|  |
| --- |
| BỘ GIÁO DỤC VÀ ĐÀO ĐẠO  TRƯỜNG ĐẠI HỌC CẦN THƠ  KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG    LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC  NGÀNH CÔNG NGHỆ THÔNG TIN  ĐỀ TÀI  Xây dựng website thương mại điện tử bán giày  trên nền tảng framework ExpressJS  và thư viện ReactJS  Building an e-commerce website for selling shoes  based on ExpressJS framework and ReactJS library  Sinh viên: Đào Minh Khoa  Mã số: B1809248  Khóa: K44  *Cần Thơ, 05/2022*  BỘ GIÁO DỤC VÀ ĐÀO TẠO  TRƯỜNG ĐẠI HỌC CẦN THƠ  KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG    LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC  NGÀNH CÔNG NGHỆ THÔNG TIN  ĐỀ TÀI  Xây dựng website thương mại điện tử bán giày  trên nền tảng framework ExpressJS  và thư viện ReactJS  Building an e-commerce website for selling shoes  based on ExpressJS framework and ReactJS library  Giáo viên hướng dẫn Sinh viên thực hiện  Ths. Nguyễn Minh Trung Đào Minh Khoa  B1809248  K44  *Cần Thơ, 05/2022* |

# LỜI CẢM ƠN

Trước hết với tình cảm sâu sắc và chân thành nhất, em được bày tỏ lòng biết ơn đến các quý thầy cô giảng viên trường Đại học Cần Thơ đã tận tình giảng dạy và thầy cô trong khoa Công nghệ Thông tin và Truyền thông đã mang lại cho em những kiến thức vô cùng quý giá và bổ ích trong quá trình học tập.

Trong quá trình thực hiện và hoàn thiện luận văn tốt nghiệp, em xin chân thành cảm ơn thầy ThS. Nguyễn Minh Trung, bộ môn Công Nghệ Thông Tin, khoa Công Nghệ Thông Tin và Truyền Thông. Thầy đã luôn tận tình trực tiếp hướng dẫn, giúp đỡ em trong suốt quá trình nghiên cứu, giúp em tiếp cận được đề tài một cách tốt nhất, xây dựng và hoàn thành Luận văn này.

Trong quá trình thực hiện đề tài luận văn tốt nghiệp rất khó khăn nhưng bản thân em đã cố gắng hoàn thành nhiệm vụ đã thể hiện được tinh thần trách nhiệm và tự giác, song song với đó là sự hướng dẫn nhiệt tình của thầy Nguyễn Minh Trung. Vì kiến thức bản thân còn nhiều hạn chế, trong quá trình thực hiện đề tài này em không tránh khỏi những sai sót, kính mong nhận được những ý kiến đóng góp từ quý thầy cô.

Em xin chân thành cảm ơn!

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

*Cần Thơ, ngày tháng năm 2022*

(chữ ký của giảng viên)

# NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

*Cần Thơ, ngày tháng năm 2022*

(chữ ký của giảng viên)

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc105055655)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN ii](#_Toc105055656)

[NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN iii](#_Toc105055657)

[MỤC LỤC iv](#_Toc105055658)

[MỤC LỤC HÌNH vii](#_Toc105055659)

[MỤC LỤC BẢNG ix](#_Toc105055660)

[DANH MỤC TỪ VIẾT TẮT xi](#_Toc105055661)

[TÓM TẮT xii](#_Toc105055662)

[ABSTRACT xiii](#_Toc105055663)

[CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN 1](#_Toc105055664)

[1.1. Đặt vấn đề 1](#_Toc105055665)

[1.2. Lịch sử giải quyết vấn đề 1](#_Toc105055666)

[1.3. Mục tiêu đề tài 1](#_Toc105055667)

[1.4. Đối tượng nghiên cứu 1](#_Toc105055668)

[1.5. Phạm vi nghiên cứu 2](#_Toc105055669)

[1.6. Phương pháp nghiên cứu 2](#_Toc105055670)

[1.7. Nội dung nghiên cứu 2](#_Toc105055671)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 3](#_Toc105055672)

[2.1. Giới thiệu về API và RESTful API 3](#_Toc105055673)

[2.2. ReactJS 4](#_Toc105055674)

[2.2.1. JavaScript XML 5](#_Toc105055675)

[2.2.2. Components 5](#_Toc105055676)

[2.2.3. Props và State 6](#_Toc105055677)

[2.2.4. LifeCycle 7](#_Toc105055678)

[2.2.5. Hook 7](#_Toc105055679)

[2.3. Thư viện Redux 8](#_Toc105055680)

[2.4. Thư viện axios 9](#_Toc105055681)

[2.5. Thư viện React UI – Material UI 9](#_Toc105055682)

[2.6. Cơ sở dữ liệu MySQL 10](#_Toc105055683)

[2.7. NodeJS 10](#_Toc105055684)

[2.7.1. Blocking I/O và Nonblocking I/O 11](#_Toc105055685)

[2.7.2. Synchronous và Asynchronous 11](#_Toc105055686)

[2.7.3. Callback 11](#_Toc105055687)

[2.7.4. Event Loop 12](#_Toc105055688)

[2.7.5. Module 13](#_Toc105055689)

[2.7.6. NPM - Node Package Manager 13](#_Toc105055690)

[2.8. ExpressJS Framework 14](#_Toc105055691)

[2.9. Mô hình MVC 16](#_Toc105055692)

[CHƯƠNG 3: NỘI DUNG 17](#_Toc105055693)

[3.1. Đặc tả 17](#_Toc105055694)

[3.2. Các chức năng chính của hệ thống 17](#_Toc105055695)

[3.3. Môi trường vận hành 18](#_Toc105055696)

[3.4. Use case diagram 18](#_Toc105055697)

[3.4.1. Use case diagram người dùng thành viên và người dùng khách 18](#_Toc105055698)

[3.4.2. Use case diagram quản lý 19](#_Toc105055699)

[3.4.3. Mô tả chi tiết các use case 20](#_Toc105055700)

[3.5. Mô hình dữ liệu mức quan niệm (CDM) 27](#_Toc105055701)

[3.6. Mô hình dữ liệu mức vật lý (PDM) 28](#_Toc105055702)

[3.7. Các bảng thực thể 29](#_Toc105055703)

[3.8. Sơ đồ chức năng 36](#_Toc105055704)

[3.8.1. Sơ đồ chức năng của người dùng khách 36](#_Toc105055705)

[3.8.2. Sơ đồ chức năng của khách hàng thành viên 36](#_Toc105055706)

[3.8.3. Sơ đồ chức năng của quản lý 37](#_Toc105055707)

[3.9. Quy trình đặt hàng 38](#_Toc105055708)

[3.9.1. Thanh toán giỏ hàng 38](#_Toc105055709)

[3.9.2. Thanh toán mua ngay 39](#_Toc105055710)

[3.9.3. Các yêu cầu ràng buộc chính 40](#_Toc105055711)

[3.9.4. Ràng buộc liên bộ, liên thuộc tính, liên quan hệ 40](#_Toc105055712)

[3.10. Giao diện khách hàng 42](#_Toc105055713)

[3.10.1. Giao diện chính 42](#_Toc105055714)

[3.10.2. Giao diện đăng ký, đăng nhập và quên mật khẩu 43](#_Toc105055715)

[3.10.3. Giao diện sản phẩm 45](#_Toc105055716)

[3.10.4. Xem thông tin chi tiết sản phẩm 46](#_Toc105055717)

[3.10.5. Giỏ hàng 47](#_Toc105055718)

[3.10.6. Thanh toán sản phẩm 48](#_Toc105055719)

[3.10.7. Lịch sử mua hàng 51](#_Toc105055720)

[3.10.8. Đánh giá sản phẩm 52](#_Toc105055721)

[3.11. Trò chuyện qua plugin messenger facebook 53](#_Toc105055722)

[3.12. Giao diện quản lý 53](#_Toc105055723)

[3.12.1. Sản phẩm 54](#_Toc105055724)

[3.12.2. Quản lý danh mục 55](#_Toc105055725)

[3.12.3. Quản lý nhập kho 56](#_Toc105055726)

[3.12.4. Quản lý hóa đơn 57](#_Toc105055727)

[3.12.5. Quản lý nhân viên 58](#_Toc105055728)

[CHƯƠNG 4: KIỂM THỬ VÀ ĐÁNH GIÁ 59](#_Toc105055729)

[4.1. Kiểm thử 59](#_Toc105055730)

[4.1.1. Mục tiêu 59](#_Toc105055731)

[4.1.2. Phạm vi kiểm thử 59](#_Toc105055732)

[4.2. Quản lý kiểm thử 59](#_Toc105055733)

[4.2.1. Môi trường kiểm thử 59](#_Toc105055734)

[4.2.2. Tài nguyên và cấp phát tài nguyên 60](#_Toc105055735)

[4.3. Các trường hợp kiểm thử 60](#_Toc105055736)

[4.3.1. Chức năng đăng ký, đăng nhập và lấy lại mật khẩu 60](#_Toc105055737)

[4.3.2. Chức năng đặt hàng 61](#_Toc105055738)

[4.3.3. Chức năng tìm kiếm, lọc sản phẩm 62](#_Toc105055739)

[4.3.4. Chức năng quản lý đơn hàng 62](#_Toc105055740)

[4.3.5. Chức năng lập hóa đơn 63](#_Toc105055741)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 64](#_Toc105055742)

[5.1. Kết luận 64](#_Toc105055743)

[5.1.1. Kết quả đạt được 64](#_Toc105055744)

[5.1.2. Hạn chế 64](#_Toc105055745)

[5.2. Hướng phát triển 64](#_Toc105055746)

[TÀI LIỆU THAM KHẢO 65](#_Toc105055747)

[PHỤ LỤC 66](#_Toc105055748)

[1. Hướng dẫn cài đặt 66](#_Toc105055749)

# MỤC LỤC HÌNH

[Hình 1: Mô hình hoạt động của REST API 4](#_Toc103894060)

[Hình 2: Logo thư viện ReactJS 4](#_Toc103894061)

[Hình 3: Function components hợp lệ 5](#_Toc103894062)

[Hình 4: Function components sử dụng arrow function 5](#_Toc103894063)

[Hình 5: Ví dụ class components 6](#_Toc103894064)

[Hình 6: Props 6](#_Toc103894065)

[Hình 7: LifeCycle trong ReactJS 7](#_Toc103894066)

[Hình 8: Logo Redux 8](#_Toc103894067)

[Hình 9: Logo cơ sở dữ liệu MySQL 10](#_Toc103894068)

[Hình 10: Logo NodeJS 10](#_Toc103894069)

[Hình 11: Logo ExpressJS 14](#_Toc103894070)

[Hình 12: Ví dụ cấu trúc route 15](#_Toc103894071)

[Hình 13: Mô hình MVC 16](#_Toc103894072)

[Hình 14: Sơ đồ use case người dùng thành viên và người dùng khách 18](#_Toc103894073)

[Hình 15: Sơ đồ use case người quản lý 19](#_Toc103894074)

[Hình 16: Mô hình dữ liệu mức quan niệm (CDM) 27](#_Toc103894075)

[Hình 17: Mô hình dữ liệu mức vật lý (PDM) 28](#_Toc103894076)

[Hình 18: Sơ đồ chức năng người dùng khách 36](#_Toc103894077)

[Hình 19: Sơ đồ chức năng của khách hàng thành viên 36](#_Toc103894078)

[Hình 20: Sơ đồ chức năng của quản lý 37](#_Toc103894079)

[Hình 21: Quy trình thanh toán giỏ hàng 38](#_Toc103894080)

[Hình 22: Quy trình thanh toán mua ngay 39](#_Toc103894081)

[Hình 23: Giao diện khách hàng khi chưa đăng nhập 42](#_Toc103894082)

[Hình 24: Giao diện khách hàng đã đăng nhập 42](#_Toc103894083)

[Hình 25: Giao diện đăng ký 43](#_Toc103894084)

[Hình 26: Giao diện nhập mã xác thực 43](#_Toc103894085)

[Hình 27: Giao diện đăng nhập 44](#_Toc103894086)

[Hình 28: Giao diện lấy lại mật khẩu 44](#_Toc103894087)

[Hình 29: Giao diện trang sản phẩm 45](#_Toc103894088)

[Hình 30: Chức năng lọc sản phẩm 45](#_Toc103894089)

[Hình 31: Giao diện thông tin chi tiết sản phẩm 46](#_Toc103894090)

[Hình 32: Chức năng phóng to sản phẩm 46](#_Toc103894091)

[Hình 33: Giao diện đánh giá 47](#_Toc103894092)

[Hình 34: Giao diện giỏ hàng 47](#_Toc103894093)

[Hình 35: Giao diện thanh toán sản phẩm 48](#_Toc103894094)

[Hình 36: Chọn thông tin nhận hàng 48](#_Toc103894095)

[Hình 38: Giao diện chọn phương thức thanh toán 49](#_Toc103894096)

[Hình 39: Giao diện chọn ngân hàng thanh toán 49](#_Toc103894097)

[Hình 40: Giao diện nhập thông tin ngân hàng thanh toán 50](#_Toc103894098)

[Hình 41: Giao diện danh sách đơn hàng 51](#_Toc103894099)

[Hình 42: Giao diện chi tiết hóa đơn 52](#_Toc103894100)

[Hình 43: Giao diện chức năng đánh giá 52](#_Toc103894101)

[Hình 44: Flugin messager facebook 53](#_Toc103894102)

[Hình 45: Giao diện trang thống kê 54](#_Toc103894103)

[Hình 46: Giao diện chức năng thêm sản phẩm 54](#_Toc103894104)

[Hình 47: Giao diện danh sách sản phẩm 55](#_Toc103894105)

[Hình 48: Giao diện danh mục màu sắc 56](#_Toc103894106)

[Hình 49: Giao diện danh mục kích thước 56](#_Toc103894107)

[Hình 50: Giao diện danh mục thương hiệu 56](#_Toc103894108)

[Hình 51: Giao diện danh mục loại sản phẩm 56](#_Toc103894109)

[Hình 52: Chức năng nhập sản phẩm 56](#_Toc103894110)

[Hình 53: Thông tin chi tiết hóa đơn nhập 57](#_Toc103894111)

[Hình 54: Danh sách hóa đơn 57](#_Toc103894112)

[Hình 55: Chức năng xem hóa đơn 58](#_Toc103894113)

[Hình 56: Danh sách nhân viên 58](#_Toc103894114)

# MỤC LỤC BẢNG

[Bảng 1: Nội dung nghiên cứu 2](#_Toc105055753)

[Bảng 2: Các tác vụ cơ bản của REST dựa trên phương thức HTTP 3](#_Toc105055754)

[Bảng 3: Một số phương thức (response methods) hỗ trợ hay dùng nhất 15](#_Toc105055755)

[Bảng 4: Use case đăng ký 20](#_Toc105055756)

[Bảng 5: Use case đăng nhập 20](#_Toc105055757)

[Bảng 6: Use case xem thông tin chi tiết sản phẩm 21](#_Toc105055758)

[Bảng 7: Use case quản lý giỏ hàng 21](#_Toc105055759)

[Bảng 8: Use case thanh toán giỏ hàng 22](#_Toc105055760)

[Bảng 9: Use case thanh toán sản phẩm 22](#_Toc105055761)

[Bảng 10: Use case nhập thông tin giao hàng 23](#_Toc105055762)

[Bảng 11: Use case quên mật khẩu 23](#_Toc105055763)

[Bảng 12: Use case xem đơn hàng 24](#_Toc105055764)

[Bảng 13: Use case hủy đơn hàng 24](#_Toc105055765)

[Bảng 14: Use case đăng xuất 25](#_Toc105055766)

[Bảng 15: Use case thêm sản phẩm 25](#_Toc105055767)

[Bảng 16: Use case sửa sản phẩm 26](#_Toc105055768)

[Bảng 17: Use case xóa sản phẩm 26](#_Toc105055769)

[Bảng 18: Các bảng thực thể 29](#_Toc105055770)

[Bảng 19: Bảng khách hàng 30](#_Toc105055771)

[Bảng 20: Bảng địa chỉ 30](#_Toc105055772)

[Bảng 21: Bảng sản phẩm 30](#_Toc105055773)

[Bảng 22: Bảng chi tiết sản phẩm 31](#_Toc105055774)

[Bảng 23: Bảng màu sắc 31](#_Toc105055775)

[Bảng 24: Bảng kích thước 31](#_Toc105055776)

[Bảng 25: Bảng hình ảnh 31](#_Toc105055777)

[Bảng 26: Bảng loại sản phẩm 31](#_Toc105055778)

[Bảng 27: Bảng thương hiệu 32](#_Toc105055779)

[Bảng 28: Bảng khuyến mãi 32](#_Toc105055780)

[Bảng 29: Bảng đánh giá 32](#_Toc105055781)

[Bảng 30: Bảng hình ảnh đánh giá 33](#_Toc105055782)

[Bảng 31: Bảng hóa đơn xuất 33](#_Toc105055783)

[Bảng 32: Bảng chi tiết hóa đơn xuất 33](#_Toc105055784)

[Bảng 33: Bảng giao hàng 34](#_Toc105055785)

[Bảng 34: Bảng hóa đơn nhập 34](#_Toc105055786)

[Bảng 35: Bảng chi tiết hóa đơn nhập 34](#_Toc105055787)

[Bảng 36: Bảng nhân viên 35](#_Toc105055788)

[Bảng 37:Bảng chức vụ 35](#_Toc105055789)

[Bảng 38:Kiểm thử chức năng đăng ký và đăng nhập 60](#_Toc105055790)

[Bảng 39: Kiểm thử chức năng đặt hàng 61](#_Toc105055791)

[Bảng 40:Kiểm thử chức năng tìm kiếm và lọc sản phẩm 62](#_Toc105055792)

[Bảng 41:Kiểm thử chức năng quản lý đơn hàng 63](#_Toc105055793)

[Bảng 42:Kiểm thử chức năng lập hóa đơn 63](#_Toc105055794)

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Diễn giải |
| 1 | CRUD | Create – Read – Update - Delete |
| 2 | HTTP | HyperText Transfer Protocol |
| 3 | API | Application Programming Interface |
| 4 | MVC | Model – View - Controller |
| 5 | URL | Uniform Resource Locator |
| 6 | XML | Extensible Markup Language |
| 7 | JSON | JavaScript Object Notation |
| 8 | REST | Representational State Transfer |
| 9 | URI | Uniform Resource Identifier |
| 10 | ES6 | ECMAScript 6 |

# TÓM TẮT

Đề tài “Xây dựng website thương mại điện tử bán giày dựa trên nền tảng framework ExpressJS và thư viện ReactJS” được xây dựng đề phù hợp với nhu cầu hiện nay và cũng là quảng bá, mở rộng thị trường cho doanh nghiệp. Đề tài được xây dựng trên nền tảng framework ExpressJS và thư viện ReactJS, được quản lý bởi cơ sở dữ liệu MySQL và công cụ lập trình VS Code. Hệ thống được dùng cho 2 nhóm người dùng: khách hàng và quản lý, website giúp khách hàng mua sản phẩm dễ dàng và tiếp cận được nhiều sản phẩm nhanh chóng thông qua tìm kiếm bằng giọng nói, hỗ trợ khách hàng thông qua plugin messenger của facebook. Website còn tích hợp cổng thanh toán trực tuyến VNPAY giúp cho việc thanh toán diễn ra dễ dàng, giảm được nhiều chi phí. Tiếp đến. website hỗ trợ thống kê, quản lý sản phẩm, quản lý hóa đơn, quản lý nhân viên, ... dành cho người quản lý, giúp việc quản lý dễ dàng và linh hoạt.

# ABSTRACT

The topic "Building an e-commerce website selling shoes based on the ExpressJS framework and ReactJS library" was created to meet current needs while also promoting and expanding the business market. The theme is based on the ExpressJS framework and ReactJS library, with MySQL as the database and VS Code as the development tool. The system is utilized by two types of users: customers and managers. The website allows customers to buy things fast and easily via voice search, and provides customer care using Facebook's plugin messenger. The website also includes the VNPAY online payment gateway, which makes payments simple and affordable. Next, Statistics, product management, invoice management, staff management, etc. are all supported by the website. Making management simple and flexible for managers.

# CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN

## 1.1. Đặt vấn đề

Ngày nay, Công nghệ thông tin đã và đang được ứng dụng rất nhiều trong lĩnh vực kinh doanh. Nổi bật trong đó là lĩnh vực thương mại điện tử. Thời gian gần đây, thương mại điện tử đang phát triển vượt bậc bởi sự tiện dụng, giảm thiểu nhiều chi phí hơn so với phương thức bán hàng truyền thống. Mô hình này giúp khách hàng dễ dàng tiếp cận với các doanh nghiệp, Khách hàng có thể liên lạc, trao đổi, tìm hiểu thông tin, mua sắm tại nhà mà không cần đến cửa hàng. Đồng thời, doanh nghiệp cũng giảm bớt chi phí cho việc quảng cáo, thuê mặt bằng

Nắm bắt được tình hình đó, đề tài “Xây dựng website thương mại điện tử bán giày dựa trên nền tảng framework ExpressJS và thư viện ReactJS” đã được lên ý tưởng và phát triển thành một website hoàn thiện với các chức năng tiện ích, giúp khách hàng tìm kiếm, xem thông tin sản phẩm dễ dàng, theo dõi đơn hàng, thanh toán nhanh chóng. Giúp người quản lý nắm bắt được nhu cầu mua sắm của khách hàng, quản lý kho hàng, thống kê chi phí doanh thu chính xác.

## 1.2. Lịch sử giải quyết vấn đề

Lĩnh vực thương mại điện tử hiện nay đang rất phát triển, nhiều website lớn được xây dựng để phục vụ cho nhu cầu mua sắm của khách hàng. Điển hình một số website mua sắm lớn như: tiki.vn, lazada.vn, shopee.vn, fptshop.com.vn, thegioididong.com, … Đã và đang mang lại lượng khách hàng và doanh thu cực kỳ lớn cho doanh nghiệp.

Chính vì vậy, việc xây dựng website ShopShose được xây dựng để nắm bắt xu hướng thời đại, cũng là để doanh ghiệp tiếp cận với số lượng khách hàng mới, giúp cho sản phẩm tiếp cận được với nhiều khách hàng, đem lại doanh thu lớn cho doanh nghiệp.

## 1.3. Mục tiêu đề tài

Website bán giày giúp cho doanh nghiệp hay các chủ cửa hàng giới thiệu và bán sản phẩm một cách nhanh chóng. Giúp người dùng tiếp cận các mặt hàng chỉ với thao tác đơn giản trên máy tính có kết nối internet mà không mất nhiều thời gian hay phải đến trực tiếp cửa hàng.

## 1.4. Đối tượng nghiên cứu

Đối tượng nghiên cứu chính của đề tài là sử dụng ReactJS và ExpressJS để xây dựng website bán quần áo. Ngoài ra, cũng sẽ nghiên cứu thêm về việc xây dựng một kiến trúc hệ thống ứng dụng sử dụng máy chủ dữ liệu (API server) và công cụ quản lý (Web client). Đối tượng nghiên cứu đầy đủ sẽ là một hệ thống gồm 2 thành phần:

- REST API server: xây dựng một server cung cấp các thao tác tạo, thêm, sửa, xóa (CRUD) dữ liệu của hệ thống thông qua HTTP request.

- Web client: xây dựng một ứng dụng web giao tiếp với server thông qua REST API, nhằm mục đích quản lý dữ liệu cốt lõi của hệ thống.

## 1.5. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài chủ yếu là làm việc với React JS để tạo ra website bán giày. Ngoài ra, sẽ sử dụng thư viện UI (User Interface) của React JS là Material UI để xây dựng giao diện website, Axios để giao tiếp với API server, phía backend sẽ sử dụng NodeJS với framework ExpressJS để xây dựng nhanh hệ thống RESTful API.

## 1.6. Phương pháp nghiên cứu

Nội dung nghiên cứu khá phổ biến, do đó phương pháp nghiên cứu chủ yếu là tìm hiểu thông qua tài liệu trang chủ của các thư viện, thông qua các video trên YouTube và làm các ví dụ trực tiếp. Từ đó, sẽ vận dụng các kiến thức đã tiếp thu được vào thực hiện đề tài. Tương ứng với mỗi thành phần của hệ thống sẽ có những phương pháp nghiên cứu khác nhau: API server: Nghiên cứu cơ bản về NodeJS với framework Express để xây dựng một hệ thống RESTful và hệ quản trị cơ sở dữ liệu MySQL. Web client: Nghiên cứu cơ bản về ứng dụng web sử dụng các thư viện react, redux, axios, tích hợp cổng thanh toán VNPAY và material-ui.

## 1.7. Nội dung nghiên cứu

|  |  |  |
| --- | --- | --- |
| STT | Đối tượng | Nội dung |
| 1 | API và RESTful API | - Tìm hiểu về API và RESTful API |
| 2 | NodeJS – ExpressJS | - Tìm hiểu NodeJS với ExpressJS |
| 3 | MVC | - Tìm hiểu mô hình MVC (model-view-controller) |
| 4 | MySQL | - Tìm hiểu hệ quản trị cơ sở dữ liệu MySQL  Cấu hình MySQL trong NodeJS |
| 5 | ReactJS | - Tìm hiểu ReactJS   * Components * State * Props   - Sử dụng webhook  - Sử dụng routing bằng react-router-dom v6 và chuyển hướng đăng nhập. |
| 6 | Redux | - Tìm hiểu về Redux   * State * Action * Container * Dispatcher   - Sử dụng redux vào website |
| 7 | Axios | - Tìm hiểu về axios.  - Trao đổi với API server thông qua axios. |

Bảng 1: Nội dung nghiên cứu

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. Giới thiệu về API và RESTful API

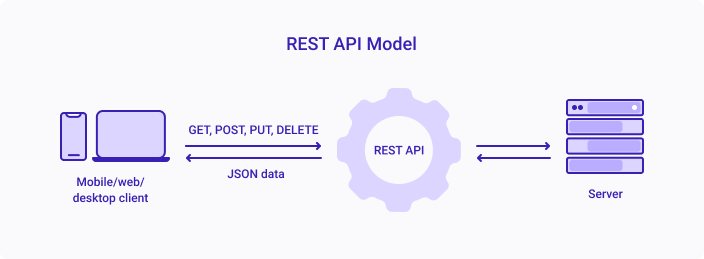
API (Application programming interface) là những quy tắc tương tác để các ứng dụng hoặc thành phần của ứng dụng có thể giao tiếp được với nhau. Mục đích của API là để đơn giản hóa việc lập trình bằng cách trừu tượng hóa đi những cài đặt phức tạp và chỉ phơi bày ra các cách thức để thực hiện hành động mà lập trình viên cần. Trong ngữ cảnh lập trình web, API là cách thức để ứng dụng trao đổi dữ liệu với một dịch vụ trực tuyến. API sẽ cung cấp các chức năng dịch vụ thông qua đường dẫn URL (Uniform Resource Locator) với định nghĩa thông số kỹ thuật như gửi request bằng HTTP (Hypertext Transfer Protocol) và trả về response message theo định dạng XML (Extensible Markup Language) hoặc JSON (JavaScript Object Notation).

REST (Representational State Transfer) là một kiến trúc phần mềm bao gồm các quy tắc để tạo ra dịch vụ web (web service). Một webservice tuân thủ theo kiến trúc REST thì được gọi là RESTful webservice. Webservice này sử dụng giao thức HTTP để triển khai kiến trúc web. Như vậy, RESTful API chính là kiến trúc thiết kế API tuân thủ theo kiến trúc REST thông qua các phương thức của HTTP (GET, POST, PUT, DELETE, ...). Tương ứng với mỗi phương thức HTTP sẽ thực hiện những tác vụ tương ứng:

|  |  |
| --- | --- |
| Phương thức HTTP | Tác vụ |
| GET | Lấy dữ liệu |
| POST | Tạo mới dữ liệu |
| PUT | Cập nhật dữ liệu |
| DELETE | Xóa dữ liệu |

Bảng 2: Các tác vụ cơ bản của REST dựa trên phương thức HTTP

Các tác vụ đọc, tạo, cập nhật, xóa được gọi là CRUD service (Create, Read, Update, Delete). Mỗi tác vụ trên phải được gọi thông qua địa chỉ URI (Uniform Resource Identifier) kèm theo phương thức và payload (có thể có hoặc không, thường là định dạng XML hoặc JSON).



Hình 1: Mô hình hoạt động của REST API

*Nguồn:* [*https://www.rlogical.com/blog/what-is-rest-api-vs-web-api-vs-soap-api*](https://www.rlogical.com/blog/what-is-rest-api-vs-web-api-vs-soap-api)

RESTful API sử dụng giao thức stateless (là một giao thức truyền thông không sử dụng session) và theo tiêu chuẩn nên hệ thống sẽ nhanh, đáng tin cậy và có thể mở rộng dễ dàng. Thông thường, RESTful API sẽ xác thực người dùng khi gửi yêu cầu đối với những tác vụ nguy hiểm như cập nhật hoặc xóa dữ liệu hoặc chỉ cho phép đối với người quản trị.

## 2.2. ReactJS



Hình 2: Logo thư viện ReactJS

*Nguồn: https://200lab.io/blog/reactjs-la-gi*

ReactJS (hay React, React.js) là một thư viện mã nguồn mở (MIT License) dùng để xây dựng giao diện người dùng được viết bằng JavaScript. React được phát hành lần đầu tiên vào năm 2013 bởi Facebook. React hiện nay được duy trì bởi Facebook và cộng đồng lập trình viên. React được sử dụng bởi những doanh nghiệp công nghệ hàng đầu như Facebook, Twitter, Instagram.

Sức mạnh của React nằm ở việc tự động cập nhật lại UI (user interfaces) khi dữ liệu thay đổi với chi phí ít nhất mà không cần phải tải lại toàn bộ trang web. Điều này làm cho ứng dụng phản hồi lại nhanh hơn, tiết kiệm băng thông, tăng tốc ứng dụng và tăng trải nghiệm người dùng. Trong React sẽ có một số khái niệm cơ bản về JSX, components, props, state và hook.

### 2.2.1. JavaScript XML

JSX (JavaScript XML) là một cú pháp mở rộng của JavaScript cho phép định nghĩa các thành phần HTML trong React. Nói một cách đơn giản, cú pháp này cho phép ta viết các HTML tag trực tiếp trong JavaScript.

Trên thực tế, các trình duyệt ngay cả các trình duyệt mới nhất cũng không hỗ trợ cú pháp của JSX. Do đó mã nguồn sử dụng JSX cần được chuyển về JavaScript thông qua một thư viện có tên là Babel (một JavaScript compiler).

### 2.2.2. Components

Components là những thành phần UI được chia nhỏ ra, độc lập và có thể tái sử dụng. Component có thể là những function (stateless) hoặc class (stateful) trong JS.

Functional component: là một hàm Javascript (hoặc ES6) trả về một React element. Theo tài liệu chính thức của React, hàm dưới đây là một component hợp lệ.



Hình 3: Function components hợp lệ

Function này là một component React hợp lệ vì nó nhận một "props" làm tham số và trả về 1 React element.

Hoặc theo ES6 arrow function:



Hình 4: Function components sử dụng arrow function

Functional component cũng được biết tới với cái tên là stateless components. Bởi vì chúng không thể làm nhiều thứ phức tạp như quản lý React State (data) hoặc xử lý vấn đề liên quan tới life-cycle trong functional components. Tuy nhiên, từ phiên bản React 16.8, nhà phát hành giới thiệu tính năng React Hooks. Với Hooks, chúng ta có thể sử dụng state và những features khác trong functional components.

Class components: Các Class components là những class ES6. Chúng phức tạp hơn functional components.

Class components còn có:

- Phương thức khởi tạo, có hàm về vòng đời component, hàm render().

- State (dữ liệu ứng dụng).



Hình 5: Ví dụ class components

- Tóm lại, một class components là một class kế thừa từ React.Component, có thể nhận props (trong hàm khởi tạo) nếu cần.

### 2.2.3. Props và State

Props là những thuộc tính được truyền vào một component và chỉ có thể đọc. Ví dụ như thẻ Post có thuộc tính được truyền vào là title. Truy xuất bằng cú pháp props.title sẽ cho giá trị là “My Post Title”.



Hình 6: Props

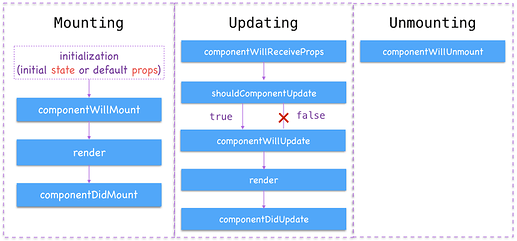
State là trạng thái thuộc về chính component đó, được quản lý bởi chính nó và không được truy xuất từ bên ngoài. Chỉ có thể sử dụng state khi dùng stateful component.

### 2.2.4. LifeCycle

LifeCycle là một vòng đời của một React component từ lúc được render lần đầu tiên và mỗi lần render lại (mounting) và khi gỡ bỏ component (unmounting).

Có 2 phương thức được tự động gọi khi sự kiện mounting (component được render lần đầu tiên hoặc được render lại) và unmounting (component bị gỡ bỏ) xảy ra lần lượt là componentDidMount và componentWillUnmount.

Có thể ghi đè 2 phương thức này khi sử dụng stateful component (class).



Hình 7: LifeCycle trong ReactJS

*Nguồn:*[*https://stories.magestore.com/t/tim-hi-u-v-lifecycle-c-a-class-trong-react-js/1427*](https://stories.magestore.com/t/tim-hi-u-v-lifecycle-c-a-class-trong-react-js/1427)

### 2.2.5. Hook

Hook là tính năng được thêm vào React ở phiên bản 16.8. Cho phép sử dụng một số tính năng chỉ có ở stateful component (class) khi dùng stateless component (function) như state (useState), life cycle (useEffect), ...

## 2.3. Thư viện Redux



Hình 8: Logo Redux

*Nguồn:* [*https://redux.js.org/*](https://redux.js.org/)

Redux là một thư viện JavaScript mã nguồn mở dùng để quản lý state của ứng dụng, được sử dụng phổ biến với React hoặc Angular để xây dựng giao diện người dùng. Redux được tạo bởi Dan Abramov và Andrew Clark vào năm 2015.

State trong redux sẽ có các tính chất sau:

− Từ một nguồn duy nhất: state của ứng dụng được là một state toàn cục được lưu trữ thành một object tree.

− State chỉ có thể đọc: state không thể thay đổi trực tiếp mà chỉ có thể phát động bằng một action (object) mô tả những gì đã xảy ra.

− State được thay đổi bởi những reducer (JS function): khi muốn thay đổi state sẽ phát động ra một action, reducer sẽ tiếp nhận và xử lý thay đổi state tương 19 ứng. Vì reducer chỉ là những hàm js bình thường, có thể truyền vào dữ liệu, gọi theo thứ tự hoặc tái sử dụng các reducer.

Redux được thiết kế cho việc quản lý tập trung, chia sẻ state giữa các component trở nên đơn giản, nhất quán. Để sử dụng redux với react cần phải nắm một số khái niệm cơ bản sau:

− Action: là những thông tin payload được dùng để gửi dữ liệu đến store. Hiểu đơn giản, action chính là một event được gửi lên store kèm theo dữ liệu và loại action sẽ được thực hiện.

− Reducer: là những hàm thuần túy thực hiện việc nhận action được gửi lên và lấy state hiện tại của ứng dụng, xử lý và trả về một state mới. Global state chính là một object tree lưu trữ toàn bộ state của ứng dụng, action gửi lên sẽ mô tả những gì xảy ra. Khi đó, reducer sẽ tiếp nhận action và lấy một nhánh state nhỏ tương ứng từ global state, sau đó tạo ra một nhánh state mới dựa trên action và cập nhật vào global state.

− Store: là một object kết nối action và reducer lại với nhau, chịu trách nhiệm lưu giữ state của ứng dụng, cho phép truy xuất và cập nhật state thông qua dispatch (là một dịch vụ của store cho phép gửi action thông qua nó).

## 2.4. Thư viện axios

Axios là một thư viện HTTP client dựa trên Promise dùng để gửi các request HTTP bất đồng bộ đến REST endpoint để sử dụng các dịch vụ CRUD.

Promise là một cải tiến để loại bỏ try catch, callback rườm rà trong xử lý bất đồng bộ, thay vào đó là then – catch. Hiểu đơn giản là khi thực hiện một tác vụ bất đồng bộ, sau khi thực hiện thành công thì “then” sẽ được gọi, ngược lại khi có lỗi phát sinh thì “catch” sẽ được gọi. Tương tự như promise, axios cũng có thể sử dụng theo kiểu gửi request lồng nhau.

Ngoài ra, axios còn hỗ trợ interceptor dùng để thực hiện một số công việc khác trước khi gửi request hoặc ngay khi nhận được response. Ví dụ như tạo mới access token khi nhận được response thông báo lỗi “401 – Unauthorized”.

## 2.5. Thư viện React UI – Material UI

Material UI là một thư viện các React Component đã được tích hợp thêm cả Google's Material Design. Các component trong Material UI là những React component, hỗ trợ hầu hết các thành phần cốt yếu để tạo dựng nên một trang web hoàn chỉnh. Khi sử dụng Material UI thì hầu như không cần phải cài thêm một thư viện UI bổ sung nào nữa thì cũng đủ để đáp ứng nhu cầu của dự án đưa ra.

Dưới đây là danh sách các component mà Material UI hỗ trợ:

Layout: Box, Container, Gird, Grid List, Hidden.

Input: Button, Button Group, Checkbox, Floating Action Button, Date/Time, Radio, Select, Slider, Switch, TextField, Transfer List.

Navigation: Bottom Navigation, Breadcrum, Drawer, Link, Menu, Stepper, Tabs.

Surfaces: App bar, Paper, Card, Accordion.

Feedback: Progress, Dialog, Snackbar, Backdrop.

Data Display: Avatar, Badge, Chip, Divider, Icons, Material Icons, List, Table, Tooltip, Typography.

Utils: Click Away Listener, CSS Baseline, Modal, No SSR, Popover, Popper, Portal, Textarea Autosize, Transition, useMediaQuery.

Lab: Alert, Autocomplete, Data Grid, Pagination, Rating, Skeleton, Speed Dial, Timeline, Toggle Button, Tree View.

## 2.6. Cơ sở dữ liệu MySQL



Hình 9: Logo cơ sở dữ liệu MySQL

*Nguồn:* [*https://vi.cleanpng.com/png-gwfs4v/*](https://vi.cleanpng.com/png-gwfs4v/)

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở miễn phí, được những doanh nghiệp lớn nhất thế giới như Facebook, Google, Adobe sử dụng để tiết kiệm thời gian và chi phí cho việc quản lý dữ liệu hiệu quả. MySQL được sáng lập bởi Michael "Monty" Widenius và David Axmark năm 1995. Hiện nay được phát triển bởi tập đoàn Oracle.

MySQL hỗ trợ trên nhiều hệ điều hành: Windows, Linux, MacOS, ...

MySQL cũng được công nhận bởi DB-Engines là hệ quản trị cơ sở dữ liệu của năm 2019.

## 2.7. NodeJS



Hình 10: Logo NodeJS

*Nguồn:* [*https://nodejs.org/en/about/resources/*](https://nodejs.org/en/about/resources/)

NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.

NodeJS được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau: OS X, Microsoft Windows, Linux.

### 2.7.1. Blocking I/O và Nonblocking I/O

I/O là quá trình giao tiếp (lấy dữ liệu vào, trả dữ liệu ra) giữa một hệ thống thông tin và môi trường bên ngoài. Với CPU, thậm chí mọi giao tiếp dữ liệu với bên ngoài cấu trúc chip như việc nhập/ xuất dữ liệu với memory (RAM) cũng là tác vụ I/O.

Blocking I/O: Yêu cầu thực thi một IO operation, sau khi hoàn thành thì trả kết quả lại. Process/Theard gọi bị block cho đến khi có kết quả trả về hoặc xảy ra ngoại lệ.

Nonblocking I/O: Yêu cầu thực thi IO operation và trả về ngay lập tức (timeout = 0). Nếu operation chưa sẵn sàng để thực hiện thì thử lại sau. Tương đương với kiểm tra IO operation có sẵn sàng ngay hay không, nếu có thì thực hiện và trả về, nếu không thì thông báo thử lại sau.

### 2.7.2. Synchronous và Asynchronous

Synchronous: Các sự kiện diễn ra theo thứ tự. Một sự kiện chỉ được bắt đầu khi sự kiện trước kết thúc.

Asynchronous: Không theo thứ tự, các hành động có thể xảy ra đồng thời hoặc chí ít, mặc dù các hành động bắt đầu theo thứ tự nhưng kết thúc thì không. Một hành động có thể bắt đầu (và thậm chí kết thúc) trước khi hành động trước đó hoàn thành.

Libuv – một thư viện multi-platform hỗ trợ asynchronous I/O giúp Nodejs mang lại cơ chế Asynchronous Event-Driven Non-Blocking I/O.

### 2.7.3. Callback

Callback là một hàm sẽ được thực hiện sau khi một hàm khác đã thực hiện xong.

Trong JavaScript, hàm là đối tượng. Do đó, các hàm có thể lấy các hàm làm đối số và có thể được trả về bởi các hàm khác. Các hàm thực hiện điều này được gọi là higher - order function (Hàm bậc cao hơn). Bất kỳ hàm nào được truyền dưới dạng đối số được gọi là hàm callback.

Javascript là một ngôn ngữ lập trình hướng sự kiện và bất đồng bộ nên callback function đóng vai trò rất quan trọng.

ES6 định nghĩa 3 state cho một lời gọi hàm không đồng bộ:

* Pending: hàm đang được thực hiện và chưa trả về kết quả. Trong lúc này, nếu cố tình console.log biến kết quả sẽ nhận được output < pending > .
* Fulfilled: hàm đã thực hiện xong – thành công và trả về kết quả.
* Rejected: hàm đã thực hiện xong – không thành công. Thường thì sẽ bắt exception tại bước này.

Máy chủ NodeJS có thể nhận rất nhiều các yêu cầu (request) từ rất nhiều người dùng. Vì vậy để nâng cao khả năng phục vụ, tất cả các API của NodeJS được thiết kế hỗ trợ Callback.

### 2.7.4. Event Loop

NodeJS là một ứng dụng đơn luồng (Single Thread), nó hoạt động phía trên một nền tảng được viết bởi C++, nền tảng này sử dụng đa luồng (Multi-Thread) để thực hiện đồng thời các nhiệm vụ.

Mỗi yêu cầu (request) từ phía người dùng được NodeJS coi là một sự kiện (event), chúng được đặt vào một Event Queue (Hàng đợi sự kiện). NodeJS sử dụng quy tắc FIFO (First In First Out), điều này có nghĩa là những yêu cầu đến trước sẽ được xử lý trước.

Event Loop là một vòng lặp vô tận, nó sẽ chuyển các yêu cầu sang Thread Pool (Bể chứa các luồng), đồng thời mỗi yêu cầu sẽ được đăng ký một hàm Callback. Khi một yêu cầu được xử lý xong, hàm Callback tương ứng sẽ được gọi thực thi.

Thread Pool là một chương trình viết bằng ngôn ngữ C++, nó hỗ trợ đa luồng (Multi Threads), chính vì vậy tại đây các yêu cầu sẽ được xử lý trên các luồng khác nhau. NodeJS cũng hỗ trợ đa tiến trình (Multi Processes), điều này có nghĩa là chúng có thể được thực thi trên các lõi (Core) khác nhau.

Khi một yêu cầu được xử lý xong, NodeJS sẽ gọi hàm Callback (Đã được đăng ký cho yêu cầu này) để thực thi nó.

Nếu mỗi kết nối tới Server đều mở ra một luồng (Thread) sẽ rất tốn bộ nhớ. Điều này đã được chứng thực khi so sánh Apache và Nginx (Hai Web Server triển khai các ứng dụng PHP). Apache đã tiêu tốn bộ nhớ hơn rất nhiều so với Nginx.

NodeJS giống với Nginx là chúng chỉ sử dụng một luồng đơn (Single thread) để đón tiếp các kết nối từ phía người dùng, và coi mỗi yêu cầu của người dùng là một sự kiện.

Các hoạt động I/O rất tốn tài nguyên của hệ thống, vì vậy NodeJS quản lý chặt chẽ việc sử dụng các hoạt động I/O. Vì vậy chỉ cần sử dụng Callback khi bạn thực thi các nhiệm vụ liên quan tới I/O.

### 2.7.5. Module

Node.js sử dụng kiến trúc Module để đơn giản hóa việc tạo ra các ứng dụng phức tạp. Module là giống như các thư viện trong C, C#, Java, … Mỗi module chứa một tập các hàm chức năng có liên quan đến một "đối tượng" của Module.

Các module được giữ tách biệt riêng với nhau, tách riêng với code base, khi nào cần sử dụng những cái nào thì gọi chúng ra và kết hợp lại với nhau tùy logic xử lý của bạn.

### 2.7.6. NPM - Node Package Manager

NPM viết tắt của Node Package Manager là một công cụ tạo và quản lý các thư viện lập trình Javascript cho NodeJS. Trong cộng đồng Javascript, các lập trình viên chia sẻ hàng trăm nghìn các thư viện với các đoạn code đã thực hiện sẵn một chức năng nào đó. Nó giúp cho các dự án mới tránh phải viết lại các thành phần cơ bản, các thư viện lập trình hay thậm chí cả các framework.

Các loại package

Dựa theo chức năng mà ta chia package ra làm 2 loại, đó là Simple dependencies và Development dependencies.

Simple dependencies là những package bắt buộc phải có trong quá trình chạy sản phẩm. Khi cài đặt Simple dependencies, Npm sẽ tự động cài đặt tất cả các dependencies cần thiết.

Development dependencies là những package bắt buộc khi phát triển cũng như phát hành sản phẩm. Khi cài đặt Development dependencies, Npm sẽ chỉ cài đặt các dependencies cần thiết.

## 2.8. ExpressJS Framework



Hình 11: Logo ExpressJS

*Nguồn:* [*https://itnavi.com.vn/blog/expressjs-la-gi/?amp*](https://itnavi.com.vn/blog/expressjs-la-gi/?amp)

Expressjs là một framework được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Expressjs hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

Định tuyến (Routing)

Routing trong Node.js là một khái niệm nói đến việc xác định ứng dụng sẽ đáp ứng như thế nào khi người dùng tạo một request đến một endpoint (Điểm cuối) cụ thể nào đó. Điểm cuối đó thường là một URI hoặc một đường dẫn (Path) với một Request method (POST, PUT, GET, ... ) cụ thể.

Cấu trúc định tuyến cơ bản:

app.METHOD (Path, Handler...)

Trong đó:

* app : là một instance của express
* METHOD: là một HTTP Method

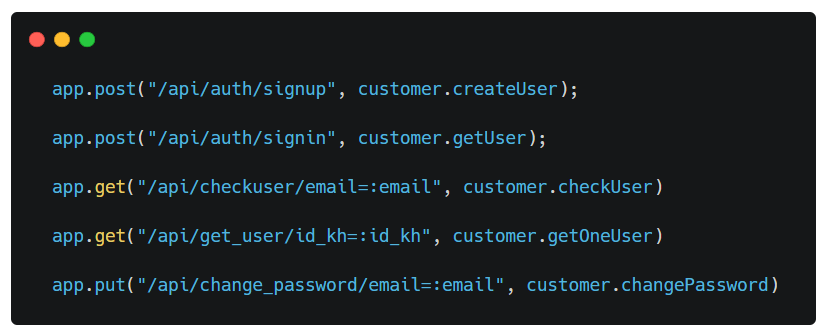
Express hỗ trợ rất nhiều loại HTTP methods khác nhau, bao gồm: get, post, put, head, delete, options, trace, copy, lock, mkcol, move, purge, unlock, report, mkactivity, checkout, merge, m-search, notify, subscribe, unsubscribe, patch.

* Path: là một đường dẫn trên máy chủ

Route path có thể là một chuỗi thông thường (String) hoặc là một chuỗi có ký hiệu biểu thức chính quy (string patterns) hoặc là một biểu thức chính quy (regular expressions)

* Handler: là một function sẽ thực thi khi một route được trùng khớp

Đơn giản là một hoặc nhiều function sẽ được gọi khi một route trùng khớp để đáp ứng một yêu cầu nào đó. Lưu ý các handler sẽ được gọi đúng theo thứ tự truyền vào.



Hình 12: Ví dụ cấu trúc route

Route parameters

Route parameters là những vị trí trên URL được đánh dấu bằng cách đặt tên, mục đích là để lấy ra các giá trị tương ứng. Tất cả cá giá trị đối số sẽ được đặt vào đối tượng req trong thuộc tính params. Với tên thuộc tính trùng khớp với từ khóa được xác định trên URL.

Ví dụ, chúng ta định nghĩa một path là /api/get\_user/id\_kh=:id\_kh. Thì ở đây :id\_kh chính là một route param. Khi đó nếu người dùng truy cập đường dẫn như là /api/get\_user/id\_kh=KH01 ta lấy ra được :id\_kh = KH01 và giá trị này sẽ nằm ở req.params.id\_kh.

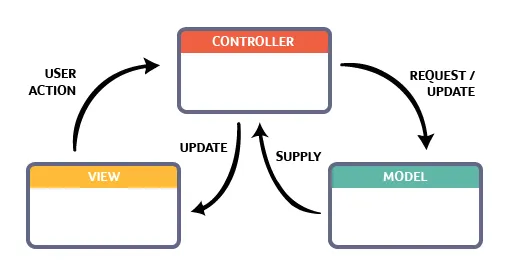
Response methods

Sau việc tiếp nhận và xử lý, thì việc tiếp theo đó là đáp ứng (Response). Trong express định nghĩa sẵn một số phương thức hỗ trợ hay dùng nhất là:

|  |  |
| --- | --- |
| Tên phương thức | Ý nghĩa |
| res.json() | Trả về một dữ liệu dạng JSON |
| res.redirect() | Chuyển hướng đến một đường dẫn nào đó |
| res.render() | Trả về một view template |
| res.send() | Gửi dữ liệudạng text |

Bảng 3: Một số phương thức (response methods) hỗ trợ hay dùng nhất

## 2.9. Mô hình MVC



Hình 13: Mô hình MVC

*Nguồn:* [*https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/*](https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/)

* Model: có nhiệm vụ thao tác với cơ sở dữ liệu, nghĩa là nó sẽ chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu và controller sẽ thông qua các hàm, phương thức đó để lấy dữ liệu rồi gửi qua View
* View: có nhiệm vụ tiếp nhận dữ liệu từ controller và hiển thị nội dung sang các đoạn mã HTML, bạn có thể hiểu nôm na đây người ta còn gọi là thành phần giao diện.
* Controller: đóng vai trò trung gian giữa Model và View. Nó có nhiệm vụ tiếp nhận yêu cầu từ client sau đó xử lý request, load model tương ứng và gửi data qua view tương ứng rồi trả kết quả về cho client.
* Luồng xử lý:

Ví dụ: URL là http://localhost:3000/shop/id=SP033 thì hệ thống MVC sẽ xử lý như sau:

* Bước 1: Dựa vào yêu cầu lấy sản phẩm có id=SP033 nên Controller sẽ gọi tới một hàm lấy dữ liệu theo id trong Model
* Bước 2: Sau khi có dữ liệu Controller sẽ gửi qua View, lúc này View có nhiệm vụ xử lý dữ liệu.
* Bước 3: Sau khi View kết thúc thì Controller sẽ gửi trả dữ liệu của View về cho Client và người dùng sẽ xem được nội dung của sản phẩm có id=SP033

# CHƯƠNG 3: NỘI DUNG

## 3.1. Đặc tả

Website có nhiệm vụ là giới thiệu sản phẩm, chức năng đặt hàng và theo dõi đơn hàng thông minh với giao diện thân thiện, dễ sử dụng.

Người quản trị sẽ thêm các sản phẩm lên website với đầy đủ các thông tin về sản phẩm như: tên sản phẩm, thông tin sản phẩm, kích thước, màu sắc, thương hiệu, loại sản phẩm, giá bán, hình ảnh sản phẩm. Khi khách hàng truy cập vào website của cửa hàng, khách hàng có thể lựa chọn hoặc tìm kiếm sản phẩm mình cần mua. Khách hàng muốn mua ngay hoặc thêm sản phẩm vào giỏ hàng để thanh toán sau, khách hàng cần phải đăng nhập để thêm sản phẩm vào giỏ hàng và thanh toán sản phẩm, khách hàng có thể đăng ký tài khoản nếu chưa có tài khoản.

Sau khi khách hàng thêm các sản phẩm cần mua vào giỏ hàng, khách hàng có thể vào lại giỏ hàng để xem lại các mặt hàng, nếu có sản phẩm nào không mua thì khách hàng có thể xóa sản phẩm đó khỏi giỏ hàng.

Khi khách hàng đã lựa chọn xong, khách hàng sẽ tiến hành đặt hàng và thanh toán qua hai hình thức là thanh toán khi nhận hàng và thanh toán trực tuyến qua VNPAY. Khách hàng có thể hủy đơn hàng nếu quản lý chưa xác nhận đơn hàng. Khi khách hàng gửi yêu cầu đặt hàng hệ thống sẽ kiểm tra thông tin khách hàng có hợp lệ hay không, nếu không hợp lệ người quản trị sẽ liên hệ lại khách hàng hoặc hủy đơn hàng đó. Ngược lại, thông tin đơn hàng sẽ được cập nhật trong danh sách đơn hàng của khách hàng, khách hàng cũng có thể theo dõi đơn hàng thông qua thanh trạng thái của đơn hàng đó. Nếu khách hàng cần thêm thông tin thì có thể nhắn tin qua plugin messenger.

Quản trị viên sẽ tương tác trên giao diện dành cho quản trị viên. Hệ thống cho phép quản trị viên quản lý được số lượng sản phẩm được nhập vào, thêm sản phẩm, sửa và xóa sản phẩm. Nhận thông tin khách hàng mua sản phẩm, quản lý danh sách khách hàng, thông tin hóa đơn của khách hàng, thống kê sản số hóa đơn, doanh thu và sản phẩm bán chạy nhất.

## 3.2. Các chức năng chính của hệ thống

1. Khách hàng

* Khách hàng chưa đăng ký tài khoản: có thể truy cập website để đăng ký tài khoản, tìm kiếm sản phẩm bằng giọng nói, lọc sản phẩm, xem thông tin của sản phẩm và có thể nhắn tin với cửa hàng để được hỗ trợ, … Nhưng không thể thêm sản phẩm vào giỏ hàng và mua sản phẩm.
* Khách hàng đã có tài khoản: có thể truy cập website và đăng nhập, nếu có tài khoản khách hàng sẽ có tất cả các chức năng của người chưa có tài khoản, thêm sản phẩm vào giỏ hàng và thanh toán được sản phẩm. việc thanh toán sản phẩm sẽ được thực hiện thông qua 2 hình thức: thanh toán khi nhận sản phẩm và thanh toán trực tuyến qua VNPAY.

1. Quản lý

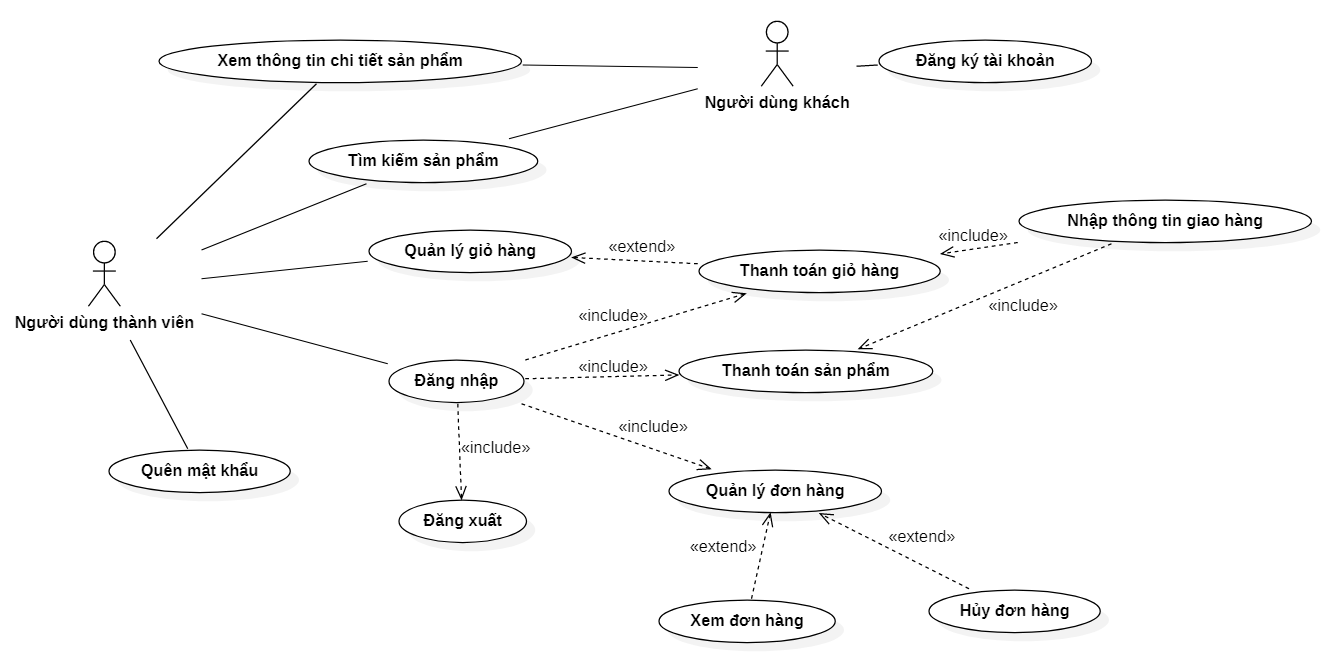
* Người quản lý sẽ nắm tất cả các quyền trong hệ thống như xem thống kê doanh thu, thêm sửa xóa và khuyến mãi sản phẩm, nhập hóa đơn, thêm sửa và xóa nhân viên, phân quyền cho nhân viên, thêm các danh mục, xác nhận và hủy hóa đơn. Ngoài ra, quản lý có thể phân quyền cho nhân viên phụ trách với các chức vụ khác nhau:
  + Nhân viên: xem đơn hàng.
  + Nhân viên kho: thêm sản phẩm mới và nhập hóa đơn.
  + Giao hàng: chấp nhận đơn hàng hoặc không nhận đơn hàng đó.

## 3.3. Môi trường vận hành

* Website hoạt động tốt trên trình duyệt Chrome, Edge, …
* Chương trình tạo máy chủ Web (Web server) XAMPP.
* Yêu cầu kết nối Internet.

## 3.4. Use case diagram

### 3.4.1. Use case diagram người dùng thành viên và người dùng khách

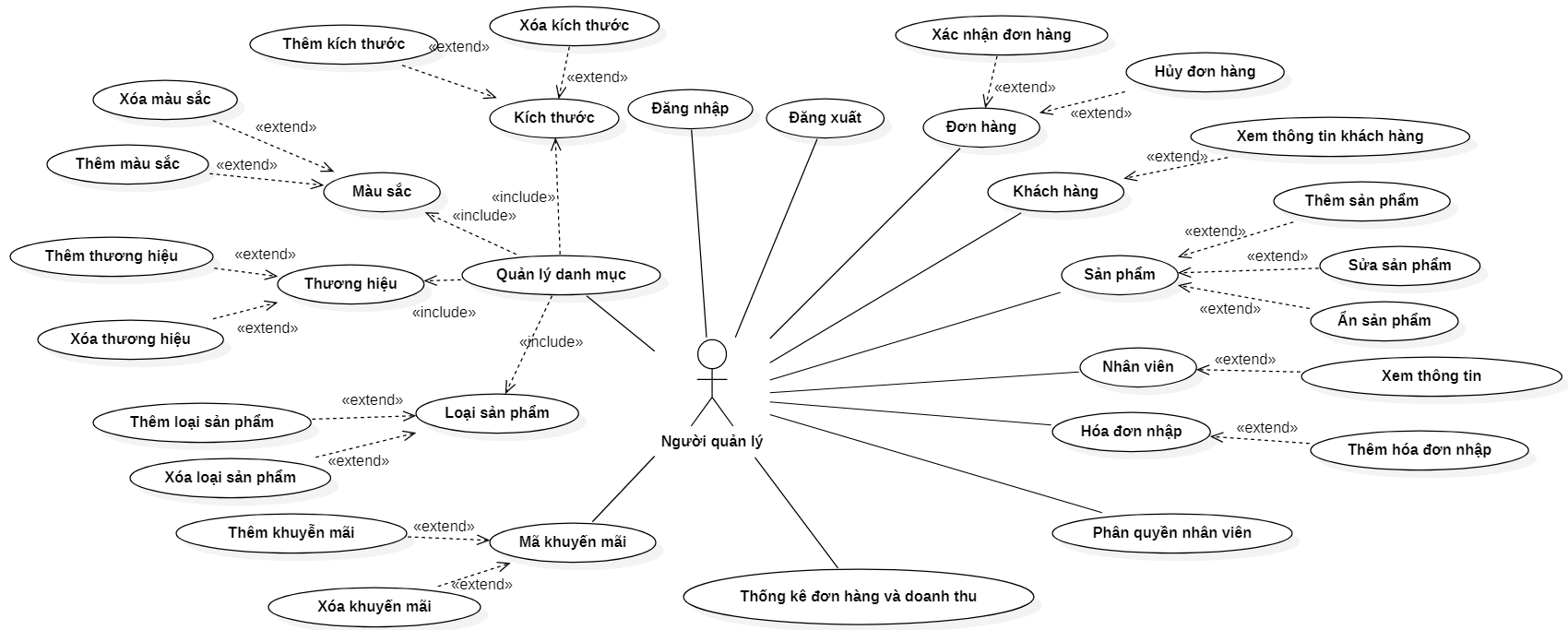


Hình 14: Sơ đồ use case người dùng thành viên và người dùng khách

Người dùng khách (khách hàng không có tài khoản): khách hàng có thể thực hiện các chức năng sau: Đăng ký tài khoản, xem thông tin chi tiết sản phẩm, tìm kiếm sản phẩm.

Người dùng thành viên (khách hàng đã có tài khoản): khách hàng có thể thực hiện các chức năng sau: Đăng nhập tài khoản, xem thông tin chi tiết sản phẩm, tìm kiếm sản phẩm, thêm sản phẩm vào giỏ hàng, thanh toán sản phẩm, thanh toán giỏ hàng, thêm thông tin giao hàng, xem lịch sử đơn hàng và hủy đơn hàng khi quản lý chưa xác nhận đơn hàng.

### 3.4.2. Use case diagram quản lý



Hình 15: Sơ đồ use case người quản lý

Sơ đồ usecase của quản lý. Trong đó, quản lý có toàn quyền sử dụng các chức năng của hệ thống. quản lý có thể đăng nhập, đăng xuất vào hệ thống. quản lý các danh mục màu sắc, kích thước, loại sản phẩm, mã khuyến mãi, thông tin đơn hàng, hóa đơn nhập, thông tin khách hàng, thông tin sản phẩm, nhân viên và phân quyền nhân viên. Ngoài ra, quản lý còn xem được thống kê doanh thu của cửa hàng, lợi nhuận, số lượng sản phẩm tồn kho và đã bán.

### 3.4.3. Mô tả chi tiết các use case

Use case đăng ký

|  |  |
| --- | --- |
| Tên Use case | Đăng ký |
| Mô tả | Cung cấp chức năng cho phép đăng ký khách hàng thành viên |
| Tác nhân chính | Người dùng khách |
| Tiền điều kiện | Không cần đăng nhập |
| Cách kích hoạt | Bấm nút “Đăng ký” trên giao diện website |
| Luồng xử lý | 1. Nhập email 2. Nhập mật khẩu đăng ký 3. Bấm nút “Đăng ký” 4. Hệ thống gửi mã xác minh đến email 5. Nhập mã xác minh 6. Hệ thống chuyển về trang đăng nhập cho khách hàng đăng nhập |
| Dòng sự kiện lỗi | 1, 3. Hệ thống sẽ yêu cầu khách hàng nhập lại |

Bảng 4: Use case đăng ký

Use case đăng nhập

|  |  |
| --- | --- |
| Tên Use case | Đăng nhập |
| Mô tả | Cung cấp chức năng cho phép khách hàng đăng nhập hệ thống |
| Tác nhân chính | Người quản lý, khách hàng |
| Tiền điều kiện | Khách hàng đã đăng ký vào hệ thống |
| Cách kích hoạt | Bấm nút “Đăng nhập” trên giao diện website |
| Luồng xử lý | 1. Khách hàng nhập thông tin tài khoản vào các trường tương ứng 2. Bấm nút “Đăng nhập” 3. Hệ thống xác thực việc đăng nhập |
| Dòng sự kiện lỗi | 3. Hệ thống thông báo lỗi, yêu cầu khách hàng đăng nhập lại |

Bảng 5: Use case đăng nhập

Use case xem thông tin chi tiết sản phẩm

|  |  |
| --- | --- |
| Tên Use case | Xem thông tin chi tiết sản phẩm |
| Mô tả | Cung cấp chức năng cho phép khách hàng xem các sản phẩm của cửa hàng |
| Tác nhân chính | Tất cả các tác nhân |
| Tiền điều kiện |  |
| Cách kích hoạt | Truy cập website |
| Luồng xử lý | 1. Khách hàng truy cập vào website xem sản phẩm 2. Khách hàng chọn sản phẩm |
| Dòng sự kiện lỗi |  |

Bảng 6: Use case xem thông tin chi tiết sản phẩm

Use case quản lý giỏ hàng

|  |  |
| --- | --- |
| Tên Use case | Cập nhật giỏ hàng |
| Mô tả | Cung cấp chức năng cho phép khách hàng thêm, xóa các sản phẩm trong giỏ hàng của mình |
| Tác nhân chính | Khách hàng |
| Tiền điều kiện | Khách hàng đã đăng nhập, chọn màu sắc và kích thước |
| Cách kích hoạt | Nhấn nút “Thêm vào giỏ hàng” để thêm sản phẩm vào giỏ hàng.  Vào giỏ hàng nhấn nút xóa của sản phẩm trong giỏ hàng để xóa sản phẩm ra khỏi giỏ hàng. |
| Luồng xử lý | 1. Khách hàng truy cập vào thông tin chi tiết của sản phẩm 2. Nhấn biểu tượng giỏ hàng để thêm sản phẩm vào giỏ hàng hoặc vào giỏ hàng nhấn nút “xóa” của sản phẩm trong giỏ hàng để xóa sản phẩm ra khỏi giỏ hàng. 3. Hệ thống cập nhập thông tin giỏ hàng |
| Dòng sự kiện lỗi |  |

Bảng 7: Use case quản lý giỏ hàng

Use case thanh toán giỏ hàng

|  |  |
| --- | --- |
| Tên Use case | Thanh toán giỏ hàng |
| Mô tả | Cung cấp chức năng cho phép khách hàng thanh toán các sản phẩm trong giỏ hàng của mình qua cổng thanh toán VNPAY hoặc thanh toán khi nhận hàng |
| Tác nhân chính | Khách hàng |
| Tiền điều kiện | Khách hàng đã đăng nhập |
| Cách kích hoạt | Nhấn nút “Thanh toán” để thanh toán. |
| Luồng xử lý | 1. Khách hàng truy cập vào giỏ hàng 2. Nhấn nút “Thanh toán”. 3. Nhập thông tin giao hàng. 4. Tiến hành thanh toán. 5. Hệ thống hiển thị thông báo “Chờ xác nhận”. |
| Dòng sự kiện lỗi |  |

Bảng 8: Use case thanh toán giỏ hàng

Use case thanh toán sản phẩm

|  |  |
| --- | --- |
| Tên Use case | Thanh toán sản phẩm |
| Mô tả | Cung cấp chức năng cho phép khách hàng thanh toán các sản phẩm trong giỏ hàng của mình qua cổng thanh toán VNPAY hoặc thanh toán khi nhận hàng |
| Tác nhân chính | Khách hàng |
| Tiền điều kiện | Khách hàng đã đăng nhập |
| Cách kích hoạt | Nhấn nút “Mua ngay” để thanh toán. |
| Luồng xử lý | 1. Khách hàng truy cập vào chi tiết sản phẩm. 2. Chọn màu sắc và kích thước, nhấn nút “Mua ngay”. 3. Nhập thông tin giao hàng. 4. Tiến hành thanh toán. 5. Hệ thống hiển thị thông báo “Chờ xác nhận”. |
| Dòng sự kiện lỗi |  |

*Bảng 9: Use case thanh toán sản phẩm*

Use case nhập thông tin giao hàng

|  |  |
| --- | --- |
| Tên Use case | Nhập thông tin giao hàng |
| Mô tả | Cung cấp chức năng cho phép khách hàng thêm thông tin giao hàng |
| Tác nhân chính | Khách hàng |
| Tiền điều kiện | Khách hàng đã đăng nhập |
| Cách kích hoạt | Nhấn nút “Chọn địa chỉ” để thanh toán. |
| Luồng xử lý | 1. Khách hàng truy cập vào chi tiết sản phẩm. 2. Chọn màu sắc và kích thước, nhấn nút “Mua ngay”. 3. Nhập thông tin giao hàng. 4. Nhấn nút “chọn địa chỉ” |
| Dòng sự kiện lỗi |  |

Bảng 10: Use case nhập thông tin giao hàng

Use case quên mật khẩu

|  |  |
| --- | --- |
| Tên Use case | Quên mật khẩu |
| Mô tả | Cung cấp chức năng cho phép khách hàng thay đổi mật khẩu tài khoản của mình |
| Tác nhân chính | Khách hàng |
| Tiền điều kiện | Khách hàng chưa đăng nhập |
| Cách kích hoạt | Nhấn nút “quên mật khẩu” để thay đổi mật khẩu |
| Luồng xử lý | 1. Khách hàng truy cập trang đăng nhập 2. Khách hàng nhấn chọn “Đổi mật khẩu”. 3. Nhập email và nhấn nút “gửi”. 4. Nhập mật khẩu, và mã xác thực được gửi đến email |
| Dòng sự kiện lỗi | 4. Khách hàng nhập sai mã xác thực sẽ hiển thị thông báo |

Bảng 11: Use case quên mật khẩu

Use case xem đơn hàng

|  |  |
| --- | --- |
| Tên Use case | Xem đơn hàng |
| Mô tả | Cung cấp chức năng cho phép khách hàng xem thông tin các đơn hàng của mình |
| Tác nhân chính | Khách hàng |
| Tiền điều kiện | Khách hàng đã đăng nhập |
| Cách kích hoạt | Nhấn nút “Đơn hàng” để xem các đơn hàng |
| Luồng xử lý | 1. Khách hàng truy cập “Tài khoản”. 2. Khách hàng nhấn chọn “Đơn hàng”. 3. Hệ thống hiển thị các đơn hàng của khách hàng. |
| Dòng sự kiện lỗi |  |

Bảng 12: Use case xem đơn hàng

Use case hủy đơn hàng

|  |  |
| --- | --- |
| Tên Use case | Hủy đơn hàng |
| Mô tả | Cung cấp chức năng cho phép khách hàng hủy các đơn hàng khi quản lý chưa xác nhận đơn hàng. |
| Tác nhân chính | Khách hàng |
| Tiền điều kiện | Khách hàng đã đăng nhập |
| Cách kích hoạt | Nhấn nút “Đơn hàng” để xem các đơn hàng |
| Luồng xử lý | 1. Khách hàng truy cập “Tài khoản”. 2. Khách hàng nhấn chọn “Đơn hàng”. 3. Chọn đơn hàng cần hủy (nếu chủ cửa hàng chưa xác nhận). 4. Nhấn nút “Hủy” để hủy đơn hàng. 5. Hệ thống cập nhật trạng thái đơn hàng của khách hàng. |
| Dòng sự kiện lỗi |  |

Bảng 13: Use case hủy đơn hàng

Use case đăng xuất

|  |  |
| --- | --- |
| Tên Use case | Đăng xuất |
| Mô tả | Cung cấp chức năng cho phép người dùng đăng xuất khỏi tài khoản |
| Tác nhân chính | Khách hàng thành viên, người quản lý |
| Tiền điều kiện | Khách hàng đã đăng nhập vào hệ thống |
| Cách kích hoạt | Bấm nút “Đăng xuất” trên giao diện website |
| Luồng xử lý | 1. Khách hàng truy cập tài khoản 2. Bấm nút “Đăng xuất” 3. Hệ thống xác thực việc đăng xuất 4. Hệ thống chuyển về trang chủ |
| Dòng sự kiện lỗi |  |

Bảng 14: Use case đăng xuất

Use case thêm sản phẩm

|  |  |
| --- | --- |
| Tên Use case | Thêm sản phẩm |
| Mô tả | Cung cấp chức năng cho phép người quản lý thêm sản phẩm mới |
| Tác nhân chính | Người quản lý |
| Tiền điều kiện | Người quản lý đã đăng nhập vào hệ thống |
| Cách kích hoạt | Bấm nút “Thêm sản phẩm” |
| Luồng xử lý | 1. Người quản lý truy cập tab “Sản phẩm” 2. Bấm nút “Thêm sản phẩm” 3. Nhập thông tin sản phẩm. 4. Hệ thống cập nhật sản phẩm. |
| Dòng sự kiện lỗi |  |

Bảng 15: Use case thêm sản phẩm

Use case sửa sản phẩm

|  |  |
| --- | --- |
| Tên Use case | Sửa sản phẩm |
| Mô tả | Cung cấp chức năng cho phép người quản lý sửa thông tin sản phẩm |
| Tác nhân chính | Người quản lý |
| Tiền điều kiện | Người quản lý đã đăng nhập vào hệ thống |
| Cách kích hoạt | Bấm nút “Sửa” |
| Luồng xử lý | 1. Người quản lý truy cập tab “Sản phẩm”. 2. Chọn sản phẩm cần sửa thông tin. 3. Nhập thông tin cần thay đổi. 4. Hệ thống xác nhận yêu cầu. |
| Dòng sự kiện lỗi |  |

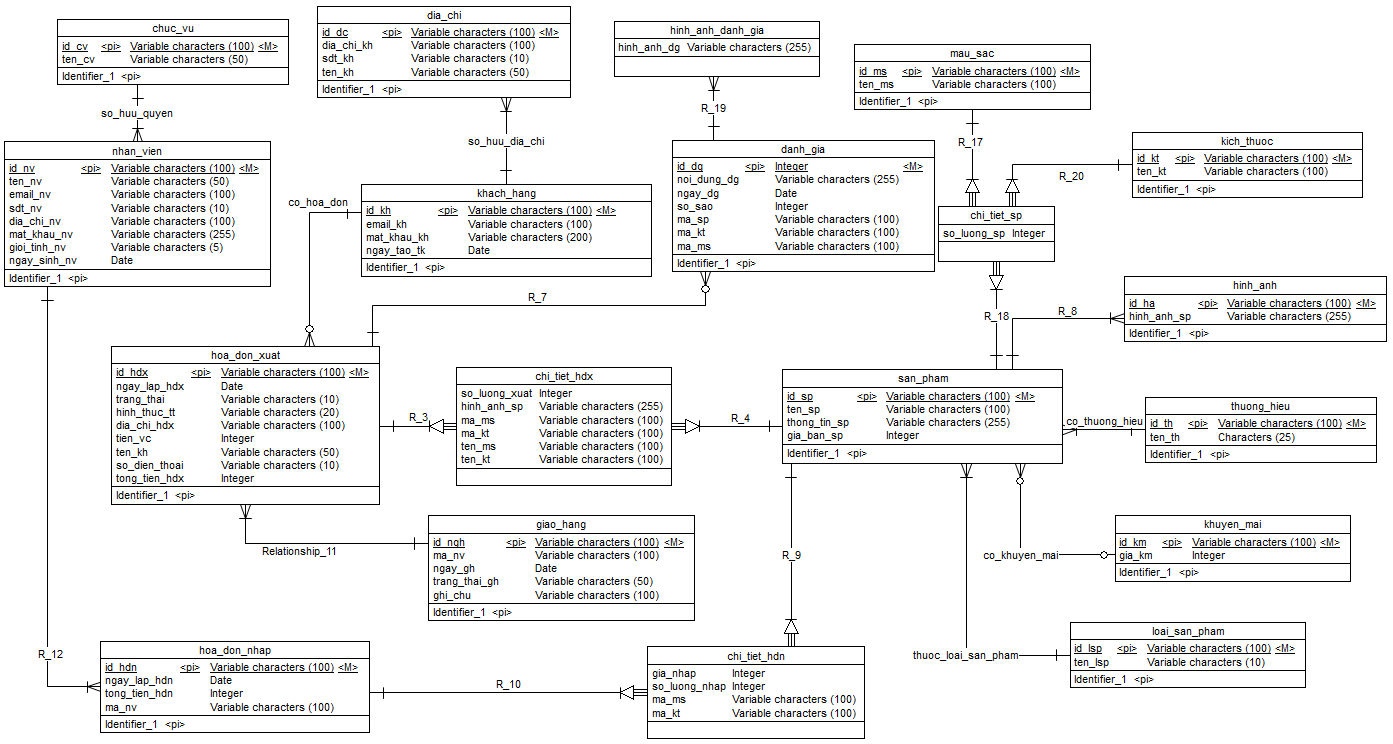
Bảng 16: Use case sửa sản phẩm

Use case xóa sản phẩm

|  |  |
| --- | --- |
| Tên Use case | Xóa sản phẩm |
| Mô tả | Cung cấp chức năng cho phép người quản lý xóa sản phẩm |
| Tác nhân chính | Người quản lý |
| Tiền điều kiện | Người quản lý đã đăng nhập vào hệ thống |
| Cách kích hoạt | Bấm nút “Xóa” |
| Luồng xử lý | 1. Người quản lý truy cập tab “Sản phẩm”. 2. Chọn sản phẩm cần xóa. 3. Hệ thống xác nhận yêu cầu. |
| Dòng sự kiện lỗi |  |

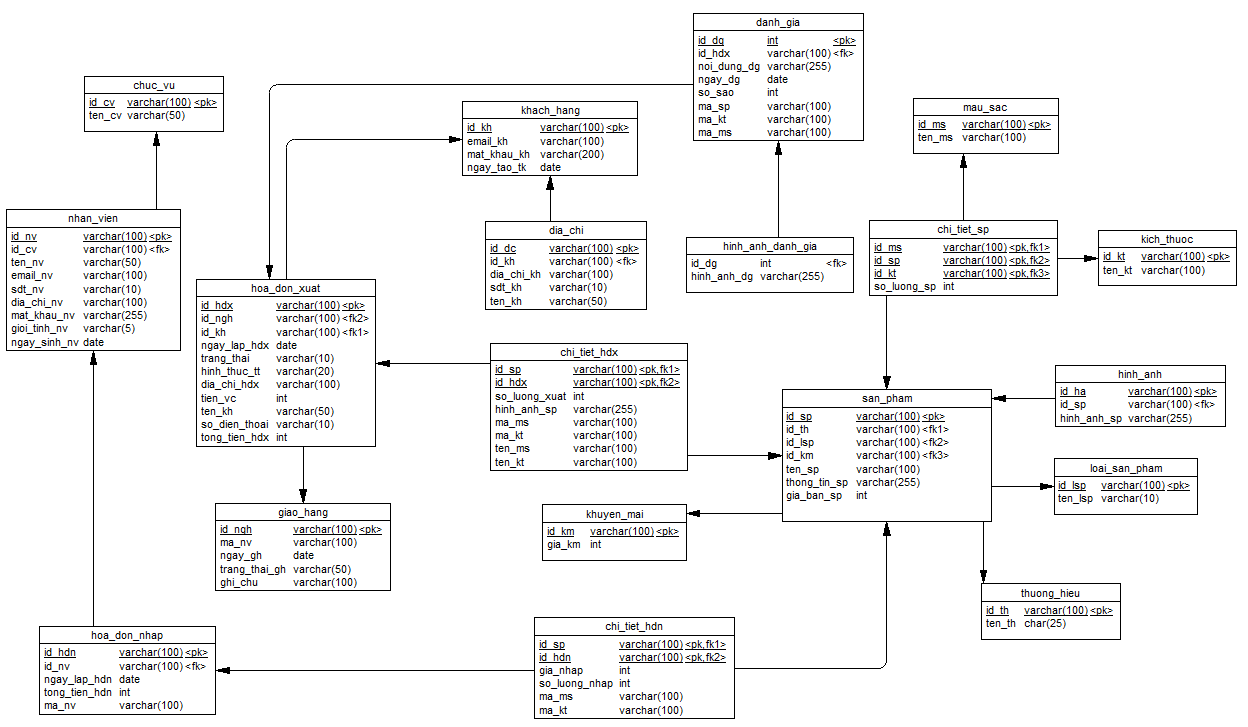
Bảng 17: Use case xóa sản phẩm

## 3.5. Mô hình dữ liệu mức quan niệm (CDM)



Hình 16: Mô hình dữ liệu mức quan niệm (CDM)

## 3.6. Mô hình dữ liệu mức vật lý (PDM)



Hình 17: Mô hình dữ liệu mức vật lý (PDM)

## 3.7. Các bảng thực thể

Danh sách các bảng

|  |  |  |
| --- | --- | --- |
| STT | Tên bảng | Diễn dải |
| 1 | khach\_hang | Khách hàng |
| 2 | dia\_chi | Đia chỉ |
| 3 | san\_pham | Sản phẩm |
| 4 | chi\_tiet\_sp | Chi tiết sản phẩm |
| 5 | mau\_sac | Màu sắc |
| 6 | kich\_thuoc | Kích thước |
| 7 | loai\_san\_pham | Loại sản phẩm |
| 8 | hinh\_anh | Hình ảnh |
| 9 | thuong\_hieu | Thương hiệu |
| 10 | khuyen\_mai | Khuyến mãi |
| 11 | danh\_gia | Đánh giá |
| 12 | hinh\_anh\_danh\_gia | Hình ảnh đánh giá |
| 13 | hoa\_don\_xuat | Hóa đơn xuất |
| 14 | chi\_tiet\_hdx | Chi tiết hóa đơn xuất |
| 15 | giao\_hang | Giao hàng |
| 16 | nhan\_vien | Nhân viên |
| 17 | chuc\_vu | Chức vụ |
| 18 | hoa\_don\_nhap | Hóa đơn nhập |
| 19 | chi\_tiet\_hdn | Chi tiết hóa đơn nhận |

Bảng 18: Các bảng thực thể

Bảng khách hàng: dùng lưu dữ liệu khách hàng, mỗi khách hàng có 1 tài khoản.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| khach\_hang | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_kh | varchar(100) | Khóa chính | ID khách hàng |
| 2 | email\_kh | varchar(100) |  | Email khách hàng |
| 3 | mat\_khau\_kh | varchar(255) |  | Mật khẩu khách hàng |
| 4 | ngay\_tao\_tk | date |  | Ngày tạo tài khoản |

Bảng 19: Bảng khách hàng

Bảng địa chỉ: dùng lưu dữ liệu địa chỉ, mỗi khách hàng có thể có nhiều địa chỉ.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| dia\_chi | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_dc | varchar(100) | Khóa chính | ID địa chỉ |
| 2 | dia\_chi\_kh | varchar(100) |  | Địa chỉ khách hàng |
| 3 | sdt\_kh | varchar(10) |  | Số điện thoại khách hàng |
| 4 | ten\_kh | varchar(20) |  | Tên khách hàng |

Bảng 20: Bảng địa chỉ

Bảng sản phẩm: dùng lưu dữ liệu sản phẩm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| san\_pham | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_sp | varchar(100) | Khóa chính | ID sản phẩm |
| 2 | ten\_sp | varchar(100) |  | Tên sản phẩm |
| 3 | thong\_tin\_sp | varchar(255) |  | Thông tin sản phẩm |
| 4 | gia\_ban\_sp | integer |  | Giá bán sản phẩm |

Bảng 21: Bảng sản phẩm

Bảng chi tiết sản phẩm: dùng để lưu dữ liệu chi tiết sản phẩm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| chi\_tiet\_sp | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | so\_luong | integer |  | Số lượng sản phẩm |

Bảng 22: Bảng chi tiết sản phẩm

Bảng màu sắc: dùng để lưu dữ liệu màu sắc.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| mau\_sac | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_ms | varchar(100) | Khóa chính | ID màu sắc |
| 2 | ten\_ms | varchar(100) |  | Tên màu sắc |

Bảng 23: Bảng màu sắc

Bảng kích thước: dùng để lưu dữ liệu kích thước.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| kich\_thuoc | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_kt | varchar(100) | Khóa chính | ID kích thước |
| 2 | ten\_kt | varchar(5) |  | Tên kích thước |

Bảng 24: Bảng kích thước

Bảng hình ảnh: dùng để lưu dữ liệu hình ảnh.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| hinh\_anh | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_ha | varchar(100) | Khóa chính | ID hình ảnh |
| 2 | hinh\_anh\_sp | varchar(255) |  | hình ảnh sản phẩm |

Bảng 25: Bảng hình ảnh

Bảng loại sản phẩm: dùng để lưu dữ liệu loại sản phẩm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| loai\_san\_pham | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_lsp | varchar(100) | Khóa chính | ID loại sản phẩm |
| 2 | ten\_lsp | varchar(10) |  | Tên loại sản phẩm |

Bảng 26: Bảng loại sản phẩm

Bảng thương hiệu: dùng để lưu dữ liệu thương hiệu.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| thuong\_hieu | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_th | varchar(100) | Khóa chính | ID thương hiệu |
| 2 | ten\_th | varchar(25) |  | Tên thương hiệu |

Bảng 27: Bảng thương hiệu

Bảng khuyến mãi: dùng để lưu dữ liệu khuyến mãi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| khuyen\_mai | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_km | varchar(100) | Khóa chính | ID khuyến mãi |
| 2 | gia\_km | integer |  | Giá khuyến mãi |

Bảng 28: Bảng khuyến mãi

Bảng đánh giá: dùng để lưu dữ liệu đánh giá.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| danh\_gia | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_dg | integer | Khóa chính | ID đánh giá |
| 2 | so\_sao | integer |  | Số sao |
| 3 | noi\_dung\_dg | varchar(255) |  | Nội dung đánh giá |
| 4 | ngay\_dg | date |  | Ngày đánh giá |
| 5 | ma\_sp | varchar(100) |  | Mã sản phẩm |
| 6 | ma\_ms | varchar(100) |  | Mã màu sắc |
| 7 | ma\_kt | varchar(100) |  | Mã kích thước |

Bảng 29: Bảng đánh giá

Bảng hình ảnh đánh giá: dùng lưu dữ liệu hình ảnh đánh giá.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| hinh\_anh\_danh\_gia | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | hinh\_anh\_dg | varchar(255) | Khóa chính | Hình ảnh đánh giá |

Bảng 30: Bảng hình ảnh đánh giá

Bảng hóa đơn xuất: dùng lưu dữ liệu hóa đơn xuất.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| hoa\_don\_xuat | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_hdx | varchar(100) | Khóa chính | ID hóa đơn xuất |
| 2 | ten\_kh | varchar(50) |  | Tên khách hàng |
| 3 | so\_dien\_thoai | varchar(10) |  | Số điện thoại |
| 4 | tong\_tien\_hdx | integer |  | Tổng tiền hóa đơn xuất |
| 5 | ten\_vc | integer |  | Tiền vận chuyển |
| 6 | ngay\_lap\_hdx | date |  | Ngày lập hóa đơn xuất |
| 7 | trang\_thai | varchar(10) |  | Trạng thái |
| 8 | hinh\_thuc\_tt | varchar(20) |  | Hình thức thanh toán |
| 9 | dia\_chi\_hdx | varchar(100) |  | Địa chỉ hóa đơn xuất |

Bảng 31: Bảng hóa đơn xuất

Bảng chi tiết hóa đơn xuất: dùng để lưu dữ liệu chi tiết hóa đơn xuất.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| chi\_tiet\_hdx | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | so\_luong\_xuat | integer | Khóa chính | Số lượng xuất |
| 2 | hinh\_anh\_sp | varchar(255) |  | Hình ảnh sản phẩm |
| 3 | ma\_ms | varchar(100) |  | Mã màu sắc |
| 4 | ma\_kt | varchar(100) |  | Mã kích thước |
| 5 | ten\_ms | varchar(100) |  | Tên màu sắc |
| 6 | ten\_kt | varchar(100) |  | Tên kích thước |

Bảng 32: Bảng chi tiết hóa đơn xuất

Bảng giao hàng: dùng lưu dữ liệu giao hàng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| giao\_hang | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_ngh | varchar(100) | Khóa chính | ID người giao hàng |
| 2 | ma\_nv | varchar(100) |  | Mã nhân viên |
| 3 | ngay\_gh | date |  | Ngày giao hàng |
| 4 | trang\_thai\_gh | varchar(50) |  | Trạng thái giao hàng |
| 5 | ghi\_chu | varchar(100) |  | Ghi chú |

Bảng 33: Bảng giao hàng

Bảng hóa đơn nhập: dùng lưu dữ liệu hóa đơn nhập.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| hoa\_don\_nhap | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_hdn | varchar(100) | Khóa chính | ID hóa đơn nhập |
| 2 | ngay\_lap\_hdn | date |  | Ngày lập hóa đơn nhập |
| 3 | tong\_tien\_hdn | integer |  | Tổng tiền hóa đơn nhập |
| 4 | ma\_nv | varchar(100) |  | Mã nhân viên |

Bảng 34: Bảng hóa đơn nhập

Bảng chi tiết hóa đơn nhập: dùng lưu dữ liệu chi tiết hóa đơn nhập.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| chi\_tiet\_hdn | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | gia\_nhap | integer |  | Giá nhập |
| 2 | so\_luong\_nhap | integer |  | Số lượng nhập |
| 3 | ma\_ms | varchar(100) |  | Mã màu sắc |
| 4 | ma\_kt | varchar(100) |  | Mã kích thước |

Bảng 35: Bảng chi tiết hóa đơn nhập

Bảng nhân viên: dùng lưu dữ liệu nhân viên, Mỗi nhân viên chỉ có 1 tài khoản.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nhan\_vien | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_nv | varchar(100) | Khóa chính | ID nhân viên |
| 2 | ten\_nv | varchar(50) |  | Tên nhân viên |
| 3 | email\_nv | varchar(100) |  | Email nhân viên |
| 4 | sdt\_nv | varchar(10) |  | Số điện thoại nhân viên |
| 5 | dia\_chi\_nv | varchar(100) |  | Địa chỉ nhân viên |
| 6 | mat\_khau\_nv | varchar(255) |  | Mật khẩu nhân viên |
| 7 | gioi\_tinh\_nv | varchar(5) |  | Giới tính nhân viên |
| 8 | ngay\_sinh\_nv | date |  | Ngày sinh nhân viên |

Bảng 36: Bảng nhân viên

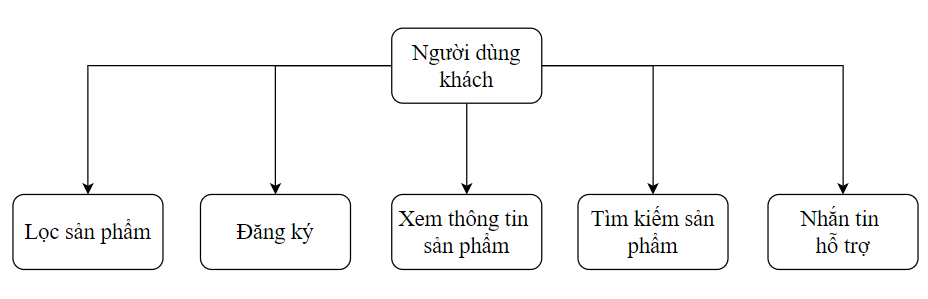
Bảng chức vụ: dùng lưu dữ liệu chức vụ, mỗi nhân viên chỉ có 1 chức vụ.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| chuc\_vu | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | id\_cv | varchar(100) | Khóa chính | ID chức vụ |
| 2 | ten\_cv | varchar(50) |  | Tên chức vụ |

Bảng 37:Bảng chức vụ

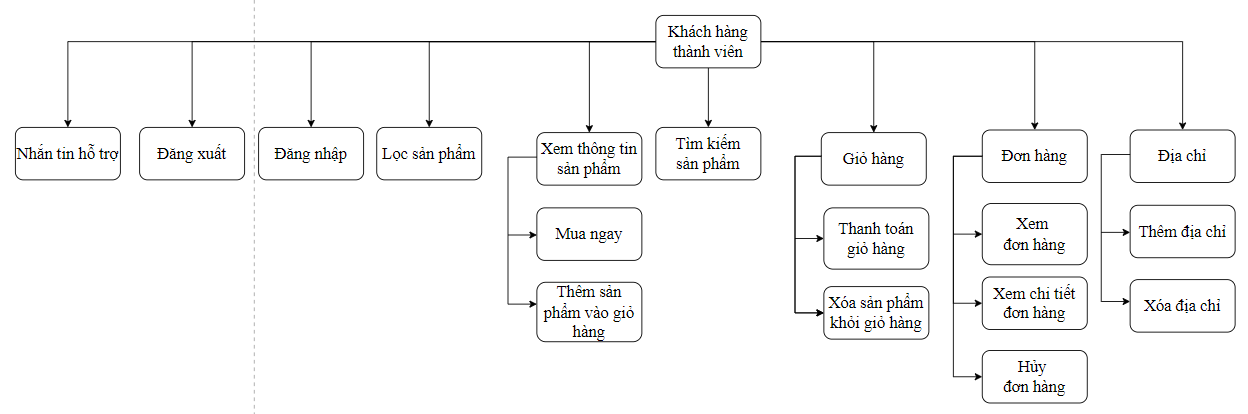
## 3.8. Sơ đồ chức năng

### 3.8.1. Sơ đồ chức năng của người dùng khách



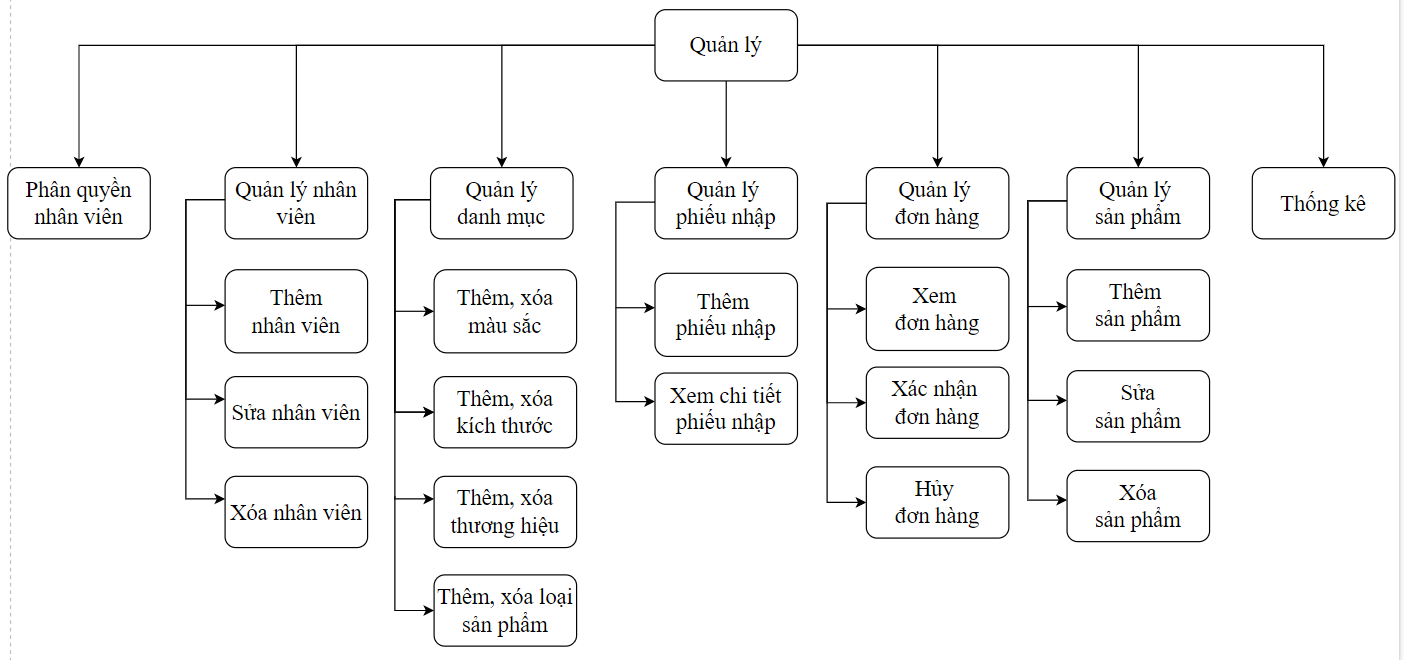
Hình 18: Sơ đồ chức năng người dùng khách

### 3.8.2. Sơ đồ chức năng của khách hàng thành viên

**

Hình 19: Sơ đồ chức năng của khách hàng thành viên

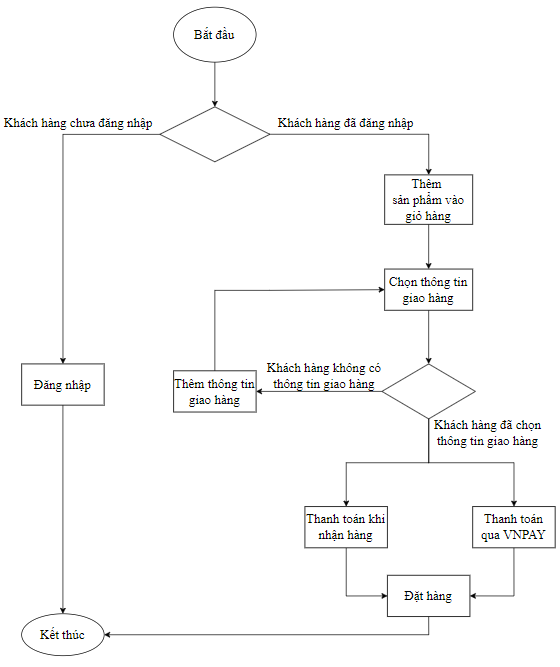
### 3.8.3. Sơ đồ chức năng của quản lý



Hình 20: Sơ đồ chức năng của quản lý

## 3.9. Quy trình đặt hàng

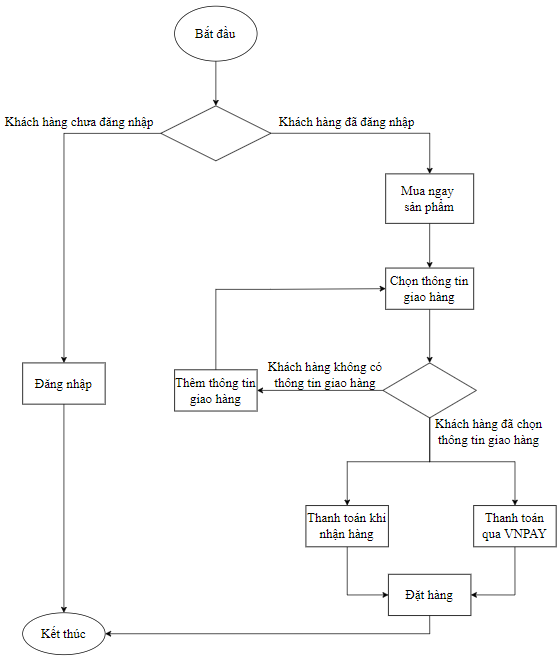
### 3.9.1. Thanh toán giỏ hàng



Hình 21: Quy trình thanh toán giỏ hàng

Để mua sản phẩm, khách VNPAY. Cuối cùng, nếu khách hàng chọn phương thức thanh toán trực tiếp khi giao hàng thì hệ thống sẽ ghi nhận đơn hàng đã tạo thành công và kết thúc quy trình mua và thanh toán giỏ hàng. Nếu khách hàng chọn phương thức thanh toán online thì khách hàng cần lựa chọn và điền đầy đủ, chính xác thông tin theo quy trình thanh toán của VNPAY và kết thúc quy trình đặt hàng. Đặt hàng cần đăng nhập vào tài khoản của mình. Sau khi đã thêm sản phẩm cần mua vào giỏ hàng, khách hàng nhấn thanh toán và chọn thông tin giao hàng của mình gồm thông tin: tên, số điện thoại và địa chỉ. Nếu thông tin giao hàng chưa có, khách hàng có thể thêm trực tiếp thông tin giao hàng và chọn lại thông tin giao hàng. Khi đã chọn thông tin giao hàng, khách hàng có thể thanh toán sản phẩm qua 2 hình thức: thanh toán khi nhận hàng và thanh toán trực tuyến qua.

### 3.9.2. Thanh toán mua ngay

*Hình 22: Quy trình thanh toán mua ngay*

Tương tự như thanh toán giỏ hàng, thanh toán mua ngay chỉ có thể thực hiện thanh toán đối với một sản phẩm, còn thanh toán giỏ hàng sẽ thực hiện thanh toán được nhiều sản phẩm cùng lúc.

### 3.9.3. Các yêu cầu ràng buộc chính

Giỏ hàng: khi thêm sản phẩm vào giỏ hàng, số lượng sản phẩm được thêm phải nhỏ hơn hoặc bằng số lượng sản phẩm tồn kho.

Sản phẩm: khi mua sản phẩm, số lượng sản phẩm được mua phải nhỏ hơn hoặc bằng số lượng sản phẩm tồn kho. Số lượng sản phẩm sẽ được cập nhật khi khách hàng hoàn tất quy trình mua hàng.

Đơn hàng: đơn hàng bị hủy sẽ cập nhật lại số lượng sản phẩm trong kho.

Đánh giá: khách hàng sẽ được đánh giá khi đơn hàng đã được giao thành công.

### 3.9.4. Ràng buộc liên bộ, liên thuộc tính, liên quan hệ

R1: Mỗi sản phẩm thuộc một loại sản phẩm

Phát biểu: sp san\_pham, lsp loai\_san\_pham (sp.id\_lsp = lsp.id\_lsp)

Bối cảnh: san\_pham, loai\_san\_pham

Tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Xóa | Sửa |
| san\_pham | + | - | +(id\_lsp) |
| loai\_san\_pham | - | + | -(\*) |

R2: Tất cả các sản phẩm đều có id khác nhau

Phát biểu: sp1, sp2 san\_pham: nếu sp1 khác sp2 thì sp1.id\_sp sp2.id\_sp

Bối cảnh: san\_pham

Tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R2 | Thêm | Xóa | Sửa |
| san\_pham | + | - | -(\*) |

R3: Mỗi hình ảnh thuộc 1 sản phẩm

Phát biểu: ha ∈ hinh\_anh, sp ∈ san\_pham (ha.id\_sp = sp.id\_sp)

Bối cảnh: san\_pham, hinh\_anh

Tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R3 | Thêm | Xóa | Sửa |
| san\_pham | + | - | +(id\_sp) |
| hinh\_anh | - | + | -(\*) |

R4: Số lượng sản phẩm trong chi tiết đặt hàng phải bằng số lượng tồn kho

Phát biểu:

ct\_hdx ∈ chi\_tiet\_hdx, hdx ∈ hoa\_don\_xuat, sp ∈ san\_pham,

ct\_sp ∈ chi\_tiet\_sp:

nếu (hdx.ttdhID = 1) (hdx.hdxID = ct\_hdx.hdxID) (ct\_hdx.spID = sp.spID )

thì ct\_hdx.so\_luong\_xuat = ct\_sp.so\_luong\_sp

Bối cảnh: san\_pham, chi\_tiet\_sp, hoa\_don\_xuat, chi\_tiet\_hdx

Tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R4 | Thêm | Xóa | Sửa |
| san\_pham | - | + | +(so\_luong\_sp) |
| chi\_tiet\_sp | - | + | +(id\_sp) |
| hoa\_don\_xuat | - | - | +(id\_hdx) |
| chi\_tiet\_hdx | - | + | +(so\_luong\_xuat) |

## 3.10. Giao diện khách hàng

### 3.10.1. Giao diện chính

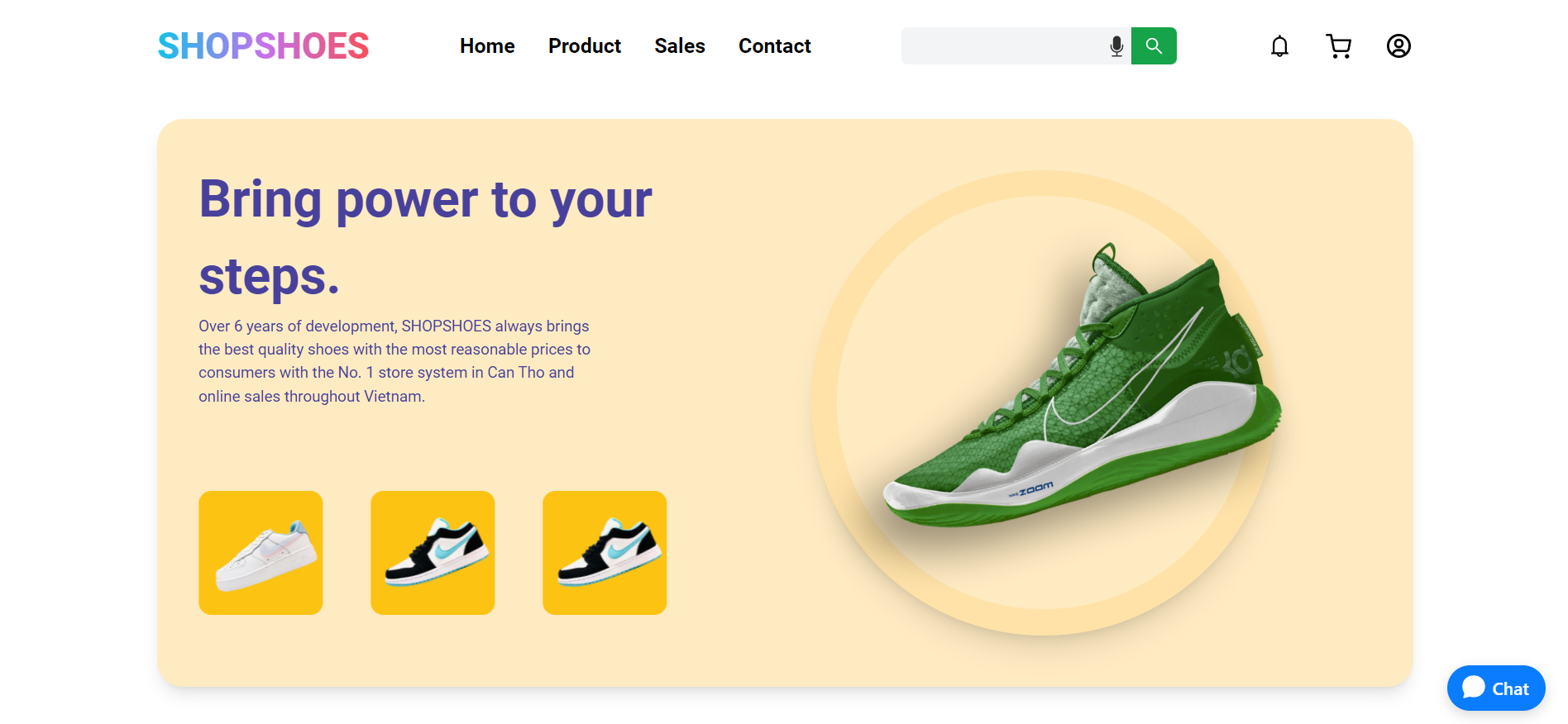
Giao diện hợp lý giúp tạo điểm nhấn, tạo hình ảnh và truyền tải nội dung đến người xem. Chính vì thế, một giao diện phù hợp với một website sẽ làm website có giá trị hơn và được đánh giá cao và tạo sự thích thú trong mắt người xem. Giao diện chính có chức năng đăng ký, đăng nhập, tìm kiếm, …Nơi liệt kê những sản phẩm mới nhất và sản phẩm giảm giá, … Khách hàng có thể nhắn tin cho cửa hàng ngay trên website để được hỗ trợ sớm nhất.

* Giao diện khách hàng khi chưa đăng nhập



Hình 23: Giao diện khách hàng khi chưa đăng nhập

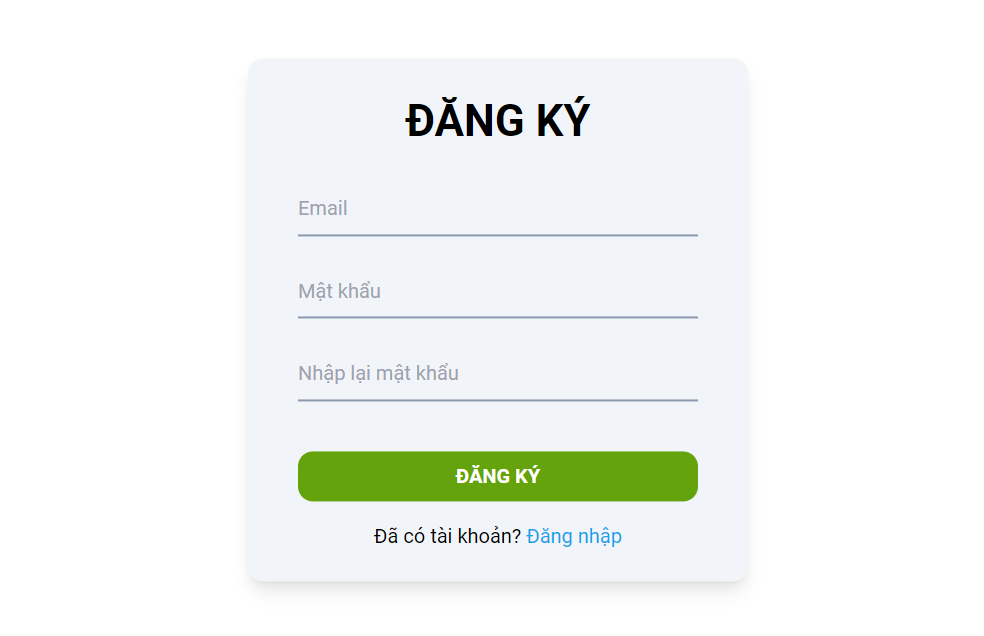
* Giao diện khách hàng đã đăng nhập



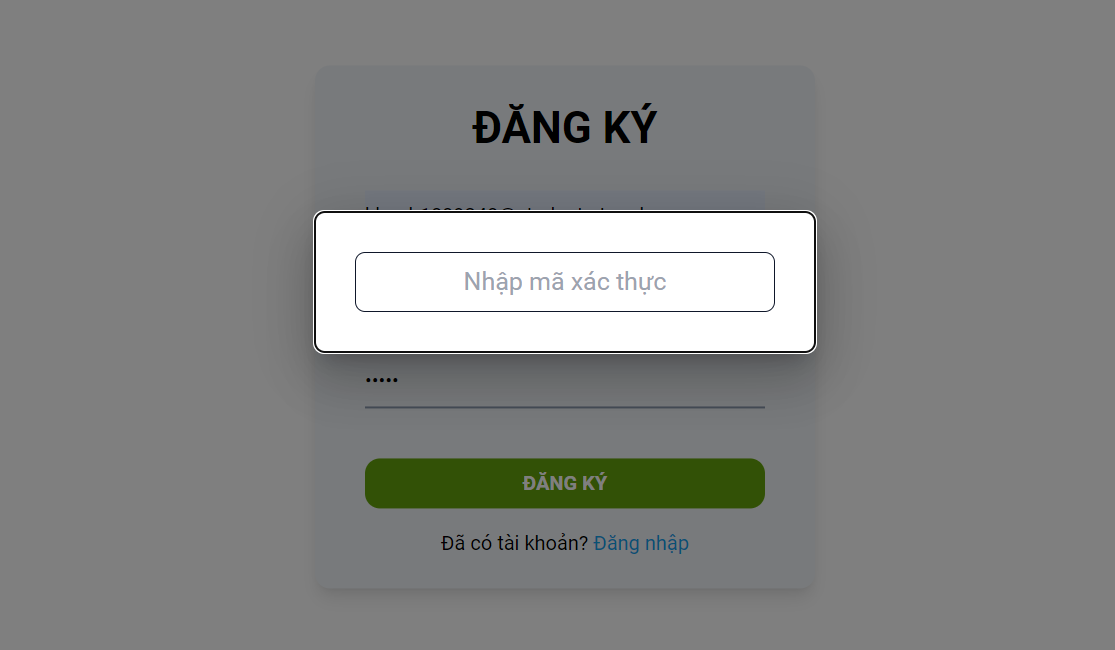
Hình 24: Giao diện khách hàng đã đăng nhập

### 3.10.2. Giao diện đăng ký, đăng nhập và quên mật khẩu

Chức năng đăng ký cho phép khách hàng tạo một tài khoản thành viên, khách hàng sẽ cần cung cấp một email và mật khẩu. Sau đó, một email chứa mã xác thực gồm 6 số sẽ được gửi đến email đã đăng ký của khách hàng, khách hàng sẽ dùng 6 số đó để nhập vào ô xác thực để hoàn tất quá trình đăng ký tài khoản. Việc hệ thống gửi mã xác thực để xác nhận email đó là của người dùng thực, tránh tình trạng quá tải hệ thống.

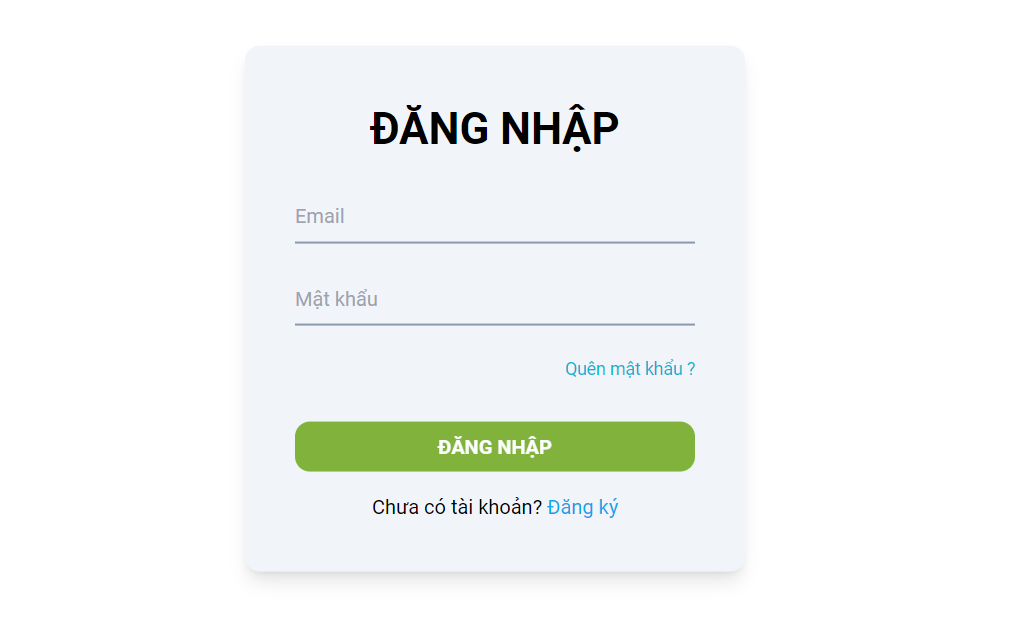


Hình 25: Giao diện đăng ký



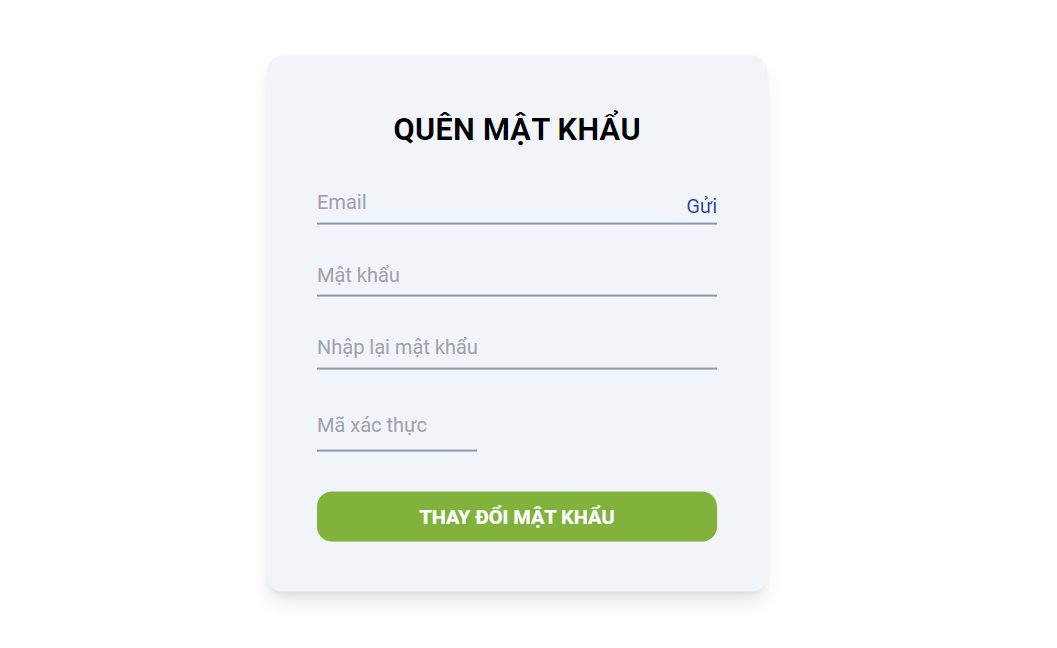
Hình 26: Giao diện nhập mã xác thực

Chức năng đăng nhập cho phép khách hàng truy cập tài khoản mua hàng của mình bằng việc nhập đúng email và mật khẩu đã đăng ký trước đó.



Hình 27: Giao diện đăng nhập

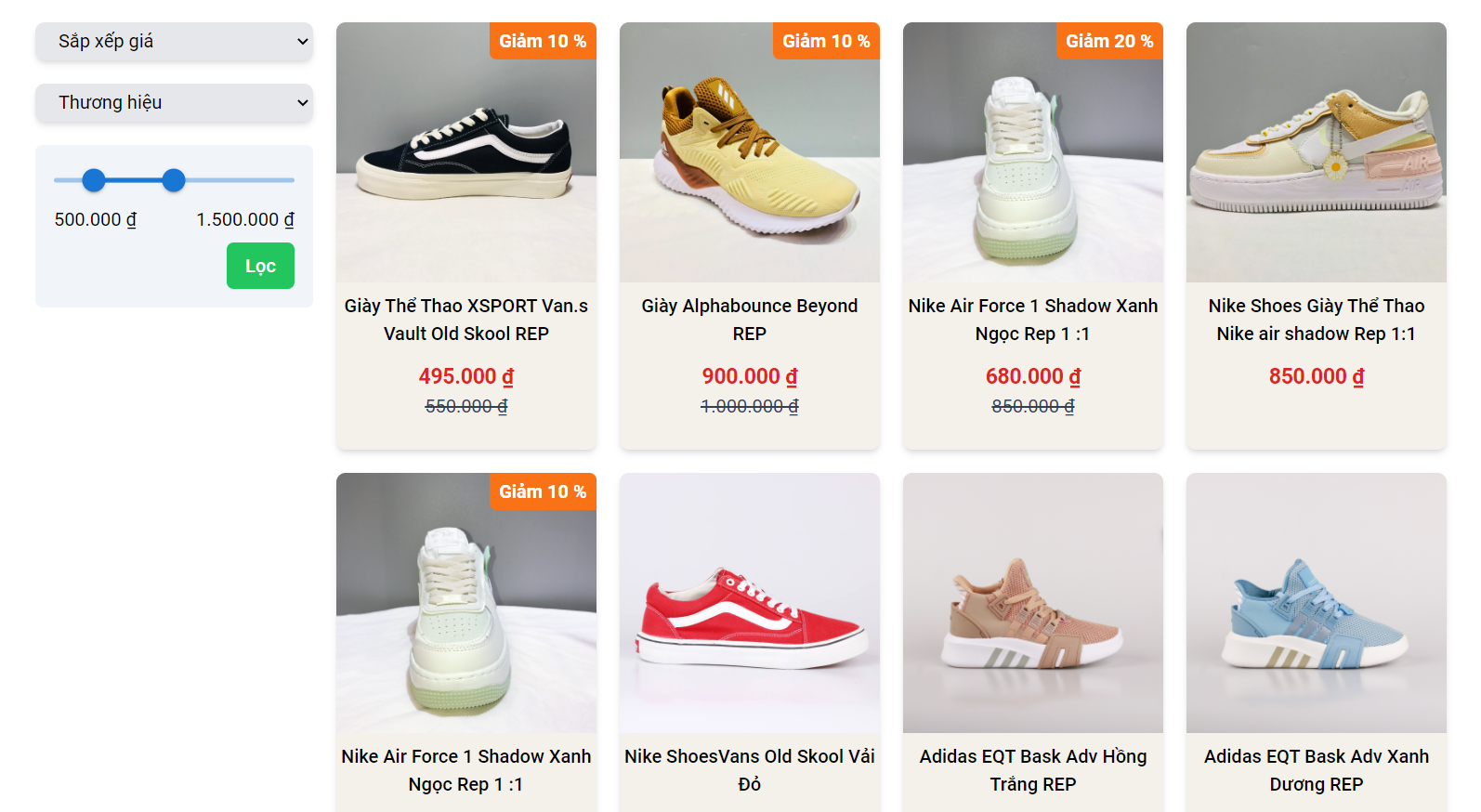
Hệ thống cũng cho phép lấy lại mật khẩu khi khách hàng quên bằng việc cung cấp email cần lấy lại mật khẩu và mật khẩu mới. Hệ thống sẽ gửi mã xác thực gồm 6 số đến email đó và khách hàng cần nhập đúng 6 số xác thực vào trường để hoàn tất quá trình lấy lại mật khẩu.



Hình 28: Giao diện lấy lại mật khẩu

### 3.10.3. Giao diện sản phẩm

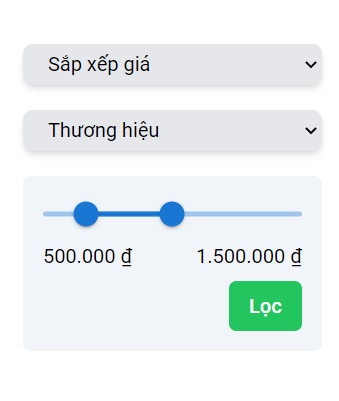
Khách hàng có thể xem tất cả sản phẩm thông qua danh mục sản phẩm, giao diện sản phẩm sẽ liệt kê những sản phẩm gồm: tên sản phẩm, giá của sản phẩm, ... của cửa hàng.



Hình 29: Giao diện trang sản phẩm

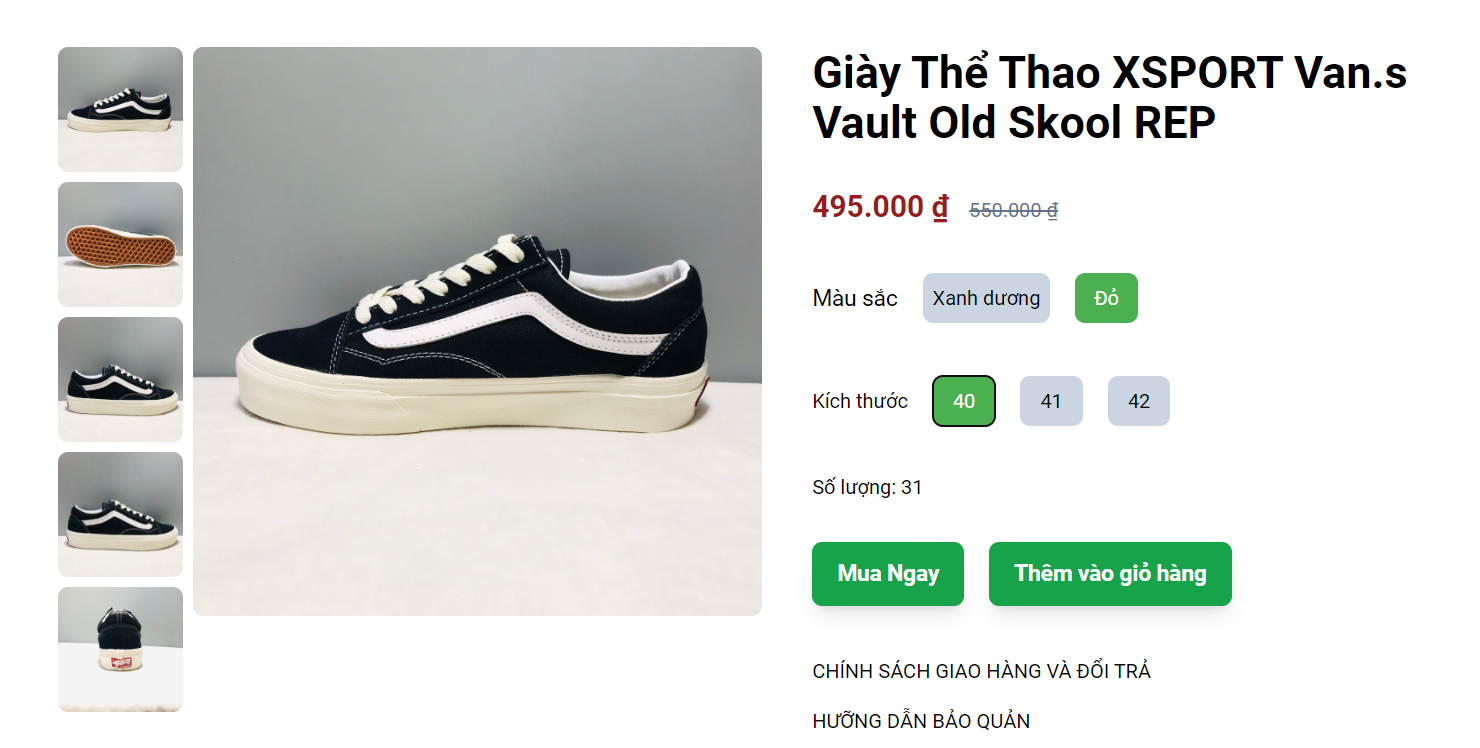
Khách hàng cũng có thể xem chi tiết sản phẩm bằng việc chọn sản phẩm cần xem, giao diện chi tiết sản phẩm sẽ hiển thị tên sản phẩm, giá sản phẩm, màu sắc, kích thước và số lượng còn lại của sản phẩm. Ngoài ra, còn hiển thị mô tả về sản phẩm và các đánh giá của những người đã mua sản phẩm đó trước.

Giao diện sản phẩm cũng hiển thị bộ lọc sản phẩm, giúp khách hàng tìm kiếm nhanh sản phẩm theo yêu cầu của mình.



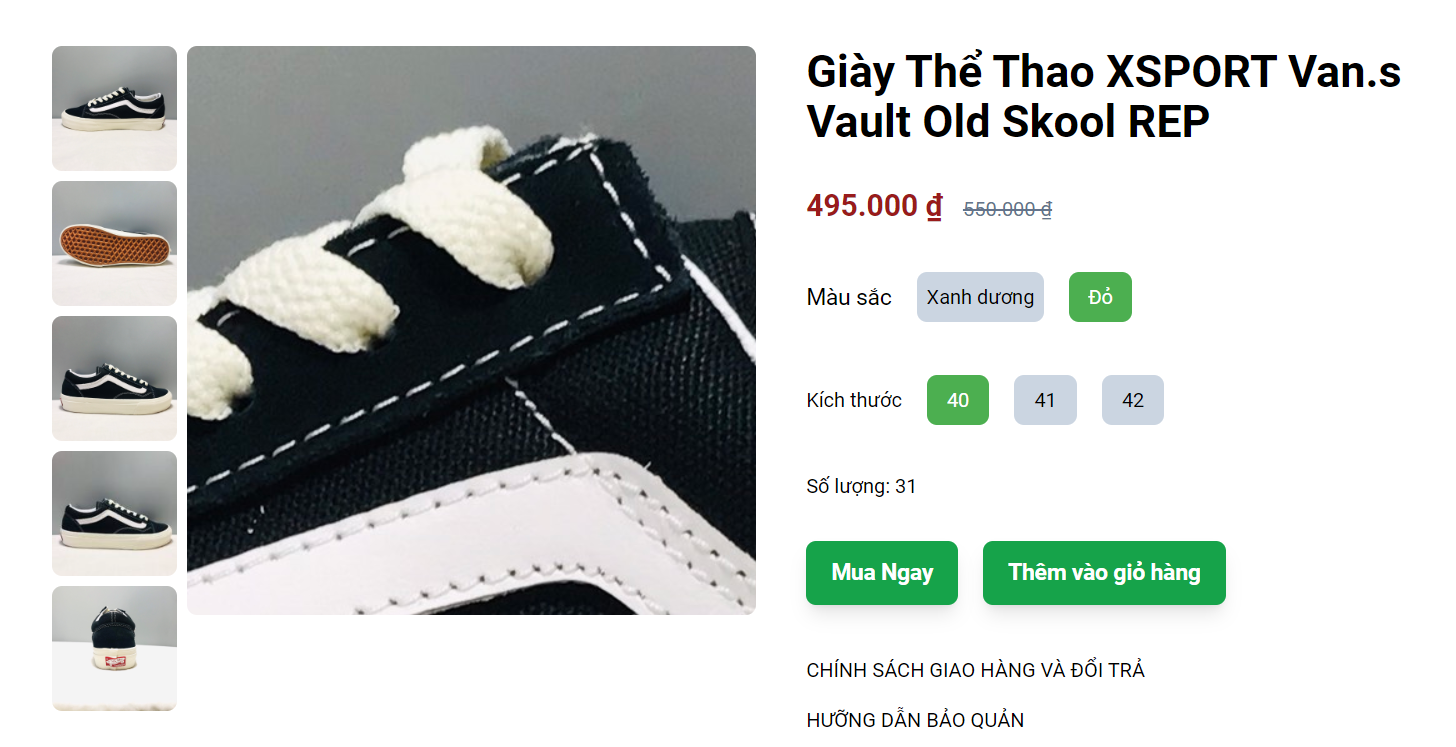
Hình 30: Chức năng lọc sản phẩm

### 3.10.4. Xem thông tin chi tiết sản phẩm



Hình 31: Giao diện thông tin chi tiết sản phẩm

Khách hàng có tài khoản và khách hàng không có tài khoản đều có thể xem thông tin chi tiết của một sản phẩm nào đó bằng việc chọn vào một sản phẩm, giao diện chi tiết sản phẩm đó sẽ hiện ra bao gồm các thông tin như: hình ảnh, tên sản phẩm, giá của sản phẩm, giá khuyến mãi (nếu có), màu sắc, kích thước và số lượng sản phẩm còn lại trong kho. Ngoài ra còn có thông tin mô tả về sản phẩm và đánh giá của sản phẩm, để quan sát sản phẩm, khách hàng có thể phóng to sản sản phẩm bằng cách rê chuột vào hình ảnh sản phẩm. Tại đây, khi muốn thêm sản phẩm vào giỏ hàng hoặc mua ngay sản phẩm, khách hàng cần đăng nhập vào tài khoản của mình.



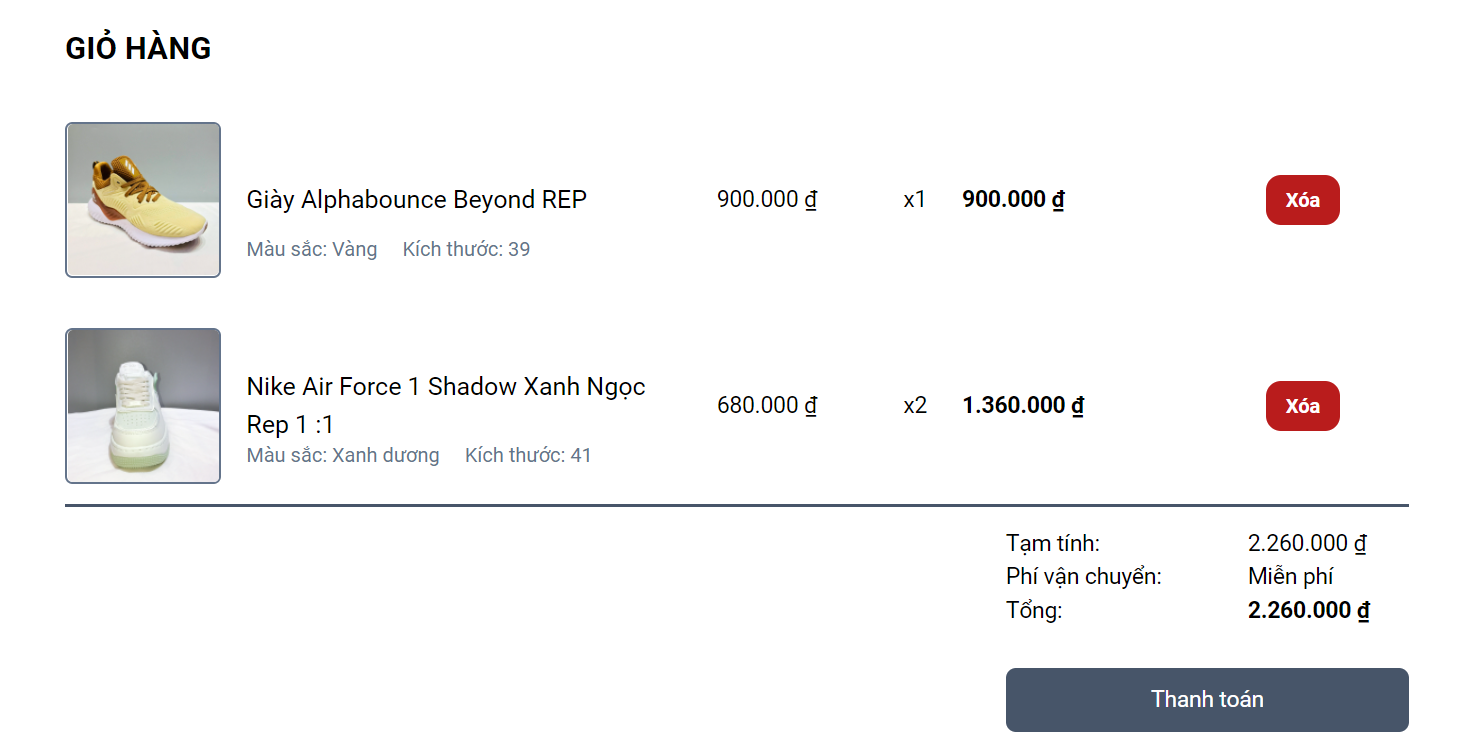
Hình 32: Chức năng phóng to sản phẩm



Hình 33: Giao diện đánh giá

### 3.10.5. Giỏ hàng

Khách hàng có thể thêm nhiều sản phẩm vào giỏ hàng và thanh toán tất cả sản phẩm trong giỏ hàng. Khách hàng cũng có thể xóa sản phẩm không ứng ý nữa khỏi giỏ hàng. Để thêm sản phẩm vào giỏ hàng, khách hàng cần lựa chọn màu sắc và kích thước của sản phẩm và khách hàng cần đăng nhập tài khoản của mình.

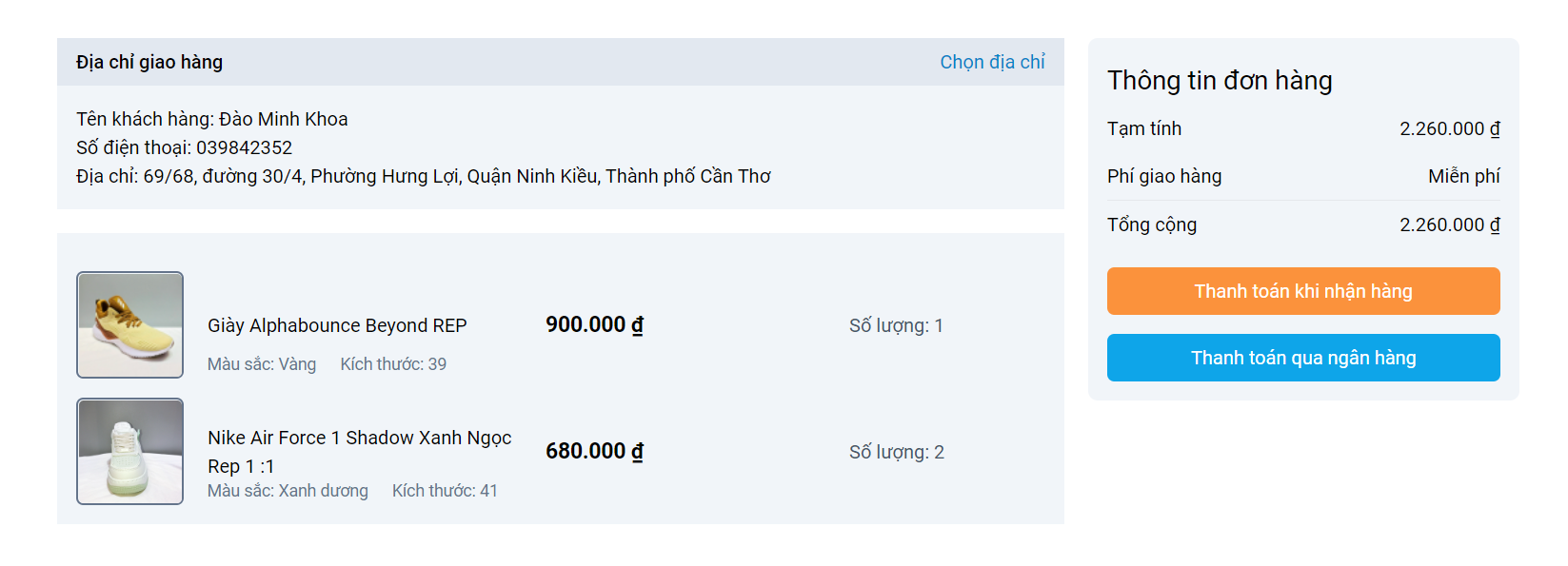


Hình 34: Giao diện giỏ hàng

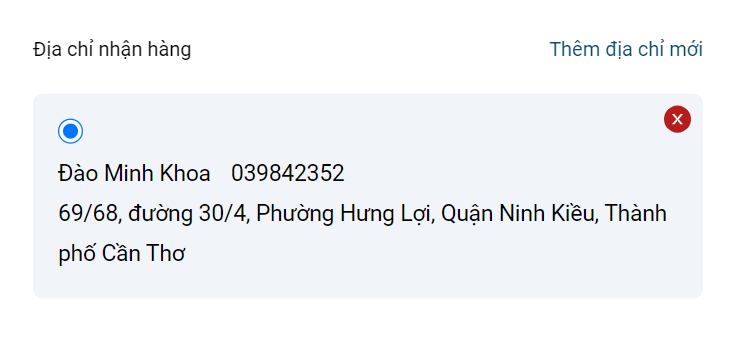
### 3.10.6. Thanh toán sản phẩm

Giao diện thanh toán sẽ hiển thị những sản phẩm được thanh toán gồm: tên sản phẩm, kích thước, màu sắc, số lượng sản phẩm, phí vận chuyển và tổng số tiền cần thanh toán.

Khách hàng có thể thực hiện thanh toán trực tuyến qua VNPAY hoặc thanh toán khi nhận hàng. Để thanh toán được sản phẩm, khách hàng cần chọn thông tin giao hàng, nếu chưa có thông tin giao hàng, khách hàng có thể thêm thông tin giao hàng mới.



Hình 35: Giao diện thanh toán sản phẩm



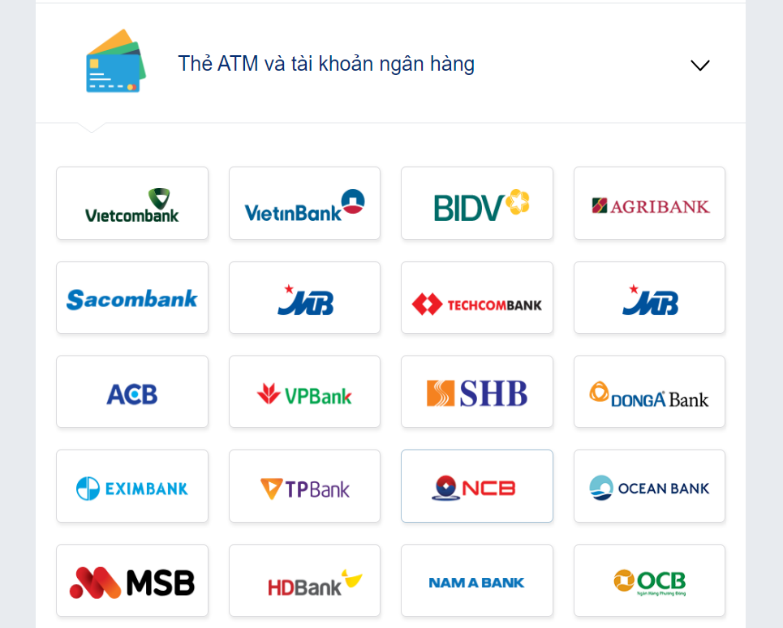
Hình 36: Chọn thông tin nhận hàng

Thanh toán trực tuyến qua VNPAY

Sau khi chọn thanh toán qua ngân hàng, website sẽ điều hướng khách hàng đến website của VNPAY để thực hiện thanh toán, khách hàng cần chọn phương thức thanh toán sau đó chọn ngân hàng mình sử dụng và điền chính xác thông tin để hoàn tất việc thanh toán.



Hình 37: Giao diện chọn phương thức thanh toán



Hình 38: Giao diện chọn ngân hàng thanh toán



Hình 39: Giao diện nhập thông tin ngân hàng thanh toán

### 3.10.7. Lịch sử mua hàng

Giao diện lịch sử mua hàng sẽ hiển thị danh sách sản phẩm theo từng tiến trình của đơn hàng.

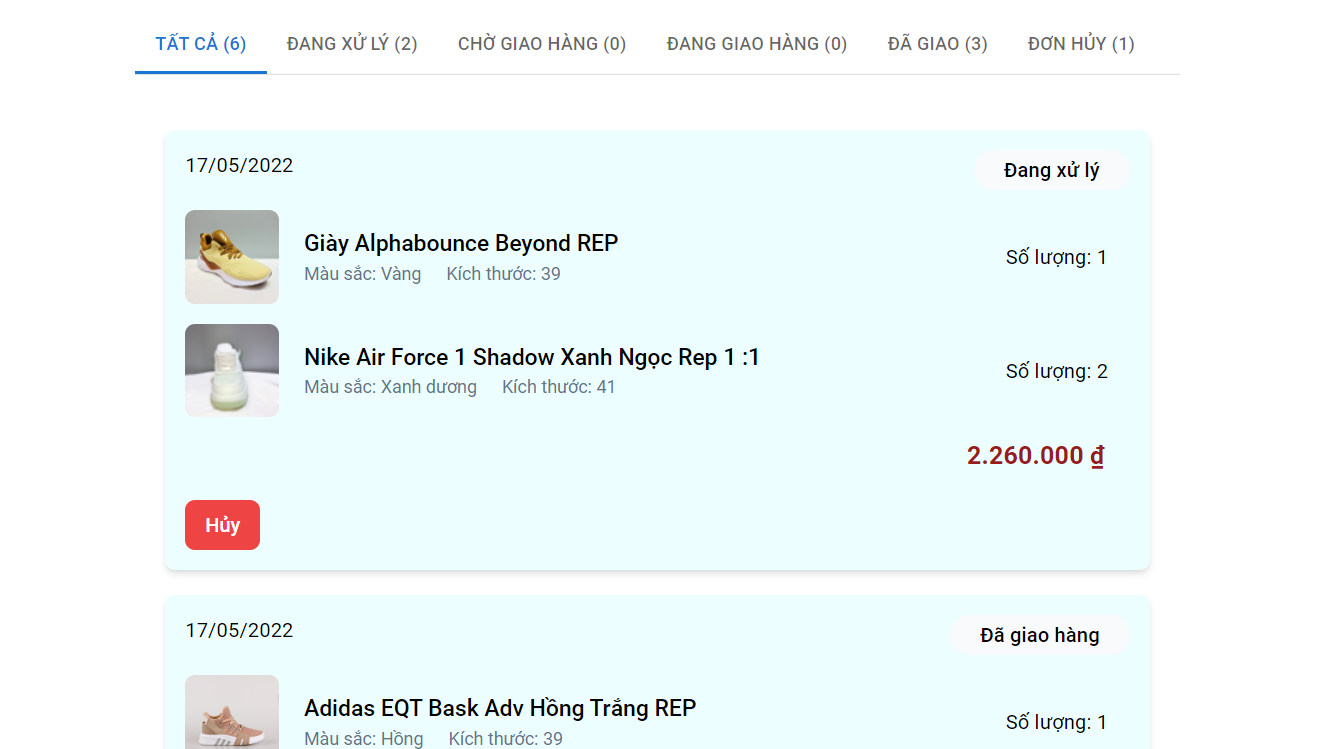
Khi đơn hàng đang đợi quản lý xác nhận thì đơn hàng đó sẽ ở mục đang xử lý

Khi quản lý xác nhận đơn hàng đó, đơn hàng sẽ được chuyển qua mục đang giao hàng

Khi người giao hàng chấp nhận đơn hàng đó, đơn hàng sẽ được chuyển qua mục đang giao hàng.

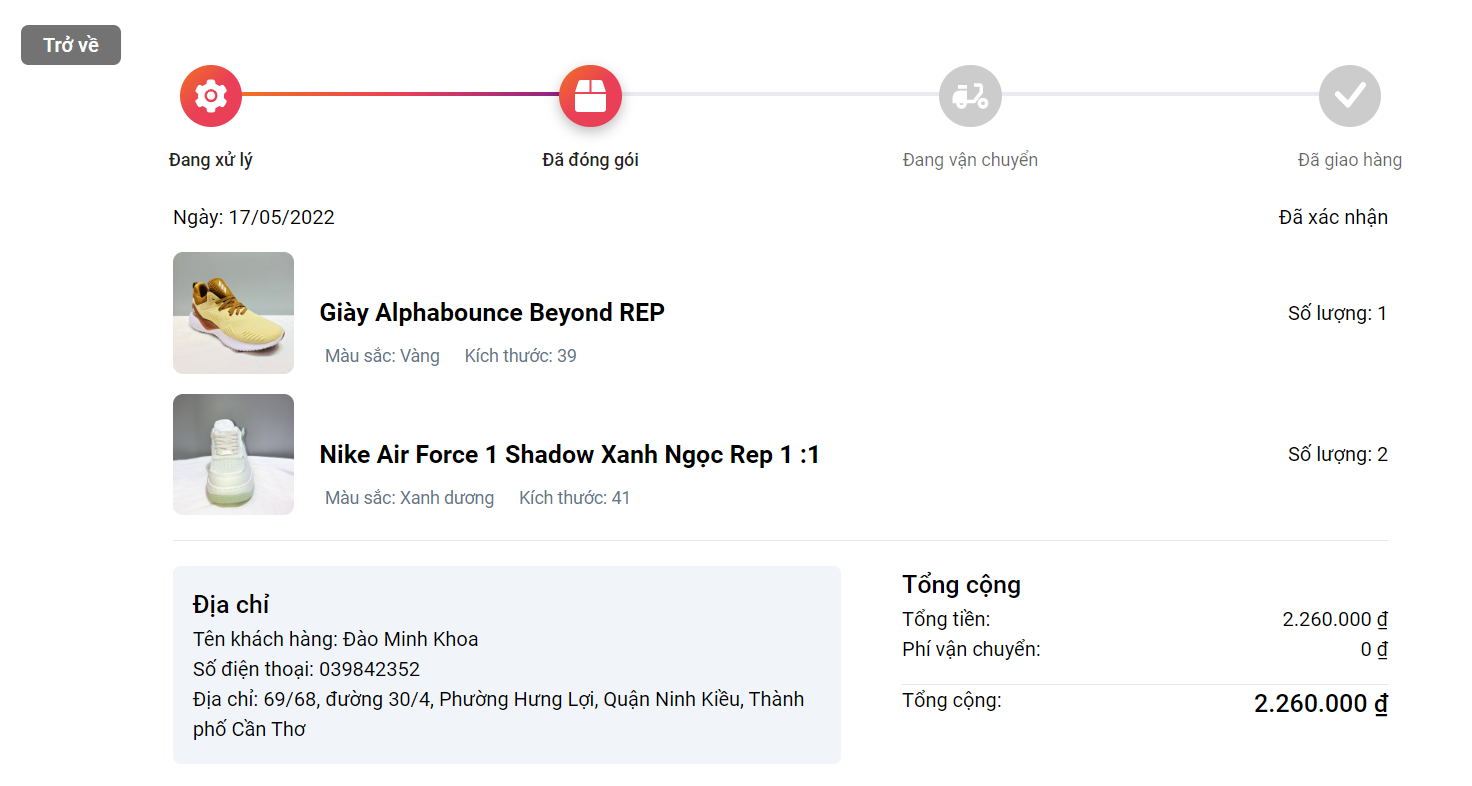
Khi khách hàng giao hàng thành công, đơn hàng sẽ được chuyển qua mục đã giao.

Ngoài ra, khách hàng hủy đơn khi quản lý chưa xác nhận, đơn hàng sẽ chuyển qua mục đơn hủy



Hình 40: Giao diện danh sách đơn hàng

Để xem chi tiết của một đơn hàng, khách hàng có thể chọn đơn hàng cần xem. Giao diện chi tiết đơn hàng sẽ hiển thị thanh trạng thái của đơn hàng, sản phẩm của đơn hàng, tổng tiền đơn hàng và thông tin giao hàng.

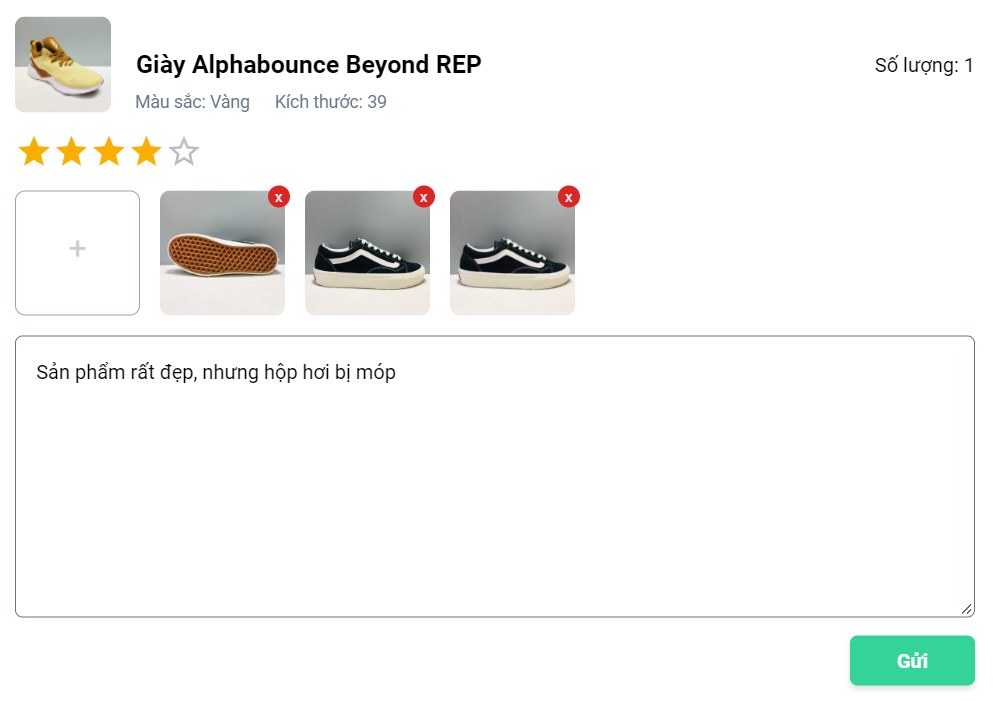


Hình 41: Giao diện chi tiết hóa đơn

### 3.10.8. Đánh giá sản phẩm

Để tiếp nhận thông tin phản hồi từ khách hàng, cửa hàng cần những đánh giá của khách hàng để cải thiện sản phẩm cũng như quy trình bán hàng, và vận chuyển.

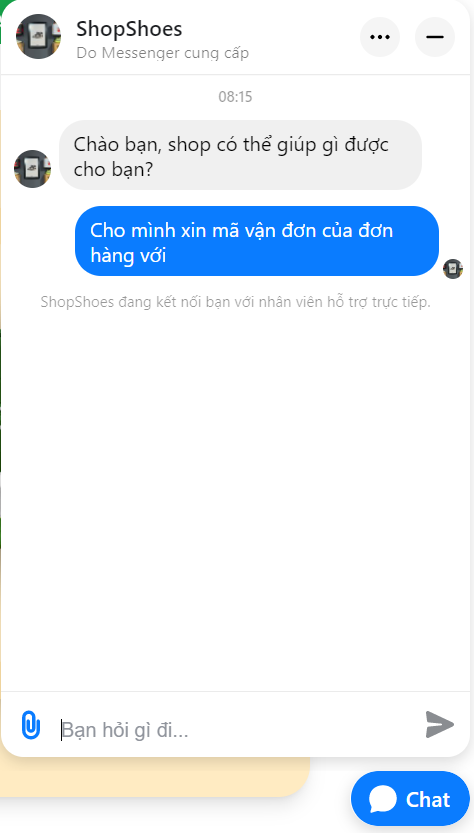
Khách hàng sẽ được đánh giá sản phẩm khi đã nhận hàng thành công. Khách hàng sẽ chọn mức độ hài lòng của mình qua số sao (từ 1 đến 5 sao), thêm hình ảnh sản phẩm và nhận xét về sản phẩm. sau khi đánh giá, đánh giá của khách hàng sẽ được hiển thị ở trang chi tiết sản phẩm để những khách hàng khác có thể tham khảo.



Hình 42: Giao diện chức năng đánh giá

## 3.11. Trò chuyện qua [plugin messenger](https://docs.google.com/document/d/1jwGFYHO311zTb1w8LDWIOa3Hen2bVSLxlaFtuKWO8vM/edit#heading=h.20xfydz) facebook

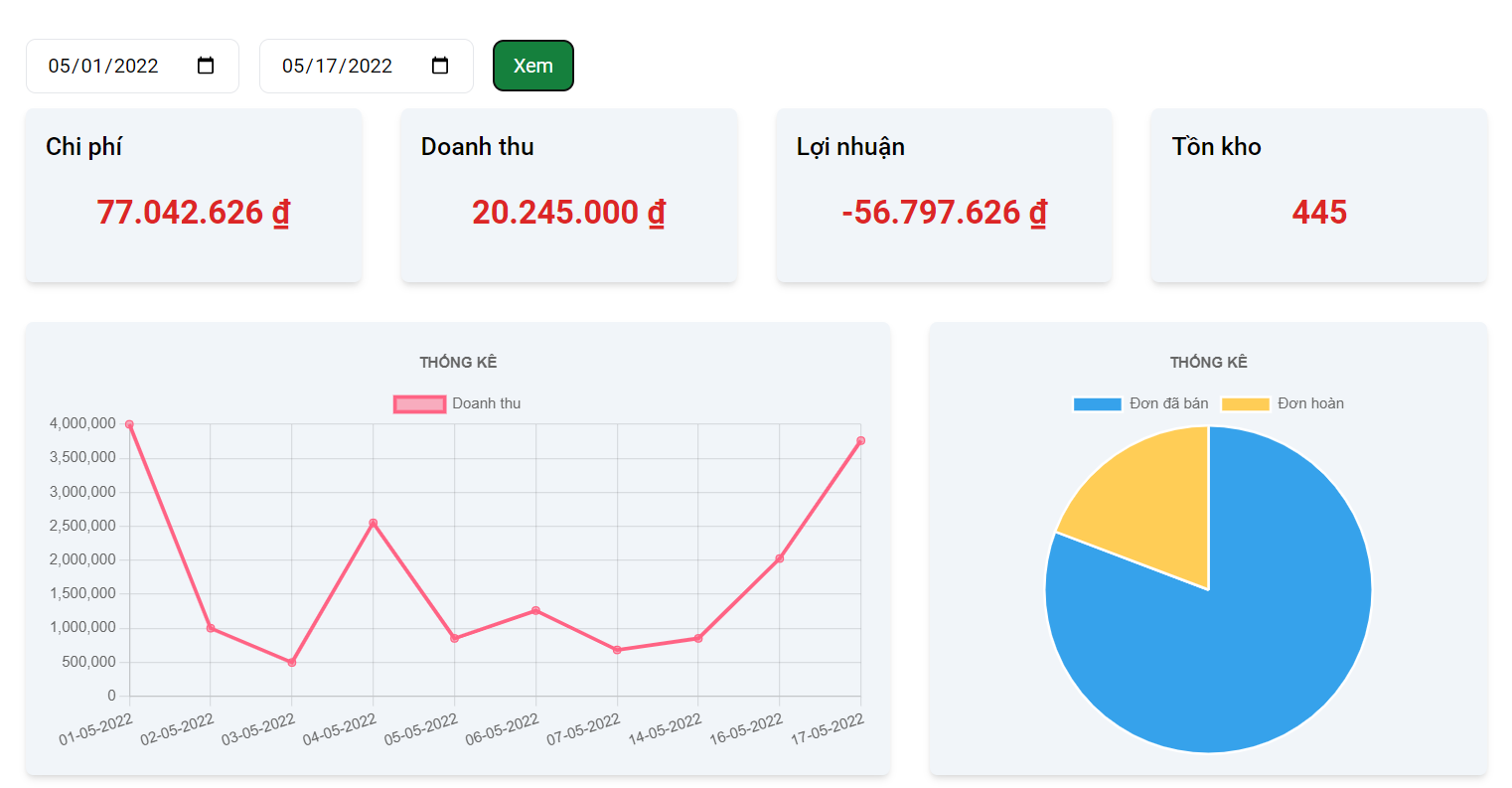
Để có thể hỗ trợ về sản phẩm cũng như giải đáp thắc mắc, nhận góp ý trực tiếp từ khách hàng. [Plugin messenger](https://docs.google.com/document/d/1jwGFYHO311zTb1w8LDWIOa3Hen2bVSLxlaFtuKWO8vM/edit#heading=h.20xfydz) giúp khách hàng nhắn tin trực tiếp với cửa hàng thông qua tài khoản facebook.



Hình 43: Flugin messenger facebook

## 3.12. Giao diện quản lý

Giao diện thống kê sẽ hiển thị khi quản lý đăng nhập vào trang quản lý. Chức năng thống kê thể hiện cửa hàng phát triển dựa vào doanh thu, chi phí và lợi nhuận. Thống kê qua biểu đồ giúp người quản lý nắm rõ được thông tin phát triển của cửa hàng hơn. Việc thống kê cũng có thể chọn từ ngày bắt đầu và ngày kết thúc



Hình 44: Giao diện trang thống kê

### 3.12.1. Sản phẩm

- Thêm sản phẩm

Thêm sản phẩm được coi là chức năng quan trọng của hệ thống. việc thêm sản phẩm gồm những thông tin như: tên sản phẩm, hình ảnh, loại sản phẩm, thương hiệu, giá bán, màu sắc, kích thước và thông tin của sản phẩm.

Một sản phẩm có thể có nhiều màu sắc và nhiều kích thước.

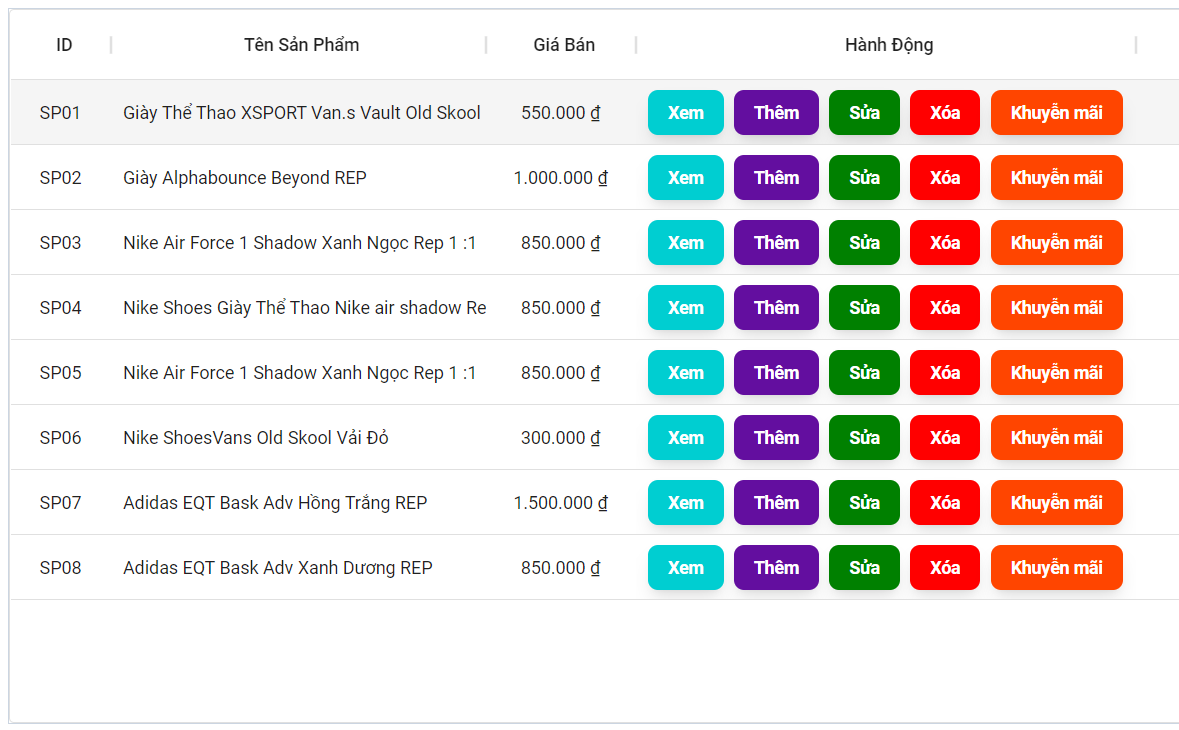


Hình 45: Giao diện chức năng thêm sản phẩm

* Danh sách sản phẩm

Tất cả sản phẩm sau khi được thêm sẽ hiển thị dưới dạng danh sách. Mỗi sản phẩm sẽ có 5 chức năng:

* + Xem: chức năng xem sẽ hiển thị tất cả thông tin của sản phẩm bao gồm: tên sản phẩm, giá sản phẩm, loại sản phẩm, màu sác, kích thước, thông tin sản phẩm, …
  + Thêm: sản phẩm sẽ được thêm khi sản phẩm đó đã hết hàng.
  + Sửa: thông tin sản phẩm sẽ được thay đổi để phù hợp với sản phẩm hiện tại.
  + Xóa: sản phẩm sẽ bị xóa khỏi hệ thống.
  + Khuyến mãi: sản phẩm sẽ được giảm giá theo phần trăm giảm giá.



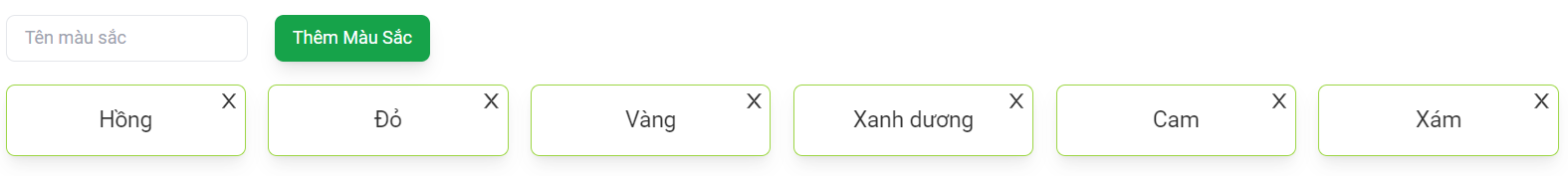
Hình 46: Giao diện danh sách sản phẩm

### 3.12.2. Quản lý danh mục

Quản lý danh mục bao gồm:

* Màu sắc
* Kích thước
* Loại sản phẩm
* Thương hiệu

Quản lý có thể thêm và xóa danh mục



Hình 47: Giao diện danh mục màu sắc



Hình 48: Giao diện danh mục kích thước



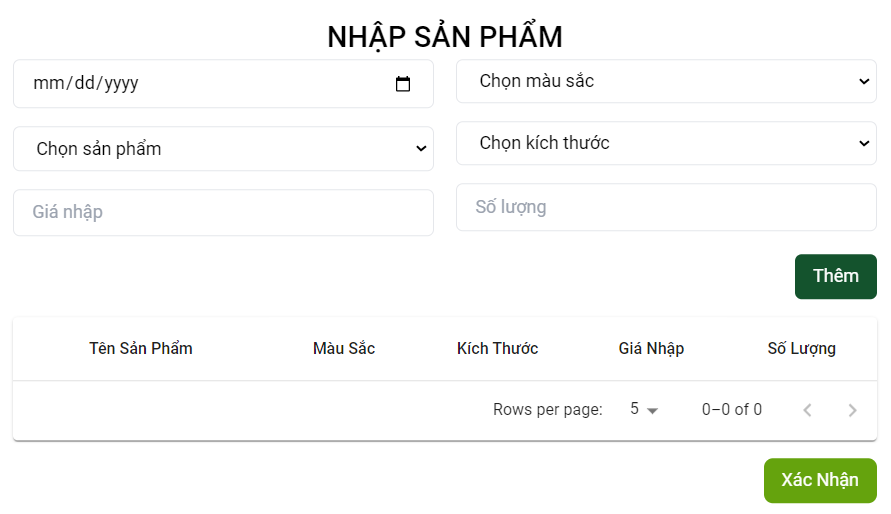
Hình 49: Giao diện danh mục thương hiệu



Hình 50: Giao diện danh mục loại sản phẩm

### 3.12.3. Quản lý nhập kho

Để thêm số lượng sản phẩm cũng như giá đã nhập sản phẩm. quản lý phải lập phiếu nhập, phiếu nhập bao gồm: ngày nhập sản phẩm, chọn sản phẩm, giá nhập, màu sắc, kích thước và số lượng sản phẩm. Quản lý có thể thêm nhiều sản phẩm trong một phiếu nhập.



Hình 51: Chức năng nhập sản phẩm

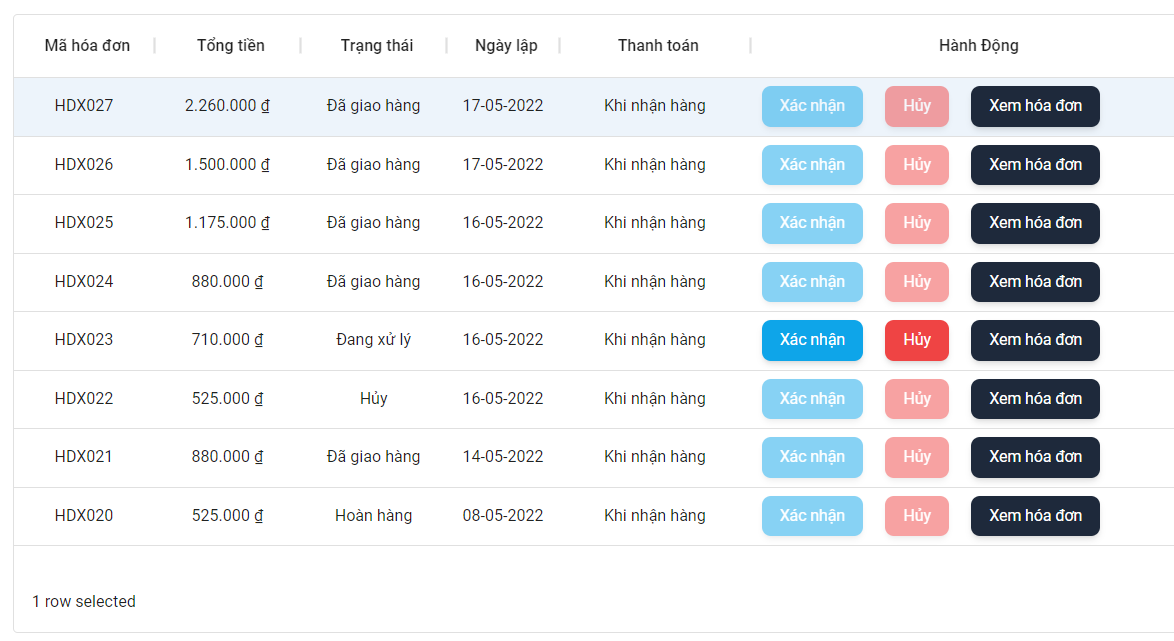
Thông tin chi tiết phiếu nhập sẽ được hiển thị chi tiết gồm: thông tin sản phẩm được thêm và tổng tiền của phiếu nhập đó.



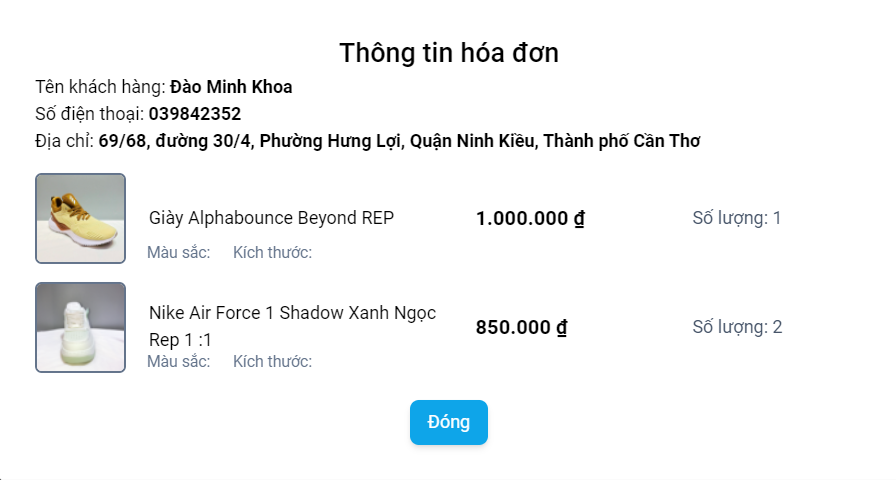
Hình 52: Thông tin chi tiết hóa đơn nhập

### 3.12.4. Quản lý hóa đơn

Sau khi khách hàng thực hiện thanh toán đơn hàng, quản lý sẽ chuẩn bị sản phẩm và xác nhận đơn hàng đó. Nếu cảm thấy đơn hàng có vấn đề, người quản lý có thể hủy đơn hàng đó, quản lý cũng có thể xem nhiều thông tin nữa về đơn hàng qua chức năng xem hóa đơn.



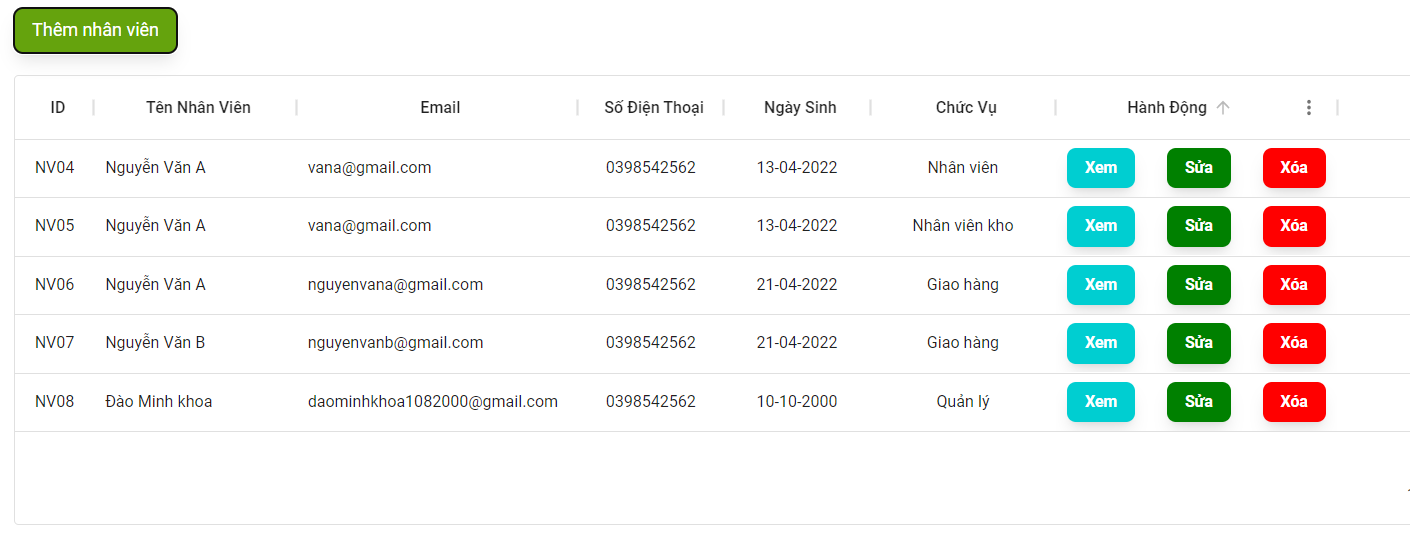
Hình 53: Danh sách hóa đơn



Hình 54: Chức năng xem hóa đơn

### 3.12.5. Quản lý nhân viên

Nhân viên của cửa hàng sẽ được liệt kê dưới dạng danh sách gồm: tên nhân viên, email, số điện thoại, ngày sinh, chức vụ, …Quản lý cũng có thể xem chi tiết nhân viên hơn qua chức năng xem, sửa và xóa nhân viên.



Hình 55: Danh sách nhân viên

Nhân viên sẽ được phân quyền ngay khi quản lý thêm nhân viên. Việc thêm nhân viên sẽ phải điền đầy đủ các thông tin như: họ tên, email, mật khẩu, số điện thoại, ngay sinh, giới tính, địa chỉ. Ở phần chức vụ, người quan r lý sẽ phân quyền nhân viên/

# CHƯƠNG 4: KIỂM THỬ VÀ ĐÁNH GIÁ

## 4.1. Kiểm thử

### 4.1.1. Mục tiêu

* Ngăn chặn lỗi.
* Tìm các lỗi phát sinh khi code.
* Các chức năng của hệ thống hoạt động tốt không.
* Đảm bảo sản phẩm cuối cùng được hoàn thiện trước khi bàn giao sản phẩm cho người kinh doanh.

Quá trình kiểm thử:

* + Kiểm tra điều hướng của website: **Kiểm thử website**với tất cả các tùy chọn như UI/ UX, menu, liên kết hoặc các button trên website phải hiển thị và có thể truy cập được. Điều hướng trang web dễ dàng sử dụng. Nội dung hướng dẫn rõ ràng và đáp ứng được mục đích. Đảm bảo tất cả các tùy chọn trên header, footer và các điều hướng nhất quán trên mỗi trang.
  + Kiểm thử cơ sở dữ liệu: Kiểm thử website về cơ sở dữ liệu để đảm bảo độ tin cậy của dữ liệu. Các hoạt động kiểm tra bao gồm: Kiểm tra nếu các truy vấn được thực hiện mà không xảy ra lỗi. Kiểm tra việc thêm mới, cập nhật hoặc xóa dữ liệu trong cơ sở dữ liệu nên duy trì tính toàn vẹn. Truy vấn dữ liệu không nên mất quá nhiều thời gian. Kiểm tra việc tải dữ liệu và kết quả nhận được với các truy vấn dài. Dữ liệu nhận được có chính xác hay không…
  + Kiểm tra nội dung của trang web: Nội dung website có bị mắc lỗi chính tả hoặc lỗi ngữ pháp trong nội dung không? Nội dung trên website phải rõ ràng và dễ hiểu không.

### 4.1.2. Phạm vi kiểm thử

Tất cả chức năng của khách hàng và người quản lý, giao diện, và điều hướng website

## 4.2. Quản lý kiểm thử

### 4.2.1. Môi trường kiểm thử

* Phần cứng:
  + Bộ xử lý: Intel core I5 8250U
  + RAM: 8GB
  + Ổ cứng: SSD 240GB
  + Cấu hình mạng: có kết nối Internet
* Phần mềm:
  + Hệ điều hành: Window 11
  + Trình duyệt: Google Chrome, Microsoft Edge
  + Hệ quản trị cơ sở dữ liệu: MySQL

### 4.2.2. Tài nguyên và cấp phát tài nguyên

Tài nguyên kiểm thử: Laptop

## 4.3. Các trường hợp kiểm thử

### 4.3.1. Chức năng đăng ký, đăng nhập và lấy lại mật khẩu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường hợp  kiểm thử | Các bước kiểm thử | Kết quả mong đợi | Kết quả thực tế | Thanh công/  thất bại |
| 1 | Khách hàng đăng ký với thông tin hợp lệ | B1: chọn đăng ký tại trang chủ.  B2: Điền đầy đủ thông tin  B3: Nhập mã xác thực gửi về email | Đăng ký thành công | Đăng ký thành công | Thành công |
| 2 | Khách hàng đăng nhập với thông tin hợp lệ | B1: Chọn đăng nhập tại trang chủ  B2: Điền đầy đủ thông tin tài khoản  B3: Nhập mã xác thực | Đăng nhập thành công | Đăng nhập thành công | Thành công |
| 3 | Quản lý và nhân viên đăng nhập với thông tin hợp lệ | B1: Chọn đăng nhập tại trang quản lý  B2: Điền đầy đủ thông tin tài khoản | Đăng nhập thành công | Đăng nhập thành công | Thành công |
| 4 | Khách hàng chọn lấy lại mật khẩu tại trang đăng nhập | B1: Chọn lấy lại mật khẩu  B2: Nhập email và nhấn gửi  B3: Nhập mật khẩu và nhập mã xác thực đã gửi đến email  B4: Nhấn đổi mật khẩu | Lấy mật khẩu thành công | Lấy mật khẩu thành công | Thành công |

Bảng 38:Kiểm thử chức năng đăng ký và đăng nhập

### 4.3.2. Chức năng đặt hàng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường hợp  kiểm thử | Các bước kiểm thử | Kết quả mong đợi | Kết quả thực tế | Thanh công/  thất bại |
| 1 | Đặt hàng không chọn màu sắc và kích thước | B1: Đăng nhập và chọn sản phẩm bất kỳ  B2: Nhấn chọn “Mua ngay” hoặc “Thêm vào giỏ hàng” | Hiển thị thông báo lỗi | Hiển thị thông báo lỗi | Thành công |
| 2 | Đặt hàng khi sản phẩm tồn kho đã hết | B1: Đăng nhập và chọn sản phẩm đã hết hàng.  B2: Chọn màu sắc và kích thước  B3: Nhấn chọn “Mua ngay” hoặc “Thêm vào giỏ hàng” | Hiển thị thông báo lỗi | Hiển thị thông báo lỗi | Thành công |
| 3 | Đặt hàng với số lượng lớn hơn số lượng sản phẩm trong kho | B1: Đăng nhập và chọn sản phẩm bất kỳ  B2: thêm sản phẩm vào giỏ hàng quá với số lượng tồn kho | Hiển thị thông báo lỗi | Hiển thị thông báo lỗi | Thành công |
| 4 | Đặt hàng khi chưa đăng nhập | B1: Đăng xuất nếu đã đăng nhập  B2: Chọn sản phẩm bất kì, chọn màu sắc và kích thước  B3: Nhấn chọn “Mua ngay” hoặc “Thêm vào giỏ hàng” | Hiển thị thông báo lỗi | Hiển thị thông báo lỗi | Thành công |

Bảng 39: Kiểm thử chức năng đặt hàng

### 4.3.3. Chức năng tìm kiếm, lọc sản phẩm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường hợp  kiểm thử | Các bước kiểm thử | Kết quả mong đợi | Kết quả thực tế | Thanh công/  thất bại |
| 1 | Tìm kiếm với từ khóa có trong hệ thống | Nhập từ khóa có trong hệ thống vào ô tìm kiếm | Có sản phẩm | Có sản phẩm | Thành công |
| 2 | Tìm kiếm với từ khóa không có trong hệ thống | Nhập từ khóa không có trong hệ thống vào ô tìm kiếm | Không có sản phẩm | Không có sản phẩm | Thành công |

Bảng 40:Kiểm thử chức năng tìm kiếm và lọc sản phẩm

### 4.3.4. Chức năng quản lý đơn hàng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường hợp  kiểm thử | Các bước kiểm thử | Kết quả mong đợi | Kết quả thực tế | Thanh công/  thất bại |
| 1 | Xác nhận đơn hàng | B1: Đăng nhập vào trang quản lý  B2: Chọn đơn hàng và tìm đơn hàng cần xác nhận  B3: Nhấn chọn “Xác nhận” và chọn người giao hàng và nhấn “Xác nhận hóa đơn” | Đơn hàng được xác nhận | Đơn hàng được xác nhận | Thành công |
| 2 | Hủy đơn hàng | B1: Đăng nhập vào trang quản lý  B2: Chọn đơn hàng và tìm đơn hàng cần xác nhận  B3: Nhấn chọn “Hủy” | Đơn hàng được hủy và số lượng sản phẩm được cập nhật lại | Đơn hàng được hủy và số lượng sản phẩm được cập nhật lại | Thành công |

Bảng 41:Kiểm thử chức năng quản lý đơn hàng

### 4.3.5. Chức năng lập hóa đơn

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Trường hợp  kiểm thử | Các bước kiểm thử | Kết quả mong đợi | Kết quả thực tế | Thanh công/  thất bại |
| 1 | Không chọn sản phẩm | B1: Đăng nhập vào trang quản lý  B2: Chọn lập hóa đơn và chọn “Nhập hóa đơn”  B3: Điền thông tin nhưng không chọn sản phẩm | Báo lỗi | Báo lỗi | Thành công |
| 2 | Không nhập số lượng | B1: Đăng nhập vào trang quản lý  B2: Chọn lập hóa đơn và chọn “Nhập hóa đơn”  B3: Điền thông tin nhưng không nhập số lượng sản phẩm | Báo lỗi | Báo lỗi | Thành công |

Bảng 42:Kiểm thử chức năng lập hóa đơn

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Kết luận

### 5.1.1. Kết quả đạt được

Sau thời gian nghiên cứu, tìm hiểu công nghệ, kiến thức chuyên môn để thực hiện đề tài thì giúp cho người thực hiện đề tài có cái nhìn tổng quan về quy trình phát triển phần mềm là như thế nào, làm sao để xây dựng được một hệ thống quản lý cửa hàng hoàn chỉnh từ các khâu phân tích, thiết kế đến hoàn thiện hệ thống. Bên cạnh đó, còn học hỏi được rất nhiều những tiện ích khi sử dụng các công cụ StarUML, PowerDesigner, MySQL Workbench ... vào việc phát triển ứng dụng. Xây dựng hoàn thiện hệ thống gồm 2 thành phần RESTful API server, Web client đạt mục tiêu đề ra ban đầu.

### 5.1.2. Hạn chế

* Website chưa hoàn thiện giao diện cho Ipad, điện thoại di động và các giao diện của thiết bị khách.
* Website chưa được tối ưu.
* Cơ sở dữ liệu không nhiều.

## 5.2. Hướng phát triển

1. Tối ưu lại website
2. Thiết kế giao diện cho nhiều thiết bị khác nhau.
3. Thêm nhiều chức năng khác như: bản đồ, trò chuyện real time, quản lý khách hàng, …
4. Tăng cường bảo mật cho website.

# TÀI LIỆU THAM KHẢO

[1]. Nguyễn Cao Hồng Ngọc. Slide bài giảng Lập trình web. Khoa Công nghệ thông tin và Truyền thông, Đại học Cần Thơ.

[2]. Trần Cao Đệ và Nguyễn Công Danh, 2014. Giáo trình đảm bảo chất lượng phần mềm. Nxb Đại học Cần Thơ. Ninh Kiều, Cần Thơ.

[3]. Phạm Thị Xuân Lộc và Phạm Thị Ngọc Diễm, 2012. Giáo trình Ngôn ngữ mô hình hóa UML. Nxb Đại học Cần Thơ. Ninh Kiều, Cần Thơ.

[4]. Nguyễn Thái Nghe, 2014. Giáo trình hệ quản trị cơ sở dữ liệu. Nxb Đại học Cần Thơ. Ninh Kiều, Cần Thơ.

[5]. Huỳnh Xuân Hiệp, 2015. Giáo trình kiến trúc và thiết kế phần mềm. Nxb Đại học Cần Thơ. Ninh Kiều, Cần Thơ.

[6]. ReactJS [online] 01/02/2022: <https://reactjs.org/docs/getting-started.html>

[7]. NodeJS [online] 02/03/2022: <https://www.w3schools.com/nodejs/>

[8]. Axios [online] 01/04/2022: <https://axios-http.com/docs/req_config>

[9]. React router [online] 30/4/2022:<https://reactrouter.com/web/guides/quick-start>

[10]. VNPAY [online] 01/05/2022: <https://sandbox.vnpayment.vn/apis/docs/huong-dan-tich-hop/>

# PHỤ LỤC

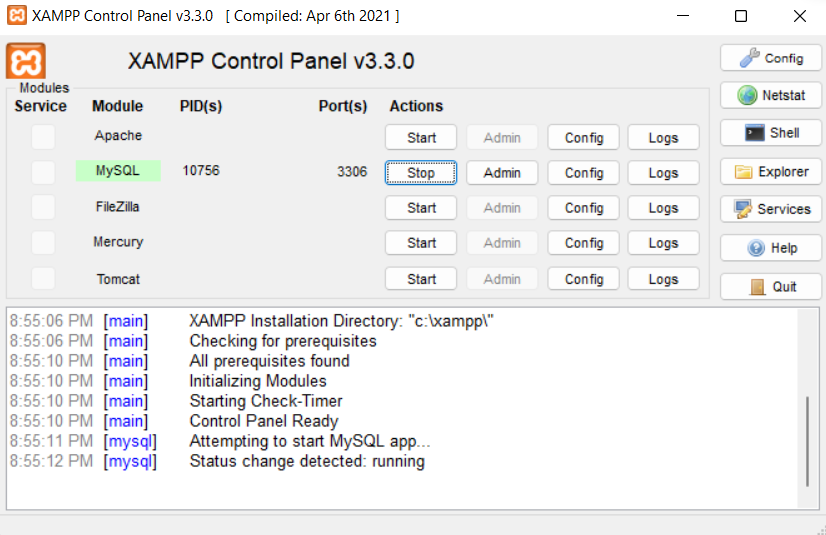
## 1. Hướng dẫn cài đặt

Bước 1: Mở cửa sổ lệnh CMD trên hệ điều hành.

Bước 2: Truy cập vào thư mục chứa dự án.

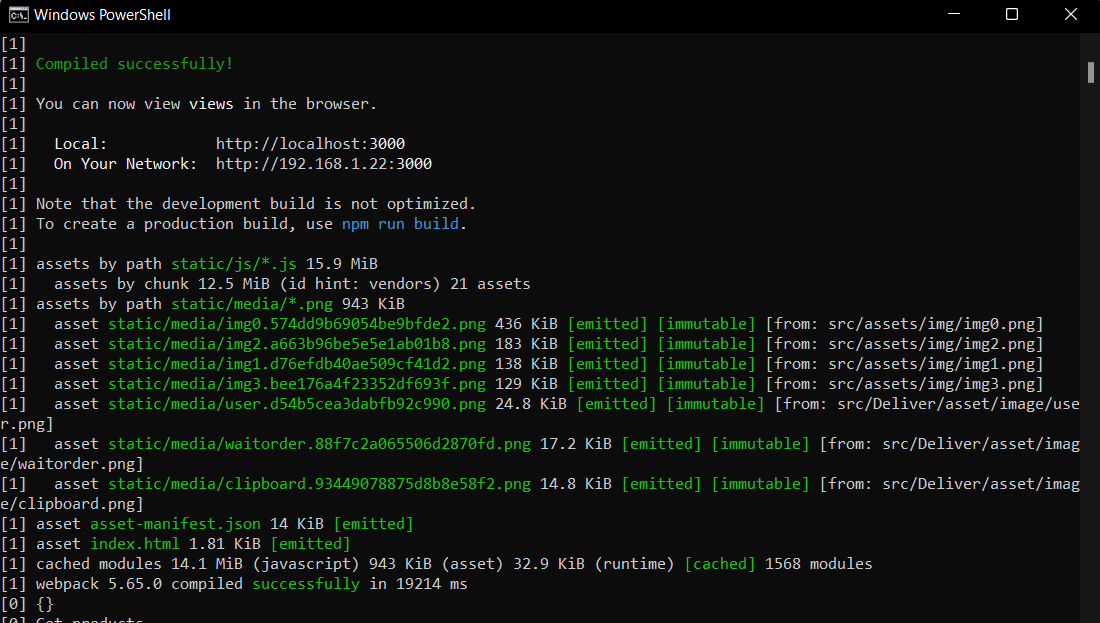
Bước 3: Gõ lệnh npm install và chờ cho đến khi hoàn tất việc cài đặt dự án.

Bước 4: Mở Xampp, start MySQL. Nhập cơ sở dữ iệu.



*Phụ lục – Hình 1: Giao diện phần mềm XAMPP*

Bước 5: Gõ lệnh npm run dev trong cửa sổ lệnh CMD và đợi dự án chạy



*Phụ lục – Hình 2: Giao diện cửa sổ dòng lệnh*