**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

Em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Trầm Hoàng Nam – người đã tận tình hướng dẫn và chỉ bảo em trong suốt quá trình thực hiện đồ án.

Thầy không chỉ giúp em nắm vững kiến thức chuyên môn, mà còn truyền đạt những kinh nghiệm quý báu trong việc phân tích, thiết kế và triển khai hệ thống. Những chỉ dẫn tận tâm của thầy đã giúp em vượt qua nhiều khó khăn, hoàn thành tốt nhiệm vụ được giao và nâng cao kỹ năng làm việc thực tiễn.

Em cũng xin cảm ơn sự quan tâm, khích lệ của thầy trong suốt thời gian thực hiện đồ án. Chính sự hỗ trợ và hướng dẫn của thầy đã là nguồn động lực to lớn giúp em tự tin hơn và hoàn thành dự án này đúng tiến độ.

Một lần nữa, em xin gửi lời tri ân đến thầy và kính chúc thầy dồi dào sức khỏe, luôn thành công trong sự nghiệp giảng dạy và nghiên cứu.

**MỤC LỤC**

[LỜI CẢM ƠN 2](#_Toc18998)

[DANH MỤC HÌNH ẢNH 5](#_Toc27957)

[DANH MỤC BẢNG BIỂU 6](#_Toc32078)

[TÓM TẮT NIÊN LUẬN ĐỒ ÁN CHUYÊN NGÀNH 7](#_Toc26161)

[MỞ ĐẦU 9](#_Toc4786)

[CHƯƠNG 1: TỔNG QUAN 12](#_Toc32676)

[1.1 Tổng Quan Dự Án 12](#_Toc10656)

[1.2 Tổng quan các vấn đề cần giải quyết khi xây dựng website thương mại điện tử 12](#_Toc16548)

[CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT 14](#_Toc18020)

[2.1 Kiến Thức Cơ Bản Các Ngôn Ngữ, Thư Viện và Framework 14](#_Toc30639)

[2.1.1 Giới Thiệu Về HTML 14](#_Toc29529)

[2.1.2 Giới Thiệu Về CSS 14](#_Toc3112)

[2.1.3 Giới Thiệu Về JavaScript 15](#_Toc12038)

[2.1.4 Giới Thiệu Về Node.js 16](#_Toc3196)

[2.1.5 Giới Thiệu Về React.js 16](#_Toc14362)

[2.1.6 Giới thiệu về cơ sở dữ liệu Mysql 17](#_Toc26288)

[2.1.7 Giới thiệu về MUI (Material-UI) trong React.js 19](#_Toc3667)

[2.1.8 Giới thiệu về RESTful API với Node.js 21](#_Toc26154)

[CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU 23](#_Toc30613)

[3.1 Mô Tả Hệ Thống 23](#_Toc10480)

[3.2 Mô Tả Chức Năng 23](#_Toc4795)

[3.3 Mô Tả Phi Chức Năng 24](#_Toc4020)

[3.4 Thiết kế mô hình dữ liệu 25](#_Toc3559)

[3.4.1 Mô hình thực thể kết hợp 25](#_Toc1892)

[3.4.2 Mô hình vật lý 26](#_Toc1116)

[3.4.2 Mô tả các bảng trong lược đồ CSDL 26](#_Toc16409)

[3.4.3 Mô hình quan hệ 31](#_Toc26665)

[3.4.4 Thiết kế xử lý 31](#_Toc17557)

[CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU 34](#_Toc15918)

[4.1 Hình ảnh các chức năng 34](#_Toc16058)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 43](#_Toc26433)

[5.1 Kết luận 43](#_Toc22680)

[5.2 Hướng phát triển 43](#_Toc18301)

[DANH MỤC TÀI LIỆU THAM KHẢO 45](#_Toc26371)

[PHỤ LỤC 46](#_Toc25854)

**DANH MỤC HÌNH ẢNH**

[Hình 1 . Mô hình thực thể kết hợp 24](#_Toc13818)

[Hình 2 . Mô hình vậy lý 25](#_Toc1125)

[Hình 3 . Giao diện trang chủ 30](#_Toc31011)

[Hình 4 . Giao diện trang thông tin cá nhân 31](#_Toc7465)

[Hình 5 . Giao diện trang sản phẩm 31](#_Toc26573)

[Hình 6 . Giao diện trang lịch sử đơn hàng 31](#_Toc2047)

[Hình 7 . Giao diện trang quản lý đơn hàng 32](#_Toc10292)

[Hình 8 . Trang chủ 33](#_Toc2735)

[Hình 9 . Giao diện trang bài viết 33](#_Toc2319)

[Hình 10 . Giao diện chi tiết bài viết 34](#_Toc6471)

[Hình 11 . Giao diện trang hiển thị sản phẩm 34](#_Toc19314)

[Hình 12 . Giao diện trang giỏ hàng 35](#_Toc3867)

[Hình 13 . Giao diện trang sản phẩm yêu thích 35](#_Toc17395)

[Hình 14 . Giao diện quản lý bài viết cho người quản trị 36](#_Toc30987)

[Hình 15 . Giao diện quản lý thông tin sản phảm ở phía người quản trị 36](#_Toc5900)

[Hình 16 . Giao diện quản lý thêm sản phẩm cho người quản trị 37](#_Toc21764)

[Hình 17 . Giao diện quản lý hiển thị sản phẩm ưu tiên dành cho người quản trị 37](#_Toc5312)

[Hình 18 . Giao diện quản lý thông tin người dùng dành cho người quản trị viên 38](#_Toc27338)

[Hình 19 . Giao diện chức năng đăng nhập 38](#_Toc30593)

[Hình 20 . Giao diện đăng ký tài khoản của hệ thống 39](#_Toc8692)

[Hình 21 . Giao diện thanh toán bằng momo 39](#_Toc2642)

[Hình 22 . Giao diện người dùng nhập đầy đủ thông tin ngân hàng của mình 40](#_Toc31032)

[Hình 23 . Giao diện người dùng nhập mã xác nhận 40](#_Toc31273)

[Hình 24 .Giao diện thông báo khi người dùng thanh toán momo thành công 41](#_Toc744)

[Hình 25 . Giao diện thống kê doanh thu sản phẩm dành cho quản trị viên 41](#_Toc22167)

**DANH MỤC BẢNG BIỂU**

[Bảng 1 .Bảng mô tả các bảng trong lược đồ cơ sở dữ liệu 26](#_Toc24478)

[Bảng 2 . Lưu thông tin người dùng 27](#_Toc8183)

[Bảng 3 . Lưu thông tin sản phẩm 28](#_Toc32342)

[Bảng 4 . Danh mục sản phẩm 28](#_Toc5025)

[Bảng 5 . Lưu thông tin đơn hàng 28](#_Toc31001)

[Bảng 6 . Chi tiết đơn hàng 29](#_Toc13691)

[Bảng 7 .Quản lý khuyến mãi 29](#_Toc25582)

[Bảng 8 . Danh sách yêu thích 30](#_Toc27501)

[Bảng 9 . Giỏ hàng của người dùng 30](#_Toc4766)

[Bảng 10 . Quản lý bài viết 30](#_Toc23339)

**TÓM TẮT NIÊN LUẬN ĐỒ ÁN CHUYÊN NGÀNH**

**Vấn đề nghiên cứu**

Xây dựng website thương mại điện tử bán trang sức bằng Node.js.

**Các bước tiếp cận**

* **Nghiên cứu thị trường**: Phân tích các trang web bán trang sức tương tự để xác định các chức năng cơ bản như quản lý sản phẩm, giỏ hàng, thanh toán, tài khoản người dùng và tìm kiếm theo chất liệu, kiểu dáng, thương hiệu.
* **Lựa chọn công nghệ**: Sử dụng Node.js cho backend, MySQL làm cơ sở dữ liệu, React.js cho frontend, và tích hợp các công cụ thanh toán như Momo hoặc ZaloPay.
* **Thiết kế giao diện**: Sử dụng MUI và Bootstrap để xây dựng giao diện hiện đại, thân thiện với người dùng.
* **Phát triển chức năng**: Xây dựng các API RESTful để quản lý sản phẩm, giỏ hàng, thanh toán và tài khoản người dùng.
* **Kiểm thử và triển khai**: Đảm bảo hệ thống hoạt động ổn định, kiểm tra bảo mật và hiệu suất trước khi triển khai lên nền tảng lưu trữ.
* **Bảo trì**: Định kỳ nâng cấp giao diện và chức năng dựa trên phản hồi từ người dùng.

**Cách giải quyết vấn đề**

* Sử dụng Node.js và React.js để phát triển backend và frontend.
* Thiết kế giao diện hiện đại, tối ưu trải nghiệm người dùng bằng MUI và Bootstrap.
* Xây dựng các API RESTful để kết nối client và server.
* Kiểm tra, tối ưu hóa code và liên tục cải thiện giao diện, tính năng dựa trên phản hồi.
* Phân tích chức năng

Website bán trang sức sẽ cung cấp nền tảng mua sắm tiện lợi và an toàn, với các chức năng chính:

* **Quản lý sản phẩm**: Hỗ trợ thêm, sửa, xóa sản phẩm với thông tin chi tiết như chất liệu, kiểu dáng, kích cỡ, giá và hình ảnh.
* **Giỏ hàng**: Cho phép người dùng thêm, cập nhật và xóa sản phẩm.
* **Thanh toán**: Thu thập thông tin giao hàng và hỗ trợ nhiều phương thức thanh toán.
* **Tài khoản người dùng**: Đăng ký, đăng nhập, cập nhật thông tin cá nhân, lịch sử mua hàng.
* **Tìm kiếm và lọc sản phẩm**: Theo danh mục, chất liệu, thương hiệu, hoặc giá.

Backend được xây dựng với API RESTful để đảm bảo bảo mật và hiệu suất cao, frontend thiết kế thân thiện để tối ưu hóa trải nghiệm người dùng.

**Kết quả đạt được**

* Hoàn thiện website bán trang sức với giao diện thân thiện, hiện đại.
* Chức năng đầy đủ bao gồm quản lý sản phẩm, giỏ hàng, thanh toán, tài khoản người dùng và tìm kiếm sản phẩm.
* API RESTful linh hoạt, cơ sở dữ liệu tối ưu đảm bảo hiệu suất và khả năng mở rộng.
* Dự án giúp nâng cao kỹ năng sử dụng công nghệ Node.js, React.js, và MUI, MySQL, đồng thời hỗ trợ kinh doanh hiệu quả trong ngành trang sức.

**MỞ ĐẦU**

**Lý do chọn đề tài**

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, mua sắm trực tuyến đã trở thành xu hướng phổ biến và không thể thiếu trong đời sống của nhiều người. Ngành trang sức cũng không ngoại lệ, khi ngày càng nhiều khách hàng lựa chọn các trang web mua bán để tìm kiếm và sở hữu các sản phẩm trang sức chất lượng. Việc xây dựng một website bán trang sức không chỉ là cơ hội mở rộng thị trường mà còn là cách để các doanh nghiệp, cửa hàng tạo ra một kênh giao tiếp thuận tiện, tiết kiệm chi phí và nâng cao trải nghiệm cho khách hàng.

Mặc dù vậy, xây dựng một website bán trang sức không đơn giản chỉ là việc trưng bày sản phẩm. Các yếu tố như giao diện đẹp, dễ sử dụng, khả năng thanh toán an toàn, và trải nghiệm người dùng mượt mà đóng vai trò quan trọng trong sự thành công của một website thương mại điện tử. Đặc biệt, trong ngành trang sức, hình ảnh sản phẩm và khả năng tương tác giữa người bán và khách hàng phải được chú trọng. Vì vậy, việc xây dựng một website bán trang sức chất lượng là một vấn đề thú vị và đầy thử thách.

**Mục tiêu nghiên cứu**

Mục tiêu của nghiên cứu này là xây dựng một website thương mại điện tử chuyên về **trang sức**, áp dụng các công nghệ hiện đại như Node.js, React.js và MySQL để phát triển hệ thống hoàn chỉnh. Website sẽ cung cấp các tính năng như quản lý sản phẩm (thêm, sửa, xóa, tìm kiếm), giỏ hàng, thanh toán trực tuyến, danh sách yêu thích, và theo dõi đơn hàng. Ngoài ra, hệ thống sẽ đảm bảo bảo mật cao, tối ưu hóa hiệu suất và sử dụng API backend linh hoạt. Giao diện sẽ được thiết kế trực quan, hiện đại bằng Bootstrap để mang đến trải nghiệm mua sắm tối ưu. Dự án không chỉ tạo ra nền tảng thương mại điện tử chuyên nghiệp mà còn giúp nâng cao kỹ năng lập trình, áp dụng công nghệ mới và giải quyết các vấn đề thực tiễn trong ngành kinh doanh trang sức trực tuyến.

**Phương pháp nghiên cứu**

**Phương pháp nghiên cứu lý thuyết:**

* Nghiên cứu các tài liệu, bài viết, và hướng dẫn liên quan đến phát triển hệ thống thương mại điện tử, đặc biệt trong lĩnh vực bán trang sức.
* Tham khảo tài liệu từ các nguồn đáng tin cậy về Node.js, React.js, và MUI, MySQL để hiểu rõ cách áp dụng chúng vào hệ thống.
* Tìm hiểu các tiêu chuẩn về trải nghiệm người dùng (UX/UI) và cấu trúc API RESTful nhằm tối ưu hóa hiệu quả hoạt động của hệ thống.

**Phương pháp thực nghiệm:**

* Sử dụng Node.js để xây dựng backend, triển khai API RESTful để quản lý dữ liệu và kết nối giữa client và server.
* Phát triển giao diện người dùng với React.js và thư viện MUI, đảm bảo giao diện thân thiện và hiện đại.
* Sử dụng MySQL để xây dựng và quản lý cơ sở dữ liệu, tối ưu hóa các truy vấn nhằm đảm bảo hiệu suất cao.

**Đối tượng nghiên cứu**

**Đối tượng:**

* Các giải pháp công nghệ để phát triển một hệ thống thương mại điện tử bán trang sức với đầy đủ tính năng: quản lý sản phẩm, giỏ hàng, đặt hàng, thanh toán trực tuyến và danh sách yêu thích.
* Các công nghệ và công cụ hiện đại như Node.js, React.js, và MUI, MySQL trong việc xây dựng một hệ thống web hiệu quả.
* Quy trình và trải nghiệm người dùng (UX/UI) để tối ưu hóa việc mua sắm trực tuyến, đảm bảo giao diện thân thiện và dễ sử dụng.

**Phạm vi nghiên cứu**

Nghiên cứu tập trung vào việc xây dựng một website thương mại điện tử bán trang sức, bao gồm thiết kế, phát triển, kiểm thử và triển khai, với các chức năng đáp ứng nhu cầu mua sắm trực tuyến hiện đại.

# CHƯƠNG 1: TỔNG QUAN

## 1.1 Tổng Quan Dự Án

**Tên dự án**: Xây dựng website thương mại điện tử bán trang sức

Dự án “Xây dựng website thương mại điện tử bán trang sức ” nhằm tạo ra một nền tảng mua sắm trực tuyến chuyên nghiệp, hiện đại và thân thiện với người dùng. Trong bối cảnh thương mại điện tử ngày càng phát triển, website này sẽ giúp các doanh nghiệp kinh doanh trang sức tiếp cận khách hàng nhanh chóng, tối ưu hóa hoạt động kinh doanh và xây dựng thương hiệu một cách hiệu quả. Với Node.js làm nền tảng công nghệ, dự án hướng tới việc cung cấp một hệ thống mạnh mẽ, tốc độ cao và dễ dàng mở rộng, đáp ứng tốt các nhu cầu như quản lý sản phẩm, xử lý đơn hàng, và tích hợp thanh toán trực tuyến. Website cũng sẽ tập trung mang lại trải nghiệm mua sắm tối ưu thông qua giao diện bắt mắt, tính năng cá nhân hóa và khả năng tương thích trên mọi thiết bị. Đồng thời, hệ thống sẽ đảm bảo các tiêu chuẩn bảo mật cao nhất để bảo vệ thông tin khách hàng.

## 1.2 Tổng quan các vấn đề cần giải quyết khi xây dựng website thương mại điện tử

Nghiên cứu này tập trung vào việc xây dựng một website thương mại điện tử cho sản phẩm trang sức, nhằm giải quyết các vấn đề cốt lõi bao gồm:

* **Quản lý thông tin sản phẩm chi tiết**: Hiển thị đầy đủ các thông tin về chất liệu, kiểu dáng, giá cả, hình ảnh và mô tả sản phẩm.
* **Tạo giỏ hàng và quy trình thanh toán an toàn**: Hỗ trợ người dùng thêm sản phẩm vào giỏ, cập nhật số lượng, theo dõi đơn hàng, và thực hiện thanh toán qua các phương thức trực tuyến bảo mật.
* **Cung cấp hệ thống tài khoản người dùng tiện lợi**: Đăng ký, đăng nhập, quản lý thông tin cá nhân, và theo dõi lịch sử mua hàng.
* **Đảm bảo bảo mật và hiệu suất cao**: Sử dụng các biện pháp bảo mật dữ liệu như mã hóa thông tin và xác thực người dùng, đồng thời tối ưu hóa hệ thống để xử lý tốt lượng truy cập lớn.

Bên cạnh đó, dự án còn hướng tới việc cải thiện trải nghiệm người dùng thông qua:

* **Giao diện trực quan và hiện đại**: Sử dụng MUI và Bootstrap để thiết kế giao diện đẹp mắt, dễ sử dụng.
* **Tính năng cá nhân hóa**: Đề xuất sản phẩm dựa trên sở thích và hành vi của người dùng.

Với các mục tiêu trên, dự án không chỉ giải quyết các vấn đề kỹ thuật mà còn mang đến một giải pháp thực tiễn, phù hợp với nhu cầu ngày càng cao của thị trường thương mại điện tử trong lĩnh vực trang sức.

# CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

## 2.1 Kiến Thức Cơ Bản Các Ngôn Ngữ, Thư Viện và Framework

### Giới Thiệu Về HTML

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu tiêu chuẩn dùng để tạo và thiết kế cấu trúc của các trang web. HTML cung cấp các yếu tố và thẻ để xác định nội dung trên trang web, như văn bản, hình ảnh, liên kết, danh sách, bảng và nhiều thành phần khác. HTML giúp trình duyệt web hiểu cách hiển thị thông tin cho người dùng một cách hợp lý và dễ hiểu.

Một trang HTML cơ bản bao gồm các thành phần chính như:

- <html>: Thẻ bao quanh toàn bộ trang web.

- <head>: Chứa thông tin meta về trang web, chẳng hạn như tiêu đề trang và liên kết đến các tệp CSS hoặc JavaScript.

- <body>: Chứa nội dung chính mà người dùng nhìn thấy trên trang web.

- <title>: Đặt tiêu đề cho trang web, hiển thị trên thanh tiêu đề của trình duyệt.

HTML là ngôn ngữ cơ bản, không yêu cầu người dùng phải có kiến thức lập trình sâu rộng, nhưng rất quan trọng trong việc tạo dựng các trang web.

### Giới Thiệu Về CSS

CSS (Cascading Style Sheets) là một ngôn ngữ dùng để định dạng và thiết kế giao diện cho các trang web. CSS giúp tách biệt phần nội dung (HTML) và phần giao diện (style), từ đó tạo ra các trang web có giao diện đẹp mắt, dễ sử dụng và dễ bảo trì.

Các đặc điểm chính của CSS bao gồm:

* **Tính linh hoạt và tái sử dụng**: Có thể định nghĩa một kiểu cho nhiều phần tử trong trang web, giúp việc thay đổi và bảo trì trở nên dễ dàng hơn.
* **Đáp ứng nhiều thiết bị**: CSS hỗ trợ thiết kế đáp ứng (responsive design), giúp trang web hiển thị tốt trên mọi thiết bị từ máy tính để bàn đến điện thoại di động.
* **Hiệu quả và tối ưu**: CSS giúp cải thiện tốc độ tải trang vì nó tách biệt phần giao diện với phần nội dung HTML, và có thể được lưu trữ trong các tệp riêng biệt.

CSS có thể được áp dụng theo ba cách:

* **Inline CSS**: Áp dụng trực tiếp vào thẻ HTML.
* **Internal CSS**: Đặt các quy tắc CSS trong phần <style> của tài liệu HTML.
* **External CSS**: Đưa CSS vào một tệp riêng và liên kết tệp đó với trang HTML.

CSS là công cụ mạnh mẽ giúp tối ưu giao diện và trải nghiệm người dùng trên web. Tuy có một số hạn chế, nhưng việc sử dụng CSS hợp lý sẽ giúp cải thiện hiệu suất và dễ dàng bảo trì trang web. Vì vậy, CSS là yếu tố không thể thiếu trong phát triển web hiện đại.

### Giới Thiệu Về JavaScript

JavaScript (JS) là ngôn ngữ lập trình chủ yếu dùng để tạo trang web động và tương tác. JS chạy phía client, cho phép xử lý sự kiện, thay đổi nội dung trang mà không cần tải lại (AJAX) và kiểm tra tính hợp lệ của biểu mẫu. Nó tương tác với trang web qua DOM, hỗ trợ lập trình bất đồng bộ và tích hợp với các thư viện, framework như React, Angular, và Vue.js để xây dựng ứng dụng web phức tạp.

Một số đặc điểm chính của JavaScript:

* **Tính động**: JS có thể thay đổi nội dung trang web mà không cần tải lại trang.
* **Tương tác với người dùng**: JS có thể xử lý các sự kiện như click chuột, nhập liệu từ bàn phím, kéo thả, v.v.
* **Lập trình bất đồng bộ**: JS hỗ trợ các thao tác bất đồng bộ với APIs và hoạt động trên dữ liệu mà không làm gián đoạn trải nghiệm người dùng.
* **Dễ học và dễ sử dụng**: JS là một ngôn ngữ dễ tiếp cận, thích hợp cho người mới bắt đầu lập trình.

JavaScript là ngôn ngữ quan trọng giúp tạo ra các trang web động, tương tác và cải thiện trải nghiệm người dùng. Với khả năng xử lý sự kiện, thay đổi nội dung trang và tích hợp các framework, JS đóng vai trò thiết yếu trong phát triển web hiện đại.

### Giới Thiệu Về Node.js

Node.js là một môi trường chạy JavaScript trên server, được phát triển dựa trên V8 JavaScript Engine của Google Chrome. Nó cho phép các lập trình viên viết mã JavaScript bên ngoài trình duyệt, giúp tạo ra các ứng dụng máy chủ nhanh chóng, hiệu quả và dễ mở rộng.

Với Node.js, JavaScript không còn chỉ là ngôn ngữ lập trình dành riêng cho trình duyệt mà còn có thể sử dụng để phát triển các ứng dụng web phức tạp và các dịch vụ API. Đặc biệt, Node.js nổi bật với cơ chế **non-blocking I/O** (không đồng bộ), giúp xử lý hàng ngàn kết nối cùng lúc mà không làm gián đoạn tiến trình của hệ thống.

### Giới Thiệu Về React.js

React.js là một thư viện JavaScript được phát triển bởi Facebook, dùng để xây dựng giao diện người dùng (UI) cho các ứng dụng web. React.js giúp tạo ra các ứng dụng web có giao diện động, dễ quản lý và có khả năng mở rộng cao thông qua việc sử dụng các **component** (thành phần) có thể tái sử dụng.

#### Các đặc điểm chính của React.js:

* **Component-Based Architecture**: React chia giao diện thành các thành phần độc lập, có thể tái sử dụng, giúp việc xây dựng và quản lý ứng dụng trở nên dễ dàng hơn.
* **Virtual DOM**: React sử dụng Virtual DOM để tối ưu hóa việc cập nhật giao diện, giúp ứng dụng hoạt động nhanh và hiệu quả hơn.
* **One-Way Data Binding**: Dữ liệu trong React truyền theo một chiều, giúp dễ dàng kiểm soát luồng dữ liệu và giảm lỗi trong ứng dụng.
* **JSX (JavaScript XML)**: Cung cấp cú pháp giống HTML để viết các component, giúp code dễ đọc và dễ bảo trì.
* **Tương tác với Backend**: React.js có thể dễ dàng kết hợp với các RESTful API hoặc GraphQL để xây dựng ứng dụng full-stack.
* **Hỗ trợ phát triển ứng dụng di động**: React Native, dựa trên React.js, được sử dụng để phát triển ứng dụng di động trên cả iOS và Android.

#### Lợi ích của việc sử dụng React.js:

* **Hiệu suất cao**: Với Virtual DOM, React cập nhật và hiển thị giao diện nhanh hơn so với các phương pháp DOM truyền thống.
* **Tái sử dụng code**: Các component trong React giúp giảm thiểu việc lặp lại code, tiết kiệm thời gian phát triển.
* **Hỗ trợ cộng đồng mạnh mẽ**: React.js có một cộng đồng lớn, tài liệu phong phú và hỗ trợ từ Facebook, giúp dễ dàng tiếp cận và giải quyết các vấn đề phát sinh trong quá trình phát triển.

#### Ứng dụng của React.js trong website thương mại điện tử bán trang sức:

* **Hiển thị sản phẩm**: Tạo các component hiển thị danh sách trang sức, bộ lọc sản phẩm theo danh mục, giá cả, và thương hiệu.
* **Giỏ hàng và thanh toán**: Xây dựng các component giỏ hàng động, hiển thị sản phẩm được thêm, và xử lý quy trình thanh toán.
* **Cá nhân hóa giao diện**: React cho phép tạo các trang cá nhân hóa như danh sách yêu thích, lịch sử đơn hàng.
* **Tối ưu trải nghiệm người dùng**: Giao diện phản hồi nhanh, trực quan và thân thiện trên mọi thiết bị, từ máy tính để bàn đến điện thoại di động.

React.js đóng vai trò quan trọng trong việc xây dựng giao diện người dùng hiện đại, dễ sử dụng, giúp website bán trang sức mang lại trải nghiệm mua sắm trực tuyến tối ưu.

### Giới thiệu về cơ sở dữ liệu Mysql

MySQL là một hệ quản trị cơ sở dữ liệu (DBMS - Database Management System) mã nguồn mở, sử dụng SQL (Structured Query Language) để thao tác và truy vấn dữ liệu. MySQL rất phổ biến trong việc xây dựng các ứng dụng web nhờ vào tính hiệu suất cao, dễ sử dụng và khả năng tích hợp tốt với các công nghệ phát triển web.

Lịch sử và Phát triển của MySQL: MySQL được phát triển bởi MySQL AB, một công ty phần mềm Thụy Điển, và sau này được Sun Microsystems mua lại vào năm 2008. Vào năm 2010, Oracle Corporation đã mua lại Sun Microsystems, và MySQL trở thành một phần của Oracle. Dù vậy, MySQL vẫn giữ được tính mã nguồn mở và được duy trì bởi cộng đồng.

**Các Đặc Trưng và Điểm Mạnh của MySQL:**

**Mã nguồn mở:** MySQL được phát triển và duy trì dưới giấy phép mã nguồn mở (GPL), giúp người dùng có thể tự do sử dụng, sửa đổi và phân phối phần mềm mà không phải trả phí.

**Hiệu suất cao:** MySQL là một trong những hệ quản trị cơ sở dữ liệu có hiệu suất cao, tối ưu hóa cho các truy vấn phức tạp và có thể xử lý hàng triệu bản ghi trong một thời gian ngắn.

**Tính mở rộng:** MySQL có khả năng mở rộng dễ dàng, từ các ứng dụng nhỏ đến các hệ thống lớn với hàng triệu người dùng đồng thời.

**Tính tương thích và tích hợp:** MySQL hỗ trợ rất nhiều hệ điều hành, bao gồm Linux, Windows, macOS và các hệ điều hành khác. Nó cũng tương thích tốt với nhiều ngôn ngữ lập trình như PHP, Java, Python, và Node.js, khiến nó trở thành một lựa chọn phổ biến trong các ứng dụng web.

**Hệ thống quản lý người dùng và bảo mật:** MySQL cung cấp các tính năng bảo mật mạnh mẽ như xác thực người dùng, phân quyền truy cập vào các bảng và cơ sở dữ liệu, và mã hóa dữ liệu.

**Hỗ trợ các loại dữ liệu phong phú:** MySQL hỗ trợ nhiều loại dữ liệu như các kiểu số, văn bản, ngày giờ, dữ liệu nhị phân, v.v., cho phép người phát triển ứng dụng quản lý dữ liệu một cách linh hoạt.

**Cấu trúc cơ sở dữ liệu MySQL:**

**Database:** MySQL lưu trữ dữ liệu trong các cơ sở dữ liệu. Một cơ sở dữ liệu có thể chứa nhiều bảng dữ liệu.

**Table:** Bảng là một cấu trúc dữ liệu dùng để lưu trữ dữ liệu dạng hàng và cột. Mỗi bảng chứa các trường (columns) và mỗi bản ghi (row) là một đối tượng dữ liệu.

**Column:** Cột trong bảng, đại diện cho các thuộc tính của đối tượng mà bảng đang lưu trữ.

**Row:** Dòng trong bảng, đại diện cho một bản ghi duy nhất với đầy đủ các giá trị cho các cột.

**Primary Key:** Là khóa chính trong bảng, dùng để xác định duy nhất mỗi bản ghi.

**Foreign Key:** Là khóa ngoại, dùng để thiết lập mối quan hệ giữa các bảng.

**Index:** Cung cấp một cách nhanh chóng để tìm kiếm dữ liệu trong bảng.

MySQL là một hệ quản trị cơ sở dữ liệu mạnh mẽ và linh hoạt, được sử dụng rộng rãi trong phát triển ứng dụng web. Với các tính năng như mã nguồn mở, hiệu suất cao, hỗ trợ ACID và khả năng mở rộng, MySQL là một công cụ tuyệt vời cho việc lưu trữ và quản lý dữ liệu trong các ứng dụng web, đặc biệt là khi kết hợp với các công nghệ như Node.js.

### Giới thiệu về MUI (Material-UI) trong React.js

MUI (Material-UI) là một thư viện giao diện người dùng (UI) mạnh mẽ và phổ biến cho React, cung cấp các thành phần giao diện được thiết kế theo nguyên lý Material Design của Google. MUI giúp xây dựng các ứng dụng React với giao diện đẹp mắt, dễ sử dụng và dễ dàng tùy chỉnh. MUI là một lựa chọn lý tưởng khi bạn cần tạo ra các giao diện người dùng hiện đại, đồng nhất và responsive.

MUI cung cấp một bộ sưu tập phong phú các thành phần UI như buttons, dialogs, inputs, tables, forms, icons, grids, và nhiều thành phần khác, giúp bạn phát triển các ứng dụng nhanh chóng mà không cần phải tự xây dựng từ đầu.

**Các Tính Năng Chính của MUI:**

**Tính nhất quán trong thiết kế:** MUI cung cấp một bộ thành phần UI đồng nhất, giúp tạo ra các ứng dụng có giao diện người dùng thống nhất mà không phải lo lắng về việc phối hợp các thành phần khác nhau.

**Responsive Design:** Các thành phần của MUI được thiết kế để tương thích với mọi kích thước màn hình, giúp bạn xây dựng các ứng dụng web hoặc mobile-friendly mà không cần phải lo lắng về việc tối ưu hóa cho các thiết bị khác nhau.

**Tùy chỉnh linh hoạt:** MUI cho phép tùy chỉnh rất sâu, từ việc thay đổi màu sắc, phông chữ, kích thước thành phần cho đến việc thay đổi các hành vi của các thành phần UI để phù hợp với yêu cầu thiết kế của bạn.

**Hỗ trợ theme:** MUI có hệ thống theme mạnh mẽ cho phép bạn định nghĩa màu sắc, typography, spacing và các giá trị khác của toàn bộ ứng dụng từ một nơi duy nhất, giúp quản lý và thay đổi giao diện dễ dàng.

**Accessibility:** MUI được xây dựng với sự chú trọng đến khả năng tiếp cận (accessibility), đảm bảo rằng các thành phần UI hỗ trợ các tính năng như bàn phím và các tiêu chuẩn WCAG (Web Content Accessibility Guidelines).

**Tích hợp dễ dàng với các công cụ khác:** MUI dễ dàng tích hợp với các thư viện và công cụ khác trong hệ sinh thái React, chẳng hạn như React Router, Redux, và Form libraries như React Hook Form.

**Các Thành Phần Chính của MUI:**

MUI cung cấp một bộ sưu tập phong phú các thành phần UI mà bạn có thể sử dụng để xây dựng giao diện người dùng cho ứng dụng React. Một số thành phần phổ biến bao gồm:

**Button:** Thành phần để tạo các nút trong ứng dụng.

**Typography:** Thành phần giúp định dạng và hiển thị các đoạn văn bản.

**Grid:** Thành phần giúp xây dựng layout grid để bố trí các thành phần UI theo dạng lưới.

**TextField:** Thành phần để tạo các ô nhập liệu (input fields).

**Dialog:** Thành phần hiển thị hộp thoại (modals) cho các thông báo hoặc các hoạt động người dùng.

**AppBar:** Thành phần để tạo thanh điều hướng (navbar).

Card: Thành phần để hiển thị các phần tử thông tin trong dạng thẻ (card).

**Snackbar:** Thành phần để hiển thị các thông báo tạm thời cho người dùng (ví dụ: thông báo thành công, lỗi).

MUI là một thư viện mạnh mẽ, linh hoạt và dễ sử dụng cho React, giúp xây dựng các giao diện người dùng đẹp mắt, đồng nhất và có khả năng tùy chỉnh cao. Với MUI, có thể nhanh chóng tạo ra các ứng dụng web với giao diện hiện đại và khả năng tương thích tốt với các thiết bị khác nhau, đồng thời dễ dàng quản lý và mở rộng giao diện người dùng.[9]

### Giới thiệu về RESTful API với Node.js

RESTful API là một kiểu kiến trúc (architecture style) để xây dựng các dịch vụ web. REST (Representational State Transfer) sử dụng các phương thức HTTP chuẩn như GET, POST, PUT, DELETE để tương tác với các tài nguyên (resources) và thực hiện các thao tác trên các tài nguyên đó. Các tài nguyên này có thể là dữ liệu trong cơ sở dữ liệu hoặc bất kỳ thứ gì có thể được biểu diễn dưới dạng JSON hoặc XML.

Node.js là một môi trường chạy JavaScript trên phía server, giúp phát triển các ứng dụng web nhanh chóng và mạnh mẽ. Node.js rất phù hợp để xây dựng các RESTful API nhờ khả năng xử lý song song tốt và cấu trúc đơn giản. Kết hợp Node.js với Express (một framework nhẹ nhàng dành cho Node.js) giúp việc xây dựng các API trở nên dễ dàng và hiệu quả.

**Kiến Trúc RESTful API:**

Trong kiến trúc RESTful, tài nguyên được xác định bởi các URL (Uniform Resource Locator), và các phương thức HTTP sẽ thao tác với các tài nguyên này. Các phương thức HTTP phổ biến trong RESTful API là:

**GET:** Lấy dữ liệu từ server (không thay đổi trạng thái).

**POST:** Tạo một tài nguyên mới trên server.

**PUT:** Cập nhật một tài nguyên hiện có.

**DELETE:** Xóa một tài nguyên.

**PATCH:** Cập nhật một phần của tài nguyên.

Mỗi tài nguyên trong RESTful API thường sẽ có một URL riêng biệt. Ví dụ:

**GET /users:** Lấy danh sách người dùng.

**POST /users:** Tạo người dùng mới.

**GET /users/:id:** Lấy thông tin người dùng theo ID.

**PUT /users/:id:** Cập nhật thông tin người dùng theo ID.

**DELETE /users/:id:** Xóa người dùng theo ID.

Việc xây dựng một RESTful API bằng Node.js và Express là một quá trình đơn giản nhưng mạnh mẽ. Bạn có thể dễ dàng mở rộng API của mình, kết nối với cơ sở dữ liệu (MySQL, MongoDB, v.v.), xử lý xác thực người dùng, và triển khai API cho các ứng dụng frontend. Với Node.js và Express, bạn có thể nhanh chóng phát triển một API mạnh mẽ và hiệu quả cho các ứng dụng web.

# CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

(Mô tả các bước nghiên cứu đã tiến hành, các bản thiết kế, cách thức cài đặt chương trình hoặc hiện thực hóa nghiên cứu. Đối với các đề tài ứng dụng có kết quả là sản phẩm phần mềm phải có hồ sơ thiết kế, cài đặt, ... theo các dạng lược đồ, mô hình phổ biến trong ngành. Nội dung đặc tả nhu cầu, phân tích thiết kế hệ thống cũng thể hiện trong chương này.)

## 3.1 Mô Tả Hệ Thống

* **Website thương mại điện tử trang sức** là một nền tảng trực tuyến cho phép người dùng tìm kiếm, xem, và mua sắm các sản phẩm trang sức như nhẫn, vòng tay, lắc tay từ nhiều thương hiệu khác nhau. Hệ thống cung cấp trải nghiệm mua sắm thân thiện, nhanh chóng và tiện lợi với các tính năng như tìm kiếm, lọc sản phẩm, quản lý giỏ hàng, và thanh toán trực tuyến.
* **Quản trị viên**: Có các quyền quản lý sản phẩm, đơn hàng, người dùng, bình luận, đánh giá, các chương trình khuyến mãi, thống kê báo cáo doanh thu và tồn kho. Ngoài ra, có thể giao tiếp với khách hàng qua khung chat.
* **Khách hàng**: Bao gồm cả khách vãng lai và khách hàng đã đăng ký. Khách hàng đã đăng ký có thể quản lý giỏ hàng, theo dõi đơn hàng, quản lý tài khoản, thêm sản phẩm yêu thích, và đánh giá, bình luận sản phẩm đã mua.
* **Khách vãng lai**: Có thể xem thông tin sản phẩm, tìm kiếm sản phẩm, liên hệ với quản trị viên qua chat, và đăng ký tài khoản để trở thành khách hàng chính thức.

## 3.2 Mô Tả Chức Năng

**Người Dùng**

* **Xem sản phẩm**: Duyệt qua các sản phẩm trên trang chủ hoặc danh mục, hiển thị tên, giá, hình ảnh, mô tả ngắn, chất liệu, kích thước và trạng thái còn hàng.
* **Tìm kiếm và lọc sản phẩm**: Tìm kiếm theo từ khóa hoặc lọc theo tiêu chí như chất liệu, giá cả, kiểu dáng, trọng lượng, và kích cỡ.
* **Xem chi tiết sản phẩm**: Xem đầy đủ thông tin về sản phẩm, bao gồm hình ảnh, giá, chất liệu, kích thước, trọng lượng, và đánh giá từ khách hàng.
* **Thêm vào giỏ hàng**: Thêm sản phẩm vào giỏ hàng và điều chỉnh số lượng mong muốn.
* **Quản lý giỏ hàng**: Cập nhật số lượng, xóa sản phẩm khỏi giỏ.
* **Yêu thích sản phẩm**: Thêm sản phẩm vào danh sách yêu thích để dễ dàng theo dõi.
* **Đặt hàng**: Điền thông tin đặt hàng, chọn phương thức thanh toán, và nhận thông báo đơn hàng qua email.
* **Theo dõi đơn hàng**: Xem trạng thái đơn hàng và lịch sử mua hàng.
* **Đánh giá và bình luận**: Đánh giá sản phẩm và viết bình luận.
* **Quản lý tài khoản**: Cập nhật thông tin cá nhân, thay đổi mật khẩu, và khôi phục tài khoản qua OTP email.
* **Chat trực tuyến**: Giao tiếp với quản trị viên qua khung chat tích hợp.

**Quản Trị Viên**

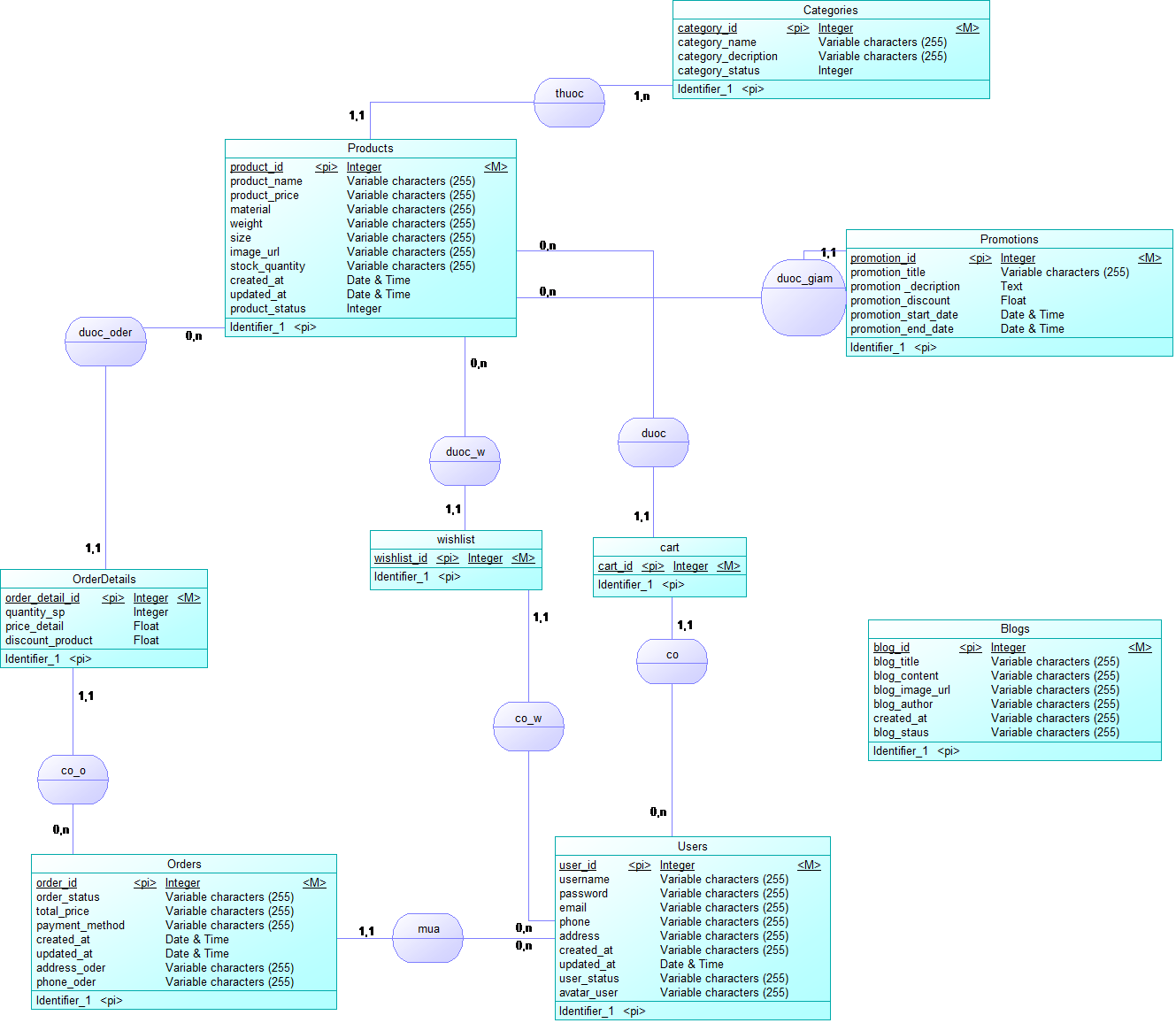
* **Quản lý sản phẩm**: Thêm, sửa, xóa sản phẩm cùng với các thông tin chi tiết.
* **Quản lý danh mục và thương hiệu**: Thêm, sửa, xóa danh mục sản phẩm.
* **Quản lý đơn hàng**: Xem danh sách đơn hàng và cập nhật trạng thái (đang xử lý, đã giao, hủy).
* **Quản lý người dùng**: Quản lý danh sách người dùng, kích hoạt hoặc khóa tài khoản.
* **Thống kê và báo cáo**: Báo cáo doanh thu, sản phẩm bán chạy, và tồn kho.
* **Quản lý khuyến mãi**: Tạo và quản lý các chương trình ưu đãi.
* **Quản lý đánh giá và bình luận**: Duyệt, ẩn hoặc xóa nội dung không phù hợp.
* **Quản lý giao diện**: Thay đổi nội dung trang chủ và banner khuyến mãi.
* **Hỗ trợ khách hàng**: Giải đáp thắc mắc qua chat hoặc email.

## 3.3 Mô Tả Phi Chức Năng

* **Bảo mật**: Mã hóa thông tin cá nhân và giao dịch, bảo vệ hệ thống khỏi các tấn công như SQL Injection và XSS.
* **Khả năng mở rộng**: Hệ thống hỗ trợ tăng lượng người dùng và dữ liệu mà không ảnh hưởng hiệu suất.
* **Khả năng duy trì**: Dễ nâng cấp, sửa lỗi, và thêm tính năng mới.
* **Tương thích**: Hoạt động mượt mà trên các trình duyệt (Chrome, Firefox, Edge) và thiết bị di động.
* **Trải nghiệm người dùng**: Giao diện thân thiện, dễ sử dụng, tối ưu cho thiết bị di động và máy tính.
* **Tích hợp**: Hỗ trợ các cổng thanh toán như Momo, ZaloPay, và dịch vụ bên thứ ba như email hoặc chatbot​​

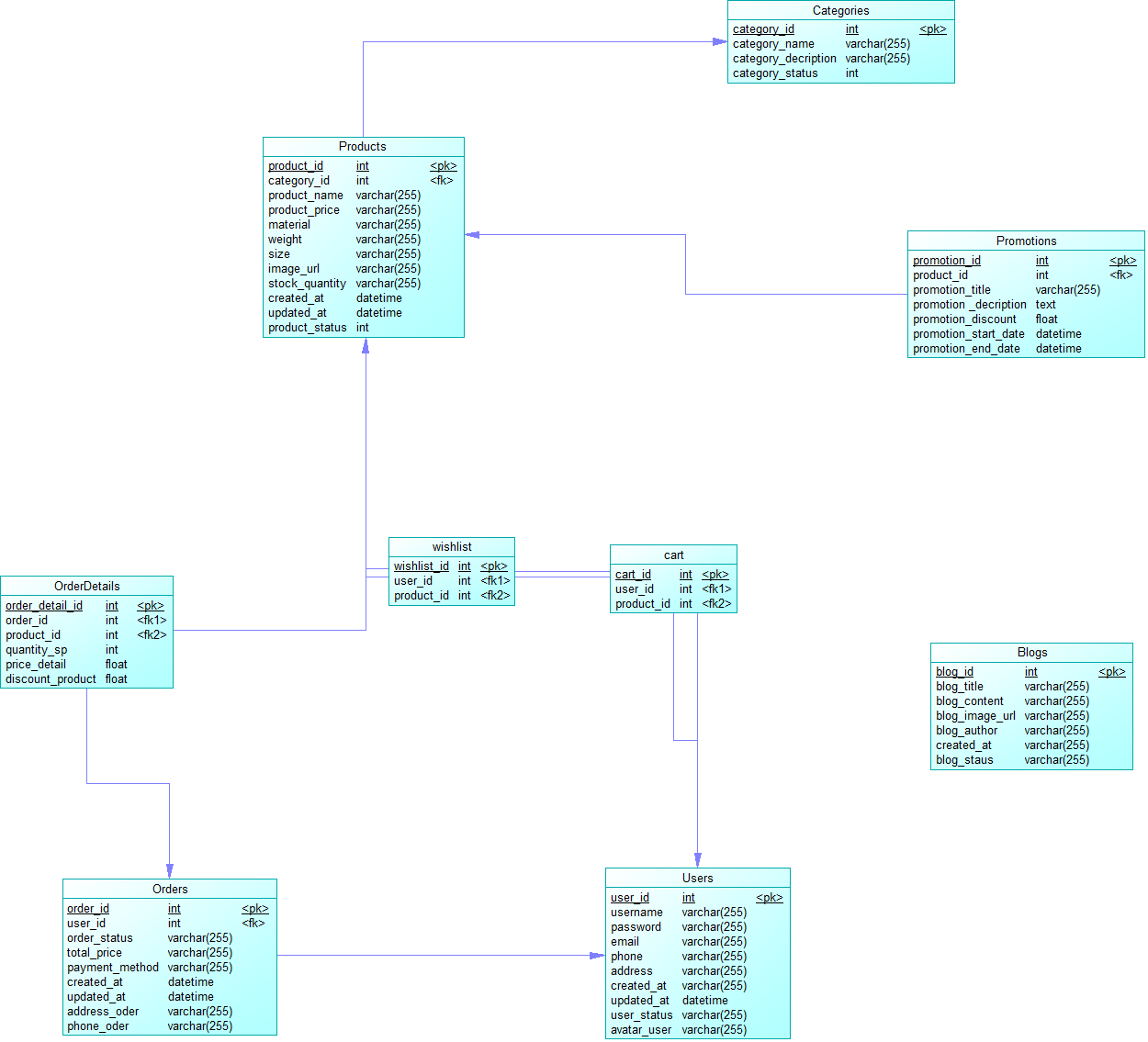
## 3.4 Thiết kế mô hình dữ liệu

### **3.4.1 Mô hình thực thể kết hợp**

****

**Hình 1**. Mô hình thực thể kết hợp

### 3.4.2 Mô hình vật lý



**Hình 2****.** Mô hình vậy lý

### 3.4.2 Mô tả các bảng trong lược đồ CSDL

|  |  |
| --- | --- |
| **Tên bảng** | **Diễn giải** |
| USER | Người dùng |
| PRODUCT | Sản phẩm |
| CATEGORIES | Danh mục sản phẩm |
| ORDERS | Đơn hàng |
| ORDERDETAILS | Chi tiết đơn hàng |
| PROMOTIONS | Khuyến mãi |
| WISHLIST | Danh sách yêu thích |
| CART | Giỏ hàng |
| BLOGS | Bài viết |

**Bảng 1****.**Bảng mô tả các bảng trong lược đồ cơ sở dữ liệu

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| 1 | user\_id | Mã người dùng | Bắt buộc | Int | Khóa chính |  |  |
| 2 | username | Tên đăng nhập | Bắt buộc | Varchar |  | 255 |  |
| 3 | password | Mật khẩu | Bắt buộc | Varchar |  | 255 |  |
| 4 | email | Email | Bắt buộc | Varchar |  | 255 |  |
| 5 | phone | Số điện thoại |  | Varchar |  | 255 |  |
| 6 | address | Đại chỉ |  | Varchar |  | 255 |  |
| 7 | created\_at | Ngày tạo | Bắt buộc | DateTime |  |  |  |
| 8 | updated\_at | Ngày cập nhật |  | DateTime |  |  |  |
| 9 | user\_status | Trạng thái người dùng |  | Int |  |  |  |
| 10 | avatar\_user | Ảnh đại diện người dùng |  | Varchar |  | 255 |  |

**Bảng 2****.** Lưu thông tin người dùng

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| 1 | product\_id | Mã sản phẩm | Bắt buộc | Int | Khóa chính |  |  |
| 2 | category\_id | Mã danh mục | Bắt buộc | Int | Khóa ngoại |  |  |
| 3 | product\_name | Tên sản phẩm | Bắt buộc | Varchar |  | 255 |  |
| 4 | product\_price | Giá sản phẩm | Bắt buộc | Float |  |  |  |
| 5 | material | Chất liệu |  | Varchar |  | 255 |  |
| 6 | weight | Trọng lượng |  | Float |  |  |  |
| 7 | size | Kích thước |  | Varchar |  | 255 |  |
| 8 | image\_url | URL hình ảnh |  | Varchar |  | 255 |  |
| 9 | stock\_quantity | Số lượng tồn kho |  | Int |  |  |  |
| 10 | created\_at | Ngày tạo | Bắt buộc | DateTime |  |  |  |
| 11 | updated\_at | Ngày cập nhật |  | DateTime |  |  |  |
| 12 | product\_status | Trạng thái sản phẩm |  | Int |  |  |  |

**Bảng 3****.** Lưu thông tin sản phẩm

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| 1 | category\_id | Mã danh mục | Bắt buộc | Int | Khóa chính |  |  |
| 2 | category\_name | Tên danh mục | Bắt buộc | Varchar |  | 255 |  |
| 3 | category\_description | Mô tả danh mục |  | Varchar |  | 255 |  |
| 4 | category\_status | Trạng thái danh mục |  | Int |  |  |  |

**Bảng 4****.** Danh mục sản phẩm

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| 1 | order\_id | Mã đơn hàng | Bắt buộc | Int | Khóa chính |  |  |
| 2 | user\_id | Mã người dùng | Bắt buộc | Int | Khóa ngoại |  |  |
| 3 | order\_status | Trạng thái đơn hàng |  | Varchar |  | 255 |  |
| 4 | total\_price | Tổng giá trị | Bắt buộc | Float |  |  |  |
| 5 | payment\_method | Phương thức thanh toán |  | Varchar |  | 255 |  |
| 6 | created\_at | Ngày tạo | Bắt buộc | DateTime |  |  |  |
| 7 | updated\_at | Ngày cập nhật |  | DateTime |  |  |  |
| 8 | address\_order | Địa chỉ đơn hàng |  | Varchar |  | 255 |  |
| 9 | phone\_order | Số điện thoại |  | Varchar |  | 255 |  |

**Bảng 5****.** Lưu thông tin đơn hàng

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| 1 | order\_detail\_id | Mã chi tiết đơn hàng | Bắt buộc | Int | Khóa chính |  |  |
| 2 | order\_id | Mã đơn hàng | Bắt buộc | Int | Khóa ngoại |  |  |
| 3 | product\_id | Mã sản phẩm | Bắt buộc | Int | Khóa ngoại |  |  |
| 4 | quantity\_sp | Số lượng sản phẩm |  | Int |  |  |  |
| 5 | price\_detail | Giá chi tiết |  | Float |  |  |  |
| 6 | discount\_product | Chiết khấu |  | Float |  |  |  |

**Bảng 6****.** Chi tiết đơn hàng

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| 1 | promotion\_id | Mã khuyến mãi | Bắt buộc | Int | Khóa chính |  |  |
| 2 | product\_id | Mã sản phẩm | Bắt buộc | Int | Khóa ngoại |  |  |
| 3 | promotion\_title | Tiêu đề khuyến mãi | Bắt buộc | Varchar |  | 255 |  |
| 4 | promotion\_description | Mô tả khuyến mãi |  | Varchar |  | 255 |  |
| 5 | promotion\_discount | Giảm giá |  | Float |  |  |  |
| 6 | promotion\_start\_date | Ngày bắt đầu |  | DateTime |  |  |  |
| 7 | promotion\_end\_date | Ngày kết thúc |  | DateTime |  |  |  |

**Bảng 7****.**Quản lý khuyến mãi

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| 1 | wishlist\_id | Mã danh sách yêu thích | Bắt buộc | Int | Khóa chính |  |  |
| 2 | user\_id | Mã người dùng | Bắt buộc | Int | Khóa ngoại |  |  |
| 3 | product\_id | Mã sản phẩm | Bắt buộc | Int | Khóa ngoại |  |  |

**Bảng 8****.** Danh sách yêu thích

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| 1 | cart\_id | Mã giỏ hàng | Bắt buộc | Int | Khóa chính |  |  |
| 2 | user\_id | Mã người dùng | Bắt buộc | Int | Khóa ngoại |  |  |
| 3 | product\_id | Mã sản phẩm | Bắt buộc | Int | Khóa ngoại |  |  |

**Bảng 9****.** Giỏ hàng của người dùng

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| 1 | blog\_id | Mã bài viết | Bắt buộc | Int | Khóa chính |  |  |
| 2 | blog\_title | Tiêu đề bài viết |  | Varchar |  | 255 |  |
| 3 | blog\_content | Nội dung bài viết |  | Longtext |  |  |  |
| 4 | blog\_image\_url | Hình ảnh bài viết |  | Varchar |  | 255 |  |
| 5 | blog\_author | Tác giả bài viết |  | Varchar |  | 255 |  |
| 6 | created\_at | Ngày tạo bài viết | Bắt buộc | DateTime |  |  |  |
| 7 | blog\_status | Trạng thái bài viết |  | Int |  |  |  |

**Bảng 10****.** Quản lý bài viết

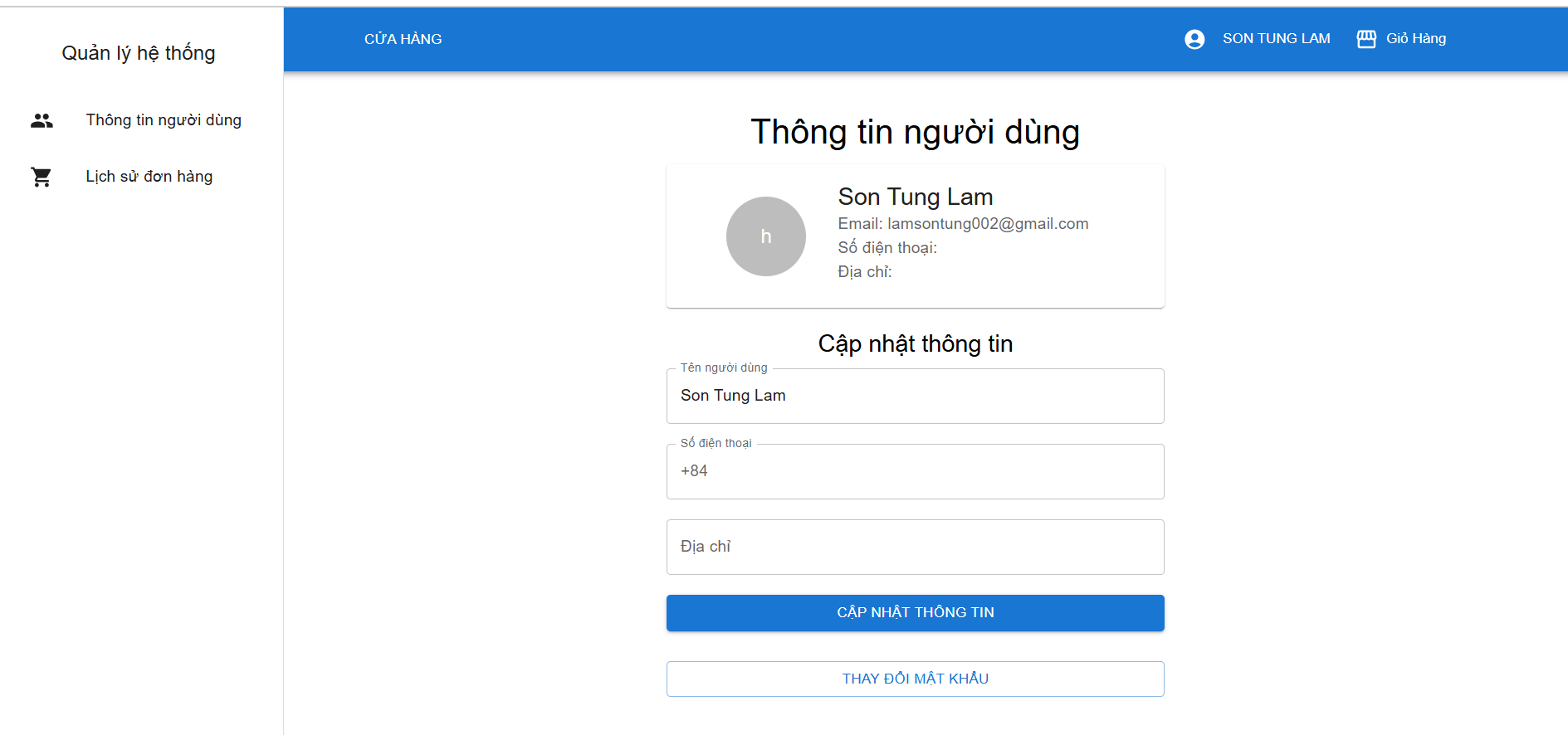
### 3.4.3 Mô hình quan hệ

* USERS (USER\_ID, USERNAME, PASSWORD, EMAIL, PHONE, ADDRESS, CREATED\_AT, UPDATED\_AT, USER\_STATUS, AVATAR\_USER).
* WISHLIST (WISHLIST\_ID, USER\_ID, PRODUCT\_ID).
* BLOGS (BLOG\_ID, BLOG\_TITLE, BLOG\_CONTENT, BLOG\_IMAGE\_URL, BLOG\_AUTHOR, CREATED\_AT, BLOG\_STATUS).
* CART (CART\_ID, PRODUCT\_ID).
* CATEGORIES (CATEGORY\_ID, CATEGORY\_NAME, CATEGORY\_DESCRIPTION, CATEGORY\_STATUS).
* PRODUCTS (PRODUCT\_ID, CATEGORY\_ID, NAME, PRODUCT\_PRICE, MATERIAL, WEIGHT, SIZE, IMAGE\_URL, STOCK\_QUANTITY, CREATED\_AT, UPDATED\_AT, PRODUCT\_STATUS).
* PROMOTIONS (PROMOTION\_ID, PRODUCT\_ID, PROMOTION\_TITLE, PROMOTION\_DESCRIPTION, PROMOTION\_DISCOUNT, PROMOTION\_START\_DATE, PROMOTION\_END\_DATE).
* ORDERS (ORDER\_ID, USER\_ID, ORDER\_STATUS, TOTAL\_PRICE, PAYMENT\_METHOD, CREATED\_AT, UPDATED\_AT, ADDRESS\_ORDER, PHONE\_ORDER).
* ORDERDETAILS (ORDER\_DETAIL\_ID, ORDER\_ID, PRODUCT\_ID, QUANTITY, PRICE, DISCOUNT\_PRODUCT).

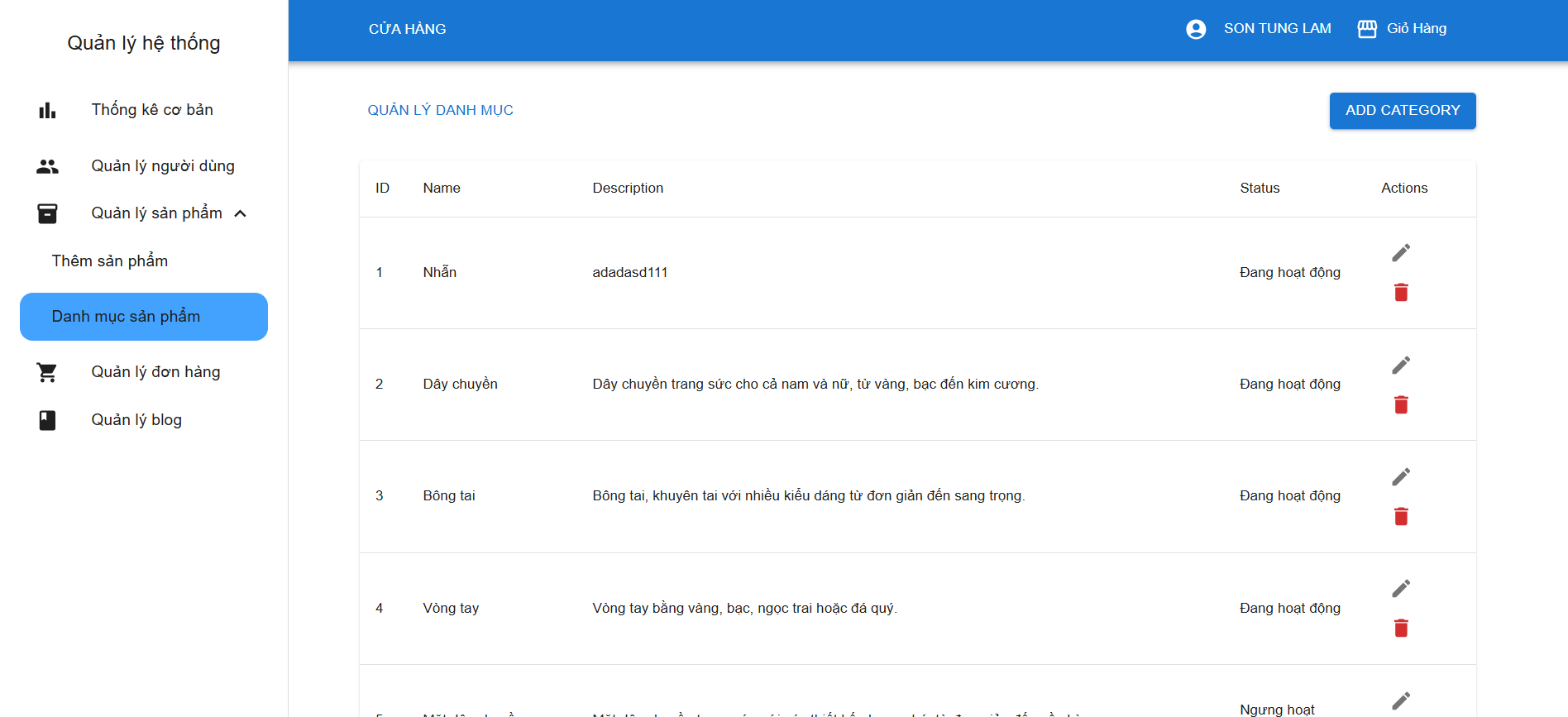
### 3.4.4 Thiết kế xử lý



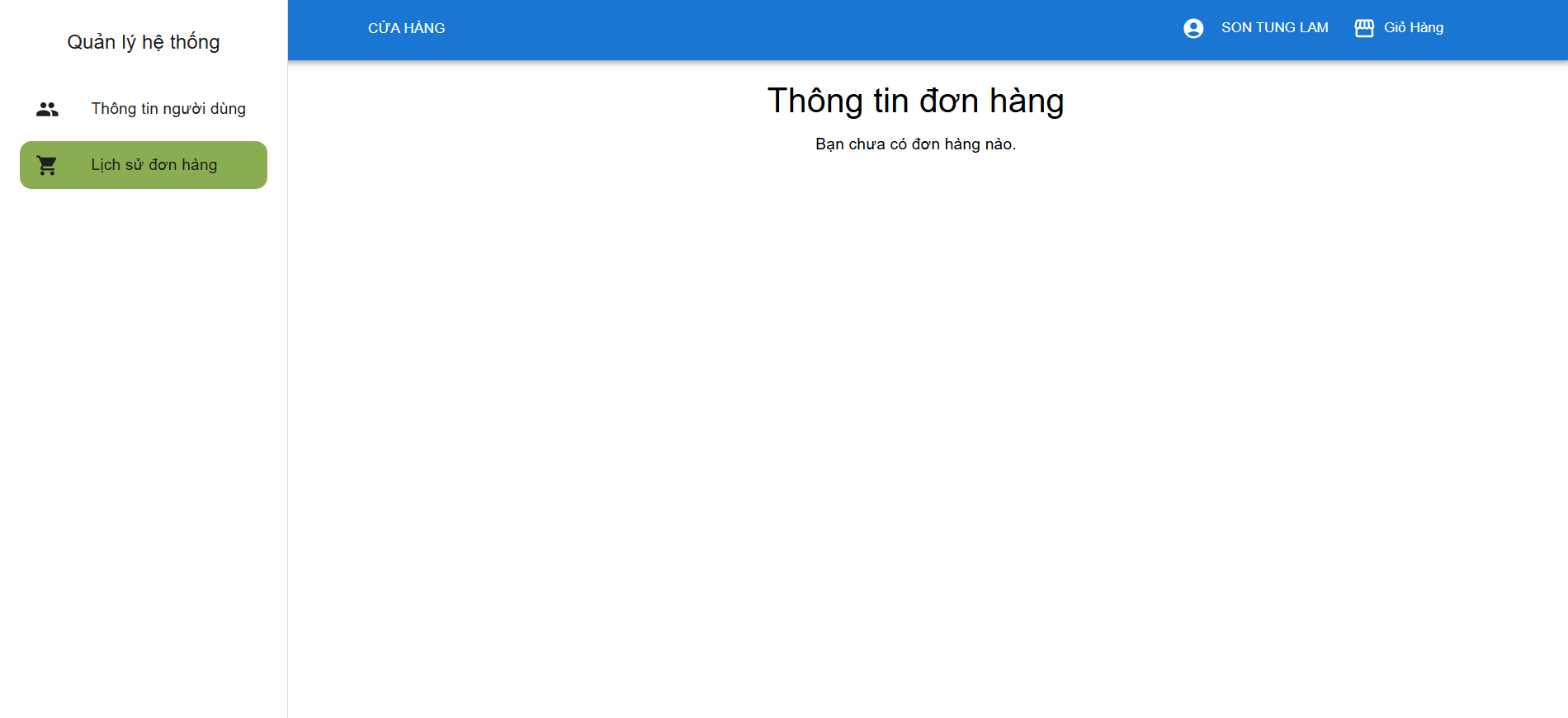
**Hình 3****.** Giao diện trang chủ



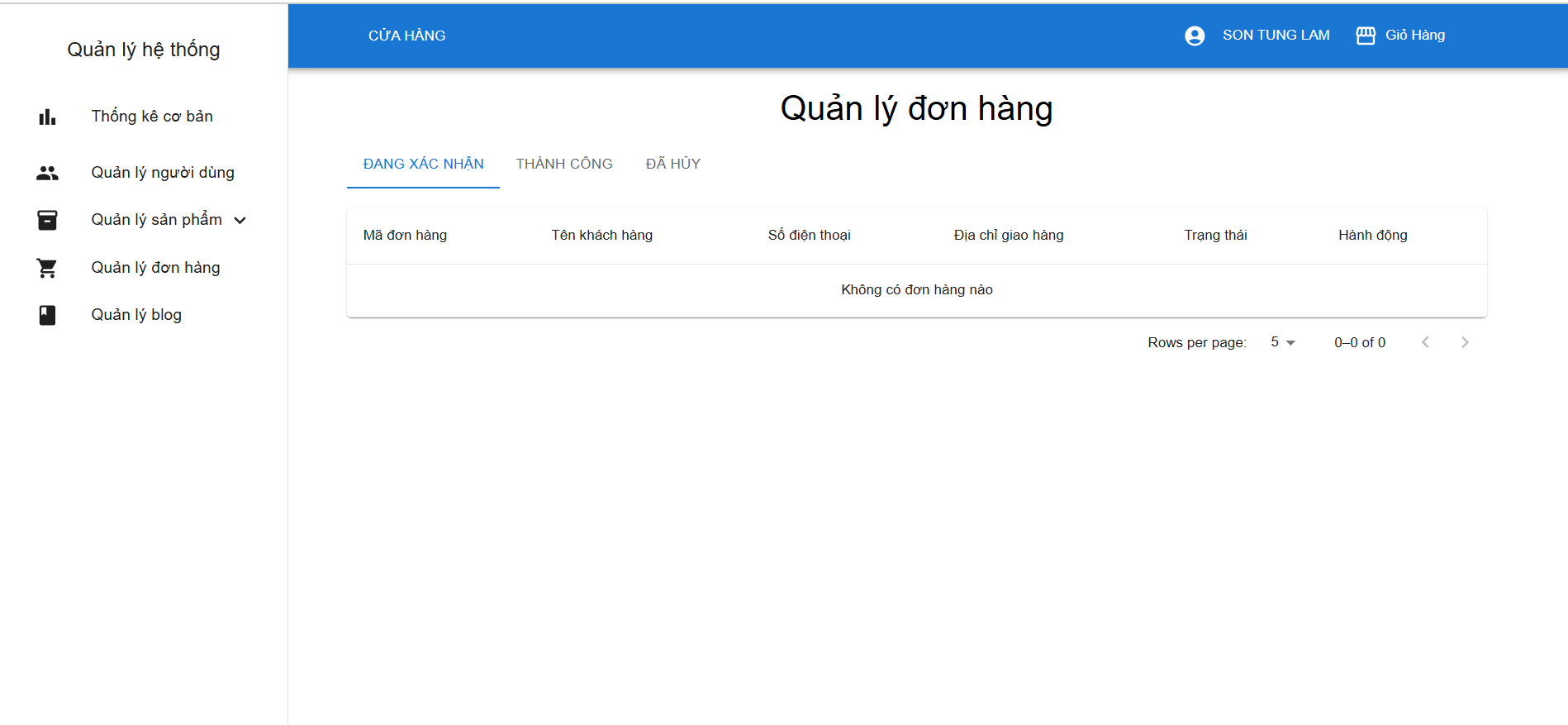
**Hình 4****.** Giao diện trang thông tin cá nhân



**Hình 5****.** Giao diện trang sản phẩm



**Hình 6****.** Giao diện trang lịch sử đơn hàng



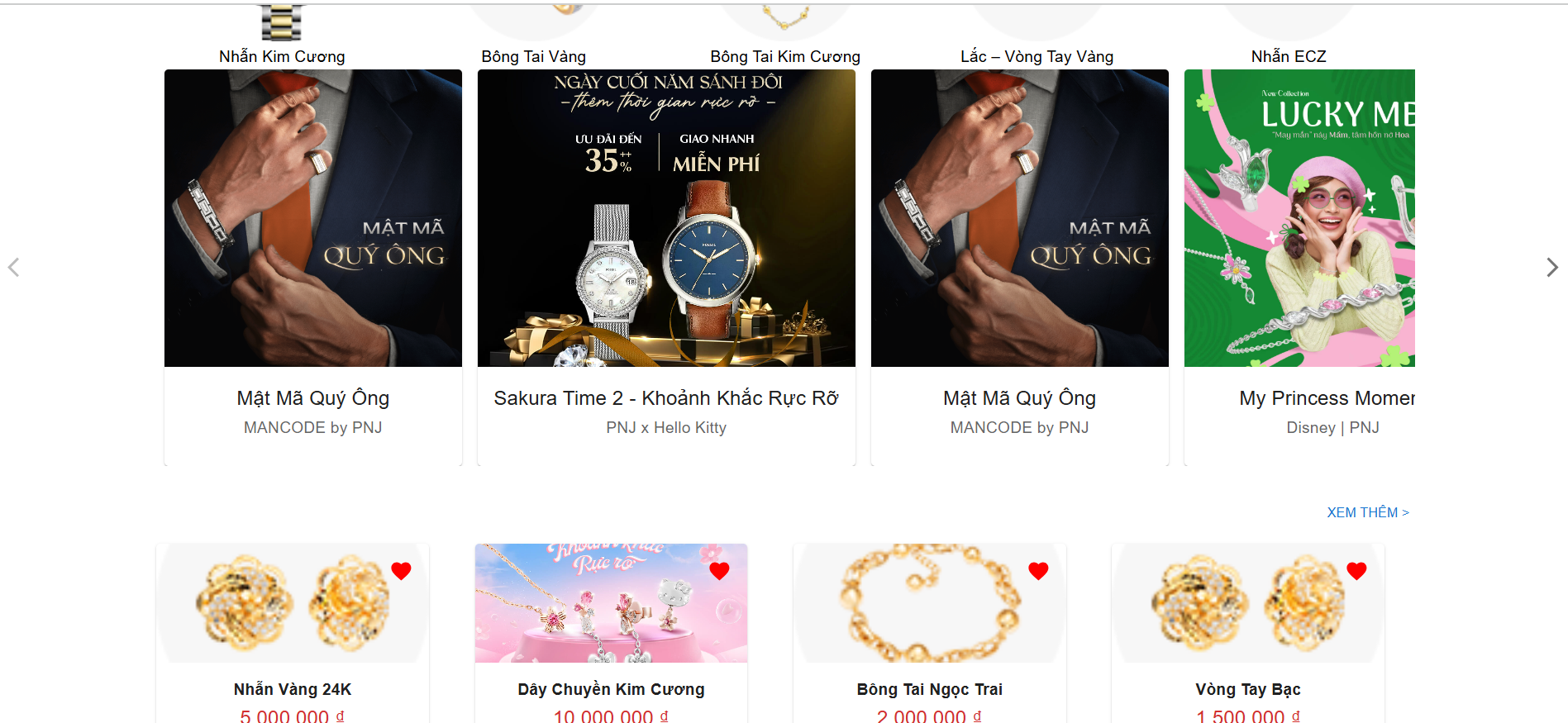
**Hình 7****.** Giao diện trang quản lý đơn hàng

# CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

Trình bày các kết quả đạt được sau quá trình thực hiện đồ án. Có thể đánh giá về hiệu năng, trải nghiệm người dùng, hoặc trình bày các giao diện chức năng của nghiên cứu ở phần này.

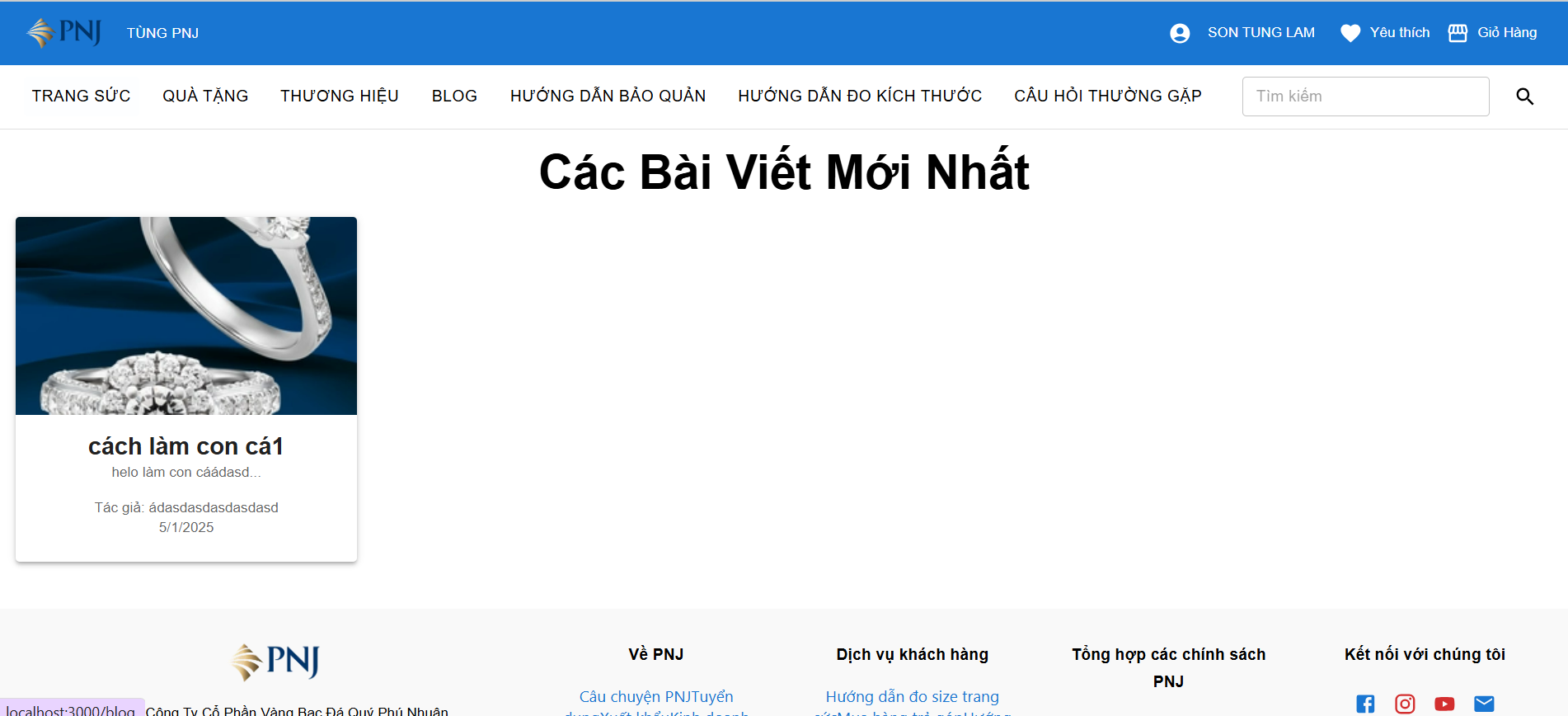
## 4.1 Hình ảnh các chức năng

Giao diện trang chủ tạo cho người dùng cảm giác thoải mái khi mua hàng



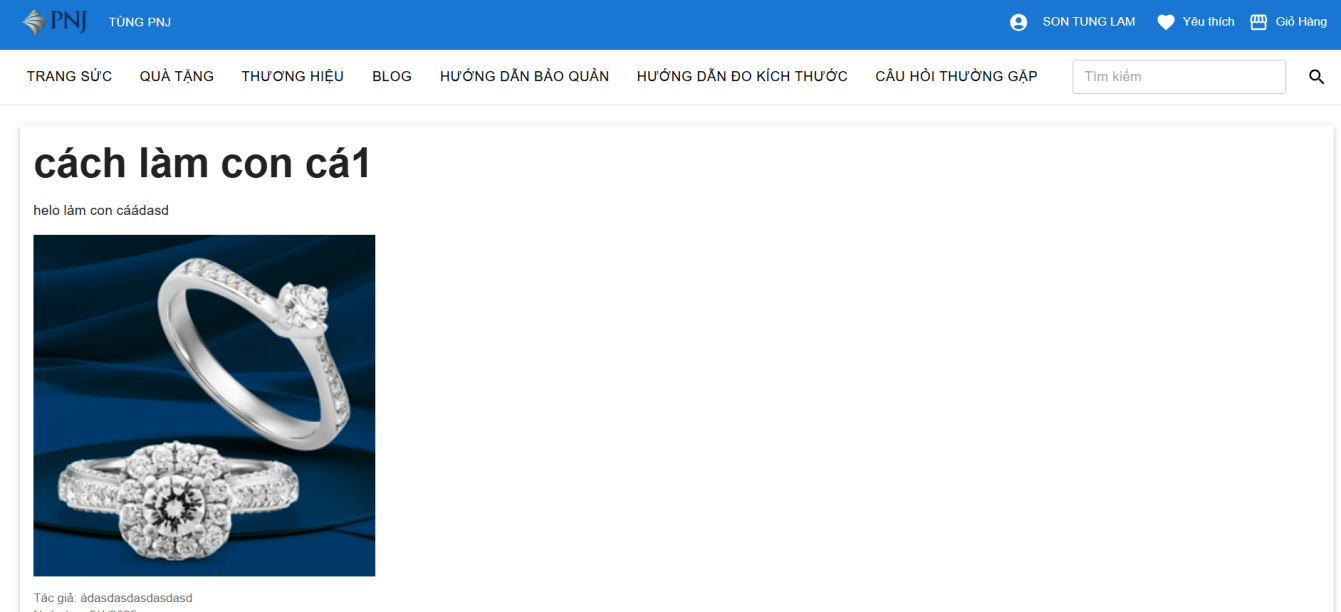
**Hình 8****.** Trang chủ

Giao diện trang tin tức hiển thị các tin tức nổi bật của sản phẩm



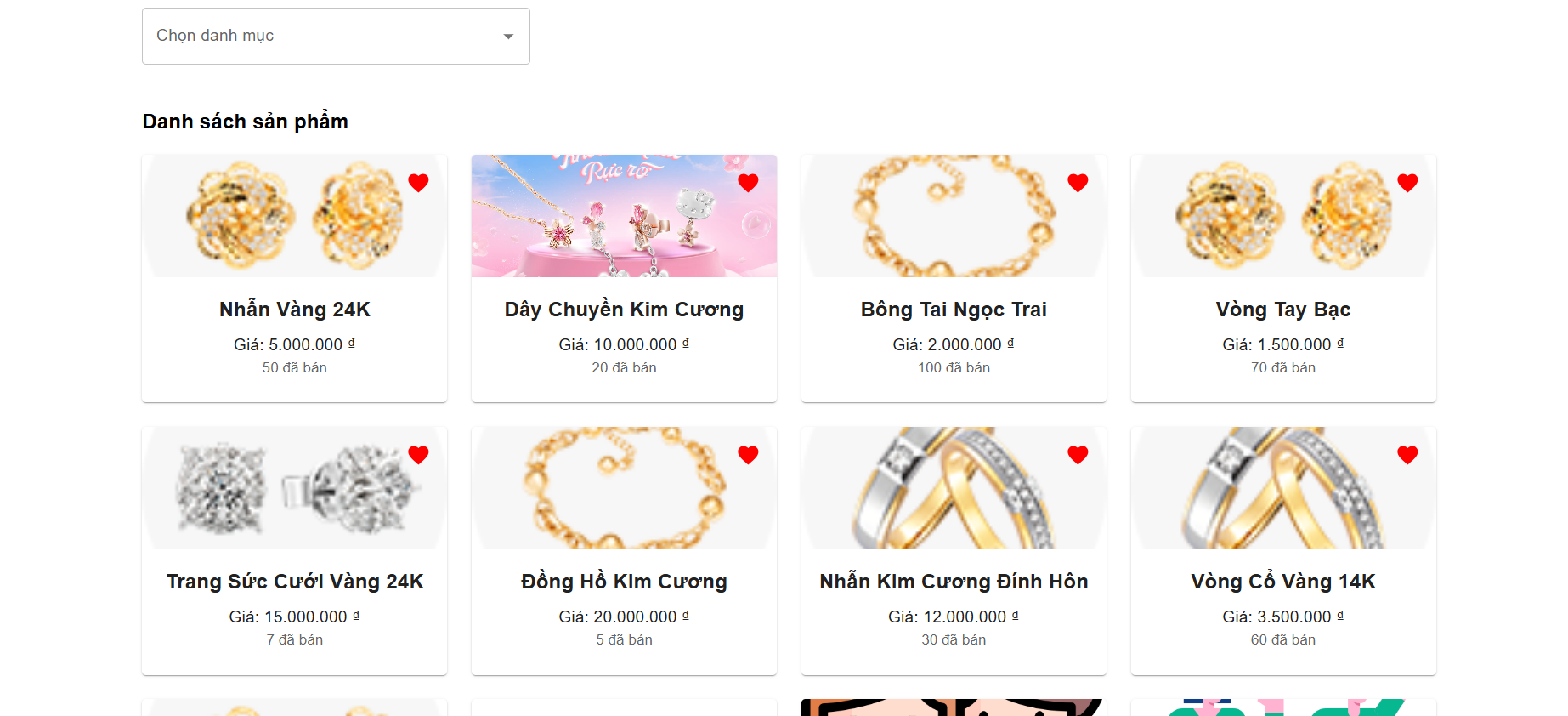
**Hình 9****.** Giao diện trang bài viết

Giao diện hiển thị chi tiết từng bài viết mà người quản trị có thể tạo thêm dể giới thiệu các sản phẩm mới ra mắt



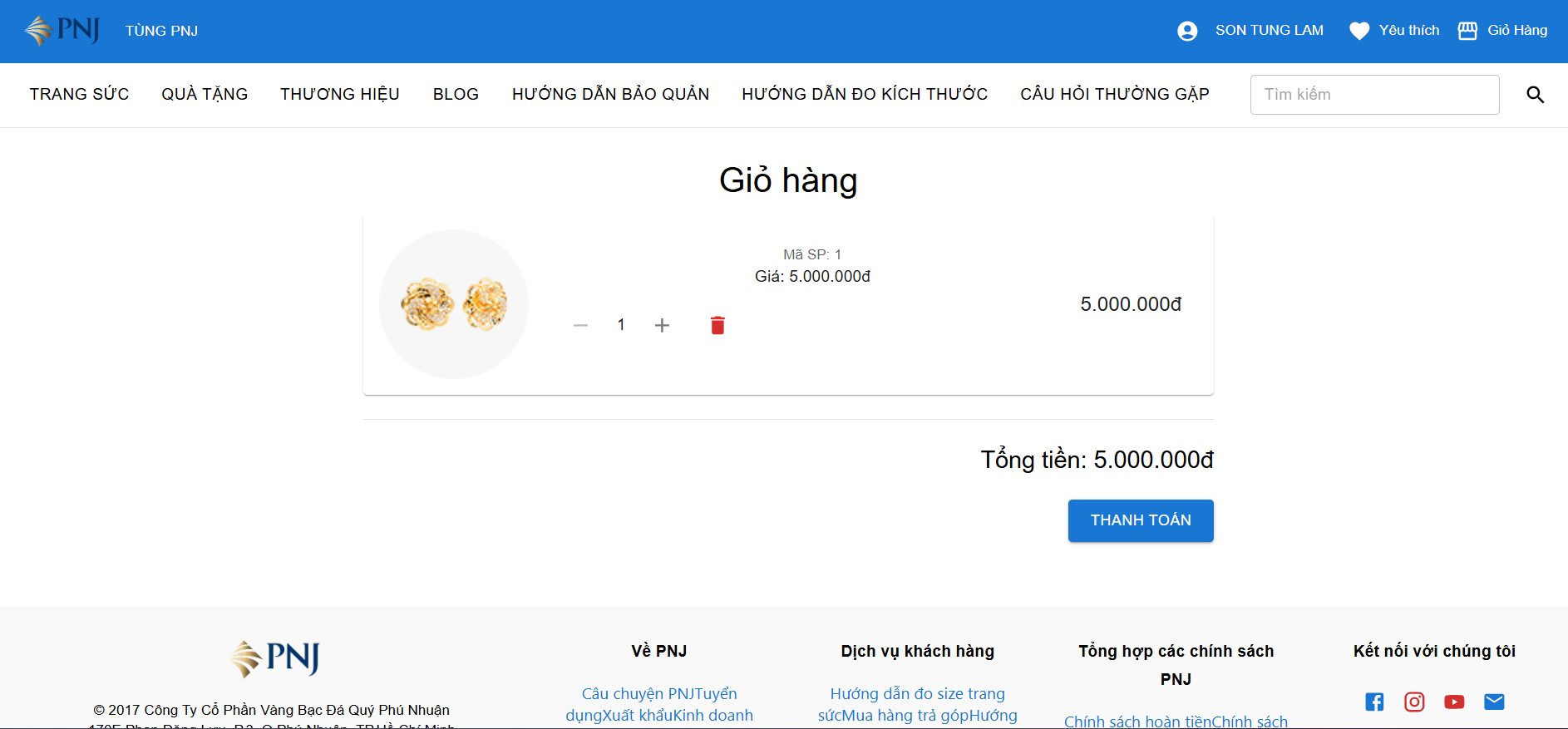
**Hình 10****.** Giao diện chi tiết bài viết

Giao diện hiển thị tất cả sản phẩm, ở đây người dùng có thể xem tất cả sản phẩm, và chọn lọc sản phẩm theo mong muốn của bản thân



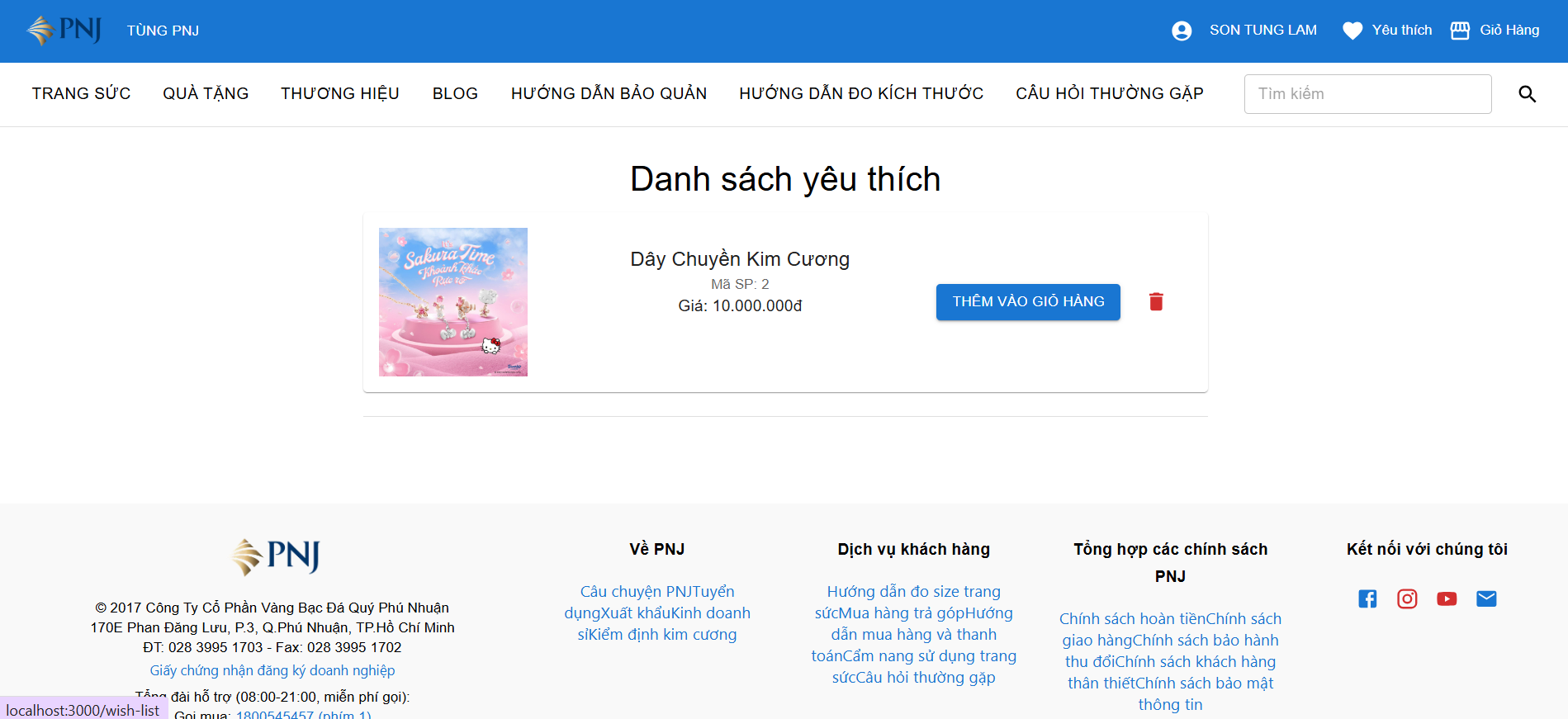
**Hình 11**. Giao diện trang hiển thị sản phẩm

Giao diện hiển thị giỏ hàng của người dùng khi người dùng đăng nhập thành công và lưu trữ sản phẩm trong giỏ hàng.



**Hình 12****.** Giao diện trang giỏ hàng

Giao diện hiển thị sản phẩm yêu thích của người dùng, khi mà người dùng muốn muốn lưu trữ lại sản phẩm mà bản thân yêu thích trước đó.



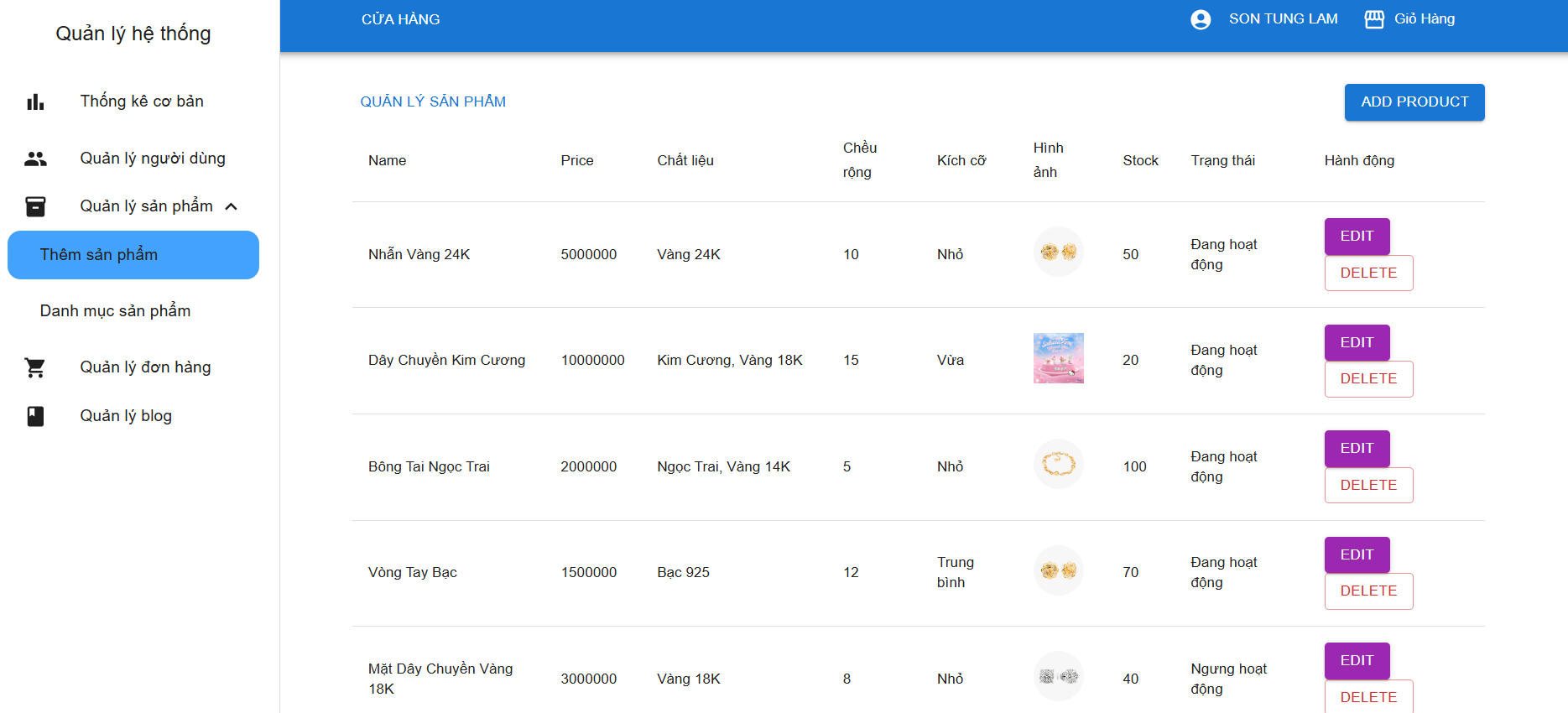
**Hình 13****.** Giao diện trang sản phẩm yêu thích

Giao diện quản lý thông tin bài viết, người quản trị có thể tạo ra các bài viết và xóa các bài viết mà người quản trị muốn.



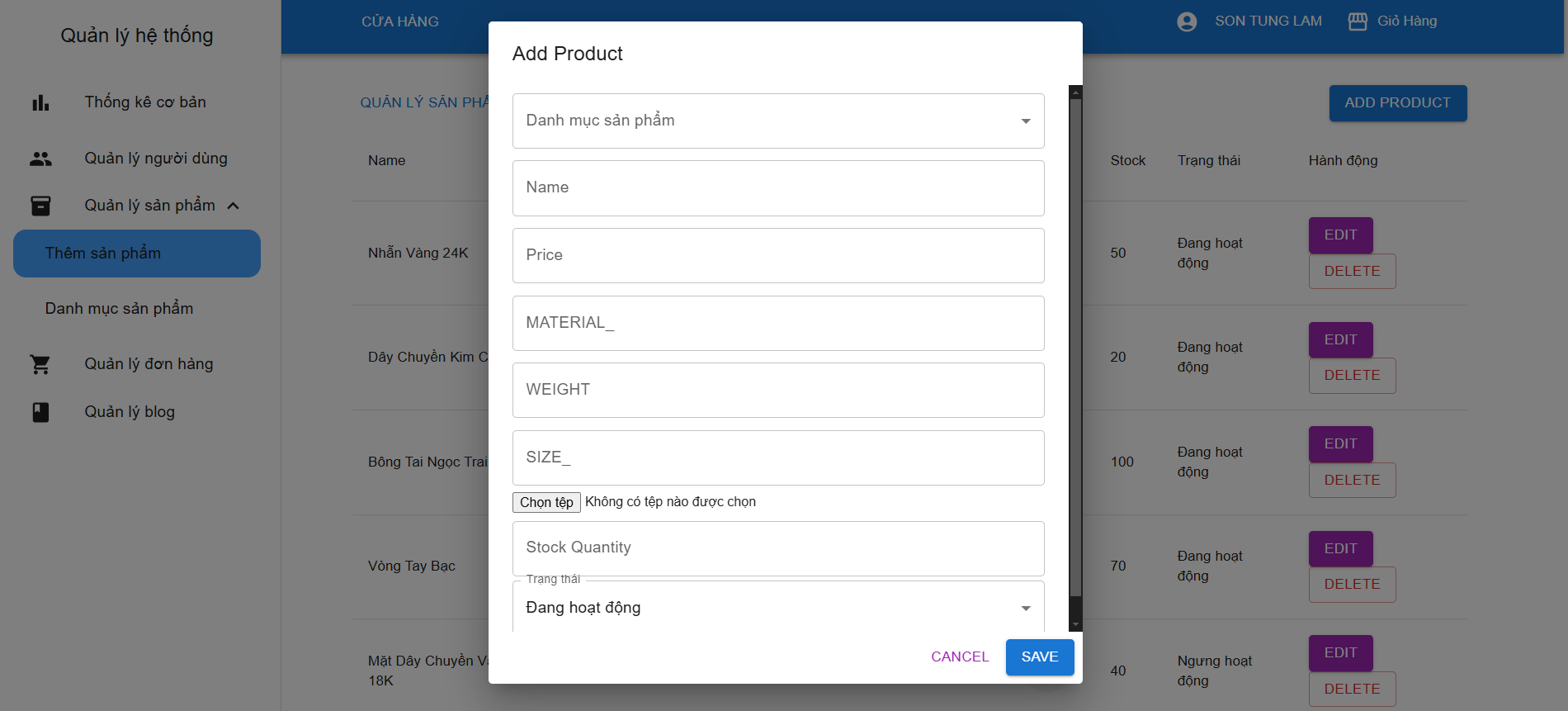
**Hình 14****.** Giao diện quản lý bài viết cho người quản trị

Giao diện quản lý sản phảm của quản trị viên, ở trang này quản trị viên có thể thêm, sửa, xóa thông tin sản phẩm mà người quản trị viên muốn, cũng như có thể tìm kiếm, lọc sản phẩm.



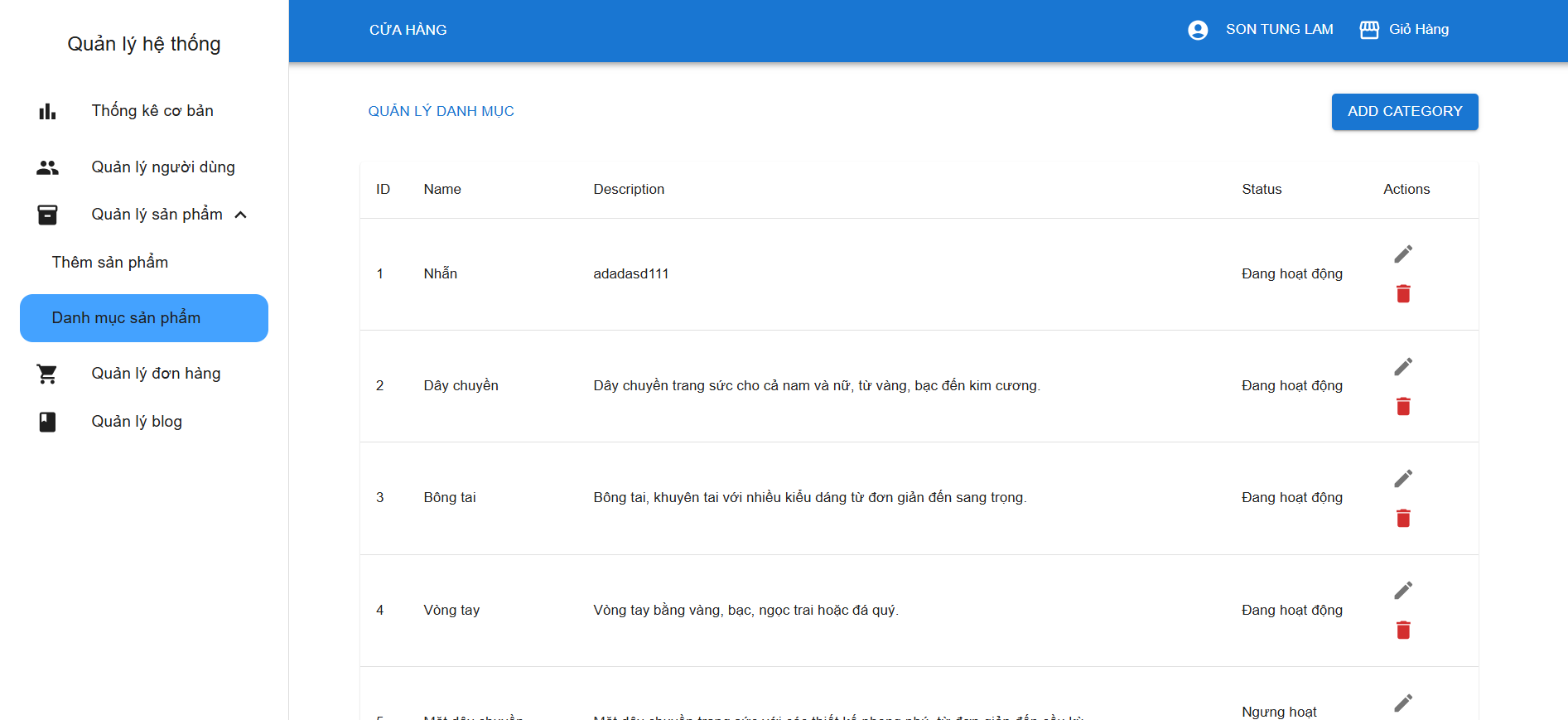
**Hình 15**. Giao diện quản lý thông tin sản phảm ở phía người quản trị

Giao diện quản thêm sản phẩm mới cho người quản trị.



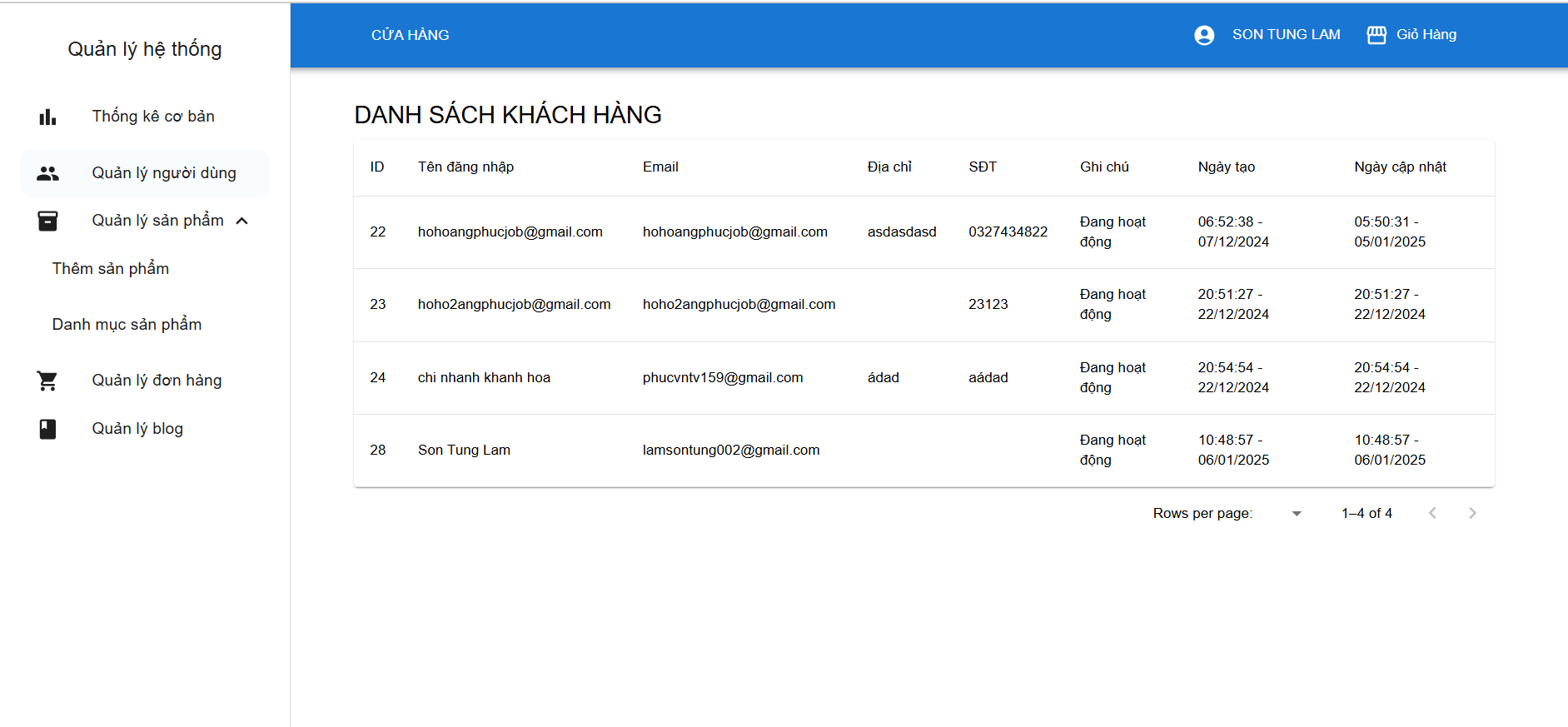
**Hình 16****.** Giao diện quản lý thêm sản phẩm cho người quản trị

Giao diện quản lý sản phẩm ưu tiên mà người quản trị mong muốn sản phẩm đó hiển thị lên đầu tiên, phục vụ cho việc giới thiệu các sản phẩm mà họ mong muốn.



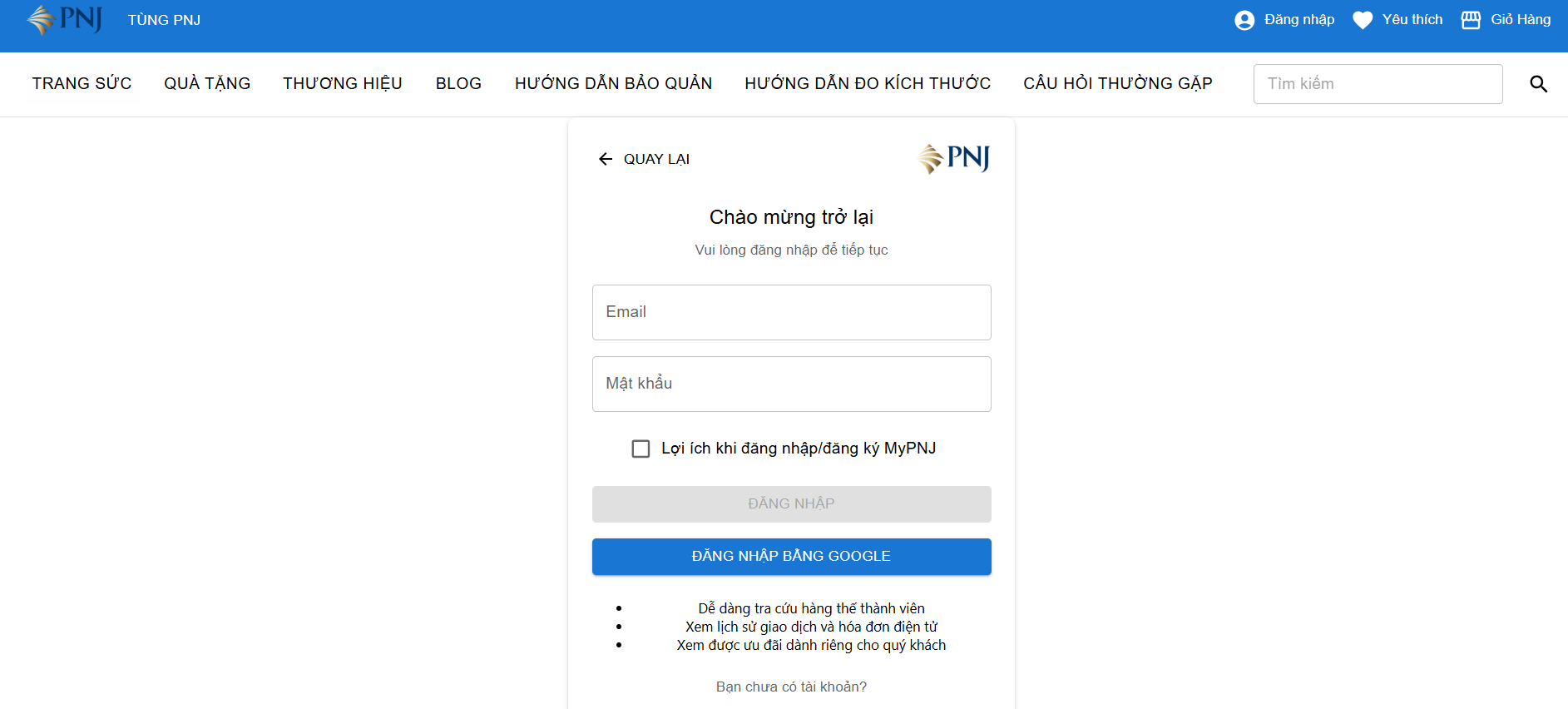
**Hình 17****.** Giao diện quản lý hiển thị sản phẩm ưu tiên dành cho người quản trị

Giao diện quản lý người dùng dành cho quản trị viên, ở trang quản lý người dùng này người quản trị có thể kiểm soát, thay đổi thông tin, khóa tài khoản người dùng đó.



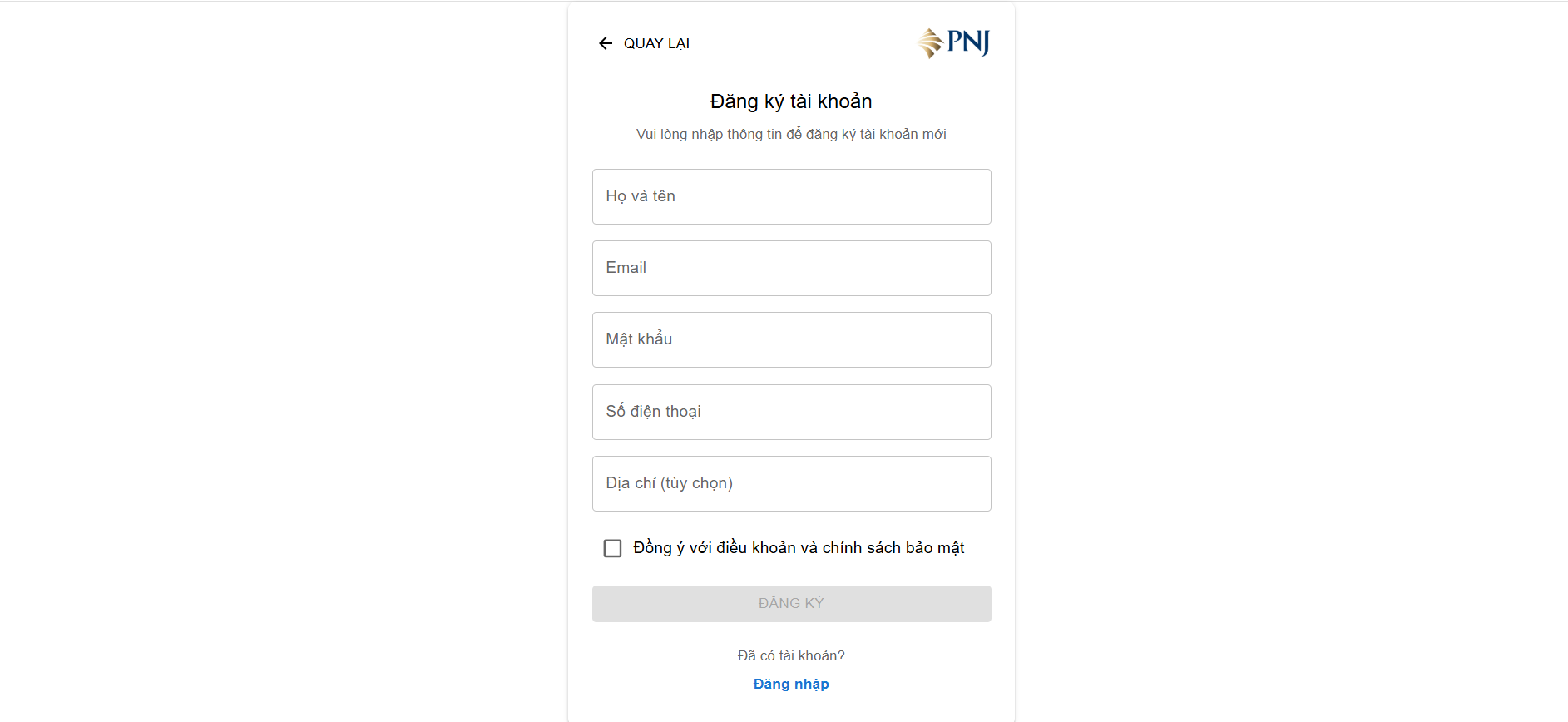
**Hình 18****.** Giao diện quản lý thông tin người dùng dành cho người quản trị viên

Giao diện đăng nhập của người dùng, khi mà người dùng bấm vào đăng nhập thì sẽ có hai phương thức đăng nhập trong một tài khoản: đăng nhập bằng tài khoản google, và đăng nhập bằng địa chỉ email và mật khẩu.



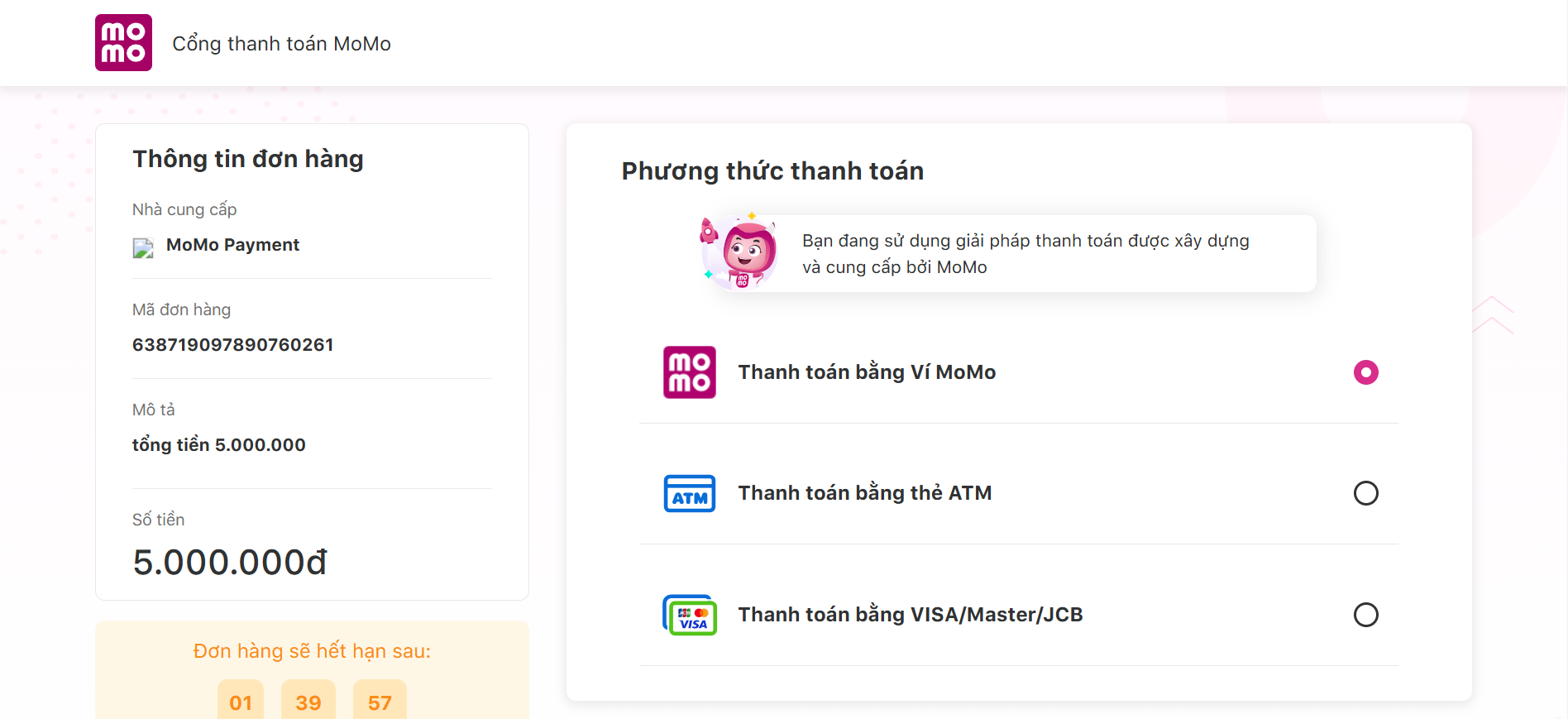
**Hình 19****.** Giao diện chức năng đăng nhập

Giao diện đăng ký tài khoản, khi mà người dùng nhập đủ thông tin để đăng ký thì hệ thống sẽ gửi mã xác nhận về gmail của người dùng nếu người dùng nhập mã xác nhận chính xác thì có thể đăng ký tài khoản của hệ thống này



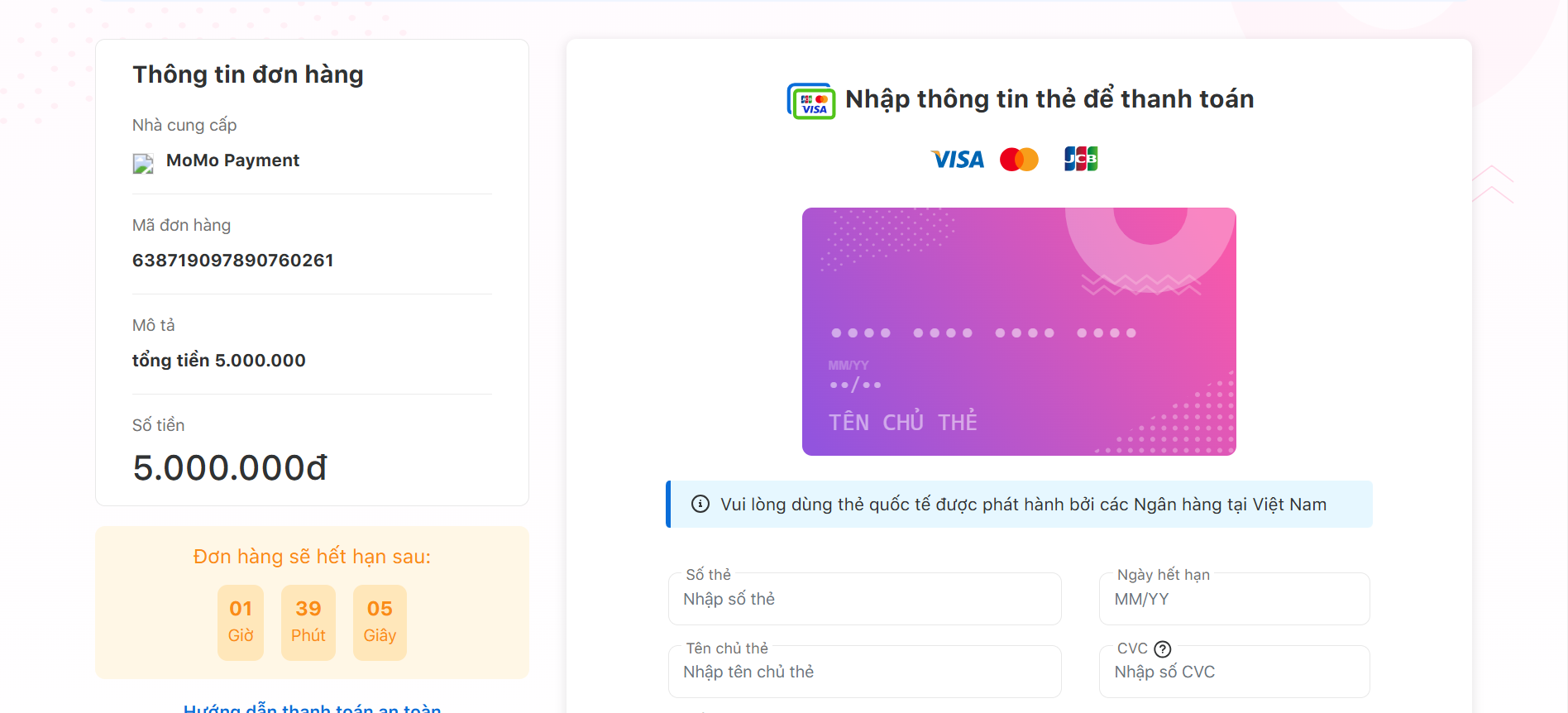
**Hình 20****.** Giao diện đăng ký tài khoản của hệ thống

Giao diện cổng thanh toán momo khi người dùng chọn chức năng thanh toán bằng momo.



**Hình 21****.** Giao diện thanh toán bằng momo

Người khách hàng nhập đầy đủ thông tin thẻ ngân hàng của mình



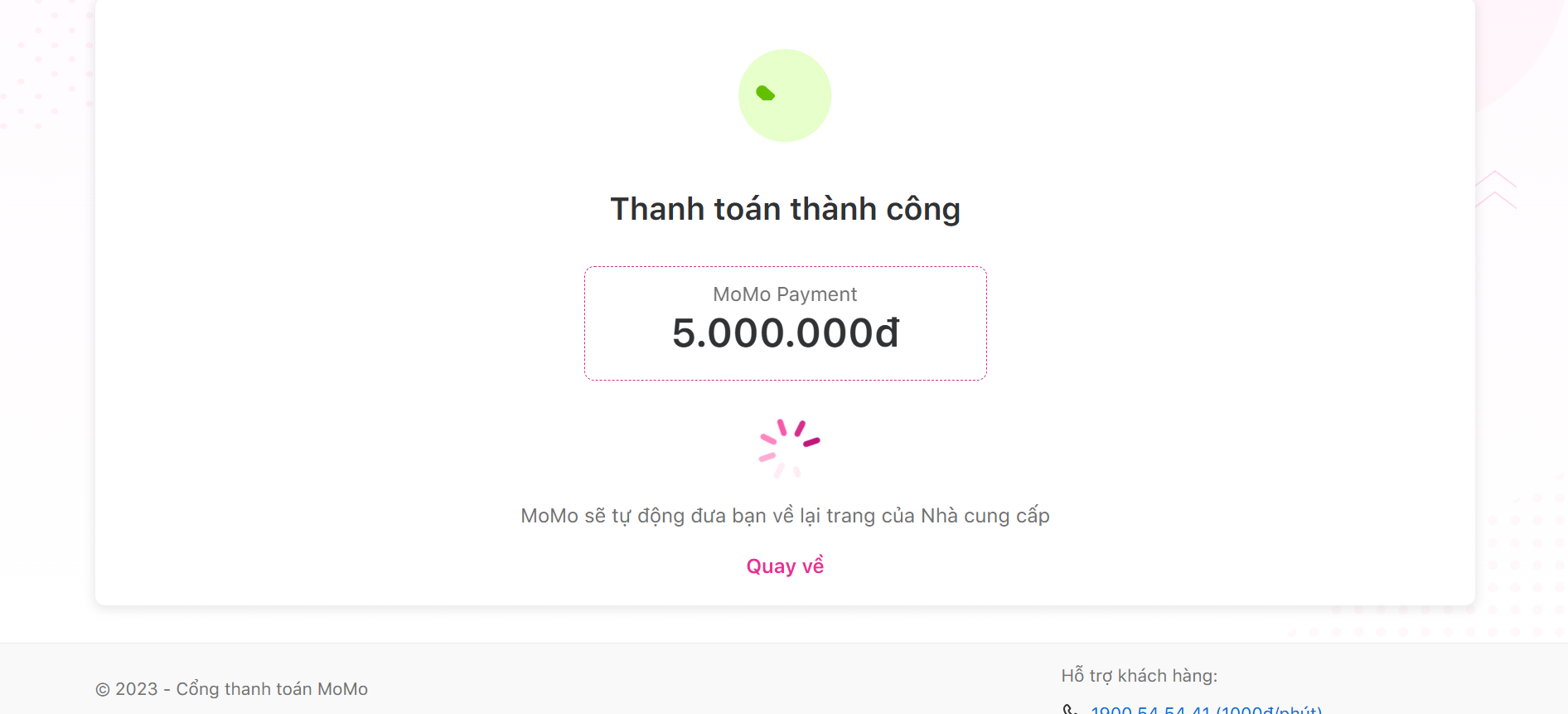
**Hình 22****.** Giao diện người dùng nhập đầy đủ thông tin ngân hàng của mình

Khi mà người dùng nhập đầy đủ thông tin thì sẽ thực hiện nhập mã xác nhận

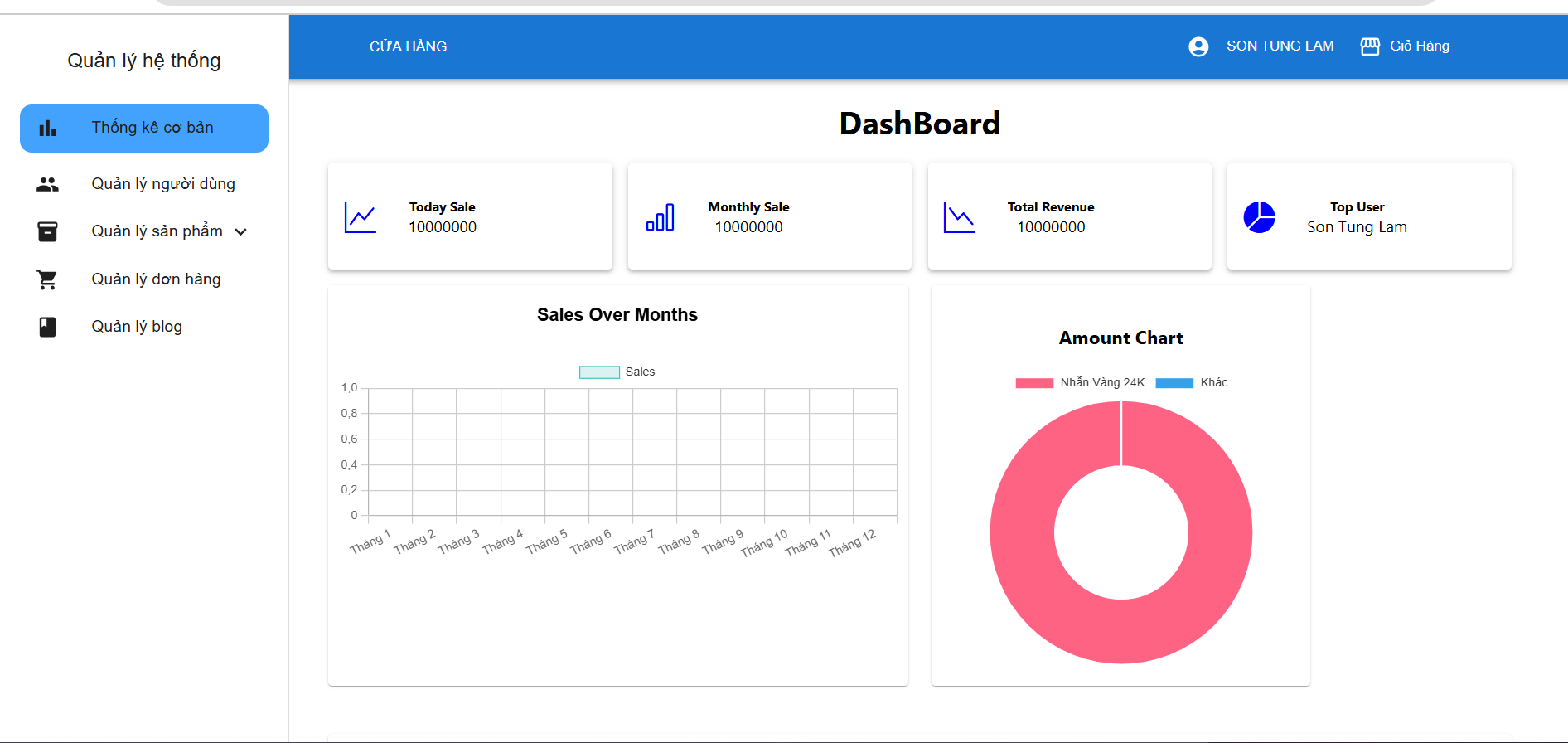


**Hình 23****.** Giao diện người dùng nhập mã xác nhận

Khi người dùng nhập đúng thông tin thì sẽ có thông báo thanh toán thành công.



**Hình 24****.**Giao diện thông báo khi người dùng thanh toán momo thành công



**Hình 25****.** Giao diện thống kê doanh thu sản phẩm dành cho quản trị viên

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết luận: Trình bày những kết quả đạt được, những đóng góp mới và những đề xuất mới. Phần kết luận cần ngắn gọn, không có lời bàn và bình luận thêm.

5.1 Kết luận  
 Nghiên cứu và phát triển hệ thống website thương mại điện tử cho sản phẩm trang sức đã hoàn thành các mục tiêu đề ra. Hệ thống đáp ứng đầy đủ các chức năng quan trọng như quản lý sản phẩm, giỏ hàng, thanh toán, quản lý tài khoản người dùng, và hỗ trợ khách hàng. Ngoài ra, các yêu cầu phi chức năng như bảo mật, hiệu suất, và trải nghiệm người dùng cũng được đảm bảo, giúp hệ thống hoạt động ổn định và mang lại trải nghiệm tốt cho người dùng.  
 Dự án không chỉ giải quyết được các vấn đề thực tế trong thương mại điện tử mà còn áp dụng thành công các công nghệ hiện đại như Node.js, React.js, và MySQL, góp phần nâng cao kỹ năng lập trình và khả năng quản lý dự án của nhóm thực hiện. Tuy nhiên, vẫn còn một số hạn chế cần cải thiện, như tối ưu hệ thống gợi ý sản phẩm và mở rộng tích hợp các cổng thanh toán quốc tế.

5.2 Hướng phát triển  
Dựa trên kết quả đạt được, hệ thống có thể được phát triển thêm ở các khía cạnh sau:

* **Tối ưu thuật toán gợi ý sản phẩm**: Sử dụng các thuật toán máy học (Machine Learning) để cá nhân hóa đề xuất sản phẩm dựa trên hành vi và sở thích của người dùng.
* **Phát triển ứng dụng di động**: Xây dựng ứng dụng di động trên cả hai nền tảng iOS và Android để tăng khả năng tiếp cận người dùng và nâng cao trải nghiệm mua sắm.
* **Hỗ trợ đa ngôn ngữ**: Thêm tính năng hỗ trợ nhiều ngôn ngữ để mở rộng phạm vi hoạt động, tiếp cận khách hàng quốc tế.
* **Tích hợp AI chatbot**: Sử dụng chatbot trí tuệ nhân tạo để hỗ trợ khách hàng tự động, nhanh chóng và hiệu quả.
* **Tăng cường bảo mật**: Triển khai các phương pháp bảo mật như xác thực hai yếu tố (2FA) và mã hóa dữ liệu để đảm bảo an toàn thông tin người dùng.
* **Mở rộng tính năng quản lý**: Phát triển thêm hệ thống quản lý kho, nhập hàng, và nhà cung cấp để hỗ trợ vận hành hiệu quả hơn.
* **Phân tích dữ liệu người dùng**: Tích hợp hệ thống phân tích dữ liệu để hiểu rõ hành vi khách hàng và đưa ra các chiến lược kinh doanh phù hợp.
* **Hỗ trợ các cổng thanh toán đa dạng**: Mở rộng tích hợp với nhiều cổng thanh toán quốc tế như PayPal, Stripe, và các ví điện tử phổ biến khác.

Những hướng phát triển này không chỉ giúp hoàn thiện hệ thống mà còn mở rộng phạm vi ứng dụng, đáp ứng tốt hơn nhu cầu đa dạng của người dùng trong lĩnh vực thương mại điện tử.

# DANH MỤC TÀI LIỆU THAM KHẢO

# PHỤ LỤC