

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 HỆ THỐNG QUẢN LÝ KINH DOANH XE Ô TÔ, XE MÁY ĐIỆN TÍCH HỢP AI CHATBOT

Người hướng dẫn: ThS. Đỗ Thị Tuyết Hoa

Sinh viên thực hiện:

Lâm Nhật Minh

MSSV: 102210262

Lớp: 21TCLC_DT4

Ngô Đình Phước

MSSV: 102210226

Lớp: 21TCLC_DT3

Đà Nẵng, 3/2025

TÓM TẮT

Tên đề tài: Xây dựng hệ thống quản lý kinh doanh xe ô tô, xe máy điện tích hợp chatbot AI.

Sinh viên thực hiện: Lâm Nhật Minh

Số thẻ SV: 102210262 Lớp: 21TCLC_DT4

Sinh viên thực hiện: Ngô Đình Phước

Số thẻ SV: 102210226 Lớp: 21TCLC_DT3

Hệ thống được xây dựng là một nền tảng quản lý và kinh doanh xe ô tô, xe máy điện đa biến thể, tích hợp các công nghệ hiện đại nhằm hỗ trợ doanh nghiệp tối ưu hóa hoạt động vận hành, bán hàng và chăm sóc khách hàng. Hệ thống cho phép quản lý chi tiết sản phẩm theo từng biến thể (màu sắc, tải trọng, công suất...), quản lý kho hàng theo từng lô nhập và theo dõi trạng thái tồn kho một cách chính xác. Quy trình xử lý đơn hàng được phân luồng rõ ràng qua nhiều trạng thái (chờ xử lý, xác nhận, xuất kho, bàn giao, vận chuyển, giao thành công hoặc thất bại và hủy đơn), đồng thời phân quyền thao tác chặt chẽ theo vai trò người dùng như quản trị viên, nhân viên bán hàng, quản lý kho và nhân viên giao hàng. Bên cạnh đó, hệ thống còn cung cấp chức năng thống kê – báo cáo chi tiết về doanh thu, đơn hàng, khách hàng, sản phẩm, giúp nhà quản lý đưa ra quyết định hiệu quả. Các tính năng hỗ trợ như quản lý chương trình khuyến mãi, chăm sóc khách hàng thân thiết (mã giảm giá, quà sinh nhật), tích hợp thanh toán điện tử (ZaloPay, PayOS) và chatbot AI hỗ trợ tư vấn 24/7 giúp nâng cao trải nghiệm người dùng và tăng năng lực cạnh tranh của doanh nghiệp trong thời đại số hóa.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Lâm Nhật Minh

Số thẻ sinh viên: 102210262

Lớp: 21TCLC_DT4

Họ tên sinh viên: Ngô Đình Phước

Số thẻ sinh viên: 102210226

Lớp: 21TCLC_DT3

Khoa: Công nghệ thông tin

Ngành: Công nghệ thông tin

- Tên đề tài đồ án:* Xây dựng hệ thống quản lý kinh doanh xe ô tô, xe máy điện tích hợp chatbot AI.
- Đề tài thuộc diện:* Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện
- Các số liệu và dữ liệu ban đầu:* Không có
- Nội dung các phần thuyết minh và tính toán:*
 - Mở đầu: Giới thiệu bối cảnh và lí do chọn đề tài, mục đích của đề tài, phạm vi, đối tượng sử dụng đề tài
 - Chương 1: Cơ sở lý thuyết và công cụ sử dụng
 - Chương 2: Phân tích và thiết kế hệ thống
 - Chương 3: Xây dựng và triển khai hệ thống
 - Chương 4: Kết luận và hướng phát triển
- Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):*
 - Biểu đồ usecase và đặc tả usecase
 - Biểu đồ hoạt động
 - Bảng thiết kế cơ sở dữ liệu
 - Sơ đồ tổng quan hệ thống
 - Sơ đồ tổng quan luồng dữ liệu của hệ thống chatbot AI
- Họ tên người hướng dẫn:* ThS. Đỗ Thị Tuyết Hoa
- Ngày giao nhiệm vụ đồ án:* 01/04/ 2025
- Ngày hoàn thành đồ án:* 17/06/ 2025

Đà Nẵng, ngày 18 tháng 06, năm 2025

Trưởng Bộ môn

Người hướng dẫn

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến quý thầy cô Khoa Công nghệ Thông tin cùng toàn thể giảng viên Trường Đại học Bách khoa – Đại học Đà Nẵng. Trong suốt thời gian học tập và rèn luyện tại trường, chúng em đã được truyền đạt không chỉ những kiến thức chuyên môn quý báu mà còn cả tinh thần học hỏi, sự tận tụy và đạo đức nghề nghiệp từ quý thầy cô. Những điều đó chính là nền tảng vững chắc giúp chúng em trưởng thành và phát triển.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc đến cô ThS. Đỗ Thị Tuyết Hoa – giảng viên hướng dẫn đã luôn tận tình chỉ bảo, đồng hành cùng chúng em trong suốt quá trình thực hiện đề án tốt nghiệp. Sự hướng dẫn tận tâm, những góp ý quý giá và sự khích lệ kịp thời của cô chính là nguồn động lực lớn giúp chúng em hoàn thành đề án đúng tiến độ và tích lũy thêm nhiều kiến thức thực tiễn quý báu.

Mặc dù đã nỗ lực hoàn thành đề án với tinh thần trách nhiệm cao nhất, nhưng do thời gian có hạn và kiến thức thực tế còn hạn chế, chắc chắn không tránh khỏi những thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp, nhận xét từ quý thầy cô và các bạn để đề án được hoàn thiện hơn.

Một lần nữa, chúng em xin chân thành cảm ơn!

LỜI CAM ĐOAN

Chúng em xin cam đoan rằng:

1. Toàn bộ nội dung trong đồ án này là kết quả do chính chúng em thực hiện dưới sự hướng dẫn trực tiếp và tận tình của cô ThS. Đỗ Thị Tuyết Hoa.
2. Các tài liệu, số liệu, hình ảnh và thông tin tham khảo sử dụng trong quá trình thực hiện đồ án đều được trích dẫn rõ ràng, đầy đủ về tên tác giả, tên công trình, thời gian và nguồn công bố theo đúng quy định.
3. Nếu phát hiện có bất kỳ hành vi sao chép không hợp lệ hoặc vi phạm quy chế đào tạo, chúng em xin hoàn toàn chịu trách nhiệm trước Hội đồng chấm đồ án và trước quy định của nhà trường.

Chúng em xin chân thành cam kết và chịu trách nhiệm với những gì đã trình bày trong đồ án này.

MỤC LỤC

LỜI CẢM ƠN.....	1
LỜI CAM ĐOAN.....	2
MỤC LỤC	3
DANH SÁCH CÁC HÌNH ẢNH	5
DANH SÁCH CÁC BẢNG BIỂU	7
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT.....	8
MỞ ĐẦU	9
1. Bối cảnh và lý do chọn đề tài	9
2. Mục tiêu và nhiệm vụ của đề tài.....	9
2.1. Mục tiêu	9
2.2. Nhiệm vụ.....	9
3. Nội dung của đề tài.....	11
4. Phạm vi sử dụng đề tài	11
5. Bố cục của đề tài.....	11
PHÂN CÔNG CÔNG VIỆC	13
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG.....	14
1.1. Tổng quan về NestJS	14
1.2. Tổng quan về ReactJS.....	16
1.3. Tổng quan về NextJS.....	20
1.4. Tổng quan về RestfulAPI	23
1.5. Tổng quan về n8n	23
1.6. Tổng quan về RAG	24
1.7. Công nghệ sử dụng	25
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	26
2.1 Tác nhân.....	26
2.1. Phân tích chức năng.....	27
2.1.1. Admin.....	27
2.1.2. Nhân viên bán hàng.....	29
2.1.3. Nhân viên quản lí kho	30
2.1.4. Nhân viên giao hàng.....	30
2.1.5. Khách hàng.....	30
2.1.6. Chat với ChatbotAI	31
2.2. Sơ đồ usecase	32

2.2.1. Admin quản lí đơn hàng.....	33
2.2.2. Admin quản lí sản phẩm	35
2.2.3. Admin thống kê.....	38
2.2.4. Admin quản lí phân quyền	40
2.2.5. Admin quản lí voucher.....	42
2.2.6. User quản lí thông tin cá nhân	44
2.2.7. User quản lí địa chỉ đặt hàng.....	46
2.2.8. User quản lí giỏ hàng	47
2.2.9. User quản lí đơn hàng	48
2.3. Sơ đồ hoạt động	50
2.3.1. Khách hàng.....	50
2.3.2. Quản trị viên admin.....	51
2.3.3. Nhân viên bán hàng.....	52
2.3.4. Quản lí kho	53
2.3.5. Nhân viên giao hàng.....	54
2.4. Thiết kế cơ sở dữ liệu	54
2.4.1. Mô hình thiết kế cơ sở dữ liệu	54
2.4.2. Mô tả bảng.....	56
2.5. Kiến trúc hệ thống.....	78
2.5.1. Sơ đồ tổng quan hệ thống.....	78
2.5.2. Sơ đồ tổng quan hệ thống AI Chatbot.....	79
CHƯƠNG 3: TRIỂN KHAI VÀ CHẠY THỬ NGHIỆM	86
3.1. Môi trường triển khai.....	86
3.1.1. Quản lí mã nguồn	86
3.1.2. Môi trường cài đặt	86
3.2. Hướng dẫn sử dụng	87
3.2.1. Backend	87
3.3. Frontend.....	87
3.4. Kết quả demo.....	88
3.4.1. Giao diện website admin	88
3.4.2. Giao diện website user.....	98
CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN	104
4.1. Kết quả đạt được.....	104
4.2. Hạn chế và vấn đề chưa giải quyết.....	105
4.3. Hướng phát triển.....	105
TÀI LIỆU THAM KHẢO	107

DANH SÁCH CÁC HÌNH ẢNH

Hình 1.1. Cấu trúc của NestJS.....	14
Hình 1.2. Component trong ReactJS	17
Hình 1.3. DOM ảo trong ReactJS.....	17
Hình 1.4. JSX của ReactJS	18
Hình 1.5. Server Side Rendering của NextJS.....	20
Hình 1.6. Client Side Rendering của NextJS	21
Hình 1.7. Static Generation Side của NextJS	21
Hình 1.8. Mô tả Restful API.....	23
Hình 1.9. Quy trình hoạt động RAG	25
Hình 2.1. Sơ đồ usecase tổng quát.....	32
Hình 2.2. Usecase admin quản lý đơn hàng	33
Hình 2.3. Usecase admin quản lý sản phẩm.....	35
Hình 2.4. Usecase admin thống kê	38
Hình 2.5. Usecase admin quản lý role.....	40
Hình 2.6. Usecase admin quản lý voucher	42
Hình 2.7. Usecase user quản lý profile.....	44
Hình 2.8. Usecase user quản lý địa chỉ đặt hàng.....	46
Hình 2.9. Usecase user quản lý giỏ hàng	47
Hình 2.10. Usecase user quản lý đơn hàng	48
Hình 2.11. Sơ đồ hoạt động của khách hàng	50
Hình 2.12. Sơ đồ hoạt động của admin	51
Hình 2.13. Sơ đồ hoạt động của nhân viên bán hàng.....	52
Hình 2.14. Sơ đồ hoạt động của quản lý kho	53
Hình 2.15. Sơ đồ hoạt động của nhân viên giao hàng.....	54
Hình 2.16. ERD	56
Hình 2.17. Sơ đồ tổng quan hệ thống	78
Hình 2.18. Luồng xử lý khi có file được tạo của chatbot AI	79
Hình 2.19. Luồng thêm lịch hẹn.....	81
Hình 2.20. Luồng xử lý dữ liệu đầu vào và phản hồi về web của Chatbot AI.....	83
Hình 3.1 Giao diện đăng nhập admin.....	88
Hình 3.2. Tổng quan.....	89
Hình 3.3 Màn hình nhân viên	89
Hình 3.4. Giao diện khách hàng	90
Hình 3.5. Giao diện quản lý sản phẩm	90
Hình 3.6. Giao diện biến thể.....	91
Hình 3.7. Giao diện danh mục sản phẩm.....	91
Hình 3.8. Giao diện danh mục tin tức	92
Hình 3.9. Quản lý đơn hàng	92
Hình 3.10. Admin quản lý kho	93
Hình 3.11 Giao diện nhà cung cấp	93
Hình 3.12. Giao diện nhập kho.....	94
Hình 3.13. Admin chọn biến thể để nhập vào kho	94
Hình 3.14. Admin nhập biến thể vào nhiều kho khác nhau	95
Hình 3.15. Giao diện xuất kho.....	96

Hình 3.16. Admin chọn đơn hàng để xuất kho.....	96
Hình 3.17. Admin nhập thông tin xuất kho cho mỗi đơn hàng.....	97
Hình 3.18. Giao diện cấu hình.....	97
Hình 3.19. Giao diện đăng kí và đăng nhập.	98
Hình 3.20. Giao diện trang chủ	99
Hình 3.21. Giao diện danh sách sản phẩm	99
Hình 3.22. Giao diện chi tiết sản phẩm	100
Hình 3.23. Giao diện tin tức	100
Hình 3.24. Giao diện giỏ hàng.....	101
Hình 3.25. Giao diện thanh toán.....	102
Hình 3.26. Giao diện lịch sử đơn hàng.....	102
Hình 3.27. Giao diện tài khoản khách hàng	103
Hình 3.28. Giao diện quản lí địa chỉ.....	103
Hình 3.29. Giao diện chatbot AI	104

DANH SÁCH CÁC BẢNG BIỂU

Bảng 2.1. Các tác nhân của hệ thống.....	26
Bảng 2.2. Đặc tả usecase admin quản lý đơn hàng	33
Bảng 2.3. Đặc tả usecase admin quản lý sản phẩm.....	35
Bảng 2.4. Đặc tả usecase admin thống kê	38
Bảng 2.5. Đặc tả usecase admin quản lý phân quyền	40
Bảng 2.6. Đặc tả usecase admin quản lý voucher	42
Bảng 2.7. Đặc tả usecase user quản lý profile.....	45
Bảng 2.8. Đặc tả usecase user quản lý địa chỉ đặt hàng.....	46
Bảng 2.9. Đặc tả usecase user quản lý giỏ hàng	47
Bảng 2.10. Đặc tả usecase user quản lý đơn hàng	49
Bảng 2.11. Mô tả bảng sản phẩm	56
Bảng 2.12. Mô tả bảng brand	57
Bảng 2.13. Mô tả bảng Categories	58
Bảng 2.14. Mô tả bảng Specification	58
Bảng 2.15. Mô tả bảng skus	59
Bảng 2.16. Mô tả bảng Option	60
Bảng 2.17. Mô tả bảng Option_Value.....	60
Bảng 2.18. Mô tả bảng Import.....	61
Bảng 2.19. Mô tả bảng chi tiết nhập kho.....	61
Bảng 2.20. Mô tả bảng Warehouse	62
Bảng 2.21. Mô tả bảng Export.....	63
Bảng 2.22. Mô tả bảng Detail Export.....	63
Bảng 2.23. Mô tả bảng blog	64
Bảng 2.24. Mô tả bảng Blog_Category	65
Bảng 2.25. Mô tả bảng Review	65
Bảng 2.26. Mô tả bảng Role.....	66
Bảng 2.27. Mô tả bảng Permission.....	67
Bảng 2.28. Mô tả bảng User.....	68
Bảng 2.29. Mô tả bảng User_have_roles.....	69
Bảng 2.30. Mô tả bảng Role_Permission	69
Bảng 2.31. Mô tả bảng Customer.....	70
Bảng 2.32. Mô tả bảng Receive_Address	71
Bảng 2.33. Mô tả bảng Orders.....	72
Bảng 2.34. Mô tả bảng Order_Details.....	73
Bảng 2.35. Mô tả bảng Voucher.....	74
Bảng 2.36. Mô tả bảng kiểu TypeVoucher.....	75
Bảng 2.37. Mô tả bảng Customer_Voucher	75
Bảng 2.38. Mô tả bảng Cart.....	76
Bảng 2.39. Mô tả bảng Cart_Item	76
Bảng 2.40. Mô tả bảng Delivery_Method.....	77
Bảng 2.41. Mô tả bảng Payment_Method.....	77
Bảng 3.1. Môi trường cài đặt.....	86
Bảng 3.2. Môi trường triển khai thực tế.....	86

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Từ	Viết tắt của	Diễn 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
DB	Database	Cơ sở dữ liệu
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
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
SEO	Search Engine Optimization	Tối ưu hóa công cụ tìm kiếm
DOM	Document Object Model	Mô hình đối tượng tài liệu
UI	User Interface	Giao diện người dùng

MỞ ĐẦU

1. Bối cảnh và lý do chọn đề tài

Trong bối cảnh công nghệ phát triển vượt bậc và dịch vụ mua sắm trực tuyến đang trở thành một trong những xu hướng tất yếu, các doanh nghiệp kinh doanh xe ô tô và xe máy điện đang đối mặt với không ít thách thức trong việc quản lý sản phẩm, quản lý kho hàng và tối ưu hóa quy trình bán hàng. Sự gia tăng nhanh chóng về số lượng mẫu mã, màu sắc, phiên bản, cùng với các tùy chọn đa dạng như dung lượng pin, công suất động cơ hay tính năng thông minh của nhiều loại xe ô tô và xe máy điện, đòi hỏi một hệ thống quản lý sản phẩm theo biến thể hiệu quả. Đây không chỉ là chìa khóa để nâng cao hiệu suất vận hành mà còn là yếu tố quyết định trong việc cải thiện trải nghiệm khách hàng, từ đó giúp doanh nghiệp duy trì vị thế trong thị trường cạnh tranh khốc liệt.

Bên cạnh đó, khách hàng hiện đại ngày càng có kỳ vọng cao hơn: họ mong muốn nhận được thông tin sản phẩm tức thì, tư vấn cá nhân hóa dựa trên nhu cầu riêng biệt, và hỗ trợ liên tục 24/7 – những yêu cầu mà các phương pháp quản lý thủ công truyền thống không thể đáp ứng một cách trọn vẹn. Sự phát triển mạnh mẽ của công nghệ trí tuệ nhân tạo (AI), đặc biệt là các ứng dụng như chatbot thông minh, đã mở ra cơ hội đột phá để các doanh nghiệp tối ưu hóa quy trình kinh doanh. Một chatbot được tích hợp AI có khả năng phản hồi nhanh chóng, tư vấn chi tiết về các dòng xe.

Vì vậy, việc xây dựng một hệ thống quản lý và kinh doanh xe ô tô, xe máy điện tích hợp chatbot thông minh không chỉ giúp doanh nghiệp vận hành hiệu quả hơn mà còn mang lại trải nghiệm vượt trội cho khách hàng. Đây chính là giải pháp chiến lược giúp doanh nghiệp tạo dựng lợi thế cạnh tranh bền vững, đáp ứng kịp thời nhu cầu thị trường và khẳng định vị thế trong kỷ nguyên số hóa.

2. Mục tiêu và nhiệm vụ của đề tài

2.1. Mục tiêu

- Xây dựng hệ thống quản lý kho chi tiết theo từng lô hàng.
- Phát triển mô hình quản lý sản phẩm và các biến thể.
- Thiết kế hệ thống phân quyền theo vai trò người dùng quản trị hệ thống.
- Quản lý đơn hàng linh hoạt, rõ ràng theo từng giai đoạn xử lý và vai trò người dùng.
- Tích hợp trí tuệ nhân tạo (AI) thông qua chatbot hỗ trợ khách hàng.
- Áp dụng thanh toán điện tử đa phương thức.
- Phân tích và thống kê dữ liệu phục vụ ra quyết định kinh doanh.
- Mở rộng khả năng áp dụng cho các ngành hàng khác có sản phẩm đa biến thể.

2.2. Nhiệm vụ

- Quản lý kho theo lô hàng:

- Xây dựng hệ thống nhập – xuất kho phân theo từng lô hàng.
- Theo dõi trạng thái tồn kho (còn hàng, sắp hết, hết hàng).
- Quản lý sản phẩm và biến thể linh hoạt:
 - Thiết kế mô hình dữ liệu cho sản phẩm gồm: màu sắc, tải trọng, công suất, ...
 - Quản lý giá cả, thuộc tính và tình trạng của từng biến thể.
- Phân quyền theo vai trò người dùng:
 - Phân chia chức năng hệ thống cho từng vai trò: quản trị viên, quản lý kho, nhân viên bán hàng, giao hàng.
 - Đảm bảo người dùng chỉ truy cập các phần liên quan đến nhiệm vụ của mình.
- Xây dựng hệ thống quản lý trạng thái đơn hàng và phân quyền thao tác
 - Thiết kế luồng xử lý đơn hàng gồm các trạng thái rõ ràng:
Đang xử lý → Đã xác nhận → Đã xuất kho → Đã bàn giao cho đơn vị vận chuyển → Đang vận chuyển → Đang giao hàng → Đã giao hàng / Giao hàng thất bại
 - Thiết kế các quyền quản lý, giám sát đơn hàng rõ ràng, cụ thể cho từng vai trò của người dùng quản trị hệ thống
- Tích hợp AI chatbot:
 - Xây dựng chatbot thông minh trả lời câu hỏi thường gặp.
 - Duy trì phản hồi 24/7 giúp tăng tính chuyên nghiệp.
- Tích hợp thanh toán điện tử:
 - Kết nối với các nền tảng thanh toán như ZaloPay, PayOS,...
 - Tạo đơn hàng và xử lý thanh toán trực tiếp trên hệ thống.
- Thống kê – phân tích dữ liệu:
 - Thống kê doanh thu, lợi nhuận, số lượng khách hàng, đơn hàng, sản phẩm, nhà cung cấp
 - Báo cáo theo trạng thái đơn hàng, thời gian
- Chăm sóc khách hàng – khuyến mãi:
 - Tạo mã giảm giá, quà tặng sinh nhật.
 - Quản lý chương trình khách hàng thân thiết và khuyến khích mua lại.

- Khả năng mở rộng hệ thống: Không giới hạn ở ngành xe ô tô, xe máy điện – có thể triển khai cho thời trang, đồ điện tử, nội thất,..., các ngành mà có sử dụng sản phẩm đa biến thể

3. Nội dung của đề tài

Những nội dung công việc cần thực hiện để hoàn thành được mục tiêu và nhiệm vụ đề ra của đề tài bao gồm:

- Phân tích và mô hình hóa yêu cầu hệ thống bằng công cụ UML
- Phân tích và thiết kế kiến trúc hệ thống thông tin
- Thiết kế cơ sở dữ liệu cho hệ thống
- Nghiên cứu và sử dụng các công nghệ phù hợp để xây dựng hệ thống
- Xây dựng hệ thống các chức năng chính
- Viết báo cáo tổng kết quá trình nghiên cứu, thiết kế và xây dựng hệ thống, đưa ra nhận xét, đánh giá, bài học và hướng phát triển mở rộng trong tương lai.

4. Phạm vi sử dụng đề tài

- **Ngành:**
 - + Hệ thống hướng đến các doanh nghiệp siêu nhỏ, hộ kinh doanh bán lẻ/sỉ trong lĩnh vực xe hơi và xe máy điện tại Việt Nam (quy mô 1–10 người).
 - + Phù hợp với cá nhân kinh doanh cần quản lý đơn hàng, sản phẩm và khách hàng trên nền tảng trực tuyến
- **Kỹ thuật:**
 - + Phát triển hệ thống trên nền tảng website
 - + Không bao gồm quản lý vận chuyển chi tiết, tích hợp với bên vận chuyển thứ ba
 - + AI chatbot tập trung vào hỗ trợ khách hàng, không thay thế hoàn toàn nhân viên tư vấn.

5. Bố cục của đề tài

Đề án được chia làm 3 chương chính:

- Chương thứ nhất là quá trình tìm hiểu và nghiên cứu cơ sở lý thuyết, các công nghệ sử dụng để thực hiện áp dụng vào đề tài.
- Chương thứ hai đi vào phân tích và thiết kế hệ thống thông tin, xác định các tác nhân chính, phân tích các chức năng của từng tác nhân trong hệ thống, thiết kế và xây dựng cơ sở dữ liệu, thiết kế luồng hệ thống AI chatbot
- Chương thứ ba là 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.

PHÂN CÔNG CÔNG VIỆC

Ngô Đình Phước	<ul style="list-style-type: none">- Xây dựng database- Viết API đăng nhập, đăng kí, kích hoạt tài khoản, quên mật khẩu, giỏ hàng, sản phẩm, user, customer, nhập kho, phân quyền, role, voucher, thương hiệu, tin tức, danh mục, danh mục tin tức, nhãn hàng, đơn hàng, kho hàng...- Thiết kế giao diện user trang đăng nhập, đăng kí, quên mật khẩu, kích hoạt tài khoản, chi tiết sản phẩm, giỏ hàng, thanh toán, thông tin người dùng, lịch sử đơn hàng, quản lý địa chỉ nhận hàng, phân quyền, ...- Nghiên cứu và xây dựng chatbot AI- Triển khai website user, chatbot AI và đánh giá, bảo trì- Làm slide, viết báo cáo
Lâm Nhật Minh	<ul style="list-style-type: none">- Xây dựng database- Viết API thanh toán, sản phẩm, biến thể sản phẩm, thống kê, đơn hàng, địa chỉ các xã, huyện, quận, thành phố trong việt nam, nhập kho, xuất kho, danh mục, danh mục tin tức, tin tức, nhãn hàng, nhân viên, khách hàng, quyền hạn, ...- Thiết kế giao diện admin layout, thống kê, sản phẩm, biến thể sản phẩm, đơn hàng, nhãn hàng cung cấp, nhập kho, xuất kho, danh mục, danh mục tin tức, tin tức, quyền hạn, nhân viên, khách hàng, ...- Thiết kế giao diện user layout, trang chủ, trang tin tức, giới thiệu cửa hàng, tìm kiếm sản phẩm, ...- Thiết kế responsive tương thích với các thiết bị tương ứng- Nghiên cứu và xây dựng chatbot AI- Triển khai website admin và be, và đánh giá, bảo trì- Làm slide, viết báo cáo

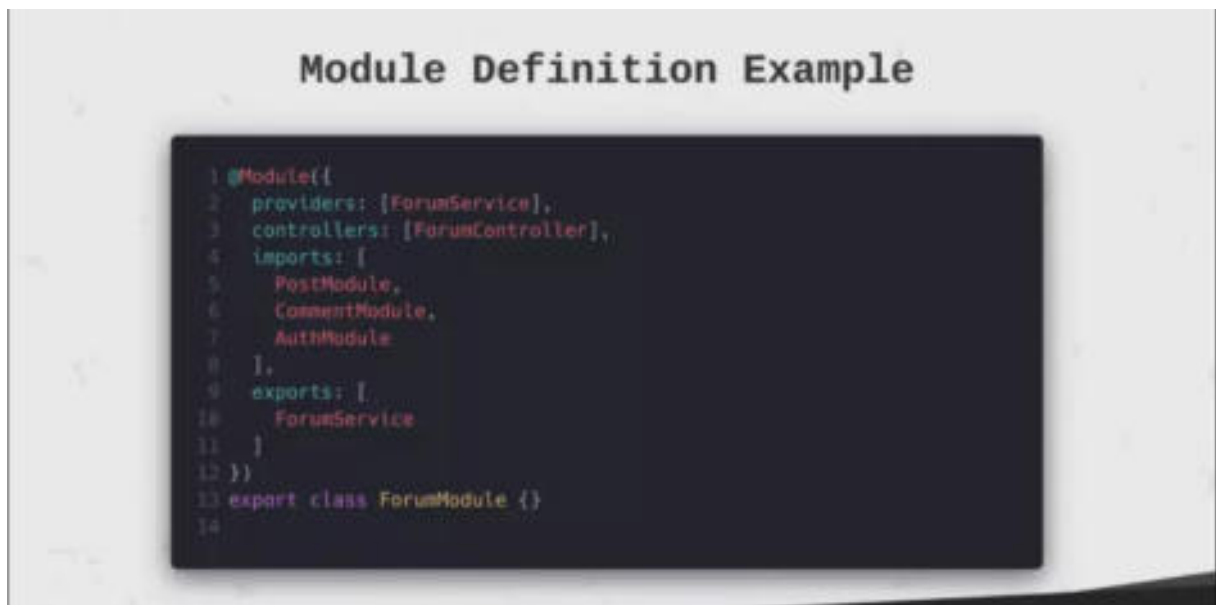
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG

1.1. Tổng quan về NestJS

NestJS là một framework mã nguồn mở để phát triển ứng dụng server-side (backend applications) bằng ngôn ngữ TypeScript hoặc JavaScript. Nó được xây dựng trên cơ sở của Node.js và sử dụng các khái niệm từ TypeScript để tạo ra một môi trường phát triển hiện đại và mạnh mẽ cho việc xây dựng các ứng dụng web và API.

Mục tiêu chính của NestJS là cung cấp một cấu trúc ứng dụng rõ ràng và dễ quản lý, giúp tăng tính bảo trì và sự tổ chức trong mã nguồn. Để đạt được điều này, NestJS triển khai mô hình kiến trúc lõi (core architecture) dựa trên các nguyên tắc của Angular, đặc biệt là sử dụng Dependency Injection (DI) và Modules (Các module).

❖ Cấu trúc của NestJS



Hình 1.1. Cấu trúc của NestJS

Cấu trúc của NestJS được xây dựng dựa trên mô hình kiến trúc lõi (core architecture) giúp tạo ra một ứng dụng server-side (backend application) rõ ràng, dễ quản lý và dễ mở rộng. Cấu trúc NestJS thường được tổ chức thành các phần chính sau:

Module (Các module): Module là một phần cơ bản trong cấu trúc NestJS. Mỗi ứng dụng NestJS bao gồm ít nhất một module gốc (root module) và có thể có nhiều module con. Module là nơi tổ chức các thành phần của ứng dụng như Controllers, Providers và các thành phần khác. Mỗi module đại diện cho một phần chức năng cụ thể của ứng dụng.

Controller (Bộ điều khiển): Controllers là thành phần chịu trách nhiệm xử lý các yêu cầu HTTP từ phía client và trả về kết quả tương ứng. Controllers là nơi xử lý các request và trả về các response. Các phương thức của controller được chú thích

(decorated) bằng các decorator như `@Get()`, `@Post()`, `@Put()`, v.v., để chỉ định các route và phương thức HTTP tương ứng.

Provider (Các nhà cung cấp): Providers là thành phần chịu trách nhiệm cung cấp các dịch vụ cho ứng dụng. Đây có thể là các service, repository, logger, v.v. Providers sử dụng dependency injection để chèn vào các thành phần khác và có thể được sử dụng bởi các controllers hoặc các providers khác.

Middleware (Trung gian): Middleware là các hàm xử lý mà NestJS sử dụng để xử lý các yêu cầu HTTP trước khi chúng đến các route xử lý chính. Middleware có thể được sử dụng để thực hiện các thao tác chung như xác thực, ghi log, xử lý lỗi, v.v.

Filter (Bộ lọc): Filters được sử dụng để xử lý các exception (ngoại lệ) xảy ra trong ứng dụng. Filters cho phép bạn xử lý và thay đổi response trước khi gửi về client khi có exception xảy ra.

Guard (Bảo vệ): Guards được sử dụng để kiểm tra xem một yêu cầu có thể được xử lý hoặc không. Guards cho phép bạn thực hiện các kiểm tra xác thực hoặc kiểm tra quyền trước khi xử lý một yêu cầu.

Interceptor (Bộ chặn): Interceptors là các hàm xử lý mà NestJS sử dụng để chặn và thay đổi response trước khi nó được gửi về client. Interceptors có thể được sử dụng để thực hiện các thao tác chung trên response trước khi nó đi ra ngoài.

Exception (Ngoại lệ): Exception handling (xử lý ngoại lệ) là một phần quan trọng của cấu trúc NestJS. Exception handling cho phép bạn xử lý các exception xảy ra trong ứng dụng và trả về các thông báo lỗi thích hợp cho client.

❖ Ưu và nhược điểm của NestJS:

- Ưu điểm:

- + **Mở rộng và hiệu quả:** NestJS cho phép xây dựng các ứng dụng mở rộng, nhanh chóng và hiệu quả, đồng thời dễ dàng bảo trì. Nó cung cấp cấu trúc rõ ràng và các khái niệm như modules, providers và controllers để tạo ra mã linh hoạt và dễ mở rộng.
- + **Sử dụng TypeScript:** NestJS được viết bằng TypeScript, cho phép sử dụng các tính năng nâng cao của TypeScript và kiểm tra lỗi tĩnh để phát hiện lỗi sớm hơn trong quá trình phát triển.
- + **Tài liệu đầy đủ:** NestJS đi kèm với tài liệu phong phú và hướng dẫn chi tiết, giúp người dùng nắm bắt nhanh chóng các khái niệm và tính năng của framework.
- + **Hỗ trợ cho các loại ứng dụng:** NestJS hỗ trợ xây dựng các loại ứng dụng như REST API, MVC, Microservices, Websockets, GraphQL và CRON Job, đáp ứng nhu cầu phát triển đa dạng của các dự án.

- + **Tích hợp với Angular:** NestJS lấy cảm hứng từ cấu trúc của Angular, giúp người dùng quen thuộc với Angular dễ dàng chuyển sang phát triển ứng dụng backend với NestJS.
- + **Kiểm thử và bảo trì dễ dàng:** Nhờ cấu trúc module và dependency injection, NestJS cho phép kiểm thử và bảo trì các thành phần của ứng dụng một cách độc lập và dễ dàng.
- **Nhược điểm:**
 - + **Khả năng học và sử dụng ban đầu:** NestJS đòi hỏi kiến thức khá cao, đồng thời cần mất một thời gian để hiểu và làm quen với cú pháp và cấu trúc của nó. Điều này có thể tạo ra một rào cản đối với những người mới bắt đầu sử dụng framework này.
 - + **Quá phụ thuộc vào TypeScript:** Mặc dù TypeScript mang lại nhiều lợi ích, nhưng NestJS yêu cầu sử dụng TypeScript, điều này có thể là rào cản đối với những người đã quen với việc sử dụng JavaScript thuần.
 - + **Có thể có hiệu suất chậm hơn:** Mặc dù NestJS là một framework nhanh chóng, nhưng do tính phức tạp và cấu trúc của nó, có thể xảy ra hiệu suất chậm hơn so với một số framework Node.js đơn giản khác như Express.
 - + **Quá mạnh mẽ cho các ứng dụng nhỏ:** Đối với các dự án nhỏ và đơn giản, việc sử dụng NestJS có thể là quá mức và không cần thiết. Framework này thích hợp hơn cho các ứng dụng lớn và phức tạp hơn.

1.2. Tổng quan về ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở do **Facebook phát triển**, dùng để xây dựng **giao diện người dùng (UI)**, đặc biệt là các ứng dụng web đơn trang (**SPA – Single Page Application**).


React hoạt động theo mô hình **component-based**, cho phép chia giao diện thành các thành phần nhỏ, có thể tái sử dụng và dễ kiểm soát trạng thái.

❖ Các tính năng nổi bật của ReactJS:

Components: ReactJS cho phép phát triển ứng dụng web theo mô hình component. Các component là các phân tử UI độc lập có thể được tái sử dụng trong nhiều phần khác nhau của ứng dụng.

```
import React from 'react';
import ReactDOM from 'react-dom';

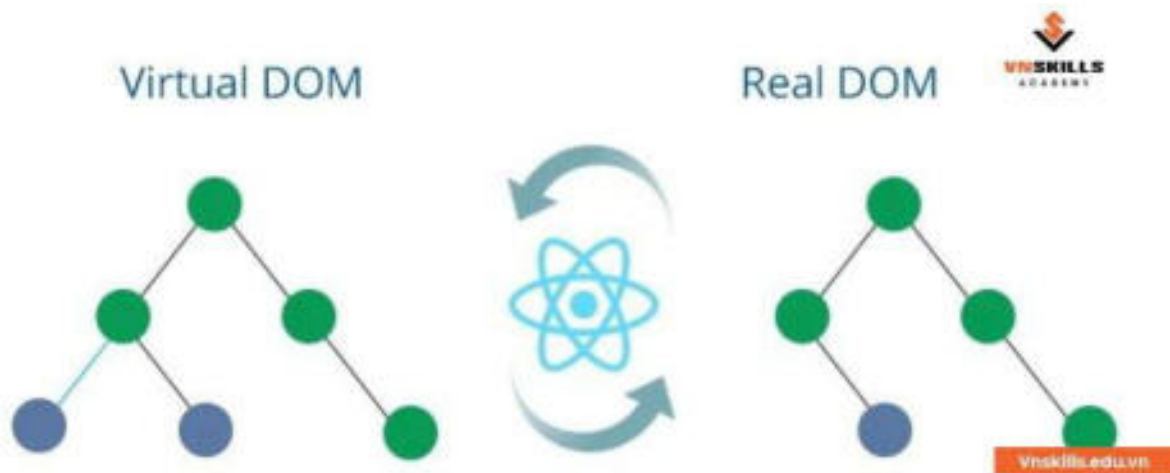
class ReactJSOverview extends React.Component {
  render() {
    return (
      <div>
        ReactJS facilitates the building of responsive user interfaces,
        able to perform well in highly scalable environment
      </div>
    );
  }
}
```



Vnskills.edu.vn

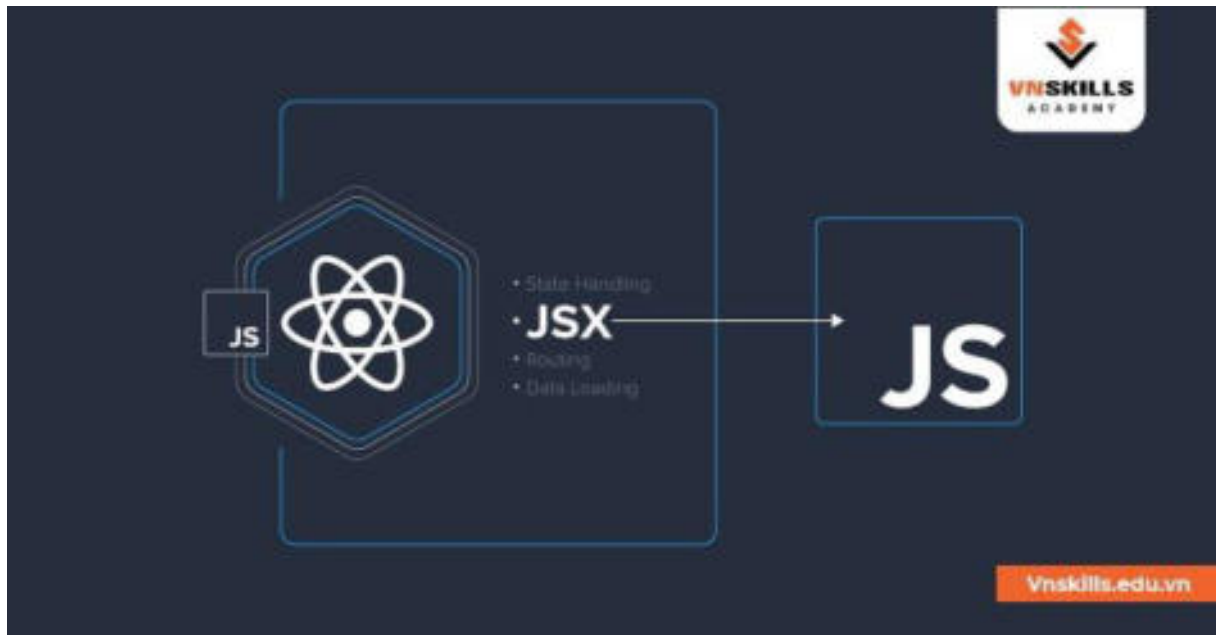
Hình 1.2. Component trong ReactJS

Virtual DOM: ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM là một bản sao của DOM được lưu trữ trong bộ nhớ và được cập nhật một cách nhanh chóng khi có thay đổi, giúp tăng tốc độ và hiệu suất của ứng dụng.



Hình 1.3. DOM ảo trong ReactJS

JSX: JSX là một ngôn ngữ lập trình phân biệt được sử dụng trong ReactJS để mô tả các thành phần UI. JSX kết hợp HTML và JavaScript, giúp cho việc viết mã dễ hiểu và dễ bảo trì hơn.



Hình 1.4. JSX của ReactJS

State và Props: ReactJS cho phép quản lý trạng thái của các thành phần UI thông qua State và Props. State là trạng thái của một thành phần được quản lý bởi nó chính, trong khi Props là các giá trị được truyền vào từ bên ngoài để tùy chỉnh hoặc điều khiển hành vi của một thành phần.

Hỗ trợ tốt cho SEO: ReactJS hỗ trợ tốt cho việc tối ưu hóa SEO. Với các thư viện như React Helmet, các nhà phát triển có thể quản lý các phần tử meta và title cho từng trang web, giúp tăng khả năng tìm kiếm và tăng cường trải nghiệm người dùng.

Hỗ trợ đa nền tảng: ReactJS không chỉ được sử dụng để phát triển ứng dụng web, mà còn được sử dụng để phát triển ứng dụng di động với React Native. Sử dụng React Native, các nhà phát triển có thể xây dựng ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn.

Redux: Redux là một thư viện quản lý trạng thái cho các ứng dụng ReactJS. Nó giúp quản lý trạng thái của ứng dụng một cách chính xác và dễ dàng, đồng thời giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.

❖ 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 nếu 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.

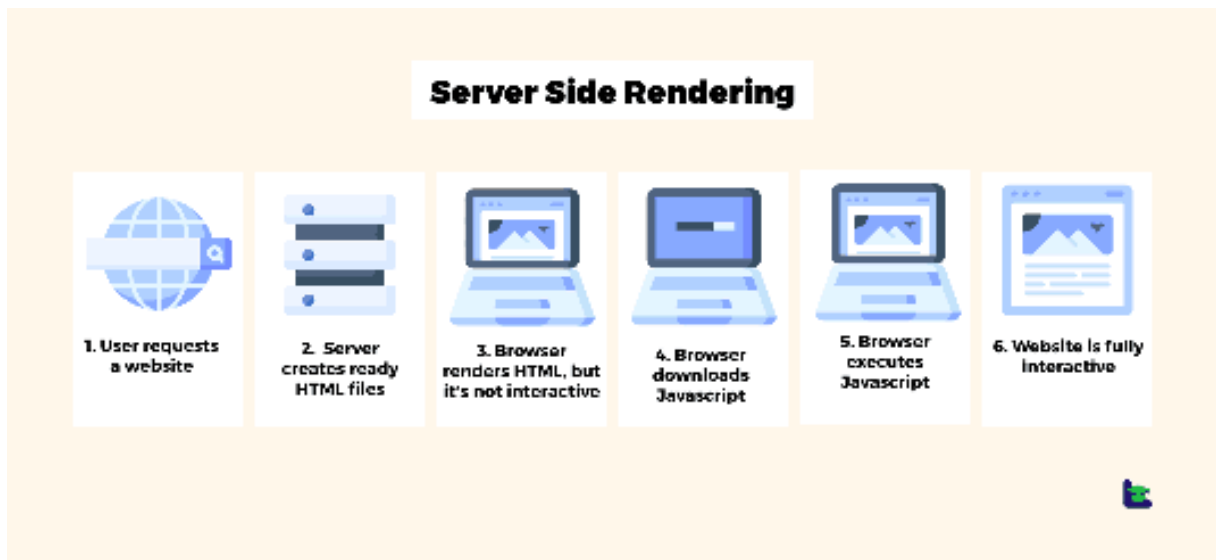
1.3. Tổng quan về NextJS

NextJS là một framework có mã nguồn mở được xây dựng trên nền tảng của React, cho phép chúng ta xây dựng các trang web tĩnh có tốc độ siêu nhanh và thân thiện với người dùng, cũng như xây dựng các ứng dụng web React.

NextJS được ra đời vào năm 2016, thuộc sở hữu của Vercel. NextJS bắt đầu trở nên phổ biến vào năm 2018 và tiếp tục tăng trưởng mạnh mẽ trong cộng đồng phát triển web. Sự kết hợp của các tính năng như Server-side Rendering (SSR) với Static Site Generation (SSG) đã giúp NextJS trở thành sự lựa chọn hấp dẫn cho nhiều dự án.

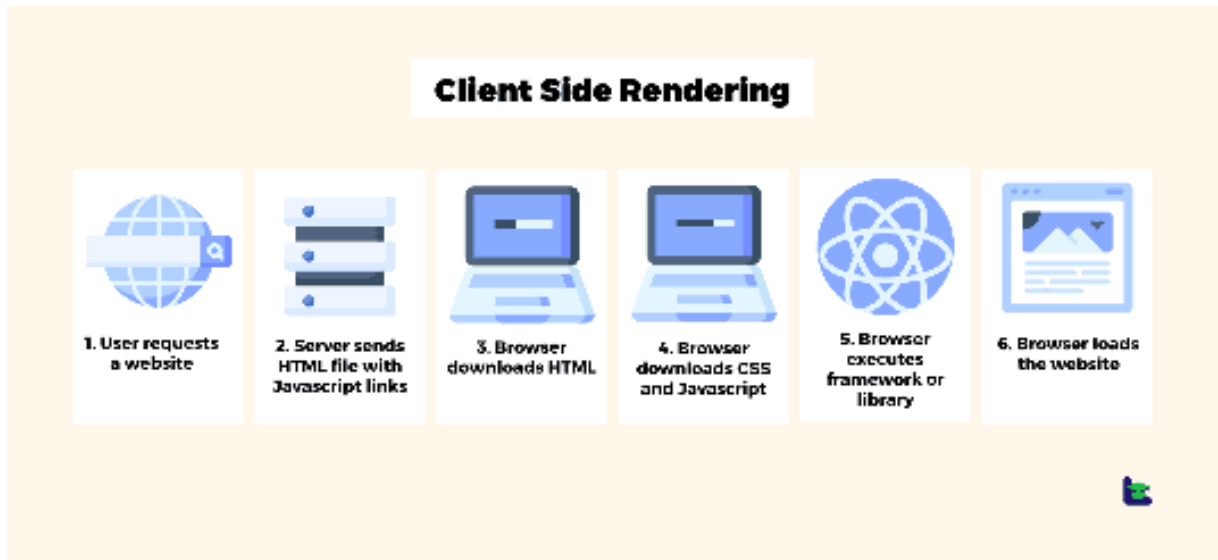
❖ Các hình thức rendering nổi bật trong NextJS:

- **SSR (Server Side Rendering):** là cách render nội dung HTML trên server rồi mới gửi về trình duyệt. Nhờ đó, trang có thể hiển thị nhanh hơn ngay từ lần tải đầu tiên và được tối ưu tốt hơn cho các công cụ tìm kiếm.



Hình 1.5. Server Side Rendering của NextJS

- **CSR (Client-Side Rendering)** là phương pháp render nội dung trên trình duyệt bằng JavaScript sau khi toàn bộ mã được tải về. Điều này giúp trải nghiệm người dùng mượt mà khi chuyển trang, nhưng lại khiến tốc độ tải ban đầu chậm và không tối ưu cho SEO.

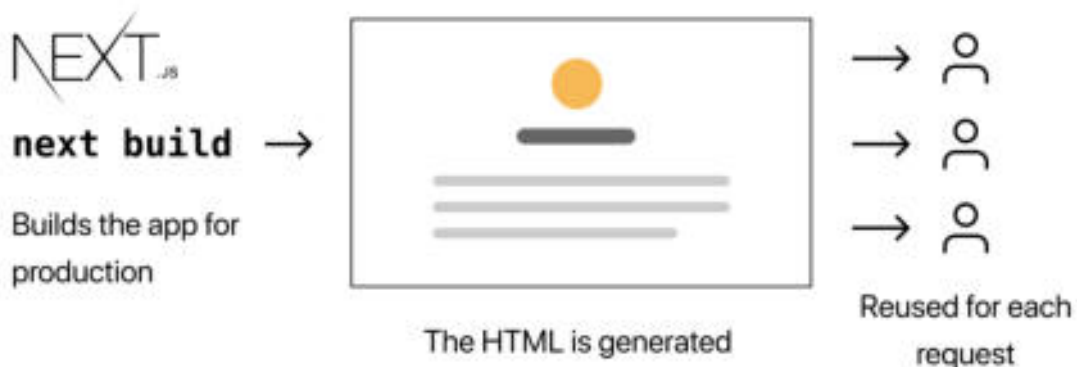


Hình 1.6. Client Side Rendering của NextJS

- **SSG (Static Site Generation)** là một phương pháp tạo các trang tĩnh và lưu chúng xuống dưới dạng file html tĩnh. Điều này giúp cải thiện hiệu suất tải trang và cung cấp trải nghiệm người dùng tốt hơn vì nội dung được lấy từ file html và hiển thị ngay lập tức mà không cần đợi việc tải về từ phía server.

Static Generation

The HTML is generated at **build-time** and is reused for each request.



Hình 1.7. Static Generation Side của NextJS

❖ **Ưu và nhược điểm của NextJS:**

- **Ưu điểm:**

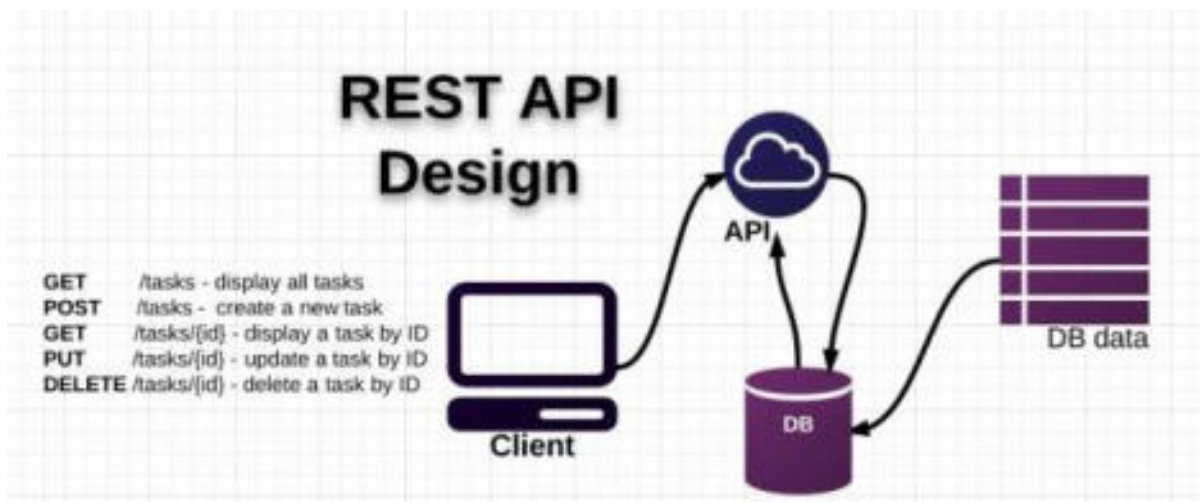
- + Sử dụng SSR và SSG: Giúp cải thiện tốc độ tải trang và khả năng SEO.
- + Có nhiều tính năng giúp tối ưu hoá hiệu suất như Code Splitting, Lazy Loading, Image Optimization,...
- + Fast Refresh: Tính năng giúp tự động làm mới giao diện mà không cần load lại toàn bộ trang.
- + Tự động tạo file CSS dành riêng cho mỗi trang, giúp tránh xung đột trong việc sử dụng và quản lý các file CSS.
- + Hỗ trợ TypeScript: NextJS cũng hỗ trợ sử dụng Typescript giúp cải thiện tính rõ ràng cho code và thuận tiện cho việc debug về sau.
- + Cộng đồng lớn: NextJS có một cộng đồng sử dụng đông đảo, điều này được chứng minh ở trên chính trang Github của NextJS khi nó đang đạt khoảng hơn 100k sao. Điều này giúp cho NextJS có thêm nhiều nguồn tài liệu phong phú và các plugin hữu ích.
- + Hệ sinh thái mạnh mẽ: NextJS kết hợp tốt với các thư viện và công cụ như Redux, React Query, Apollo Client và nhiều thư viện khác nằm trong hệ sinh thái của React.
- + Tích hợp tốt với React: Nếu bạn đã quen với việc sử dụng React trước đó thì việc làm quen với NextJS sẽ đơn giản hơn rất nhiều.

- **Nhược điểm:**

- + Khó học cho người mới: Nếu chưa có hiểu biết cơ bản về Web Fundamentals, JavaScript và React thì việc học NextJS sẽ hơi khó khăn. Nhất là khi gặp các khái niệm như SSR hay SSG.
- + Khó khăn trong việc tích hợp với một số thư viện bên ngoài: Một số thư viện và plugin có thể cần phải điều chỉnh hoặc tùy chỉnh để hoạt động tốt với Next.js. Ví dụ như để sử dụng [Redux](#) trong ứng dụng NextJS, cần cài thêm thư viện next-redux-wrapper để quản lý state trên cả server và client.

1.4. Tổng quan về RestfulAPI

RESTful API (Representational State Transfer) là một kiểu kiến trúc dùng để thiết kế các dịch vụ web, trong đó các tài nguyên được truy cập và thao tác thông qua các phương thức HTTP như GET, POST, PUT, DELETE. Mỗi tài nguyên thường được định danh bằng một URL duy nhất, và dữ liệu được trao đổi chủ yếu dưới dạng JSON. RESTful API được đánh giá cao vì tính đơn giản, dễ mở rộng và khả năng tương thích tốt với nhiều nền tảng khác nhau, nên được sử dụng phổ biến trong việc xây dựng các hệ thống backend và dịch vụ web hiện đại.



Hình 1.8. Mô tả Restful API

1.5. Tổng quan về n8n

n8n (viết tắt của "**nodemation**") là một nền tảng **tự động hóa workflow mã nguồn mở**, được thiết kế nhằm hỗ trợ người dùng tích hợp và điều phối các dịch vụ, ứng dụng khác nhau thông qua giao diện **kéo-thả trực quan (low-code)**. Đây là một giải pháp rất phổ biến trong việc xây dựng các quy trình nghiệp vụ tự động mà **không yêu cầu quá nhiều kiến thức lập trình**.

n8n cho phép tạo ra các **luồng công việc (workflow)** linh hoạt, bao gồm nhiều bước (nodes) được kết nối với nhau. Mỗi bước có thể thực hiện một hành động cụ thể như gọi API, xử lý dữ liệu, gửi email, lưu thông tin vào cơ sở dữ liệu, v.v. Một điểm mạnh của n8n là hỗ trợ **viết mã JavaScript tùy chỉnh** trong từng node, giúp người dùng kiểm soát và xử lý logic một cách chi tiết khi cần thiết.

Với kiến trúc mở và khả năng mở rộng, n8n hỗ trợ hàng trăm tích hợp sẵn với các dịch vụ phổ biến như: Google Sheets, Discord, Slack, HTTP Request, Webhook, MySQL, MongoDB, v.v. Người dùng cũng có thể dễ dàng tạo node tích hợp riêng cho các hệ thống nội bộ hoặc API tùy chỉnh.

Trong bối cảnh các hệ thống hiện đại ngày càng cần đến việc tích hợp giữa các nền tảng, **nên đóng vai trò như một công cụ trung gian mạnh mẽ** để đồng bộ dữ liệu, điều phối tác vụ, và cải thiện hiệu suất làm việc. Việc sử dụng n8n trong các đồ án hệ thống thông tin, đặc biệt là các hệ thống có yêu cầu tích hợp chatbot, gửi thông báo, đặt lịch, hay quản lý dữ liệu giữa nhiều dịch vụ, sẽ mang lại tính hiệu quả và khả năng mở rộng cao.

1.6. Tổng quan về RAG

RAG (viết tắt của Retrieval-Augmented Generation) là một kỹ thuật tiên tiến trong lĩnh vực Xử lý ngôn ngữ tự nhiên (NLP), kết hợp giữa truy xuất thông tin (retrieval) và mô hình sinh văn bản (generation). Thay vì chỉ dựa vào những gì mô hình ngôn ngữ như GPT, Gemini, hay BERT đã được huấn luyện từ trước, RAG cho phép mô hình truy vấn dữ liệu từ các nguồn tài liệu bên ngoài như cơ sở dữ liệu, file văn bản, hoặc kho tri thức chuyên biệt để tạo ra câu trả lời chính xác, cập nhật và phù hợp hơn với ngữ cảnh. Mục tiêu của RAG là nâng cao chất lượng câu trả lời

Quy trình hoạt động tổng quát của RAG:

1. Tạo cơ sở dữ liệu vector (Vector Database):

Dữ liệu tri thức (tài liệu, hướng dẫn, câu hỏi thường gặp,...) được chuyển đổi thành các vector nhúng ngữ nghĩa thông qua mô hình embedding (ví dụ: Sentence-BERT, OpenAI Embedding). Các vector này sau đó được lưu trữ vào một cơ sở dữ liệu vector như Qdrant, Pinecone, hoặc FAISS để phục vụ cho truy vấn ngữ nghĩa.

2. Tiếp nhận truy vấn từ người dùng (User Input):

Người dùng nhập một câu truy vấn bằng ngôn ngữ tự nhiên nhằm tìm kiếm thông tin hoặc yêu cầu sinh câu trả lời.

3. Truy xuất thông tin liên quan (Information Retrieval):

Truy vấn được chuyển thành vector và dùng để quét toàn bộ cơ sở dữ liệu vector, từ đó truy xuất các đoạn văn bản có ngữ nghĩa gần nhất. Các đoạn văn bản này sẽ đóng vai trò là ngữ cảnh đầu vào cho mô hình ngôn ngữ lớn (LLM).

4. Kết hợp truy vấn và dữ liệu truy xuất (Combining Data):

Các đoạn văn bản được truy xuất sẽ được kết hợp với câu truy vấn ban đầu để tạo thành một prompt đầy đủ, mang thông tin nền cần thiết.

5. Sinh phản hồi cuối cùng (Generate Text):

Prompt đã được bổ sung ngữ cảnh được đưa vào mô hình sinh ngôn ngữ như GPT hoặc Gemini để sinh ra câu trả lời tự nhiên, có căn cứ và phù hợp với ngữ cảnh.



Hình 1.9. Quy trình hoạt động RAG

1.7. Công nghệ sử dụng

- Frontend:

- **Ngôn ngữ & Framework:** ReactJS, NextJS, TypeScript
- **Thư viện & Công cụ UI:** Tailwind CSS, Ant Design
- **Giao tiếp API:** Axios
- **Deploy:** Netlify

- Backend:

- **Framework:** NestJS (sử dụng TypeScript)
- **Deploy:** Render
- **Lưu trữ media:** Cloudinary
- **Tài liệu API:** Swagger
- **Cơ sở dữ liệu:** PostgreSQL

- Công cụ phát triển:

- VS Code
- GitHub (quản lý mã nguồn)
- Jira (quản lý công việc, task)

- Tích hợp AI & Tự động hóa:

- **n8n:** Workflow automation
- **RAG (Retrieval-Augmented Generation):** Tăng cường khả năng truy xuất dữ liệu cho các mô hình AI

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1 Tác nhân

Bảng biểu dưới đây thể hiện toàn bộ các tác nhân tương tác với hệ thống quản lý và kinh doanh ô tô, xe máy điện. Mỗi tác nhân đóng một vai trò quan trọng và đảm nhiệm những chức năng riêng biệt nhằm đảm bảo hệ thống vận hành hiệu quả và linh hoạt.

Hiện tại, hệ thống bao gồm **năm tác nhân chính**:

- **Quản trị viên (Admin)**
- **Nhân viên bán hàng**
- **Nhân viên quản lý kho**
- **Nhân viên giao hàng**
- **Khách hàng**

Mỗi tác nhân sẽ được phân quyền cụ thể, tương ứng với các chức năng và thao tác mà họ được phép thực hiện trên hệ thống. Việc phân chia rõ ràng vai trò và quyền hạn giúp tăng cường bảo mật, đồng thời tối ưu hóa quy trình quản lý và phục vụ khách hàng.

Bảng 2.1. Các tác nhân của hệ thống

Tác nhân	Mô tả
Admin	<ul style="list-style-type: none">- Toàn quyền quản lý với:<ul style="list-style-type: none">• Sản phẩm:<ul style="list-style-type: none">○ Ô tô○ Xe máy điện• Biến thể sản phẩm• Thống kê• Danh mục sản phẩm• Danh mục tin tức• Tin tức• Đơn hàng• Nhãn hàng cung cấp• Nhập kho• Xuất kho• Voucher• Kho• Thông tin tài khoản cá nhân• Người dùng:<ul style="list-style-type: none">○ Khách hàng○ Nhân viên

	<ul style="list-style-type: none">• Cấu hình hệ thống:<ul style="list-style-type: none">○ Quản lý role○ Quản lý phân quyền
Nhân viên bán hàng	<ul style="list-style-type: none">- Quản lý đơn hàng- Quản lý khách hàng- Quản lý thông tin tài khoản cá nhân
Nhân viên quản lý kho	<ul style="list-style-type: none">- Quản lý đơn hàng- Quản lý kho- Quản lý thông tin tài khoản cá nhân
Nhân viên giao hàng	<ul style="list-style-type: none">- Quản lý đơn hàng- Quản lý thông tin tài khoản cá nhân
Khách hàng	<ul style="list-style-type: none">- Quản lý thông tin tài khoản cá nhân- Tìm kiếm và xem các tin tức liên quan- Quản lý giỏ hàng- Quản lý địa chỉ nhận hàng- Xem lịch sử và theo dõi trạng thái đơn hàng- Tìm kiếm và tra cứu thông tin sản phẩm- Thanh toán thực tế- Chat với chatbot AI.

2.1. Phân tích chức năng

2.1.1. Admin

2.1.1.1. Thống kê

- Thống kê doanh thu và lợi nhuận theo các tháng trong năm, các ngày trong tháng
- Thống kê tổng khách hàng, tổng hóa đơn, tổng sản phẩm, tổng doanh thu theo năm, tổng nhân hàng cung cấp
- Thống kê tổng số lượng các trạng thái của hóa đơn theo các năm

2.1.1.2. Quản lý người dùng

- Hiện thị danh sách các user với các thông tin liên quan
- Xem chi tiết thông tin của user
- Tìm kiếm user theo ID, username, gmail, số điện thoại, ...
- Lọc user theo vai trò, trạng thái hoạt động, khoảng thời gian user tạo tài khoản

2.1.1.3. Quản lý sản phẩm

- Hiện thị danh sách các sản phẩm với thông tin cần thiết
- Thêm mới, sửa lại thông tin hoặc xóa, xem thông tin chi tiết của sản phẩm
- Hỗ trợ tìm kiếm sản phẩm theo ID, tên, lọc theo thể loại, danh mục, nhãn hàng, trạng thái hoạt động của sản phẩm

2.1.1.4. Quản lý biến thể của sản phẩm

- Hiện thị danh sách các biến thể của sản phẩm
- Thêm mới, xóa, cập nhật thông tin biến thể
- Tìm kiếm biến thể theo ID, tên biến thể, lọc theo sản phẩm, kho hàng, nhà cung cấp

2.1.1.5. Quản lý danh mục sản phẩm

- Hiện thị danh sách các danh mục sản phẩm theo cấu trúc cây hoặc danh sách phẳng
- Thêm mới, sửa, xóa danh mục
- Tìm kiếm danh mục theo ID, tên danh mục, slug, mô tả
- Lọc ra các danh mục theo thể loại, kiểu hiển thị, khoảng thời gian khởi tạo danh mục

2.1.1.6. Quản lý danh mục tin tức

- Hiện thị danh sách danh mục tin tức với các thông tin cần thiết
- Thêm, xóa, sửa danh mục tin tức
- Tìm kiếm theo tên, nội dung, tiêu đề, khoảng thời gian khởi tạo danh mục tin tức
- Quản lý các tin tức có trong danh mục, có thể xem chi tiết, thêm, sửa, xóa tin tức, tìm kiếm tin tức theo nội dung, tiêu đề

2.1.1.7. Quản lý đơn hàng

- Hiện thị danh sách các đơn hàng với các thông tin cần thiết
- Xem chi tiết đơn hàng, in thông tin đơn hàng ra file pdf để có thể giao đơn cho khách hàng nếu cần hoặc giao cho nhân viên quản lý kho để tiến hành xuất hàng ra kho.
- Cập nhật trạng thái đơn hàng bao gồm các trạng thái: đang xử lý, chờ xác nhận, đã xuất kho, đang vận chuyển, đang giao hàng, đã giao hàng, giao hàng thất bại, hủy đơn hàng
- Tìm kiếm đơn hàng theo mã đơn hàng, tên khách hàng, mã khách hàng, email, ...
- Lọc ra các đơn hàng theo các trạng thái của đơn hàng, thanh toán, các phương thức thanh toán, phương thức vận chuyển

2.1.1.8. Quản lý kho hàng

- Hiện thị danh sách kho hàng có sẵn

- Thêm, xóa, sửa thông tin kho, đặt kho làm mặc định

2.1.1.9. Quản lý nhãn hàng cung cấp

- Hiện thị các nhãn hàng với các thông tin liên quan
- Thêm mới, chỉnh sửa, xóa nhãn hàng
- Tìm kiếm nhãn hàng theo ID, tên, khoảng thời gian thêm mới

2.1.1.10. Quản lý nhập kho

- Hiện thị danh sách nhập kho với các thông tin liên quan
- Tạo phiếu nhập kho, nhập kho từ các biến thể sản phẩm đã có sẵn trong kho hoặc các sản phẩm hoàn toàn mới, chưa có trong hệ thống, biến thể có thể được nhập tại nhiều kho khác nhau, tại nhiều lô khác nhau, mỗi lô sẽ có giá nhập, số lượng nhập khác nhau
- Tìm kiếm phiếu nhập theo ID phiếu, ghi chú, lọc theo nhà kho hàng, sản phẩm, biến thể và khoảng thời gian tạo phiếu nhập

2.1.1.11. Quản lý xuất kho

- Hiện thị danh sách xuất kho với các thông tin liên quan
- Tạo phiếu xuất kho từ mã đơn hàng, có thể xuất kho nhiều đơn cùng lúc, có thể chọn lô để xuất, và 1 biến thể có thể được xuất tại nhiều lô hàng khác nhau
- Tìm kiếm phiếu nhập theo ID phiếu, ghi chú, lọc theo nhà kho hàng, sản phẩm, biến thể và khoảng thời gian tạo phiếu nhập

2.1.1.12. Quản lý phân quyền

- Hiện thị danh sách các role có trong hệ thống
- Cấp quyền, sửa hoặc xóa quyền trong role

2.1.1.13. Quản lý voucher

- Hiện thị danh sách voucher với các thông tin cần thiết
- Thêm, xóa sửa voucher
- Tìm kiếm, lọc voucher theo các trạng thái và hiệu lực dùng
- Tặng voucher cho khách hàng lâu năm

2.1.2. Nhân viên bán hàng

- Tìm kiếm, lọc, tra cứu thông tin về các đối tượng có trong hệ thống
- Tra thông tin và cập nhật trạng thái đơn hàng, xác nhận đơn hàng, in hóa đơn cho nhân viên quản lý kho và cho khách hàng nếu cần
- Tạo tài khoản, cập nhật thông tin tài khoản cho khách hàng

- Thống kê

2.1.3. Nhân viên quản lý kho

- Quản lý xuất, nhập kho
- Quản lý các kho khả dụng
- Tra thông tin và cập nhật đơn hàng lên trạng thái đã xuất kho, đã bàn giao cho đơn vị vận chuyển

2.1.4. Nhân viên giao hàng

- Tra thông tin và cập nhật đơn hàng lên các trạng thái đang vận chuyển, đang giao hàng, giao hàng thất bại, giao hàng thành công

2.1.5. Khách hàng

2.1.5.1. Đăng ký

- Người dùng cung cấp thông tin: Họ tên, email, mật khẩu.
- Kiểm tra email hợp lệ, đảm bảo email đó chưa đăng ký tài khoản trước đây.
- Lưu thông tin tài khoản vào cơ sở dữ liệu nếu hợp lệ, hiển thị thông báo thành công hoặc lỗi.
- Sau khi đăng ký thành công, người dùng có thể đăng nhập ngay.

2.1.5.2. Đăng nhập

- Người dùng nhập email và mật khẩu để truy cập hệ thống
- Xác thực thông tin; nếu đúng, chuyển đến trang chính, nếu sai hiển thị lỗi.

2.1.5.3. Tìm kiếm và lọc sản phẩm

- Người dùng nhập tên sản phẩm vào thanh tìm kiếm.
- Hệ thống hiển thị danh sách kết quả phù hợp, hỗ trợ lọc theo giá và loại sản phẩm, nhãn hàng cung cấp,..

2.1.5.4. Xem chi tiết sản phẩm

- Người dùng nhấn vào sản phẩm để xem thông tin chi tiết: tên, hình ảnh, giá, mô tả, tồn kho, đã bán, hướng dẫn sử dụng.

2.1.5.5. Quản lý giỏ hàng

- Người dùng thêm, xóa sản phẩm trong giỏ hàng từ trang chi tiết hoặc danh mục sản phẩm.
- Cập nhật số lượng sản phẩm trong giỏ hoặc xóa sản phẩm khỏi giỏ hàng.
- Hiển thị tổng tiền tạm tính khi có thay đổi trong giỏ hàng.

2.1.5.6. Đặt hàng

- Người dùng chọn sản phẩm từ giỏ hàng hoặc từ trang chi tiết để đặt hàng: điền thông tin giao hàng, chọn phương thức thanh toán.

- Nếu chọn thanh toán online thì chuyển tới trang nhận mã QR thanh toán; thanh toán thành công thì hệ thống trả về trang chi tiết đơn hàng đã đặt.
- Chọn thanh toán khi nhận hàng thì hệ thống trả về trang hiển thị thông tin chi tiết đơn đã đặt.

2.1.5.7. Xem lịch sử đơn hàng

- Hiển thị danh sách đơn hàng đã đặt kèm trạng thái
- Cho phép người dùng lọc lịch sử đơn hàng theo trạng thái.

2.1.5.8. Quản lý thông tin cá nhân

- Người dùng xem và chỉnh sửa thông tin như họ tên, số điện thoại, địa chỉ giao hàng, mật khẩu tài khoản

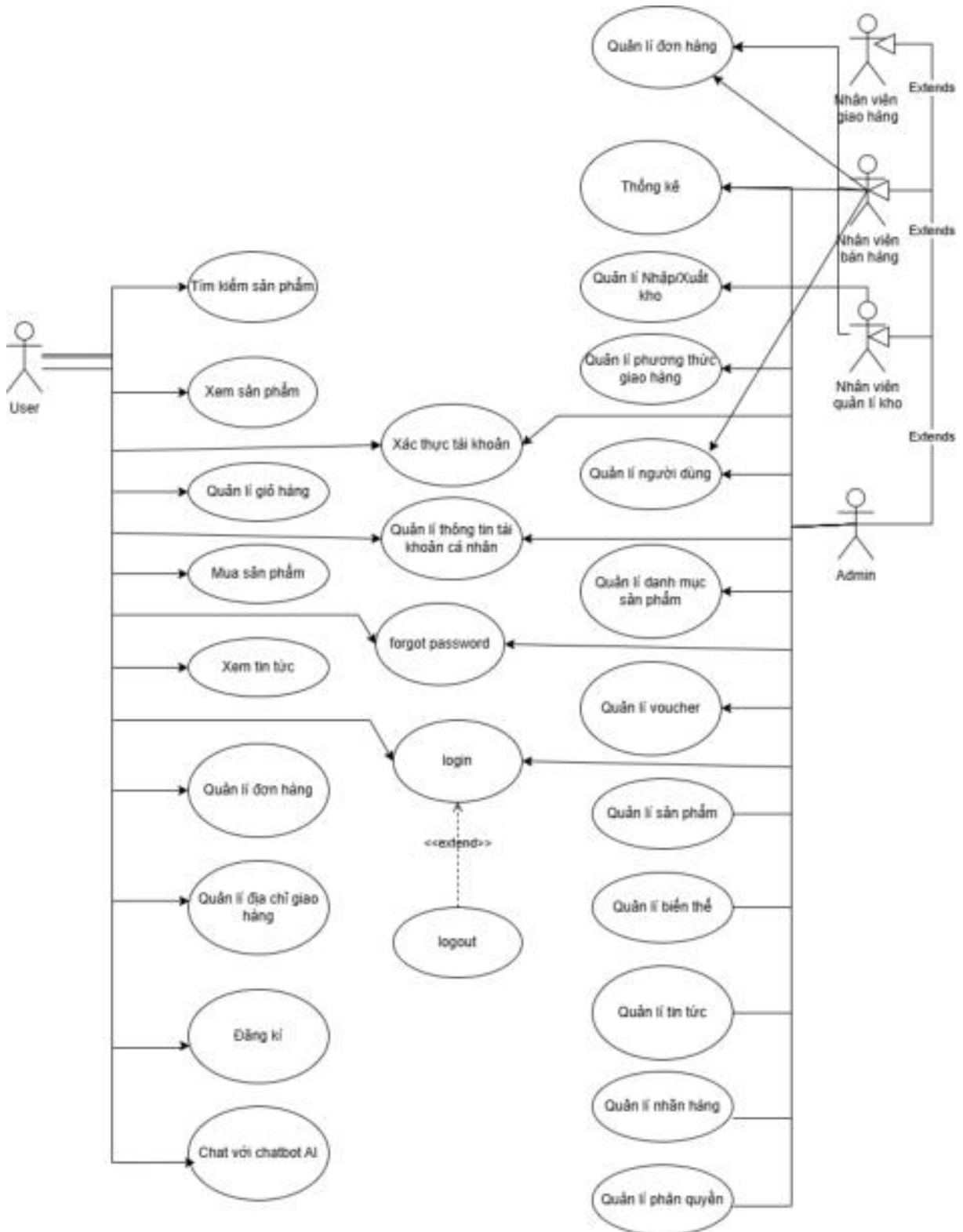
2.1.6. Chat với ChatbotAI

Chức năng này cho phép người dùng tương tác trực tiếp với Chatbot AI trên nền tảng website nhằm hỗ trợ các hoạt động tư vấn, tra cứu và đặt lịch một cách tự động, bao gồm:

- Hỏi đáp thông tin sản phẩm: Người dùng có thể đặt câu hỏi liên quan đến các dòng xe điện, thông số kỹ thuật, tính năng, giá cả,...
- Tư vấn doanh nghiệp: Chatbot có khả năng cung cấp thông tin tổng quan về doanh nghiệp như địa chỉ, giờ mở cửa, chính sách bảo hành, dịch vụ đi kèm,
- Đặt lịch hẹn: Cho phép người dùng lên lịch bảo dưỡng, sửa chữa, hoặc thử xe điện trực tiếp qua chatbot.
- Báo giá và đặt hàng: Người dùng có thể yêu cầu báo giá sản phẩm hoặc tiến hành đặt hàng thông qua các bước hướng dẫn của chatbot.

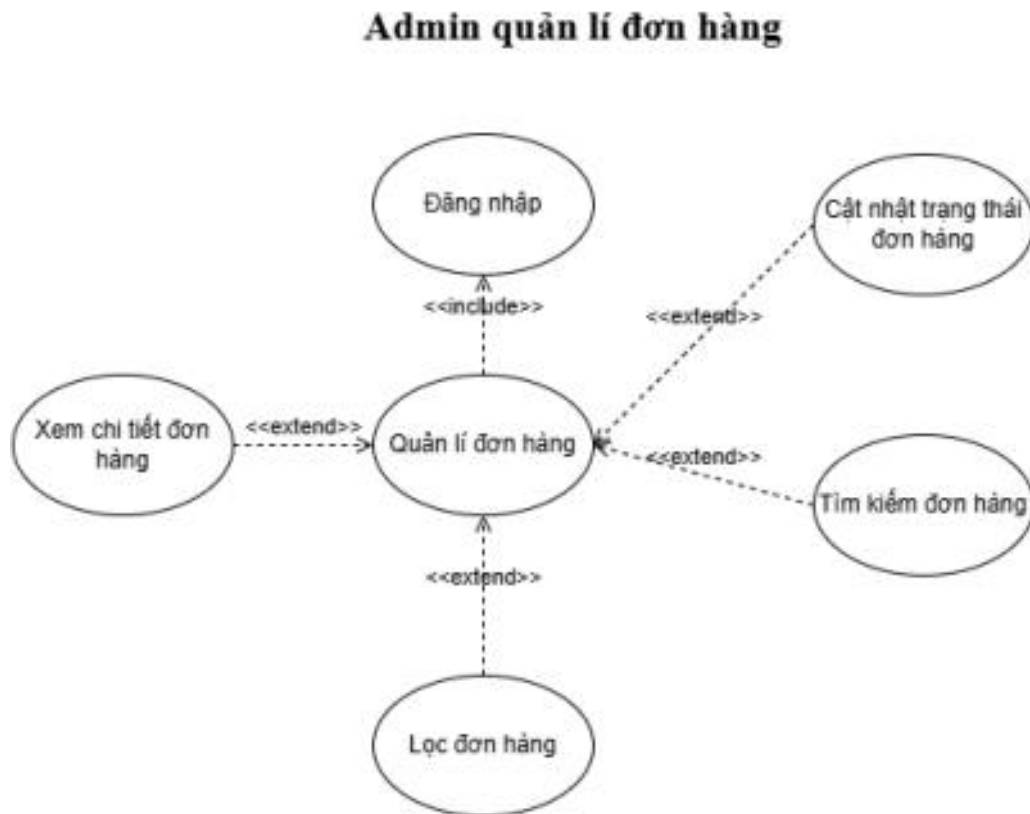
2.2. Sơ đồ usecase

Sơ đồ này cung cấp một mô hình tổng thể về các nhóm chức năng tương ứng với các tác nhân có trong hệ thống. Các chức năng được chia làm các Usecase sau đây



Hình 2.1. Sơ đồ usecase tổng quát

2.2.1. Admin quản lý đơn hàng



Hình 2.2. Usecase admin quản lý đơn hàng

Bảng 2.2. Đặc tả usecase admin quản lý đơn hàng

Tên Usecase	Admin quản lý đơn hàng
Tác nhân chính	Admin
Mô tả Usecase	Cho phép admin có thể xem chi tiết, cập nhật, lọc và tìm kiếm các đơn hàng
Tiền điều kiện	Đăng nhập vào hệ thống với quyền admin
Hậu điều kiện	Admin cập nhật, lọc và tìm kiếm các đơn hàng thành công Hệ thống cập nhật lại bảng dữ liệu
Kích hoạt	Khi admin chọn quản lý đơn hàng

Chuỗi sự kiện:

Giao diện bao gồm các chức năng : xem chi tiết, cập nhật, lọc và tìm kiếm đơn hàng

+ Cập nhật trạng thái đơn hàng:

- ❖ Hệ thống hiển thị giao diện quản lý đơn hàng
- ❖ Admin bấm xem chi tiết đơn hàng của record bất kì
- ❖ Admin chọn cập nhật các trạng thái đơn hàng khác nhau
- ❖ Hệ thống hiển thị thông báo trạng thái đơn hàng sau khi cập nhật, nếu thành công dữ liệu bảng sẽ được cập nhật lại, ngược lại hiển thị thông báo lỗi

+ Tìm kiếm, lọc thông tin đơn hàng:

- ❖ Admin gõ vào tên người dùng, mã order và lọc ra các order với trạng thái đơn hàng, trạng thái thanh toán, phương thức thanh toán, phương thức giao hàng và khoảng thời gian đơn hàng được tạo
- ❖ Hệ thống sẽ thực hiện tìm kiếm và cập nhật lại bảng với dữ liệu tìm kiếm được

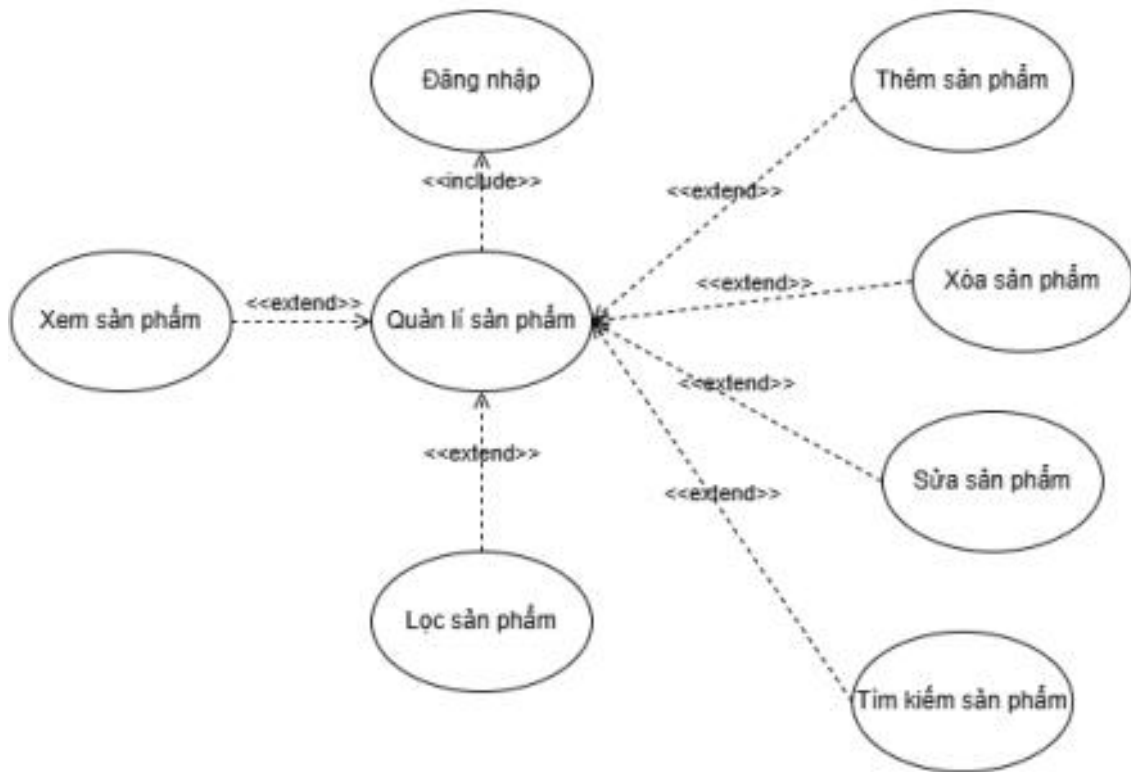
+ Xem chi tiết đơn hàng:

- ❖ Admin click xem chi tiết đơn hàng của record bất kì
- ❖ Hệ thống hiển thị giao diện bao gồm các thông tin có trong đơn hàng

Ngoại lệ

2.2.2. Admin quản lý sản phẩm

Admin quản lý sản phẩm



Hình 2.3. Usecase admin quản lý sản phẩm

Bảng 2.3. Đặc tả usecase admin quản lý sản phẩm

Tên Usecase	Admin quản lý sản phẩm
Tác nhân chính	Admin
Mô tả Usecase	Cho phép admin có thể thêm, sửa, xóa, tìm kiếm, lọc, xem chi tiết sản phẩm
Tiền điều kiện	Đăng nhập vào hệ thống với quyền admin
Hậu điều kiện	Admin thêm, xóa, sửa, tìm kiếm thành công Hệ thống cập nhật lại bảng dữ liệu
Kích hoạt	Khi admin chọn quản lý sản phẩm

Chuỗi sự kiện:

Giao diện bao gồm các chức năng : Thêm, xóa, sửa, tìm kiếm, lọc sản phẩm

+ Thêm sản phẩm:

- ❖ Hệ thống hiển thị giao diện quản lý sản phẩm
- ❖ Admin click vào nút thêm mới sản phẩm
- ❖ Hệ thống hiển thị giao diện thêm mới sản phẩm
- ❖ Admin nhập vào dữ liệu tương ứng và nhấn nút lưu
- ❖ Hệ thống xác thực thông tin đầu vào
- ❖ Nếu hợp lệ hệ thống sẽ hiển thị thông báo thêm sản phẩm thành công nếu không hợp lệ sẽ hiển thị thông báo lỗi
- ❖ Hệ thống quay lại giao diện quản lý sản phẩm và cập nhật lại bảng dữ liệu
- ❖ Trong trường hợp admin nhấn nút cancel
- ❖ Hệ thống sẽ hủy bỏ việc thêm mới và quay lại trang quản lý sản phẩm

+ Sửa sản phẩm:

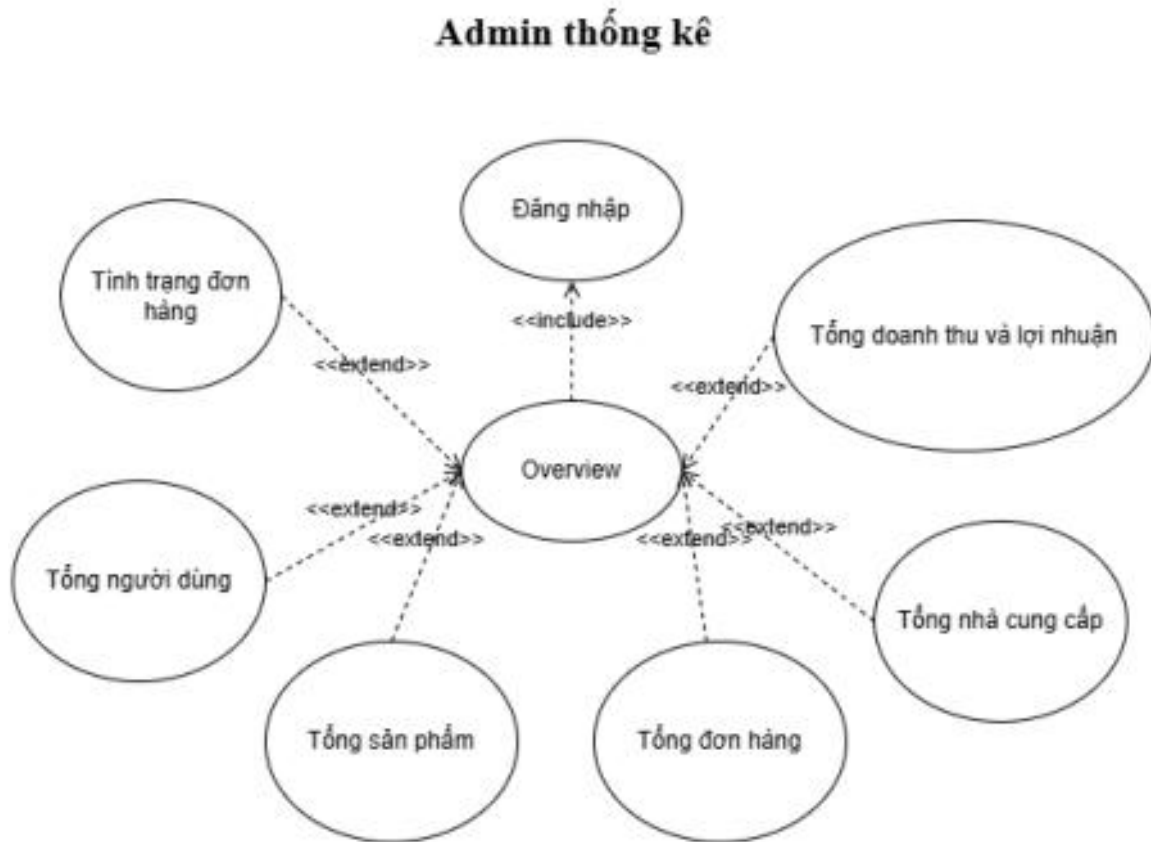
- ❖ Admin click vào nút action của một record bất kì trong bảng
- ❖ Hệ thống hiển thị dropdown gồm chức năng xem, sửa, xóa
- ❖ Admin click sửa
- ❖ Hệ thống hiển thị giao diện cập nhật sản phẩm
- ❖ Admin nhập vào dữ liệu tương ứng và nhấn nút lưu
- ❖ Hệ thống xác thực thông tin đầu vào
- ❖ Nếu hợp lệ hệ thống sẽ hiển thị thông báo cập nhật sản phẩm thành công nếu không hợp lệ sẽ hiển thị thông báo lỗi
- ❖ Hệ thống quay lại giao diện quản lý sản phẩm và cập nhật lại bảng dữ liệu
- ❖ Trong trường hợp admin nhấn nút cancel
- ❖ Hệ thống sẽ hủy bỏ việc sửa mới và quay lại trang quản lý sản phẩm

+ Xóa sản phẩm:

- ❖ Admin click vào nút action của một record bất kì trong bảng
- ❖ Hệ thống hiển thị dropdown gồm chức năng xem, sửa, xóa
- ❖ Admin click xóa
- ❖ Hệ thống hiển thị thông báo bạn có chắc chắn muốn xóa record này
- ❖ Admin click Có
- ❖ Hệ thống hiển thị thông báo xóa thành công và cập nhật lại bảng
- ❖ Trường hợp admin click cancel

<ul style="list-style-type: none">❖ Hệ thống hủy bỏ việc xóa và tắt thông báo+ Xem sản phẩm:❖ Admin click xem trong action của record bất kì❖ Hệ thống hiển thị giao diện gồm các thông tin chi tiết của sản phẩm+ Tìm kiếm, lọc sản phẩm:❖ Admin gõ vào dữ liệu cần tìm vào ô search và đồng thời chọn thêm các lựa chọn lọc khác❖ Hệ thống sẽ thực hiện tìm kiếm và cập nhật lại bảng với dữ liệu tìm kiếm được	
Ngoại lệ	Hệ thống hủy việc cập nhật khi admin nhấn nút điều hướng trên sidebar

2.2.3. Admin thống kê



Hình 2.4. Usecase admin thống kê

Bảng 2.4. Đặc tả usecase admin thống kê

Tên Usecase	Admin Thống kê
Tác nhân chính	Admin
Mô tả Usecase	Cho phép Admin thống kê tổng doanh thu và lợi nhuận theo ngày tháng năm, tình trạng các đơn hàng, thống kê được tổng người dùng, sản phẩm, đơn hàng và nhà cung cấp trong từng năm
Tiền điều kiện	Đăng nhập vào hệ thống với quyền admin
Hậu điều kiện	Admin lựa chọn xem các thống kê thành công Biểu đồ hiển thị các dữ liệu khác nhau hoặc hệ thống sẽ chuyển hướng sang các trang khác để xem chi tiết
Kích hoạt	Khi admin chọn tổng quan

Chuỗi sự kiện:

Giao diện bao gồm các chức năng : Thống kê theo năm và theo khoảng ngày trong tháng

+ Thống kê theo năm:

- ❖ Hệ thống hiển thị giao diện overview
- ❖ Admin chọn năm cần thống kê
- ❖ Các thông số thống kê sẽ thay đổi và hiển thị dữ liệu trong 12 tháng của năm tương ứng

+ Thống kê theo ngày trong tháng:

- ❖ Admin chọn vào khoảng ngày cần thống kê trong biểu đồ
- ❖ Các thông số thống kê của biểu đồ sẽ thay đổi và hiển thị dữ liệu của khoảng ngày trong tháng của năm (mặc định là năm hiện tại hoặc là năm do admin đã chọn trước đó)

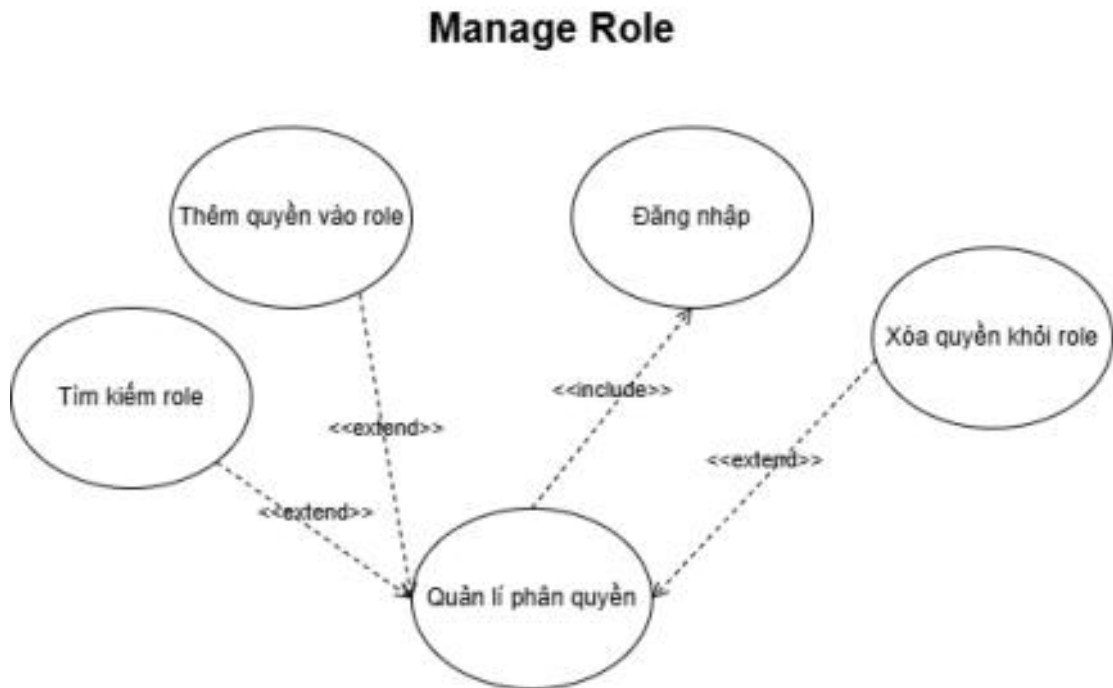
+ Xem thông tin chi tiết:

- ❖ Admin click vào các xem thông tin chi tiết của khối dữ liệu thống kê bất kì
- ❖ Hệ thống sẽ chuyển admin đến trang thông tin chi tiết của dữ liệu đó

Ngoại lệ

Hệ thống hủy việc thống kê khi admin nhấn nút điều hướng trên thanh navigation

2.2.4. Admin quản lý phân quyền



Hình 2.5. Usecase admin quản lý role

Bảng 2.5. Đặc tả usecase admin quản lý phân quyền

Tên Usecase	Admin quản lý phân quyền
Tác nhân chính	Admin
Mô tả Usecase	Cho phép admin có thể tìm kiếm, thêm xóa quyền của bất kì role nào
Tiền điều kiện	Đăng nhập vào hệ thống với quyền admin
Hậu điều kiện	Admin tìm kiếm, thêm, xóa quyền trong role thành công Hệ thống cập nhật lại bảng dữ liệu
Kích hoạt	Khi admin chọn quản lý phân quyền

Chuỗi sự kiện:

Giao diện bao gồm các chức năng : Tìm kiếm, thêm, xóa quyền trong role

+ Thêm hoặc xóa quyền trong role:

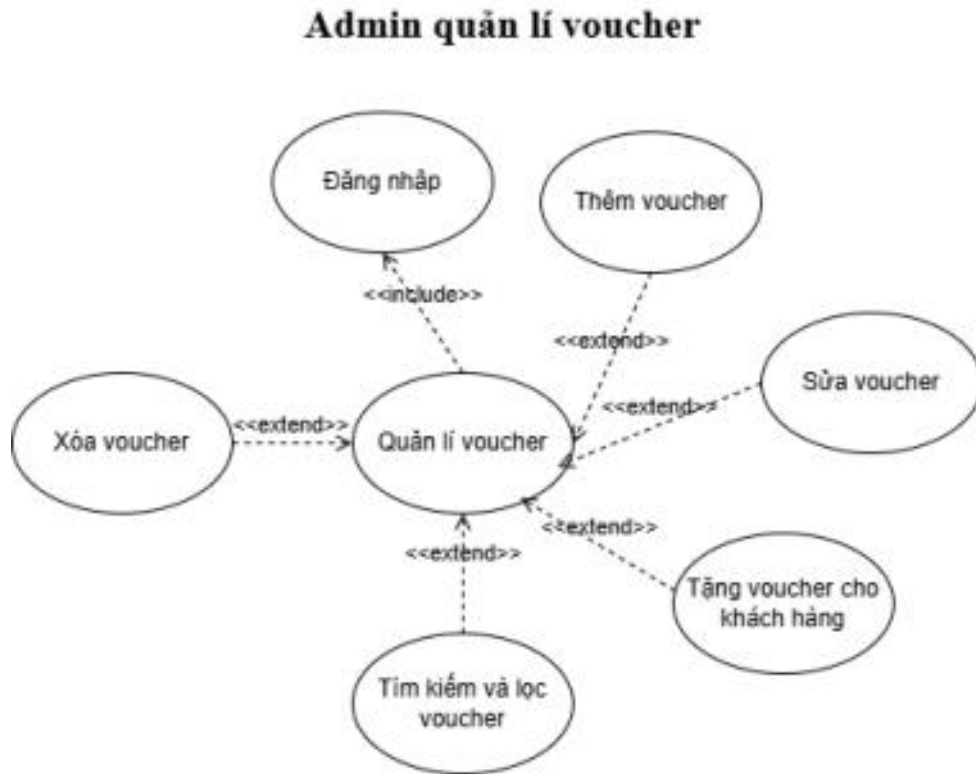
- ❖ Hệ thống hiển thị giao diện quản lý role
- ❖ Admin chọn vào record của role bất kì
- ❖ Nhấn thêm hoặc xóa quyền
- ❖ Hệ thống hiển thị thông báo để xác nhận thêm hoặc xóa quyền
- ❖ Admin nhấn vào xác nhận
- ❖ Hệ thống sẽ hiển thị thông báo thành công và quay trở lại trang quản lý role
- ❖ Trong trường hợp Admin chọn cancel
- ❖ Hệ thống sẽ hủy việc thêm hoặc xóa quyền và quay lại trang quản lý role

+ Tìm kiếm, role:

- ❖ Admin gõ vào tên role cần tìm vào ô search
- ❖ Hệ thống sẽ thực hiện tìm kiếm và cập nhật lại bảng với dữ liệu tìm kiếm được

Ngoại lệ

2.2.5. Admin quản lý voucher



Hình 2.6. Usecase admin quản lý voucher

Bảng 2.6. Đặc tả usecase admin quản lý voucher

Tên Usecase	Admin quản lý voucher
Tác nhân chính	Admin
Mô tả Usecase	Cho phép admin có thể tìm kiếm, thêm, sửa, xóa, tặng voucher
Tiền điều kiện	Đăng nhập vào hệ thống với quyền admin
Hậu điều kiện	Admin tìm kiếm, thêm, sửa xóa, tặng voucher thành công Hệ thống cập nhật lại bảng dữ liệu
Kích hoạt	Khi admin chọn quản lý voucher

Chuỗi sự kiện:

Giao diện bao gồm các chức năng : tìm kiếm, thêm, sửa, xóa voucher

+ Thêm voucher:

- ❖ Hệ thống hiển thị giao diện quản lý voucher
- ❖ Admin click vào nút thêm mới voucher
- ❖ Hệ thống hiển thị giao diện thêm mới voucher
- ❖ Admin nhập vào dữ liệu tương ứng và nhấn nút lưu
- ❖ Hệ thống xác thực thông tin đầu vào
- ❖ Nếu hợp lệ hệ thống sẽ hiển thị thông báo thêm sản phẩm thành công nếu không hợp lệ sẽ hiển thị thông báo lỗi
- ❖ Hệ thống quay lại giao diện quản lý voucher và cập nhật lại bảng dữ liệu
- ❖ Trong trường hợp admin nhấn nút cancel
- ❖ Hệ thống sẽ hủy bỏ việc thêm mới và quay lại trang quản lý voucher

+ Sửa voucher:

- ❖ Admin click vào nút sửa action của một record bất kì trong bảng
- ❖ Hệ thống hiển thị giao diện cập nhật voucher
- ❖ Admin nhập vào dữ liệu tương ứng và nhấn nút lưu
- ❖ Hệ thống xác thực thông tin đầu vào
- ❖ Nếu hợp lệ hệ thống sẽ hiển thị thông báo cập nhật sản phẩm thành công nếu không hợp lệ sẽ hiển thị thông báo lỗi
- ❖ Hệ thống quay lại giao diện quản lý voucher và cập nhật lại bảng dữ liệu
- ❖ Trong trường hợp admin nhấn nút cancel
- ❖ Hệ thống sẽ hủy bỏ việc sửa mới và quay lại trang quản lý voucher

+ Xóa voucher:

- ❖ Admin click vào nút action của một record bất kì trong bảng
- ❖ Hệ thống hiển thị dropdown gồm chức năng xem, sửa, xóa
- ❖ Admin click xóa
- ❖ Hệ thống hiển thị thông báo bạn có chắc chắn muốn xóa record này
- ❖ Admin click Có
- ❖ Hệ thống hiển thị thông báo xóa thành công và cập nhật lại bảng
- ❖ Trường hợp admin click cancel
- ❖ Hệ thống hủy bỏ việc xóa và tắt thông báo

+ Tìm kiếm và lọc voucher:

- ❖ Admin gõ vào tên, id của voucher cần tìm
- ❖ Hệ thống sẽ thực hiện tìm kiếm và cập nhật lại bảng với dữ liệu tìm kiếm được.

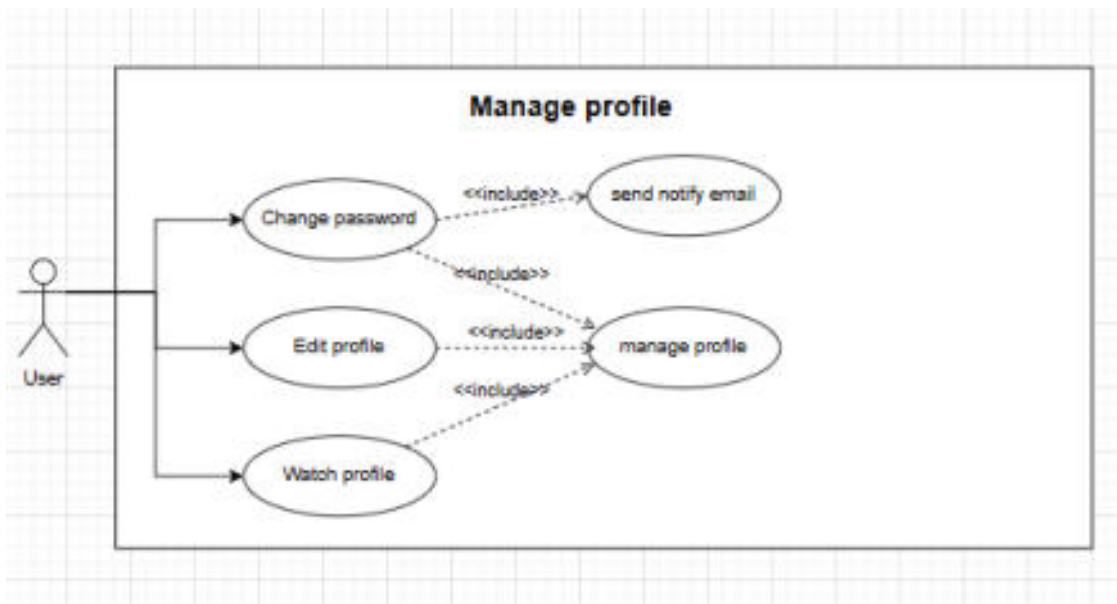
+ Tặng voucher cho khách hàng

- ❖ Admin nhấn nút tặng của record voucher bất kì
- ❖ Hệ thống hiển thị danh sách khách hàng có khả năng được tặng
- ❖ Admin chọn danh sách khách hàng và nhấn lưu
- ❖ Hệ thống quay trở lại giao diện quản lý voucher và cập nhật lại dữ liệu

Ngoại lệ

Hệ thống hủy việc cập nhật khi admin nhấn nút điều hướng trên sidebar ng hủy việc cập nhật khi admin nhấn nút điều hướng trên sidebar

2.2.6. User quản lý thông tin cá nhân

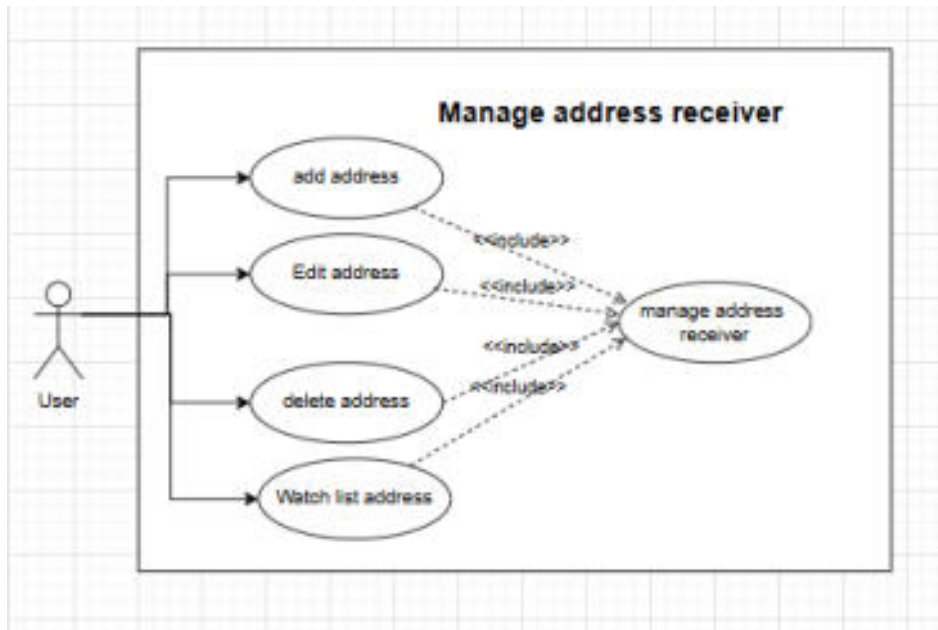


Hình 2.7. Usecase user quản lý profile

Bảng 2.7. Đặc tả usecase user quản lý profile

Tên Usecase	Quản lý profile
Tác nhân chính	Người dùng (Khách hàng)
Mô tả Usecase	Người dùng thực hiện các thao tác quản lý profile bao gồm xem thông tin profile, chỉnh sửa thông tin, đổi mật khẩu và các thao tác khác liên quan đến profile.
Tiền điều kiện	<ol style="list-style-type: none">1. Người dùng đã đăng nhập vào hệ thống.2. Profile người dùng đã được tạo trong cơ sở dữ liệu.
Hậu điều kiện	<ol style="list-style-type: none">1. Thông tin profile được cập nhật thành công (nếu chỉnh sửa).2. Mật khẩu được thay đổi và xác nhận thành công (nếu thay đổi).
Kích hoạt	<ol style="list-style-type: none">1. Người dùng nhấn vào "Quản lý profile" trong giao diện tài khoản.2. Thực hiện một trong các hành động sau:<ul style="list-style-type: none">○ Xem thông tin profile: Hệ thống hiển thị thông tin hiện tại của người dùng.○ Chỉnh sửa thông tin: Nhập thông tin mới và nhấn "Lưu".○ Đổi mật khẩu: Nhập mật khẩu cũ, mật khẩu mới và nhấn "Xác nhận".3. Hệ thống xác nhận và thực hiện yêu cầu.

2.2.7. User quản lí địa chỉ đặt hàng



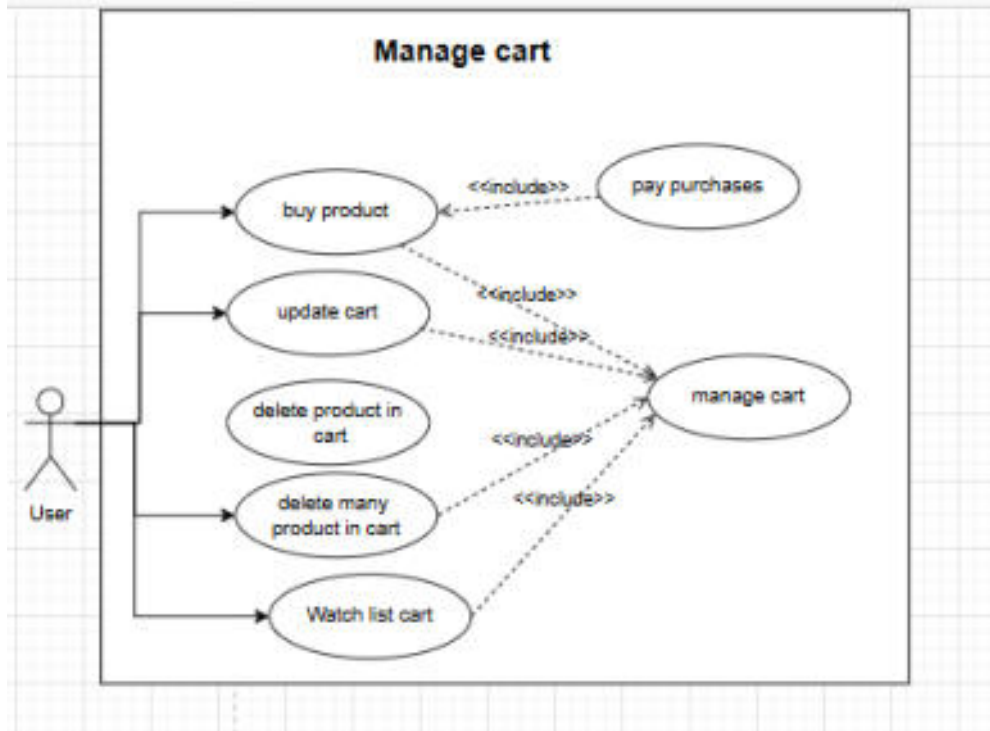
Hình 2.8. Usecase user quản lí địa chỉ đặt hàng

Bảng 2.8. Đặc tả usecase user quản lí địa chỉ đặt hàng

Tên Usecase	Quản lý địa chỉ nhận hàng
Tác nhân chính	Người dùng (Khách hàng)
Mô tả Usecase	Người dùng thêm mới, sửa, xóa hoặc chọn địa chỉ nhận hàng mặc định trong danh sách các địa chỉ đã lưu trữ.
Tiền điều kiện	- Người dùng đã đăng nhập vào hệ thống.
Hậu điều kiện	<ol style="list-style-type: none"> Danh sách địa chỉ nhận hàng được cập nhật thành công (thêm, sửa, xóa hoặc chọn mặc định). Thông tin địa chỉ nhận hàng hiển thị đúng trên giao diện.
Kích hoạt	<ol style="list-style-type: none"> Người dùng nhấn vào "Quản lý địa chỉ" trong giao diện tài khoản. Thực hiện một trong các hành động sau: <ul style="list-style-type: none"> Thêm địa chỉ mới: Nhập thông tin địa chỉ mới (tên, số điện thoại, địa chỉ) và nhấn "Lưu". Sửa địa chỉ: Nhấn vào địa chỉ trong danh sách, cập nhật thông tin mới, nhấn "Lưu". Xóa địa chỉ: Nhấn nút "Xóa" kế bên địa chỉ mong muốn.

- **Chọn mặc định:** Nhấn nút "Chọn làm mặc định" bên cạnh địa chỉ mong muốn.
3. Hệ thống xác nhận và lưu thay đổi.

2.2.8. User quản lý giỏ hàng



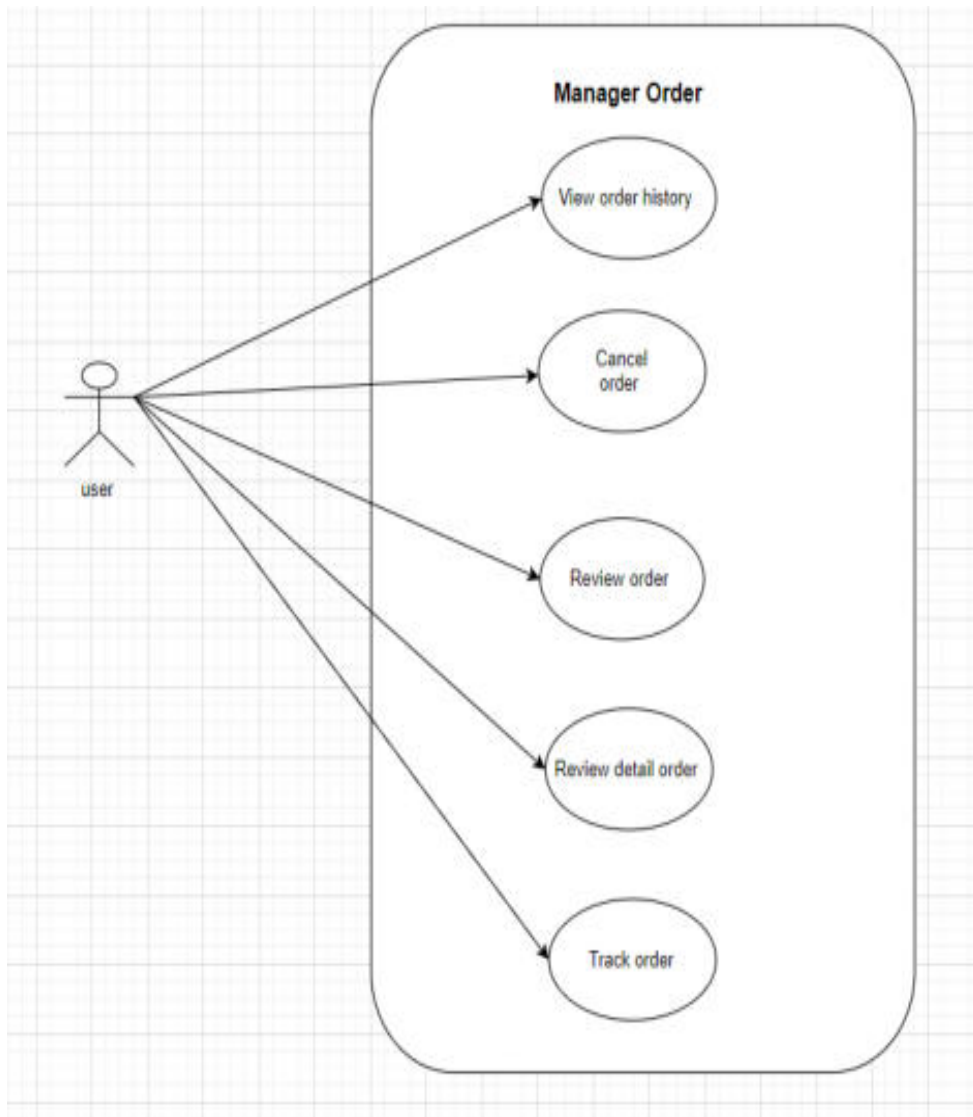
Hình 2.9. Usecase user quản lý giỏ hàng

Bảng 2.9. Đặc tả usecase user quản lý giỏ hàng

Tên Usecase	Quản lý giỏ hàng
Tác nhân chính	Người dùng (Khách hàng)
Mô tả Usecase	Người dùng thêm, sửa, xóa sản phẩm trong giỏ hàng hoặc cập nhật số lượng sản phẩm.
Tiền điều kiện	<ul style="list-style-type: none">- Người dùng đã đăng nhập vào hệ thống.- Hệ thống đã có danh sách sản phẩm hiển thị cho người dùng.
Hậu điều kiện	<ul style="list-style-type: none">• Giỏ hàng được cập nhật thành công (thêm, sửa, xóa sản phẩm).

	<ul style="list-style-type: none">• Thông tin giỏ hàng được lưu trữ và hiển thị đúng trên giao diện.
Kích hoạt	<ul style="list-style-type: none">• Người dùng truy cập giao diện sản phẩm.• Nhấn "Thêm vào giỏ hàng".• Vào mục "Giỏ hàng" và xem danh sách các sản phẩm đã thêm.• Thực hiện một trong các tác vụ:<ul style="list-style-type: none">○ Thay đổi số lượng○ Xóa sản phẩm: Nhấn nút "Xóa" kế bên sản phẩm.

2.2.9. User quản lý đơn hàng



Hình 2.10. Usecase user quản lý đơn hàng

Bảng 2.10. Đặc tả usecase user quản lý đơn hàng

Tên Usecase	Quản lý đơn hàng
Tác nhân chính	Người dùng (Khách hàng)
Mô tả Usecase	Người dùng thực hiện các thao tác quản lý đơn hàng bao gồm xem lịch sử đơn hàng, hủy đơn hàng, đánh giá đơn hàng, xem chi tiết đơn hàng, và theo dõi tình trạng đơn hàng.
Tiền điều kiện	<ol style="list-style-type: none">1. Người dùng đã đăng nhập vào hệ thống.2. Đã có đơn hàng được tạo trong hệ thống.
Hậu điều kiện	<ol style="list-style-type: none">1. Lịch sử đơn hàng được hiển thị (nếu xem).2. Đơn hàng đã hủy được cập nhật tình trạng (nếu hủy).3. Đánh giá đơn hàng được lưu (nếu đánh giá).4. Thông tin chi tiết đơn hàng được hiển thị (nếu xem).5. Tình trạng đơn hàng được cập nhật trong giao diện (nếu theo dõi).
Kích hoạt	<ol style="list-style-type: none">1. Người dùng nhấn vào "Quản lý đơn hàng" trong giao diện tài khoản.2. Thực hiện một trong các hành động sau:<ul style="list-style-type: none">○ Xem lịch sử đơn hàng: Hệ thống hiển thị danh sách các đơn hàng.○ Hủy đơn hàng: Chọn đơn hàng, nhấn "Hủy", nhập lý do (nếu có), và xác nhận.○ Đánh giá đơn hàng: Chọn đơn hàng, nhập đánh giá và nhấn "Gửi".○ Xem chi tiết đơn hàng: Nhấn vào đơn hàng mong muốn, xem thông tin chi tiết.○ Theo dõi đơn hàng: Nhấn vào tình trạng đơn hàng để xem các cập nhật.3. Hệ thống xác nhận và thực hiện yêu cầu.

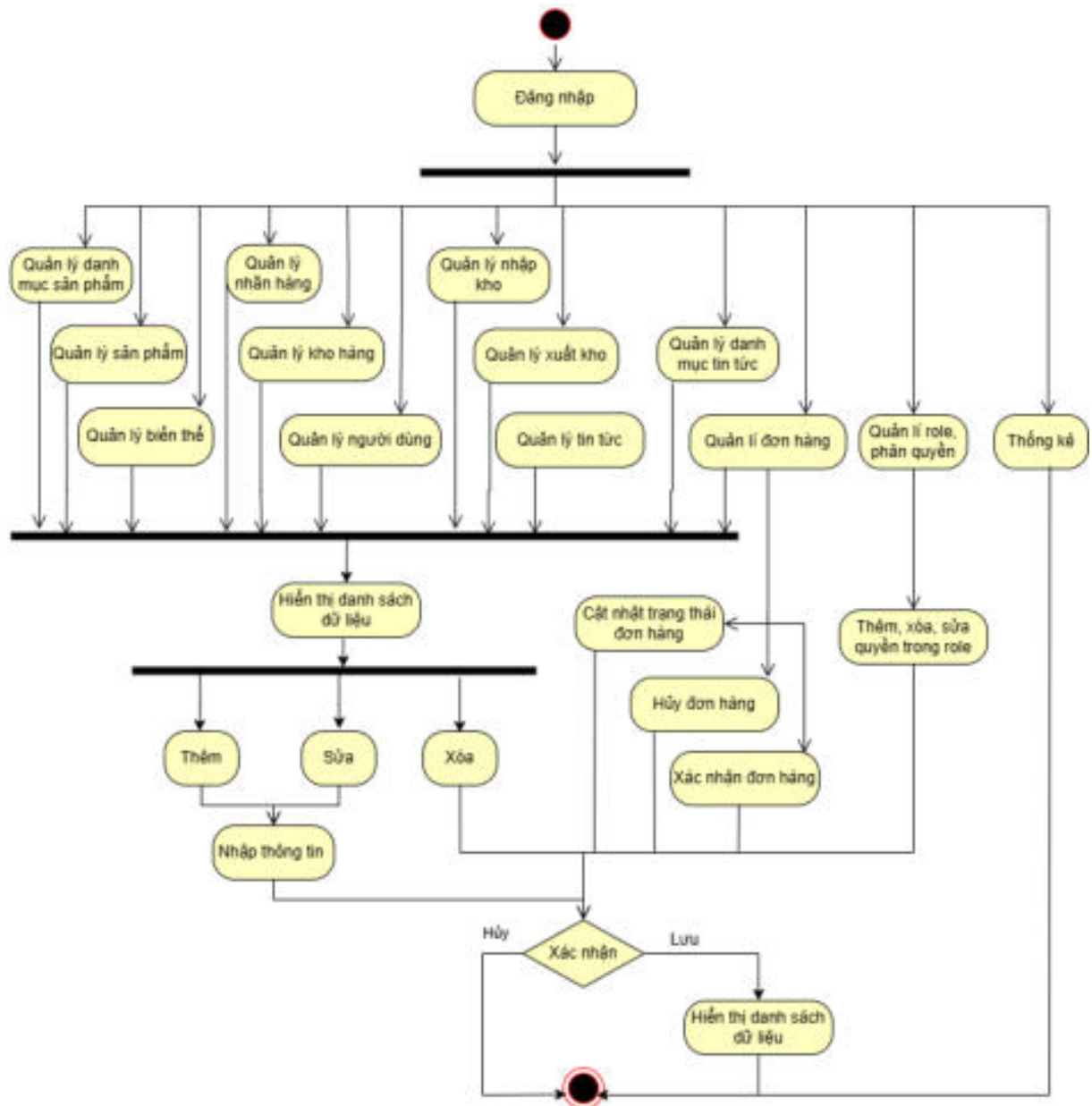
2.3. Sơ đồ hoạt động

2.3.1. Khách hàng



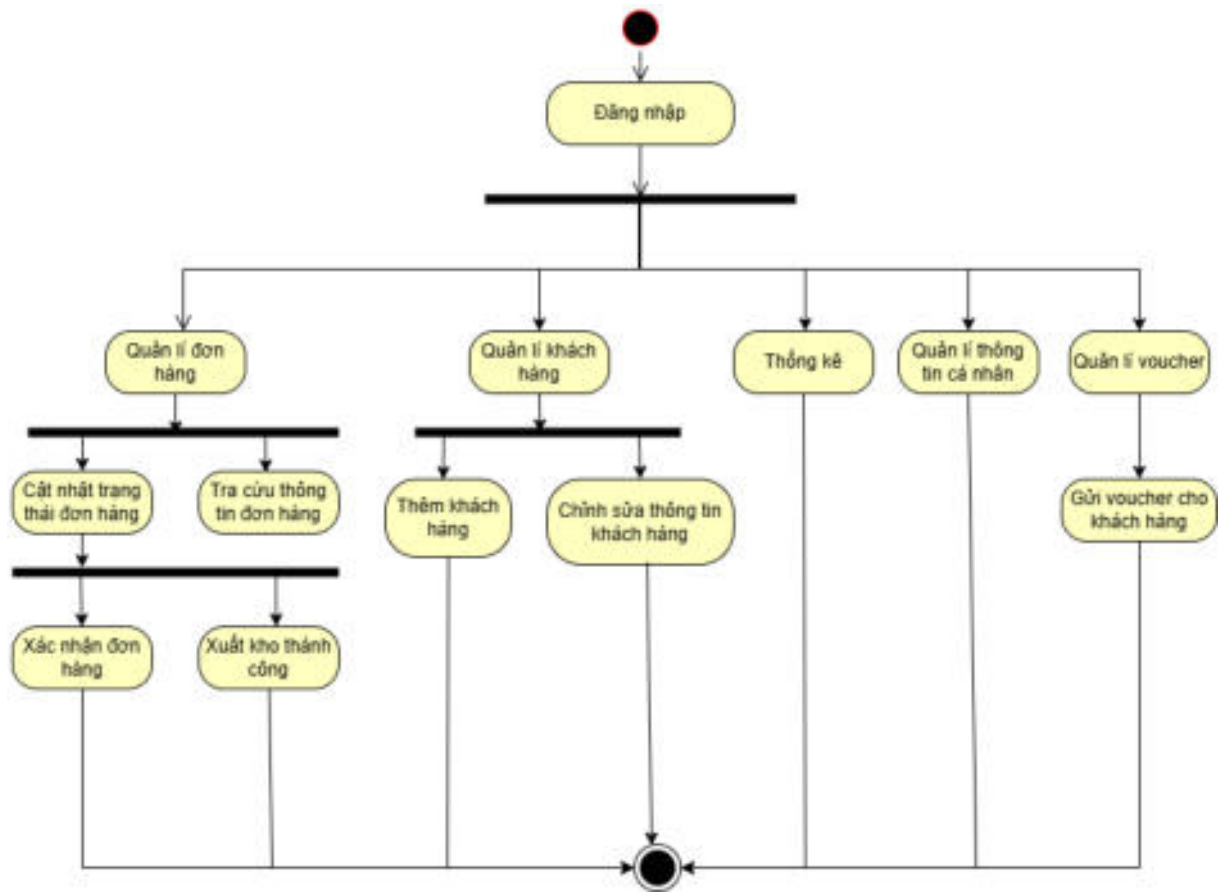
Hình 2.11. Sơ đồ hoạt động của khách hàng

2.3.2. Quản trị viên admin



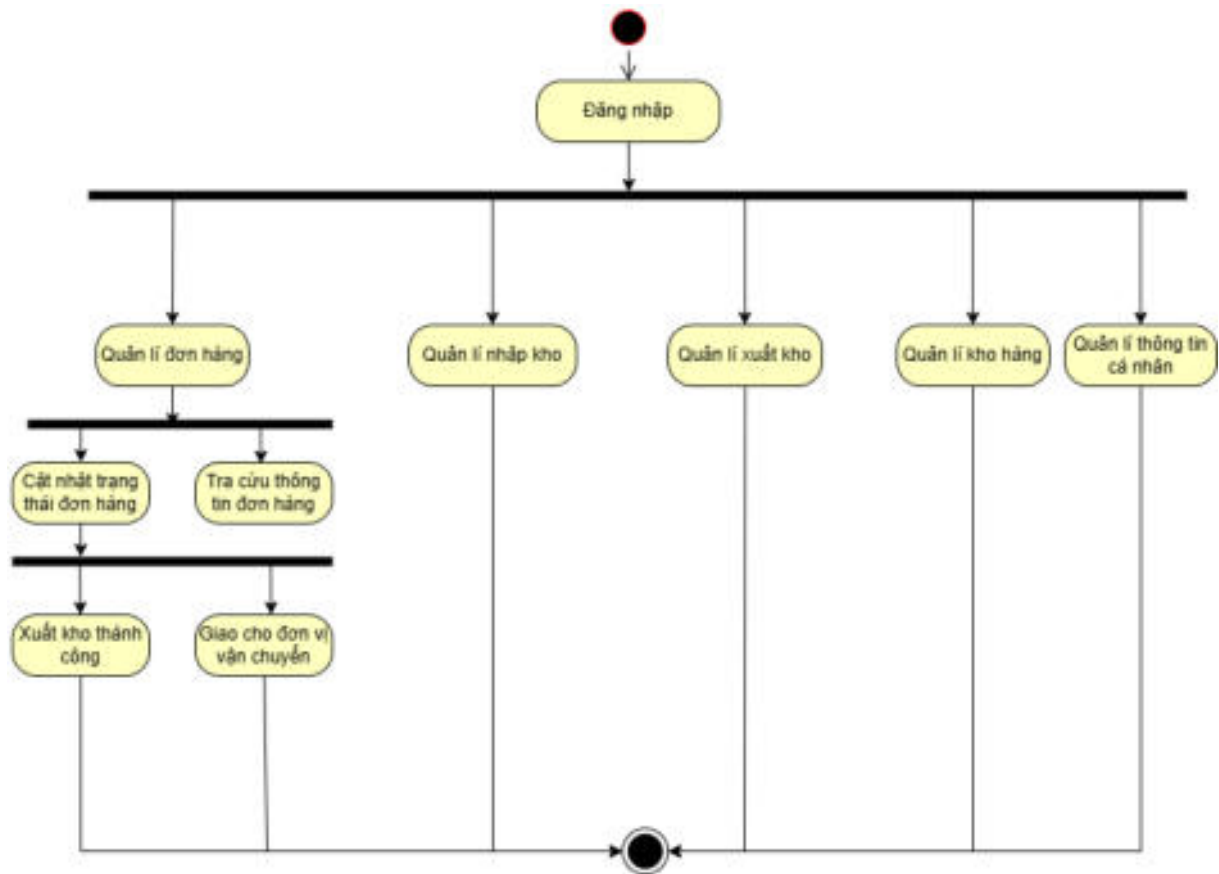
Hình 2.12. Sơ đồ hoạt động của admin

2.3.3. Nhân viên bán hàng



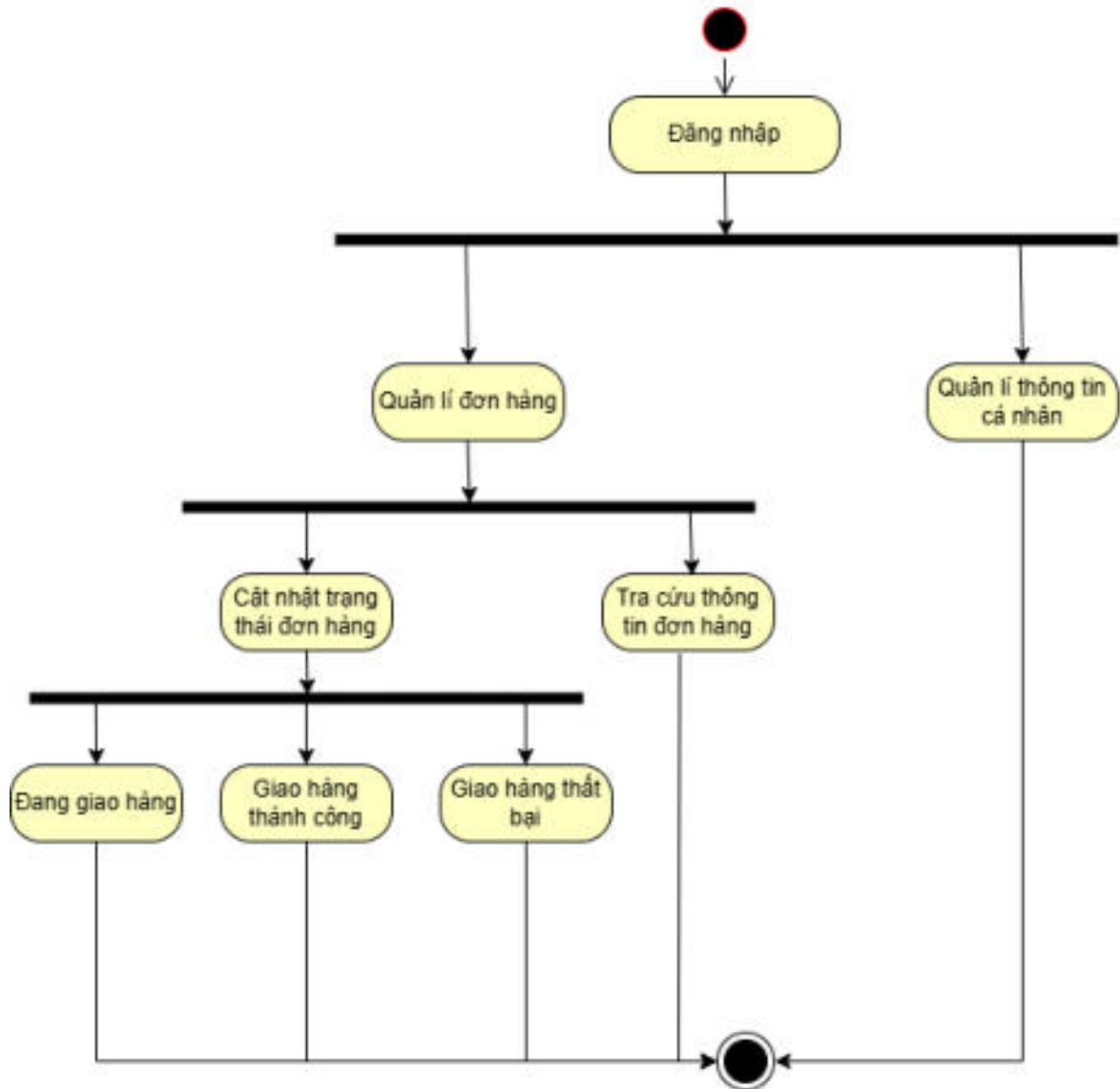
Hình 2.13. Sơ đồ hoạt động của nhân viên bán hàng

2.3.4. Quản lí kho



Hình 2.14. Sơ đồ hoạt động của quản lí kho

2.3.5. Nhân viên giao hàng



Hình 2.15. Sơ đồ hoạt động của nhân viên giao hàng

2.4. Thiết kế cơ sở dữ liệu

2.4.1. Mô hình thiết kế cơ sở dữ liệu

2.4.1.1. Các bảng trong cơ sở dữ liệu

- **Products:** Là bảng chứa thông tin chi tiết của sản phẩm
- **Brand:** Là bảng chứa thông tin nhãn hàng cung cấp hàng hóa cho hệ thống
- **Category:** Là bảng chứa các thông tin về danh mục sản phẩm
- **Specification:** Là bảng thông số kỹ thuật, chứa các thông số kỹ thuật riêng biệt, cụ thể của sản phẩm, cụ thể là xe hơi và xe máy điện
- **Skus:** Là bảng đơn vị lưu kho của sản phẩm, chứa các thông tin, giá trị riêng biệt của từng loại sản phẩm, mỗi SKU đại diện cho một biến thể cụ thể của sản phẩm

- **Option:** Là bảng thuộc tính, chứa thông tin về các thuộc tính của biến thể sản phẩm như màu sắc, trọng lượng, kích cỡ, ...
- **Option_Value:** Là bảng giá trị thuộc tính, chứa một hoặc nhiều giá trị của mỗi thuộc tính
- **Import:** Là bảng chứa thông tin của các phiếu nhập kho trong hệ thống
- **Detail_Import:** Là bảng chi tiết phiếu nhập kho của biến thể sản phẩm, chứa các giá trị liên quan đến tồn kho, như số lượng tồn kho, giá vốn nhập, số lượng nhập vào, ... Mỗi chi tiết nhập kho đại diện cho một lô hàng của biến thể sản phẩm
- **Warehouse:** Là bảng chứa thông tin chi tiết của các kho hàng có trong hệ thống
- **Export:** Là bảng chứa thông tin các phiếu xuất kho trong hệ thống
- **Detail_Export:** Là bảng chi tiết phiếu xuất kho, chứa thông tin liên quan đến giá trị xuất kho, như số lượng xuất kho của mỗi biến thể sản phẩm, kho xuất, lô xuất, ...
- **Blog:** Là bảng chứa thông tin các danh mục tin tức
- **Blog_Category:** Là bảng chứa thông tin các tin tức có trong mỗi danh mục tin tức
- **Review:** Là bảng chứa các dữ liệu đánh giá gồm rating 1-5 sao và bình luận của khách hàng
- **Role:** Là bảng định nghĩa **vai trò** (role) trong hệ thống như: Quản trị viên, Nhân viên quản lý kho, Nhân viên bán hàng... Mỗi vai trò sẽ được gán các quyền khác nhau.
- **Permission:** Là bảng định nghĩa các **quyền thao tác** trong hệ thống (ví dụ: Xem danh sách người dùng, Tạo sản phẩm, Xuất kho...), dùng để gán cho các vai trò (Role).
- **User:** Là bảng chứa thông tin tài khoản cá nhân của mỗi người dùng có vai trò quản lý trong hệ thống
- **User_have_roles:** Là bảng chứa vai trò của một tài khoản user dùng để quản lý, vận hành hệ thống
- **Role_Permission:** Là bảng chứa các quyền hạn của một role trong hệ thống
- **Customer:** Là bảng chứa thông tin về tài khoản của mỗi khách hàng sử dụng hệ thống
- **Receive_address:** Là bảng chứa thông tin về các địa chỉ nhận hàng của khách hàng
- **Order:** Là bảng chứa thông tin của một hóa đơn khi mua hàng của khách
- **Order_Detail:** Là bảng chứa thông tin chi tiết của một đơn hàng
- **Vouchers:** Là bảng chứa thông tin chi tiết về các phiếu giảm giá, quà tặng, mã khuyến mãi
- **TypeVoucher:** Là bảng chứa kiểu của phiếu giảm giá
- **Customer_Voucher:** Là bảng chứa thông tin về các phiếu giảm giá của một khách hàng hoặc nhiều khách hàng
- **Cart:** Là bảng chứa thông tin về giỏ hàng của một khách hàng

5	status	bool	Trạng thái sản phẩm (hoạt động/không)
6	brand_id	UUID	Mã thương hiệu của sản phẩm
7	category_id	UUID	Mã danh mục của sản phẩm
8	Type	_text	Loại sản phẩm
9	images	JSON	Các ảnh của sản phẩm
10	createdAt	timestamp	Thời gian tạo sản phẩm
11	updatedAt	timestamp	Thời gian cập nhật sản phẩm
12	deletedAt	timestamp	Thời gian xóa sản phẩm

2.4.2.2. Brand

Bảng 2.12. Mô tả bảng brand

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh thương hiệu
2	name	varchar(255)	Tên thương hiệu
3	description	text	Mô tả thương hiệu
4	thumbnailUrl	varchar(255)	Đường dẫn ảnh đại diện thương hiệu
5	createdAt	timestamp	Thời gian tạo thương hiệu
6	updatedAt	timestamp	Thời gian cập nhật thương hiệu
7	slug	varchar(255)	Slug của thương hiệu

2.4.2.3. Categories

Bảng 2.13. Mô tả bảng Categories

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh danh mục
2	name	varchar(255)	Tên danh mục
3	createdAt	timestamp	Thời gian tạo danh mục
4	updatedAt	timestamp	Thời gian cập nhật danh mục
5	deletedAt	timestamp	Thời gian xóa danh mục
6	slug	string	Slug của danh mục
7	parentCategoryId	UUID	Mã danh mục cha (nếu có)
8	description	text	Ghi chú về danh mục
9	type	enum	Kiểu hiển thị của danh mục

2.4.2.4. Specification

Bảng 2.14. Mô tả bảng Specification

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	serial4	Mã định danh thông số kỹ thuật
2	product_id	UUID	Mã định danh sản phẩm
3	name	Varchar(255)	Tên thông số kỹ thuật
4	value	text	Giá trị thông số kỹ thuật

2.4.2.5. Skus

Bảng 2.15. Mô tả bảng skus

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh SKU (dạng UUID)
2	masku	Varchar	Mã SKU (mã nội bộ)
3	barcode	Varchar	Mã vạch của SKU
4	name	Varchar	Tên SKU (ví dụ: màu sắc, kích thước)
5	status	Boolean	Trạng thái SKU (hoạt động/không)
6	product_id	UUID	Mã sản phẩm liên kết
7	price_sold	Numeric	Giá bán của SKU
8	price_compare	Numeric	Giá gốc để so sánh (giảm giá)
9	image	Text	Đường dẫn ảnh của SKU
10	createdAt	Timestamp	Thời gian tạo SKU
11	updatedAt	Timestamp	Thời gian cập nhật SKU
12	deletedAt	Timestamp	Thời gian xóa SKU

2.4.2.6. Option

Bảng 2.16. Mô tả bảng Option

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh thuộc tính
2	name	varchar(255)	Tên tài khoản người dùng
3	createdAt	timestamp	Thời gian tạo thuộc tính
4	updatedAt	timestamp	Thời gian cập nhật thuộc tính

2.4.2.7. Option_Value

Bảng 2.17. Mô tả bảng Option_Value

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh giá trị thuộc tính
2	value	varchar(255)	Giá trị thuộc tính
3	createdAt	timestamp	Thời gian tạo giá trị thuộc tính
4	updatedAt	timestamp	Thời gian cập nhật giá trị thuộc tính
5	option_id	UUID	Mã định danh thuộc tính
6	skus_id	UUID	Mã định danh skus

2.4.2.8. Import

Bảng 2.18. Mô tả bảng Import

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh phiếu nhập hàng
2	username	varchar(255)	Tên tài khoản người dùng
3	createdAt	timestamp	Thời gian tạo phiếu nhập hàng
4	updatedAt	timestamp	Thời gian cập nhật phiếu nhập hàng
5	user_id	UUID	Mã định danh người dùng

2.4.2.9. Detail_Import

Bảng 2.19. Mô tả bảng chi tiết nhập kho

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	price_import	Numeric	Giá nhập hàng
2	quantity_import	Int4	Số lượng nhập
3	quantity_sold	Int4	Số lượng đã bán
4	created_at	Timestamp	Thời gian tạo chi tiết nhập
5	updated_at	Timestamp	Thời gian cập nhật chi tiết nhập
6	import_id	UUID	Mã phiếu nhập hàng liên kết
7	skus_id	UUID	Mã SKU của sản phẩm nhập

8	warehouse_id	UUID	Mã kho hàng liên kết
9	id	UUID	Mã định danh chi tiết nhập
10	deleted_at	Timestamp	Thời gian xóa chi tiết nhập
11	quantity_remaining	Int4	Số lượng còn lại trong kho
12	lot_name	Varchar	Tên lô hàng

2.4.2.10. Warehouse

Bảng 2.20. Mô tả bảng Warehouse

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh kho
2	name	varchar(255)	Tên kho
3	description	text	Mô tả kho
4	created_at	Timestamp	Thời gian tạo kho
5	updated_at	Timestamp	Thời gian cập nhật kho
6	address	varchar(255)	Địa chỉ kho

2.4.2.11. Export

Bảng 2.21. Mô tả bảng Export

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	uuid	Mã phiếu xuất
2	user_id	uuid	Người tạo phiếu xuất.
3	order_id	uuid	Đơn hàng xuất
4	note	varchar(255)	Ghi chú.
5	createdAt	Timestamp	Thời gian tạo phiếu xuất
6	updatedAt	Timestamp	Thời gian cập nhật phiếu xuất
7	deletedAt	Timestamp	Thời gian xóa phiếu xuất

2.4.2.12. Detail Export

Bảng 2.22. Mô tả bảng Detail Export

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	uuid	Mã detail export.
2	Skus_id	uuid	Mã biến thể sản phẩm liên kết với bảng Skus.
3	Export_id	uuid	Mã export liên kết bảng export.
4	Warehouse_id	uuid	Mã kho liên kết với bảng Warehouse.
5	Detail_import_id	uuid	Mã chi tiết phiếu nhập liên kết bảng Detail_import.
6	Quantity_export	Int4	Số lượng xuất.

7	Created_At	timestamp	Ngày tạo chi tiết phiếu xuất
8	updatedAt	Timestamp	Thời gian cập nhật chi tiết phiếu xuất
9	deletedAt	Timestamp	Thời gian xóa chi tiết phiếu xuất

2.4.2.13. Blog

Bảng 2.23. Mô tả bảng blog

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh bài viết
2	title	varchar(255)	Tiêu đề bài viết
3	slug	varchar(255)	Đường dẫn tĩnh của bài viết
4	thumbnail	varchar(255)	Đường dẫn ảnh đại diện bài viết
5	content	text	Nội dung bài viết
6	blogImages	text	Danh sách ảnh trong bài viết
7	blogCategoryId	UUID	Mã danh mục của bài viết
8	customerId	UUID	Mã khách hàng viết bài
9	createdAt	timestamp	Thời gian tạo bài viết
10	updatedAt	timestamp	Thời gian cập nhật bài viết
11	deletedAt	timestamp	Thời gian xóa bài viết

2.4.2.14. Blog_Category

Bảng 2.24. Mô tả bảng Blog_Category

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh danh mục bài viết (dạng UUID)
2	name	Varchar	Tên danh mục bài viết
3	description	Varchar	Mô tả danh mục bài viết
4	slug	Varchar	Đường dẫn tĩnh của danh mục
5	createdAt	Timestamp	Thời gian tạo danh mục
6	updatedAt	Timestamp	Thời gian cập nhật danh mục
7	deletedAt	Timestamp	Thời gian xóa danh mục

2.4.2.15. Review

Bảng 2.25. Mô tả bảng Review

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh đánh giá (dạng UUID)
2	createdAt	Timestamp	Thời gian tạo đánh giá
3	updatedAt	Timestamp	Thời gian cập nhật đánh giá
4	deletedAt	Timestamp	Thời gian xóa đánh giá
5	review_rating	Float8	Điểm đánh giá (thang điểm số thực)

6	review_comment	Text	Nội dung đánh giá
7	customer_id	UUID	Mã khách hàng viết đánh giá
8	product_id	UUID	Mã sản phẩm được đánh giá
9	order_id	UUID	Mã đơn hàng liên quan đến đánh giá

2.4.2.16. Role

Bảng 2.26. Mô tả bảng Role

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	uuid	Mã định danh vai trò, khóa chính.
2	name	varchar(255)	Tên vai trò (ví dụ: "Admin", "User", "Manager").
3	description	Text	Mô tả chi tiết về vai trò, có thể NULL.
4	isActive	bool	Trạng thái hoạt động của vai trò (true: hoạt động, false: không hoạt động).
5	createdAt	timestamp	Thời gian tạo vai trò.
6	updatedAt	timestamp	Thời gian cập nhật vai trò.
7	deletedAt	timestamp	Thời gian xóa mềm vai trò, có thể NULL.

2.4.2.17. Permission

Bảng 2.27. Mô tả bảng Permission

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	uuid	Mã định danh quyền, khóa chính.
2	name	varchar(255)	Tên quyền (ví dụ: "Xem đơn hàng", "Sửa sản phẩm").
3	path	varchar(255)	Đường dẫn API hoặc tài nguyên mà quyền áp dụng (ví dụ: "/api/orders").
4	method	varchar(255)	Phương thức HTTP áp dụng cho quyền (ví dụ: "GET", "POST", "PUT").
5	module	varchar(255)	Tên module hoặc phân hệ mà quyền thuộc về (ví dụ: "Order", "Product").
6	createdAt	timestamp	Thời gian tạo quyền.
7	updatedAt	timestamp	Thời gian cập nhật quyền
8	deletedAt	timestamp	Thời gian xóa mềm quyền, có thể NULL.

2.4.2.18. User

Bảng 2.28. Mô tả bảng User

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh người dùng
2	username	varchar(255)	Tên tài khoản người dùng
3	password	varchar(255)	Mật khẩu người dùng
4	email	varchar(255)	Email người dùng
5	age	int4	Tuổi người dùng
6	address	varchar(255)	Địa chỉ người dùng
7	phoneNumber	varchar(255)	Số điện thoại người dùng
8	gender	enum	Giới tính người dùng
9	avatarUrl	varchar(255)	Đường dẫn ảnh đại diện
10	isActive	bool	Trạng thái hoạt động của tài khoản
11	codeId	varchar(255)	Mã xác thực của người dùng
12	codeExpired	timestamp	Thời gian hết hạn mã xác thực

13	birthday	timestamp	Ngày sinh người dùng
14	createdAt	timestamp	Thời gian tạo tài khoản
15	updatedAt	timestamp	Thời gian cập nhật tài khoản
16	deletedAt	timestamp	Thời gian xóa tài khoản

2.4.2.19. User_have_roles

Bảng 2.29. Mô tả bảng User_have_roles

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	role_id	uuid	Mã role
2	user_id	uuid	Mã user

2.4.2.20. Role_Permission

Bảng 2.30. Mô tả bảng Role_Permission

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	role_id	uuid	Mã role
2	permission_id	uuid	Mã quyền

2.4.2.21. Customer

Bảng 2.31. Mô tả bảng Customer

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	uuid	Mã khách hàng, khóa chính của bảng khách hàng
2	username	varchar(255) utf8mb4_unicode_ci	Tên đầy đủ của khách hàng
3	email	varchar(255) utf8mb4_unicode_ci	Địa chỉ email của khách hàng (có thể dùng làm tên đăng nhập).
4	password	varchar(255) utf8mb4_unicode_ci	Mật khẩu của khách hàng
5	phoneNumber	varchar(255) utf8mb4_unicode_ci	Số điện thoại của khách hàng
6	Birthday	date	Ngày sinh của khách hàng
7	gender	tinyint(1)	Giới tính của khách hàng (1: Nam, 0: Nữ).
8	AvatarUrl	varchar(255) utf8mb4_unicode_ci	Hình đại diện của khách hàng
12	role_id	Uuid()	id của vai trò
13	Code_id	Uuid()	Mã code khi đăng kí tài khoản
14	isActive	bool	Trạng thái kích hoạt tài khoản
14	createdAt	timestamp	Thời gian tạo tài khoản khách hàng
15	updatedAt	timestamp	Thời gian cập nhật tài khoản khách hàng
16	deletedAt	timestamp	Thời gian xóa tài khoản khách hàng

2.4.2.22. Receive_Address

Bảng 2.32. Mô tả bảng Receive_Address

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh địa chỉ nhận hàng (dạng UUID)
2	createdAt	Timestamp	Thời gian tạo địa chỉ
3	updatedAt	Timestamp	Thời gian cập nhật địa chỉ
4	deletedAt	Timestamp	Thời gian xóa địa chỉ
5	receiver_name	Varchar(5)	Tên người nhận hàng
6	receiver_phone	Varchar(15)	Số điện thoại người nhận
7	street	Varchar(100)	Tên đường
8	ward	Varchar(100)	Phường/xã
9	district	Varchar(100)	Quận/huyện
10	province	Varchar(100)	Tỉnh/thành phố
11	postal_code	Varchar(10)	Mã bưu điện
12	note	Varchar(100)	Ghi chú địa chỉ nhận hàng
13	is_default	Boolean	Địa chỉ mặc định hay không
14	customer_id	UUID	Mã khách hàng sở hữu địa chỉ

2.4.2.23. Orders

Bảng 2.33. Mô tả bảng Orders

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh đơn hàng
2	createdAt	timestamp	Thời gian tạo đơn hàng
3	updatedAt	timestamp	Thời gian cập nhật đơn hàng
4	deletedAt	timestamp	Thời gian xóa đơn hàng
5	order_status	enum	Trạng thái đơn hàng (ví dụ: pending, shipping)
6	note	text	Ghi chú về đơn hàng
7	customer_id	UUID	Mã khách hàng đặt đơn
8	receive_address_id	UUID	Mã địa chỉ nhận hàng
9	payment_time	timestamp	Thời gian thanh toán
10	payment_status	enum	Trạng thái thanh toán (ví dụ: success, failed)
11	payment_method_id	UUID	Mã phương thức thanh toán
12	delivery_method_id	UUID	Mã phương thức giao hàng
13	delivery_time	timestamp	Thời gian giao hàng dự kiến
14	total_price	float8	Tổng giá trị đơn hàng
15	discount_price	float8	Giá sau khi giảm giá
16	payment_url	text	Đường dẫn thanh toán

17	payment_url_expired	timestamp	Thời gian hết hạn đường dẫn thanh toán
18	refund_time	timestamp	Thời gian hoàn tiền (nếu có)
19	refund_reason	text	Lý do hoàn tiền

2.4.2.24. Order_Details

Bảng 2.34. Mô tả bảng Order_Details

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	uuid	Mã định danh chi tiết đơn hàng, tự tăng, khóa chính.
2	order_id	uuid	Mã đơn hàng, tham chiếu đến bảng "order".
3	skus_id	uuid	Mã biến thể sản phẩm, tham chiếu đến bảng "skus".
4	quantity	int4	Số lượng biến thể sản phẩm trong đơn hàng.
5	createdAt	timestamp	Thời gian tạo chi tiết đơn hàng
6	updatedAt	timestamp	Thời gian cập nhật chi tiết đơn hàng
7	deletedAt	timestamp	Thời gian xóa chi tiết đơn hàng

2.4.2.25. Voucher

Bảng 2.35. Mô tả bảng Voucher

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh phiếu giảm giá (dạng UUID)
2	voucher_code	Varchar	Mã code của phiếu giảm giá
3	voucher_name	Varchar	Tên phiếu giảm giá
4	description	Text	Mô tả chi tiết phiếu giảm giá
5	uses	Int4	Số lần đã sử dụng
6	max_uses_user	Int4	Số lần tối đa mỗi người dùng
7	discount_amount	Numeric(10,2)	Số tiền giảm giá
8	fixed	Boolean	Trạng thái cố định (có/không)
9	status	Public("vouchers_status_enum")	Trạng thái phiếu (enum)
10	typeVoucherId	UUID	Mã loại phiếu giảm giá
11	start_date	Date	Ngày bắt đầu hiệu lực
12	end_date	Date	Ngày kết thúc hiệu lực
13	limit	Int4	Giới hạn sử dụng (nếu có)
14	count_user_get	Int4	Số lượng người dùng nhận
15	createdAt	Timestamp	Thời gian tạo phiếu giảm giá

16	updatedAt	Timestamp	Thời gian cập nhật phiếu giảm giá
----	-----------	-----------	-----------------------------------

2.4.2.26. TypeVoucher

Bảng 2.36. Mô tả bảng kiểu TypeVoucher

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	uuid	Mã địa chỉ người nhận, tự tăng, khóa chính.
2	name_type_voucher	Varchar	Tên loại voucher.
3	createdAt	timestamp	Thời gian tạo.
4	updatedAt	timestamp	Thời gian cập nhật.
5	deletedAt	timestamp	Thời gian xóa.

2.4.2.27. Customer_Voucher

Bảng 2.37. Mô tả bảng Customer_Voucher

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh phiếu giảm giá của khách hàng (dạng UUID)
2	is_used	Boolean	Trạng thái sử dụng (đã dùng/chưa dùng)
3	used_at	Timestamp	Thời gian sử dụng phiếu
4	customer_id	UUID	Mã khách hàng sở hữu phiếu
5	voucher_id	UUID	Mã phiếu giảm giá liên kết
6	createdAt	Timestamp	Thời gian tạo phiếu cho khách

7	updatedAt	Timestamp	Thời gian cập nhật phiếu
8	deletedAt	Timestamp	Thời gian xóa phiếu

2.4.2.28. Cart

Bảng 2.38. Mô tả bảng Cart

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh giỏ hàng, tự tăng, khóa chính.
2	customer_id	UUID	Mã người dùng, tham chiếu đến bảng "customer".
3	createdAt	timestamp	Thời gian tạo giỏ hàng
4	updatedAt	timestamp	Thời gian cập nhật giỏ hàng
5	deletedAt	timestamp	Thời gian xóa giỏ hàng

2.4.2.29. Cart_Item

Bảng 2.39. Mô tả bảng Cart_Item

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh mục trong giỏ hàng (dạng UUID)
2	quantity	Int4	Số lượng sản phẩm trong mục
3	cart_id	UUID	Mã giỏ hàng chứa mục này
4	skus_id	UUID	Mã SKU của sản phẩm trong mục
5	createdAt	Timestamp	Thời gian tạo mục trong giỏ
6	updatedAt	Timestamp	Thời gian cập nhật mục trong giỏ

7	deletedAt	Timestamp	Thời gian xóa mục khỏi giỏ
---	-----------	-----------	----------------------------

2.4.2.30. Delivery_Method

Bảng 2.40. Mô tả bảng Delivery_Method

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	UUID	Mã định danh phương thức giao hàng (dạng UUID)
2	createdAt	Timestamp	Thời gian tạo phương thức giao hàng
3	updatedAt	Timestamp	Thời gian cập nhật phương thức giao hàng
4	deletedAt	Timestamp	Thời gian xóa phương thức giao hàng
5	name	Varchar(100)	Tên phương thức giao hàng
6	description	Text	Mô tả phương thức giao hàng
7	fee	Int4	Phí giao hàng
8	logo	Varchar	Đường dẫn logo của phương thức giao hàng

2.4.2.31. Payment_Method

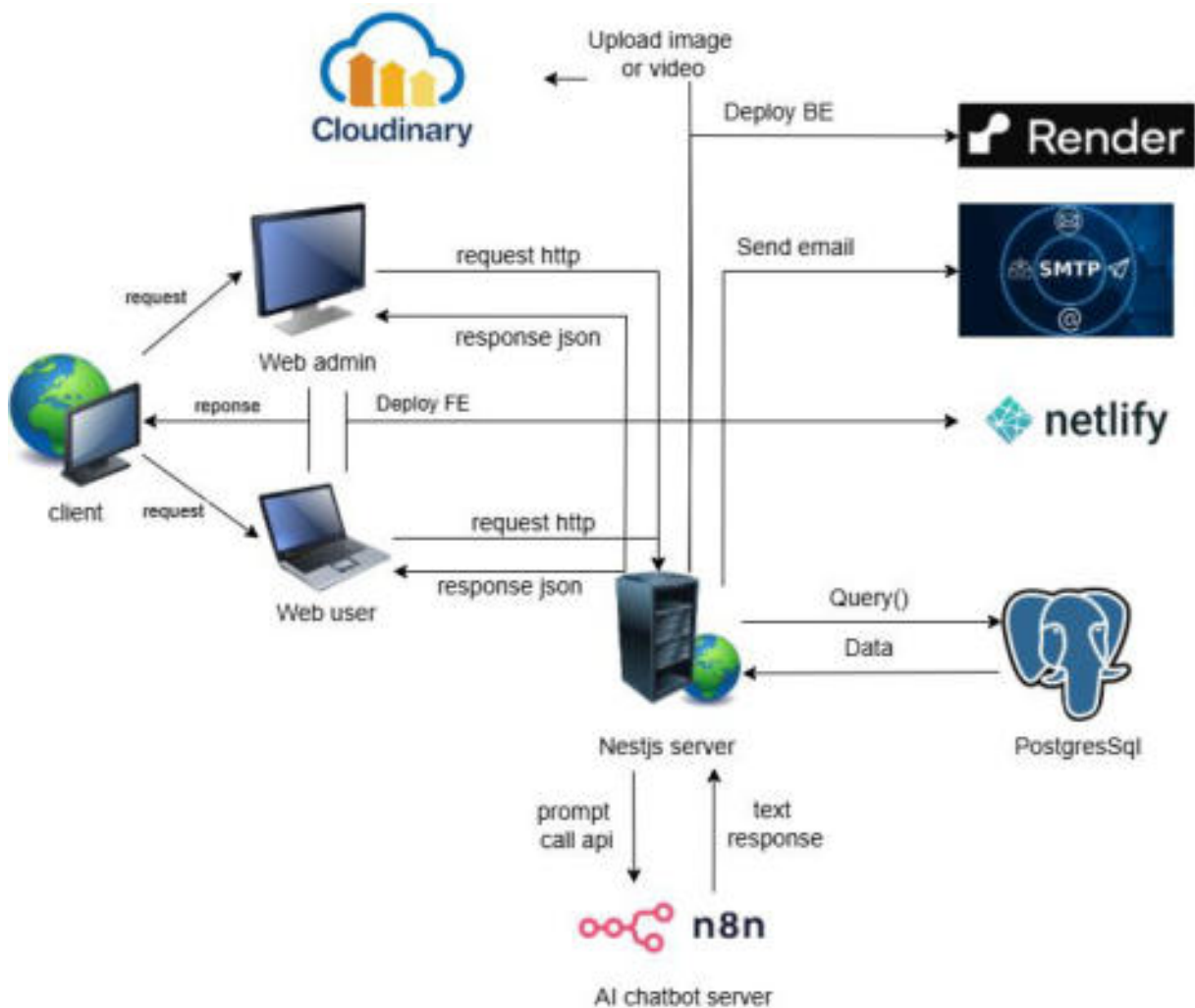
Bảng 2.41. Mô tả bảng Payment_Method

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	uuid	Mã thanh toán, tự tăng, khóa chính.
2	name	varchar	Tên phương thức thanh toán.
3	description	Text	Tóm tắt miêu tả.

4	logo	Varchar	Ảnh logo phương thức thanh toán.
5	createdAt	Timestamp	Thời gian tạo phương thức thanh toán
6	updatedAt	Timestamp	Thời gian cập nhật phương thức thanh toán
7	deletedAt	Timestamp	Thời gian xóa phương thức thanh toán

2.5. Kiến trúc hệ thống

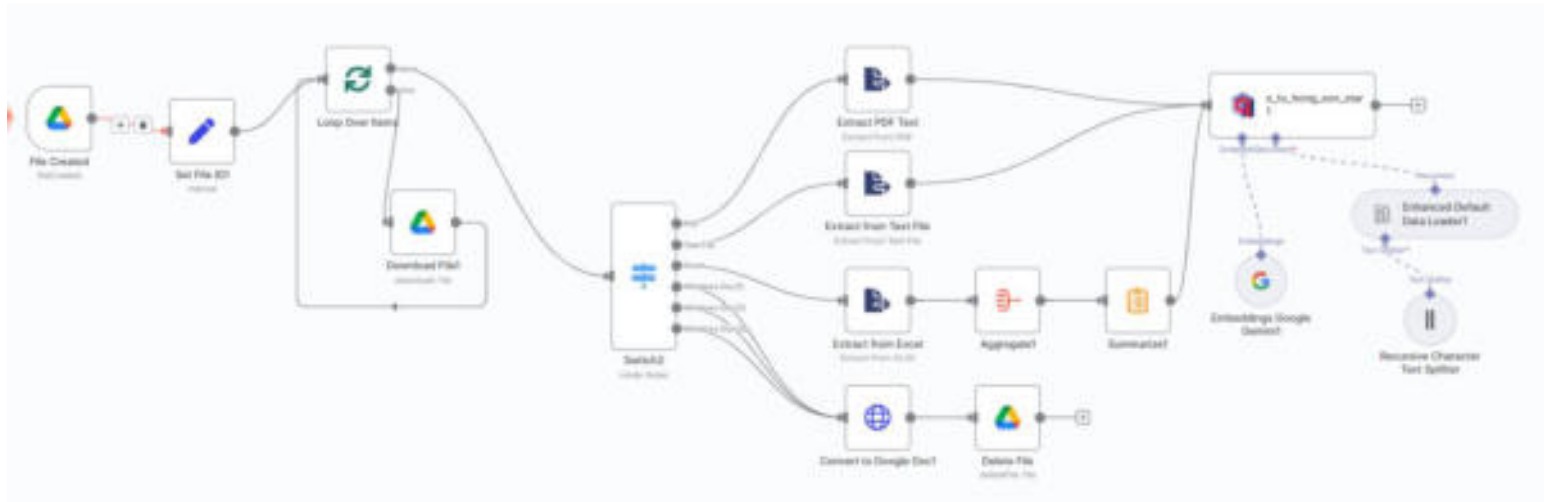
2.5.1. Sơ đồ tổng quan hệ thống



Hình 2.17. Sơ đồ tổng quan hệ thống

2.5.2. Sơ đồ tổng quan hệ thống AI Chatbot

2.5.2.1. Luồng xử lý và lưu trữ dữ liệu của file



Hình 2.18. Luồng xử lý khi có file được tạo của chatbot AI

Mục tiêu tổng thể: Luồng tự động theo dõi file mới trong Google Drive, trích xuất nội dung, chuyển đổi thành vector embedding và lưu trữ vào cơ sở dữ liệu Qdrant để hỗ trợ tìm kiếm ngữ nghĩa hoặc tích hợp vào hệ thống chatbot AI.

Quy trình chi tiết (từ trái sang phải):

1. File Created (fileCreated)

- **Loại node:** Trigger
- **Chức năng:** Theo dõi thư mục Drive/BackOffice/Backoffice_Contracts trên Google Drive → Khi có file mới được thêm vào, node này sẽ kích hoạt luồng.

2. Set File ID (Set File ID1)

- **Loại:** Set
- **Chức năng:** Gán ID của file vừa tải lên để chuẩn hóa dữ liệu đầu vào và truyền cho các bước tiếp theo.

3. Loop Over Items (Loop Over Items)

- **Loại:** Loop
- **Chức năng:** Duyệt qua từng file nếu nhiều file được tải lên cùng lúc, đảm bảo từng file được xử lý độc lập.

4. Download File (Download File1)

- **Loại:** Google Drive

Chức năng: Tải nội dung file từ Google Drive về hệ thống để xử lý tiếp.

5. Switch2

- **Loại:** Switch
- **Chức năng:** Phân loại file theo định dạng (.pdf, .txt, .xlsx, .doc, .docx) để xử lý phù hợp.

6. Các nhánh xử lý theo định dạng file:

6.1. Extract PDF Text

- **Loại:** Extract
- **Chức năng:** Trích xuất văn bản từ file PDF.

6.2. Extract from Text File

- **Loại:** Extract
- **Chức năng:** Đọc nội dung từ file .txt.

6.3. Extract from Excel

- **Loại:** Extract from XLSX
- **Chức năng:** Trích xuất nội dung từ file Excel (.xlsx).

6.4. Convert to Google Doc1

- **Loại:** Google Docs
- **Chức năng:** Chuyển .doc/.docx thành Google Doc để trích xuất nội dung dễ hơn.

6.5. Delete File

- **Loại:** Delete
- **Chức năng:** Xóa file trung gian sau khi đã xử lý để tránh chiếm dung lượng bộ nhớ.

7. Agregate1

- **Loại:** Merge / Combine
- **Chức năng:** Gộp tất cả nội dung từ các nhánh thành 1 đoạn văn bản lớn.

8. Summarize1

- **Loại:** AI/NLP
- **Chức năng:** Tóm tắt nội dung văn bản nếu quá dài hoặc cần chuẩn hóa trước khi nhúng (embedding).

9. Recursive Character Text Splitter

- **Loại:** Splitter
- **Chức năng:** Tách văn bản lớn thành các đoạn nhỏ hơn (theo số ký tự).

10. Embeddings Google Gemini1

- **Loại:** AI Embedding
- **Chức năng:** Biến các đoạn văn bản nhỏ thành vector embedding sử dụng mô hình Gemini.

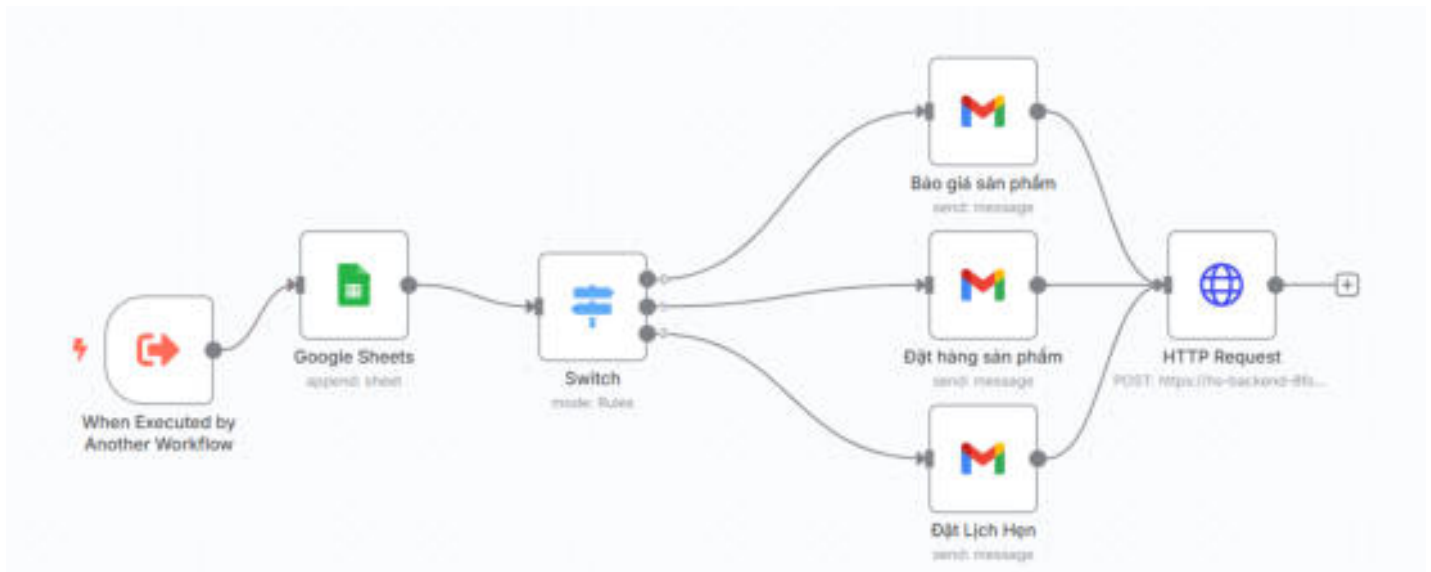
11. Qdrant Vector Store1

- **Loại:** Vector DB
- **Chức năng:**
 - Lưu các vector cùng văn bản gốc vào cơ sở dữ liệu vector Qdrant.
 - Phục vụ cho tìm kiếm ngữ nghĩa (Semantic Search) hoặc hệ thống RAG/chatbot.

12. Enhanced Default Data Loader1

- **Loại:** Data Loader
- **Chức năng:** Phụ trợ – cấu hình hoặc tải dữ liệu đầu vào nếu chưa được truyền từ bước trước.

2.5.2.2. Luồng xử lý thêm lịch hẹn



Hình 2.19. Luồng thêm lịch hẹn

Mục tiêu tổng thể: Tự động tiếp nhận dữ liệu từ một workflow khác, sau đó lưu trữ dữ liệu này vào một bảng tính Google Sheets và gửi email thông báo thông qua Gmail. Luồng này thường được sử dụng để ghi log, lưu kết quả xử lý, hoặc thông báo trạng thái đến người quản lý hoặc khách hàng.

Quy trình chi tiết (từ trái sang phải):

1. When Executed by Another Workflow

- Loại node: Trigger
- Chức năng: Kích hoạt luồng khi được gọi bởi một workflow khác.
- Ý nghĩa: Cho phép workflow hiện tại hoạt động như một luồng con (sub-workflow), có thể tái sử dụng nhiều lần từ các quy trình khác nhau.

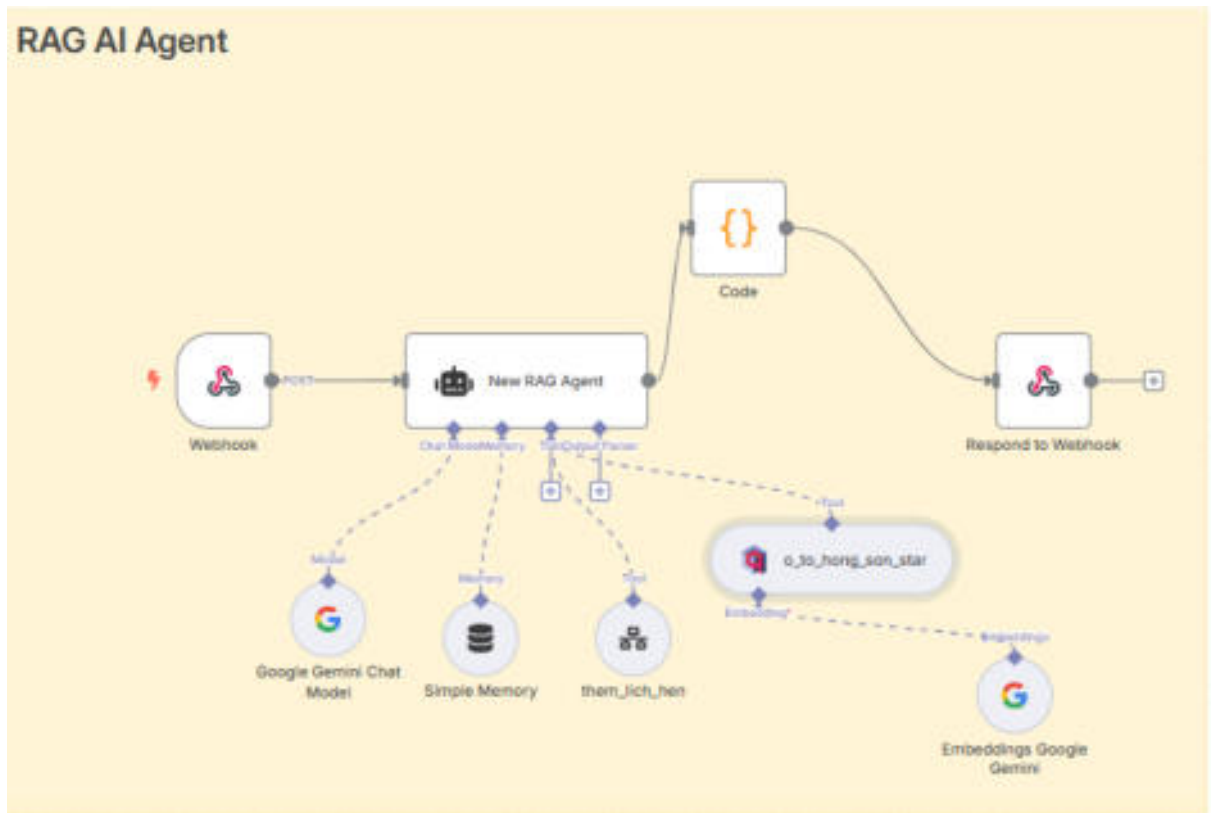
2. Google Sheets (append: sheet)

- Loại node: Google Sheets
- Chức năng: Ghi thêm một dòng dữ liệu mới vào một sheet cụ thể trong Google Sheets.
- Ý nghĩa: Lưu trữ dữ liệu đã nhận được từ workflow trước đó, thường dùng để tạo log, thống kê hoặc lưu phản hồi người dùng.

3. Gmail (send: message)

- Loại node: Gmail
- Chức năng: Gửi email đến một hoặc nhiều người nhận kèm theo nội dung được lấy từ dữ liệu đã xử lý.
- Ý nghĩa: Thông báo cho quản trị viên, khách hàng, hoặc các bên liên quan về kết quả xử lý hoặc dữ liệu vừa được lưu.

2.5.2.3. Luồng xử lý dữ liệu đầu vào và phản hồi về web



Hình 2.20. Luồng xử lý dữ liệu đầu vào và phản hồi về web của Chatbot AI

Mục tiêu tổng thể: Luồng tiếp nhận câu hỏi từ người dùng qua Webhook, truy vấn dữ liệu ngữ nghĩa từ Qdrant Vector Store, kết hợp với trí nhớ hội thoại, và sinh câu trả lời tự nhiên bằng mô hình Gemini Chat – nhằm tạo ra phản hồi tự nhiên và chính xác cho người dùng dựa trên ngữ cảnh đối thoại. thực hiện chức năng của hệ thống chatbot RAG. và thậm chí thực hiện hành động như thêm lịch hẹn thông qua các công cụ tích hợp.

Quy trình chi tiết (từ trái sang phải):

1. Webhook

- **Loại node:** Trigger
- **Chức năng:** Nhận yêu cầu từ bên ngoài qua HTTP POST(truy vấn, câu hỏi)., ví dụ từ frontend chatbot hoặc API client.
- **Dữ liệu đầu vào:** câu hỏi, sessionId để phân biệt người dùng hoặc phiên trò chuyện, các metadata khác (nếu có)

2. New RAG Agent

- **Loại node:** Agent/AI RAG (mô hình điều phối RAG)
- **Chức năng:** Là trung tâm xử lý chính trong toàn bộ hệ thống. điều phối của mô hình RAG:
 - Nhận câu hỏi từ webhook
 - Truy vấn dữ liệu từ Qdrant Vector Store.
 - Kết hợp ngữ cảnh từ Simple Memory
 - Gửi toàn bộ context cho mô hình Gemini Chat để sinh câu trả lời
- **Input:**
 - Chat Request: đến từ Webhook
 - Memory: lưu hội thoại (Session-based)
 - Tool/Document Parser: là Qdrant Vector Store
 - Model: Gemini Chat Model để sinh câu trả lời

3. Google Gemini Chat Model

- **Loại node:** AI Model
- Là mô hình sinh ngôn ngữ (LLM) từ Google.
- **Chức năng:**
 - Sinh câu trả lời bằng ngôn ngữ tự nhiên
 - Dựa trên thông tin đã truy vấn từ Qdrant + ngữ cảnh trong Simple Memory
 - Là bước xử lý ngôn ngữ tự nhiên (Natural Language Generation)

4. Simple Memory

- **Loại node:** Memory store
- **Chức năng:** Ghi nhớ ngữ cảnh hội thoại theo từng phiên (sessionId) (tức là nhớ các câu hỏi/trả lời trước đó). Đảm bảo câu trả lời phản ánh đúng ngữ cảnh lịch sử của người dùng, giúp cải thiện độ mạch lạc khi người dùng đặt các câu hỏi liên tiếp

5. o_to_hong_son_star (Qdrant Vector Store)

- **Loại node:** Cơ sở dữ liệu vector, Vector Database
- **Vai trò chính:** Đây là nơi lưu trữ tri thức dạng vector, được trích xuất từ các tài liệu liên quan đến sản phẩm, dịch vụ (ví dụ: thông tin về xe điện, chính sách bảo hành, v.v.).
- **Chức năng:** Khi người dùng đặt câu hỏi, hệ thống cần "tìm trong dữ liệu xem đoạn nào liên quan".

- Node này sẽ **tìm kiếm các đoạn văn bản có ý nghĩa gần giống với câu hỏi** (tức là có vector embedding tương đồng).
- Node **“Embeddings Google Gemini”** để so sánh độ tương đồng giữa vector của câu hỏi và vector trong cơ sở dữ liệu
- Đây là phần thực hiện **retrieval (truy xuất)** trong mô hình RAG

6. Embeddings Google Gemini

- **Loại node:** AI Embedding
- **Chức năng:** **Chuyển đổi câu hỏi thành một vector số học (embedding)** – thể hiện ý nghĩa của câu hỏi theo cách mà máy tính có thể hiểu và xử lý.
- **Vai trò trong RAG:** vector này sẽ được gửi đến node Qdrant để tìm các đoạn văn có embedding tương tự.
 - Lưu ý: **Không dùng để huấn luyện hay thêm dữ liệu mới**, chỉ để ánh xạ câu hỏi và tìm thông tin có sẵn

7. them_lich_hen (Thêm lịch hẹn)

- **Loại node:** Tool (Hành động)
- **Chức năng:**
 - Cho phép chatbot thực hiện hành động tự động đặt lịch hẹn dựa trên yêu cầu của người dùng
 - Ví dụ: Khi người dùng yêu cầu “Đặt lịch hẹn bảo dưỡng xe vào chiều mai”, node này sẽ ghi nhận và lưu thông tin vào hệ thống

8. Respond to Webhook

- **Loại node:** Webhook Response
- **Chức năng:** Trả câu trả lời sinh ra từ Gemini về cho client gọi Webhook ban đầu.

CHƯƠNG 3: TRIỂN KHAI VÀ CHẠY THỬ NGHIỆM

3.1. Môi trường triển khai

3.1.1. Quản lý mã nguồn

Git là một **hệ thống quản lý phiên bản phân tán (Distributed Version Control System)**, giúp theo dõi và kiểm soát sự thay đổi trong mã nguồn trong quá trình phát triển phần mềm. Với Git, các lập trình viên có thể làm việc nhóm hiệu quả, quản lý lịch sử chỉnh sửa, tạo nhánh, gộp nhánh và khôi phục mã nguồn một cách dễ dàng. Git hoạt động cục bộ, nhanh chóng và là công cụ tiêu chuẩn trong các quy trình phát triển phần mềm hiện đại.

GitHub là một nền tảng lưu trữ mã nguồn và cộng tác trực tuyến, được xây dựng dựa trên Git. Ngoài chức năng lưu trữ và quản lý mã nguồn, GitHub còn cung cấp các tính năng hỗ trợ làm việc nhóm như quản lý issue, pull request, CI/CD, và theo dõi tiến độ dự án. GitHub đóng vai trò là nơi chia sẻ, đóng góp mã nguồn mở và triển khai các dự án phần mềm trong cộng đồng lập trình viên toàn cầu.

3.1.2. Môi trường cài đặt

- Hệ thống được xây dựng trên nền tảng client-server, sử dụng:
 - NestJS cho back-end
 - ReactJS , Nextjs cho front-end
- Thông tin môi trường cài đặt:

Bảng 3.1. Môi trường cài đặt

Thành phần	Công nghệ sử dụng	Phiên bản
Back-end	NestJS (Node.js)	Node v18.x, NestJS v9.x
Front-end	ReactJS , Next.js	v18.x
Cơ sở dữ liệu	PostgreSQL	v15.x
Quản lý gói	npm	v9.x
Trình soạn thảo	Visual Studio Code	Mới nhất

- Triển khai thực tế:

Bảng 3.2. Môi trường triển khai thực tế

Thành phần	Công cụ triển khai	Ghi chú
Back-end	Render	Deploy API NestJS
Front-end	Netlify	Deploy Next.js + ReactJS
Cơ sở dữ liệu	Amazon Web Service	Cơ sở dữ liệu của hệ thống
Quản lý mã nguồn	Git + GitHub	Lưu trữ và CI/CD

3.2. Hướng dẫn sử dụng

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

3.2.1. Backend

1. Lấy mã nguồn:

- Sử dụng lệnh git clone để tải toàn bộ mã nguồn về máy. VD:

Git clone: <https://github.com/OpenverseHQ/hs-backend.git>

2. Cài đặt các gói phụ thuộc:

- Cài đặt các gói phụ thuộc tại thư mục gốc của dự án bằng lệnh yarn install,

3. Thiết lập cơ sở dữ liệu.

- Khởi động dịch vụ cơ sở dữ liệu (ví dụ: PostgreSQL).
- Tạo một cơ sở dữ liệu mới tương ứng với tên bạn sẽ cấu hình

4. Cấu hình biến môi trường:

- Mở file .env trong thư mục dự án, chỉnh sửa các thông tin kết nối cơ sở dữ liệu cho đúng với thiết lập ở bước trên, ví dụ:

DATABASE_URL=postgresql://postgres:123456@localhost:5432/postgres

5. Tạo bảng dữ liệu:

Chạy lệnh sau để thực hiện migration và tạo các bảng trong cơ sở dữ liệu:

yarn migration:run

6. Khởi động ứng dụng: **yarn dev**

3.3. Frontend

1. Cài đặt yarn: nếu chưa cài đặt yarn, hãy chạy lệnh sau:

npm install -g yarn

2. Clone dự án

Git clone: <https://github.com/OpenverseHQ/hongson-fe.git>

3. Chạy dự án: **yarn start**

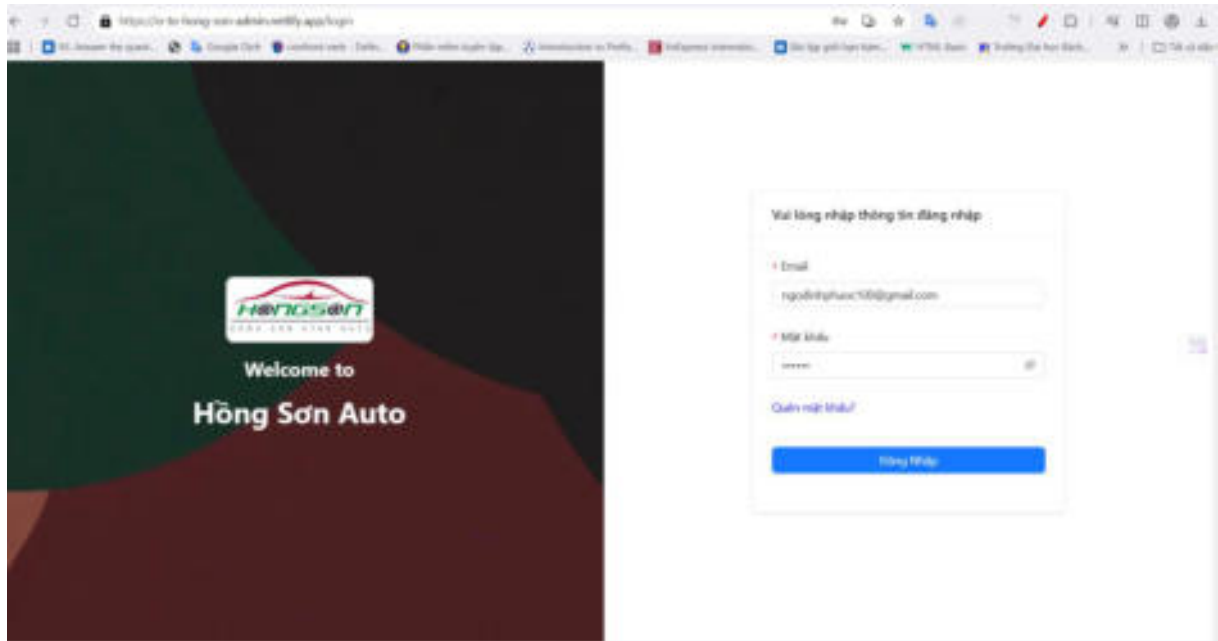
4. Build dự án cho production: **yarn build**

3.4. Kết quả demo

3.4.1. Giao diện website admin

3.4.1.1. Đăng nhập

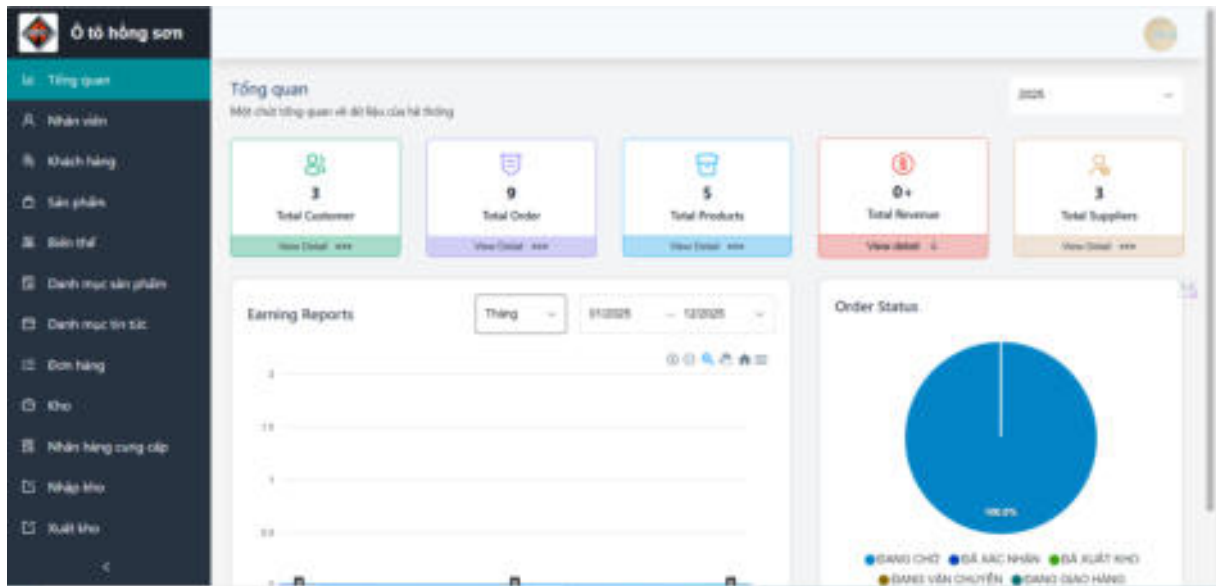
Người dùng đăng nhập vào hệ thống bằng tài khoản hợp lệ để thực hiện chức năng của mình.



Hình 3.1 Giao diện đăng nhập admin

3.4.1.2. Trang chủ

Sau khi người dùng đăng nhập thành công, sẽ điều hướng đến trang chủ (trang tổng quan). Người Admin và nhân viên bán hàng xem được được thống kê tổng số lượng khách hàng ,sản phẩm bán, đơn hàng ,doanh thu theo năm. Biểu đồ trực quan trạng doanh thu với lợi nhuận theo khoảng tháng trong năm, khoảng các ngày trong cùng 1 tháng, tổng số lượng trạng thái của các đơn hàng theo các năm.



Hình 3.2. Tổng quan

3.4.1.3. Quản lý nhân viên

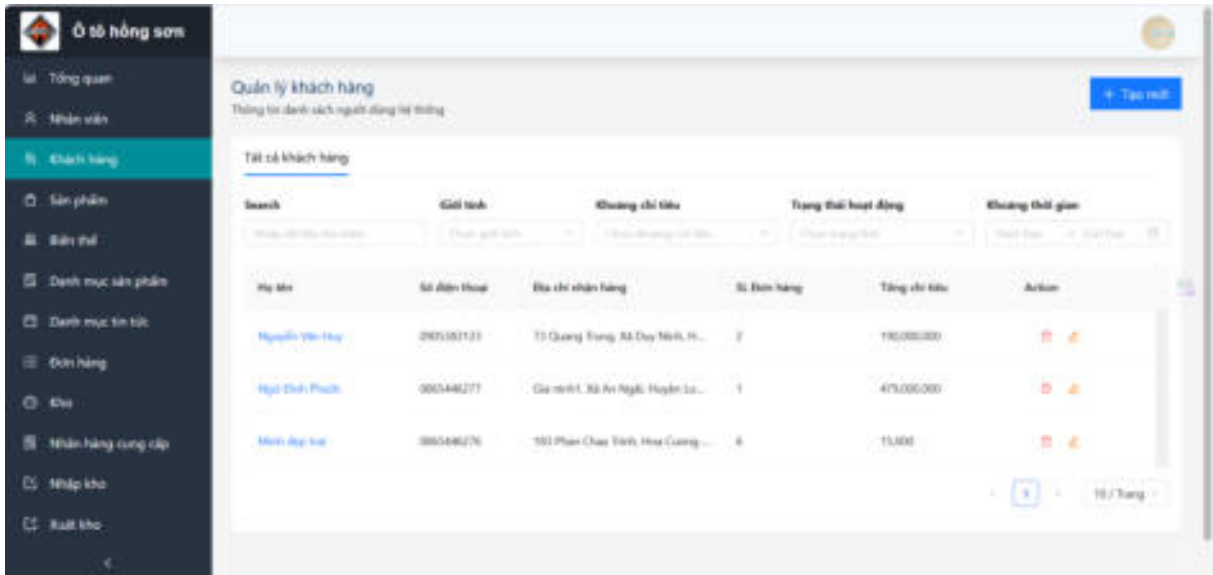
Người Admin và nhân viên bán hàng xem danh sách nhân viên. Và thực hiện chức năng xem chi tiết, thêm, xóa, sửa, lọc, import, export danh sách nhân viên với excel.

Họ tên	Email	Số điện thoại	Ngày sinh	Vai trò	Action
nguyễn thiệp	nguyenthiep123@ex...	090546275	21/10/2001	Nhân viên bán hàng	[Edit] [Delete]
phạm văn hải hải	phamvanhai@example.c...	090546275	10/11/2001	Nhân viên giao hàng	[Edit] [Delete]
phạm văn hải hải	phamvanhai1234567@ex...	090546275	11/12/2001	Nhân viên bán hàng	[Edit] [Delete]

Hình 3.3 Màn hình nhân viên

3.4.1.4. Quản lý khách hàng

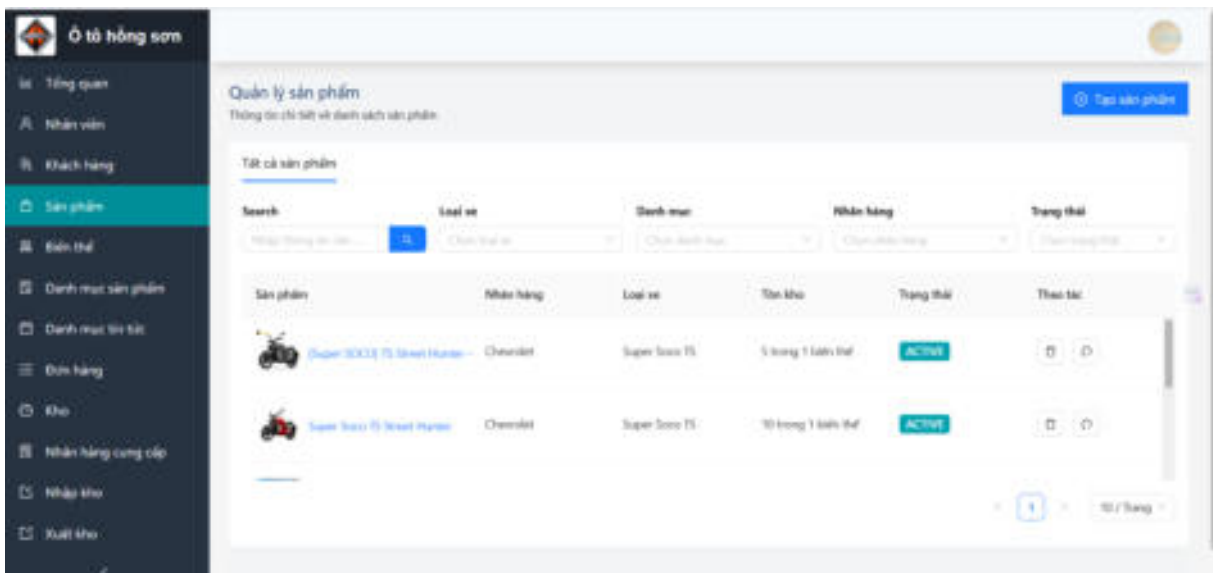
Người Admin và nhân viên bán hàng xem danh sách khách hàng. Và thực hiện chức năng xem chi tiết, thêm, xóa, sửa, lọc . Khi xem chi tiết khách hàng, người dùng có thể thêm, sửa địa chỉ.



Hình 3.4. Giao diện khách hàng

3.4.1.5. Quản lý sản phẩm

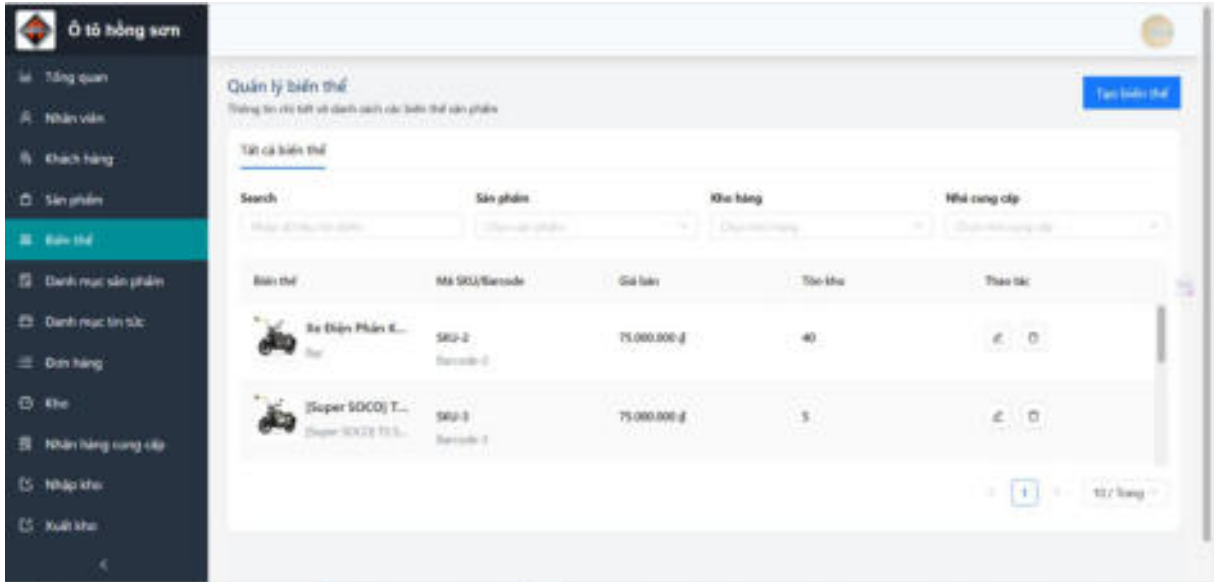
Người Admin xem danh sách sản phẩm. Và thực hiện chức năng xem chi tiết, thêm, xóa, sửa, lọc.



Hình 3.5. Giao diện quản lý sản phẩm

3.4.1.6. Quản lý biển thể

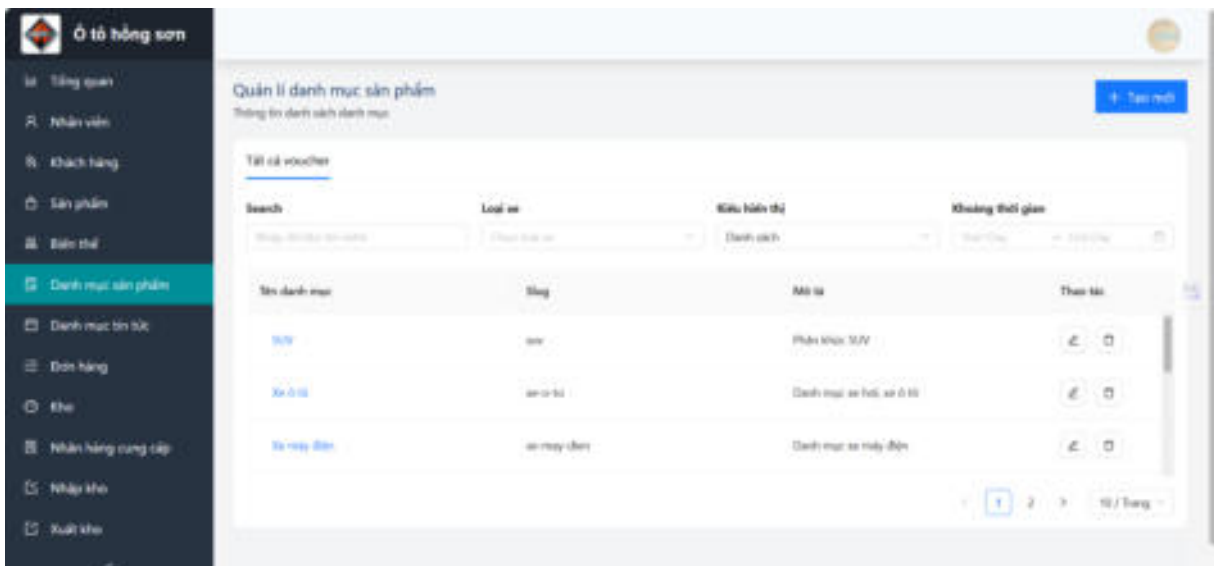
Người admin xem danh sách biển thể. Và thực hiện chức năng xem chi tiết, thêm, xóa, sửa, lọc.



Hình 3.6. Giao diện biển thể

3.4.1.7. Danh mục sản phẩm

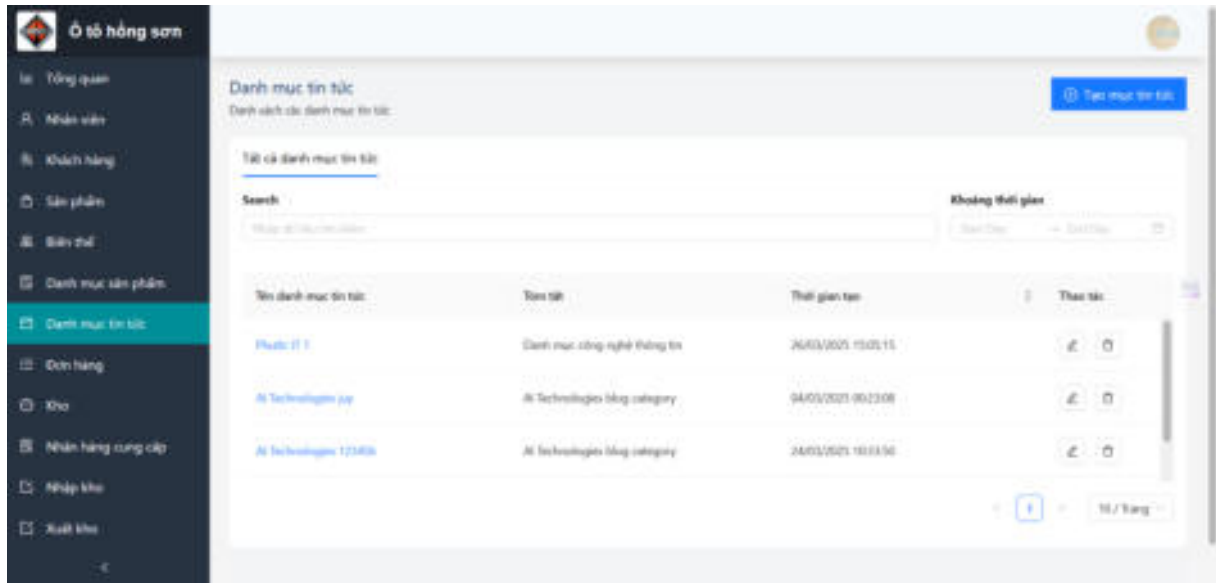
Người admin xem danh sách danh mục sản phẩm. Và thực hiện chức năng xem chi tiết, thêm, xóa, sửa, lọc.



Hình 3.7. Giao diện danh mục sản phẩm

3.4.1.8. Danh mục tin tức

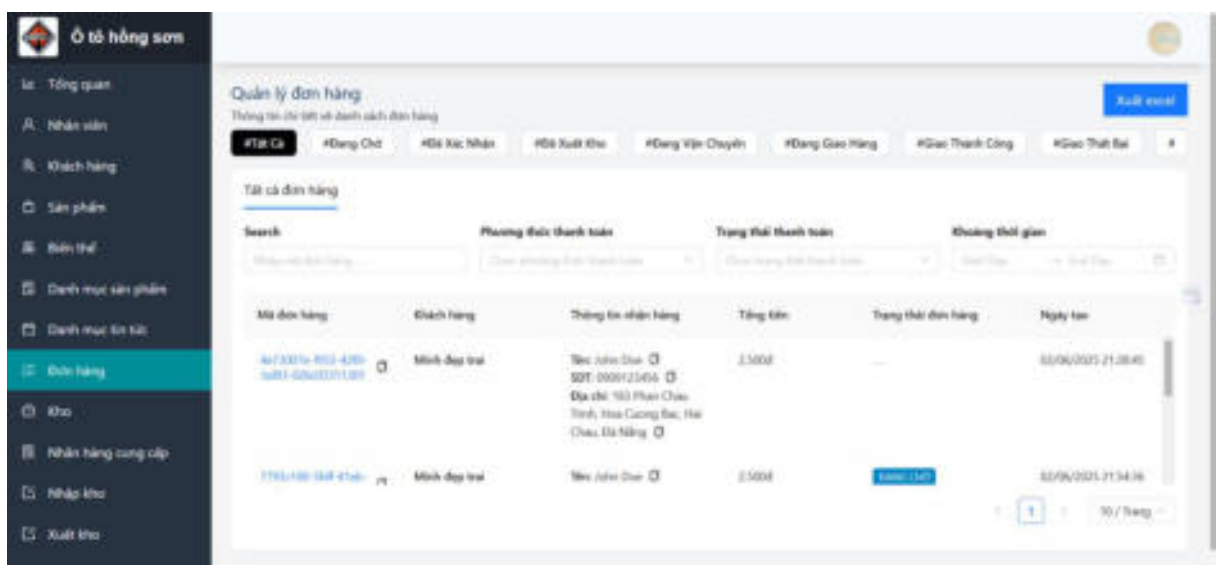
Admin tra cứu danh sách danh mục tin tức. Và thực hiện chức năng xem chi tiết, thêm, xóa, sửa danh mục tin tức



Hình 3.8. Giao diện danh mục tin tức

3.4.1.9. Quản lý đơn hàng

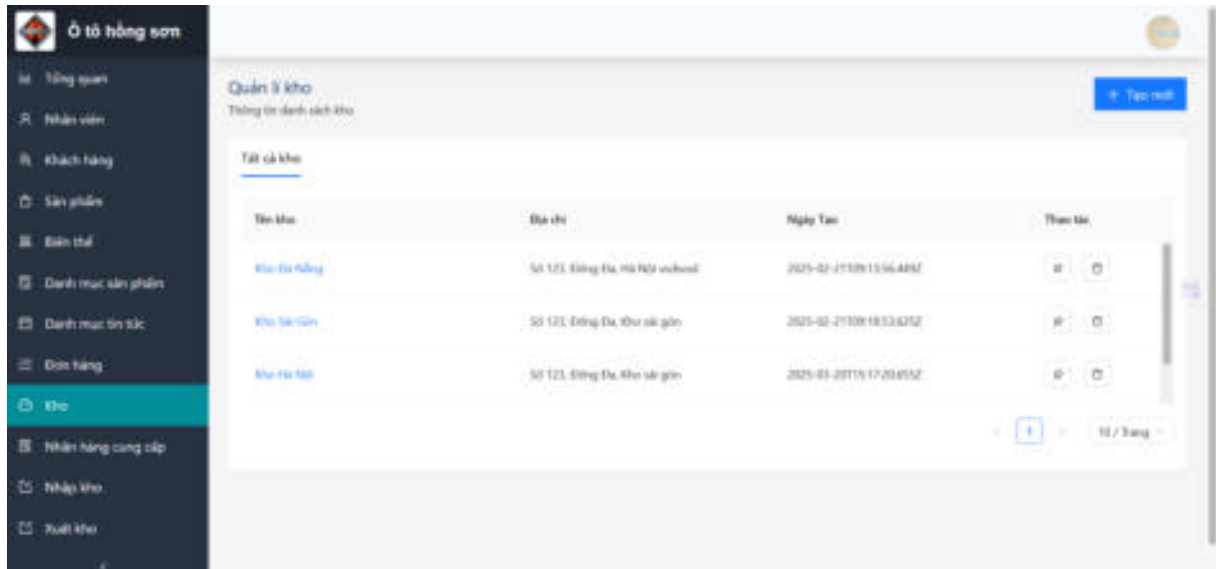
Admin tra cứu thông tin đơn hàng, tìm kiếm, xem chi tiết trạng thái đơn hàng, cập nhật trạng thái đơn hàng bao gồm các trạng thái: Confirmed (Đã xác nhận đơn hàng), Exported (Đã xuất kho thành công), Delivering (Đang vận chuyển), Shipping (Đang giao hàng), Failed_Delivery (Giao hàng thất bại), Delivered (Đã giao hàng thành công)



Hình 3.9. Quản lý đơn hàng

3.4.1.10. Quản lý kho

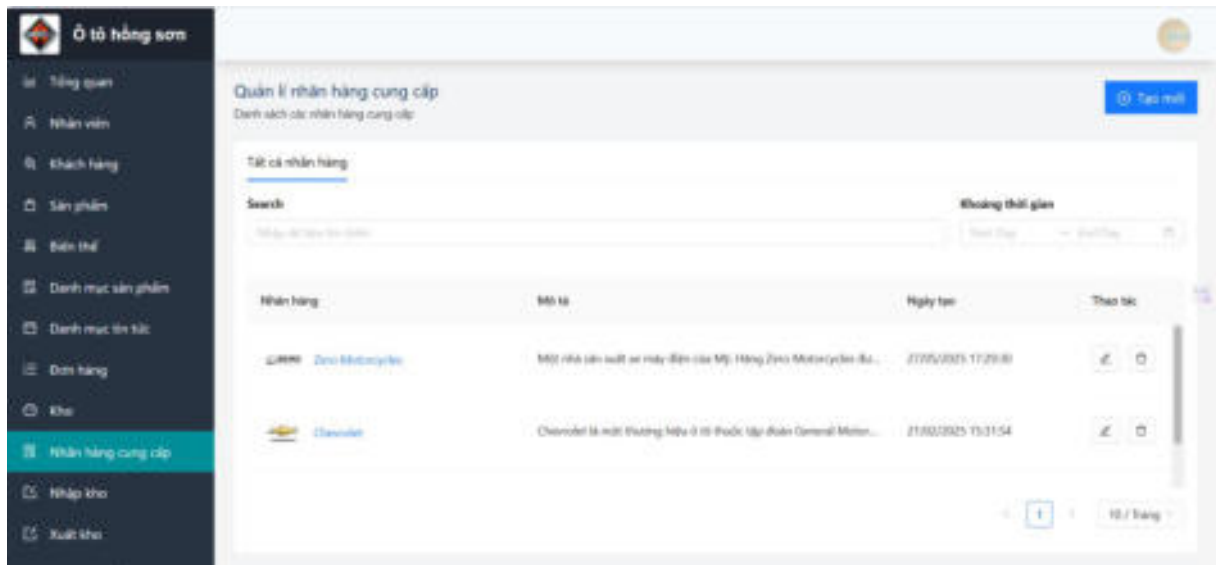
Admin có thể thêm, sửa, xóa, đặt kho làm mặc định



Hình 3.10. Admin quản lí kho

3.4.1.11. Nhân hàng cung cấp

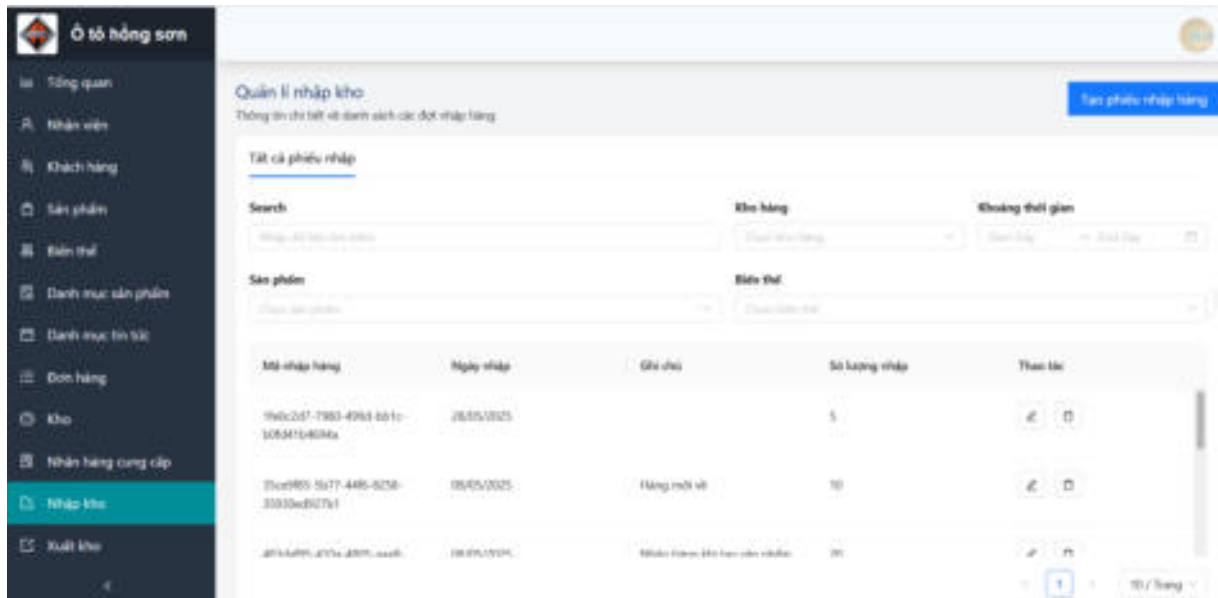
Admin tra cứu danh sách, thêm, sửa, xóa thông tin các nhà cung cấp



Hình 3.11 Giao diện nhà cung cấp

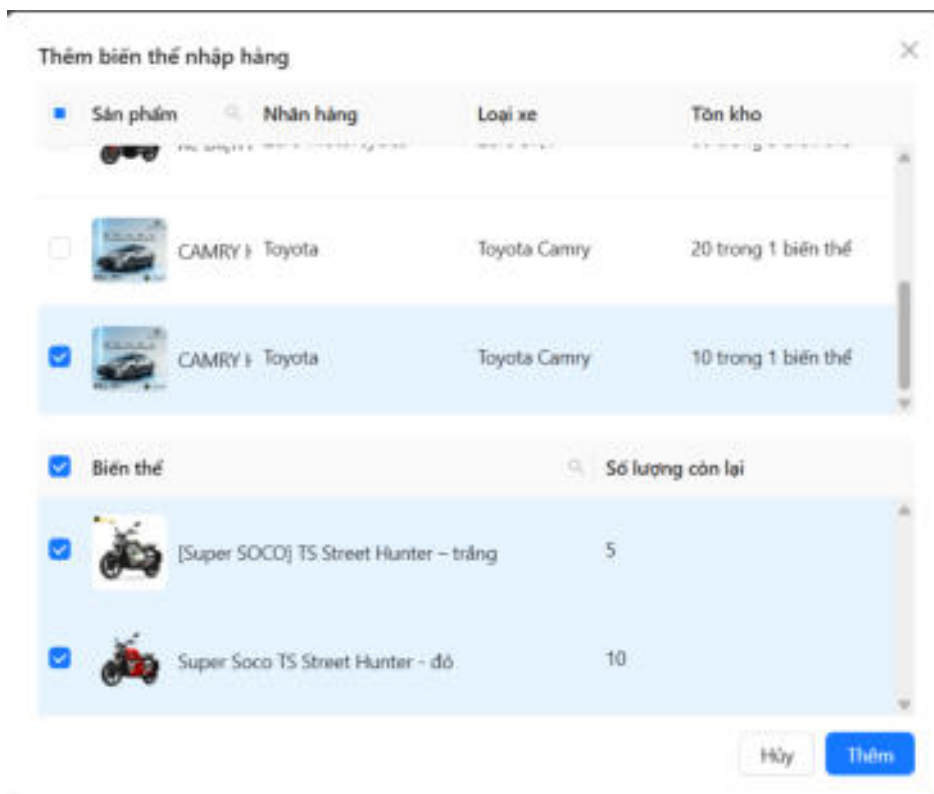
3.4.1.12. Nhập kho

Admin có thể tra cứu thông tin các phiếu nhập kho thông qua mã phiếu nhập, ghi chú phiếu, tên người nhập, kho hàng, khoảng thời gian nhập, tên sản phẩm và biến thể sản phẩm



Hình 3.12. Giao diện nhập kho

Khi tạo phiếu nhập kho, admin có thể cùng chọn 1 hoặc nhiều biến thể cùng nhập chung trong cùng 1 phiếu



Hình 3.13. Admin chọn biến thể để nhập vào kho

Sau khi thêm các biến thể cần nhập, admin có thể nhập vào nhiều thông tin nhập kho như giá nhập, số lượng nhập, kho nhập, và hỗ trợ nhập tại nhiều kho khác nhau, mỗi dữ liệu chi tiết nhập kho đại diện cho một lô hàng, admin có thể đặt tên lô theo ý muốn để thuận tiện cho việc tra cứu sau này. Sau khi thêm phiếu thành công, admin có thể sửa lại phiếu khi có sai sót phát sinh.

Tạo phiếu nhập hàng [X]

Ghi chú

Nhập ghi chú

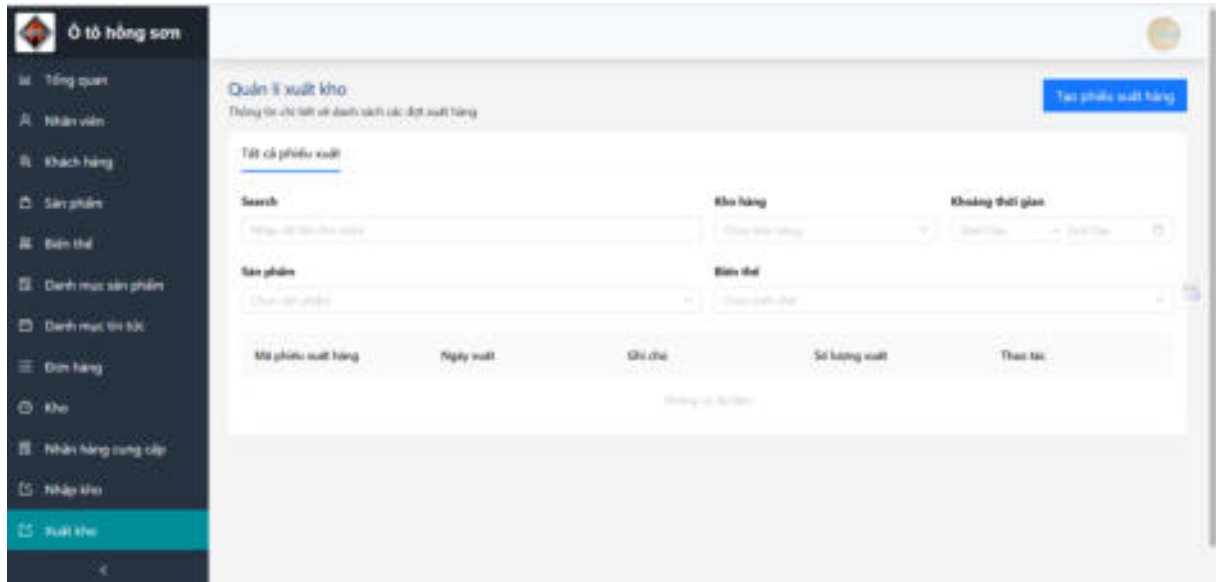
* Biến thể nhập [X] [+]

<input type="checkbox"/> Biến thể	Giá nhập	Thông tin chi tiết nhập kho (kho, số lượng, tên lô)	
<input type="checkbox"/> CAMRY HEV TOP - bạc	11	Kho Đà Nẵng	2
		Lô A	[+]
		Kho Sài Gòn	2
		Lô B	[+]
		[+]	[+]

Hình 3.14. Admin nhập biến thể vào nhiều kho khác nhau

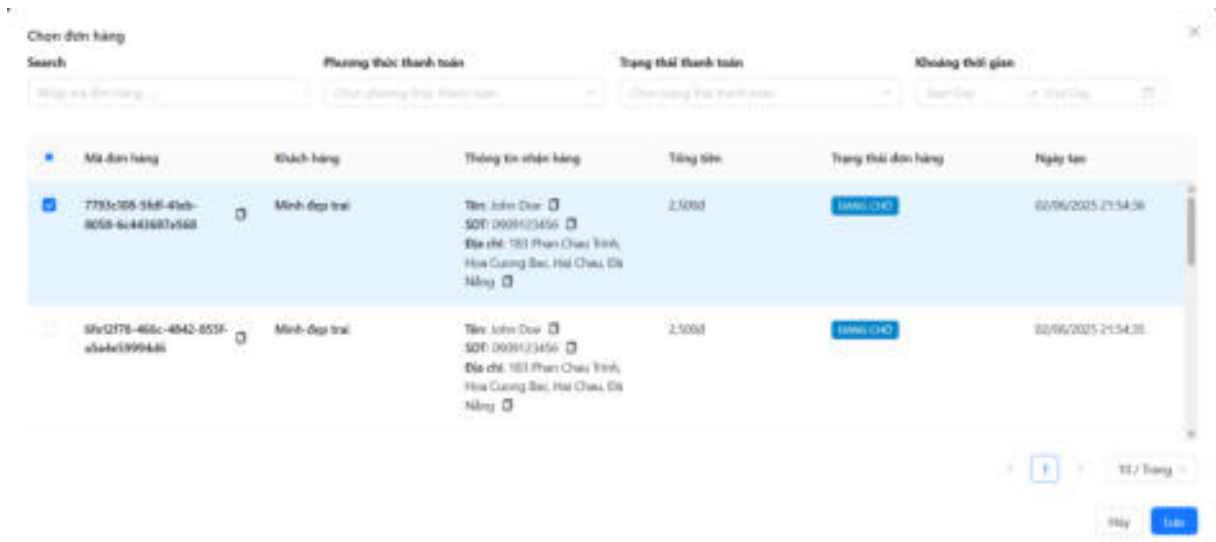
3.4.1.13. Xuất kho

Admin có thể tra cứu thông tin các phiếu xuất kho thông qua mã phiếu xuất, ghi chú phiếu, tên người xuất, kho hàng, khoảng thời gian xuất, tên sản phẩm và biển thể sản phẩm



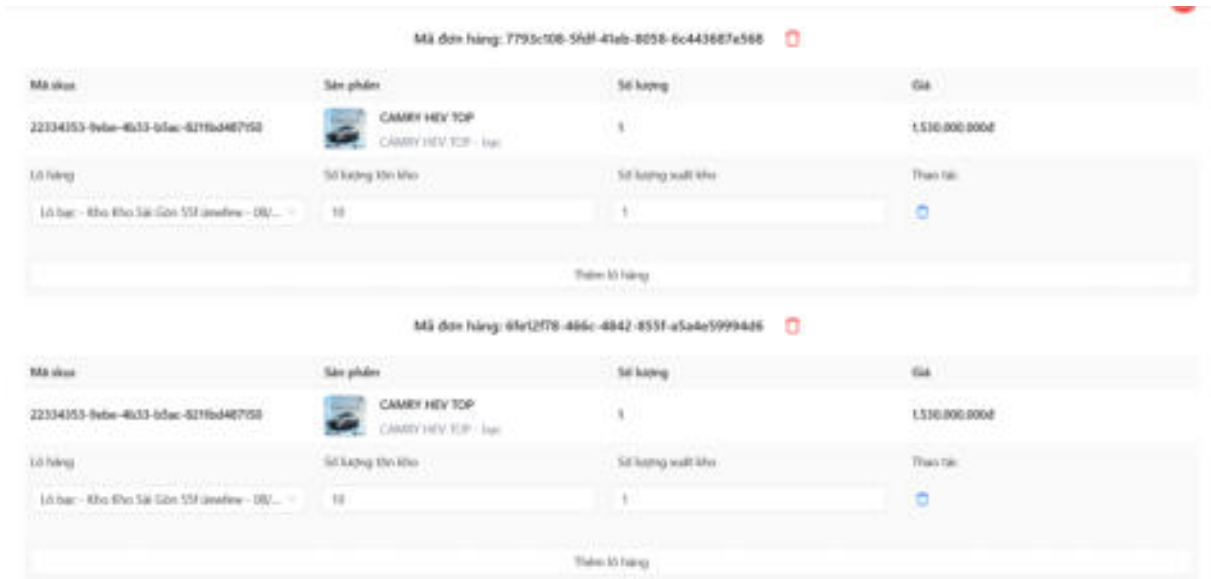
Hình 3.15. Giao diện xuất kho

Admin có thể chọn 1 hoặc nhiều order để cùng xuất chung 1 phiếu xuất



Hình 3.16. Admin chọn đơn hàng để xuất kho

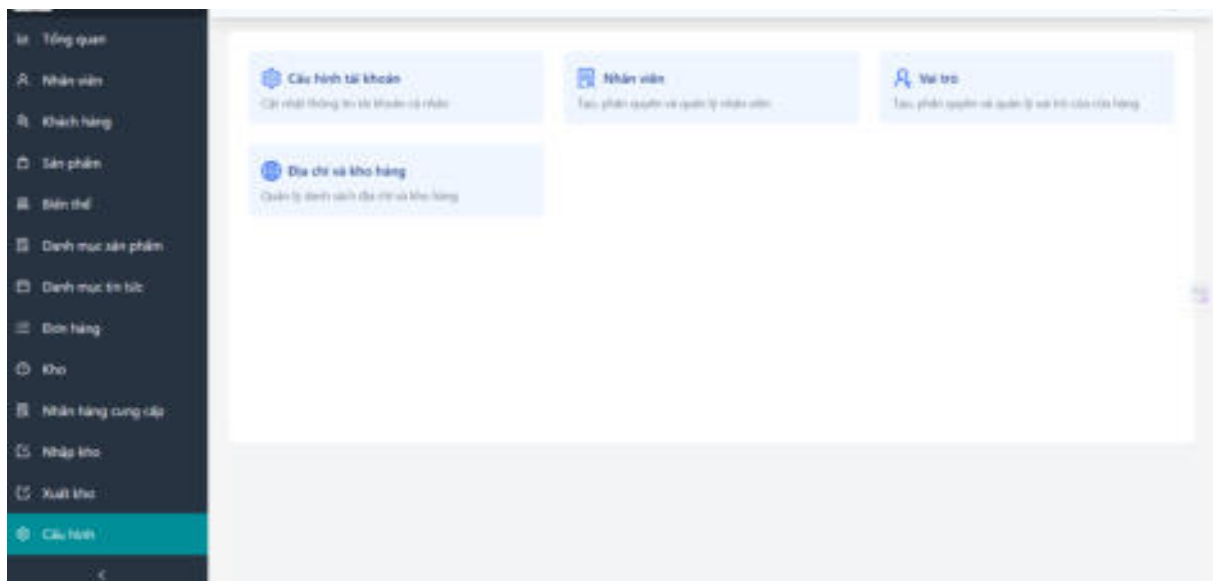
Sau đó admin có thể chọn các lô hàng cần xuất tương ứng với các sản phẩm có trong từng đơn hàng và nhập thông tin xuất kho cần thiết. Sau khi tạo phiếu xuất thành công, có thể chỉnh sửa lại nếu có sai sót phát sinh



Hình 3.17. Admin nhập thông tin xuất kho cho mỗi đơn hàng

3.4.1.14. Cấu hình

Chỉ admin có quyền trong cấu hình. Đối với role, người admin có thể thêm xóa, cập nhật role và cập nhật permission trong role. Đối với permission thì admin cũng thực hiện thêm, xóa, sửa.



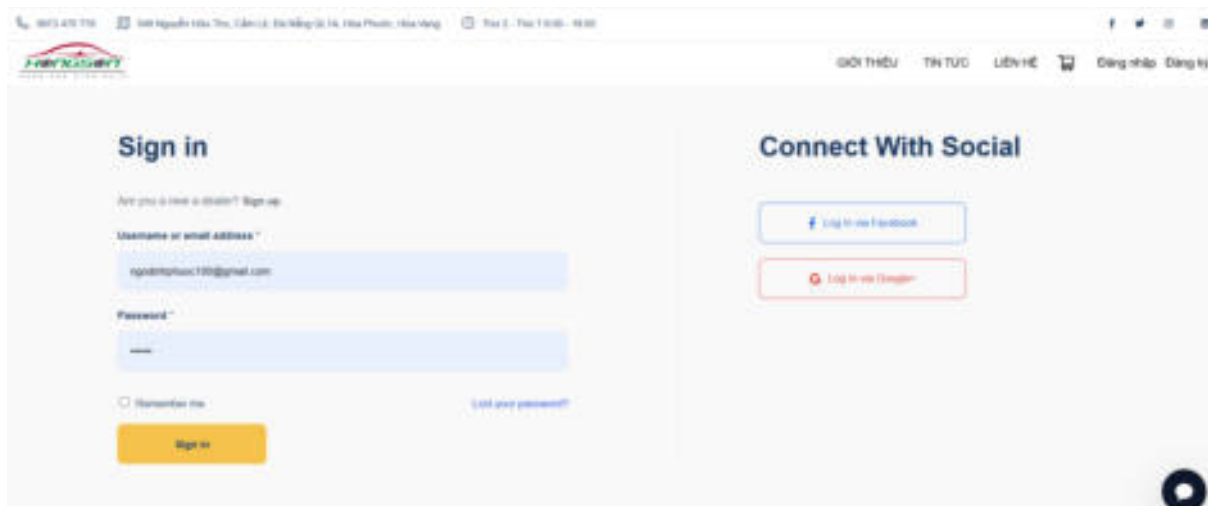
Hình 3.18. Giao diện cấu hình

3.4.2. Giao diện website user

3.4.2.1. Trang đăng kí & đăng nhập

Khách hàng nếu chưa có tài khoản thì cần thực hiện đăng ký. Sau khi đăng ký thành công với thông tin hợp lệ, hệ thống sẽ gửi mã kích hoạt gồm **4 chữ số** đến email đã đăng ký. Khách hàng cần nhập mã kích hoạt này trong **vòng 5 phút** để hoàn tất quá trình xác thực. Sau khi kích hoạt thành công, hệ thống sẽ tự động chuyển hướng đến trang đăng nhập. Khi đăng nhập thành công, khách hàng sẽ được chuyển đến **trang chủ dành cho người dùng**.

Trong trường hợp quên mật khẩu, khách hàng chỉ cần nhập địa chỉ email đã đăng ký. Hệ thống sẽ gửi mã xác thực gồm **4 chữ số** đến email đó. Sau khi nhập mã xác thực, khách hàng sẽ tiến hành đặt lại mật khẩu mới để hoàn tất quá trình khôi phục tài khoản.



Hình 3.19. Giao diện đăng kí và đăng nhập.

3.4.2.2. Trang home

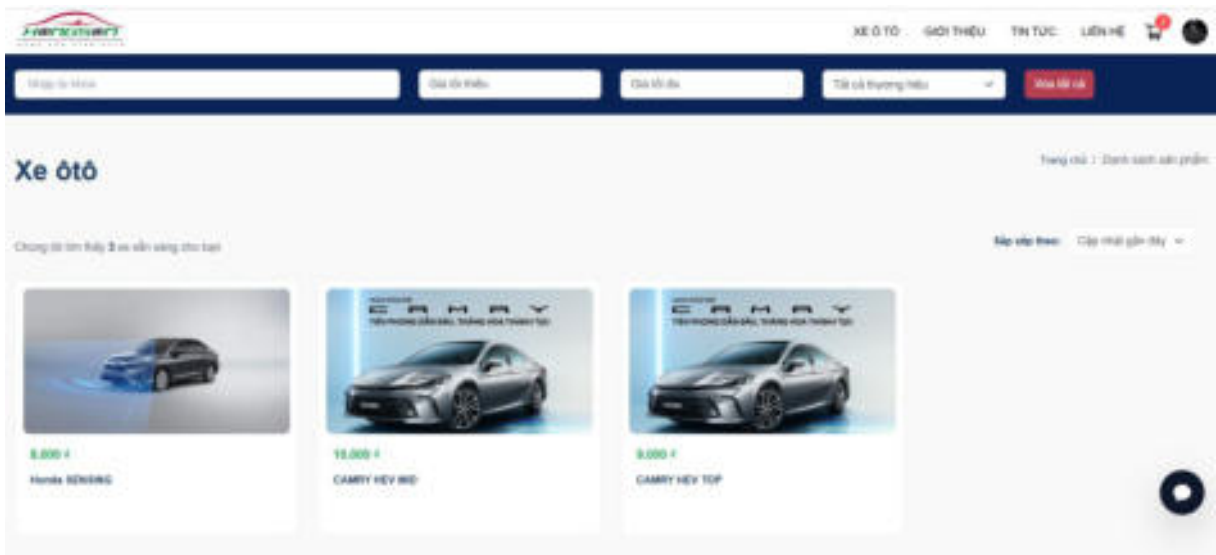
Tại **giao diện trang chủ**, khách hàng có thể dễ dàng truy cập các thành phần chính như: thanh menu, slider trình chiếu các dòng xe, danh sách sản phẩm bán chạy và nổi bật, các bản tin tức mới, thông tin đối tác và phần giới thiệu tổng quan về cửa hàng.



Hình 3.20. Giao diện trang chủ

3.4.2.3. Trang danh sách sản phẩm

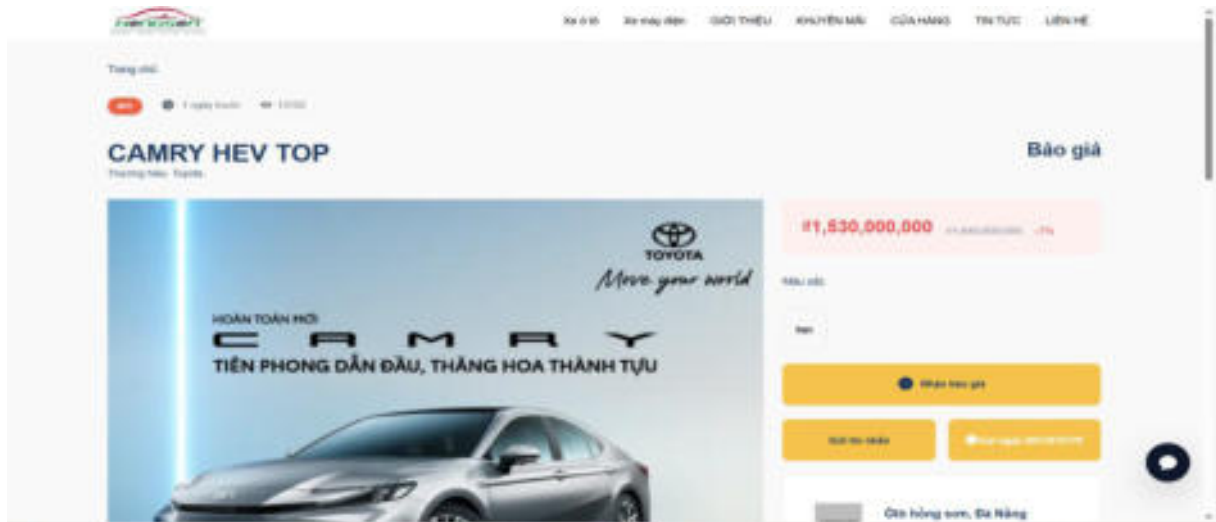
Khách hàng có thể xem danh sách xe ô tô hay xe máy điện. Sử dụng bộ lọc , tìm kiếm , sắp xếp sản phẩm.



Hình 3.21. Giao diện danh sách sản phẩm

3.4.2.4. Trang chi tiết sản phẩm

Khách hàng có thể xem chi tiết thông tin sản phẩm mà người dùng đã chọn. Ở trang này họ có thể nhận báo giá, xem hình ảnh xe theo màu sắc, xem thông tin mô tả để nắm được đầy đủ hơn về xe. Và có thêm vào giỏ hàng và mua ngay.



Hình 3.22 Giao diện chi tiết sản phẩm

3.4.2.5. Trang tin tức

Khách hàng có thể xem danh sách tin tức và đồng thời xem chi tiết tin tức để có thêm thông tin, kiến thức liên quan đến xe ô tô, xe máy điện.



Hình 3.23. Giao diện tin tức

3.4.2.6. Trang giỏ hàng

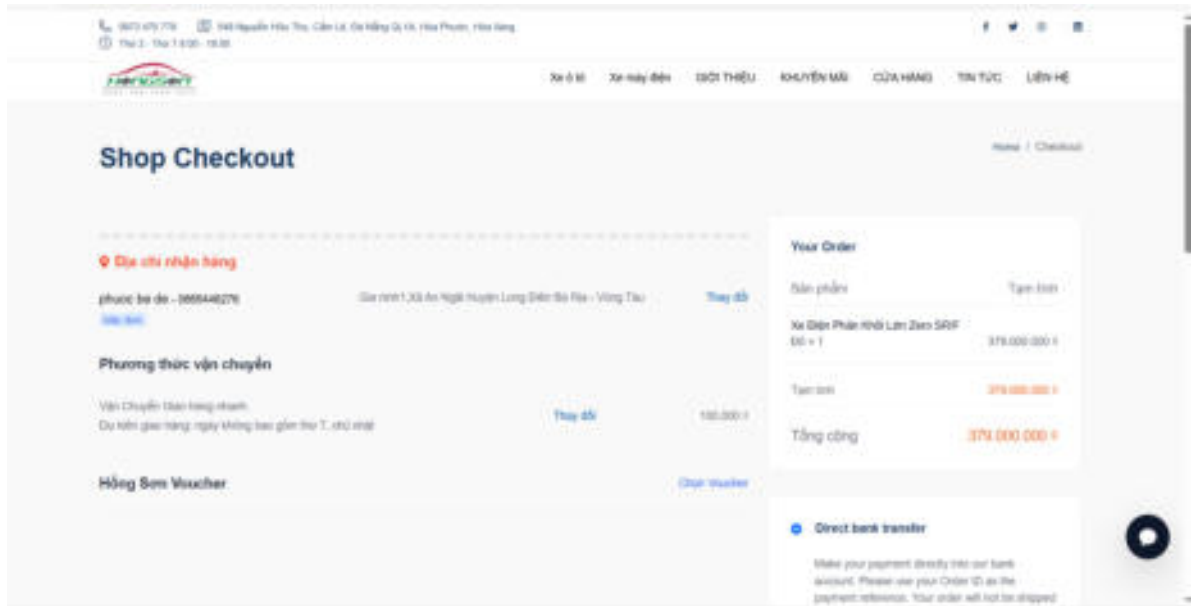
Khách hàng có thể xem danh sách biến thể sản phẩm đã thêm vào giỏ hàng. Khách hàng có thể xóa, cập nhật số lượng mua. Để ấn nút Proceed to Checkout thì cần chọn biến thể sản phẩm cần mua để chuyển hướng sang trang thanh toán.



Hình 3.24. Giao diện giỏ hàng

3.4.2.7. Trang thanh toán

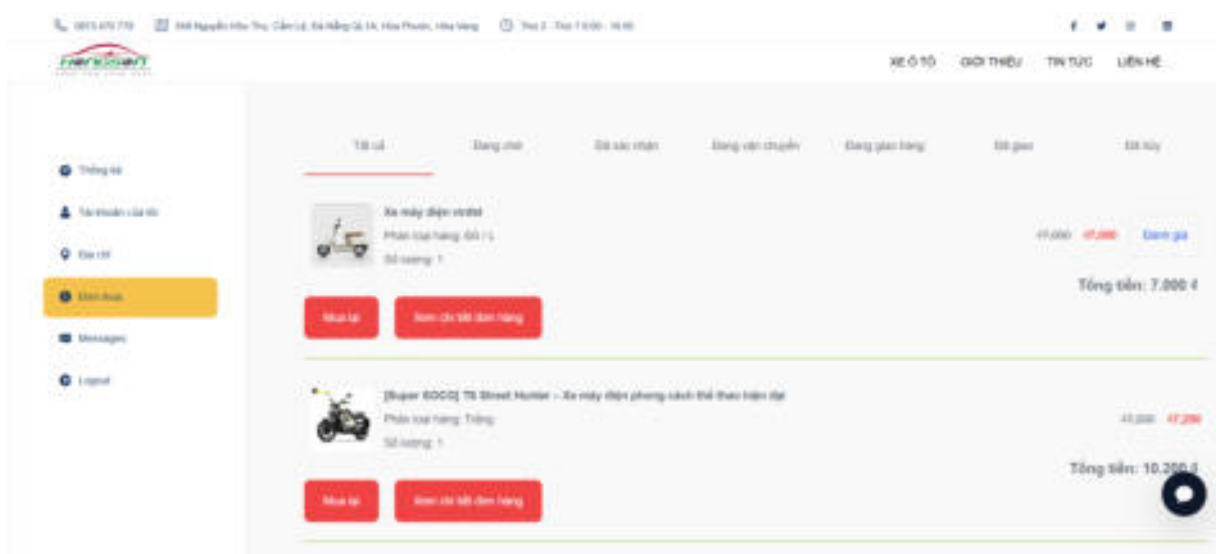
Khách hàng sẽ mua biến thể sản phẩm, thanh toán bằng 3 phương thức thanh toán cùng với địa chỉ mua hàng, voucher giảm giá và phương thức vận chuyển. Khi ấn mua hàng, nếu mua thành công thì sẽ chuyển sang lịch sử đơn hàng với trạng thái pending của đơn hàng đó nếu chọn thanh toán bằng tiền mặt, nếu thanh toán bằng phương thức chuyển khoản thì sẽ đưa khách hàng đến trang thanh toán quét mã chuyển tiền. nếu chuyển thành công sẽ điều hướng về lịch sử đơn hàng cùng với trạng thái. Nếu thanh toán thất bại thì đơn sẽ bị hủy quay về trang lịch sử đơn hàng.



Hình 3.25. Giao diện thanh toán

3.4.2.8. Trang đơn mua

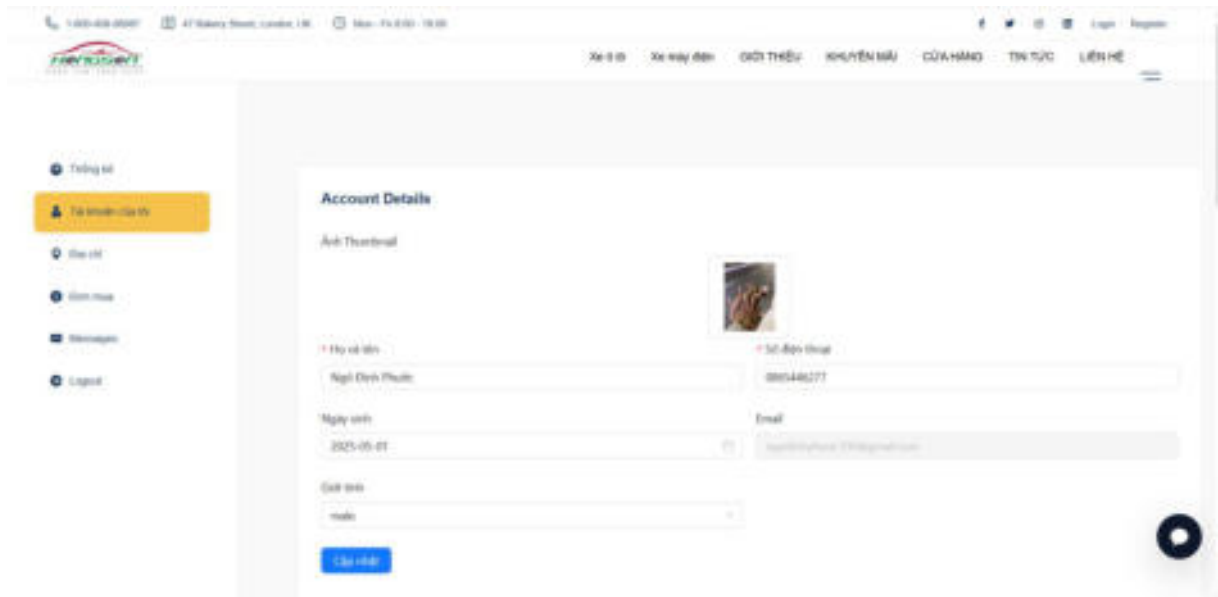
Khách hàng xem lịch sử đơn hàng với 7 trạng thái. Khách hàng còn có thể hủy đơn khi đơn hàng ở trạng thái pending, mua lại, xem chi tiết đơn hàng đã mua.



Hình 3.26. Giao diện lịch sử đơn hàng

3.4.2.9. Tài khoản của tôi

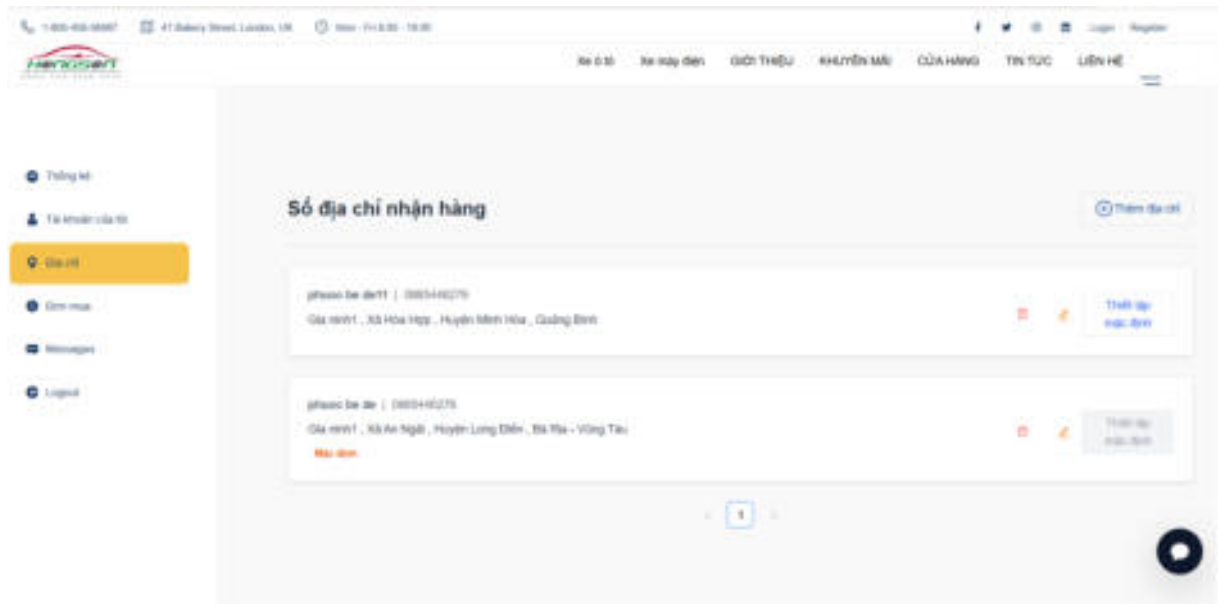
Khách hàng xem thông tin cá nhân bản thân, có thể cập nhật lại thông tin cũng như mật khẩu.



Hình 3.27. Giao diện tài khoản khách hàng

3.4.2.10. Trang địa chỉ

Khách hàng quản lý địa chỉ của họ. Thực hiện chức năng thêm, xóa, sửa, thiết lập mặc định cho địa chỉ giao hàng.



Hình 3.28. Giao diện quản lý địa chỉ

3.4.2.11. Chat bot AI

Khách hàng có thể hỗ trợ tìm kiếm sản phẩm, tìm kiếm thông tin về doanh nghiệp, nhận thông tin rõ ràng chính xác.



Hình 3.29. Giao diện chatbot AI

CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết quả đạt được

Sau quá trình nghiên cứu và xây dựng, hệ thống đã đạt được những kết quả nổi bật như sau:

- **Hoàn thiện hệ thống quản lý kho và bán hàng** với khả năng theo dõi chi tiết từng biến thể sản phẩm: số lượng, trạng thái, giúp giảm thiểu sai sót trong kiểm kê và tăng độ chính xác trong báo cáo tồn kho.
- **Triển khai thành công hệ thống phân quyền người dùng**, hỗ trợ nhiều vai trò khác nhau như quản trị viên, nhân viên kho, nhân viên bán hàng và nhân viên giao hàng.
- **Xây dựng giao diện quản lý sản phẩm đa biến thể**: Cho phép cấu hình thuộc tính sản phẩm như màu sắc, kích thước, giá trị cụ thể cho từng biến thể một cách linh hoạt và dễ dàng.
- **Tích hợp thanh toán điện tử thực tế**: Hệ thống đã tích hợp thành công các phương thức thanh toán như ZaloPay và PayOS.
- **Hệ thống thống kê đầy đủ và trực quan**: Doanh thu, lợi nhuận, đơn hàng, người dùng và số lượng sản phẩm đều được hiển thị rõ ràng qua các biểu đồ và bảng dữ liệu.
- **Tích hợp AI Chatbot hỗ trợ khách hàng**, có khả năng phản hồi các câu hỏi thường gặp, hỗ trợ tìm kiếm sản phẩm, thông tin khuyến mãi mà không cần đến sự can thiệp thủ công từ nhân viên.

- **Triển khai thực tế hệ thống trên nền tảng web:** Với backend được triển khai trên Render, frontend trên Netlify và cơ sở dữ liệu PostgreSQL trên Aws, hỗ trợ lưu trữ hình ảnh qua Cloudinary.

Những kết quả trên đã cho thấy tính khả thi, hiệu quả và tiềm năng ứng dụng của hệ thống vào thực tế kinh doanh, đặc biệt là với các hộ kinh doanh nhỏ và cá nhân trong lĩnh vực xe ô tô và xe máy điện.

4.2. Hạn chế và vấn đề chưa giải quyết

Bên cạnh những thành tựu đạt được, hệ thống vẫn còn một số hạn chế và vấn đề cần tiếp tục hoàn thiện:

- **Chưa có phiên bản ứng dụng di động (mobile app),** gây hạn chế trong việc sử dụng trên thiết bị cầm tay, đặc biệt là với nhân viên bán hàng thường xuyên di chuyển.
- **Độ chính xác** trong việc trả lời câu hỏi của **AI Chatbot** vẫn còn hạn chế, đặc biệt là đối với những câu hỏi phức tạp, không rõ ngữ cảnh hoặc yêu cầu kiến thức chuyên sâu.
- **Chưa tích hợp hệ thống giao hàng** với bên thứ ba như Giao Hàng Nhanh, ViettelPost..., khiến quy trình vận chuyển vẫn còn thủ công.
- **Giao diện người dùng chưa tối ưu hóa cho mọi phân khúc,** đặc biệt là với người lớn tuổi hoặc người không rành công nghệ.

Việc nhận diện các hạn chế này là bước quan trọng giúp định hướng rõ ràng cho việc phát triển hệ thống trong tương lai.

4.3. Hướng phát triển

Trong tương lai, hệ thống sẽ tiếp tục được nâng cấp và mở rộng với các định hướng cụ thể như sau:

- **Mở rộng phạm vi ứng dụng:** Không chỉ giới hạn trong ngành xe hơi và xe máy điện, hệ thống có thể được điều chỉnh để áp dụng cho các ngành hàng có đặc điểm sản phẩm đa biến thể như thời trang, nội thất, mỹ phẩm, v.v.
- **Phát triển ứng dụng di động (mobile app):** Tối ưu hóa trải nghiệm người dùng trên các thiết bị di động, giúp quản lý bán hàng mọi lúc mọi nơi.
- **Nâng cấp AI Chatbot thông minh hơn:** Sử dụng các mô hình học sâu (Deep Learning) và giải pháp RAG (Retrieval-Augmented Generation) để hỗ trợ khách hàng hiệu quả hơn, xử lý ngôn ngữ tự nhiên linh hoạt và gần với con người.
- **Tự động hóa quy trình làm việc với n8n:** Tích hợp công cụ workflow automation giúp tiết kiệm thời gian trong các tác vụ như xử lý đơn hàng, cập

nhật tồn kho, chăm sóc khách hàng định kỳ và một số tính năng có thể tự động hóa khác.

- **Áp dụng AI phân tích và báo cáo theo thời gian thực:** Phân tích dữ liệu và báo cáo dữ liệu thống kê tự động nhằm hỗ trợ nhà quản lý ra quyết định kịp thời.
- **Ứng dụng websocket:** Thiết lập cơ chế phản hồi tức thì cho khách hàng, đồng thời cho phép người dùng trò chuyện trực tiếp với quản trị viên nhằm nâng cao trải nghiệm tương tác và hỗ trợ kịp thời.
- **Tích hợp với bên vận chuyển:** Trong tương lai, hệ thống có thể bổ sung các API để liên kết với đơn vị giao hàng như Giao Hàng Nhanh, Giao Hàng Tiết Kiệm, ViettelPost...
- **Mở rộng các phương thức thanh toán:** Thêm các ví điện tử khác như Momo để phù hợp với nhu cầu người dùng đa dạng.

TÀI LIỆU THAM KHẢO

TÀI LIỆU TIẾNG VIỆT

- [1] Tài liệu Postgresql, <https://www.postgresql.org/docs>, 10/04/2025
- [2] Tài liệu thiết kế RESTful APIs, <https://techtalk.vn/thiet-ke-restful-apis.html>, 13/04/2025
- [3] Tài liệu deploy Netlify <https://blog.thanhnamnguyen.dev/deploy-project-len-netlify>, 15/05/2025
- [4] Tài liệu RAG <https://viblo.asia/p/chatgpt-series-5-tim-hieu-ve-retrieval-augmented-generation-rag-Ny0VGRd7LPA>, 10/06/2025

TÀI LIỆU TIẾNG ANH

- [5] Tài liệu Nestjs <https://docs.nestjs.com/>, 20/04/2025
- [6] Tài liệu Javascript, <https://vi.wikipedia.org/wiki/JavaScript>, 28/04/2025
- [7] Tài liệu ReactJs, <https://reactjs.org>, 20/04/2025
- [8] Tài liệu NextJs, <https://nextjs.org>, 26/04/2025
- [9] Tài liệu n8n, <https://n8n.io>, 30/04/2025
- [10] Tài liệu deploy Render <https://render.com/docs/free>, 15/05/2025