

BỘ GIÁO DỤC ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC ĐẠI NAM  
--- 📖 ---



# ĐỒ ÁN TỐT NGHIỆP

## ỨNG DỤNG XÂY DỰNG PHẦN MỀM BÁN HÀNG SỬ DỤNG FLASK VÀ SQL SERVER

SINH VIÊN THỰC HIỆN  
MÃ SINH VIÊN  
KHOA

: BÙI KHẮC MẠNH  
: 1451020152  
: CÔNG NGHỆ THÔNG TIN

HÀ NỘI – 2024

**BỘ GIÁO DỤC ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC ĐẠI NAM**

-----



**BÙI KHẮC MẠNH**

**ỨNG DỤNG XÂY DỰNG PHẦN MỀM BÁN  
HÀNG SỬ DỤNG FLASK VÀ SQL SERVER**

<b>CHUYÊN NGÀNH</b>	<b>: CÔNG NGHỆ THÔNG TIN</b>
<b>MÃ SỐ</b>	<b>: 74.80.201</b>
<b>NGƯỜI HƯỚNG DẪN</b>	<b>: TS. LƯƠNG CAO ĐÔNG</b>

**HÀ NỘI - 2024**

## **LỜI CAM ĐOAN**

Em xin cam đoan rằng đề tài " ứng dụng xây dựng phần mềm bán hàng sử dụng flask và sql server" là một dự án nghiên cứu mà em thực hiện độc lập dưới sự hướng dẫn của giáo viên hướng dẫn: Lương Cao Đông. Em cam kết không sao chép từ bất kỳ nguồn nào khác. Nội dung của báo cáo tốt nghiệp này là kết quả của quá trình nỗ lực và nghiên cứu của bản thân em trong suốt thời gian học tập tại trường và thực tập. Mọi thông tin, số liệu và kết quả được trình bày trong báo cáo này là hoàn toàn trung thực và đáng tin cậy. Em sẵn sàng chịu trách nhiệm và chấp nhận mọi hình thức kỷ luật theo quy định của bộ môn và nhà trường nếu có bất kỳ vấn đề nào phát sinh.

Sinh viên

Bùi Khắc Mạnh

## LỜI CẢM ƠN

Trong thời gian thực hiện đồ án với sự giúp đỡ tạo điều kiện của trường Đại Học Đại Nam, sự góp ý của các bạn và đặc biệt là sự hướng dẫn trực tiếp, chỉ bảo tận tình của giảng viên bộ môn Lương Cao Đông, chúng em đã hoàn thành đề tài cùng với bản báo cáo đúng thời gian quy định.

Với khả năng và thời gian có hạn nên không tránh khỏi những thiếu sót, chúng em rất mong nhận được sự quan tâm, giúp đỡ, tạo điều kiện của thầy cô giáo để chúng em hoàn thiện hơn đề tài nghiên cứu trong thời gian tới.

Một lần nữa chúng em xin chân thành cảm ơn tất cả các thầy, cô giáo trong trường Đại Học Đại Nam dạy dỗ, chỉ bảo em trong suốt thời gian học. Đặc biệt chúng em xin gửi lời cảm ơn sâu sắc tới thầy giáo Lương Cao Đông đã hướng dẫn chúng em suốt quá trình làm báo cáo.

*Chúng em xin chân thành cảm ơn*

## DANH MỤC KÝ HIỆU HOẶC CHỮ VIẾT TẮT

STT	Ký hiệu chữ viết tắt	Chữ viết tắt đầy đủ
1	SQL	Microsoft SQL Server
2	SSMS	SQL Server Management Studio
3	API	Application Programming Interface
4	ORM	object-relational mapper
5	JS	Javascript
6	HTML	Hypertext Markup Language
7	CSS	Cascading Style Sheet

## MỤC LỤC

<b>LỜI CAM ĐOAN.....</b>	<b>3</b>
<b>LỜI CẢM ƠN .....</b>	<b>4</b>
<b>DANH MỤC KÝ HIỆU HOẶC CHỮ VIẾT TẮT .....</b>	<b>5</b>
<b>MỤC LỤC HÌNH ẢNH VÀ BẢNG .....</b>	<b>9</b>
<b>MỞ ĐẦU.....</b>	<b>11</b>
<b>CHƯƠNG I: TỔNG QUAN.....</b>	<b>12</b>
<b>1.1. Giới thiệu về đề tài.....</b>	<b>12</b>
<b>1.2. Lý do chọn đề tài. ....</b>	<b>12</b>
<b>1.3. Mục đích phát triển đề tài. ....</b>	<b>13</b>
<b>1.4. Các chức năng và hệ thống hoạt động của website. ....</b>	<b>14</b>
1.4.1. Chức năng Quản lý Tài khoản. ....	14
1.4.2. Chức năng Quản lý Sản phẩm. ....	14
1.4.3. Chức năng Quản lý Bình luận.....	15
1.4.4. Chức năng Giỏ hàng. ....	15
1.4.5. Chức năng Tìm kiếm và Xem Chi tiết Sản phẩm.....	15
<b>1.5. Phân chia hệ thống.....</b>	<b>16</b>
1.5.1. Phần thứ nhất: Giao diện Khách hàng.....	16
1.5.2. Phần thứ hai: Giao diện Quản trị. ....	17
<b>1.6. Kiến thức thực tế còn hạn chế: .....</b>	<b>17</b>
<b>CHƯƠNG II: TÌM HIỂU 1 SỐ CÔNG NGHỆ XÂY DỰNG WEBSITE.....</b>	<b>19</b>
<b>2.1. Tìm hiểu về flask trong python.....</b>	<b>19</b>
2.1.1. Lịch sử phát triển của flask.....	19
2.1.2. Tính thông dụng.....	19
2.1.3. Tổng quan về FLASK trong Python.....	20
2.1.4. Công cụ hỗ trợ. ....	20
2.1.5. Ý nghĩa.....	21
<b>2.2. Tìm hiểu về SQLAlchemy. ....</b>	<b>21</b>
2.2.1Khái niệm SQLAlchemy. ....	21
2.2.2. Thành phần chính của SQLAlchemy.....	21
2.2.3Lợi ích của SQLAlchemy.....	22
2.2.4. Tính năng nổi bật của SQLAlchemy.....	23

<b>2.3. Tìm hiểu về API trong Flask.....</b>	<b>23</b>
2.3.1. Khái niệm về API. ....	23
2.3.2. Lợi ích của việc sử dụng API trong Flask. ....	23
<b>2.4. Tìm hiểu về HTML, CSS và JavaScript trong lập trình Flask. ....</b>	<b>24</b>
2.4.1. Giới thiệu về Flask và mối liên hệ với HTML, CSS, JavaScript. ....	24
2.4.2. HTML (HyperText Markup Language).....	24
2.4.3. CSS (Cascading Style Sheets). ....	25
2.4.4. JavaScript.....	25
<b>2.5. Tìm hiểu về SQL Server. ....</b>	<b>26</b>
2.5.1. Giới thiệu về SQL Server. ....	26
2.5.2. Các thành phần chính của SQL Server.....	27
2.5.3. Các tính năng chính của SQL Server.....	27
2.5.4. Các phiên bản của SQL Server.....	27
<b>CHƯƠNG III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....</b>	<b>29</b>
<b>3.1. Yêu cầu bài toán.....</b>	<b>29</b>
3.1.1. Yêu cầu bài toán. ....	29
3.1.2. Yêu cầu phi chức năng.....	29
<b>3.2. Yêu cầu hệ thống.....</b>	<b>29</b>
<b>3.3. Thiết kế hệ thống.....</b>	<b>30</b>
3.3.1. Kiến trúc tổng quan của hệ thống: .....	30
3.3.2. Xác Định Actor Và Usecase .....	31
<b>3.4. Biểu Đồ Ca Sử Dụng (Usecase Diagram).....</b>	<b>33</b>
3.4.1. Usecase tổng quát. ....	33
3.4.2. Usecase dành cho các tác nhân.....	34
3.4.3. Các biểu đồ usecase quản lý chức năng.....	36
<b>3.5. Đặc Tả Các Use Case .....</b>	<b>42</b>
Biểu đồ trình tự đăng nhập .....	46
<b>3.6. Biểu Đồ Lớp và ER .....</b>	<b>52</b>
3.6.1. Biểu Đồ Lớp (Class Diagram) .....	52
3.6.2. Biểu ER.....	53
<b>3.7. Mô hình cơ sở dữ liệu .....</b>	<b>53</b>
<b>3.8. Lưu đồ thuật toán .....</b>	<b>58</b>

<b>CHƯƠNG IV: PHÁT TRIỂN CHƯƠNG TRÌNH.....</b>	<b>62</b>
<b>4.1. Giao diện hệ thống website bán hàng của người dùng.....</b>	<b>62</b>
4.1.1. Giao diện trang đăng nhập.....	62
<b>4.1.2 Giao diện đăng kí tài khoản cho khách hàng.....</b>	<b>62</b>
4.1.3. Giao diện trang index của website.....	63
4.1.4. Giao diện giỏ hàng.....	65
4.1.5. Giao diện chi tiết sản phẩm.....	66
4.1.6. Giao diện chức năng đánh giá và bình luận cho từng sản phẩm.....	67
<b>4.2. Giao diện quản trị admin.....</b>	<b>68</b>
4.2.1. Giao diện đăng nhập admin.....	68
4.2.2. Giao diện đăng kí tài khoản admin.....	69
4.2.3. Giao diện quản lý sản phẩm.....	70
4.2.4. Giao diện quản lý thêm sản phẩm.....	70
4.2.5. Giao diện quản lý sửa phẩm.....	71
4.2.6. Giao diện quản lý tài khoản khách hàng.....	71
4.2.7. Giao diện quản lý tài khoản admin.....	72
<b>KẾT LUẬN.....</b>	<b>73</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>74</b>
<b>PHỤ LỤC.....</b>	<b>75</b>
<b>Các code cần thiết cho các chức năng của website.....</b>	<b>75</b>
App.py.....	75
Models.py.....	75
Product.py.....	77
Admin_login.html.....	78



## MỤC LỤC HÌNH ẢNH VÀ BẢNG

Hình 1: Flask .....	19
Hình 2: ORM.....	22
Hình 3: HTML .....	25
Hình 4: CSS.....	25
Hình 5: JavaScript .....	26
Hình 6: Usecase tổng quát.....	33
Hình 7: Usecase Admin.....	34
Hình 8: Usecase Khách hàng.....	35
Hình 9: Biểu đồ chi tiết usecase ‘Đăng nhập’ .....	36
Hình 10: Biểu đồ chi tiết usecase ‘Quản lý tài khoản’ .....	38
Hình 11: Biểu đồ chi tiết usecase ‘Đặt hàng’ .....	38
Hình 12: Biểu đồ chi tiết usecase ‘Quản lý sản phẩm’ .....	39
Hình 13: Biểu đồ chi tiết usecase ‘Quản lý danh mục’ .....	39
Hình 14: Biểu đồ chi tiết usecase ‘Quản lý tin tức’ .....	40
Hình 15: Biểu đồ chi tiết usecase ‘Quản lý giỏ hàng’ .....	40
Hình 16: Biểu đồ chi tiết usecase ‘Đăng kí’ .....	41
Hình 17: Biểu đồ chi tiết usecase ‘Quản lý tài khoản cá nhân’ .....	41
Hình 18: Biểu đồ chi tiết chức năng tìm kiếm.....	41
Hình 19: Đặc tả Use case Đăng nhập.....	42
Hình 20: Đặc tả Use case Quản lý sản phẩm.....	43
Hình 21: Đặc tả Use case quản lý khách hàng. ....	44
Hình 22: Đặc tả Use case quản lý danh mục bài viết. ....	45
Hình 23: . Đặc tả Use case quản lý danh mục bài viết. ....	45
Hình 24: Biểu đồ tuần tự đăng nhập.....	46
Hình 25: Biểu đồ tuần tự đăng kí .....	47
Hình 26: Biểu đồ hoạt động chức năng delete người dùng.....	48
Hình 27: Biểu đồ chức năng thêm mới bài viết.....	49
Hình 28: Biểu đồ chức năng sửa mới bài viết .....	50
Hình 29: Biểu đồ chức năng xóa bài viết .....	51
Hình 30: Biểu đồ lớp (class diagram).....	52
Hình 31: Lưu đồ thuật toán đăng ký.....	58
Hình 32: Lưu đồ thuật toán đăng nhập.....	59
Hình 33: Lưu đồ thuật toán thêm sản phẩm. ....	60
Hình 34: Lưu đồ thuật toán thêm sản phẩm vào giỏ hàng.....	61
Hình 35: Giao diện trang đăng nhập. ....	62
Hình 36: Giao diện đăng kí tài khoản cho khách hàng.....	63

Hình 37: Giao diện trang index của website. ....	63
Hình 38: Các sản phẩm .....	64
Hình 39: Giao diện giỏ hàng. ....	65
Hình 40: Giao diện chi tiết sản phẩm. ....	66
Hình 41: Giao diện chức năng đánh giá và bình luận cho từng sản phẩm. ....	67
Hình 42: Giao diện đăng nhập admin.....	68
Hình 43: Giao diện đăng kí tài khoản admin.....	69
Hình 44: Giao diện quản lý sản phẩm. ....	70
Hình 45: Giao diện quản lý thêm sản phẩm .....	70
Hình 46: Giao diện quản lý sửa phẩm. ....	71
Hình 47: Giao diện quản lý tài khoản khách hàng. ....	71
Hình 48: Giao diện quản lý tài khoản admin.....	72
 Bảng 1: Account.....	 54
Bảng 2: CSDL bảng admin .....	54
Bảng 3: CSDL bảng product .....	55
Bảng 4: CSDL bảng cart_item .....	55
Bảng 5: CSDL bảng shopping_cart.....	56
Bảng 6: CSDL bảng REVIEW .....	56
Bảng 7: CSDL bảng hóa đơn .....	57

# MỞ ĐẦU

Trong thời đại số hóa ngày càng phát triển, việc xây dựng các ứng dụng trên nền tảng web và di động đang trở thành xu hướng hàng đầu. Các ứng dụng bán hàng trực tuyến ngày càng phổ biến và trở thành một phần không thể thiếu của nền kinh doanh hiện đại. Để đáp ứng nhu cầu này, việc phát triển một hệ thống cơ sở dữ liệu mạnh mẽ và linh hoạt là vô cùng quan trọng. Đề tài này sẽ giúp chúng ta khám phá cách xây dựng một cơ sở dữ liệu cho ứng dụng bán hàng sử dụng SQL Server - một hệ quản trị cơ sở dữ liệu mạnh mẽ và phổ biến.

SQL Server cung cấp cho chúng ta một nền tảng để quản lý dữ liệu một cách an toàn và hiệu quả. Hệ quản trị này không chỉ hỗ trợ các tính năng lưu trữ và truy xuất dữ liệu mà còn cung cấp các công cụ mạnh mẽ để bảo mật, sao lưu và tối ưu hóa hiệu năng. Việc sử dụng SQL Server trong phát triển ứng dụng bán hàng sẽ giúp chúng ta tạo ra một hệ thống ổn định, bảo mật và dễ dàng mở rộng.

Em sẽ bắt đầu bằng việc thiết lập môi trường phát triển và cài đặt các công cụ cần thiết, bao gồm SQL Server và SQL Server Management Studio (SSMS). Sau đó, em sẽ xây dựng cơ sở dữ liệu và các bảng (tables) cho sản phẩm, đơn hàng, và khách hàng. Trong quá trình này, em sẽ thiết kế các bảng với các khóa chính (primary keys) và khóa ngoại (foreign keys) để đảm bảo tính toàn vẹn dữ liệu.

Tiếp theo, em sẽ tạo các chức năng cơ bản như thêm, sửa, và xóa sản phẩm thông qua các câu lệnh SQL. Các chức năng này sẽ được thực hiện bằng cách sử dụng các câu lệnh INSERT, UPDATE và DELETE. Em cũng sẽ học cách truy vấn dữ liệu (SELECT) để hiển thị thông tin sản phẩm, tìm kiếm và lọc dữ liệu theo các tiêu chí khác nhau. Việc này sẽ giúp em hiểu rõ hơn về cách thao tác với cơ sở dữ liệu và tối ưu hóa các truy vấn để đạt được hiệu năng tốt nhất. Em cũng sẽ tập trung vào các kỹ thuật bảo mật cơ sở dữ liệu, bao gồm việc quản lý người dùng và quyền truy cập, mã hóa dữ liệu và thiết lập các biện pháp bảo vệ chống lại các cuộc tấn công SQL injection.

Cuối cùng, em sẽ kiểm tra và sử dụng cơ sở dữ liệu thông qua các công cụ quản trị như SQL Server Management Studio (SSMS). Em sẽ thực hiện các thao tác kiểm thử để đảm bảo rằng các chức năng của cơ sở dữ liệu hoạt động đúng như mong đợi và đáp ứng được các yêu cầu của ứng dụng bán hàng. Em sẽ thử nghiệm các kịch bản khác nhau để đảm bảo tính ổn định và độ tin cậy của hệ thống.

Mục tiêu của đề tài này không chỉ là giúp em có được một bức tranh tổng quan về cách xây dựng một cơ sở dữ liệu sử dụng SQL Server cho ứng dụng bán hàng, mà còn là cung cấp cho em kỹ năng và kiến thức cần thiết để bắt đầu phát triển và mở rộng các dự án của em trong tương lai. Từ đó, mở ra nhiều cơ hội và tiềm năng trong sự nghiệp và học tập của em. Việc nắm vững các kỹ năng này sẽ giúp em tự tin hơn khi tham gia vào các dự án thực tế và đóng góp vào sự phát triển của lĩnh vực công nghệ thông tin.

# CHƯƠNG I: TỔNG QUAN

## 1.1. Giới thiệu về đề tài.

Trong bối cảnh hiện nay, với sự phát triển vượt bậc của công nghệ và internet, các doanh nghiệp đang ngày càng chuyển hướng sang nền tảng trực tuyến để mở rộng thị trường và tiếp cận khách hàng. Một trong những công cụ quan trọng hỗ trợ xu hướng này là các ứng dụng bán hàng trực tuyến, giúp doanh nghiệp quản lý hàng hóa, đơn hàng và nâng cao trải nghiệm mua sắm của khách hàng. Đề tài này tập trung vào việc xây dựng một ứng dụng bán hàng trên nền tảng Flask - một microframework viết bằng Python, nổi tiếng với tính nhẹ nhàng, dễ sử dụng và mạnh mẽ. Mục tiêu chính bao gồm thiết kế và phát triển một hệ thống bán hàng trực tuyến hoàn chỉnh từ giao diện người dùng đến backend xử lý dữ liệu, tích hợp cơ sở dữ liệu để quản lý thông tin sản phẩm, đơn hàng và người dùng, đảm bảo tính bảo mật cao và hiệu năng ổn định, đồng thời tạo ra giao diện thân thiện, dễ sử dụng. Ứng dụng sẽ bao gồm các chức năng chính như quản lý sản phẩm, giỏ hàng và thanh toán, quản lý người dùng, hỗ trợ tìm kiếm và bộ lọc, cùng tích hợp thanh toán trực tuyến. Flask được lựa chọn vì tính nhẹ và linh hoạt, cộng đồng hỗ trợ lớn và khả năng tích hợp dễ dàng với các thư viện và công cụ khác trong hệ sinh thái Python. Việc phát triển ứng dụng bán hàng trên nền tảng Flask không chỉ là một bài tập kỹ thuật mà còn là một giải pháp thực tiễn, đáp ứng nhu cầu của doanh nghiệp trong thời đại kỹ thuật số, mang lại nhiều giá trị thực tế cho người dùng và doanh nghiệp, đồng thời tạo tiền đề cho những dự án công nghệ tiếp theo.

## 1.2. Lý do chọn đề tài.

Lý do chọn đề tài phát triển ứng dụng bán hàng trên nền tảng Flask xuất phát từ nhu cầu thực tiễn và tiềm năng phát triển trong lĩnh vực thương mại điện tử. Trước hết, với sự bùng nổ của internet và công nghệ, các doanh nghiệp đang dần chuyển dịch sang kinh doanh trực tuyến để tiếp cận một lượng khách hàng rộng lớn hơn và tối ưu hóa hoạt động kinh doanh. Ứng dụng bán hàng trực tuyến không chỉ giúp doanh nghiệp quản lý hiệu quả các sản phẩm, đơn hàng, và khách hàng mà còn nâng cao trải nghiệm mua sắm, từ đó gia tăng doanh thu và sự hài lòng của khách hàng.

Flask, một microframework viết bằng Python, được lựa chọn vì tính linh hoạt và dễ sử dụng. Flask cho phép các nhà phát triển xây dựng ứng dụng một cách nhanh chóng và hiệu quả, đồng thời dễ dàng mở rộng và tùy chỉnh theo nhu cầu cụ thể của từng doanh nghiệp. Với cộng đồng hỗ trợ rộng lớn và tài liệu phong phú, Flask cung cấp một nền tảng vững chắc cho việc phát triển ứng dụng web. Khả năng tích hợp dễ dàng với các thư viện và công cụ khác trong hệ sinh thái Python, như SQLAlchemy cho ORM, Jinja2 cho template engine, và nhiều thư viện khác, là một điểm mạnh giúp tối ưu hóa quá trình phát triển.

Chọn đề tài này còn mang lại cơ hội học hỏi và áp dụng các kiến thức về phát triển ứng dụng web, bảo mật, cơ sở dữ liệu và trải nghiệm người dùng. Việc xây dựng một ứng dụng bán hàng hoàn chỉnh không chỉ là một thử thách kỹ thuật mà còn là một trải nghiệm quý báu giúp các nhà phát triển nâng cao kỹ năng và hiểu biết về quy trình phát triển phần mềm từ đầu đến cuối. Với những lợi ích và tiềm năng phát triển, đề tài phát triển ứng dụng bán hàng trên nền tảng Flask không chỉ đáp ứng nhu cầu của thị trường mà còn góp phần thúc đẩy sự phát triển của các giải pháp công nghệ trong lĩnh vực thương mại điện tử[5].

### **1.3. Mục đích phát triển đề tài.**

Mục đích phát triển đề tài "Ứng dụng bán hàng trên nền tảng Flask" là nhằm tạo ra một giải pháp công nghệ hữu hiệu phục vụ cho hoạt động thương mại điện tử, đồng thời mang lại những giá trị thiết thực cho doanh nghiệp và người tiêu dùng. Cụ thể, mục đích của đề tài này bao gồm:

Xây dựng hệ thống bán hàng trực tuyến hoàn chỉnh: Phát triển một ứng dụng có khả năng quản lý toàn bộ quy trình kinh doanh từ việc giới thiệu sản phẩm, quản lý giỏ hàng, thực hiện thanh toán, đến việc theo dõi và quản lý đơn hàng. Điều này giúp các doanh nghiệp dễ dàng kiểm soát hoạt động kinh doanh của mình và nâng cao hiệu quả vận hành.

Nâng cao trải nghiệm người dùng: Tạo ra một giao diện thân thiện, dễ sử dụng và hấp dẫn, giúp khách hàng có trải nghiệm mua sắm trực tuyến tốt nhất. Ứng dụng sẽ tích hợp các chức năng tìm kiếm, bộ lọc sản phẩm, và hệ thống giỏ hàng tiện lợi, nhằm đáp ứng tối đa nhu cầu của người dùng.

Đảm bảo tính bảo mật và ổn định: Phát triển một hệ thống với độ bảo mật cao để bảo vệ thông tin cá nhân và dữ liệu giao dịch của người dùng. Đồng thời, ứng dụng cần đảm bảo hoạt động ổn định dưới tải trọng lớn, đáp ứng nhu cầu truy cập và sử dụng liên tục của khách hàng.

Tích hợp cơ sở dữ liệu hiệu quả: Sử dụng các hệ quản trị cơ sở dữ liệu như SQLite hoặc MySQL để lưu trữ và quản lý thông tin sản phẩm, đơn hàng, và người dùng một cách khoa học và hiệu quả, giúp doanh nghiệp dễ dàng truy xuất và xử lý dữ liệu.

Áp dụng và nâng cao kỹ năng công nghệ: Thông qua việc phát triển ứng dụng, các nhà phát triển có cơ hội áp dụng kiến thức về lập trình web, quản lý cơ sở dữ liệu, bảo mật thông tin và trải nghiệm người dùng. Đây là cơ hội để nâng cao kỹ năng và kinh nghiệm trong lĩnh vực phát triển phần mềm.

Tạo tiền đề cho các dự án tương lai: Dự án này không chỉ dừng lại ở việc tạo ra một ứng dụng bán hàng mà còn đặt nền tảng cho những phát triển công nghệ sau này, có thể mở rộng và tùy biến cho các mục đích kinh doanh khác nhau, từ đó góp phần vào sự phát triển chung của công nghệ thông tin và thương mại điện tử.

Tóm lại, mục đích của đề tài là phát triển một ứng dụng bán hàng trực tuyến trên nền

tăng Flask với tính năng toàn diện, bảo mật cao và trải nghiệm người dùng tốt, đồng thời cung cấp cơ hội học hỏi và phát triển kỹ năng công nghệ cho các nhà phát triển. Ý nghĩa thực tiễn. Phát triển các sản phẩm công nghệ phù hợp với nhu cầu của xã hội, qua đó rèn luyện, trau dồi, phát triển kiến thức của bản thân nhằm đáp ứng nhu cầu của các nhà tuyển dụng sau khi ra trường, tạo ra sản phẩm thực tế có giá trị sử dụng trong cuộc sống.

Đặt bài toán

Bài toán đặt ra: Xây dựng website bán hàng gồm bao gồm các chức năng của 1 website bán hàng đơn thuần

Qua quá trình khảo sát các nguồn thông tin từ thầy giáo hướng dẫn, bạn bè, những người có kinh nghiệm trong phát triển website, trải nghiệm thực tế trên một số website bán hàng trực tuyến có thương hiệu của nước ngoài như [5]

- <http://www.ebay.com>,
- <http://www.alibaba.com>
- <https://www.thegioididong.com>
- <http://www.trananh.vn/>

em phân tích bài toán:

## **1.4. Các chức năng và hệ thống hoạt động của website.**

### **1.4.1. Chức năng Quản lý Tài khoản.**

A, Đăng ký và Đăng nhập:

- Người dùng có thể tạo tài khoản mới bằng cách cung cấp thông tin cá nhân cơ bản như tên, email và mật khẩu.
- Hệ thống sẽ xác thực thông tin và tạo tài khoản mới, sau đó cho phép người dùng đăng nhập vào hệ thống.

B, Quản lý Hồ sơ Người dùng:

- Người dùng có thể cập nhật thông tin cá nhân như tên, địa chỉ, số điện thoại và email.
- Tính năng này giúp người dùng duy trì thông tin chính xác và cập nhật nhất cho tài khoản của mình.

C, Quên Mật khẩu:

- Cung cấp tính năng để người dùng có thể khôi phục mật khẩu qua email nếu họ quên.

D, Đổi Mật khẩu:

- Người dùng có thể thay đổi mật khẩu hiện tại của mình trong phần cài đặt tài khoản.
- Hệ thống sẽ yêu cầu người dùng nhập mật khẩu cũ để xác thực trước khi cho phép đặt mật khẩu mới.

### **1.4.2. Chức năng Quản lý Sản phẩm.**

A, Thêm Sản phẩm Mới:

- Quản trị viên có thể thêm sản phẩm mới vào hệ thống bằng cách nhập thông tin chi tiết như tên sản phẩm, mô tả, giá, số lượng tồn kho và hình ảnh sản phẩm.
- Sửa đổi Thông tin Sản phẩm:
- Quản trị viên có thể cập nhật thông tin sản phẩm hiện có, bao gồm thay đổi giá, mô tả, số lượng tồn kho và hình ảnh.

B, Xóa Sản phẩm:

- Quản trị viên có thể xóa các sản phẩm không còn kinh doanh hoặc bị lỗi.

#### ***1.4.3. Chức năng Quản lý Bình luận.***

A, Thêm Bình luận:

- Người dùng đã đăng nhập có thể thêm bình luận và đánh giá cho các sản phẩm mà họ đã mua.

B, Xóa và Sửa Bình luận:

- Quản trị viên có quyền xóa các bình luận không phù hợp hoặc vi phạm chính sách của cửa hàng.
- Người dùng có thể sửa hoặc xóa bình luận của chính mình.

#### ***1.4.4. Chức năng Giỏ hàng.***

A, Thêm vào Giỏ hàng:

- Người dùng có thể thêm sản phẩm vào giỏ hàng từ trang chi tiết sản phẩm.

B, Quản lý Giỏ hàng:

- Người dùng có thể xem danh sách các sản phẩm trong giỏ hàng, thay đổi số lượng hoặc xóa sản phẩm khỏi giỏ hàng.

C, Thanh toán:

- Người dùng có thể tiến hành thanh toán các sản phẩm trong giỏ hàng. Hệ thống sẽ tính tổng giá trị đơn hàng và cung cấp các phương thức thanh toán trực tuyến an toàn.

#### ***1.4.5. Chức năng Tìm kiếm và Xem Chi tiết Sản phẩm.***

A, Tìm kiếm Sản phẩm:

- Người dùng có thể tìm kiếm sản phẩm dựa trên từ khóa liên quan như tên sản phẩm, mô tả, hoặc danh mục sản phẩm.
- Hệ thống sẽ hiển thị các kết quả tìm kiếm phù hợp để người dùng dễ dàng tìm thấy sản phẩm mình cần.

B, Bộ lọc Sản phẩm:

- Người dùng có thể lọc sản phẩm theo các tiêu chí như giá cả, danh mục, đánh giá, hoặc các thuộc tính khác để thu hẹp kết quả tìm kiếm.

C, Xem Chi tiết Sản phẩm:

- Khi người dùng chọn một sản phẩm từ danh sách, họ có thể xem trang chi tiết

sản phẩm với đầy đủ thông tin như mô tả chi tiết, giá cả, hình ảnh, đánh giá từ người dùng khác, và các tùy chọn mua hàng.

- Trang chi tiết sản phẩm cung cấp thông tin rõ ràng và toàn diện giúp người dùng quyết định mua hàng.

Các chức năng trên không chỉ giúp nâng cao trải nghiệm người dùng mà còn tối ưu hóa quy trình quản lý cho các quản trị viên, đảm bảo hệ thống hoạt động hiệu quả và đáp ứng đầy đủ nhu cầu của cả người mua và người bán. *Yêu cầu đặt ra cho Hệ thống Bán hàng trên nền tảng Flask*

### **1.5. Phân chia hệ thống.**

Hệ thống được chia thành hai phần chính: Giao diện Khách hàng và Giao diện Quản trị. Mỗi phần có những yêu cầu cụ thể nhằm đảm bảo tính năng và trải nghiệm người dùng tốt nhất.

#### **1.5.1. Phần thứ nhất: Giao diện Khách hàng.**

Khách hàng truy cập vào website để tìm kiếm và mua hàng. Giao diện dành cho khách hàng cần phải có các chức năng và yêu cầu sau:

A, Giao diện Người dùng Thân thiện:

- Thiết kế giao diện dễ sử dụng, trực quan và hấp dẫn.
- Có khả năng thay đổi giao diện theo các dịp lễ, khuyến mãi, và sự kiện đặc biệt để thu hút người dùng và kích thích nhu cầu mua sắm.

B, Hiển thị Danh sách Sản phẩm:

- Cung cấp danh sách các mặt hàng với thông tin chi tiết như tên sản phẩm, mô tả, giá cả, hình ảnh và đánh giá.
- Các sản phẩm được phân loại theo danh mục, giúp người dùng dễ dàng tìm kiếm và lựa chọn.

C, Chức năng Tìm kiếm và Bộ lọc:

- Cho phép người dùng tìm kiếm sản phẩm bằng từ khóa liên quan.
- Cung cấp bộ lọc sản phẩm theo các tiêu chí như giá, danh mục, đánh giá, v.v.

D, Trang Chi tiết Sản phẩm:

- Hiển thị thông tin chi tiết của từng sản phẩm bao gồm mô tả, giá, hình ảnh, và đánh giá từ khách hàng.
- Có nút "Thêm vào giỏ hàng" để người dùng có thể nhanh chóng thêm sản phẩm vào giỏ hàng của mình.

E, Chức năng Giỏ hàng:

- Hiển thị các sản phẩm đã được thêm vào giỏ hàng, cho phép người dùng thay đổi số lượng hoặc xóa sản phẩm khỏi giỏ hàng.
- Hiển thị tổng tiền đơn hàng và các chi tiết liên quan.

F, Giao diện Đặt hàng:

- Cung cấp biểu mẫu đặt hàng bao gồm thông tin về hàng hóa đã chọn, số lượng,



số tiền và tổng tiền đơn hàng.

- Yêu cầu thông tin cá nhân của khách hàng như tên, địa chỉ, số điện thoại và email để phục vụ việc liên lạc và giao hàng.

G, Trang Phản hồi Khách hàng:

- Tạo trang để khách hàng có thể gửi ý kiến, câu hỏi và phản hồi về sản phẩm và dịch vụ.
- Thu thập thông tin phản hồi để cải thiện chất lượng sản phẩm và dịch vụ.

### **1.5.2. Phần thứ hai: Giao diện Quản trị.**

Giao diện quản trị cho phép người quản lý kiểm soát toàn bộ hoạt động của ứng dụng. Các chức năng chính bao gồm:

A, Quản lý Sản phẩm:

- Chức năng thêm, sửa, xóa sản phẩm, danh mục sản phẩm và nhà sản xuất.
- Cập nhật thông tin chi tiết của sản phẩm như tên, mô tả, giá cả, số lượng tồn kho và hình ảnh.

B, Xử lý Đơn hàng:

- Tiếp nhận đơn hàng từ khách hàng và xử lý chúng.
- Cập nhật trạng thái đơn hàng (đang xử lý, đã giao, đã hủy, v.v.).

C, Thống kê Doanh thu:

- Cung cấp báo cáo thống kê doanh thu theo thời gian (ngày, tuần, tháng, năm).
- Hiển thị các thông tin về doanh số bán hàng, sản phẩm bán chạy và tình hình kinh doanh tổng quát.

D, Chức năng Nâng cao:

- Tạo và tùy chỉnh giao diện menu động theo yêu cầu của người quản trị.
- Các tùy chỉnh khác nhằm tối ưu hóa và cá nhân hóa giao diện quản trị.

E, Giao diện Đơn giản và Dễ sử dụng:

- Thiết kế giao diện quản trị đơn giản, tập trung vào các chức năng chính.
- Đảm bảo giao diện dễ sử dụng và trực quan cho người quản trị.
- Những yêu cầu trên đảm bảo hệ thống bán hàng trên nền tảng Flask sẽ đáp ứng đầy đủ nhu cầu của khách hàng và người quản trị, từ đó nâng cao trải nghiệm người dùng và hiệu quả quản lý kinh doanh. Các giai đoạn triển khai dự án.
- Vấn đề - Giải pháp
- Vấn đề - Giải pháp
- Khó khăn

### **1.6. Kiến thức thực tế còn hạn chế:**

- Khi phát triển ứng dụng bán hàng trên nền tảng Flask, sự thiếu hụt kiến thức thực tế có thể gây ra nhiều khó khăn trong việc áp dụng lý thuyết vào thực tiễn.
- Điều này có thể ảnh hưởng đến hiệu quả và chất lượng của sản phẩm cuối cùng.

A, Tự tìm hiểu và dành thời gian học hỏi công nghệ mới:

- Để khắc phục hạn chế về kiến thức, cần chủ động tìm kiếm tài liệu, khóa học trực tuyến, và các nguồn tài nguyên học tập khác để nâng cao hiểu biết về công nghệ Flask và các công nghệ liên quan.
- Tham gia các diễn đàn, cộng đồng lập trình như Stack Overflow, GitHub, hoặc các nhóm trên mạng xã hội để tìm hiểu về các vấn đề thực tế và cách giải quyết chúng.

**B, Giao lưu, học hỏi từ những người có kinh nghiệm:**

- Tham gia các hội thảo, seminar, workshop, và các sự kiện công nghệ để giao lưu và học hỏi từ những chuyên gia và người có kinh nghiệm trong ngành công nghệ thông tin.
- Xây dựng mối quan hệ với các đồng nghiệp, giảng viên và chuyên gia để có thể trao đổi và nhận được những ý kiến phản hồi hữu ích cho dự án.

**C, Sử dụng chiến lược phát triển linh hoạt:**

- Áp dụng phương pháp phát triển linh hoạt (Agile) để thích ứng nhanh với những thay đổi về công nghệ. Việc phát triển theo từng giai đoạn ngắn với các vòng lặp liên tục giúp kiểm soát tiến độ và chất lượng sản phẩm tốt hơn.
- Lên kế hoạch học tập và cập nhật kiến thức theo từng giai đoạn cụ thể, đảm bảo luôn nắm bắt được những xu hướng và công nghệ mới.

**D, Tăng cường sự giao lưu ý tưởng:**

- Tìm cách hợp tác với những người cùng chí hướng hoặc tham gia vào các dự án nhóm để tăng cường sự trao đổi ý tưởng và giải pháp. Điều này không chỉ giúp cải thiện chất lượng dự án mà còn mở rộng góc nhìn và sáng tạo.
- Sử dụng các công cụ cộng tác trực tuyến như Slack, Trello, hoặc Asana để duy trì sự kết nối và hợp tác với các thành viên khác trong dự án.
- Bằng cách áp dụng những giải pháp này, có thể khắc phục những khó khăn và nâng cao hiệu quả trong việc phát triển ứng dụng bán hàng trên nền tảng Flask. Việc liên tục học hỏi và thích ứng với sự thay đổi của công nghệ sẽ giúp đảm bảo chất lượng và tính cạnh tranh của sản phẩm cuối cùng.

## CHƯƠNG II: TÌM HIỂU 1 SỐ CÔNG NGHỆ XÂY DỰNG WEBSITE

### 2.1. Tìm hiểu về flask trong python.

#### 2.1.1. Lịch sử phát triển của flask.

Flask bắt nguồn từ ý tưởng của Armin Ronacher vào năm 2010, người đã muốn tạo ra một framework web Python đơn giản nhưng vẫn đủ mạnh mẽ để giúp nhà phát triển xây dựng các ứng dụng web. Với triết lý "micro framework", Flask ra đời không phải là một "all-in-one" như các framework lớn khác như Django, mà là một bộ công cụ nhỏ gọn ,giúp nhà phát triển linh hoạt hơn trong việc lựa chọn và kết hợp các công nghệ.

Sau phiên bản ban đầu, Flask đã trải qua nhiều phiên bản cập nhật với các tính năng mới và cải tiến, như hỗ trợ WebSocket, RESTful API, Jinja2 templates, và nhiều thứ khác. Cộng đồng của Flask cũng phát triển mạnh mẽ, với hàng ngàn thư viện mở rộng và công cụ hỗ trợ được phát triển bởi cộng đồng [1].



Hình 1: Flask

#### 2.1.2. Tính thông dụng.

Flask đã nhanh chóng trở thành một trong những framework web Python được ưa chuộng nhất. Điều này không chỉ bởi độ đơn giản và dễ dàng sử dụng của nó mà còn là do sự linh hoạt và tiềm năng mở rộng. Flask phù hợp với các dự án từ cá nhân đến doanh nghiệp, từ ứng dụng web nhỏ đến hệ thống lớn.

Với Flask, các nhà phát triển có thể dễ dàng tích hợp với các công nghệ khác như database, cache, authentication, và nhiều hơn nữa. Flask cũng được sử dụng rộng rãi

trong việc xây dựng các dịch vụ API, ứng dụng di động, và ứng dụng IoT [1].

### **2.1.3. Tổng quan về FLASK trong Python.**

- Flask không chỉ là một framework web, mà còn là một cộng đồng, một triết lý, và một hệ sinh thái đầy đủ. Dưới đây là một số điểm nổi bật về Flask:
- Linh hoạt: Flask không ép buộc bạn vào một cấu trúc cụ thể, cho phép bạn tự do tổ chức và thiết kế ứng dụng theo ý muốn. Điều này giúp các nhà phát triển có thể tạo ra các ứng dụng độc đáo và cá nhân hóa.
- Dễ dàng bắt đầu: Với Flask, bạn chỉ cần vài dòng code là đã có thể khởi tạo một ứng dụng web cơ bản. Điều này giúp người mới bắt đầu nhanh chóng có được trải nghiệm và cảm giác về việc phát triển ứng dụng web.
- Thư viện mở rộng: Flask có một hệ thống thư viện mở rộng (Flask extensions) rất phong phú và đa dạng, giúp bạn mở rộng các tính năng và chức năng của ứng dụng một cách dễ dàng. Các thư viện này được cộng đồng phát triển và cung cấp sẵn để giúp các nhà phát triển giải quyết các vấn đề phổ biến mà không cần phải tự viết lại từ đầu.
- Cộng đồng và hỗ trợ: Flask có một cộng đồng lớn và tích cực. Điều này không chỉ giúp giải đáp các câu hỏi và vấn đề mà còn cung cấp cho người mới nhiều tài nguyên và hướng dẫn để bắt đầu và phát triển kỹ năng của mình [1].

### **2.1.4. Công cụ hỗ trợ.**

Flask, dù là một micro-framework, nhưng đã tạo ra một hệ sinh thái phong phú các công cụ và thư viện hỗ trợ, giúp các nhà phát triển xây dựng và mở rộng ứng dụng của họ một cách linh hoạt và hiệu quả.

- Jinja2: Là một hệ thống templates mạnh mẽ, cho phép tạo ra các giao diện người dùng linh hoạt và dễ dàng thay đổi. Jinja2 hỗ trợ các biểu thức, vòng lặp, và điều kiện, giúp các nhà phát triển tạo ra các trang web động và hấp dẫn.
- Flask-WTF: Thư viện này cung cấp các công cụ quản lý và xác thực dữ liệu nhập từ người dùng. Đồng thời, nó giúp xây dựng các form một cách tiện lợi và nhanh chóng, từ đó giúp tối ưu hóa quá trình xây dựng và quản lý ứng dụng.
- Flask-SQLAlchemy: Đây là một extension mạnh mẽ giúp kết nối và tương tác với các cơ sở dữ liệu. Flask-SQLAlchemy giúp tạo ra các model, thực thi các truy vấn và quản lý các mối quan hệ dữ liệu một cách dễ dàng và linh hoạt.
- Flask-Login: Được sử dụng rộng rãi để quản lý việc đăng nhập và xác thực người dùng. Với Flask-Login, việc xác thực, quản lý session và bảo mật người dùng trở nên đơn giản và hiệu quả.
- Flask-RESTful: Đây là một extension giúp xây dựng các API RESTful trong ứng dụng Flask. Flask-RESTful cung cấp một cách tiện lợi và hiệu quả để quản lý các route và endpoints, giúp tạo ra các API mạnh mẽ và linh hoạt [1].

### **2.1.5. Ý nghĩa.**

Flask không chỉ đơn thuần là một framework web, nó còn mang lại một loạt các ý nghĩa và giá trị quan trọng cho cộng đồng và lập trình viên Python:

- **Sự Linh Hoạt:** Flask đã mở ra một cách tiếp cận mới cho việc xây dựng ứng dụng web. Sự linh hoạt của nó cho phép các nhà phát triển tự do tạo ra các ứng dụng theo ý muốn, không bị ràng buộc bởi cấu trúc và quy chuẩn cụ thể.
- **Dễ Dàng Bắt Đầu:** Flask đã giúp giảm bớt sự phức tạp và ngưỡng cửa cho người mới bắt đầu trong lập trình web. Nhờ vào cú pháp đơn giản và hướng dẫn rõ ràng, Flask đã trở thành một lựa chọn lý tưởng cho những người mới bước chân vào lập trình web.
- **Cộng Đồng và Hỗ Trợ:** Cộng đồng Flask rất mạnh mẽ và tích cực. Nhờ sự đóng góp của cộng đồng, Flask ngày càng được cải thiện và phát triển, đồng thời cung cấp cho người mới nhiều nguồn học tập và hỗ trợ.
- **Khả Năng Mở Rộng:** Mặc dù là một micro-framework, nhưng Flask có khả năng mở rộng tốt. Từ các dự án nhỏ cá nhân đến các hệ thống ứng dụng lớn, Flask đều có khả năng đáp ứng và phát triển một cách linh hoạt và hiệu quả.
- **thúc Đẩy Sự Phát Triển của Python:** Flask đã và đang đóng góp không nhỏ vào sự phát triển của ngôn ngữ Python trong lĩnh vực web development. Nhờ vào Flask, Python đã trở thành một trong những ngôn ngữ phổ biến nhất trong việc xây dựng ứng dụng web, thu hút nhiều lập trình viên và công ty công nghệ quan tâm và đầu tư.

Tóm lại, Flask không chỉ đơn giản là một công cụ, mà còn là một biểu tượng, một triết lý và một phần của lịch sử và sự phát triển của ngôn ngữ lập trình Python trong lĩnh vực web development[1].

## **2.2. Tìm hiểu về SQLAlchemy.**

### **2.2.1 Khái niệm SQLAlchemy.**

SQLAlchemy là một thư viện SQL toolkit và Object-Relational Mapping (ORM) cho ngôn ngữ lập trình Python. Nó cung cấp các công cụ và API giúp các nhà phát triển làm việc với cơ sở dữ liệu một cách hiệu quả, linh hoạt và tiện lợi hơn. SQLAlchemy hỗ trợ nhiều hệ quản trị cơ sở dữ liệu (DBMS) và cho phép lập trình viên tương tác với cơ sở dữ liệu thông qua cả ngôn ngữ biểu thức SQL cấp thấp và ánh xạ đối tượng ORM cấp cao[2].

### **2.2.2. Thành phần chính của SQLAlchemy.**

*SQLAlchemy được chia thành hai thành phần chính:*

A, SQLAlchemy Core:

- **SQL Expression Language:** Đây là ngôn ngữ biểu thức SQL cấp thấp mà SQLAlchemy cung cấp. Nó cho phép lập trình viên xây dựng và thực thi các câu

lệnh SQL một cách trực quan và linh hoạt. Ngôn ngữ này hỗ trợ đầy đủ các thao tác SQL như SELECT, INSERT, UPDATE, DELETE, JOIN và các loại biểu thức khác.

- Schema Metadata: SQLAlchemy Core cung cấp các công cụ để định nghĩa và quản lý các bảng, cột và các ràng buộc của cơ sở dữ liệu. Bạn có thể tạo, sửa đổi và xóa các cấu trúc bảng trực tiếp từ mã nguồn Python.
- Connection Pooling: Một tính năng quan trọng của SQLAlchemy Core là quản lý kết nối cơ sở dữ liệu thông qua connection pooling, giúp tối ưu hóa hiệu suất và quản lý tài nguyên kết nối một cách hiệu quả[2].

#### B, SQLAlchemy ORM:

- Object-Relational Mapping (ORM): ORM là một kỹ thuật lập bản đồ giữa các bảng trong cơ sở dữ liệu và các lớp trong Python. SQLAlchemy ORM cho phép bạn tương tác với cơ sở dữ liệu thông qua các đối tượng Python, giúp mã nguồn trở nên dễ đọc và dễ bảo trì hơn. Các bảng và cột trong cơ sở dữ liệu được ánh xạ thành các lớp và thuộc tính trong Python.
- Session Management: ORM cung cấp một lớp Session để quản lý các phiên làm việc với cơ sở dữ liệu. Session theo dõi các thay đổi đối với các đối tượng và thực hiện các giao dịch (transactions) một cách tự động.
- Query Building: ORM cho phép bạn xây dựng các truy vấn cơ sở dữ liệu một cách dễ dàng và linh hoạt thông qua các phương thức truy vấn của Python. Điều này giúp bạn thực hiện các truy vấn phức tạp mà không cần viết các câu lệnh SQL thủ công[2].



Hình 2: ORM

#### 2.2.3 Lợi ích của SQLAlchemy.

- Tương thích đa cơ sở dữ liệu: SQLAlchemy hỗ trợ nhiều DBMS phổ biến như SQLite, MySQL, PostgreSQL, Oracle, và Microsoft SQL Server. Điều này giúp lập trình viên dễ dàng chuyển đổi giữa các DBMS mà không cần thay đổi mã

nguồn.

- Linh hoạt và mạnh mẽ: SQLAlchemy cung cấp cả công cụ cấp thấp (SQL Expression Language) và cấp cao (ORM), cho phép lập trình viên lựa chọn phương pháp phù hợp nhất cho dự án của mình. Bạn có thể tận dụng sức mạnh của SQL khi cần và sự tiện lợi của ORM khi muốn.
- Đơn giản hóa các thao tác với cơ sở dữ liệu: ORM giúp bạn thao tác với cơ sở dữ liệu thông qua các đối tượng Python, giảm thiểu lỗi và tăng tốc độ phát triển. Bạn không cần phải viết các câu lệnh SQL phức tạp mà vẫn có thể thực hiện các thao tác CRUD dễ dàng.
- Cộng đồng mạnh mẽ và tài liệu phong phú: SQLAlchemy có một cộng đồng lớn và tài liệu phong phú, giúp bạn dễ dàng tìm kiếm sự hỗ trợ và giải pháp cho các vấn đề gặp phải trong quá trình phát triển[2].

#### **2.2.4. Tính năng nổi bật của SQLAlchemy.**

- Declarative Syntax: Cho phép định nghĩa các lớp và ánh xạ với các bảng cơ sở dữ liệu một cách rõ ràng và dễ hiểu.
- Automatic Schema Generation: SQLAlchemy có khả năng tự động tạo các cấu trúc bảng trong cơ sở dữ liệu từ các lớp Python được định nghĩa.
- Transaction Management: Quản lý các giao dịch cơ sở dữ liệu một cách tự động và hiệu quả, đảm bảo tính toàn vẹn dữ liệu.
- Lazy Loading: Hỗ trợ kỹ thuật tải dữ liệu "lười biếng", tức là chỉ tải dữ liệu khi cần thiết, giúp tối ưu hóa hiệu suất.
- Eager Loading: Hỗ trợ tải dữ liệu "háo hức", tức là tải trước dữ liệu liên quan, giúp giảm số lượng truy vấn cần thiết.

#### **Kết luận**

SQLAlchemy là một thư viện quan trọng và hữu ích cho các lập trình viên Python khi làm việc với cơ sở dữ liệu. Nó cung cấp một bộ công cụ linh hoạt và mạnh mẽ để xử lý các tác vụ liên quan đến SQL và ánh xạ đối tượng. Với sự hỗ trợ rộng rãi cho nhiều hệ quản trị cơ sở dữ liệu và khả năng mở rộng cao, SQLAlchemy là một lựa chọn tuyệt vời cho các dự án phần mềm cần làm việc với cơ sở dữ liệu[2].

### **2.3. Tìm hiểu về API trong Flask.**

#### **2.3.1. Khái niệm về API.**

API (Application Programming Interface) là một tập hợp các giao thức và công cụ cho phép các ứng dụng phần mềm giao tiếp với nhau. Trong bối cảnh phát triển web, API thường đề cập đến các dịch vụ web, cho phép các ứng dụng khác truy cập và sử dụng các chức năng hoặc dữ liệu của một ứng dụng.

#### **2.3.2. Lợi ích của việc sử dụng API trong Flask.**

- Modularity: API giúp tách biệt logic xử lý phía server từ giao diện người dùng, giúp việc phát triển và bảo trì dễ dàng hơn.

- Scalability: Dễ dàng mở rộng ứng dụng bằng cách thêm các điểm cuối (endpoints) mới mà không ảnh hưởng đến các phần khác của ứng dụng.
- Interoperability: Các ứng dụng khác nhau có thể giao tiếp với nhau thông qua API, bất kể chúng được viết bằng ngôn ngữ lập trình nào.
- Reusability: Các thành phần API có thể được tái sử dụng trong nhiều ứng dụng hoặc dịch vụ khác nhau.
- Các thành phần chính của một API trong Flask
- Endpoints: Là các URL tại đó API có thể được truy cập. Mỗi endpoint thường đại diện cho một tài nguyên cụ thể hoặc một hành động trên tài nguyên đó.
- HTTP Methods: Các phương thức HTTP (GET, POST, PUT, DELETE, etc.) được sử dụng để tương tác với các endpoints. Mỗi phương thức thường đại diện cho một hành động CRUD (Create, Read, Update, Delete).
- Request và Response: Request là yêu cầu từ client đến server, chứa thông tin như phương thức HTTP, URL, headers, và dữ liệu gửi kèm. Response là phản hồi từ server đến client, chứa thông tin như mã trạng thái HTTP, headers, và dữ liệu trả về[2].

## **2.4. Tìm hiểu về HTML, CSS và JavaScript trong lập trình Flask.**

### **2.4.1. Giới thiệu về Flask và mối liên hệ với HTML, CSS, JavaScript.**

Flask là một micro-framework cho Python, được sử dụng để phát triển các ứng dụng web. Flask chủ yếu xử lý logic phía server, nhưng để tạo ra một ứng dụng web hoàn chỉnh, cần sử dụng các công nghệ phía client như HTML, CSS và JavaScript. Các công nghệ này giúp định hình giao diện người dùng và cải thiện trải nghiệm người dùng[3].

### **2.4.2. HTML (HyperText Markup Language).**

HTML là ngôn ngữ đánh dấu tiêu chuẩn được sử dụng để tạo và cấu trúc các trang web. Nó cung cấp các phần tử cơ bản như tiêu đề, đoạn văn, liên kết, hình ảnh và các yếu tố khác của trang web.

Vai trò của HTML trong Flask:

- Cấu trúc trang web: HTML tạo nên khung sườn của các trang web mà Flask sẽ gửi tới client.
- Templates: Flask sử dụng Jinja2 template engine để nhúng Python vào HTML, cho phép tạo ra các trang web động. Các tệp HTML được lưu trong thư mục templates của dự án Flask và có thể bao gồm các biến, vòng lặp và điều kiện để tạo ra nội dung động[3].





Hình 3: HTML

### 2.4.3. CSS (Cascading Style Sheets).

CSS là ngôn ngữ dùng để mô tả cách trình bày các tài liệu HTML. CSS điều khiển giao diện của trang web, bao gồm màu sắc, bố cục, phông chữ và nhiều yếu tố khác.

Vai trò của CSS trong Flask:

- Styling: CSS được sử dụng để tạo kiểu cho các thành phần HTML, làm cho trang web trở nên hấp dẫn và thân thiện hơn với người dùng.
- Tách biệt nội dung và trình bày: CSS giúp tách biệt nội dung (HTML) và trình bày (CSS), giúp việc quản lý và cập nhật trang web dễ dàng hơn[3].



Hình 4: CSS

### 2.4.4. JavaScript.

JavaScript là ngôn ngữ lập trình phía client, cho phép tạo các tương tác động trên trang web. Nó có thể thay đổi nội dung HTML, CSS và xử lý các sự kiện như nhấn nút, di chuột và nhập liệu.

Vai trò của JavaScript trong Flask:

Tương tác động: JavaScript cho phép thêm các tính năng tương tác như xử lý form, cập nhật nội dung động và phản hồi sự kiện của người dùng mà không cần tải lại trang.

Giao tiếp với server: JavaScript, thông qua AJAX (Asynchronous JavaScript and XML), có thể gửi và nhận dữ liệu từ server mà không cần tải lại trang, giúp cải thiện trải nghiệm người dùng[3].



*Hình 5: JavaScript*

### Kết luận

HTML, CSS và JavaScript là những công nghệ cần thiết để xây dựng các ứng dụng web hiện đại. HTML cung cấp cấu trúc cơ bản cho trang web, CSS định hình giao diện, và JavaScript thêm tính năng tương tác động. Khi kết hợp với Flask, một framework mạnh mẽ cho Python, bạn có thể tạo ra các ứng dụng web mạnh mẽ, linh hoạt và thân thiện với người dùng. Việc sử dụng các công nghệ này cùng với Flask giúp tạo ra trải nghiệm người dùng tốt hơn và làm cho quá trình phát triển web trở nên hiệu quả và dễ quản lý hơn[3].

## **2.5. Tìm hiểu về SQL Server.**

### **2.5.1. Giới thiệu về SQL Server.**

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) do Microsoft phát triển. Nó được thiết kế để quản lý và lưu trữ dữ liệu, hỗ trợ các hoạt động CRUD (Create, Read, Update, Delete) và cung cấp nhiều tính năng mạnh mẽ để bảo mật, quản lý và phân tích dữ liệu. SQL Server được sử dụng rộng rãi trong các ứng

dụng doanh nghiệp, từ các ứng dụng nhỏ đến các hệ thống quy mô lớn[3].

### **2.5.2. Các thành phần chính của SQL Server.**

- Database Engine: Là thành phần cốt lõi của SQL Server, chịu trách nhiệm quản lý, xử lý và lưu trữ dữ liệu. Nó hỗ trợ các hoạt động giao dịch, khóa, chỉ mục và thực hiện các truy vấn SQL.
- SQL Server Agent: Là một dịch vụ giúp tự động hóa các tác vụ quản lý cơ sở dữ liệu, chẳng hạn như sao lưu, lịch trình công việc, và gửi thông báo.
- SQL Server Integration Services (SSIS): Là một nền tảng để tích hợp dữ liệu và xây dựng các giải pháp ETL (Extract, Transform, Load) nhằm chuyển đổi và tải dữ liệu từ các nguồn khác nhau.
- SQL Server Reporting Services (SSRS): Là một dịch vụ tạo báo cáo và quản lý báo cáo cho các ứng dụng web và di động.
- SQL Server Analysis Services (SSAS): Là một dịch vụ phân tích dữ liệu và xây dựng các mô hình phân tích dữ liệu phức tạp, như các kho dữ liệu OLAP.
- SQL Server Management Studio (SSMS): Là công cụ quản lý SQL Server, cung cấp giao diện đồ họa để tạo và quản lý cơ sở dữ liệu, viết và chạy các truy vấn SQL, và thực hiện các nhiệm vụ quản trị khác[3].

### **2.5.3. Các tính năng chính của SQL Server.**

- Tính nhất quán và an toàn của dữ liệu: SQL Server sử dụng các cơ chế giao dịch ACID (Atomicity, Consistency, Isolation, Durability) để đảm bảo tính toàn vẹn và nhất quán của dữ liệu.
- Hiệu năng cao: SQL Server tối ưu hóa các truy vấn và xử lý dữ liệu nhanh chóng nhờ vào các công cụ quản lý bộ nhớ và chỉ mục hiệu quả.
- Bảo mật: SQL Server cung cấp các tính năng bảo mật như mã hóa dữ liệu, kiểm soát truy cập, và xác thực mạnh mẽ để bảo vệ dữ liệu khỏi các mối đe dọa.
- Khả năng mở rộng: SQL Server có thể mở rộng từ các ứng dụng nhỏ đến các hệ thống doanh nghiệp lớn, hỗ trợ các cơ sở dữ liệu có dung lượng lớn và số lượng người dùng cao.
- Tích hợp tốt với các sản phẩm khác của Microsoft: SQL Server tích hợp tốt với các ứng dụng và dịch vụ khác của Microsoft, chẳng hạn như Azure, .NET, Office, và Power BI[4].

### **2.5.4. Các phiên bản của SQL Server.**

- SQL Server có nhiều phiên bản khác nhau, phù hợp với các nhu cầu sử dụng khác nhau:
- SQL Server Express: Phiên bản miễn phí, nhẹ, phù hợp cho các ứng dụng nhỏ và phát triển.
- SQL Server Standard: Phiên bản có đầy đủ tính năng cơ bản, phù hợp cho các

doanh nghiệp vừa và nhỏ.

- SQL Server Enterprise: Phiên bản cao cấp nhất, bao gồm tất cả các tính năng, phù hợp cho các doanh nghiệp lớn với yêu cầu cao về hiệu năng và khả năng mở rộng.
- SQL Server Developer: Phiên bản có đầy đủ tính năng của phiên bản Enterprise, nhưng được sử dụng cho mục đích phát triển và thử nghiệm, không dùng trong môi trường sản xuất.
- SQL Server Web: Phiên bản tối ưu hóa cho các ứng dụng web, với chi phí thấp và tính năng hạn chế[4].

#### 2.5.5. Cài đặt và sử dụng SQL Serve

- Cài đặt: Quá trình cài đặt SQL Server khá đơn giản với hướng dẫn cụ thể từ Microsoft. Người dùng có thể tùy chỉnh các tính năng và dịch vụ cần thiết trong quá trình cài đặt.
- Quản lý: SQL Server Management Studio (SSMS) là công cụ chính để quản lý SQL Server, cung cấp giao diện người dùng thân thiện để thực hiện các nhiệm vụ quản trị cơ sở dữ liệu.
- Truy vấn dữ liệu: SQL Server sử dụng T-SQL (Transact-SQL) làm ngôn ngữ truy vấn chính. T-SQL mở rộng SQL với các tính năng bổ sung như biến, điều kiện, vòng lặp và xử lý ngoại lệ[4].

#### Kết luận

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu mạnh mẽ và linh hoạt, phù hợp với nhiều loại ứng dụng và nhu cầu kinh doanh khác nhau. Với các tính năng tiên tiến về bảo mật, hiệu năng, và khả năng mở rộng, SQL Server đã trở thành lựa chọn phổ biến cho các doanh nghiệp và tổ chức trên toàn thế giới. Dù bạn là nhà phát triển, quản trị viên cơ sở dữ liệu, hay nhà phân tích dữ liệu, SQL Server cung cấp các công cụ và dịch vụ cần thiết để quản lý và khai thác dữ liệu một cách hiệu quả[4].

## CHƯƠNG III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 3.1. Yêu cầu bài toán.

#### 3.1.1. Yêu cầu bài toán.

- Mục tiêu: Phát triển một website bán hàng sử dụng Flask để cung cấp cho người dùng trải nghiệm mua sắm trực tuyến thuận tiện và thú vị.
- Chức năng chính:
- Hiện thị danh sách sản phẩm.
- Cho phép người dùng xem chi tiết sản phẩm.
- Cho phép người dùng thêm sản phẩm vào giỏ hàng.
- Quản lý giỏ hàng.
- Đặt hàng và thanh toán[5].

#### Đặc điểm:

- Giao diện thân thiện và dễ sử dụng.
- Tích hợp các tính năng tìm kiếm và lọc sản phẩm.
- Tích hợp hệ thống thanh toán trực tuyến an toàn và đáng tin cậy.

#### 3.1.2. Yêu cầu phi chức năng

#### Hiệu suất:

- Website phải có thời gian tải trang nhanh, đáp ứng nhanh chóng yêu cầu của người dùng.
- Hệ thống phải có khả năng mở rộng dễ dàng để đối phó với sự gia tăng lưu lượng truy cập.

#### Bảo mật:

- Bảo vệ thông tin cá nhân của người dùng, bao gồm thông tin thanh toán.
- Áp dụng các biện pháp bảo mật để ngăn chặn các cuộc tấn công từ bên ngoài và bên trong.

#### Tương thích:

- Website phải hoạt động tốt trên các thiết bị di động và các trình duyệt web phổ biến.
- Đảm bảo tương thích với nhiều hệ điều hành và nền tảng.
- Hỗ trợ khách hàng:
- Cung cấp cơ chế liên lạc và hỗ trợ khách hàng hiệu quả, bao gồm hỗ trợ qua email, chat trực tuyến, hoặc điện thoại.
- Tạo kênh phản hồi để người dùng có thể gửi ý kiến và đánh giá về sản phẩm và dịch vụ. [5].

### 3.2. Yêu cầu hệ thống.

- Phân tích làm rõ quy trình nghiệp vụ của ứng dụng bán đồ uống bao gồm các chức năng :
- Quản lý tài khoản: Người dùng có thể Đăng nhập, Đăng xuất, Đổi mật khẩu,

Đăng ký tài khoản (user), Sửa tài khoản, Xóa tài khoản, Thêm tài khoản và Phân quyền

- Quản lý giao dịch: bao gồm xem danh sách giao dịch, hủy giao dịch, xử lý giao dịch.
- Quản lý danh mục: bao gồm xem danh sách danh mục, thêm danh mục, sửa danh mục, xóa danh mục.
- Quản lý sản phẩm: bao gồm xem danh sách sản phẩm, thêm sản phẩm, sửa sản phẩm, xóa sản phẩm.
- Thống kê: bao gồm thống kê doanh thu, thống kê dữ liệu.
- Quản lý thành viên hỗ trợ: bao gồm xem danh sách thành viên hỗ trợ, thêm thành viên, sửa thành viên, xóa thành viên.
- Quản lý tin tức: bao gồm xem danh sách tin tức, thêm tin tức, sửa tin tức, xóa tin tức
- Giỏ hàng
- Thanh toán hóa đơn [5]

### **3.3. Thiết kế hệ thống.**

#### **3.3.1. Kiến trúc tổng quan của hệ thống:**

Frontend (Giao diện người dùng):

- Sử dụng HTML, CSS và JavaScript để tạo giao diện người dùng thân thiện và hấp dẫn.
- Sử dụng Jinja2 template engine trong Flask để tạo các trang web động.
- Tích hợp các thư viện và framework như Bootstrap để tối ưu hóa trải nghiệm người dùng trên các thiết bị di động.

Backend (Phần xử lý logic):

- Sử dụng Flask để xây dựng các API để xử lý yêu cầu từ phía client và truy xuất dữ liệu từ cơ sở dữ liệu.
- Sử dụng SQLAlchemy để tương tác với cơ sở dữ liệu SQL Server.

Cơ sở dữ liệu (Database):

- Sử dụng SQL Server để lưu trữ dữ liệu về sản phẩm, người dùng, đơn hàng và các thông tin liên quan.
- Thiết kế cơ sở dữ liệu để phù hợp với nhu cầu của ứng dụng, bao gồm các bảng như Product, User, Order, và các quan hệ giữa chúng.

### 3.3.2 Xác Định Actor Và Usecase

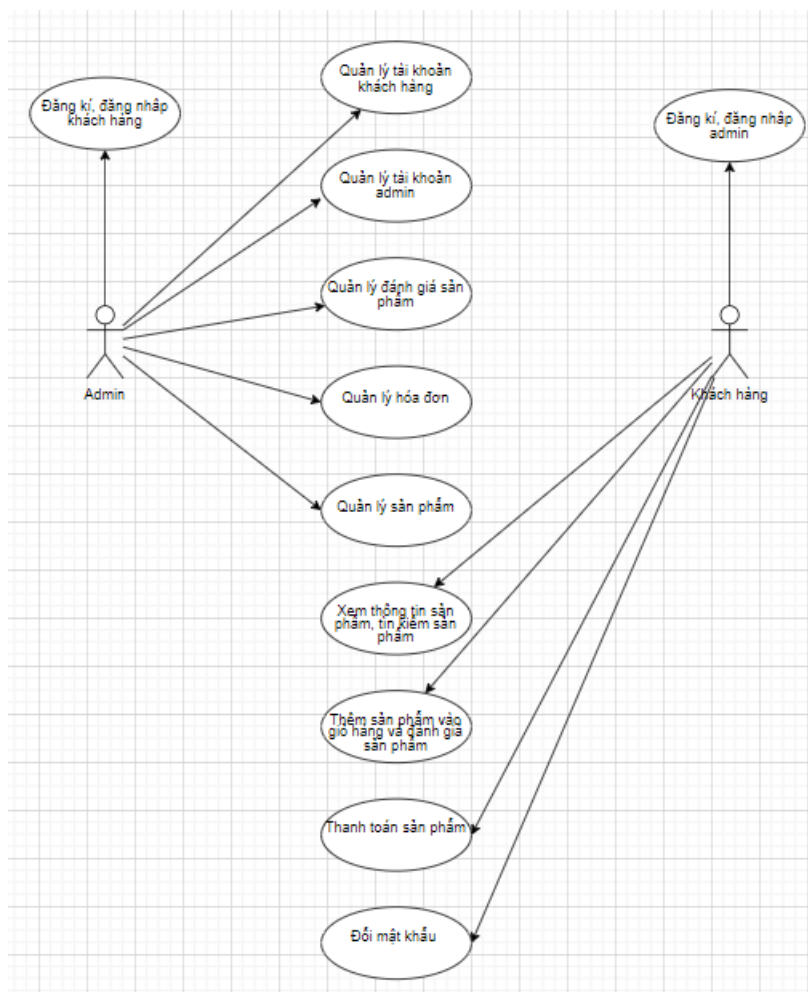
S T T	ACTOR	USECASE
1	Admin	<ul style="list-style-type: none"> <li>- Quản lý sản phẩm: Thêm, sửa, xóa sản phẩm</li> <li>- Quản lý tin tức: Thêm, sửa, xóa tin tức</li> <li>- Quản lý tài khoản: Thêm, sửa, xóa và phân quyền tài khoản</li> <li>- Quản lý danh mục: Thêm, sửa, xóa danh mục</li> <li>- Quản lý kho hàng: Thêm sửa xóa các đơn hàng</li> <li>- Quản lý nhân viên viên hỗ trợ: Thêm, sửa, xóa nhân viên hỗ trợ</li> <li>- Quản lý khách hàng: Thêm, sửa, xóa thông tin khách hàng.</li> <li>- Quản lý danh mục bài viết: Thêm, sửa, xóa bài viết.</li> <li>- Đăng nhập/Đăng xuất</li> <li>- Quản lý sự kiện: Thêm, sửa, xóa các sự kiện giảm giá của cửa hàng.</li> </ul>
2	Nhân Viên	<ul style="list-style-type: none"> <li>- Quản lý giao dịch: xem, xóa giao dịch và xử lý đơn hàng</li> <li>- Quản lý danh mục: Thêm, sửa, xóa danh mục</li> <li>- Quản lý sản phẩm: Thêm, sửa, xóa sản phẩm</li> </ul>

		<ul style="list-style-type: none"> <li>- Quản lý tin tức, sự kiện: Thêm, sửa, xóa tin tức</li> <li>- Quản lý tài khoản cá nhân <ul style="list-style-type: none"> <li>+ Thay đổi thông tin tài khoản</li> <li>+ Xem tài khoản</li> </ul> </li> <li>- Quản lý nhân viên hỗ trợ: Thêm, sửa, xóa nhân viên hỗ trợ</li> <li>- Đăng nhập/Đăng xuất</li> <li>- Thống kê <ul style="list-style-type: none"> <li>+ Thống kê doanh thu</li> <li>+ Thống kê dữ liệu</li> </ul> </li> </ul>
<b>3</b>	<b>Khách Hàng</b>	<ul style="list-style-type: none"> <li>- Đăng ký tài khoản</li> <li>- Quản lý giỏ hàng: Thêm, xóa sản phẩm</li> <li>- Tìm kiếm</li> <li>- Đặt hàng</li> <li>- Quản lý tài khoản cá nhân <ul style="list-style-type: none"> <li>+ Đổi mật khẩu</li> <li>+ Thay đổi thông tin</li> </ul> </li> <li>- Đăng nhập/Đăng xuất</li> </ul>



### 3.4. Biểu Đồ Ca Sử Dụng (Usecase Diagram).

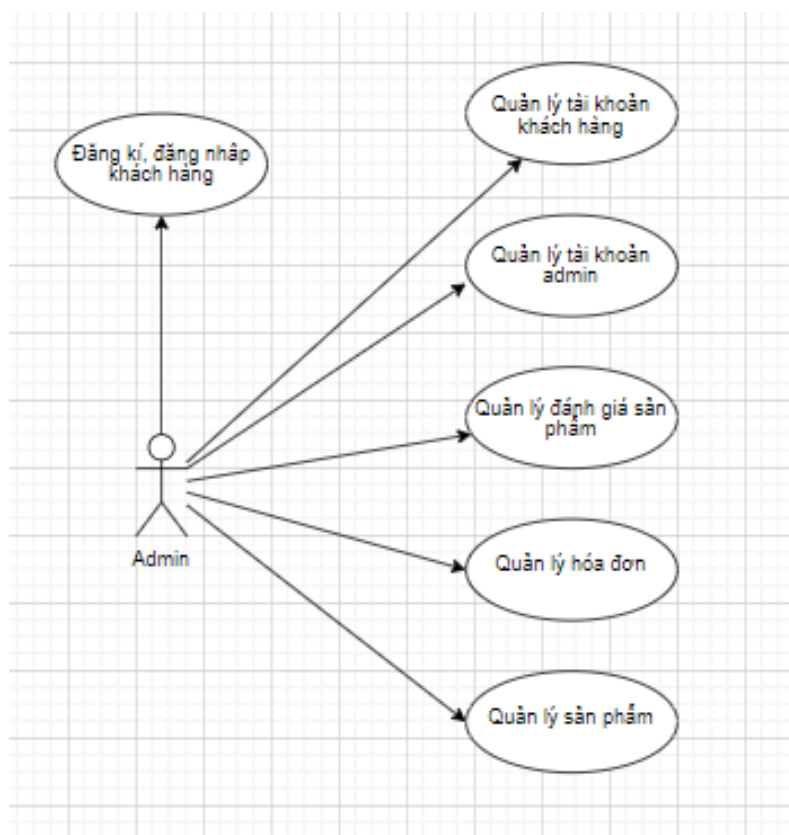
#### 3.4.1. Usecase tổng quát.



Hình 6: Usecase tổng quát

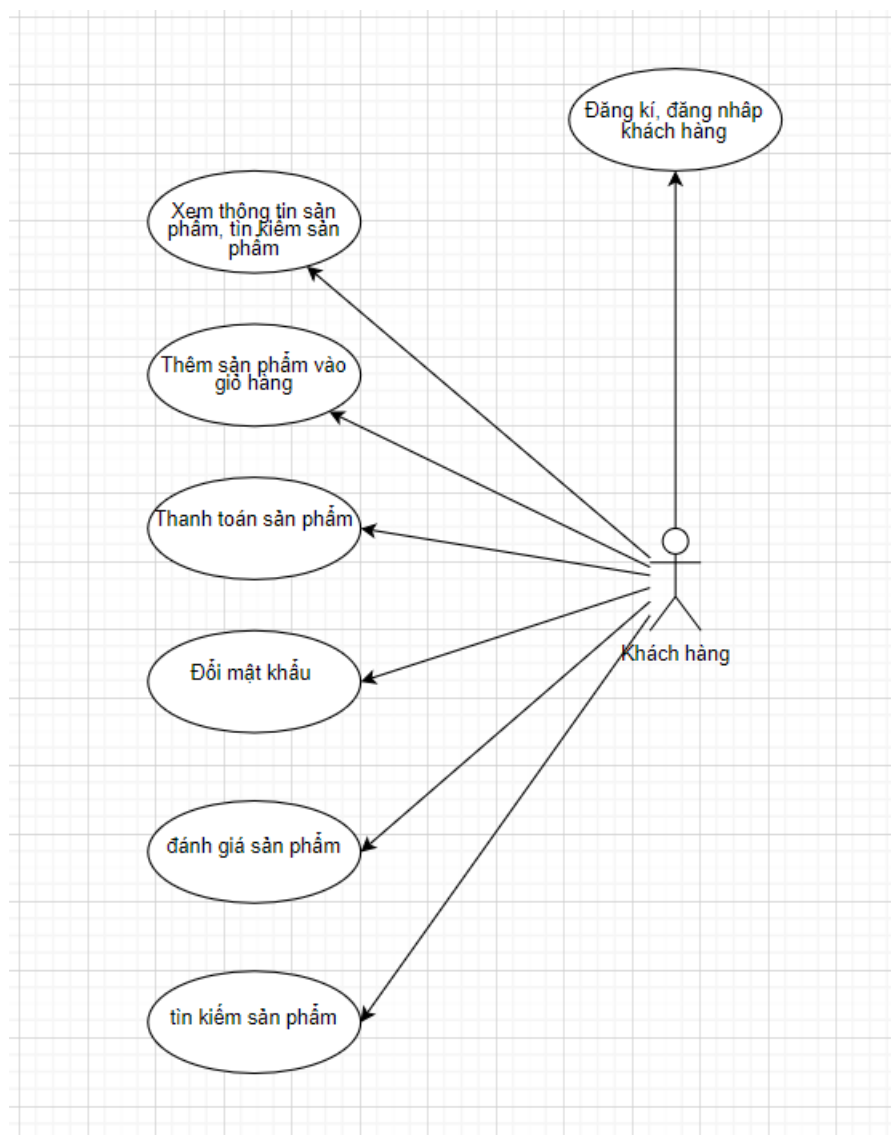
### 3.4.2. Usecase dành cho các tác nhân.

#### a. Usecase dành cho tác nhân ‘Admin’



Hình 7: Usecase Admin

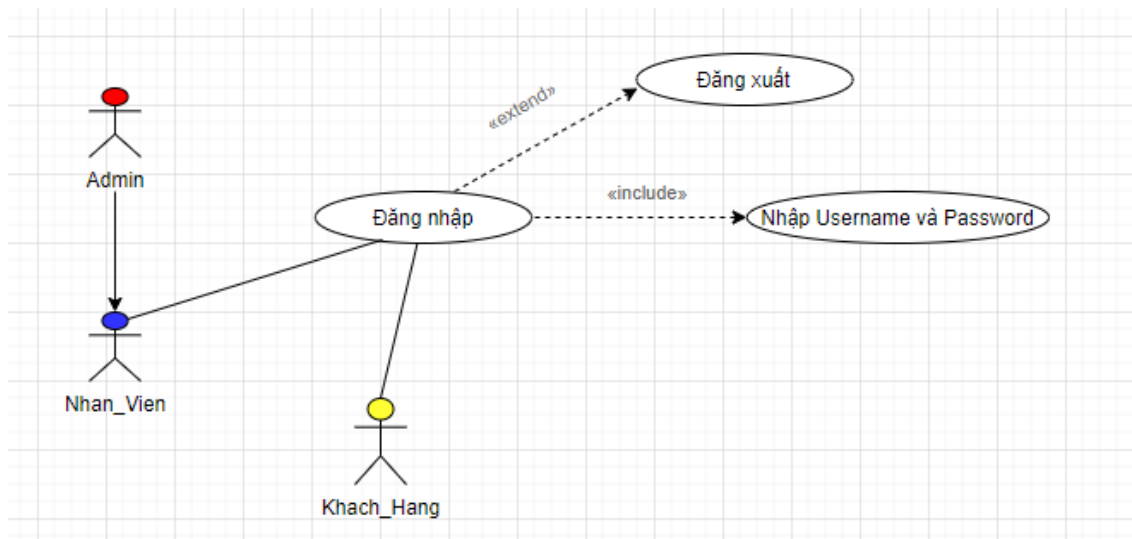
***b. Usecase dành cho tác nhân ‘Khách\_Hàng’:***



*Hình 8: Usecase khách hàng*

### 3.4.3. Các biểu đồ usecase quản lý chức năng.

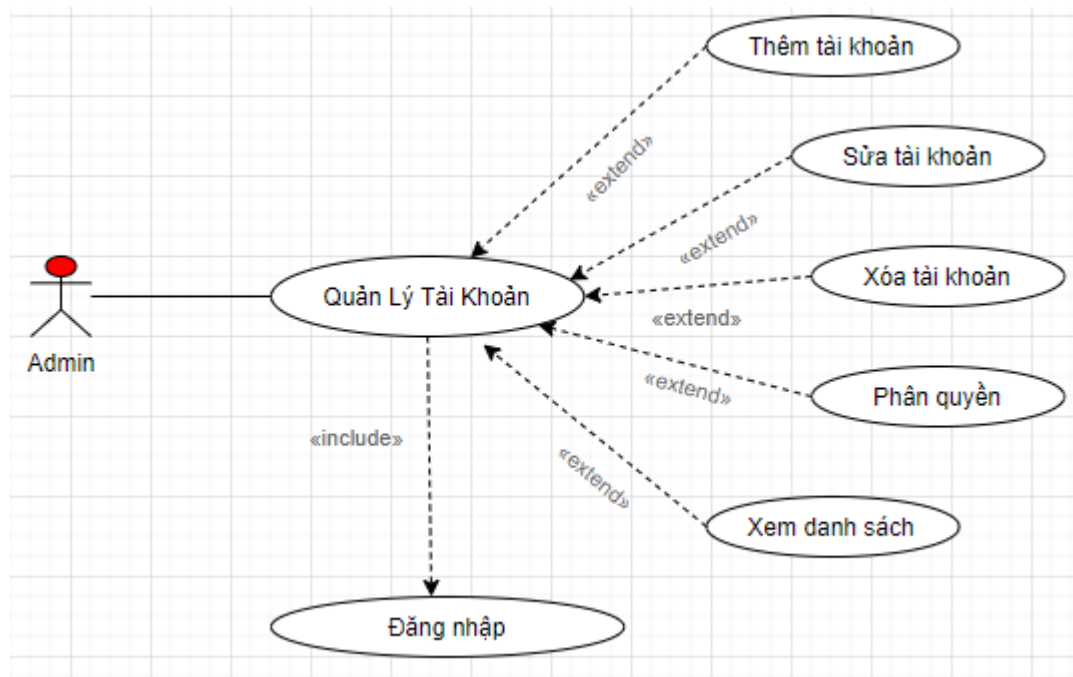
Biểu đồ usecase chi tiết



Hình 9: Biểu đồ chi tiết usecase 'Đăng nhập'

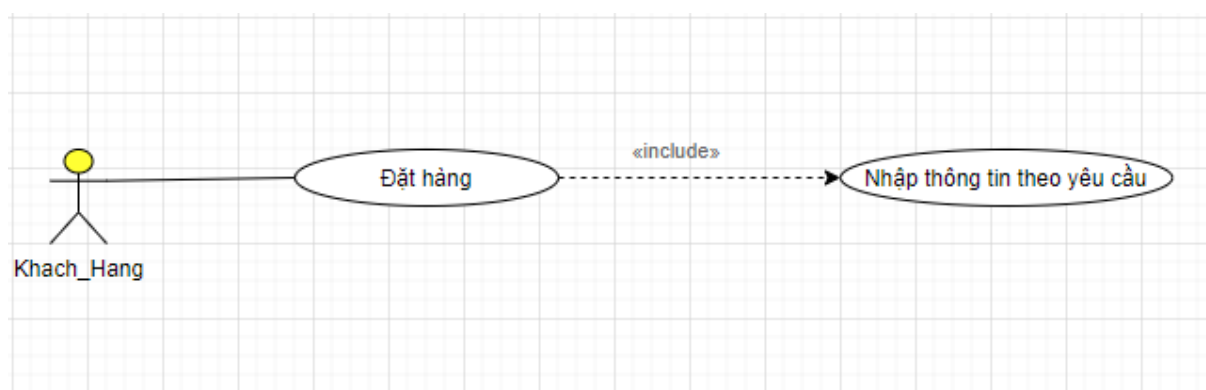
<b>Tên UC</b>	<b>Đăng nhập</b>
<b>Tác nhân</b>	Quản lý, khách hàng, nhân viên
<b>Mục đích</b>	Đăng nhập hệ thống
<b>Mô tả</b>	Người sử dụng đăng nhập vào hệ thống thông qua “SDT or Email” và “Password”
<b>Tiền điều kiện</b>	Đã có tài khoản
<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Người sử dụng chọn chức năng “Đăng nhập”</li> <li>2. Hệ thống hiển thị form “Đăng nhập”</li> <li>3. Người sử dụng nhập “SDT or Email” và “Password”, gửi thông tin đến hệ thống</li> <li>4. Gửi đến hệ thống CSDL kiểm tra. Nếu đúng thì cho phép truy cập, và sử dụng. Nếu sai hệ thống CSDL thông báo lỗi và yêu cầu người dùng nhập lại thông tin</li> </ol>
<b>Luồng sự kiện phụ</b>	<ol style="list-style-type: none"> <li>1. Nhập sai định dạng “SDT or Email” hoặc “Password”, yêu cầu nhập lại</li> <li>2. Nhập thiếu thông tin Email hoặc SDT và Password, yêu cầu nhập đầy đủ</li> <li>3. Bỏ trống thông tin Email or SDT, Password, yêu cầu nhập đầy đủ</li> <li>4. Nhập sai Email or SDT, Password, đăng nhập thất bại</li> </ol>

***Biểu đồ chi tiết usecase ‘Quản lý tài khoản’***



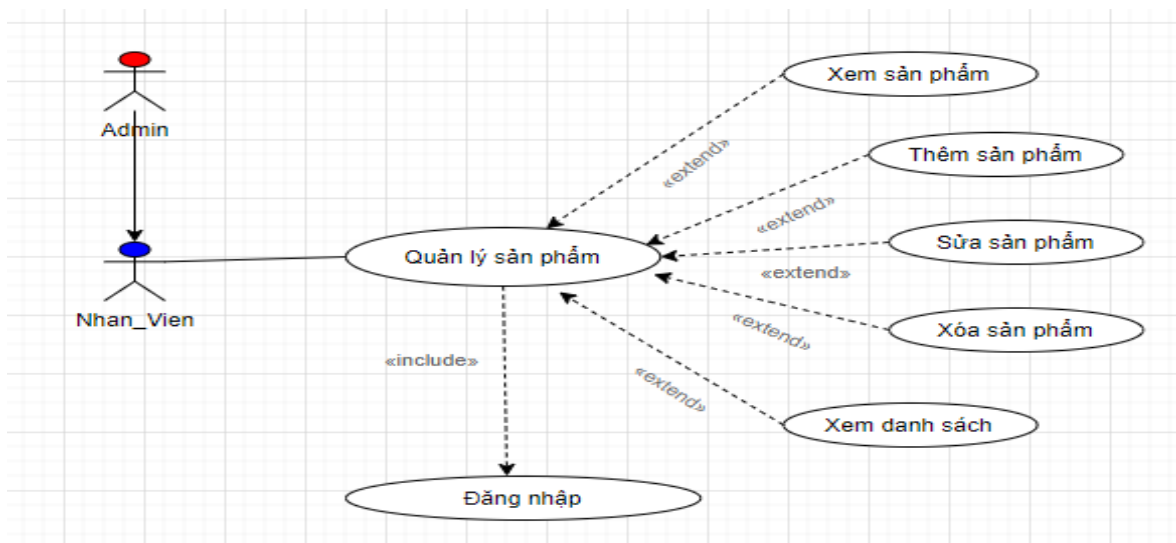
*Hình 10: Biểu đồ chi tiết usecase ‘Quản lý tài khoản’*

***Biểu đồ chi tiết usecase ‘Đặt hàng’***



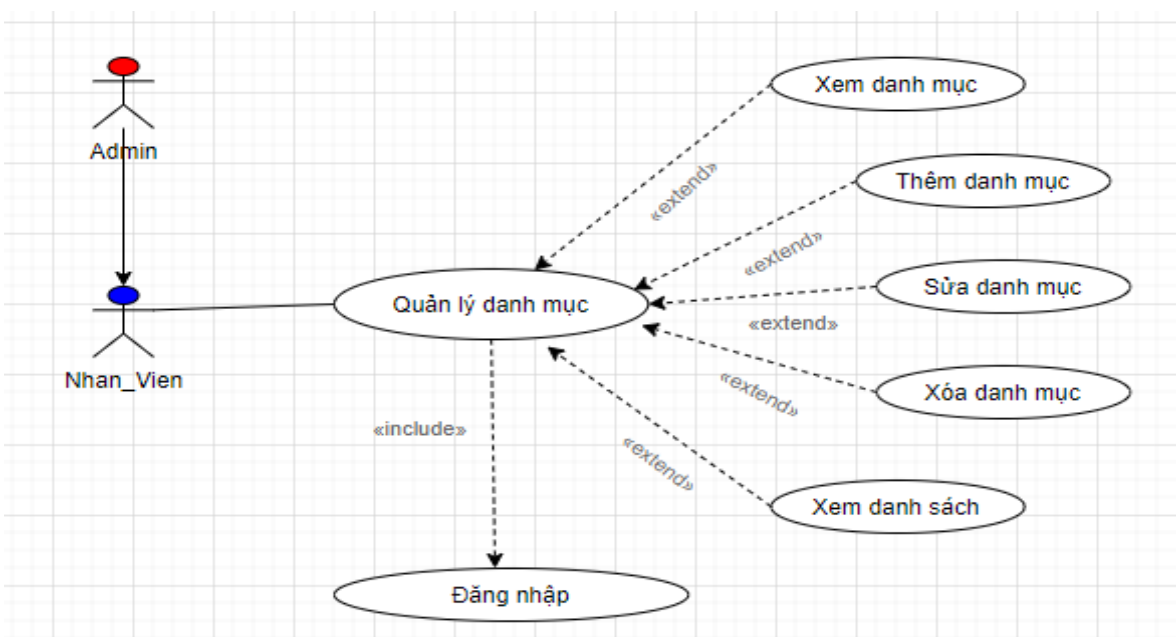
*Hình 11: Biểu đồ chi tiết usecase ‘Đặt hàng’*

### ***Biểu đồ chi tiết usecase ‘Quản lý sản phẩm’***



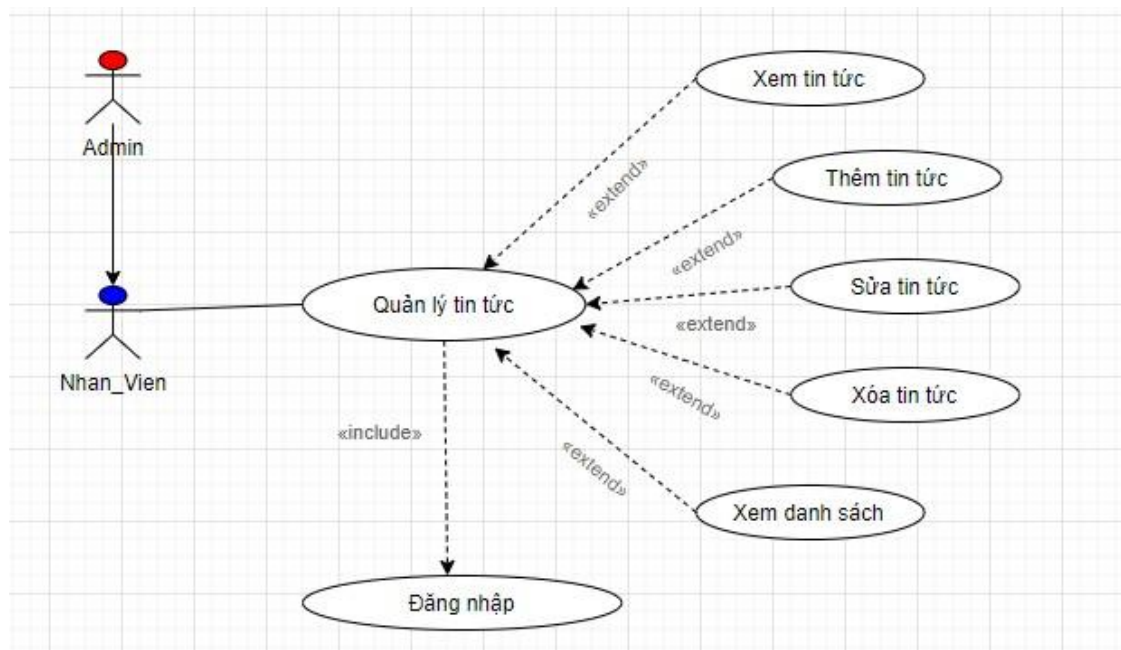
*Hình 12: Biểu đồ chi tiết usecase ‘Quản lý sản phẩm’*

### ***Biểu đồ chi tiết usecase ‘Quản lý danh mục’***



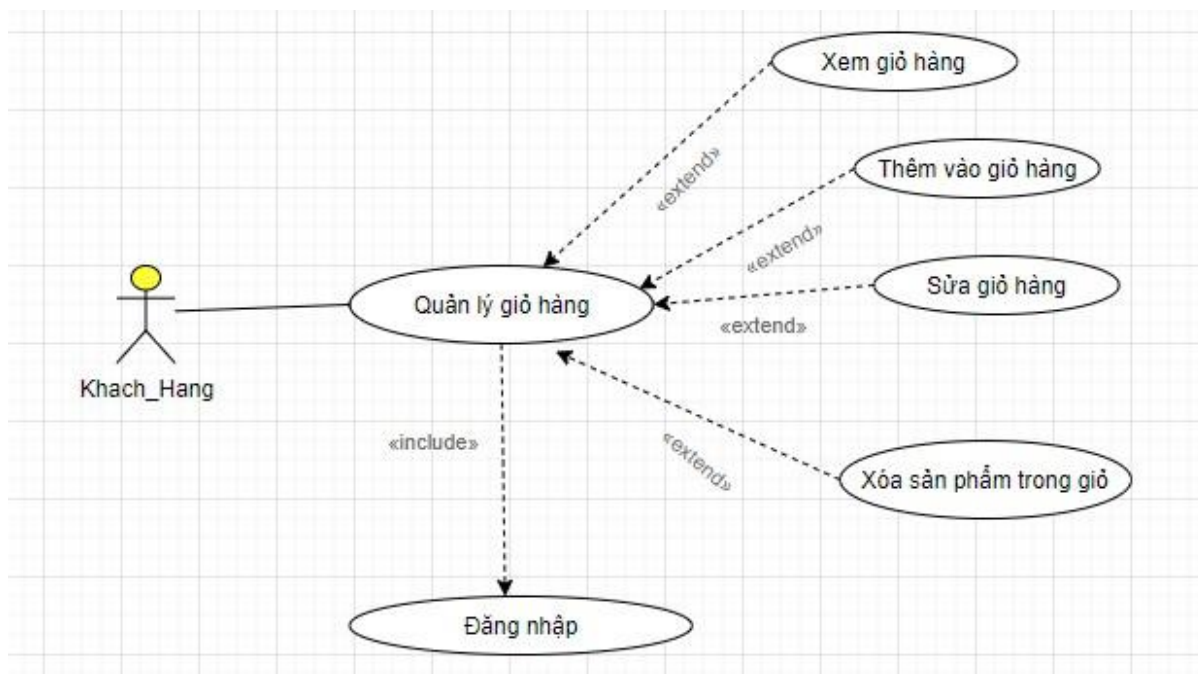
*Hình 13: Biểu đồ chi tiết usecase ‘Quản lý danh mục’*

### ***Biểu đồ chi tiết usecase ‘Quản lý tin tức’***



*Hình 14: Biểu đồ chi tiết usecase ‘Quản lý tin tức’*

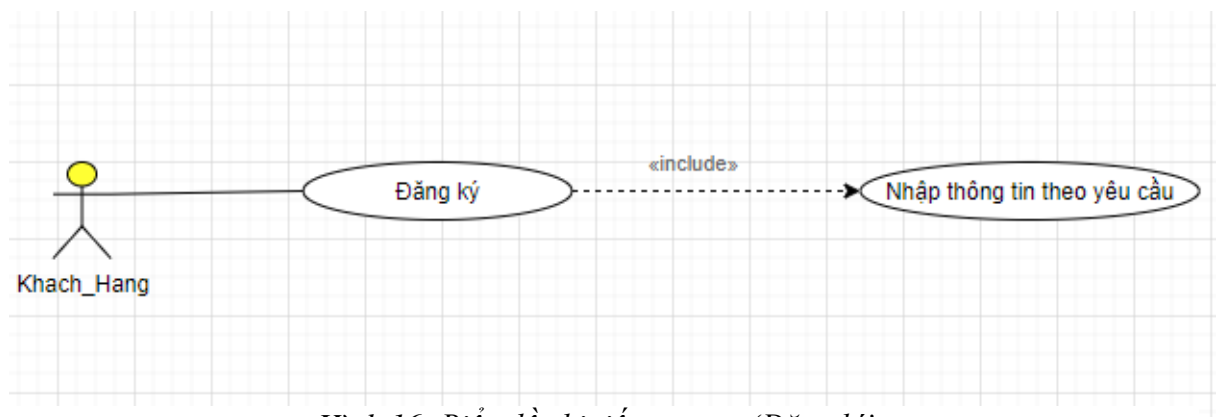
### ***Biểu đồ chi tiết usecase ‘Quản lý giỏ hàng’***



*Hình 15: Biểu đồ chi tiết usecase ‘Quản lý giỏ hàng’*

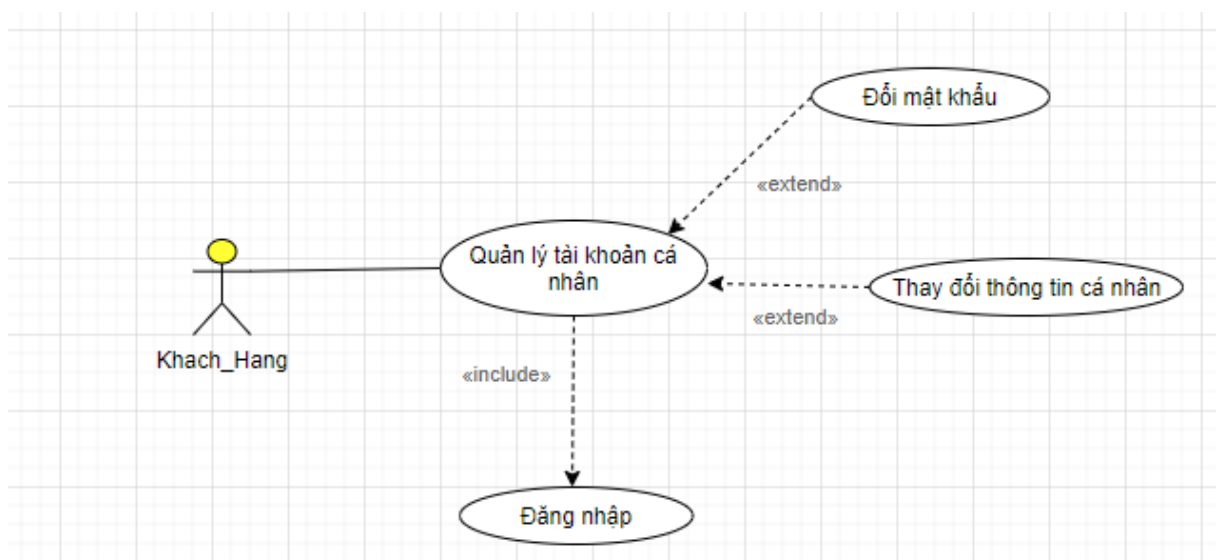


### ***Biểu đồ chi tiết đăng kí usecase***



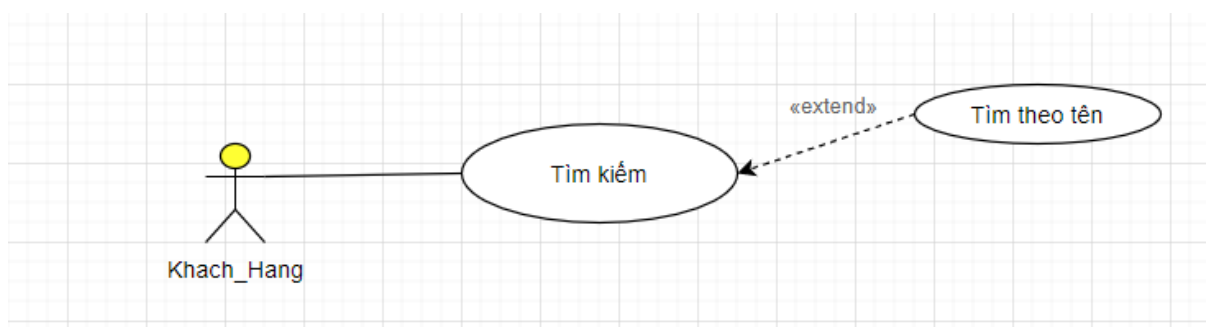
Hình 16: Biểu đồ chi tiết usecase 'Đăng kí'

### ***Biểu đồ chi tiết usecase 'Quản lý tài khoản cá nhân'***



Hình 17: Biểu đồ chi tiết usecase 'Quản lý tài khoản cá nhân'

### ***Biểu đồ chi tiết chức năng tìm kiếm***

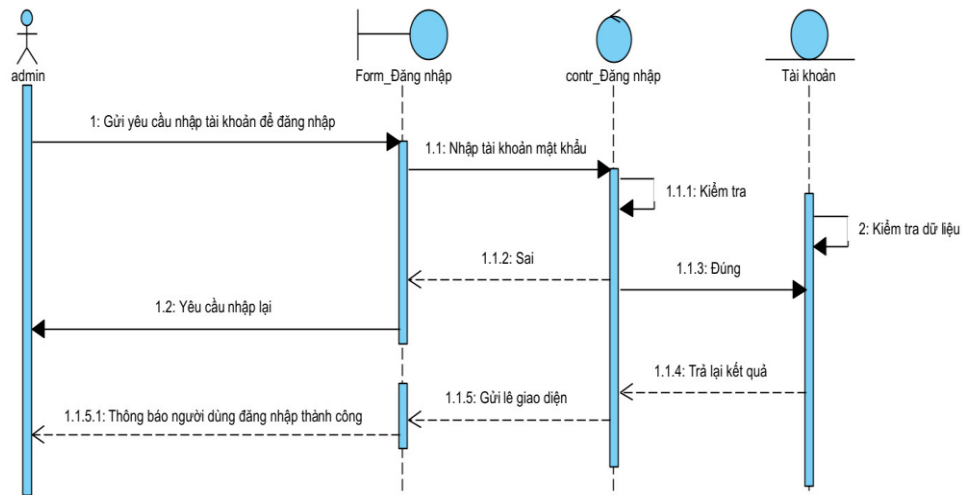


Hình 18: Biểu đồ chi tiết chức năng tìm kiếm

### 3.5. Đặc Tả Các Use Case

#### Đặc tả Use case Đăng nhập

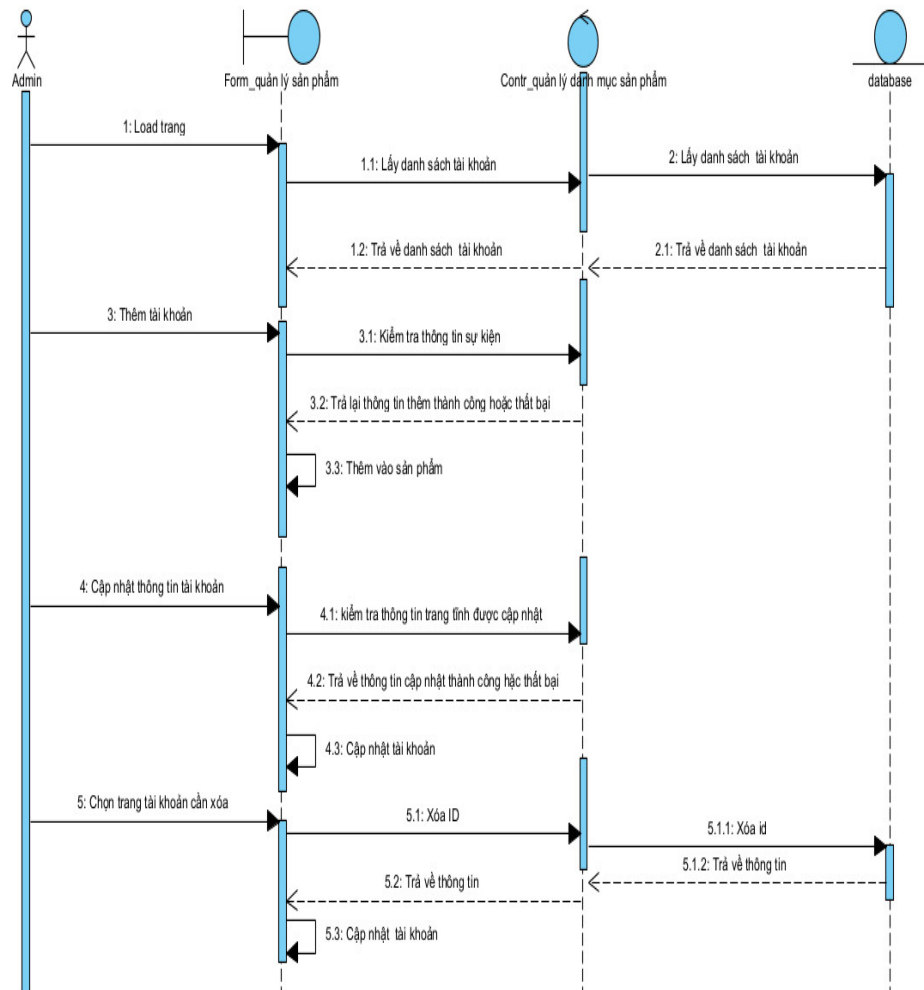
sđ [Biểu đồ tuần tự đăng nhập]



Hình 19: Đặc tả Use case Đăng nhập

## Đặc tả Use case Quản lý sản phẩm

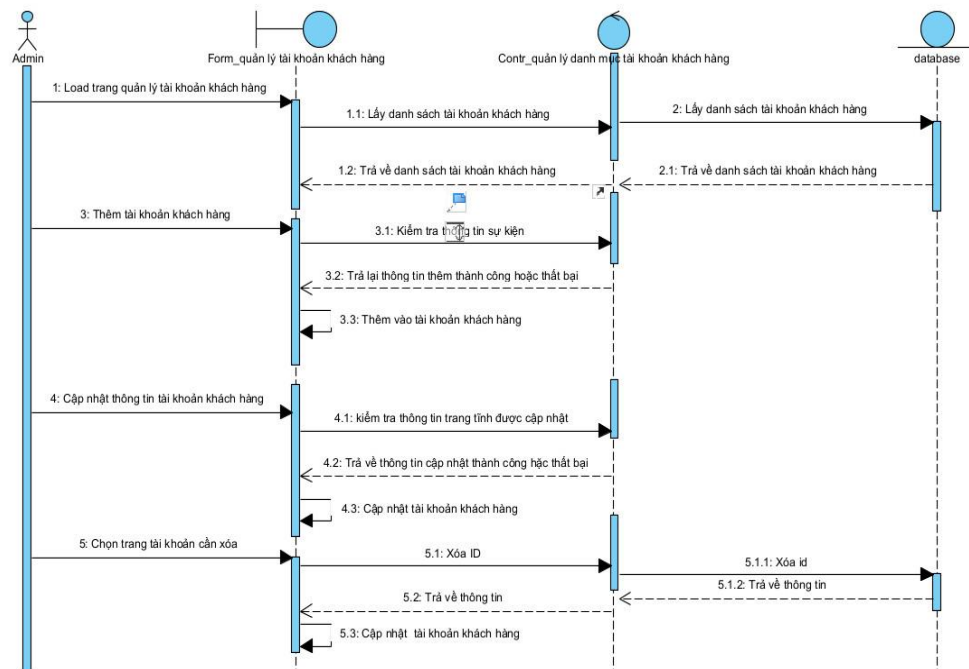
sd [Biểu đồ tuần tự quản lý tài khoản]



Hình 20: Đặc tả Use case Quản lý sản phẩm

## Đặc tả Use case quản lý khách hàng.

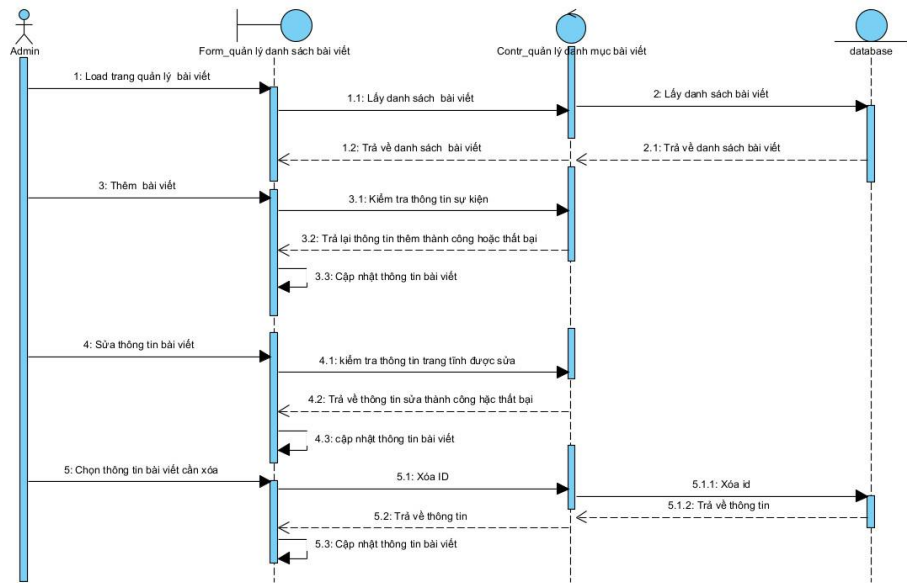
sd [Biểu đồ tuần tự quản lý tài khoản khách hàng và người dùng]



Hình 21: Đặc tả Use case quản lý khách hàng.

**Đặc tả Use case quản lý danh mục bài viết.**

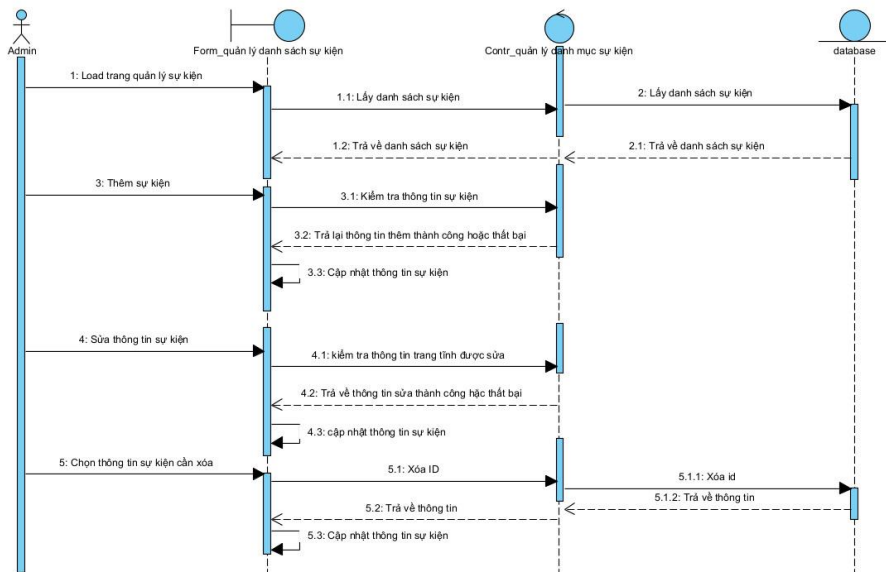
sd [Biểu đồ tuần tự quản lý các bài viết]



Hình 22: Đặc tả Use case quản lý danh mục bài viết.

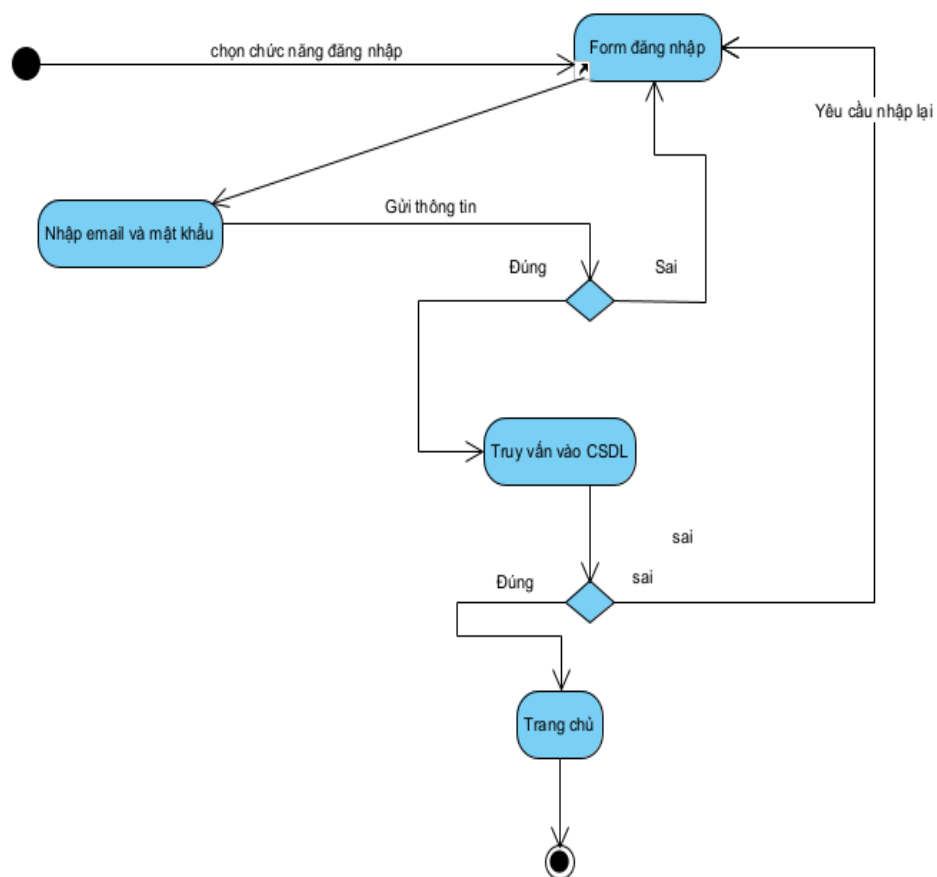
**Đặc tả Use case quản lý danh mục bài viết.**

sd [Biểu đồ tuần tự quản lý các sự kiện]



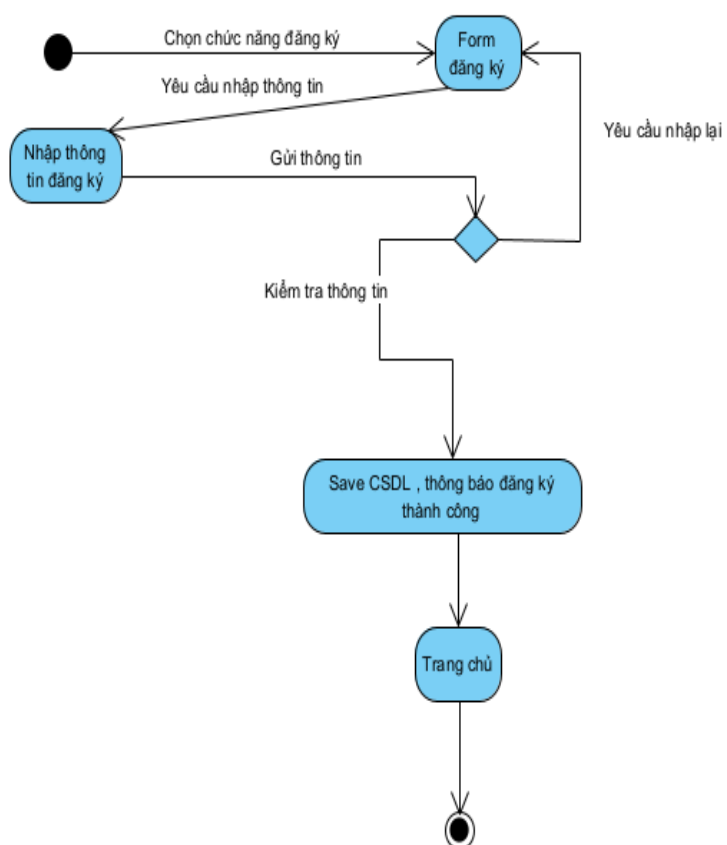
Hình 23: . Đặc tả Use case quản lý danh mục bài viết.

### ***Biểu đồ trình tự đăng nhập***



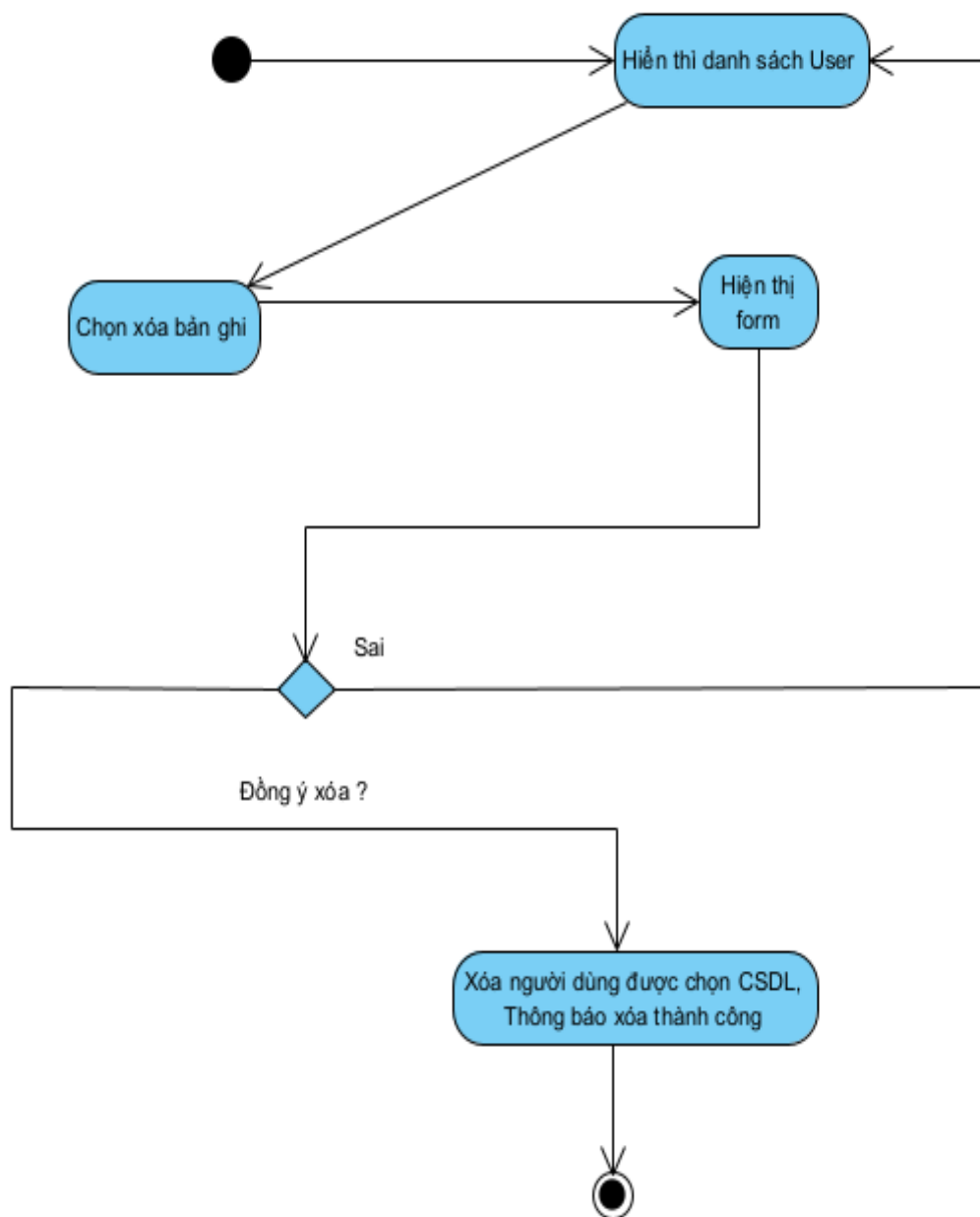
*Hình 24: Biểu đồ tuần tự đăng nhập*

### *Biểu đồ tuần tự đăng kí*



*Hình 25: Biểu đồ tuần tự đăng kí*

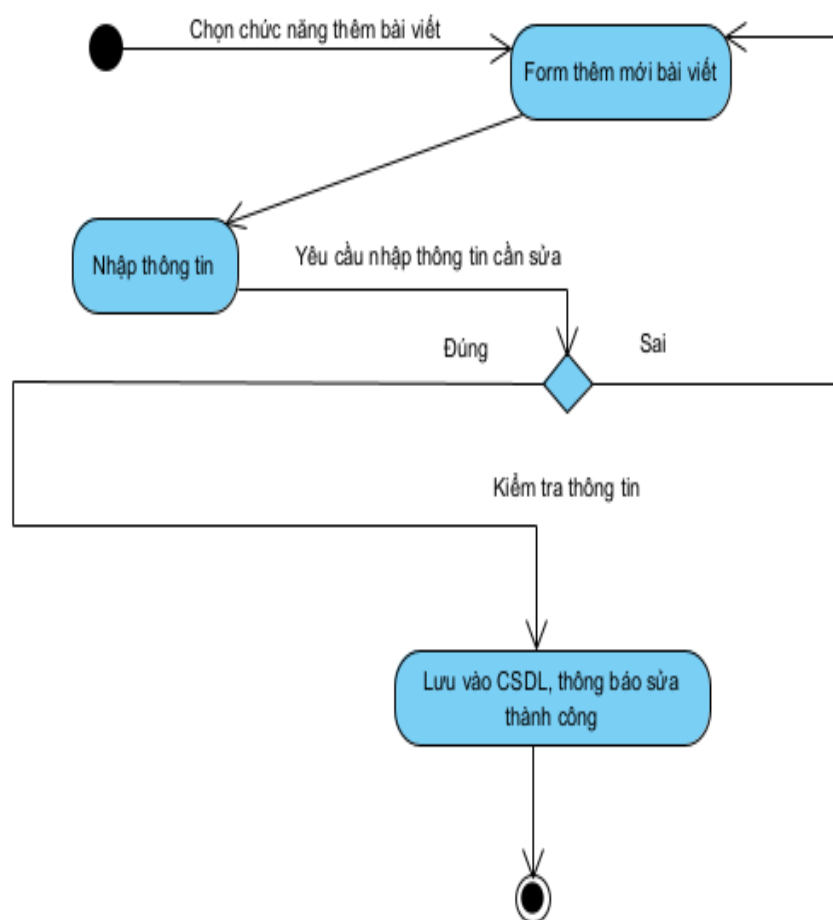
*Biểu đồ hoạt động chức năng delete người dùng*



*Hình 26: Biểu đồ hoạt động chức năng delete người dùng*

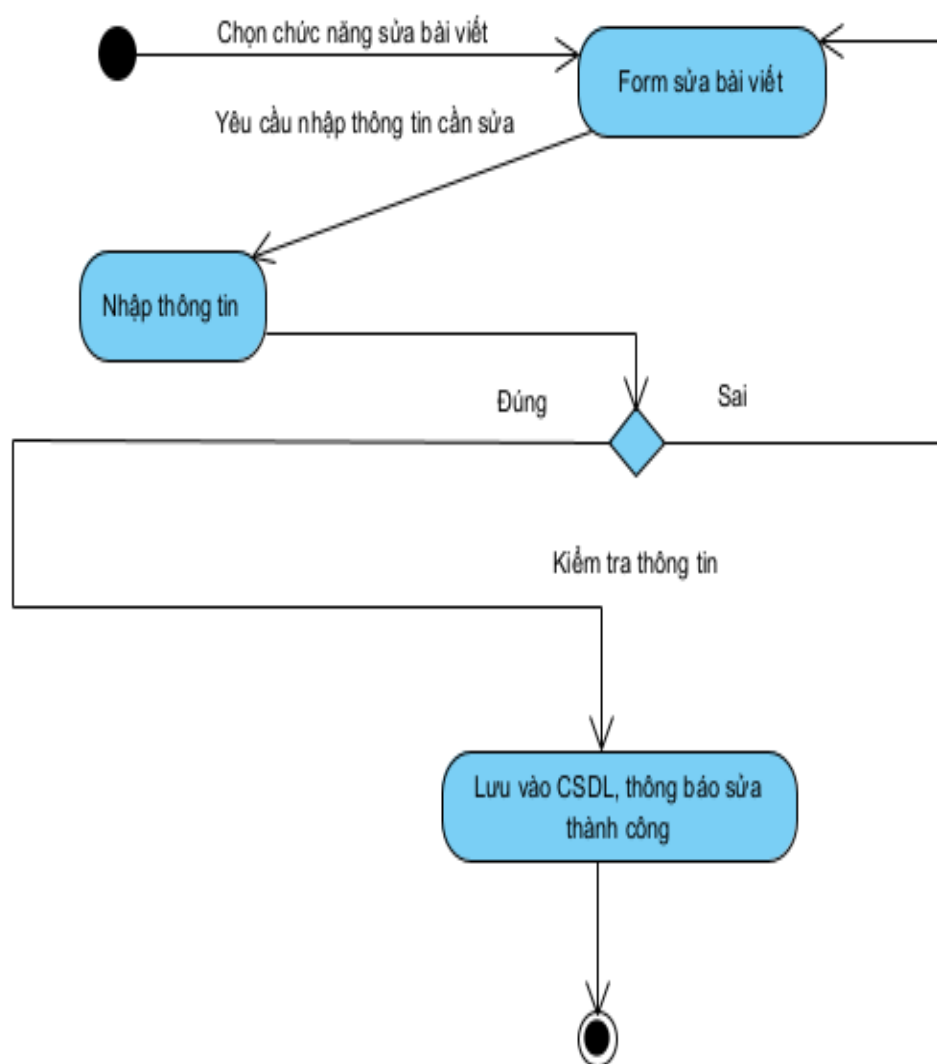


### ***Biểu đồ chức năng thêm mới bài viết***



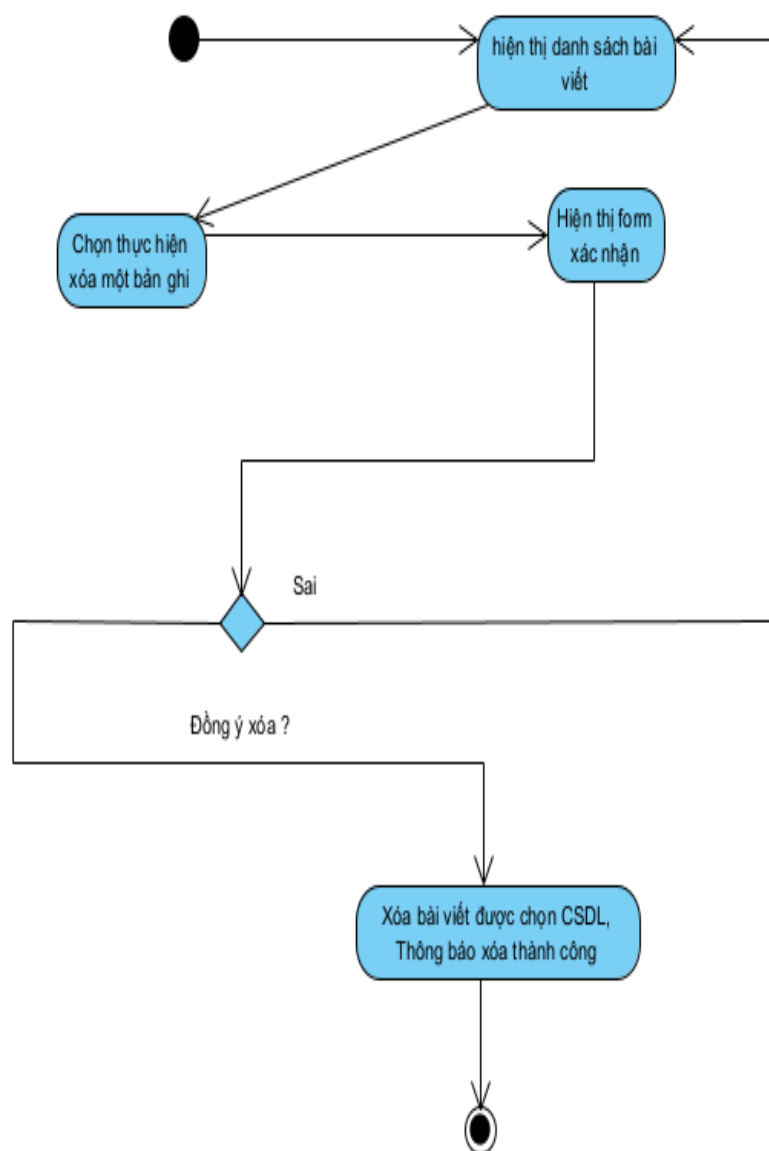
*Hình 27: Biểu đồ chức năng thêm mới bài viết*

### ***Biểu đồ chức năng sửa mới bài viết***



*Hình 28: Biểu đồ chức năng sửa mới bài viết*

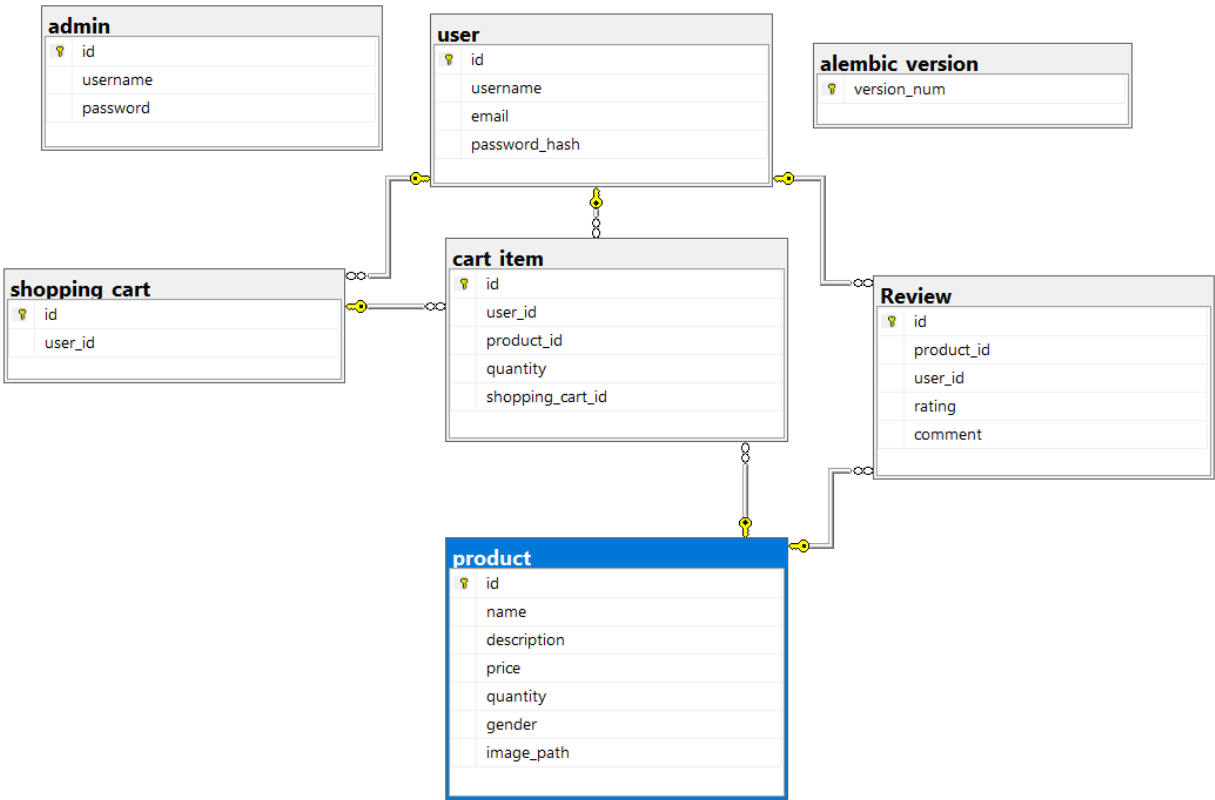
### *Biểu đồ chức năng xóa bài viết*



*Hình 29: Biểu đồ chức năng xóa bài viết*

### 3.6. Biểu Đồ Lớp và ER

#### 3.6.1. Biểu Đồ Lớp (Class Diagram)



Hình 30: Biểu đồ lớp (class diagram)

### 3.6.2. Biểu ER

Mô hình Product và CartItem:

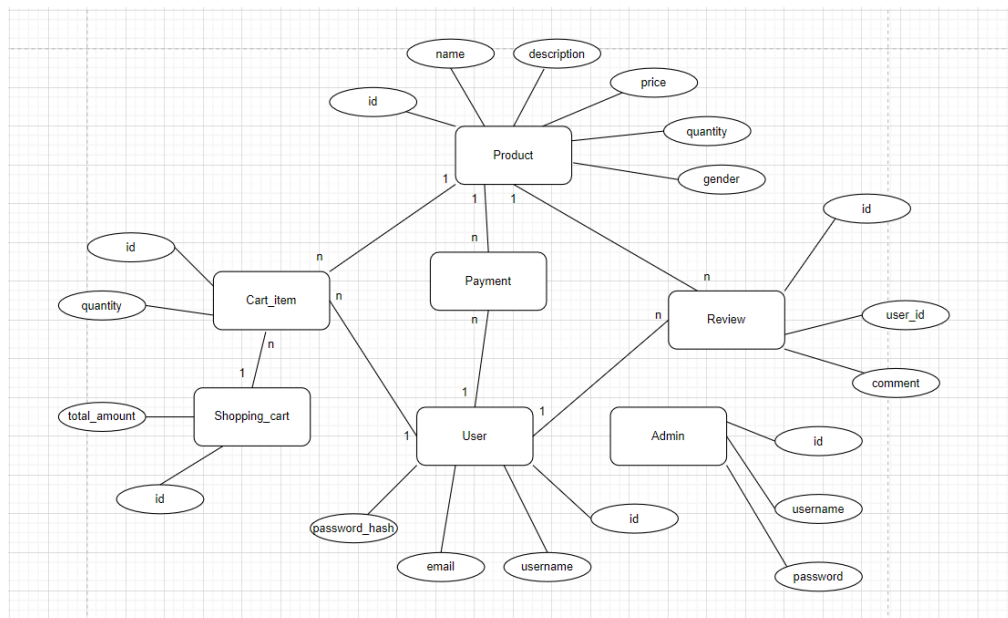
- Product: Đại diện cho các sản phẩm trong cửa hàng.
- CartItem: Đại diện cho các mục trong giỏ hàng, mỗi mục liên kết với một sản phẩm.
- Mỗi quan hệ: Mỗi sản phẩm có thể có nhiều mục trong giỏ hàng. Quan hệ được thiết lập thông qua khóa ngoại product\_id trong CartItem đến bảng Product.

Mô hình User và Payment:

- User: Đại diện cho người dùng trong hệ thống.
- Payment: Đại diện cho các thanh toán được thực hiện bởi người dùng.
- Mỗi quan hệ: Mỗi người dùng có thể có nhiều thanh toán. Quan hệ được thiết lập qua khóa ngoại user\_id trong Payment đến bảng User.

Mô hình Product và Review:

- Product: Đại diện cho các sản phẩm trong cửa hàng.
- Review: Đại diện cho các đánh giá của người dùng đối với sản phẩm.
- Mỗi quan hệ: Mỗi sản phẩm có thể có nhiều đánh giá từ người dùng. Quan hệ được thiết lập qua khóa ngoại product\_id trong Review đến bảng Product[5].



### 3.7. Mô hình cơ sở dữ liệu

- **Bảng user:**

**USER** (id, UserName, email,password\_hash)

Thuộc tính	Kiểu dữ liệu	Độ rộng	Ý nghĩa
Id	Int	Int	Khóa chính bảng account
Username	Nvarchar	24	Tên tài khoản
Email	Nvarchar	24	Gmail người dùng
Password_ hash	Varchar	8	Mật khẩu người dùng

*Bảng 1: Account*

- **Bảng admin:**

**ADMIN** (id, UserName, email,password)

Thuộc tính	Kiểu dữ liệu	Độ rộng	Ý nghĩa
ID	int	int	Khóa chính bảng Account
UserName	Nvarchar	240	Tên tài khoản
email	Nvarchar	240	Gmail admin
Password	Nvarchar	50	Mật khẩu người dùng

*Bảng 2: CSDL bảng admin*

## Bảng 1. 28 Bảng Khách hàng

### - Bảng Sản phẩm:

**PRODUCT** (Id, name, description, price, quantity, gender, image\_path)

Thuộc tính	Kiểu dữ liệu	Độ rộng	Ý nghĩa
Id	Int	Int	Khóa chính bảng Sản phẩm
Name	Nvarchar	200	Tên sản phẩm
Description	Nvarchar	200	Thông tin sản phẩm
Price	Float	Float	Giá sản phẩm
Quantity	Float	Float	Số lượng sản phẩm
Gender	Nvarchar	200	Loại sản phẩm ( nam or nữ)
Image_path	Nvarchar	200	Ảnh sản phẩm

Bảng 3: CSDL bảng product

### - Bảng Hóa Đơn:

**CART\_ITEM** (id, user\_id, product\_id, quantity, shopping\_cart\_id)

Thuộc tính	Kiểu dữ liệu	Độ rộng	Ý nghĩa
Id	Int	Int	Khóa chính bảng hóa đơn
User_id	Int	Int	Khóa phụ bảng user
Product_id	Int	Int	Khóa phụ bảng product
Quantity	Nvarchar	240	Số lượng sản phẩm
Shopping_cart_id	Int	Int	Khóa phụ bảng shopping_cart

Bảng 4: CSDL bảng cart\_item

- **Bảng Hóa Đơn:**

**SHOPPING\_CART** (id, user\_id)

Thuộc tính	Kiểu dữ liệu	Độ rộng	Ý nghĩa
Id	Int	Int	Khóa chính bảng hóa đơn
User_id	Int	Int	Khóa phụ bảng user

*Bảng 5: CSDL bảng shopping\_cart*

**Bảng review:**

**REVIEW** (id, product\_id, user\_id, rating, comment)

Thuộc tính	Kiểu dữ liệu	Độ rộng	Ý nghĩa
Id	Int	Int	Khóa chính bảng danh mục
Product_id	Int	Int	Id bảng product khóa phụ
User_id	Int	Int	Id bảng user khóa phụ
Rating	Int	Int	Đánh giá sản phẩm
Comment	Nvarchar	240	Bình luận sản phẩm

*Bảng 6: CSDL bảng REVIEW*



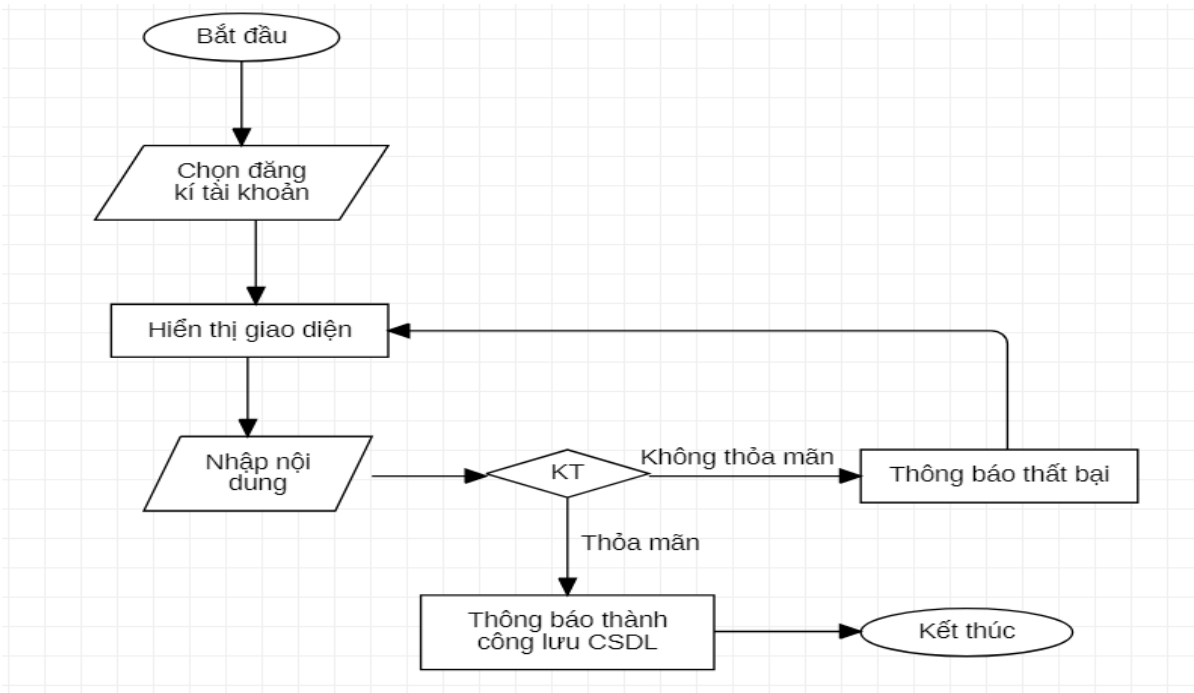
- **Bảng hóa đơn:**

**Payment** (Id, Product\_id, User\_id, Amount, Payment\_status, Created\_at, Approved, Completed, Address, Quantity, Phone\_number)

Thuộc tính	Kiểu dữ liệu	Độ rộng	Ý nghĩa
Id	Varchar	24	Khóa chính bảng tin tức
Product_id	Int	Int	Id bảng product khóa phụ
User_id	Int	Int	Id bảng user khóa phụ
Amount	Float	Float	Tổng giá tiền sản phẩm
Payment_status	Varchar	100	Trạng thái sản phẩm
Created_at	Time	Time	Ngày đặt hàng
Approved	Nvarchar	100	Hiện thị admin duyệt đơn hay chưa
Completed	Nvarchar	100	Hiện thị đơn đã hoàn thành chưa
Address	Nvarchar	100	Địa chỉ khách hàng
Quantity	Quantity	Quantity	Số lượng sản phẩm
Phone_number	Nvarchar	15	Số điện thoại khách hàng

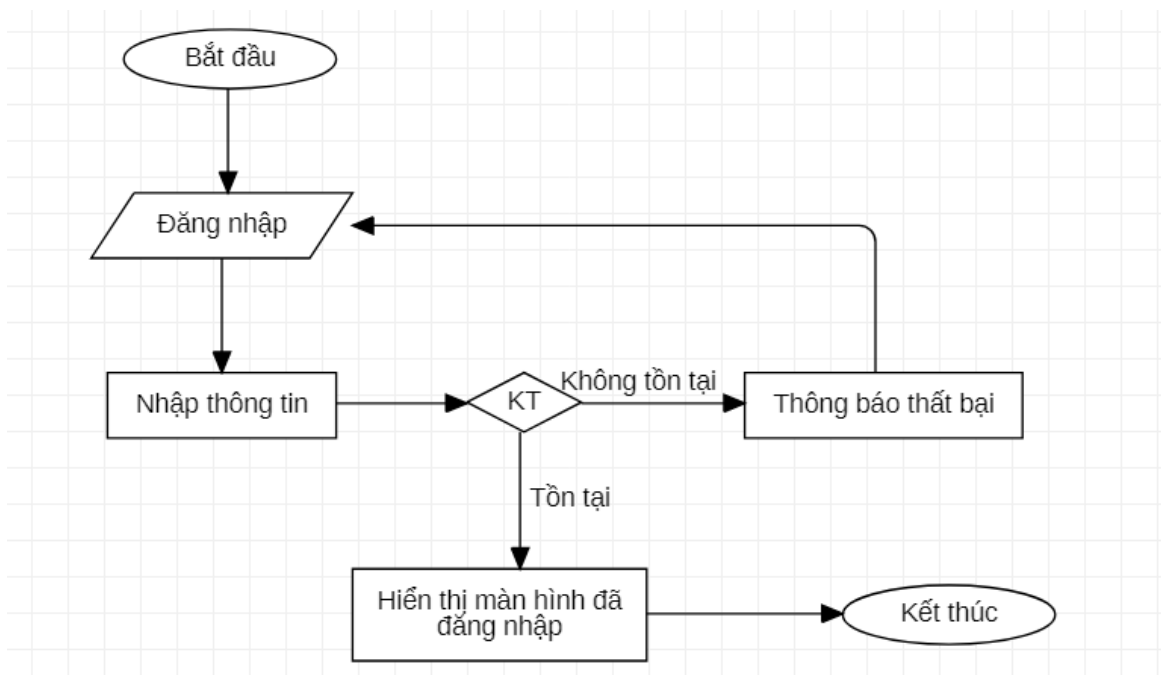
*Bảng 7: CSDL bảng hóa đơn*

### 3.8. Lưu đồ thuật toán



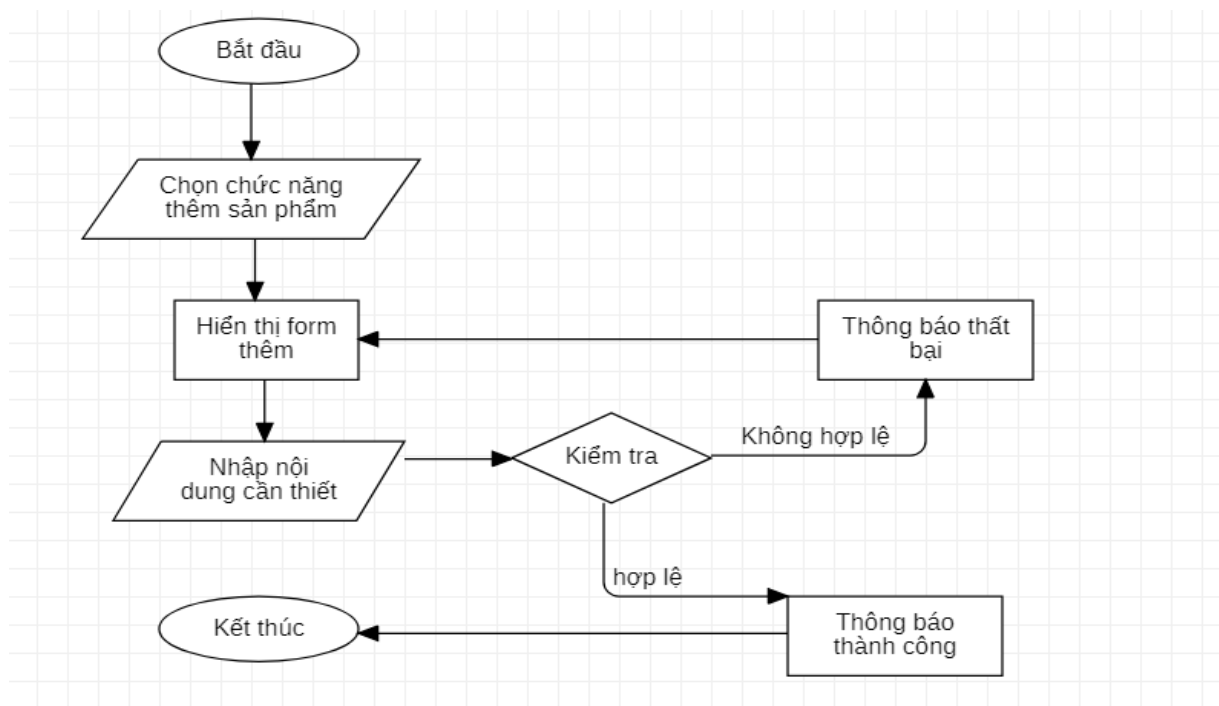
Hình 31: Lưu đồ thuật toán đăng ký

Khi khách hàng muốn mua hàng trên hệ thống thì sẽ phải đăng kí tài khoản trên hệ thống. Khách hàng sẽ chọn đăng kí và nhập các thông tin cần thiết mà giao diện yêu cầu nhập. Sau khi xác nhận thì hệ thống kiểm tra thông tin: Nếu hợp lệ sẽ lưu CSDL, ngược lại nếu xảy ra lỗi sẽ thông báo thất bại và cho phép nhập lại.



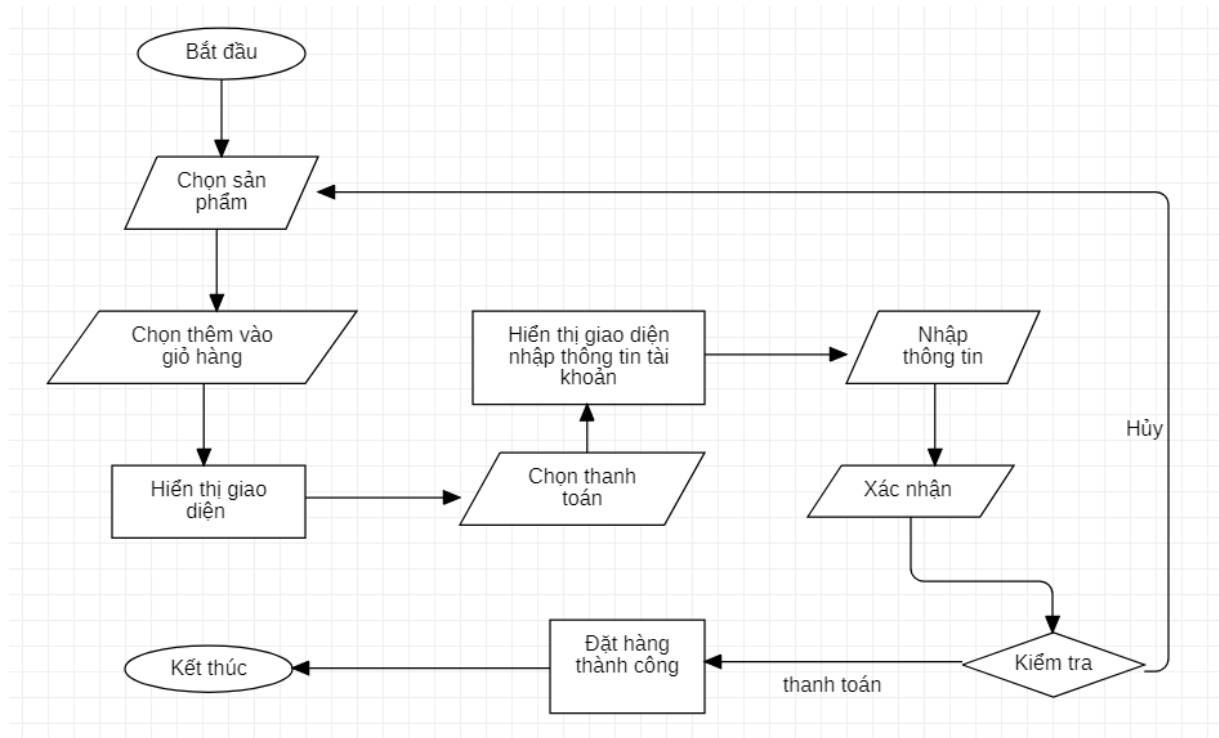
Hình 32: Lưu đồ thuật toán đăng nhập.

Khi khách hàng muốn mua hàng trên hệ thống thì sẽ phải đăng nhập tài khoản đã đăng kí trên hệ thống. Khách hàng sẽ chọn đăng nhập và nhập tên tài khoản và mật khẩu. Sau khi xác nhận thì hệ thống kiểm tra thông tin: Nếu tồn tại sẽ đăng nhập tài khoản khách hàng, ngược lại nếu xảy ra lỗi sẽ thông báo thất bại và cho phép nhập lại.



Hình 33: Lưu đồ thuật toán thêm sản phẩm.

Khi muốn thêm sản phẩm lên bán thì admin hệ thống (Nhân viên hay quản lý) và chức năng thêm sản phẩm sau đó nhập các thông tin về sản phẩm và xác nhận lưu. Sau khi xác nhận thì hệ thống kiểm tra thông tin: Nếu hợp lệ sẽ lưu CSDL, ngược lại nếu xảy ra lỗi sẽ thông báo thất bại và cho phép nhập lại.



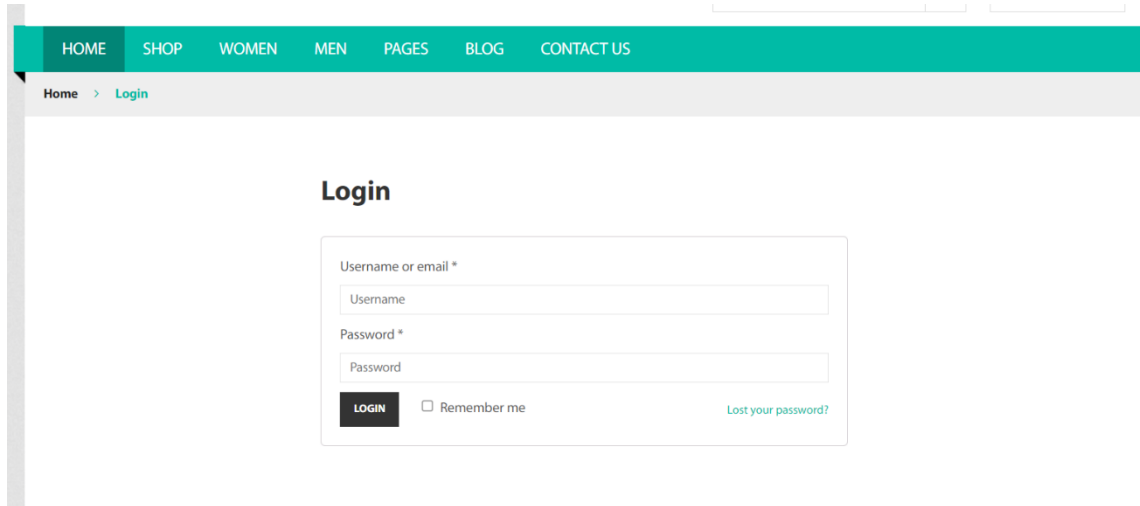
Hình 34: Lưu đồ thuật toán thêm sản phẩm vào giỏ hàng.

Khách hàng xem sản phẩm trên web và muốn mua hàng thì có thể thêm sản phẩm vào giỏ hàng. Khách hàng vào giỏ hàng và chọn lại sản phẩm muốn mua và ấn thanh toán. Hệ thống kiểm tra thông tin số lượng và dẫn đến trang điền thông tin giao hàng (nếu không đủ tiền sẽ thông báo hiển thị lên màn hình). Khách hàng nhập thông tin theo hệ thống yêu cầu và xác nhận. Hệ thống kiểm tra: Thông báo đặt thành công nếu KH xác nhận. Trở về giao diện giỏ hàng nếu KH hủy không đặt nữa.

## CHƯƠNG IV: PHÁT TRIỂN CHƯƠNG TRÌNH

### 4.1. Giao diện hệ thống website bán hàng của người dùng.

#### 4.1.1. Giao diện trang đăng nhập.



Hình 35: Giao diện trang đăng nhập.

Để đăng nhập đầu tiên:

Truy cập trang đăng nhập: Hướng dẫn mở trình duyệt và vào trang đăng nhập của website.

Nhập thông tin đăng nhập:

- Tên đăng nhập hoặc Email
- Mật khẩu

Tùy chọn ghi nhớ đăng nhập: Hướng dẫn khách hàng sử dụng chức năng "Ghi nhớ tôi".

Gửi thông tin đăng nhập: Nhấn nút "Đăng nhập".

Đăng ký tài khoản mới: Hướng dẫn tạo tài khoản nếu chưa có.

Mẹo khắc phục sự cố:

- Quên mật khẩu
- Đảm bảo nhập đúng thông tin
- Liên hệ hỗ trợ khách hàng khi cần thiết.

#### 4.1.2 Giao diện đăng kí tài khoản cho khách hàng.

Hình 36: Giao diện đăng kí tài khoản cho khách hàng.

Truy cập trang đăng ký: Mở trình duyệt và vào trang đăng ký của website.

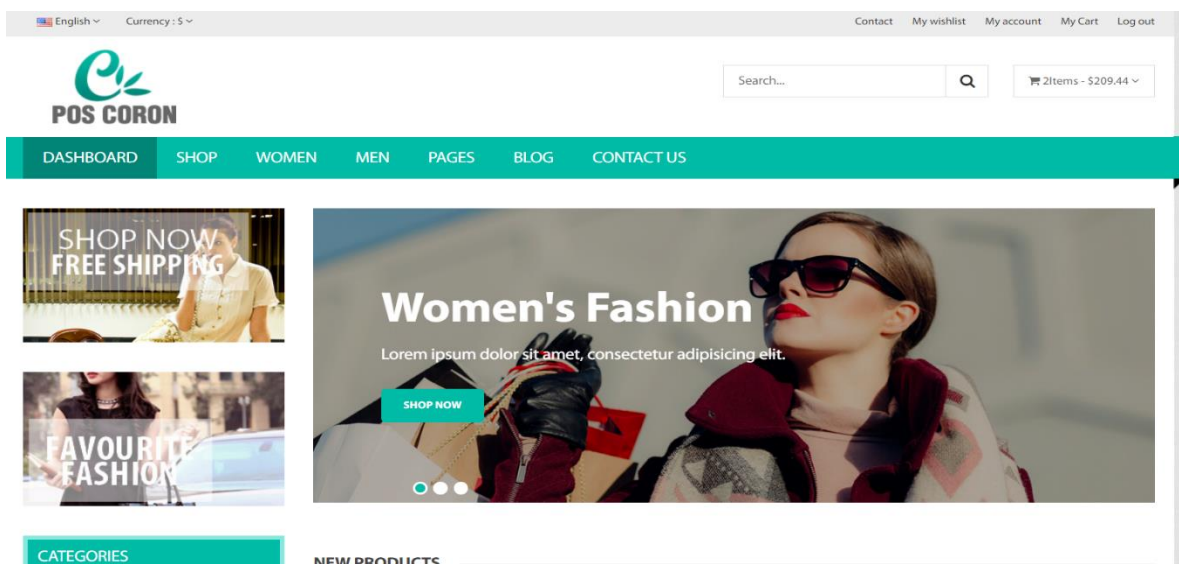
Nhập thông tin đăng ký:

- Tên đăng nhập
- Email
- Mật khẩu

Gửi thông tin đăng ký: Nhấn nút "Register".

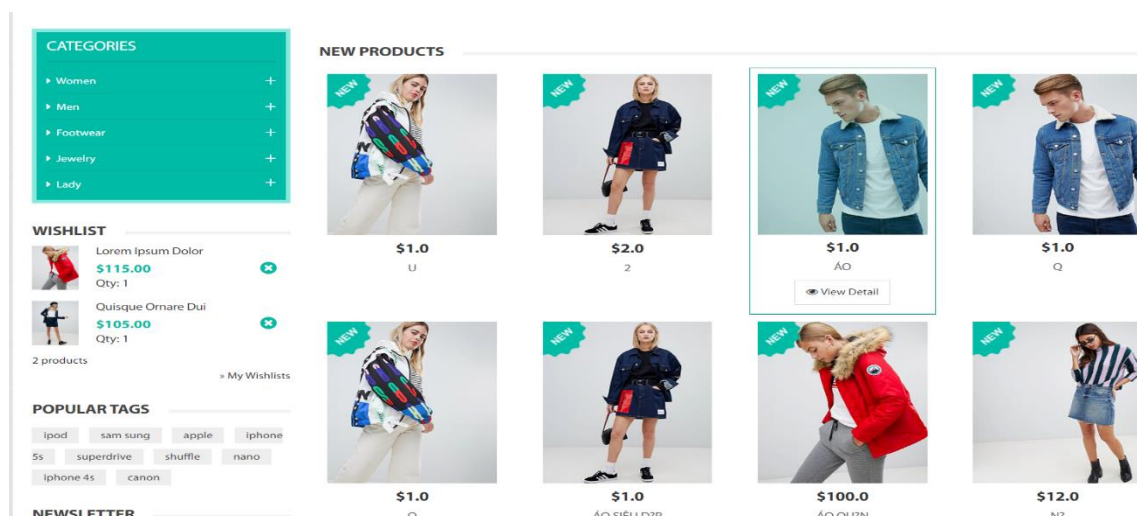
Đăng nhập nếu đã có tài khoản: Nhấn vào liên kết "Login an account?".

#### 4.1.3. Giao diện trang index của website.



Hình 37: Giao diện trang index của website.

## Các sản phẩm



Hình 38: Các sản phẩm

### Hiện thị danh sách sản phẩm mới

Trang web của em có một khu vực được đặt tên là "New Products", trong đó mỗi sản phẩm được hiển thị trong một thẻ div với class là `product_card`. Mỗi sản phẩm có các thông tin như hình ảnh, giá cả, tên sản phẩm, số lượng bán ra (nếu có), và thông báo nếu sản phẩm đã hết hàng.

### Chức năng tìm kiếm sản phẩm

Đầu vào: Có một ô tìm kiếm (input) cho phép người dùng nhập từ khóa để tìm kiếm sản phẩm.

Cơ chế hoạt động: Khi em nhập từ khóa vào ô tìm kiếm:

- JavaScript sẽ lấy giá trị của ô tìm kiếm và chuyển đổi thành chữ hoa để so sánh không phân biệt chữ hoa/thường.
- Sau đó, JavaScript sẽ duyệt qua từng sản phẩm được hiển thị và so sánh từ khóa tìm kiếm với tiêu đề của từng sản phẩm.
- Nếu tiêu đề của sản phẩm chứa từ khóa tìm kiếm, sản phẩm đó sẽ được hiển thị; nếu không, sản phẩm sẽ bị ẩn đi.

### Xem chi tiết sản phẩm

- Mỗi sản phẩm có một nút hoặc liên kết "View Detail" cho phép người dùng xem chi tiết sản phẩm. Liên kết này có thể dẫn đến một trang chi tiết riêng biệt của sản phẩm.





Lợi ích của tính năng này

- Tăng trải nghiệm người dùng: Người dùng có thể dễ dàng tìm kiếm và xem thông tin chi tiết về các sản phẩm mà họ quan tâm.
- Tối ưu hóa hiệu quả: Tính năng tìm kiếm giúp giảm thời gian tìm kiếm của người dùng và tăng khả năng tương tác trên trang web của em.

#### 4.1.4. Giao diện giỏ hàng.

### Your Shopping Cart

Item removed from cart.

DELETE	IMAGE	PRODUCT	PRICE	QUANTITY	TOTAL
		U	1.0	<input type="text" value="6"/>	6.0
					<button>UPDATE CART</button>

Hình 39: Giao diện giỏ hàng.

Xóa sản phẩm trong giỏ hàng

Để người dùng có thể xóa sản phẩm trong giỏ hàng, chúng ta cần cung cấp một cách thực hiện dễ dàng và trực quan. Dưới đây là các bước cần thiết để thực hiện tính năng này:

- Hiển thị danh sách sản phẩm trong giỏ hàng: Đảm bảo rằng trong giao diện giỏ hàng, mỗi sản phẩm được liệt kê với các thông tin như hình ảnh, tên sản phẩm, giá, số lượng và tùy chọn để xóa sản phẩm.
- Thêm nút hoặc liên kết Xóa: Mỗi sản phẩm trong giỏ hàng nên có một nút hoặc liên kết "Xóa" mà người dùng có thể nhấp để loại bỏ sản phẩm đó khỏi giỏ hàng.
- Thực hiện hành động Xóa bằng JavaScript: Sử dụng JavaScript để xử lý sự kiện khi người dùng nhấp vào nút Xóa. Bạn có thể sử dụng các công cụ như jQuery hoặc Vanilla JavaScript để lắng nghe sự kiện và thực hiện xóa sản phẩm khỏi danh sách giỏ hàng.

Cập nhật số lượng sản phẩm trong giỏ hàng

Khi người dùng thay đổi số lượng sản phẩm trong giỏ hàng hoặc tìm kiếm sản

phẩm mới, cần có một cơ chế để cập nhật số lượng sản phẩm một cách dễ dàng

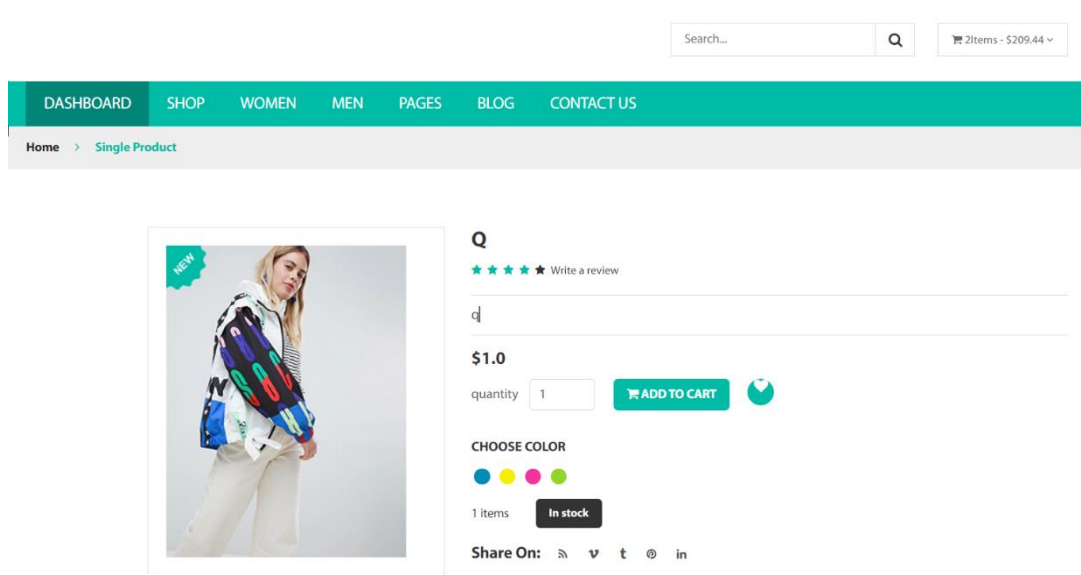
Lợi ích của tính năng này:

- Tăng tính tương tác: Cho phép người dùng điều chỉnh và quản lý sản phẩm trong giỏ hàng một cách linh hoạt.
- Tối ưu trải nghiệm người dùng: Giúp người dùng dễ dàng thực hiện các thay đổi mà không cần phải tải lại trang.

#### 4.1.5. Giao diện chi tiết sản phẩm.

Để người dùng có thể thêm sản phẩm vào giỏ hàng một cách thuận tiện, bạn cần cung cấp một giao diện rõ ràng và các chức năng tương tác như sau:

- Nút "Thêm vào giỏ hàng": Mỗi sản phẩm trong trang chi tiết sản phẩm cần có một nút hoặc liên kết "Thêm vào giỏ hàng". Người dùng có thể nhấp vào nút này để đưa sản phẩm vào giỏ hàng.
- Chọn số lượng: Khi người dùng nhấp vào nút "Thêm vào giỏ hàng", cần cung cấp một cửa sổ hoặc hộp thoại nhỏ cho phép người dùng chọn số lượng sản phẩm mà họ muốn thêm vào giỏ hàng.



Hình 40: Giao diện chi tiết sản phẩm.

#### 4.1.6. Giao diện chức năng đánh giá và bình luận cho từng sản phẩm.

Để người dùng có thể đánh giá sản phẩm một cách dễ dàng và nhanh chóng, bạn có thể cung cấp một hình thức đánh giá ngắn gọn như sau:

Hệ thống đánh giá: Dưới phần chi tiết sản phẩm, hiển thị một hệ thống đánh giá với các sao (5 sao thường được sử dụng). Người dùng có thể nhấp vào số sao tương ứng để đánh giá sản phẩm.

Ô nhận xét: Cung cấp một ô nhập nhận xét ngắn gọn cho người dùng. Họ có thể nhập một vài từ để chia sẻ cảm nhận của mình về sản phẩm.

Lợi ích của tính năng này:

- **Tăng trải nghiệm người dùng:** Người dùng có thể dễ dàng thêm sản phẩm vào giỏ hàng và đánh giá sản phẩm một cách thuận tiện.
- **Giúp cải thiện chất lượng sản phẩm:** Nhờ vào đánh giá của người dùng, bạn có thể thu thập phản hồi để cải thiện chất lượng và dịch vụ của sản phẩm.

phuong - Rating: 2  
22222222222222222222222222222222

phuong - Rating: 4  
ddddd

phuong - Rating: 2  
3333333333333333

nam - Rating: 4  
good

phuong - Rating: 5  
verygood

Add a Review

Rating:

Comment:

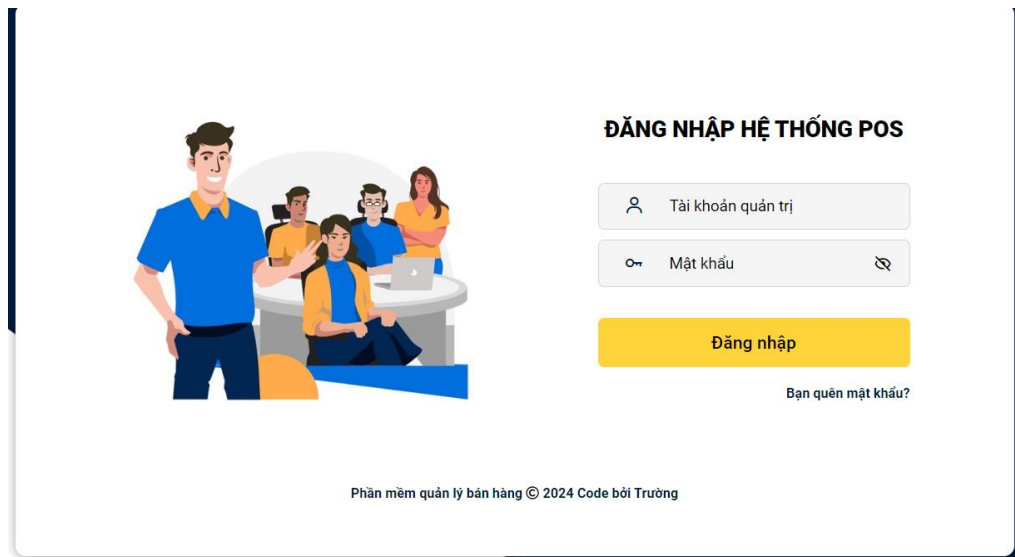
Submit Review

MORE INFODATA SHEETREVIEWS

Hình 41: Giao diện chức năng đánh giá và bình luận cho từng sản phẩm.

## 4.2. Giao diện quản trị admin

### 4.2.1. Giao diện đăng nhập admin.



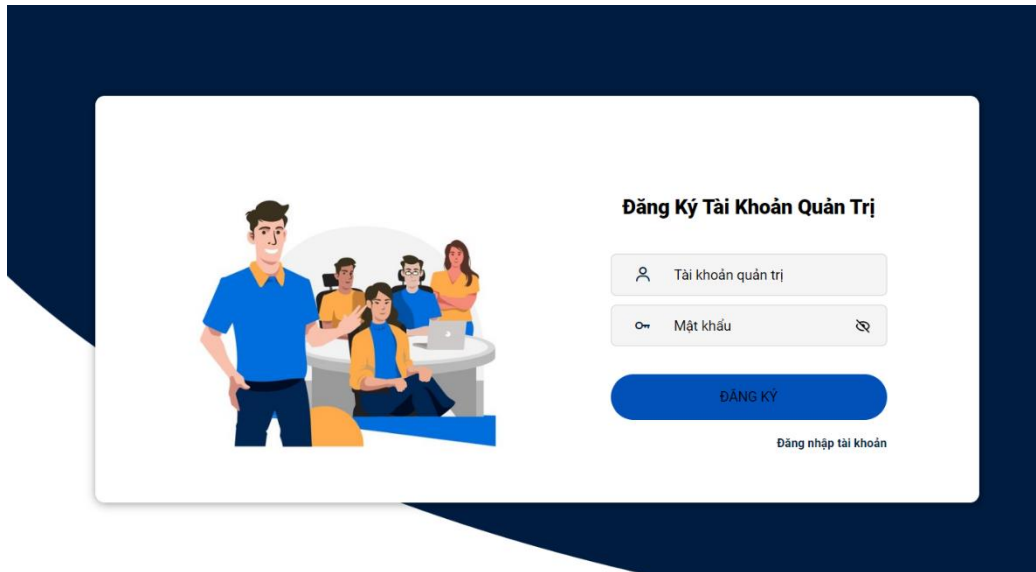
Hình 42: Giao diện đăng nhập admin

Chức năng Mật khẩu là công cụ quan trọng giúp người dùng bảo vệ và quản lý thông tin cá nhân trên trang web. Bao gồm:

- Thay đổi mật khẩu: Cho phép người dùng tự thay đổi mật khẩu hiện tại bằng một mật khẩu mới để tăng cường bảo mật.
- Khôi phục mật khẩu: Lựa chọn khi người dùng quên mật khẩu, cho phép họ thiết lập lại mật khẩu bằng cách xác nhận qua email.
- Chức năng Mật khẩu đảm bảo rằng thông tin cá nhân của người dùng luôn được bảo vệ và duy trì an toàn trên nền tảng trực tuyến.

#### 4.2.2. Giao diện đăng kí tài khoản admin.

Chức năng Đăng ký tài khoản Admin cho phép người dùng tạo một tài khoản quản trị viên mới trên trang web. Người dùng cung cấp thông tin cá nhân và quyền hạn để truy cập và quản lý các tính năng quan trọng của hệ thống. Chức năng này giúp đảm bảo an toàn và hiệu quả trong quản lý hệ thống trực tuyến.



Hình 43: Giao diện đăng kí tài khoản admin

### 4.2.3. Giao diện quản lý sản phẩm.

Chức năng này cho phép quản trị viên thêm, sửa đổi, xóa sản phẩm và cung cấp tính năng tìm kiếm nhanh chóng. Người dùng có thể dễ dàng tìm và quản lý các sản phẩm trên trang web một cách hiệu quả, từ đó cải thiện trải nghiệm mua sắm và quản lý kinh doanh trực tuyến.

QUẢN LÝ SẢN PHẨM

QUẢN LÝ SẢN PHẨMQUẢN LÝ TÀI KHOẢNQUẢN LÝ ADMINLỊCH CÔNG TÁCBAO CẢOSỰ KIỆNĐĂNG XUẤT



TÌM KIẾM NHÂN VIÊN:

Nhập tên nhân viên cần tìm...

CHỨC NĂNG CHÍNH:

Add Product

Products

Name	Description	Price	Quantity	Gender	Image	Actions
sp	u	1.0	1	male		<a href="#">Edit</a> <a href="#">Delete</a>
2	2	2.0	2	male		<a href="#">Edit</a> <a href="#">Delete</a>

Hình 44: Giao diện quản lý sản phẩm.

### 4.2.4. Giao diện quản lý thêm sản phẩm.

QUẢN LÝ SẢN PHẨM

QUẢN LÝ SẢN PHẨMQUẢN LÝ TÀI KHOẢNTIỀN LƯƠNGLỊCH CÔNG TÁCBAO CẢOSỰ KIỆNĐĂNG XUẤT

Thêm Sản Phẩm

Tên sản phẩm:

Tên sản phẩm

Mô tả sản phẩm:

Mô tả sản phẩm

Giá sản phẩm:

Giá sản phẩm

Số lượng:

Số lượng

Giới tính:

Nam

Hình ảnh:

Choose File

No file chosen

Thêm Sản Phẩm

Hình 45: Giao diện quản lý thêm sản phẩm

#### 4.2.5. Giao diện quản lý sửa phẩm.

QUẢN LÝ SẢN PHẨM

QUẢN LÝ SẢN PHẨMQUẢN LÝ TÀI KHOẢNTIỀN LƯƠNGLỊCH CÔNG TÁCBAO CÁOSỰ KIỆNĐĂNG XUẤT

Edit Product

Tên sản phẩm:

sp

Mô tả sản phẩm:

u

Giá sản phẩm:

1.0

Số lượng:

1

Giới tính:

Nam

Hình ảnh:

Choose FileNo file chosen

Cập nhật sản phẩm

Hình 46: Giao diện quản lý sửa phẩm.

#### 4.2.6. Giao diện quản lý tài khoản khách hàng.

QUẢN LÝ TÀI KHOẢN

QUẢN LÝ SẢN PHẨMQUẢN LÝ TÀI KHOẢNQUẢN LÝ ADMINLỊCH CÔNG TÁCBAO CÁOSỰ KIỆNĐĂNG XUẤT

TÌM KIẾM TÀI KHOẢN:

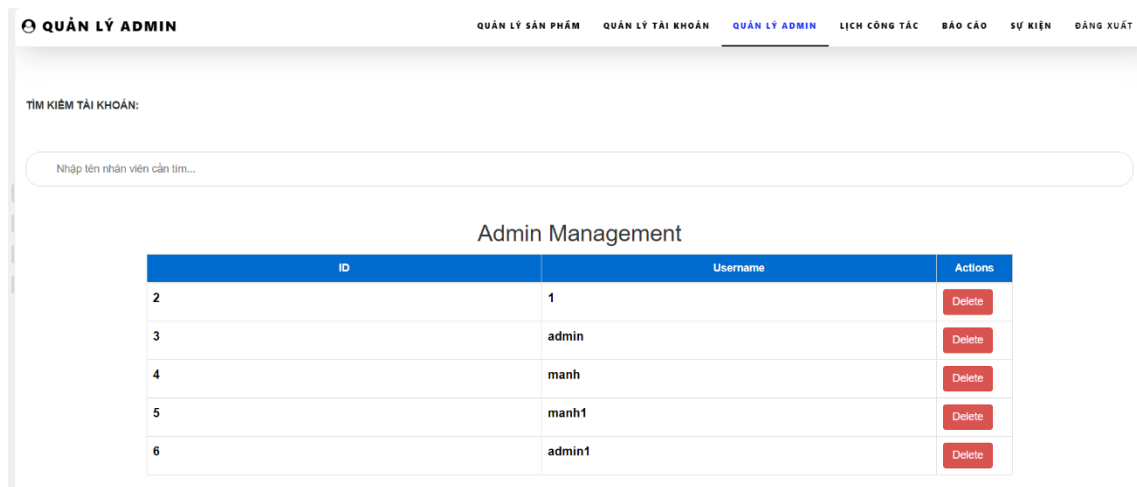
Nhập tên nhân viên cần tìm...

USER

ID	Username	Email	Actions
7	manh	manh111@gmail.com	Delete
8	ha	ha@gmail.com	Delete
10	1ha	1112@gmail.com	Delete
11	manhday	manhday@gmail.com	Delete
12	123	121@gmail.com	Delete

Hình 47: Giao diện quản lý tài khoản khách hàng.

#### 4.2.7. Giao diện quản lý tài khoản admin.



Hình 48: Giao diện quản lý tài khoản admin.

- Tìm kiếm tài khoản: Người dùng có thể nhập từ khóa vào ô tìm kiếm và nhấn nút để lọc danh sách tài khoản theo tên tài khoản hoặc email.
- Danh sách tài khoản: Hiện thị danh sách các tài khoản admin trong một bảng, bao gồm ID, tên tài khoản, email và nút "Xóa" để xóa tài khoản.
- Xóa tài khoản: Khi người dùng nhấn nút "Xóa" trên mỗi dòng, sẽ hiển thị cửa sổ xác nhận trước khi thực hiện xóa. Sau khi xác nhận, tài khoản sẽ được xóa khỏi danh sách và cập nhật lại giao diện ngay lập tức.



## KẾT LUẬN

Quá trình nghiên cứu và thực hiện đề tài "Xây dựng API cho ứng dụng bán hàng trên nền tảng Flask và SQL Server" đã mang lại cho em nhiều kiến thức và kỹ năng quý báu, từ lý thuyết cơ bản đến thực hành chuyên sâu. Đây không chỉ là hành trình học hỏi về công nghệ và kỹ thuật mà còn là quá trình phát triển và nâng cao khả năng sáng tạo, giải quyết vấn đề, và quản lý dự án trong lĩnh vực công nghệ thông tin.

Em đã bắt đầu bằng việc nắm vững các kiến thức cơ bản về API và cách sử dụng Flask để xây dựng các ứng dụng web và API. Điều này bao gồm việc thiết lập môi trường phát triển, cài đặt các thư viện và công cụ cần thiết, và thiết kế một kiến trúc ứng dụng đơn giản nhưng hiệu quả. Tiếp theo, em đã tích hợp SQL Server vào ứng dụng, học cách thiết kế cơ sở dữ liệu, tạo các model và thiết lập kết nối với Flask. Em đã xây dựng một API đầy đủ chức năng cho ứng dụng bán hàng, xử lý các yêu cầu từ người dùng thông qua các endpoint API, và áp dụng các kiến thức về thiết kế cơ sở dữ liệu.

Trong suốt quá trình thực hiện đề tài, em đã đối mặt với nhiều khó khăn và thách thức. Một trong những thách thức lớn nhất là việc tối ưu hóa hiệu năng và bảo mật cho API và cơ sở dữ liệu. Em phải tìm hiểu và áp dụng các kỹ thuật tối ưu hóa để đảm bảo rằng API hoạt động một cách hiệu quả, xử lý được lượng lớn yêu cầu từ người dùng mà không gặp sự cố, đồng thời đảm bảo rằng các truy vấn đến SQL Server diễn ra nhanh chóng và hiệu quả. Việc bảo mật cũng là một thách thức lớn, em đã áp dụng nhiều biện pháp bảo mật để bảo vệ ứng dụng khỏi các mối đe dọa tiềm ẩn như tấn công SQL injection, cross-site scripting (XSS), và các hình thức tấn công khác. Điều này đòi hỏi em phải liên tục cập nhật kiến thức và kỹ năng, cũng như thử nghiệm và kiểm tra kỹ lưỡng từng phần của ứng dụng.

Một thách thức khác là việc tích hợp các chức năng nâng cao như tìm kiếm, lọc. Mặc dù gặp nhiều khó khăn và thách thức, em đã nỗ lực vượt qua và đạt được những kết quả đáng kể. Em đã xây dựng được một API hoàn chỉnh tích hợp với SQL Server, có thể đáp ứng được các yêu cầu cơ bản và nâng cao của ứng dụng bán hàng, đồng thời đảm bảo được tính hiệu quả và bảo mật. Điều này không chỉ giúp em nâng cao kiến thức và kỹ năng mà còn mang lại sự tự tin và kinh nghiệm quý báu cho các dự án trong tương lai.

Tóm lại, đề tài này đã mang lại cho em một hành trình học hỏi và trải nghiệm phong phú, không chỉ trong lĩnh vực công nghệ thông tin mà còn trong việc phát triển các kỹ năng và kiến thức cần thiết để thực hiện và quản lý các dự án ứng dụng khác trong tương lai. Từ đó, mở ra nhiều cơ hội và tiềm năng trong sự nghiệp và học tập của em..

# TÀI LIỆU THAM KHẢO

## ***Tiếng việt:***

- [1]. Lương Cao Đông.(2024). Giáo trình lý thuyết và thiết kế cơ sở dữ liệu. Nhà xuất bản Đại học Quốc gia Hà Nội.
- [2]. Chu Minh Tuệ. Giáo trình phân tích và thiết kế đối tượng với UML.
- [3]. Lê Văn Phong. Bài giảng cơ sở dữ liệu phân tán. Đại học Đại Nam

## ***Tiếng anh:***

- [1]. Flask Python Tutorial – Self-study Flask Python For Beginners Core. Apress.
- [2]. Petzold, Charles. (2002). "Programming Microsoft Windows Forms"

## ***Danh mục website tham khảo:***

- [1]. <https://flask.palletsprojects.com/en/3.0.x/>
- [2]. <https://vn.got-it.ai/blog/orm-va-sqlalchemy-chiec-dua-than-trong-quan-tri-co-so-du-lieu>
- [3]. <https://flask.palletsprojects.com/en/3.0.x/>
- [4]. <https://rikkei.edu.vn/html-css-javascript-la-gi/>
- [5]. <https://chatgpt.com/>

## PHỤ LỤC

### Các code cần thiết cho các chức năng của website.

#### *App.py.*

App.py file chính của ứng dụng quản lý các blueprint và kết nối sql s

```
from flask import Flask, render_template, session, redirect, url_for, request, flash
from models import db, Product, User, Review, Admin
from auth.auth import auth_blueprint
from product.product import product_blueprint
from cart.cart import cart_bp
from flask_login import LoginManager, current_user, login_required
from admin_auth.admin_auth import admin_auth_blueprint

app = Flask(__name__)
login_manager = LoginManager(app)
app.secret_key = "your_secret_key"
app.config['SQLALCHEMY_DATABASE_URI'] = 'mssql+pyodbc://@LAPTOP-JCCKGROC\\SA/Do_anToiNghiep?driver=ODBC+Driver+17+for+SQL+Server'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

# Bind SQLAlchemy instance with the Flask app
db.init_app(app)

# Register blueprints
app.register_blueprint(auth_blueprint)
app.register_blueprint(product_blueprint)
app.register_blueprint(cart_bp, url_prefix='/cart')
```

#### *Models.py*

Models.py file quản lý các cdsi của web gồm các class chính như product, user, cart....

```
db = SQLAlchemy()

class Product(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    description = db.Column(db.String(255))
    price = db.Column(db.Float, nullable=False)
    quantity = db.Column(db.Integer, nullable=False)
    gender = db.Column(db.String(10))
    image_path = db.Column(db.String(255))

class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(50), unique=True, nullable=False)
```

```

email = db.Column(db.String(120), unique=True, nullable=False)
password_hash = db.Column(db.String(128), nullable=False)

def __init__(self, username, email, password):
    self.username = username
    self.email = email
    self.password_hash = generate_password_hash(password)

def check_password(self, password):
    return check_password_hash(self.password_hash, password)

class CartItem(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)
    product_id = db.Column(db.Integer, db.ForeignKey('product.id'),
nullable=False)
    quantity = db.Column(db.Integer, nullable=False, default=1)
    shopping_cart_id = db.Column(db.Integer,
db.ForeignKey('shopping_cart.id'), nullable=False)
    product = db.relationship('Product', backref=db.backref('cart_items',
lazy=True))

class ShoppingCart(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)
    items = db.relationship('CartItem', backref='shopping_cart', lazy=True)

class Admin(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), unique=True, nullable=False)
    password = db.Column(db.String(200), nullable=False)

    def __init__(self, username, password):
        self.username = username
        self.password = password

class Review(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    product_id = db.Column(db.Integer, db.ForeignKey('product.id'),
nullable=False)
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)
    rating = db.Column(db.Integer, nullable=False)
    comment = db.Column(db.String(255))

    def __init__(self, product_id, user_id, rating, comment):
        self.product_id = product_id
        self.user_id = user_id
        self.rating = rating
        self.comment = comment

```

## *Product.py*

Product.py xây dựng chức năng thêm sửa xóa sản phẩm

```
@product_blueprint.route('/add_product', methods=['GET', 'POST'])
def add_product():
    if request.method == 'POST':
        name = request.form['name']
        description = request.form['description']
        price = request.form['price']
        quantity = request.form['quantity']
        gender = request.form['gender']
        image = request.files['image'] # Lấy tệp hình ảnh từ form

        # Xử lý lưu trữ tệp hình ảnh và lấy đường dẫn của nó
        image_path = save_image(image)

        new_product = Product(name=name, description=description, price=price,
                               quantity=quantity, gender=gender, image_path=image_path)
        db.session.add(new_product)
        db.session.commit()

        # Chuyển hướng người dùng trở lại trang dashboard
        return redirect(url_for('admin'))
    else:
        return render_template('add_product.html')

@product_blueprint.route('/product/<int:product_id>/edit', methods=['GET',
'POST'])
def edit_product(product_id):
    product = Product.query.get_or_404(product_id)

    if request.method == 'POST':
        product.name = request.form['name']
        product.description = request.form['description']
        product.price = request.form['price']
        product.quantity = request.form['quantity']
        product.gender = request.form['gender']

        if 'image' in request.files:
            image = request.files['image']
            if image.filename != '':
                # Xử lý lưu trữ tệp hình ảnh và cập nhật đường dẫn
                image_path = save_image(image)
                product.image_path = image_path

        db.session.commit()

        # Chuyển hướng người dùng trở lại trang chi tiết sản phẩm sau khi
        chỉnh sửa
```

```

        return redirect(url_for('admin', product_id=product_id))
    else:
        return render_template('edit_product.html', product=product)

@product_blueprint.route('/product/<int:product_id>/delete', methods=['POST'])
def delete_product(product_id):
    product = Product.query.get_or_404(product_id)
    db.session.delete(product)
    db.session.commit()

    # Chuyển hướng người dùng trở lại trang dashboard
    return redirect(url_for('admin'))

@product_blueprint.route('/products', methods=['GET'])
def display_products():
    products = Product.query.all()
    return render_template('products.html', products=products)

def save_image(image):
    upload_folder = os.path.join('static', 'uploads')
    if not os.path.exists(upload_folder):
        os.makedirs(upload_folder)

    filename = secure_filename(image.filename)
    image_path = os.path.join(upload_folder, filename)
    image.save(image_path)

    return os.path.join('uploads', filename)

```

### *Admin\_login.html*

Admin\_login.html đoạn code này xử lý chức năng đăng nhập admin và sử dụng api

```

<script>
    document.getElementById("loginForm").addEventListener("submit",
function(event) {
    event.preventDefault(); // Prevent the default form submission

    // Collect input data
    var formData = {
        username: document.getElementById("username").value,
        password: document.getElementById("password").value
    };

    // Send a POST request to the login API endpoint
    fetch("/admin/api/login", {
        method: "POST",
        headers: {
            "Content-Type": "application/json"
        },
        body: JSON.stringify(formData)
    })

```

```

    })
    .then(response => {
        if (response.ok) {
            window.location.href = "/admin"; // Redirect to admin
            dashboard on successful login
        } else {
            return response.json();
        }
    })
    .then(data => {
        document.getElementById("message").innerText = data.error ||
        "Unknown error occurred";
    })
    .catch(error => {
        console.error("Error:", error);
        document.getElementById("message").innerText = "An error
        occurred while processing your request";
    });
});
</script>
</body>

```