

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH



Nguyễn Hữu Độ

WEBSITE MUA BÁN SẢN PHẨM NHẬP KHẨU

ĐỒ ÁN NGÀNH
NGÀNH Công nghệ thông tin

TP. HỒ CHÍ MINH, NĂM 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH



Nguyễn Hữu Độ

Website mua bán sản phẩm nhập khẩu

Mã số sinh viên: 2151050099

ĐỀ ÁN NGÀNH
NGÀNH Công nghệ thông tin

Giảng viên hướng dẫn: ThS. Nguyễn Văn Bảy

TP. HỒ CHÍ MINH, NĂM 2024

LỜI CẢM ƠN

Lời cảm ơn, em xin chân thành cảm ơn Thầy/Cô khoa Công nghệ thông tin đã giảng dạy, trang bị cho em những kiến thức thiết thực và cần thiết, trong suốt quá trình học tập tại trường Đại học Mở TP HCM. Nhờ sự giảng dạy và chỉ bảo của Thầy/Cô mà em đã trang bị cho bản thân những kiến thức về chuyên ngành công nghệ thông tin để có thể tiếp tục con đường học tập, áp dụng kiến thức đã học và phát triển những sản phẩm có giá trị, phục vụ cho nhiều lĩnh vực khác nhau.

Em xin gửi lời cảm ơn đến giảng viên hướng dẫn Nguyễn Văn Bay, đã tận tình giúp đỡ em từ những bước đầu tiên khi xây dựng ý tưởng nghiên cứu, đề cương cũng như trong suốt quá trình thực hiện và hoàn thiện Đồ án tốt nghiệp này.

Trong suốt quá trình thực hiện đồ án, mặc dù em đã rất lỗ lực và cố gắng để hoàn thành. Nhưng trong khả năng còn hạn chế, chắc chắn Đồ án sẽ không tránh khỏi những thiếu sót. Em hy vọng nhận được sự góp ý, thông cảm và tận tình chỉ bảo của Thầy/Cô để hoàn thiện báo cáo một cách tốt nhất.

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

TÓM TẮT ĐỒ ÁN NGÀNH

“Nông sản việt” là một website mua bán sản phẩm ngoại nhập dành cho các nhà bán lẻ trực tuyến, với mục tiêu chính của website là thiết kế tạo ra sự tương tác dễ dàng và thân thiện đối với người dùng có nhu cầu mua sắm các loại sản phẩm nhập khẩu. Website có giao diện được tối ưu cho trải nghiệm người dùng, bố cục đơn giản và hiện đại giúp cho việc tìm kiếm, xem và lựa chọn sản phẩm của người dùng trở nên thuận tiện hơn.

Hệ thống của website tích hợp các công cụ tìm kiếm, lọc sản phẩm phù hợp với nhu cầu cá nhân một cách nhanh chóng. Mỗi sản phẩm được trình bày với thông tin chi tiết rõ ràng, bao gồm các thông tin về hình ảnh, mô tả, giá bán, giá khuyến mãi mà người bán cung cấp. Ngoài ra, người dùng có thể xem và đánh giá sản phẩm của những người mua trước hoặc tự viết đánh giá riêng.

Khách hàng trước khi mua sắm, đánh giá sản phẩm trên website trước tiên phải tiến hành đăng ký và đăng nhập tài khoản. Hệ thống quản lý tài khoản người dùng được thiết kế với độ bảo mật cao, đảm bảo an toàn cho thông tin cá nhân của khách hàng. Bên cạnh đó, website phân chia rõ ràng giữa quyền hạn của người dùng và admin, tạo sự linh hoạt trong việc quản lý và vận hành hệ thống.

Website hỗ trợ chức năng giỏ hàng tiện dụng, khách hàng thêm sản phẩm vào giỏ hàng chỉ với thao tác đơn giản. Sau khi thêm sản phẩm vào giỏ hàng khách hàng sẽ được sử dụng các tính năng thanh toán được cập nhật theo xu hướng mới nhất, hỗ trợ nhiều phương thức thanh toán đa dạng như ví điện tử, thẻ tín dụng hoặc thanh toán khi nhận hàng. Khách hàng cũng có thể theo dõi tình trạng đơn hàng từ lúc thanh toán đến nhận hàng thành công.

MAJOR PROJECT SUMMARY

“Nong san Viet” is a website for buying and selling imported foods for online retailers, with the main goal of the website being designed to create easy and friendly interactions for users who need to buy imported foods. The website has an interface optimized for user experience, a simple and modern layout that makes it more convenient for users to search, view and select products

The website system integrates search tools, filters products to quickly suit individual needs. Each product is presented with clear detailed information, including information about images, descriptions, prices, and promotional prices provided by the seller. In addition, users can view and rate products from previous buyers or write their own reviews.

Before shopping and rating products on the website, customers must first register and log in to their account. The user account management system is designed with high security, ensuring the safety of customers' personal information. In addition, the website clearly divides the authority of users and admins, creating flexibility in managing and operating the system.

The website supports a convenient shopping cart function, customers can add products to the cart with just a simple operation. After adding products to the cart, customers will be able to use payment features updated according to the latest trends, supporting a variety of payment methods such as e-wallets, credit cards or cash on delivery. Customers can also track the status of orders from payment to successful receipt.

MỤC LỤC

LỜI CẢM ƠN	i
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN.....	ii
TÓM TẮT ĐỒ ÁN NGÀNH.....	iii
MAJOR PROJECT SUMMARY	iv
MỤC LỤC	v
DANH MỤC TỪ VIẾT TẮT	viii
DANH MỤC HÌNH ẢNH	ix
DANH MỤC BẢNG	xii
CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI	1
1.1. Giới thiệu.....	1
1.2. Mục tiêu đề tài	2
1.3. Phạm vi và đối tượng nghiên cứu	3
1.3.1 Phạm vi nghiên cứu	3
1.3.2 Đối tượng nghiên cứu	3
1.3.3 Phương pháp nghiên cứu	3
1.3.4 Bố cục đề tài	4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	5
2.1. Tổng quan công nghệ sử dụng.....	5
2.1.1. Khái quát về Technical Stack	5
2.1.2. Khái quát MERN Stack	5
2.2. Tổng quan về ReactJS.....	5
2.2.1. Giới thiệu về ReactJS	5
2.2.2. Chức năng React.....	6
2.2.3. Sơ đồ hoạt động của React	7

2.2.4. JSX	7
2.2.5. Props và State	7
2.2.6. Tổng quan về Node.js.....	9
2.2.7. Tổng quát về MongoDB.....	19
2.2.8. Tổng quan về ExpressJS.....	24
2.2.9. Tìm hiểu Router trong Express	26
2.2.10. Giới thiệu về RESTful API	27
2.2.11. Giới thiệu về Postman	29
CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG	31
3.1. Giới thiệu hệ thống	31
3.2. Các chức năng của hệ thống	31
3.3. Sơ đồ use-case	33
3.3.1. Danh sách Actor.	33
3.3.2. Sơ đồ Use case.....	34
3.4. Sơ đồ lớp	46
3.4.1. Users (Người dùng)	46
3.4.2. Order (đơn hàng)	47
3.4.3. Products (sản phẩm)	47
3.4.4. Listtypeproducts (danh mục loại sản phẩm)	48
3.5. Biểu đồ tuần tự của các chức năng trong hệ thống.....	49
3.5.1. Biểu đồ tuần tự đăng nhập.....	49
3.5.2. Biểu đồ tuần tự đăng ký thành viên.....	50
3.5.3. Biểu đồ tuần tự tìm kiếm sản phẩm.....	51
3.5.4. Biểu đồ tuần tự đặt hàng.....	53
3.5.5. Biểu đồ tuần tự xem thông tin sản phẩm.....	54
3.5.6. Biểu đồ tuần tự bình luận (gửi câu hỏi).....	56

3.5.7. Biểu đồ tuần tự hiển thị danh sách sản phẩm	57
3.5.8. Biểu đồ tuần tự thêm sản phẩm	58
3.5.9. Biểu đồ tuần tự xóa sản phẩm	60
3.5.11. Biểu đồ tuần tự xóa thành viên.....	63
CHƯƠNG 4: GIAO DIỆN ÚNG DỤNG.....	64
 4.1. Giao diện ứng dụng	64
 4.2. Giao diện người dùng và khách hàng.....	64
4.2.1. Trang đăng ký	64
4.2.2. Trang đăng nhập	65
4.2.3. Giao diện trang chủ	66
4.2.4. Giao diện sản phẩm	67
4.2.5. Giao diện xem thông tin sản phẩm	67
4.2.6. Giao diện đánh giá của khách hàng.....	68
4.2.7. Giao diện sản phẩm liên quan	69
4.2.8. Giao diện hồ sơ người dùng	69
4.2.9. Giao diện lịch sử mua hàng.....	70
4.2.10. Giao diện giỏ hàng	71
4.2.11. Giao diện thanh toán.....	72
 4.3. Giao diện quản trị viên	82
CHƯƠNG 5: KIỂM TRA VÀ THỬ NGHIỆM.....	86
 5.1. Kiểm thử khả năng tương thích	86
CHƯƠNG 6: KẾT LUẬN.....	89
 6.1. Kết quả đạt được.	89
6.1.1. Ưu điểm	89
6.1.2. Hạn chế	89
 6.2. Phương hướng phát triển.....	89

6.2.1. Cải thiện tính năng hiện có	89
6.2.2. Phát triển tính năng mới.....	90
6.3. Kết luận.....	90
TÀI LIỆU THAM KHẢO	x

DANH MỤC TỪ VIẾT TẮT

TỪ VIẾT TẮT	VIẾT ĐÀY ĐỦ
CNTT	Công nghệ thông tin
NSD	Người sử dụng
NCC	Nhà cung cấp

DANH MỤC HÌNH ẢNH

Hình 2.1. Sơ đồ hoạt động của React	7
Hình 2.2. Sơ đồ hoạt động của Node.js	12
Hình 2.3. Truy cập nodejs.org	13
Hình 2.4. Mở file	13
Hình 2.5. Click vào I accept the terms in the License Agreement	14
Hình 2.6. Nhấn next.....	15
Hình 2.7. Cửa sổ hiện ra (Chọn theo gợi ý)	16
Hình 2.8. Install	17
Hình 2.9. Chờ hoàn tất.....	18
Hình 2.10. Kiểm tra thông qua cmd	19
Hình 2.11. Truy cập trang mongodb.com/try	20
Hình 2.12. Chọn phiên bản.....	21
Hình 2.13. Tải và cài đặt	21
Hình 2.14. Cửa sổ hiện lên, chọn next	22
Hình 2.15. Click I accept the tem in the License Argreement, nhấn next.....	22
Hình 2. 16. Chọn Complete	23
Hình 2. 17. Có thể thay đổi mục lưu trữ, sau đó chọn next.....	23
Hình 2.18. Hoàn tất	24
Hình 2.19. Cấu trúc của ExpressJS	25
Hình 2.20. Cách thức hoạt động của API	28
Hình 2.21. Cách hoạt động của RESTful API.....	28
Hình 2.22. Giới thiệu về Postman	29
 Hình 3.1. Sơ đồ Use case.....	34
Hình 3.2. Sơ đồ lớp.....	46
Hình 3.3. Biểu đồ tuần tự đăng nhập	49
Hình 3.4. Biểu đồ tuần tự đăng ký thành viên.....	50
Hình 3.5. đồ tuần tự tìm kiếm sản phẩm	51
Hình 3.6. Biểu đồ tuần tự đặt hàng.....	53
Hình 3.7. Biểu đồ tuần tự xem thông tin sản phẩm.....	55
Hình 3.8. Biểu đồ tuần tự bình luận (gửi câu hỏi).....	56

Hình 3.9. Biểu đồ tuần tự hiển thị danh sách sản phẩm	57
Hình 3.10. Biểu đồ tuần tự thêm sản phẩm	58
Hình 3.11. Biểu đồ tuần tự xóa sản phẩm	60
Hình 3.12. Biểu đồ tuần tự xem đơn hàng.....	61
Hình 3.13. Biểu đồ tuần tự xóa thành viên.....	63
Hình 3.26. Thanh toán bằng Paypal	74
Hình 4.1. Trang đăng ký	64
Hình 4.2. Trang đăng nhập	65
Hình 4.3. Giao diện trang chủ	66
Hình 4.4. Giao diện sản phẩm	67
Hình 4.5. Giao diện xem thông tin sản phẩm	68
Hình 4.6. Giao diện đánh giá của khách hàng.....	68
Hình 4.7. Sản phẩm liên quan	69
Hình 4.8. Giao diện hồ sơ người dùng	69
Hình 4.9. Giao diện lịch sử mua hàng	70
Hình 4.10. Giao hiện gio hàng.....	71
Hình 4.11. Giao diện thanh toán.....	72
Hình 4.12. Thanh toán khi nhận hàng	73
Hình 4.13. Thực hiện đăng nhập trước khi thanh toán.....	75
Hình 4.14. Thực hiện lựa chọn thành toán	76
Hình 4.15. Thanh toán bằng Vnpay	77
Hình 4.16. Thực hiện chọn ngân hàng thanh toán.....	78
Hình 4.17. Nhập tên ngân hàng thực hiện thanh toán	79
Hình 4.18. Nhập thông tin để tiến hành thanh toán.....	80
Hình 4.19. Nhập OTP tiến hành xác thực	81
Hình 4.20. Thanh toán thành công	81
Hình 4.21. Giao diện trang quản lý	82
Hình 4.22. Giao diện biểu đồ.....	82
Hình 4.23. Giao diện quản lý người dùng	83
Hình 4.24. Giao diện quản lý sản phẩm	83
Hình 4.25. Giao diện thêm sản phẩm	84

Hình 4.26. Giao diện quản lý sản phẩm	84
Hình 4.27. Giao diện quản lý đơn hàng.....	85
Hình 5.1. Trên thiết bị PC, laptop, ipad.....	86
Hình 5.2. Trên thiết bị di động	87

DANH MỤC BẢNG

Bảng 3.1. Danh sách actor	33
Bảng 3.2. Bảng chú thích sơ đồ Use case.....	34
Bảng 3.3. Bảng Users	46
Bảng 3.4. Bảng Order	47
Bảng 3.5. Bảng Products	48
Bảng 3.6. Bảng Listtypeproducts	48

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

Chương này sẽ giới thiệu về đề tài nghiên cứu, mục tiêu của đề tài cũng như phạm vi và đối tượng nghiên cứu của đề tài

1.1. Giới thiệu

Trong bối cảnh công nghệ đang phát triển, các thiết bị điện tử như máy tính và điện thoại di động không chỉ đơn thuần phục vụ cho nhu cầu giải trí mà còn trở thành công cụ đắc lực hỗ trợ giải quyết các công việc hàng ngày trong cuộc sống một cách nhanh chóng và tiện lợi. Cùng với sự phát triển của công nghệ, các dịch vụ trực tuyến không ngừng mở rộng, đa dạng hóa, đặc biệt là sự bùng nổ của website thương mại điện tử. Điều này tạo ra sự thúc đẩy mua sắm trực tuyến ngày càng cao, nhất là đối với các mặt hàng nhập khẩu, khi người tiêu dùng muốn tiếp cận các sản phẩm chất lượng từ khắp nơi trên thế giới một cách nhanh chóng, thuận tiện và chi phí hợp lý.

Hiện nay, một số ứng dụng và trang web mua bán các sản phẩm khẩu đã được triển khai rộng rãi trên thị trường như Shopee, Vinffood, AnNamMarket, những sàn thương mại điện tử này mang lại nhiều tiện ích cho người tiêu dùng như dễ dàng tìm kiếm, đặt hàng và thanh toán trực tuyến. Tuy nhiên, chúng vẫn còn tồn tại một số hạn chế như thiếu đa dạng nguồn cung cấp (các trang web lớn thường ưu tiên các sản phẩm phổ biến, khiến các sản phẩm nhập khẩu đặc trưng từ nhiều quốc gia khó có thể tìm thấy), giao diện người dùng chưa tối ưu (một số trang web lớn có giao diện phức tạp, khiến cho người dùng gặp khó khăn trong việc tìm kiếm và đặt hàng, đặc biệt với những người chưa quen với công nghệ).

Đáp ứng nhu cầu đó, em đã nghiên cứu và phát triển một website chuyên bán các mặt hàng sản phẩm nhập với tên gọi là “Nông sản Việt”. Nông sản Việt được xây dựng dựa trên MERN stack (MongoDB, Express, React, Nodejs) là bộ công nghệ hiện đại kết hợp sức mạnh của cơ sở dữ liệu NoSQL MongoDB, backend hiệu quả với Express, giao diện người dùng mượt mà nhờ React, và khả năng xử lý mạnh mẽ của Node.js trên sever.

Phần Frontend của Nông sản Việt sử dụng React để tạo giao diện thân thiện, tương tác cao, mang đến trải nghiệm mua sắm mượt mà và dễ sử dụng. Material-UI được tích hợp vào trong để phát triển các thành phần giao diện hiện đại, gọn gàng và thu hút người dùng. Về phần backend, Express kết hợp với Node.js giúp xử lý nhanh chóng các yêu cầu từ phía người dùng, đồng thời đảm bảo được cả tính bảo mật và mở rộng hệ thống. MongoDB, hệ quản trị dữ liệu NoSQL, được sử dụng để lưu trữ thông tin sản phẩm và dữ liệu khách hàng, mang lại hiệu quả cao trong việc quản lý dữ liệu lớn và phức tạp.

1.2. Mục tiêu đề tài

Hệ thống website mua bán thực phẩm nhập khẩu “Nông sản Việt” ra đời với mục tiêu mang đến sự tiện lợi, nhanh chóng và hiệu quả cho cả người tiêu dùng lẫn nhà cung cấp. Nông sản Việt không chỉ giúp người sử dụng dễ dàng tìm kiếm sản phẩm ngoại nhập mà còn tạo cơ hội cho các nhà cung cấp sản phẩm mở rộng thị trường, tiếp cận khách hàng một cách nhanh chóng, tiết kiệm thời gian và chi phí.

Nhà cung cấp có thể tiết kiệm rất nhiều thời gian trong việc quản lý và giới thiệu sản phẩm. Khi có sản phẩm mới, chỉ cần đăng tải thông tin về sản phẩm cùng với các chi tiết như nguồn gốc, giá cả, mô tả chi tiết lên Nông sản Việt. Người tiêu dùng sau đó có thể dễ dàng tìm kiếm, tham khảo và đánh giá sản phẩm xem có phù hợp với nhu cầu của mình hay không. Quá trình mua bán từ đó trở nên nhanh chóng và dễ dàng hơn. Các nhà cung cấp có thể dễ dàng tiếp cận để từng nhóm các khách hàng, dựa trên thông tin khách hàng đã được hệ thống phân loại, để tối ưu hóa quá trình bán hàng.

Người tiêu dùng dễ dàng tìm kiếm sản phẩm với vài thao tác đơn giản. Họ có thể sử dụng thanh tìm kiếm để nhập thông tin về sản phẩm mong muốn như loại thực phẩm, giá thành, hay những tiêu chí khác. Nông sản Việt sẽ đề xuất những sản phẩm phù hợp với nhu cầu của họ. Khách hàng có thể xem xét chi tiết sản phẩm, thêm vào giỏ hàng và tiến hành mua sắm một cách tiện lợi. Ngoài ra, người dùng cũng có thể tạo tài khoản, lưu trữ thông tin mua sắm để việc giao dịch trở nên nhanh chóng và dễ dàng hơn.

1.3. Phạm vi và đối tượng nghiên cứu

1.3.1 Phạm vi nghiên cứu

Phạm vi của đồ án là thiết kế và tìm ra giải pháp hiệu quả, để giải quyết công việc bán hàng cho các cửa hàng sản phẩm. Các mục tiêu cụ thể của đồ án bao gồm:

- Thiết kế một website hợp lý, hiệu quả trong việc mua hàng của khách hàng và việc quản lý các thông tin sản phẩm.
- Ứng dụng MERN Stack (MongoDB, Express, React, Node.js) để xây dựng website mua bán sản phẩm ngoại nhập. MERN Stack kết hợp các công nghệ hiện đại, bao gồm MongoDB để lưu trữ dữ liệu, Express là backend framework, React làm frontend framework, và Node.js để chạy mã JavaScript phía server.

Giới hạn phạm vi nghiên cứu chủ yếu ở các chi nhánh cửa hàng, Blog và các thông tin cửa hàng quảng bá.

1.3.2 Đối tượng nghiên cứu

Đối tượng nghiên cứu được nhắm đến bao gồm: Các khách hàng của cửa hàng sản phẩm, vai trò của người quản lý cửa hàng, các loại sản phẩm trên thị trường và nghiên cứu về sử dụng công nghệ MERN STACK.

Các từ khóa được đề cập nghiên cứu bao gồm:

- ReactJS: Tìm hiểu khái quát và nghiên cứu về cách sử dụng Component để render HTML, hiểu được vòng đời của Component, prop của component,...
- NodeJS: Tham khảo, nghiên cứu về câu lệnh, cú pháp của JavaScript nói chung và của NodeJS nói riêng, cách sử dụng npm để install các package cần thiết cho ứng dụng web.
- ExpressJS: Học cách sử dụng các tính năng như Routing, middlewares, ... để xây dựng các API và backend.
- MongoDB: Học cách làm việc với Collection, thực thi câu lệnh query dùng để truy vấn dữ liệu.

1.3.3 Phương pháp nghiên cứu

Phương pháp tài liệu: Nghiên cứu các tài liệu liên quan đến đề tài nghiên cứu.

Phương pháp thực nghiệm: Nghiên cứu, xây dựng các phương pháp đặt hàng và quản lý sản phẩm hiệu của cho cửa hàng. Phát triển phần mềm quản lý bán sản phẩm cho cửa hàng và đánh giá kết quả.

1.3.4 Bố cục đề tài

Chương 1: Giới thiệu đề tài.

Giới thiệu về đề tài và đối tượng nghiên cứu, phạm vi, đối tượng và phương pháp nghiên cứu đề tài.

Chương 2: Cơ sở lý thuyết.

Mô tả các lý thuyết sử dụng để nghiên cứu, các ngôn ngữ lập trình, framework và các thư viện được sử dụng.

Chương 3: Phân tích hệ thống.

Đưa ra các mô hình hệ thống, đề xuất các chức năng cho hệ thống. Trình bày các mô hình thực thể, quan hệ. Phân tích chi tiết các chức năng của người dùng và người quản lý.

Chương 4: Giao diện ứng dụng.

Trình bày giao diện người dùng, giao diện người quản lý.

Chương 5: Kiểm tra và thử nghiệm

Kiểm thử trên các giao diện thiết bị và trình bày kết quả thực hiện.

Chương 6: Kết luận

Kết luận đề tài, nêu những vấn đề và phát triển đề tài.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Chương này mô tả cơ sở lý thuyết được sử dụng để nghiên cứu, các ngôn ngữ lập trình, framework và các thư viện được sử dụng để phát triển đề tài.

2.1. Tổng quan công nghệ sử dụng

2.1.1. Khái quát về Technical Stack

Technical Stack là một tập hợp các phần mềm/công nghệ, framework, ... tạo nên một phần mềm hoàn chỉnh. Hiện nay với các phần mềm thường có 2 phần là : Client và Server (Còn được gọi là Frontend và Backend). Nên được phân ra Frontend stack và Backend stack.

Đối với tên, ta dùng chữ cái đầu để đặt tên cho Technical Stack như: MERN (MongoDB, Express, ReacJS, NodeJS).

2.1.2. Khái quát MERN Stack

MERN Stack là tập hợp các công nghệ dùng để phát triển web, được sử dụng để xây dựng các ứng web full-stack. Các công nghệ này đều dựa trên JavaScript, giúp cho nhà phát triển ứng dụng web nhanh và dễ dàng hơn.

Khái niệm về MonggoDB, đây là cơ sở dữ liệu NoSQL, nhằm lưu trữ các dữ liệu dưới dạng document (tài liệu), sử dụng định dạng BSON (dạng mở rộng của JSON). MongoDB phù hợp với các dạng ứng dụng yêu cầu cơ sở dữ liệu linh hoạt, dễ mở rộng.

Khái niệm về Rect.js, React (hay Reacjs, Reac.js) một thư viện JavaScript mã nguồn mở do Facebook (Meta) phát triển, dùng để phát triển Front-end. React cho phép tạo ra các ứng dụng web có khả năng cập nhật và phản hồi vô cùng nhanh chóng thông qua các compoment.

Khái niệm về Mode.js, đây là môi trường JacvaScript phía Server, cho phép chạy JacvaScript ngoài trình duyệt. Node.js dùng để xây dựng các dịch vụ phía Server, xử lý các yêu cầu API và giao tiếp với cơ sở dữ liệu.

2.2. Tổng quan về ReactJS

2.2.1. Giới thiệu về ReactJS

React là một thư viện JavaScript mã nguồn mở, được xây dựng và phát triển bởi Facebook, giúp xây dựng giao diện người dùng. Rất nhiều thương hiệu

lớn hàng đầu trên thế giới sử dụng React như Facebook, Instagram, Lazada, Mashable, Twitter,... [1]

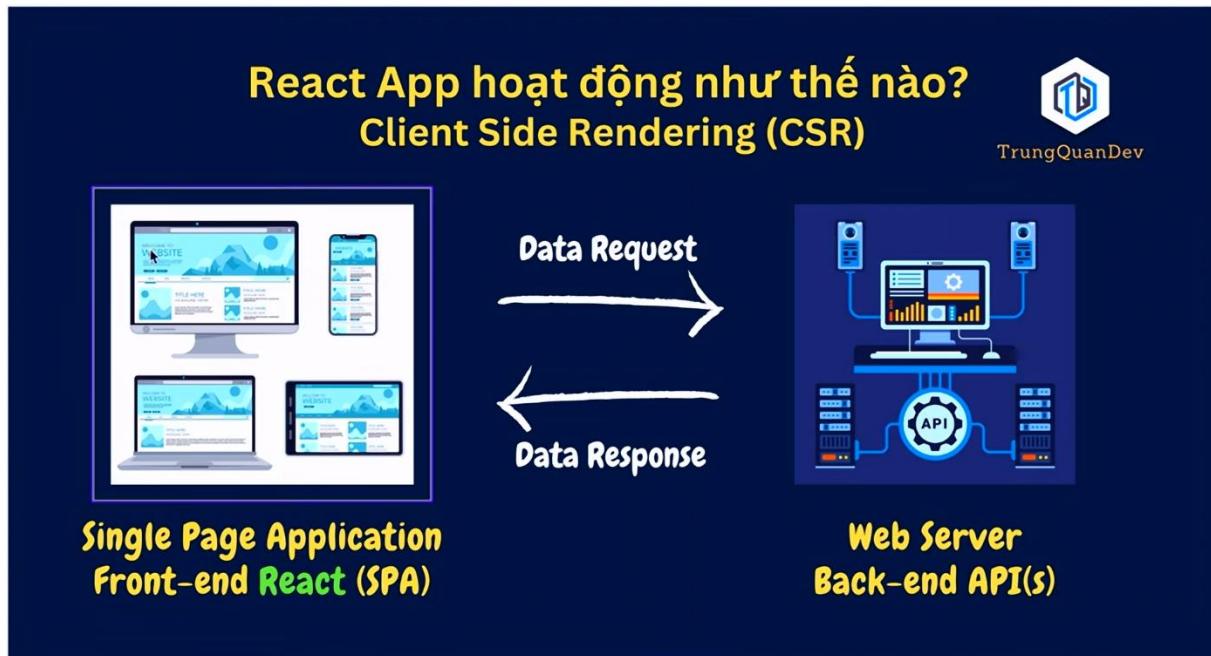
React JS chỉ là một thư viện, không phải là một framework và trong mô hình MVC nó ứng với tầng View (V) [1].

2.2.2. Chức năng React

Single Page Applications (SPA) là một mô hình ứng dụng web vô cùng hiện đại, nó cho phép người dùng tương tác và điều hướng trên trang mà không cần phải tải lại toàn bộ. Mang lại trải nghiệm liền mạch và nhanh chóng cho người dùng [2]. SPA hoạt động bằng cách gửi yêu cầu lấy dữ liệu từ backend thông qua API và sau đó hiển thị dữ liệu này trực tiếp trên giao diện frontend mà không cần tải lại trang. Điều này giúp tăng hiệu xuất và tối ưu hóa trải nghiệm người dùng.

Trong một mô hình SPA, React thường được sử dụng để thực hiện Client Side Rendering (CRS), tức là việc hiển thị nội dung được hiển thị ngay trên trình duyệt của người dùng. Điều này khác với các phương pháp truyền thống khác, nơi toàn bộ trang web phải được tải lại mỗi khi có thay đổi. Với CRS, chỉ những phần cần thiết của trang được cập nhật, mang lại sự mượt mà trong thao tác. So với các phương pháp khác, CRS không chỉ giúp cải thiện tốc độ mà còn giúp nâng cao trải nghiệm tổng thể người dùng.

2.2.3. Sơ đồ hoạt động của React



Hình 2.1. Sơ đồ hoạt động của React

2.2.4. JSX

JSX (JavaScript XML) là một cú pháp mở rộng của JavaScript, được sử dụng phổ biến trong React để mô tả giao diện người dùng [3]. Nó cho phép chúng ta viết các phần tử dao diện (UI element) trông giống như HTML, nhưng thực ra là JavaScript. JSX giúp lập trình viên dễ dàng hơn trong việc tạo ra các thành phần dao diện bằng cách kết hợp JavaScript và HTML trong cùng một File.

Cú pháp JSX: JSX trông rất giống HTML, nhưng nó là JavaScript. Bạn có thể viết các thẻ như trong HTML, nhưng các thẻ này sẽ được biên dịch thành các lời gọi là JavaScript khi ứng dụng được chạy.

Biên dịch: Trước khi được đưa vào trình duyệt, JSX sẽ được chuyển đổi thành các lời gọi React.createElement(). Điều này có nghĩa là dù viết dưới dạng “HTML”, nó vẫn hoạt động như JavaScript thuần phía sau.

Ví dụ về JSX:

```
const element = <h1>Hello, world!</h1>;
```

2.2.5. Props và State

2.2.5.1. *Props*

Prop là viết tắt của “Properties” trong React. Chúng giống như các thuộc tính (attribute) của thẻ HTML, nhưng được dùng để truyền dữ liệu giữa các component trong React.

Một trong số điểm quan trọng và đặc trưng của Props là chúng chỉ có thể đọc trong component con, điều này có nghĩa là component con không được phép thay đổi giá trị của Props. Điều này giúp đảm bảo rằng khi một component nhận cùng một giá trị prop, nó sẽ luôn hiển thị các kết quả giống nhau, tạo ra một sự ổn định và dễ dàng kiểm soát, tái sử dụng trong nhiều phần khác nhau của ứng dụng.

Ví dụ về Props trong đoạn mã sau

```
function Greeting(props) {
  return <h1>Hello, {props.name}!</h1>;
}

function App() {
  return <Greeting name="John" />;
}
```

Component **Greeting** nhận prop **name** và hiển thị “Hello, Jonh!”. Đây là một ví dụ đơn giản cho việc Props truyền dữ liệu từ **App** xuống dưới **Greeting**.

2.2.5.2. *State*

Ngoài ra trong React còn có khái niệm về State, một đối tượng được sử dụng để lưu trữ dữ liệu hoặc trạng thái của một component. State nó có thể thay đổi trong suốt vòng đời của component, và mỗi khi State thay đổi, React sẽ tự động render lại component đó để hiển thị nội dung mới. Điều này khác với Props, vốn chỉ là dữ liệu đọc từ component cha mà không thể bị thay đổi trong component con.

Ví dụ về State trong đoạn mã sau

```

import React, { useState } from 'react';

function Counter() {
    const [count, setCount] = useState(0);

    return (
        <div>
            <p>You clicked {count} times</p>
            <button onClick={() => setCount(count +
1)}>
                Click me
            </button>
        </div>
    );
}

```

Ở đây, count là một State của component counter. Mỗi lần người dùng bấm nút, count sẽ tăng lên, và component sẽ tự động được render lại để phản ánh sự thay đổi đó. Điều này làm cho State trở thành một công cụ mạnh mẽ trong việc quản lý dữ liệu thay đổi trong giao diện.

2.2.6. Tổng quan về Node.js

2.2.6.1. Giới thiệu về Node.js

Node.js là một môi trường chạy bằng mã JavaScript mã nguồn mở và đa nền tảng, cho phép thực thi mã bên ngoài trình duyệt [4]. Node.js thường được sử dụng để xây dựng các dịch vụ phía server (Còn gọi là API), phục vụ cho các ứng dụng web hoặc di động mà người dùng tương tác.

2.2.6.2. Đặc điểm Node.js

Với quy mô cao và hiệu xuất, Node.js lý tưởng cho việc xây dựng các dịch vụ có khả năng mở rộng lớn, xử lý dữ liệu nặng và yêu cầu thời gian thực. Nhiều công ty lớn như PayPal, Uber, Netflix và Walmart đã áp dụng Node.js thành công. Ngoài ra tốc độ phát triển nhanh với Node.js, một ứng dụng có thể được phát triển nhanh hơn với ít dòng mã và tập tin hơn so với việc sử dụng các công nghệ khác như Java hay Spring. Ví dụ, PayPal đã tái xây dựng một ứng dụng và thấy rằng nó mất ít thời gian và tài nguyên hơn.

Sử dụng JavaScript, nếu bạn là lập trình viên Front-end biết JavaScript, bạn có thể dễ dàng chuyển qua phát triển full-stack mà không cần học ngôn ngữ mới. Điều này tạo mã nguồn sạch hơn và nhất quán hơn.

Hệ sinh thái phong phú, Node.js cung cấp một hệ sinh thái lớn các thư viện open-source, cho phép bạn dễ dàng thêm tính năng mà không cần phải xây dựng lại từ đầu.

2.2.6.3. Cấu trúc cơ bản Node.js

Module:

Thư viện JavaScript được tích hợp sẵn trong Node.js, chứa các chức năng cơ bản để phát triển ứng dụng web. Một trong những module quan trọng là “Http” giúp Xử lý các yêu cầu HTTP trong hệ thống một cách hiệu quả, Util cung cấp các tiện ích chung cho việc lập trình, Fs cho việc quản lý hệ thống tệp (file system) hay Url để phân tích và xử lý các URL, đều mang lại những sự thuận tiện và linh hoạt cho việc lập trình hơn. Hơn nữa, khả năng về quản lý dữ liệu qua các stream (luồng) của Node.js rất là ấn tượng với các loại stream có thể đọc và ghi từng dữ liệu liên tục, hoặc là vừa đọc vừa ghi như trong các ứng dụng ghi thời gian thực.

Console:

Trong Node.js tương tự như bảng điều khiển (Console) trong trình duyệt khác. Giúp nó in thông tin ra stdout (tiêu chuẩn đầu ra) và stderr (tiêu chuẩn lỗi), dùng để debug hoặc theo dõi trạng thái.

Cluster:

Module Cluster một module cho phép chạy đa luồng (multi-threading) bằng cách tạo ra các quy trình con (Child processes), giúp phân phối tải và tăng hiệu xuất xử lý, đồng thời cho phép người dùng chạy đồng bộ trên cùng một cổng máy chủ, giúp cho tăng cường khả năng xử lý đồng thời

Global:

Là các biến toàn cục trong Node.js có thể truy cập từ bất kỳ nơi đâu trong ứng dụng. Các biến phổ biến bao gồm:

- dirname: Đường dẫn thư mục hiện tại.

- filename: Đường dẫn đầy đủ của tệp hiện tại.
- Exports: Đè xuất dữ liệu từ một module.
- Require: Nhập một module khác vào.

Error Handling (Xử lý lỗi):

Về **Error Handling**, Node.js cũng cung cấp nhiều cơ chế về mặt xử lý khác nhau, từ lỗi chuẩn của JavaScript như TypeError, ReferenceError, đến các lỗi của hệ thống liên quan đến tài nguyên của hệ điều hành. Nó cũng hỗ trợ xử lý lỗi cụ thể do người dùng tạo ra và các lỗi khảng định khác, khi điều kiện không được thỏa mãn trong mã nguồn.

Streaming:

Một khía cạnh khác nữa là **Streaming**, Node.js sử dụng các stream khác để đọc và ghi dữ liệu một cách liên tục mà không cần tải lên toàn bộ dữ liệu và bên trong của bộ nhớ. Có đến bốn loại stream chính: Readable để đọc dữ liệu, writable để ghi dữ liệu, Duplex để thực hiện đồng thời cả hai thao tác cả đọc và ghi, và Transform cho phép xử lý dữ liệu ngay trong quá trình đọc và ghi,

Domain:

Module **Domain** nó hỗ trợ các quản lý các lỗi chưa được kiểm soát, giúp ngăn ngừa ứng dụng bị crash khi nó xảy ra lỗi. Bạn có thể sử dụng phương pháp Internal Binding để khởi chạy mã trong một domain bằng cách sử dụng run(), hoặc External Binding để có thể thêm các đối tượng hoặc sự kiện vào domain với add.

DNS:

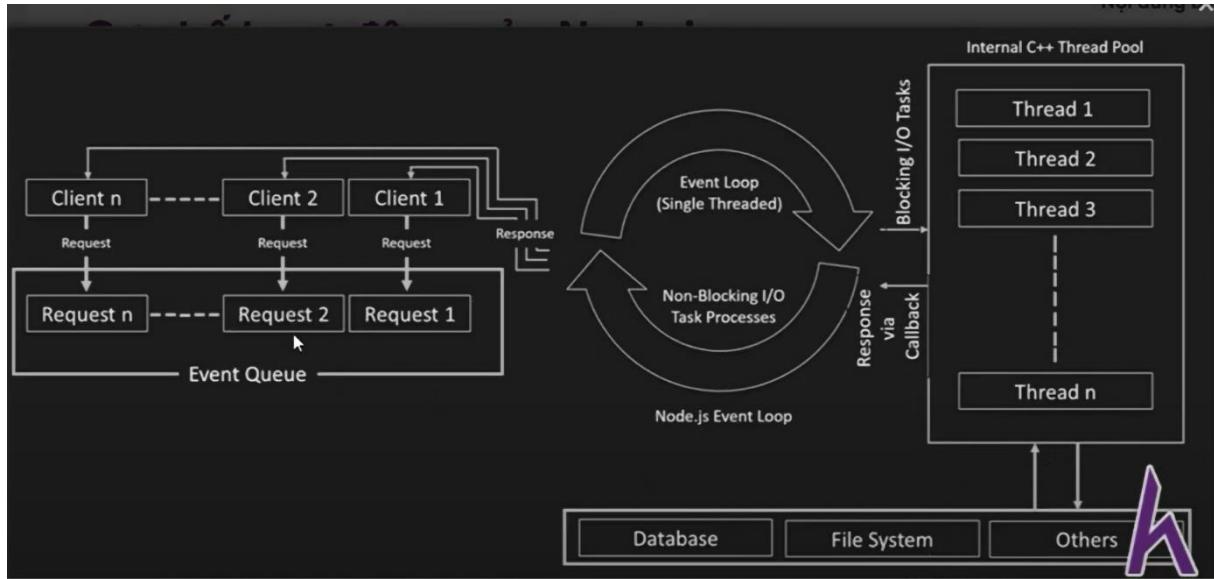
Một module khác trong Node.js, cung cấp khả năng kết nối và phân giải tên miền. Với phương thức dns.resolve, bạn có thể kết nối máy chủ DNS để phân giải tên miền, trong khi dns.lookup nó cho phép thực hiện phân giải mà không cần kết nối mạng.

Debugger:

Cuối cùng là **Debugger**, là một công cụ tích hợp Node.js giúp đỡ gỡ lỗi ứng dụng. Để sử dụng, bạn chỉ cần thêm từ khóa inspect trước khi bạn cho chạy

tệp JavaScript, giúp theo dõi quá trình thực thi và phát hiện lỗi một cách dễ dàng.

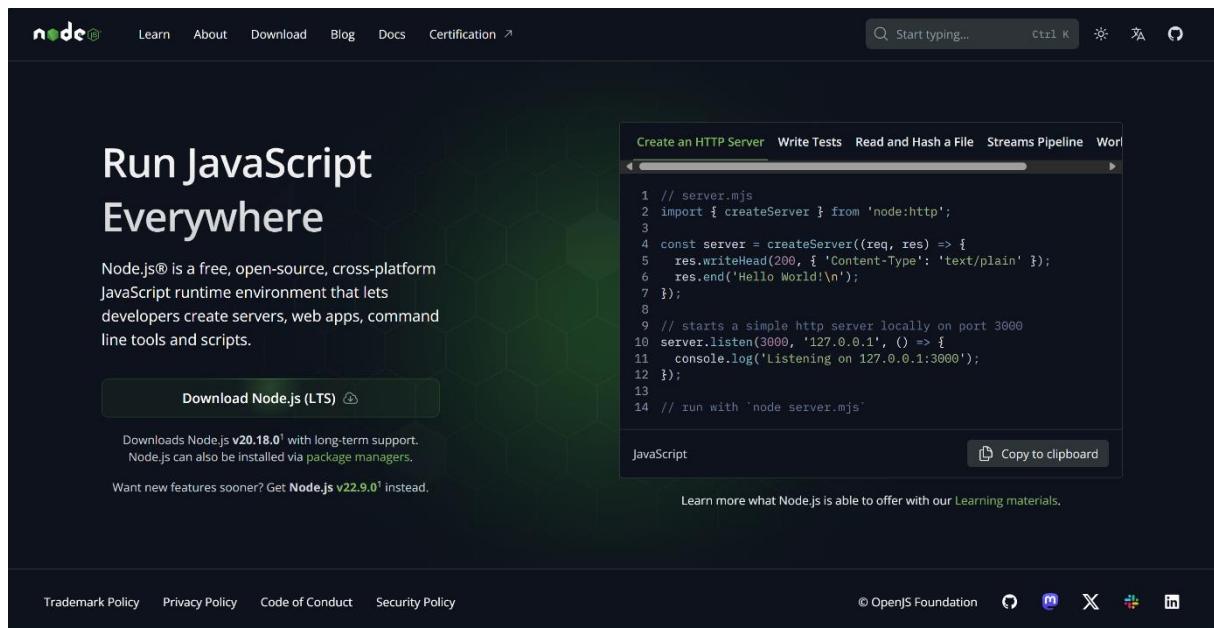
2.2.6.4. Sơ đồ hoạt động Node.js



Hình 2.2. Sơ đồ hoạt động của Node.js

2.2.6.5. Hướng dẫn cài đặt Node.js

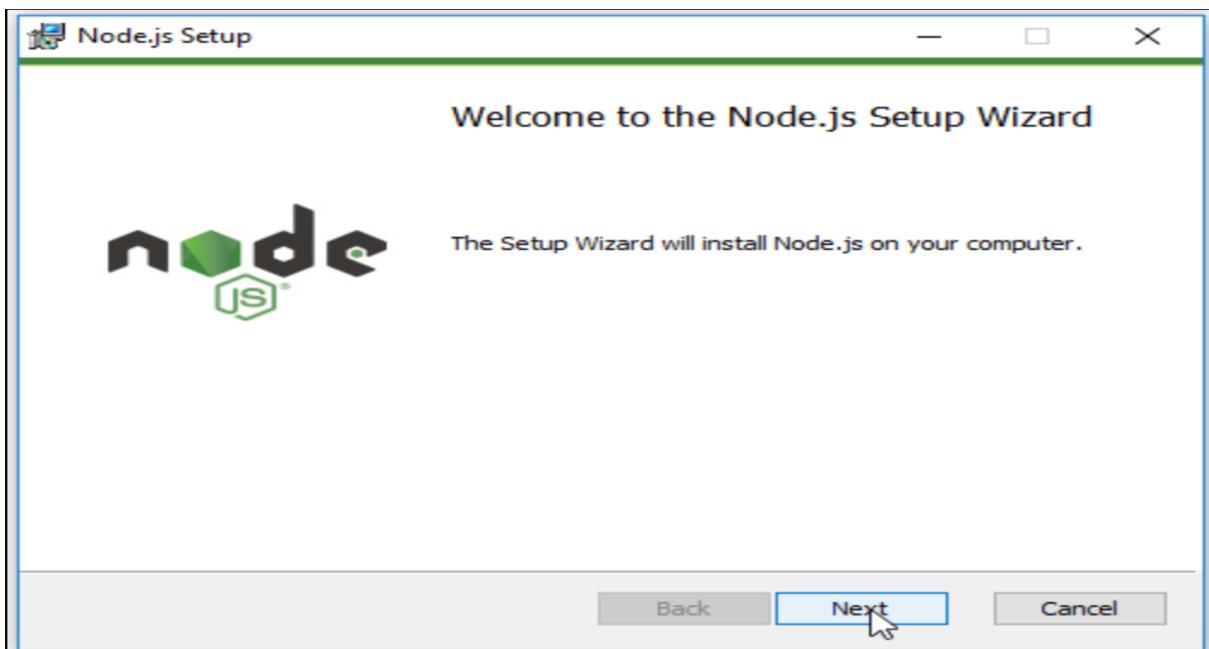
Truy cập nodejs.org. Màn hình chính của Website sẽ trông như sau



Hình 2.3. Truy cập nodejs.org

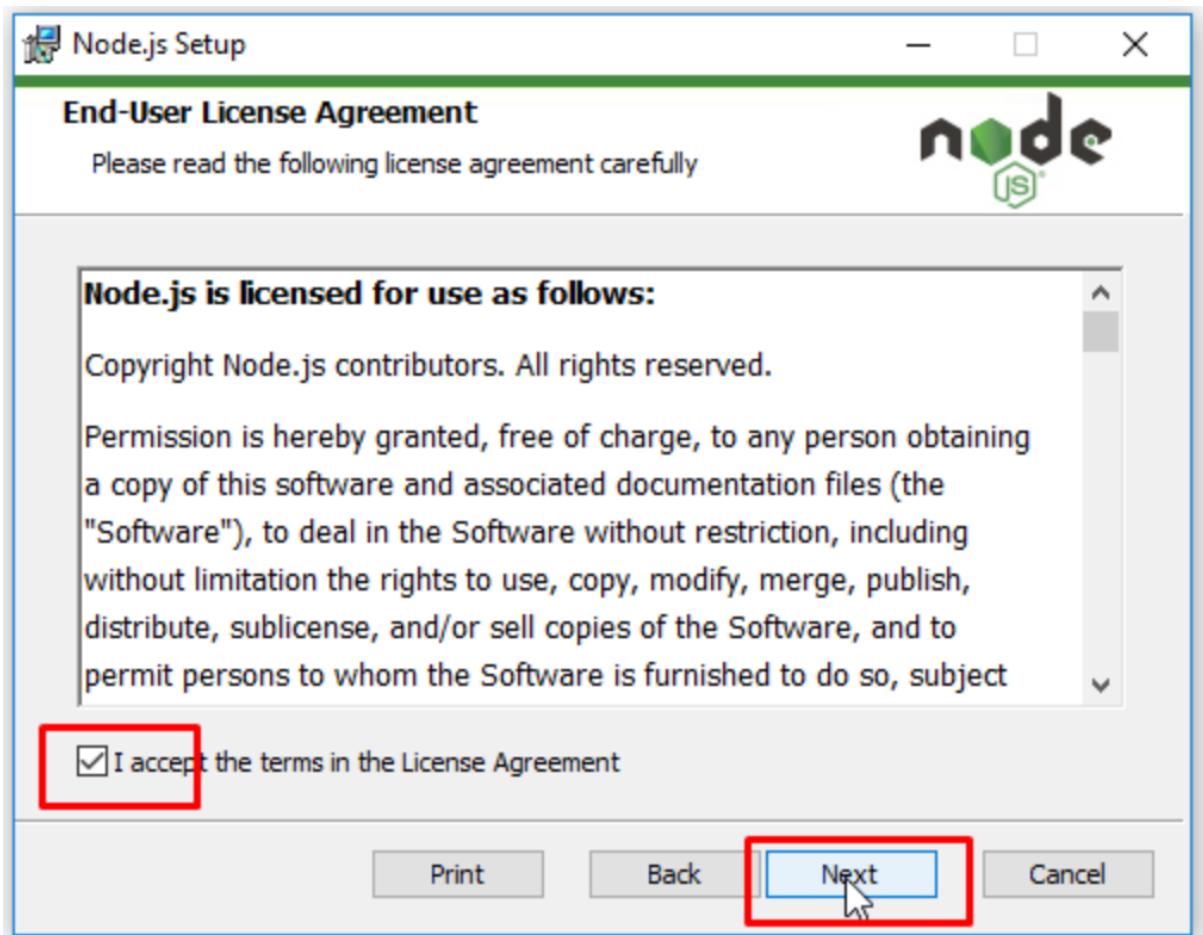
Tiếp theo để tải Node.js, chọn ô Download Node.js (LTS) (khuyến khích nên chọn bản có chữ LTS). Sau khi chọn chỉ cần chọn vị trí lưu File và click “Save” để tải về.

Mở File và cài đặt lên máy, chờ màn hình hiển thị.



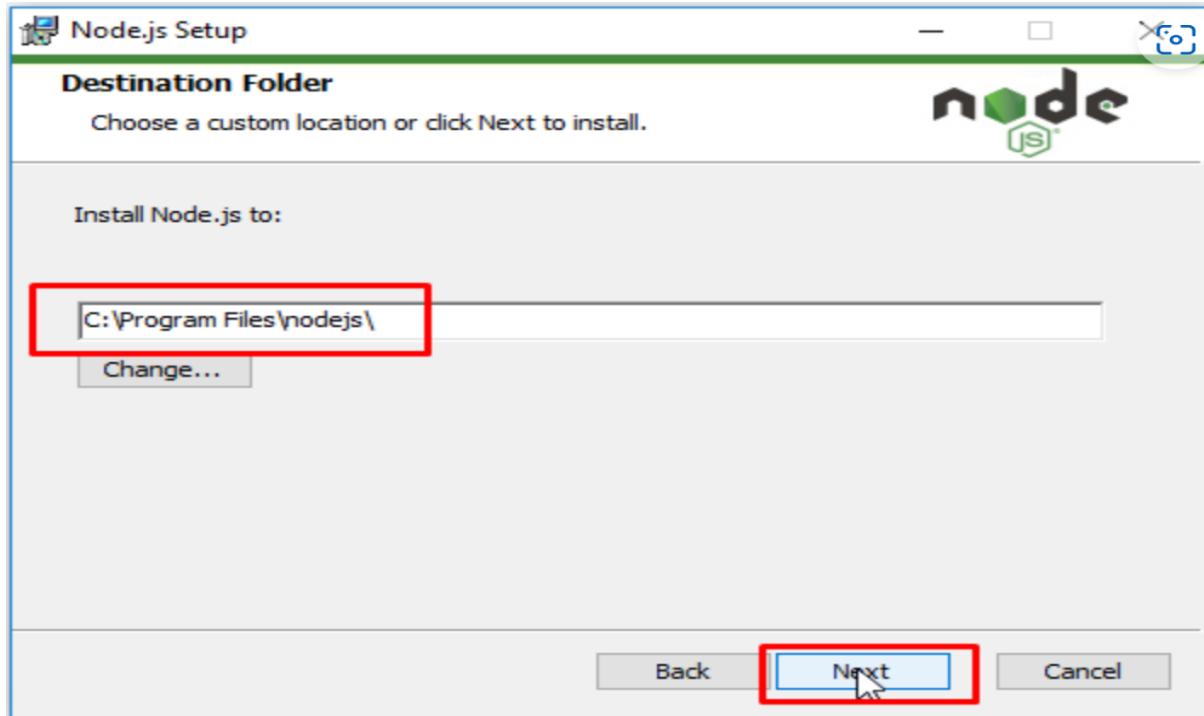
Hình 2.4. Mở file

Chọn **next**, sau đó click vào I accept the terms in the License Agreement, nhấn **next**.



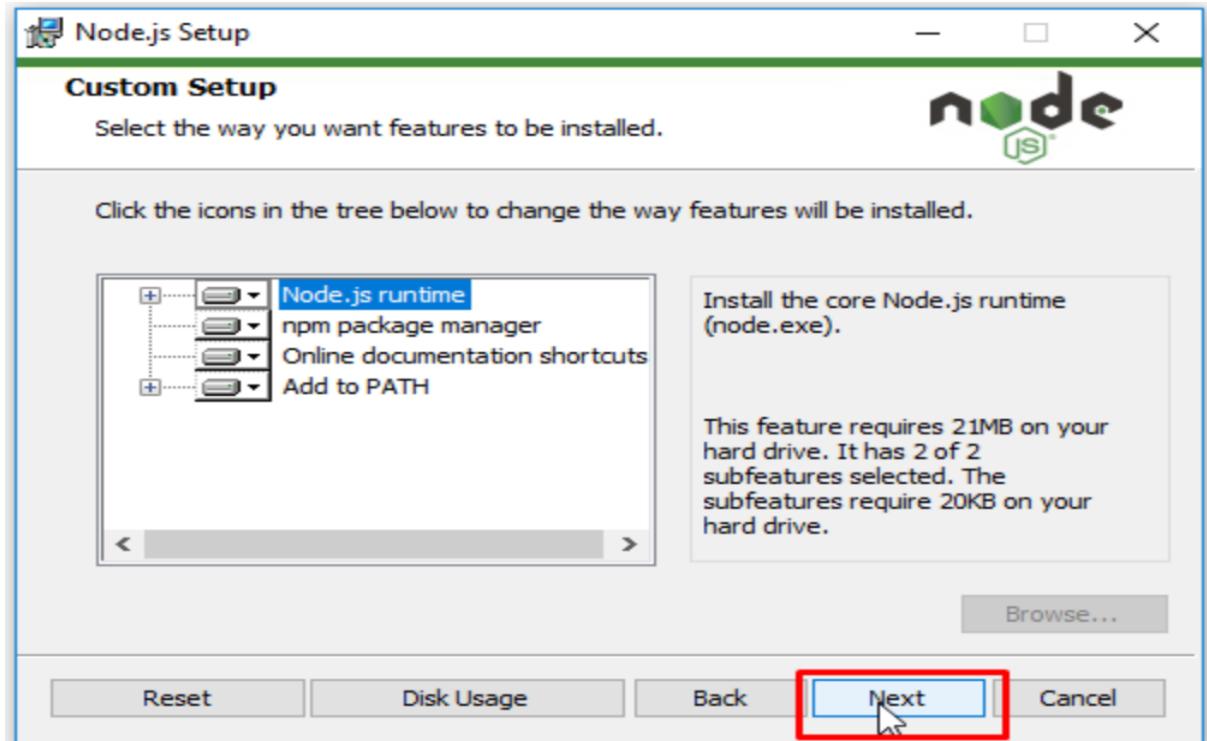
Hình 2.5. Click vào I accept the terms in the License Agreement

Chọn đường dẫn cho Node, nhấn **next**.



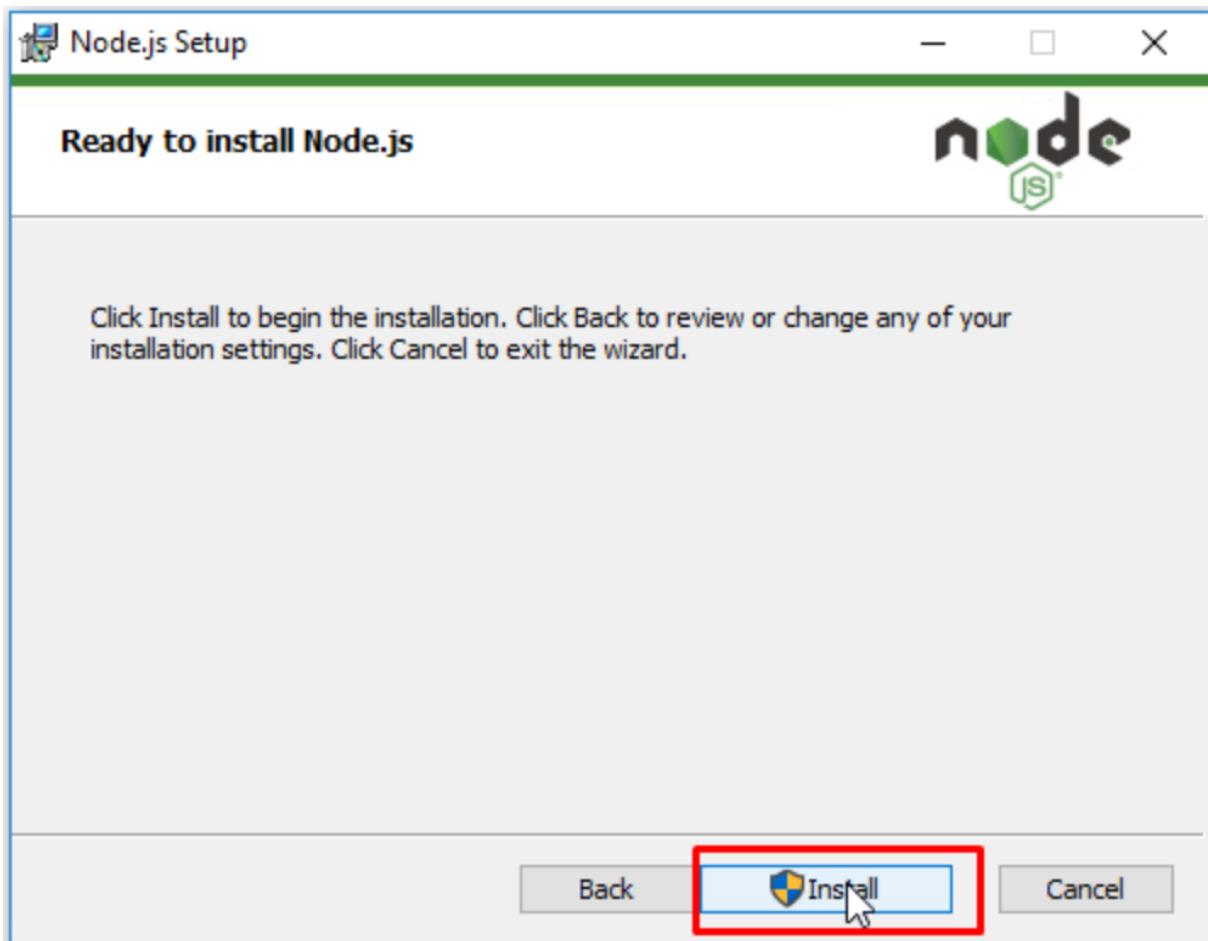
Hình 2.6. Nhấn next

Cửa sổ hiện ra (Chọn theo gợi ý).



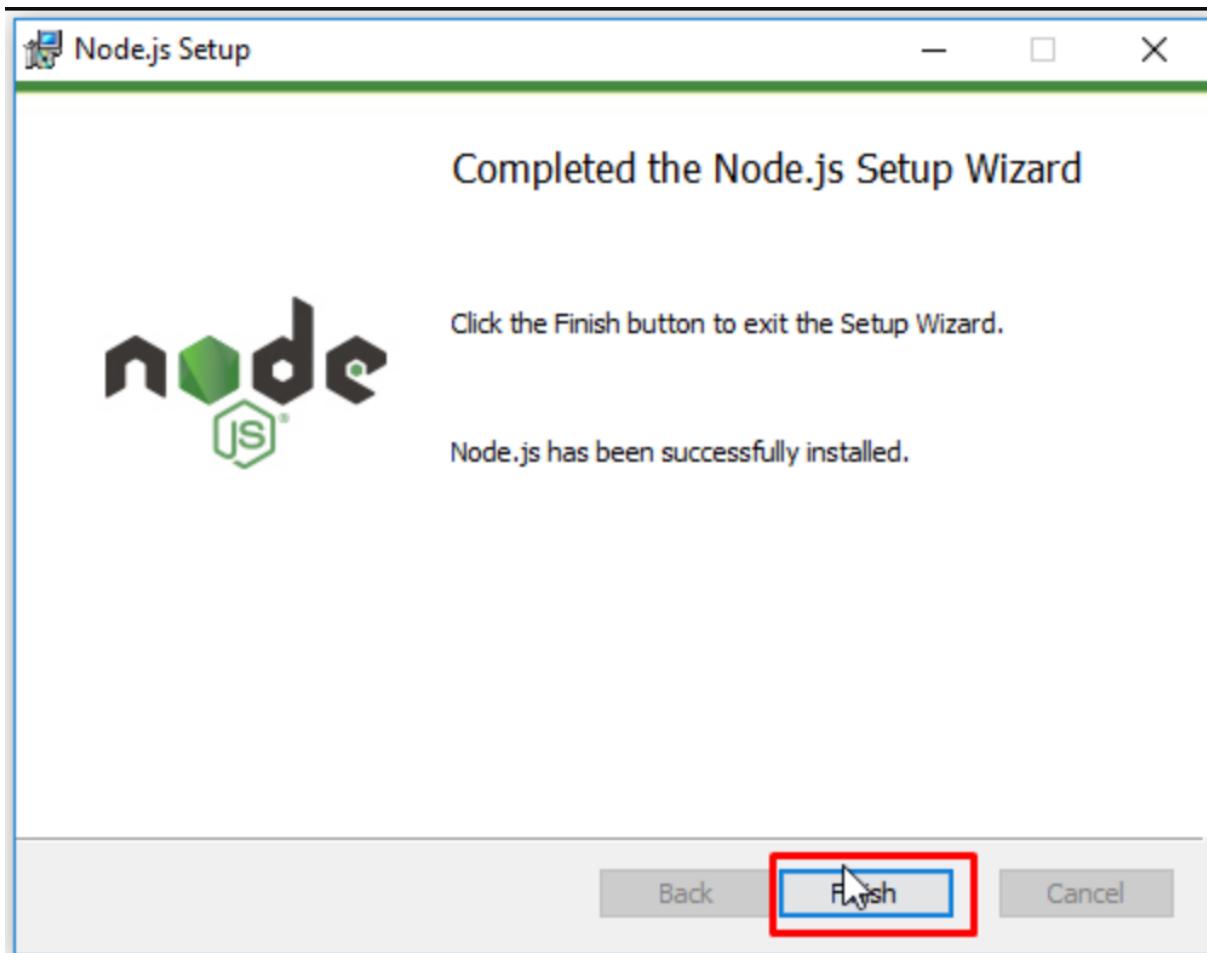
Hình 2.7. Cửa sổ hiện ra (Chọn theo gợi ý)

Chọn **Install**.



Hình 2.8. Install

Chờ hoàn tất.



Hình 2.9. Chờ hoàn tất

Sau khi hoàn tất, mở **cmd** và kiểm tra xem node được cài đặt hay chưa bằng cú pháp **node -v**. Hiện ra phiên bản cài đặt thì xem như đã thành công.

```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>node -v
v20.17.0

C:\Windows\System32>

```

Hình 2.10. Kiểm tra thông qua cmd

2.2.7. Tổng quát về MongoDB.

2.2.7.1. Giới thiệu MongoDB

MongoDB là một cơ sở dữ liệu NoSql, cho phép người dùng lưu trữ dữ liệu dưới dạng tài liệu (document) với cấu trúc linh hoạt [5]. Khác với các hệ thống cơ sở dữ liệu quan hệ (RDBMS) truyền thống sử dụng bảng và hàng để lưu trữ dữ liệu, MongoDB sử dụng các bộ sưu tập (Collections) và tài liệu [5]. Đây là một giải pháp tối ưu cho việc xử lý các ứng dụng yêu cầu tính linh hoạt và mở rộng dữ liệu nhanh chóng, đặc biệt là các ứng dụng web hiện đại.

2.2.7.2. Cơ chế hoạt động.

MongoDB hoạt động dựa trên mô hình lưu trữ tài liệu, khác với các cơ sở dữ liệu quan hệ.

Tài liệu (Document)

Giới thiệu về tài liệu (Document), tài liệu là đơn vị lưu trữ rất cơ bản của MongoDB, được định nghĩa bằng các cặp khóa-giá trị (key-value pairs), rất

giống với định dạng Json. Cho phép tài liệu chứa dữ liệu có cấu trúc khác nhau và dễ dàng mở rộng.

Bộ sưu tập (Collections)

Một collections là tập hợp của các tài liệu có liên quan. Collection không yêu cầu các tài liệu phải có cùng một cấu trúc, điều này khác với các bảng trong cơ sở dữ liệu quan hệ.

Mở rộng ngang (Horizontal Scaling)

MongoDB hỗ trợ tính năng mở rộng ngang (sharding), cho phép hệ thống tự phân chia dữ liệu trên nhiều máy chủ. Giúp xử lý các yêu cầu lớn mà không gặp phải các giới hạn phần cứng.

Sao lưu và phục hồi (Replication)

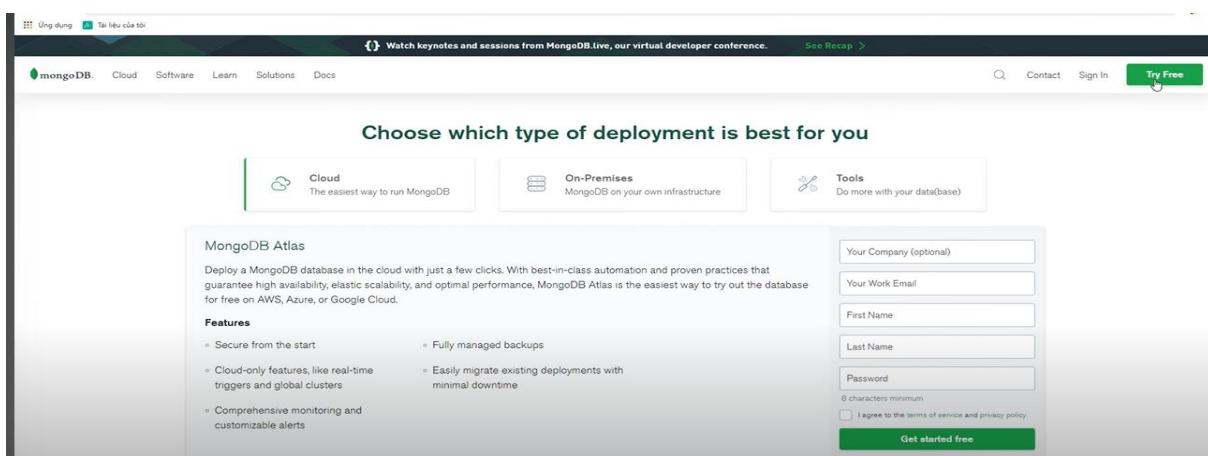
MongoDB hỗ trợ cơ chế sao lưu dữ liệu, để đảm bảo sẵn sàng và an toàn cho dữ liệu. Cụ thể, sử dụng “replica set”, bao gồm một bản chính (primary) và nhiều bản sao (secondary).

Hỗ trợ các CRUD

Cung cấp các thao tác cơ bản như tạo (Create), đọc (Read), cập nhật (Update) và xóa (Delete), cho phép thực hiện các tác vụ quản lý dữ liệu dễ dàng và nhanh chóng.

2.2.7.3. Cài đặt MongoDB.

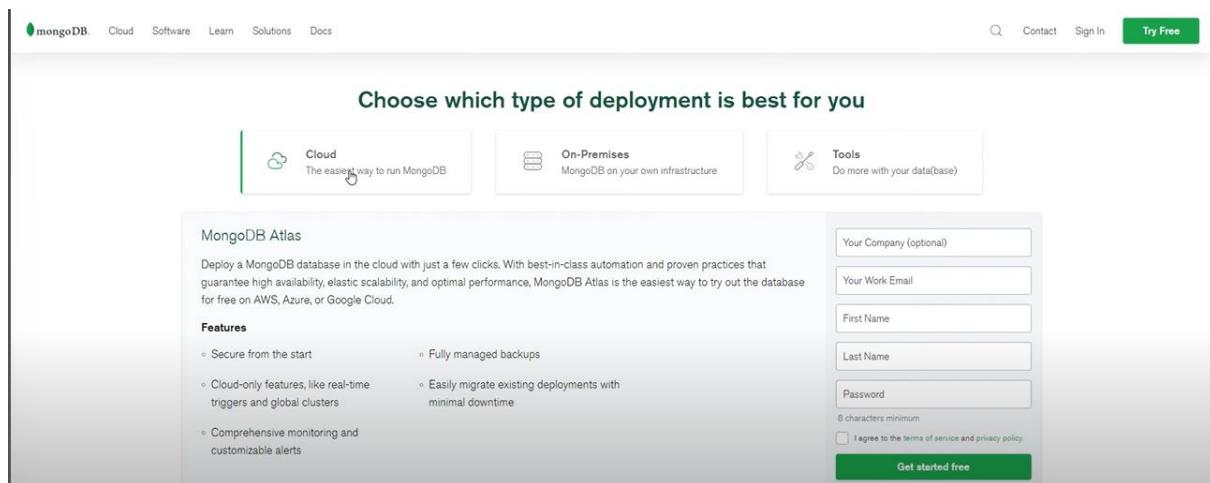
Truy cập trang mongodb.com/try



Hình 2.11. Truy cập trang mongodb.com/try

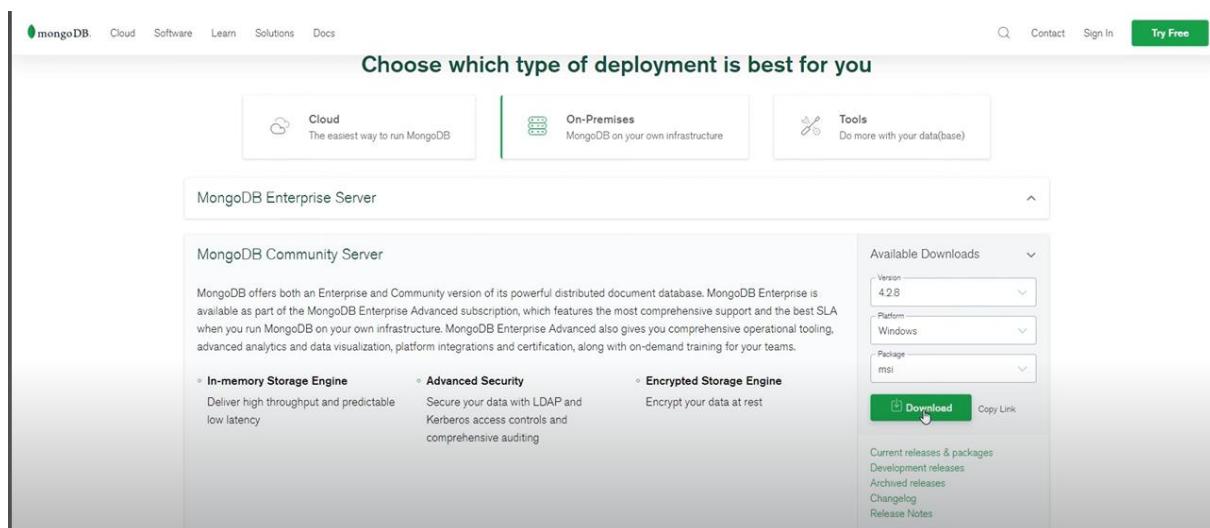
Chọn phiên bản.

- Phiên bản Cloud (trên đám mây).
- Phiên bản Community (miễn phí).
- Phiên bản Enterprise (có phí).



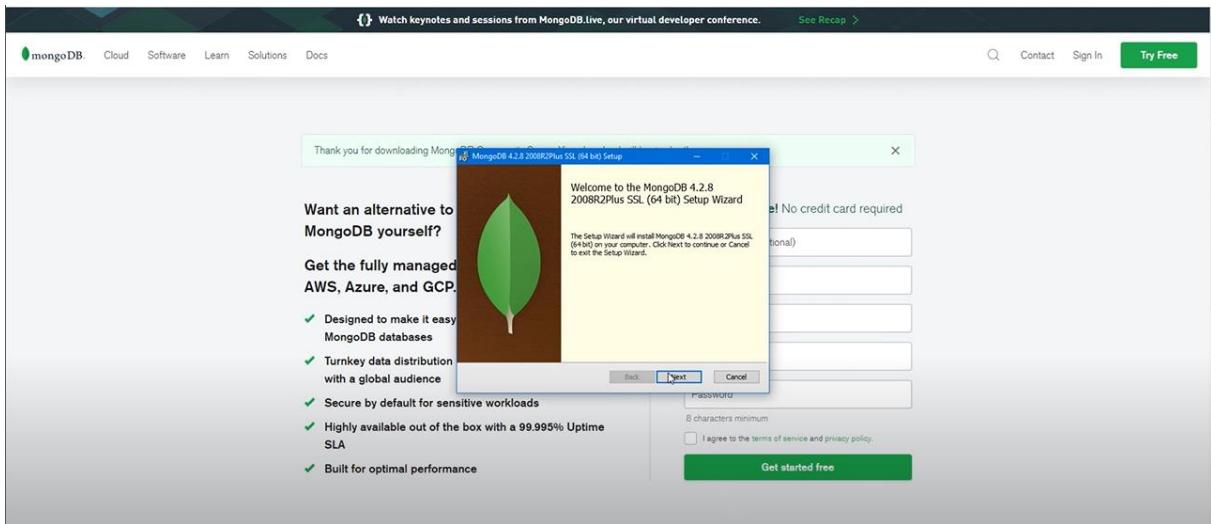
Hình 2.12. Chọn phiên bản

Tải và cài đặt



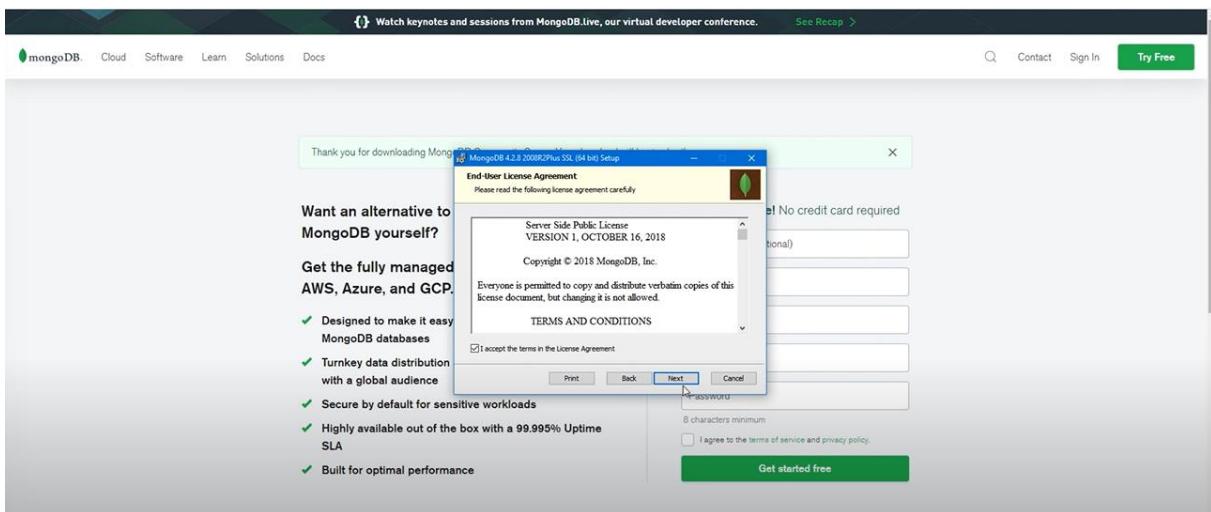
Hình 2.13. Tải và cài đặt

Cửa sổ hiện lên, chọn **next**.



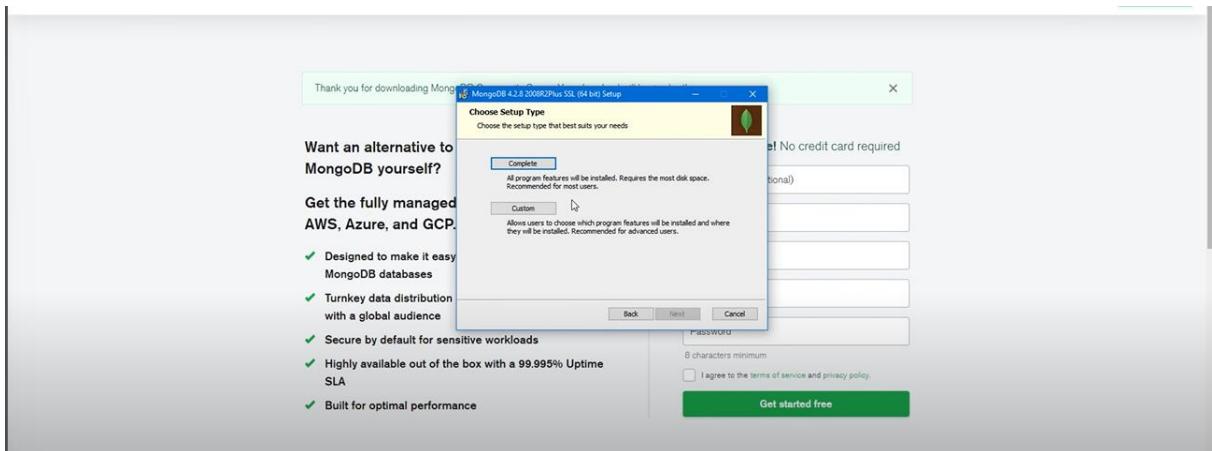
Hình 2.14. Cửa sổ hiện lên, chọn next

Click vào I accept the terms in the License Argreement, nhấn **next**.



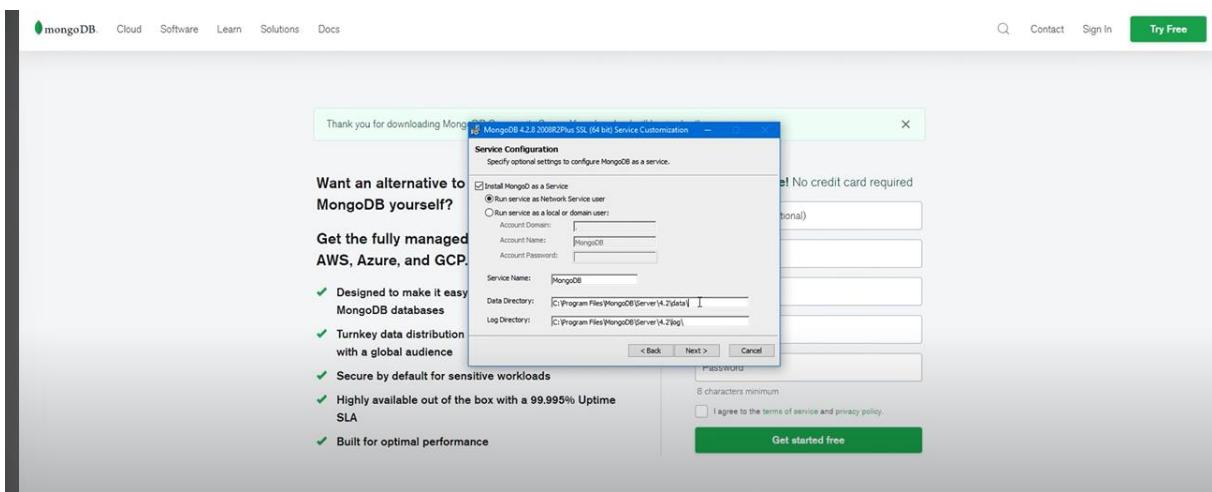
Hình 2.15. Click I accept the terms in the License Argreement, nhấn next

Chọn Complete



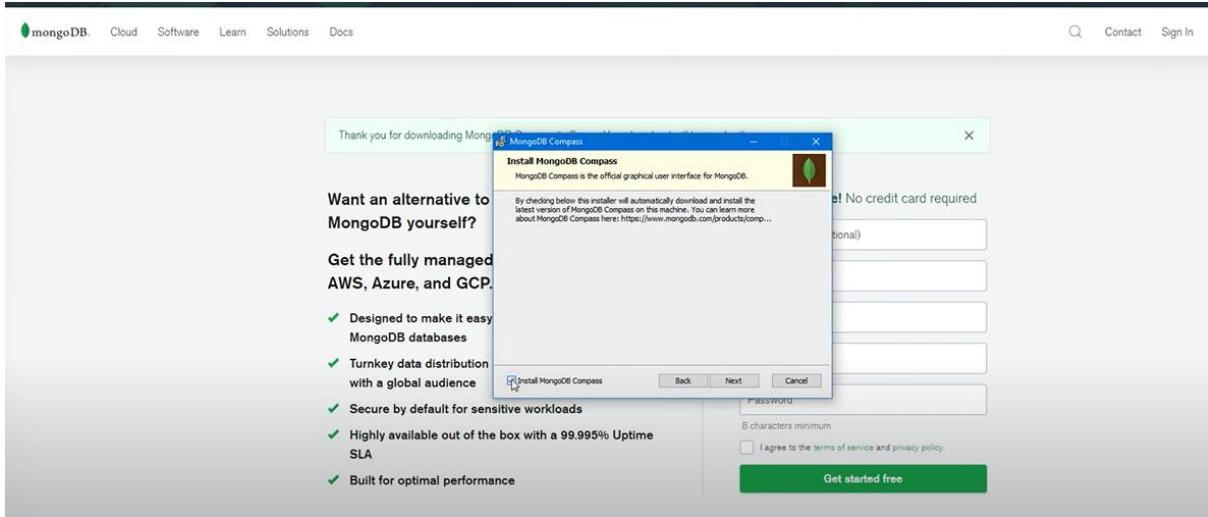
Hình 2. 16. Chọn Complete

Có thể thay đổi mục lưu trữ sau đó chọn next.



Hình 2. 17. Có thể thay đổi mục lưu trữ, sau đó chọn next

Tiếp tục Click vào **Install MongoDB Compass**, nhấn **next**.



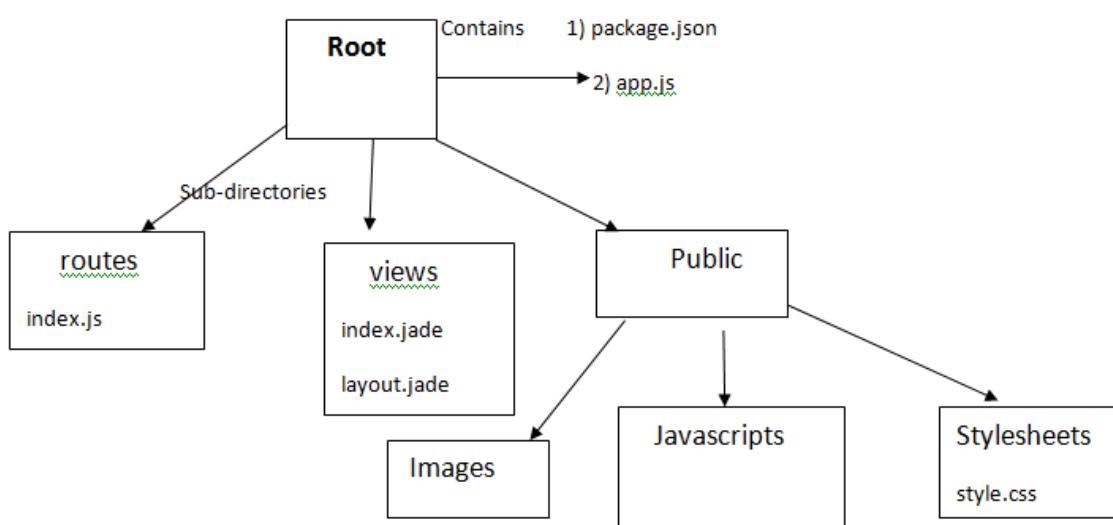
Hình 2.18. Hoàn tất

2.2.8. Tổng quan về ExpressJS

2.2.8.1. Giới thiệu về ExpressJS

Là một framework nhỏ gọn nhưng rất linh hoạt, được xây dựng trên nền tảng Node.js. Nó cung cấp nhiều tính năng mạnh mẽ hỗ trợ cho việc phát triển ứng dụng web hoặc mobile [6].

2.2.8.2. Cấu trúc của ExpressJS



Hình 2.19. Cấu trúc của ExpressJS

Cấu trúc của một ứng dụng Express.js rất đơn giản bao gồm các mục:

Root (Thư mục gốc): Đây là nơi chứa toàn bộ các tệp và thư mục ứng dụng

App.js: File này chứa các thiết lập và cấu hình cần thiết để ứng dụng hoạt động, bao gồm việc khai báo các middleware, định nghĩa route và các cấu hình khác.

package.json: Tệp này liệt kê các gói (package) mà ứng dụng cần để hoạt động. Nó tương tự như composer.json trong PHP hoặc Gemfile trong Ruby on Rails, được dùng để quản lý các thư viện và phụ thuộc của ứng dụng.

Folder routes: Chứa các file quản lý đường dẫn (route) của ứng dụng, nói xác định các URL và hành động tương ứng.

Folder views: Chứa các tệp giao diện (view/template) của ứng dụng, thường là các file HTML hoặc các file động như EJS, Pug.

Folder public: chứa các tệp tĩnh như CSS, JavaScript, hình ảnh... dùng để phục vụ giao diện của ứng dụng.

2.2.8.3. Tại sao sử dụng Express.JS

Sử dụng Express.JS mang lại nhiều lợi ích cho nhà phát triển ứng dụng web với Node.js, giúp đơn giản hóa các thao tác và tăng hiệu quả phát triển.

Nhẹ và linh hoạt: Express.JS là một framework nhỏ gọn và không yêu cầu cấu hình phức tạp [7]. Nó cho phép bạn tự do xây dựng ứng dụng theo cách riêng, mà không bị ràng buộc bởi các quy tắc nghiêm ngặt.

Tối ưu hóa và phát triển: : Express.JS đơn giản hóa việc quản lý các tuyến đường (router) và xử lý HTTP requests [7]. Điều này giúp lập trình viên dễ dàng xử lý các yêu cầu GET, POST, PUT, DELETE từ Client một cách nhanh chóng, không cần viết mã phức tạp như sử dụng Node.js thuần.

Quản lý middleware mạnh mẽ: : Express.JS hỗ trợ việc sử dụng các middleware- các chức năng được xử lý tuần tự cho mỗi request, từ kiểm tra xác thực (authentication), xử lý dữ liệu đến quản lý lỗi [7]. Bạn có thể dễ dàng chèn

các middleware tùy chỉnh vào ứng dụng của mình để xử lý các yêu cầu trước khi gửi phản hồi đến người dùng.

Hỗ trợ cho API: : Express.JS là một lựa chọn hoàn hảo để xây dựng các API Restful [7]. Với các công cụ đơn giản và linh hoạt, Express giúp tạo ra các API mạnh mẽ để quản lý tài nguyên và giao tiếp với cơ sở dữ liệu.

Cộng đồng lớn và hệ sinh thái phong phú: Express.JS có cộng đồng lớn và hệ sinh thái phong phú, hỗ trợ rất nhiều thư viện, module có sẵn thông qua NPM (Node package manager) [7]. Điều này giúp tiết kiệm thời gian và công sức cho lập trình viên khi cần thêm các tính năng như xác thực, xử lý tập tin và bảo mật.

Khả năng mở rộng cao: với sự nhẹ nhàng và linh hoạt, Express.JS có khả năng mở rộng dễ dàng, phù hợp để phát triển tất cả ứng dụng nhỏ và các hệ thống lớn, phức tạp.

Xử lý đồng thời nhiều yêu cầu: Nhờ sử dụng mô hình không đồng bộ (asynchronous) của Node.js, Express.JS giúp xử lý nhiều yêu cầu cùng một lúc mà không làm chậm hệ thống [7], giúp tối ưu hóa hiệu suất sử dụng web.

2.2.9. Tìm hiểu Router trong Express

Router là một đoạn URL cụ thể, cùng với phương thức HTTP (GET, POST, PUT, DELETE,...) được dùng để xử lý các yêu cầu từ client. Một router sẽ có cấu trúc như sau:

```
App.get('/example', (req,res)=> {
    res.send('This is example route');
});
```

Ví dụ trên, ứng dụng sẽ trả về thông báo “This is example route” khi người dùng truy cập vào đường dẫn /example với phương thức GET.

Router instance: trong Express, bạn có thể tạo một instance của router bằng cách sử dụng express.Router(). Điều này cho phép tách route thành nhiều module hoặc file riêng biệt, giúp mã nguồn dễ quản lý hơn khi ứng dụng chở lên lớn hơn.

Ví dụ:

```

Const express = require('express');
Const router = express.Router();
router.get('/users', (req, res) => {
    res.send('List of user');
});
Const app = express();
app.use('/api', router);

```

Middleware : Router cũng hỗ trợ middleware, là các hàm được gọi trước khi route handler thực sự thực thi. Middleware trong router có thể được sử dụng để kiểm tra xác thực (authentication), log dữ liệu, hoặc tiền xử lý yêu cầu.

Phân chia roue : Router giúp phân chia các route theo từng phần cụ thể của ứng dụng, ví dụ: route liên quan đến người dùng, sản phẩm, đơn hàng... Bạn có thể dễ dàng tách riêng và quản lý từng phần bằng router.

2.2.10. Giới thiệu về RESTful API

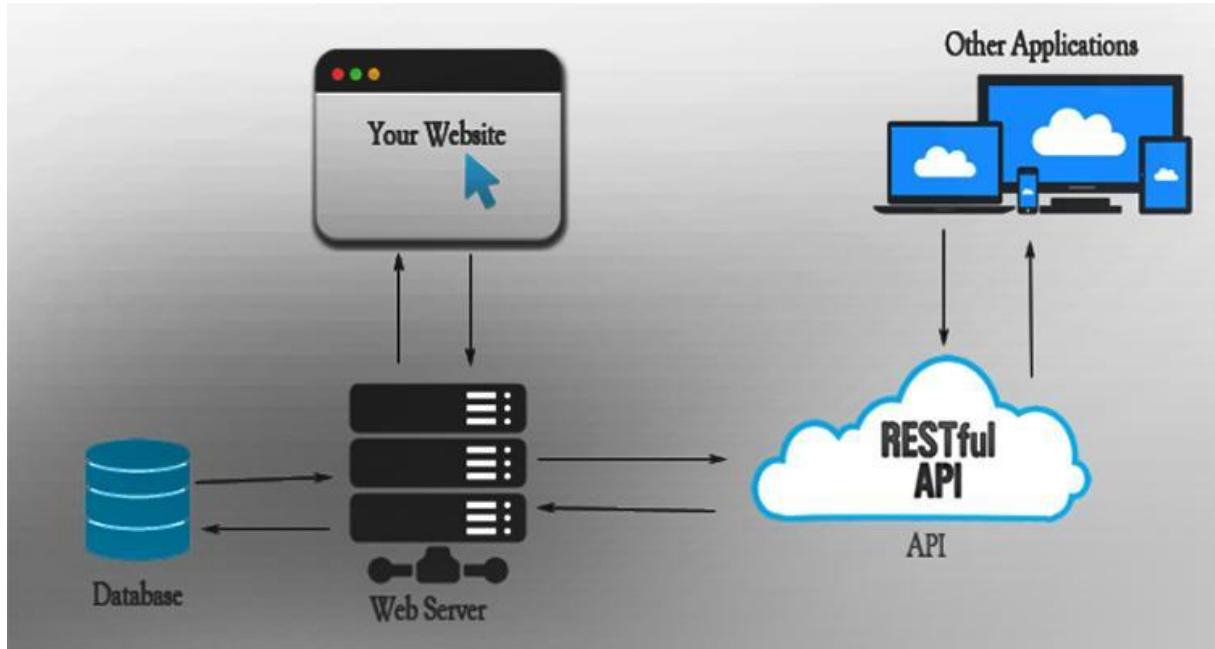
2.2.10.1. Khái quát về RESTful API

RESTful API (Representational State Transfer API) là một kiểu kiến trúc cho việc thiết kế các dịch vụ web (web services), được xây dựng dựa trên các nguyên tắc của giao thứ HTTP. RESTful API cho phép các hệ thống khác nhau giao tiếp với nhau thông qua HTTP mà không cần trạng thái (stateless) giữa các yêu cầu (requests).

API: Được viết là (Application Programming Interface) là một tập hợp các giao thức, công cụ và định dạng để xây dựng phần mềm và cho phép các ứng dụng giao tiếp với nhau. API hoạt động như cầu nối, giúp các ứng dụng hoặc hệ thống có thể trao đổi dữ liệu và dịch vụ một cách dễ dàng mà không cần biết chi tiết nội bộ của nhau. API là một phần quan trọng trong lập trình hiện đại, giúp các hệ thống và ứng dụng kết nối và tương tác với nhau dễ dàng. Nhờ có API, việc phát triển ứng dụng trở nên nhanh chóng, dễ dàng và hiệu quả hơn.

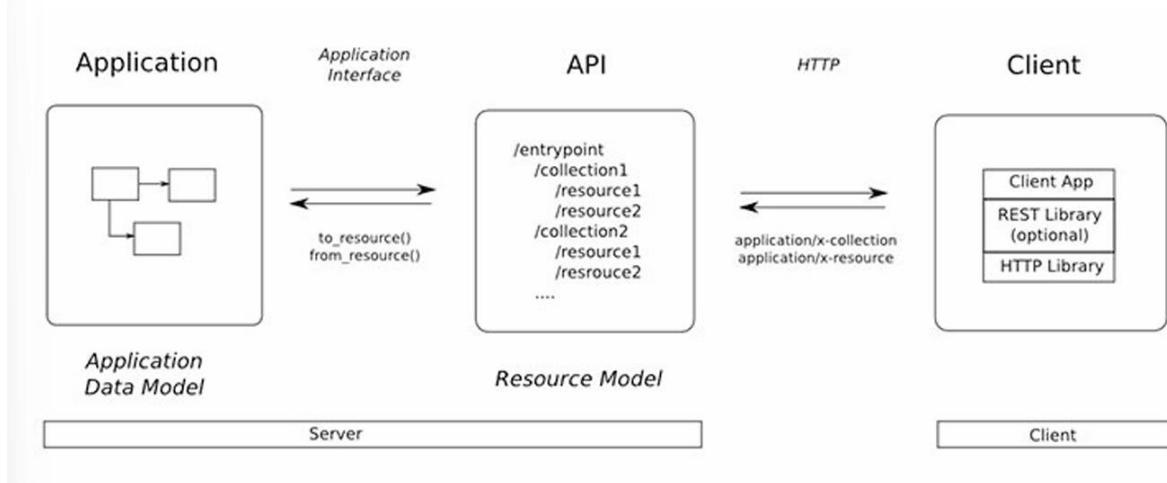
REST: Được viết là (Representational State Transfer) là một kiến trúc phần mềm cho các hệ thống phân tán, đặc biệt là các dịch vụ web. Được định nghĩa bởi Roy Fielding trong luận án tiến sĩ của ông năm 2000 [8] ,REST để

xuất một cách tiếp cận đơn giản và hiệu quả để tạo ra các dịch vụ web dễ mở rộng, dễ hiểu và có khả năng tương tác cao.



Hình 2.20. Cách thức hoạt động của API

2.2.10.2. Cách hoạt động của RESTful API



Hình 2.21. Cách hoạt động của RESTful API

REST hoạt động chủ yếu dựa vào giao thức HTTP, trong đó các thao tác chính sẽ sử dụng các phương thức HTTP tương ứng như sau:

GET (SELECT): Dùng để truy xuất và trả về một tài nguyên hoặc danh sách các tài nguyên.

POST (CREATE): Dùng để tạo mới một tài nguyên.

PUT (UPDATE): Dùng để cập nhật thông tin cho tài nguyên hiện có.

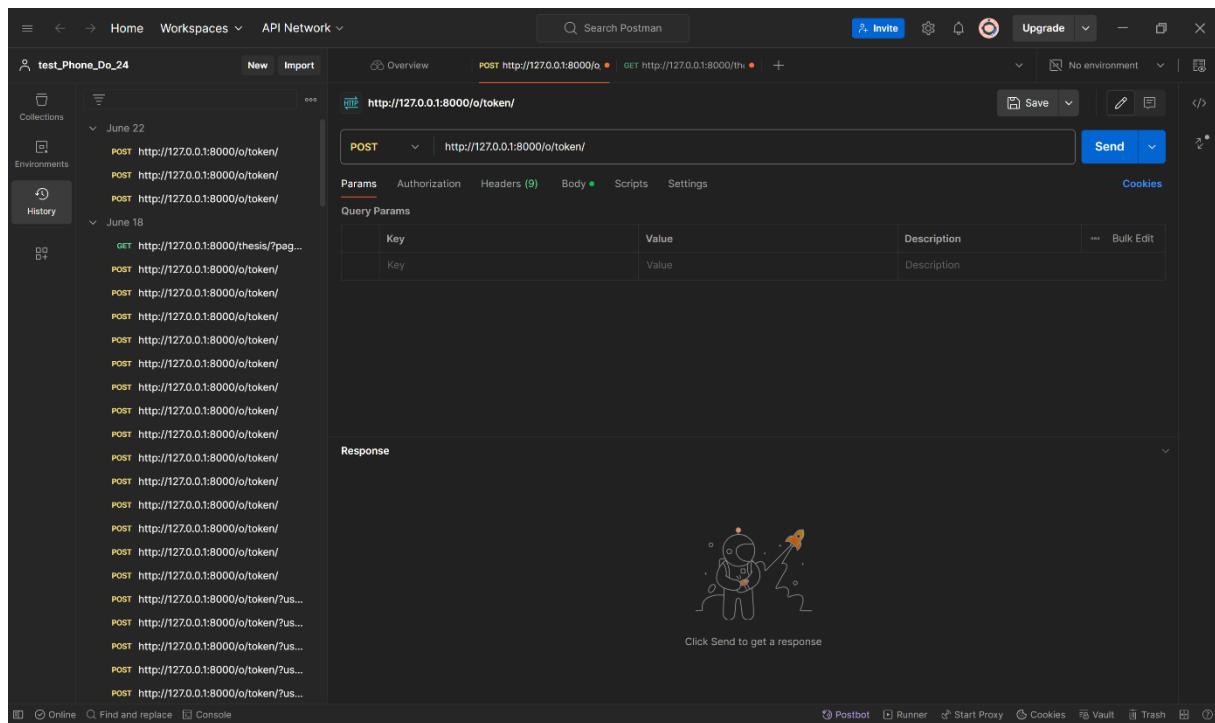
DELETE (DELETE): Dùng để xóa một tài nguyên.

Những phương thức trên thường được gọi là các thao tác CRUD, viết tắt của Create, Read, Update và Delete – tương ứng với các hành động Tạo, Đọc, Cập nhật, Xóa.

2.2.11. Giới thiệu về Postman

2.2.11.1. Khái niệm về Postman

Postman là một công cụ hữu ích cho việc tương tác và làm việc với API, đặc biệt là các API theo chuẩn REST. Postman được sử dụng rộng rãi bởi lập trình viên để kiểm thử API mà không cần viết bất kỳ dòng code nào.



Hình 2.22. Giới thiệu về Postman

2.2.11.2. Một số tính năng của Postman

Thử nghiệm API: Postman cho phép lập trình viên yêu cầu đến API và nhận phản hồi một cách dễ dàng. Điều này giúp kiểm tra các endpoint của API mà không cần phải tạo ứng dụng đầy đủ. Hỗ trợ nhiều phương thức HTTP: Postman hỗ trợ tất cả các phương thức HTTP phổ biến như GET, POST, DELETE, PATCH... giúp lập trình viên dễ dàng kiểm tra mọi loại yêu cầu với

API. Lưu trữ lịch sử yêu cầu, một trong những tính năng tiện lợi của Postman là khả năng lưu lại lịch sử các lần gửi yêu cầu .

2.2.11.3. Lợi ích của Postman

Đối với những lợi ích khi sử dụng Postman là không cần code: Postman cho phép thực hiện các thao tác với API mà không cần viết mã. Quản lý tổ chức yêu cầu, bạn có thể lưu trữ và quản lý các bộ sưu tập (collections) yêu cầu để kiểm tra lại hoặc chia sẻ với người khác.

CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG

Chương này phân tích thiết kế hệ thống, thiết kế cơ sở dữ liệu, đặc tả về lược đồ usecase, sơ đồ hoạt động và các quy trình nghiệp và chức năng của hệ thống..

3.1. Giới thiệu hệ thống

Với đề tài website mua bán thực phẩm ngoại nhập, đây là một loại hình kinh doanh ảo qua mạng bằng cách xây dựng một trang web bán sản phẩm. Để mọi người có thể vào đó xem các loại sản phẩm, lựa chọn giữa các mặt hàng, so sánh các loại thực phẩm, giá tiền. Với trang web này, cửa hàng sẽ đưa thông tin các sản phẩm mới ra mắt, hoặc đang kinh doanh lên trang web của mình. Việc đưa thông tin phải theo trật tự hợp lý. Ví dụ, khi đưa thông tin sản phẩm thì các thông tin sản phẩm phải đầy đủ gồm : Tên sản phẩm, giá tiền, ... Người quản trị trang web cũng có quyền thêm mới, sửa đổi thông tin sản phẩm, hoặc xóa sản phẩm đã dừng bán hoặc hết hàng. Mọi sự thay đổi đều được ghi nhận trong cơ sở dữ liệu để đến với người sử dụng.

Khách hàng thì được xem trang web, xem thông tin các loại sản phẩm đang bán, nhưng không có quyền đặt hàng. Để có thể đặt hàng, khách hàng cần phải đăng ký là thành viên của trang web. Khi đã là thành viên, khách hàng đăng nhập vào website theo email và mật khẩu của mình. Khi đó, khách hàng có quyền đặt hàng. Bên cạnh đó, khách hàng có thể tìm kiếm sản phẩm theo tên sản phẩm, khi đặt hàng xong khách hàng có thể thay đổi số lượng hàng cần đặt theo ý muốn. Khi khách hàng đặt hàng thành công, người quản lý đối chiếu thông tin đơn hàng của khách hàng và thông tin được cập nhật trong hệ thống để tiến hành vận chuyển sản phẩm và thanh toán cho khách hàng.

Ngoài ra trang web còn được tích hợp những tính năng thanh toán đa dạng, giúp khách hàng có nhiều sự lựa chọn khi thanh toán sản phẩm.

3.2. Các chức năng của hệ thống

Các chức năng dành cho khách hàng (Chưa đăng kí)

- Xem các loại sản phẩm, xem thông tin sản phẩm, xem giá tiền, xem sản phẩm theo giá (Từ thấp đến cao hoặc ngược lại).
- Tìm kiếm sản phẩm.
- Đăng kí, đăng nhập.
- Chat với admin

Các chức năng dành cho khách hàng (Đã đăng kí)

- Đăng nhập.
- Xem các loại sản phẩm, xem thông tin sản phẩm, xem giá tiền, xem sản phẩm theo giá (Từ thấp đến cao hoặc ngược lại).
- Tìm kiếm sản phẩm.
- Xem thông tin và cập nhật hồ sơ của mình.
- Đặt hàng, thanh toán hàng.
- Xem thông tin đơn hàng.
- Đặt câu hỏi, đánh giá sản phẩm.

Các chức năng dành cho người dùng là quản lý.

- Thêm thông tin các loại sản phẩm : Khi có sản phẩm mới admin sẽ thêm mới thông tin về sản phẩm đó vào danh sách các sản phẩm.
- Chính sửa thông tin các sản phẩm : Khi có sự thay đổi về thông tin sản phẩm, người quản lý có thể thay đổi thông tin để người dùng cập nhật kịp thời.
- Xóa sản phẩm : Có thể vì một lý do nào đó mà một mặt hàng bị dừng bán và buộc phải xóa khỏi hệ thống, người quản lý sẽ xóa được sản phẩm đó.
- Xóa thông tin các thành viên : Khi tài khoản vì một lý do nào đó phải xóa, người quản lý có thể xóa toàn bộ thông tin liên quan đến tài khoản đó.
- Viết mô tả sản phẩm : Khi ra mắt sản phẩm mới, người quản lý có thể viết bài mô tả cho sản phẩm đó, điều này giúp thu hút khách hàng hơn.
- Trả lời câu hỏi : khi khách hàng đặt câu hỏi mua một sản phẩm, người quản lý có thể trả lời câu hỏi đó.

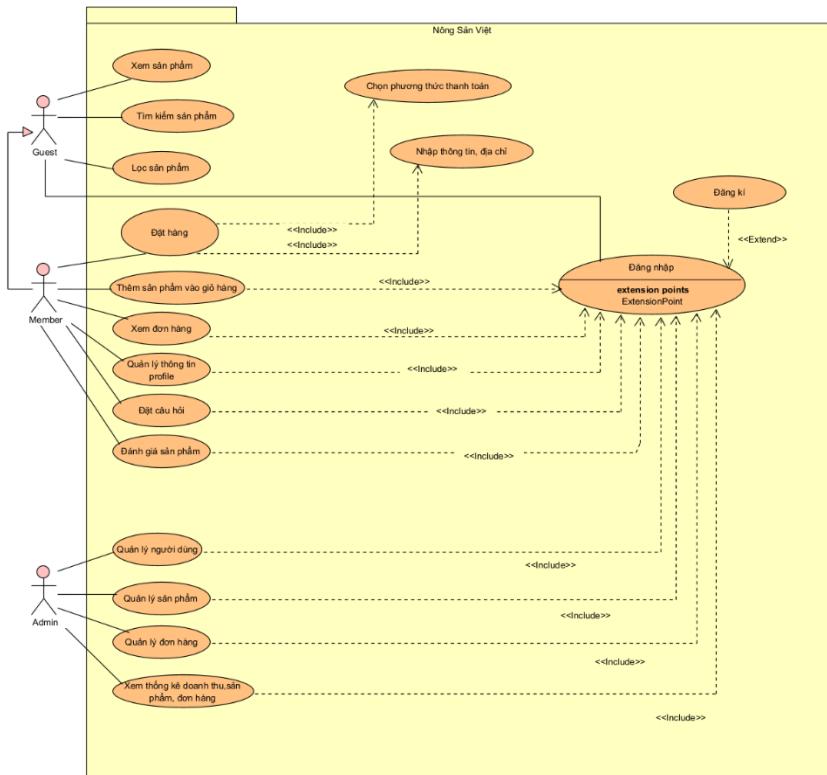
3.3. Sơ đồ use-case

3.3.1. Danh sách Actor.

Bảng 3.1. Danh sách actor

STT	Tên actor	Ý nghĩa
1	Guest (khách chưa đăng kí thành viên)	Chỉ có thể xem sản phẩm và chat với Admin.
2	Member (khách đã đăng kí thành viên)	Khách đã đăng kí thành viên được phép đăng nhập vào hệ thống. Thực hiện các hành động như: thêm sản phẩm vào giỏ hàng, đánh giá mặt hàng đã mua, quản lý thông tin cá nhân và xem đơn hàng.
3	Admin (Người quản lý)	Quản lý hệ thống.

3.3.2. Sơ đồ Use case



Hình 3.1. Sơ đồ Use case

3.3.2.1. Bảng chú thích

Bảng 3.2. Bảng chú thích sơ đồ Use case

Mã Use Case	Tên Use Case	Mô Tả
UC01	Đăng nhập	Người dùng đăng nhập vào hệ thống khi có tài khoản
UC02	Đăng kí	Đăng kí tài khoản người dùng
UC03	Xem sản phẩm	Người dùng xem danh sách sản phẩm
UC04	Tìm kiếm sản phẩm	Người dùng tìm kiếm sản phẩm theo từ khóa hoặc tiêu chí khác

UC05	Lọc sản phẩm	Người dùng học sản phẩm theo các thuộc tính (loại sản phẩm, giá cả,...)
UC07	Đặt hàng	Người dùng thực hiện đặt hàng sản phẩm
UC08	Chọn phương thức thanh toán	Người dùng lựa chọn phương thức thanh toán khi thanh toán đơn hàng.
UC09	Nhập thông tin, địa chỉ	Người dùng cần nhập thông tin và địa chỉ để tiến hành thanh toán
UC10	Thêm sản phẩm vào giỏ hàng	Người dùng thêm sản phẩm vào giỏ hàng
UC11	Xem đơn hàng	Người dùng có thể xem danh sách đơn hàng đã đặt
UC12	Quản lý thông tin profile	Người dùng chỉnh sửa thông tin cá nhân trong profile
UC13	Đặt câu hỏi	Người dùng có thể đặt câu hỏi về sản phẩm hoặc dịch vụ
UC14	Đánh giá sản phẩm	Người dùng có thể đánh giá sản phẩm sau khi mua sản phẩm
UC15	Quản lý người dùng	Admin quản lý thông tin người dùng trong hệ thống
UC16	Quản lý sản phẩm	Admin quản lý sản phẩm, thêm, sửa, xóa, sản phẩm
UC17	Quản lý đơn hàng	Admin quản lý thông tin đơn hàng và trạng thái của chúng
UC18	Xem thống kê doanh thu, sản phẩm, đơn hàng	Admin xem báo cáo thống kê doanh thu sản phẩm và đơn hàng

3.3.2.2. *Đặc tả*

1. Use Case Đăng nhập (UC01)

- Mô tả vắn tắt: Cho phép người dùng đăng nhập vào hệ thống để thực hiện các chức năng theo quyền đã được cấp.
- Actor chính: Member, Admin.
- Actor phụ: Không có.
- Tiền điều kiện: Người dùng đã có tài khoản trong hệ thống.
- Hậu điều kiện: Người dùng được phép thực hiện các chức năng trong phạm vi và quyền hạn cho phép sau khi đăng nhập thành công.
- Luồng hoạt động:
 1. Người dùng truy cập vào website .
 2. Người dùng chọn “Đăng nhập”.
 3. Hệ thống hiển thị form đăng nhập.
 4. Người dùng nhập email, mật khẩu và nhấn “Đăng nhập”.
 5. Hệ thống kiểm tra thông tin đăng nhập của người dùng.
 6. Hệ thống ghi nhận đăng nhập thành công.
- Luồng thay thế: Không có
- Luồng ngoại lệ: Thông tin đăng nhập không hợp lệ, hệ thống thông báo lỗi và yêu cầu đăng nhập lại.

2. Use Case Đăng ký (UC02)

- Mô tả vắn tắt: Cho phép người dùng đăng ký tài khoản để mua sản phẩm.
- Actor chính: Guest.
- Actor phụ: Không có.
- Tiền điều kiện: Người dùng chưa có tài khoản trong hệ thống.
- Hậu điều kiện: Người dùng có tài khoản và có thể đăng nhập vào hệ thống.
- Luồng hoạt động:
 1. Người dùng truy cập vào trang, chọn mục “đăng ký”.

2. Người dùng nhập thông tin tài khoản đăng ký (Tên, email, mật khẩu).
 3. Người dùng chọn “Đăng ký”.
 4. Hệ thống xác nhận và tạo tài khoản mới.
- Luồng thay thế: Không có.
 - Luồng ngoại lệ: Thông tin đăng ký không hợp lệ, hệ thống thông báo lỗi và yêu cầu đăng ký lại.
3. Use Case Xem sản phẩm (UC03)
- Mô tả văn tắt: Người dùng có thể xem các sản phẩm có sẵn trên website.
 - Actor chính: Guest, Member.
 - Actor phụ: Không có.
 - Tiền điều kiện: Không có.
 - Hậu điều kiện: Hệ thống hiển thị các sản phẩm.
 - Luồng hoạt động:
 1. Người dùng truy cập vào trang sản phẩm.
 2. Hệ thống hiển thị các danh sách sản phẩm nhập khẩu.
 3. Người dùng có thể lọc, tìm kiếm và xem chi tiết sản phẩm.
 - Luồng thay thế: Không có.
 - Luồng ngoại lệ: Nếu không có sản phẩm hệ thống hiển thị “Không tìm thấy sản phẩm phù hợp”.
4. Use Case Tìm kiếm sản phẩm (UC04)
- Mô tả văn tắt: Người dùng có thể tìm kiếm sản phẩm bằng từ khóa.
 - Actor chính: Guest, Member.
 - Actor phụ: Không có.
 - Tiền điều kiện: Không có.
 - Hậu điều kiện: Hệ thống hiển thị kết quả tìm kiếm phù hợp.
 - Luồng hoạt động:

1. Người dùng nhập từ khóa (tên sản phẩm có dấu hoặc không dấu)
 2. Hệ thống hiển thị kết quả tìm kiếm.
 3. Người dùng có thể chọn sản phẩm và xem chi tiết.
- Luồng thay thế: Không có.
 - Luồng ngoại lệ: Nếu không có sản phẩm thỏa mãn, hệ thống hiển thị thông báo “Không tìm thấy sản phẩm phù hợp”.

5.Use Case Lọc sản phẩm (UC05)

- Mô tả văn tắt: Người dùng có thể lọc sản phẩm theo các tiêu chí như giá tiền, các loại sản phẩm .
- Actor chính: Guest, Member.
- Actor phụ: Không có.
- Tiền điều kiện: Không có.
- Hậu điều kiện: Hệ thống hiển thị sản phẩm theo bộ lọc được chọn.
- Luồng hoạt động:
 1. Người dùng chọn bộ lọc như giá, loại sản phẩm.
 2. Hệ thống cập nhật danh sách sản phẩm theo các tiêu chí lọc.
- Luồng thay thế: Không có.
- Luồng ngoại lệ: Nếu không có sản phẩm thỏa mãn theo bộ lọc, hệ thống hiển thị thông báo “Không tìm thấy sản phẩm phù hợp”.

7.Use Case Đặt hàng (UC07)

- Mô tả văn tắt: Người dùng có thể thêm sản phẩm vào giỏ hàng và thực hiện đặt hàng .
- Actor chính: Member.
- Actor phụ: Không có.
- Tiền điều kiện: Người dùng đã đăng nhập và có giỏ hàng với các sản phẩm.

- Hậu điều kiện: Đơn hàng được tạo và hệ thống hiển thị xác nhận đơn hàng.
- Luồng hoạt động:
 1. Người dùng chọn sản phẩm và thêm vào giỏ hàng.
 2. Người dùng vào giỏ hàng và chọn “Đặt hàng”.
 3. Hệ thống hiển thị form nhập thông tin giao hàng.
 4. Người dùng nhập thông tin và chọn phương thức thanh toán.
 5. Hệ thống xác nhận đơn hàng thành công.
- Luồng thay thế: Không có.
- Luồng ngoại lệ: Nếu thông tin giao hàng không hợp lệ, hệ thống báo lỗi và yêu cầu nhập lại.

8. Use Case Chọn phương thức thanh toán (UC8)

- Mô tả vấn đề: Người dùng có thể chọn phương thức thanh toán phù hợp khi đặt hàng bao gồm các phương thức như thanh toán qua thẻ tín dụng, chuyển khoản, hoặc thanh toán khi nhận hàng.
- Actor chính: Member.
- Actor phụ: Không có.
- Tiền điều kiện:
 1. Người dùng đã đăng nhập vào hệ thống.
 2. Người dùng đã thêm ít nhất một sản phẩm vào giỏ hàng.
- Hậu điều kiện: Phương thức thanh toán được xác nhận và chuyển sang bước tiếp theo của quá trình đặt hàng.
- Luồng hoạt động:
 1. Người dùng truy cập vào giỏ hàng và nhấn “Đặt hàng”.
 2. Hệ thống hiển thị trang thông tin cần người dùng nhập để tiến hành thanh toán .
 3. Người dùng chọn phương thức thanh toán mong muốn .
 4. Hệ thống hiển thị trang để người dùng nhập thông tin cần thiết (số thẻ, thông tin ngân hàng, v.v. Tùy vào phương thức thanh toán).

5. Người dùng nhập đầy đủ thông tin và nhấn nút “Xác nhận”.
 6. Hệ thống kiểm tra xác nhận thông tin thanh toán thành công, sau đó chuyển sang bước xác nhận đơn hàng.
 7. Hệ thống thông báo đơn hàng đã được đặt thành công.
- Luồng thay thế: Không có
 - Luồng ngoại lệ:
 1. Nếu thông tin thanh toán không hợp lệ (sai số thẻ, số dư tài khoản không đủ), hệ thống thông báo lỗi yêu cầu người dùng kiểm tra lại thông tin.
 2. Nếu người dùng chọn phương thức thanh toán khi nhận hàng, hệ thống bỏ qua bước nhập thông tin thanh toán mà chỉ yêu cầu nhập thông tin cá nhân và địa chỉ.
9. Use Case Nhập thông tin, địa chỉ (UC09)
- Mô tả vấn đề: Người dùng nhập các thông tin cá nhân trước khi tiến hành chọn phương thức thanh toán.
 - Actor chính: Member.
 - Actor phụ: Không có
 - Tiên điều kiện: Người dùng đã có giỏ hàng và có ít nhất một sản phẩm trong giỏ.
 - Hậu điều kiện: Người dùng sang bước tiếp theo để thanh toán.
 - Luồng hoạt động:
 1. Người dùng vào giỏ hàng.
 2. Người nhấn “Đặt hàng” và tiến hành nhập thông tin, địa chỉ.
 3. Thông tin nhập thành công, người dùng tiến hành thanh toán.
 - Luồng thay thế: Không có.
 - Luồng ngoại lệ: Nếu người dùng nhập sai thông tin, hệ thống thông báo yêu cầu người dùng kiểm tra lại.
10. Use Case Thêm sản phẩm vào giỏ hàng (UC10)

- Mô tả văn tắt: Người dùng có thể quản lý các sản phẩm trong giỏ hàng, bao gồm thêm, xóa hoặc thay đổi số lượng sản phẩm.
- Actor chính: Member.
- Actor phụ: Không có.
- Tiền điều kiện: Người dùng đã có giỏ hàng.
- Hậu điều kiện: Giỏ hàng được cập nhật thành công.
- Luồng hoạt động:
 - 1.Người dùng vào giỏ hàng.
 - 2.Người dùng thêm, xóa, hoặc thay đổi số lượng sản phẩm.
 - 3.Hệ thống cập nhật giỏ hàng.
- Luồng thay thế: Không có.
- Luồng ngoại lệ: Nếu giỏ hàng trống, hệ thống không hiển thị sản phẩm.

11.Use Case Xem đơn hàng (UC11)

- Mô tả văn tắt: Người dùng có thể kiểm tra tình trạng đơn hàng đã đặt, từ khi tạo đến khi giao hàng hoàn tất.
- Actor chính: Member.
- Actor phụ: Không có.
- Tiền điều kiện: Người dùng đã đăng nhập và có ít nhất một đơn hàng.
- Hậu điều kiện: Người dùng có thể theo dõi, kiểm tra tình trạng đơn hàng.
- Luồng hoạt động:
 1. Người dùng vào trang “Đơn hàng”.
 2. Hệ thống hiển thị danh sách các đơn hàng với tình trạng hiện tại (tất cả, chờ xử lý, đang giao, đã giao).
- Luồng thay thế: Không có.
- Luồng ngoại lệ: Nếu không có đơn hàng, hệ thống hiển thị “Bạn không có đơn hàng nào”.

12. Use Case Quản lý thông tin profile (UC12)

- Mô tả văn tắt: Người dùng có thể xem và chỉnh sửa thông tin cá nhân như tên, địa chỉ email, số điện thoại, địa chỉ.
- Actor chính: Member.
- Actor phụ: Không có.
- Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống.
- Hậu điều kiện: Thông tin cá nhân của người dùng được cập nhật thành công.
- Luồng hoạt động:
 1. Người dùng truy cập vào trang “Trang cá nhân”.
 2. Người dùng chỉnh sửa như tên, địa chỉ email, số điện thoại, địa chỉ.
 3. Hệ thống lưu thông tin và hiển thị thông báo “Cập nhật thành công”.
- Luồng thay thế: Không có.
- Luồng ngoại lệ: Nếu thông tin nhập không hợp lệ, hệ thống thông báo lỗi và yêu cầu nhập lại.

13. Use Case Đặt câu hỏi (UC13)

- Mô tả văn tắt: Người dùng gửi câu hỏi liên quan đến sản phẩm.
- Actor chính: Member.
- Actor phụ: Không có.
- Tiền điều kiện: Người dùng đã đăng nhập vào hệ thống.
- Hậu điều kiện:
 1. Câu hỏi người dùng được ghi nhận và hiển thị công khai.
 2. Admin trả lời câu hỏi của người dùng.
- Luồng hoạt động:
 1. Người dùng sau khi đăng nhập, tìm sản phẩm muốn đặt câu hỏi.
 2. Người dùng ghi câu hỏi và nhấn “Gửi”.
 3. Hệ thống xác nhận và hiển thị công khai.

4. Admin trả lời câu hỏi của người dùng.
- Luồng thay thế: Không có.
 - Luồng ngoại lệ: Không có.

14. Use Case Đánh giá sản phẩm (UC14)

- Mô tả văn tắt: Người dùng có thể đánh giá sản phẩm mà họ đã mua.
- Actor chính: Member.
- Actor phụ: Không có.
- Tiền điều kiện: Người dùng đã mua sản phẩm và nhận hàng.
- Hậu điều kiện: Đánh giá sản phẩm được lưu vào hệ thống và hiển thị công khai ..
- Luồng hoạt động:
 8. Người dùng truy cập vào trang sản phẩm đã mua.
 9. Người dùng chọn “Gửi đánh giá”.
 10. Người dùng chọn thang sao và ghi nhận xét cho sản phẩm.
 11. Hệ thống lưu lại đánh giá và hiển thị công khai trên trang sản phẩm.
- Luồng thay thế: Không có.
- Luồng ngoại lệ: Không có.

15. Use Case Quản lý người dùng (UC15)

- Mô tả văn tắt: Admin có thể quản lý thông tin người dùng, bao gồm xem thông tin, chỉnh sửa quyền truy cập, hoặc xóa tài khoản người dùng.
- Actor chính: Admin.
- Actor phụ: Không có.
- Tiền điều kiện: Admin đã đăng nhập vào hệ thống.
- Hậu điều kiện: Thông tin người dùng được cập nhật hoặc xóa thành công .
- Luồng hoạt động:
 1. Admin vào mục “Customer” ở trang “Admin”.

- 2. Admin có thể xem thông tin người dùng.
- 3. Admin có thể xóa tài khoản người dùng.
- 4. Hệ thống cập nhật hoặc xóa tài khoản theo yêu cầu Admin.
- Luồng thay thế: Không có.
- Luồng ngoại lệ: Không có tài khoản người dùng, hệ thống không hiển thị.

16. Use Case Quản lý sản phẩm (UC16)

- Mô tả văn tắt: Admin có thể quản lý sản phẩm trong hệ thống, bao gồm thêm mới, sửa hoặc xóa sản phẩm.
- Actor chính: Admin.
- Actor phụ: Không có.
- Tiền điều kiện: Admin đã đăng nhập vào hệ thống.
- Hậu điều kiện: Sản phẩm được cập nhật trong hệ thống.
- Luồng hoạt động:
 1. Admin vào trang “Admin”.
 2. Admin chọn “Create product” hoặc chọn sửa/xóa sản phẩm hiện có.
 3. Admin nhập thông tin sản phẩm (tên, số lượng, giá, giá giảm, hình ảnh).
 4. Hệ thống lưu thông tin sản phẩm và cập nhật trên website.
- Luồng thay thế: Không có.
- Luồng ngoại lệ: Nếu không có sản phẩm hợp lệ, hệ thống báo lỗi và yêu cầu chỉnh sửa.

17. Use Case Quản lý đơn hàng (UC17)

- Mô tả văn tắt: Admin có thể xem, cập nhật trạng thái và quản lý toàn bộ đơn hàng trong hệ thống.
- Actor chính: Admin.
- Actor phụ: Không có.

- Tiền điều kiện:
 1. Admin đã đăng nhập vào hệ thống.
 2. Đã ít nhất một đơn hàng đã tạo trong hệ thống.
- Hậu điều kiện: Admin quản lý đơn hàng “Xác nhận đơn hàng” và xác nhận “Đã giao hàng thành công”.
- Luồng hoạt động:
 1. Admin đăng nhập vào hệ thống .
 2. Admin vào mục “Order”.
 3. Hệ thống hiển thị danh sách thông tin toàn bộ những đơn hàng đã được đặt (gồm số lượng, giá, sản phẩm, số tiền, thời gian đặt hàng, trạng thái đơn hàng).
 4. Admin chọn đơn hàng cụ thể và xem chi tiết.
 5. Admin thay đổi trạng thái đơn hàng (ví dụ “Đang xử lý”, “Đã giao”).
 6. Hệ thống lưu lại thay đổi.
- Luồng thay thế: Không có.
- Luồng ngoại lệ: Nếu hệ thống gặp sự cố khi cập nhật trạng thái đơn hàng, Admin nhận được thông báo lỗi và yêu cầu thử lại.

18. Use Case Thống kê doanh thu (UC18)

- Mô tả vắn tắt: Admin có thể xem thống kê về doanh thu, sản phẩm bán chạy và các đơn hàng trong hệ thống.
- Actor chính: Admin.
- Actor phụ: Không có.
- Tiền điều kiện: Admin đã đăng nhập vào hệ thống.
- Hậu điều kiện: Thống kê doanh thu và các thông tin liên quan được hiển thị.
- Luồng hoạt động:
 1. Admin vào “Dashboard”.
 2. Hệ thống hiển thị báo cáo về doanh thu, số lượng sản phẩm, số lượng đơn hàng.

3. Admin có thể lọc theo ngày, tháng, năm.
- Luồng thay thế: Không có.
 - Luồng ngoại lệ: Nếu không có dữ liệu thống kê, hệ thống hiển thị “Không có dữ liệu.”

3.4. Sơ đồ lớp

<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;">users</th></tr> </thead> <tbody> <tr> <td style="padding: 5px;"> -_id : ObjectId -address : String -createdAt : Date -email : String -isAdmin : Bool -name : String -password : String -phone : String -updatedAt : Date </td></tr> <tr> <td style="padding: 5px;"> +register() +login() +logout() +refreshToken() +updateUser() +getUser() +deleteUser() </td></tr> </tbody> </table>	users	-_id : ObjectId -address : String -createdAt : Date -email : String -isAdmin : Bool -name : String -password : String -phone : String -updatedAt : Date	+register() +login() +logout() +refreshToken() +updateUser() +getUser() +deleteUser()	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;">orders</th></tr> </thead> <tbody> <tr> <td style="padding: 5px;"> -_id : ObjectId -cancelOrder : Bool -createdAt : Date -name : String -orderItems : Array -order_code : String -paymentMethod : String -paymentResult : ObjectId -shippingAddress : ObjectId -status : String -to_district_id : Int32 -to_ward_code : String -totalPrice : Int32 -updatedAt : Date -user : ObjectId </td></tr> <tr> <td style="padding: 5px;"> +createOrder() +cancelOrder() +updateOrderStatus() +getOrderDetails() +changeShippingAddress() +getOrders() </td></tr> </tbody> </table>	orders	-_id : ObjectId -cancelOrder : Bool -createdAt : Date -name : String -orderItems : Array -order_code : String -paymentMethod : String -paymentResult : ObjectId -shippingAddress : ObjectId -status : String -to_district_id : Int32 -to_ward_code : String -totalPrice : Int32 -updatedAt : Date -user : ObjectId	+createOrder() +cancelOrder() +updateOrderStatus() +getOrderDetails() +changeShippingAddress() +getOrders()	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;">products</th></tr> </thead> <tbody> <tr> <td style="padding: 5px;"> -_id : ObjectId -amount : Number -cloudinary_id : String -comments : Array -createdAt : Date -image : String -name : String -price : Int32 -rating : Int32 -reviews : Array -salePrice : Int32 -type : String -updatedAt : Date </td></tr> <tr> <td style="padding: 5px;"> +getProducts() +createProduct() +updateProduct() +deleteProduct() +changePublishStatus() </td></tr> </tbody> </table>	products	-_id : ObjectId -amount : Number -cloudinary_id : String -comments : Array -createdAt : Date -image : String -name : String -price : Int32 -rating : Int32 -reviews : Array -salePrice : Int32 -type : String -updatedAt : Date	+getProducts() +createProduct() +updateProduct() +deleteProduct() +changePublishStatus()
users											
-_id : ObjectId -address : String -createdAt : Date -email : String -isAdmin : Bool -name : String -password : String -phone : String -updatedAt : Date											
+register() +login() +logout() +refreshToken() +updateUser() +getUser() +deleteUser()											
orders											
-_id : ObjectId -cancelOrder : Bool -createdAt : Date -name : String -orderItems : Array -order_code : String -paymentMethod : String -paymentResult : ObjectId -shippingAddress : ObjectId -status : String -to_district_id : Int32 -to_ward_code : String -totalPrice : Int32 -updatedAt : Date -user : ObjectId											
+createOrder() +cancelOrder() +updateOrderStatus() +getOrderDetails() +changeShippingAddress() +getOrders()											
products											
-_id : ObjectId -amount : Number -cloudinary_id : String -comments : Array -createdAt : Date -image : String -name : String -price : Int32 -rating : Int32 -reviews : Array -salePrice : Int32 -type : String -updatedAt : Date											
+getProducts() +createProduct() +updateProduct() +deleteProduct() +changePublishStatus()											
listtypeproducts											
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;">listtypeproducts</th></tr> </thead> <tbody> <tr> <td style="padding: 5px;"> -_id : ObjectId -cloudinary_id : String -createAt : Date -img : String -name : String -updateAt : Date </td></tr> <tr> <td style="padding: 5px;"> +createTypeProduct() +getTypeProducts() +updateTypeProduct() +deleteTypeProduct() </td></tr> </tbody> </table>			listtypeproducts	-_id : ObjectId -cloudinary_id : String -createAt : Date -img : String -name : String -updateAt : Date	+createTypeProduct() +getTypeProducts() +updateTypeProduct() +deleteTypeProduct()						
listtypeproducts											
-_id : ObjectId -cloudinary_id : String -createAt : Date -img : String -name : String -updateAt : Date											
+createTypeProduct() +getTypeProducts() +updateTypeProduct() +deleteTypeProduct()											

Hình 3.2. Sơ đồ lớp

3.4.1. Users (*Người dùng*)

Bảng 3.3. Bảng Users

Thuộc tính	Mô tả
id	ID người dùng
address	Địa chỉ người dùng
createdAt	Ngày tạo tài khoản
email	Email người dùng
isAdmin	Người dùng là quản trị viên hay không (Boolean)

name	Tên người dùng
password	Mật khẩu
phone	Số điện thoại
updatedAt	Ngày cập nhật tài khoản
register()	Đăng ký tài khoản
login()	Đăng nhập
logout()	Đăng xuất
refreshToken()	Làm mới token
updateUser()	Cập nhật thông tin người dùng
ordersHistory	Xem lịch sử đơn hàng
deleteUser()	Xóa tài khoản người dùng

3.4.2. Order (đơn hàng)

Bảng 3.4. Bảng Order

Thuộc tính	Mô tả
id	ID đơn hàng
cancelOrder	Trạng thái hủy đơn hàng (Boolean)
createdAt	Ngày tạo đơn hàng
name	Tên người đặt hàng
orderItems	Danh sách sản phẩm trong đơn
Order-code	Mã đơn hàng
paymentMenthod	Phương thức thanh toán
paymentResult	Kết quả thanh toán
shippingAddress	Địa chỉ giao hàng
status	Trạng thái đơn hàng
to-district-id	ID quận giao hàng
to-ward-code	Mã phường giao hàng
totalPrice	Tổng giá trị đơn hàng
updatedAt	Ngày cập nhật đơn hàng
user	ID người dùng (ID của người đặt hàng)
createOrder()	Tạo đơn hàng mới
updateOrderStatus()	Cập nhật trạng thái đơn hàng
getOrderDetails()	Lấy chi tiết đơn hàng
changeShippingAddress	Thay đổi địa chỉ giao hàng
getOrders()	Lấy danh sách đơn hàng

3.4.3. Products (sản phẩm)

Bảng 3.5. Bảng Products

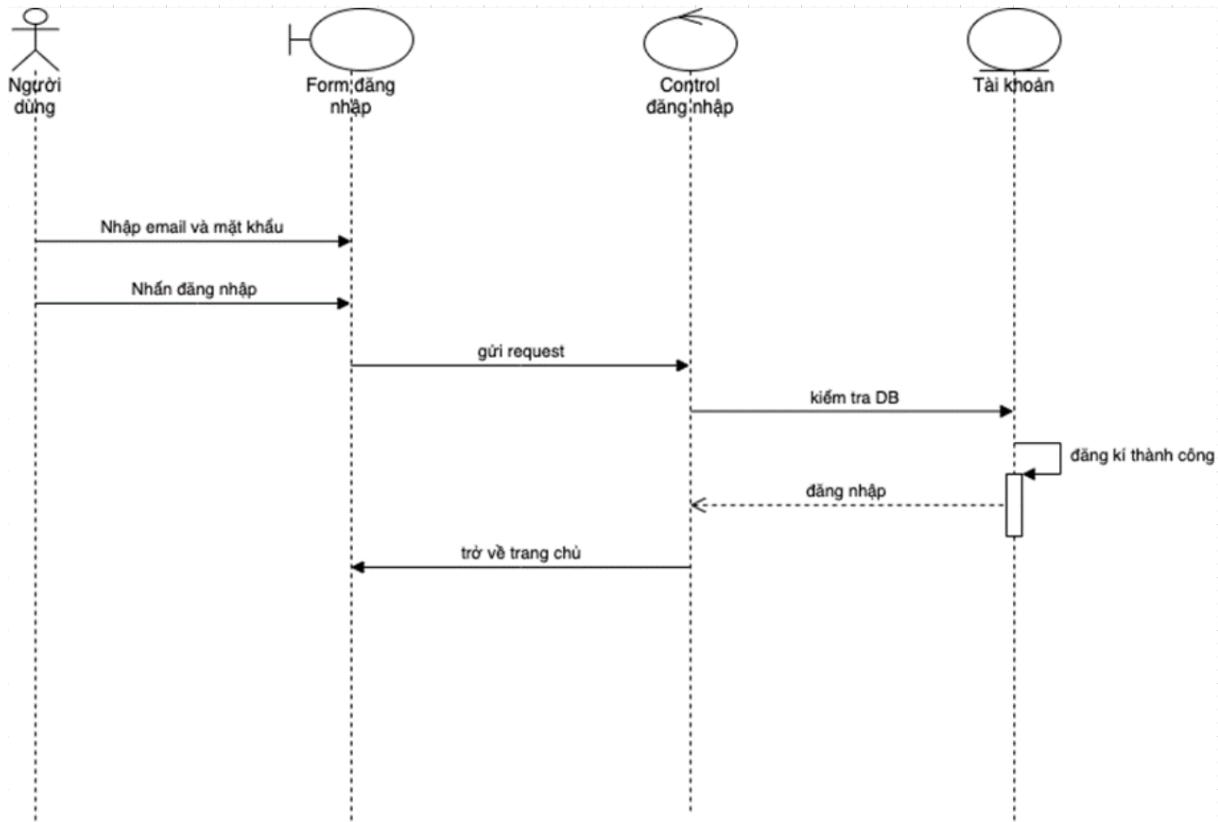
Thuộc tính	Mô tả
id	ID sản phẩm
amount	Số lượng sản phẩm
cloudinary-id	ID hình ảnh trên Cloudinary
comment	Các bình luận về sản phẩm
createAt	Ngày tạo sản phẩm
image	URL hình ảnh sản phẩm
name	Tên sản phẩm
price	Giá sản phẩm
rating	Xếp hạng sản phẩm (số sao)
reviews	Các đánh giá sản phẩm
salePrice	Giá khuyến mãi
Type	Loại sản phẩm
updatedAt	Ngày cập nhật sản phẩm
getProducts()	Lấy danh sách sản phẩm
createProduct()	Tạo sản phẩm mới
updateProduct()	Cập nhật thông tin sản phẩm
deleteProduct()	Xóa sản phẩm

3.4.4. Listtypeproducts (danh mục loại sản phẩm)**Bảng 3.6. Bảng Listtypeproducts**

Thuộc tính	Mô tả
Id	ID loại sản phẩm
cloudinary-id	ID hình ảnh sản phẩm
createAt	Ngày tạo loại sản phẩm
img	Hình ảnh loại sản phẩm
Name	Tên loại sản phẩm
updateAt	Ngày cập loại sản phẩm
getTypeProducts()	Lấy danh sách các loại sản phẩm
createTypeProduct()	Tạo loại sản phẩm mới
updateTypeProduct()	Cập nhật loại sản phẩm
deleteTypeProduct()	Xóa loại sản phẩm

3.5. Biểu đồ tuần tự của các chức năng trong hệ thống

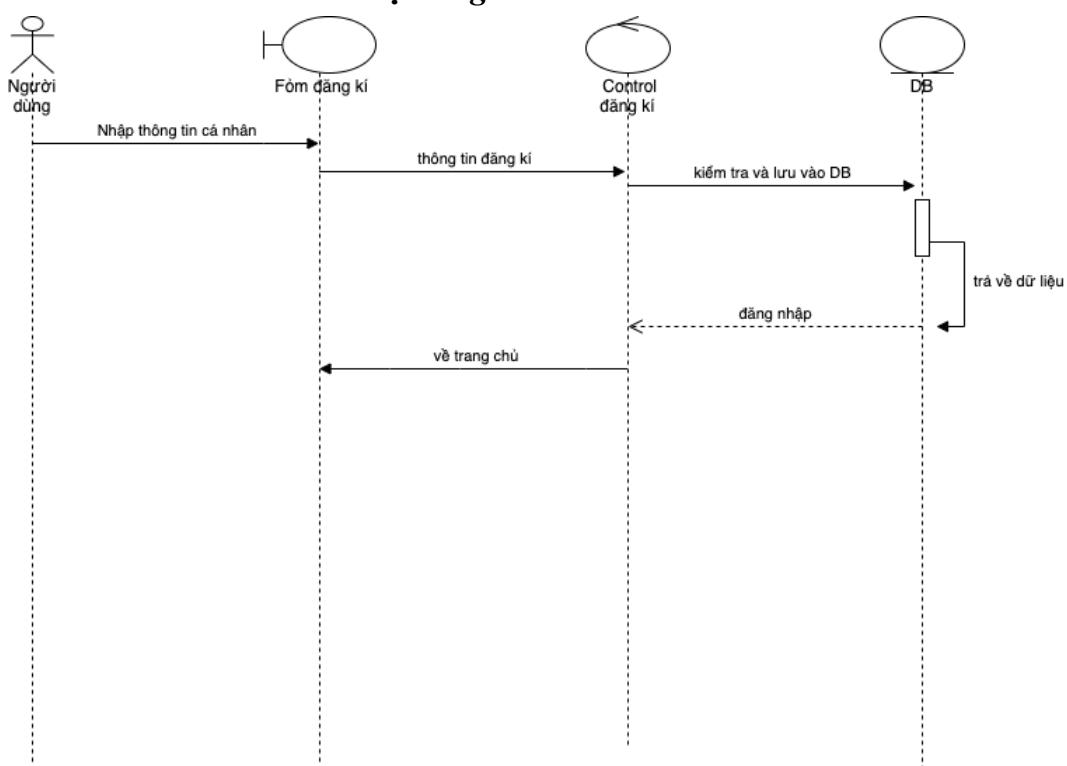
3.5.1. Biểu đồ tuần tự đăng nhập



1. **Người dùng** nhập email và mật khẩu vào **From đăng nhập**.
2. **Người dùng** nhấn nút đăng nhập, và **Form đăng nhập** gửi yêu cầu đến **Cotrol đăng nhập**.
3. **Cotrol đăng nhập** kiểm tra thông tin trong cơ sở dữ liệu (**DB**) thông qua đối tượng **Tài khoản**.
4. Nếu thông tin đúng, **Tài khoản** phản hồi thành công, và người dùng được đăng nhập.
5. Cuối cùng, **Form đăng nhập** trả người dùng về trang chủ.

Hình 3.3. Biểu đồ tuần tự đăng nhập

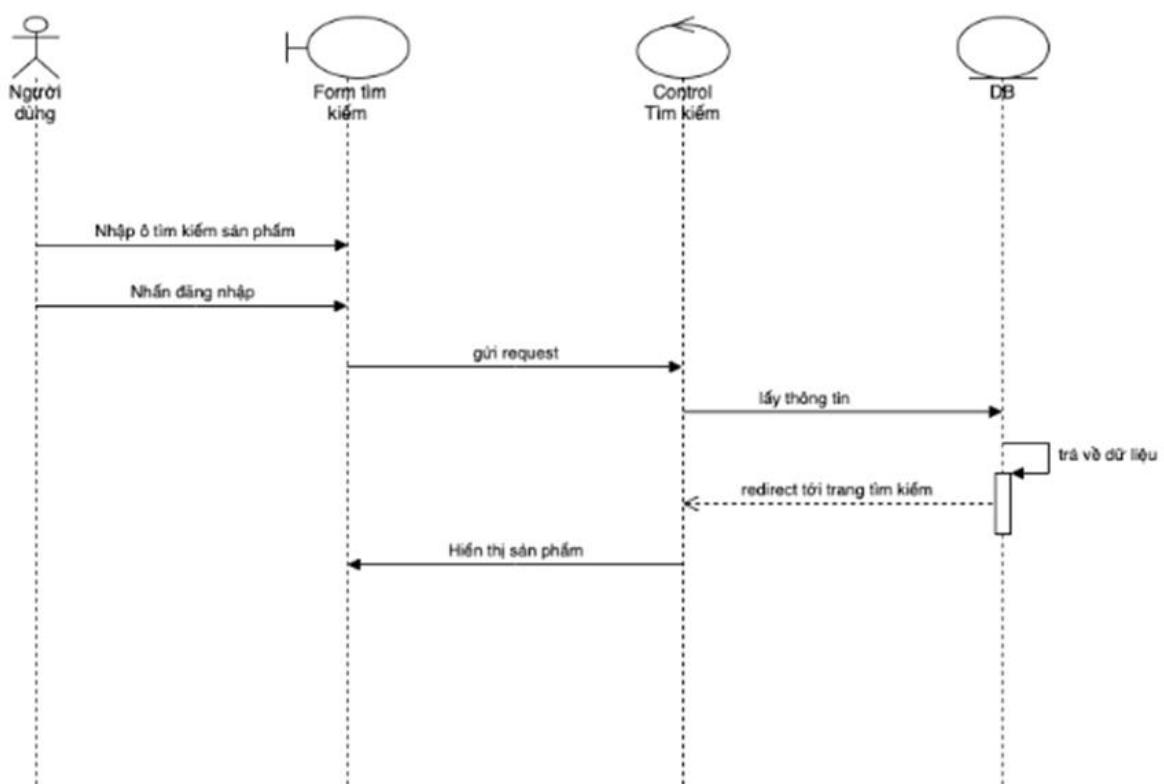
3.5.2. Biểu đồ tuần tự đăng ký thành viên



1. **Người dùng** nhập thông tin cá nhân vào **From đăng kí**.
2. Thông tin đăng kí được gửi đến **Control đăng kí**.
3. **Control đăng kí** kiểm tra và lưu thông tin vào cơ sở dữ liệu (**DB**).
4. Sau khi lưu thành công, cơ sở dữ liệu trả về thông báo xác nhận.
5. **Người dùng** sau đó có thể đăng nhập vào hệ thống.
6. Cuối cùng, người dùng được chuyển hướng về trang chủ.

Hình 3.4. Biểu đồ tuần tự đăng ký thành viên

3.5.3. Biểu đồ tuần tự tìm kiếm sản phẩm

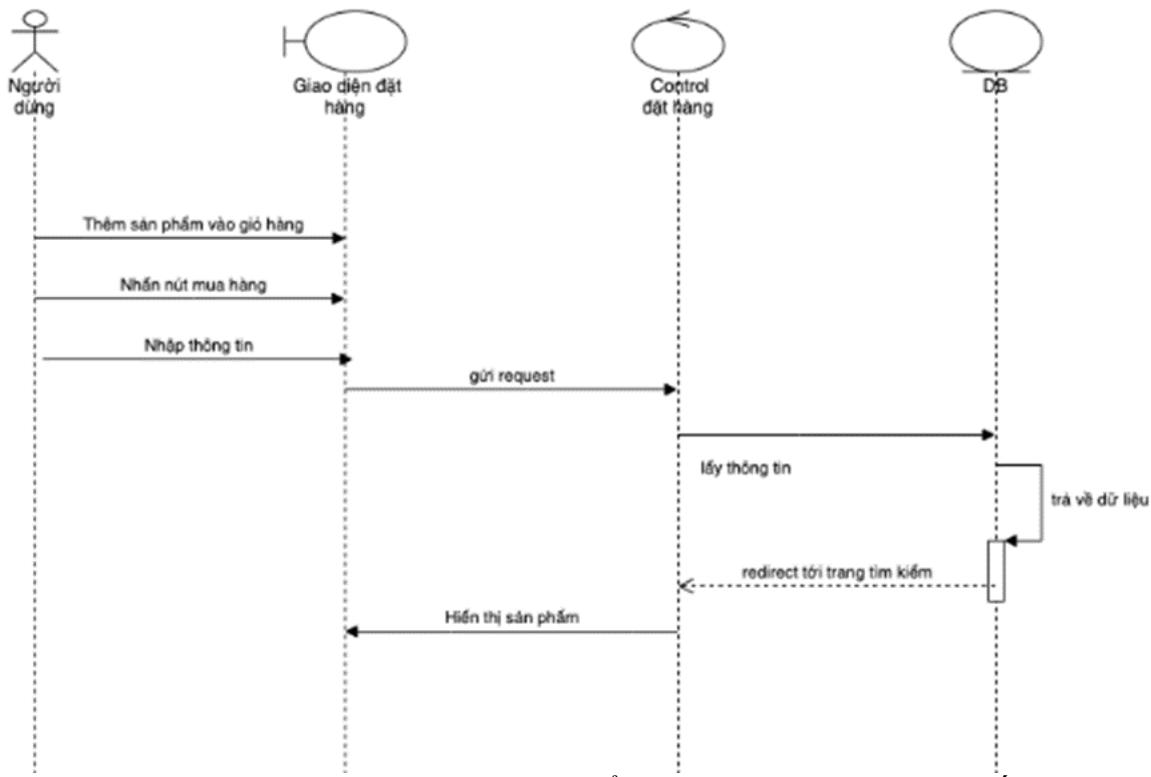


1. Người dùng nhập từ khóa vào ô tìm kiếm sản phẩm trong Form tìm kiếm.

2. Sau khi nhấn tìm kiếm, **Form tìm kiếm** gửi yêu cầu đến **Control tìm kiếm**.
3. **Control tìm kiếm** yêu cầu lấy thông tin từ cơ sở dữ liệu (**DB**).
4. Cơ sở dữ liệu trả về kết quả thông tin sản phẩm cho **Control tìm kiếm**.
5. **Control tìm kiếm** chuyển hướng người dùng đến trang kết quả tìm kiếm và hiển thị danh sách sản phẩm.

Hình 3.5. đồ tuần tự tìm kiếm sản phẩm

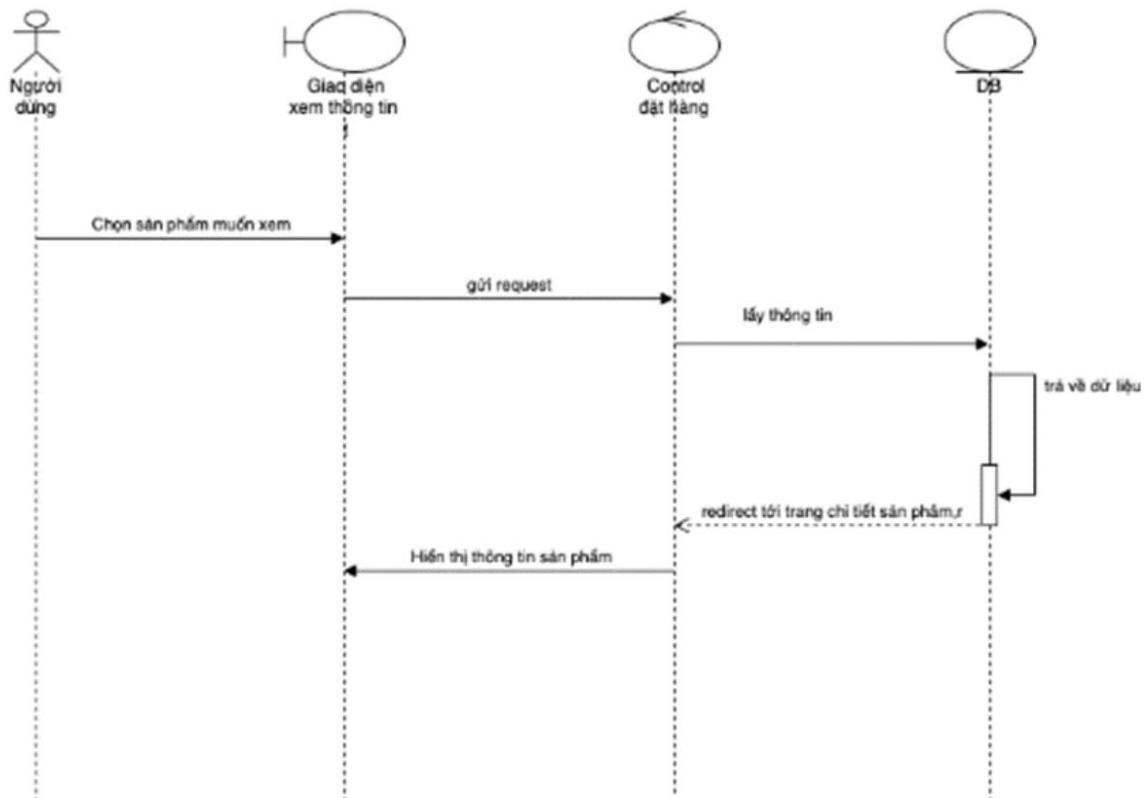
3.5.4. Biểu đồ tuần tự đặt hàng



1. **Người dùng** thêm sản phẩm vào giỏ hàng, sau đó nhấn nút mua hàng.
2. **Người dùng** nhập thông tin và gửi yêu cầu đến **Giao diện đặt hàng**.
3. Giao diện đặt hàng chuyển yêu cầu đến **Control đặt hàng**.
4. **Control đặt hàng** yêu cầu thông tin từ cơ sở dữ liệu (**DB**).
5. Cơ sở dữ liệu trả về dữ liệu sản phẩm.
6. **Control đặt hàng** sau đó chuyển hướng người dùng tới trang tìm kiếm và hiển thị sản phẩm.

Hình 3.6. Biểu đồ tuần tự đặt hàng

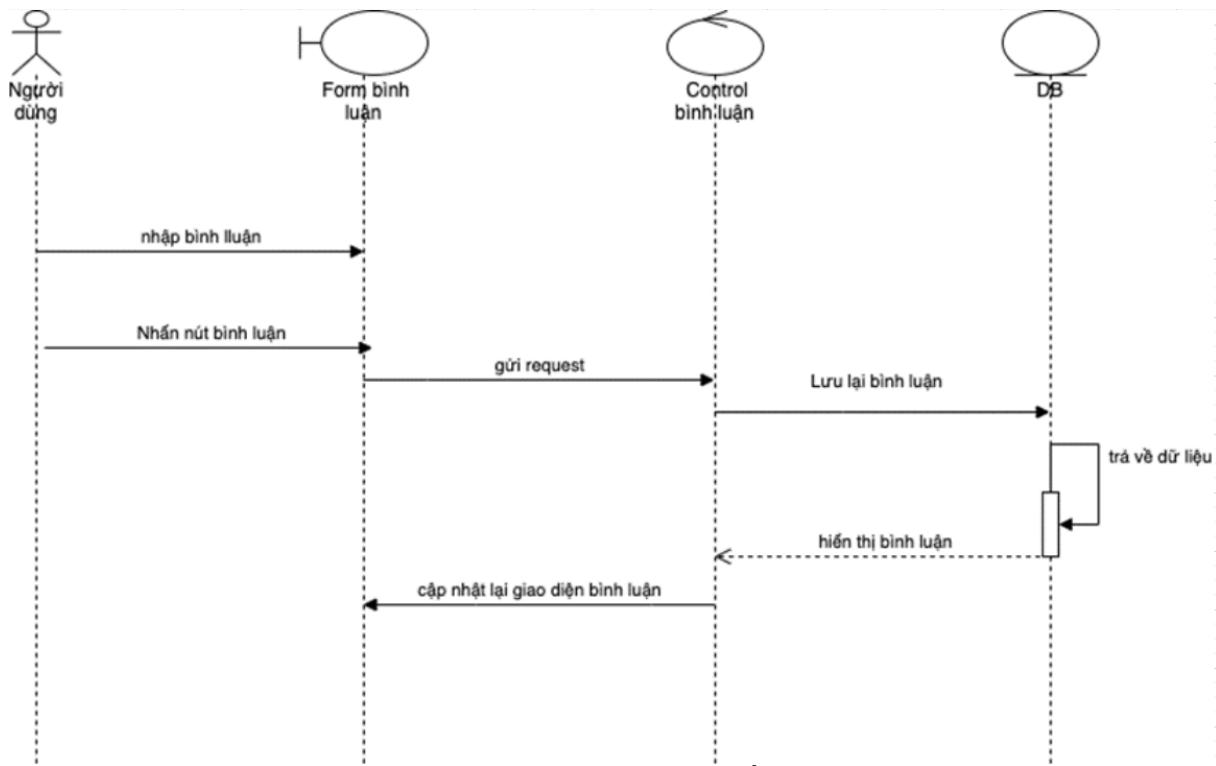
3.5.5. Biểu đồ tuần tự xem thông tin sản phẩm



1. **Người dùng** chọn sản phẩm muốn xem.
2. Giao diện xem thông tin nhận yêu cầu và gửi request **đến Control đặt hàng**.
3. **Control đặt hàng** lấy thông tin từ cơ sở dữ liệu (**DB**).
4. Cơ sở dữ liệu trả về dữ liệu sản phẩm.
5. **Control đặt hàng** chuyển hướng người dùng đến trang chi tiết sản phẩm và hiển thị thông tin sản phẩm.

Hình 3.7. Biểu đồ tuần tự xem thông tin sản phẩm

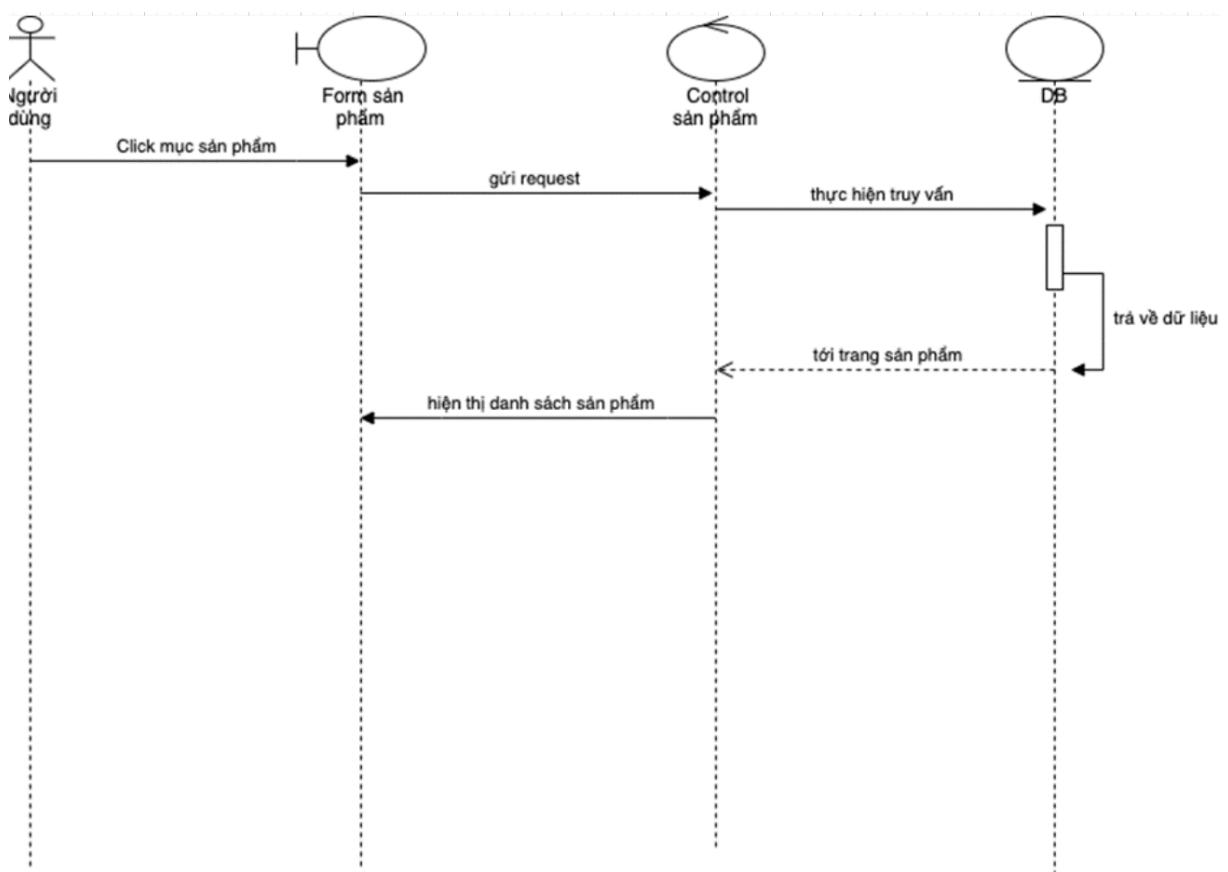
3.5.6. Biểu đồ tuần tự bình luận (gửi câu hỏi)



1. **Người dùng** nhập bình luận và nhấn nút gửi.
2. **Form bình luận** nhận bình luận và gửi request đến **Control bình luận**.
3. **Control bình luận** lưu bình luận vào cơ sở dữ liệu (**DB**).
4. Cơ sở dữ liệu trả về dữ liệu, xác nhận bình luận đã được lưu.
5. **Form bình luận** cập nhật lại giao diện để hiển thị bình luận mới.

Hình 3.8. Biểu đồ tuần tự bình luận (gửi câu hỏi)

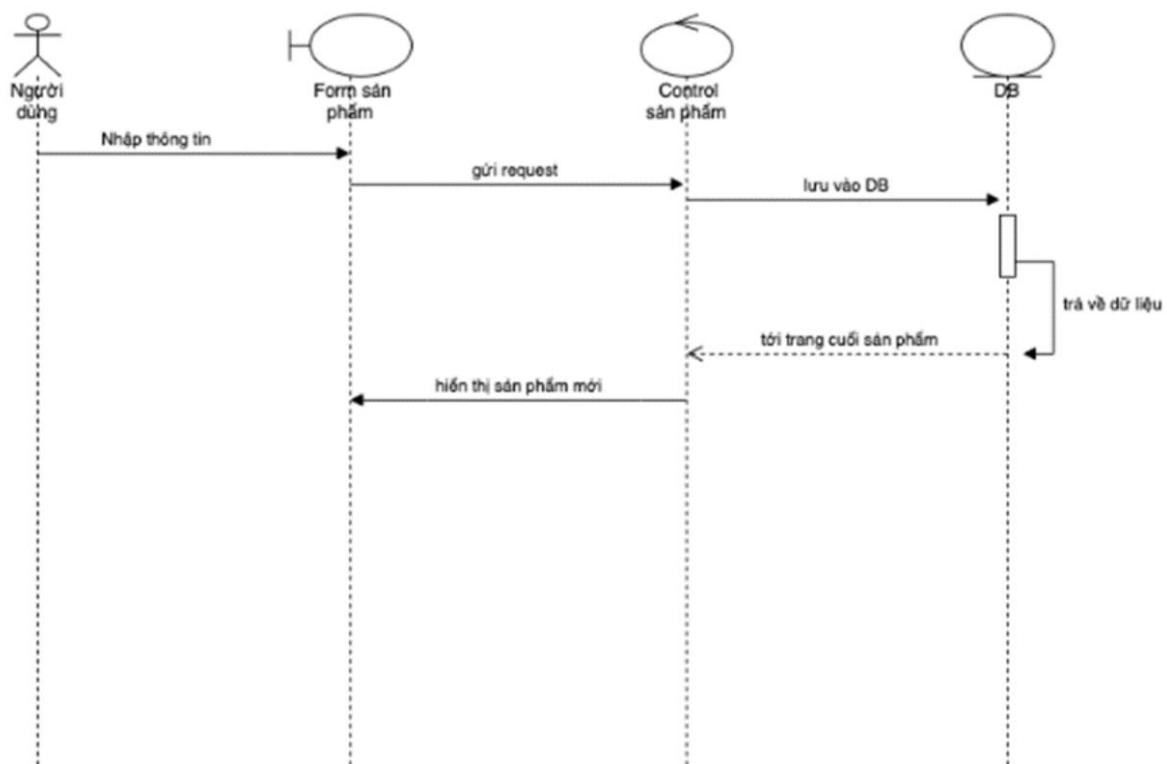
3.5.7. Biểu đồ tuần tự hiển thị danh sách sản phẩm



1. **Người dùng** click vào mục sản phẩm trên giao diện.
2. **From đăng nhập** gửi request đến **Control sản phẩm**.
3. **Control sản phẩm** thực hiện truy vấn dữ liệu từ cơ sở dữ liệu (**DB**).
4. Cơ sở dữ liệu trả lại dữ liệu sản phẩm cho **Control sản phẩm**.
5. **Control sản phẩm** gửi thông tin sản phẩm về **From sản phẩm**.
6. Cuối cùng, **From sản phẩm** hiển thị danh sách sản phẩm cho người dùng.

Hình 3.9. Biểu đồ tuần tự hiển thị danh sách sản phẩm

3.5.8. Biểu đồ tuần tự thêm sản phẩm

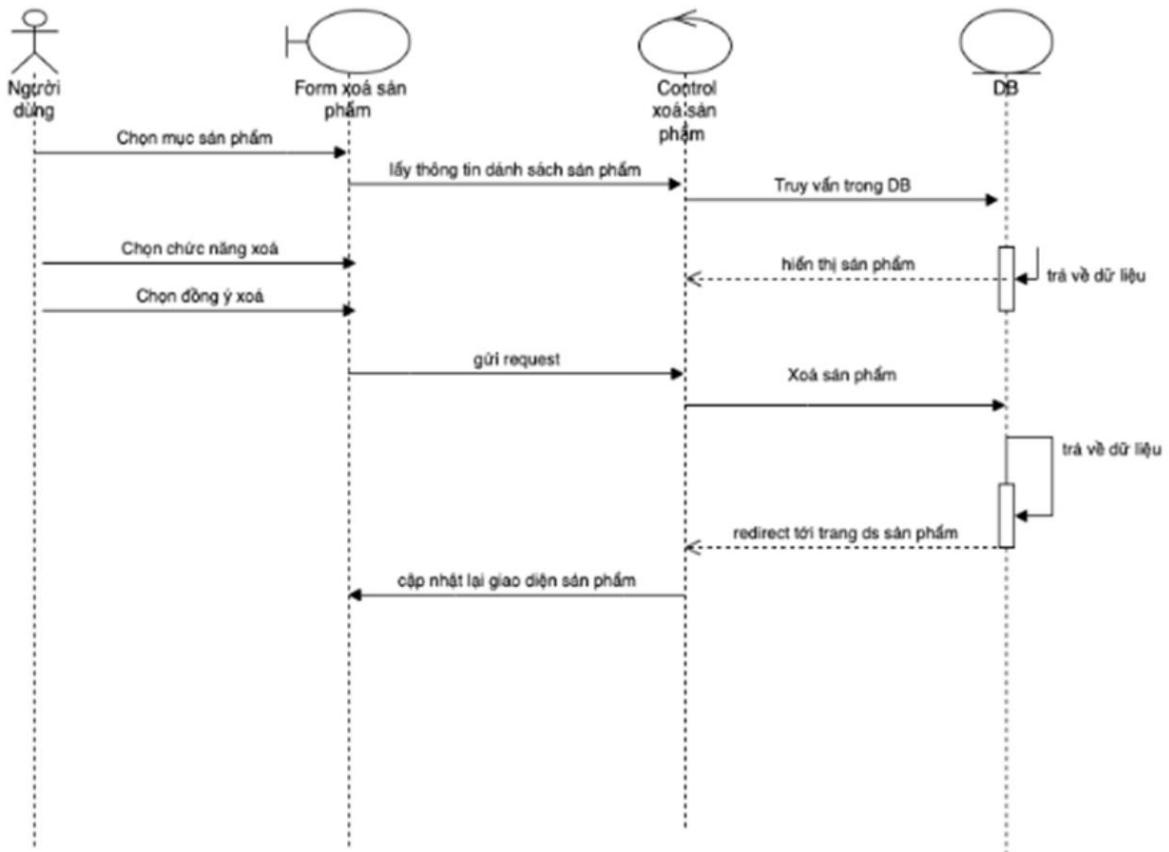


1. Người dùng (Admin) nhập thông tin sản phẩm trên dao diệu.
2. Form sản phẩm gửi request đến Control sản phẩm.

3. **Control sản phẩm** thực hiện lưu thông tin sản phẩm vào cơ sở dữ liệu (**DB**).
4. Cơ sở dữ liệu lại dữ liệu sau khi lưu thành công cho **Control sản phẩm**.
5. **Control sản phẩm** chuyển tiếp dữ liệu đến **From sản phẩm**.
6. **From sản phẩm** hiển thị sản phẩm mới cho người dùng.

Hình 3.10. Biểu đồ tuần tự thêm sản phẩm

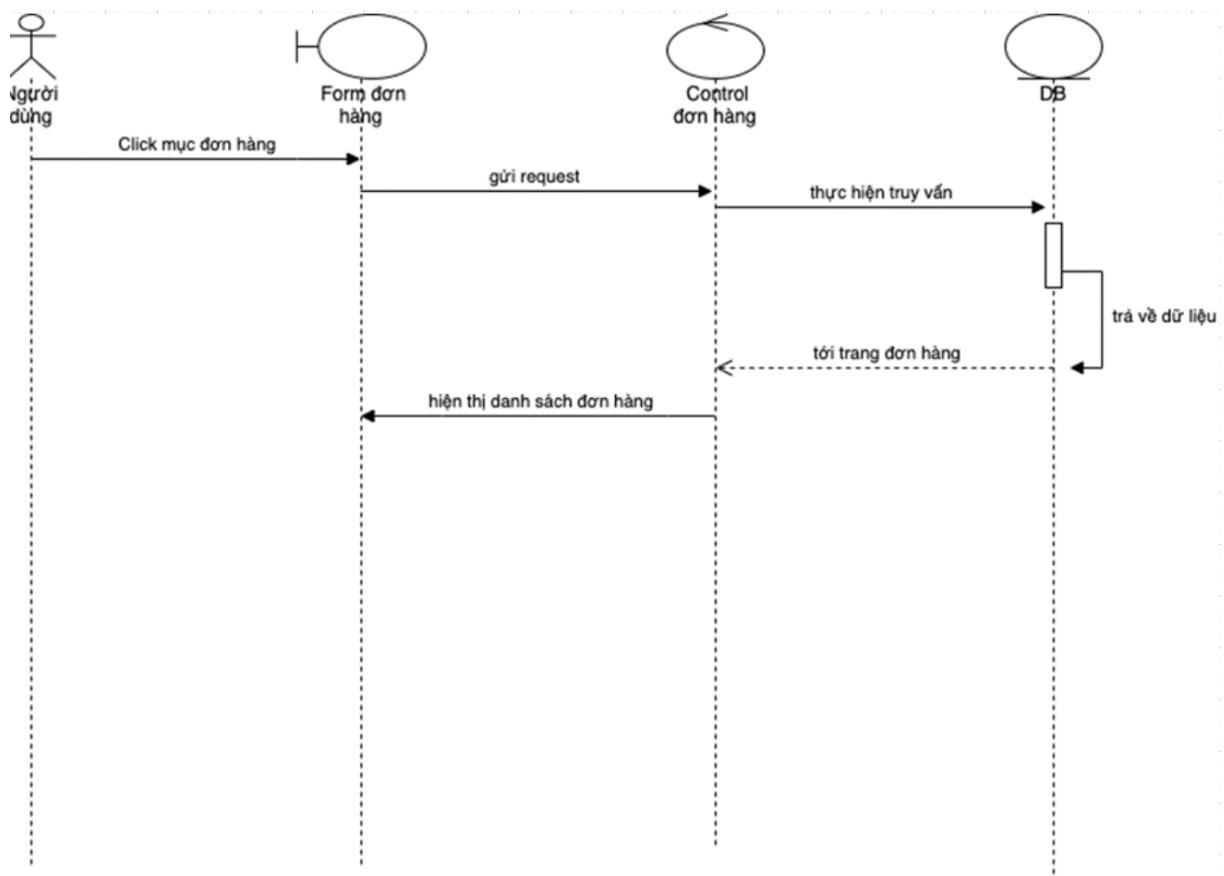
3.5.9. Biểu đồ tuần tự xóa sản phẩm



1. **Người dùng (Admin)** chọn mục sản phẩm.
2. **Form xóa sản phẩm** lấy thông tin danh sách sản phẩm từ **Control xóa sản phẩm**.
3. **Control xóa sản phẩm** truy vấn cơ sở dữ liệu (**DB**) để lấy danh sách sản phẩm và hiển thị danh sách đó.
4. **Người dùng** chọn chức năng xóa và chọn đồng ý để tiến hành.
5. **Form xóa sản phẩm** gửi request xóa sản phẩm đến **control xóa sản phẩm**.
6. **Control xóa sản phẩm** thực hiện xóa sản phẩm trong cơ sở dữ liệu (**DB**).
7. Cơ sở dữ liệu trả về dữ liệu xác nhận đã xóa sản phẩm.
8. **Control xóa sản phẩm** chuyển về trang danh sách sản phẩm.
9. Giao diện sản phẩm được cập nhật lại cho người dùng sau khi sản phẩm đã bị xóa.

Hình 3.11. Biểu đồ tuần tự xóa sản phẩm

3.5.10. Biểu đồ tuần tự xem đơn hàng

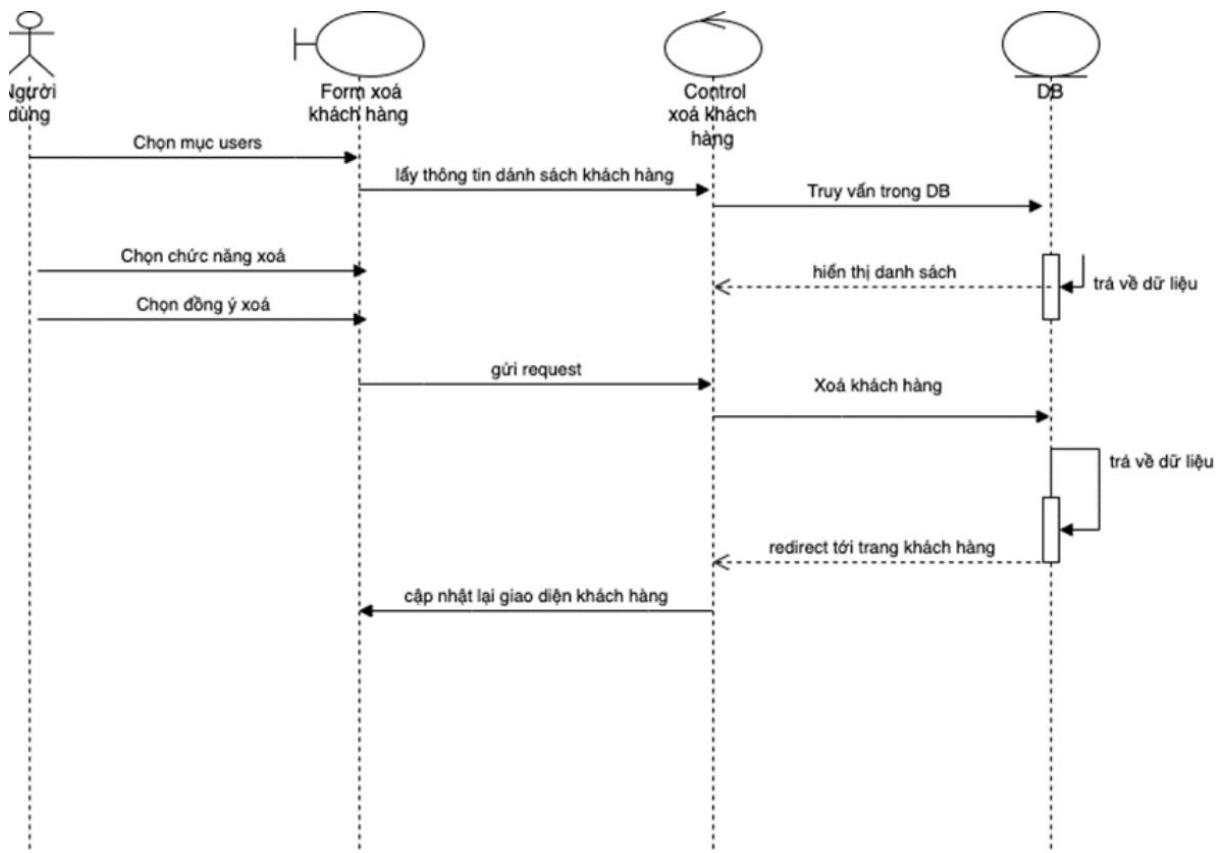


1. Người dùng click vào mục đơn hàng trên giao diện.

2. **Form đơn hàng** gửi yêu cầu đến phần **Control đơn hàng**.
3. **Control đơn hàng** thực hiện truy vấn dữ liệu từ cơ sở dữ liệu (**DB**)
4. Cơ sở dữ liệu trả về dữ liệu đơn hàng cho **Control đơn hàng**.
5. **Control đơn hàng** xử lý dữ liệu và chuyển đến trang đơn hàng.
6. Giao diện hiển thị danh sách đơn hàng cho người dùng.

Hình 3.12. Biểu đồ tuần tự xem đơn hàng

3.5.11. Biểu đồ tuần tự xóa thành viên



1. **Người dùng (Admin)** chọn mục users trên giao diện.
2. Hệ thống lấy thông tin danh sách khách hàng từ cơ sở dữ liệu (**DB**) và hiển thị
3. **Người dùng** chọn chức năng xóa và xác nhận xóa khách hàng.
4. Hệ thống gửi yêu cầu xóa đến phần kiểm soát.
5. Phần kiểm soát thực hiện xóa khách hàng trong cơ sở dữ liệu (**DB**)
6. Cơ sở dữ liệu trả về kết quả.
7. Hệ thống chuyển hướng về trang danh sách khách hàng.
8. Giao diện được cập nhật, hiển thị danh sách khách hàng mới.

Hình 3.13. Biểu đồ tuần tự xóa thành viên

CHƯƠNG 4: GIAO DIỆN ÚNG DỤNG

Chương này tiến hành đưa ra các hình ảnh hệ thống và phân tích một cách chi tiết, từ việc xác định các yêu cầu chức năng và phi chức năng đến việc thiết lập cấu trúc tổng thể cho trang web.

4.1. Giao diện ứng dụng

Giao diện ứng dụng đóng vai trò vô cùng quan trọng, không chỉ cung cấp về trải nghiệm người dùng mà còn trong việc truyền tải thông tin một cách hiệu quả, trực quan. Đặc biệt là đối với các ứng dụng web, di động.

4.2. Giao diện người dùng và khách hàng

4.2.1. Trang đăng ký

The image shows a registration form titled "ĐĂNG KÝ". It consists of four input fields: "Name", "Email", "Password", and "Repeat password". Each field has a placeholder text inside it. Below the fields is a red rectangular button with the text "Đăng Ký" in white.

Hình 4.1. Trang đăng ký

Khi vào trang web, đối với người dùng chưa có tài khoản sẵn có thể tiến hành tạo tài khoản thông qua chuỗi hoạt động gồm: Ấn vào nút đăng ký Người dùng (khách hàng) đăng ký tài khoản phải điền đầy đủ thông tin về tên tài khoản, email, mật khẩu (tối thiểu 6 ký tự) và nhập lại mật khẩu lần nữa.

4.2.2. Trang đăng nhập

The image shows a user interface for a login page. At the top center, the text 'ĐĂNG NHẬP' is displayed in a bold, black, sans-serif font. Below this, there are two input fields: the first is labeled 'Email' and the second is labeled 'Password'. A large red rectangular button with the white text 'Đăng Nhập' is positioned below the password field. To the right of the button, a blue link with the text 'Tạo tài khoản?' is visible.

Hình 4.2. Trang đăng nhập

Giao diện của trang đăng nhập khá đơn giản để người dùng dễ sử dụng bao gồm: Tiêu đề, khung thông tin email và mật khẩu. Người dùng sử dụng email và mật khẩu mà người dùng đã đăng ký để đăng nhập và bắt đầu sử dụng dịch vụ của website.

4.2.3. Giao diện trang chủ

The screenshot displays the homepage of the Nong San Viet website. At the top, there is a red header bar with the text "NÔNG SẢN VIỆT" and a search bar. Below the header, a large promotional banner for "ĐỒNG HÀNH CHUYỂN ĐỔI SỐ" (Digital Transformation Journey) featuring the FPT Polytechnic logo and a digital marketing theme. To the right of the banner are three smaller images: a wooden crate filled with fresh vegetables, a bowl of avocados, and a variety of vegetables like broccoli and radishes.

Below the banner, there are four promotional boxes:

- ƯU ĐÃI NÔNG SAN**: Sản phẩm nhập khẩu tươi ngon
- TRAI CÁY NHẬP KHẨU**: Giảm giá sốc mùa thu hoạch
- RAU CỦ SẠCH**: Giá tốt mỗi ngày
- GẠO HỮU CƠ CAO CẤP**: Ưu đãi đặc biệt

The main content area features several sections of products:

- RAU CỦ**: A grid of six vegetable images with their respective discount percentages: Apple (17%), Bananas (8%), Blueberries (10%), Grapes (19%), and Kiwi (20%).
- NÔNG SẢN VIỆT**: A grid of five product cards for different types of meat, each with a price, weight, and a "Mua Ngay" button.
- THỊT SẠCH**: A grid of five meat product cards, each with a price, weight, and a "Mua Ngay" button.
- KHÁC**: A grid of five other product cards, including bun gao, rice, bún bi, bánh mì, and cá chua.

At the bottom of the page is a footer navigation bar with links for "NÔNG SẢN", "ABOUT", "ALWAYS-ON SUPPORT", and "DELIVERY". It also includes links for "Về Chúng Tôi", "Blog", "Cơ Hội Nghề Nghiệp", "Giới Hàng", "Support 028.71.087.088 (07:00-21:00)", "Delivery 1800.6996 (07:00-21:00)", "Payment", "Cash voucher", and "Shipping methods". The footer also contains a copyright notice: "Copyright © 2020 Nông sản Việt. All rights reserved."

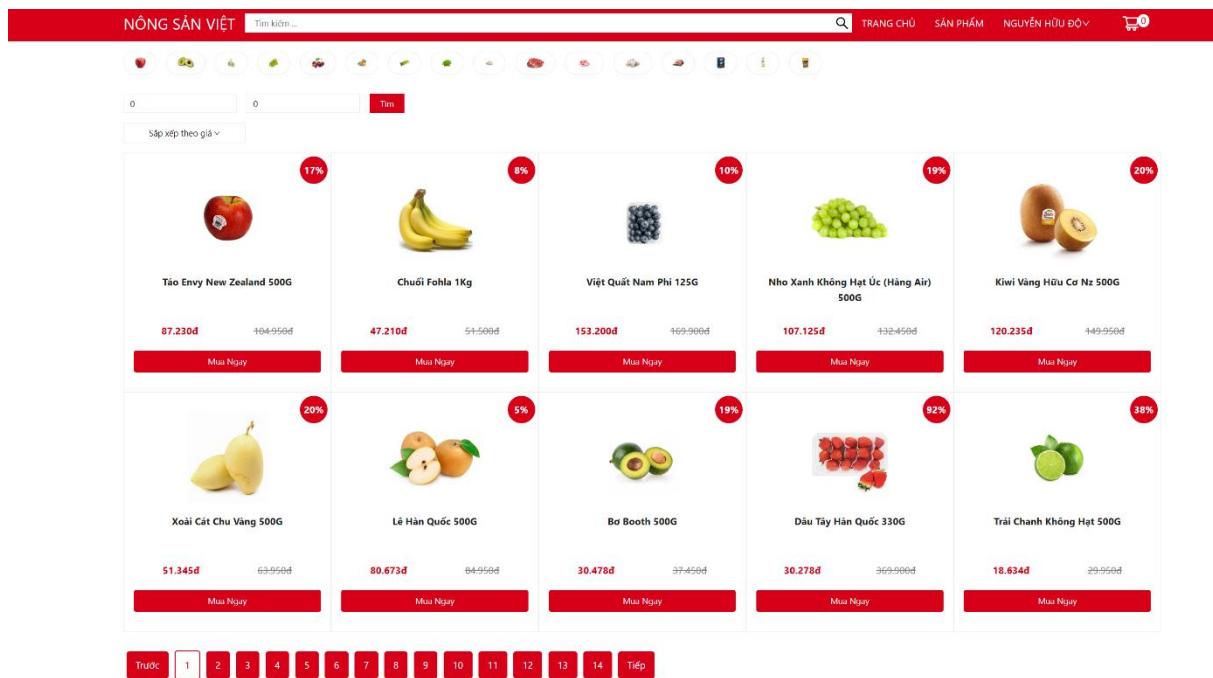
Hình 4.3. Giao diện trang chủ

Giao diện gồm các thành phần như:

- Header: Chứa Logo của website và thanh điều hướng giúp người dùng dễ dàng truy cập các trang khác nhau trên website.
- Slider: Giới thiệu và quảng cáo các sản phẩm với hình ảnh sinh động và thông điệp hấp, giúp thu hút người dùng.

- Sản phẩm Rau củ: Hiển thị các sản phẩm rau củ với hình ảnh, tên sản phẩm và giá cả, cho phép người dùng dễ dàng tìm kiếm và lựa chọn.
- Sản phẩm Thịt sạch: Phần dành riêng cho các sản phẩm về thịt (thịt bò, thịt heo, thịt gà), bao gồm hình ảnh thông tin chi tiết về giá cả.
- Sản phẩm Khác: Danh sách các sản phẩm khác như sản phẩm khô, đóng hộp,... Thông tin về giá cả giúp người dùng có cái nhìn toàn diện về các lựa chọn.
- Footer: Chứa thông tin liên hệ, liên kết các trang mạng xã hội và các thông tin khác liên quan đến dịch vụ khách hàng

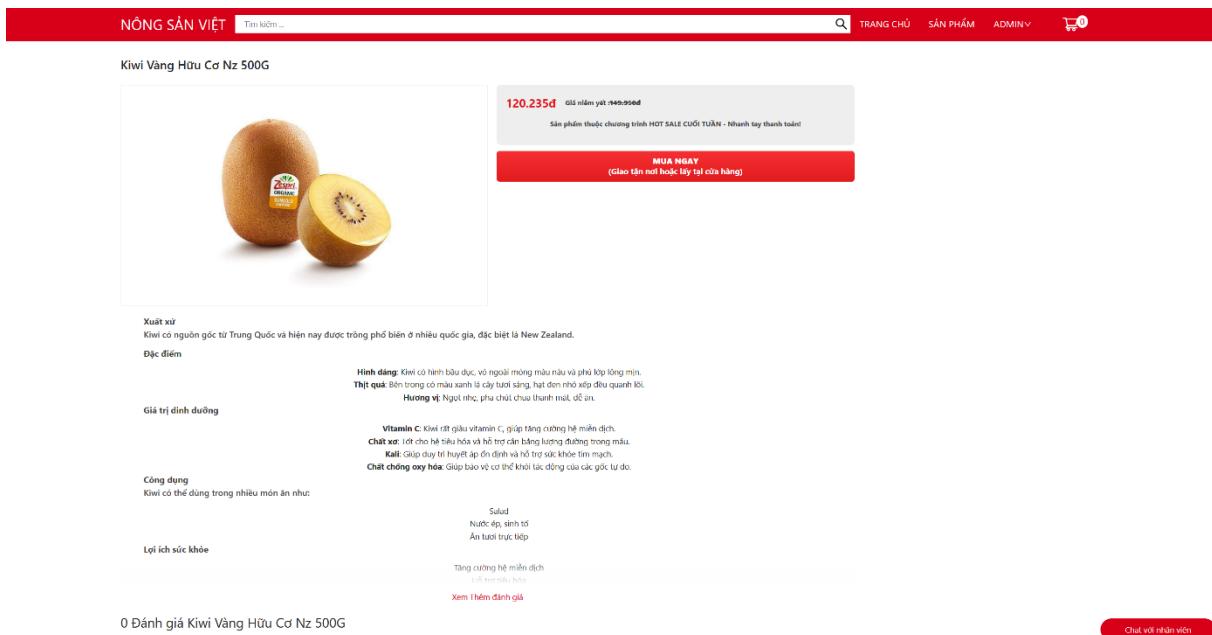
4.2.4. Giao diện sản phẩm



Hình 4.4. Giao diện sản phẩm

Giao diện hiển thị toàn bộ sản phẩm của trang web. Người dùng chọn bộ lọc để xem những sản phẩm phù hợp với nhu cầu. Ta có thể lục sản phẩm theo hình ảnh, tên hoặc theo giá cả (người dùng có thể nhập giá muốn tìm tùy theo nhu cầu).

4.2.5. Giao diện xem thông tin sản phẩm



Hình 4.5. Giao diện xem thông tin sản phẩm

Phần giao diện này bao gồm các thông tin về tên sản phẩm, hình ảnh, giá cả, đánh giá và mô tả chi tiết sản phẩm. Người dùng click vào sản phẩm và xem các thông tin trên.

4.2.6. Giao diện đánh giá của khách hàng



Hình 4.6. Giao diện đánh giá của khách hàng

Người dùng xem thông tin đánh giá sản phẩm hoặc tự mình đánh giá (nếu người dùng đã mua sản phẩm đó). Thông tin đánh giá sẽ được đăng công khai trong giao diện sản phẩm.

4.2.7. Giao diện sản phẩm liên quan

Hình 4.7. Sản phẩm liên quan

Hiển thị các sản phẩm liên quan trong danh mục sản phẩm khách hàng lựa chọn, gồm các thông tin về tên và giá cả.

4.2.8. Giao diện hồ sơ người dùng

Hình 4.8. Giao diện hồ sơ người dùng

Trong giao diện hồ sơ người dùng, người dùng có thể cập nhật thông tin sau khi tạo tài khoản như thay đổi họ & tên, địa chỉ email, số điện thoại và địa chỉ giao hàng.

4.2.9. Giao diện lịch sử mua hàng

Sản Phẩm	Số Lượng	Giá
Dứa Lưới Ruột Cam Amore 1.5-2.5Kg	x1	21.578
Nho Mẫu Đơn Hàn Quốc Hộp 450Gr +	x1	300.576
Tổng số tiền : 322.154đ		
Tổng số tiền: 0đ		
Việt Quất Nam Phi 125G	x1	153.200
Bà Rịa Meat Master 400G	x1	90.174
Tổng số tiền : 243.374đ		
Việt Quất Nam Phi 125G	x1	153.200
Bà Rịa Meat Master 400G	x1	90.174
Tổng số tiền : 243.374đ		

Hình 4.9. Giao diện lịch sử mua hàng

Trong giao diện lịch sử mua hàng, người dùng có thể xem tất cả các đơn đặt hàng và trạng thái đơn hàng. Bao gồm thông tin về đơn hàng.

4.2.10. Giao diện giỏ hàng

Tiếp tục mua hàng

Giỏ hàng

Sản Phẩm	Mô tả	Giá	Số Lượng	Tác vụ
Táo Envy New Zealand 500G	87.230	1	Xóa khỏi giỏ hàng	
Việt Quất Nam Phi 125G	153.200	1	Xóa khỏi giỏ hàng	
Thịt Thăn Ngoại Bò Úc Teys White 300G	204.535	1	Xóa khỏi giỏ hàng	
Thịt Đầu Thăn Ngoại Bò Úc Không Xương O'Connor 300G	426.247	1	Xóa khỏi giỏ hàng	

Tổng tiền **871.212**

Đặt Hàng

Hình 4.10. Giao hiện giỏ hàng

Trong giao diện giỏ hàng, khách hàng có xem các sản phẩm đã thêm vào giỏ. Gồm các thông tin chi tiết về:

- Tên, số lượng, giá cả các sản phẩm đã thêm vào giỏ.
- Tổng tiền của tất cả các sản phẩm.

4.2.11. Giao diện thanh toán

The screenshot displays a user interface for payment selection. At the top, there is a section labeled 'THÔNG TIN KHÁCH HÀNG' (Customer Information) with fields for name ('Nguyễn Hữu Độ') and phone number ('0899981800'). Below this is a 'CHỌN ĐỊA CHỈ' (Select Address) section with dropdown menus for city ('Hồ Chí Minh'), district ('Quận Gò Vấp'), street ('Phường 1'), and house number ('29, Tân Phú'). At the bottom, there is a 'CHỌN PHƯƠNG THỨC THANH TOÁN' (Select Payment Method) section with two options: 'Thanh toán khi nhận hàng' (Pay on delivery) and 'Thanh toán Online' (Online payment).

Hình 4.11. Giao diện thanh toán

Sau khi đặt hàng, khách hàng tiến hành thanh toán sản phẩm, bao gồm các bước:

- Nhập thông tin bao gồm tên, số điện thoại, địa chỉ.
- Lựa chọn phương thức thanh toán.
- Thanh toán khi nhận hàng.
- Thanh toán Online (bằng Paypal, thẻ tín dụng ghi nợ hoặc Vnpay).

THÔNG TIN KHÁCH HÀNG

Nguyễn Hữu Độ	0899981800
---------------	------------

CHỌN ĐỊA CHỈ

Hồ Chí Minh	Quận Gò Vấp
Phường 1	29, Tân Phú

CHỌN PHƯƠNG THỨC THANH TOÁN

Thanh toán khi nhận hàng	Thanh toán Online
--------------------------	-------------------

Đặt Hàng

The screenshot shows a user interface for placing an order. At the top, it displays 'THÔNG TIN KHÁCH HÀNG' (Customer Information) with fields for name ('Nguyễn Hữu Độ') and phone number ('0899981800'). Below this is a section titled 'CHỌN ĐỊA CHỈ' (Select Address) with dropdown menus for 'Hồ Chí Minh' and 'Quận Gò Vấp', and further details for 'Phường 1' and '29, Tân Phú'. Under 'CHỌN PHƯƠNG THỨC THANH TOÁN' (Select Payment Method), there are two options: 'Thanh toán khi nhận hàng' (Pay on delivery) and 'Thanh toán Online' (Online payment). A large red button at the bottom is labeled 'Đặt Hàng' (Place Order).

Hình 4.12. Thanh toán khi nhận hàng

Đối với thanh toán khi nhận hàng, khách hàng nhập thông tin bao gồm tên, số điện thoại, địa chỉ. Sau đó lựa chọn “Thanh toán khi nhận hàng”.

THÔNG TIN KHÁCH HÀNG

Nguyễn Văn A	0899981800
--------------	------------

CHỌN ĐỊA CHỈ

Hồ Chí Minh	Huyện Cần Giờ
Xã Tam Thôn Hiệp	29, Tân Phú

CHỌN PHƯƠNG THỨC THANH TOÁN

Thanh toán khi nhận hàng	Thanh toán Online
--------------------------	-------------------



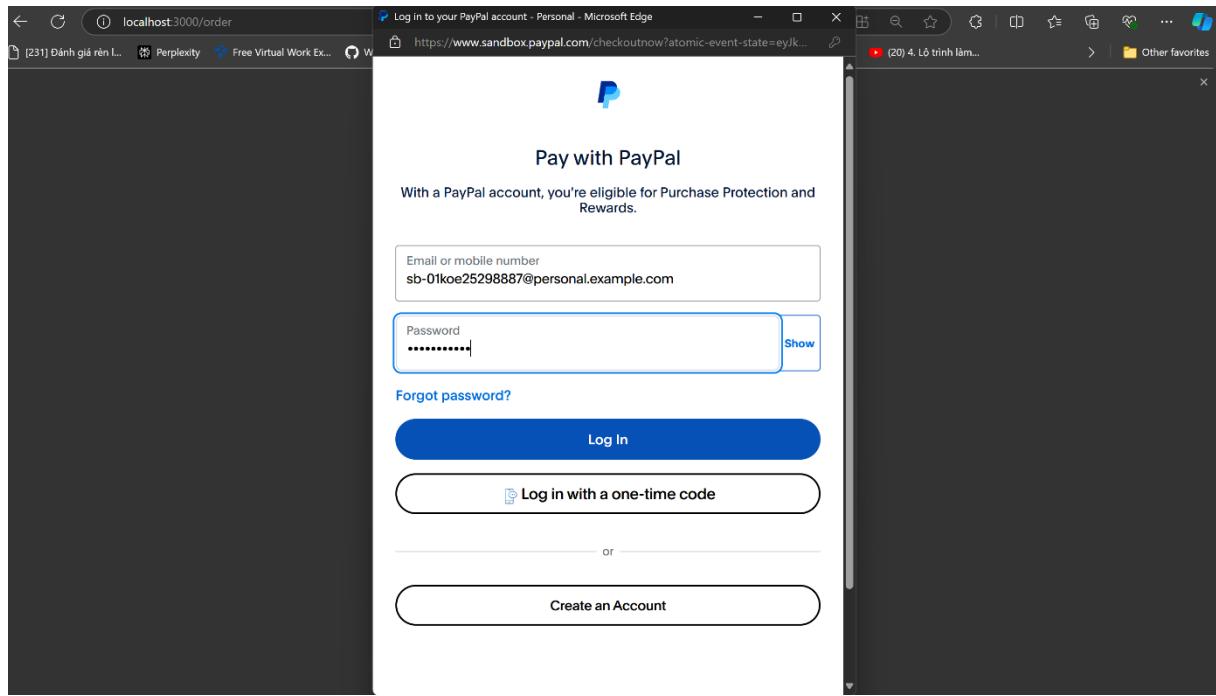


 Thẻ ghi nợ hoặc tín dụng

Được hỗ trợ bởi PayPal

Hình 3.14. Thanh toán bằng Paypal

Đối với thanh toán Online, cụ thể là thanh toán bằng Paypal. Khách hàng nhập thông tin sau đó chọn “Thanh toán Online” chọn “Paypal”.



Hình 4.13. Thực hiện đăng nhập trước khi thanh toán

Hệ thống hiển thị yêu cầu người dùng đăng nhập thông tin tài khoản Paypal (nếu đã có).

 PayPal Checkout - Personal - Microsoft Edge — □ ×

 <https://www.sandbox.paypal.com/webapps/hermes?flow=1-P&ulRe...>  

		\$1.00
---	---	---------------

Ship to John Doe [Change](#)
1 Main St, San Jose, CA 95131

Pay with Split

 **CREDIT UNION 1** \$1.00
Checking5013
Backup: Visa4452
 Make this my preferred way to pay

 **PayPal Credit**
You have PayPal Credit! Pay over time for your purchase of \$1.00. [See terms](#)

 **Visa**
Credit4452

+ Add debit or credit card

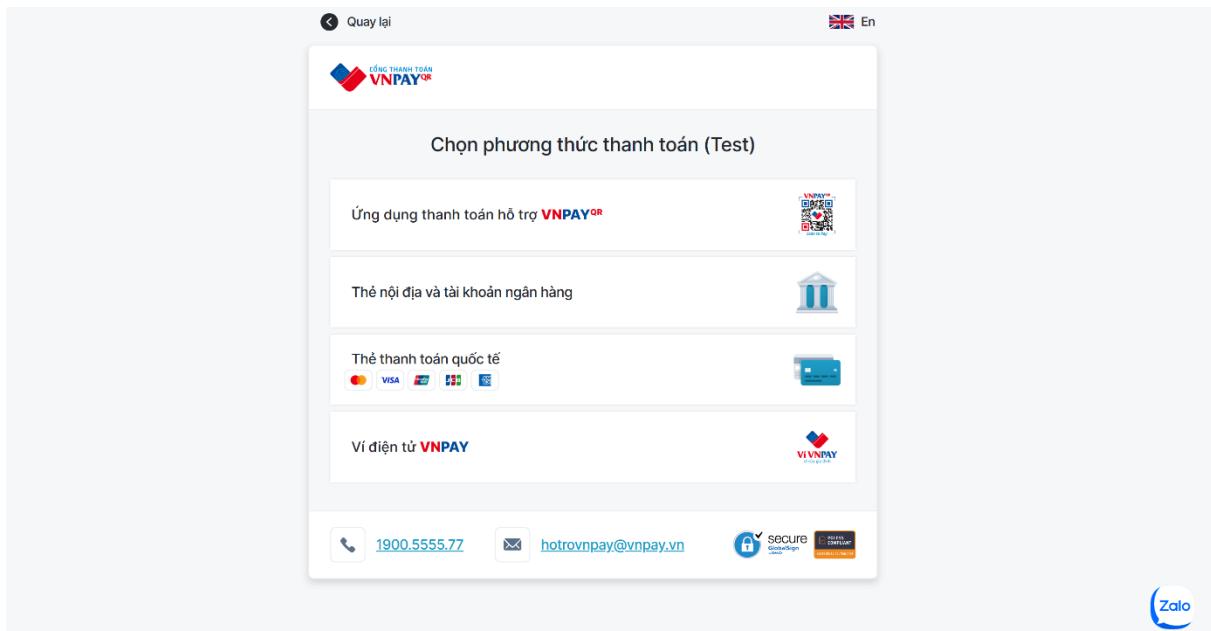
Pay Later

Complete Purchase

[Payment method rights](#)

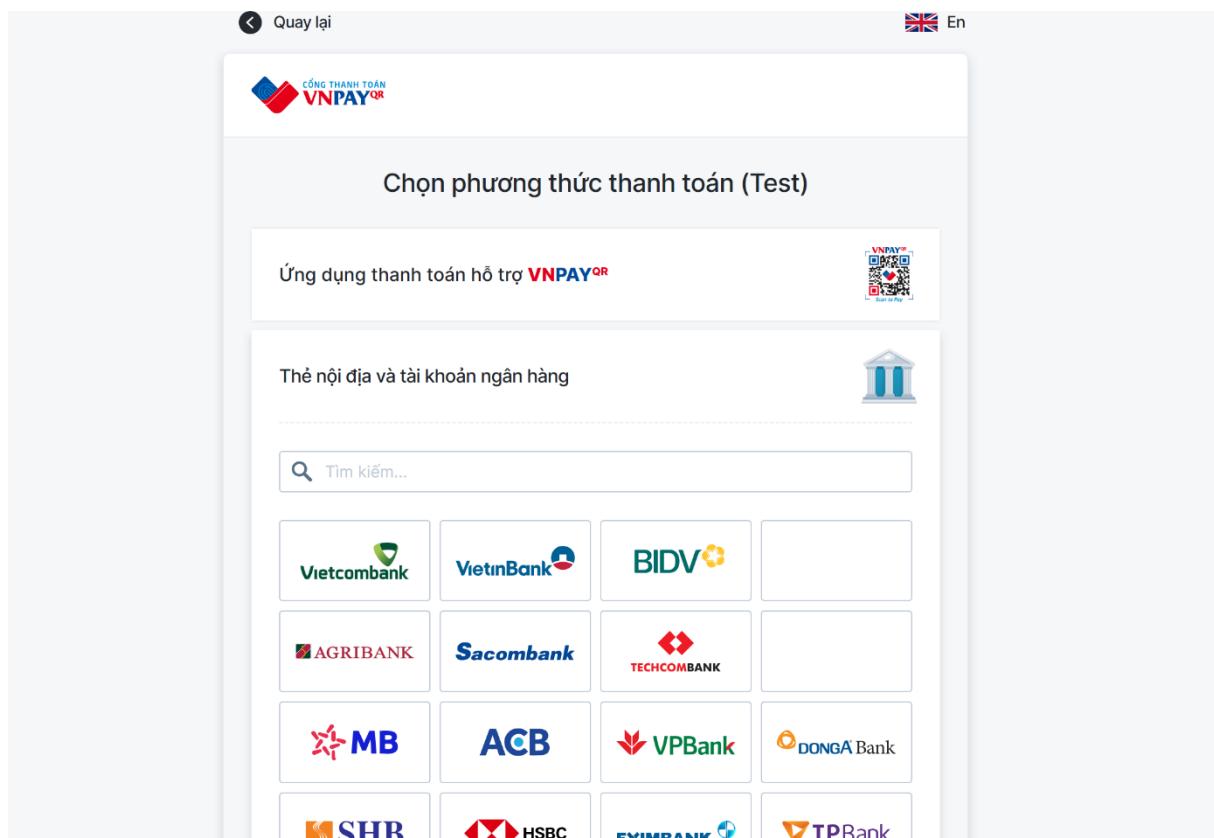
Hình 4.14. Thực hiện lựa chọn thành toán

Lựa chọn thanh toán và chuyển sang bước tiếp theo.



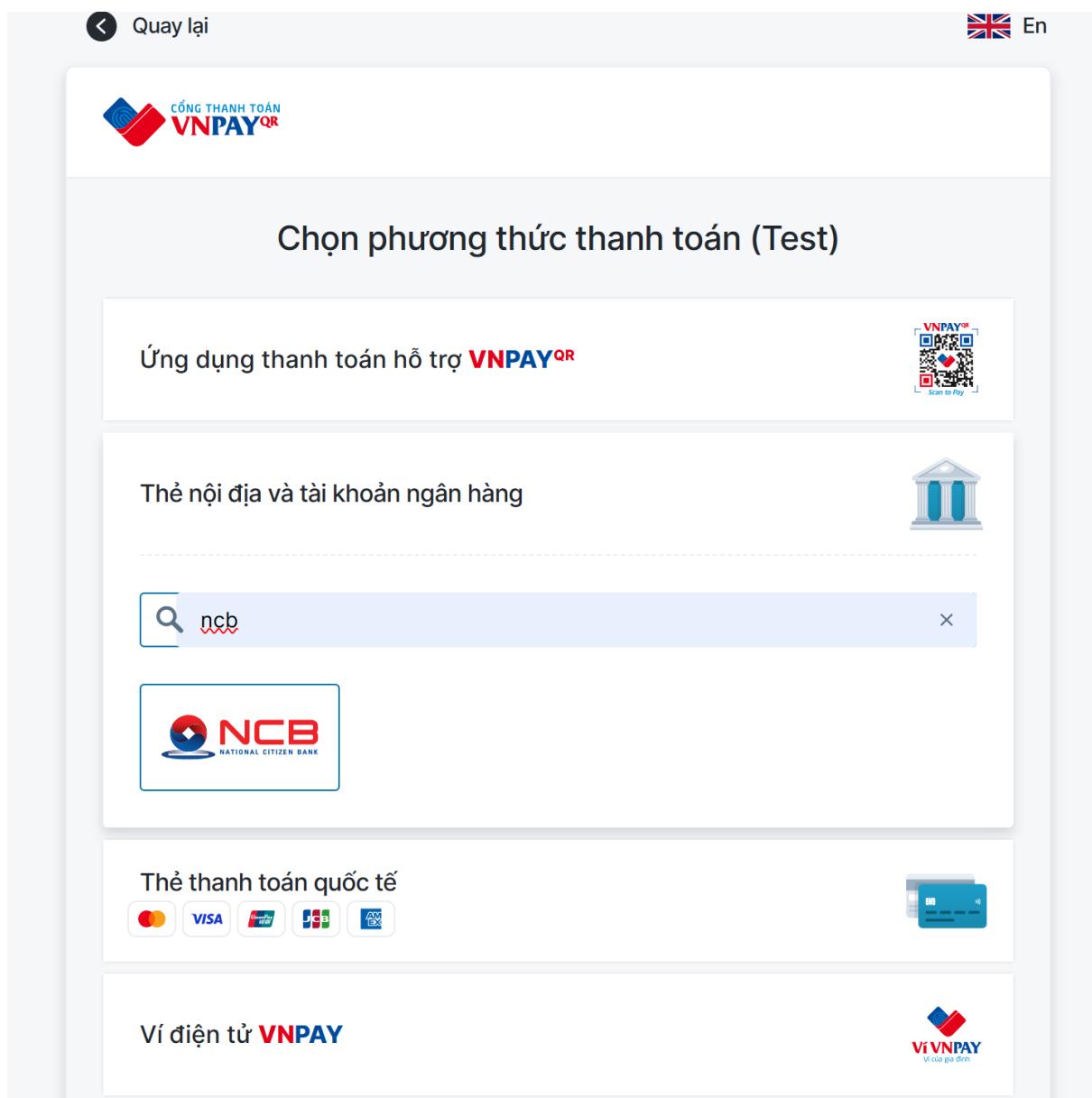
Hình 4.15. Thanh toán bằng Vnpay

Đối với thanh toán Online bằng Vnpay, khách hàng vẫn nhập thông tin. Sau đó tiến hành chọn “Ví Vnpay”, hệ thống sẽ hiển thị chọn phương thức thanh toán.



Hình 4.16. Thực hiện chọn ngân hàng thanh toán

Khi thanh toán bằng ngân hàng, khách hàng chọn ngân hàng phù hợp để tiến hành thanh toán.



Hình 4.17. Nhập tên ngân hàng thực hiện thanh toán
Nhập tên ngân hàng

The screenshot shows a two-panel payment interface. The left panel, titled 'Thông tin đơn hàng (Test)', displays the following details:

- Số tiền thanh toán: **322.154VND**
- Giá trị đơn hàng: **322.154VND**
- Phí giao dịch: **0VND**
- Mã đơn hàng: **1728488591656**
- Nhà cung cấp: **VNPAY - TryItNow**

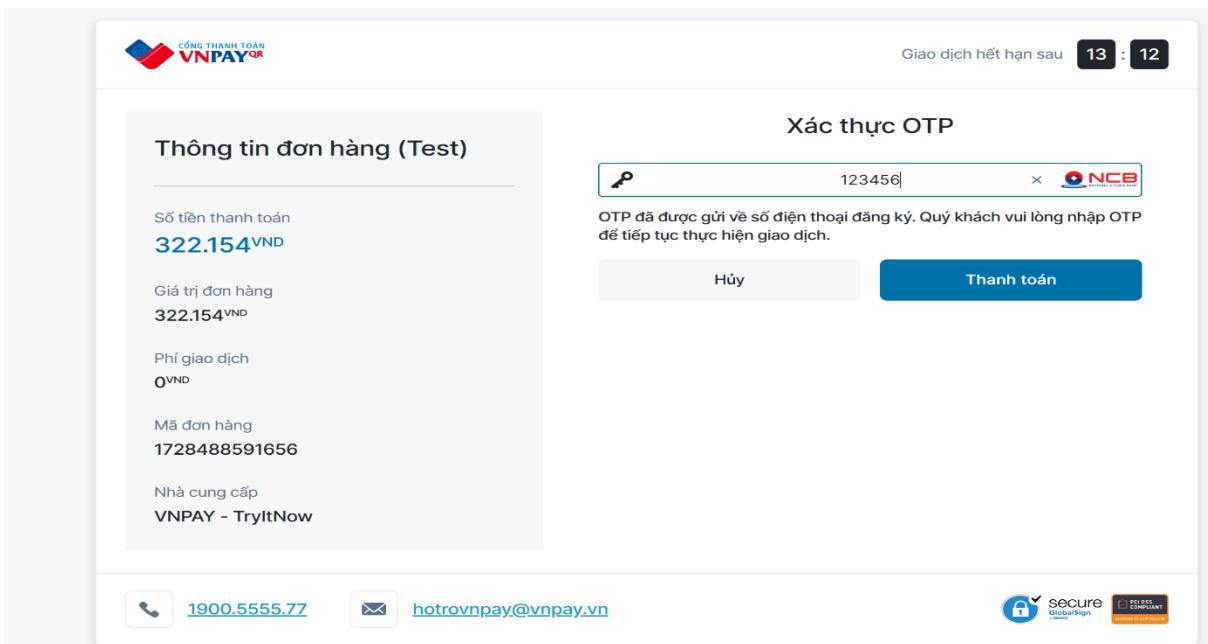
The right panel, titled 'Thanh toán qua Ngân hàng NCB', is for entering card information:

- Thẻ nội địa**
- Số thẻ: *******2198** (with the NCB logo)
- Tên chủ thẻ: **NGUYEN VAN A**
- Ngày phát hành: **07/15**
- Mã khuyến mại: **Chọn hoặc nhập mã**
- [Điều kiện sử dụng dịch vụ](#)
- Buttons: **Hủy thanh toán** and **Tiếp tục**

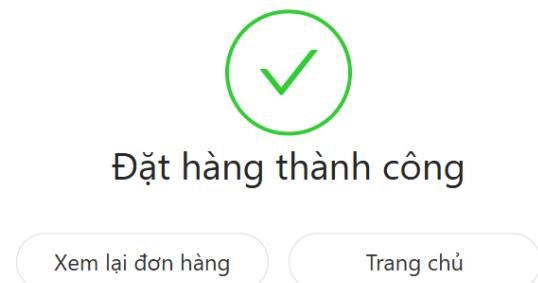
A small Zalo logo is visible in the bottom right corner.

Hình 4.18. Nhập thông tin để tiến hành thanh toán

Khách hàng nhập thông tin cụ thể gồm tên tài khoản, số thẻ, ngày phát hành sau đó nhấn “Tiếp tục”



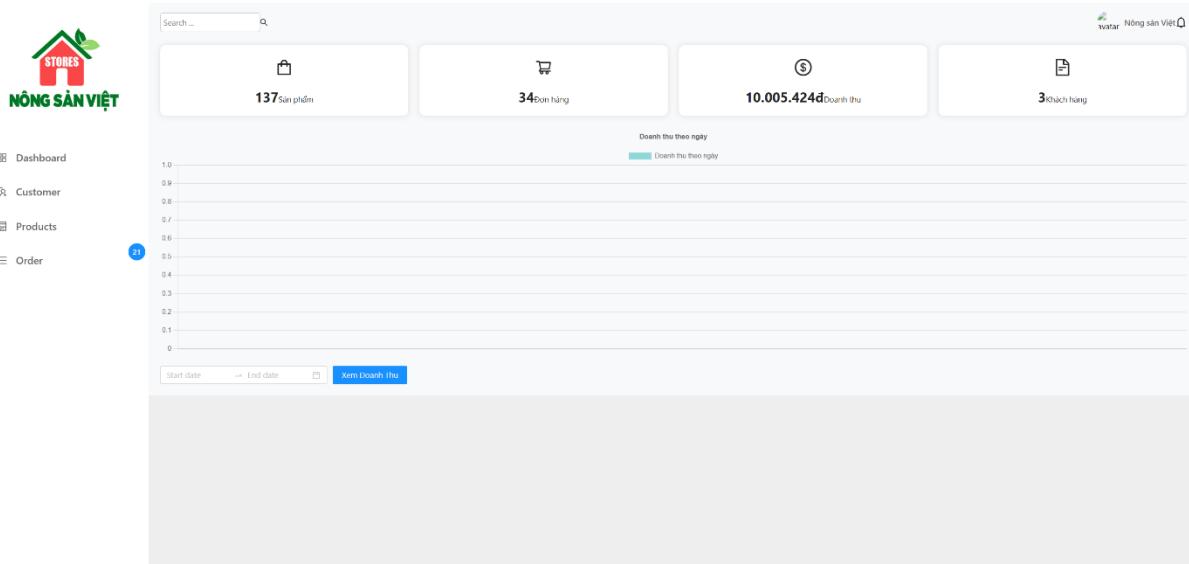
Hình 4.19. Nhập OTP tiến hành xác thực



Hình 4.20. Thanh toán thành công

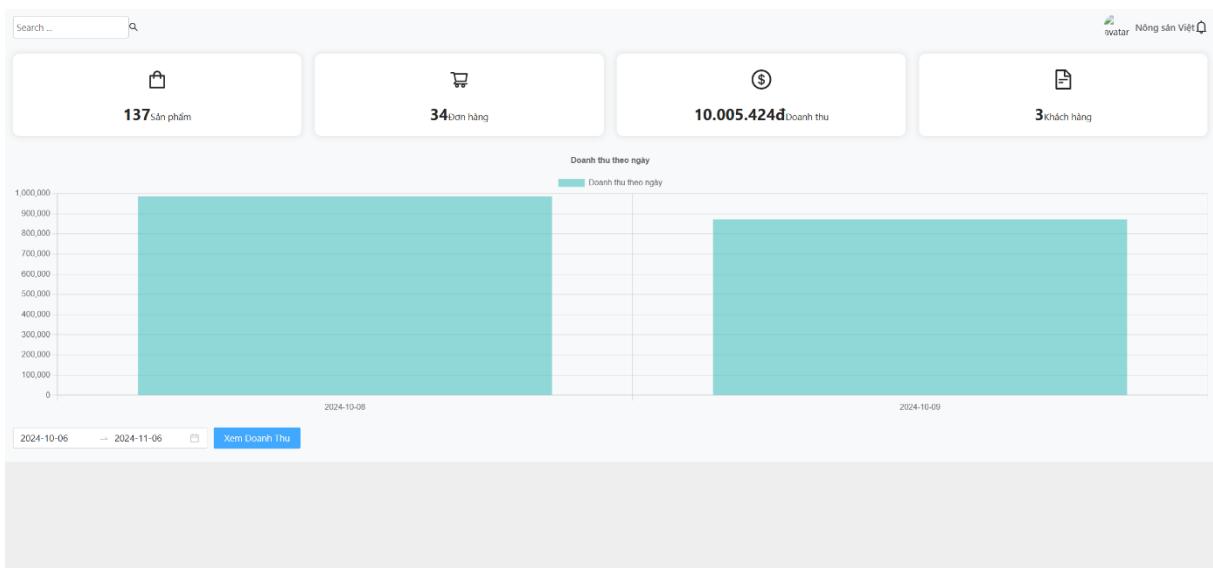
Dù sử dụng phương thức nào để thanh toán thì cuối cùng hệ thống sẽ xác nhận và hiển thị thông báo đặt hàng thành công.

4.3. Giao diện quản trị viên



Hình 4.21. Giao diện trang quản lý

Khi đăng nhập bằng tài khoản Admin, người dùng chọn mục “Admin” và được chuyển tới trang Quản lý. Tạo trang này, Admin quản lý doanh thu, sản phẩm, đơn hàng, khách hàng và biểu đồ thống kê báo cáo theo thời gian.



Hình 4.22. Giao diện biểu đồ

Khu vực biểu đồ có thể thống kê được hiện tại doanh thu trong thời gian Admin muốn hiển thị.

The screenshot shows a web-based application for managing a store. On the left is a sidebar with a logo for 'NÔNG SẢN VIỆT STORES' and navigation links for Dashboard, Customer, Products, and Order. The main area is titled 'Customers' and displays a table with two rows of user information:

	Name	Email	Address	Phone
1	admin	admin@gmail.com	Bình Thạnh	0899981800
2	Nguyễn Hữu Độ	nhd03101@gmail.com		

Hình 4.23. Giao diện quản lý người dùng

Quản lý số lượng về người dùng hiện có, có thể xóa tài khoản người dùng.

The screenshot shows a 'Products' management interface. The sidebar includes the 'NÔNG SẢN VIỆT STORES' logo and links for Dashboard, Customer, Products, and Order. The main content area lists four products with their details:

	Image	Name	Price	Type
1		Kiwi Vàng Hữu Cơ Nz 500G	120.235	rau củ
2		Xoài Cát Chu Vàng 500G	51.345	rau củ
3		Lê Hàn Quốc 500G	80.673	rau củ
4		Bơ Booth 500G	30.478	rau củ

A pagination bar at the bottom indicates page 2 of 10.

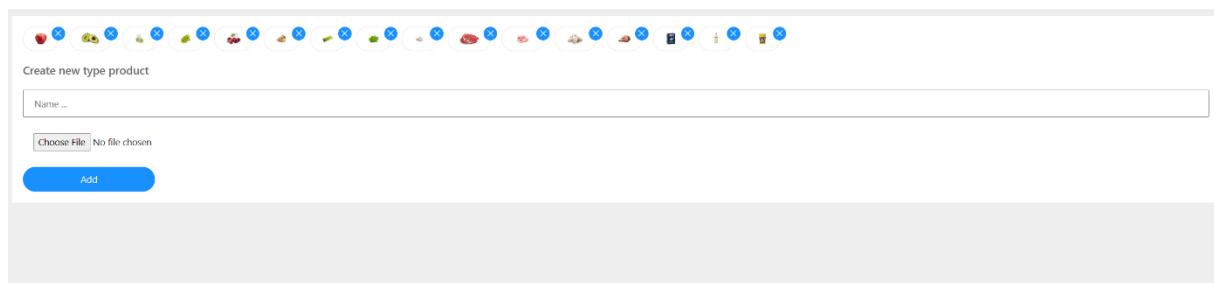
Hình 4.24. Giao diện quản lý sản phẩm

Trang quản lý sản phẩm này giúp chủ cửa hàng quản lý chi tiết về sản phẩm hiện có. Đồng thời thêm sản phẩm, chỉnh sửa, ẩn, xóa sản phẩm. Bên cạnh đó có thể thêm mô tả chi tiết các sản phẩm, phân loại sản phẩm.



Hình 4.25. Giao diện thêm sản phẩm

Admin tiến hành thêm sản phẩm.



Hình 4.26. Giao diện quản lý sản phẩm

Admin tiến hành tạo lựa chọn sản phẩm, theo hình ảnh bao gồm tên và hình ảnh loại sản phẩm đó.

Orders

[All Orders](#)

Order 1:

Qty: 1	Qty: 1	Qty: 1	Qty: 1
Táo Envy New Zealand 500G 87.230	Vịt Quail Nam Phi 125G 153.200	Thịt Thăn Ngoại Bò Úc Teys White 300G 204.535	Thịt Dầu Thăn Ngoại Bò Úc Không Xương O'Connor 300G 426.247

Tổng tiền: 871.212
Địa chỉ: Nguyễn Hữu Đô, Hồ Chí Minh, Quận Gò Vấp, Phường 1, 29, Tân Phú, 0899981800
Paid : 20/7, Tue Oct 08 2024

[Xác nhận đơn hàng](#)

Order 2:

Qty: 8	Qty: 1	Qty: 1
Táo Envy New Zealand 500G 87.230	Chả Sô Iohà 1kg 47.210	Vịt Quail Nam Phi 125G 153.200

Tổng tiền: 898.250
Địa chỉ: Nguyễn Hữu Đô, Hồ Chí Minh, Thành Phố Thủ Đức, Phường Bình Trưng Tây, 29, Tân Phú, 0899981100
Paid : 20/7, Tue Oct 08 2024

Order 3:

Qty: 1
Táo Envy New Zealand 500G 87.230

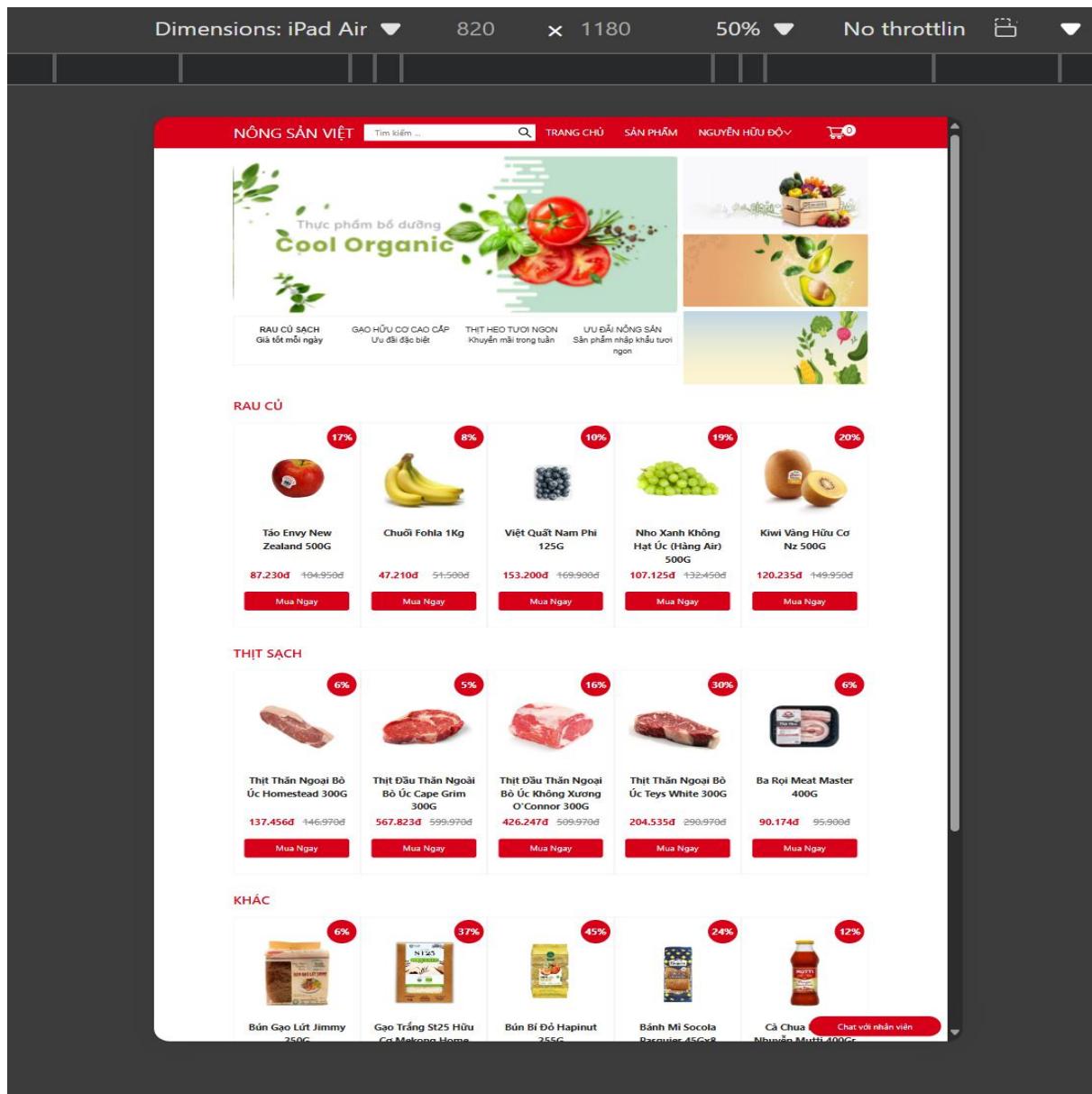
Hình 4.27. Giao diện quản lý đơn hàng

Thông tin tất cả đơn hàng của khách hàng được hiển thị ở trang này. Trang này có thể xem thông tin khách hàng, số điện thoại, số tiền đơn đặt, trạng thái thanh toán, trạng thái giao hàng. Admin có thể cập nhật trạng thái đơn hàng.

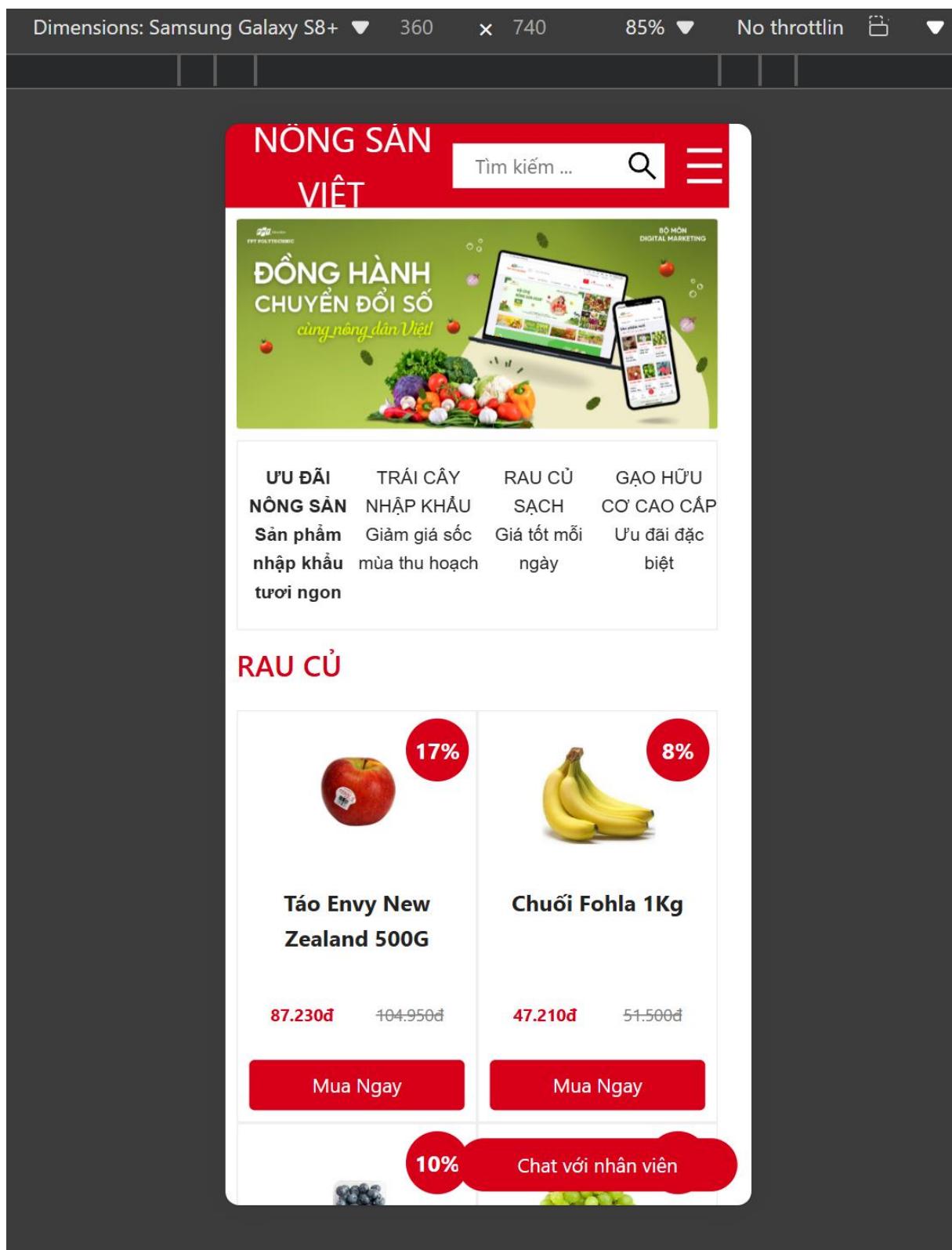
CHƯƠNG 5: KIỂM TRA VÀ THỬ NGHIỆM

Chương này sẽ tiến hành kiểm thử sử dụng trang web trên nhiều thiết bị như điện thoại, ipad,... Cùng với các giao diện tương thích và các trình duyệt phổ biến nhất hiện nay.

5.1. Kiểm thử khả năng tương thích



Hình 5.1. Trên thiết bị PC, laptop, ipad



Hình 5.2. Trên thiết bị di động

CHƯƠNG 6: KẾT LUẬN

Chương này nêu kết luận và hướng phát triển của đề tài

6.1. Kết quả đạt được.

6.1.1. Ưu điểm

Qua quá trình học tập và nghiên cứu đồ án, bản thân đã cơ bản nắm rõ quy trình xây dựng một website thương mại điện tử hoàn chỉnh sử dụng công nghệ MERN Stack. Có thêm kiến thức và kỹ năng sử dụng NodeJS và ExpressJS để xây dựng các API ở phía Backend, ReactJS phía Frontend xử lý và hiển thị cho giao diện người dùng. Cùng với hệ quản trị cơ sở dữ liệu MongoDB dùng để lưu trữ dữ liệu của hệ thống website. Với đề tài xây dựng một website thương mại điện tử tuy chưa được hoàn thiện hoàn toàn, nhưng nó mang đến sự tiện lợi, nhanh chóng và hiệu quả cho việc quản lý sản phẩm, đơn hàng cũng như tạo sự trải nghiệm mua sắm trực tuyến thuận tiện cho người dùng.

6.1.2. Hạn chế

Do thời gian thực hiện đề tài còn hạn chế nên không thể những sai sót trong quá trình nghiên cứu và phát triển cụ thể là chưa hoàn thiện một số tính năng như phân quyền người dùng chưa được chi tiết (về quyền cho nhân viên, khách hàng, admin), chưa có tính năng hủy sản phẩm cho khách hàng muốn hủy sản phẩm không muốn mua nữa. Thiếu các tính năng nâng cao về trải nghiệm người dùng như tính năng đa ngôn ngữ chưa được phát triển và chưa có phiên bản cho ứng dụng di động.

6.2. Phương hướng phát triển

6.2.1. Cải thiện tính năng hiện có

Đối với hệ thống quản lý, nâng cấp hệ thống phân quyền người dùng. Hoàn thiện chức năng hủy sản phẩm, bổ sung về tính năng đa ngôn ngữ, bổ sung thông tin chi tiết về sản phẩm (kích thước, số lượng, sản phẩm).

6.2.2. Phát triển tính năng mới

Xây dựng hệ thống gợi ý sản phẩm (Recommendation system) dựa trên hành vi người dùng. Tích hợp các tính năng sử dụng AI (chat, hỏi đáp, thông báo). Phát triển phiên bản ứng dụng di động sử dụng React Native.

6.3. Kết luận

Để tiếp tục hoàn thiện và phát triển đê tài, rất mong nhận được sự góp ý và hướng dẫn từ các Thầy/Cô. Những nhận xét và đề xuất từ Thầy/Cô là động lực và định hướng vô cùng quan trọng, giúp đê tài không ngừng cải thiện, khắc phục các hạn chế hiện tại và phát huy tối đa tiềm năng của hệ thống. Sự hỗ trợ của các thầy cô sẽ là nền tảng vững chắc để đê tài ngày càng hoàn thiện, đáp ứng tốt hơn với nhu cầu thực tế của thị trường thương mại điện tử đang phát triển.

TÀI LIỆU THAM KHẢO

- [1] A. Technologie, “Giới thiệu về ReactJS - Một thư viện JavaScript hỗ trợ xây dựng giao diện,” Arrow Technologies, 7 October 2023. [Trực tuyến]. Available: <https://arrow-tech.vn/gioi-thieu-ve-reactjs-mot-thu-vien-javascript-ho-tro-xay-dung-giao-dien>. [Đã truy cập 10 October 2024].
- [2] “SPA (Single Page Application) Là Gì? Tại Sao Được Nhiều Doanh Nghiệp Tin Dùng,” JAMstack Vietnam (as the source is from jamstackvietnam.com), 15 May 2023 . [Trực tuyến]. Available: <https://jamstackvietnam.com/blog/thiet-ke-website/spa-single-page-application-la-gi-tai-sao-duoc-nhieu-doanh-nghiep-tin-dung>. [Đã truy cập 11 October 2024].
- [3] R. D. Team, “Introducing JSX,” Meta , 15 August 2023 . [Trực tuyến]. Available: <https://legacy.reactjs.org/docs/introducing-jsx.html>. [Đã truy cập 12 October 2024].
- [4] N. Foundation, “About Node.js,” Node.js, 20 June 2023 . [Trực tuyến]. Available: <https://nodejs.org/en/about/>. [Đã truy cập 12 October 2024].
- [5] M. Inc., “What is MongoDB?,” MongoDB, 10 July 2023 . [Trực tuyến]. Available: <https://www.mongodb.com/company/what-is-mongodb>. [Đã truy cập 12 October 2024].
- [6] “Phần 1: Tìm hiểu Express.js Framework,” Viblo, 1 September 2023 . [Trực tuyến]. Available: <https://viblo.asia/p/phan-1-tim-hieu-express-js-framework-Qbq5Qq7m5D8>. [Đã truy cập 12 October 2024].

- [7] E. D. Team, “Express 4.x API Reference,” Express.js, 10 May 2023. [Trực tuyến]. Available: <https://expressjs.com/en/4x/api.html>. [Đã truy cập 12 October 2024].
- [8] R. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” University of California, Irvine, 6 June 2000. [Trực tuyến]. Available: https://ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm. [Đã truy cập 12 October 2024].
- [9] [Trực tuyến].