

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:

**XÂY DỰNG WEBSITE QUẢN LÝ BÁN
NUỚC HOA**

Người hướng dẫn: **THS. NGUYỄN THỊ MINH HỶ**

Sinh viên thực hiện: **Nguyễn Thanh Lịch**

Số thẻ sinh viên: **102200021**

Lớp: **20T1**

Đà Nẵng, 6/2024

NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN

Dà Nẵng, ngày ... tháng ... năm 2024
Giảng viên hướng dẫn

Ths. Nguyễn Thị Minh Hỷ

NHẬN XÉT CỦA NGƯỜI PHẢN BIỆN

Đà Nẵng, ngày ... tháng ... năm 2024

TÓM TẮT

Tên đề tài: Xây dựng website quản lý bán nước hoa

Sinh viên thực hiện: Nguyễn Thanh Lịch

Số thẻ SV: 102200021 Lớp: 20T1

Hệ thống bán nước hoa trực tuyến cung cấp một nền tảng thương mại điện tử toàn diện, giúp người dùng có thể mua sắm các sản phẩm nước hoa từ nhiều thương hiệu khác nhau một cách tiện lợi và nhanh chóng thông qua internet. Hệ thống này được thiết kế nhằm tạo ra một trải nghiệm mua sắm mượt mà, từ khâu tìm kiếm sản phẩm, xem thông tin chi tiết, đặt hàng, thanh toán trực tuyến cho đến việc giao hàng tận nơi.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Nguyễn Thanh Lịch

Số thẻ sinh viên: 102200021

Lớp: 20T1 Khoa: Công nghệ thông tin

Ngành: Công nghệ thông tin

1. *Tên đề tài đồ án:*

.....
.....

2. *Đề tài thuộc diện:* Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. *Các số liệu và dữ liệu ban đầu:*

.....
.....
.....

4. *Nội dung các phần thuyết minh và tính toán:*

.....
.....
.....
.....
.....

5. *Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):*

.....
.....
.....
.....

6. *Họ tên người hướng dẫn:*

7. *Ngày giao nhiệm vụ đồ án:*/...../2024

8. *Ngày hoàn thành đồ án:*/...../2024

| | |
|----------------------------|-------------------------------------|
| | <i>Dà Nẵng, ngày tháng năm 2024</i> |
| Trưởng Bộ môn | Người hướng dẫn |

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sâu sắc đến các thầy cô trong Khoa Công nghệ thông tin, cũng như tất cả các thầy cô trong trường Đại học Bách khoa – Đại học Đà Nẵng đã dùn dắt, dạy dỗ và truyền đạt kiến thức, kinh nghiệm quý báu của mình trong suốt quá trình em học tập và nghiên cứu tại trường.

Em xin bày tỏ tình cảm và lòng biết ơn chân thành của em tới cô giáo Ths. Nguyễn Thị Minh Hỷ, người đã từng bước hướng dẫn, giúp đỡ em tận tình trong quá trình thực hiện đồ án tốt nghiệp của mình. Nhờ đó em có thể hoàn thành đồ án đúng tiến độ và tích lũy cho mình nhiều kiến thức quý báu.

Con xin gửi lời cảm ơn to lớn nhất đến cha, mẹ và gia đình. Cha, mẹ và gia đình đã luôn ở bên con, là nguồn động lực không mệt mỏi và là chỗ dựa tinh thần vững chắc giúp con vượt qua những khó khăn để hoàn thành đồ án này.

Mặc dù đã cố gắng hoàn thành đồ án tốt nhất nhưng thời gian và kiến thức còn có hạn nên sẽ không tránh khỏi những thiếu sót nhất định, rất mong được sự cảm thông và tận tình chỉ bảo, góp ý của quý thầy cô giáo cũng như tất cả các bạn để kết quả của em được hoàn thiện hơn.

Một lần nữa em xin chân thành cảm ơn!

Nguyễn Thanh Lịch

CAM ĐOAN

Em xin cam đoan:

1. Nội dung trong đồ án này là do em thực hiện dưới sự hướng dẫn trực tiếp của ThS Nguyễn Thị Minh Hỷ.
2. Các tham khảo dùng trong đồ án đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.
3. Nếu có những sao chép không hợp lệ, vi phạm quy chế đào tạo, em xin chịu hoàn toàn trách nhiệm.

Đà Nẵng, ngày tháng 6 năm 2024

Sinh viên thực hiện

Nguyễn Thanh Lịch

MỤC LỤC

| | |
|---|------|
| TÓM TẮT | |
| NHIỆM VỤ ĐỒ ÁN | i |
| LỜI CẢM ƠN | |
| CAM ĐOAN | ii |
| MỤC LỤC | iii |
| DANH SÁCH CÁC BẢNG | v |
| DANH SÁCH CÁC HÌNH VẼ | vi |
| DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT | viii |
| MỞ ĐẦU | 1 |
| CHƯƠNG I: CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG | 3 |
| 1.1. Kiến thức cơ bản | 3 |
| 1.2. Tổng quan về Node.js và Express.js framework | 4 |
| 1.3. Tổng quan về RESTful API | 6 |
| 1.4. Tổng quan về ReactJS | 9 |
| 1.5. Công cụ sử dụng | 15 |
| CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG | 17 |
| 2.1. KHẢO SÁT BÀI TOÁN THỰC TẾ | 17 |
| 2.1.1. Tổng quan hiện trạng | 17 |
| 2.1.2. Nhu cầu sử dụng nước hoa | 17 |
| 2.2. PHÂN TÍCH CHỨC NĂNG | 18 |
| 2.2.1. Khách có các chức năng sau | 18 |
| 2.2.2. Khách hàng có các chức năng sau | 19 |
| 2.2.3. Quản trị viên có các chức năng sau | 19 |
| 2.3. SƠ ĐỒ CA SỬ DỤNG | 20 |
| 2.3.1. Các tác nhân trong hệ thống | 20 |
| 2.3.2. Sơ đồ ca sử dụng tổng quát của hệ thống | 21 |
| 2.3.3. Sơ đồ ca sử dụng của Khách | 22 |
| 2.3.4. Sơ đồ ca sử dụng của Khách hàng | 22 |
| 2.3.5. Sơ đồ ca sử dụng của Quản trị viên | 23 |
| 2.4. ĐẶC TẢ CA SỬ DỤNG | 26 |
| 2.4.1. Đặc tả ca sử dụng Đăng ký | 26 |
| 2.4.2. Đặc tả ca sử dụng Đăng nhập | 27 |
| 2.4.3. Đặc tả ca sử dụng Quản lý tài khoản | 27 |
| 2.4.4. Đặc tả ca sử dụng Quản lý các đơn hàng | 28 |
| 2.4.5. Đặc tả ca sử dụng Quản lý mua hàng | 28 |

| | |
|--|-----------|
| 2.4.6. <i>Đặc tả ca sử dụng Quản lý Thêm danh mục</i> | 29 |
| 2.4.7. <i>Đặc tả ca sử dụng Quản lý thêm sản phẩm.</i> | 29 |
| 2.4.8. <i>Đặc tả ca sử dụng Quản lý người dùng</i> | 30 |
| 2.4.9. <i>Đặc tả ca sử dụng Quản lý thêm thương hiệu</i> | 30 |
| 2.4.10. <i>Đặc tả ca sử dụng Quản lý thêm phiếu giảm giá</i> | 31 |
| 2.4.11. <i>Đặc tả ca sử dụng Quản lý đơn hàng</i> | 31 |
| 2.5. SƠ ĐỒ HOẠT ĐỘNG | 32 |
| 2.5.1. <i>Sơ đồ hoạt động của khách hàng</i> | 32 |
| 2.5.2. <i>Sơ đồ hoạt động của quản trị viên</i> | 33 |
| 2.6. KIẾN TRÚC HỆ THỐNG | 33 |
| 2.7. THIẾT KẾ CƠ SỞ DỮ LIỆU | 33 |
| 2.7.1. <i>Mô hình thiết kế cơ sở dữ liệu</i> | 33 |
| 2.8.2. <i>Chi tiết các bảng dữ liệu</i> | 35 |
| CHƯƠNG III: TRIỀN KHAI VÀ CHẠY THỬ NGHIỆM | 39 |
| 3.1. MÔI TRƯỜNG TRIỀN KHAI | 39 |
| 3.2. KẾT QUẢ DEMO VÀ HƯỚNG DẪN SỬ DỤNG CHƯƠNG TRÌNH | 41 |
| KẾT LUẬN | 51 |
| 1. KẾT QUẢ ĐẠT ĐƯỢC | 51 |
| 2. NHỮNG VẤN ĐỀ CHUA GIẢI QUYẾT | 51 |
| 3. HƯỚNG PHÁT TRIỂN | 52 |
| TÀI LIỆU THAM KHẢO | 53 |

DANH SÁCH CÁC BẢNG

| | |
|--|----|
| Bảng 2.1: Đặc tả ca sử dụng đăng ký | 26 |
| Bảng 2.2: Đặc tả ca sử dụng đăng nhập | 27 |
| Bảng 2.3: Đặc tả ca sử dụng quản lý tài khoản | 27 |
| Bảng 2.4: Đặc tả ca quản lý các đơn hàng | 28 |
| Bảng 2.5: Đặc tả ca sử dụng quản lý mua hàng | 28 |
| Bảng 2.6: Đặc tả ca sử dụng quản lý thêm danh mục | 29 |
| Bảng 2.7: Đặc tả ca sử dụng quản lý thêm sản phẩm | 29 |
| Bảng 2.8: Đặc tả ca sử dụng quản lý người dùng | 30 |
| Bảng 2.9: Đặc tả ca sử dụng quản lý thêm thương hiệu | 30 |
| Bảng 2.10: Đặc tả ca sử dụng quản lý thêm phiếu giảm giá | 31 |
| Bảng 2.11: Đặc tả ca sử dụng quản lý đơn hàng | 31 |
| Bảng 2.12: Bảng dữ liệu User | 35 |
| Bảng 2.13: Bảng dữ liệu Voucher | 36 |
| Bảng 2.14: Bảng dữ liệu Cart | 36 |
| Bảng 2.15: Bảng dữ liệu Product | 36 |
| Bảng 2.16: Bảng dữ liệu Order | 37 |
| Bảng 2.17: Bảng dữ liệu Favorite | 37 |
| Bảng 2.18: Bảng dữ liệu Comment | 38 |
| Bảng 2.19: Bảng dữ liệu Contact | 38 |
| Bảng 2.20: Bảng dữ liệu Category | 38 |
| Bảng 2.21: Bảng dữ liệu Brand | 38 |

DANH SÁCH CÁC HÌNH VẼ

| | |
|---|----|
| Hình 1.1: Mô hình MVC | 5 |
| Hình 1.2: Cấu trúc tổ chức thư mục và file trong Nodejs. | 6 |
| Hình 1.3: Mô tả RESTful API | 7 |
| Hình 1.4: Các phương thức HTTP của RESTful API | 8 |
| Hình 1.5: Hoạt động của RESTful API | 8 |
| Hình 1.6: Jordan Walke, người sáng lập ra ReactJS | 10 |
| Hình 1.7: Sơ đồ vòng đời của component trong ReactJS | 12 |
| Hình 1.8: Công cụ draw.io | 15 |
| Hình 1.9: Công cụ Editor VSCode | 15 |
| Hình 1.10: Công cụ Postman | 16 |
| Hình 2.1: Sơ đồ ca sử dụng tổng quát của hệ thống | 21 |
| Hình 2.2: Sơ đồ ca sử dụng quản lý giỏ hàng | 22 |
| Hình 2.3: Sơ đồ ca sử dụng quản lý tài khoản trong hệ thống | 22 |
| Hình 2.4: Sơ đồ ca sử dụng quản lý đơn hàng | 22 |
| Hình 2.5: Sơ đồ ca sử dụng quản lý bình luận, đánh giá, yêu thích | 23 |
| Hình 2.6: Sơ đồ ca sử dụng quản lý mua hàng | 23 |
| Hình 2.7: Sơ đồ ca sử dụng quản lý người dùng trong hệ thống | 23 |
| Hình 2.8: Sơ đồ ca sử dụng quản lý sản phẩm | 24 |
| Hình 2.9: Sơ đồ ca sử dụng quản lý danh mục | 24 |
| Hình 2.10: Sơ đồ ca sử dụng quản lý thương hiệu | 25 |
| Hình 2.11: Sơ đồ ca sử dụng quản lý giảm giá | 25 |
| Hình 2.12: Sơ đồ ca sử dụng quản lý đơn hàng | 26 |
| Hình 2.13: Sơ đồ hoạt động của khách hàng | 32 |
| Hình 2.14: Sơ đồ hoạt động của quản trị viên | 33 |
| Hình 2.15: Sơ đồ kiến trúc hệ thống | 33 |
| Hình 2.16: Sơ đồ quan hệ cơ sở dữ liệu | 35 |
| Hình 3.1: Node.js | 39 |
| Hình 3.2: Trình quản lý thư viện npm | 39 |
| Hình 3.3: Ngôn ngữ back-end sử dụng | 40 |
| Hình 3.4: Ngôn ngữ front-end sử dụng | 40 |
| Hình 3.5: Hệ quản trị cơ sở dữ liệu | 40 |
| Hình 3.6: Môi trường vercel deploy bản production | 41 |
| Hình 3.7: Màn hình chính của hệ thống | 42 |
| Hình 3.8: Màn hình đăng ký | 42 |

| | |
|--|----|
| Hình 3.9: Màn hình đăng nhập hệ thống | 43 |
| Hình 3.10: Màn hình tìm kiếm sản phẩm | 43 |
| Hình 3.11: Màn hình xem chi tiết sản phẩm | 44 |
| Hình 3.12: Màn hình xem đánh giá sản phẩm | 44 |
| Hình 3.13: Màn hình khi nhấp vào icon giỏ hàng trên header | 45 |
| Hình 3.14: Màn hình khi nhấp vào icon giỏ hàng trên header | 45 |
| Hình 3.15: Màn hình trang mua hàng | 46 |
| Hình 3.16: Màn hình phương thức thanh toán bằng momo | 46 |
| Hình 3.17: Màn hình đơn hàng của bạn | 47 |
| Hình 3.18: Màn hình xem chi tiết đơn hàng | 47 |
| Hình 3.19: Màn hình quản lý thống kê | 48 |
| Hình 3.20: Màn hình quản lý sản phẩm | 48 |
| Hình 3.21: Màn hình quản lý danh mục | 49 |
| Hình 3.22: Màn hình quản lý thương hiệu | 49 |
| Hình 3.23: Màn hình quản lý đơn hàng | 50 |
| Hình 3.24: Màn hình quản lý chi tiết đơn hàng | 50 |

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

| Từ | Viết tắt của | Điễn giải |
|---------|--------------------------------------|---|
| API | Application Programming Interface | Giao diện lập trình ứng dụng |
| RESTful | REpresentational State Transfer | Một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API |
| HTTP | Hypertext Transfer Protocol | Giao thức truyền tải siêu văn bản |
| MVC | Model-View-Controller | Mô hình - Giao diện - Bộ điều khiển |
| DB | Database | Cơ sở dữ liệu |
| HTML | HyperText Markup Language | Ngôn ngữ đánh dấu siêu văn bản |
| SGML | Standard Generalized Markup Language | Một hệ thống tổ chức và gắn thẻ yếu tố của một tài liệu |
| CSS | Cascading Style Sheets | Điều khiển cách hiện thị nội dung trang web |
| W3C | World Wide Web Consortium | Chuẩn được các nhà thiết kế website sử dụng làm thước đo khi thiết kế các website |
| UI | User Interface | Giao diện người dùng |
| DOM | Document Object Model | Mô hình đối tượng tài liệu |
| SEO | Search Engine Optimization | Tối ưu hóa công cụ tìm kiếm |
| JSX | JavaScript XML | Một loại cú pháp mở rộng dành cho ngôn ngữ JavaScript viết theo kiểu XML |
| SQL | Structured Query Language | Ngôn ngữ truy vấn cấu trúc |
| JSON | JavaScript Object Notation | Ký hiệu đối tượng JavaScript |

Framework: là một thư viện các lớp đã được xây dựng hoàn chỉnh, bộ khung để phát triển các Phần mềm ứng dụng. Có thể ví Framework như tập các “ Vật liệu” ở từng lĩnh vực cho người lập trình viên, thay vì họ phải mất nhiều thời gian để tự thiết kế trước khi dùng. Do vậy, người lập trình viên chỉ cần tìm hiểu và khai thác các vật liệu này rồi thực hiện để gắn kết chúng lại với nhau, tạo ra sản phẩm.

MỞ ĐẦU

1. Lý do chọn đề tài

Với sự phát triển không ngừng của kinh tế và công nghệ, thói quen mua sắm của người tiêu dùng đang dần thay đổi. Ngày nay, người tiêu dùng không chỉ mua sắm tại các cửa hàng truyền thống mà còn ưu tiên lựa chọn mua sắm trực tuyến vì sự tiện lợi và nhanh chóng. Trong bối cảnh này, việc xây dựng các hệ thống bán hàng trực tuyến trở nên vô cùng cần thiết và phổ biến.

Nước hoa là một mặt hàng xa xỉ, có nhu cầu cao và đa dạng về chủng loại. Người tiêu dùng thường mong muốn tiếp cận các sản phẩm nước hoa từ nhiều thương hiệu nổi tiếng trên thế giới mà không cần phải đến các cửa hàng trực tiếp. Việc mua sắm nước hoa trực tuyến giúp người tiêu dùng dễ dàng tìm kiếm, so sánh giá cả và lựa chọn sản phẩm phù hợp với sở thích và ngân sách của mình.

Hiện nay, tại Việt Nam, thị trường bán lẻ nước hoa đang phát triển mạnh mẽ. Tuy nhiên, nhiều cửa hàng bán nước hoa vẫn chưa tận dụng tối đa các lợi ích mà công nghệ thông tin và internet mang lại. Các cửa hàng này vẫn chủ yếu kinh doanh theo phương thức truyền thống, chưa chú trọng đến việc xây dựng các kênh bán hàng trực tuyến. Điều này dẫn đến hạn chế trong việc tiếp cận khách hàng tiềm năng và mở rộng thị trường.

Trong bối cảnh công nghệ thông tin và internet ngày càng phát triển, việc xây dựng một hệ thống website bán nước hoa trực tuyến không chỉ giúp các doanh nghiệp mở rộng phạm vi kinh doanh, tiếp cận được nhiều khách hàng hơn mà còn giúp tối ưu hóa quy trình bán hàng, tiết kiệm chi phí và nâng cao hiệu quả kinh doanh.

Vì những lý do trên, em đã quyết định chọn đề tài "**Xây dựng website quản lý bán nước hoa**" làm đề tài tốt nghiệp của mình. Hy vọng rằng hệ thống này sẽ mang lại nhiều tiện ích cho cả người tiêu dùng và doanh nghiệp, góp phần thúc đẩy sự phát triển của thị trường bán lẻ nước hoa tại Việt Nam.

2. Mục đích của đề tài

"**Xây dựng website quản lý bán nước hoa**" là tạo ra một hệ thống trực tuyến đơn giản, thân thiện và dễ sử dụng, chạy trên nền tảng web. Mục tiêu là tạo ra một cầu nối giữa người tiêu dùng và người bán nước hoa, mang lại nhiều lợi ích đối với cả hai bên.

Hệ thống này cho phép người dùng đăng ký tài khoản để có thể mua hàng trực tuyến từ bất kỳ đâu và bất kỳ khi nào, giúp tiết kiệm thời gian và công sức. Đồng thời, nó cũng tạo ra một môi trường trực tuyến, nơi mà người tiêu dùng và người bán có thể tiếp cận và giao dịch một cách thuận lợi và an toàn.

Ngoài ra, hệ thống cung cấp các thông tin chi tiết về sản phẩm nước hoa, từ mô tả, hình ảnh đến đánh giá từ người dùng khác, giúp người tiêu dùng có trải nghiệm mua sắm thông tin và dễ dàng quyết định mua hàng.

3. Nội dung của đề tài

Những nội dung công việc cần thực hiện để hoàn thành đề tài bao gồm:

- Tìm hiểu về thị trường bán nước hoa trực tuyến, đánh giá nhu cầu của người tiêu dùng, và các đối thủ cạnh tranh.

- Tìm hiểu và phân tích UML.
- Phân tích và thiết kế hệ thống thông tin.
- Thiết kế cơ sở dữ liệu cho hệ thống.
- Tìm hiểu và nghiên cứu, áp dụng các ngôn ngữ, công cụ lập trình Javascript Nodejs và ReactJS, cách thức xây dựng Web API theo cơ chế RESTful.
- Nghiên cứu và ứng dụng Mongo làm nền tảng cơ sở dữ liệu.
- Xây dựng ứng dụng với các chức năng chính
- Viết báo cáo.

4. Phạm vi sử dụng đề tài

Ứng dụng được sử dụng trên môi trường internet giúp người hàng có thể tiếp cận dễ dàng với việc mua hàng mà không cần tới tận nơi và người bán có như cầu kinh doanh online.

5. Bộ cục của báo cáo

Báo cáo của đồ án được chia làm 3 chương chính:

- Với chương thứ nhất đi vào tìm hiểu và nghiên cứu cơ sở lý thuyết để thực hiện áp dụng vào đề tài. Tìm hiểu công cụ và ngôn ngữ lập trình.
- Trong chương thứ hai đi vào quá trình khảo sát hiện trạng thực tế, phân tích và thiết kế hệ thống thông tin, xác định các tác nhân chính của hệ thống, thiết kế và xây dựng cơ sở dữ liệu.
- Tiếp nối chương thứ hai, chương thứ ba sẽ đi vào quá trình triển khai hệ thống, thực hiện xây dựng mã nguồn chương trình và chạy thử nghiệm.

Đề tài được đánh giá lại qua phần kết luận, đưa ra các đánh giá kết quả về những vấn đề đã giải quyết, những vấn đề chưa giải quyết được và các giải pháp cho đề cải tiến, khắc phục những vấn đề đó.

CHƯƠNG I: CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG

1.1. Kiến thức cơ bản

a. HTML

Là viết tắt cho HyperText Markup Language, hay là "Ngôn ngữ đánh dấu siêu văn bản", là ngôn ngữ đánh dấu chuẩn được sử dụng để tạo các trang web và ứng dụng web. HTML cho phép người dùng tạo và cấu trúc hóa các thành phần trên một trang web như đoạn văn, tiêu đề, liên kết, trích dẫn, bảng biểu... Đây là nền tảng cơ bản của World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web. HTML được định nghĩa như là một ứng dụng đơn giản của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium duy trì. HTML được phát triển bởi Tim Berners-Lee vào cuối những năm 1980 và đầu những năm 1990 khi ông làm việc tại CERN. Phiên bản đầu tiên của HTML được phát hành vào năm 1991. Từ đó, HTML đã phát triển qua nhiều phiên bản, mỗi phiên bản mang lại các tính năng và cải tiến mới. Các phiên bản chính bao gồm HTML 2.0 (1995), HTML 3.2 (1997), HTML 4.01 (1999), XHTML (2000), và HTML5 (2014).

b. CSS

Là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để mô tả cách trình bày của các tài liệu viết bằng HTML hoặc XML. Được phát triển bởi W3C, CSS giúp tách biệt nội dung và hình thức của trang web, cho phép nhà phát triển dễ dàng thay đổi bố cục, màu sắc, phông chữ và các yếu tố hình ảnh khác mà không cần chỉnh sửa mã HTML. Nhờ CSS, các trang web có thể trở nên đẹp mắt, linh hoạt và dễ duy trì hơn, đồng thời hỗ trợ thiết kế web đáp ứng, giúp trang web hiển thị tốt trên mọi loại thiết bị, từ máy tính để bàn đến điện thoại di động.

c. Javascript

Là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu. Nó được sử dụng để thêm tính năng tương tác và động vào các trang web, cho phép người dùng tương tác với nội dung và tạo ra trải nghiệm web đa dạng. JavaScript được tích hợp trực tiếp vào các trang web và có thể được sử dụng để thực hiện nhiều công việc khác nhau, từ thay đổi nội dung của trang đến giao tiếp với máy chủ và xử lý dữ liệu người dùng. Ngôn ngữ này được dùng rộng rãi cho các trang web, nhưng cũng được dùng để tạo khả năng viết script sử dụng các đối tượng nằm

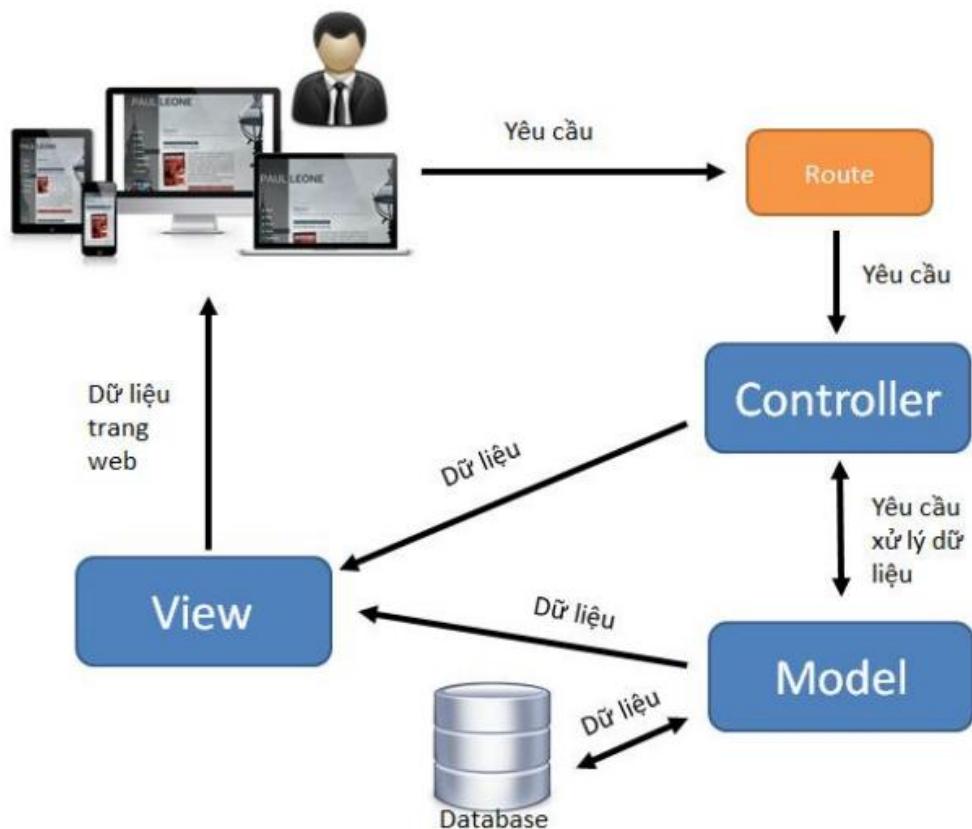
sẵn trong các ứng dụng. Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java.js là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.

1.2. Tổng quan về Node.js và Express.js framework

Node.js là một nền tảng phát triển mã nguồn mở dựa trên Chrome's V8 JavaScript engine, cho phép sử dụng JavaScript để xây dựng các ứng dụng phía server (server-side). Với hiệu suất cao và thời gian đáp ứng nhanh, Node.js là lựa chọn lý tưởng cho việc xây dựng các ứng dụng yêu cầu xử lý không đồng bộ. Node.js cũng hỗ trợ mô hình non-blocking và event-driven, giúp tối ưu hóa hiệu suất và khả năng mở rộng của ứng dụng. Node.js có thể chạy trên nhiều nền tảng hệ điều hành, tạo điều kiện thuận lợi cho việc phát triển và triển khai ứng dụng trên môi trường phát triển mong muốn.

❖ Mô hình MVC trong Node.js

MVC là viết tắt của Model – View – Controller. Là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng để tổ chức mã nguồn thành ba phần: Model, View, và Controller. Nó phân bố source code thành 3 phần, mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác. Để từ đó chúng ta có thể giải quyết các công việc một cách nhanh gọn và dễ dàng hơn.



Hình 1.1: Mô hình MVC

– Thành phần:

- + **Controller:** Giữ nhiệm vụ nhận các yêu cầu từ người dùng và quản lý việc điều hướng chúng đến các phương thức xử lý tương ứng... Chẳng hạn thành phần này sẽ nhận request từ url và form để thao tác trực tiếp với Model và render View. Nó giúp tách biệt logic xử lý và giao diện người dùng, giúp mã nguồn dễ bảo trì và mở rộng.
- + **Model:** Đây là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất cơ sở dữ liệu và định nghĩa cấu trúc dữ liệu như các Class, hàm xử lý...
- + **View:** Là thành phần chịu trách nhiệm hiển thị giao diện cho ứng dụng, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như textbox, images... Hiểu một cách đơn giản, nó là tập hợp các form hoặc các file HTML.

❖ Express.js

Đây là một framework mã nguồn mở và miễn phí cho Node.js. Express.js được sử dụng trong thiết kế và xây dựng các ứng dụng web một cách đơn giản và nhanh

chóng. Express.js được sử dụng để phát triển các ứng dụng web theo kiến trúc Model-View-Controller (MVC). Với cú pháp dễ hiểu và rõ ràng, Express cho phép bạn dễ dàng định nghĩa các tuyến đường (routes) và điều hướng các yêu cầu HTTP đến các xử lý tương ứng. Hệ thống middleware mạnh mẽ của Express cung cấp khả năng thêm các chức năng trung gian vào ứng dụng, từ xác thực đến xử lý lỗi, tạo điều kiện cho việc tăng cường tính linh hoạt và bảo mật.

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows a project structure with folders like `backend`, `controllers`, `models`, `node_modules`, `routes`, `utils`, `.env`, `.gitignore`, and files `index.js`, `package-lock.json`, `package.json`, and `vercel.json`.
- Code Editor:** Displays the content of `index.js` which is a Node.js application starting with mongoose and express, connecting to a MongoDB database, and setting up routes and port.
- Terminal:** Shows the command `DATN` at the top.

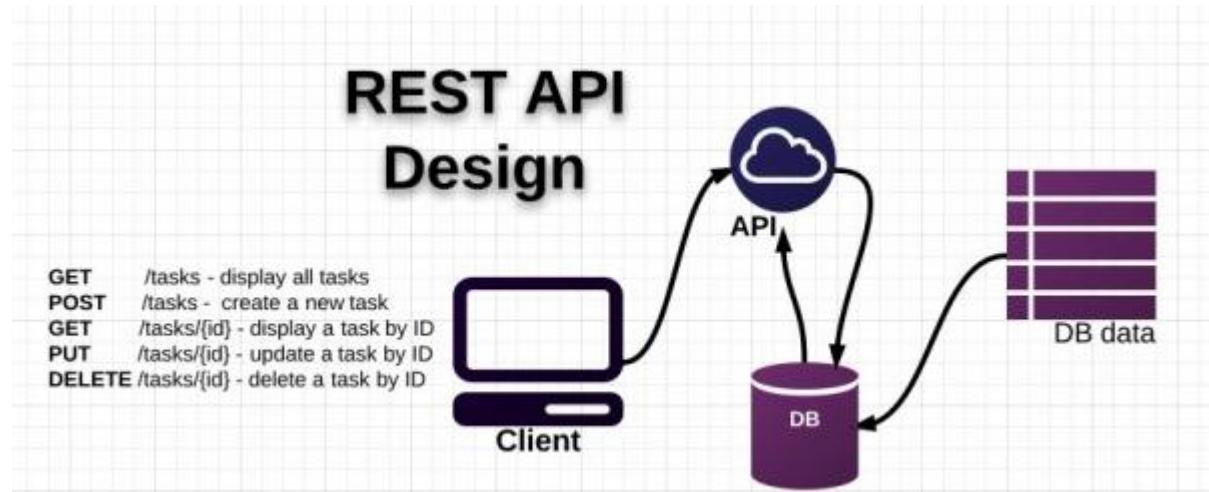
```
backend > index.js > ...
1  const mongoose = require("mongoose");
2  const express = require("express");
3  const dotenv = require("dotenv");
4  const router = require("./routes/index");
5  const morgan = require("morgan");
6  const cookieParser = require("cookie-parser");
7  const cors = require("cors");
8  require("dotenv").config();
9  const errorHandler = require("./utils/errorHandler");
10
11 mongoose
12   .connect(process.env.DATABASE, {})
13   .then(() => {
14     console.log("connected to MongoDB");
15   })
16   .catch(error => {
17     console.log("error connecting to MongoDB:", error.message);
18   });
19 const app = express();
20 app.use(cors());
21 app.use(express.json());
22 app.use(cookieParser());
23 app.use(morgan("dev"));
24 app.use("/", router);
25 const port = process.env.PORT;
26 app.use(errorHandler);
27 const server = app.listen(port, () => {
28   console.log(`http://localhost:${port}`);
29 });
30
31 module.exports = app;
```

Hình 1.2: Cấu trúc tổ chức thư mục và file trong Nodejs.

1.3. Tổng quan về RESTful API

❖ Định nghĩa:

RESTful API (Representational State Transferful API) là một tiêu chuẩn thiết kế API cho các ứng dụng web (web services) nhằm quản lý các tài nguyên của hệ thống. Tiêu chuẩn này tập trung vào việc xem các thành phần của hệ thống như là các tài nguyên, bao gồm các loại như tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động. RESTful API sử dụng các phương thức và trạng thái của giao thức HTTP để truyền tải và định dạng các trạng thái của các tài nguyên này.



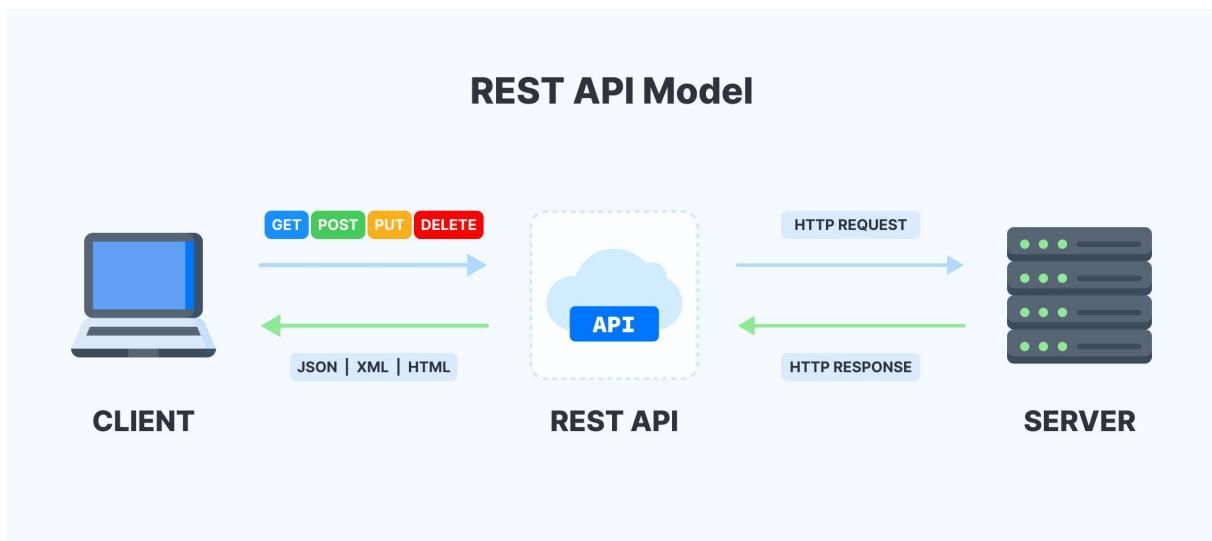
Hình 1.3: Mô tả RESTful API

API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, PUT, DELETE, vv đến một URL để xử lý dữ liệu.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của **REST** là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một **RESTful API**.



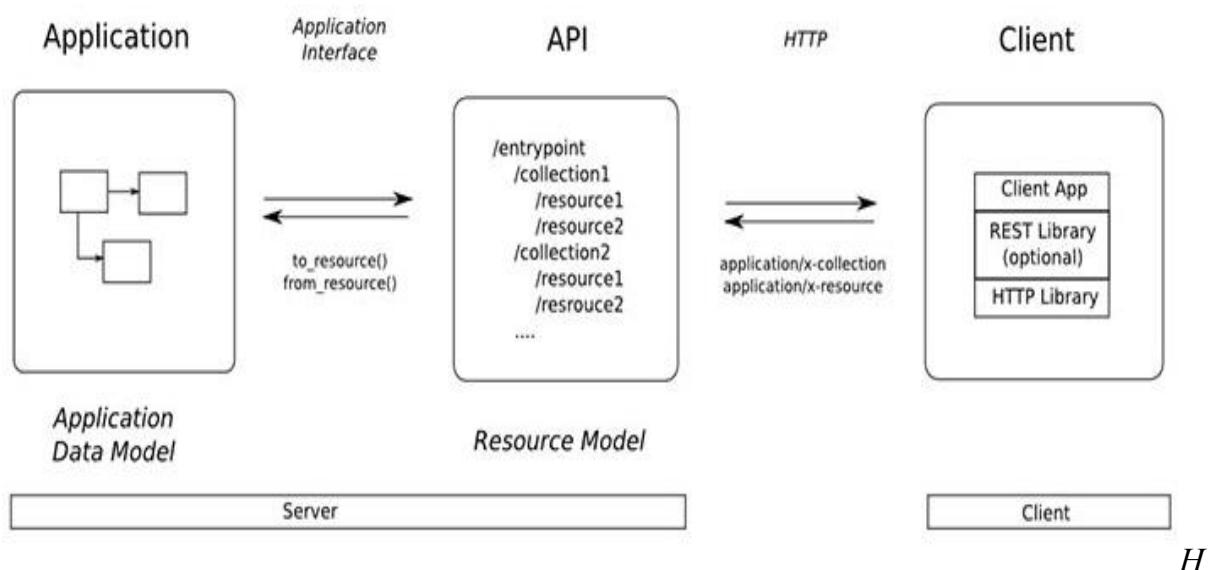
Hình 1.4: Các phương thức HTTP của RESTful API

❖ Hoạt động

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng 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): Xoá một Resource.

Những phương thức hay hoạt động này thường được gọi là **CRUD** tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.



Hình 1.5: Hoạt động của RESTful API

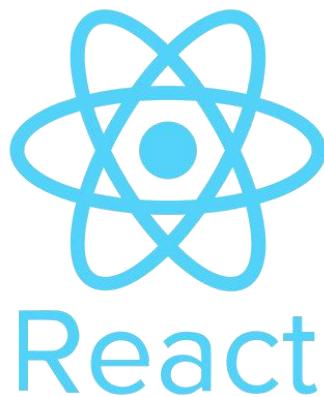
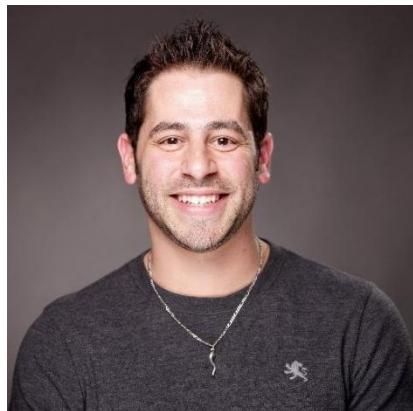
RESTful API hoạt động bằng cách xác định các tài nguyên, định nghĩa các phương thức cho mỗi tài nguyên, triển khai các endpoint tương ứng, xử lý yêu cầu từ client, và trả về phản hồi tương ứng.

Sau quá trình gọi API việc trả về những mã trạng thái cụ thể khi nhận và phiên dịch một HTTP Request, Server sẽ gửi tín hiệu phản hồi là một HTTP Response là một trong số các loại trạng thái sau:

- 200 OK – Trả về thành công cho những phương thức GET, PUT, PATCH hoặc DELETE.
- 201 Created – Trả về khi một Resource vừa được tạo thành công.
- 204 No Content – Trả về khi Resource xóa thành công.
- 304 Not Modified – Client có thể sử dụng dữ liệu cache.
- 400 Bad Request – Request không hợp lệ
- 401 Unauthorized – Request cần có quyền.
- 403 Forbidden – bị từ chối không cho phép.
- 404 Not Found – Không tìm thấy resource từ URI
- 405 Method Not Allowed – Phương thức không cho phép với user hiện tại.
- 410 Gone – Resource không còn tồn tại, Version cũ đã không còn hỗ trợ.
- 415 Unsupported Media Type – Không hỗ trợ kiểu Resource này.
- 422 Unprocessable Entity – Dữ liệu không được xác thực
- 429 Too Many Requests – Request bị từ chối do bị giới hạn

1.4. Tổng quan về ReactJS

React (Hay ReactJS, React.js) là một thư viện Javascript mã nguồn mở để xây dựng các thành phần giao diện có thể tái sử dụng. Nó được tạo ra bởi *Jordan Walker*, một kỹ sư phần mềm tại Facebook. Người bị ảnh hưởng bởi XHP (Một nền tảng thành phần HTML cho PHP). React lần đầu tiên được triển khai cho ứng dụng Newsfeed của Facebook năm 2011, sau đó được triển khai cho Instagram.com năm 2012. Nó được mở mã nguồn (open-sourced) tại JSConf US tháng 5 năm 2013. React hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được.



Hình 1.6: Jordan Walke, người sáng lập ra ReactJS

– **Một số khái niệm cơ bản:**

- + **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 hóa việc re-render DOM tree thật.
- + **Giới thiệu về JSX:** JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javascript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript. 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. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascript có thể sử dụng.

– **Đặc tính của ReactJS:**

- + Tư tưởng của ReactJS là xây dựng lên các **components** có tính tái sử dụng, dễ dàng cho việc chia nhỏ vấn đề, testing. Nó giúp chúng ta dễ dàng quản lý, mở rộng hệ thống.
- + ReactJS luôn giữ các components ở trạng thái stateless (nhiều nhất có thể) khiến ta dễ dàng quản lý bởi nó chẳng khác gì một trang static HTML. Bản thân các components này không có trạng thái (state) nó nhận đầu vào từ bên ngoài và

chỉ hiển thị ra dựa vào các đầu vào đó, điều này giải thích tại sao nó mang tính tái sử dụng và dễ dàng cho việc test.

❖ Các đặc trưng của ReactJS:

- **Component:** React được xây dựng xung quanh các component, 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.

Đặc điểm:

- + Components được thiết kế để có thể tái sử dụng trong nhiều phần của ứng dụng. Điều này giúp giảm lặp lại mã và làm cho mã nguồn dễ bảo trì hơn.
- + Mỗi Component tồn tại độc lập với các Component khác, điều này giúp tách biệt logic và hiển thị, làm cho mã nguồn trở nên dễ đọc và dễ hiểu hơn.
- + Mỗi Component phải cung cấp một phương thức render() để mô tả giao diện của nó. Phương thức này trả về một cây DOM (Document Object Model) hoặc một cây React Elements để hiển thị nội dung của Component.
- **Props:** viết tắt của "Properties" (Thuộc tính). Là một cách để truyền dữ liệu từ Component cha đến Component con. Chúng đóng vai trò như các tham số được truyền cho một Component khi nó được sử dụng trong một cấu trúc JSX. Props cho phép bạn truyền dữ liệu tĩnh từ một Component đến một Component khác, giúp tái sử dụng các Component và làm cho mã nguồn trở nên linh hoạt và dễ bảo trì hơn.

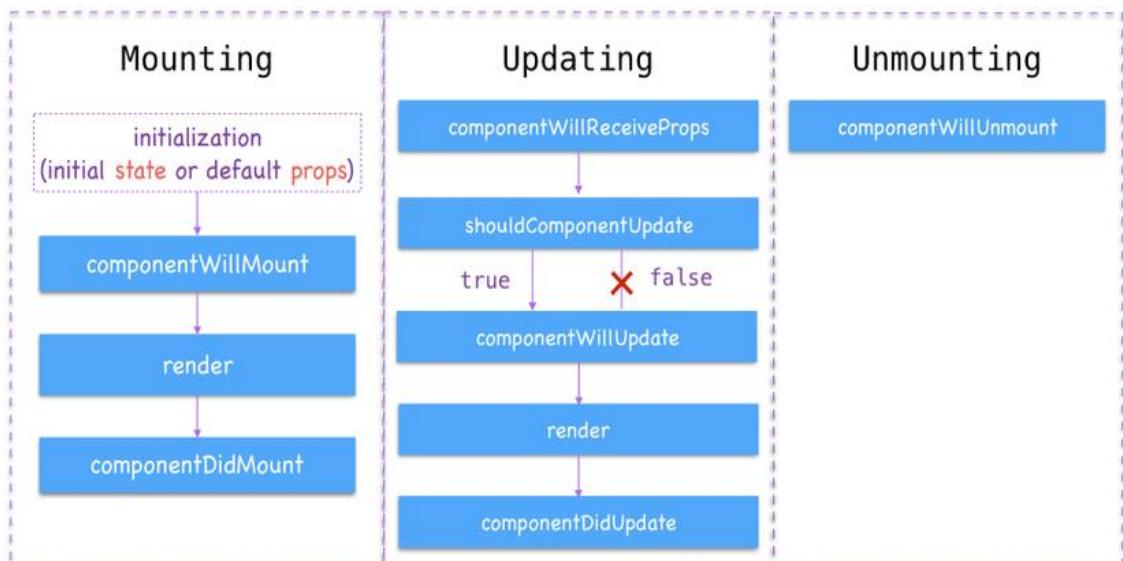
Đặc điểm:

- + Props cho phép bạn truyền dữ liệu từ Component cha xuống Component con thông qua các thuộc tính.
- + Props được xem như là "read-only" trong Component con, có nghĩa là chúng không thể được thay đổi bởi Component con.
- + Props có thể được truyền từ một Component cha sang một hoặc nhiều Component con, và chúng có thể được truyền qua nhiều cấp độ lồng nhau trong cây Component của ứng dụng.
- + Props có thể chứa dữ liệu động (được xác định bởi State của Component cha) hoặc dữ liệu tĩnh (được truyền trực tiếp từ Component cha).
- **State:** là một đối tượng chứa các dữ liệu có thể thay đổi trong một Component. Nó là một trong những khái niệm quan trọng nhất trong React và được sử dụng để

quản lý trạng thái của các Component. Khi state của một Component thay đổi, React sẽ tự động cập nhật giao diện người dùng để phản ánh trạng thái mới.

Đặc điểm:

- + State chứa dữ liệu mà Component có thể thay đổi trong quá trình thực thi. Ví dụ, nó có thể bao gồm thông tin hiển thị trên giao diện hoặc dữ liệu được lấy từ API.
- + State trong React là mutable, có nghĩa là bạn có thể thay đổi giá trị của nó bằng cách gọi phương thức `setState()` của Component.
- + State thường được sử dụng để quản lý các trạng thái của giao diện người dùng, chẳng hạn như việc mở hoặc đóng một hộp thoại, hiển thị thông báo lỗi, hoặc đổi màu sắc của một thành phần.
- + Khi State của một Component thay đổi, React sẽ tự động cập nhật giao diện người dùng để phản ánh trạng thái mới. Điều này giúp cho việc làm việc với dữ liệu động trở nên dễ dàng và hiệu quả hơn.
- + Một component trong React có 2 cách để lấy thông tin props và state. Không như props, state của 1 component không được truyền từ bên ngoài vào. Một component sẽ tự quyết định state của chính nó. Để tạo ra state cho component, ta sẽ phải khai báo nó trong hàm constructor.
- **Lifecycle:** Một thành phần đặc trưng không kém Props và State của React đó là LifeCycle. LifeCycle là những phương thức được viết sẵn truy cập đến từng thời điểm trong một vòng đời của component, từ đó chúng ta có thể triển khai ý tưởng một cách dễ dàng mà linh động.



Hình 1.7: Sơ đồ vòng đời của component trong ReactJS

+ **constructor(props):**

Hàm này là hàm khởi tạo, là hàm chạy ngay sau khi component được tạo ra, đầu vào của hàm này là props (tức là thuộc tính của component được dùng để truyền dữ liệu từ bên ngoài vào). Trong hàm này chúng ta cũng có thể khởi tạo state cho component.

+ **componentWillMount:**

Trước khi render thì sẽ gọi đến hàm này. Ở đây chúng ta có thể đăng kí các biến toàn cục và có thể dựa vào props tính toán và setState nếu có.

+ **render**

Là hàm bắt buộc, vì đây là đầu ra của component. Nó chịu trách nhiệm trả về một đối tượng JSX những gì chúng ta muốn hiển thị ra bên ngoài.

+ **componentDidMount:**

Là hàm được gọi ngay sau khi giao diện được render xong, có nghĩa là khi giao diện hiện lên sẽ gọi hàm này. Trong hàm này chúng ta có thể dùng để lấy dữ liệu từ api, server... và setState dữ liệu ra.

+ **componentWillReceiveProps(nextProps):**

Hàm này được gọi đến mỗi khi props thay đổi. Do vậy được sử dụng để thay đổi trạng thái của các component phụ thuộc vào props và sử dụng các kết quả, khởi tạo các biến có tính chất bất đồng bộ.

+ **shouldComponentUpdate(nextProps, nextState):**

Hàm này được gọi khi state và props thay đổi và sẽ trả về kết quả true/false, chúng ta sử dụng hàm này để xử lý xem có cần update component hay không. Mặc định hàm này sẽ trả về true.

+ **componentWillUpdate(nextProps, nextState):**

Hàm này được gọi dựa vào kết quả của hàm shouldComponentUpdate, nếu trả về false thì hàm này sẽ không được gọi.

+ **componentDidUpdate(prevProps, prevState):**

Hàm này được gọi ngay sau khi component được render lại.

+ **componentWillUnmount:**

Hàm này được gọi khi component loại bỏ DOM, thực hiện các thao tác dọn dẹp, loại bỏ những phần tử dư thừa và hủy các timer...

❖ **Ưu điểm và nhược điểm của ReactJS:**

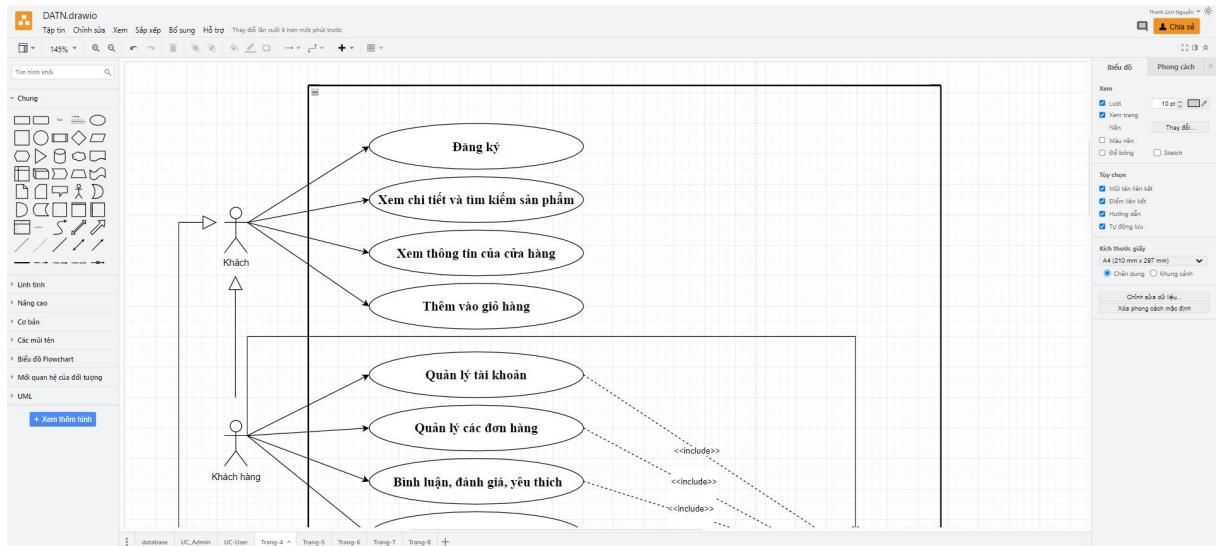
- **Ưu điểm:**

- + ReactJS cực kì hiệu quả: ReactJS tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. ReactJS cũng tính toán những thay đổi nào cần cập nhật lên DOM và chỉ thực hiện chúng. Điều này giúp ReactJS tránh những thao tác cần trên DOM mà nhiều chi phí.
- + ReactJS giúp việc viết các đoạn code JS dễ dàng hơn: Nó dùng cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể thêm vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của Reactjs. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
- + Reactjs có nhiều công cụ phát triển: Khi bạn bắt đầu Reactjs, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho ReactJS. Nó giúp bạn debug code dễ dàng hơn. Sau khi bạn cài đặt ứng dụng này, bạn sẽ có cái nhìn trực tiếp vào virtual DOM như thể bạn đang xem cây DOM thông thường
- + Render tầng server: Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO và thời gian tải trang. Nếu tất cả việc xây dựng và hiển thị trang đều thực hiện ở client, thì người dùng sẽ phải chờ cho trang được khởi tạo và hiển thị lên. Điều này thực tế là chậm. Hoặc nếu giả sử người dùng vô hiệu hóa Javascript thì sao? Reactjs là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về
- + Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.
- + Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi
- **Nhược điểm:**
 - + ReactJS chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax.
 - + Tích hợp ReactJS vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.

- + React khá nặng nề so với các framework khác, React có kích thước tương đương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh.

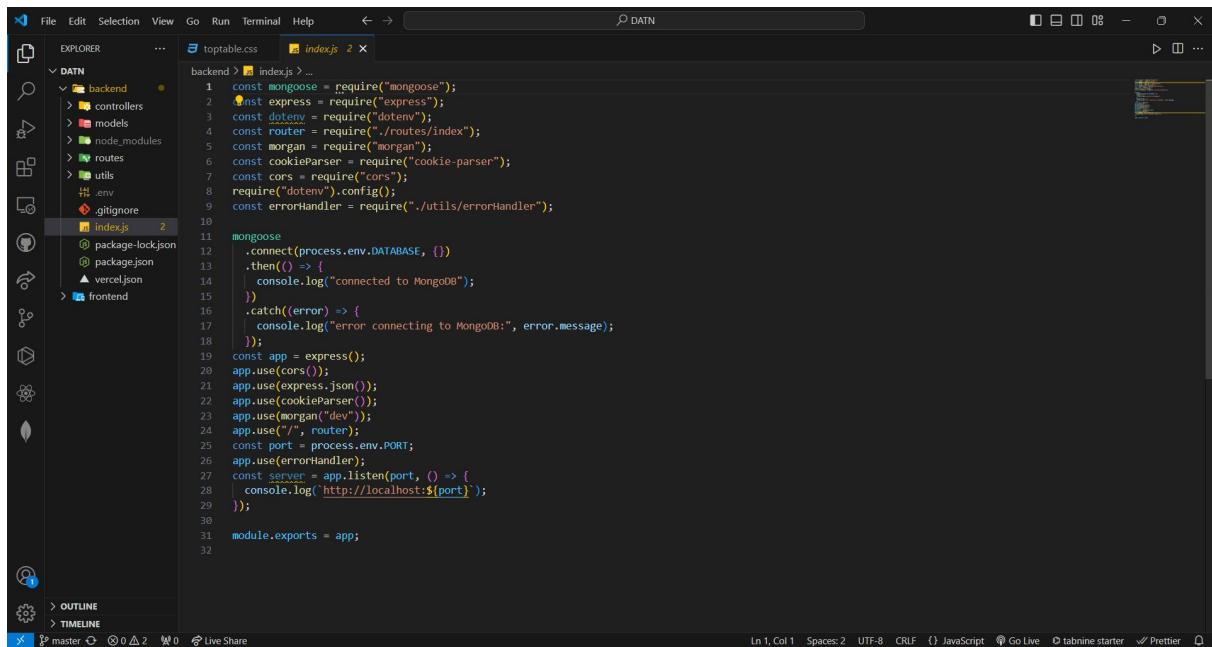
1.5. Công cụ sử dụng

- Quá trình phân tích và thiết kế UML thì công cụ **draw.io** (link truy cập trên website: <https://www.draw.io/>) được sử dụng. Công cụ này giúp chúng ta có thể dễ dàng thao tác và sử dụng để tạo các sơ đồ ca sử dụng, sơ đồ hoạt động,....



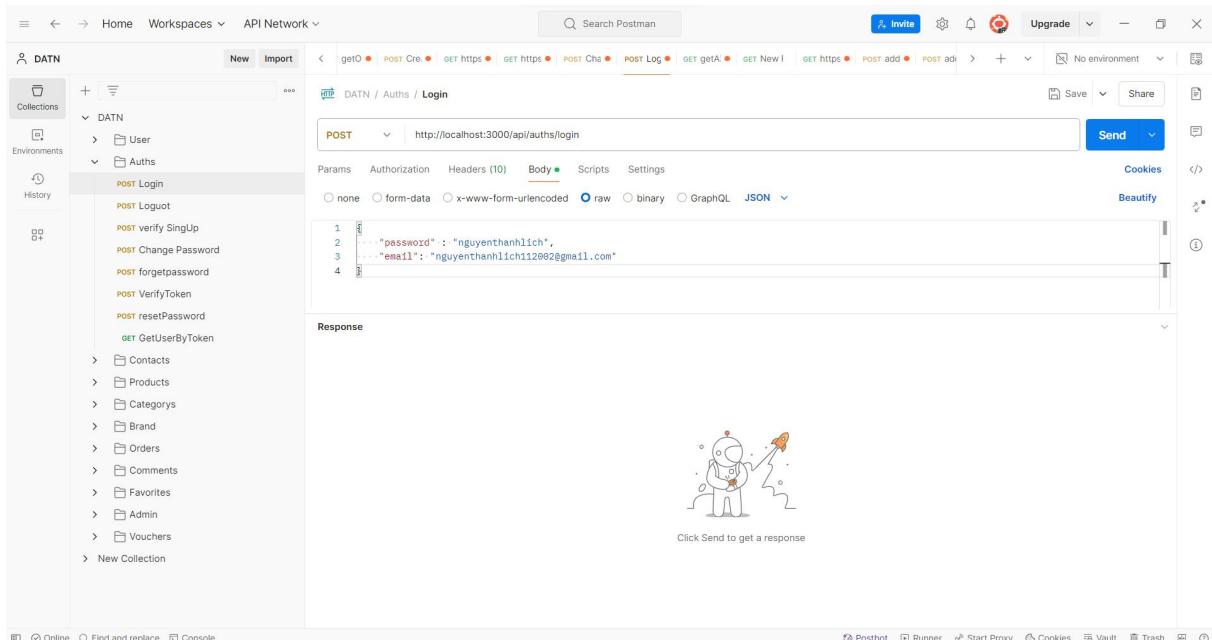
Hình 1.8: Công cụ draw.io

- Quá trình xây dựng mã chương trình, công cụ, editor được dùng là VSCode, đặc điểm nổi bật là đơn giản, gọn nhẹ, dễ dàng cài đặt, hỗ trợ nhiều hệ điều hành cũng như nhiều ngôn ngữ lập trình khác nhau.



Hình 1.9: Công cụ Editor VSCode

- Quá trình kiểm tra, test và chạy thử các API đã được viết chương trình trước khi áp dụng để gọi từ các request ở website, công cụ được sử dụng là Postman.



Hình 1.10: Công cụ Postman

CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. KHẢO SÁT BÀI TOÁN THỰC TẾ

2.1.1. Tổng quan hiện trạng

Hiện tại, việc xây dựng một website quản lý bán nước hoa mang lại nhiều lợi ích đối với cả người bán và người tiêu dùng. Với sự phát triển của công nghệ và sự phổ biến của internet, việc thực hiện giao dịch mua bán trực tuyến trở nên phổ biến và tiện lợi hơn bao giờ hết.

Đối với người bán, website quản lý bán nước hoa giúp họ tiếp cận một phạm vi khách hàng rộng lớn hơn, không giới hạn bởi địa lý. Họ có thể quảng cáo sản phẩm của mình một cách hiệu quả thông qua website, thu hút khách hàng tiềm năng và tăng doanh số bán hàng. Ngoài ra, hệ thống quản lý bán hàng cung cấp các công cụ giúp họ quản lý sản phẩm, đơn hàng, và giao dịch một cách tự động và hiệu quả.

Đối với người tiêu dùng, việc có một website quản lý bán nước hoa mang lại sự tiện lợi và linh hoạt khi mua sắm. Họ có thể dễ dàng duyệt qua các sản phẩm, so sánh giá cả và lựa chọn sản phẩm phù hợp mà không cần phải di chuyển đến cửa hàng truyền thống. Hơn nữa, hệ thống thanh toán trực tuyến an toàn và tiện lợi giúp họ hoàn thành giao dịch một cách nhanh chóng và dễ dàng từ bất kỳ đâu.

Tóm lại, việc xây dựng một website quản lý bán nước hoa mang lại nhiều lợi ích cho cả người bán và người tiêu dùng, đồng thời cũng thúc đẩy sự phát triển của ngành công nghiệp bán lẻ trực tuyến.

2.1.2. Nhu cầu sử dụng nước hoa

Trong những năm gần đây, nhu cầu sử dụng nước hoa đã tăng mạnh, đặc biệt trong giới trẻ. Đây không chỉ là một xu hướng thời trang mà còn trở thành một phần thiết yếu trong phong cách sống hiện đại.

Nhu cầu sử dụng nước hoa trong giới trẻ hiện nay không chỉ là một xu hướng thời trang mà còn phản ánh sự thay đổi trong cách họ chăm sóc và thể hiện bản thân. Với sự đa dạng và phong phú của thị trường nước hoa, cùng với ảnh hưởng mạnh mẽ từ mạng xã hội và người nổi tiếng, việc sử dụng nước hoa đã trở thành một phần không

thể thiêu trong cuộc sống hàng ngày của giới trẻ. Điều này tạo ra cơ hội lớn cho các nhà sản xuất và kinh doanh nước hoa để phát triển và đáp ứng nhu cầu ngày càng tăng của thị trường này.

Với sự phát triển của không gian mạng hiện nay, một trang web bán nước hoa là cách dễ dàng và hiệu quả nhất để tiếp cận với người mua. Qua đó, người tiêu dùng có thể tìm hiểu, lựa chọn và mua sắm sản phẩm một cách thuận tiện, nhanh chóng, mà không cần phải đến cửa hàng trực tiếp. Đồng thời, người bán có thể mở rộng thị trường, tăng doanh số và quảng bá thương hiệu của mình đến một lượng khách hàng rộng lớn hơn thông qua nền tảng trực tuyến.

2.2. PHÂN TÍCH CHỨC NĂNG

Xây dựng website quản lý bán nước hoa cung cấp cho cửa hàng khả năng tiếp cận với người mua hàng một cách dễ dàng hơn trên môi trường internet. Đồng thời, hệ thống giúp người mua dễ dàng tiếp cận được các sản phẩm tốt mà không cần phải đến trực tiếp cửa hàng để mua hàng. Trang web rút ngắn khoảng cách địa lý giữa người bán và người mua, từ đó tăng thêm lợi nhuận cho cửa hàng.

Hệ thống phục vụ cho 3 loại người dùng: Người quản trị, Khách hàng và Khách. Các chức năng theo từng nhóm người dùng sẽ được mô tả dưới đây.

2.2.1. Khách có các chức năng sau

– Chức năng đăng ký

Chức năng này cho phép khách có thể đăng ký tài khoản vào hệ thống để sử dụng.

– Chức năng xem chi tiết và tìm kiếm sản phẩm

Chức năng này cho phép khách hàng có thể xem được chi tiết thông tin của sản phẩm, các đặc điểm, thành phần, và giá cả, đồng thời có thể tìm kiếm sản phẩm theo nhiều tiêu chí khác nhau.

– Chức năng xem thông tin liên quan đến cửa hàng

Chức năng này cho phép khách hàng có thể xem được chi tiết các thông tin liên quan đến cửa hàng như địa chỉ, giờ mở cửa, các chương trình khuyến mãi, và các bài viết liên quan đến sản phẩm.

– Chức năng thêm vào giỏ hàng

Chức năng này cho phép khách hàng có thể thêm sản phẩm vào giỏ hàng của mình, kiểm tra số lượng, và tổng giá trị đơn hàng trước khi tiến hành thanh toán.

2.2.2. Khách hàng có các chức năng sau

– Chức năng đăng nhập/đăng xuất

Chức năng này cho phép khách hàng có thể đăng nhập vào hệ thống để thực hiện các chức năng của mình và đăng xuất khỏi hệ thống khi không còn sử dụng..

– Chức năng quản lý tài khoản

Chức năng này cho phép khách hàng có thể xem và chỉnh sửa các thông tin cá nhân đã đăng ký trong hệ thống như họ tên, email , địa chỉ,... và quản lý bảo mật cá nhân thông qua việc thay đổi, cập nhật mật khẩu đăng nhập.

– Chức năng quản lý các đơn hàng

Chức năng này cho phép khách hàng xem được danh sách các đơn hàng đã đặt, chi tiết đơn hàng, trạng thái giao hàng, và lịch sử mua hàng.

– Chức năng bình luận, đánh giá, yêu thích

Chức năng này cho phép khách hàng có thể bình luận, đánh giá sản phẩm(dành cho những khách hàng đã mua sản phẩm) và thêm sản phẩm vào danh sách yêu thích của mình, giúp cải thiện trải nghiệm mua sắm và chia sẻ ý kiến cá nhân về sản phẩm.

– Chức năng mua hàng

Chức năng này cho phép khách hàng có thể chọn sản phẩm, thêm vào giỏ hàng và thực hiện thanh toán trực tuyến. Khách hàng cũng có thể xem lại tất cả các đơn hàng đã mua và kiểm tra trạng thái giao hàng của từng đơn hàng.

2.2.3. Quản trị viên có các chức năng sau

– Chức năng quản lý người dùng

Chức năng này cho phép quản trị viên quản lý được tất cả người dùng, các công ty đã đăng ký tồn tại trong hệ thống: Xác nhận, chỉnh sửa thông tin, xóa, khóa một tài khoản bất kì.

– Chức năng quản lý sản phẩm

Chức năng này cho phép quản trị viên tạo mới, chỉnh sửa, xóa sản phẩm trong hệ thống. Quản trị viên cũng có thể cập nhật trạng thái hiển thị của các sản phẩm..

– Chức năng quản lý danh mục

Chức năng này cho phép quản trị viên tạo mới, chỉnh sửa các danh mục sản phẩm..

– Chức năng quản lý thương hiệu

Chức năng này cho phép quản trị viên tạo mới, chỉnh sửa các thương hiệu sản phẩm.

– Chức năng quản lý giảm giá

Chức năng này cho phép quản trị viên tạo mới, chỉnh sửa, xóa các mã giảm giá. Quản trị viên cũng có thể quản lý trạng thái và điều kiện sử dụng của các mã giảm giá.

– **Chức năng quản lý đơn hàng**

Chức năng này cho phép quản trị viên theo dõi và quản lý tất cả các đơn hàng trong hệ thống. Quản trị viên có thể cập nhật trạng thái đơn hàng (đang xử lý, đã thanh toán, đã giao, đã nhận hàng, đã hủy), xem chi tiết đơn hàng, viết lưu ý cho từng đơn hàng.

– **Chức năng quản lý thống kê**

Chức năng quản lý đơn hàng cho phép quản trị viên thống kê doanh thu theo thời gian, xem sản phẩm, danh mục và thương hiệu bán chạy của cửa hàng.

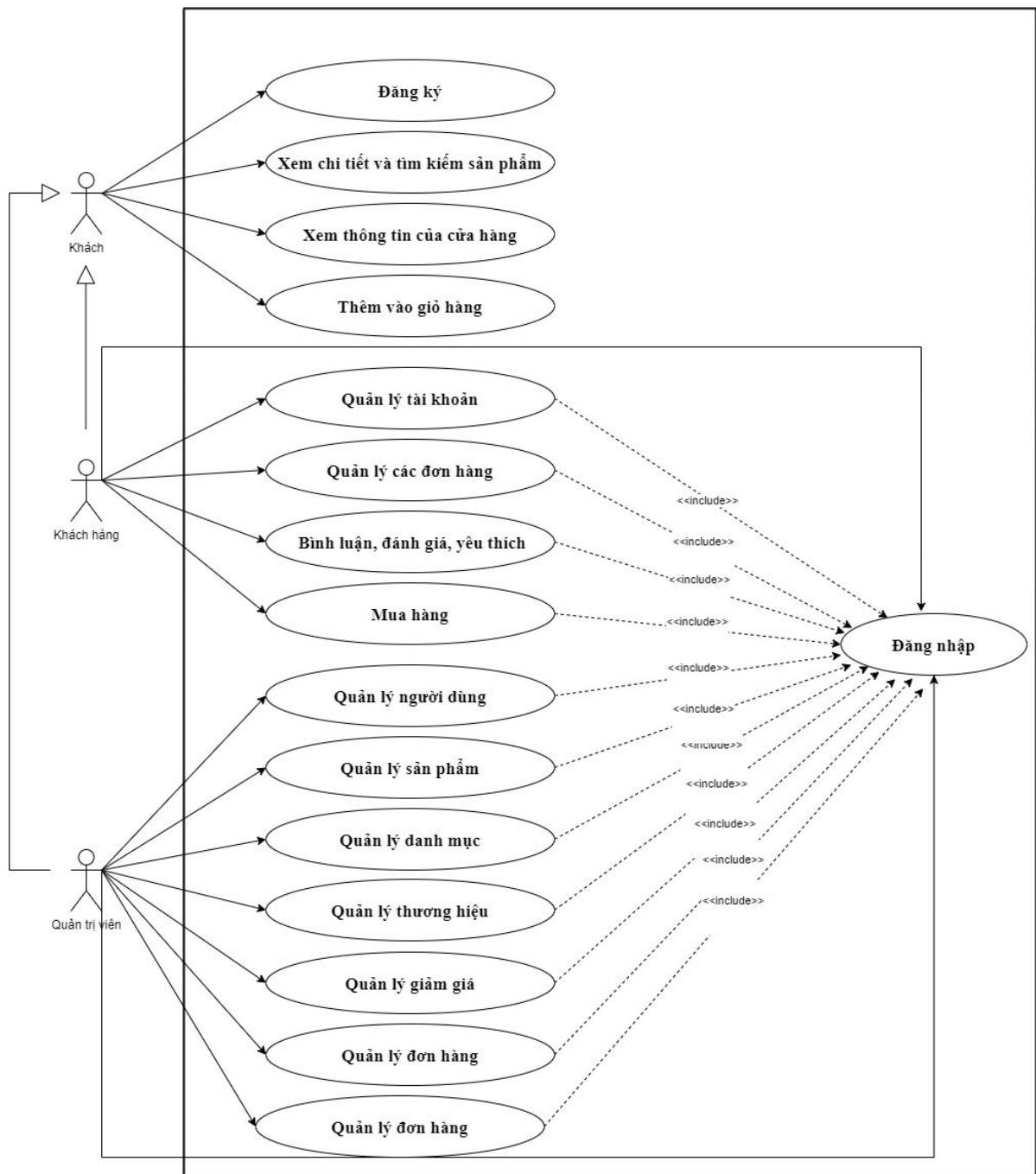
2.3. SƠ ĐỒ CA SỬ DỤNG

2.3.1. Các tác nhân trong hệ thống

Hệ thống có 3 tác nhân là Khách, Quản trị viên và Khách hàng:

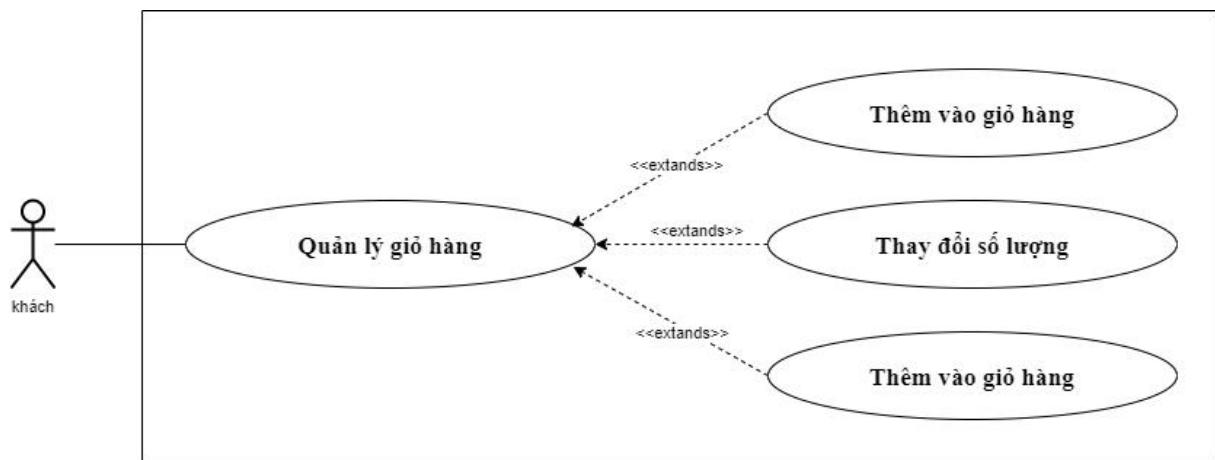
- Khách: là người dùng truy cập vào trang web nhưng chưa có tài khoản. Khách có thể duyệt xem sản phẩm, tìm kiếm thông tin về nước hoa, và xem đánh giá liên quan đến sản phẩm.
- Khách hàng: là người đã đăng ký tài khoản hợp lệ trong hệ thống. Khách hàng có thể đăng nhập vào hệ thống để thực hiện các chức năng như mua hàng, xem chi tiết sản phẩm, thêm sản phẩm vào giỏ hàng, quản lý đơn hàng của mình, đánh giá và bình luận sản phẩm. Khách hàng cũng có thể xem và chỉnh sửa thông tin cá nhân của mình trong hệ thống.
- Quản trị viên: là người quản lý và điều hành hệ thống. Quản trị viên có các chức năng như quản lý người dùng, quản lý sản phẩm, quản lý đơn hàng, quản lý các danh mục sản phẩm, thương hiệu, quản lý các chương trình giảm giá.

2.3.2. Sơ đồ ca sử dụng tổng quát của hệ thống



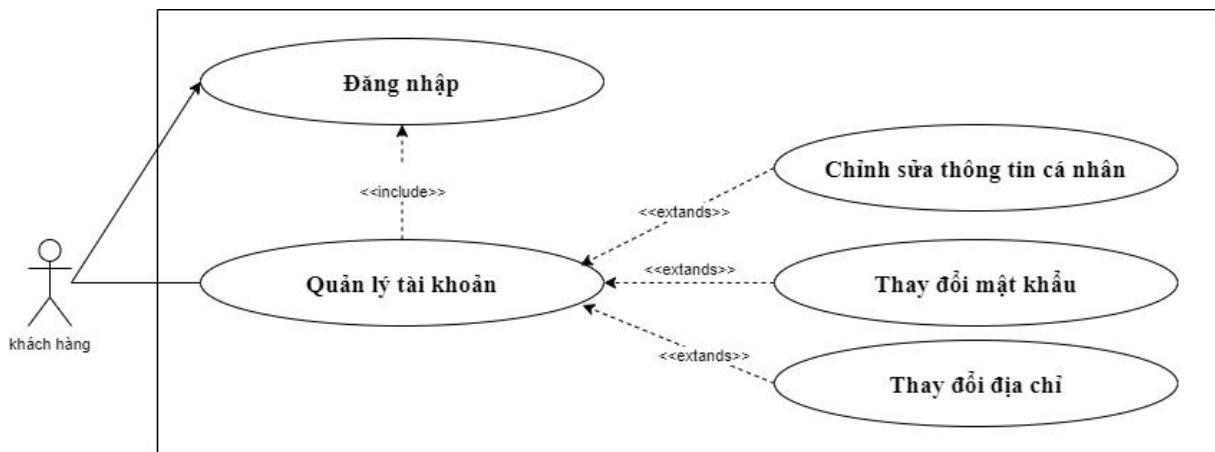
Hình 2.1: Sơ đồ ca sử dụng tổng quát của hệ thống

2.3.3. Sơ đồ ca sử dụng của Khách

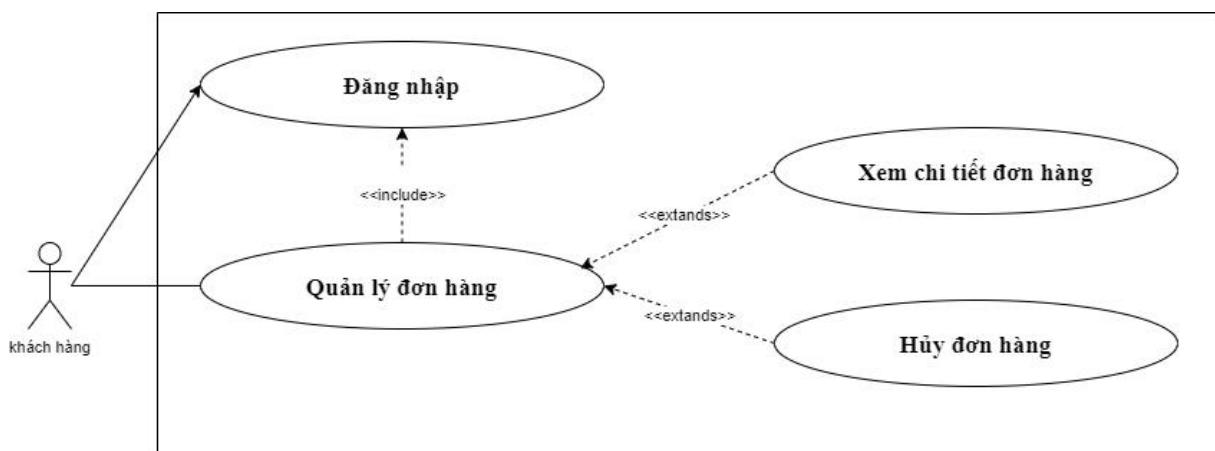


Hình 2.2: Sơ đồ ca sử dụng quản lý giỏ hàng

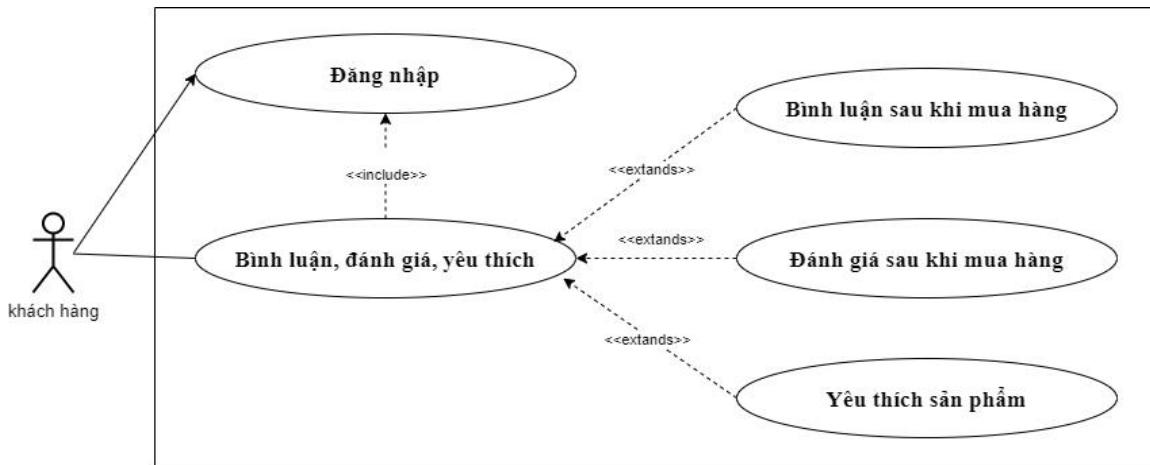
2.3.4. Sơ đồ ca sử dụng của Khách hàng



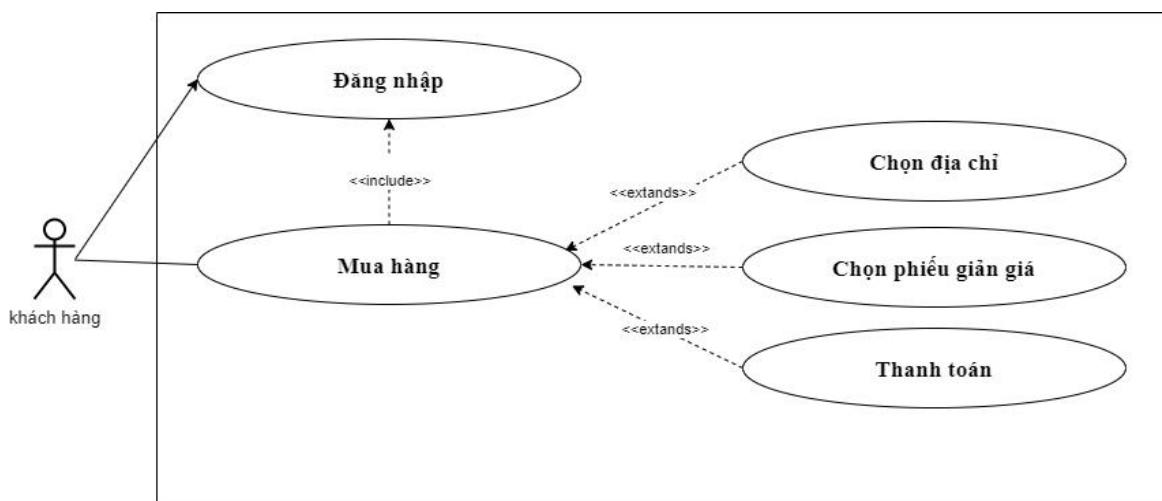
Hình 2.3: Sơ đồ ca sử dụng quản lý tài khoản trong hệ thống.



Hình 2.4: Sơ đồ ca sử dụng quản lý đơn hàng.

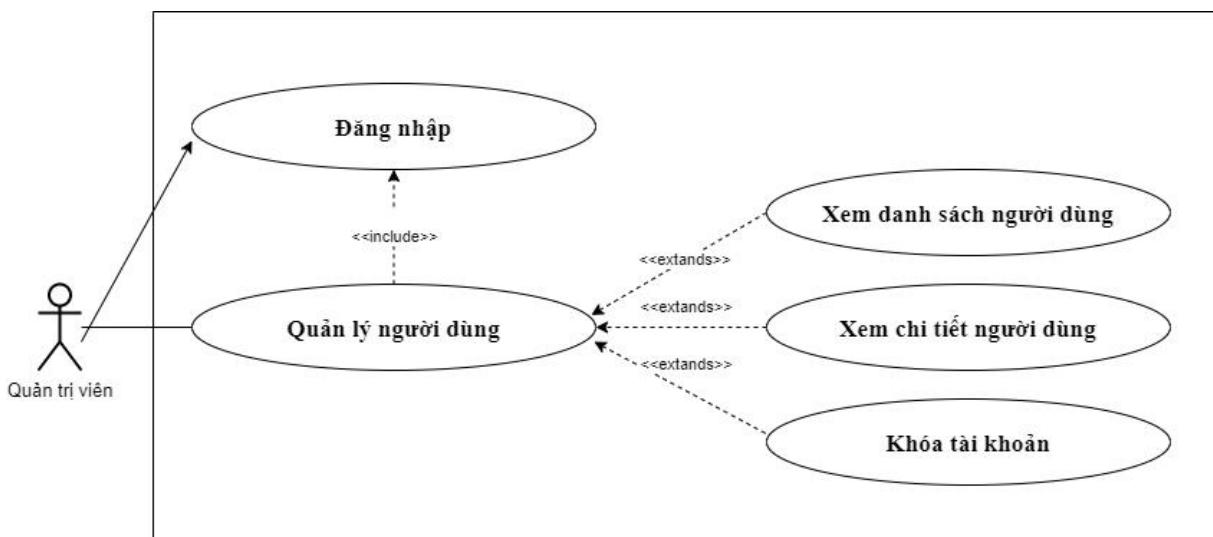


Hình 2.5: Sơ đồ ca sử dụng quản lý bình luận, đánh giá, yêu thích.

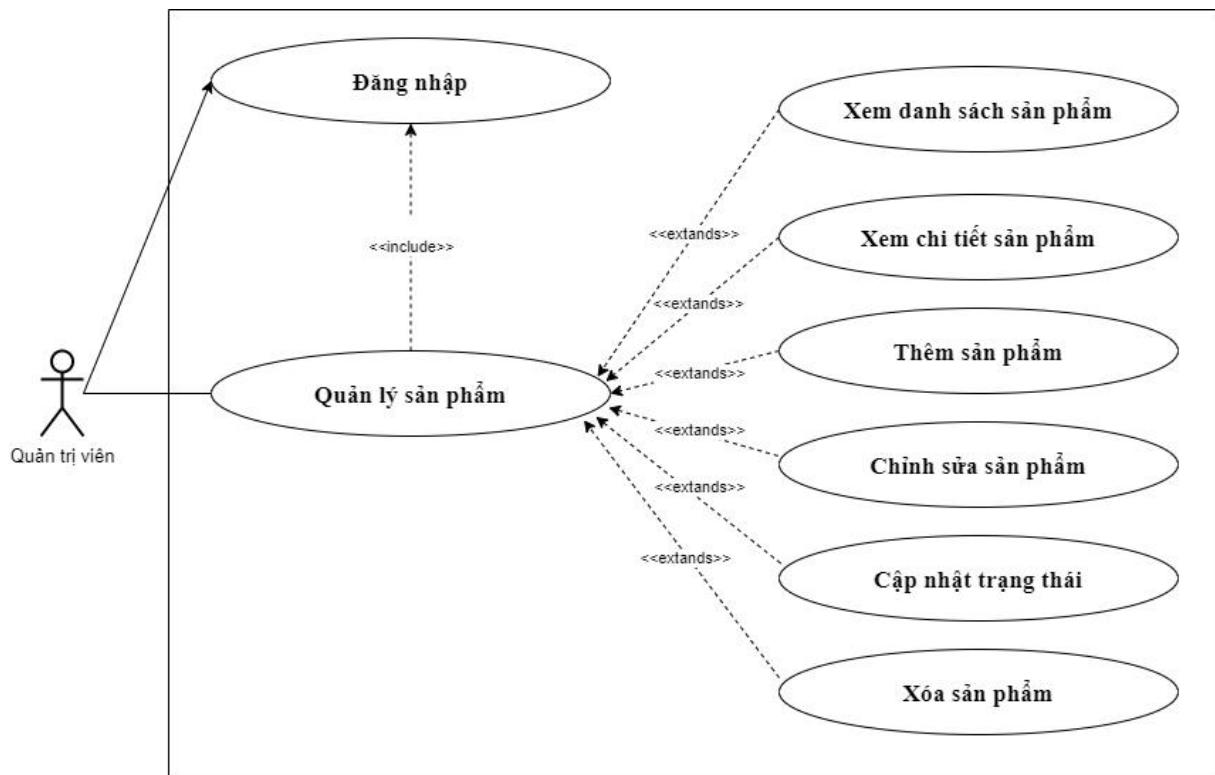


Hình 2.6: Sơ đồ ca quản lý mua hàng

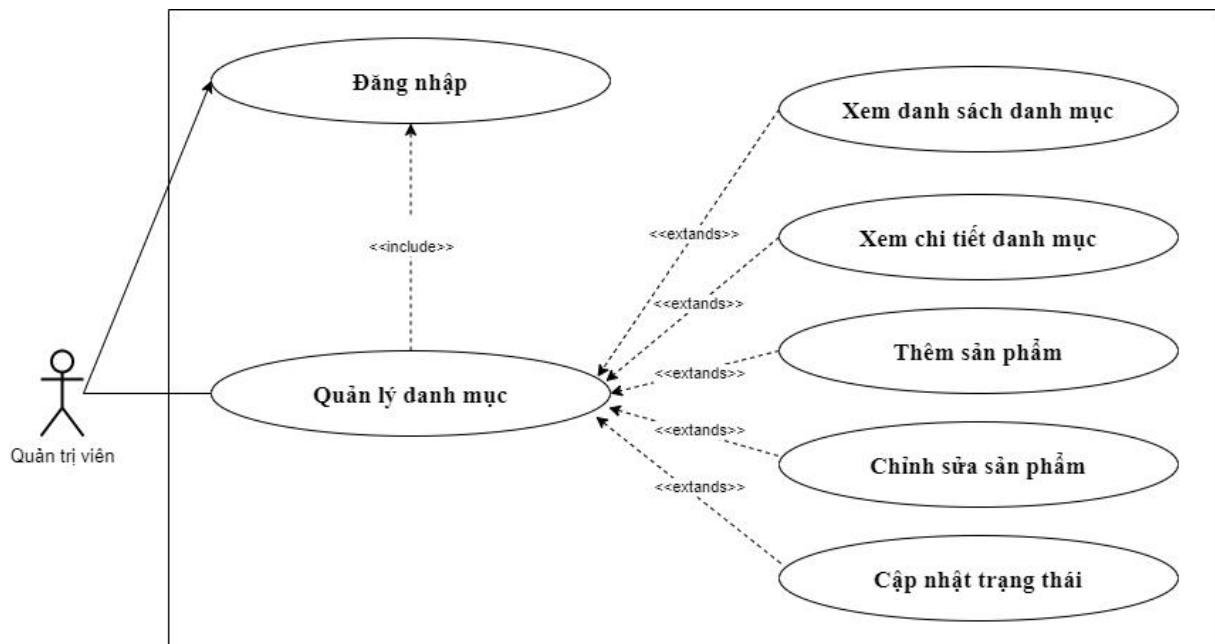
2.3.5. Sơ đồ ca sử dụng của Quản trị viên



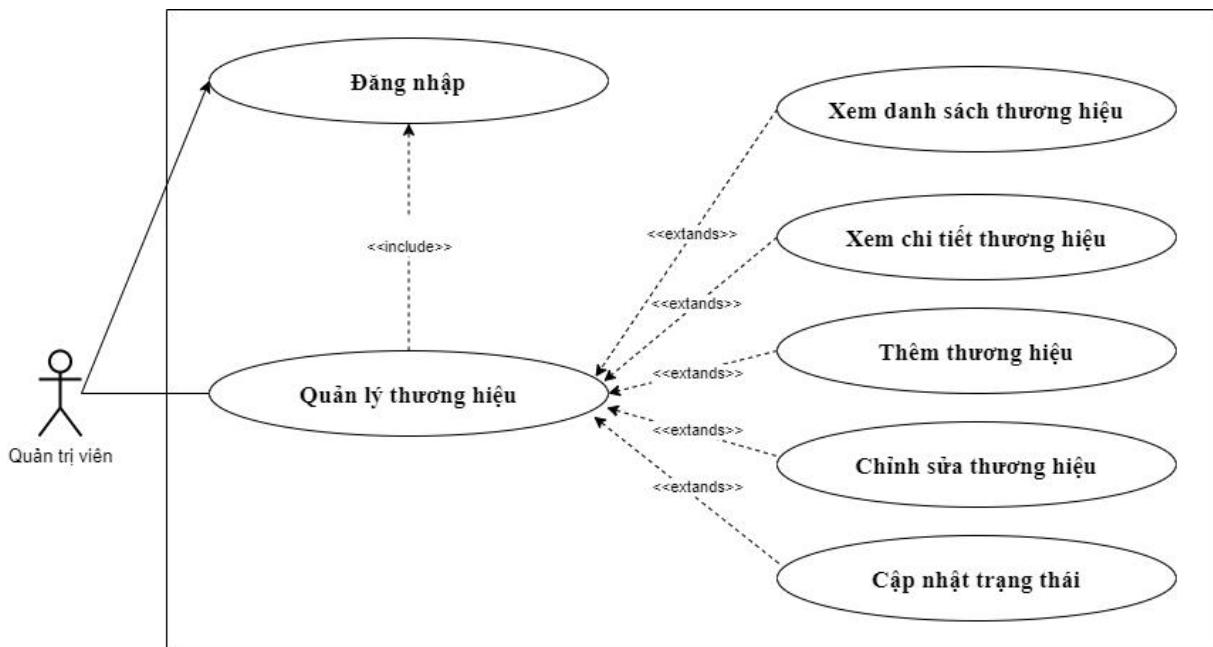
Hình 2.7: Sơ đồ ca sử dụng quản lý người dùng trong hệ thống



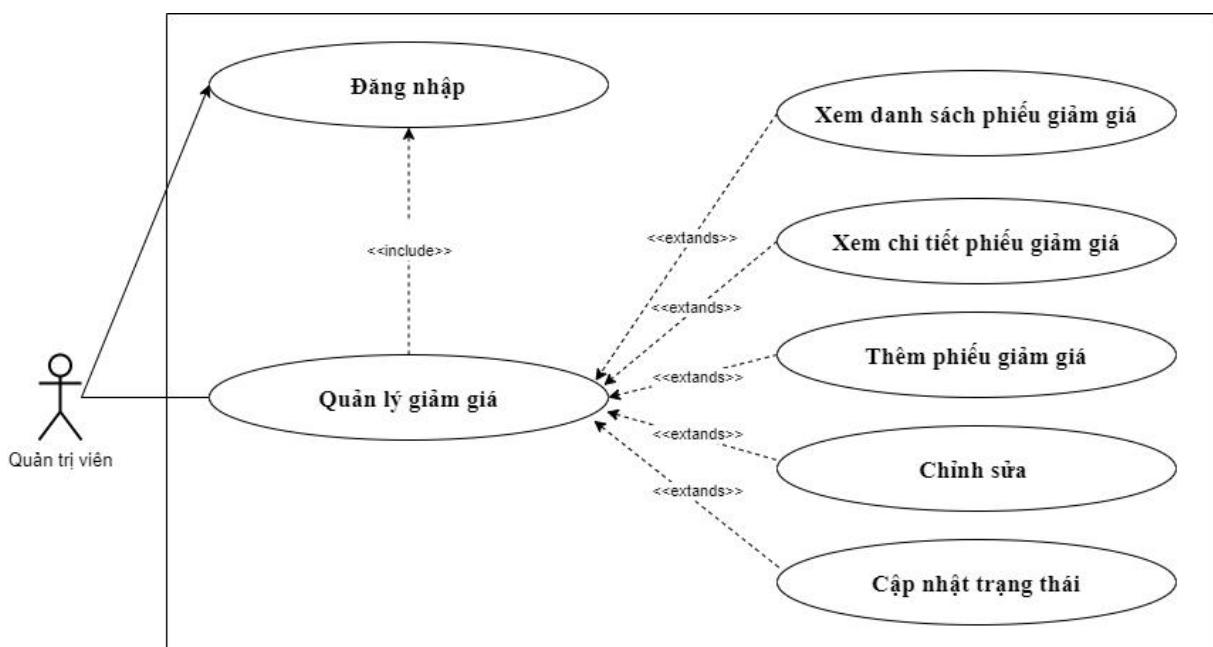
Hình 2.8: Sơ đồ ca sử dụng quản lý sản phẩm.



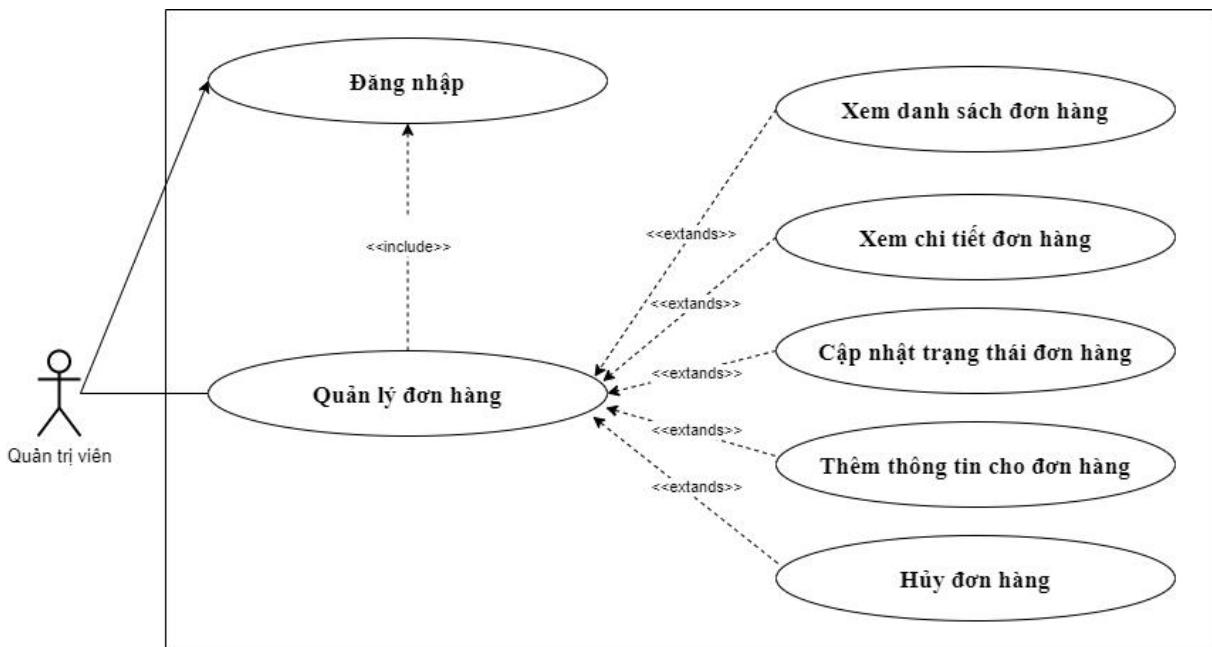
Hình 2.9: Sơ đồ ca sử dụng quản lý danh mục



Hình 2.10: Sơ đồ ca sử dụng quản lý thương hiệu



Hình 2.11: Sơ đồ ca sử dụng quản lý giảm giá



Hình 2.12: Sơ đồ ca sử dụng quản lý đơn hàng

2.4. ĐẶC TẢ CA SỬ DỤNG

2.4.1. Đặc tả ca sử dụng Đăng ký

Bảng 2.1: Đặc tả ca sử dụng đăng ký

| Tên ca sử dụng | Đăng ký |
|---------------------|--|
| Tác nhân | Khách |
| Mô tả | Cho phép khách đăng ký để trở thành thành viên của hệ thống (bao gồm tạo Tài khoản cho công ty và tài khoản cá nhân). |
| Điều kiện kích hoạt | Người sử dụng truy cập hệ thống nhưng chưa có tài khoản. Chọn mục Đăng ký . |
| Các bước thực hiện | <ol style="list-style-type: none"> Khách nhập các thông tin đăng ký: email , họ , tên, mật khẩu, xác nhận mật khẩu, và địa chỉ. Bấm vào button Đăng ký. Hệ thống kiểm tra dữ liệu. Thông báo kết quả đăng ký. |
| Kết quả | Nếu các thông tin hợp lệ, thông báo đăng ký thành công và cho phép truy cập hệ thống. |
| Trường hợp lỗi | <ol style="list-style-type: none"> Người dùng nhập các thông tin đăng ký không hợp lệ hoặc bị thiếu |

| | |
|--|--|
| | <ol style="list-style-type: none"> 2. Hệ thống kiểm tra dữ liệu nhập vào là sai. 3. Các thông tin trùng lặp, đã tồn tại trong hệ thống. 4. Thông báo lỗi ra màn hình. |
|--|--|

2.4.2. Đặc tả ca sử dụng Đăng nhập

Bảng 2.2: Đặc tả ca sử dụng đăng nhập

| Tên ca sử dụng | Đăng nhập |
|---------------------|--|
| Tác nhân | Khách hàng, Quản trị viên |
| Mô tả | Cho phép khách hàng, quản trị viên đăng nhập vào hệ thống để thực hiện các chức năng của mình. |
| Điều kiện kích hoạt | Khách hàng, quản trị viên truy cập hệ thống nhưng chưa đăng nhập. Chọn mục Đăng nhập . |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Người dùng nhập các thông tin email và mật khẩu. 2. Bấm nút Đăng nhập. 3. Hệ thống kiểm tra dữ liệu nhập là đúng. 4. Đưa đến trang chủ. |
| Kết quả | Đưa đến trang chủ của website hoặc trang quản lý của admin. |
| Trường hợp lỗi | <ol style="list-style-type: none"> 1. Người dùng không nhập các thông tin email và mật khẩu. 2. Hệ thống kiểm tra dữ liệu nhập email và mật khẩu sai. 3. Thông báo lỗi ra màn hình. |

2.4.3. Đặc tả ca sử dụng Quản lý tài khoản

Bảng 2.3: Đặc tả ca sử dụng quản lý tài khoản

| Tên ca sử dụng | Quản lý tài khoản |
|---------------------|--|
| Tác nhân | Khách hàng |
| Mô tả | Cho phép khách hàng đăng nhập vào hệ thống để quản lý tài khoản cá nhân của mình bao gồm chỉnh sửa thông tin cá nhân, địa chỉ, và mật khẩu. |
| Điều kiện kích hoạt | khách hàng phải đăng nhập vào hệ thống. Chọn mục Tài khoản cá nhân . |
| Các bước thực hiện | <ol style="list-style-type: none"> 1. Người dùng nhập các thông tin cá nhân (ở trang thông tin tài khoản) và mật khẩu (ở trang đổi mật khẩu) muốn chỉnh sửa. 2. Bấm nút: Lưu thay đổi |

| | |
|-----------------------|---|
| | 3. Hệ thống kiểm tra dữ liệu nhập là đúng. 4. Thực hiện cập nhật thông tin. |
| Kết quả | Đưa đến trang quản lý tài khoản. |
| Trường hợp lỗi | 1. Người dùng nhập không chính xác các thông tin cá nhân và mật khẩu. 2. Hệ thống kiểm tra dữ liệu nhập vào là sai. 3. Thông báo lỗi ra màn hình. |

2.4.4. Đặc tả ca sử dụng Quản lý các đơn hàng

| Tên ca sử dụng | Quản lý đơn hàng |
|----------------------------|--|
| Tác nhân | Khách hàng |
| Mô tả | Cho phép khách hàng xem lại các đơn hàng đã đặt và hủy các đơn hàng khi chưa được vận chuyển hoặc sản phẩm bị lỗi. |
| Điều kiện kích hoạt | Khách hàng đăng nhập vào hệ thống. Chọn mục Thông tin tài khoản |
| Các bước thực hiện | 1. Chọn mục Thông tin tài khoản nhấp vào mục đơn hàng của bạn . 2. Xem danh sách các đơn hàng đã đặt. 3. Khách hàng nhấp vào button hủy đơn hàng nếu muốn hủy đơn. |
| Kết quả | Đưa đến trang đơn hàng của bạn. |
| Trường hợp lỗi | |

Bảng 2.4: Đặc tả ca sử dụng quản lý các đơn hàng

2.4.5. Đặc tả ca sử dụng Quản lý mua hàng

Bảng 2.5: Đặc tả ca sử dụng quản lý mua hàng

| Tên ca sử dụng | Quản lý mua hàng |
|----------------------------|--|
| Tác nhân | Khách hàng |
| Mô tả | Cho phép người dùng đặt sản phẩm và thanh toán trực tuyến. |
| Điều kiện kích hoạt | Khách hàng đăng nhập vào hệ thống. Chọn mục Giỏ hàng nhấp vào button Mua ngay. |
| Các bước thực hiện | 1. Chọn địa chỉ 2. Nhập phiếu giảm giá nếu có. |

| | |
|-----------------------|--|
| | 3. Nhấn button: Đặt hàng |
| Kết quả | Chuyển đến trang thanh toán của MoMo Sau khi thanh toán chuyển đến trang lịch sử đơn hàng |
| Trường hợp lỗi | 1. Nhập phiếu giảm giá không hợp lệ. 2. Thông báo lỗi ra màn hình. |

2.4.6. **Đặc tả ca sử dụng Quản lý Thêm danh mục**

Bảng 2.6: Đặc tả ca sử dụng thêm danh mục

| Tên ca sử dụng | Quản lý thêm danh mục |
|----------------------------|--|
| Tác nhân | Quản trị viên. |
| Mô tả | Cho phép quản trị viên thêm mới danh mục sản phẩm vào hệ thống. |
| Điều kiện kích hoạt | Quản trị viên đăng nhập vào hệ thống, chọn mục Quản lý người dùng . |
| Các bước thực hiện | 1. Quản trị viên truy cập vào danh sách danh mục và nhấn vào button Thêm danh mục . 2. Quản trị viên nhập đầy đủ thông tin cần thiết cho danh mục mới. 3. Quản trị viên nhấn button Lưu để lưu danh mục mới. |
| Kết quả | Danh mục mới được thêm vào và quay lại trang quản lý danh mục. |
| Trường hợp lỗi | 1. Lỗi nhập sai hoặc thiếu dữ liệu. 2. Thông báo lỗi ra màn hình. |

2.4.7. **Đặc tả ca sử dụng Quản lý thêm sản phẩm.**

Bảng 2.7: Đặc tả ca sử dụng quản lý thêm sản phẩm

| Tên ca sử dụng | Quản lý thêm sản phẩm |
|----------------------------|--|
| Tác nhân | Quản trị viên |
| Mô tả | Cho phép người quản lý thêm mới sản phẩm . |
| Điều kiện kích hoạt | Người quản lý đăng nhập vào hệ thống. Chọn mục QL sản phẩm ở thanh sidebar. |
| Các bước thực hiện | 1. Người dùng chọn mục nhấp vào button Thêm sản phẩm . ở trang quản lý sản phẩm 2.Nhập đầy đủ thông tin được yêu cầu. |

| | |
|-----------------------|--|
| | 3.Nhấn button Lưu . |
| Kết quả | Quay lại trang danh sách sản phẩm. |
| Trường hợp lỗi | 1. Lỗi nhập sai hoặc thiếu dữ liệu. 2. Thông báo lỗi ra màn hình. |

2.4.8. Đặc tả ca sử dụng Quản lý người dùng

Bảng 2.8: Đặc tả ca sử dụng quản lý người dùng

| Tên ca sử dụng | Quản lý người dùng |
|----------------------------|---|
| Tác nhân | Quản trị viên |
| Mô tả | Cho phép người quản trị viên quản lý được toàn bộ thông tin về các khách hàng trong hệ thống. |
| Điều kiện kích hoạt | Người quản trị viên đăng nhập vào hệ thống. Chọn mục QL người dùng ở thanh sidebar. |
| Các bước thực hiện | 1. Quản trị viên chọn mục Quản lý người dùng để xem toàn bộ danh sách các tài khoản trong hệ thống. 2. Quản trị viên có thể xóa hoặc khóa một tài khoản bất kỳ. 3. Lưu lại thông tin đã chỉnh sửa. |
| Kết quả | Đưa đến trang danh sách người dùng |
| Trường hợp lỗi | 1. Lỗi do không có dữ liệu để hiển thị 2. Thông báo lỗi ra màn hình. |

2.4.9. Đặc tả ca sử dụng Quản lý thêm thương hiệu

Bảng 2.9: Đặc tả ca sử dụng quản lý thêm thương hiệu

| Tên ca sử dụng | Quản lý thêm thương hiệu |
|----------------------------|---|
| Tác nhân | Quản trị viên |
| Mô tả | Cho phép quản trị viên tạo thêm thương hiệu với hệ thống của mình. |
| Điều kiện kích hoạt | Quản trị viên đăng nhập vào hệ thống. Chọn mục QL Thương hiệu ở thanh sidebar. |
| Các bước thực hiện | 1. Quản trị viên chọn mục quản lý thương hiệu để hiển thị tất cả thương hiệu đang có. 2. Quản trị viên chọn Thêm thương hiệu 3. Nhập đầy đủ thông tin được yêu cầu. 4. Nhấn button Lưu . |

| | |
|-----------------------|--|
| Kết quả | Dưa đến trang danh sách thương hiệu |
| Trường hợp lỗi | 1. Lỗi do không có dữ liệu để hiển thị. 2. Thông báo lỗi ra màn hình. |

2.4.10. Đặc tả ca sử dụng Quản lý thêm phiếu giảm giá

Bảng 2.10: Đặc tả ca sử dụng quản lý thêm phiếu giảm giá

| Tên ca sử dụng | Quản lý thêm phiếu giảm giá |
|----------------------------|--|
| Tác nhân | Quản trị viên |
| Mô tả | Cho phép người quản trị viên quản lý các phiếu giảm giá trong hệ thống. |
| Điều kiện kích hoạt | Quản trị viên đăng nhập vào hệ thống. Chọn mục QL Giảm giá ở thanh sidebar. |
| Các bước thực hiện | 1. Quản trị viên chọn mục quản lý giảm giá để xem toàn bộ danh sách phiếu giảm giá còn giá trị. 2. Quản trị viên chọn Thêm phiếu giảm giá . 3. Nhập đầy đủ thông tin được yêu cầu. 4. Nhấn button Lưu . |
| Kết quả | Dưa đến trang danh sách phiếu giảm giá. |
| Trường hợp lỗi | 1. Lỗi do không có dữ liệu để hiển thị 2. Thông báo lỗi ra màn hình. |

2.4.11. Đặc tả ca sử dụng Quản lý đơn hàng

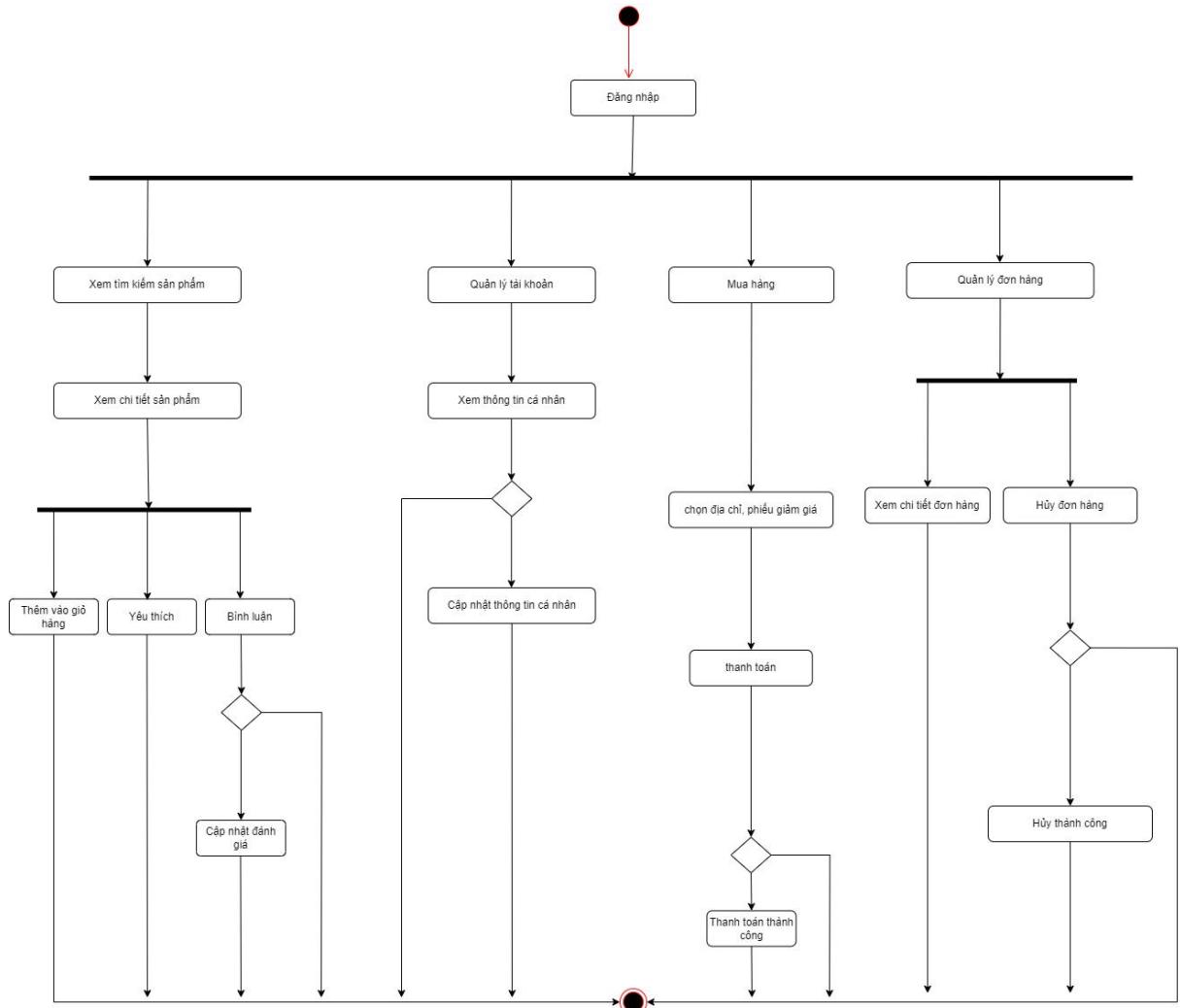
Bảng 2.11: Đặc tả ca sử dụng quản lý đơn hàng

| Tên ca sử dụng | Quản lý đơn hàng |
|----------------------------|--|
| Tác nhân | Quản trị viên |
| Mô tả | Cho phép quản trị viên quản lý tất cả các đơn hàng, chuyển đổi trạng thái , xem chi tiết , thêm thông báo |
| Điều kiện kích hoạt | Người quản trị viên đăng nhập vào hệ thống. Chọn mục QL Đơn hàng ở thanh sidebar |
| Các bước thực hiện | 1. Quản trị viên chọn mục Quản lý đơn hàng để xem toàn bộ danh sách các đơn hàng tồn tại trong hệ thống. 2. Ở trong bảng đơn hàng, quản trị viên có thể nhấn button chi tiết để xem chi tiết đơn hàng 3. Quản trị viên có thể thay đổi trạng thái đơn hàng dựa trên |

| | |
|-----------------------|---|
| | tình hình thực tế. |
| | 4. Quản trị viên có thể hủy các đơn hàng |
| Kết quả | Xử lý các đơn hàng. |
| Trường hợp lỗi | 1. Lỗi do không có dữ liệu để hiển thị 2. Thông báo lỗi ra màn hình. |

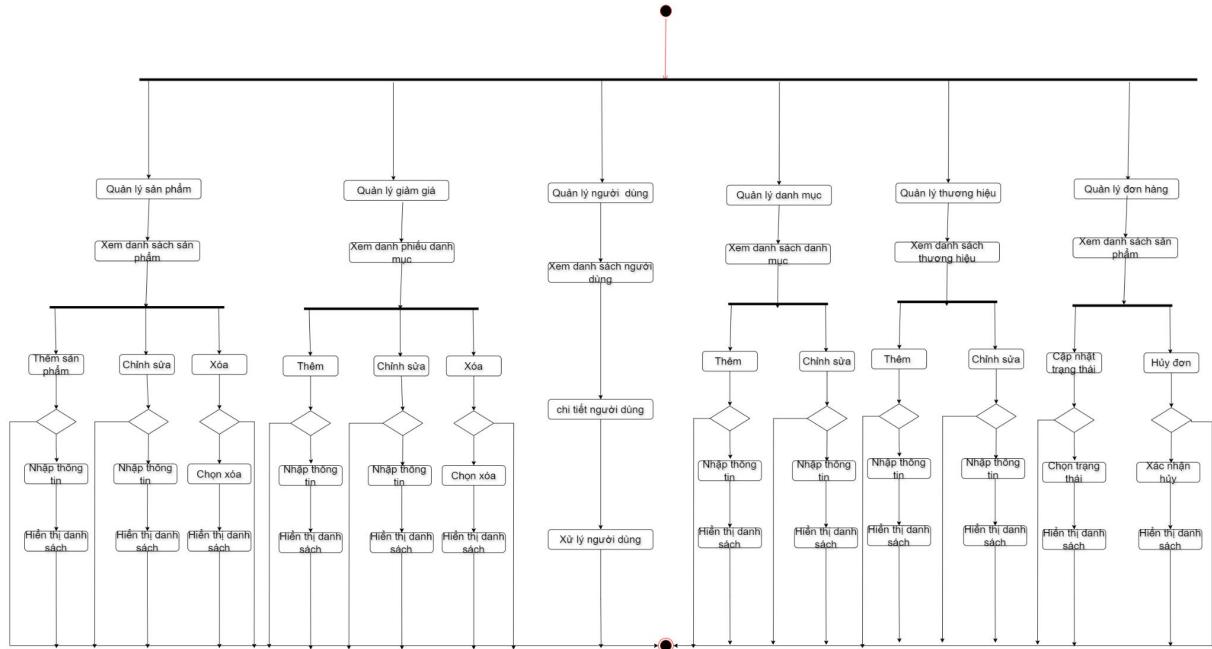
2.5. SƠ ĐỒ HOẠT ĐỘNG

2.5.1. Sơ đồ hoạt động của khách hàng



Hình 2.13: Sơ đồ hoạt động của khách hàng.

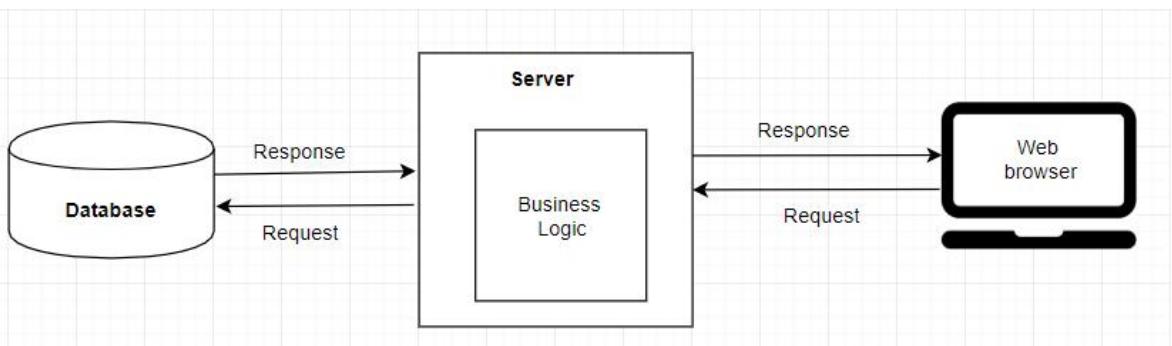
2.5.2. Sơ đồ hoạt động của quản trị viên



Hình 2.14: Sơ đồ hoạt động của quản trị viên

2.6. KIẾN TRÚC HỆ THỐNG

Đây là ứng dụng có sự tương tác trên môi trường web và hệ thống sẽ được xây dựng theo mô hình client-server và dựa trên kiến trúc RESTful. Trong đó server nắm vai trò chính trong xử lý logic nghiệp vụ và quản trị cơ sở dữ liệu, phía clients có trình duyệt web.



Hình 2.15: Sơ đồ kiến trúc hệ thống

2.7. THIẾT KẾ CƠ SỞ DỮ LIỆU

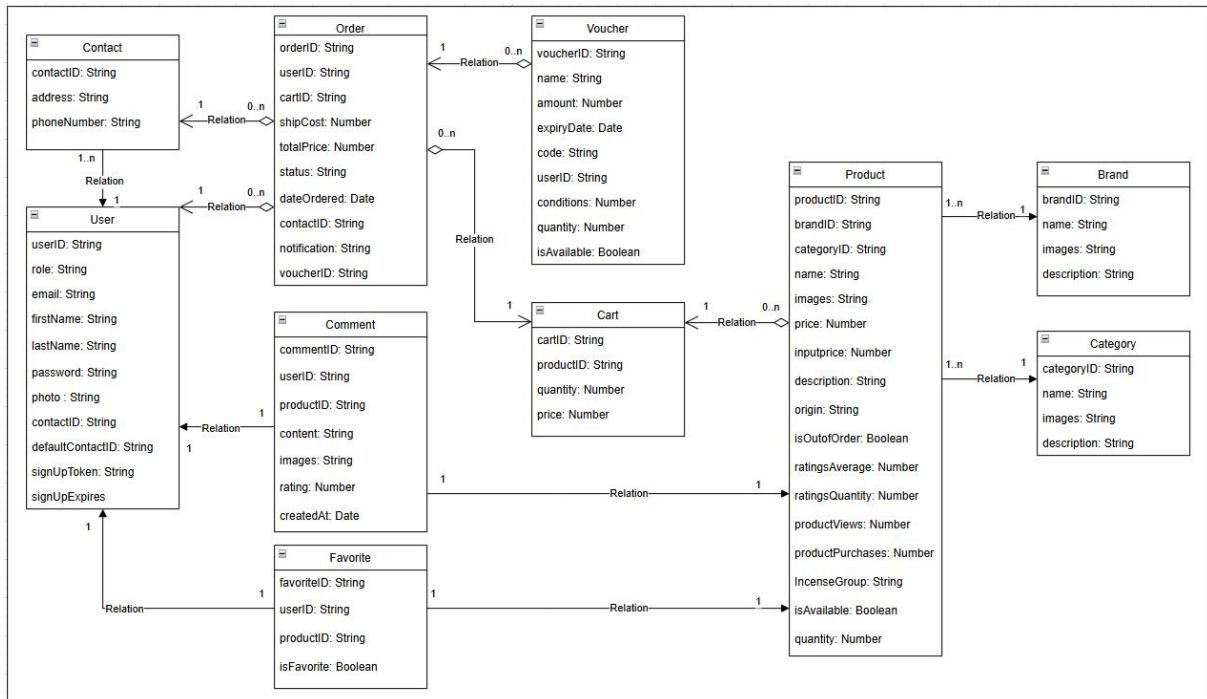
2.7.1. Mô hình thiết kế cơ sở dữ liệu

Cơ sở dữ liệu của hệ thống sẽ gồm các bảng sau:

- User: là bảng dữ liệu người dùng, chứa thông tin của người dùng trong hệ thống.

- Voucher: là bảng dữ liệu về mã giảm giá, lưu trữ thông tin về các mã giảm giá.
- Cart: là bảng dữ liệu thông tin giỏ hàng. Lưu trữ thông tin về giỏ hàng của khách hàng.
- Product: là bảng dữ liệu thông tin sản phẩm, chứa thông tin thông tin về các sản phẩm.
- Order: là bảng lưu trữ thông tin về các đơn hàng của người dùng.
- Favorite: là bảng dữ liệu lưu trữ thông tin về các sản phẩm yêu thích của người dùng.
- Comment: là bảng dữ liệu các phản bình luận. Lưu trữ thông tin về các bình luận của người dùng về sản phẩm.
- Contact: là bảng dữ liệu về thông tin liên hệ. Lưu trữ thông tin liên hệ từ khách hàng.
- Category: là bảng dữ liệu về danh mục sản phẩm. Lưu trữ thông tin về các danh mục sản phẩm.
- Brand: là bảng dữ liệu về thương hiệu sản phẩm. Lưu trữ thông tin về các thương hiệu.

Mối quan hệ giữa các bảng dữ liệu được thể hiện dưới đây:



Hình 2.16: Sơ đồ quan hệ cơ sở dữ liệu

2.7.2. Chi tiết các bảng dữ liệu

Bảng 2.12: Bảng dữ liệu User

| User | | | | |
|------------------|--------------|---------------|---------------------------------------|-------------------------------|
| Tên cột | Kiểu dữ liệu | Độ dài tối đa | Ghi chú | Giải thích |
| userID | String | 50 | Khóa chính, Duy nhất | Khóa chính của bảng |
| email | String | 50 | Duy nhất | Email |
| password | String | 20 | | Mật khẩu |
| role | String | 10 | Giá trị mặc định: User | Vai trò |
| firstName | String | 20 | | Tên của người dùng |
| lastName | String | 20 | | Họ của người dùng |
| photo | String | 50 | | Ảnh đại diện |
| contactID | String | 50 | Khóa ngoại, liên kết với bảng Contact | Mã thông tin liên hệ |
| defaultContactID | String | 50 | Khóa ngoại, liên kết với bảng Contact | Mã thông tin liên hệ mặc định |
| signUpToken | String | 100 | | Token để đăng ký |
| signUpExpires | Date | | | Ngày hết hạn đăng ký |

Bảng 2.13: Bảng dữ liệu Voucher

| Voucher | | | | |
|-------------|--------------|---------------|----------------------|-------------------------|
| Tên cột | Kiểu dữ liệu | Độ dài tối đa | Ghi chú | Giải Thích |
| voucherID | String | 50 | Khóa chính, duy nhất | Khóa chính của bảng |
| name | String | 50 | | Tên mã khuyến mãi |
| amount | Number | | | Số tiền giảm |
| code | String | 50 | | Mã phiếu giảm giá |
| conditions | Number | | | Điều kiện giảm giá |
| quantity | Number | 5 | | Số lượng phiếu giảm giá |
| expiryDate | Date | | | Ngày hết hạn |
| isAvailable | Boolean | | Mặc định :true | Trạng thái |

Bảng 2.14: Bảng dữ liệu Cart

| Cart | | | | |
|-----------|--------------|---------------|---------------------------------------|---------------------|
| Tên cột | Kiểu dữ liệu | Độ dài tối đa | Ghi chú | Giải thích |
| cartID | String | 50 | Khóa chính, duy nhất | Khóa chính của bảng |
| productID | String | 50 | Khóa ngoại, liên kết với bảng Product | Mã sản phẩm |
| notes | String | 100 | | Ghi chú |
| price | Number | | | Số tiền |

Bảng 2.15: Bảng dữ liệu Product

| Product | | | | |
|----------------|--------------|---------------|--|------------------------------|
| Tên cột | Kiểu dữ liệu | Độ dài tối đa | Ghi chú | Giải thích |
| productID | String | 50 | Khóa chính, duy nhất | Khóa chính của bảng |
| brandID | String | 50 | Khóa ngoại, liên kết với bảng Brand | Mã thương hiệu |
| category | String | 50 | Khóa ngoại, liên kết với bảng Category | Mã danh mục |
| name | String | 50 | | Tên sản phẩm |
| images | String | 50 | | Hình ảnh sản phẩm |
| price | Number | | | Giá tiền |
| inputPrice | Number | | Giá trị mặc định: 0 | Giá tiền nhập hàng |
| description | String | 2000 | | Mô tả |
| origin | String | 2000 | | Xuất xứ |
| ratingsAverage | Number | | Giá trị mặc định: 5.0 Giá trị lớn nhất: 5.0 | Trung bình các lượt đánh giá |

| | | | | |
|------------------|---------|------|---|-------------------------------------|
| | | | Giá trị nhỏ nhất: 1.0 | |
| ratingsQuantity | Number | | Giá trị mặc định: 0 | Số lượng đánh giá |
| productViews | Number | | Giá trị mặc định: 0 | Số lượt xem của sản phẩm |
| productPurchases | Number | | Giá trị mặc định: 0 | Số lượng sản phẩm đã bán |
| IncenseGroup | String | 2000 | | Nhóm hương |
| isAvailable | Boolean | | false: Không có sẵn true: Có sẵn Giá trị mặc định: true | Thể hiện còn bán sản phẩm hay không |
| outfoOrder | Boolean | | Giá trị mặc định: true | Còn hay hết hàng |
| quantity | Number | | | Số lượng sản phẩm |

Bảng 2.16: Bảng dữ liệu Order

| Order | | | | |
|--------------|--------------|---------------|---------------------------------------|----------------------|
| Tên cột | Kiểu dữ liệu | Độ dài tối đa | Ghi chú | Giải thích |
| orderID | String | 50 | Khóa chính, duy nhất | Khóa chính của bảng |
| userID | String | 50 | Khóa ngoại, liên kết với bảng User | Mã người dùng |
| cartID | String | 50 | Khóa ngoại, liên kết với bảng Cart | Mã giỏ hàng |
| shipCost | Number | | Giá trị mặc định: 0 | Giá tiền vận chuyển |
| totalPrice | Number | | Giá trị mặc định: 0 | Tổng số tiền |
| status | String | 20 | Giá trị mặc định: Pending | Trạng thái |
| dateOrdered | Date | | | Ngày đặt hàng |
| contactID | String | 50 | Khóa ngoại, liên kết với bảng Contact | Mã thông tin liên hệ |
| notification | String | 200 | | Thông báo |

Bảng 2.17: Bảng dữ liệu Favorite

| Favorite | | | | |
|------------|--------------|---------------|--|---|
| Tên cột | Kiểu dữ liệu | Độ dài tối đa | Ghi chú | Giải thích |
| favoriteID | String | 50 | Khóa chính, duy nhất | Khóa chính của bảng |
| userID | String | 50 | Khóa ngoại, liên kết với bảng User | Mã người dùng |
| productID | String | 50 | Khóa ngoại, liên kết với bảng Product | Mã sản phẩm |
| isFavorite | Boolean | | Giá trị mặc định: false false: Không yêu thích true: Yêu thích | Sản phẩm có được người dùng yêu thích hay không |

Bảng 2.18: Bảng dữ liệu Comment

| Comment | | | | |
|-----------|--------------|---------------|---------------------------------------|---------------------|
| Tên cột | Kiểu dữ liệu | Độ dài tối đa | Ghi chú | Giải thích |
| commentID | String | 50 | Khóa chính, duy nhất | Khóa chính của bảng |
| userID | String | 50 | Khóa ngoại, liên kết với bảng User | Mã người dùng |
| productID | String | 50 | Khóa ngoại, liên kết với bảng Product | Mã sản phẩm |
| content | String | 2000 | | Nội dung |
| images | String | 200 | | Hình ảnh |
| rating | Number | | | Đánh giá |
| createdAt | Date | | | Ngày tạo |

Bảng 2.19: Bảng dữ liệu Contact

| Contact | | | | |
|-------------|--------------|---------------|----------------------|---------------------|
| Tên cột | Kiểu dữ liệu | Độ dài tối đa | Ghi chú | Giải thích |
| contactID | String | 50 | Khóa chính, duy nhất | Khóa chính của bảng |
| phoneNumber | String | 20 | | Số điện thoại |
| address | String | 200 | | Địa chỉ |

Bảng 2.20: Bảng dữ liệu Category

| Category | | | | |
|-------------|--------------|---------------|----------------------|---------------------|
| Tên cột | Kiểu dữ liệu | Độ dài tối đa | Ghi chú | Giải thích |
| categoryID | String | 50 | Khóa chính, duy nhất | Khóa chính của bảng |
| name | String | 200 | | Tên danh mục |
| images | String | 100 | | Hình ảnh |
| description | String | 2000 | | Mô tả |

Bảng 2.21: Bảng dữ liệu Brand

| Brand | | | | |
|-------------|--------------|---------------|----------------------|---------------------|
| Tên cột | Kiểu dữ liệu | Độ dài tối đa | Ghi chú | Giải thích |
| brandID | String | 50 | Khóa chính, duy nhất | Khóa chính của bảng |
| name | String | 200 | | Tên thương hiệu |
| description | String | 200 | | Mô tả |

| | | | |
|--------|--------|-----|----------|
| images | String | 100 | Hình ảnh |
|--------|--------|-----|----------|

CHƯƠNG III: TRIỂN KHAI VÀ CHẠY THỬ NGHIỆM

3.1. MÔI TRƯỜNG TRIỂN KHAI

3.1.1. Quản lý mã nguồn

Trong quy trình phát triển phần mềm chuyên nghiệp, việc quản lý các phiên bản và lưu trữ mã nguồn là rất quan trọng hỗ trợ đắc lực cho lập trình viên. Trong đề tài này em đã sử dụng Git lưu trữ mã nguồn và quản lý các phiên bản.

➤ Git

Git là một hệ thống VCS (Version Control System) dùng để quản lý và kiểm tra các phiên bản source code khác nhau trong quá trình phát triển. Trên Git, có thể lưu trạng thái của file khi có nhu cầu dưới dạng lịch sử cập nhật.

3.1.2. Môi trường cài đặt

- ❖ Môi trường máy để chạy chương trình cả front-end và cả backend cần đáp ứng:
 - Node.js



Hình 3.1: Node.js

- Npm: là chương trình quản lý thư viện ngầm định trong môi trường Javascript Node.js.



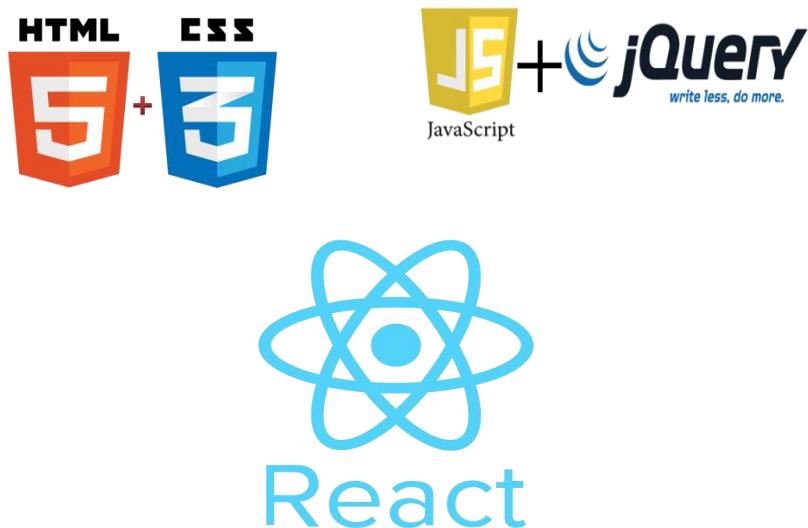
Hình 3.2: Trình quản lý thư viện npm

- ❖ Back-end là web API được xây dựng bằng Express là một framework của Node.js



Hình 3.3: Ngôn ngữ backend sử dụng

- ❖ Front-end là web application được sử dụng HTML, CSS, JavaScript với thư viện ReactJS



Hình 3.4: Ngôn ngữ front-end sử dụng

- ❖ Hệ quản trị cơ sở dữ liệu là Monggo



Hình 3.5: Hệ quản trị cơ sở dữ liệu

- ❖ Triển khai hệ thống trên Vercel:

Vercel là một nền tảng đám mây hiện đại cho phép các lập trình viên xây dựng, triển khai, và quản lý các ứng dụng web (JAMstack) một cách dễ dàng và nhanh chóng. Được biết đến với sự tích hợp mạnh mẽ với các công cụ phát triển hiện đại như Next.js, Vercel cung cấp một môi trường tối ưu để phát triển các trang web tĩnh và ứng dụng web động.



Hình 3.6: Môi trường vercel deploy bản production

3.2. KẾT QUẢ DEMO VÀ HƯỚNG DẪN SỬ DỤNG CHƯƠNG TRÌNH

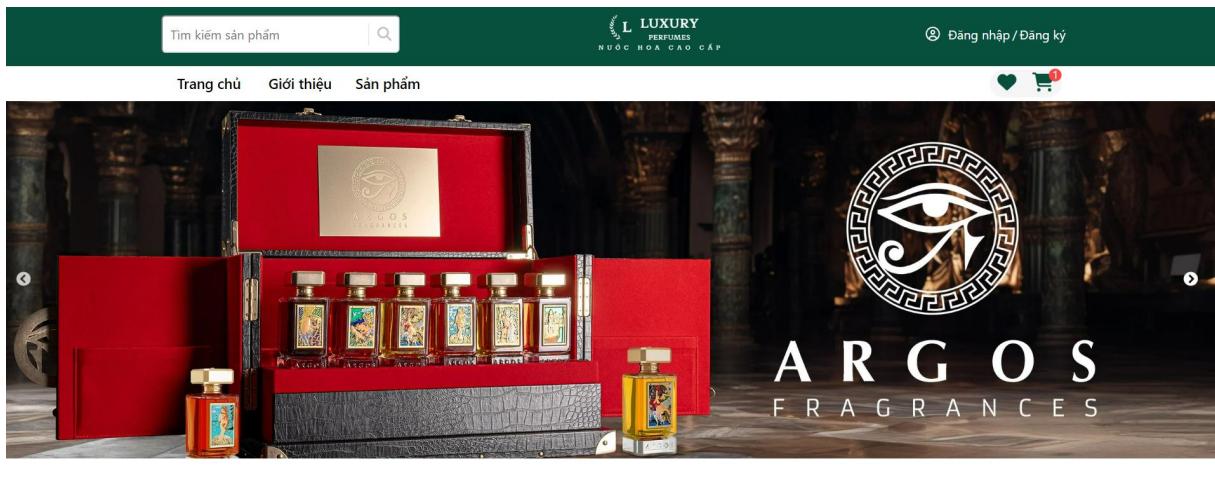
Source code được lưu trữ ở bitbucket, để chạy chương trình hoàn chỉnh chúng ta thực hiện theo các bước:

- Backend: Sử dụng lệnh **git clone** để tải toàn bộ mã chương trình về máy. Tiếp theo ở thư mục gốc của mã nguồn chạy lệnh **php artisan install** để cài đặt những gói, thư viện liên quan cần thiết. Sau đó, bật wamp tạo cơ sở dữ liệu, rồi vào file **.env** trong mã nguồn để thay đổi các thông tin kết nối cho đúng với cơ sở dữ liệu vừa tạo. Cuối cùng chạy lệnh **php artisan migrate** để tạo các bảng dữ liệu và **php artisan db:seed** để tạo dữ liệu mẫu.
- Frontend: Sử dụng lệnh **git clone** để tải toàn bộ mã chương trình web application về máy. Tiếp theo ở thư mục gốc của mã nguồn chạy lệnh **npm install** để cài đặt những gói, thư viện liên quan cần thiết. Cuối cùng chạy lệnh **npm start** để chạy chương trình lên.

Sau khi hoàn thành các bước chuẩn bị trên chúng ta sẽ chạy chương trình cụ thể và dưới đây là một số hình ảnh về hệ thống:

- Đăng ký

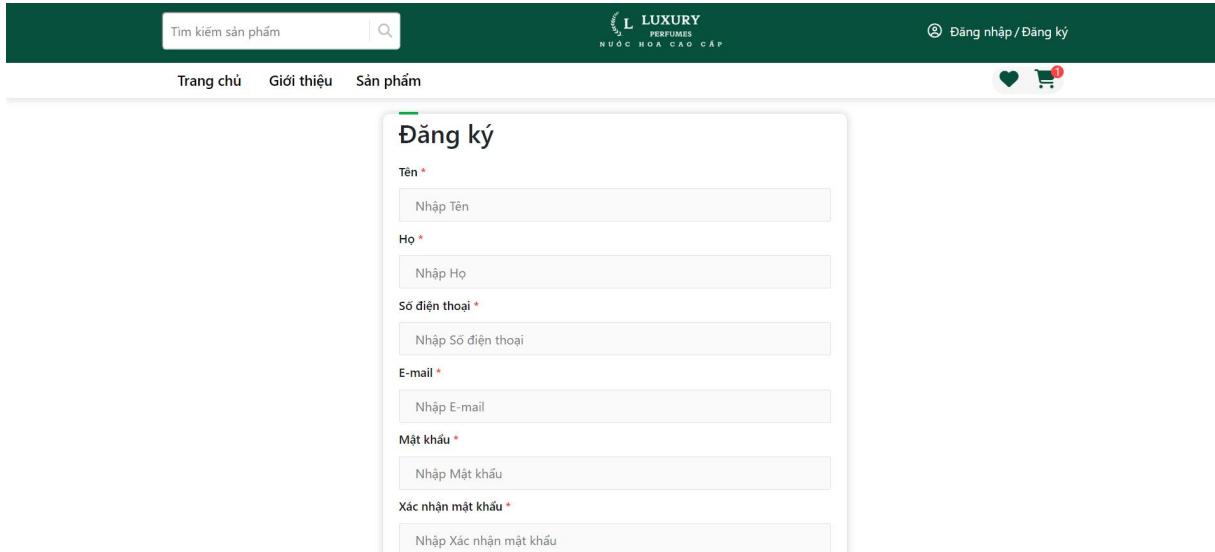
Ở phần header tại màn hình chính của trang web sẽ xuất hiện 2 liên kết là đăng ký và đăng nhập. Với trường hợp chưa có tài khoản chúng ta sẽ nhấp vào liên kết đăng ký.



Luxury Perfumes

Hình 3.7: Màn hình chính của hệ thống

Sau đó hệ thống sẽ hiện lên màn hình để đăng ký.

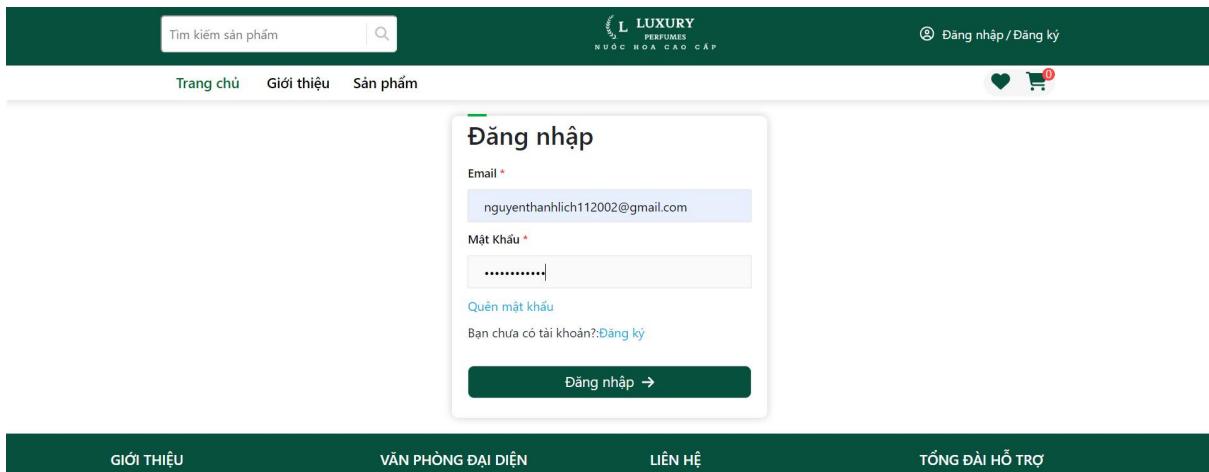


Hình 3.8: Màn hình đăng ký

Điền hợp lệ các thông tin đăng ký sau đó gửi lên hệ thống, trong trường hợp thành công hệ thống sẽ tự động đăng nhập tài khoản của người dùng vào hệ thống. Sau đó người dùng có thể sử dụng các chức năng của khách hàng trên hệ thống.

- **Đăng nhập**

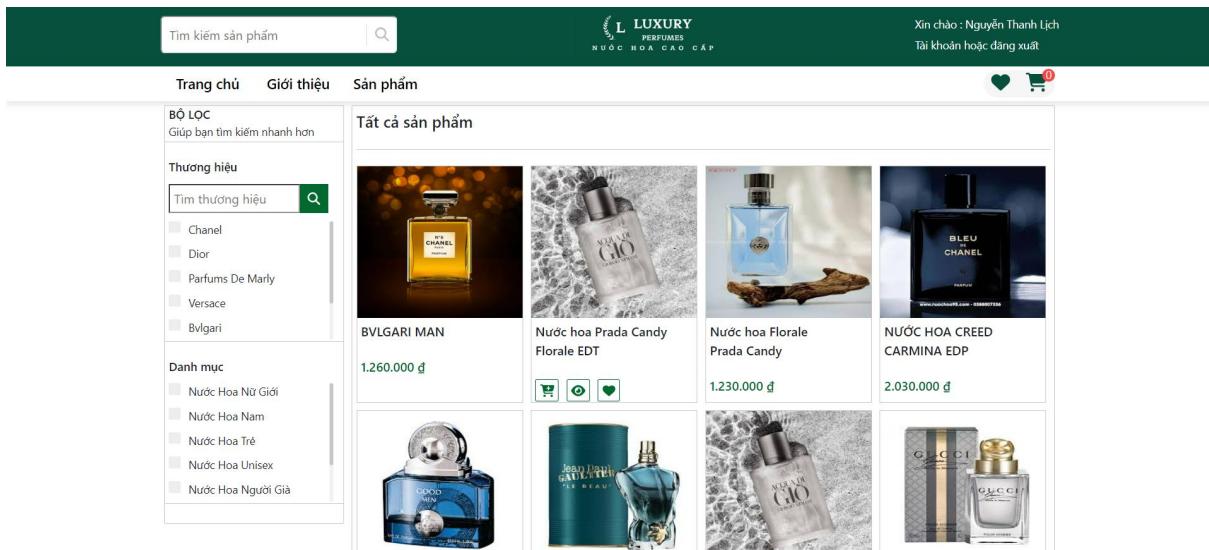
Sau khi đã có tài khoản tài khoản hợp lệ, thì người dùng có thể đăng nhập vào hệ thống bằng tài khoản đó để thực hiện các chức năng của mình.



Hình 3.9: Màn hình đăng nhập hệ thống

- **Tìm kiếm sản phẩm**

Người dùng có thể xem tìm kiếm sản phẩm ở mục sản phẩm, ở đây người dùng có thể lựa chọn danh mục, thương hiệu tên của sản phẩm tùy theo mục đích và nhu cầu cá nhân của khách hàng



Hình 3.10: Màn hình tìm kiếm sản phẩm

- **Xem chi tiết sản phẩm và xem đánh giá sản phẩm**

Trong trang sản phẩm người dùng có thể nhấn vào trang 1 sản phẩm cụ thể để có thể xem các thông tin chi tiết của nó phía dưới là đánh giá của những khách hàng đã mua hàng và để lại đánh giá.



Tim kiếm sản phẩm

LUXURY
NUOC HOA CAO CAP

Xin chào : Nguyễn Thanh Lịch
Tài khoản hoặc đăng xuất

Trang chủ Giới thiệu Sản phẩm

BVLGARI MAN

Tình trạng:
Có sẵn

1260000 đ

Trong bộ sưu tập BVLGARI MAN, Bvlgari kể câu chuyện về sự gắn kết của người đàn ông với thế giới xung quanh anh ấy, tạo ra một trải nghiệm về các cung bậc cảm xúc xoay quanh những bản năng và lưu giữ anh ấy với cội nguồn của sự sống, đồng thời tạo ra sự cân bằng tinh tế giữa con người và các yếu tố thiên nhiên. Mùi hương này tiết lộ bản chất thật sự của người đàn ông, gắn kết anh ấy lại với giá trị tinh thần nguyên bản của chính mình.

Xuất sứ:
Hà Lan

Danh mục:
Nước Hoa Nữ Giới

Thương hiệu:
Chanel

Thêm vào yêu thích

localhost:3001

Hình 3.11: Màn hình xem chi tiết sản phẩm



tiết lộ bản chất thật sự của người đàn ông, gắn kết anh ấy lại với giá trị tinh thần nguyên bản của chính mình.

Xuất sứ:
Hà Lan

Danh mục:
Nước Hoa Nữ Giới

Thương hiệu:
Chanel

Thêm vào yêu thích

Đánh giá sản phẩm



Nguyễn Thanh Lịch

16:30 6/13/2024



Sản phẩm chất lượng rất tốt



Sản phẩm

Đánh giá: 4 sao

Số lượng đánh giá: 1 lượt

Số lượt mua: 10 lượt

Số lượt thích: 20 lượt

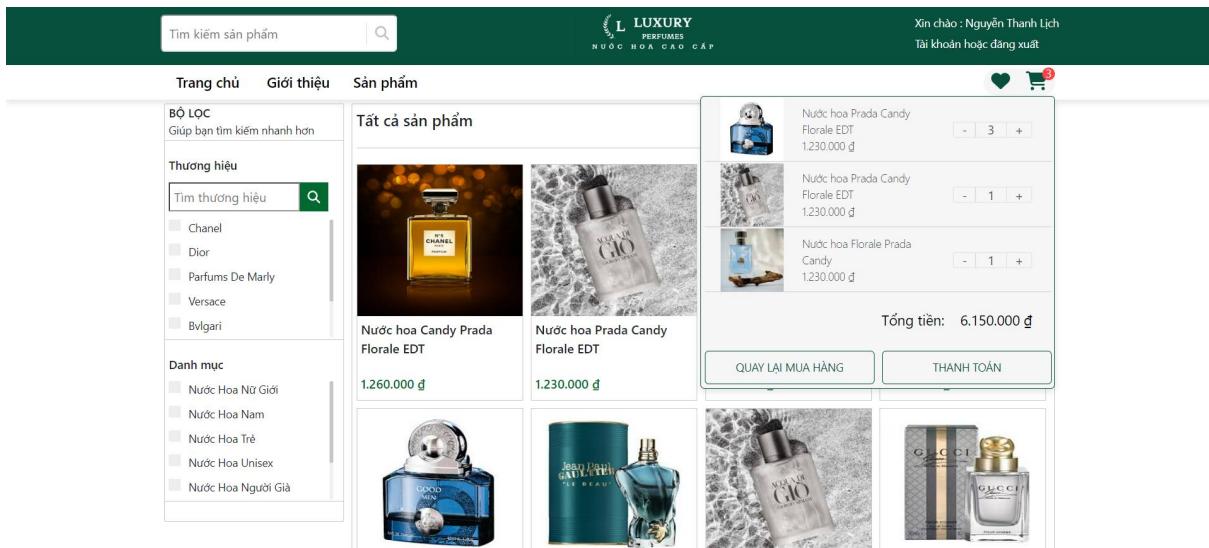
Số lượt truy cập: 100 lượt



Hình 3.12: Màn hình xem đánh giá sản phẩm

● Thêm sản phẩm vào giỏ hàng

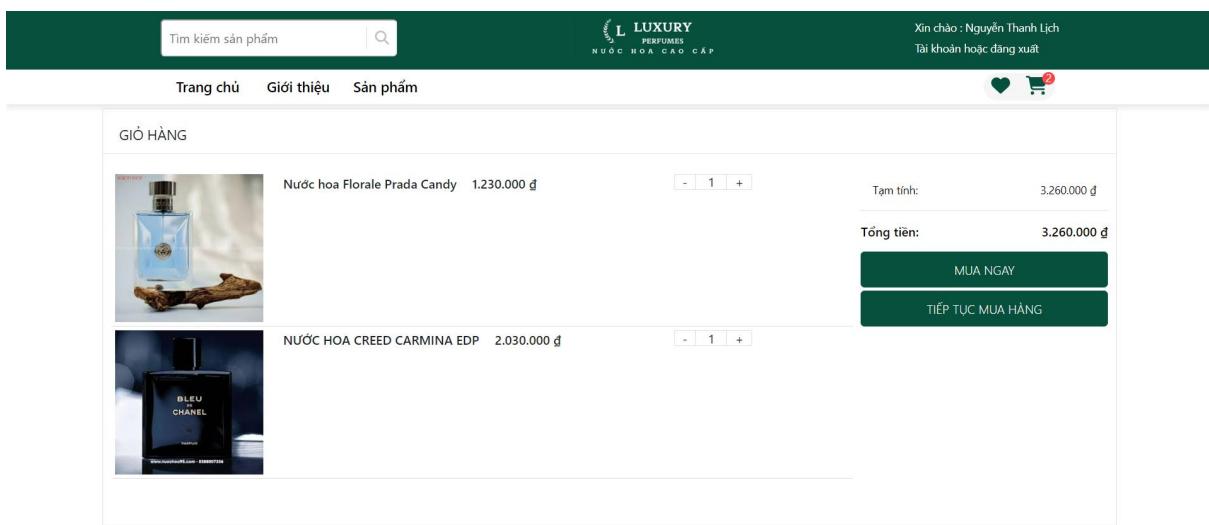
Người dùng có thể nhấn vào thêm vào giỏ hàng ở trang chi tiết sản phẩm hoặc người dùng có thể nhấn vào logo giỏ hàng sẽ xuất hiện khi đưa con trỏ chuột vào ô sản phẩm.



Hình 3.13: Màn hình khi nhấp vào icon giỏ hàng trên header

- Xem chi tiết giỏ hàng

Người dùng nhấp vào thanh toán ở giỏ hàng mini để đến trang giỏ hàng



Hình 3.14: Màn hình khi nhấp vào icon giỏ hàng trên header

- Thực hiện mua hàng

Khi người dùng nhấp vào nút "Mua ngay", họ sẽ được chuyển đến trang mua hàng. Tại đây, người dùng có thể chọn thông tin như địa chỉ, số điện thoại và mã giảm giá để hoàn thành quá trình mua hàng.

Đề tài: Xây dựng website quản lý bán nước hoa

The screenshot shows a shopping cart page for a perfume store. At the top, there's a search bar, the store logo 'LUXURY NƯỚC HOA CAO CẤP', and account information. Below the header, there are tabs for 'Trang chủ', 'Giới thiệu', and 'Sản phẩm'. The main content area is divided into sections: 'Thông tin mua hàng' (Shipping info), 'Vận chuyển' (Shipping method), 'Thanh toán' (Payment method), and a summary table for the order.

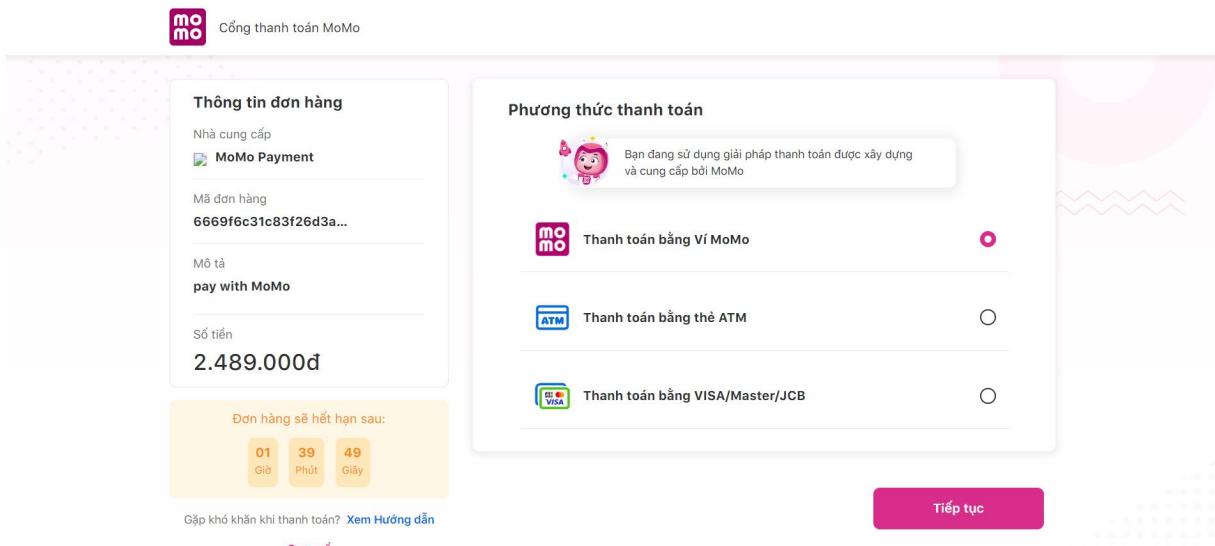
| Đơn hàng (3 sản phẩm) | |
|--|------------|
| 1 Nước hoa Prada Candy Flora EDT Nước Hoa Nữ Giả/Chanel | 1.230.000đ |
| 1 Nước hoa Flora Prada Candy Nước Hoa Nữ Giả/Chanel | 1.230.000đ |
| Nhập mã giảm giá GG20 | |
| Tạm tính | 6.150.000đ |
| Phí vận chuyển | 29.000đ |
| Giảm giá | 200.000đ |
| Điều kiện | 2.000.000đ |
| Tổng cộng | 5.979.000đ |

At the bottom right, there are buttons for 'Quay về giỏ hàng' (Return to cart) and a large green 'ĐẶT HÀNG' (Place Order) button.

Hình 3.15: Màn hình trang mua hàng

● Thực hiện thanh toán

Sau khi nhấn đặt hàng người dùng sẽ được chuyển đến trang thanh toán của trực tuyến của momo ở đây người dùng có thể chọn các chức năng thanh toán phù hợp để thanh toán đơn hàng của mình



Hình 3.16: Màn hình phương thức thanh toán bằng momo

● Quản lý đơn hàng

Sau khi thanh toán thành công người dùng sẽ được trả về trang đơn hàng của bạn, ở đây người dùng có thể xem lại đơn hàng của mình hoặc hủy đơn nếu muốn.

The screenshot shows a website interface for managing perfume sales. At the top, there is a search bar labeled "Tim kiem sản phẩm" and a magnifying glass icon. To the right of the search bar is the logo "LUXURY PERFUMES NƯỚC HOA CAO CẤP". Further right, it says "Xin chào: Nguyễn Thanh Lịch" and "Tài khoản hoặc đăng xuất". Below the header, there are navigation links: "Trang chủ", "Giới thiệu", "Sản phẩm", "Loving", and a shopping cart icon with a red notification dot.

The main content area has a title "TRANG TÀI KHOẢN" and a subtitle "Đơn hàng của bạn". A table lists five orders:

| Thông tin tài khoản | Đơn hàng | Địa chỉ | Giá trị đơn hàng | TT đơn hàng | Chi tiết | Hủy đơn |
|---------------------|--------------------------|---|------------------|-------------|---------------------------|----------------------|
| Đơn hàng của bạn | 6669f6c31c83f26d3af01ac0 | Phường An Lạc - Tỉnh Hải Dương - Thành phố Chí Linh | 2489000 | Confirmed | <button>Chi tiết</button> | <button>Hủy</button> |
| Đổi mật khẩu | 666754f7447a2141a7338bed | Phường An Lạc - Tỉnh Hải Dương - Thành phố Chí Linh | 2489000 | Pending | <button>Chi tiết</button> | <button>Hủy</button> |
| Địa chỉ | 666751f24a685f753fe88c80 | Phường An Lạc - Tỉnh Hải Dương - Thành phố Chí Linh | 2389000 | Pending | <button>Chi tiết</button> | <button>Hủy</button> |
| | 66672289f6420db2eb5b0443 | Phường An Lạc - Tỉnh Hải Dương - Thành phố Chí Linh | 1259000 | Confirmed | <button>Chi tiết</button> | <button>Hủy</button> |

The screenshot shows a detailed view of an order. At the top, there is a search bar labeled "Tim kiem sản phẩm" and a magnifying glass icon. To the right of the search bar is the logo "LUXURY PERFUMES NƯỚC HOA CAO CẤP". Further right, it says "Xin chào: Nguyễn Thanh Lịch" and "Tài khoản hoặc đăng xuất". Below the header, there are navigation links: "GIỚI THIỆU", "VĂN PHÒNG ĐẠI DIỆN", "LIÊN HỆ", and "TỔNG ĐÀI HỖ TRỢ".

The main content area shows a modal window titled "Đơn hàng". Inside the modal, the following information is displayed:

- ID: 6669f6c31c83f26d3af01ac0
- Ngày đặt hàng: 2024-06-13T02:28:03.044Z
- Địa chỉ: Phường An Lạc - Tỉnh Hải Dương - Thành phố Chí Linh
- Giá trị đơn hàng: 2489000
- Phi vận chuyển: 29000
- Tình trạng: Confirmed

The modal also contains a section for "Thông báo:" which is currently empty. Below this, two items are listed:

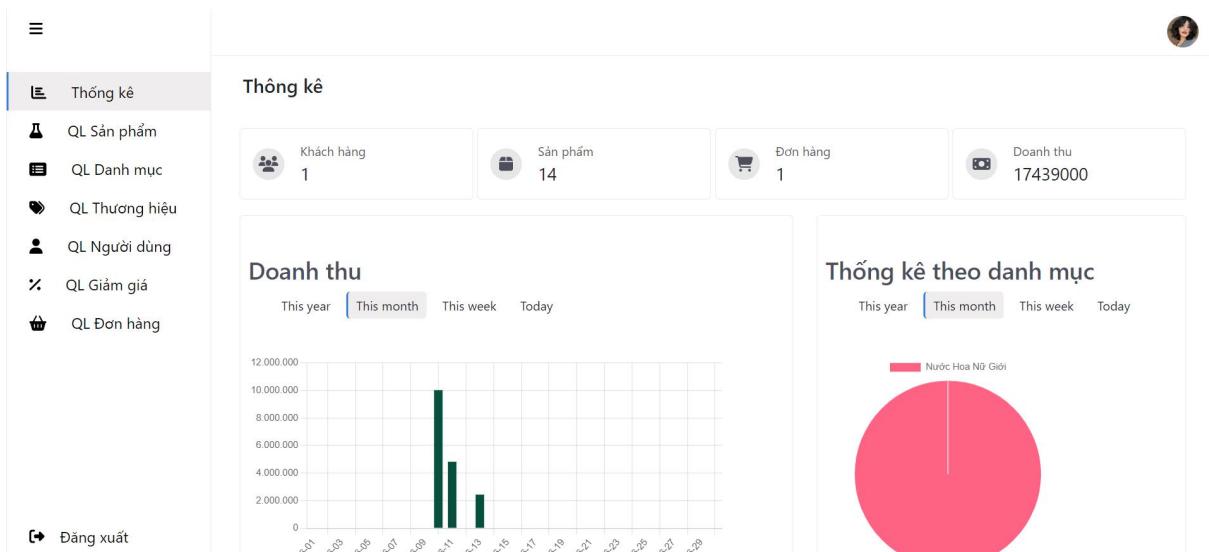
- Nước hoa Prada Candy Florale EDT
- Nước hoa Florale Prada Candy

Each item has a small thumbnail image, the product name, price (1.230.000 đ), quantity (Số lượng 1), and a dropdown menu for quantity selection.

Hình 3.17: Màn hình đơn hàng của bạn

● Quản lý thống kê

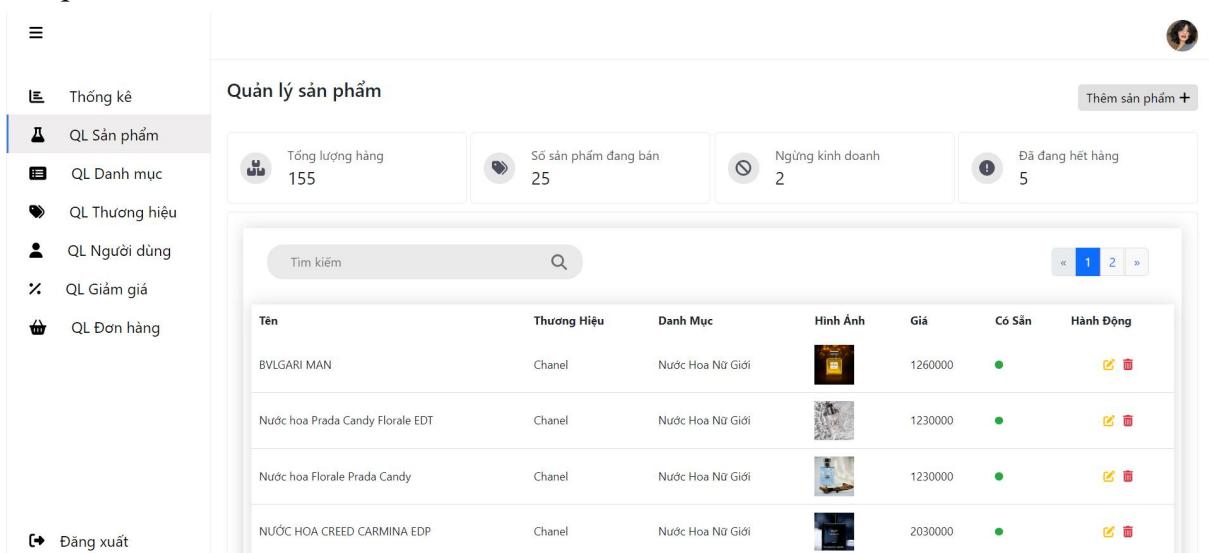
Khi đăng nhập vào hệ thống với tài khoản là Quản trị viên ở đây quản trị viên có thể xem thống kê doanh thu, tỷ trọng số lượng giữa các danh mục, sản phẩm top đầu, và danh mục top đầu.



Hình 3.19: Màn hình quản lý thống kê

- **Quản lý sản phẩm**

Quản trị viên có thể xem tất cả sản phẩm, số sản phẩm đang bán, ngừng kinh doanh, hết hàng. Ngoài ra quản trị viên còn có thể thực hiện các nồng thêm, xóa và cập nhật sản phẩm



Hình 3.20: Màn hình quản lý sản phẩm

- **Quản lý danh mục**

Quản trị viên có thể xem thông số lượng danh mục, danh mục đang hoạt động, danh mục ngành kinh doanh, danh mục hết hàng. Ngoài ra quản trị viên còn có thể thêm và cập nhật danh mục.

| ID | Tên | Mô tả | Hình ảnh | Hành Động |
|--------------------------|------------------|---|----------|-------------------------------------|
| 666400275a3db59b521fb4b | Nước Hoa Nữ Giới | Nước hoa nữ này là sự kết hợp tinh tế của hương hoa tươi mới và nốt hương gỗ ấm áp, tạo nên một cảm... | | <input checked="" type="checkbox"/> |
| 666400505a3db59b521fb50 | Nước Hoa Nam | Nước hoa nam này mang đến một hương thơm nam tính và quyến rũ, kết hợp giữa sự tươi mát của cam berg... | | <input checked="" type="checkbox"/> |
| 66689f8844264ca44fd9e5f8 | Nước Hoa Trẻ | ds | | <input checked="" type="checkbox"/> |

Hình 3.21: Màn hình quản lý danh mục

- Quản lý thương hiệu

Quản trị viên có thể xem thống số lượng thương hiệu, thương hiệu đang hoạt động, thương hiệu ngừng kinh doanh, thương hiệu hết hàng. Ngoài ra quản trị viên còn có thể thêm và cập nhật thương hiệu.

| ID | Tên | Mô tả | Hình ảnh | Hành Động |
|--------------------------|------------------|--|----------|-------------------------------------|
| 6629eb6d1d4442f928e1c9c8 | Chanel | Chúng tôi xin đưa ra tư vấn như sau: "Mô tả nhãn hiệu" là một mục thể hiện trong Tờ khai đăng ký nhãn... | | <input checked="" type="checkbox"/> |
| 666401104783c13a9a5bd833 | Dior | Chúng tôi xin đưa ra tư vấn như sau: "Mô tả nhãn hiệu" là một mục thể hiện trong Tờ khai đăng ký nhãn... | | <input checked="" type="checkbox"/> |
| 6668af6ac6850a764788fcdf | Parfums De Marly | Chúng tôi xin đưa ra tư vấn như sau: "Mô tả nhãn hiệu" là một mục thể hiện trong Tờ khai đăng ký nhãn... | | <input checked="" type="checkbox"/> |
| 6668af6cc6850a764788fce0 | Versace | Versace xin đưa ra tư vấn như sau: "Mô tả nhãn hiệu" là một mục thể hiện trong Tờ khai đăng ký nhãn ... | | <input checked="" type="checkbox"/> |

Hình 3.22: Màn hình quản lý thương hiệu

- Quản lý đơn hàng

Quản trị viên có thể xem thống số đơn hàng theo từng trạng thái khác nhau. Ngoài ra quản trị viên còn có thể tìm kiếm lọc theo từng tiêu chí khác nhau, xem chi tiết, cập nhật trạng thái và thêm thông tin của đơn hàng cho người dùng.

| Đơn hàng | Người đặt hàng | Ngày | Địa chỉ | Giá trị đơn hàng | TT đơn hàng | Chi tiết |
|--------------------------|--------------------------|--------------------------|---|------------------|-------------|---------------------------|
| 6669f6c31c83f26d3af01ac0 | 66648ec2d4f838e0a598251b | 2024-06-13T02:28:03.044Z | Phường An Lạc - Tỉnh Hải Dương - Thành phố Chí Linh | 2489000 | Confirmed | <button>Chi tiết</button> |
| 666754f7447a2141a7338bed | 66648ec2d4f838e0a598251b | 2024-06-11T02:33:11.961Z | Phường An Lạc - Tỉnh Hải Dương - Thành phố Chí Linh | 2489000 | Pending | <button>Chi tiết</button> |
| 666751f24a685f753fe88c80 | 66648ec2d4f838e0a598251b | 2024-06-11T02:20:18.149Z | Phường An Lạc - Tỉnh Hải Dương - Thành phố Chí Linh | 2389000 | Pending | <button>Chi tiết</button> |
| 666a03393584341e40756853 | 66648ec2d4f838e0a598251b | 2024-06-11T02:20:18.149Z | Phường An Lạc - Tỉnh Hải Dương - Thành phố Chí Linh | 1259000 | Confirmed | <button>Chi tiết</button> |

Hình 3.23: Màn hình quản lý đơn hàng

Hình 3.24: Màn hình quản lý chi tiết đơn hàng

KẾT LUẬN

1. KẾT QUẢ ĐẠT ĐƯỢC

Trong thời gian tìm hiểu, nghiên cứu cơ sở lý thuyết và triển khai ứng dụng công nghệ, đồ án đã đạt được những kết quả sau:

- Về mặt lý thuyết: Em đã nắm được các kiến thức cơ bản để xây dựng một ứng dụng website tùy biến cho các loại làm hình thông qua Restful API, sử dụng framework Express, ReactJs. Ngoài ra em cũng đã tìm hiểu thêm và ứng dụng những kiến thức về mạng để triển khai hệ thống.
- Về mặt ứng dụng: Đã xây dựng và triển khai thành công website quản lý bán nước hoa trực tuyến. Hệ thống giúp người dùng dễ tiếp cận hơn với cách sản phẩm nước hoa trên thị trường cũng như tiếp cận dễ dàng hơn với người bán hàng.
 - Hệ thống có thể giải quyết được vấn đề đã đề ra, cơ bản hoàn thành đúng như hệ thống được thiết kế, tiến độ đặt ra ban đầu.
 - Giao diện sử dụng không có quá nhiều thao tác phức tạp, giúp cho người dùng dễ dàng sử dụng.
 - Hệ thống có đủ các chức năng cho tất cả các tác nhân như Khách, Khách hàng, Quản trị viên.

2. NHỮNG VẤN ĐỀ CHƯA GIẢI QUYẾT

Ngoài những mặt đã đạt được, ứng dụng vẫn còn một số nhược điểm cần được cải thiện như:

- Chưa tích hợp được các ứng dụng giao hàng tiện lợi hiện nay để tính phí giao hàng thuận tiện cho người dùng.
- Chưa áp các thuật toán AI hiện tại giúp người dùng cải thiện chất lượng mua hàng.
- Tốc độ truy cập hệ thống vẫn đang còn chậm.
- Website chưa thể hoạt động tốt khi có nhiều người cùng truy cập vào cùng 1 thời điểm.
- Chưa có các chức năng giúp người dùng có thể trao đổi trực tiếp với người bán hàng.

3. HƯỚNG PHÁT TRIỂN

Với những mặt hạn chế đã kết luận, ứng dụng cần có những cải thiện trong tương lai như sau:

- Nâng cao tốc độ xử lý của ứng dụng web bằng việc tối ưu cấu hình, xử lý cơ sở dữ liệu để cải thiện tốc độ.
- Kết hợp thêm các hệ thống gợi ý sản phẩm và hệ thống chăm sóc khách hàng tự động để khách hàng có trải nghiệm tốt hơn.
- Xử lý khi có nhiều người cùng truy cập 1 lúc.

TÀI LIỆU THAM KHẢO

TÀI LIỆU TIẾNG VIỆT

- [1] Lê Thị Mỹ Hạnh, Giáo trình môn công nghệ phần mềm, Khoa Công nghệ Thông tin, Đại học Bách Khoa, Đại học Đà Nẵng.
- [2] Thiết kế RESTful APIs, <https://techtalk.vn/thiet-ke-restful-apis.html>

TÀI LIỆU TIẾNG ANH

- [1] Nguyen Thanh Binh, Vo Duc An, Le Thi My Hanh, Object Oriented Analysis and Design, Faculty of Information and Technology, Da Nang University of Technology: Book.
- [2] The Unified Modeling Language, <https://www.uml-diagrams.org>
- [3] Javascript, <https://vi.wikipedia.org/wiki/JavaScript>
- [4] Node.js, <https://nodejs.org/en>
- [5] ExpressJs, <https://expressjs.com>
- [6] ReactJs, <https://react.dev>
- [7] ETS homepage, <https://www.ets.org/>