

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:
XÂY DỰNG WEBSITE MUA BÁN COFFEE
CHO CÔNG TY TASTETHECOFFEE

Người hướng dẫn: **TS.Lý Quỳnh Trân**

Sinh viên thực hiện: **Phan Phước Lợi**

Mã số sinh viên: 2151220170

Lớp: **21CT4**

Đà Nẵng, 12/2025

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:
XÂY DỰNG WEBSITE MUA BÁN COFFEE
CHO CÔNG TY TASTETHECOFFEE

Người hướng dẫn: : TS.Lý Quỳnh Trân

Sinh viên thực hiện: **PHAN PHƯỚC LỢI**

Mã số sinh viên: 2151220170

Lớp: **21CT4**

Đà Nẵng, 12/2025

TÓM TẮT

Tên đề tài: Xây dựng Website mua bán Coffee cho công ty TasteTheCoffee

Sinh viên thực hiện: Phan Phước Lợi

Mã số sinh viên : 2151220170 Lớp : 21CT4

Trong bối cảnh chuyển đổi số mạnh mẽ hiện nay, thương mại điện tử đã và đang trở thành xu hướng tất yếu trong nhiều lĩnh vực, đặc biệt là ngành kinh doanh coffee. Đề tài “Xây dựng website mua bán Coffee cho công ty TasteTheCoffee” được thực hiện nhằm nghiên cứu, ứng dụng các công nghệ web hiện đại để xây dựng một nền tảng mua sắm trực tuyến tiện lợi, hiệu quả và thân thiện với người dùng.

Website không chỉ hỗ trợ khách hàng dễ dàng tìm kiếm, lựa chọn và đặt mua sản phẩm coffee, mà còn cung cấp các chức năng quản lý danh mục sản phẩm, giỏ hàng, đặt hàng và thanh toán. Đồng thời, hệ thống còn trang bị các công cụ quản trị dành cho công ty để quản lý kho hàng, đơn hàng và khách hàng, đáp ứng nhu cầu thực tế trong hoạt động kinh doanh và mở rộng thị trường.

LỜI NÓI ĐẦU

Trong thời gian làm đồ án tốt nghiệp, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè.

Em xin chân thành cảm ơn các thầy cô giáo trong trường ĐH Kiến Trúc Đà Nẵng nói chung, các thầy cô trong khoa công nghệ thông tin nói riêng đã dạy dỗ cho em kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ em trong suốt quá trình học tập.

Đặc biệt em xin gửi lời cảm ơn đến cô giáo Lý Quỳnh Trân đã tận tình giúp đỡ, chỉ bảo, hướng dẫn em trong suốt quá trình thực hiện đồ án. Trong thời gian qua, em đã không ngừng tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần làm việc, thái độ nghiên cứu khoa học nghiêm túc, hiệu quả, đây là những điều rất cần thiết cho em trong quá trình học tập và công tác sau này.

Cuối cùng, em xin chân thành cảm ơn gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên em trong suốt quá trình học tập và hoàn thành đồ án tốt nghiệp.

Em xin trân trọng cảm ơn !

LỜI CAM ĐOAN

Em xin cam đoan đồ án tốt nghiệp " Xây dựng Website mua bán Coffee cho công ty TasteTheCoffee" là công trình nghiên cứu của riêng cá nhân em, được thực hiện dưới sự hướng dẫn khoa học của TS.Lý Quỳnh Trân.

Các kết quả, số liệu nêu trong đồ án là trung thực. Những phần sử dụng tài liệu tham khảo hay kế thừa từ các nguồn khác đều được trích dẫn và chú thích rõ ràng theo quy định. Em không sao chép trái phép công trình của bất kỳ ai khác.

Em xin hoàn toàn chịu trách nhiệm trước Hội đồng bảo vệ và Nhà trường về sự trung thực của lời cam đoan này.

Sinh viên thực hiện

PHAN PHƯỚC LỢI

MỤC LỤC

TÓM TẮT	i
LỜI NÓI ĐẦU	ii
LỜI CAM ĐOAN	iii
DANH MỤC CÁC BẢNG	ix
DANH SÁCH CÁC KÍ HIỆU CHỮ VIẾT TẮT	xi

Trang

MỞ ĐẦU	1
1. Mục tiêu đề tài	1
2. Đối tượng, phạm vi nghiên cứu	2
2.1. Đối tượng nghiên cứu	2
2.2. Phạm vi nghiên cứu	2
3. Phương pháp nghiên cứu	3
4. Giải pháp công nghệ	3
5. Cấu trúc đồ án	4
1.1. Lý do chọn đề tài	5
1.2. Lịch sử nghiên cứu	6
1.2.1. Lịch sử nghiên cứu trong nước:	6
1.2.2. Lịch sử nghiên cứu ngoài nước:	7
1.3. Mục tiêu đề tài	7
1.3.1 Mục tiêu cụ thể	7
1.3.2 Mục tiêu ứng dụng công nghệ	8
1.4 Đối tượng phạm vi nghiên cứu	9
1.4.1. Phạm vi nội dung	9
1.4.2. Phạm vi không gian	10

2.1 Giới thiệu về ngôn ngữ JavaScript	11
2.1.1.Khái niệm	11
2.1.2.Uu điểm và nhược điểm	11
2.1.3. Tính năng của JavaScript	12
2.2 Giới thiệu về ReactJs	12
2.2.1.Khái niệm	12
2.2.2. Tính năng của ReactJs	14
2.3 Giới thiệu về Monggo	14
2.3.1.Khái niệm	14
2.3.2. Tính năng của MongoDB	15
2.4 Giới thiệu về NodeJs	15
2.4.1.Khái niệm	15
2.4.2. Tính năng của NodeJS	16
2.5. Giới thiệu ExpressJS	16
2.5.1. . Khái niệm	16
2.5.2. Tính năng của ExpressJS	17
CHƯƠNG 3 : PHÂN TÍCH THUYẾT KẾ HỆ THỐNG	18
3.1. Xác định yêu cầu xây dựng website mua bán Coffee	18
3.1.1. Yêu cầu chức năng	18
3.1.2. Yêu cầu phi chức năng	19
3.1.3. Yêu cầu hệ thống	19
3.2.Phân tích yêu cầu :	19
3.2.1 Yêu cầu chức năng chi tiết	19
3.2.2 Mô hình hóa yêu cầu	24
3.3.1 Chức năng đăng kí	26
3.3.2. Chức năng đăng nhập	28
3.3.3. Chức năng mua hàng	31

3.3.4. Chức năng tìm kiếm	35
3.3.5. Chức năng xem hàng đã đặt.....	37
3.3.6. Chức năng thanh toán	41
3.3.7. Chức năng quản lí đặt hàng	43
3.3.8. Chức năng quản lí hàng	47
3.3.9. Chức năng quản lí khách hàng	50
3.4. Thiết kế cơ sở dữ liệu	52
3.4.1. Mô hình thực thể liên kết.....	52
3.4.2 Mô hình quan hệ.....	52
3.4.3. Mô hình lớp	53
3.5 Danh mục các bảng.....	53
CHƯƠNG 4 : CÀI ĐẶT DEMO GIAO DIỆN	63
4.1 Giao diện người dùng.....	63
4.2. Giao diện admin	67
4.3. Cấu trúc thư mục code	72
KẾT LUẬN.....	74
TÀI LIỆU THAM KHẢO	76

DANH MỤC HÌNH VẼ

Hình 3.1 Usecase tổng quát	25
Hình 3.2 Biểu đồ Use Case đăng ký	26
Hình 3.3 Biểu đồ tuần tự đăng kí.....	28
Hình 3.4. Biểu đồ Use Case đăng nhập	28
Hình 3.5 Biểu đồ tuần tự đăng nhập	31
Hình 3.6 Biểu đồ Use Case đặt hàng	31
Hình 3.7 Biểu đồ tuần tự đặt hàng	34
Hình 3.8 Biểu đồ use case tìm kiếm	35
Hình 3.9 Biểu đồ tuần tự tìm kiếm	37
Hình 3.10 Biểu đồ use case hàng đã đặt	37
Hình 3.11 Biểu đồ tuần tự xem hàng đã đặt	40
Hình 3.12 Biểu đồ use case thanh toán.....	41
Hình 3.13 Biểu đồ tuần tự thanh toán.....	43
Hình 3.14 Biểu đồ use case quản lí đặt hàng.....	43
Hình 3.15. Biểu đồ tuần tự quản lí đặt hàng.....	46
Hình 3.16 Biểu đồ use case quản lí hàng.....	47
Hình 3.17. Biểu đồ trình tự quản lí hàng	49
Hình 3.18 Biểu đồ use case quản lí khách hàng	50
Hình 3.19 Biểu đồ tuần tự quản lí khác hàng	51
Hình 3.20 Mô hình thực thể liên kết.....	52
Hình 3.21 Mô hình quan hệ	52
Hình 3.22 Mô hình lớp.....	53
Hình 4.1 Giao diện đăng nhập	63
Hình 4.2 Giao diện đăng kí.....	63
Hình 4.3 Màn hình trang chủ	64

Hình 4.4 Giao diện sản phẩm	64
Hình 4.5 Giao diện giỏ hàng.....	65
Hình 4.6 Giao diện thanh toán.....	65
Hình 4.7 Giao diện lịch sử đơn hàng	66
Hình 4.8 Giao diện chat bot AI	66
Hình 4.9 Giao diện màn hình chính trang admin.....	67
Hình 4.10 Giao diện quản lí người dùng trang admin	67
Hình 4.11 Giao diện thêm sản phẩm.....	68
Hình 4.12 Giao diện quản lí gia công coffee	68
Hình 4.13 Giao diện quản lí đơn hàng.....	69
Hình 4.14 Giao diện quản lí thua mua coffee	69
Hình 4.15 Giao diện quản lí mã khuyến mãi	70
Hình 4.16 Giao diện quản lí Blog.....	70
Hình 4.17 Giao diện quản lý đánh giá	71
Hình 4.18 Cấu trúc thư mục frontend.....	72
Hình 4.19 Cấu trúc thư mục backend	73

DANH MỤC CÁC BẢNG

Bảng 3.1 Yêu cầu chức năng chi tiết khách hàng	20
Bảng 3.3. Yêu cầu phi chức năng	23
Bảng 3.4 Mô tả user đăng kí	26
Bảng 3.5 Mô tả Use Case đăng kí	27
Bảng 3.6 Mô tả Use Case đăng nhập	29
Bảng 3.7 Mô tả dòng chính Use case đăng nhập	30
Bảng 3.8 Mô tả Bảng Use case đặt hàng	32
Bảng 3.9 Mô tả dòng chính Use case đặt hàng	33
Bảng 3.10 Mô tả use case tìm kiếm	35
Bảng 3.11 Mô tả dòng chính use case: tìm kiếm	36
Bảng 3.12 Mô tả dòng chính use case xem hàng đã đặt	39
Bảng 3.13 Use case thanh toán	41
Bảng 3.14 Mô tả dòng chính use case thanh toán	42
Bảng 3.15 Use case quản lí đặt hàng	44
Bảng 3.16 Mô tả dòng chính use case quản lí đặt hàng	45
Bảng 3.17 Use case quản lí đặt hàng	47
Bảng 3.18 Mô tả dòng chính use case quản lí hàng	48
Bảng 3.19 Use case quản lí khách hàng	50
Bảng 3.20 Mô tả dòng chính use case quản lý khác hàng	51
Bảng 3.21 Bảng cơ sở dữ liệu	53
Bảng 3.22 Bảng mô tả user	54
Bảng 3.23 Bảng mô tả Product	55
Bảng 3.24 Bảng mô tả Order	56
Bảng 3.25 Bảng mô tả PromoCode	57
Bảng 3.26 Bảng mô tả Processing	58

Bảng 3.27 Bảng mô tả purchase	59
Bảng 3.28 Bảng mô tả Blog.....	59
Bảng 3.29 Bảng mô tả ChatHistory	60
Bảng 3.30 Bảng mô tả REVIEW	62

DANH SÁCH CÁC KÍ HIỆU CHỮ VIẾT TẮT

Từ viết tắt	Tiếng Anh	Tiếng Việt
API	Application Programming Interface	Giao diện lập trình ứng dụng
UI	User Interface	Giao diện người dùng
UX	User Experience	Trải nghiệm người dùng
DB	Database	Cơ sở dữ liệu
NoSQL	Not Only SQL	Cơ sở dữ liệu phi quan hệ
HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
HTTPS	HyperText Transfer Protocol Secure	Giao thức truyền tải siêu văn bản bảo mật
JWT	JSON Web Token	Chuẩn giao thức xác thực dạng token
CRUD	Create – Read – Update – Delete	Các thao tác với dữ liệu
MVC	Model – View – Controller	Mô hình kiến trúc phần mềm
REST	Representational State Transfer	Kiến trúc API nhẹ
ID	Identification	Mã định danh
UID	User ID	Mã định danh người dùng
PID	Product ID	Mã định danh sản phẩm
CID	Category ID	Mã định danh danh mục
OID	Order ID	Mã định danh đơn hàng
COD	Cash On Delivery	Thanh toán khi nhận hàng
OTP	One Time Password	Mã xác thực một lần
QR	Quick Response	Mã phản hồi nhanh

JSON	JavaScript Object Notation	Định dạng dữ liệu dạng đối tượng
JS	JavaScript	Ngôn ngữ lập trình phía client & server
CSS	Cascading Style Sheets	Ngôn ngữ thiết kế giao diện
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
SEO	Search Engine Optimization	Tối ưu hóa công cụ tìm kiếm
URL	Uniform Resource Locator	Đường dẫn tài nguyên
SSL	Secure Sockets Layer	Lớp bảo mật kết nối
TLS	Transport Layer Security	Bảo mật tầng truyền tải
CPU	Central Processing Unit	Bộ xử lý trung tâm
RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên

MỞ ĐẦU

Ngày nay, cùng với sự phát triển mạnh mẽ của công nghiệp hóa – hiện đại hóa, thương mại điện tử đang trở thành xu hướng tất yếu trong hoạt động kinh doanh. Việc ứng dụng công nghệ thông tin vào bán hàng giúp các doanh nghiệp tiết kiệm thời gian, chi phí và nâng cao hiệu quả quản lý.

Đối với các công ty kinh doanh coffee như TasteTheCoffee, việc quản lý sản phẩm, đơn hàng, khách hàng hay doanh thu nếu thực hiện bằng phương pháp thủ công sẽ mất nhiều thời gian, dễ xảy ra sai sót và không đáp ứng được nhu cầu mở rộng kinh doanh. Bên cạnh đó, công ty còn phải tốn nhiều chi phí cho hoạt động kiểm kê, báo cáo và quảng bá sản phẩm mà hiệu quả mang lại chưa tối ưu. Đặc biệt, trong bối cảnh người tiêu dùng ngày càng quan tâm đến chất lượng và nguồn gốc coffee, việc cung cấp thông tin chi tiết về sản phẩm, tư vấn công thức pha chế và hỗ trợ khách hàng một cách chuyên nghiệp trở thành yếu tố quan trọng để tạo dựng niềm tin và nâng cao trải nghiệm mua sắm.

Từ thực tế đó, em tiến hành xây dựng website mua bán coffee cho công ty TasteTheCoffee với mục tiêu hỗ trợ số hóa quy trình kinh doanh, giúp công ty dễ dàng quản lý sản phẩm, theo dõi doanh thu, tiếp nhận đơn hàng và chăm sóc khách hàng. Đồng thời, website còn đóng vai trò như một kênh trưng bày và quảng bá sản phẩm, tạo cơ hội mở rộng thị trường và nâng cao năng lực cạnh tranh. Đặc biệt, hệ thống tích hợp AI Chatbot sử dụng Google Gemini để tư vấn công thức pha coffee, gợi ý sản phẩm phù hợp với sở thích người dùng, mang lại trải nghiệm mua sắm thông minh và hiện đại.

1. Mục tiêu đề tài

Mục tiêu chính của dự án là xây dựng một hệ thống website mua bán coffee cho công ty TasteTheCoffee, giúp tạo ra một kênh bán hàng trực tuyến hiện đại, tiện lợi và đáng tin cậy. Hệ thống hướng đến việc mang lại trải nghiệm mua sắm thân thiện, dễ sử dụng cho khách hàng, đồng thời hỗ trợ công ty quản lý hoạt động kinh doanh một cách hiệu quả. Thông qua dự án này, TasteTheCoffee không chỉ tập trung đáp ứng nhu cầu mua sắm ngày càng cao của khách hàng mà còn hướng đến việc khẳng định thương hiệu và nâng cao khả năng cạnh tranh trên thị trường.

2. Đối tượng, phạm vi nghiên cứu

2.1. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài là hệ thống website mua bán coffee phục vụ hoạt động kinh doanh của công ty TasteTheCoffee, bao gồm các chức năng quản lý sản phẩm, đơn hàng, khách hàng, mã khuyến mãi, blog, đánh giá và giao diện mua sắm dành cho người dùng. Hệ thống còn tích hợp AI Chatbot để tư vấn công thức pha coffee và gợi ý sản phẩm, cùng các dịch vụ hỗ trợ như thu mua và gia công coffee.

2.2. Phạm vi nghiên cứu

Website được thiết kế nhằm hỗ trợ hoạt động mua bán coffee của công ty TasteTheCoffee. Phạm vi bao gồm các chức năng:

Đối với người dùng:

Đăng ký/Đăng nhập bằng Google Oauth.

Duyệt và tìm kiếm sản phẩm coffee theo danh mục (coffee nguyên chất, coffee tiện lợi, coffee truyền thống).

Xem chi tiết sản phẩm với hình ảnh, mô tả, đánh giá.

Quản lý giỏ hàng và đặt hàng với nhiều phương thức thanh toán (COD, PayPal, MoMo).

Áp dụng mã khuyến mãi.

Xem lịch sử đơn hàng, hủy đơn hàng và xem thông tin hoàn tiền.

Đánh giá và review sản phẩm sau khi nhận hàng.

Tương tác với AI Chatbot để được tư vấn công thức pha coffee.

Đọc blog về coffee.

Xem thông tin dịch vụ thu mua và gia công coffee.

Đối với quản trị viên:

Quản lý người dùng (xem, sửa, xóa).

Quản lý sản phẩm (thêm, sửa, xóa, cập nhật giá, hình ảnh, tồn kho)

Quản lý đơn hàng (xem, cập nhật trạng thái, xem thông tin hoàn tiền)

Quản lý mã khuyến mãi (tạo, sửa, xóa, theo dõi sử dụng)

Quản lý dịch vụ gia công và thu mua coffee.

Quản lý blog (tạo, sửa, xóa, publish).

Quản lý đánh giá và review của người dùng.

Xem thống kê doanh thu tổng quan (hôm nay, tuần này, tháng này).

Thống kê doanh thu theo tháng được chọn.

Biểu đồ cột doanh thu từng tháng trong năm.

Xem thống kê đơn hàng hủy và số tiền hoàn.

Đối tượng sử dụng hệ thống là nhân viên quản trị của công ty TasteTheCoffee và khách hàng có nhu cầu mua sắm coffee trực tuyến, bao gồm cả người dùng phổ thông và người sành coffee.

3. Phương pháp nghiên cứu

Nghiên cứu lý thuyết: Thu thập và tham khảo thông tin từ sách, tài liệu, bài báo khoa học và các website liên quan đến thương mại điện tử, thiết kế giao diện, quản lý cơ sở dữ liệu, tích hợp AI và các mô hình bán hàng trực tuyến. Từ đó xây dựng cơ sở lý thuyết làm nền tảng cho quá trình thiết kế hệ thống. Nghiên cứu các công nghệ như React, Node.js, MongoDB, RESTful API, JWT authentication, và các dịch vụ bên thứ ba như Google Gemini AI, PayPal, MoMo, Firebase Authentication.

Triển khai thực nghiệm: Tiến hành xây dựng website mua bán coffee cho công ty TasteTheCoffee theo các chức năng đã đề xuất. Hệ thống được phát triển theo mô hình Agile, chia thành 12 tuần với các giai đoạn: phân tích yêu cầu và thiết kế hệ thống, xây dựng frontend và backend, tích hợp các dịch vụ bên thứ ba, kiểm thử và hoàn thiện. Hệ thống được triển khai thử nghiệm trên môi trường development (localhost) để kiểm tra hoạt động, đánh giá hiệu quả và điều chỉnh trước khi hoàn thiện.

Phương pháp phân tích và thiết kế: Sử dụng các công cụ như UML để thiết kế sơ đồ use case, sơ đồ luồng dữ liệu, sơ đồ database schema. Phân tích và thiết kế API RESTful, giao diện người dùng responsive, và kiến trúc hệ thống full-stack.

Phương pháp kiểm thử: Thực hiện kiểm thử chức năng, kiểm thử bảo mật, kiểm thử hiệu năng và kiểm thử tích hợp để đảm bảo hệ thống hoạt động ổn định và đáp ứng các yêu cầu đã đề ra.

4. Giải pháp công nghệ

Ngôn ngữ lập trình: JavaScript (ES6+).

Nền tảng Backend: Node.js (Framework Express.js 5.1.0).

Frontend: React 18.3.1 (Build tool: Vite 6.3.6).

Hệ quản trị cơ sở dữ liệu: MongoDB 6.0+ (ODM: Mongoose 8.19.0).

Authentication & Authorization:

Firebase Authentication (Google OAuth).

JWT (jsonwebtoken) cho API authentication.

Bcryptjs cho password hashing.

AI Integration: Google Gemini API cho chatbot tư vấn coffee.

Payment Gateway:

PayPal (tích hợp callback và capture payment).

MoMo (tích hợp callback và IPN).

Thư viện và công cụ hỗ trợ:

React Router DOM: Routing.

Axios: HTTP client.

React Icons: Icon library.

Leaflet & React-Leaflet: Maps integration.

Marked & DOMPurify: Markdown parsing và sanitization.

EmailJS: Email service.

React Toastify: Toast notifications.

Recharts 2.15.4: Chart library cho biểu đồ thống kê.

CORS: Cross-origin resource sharing.

Công cụ hỗ trợ phát triển: Visual Studio Code, MongoDB Compass, Postman (cho API testing), Git (cho version control).

5. Cấu trúc đồ án

Chương 1: Giới thiệu tổng quan.

Chương 2: Cơ sở lý thuyết.

Chương 3: Phân tích thuyết kế hệ thống .

Chương 4: Cài đặt demo giao diện.

Kết luận và hướng phát triển.

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN

1.1. Lý do chọn đề tài

Trong bối cảnh chuyển đổi số mạnh mẽ, mua sắm trực tuyến đã trở thành xu hướng tất yếu, đặc biệt với các mặt hàng tiêu dùng như coffee – sản phẩm gắn liền với đời sống hàng ngày của người Việt Nam. Việc xây dựng một hệ thống bán coffee trực tuyến không chỉ giúp công ty TasteTheCoffee mở rộng thị trường, tiếp cận khách hàng trên toàn quốc mà còn nâng cao tính chuyên nghiệp trong quản lý, vận hành và chăm sóc khách hàng.

ReactJS và NodeJS là hai công nghệ web mạnh mẽ, phù hợp để xây dựng hệ thống thương mại điện tử. ReactJS mang lại khả năng cập nhật giao diện linh hoạt, giúp khách hàng dễ dàng duyệt danh mục coffee, xem chi tiết từng sản phẩm, thêm vào giỏ hàng và đặt mua một cách trực quan. Trong khi đó, NodeJS hỗ trợ xử lý nhiều yêu cầu đồng thời, giúp hệ thống vận hành mượt mà, tốc độ phản hồi nhanh và đảm bảo an toàn dữ liệu.

Đề tài “Xây dựng website mua bán Coffee cho công ty TasteTheCoffee” tập trung vào việc ứng dụng các công nghệ hiện đại để phát triển một nền tảng mua sắm trực tuyến tiện lợi, hiệu quả và an toàn. Website sẽ cung cấp cho khách hàng các chức năng chính như:

Xem danh mục sản phẩm coffee (coffee tiện lợi, coffee nguyên chất, coffee truyền thống).

Xem chi tiết sản phẩm: Hiện thị hình ảnh, mô tả chi tiết (nguồn gốc, cách chế biến, loại hạt), giá bán, thông tin ưu đãi, đánh giá thực tế từ khách hàng.

Quản lý giỏ hàng: Người dùng có thể thêm sản phẩm vào giỏ, xóa, chỉnh sửa số lượng, xem tổng giá trị đơn hàng. Giao diện giỏ hàng trực quan, thao tác nhanh chóng.

Đặt hàng và thanh toán trực tuyến: Hỗ trợ đặt hàng nhanh, lựa chọn phương thức thanh toán (VNPay, MoMo, COD, Paypal). Quản lý trạng thái đơn hàng: chờ xác nhận, đang giao, hoàn thành, hủy.

Quản lý tài khoản người dùng: Đăng ký, đăng nhập (tài khoản thường hoặc Google), cập nhật thông tin cá nhân, đổi mật khẩu, xem lịch sử mua hàng, theo dõi trạng thái đơn hàng.

Khuyến mãi, tin tức hoặc trạng thái đơn hàng: Hệ thống gửi thông báo về các chương trình khuyến mãi, tin tức mới, cập nhật trạng thái đơn hàng qua giao diện website.

Tích hợp AI hỗ trợ khách hàng: Gợi ý sản phẩm phù hợp, trả lời câu hỏi về coffee, lưu lịch sử chat AI.

Hệ thống quản trị (Admin) hỗ trợ các chức năng:

Quản lý sản phẩm: Thêm, sửa, xóa sản phẩm, cập nhật tồn kho, quản lý phân loại, chương trình khuyến mãi.

Quản lý đơn hàng và khách hàng: Xem, xác nhận, cập nhật trạng thái đơn hàng; quản lý thông tin khách hàng, lịch sử giao dịch.

Quản lý thu mua nguyên liệu và quy trình gia công: Quản lý thông tin thu mua, nhà cung cấp, chất lượng nguyên liệu; quản lý quy trình rang xay, phối trộn, đóng gói.

Quản lý blog/tin tức, đánh giá sản phẩm: Thêm, sửa, xóa bài viết blog, tin tức; kiểm duyệt, quản lý đánh giá sản phẩm.

Thống kê doanh thu, báo cáo hoạt động kinh doanh: Dashboard tổng quan, thống kê doanh số, số lượng đơn hàng, sản phẩm bán chạy, hiệu quả chương trình khuyến mãi.

Giải pháp mở rộng và bảo trì hệ thống: Thiết kế hệ thống linh hoạt, dễ dàng mở rộng cho nhiều chi nhánh, tích hợp thêm dịch vụ vận chuyển, chăm sóc khách hàng tự động. Tích hợp AI/ML để phân tích dữ liệu bán hàng, tối ưu chiến lược marketing, nâng cao trải nghiệm khách hàng.

Website mua bán Coffee của TasteTheCoffee sẽ trở thành giải pháp công nghệ hữu ích, góp phần nâng cao trải nghiệm khách hàng, tối ưu quy trình bán hàng và hỗ trợ công ty mở rộng thị trường trực tuyến hiệu quả hơn.

1.2. Lịch sử nghiên cứu

1.2.1. Lịch sử nghiên cứu trong nước:

Thương mại điện tử Việt Nam: Từ đầu những năm 2010, các nền tảng thương mại điện tử như Tiki, Shopee, Lazada phát triển mạnh mẽ, thúc đẩy thói quen mua sắm trực tuyến. Các website chuyên ngành thực phẩm, đồ uống như The Coffee House, Trung Nguyên, Highlands Coffee cũng đã triển khai hệ thống bán hàng online, tích hợp thanh toán điện tử, quản lý đơn hàng, chăm sóc khách hàng.

Ứng dụng công nghệ mới: Nhiều doanh nghiệp Việt đã ứng dụng ReactJS, NodeJS, MongoDB để xây dựng hệ thống bán hàng trực tuyến, tối ưu trải nghiệm người dùng,

tăng tốc độ xử lý và bảo mật dữ liệu. Một số dự án nghiên cứu tại các trường đại học cũng tập trung vào xây dựng website bán hàng, quản lý sản phẩm, đơn hàng, tích hợp AI gợi ý sản phẩm và chatbot hỗ trợ khách hàng.

Thanh toán điện tử: Các công thanh toán như VNPay, MoMo, ZaloPay ngày càng phổ biến, được tích hợp vào nhiều website bán hàng, giúp giao dịch nhanh chóng, an toàn.

1.2.2. Lịch sử nghiên cứu ngoài nước:

Phát triển thương mại điện tử toàn cầu: Trên thế giới, các nền tảng như Amazon, eBay, Alibaba đã xây dựng hệ thống bán hàng trực tuyến quy mô lớn, ứng dụng công nghệ hiện đại (ReactJS, NodeJS, NoSQL database) để xử lý hàng triệu giao dịch mỗi ngày, tối ưu hóa trải nghiệm khách hàng.

Website chuyên ngành coffee: Các thương hiệu quốc tế như Starbucks, Blue Bottle Coffee, Peet's Coffee đều có website bán hàng trực tuyến, cung cấp thông tin chi tiết sản phẩm, chương trình khuyến mãi, quản lý đơn hàng, tích hợp thanh toán điện tử và dịch vụ khách hàng trực tuyến.

Ứng dụng AI và dữ liệu lớn: Nhiều nghiên cứu quốc tế tập trung vào việc tích hợp AI/ML vào hệ thống bán hàng để gợi ý sản phẩm, phân tích hành vi khách hàng, tối ưu hóa marketing và chăm sóc khách hàng tự động.

Tóm lại, dự án TasteTheCoffee kế thừa và phát triển các thành tựu nghiên cứu trong và ngoài nước, ứng dụng công nghệ hiện đại để xây dựng nền tảng bán coffee trực tuyến chuyên nghiệp, đáp ứng nhu cầu thực tế của doanh nghiệp và khách hàng.

1.3. Mục tiêu đề tài

1.3.1 Mục tiêu cụ thể

Đề đạt được mục tiêu chung, đề tài “Xây dựng website mua bán Coffee cho công ty TasteTheCoffee” phục vụ khách hàng tại Việt Nam, cung cấp trải nghiệm mua sắm hiện đại, tiện lợi và bảo mật các nội dung chính được triển khai như sau:

Xây dựng giao diện và trải nghiệm người dùng trực quan: Thiết kế giao diện thân thiện, dễ sử dụng, giúp khách hàng dễ dàng tìm kiếm sản phẩm coffee theo loại (coffee nguyên chất, coffee truyền thống, coffee tiện lợi), mức giá, hoặc chương trình khuyến mãi. Giao diện tối ưu cho cả desktop và mobile.

Quản lý thông tin sản phẩm: Cho phép quản trị viên thêm, sửa, xóa và cập nhật tồn kho sản phẩm. Người dùng có thể xem chi tiết sản phẩm với hình ảnh, nguồn gốc, mô

tả, giá bán, và các chương trình ưu đãi. Sản phẩm được phân loại rõ ràng, hỗ trợ tìm kiếm và lọc.

Phát triển chức năng giỏ hàng và đặt hàng trực tuyến: Người dùng có thể thêm sản phẩm vào giỏ, chỉnh sửa số lượng, xóa sản phẩm, tiến hành đặt hàng và lựa chọn phương thức thanh toán phù hợp (VNPay, MoMo, COD). Hệ thống quản lý trạng thái đơn hàng, hỗ trợ theo dõi tiến trình giao dịch.

Quản lý thông tin người dùng: Lưu trữ và quản lý thông tin tài khoản khách hàng, lịch sử mua hàng; hỗ trợ người dùng đăng ký, đăng nhập, cập nhật thông tin cá nhân và theo dõi tình trạng đơn hàng. Đảm bảo bảo mật thông tin và quyền riêng tư.

Quản trị hệ thống: Quản trị viên có thể quản lý sản phẩm, đơn hàng, tồn kho, thu mua nguyên liệu, quy trình gia công, khuyến mãi, blog/tin tức, đánh giá sản phẩm. Truy cập dashboard tổng quan để theo dõi hiệu suất kinh doanh.

Cải tiến và tối ưu hóa: Theo dõi hiệu suất website, đảm bảo tốc độ truy cập nhanh, hỗ trợ đa thiết bị và mang lại trải nghiệm tốt nhất cho khách hàng. Tích hợp AI để gợi ý sản phẩm, hỗ trợ khách hàng trực tuyến và tối ưu hóa hoạt động kinh doanh.

1.3.2 Mục tiêu ứng dụng công nghệ

Node.js & Express: Xây dựng hệ thống back-end mạnh mẽ, xử lý nhanh các yêu cầu đặt hàng, quản lý sản phẩm, người dùng, đơn hàng, thu mua nguyên liệu, gia công, khuyến mãi, blog/tin tức và đánh giá sản phẩm.

Cung cấp các dịch vụ RESTful API để kết nối front-end (ReactJS) và back-end, phục vụ các chức năng quản lý sản phẩm, giỏ hàng, đơn hàng, tài khoản, thanh toán trực tuyến (VNPay, MoMo), tích hợp AI (Google Gemini).

Đảm bảo bảo mật thông tin, xác thực người dùng bằng JWT và Google OAuth, phân quyền truy cập cho quản trị viên và khách hàng.

ReactJS (Vite): Xây dựng giao diện người dùng hiện đại, trực quan, thân thiện, tối ưu trải nghiệm mua sắm trên cả desktop và mobile.

Hỗ trợ cập nhật dữ liệu theo thời gian thực, giúp khách hàng thao tác giỏ hàng, đặt hàng, theo dõi trạng thái đơn hàng mượt mà.

Tích hợp các chức năng nâng cao như gợi ý sản phẩm AI, chatbot hỗ trợ khách hàng, quản lý tài khoản, lịch sử mua hàng, blog/tin tức.

MongoDB (Mongoose): Lưu trữ và quản lý dữ liệu sản phẩm coffee, tài khoản khách hàng, đơn hàng, thu mua nguyên liệu, quy trình gia công, khuyến mãi, blog/tin tức, đánh giá sản phẩm và lịch sử chat AI.

Hỗ trợ khả năng truy vấn nhanh, mở rộng linh hoạt, phù hợp với hệ thống thương mại điện tử có lượng dữ liệu lớn và đa dạng.

Đảm bảo hiệu năng cao, an toàn dữ liệu, phục vụ mở rộng hệ thống cho nhiều chi nhánh và dịch vụ trong tương lai.

Tích hợp dịch vụ bên ngoài: Kết nối cổng thanh toán điện tử (VNPay, MoMo), hỗ trợ xác thực giao dịch và cập nhật trạng thái đơn hàng tự động.

Tích hợp Google Gemini AI để gợi ý sản phẩm, trả lời câu hỏi, nâng cao trải nghiệm khách hàng.

DevOps & Bảo mật: Sử dụng các script tự động kiểm tra, khởi tạo dữ liệu, quản lý kết nối MongoDB, thiết lập Ngrok để hỗ trợ truy cập từ internet, Quản lý biến môi trường qua file .env, đảm bảo an toàn thông tin và dễ dàng triển khai trên nhiều môi trường.

1.4 Đối tượng phạm vi nghiên cứu

1.4.1. Phạm vi nội dung

Quản lý người dùng: Khách hàng có thể đăng ký, đăng nhập (tài khoản thường hoặc Google), cập nhật thông tin cá nhân, đổi mật khẩu, theo dõi lịch sử mua hàng và trạng thái đơn hàng, hệ thống sử dụng JWT để xác thực và bảo vệ các chức năng quan trọng.

Xem chi tiết sản phẩm: Hiển thị hình ảnh, mô tả chi tiết (nguồn gốc, cách chế biến, loại hạt), giá bán, thông tin ưu đãi, đánh giá từ khách hàng, sản phẩm được phân loại theo danh mục, hỗ trợ tìm kiếm và lọc.

Giỏ hàng, đặt hàng và thanh toán trực tuyến: Khách hàng có thể thêm sản phẩm vào giỏ, chỉnh sửa số lượng, xóa sản phẩm khỏi giỏ. Tiến hành đặt hàng, lựa chọn phương thức thanh toán (VNPay, MoMo, COD). Hệ thống quản lý trạng thái đơn hàng: chờ xác nhận, đang giao, hoàn thành, hủy.

Quản lý tài khoản khách hàng: Quản lý thông tin cá nhân, đổi mật khẩu, bảo mật tài khoản, xem lịch sử mua hàng, trạng thái đơn hàng.

Chức năng quản trị viên: Quản lý sản phẩm (thêm/sửa/xóa), quản lý đơn hàng, quản lý tồn kho, quản lý thu mua nguyên liệu, quy trình gia công, khuyến mãi, blog/tin

tức, đánh giá sản phẩm. Truy cập dashboard tổng quan: doanh thu, số lượng đơn hàng, sản phẩm, người dùng.

Quản lý thu mua nguyên liệu và gia công: Quản lý thông tin thu mua nguyên liệu (loại, số lượng, giá, chất lượng), lịch sử thu mua, nhà cung cấp. Quản lý quy trình gia công (rang xay, phối trộn, đóng gói), trạng thái và mô tả chi tiết.

Quản lý khuyến mãi và blog/tin tức: Tạo, sửa, xóa mã giảm giá, áp dụng khi thanh toán. Quản lý bài viết blog, tin tức, hiển thị cho khách hàng.

Quản lý đánh giá sản phẩm: Khách hàng thêm, sửa, xóa đánh giá sản phẩm, hiển thị điểm trung bình và nhận xét.

Tích hợp AI: Gợi ý sản phẩm phù hợp, trả lời câu hỏi về coffee, lưu lịch sử chat AI cho từng người dùng.

Hệ thống tập trung vào: Tính tiện lợi, tốc độ phản hồi nhanh, giao diện trực quan, bảo mật thông tin người dùng.

1.4.2. Phạm vi không gian

Website được thiết kế phục vụ cho công ty TasteTheCoffee tại Việt Nam, hướng tới khách hàng mua coffee trực tuyến.

Giai đoạn đầu hệ thống phục vụ cho cửa hàng/chi nhánh chính của công ty.

Có khả năng mở rộng cho nhiều chi nhánh, tích hợp thêm cổng thanh toán điện tử, hệ thống vận chuyển và dịch vụ khách hàng trực tuyến.

Hệ thống sử dụng công nghệ Node.js, Express, MongoDB cho backend; React cho frontend; tích hợp các dịch vụ bên ngoài như VNPay, MoMo, Google Gemini AI.

Phạm vi thời gian:

Đề tài thực hiện trong thời gian khóa luận tốt nghiệp, tập trung vào nghiên cứu, xây dựng và thử nghiệm các chức năng cốt lõi của hệ thống.

Trong tương lai, hệ thống có thể nâng cấp và mở rộng thêm các tính năng nâng cao như:

Gợi ý sản phẩm cá nhân hóa dựa trên thói quen mua sắm.

Chatbot hỗ trợ khách hàng trực tuyến.

Tích hợp AI/ML để phân tích dữ liệu bán hàng và tối ưu chiến lược marketing.

Quản lý vận chuyển, chăm sóc khách hàng tự động, mở rộng cho nhiều chi nhánh.

CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu về ngôn ngữ JavaScript

2.1.1. Khái niệm

Javascript chính là một ngôn ngữ lập trình với khả năng đem tới sự sinh động khi thiết kế website. Đây là dạng ngôn ngữ theo kịch bản, dựa trên chính đối tượng phát triển có sẵn, hoặc là tự định nghĩa ra. Chính vì tính tiện lợi, hiệu quả ứng dụng cao mà ngôn ngữ lập trình này được tin dùng ngày càng nhiều, ứng dụng rộng rãi trong các website hiệu quả.

Việc sử dụng ngôn ngữ JS có thể ứng dụng cho mọi trình duyệt khác nhau, hiện được sử dụng phổ biến như Chrome, hay Firefox, ... Hơn nữa, đây còn là ngôn ngữ lập trình hoạt động hiệu quả, được hỗ trợ đầy đủ trên các trình duyệt của thiết bị di động. Bởi thế mà việc sử dụng đa dạng, có thể đáp ứng tốt cho nhiều nhu cầu, những đòi hỏi khác nhau của người dùng.

Nhìn chung, ngôn ngữ lập trình Javascript khi sử dụng được đánh giá dễ học, được phát triển bởi Netscape và sử dụng trong hơn 92% các website ngày nay. Việc sử dụng JS có thể gắn trong một element trong website, hoặc có thể trong một sự kiện của trang web tiện lợi và hiệu quả. Chính việc có thể hoạt động nhanh, nhẹ và mượt mà,... nên ngôn ngữ lập trình này càng được tin tưởng và ứng dụng nhiều hơn cho nhu cầu của người dùng. Chúng ta có thể thêm trực tiếp JS trực tiếp với HTML, cũng có thể lưu lại trên một files hoàn toàn riêng biệt và sử dụng khi cần thiết.

2.1.2. Ưu điểm và nhược điểm

Là ngôn ngữ lập trình dễ dàng sử dụng: Đơn giản, dễ dàng tìm hiểu và sử dụng là đặc điểm nổi bật của ngôn ngữ lập trình này. Với những cú pháp khá tương đồng với tiếng Anh thì việc sử dụng Javascript trở nên dễ dàng, dễ tiếp cận hơn rất nhiều. Thông qua mô hình DOM được sử dụng, cung cấp tới nhiều tính năng hữu ích, được viết sẵn đem lại khả năng đáp ứng tốt cho những nhu cầu, những đòi hỏi khác nhau từ phía người dùng. Đánh giá cao ở độ nhanh chóng với người dùng: Việc sử dụng ngôn ngữ JS có khả năng đảm bảo khi mã được thực hiện trên thiết bị máy tính của người dùng

Sử dụng đa dạng trên nhiều trình duyệt: Một ưu điểm nổi bật giúp JS ngày càng được tin dùng, ứng dụng phong phú là ở khả năng sử dụng thông qua nhiều trình duyệt

khác nhau. Từ các trình duyệt trên máy tính như Chrome, tới Firefox, ... hay những trình duyệt trên thiết bị di động, ... đều có thể sử dụng được đầy đủ và hiệu quả. Chính vì sự thích hợp với nhiều trình duyệt, nhiều nền tảng khác nhau giúp nó được tin dùng nhiều hơn để đáp ứng tốt cho đòi hỏi của con người.

Ví dụ:

```
// Hiển thị một thông báo chào mừng
```

```
function greetUser(name) {  
    alert(`Xin chào, ${name}! Chào mừng bạn đến với JavaScript.`;  
}  
  
// Gọi hàm greetUser('Tiến');
```

2.1.3. Tính năng của JavaScript

Tính năng tương tác (Interactivity): JavaScript giúp tạo các trang web tương tác, như hiển thị thông báo, xử lý sự kiện (click, hover), hoặc cập nhật nội dung trang mà không cần tải lại.

Khả năng đa nền tảng: JavaScript có thể chạy trên mọi trình duyệt hiện đại và không cần phần mềm bổ sung.

Thời gian thực (Real-time): JavaScript hỗ trợ xây dựng các ứng dụng thời gian thực như chat, thông báo, và các ứng dụng live-streaming.

Hỗ trợ thư viện và framework mạnh mẽ: Các thư viện và framework phổ biến như React, Angular, và Vue.js giúp tăng tốc độ phát triển ứng dụng.

Tích hợp dễ dàng với HTML/CSS: JavaScript hoạt động liền mạch với HTML và CSS, giúp tạo ra giao diện người dùng hấp dẫn và dễ sử dụng.

Chạy trên cả phía client và server: Với Node.js, JavaScript không chỉ giới hạn ở phía client mà còn được sử dụng cho các ứng dụng server-side mạnh mẽ.

2.2 Giới thiệu về ReactJs

2.2.1. Khái niệm

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, dùng để xây dựng giao diện người dùng (UI) cho các ứng dụng web. ReactJS giúp các

nhà phát triển tạo ra các ứng dụng web có khả năng tương tác cao và hiệu suất tốt bằng cách sử dụng khái niệm component-based architecture (kiến trúc dựa trên thành phần).

Khi sử dụng ReactJS, người dùng có thể khai thác được nhiều tính năng, trong đó có:

Xây dựng giao diện người dùng (UI) linh hoạt và tái sử dụng.

Tối ưu hóa hiệu suất với Virtual DOM.

Tích hợp dễ dàng với các công nghệ khác.

Tính năng cập nhật nhanh chóng với ES6+.

Lý do ReactJS được nhiều doanh nghiệp phần mềm hướng tới trong thương mại điện tử:

Hiệu suất cao và tối ưu trải nghiệm người dùng:

Virtual DOM giúp React sử dụng DOM ảo, tối ưu quá trình cập nhật giao diện khi có thay đổi.

Trong thương mại điện tử, người dùng thường xuyên tương tác với trang (tìm kiếm sản phẩm, thêm sản phẩm vào giỏ hàng, thanh toán,...). Do đó tốc độ tải nhanh và trải nghiệm mượt mà giúp cải thiện tỷ lệ chuyển đổi.

Lazy Loading: React hỗ trợ tính năng tải lười, chỉ tải nội dung khi cần thiết → trang web tải nhanh hơn và tối ưu trải nghiệm.

Tính linh hoạt và tái sử dụng cao.

Component-based Architecture: Trong thương mại điện tử, nhiều phần của trang web có thể tái sử dụng như danh sách sản phẩm, form thanh toán, thanh điều hướng,...

React giúp tạo component tái sử dụng, giảm chi phí phát triển, bảo trì và dễ dàng mở rộng khi thêm tính năng mới.

Modular Development: Kiến trúc mô-đun cho phép đội ngũ phát triển làm việc song song trên các phần khác nhau của ứng dụng, tăng tốc độ triển khai.

Khả năng tối ưu hóa SEO.

Server-side Rendering (SSR): React có thể kết hợp với các framework như Next.js để hỗ trợ render phía server, giúp cải thiện SEO.

Improved SEO Performance: Trong thương mại điện tử, SEO cực kỳ quan trọng để sản phẩm hiển thị trên Google. React kết hợp với SSR hoặc prerendering giúp trang web dễ lập chỉ mục và tăng thứ hạng tìm kiếm.

Hỗ trợ cá nhân hóa trải nghiệm người dùng:

React giúp dễ dàng xây dựng các tính năng thương mại điện tử như: đề xuất sản phẩm, danh sách yêu thích, bộ lọc tìm kiếm.

Từ đó, website có thể tạo ra trải nghiệm mua sắm cá nhân hóa theo hành vi và sở thích → tăng tỷ lệ chuyển đổi và doanh thu.

2.2.2. Tính năng của ReactJs

Tái sử dụng thành phần (Reusable Components): React cho phép xây dựng các thành phần UI độc lập, có thể tái sử dụng trong toàn bộ ứng dụng → giúp tiết kiệm thời gian phát triển và giảm lặp lại mã nguồn.

Hiệu suất cao (High Performance): Với Virtual DOM, React chỉ cập nhật những phần giao diện thay đổi thay vì tải lại toàn bộ trang → tăng tốc độ xử lý và mang lại trải nghiệm mượt mà cho người dùng.

Định hướng dữ liệu (Unidirectional Data Flow): React quản lý dữ liệu theo hướng đơn chiều, giúp dễ dàng kiểm soát trạng thái và luồng dữ liệu trong ứng dụng, hạn chế lỗi và tăng tính ổn định.

Dễ bảo trì: Cấu trúc thành phần của React giúp mã nguồn rõ ràng, dễ đọc, dễ mở rộng và bảo trì, đặc biệt hiệu quả trong các dự án lớn.

SEO thân thiện (SEO-Friendly): React có thể kết hợp với Server-side Rendering (SSR) hoặc các framework như Next.js để tăng khả năng SEO, giúp website thương mại điện tử dễ dàng được lập chỉ mục trên công cụ tìm kiếm.

Hỗ trợ cộng đồng và công cụ mạnh mẽ: React sở hữu cộng đồng phát triển lớn, tài liệu phong phú và hệ sinh thái mạnh mẽ với nhiều thư viện hữu ích như React Router, Redux, Next.js,... giúp phát triển nhanh chóng và thuận tiện.

2.3 Giới thiệu về Monggo

2.3.1. Khái niệm

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL phổ biến, sử dụng mô hình lưu trữ dạng tài liệu (document) với cấu trúc BSON (Binary JSON). Không giống như các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) như MySQL hay PostgreSQL, MongoDB không sử dụng bảng và dòng mà tổ chức dữ liệu thành các collection và document.

2.3.2. Tính năng của MongoDB

Định hướng Tài liệu (Document-Oriented): Lưu trữ dữ liệu dưới dạng tài liệu BSON (JSON Binary), cho phép các cấu trúc dữ liệu phức tạp được lồng vào nhau, giúp ánh xạ trực tiếp với các đối tượng trong lập trình hướng đối tượng.

Khả năng Mở rộng Ngang (Horizontal Scalability): Hỗ trợ Sharding (phân mảnh dữ liệu) để phân tán dữ liệu và tải xử lý trên nhiều máy chủ. Điều này cho phép mở rộng hệ thống bằng cách thêm máy chủ mới thay vì nâng cấp phần cứng máy chủ hiện có.

Tính sẵn sàng cao (High Availability): Sử dụng tính năng Replication (Bộ Replica - Replica Sets) để tạo các bản sao dữ liệu trên nhiều máy chủ. Nếu máy chủ chính (Primary) gặp sự cố, một máy chủ phụ (Secondary) sẽ tự động được chọn làm máy chủ chính mới.

Truy vấn Đa dạng: Hỗ trợ truy vấn linh hoạt, truy vấn theo trường (field), phạm vi (range), biểu thức chính quy (regex) và một khung tập hợp dữ liệu mạnh mẽ (Aggregation Framework).

2.4 Giới thiệu về NodeJs

2.4.1. Khái niệm

Node.js là một nền tảng chạy JavaScript phía server, giúp các nhà phát triển xây dựng các ứng dụng mạng nhanh chóng, hiệu quả và có khả năng mở rộng cao. Node.js được phát triển dựa trên V8 JavaScript Engine của Google, trình thông dịch JavaScript được sử dụng trong trình duyệt Chrome. Điểm đặc trưng của Node.js là kiến trúc hướng sự kiện (event-driven) và mô hình không đồng bộ (asynchronous), giúp xử lý các yêu cầu từ client nhanh chóng và hiệu quả.

JavaScript phía server: Node.js cho phép các nhà phát triển sử dụng JavaScript ở phía server. Điều này mang lại sự thống nhất về ngôn ngữ lập trình giữa client và server, giúp giảm bớt sự phức tạp và cải thiện năng suất phát triển.

Không đồng bộ và hướng sự kiện: Node.js sử dụng mô hình non-blocking I/O (nhập/xuất không chặn) và xử lý theo hướng sự kiện, giúp hệ thống có thể xử lý nhiều yêu cầu cùng lúc mà không bị chặn bởi các yêu cầu trước đó. Điều này giúp tối ưu hiệu suất, đặc biệt trong các ứng dụng có nhiều người dùng hoặc yêu cầu thời gian phản hồi nhanh.

Cross-platform: Node.js hoạt động trên nhiều hệ điều hành như Windows, macOS, Linux và thậm chí trên các thiết bị nhúng như Raspberry Pi, giúp nó trở thành một nền tảng linh hoạt cho các loại dự án khác nhau.

Xử lý đơn luồng với Event Loop: Mặc dù Node.js chỉ sử dụng một luồng để xử lý các yêu cầu, nó sử dụng Event Loop để quản lý nhiều kết nối cùng một lúc mà không cần nhiều luồng. Điều này giúp Node.js có thể xử lý nhiều yêu cầu đồng thời mà không gặp các vấn đề thường gặp với các hệ thống đa luồng như cạnh tranh tài nguyên (resource contention).

Hỗ trợ WebSockets: Node.js hỗ trợ các kết nối WebSocket, giúp dễ dàng xây dựng các ứng dụng thời gian thực (real-time) như ứng dụng chat, các trò chơi nhiều người chơi, hoặc hệ thống theo dõi sự kiện.

2.4.2. Tính năng của NodeJS

Ứng dụng của Node.Js:

API Backend: Node.js rất phổ biến để xây dựng các API RESTful với các thư viện như Express.js.

Ứng dụng real-time: Node.js lý tưởng cho các ứng dụng thời gian thực như chat hoặc hệ thống thông báo.

Xử lý file: Node.js có thể xử lý file (đọc, ghi, xóa) trên server một cách dễ dàng.

Microservices: Node.js rất phù hợp cho các kiến trúc microservices nhờ khả năng nhẹ và hiệu suất cao.

Ứng dụng IoT: Các thiết bị IoT thường sử dụng Node.js để xử lý dữ liệu cảm biến trong thời gian thực.

Các thư viện phổ biến của Node.Js:

Express.js: Framework nhẹ, linh hoạt để xây dựng web server và API. • Socket.io: Hỗ trợ giao tiếp real-time giữa client và server.

Mongoose: Thư viện để làm việc với MongoDB dễ dàng hơn.

JWT (jsonwebtoken): Dùng để xác thực người dùng qua token. • Lodash: Cung cấp các hàm tiện ích để xử lý dữ liệu.

Lợi ích của Node.Js:

Đơn giản hóa ngăn xếp công nghệ: Cả frontend và backend đều dùng JavaScript, giúp dễ dàng quản lý và đồng bộ mã nguồn.

2.5. Giới thiệu ExpressJS

2.5.1. . Khái niệm

ExpressJS là một framework đơn giản được xây dựng trên nền tảng Node.js, ra đời với mục đích làm cho việc phát triển các ứng dụng web và API trở nên đơn giản, hiệu quả và dễ bảo trì hơn. Bằng cách cung cấp các công cụ tiện lợi để xử lý yêu cầu HTTP, quản lý định tuyến (routing), và thêm các chức năng khác vào ứng dụng thông qua middleware

2.5.2. Tính năng của ExpressJS

Phát triển máy chủ nhanh hơn: ExpressJS tối ưu hóa cú pháp và cung cấp các phương thức và hàm tiện ích để xử lý các tác vụ phổ biến trong lập trình web. Nhờ đó, bạn có thể viết code ngắn gọn và tối giản hóa quy trình phát triển.

Định tuyến (Routing): ExpressJS cung cấp một hệ thống định tuyến mạnh mẽ, cho phép bạn xác định các tuyến đường (routes) để xử lý yêu cầu từ người dùng và phản hồi tương ứng. Điều này giúp tổ chức và quản lý các thành phần của ứng dụng một cách dễ dàng.

Middleware: ExpressJS hỗ trợ middleware, cho phép bạn thêm các chức năng trung gian vào quy trình xử lý yêu cầu và phản hồi. Middleware giúp xác thực người dùng, ghi log, xử lý lỗi, nén dữ liệu và thực hiện nhiều tác vụ khác một cách linh hoạt.

Cấu hình môi trường: ExpressJS cung cấp một cách dễ dàng để cấu hình môi trường phát triển và môi trường sản xuất. Bạn có thể thiết lập các biến môi trường, cấu hình định dạng và quy tắc, tùy chỉnh ứng dụng của mình theo các môi trường khác nhau.

Xử lý lỗi: ExpressJS cung cấp cơ chế xử lý lỗi cho phép bạn kiểm soát và xử lý các lỗi xảy ra trong quá trình xử lý yêu cầu. Bạn có thể tạo ra các middleware để xử lý lỗi và phản hồi với các thông báo lỗi tùy chỉnh.

CHƯƠNG 3 : PHÂN TÍCH THUYẾT KẾ HỆ THỐNG

3.1. Xác định yêu cầu xây dựng website mua bán Coffee

Việc xác định yêu cầu là bước quan trọng trong quá trình phát triển phần mềm. Trong đề tài “Xây dựng website mua bán Coffee cho công ty TasteTheCoffee”, hệ thống cần đáp ứng được nhu cầu mua sắm trực tuyến của khách hàng và hỗ trợ quản trị viên trong việc quản lý sản phẩm, đơn hàng và khách hàng.

3.1.1. Yêu cầu chức năng

Đăng ký/Đăng nhập người dùng:

- Khách hàng có thể tạo tài khoản, đăng nhập để mua hàng.
- Hỗ trợ đăng nhập qua email/mật khẩu hoặc Google OAuth.
- Quản trị viên có tài khoản riêng để quản lý sản phẩm, đơn hàng và khách hàng.

Xem thông tin sản phẩm (Coffee) :

- Hiện thị danh sách sản phẩm theo loại: coffee tiện lợi, coffee truyền thống, coffee nguyên chất.
- Cung cấp thông tin chi tiết: hình ảnh, mô tả sản phẩm (nguồn gốc, cách chế biến), giá, khuyến mãi, đánh giá.

Tìm kiếm và lọc sản phẩm:

- Người dùng có thể tìm kiếm sản phẩm theo tên, giá hoặc thương hiệu.

Bộ lọc hỗ trợ lọc nhanh theo khoảng giá, thương hiệu, tình trạng khuyến mãi.

Quản lý giỏ hàng:

- Người dùng có thể thêm sản phẩm vào giỏ, cập nhật số lượng hoặc xóa sản phẩm.
- Hiện thị tổng giá trị đơn hàng trước khi thanh toán.

Đặt hàng & Thanh toán:

- Người dùng xác nhận thông tin cá nhân và địa chỉ giao hàng.
- Hệ thống hỗ trợ nhiều phương thức thanh toán: thanh toán khi nhận hàng (COD), ví điện tử, thẻ ngân hàng.

Xem lịch sử mua hàng:

- Người dùng có thể xem lại các đơn hàng đã mua, tình trạng đơn (đang xử lý, đã giao, đã hủy).

Quản lý của Admin:

- Quản lý sản phẩm: thêm, sửa, xóa sản phẩm và các thông tin liên quan.
- Quản lý đơn hàng: xem, duyệt, cập nhật trạng thái đơn hàng.
- Quản lý người dùng: phân quyền khách hàng, nhân viên, quản trị viên.
- Quản lý khuyến mãi: tạo các chương trình giảm giá, voucher.

3.1.2. Yêu cầu phi chức năng

- Hiệu suất: Hệ thống phản hồi nhanh với số lượng người dùng lớn truy cập đồng thời.
- Bảo mật: Dữ liệu người dùng và giao dịch cần được bảo vệ, mã hóa mật khẩu, xác thực truy cập.
- Tính mở rộng: Hệ thống dễ dàng nâng cấp để thêm tính năng mới hoặc tích hợp dịch vụ bên thứ ba.
- Thân thiện người dùng: Giao diện trực quan, dễ sử dụng trên cả máy tính và thiết bị di động.

3.1.3. Yêu cầu hệ thống

- Frontend: Sử dụng ReactJS để xây dựng giao diện người dùng, đảm bảo trải nghiệm trực quan và thân thiện.
- Backend: Sử dụng NodeJS (ExpressJS) để xử lý logic nghiệp vụ và kết nối cơ sở dữ liệu.
- Database: Sử dụng MongoDB hoặc MySQL để lưu trữ dữ liệu sản phẩm (coffee), người dùng, đơn hàng, giỏ hàng,...
- Môi trường triển khai: Hệ thống có thể triển khai trên máy chủ cloud (AWS, Heroku, Vercel, Render,...) hoặc local server tùy điều kiện thực tế.

3.2. Phân tích yêu cầu :

3.2.1 Yêu cầu chức năng chi tiết

Bảng 3.1 Yêu cầu chức năng chi tiết khách hàng

STT	Chức năng	Mô tả
1	Đăng ký/Đăng nhập	Cho phép khách hàng tạo tài khoản, đăng nhập bằng email/mật khẩu hoặc Google OAuth để mua hàng. Hệ thống tự động hash mật khẩu và cấp JWT token.
2	Xem sản phẩm	Khách hàng có thể xem danh sách coffee theo danh mục, chi tiết sản phẩm (hình ảnh, mô tả, xuất xứ, giá, rating, số lượng review). Hiển thị đánh giá của người dùng khác.
3	Tìm kiếm sản phẩm	Cho phép tìm kiếm real-time theo tên sản phẩm, loại coffee, thương hiệu. Hiển thị kết quả trong dropdown, có thể thêm vào giỏ từ kết quả tìm kiếm.
4	Quản lý giỏ hàng	Thêm sản phẩm vào giỏ, cập nhật số lượng, xóa sản phẩm, tính tổng tiền. Giỏ hàng được lưu theo từng user trong localStorage. Hiển thị số lượng sản phẩm trên icon giỏ hàng..
5	Đặt hàng & Thanh toán	Chọn địa chỉ giao hàng (có thể lấy tự động bằng GPS), áp dụng mã khuyến mãi, chọn phương thức thanh toán (COD, PayPal, MoMo), xác nhận đơn hàng. Tính phí ship tự động (miễn phí cho đơn $\geq 500k$).
6	Xem lịch sử mua hàng	Xem lại tất cả đơn hàng đã mua, trạng thái đơn (Xác nhận đơn hàng, Chuẩn bị đơn hàng, Đang giao hàng, Giao thành công). Hiển thị timeline theo dõi và trạng thái thanh toán chi tiết.
7	Khuyến mãi	Áp dụng mã khuyến mãi khi đặt hàng. Hệ thống validate mã (hạn sử dụng, đơn hàng tối thiểu, số lần sử dụng, quyền user). Hỗ trợ giảm giá theo phần trăm hoặc số tiền cố định.

8	Quản lý tài khoản cá nhân	Đăng xuất khỏi hệ thống. Thông tin cá nhân được tự động điền khi đặt hàng. Giỏ hàng được giữ lại sau khi đăng xuất.
9	AI Chatbot tư vấn	Tương tác với AI chatbot để được tư vấn công thức pha coffee. AI gợi ý sản phẩm dựa trên sở thích và sản phẩm trong giỏ hàng.

Bảng 3.2 Yêu cầu chức năng chi tiết admin

STT	Chức năng	Mô tả
1	Dashboard thống kê	Xem tổng quan: tổng đơn hàng, tổng thu nhập, số đơn đã giao, số đơn đang đợi giao. Thống kê theo thời gian: hôm nay, tuần này, tháng này. Có nút làm mới dữ liệu.
2	Quản lý người dùng	Xem danh sách tất cả người dùng. Cập nhật thông tin: tên, email, số điện thoại (cho user local), vai trò (user/admin). Xóa người dùng (không được xóa chính mình). Hiển thị phương thức đăng nhập (local/Google).
3	Quản lý sản phẩm	Thêm sản phẩm mới: mã SP, tên, thương hiệu, giá, trọng lượng, danh mục, mô tả, hình ảnh, trạng thái tồn kho. Sửa thông tin sản phẩm. Xóa sản phẩm. Xem danh sách tất cả sản phẩm với đầy đủ thông tin.
4	Quản lý đơn hàng	Xem tất cả đơn hàng, sắp xếp theo thời gian mới nhất. Cập nhật trạng thái đơn hàng (Xác nhận đơn hàng, Chuẩn bị đơn hàng, Đang giao hàng, Giao thành công). Cập nhật trạng thái thanh toán (paid/unpaid/failed). Sửa thông tin đơn hàng: items, thông tin giao hàng, tổng tiền, mã khuyến mãi. Xóa đơn hàng.
5	Quản lý khuyến mãi	Tạo mã khuyến mãi mới: code, mô tả, loại giảm giá (phần trăm/số tiền cố định), giá trị giảm, đơn hàng tối thiểu, giảm tối đa, ngày bắt đầu/kết thúc, số lần sử dụng,

		áp dụng cho user cụ thể. Sửa và xóa mã khuyến mãi. Xem danh sách tất cả mã, theo dõi số lần đã sử dụng.
6	Quản lý đánh giá	Xem tất cả đánh giá của khách hàng. Hiển thị thông tin: user, sản phẩm, đơn hàng, rating, comment, ngày đánh giá. Xóa bất kỳ đánh giá nào.
7	Quản lý Blog	Tạo blog mới: tiêu đề, nội dung (markdown), tóm tắt, tác giả, hình ảnh, danh mục, tags, trạng thái publish, featured. Sửa và xóa blog. Xem danh sách tất cả blog (bao gồm chưa publish). Tự động tạo slug từ tiêu đề.
8	Quản lý gia công coffee	Thêm phương thức gia công: tên, giá (VND/kg), mô tả. Sửa và xóa phương thức gia công. Xem danh sách tất cả phương thức gia công.
9	Quản lý thu mua coffee	Thêm dịch vụ thu mua: tên, loại (tươi/khô), trọng lượng (số hoặc khoảng min-max), giá, chất lượng. Sửa và xóa dịch vụ thu mua. Xem danh sách tất cả dịch vụ thu mua.
10	Xác thực quyền Admin	Kiểm tra JWT token và vai trò admin trước khi cho phép truy cập các chức năng. Tự động đăng xuất nếu token hết hạn hoặc không có quyền.

Bảng 3.3. Yêu cầu phi chức năng

STT	Chức năng	Mô tả
1	Bảo mật	Hệ thống sử dụng JWT token để xác thực API (hết hạn 7 ngày). Mật khẩu được hash bằng bcryptjs với salt rounds 10. Cấu hình CORS để kiểm soát truy cập từ các origin. Validate và sanitize input để chống injection. Xác thực chữ ký cho payment gateway (PayPal, MoMo).
2	Hiệu năng	Database indexing cho các collection quan trọng (Review, Blog, PromoCode) để tối ưu truy vấn. Pagination cho danh sách blog. Debounce 300ms cho tìm kiếm real-time. MongoDB connection pooling với timeout 5 giây. Giới hạn số lượng reviews hiển thị (100 reviews mới nhất).
3	Giao diện người dùng	Giao diện responsive, hỗ trợ desktop và mobile. Sử dụng React 18 với Vite để build nhanh. Toast notifications cho thông báo. Loading states cho các thao tác bất đồng bộ. Hỗ trợ dark mode (nếu có).
4	Khả năng mở rộng	Kiến trúc modular: tách controllers, routes, models. Sử dụng environment variables để cấu hình. Hỗ trợ nhiều payment gateway (COD, PayPal, MoMo). Dễ thêm tính năng mới.
5	Tương thích	Hỗ trợ các trình duyệt hiện đại (Chrome, Firefox, Safari, Edge). Responsive trên nhiều kích thước màn hình. API RESTful chuẩn. Tương thích với Google OAuth.
6	Khả năng sử dụng	Giao diện thân thiện, dễ điều hướng. Validation form với thông báo rõ ràng. Hỗ trợ tự động điền thông tin từ tài khoản. Lấy địa chỉ tự động bằng GPS. Hướng dẫn sử dụng trong giao diện.

7	Độ tin cậy	Xử lý lỗi toàn diện với thông báo phù hợp. Timeout cho kết nối MongoDB (5 giây). Xử lý callback payment (PayPal, MoMo). Retry cho các thao tác quan trọng. Logging chi tiết cho debugging.
8	Khả năng bảo trì	Code được tổ chức theo cấu trúc rõ ràng (MVC pattern). Logging đầy đủ cho các request và error. Thông báo lỗi chi tiết trong development mode. Sử dụng TypeScript/ES6+ để dễ bảo trì.
9	Khả năng truy cập	Hỗ trợ keyboard navigation. Alt text cho hình ảnh. Contrast màu đạt chuẩn. Font size có thể đọc được.
10	Tích hợp bên thứ ba	Tích hợp Google Gemini AI cho chatbot. Tích hợp PayPal và MoMo cho thanh toán. Tích hợp Google OAuth cho đăng nhập. Tích hợp OpenStreetMap cho địa chỉ. Tích hợp EmailJS cho gửi email.
11	Quản lý phiên bản	Sử dụng npm/yarn để quản lý dependencies. Version control với Git. Cấu hình môi trường riêng cho development và production.
12	Tài liệu	README chi tiết với hướng dẫn cài đặt. Tài liệu API endpoints. Hướng dẫn cấu hình môi trường. Ghi chú trong code (comments).

3.2.2 Mô hình hóa yêu cầu

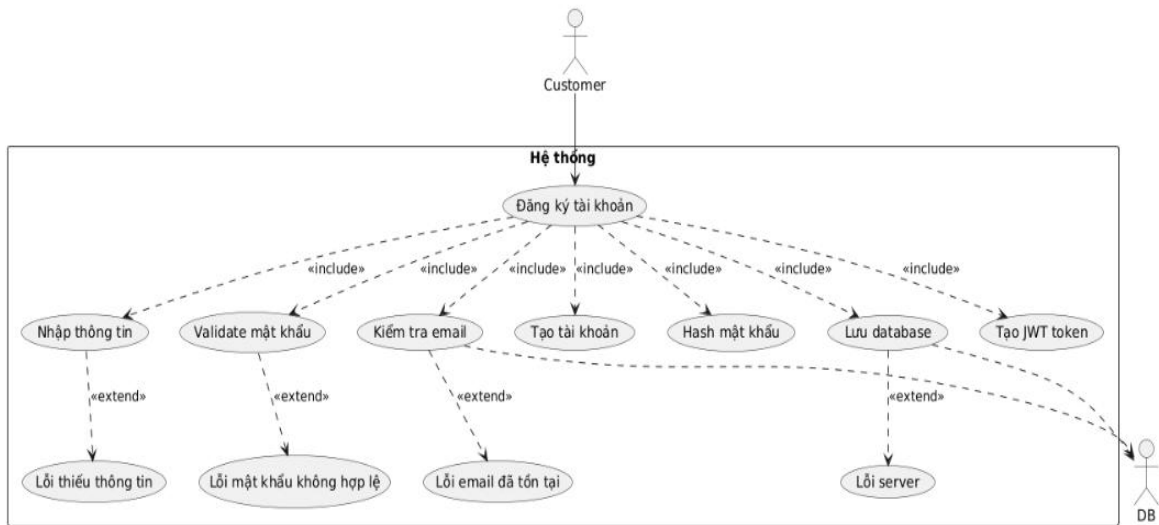
- Usecase tổng quát :



Hình 3.1 Usecase tổng quát

3.3.1 Chức năng đăng kí

- Biểu đồ Use Case đăng ký



Hình 3.2 Biểu đồ Use Case đăng ký

- Mô tả user đăng kí :

Bảng 3.4 Mô tả user đăng kí

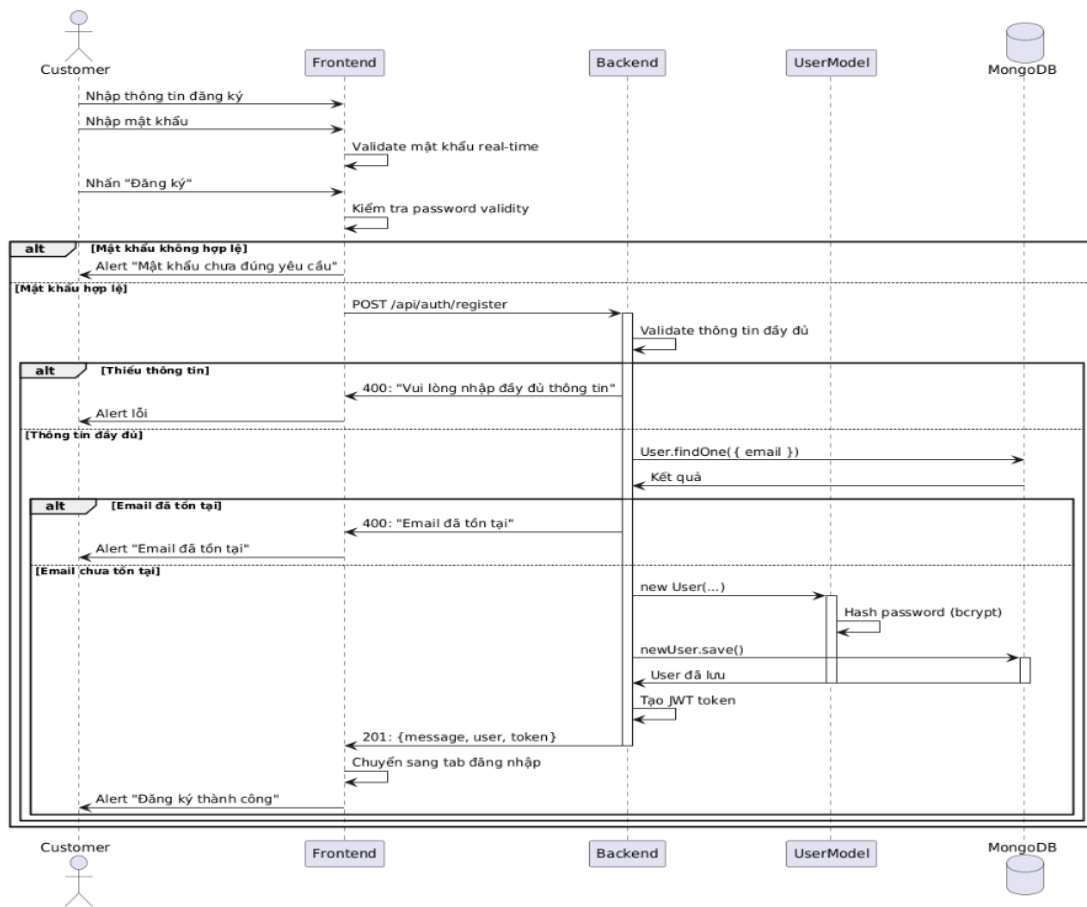
Tên Use Case	Đăng ký
Tác Nhân	Khách hàng
Điều Kiện đầu vào	Khách hàng chưa có tài khoản trong hệ thống. Hệ thống đang hoạt động bình thường.
Kết quả đầu ra	Khách hàng có tài khoản mới được lưu trong hệ thống. JWT token được tạo và trả về. Khách hàng có thể đăng nhập ngay sau khi đăng ký.
Mô tả	Use Case cho phép khách hàng nhập thông tin (họ tên, email, mật khẩu, số điện thoại) để tạo tài khoản. Hệ thống kiểm tra tính hợp lệ của mật khẩu (≥ 6 ký tự, chữ hoa đầu, ký tự đặc biệt, số), kiểm tra email đã tồn tại chưa. Nếu hợp lệ, hệ thống hash mật khẩu bằng bcryptjs, lưu vào cơ sở dữ liệu MongoDB, tạo JWT token (hết hạn 7 ngày) và thông báo đăng ký thành công. Nếu không hợp lệ, hệ thống hiển thị thông báo lỗi tương ứng.

- Dòng chính Use Case: Đăng ký

Bảng 3.5 Mô tả Use Case đăng kí

STT	Thực hiện	Hành động
1	Tác Nhân	Khách hàng truy cập vào website và chọn chức năng "Đăng ký".
2	Hệ Thống	Hiển thị form đăng ký (họ tên, email, số điện thoại, mật khẩu). Hệ thống validate mật khẩu real-time khi người dùng nhập (≥ 6 ký tự, chữ hoa đầu, ký tự đặc biệt, có số).
3	Tác Nhân	Nhập đầy đủ thông tin yêu cầu vào form đăng ký và nhấn nút "Đăng ký".
4	Hệ Thống	Kiểm tra tính hợp lệ của thông tin: email chưa tồn tại trong database, mật khẩu đáp ứng yêu cầu (≥ 6 ký tự, chữ hoa đầu, ký tự đặc biệt, có số), tất cả trường đều đã nhập.
5	Hệ Thống	Nếu thông tin hợp lệ, hash mật khẩu bằng bcryptjs (salt rounds: 10), tạo đối tượng User mới với provider="local", role="user", và lưu dữ liệu vào cơ sở dữ liệu MongoDB.
6	Hệ Thống	Tạo JWT token với thời hạn 7 ngày, hiển thị thông báo "Đăng ký thành công! Vui lòng đăng nhập" và tự động chuyển sang tab đăng nhập với email đã được điền sẵn.

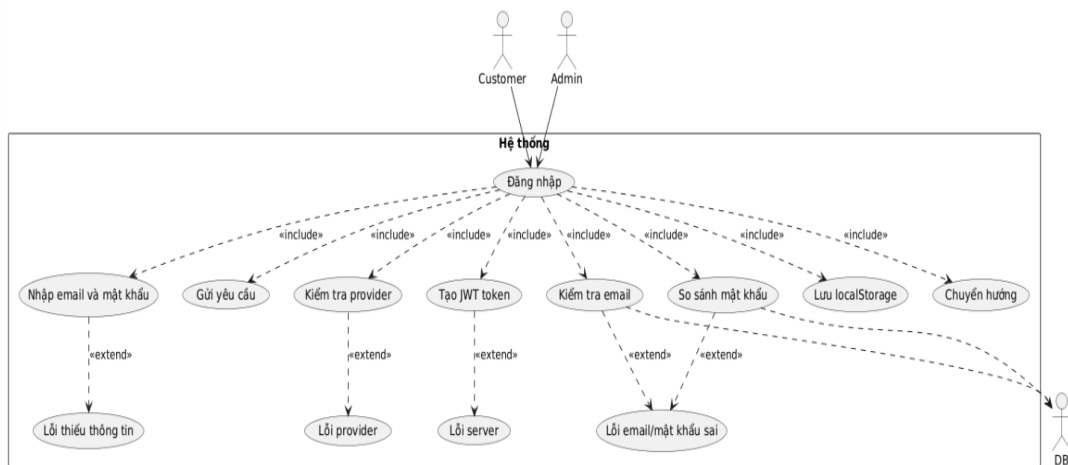
Biểu đồ tuần tự đăng kí :



Hình 3.3 Biểu đồ tuần tự đăng kí

3.3.2. Chức năng đăng nhập

- Biểu đồ Use Case đăng nhập :



Hình 3.4. Biểu đồ Use Case đăng nhập

- Mô tả Use Case đăng nhập

Bảng 3.6 Mô tả Use Case đăng nhập

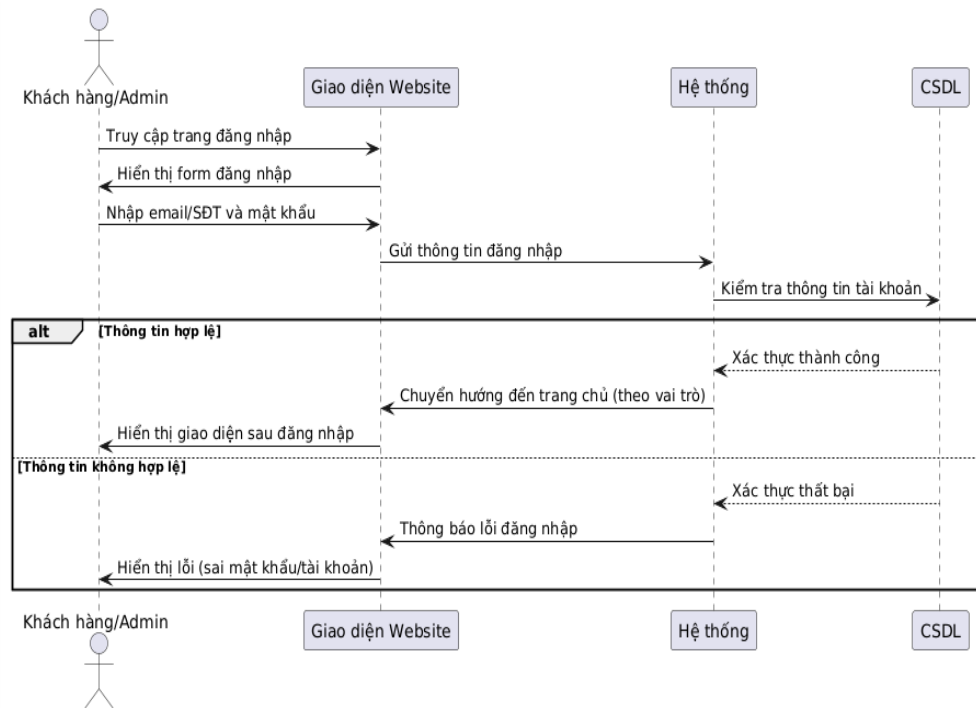
Tên Use Case	Đăng nhập
Tác nhân	Khách hàng, Admin
Điều kiện đầu vào	Khách hàng/Quản trị viên đã có tài khoản trong hệ thống. Hệ thống đang hoạt động bình thường.
Kết quả đầu ra	Khách hàng/Quản trị viên đăng nhập thành công. JWT token được tạo và lưu vào localStorage. Thông tin user được lưu vào localStorage. Hệ thống tự động chuyển hướng: Admin → /admin/dashboard, User → trang chủ.
Mô tả	Use Case cho phép khách hàng hoặc quản trị viên đăng nhập vào hệ thống bằng email và mật khẩu. Hệ thống kiểm tra email và mật khẩu có đầy đủ, kiểm tra email có tồn tại trong database, kiểm tra provider phải là "local" (không phải Google OAuth), so sánh mật khẩu bằng bcrypt. Nếu hợp lệ, hệ thống tạo JWT token (hết hạn 7 ngày), lưu user và token vào localStorage, thiết lập Authorization header, và chuyển hướng theo role. Nếu không hợp lệ, hệ thống hiển thị thông báo lỗi tương ứng.

- Dòng chính use case đăng nhập :

Bảng 3.7 Mô tả dòng chính Use case đăng nhập

STT	Thực hiện	Hành động
1	Tác nhân	Khách hàng/Quản trị viên truy cập vào website và chọn chức năng "Đăng nhập".
2	Hệ thống	Hiển thị form đăng nhập với các trường: Email và Mật khẩu.
3	Tác nhân	Nhập email và mật khẩu vào form đăng nhập và nhấn nút "Đăng nhập".
4	Hệ thống	Kiểm tra email và mật khẩu đã được nhập đầy đủ. Nếu thiếu → trả lời "Vui lòng nhập email và mật khẩu".
5	Hệ thống	Tìm kiếm user trong database theo email. Nếu không tìm thấy → trả lời "Email hoặc mật khẩu không đúng".
6	Hệ thống	Kiểm tra provider của user phải là "local" (không phải Google OAuth). Nếu provider khác → trả lời "Vui lòng đăng nhập bằng [provider]".
7	Hệ thống	So sánh mật khẩu người dùng nhập với mật khẩu đã hash trong database bằng bcrypt.compare(). Nếu không khớp → trả lời "Email hoặc mật khẩu không đúng".
8	Hệ thống	Nếu thông tin hợp lệ, tạo JWT token với payload { id: user.id } và thời hạn 7 ngày. 9 Hệ Thống Trả về response thành công: { message: "Đăng nhập thành công", user, token } với status 200. 10 Hệ Thống Lưu thông tin user và token vào localStorage. Thiết lập Authorization header cho các request tiếp theo. 11 Hệ Thống Chuyển hướng người dùng: Nếu role = "admin" → /admin/dashboard, nếu role = "user" → trang chủ. Hiển thị thông báo "Đăng nhập thành công".

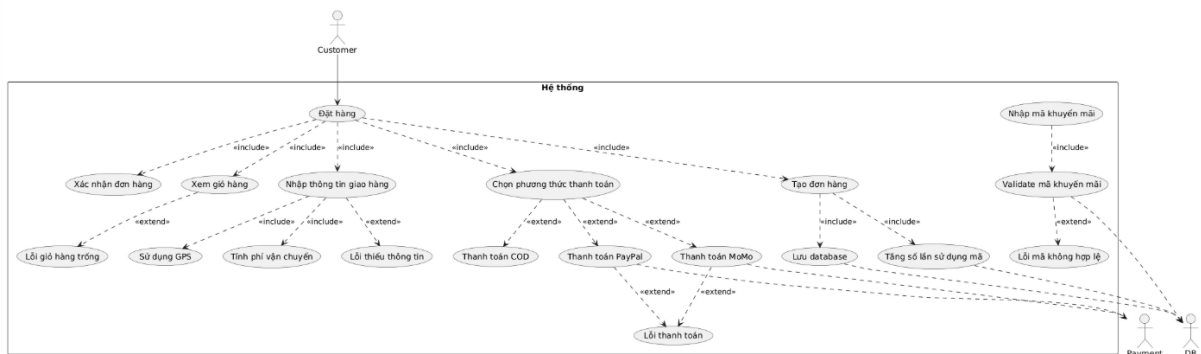
Biểu đồ tuần tự đăng nhập :



Hình 3.5 Biểu đồ tuần tự đăng nhập

3.3.3. Chức năng mua hàng

- Biểu đồ Use Case đặt hàng :



Hình 3.6 Biểu đồ Use Case đặt hàng

Mô tả use case đặt hàng :

Bảng 3.8 Mô tả Bảng Use case đặt hàng

Tên Use Case	Đặt hàng
Tác nhân	Khách hàng
Điều kiện đầu vào	Khách hàng đã có tài khoản và đăng nhập hệ thống. Có sản phẩm trong danh mục và còn hàng. Giỏ hàng có ít nhất một sản phẩm.
Kết quả đầu ra	Đơn hàng được tạo và lưu vào database. Nếu COD: đơn hàng có trạng thái "Xác nhận đơn hàng". Nếu PayPal/MoMo: chuyển hướng đến trang thanh toán. Mã khuyến mãi (nếu có) được tăng số lần sử dụng. Giỏ hàng được xóa.
Mô tả	Khách hàng nhập thông tin giao hàng (họ tên, số điện thoại, địa chỉ), có thể dùng GPS lấy địa chỉ, hệ thống tính phí vận chuyển. Có thể nhập và áp dụng mã khuyến mãi (hệ thống validate: tồn tại, hạn sử dụng, số lần sử dụng, đơn hàng tối thiểu). Chọn phương thức thanh toán (COD, PayPal, MoMo). Hệ thống validate thông tin, tạo đơn hàng và lưu vào database. Với COD: thông báo thành công và xóa giỏ hàng. Với PayPal/MoMo: tạo link thanh toán và chuyển hướng.

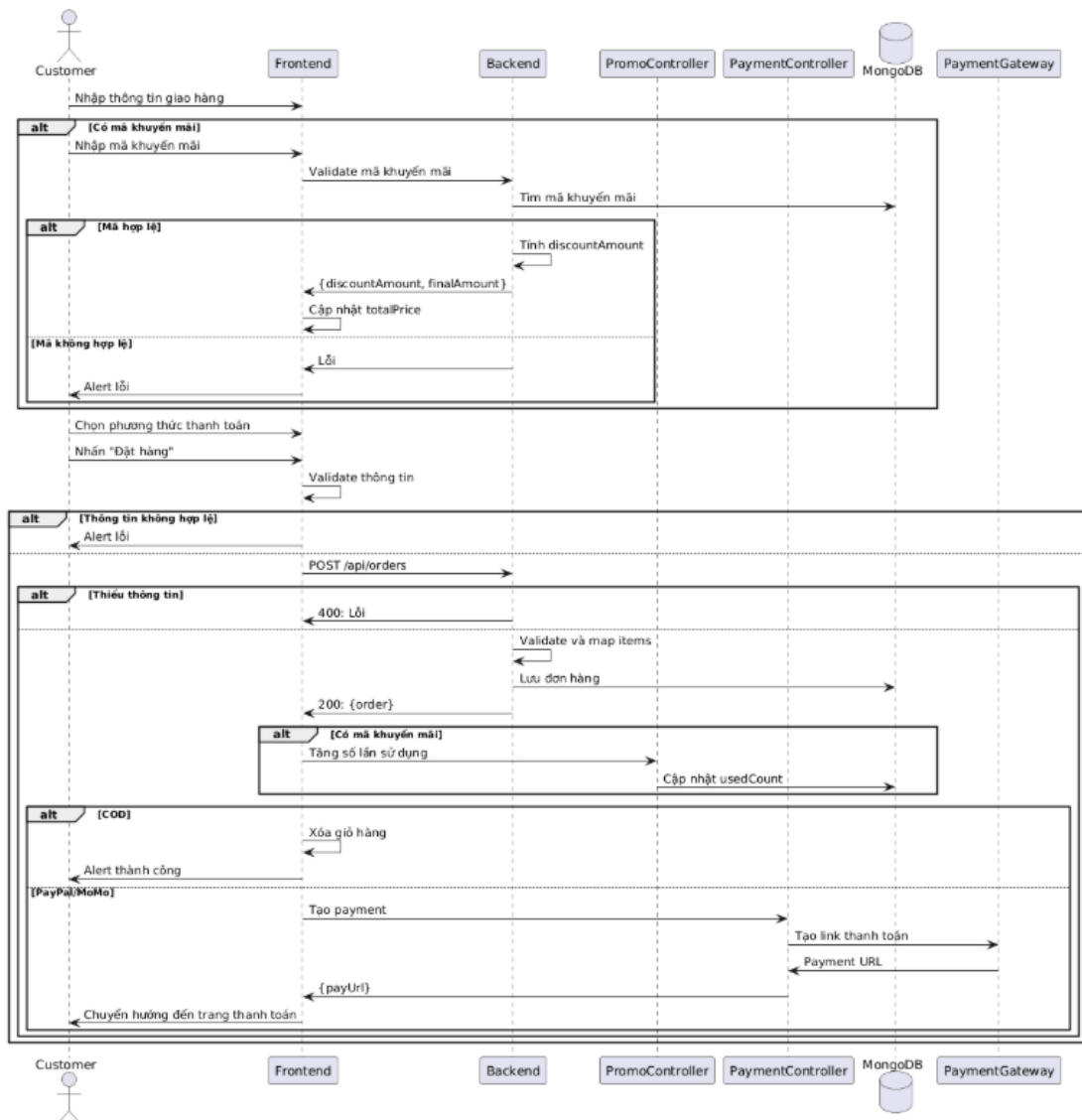
- Bảng dòng chính Use Case: Đặt hàng

Bảng 3.9 Mô tả dòng chính Use case đặt hàng

STT	Thực hiện	Hành động
1	Tác Nhân	Khách hàng truy cập vào trang thanh toán (Checkout) từ giỏ hàng.
2	Hệ Thống	Hiển thị danh sách sản phẩm trong giỏ hàng, tổng tiền, form nhập thông tin giao hàng và các phương thức thanh toán.
3	Tác Nhân	Nhập thông tin giao hàng: họ tên, số điện thoại, địa chỉ (có thể sử dụng GPS để lấy địa chỉ tự động), ghi chú (nếu có).
4	Hệ thống	Tính phí vận chuyển dựa trên địa chỉ giao hàng. Hiển thị tổng tiền bao gồm phí vận chuyển.
5	Tác Nhân	(Tùy chọn) Nhập mã khuyến mãi và nhấn "Áp dụng".
6	Hệ thống	Validate mã khuyến mãi: kiểm tra mã tồn tại, hạn sử dụng, số lần sử dụng, đơn hàng tối thiểu, quyền sử dụng. Tính số tiền giảm giá và cập nhật tổng tiền.
7	Tác Nhân	Chọn phương thức thanh toán (COD, PayPal, MoMo) và nhấn nút "Đặt hàng".
8	Hệ Thống	Validate thông tin: kiểm tra đầy đủ họ tên, số điện thoại (10 chữ số), địa chỉ. Kiểm tra giỏ hàng không rỗng. Nếu thiếu → hiển thị lỗi.
9	Hệ Thống	Tạo đơn hàng mới với thông tin: userId, items, info, subtotal, shippingFee, totalPrice, method, promoCode, discountAmount, status="Xác nhận đơn hàng", paymentStatus="unpaid".
10	Hệ Thống	Lưu đơn hàng vào database MongoDB. Nếu có mã khuyến mãi, tăng số lần sử dụng (usedCount) của mã.

11	Hệ Thống	Xử lý thanh toán theo phương thức đã chọn: - COD: Hiện thị thông báo "Đặt hàng thành công", xóa giỏ hàng, chuyển về trang chủ. - PayPal/MoMo: Tạo link thanh toán, chuyển hướng đến trang thanh toán của PayPal/MoMo.
----	----------	---

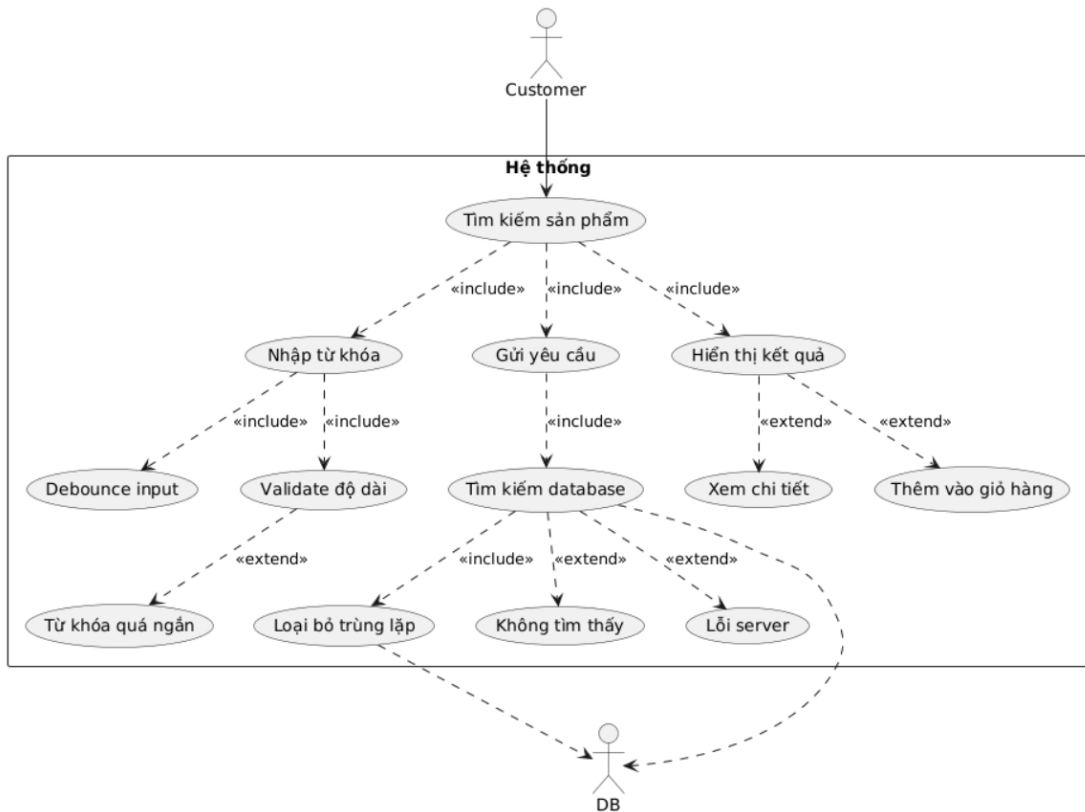
Biểu đồ tuần tự đặt hàng :



Hình 3.7 Biểu đồ tuần tự đặt hàng

3.3.4. Chức năng tìm kiếm

- Biểu đồ use case tìm kiếm :



Hình 3.8 Biểu đồ use case tìm kiếm

- Mô tả use case tìm kiếm :

Bảng 3.10 Mô tả use case tìm kiếm

STT	Thực hiện	Hành động
1	Tác nhân	Khách hàng nhập từ khóa tìm kiếm vào thanh tìm kiếm trên website.
2	Điều kiện đầu vào	Khách hàng đang ở trang web. Hệ thống đang hoạt động bình thường. Database có sản phẩm.
3	Kết quả đầu ra	Hiển thị danh sách sản phẩm phù hợp với từ khóa tìm kiếm trong dropdown. Khách hàng có thể xem chi tiết sản phẩm hoặc thêm vào giỏ hàng từ kết quả tìm kiếm.
4	Mô tả	Use Case cho phép khách hàng tìm kiếm sản phẩm bằng cách nhập từ khóa vào ô tìm kiếm. Hệ thống sử

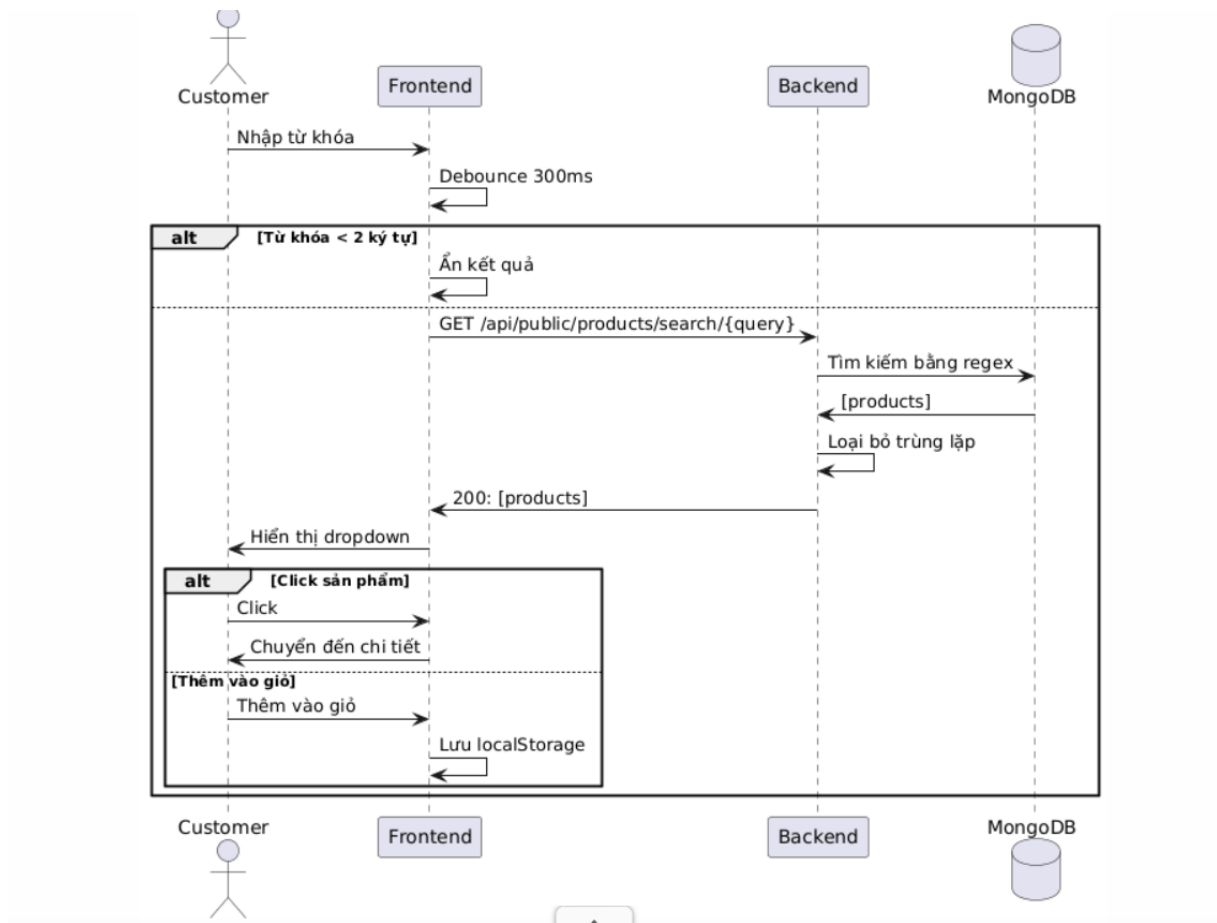
		dùng debounce (300ms) để tránh gọi API quá nhiều khi người dùng đang gõ. Khi từ khóa có độ dài ≥ 2 ký tự, hệ thống gửi yêu cầu tìm kiếm đến backend. Backend tìm kiếm trong database MongoDB sử dụng regex không phân biệt hoa thường, loại bỏ kết quả trùng lặp. Nếu tìm thấy, hiển thị danh sách sản phẩm trong dropdown với thông tin: tên, giá, hình ảnh. Khách hàng có thể click vào sản phẩm để xem chi tiết hoặc thêm vào giỏ hàng trực tiếp từ kết quả. Nếu không tìm thấy hoặc từ khóa quá ngắn, hiển thị thông báo tương ứng.
--	--	--

- Bảng dòng chính Use Case: Tìm kiếm

Bảng 3.11 Mô tả dòng chính use case: tìm kiếm

STT	Thực hiện	Hành động
1	Tác Nhân	Khách hàng nhập từ khóa vào ô tìm kiếm.
2	Hệ Thống	Kiểm tra độ dài từ khóa. Nếu < 2 ký tự $\rightarrow$ ẩn kết quả. Nếu ≥ 2 ký tự $\rightarrow$ thiết lập debounce 300ms.
3	Hệ Thống	Sau 300ms, gửi GET request đến <code>/api/public/products/search/{query}</code> .
4	Hệ thống	Hiển thị "🔍 Đang tìm kiếm..." trong dropdown.
5	Hệ thống	Backend tìm kiếm trong MongoDB bằng regex không phân biệt hoa thường, loại bỏ trùng lặp.
6	Hệ thống	Trả về danh sách sản phẩm hoặc thông báo "Không tìm thấy".
7	Hệ thống	Hiển thị dropdown với danh sách sản phẩm (tên, giá, hình ảnh) và số lượng kết quả.
8	Tác Nhân	(Tùy chọn) Click sản phẩm để xem chi tiết hoặc thêm vào giỏ hàng.

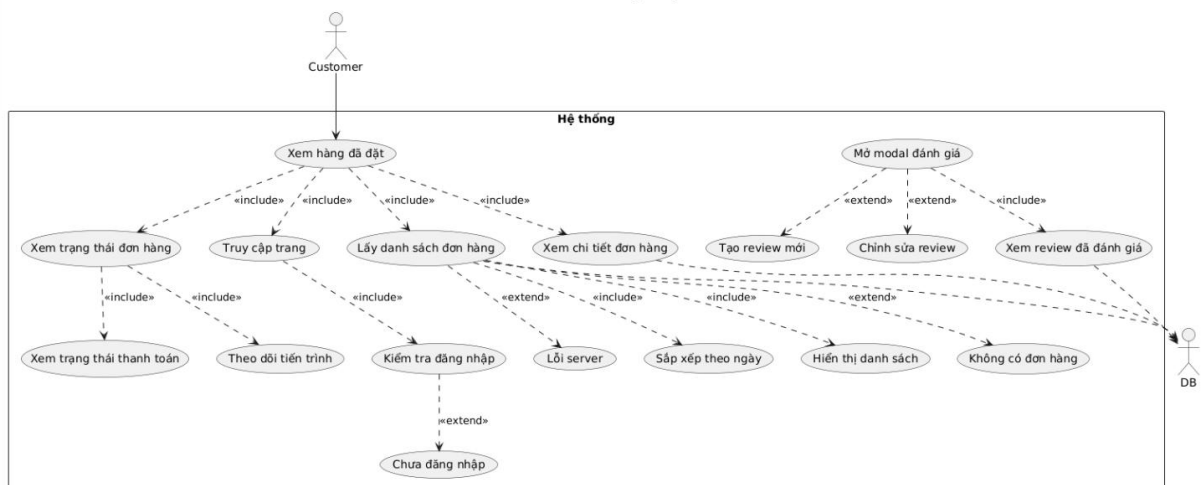
- Biểu đồ tuần tự tìm kiếm :



Hình 3.9 Biểu đồ tuần tự tìm kiếm

3.3.5. Chức năng xem hàng đã đặt

- Biểu đồ use case hàng đã đặt :



Hình 3.10 Biểu đồ use case hàng đã đặt

- Mô tả use case xem hàng đã đặt :

Bảng 3.11 Mô tả use case xem hàng đã đặt

Tên Use Case	Xem hàng đã đặt
Tác nhân	Khách hàng
Điều kiện đầu vào	Khách hàng đã đăng nhập. Hệ thống đang hoạt động bình thường.
Kết quả đầu ra	Khách hàng xem được danh sách tất cả đơn hàng đã đặt, sắp xếp theo thời gian mới nhất. Khách hàng xem được chi tiết từng đơn hàng (sản phẩm, thông tin giao hàng, tổng tiền). Khách hàng theo dõi được trạng thái đơn hàng và trạng thái thanh toán. Khách hàng có thể đánh giá sản phẩm nếu đơn hàng đã giao thành công.
Mô tả	Use Case cho phép khách hàng xem lịch sử đơn hàng đã đặt. Hệ thống kiểm tra khách hàng đã đăng nhập, lấy userId từ localStorage. Backend tìm kiếm tất cả đơn hàng của user trong database MongoDB, sắp xếp theo createdAt giảm dần (mới nhất trước). Hiện thị danh sách đơn hàng với thông tin: mã đơn hàng, ngày đặt, trạng thái đơn hàng (Xác nhận đơn hàng, Chuẩn bị đơn hàng, Đang giao hàng, Giao thành công, Đã hủy), trạng thái thanh toán (Đã thanh toán, Chưa thanh toán, Thanh toán thất bại, Thanh toán khi nhận hàng), tổng tiền, số lượng sản phẩm. Khách hàng có thể xem chi tiết từng đơn hàng: danh sách sản phẩm (tên, giá, số lượng, hình ảnh), thông tin giao hàng (họ tên, số điện thoại, địa chỉ, ghi chú), phí vận chuyển, mã khuyến mãi, số tiền giảm giá. Hệ thống hiện thị tiến trình đơn hàng với các bước: Xác nhận → Chuẩn bị → Đang giao → Giao thành công. Nếu đơn hàng đã giao thành công, khách hàng có thể đánh giá sản phẩm (xem review đã đánh giá, tạo review mới, chỉnh sửa review). Nếu chưa đăng nhập hoặc không có đơn hàng, hiện thị thông báo tương ứng.

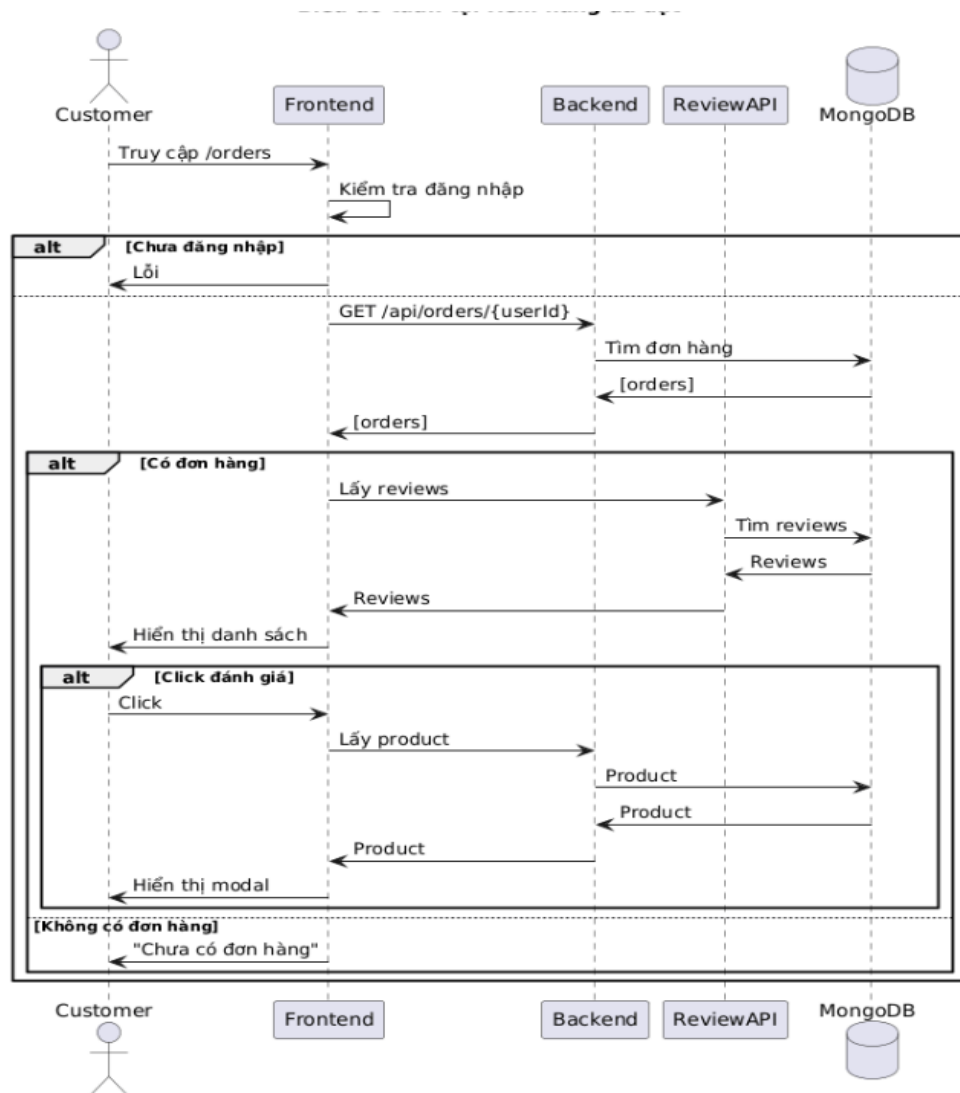
- Dòng chính use case xem hàng đã đặt :

Bảng 3.12 Mô tả dòng chính use case xem hàng đã đặt

STT	Thực hiện	Hành động
1	Khách hàng	Khách hàng truy cập vào trang "Lịch sử đơn hàng" (route /orders).
2	Hệ Thống	Kiểm tra khách hàng đã đăng nhập bằng cách lấy user từ localStorage. Nếu chưa đăng nhập → hiển thị lỗi "⚠️ Bạn cần đăng nhập để xem đơn hàng".
3	Hệ thống	Lấy userId từ user data trong localStorage.
4	Hệ Thống	Gửi GET request đến /api/orders/{userId} để lấy danh sách đơn hàng.
5	Hệ thống	Backend tìm kiếm tất cả đơn hàng của user trong MongoDB: Order.find({ userId }).sort({ createdAt: -1 }) (sắp xếp theo ngày mới nhất).
6	Hệ thống	Backend trả về danh sách đơn hàng (mảng) hoặc mảng rỗng nếu không có đơn hàng.
7	Hệ Thống	Frontend nhận kết quả, cập nhật state orders. Nếu không có đơn hàng → hiển thị "Bạn chưa có đơn hàng nào".
8	Hệ Thống	Với mỗi đơn hàng đã giao thành công, fetch reviews đã đánh giá từ API /api/reviews/user-review để hiển thị trạng thái đánh giá.
9	Hệ Thống	Hiển thị danh sách đơn hàng với thông tin: mã đơn hàng, ngày đặt, số lượng sản phẩm, tổng tiền, trạng thái đơn hàng, trạng thái thanh toán.
10	Tác Nhân	(Tùy chọn) Click vào đơn hàng để xem chi tiết: danh sách sản phẩm, thông tin giao hàng, phí vận chuyển, mã khuyến mãi, số tiền giảm giá.

11	Hệ Thống	Hiển thị tiến trình đơn hàng với các bước: Xác nhận đơn hàng → Chuẩn bị đơn hàng → Đang giao hàng → Giao thành công (nếu đã giao).
12	Tác Nhân	(Tùy chọn) Nếu đơn hàng đã giao thành công, click "Đánh giá" để mở modal đánh giá sản phẩm.
13	Hệ Thống	Hiển thị modal đánh giá với thông tin sản phẩm. Nếu đã có review → hiển thị review cũ để chỉnh sửa. Nếu chưa có → hiển thị form tạo review mới.

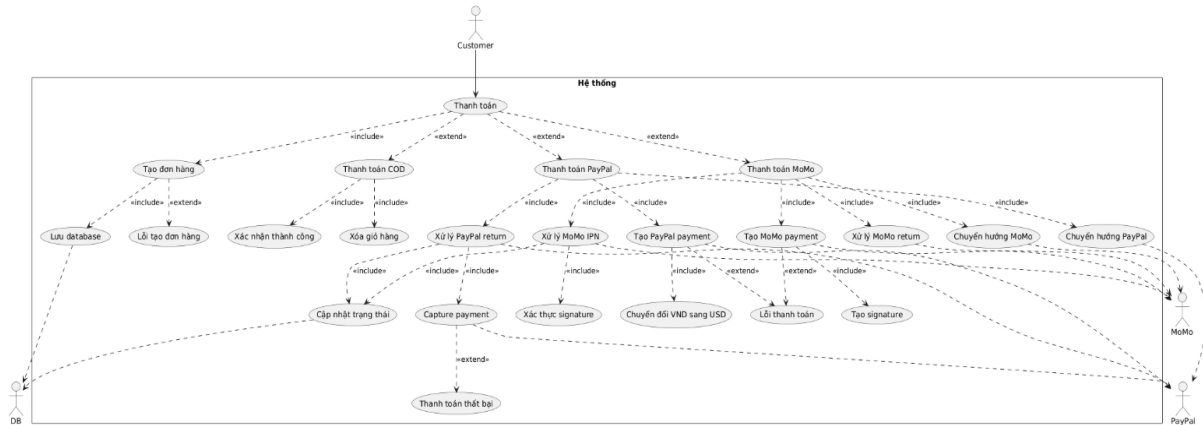
- Biểu đồ tuần tự xem hàng đã đặt :



Hình 3.11 Biểu đồ tuần tự xem hàng đã đặt

3.3.6. Chức năng thanh toán

- Biểu đồ use case thanh toán :



Hình 3.12 Biểu đồ use case thanh toán

- Mô tả use case thanh toán :

Bảng 3.13 Use case thanh toán

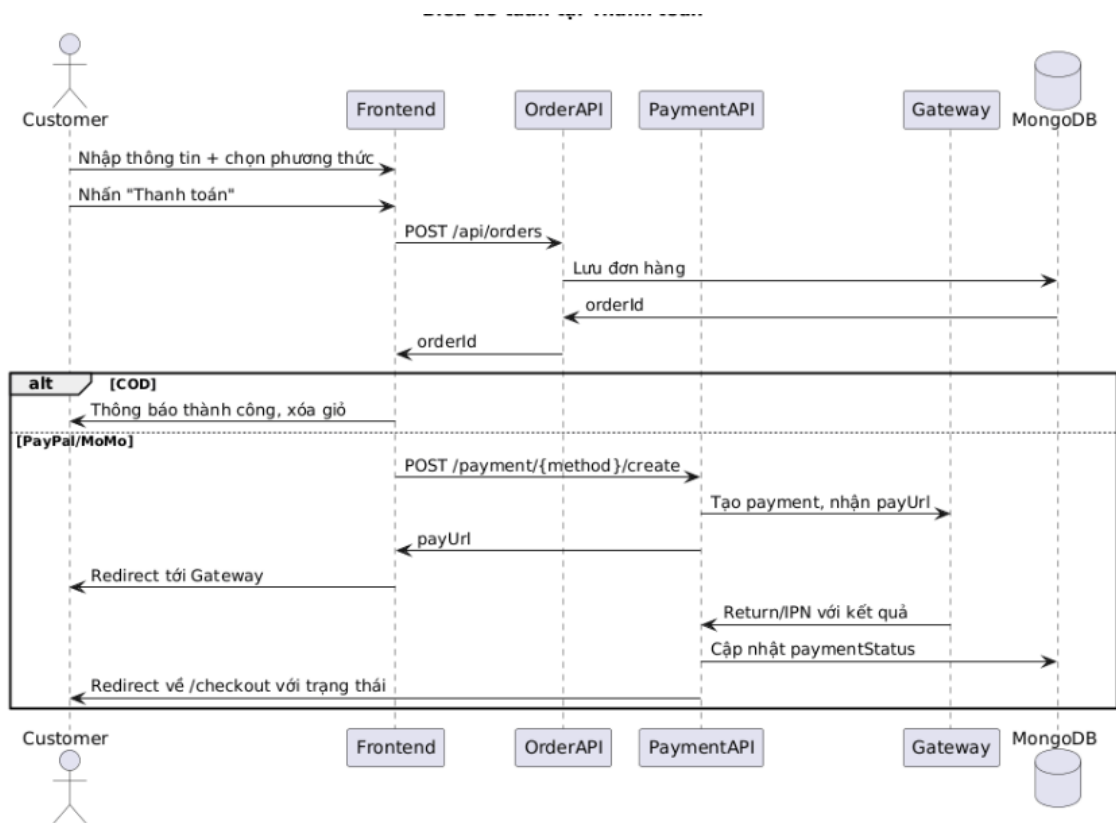
Tên Use Case	Thanh toán
Tác nhân	Khách hàng
Điều kiện đầu vào	Giỏ hàng có sản phẩm, đã nhập đủ thông tin giao hàng.
Kết quả đầu ra	Đơn hàng được tạo và lưu. COD: thông báo thành công, xóa giỏ hàng. PayPal/MoMo: chuyển đến cổng thanh toán, cập nhật trạng thái khi callback/IPN thành công.
Mô tả	Khách hàng xác nhận thông tin, hệ thống tạo đơn hàng (status “Xác nhận đơn hàng”, paymentStatus “unpaid”). Chọn COD → cảnh báo thành công và xóa giỏ. Chọn PayPal/MoMo → hệ thống tạo payment request, chuyển hướng người dùng, xử lý callback để xác nhận và cập nhật trạng thái thanh toán. Nếu lỗi, thông báo cho khách hàng.

- Dòng chính use case thanh toán :

Bảng 3.14 Mô tả dòng chính use case thanh toán

STT	Thực hiện	Hành động
1	Tác Nhân	Khách hàng tại trang thanh toán kiểm tra giỏ hàng, nhập đầy đủ thông tin giao hàng (họ tên, số điện thoại 10 số, địa chỉ).
2	Hệ thống	Validate thông tin và kiểm tra giỏ hàng. Nếu thiếu → hiển thị cảnh báo và dừng.
3	Hệ thống	Gọi API /api/orders để tạo đơn hàng với userId, items, info, subtotal, shippingFee, totalPrice, phương thức, mã giảm giá.
4	Hệ Thống	Lưu đơn hàng vào MongoDB (status “Xác nhận đơn hàng”, paymentStatus “unpaid”). Nếu có mã giảm giá, gọi /promo-code/increment-usage.
5	Tác Nhân	Chọn phương thức thanh toán: COD, PayPal hoặc MoMo.
6	Hệ thống	Nếu COD → thông báo “Đặt hàng thành công”, xóa giỏ hàng, chuyển về trang chủ.
7	Hệ thống	Nếu PayPal → gọi /payment/paypal/create, nhận payUrl và chuyển hướng khách hàng đến PayPal.
8	Hệ thống	Sau khi PayPal return/cancel: capture kết quả, cập nhật paymentStatus (paid/failed), điều hướng về /checkout với trạng thái tương ứng.
9	Hệ thống	Nếu MoMo → gọi /payment/momo/create, ký HMAC, nhận payUrl và chuyển hướng khách hàng đến MoMo.
10	Hệ Thống	Nhận return/IPN từ MoMo, xác thực chữ ký, cập nhật paymentStatus và điều hướng về /checkout với trạng thái tương ứng.

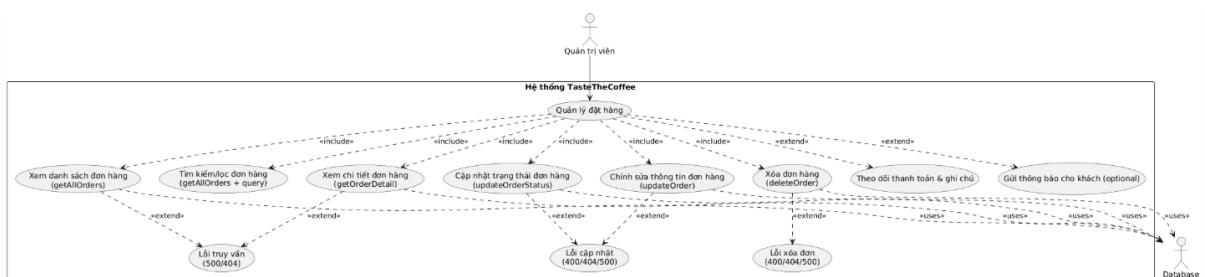
- Biểu đồ tuần tự thanh toán :



Hình 3.13 Biểu đồ tuần tự thanh toán

3.3.7. Chức năng quản lý đặt hàng

- Biểu đồ use case quản lý đặt hàng :



Hình 3.14 Biểu đồ use case quản lý đặt hàng

- Mô tả use case quản lí đặt hàng :

Bảng 3.15 Use case quản lí đặt hàng

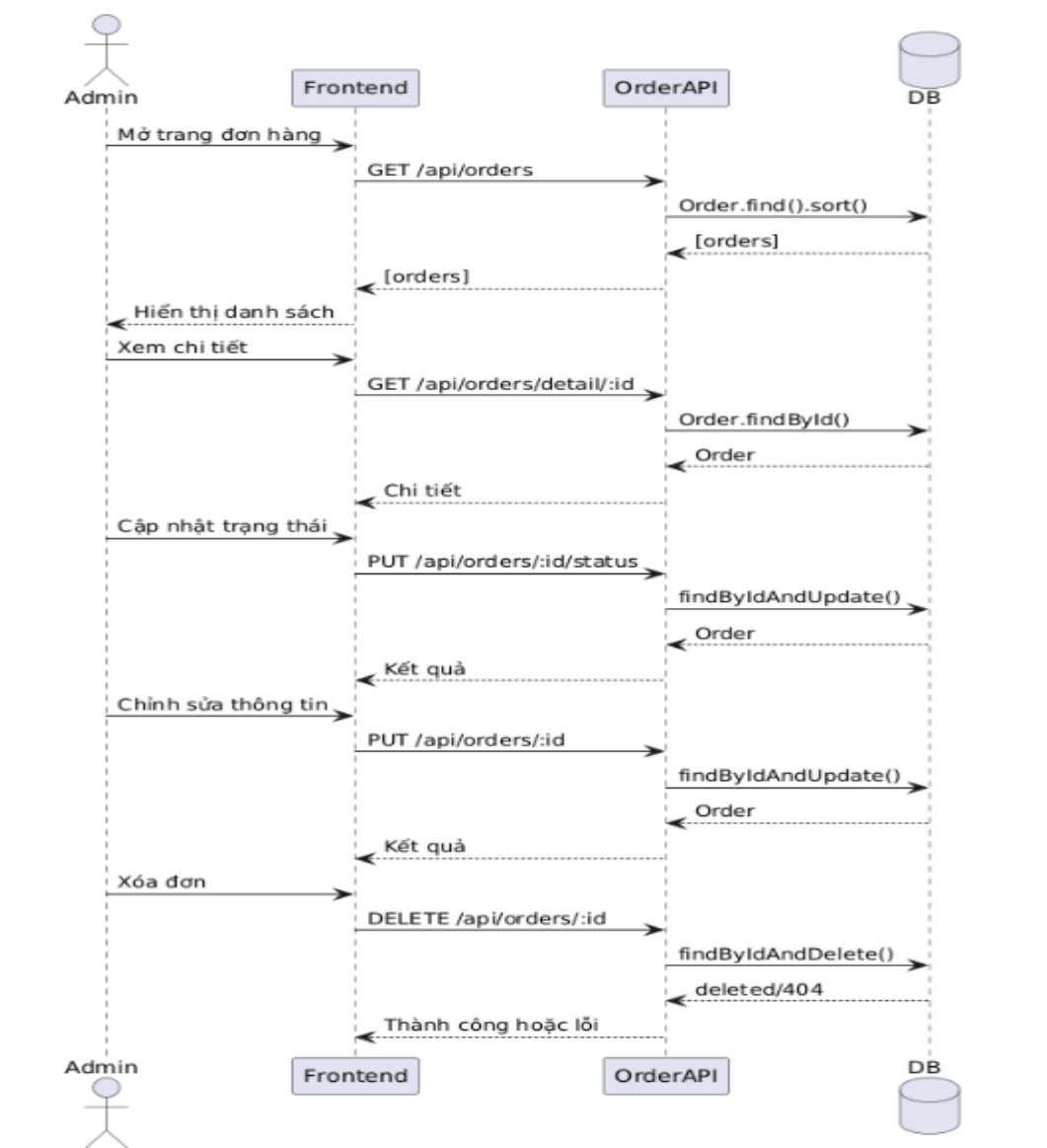
Tên Use Case	Xem hàng đã đặt
Tác nhân	Admin / Quản trị viên
Điều kiện đầu vào	Quản trị viên đã đăng nhập và có quyền truy cập trang quản trị. Cơ sở dữ liệu có đơn hàng.
Kết quả đầu ra	Quản trị viên xem được danh sách/từng đơn hàng, cập nhật trạng thái & thông tin thanh toán, chỉnh sửa nội dung đơn hàng, hoặc xóa đơn hàng. Mọi thay đổi được lưu lại trong MongoDB và trả về kết quả thành công/ lỗi tương ứng.
Mô tả	Use Case hỗ trợ quản trị viên theo dõi và điều chỉnh đơn hàng. Hệ thống cung cấp các chức năng: xem toàn bộ đơn hàng (getAllOrders), xem chi tiết (getOrderDetail), cập nhật trạng thái & paymentStatus (updateOrderStatus), chỉnh sửa thông tin (items, địa chỉ, tổng tiền, phương thức, mã khuyến mãi) qua updateOrder, và xóa đơn hàng (deleteOrder). Mỗi thao tác đều ghi log, validate ID hợp lệ, phản hồi lỗi 400/404/500 nếu dữ liệu sai hoặc đơn không tồn tại.

- Dòng chính use case quản lý đặt hàng :

Bảng 3.16 Mô tả dòng chính use case quản lý đặt hàng

STT	Thực hiện	Hành động
1	Quản trị viên	Đăng nhập vào trang quản trị và mở module “Đơn hàng”.
2	Hệ thống	Gọi API GET /api/admin/orders (hoặc /api/orders) để lấy danh sách toàn bộ đơn hàng (sắp xếp theo ngày tạo giảm dần).
3	Hệ thống	Hiển thị bảng đơn hàng với thông tin chính (mã đơn, khách hàng, trạng thái, thanh toán, tổng tiền).
4	Quản trị viên	Chọn một đơn hàng để xem chi tiết.
5	Hệ thống	Gọi GET /api/orders/detail/:orderId và hiển thị đầy đủ thông tin: sản phẩm, địa chỉ giao, tổng tiền, trạng thái hiện tại.
6	Quản trị viên	(Tùy chọn) Cập nhật trạng thái/ thanh toán (ví dụ: “Chuẩn bị”, “Đang giao”, “Paid”).
7	Hệ thống	Gọi PUT /api/orders/:orderId/status với dữ liệu mới; cập nhật vào MongoDB và trả về kết quả.
8	Quản trị viên	(Tùy chọn) Chỉnh sửa thông tin đơn hàng: sản phẩm, địa chỉ, phương thức, mã khuyến mãi.
9	Hệ thống	Gọi PUT /api/orders/:id để cập nhật nội dung đơn, validate dữ liệu, lưu lại và trả kết quả.
10	Quản trị viên	(Tùy chọn) Xóa đơn hàng (đã hủy/đơn test).
11	Hệ thống	Gọi DELETE /api/orders/:id, kiểm tra hợp lệ, xóa khỏi DB và phản hồi “ ☒ Xóa đơn hàng thành công”.

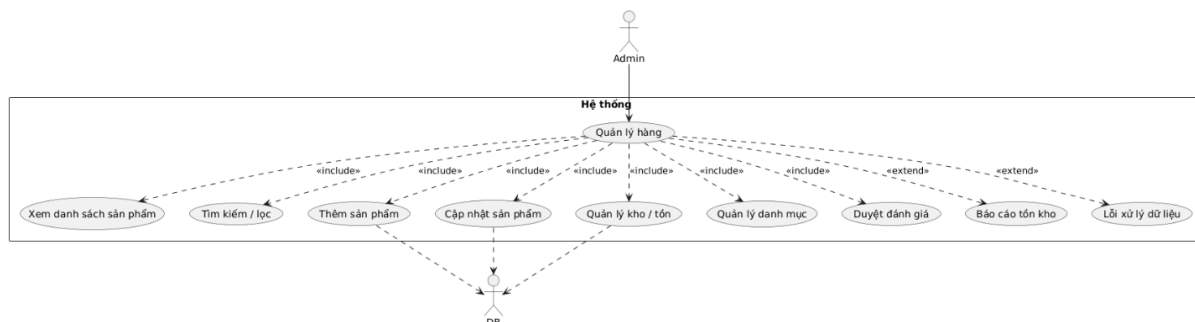
- Biểu đồ tuần tự quản lí đặt hàng :



Hình 3.15. Biểu đồ tuần tự quản lí đặt hàng

3.3.8. Chức năng quản lý hàng

- Biểu đồ use case quản lý hàng :



Hình 3.16 Biểu đồ use case quản lý hàng

- Mô tả use case quản lý hàng :

Bảng 3.17 Use case quản lý đặt hàng

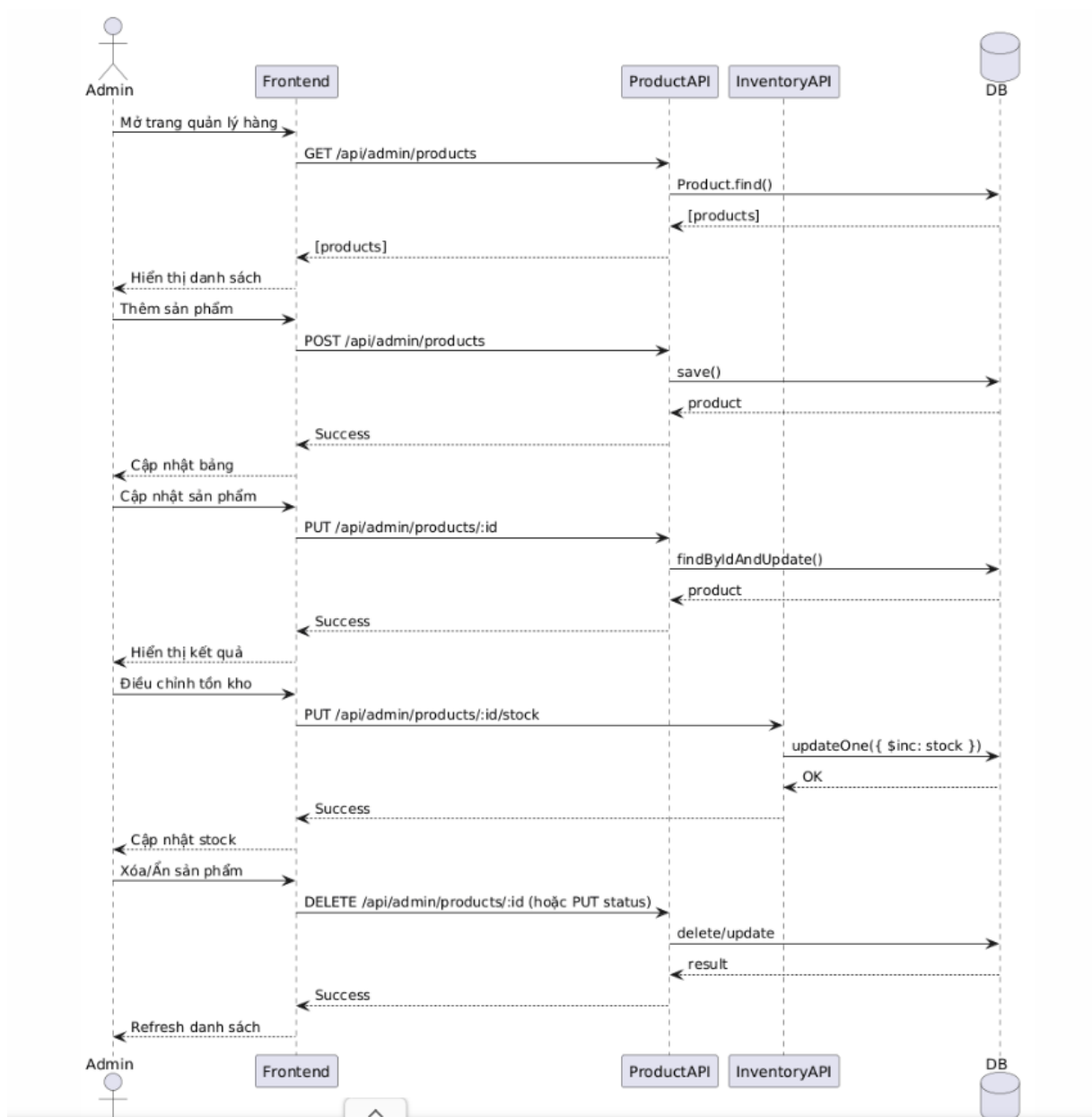
Tên Use Case	Quản lý hàng
Tác nhân	Admin / Quản trị viên
Điều kiện đầu vào	Quản trị viên đã đăng nhập vào hệ thống admin. CSDL đang hoạt động và đã có dữ liệu sản phẩm/kho.
Kết quả đầu ra	Sản phẩm được thêm/cập nhật/xóa đúng yêu cầu; thông tin kho, danh mục, đánh giá được cập nhật; dữ liệu lưu về MongoDB và phản hồi thành công hoặc thông báo lỗi nếu có vấn đề.
Mô tả	Use Case cho phép quản trị viên quản lý toàn bộ hàng hóa trên TasteTheCoffee: xem/tìm kiếm sản phẩm, thêm mới, cập nhật thông tin (giá, số lượng, mô tả, hình ảnh, danh mục), điều chỉnh tồn kho, quản lý danh mục-thuộc tính, duyệt hoặc xóa đánh giá sản phẩm, theo dõi báo cáo tồn kho. Mọi thao tác gọi API backend (productController, inventory/warehouse, reviewController) và ghi nhận thay đổi trong MongoDB, đồng thời trả thông báo thành công hoặc lỗi về giao diện admin.

- Dòng chính use case quản lý hàng :

Bảng 3.18 Mô tả dòng chính use case quản lý hàng

STT	Thực hiện	Hành động
1	Quản trị viên	Đăng nhập vào trang admin và mở module “Quản lý hàng”.
2	Hệ thống	Gọi API (ví dụ GET /api/admin/products) để tải danh sách sản phẩm, tồn kho, danh mục -> hiển thị bảng.
3	Quản trị viên	Chọn thao tác: xem chi tiết, tìm kiếm/lọc, hoặc chỉnh sửa một sản phẩm.
4	Hệ thống	(Nếu xem/ lọc) Trả về kết quả sản phẩm phù hợp, cập nhật bảng hiển thị.
5	Quản trị viên	(Thêm mới) Nhập thông tin sản phẩm, tồn kho, danh mục, hình ảnh -> nhấn “Lưu”.
6	Hệ thống	Gọi POST /api/admin/products (hoặc API tương ứng) để lưu sản phẩm mới vào MongoDB -> trả thông báo thành công/lỗi.
7	Quản trị viên	(Cập nhật) Chỉnh sửa thông tin sản phẩm/tồn kho/danh mục -> nhấn “Cập nhật”.
8	Hệ thống	Gọi PUT /api/admin/products/:id để lưu thay đổi vào DB -> phản hồi kết quả.
9	Quản trị viên	(Tùy chọn) Duyệt hoặc xóa đánh giá/ẩn sản phẩm nếu cần.
10	Hệ thống	Gọi API tương ứng (PUT/DELETE) để áp dụng thay đổi -> thông báo kết quả.

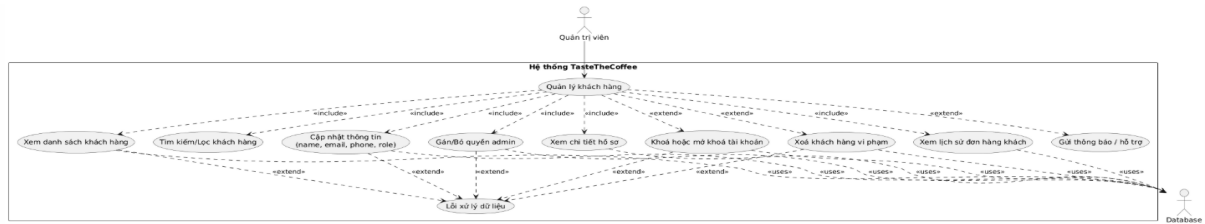
- Biểu đồ trình tự quản lý hàng :



Hình 3.17. Biểu đồ trình tự quản lý hàng

3.3.9. Chức năng quản lý khách hàng

- Biểu đồ use case quản lý khách hàng :



Hình 3.18 Biểu đồ use case quản lý khách hàng

- Mô tả use case quản lý khách hàng :

Bảng 3.19 Use case quản lý khách hàng

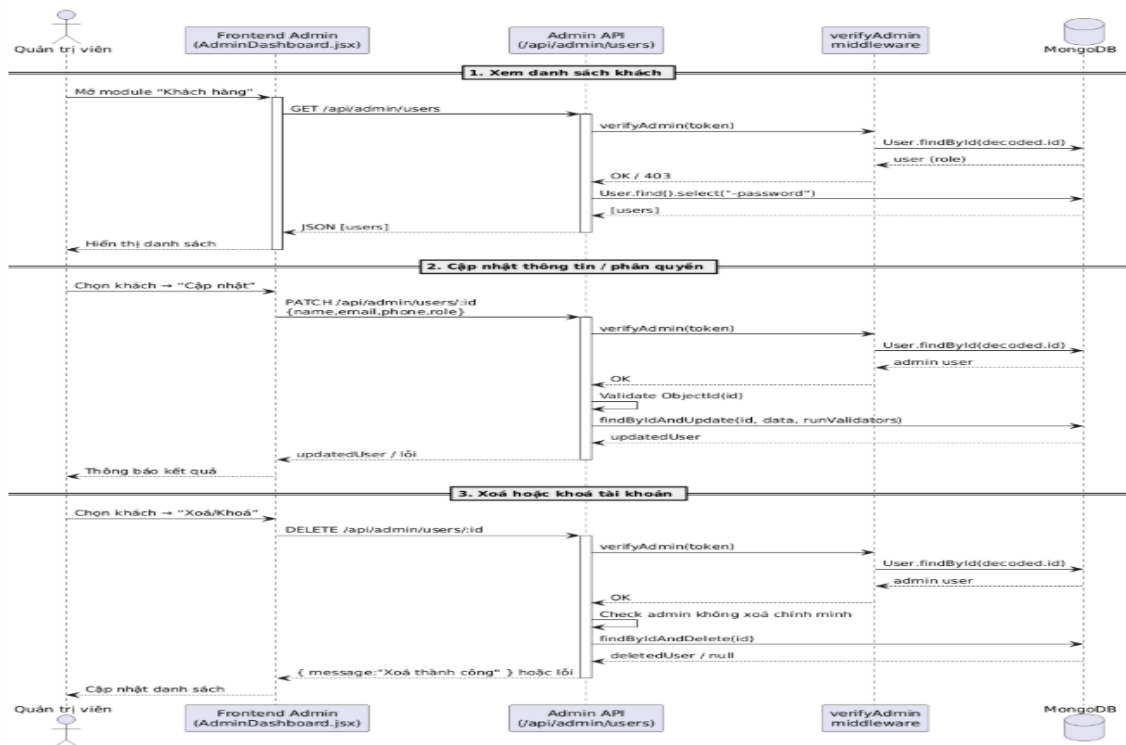
Tên Use Case	Quản lý khách hàng
Tác nhân	Admin / Quản trị viên
Điều kiện đầu vào	Quản trị viên đã đăng nhập vào trang admin; hệ thống hoạt động bình thường; dữ liệu người dùng tồn tại trong MongoDB.
Kết quả đầu ra	Danh sách khách hàng được hiển thị, thông tin cá nhân có thể được cập nhật, quyền (role) có thể thay đổi, tài khoản có thể khóa/xóa; mọi thay đổi lưu về DB và báo trạng thái thành công/lỗi.
Mô tả	Use Case cho phép quản trị viên xem danh sách khách hàng (GET /api/admin/users), tìm kiếm/lọc, xem hồ sơ chi tiết, cập nhật thông tin hoặc phân quyền (PATCH /users/:id), khóa/xóa tài khoản vi phạm (DELETE /users/:id). Admin cũng có thể xem lịch sử đơn hàng của từng khách để hỗ trợ. Hệ thống kiểm tra quyền admin thông qua middleware verifyAdmin, validate ObjectId, trả thông báo lỗi nếu dữ liệu không hợp lệ hoặc người dùng không tồn tại.

- Dòng chính Use case quản lý Khách Hàng :

Bảng 3.20 Mô tả dòng chính use case quản lý khách hàng

STT	Thực hiện	Hành động
1	Quản trị viên	Đăng nhập admin và mở module “Khách hàng”.
2	Hệ thống	Gọi GET /api/admin/users (verifyAdmin) để lấy danh sách, loại bỏ password và hiển thị bảng.
3	Quản trị viên	Tìm kiếm/lọc khách hàng hoặc chọn một khách để xem hồ sơ.
4	Hệ thống	Trả về thông tin chi tiết khách (hồ sơ, đơn hàng liên quan).
5	Quản trị viên	Cập nhật thông tin (name, email, phone, role) hoặc gán/bỏ quyền admin.
6	Hệ thống	Gọi PATCH /api/admin/users/:id, validate ObjectId, lưu vào MongoDB, phản hồi kết quả.
7	Quản trị viên	(Tùy chọn) Khoá/xoá khách vi phạm
8	Hệ thống	Gọi DELETE /api/admin/users/:id, kiểm tra không tự xoá chính mình, thông báo thành công hoặc lỗi.

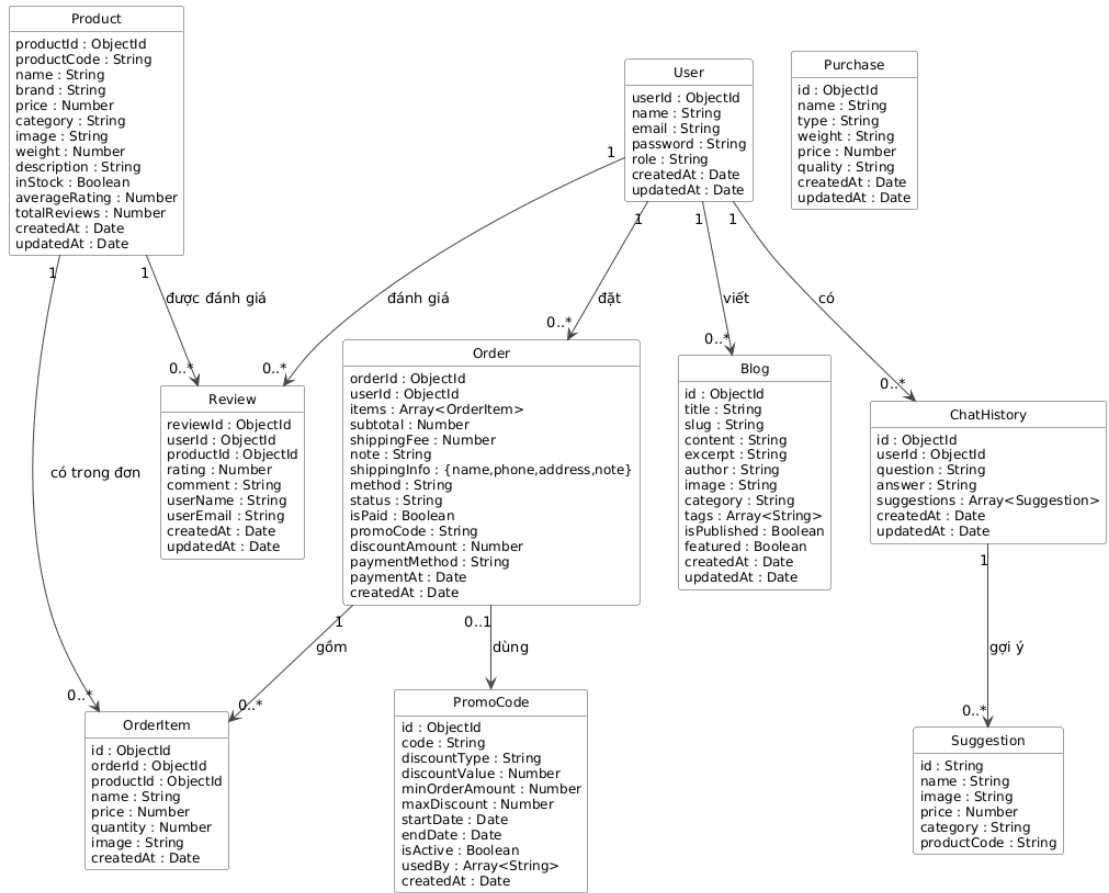
- Biểu đồ tuần tự quản lý khách hàng :



Hình 3.19 Biểu đồ tuần tự quản lý khách hàng

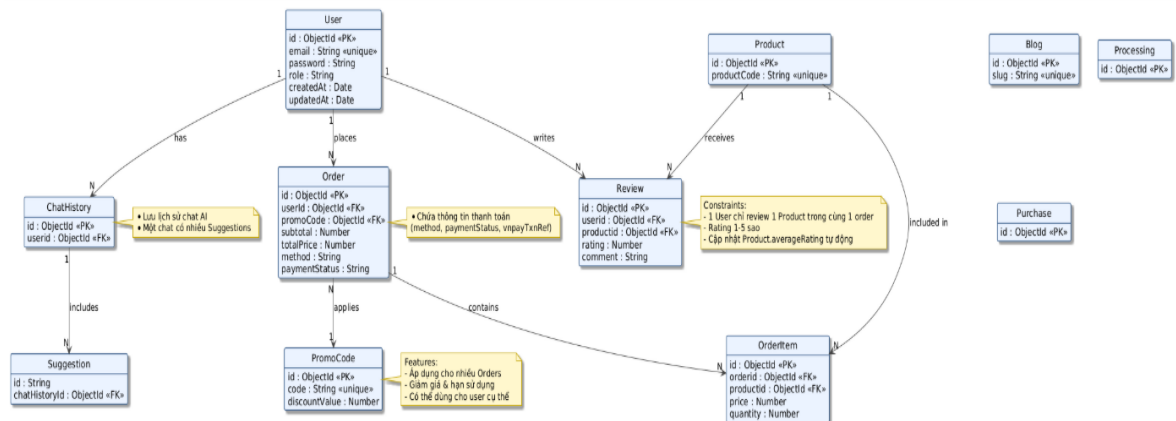
3.4. Thiết kế cơ sở dữ liệu

3.4.1. Mô hình thực thể liên kết



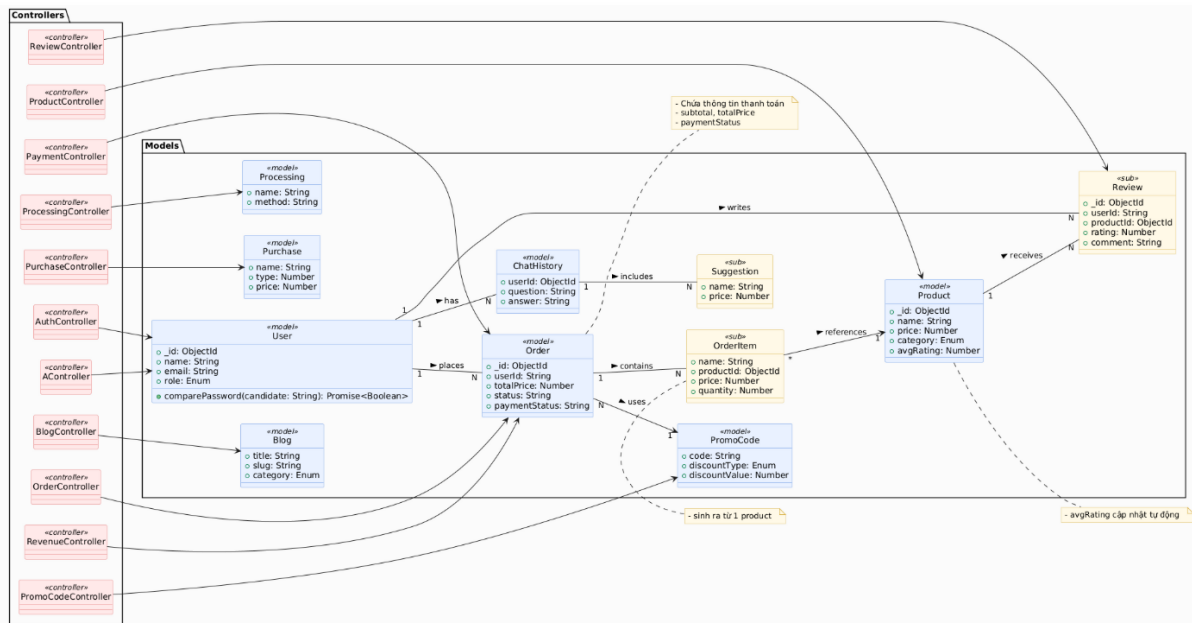
Hình 3.20 Mô hình thực thể liên kết

3.4.2 Mô hình quan hệ



Hình 3.21 Mô hình quan hệ

3.4.3. Mô hình lớp



Hình 3.22 Mô hình lớp

3.5 Danh mục các bảng

- Mô tả các bảng cơ sở dữ liệu :

Bảng 3.21 Bảng cơ sở dữ liệu

TT	Tên Bảng	Mô tả
1	User	Đăng nhập/đăng ký, phân quyền, Google OAuth
2	Product	Catalog sản phẩm, hình ảnh, thương hiệu, tồn kho
3	Order	Đơn hàng + COD + MOMO
4	PromoCode	Mã giảm giá, điều kiện, thời hạn
5	Processing	Gia công coffee (public + admin)
6	Purchase	Thu mua coffee (public + admin)
7	Blog	Bài viết / SEO / Content admin
8	ChatHistory	Lưu tương tác AI + gợi ý sản phẩm

9	Review	Đánh giá,mô tả chất lượng
---	--------	---------------------------

Bảng 3.22 Bảng mô tả user

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
_id	ObjectId	NOT NULL	PRIMARY KEY	ID người dùng
name	String	NOT NULL		Họ tên
email	String	NOT NULL	UNIQUE	Email đăng nhập
password	String	NULL		Mật khẩu (null khi đăng nhập Google)
phone	String	NOT NULL (khi provider = "local")		Số điện thoại
provider	String	NOT NULL		Kiểu đăng nhập: "local" hoặc "google"
providerId	String	NULL		Google sub ID nếu OAuth
role	String	NOT NULL		Quyền: "user", "admin"
createdAt	Date	NOT NULL		Tự tạo bởi MongoDB
updatedAt	Date	NOT NULL		Cập nhật lần cuối

Bảng 3.23 Bảng mô tả Product

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
_id	ObjectId	NO	PRIMARY	Khóa chính tự động của MongoDB
productCode	String	NO	UNIQUE	Mã sản phẩm, bắt buộc, duy nhất, trim
name	String	NO		Tên sản phẩm, bắt buộc, trim
brand	String	NO		Thương hiệu, bắt buộc, trim
price	Number	NO		Giá sản phẩm, bắt buộc, ≥ 0
category	String	NO		Loại sản phẩm, bắt buộc, chỉ nhận: "ca-phe-tien-loi", "ca-phe-truyen-thong", "ca-phe-nguyen-chat"
weight	String	YES		Trọng lượng sản phẩm, mặc định ""
description	String	YES		Mô tả sản phẩm, mặc định ""
image	String	YES		URL ảnh sản phẩm, mặc định ""
inStock	Boolean	YES		Còn hàng hay hết hàng, mặc định true
createdAt	Date	NO		Thời gian tạo sản phẩm, tự động từ timestamps
updatedAt	Date	NO		Thời gian cập nhật sản phẩm, tự động từ timestamps

Bảng 3.24 Bảng mô tả Order

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
_id	ObjectId	NO	PRIMARY	Khóa chính tự động của MongoDB
userId	String	NO		ID người dùng, bắt buộc
items	Array of Object	NO		Danh sách sản phẩm trong đơn hàng
items.name	String	YES		Tên sản phẩm
items.price	Number	YES		Giá sản phẩm
items.quantity	Number	YES		Số lượng sản phẩm
items.image	String	YES		URL ảnh sản phẩm
info	Object	YES		Thông tin nhận hàng
info.fullName	String	YES		Họ tên người nhận
info.phone	String	YES		Số điện thoại người nhận
info.address	String	YES		Địa chỉ giao hàng
info.note	String	YES		Ghi chú thêm
subtotal	Number	YES		Tổng tiền trước khi giảm giá và phí ship, mặc định 0
shippingFee	Number	YES		Phí vận chuyển, mặc định 0
totalPrice	Number	NO		Tổng tiền cuối cùng (subtotal + shippingFee - discountAmount)
method	String	YES		Phương thức thanh toán: "cod" hoặc "vnpay"
promoCode	String	YES		Mã khuyến mãi đã sử dụng, mặc định null
discountAmount	Number	YES		Số tiền giảm giá, mặc định 0

status	String	YES		Trạng thái đơn hàng, mặc định "Đang xử lý"
paymentStatus	String	YES		Trạng thái thanh toán: "paid", "unpaid", "failed", mặc định "unpaid"
vnpayTxnRef	String	YES		Mã giao dịch VNPay, mặc định null

Bảng 3.25 Bảng mô tả PromoCode

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
_id	ObjectId	NO	PRIMARY	Khóa chính tự động của MongoDB
code	String	NO	UNIQUE	Mã khuyến mãi, bắt buộc, duy nhất, trim, tự chuyển sang chữ hoa
description	String	YES		Mô tả mã khuyến mãi, mặc định ""
discountType	String	NO		Loại giảm giá: "percentage" (giảm %) hoặc "fixed" (giảm số tiền cố định)
discountValue	Number	NO		Giá trị giảm, ≥ 0
minOrderAmount	Number	YES		Đơn hàng tối thiểu để áp dụng mã, mặc định 0
maxDiscountAmount	Number	YES		Số tiền giảm tối đa (dành cho discountType = percentage), mặc định null

startDate	Date	YES		Ngày bắt đầu áp dụng mã, mặc định Date.now
endDate	Date	NO		Ngày kết thúc áp dụng mã, bắt buộc
usageLimit	Number	YES		Giới hạn số lần sử dụng, null = không giới hạn
usedCount	Number	YES		Số lần đã sử dụng, mặc định 0
isActive	Boolean	YES		Trạng thái mã, mặc định true
applicableUsers	Array of String	YES		Danh sách userId được áp dụng, rỗng = tất cả
createdAt	Date	NO		Thời gian tạo bản ghi, tự động từ timestamps
updatedAt	Date	NO		Thời gian cập nhật bản ghi, tự động từ timestamps

Bảng 3.26 Bảng mô tả Processing

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
_id	ObjectId	NO	PRIMARY	Khóa chính tự động của MongoDB
name	String	NO		Tên phương pháp gia công, bắt buộc, trim
method	String	NO		Cách thực hiện/gia công, bắt buộc, trim
description	String	YES		Mô tả chi tiết, mặc định ""

createdAt	Date	NO		Thời gian tạo bản ghi, tự động từ timestamps
updatedAt	Date	NO		Thời gian cập nhật bản ghi, tự động từ timestamps

Bảng 3.27 Bảng mô tả purchase

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
name	String	NO		Tên sản phẩm, bắt buộc, loại bỏ khoảng trắng đầu/cuối
type	String	YES		Loại sản phẩm, chỉ nhận "tươi" hoặc "khô", mặc định "tươi", tự chuyển về chữ thường và trim
weight	String	NO		Trọng lượng sản phẩm, bắt buộc; giá trị hợp lệ là số hoặc dạng "min-max" ($\text{min} \leq \text{max}$)
price	Number	NO		Giá thu mua, bắt buộc, giá trị ≥ 0
quality	String	NO		Chất lượng, bắt buộc; chỉ nhận "loại 1", "loại 2", "đặc biệt", tự chuyển về chữ thường và trim
createdAt	Date	NO		Thời gian tạo bản ghi, tự động

Bảng 3.28 Bảng mô tả Blog

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
_id	ObjectId	NO	PRIMARY	Khóa chính tự động của MongoDB
title	String	NO		Tiêu đề blog, bắt buộc, trim
slug	String	YES	UNIQUE	Đường dẫn SEO, tự sinh từ title, lowercase, trim
content	String	NO		Nội dung blog, bắt buộc, trim

excerpt	String	YES		Tóm tắt ngắn gọn, mặc định "", tối đa 300 ký tự
author	String	YES		Tác giả, mặc định "Admin", trim
image	String	YES		URL ảnh đại diện blog, mặc định ""
category	String	YES		Chuyên mục blog, enum: "tin-tuc", "huong-dan", "cong-thuc", "kinh-nghiem", "san-pham", "khac"; mặc định "tin-tuc"
tags	Array of String	YES		Danh sách tag, mặc định rỗng
isPublished	Boolean	YES		Trạng thái xuất bản, mặc định false
views	Number	YES		Số lượt xem blog, mặc định 0
featured	Boolean	YES		Blog nổi bật, mặc định false
createdAt	Date	NO		Thời gian tạo blog, tự động từ timestamps
updatedAt	Date	NO		Thời gian cập nhật blog, tự động từ timestamps

Bảng 3.29 Bảng mô tả ChatHistory

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
_id	ObjectId	NO	PRIMARY	Khóa chính tự động của MongoDB
userId	ObjectId	NO		Tham chiếu đến User, bắt buộc

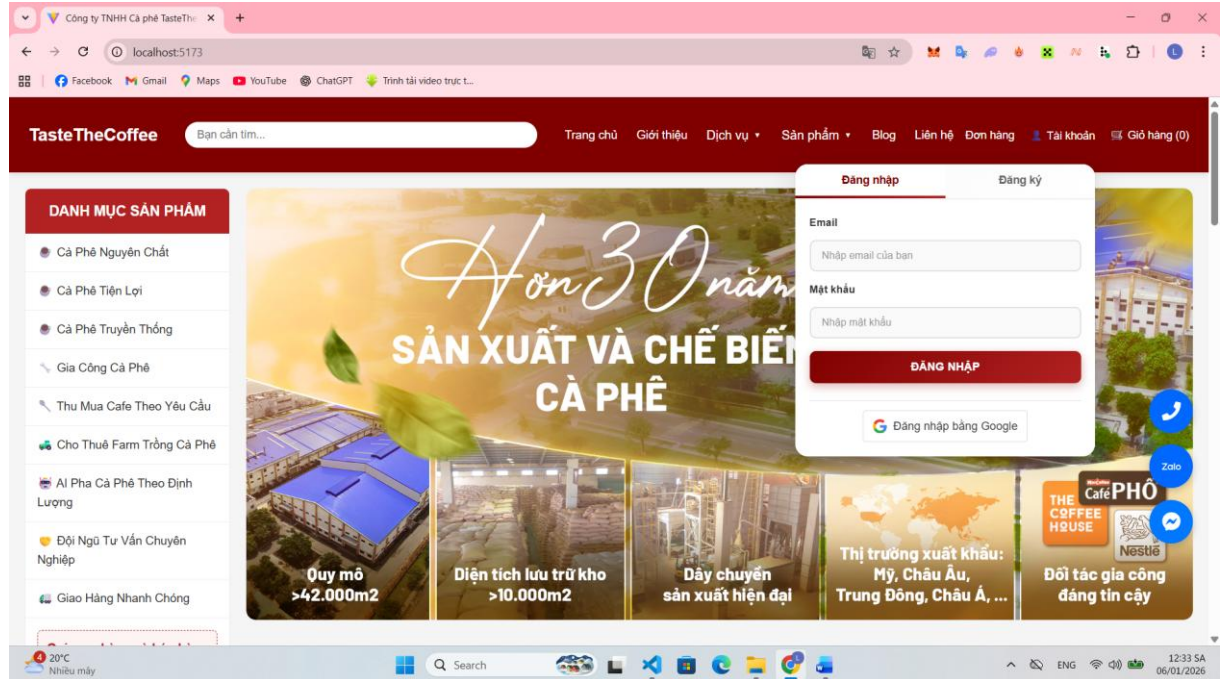
question	String	NO		Câu hỏi của người dùng, bắt buộc, trim
answer	String	NO		Câu trả lời, bắt buộc, trim
suggestions	Array of Object	YES		Danh sách gợi ý sản phẩm, mỗi suggestion gồm các trường: id, name, price, image, category, productCode
suggestions.id	String	NO		ID gợi ý sản phẩm, trim
suggestions.name	String	NO		Tên gợi ý sản phẩm, trim
suggestions.price	Number	YES		Giá gợi ý sản phẩm, mặc định 0
suggestions.image	String	YES		URL ảnh gợi ý sản phẩm, mặc định ""
suggestions.category	String	YES		Loại sản phẩm, mặc định ""
suggestions.productCode	String	YES		Mã sản phẩm, mặc định ""
createdAt	Date	NO		Thời gian tạo lịch sử chat, tự động từ timestamps
updatedAt	Date	NO		Thời gian cập nhật lịch sử chat, tự động từ timestamps

Bảng 3.30 Bảng mô tả REVIEW

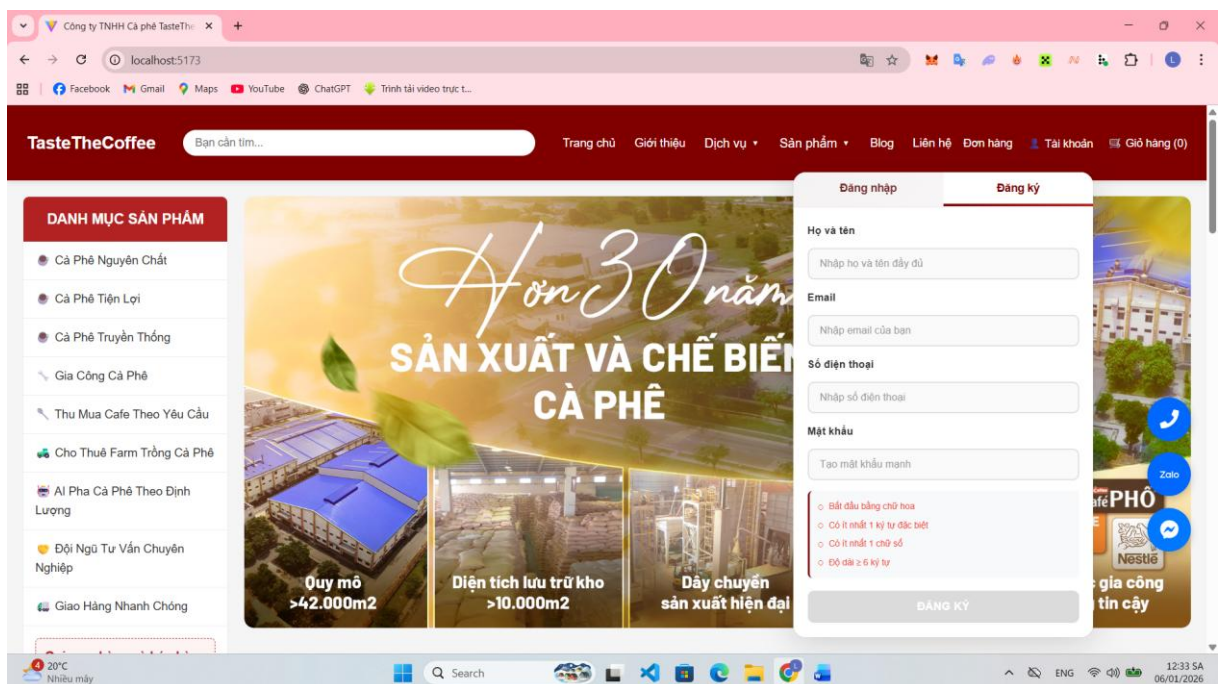
Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
userId	String	NOT NULL	FK	ID người dùng, liên kết với bảng <i>User</i>
productId	ObjectId	NOT NULL	FK	ID sản phẩm, liên kết với bảng <i>Product</i>
orderId	ObjectId	NOT NULL	FK	ID đơn hàng, đảm bảo chỉ đánh giá sau khi mua
rating	Number	NOT NULL		Điểm đánh giá (1–5)
comment	String	Có		Nội dung bình luận (có thể để trống)
userName	String	NOT NULL		Tên người dùng hiển thị khi đánh giá
userEmail	String	Có		Email người dùng (không bắt buộc)
createdAt	Date	NOT NULL		Tự động tạo lúc thêm review
updatedAt	Date	NOT NULL		Tự động cập nhật khi review thay đổi

CHƯƠNG 4 : CÀI ĐẶT DEMO GIAO DIỆN

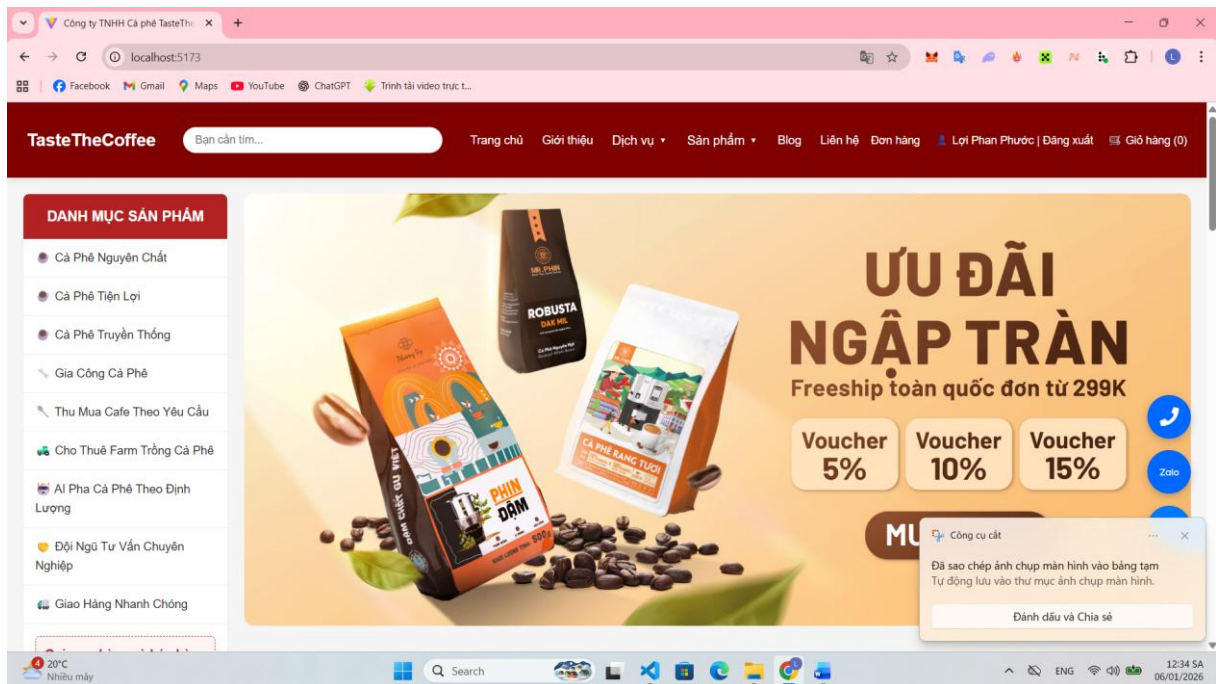
4.1 Giao diện người dùng



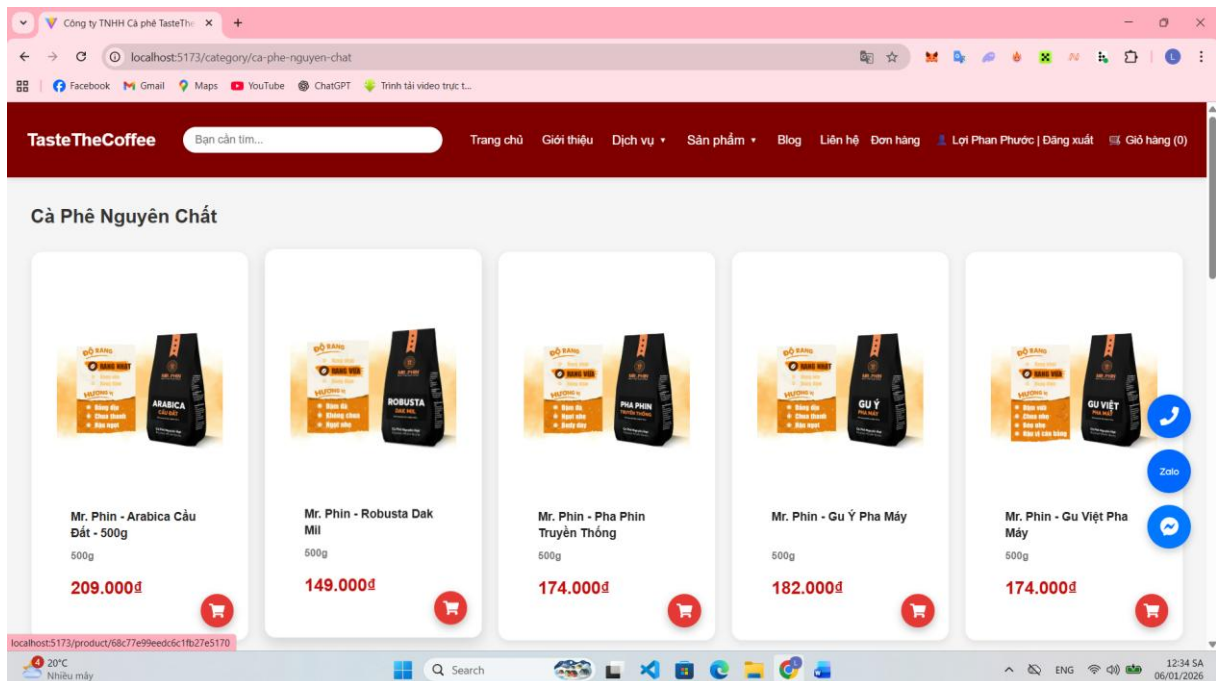
Hình 4.1 Giao diện đăng nhập



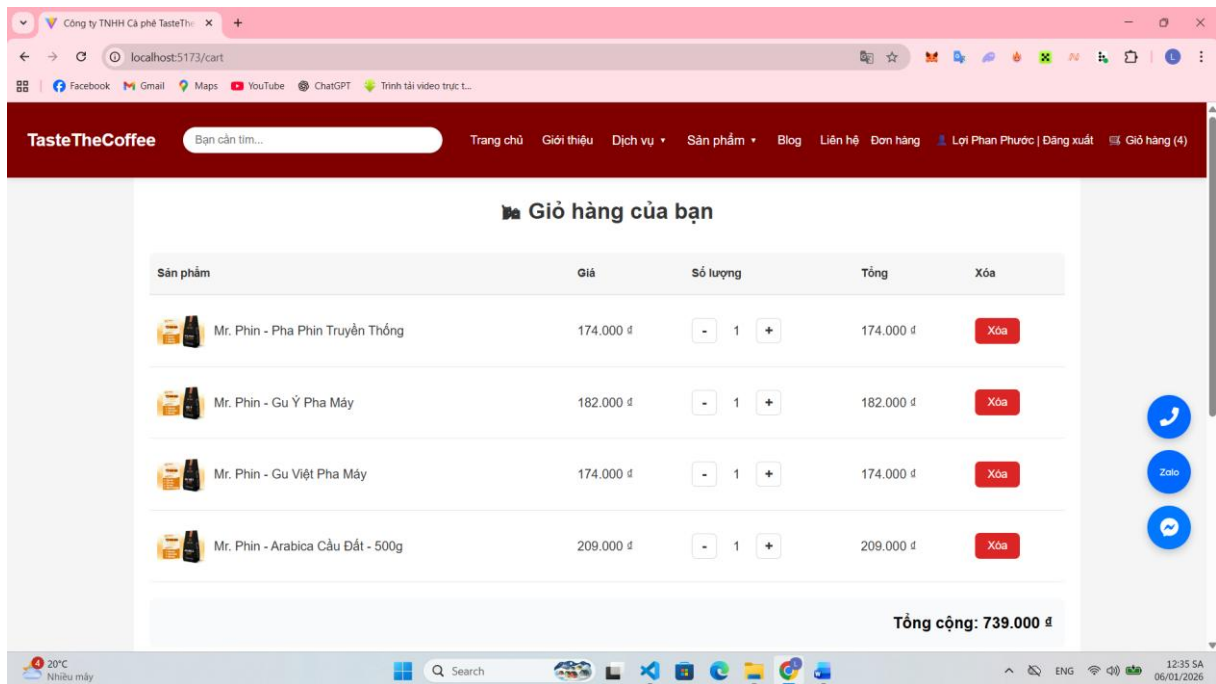
Hình 4.2 Giao diện đăng kí



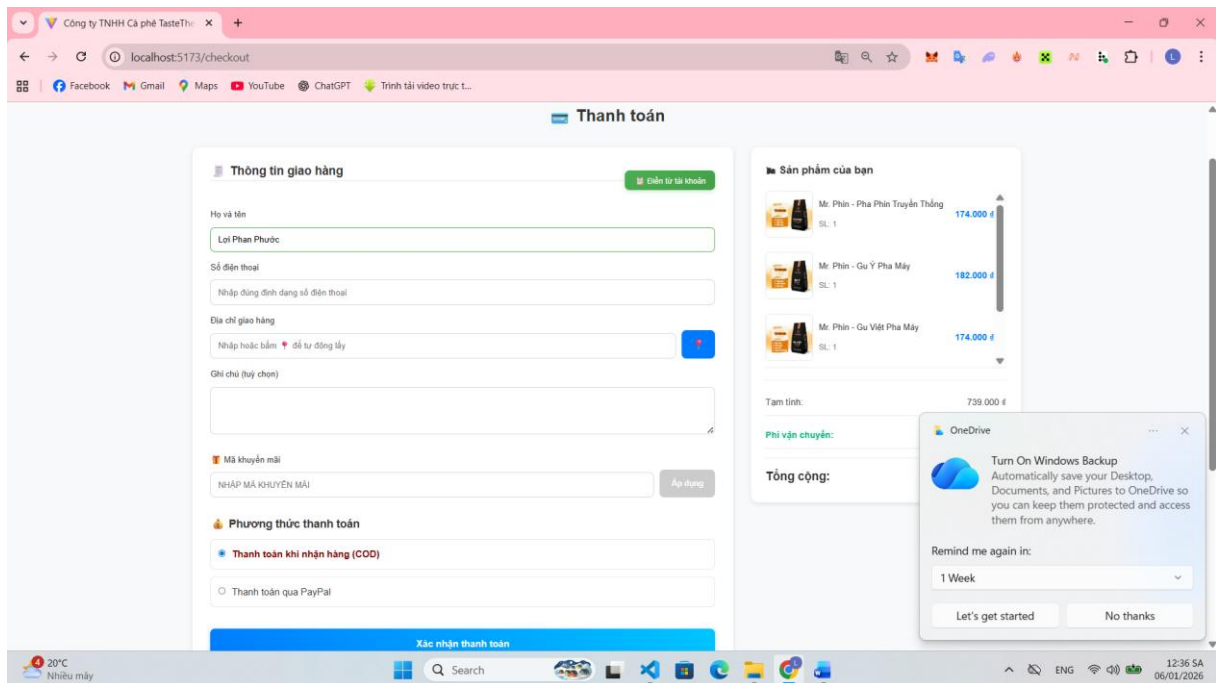
Hình 4.3 Màn hình trang chủ



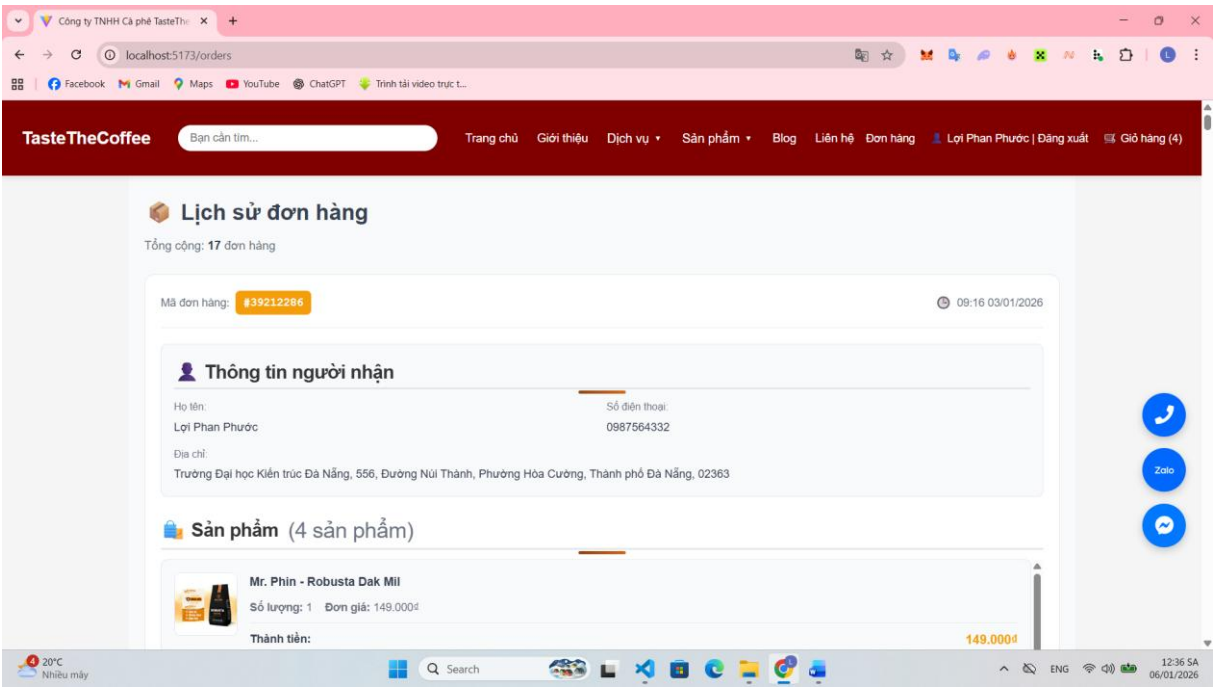
Hình 4.4 Giao diện sản phẩm



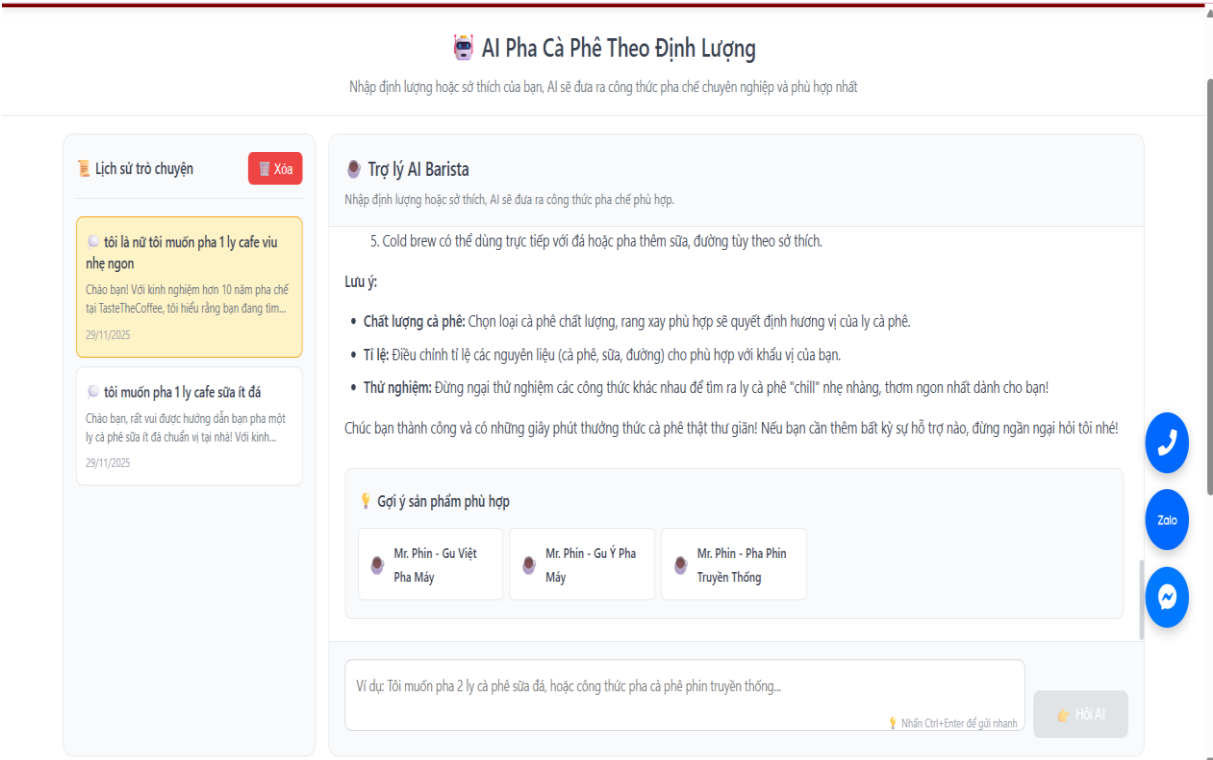
Hình 4.5 Giao diện giỏ hàng



Hình 4.6 Giao diện thanh toán

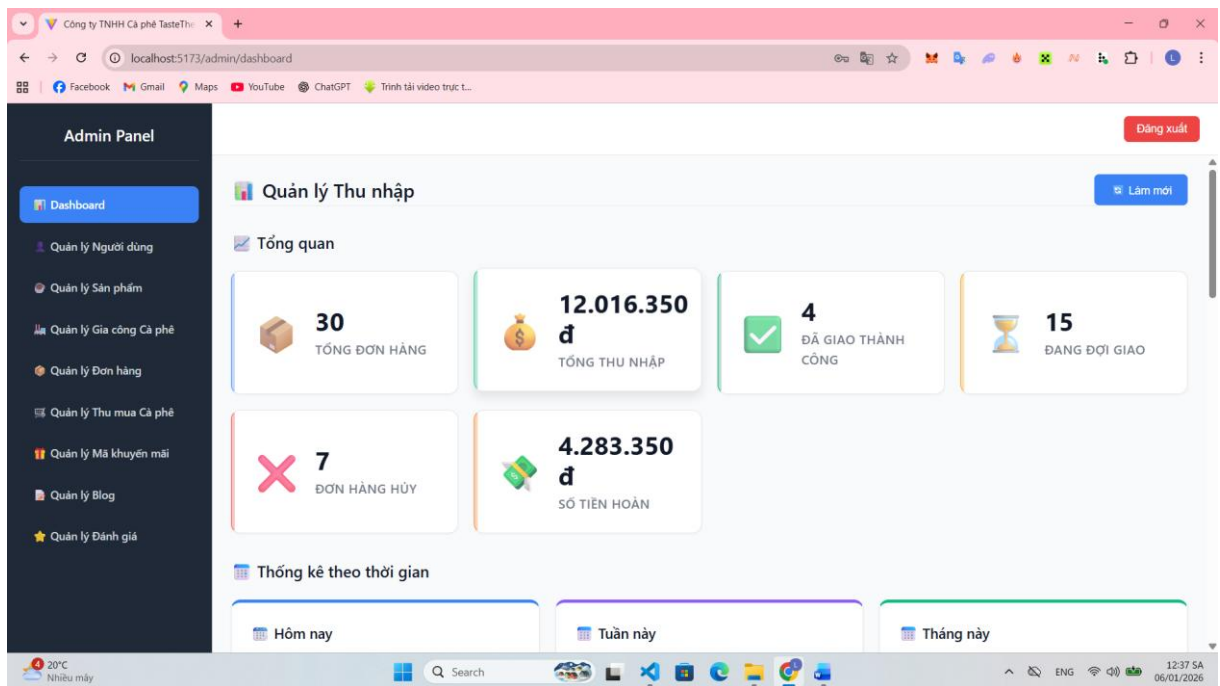


Hình 4.7 Giao diện lịch sử đơn hàng

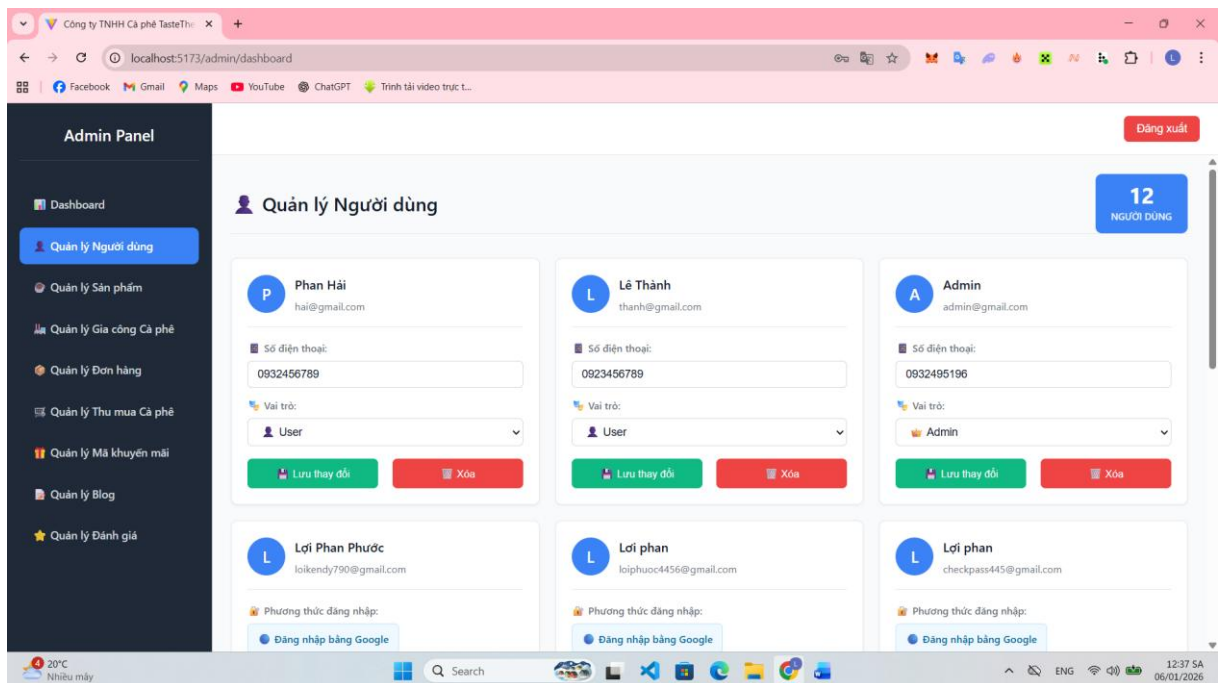


Hình 4.8 Giao diện chat bot AI

4.2. Giao diện admin



Hình 4.9 Giao diện màn hình chính trang admin



Hình 4.10 Giao diện quản lý người dùng trang admin

The screenshot shows the 'Thêm sản phẩm mới' (Add new product) form in the admin panel. The form is titled '+ Thêm sản phẩm mới' and includes a subtitle 'Điền thông tin sản phẩm để thêm vào hệ thống'. The form fields are:

- Mã sản phẩm * (Required): VD: CF001
- Tên sản phẩm * (Required): Tên sản phẩm
- Thương hiệu * (Required): Thương hiệu
- Giá (đ) * (Required): 0
- Trọng lượng: VD: 500g, 1kg
- Danh mục * (Required): -- Chọn danh mục --
- Mô tả: Mô tả chi tiết về sản phẩm...

The form is located in the 'Admin Panel' section, which also includes a sidebar with navigation links: Dashboard, Quản lý Người dùng, Quản lý Sản phẩm (highlighted), Quản lý Gia công Cà phê, Quản lý Đơn hàng, Quản lý Thu mua Cà phê, Quản lý Mã khuyến mãi, Quản lý Blog, and Quản lý Đánh giá. A 'Đăng xuất' (Logout) button is visible in the top right corner.

Hình 4.11 Giao diện thêm sản phẩm

The screenshot shows the 'Thêm gia công cà phê' (Add coffee processing) form and the 'Danh sách gia công' (Processing list) table in the admin panel. The form is titled '+ Thêm gia công cà phê' and includes a subtitle 'Thêm phương thức gia công mới vào hệ thống'. The form fields are:

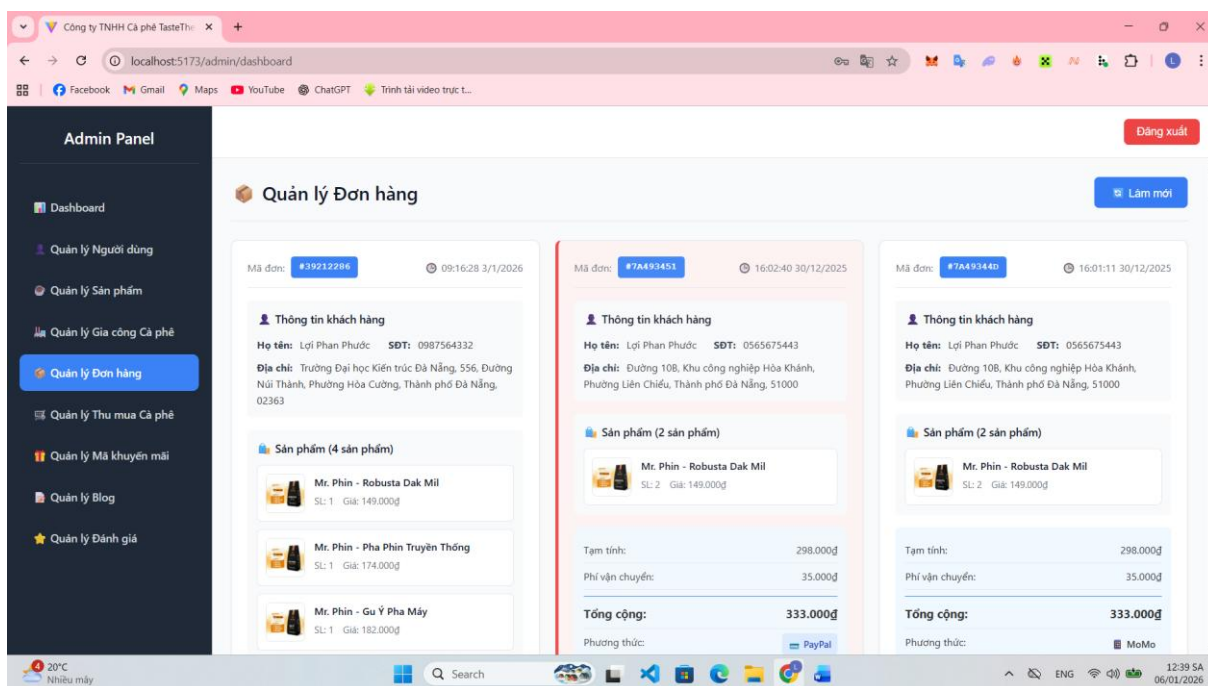
- Tên gia công * (Required): VD: Rang xay, Pha chế...
- Giá (VND/kg) * (Required): 0
- Ghi chú: Mô tả chi tiết về phương thức gia công...

The form is located in the 'Admin Panel' section, which also includes a sidebar with navigation links: Dashboard, Quản lý Người dùng, Quản lý Sản phẩm, Quản lý Gia công Cà phê (highlighted), Quản lý Đơn hàng, Quản lý Thu mua Cà phê, Quản lý Mã khuyến mãi, Quản lý Blog, and Quản lý Đánh giá. A 'Đăng xuất' (Logout) button is visible in the top right corner.

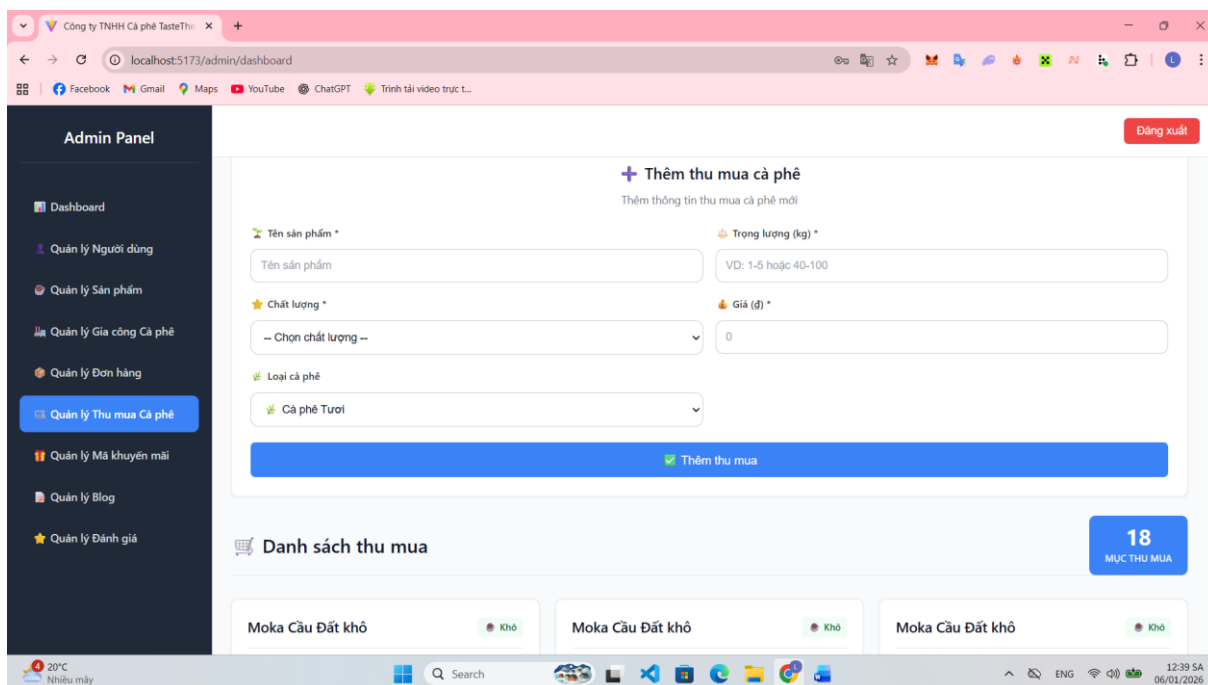
Below the form, the 'Danh sách gia công' (Processing list) table is displayed, showing 7 processing methods. The table has columns for the processing method and a status column. The processing methods listed are:

- Rang cà phê nhân
- Rang thành phẩm
- Trộn hạt hoặc xay bột

Hình 4.12 Giao diện quản lý gia công coffee



Hình 4.13 Giao diện quản lý đơn hàng



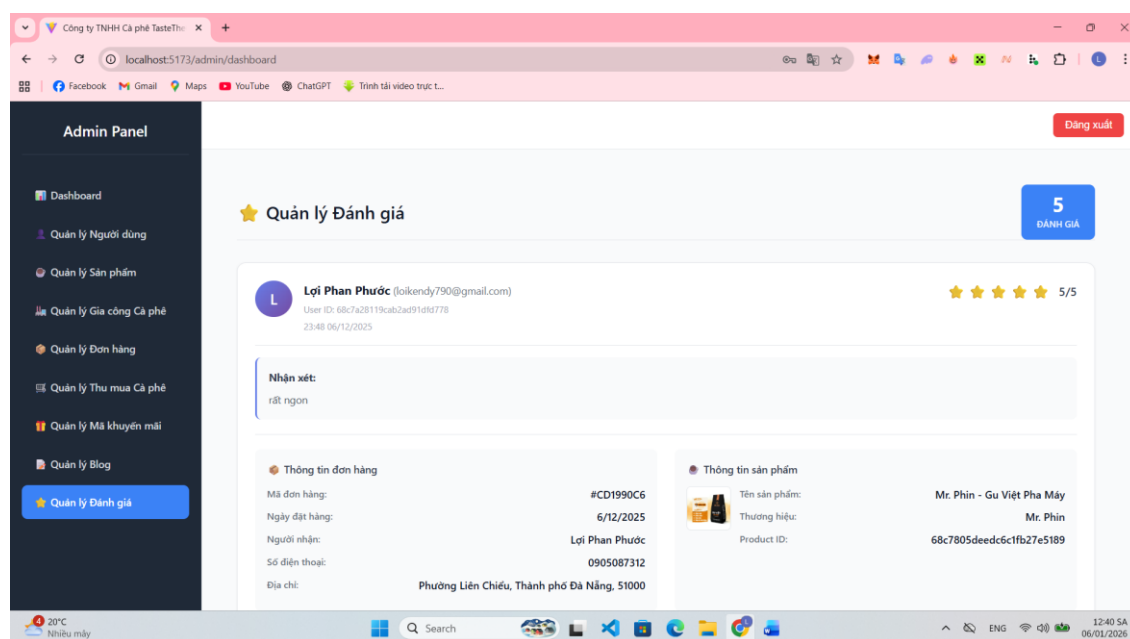
Hình 4.14 Giao diện quản lý thu mua coffee

The screenshot displays the 'Admin Panel' interface for 'Công ty TNHH Cà phê TasteThe'. The left sidebar contains navigation links: Dashboard, Quản lý Người dùng, Quản lý Sản phẩm, Quản lý Gia công Cà phê, Quản lý Đơn hàng, Quản lý Thu mua Cà phê, **Quản lý Mã khuyến mãi** (highlighted), Quản lý Blog, and Quản lý Đánh giá. The main content area is titled '+ Thêm mã khuyến mãi mới' with the subtitle 'Tạo mã khuyến mãi để khách hàng sử dụng khi thanh toán'. The form includes fields for: Mã khuyến mãi * (VD: WELCOME10), Mô tả (Mô tả mã khuyến mãi), Loại giảm giá * (Phân trăm (%)), Giá trị giảm * (10), Đơn hàng tối thiểu (đ) (0), Giảm tối đa (đ) - chỉ cho % (Để trống = không giới hạn), Ngày bắt đầu (dd/mm/yyyy), Ngày kết thúc * (dd/mm/yyyy), Số lần sử dụng tối đa (Để trống = không giới hạn), and Trạng thái (Kích hoạt). A blue button at the bottom says 'Thêm mã khuyến mãi'.

Hình 4.15 Giao diện quản lý mã khuyến mãi

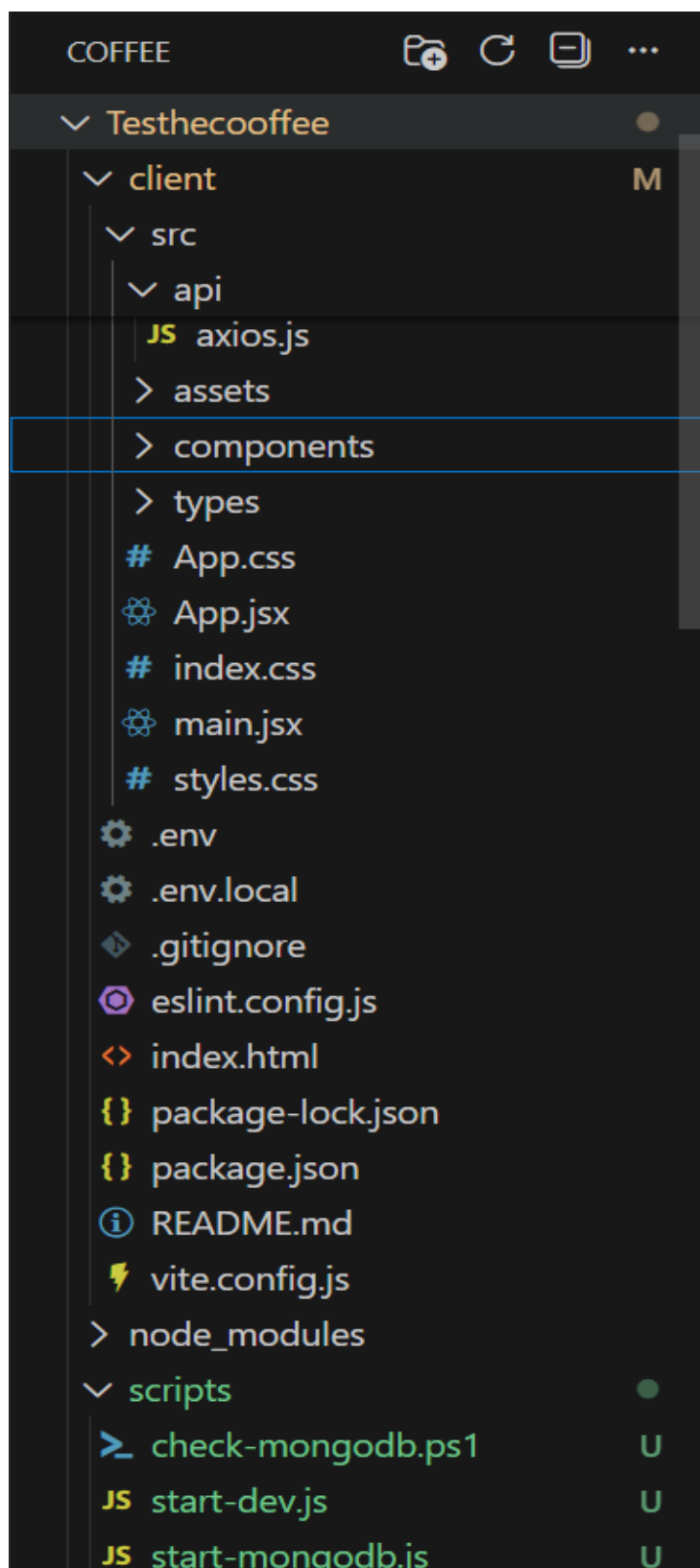
The screenshot displays the 'Admin Panel' interface for 'Công ty TNHH Cà phê TasteThe'. The left sidebar is identical to the previous image, with 'Quản lý Blog' highlighted. The main content area is titled '+ Thêm blog mới' with the subtitle 'Tạo bài viết blog để chia sẻ với khách hàng'. The form includes fields for: Tiêu đề * (VD: Cách pha cà phê ngon nhất), URL ảnh (https://example.com/image.jpg), Tóm tắt (excerpt) (Mô tả ngắn gọn về blog (tối đa 300 ký tự)), and Nội dung * (Nội dung bài viết...). A blue button at the bottom says 'Thêm blog mới'.

Hình 4.16 Giao diện quản lý Blog

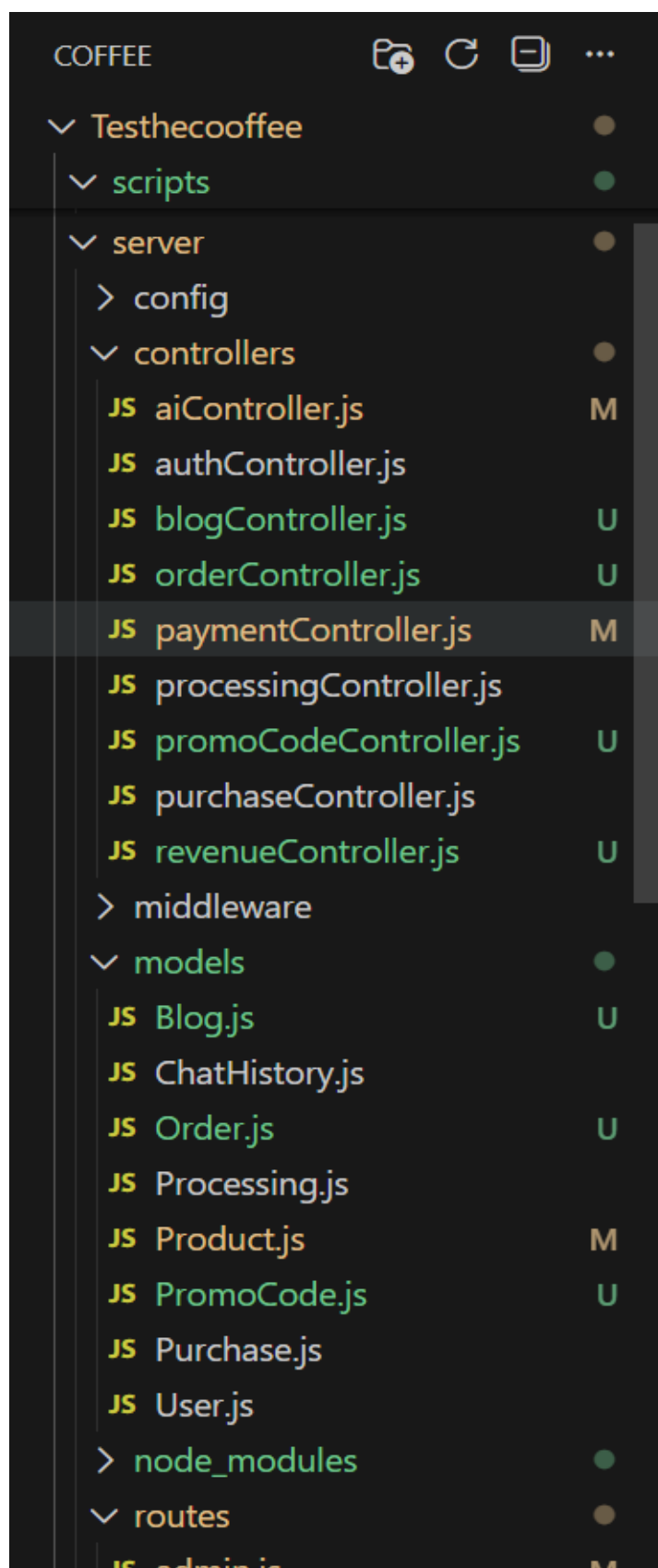


Hình 4.17 Giao diện quản lý đánh giá

4.3. Cấu trúc thư mục code



Hình 4.18 Cấu trúc thư mục frontend



Hình 4.19 Cấu trúc thư mục backend

KẾT LUẬN

Trong quá trình nghiên cứu và xây dựng hệ thống, em đã hoàn thành đề tài "Xây dựng Website mua bán Coffee cho công ty TasteTheCoffee" theo đúng yêu cầu đề ra. Hệ thống sử dụng React + Vite cho frontend, Node.js/Express cho backend và MongoDB để quản lý dữ liệu, giúp website phản hồi nhanh, giao diện trực quan và tương thích đa thiết bị. Hệ thống cho phép khách hàng tra cứu sản phẩm, khám phá blog, đặt hàng, thanh toán trực tuyến qua PayPal/MoMo và tương tác với AI Chatbot (Google Gemini) để nhận tư vấn công thức pha chế cá nhân hóa. Admin có thể quản lý sản phẩm, đơn hàng, người dùng, blog, mã khuyến mãi, dịch vụ gia công và thu mua thông qua dashboard tập trung, cùng hệ thống thống kê doanh thu với biểu đồ trực quan. Hệ thống giúp người thực hiện nắm vững quy trình phát triển ứng dụng web theo mô hình frontend – backend tách biệt, xây dựng API RESTful, thiết kế cơ sở dữ liệu linh hoạt và áp dụng các kiến thức lập trình vào thực tiễn.

Kết quả đạt được :

Hệ thống Website mua bán Coffee cho công ty TasteTheCoffee đã được xây dựng thành công, vận hành mượt mà và đáp ứng đầy đủ các yêu cầu đề ra. Về trải nghiệm khách hàng, frontend cung cấp router cho trang chủ, danh mục sản phẩm, chi tiết sản phẩm, blog, AI Chatbot, checkout và lịch sử đơn hàng, giúp người dùng tra cứu thông tin, nhận gợi ý từ AI, áp dụng mã khuyến mãi và thanh toán trực tuyến. Về quản trị, dashboard tập trung quản lý user, sản phẩm, đơn hàng, mã khuyến mãi, blog, dịch vụ gia công và thu mua, đồng thời cung cấp thống kê doanh thu tổng quan, theo tháng và biểu đồ cột bằng Recharts. Về công nghệ, hệ thống sử dụng React + Vite cho frontend, Node.js/Express cho backend và MongoDB quản lý dữ liệu, tích hợp Google Gemini AI, cổng thanh toán PayPal/MoMo, Firebase Authentication và các thư viện hỗ trợ như React Router, Axios, DOMPurify, EmailJS, Recharts, giúp hệ thống mở rộng dễ dàng và tối ưu trải nghiệm người dùng. Qua quá trình thực hiện, đồ án giúp người thực hiện hiểu rõ và vận dụng thành thạo các kiến thức về lập trình web, thiết kế API, quản lý cơ sở dữ liệu và triển khai hệ thống thực tế.

Hạn chế :

Mặc dù đạt được nhiều kết quả tích cực, hệ thống vẫn còn một số hạn chế cần khắc phục. Về bảo mật, một số bí mật nhạy cảm như khóa MoMo và JWT secret vẫn được hard-code với fallback values, phân quyền backend còn lỏng, client lưu token plain trong localStorage, chưa triển khai đầy đủ các biện pháp chống XSS, CSRF, rate

limiting, thiếu security headers và audit log, điều này tiềm ẩn rủi ro bảo mật. Về chức năng, chưa có audit log, chưa triển khai caching câu trả lời AI, đồng bộ cart đa thiết bị, kiểm thử tự động, hệ thống gửi email tự động, thông báo real-time, quản lý kho tự động và tích hợp vận chuyển. Về trải nghiệm, khi xử lý đồng thời nhiều người dùng có thể xảy ra độ trễ, giao diện trên mobile chưa tối ưu, thông báo tự động qua email/SMS chưa có, chưa có dark mode, tìm kiếm nâng cao, wishlist và so sánh sản phẩm. Về hiệu năng, chưa có caching với Redis, CDN cho static assets, image optimization và database indexing tối ưu. Về môi trường, chưa triển khai production, CI/CD pipeline, automated testing và backup.

Hướng phát triển :

Trong tương lai, hệ thống có thể được nâng cấp để tăng cường hiệu suất, bảo mật và mở rộng tính năng. Cụ thể, bảo mật có thể được củng cố bằng cách tách toàn bộ bí mật ra .env, triển khai xác thực hai lớp (2FA), RBAC chi tiết, rate limiting, sanitization, audit log bảo mật, security headers và CSRF protection. Thanh toán và thông báo có thể hoàn thiện callback MoMo/PayPal, lưu log giao dịch, gửi thông báo qua email/SMS, tích hợp thêm các cổng thanh toán nội địa khác. Trải nghiệm người dùng có thể nâng cao bằng đồng bộ hóa cart trên nhiều thiết bị, tối ưu UI/UX mobile, SEO/SSR cho blog, dark mode, tìm kiếm nâng cao, wishlist và so sánh sản phẩm. AI và dữ liệu có thể được cải thiện thông qua caching câu trả lời, fine-tune prompt, gợi ý sản phẩm tự động và tích hợp vector search. Về chất lượng và DevOps, có thể viết unit/integration test, thiết lập CI/CD pipeline, Docker Compose, giám sát realtime, backup tự động, CDN, image optimization và caching với Redis. Về mở rộng, có thể phát triển ứng dụng mobile, đa ngôn ngữ, đa tiền tệ và tích hợp mạng xã hội. Các cải tiến này sẽ giúp hệ thống TasteTheCoffee trở thành một nền tảng thương mại điện tử toàn diện, an toàn, thân thiện với người dùng và dễ mở rộng trong tương lai.

TÀI LIỆU THAM KHẢO

1. "ExpressJS" [Online]. Available: <https://expressjs.com/>.
2. "MongoDB" [Online]. Available: <https://www.mongodb.com>.
3. "NodeJS" [Online]. Available: <https://nodejs.org/en/docs/>.
4. Tài liệu JavaScript: <https://javascript.info/37>
5. Tài liệu React: <https://reactjs.org/docs/getting-started.html>