

This image shows a full page of a document template designed for handwriting practice or general note-taking. It consists of approximately 30 evenly spaced horizontal dotted lines across the entire width of the page. The background is plain white, and there are no margins, headers, footers, or other markings present.

Vĩnh Long, ngày tháng năm
Giảng viên hướng dẫn
(Ký tên và ghi rõ họ tên)

[illegible]

Vĩnh Long, ngày tháng năm
Thành viên hội đồng
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Trước hết, tôi cũng xin gửi lời cảm ơn đến Thầy Phạm Minh Dương, người đã trực tiếp hướng dẫn, giúp đỡ, góp ý và giải đáp những thắc mắc của tôi trong suốt quá trình thực hiện để tôi có thể hoàn thiện đề tài.

Tôi cũng xin chân thành cảm ơn các Thầy, Cô trong Khoa Công nghệ thông tin đã giảng dạy và trang bị cho tôi những kiến thức và kỹ năng cơ bản để có nền tảng thực hiện đề tài.

Cuối cùng, tôi xin gửi đến quý Thầy, Cô lời chúc sức khỏe, hạnh phúc và thành công trong sự nghiệp giảng dạy.

Tôi xin chân thành cảm ơn!

Vĩnh Long, ngày tháng 12 năm 2025

Sinh viên thực hiện

Nguyễn Phước Hiệp

MỤC LỤC

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH	10
MỞ ĐẦU.....	11
1. Lý do chọn đề tài	11
2. Mục đích nghiên cứu	11
3. Đối tượng nghiên cứu.....	12
4. Phạm vi nghiên cứu	12
CHƯƠNG 1: TỔNG QUAN.....	13
1.1 Mô tả bài toán.....	13
1.2 Công cụ và cách thức thực hiện	13
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT	15
2.1 Tìm hiểu về Restful API.....	15
2.1.1 Tổng quan về Restful API.....	15
2.1.2 Các phương thức sử dụng trong Restful API.....	15
2.1.3 Các mã trạng thái phản hồi	16
2.1.4 Ưu và nhược điểm.....	16
2.2 Kiến trúc MVC kết hợp Service Layer	17
2.2.1 Tổng quan về kiến trúc MVC kết hợp Service Layer.....	17
2.2.2 Ưu điểm của mô hình.....	18
2.2.3 Nhược điểm của mô hình.....	18
2.3 Tìm hiểu về ReactJS và công cụ Vite	19
2.3.1 Giới thiệu tổng quan về ReactJS.....	19
2.3.2 Các đặc điểm nổi bật của ReactJS	19
2.3.3 React Hooks	19
2.3.4 Công cụ ViteJS	20
2.4 Tìm hiểu về Tailwind CSS	20

2.4.1	Giới thiệu tổng quan	20
2.4.2	Lịch sử hình thành và phát triển	20
2.4.3	Ưu điểm của Tailwind CSS	21
2.4.4	Nhược điểm và hạn chế.....	21
2.5	Tìm hiểu về MySQL.....	22
2.5.1	Giới thiệu tổng quan	22
2.5.2	Lịch sử hình thành.....	22
2.5.3	Ưu điểm của MySQL	22
2.5.4	Nhược điểm và hạn chế.....	22
2.6	Tìm hiểu về Spring Boot	23
2.6.1	Giới thiệu tổng quan và Lịch sử.....	23
2.6.2	Các tính năng nổi bật và Ưu điểm	23
2.6.3	Ưu và nhược điểm của Spring Boot.....	23
2.7	Tìm hiểu về cổng thanh toán VnPay	24
2.7.1	Giới thiệu và Vai trò trong hệ thống.....	24
2.7.2	Kiến trúc tích hợp và luồng xử lý dữ liệu	24
2.8	Tìm hiểu về OpenAI API	26
2.8.1	Giới thiệu tổng quan	26
2.8.2	Các mô hình và Cơ chế hoạt động	26
2.9	Các nghiệp vụ liên quan đến đề tài	27
2.9.1	Nghiệp vụ quản trị hệ thống và phân quyền người dùng.....	27
2.9.2	Nghiệp vụ quản lý danh mục và hàng hóa.....	27
2.9.3	Nghiệp vụ bán hàng và xử lý đơn hàng trực tuyến.....	27
2.9.4	Nghiệp vụ chăm sóc khách hàng, hỗ trợ và thanh toán trực tuyến.....	28
2.10	Các công trình nghiên cứu liên quan.....	28
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU		31

3.1.	Mô tả hệ thống.....	31
3.2.	Xác định các yêu cầu chức năng của hệ thống.....	32
3.2.1	Yêu cầu chức năng dành cho khách hàng, người xem.....	32
3.2.2	Yêu cầu chức năng dành cho người quản trị	33
3.2.3	Yêu cầu phi chức năng của hệ thống	33
3.3.	Phân tích các tác nhân của hệ thống.....	33
3.3.1	Các chức năng và quyền hạn của các tác nhân trong hệ thống.....	34
3.3.2	Sơ đồ Use case tổng quát của hệ thống.....	34
3.3.3	Sơ đồ Use case đặt hàng	35
3.3.4	Sơ đồ Use case quản lí đơn hàng	35
3.3.5	Sơ đồ Use case quản lí sản phẩm.....	36
3.3.6	Sơ đồ Use case quản lí giỏ hàng	36
3.4.	Mô hình dữ liệu hệ thống	37
3.5.	Kiến trúc hệ thống và luồng xử lý Chatbot AI.....	45
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU		48
4.1.	Thử nghiệm API với Postman.....	48
4.2.	Giao diện người dùng.....	50
4.3.	Giao diện quản trị hệ thống	53
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....		56
5.1.	Kết quả đạt được	56
5.2.	Hướng phát triển.....	56
DANH MỤC TÀI LIỆU THAM KHẢO		57
PHỤ LỤC.....		58

DANH MỤC HÌNH ẢNH

<i>Hình 2. 1 Mô tả RESTful API (nguồn Internet)</i>	15
<i>Hình 2. 2 Mô tả kiến trúc MVC kết hợp Service Layer (nguồn Internet)</i>	17
<i>Hình 2. 3 Mô tả luồng xử lý dữ liệu của VNPay (nguồn VNPay)</i>	25
<i>Hình 3. 1 Use case tổng quát của hệ thống</i>	35
<i>Hình 3. 2 Use case đặt hàng của hệ thống</i>	35
<i>Hình 3. 3 Use case quản lý đơn hàng của hệ thống</i>	36
<i>Hình 3. 4 Use case quản lý sản phẩm của hệ thống</i>	36
<i>Hình 3. 5 Use case quản lý giỏ hàng của hệ thống</i>	36
<i>Hình 3. 6 Mô hình dữ liệu của hệ thống</i>	37
<i>Hình 3. 7 Sơ đồ mô tả kiến trúc của hệ thống</i>	46
<i>Hình 3. 8 Mô tả luồng hoạt động của chatbot AI với website</i>	47
<i>Hình 4. 1 Thử nghiệm API đăng nhập với Postman</i>	48
<i>Hình 4. 2 Thử nghiệm API đăng ký với Postman</i>	48
<i>Hình 4. 3 Thử nghiệm API lấy danh sách sản phẩm với Postman</i>	49
<i>Hình 4. 4 Thử nghiệm API đặt hàng với Postman</i>	49
<i>Hình 4. 5 Giao diện đăng nhập</i>	50
<i>Hình 4. 6 Giao diện đăng ký</i>	50
<i>Hình 4. 7 Giao diện trang chủ</i>	51
<i>Hình 4. 8 Giao diện trang giỏ hàng</i>	51
<i>Hình 4. 9 Giao diện trang đặt hàng</i>	52
<i>Hình 4. 10 Giao diện trang đơn hàng của tôi</i>	52
<i>Hình 4. 11 Giao diện trò chuyện với chatbot AI tư vấn</i>	53
<i>Hình 4. 12 Giao diện trang Admin Dashboard</i>	53
<i>Hình 4. 13 Giao diện trang quản lý đơn hàng</i>	54

<i>Hình 4. 14 Giao diện trang quản lí người dùng</i>	<i>54</i>
<i>Hình 4. 15 Giao diện trang quản lí sản phẩm</i>	<i>55</i>
<i>Hình 4. 16 Giao diện trang quản lí mã giảm giá</i>	<i>55</i>

DANH MỤC BẢNG BIỂU

<i>Bảng 2. 1 Các mô hình OPENAI API phổ biến</i>	<i>26</i>
<i>Bảng 3. 1 Mô tả các tác nhân tham gia vào hệ thống.....</i>	<i>34</i>
<i>Bảng 3. 2 Mô tả bảng USERS</i>	<i>37</i>
<i>Bảng 3. 3 Mô tả bảng CATEGORIES</i>	<i>38</i>
<i>Bảng 3. 4 Mô tả bảng PRODUCTS</i>	<i>38</i>
<i>Bảng 3. 5 Mô tả bảng COUPONS</i>	<i>39</i>
<i>Bảng 3. 6 Mô tả bảng PAYMENT_METHODS</i>	<i>41</i>
<i>Bảng 3. 7 Mô tả bảng ORDERS.....</i>	<i>41</i>
<i>Bảng 3. 8 Mô tả bảng ORDER_DETAILS</i>	<i>42</i>
<i>Bảng 3. 9 Mô tả bảng PAYMENTS</i>	<i>43</i>
<i>Bảng 3. 10 Mô tả bảng REVIEWS</i>	<i>43</i>
<i>Bảng 3. 11 Mô tả bảng FLASH_SALE.....</i>	<i>44</i>
<i>Bảng 3. 12 Mô tả bảng FLASH_SALE_ITEMS</i>	<i>44</i>

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Đồ án tập trung nghiên cứu và xây dựng một hệ thống website thương mại điện tử bán nội thất, đồng thời tích hợp trí tuệ nhân tạo để nâng cao trải nghiệm của khách hàng. Vấn đề cốt lõi được đặt ra là làm thế nào để hỗ trợ người dùng tìm kiếm, tra cứu thông tin sản phẩm thuận tiện và nhận được sự tư vấn, hỗ trợ kịp thời tương tự như mua sắm trực tiếp.

Để giải quyết vấn đề này, đề tài tiếp cận theo hướng xây dựng ứng dụng web hiện đại với kiến trúc tách biệt giữa Backend và Frontend. Cụ thể, hệ thống sử dụng Spring Boot để xây dựng các API xử lý nghiệp vụ và MySQL để quản trị cơ sở dữ liệu. Giao diện người dùng được phát triển bằng ReactJS kết hợp Tailwind CSS mang lại trải nghiệm trực quan và thân thiện. Điểm đột phá trong phương pháp giải quyết vấn đề là việc tích hợp OpenAI API để phát triển chatbot AI thông minh, cùng với cổng thanh toán VNPAY để thực hiện quy trình giao dịch trực tuyến.

Kết quả đạt được của đồ án là một website bán nội thất hoàn chỉnh với đầy đủ các chức năng quản lý cho Admin (quản lý sản phẩm, đơn hàng, mã giảm giá, thống kê) và chức năng mua sắm cho Khách hàng (đặt hàng, thanh toán trực tuyến). Quan trọng nhất là tính năng chatbot AI, cho phép tự động tư vấn, gợi ý sản phẩm và giải đáp thắc mắc cho người dùng, qua đó tối ưu hóa quy trình bán hàng trực tuyến.

MỞ ĐẦU

1. Lý do chọn đề tài

Ngày nay, thương mại điện tử đã trở thành một xu hướng tất yếu, thay đổi hoàn toàn thói quen mua sắm của người tiêu dùng. Đặc biệt, đối với ngành kinh doanh nội thất, một lĩnh vực đòi hỏi tính thẩm mỹ và sự phù hợp cao với không gian sống và nhu cầu mua sắm trực tuyến đang tăng trưởng nhanh chóng. Tuy nhiên, các website bán nội thất truyền thống hiện nay phần lớn chỉ dừng lại ở việc trưng bày hình ảnh và thông số sản phẩm khô khan.

Vấn đề lớn nhất mà khách hàng gặp phải khi mua nội thất online là thiếu sự tư vấn trực tiếp và mang tính cá nhân hóa. Họ thường băn khoăn liệu sản phẩm này có phù hợp với phong cách nhà mình không, hay kích thước này có vừa vặn không. Việc thiếu vắng một "trợ lý ảo" am hiểu sản phẩm để giải đáp thắc mắc ngay lập tức dẫn đến tỷ lệ chuyển đổi đơn hàng thấp và trải nghiệm người dùng chưa trọn vẹn.

Mặt khác, công nghệ Trí tuệ nhân tạo (AI), đặc biệt là các mô hình ngôn ngữ lớn (LLM) như GPT của OpenAI, đang mở ra những cơ hội đột phá trong việc tự động hóa quy trình chăm sóc khách hàng. Việc tích hợp AI vào website không chỉ giúp doanh nghiệp tiết kiệm chi phí nhân sự mà còn mang lại trải nghiệm tương tác 24/7 cho người dùng.

Xuất phát từ thực tế đó, cùng với mong muốn làm chủ các công nghệ lập trình web hiện đại như Spring Boot (Backend) và ReactJS (Frontend) đề tài “Xây dựng website bán nội thất tích hợp chatbot AI tư vấn khách hàng” là giải pháp kết hợp giữa nền tảng thương mại điện tử vững chắc và công nghệ AI tiên tiến, nhằm giải quyết bài toán tư vấn trực tuyến và nâng cao trải nghiệm mua sắm nội thất.

2. Mục đích nghiên cứu

Tìm hiểu sâu và nắm vững kiến trúc xây dựng ứng dụng web hiện đại, quy trình phát triển phần mềm từ Backend đến Frontend và cơ chế tích hợp API của bên thứ ba (OpenAI, VNPAY) vào hệ thống.

Xây dựng thành công một hệ thống website bán nội thất hoàn chỉnh, hoạt động ổn định với đầy đủ chức năng quản lý và bán hàng. Đặc biệt là ứng dụng thành công Chatbot AI có khả năng hiểu ngôn ngữ tự nhiên, tư vấn sản phẩm thông minh dựa

trên dữ liệu thực tế của website, giúp người dùng hiểu rõ hơn về sản phẩm và dễ dàng đưa ra quyết định mua sắm.

3. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài tập trung vào việc tìm hiểu và ứng dụng các kiến thức chuyên sâu về kiến trúc RESTful API cùng các phương thức xác thực người dùng hiện đại. Về mặt công nghệ, nghiên cứu đi sâu vào hệ sinh thái Java thông qua framework Spring Boot cho phía máy chủ (Back-end), kết hợp với ReactJS và Tailwind CSS để xây dựng giao diện người dùng (Front-end) tối ưu và linh hoạt. Hệ quản trị cơ sở dữ liệu MySQL được sử dụng để quản lý dữ liệu hệ thống. Bên cạnh đó, đề tài còn nghiên cứu việc tích hợp OpenAI API thông qua kỹ thuật Prompt Engineering nhằm xây dựng chatbot tư vấn thông minh, đồng thời ứng dụng cổng thanh toán VNPAY để thực hiện các giao dịch trực tuyến an toàn và tiện lợi.

4. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài giới hạn trong việc xây dựng và hoàn thiện một ứng dụng website thương mại điện tử dành riêng cho cửa hàng kinh doanh đồ nội thất. Trong đó, nghiên cứu tập trung vào việc triển khai cơ sở lý thuyết về mô hình kiến trúc MVC kết hợp Service Layer nhằm đảm bảo tính phân lớp và dễ bảo trì cho mã nguồn. Nội dung thực hiện bao gồm quy trình phân tích, thiết kế hệ thống cơ sở dữ liệu quan hệ thông qua sơ đồ ERD, thiết kế quy trình nghiệp vụ bán hàng và đặc biệt là tích hợp chatbot AI hỗ trợ tư vấn khách hàng trực tuyến.

CHƯƠNG 1: TỔNG QUAN

1.1 Mô tả bài toán

Sự bùng nổ của cuộc cách mạng công nghiệp 4.0 và làn sóng chuyển đổi số đã thay đổi căn bản thói quen mua sắm của người tiêu dùng trên toàn cầu. Từ phương thức giao dịch truyền thống tại các cửa hàng vật lý, xu hướng tiêu dùng hiện đại đã chuyển dịch mạnh mẽ sang thương mại điện tử nhờ vào sự tiện lợi, khả năng tiếp cận đa dạng và tốc độ xử lý nhanh chóng. Những bước tiến này không chỉ giúp xóa bỏ rào cản về địa lý, thời gian mà còn mở ra những trải nghiệm mua sắm thông minh, cá nhân hóa trong nhiều lĩnh vực, từ thời trang, điện tử cho đến trang trí nhà cửa.

Kinh doanh nội thất là một lĩnh vực giàu tiềm năng nhưng cũng đầy thách thức khi bước chân lên nền tảng số. Đây là mặt hàng đặc thù với giá trị cao, yêu cầu khắt khe về thông số kỹ thuật, chất liệu và tính thẩm mỹ.

Khác với các mặt hàng tiêu dùng nhanh, khách hàng khi mua nội thất thường cần nhiều thông tin chi tiết và sự tư vấn kỹ lưỡng trước khi ra quyết định. Tuy nhiên, các website bán hàng truyền thống thường chỉ dừng lại ở việc trưng bày hình ảnh tĩnh, thiếu đi sự tương tác trực tiếp và các công cụ hỗ trợ thanh toán linh hoạt, khiến trải nghiệm người dùng chưa thực sự trọn vẹn.

Đề tài này hướng đến việc phát triển hệ thống website thương mại điện tử bán nội thất được xây dựng trên nền tảng công nghệ web hiện đại (Spring Boot và ReactJS) để giải quyết các vấn đề trên. Hệ thống không chỉ cung cấp đầy đủ các chức năng quản lý bán hàng chuyên nghiệp, tích hợp cổng thanh toán số an toàn (VNPay) mà còn ứng dụng Trí tuệ nhân tạo (AI Chatbot) để nâng cao trải nghiệm khách hàng. Việc tích hợp Chatbot AI giúp tự động hóa quy trình tư vấn, giải đáp thắc mắc về sản phẩm và hỗ trợ người dùng 24/7, qua đó tạo ra một môi trường mua sắm nội thất trực tuyến tin cậy, tiện lợi và mang tính tương tác cao.

1.2 Công cụ và cách thức thực hiện

Để thực hiện đề tài xây dựng website bán nội thất, cần tập trung nghiên cứu các kiến thức về ngôn ngữ lập trình Java, framework Spring Boot, thư viện ReactJS, hệ quản trị cơ sở dữ liệu quan hệ MySQL, cũng như các kỹ thuật tích hợp thanh toán điện tử và AI Chatbot. Cụ thể, cần nghiên cứu các nội dung sau:

Java và Spring Boot: Java là ngôn ngữ lập trình hướng đối tượng mạnh mẽ, có tính bảo mật cao và khả năng xử lý đa luồng tốt. Spring Boot là một framework được xây dựng trên nền tảng Java, giúp đơn giản hóa quá trình phát triển ứng dụng web Enterprise. Đây là lựa chọn phù hợp cho phần Backend của hệ thống thương mại điện tử nhờ khả năng xử lý các nghiệp vụ phức tạp như đơn hàng, thanh toán và quản lý tồn kho một cách chặt chẽ.

ReactJS: Là một thư viện JavaScript mã nguồn mở được sử dụng để xây dựng giao diện người dùng (User Interface). ReactJS hoạt động dựa trên mô hình Component và Virtual DOM, cho phép tạo ra các ứng dụng Single Page Application (SPA) với trải nghiệm mượt mà. Trong đề tài này, ReactJS được sử dụng để xây dựng giao diện phía Client, giúp hiển thị hình ảnh sản phẩm nội thất trực quan, xử lý giỏ hàng nhanh chóng mà không cần tải lại trang.

MySQL: Là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phổ biến nhất hiện nay. MySQL lưu trữ dữ liệu dưới dạng các bảng có quan hệ chặt chẽ với nhau thông qua khóa chính và khóa ngoại. Với đặc thù của ngành kinh doanh nội thất cần sự chính xác tuyệt đối về số liệu, MySQL đảm bảo tính toàn vẹn dữ liệu.

VNPay và OpenAI API: Nghiên cứu tài liệu kỹ thuật của VNPay để tích hợp cổng thanh toán trực tuyến an toàn. Đồng thời, tìm hiểu cách tích hợp API của OpenAI để xây dựng Chatbot hỗ trợ tư vấn khách hàng tự động.

Ưu điểm của việc sử dụng các công nghệ này:

Spring Boot: Xây dựng kiến trúc dễ dàng, dễ bảo trì và mở rộng, đặc biệt an toàn cho các giao dịch tài chính.

ReactJS: Tối ưu hóa trải nghiệm người dùng, giúp website tương tác nhanh, giao diện thân thiện trên cả máy tính và thiết bị di động.

MySQL: Đảm bảo tính nhất quán của dữ liệu, tránh các lỗi sai sót về tồn kho hoặc doanh thu.

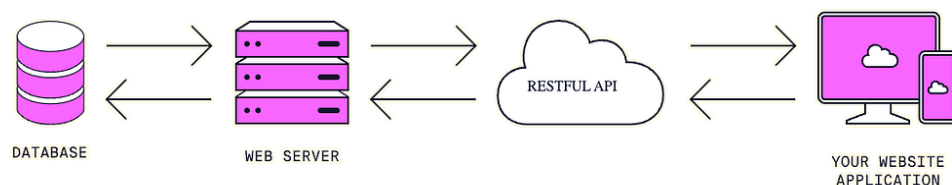
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Tìm hiểu về Restful API

2.1.1 Tổng quan về Restful API

RESTful API (Representational State Transfer API) là một kiến trúc thiết kế cho việc xây dựng các dịch vụ web linh hoạt, đơn giản và dễ dàng tích hợp. Được đặt ra bởi Roy Fielding trong luận văn tiến sĩ của mình vào năm 2000, RESTful API dựa trên các nguyên tắc cơ bản của REST, một mô hình truyền thông không trạng thái và có khả năng mở rộng.

RESTful API sử dụng các URL để đọc và để hiểu và sử dụng các định dạng dữ liệu như JSON hoặc XML để trao đổi thông tin giữa máy chủ và máy khách. RESTful API có tính khả di động cao, cho phép các ứng dụng khác nhau có thể truy cập và sử dụng các tài nguyên một cách dễ dàng[1].



Hình 2. 1 Mô tả RESTful API (nguồn Internet)

2.1.2 Các phương thức sử dụng trong Restful API

[GET]: Sử dụng để đọc dữ liệu từ một tài nguyên. Một yêu cầu GET sẽ trả về thông tin của tài nguyên được yêu cầu.

[POST]: Được sử dụng để tạo mới một tài nguyên. Thông thường, dữ liệu gửi kèm theo yêu cầu POST được sử dụng để tạo mới tài nguyên trên máy chủ.

[PUT]: Sử dụng để cập nhật một tài nguyên hoặc tạo mới nếu nó không tồn tại. Yêu cầu PUT thường đi kèm với dữ liệu hoặc thông tin cần cập nhật.

[PATCH]: Sử dụng để thực hiện cập nhật một phần nhỏ của tài nguyên. Nó giúp giảm lượng dữ liệu gửi đi so với việc sử dụng PUT.

[DELETE]: Dùng để xóa một tài nguyên. Yêu cầu DELETE sẽ gửi thông điệp đến máy chủ để yêu cầu xóa tài nguyên được xác định.

[OPTIONS]: Yêu cầu thông tin về các phương thức được hỗ trợ trên một tài nguyên cụ thể hoặc toàn bộ ứng dụng.

[HEAD]: Tương tự như GET, nhưng máy chủ chỉ trả về tiêu đề của yêu cầu, không có nội dung thực tế. Thường được sử dụng để kiểm tra trạng thái của một tài nguyên mà không tải toàn bộ nội dung.

2.1.3 Các mã trạng thái phản hồi

Thành công (2xx): 200 (OK), 201 (Đã tạo thành công), 204 (Xóa thành công/Không có nội dung).

Điều hướng/Cache (3xx): 304 (Dữ liệu chưa thay đổi, dùng cache).

Lỗi từ phía Client (4xx): 400 (Yêu cầu không hợp lệ), 404 (Không tìm thấy), 401 (Cần xác thực), 403 (Bị từ chối truy cập), 405 (Phương thức không cho phép), 422 (Dữ liệu lỗi), 429 (Gửi quá nhiều yêu cầu), 410 (Tài nguyên không còn), 415 (Sai định dạng).

2.1.4 Ưu và nhược điểm

Ưu điểm

Đơn giản hóa: Sử dụng chuẩn HTTP giúp lập trình viên tập trung vào chức năng chính.

Tính linh hoạt và tương thích: Dễ dàng kết nối, mở rộng và tích hợp giữa các ứng dụng khác nhau.

Tính độc lập: Các thành phần hoạt động tách biệt, dễ bảo trì và nâng cấp.

Nhược điểm

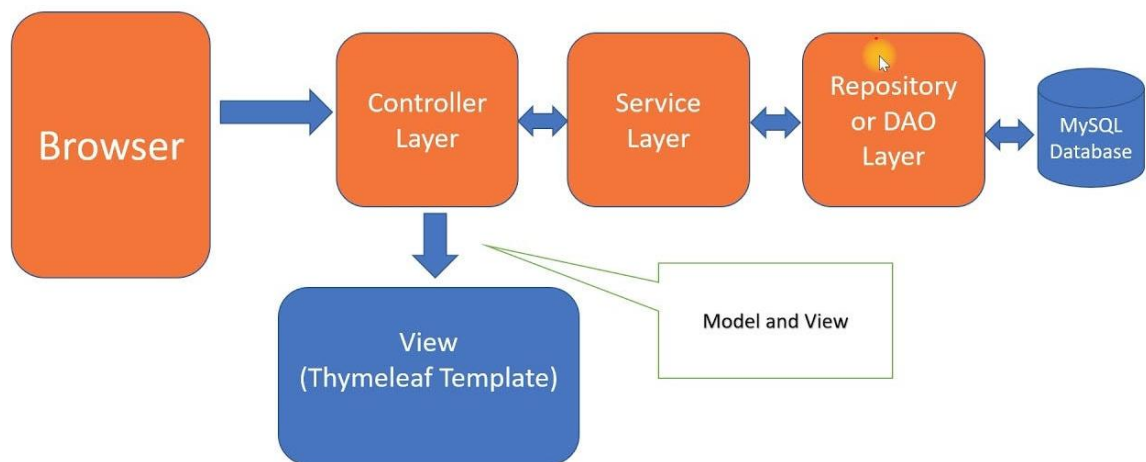
Chi phí: Việc phát triển và triển khai API đôi khi rất tốn kém và đòi hỏi bảo trì cũng như hỗ trợ cao từ các nhà phát triển.

Bảo mật: Việc thêm tầng giao tiếp API và phụ thuộc vào giao thức HTTP làm tăng nguy cơ bị tấn công, đòi hỏi các biện pháp bảo mật cao.

2.2 Kiến trúc MVC kết hợp Service Layer

2.2.1 Tổng quan về kiến trúc MVC kết hợp Service Layer

Trong phát triển phần mềm hiện đại, đặc biệt là các hệ thống Backend sử dụng Java Spring Boot, mô hình MVC (Model-View-Controller) truyền thống thường được mở rộng thêm tầng Service Layer. Đây là một biến thể kiến trúc nhằm giải quyết vấn đề Controller quá tải bằng cách tách biệt logic nghiệp vụ ra khỏi luồng điều hướng.



Hình 2. 2 Mô tả kiến trúc MVC kết hợp Service Layer (nguồn Internet)

Hệ thống được tổ chức thành 4 thành phần tương tác chặt chẽ:

- Controller Layer:

Đóng vai trò là cửa ngõ tiếp nhận các yêu cầu từ phía Client.

Thực hiện kiểm tra tính hợp lệ của dữ liệu đầu vào.

Không chứa logic nghiệp vụ, mà chỉ điều phối yêu cầu xuống tầng Service và trả về kết quả dưới dạng JSON.

- Service Layer:

Đây là phần quan trọng của hệ thống, nằm giữa Controller và Repository.

Chứa toàn bộ các thuật toán quan trọng của hệ thống.

- Repository Layer:

Chịu trách nhiệm giao tiếp trực tiếp với Cơ sở dữ liệu.

Thực hiện các câu lệnh truy vấn hoặc cập nhật mà không chứa bất kỳ logic nghiệp vụ nào.

- Model hoặc Entity:

Là các đối tượng đại diện cho dữ liệu, ánh xạ trực tiếp với các bảng trong cơ sở dữ liệu.

2.2.2 Ưu điểm của mô hình

Tách biệt mối quan tâm: Mỗi tầng chỉ tập trung vào một nhiệm vụ duy nhất. Controller chỉ lo điều hướng, Service chỉ lo tính toán, Repository chỉ lo lưu trữ. Điều này giúp mã nguồn dễ đọc và dễ quản lý.

Khả năng tái sử dụng cao: Một hàm xử lý logic trong Service có thể được gọi bởi nhiều Controller khác nhau mà không cần viết lại mã.

Dễ dàng kiểm thử: Nhờ sự tách biệt, ta có thể viết Unit Test cho tầng Service một cách độc lập bằng cách giả lập tầng Repository và Controller.

Tính bảo mật và toàn vẹn: Service Layer là nơi lý tưởng để đặt các ranh giới bảo mật và quản lý giao dịch cơ sở dữ liệu, đảm bảo dữ liệu không bị sai lệch khi có lỗi xảy ra giữa chừng.

Dễ bảo trì và nâng cấp: Khi cần thay đổi quy tắc nghiệp vụ, chỉ cần sửa tại Service mà không ảnh hưởng đến Controller hay cấu trúc Database.

2.2.3 Nhược điểm của mô hình

Tăng độ phức tạp ban đầu: Việc chia nhỏ hệ thống thành nhiều tầng buộc lập trình viên phải tạo nhiều file và class hơn. Điều này có thể gây cảm giác cồng kềnh đối với các dự án quá nhỏ hoặc đơn giản.

Mã lặp: Trong các chức năng CRUD đơn giản, tầng Service đôi khi chỉ đóng vai trò trung gian truyền tin từ Controller xuống Repository, tạo ra các đoạn mã dư thừa không thực sự xử lý logic.

Độ trễ: Việc đi qua nhiều tầng trung gian sẽ tốn nhiều tài nguyên hệ thống hơn so với việc gọi trực tiếp. Tuy nhiên, với sức mạnh phần cứng hiện nay, độ trễ này là không đáng kể so với lợi ích về mặt kiến trúc mà nó mang lại.

2.3 Tìm hiểu về ReactJS và công cụ Vite

2.3.1 Giới thiệu tổng quan về ReactJS

ReactJS (thường gọi là React) là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook (nay là Meta) vào năm 2013. React được sử dụng để xây dựng giao diện người dùng. [2]

Khác với các framework truyền thống thao tác trực tiếp với DOM, React giới thiệu tư duy lập trình theo hướng khai báo, giúp mã nguồn dễ đọc, dễ bảo trì. Hiện nay, React là một trong những thư viện Front-end phổ biến nhất thế giới nhờ cộng đồng lớn mạnh và các công cụ, thư viện và tài nguyên hỗ trợ phong phú.

2.3.2 Các đặc điểm nổi bật của ReactJS

Component-based: React chia nhỏ giao diện người dùng thành các thành phần độc lập, có thể tái sử dụng gọi là Component. Mỗi Component quản lý trạng thái riêng của nó, giúp việc phát triển các giao diện phức tạp trở nên dễ dàng hơn.

Virtual DOM: Đây là yếu tố quan trọng nhất tạo nên tốc độ của React. Thay vì cập nhật trực tiếp lên Real DOM (thường rất chậm), React tạo ra một bản sao trong bộ nhớ. Khi trạng thái thay đổi, React so sánh Virtual DOM mới và cũ, sau đó chỉ cập nhật những phần thực sự thay đổi lên Real DOM.

JSX (JavaScript XML): Một cú pháp mở rộng cho phép viết mã HTML ngay trong file JavaScript. JSX giúp code trở nên trực quan và tận dụng được sức mạnh của JavaScript để xử lý logic giao diện.

2.3.3 React Hooks

Hooks cho phép sử dụng State và các tính năng vòng đời (lifecycle) của React ngay trong các Functional Component mà không cần phải viết Class Component. Điều này giúp mã nguồn gọn nhẹ và dễ tái sử dụng logic hơn.

Các Hooks phổ biến thường được áp dụng trong dự án bao gồm:

useState: Dùng để khai báo và quản lý trạng thái của component.

useEffect: Dùng để xử lý các tác vụ như gọi API, thay đổi DOM,...

useContext: Hỗ trợ chia sẻ dữ liệu toàn cục giữa các component mà không cần truyền props qua nhiều cấp trung gian.

2.3.4 Công cụ ViteJS

Vite là một công cụ xây dựng thể hệ mới dành cho Front-end, được tạo ra bởi Evan You. Công cụ này ra đời nhằm giải quyết các hạn chế về hiệu suất, đặc biệt là tình trạng chậm trễ khi khởi động server và cập nhật code thường gặp ở các công cụ đóng gói truyền thống khi dự án quy mô lớn.

Vite hoạt động dựa trên cơ chế tận dụng các module ES gốc của trình duyệt. Thay vì phải đóng gói toàn bộ mã nguồn thành một tệp tin lớn trước khi chạy server, Vite biên dịch và phục vụ từng module riêng biệt ngay khi trình duyệt gửi yêu cầu.

Bên cạnh tốc độ khởi động, Vite cho phép cập nhật các thay đổi trong mã nguồn lên giao diện ngay lập tức mà không cần tải lại toàn bộ trang, giúp duy trì trạng thái của ứng dụng trong quá trình phát triển.

2.4 Tìm hiểu về Tailwind CSS

2.4.1 Giới thiệu tổng quan

Tailwind CSS là một framework mã nguồn mở dành cho CSS hoạt động theo cơ chế “Utility-first” (ưu tiên lớp tiện ích). Khác với các framework UI truyền thống thường cung cấp sẵn các thành phần giao diện hoàn chỉnh, Tailwind cung cấp một tập hợp các lớp tiện ích cấp thấp. Các lớp này tương ứng trực tiếp với các thuộc tính CSS cụ thể.

Phương pháp này cho phép lập trình viên xây dựng các giao diện người dùng tùy chỉnh hoàn toàn ngay trong tệp HTML hoặc JSX mà không cần phải viết từng dòng CSS riêng biệt. Tailwind CSS được đánh giá là một trong những công cụ tạo giao diện linh hoạt và phổ biến nhất trong phát triển Web hiện đại. [3]

Lịch sử hình thành và phát triển

Tailwind CSS được tạo ra bởi Adam Wathan, một lập trình viên và doanh nhân trong lĩnh vực phần mềm. Ý tưởng về Tailwind bắt nguồn từ quá trình Wathan làm việc với các dự án cá nhân, nơi ông nhận ra những hạn chế của phương pháp viết CSS theo hướng ngữ nghĩa và các framework dựa trên thành phần.

Năm 2017, Phiên bản alpha đầu tiên được công bố. Ban đầu, nó chỉ là một dự án phụ nhằm giải quyết vấn đề tái sử dụng code CSS. Tuy nhiên, cộng đồng lập trình viên nhanh chóng đón nhận nhờ tính thực tế cao.

Tháng 5/2019, Tailwind CSS v1.0 chính thức phát hành, đánh dấu sự ổn định về cấu trúc và khẳng định một hướng đi nghiêm túc trong thiết kế web.

Các phiên bản sau này tích hợp thêm các tính năng quan trọng như Dark Mode và đặc biệt là công cụ biên dịch Just-In-Time (JIT) trong phiên bản 3.0, giúp cải thiện tốc độ build và cho phép sử dụng các giá trị tùy ý.

2.4.2 Ưu điểm của Tailwind CSS

Tốc độ phát triển nhanh: Lập trình viên không cần chuyển đổi qua lại giữa tệp HTML và CSS hay mất thời gian suy nghĩ tên class, việc định kiểu diễn ra trực tiếp và tức thì.

Tối ưu hóa hiệu suất: Tailwind tự động loại bỏ tất cả các class không sử dụng khi đóng gói sản phẩm. Điều này giúp tệp CSS cuối cùng có dung lượng cực kỳ nhỏ, giúp trang web tải nhanh hơn.

Khả năng tùy biến cao: Tailwind cung cấp một tệp cấu hình cho phép định nghĩa lại toàn bộ hệ thống màu sắc, phông chữ, khoảng cách để phù hợp với nhận diện thương hiệu của dự án (tệp `tailwind.config.js`).

Hỗ trợ Responsive Design: Việc xây dựng giao diện thích ứng với nhiều thiết bị trở nên đơn giản được tích hợp sẵn ngay trong tên class.

2.4.3 Nhược điểm và hạn chế

Mã nguồn HTML trở nên phức tạp: Do phải liệt kê nhiều class tiện ích cho một phần tử, mã nguồn HTML có thể trở nên dài dòng và rối mắt. Điều này có thể gây khó khăn cho việc đọc hiểu cấu trúc DOM.

Đường cong học tập ban đầu: Người mới bắt đầu cần thời gian để ghi nhớ các tên class quy ước của Tailwind.

Phụ thuộc vào công cụ Build: Tailwind CSS yêu cầu phải có quy trình thiết lập môi trường để hoạt động hiệu quả nhất, không đơn giản chỉ là nhúng một đường link CDN như CSS truyền thống.

2.5 Tìm hiểu về MySQL

2.5.1 Giới thiệu tổng quan

MySQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS - Relational Database Management System) mã nguồn mở phổ biến nhất trên thế giới. MySQL hoạt động dựa trên mô hình Client-Server và sử dụng Ngôn ngữ truy vấn mang tính cấu trúc để thực hiện các thao tác quản lý dữ liệu như thêm, sửa, xóa và truy xuất.

Dữ liệu trong MySQL được lưu trữ dưới dạng các bảng có liên kết chặt chẽ với nhau thông qua các khóa chính và khóa ngoại. Đảm bảo tính toàn vẹn và nhất quán của dữ liệu.

2.5.2 Lịch sử hình thành

MySQL được phát triển ban đầu bởi công ty MySQL AB vào năm 1995. Sau đó, công ty này được Sun Microsystems mua lại. Đến năm 2010, khi Oracle Corporation mua lại Sun Microsystems, MySQL chính thức thuộc quyền quản lý và phát triển của Oracle cho đến nay. [4]

2.5.3 Ưu điểm của MySQL

Mã nguồn mở và miễn phí: Có phiên bản Community miễn phí, giúp tiết kiệm chi phí cho các dự án vừa và nhỏ.

Hiệu năng cao: Tốc độ truy xuất dữ liệu nhanh, ổn định, đáp ứng tốt cho các ứng dụng web có lưu lượng truy cập lớn.

Tính bảo mật: Hệ thống phân quyền người dùng chi tiết và hỗ trợ mã hóa dữ liệu mạnh mẽ.

Khả năng tương thích: Hoạt động tốt trên nhiều hệ điều hành và hỗ trợ hầu hết các ngôn ngữ lập trình phổ biến.

2.5.4 Nhược điểm và hạn chế

Khả năng mở rộng: Tuy hoạt động tốt với dữ liệu lớn, nhưng khi quy mô dữ liệu trở nên cực lớn, MySQL có thể gặp hạn chế về hiệu suất so với các hệ quản trị chuyên dụng khác như PostgreSQL hay các giải pháp NoSQL.

Hiệu suất truy vấn phức tạp: Khi xử lý các truy vấn JOIN phức tạp trên nhiều bảng lớn, MySQL có thể hoạt động chậm hơn và việc tối ưu hóa truy vấn đôi khi khó khăn hơn so với các RDBMS chuyên nghiệp khác.

2.6 Tìm hiểu về Spring Boot

2.6.1 Giới thiệu tổng quan và Lịch sử

Spring Boot là một dự án mã nguồn mở nằm trong hệ sinh thái Spring Framework (Java), được phát triển bởi Pivotal Software và ra mắt lần đầu vào năm 2014. Nó được xem là bản nâng cấp giúp đơn giản hóa việc khởi tạo và phát triển các ứng dụng dựa trên Spring.

Mục tiêu cốt lõi của Spring Boot là cung cấp trải nghiệm phát triển ứng dụng nhanh chóng. Thay vì yêu cầu lập trình viên phải thiết lập thủ công các tệp phức tạp để kết nối các thành phần, Spring Boot tự động hóa phần lớn các cấu hình mặc định, cho phép đội ngũ phát triển tập trung vào logic nghiệp vụ ngay lập tức. [5]

2.6.2 Các tính năng nổi bật và Ưu điểm

Embedded Server (Máy chủ nhúng): Spring Boot tích hợp sẵn các web server như Tomcat, Jetty hoặc Undertow ngay trong ứng dụng. Điều này cho phép ứng dụng chạy độc lập dưới dạng một tệp .jar mà không cần phải cài đặt và triển khai trên một web server riêng biệt.

Tự động cấu hình: Dựa trên các thư viện có trong classpath, Spring Boot sẽ tự động cấu hình các Bean cần thiết.

Starter Dependencies (Các gói phụ thuộc): Spring Boot cung cấp các gói “starter” giúp gộp các thư viện liên quan vào một gói duy nhất, đơn giản hóa việc quản lý phiên bản và sự phụ thuộc trong tệp pom.xml.

2.6.3 Ưu và nhược điểm của Spring Boot

Ưu điểm

- Tăng tốc độ phát triển dự án nhờ giảm thiểu mã nguồn cấu hình.
- Hệ sinh thái rộng lớn, cộng đồng hỗ trợ mạnh mẽ và tài liệu phong phú.
- Dễ dàng tích hợp với các công nghệ như Security, Cloud, Microservices.

Nhược điểm

- Tiêu tốn tài nguyên phần cứng nhiều hơn so với một số framework nhẹ khác do tải nhiều thư viện ngầm định.
- Cơ chế tự động hóa đôi khi gây khó khăn cho người mới trong việc debug lỗi sâu nếu không hiểu rõ nền tảng Spring Framework.

2.7 Tìm hiểu về cổng thanh toán VnPay

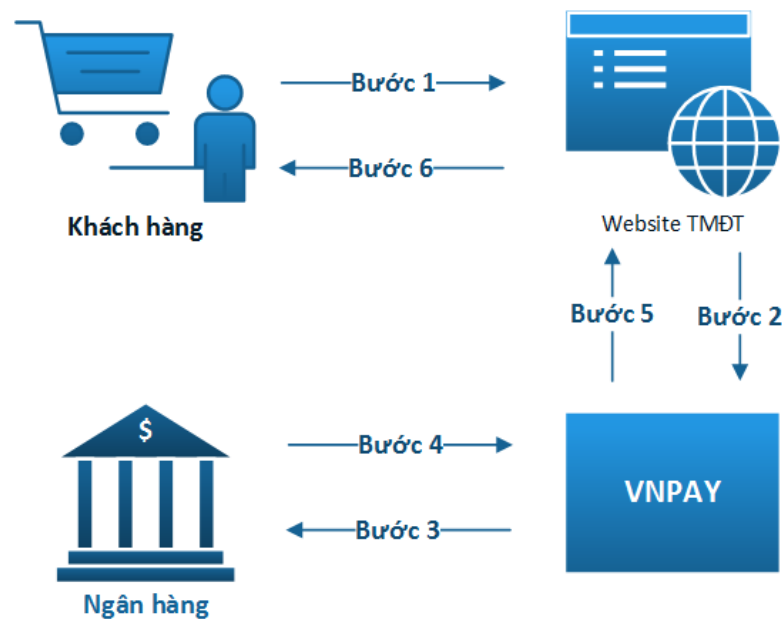
2.7.1 Giới thiệu và Vai trò trong hệ thống

VnPay (Công ty Cổ phần Giải pháp Thanh toán Việt Nam) là một nền tảng trung gian thanh toán điện tử, cung cấp hạ tầng kỹ thuật để kết nối các doanh nghiệp kinh doanh thương mại điện tử với hệ thống ngân hàng và các tổ chức tài chính.

Trong phạm vi dự án, VnPay đóng vai trò là cổng thanh toán chịu trách nhiệm xử lý các giao dịch tài chính trực tuyến. Giải pháp này cho phép hệ thống phần mềm chấp nhận thanh toán thông qua nhiều phương thức khác nhau như: ứng dụng Mobile Banking (quét mã QR), thẻ ATM nội địa và thẻ thanh toán quốc tế (Visa, Mastercard, JCB). Hệ thống tuân thủ các tiêu chuẩn bảo mật khắt khe như PCI DSS để đảm bảo an toàn dữ liệu cho người dùng cuối.

2.7.2 Kiến trúc tích hợp và luồng xử lý dữ liệu

Hệ thống sử dụng mô hình kết nối Redirect (Chuyển hướng). Đây là mô hình phổ biến nhất, đảm bảo tính bảo mật cao do toàn bộ quá trình nhập liệu thông tin thẻ/tài khoản ngân hàng của người dùng đều diễn ra trên hệ thống của VNPAY và Ngân hàng, Website TMĐT không lưu trữ các thông tin nhạy cảm này.



Hình 2. 3 Mô tả luồng xử lý dữ liệu của VNPay (nguồn VNPay)

Bước 1: Khách hàng thực hiện mua hàng trên Website - ứng dụng TMĐT và tiến hành thanh toán trực tuyến cho đơn hàng.

Bước 2: Website - ứng dụng TMĐT thành lập yêu cầu thanh toán dưới dạng URL mang thông tin thanh toán và chuyển hướng khách hàng sang Cổng thanh toán VNPAY bằng URL đó. Cổng thanh toán VNPAY xử lý yêu cầu thanh toán mà Website - ứng dụng TMĐT gửi sang. Khách hàng tiến hành nhập hoặc xử lý xác thực các thông tin được yêu cầu Thanh toán.

Bước 3,4: Khách hàng nhập thông tin để xác minh tài khoản Ngân hàng của khách hàng và xác thực giao dịch (Nhập thông tin tài khoản, thẻ hoặc quét mã VNPAY-QR).

Bước 5: Giao dịch thành công tại Ngân hàng, VNPAY tiến hành:

- Chuyển hướng khách hàng về Website - ứng dụng TMĐT.
- Thông báo cho Website - ứng dụng TMĐT kết quả thanh toán của khách hàng thông qua IPN URL. Merchant cập nhật kết quả thanh toán VNPAY gửi tại URL này.

Bước 6: Merchant hiển thị kết quả giao dịch tới khách hàng (vnp_ReturnUrl).

2.8 Tìm hiểu về OpenAI API

2.8.1 Giới thiệu tổng quan

OpenAI API là dịch vụ cung cấp giao diện lập trình ứng dụng, cho phép các nhà phát triển phần mềm kết nối và sử dụng các mô hình trí tuệ nhân tạo (AI) tiên tiến do OpenAI huấn luyện. Thay vì phải xây dựng và vận hành các máy chủ GPU đắt tiền để chạy mô hình AI, ứng dụng chỉ cần gửi dữ liệu qua Internet đến máy chủ của OpenAI để xử lý.

Cốt lõi của dịch vụ này là các mô hình ngôn ngữ thuộc dòng GPT. Các mô hình này được huấn luyện trên một lượng dữ liệu khổng lồ, cho phép chúng có khả năng hiểu ngữ cảnh, suy luận logic và tạo ra văn bản tự nhiên giống như con người.

2.8.2 Các mô hình và Cơ chế hoạt động

OpenAI cung cấp quyền truy cập vào nhiều lớp mô hình khác nhau để phục vụ các tác vụ chuyên biệt. Về mặt kỹ thuật, OpenAI API hoạt động theo kiến trúc RESTful. Quy trình tương tác dựa trên khái niệm Token.

Token là đơn vị xử lý văn bản cơ bản của mô hình AI. Một Token tương đương khoảng 4 ký tự tiếng Anh, hoặc xấp xỉ 0.75 từ tiếng Anh.

Input Token là chi phí cho văn bản gửi vào (Prompt).

Output Token là chi phí cho văn bản mô hình tạo ra. Chi phí Output luôn cao hơn Input vì việc tạo ra văn bản mới đòi hỏi nhiều tài nguyên xử lý hơn.

Context Window là tổng token tối đa có thể xử lý trong 1 request (input + output) từ người sử dụng.

Bảng 2. 1 Các mô hình OPENAI API phổ biến

Mô hình	Tên mô hình	Cửa sổ ngữ cảnh (Context Window)	Chi phí Input (USD/Token)	Chi phí Output (USD/Token)
GPT-5.1	gpt-5.1	400,000	\$1.25 / 1M	\$10.00 / 1M
GPT-4.1	gpt-4.1	1,047,576	\$2.00 / 1M	\$8.00 / 1M

GPT-4o	gpt-4o	128,000	\$2.50 / 1M	\$10.00 / 1M
GPT-4o-mini	gpt-4o-mini	128,000	\$0.15 / 1M	\$0.60 / 1M
GPT-3.5 Turbo	gpt-3.5-turbo	16,385	\$0.50 / 1M	\$1.50 / 1M

2.9 Các nghiệp vụ liên quan đến đề tài

2.9.1 Nghiệp vụ quản trị hệ thống và phân quyền người dùng

Quản lý định danh: Hệ thống thực hiện quy trình đăng ký thành viên mới, xác thực thông tin đăng nhập và duy trì phiên làm việc an toàn. Mật khẩu của người dùng luôn đảm bảo được mã hóa để bảo vệ dữ liệu cá nhân.

Phân cấp quyền hạn: Hệ thống phân chia người dùng thành hai nhóm đối tượng chính. Nhóm Quản trị viên (Admin) nắm quyền kiểm soát toàn bộ dữ liệu, cấu hình hệ thống và xem báo cáo. Nhóm Khách hàng (User) chỉ có quyền thao tác trên dữ liệu cá nhân và thực hiện các giao dịch mua sắm.

2.9.2 Nghiệp vụ quản lý danh mục và hàng hóa

Phân loại sản phẩm: Hàng hóa được tổ chức theo cấu trúc cây danh mục giúp tối ưu hóa khả năng tìm kiếm và lọc sản phẩm.

Quản lý vòng đời sản phẩm: Bao gồm các quy trình thêm mới sản phẩm vào hệ thống, cập nhật các thông số kỹ thuật chi tiết (kích thước, chất liệu, màu sắc, chế độ bảo hành) và thay đổi trạng thái kinh doanh (ngừng kinh doanh hoặc tạm hết hàng).

Kiểm soát tồn kho: Hệ thống theo dõi biến động số lượng sản phẩm theo thời gian thực. Mỗi khi có đơn hàng được xác nhận hoặc hủy bỏ, số lượng tồn kho sẽ tự động được điều chỉnh tăng/giảm tương ứng để phản ánh đúng thực tế kho hàng.

2.9.3 Nghiệp vụ bán hàng và xử lý đơn hàng trực tuyến

Quy trình đặt hàng: Cho phép khách hàng mua ngay hoặc thêm sản phẩm vào giỏ, kiểm tra lại đơn giá, số lượng và nhập thông tin giao nhận. Lưu trữ giá tại thời điểm đặt hàng để tránh xung đột dữ liệu khi giá sản phẩm thay đổi trong tương lai.

Xử lý trạng thái đơn hàng: Đơn hàng sau khi khởi tạo sẽ đi qua quy trình xử lý khép kín: Tiếp nhận, Đóng gói, Đang vận chuyển, Đang giao, Hoàn tất. Ngoài ra, cần xử lý thêm các trường hợp ngoại lệ như khách hàng hủy đơn.

Thanh toán trực tuyến: Tích hợp đa dạng các phương thức thanh toán, cho phép khách hàng thanh toán trực tiếp hoặc thanh toán trực tuyến. Hệ thống ghi nhận trạng thái thanh toán của từng giao dịch, đối soát mã giao dịch và xử lý các trường hợp hoàn tiền kNghiep vụ tiếp thị và xúc tiến thương mại

Quản lý Flash Sale: Thiết lập các chiến dịch bán hàng siêu giảm giá trong khung giờ nhất định. Bao gồm việc lựa chọn sản phẩm, định giá khuyến mãi đặc biệt và giới hạn số lượng bán ra cho từng sản phẩm trong đợt giảm giá.

Phát hành và quản lý mã giảm giá: Tạo lập các mã giảm giá với các ràng buộc như: hạn sử dụng, giá trị đơn hàng tối thiểu để áp dụng và số lượt sử dụng tối đa. Hệ thống tự động tính toán mức giảm giá và trừ trực tiếp vào tổng tiền đơn hàng khi mã hợp lệ được áp dụng.

2.9.4 Nghiệp vụ chăm sóc khách hàng, hỗ trợ và thanh toán trực tuyến

Thu thập đánh giá: Cho phép khách hàng đã mua sản phẩm thực hiện chấm điểm và viết bình luận. Các nội dung này được hiển thị công khai, giúp người mua sau có thêm cơ sở tham khảo đồng thời lắng nghe góp ý của khách hàng để khắc phục.

Hỗ trợ khách hàng: Giải đáp các thắc mắc thường gặp của khách hàng, tư vấn sản phẩm theo mô tả mà khách hàng cung cấp.

Thanh toán trực tuyến: Tích hợp cổng thanh toán trực tuyến VNPAY cho phép khách hàng thanh toán nhanh chóng qua thẻ ngân hàng, mã QR hoặc ví điện tử. Đảm bảo tính an toàn, bảo mật cho thông tin giao dịch của khách hàng. Sau khi xác thực thanh toán thành công, hệ thống sẽ tự động cập nhật trạng thái đơn hàng, gửi thông báo xác nhận và lưu lịch sử giao dịch để phục vụ công tác quản lý và đối soát.

2.10 Các công trình nghiên cứu liên quan

Việc nghiên cứu và xây dựng hệ thống thương mại điện tử cho mặt hàng nội thất đã được nhiều cá nhân và tổ chức triển khai với các hướng tiếp cận đa dạng. Dưới đây là các công trình tiêu biểu có liên quan trực tiếp đến đề tài:

Tác giả Nguyễn Thanh Kim Ngân – Trường Đại học Tài nguyên và Môi trường Hà Nội [6] đã thực hiện báo cáo thực tập về việc xây dựng website bán đồ nội thất, tập trung vào việc chuẩn hóa quy trình lập trình web. Nghiên cứu này nhấn mạnh vào việc thiết kế giao diện thân thiện với người dùng và tối ưu hóa các chức năng tương tác cơ bản, tạo tiền đề cho việc triển khai các hệ thống bán hàng trực tuyến ổn định.

Tác giả Nguyễn Thị Hà [7] thông qua dự án xây dựng website cho Công ty nội thất Sofia đã đưa ra giải pháp thực tiễn gắn liền với yêu cầu của doanh nghiệp. Nghiên cứu này tập trung vào việc quản lý danh mục sản phẩm theo đặc thù ngành nội thất (kích thước, chất liệu) và tích hợp các công cụ quảng bá hình ảnh sản phẩm một cách chuyên nghiệp.

Tác giả Trần Thị Lê [8] trong báo cáo thực tập đã trình bày chi tiết về quy trình kinh doanh đồ nội thất. Công trình này cung cấp các sơ đồ luồng dữ liệu chi tiết, giúp định hình logic nghiệp vụ từ khâu nhập hàng, quản lý kho cho đến khi sản phẩm được khách hàng đặt mua và vận chuyển.

Tác giả Đinh Thị Thanh Xuân [9] cũng đã tóm tắt các khía cạnh cốt lõi của đồ án xây dựng website kinh doanh nội thất, trong đó tập trung mạnh vào phân tích yêu cầu hệ thống. Nghiên cứu chỉ ra rằng đối với website nội thất, việc xử lý cơ sở dữ liệu (ERD) cho các biến thể sản phẩm là yếu tố quan trọng nhất để đảm bảo tính chính xác trong giao dịch thương mại điện tử.

Tác giả Nguyễn Thanh Tùng [10] trong báo cáo nghiên cứu về việc xây dựng trang web quảng cáo sản phẩm nội thất đã tập trung vào việc thiết kế giải pháp quảng bá thương hiệu trực tuyến cho doanh nghiệp. Công trình này không chỉ đơn thuần là một website bán hàng mà còn nhấn mạnh vào khả năng giới thiệu năng lực sản xuất và giá trị thẩm mỹ của các sản phẩm nội thất. Đây là tài liệu quan trọng trong việc định hình cách thức trình bày giao diện trực quan nhằm thu hút người dùng ngay từ lần truy cập đầu tiên.

Điểm chung của các công trình này là đều tập trung vào việc xây dựng các tính năng nền tảng của một website thương mại điện tử chuyên ngành nội thất, bao gồm quản lý danh mục sản phẩm, quy trình đặt hàng và thiết kế cơ sở dữ liệu phù hợp với đặc thù ngành. Tuy nhiên, phần lớn các hệ thống trên vẫn chỉ dừng lại ở mức độ cung

cấp thông tin một chiều, chưa tối ưu hóa được sự tương tác giữa hệ thống và khách hàng, đặc biệt là thiếu vắng việc trao đổi thông tin giữa khách hàng với hệ thống.

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Mô tả hệ thống

Hệ thống được xây dựng nhằm cung cấp giải pháp thương mại điện tử toàn diện cho lĩnh vực kinh doanh nội thất, được vận hành dựa trên cơ chế phân quyền chặt chẽ, tách biệt hoàn toàn chức năng giữa người quản trị hệ thống và khách hàng.

Đối với người mua, hệ thống lưu trữ và quản lý tập trung hồ sơ cá nhân, bao gồm lịch sử mua sắm và thông tin giao nhận hàng. Tính năng bảo mật được chú trọng thông qua việc mã hóa thông tin đăng nhập và kiểm soát chặt chẽ dữ liệu người dùng.

Cốt lõi của hệ thống là khả năng quản lý danh mục hàng hóa đa tầng, phù hợp với sự đa dạng của ngành nội thất. Mỗi sản phẩm được định danh và mô tả chi tiết không chỉ qua hình ảnh hay giá bán mà còn qua các thông số kỹ thuật chuyên biệt như chất liệu, kích thước, màu sắc, xuất xứ và bảo hành. Quản lý sản phẩm theo thời gian thực, tự động cập nhật trạng thái khả dụng của sản phẩm dựa trên các giao dịch phát sinh, đảm bảo thông tin hiển thị luôn đồng bộ với thực tế lưu kho.

Để thúc đẩy doanh số và tăng sự thu hút khách hàng, hệ thống có áp dụng các chương trình khuyến mãi gồm mã giảm giá và các đợt mở bán siêu khuyến mãi. Cơ chế mã giảm giá được thiết lập với các ràng buộc như giới hạn thời gian hiệu lực, giá trị đơn hàng tối thiểu, hạn mức giảm giá tối đa và số lượt sử dụng, giúp kiểm soát ngân sách khuyến mãi hiệu quả. Song song đó, tính năng Flash Sale cho phép thiết lập các khung giờ bán hàng đặc biệt. Tại đây, một lượng hàng hóa nhất định được tách riêng để bán với giá ưu đãi, có cơ chế đếm ngược và khóa số lượng nhằm ngăn chặn tình trạng bán vượt quá mức phân bổ cho phép.

Luồng vận hành đơn hàng được thiết kế khép kín từ lúc khởi tạo đến khi hoàn tất giao nhận. Khi một đơn hàng được ghi nhận, hệ thống sẽ thực hiện việc ghi lại giá bán tại thời điểm giao dịch để lưu trữ vào chi tiết đơn hàng, đảm bảo tính chính xác của dữ liệu lịch sử ngay cả khi giá gốc sản phẩm có sự thay đổi về sau. Trạng thái đơn hàng được cập nhật liên tục qua các khâu: chờ xử lý, đóng gói, vận chuyển, giao thành công hoặc hủy bỏ. Đi kèm với đó, phân hệ thanh toán hỗ trợ đa dạng phương thức gồm thanh toán khi nhận hàng (COD) và các kênh thanh toán trực tuyến (như VNPAY) cho phép theo dõi chi tiết mã giao dịch và trạng thái dòng tiền, bao gồm cả quy trình xử lý hoàn tiền hoặc các giao dịch thất bại.

Nhằm nâng cao uy tín và chất lượng dịch vụ, hệ thống cung cấp tính năng đánh giá sản phẩm minh bạch. Sau khi trải nghiệm sản phẩm, khách hàng có thể để lại bình luận và chấm điểm chất lượng theo thang đo tiêu chuẩn. Dữ liệu này được lưu trữ và hiển thị công khai, đóng vai trò tham khảo quan trọng cho những người mua sau, đồng thời giúp bộ phận quản trị nắm bắt thị hiếu và cải thiện chất lượng nguồn hàng.

Bên cạnh các chức năng thương mại cốt lõi, hệ thống tích hợp một trợ lý ảo thông minh (AI Chatbot) nhằm tối ưu hóa quy trình chăm sóc khách hàng và nâng cao trải nghiệm mua sắm trực tuyến. AI Chatbot được hoạt động liên tục, đóng vai trò tiếp nhận và xử lý các yêu cầu từ người dùng bất kể khung giờ nào.

Dựa trên khả năng kết nối sâu với dữ liệu hệ thống, Chatbot có thể tự động truy xuất và cung cấp thông tin chi tiết về đặc tính kỹ thuật của sản phẩm nội thất như chất liệu, kích thước, hướng dẫn bảo quản hay tình trạng tồn kho thực tế, gợi ý sản phẩm phù hợp với nhu cầu của khách hàng. Không chỉ dừng lại ở việc tư vấn sản phẩm, hệ thống AI còn có thể gợi ý các mã giảm giá hoặc chương trình khuyến mãi phù hợp, giúp thúc đẩy quyết định mua hàng nhanh chóng.

3.2. Xác định các yêu cầu chức năng của hệ thống

3.2.1 Yêu cầu chức năng dành cho khách hàng, người xem

- Chức năng tìm kiếm và lọc sản phẩm: Cung cấp cho khách hàng khả năng tìm kiếm nhanh chóng các món đồ nội thất theo tên, danh mục hoặc lọc theo mức giá.
- Chức năng xem chi tiết và đánh giá: Cho phép người xem thấy đầy đủ thông số kỹ thuật, xuất xứ, chế độ bảo hành của sản phẩm. Người dùng đã mua hàng có thể gửi đánh giá về chất lượng của sản phẩm.
- Chức năng giỏ hàng và đặt hàng: Hỗ trợ khách hàng thêm, xóa sản phẩm vào giỏ, chọn hoặc nhập mã giảm giá và tiến hành gửi đơn hàng với các thông tin giao nhận, ghi chú và lựa chọn phương thức thanh toán phù hợp.
- Chức năng đăng ký, đăng nhập: Cho phép khách hàng tạo tài khoản mới và đăng nhập vào hệ thống để lưu trữ thông tin cá nhân và mua hàng.
- Chức năng quản lý đơn hàng cá nhân: Cung cấp khả năng theo dõi trạng thái xử lý của đơn hàng hiện tại và xem lại lịch sử các đơn hàng đã thực hiện trước đó.

- Chức năng hỗ trợ tự động: Cho phép khách hàng tra cứu thông tin sản phẩm, kiểm tra đơn hàng và nhận tư vấn về sản phẩm và khuyến mãi tức thời thông qua trợ lý ảo thông minh.

3.2.2 Yêu cầu chức năng dành cho người quản trị

- Quản lý danh mục và sản phẩm: Thực hiện các tác vụ thêm mới, cập nhật hoặc xóa các danh mục và sản phẩm nội thất. Quản lý chi tiết các thông số như giá bán, giá giảm, số lượng tồn kho và các thuộc tính mô tả.

- Quản lý và xử lý đơn hàng: Tiếp nhận đơn đặt hàng từ khách hàng, cập nhật trạng thái đơn hàng, xử lý các trường hợp hủy đơn hoặc hoàn tiền.

- Quản lý chương trình khuyến mãi: Thiết lập và cấu hình các mã giảm giá và các đợt Flash Sale bao gồm việc quy định thời gian, mức giảm giá và số lượng sản phẩm áp dụng.

- Thống kê và báo cáo: Xem tổng quan về tình hình kinh doanh, thu nhập theo ngày, tháng... Xem số lượng hàng tồn kho, sản phẩm sắp hết hàng.

3.2.3 Yêu cầu phi chức năng của hệ thống

Tính bảo mật: Hệ thống phải đảm bảo an toàn thông tin cá nhân của người dùng, mật khẩu phải được mã hóa trước khi lưu trữ. Các giao dịch thanh toán phải tuân thủ các tiêu chuẩn bảo mật cơ bản.

Tính toàn vẹn dữ liệu: Đảm bảo sự chính xác tuyệt đối về số lượng tồn kho, đặc biệt khi có nhiều người cùng đặt mua một sản phẩm trong cùng một thời điểm.

Giao diện người dùng: Giao diện cần thân thiện, dễ sử dụng, bố cục trình bày sản phẩm nội thất phải trực quan, hỗ trợ hiển thị tốt trên nhiều thiết bị khác nhau.

Khả năng mở rộng: Hệ thống cần được thiết kế để dễ dàng nâng cấp, thêm mới các tính năng hoặc mở rộng danh mục sản phẩm trong tương lai mà không làm gián đoạn hoạt động hiện tại.

3.3. Phân tích các tác nhân của hệ thống

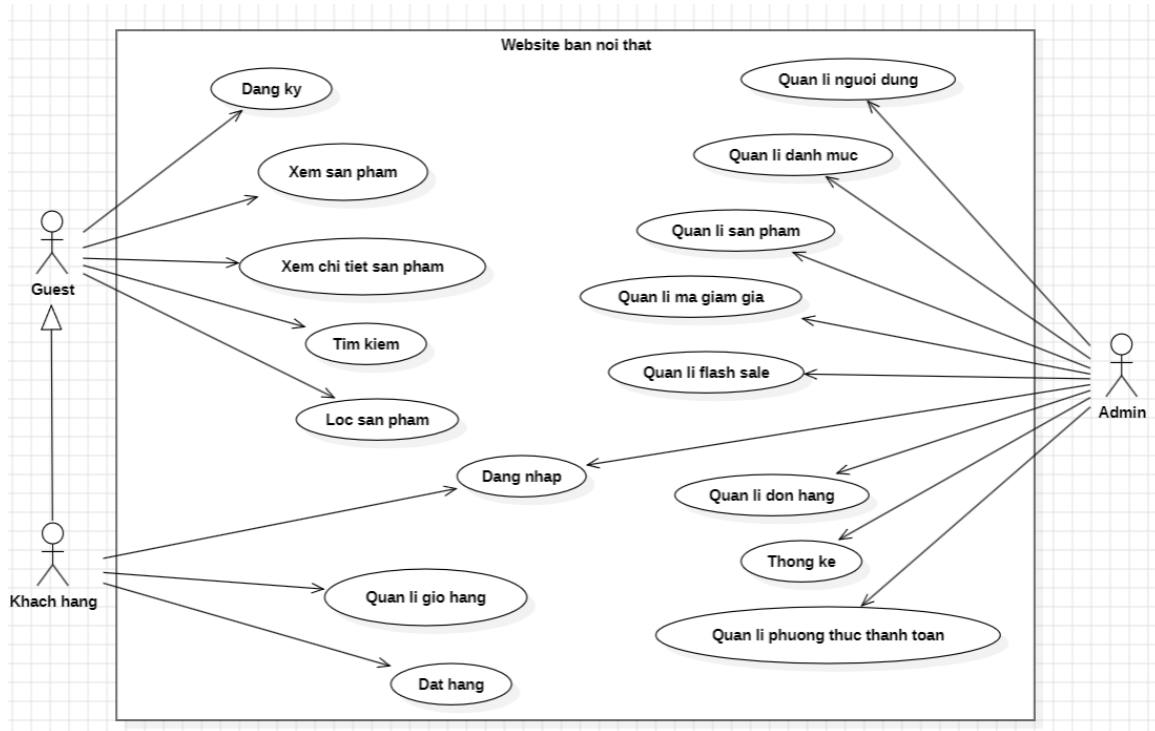
Hệ thống được thiết kế để phục vụ ba nhóm tác nhân chính gồm khách vãng lai, khách hàng và Admin. Mỗi nhóm có quyền hạn và phạm vi chức năng riêng biệt.

3.3.1 Các chức năng và quyền hạn của các tác nhân trong hệ thống

Bảng 3. 1 Mô tả các tác nhân tham gia vào hệ thống

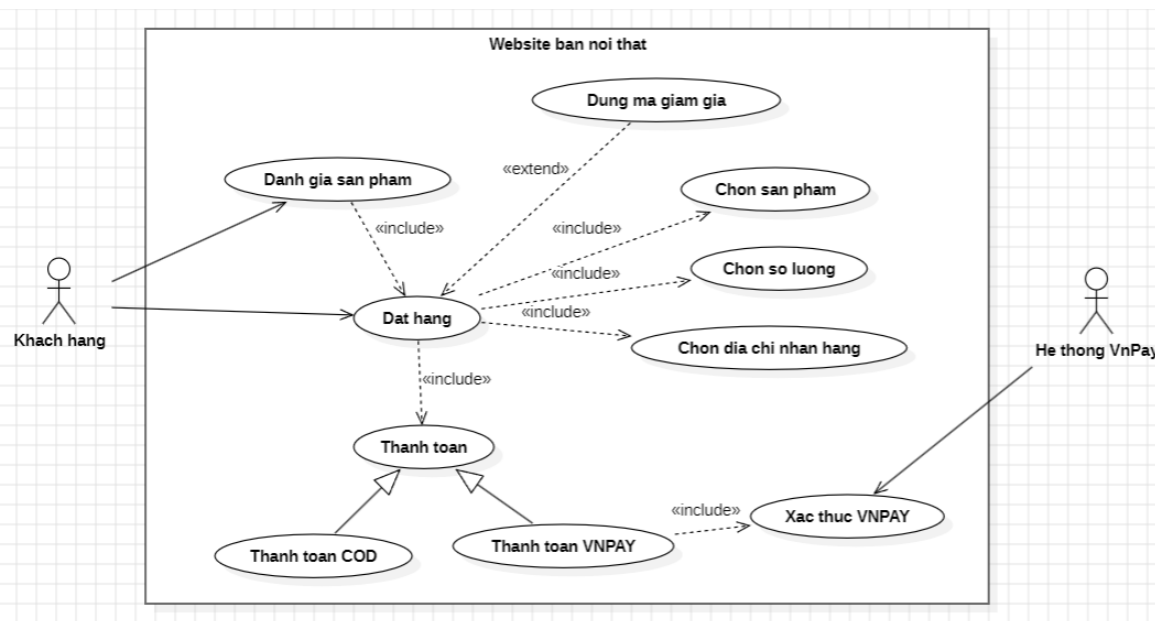
Tác nhân	Mô tả	Chức năng và quyền hạn
Khách vãng lai (Guest)	Là người dùng truy cập vào website nhưng chưa có tài khoản hoặc chưa đăng nhập. Khách vãng lai chỉ có quyền tiếp cận các thông tin công khai và thực hiện các thao tác cơ bản để tìm hiểu sản phẩm.	<ul style="list-style-type: none"> - Đăng ký tài khoản - Xem sản phẩm - Tìm kiếm và lọc sản phẩm - Tư vấn AI
Khách hàng	Là người dùng đã đăng ký và đăng nhập thành công. Đây là tác nhân chính thực hiện các giao dịch mua sắm, có quyền quản lý thông tin cá nhân và giỏ hàng. Khách hàng kế thừa toàn bộ chức năng của Khách vãng lai.	<ul style="list-style-type: none"> - Đăng nhập - Quản lý giỏ hàng - Đặt hàng - Thanh toán - Quản lý thông tin cá nhân
Admin	Là người vận hành hệ thống, nắm toàn quyền kiểm soát dữ liệu, cấu hình các chính sách bán hàng và xử lý các nghiệp vụ phía sau.	<ul style="list-style-type: none"> - Quản lý người dùng - Quản lý sản phẩm và danh mục - Quản lý đơn hàng - Quản lý khuyến mãi - Thống kê

3.3.2 Sơ đồ Use case tổng quát của hệ thống



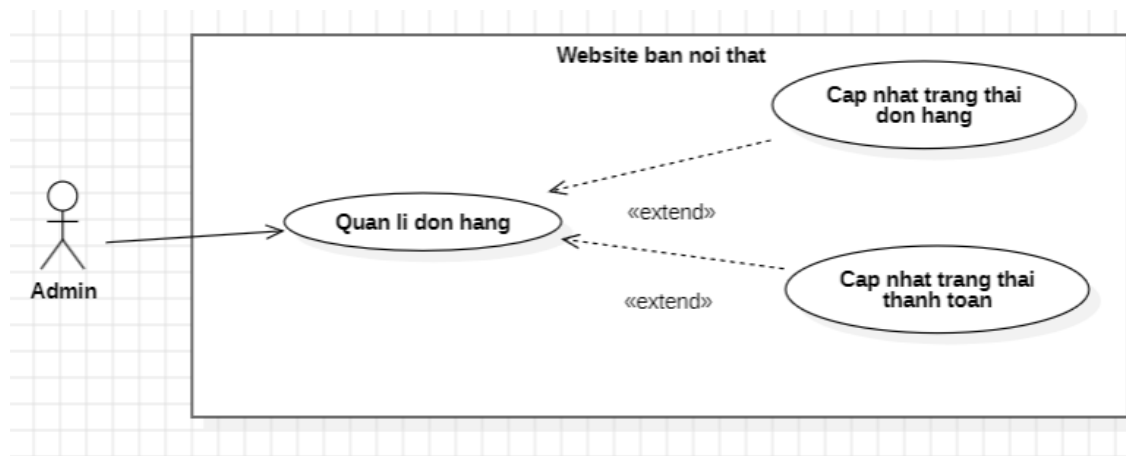
Hình 3. 1 Use case tổng quát của hệ thống

3.3.3 Sơ đồ Use case đặt hàng



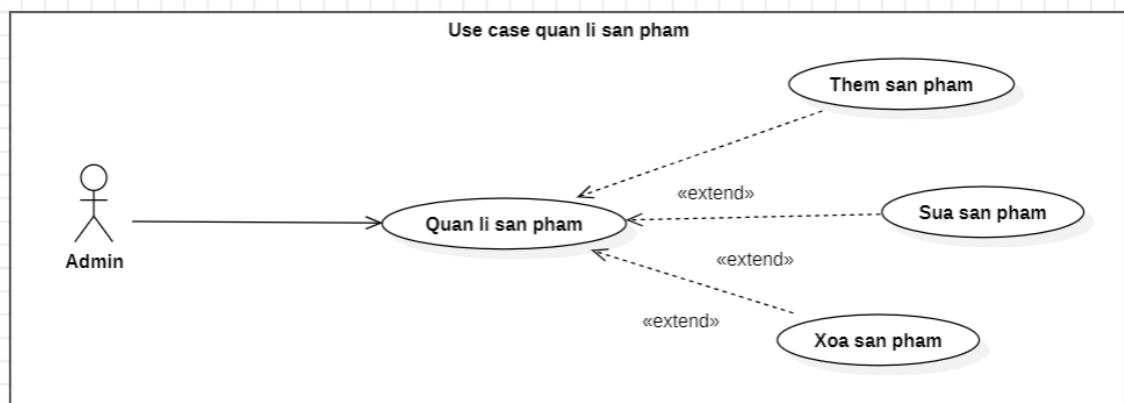
Hình 3. 2 Use case đặt hàng của hệ thống

3.3.4 Sơ đồ Use case quản lý đơn hàng



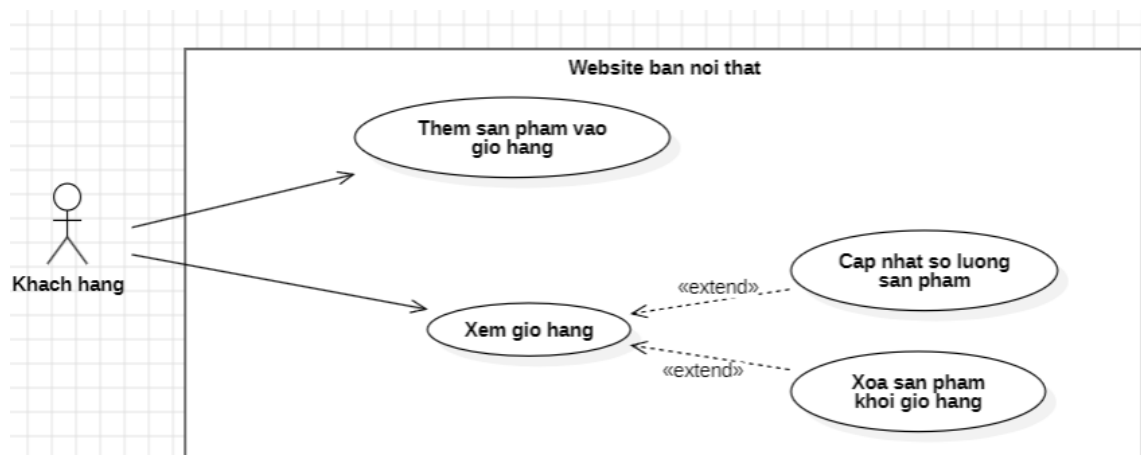
Hình 3. 3 Use case quản lí đơn hàng của hệ thống

3.3.5 Sơ đồ Use case quản lí sản phẩm



Hình 3. 4 Use case quản lí sản phẩm của hệ thống

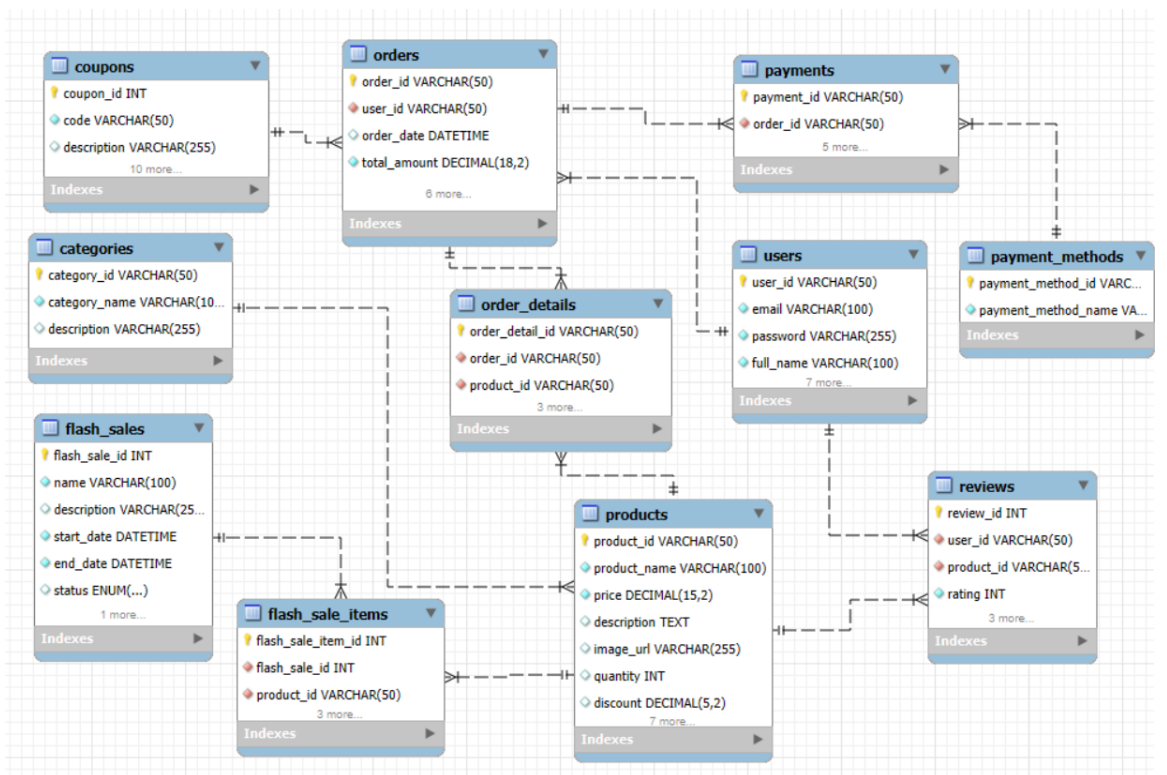
3.3.6 Sơ đồ Use case quản lí giỏ hàng



Hình 3. 5 Use case quản lí giỏ hàng của hệ thống

3.4. Mô hình dữ liệu hệ thống

Các bảng được liên kết chặt chẽ với nhau, đảm bảo tính toàn vẹn dữ liệu và hỗ trợ tốt cho các nghiệp vụ quản lý và bán nội thất.



Hình 3. 6 Mô hình dữ liệu của hệ thống

Mô tả các bảng

Bảng 3. 2 Mô tả bảng USERS

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
user_id	VARCHAR(50)	PRIMARY KEY	Mã định danh duy nhất của người dùng.
email	VARCHAR(100)	NOT NULL, UNIQUE	Địa chỉ email dùng để đăng nhập, không được trùng.
password	VARCHAR(255)	NOT NULL	Mật khẩu đã được mã hóa.
full_name	VARCHAR(100)	NOT NULL	Họ và tên đầy đủ của người dùng.

avatar	VARCHAR(255)	NULL	Đường dẫn đến ảnh đại diện.
address	VARCHAR(255)	NULL	Địa chỉ liên hệ của người dùng.
gender	ENUM('Nam', 'Nữ', 'Khác')	DEFAULT 'Khác'	Giới tính.
birth_date	DATE		Ngày sinh.
phone_number	VARCHAR(20)		Số điện thoại liên hệ.
Role	ENUM('ADMIN', 'USER')	DEFAULT 'USER'	Vai trò của người dùng trong hệ thống.
created_at	TIMESTAMP		Thời điểm tạo tài khoản.

Bảng 3. 3 Mô tả bảng CATEGORIES

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
category_id	VARCHAR(50)	PRIMARY KEY	Mã định danh duy nhất của danh mục.
category_name	VARCHAR(100)	NOT NULL	Tên danh mục.
description	VARCHAR(255)	NULL	Mô tả chi tiết danh mục.

Bảng 3. 4 Mô tả bảng PRODUCTS

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
product_id	VARCHAR(50)	PRIMARY KEY	Mã định danh sản phẩm.

product_name	VARCHAR(100)	NOT NULL	Tên sản phẩm.
price	DECIMAL(15,2)	NOT NULL	Giá bán niêm yết.
description	TEXT		Mô tả chi tiết sản phẩm.
image_url	VARCHAR(255)		Đường dẫn hình ảnh sản phẩm.
quantity	INT	DEFAULT 0	Số lượng tồn kho.
discount	DECIMAL(5,2)	DEFAULT 0	Phần trăm giảm giá (0–100%).
category_id	VARCHAR(50)	Khóa ngoại, liên kết đến bảng CATEGORIES	Liên kết đến danh mục sản phẩm.
Size	VARCHAR(50)		Kích thước sản phẩm.
color	VARCHAR(50)		Màu sắc.
material	VARCHAR(100)		Chất liệu.
warranty	VARCHAR(50)		Thông tin bảo hành.
origin	VARCHAR(100)		Xuất xứ sản phẩm.
created_at	TIMESTAMP		Thời gian tạo sản phẩm.

Bảng 3. 5 Mô tả bảng COUPONS

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
------------	--------------	-----------	-------

coupon_id	INT	PRIMARY KEY	ID tự tăng của mã giảm giá.
code	VARCHAR(50)	NOT NULL, UNIQUE	Mã khuyến mãi nhập vào hệ thống (VD: SALE50).
description	VARCHAR(255)		Mô tả chương trình khuyến mãi.
discount_type	ENUM('percent')	NOT NULL	Loại giảm giá (hiện chỉ hỗ trợ phần trăm).
discount_value	DECIMAL(10,2)	NOT NULL	Giá trị giảm (VD: 10 = 10%).
min_order_amount	DECIMAL(18,2)	DEFAULT 0	Giá trị đơn hàng tối thiểu.
max_discount	DECIMAL(18,2)		Mức giảm tối đa (áp dụng với %).
start_date	DATETIME	NOT NULL	Ngày bắt đầu hiệu lực.
end_date	DATETIME	NOT NULL	Ngày hết hạn.
usage_limit	INT		Giới hạn số lượt dùng tối đa.
used_count	INT	DEFAULT 0	Số lượt mã đã được sử dụng.
is_active	BOOLEAN		Trạng thái hiệu lực.
created_at	DATETIME		Ngày tạo mã.

Bảng 3. 6 Mô tả bảng PAYMENT METHODS

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
payment_method_id	VARCHAR(50)	PRIMARY KEY	Mã phương thức thanh toán.
payment_method_name	VARCHAR(50)	NOT NULL	Tên phương thức thanh toán.

Bảng 3. 7 Mô tả bảng ORDERS

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
order_id	VARCHAR(50)	PRIMARY KEY	Mã đơn hàng.
user_id	VARCHAR(50)	Khóa ngoại đến bảng USERS	Khách hàng đặt đơn.
order_date	DATETIME		Ngày đặt hàng.
total_amount	DECIMAL(18,2)	NOT NULL	Tổng tiền đơn hàng sau tính toán.
shipping_address	VARCHAR(255)	NOT NULL	Địa chỉ giao hàng.
customer_note	TEXT		Ghi chú của khách hàng.
payment_method_id	VARCHAR(50)	Khóa ngoại đến bảng	Phương thức thanh toán.

		PAYMENT METHOD	
order_status	ENUM('Pending', 'Processing','Shipping', 'Completed','Cancelled')	DEFAULT 'Pending'	Trạng thái đơn hàng.
is_order	BOOLEAN		Xác định có phải đơn hàng không
coupon_id	INT	Khóa ngoại đến bảng COUPONS	Mã giảm giá áp dụng (nếu có).
updated_at	DATETIME		Thời gian cập nhật gần nhất.

Bảng 3. 8 Mô tả bảng ORDER DETAILS

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
order_detail_id	VARCHAR(50)	PRIMARY KEY	Mã chi tiết đơn hàng.
order_id	VARCHAR(50)	Khóa ngoại đến bảng ORDER	Thuộc về đơn hàng nào.
product_id	VARCHAR(50)	Khóa ngoại đến bảng PRODUCT	Sản phẩm được mua.
quantity	INT	NOT NULL	Số lượng mua.
original_unit_price	DECIMAL(18,2)	NOT NULL	Đơn giá gốc tại thời điểm mua.

unit_price	DECIMAL(18,2)	NOT NULL	Đơn giá thực tế (sau giảm giá).
------------	---------------	----------	---------------------------------

Bảng 3. 9 Mô tả bảng PAYMENTS

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
payment_id	VARCHAR(50)	PRIMARY KEY	Mã giao dịch thanh toán.
order_id	VARCHAR(50)	Khóa ngoại đến bảng ORDER	Thanh toán cho đơn hàng nào.
payment_method_id	VARCHAR(50)	Khóa ngoại đến bảng PAYMENT METHOD	Phương thức thanh toán.
transaction_id	VARCHAR(100)		Mã giao dịch từ cổng thanh toán (nếu có).
payment_date	DATETIME		Thời gian thanh toán.
amount	DECIMAL(18,2)	NOT NULL	Số tiền đã thanh toán.

Bảng 3. 10 Mô tả bảng REVIEWS

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
review_id	INT	PRIMARY KEY	Mã đánh giá, tự động tăng
user_id	VARCHAR(50)	Khóa ngoại đến bảng USERS	Mã người dùng

product_id	VARCHAR(50)	Khóa ngoại đến bảng PRODUCTS	Mã sản phẩm
rating	INT		Số sao đánh giá
comment	TEXT		Nội dung đánh giá
created_at	DATETIME		Thời điểm tạo
updated_at	DATETIME		Thời điểm cập nhật

Bảng 3. 11 Mô tả bảng FLASH SALE

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
flash_sale_id	INT	PRIMARY KEY	Mã Flash Sale
name	VARCHAR(100)		Tên chương trình
description	VARCHAR(255)		Mô tả
start_date	DATETIME		Thời gian bắt đầu
end_date	DATETIME		Thời gian kết thúc
status	ENUM		Trạng thái chương trình
created_at	DATETIME		Ngày tạo

Bảng 3. 12 Mô tả bảng FLASH SALE ITEMS

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
flash_sale_item_id	INT	PRIMARY KEY	Mã chi tiết Flash Sale
flash_sale_id	INT	Khóa ngoại đến bảng USERS	Mã Flash Sale

product_id	VARCHAR(50)	Khóa ngoại đến bảng PRODUCTS	Mã sản phẩm
flash_sale_price	DECIMAL(15,2)		Giá Flash Sale
quantity	INT		Số lượng bán
sold_count	INT		Số lượng đã bán

3.5. Kiến trúc hệ thống và luồng xử lý Chatbot AI

3.5.1 Kiến trúc hệ thống

Quy trình xử lý dữ liệu diễn ra tuần tự theo các bước sau:

Gửi yêu cầu: Người dùng thao tác trên giao diện Client bằng trình duyệt web, hệ thống gửi một HTTP Request đến Server.

Bảo mật và cấu hình: Yêu cầu đi qua tầng Security và Config. Tại đây, hệ thống kiểm tra token của người dùng, xác thực quyền truy cập.

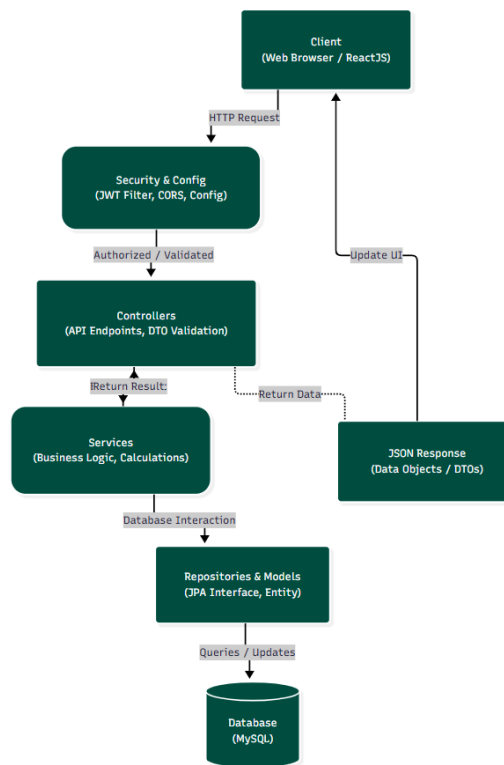
Tiếp nhận yêu cầu: Nếu hợp lệ, yêu cầu được chuyển đến Controller. Tại đây, các API Endpoint sẽ tiếp nhận và kiểm tra định dạng dữ liệu đầu vào mà người dùng gửi lên.

Xử lý nghiệp vụ: Controller gọi xuống tầng Services. Đây là nơi thực hiện các tính toán, logic chính của hệ thống.

Truy xuất dữ liệu: Service tương tác với tầng Repositories và Models. Tầng này sử dụng JPA Interface để ánh xạ các đối tượng thành câu lệnh SQL.

Tương tác Database: Các câu lệnh được gửi xuống Database để lấy hoặc lưu dữ liệu.

Phản hồi: Kết quả được trả ngược lại theo quy trình, cuối cùng Server trả về JSON Response để Client cập nhật giao diện



Hình 3. 7 Sơ đồ mô tả kiến trúc của hệ thống

3.5.2 Luồng xử lý Chatbot AI

Chatbot AI được thiết kế chuyên biệt để xử lý các truy vấn ngôn ngữ tự nhiên từ người dùng, tích hợp giữa dữ liệu nội bộ (MySQL) và trí tuệ nhân tạo (OpenAI API). Quy trình xử lý được mô tả chi tiết như sau:

- Các thành phần hệ thống:

Client: Giao diện người dùng, nơi khởi tạo yêu cầu và hiển thị phản hồi.

Spring Boot Backend: Toàn bộ logic xử lý tập trung trong khối này, bao gồm: Security và Config: Tầng lọc bảo mật và cấu hình hệ thống. ChatbotController: Lớp tiếp nhận yêu cầu điều hướng. Business Logic Layer: Lớp thực thi nghiệp vụ chính của chatbot.

Hệ thống bên ngoài: Tích hợp với OpenAI API để xử lý ngôn ngữ.

Database Layer: Sử dụng Repositories để giao tiếp với cơ sở dữ liệu MySQL.

- Quy trình xử lý dữ liệu tuần tự:

Luồng xử lý diễn ra khép kín từ lúc người dùng gửi câu hỏi cho đến khi nhận được phản hồi.

Khởi tạo yêu cầu: Người dùng gửi yêu cầu thông qua đối tượng ChatRequest (JSON) từ giao diện ReactJS đến Backend.

Xác thực: Yêu cầu đi qua tầng Security Config. Sau khi được xác thực (đã hợp lệ), dữ liệu được chuyển đến ChatbotController.

Điều phối nghiệp vụ: ChatbotController nhận ChatRequest và gọi xuống lớp ChatbotServiceImpl nằm trong tầng Business Logic.

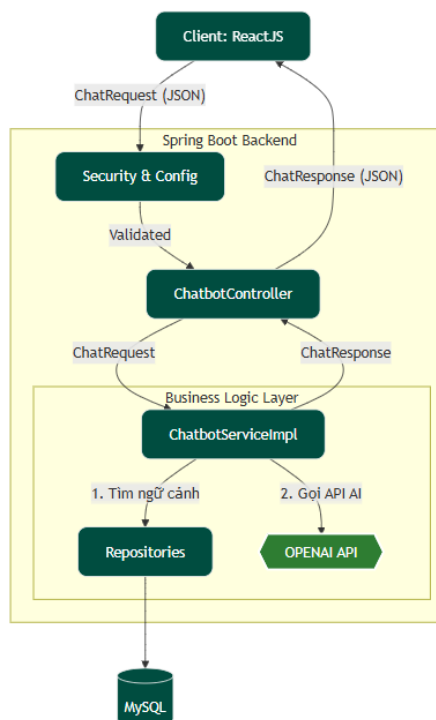
Xử lý nghiệp vụ phức hợp: Tại ChatbotServiceImpl, hệ thống thực hiện đồng thời ba nhiệm vụ quan trọng:

Tìm ngữ cảnh: Truy cập tầng Repositories để lấy dữ liệu sản phẩm hoặc thông tin liên quan từ MySQL nhằm làm giàu nội dung câu hỏi.

Gọi API AI: Gửi dữ liệu đã được xử lý (Prompt) đến OpenAI API để lấy câu trả lời thông minh.

Trả kết quả: Sau khi nhận phản hồi từ AI, Service đóng gói dữ liệu vào đối tượng ChatResponse gửi lại cho Controller.

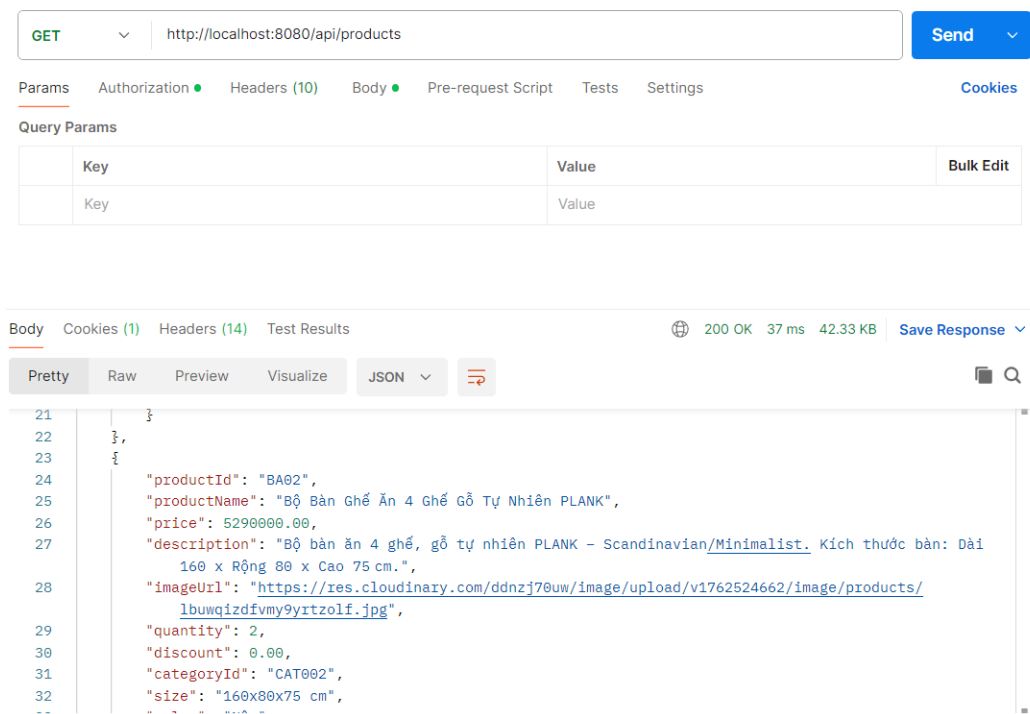
Cập nhật giao diện: Controller trả về ChatResponse cho Client. Giao diện ReactJS nhận dữ liệu và thực hiện Update UI để hiển thị câu trả lời cho người dùng.



Hình 3. 8 Mô tả luồng hoạt động của chatbot AI với website

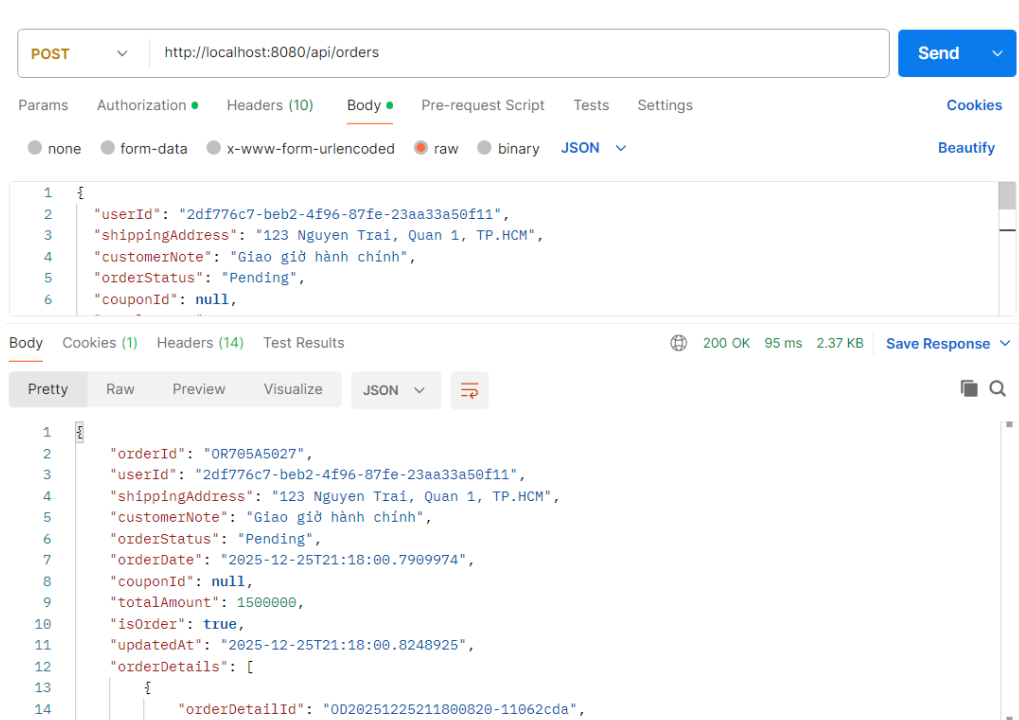
Xây dựng website bán nội thất tích hợp chatbot AI tư vấn khách hàng

API lấy danh sách sản phẩm: Khi nhận yêu cầu lấy toàn bộ danh sách sản phẩm, hệ thống trả về danh sách sản phẩm định dạng JSON chi tiết của các sản phẩm.



Hình 4. 3 Thử nghiệm API lấy danh sách sản phẩm với Postman

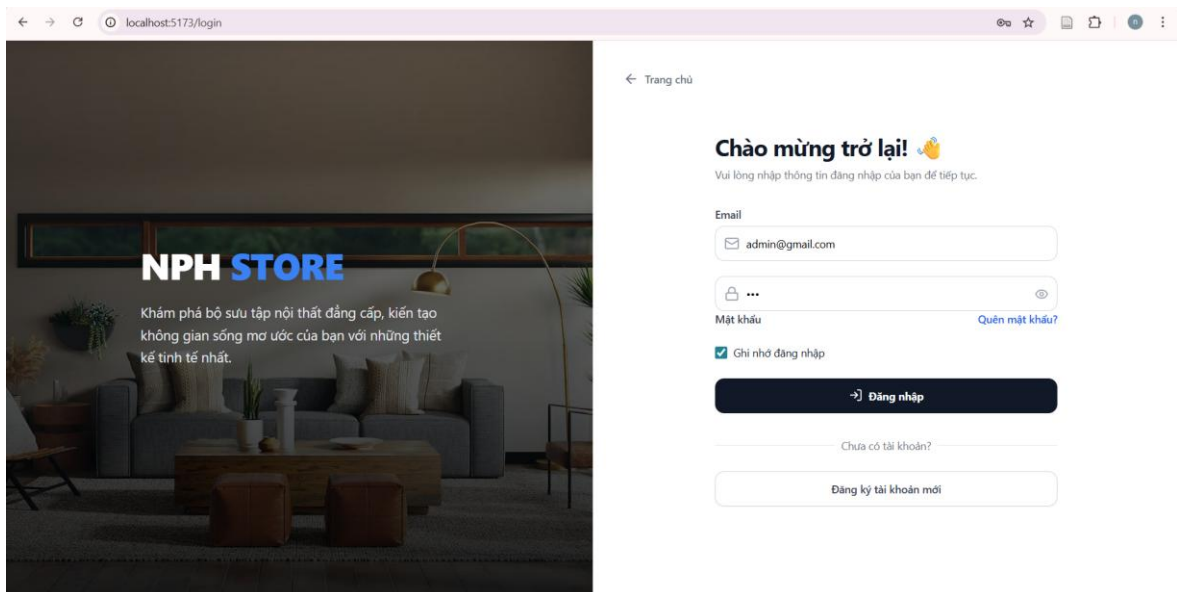
API đặt hàng: Thực hiện đặt hàng, gửi thông tin người mua và địa chỉ giao hàng, sau đó nhận lại phản hồi chứa mã đơn hàng và tổng giá trị thanh toán.



Hình 4. 4 Thử nghiệm API đặt hàng với Postman

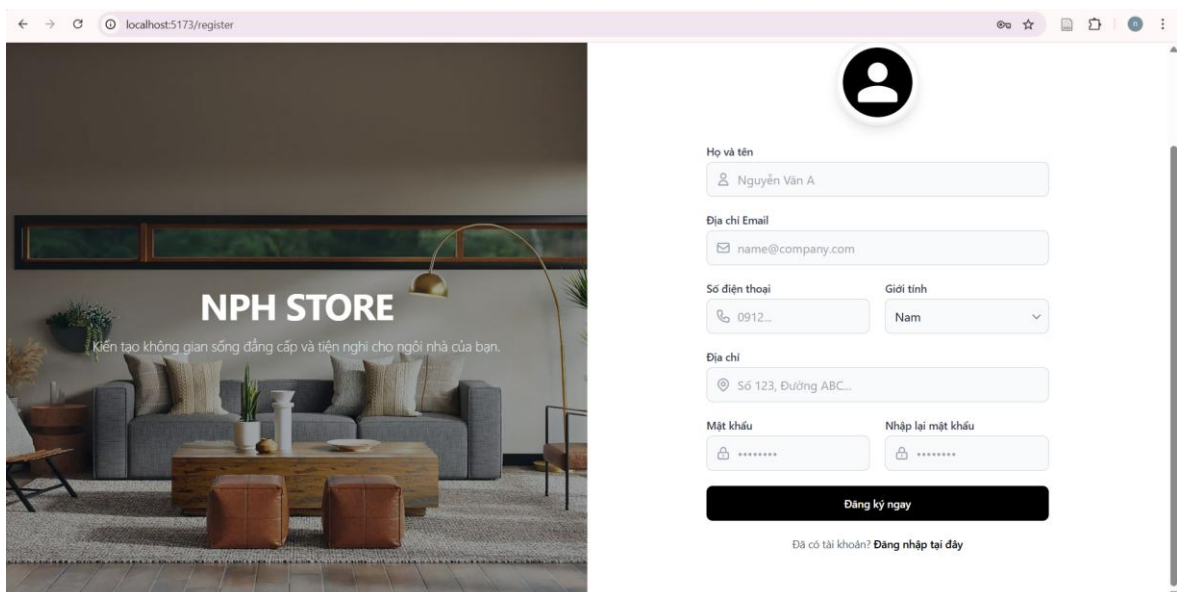
4.2. Giao diện người dùng

Giao diện đăng nhập: Người dùng có thể đăng nhập bằng tài khoản (nếu có).



Hình 4. 5 Giao diện đăng nhập

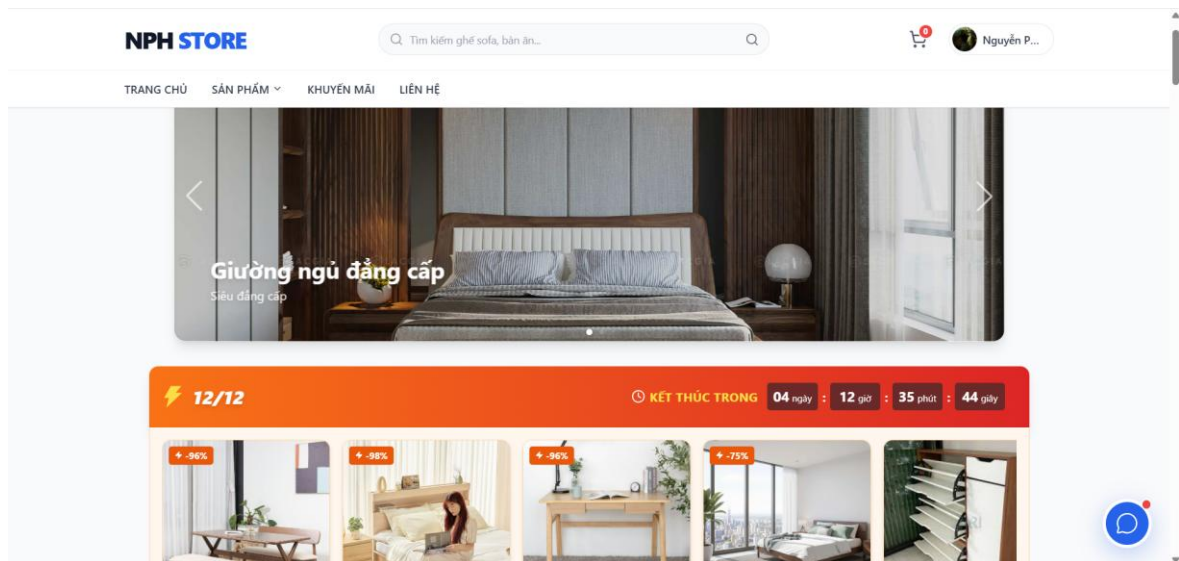
Giao diện đăng ký: Người dùng có thể đăng ký tài khoản của website và đăng nhập vào website với tài khoản đó.



Hình 4. 6 Giao diện đăng ký

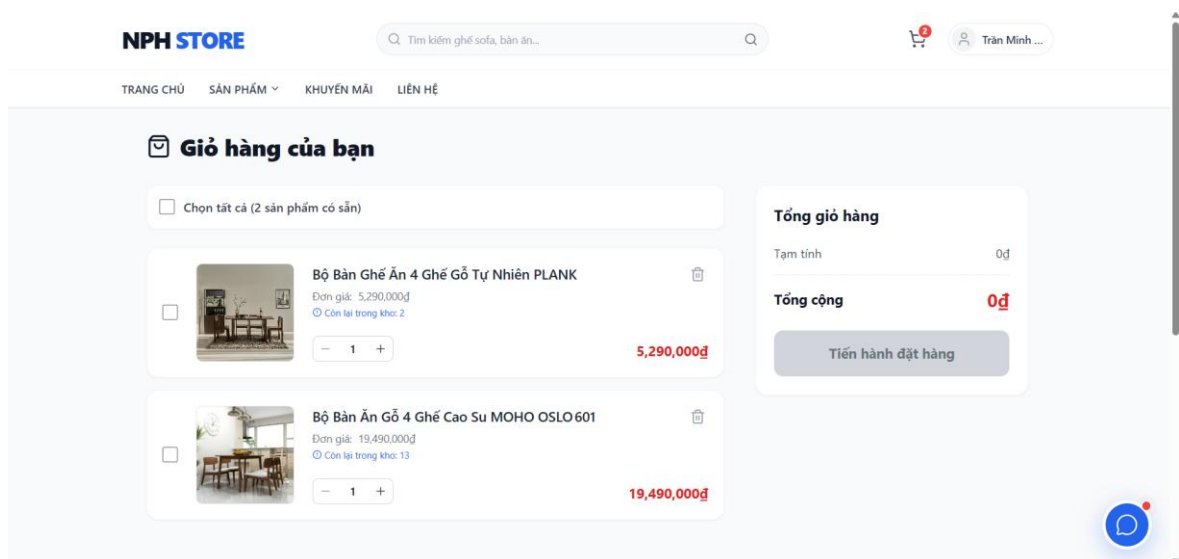
Xây dựng website bán nội thất tích hợp chatbot AI tư vấn khách hàng

Giao diện trang chủ: Khách hàng có thể xem các sản phẩm của website.



Hình 4. 7 Giao diện trang chủ

Giao diện trang giỏ hàng: Khách hàng có thể quản lý giỏ hàng như thêm, xóa, cập nhật số lượng sản phẩm trong giỏ hàng và chọn sản phẩm để đặt hàng.



Hình 4. 8 Giao diện trang giỏ hàng

Xây dựng website bán nội thất tích hợp chatbot AI tư vấn khách hàng

Giao diện trang đặt hàng: Khách hàng có thể chỉnh sửa thông tin nhận hàng, áp dụng mã giảm giá, thêm ghi chú và chọn phương thức thanh toán để thực hiện đặt hàng.

The screenshot displays the checkout interface of NPH STORE. At the top, there's a search bar and a user profile icon. Below the navigation bar, the checkout process is divided into several sections:

- Địa chỉ nhận hàng:** A field for the delivery address, currently showing "Trần Minh Quân | 0905123" and "Hồ Chí Minh".
- Sản phẩm (2):** A list of items being purchased:
 - Bộ Bàn Ghế Ăn 4 Ghế Gỗ Tự Nhiên PLANK, Số lượng: x1, Đơn giá: 5,290,000 đ, Total: 5,290,000 đ.
 - Bộ Bàn Ăn Gỗ 4 Ghế Cao Su MOHO OSLO 601, Số lượng: x1, Đơn giá: 19,490,000 đ, Total: 19,490,000 đ.
- Phương thức thanh toán:** Two options are available: "Thanh toán khi nhận hàng (COD)" and "VN Pay".
- Tổng quan đơn hàng:** A summary section on the right showing the total amount: 24,780,000 đ, with a red "Đặt hàng" button.

Hình 4. 9 Giao diện trang đặt hàng

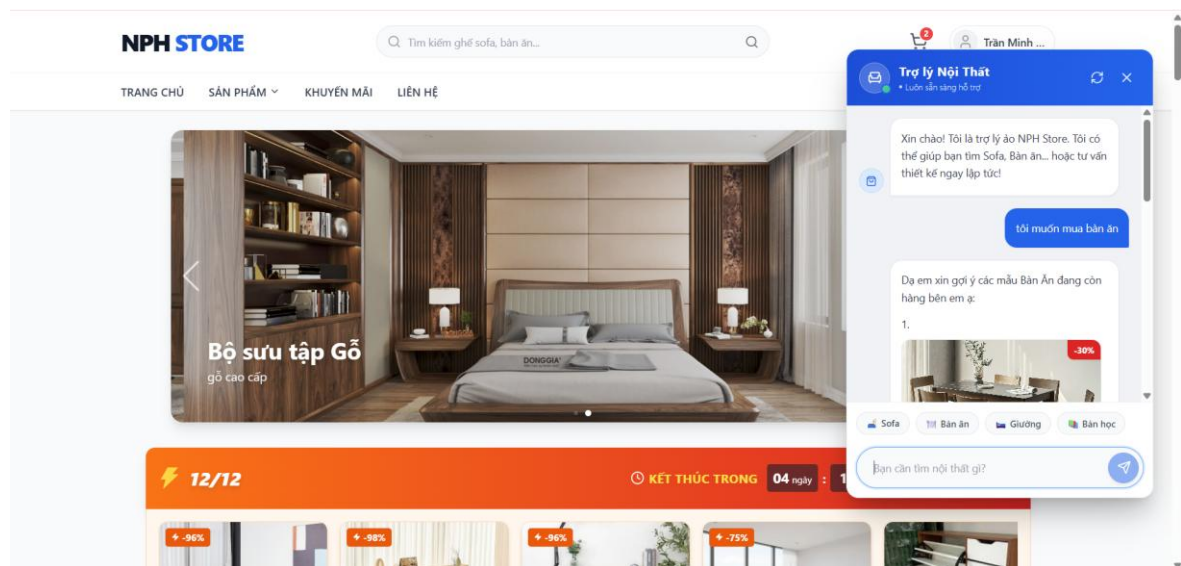
Giao diện trang đơn hàng của tôi: Khách hàng có thể xem lại các đơn hàng đã đặt và xem lại thông tin về đơn hàng đó.

The screenshot shows the "Đơn hàng của tôi" (My Orders) section of the NPH STORE website. The interface includes a navigation bar with tabs for "Tất cả", "Chờ xác nhận", "Đang xử lý", "Đang giao", "Hoàn thành", and "Đã hủy". Below the tabs, there's a list of orders:

- Order 1:** Mã đơn: #ORF908A279, 07:50:08 25/12/2025, Status: GIAO THÀNH CÔNG. Item: Bàn Làm Việc Gỗ Sồi 2 Ngăn Kéo, Số lượng: 2, Price: 247,356đ.
- Order 2:** Mã đơn: #OR0E4E4396, 09:26:34 20/12/2025, Status: GIAO THÀNH CÔNG. Item: Bộ Bàn Ăn Gỗ 4 Ghế - 6 Ghế SCANIA, Số lượng: 1, Price: 3,990,000đ.

Hình 4. 10 Giao diện trang đơn hàng của tôi

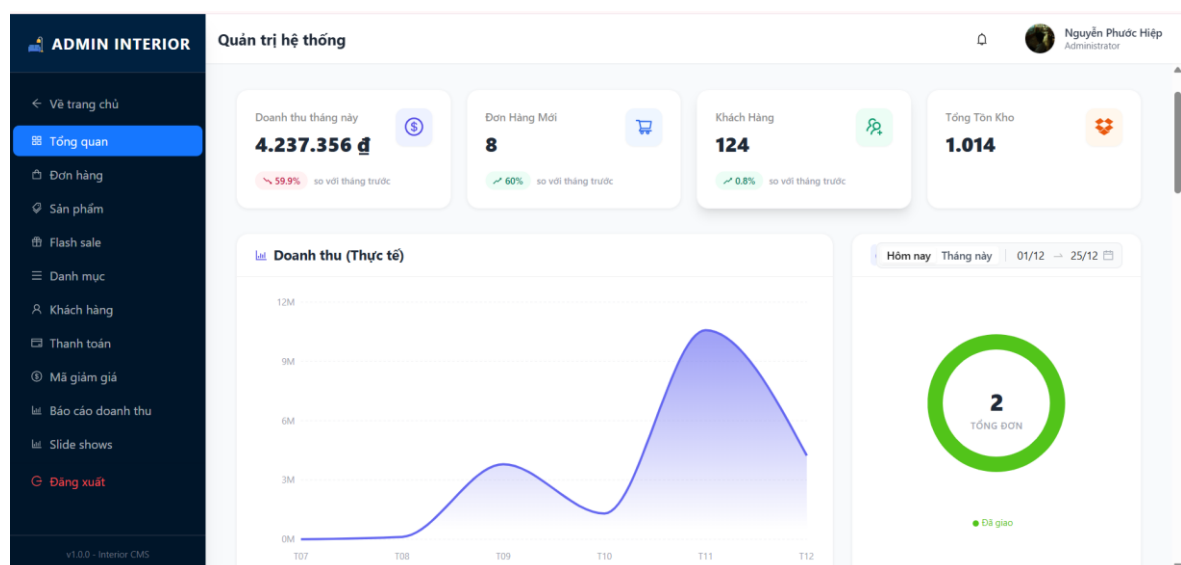
Giao diện trò chuyện với chatbot AI: Khách hàng có thể hỏi các câu hỏi về sản phẩm trong website để chatbot tư vấn chọn ra sản phẩm phù hợp.



Hình 4. 11 Giao diện trò chuyện với chatbot AI tư vấn

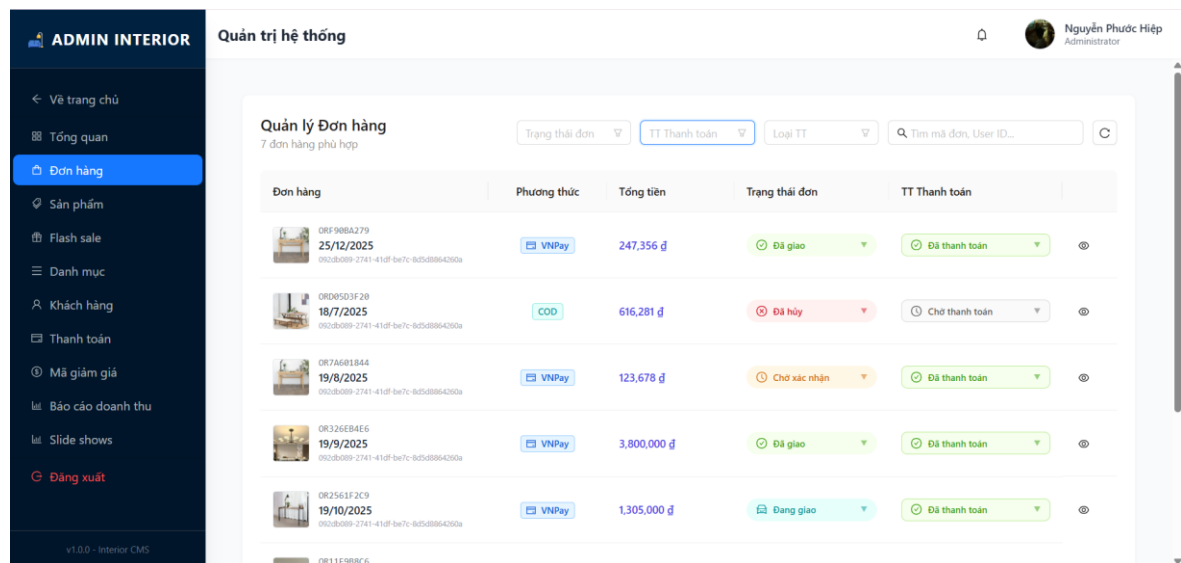
4.3. Giao diện quản trị hệ thống

Giao diện Admin Dashboard: Admin có thể xem thống kê tổng quan về website.



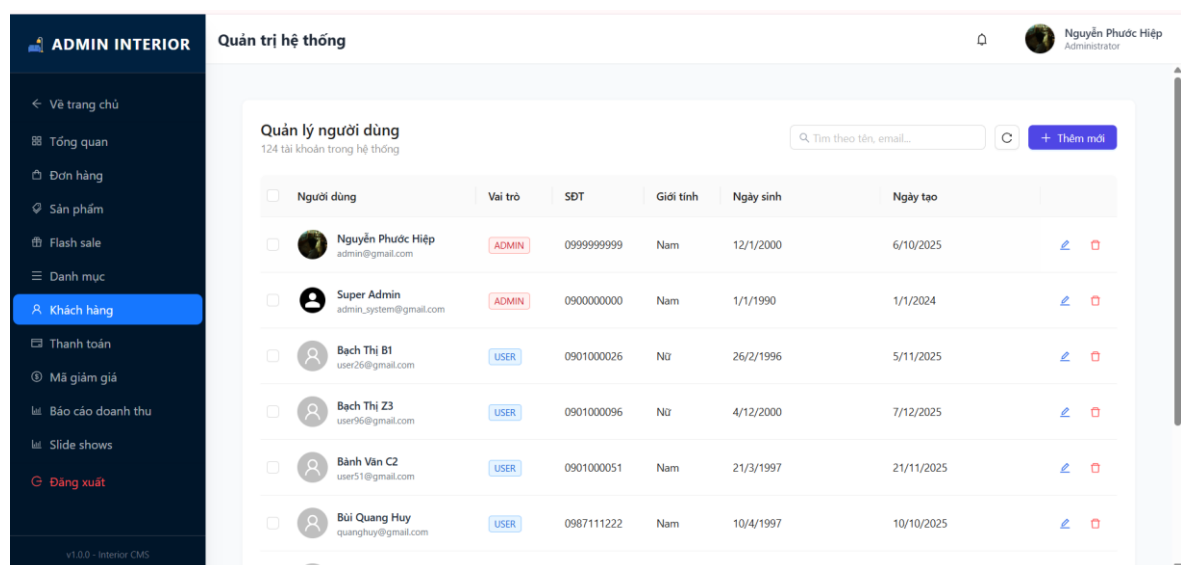
Hình 4. 12 Giao diện trang Admin Dashboard

Trang quản lý đơn hàng: Admin có thể xem danh sách đơn hàng, cập nhật trạng thái vận chuyển và trạng thái thanh toán của đơn hàng.



Hình 4. 13 Giao diện trang quản lý đơn hàng

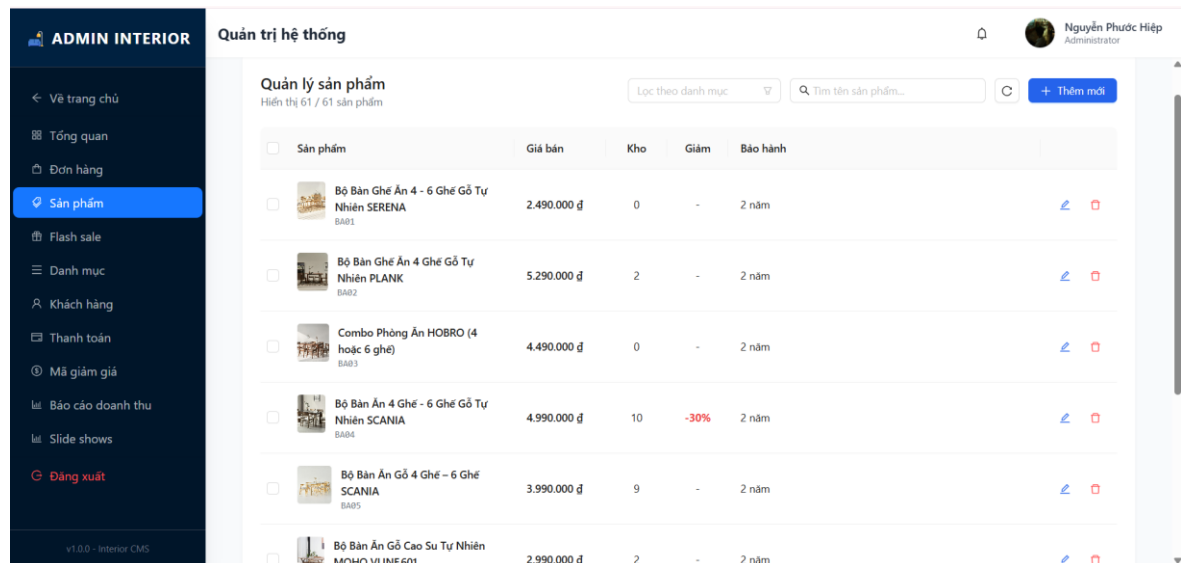
Trang quản lý người dùng: Admin có thể quản lý toàn bộ người dùng của website gồm các thao tác thêm, xóa, sửa người dùng.



Hình 4. 14 Giao diện trang quản lý người dùng

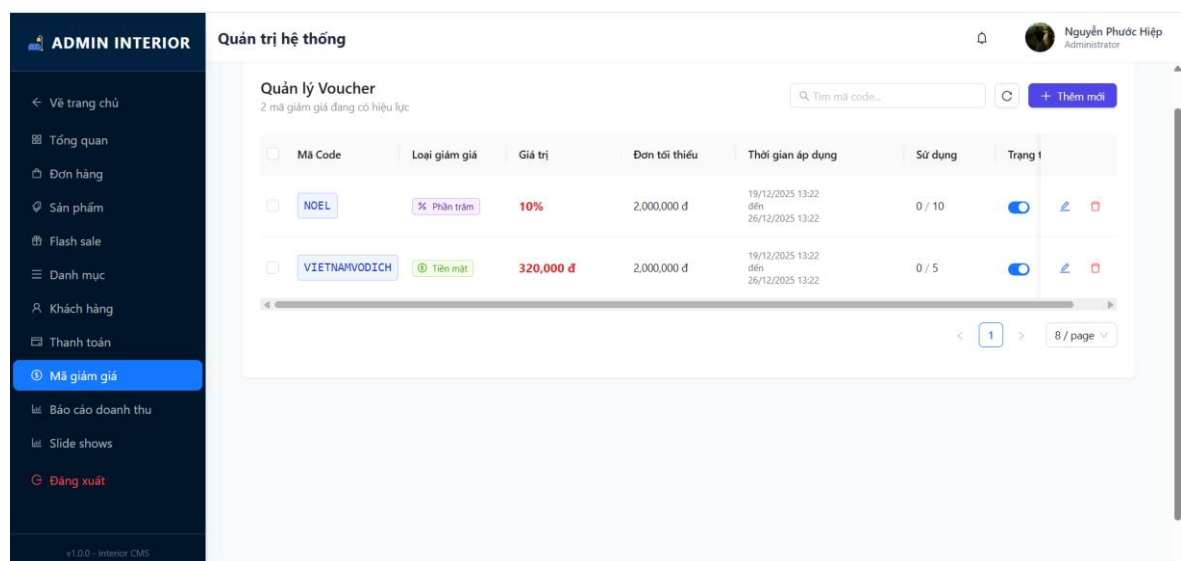
Xây dựng website bán nội thất tích hợp chatbot AI tư vấn khách hàng

Trang quản lý sản phẩm: Admin có thể cập nhật số lượng sản phẩm, giảm giá và thông tin chi tiết của sản phẩm.



Hình 4. 15 Giao diện trang quản lý sản phẩm

Trang quản lý mã giảm giá: Admin có quyền tạo mới, thay đổi điều kiện áp dụng và thời gian áp dụng mã giảm giá.



Hình 4. 16 Giao diện trang quản lý mã giảm giá

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết quả đạt được

- Xây dựng thành công website bán nội thất sử dụng ReactJS, Tailwind CSS và Spring boot.
- Tích hợp thành công cổng thanh toán VNPay, cho phép khách hàng có thể thanh toán trực tuyến.
- Tích hợp thành công chatbot AI tư vấn khách hàng sử dụng OPENAI API.
- Website được thiết kế với giao diện Responsive, đảm bảo hiển thị tối ưu và hoạt động ổn định trên nhiều loại thiết bị khác nhau.

5.2. Hướng phát triển

Mặc dù hệ thống đã đáp ứng được các yêu cầu cơ bản của một website thương mại điện tử nội thất, trong tương lai, đề tài có thể được mở rộng và phát triển theo các hướng sau:

- Tích hợp tính năng tìm kiếm sản phẩm thông qua hình ảnh bằng AI.
- Tích hợp công nghệ thực tế tăng cường (AR) cho phép người dùngướm thử nội thất vào không gian thực tế của căn phòng trước khi quyết định mua.
- Đa dạng hóa các phương thức thanh toán điện tử khác như ví MoMo, ShopeePay hoặc thẻ quốc tế.
- Xây dựng hệ thống gợi ý sản phẩm thông minh dựa trên lịch sử xem hàng và sở thích của từng khách hàng.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Amazon Web Services, Inc., “What is RESTful API? - RESTful API Explained - AWS.” [Online]. Available: <https://aws.amazon.com/what-is/restful-api/>. [Accessed: Dec. 26, 2025].
- [2] 200Lab Blog, “Vite là gì? Hướng dẫn Khởi tạo dự án React với Vite.” [Online]. Available: <https://200lab.io/blog/vite-la-gi>. [Accessed: Dec. 12, 2025].
- [3] CareerLink, “Tailwind CSS Là Gì? Ưu Nhược điểm Và ứng Dụng Trong Thực Tế.” [Online]. Available: <https://www.careerlink.vn/cam-nang-viec-lam/tu-van-nghe-nghiep/tailwind-css-la-gi>. [Accessed: Dec. 12, 2025].
- [4] C. L. V. Tiến, “MySQL là gì? So sánh MySQL và SQL | Cài đặt, sử dụng cơ bản,” *Vietnix*. [Online]. Available: <https://vietnix.vn/mysql-la-gi/>. [Accessed: Dec. 12, 2025].
- [5] TopDev, “Spring boot là gì? Tính năng nổi bật và cách sử dụng Spring Boot.” [Online]. Available: <https://topdev.vn/blog/gioi-thieu-ve-spring-boot-spring-boot-la-gi/>. [Accessed: Dec. 12, 2025].
- [6] N. T. K. Ngân, “BÁO CÁO THỰC TẬP TỐT NGHIỆP: XÂY DỰNG WEBSITE BÁN ĐỒ NỘI THẤT,” *Studocu*. [Online]. Available: <https://www.studocu.vn/vn/document/hoc-vien-cong-nghe-buu-chinh-vien-thong/lap-trinh-web/bao-cao-thuc-tap-tot-nghiep-xay-dung-website-ban-do-noi-that/145226725>. [Accessed: Dec. 26, 2025].
- [7] N. T. Hà, “Xây dựng Website bán đồ nội thất cho công ty nội thất Sofia.” [Online]. Available: <https://khotrithucso.com/doc/p/xay-dung-website-ban-do-noi-that-cho-cong-ty-noi-that-sofia-1540503>. [Accessed: Dec. 26, 2025].
- [8] T. T. Lân, “Báo cáo thực tập - Xây dựng Website kinh doanh đồ nội thất.doc.” [Online]. Available: <https://www.slideshare.net/slideshow/bo-co-thc-tp-xy-dng-website-kinh-doanh-ni-thtdoc/260243840>. [Accessed: Dec. 26, 2025].
- [9] Đ. T. T. Xuân, “Tóm tắt Đồ án Xây dựng website kinh doanh nội thất.” [Online]. Available: <https://luanvan.org/tom-tat-do-an-xay-dung-website-kinh-doanh-noi-that-3879/>. [Accessed: Dec. 26, 2025].
- [10] N. T. Tùng, “Báo cáo xây dựng trang web quảng cáo sản phẩm nội thất của một công ty,” *SlideShare*, 2023. [Online]. Available: <https://www.slideshare.net/slideshow/bo-co-xy-dng-trang-web-qung-co-sn-phm-ni-tht-ca-mt-cng-tydoc/261671881>. [Accessed: Dec. 26, 2025].

PHỤ LỤC