

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM**



NGUYỄN VĂN SANH: 20110556

NGUYỄN TẤN PHÁT: 20110334

Đề Tài :

**XÂY DỰNG WEB QUẢN LÝ SẢN
THƯƠNG MẠI ĐIỆN TỬ NEWEE**

KHÓA LUẬN TỐT NGHIỆP KỸ SƯ CNTT

**GIÁO VIÊN HƯỚNG DẪN
ThS. NGUYỄN MINH ĐẠO**

KHÓA 2020 – 2024

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM



NGUYỄN VĂN SANH: 20110556

NGUYỄN VĂN SANH: 20110334

Đề Tài :

**XÂY DỰNG WEB QUẢN LÝ SẢN
THƯƠNG MẠI ĐIỆN TỬ NEWEE**

KHÓA LUẬN TỐT NGHIỆP KỸ SƯ CNTT

GIÁO VIÊN HƯỚNG DẪN
ThS. NGUYỄN MINH ĐẠO

KHÓA 2020 – 2024

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên 1 : **Nguyễn Văn Sanh**

MSSV 1: **20110556**

Họ và tên Sinh viên 2 : **Nguyễn Tấn Phát**

MSSV 2: **20110334**

Ngành: **Công nghệ Thông tin**

Tên đề tài: **Xây dựng web quản lý sàn thương mại điện tử**

Họ và tên Giảng viên hướng dẫn: **ThS. Nguyễn Minh Đạo**

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm

4. Đề nghị cho bảo vệ hay không ?

5. Đánh giá loại :

6. Điểm :

Tp. Hồ Chí Minh, ngày tháng năm 2024

Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên Sinh viên 1 : **Nguyễn Văn Sanh**

MSSV 1: **20110556**

Họ và tên Sinh viên 2 : **Nguyễn Tấn Phát**

MSSV 2: **20110334**

Ngành: **Công nghệ Thông tin**

Tên đề tài: **Xây dựng web quản lý sàn thương mại điện tử**

Họ và tên Giảng viên phản biện: **TS. Lê Vĩnh Thịnh**

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm

4. Đề nghị cho bảo vệ hay không ?

5. Đánh giá loại :

6. Điểm :

Tp. *Hồ Chí Minh*, ngày *tháng* năm 2024

Giáo viên phản biện

(Ký & ghi rõ họ tên)

LỜI CẢM ƠN

Nhóm chúng em xin chân thành cảm ơn sâu sắc đến với các thầy cô của trường Đại học Sư phạm Kỹ thuật TP.HCM, và đặc biệt là thầy Nguyễn Minh Đạo đã giúp đỡ, hướng dẫn tận tình, chỉ bảo chúng em trong suốt thời gian học tập, trang bị những kiến thức cũng như những kỹ năng cần thiết để nhóm chúng em thực hiện đồ án.

Tuy nhiên, thời gian và lượng kiến thức có hạn nên nhóm chúng em không thể phát huy hết khả năng, ý tưởng vào đồ án lần này cũng như không thể tránh khỏi những sai sót khi báo cáo và thực hiện. Rất mong nhận được sự góp ý, đánh giá từ thầy để đồ án của nhóm chúng em được hoàn thiện và phát triển hơn.

Chúng em xin chân thành cảm ơn.

Trường Đại học Sư phạm Kỹ Thuật TP.HCM

Khoa Công nghệ thông tin

ĐỀ CƯƠNG KHOÁ LUẬN TỐT NGHIỆP

Họ và tên Sinh viên 1: **Nguyễn Văn Sanh**

MSSV 1: **20110556**

Họ và tên Sinh viên 2: **Nguyễn Tấn Phát**

MSSV 2: **20110334**

Thời gian làm luận văn: **Từ tháng 02 năm 2024 đến tháng 07 năm 2024**

Chuyên ngành: **Công nghệ phần mềm**

Tên tiểu luận: **Xây dựng web quản lý sàn thương mại điện tử**

Giảng viên hướng dẫn: **ThS. Nguyễn Minh Đạo**

Nhiệm vụ của khóa luận:

1. Lý thuyết:

- Tìm hiểu được các công nghệ hỗ trợ: ReactJs, NextJs, RESTful API, WebSocket ...
- Tìm hiểu các yêu cầu và chức năng cơ bản của một sàn thương mại điện tử.
- Tìm hiểu về ngôn ngữ và kiến thức tương tác: Express routing, middleware, mongoose.
- Hiểu rõ về quá trình thanh toán trực tuyến, sự hỗ trợ từ các công nghệ để lưu trữ dữ liệu mua hàng của người dùng vào database.

2. Tính năng:

Xây dựng với 3 role User, Vendor và Admin gồm các chức năng cụ thể phục vụ cho việc mua bán hoặc quản lý như sau:

• User:

- Đăng ký, đăng nhập
- Chỉnh sửa thông tin cá nhân, địa chỉ giao hàng
- Xem sản phẩm
- Thêm sản phẩm vào giỏ hàng

- Đặt hàng (thanh toán VNPay)
- Xác nhận trạng thái đơn, hủy đơn
- Đánh giá sản phẩm sau khi đơn hàng hoàn thành
- Đăng ký bán hàng
- Báo cáo sản phẩm, review sản phẩm
- Đổi mật khẩu và quên mật khẩu
- **Vendor:**
 - Quản lý thông tin shop
 - Xem số tiền trong tài khoản
 - Quản lý sản phẩm
 - Quản lý danh mục shop
 - Quản lý đơn hàng
 - Xem thông báo (đơn hàng, review)
- **Admin:**
 - Xem sản phẩm
 - Vô hiệu hóa sản phẩm, review không phù hợp
 - Quản lý danh mục sản phẩm (category)
 - Quản lý user, shop
 - Quản lý các banner của sàn
 - Xem thông báo (tổ cáo)

Thời gian thực hiện: Bắt đầu từ ngày 19/02/2024 đến 11/07/2024

Mục lục:

1. Phần MỞ ĐẦU

1. Chương 1: TỔNG QUAN VẤN ĐỀ NGHIÊN CỨU
 - 1.1. Đặt vấn đề
 - 1.2. Mục đích nghiên cứu
 - 1.3. Mục tiêu đề tài
 - 1.4. Đối tượng và phạm vi nghiên cứu
 - 1.5. Phương pháp nghiên cứu

2. Phần NỘI DUNG

1. Chương 2: Cơ sở lý thuyết

- 1.1. MERN Stack
- 1.2. Mô hình Client-Server
- 1.3. NodeJS
- 1.4. ExpressJS
- 1.5. ReactJS
- 1.6. Redux
- 1.7. NextJs
- 1.8. WebSocket và SocketIO
- 1.9. MongoDB

2. Chương 3: Xác định và mô hình hoá

- 2.1. Hiện trạng TMĐT ở Việt Nam
- 2.2. Đặc tả chức năng
- 2.3. Chức năng của từng actors
- 2.4. Sơ đồ Use-case
- 2.5. Đặc tả Use-case

3. Chương 3: Thiết kế hệ thống

- 3.1. Kiến trúc hệ thống
- 3.2. ERD
- 3.3. Đặc tả Cơ sở dữ liệu
- 3.4 Sequence Diagram
- 3.5. Thiết kế giao diện

4. Chương 5: Cài đặt và kiểm thử

- 4.1. Các công cụ
- 4.2. Các công nghệ
- 4.3. Cài đặt
- 4.4. Kiểm thử

3. Phần KẾT LUẬN

4. Phần TÀI LIỆU THAM KHẢO

- [1] ExpressJS, đăng bởi Lê Văn Liêm : <https://viblo.asia/p/phan-1-tim-hieu-express-js-framework-Qbq5Qq7m5D8>
- [2] NodeJS, đăng bởi TopDev : <https://topdev.vn/blog/node-js-la-gi/>
- [3] Nodemailer, đăng bởi nodemailer.com : <https://nodemailer.com>
- [4] MongoDB, đăng bởi ITViec: <https://itviec.com/blog/mongodb-la-gi/>
- [5] Mongoose, đăng bởi Hoàng Hùng: <https://viblo.asia/p/mongoose-cho-mongodb-nodejs-Qbq5QWvJZD8>
- [6] SocketIO, đăng bởi Lê Xuân Duy: <https://viblo.asia/p/buoc-dau-lam-quen-voi-nodejs-va-socketio-MJyGjQrWvPB>
- [7] JavaScript, đăng bởi tenten.vn: <https://tenten.vn/tin-tuc/javascript-la-gi/>
- [8] ReactJS, đăng bởi Từ Quốc Hưng : <https://200lab.io/blog/reactjs-la-gi/>
- [9] JWT, đăng bởi Nguyễn Tâm: <https://viblo.asia/p/json-web-token-la-gi-aWj533go56m>
- [10] Redux, đăng bởi Ngô Tuấn Doãn : <https://viblo.asia/p/redux-trong-react-la-gi-1Je5EDX0lnL>

Kế hoạch thực hiện:

STT	Thời gian	Công việc	Ghi chú
1	15/2 – 14/3	Khảo sát thực trạng, thiết kế tính năng tổng quát của hệ thống, thiết kế UI trang User, Admin	
2	15/3 – 30/3	Thiết kế cơ sở dữ liệu model entity, xây dựng luồng routing, luồng bảo mật, dựng UI trang User, Admin	
3	31/3 – 10/4	Xây dựng tính năng quản lý sản phẩm trên trang Admin	
4	11/4 – 10/5	Chỉnh sửa UI và xử lý các dữ liệu trên trang Admin	
5	11/5 – 15/6	Chỉnh sửa UI và xử lý dữ liệu trên trang User	
6	16/6 – 11/7	Điều chỉnh, sửa lỗi và viết báo cáo khóa luận	

MỤC LỤC

LỜI NÓI ĐẦU.....	1
CHƯƠNG 1: TỔNG QUAN VẤN ĐỀ NGHIÊN CỨU	2
3. Đặt vấn đề:	2
4. Mục đích nghiên cứu:.....	2
5. Mục tiêu đề tài:	3
6. Đối tượng và phạm vi nghiên cứu.....	4
7. Phương pháp nghiên cứu.....	5
CHƯƠNG 2: TÌM HIỂU VỀ CƠ SỞ LÝ THUYẾT	6
1. MERN Stack	6
1.1. Định nghĩa MERN Stack:	6
1.2. Cách thức hoạt động:	7
1.3. Vai trò của MERN Stack:	7
2. Mô hình về Client – Server:	8
2.1. Định nghĩa về mô hình Client Server:	8
2.2. Nguyên tắc khi hoạt động	8
2.3. Những ưu điểm	9
2.4. Các nhược điểm	10
3. Giới thiệu về NodeJS.....	10
3.1. Định nghĩa NodeJS	10
3.2. Ưu điểm của NodeJS	11
3.3. Một số tính năng của NodeJS.....	11
4. Giới thiệu về ExpressJS	12
4.1. Định nghĩa ExpressJS	12
4.2. Một số tính năng của Express.....	12
4.3. Vai trò của Express.....	13
5. Giới thiệu về ReactJS.....	13
5.1. Định nghĩa ReactJS	13
5.2. Vai trò của ReactJS	13
5.3. Định nghĩa JSX	14

5.4.	Định nghĩa Virtual Dom.....	14
5.5.	Định nghĩa Hooks	14
5.6.	Định nghĩa Component	14
6.	Định nghĩa Redux.....	15
6.1.	Cách thức hoạt động của Redux	15
6.2.	Vai trò của Redux	16
7.	Giới thiệu về NextJS.....	17
8.	Giới thiệu về WebSocket và SocketIO.....	20
8.1.	Giao thức WebSocket.....	21
8.2.	Socket.IO	22
9.	MongoDB:	23
9.1.	Định nghĩa:	23
9.2.	Một số tính năng	23
CHƯƠNG 3: XÁC ĐỊNH VÀ MÔ HÌNH HOÁ YÊU CẦU.....		25
1.	Khảo sát hiện trạng	25
1.1.	Hiện trạng TMDT ở Việt Nam	25
1.2.	So sánh ưu/nhược của các sàn TMDT thông dụng ở Việt Nam.....	27
2.	Đặc tả chức năng	30
3.	Chức năng của các actors:	31
4.	Sơ đồ Use-case.....	34
5.	Đặc tả Use-case:	35
5.1.	Đăng ký:.....	35
5.2.	Lọc sản phẩm:	36
5.3.	Xem danh sách sản phẩm:	38
5.4.	Đặt hàng:	39
5.5.	Xem chi tiết sản phẩm:.....	42
5.6.	Tìm kiếm sản phẩm:.....	43
5.7.	Thanh toán VNPAY:	45
5.8.	Đăng nhập:	47
5.9.	Đăng xuất:	48

5.10. Quản lý địa chỉ:.....	50
5.11. Quản lý giỏ hàng:.....	52
5.12. Quản lý thông tin cá nhân:	53
5.13. Thay đổi mật khẩu:	55
5.14. Thêm review:.....	57
5.15. Quản lý sản phẩm:.....	58
5.16. Quản lý đơn hàng:	60
5.17. Xem thông báo:	62
5.18. Quản lý danh mục sản phẩm (Category):	63
5.19. Quản lý User/Shop:	65
5.20. Quản lý Report:	67
CHƯƠNG 4: THIẾT KẾ HỆ THỐNG	69
1. Kiến trúc hệ thống.....	69
2. ERD.....	70
3. Đặc tả Cơ sở dữ liệu:	71
3.1. Bảng User	71
3.2. Bảng Address	71
3.3. Bảng Classify	72
3.4. Bảng Category.....	72
3.5. Bảng Cart	72
3.6. Bảng Notifications.....	72
3.7. Bảng Order.....	73
3.8. Bảng Products	74
3.9. Bảng ProductSnapshots	75
3.10. Bảng Shop.....	75
3.11. Bảng Report	76
3.12. Bảng Reviews	76
3.13. Bảng Conversation	76
3.14. Bảng Message.....	77
3.15. Bảng Banner.....	77

4. Sequence Diagram	78
4.1. Sequence Diagram Thanh toán	78
4.1. Sequence Diagram Tạo sản phẩm.....	79
4.2. Sequence Diagram Vô hiệu hoá sản phẩm	79
4.3. Sequence Diagram Đăng nhập	80
4.4. Sequence Diagram Đăng nhập role Vendor.....	81
4.5. Sequence Diagram Hệ thống gợi ý	82
4.6. Sequence Diagram Đăng ký	83
4.7. Sequence Diagram Lấy lại mật khẩu	84
5. Thiết kế giao diện:	85
5.21. Giao diện trang Admin:	85
5.22. Giao diện trang User:	92
CHƯƠNG 5: CÀI ĐẶT VÀ KIỂM THỬ.....	97
1. Các công cụ	97
2. Các công nghệ	97
3. Cài đặt	97
4. Kiểm thử.....	97
CHƯƠNG 6: KẾT LUẬN.....	100
1. Kết quả đạt được	100
2. Ưu điểm và nhược điểm.....	101
2.1. Ưu điểm:	101
2.2. Nhược điểm:	102
3. Hướng phát triển	102
TÀI LIỆU THAM KHẢO.....	103

DANH MỤC ẢNH

Hình 1. Mô hình Client-Server	8
Hình 2. Nguyên tắc hoạt động Client-Server	9
Hình 3. Định nghĩa NodeJS	10
Hình 4. Định nghĩa ExpressJS.....	12
Hình 5. Định nghĩa ReactJS	13
Hình 6. Định nghĩa Virtual Dom.....	14
Hình 7. Định nghĩa Redux	15
Hình 8. Cách thức hoạt động của Redux.....	16
Hình 9. Định nghĩa của NextJS	17
Hình 10. Định nghĩa của SSR	18
Hình 11. Định nghĩa của CSR	19
Hình 12. Định nghĩa của SSG	19
Hình 13. Định nghĩa giao thức WebSocket	21
Hình 14. Định nghĩa của Socket.io	22
Hình 15. Định nghĩa MongoDB.....	23
Hình 16. Sàn thương mại điện tử Shopee.....	27
Hình 17. Sàn thương mại điện tử Lazada	28
Hình 18. Sàn thương mại điện tử Tiki.....	29
Hình 19. Use-case tổng quát	34
Hình 20. Kiến trúc hệ thống	69
Hình 21. ERD	70
Hình 22. Sequence Diagram Thanh Toán	78
Hình 23. Sequence Diagram Tạo sản phẩm	79
Hình 24. Sequence Diagram Vô hiệu hoá sản phẩm	79
Hình 25. Sequence Diagram Đăng nhập	80
Hình 26. Sequence Diagram Đăng nhập role Vendor.....	81
Hình 27. Sequence Diagram Hệ thống gợi ý	82
Hình 28. Sequence Diagram Đăng ký	83
Hình 29. Sequence Diagram Lấy lại mật khẩu	84

Hình 30. Sequence Diagram Thanh toán VNPay	85
Hình 31. Giao diện trang đăng nhập (Admin)	85
Hình 32. Giao diện trang chủ (Admin)	87
Hình 33. Giao diện quản lý category (Admin)	88
Hình 34. Giao diện quản lý đơn hàng (Vendor)	89
Hình 35. Giao diện quản lý báo cáo (Admin)	90
Hình 36. Giao diện quản lý shop (Admin)	91
Hình 37. Giao diện trang quản lý báo cáo (Admin)	92
Hình 38. Giao diện trang chủ (User)	92
Hình 39. Giao diện trang quản lý thông tin cá nhân (User)	93
Hình 40. Giao diện trang đăng nhập/dăng ký (User)	94
Hình 41. Giao diện trang thanh toán (User)	95
Hình 42. Giao diện trang quản lý giỏ hàng (User)	96

DANH MỤC BẢNG

Bảng 1. Đặc tả Use-case Đăng ký	36
Bảng 2. Đặc tả Use-case Lọc sản phẩm	38
Bảng 3. Đặc tả Use-case Xem danh sách sản phẩm.....	39
Bảng 4. Đặc tả Use-case Đặt hàng.....	42
Bảng 5. Đặc tả Use-case Xem chi tiết sản phẩm	43
Bảng 6. Đặc tả Use-case tìm kiếm sản phẩm.....	45
Bảng 7. Đặc tả Use-case thanh toán VNPay	47
Bảng 8. Đặc tả Use-case Đăng nhập.....	48
Bảng 9. Đặc tả Use-case Đăng xuất.....	49
Bảng 10. Đặc tả Use-case Quản lý địa chỉ	52
Bảng 11. Đặc tả Use-case Quản lý giỏ hàng	53
Bảng 12. Đặc tả Use-case Quản lý thông tin cá nhân.....	55
Bảng 13. Đặc tả Use-case Thay đổi mật khẩu.....	56
Bảng 14. Đặc tả Use-case Thêm Review	58
Bảng 15. Đặc tả Use-case Quản lý sản phẩm	60
Bảng 16. Đặc tả Use-case Quản lý đơn hàng.....	62
Bảng 17. Đặc tả Use-case Xem thông báo	63
Bảng 18. Đặc tả Use-case quản lý danh mục sản phẩm	65
Bảng 19. Đặc tả Use-case Quản lý User/Shop	67
Bảng 20. Đặc tả Use-case Quản lý shop.....	68
Bảng 21. Bảng User	71
Bảng 22. Bảng Address	72
Bảng 23. Bảng Classify.....	72
Bảng 24. Bảng Category.....	72
Bảng 25. Bảng Cart	72
Bảng 26. Bảng Notification	73
Bảng 27. Bảng Order.....	74
Bảng 28. Bảng Products	75
Bảng 29. Bảng ProductSnapshot.....	75

Bảng 30. Bảng Shop.....	75
Bảng 31. Bảng Report	76
Bảng 32. Bảng Reviews	76
Bảng 33. Bảng Conversation	76
Bảng 34. Bảng Message	77
Bảng 35. Bảng Banner.....	77
Bảng 36. Bảng Đặc tả giao diện trang đăng nhập (Admin).....	86
Bảng 37. Bảng Đặc tả giao diện trang chủ (Admin).....	88
Bảng 38. Bảng Đặc tả giao diện quản lý category (Admin).....	89
Bảng 39. Bảng Đặc tả giao diện quản lý đơn hàng (Vendor)	90
Bảng 40. Bảng Đặc tả giao diện quản lý báo cáo (Admin)	90
Bảng 41. Bảng Đặc tả giao diện quản lý shop (Admin).....	91
Bảng 42. Bảng Đặc tả giao diện trang chủ (User).....	93
Bảng 43. Bảng Đặc tả giao diện trang quản lý thông tin cá nhân (User)	94
Bảng 44. Bảng Đặc tả giao diện trang đăng nhập/đăng ký (User).....	94
Bảng 45. Bảng Đặc tả giao diện trang thanh toán (User)	96
Bảng 46. Bảng Đặc tả giao diện trang quản lý giỏ hàng (User)	96
Bảng 47. Các công cụ sử dụng.....	97
Bảng 48. Các công nghệ sử dụng.....	97
Bảng 49. Kết quả kiểm thử	99

LỜI NÓI ĐẦU

Ngày nay, với sự phát triển mạnh mẽ của CNTT và những lợi ích từ việc ứng dụng nó đi vào đời sống, máy tính và Internet đã không còn là những khái niệm xa lạ đối với mọi người, mà nó dần trở thành công cụ hữu ích hỗ trợ con người học tập, giải trí và làm việc.

Trong nền kinh tế hiện nay, với xu thế toàn cầu hóa nền kinh tế thế giới, mọi mặt của đời sống ngày càng được nâng cao, đặc biệt nhu cầu mua sắm các loại mặt hàng của con người ngày càng tăng. Môi trường Internet tự do, rộng lớn là mảnh đất màu mỡ cho các doanh nghiệp có thể quảng bá sản phẩm của mình đến người tiêu dùng một cách nhanh chóng, ít tốn kém, giúp mở rộng hoạt động kinh doanh, tăng trưởng doanh thu.

Kinh doanh online là một ví dụ điển hình, nếu không có bán hàng online các doanh nghiệp sẽ khó có cơ hội tiếp cận được với khách hàng tiềm năng, đặc biệt là những người không thường xuyên ghé thăm cửa hàng hoặc sống ở xa, thực tế như là đại dịch Covid vừa qua. Bên cạnh đó, nếu không có website, doanh nghiệp sẽ bị giới hạn trong việc quảng bá và xây dựng thương hiệu, khi mà nhiều người dùng internet sử dụng công cụ tìm kiếm để tìm kiếm thông tin sản phẩm. Việc quản lý kho hàng và vận chuyển sản phẩm. Nếu không có website, doanh nghiệp sẽ gặp khó khăn trong việc quản lý số lượng sản phẩm, lịch sử đơn hàng và thông tin khách hàng

Năm 2021, khi con người vẫn còn đang đối mặt với đại dịch Covid thì việc quảng bá các mặt hàng online đã được đẩy mạnh để đáp ứng nhu cầu mọi người, sự tiện lợi, an toàn trong mùa dịch mà không cần phải đến nơi đông người để mua sắm đã góp phần thay đổi đời sống một cách hiện đại hóa.

Với mục tiêu tìm hiểu cấu trúc, cách thức hoạt động và khái quát hóa về thương mại điện tử trên mạng Internet ở Việt Nam, nhóm chúng em đã xây dựng và đưa vào thử nghiệm thực tế một website quản lý sản phẩm thương mại điện tử trực tuyến. Đồng thời cung cấp giao diện đơn giản, bắt mắt để người dùng dễ dàng quản lý và cải tiến hệ thống, giúp việc quản lý việc bán hàng được thực hiện nhanh chóng, tránh sai sót. Đây sẽ là bước đi đầu tiên của nhóm chúng em trước khi tiếp cận với công việc quản lý website của một doanh nghiệp thực sự.

CHƯƠNG 1: TỔNG QUAN VẤN ĐỀ NGHIÊN CỨU

3. Đặt vấn đề:

Trong các năm gần đây, việc ứng dụng CNTT về xây dựng và quản lý website ngày càng phát triển mạnh mẽ, giúp gia tăng lợi nhuận cho các cửa hàng nhờ vào các trang website bán hàng trực tuyến. Hiểu được yêu cầu đó, em quyết định chọn đề tài “Xây dựng sàn thương mại điện tử” làm mục tiêu chính cho đồ án lần này, với mong muốn xây dựng một nền tảng cung cấp cho người dùng một nơi để tìm kiếm các sản phẩm theo nhu cầu cũng như cung cấp tính năng buôn bán để có thể tạo ra thu nhập cho chính họ. Ngoài ra, nền tảng cũng cung cấp khả năng trò chuyện trực tuyến giữa người bán và người mua nhằm tăng sự tiện lợi khi sử dụng dịch vụ trên nền tảng.

Một trong những công nghệ nổi bật hiện nay là Node.js. Được sáng lập bởi Ryan Dahl từ những năm 2009, Node.js và các công nghệ JavaScript liên quan như là ExpressJS đã phát triển rất mạnh để hỗ trợ các nhà phát triển các ứng dụng web hiệu quả và linh hoạt. Sự linh hoạt và hiệu suất của Node.js, cùng với các công nghệ frontend hiện đại như React.js và cơ sở dữ liệu không cấu trúc như MongoDB, đã chứng minh khả năng mạnh mẽ trong việc xây dựng các ứng dụng thương mại điện tử.

Do đó, em chọn đề tài lần này để ứng dụng công nghệ phát triển web với Node.js, Express.js, React.js và MongoDB, thực hiện xây dựng web thương mại điện tử đáp ứng theo chương trình đào tạo của Trường, đồng thời mong muốn góp phần đưa các ứng dụng về CNTT vào sản phẩm để cho việc bán hàng trực tuyến được hoàn thiện, nhận được sự hài lòng từ khách hàng.

4. Mục đích nghiên cứu:

- Nâng cao kỹ năng và kiến thức trong lĩnh vực phát triển web
- Xây dựng một nền tảng thương mại điện tử hiện đại và hiệu quả.

- Tìm hiểu được các công nghệ hỗ trợ: HTML, CSS, BOOTSTRAP, Web Service, ...
- Xây dựng trang web bán hàng giao diện chuẩn.
- Tìm hiểu về ngôn ngữ và kiến thức tương tác: Express routing, middleware, mongoose.
- Cách điều khiển dữ liệu thông qua Controller.
- Hiểu rõ về quá trình thanh toán trực tuyến, sự hỗ trợ từ các công nghệ để lưu trữ dữ liệu mua hàng của người dùng vào database.

5. Mục tiêu đề tài:

Xây dựng một sàn thương mại điện tử có các chức năng cơ bản như là tìm kiếm sản phẩm, lọc sản phẩm, đặt hàng, thanh toán và bảo mật tài khoản cá nhân. Tích hợp VNPay là dịch vụ thanh toán bên thứ ba. Cho phép người dùng đăng ký làm người bán với khả năng thêm, chỉnh sửa và phân loại sản phẩm, quản lý các đơn hàng cũng như doanh thu của cửa hàng. Sàn thương mại điện tử còn có thêm tính năng trò chuyện với các người dùng khác và cá nhân hóa trải nghiệm bằng cách thu thập các hành động trong khi sử dụng dịch vụ.

- **User:**
 - Đăng ký, đăng nhập
 - Chỉnh sửa thông tin cá nhân, địa chỉ giao hàng
 - Xem sản phẩm
 - Thêm sản phẩm vào giỏ hàng
 - Đặt hàng (thanh toán VNPay)
 - Xác nhận trạng thái đơn, hủy đơn
 - Đánh giá sản phẩm sau khi đơn hàng hoàn thành
 - Đăng ký bán hàng
 - Báo cáo sản phẩm, review sản phẩm

- Đổi mật khẩu và quên mật khẩu
- **Vendor:**
 - Quản lý thông tin shop
 - Xem số tiền trong tài khoản
 - Quản lý sản phẩm
 - Quản lý danh mục shop
 - Quản lý đơn hàng
 - Xem thông báo (đơn hàng, review)
- **Admin:**
 - Xem sản phẩm
 - Vô hiệu hóa sản phẩm, review không phù hợp
 - Quản lý danh mục sản phẩm (category)
 - Quản lý user, shop
 - Quản lý các banner của sàn
 - Xem thông báo (tổ cáo)
- **Tất cả:**
 - Chat giữa các người dùng

6. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu: Các hoạt động và quy trình liên quan đến việc quản lý và tối ưu hóa sàn thương mại điện tử với ba vai trò chính: admin, người bán, và người dùng. Đối tượng này bao gồm các doanh nghiệp vừa và nhỏ cũng như các startup mới trong lĩnh vực thương mại điện tử tại Việt Nam.

Khách thể nghiên cứu: Các sàn thương mại điện tử hiện đang hoạt động tại Việt Nam. Các nền tảng này đóng vai trò là "không gian" cho việc thực hiện và đánh giá các hoạt động tối ưu hóa.

Phạm vi nghiên cứu:

- **Phạm vi không gian:** Các sản phẩm thương mại điện tử ở Việt Nam, chủ yếu là các doanh nghiệp nhỏ và vừa và startup.
- **Phạm vi thời gian:** Đề tài được phát triển từ tiểu luận chuyên ngành trong 3 tháng, từ tháng 3 đến hết tháng 5 của năm 2024.
- **Phạm vi nội dung:** Tập trung vào các chiến lược tối ưu hóa cho ba vai trò chính trên sàn thương mại điện tử:
 - **Admin:** Tối ưu hóa quy trình quản lý, giám sát và duy trì hoạt động của sàn.
 - **Người bán:** Cải tiến các tính năng hỗ trợ việc đăng bán sản phẩm, quản lý đơn hàng, và chăm sóc khách hàng.
 - **Người dùng:** Nâng cao trải nghiệm mua sắm, bao gồm giao diện người dùng thân thiện, tìm kiếm và gợi ý sản phẩm hiệu quả, và quy trình thanh toán tiện lợi.

Mục đích: Phát triển các giải pháp tối ưu hóa hiệu quả, có khả năng áp dụng ngay cả trong bối cảnh nguồn lực hạn chế, nhằm đảm bảo sự ổn định và phát triển bền vững cho các sản phẩm thương mại điện tử tại Việt Nam. Điều này không chỉ cải thiện trải nghiệm người dùng mà còn tăng cường hiệu quả kinh doanh của các doanh nghiệp tham gia trên sàn.

7. Phương pháp nghiên cứu

Để thực hiện đề tài nghiên cứu, nhóm đã sử dụng các phương pháp sau:

- Phương pháp nghiên cứu tài liệu, các tài liệu tham khảo, các bài viết, các video liên quan đến MERN Stack của một số trang web Ecommerce
- Phương pháp chuyên gia: nhờ sự góp ý, tư vấn và hỗ trợ từ giảng viên hướng dẫn và các thầy cô trong khoa Đào tạo chất lượng cao trường Đại học Sư phạm Kỹ thuật TP. HCM
- Phương pháp làm việc nhóm: kết hợp làm việc nhóm hiệu quả, phân chia công việc theo kế hoạch.

CHƯƠNG 2: TÌM HIỂU VỀ CƠ SỞ LÝ THUYẾT

1. MERN Stack

1.1. Định nghĩa MERN Stack:

MERN Stack là một bộ công nghệ phát triển web toàn diện, bao gồm bốn thành phần chính: MongoDB, Express.js, React.js, và Node.js. Các công nghệ này kết hợp với nhau để tạo ra các ứng dụng web đầy đủ chức năng, từ phía máy chủ đến phía người dùng.

- **MongoDB:**
 - **Cách thức hoạt động:** MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL, lưu trữ dữ liệu dưới dạng JSON-like documents. Điều này giúp cho việc lưu trữ và truy xuất dữ liệu trở nên linh hoạt và dễ dàng mở rộng.
 - **Vai trò:** Lưu trữ và quản lý dữ liệu của ứng dụng. Nó cung cấp một cơ sở dữ liệu phi cấu trúc, giúp xử lý dữ liệu lớn và phức tạp một cách hiệu quả.
- **Express.js:**
 - **Cách thức hoạt động:** Express.js là một framework cho Node.js, giúp xây dựng các ứng dụng web và API mạnh mẽ và linh hoạt. Nó cung cấp một bộ công cụ phong phú để xử lý các yêu cầu HTTP và quản lý các middlewares.
 - **Vai trò:** Tạo và quản lý các API và định tuyến của ứng dụng. Nó giúp xử lý các yêu cầu từ phía người dùng và tương tác với cơ sở dữ liệu MongoDB.
- **React.js:**
 - **Cách thức hoạt động:** React.js là một thư viện JavaScript phục vụ cho việc xây dựng giao diện người dùng. Nó sử dụng cách tiếp cận dựa trên thành phần (component-based) để tạo ra các UI động và hiệu quả.
 - **Vai trò:** Tạo ra giao diện người dùng phía frontend. React.js giúp xây dựng các ứng dụng web tương tác, nhanh chóng và dễ dàng quản lý trạng thái.

- **Node.js:**

- **Cách thức hoạt động:** Node.js là một môi trường chạy JavaScript phía server, cho phép xây dựng các ứng dụng nhanh chóng nhờ các gói thư viện lớn và có khả năng mở rộng. Nó sử dụng mô hình sự kiện không đồng bộ, giúp xử lý hàng ngàn kết nối đồng thời.
- **Vai trò:** Chạy mã JavaScript trên máy chủ, phục vụ các yêu cầu từ phía client, và quản lý toàn bộ logic phía backend của ứng dụng.

1.2. Cách thức hoạt động:

- **Frontend (React.js):** Người dùng tương tác với giao diện người dùng được xây dựng bằng React.js. Khi người dùng thực hiện các hành động như nhấp chuột hoặc nhập liệu, React sẽ cập nhật giao diện tương ứng và gửi các yêu cầu API đến server.
- **Backend (Express.js và Node.js):** Server nhận các yêu cầu từ frontend thông qua Express.js. Node.js xử lý các yêu cầu này, thực hiện các thao tác cần thiết, và truy xuất dữ liệu từ MongoDB nếu cần.
- **Database (MongoDB):** Dữ liệu được lưu trữ và truy xuất từ MongoDB. Các thao tác CRUD (Create, Read, Update, Delete) được thực hiện thông qua các API được định nghĩa bởi Express.js.
- **Response:** Sau khi xử lý xong, server sẽ gửi phản hồi trở lại cho frontend. React.js sẽ cập nhật giao diện người dùng dựa trên phản hồi này.

1.3. Vai trò của MERN Stack:

- **Phát triển nhanh:** Sử dụng JavaScript cho cả frontend và backend giúp giảm bớt thời gian học tập và phát triển.
- **Hiệu suất cao:** Các thành phần của MERN Stack được thiết kế để làm việc cùng nhau, tối ưu hóa hiệu suất và khả năng mở rộng của ứng dụng.
- **Quản lý dữ liệu hiệu quả:** MongoDB cung cấp một cơ sở dữ liệu linh hoạt và mạnh mẽ, phù hợp cho các ứng dụng hiện đại với dữ liệu không cấu trúc.

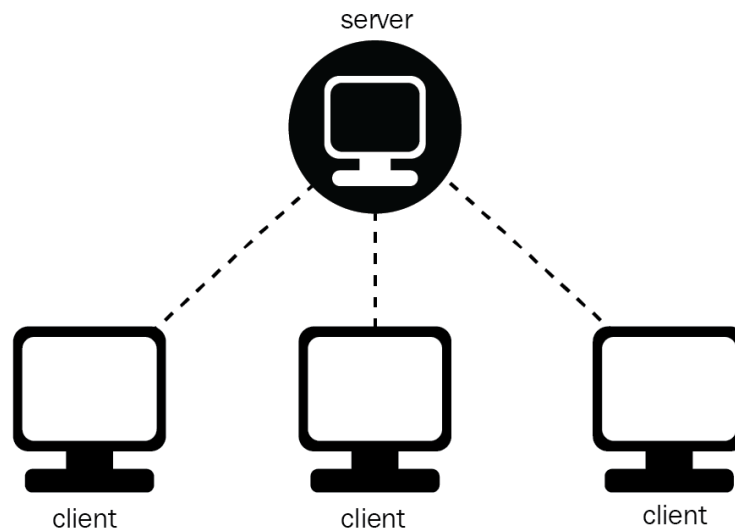
- **Tương tác người dùng tốt:** React.js giúp tạo ra các giao diện người dùng hiện đại và phản hồi nhanh, cải thiện trải nghiệm người dùng.

MERN Stack là một lựa chọn phổ biến cho việc phát triển các ứng dụng web hiện đại, giúp tiết kiệm thời gian và công sức cho các nhà phát triển trong quá trình xây dựng, triển khai và bảo trì ứng dụng.

2. Mô hình về Client – Server:

2.1. Định nghĩa về mô hình Client Server:

Mô hình Client Server là mô hình mạng máy tính trong đó các máy tính con sẽ đóng vai trò như một máy khách, chúng có trách nhiệm gửi yêu cầu đến cho máy chủ. Để máy chủ tiếp nhận yêu cầu và trả kết quả về các máy khách đó. [1]

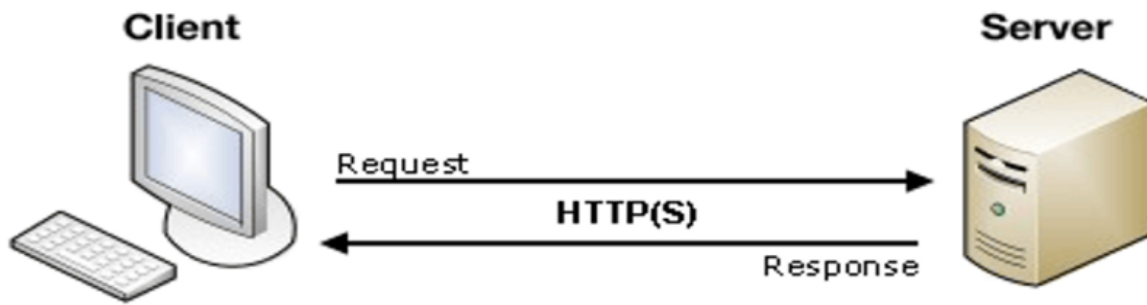


Hình 1. Mô hình Client-Server

2.2. Nguyên tắc khi hoạt động

Trong mô hình Client Server, server chấp nhận toàn bộ các yêu cầu đến từ mọi máy tính khác nhau trên Internet, từ đó trả kết quả cho máy tính đã gửi yêu cầu hợp lệ. [1]

Máy tính được gọi là máy khách khi chúng có trách nhiệm chuyển thông tin đến cho máy chủ và chờ câu trả lời được gửi về. [1]



Hình 2. Nguyên tắc hoạt động Client-Server

Để máy khách và máy chủ có thể giao tiếp được với nhau thì giữa chúng phải có một chuẩn nhất định, và chuẩn ấy được gọi là giao thức. Một số giao thức được dùng phổ biến hiện nay bao gồm: HTTPS, TCP/IP, FTP, ...

Nếu máy khách muốn có được thông tin từ máy chủ, chúng phải tuân thủ theo một giao thức mà máy chủ đã đặt ra. Nếu yêu cầu đã được chấp nhận thì máy chủ sẽ thu thập thông tin và trả về kết quả mà máy khách yêu cầu. Bởi vì Server - máy chủ luôn luôn trong tình trạng sẵn sàng để tiếp nhận request từ client cho nên chỉ cần client gửi yêu cầu dữ liệu và chấp nhận yêu cầu đó thì server sẽ trả kết quả cho phía client trong thời gian nhanh nhất.

2.3. Những ưu điểm

- Giúp chúng ta có thể chạy trên bất cứ một máy tính nào có hỗ trợ giao thức truyền thông. Giao thức chuẩn này cũng giúp cho nhà sản xuất cài đặt lên nhiều thiết bị khác nhau mà không vấp phải vấn đề gì.
- Có thể có nhiều server cùng chạy một dịch vụ, chúng có thể làm việc trên nhiều máy tính hoặc một máy tính.
- Chỉ mang đặc tính của phần mềm chứ không cần quan tâm đến cấu hình, với điều kiện duy nhất là server cần có cấu hình cao hơn các client.
- Hỗ trợ người dùng nhiều tính năng hơn và rất tiện lợi nhờ tính năng truy cập từ xa.
- Cung cấp một nền tảng lý tưởng, có thể cung cấp tích hợp các công nghệ tiên tiến như mô hình kiến trúc hướng đối tượng, hệ chuyên gia, hệ thông tin địa lý (GIS).

2.4. Các nhược điểm

- Vấn đề trao đổi dữ liệu thông tin đôi lúc vẫn không được an toàn. Vì còn phải truyền dữ liệu giữa 2 máy tính khác nhau ở 2 khu vực địa lý khác xa nhau.
- Và đây cũng chính là nhược điểm duy nhất của mô hình này. Tuy nhiên vấn đề bảo mật thì có một vài giao thức đã giúp bảo vệ dữ liệu khi truyền tải. Giao thức được sử dụng rộng rãi nhất là HTTPS.

3. Giới thiệu về NodeJS

3.1. Định nghĩa NodeJS



Hình 3. Định nghĩa NodeJS

Nodejs là một nền tảng được xây dựng trên V8 JavaScript Engine từ năm 2009, tài trợ bởi công ty Joyent có trụ sở tại California. Đây là một trình thông dịch thực thi mã JavaScript giúp phát triển các ứng dụng web như video clip và diễn đàn, nhằm giúp triển khai hiệu quả và nhanh chóng cũng như thu hẹp phạm vi hoạt động của các trang mạng xã hội. Ý tưởng chính của NodeJS là thực hiện non – blocking, nhanh chóng định tuyến đầu vào và đầu ra dữ liệu cho các tác vụ thời gian thực. Bởi vì NodeJS có khả năng mở rộng tốt và cho phép thực hiện một số lượng lớn các tác vụ song song với hiệu suất cao. Đa số các ứng dụng web truyền thống sử dụng tài nguyên không hiệu quả khi các yêu cầu phát sinh thêm các luồng xử lý mới làm tiêu tốn các nguồn tài nguyên của hệ thống. Do đó NodeJS đã đưa ra giải pháp là sử dụng luồng đơn (với tên tiếng anh là Single – Threaded), đây là sự kết hợp single – blocking I/O nhằm thực hiện xử lý các yêu cầu, điều này cho phép hệ thống xử lý nhiều kết nối đồng thời.

Hầu hết các ứng dụng web truyền thống sử dụng tài nguyên hệ thống không hiệu quả khi các yêu cầu sinh thêm các luồng xử lý yêu cầu mới và tiêu tốn nhiều tài nguyên của hệ

thống. Vì vậy giải pháp mà NodeJS đưa ra chính là sử dụng luồng đơn có tên tiếng anh là Single – Threaded, đây là sự kết hợp single – blocking I/O nhằm thực hiện xử lý các yêu cầu, điều này cho phép hệ thống xử lý hàng chục hàng nghìn kết nối đồng thời.

3.2. Ưu điểm của NodeJS

- IO hướng sự kiện bất đồng bộ, cho phép xử lý nhiều yêu cầu đồng thời.
- Sử dụng JavaScript – một ngôn ngữ lập trình dễ hiểu, dễ dàng lập trình.
- Chia sẻ cùng code ở cả phía trình duyệt và máy chủ.
- NPM (Node Package Manager) hỗ trợ nhiều thư viện mạnh mẽ cho nhiều mục đích khác nhau.
- Cộng đồng hỗ trợ tích cực.
- Cho phép stream các file có kích cỡ lớn

3.3. Một số tính năng của NodeJS

- Lập trình hướng sự kiện và bất đồng bộ: Tất cả API trong thư viện NodeJS sẽ bất đồng bộ, nhưng không bị chặn. Về cơ bản điều này có nghĩa là một server sử dụng NodeJS sẽ không phải chờ một API trả về dữ liệu. Server sẽ chuyển sang API kế tiếp từ lần gọi API đầu tiên vì cơ chế thông báo của Events trong NodeJS giúp server nhận được thông báo từ lần gọi API trước.
- Rất linh hoạt: Vì NodeJS được dựng trên Công cụ JavaScript V8 của Google Chrome, do đó tốc độ thực thi là rất cao.
- Đơn luồng (Single thread) nhưng cung cấp khả năng mở rộng cao: NodeJS cung cấp mô hình đơn luồng dựa trên vòng lặp sự kiện (event). Cơ chế này cho phép máy chủ phản hồi non-blocking và đồng thời cho khả năng mở rộng cao so với các server thông thường do hỗ trợ nhiều thread để thực thi tác vụ. NodeJS sử dụng một chương trình đơn luồng, vì vậy chương trình có thể thực thi công việc trên một số lượng yêu cầu lớn hơn so với các ứng dụng thông thường như Apache HTTP Server.
- Không có buffer: Các ứng dụng NodeJS không có vùng nhớ tạm thời (buffer) đối với bất cứ dữ liệu nào. Các ứng dụng thông thường chỉ đơn thuần là xuất dữ liệu theo yêu cầu.

- License: NodeJS được xuất bản theo giấy phép MIT.

4. Giới thiệu về ExpressJS

4.1. Định nghĩa ExpressJS



Hình 4. Định nghĩa ExpressJS

ExpressJS là một framework mã nguồn mở miễn phí dùng cho mục đích xây dựng ứng dụng web với nền tảng Node.js. Người dùng có thể dễ dàng thiết kế và phát triển các ứng dụng web và API một cách dễ dàng đặc biệt với các lập trình viên đã làm quen với JavaScript.

4.2. Một số tính năng của Express

- Phát triển máy chủ nhanh chóng: ExpressJS cung cấp nhiều chức năng dưới dạng các hàm giúp có thể chạy ở bất cứ đâu trong chương trình. Điều này giảm bớt thời gian khi viết code và từ đó tiết kiệm được thời gian.
- Phần mềm trung gian Middleware: Là các chức năng trung gian trong stack của api, các chức năng này được định nghĩa dưới dạng hàm và có chức năng hỗ trợ cho nhau trong stack api như là tiền xử lý dữ liệu hoặc xác thực quyền truy cập của người dùng khi truy cập vào api.
- Định tuyến – Routing: ExpressJS cung cấp cơ chế định tuyến để duy trì hoạt động của website với sự trợ giúp của URL - Tạo mẫu – Templating: Các công cụ tạo dựng khuôn mẫu mà ExpressJS cung cấp giúp các nhà tạo nội dung trên các website bằng việc tạo lập các mẫu HTML phía máy chủ..
- Gỡ lỗi – Debugging: Việc xây dựng thành công của các ứng dụng web không thể nào thiếu được việc gỡ lỗi. Việc gỡ lỗi đã trở nên đơn giản hơn bao giờ hết với ExpressJS bởi nó tìm đúng từng thành phần trong ứng dụng web có vấn đề.

4.3. Vai trò của Express

- Rất dễ tiếp cận đối với lập trình viên front-end vì sử dụng đồng nhất một ngôn ngữ là JavaScript.
- Với các chức năng cung cấp sẵn của ExpressJS thì việc thiết kế back-end sẽ dễ dàng hơn.
- ExpressJS rất đơn giản để tùy chỉnh và sử dụng theo yêu cầu.
- ExpressJS có hỗ trợ xây dựng ứng dụng web theo mô hình MVC - Xây dựng các các lớp trung gian phục vụ việc trả về của các yêu cầu HTTP.
- Thiết kế bộ định tuyến cho phép sử dụng với các chức năng khác nhau dựa trên phương thức HTTP và URL.
- Cho phép trả về các trang HTML dựa trên các tham số

5. Giới thiệu về ReactJS

5.1. Định nghĩa ReactJS



Hình 5. Định nghĩa ReactJS

React là một thư viện JavaScript nguồn mở và miễn phí phục vụ cho việc xây dựng phát triển giao diện người dùng. Được tạo ra bởi Meta (trước đây là Facebook), ReactJS được dùng được sử dụng rộng rãi trong các trang web có yêu cầu cao về giao diện người dùng, hiển thị nhanh và tái sử dụng tốt.

5.2. Vai trò của ReactJS

React là một thư viện JavaScript xây dựng giao diện người dùng hoặc UIs. Về trang web và ứng dụng web, giao diện người dùng là tập hợp các menu trên màn hình, thanh tìm kiếm, nút và các thành phần khác mà người dùng có thể tương tác.

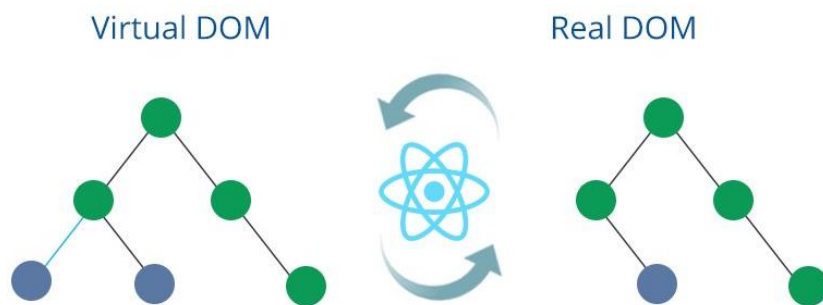
Ngoài ra, React có thể dễ dàng tái sử dụng (tiết kiệm thời gian phát triển và giảm nguy cơ xảy ra lỗi), React đi kèm với hai tính năng chính:

- JSX
- Virtual Dom

5.3. Định nghĩa JSX

JSX là cú pháp mở rộng của Javascript giúp dễ dàng sửa đổi DOM bằng cách sử dụng mã HTML đơn giản. Đồng thời JSX còn tương thích với hầu hết các nền tảng trình duyệt đang sử dụng.

5.4. Định nghĩa Virtual Dom



Hình 6. Định nghĩa Virtual Dom

Virtual DOM là một định dạng dữ liệu của Javascript giúp thể hiện nội dung của DOM tại một thời điểm nhất định. Khi trạng thái của đối tượng thay đổi, Virtual DOM chỉ thay đổi đối tượng đó trong DOM thực thay vì cập nhật lại tất cả các đối tượng trong DOM.

5.5. Định nghĩa Hooks

Hooks là một cập nhật mới ở React 16.8. Chúng cho phép sử dụng trạng thái và các tính năng khác của React mà không cần phải viết một lớp.

5.6. Định nghĩa Component

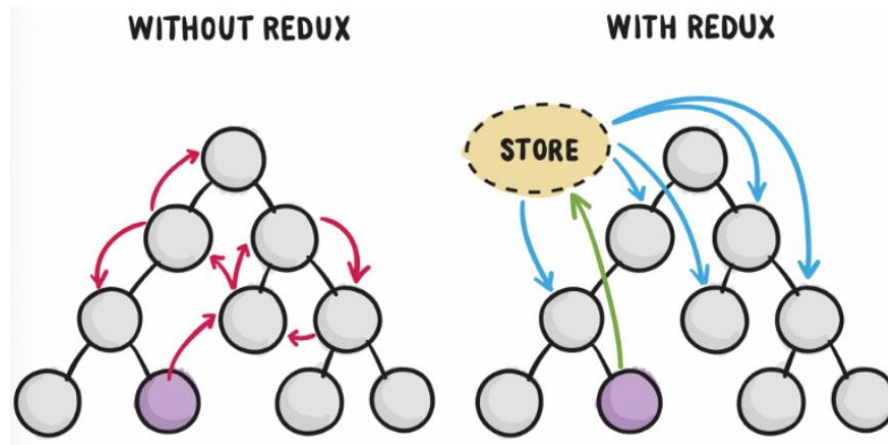
Component là các thành phần độc lập của giao diện và có thể tái sử dụng lại. Các component được dùng bằng cách gọi tên tương tự các thẻ HTML và nhiệm vụ chính là trả về HTML thông qua hàm render. Có 2 loại component trong ReactJS:

- Function Component

- Class Component

6. Định nghĩa Redux

Redux là một công cụ cho phép quản lý state trong ứng dụng Javascript với khả năng hỗ trợ viết các ứng dụng hoạt động một cách nhất quán, linh hoạt trong các môi trường khác nhau (client, server, native) và dễ dàng test.

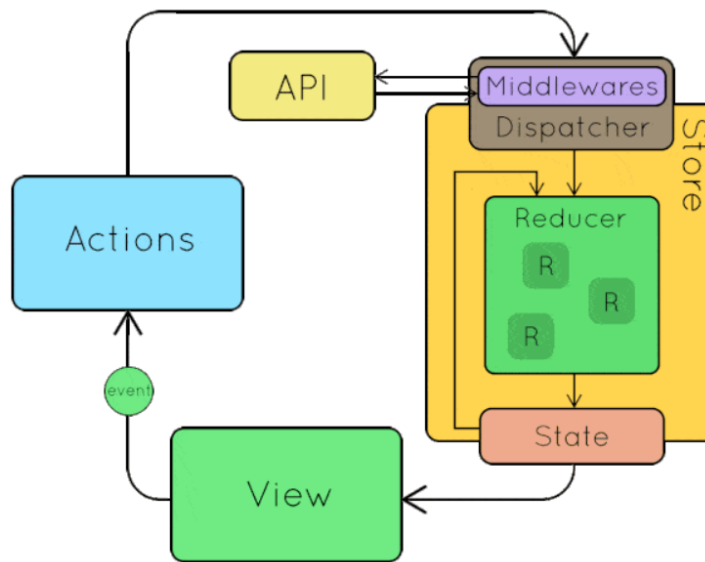


Hình 7. Định nghĩa Redux

Redux là một công cụ cho phép quản lý state trong ứng dụng Javascript với khả năng hỗ trợ viết các ứng dụng hoạt động một cách nhất quán, linh hoạt trong các môi trường khác nhau (client, server, native) và dễ dàng test.

6.1. Cách thức hoạt động của Redux

Cách thức hoạt động của Redux: Có một store lưu giữ toàn bộ state của ứng dụng. Mỗi component có thể truy cập state được lưu trữ mà không cần phải gửi các props từ component này sang component khác.



Hình 8. Cách thức hoạt động của Redux

Redux hoạt động dựa vào 3 thành phần cơ bản: Actions, Reducers và Store.

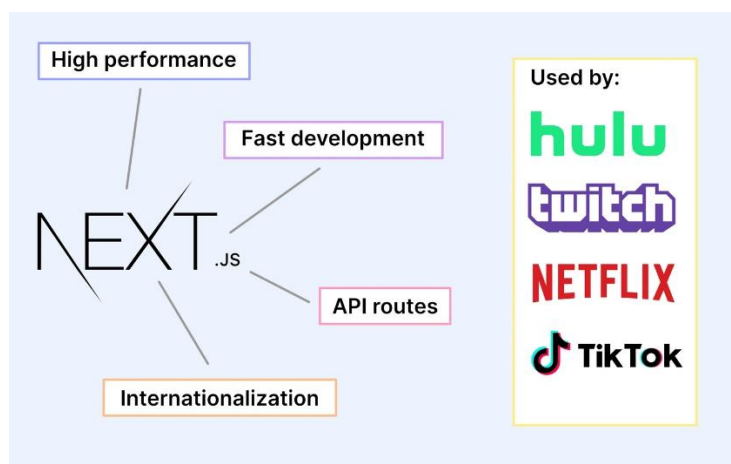
- Action: các events giúp gửi dữ liệu từ ứng dụng tới store. Các tác vụ được gửi bằng phương thức `store.dispatch()`. Các actions là các đối tượng JavaScript và chúng phải có type để chỉ ra loại action sẽ được thực hiện. Chúng cũng phải có một payload chứa thông tin cần được thực hiện bởi action.
- Reducer: các function lấy state hiện tại của ứng dụng, thực hiện một action và trả về state mới. Các states này được lưu trữ dưới dạng các đối tượng và chúng chỉ định cách state của ứng dụng thay đổi để đáp ứng với một action được gửi đến store.
- Store: lưu trữ các states của ứng dụng. Các actions được thực hiện trên state luôn trả về state mới. Như vậy, state rất dễ dự đoán trước.

6.2. Vai trò của Redux

- Hỗ trợ dự đoán và quản lý các states
- Dễ dàng bảo trì, kiểm thử và gỡ lỗi
- Tối ưu hóa hiệu suất giúp các components được kết nối dễ dàng và hiển thị ngay khi thực hiện
- Giữ state trong bộ nhớ cục bộ của ứng dụng và khôi phục dễ dàng
- Xử lý render ban đầu của ứng dụng bằng cách gửi state đến server cùng với phản hồi của nó

7. Giới thiệu về NextJS

NextJS là một framework được xây dựng trên nền tảng của React. NextJs cho phép chúng ta xây dựng các trang web tĩnh có hiệu suất cao và thân thiện với người dùng, cũng như phát triển các ứng dụng React hiện đại. NextJs ra đời vào năm 2016 và thuộc quyền sở hữu của Vercel, nó được biết đến rộng rãi vào năm 2018 và được trọng dụng trong cộng đồng các nhà phát triển web. Sự kết của các tính năng như Server-side Rendering (SSR) và Static Site Generation (SSG) đã giúp NextJS trở thành một trong các lựa chọn hàng đầu của nhiều dự án phát triển ứng dụng web nổi tiếng.



Hình 9. Định nghĩa của NextJS

- Các tính năng chính của NextJS

• Routing trong NextJS

1. Automatic Routing

NextJS tự động cấu hình các router dựa vào cấu trúc thư mục. Ví dụ, nếu tạo file `home.js` trong thư mục `pages` thì NextJS sẽ tạo router `/home`.

2. Nested Routing

NextJs cho phép tạo các thư mục con trong thư mục `pages` thể hiện cho cấu trúc cây của bộ định tuyến

3. Dynamic Routes

Bạn có thể tạo các router động bằng cách sử dụng dấu `[]` trong tên file. Ví dụ, đường dẫn `pages/blog/[slug].js` sẽ tạo ra các router như `/blog/blog-dau-tien` hoặc `/blog/blog-thu-hai`, với `slug` là giá trị động.

4. Link Component

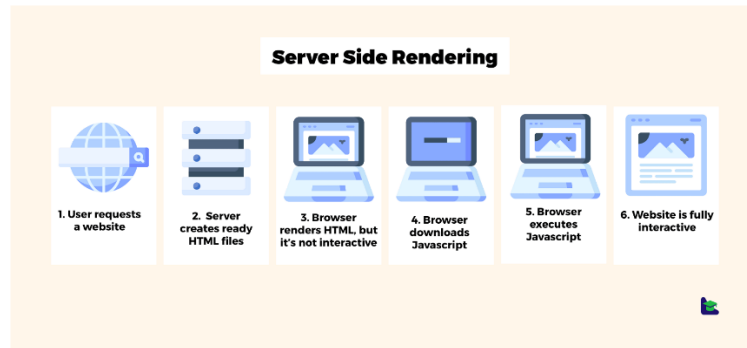
Sử dụng component Link từ thư viện next/link để xây dựng các tương tác điều hướng thay cho thẻ a nhằm tránh việc tải lại trang gây tốn kém tài nguyên không cần thiết.

5. Query Parameters

Truyền dữ liệu giữa các trang bằng cách sử dụng tham số query parameters trong router.

- Rendering trong NextJS

1. Server-side Rendering (SSR)



Hình 10. Định nghĩa của SSR

SSR xử lý logic phức tạp ở phía server. Khi user truy cập vào một trang web, trình duyệt gửi request tới server. Server xử lý request, truy cập dữ liệu trong database, render HTML và trả về cho browser. Browser nhận HTML, tải xuống và render ra UI, sau đó tải và thực thi JS.

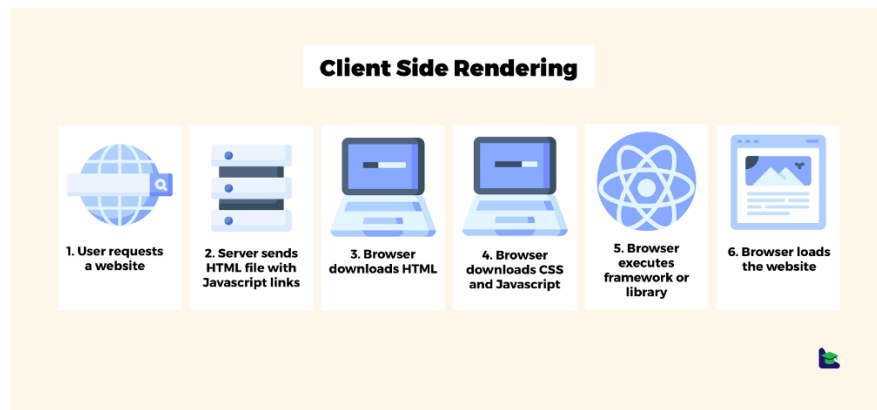
Ưu điểm của SSR:

- Load nhanh: Dữ liệu được xử lý ở phía server, client chỉ cần hiển thị.
- Tốt cho SEO: Dữ liệu được render dưới dạng HTML, dễ dàng cho bot của Google quét.
- Đơn giản hóa dự án: Một project bao gồm cả Frontend và Backend.

Nhược điểm của SSR:

- Server quá tải: Xử lý nhiều dữ liệu dẫn đến quá tải.
- Chuyển trang chậm: Mỗi lần chuyển trang cần tải lại toàn bộ.

2. Client-side Rendering (CSR)



Hình 11. Định nghĩa của CSR

CSR thực thi việc render HTML ở phía client, hay còn gọi là Single Page App (SPA).

Ưu điểm của CSR:

- Giảm tải cho server: Xử lý dữ liệu chuyển sang client.
- Trải nghiệm tốt hơn: Trang chỉ cần load một lần duy nhất, chuyển trang nhanh hơn.

Nhược điểm của CSR:

- Load trang lần đầu chậm hơn: Browser phải tải toàn bộ HTML và JS.
- Yêu cầu chia dự án thành 2: Backend và Frontend.
- Không chạy nếu tắt JavaScript: Website không hoạt động nếu JS bị tắt.

SEO kém hơn SSR: Dữ liệu được render bởi JS, không tốt cho SEO.

3. SSR trong NextJS

Khi user truy cập trang SSR của NextJS, trình duyệt sẽ gửi yêu cầu đến server. NextJS sẽ chạy hàm `getServerSideProps()` để lấy dữ liệu, sau đó render phiên bản đầy đủ và trả về browser dưới dạng HTML.

- Static Site Generation (SSG)

Static Generation

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



Hình 12. Định nghĩa của SSG

SSG tạo các trang tĩnh và lưu chúng dưới dạng file HTML tĩnh. Khi chạy lệnh `next build`, NextJS chạy hàm `getStaticProps` để lấy dữ liệu, tạo phiên bản đã render và lưu trong thư mục `.next`.

- Styling trong NextJS

CSS Modules

Bạn có thể tạo file CSS/SCSS riêng của từng component hoặc xây dựng file chung cho cả dự án.

- CSS-in-JS Libraries

Sử dụng thư viện CSS-in-JS như `Styled-Components` để viết trực tiếp CSS vào code.

- CSS Frameworks

NextJS hỗ trợ sử dụng các CSS framework như `TailwindCSS`, `Bootstrap` hoặc `MaterialUI`.

Ưu điểm của NextJS

- Sử dụng SSR và SSG: Cải thiện tốc độ tải trang và khả năng SEO.
- Tính năng tối ưu hóa hiệu suất: `Code Splitting`, `Lazy Loading`, `Image Optimization`,...
- `Fast Refresh`: Tự động làm mới giao diện mà không cần tải lại toàn bộ trang.
- Quản lý CSS hiệu quả: Tự động tạo file CSS riêng cho mỗi trang.
- Hỗ trợ TypeScript: Cải thiện tính rõ ràng cho code và thuận tiện cho việc debug.
- Cộng đồng lớn: Nhiều tài liệu phong phú và plugin hữu ích.
- Hệ sinh thái mạnh mẽ: Kết hợp tốt với các thư viện và công cụ như `Redux`, `React Query`, `Apollo Client`.
- Tích hợp tốt với React: Dễ dàng làm quen nếu đã tiếp cận React từ trước.

Nhược điểm của NextJS

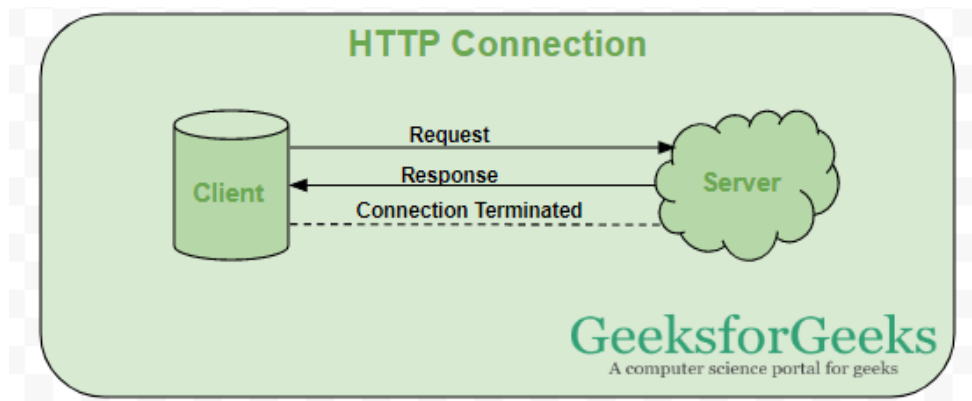
- Khó học cho người mới: Cần hiểu biết cơ bản về Web, JavaScript và React.
- Khó tích hợp với một số thư viện bên ngoài: Cần điều chỉnh hoặc tùy chỉnh.
- Phụ thuộc vào React: Không thích hợp nếu không muốn sử dụng React.
- Yêu cầu server NodeJS: Tăng chi phí và phức tạp trong triển khai.
- Cấu trúc dự án phức tạp: Khó quản lý trong các dự án lớn nếu không cẩn thận.

8. Giới thiệu về WebSocket và SocketIO

8.1. Giao thức WebSocket

WebSocket là một giao thức giúp truyền dữ liệu hai chiều giữa server và client qua một kết nối TCP duy nhất. WebSocket được thiết kế để truyền dữ liệu bằng cách sử dụng các cổng 80 và 443, là một phần của HTML5. Do đó, WebSocket có thể hoạt động trên các cổng web tiêu chuẩn, giúp tránh được rắc rối khi mở cổng cho các ứng dụng và lo lắng về việc bị chặn bởi các tường lửa hay proxy server.

Khác với giao thức HTTP, yêu cầu client phải gửi yêu cầu đến server và chờ đợi phản hồi, giao thức WebSocket cho phép server chủ động gửi thông tin về client mà không cần phải có yêu cầu từ trước của client.



Hình 13. Định nghĩa giao thức WebSocket

- **Quy trình của WebSocket:**

WebSocket có hai phần chính: Bắt tay và truyền dữ liệu. Ban đầu, client gửi yêu cầu khởi tạo kết nối WebSocket đến server. Server kiểm tra và gửi trả kết quả chấp nhận kết nối, sau đó kết nối được tạo lập và quá trình truyền dữ liệu có thể được thực hiện. Dữ liệu được gửi qua các khung dữ liệu (Ws frame).

- **Ưu điểm của WebSocket**

Truyền dữ liệu nhanh chóng và liên tục: Tất cả dữ liệu giữa client và server được gửi qua một kết nối cố định, giúp giảm độ trễ do không cần phải chờ đợi yêu cầu từ client.

Tương tác thời gian thực: Server có thể gửi dữ liệu đến client bất cứ lúc nào, giúp tăng tốc độ phản hồi và giảm độ trễ.

Sử dụng cổng tiêu chuẩn: Hoạt động trên cổng 80 và 443, giúp tránh các vấn đề liên quan đến tường lửa và proxy.

- **Yêu cầu**

Để sử dụng WebSocket, trình duyệt phải hỗ trợ và cần có server WebSocket. Server này có thể được tạo ra bằng bất kỳ ngôn ngữ server-side nào, nhưng Node.js thường được sử dụng rộng rãi nhờ viết bằng JavaScript, mang lại nhiều ưu điểm so với các ngôn ngữ server-side truyền thống.

8.2. Socket.IO

Socket.IO là một thư viện trong Node.js. Ra đời năm 2010, Socket.IO tạo ra môi trường giao tiếp thời gian thực và hai chiều giữa máy chủ và máy khách. Socket.IO hỗ trợ nhiều ngôn ngữ như PHP, ASP.NET, và Node.js.



Hình 14. Định nghĩa của Socket.io

- Ưu điểm của Socket.IO

- Nâng cao tính bảo mật:

Bảo mật dựa trên TLS/SSL: Hỗ trợ kết nối an toàn qua HTTPS và WSS (WebSocket over TLS/SSL), mã hóa dữ liệu trên đường truyền.

Hỗ trợ cân bằng tải và proxy: Dễ dàng tích hợp với cơ chế cân bằng tải và proxy để tăng cường bảo mật và khả năng chịu tải.

Hỗ trợ cho tường lửa: Sử dụng các cổng phổ biến như cổng 443 để vượt qua tường lửa và phần mềm chống virus.

- Chế độ kết nối tự động:

Tự động kết nối lại: Khi mất kết nối, Socket.IO sẽ tự động cố gắng kết nối lại đến server cho đến khi thành công hoặc đạt đến thời gian chờ đặt trước.

- Công nghệ mã hóa nhị phân:

Trình duyệt: Hỗ trợ truyền dữ liệu dạng ArrayBuffer và Blob, cho phép truyền hình ảnh, video, hoặc âm thanh.

Node.js: Hỗ trợ truyền dữ liệu dạng Buffer, cho phép truyền file và dữ liệu cấu trúc.

Tiện ích tạo kênh và phòng:

Kênh (namespace): Tạo ra các kênh phân tán để quản lý thông tin tương tác giữa server và client một cách riêng biệt.

Phòng (room): Hỗ trợ tạo phòng để nhóm các client lại với nhau, gửi thông báo hoặc dữ liệu đến toàn bộ nhóm một cách hiệu quả.

Socket.IO giúp xây dựng các ứng dụng đòi hỏi tốc độ phản hồi cao như trò chơi trực tuyến, chat trực tuyến, và cập nhật thông tin thời gian thực, mang lại một môi trường giao tiếp hai chiều linh hoạt và mạnh mẽ.

9. MongoDB:

9.1. Định nghĩa:

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở đa nền tảng viết bằng C++. Bản ghi trong MongoDB được lưu trữ dạng một dữ liệu văn bản (Document), là một cấu trúc dữ liệu bao gồm các cặp giá trị và trường tương tự như các đối tượng JSON.[5] MongoDB được phát triển bởi MongoDB Inc. dưới dạng giấy phép Server Side Public License (SSPL)



Hình 15. Định nghĩa MongoDB

9.2. Một số tính năng

- Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ.
- Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.

- MongoDB hỗ trợ một số công cụ lưu trữ và cung cấp API công cụ lưu trữ có thể cắm được (pluggable storage engine API) cho phép các bên thứ ba phát triển công cụ lưu trữ dữ liệu riêng

CHƯƠNG 3: XÁC ĐỊNH VÀ MÔ HÌNH HOÁ YÊU CẦU

1. Khảo sát hiện trạng

1.1. Hiện trạng TMĐT ở Việt Nam

Thương mại điện tử (TMĐT) đang ngày càng khẳng định vai trò chủ đạo trong tiến trình phát triển kinh tế số của Việt Nam, mở ra một viễn cảnh mới mẻ cho hoạt động kinh doanh và nhu cầu tiêu dùng của người dân.

- **Nền tảng cho sự bùng nổ:**

- **Cấu trúc dân số trẻ:** Việt Nam sở hữu dân số trẻ năng động, am hiểu công nghệ, tạo điều kiện thuận lợi cho việc tiếp cận và sử dụng TMĐT.
- **Sự phát triển của Internet:** Hạ tầng Internet được cải thiện mạnh mẽ, phủ sóng rộng khắp, tạo môi trường lý tưởng cho TMĐT phát triển.
- **Sự gia tăng smartphone:** Tỷ lệ người dùng smartphone ngày càng cao, mở ra kênh mua sắm trực tuyến tiện lợi và linh hoạt.
- **Thay đổi thói quen tiêu dùng:** Người tiêu dùng ngày nay ưa chuộng sự tiện lợi, nhanh chóng và đa dạng lựa chọn mà TMĐT mang lại.

- **Bức tranh TMĐT Việt Nam:**

- **Thị trường sôi động:** Việt Nam được đánh giá là một trong những thị trường TMĐT tăng trưởng nhanh nhất khu vực Đông Nam Á, với tốc độ tăng trưởng ấn tượng 18% trong năm 2020.
- **Quy mô tiềm năng:** Theo dự báo, TMĐT Việt Nam sẽ đạt quy mô 52 tỷ USD vào năm 2025, đứng thứ 3 ASEAN.
- **Mô hình đa dạng:** Các mô hình TMĐT phổ biến tại Việt Nam bao gồm B2C, C2C, B2B, m-commerce, social commerce,... đáp ứng nhu cầu đa dạng của người tiêu dùng và doanh nghiệp.
- **Sự tham gia của nhiều đối tượng:** Doanh nghiệp mọi quy mô, cá nhân và hộ gia đình đều có thể tham gia vào thị trường TMĐT.

- **Thách thức và giải pháp:**

- **Thiếu hụt nguồn nhân lực:** Nhu cầu nhân lực cho TMĐT ngày càng cao, đặc biệt trong lĩnh vực kỹ thuật và marketing.
- **Hạ tầng chưa hoàn thiện:** Hệ thống logistics và thanh toán trực tuyến còn nhiều hạn chế, ảnh hưởng đến trải nghiệm mua sắm của người tiêu dùng.
- **Niềm tin của người tiêu dùng:** Vấn đề hàng giả, hàng nhái, lừa đảo vẫn còn tồn tại, gây ảnh hưởng đến sự phát triển bền vững của TMĐT.

- **Giải pháp:**

- **Đào tạo nguồn nhân lực:** Nâng cao chất lượng đào tạo, tập trung vào các kỹ năng cần thiết cho TMĐT.
- **Hoàn thiện hạ tầng:** Phát triển hệ thống logistics và thanh toán trực tuyến an toàn, tiện lợi.
- **Tăng cường quản lý:** Nâng cao hiệu quả quản lý thị trường TMĐT, bảo vệ người tiêu dùng và thúc đẩy cạnh tranh lành mạnh.
- **Nâng cao nhận thức:** Nâng cao nhận thức của người tiêu dùng về TMĐT, khuyến khích sử dụng TMĐT một cách an toàn và hiệu quả.

- **Kết luận:**

TMĐT Việt Nam đang ở giai đoạn phát triển bùng nổ, tiềm năng to lớn và đóng vai trò quan trọng trong nền kinh tế số. Với sự chung tay của chính phủ, doanh nghiệp và người dân, TMĐT Việt Nam sẽ tiếp tục phát triển mạnh mẽ, góp phần nâng cao đời sống và thúc đẩy sự phát triển kinh tế - xã hội của đất nước.

1.2. So sánh ưu/nhược của các sàn TMĐT thông dụng ở Việt Nam

1.2.1. Shopee: "Ông vua" TMĐT Việt Nam



Hình 16. Sàn thương mại điện tử Shopee

Shopee hiện đang là sàn TMĐT lớn nhất Việt Nam với lượng người dùng khổng lồ và tốc độ tăng trưởng ấn tượng. Sức hút của Shopee đến từ nhiều yếu tố, bao gồm các ưu điểm:

- **Giao diện bắt mắt, dễ sử dụng:** Shopee có giao diện thân thiện, dễ dàng thao tác trên cả máy tính và điện thoại thông minh, phù hợp với mọi đối tượng người dùng.
- **Chương trình khuyến mãi đa dạng:** Shopee liên tục tung ra các chương trình khuyến mãi, giảm giá, flash sale,... thu hút người mua sắm.
- **Thanh toán tiện lợi:** Shopee hỗ trợ nhiều hình thức thanh toán như thanh toán trực tuyến, thu tiền mặt khi nhận hàng,... tạo sự thuận tiện cho người mua.
- **Dịch vụ giao hàng nhanh chóng:** Shopee hợp tác với nhiều nhà vận chuyển uy tín, đảm bảo giao hàng nhanh chóng và an toàn đến tay khách hàng.

- Nhược Điểm của Shopee:

- **Cạnh tranh cao:** Số lượng người bán gia tăng dẫn đến sự cạnh tranh khốc liệt.

Chất lượng sản phẩm chưa kiểm soát chặt chẽ: Một số sản phẩm không đảm bảo chất lượng.

- **Quy định đánh giá sản phẩm nghiêm ngặt:** Thời gian đánh giá lâu và nhiều sản phẩm cần đánh giá nhiều lần.
- **Chính sách không ưu tiên người bán:** Khi có xung đột với khách hàng, người bán thường gặp bất lợi.

- **Điều kiện trợ giá vận chuyển khẩn khe:** Yêu cầu giá trị đơn hàng tối thiểu cao khiến các cửa hàng nhỏ gặp khó khăn.

1.2.2. Lazada: Kẻ thách thức mạnh mẽ



Hình 17. Sàn thương mại điện tử Lazada

Lazada là đối thủ cạnh tranh trực tiếp của Shopee trên thị trường TMĐT Việt Nam. Sàn TMĐT này sở hữu những ưu điểm nổi bật như:

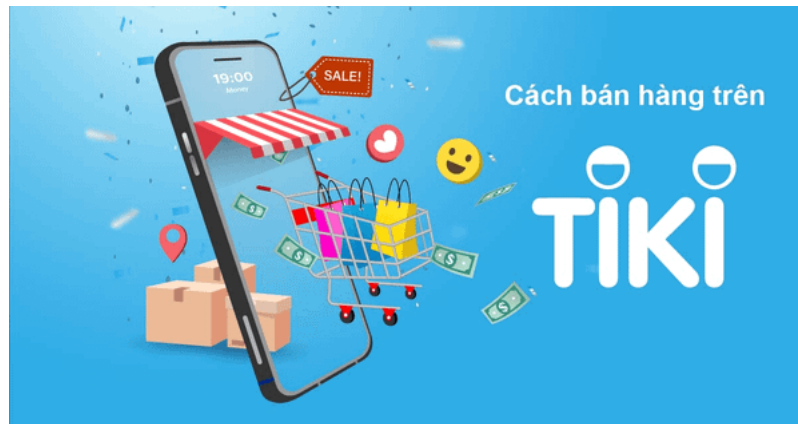
- **Lượng truy cập cao:** Lazada có lượng truy cập lớn thứ hai tại Việt Nam, chỉ sau Shopee.
- **Hỗ trợ đa dạng cho người bán:** Lazada cung cấp nhiều chương trình hỗ trợ cho người bán như hỗ trợ quảng cáo, kho bãi, vận chuyển,... giúp người bán dễ dàng tiếp cận khách hàng.
- **Nhiều chương trình khuyến mãi hấp dẫn:** Lazada thường xuyên tung ra các chương trình khuyến mãi, giảm giá, voucher,... thu hút người mua sắm.
- **Chính sách bảo vệ người tiêu dùng tốt:** Lazada có chính sách bảo vệ người tiêu dùng tốt, giúp người mua an tâm khi mua sắm.

- Nhược Điểm của Lazada:

- **Giao diện không nổi bật:** Giao diện không hấp dẫn như một số đối thủ.
- **Chi phí logistics cao:** Chi phí vận chuyển và nhận hàng khá cao.

- **Quy trình đăng ký phức tạp:** Yêu cầu hoàn thành các khóa học trực tuyến và khai báo nhiều giấy tờ để mở gian hàng.
- **Tăng trưởng khó khăn:** Tập trung vào người mua và có nhiều quy định khắt khe với người bán.

1.2.3. Tiki: Sàn TMĐT cao cấp với phong cách riêng



Hình 18. Sàn thương mại điện tử Tiki

Hình 18. Sàn thương mại điện tử Tiki

Tiki là sàn TMĐT được đánh giá cao về chất lượng sản phẩm và dịch vụ. Sàn TMĐT này có những ưu điểm như:

- **Sản phẩm chính hãng:** Tiki cam kết 100% sản phẩm chính hãng, đảm bảo chất lượng.
- **Dịch vụ khách hàng chuyên nghiệp:** Tiki có đội ngũ nhân viên chăm sóc khách hàng tận tâm, chu đáo, luôn sẵn sàng giải đáp mọi thắc mắc của khách hàng.
- **Giao hàng miễn phí:** Tiki thường xuyên có chương trình giao hàng miễn phí, giúp tiết kiệm chi phí cho người mua.
- **Giao diện sang trọng, hiện đại:** Tiki có giao diện sang trọng, hiện đại, mang đến trải nghiệm mua sắm cao cấp cho người dùng.

- Nhược Điểm của Tiki:

- **Ưu đãi giữ hàng trong kho hạn chế:** Hạn chế sự lựa chọn của người bán do chính sách giữ hàng trong kho.

- **Chi phí dịch vụ cao:** Để sử dụng dịch vụ chuyển phát nhanh, khách hàng phải đăng ký gói dịch vụ có giá trị cao.

1.2.4. Kết luận:

Từ phân tích các ưu và nhược điểm của Shopee, Lazada và Tiki, ta có thể rút ra các điểm cốt lõi để phát triển một ứng dụng thương mại điện tử hiệu quả:

- Chọn Lọc Các Ưu Điểm:

- Giao diện hấp dẫn và dễ sử dụng: Học hỏi từ Shopee.
- Chiến dịch marketing mạnh mẽ: Áp dụng từ Shopee và Lazada.
- Dịch vụ khách hàng hoàn hảo: Theo mẫu Lazada.
- Kiểm soát chất lượng sản phẩm nghiêm ngặt: Như Tiki.
- Miễn phí đăng ký và duy trì gian hàng: Giống Shopee và Lazada.

- Hạn Chế Các Nhược Điểm:

- Kiểm soát chặt chẽ chất lượng sản phẩm: Khắc phục nhược điểm của Shopee.
- Giảm chi phí logistics: Tối ưu hóa như Lazada.
- Đơn giản hóa quy trình đăng ký: Học hỏi từ Lazada.
- Chính sách ưu đãi vận chuyển hợp lý: Điều chỉnh để phù hợp với cửa hàng nhỏ.
- Hỗ trợ người bán hiệu quả: Cải thiện chính sách hỗ trợ như của Shopee.

- Định Hướng Phát Triển:

- Tích hợp công nghệ hiện đại: Sử dụng AI và machine learning.
- Phát triển ứng dụng di động thân thiện: Mua sắm mọi lúc, mọi nơi.
- Hợp tác với đối tác chiến lược: Đảm bảo trải nghiệm liền mạch và an toàn.

Kết hợp những yếu tố này sẽ giúp xây dựng một nền tảng thương mại điện tử mạnh mẽ và bền vững, đáp ứng nhu cầu ngày càng cao của thị trường.

2. Đặc tả chức năng

- Yêu cầu Chức năng:
 - Đăng ký và đăng nhập: Cho phép người dùng đăng ký tài khoản và đăng nhập vào ứng dụng bằng email, số điện thoại hoặc tài khoản mạng xã hội.

- Tạo và quản lý hồ sơ: Cho phép người dùng thay đổi thông tin cá nhân, địa chỉ nhận hàng và trạng thái xác thực của tài khoản.
- Tìm kiếm thông tin sản phẩm: Người dùng có thể truy cập các sản phẩm qua công cụ tìm kiếm, các đề xuất được đưa ra dựa vào lịch sử hoạt động trên dịch vụ.
- Gửi tin nhắn: Trò chuyện với các kênh người bán để trao đổi thông tin cần thiết và giải đáp thắc mắc.
- Đặt hàng và thanh toán: Đặt hàng sản phẩm và thanh toán với dịch vụ VNPay.
- Cá nhân hóa trải nghiệm: Mỗi tài khoản sẽ lưu trữ các lịch sử hoạt động trên sàn và đưa ra các đề xuất về sản phẩm nhằm phục vụ cho trải nghiệm cũng như tăng khối lượng đặt hàng.
- Mở kênh người bán: Cho phép khả năng đăng bán các sản phẩm của mình, chỉnh sửa các chiến lược buôn bán nhằm mang lại doanh thu cho bản thân.
- Yêu cầu Phi chức năng:
 - Tính sẵn sàng và Ổn định: Ứng dụng phải hoạt động ổn định và sẵn sàng sử dụng mọi lúc, ngay cả khi có nhiều người dùng cùng truy cập.
 - Bảo mật dữ liệu: Bảo vệ dữ liệu cá nhân của người dùng và thông tin nhạy cảm khỏi việc truy cập trái phép hoặc lộ thông tin.
 - Hiệu suất và tốc độ: Đảm bảo ứng dụng chạy mượt mà và có thời gian đáp ứng nhanh.
 - Hướng dẫn và hỗ trợ Người dùng: Cung cấp tài liệu hướng dẫn sử dụng và cơ chế hỗ trợ người dùng thông qua email hoặc chat trực tiếp.
 - Phân quyền người dùng: Quản lý quyền truy cập và khả năng thực hiện các thao tác khác nhau dựa trên vai trò của người dùng (người dùng thường, quản trị viên, v.v.).

3. Chức năng của các actors:

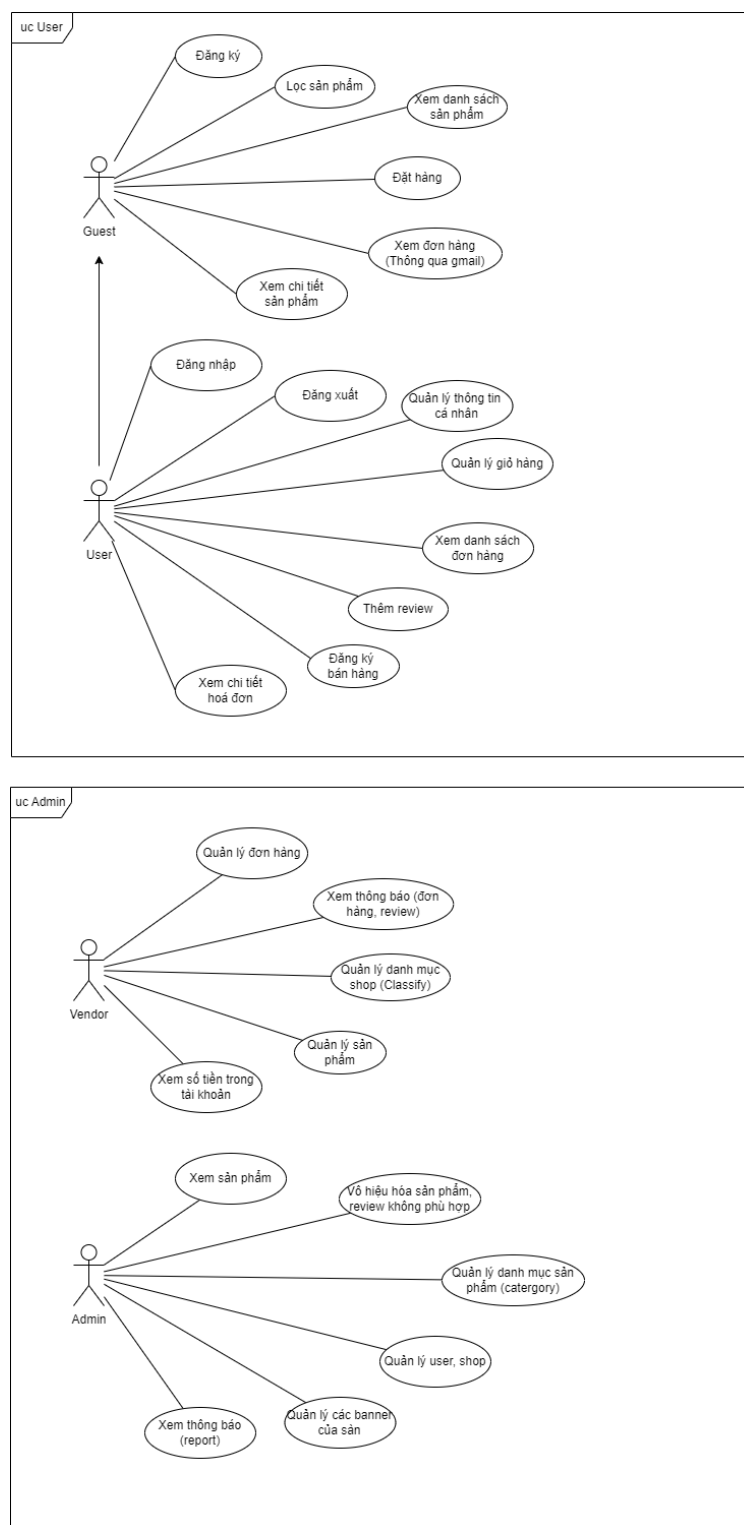
- **Guest:** Là khách hàng vắng lai có thể:
 - Xem danh sách sản phẩm trên trang chủ được hiển thị theo các chương trình khuyến mãi, theo cố định theo giới tính hoặc xem toàn bộ sản phẩm đi kèm là các

tính năng tìm kiếm dựa trên thông tin của sản phẩm, bộ lọc chi tiết với theo danh mục, thương hiệu, giảm giá, đánh giá, chương trình khuyến mãi của sản phẩm.

- Được hỗ trợ các thông tin liên quan đến shop thông qua hình thức hỗ trợ tư vấn trực tuyến.
- Xem chi tiết sản phẩm bao gồm các thông tin về sản phẩm với các biến thể hiện có cùng số lượng và giá cả của từng biến thể sản phẩm, ngoài ra còn có những đánh giá từ những người dùng khác cho sản phẩm đặc biệt là đề xuất các sản phẩm phối cùng.
- Có thể mua ngay sản phẩm và thanh toán theo các hình thức hiện có.
- Có thể trở thành user của hệ thống.
- **User:** Là khách hàng đã có tài khoản trong hệ thống có thể:
 - Xem danh sách sản phẩm trên trang chủ được hiển thị theo các chương trình khuyến mãi, theo cố định theo giới tính hoặc xem toàn bộ sản phẩm đi kèm là các tính năng tìm kiếm dựa trên thông tin của sản phẩm, bộ lọc chi tiết với theo danh mục, thương hiệu, giảm giá, đánh giá, chương trình khuyến mãi, giá.
 - Được hỗ trợ các thông tin liên quan đến shop thông qua hình thức hỗ trợ tư vấn tự động.
 - Xem chi tiết sản phẩm bao gồm các thông tin về sản phẩm với các biến thể hiện có cùng số lượng và giá cả của từng biến thể sản phẩm, ngoài ra còn có những đánh giá từ những người dùng khác cho sản phẩm đặc biệt là đề xuất các sản phẩm phối cùng.
 - Thêm sản phẩm muốn mua vào giỏ hàng, cập nhật hoặc xóa khỏi giỏ hàng, lựa chọn các sản phẩm muốn mua để tiến hành đặt hàng và thanh toán theo các hình thức hiện có.
 - Lưu và áp dụng các phiếu giảm giá theo từng cá nhân khách hàng.
 - Tạo ra và quản lý các địa chỉ riêng tư của cá nhân khách hàng.
 - Đánh giá sản phẩm khi đơn hàng được giao hoàn tất.
 - Quản lý các đơn hàng đã đặt và các thông tin chi tiết của đơn hàng.
- **Admin:** Là người kiểm soát hoạt động của hệ thống:

- Quản lý các sản phẩm bao gồm các thông tin liên quan cùng với việc quản lý các biến thể của sản phẩm.
- Quản lý danh mục sản phẩm.
- Quản lý thương hiệu sản phẩm.
- Quản lý các thông báo khi khách hàng đặt hàng và đánh giá.
- Quản lý đơn hàng và trạng thái đơn hàng.
- Quản lý các tài khoản người dùng bao gồm khách hàng và nhân viên.
- Quản lý các chương trình khuyến mãi.
- Quản lý hạng khách hàng.
- Quản lý phiếu giảm giá.
- Phản hồi đánh giá.
- Thống kê doanh thu theo từng khoảng thời gian khác nhau.
- **Vendor:** Là người bán hàng và quản lý các sản phẩm của họ trên hệ thống:
 - Thêm mới và quản lý các sản phẩm của họ, bao gồm thông tin chi tiết và biến thể của sản phẩm.
 - Quản lý danh mục sản phẩm liên quan đến cửa hàng của họ.
 - Quản lý thương hiệu sản phẩm mà họ bán.
 - Nhận và quản lý thông báo khi khách hàng đặt hàng và đánh giá sản phẩm.
 - Quản lý đơn hàng và cập nhật trạng thái đơn hàng.
 - Tham gia và quản lý các chương trình khuyến mãi mà họ đăng ký.
 - Phản hồi đánh giá của khách hàng cho các sản phẩm của họ.
 - Thống kê doanh thu và hiệu quả bán hàng của họ theo từng khoảng thời gian khác nhau.
 - Quản lý và áp dụng các phiếu giảm giá liên quan đến cửa hàng của họ.

4. Sơ đồ Use-case



Hình 19. Use-case tổng quát

5. Đặc tả Use-case:

5.1. Đăng ký:

Mã Use Case	UC-Guest01		
Tên Use Case	Đăng ký		
Mô tả	Use Case này cho phép thực hiện đăng ký tài khoản		
Tác nhân	Guest		
Điều kiện tiên quyết	Không có		
Điều kiện kết thúc	Thành công: Guest đăng ký tài khoản thành công Thất bại: Hệ thống sẽ thông báo lỗi xảy ra.		
Luồng sự kiện chính		Khách	Hệ thống
	1	Ở trang chủ nhấn nút “ <u>Log In / Sign Up</u> ”	
	2		Hệ thống chuyển hướng tới trang Đăng ký
	3	Khách hàng nhập “Username”, “Email”, “Mật khẩu”, “Xác nhận mật khẩu” và nhấn nút Submit & Register	

	4		Hệ thống kiểm tra thông tin vừa nhập, tạo tài khoản cho Khách hàng
		Thông báo Successfully registered và chuyển hướng tới trang đăng nhập	
Luồng sự kiện ngoại lệ		Khách	Hệ thống
	3.1	Khách nhập không đúng định dạng Email hoặc Xác nhận mật khẩu không trùng khớp với mật khẩu, hệ thống thông báo yêu cầu nhập lại	
	4.1		Hệ thống kiểm tra bị trùng Email, hệ thống thông báo lỗi và yêu cầu nhập lại

Bảng 1. Đặc tả Use-case Đăng ký

5.2. Loại sản phẩm:

Mã Use Case	UC-Guest02
Tên Use Case	Loại sản phẩm

Mô tả	Use Case này cho phép lọc sản phẩm thông qua tiêu chí lựa chọn		
Tác nhân	Guest		
Điều kiện tiên quyết	Không có		
Điều kiện kết thúc	Thành công: Hệ thống sẽ lọc sản phẩm theo lựa chọn mà người dùng muốn. Thất bại: Hệ thống sẽ thông báo lỗi xảy ra		
Luồng sự kiện chính		Khách	Hệ thống
	1	Khách thực hiện truy cập tới website	
	2	Chọn tab Products trên giao diện hệ thống	
	3		Hệ thống điều hướng tới trang Products
	4	Khách chọn filter để lọc dựa theo tiêu chí (All, Trending, Popular, New added)	
	5		Hệ thống hiển thị sản phẩm được lọc theo tiêu chí được chọn

Luồng sự kiện ngoại lệ		Khách	Hệ thống
	3.1		Lỗi kết nối đến Server

Bảng 2. Đặc tả Use-case Lọc sản phẩm

5.3. Xem danh sách sản phẩm:

Mã Use Case	UC-Guest03		
Tên Use Case	Xem danh sách sản phẩm		
Mô tả	Use Case này cho phép xem tất cả các sản phẩm có trên hệ thống		
Tác nhân	Guest		
Điều kiện tiên quyết	Không có		
Điều kiện kết thúc	Thành công: Hệ thống sẽ hiển thị tất cả các sản phẩm trên giao diện Thất bại: Hệ thống sẽ thông báo lỗi xảy ra.		
Luồng sự kiện chính		Khách	Hệ thống
	1	Khách thực hiện truy cập đến trang web	
	2	Chọn tab Products trên giao diện hệ thống	

	3		Hiển thị toàn bộ sản phẩm của cả sàn
Luồng sự kiện ngoại lệ		Khách	Hệ thống
	2.1		Lỗi kết nối đến Server

Bảng 3. Đặc tả Use-case Xem danh sách sản phẩm

5.4. Đặt hàng:

Mã Use Case	UC-Guest04
Tên Use Case	Đặt hàng (Khách)
Mô tả	Use Case này cho phép đặt hàng và mua sản phẩm mà không cần đăng nhập
Tác nhân	Guest
Điều kiện tiên quyết	Không có
Điều kiện kết thúc	Thành công: Guest đặt và mua ngay sản phẩm thành công Thất bại: Hệ thống sẽ thông báo lỗi xảy ra

Luồng sự kiện chính	Khách	Hệ thống
	1 Khách thực hiện truy cập tới website	
	2 Khách thực hiện click vào bất kỳ sản phẩm trên giao diện	
	3	Hệ thống điều hướng tới trang chi tiết sản phẩm, thông tin của sản phẩm sẽ được hiển thị
	4 Khách thực hiện chọn biến thể sản phẩm và nhập số lượng muốn mua	
	5 Khách thực hiện click chọn button “Add to cart”	
	6 Khách thực hiện click vào icon giỏ hàng để xác nhận và thanh toán	
	7	Hệ thống sẽ điều hướng tới trang quản lý giỏ hàng
	8 Khách chọn các mặt hàng muốn thanh toán trong cart, kiểm tra số lượng, hình ảnh sản phẩm...	
	9 Khách thực hiện click button Proceed To CheckOut	
	10	Hệ thống sẽ điều hướng tới trang Billing Details

	11	Khách nhập các thông tin bao gồm họ và tên, email, sdt, địa chỉ, phương thức thanh toán click chọn button Place Order	
	12		Hệ thống thông báo đặt hàng thành công, gửi mail thông báo về đơn hàng và quay lại trang chủ
Luồng thay thế	(A1). Khách lựa chọn phương thức thanh toán Online (VNPay) và click chọn “Hoàn thành đơn hàng”, chuyển Khách tới Use Case UC-Guest07		

Luồng sự kiện ngoại lệ		Khách	Hệ thống
	8.1	Khách chưa chọn sản phẩm hoặc shop muốn thanh toán, biến thể sản phẩm	Hệ thống thông báo yêu cầu chọn biến thể của sản phẩm muốn mua
	11.1	Khách nhập không đầy đủ thông tin	Hệ thống thông báo yêu cầu nhập lại
	11.2	Khách click vào button Continue shopping	Hệ thống sẽ điều hướng về lại trang chủ

Bảng 4. Đặc tả Use-case Đặt hàng

5.5. Xem chi tiết sản phẩm:

Mã Use Case	UC-Guest05
Tên Use Case	Xem chi tiết sản phẩm
Mô tả	Use Case này cho phép xem tất cả các sản phẩm có trên hệ thống
Tác nhân	Guest

Điều kiện tiên quyết	Không có		
Điều kiện kết thúc	Thành công: Hệ thống sẽ hiển thị thông tin chi tiết của sản phẩm. Thất bại: Hệ thống sẽ thông báo lỗi xảy ra.		
Luồng sự kiện chính		Khách	Hệ thống
	1	Khách thực hiện truy cập đến trang web	
	2	Chọn tab Products trên giao diện hệ thống	
	3		Hệ thống điều hướng tới trang Products
	4	Khách click chọn vào sản phẩm	
	5		Hệ thống điều hướng tới trang chi tiết sản phẩm, thông tin của sản phẩm sẽ được hiển thị
Luồng sự kiện ngoại lệ		Khách	Hệ thống
	2.1		Lỗi kết nối đến Server

Bảng 5. Đặc tả Use-case Xem chi tiết sản phẩm

5.6. Tìm kiếm sản phẩm:

Mã Use Case	UC-Guest06
--------------------	-------------------

Tên Use Case	Xem chi tiết sản phẩm		
Mô tả	Use Case này cho phép tìm kiếm sản phẩm thông qua từ khóa		
Tác nhân	Guest		
Điều kiện tiên quyết	Không có		
Điều kiện kết thúc	<p>Thành công: Hệ thống sẽ xuất hiện tên hoặc tag các sản phẩm phụ thuộc vào từ khóa mà người dùng đã nhập.</p> <p>Thất bại: Hệ thống sẽ thông báo lỗi xảy ra hoặc không tìm thấy sản phẩm.</p>		
Luồng sự kiện chính		Khách	Hệ thống
	1	Khách thực hiện truy cập đến trang web	
	2	Nhập từ khóa tìm kiếm vào ô tìm kiếm	
	3		Hệ thống sẽ hiển thị các sản phẩm theo từ khóa đã nhập.
	5		Hệ thống điều hướng tới trang chi tiết sản phẩm, thông tin của sản phẩm sẽ được hiển thị

Luồng sự kiện ngoại lệ		Khách	Hệ thống
	3.1		Lỗi kết nối đến Server

Bảng 6. Đặc tả Use-case tìm kiếm sản phẩm

5.7. Thanh toán VNPA Y:

Mã Use Case	UC-Guest07		
Tên Use Case	Thanh Toán VNPay		
Mô tả	UC cho phép khách thanh toán đơn hàng bằng VNPay		
Tác nhân	Guest		
Điều kiện tiên quyết	Không có		
Điều kiện kết thúc	Thành công: Khách thanh toán cho đơn hàng thành công Thất bại: Hệ thống sẽ thông báo lỗi đến khách: Thanh toán thất bại		
Luồng sự kiện chính		Khách	Hệ thống
	1		Hệ thống chuyển Guest đến trang thanh toán của VNPay.
	2	Khách chọn phương thức thanh toán.	

	3	Tuỳ theo phương thức thanh toán được chọn mà nhập các thông tin hoặc quét QR.	
	4		Server VNPay xác thực thông tin thẻ thành công, gửi một OTP đến số điện thoại đăng kí của thẻ và hiển thị Form nhập xác thực OTP để thanh toán
	5	Khách nhận và nhập OTP vào Form. Sau đó, nhấn nút “Thanh toán”.	
	6		Hệ thống VNPay xử lý thanh toán thành công và gửi thông báo đến hệ thống website, hệ thống website chuyển khách về lại trang chủ.
Luồng thay thế	A1. Nếu không muốn tiếp tục thanh toán, khách có thể click button Quay lại và chọn Huỷ thanh toán		

Luồng sự kiện ngoại lệ		Khách	Hệ thống
	3.1	Thông tin nhập vào bị sai, qr hết hạn	Hệ thống thông báo thanh toán thất bại
	4.1		Hệ thống VNPay xác thực OTP thất bại

Bảng 7. Đặc tả Use-case thanh toán VNPay

5.8. Đăng nhập:

Mã Use Case	UC-User01		
Tên Use Case	Đăng nhập		
Mô tả	Use Case này cho phép thực hiện đăng nhập		
Tác nhân	User		
Điều kiện tiên quyết	Đã có tài khoản trong hệ thống		
Điều kiện kết thúc	Thành công: Hệ thống sẽ điều hướng người dùng quay lại trang chủ sau khi đã đăng nhập. Thất bại: Hệ thống sẽ thông báo lỗi xảy ra.		
Luồng sự kiện chính		Người dùng	Hệ thống

	1	Người dùng nhấn nút “ <u>Log In / Sign Up</u> ” ở trang chủ	
	2		Hệ thống chuyển hướng tới trang Đăng nhập
	3	Người dùng nhập “Email” và “Mật khẩu” và nhấn button Login	
	4		Hệ thống kiểm tra thông tin trùng khớp trong hệ thống và chuyển hướng tới trang chủ
Luồng sự kiện ngoại lệ		Người dùng	Hệ thống
	3.1	Người dùng điền không đầy đủ thông tin	Hệ thống yêu cầu điền các thông tin còn thiếu
	4.1		Hệ thống kiểm tra sai thông tin email hoặc mật khẩu, thông báo lỗi cho người dùng.
	4.2		Hệ thống kiểm tra tình trạng của tài khoản là active hay banned.

Bảng 8. Đặc tả Use-case Đăng nhập

5.9. Đăng xuất:

Mã Use Case	UC-User02
Tên Use Case	Đăng xuất

Mô tả	Use Case này cho phép người dùng thực hiện đăng xuất ra khỏi hệ thống		
Tác nhân	User		
Điều kiện tiên quyết	Đã đăng nhập		
Điều kiện kết thúc	Thành công: Hệ thống sẽ điều hướng tới trang chủ		
Luồng sự kiện chính		Người dùng	Hệ thống
	1	Người dùng click chuột vào icon Welcome User	
	2		Hệ thống chuyển hướng người dùng đến trang Quản lí tài khoản
	3	Người dùng click vào button Logout	
	4		Hệ thống chuyển người dùng về trang chủ khi user đăng xuất thành công
Luồng sự kiện ngoại lệ		Người dùng	Hệ thống
	2.1		Lỗi kết nối đến Server

Bảng 9. Đặc tả Use-case Đăng xuất

5.10. Quản lý địa chỉ:

Mã Use Case	UC-User03		
Tên Use Case	Quản lý địa chỉ		
Mô tả	Use Case này cho phép người dùng quản lý địa chỉ		
Tác nhân	User		
Điều kiện tiên quyết	Đã đăng nhập		
Điều kiện kết thúc	Thành công: Người dùng thêm, xoá, sửa địa chỉ mới thành công Thất bại: Hệ thống sẽ thông báo lỗi xảy ra		
Luồng sự kiện chính		Người dùng	Hệ thống
	1	Người dùng click chuột vào icon Welcome User	
	2		Hệ thống chuyển hướng người dùng đến trang Quản lý tài khoản
	3	Người dùng click vào button My address	
	4		Hệ thống chuyển người dùng đến trang quản lý địa chỉ
	5	Người dùng click vào button Add	

	6		Hệ thống hiển thị form Add new address
	7	Người dùng nhập vào các thông tin như tên, số điện thoại, tỉnh, huyện, thành phố,... và click button Save	
	8		Hệ thống trả về trang Quản lí địa chỉ sau khi thêm thông tin địa chỉ người dùng vừa nhập.
Luồng sự kiện thay thế			
		Người dùng	Hệ thống
	5A	Nếu người dùng đã nhập address trước đó, chọn button Edit.	Hệ thống hiển thị form Add new address nhưng với các thông tin trước đó của người dùng nhập.
	6A	Người dùng click button Save	Lưu các thông tin địa chỉ mà người dùng đã sửa.
	5B	Nếu người dùng đã nhập address trước đó, chọn button Delete.	Hệ thống hiển thị form để người dùng xác nhận xoá.
	6B	Người dùng click button Delete	Hệ thống xoá địa chỉ được chọn.

Luồng sự kiện ngoại lệ		Người dùng	Hệ thống
	7.1	Người dùng nhập thiếu thông tin	Thông báo lỗi để người dùng thêm vào

Bảng 10. Đặc tả Use-case Quản lý địa chỉ

5.11. Quản lý giỏ hàng:

Mã Use Case	UC-User04		
Tên Use Case	Quản lý giỏ hàng		
Mô tả	Use Case này cho phép thực hiện quản lý sản phẩm vào giỏ hàng		
Tác nhân	User		
Điều kiện tiên quyết	Đã đăng nhập		
Điều kiện kết thúc	Thành công: Người dùng xem chi tiết đơn hàng, in hoá đơn, huỷ đơn. Thất bại: Hệ thống sẽ thông báo lỗi xảy ra.		
Luồng sự kiện chính		Người dùng	Hệ thống
	1	Người dùng click chuột vào icon Welcome User	
	2		Hệ thống chuyển hướng người dùng đến trang Quản lí tài khoản

	3	Người dùng click vào button Orders	
	4		Hệ thống chuyển người dùng đến trang quản lý đơn hàng
	5	Nếu đã có đơn hàng, người dùng click vào button View	
	6		Hệ thống hiển thị form Order detail
	7	Người dùng click button Cancel order để hủy đơn, Print để in đơn	
	8		Hệ thống sửa lại status đơn hàng nếu người dùng hủy.
Luồng sự kiện ngoại lệ			
		Người dùng	Hệ thống
	3.1	Người dùng điền không đầy đủ thông tin	Hệ thống yêu cầu điền các thông tin còn thiếu

Bảng 11. Đặc tả Use-case Quản lý giỏ hàng

5.12. Quản lý thông tin cá nhân:

Mã Use Case	UC-User05
Tên Use Case	Quản lý thông tin cá nhân

Mô tả	Use Case này cho phép người dùng thực hiện quản lý thông tin cá nhân của mình		
Tác nhân	User		
Điều kiện tiên quyết	Đã đăng nhập		
Điều kiện kết thúc	Thành công: Người dùng thay đổi display name và verify mail. Thất bại: Hệ thống sẽ thông báo lỗi xảy ra.		
Luồng sự kiện chính		Người dùng	Hệ thống
	1	Người dùng click chuột vào icon Welcome User	
	2		Hệ thống chuyển hướng người dùng đến trang Quản lí tài khoản
	3	Người dùng click vào button Account details	
	4		Hệ thống chuyển người dùng đến trang quản lý thông tin cá nhân
	5	Người dùng nhập vào Display Name	
	6	Người dùng chọn button Verify mail	
	7		Hệ thống gửi mail về để người dùng xác thực

	8	Người dùng click button Save	
	9		Lưu thông tin Display name sau khi người dùng thay đổi
Luồng sự kiện ngoại lệ		Người dùng	Hệ thống
	5.1	Người dùng điền không đầy đủ thông tin	Hệ thống yêu cầu điền các thông tin còn thiếu

Bảng 12. Đặc tả Use-case Quản lý thông tin cá nhân

5.13. Thay đổi mật khẩu:

Mã Use Case	UC-User06
Tên Use Case	Thay đổi mật khẩu
Mô tả	Use Case này cho phép người dùng thực hiện thay đổi mật khẩu
Tác nhân	User
Điều kiện tiên quyết	Đã đăng nhập
Điều kiện kết thúc	Thành công: Người dùng thay đổi display name và verify mail. Thất bại: Hệ thống sẽ thông báo lỗi xảy ra.

Luồng sự kiện chính		Người dùng	Hệ thống
	1	Người dùng click chuột vào icon Welcome User	
	2		Hệ thống chuyển hướng người dùng đến trang Quản lí tài khoản
	3	Người dùng click vào button Change Password	
	4		Hệ thống chuyển người dùng đến trang thay đổi mật khẩu
	5	Người dùng nhập vào Current Password, New Password và Confirm Password	
	6		Hệ thống thông báo thay đổi mật khẩu thành công và chuyển về trang chủ
Luồng sự kiện ngoại lệ		Người dùng	Hệ thống
	5.1	Người dùng điền không đầy đủ thông tin	Hệ thống yêu cầu điền các thông tin còn thiếu
	5.2	Người dùng điền sai Confirm Password	Thông báo lỗi để người dùng nhập lại

Bảng 13. Đặc tả Use-case Thay đổi mật khẩu

5.14. Thêm review:

Mã Use Case	UC-User07		
Tên Use Case	Thêm review sản phẩm		
Mô tả	Use Case này cho phép người dùng thực hiện review sản phẩm		
Tác nhân	User		
Điều kiện tiên quyết	Người dùng xác nhận đơn hàng ở status DELIVERED		
Điều kiện kết thúc	Thành công: Người dùng thêm review thành công. Thất bại: Hệ thống sẽ thông báo lỗi xảy ra.		
Luồng sự kiện chính		Người dùng	Hệ thống
	1	Người dùng click vào button Review trong form Order detail	
	2		Hệ thống hiện lên form Add Review
	3	Người dùng thêm rating và comments cho sản phẩm, hình ảnh đi kèm.	
	4	Người dùng click button Publish sau khi đã điền đủ thông tin.	

	5		Hệ thống thông báo đã thêm Review thành công
	6	Người dùng click button Close để tắt form Add Review	
Luồng sự kiện ngoại lệ		Người dùng	Hệ thống
	3.1	Người dùng không điền rating hoặc bỏ trống text box comments.	Thông báo lỗi để người dùng điền.

Bảng 14. Đặc tả Use-case Thêm Review

5.15. Quản lý sản phẩm:

Mã Use Case	UC-Vendor01
Tên Use Case	Quản lý sản phẩm
Mô tả	Use Case này cho phép người bán hàng quản lý sản phẩm
Tác nhân	Vendor
Điều kiện tiên quyết	Đã đăng nhập trong Vendor Page
Điều kiện kết thúc	<p>Thành công: Người bán xem chi tiết sản phẩm, sửa chi tiết sản phẩm, xoá sản phẩm.</p> <p>Thất bại: Hệ thống sẽ thông báo lỗi xảy ra.</p>

Luồng sự kiện chính		Người bán	Hệ thống
	1	Người bán click chuột vào tab Products, sau đó chọn Product List	
	2		Hệ thống trả về trang quản lý sản phẩm
	3	Người bán chọn icon Xem chi tiết sản phẩm	
	4		Hệ thống chuyển đến trang Product View hiển thị các thông tin: Gallery, Detail, Description, Rating, Review
Luồng sự kiện thay thế		Người bán	Hệ thống
	3A	Người bán chọn icon Sửa sản phẩm	
	4A		Chuyển người bán đến trang Product Upload (UC-Vendor02)
	5A	Người bán nhập các thông tin muốn chỉnh sửa trên trang Product Upload	
	6A		Hệ thống kiểm tra thông tin người bán thay đổi hợp lệ, update và chuyển hướng về lại trang Product List
	3B	Người bán chọn icon Xóa sản phẩm	
	4B		Hệ thống hiển thị form Xác nhận xóa

	5B	Người bán click vào button Delete	
	6B		Hệ thống xóa sản phẩm người dùng chọn và cập nhật lại danh sách sản phẩm trong Product List
Luồng sự kiện ngoại lệ		Người bán	Hệ thống
	5A.1	Người bán nhập thiếu thông tin không được bỏ trống.	Hệ thống thông báo để người bán nhập đủ các thông tin cần thiết

Bảng 15. Đặc tả Use-case Quản lý sản phẩm

5.16. Quản lý đơn hàng:

Mã Use Case	UC-Vendor02
Tên Use Case	Quản lý đơn hàng
Mô tả	Use Case này cho phép người bán hàng quản lý đơn hàng
Tác nhân	Vendor
Điều kiện tiên quyết	Đã đăng nhập trong Vendor Page

Điều kiện kết thúc	Thành công: Người bán xem chi tiết đơn hàng, in hóa đơn và set status đơn hàng. Thất bại: Hệ thống sẽ thông báo lỗi xảy ra.		
Luồng sự kiện chính		Người bán	Hệ thống
	1	Người bán click chuột vào tab Orders	
	2		Hệ thống trả về trang quản lý đơn hàng
	3	Nếu đã có đơn hàng, người bán chọn icon Xem chi tiết đơn hàng	
	4		Hệ thống chuyển đến trang Invoice Details bao gồm toàn bộ thông tin về đơn hàng và button Set Status
	5	Người bán chọn button Set Status để chỉnh sửa status đơn hàng: pending, delivered, shipped,...	
	6	Người bán click save sau khi đã thay đổi status	
	7		Hệ thống chuyển hướng về trang Invoice Details và cập nhật satus của đơn hàng
Luồng sự kiện thay thế			
		Người bán	Hệ thống
	5A	Người bán chọn icon Print This Reciept	

	6A		Hệ thống hiển thị dialog để người bán lưu bản pdf của hoá đơn về máy tính.
	7A	Người bán chọn vị trí lưu và Save	
Luồng sự kiện ngoại lệ		Người bán	Hệ thống

Bảng 16. Đặc tả Use-case Quản lý đơn hàng

5.17. Xem thông báo:

Mã Use Case	UC-Vendor02
Tên Use Case	Xem thông báo
Mô tả	Use Case này cho phép người bán hàng xem thông báo
Tác nhân	Vendor
Điều kiện tiên quyết	Đã đăng nhập trong Vendor Page
Điều kiện kết thúc	Thành công: Người bán xem thông báo về review mới, đơn hàng mới. Thất bại: Hệ thống sẽ thông báo lỗi xảy ra.

Luồng sự kiện chính		Người bán	Hệ thống
	1	Người bán click chuột vào tab Notifications	
	2		Hệ thống trả về trang Thông báo
	3	Nếu có thông báo thì người bán có thể Click vào ... để set trạng thái của thông báo: Mark as Read, Mark as Unread, Delete.	
	4	Người bán click vào thông báo	
	5		Tuỳ theo thông báo người dùng click mà chuyển hướng người bán tới: ví dụ như review thì trả về trang review,...
Luồng sự kiện ngoại lệ		Người bán	Hệ thống

Bảng 17. Đặc tả Use-case Xem thông báo

5.18. Quản lý danh mục sản phẩm (Category):

Mã Use Case	UC-Admin01
Tên Use Case	Quản lý danh mục sản phẩm

Mô tả	Use Case này cho phép admin quản lý danh mục sản phẩm		
Tác nhân	Admin		
Điều kiện tiên quyết	Đã đăng nhập trong Host page với role Admin		
Điều kiện kết thúc	Thành công: Admin thêm xoá sửa và tìm kiếm category. Thất bại: Hệ thống sẽ thông báo lỗi xảy ra.		
Luồng sự kiện chính		Admin	Hệ thống
	1	Admin click chuột vào tab Products -> Category	
	2		Hệ thống trả về trang Category Manager
	3	Admin click vào button Add cate	
	4		Hệ thống hiển thị form Create Category
	5	Admin nhập category name và click button Create	
	6		Hệ thống điều hướng về lại trang Category Manager sau khi đã thêm mới category

Luồng sự kiện thay thế		Admin	Hệ thống
	3A	Admin click vào 1 category bất kỳ, sau đó chọn button Delete hoặc update.	
	4A		Hệ thống cập nhật lại trang category manager sau khi đã xoá hoặc cập nhật category được chọn.
	3B	Admin click vào button Search và nhập keyword vào	
	4B		Hệ thống hiển thị các category có từ khoá được nhập vào
Luồng sự kiện ngoại lệ		Admin	Hệ thống
	5.1	Không nhập vào tên của category	Không tạo category mới và trả về lại trang Category Manager

Bảng 18. Đặc tả Use-case quản lý danh mục sản phẩm

5.19. Quản lý User/Shop:

Mã Use Case	UC-Admin02
--------------------	-------------------

Tên Use Case	Quản lý User/Shop		
Mô tả	Use Case này cho phép admin quản lý tài khoản của User và Shop		
Tác nhân	Admin		
Điều kiện tiên quyết	Đã đăng nhập trong Host page với role Admin		
Điều kiện kết thúc	Thành công: Admin xem thông tin chi tiết của shop, user và ban account. Thất bại: Hệ thống sẽ thông báo lỗi xảy ra.		
Luồng sự kiện chính		Admin	Hệ thống
	1	Admin click chuột vào tab User -> User list	
	2		Hệ thống trả về trang Registered Users
	3	Nếu đã có tài khoản user, admin click vào button Xem thông tin cụ thể	
	4		Hệ thống trả về trang User Profile nếu đối tượng được chọn là user, Shop Profile nếu là shop, trong shop profile thì có cả thông tin của mặt hàng của shop...
Luồng sự kiện thay thế		Admin	Hệ thống
	3A	Admin click vào 1 button set status	
	4A		Hệ thống hiển thị dialog để admin xác nhận có chuyển

		status account thành banned hay không?						
Luồng sự kiện ngoại lệ	<table> <tr> <td></td><td>Admin</td><td>Hệ thống</td></tr> <tr> <td></td><td></td><td></td></tr> </table>		Admin	Hệ thống				
	Admin	Hệ thống						

Bảng 19. Đặc tả Use-case Quản lý User/Shop

5.20. Quản lý Report:

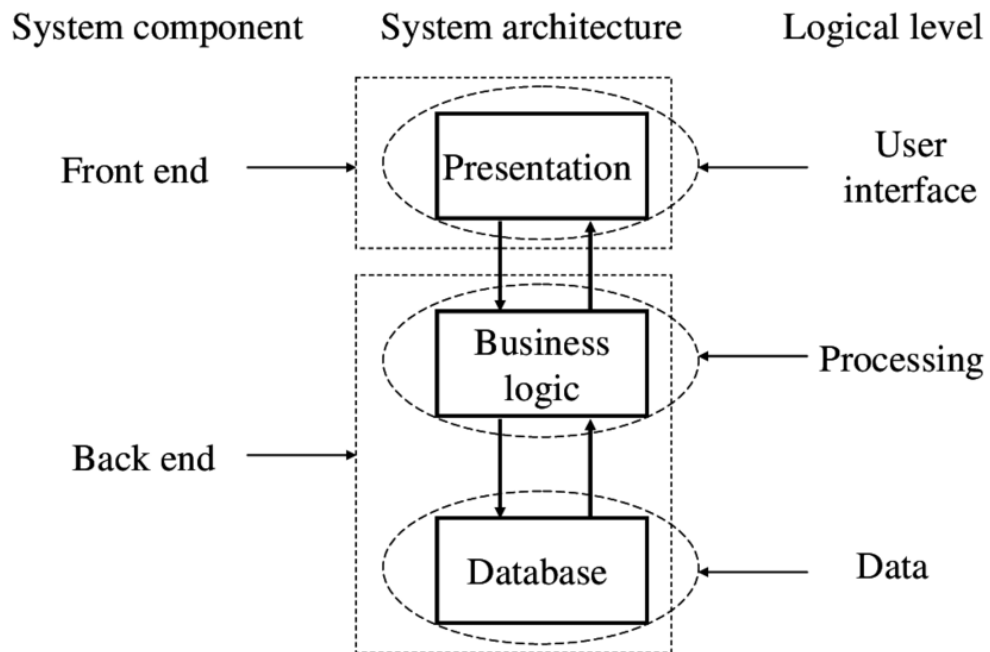
Mã Use Case	UC-Admin03
Tên Use Case	Quản lý Report
Mô tả	Use Case này cho phép admin quản lý tài khoản của User và Shop
Tác nhân	Admin
Điều kiện tiên quyết	Đã đăng nhập trong Host page với role Admin
Điều kiện kết thúc	<p>Thành công: Admin xem thông tin chi tiết của shop, user và ban account.</p> <p>Thất bại: Hệ thống sẽ thông báo lỗi xảy ra.</p>

Luồng sự kiện chính		Admin	Hệ thống
	1	Admin click chuột vào tab Reports	
	2		Hệ thống trả về trang Report List
	3	Admin click vào button Xem thông tin cụ thể của report	
	4		Hệ thống trả về form Report review
	5	Admin click vào button Disanble nếu muốn khoá report	
	6	Admin click vào button Close popup	
	7		Hệ thống trả về trang Report list sau khi đã disable report của người dùng.
Luồng sự kiện thay thế		Admin	Hệ thống
	3A	Admin click vào button Set status: chọn done hoặc pending -> Save	Hệ thống trả về trang Report list sau khi đã cập nhật status của report
Luồng sự kiện ngoại lệ		Admin	Hệ thống

Bảng 20. Đặc tả Use-case Quản lý shop

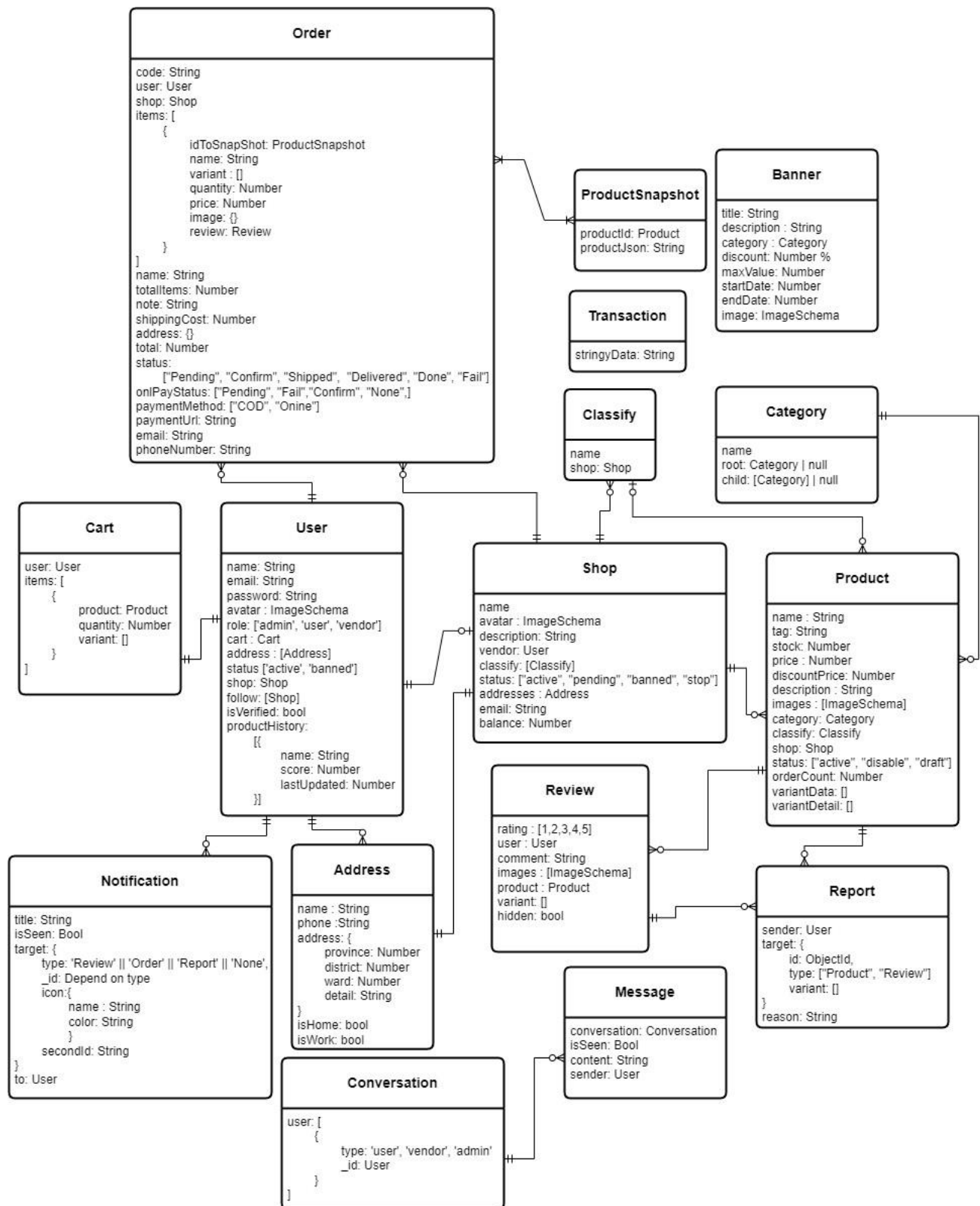
CHƯƠNG 4: THIẾT KẾ HỆ THỐNG

1. Kiến trúc hệ thống



Hình 20. Kiến trúc hệ thống

KHÓA LUÂN TỐT NGHIỆP



3. Đặc tả Cơ sở dữ liệu:

3.1. Bảng User

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null
2	name	String	not null
3	email	String	unique, not null
4	password	String	not null
5	role	String	not null, [admin, user, vendor]
6	cart	ObjectId	ref: Cart, not null
7	address	Array	ref: Address, not null
8	isVerified	bool	not null
9	shop	ObjectId	ref: Shop, not null
10	follow	[ObjectId]	ref: Shop
11	status	String	not null, [active, banned]
12	productHistory[].name	String	not null
13	productHistory[].score	Number	not null
14	productHistory[].name	Number	not null

Bảng 21. Bảng User

3.2. Bảng Address

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null
2	name	String	not null
3	phone	String	not null
4	address	String	not null
5	email	String	not null
6	isWork	Bool	not null

7	isHome	Bool	not null
---	--------	------	----------

Bảng 22. Bảng Address

3.3. Bảng Classify

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null
2	name	String	unique, not null
3	shop	ObjectId	Ref: Shop, not null

Bảng 23. Bảng Classify

3.4. Bảng Category

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null
2	name	String	not null
3	root	ObjectId	Ref: Category
4	child	ObjectId	Ref: Category

Bảng 24. Bảng Category

3.5. Bảng Cart

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null
2	user	ObjectId	ref: User, not null
3	items[].product	ObjectId	ref: Product, not null
4	items[].variant[]	[String]	
6	items[].quantity	Number	not null

Bảng 25. Bảng Cart

3.6. Bảng Notifications

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null

2	isSeen	Bool	not null
3	target.type	String	not null, ['Review' 'Order' 'Report' 'None']
4	target._id	ObjectId	Ref: depend on type, not null
5	target.icon.name	String	not null
6	target.icon.color	String	not null
7	target.secondId	ObjectId	

Bảng 26. Bảng Notification

3.7. Bảng Order

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null
2	user	ObjectId	ref: User, not null
3	code	String	not null
4	shop	ObjectId	Ref: Shop, not null
5	name	String	Not null
6	totalItem	Number	Not null
7	note	String	
8	shippingCost	Number	Not null
9	address	Object	
10	total	Number	Not null
11	status	String	["Pending", "Confirm", "Shipped", "Delivered", "Done", "Fail"], not null
12	onlPayStatus	String	["Pending", "Fail","Confirm", "None",], not null

13	paymentMethod	String	["COD", "Online"], not null
14	paymentUrl	String	
15	phone	String	Not null
16	email	String	Not null
17	items[].idToSnapShot	ObjectId	Ref: ProductSnapShot, not null
18	items[].name	String	Not null
19	items[].variant	[String]	
20	items[].quantity	Number	Not null
21	items[].price	Number	Not null
22	items[].image	Object	Not null
23	items[].review	ObjectId	Ref: Review, not null

Bảng 27. Bảng Order

3.8. Bảng Products

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null
2	name	String	not null
3	tags	String	not null
4	stock	Number	not null
5	price	Number	not null
6	discountPrice	Number	not null
7	shop	ObjectId	Ref : Shop, not null
8	status	String	not null, ["active", "disable", "draft"]
9	images	[Object]	not null
10	category	ObjectId	Ref : Category, not null
11	classify	ObjectId	Ref : Classify

12	description	Object	
13	variantData	Object	
14	variantDetail	String	

Bảng 28. Bảng Products

3.9. Bảng ProductSnapshots

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null
2	productId	ObjectId	ref: Product, not null
3	productJson	String	not null

Bảng 29. Bảng ProductSnapshot

3.10. Bảng Shop

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null
2	avatar	Object	not null
3	name	String	not null
4	description	String	not null
5	vendor	ObjectId	unique, not null, ref: User
6	classify	[ObjectId]	not null, ref: Classify
7	status	String	not null, ["active", "pending", "banned", "stop"]
8	addresses	Objectid	not null, ref: Address
9	email	String	not null
10	balance	Number	not null

Bảng 30. Bảng Shop

3.11. Bảng Report

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null
2	sender	ObjectId	not null, ref: User
3	target.id	ObjectId	not null,ref: depend on target.type
4	target.type	Number	not null
5	target.variant	[String]	
6	reason	String	not null

Bảng 31. Bảng Report

3.12. Bảng Reviews

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null
2	user	ObjectId	ref: User, not null
3	product	ObjectId	ref: Product, not null
4	variant	[String]	not null
8	rating	Number	not null
9	comment	String	not null
10	images	[Object]	
11	hidden	bool	not null

Bảng 32. Bảng Reviews

3.13. Bảng Conversation

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null
2	user[].type	String	not null, ['user', 'vendor', 'admin']
3	user[]._id	ObjectId	Ref: User, not null

Bảng 33. Bảng Conversation

3.14. Bảng Message

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null
2	conversation	ObjectId	not null, ref: Conversation
3	isSeen	Bool	not null
4	content	String	not null
5	sender	ObjectId	not null, ref: User

Bảng 34. Bảng Message

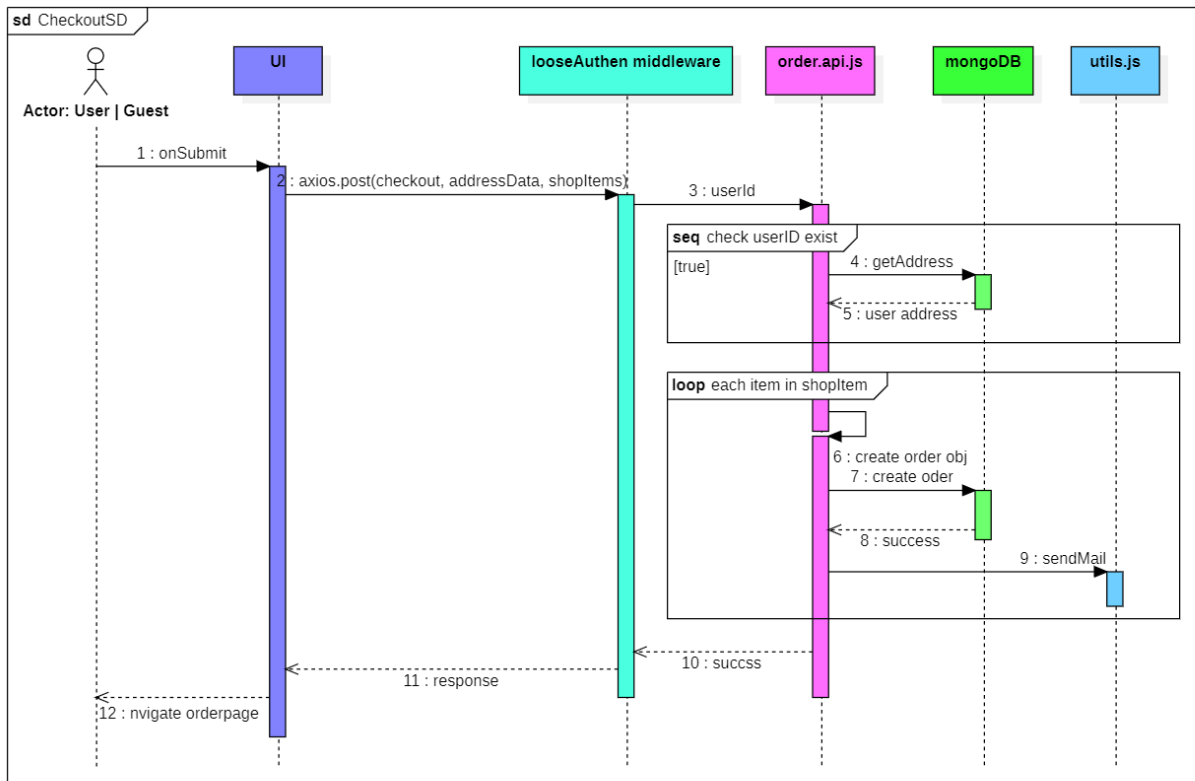
3.15. Bảng Banner

STT	Tên trường	Kiểu dữ liệu	Ràng buộc
1	_id	ObjectId	unique, not null
2	title	String	not null
3	description	String	not null
4	category	ObjectId	not null, ref: Category
5	discount	Number	not null
6	maxValue	Number	not null
7	startDate	Number	not null
8	endDate	Number	not null
9	image	Object	not null

Bảng 35. Bảng Banner

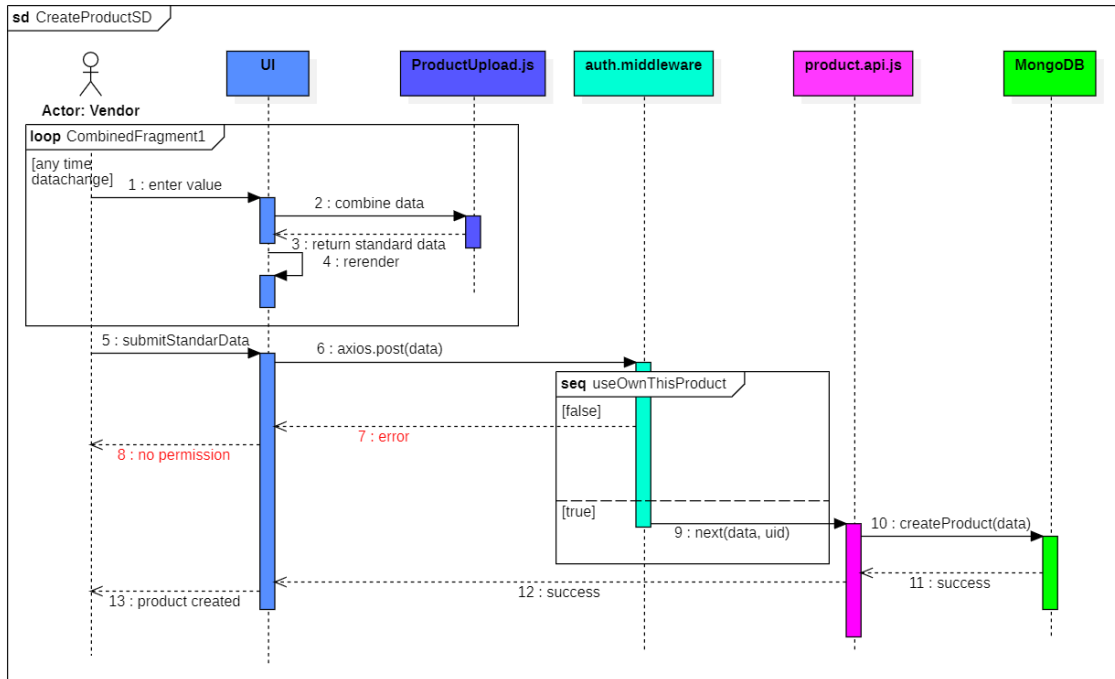
4. Sequence Diagram

4.1. Sequence Diagram Thanh toán



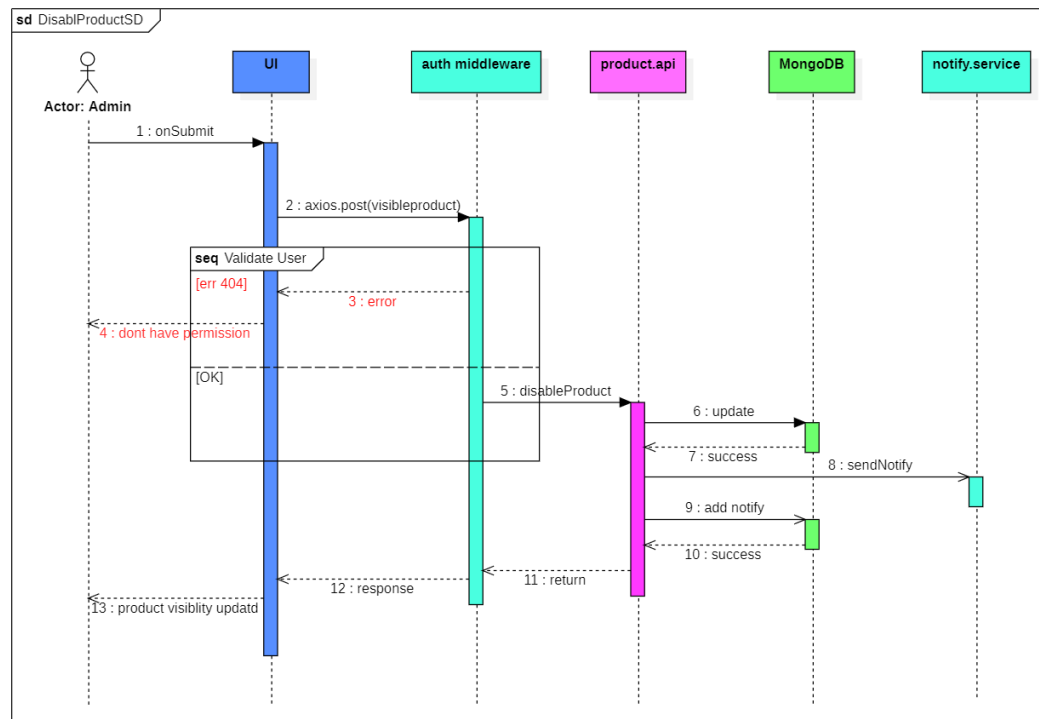
Hình 22. Sequence Diagram Thanh Toán

4.1. Sequence Diagram Tạo sản phẩm



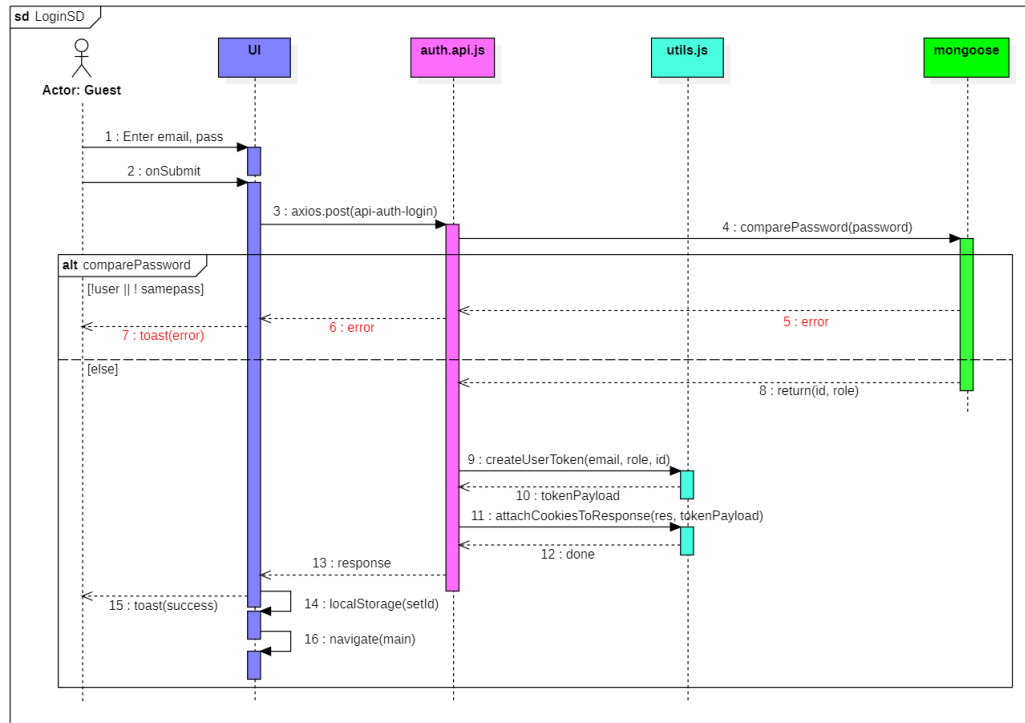
Hình 23. Sequence Diagram Tạo sản phẩm

4.2. Sequence Diagram Vô hiệu hoá sản phẩm



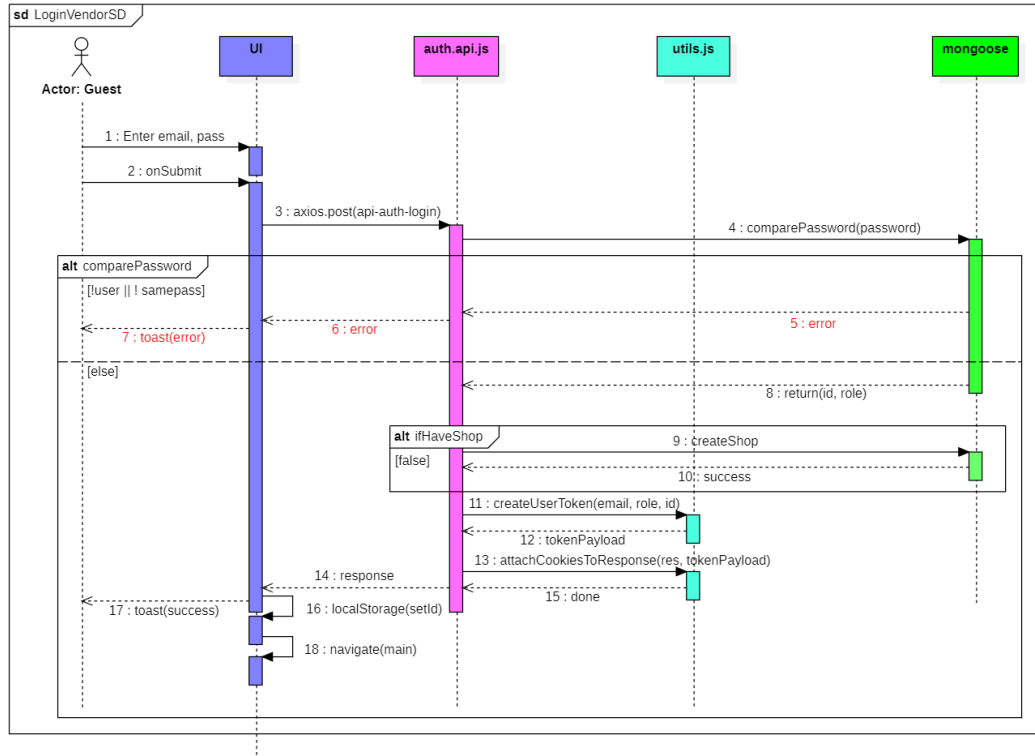
Hình 24. Sequence Diagram Vô hiệu hoá sản phẩm

4.3. Sequence Diagram Đăng nhập



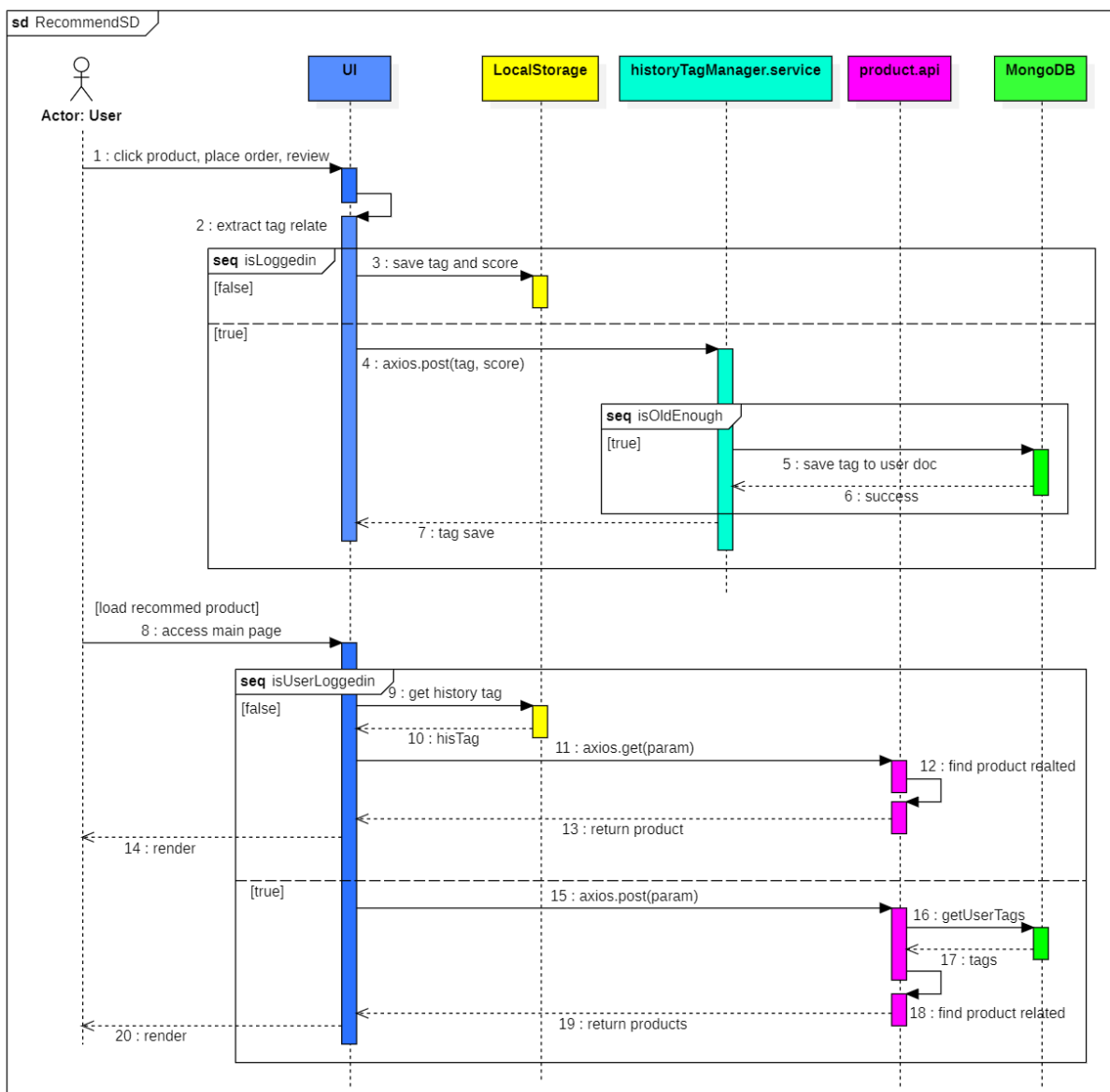
Hình 25. Sequence Diagram Đăng nhập

4.4. Sequence Diagram Đăng nhập role Vendor



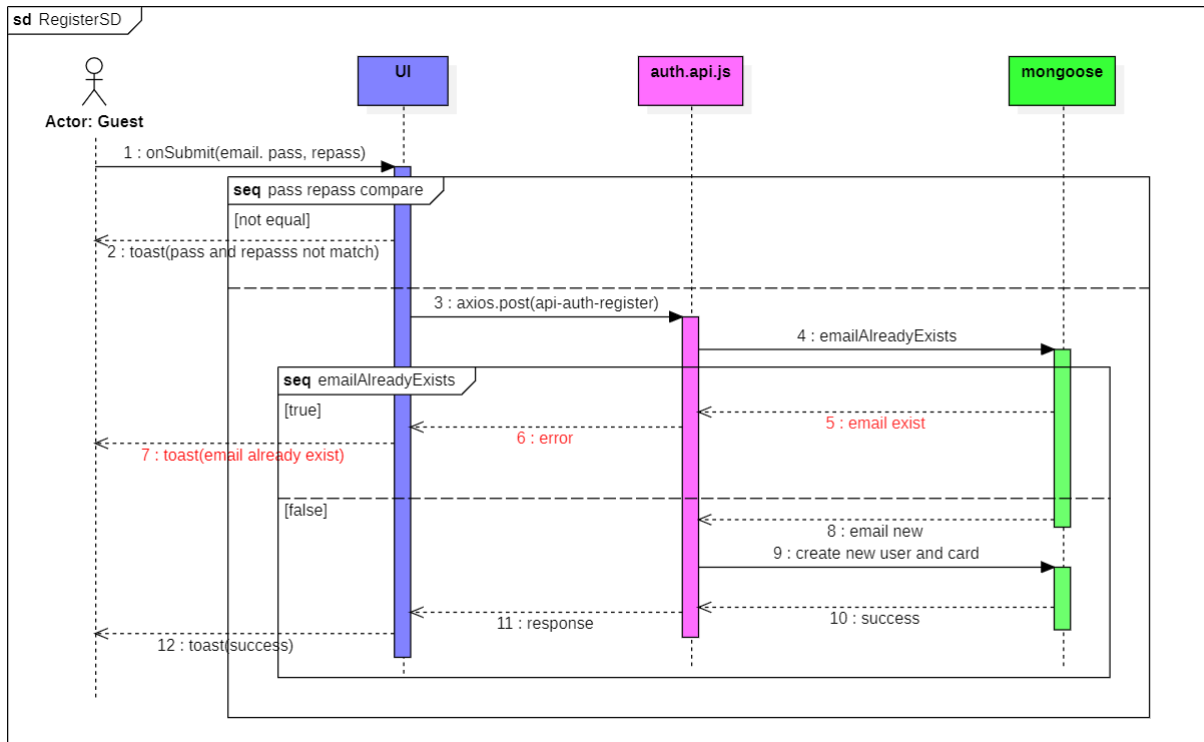
Hình 26. Sequence Diagram Đăng nhập role Vendor

4.5. Sequence Diagram Hệ thống gợi ý



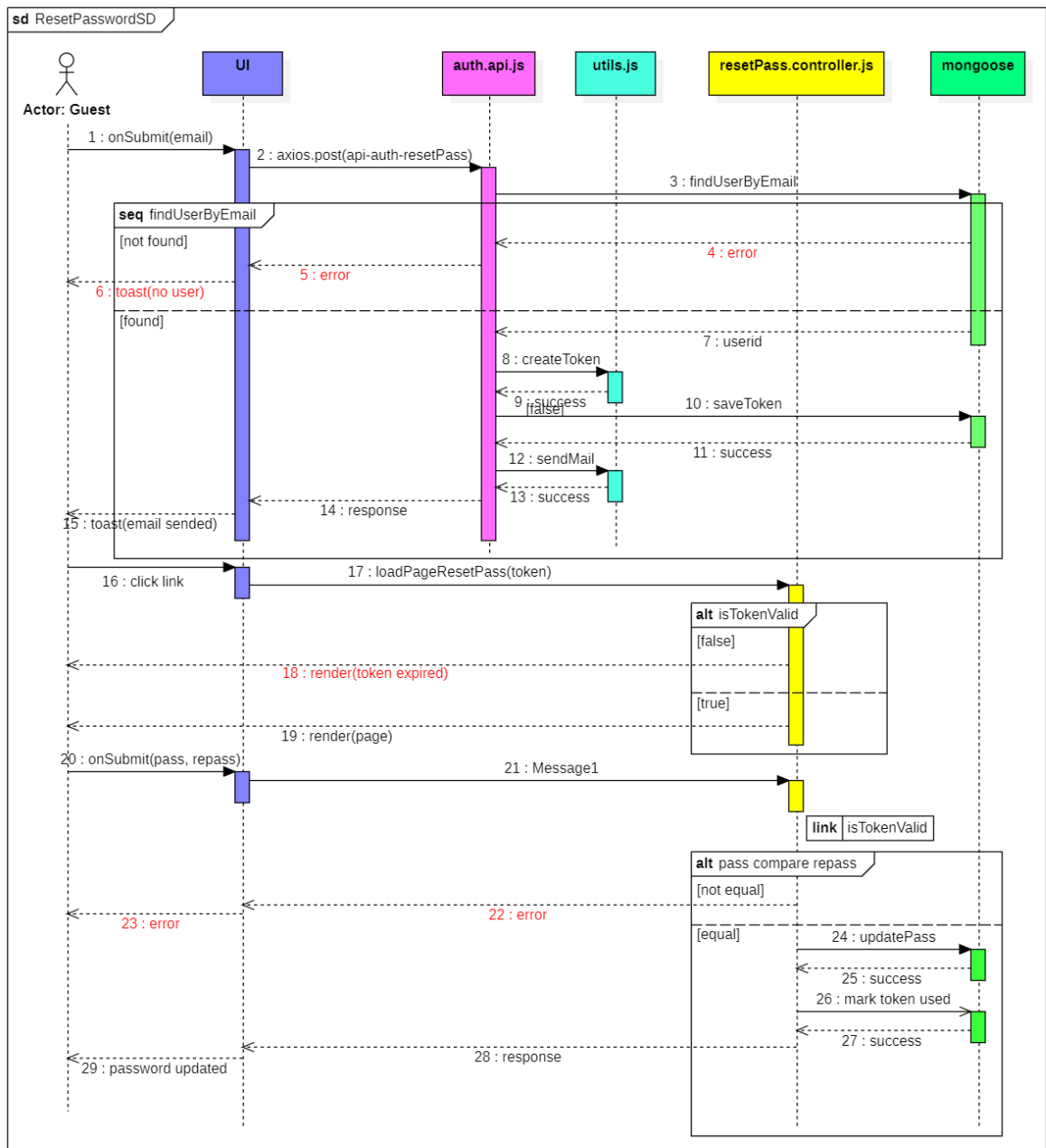
Hình 27. Sequence Diagram Hệ thống gợi ý

4.6. Sequence Diagram Đăng ký

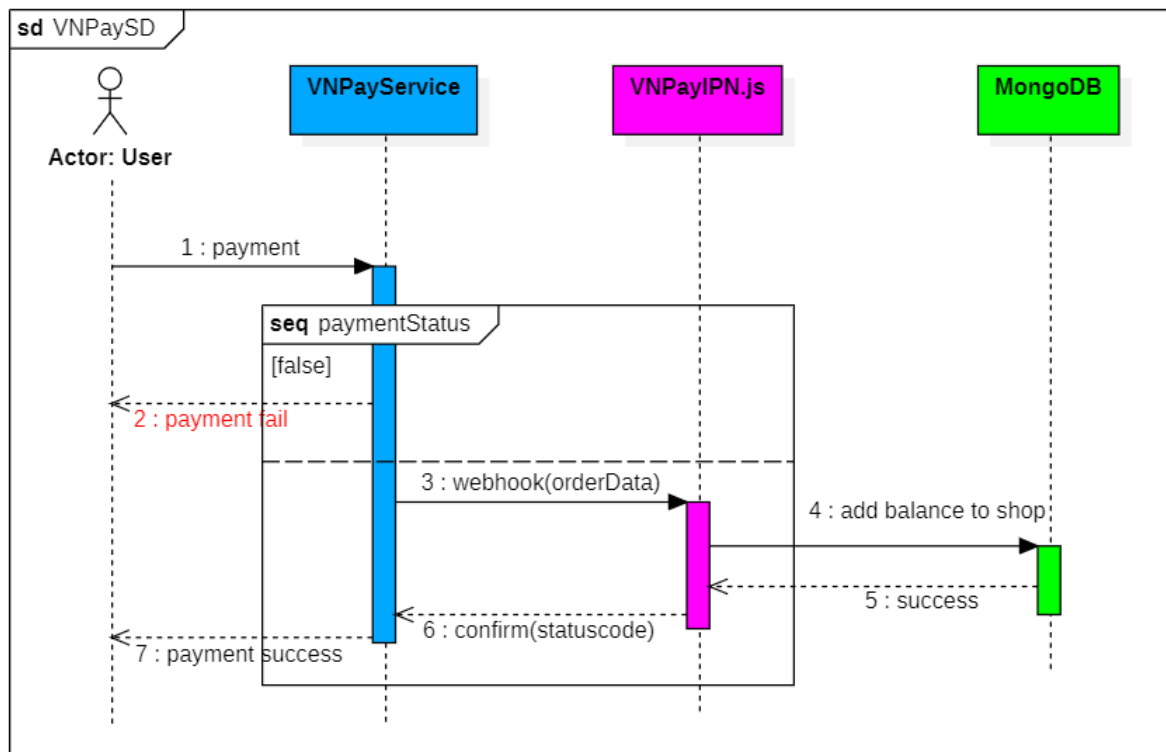


Hình 28. Sequence Diagram Đăng ký

4.7. Sequence Diagram Lấy lại mật khẩu



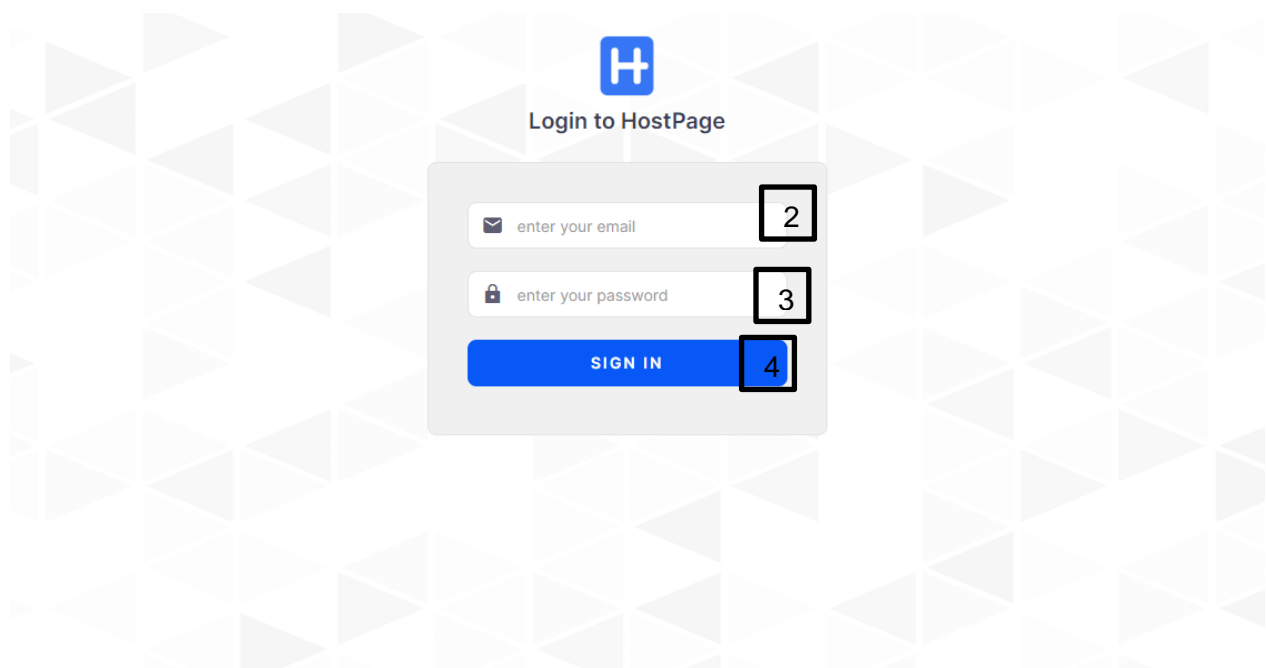
Hình 29. Sequence Diagram Lấy lại mật khẩu



Hình 30. Sequence Diagram Thanh toán VNPay

5. Thiết kế giao diện:

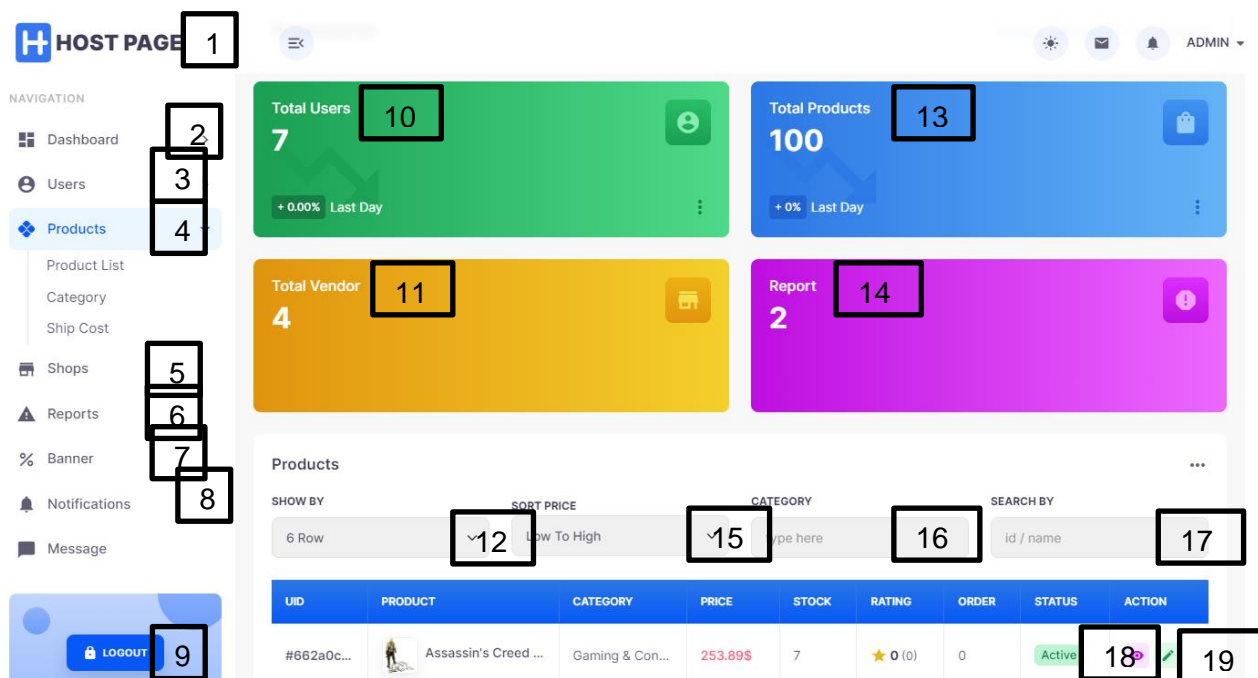
5.21. Giao diện trang Admin:



Hình 31. Giao diện trang đăng nhập (Admin)

STT	Kiểu	Mô tả
1	Image	Logo
2	Input	Nhập liệu Email
3	Input	Nhập liệu mật khẩu
4	Button	Tính năng đăng nhập

Bảng 36. Bảng Đặc tả giao diện trang đăng nhập (Admin)

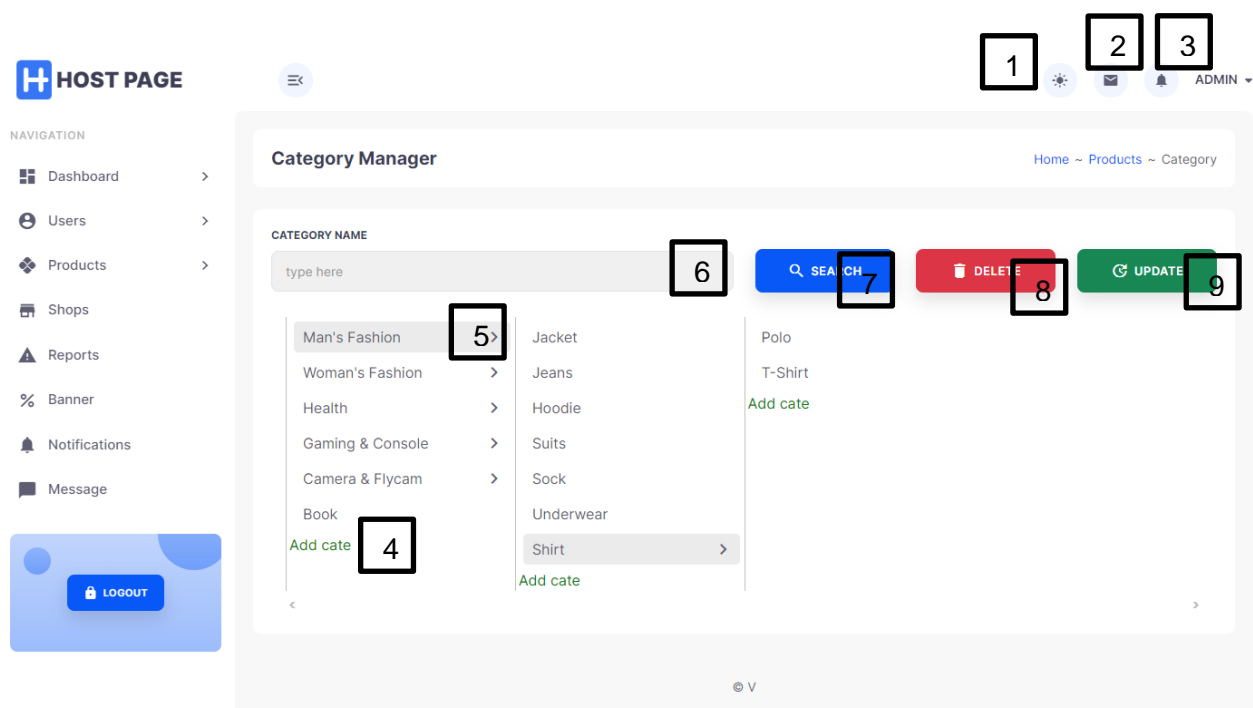


Hình 32. Giao diện trang chủ (Admin)

STT	Kiểu	Mô tả
1	Image	Logo
2	Button	Chuyển trang Dashboard
3	Button	Chuyển trang Users
4	Button	Chuyển trang Products
5	Button	Chuyển trang Shop
6	Button	Chuyển trang Reports
7	Button	Chuyển trang Banner
8	Button	Chuyển trang Notifications
9	Button	Đăng xuất
10	Label	Hiển thị Total Users
11	Label	Hiển thị Total Vendor
12	Combo Box	Số lượng item hiển thị

13	Label	Hiển thị Total Products
14	Label	Hiển thị Reports
15	Combo Box	Sort sản phẩm theo giá
16	Input	Tìm kiếm theo category
17	Input	Tìm kiếm theo id/name
18	Button	Chuyển sang trang Chi tiết sản phẩm
19	Button	Hiển thị form chỉnh sửa status

Bảng 37. Bảng Đặc tả giao diện trang chủ (Admin)



Hình 33. Giao diện quản lý category (Admin)

STT	Kiểu	Mô tả
1	Button	Đổi themes
2	Button	Mail
3	Button	Xem thông báo
4	Button	Add new category
5	Button	Hiển thị category con (nếu có)

6	Input	Nhập liệu category name để tìm kiếm
7	Button	Tìm kiếm dựa trên input (6)
8	Button	Xoá category được chọn
9	Button	Cập nhật tên category

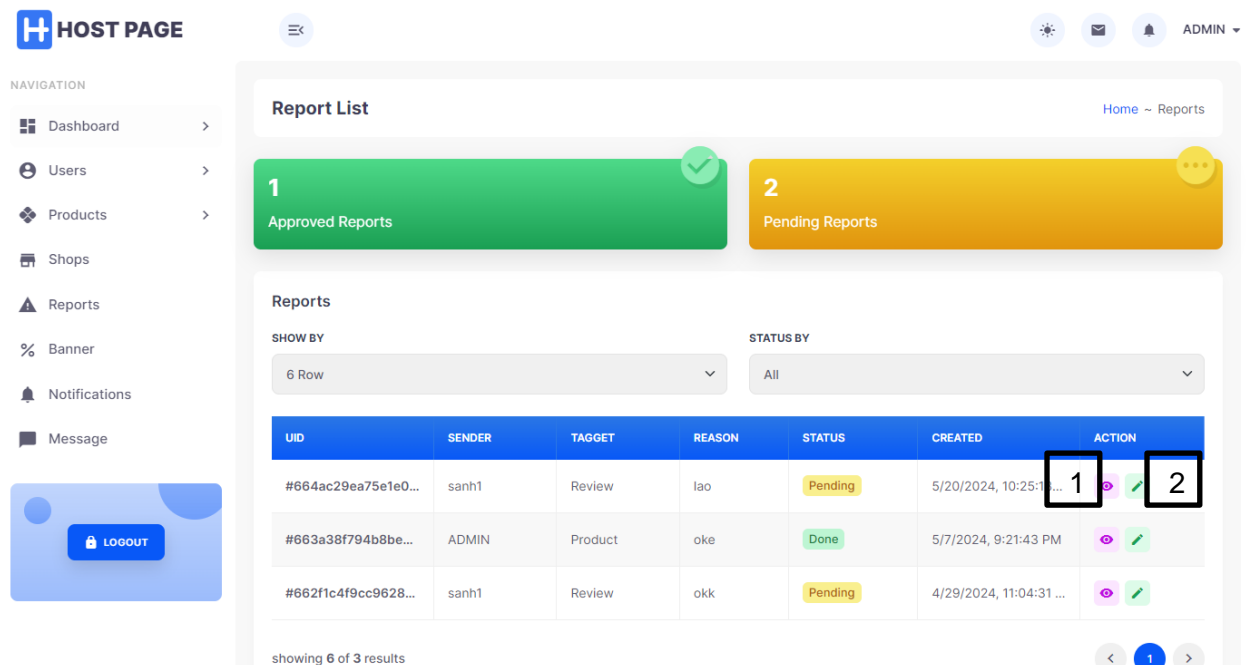
Bảng 38. Bảng Đặc tả giao diện quản lý category (Admin)

Hình 34. Giao diện quản lý đơn hàng (Vendor)

STT	Kiểu	Mô tả
1	Label	Số lượng order Pending
2	Label	Số lượng order Shipping
3	Label	Số lượng order Recieved
4	Label	Số lượng order Cancelled
5	Combo Box	Lọc đơn hàng theo status
6	Date Picker	Lọc theo ngày đặt

7	Input	Tìm kiếm đơn hàng theo code, tên hoặc email người mua
8	Button	Xem chi tiết đơn hàng
9	Button	In hoá đơn

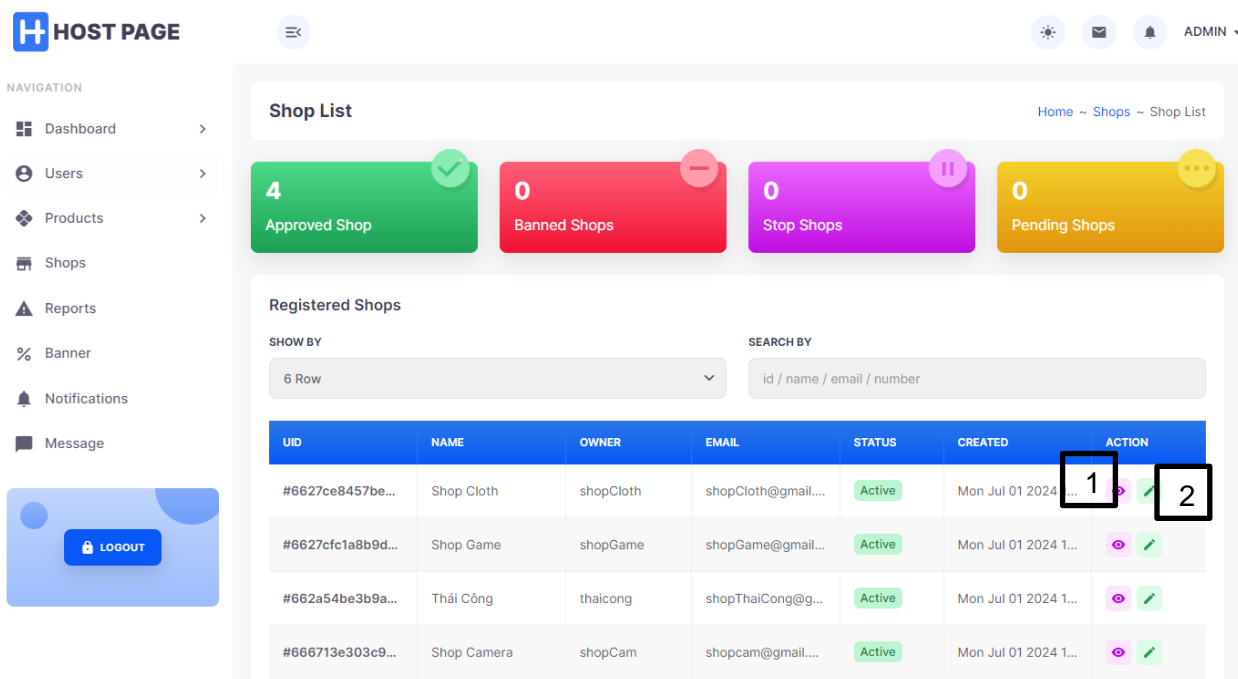
Bảng 39. Bảng Đặc tả giao diện quản lý đơn hàng (Vendor)



Hình 35. Giao diện quản lý báo cáo (Admin)

STT	Kiểu	Mô tả
1	Button	Xem chi tiết report
2	Button	Sửa status của report

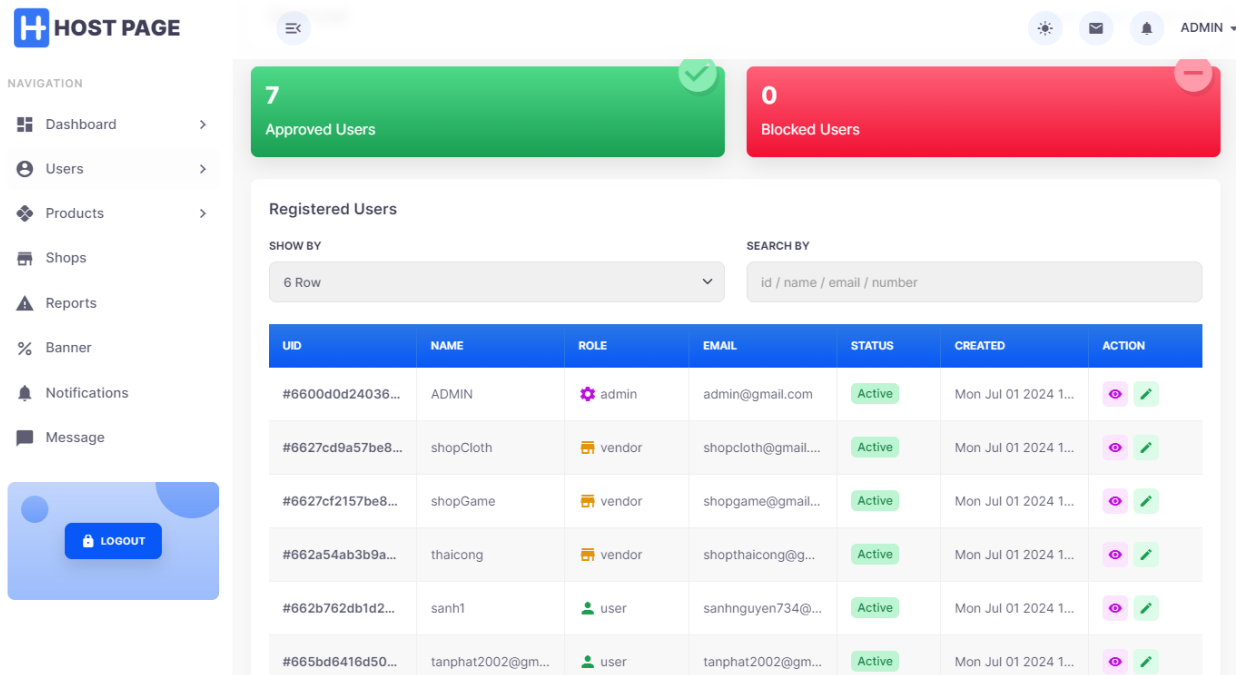
Bảng 40. Bảng Đặc tả giao diện quản lý báo cáo (Admin)



Hình 36. Giao diện quản lý shop (Admin)

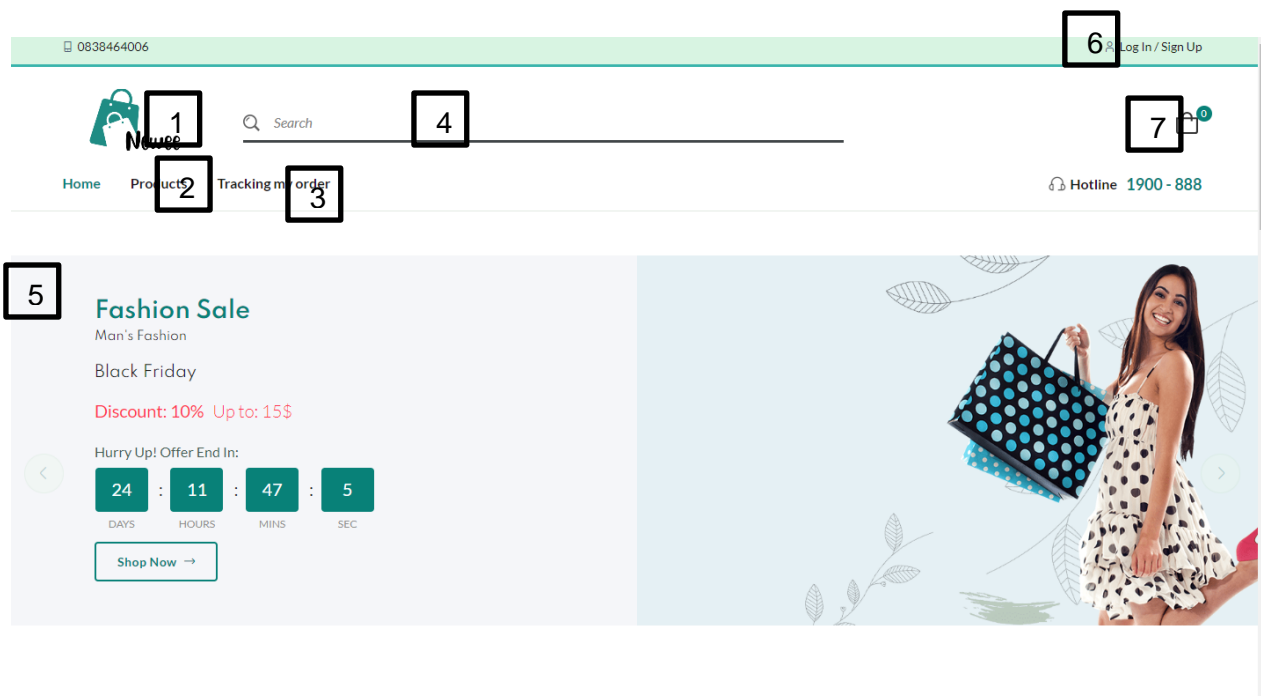
STT	Kiểu	Mô tả
1	Button	Xem chi tiết thông tin shop
2	Button	Chỉnh sửa status của shop

Bảng 41. Bảng Đặc tả giao diện quản lý shop (Admin)



Hình 37. Giao diện trang quản lý báo cáo (Admin)

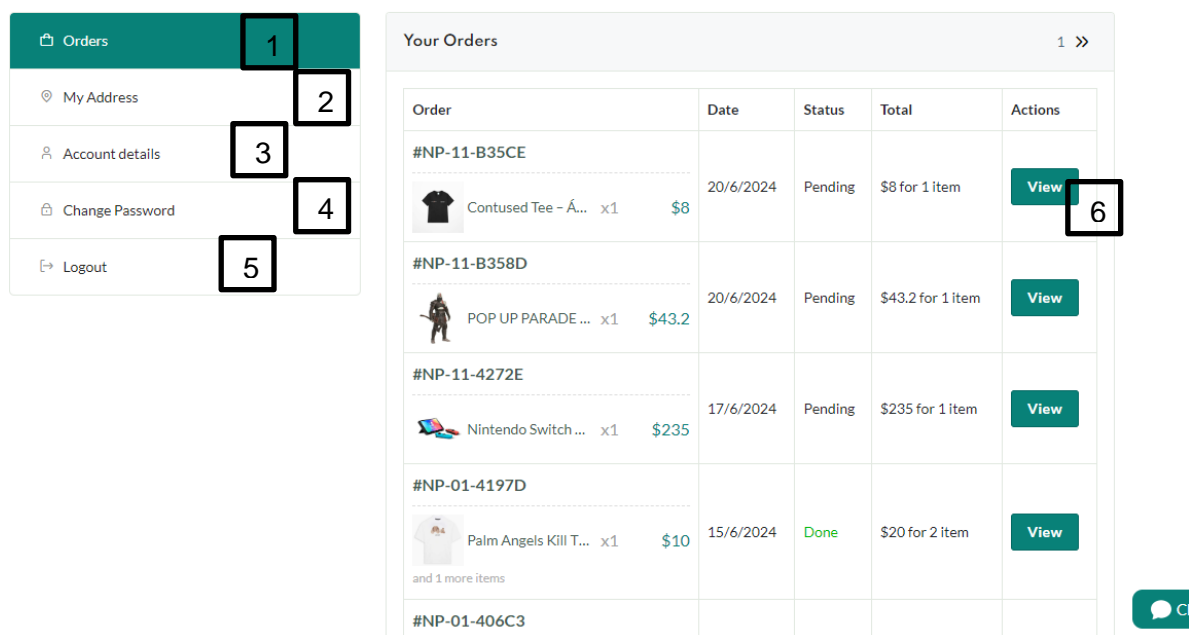
5.22. Giao diện trang User:



Hình 38. Giao diện trang chủ (User)

STT	Kiểu	Mô tả
1	Image	Logo
2	Button	Trang tất cả sản phẩm
3	Button	Tìm kiếm đơn hàng của mình
4	Input	Tìm kiếm sản phẩm bằng tên, tag...
5	Image	Banner
6	Button	Trang đăng nhập đăng ký
7	Button	Giỏ hàng

Bảng 42. Bảng Đặc tả giao diện trang chủ (User)



Hình 39. Giao diện trang quản lý thông tin cá nhân (User)

STT	Kiểu	Mô tả
1	Button	Trang quản lý đơn hàng của mình
2	Button	Trang quản lý địa chỉ
3	Button	Trang quản lý thông tin cá nhân
4	Button	Thay đổi mật khẩu

5	Button	Đăng xuất
6	Button	Xem chi tiết đơn mua

Bảng 43. Bảng Đặc tả giao diện trang quản lý thông tin cá nhân (User)

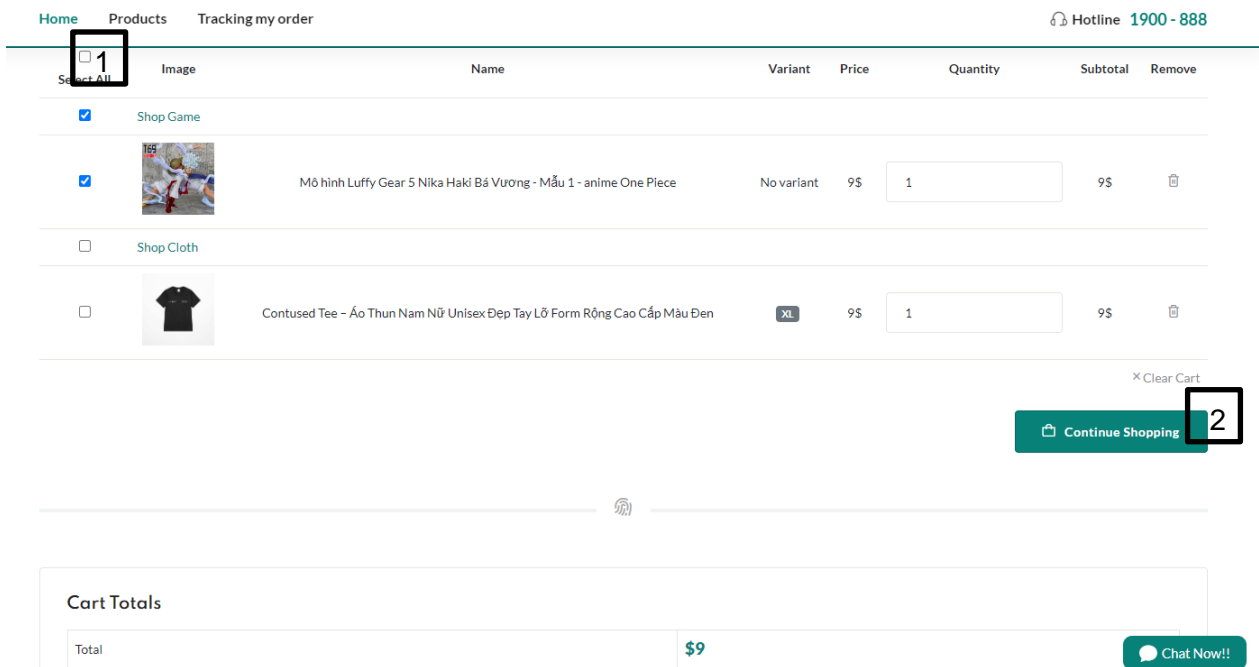
Hình 40. Giao diện trang đăng nhập/đăng ký (User)

STT	Kiểu	Mô tả
1	Input	Nhập email để đăng nhập
2	Input	Nhập password để đăng nhập
3	Button	Chuyển sang trang lấy lại mật khẩu
4	Button	Đăng nhập
5	Input	Nhập username để đăng ký
6	Input	Nhập email để đăng ký
7	Input	Nhập mật khẩu để đăng ký
8	Input	Nhập lại mật khẩu để đăng ký
9	Button	Đăng ký

Bảng 44. Bảng Đặc tả giao diện trang đăng nhập/đăng ký (User)

3	Input	Thêm ghi chú giao hàng
4	Check box	Chọn phương pháp thanh toán
5	Button	Đặt hàng
6	Button	Nhắn tin với chủ shop

Bảng 45. Bảng Đặc tả giao diện trang thanh toán (User)



Hình 42. Giao diện trang quản lý giỏ hàng (User)

STT	Kiểu	Mô tả
1	Checkbox	Chọn tất cả sản phẩm để thanh toán
2	Button	Tiếp tục mua sắm

Bảng 46. Bảng Đặc tả giao diện trang quản lý giỏ hàng (User)

CHƯƠNG 5: CÀI ĐẶT VÀ KIỂM THỬ

1. Các công cụ

Tool	Name
Environment	NodeJS
Code Editor	Visual Studio Code
Database	MongoDB
Source Control	Github

Bảng 47. Các công cụ sử dụng

2. Các công nghệ

Backend	NodeJs, ExpressJs
Frontend	ReactJS, Redux
Database	MongoDB Compass

Bảng 48. Các công nghệ sử dụng

3. Cài đặt

- Frontend User: Install npm: cmd -> npm install, run project: cmd -> npm run dev
- Frontend Admin: Install npm: cmd -> npm install, run project: cmd -> npm run start
- Backend: Install npm: cmd -> npm install, run project: cmd -> npm run devStart

4. Kiểm thử

STT	Mô tả test case	Kết quả mong đợi	Kết quả
1	Giao diện đăng nhập admin, vendor	As Expected	Pass
2	Giao diện đăng nhập dành cho user	As Expected	Pass
3	User có thể đổi mật khẩu khi quên	As Expected	Pass
4	Admin có thể tạo danh mục sản phẩm mới	As Expected	Pass
5	Admin có thể cập nhật danh mục sản phẩm	As Expected	Pass
6	Admin có thể vô hiệu hóa các sản phẩm	As Expected	Pass
7	Admin có thể quản lý tài khoản người dùng và các shop	As Expected	Pass

8	Admin có thể thêm banner	As Expected	Pass
9	Admin có thể xóa banner	As Expected	Pass
10	Admin có thể sửa banner	As Expected	Pass
11	Admin có thể quản lý các report từ người dùng	As Expected	Pass
12	Admin có thể thay đổi giá vận chuyển	As Expected	Pass
13	Admin có thể nhận các thông báo từ report của người dùng	As Expected	Pass
14	Admin có thể vô hiệu hóa tài khoản người dùng	As Expected	Pass
15	Admin có thể xem sản phẩm toàn sàn	As Expected	Pass
16	Vendor có thể xem các sản phẩm của shop	As Expected	Pass
17	Vendor có thể thêm sản phẩm cho shop	As Expected	Pass
18	Vendor có thể sửa các sản phẩm của shop	As Expected	Pass
19	Vendor có thể xóa các sản phẩm của shop	As Expected	Pass
20	Vendor có thể quản lý các phân loại sản phẩm	As Expected	Pass
21	Vendor có thể cập nhật thông tin của shop	As Expected	Pass
22	Vendor có thể xem và in các đơn hàng của shop	As Expected	Pass
23	Vendor có thể cập nhật trạng thái đơn hàng	As Expected	Pass
24	Vendor có thể xem các thống kê về lợi nhuận của shop	As Expected	Pass
25	Vendor có thể phản hồi trực tiếp các tin nhắn của khách	As Expected	Pass
26	Vendor có thể nhận thông báo từ hệ thống	As Expected	Pass
27	User, Guest có thể xem và tìm kiếm sản phẩm theo bộ lọc	As Expected	Pass
28	User, Guest có thể quản lý các sản phẩm trong giỏ hàng	As Expected	Pass
29	User, Guest có thể thay đổi số lượng trong giỏ hàng	As Expected	Pass
30	User, Guest có thể lựa chọn sản phẩm để đặt hàng	As Expected	Pass
31	User, Guest có thể thực hiện thanh toán bằng VNPay	As Expected	Pass
32	User, Guest có thể đặt hàng	As Expected	Pass
33	User, Guest có thể xem các đơn hàng đã đặt	As Expected	Pass
34	User, Guest có thể hủy đơn hoặc xác nhận đã nhận hàng	As Expected	Pass
35	User, Guest có thể xem chi tiết đơn hàng	As Expected	Pass
36	User, Guest có thể xem các sản phẩm đang được giảm giá	As Expected	Pass

37	User có quản lý các địa chỉ giao hàng cá nhân	As Expected	Pass
38	User có thể xác thực địa chỉ email	As Expected	Pass
39	User có thể cập nhật mật khẩu tài khoản	As Expected	Pass
40	User, Guest có thể xem các sản phẩm đề xuất theo nhu cầu	As Expected	Pass
41	User, Guest có thể truy cập vào shop cụ thể	As Expected	Pass

Bảng 49. Kết quả kiểm thử

CHƯƠNG 6: KẾT LUẬN

1. Kết quả đạt được

Qua việc xây dựng sàn thương mại điện tử đã thể hiện sự tiến bộ đáng kể về mặt kiến thức chuyên môn và khả năng quản lý công việc, đội nhóm. Cùng với đó là sự hiểu biết về các thành phần cơ bản và nâng cao của một sàn thương mại điện tử đặc biệt với sự phân chia chức năng rõ ràng các vai trò của admin, vendor, user và guest:

- **Kiến thức:** Xây dựng được kiến thức về quy trình phát triển 1 ứng dụng web. Sử dụng tốt các framework, thư viện lớn để tối ưu trong quá trình thiết kế và phát triển sản phẩm và rèn luyện tư duy về các thuật toán để giải quyết vấn đề một cách tối ưu và hiệu quả
- **Kỹ năng chuyên môn:** Xây dựng được tư duy về cách tối ưu hiệu suất của ReactJs, xây dựng các câu lệnh truy vấn chuẩn với MongoDB, thiết kế các luồng bảo mật hiệu quả khi sử dụng ExpressJs cho xử lý dữ liệu và tương tác thời gian thực khi sử dụng SocketIO
- **Kỹ năng mềm:** Nâng cao khả năng giao tiếp, làm việc nhóm và quản lý quy trình dự án. Cùng với đó là quản lý thời gian cá nhân, kỹ năng đọc và tìm kiếm các tài liệu liên quan
- **Sản phẩm:** Xây dựng được sản phẩm về sàn thương mại điện tử. Sản phẩm đã sở hữu đầy đủ các tính năng cơ bản như mua hàng, quản lý sản phẩm và tương tác giữa các người dùng với hiệu năng tốt. Và đồng thời cũng ứng dụng hệ thống gợi ý cơ bản để cá nhân hóa trải nghiệm người dùng

Dự án không chỉ phản ánh sự tiến bộ trong các kỹ năng chuyên và quản lý dự án mà còn chứng minh khả năng vận dụng các kiến thức này để tạo ra giải pháp công nghệ thực tế và đáp ứng nhu cầu của thị trường.

Các tính năng đạt được:

- Đăng ký, Đăng nhập, Đăng xuất

- **Tìm kiếm và Lọc Sản phẩm:** Hoàn thiện cơ bản tính năng lọc và tìm kiếm sản phẩm.
- **Xem Chi Tiết Sản phẩm:** Xem thông tin thêm và các đánh giá sản phẩm với hình ảnh và bình luận.
- **Quản lý Giỏ Hàng.**
- **Thanh Toán:** Tích hợp các phương thức thanh toán VNPay.
- **Theo Dõi Và Cập Nhật Trạng Thái Đơn Hàng**
- **Quản Lý Sản Phẩm:** Cho phép khả năng sản phẩm có nhiều biến thể
- **Quản Lý Phân Loại Sản Phẩm:** Khác với danh mục, phân loại sản phẩm cho phép các người bán sở hữu phân loại riêng cho cửa hàng của mình
- **Chi Tiết Đơn Hàng và In Hóa Đơn**
- **Quản Lý Danh Mục:** Danh mục mà các sản phẩm phải tuân theo khi đăng bán trên sàn
- **Quản Lý Tài Khoản:** Cho phép khả năng vô hiệu hóa các tài khoản
- **Quản Lý Địa Chỉ Giao Hàng**
- **Quản Lý Khuyến Mãi:** Mở rộng quản lý khuyến mãi theo danh mục với hình ảnh và giảm giá.
- **Thông Báo Thời Gian Thực:** Nhận các thông báo trong hệ thống như là đơn hàng mới, đánh giá mới,
- **Tương tác:** Tích hợp chat box cho phép trò chuyện giữa các người dùng.
- **Gợi Ý Sản Phẩm:** Dựa trên lịch sử hoạt động của người dùng trên hệ thống để đưa ra các gợi ý phù hợp.
- **Thông kê:** Quản lý các thông tin về số lượng sản phẩm và doanh thu

2. Ưu điểm và nhược điểm

2.1. Ưu điểm:

- Giao diện thân thiện với người dùng
- Người dùng có thể sử dụng chức năng dễ dàng

- Website chạy tương đối ổn định bao gồm các chức năng cơ bản
- Xây dựng bằng các công nghệ mới dễ dàng mở rộng và phát triển

2.2. Nhược điểm:

- Hiệu suất trang web chưa ở mức độ tốt nhất
- Khả năng bảo mật của trang web đang ở mức cơ bản
- Thiếu kiến thức chuyên sâu và thực tế về công nghệ sử dụng
- Hệ thống tìm kiếm và đề xuất được thiết kế ở mức độ cơ bản

3. Hướng phát triển

- Tối ưu website, tăng hiệu xuất cũng như SEO để tăng khả năng tiếp cận
- Tích hợp chức năng đăng nhập từ tài khoản bên thứ 3 như Facebook, Google, Zalo.
- Xây dựng thêm một số tính năng đánh giá, tin tức cho người dùng
- Tích hợp các mô hình trí tuệ nhân tạo trong việc cá nhân hóa trải nghiệm và tìm kiếm sản phẩm

TÀI LIỆU THAM KHẢO

- [1] ExpressJS, đăng bởi Lê Văn Liêm : <https://viblo.asia/p/phan-1-tim-hieu-express-js-framework-Qbq5Qq7m5D8>
- [2] NodeJS, đăng bởi TopDev : <https://topdev.vn/blog/node-js-la-gi/>
- [3] Nodemailer, đăng bởi nodemailer.com : <https://nodemailer.com>
- [4] MongoDB, đăng bởi ITViec: <https://itviec.com/blog/mongodb-la-gi/>
- [5] Mongoose, đăng bởi Hoàng Hùng: <https://viblo.asia/p/mongoose-cho-mongodb-nodejs-Qbq5QWvJZD8>
- [6] SocketIO, đăng bởi Lê Xuân Duy: <https://viblo.asia/p/buoc-dau-lam-quen-voi-nodejs-va-socketio-MJyGjQrWvPB>
- [7] JavaScript, đăng bởi tenten.vn: <https://tenten.vn/tin-tuc/javascript-la-gi/>
- [8] ReactJS, đăng bởi Từ Quốc Hưng : <https://200lab.io/blog/reactjs-la-gi/>
- [9] JWT, đăng bởi Nguyễn Tâm: <https://viblo.asia/p/json-web-token-la-gi-aWj533go56m>
- [10] Redux, đăng bởi Ngô Tuấn Doãn : <https://viblo.asia/p/redux-trong-react-la-gi-1Je5EDX0lnL>