**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**



**ĐỒ ÁN TỐT NGHIỆP**

**TÊN ĐỀ TÀI: NGHIÊN CỨU VÀ XÂY DỰNG WEBSITE BÁN HÀNG ONLINE**

**Sinh viên thực hiện : Lộc Văn Minh**

**Ngày sinh : 15/03/1997**

**Lớp : DCCNTT7.10**

**Khoa : Công nghệ thông tin**

**Mã sinh viên : 1652480201004**

**Giáo viên hướng dẫn : ThS. Nguyễn Viết Hùng**

**Bắc Ninh – Năm 2020**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**



**LỘC VĂN MINH**

**TÊN ĐỀ TÀI: NGHIÊN CỨU VÀ XÂY DỰNG WEBSITE BÁN HÀNG ONLINE**

**Giáo viên hướng dẫn: ThS. Nguyễn Viết Hùng**

**Bắc Ninh – Năm 2020**

MỤC LỤC

[LỜI CẢM ƠN ii](#_Toc55402641)

[MỞ ĐẦU iii](#_Toc55402642)

[DANH MỤC HÌNH VẼ iv](#_Toc55402643)

[DANH MỤC BẢNG BIỂU vi](#_Toc55402644)

[DANH MỤC TỪ VIẾT TẮT vii](#_Toc55402645)

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 2](#_Toc55402646)

[1.1. Tổng quan về đề tài 2](#_Toc55402647)

[1.1.1. Lý do chọn đề tài 2](#_Toc55402648)

[1.1.2. Mục đích , ý nghĩa của đề tài 2](#_Toc55402649)

[1.2. Mongo DB 3](#_Toc55402650)

[1.2.1. Cơ sở dữ liệu NoSQL 3](#_Toc55402651)

[1.2.2. Cơ sở dữ liệu MongoDB 6](#_Toc55402652)

[1.3. ExpressJS 8](#_Toc55402653)

[1.3.1. ExpressJs là gì 8](#_Toc55402654)

[1.3.2. Routing trong ExpressJS 8](#_Toc55402655)

[1.3.3. Middleware trong ExpressJS 10](#_Toc55402656)

[1.4. ReactJS 11](#_Toc55402657)

[1.4.1. React là gì ? 11](#_Toc55402658)

[1.4.2. JSX 11](#_Toc55402659)

[1.4.3. Ưu nhược điểm của ReactJS 11](#_Toc55402660)

[1.5. NodeJS 13](#_Toc55402661)

[1.5.1. Giới thiệu về NodeJS 13](#_Toc55402662)

[1.5.2. Ứng dụng của NodeJS 13](#_Toc55402663)

[1.5.3. Ưu nhược điểm của Nodejs 14](#_Toc55402664)

[CHƯƠNG 2. PHÂN TÍCH HỆ THỐNG 16](#_Toc55402665)

[2.1. Biểu đồ hoạt động 16](#_Toc55402666)

[2.1.1. Biểu đồ hoạt động chức năng thêm/xóa/sửa giỏ hàng 16](#_Toc55402667)

[2.1.2. Biểu đồ hoạt động chức năng thêm/xóa/sửa sản phẩm 17](#_Toc55402668)

[2.1.3. Biểu đồ hoạt động chức năng thêm/xóa/sửa danh mục sản phẩm 18](#_Toc55402669)

[2.1.4. Biểu đồ hoạt động chức năng tìm kiếm 19](#_Toc55402670)

[2.1.5. Biểu đồ hoạt động chức năng đặt hàng 20](#_Toc55402671)

[2.2. Biểu đồ ca sử dụng 21](#_Toc55402672)

[2.2.1. Biểu đồ ca sử dụng tổng quát 21](#_Toc55402673)

[2.2.2. Biểu đồ ca sử dụng khách hàng tìm kiếm sản phẩm 22](#_Toc55402674)

[2.2.3. Biểu đồ ca sử dụng khách hàng thanh toán 23](#_Toc55402675)

[2.2.4. Biểu đồ ca sử dụng thêm sản phẩm vào giỏ hàng 24](#_Toc55402676)

[2.2.5. Biểu đồ ca sử dụng quản lý danh mục sản phẩm 25](#_Toc55402677)

[2.2.6. Biểu đồ ca sử dụng quản lý đơn hàng. 26](#_Toc55402678)

[2.2.7. Biểu đồ ca sử dụng quản lý sản phẩm 27](#_Toc55402679)

[2.3. Biểu đồ tuần tự. 28](#_Toc55402680)

[2.3.1. Biểu đồ tuần tự đăng nhập 28](#_Toc55402681)

[2.3.2. Biểu đồ tuần tự cập nhật danh mục sản phẩm 29](#_Toc55402682)

[2.3.3. Biểu đồ tuần tự thêm danh mục sản phẩm 30](#_Toc55402683)

[2.3.4. Biểu đồ tuần tự xóa danh mục sản phẩm 31](#_Toc55402684)

[2.3.5. Biểu đồ tuần tự thêm sản phẩm 32](#_Toc55402685)

[2.3.6. Biểu đồ tuần tự cập nhật sản phẩm 33](#_Toc55402686)

[2.3.7. Biểu đồ tuần tự xóa sản phẩm 34](#_Toc55402687)

[2.3.8. Biểu đồ tuần tự đăng ký 35](#_Toc55402688)

[2.3.9. Biểu đồ tuần tự tìm kiếm sản phẩm 36](#_Toc55402689)

[2.3.10. Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng 37](#_Toc55402690)

[2.3.11. Biểu đồ tuần tự cập nhật sản phẩm vào giỏ hàng 38](#_Toc55402691)

[2.3.12. Biểu đồ tuần tự xóa sản phẩm khỏi giỏ hàng 39](#_Toc55402692)

[2.4. Mô hình thực thể liên kết ERD 40](#_Toc55402693)

[CHƯƠNG 3. THIẾT KẾ HỆ THỐNG 41](#_Toc55402694)

[3.1. Thiết kế cơ sở dữ liệu 41](#_Toc55402695)

[3.1.1. Thiết kế cơ sở dữ liệu mức logic 41](#_Toc55402696)

[3.1.2. Thiết kế cơ sở dữ liệu mức vật lý 41](#_Toc55402697)

[3.2. Thiết kế giao diện 44](#_Toc55402698)

[3.2.1. Giao diện trang chủ 44](#_Toc55402699)

[3.2.2. Giao diện trang sản phẩm 45](#_Toc55402700)

[3.2.3. Giao diện trang admin dashboard 46](#_Toc55402701)

[3.2.4. Giao diện trang user dashboard 46](#_Toc55402702)

[3.2.5. Giao diện trang đăng ký 47](#_Toc55402703)

[3.2.6. Giao diện trang đăng nhập 47](#_Toc55402704)

[3.2.7. Giao diện trang quản lý sản phẩm 48](#_Toc55402705)

[3.2.8. Giao diện trang thêm sản phẩm 49](#_Toc55402706)

[3.2.9. Giao diện trang sửa thông tin sản phẩm 50](#_Toc55402707)

[3.2.10. Giao diện trang quản lý danh mục sản phẩm 51](#_Toc55402708)

[3.2.11. Giao diện trang thêm danh mục sản phẩm 51](#_Toc55402709)

[3.2.12. Giao diện trang sửa danh mục sản phẩm 52](#_Toc55402710)

[3.2.13. Giao diện trang quản lý giỏ hàng 52](#_Toc55402711)

[3.2.14. Giao diện trang quản lý đơn hàng 53](#_Toc55402712)

[3.2.15. Giao diện trang xem chi tiết sản phẩm 53](#_Toc55402713)

[KẾT LUẬN 54](#_Toc55402714)

[Kết quả đạt được 54](#_Toc55402715)

[Hạn chế của đề tài 54](#_Toc55402716)

[Hướng phát triển 54](#_Toc55402717)

[TÀI LIỆU THAM KHẢO 55](#_Toc55402718)

# LỜI CẢM ƠN

Lời đầu tiên em xin trân thành cảm ơn quý thầy cô đã giúp em thực hiện đề tài này. Đặc biệt thầy **Nguyễn Viết Hùng** đã tận tình hướng dẫn, giúp đỡ, chỉ bảo em trong suốt thời gian thực hiện khóa luận tốt nghiệp.

Đồng thời em xin trân trọng cảm ơn những tình cảm quý báu mà các thầy cô trong trường Đại Học Công Nghệ Đông Á truyền đạt cho em những những kinh nghiệm, kỹ thuật và cách thức trong việc xây dựng đề tài này.

Tuy nhiên do thời gian còn hạn chế em chưa thể phát huy hết những ý tưởng, khả năng hỗ trợ của ngôn ngữ lập trình và kỹ thuật lập trình vào đề tài này. Trong thời gian thực hiện đề tài không tránh khỏi những thiếu xót, Em mong nhận được sự đóng góp từ quý thầy cô và các bạn.

Em xin chân thành cảm ơn.

|  |
| --- |
| Bắc Ninh, ngày … tháng … năm ….  Sinh viên thực thiện  Lộc Văn Minh |

# MỞ ĐẦU

Trong những năm gần đây với sự phát triển mạnh mẽ của khoa học công nghệ, Công nghệ thông tin là một trong những ngành có vai trò rất lớn trong sự phát triển của đất nước. Các ứng dụng của Công nghệ thông tin được áp dụng trong mọi lĩnh vực nghiên cứu khoa học cũng như trong đời sống. Trong đó Website đang có sự phát triển mạnh mẽ và phổ biến bởi những lợi ích mà nó đem lại cho cộng đồng.

Bằng việc lựa chọn và thực hiện đề tài **“Nghiên cứu và xây dựng website bán hàng online”,** Em muốn tìm hiểu và đưa ra giải pháp nhằm giải quyết công việc bán hàng trực tuyến mà xã hội đang có nhu cầu. Website mang lại rất nhiều lợi ích như: quảng cáo tên tuổi công ty rất nhanh trên môi trường mạng toàn cầu, việc kinh doanh có thể mở 24/24, giảm thiểu chi phí nhân viên cho tiếp thị, dễ dàng nhận phản hồi từ phía khách hàng, cơ hội mở rộng liên kết và hợp tác ở phạm vi quốc tế…

Website bán hàng giúp cho công ty cung cấp cho khách hàng những lựa chọn linh hoạt và tiện lợi trong việc tìm mua sản phẩm thông qua chức năng tìm kiếm và giỏ hàng. Các thông tin về sản phẩm được hiển thị chi tiết với từng sản phẩm, từ đó khách hàng dễ dàng nhận biết và lựa chọn thứ mình cần. Về phía công ty hệ thống mang lại sự tiện dụng trong việc cập nhập, thống kê và quản lý thông tin cho khách hàng tạo mối quan hệ thân thiện và dễ dàng với khách hàng.

# DANH MỤC HÌNH VẼ

[Hình 1.1 Cấu trúc của một Document trong MongoDB 7](#_Toc55403430)

[Hình 1.2 Code ví dụ về Route 9](#_Toc55403431)

[Hình 2.1 Biểu đồ hoạt động chức năng thêm/xóa/sửa giỏ hàng 16](#_Toc55403432)

[Hình 2.2 Biểu đồ hoạt động chức năng thêm/xóa/sửa sản phẩm 17](#_Toc55403433)

[Hình 2.3 Biểu đồ hoạt động chức năng thêm/xóa/sửa danh mục 18](#_Toc55403434)

[Hình 2.4 Biểu đồ hoạt động chức năng tìm kiếm 19](#_Toc55403435)

[Hình 2.5 Biểu đồ hoạt động chức năng đặt hàng 20](#_Toc55403436)

[Hình 2.6 Biểu đồ ca sử dụng tổng quát 21](#_Toc55403437)

[Hình 2.7 Biểu đồ ca sử dụng chức năng tìm kiếm sản phẩm 22](#_Toc55403438)

[Hình 2.8 Biểu đồ ca sử dụng khách hàng thanh toán 23](#_Toc55403439)

[Hình 2.9 Biểu đồ ca sử dụng thêm sản phẩm vào giỏ hàng 24](#_Toc55403440)

[Hình 2.10 Biểu đồ ca sử dụng quản lý danh mục sản phẩm 25](#_Toc55403441)

[Hình 2.11 Biểu đồ ca sử dụng quản lý đơn hàng 26](#_Toc55403442)

[Hình 2.12 Biểu đồ ca sử dụng quản lý sản phẩm 27](#_Toc55403443)

[Hình 2.13 Biểu đồ tuần tự đăng nhập 28](#_Toc55403444)

[Hình 2.14 Biểu đồ tuần tự cập nhật danh mục sản phẩm 29](#_Toc55403445)

[Hình 2.15 Biểu đồ tuần tự thêm dan mục sản phẩm 30](#_Toc55403446)

[Hình 2.16 Biểu đồ tuần tự xóa danh mục sản phẩm 31](#_Toc55403447)

[Hình 2.17 Biểu đồ tuần tự thêm sản phẩm 32](#_Toc55403448)

[Hình 2.18 Biểu đồ tuần tự cập nhật sản phẩm 33](#_Toc55403449)

[Hình 2.19 Biểu đồ tuần tự xóa sản phẩm 34](#_Toc55403450)

[Hình 2.20 Biểu đồ tuần tự đăng ký 35](#_Toc55403451)

[Hình 2.21 Biểu đồ tuần tự tìm kiếm sản phẩm 36](#_Toc55403452)

[Hình 2.22 Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng 37](#_Toc55403453)

[Hình 2.23 Biểu đồ tuần tự cập nhật vào giỏ hàng 38](#_Toc55403454)

[Hình 2.24 Biểu đồ tuần tự xóa sản phẩm khỏi giỏ hàng 39](#_Toc55403455)

[Hình 2.25 Mô hình thực thể liên kết 40](#_Toc55403456)

[Hình 3.1 Quan hệ giữa các document 41](#_Toc55403457)

[Hình 3.2 Giao diện trang chủ 44](#_Toc55403458)

[Hình 3.3 Giao diện trang sản phẩm 45](#_Toc55403459)

[Hình 3.4 Giao diện trang admin dashboard 46](#_Toc55403460)

[Hình 3.5 Giao diện trang user dashboard 46](#_Toc55403461)

[Hình 3.6 Giao diện trang đăng ký 47](#_Toc55403462)

[Hình 3.7 Giao diện trang đăng nhập 47](#_Toc55403463)

[Hình 3.8 Giao diện trang quản lý sản phẩm 48](#_Toc55403464)

[Hình 3.9 Giao diện trang thêm sản phẩm 49](#_Toc55403465)

[Hình 3.10 Giao diện trang sửa thông tin sản phẩm 50](#_Toc55403466)

[Hình 3.11 Giao diện trang quản lý danh mục sản phẩm 51](#_Toc55403467)

[Hình 3.12 Giao diện trang sửa danh mục sản phẩm 51](#_Toc55403468)

[Hình 3.13 Giao diện trang sửa danh mục sản phẩm 52](#_Toc55403469)

[Hình 3.14 Giao diện trang giỏ hàng 52](#_Toc55403470)

[Hình 3.15 Giao diện trang quản lý đơn hàng 53](#_Toc55403471)

[Hình 3.16 Giao diện trang xem chi tiết sản phẩm 53](#_Toc55403472)

# DANH MỤC BẢNG BIỂU

[Bảng 1.1 So sánh RDBMS và NOSQL 5](#_Toc50157557)

[Bảng 1.2 So sánh các khái niệm giữa RDBMS và MongoDB 8](#_Toc50157558)

[Bảng 3.1 Bảng user 42](#_Toc50157559)

[Bảng 3.2 Category 42](#_Toc50157560)

[Bảng 3.3 Bảng product 42](#_Toc50157561)

[Bảng 3.4 Bảng order 43](#_Toc50157562)

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Ký hiệu** | **Viết đầy đủ** |
| **1** | RDBMS | Relational Database Management System |
| **2** | JSON | Javascript Object Notation |
| **3** | API | Application Programming Interface |
| **4** | CRUD | Create, Read, Update, Delete |
| **5** | NPM | Node package manager |
| **6** | DOM | Document Object Model |
| **7** | JSX | Javascript XML |
| **8** | HTTP | Hypertext Transfer Protocol |
| **9** | URL | Uniform Resource Locator |

# TỔNG QUAN VỀ ĐỀ TÀI

## Tổng quan về đề tài

### Lý do chọn đề tài

Ngày nay, Công nghệ thông tin có vai trò quan trọng, là động lực thúc đẩy tăng trưởng kinh tế và kích thích tiềm năng đổi mới đối với nền kinh tế toàn cầu nói chung và nền kinh tế quốc dân của mỗi nước nói riêng.

Với nhiều thành quả quan trọng, Công nghệ thông tin đã được ứng dụng rộng dãi trong nhiều lĩnh vực, trở thành một trong những yếu tố quan trọng nhất của sự phát triển kinh tế - xã hội, an ninh - quốc phòng. Việc đẩy mạnh Ứng dụng công nghệ thông tin sẽ làm thay đổi tác phong làm việc, nâng cao năng suất lao động, nâng cao dân trí,…Chính vì thế, Công nghệ thông tin sẽ trở thành nền kinh tế mũi nhọn của nước ta trong những năm tới đây.

Hơn nữa thương mại điện tử là một lĩnh vực mới đang được nhiều cơ quan, nhiều công ty, nhiều đoàn thể cũng như cá nhân nghiên cứu và phát triển nhằm đưa các thiết bị, các ứng dụng liên quan đến điện tử tin học vào phục vụ cho sự phát triển kinh doanh. Thương mại điện tử là một trong những lĩnh vực được xem là mang tính thời sự ở Việt nam, ở nước ta cũng đã có nhiều tổ chức, cá nhân nghiên cứu, ứng dụng thương mại điện tử vào kinh doanh nhưng đây là một lĩnh vực mới mẻ và còn rất non trẻ. Để góp một phần nhỏ vào sự phát triển Thương mại điện tử ở nước ta Em đã quyết định thực hiện đề tài “**xây dựng website bán hàng**”.

Với những thao tác đơn giản trên máy có nối mạng Internet người tiêu dùng sẽ có tận tay những gì mình cần chỉ cần vào các trang web bán hàng, làm theo hướng dẫn và click vào những gì bạn cần. Không cần đi đâu xa, chỉ cần ngồi tại chỗ là có thể chọn lựa thỏa thích các mặt hàng mà mình yêu thích và được giao hàng tận nơi, tiết kiệm được thời gian, công sức và cả tiền bạc.

### Mục đích , ý nghĩa của đề tài

Hệ thống là một chương trình phần mềm giúp cho các doanh nghiệp hay các chủ cửa hàng giới thiệu và bán sản phẩm của mình tới người tiêu dùng một cách nhanh nhất thông qua trang web bán hàng. Và thông qua trang web này người dùng có thể tiếp cận mặt hàng chỉ với những thao tác đơn giản trên thiết bị có nối mạng Internet bạn sẽ có tận tay những gì mình cần mà không phải mất nhiều thời gian

Đề tài này giúp chúng ta hiểu rõ hơn nghiệp vụ quản lý bán hàng qua mạng.

Ứng dụng đề tài vào việc tối ưu hóa các chương trình quản lý sản phẩm trong các cửa hàng.

## Mongo DB

### Cơ sở dữ liệu NoSQL

#### Khái niệm NoSQL

NoSQL là Non-Relational: tạm dịch là không có quan hệ. NoSQL là cơ sở dữ liệu không quan hệ, không ràng buộc giữa các Collection (tương đương bảng trong cơ sở dữ liệu quan hệ). Tức là giữa các Collection (Bảng) sẽ không có khóa chính, khóa ngoại như trong cơ sở dữ liệu quan hệ. Việc này sẽ giảm thiểu độ phức tạp khi cơ sở dữ liệu chúng ta có nhiều bảng. NoSQL đặc biệt nhấn mạnh đến mô hình lưu trữ cặp key - value và hệ thống lưu trữ phân tán.

#### Ưu nhược điểm của CSDL NoSql

1. Ưu điểm của NoSQL:

Là mã nguồn mở: không phải mất chi phí và có xu hướng tin cậy, an ninh và nhanh hơn khi triển khai so với các hệ quản trị cơ sở dữ liệu độc quyền.

Linh hoạt trong việc mở rộng và phát triển: giúp dễ dàng mở rộng máy chủ khi dữ liệu càng ngày càng lớn hoặc lượng truy cập và lượng dữ liệu quá lớn, thay vì thuê một máy chủ lớn hơn để thay thế cho máy chủ hiện tại thì ta chỉ cần thuê thêm một máy chủ khác. Về phía người lập trình thì giúp dễ dàng thêm Collection hoặc cặp dữ liệu. Dễ dàng trong việc thống kê, truy vấn nhanh.

Áp dụng được công nghệ điện toán đám mây: dễ dàng mở rộng phạm vi được theo yêu cầu có sử dụng một dịch vụ như là Amazon EC2. Giống như tất cả công nghệ đám mây, EC2 dựa vào ảo hóa. Liên kết yếu của ảo hóa là sự thực thi của I/O, với bộ nhớ và CPU các các kết nối mạnh.

NoSQL lưu trữ dữ liệu thường được mở rộng phạm vi theo chiều ngang tận dụng được sự cung cấp mềm dẻo của đám mây. Giúp mở rộng dữ liệu dễ dàng hơn.

1. Nhược điểm của NoSQL:

Hỗ trợ không đồng đều cho các doanh nghiệp. Các doanh nghiệp vừa được sự hỗ trợ tốt nhất từ nhà cung cấp RMBMS (vì đã được phát triển trước NoSQL một thời gian khá dài) còn các doanh nghiệp nhỏ thì thường sử dụng các mã nguồn mở thì không được sự hỗ trợ tốt nhất.

Chưa thực sự phổ biến vì NoSQL vẫn chưa nhận được sự tin cậy với nhiều doanh nghiệp, một phần dữ liệu đã được xây dựng từ lâu, nên việc chuyển đổi cũng là vấn đề khó với nhiều doanh nghiệp và chưa được hỗ trợ tốt về chức năng cũng như sự ổn định như RMBMS.

Còn mới lạ với một số lập trình viên. Chưa được sử dụng để đào tạo rộng rãi. Chưa có nhiều công cụ hỗ trợ giao diện tương tác cũng như các phương thức tốt nhất dẫn đến còn hạn chế về mặt nghiệp vụ.

Chia sẻ dữ liệu chưa theo một tiêu chuẩn chung. Mỗi CSDL NoSQL có các giao diện lập trình ứng dụng API riêng của mình. Sự thiếu hụt các tiêu chuẩn có nghĩa là nó không có khả năng để chuyển một cách đơn giản từ một nhà cung cấp này sang một nhà cung cấp khác.

#### So sánh NoSql với CSDL quan hệ

|  |  |  |
| --- | --- | --- |
| **Tiêu chí đánh giá** | **RDBMS** | **NoSQL** |
| Đánh chỉ mục lượng lớn dữ liệu, phân trang hoặc phân phối luồng dữ liệu media | Có nhiều điểm yếu kém | Thực hiện khá tốt. |
| Mô hình dữ liệu | Table | Key/Value, Columnbased – Tabular, Document-based, Graphbased data-stores. |
| Yêu cầu phần cứng cho dữ liệu lớn | Yêu cầu phần cứng máy chủ cao. | Không nhất thiết phần cứng phải cao, có thể mở rộng máy chủ ở nhiều nơi với tài nguyên phần cứng thấp. |
| Cách lấy dữ liệu từ client. | Thông qua chuỗi kết nối (tên máy chủ, tên cơ sở dữ liệu, tên tài khoản và mật khẩu). | Thông qua API (Application Programming Interface). |
| Hiệu xuất | Kém hơn SQL Relational giữa các bảng | Cực tốt Bỏ qua SQL Bỏ qua các ràng buộc dữ liệu |
| Khả năng mở rộng | Hạn chế về lượng | Hỗ trợ lượng lớn cho các node |
| Phù hợp với công nghệ điện toán đám mây | Không | Có. |
| Hiệu suất đọc ghi | Kém do thiết kế để đảm bảo sự vào/ra liên tục của dữ liệu | Tốt với mô hình xử lý lô và những tối ưu về đọcghi dữ liệu. |
| Thay đổi số node trong hệ thống | Phải shutdown cả hệ thống. Việc thay đổi số node phức tạp. | Không cần phải shutdown cả hệ thống. Việc thay đổi số node đơn giản, không ảnh hưởng đến hệ thống. |
| Hỗ trợ lập trình hướng đối tượng | Kém | Tốt |
| Tính tiện dụng về mặt hỗ trợ | Tốt | Còn chưa hoàn thiện |

Bảng 1.1 So sánh RDBMS và NOSQL

Các RDBMs hiện tại đã bộc lộ những yếu kém, chẳng hạn như việc đánh chỉ mục một lượng lớn dữ liệu, phân trang, hoặc phân phối luồng dữ liệu media (phim, ảnh, nhạc ...). Cơ sở dữ liệu quan hệ được thiết kế cho những mô hình dữ liệu nhỏ thường xuyên đọc viết trong khi các Social Network Services lại có một lượng dữ liệu cực lớn và cập nhật liên tục do số lượng người dùng quá nhiều ở một thời điểm.

NoSQL chủ yếu lựa chọn mô hình lưu trữ tập dữ liệu theo cặp giá trị key/value để dễ dàng trong việc truy xuất. Trong NoSQL có khái niệm gọi là node được sử dụng trong quản lý dữ liệu phân tán. Hệ thống dữ liệu phân tán không có sự kiểm tra có trùng dữ liệu hay không, nên dữ liệu phân tá sẽ chấp nhận việc lưu trữ trùng lặp dữ liệu. Khi có một yêu cầu truy vấn lấy dữ liêu có thể có nhiều máy cùng lúc gửi đến, nếu có một máy nào bị hỏng thì cũng không ảnh hưởng nhiều tới toàn bộ hệ thống. Thông thường người ta sẽ tách biệt database ra nhiều database để đảm bảo lượng truy cập xử lý lớn. Một database nhỏ đảm bảo được dữ liệu nhập/ xuất liên tục, nhưng dữ liệu đạt tới ngưỡng nào đó thì sẽ gộp vào một Database lớn hơn có thiết kế tối ưu. Mô hình đó cho phép tăng cường hiệu suất nhập/ xuất.

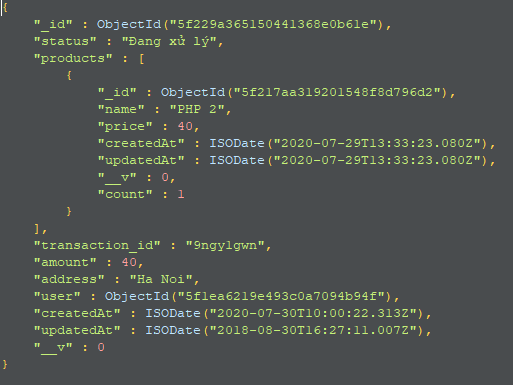
Thế hệ CSDL NoSQL - giảm thiểu tối đa các phép tính toán, tác vụ đọc ghi, đảm bảo được yêu cầu xử lý dữ liệu của các dịch vụ mạng xã hội. Hệ CSDL này có thể lưu trữ, xử lý từ lượng dữ liệu rất nhỏ đến hàng petabytes dữ liệu với khả năng chịu tải, chịu lỗi cao nhưng chỉ đòi hỏi về tài nguyên phần cứng thấp.

NoSQL sẽ là lựa chọn tốt nhất cho các công ty nhỏ không đủ kinh phí cho việc đầu tư về bản quyền hoặc server mạnh, cấu hình cao.

### Cơ sở dữ liệu MongoDB

#### Giới thiệu về MongoDB

MongoDB là một mã nguồn mở, dùng đê thiết kế cơ sở dữ liệu để dễ dàng phát triển và mở rộng, cung cấp hiệu suất và tính sẵn sàng cao. Cấu trúc của Mongodb bao gôm các cặp “thuộc tính” – “giá trị”, MongoDB document tương tự như đối tượng JSON. Giá trị của thuộc tính bao gồm các thuộc tính giá trị cụ thể ( số, chữ, …), document, mảng, mảng các đối tượng.



Hình 1.1 Cấu trúc của một Document trong MongoDB

Các tính năng:

* Hiệu suất cao: Hỗ trợ nhúng dữ liệu dạng mô hinh dữ liệu giúp giảm thiểu hoạt động của server, Truy vấn dữ liệu sử dụng chỉ mục giúp tối ưu tốc độ truy vấn.
* Dễ dàng tăng tính mở rộng: chế sharding tự động (tự động phân vùng dữ liệu trên máy chủ, động bộ hóa dữ liệu tốt

#### Các khái niệm trong MongoDB CRUD

Database là gì? Database là một Ô chứa dữ liệu ở mức vật lý (physical), mỗi database sẽ có nhiều collection và được thiết lập lưu trữ ở một nơi trong máy chủ máy tính. Một máy chủ MongoDB thường có thể tạo nhiều cơ sở dữ liệu.

Collection là gì? Collection trong MongoDB là nhóm các tài liệu (document), nó tương đương với một bảng trong cơ sở dữ liệu quan hệ nên mỗi collection sẽ thuộc về một database duy nhất. Tuy nhiên nó có một sực khác biệt đó là nó không có ràng buộc Relationship như các hệ quản trị cơ sở dữ liệu khác nên việc truy xuất rất nhanh, chính vì thế mỗi collection có thể chứa nhiều thể loại khác nhau không giống như table trong hệ quản trị mysql là các field cố định.

Document là gì? Document trong MongoDB có cấu trúc tương tự như kiểu dữ liệu JSON, nghĩa là sẽ có các cặp (key: value) nên nó có tính năng động rất lớn. Document ta có thể hiểu nó giống như các record dữ liệu trong MYSQL, tuy nhiên nó có sự khác biệt là các cặp (key: value) có thể không giống nhau ở mỗi document.

Để rõ hơn chúng ta sẽ so sánh MongoDB và một hệ quản trị CSDL.

|  |  |
| --- | --- |
| RDBMS | MongoDB |
| Database | Database |
| Table | Collection |
| Tuple/Row | Document |
| Column | Field |
| Table Join | Embedded Document |
| Primary key | Primary key (mặc định là \_id) |

Bảng 1.2 So sánh các khái niệm giữa RDBMS và MongoDB

## ExpressJS

### ExpressJs là gì

ExpressJs là một web framework nhỏ được viết bằng JavaScript và dựa trên nền tảng của NodeJs. Nó cung cấp các tính năng mạnh mẽ giúp xây dựng các ứng dụng web hoặc ứng dụng di động.

ExpressJS cung cấp rất nhiều các công cụ cần thiết để xây dựng ứng dụng. ExpressJS cũng linh hoạt vì có sẵn nhiều mô-đun khác nhau được tạo sẵn trên NPM và có thể khai báo trực tiếp khi muốn sử dụng.

Ưu điểm của ExpressJS:

* Dễ dàng cấu hình và tùy chỉnh.
* Cho phép xác định các Route của ứng dụng dựa trên các phương thức HTTP và URL.
* Dễ dàng tích hợp các template engine như Jade, EJS,…
* Cho phép tạo một server REST API
* Dễ dàng kết nối các cơ sở dữ liệu như MongoDB, MySQL

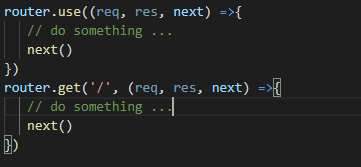
### Routing trong ExpressJS

#### Khái niệm

Routing trong express là một bộ định tuyến giúp cho định danh các request và so sánh với các route hiện có và thực hiện các hàm xử lý được liên kết với route đó và tạo ra các response thích hợp.

Một route bao gồm các HTTP method (GET, POST, PUT, …), một đường dẫn phù hợp với request URL và các hàm xử lý, Hàm xử lý được chuyên qua một request object và một response object.

Route hoạt động như một middleware nên chúng ta có thể dùng nó như một arguments. Hoặc dùng nó như một arguments cho route khác. Ví dụ



Hình 1.2 Code ví dụ về Route

#### So sánh Request

Khi một request được nhận, điều xay ra đầu tiên là so sánh request. Ví dụ:

*const app = express();*

*app.get('/hello', (req, res) => {*

*res.send('Hello World') });*

Trong ví dụ trên Request method khớp với route method (hàm get được gọi trên ứng dụng, cho biết nó chỉ nên khớp với các GET HTTP method) và cả URL yêu cầu với thông số đường dẫn '/hello', nếu một request khớp với Route này hàm xử lý sẽ in ra dòng chữ ‘Hello World’. Nói ngắn gọn khi truy cập tới URL/hello ví dụ: http://localhost:3000/hello thì sẽ thực hiện hàm này.

Thông thường bạn sẽ chỉ sử dụng một method cụ thể như app.get(), app.post(),… Nhưng nếu bạn muốn sử dụng tất cả method đó bạn có thể sử dụng app.all().

Các đường dẫn cũng có thể là các regular expression để tăng tính bảo mật, nhưng hiếm khi được sử dụng.

#### Route parameters

Thông số tuyến đường là các đoạn được đặt tên trong đặc tả đường dẫn khớp với một phần của URL. Nếu trùng khớp, giá trị trong đó của URL sẽ được cung cấp dưới dạng một biến trong request oject. Ví dụ:

app.get ('/ customer /: customerId', (req, res) =>{

// do something

})

URL/customer/1234 sẽ khớp với route ở trên và /customer/ 4567 cũng vậy. Trong cả hai trường hợp, ID khách hàng sẽ được ghi lại và cung cấp cho hàm xử lý như một phần của yêu cầu trong req.params, với tên của tham số là khóa. Do đó, req.params.customerId sẽ có giá trị 1234 hoặc 4567 tương ứng cho mỗi URL này

### Middleware trong ExpressJS

#### Khái niệm

ExpressJS khi hoạt động, về cơ bản sẽ là một loạt các hàm **Middleware** được thực hiện liên tiếp nhau. Sau khi đã thiết lập, các request từ phía người dùng khi gửi lên ExpressJS sẽ thực hiện lần lượt qua các hàm Middleware cho đến khi trả về response cho người dùng. Các hàm này sẽ được quyền truy cập đến các đối tượng đại diện cho Request - **req**, Response - **res**, hàm Middleware tiếp theo - **next**, và đối tượng lỗi - **err** nếu cần thiết.

Một hàm Middleware sau khi hoạt động xong, nếu chưa phải là cuối cùng trong chuỗi các hàm cần thực hiện, sẽ cần gọi lệnh next() để chuyển sang hàm tiếp theo, bằng không xử lý sẽ bị treo tại hàm đó.

Các chức năng mà middleware có thể thực hiện trong ExpressJS sẽ bao gồm :

* Thực hiện bất cứ đoạn code nào
* Thay đổi các đối tượng request và response
* Kết thúc một quá trình request-response
* Gọi hàm middleware tiếp theo trong stack

Trong Express, có 5 kiểu middleware có thể sử dụng :

* Application-level middleware (middleware cấp ứng dụng)
* Router-level middleware (middlware cấp điều hướng - router)
* Error-handling middleware (middleware xử lý lỗi)
* Built-in middleware (middleware sẵn có)
* Third-party middleware (middleware của bên thứ ba)

## ReactJS

### React là gì ?

React là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được. React được sử dụng tại Facebook trong các sản phẩm, và instagram được viết hoàn toàn trên React.

Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của lần render này với lần render trước và cập nhật ít thay đổi nhất trên DOM. Trươc khi đến cài đặt và cấu hình, chúng ta sẽ đi đến một số khái niệm cơ bản:

### JSX

JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. Đặc điểm: Faster: Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javacsript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript. Safer: an toàn hơn. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt. Easier: Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng

### Ưu nhược điểm của ReactJS

#### Ưu điểm

Virtual DOM, cập nhật toàn bộ DOM sẽ làm cho trang web hoạt động không hiệu quả, vì nó tốn nhiều tài nguyên. Với ReactJS, mỗi khi một thay đổi được kích hoạt (ví dụ: bởi một truy vấn hoặc hành động của người dùng), toàn bộ DOM ảo sẽ được cập nhật. ReactJS giữ hai phiên bản DOM ảo trong bộ nhớ - một DOM ảo được cập nhật và một bản sao được tạo trước khi cập nhật. Sau khi cập nhật, ReactJS so sánh hai phiên bản này để tìm ra các yếu tố đã thay đổi. Sau đó, nó chỉ cập nhật phần DOM thực đã thay đổi. Quá trình này có vẻ phức tạp và tốn thời gian. Tuy nhiên, nó mất ít thời gian hơn nhiều so với việc cập nhật toàn bộ DOM thực và tối ưu hóa thao tác DOM.

React hỗ trợ các Component có thể tái sử dụng. Sau khi một component được tạo nó có thể tái sử dụng được trong các phần khác nhau của project hoặc thậm chí trong các project khác nhau với ít hoặc không có thay đổi.

The one-way data flow (Luồng dữ liệu một chiều) trong ReactJS là một tính năng mà các nhà phát triển cho là lợi thế. Luồng dữ liệu này còn được gọi là từ trên xuống dưới hoặc từ parent đến child. Điều đó có nghĩa là chỉ có một cách dữ liệu được truyền giữa các phần tử trong ứng dụng ReactJS. Luồng dữ liệu đi xuống ngăn ngừa lỗi và tạo điều kiện debugging.

Facebook đã đầu tư rất nhiều nỗ lực vào việc biến ReactJS trở thành một công cụ mạnh mẽ có thể cải thiện giao diện người dùng của các ứng dụng web của họ và họ đang tiếp tục làm cho ReactJS trở nên hiệu quả và dễ sử dụng hơn. Nhưng ReactJS không chỉ được tạo ra bởi Facebook. Bằng cách biến nó thành mã nguồn mở vào năm 2013, Facebook đã khuyến khích các nhà phát triển sử dụng và cải thiện nó. ReactJS ngày nay là kết quả của những nỗ lực chung. Cộng đồng ReactJS cung cấp rất nhiều thư viện có giá trị hỗ trợ quá trình phát triển ReactJS. Các thư viện ReactJS cung cấp cho các nhà phát triển những khả năng và cách thức mới để cải thiện chất lượng mã và ứng dụng của họ.

#### Nhược điểm

ReactJS là một công nghệ tương đối mới và nó tiếp tục phát triển với sự trợ giúp của các nhà phát triển đóng góp vào nó và mở rộng khả năng của nó.  Nhưng tài liệu cho ReactJS và các thư viện liên quan của nó thường thiếu. Các nhà phát triển có xu hướng phát hành bản cập nhật cho thư viện mà không cập nhật tài liệu. Điều này khiến những người khác khó làm quen với các tính năng mới và tích hợp thư viện.

ReactJS sử dụng cú pháp JSX để tạo các đối tượng JavaScript với HTML. Điều này đơn giản hóa quá trình sửa đổi DOM và làm cho mã dễ đọc hơn. Việc có cú pháp HTML trong mã JavaScript có vẻ khó hiểu đối với các nhà phát triển mới, vì vậy họ sẽ cần thời gian để làm quen với nó. Các nhà phát triển ít kinh nghiệm thường thấy JSX là một nhược điểm của ReactJS.

## NodeJS

### Giới thiệu về NodeJS

Node.js là một môi trường thực thi JavaScript bên ngoài trình duyệt, được xây dựng dựa trên Chrome’s V8 JavaScript engine. Chrome’s V8 JavaScript engine được viết bằng C++, giúp biên dịch code JavaScript thành mã máy một cách nhanh chóng, nên cho tốc độ xử lý và hiệu năng khá cao.

Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.

Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

Node.js chứa một thư viện có sẵn được lưu trữ trong NPM cho phép các ứng dụng hoạt động như một Webserver mà không cần phần mềm như Nginx, Apache HTTP Server hoặc IIS.

Node.js cung cấp kiến trúc hướng sự kiện (event-driven) và non-blocking I/O API, tối ưu hóa thông lượng của ứng dụng và có khả năng mở rộng cao

Các hàm trong Node.js là các hàm không đồng bộ (asynchronous). Do đó, các tác vụ đều được xử lý và thực thi ở chế độ nền.

### Ứng dụng của NodeJS

Nodejs thường được sử dụng để làm:

* Hệ thống Notification (Giống như facebook hayTwitter)
* Websocket server: Các máy chủ web socket như là Online Chat, Game Server…
* Fast File Upload Client: là các chương trình upload file tốc độ cao.
* Ad Server: Các máy chủ quảng cáo.
* Cloud Services: Các dịch vụ đám mây.
* RESTful API: đây là những ứng dụng mà được sử dụng cho các ứng dụng khác thông qua API.
* Any Real-time Data Application: bất kỳ một ứng dụng nào có yêu cầu về tốc độ thời gian thực.
* Ứng dụng Single Page Application (SPA): bởi những ứng dụng này thường request rất nhiều đến server thông qua AJAX
* Ứng dụng truy vấn tới NoSQL database như MongoDB, CouchDB,…
* Ứng dụng CLI: đây là các công cụ sử dụng command-line.

### Ưu nhược điểm của Nodejs

#### Ưu điểm

Node js là môi trường thời gian chạy mã nguồn mở, vì vậy nó được sử dụng miễn phí theo giấy phép MIT. Node js là nền tảng đa nền tảng chạy trên windows, mac, Linux. Ngoài ra còn có nhiều mô-đun mã nguồn mở được hỗ trợ bởi Node.js.

Node js sử dụng động cơ V8 do Google phát triển cho chrome. V8 biên dịch JavaScript thành mã máy gốc trực tiếp dẫn đến tốc độ nhanh hơn và thực thi mã hiệu quả.

Node sử dụng lập trình bất đồng bộ. Bạn có thể thực hiện nhiều thao tác cùng một lúc. Người dùng có thể dễ dàng gửi nhiều thư, thao tác cơ sở dữ liệu và đọc tệp đồng thời.

Cộng đồng Nodejs là một mạng lưới bao gồm rất nhiều các nhà phát triển, những người cùng nhau thảo luận, học hỏi và đóng góp cho các dự án khác nhau. Và có một kho lưu trữ lớn trên GitHub.

Nodejs hỗ trợ phát triển RESTful Web Services API rất dễ dàng.

#### Nhược điểm

Một trong những nhược điểm lớn của Node.JS là nó không có tính nhất quán. API thay đổi thường xuyên và điều này làm tăng rắc rối cho các nhà phát triển, vì khi điều này xảy ra, họ sẽ cần thực hiện các thay đổi trong cơ sở mã hiện có để đảm bảo tính nhất quán.

Các nhà phát triển cần phải viết mọi thứ từ đầu và điều này khá tốn thời gian, do đó, để làm cho dự án của bạn không bị trục trặc, bạn cần thuê nhà phát triển Node.JS có kinh nghiệm để phát triển và bảo trì.

Các nhà phát triển chỉ ra rằng việc sử dụng Node.js với cơ sở dữ liệu quan hệ không phải là một nhiệm vụ dễ dàng. Các công cụ này vẫn còn rất kém phát triển so với các đối thủ cạnh tranh trên thị trường. Vì vậy, nếu hoạt động của bạn bao gồm cơ sở dữ liệu quan hệ, bạn có thể phải sử dụng các tùy chọn khác như Rails, AS.NET MVC hoặc Django.

Là một mô hình lập trình không đồng bộ, bạn có thể tạo ra các ứng dụng có quy mô thực sự tốt. Tuy nhiên, nhiều nhà phát triển cảm thấy mô hình lập trình khó sử dụng, đặc biệt là các nhà phát triển đã quen thuộc với lập trình Non-blocking I/O

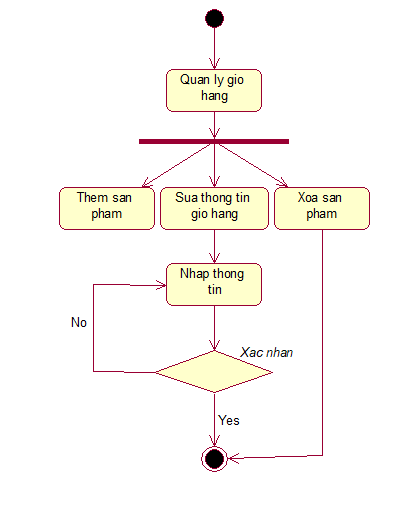
#### Kết luận

Thực tế là Node.js có nhiều ưu điểm hơn là nhược điểm, và điều đáng chú ý về nhược điểm là chúng đều có vẻ tương đối có thể sửa được. Bởi vì Node.js sử dụng JavaScript, điều này làm cho nó trở thành một hệ thống tốt để phát triển back-end vì các lập trình viên đã quen thuộc và cảm thấy thoải mái với nó. Hơn nữa, ngày càng có nhiều doanh nghiệp sử dụng Node.js và điều đó tự nó làm cho nó trở thành một hệ thống tuyệt vời cho các lập trình viên back-end sử dụng.

# PHÂN TÍCH HỆ THỐNG

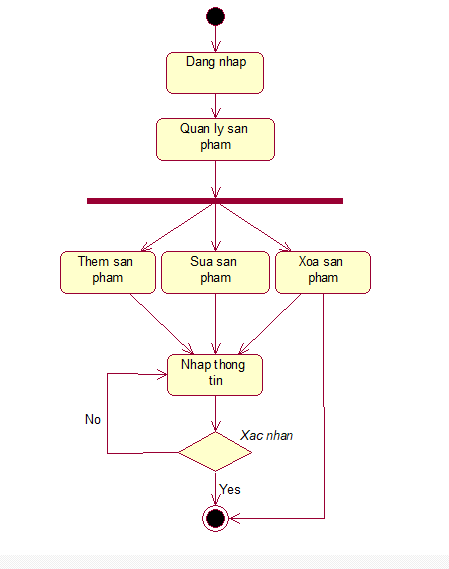
## Biểu đồ hoạt động

### Biểu đồ hoạt động chức năng thêm/xóa/sửa giỏ hàng



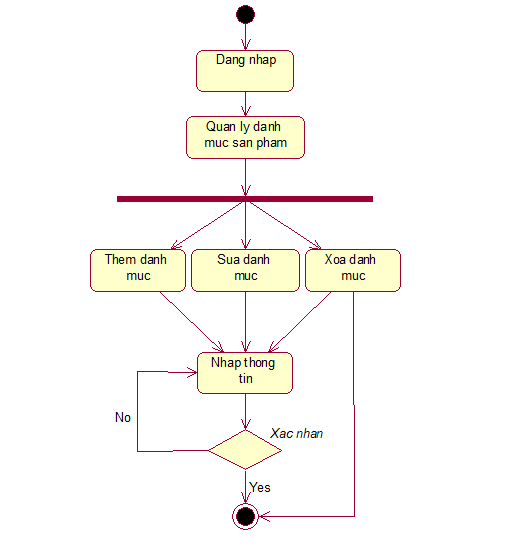
Hình 2.1 Biểu đồ hoạt động chức năng thêm/xóa/sửa giỏ hàng

### Biểu đồ hoạt động chức năng thêm/xóa/sửa sản phẩm



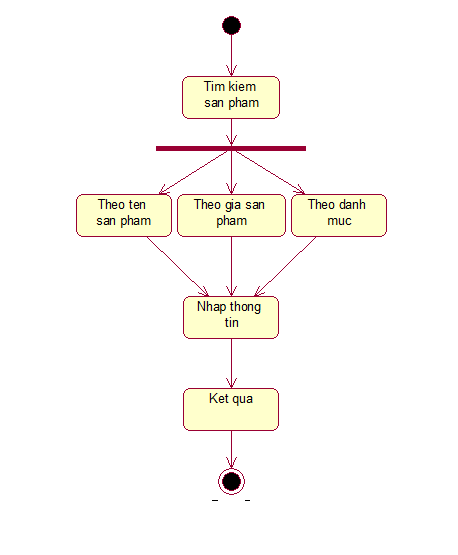
Hình 2.2 Biểu đồ hoạt động chức năng thêm/xóa/sửa sản phẩm

### Biểu đồ hoạt động chức năng thêm/xóa/sửa danh mục sản phẩm



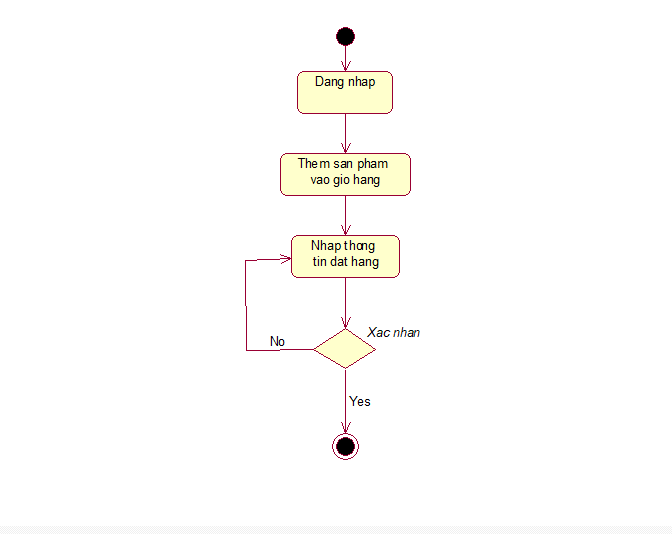
Hình 2.3 Biểu đồ hoạt động chức năng thêm/xóa/sửa danh mục

### Biểu đồ hoạt động chức năng tìm kiếm



Hình 2.4 Biểu đồ hoạt động chức năng tìm kiếm

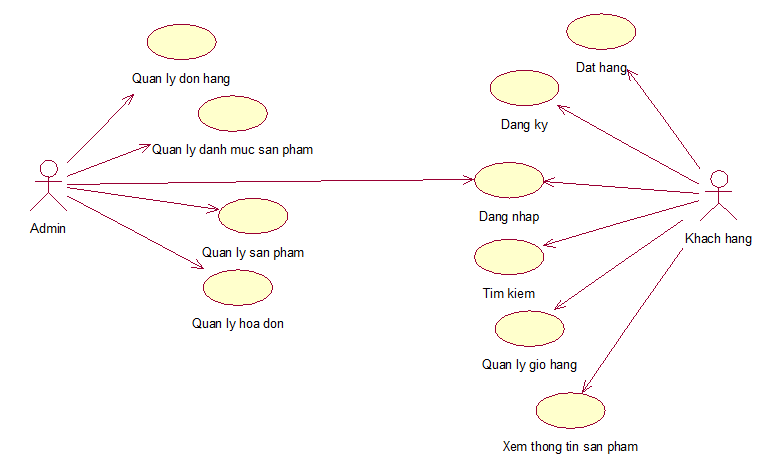
### Biểu đồ hoạt động chức năng đặt hàng



Hình 2.5 Biểu đồ hoạt động chức năng đặt hàng

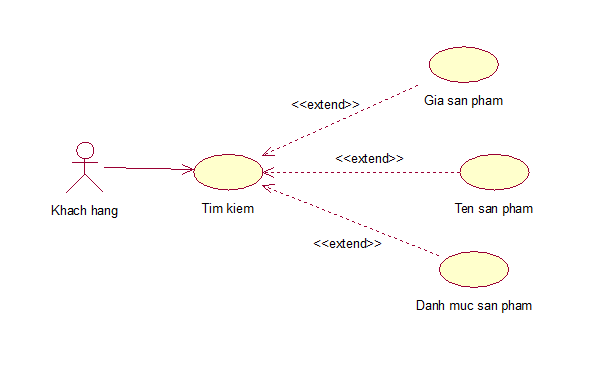
## Biểu đồ ca sử dụng

### Biểu đồ ca sử dụng tổng quát



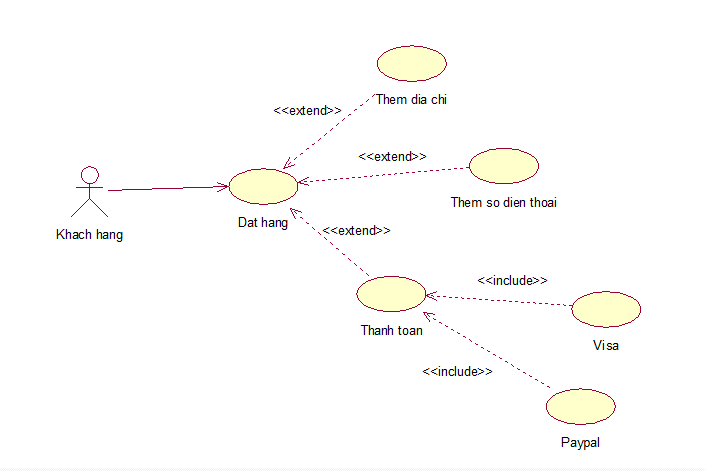
Hình 2.6 Biểu đồ ca sử dụng tổng quát

### Biểu đồ ca sử dụng khách hàng tìm kiếm sản phẩm



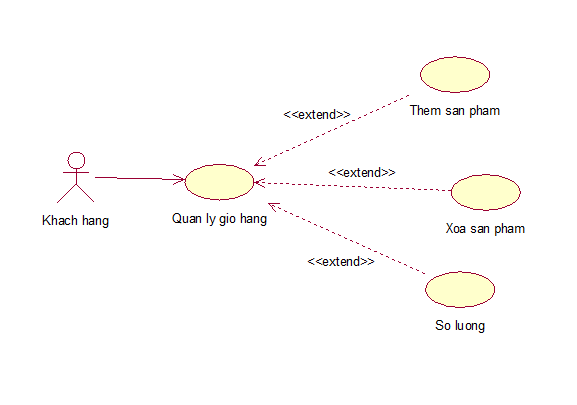
Hình 2.7 Biểu đồ ca sử dụng chức năng tìm kiếm sản phẩm

### Biểu đồ ca sử dụng khách hàng thanh toán



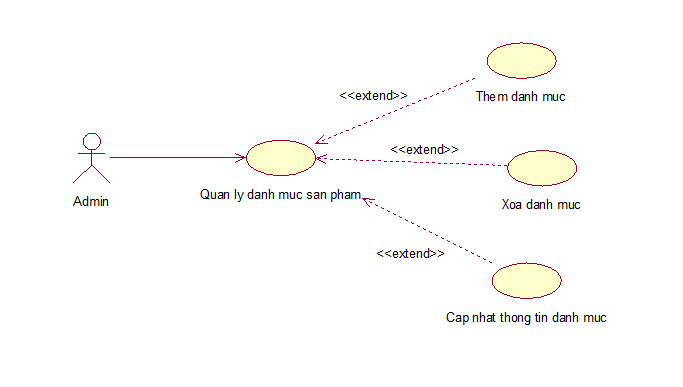
Hình 2.8 Biểu đồ ca sử dụng khách hàng thanh toán

### Biểu đồ ca sử dụng thêm sản phẩm vào giỏ hàng



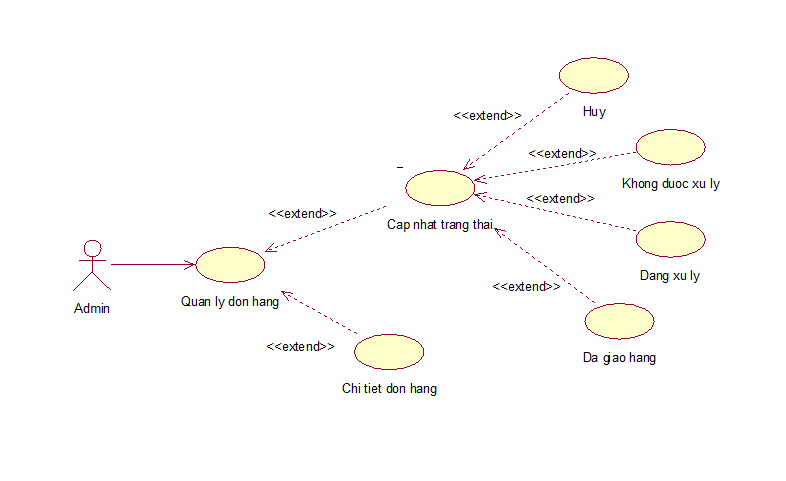
Hình 2.9 Biểu đồ ca sử dụng thêm sản phẩm vào giỏ hàng

### Biểu đồ ca sử dụng quản lý danh mục sản phẩm



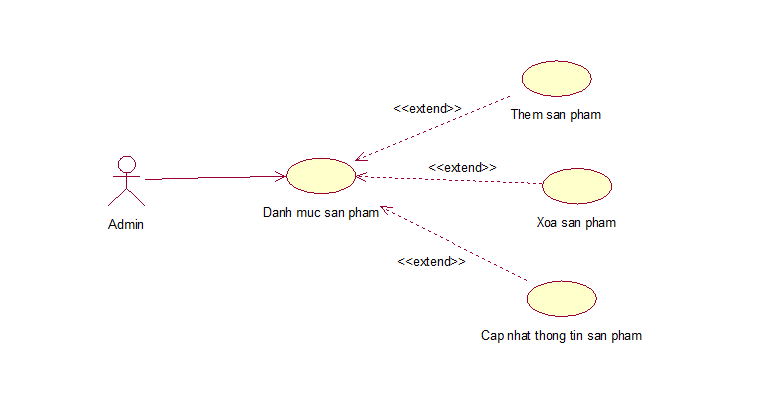
Hình 2.10 Biểu đồ ca sử dụng quản lý danh mục sản phẩm

### Biểu đồ ca sử dụng quản lý đơn hàng.



Hình 2.11 Biểu đồ ca sử dụng quản lý đơn hàng

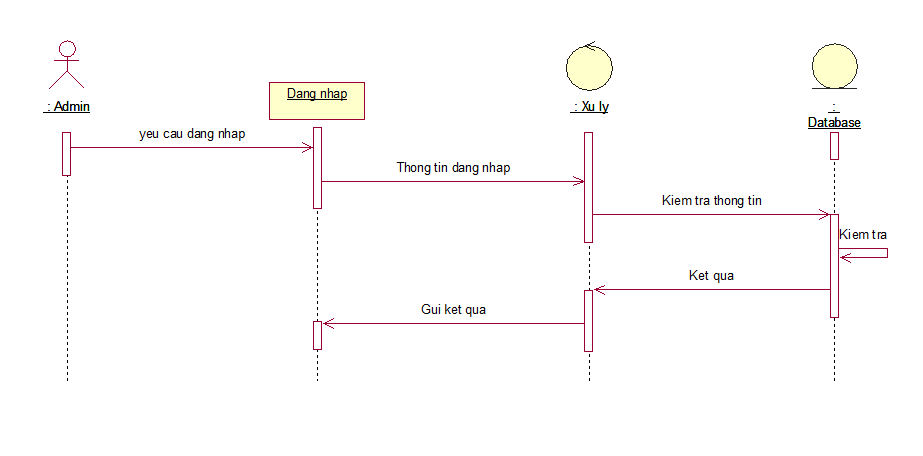
### Biểu đồ ca sử dụng quản lý sản phẩm



Hình 2.12 Biểu đồ ca sử dụng quản lý sản phẩm

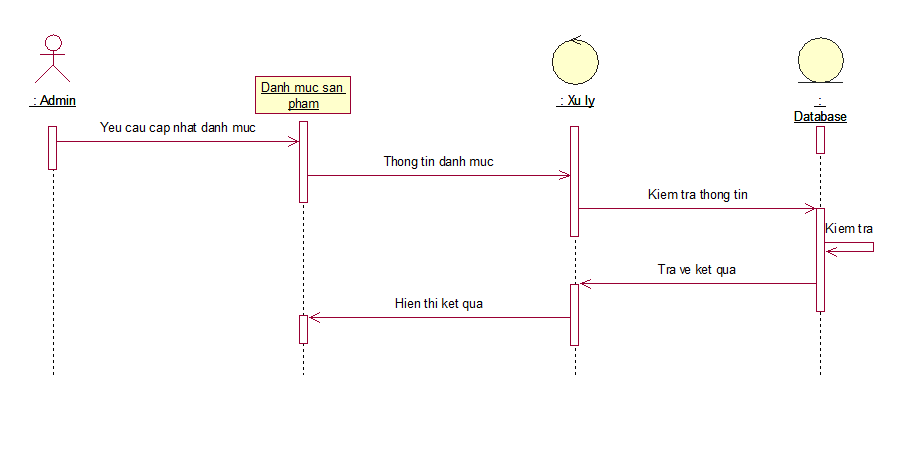
## Biểu đồ tuần tự.

### Biểu đồ tuần tự đăng nhập



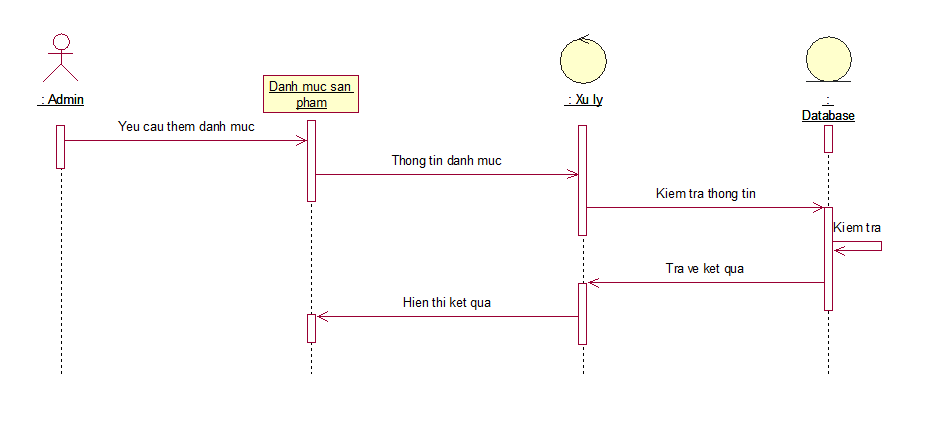
Hình 2.13 Biểu đồ tuần tự đăng nhập

### Biểu đồ tuần tự cập nhật danh mục sản phẩm



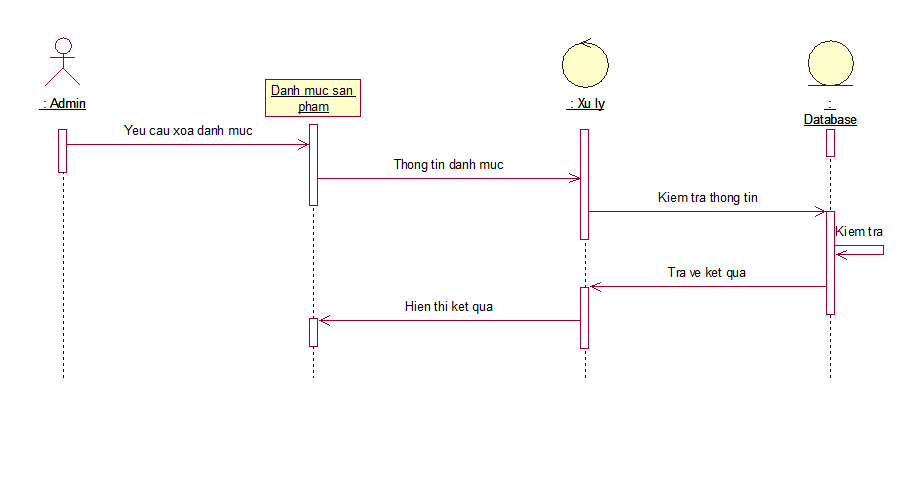
Hình 2.14 Biểu đồ tuần tự cập nhật danh mục sản phẩm

### Biểu đồ tuần tự thêm danh mục sản phẩm



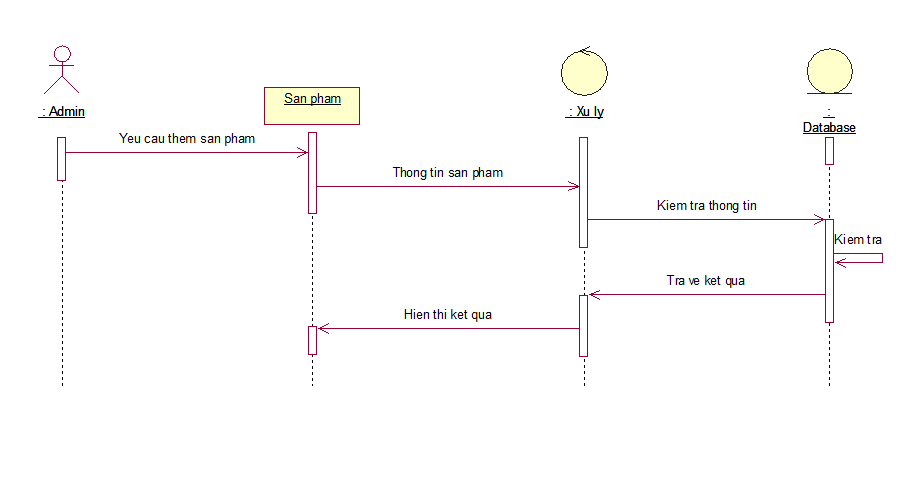
Hình 2.15 Biểu đồ tuần tự thêm danh mục sản phẩm

### Biểu đồ tuần tự xóa danh mục sản phẩm



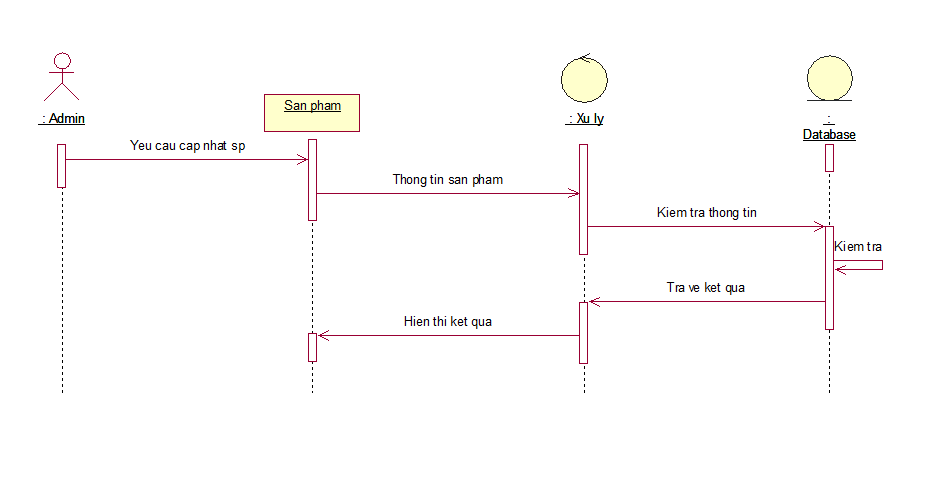
Hình 2.16 Biểu đồ tuần tự xóa danh mục sản phẩm

### Biểu đồ tuần tự thêm sản phẩm



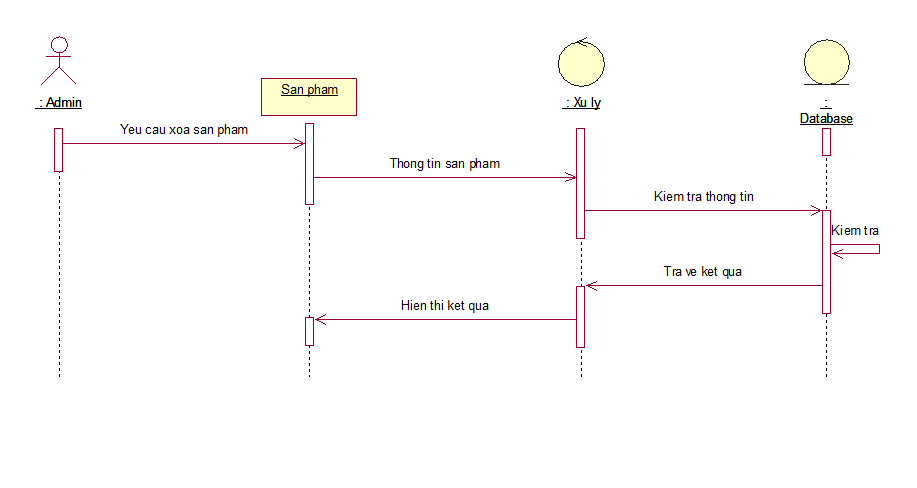
Hình 2.17 Biểu đồ tuần tự thêm sản phẩm

### Biểu đồ tuần tự cập nhật sản phẩm



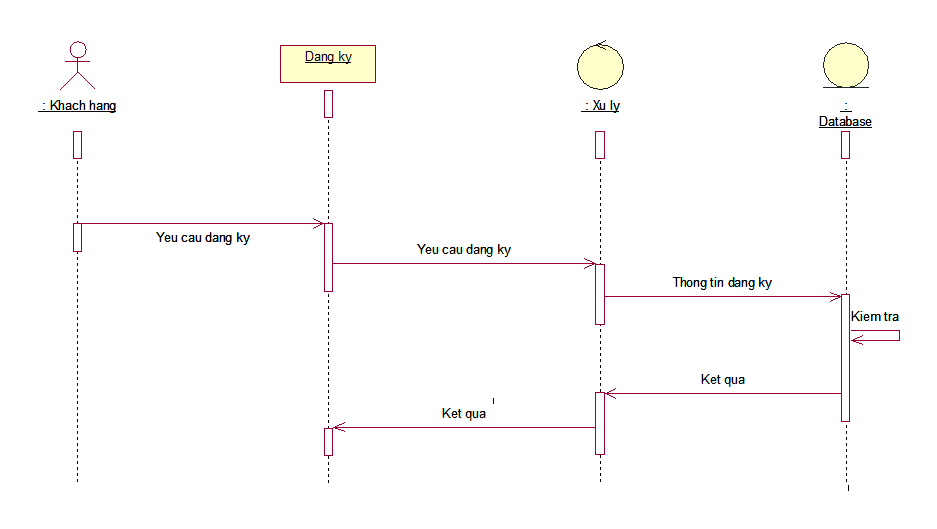
Hình 2.18 Biểu đồ tuần tự cập nhật sản phẩm

### Biểu đồ tuần tự xóa sản phẩm



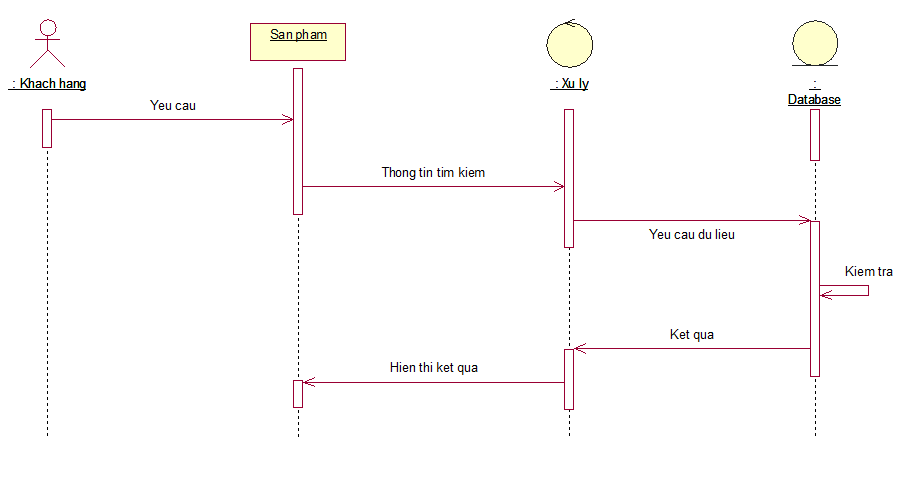
Hình 2.19 Biểu đồ tuần tự xóa sản phẩm

### Biểu đồ tuần tự đăng ký



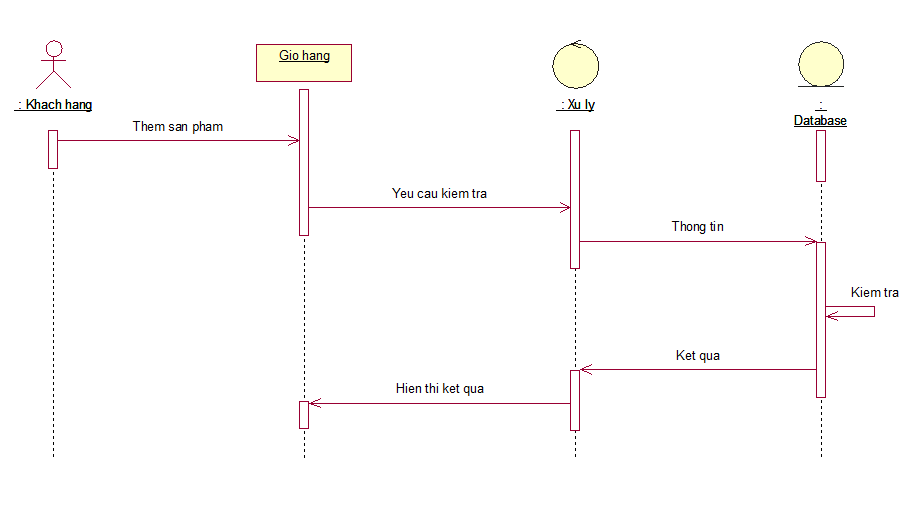
Hình 2.20 Biểu đồ tuần tự đăng ký

### Biểu đồ tuần tự tìm kiếm sản phẩm



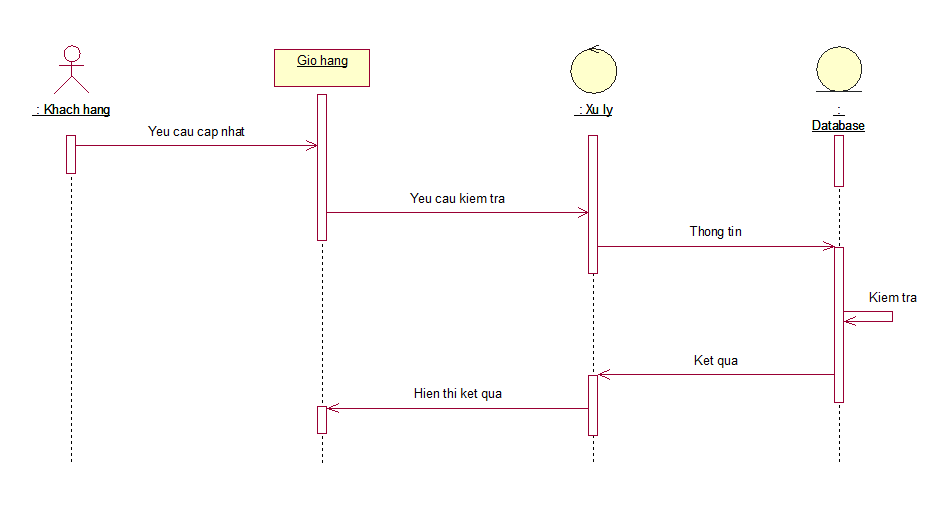
Hình 2.21 Biểu đồ tuần tự tìm kiếm sản phẩm

### Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng



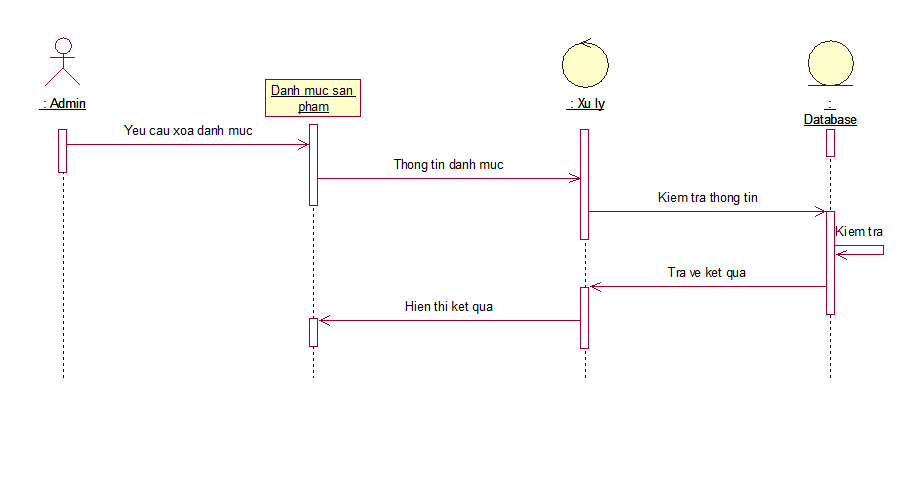
Hình 2.22 Biểu đồ tuần tự thêm sản phẩm vào giỏ hàng

### Biểu đồ tuần tự cập nhật sản phẩm vào giỏ hàng



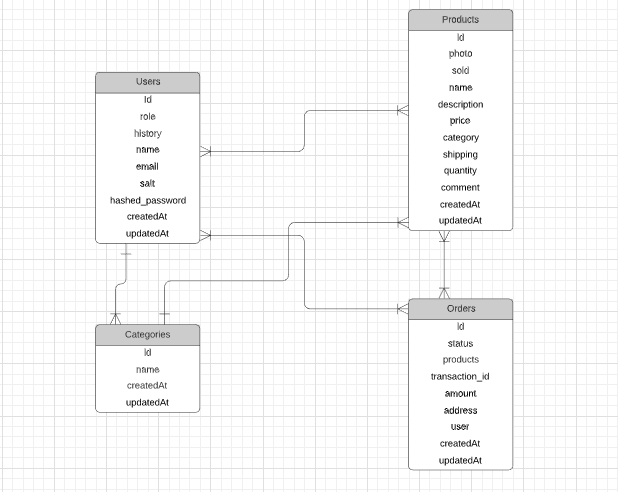
Hình 2.23 Biểu đồ tuần tự cập nhật vào giỏ hàng

### Biểu đồ tuần tự xóa sản phẩm khỏi giỏ hàng



Hình 2.24 Biểu đồ tuần tự xóa sản phẩm khỏi giỏ hàng

## Mô hình thực thể liên kết ERD

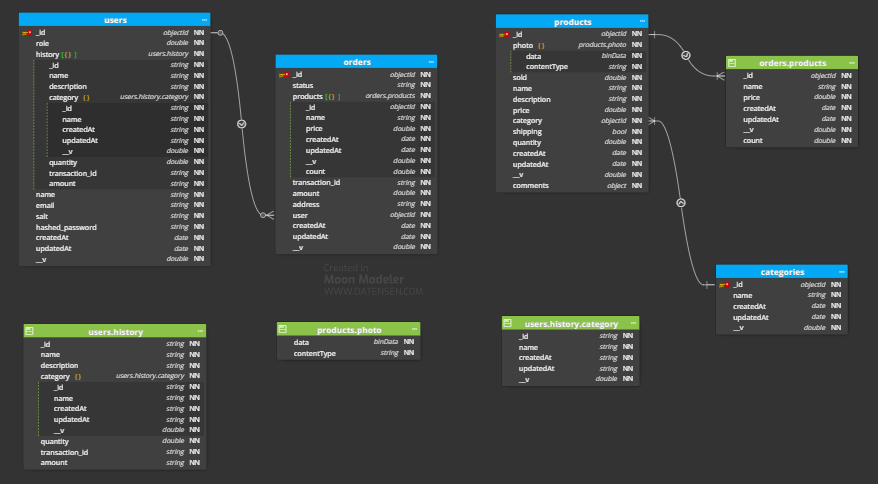


Hình 2.25 Mô hình thực thể liên kết

# THIẾT KẾ HỆ THỐNG

## Thiết kế cơ sở dữ liệu

### Thiết kế cơ sở dữ liệu mức logic



Hình 3.1 Quan hệ giữa các document

### Thiết kế cơ sở dữ liệu mức vật lý

#### Bảng User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Rằng buộc** | **Mô tả** |
| **1** | \_id | String | Primary key | Mã tài khoản |
| **2** | role | String |  | Vai trò trong hệ thống |
| **3** | history | Array |  | Lịch sử mua hàng |
| **4** | name | String |  | Tên người dùng |
| **5** | email | String |  | Email đăng nhập |
| **6** | salt | String |  | Đoạn mã được thêm cuối mật khẩu |
| **7** | hashed\_password | String |  | Mật khẩu đã được băm |
| **8** | createdAt | Date |  | Thời gian tạo tài khoản |
| **9** | updatedAt | Date |  | Thời gian sử thông tin tài khoản |

Bảng 3.1 Bảng user

#### Bảng Category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Rằng buộc** | **Mô tả** |
| **1** | \_id | String | Primary key | Mã danh mục |
| **2** | name | String |  | Tên danh mục |
| **3** | createdAt | Date |  | Thời gian tạo danh mục |
| **4** | updatedAt | Date |  | Thời gian sử thông tin danh mục |

Bảng 3.2 Category

#### Bảng Product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Rằng buộc** | **Mô tả** |
| **1** | \_id | String | Primary key | Mã sản phẩm |
| **2** | description | String |  | Mô tả sản phẩm |
| **3** | price | Number |  | Giá sản phẩm |
| **4** | category | String |  | Danh mục |
| **5** | quantity | Number |  | Số lượng |
| **6** | sold | Number |  | Số lượng sản phẩm đã bán |
| **7** | photo | Buffer |  | Hình ảnh sản phẩm |
| **8** | shipping | String |  | Hàng đã về kho chưa |
| **9** | comment | String |  | Bình luận |
| **10** | createdAt | Date |  | Thời gian tạo sản phẩm |
| **11** | updatedAt | Date |  | Thời gian sử thông tin tài khoản |

Bảng 3.3 Bảng product

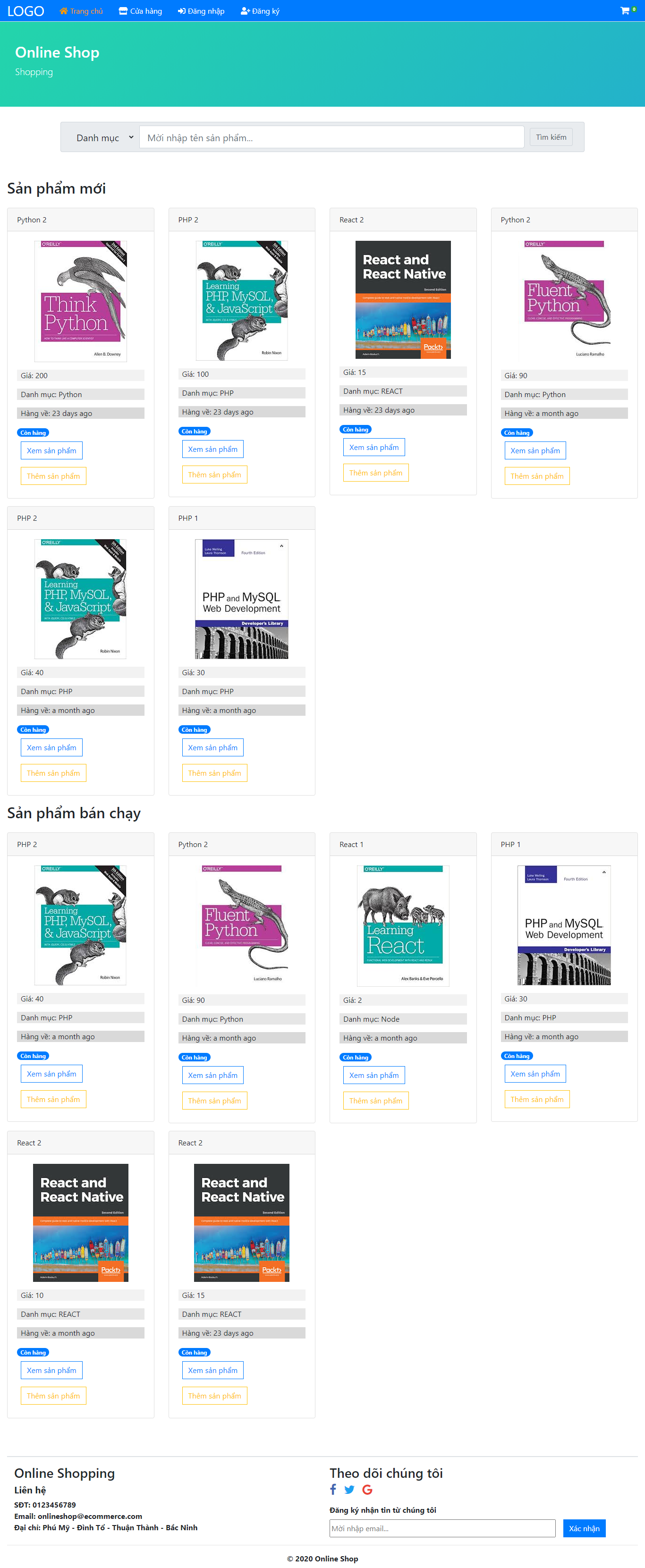
#### Bảng Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Rằng buộc** | **Mô tả** |
| **1** | \_id | String | Primary key | Mã đơn hàng |
| **2** | status | String |  | Trạng thái đơn hàng |
| **3** | products | Array |  | Sản phẩm |
| **4** | transaction\_id | String |  | Mã giao dịch |
| **5** | amount | Number |  | Tổng tiền |
| **6** | address | Number |  | Địa chỉ |
| **7** | user | Buffer |  | Tên khách hàng |
| **8** | phone | String |  | Số điện thoai |
| **9** | createdAt | Date |  | Thời gian đặt hàng |

Bảng 3.4 Bảng order

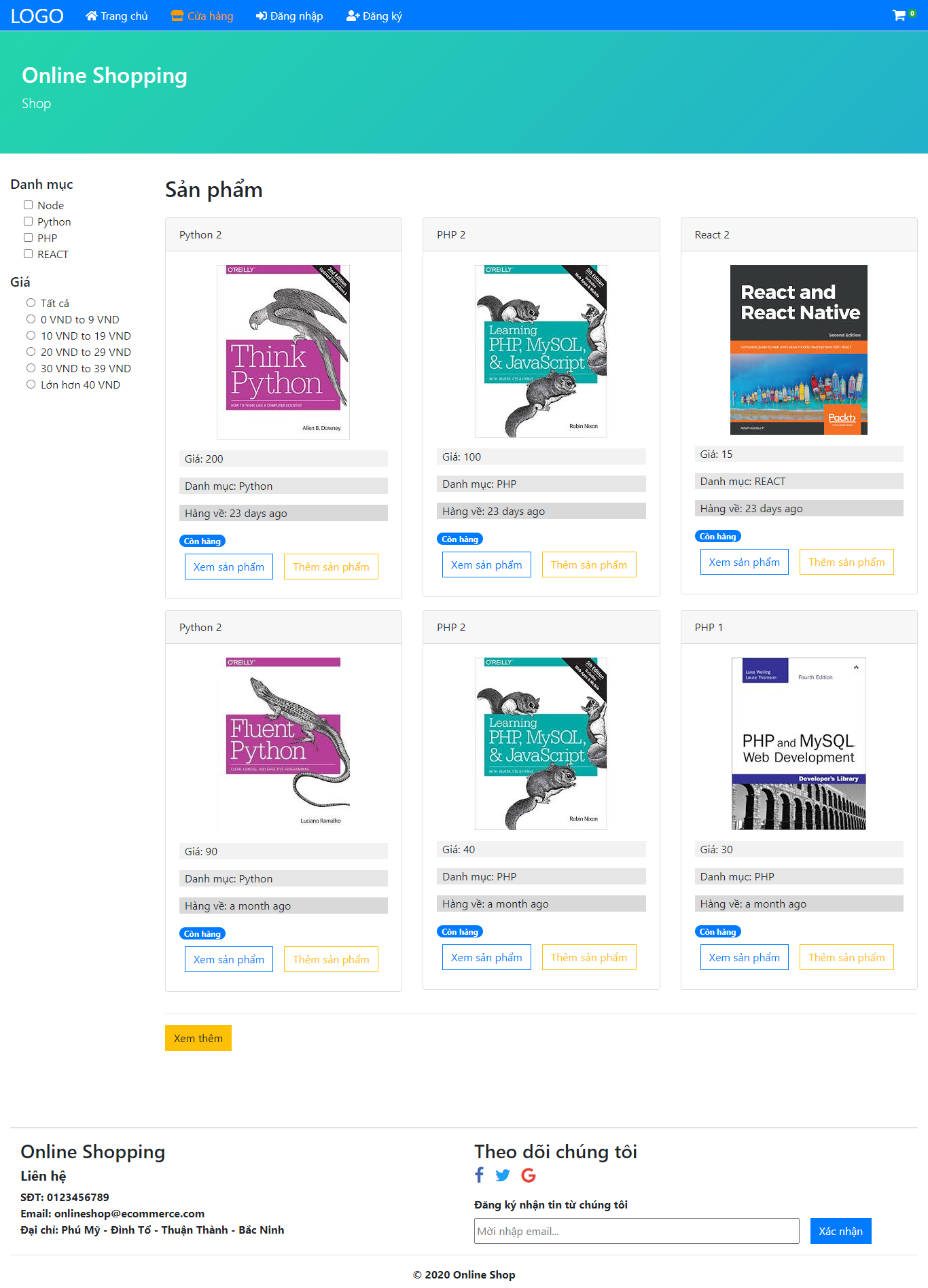
## Thiết kế giao diện

### Giao diện trang chủ



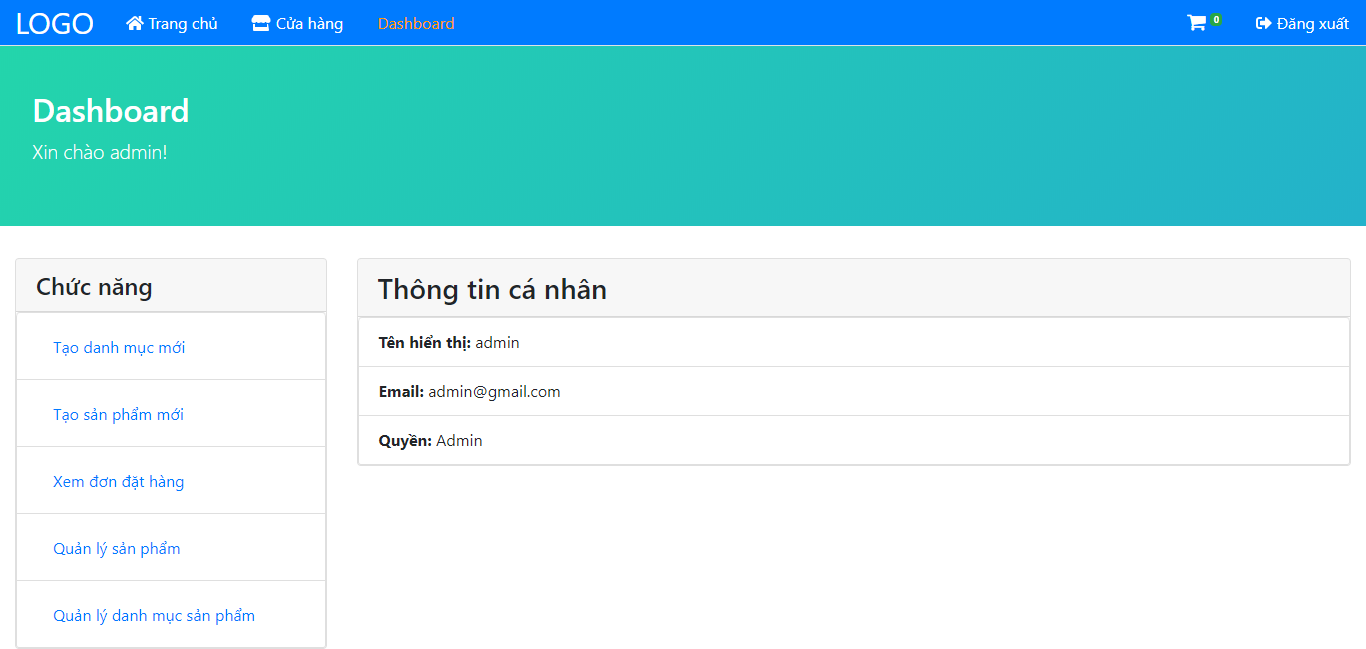
Hình 3.2 Giao diện trang chủ

### Giao diện trang sản phẩm



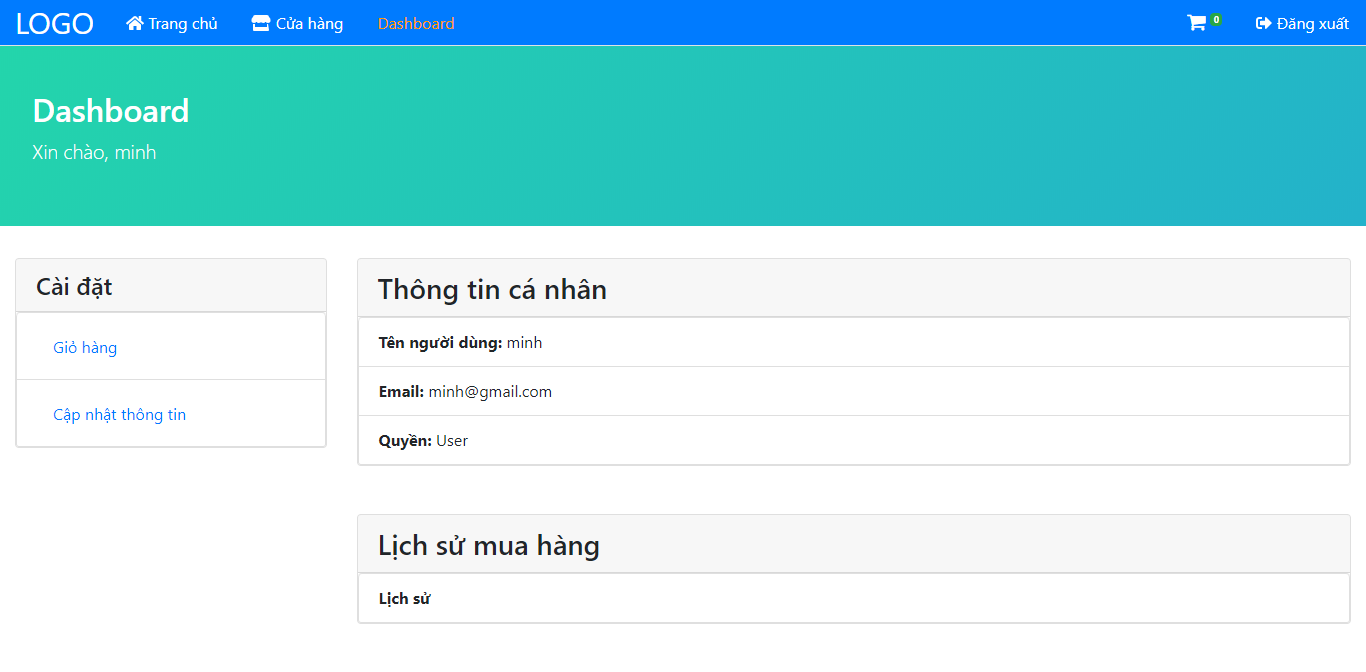
Hình 3.3 Giao diện trang sản phẩm

### Giao diện trang admin dashboard



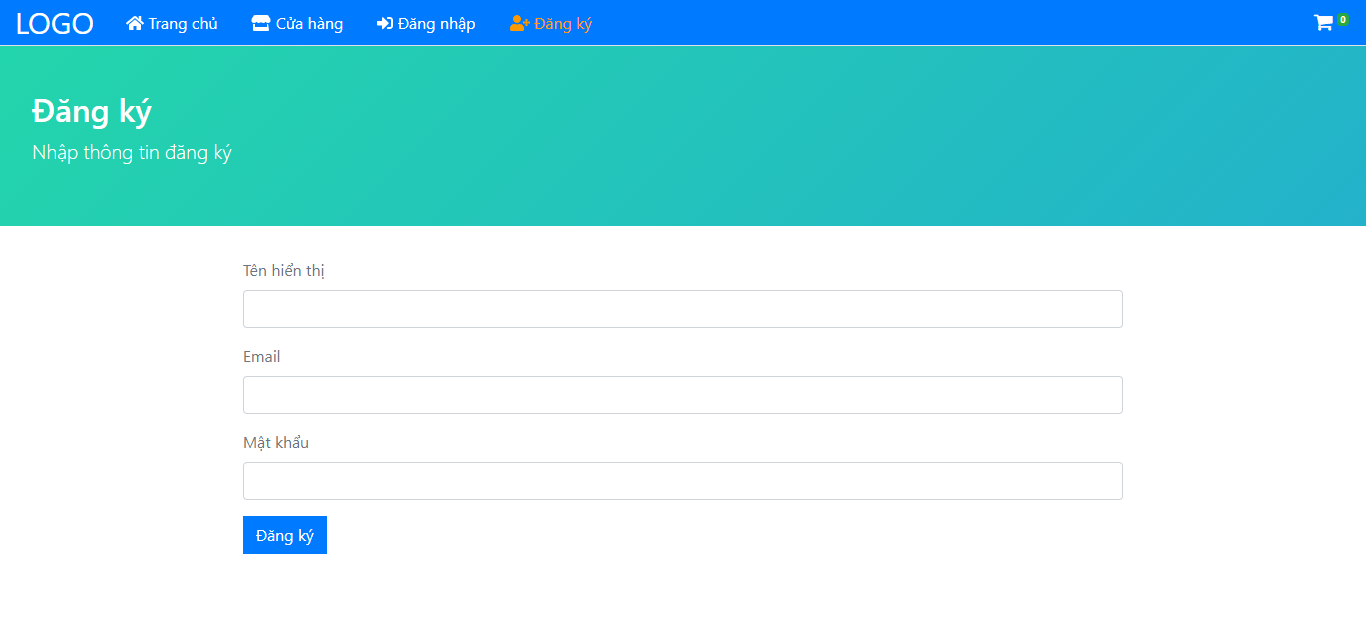
Hình 3.4 Giao diện trang admin dashboard

### Giao diện trang user dashboard



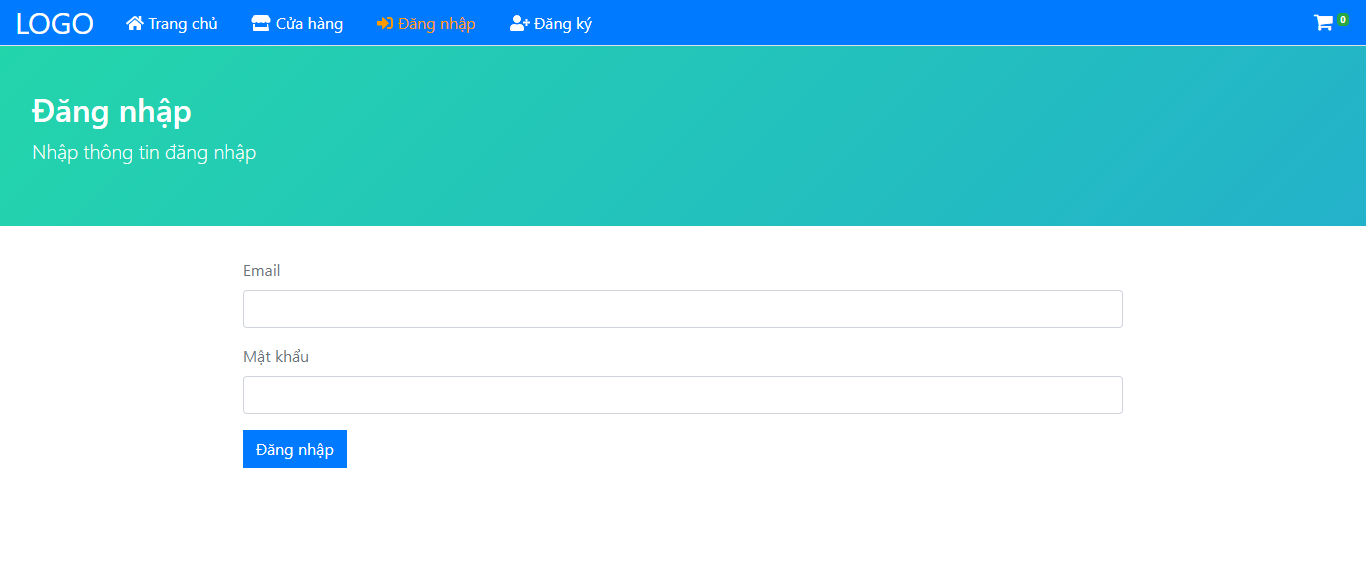
Hình 3.5 Giao diện trang user dashboard

### Giao diện trang đăng ký



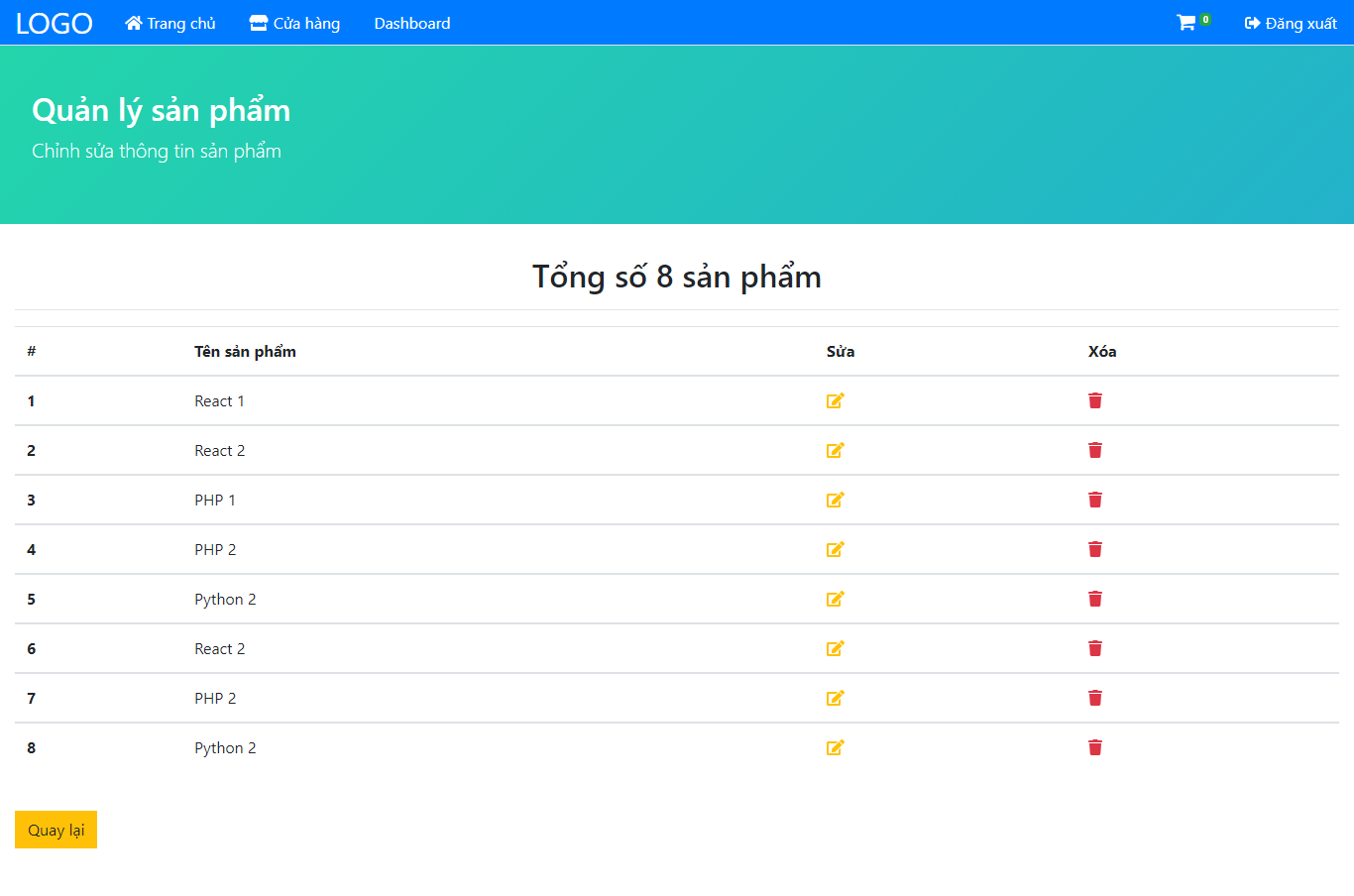
Hình 3.6 Giao diện trang đăng ký

### Giao diện trang đăng nhập



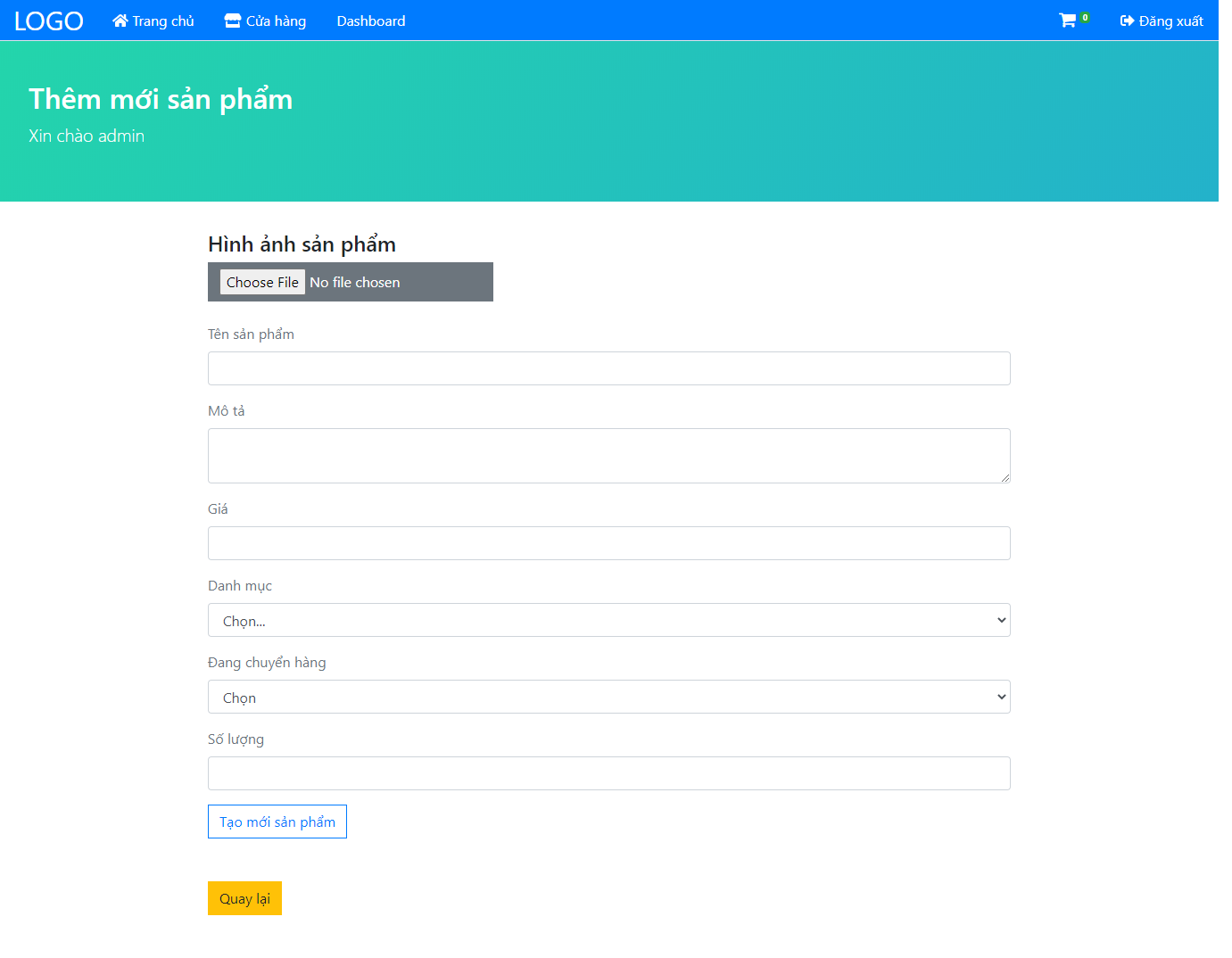
Hình 3.7 Giao diện trang đăng nhập

### Giao diện trang quản lý sản phẩm



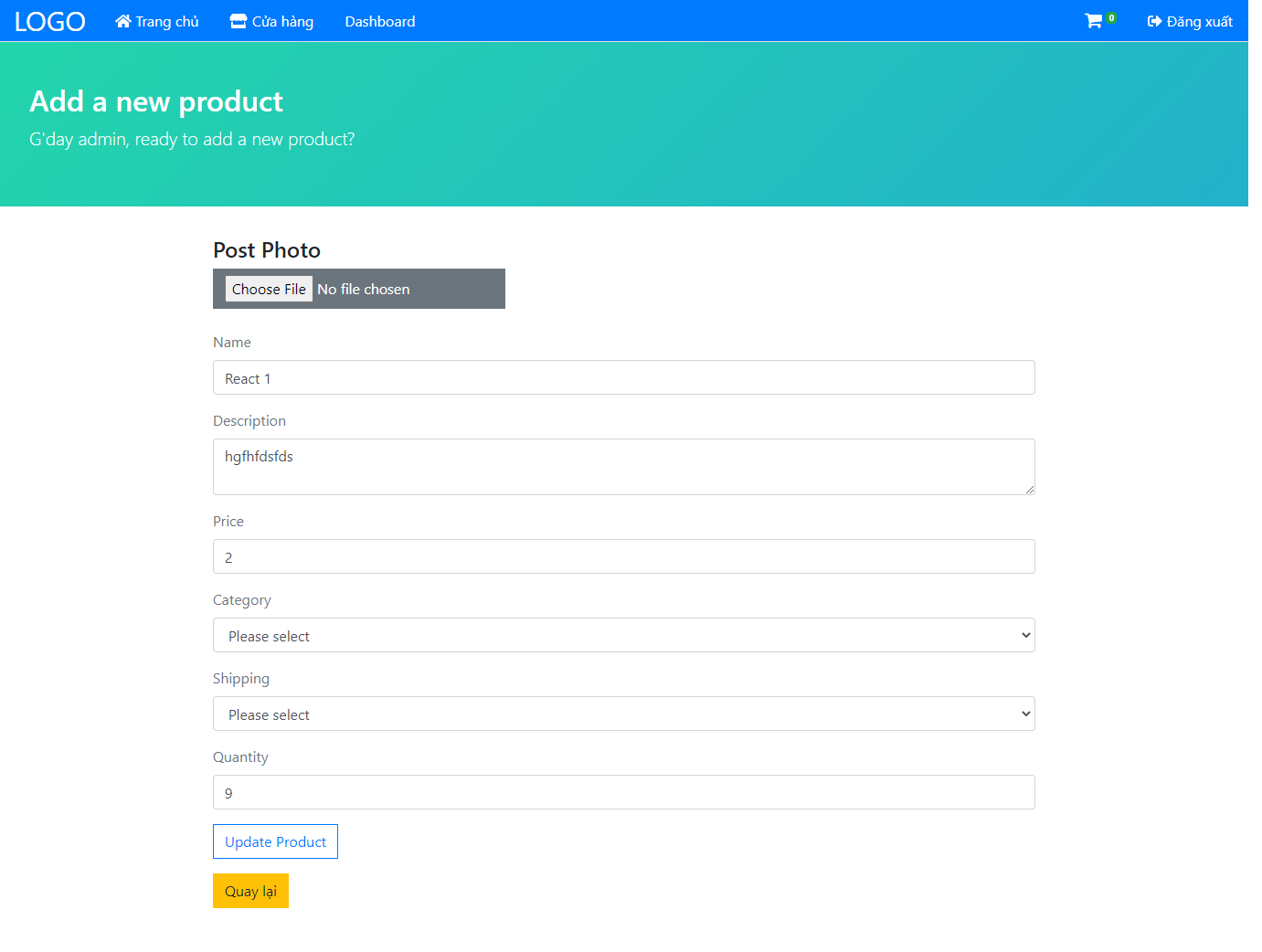
Hình 3.8 Giao diện trang quản lý sản phẩm

### Giao diện trang thêm sản phẩm



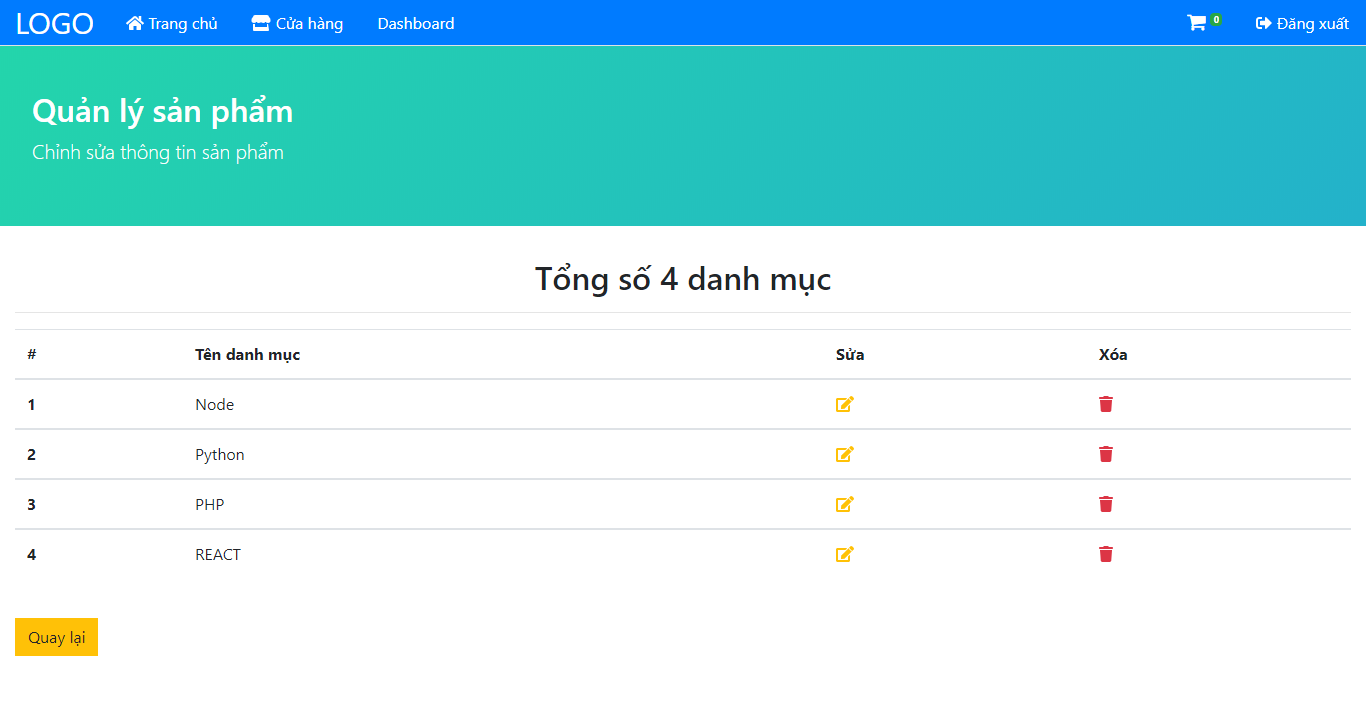
Hình 3.9 Giao diện trang thêm sản phẩm

### Giao diện trang sửa thông tin sản phẩm



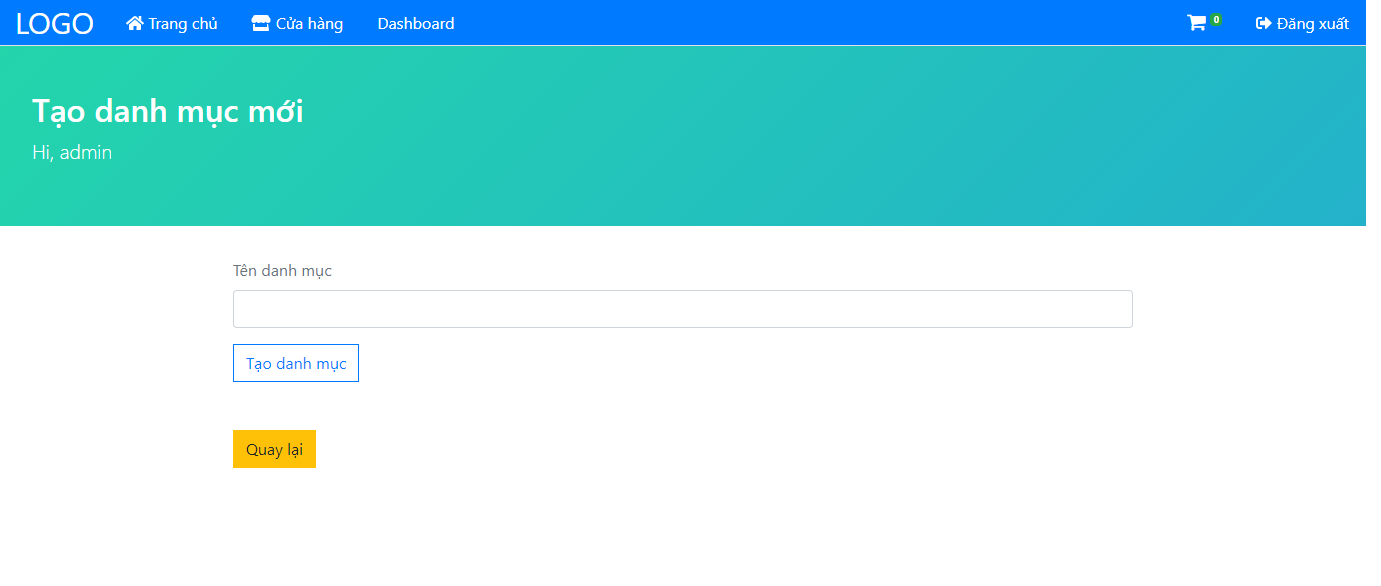
Hình 3.10 Giao diện trang sửa thông tin sản phẩm

### Giao diện trang quản lý danh mục sản phẩm



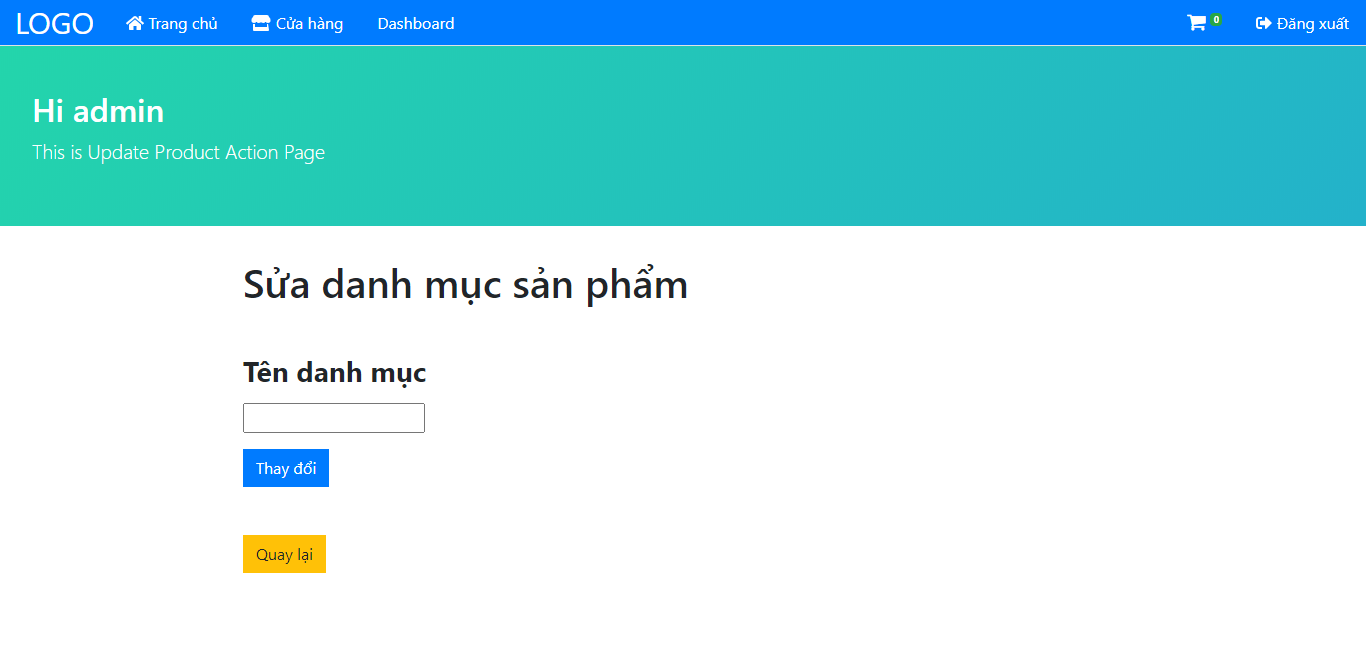
Hình 3.11 Giao diện trang quản lý danh mục sản phẩm

### Giao diện trang thêm danh mục sản phẩm



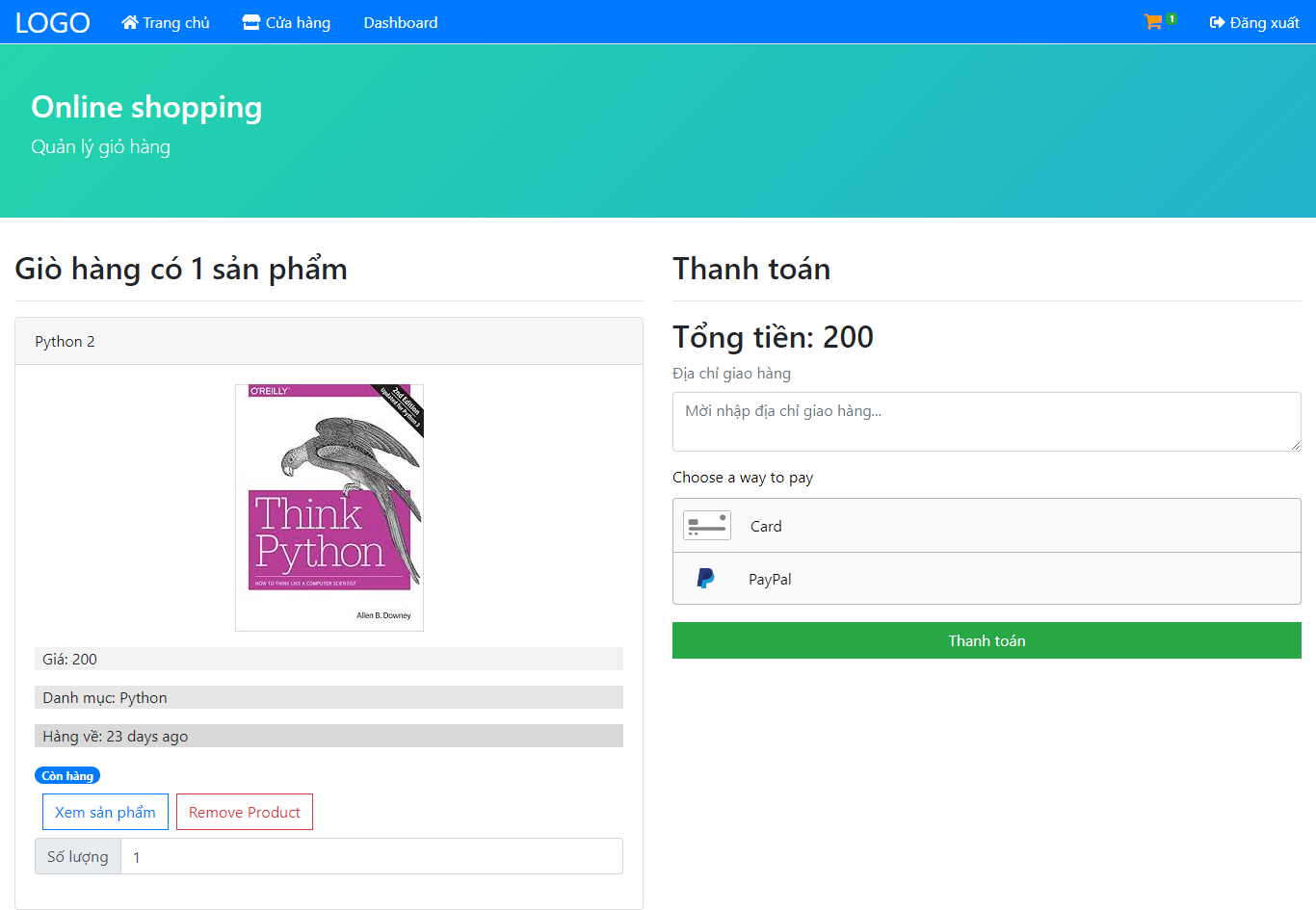
Hình 3.12 Giao diện trang sửa danh mục sản phẩm

### Giao diện trang sửa danh mục sản phẩm



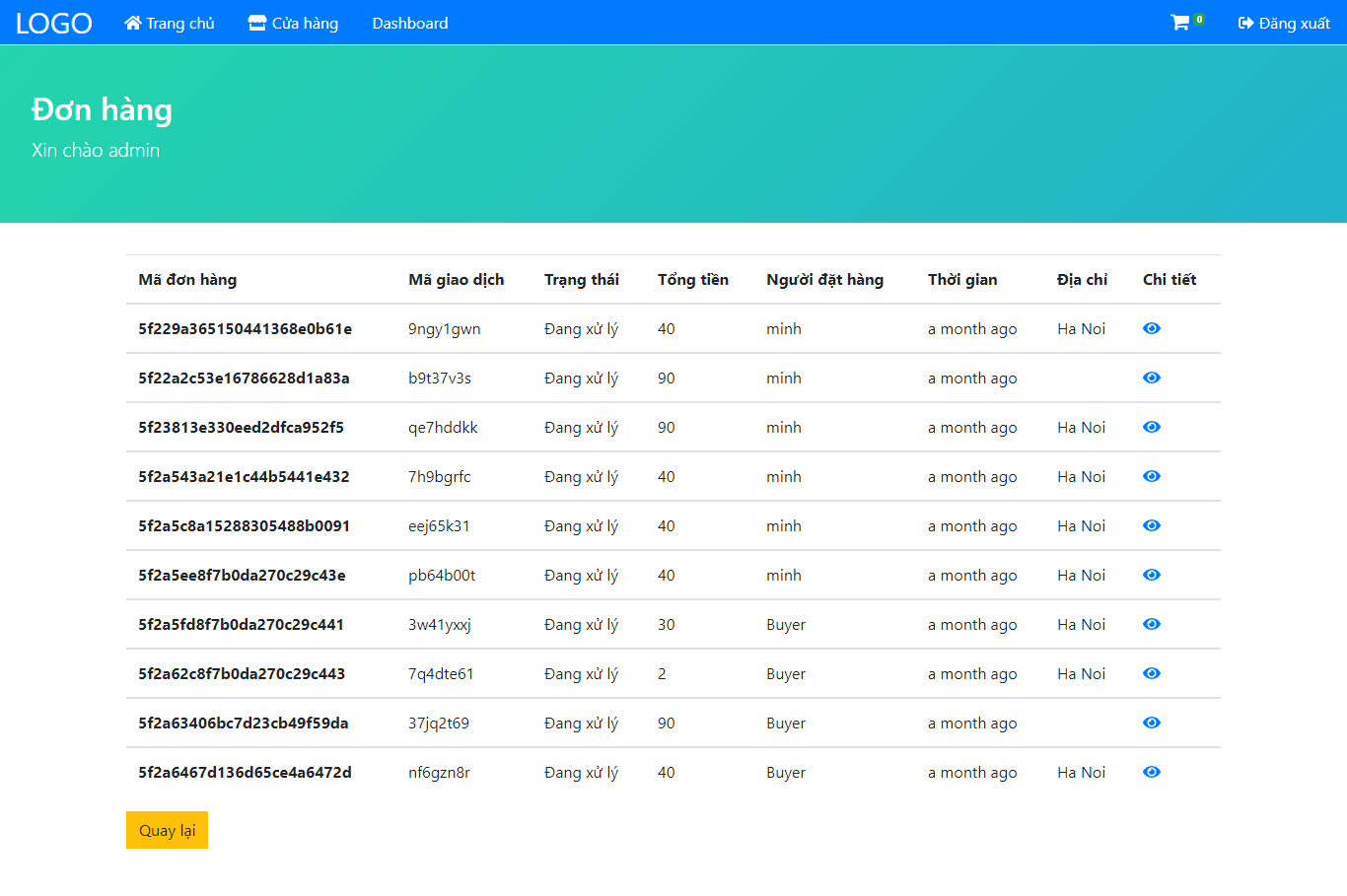
Hình 3.13 Giao diện trang sửa danh mục sản phẩm

### Giao diện trang quản lý giỏ hàng



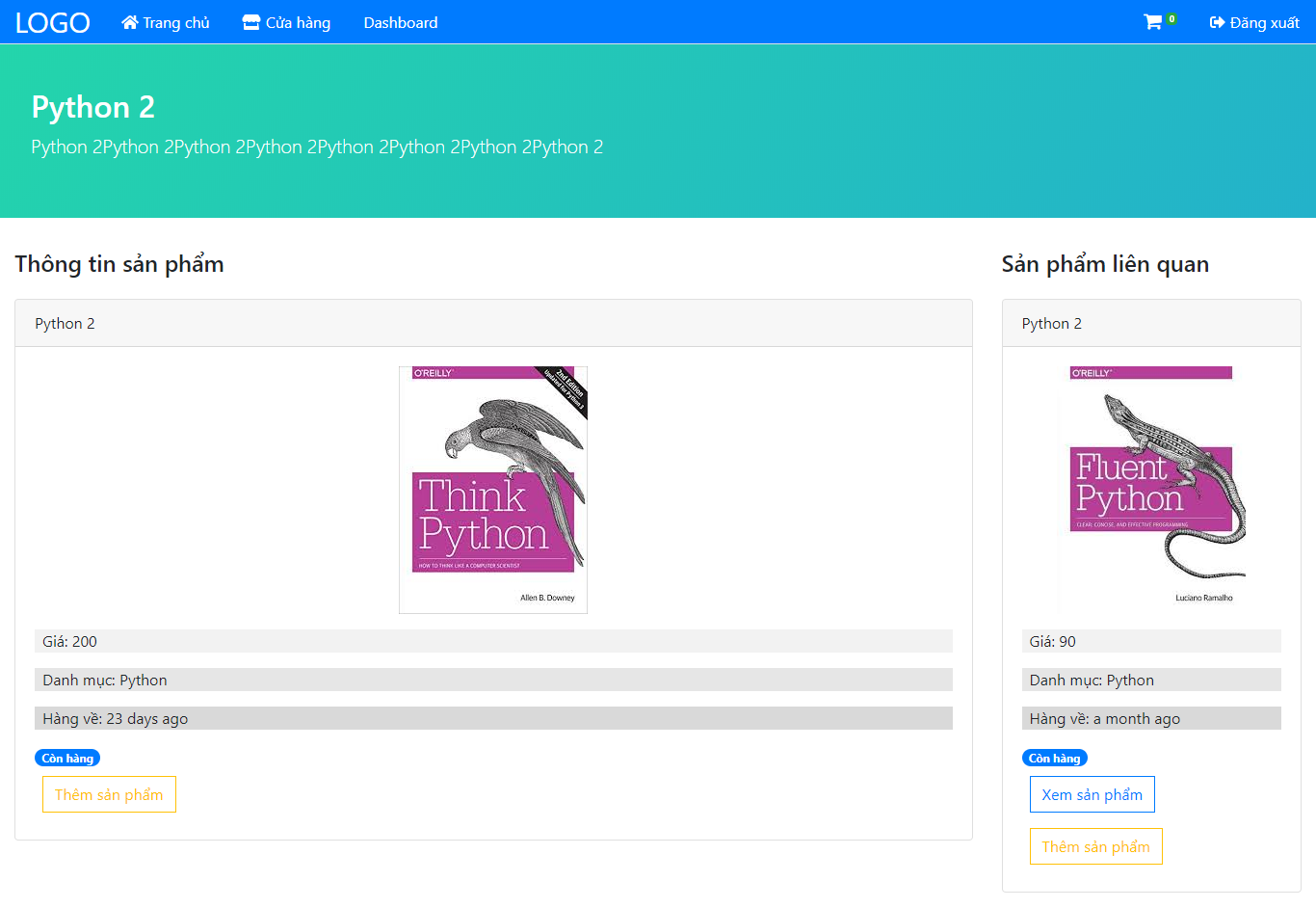
Hình 3.14 Giao diện trang giỏ hàng

### Giao diện trang quản lý đơn hàng



Hình 3.15 Giao diện trang quản lý đơn hàng

### Giao diện trang xem chi tiết sản phẩm



Hình 3.16 Giao diện trang xem chi tiết sản phẩm

# KẾT LUẬN

## Kết quả đạt được

Phần mềm có giao diện dễ nhìn, thân thiện với người sử dụng.

Sau khi nghiên cứu, em đã xây dựng được một website bán hàng, thay thế cách bán hàng truyền thống.

Phần mềm có ứng dụng thực tiễn đối với các cửa hàng muốn thay đổi cách kinh doanh từ truyền thống sang trực tuyến.

## Hạn chế của đề tài

Đề tài xây dựng website bán hàng tuy đã hoàn thành xong vẫn không tránh khỏi những thiếu xót cũng như những hạn chế nhất định:

* Website vẫn còn thiếu chức năng, chưa hoàn thiện.
* Kỹ năng phân tích hệ thống còn nhiều hạn chế.
* Chưa có chức năng sao lưu, phục hồi dữ liệu.

## Hướng phát triển

Tiếp tục bổ sung những chức năng mà website chưa có để phần mềm có thể ứng dụng rộng rãi trong mọi của hàng.

Tiếp tục hoàn thiện chương trình hi vọng có thể đáp ứng được nhu cầu của người sử dụng.

Hoàn thiện chức năng sao lưu phục hồi dữ liệu cho chương trình.

Phát triển phần mềm để có thể triển khai ở nhiều cửa hàng khác nhau.

Phát triển ứng dụng trên nền tảng di động.

# TÀI LIỆU THAM KHẢO

1. **Pro MERN Stack**: Full Stack Web App Development with Mongo, Express, React, and Node By Vasan Subramanian

2. https://expressjs.com

3. https://www.mongodb.com

4. https://www.nodejs.com

5. http://www.reactjs.org