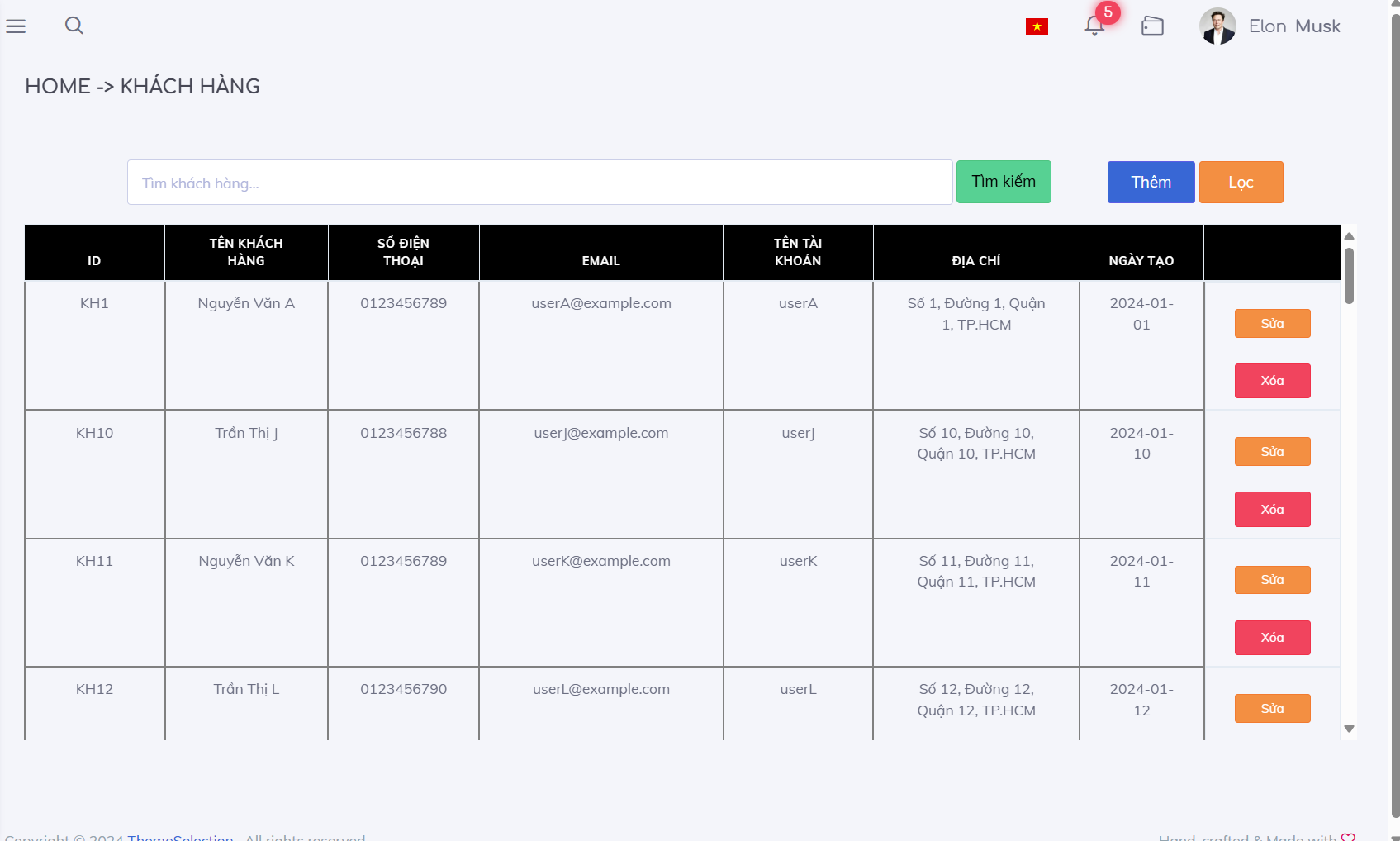
****

Giao diện trang chủ của Admin có những khung hiện quản lý các sản phẩm, đơn hàng, giỏ hàng, đơn hàng, doanh thu và người bán.

Khi ấn vào Quản lí người bán chúng ta sẽ có những danh sách người bán cũng như chúng ta có thể tìm kiếm thêm xóa sửa và lọc các người bán. Xem những thông tin của người bán như Tên người bán, số điện thoại, email, tên tài khoản, địa chỉ, ngày tạo.

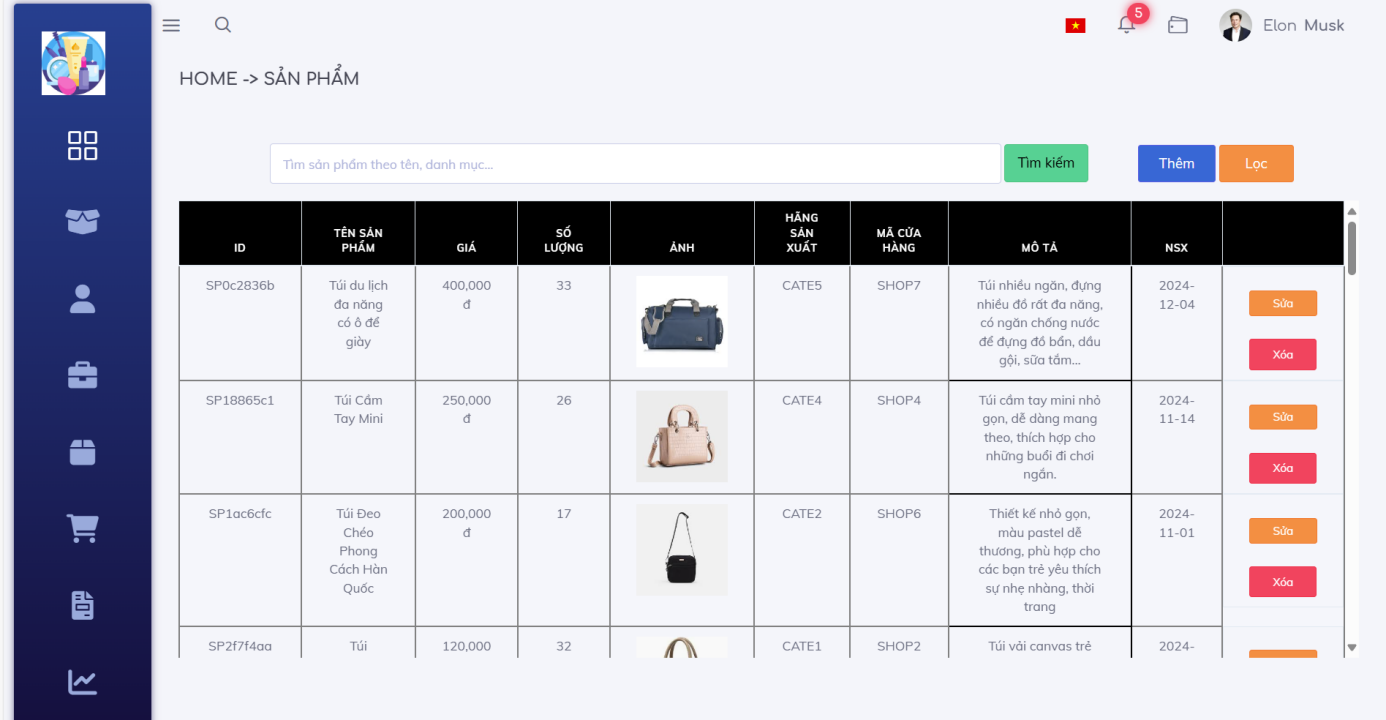


Tương tự như trang quản lí người bán thì giao diện quản lí khách hàng cũng vậy.



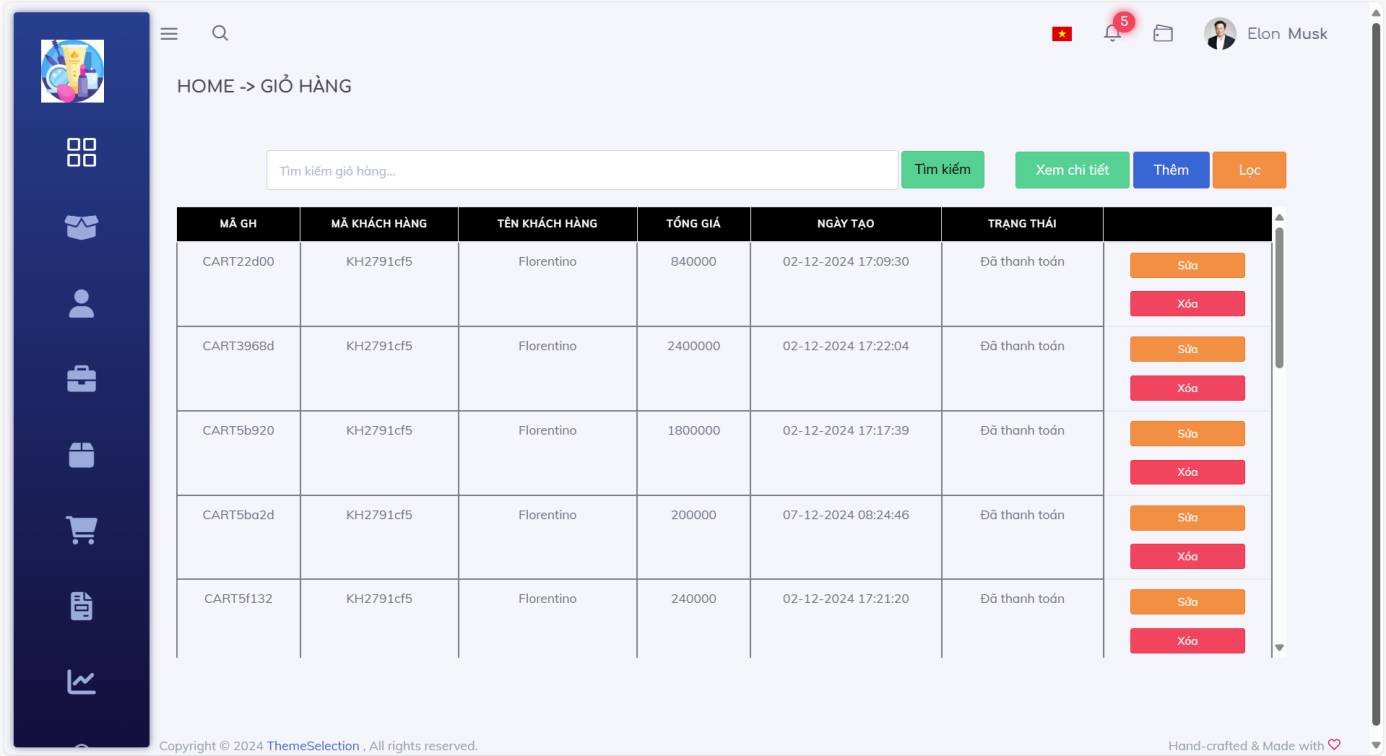
Các chức năng có thể thực hiện được trên trang này là CRUD khách hàng và xem được thông tin của khách hàng như tên khách hàng, số điện thoại, email, tên tài khoản, địa chỉ, và ngày tạo.

Giao diện của trang quản lí sản phẩm



Các chức năng có thể thực hiện được trên trang này là CRUD sản phẩm, xem thông tin của các sản phẩm như tên sản phẩm, giá, số lượng, ảnh, hảng sản xuất, mã cữa hàng, mô tả, ngày sản xuất.

Giao diện của trang quản lí giỏ hàng.

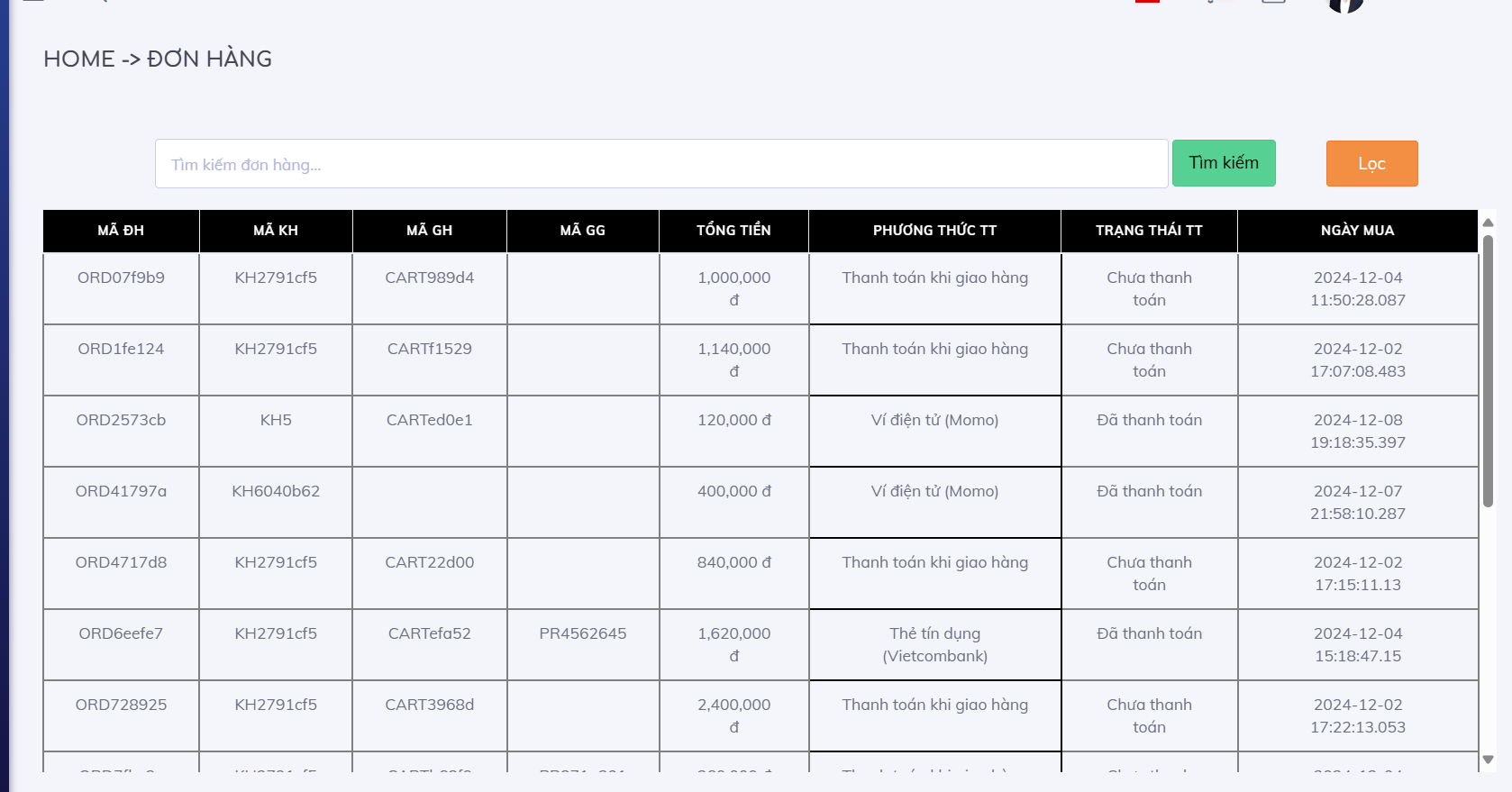


Các chức năng tương tự như quản lí người bán có thêm chức năng xem chi tiết giỏ hàng khi ấn vào nút Xem chi tiết sẽ cho ra giao diện này:



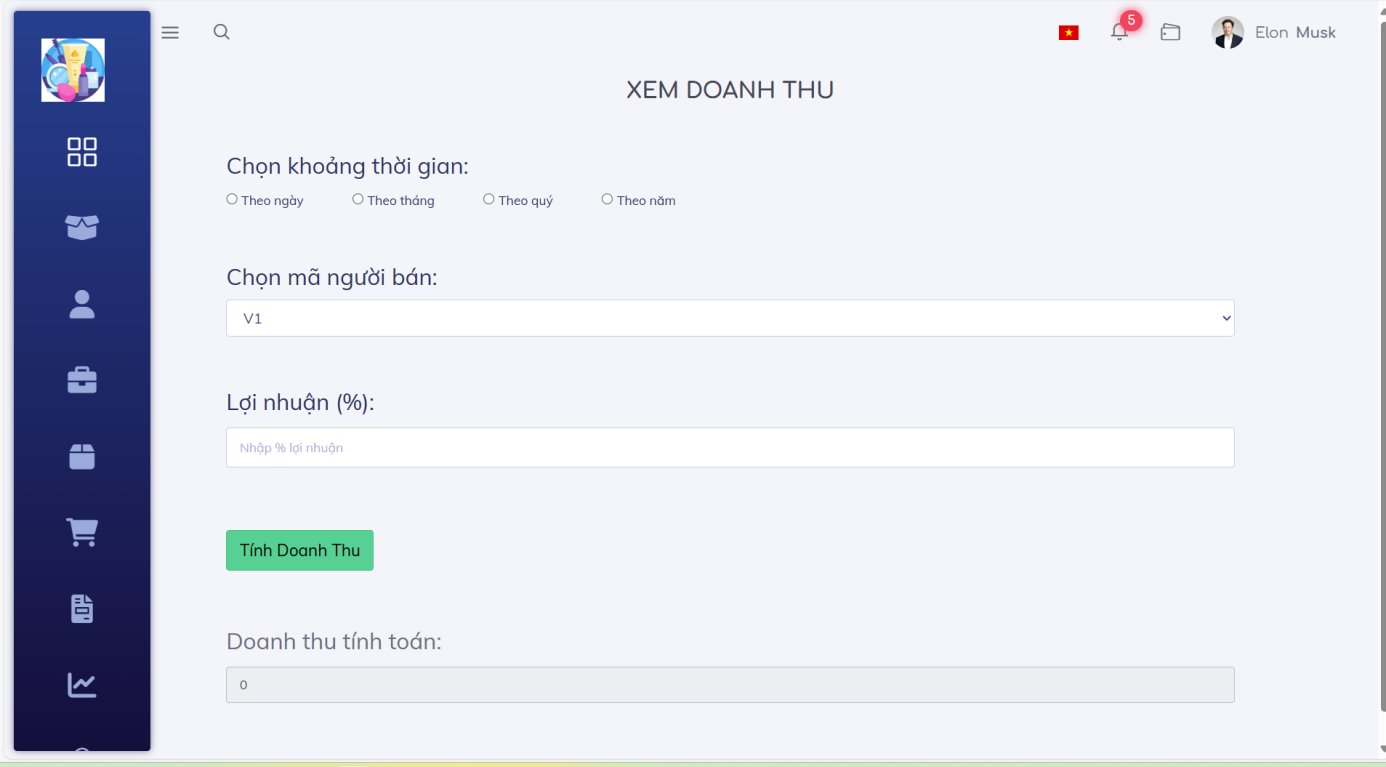
Ở đây sẽ có chức năng them xóa sửa lọc các chi tiết giỏ hàng theo mã giỏ hàng. Cũng như là xem được các thông tin của chi tiế giỏ hàng như mã sản phẩm, tên sản phẩm, số lượng, giá , trạng thái và mã giỏ hàng

Giao diện trang quản lí đơn hàng



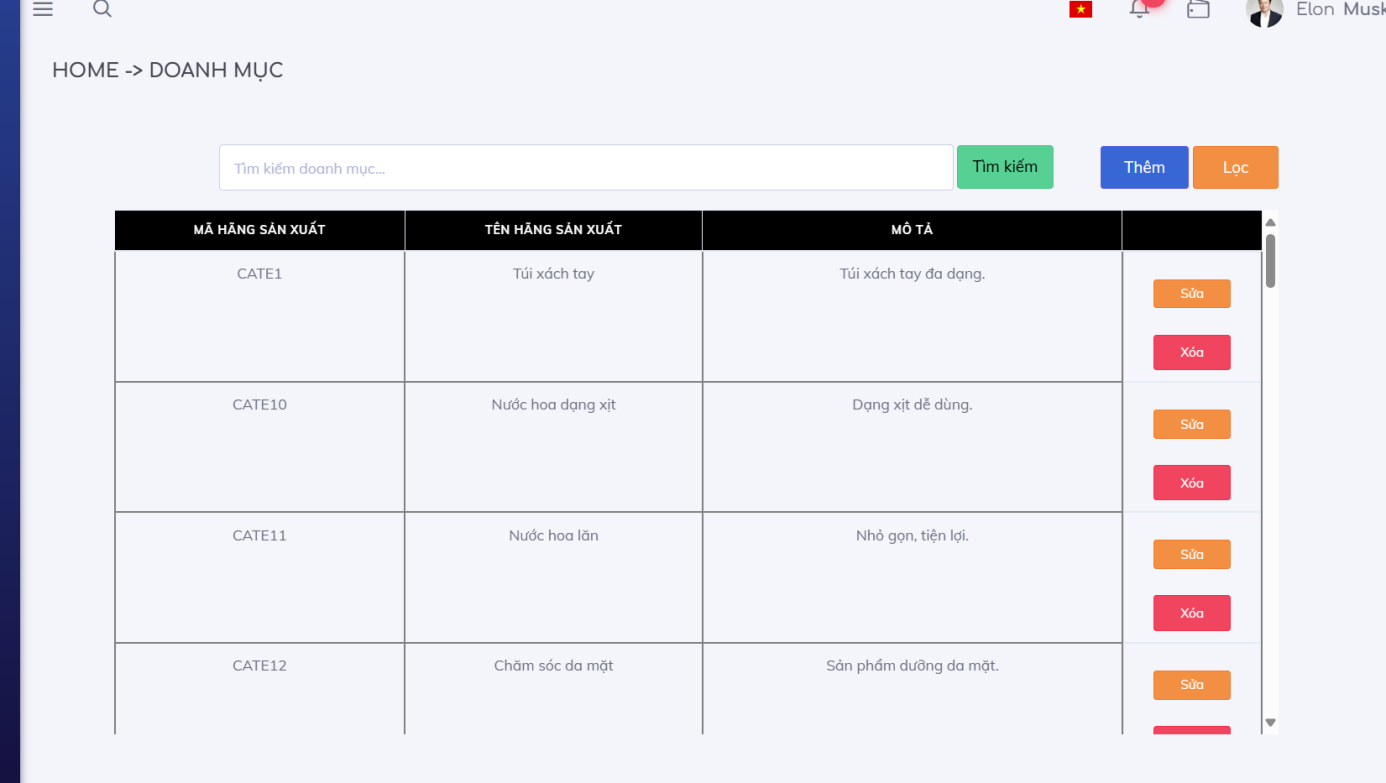
Hiện ra danh sách chứa các thông tin như mã đơn hàng, mã khách hàng, mã giỏ hàng, tổng tiền, phương thức thanh toán, trạng thái thanh toán, ngày mua. Tìm kiếm các đơn hàng và lọc

Trang quản lí doanh thu



Sẽ tính doanh thu của từng người bán theo ngày, tháng, quý và năm.

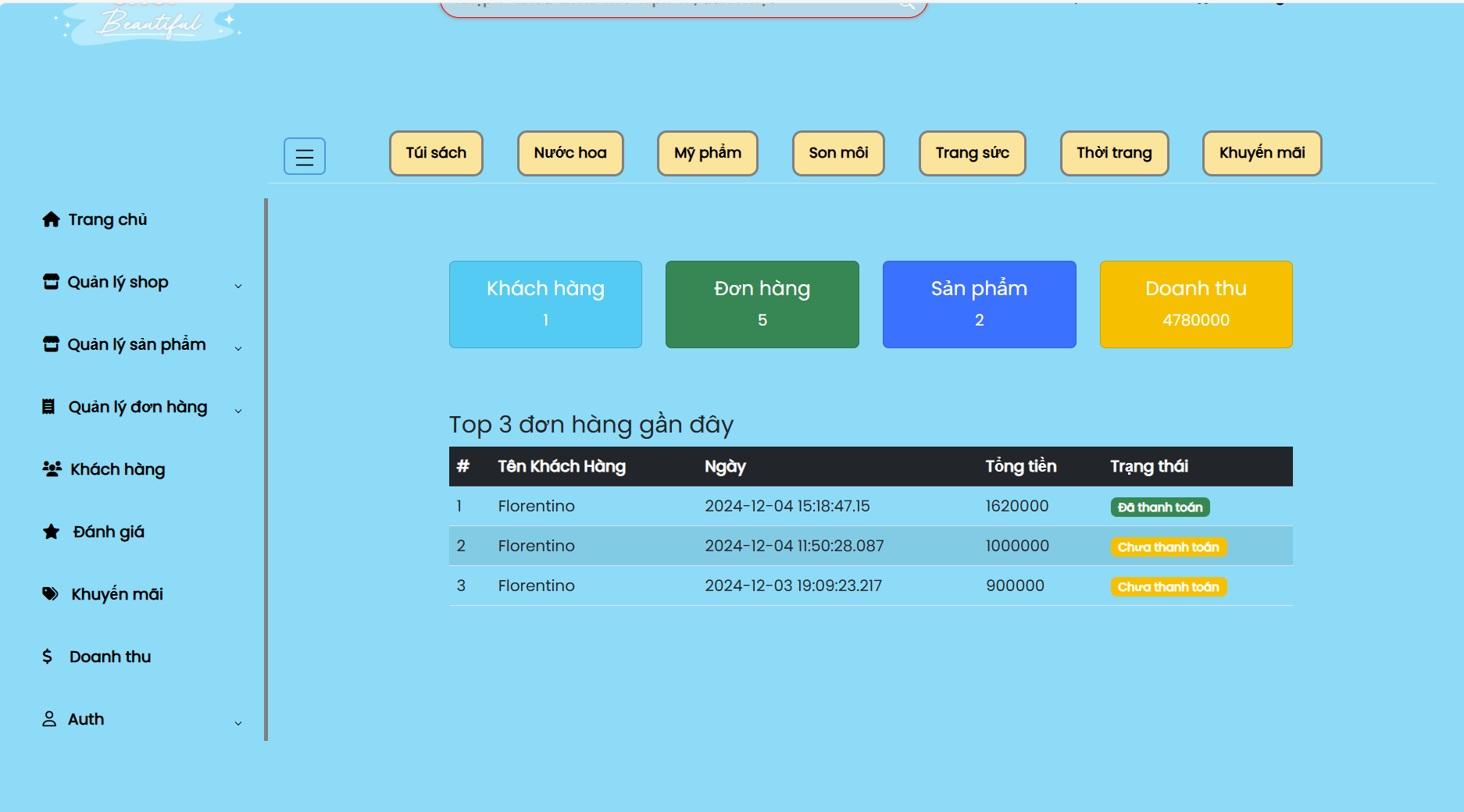
Giao diện của quản lí các doanh mục



Xem được thông tin những thông tin như mã hàng sản xuất, tên hãng sản xuất, mô tả. Và thêm xóa sửa, tìm kiếm các doanh mục.

**3.Giao diện của trang Vendor**

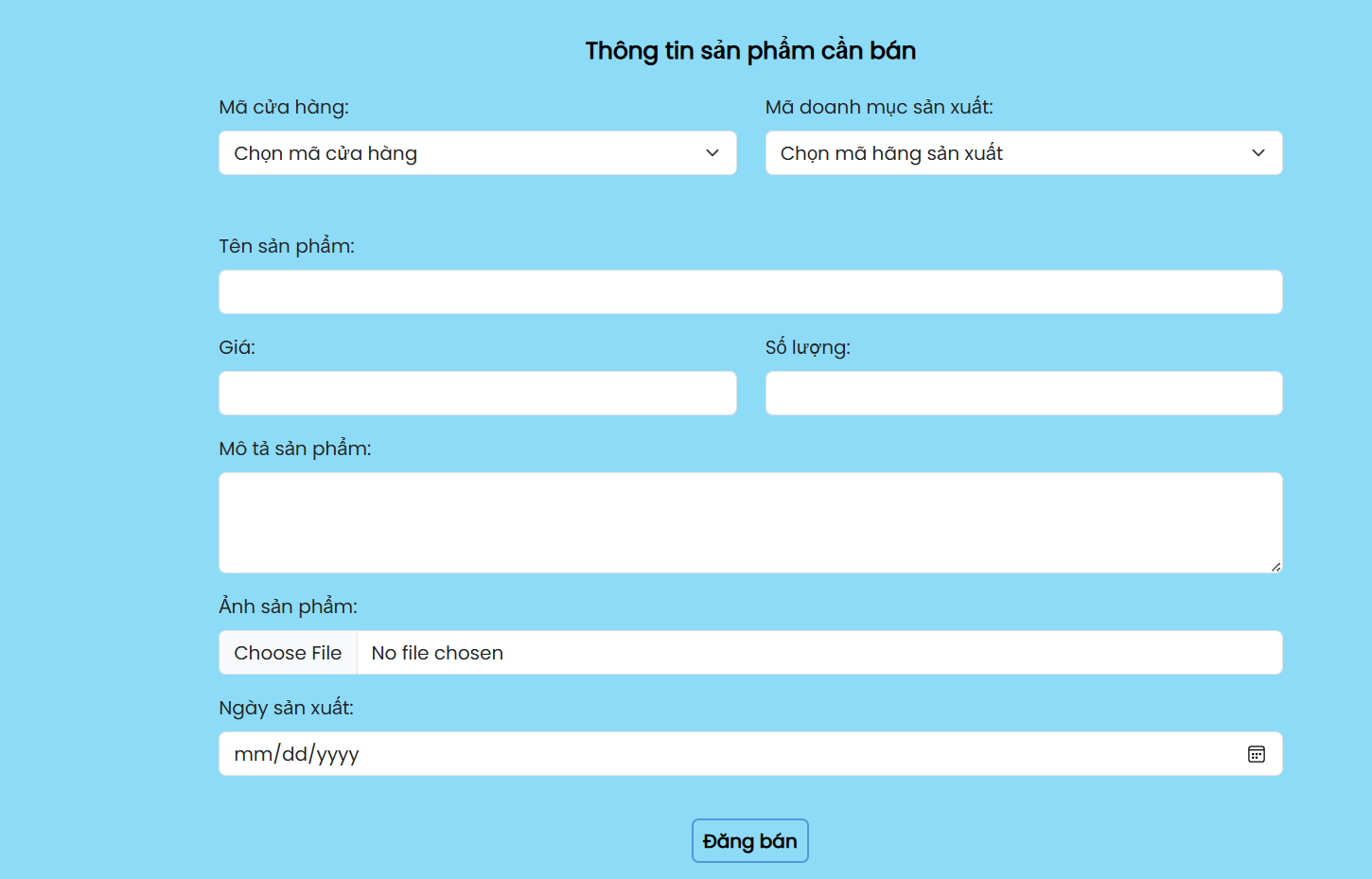
Trang chủ của Vendor

****

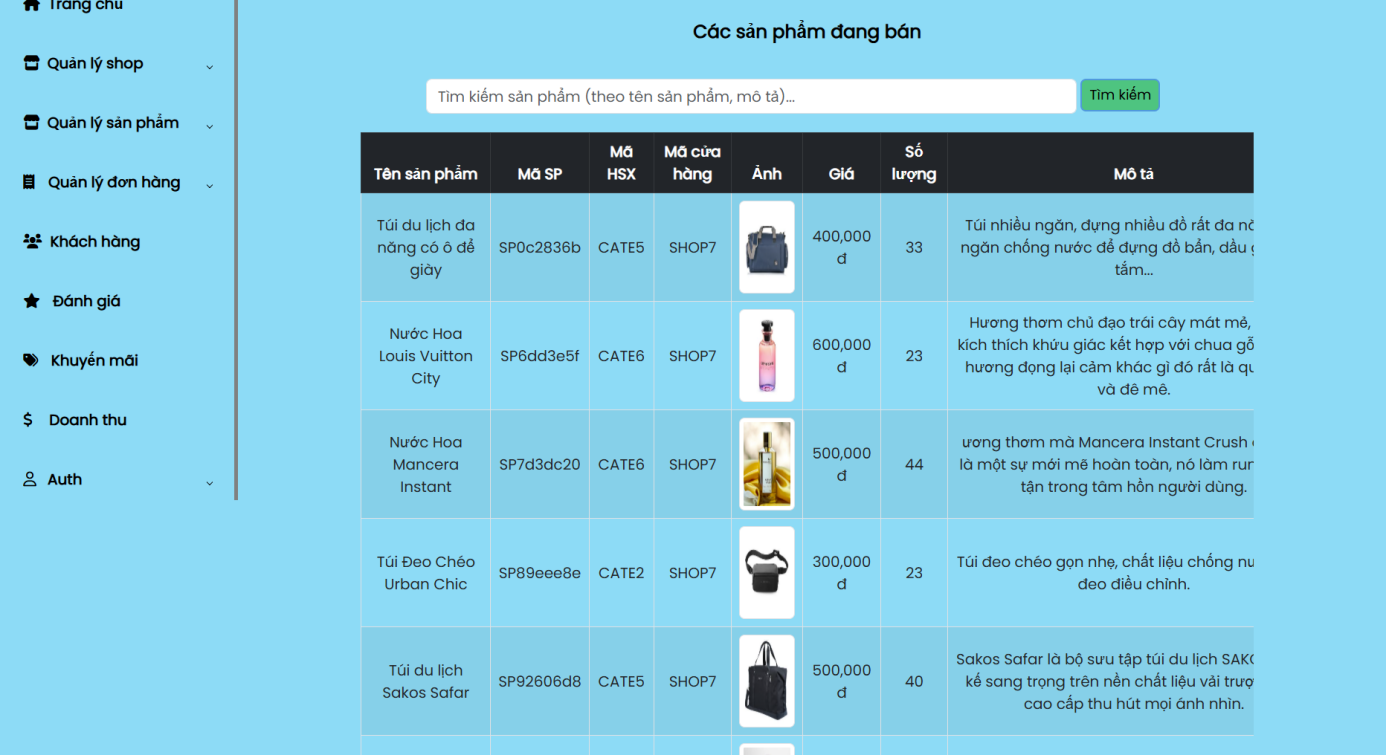
Sẽ có những thông tin như số lượng khách hàng của cửa hàng, đơn hàng, sản phẩm, và doanh thu của cửa hàng, và các đơn hàng gần đây

Quản lí các sản phẩm

Đăng bán sản phẩm



Danh sách các sản phẩm



Quản lí các đơn hàng.

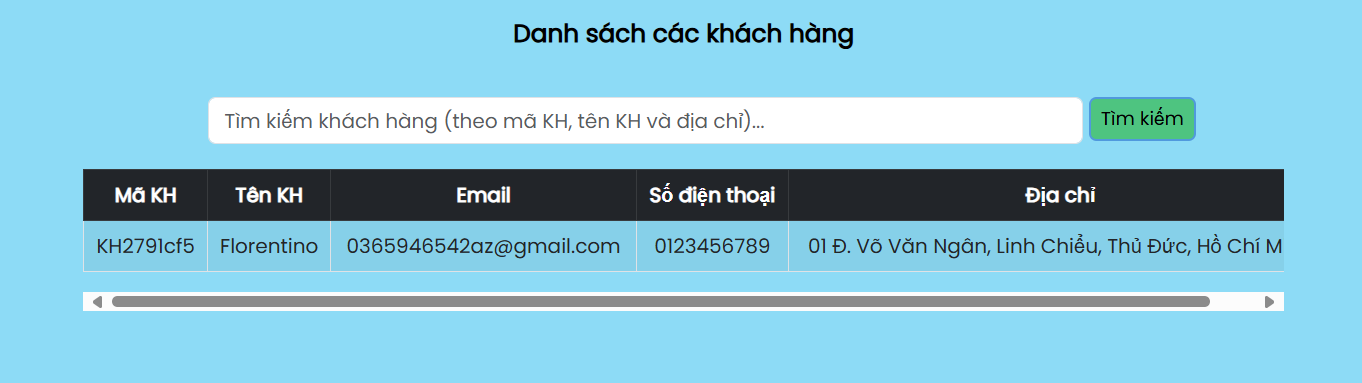
Chọn vào đơn hàng



Chi tiết đơn hàng



Quản lí các khách hàng



Đánh giá từ các khách hàng



Xem doanh thu

