|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TRƯỜNG ĐẠI HỌC NGUYỄN TRÃI**  **KHOA CÔNG NGHỆ THÔNG TIN & CHUYỂN ĐỔI SỐ**    **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **ĐỀ TÀI**  **WEBSITE GIỚI THIỆU VÀ KINH DOANH THỰC PHẨM SẠCH HTFOOD SỬ DỤNG NODE.JS, REACT.JS, MYSQL**   |  |  | | --- | --- | | Giảng viên hướng dẫn | ThS. Trịnh Văn Chung | | Sinh viên thực hiện | Lê Thương Hoài Thu | | Lớp | K22CNT3 | | Mã sinh viên | 2210900126 |   **Hà Nội – 12/2025** |
| **TRƯỜNG ĐẠI HỌC NGUYỄN TRÃI**  **KHOA CÔNG NGHỆ THÔNG TIN & CHUYỂN ĐỔI SỐ**    **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **ĐỀ TÀI**  **WEBSITE GIỚI THIỆU VÀ KINH DOANH THỰC PHẨM SẠCH HTFOOD SỬ DỤNG NODE.JS, REACT.JS, MYSQL**   |  |  | | --- | --- | | Giảng viên hướng dẫn | ThS. Trịnh Văn Chung | | Sinh viên thực hiện | Lê Thương Hoài Thu | | Lớp | K22CNT3 | | Mã sinh viên | 2210900126 |   **Hà Nội – 12/2025** |

# ****LỜI CẢM ƠN****

Trong suốt quá trình thực hiện đồ án với đề tài “Website giới thiệu và kinh doanh thực phẩm sạch sử dụng HTFood Node.js, React.js, Mysql”, em đã nhận được sự quan tâm, hướng dẫn và hỗ trợ quý báu từ nhiều cá nhân và tập thể. Nhân dịp hoàn thành đồ án, em xin được bày tỏ lòng biết ơn sâu sắc đến những người đã đồng hành, giúp đỡ và tạo điều kiện thuận lợi để em hoàn thành tốt nhiệm vụ học tập của mình.

Trước hết, em xin trân trọng gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy ThS. Trịnh Văn Chung – người đã trực tiếp hướng dẫn, theo dõi và hỗ trợ em trong suốt quá trình thực hiện đồ án. Với tinh thần làm việc khoa học, nghiêm túc, thầy đã giúp em từng bước hoàn thiện nội dung nghiên cứu, rèn luyện tư duy logic cũng như khả năng vận dụng kiến thức chuyên môn vào thực tiễn. Sự hướng dẫn tận tình và những chỉ dẫn quý báu của thầy chính là yếu tố quan trọng góp phần quyết định đến chất lượng và tiến độ hoàn thành đồ án.

Em xin chân thành cảm ơn Ban Giám hiệu Trường Đại học Nguyễn Trãi, cùng toàn thể quý thầy cô trong Khoa Công nghệ Thông tin đã tận tình giảng dạy, truyền đạt cho em những kiến thức nền tảng và chuyên sâu trong suốt quá trình học tập tại trường. Những kiến thức này không chỉ giúp em hoàn thành đồ án một cách hiệu quả mà còn là hành trang quý báu, tạo nền tảng vững chắc cho quá trình làm việc và phát triển nghề nghiệp của em trong tương lai.

Bên cạnh đó, em cũng xin gửi lời cảm ơn đến bạn bè, tập thể lớp và những người thân đã luôn sẵn sàng chia sẻ kinh nghiệm, hỗ trợ và động viên em trong suốt quá trình học tập cũng như thực hiện đồ án.

Cuối cùng, em xin bày tỏ lòng biết ơn sâu sắc nhất đến gia đình – những người luôn là chỗ dựa tinh thần vững chắc, luôn quan tâm, động viên và tạo mọi điều kiện thuận lợi để em yên tâm học tập và nghiên cứu.

Mặc dù đã rất cố gắng trong quá trình thực hiện, song do thời gian nghiên cứu và kinh nghiệm thực tế còn hạn chế, đồ án khó tránh khỏi những thiếu sót nhất định. Em kính mong nhận được sự cảm thông cùng những ý kiến đóng góp quý báu từ quý thầy cô để em có thể tiếp tục hoàn thiện kiến thức và kỹ năng trong quá trình học tập và nghiên cứu sau này.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

Lê Thương Hoài Thu

# ****LỜI CAM ĐOAN****

Em xin cam đoan rằng đồ án với đề tài “Website giới thiệu và kinh doanh thực phẩm sạch sử dụng HTFood Node.js, React.js, Mysql”,”là công trình nghiên cứu và thực hiện nghiêm túc của riêng em, được hoàn thành dưới sự hướng dẫn trực tiếp của giảng viên hướng dẫn. Toàn bộ nội dung đồ án được xây dựng dựa trên quá trình tìm hiểu tài liệu, nghiên cứu lý thuyết kết hợp với việc phân tích, thiết kế và triển khai hệ thống một cách trung thực, khách quan và có cơ sở khoa học.

Các số liệu, kết quả nghiên cứu, hình ảnh minh họa, sơ đồ, bảng biểu và nội dung trình bày trong đồ án đều phản ánh đúng quá trình thực hiện của bản thân em, không sao chép từ bất kỳ công trình nghiên cứu hay đồ án nào khác. Những nội dung có tham khảo từ sách, giáo trình, bài báo khoa học, tài liệu trên Internet hoặc các nguồn khác đều đã được trích dẫn đầy đủ, rõ ràng và đúng quy định.

Em xin cam đoan rằng đồ án không vi phạm bản quyền, không sao chép toàn phần hoặc một phần nội dung của bất kỳ công trình nào khi chưa được phép. Trong trường hợp có sai sót hoặc thiếu chính xác về nội dung, số liệu hoặc cách trình bày, em xin hoàn toàn chịu trách nhiệm trước Nhà trường, Khoa Công nghệ Thông tin và giảng viên hướng dẫn.

Em cũng xin cam đoan rằng kết quả của đồ án chưa từng được sử dụng để nộp cho bất kỳ học phần, đồ án hay luận văn nào khác tại bất kỳ cơ sở đào tạo nào. Tất cả các nội dung, kết luận và đề xuất trong đồ án đều xuất phát từ quá trình học tập, nghiên cứu và thực hiện của cá nhân em.

Em xin hoàn toàn chịu trách nhiệm về lời cam đoan này!

Người cam đoan

Lê Thương Hoài Thu

# MỤC LỤC

[LỜI CẢM ƠN ii](#_Toc220094683)

[LỜI CAM ĐOAN iii](#_Toc220094684)

[MỤC LỤC iv](#_Toc220094685)

[DANH MỤC CÁC TỪ VIẾT TẮT v](#_Toc220094686)

[DANH MỤC HÌNH ẢNH vii](#_Toc220094687)

[DANH MỤC BẢNG BIỂU x](#_Toc220094688)

[MỞ ĐẦU 1](#_Toc220094689)

[CHƯƠNG 1. NGHIÊN CỨU CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN VỀ ĐỀ TÀI 3](#_Toc220094690)

[1.1. Cơ sở lý thuyết 3](#_Toc220094691)

[1.2. Tổng quan về lĩnh vực thực phẩm sạch 15](#_Toc220094692)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 19](#_Toc220094693)

[2.1. Phân tích người dùng hệ thống 19](#_Toc220094694)

[2.1.1. Mô tả hệ thống 19](#_Toc220094695)

[2.1.2. Các đối tượng tham gia hệ thống 19](#_Toc220094696)

[2.2. Phân tích tổng thể hệ thống 22](#_Toc220094697)

[2.2.1. Biểu đồ Use Case tổng quát 22](#_Toc220094698)

[2.2.2. Biểu đồ Use Case của Admin (Quản trị viên) 26](#_Toc220094699)

[2.2.3. Biểu đồ Use Case của Khách hàng 31](#_Toc220094700)

[2.2.4. Biểu đồ hoạt động (Activity Diagram) 39](#_Toc220094701)

[2.2.5. Biểu đồ tuần tự (Sequence Diagram) 43](#_Toc220094702)

[2.2.6. Biểu đồ lớp (Class Diagram) 47](#_Toc220094703)

[2.3. Thiết kế cơ sở dữ liệu 48](#_Toc220094704)

[2.3.1. Tổng quan cơ sở dữ liệu 48](#_Toc220094705)

[2.3.2. Danh sách các bảng dữ liệu 49](#_Toc220094706)

[2.3.3. Cấu trúc chi tiết các bảng 49](#_Toc220094707)

[CHƯƠNG 3: XÂY DỰNG HỆ THỐNG 56](#_Toc220094708)

[3.1. Giao diện 56](#_Toc220094709)

[3.3. Kiểm thử hệ thống 63](#_Toc220094710)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 66](#_Toc220094711)

[TÀI LIỆU THAM KHẢO 67](#_Toc220094712)

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Giải thích** |
| API | Application Programming Interface – Giao diện lập trình ứng dụng |
| CSS | Cascading Style Sheets – Ngôn ngữ định dạng giao diện web |
| CSS3 | Cascading Style Sheets phiên bản 3 |
| CSDL | Cơ sở dữ liệu |
| DOM | Document Object Model – Mô hình tài liệu dạng cây |
| HTML | HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản |
| HTML5 | HyperText Markup Language phiên bản 5 |
| IDE | Integrated Development Environment – Môi trường phát triển tích hợp |
| JS | Javascript – Ngôn ngữ lập trình Javacript |
| MVC | Model – View – Controller – Mô hình thiết kế phần mềm |
| MySQL | My Structured Query Language – Hệ quản trị cơ sở dữ liệu quan hệ |
| RDBMS | Relational Database Management System – Hệ quản trị cơ sở dữ liệu quan hệ |
| RESTful API | Kiến trúc xây dựng API theo chuẩn REST |
| SEO | Search Engine Optimization – Tối ưu hóa công cụ tìm kiếm |
| SQL | Structured Query Language – Ngôn ngữ truy vấn có cấu trúc |
| SVG | Scalable Vector Graphics – Định dạng đồ họa vector |
| UI | User Interface – Giao diện người dùng |
| UML | Unified Modeling Language – Ngôn ngữ mô hình hóa thống nhất |
| VS Code | Visual Studio Code – Môi trường soạn thảo mã nguồn |
| W3C | World Wide Web Consortium – Tổ chức tiêu chuẩn Web toàn cầu |
| PHP | Hypertext Preprocessor – ngôn ngữ lập trình kịch bản mã nguồn mở, chạy phía máy chủ |
| HTTP | [Hypertext Transfer Protocol](https://www.google.com/search?q=Hypertext+Transfer+Protocol&oq=HTTP+l%C3%A0+g%C3%AC&gs_lcrp=EgZjaHJvbWUyCQgAEEUYORiABDIHCAEQABiABDIGCAIQABgeMgYIAxAAGB4yBggEEAAYHjIGCAUQABgeMggIBhAAGAUYHjIICAcQABgFGB4yCAgIEAAYBRgeMggICRAAGAUYHtIBCDgyMjRqMGo3qAIAsAIA&sourceid=chrome&ie=UTF-8&ved=2ahUKEwjEgdKavZqSAxUYklYBHcZQPPIQgK4QegQIARAB) – Giao thức Truyền tải Siêu văn bản |
| I/O | Input/Output – kỹ thuật xử lý bất đồng bộ (Asynchronous I/O) hoặc [Non-blocking I/O](https://www.google.com/search?q=Non-blocking+I%2FO&sca_esv=9b069d156a6bdf17&sxsrf=ANbL-n4_3szYaO4uS17Q9Dqfc9eVBU9pAw%3A1769045748747&ei=9H5xaa-oLcWP2roPrNH1mQk&ved=2ahUKEwiY4tGegZ6SAxVBklYBHfeIEsYQgK4QegQIARAC&uact=5&oq=I%3FO+l%C3%A0+c%C3%A1c+thao+t%C3%A1c+v%C3%A0o%2Fra+nh%C6%B0+%C4%91%E1%BB%8Dc+ghi+t%E1%BB%87p+tin%2C+truy+v%E1%BA%A5n+c%C6%A1+s%E1%BB%9F+d%E1%BB%AF+li%E1%BB%87u+hay+x%E1%BB%AD+l%C3%BD+request+t%E1%BB%AB+ng%C6%B0%E1%BB%9Di+d%C3%B9ng+s%E1%BA%BD+kh%C3%B4ng+l%C3%A0m+%E2%80%9Cch%E1%BA%B7n%E2%80%9D+lu%E1%BB%93ng+x%E1%BB%AD+l%C3%BD+ch%C3%ADnh+c%E1%BB%A7a+h%E1%BB%87+th%E1%BB%91ng+l%C3%A0+vi%E1%BA%BFt+t%E1%BA%AFt+c%E1%BB%A7a+g%C3%AC&gs_lp=Egxnd3Mtd2l6LXNlcnAi5gFJP08gbMOgIGPDoWMgdGhhbyB0w6FjIHbDoG8vcmEgbmjGsCDEkeG7jWMgZ2hpIHThu4dwIHRpbiwgdHJ1eSB24bqlbiBjxqEgc-G7nyBk4buvIGxp4buHdSBoYXkgeOG7rSBsw70gcmVxdWVzdCB04burIG5nxrDhu51pIGTDuW5nIHPhur0ga2jDtG5nIGzDoG0g4oCcY2jhurdu4oCdIGx14buTbmcgeOG7rSBsw70gY2jDrW5oIGPhu6dhIGjhu4cgdGjhu5FuZyBsw6Agdmnhur90IHThuq90IGPhu6dhIGfDrEgAUABYAHAAeACQAQCYAQCgAQCqAQC4AQPIAQD4AQGYAgCgAgCYAwCSBwCgBwCyBwC4BwDCBwDIBwCACAA&sclient=gws-wiz-serp) |

# DANH MỤC HÌNH ẢNH

[Hình 2.1 Sơ đồ ngữ cảnh hệ thống 23](#_Toc220082827)

[Hình 2.2 Biểu đồ use case tổng quát 24](#_Toc220082828)

[Hình 2.3 Biểu đồ Use Case Đăng nhập – đăng xuất 25](#_Toc220082829)

[Hình 2.4 Biểu đồ Use Case Quản lý sản phẩm 27](#_Toc220082830)

[Hình 2.5 biểu đồ Use Case Quản lý khách hàng 28](#_Toc220082831)

[Hình 2.6 Biểu đồ Use Case Quản lý đơn hàng 29](#_Toc220082832)

[Hình 2.7 Biểu đồ Use Case Quản lý nội dung 30](#_Toc220082833)

[Hình 2.8 Biểu đồ Use Case Thống kê – báo cáo 31](#_Toc220082834)

[Hình 2.9 Biểu đồ Use Case Đăng ký tài khoản 33](#_Toc220082835)

[Hình 2.10 Biểu đồ Use Case Thêm vào giỏ 34](#_Toc220082836)

[Hình 2.11 Biểu đồ Use Case Xem – Tìm kiếm sản phẩm 35](#_Toc220082837)

[Hình 2.12 Biểu đồ Use Case Quản lý tài khoản cá nhân 36](#_Toc220082838)

[Hình 2.13 Biểu đồ Use Case Quản lý giỏ hàng 37](#_Toc220082839)

[Hình 2.14 Biểu đồ Use Case Đặt hàng 38](#_Toc220082840)

[Hình 2.15 Biểu đồ Use Case Thanh toán 39](#_Toc220082841)

[Hình 2.16 Biểu đồ hoạt động đăng ký 40](#_Toc220082842)

[Hình 2.17 Biểu đồ hoạt động đăng nhập – đăng xuất 41](#_Toc220082843)

[Hình 2.18 Biểu đồ hoạt động mua hàng – đặt đơn 42](#_Toc220082844)

[Hình 2.19 Biểu đồ hoạt động thanh toán – theo dõi đơn hàng 43](#_Toc220082845)

[Hình 2.20 Biểu đồ tuần tự đăng nhập – đăng xuất 44](#_Toc220082846)

[Hình 2.21 Biểu đồ tuần tự đặt hàng 45](#_Toc220082847)

[Hình 2.22 Biểu đồ tuần tự thanh toán 46](#_Toc220082848)

[Hình 2.23 Biểu đồ lớp hệ thống thực phẩm sạch HTFood 48](#_Toc220082849)

[Hình 2.24 Mối quan hệ giữa các bảng 55](#_Toc220082850)

[Hình 2.25 Ảnh đăng ký 56](#_Toc220082851)

[Hình 2.26 Ảnh đăng nhập 58](#_Toc220082852)

[Hình 2.27 Danh sách sản phẩm 60](#_Toc220082853)

[Hình 2.28 Giao diện Blog 62](#_Toc220082854)

Danh mục hình ảnh

# DANH MỤC BẢNG BIỂU

[Bảng 2.1. Bảng các chức năng chính của Admin và Staff 24](#_Toc220085275)

[Bảng 2.2. Mô tả use case chức năng Đăng nhập – đăng xuất hệ thống 26](#_Toc220085276)

[Bảng 2.3. Mô tả chức năng Quản lý sản phẩm 27](#_Toc220085277)

[Bảng 2.4. Mô tả chức năng Quản lý khách hàng 28](#_Toc220085278)

[Bảng 2.5. Mô tả chức năng Quản lý đơn hàng 29](#_Toc220085279)

[Bảng 2.6. Bảng mô tả use case Quản lý nội dung 30](#_Toc220085280)

[Bảng 2.7. Bảng mô tả use case Thống kê – báo cáo 31](#_Toc220085281)

[Bảng 2.8. Bảng các chức năng chính của khách hàng 32](#_Toc220085282)

[Bảng 2.9. Bảng mô tả use case Đăng ký tài khoản 33](#_Toc220085283)

[Bảng 2.10. Bảng mô tả use case Thêm vào giỏ hàng 34](#_Toc220085284)

[Bảng 2.11. Bảng mô tả use case Tìm kiếm sản phẩm 35](#_Toc220085285)

[Bảng 2.12. Mô tả use case Quản lý tài khoản cá nhân 36](#_Toc220085286)

[Bảng 2.13. mô tả use case Quản lý giỏ hàng 37](#_Toc220085287)

[Bảng 2.14. mô tả use case Đặt hàng 38](#_Toc220085288)

[Bảng 2.15. Bảng mô tả use case Thanh toán 39](#_Toc220085289)

[Bảng 2.16. Danh sách các bảng dữ liệu 49](#_Toc220085290)

[Bảng 2.17. Bảng nguoidung 49](#_Toc220085291)

[Bảng 2.18. Bảng khachhang 50](#_Toc220085292)

[Bảng 2.19. Bảng nhacungcap 50](#_Toc220085293)

[Bảng 2.20. Bảng sanpham 51](#_Toc220085294)

[Bảng 2.21. Bảng giohang 51](#_Toc220085295)

[Bảng 2.22. Bảng khuyenmai 51](#_Toc220085296)

[Bảng 2.23. Bảng đơn hàng 52](#_Toc220085297)

[Bảng 2.24. Bảng chitiet\_donhang 52](#_Toc220085298)

[Bảng 2.25. Bảng lienhe 53](#_Toc220085299)

[Bảng 2.26. Bảng Blog 53](#_Toc220085300)

[Bảng 2.27. Bảng thanhtoan 54](#_Toc220085301)

[Bảng 2.28. Bảng nhanvien 54](#_Toc220085302)

# ****MỞ ĐẦU****

**1. Lý do chọn đề tài**

Trong những năm gần đây, vấn đề an toàn vệ sinh thực phẩm đang trở thành mối quan tâm hàng đầu của toàn xã hội. Tình trạng thực phẩm bẩn, thực phẩm không rõ nguồn gốc xuất xứ, sử dụng hóa chất độc hại trong quá trình sản xuất và bảo quản vẫn còn tồn tại, gây ảnh hưởng nghiêm trọng đến sức khỏe người tiêu dùng. Trước thực trạng đó, xu hướng sử dụng **thực phẩm sạch, an toàn và có nguồn gốc rõ ràng** ngày càng được người dân quan tâm và lựa chọn.

Cùng với sự phát triển mạnh mẽ của công nghệ thông tin và Internet, thói quen mua sắm của người tiêu dùng cũng dần thay đổi. Việc mua sắm trực tuyến thông qua các website và nền tảng thương mại điện tử ngày càng trở nên phổ biến nhờ tính tiện lợi, nhanh chóng và tiết kiệm thời gian. Đối với lĩnh vực kinh doanh thực phẩm sạch, website không chỉ là kênh giới thiệu sản phẩm mà còn là công cụ quan trọng giúp doanh nghiệp tiếp cận khách hàng, xây dựng thương hiệu và nâng cao hiệu quả kinh doanh.

Mặc dù thị trường thực phẩm sạch tại Việt Nam đang phát triển mạnh mẽ, tuy nhiên nhiều cơ sở kinh doanh vẫn chưa có hệ thống website chuyên nghiệp hoặc việc quản lý còn thủ công, thiếu đồng bộ. Điều này dẫn đến khó khăn trong việc quản lý sản phẩm, đơn hàng, khách hàng cũng như cung cấp đầy đủ thông tin về nguồn gốc và chất lượng sản phẩm đến người tiêu dùng.

Xuất phát từ thực tế đó, tôi quyết định lựa chọn đề tài **“Xây dựng website giới thiệu và kinh doanh thực phẩm sạch HTFood sử dụng Node.js, Reacte.js, Mysql”** nhằm nghiên cứu, thiết kế và triển khai một hệ thống website hỗ trợ hoạt động kinh doanh thực phẩm sạch. Đề tài hướng tới việc xây dựng một giải pháp giúp doanh nghiệp quản lý hiệu quả, đồng thời mang lại trải nghiệm mua sắm tiện lợi và an toàn cho người tiêu dùng.

**2. Mục tiêu nghiên cứu**

Đề tài được thực hiện nhằm hướng đến các mục tiêu chính sau:

* Nghiên cứu cơ sở lý thuyết về lĩnh vực thực phẩm sạch và các mô hình website thương mại điện tử hiện nay.
* Phân tích yêu cầu và thiết kế hệ thống website giới thiệu và kinh doanh thực phẩm sạch phù hợp với nhu cầu thực tế.
* Xây dựng và triển khai website hỗ trợ quản lý sản phẩm, khách hàng, giỏ hàng, đơn hàng và thông tin kinh doanh.
* Đánh giá hiệu quả của hệ thống và đề xuất các hướng phát triển, mở rộng chức năng trong tương lai.

**3. Phạm vi nghiên cứu**

* Đề tài tập trung nghiên cứu lĩnh vực kinh doanh thực phẩm sạch tại Việt Nam, bao gồm các sản phẩm như rau củ quả, thực phẩm tươi sống và thực phẩm chế biến an toàn.
* Nghiên cứu và xây dựng website giới thiệu và kinh doanh thực phẩm sạch với các chức năng cơ bản như hiển thị sản phẩm, đặt hàng trực tuyến, quản lý khách hàng và đơn hàng.
* Đề tài tập trung vào việc ứng dụng công nghệ thông tin nhằm nâng cao hiệu quả quản lý và hoạt động kinh doanh.

**4. Phương pháp nghiên cứu**

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

* **Phương pháp** nghiên **cứu tài liệu:** Thu thập, tìm hiểu các tài liệu liên quan đến thực phẩm sạch, thương mại điện tử và các mô hình website kinh doanh hiện nay.
* **Phương** pháp **khảo sát thực tế:** Tìm hiểu nhu cầu mua sắm thực phẩm sạch của người tiêu dùng và thực trạng kinh doanh của các cơ sở thực phẩm sạch.
* **Phương** pháp **phân tích và thiết kế hệ thống:** Phân tích yêu cầu hệ thống, thiết kế cơ sở dữ liệu, kiến trúc hệ thống và giao diện website.
* **Phương** pháp **thực nghiệm:** Xây dựng, kiểm thử và đánh giá hoạt động của website so với các yêu cầu ban đầu.

**5. Cấu trúc của đồ án**

Nội dung của đồ án được trình bày gồm các chương chính như sau:

**Chương 1:** Nghiên cứu cơ sở lý thuyết và tổng quan về đề tài

**Chương 2:** Phân tích và thiết kế hệ thống

**Chương 3:** Xây dựng và triển khai hệ thống

**Kết luận và hướng phát triển**

**Danh mục tài liệu tham khảo**

# CHƯƠNG 1. NGHIÊN CỨU CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN VỀ ĐỀ TÀI

## Cơ sở lý thuyết

**1.1.1. Tổng quan về lập trình thiết kế giao diện website**

Trong quá trình phát triển các hệ thống website hiện nay, đặc biệt là các website thương mại điện tử, việc thiết kế giao diện người dù ng đóng vai trò vô cùng quan trọng. Giao diện website không chỉ quyết định đến tính thẩm mỹ mà còn ảnh hưởng trực tiếp đến trải nghiệm người dùng, khả năng tiếp cận thông tin, mức độ tin cậy cũng như hiệu quả kinh doanh của hệ thống.

Lập trình thiết kế giao diện website là quá trình xây dựng phần hiển thị của website, nơi người dùng trực tiếp tương tác. Phần giao diện này thường được phát triển dựa trên các công nghệ nền tảng như HTML, CSS và Javascript. Đây là ba công nghệ cốt lõi, không thể thiếu trong lĩnh vực phát triển web hiện đại, giúp xây dựng cấu trúc, định dạng nội dung và tạo ra các chức năng tương tác cho website.

**a) Giới thiệu về HTML, HTML5**

HTML có tên đầy đủ là HyperText Markup Language, nghĩa là ngôn ngữ đánh dấu siêu văn bản. HTML được sử dụng để tạo ra và cấu trúc các thành phần cơ bản của trang web và ứng dụng web như tiêu đề, đoạn văn, hình ảnh, liên kết, bảng biểu, danh sách và các khối nội dung khác.

HTML không phải là một ngôn ngữ lập trình mà chỉ là một ngôn ngữ đánh dấu. Điều này có nghĩa HTML không thể xử lý các chức năng động hay logic nghiệp vụ. Thay vào đó, HTML chỉ đóng vai trò định dạng và trình bày nội dung hiển thị trên trình duyệt. Có thể hiểu HTML tương tự như phần mềm Microsoft Word, dùng để soạn thảo và định dạng văn bản cho website.

**Cấu trúc của HTML:** Một tệp HTML được cấu thành từ các phần tử HTML và các cặp thẻ. Mỗi thẻ HTML có chức năng và ý nghĩa riêng, được đặt trong dấu ngoặc nhọn “< >”. Các chữ nằm giữa các dấu ngoặc này được gọi là phần tử HTML.

Về cơ bản, cấu trúc của một trang HTML bao gồm ba phần chính:

* **Phần khai báo loại tài liệu (**<!DOCTYPE html>**)**: Xuất hiện ở đầu hoặc trên cùng của tệp HTML. Phần khai báo này giúp trình duyệt xác định phiên bản HTML đang được sử dụng để hiển thị trang web.
* **Phần khai báo đầu trang (**<head>**)**: Bắtđầu bằng thẻ <head> và kết thúc bằng thẻ </head>. Phần này chứa các thông tin meta, tiêu đề trang, liên kết đến các tệp CSS, Javascript và các dữ liệu phục vụ cho việc tối ưu hóa công cụ tìm kiếm (SEO). Trong đó, tiêu đề trang được hiển thị trên thanh điều hướng của trình duyệt.
* **Phần nội dung trang (**<body>**)**: Bắt đầu bằng thẻ <body> và kết thúc bằng thẻ </body>. Phần này chứa toàn bộ nội dung hiển thị cho người dùng như logo, menu điều **hướng,** danh sách sản phẩm thực phẩm sạch, hình ảnh minh họa, bài viết giới thiệu, thông tin liên hệ và chân trang.

**Ưu điểm của HTML:** HTML là công cụ hiệu quả giúp người dùng thiết kế và tạo nội dung nhanh chóng cho website. Một số ưu điểm nổi bật của HTML bao gồm:

* Kho tài nguyên phong phú với cộng đồng người dùng rộng lớn
* Mã nguồn mở, miễn phí
* Hoạt động ổn định trên hầu hết các trình duyệt
* Cú pháp đơn giản, dễ học và dễ sử dụng
* Có thể tích hợp với nhiều ngôn ngữ và nền tảng khác như PHP, Java, Node.js, Ruby
* Tuân theo tiêu chuẩn do World Wide Web Consortium (W3C) quy định

**Nhược điểm của HTML:** bên cạnh những ưu điểm trên, HTML vẫn tồn tại một số hạn chế:

* Chỉ phù hợp để xây dựng website tĩnh
* Không hỗ trợ xử lý logic hay tương tác phức tạp
* Phụ thuộc nhiều vào trình duyệt
* Cần kết hợp với CSS và Javascript để xây dựng website hoàn chỉnh

HTML5 là phiên bản thứ năm của HTML, được công bố bởi World Wide Web Consortium (W3C). HTML5 là sự kết hợp và cải tiến từ HTML4, XHTML, DOM và Javascript. HTML5 hỗ trợ chạy trên hầu hết các trình duyệt hiện đại và là phiên bản HTML mới nhất hiện nay.

HTML5 được cải tiến với nhiều thẻ và tính năng mới như hỗ trợ video, audio, canvas, SVG, Web Storage và Geolocation API. Ngoài ra, HTML5 còn hỗ trợ WebSocket, cho phép giao tiếp song song giữa server và client, giúp xây dựng các website có tính tương tác cao. HTML5 góp phần nâng cao hiệu suất, khả năng hiển thị và trải nghiệm người dùng cho website thương mại điện tử.

**b) Giới thiệu về CSS3**

CSS3 là phiên bản thứ ba và cũng là phiên bản mới nhất của CSS (Cascading Style Sheets). CSS3 được sử dụng để định dạng và tạo phong cách cho website, bao gồm bố cục, màu sắc, font chữ, hiệu ứng và khả năng hiển thị trên nhiều loại thiết bị khác nhau

CSS đóng vai trò như một công cụ giúp website trở nên sinh động và trực quan hơn. Nếu một website không sử dụng CSS thì nội dung chỉ đơn thuần là văn bản đen trắng, thiếu tính thẩm mỹ và trải nghiệm người dùng.

CSS3 mang lại nhiều ưu điểm nổi bật như:

* Dễ dàng tùy chỉnh giao diện website
* Hỗ trợ thiết kế responsive cho nhiều thiết bị
* Giảm thiểu việc sử dụng hình ảnh, tăng tốc độ tải trang
* Hỗ trợ hiệu ứng và animation hiện đại
* Được hầu hết các trình duyệt hiện đại hỗ trợ

Sự khác biệt giữa CSS và CSS3: CSS3 được chia thành các module riêng biệt, giúp việc học và sử dụng trở nên dễ dàng hơn. Ngoài ra, CSS3 còn bổ sung nhiều tính năng mới phục vụ thiết kế website thương mại điện tử hiện đại.

**c) Tổng quan về ngôn ngữ lập trình Javascript**

Javascript là ngôn ngữ lập trình kịch bản phổ biến trong phát triển web hiện nay. Javascript được tích hợp trực tiếp vào HTML, cho phép xây dựng các website động, xử lý sự kiện, tương tác với người dùng và giao tiếp với server thông qua các API.

Javascript có thể hoạt động ở cả phía client và phía server (thông qua Node.js), giúp xây dựng các hệ thống web hoàn chỉnh.

**Ưu** điểm **của Javacript:**

* Dễ học, cú pháp linh hoạt
* Hoạt động trên nhiều trình duyệt
* Tốc độ xử lý nhanh
* Tăng khả năng tương tác cho website
* Phù hợp xây dựng website thương mại điện tử

**Nhược** điểm **của Javascript:**

* Dễ bị khai thác bảo mật
* Phụ thuộc vào trình duyệt
* Không hỗ trợ đa luồng
* Có thể gây lỗi nếu không kiểm soát tốt

**d) Framework Bootstrap**

Bootstrap là framework front-end miễn phí, hỗ trợ phát triển website nhanh chóng và hiệu quả. Bootstrap cung cấp các mẫu thiết kế dựa trên HTML và CSS cho form, button, bảng biểu, điều hướng và nhiều thành phần giao diện khác.

Bootstrap hỗ trợ thiết kế responsive, giúp website hiển thị tốt trên nhiều thiết bị như máy tính, máy tính bảng và điện thoại di động. Nhờ đó, Bootstrap rất phù hợp để xây dựng website kinh doanh thực phẩm sạch.

**1.1.2. Tổng quan về hệ quản trị cơ sở dữ liệu MySQL**

Trong các hệ thống thông tin hiện đại, đặc biệt là các website thương mại điện tử, cơ sở dữ liệu đóng vai trò vô cùng quan trọng trong việc lưu trữ, quản lý và khai thác thông tin. Hệ quản trị cơ sở dữ liệu (Database Management System – DBMS) là thành phần trung tâm giúp hệ thống vận hành ổn định, đảm bảo dữ liệu được tổ chức khoa học, an toàn và nhất quán. Trong số các hệ quản trị cơ sở dữ liệu hiện nay, MySQL là một trong những hệ quản trị cơ sở dữ liệu quan hệ phổ biến và được sử dụng rộng rãi nhất.

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS), sử dụng ngôn ngữ truy vấn có cấu trúc SQL (Structured Query Language) để thực hiện các thao tác như tạo bảng, truy vấn dữ liệu, cập nhật dữ liệu và quản lý quyền truy cập. MySQL được phát triển và duy trì bởi Oracle Corporation, một trong những tập đoàn công nghệ hàng đầu thế giới, do đó MySQL luôn được cập nhật, cải tiến và đảm bảo về mặt bảo mật cũng như hiệu năng.

MySQL hoạt động dựa trên mô hình cơ sở dữ liệu quan hệ, trong đó dữ liệu được tổ chức dưới dạng các bảng (table). Mỗi bảng bao gồm các cột (column) biểu diễn các thuộc tính và các hàng (row) biểu diễn các bản ghi dữ liệu. Các bảng trong cơ sở dữ liệu không tồn tại độc lập mà có thể liên kết với nhau thông qua các mối quan hệ được xác định bởi khóa chính (Primary Key) và khóa ngoại (Foreign Key). Nhờ cơ chế này, dữ liệu được tổ chức một cách chặt chẽ, hạn chế tối đa sự trùng lặp và đảm bảo tính toàn vẹn dữ liệu trong quá trình sử dụng.

Khóa chính là thuộc tính hoặc tập thuộc tính dùng để định danh duy nhất mỗi bản ghi trong một bảng. Trong khi đó, khóa ngoại là thuộc tính dùng để liên kết dữ liệu giữa các bảng với nhau. Việc sử dụng khóa chính và khóa ngoại giúp hệ thống dễ dàng quản lý mối quan hệ giữa các thực thể như khách hàng – đơn hàng – sản phẩm, đồng thời đảm bảo rằng các dữ liệu liên quan luôn nhất quán và không bị sai lệch.

Một trong những ưu điểm nổi bật của MySQL là hiệu suất xử lý cao. MySQL có khả năng xử lý nhanh các truy vấn dữ liệu lớn, hỗ trợ nhiều cơ chế tối ưu truy vấn như indexing, caching và tối ưu hóa bộ nhớ. Nhờ đó, MySQL có thể đáp ứng tốt các hệ thống có lượng truy cập lớn và nhiều người dùng truy cập đồng thời, đặc biệt phù hợp với các website thương mại điện tử.

Bên cạnh hiệu suất, MySQL còn nổi bật với tính ổn định và độ tin cậy cao. Hệ quản trị này hỗ trợ nhiều cơ chế sao lưu và phục hồi dữ liệu, giúp hạn chế rủi ro mất dữ liệu trong trường hợp xảy ra sự cố hệ thống. Ngoài ra, MySQL còn hỗ trợ phân quyền người dùng, giúp quản lý quyền truy cập dữ liệu một cách linh hoạt và an toàn, đảm bảo rằng mỗi người dùng chỉ có thể truy cập vào các chức năng và dữ liệu được cấp phép.

Một ưu điểm quan trọng khác của MySQL là tính mã nguồn mở và khả năng triển khai linh hoạt. MySQL có thể được cài đặt và sử dụng miễn phí, phù hợp với các doanh nghiệp vừa và nhỏ cũng như các đề tài nghiên cứu, đồ án sinh viên. Đồng thời, MySQL có thể hoạt động trên nhiều hệ điều hành khác nhau như Windows, Linux và macOS, giúp việc triển khai hệ thống trở nên thuận tiện hơn.

Trong đề tài “Website giới thiệu và kinh doanh thực phẩm sạch HTFood sử dụng Node.js, React.js, Mysql”, MySQL được lựa chọn làm hệ quản trị cơ sở dữ liệu chính để lưu trữ và quản lý toàn bộ dữ liệu của hệ thống. Các dữ liệu quan trọng như thông tin sản phẩm thực phẩm sạch, danh mục sản phẩm, thông tin khách hàng, giỏ hàng, đơn hàng, chi tiết đơn hàng, thanh toán, thông tin người dùng và phân quyền quản trị đều được lưu trữ trong MySQL. Việc sử dụng MySQL giúp hệ thống quản lý dữ liệu tập trung, chính xác và thuận tiện cho việc truy vấn cũng như báo cáo.

Ngoài ra, MySQL có khả năng tích hợp tốt với nhiều ngôn ngữ lập trình và công nghệ khác nhau như PHP, Java, Python và đặc biệt là Node.js. Điều này giúp quá trình xây dựng hệ thống backend trở nên linh hoạt và hiệu quả hơn. Thông qua các thư viện kết nối cơ sở dữ liệu, Node.js có thể dễ dàng giao tiếp với MySQL để thực hiện các thao tác thêm, sửa, xóa và truy vấn dữ liệu.

Nhờ những ưu điểm về hiệu năng, tính ổn định, khả năng mở rộng và tính phổ biến, MySQL là lựa chọn phù hợp và hiệu quả cho việc xây dựng website giới thiệu và kinh doanh thực phẩm sạch trong đề tài này. Việc sử dụng MySQL không chỉ đáp ứng tốt các yêu cầu hiện tại của hệ thống mà còn tạo nền tảng vững chắc cho việc mở rộng và phát triển hệ thống trong tương lai.

**1.1.3. Tổng quan về ngôn ngữ lập trình phía máy chủ Node.js**

Trong những năm gần đây, cùng với sự phát triển mạnh mẽ của các ứng dụng web và hệ thống thương mại điện tử, yêu cầu đối với các nền tảng lập trình phía máy chủ ngày càng trở nên khắt khe hơn. Các hệ thống không chỉ cần xử lý dữ liệu chính xác mà còn phải đảm bảo hiệu suất cao, khả năng mở rộng tốt và đáp ứng đồng thời nhiều người dùng truy cập. Trong bối cảnh đó, Node.js đã trở thành một trong những nền tảng lập trình phía máy chủ được sử dụng rộng rãi và phổ biến nhất hiện nay.

Node.js là một nền tảng lập trình phía máy chủ (server-side platform) được xây dựng dựa trên ngôn ngữ Javascript và sử dụng Javascript Engine V8 của Google Chrome. Khác với mô hình truyền thống chỉ sử dụng Javascript ở phía trình duyệt, Node.js cho phép Javascript được thực thi trực tiếp trên máy chủ. Điều này giúp thống nhất ngôn ngữ lập trình giữa frontend và backend, tạo điều kiện thuận lợi cho việc phát triển và bảo trì hệ thống.

Một trong những đặc điểm nổi bật nhất của Node.js là mô hình xử lý bất đồng bộ (asynchronous) và hướng sự kiện (event-driven). Node.js sử dụng mô hình non-blocking I/O, nghĩa là các thao tác vào/ra như đọc ghi tệp tin, truy vấn cơ sở dữ liệu hay xử lý request từ người dùng sẽ không làm “chặn” luồng xử lý chính của hệ thống. Thay vào đó, các tác vụ này được đưa vào hàng đợi và xử lý khi có sự kiện tương ứng xảy ra. Nhờ cơ chế này, Node.js có thể xử lý đồng thời hàng nghìn kết nối mà không cần tạo nhiều luồng (thread) như các nền tảng truyền thống.

Cơ chế event-driven của Node.js giúp tối ưu tài nguyên hệ thống và cải thiện đáng kể hiệu suất xử lý. Khi một yêu cầu từ người dùng được gửi đến server, Node.js sẽ tiếp nhận yêu cầu, thực hiện các thao tác cần thiết và trả về kết quả thông qua các callback, promise hoặc async/await. Trong thời gian chờ đợi phản hồi từ cơ sở dữ liệu hoặc dịch vụ bên ngoài, Node.js vẫn có thể tiếp tục xử lý các yêu cầu khác, nhờ đó nâng cao khả năng đáp ứng của hệ thống.

Node.js đặc biệt phù hợp với các hệ thống web có lượng truy cập lớn và nhiều thao tác đồng thời như website thương mại điện tử. Trong các hệ thống này, người dùng thường xuyên thực hiện các hành động như xem danh sách sản phẩm, tìm kiếm, thêm sản phẩm vào giỏ hàng, đặt hàng và thanh toán. Những thao tác này yêu cầu hệ thống backend phải phản hồi nhanh, ổn định và không bị gián đoạn khi có nhiều người dùng truy cập cùng lúc. Nhờ mô hình xử lý bất đồng bộ, Node.js đáp ứng tốt các yêu cầu này.

Trong đề tài “Website giới thiệu và kinh doanh thực phẩm sạch HTFood sử dụng Node.js, React.js, Mysql”, Node.js được lựa chọn làm nền tảng lập trình phía máy chủ để xây dựng hệ thống backend. Node.js đóng vai trò trung tâm trong việc xử lý toàn bộ logic nghiệp vụ của hệ thống, bao gồm quản lý người dùng, xác thực và phân quyền, quản lý sản phẩm, giỏ hàng, đơn hàng, thanh toán và các chức năng quản trị. Việc sử dụng Node.js giúp hệ thống hoạt động linh hoạt, hiệu quả và dễ dàng mở rộng trong tương lai.

Node.js thường được kết hợp với framework Express.js để xây dựng các ứng dụng web và API RESTful. Express.js là một framework nhẹ, đơn giản nhưng mạnh mẽ, giúp lập trình viên dễ dàng định nghĩa các route, xử lý request/response và xây dựng cấu trúc ứng dụng rõ ràng. Trong đề tài này, Express.js được sử dụng để xây dựng các API phục vụ frontend, cho phép frontend gửi yêu cầu và nhận dữ liệu từ backend thông qua giao thức HTTP.

Bên cạnh đó, Node.js có hệ sinh thái thư viện phong phú thông qua trình quản lý gói npm (Node Package Manager). npm cung cấp hàng nghìn thư viện hỗ trợ cho nhiều chức năng khác nhau như xác thực người dùng, mã hóa mật khẩu, kết nối cơ sở dữ liệu, xử lý file, gửi email và bảo mật hệ thống. Việc sử dụng các thư viện này giúp giảm thời gian phát triển, đồng thời đảm bảo hệ thống tuân theo các chuẩn kỹ thuật phổ biến.

Một ưu điểm quan trọng khác của Node.js là khả năng mở rộng (scalability). Node.js cho phép dễ dàng triển khai hệ thống theo mô hình microservices hoặc mở rộng theo chiều ngang (horizontal scaling) khi lượng người dùng tăng lên. Điều này rất phù hợp với các website thương mại điện tử, nơi nhu cầu sử dụng có thể tăng cao theo thời gian hoặc theo các chương trình khuyến mãi, sự kiện bán hàng.

Ngoài ra, việc sử dụng Javascript cho cả frontend và backend giúp đội ngũ phát triển dễ dàng chia sẻ kiến thức, giảm chi phí đào tạo và tăng tính nhất quán trong quá trình phát triển hệ thống. Điều này đặc biệt có ý nghĩa đối với các đề tài nghiên cứu và đồ án sinh viên, nơi thời gian và nguồn lực phát triển thường có hạn.

Với những ưu điểm vượt trội về hiệu suất, khả năng xử lý bất đồng bộ, hệ sinh thái phong phú và khả năng mở rộng linh hoạt, Node.js là lựa chọn phù hợp để xây dựng hệ thống backend cho website giới thiệu và kinh doanh thực phẩm sạch. Việc áp dụng Node.js trong đề tài không chỉ đáp ứng tốt các yêu cầu chức năng hiện tại mà còn tạo nền tảng vững chắc cho việc phát triển và mở rộng hệ thống trong tương lai.

**1.1.4. Tổng quan về công nghệ lập trình phía người dùng (Frontend)**

Frontend là phần giao diện người dùng của website, nơi người dùng trực tiếp tương tác thông qua trình duyệt web như Google Chrome, Microsoft Edge hoặc Mozilla Firefox. Đây là lớp hiển thị toàn bộ thông tin và chức năng của hệ thống, đóng vai trò cầu nối giữa người dùng và hệ thống backend. Một frontend được thiết kế tốt không chỉ giúp website có giao diện đẹp mắt mà còn đảm bảo khả năng sử dụng cao, thao tác thuận tiện và mang lại trải nghiệm tích cực cho người dùng.

Trong các hệ thống website thương mại điện tử, frontend giữ vai trò đặc biệt quan trọng vì đây là nơi người dùng tiếp cận sản phẩm, tìm kiếm thông tin, đưa ra quyết định mua hàng và thực hiện các thao tác thanh toán. Do đó, việc lựa chọn công nghệ frontend phù hợp sẽ ảnh hưởng trực tiếp đến hiệu quả kinh doanh cũng như mức độ hài lòng của khách hàng.

Trong đề tài “Website giới thiệu và kinh doanh thực phẩm sạch HTFood sử dụng Node.js, Reacte.js, Mysql”, công nghệ frontend được lựa chọn là React.js kết hợp với Vite. Đây là sự kết hợp phổ biến trong các dự án web hiện đại nhờ khả năng xây dựng giao diện linh hoạt, hiệu suất cao và dễ mở rộng.

React.js là một thư viện Javascript được phát triển bởi Facebook, chuyên dùng để xây dựng giao diện người dùng theo mô hình component. Thay vì xây dựng giao diện dưới dạng một khối mã nguồn lớn, React cho phép chia nhỏ giao diện thành các component độc lập. Mỗi component đại diện cho một phần chức năng cụ thể của website như header, footer, menu, danh sách sản phẩm, giỏ hàng, trang đăng nhập hoặc trang thanh toán.

Việc chia giao diện thành các component mang lại nhiều lợi ích quan trọng. Thứ nhất, mã nguồn trở nên rõ ràng và có cấu trúc, giúp lập trình viên dễ đọc, dễ hiểu và dễ bảo trì. Thứ hai, các component có thể được tái sử dụng ở nhiều vị trí khác nhau trong hệ thống, giúp giảm thiểu trùng lặp mã nguồn và tiết kiệm thời gian phát triển. Thứ ba, khi cần chỉnh sửa hoặc mở rộng giao diện, lập trình viên chỉ cần thay đổi component liên quan mà không ảnh hưởng đến toàn bộ hệ thống.

React.js sử dụng cơ chế Virtual DOM, cho phép tối ưu hiệu suất hiển thị giao diện. Virtual DOM là một bản sao ảo của DOM thật, được React sử dụng để so sánh sự khác biệt khi dữ liệu thay đổi. Thay vì cập nhật toàn bộ giao diện mỗi khi có thay đổi, React chỉ cập nhật những phần giao diện thực sự bị ảnh hưởng. Cơ chế này giúp website hoạt động mượt mà, giảm thời gian tải trang và nâng cao trải nghiệm người dùng, đặc biệt khi xử lý danh sách sản phẩm lớn hoặc các thao tác tương tác liên tục.

Trong website giới thiệu và kinh doanh thực phẩm sạch, React.js được sử dụng để xây dựng toàn bộ giao diện người dùng, bao gồm:

* Giao diện trang chủ giới thiệu website và các nội dung nổi bật
* Giao diện danh sách sản phẩm theo từng danh mục
* Giao diện chi tiết sản phẩm thực phẩm sạch với hình ảnh, mô tả và giá bán
* Chức năng tìm kiếm và lọc sản phẩm
* Giao diện giỏ hàng cho phép thêm, sửa và xóa sản phẩm
* Giao diện thanh toán và xác nhận đơn hàng
* Quản lý trạng thái đăng nhập, đăng xuất của người dùng

Ngoài ra, React còn hỗ trợ quản lý trạng thái (state management), giúp đồng bộ dữ liệu giữa các component. Ví dụ, khi người dùng thêm sản phẩm vào giỏ hàng, thông tin giỏ hàng sẽ được cập nhật ngay lập tức trên toàn bộ giao diện. Điều này giúp đảm bảo tính nhất quán của dữ liệu và nâng cao trải nghiệm người dùng.

Bên cạnh React.js, đề tài sử dụng Vite làm công cụ hỗ trợ khởi tạo và chạy dự án frontend. Vite là một công cụ build hiện đại, được thiết kế nhằm khắc phục những hạn chế của các công cụ build truyền thống. Vite giúp giảm đáng kể thời gian khởi động dự án và tăng tốc quá trình phát triển nhờ cơ chế load module theo yêu cầu.

Một trong những ưu điểm nổi bật của Vite là khả năng hỗ trợ hot module replacement (HMR) nhanh và hiệu quả. Khi lập trình viên chỉnh sửa mã nguồn, giao diện sẽ được cập nhật gần như tức thì mà không cần tải lại toàn bộ trang. Điều này giúp tiết kiệm thời gian, nâng cao hiệu suất làm việc và hỗ trợ quá trình phát triển frontend diễn ra liên tục, liền mạch.

Sự kết hợp giữa React.js và Vite mang lại nhiều lợi ích cho đề tài, bao gồm:

* Tăng tốc độ phát triển giao diện người dùng
* Giảm thời gian build và chạy ứng dụng
* Cải thiện hiệu suất hiển thị website
* Dễ dàng mở rộng và nâng cấp giao diện trong tương lai

Frontend trong đề tài được thiết kế theo hướng thân thiện với người dùng, giao diện rõ ràng, bố cục hợp lý và dễ thao tác. Website không chỉ chú trọng đến yếu tố thẩm mỹ mà còn tập trung vào khả năng sử dụng, giúp người dùng dễ dàng tìm kiếm, lựa chọn và mua sản phẩm thực phẩm sạch một cách thuận tiện và nhanh chóng.

**1.1.5. Công cụ sử dụng trong quá trình phát triển hệ thống**

Để xây dựng website giới thiệu và kinh doanh thực phẩm sạch một cách hoàn chỉnh, đề tài đã sử dụng nhiều công cụ và môi trường phát triển khác nhau. Mỗi công cụ đều đảm nhiệm một vai trò riêng, hỗ trợ cho từng giai đoạn trong quá trình phân tích, thiết kế, phát triển, kiểm thử và hoàn thiện hệ thống. Việc lựa chọn đúng công cụ giúp đảm bảo chất lượng sản phẩm, nâng cao hiệu suất làm việc và giảm thiểu sai sót trong quá trình triển khai.

**a) Visual Studio Code**

Visual Studio Code là môi trường soạn thảo mã nguồn chính được sử dụng trong suốt quá trình thực hiện đề tài “Website giới thiệu và kinh doanh thực phẩm sạch HTFood sử dụng Node.js, React.js, Mysql”. Đây là công cụ phát triển phần mềm hiện đại, được nhiều lập trình viên và sinh viên công nghệ thông tin lựa chọn nhờ giao diện trực quan, dễ sử dụng, dung lượng nhẹ và khả năng hoạt động ổn định trên nhiều hệ điều hành khác nhau như Windows, Linux và macOS.

Visual Studio Code hỗ trợ mạnh mẽ cho các ngôn ngữ và công nghệ được sử dụng trong đề tài như HTML, CSS, Javascript, React.js và Node.js. Trình soạn thảo này cho phép lập trình viên làm việc đồng thời với cả mã nguồn frontend và backend trong cùng một môi trường, giúp việc phát triển hệ thống trở nên thống nhất và thuận tiện hơn. Ngoài ra, VS Code còn hỗ trợ nhiều định dạng tệp cấu hình và tệp dữ liệu, giúp quản lý toàn bộ dự án trong một không gian làm việc duy nhất.

Một trong những ưu điểm nổi bật của Visual Studio Code là hệ thống tiện ích mở rộng (Extensions) vô cùng phong phú. Thông qua các extension, lập trình viên có thể mở rộng thêm nhiều tính năng hỗ trợ như gợi ý mã thông minh, kiểm tra lỗi cú pháp theo thời gian thực, định dạng mã nguồn tự động, hỗ trợ framework React.js, hỗ trợ Node.js và quản lý cơ sở dữ liệu. Điều này giúp giảm đáng kể thời gian viết mã và hạn chế các lỗi thường gặp trong quá trình lập trình.

Trong quá trình thực hiện đề tài, Visual Studio Code được sử dụng để:

* Viết và chỉnh sửa mã nguồn frontend và backend của website.
* Tổ chức và quản lý cấu trúc thư mục dự án một cách khoa học.
* Kiểm tra lỗi cú pháp, lỗi logic và các cảnh báo trong quá trình lập trình.
* Định dạng mã nguồn theo chuẩn, giúp mã nguồn rõ ràng và dễ đọc.
* Tích hợp Git để quản lý phiên bản và theo dõi lịch sử thay đổi của dự án.

Bên cạnh đó, Visual Studio Code còn hỗ trợ tích hợp Terminal trực tiếp trong trình soạn thảo, cho phép lập trình viên chạy các lệnh Node.js, npm, yarn hoặc các lệnh Git mà không cần phải chuyển sang cửa sổ dòng lệnh riêng biệt. Tính năng này giúp quá trình phát triển và kiểm thử hệ thống diễn ra nhanh chóng và liền mạch hơn.

Ngoài ra, VS Code còn cung cấp các công cụ hỗ trợ debug mạnh mẽ, giúp lập trình viên dễ dàng theo dõi luồng thực thi của chương trình, kiểm tra giá trị biến và xác định nguyên nhân gây lỗi. Điều này đặc biệt hữu ích trong quá trình phát triển backend với Node.js, khi cần xử lý các logic nghiệp vụ phức tạp và tương tác với cơ sở dữ liệu MySQL.

Nhờ những ưu điểm vượt trội về giao diện, tính linh hoạt, khả năng mở rộng và hỗ trợ lập trình hiệu quả, Visual Studio Code đã góp phần quan trọng vào việc nâng cao chất lượng mã nguồn cũng như hiệu quả làm việc trong quá trình xây dựng website giới thiệu và kinh doanh thực phẩm sạch.

**b) XAMPP và hệ quản trị cơ sở dữ liệu MySQL**

XAMPP là một bộ công cụ tích hợp hỗ trợ cài đặt và quản lý các dịch vụ máy chủ cục bộ, bao gồm Apache, MySQL và phpMyAdmin. Đây là môi trường phát triển phổ biến, đặc biệt phù hợp với sinh viên và các dự án nghiên cứu, đồ án học tập nhờ khả năng cài đặt đơn giản, dễ sử dụng và hoàn toàn miễn phí.

Trong đề tài “Website giới thiệu và kinh doanh thực phẩm sạch HTFood sử dụng Node.js, React.js, Mysql”, XAMPP được sử dụng để xây dựng môi trường làm việc cục bộ (localhost), phục vụ cho quá trình phát triển, kiểm thử và chạy thử hệ thống trước khi triển khai chính thức. Việc sử dụng môi trường cục bộ giúp sinh viên dễ dàng kiểm soát dữ liệu, nhanh chóng phát hiện lỗi và chỉnh sửa hệ thống mà không phụ thuộc vào máy chủ bên ngoài.

Thành phần Apache trong XAMPP đóng vai trò là web server, giúp tiếp nhận và xử lý các yêu cầu từ phía trình duyệt. Apache hỗ trợ giao tiếp giữa frontend và backend thông qua giao thức HTTP, đảm bảo website có thể hoạt động ổn định trong quá trình phát triển.

MySQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) chính được sử dụng trong đề tài. Toàn bộ dữ liệu của website thực phẩm sạch đều được lưu trữ và quản lý trong MySQL, bao gồm:

* Thông tin sản phẩm thực phẩm sạch
* Thông tin danh mục và nhà cung cấp
* Thông tin khách hàng và tài khoản người dùng
* Thông tin giỏ hàng và chi tiết giỏ hàng
* Thông tin đơn hàng, chi tiết đơn hàng
* Thông tin thanh toán và trạng thái giao dịch
* Thông tin blog, bài viết và liên hệ

MySQL cho phép tổ chức dữ liệu dưới dạng các bảng có mối quan hệ chặt chẽ với nhau thông qua khóa chính (Primary Key) và khóa ngoại (Foreign Key). Nhờ đó, dữ liệu được quản lý khoa học, tránh trùng lặp, đảm bảo tính toàn vẹn và nhất quán trong suốt quá trình vận hành hệ thống.

Bên cạnh đó, MySQL hỗ trợ ngôn ngữ truy vấn SQL mạnh mẽ, cho phép thực hiện các thao tác thêm, sửa, xóa và truy vấn dữ liệu một cách nhanh chóng và chính xác. Điều này rất quan trọng đối với website thương mại điện tử, nơi dữ liệu thường xuyên thay đổi theo hoạt động mua bán của người dùng.

phpMyAdmin – một thành phần đi kèm trong XAMPP – được sử dụng để quản lý cơ sở dữ liệu MySQL thông qua giao diện web. Công cụ này giúp sinh viên dễ dàng tạo bảng, thiết lập mối quan hệ giữa các bảng, nhập dữ liệu mẫu, kiểm tra dữ liệu và thực hiện các truy vấn SQL mà không cần thao tác trực tiếp trên dòng lệnh.

Việc kết hợp XAMPP và MySQL giúp quá trình phát triển và kiểm thử website diễn ra thuận lợi, ổn định, đồng thời tạo nền tảng vững chắc cho việc mở rộng và nâng cấp hệ thống trong tương lai.

**c) Postman**

Postman là công cụ hỗ trợ kiểm thử API backend được sử dụng phổ biến trong quá trình phát triển các ứng dụng web hiện đại. Postman cho phép gửi và nhận các request HTTP như GET, POST, PUT, DELETE, từ đó giúp kiểm tra hoạt động của các API một cách trực quan và hiệu quả.

Trong đề tài này, Postman được sử dụng để kiểm thử các API được xây dựng bằng Node.js và Express.js trước khi kết nối với frontend. Việc kiểm thử API độc lập giúp sinh viên dễ dàng phát hiện lỗi logic, lỗi xử lý dữ liệu hoặc lỗi xác thực mà không cần phụ thuộc vào giao diện người dùng.

Cụ thể, Postman được sử dụng để:

* Kiểm thử API đăng ký tài khoản người dùng
* Kiểm thử API đăng nhập và xác thực người dùng
* Kiểm thử phân quyền người dùng (admin, member, customer)
* Kiểm thử các API quản lý sản phẩm (thêm, sửa, xóa, lấy danh sách)
* Kiểm thử các API giỏ hàng và chi tiết giỏ hàng
* Kiểm thử quy trình tạo đơn hàng và xử lý thanh toán
* Kiểm thử các API blog và liên hệ

Postman cho phép thiết lập header, body và các tham số truyền vào API một cách linh hoạt, giúp mô phỏng chính xác các tình huống sử dụng thực tế của người dùng. Ngoài ra, Postman còn hỗ trợ lưu lại các collection API, giúp việc kiểm thử được thực hiện nhanh chóng và có hệ thống.

Việc sử dụng Postman giúp đảm bảo rằng các API backend hoạt động đúng chức năng, ổn định và sẵn sàng kết nối với frontend. Điều này góp phần nâng cao chất lượng hệ thống và giảm thiểu lỗi phát sinh trong giai đoạn triển khai.

**d) Git**

Git là công cụ quản lý phiên bản mã nguồn được sử dụng trong suốt quá trình thực hiện đề tài. Git cho phép lưu lại lịch sử thay đổi của dự án, giúp lập trình viên theo dõi các phiên bản mã nguồn và dễ dàng quay lại trạng thái trước đó khi xảy ra lỗi.

Trong quá trình phát triển website, Git giúp quản lý sự thay đổi của mã nguồn frontend và backend một cách khoa học. Mỗi khi thực hiện một thay đổi quan trọng, mã nguồn sẽ được commit lại, giúp lưu trữ và ghi nhận nội dung chỉnh sửa.

Việc sử dụng Git mang lại nhiều lợi ích như:

* Theo dõi lịch sử phát triển của dự án
* Dễ dàng khôi phục mã nguồn khi xảy ra lỗi
* Giảm rủi ro mất dữ liệu trong quá trình lập trình
* Hỗ trợ làm việc nhóm (trong trường hợp mở rộng dự án)

Git giúp sinh viên rèn luyện kỹ năng làm việc chuyên nghiệp theo quy trình phát triển phần mềm thực tế, đồng thời tạo nền tảng cho việc phát triển các dự án lớn hơn trong tương lai.

**e) Trình duyệt web**

Trình duyệt web là công cụ không thể thiếu trong quá trình phát triển và kiểm thử website. Trong đề tài này, Google Chrome được sử dụng làm trình duyệt chính để kiểm tra giao diện và chức năng của website thực phẩm sạch.

Google Chrome hỗ trợ bộ công cụ Developer Tools mạnh mẽ, cho phép kiểm tra và phân tích giao diện người dùng, cấu trúc HTML, CSS, Javascript cũng như các yêu cầu gửi đến backend thông qua API. Nhờ đó, sinh viên có thể dễ dàng phát hiện lỗi hiển thị, lỗi logic giao diện và các vấn đề về hiệu suất.

Chrome Developer Tools hỗ trợ:

* Kiểm tra lỗi giao diện và căn chỉnh layout
* Theo dõi quá trình tải trang và tối ưu hiệu suất
* Kiểm tra request/response API giữa frontend và backend
* Debug mã Javascript trực tiếp trên trình duyệt

Việc sử dụng trình duyệt web kết hợp với các công cụ phát triển giúp website được hoàn thiện cả về mặt chức năng lẫn trải nghiệm người dùng, đảm bảo hệ thống hoạt động ổn định và đáp ứng tốt nhu cầu của khách hàng.

## Tổng quan về lĩnh vực thực phẩm sạch

**1.2.1. Giới thiệu về thực phẩm sạch**

Trong những năm gần đây, vấn đề an toàn vệ sinh thực phẩm đang trở thành mối quan tâm hàng đầu của xã hội. Thực phẩm bẩn, thực phẩm chứa hóa chất độc hại, không rõ nguồn gốc xuất xứ đã và đang gây ảnh hưởng nghiêm trọng đến sức khỏe con người. Chính vì vậy, nhu cầu sử dụng **thực phẩm sạch** ngày càng gia tăng, đặc biệt tại các khu vực đô thị và các gia đình có trẻ nhỏ, người cao tuổi.

Thực phẩm sạch là các sản phẩm nông sản, thực phẩm tươi sống hoặc thực phẩm chế biến được sản xuất theo quy trình đảm bảo các tiêu chuẩn về an toàn vệ sinh thực phẩm. Các sản phẩm này thường có nguồn gốc rõ ràng, không sử dụng hoặc hạn chế tối đa thuốc bảo vệ thực vật, chất kích thích tăng trưởng và các hóa chất độc hại trong quá trình nuôi trồng và sản xuất.

Bên cạnh yếu tố an toàn, thực phẩm sạch còn mang lại giá trị về mặt dinh dưỡng, giúp nâng cao chất lượng bữa ăn và cải thiện sức khỏe người tiêu dùng. Trước xu hướng sống xanh, sống lành mạnh, việc lựa chọn thực phẩm sạch đã trở thành thói quen tiêu dùng của nhiều người. Điều này mở ra cơ hội phát triển mạnh mẽ cho các mô hình kinh doanh thực phẩm sạch, đặc biệt là các mô hình kinh doanh kết hợp giữa cửa hàng truyền thống và nền tảng trực tuyến.

**1.2.2. Thực trạng kinh doanh thực phẩm sạch tại Việt Nam**

Tại Việt Nam, thị trường thực phẩm sạch đang có sự phát triển nhanh chóng với sự xuất hiện của nhiều cửa hàng, hợp tác xã và doanh nghiệp cung cấp thực phẩm sạch. Các mô hình kinh doanh đa dạng như cửa hàng bán lẻ, chuỗi cửa hàng, bán hàng online qua mạng xã hội và sàn thương mại điện tử ngày càng phổ biến.

Tuy nhiên, bên cạnh những cơ hội phát triển, ngành kinh doanh thực phẩm sạch tại Việt Nam vẫn tồn tại nhiều hạn chế và thách thức. Phần lớn các cơ sở kinh doanh có quy mô nhỏ, quản lý thủ công, chưa có hệ thống quản lý chuyên nghiệp. Việc theo dõi thông tin sản phẩm, tồn kho, đơn hàng và khách hàng chủ yếu được thực hiện bằng sổ sách hoặc các công cụ đơn giản, dẫn đến sai sót và kém hiệu quả.

Ngoài ra, thông tin về nguồn gốc sản phẩm, quy trình sản xuất và chứng nhận an toàn thực phẩm chưa được minh bạch và trình bày rõ ràng đến người tiêu dùng. Điều này khiến khách hàng khó phân biệt giữa thực phẩm sạch thực sự và các sản phẩm gắn mác “sạch” nhưng không đảm bảo chất lượng.

Việc ứng dụng công nghệ thông tin trong hoạt động kinh doanh thực phẩm sạch vẫn còn hạn chế, đặc biệt là trong việc xây dựng website chuyên nghiệp để giới thiệu sản phẩm, bán hàng trực tuyến và chăm sóc khách hàng. Đây chính là vấn đề cần được nghiên cứu và giải quyết nhằm nâng cao hiệu quả kinh doanh và niềm tin của người tiêu dùng.

**1.2.3. Ứng dụng công nghệ trong kinh doanh thực phẩm sạch**

Trong bối cảnh cuộc cách mạng công nghiệp 4.0, việc ứng dụng công nghệ thông tin vào hoạt động kinh doanh là xu hướng tất yếu. Đối với lĩnh vực thực phẩm sạch, công nghệ đóng vai trò quan trọng trong việc nâng cao hiệu quả quản lý và mở rộng kênh tiếp cận khách hàng.

Website thương mại điện tử là một trong những giải pháp công nghệ hiệu quả giúp doanh nghiệp giới thiệu sản phẩm, cung cấp thông tin chi tiết về nguồn gốc, giá cả và hình ảnh sản phẩm đến người tiêu dùng. Thông qua website, khách hàng có thể dễ dàng tìm kiếm, lựa chọn và đặt mua thực phẩm sạch mọi lúc, mọi nơi mà không bị giới hạn về không gian và thời gian.

Bên cạnh đó, hệ thống website còn hỗ trợ quản lý đơn hàng, giỏ hàng, thông tin khách hàng và lịch sử mua sắm. Việc tích hợp các phương thức thanh toán trực tuyến và dịch vụ giao hàng giúp nâng cao trải nghiệm mua sắm, tăng tính tiện lợi và sự hài lòng của khách hàng.

Ứng dụng công nghệ vào kinh doanh thực phẩm sạch không chỉ giúp tối ưu hóa quy trình vận hành mà còn góp phần nâng cao uy tín thương hiệu, tạo lợi thế cạnh tranh cho doanh nghiệp trên thị trường.

**1.2.4. Xu hướng phát triển của thực phẩm sạch**

Dựa trên nhu cầu thực tế của người tiêu dùng và sự phát triển của công nghệ, ngành thực phẩm sạch trong tương lai được dự đoán sẽ phát triển theo nhiều xu hướng mới.

Thứ nhất, **mua sắm trực tuyến** sẽ ngày càng phổ biến. Người tiêu dùng có xu hướng lựa chọn các nền tảng trực tuyến để tiết kiệm thời gian và công sức, đặc biệt là trong bối cảnh nhịp sống hiện đại ngày càng bận rộn.

Thứ hai, **truy xuất nguồn gốc sản phẩm** sẽ trở thành yếu tố quan trọng. Khách hàng không chỉ quan tâm đến giá cả mà còn chú trọng đến thông tin về xuất xứ, quy trình sản xuất và chứng nhận chất lượng của sản phẩm.

Thứ ba, **thanh toán điện tử và giao hàng nhanh** sẽ được ưu tiên phát triển nhằm đáp ứng nhu cầu tiện lợi và nhanh chóng của người tiêu dùng. Các phương thức thanh toán không tiền mặt giúp tăng tính an toàn và minh bạch trong giao dịch.

Cuối cùng, các doanh nghiệp kinh doanh thực phẩm sạch sẽ chú trọng hơn đến việc xây dựng thương hiệu, nâng cao chất lượng dịch vụ và chăm sóc khách hàng thông qua các kênh trực tuyến.

**1.2.5. Định hướng nghiên cứu của đề tài**

Xuất phát từ thực trạng và xu hướng phát triển của ngành thực phẩm sạch, đề tài **“Xây dựng website giới thiệu và kinh doanh thực phẩm sạch HTFood sử dụng Node.js, React.js, Mysql”** được thực hiện nhằm nghiên cứu và xây dựng một hệ thống website hỗ trợ hoạt động kinh doanh thực phẩm sạch một cách hiệu quả.

Đề tài tập trung vào các nội dung chính sau:

* Nghiên cứu tổng quan về lĩnh vực kinh doanh thực phẩm sạch và các mô hình website thương mại điện tử hiện nay
* Phân tích yêu cầu và thiết kế hệ thống website phù hợp với nhu cầu thực tế
* Xây dựng website giới thiệu sản phẩm, hỗ trợ đặt hàng và quản lý đơn hàng
* **Ứng** dụng các công nghệ hiện đại nhằm nâng cao hiệu suất và trải nghiệm người dùng

Kết quả của đề tài hướng đến việc tạo ra một website có tính ứng dụng cao, giúp doanh nghiệp kinh doanh thực phẩm sạch tối ưu hóa hoạt động quản lý, nâng cao chất lượng dịch vụ và đáp ứng tốt hơn nhu cầu của người tiêu dùng trong thời đại số.

# CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## ****2.1. Phân tích người dùng hệ thống****

### ****2.1.1. Mô tả hệ thống****

Hệ thống website giới thiệu và kinh doanh thực phẩm sạch là một hệ thống ứng dụng web nhằm hỗ trợ doanh nghiệp trong việc quảng bá sản phẩm, bán hàng trực tuyến và quản lý hoạt động kinh doanh. Hệ thống cho phép người dùng truy cập website để xem thông tin sản phẩm, tìm kiếm thực phẩm sạch theo danh mục, đặt hàng trực tuyến và theo dõi đơn hàng.

Bên cạnh đó, hệ thống còn cung cấp giao diện quản trị dành cho người quản lý (Admin) nhằm thực hiện các chức năng quản lý sản phẩm, danh mục, khách hàng, đơn hàng và theo dõi tình hình kinh doanh. Hệ thống được xây dựng theo mô hình client–server, trong đó frontend đảm nhiệm giao diện người dùng và backend xử lý logic nghiệp vụ, kết nối cơ sở dữ liệu.

Dựa trên yêu cầu nghiệp vụ và phạm vi nghiên cứu của đề tài, hệ thống bao gồm **hai nhóm đối tượng người dùng chính**, cụ thể như sau:

* Customer (Khách hàng) gồm: khách vãng lai(Guest), khách hàng đã đăng ký(Member)
* Admin (Quản trị viên) và Staff (nhân viên)

Việc phân chia rõ ràng các nhóm người dùng giúp hệ thống đảm bảo tính bảo mật, phân quyền hợp lý và đáp ứng đúng nhu cầu sử dụng thực tế.

### ****2.1.2. Các đối tượng tham gia hệ thống****

*a) Khách hàng(Customer)*

Khách hàng là nhóm người dùng trực tiếp sử dụng website để tìm hiểu thông tin và mua sắm các sản phẩm thực phẩm sạch. Đây là đối tượng trung tâm của hệ thống, quyết định đến hiệu quả hoạt động và sự phát triển lâu dài của website.

Khách hàng có thể truy cập website thông qua các thiết bị như máy tính, điện thoại thông minh hoặc máy tính bảng có kết nối Internet. Ngay cả khi chưa đăng nhập, khách hàng vẫn có thể xem các thông tin cơ bản như giới thiệu website, danh sách sản phẩm, danh mục thực phẩm, hình ảnh minh họa, giá bán và các thông tin liên quan đến nguồn gốc, xuất xứ của sản phẩm.

* Khách vãng lai (Guest)

Khách vãng lai là những người truy cập vào website giới thiệu và kinh doanh thực phẩm sạch nhưng **chưa đăng ký hoặc chưa đăng nhập tài khoản** trên hệ thống. Đây là nhóm người dùng chiếm số lượng lớn trong giai đoạn đầu tiếp cận website.

Khách vãng lai có thể thực hiện các chức năng cơ bản như:

* Xem trang chủ và các danh mục sản phẩm
* **Xem và** tìm **kiếm sản phẩm, xem chi tiết sản phẩm**
* Xem các tin tức, bài viết giới thiệu và thông tin liên hệ
* **Đăng ký tài** khoản

Tuy nhiên, do chưa có tài khoản, khách vãng lai **bị hạn chế một số chức năng**, cụ thể là:

* Không thể đặt hàng và thanh toán
* Không thể theo dõi đơn hàng
* Không thể đánh giá hoặc phản hồi sản phẩm

Để sử dụng đầy đủ các chức năng của hệ thống, khách vãng lai cần thực hiện **đăng ký tài khoản và đăng nhập**. Việc chuyển đổi từ khách vãng lai sang khách hàng đã đăng ký giúp hệ thống quản lý thông tin người dùng hiệu quả hơn, đồng thời nâng cao trải nghiệm mua sắm trực tuyến cho khách hàng.

* Khách hàng đã đăng ký(Member)

Ngoài các chức năng của khách vãng lai các chức năng chính mà khách hàng có thể thực hiện trong hệ thống bao gồm:

* **Đăng nhập hệ** thống**:** sau khi đăng ký thành công, khách hàng sử dụng thông tin tài khoản để đăng nhập vào hệ thống. Việc đăng nhập giúp khách hàng sử dụng đầy đủ các chức năng nâng cao như đặt hàng, quản lý giỏ hàng, theo dõi đơn hàng và cập nhật thông tin cá nhân.
* **Quản lý giỏ** hàng**: k**hách hàng có thể thêm sản phẩm vào giỏ hàng, thay đổi số lượng hoặc xóa sản **phẩm** trước khi tiến hành đặt hàng. Giỏ hàng giúp khách hàng kiểm soát danh sách sản phẩm cần mua và tổng giá trị đơn hàng.
* **Đặt hàng và** thanh **toán:** sau khi hoàn tất việc lựa chọn sản phẩm, khách hàng tiến hành đặt **hàng** bằng cách xác nhận thông tin giao hàng và lựa chọn phương thức thanh toán phù hợp. Hệ thống ghi nhận đơn hàng và chuyển thông tin đến bộ phận quản lý để xử lý.
* **Theo dõi đơn** hàng**:** khách hàng có thể theo dõi trạng thái đơn hàng như: chờ xác nhận, đang xử lý, đang giao hàng hoặc đã hoàn thành. Điều này giúp nâng cao sự minh bạch và trải nghiệm mua sắm của người dùng.
* **Quản lý thông** tin **cá nhân:** khách hàng có thể cập nhật thông tin cá nhân, thay đổi mật **khẩu** hoặc xem lại lịch sử mua hàng đã thực hiện trên hệ thống.

*b) Quản trị viên(Admin):*

* Admin (quản trị viên)

Quản trị viên là nhóm người dùng có quyền cao nhất trong hệ thống, chịu trách nhiệm quản lý và vận hành toàn bộ website. Admin đóng vai trò quan trọng trong việc đảm bảo hệ thống hoạt động ổn định, dữ liệu được quản lý chính xác và các quy trình kinh doanh được thực hiện hiệu quả.

Quản trị viên truy cập hệ thống thông qua trang quản trị riêng, yêu cầu đăng nhập bằng tài khoản có quyền quản trị. Sau khi đăng nhập thành công, quản trị viên có thể sử dụng các chức năng quản lý sau:

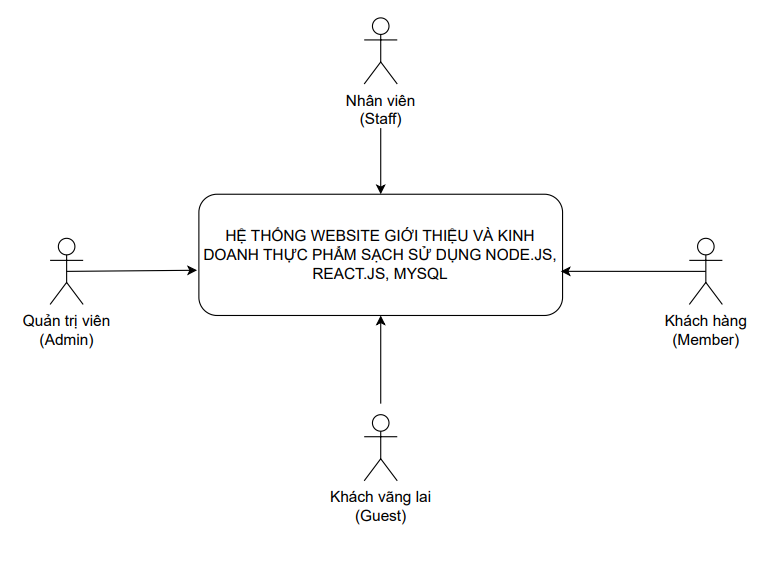
* **Quản lý sản phẩm:** admin có quyền thêm mới, chỉnh sửa, cập nhật hoặc xóa các sản phẩm thực phẩm sạch trên hệ thống. Mỗi sản phẩm được quản lý với đầy đủ thông tin như tên sản phẩm, giá bán, mô tả, hình ảnh, danh mục và trạng thái hiển thị.
* **Quản lý danh mục** sản **phẩm:** quản trị viên có thể tạo và quản lý các danh mục sản phẩm nhằm phân loại thực phẩm một cách khoa học, giúp khách hàng dễ dàng tìm kiếm và lựa chọn sản phẩm.
* **Quản lý khách** hàng**:** admin có thể xem danh sách khách hàng đã đăng ký, kiểm tra thông tin cá nhân và hỗ trợ xử lý các vấn đề phát sinh liên quan đến tài khoản hoặc đơn hàng của khách hàng.
* **Quản lý đơn** hàng**:** quản trị viên theo dõi toàn bộ đơn hàng phát sinh trên hệ thống, cập nhật trạng thái đơn hàng, xử lý yêu cầu hủy đơn hoặc thay đổi thông tin khi cần thiết.
* **Quản lý nội** dung **website:** admin có thể quản lý các nội dung như tin tức, bài viết giới **thiệu**, banner quảng cáo nhằm cung cấp thông tin và thu hút người dùng truy cập website.
* **Thống kê và báo cáo:** hệ thống cung cấp các chức năng thống kê doanh thu, số lượng đơn hàng, sản phẩm bán chạy theo thời gian. Các báo cáo này giúp quản trị viên đánh giá hiệu quả kinh doanh và đưa ra định hướng phát triển phù hợp.
* **Quản lý hệ thống:** bao gồm việc cấu hình hệ thống, phân quyền người dùng và đảm bảo an toàn, bảo mật dữ liệu.
* Nhân viên (Staff)
* **Vai trò:** Stafff là nhóm người dùng có quyền quản trị hạn chế, chỉ được tham gia quản lý sau khi **được** Admin phê duyệt.
* **Quy trình** truy **cập:**
  + Staff phải đăng ký tài khoản.
  + Admin xem xét và duyệt tài khoản.
  + Sau khi được duyệt, Staff có thể đăng nhập vào trang quản trị riêng.
* **Quyền hạn của Staff:**
  + Quản lý sản phẩm: được phép thêm mới hoặc chỉnh sửa thông tin sản phẩm (tên, giá, mô tả, hình ảnh, danh mục).
  + Quản lý danh mục sản phẩm: có thể thêm mới hoặc chỉnh sửa danh mục.
  + Quản lý nội dung website: được phép thêm hoặc chỉnh sửa tin tức, bài viết, banner quảng cáo.
  + Quản lý đơn hàng: có thể cập nhật trạng thái đơn hàng (ví dụ: đã xử lý, đang giao).
  + Thống kê và báo cáo: có quyền xem báo cáo thống kê để hỗ trợ công việc.
  + Giới hạn quyền hạn:
  + Không có quyền xóa sản phẩm, danh mục, khách hàng, đơn hàng.
  + Không có quyền phân quyền người dùng hoặc thay đổi cấu hình hệ thống.
  + Không thể truy cập các chức năng bảo mật và quản lý hệ thống cao cấp.

## ****2.2. Phân tích tổng thể hệ thống****

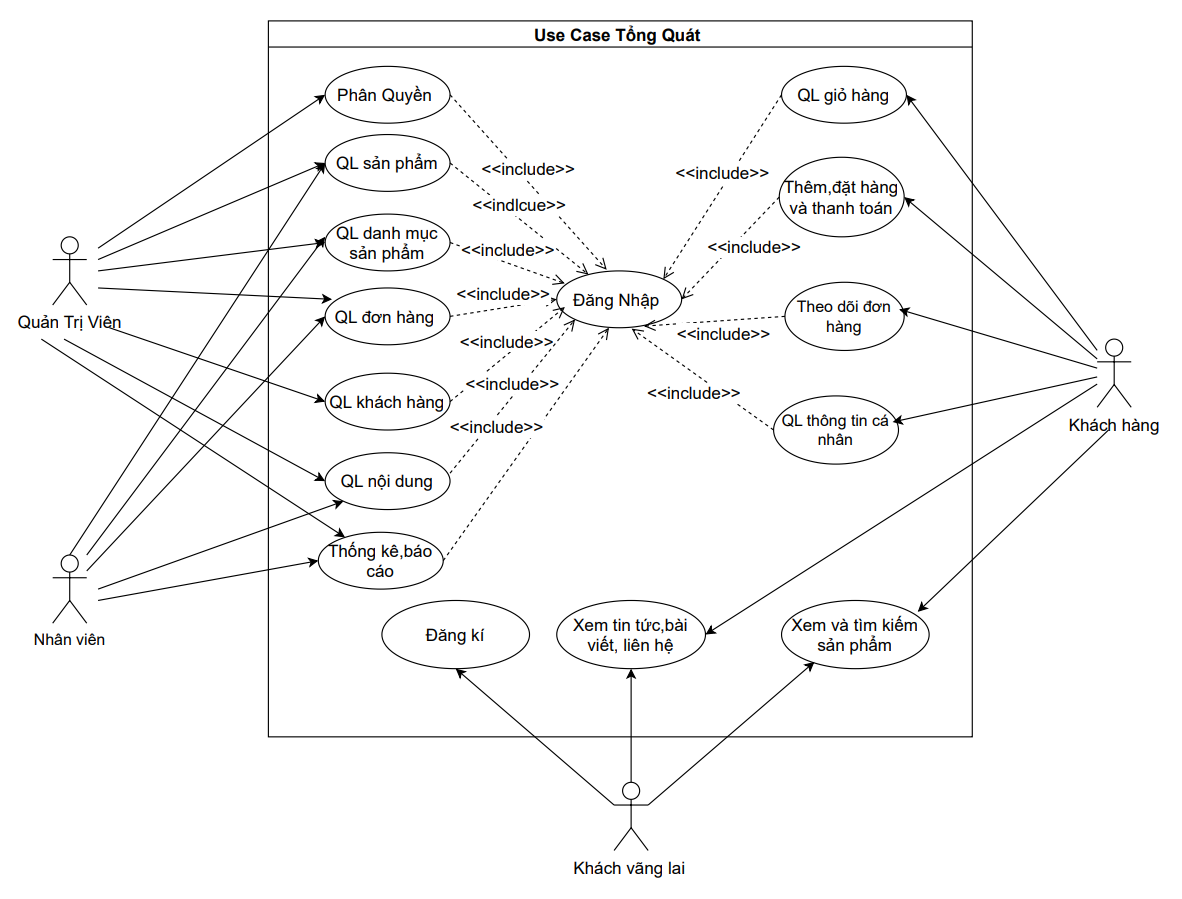
Dựa trên việc phân tích các nhóm người dùng và chức năng tương ứng, hệ thống được thiết kế theo mô hình web ứng dụng với sự phân tách rõ ràng giữa giao diện người dùng và khu vực quản trị. Thiết kế tổng thể hướng đến tính dễ sử dụng, khả năng mở rộng và đảm bảo an toàn thông tin.

### ****2.2.1.**** Biểu đồ Use Case tổng quát

Biểu đồ Use Case tổng quát được sử dụng để mô tả mối quan hệ giữa các tác nhân (Customer, Admin) và các chức năng chính của hệ thống. Thông qua biểu đồ này, có thể xác định phạm vi chức năng mà mỗi đối tượng người dùng có thể thực hiện.



Sơ đồ ngữ cảnh hệ thống



Biểu đồ use case tổng quát

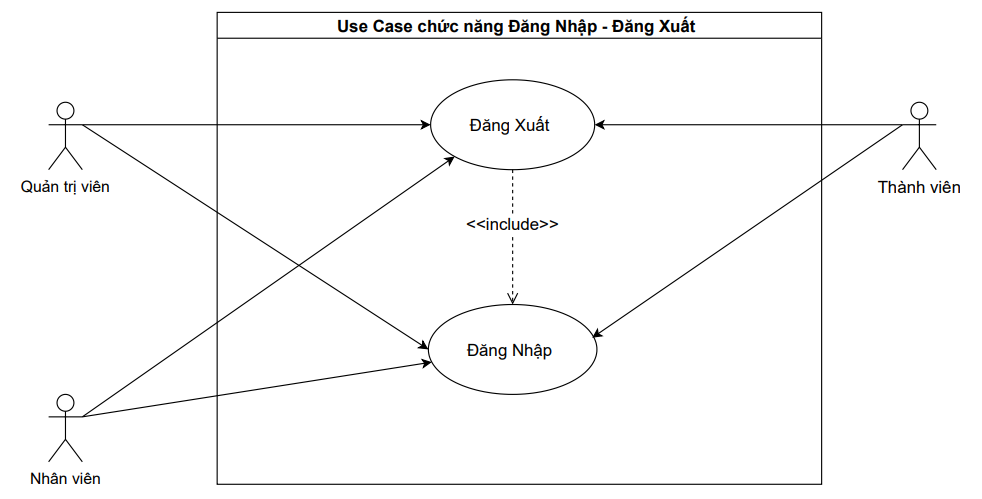
* Dưới đây là các chức năng chính của admin và staff

1. Bảng các chức năng chính của Admin và Staff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả chi tiết** | **Quyền Admin** | **Quyền Staff** |
| 1 | Đăng nhập hệ thống | Truy cập trang quản trị | ✔ | ✔ |
| 2 | Quản lý sản phẩm | Thêm, sửa, xóa sản phẩm | ✔ | Thêm, sửa |
| 3 | Quản lý đơn hàng | Xem, xác nhận, cập nhật trạng thái đơn | ✔ | ✔ |
| 4 | Quản lý khách hàng | Xem, khóa/mở tài khoản khách | ✔ | Xem |
| 5 | Quản lý blog / tin tức | Thêm, sửa, xóa bài viết | ✔ | Thêm, sửa (Admin duyệt) |
| 6 | Quản lý liên hệ | Xem, trả lời liên hệ | ✔ | ✔ |
| 7 | Quản lý khuyến mãi | Thêm, sửa, xóa chương trình khuyến mãi | ✔ | Xem |
| 8 | Thống kê – báo cáo | Doanh thu, đơn hàng, sản phẩm bán chạy | ✔ | Xem |
| 9 | Phân quyền người dùng | Nâng quyền, hạ quyền, tạo tài khoản nhân viên | ✔ | ✖ |
| 10 | Quản lý nhà cung cấp | Thêm, sửa, xóa nhà cung cấp | ✔ | Thêm, sửa |
| 11 | Quản lý tài khoản cá nhân | Cập nhật thông tin, đổi mật khẩu | ✔ | ✔ |
| 1 | Đăng xuất | Thoát khỏi hệ thống | ✔ | ✔ |

* Use Case: Đăng nhập – đăng xuất hệ thống

Người sử dụng hệ thống cần đăng nhập vào hệ thống để truy cập trang quản trị hoặc trang chủ



Biểu đồ Use Case Đăng nhập – đăng xuất

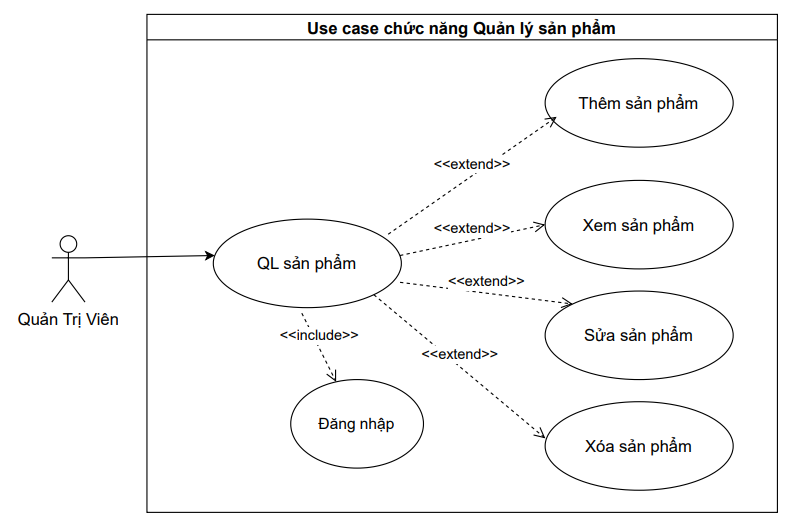
1. Mô tả use case chức năng Đăng nhập – đăng xuất hệ thống

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên use case | Đăng nhập – đăng xuất |
| Tác nhân | Quản trị viên, nhân viên, khách hàng |
| Điều kiện đầu vào | Người sử dụng hệ thống đã có tài khoản hợp lệ |
| Luồng sự kiện chính | 1. Người dùng truy cập trang đăng nhập  2. Nhập tên đăng nhập và mật khẩu  3. Hệ thống kiểm tra thông tin  4. Nếu hợp lệ, hệ thống cho phép truy cập vào hệ thống theo đúng vai trò (admin/member/customer)  5. Người dùng chọn chức năng **Đăng xuất**  6. Hệ thống kết thúc phiên làm việc và quay về trang chủ |
| Luồng mở rộng | – Thông tin đăng nhập sai → Hiển thị thông báo lỗi  – Quên mật khẩu → Hệ thống gửi liên kết đặt lại mật khẩu qua email |
| Kết quả | Đăng nhập thành công và sử dụng hệ thống hoặc đăng xuất thành công hoặc thông báo lỗi |

### ****2.2.2.**** Biểu đồ Use Case của Admin (Quản trị viên)

**a) Use Case 1: Quản lý sản phẩm**

Chức năng này cho phép quản trị viên quản lý toàn bộ thông tin sản phẩm thực phẩm sạch trên hệ thống.



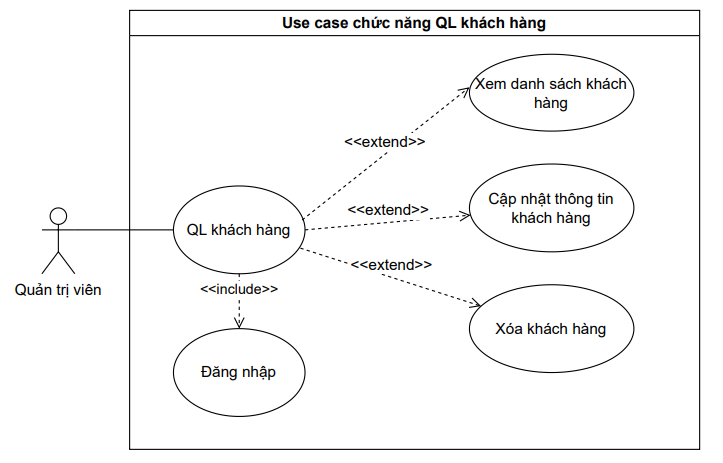
Biểu đồ Use Case Quản lý sản phẩm

1. Mô tả chức năng Quản lý sản phẩm

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên use case | Quản lý sản phẩm |
| Tác nhân | Quản trị viên |
| Điều kiện đầu vào | Đã đăng nhập hệ thống |
| Luồng sự kiện chính | Xem danh sách sản phẩm → Thêm mới → Chỉnh sửa → Xóa sản phẩm |
| Luồng mở rộng | Tìm kiếm sản phẩm theo tên hoặc danh mục |
| Kết quả | Thông tin sản phẩm được cập nhật |

**b) Use Case 2: Quản lý khách hàng**

Quản trị viên có thể theo dõi và quản lý thông tin của các khách hàng đã đăng ký tài khoản.



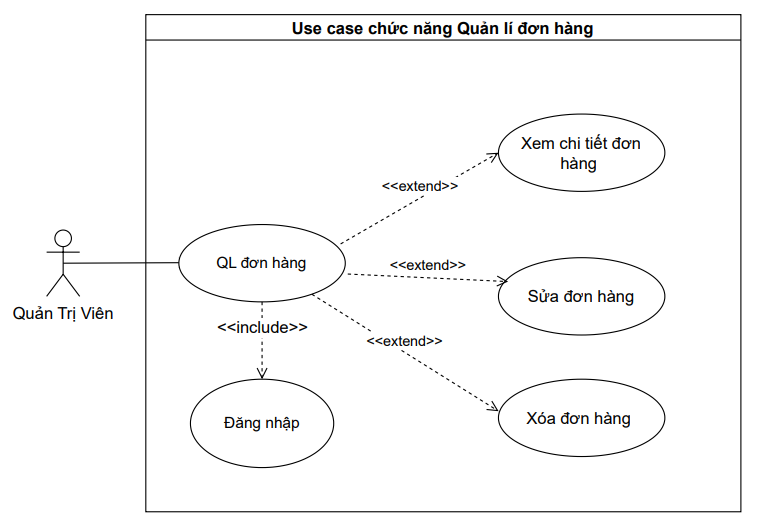
biểu đồ Use Case Quản lý khách hàng

1. Mô tả chức năng Quản lý khách hàng

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên use case | Quản lý khách hàng |
| Tác nhân | Quản trị viên |
| Điều kiện đầu vào | Đã đăng nhập |
| Luồng sự kiện chính | Xem danh sách → Tìm kiếm → Cập nhật → Xóa |
| Kết quả | Dữ liệu khách hàng được cập nhật |

**c) Use Case 3: Quản lý đơn hàng**

Chức năng này hỗ trợ quản trị viên theo dõi và xử lý các đơn hàng phát sinh trên hệ thống.



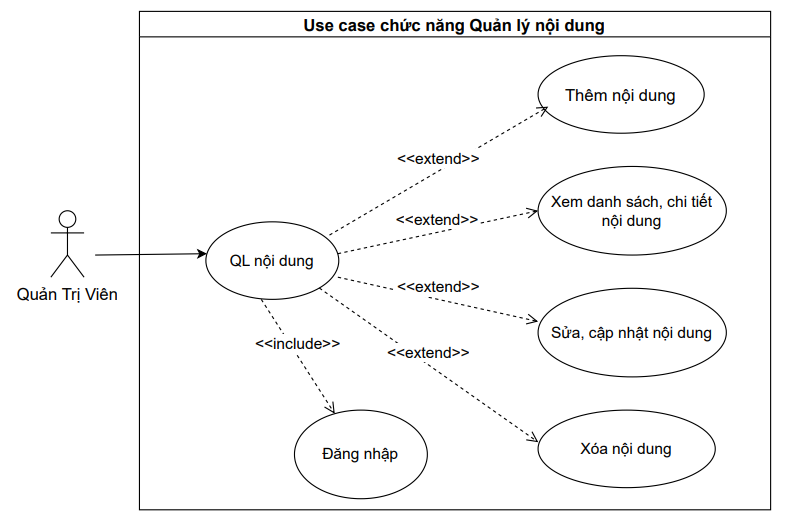
Biểu đồ Use Case Quản lý đơn hàng

1. Mô tả chức năng Quản lý đơn hàng

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên use case | Quản lý đơn hàng |
| Tác nhân | Quản trị viên |
| Điều kiện đầu vào | Đã đăng nhập |
| Luồng sự kiện chính | Xem đơn hàng → Cập nhật trạng thái → Hủy đơn |
| Luồng mở rộng | Lọc đơn hàng theo ngày, trạng thái |
| Kết quả | Trạng thái đơn hàng được cập nhật |

**d) Use Case 4: Quản lý nội dung website**

Quản trị viên quản lý các nội dung hiển thị trên website như tin tức, bài viết, banner quảng cáo.



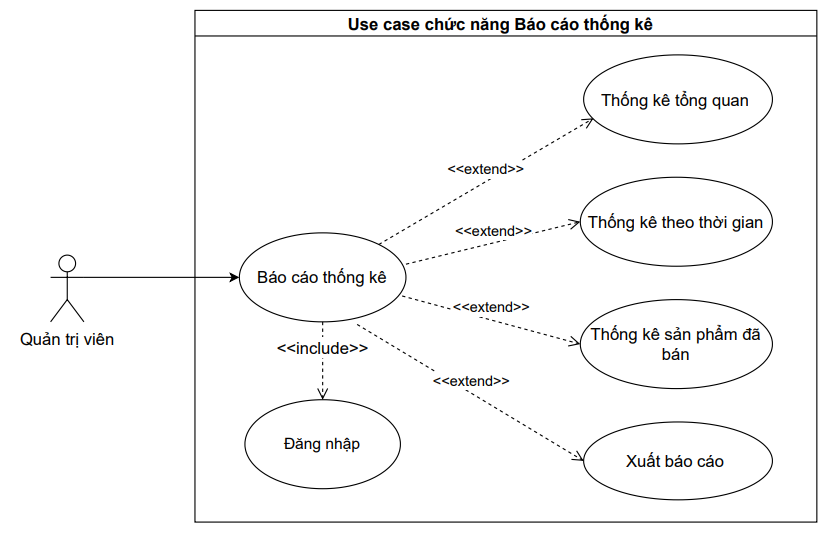
Biểu đồ Use Case Quản lý nội dung

1. Bảng mô tả use case Quản lý nội dung

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên use case | Quản lý nội dung |
| Tác nhân | Quản trị viên |
| Điều kiện đầu vào | Đã đăng nhập |
| Luồng sự kiện chính | Thêm nội dung → Sửa → Xóa |
| Kết quả | Nội dung được cập nhật |

**e) Use Case 5: Thống kê và báo cáo**

Chức năng này hỗ trợ quản trị viên theo dõi hiệu quả kinh doanh thông qua các báo cáo tổng hợp.



Biểu đồ Use Case Thống kê – báo cáo

1. Bảng mô tả use case Thống kê – báo cáo

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên use case | Thống kê – báo cáo |
| Tác nhân | Quản trị viên |
| Điều kiện đầu vào | Dữ liệu đã tồn tại |
| Luồng sự kiện chính | Chọn loại báo cáo → Xem kết quả |
| Kết quả | Báo cáo được hiển thị |

### 2.2.3. Biểu đồ Use Case của Khách hàng

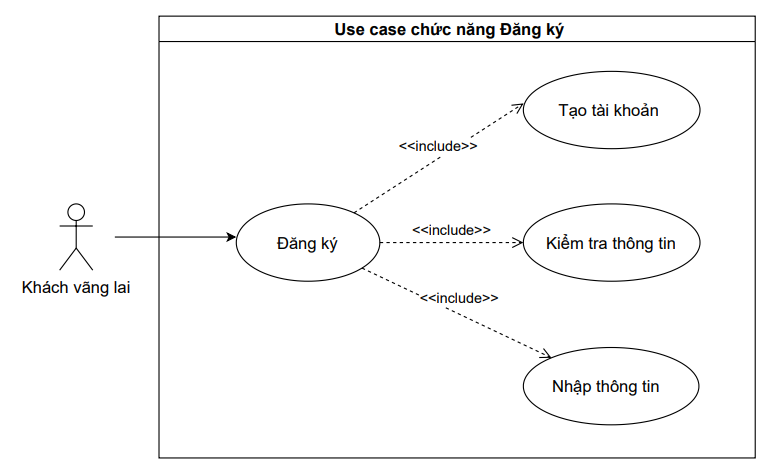
Biểu đồ Use Case của Khách hàng mô tả các chức năng mà người dùng có thể thực hiện khi truy cập vào Website giới thiệu và kinh doanh thực phẩm sạch. Khách hàng là đối tượng trực tiếp sử dụng dịch vụ của hệ thống, thực hiện các thao tác như xem thông tin sản phẩm, đăng ký tài khoản, đặt mua thực phẩm sạch và theo dõi đơn hàng.

1. Bảng các chức năng chính của khách hàng

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả chi tiết** |
| 1 | Xem danh mục sản phẩm | Hiển thị tất cả sản phẩm, phân loại theo danh mục (rau củ, hoa quả, đặc sản…) |
| 2 | Tìm kiếm sản phẩm | Tìm theo tên, giá, danh mục |
| 3 | Xem chi tiết sản phẩm | Ảnh, mô tả, giá, số lượng tồn, thêm vào giỏ hàng |
| 4 | Thêm sản phẩm vào giỏ hàng | Chọn số lượng, thêm nhiều sản phẩm |
| 5 | Xem giỏ hàng | Hiển thị sản phẩm đã chọn, tổng tiền, chỉnh sửa số lượng, xóa sản phẩm |
| 6 | Đặt hàng (Checkout) | Nhập thông tin giao hàng, chọn phương thức thanh toán, gửi đơn |
| 7 | Theo dõi đơn hàng | Xem trạng thái đơn hàng (đang xử lý, giao hàng, đã nhận) |
| 8 | Đăng ký / Đăng nhập | Tạo tài khoản, đăng nhập để mua hàng |
| 9 | Đăng xuất | Thoát khỏi hệ thống |
| 10 | Xem blog / tin tức | Hướng dẫn, kiến thức về thực phẩm sạch |
| 11 | Liên hệ | Form liên hệ, số hotline, email |

**a) Use Case 1: Đăng ký tài khoản**

Chức năng đăng ký cho phép người dùng tạo tài khoản cá nhân để sử dụng đầy đủ các tính năng của website.

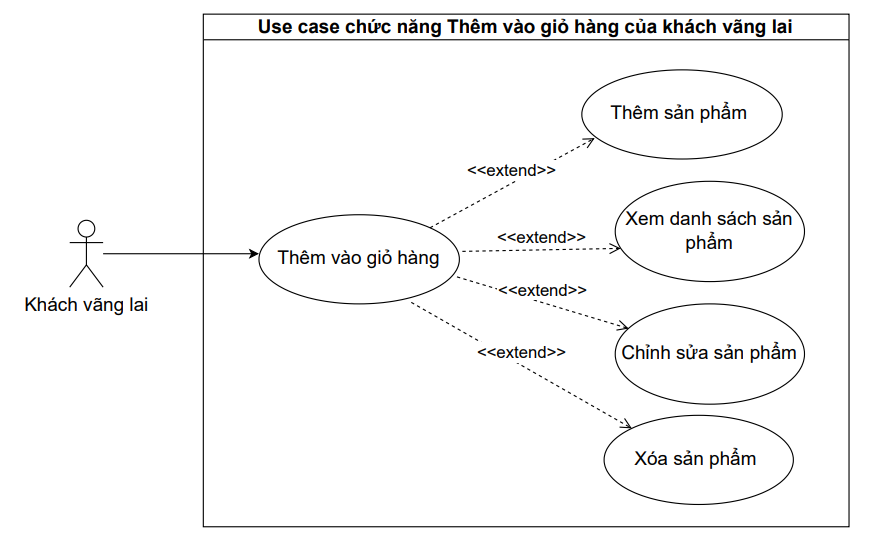


Biểu đồ Use Case Đăng ký tài khoản

1. Bảng mô tả use case Đăng ký tài khoản

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên use case | Đăng ký tài khoản |
| Tác nhân | Khách hàng |
| Điều kiện đầu vào | Người dùng chưa có tài khoản |
| Luồng sự kiện chính | 1. Khách hàng chọn “Đăng ký” 2. Hệ thống hiển thị form đăng ký 3. Khách hàng nhập thông tin cá nhân 4. Hệ thống kiểm tra dữ liệu 5. Lưu tài khoản mới |
| Luồng mở rộng | Email đã tồn tại → thông báo lỗi |
| Kết quả | Tạo tài khoản thành công |

**b) Use Case 2: Thêm vào giỏ hàng**



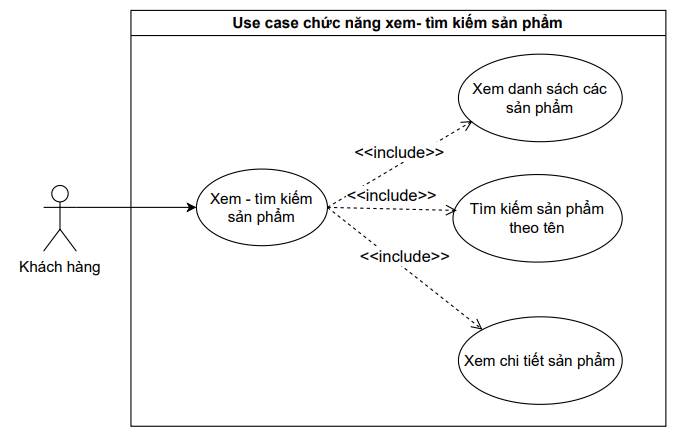
Biểu đồ Use Case Thêm vào giỏ

1. Bảng mô tả use case Thêm vào giỏ hàng

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên use case | Thêm vào giỏ hàng |
| Tác nhân | Thành viên |
| Điều kiện đầu vào | Khách hàng đã đăng nhập hệ thống |
| Luồng sự kiện chính | Khách hàng xem danh sách sản phẩm → Chọn sản phẩm muốn mua → Nhấn nút **Thêm vào giỏ hàng** → Hệ thống kiểm tra số lượng tồn kho →Thêm sản phẩm vào giỏ hàng → Hiển thị giỏ hàng cập nhật |
| Luồng mở rộng | - Sản phẩm hết hàng → Thông báo “Sản phẩm tạm hết”  - Người dùng chưa đăng nhập → Chuyển đến trang đăng nhập |
| Kết quả | Sản phẩm được thêm vào giỏ hàng thành công |

**c) Use Case 3: Xem và tìm kiếm sản phẩm**

Khách hàng có thể duyệt danh sách sản phẩm hoặc tìm kiếm thực phẩm theo nhu cầu.



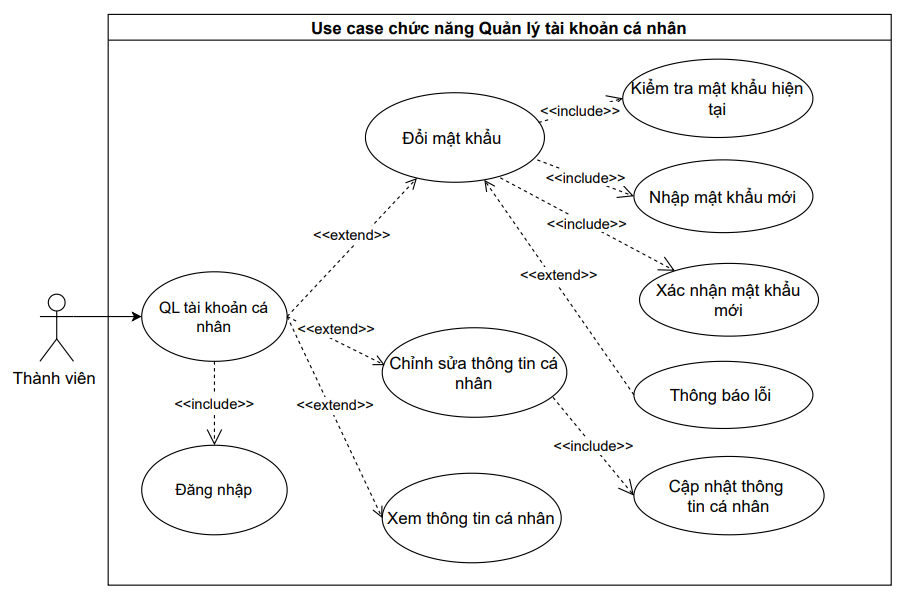
Biểu đồ Use Case Xem – Tìm kiếm sản phẩm

1. Bảng mô tả use case Tìm kiếm sản phẩm

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên use case | Tìm kiếm sản phẩm |
| Tác nhân | Khách hàng |
| Điều kiện đầu vào | Hệ thống có dữ liệu |
| Luồng sự kiện chính | Nhập từ khóa → hệ thống truy vấn → hiển thị kết quả |
| Luồng ngoại lệ | Không tìm thấy kết quả |
| Kết quả | Danh sách sản phẩm phù hợp |

**d) Use Case 4: Quản lý tài khoản cá nhân**

Chức năng cho phép thành viên xem thông tin chi tiết và chỉnh sửa thông tin cá nhân của mình.



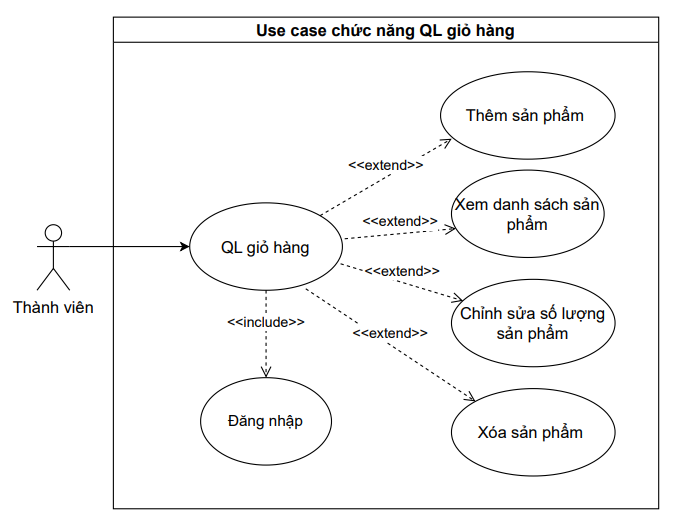
Biểu đồ Use Case Quản lý tài khoản cá nhân

1. Mô tả use case Quản lý tài khoản cá nhân

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên use case | Quản lý tài khoản cá nhân |
| Tác nhân | Thành viên |
| Điều kiện đầu vào | Khách hàng đã đăng nhập hệ thống |
| Luồng sự kiện chính | Khách hàng đăng nhập hệ thống→Truy cập trang **Tài khoản**→Xem thông tin cá nhân→Chỉnh sửa thông tin cá nhân→Lưu thay đổi→Hệ thống cập nhật thông tin |
| Luồng mở rộng | - Đổi mật khẩu: Khách hàng chọn **Đổi mật khẩu** → Nhập mật khẩu cũ → Nhập mật khẩu mới → Xác nhận mật khẩu mới  - Thông tin không hợp lệ → Hệ thống hiển thị thông báo lỗi |
| Kết quả | Thông tin tài khoản được cập nhật thành công hoặc hiển thị lỗi |

**e) Use Case 5: Quản lý giỏ hàng**

Khách hàng có thể thêm, sửa, xóa sản phẩm trong giỏ hàng.



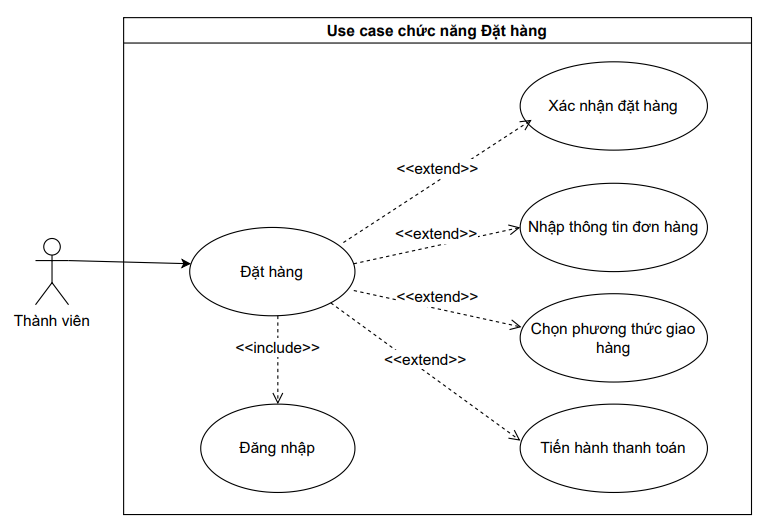
Biểu đồ Use Case Quản lý giỏ hàng

1. mô tả use case Quản lý giỏ hàng

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên use case | Quản lý giỏ hàng |
| Tác nhân | Thành viên |
| Điều kiện đầu vào | Đã đăng nhập |
| Luồng sự kiện chính | Thêm → cập nhật số lượng → xóa |
| Kết quả | Giỏ hàng được cập nhật |

**f) Use Case 6: Đặt hàng**

Cho phép khách hàng tiến hành đặt mua thực phẩm sạch.



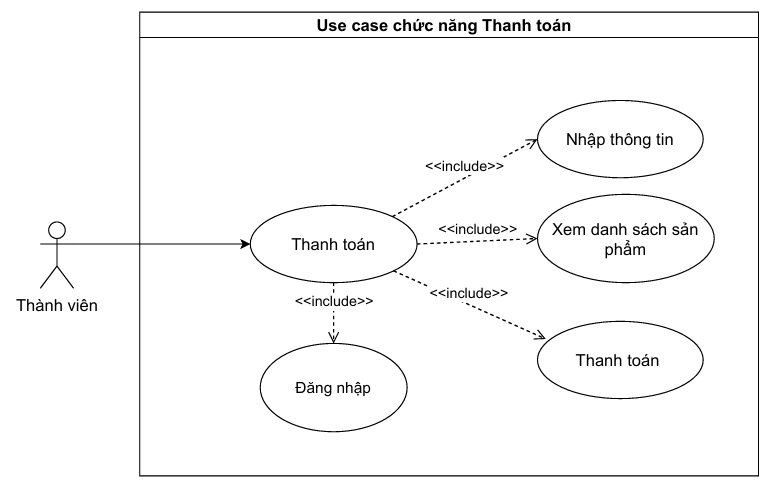
Biểu đồ Use Case Đặt hàng

1. mô tả use case Đặt hàng

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên use case | Đặt hàng |
| Tác nhân | Thành viên |
| Điều kiện đầu vào | Giỏ hàng không rỗng |
| Luồng sự kiện chính | Xem→Nhập địa chỉ → xác nhận đơn |
| Kết quả | Đơn hàng được tạo |

**g) Use Case 7: Thanh toán**

Khách hàng thực hiện thanh toán cho đơn hàng đã đặt.



Biểu đồ Use Case Thanh toán

1. Bảng mô tả use case Thanh toán

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên use case | Thanh toán |
| Tác nhân | Khách hàng |
| Điều kiện đầu vào | Đã tạo đơn hàng |
| Luồng sự kiện chính | Chọn phương thức → xác nhận |
| Luồng mở rộng | Thanh toán thất bại |
| Kết quả | Cập nhật trạng thái đơn |

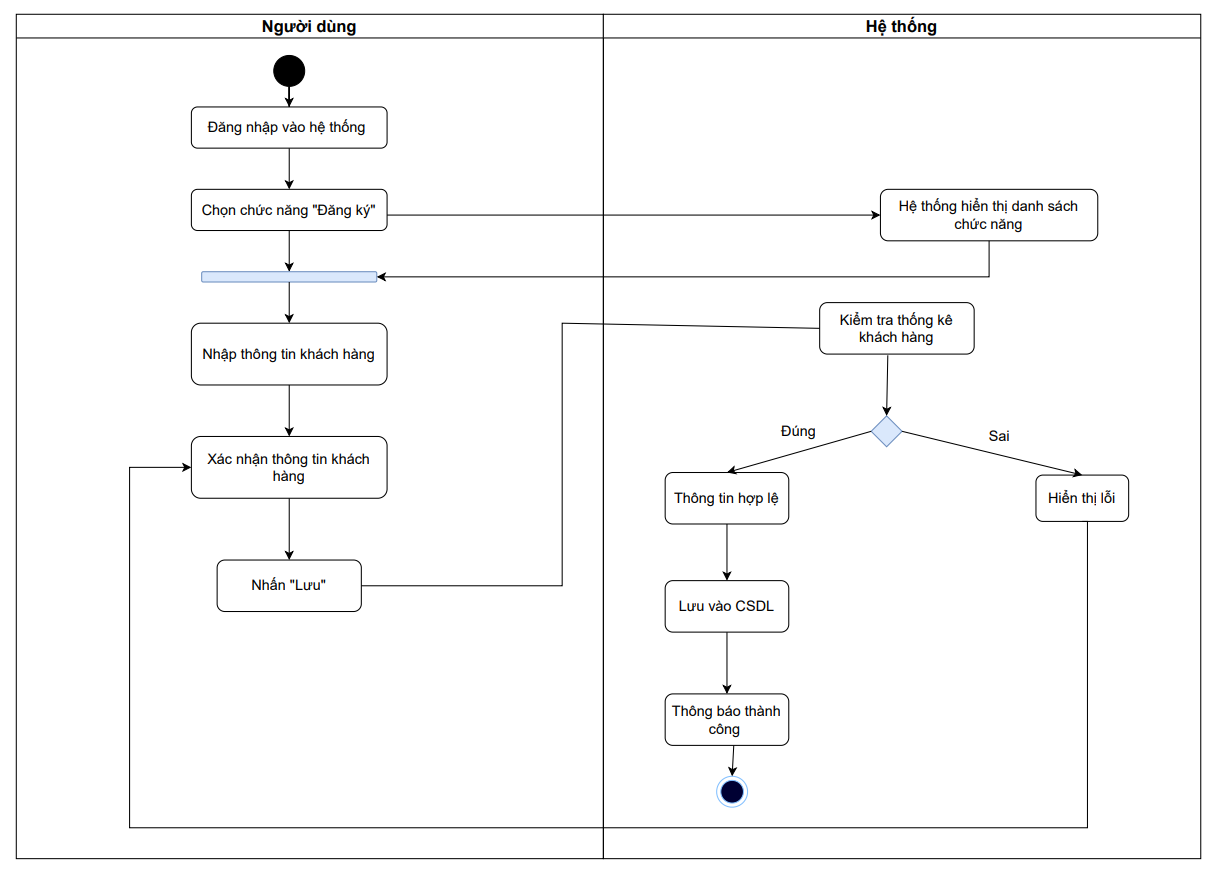
### ****2.2.4. Biểu đồ hoạt động (Activity Diagram)****

Biểu đồ hoạt động được sử dụng để mô tả luồng xử lý nghiệp vụ của hệ thống Website giới thiệu và kinh doanh thực phẩm sạch. Thông qua biểu đồ này, các bước xử lý từ khi người dùng bắt đầu thao tác cho đến khi kết thúc một chức năng cụ thể được thể hiện một cách trực quan và rõ ràng. Việc xây dựng biểu đồ hoạt động giúp làm rõ trình tự xử lý, mối quan hệ giữa các bước và hỗ trợ quá trình thiết kế, triển khai hệ thống một cách chính xác.

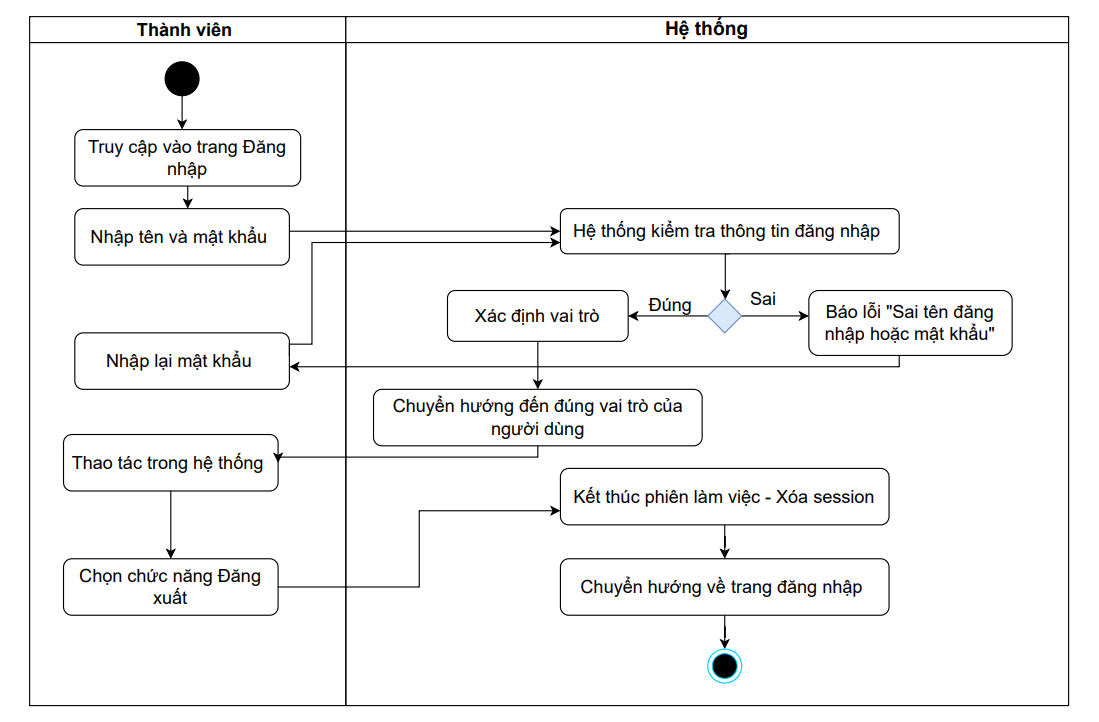
Trong phạm vi đề tài, các biểu đồ hoạt động được xây dựng cho những chức năng quan trọng, ảnh hưởng trực tiếp đến trải nghiệm người dùng và hoạt động kinh doanh của hệ thống.

**a) Biểu đồ hoạt động đăng ký và đăng nhập của khách hàng**

Biểu đồ hoạt động đăng ký và đăng nhập mô tả quá trình khách hàng tạo tài khoản và truy cập vào hệ thống website. Đây là chức năng nền tảng, giúp hệ thống quản lý thông tin người dùng, lịch sử mua hàng và các dịch vụ cá nhân hóa.



Biểu đồ hoạt động đăng ký



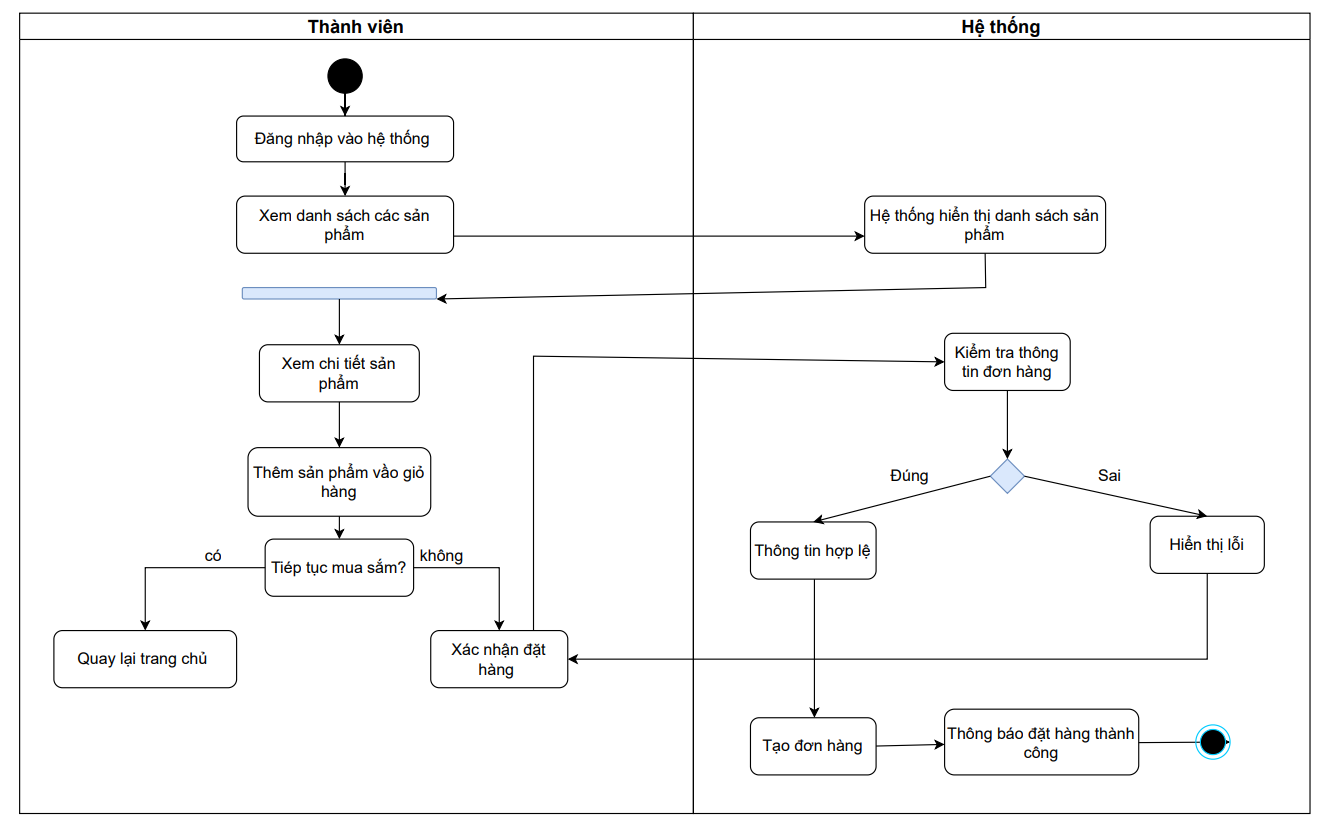
Biểu đồ hoạt động đăng nhập – đăng xuất

Mô tả luồng hoạt động:quá trình bắt đầu khi khách hàng truy cập vào website và lựa chọn chức năng **đăng nhập**. Hệ thống hiển thị form đăng nhập, yêu cầu người dùng nhập thông tin tài khoản gồm tên đăng nhập (hoặc email) và mật khẩu.

* Sau khi người dùng nhập thông tin, hệ thống tiến hành xác thực dữ liệu, kiểm tra sự tồn tại của tài khoản và tính chính xác của mật khẩu. Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại.
* Khi thông tin đăng nhập hợp lệ, hệ thống cho phép người dùng truy cập vào các chức năng của website như: xem sản phẩm, thêm sản phẩm vào giỏ hàng, đặt hàng, theo dõi đơn hàng và quản lý thông tin cá nhân.
* Trong quá trình sử dụng, người dùng có thể lựa chọn chức năng đăng xuất. Khi đó, hệ thống kết thúc phiên làm việc của người dùng, xóa thông tin phiên đăng nhập và chuyển người dùng về trạng thái khách vãng lai trên website.
* Biểu đồ hoạt động đăng nhập – đăng xuất giúp mô tả rõ quy trình xác thực người dùng, đảm bảo tính an toàn và bảo mật thông tin tài khoản, đồng thời nâng cao trải nghiệm người dùng trong quá trình sử dụng website.

**b) Hoạt động mua hàng và đặt đơn**

Biểu đồ hoạt động mua hàng mô tả toàn bộ quy trình từ khi khách hàng lựa chọn sản phẩm cho đến khi hoàn tất việc tạo đơn hàng. Đây là nghiệp vụ cốt lõi của website kinh doanh thực phẩm sạch.



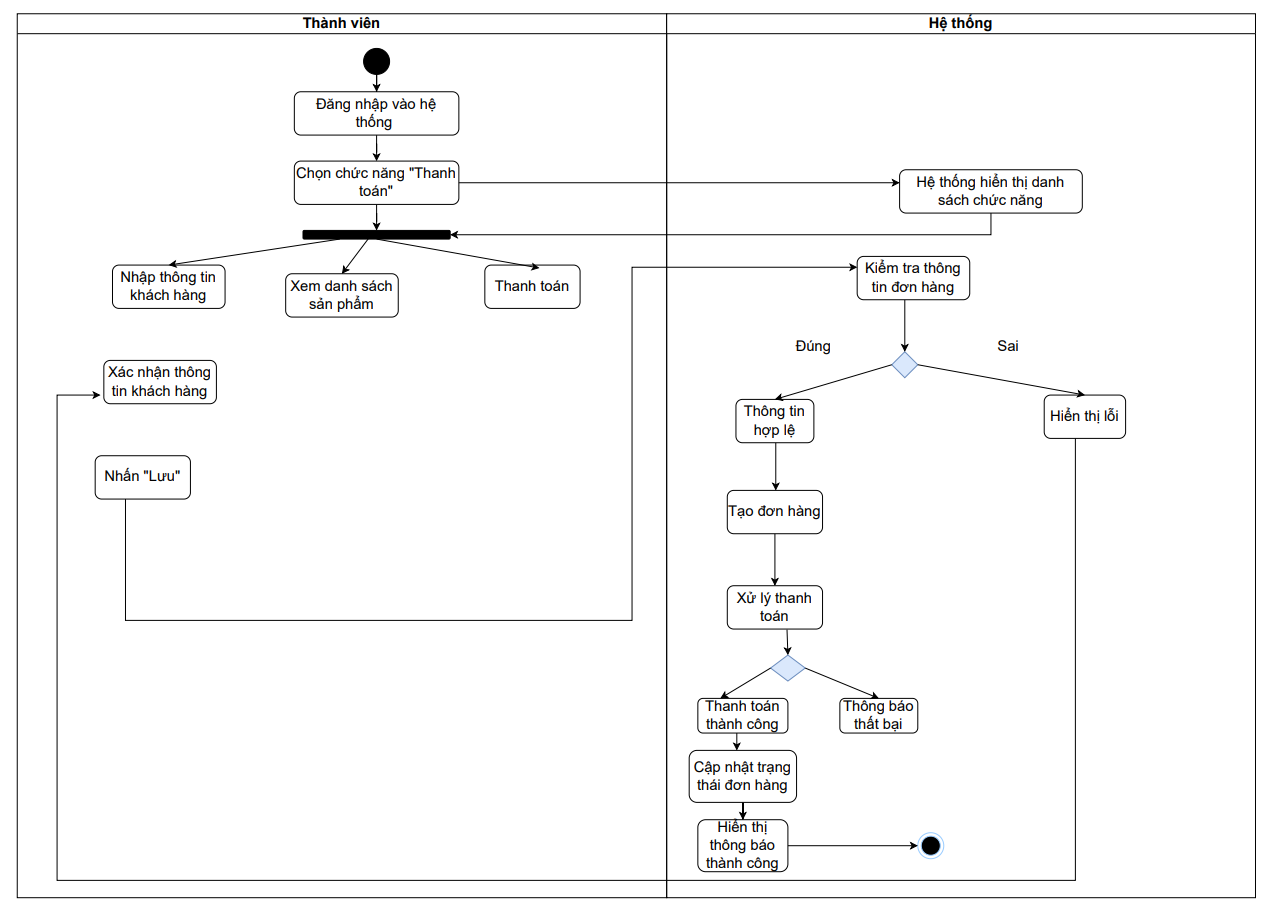
Biểu đồ hoạt động mua hàng – đặt đơn

Mô tả luồng hoạt động:Quá trình bắt đầu khi khách hàng duyệt danh sách sản phẩm hoặc tìm kiếm thực phẩm theo nhu cầu.

* Khách hàng xem chi tiết sản phẩm, bao gồm hình ảnh, mô tả, nguồn gốc xuất xứ, giá bán và tình **trạng** còn hàng.
* Sau khi lựa chọn sản phẩm phù hợp, khách hàng thêm sản phẩm vào giỏ hàng và có thể tiếp tục mua sắm hoặc chuyển sang bước đặt hàng.
* Hệ thống cho phép khách hàng kiểm tra lại giỏ hàng, điều chỉnh số lượng hoặc loại bỏ sản **phẩm** không cần thiết.
* Khách hàng nhập thông tin giao hàng và xác nhận đơn hàng.
* Sau khi xác nhận, hệ thống tiến hành tạo đơn hàng, lưu thông tin vào cơ sở dữ liệu và cập nhật trạng **thái** đơn hàng ban đầu là “Chờ xử lý”.
* Biểu đồ **hoạt** động này giúp mô tả rõ quy trình xử lý đơn hàng, đảm bảo tính chính xác và nhất quán trong hoạt động kinh doanh.

**c) Biểu đồ hoạt động thanh toán và theo dõi đơn hàng**

Biểu đồ hoạt động thanh toán và theo dõi đơn hàng mô tả quá trình khách hàng thực hiện thanh toán và kiểm tra trạng thái đơn hàng sau khi mua.



Biểu đồ hoạt động thanh toán – theo dõi đơn hàng

Sau khi đơn hàng được tạo, khách hàng lựa chọn phương thức thanh toán phù hợp như thanh toán khi nhận hàng hoặc thanh toán trực tuyến.

* Hệ thống tiếp nhận thông tin thanh toán và xử lý giao dịch.
* Trong **trường** hợp thanh toán thành công, hệ thống cập nhật trạng thái đơn hàng sang “Đã thanh toán” và chuyển sang giai đoạn xử lý – giao hàng.
* Nếu thanh **toán** không thành công, hệ thống thông báo lỗi và cho phép khách hàng thực hiện lại giao dịch.

Khách hàng có thể truy cập chức năng theo dõi đơn hàng để xem trạng thái đơn như: chờ xử lý, đang giao hàng hoặc đã hoàn thành. Việc theo dõi đơn hàng giúp nâng cao tính minh bạch và sự tin tưởng của khách hàng đối với hệ thống.

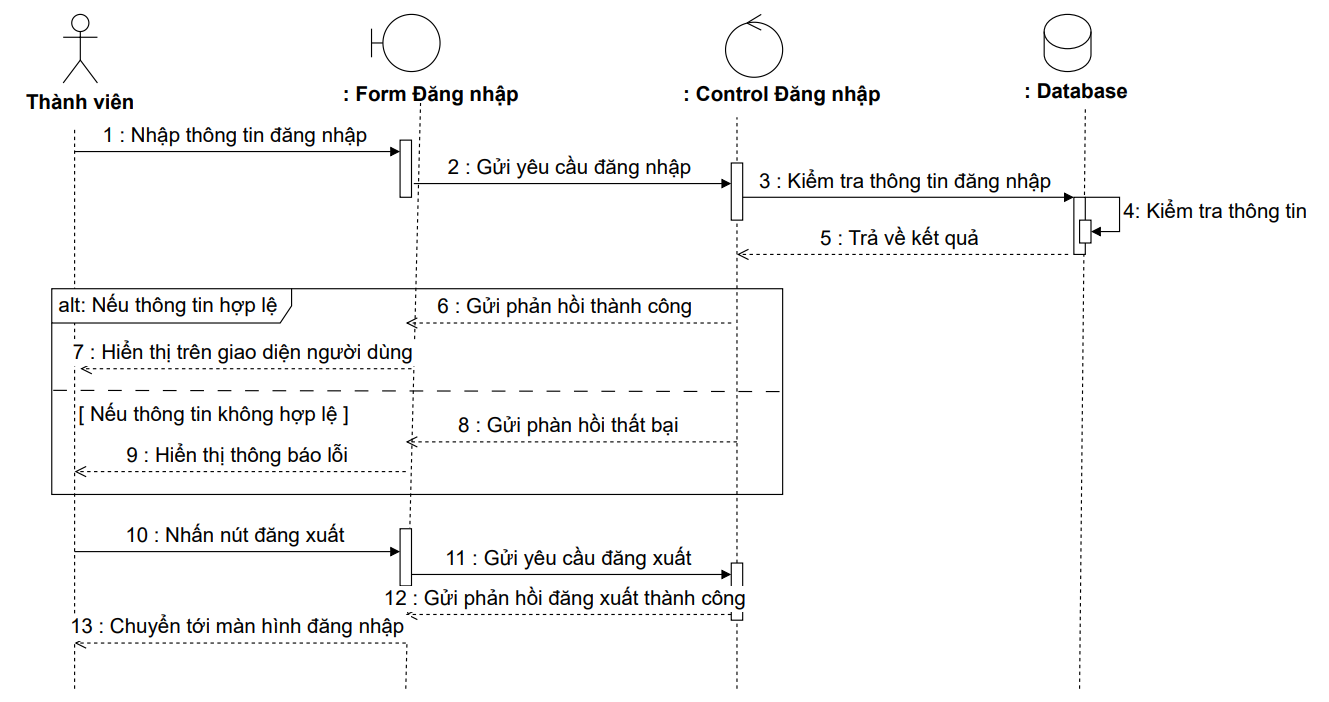
### ****2.2.5. Biểu đồ tuần tự (Sequence Diagram)****

Biểu đồ tuần tự (Sequence Diagram) được sử dụng để mô tả sự tương tác giữa các đối tượng trong hệ thống theo trình tự thời gian. Thông qua biểu đồ này, luồng trao đổi thông tin giữa người dùng, giao diện hệ thống, các thành phần xử lý nghiệp vụ và cơ sở dữ liệu được thể hiện rõ ràng, giúp làm sáng tỏ cách hệ thống vận hành trong từng chức năng cụ thể.

Trong đề tài: “**Xây dựng** website giới thiệu và kinh doanh thực phẩm sạch sử dụng Node.js, React.js, Mysql”, biểu đồ tuần tự được xây dựng cho các chức năng quan trọng như đăng nhập hệ thống, đặt hàng và thanh toán nhằm đảm bảo tính chính xác trong thiết kế và triển khai hệ thống.

**a) Biểu đồ tuần tự chức năng đăng nhập**

Biểu đồ tuần tự chức năng đăng nhập mô tả quá trình khách hàng truy cập hệ thống bằng tài khoản đã đăng ký, bao gồm các bước xác thực thông tin và phản hồi kết quả cho thành viên.



Biểu đồ tuần tự đăng nhập – đăng xuất

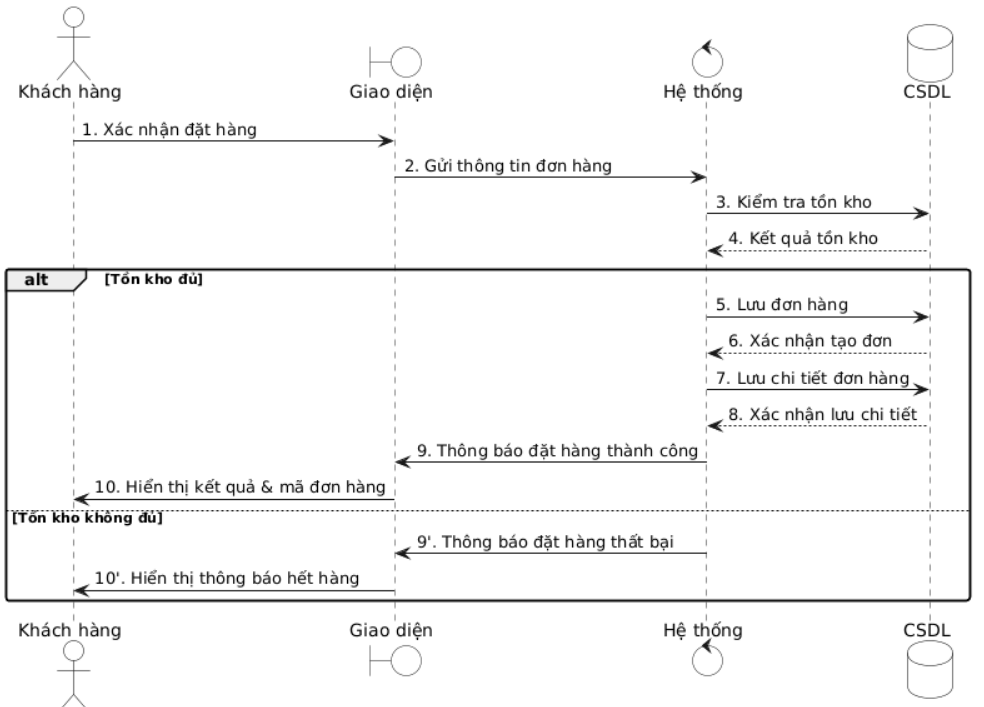
Quá trình bắt đầu khi khách hàng truy cập vào trang **đăng nhập** của website và nhập thông tin tài khoản bao gồm email và mật khẩu.

* Giao diện người dùng gửi yêu cầu đăng nhập đến hệ thống backend. Backend tiếp nhận yêu cầu và tiến hành kiểm tra thông tin đăng nhập với dữ liệu được lưu trữ trong cơ sở dữ liệu MySQL.
* Nếu thông tin hợp lệ, hệ thống tạo phiên đăng nhập (session hoặc token), lưu trạng thái người dùng và phản hồi kết quả đăng nhập thành công cho giao diện. Người dùng sau đó có thể truy cập và sử dụng các chức năng của hệ thống theo đúng quyền hạn được phân quyền.
* Trong trường hợp thông tin đăng nhập không hợp lệ, hệ thống trả về thông báo lỗi và yêu cầu người dùng nhập lại thông tin.
* Trong quá trình sử dụng website, khi người dùng lựa chọn chức năng **đăng xuất**, giao diện gửi yêu cầu đăng xuất đến backend. Hệ thống tiến hành hủy phiên đăng nhập (xóa session hoặc vô hiệu hóa token), kết thúc trạng thái đăng nhập của người dùng và phản hồi kết quả đăng xuất thành công, đưa người dùng trở về trạng thái chưa đăng nhập.

Biểu đồ tuần tự đăng nhập – đăng xuất giúp mô tả rõ quá trình tương tác giữa giao diện người dùng, hệ thống backend và cơ sở dữ liệu, đồng thời đảm bảo tính bảo mật, an toàn và độ chính xác trong việc xác thực và quản lý phiên làm việc của người dùng.

**b) Biểu đồ tuần tự chức năng đặt hàng**

Biểu đồ tuần tự chức năng đặt hàng mô tả chi tiết quá trình khách hàng tạo đơn hàng từ giỏ hàng cho đến khi hệ thống lưu thông tin đơn hàng vào cơ sở dữ liệu.



Biểu đồ tuần tự đặt hàng

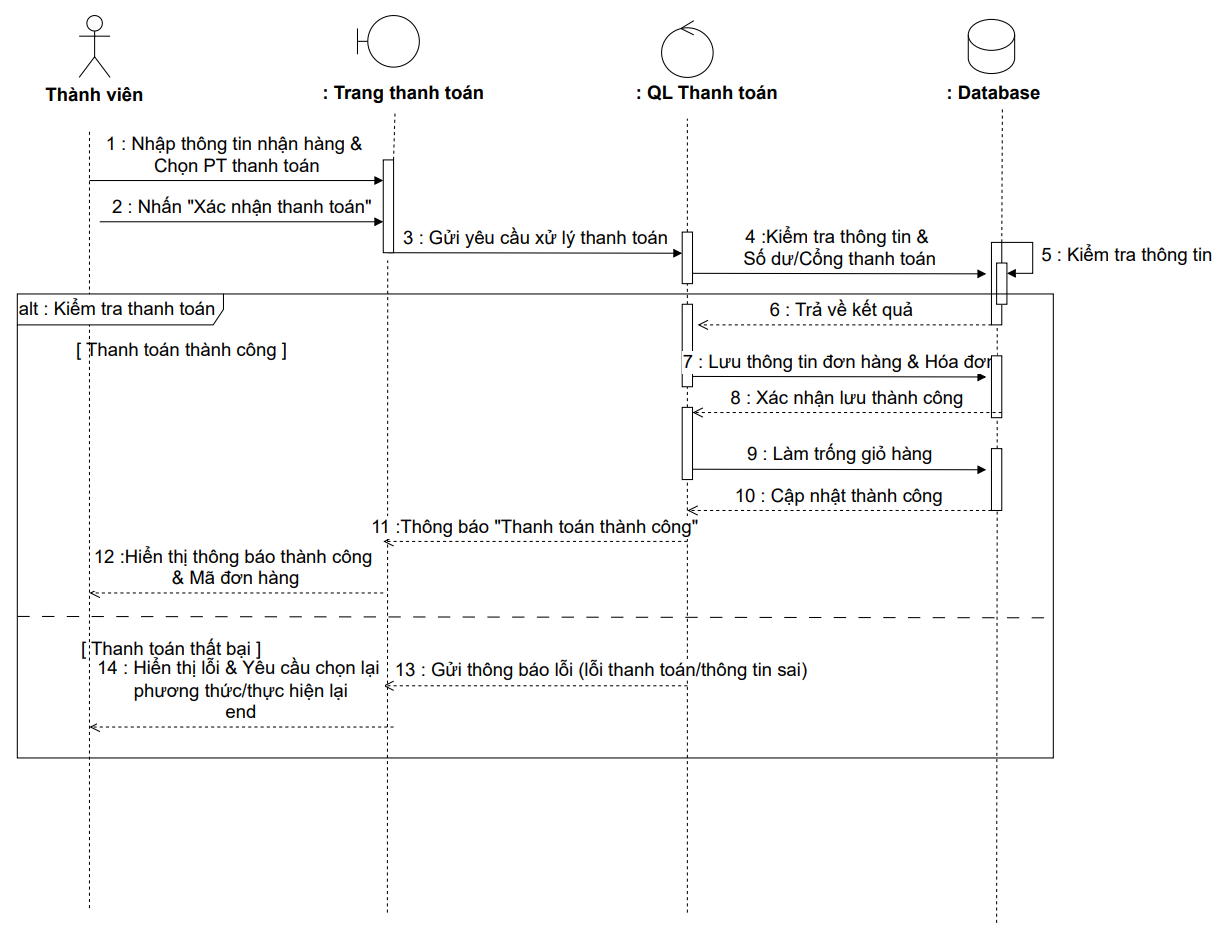
Quá trình bắt đầu khi khách hàng chọn chức năng đặt hàng từ giỏ hàng.

* Giao diện gửi yêu cầu tạo đơn hàng đến hệ thống backend kèm theo thông tin sản phẩm, số lượng và thông tin giao hàng.
* Backend tiếp nhận yêu cầu, tiến hành kiểm tra tình trạng tồn kho của từng sản phẩm.
* Nếu sản phẩm còn hàng, hệ thống tính toán tổng tiền, tạo đơn hàng mới và lưu dữ liệu vào bảng đơn hàng trong cơ sở dữ liệu.
* Sau khi lưu đơn hàng thành công, backend phản hồi thông tin xác nhận đơn hàng cho giao diện người dùng.

Trong trường hợp một hoặc nhiều sản phẩm không còn hàng, hệ thống sẽ thông báo lỗi và yêu cầu khách hàng điều chỉnh giỏ hàng.

**c) Biểu đồ tuần tự chức năng thanh toán**

Biểu đồ tuần tự chức năng thanh toán mô tả quá trình xử lý giao dịch thanh toán và cập nhật trạng thái đơn hàng.



Biểu đồ tuần tự thanh toán

Sau khi đơn hàng được tạo, khách hàng lựa chọn phương thức thanh toán phù hợp.

* Giao diện người dùng gửi yêu cầu thanh toán đến hệ thống backend.
* Backend xử lý thông tin thanh toán, xác nhận giao dịch và cập nhật trạng thái thanh toán của **đơn** hàng trong cơ sở dữ liệu.
* Trong trường hợp thanh toán thành công, hệ thống phản hồi thông báo thành công và cập **nhật** trạng thái đơn hàng sang “Đã thanh toán”.
* Nếu giao dịch thất bại, hệ thống gửi thông báo lỗi và cho phép khách hàng thực hiện thanh toán **lại**.

Biểu đồ tuần tự thanh toán giúp mô tả rõ sự phối hợp giữa các thành phần trong hệ thống, đảm bảo quy trình thanh toán diễn ra an toàn và chính xác.

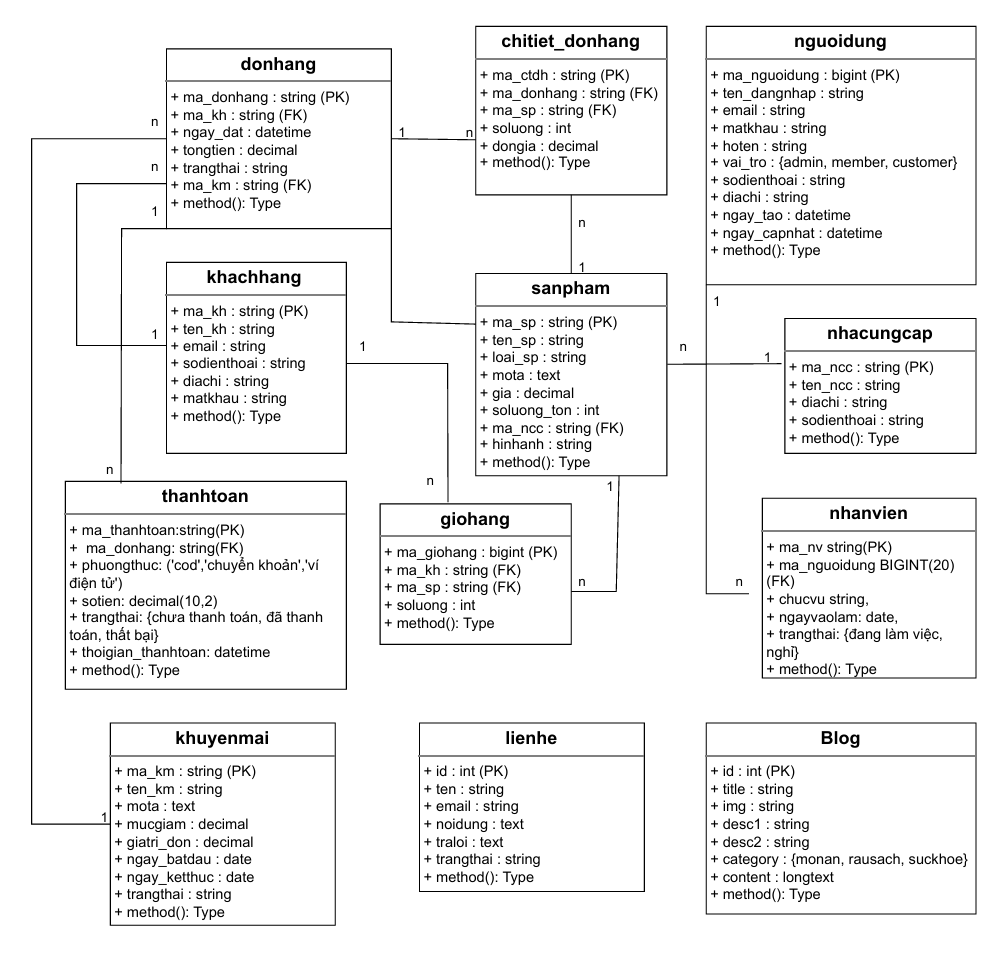
### ****2.2.6. Biểu đồ lớp (Class Diagram)****

Biểu đồ lớp (Class Diagram) là một trong những biểu đồ quan trọng trong UML, được sử dụng để mô tả cấu trúc tĩnh của hệ thống. Biểu đồ này thể hiện các lớp đối tượng chính trong hệ thống, các thuộc tính, phương thức của từng lớp cũng như mối quan hệ giữa các lớp với nhau. Việc xây dựng biểu đồ lớp giúp hệ thống được thiết kế một cách chặt chẽ, logic và thuận lợi cho quá trình lập trình và bảo trì sau này.

Trong Website giới thiệu và kinh doanh thực phẩm sạch, biểu đồ lớp được xây dựng dựa trên các nghiệp vụ cốt lõi như quản lý người dùng, sản phẩm, đơn hàng, giỏ hàng và thanh toán.

Biểu đồ lớp của Website giới thiệu và kinh doanh thực phẩm sạch phản ánh đầy đủ cấu trúc dữ liệu và nghiệp vụ của hệ thống. Mỗi lớp đảm nhận một vai trò riêng biệt nhưng có mối liên hệ mật thiết với các lớp khác. Việc phân chia lớp rõ ràng giúp tăng tính tái sử dụng, dễ bảo trì và nâng cấp hệ thống.

Biểu đồ lớp là cơ sở quan trọng để xây dựng cơ sở dữ liệu MySQL cũng như triển khai các module backend bằng Node.js và giao diện frontend bằng React + Vite. Nhờ có biểu đồ lớp, quá trình phát triển hệ thống được thực hiện một cách khoa học, giảm thiểu lỗi và nâng cao chất lượng sản phẩm.



Biểu đồ lớp hệ thống thực phẩm sạch HTFood

## ****2.3. Thiết kế cơ sở dữ liệu****

### ****2.3.1. Tổng quan cơ sở dữ liệu****

Cơ sở dữ liệu của Website giới thiệu và kinh doanh thực phẩm sạch được xây dựng trên hệ quản trị cơ sở dữ liệu MySQL, sử dụng mô hình cơ sở dữ liệu quan hệ. Việc thiết kế cơ sở dữ liệu nhằm mục đích lưu trữ, quản lý và khai thác hiệu quả các thông tin liên quan đến người dùng, khách hàng, sản phẩm, nhà cung cấp, giỏ hàng, đơn hàng, khuyến mãi và liên hệ khách hàng.

Cơ sở dữ liệu được thiết kế đảm bảo các yêu cầu sau:

* Đảm bảo tính **toàn** vẹn dữ liệu thông qua việc sử dụng khóa chính và khóa ngoại
* Hạn chế dư **thừa** dữ liệu, tránh trùng lặp thông tin
* Hỗ trợ tốt cho các chức năng nghiệp vụ của website
* Dễ dàng mở rộng và bảo trì trong tương lai

Tên cơ sở dữ liệu sử dụng: ltht\_thucphamsach

### ****2.3.2. Danh sách các bảng dữ liệu****

1. Danh sách các bảng dữ liệu

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên bảng** | **Mô tả chức năng** |
| 1 | nguoidung | Quản lý tài khoản người dùng và phân quyền hệ thống |
| 2 | khachhang | Lưu trữ thông tin khách hàng |
| 3 | nhacungcap | Quản lý thông tin nhà cung cấp thực phẩm |
| 4 | sanpham | Quản lý thông tin sản phẩm |
| 5 | giohang | Quản lý giỏ hàng của khách hàng |
| 6 | khuyenmai | Quản lý các chương trình khuyến mãi |
| 7 | donhang | Quản lý đơn hàng |
| 8 | chitiet\_donhang | Quản lý chi tiết đơn hàng |
| 9 | lienhe | Quản lý thông tin liên hệ và phản hồi khách hàng |

### ****2.3.3. Cấu trúc chi tiết các bảng****

**Bảng** nguoidung: dùng để quản lý thông tin tài khoản đăng nhập của người dùng trên hệ thống. Bảng này lưu các thông tin xác thực và phân quyền người dùng.

1. Bảng nguoidung

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | ma\_nguoidung | VARCHAR(10) | Mã người dùng (Khóa chính) |
| 2 | ten\_dangnhap | VARCHAR(50) | Tên đăng nhập (duy nhất) |
| 3 | email | VARCHAR(100) | Email người dùng |
| 4 | matkhau | VARCHAR(255) | Mật khẩu (đã mã hóa) |
| 5 | hoten | VARCHAR(100) | Họ tên người dùng |
| 6 | vai\_tro | VARCHAR(20) | Vai trò (admin/staff /member) |
| 7 | sodienthoai | VARCHAR(15) | Số điện thoại |
| 8 | diachi | TEXT | Địa chỉ |
| 9 | ngay\_tao | DATETIME | Ngày tạo tài khoản |
| 10 | ngay\_capnhat | DATETIME | Ngày cập nhật |

1. Bảng khachhang

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | ma\_kh | VARCHAR(10) | Mã khách hàng(Khóa chính) |
| 2 | ten\_kh | VARCHAR(100) | Tên khách hàng |
| 3 | email | VARCHAR(100) | Email |
| 4 | sodienthoai | VARCHAR(20) | Số điện thoại |
| 5 | diachi | VARCHAR(255) | Địa chỉ |
| 6 | matkhau | VARCHAR(255) | Mật khẩu |

1. Bảng nhacungcap

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | ma\_ncc | VARCHAR(10) | Mã nhà cung cấp (Khóa chính) |
| 2 | ten\_ncc | VARCHAR(255) | Tên nhà cung cấp |
| 3 | diachi | VARCHAR(255) | Địa chỉ |
| 4 | sodienthoai | VARCHAR(20) | Số điện thoại |

1. Bảng sanpham

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | ma\_sp | VARCHAR(10) | Mã sản phẩm (Khóa chính) |
| 2 | ten\_sp | VARCHAR(255) | Tên sản phẩm |
| 3 | loai\_sp | VARCHAR(100) | Loại sản phẩm |
| 4 | mota | TEXT | Mô tả |
| 5 | gia | DECIMAL(10,2) | Giá bán |
| 6 | soluong\_ton | INT | Số lượng tồn |
| 7 | ma\_ncc | VARCHAR(10) | Mã nhà cung cấp (Khóa ngoại) |
| 8 | hinhanh | VARCHAR(255) | Hình ảnh sản phẩm |

1. Bảng giohang

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | ma\_giohang | VARCHAR(10) | Mã giỏ hàng (Khóa chính) |
| 2 | ma\_kh | VARCHAR(10) | Mã khách hàng (Khóa ngoại) |
| 3 | ma\_sp | VARCHAR(10) | Mã sản phẩm (Khóa ngoại) |
| 4 | soluong | INT | Số lượng |

1. Bảng khuyenmai

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | ma\_km | VARCHAR(10) | Mã khuyến mãi (Khóa chính) |
| 2 | ten\_km | VARCHAR(100) | Tên khuyến mãi |
| 3 | mota | TEXT | Mô tả |
| 4 | mucgiam | DECIMAL(5,2) | Mức giảm (%) |
| 5 | giatri\_don | DECIMAL(10,2) | Giá trị đơn áp dụng |
| 6 | ngay\_batdau | DATE | Ngày bắt đầu |
| 7 | ngay\_ketthuc | DATE | Ngày kết thúc |
| 8 | trangthai | VARCHAR(50) | Trạng thái |

1. Bảng đơn hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | ma\_donhang | VARCHAR(10) | Mã đơn hàng (Khóa chính) |
| 2 | ma\_kh | VARCHAR(10) | Mã khách hàng (Khóa ngoại) |
| 3 | ngay\_dat | DATETIME | Ngày đặt |
| 4 | tongtien | DECIMAL(10,2) | Tổng tiền |
| 5 | trangthai | VARCHAR(50) | Trạng thái |
| 6 | ma\_km | VARCHAR(10) | Mã khuyến mãi (Khóa ngoại) |

1. Bảng chitiet\_donhang

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | ma\_ctdh | VARCHAR(10) | Mã chi tiết đơn hàng (Khóa chính) |
| 2 | ma\_donhang | VARCHAR(10) | Mã đơn hàng (Khóa ngoại) |
| 3 | ma\_sp | VARCHAR(10) | Mã sản phẩm (Khóa ngoại) |
| 4 | soluong | INT | Số lượng |
| 5 | dongia | DECIMAL(10,2) | Đơn giá |

1. Bảng lienhe

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | id | INT | Mã liên hệ (Khóa chính, tự tăng) |
| 2 | ten | VARCHAR(255) | Tên người liên hệ |
| 3 | email | VARCHAR(255) | Email |
| 4 | noidung | TEXT | Nội dung liên hệ |
| 5 | traloi | TEXT | Nội dung trả lời |
| 6 | trangthai | VARCHAR(50) | Trạng thái xử lý |

1. Bảng Blog

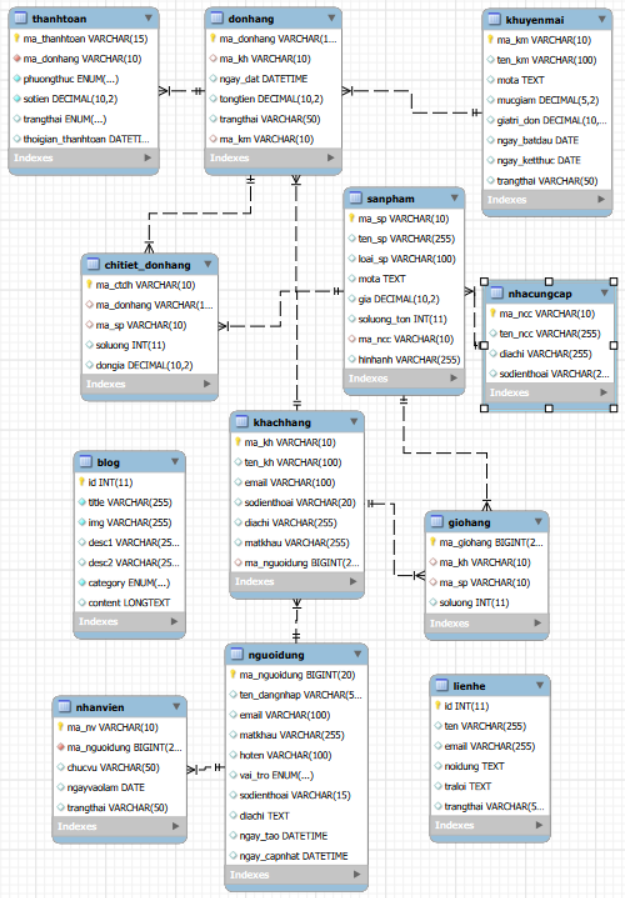
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | id | INT | Mã bài viết (Khóa chính, tự tăng) |
| 2 | title | VARCHAR(255) | Tiêu đề bài viết |
| 3 | img | VARCHAR(255) | Đường dẫn hình ảnh minh họa |
| 4 | desc1 | VARCHAR(255) | Mô tả ngắn 1 của bài viết |
| 5 | desc2 | VARCHAR(255) | Mô tả ngắn 2 của bài viết |
| 6 | category | ENUM('monan','rausach','suckhoe') | Danh mục bài viết (Món ăn, Rau sạch, Sức khỏe) |
| 7 | content | LONGTEXT | Nội dung chi tiết bài viết |

1. Bảng thanhtoan

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | ma\_tt | VARCHAR(10) | Mã thanh toán (Khóa chính) |
| 2 | ma\_donhang | VARCHAR(10) | Mã đơn hàng (Khóa ngoại) |
| 3 | phuongthuc | ENUM('COD','Chuyển khoản','Ví điện tử') | Phương thức thanh toán |
| 4 | sotien | DECIMAL(10,2) | Số tiền thanh toán |
| 5 | trangthai | VARCHAR(50) | Trạng thái thanh toán |
| 6 | thoigian | DATETIME | Thời gian thanh toán |

1. Bảng nhanvien

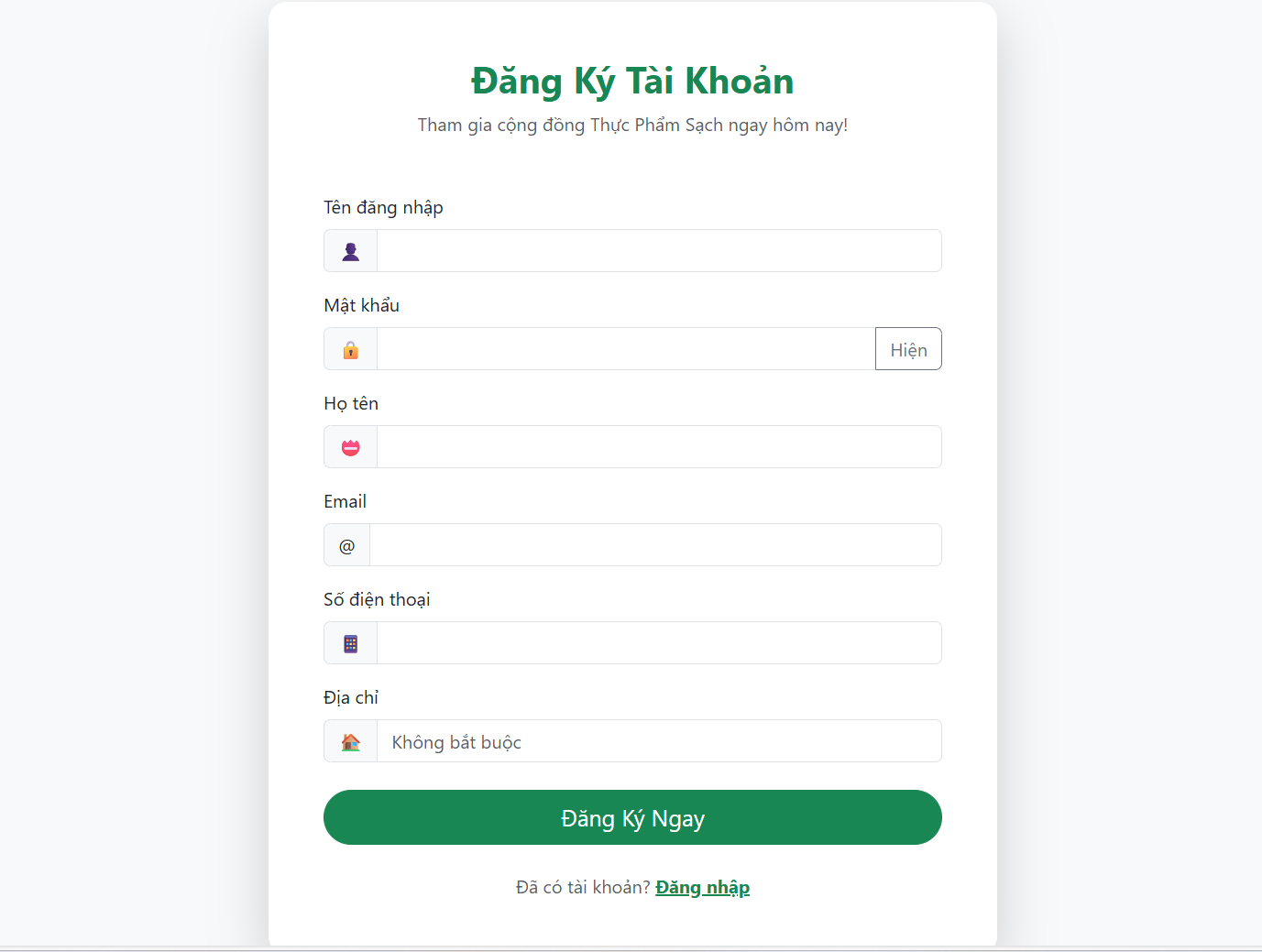
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | ma\_nv | VARCHAR(10) | Mã nhân viên (Khóa chính) |
| 2 | ma\_nguoidung | BIGINT(20) | Mã người dùng (Khóa ngoại) |
| 3 | chucvu | VARCHAR(50) | Chức vụ nhân viên |
| 4 | ngayvaolam | DATE | Ngày vào làm |
| 5 | trangthai | VARCHAR(50) | Trạng thái làm việc |



Mối quan hệ giữa các bảng

# CHƯƠNG 3: XÂY DỰNG HỆ THỐNG

## 3.1. Giao diện



1. Ảnh đăng ký

\* Mã nguồn đăng kí

import { useState } from "react";

import { register } from "../api/auth.js";

import { Link, useNavigate } from "react-router-dom";

import { Container, Card, Form, Button, InputGroup, Alert } from "react-bootstrap";

function Register() {

  const [form, setForm] = useState({

    ten\_dangnhap: "",

    matkhau: "",

    hoten: "",

    email: "",

    sodienthoai: "",

    diachi: "",

  });

  const [errors, setErrors] = useState({});

  const [showPassword, setShowPassword] = useState(false);

  const [isLoading, setIsLoading] = useState(false);

  const navigate = useNavigate();

  const handleChange = (e) => {

    const { name, value } = e.target;

    setForm({ ...form, [name]: value });

    if (errors[name]) setErrors({ ...errors, [name]: "" });

  };

  const validate = () => {

    const newErrors = {};

    if (!form.ten\_dangnhap.trim()) newErrors.ten\_dangnhap = "Tên đăng nhập bắt buộc";

    if (!form.matkhau || form.matkhau.length < 6) newErrors.matkhau = "Mật khẩu ít nhất 6 ký tự";

    if (!form.hoten.trim()) newErrors.hoten = "Họ tên bắt buộc";

    if (!form.email.trim() || !/\S+@\S+\.\S+/.test(form.email)) newErrors.email = "Email không hợp lệ";

    if (!form.sodienthoai.trim()) newErrors.sodienthoai = "Số điện thoại bắt buộc";

    return newErrors;

  };

  const handleSubmit = async (e) => {

    e.preventDefault();

    const validationErrors = validate();

    if (Object.keys(validationErrors).length > 0) {

      setErrors(validationErrors);

      return;

    }

    setIsLoading(true);

    try {

      await register(form);

      alert("Đăng ký thành công! Hãy đăng nhập nhé 🥬");

      navigate("/login");

    } catch (err) {

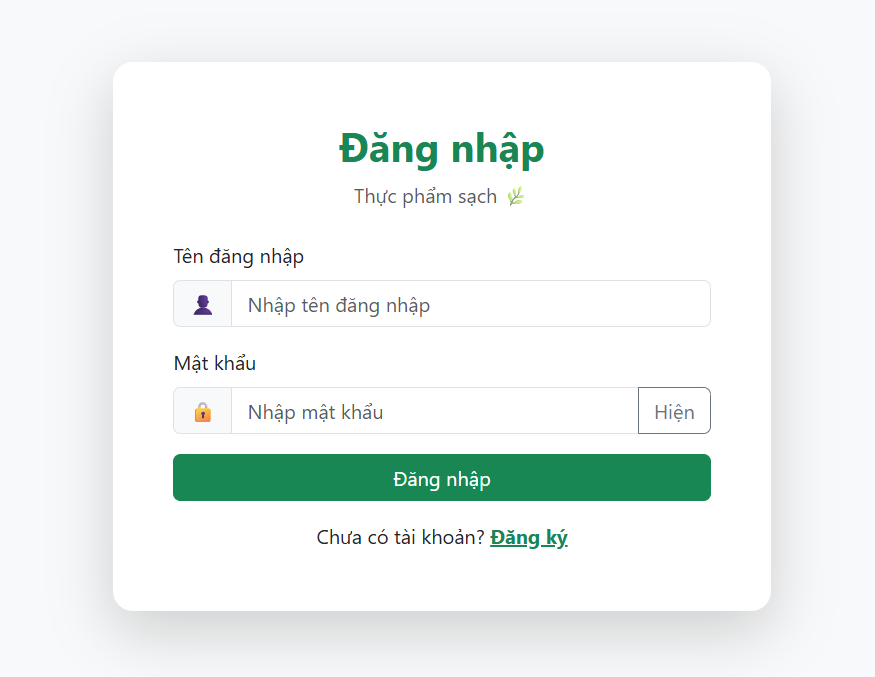
      setErrors({ server: err.response?.data?.error || "Lỗi đăng ký, thử lại sau" });

    } finally {

      setIsLoading(false);

    }

  };



1. Ảnh đăng nhập

\* Mã nguốn đăng nhập

import { useState } from "react";

import { login } from "../api/auth.js";

import { Link, useNavigate } from "react-router-dom";

import {

  Container,

  Card,

  Form,

  Button,

  InputGroup,

  Alert,

} from "react-bootstrap";

import { useAuth } from "../context/AuthContext.jsx";

function Login() {

  const [ten\_dangnhap, setTenDangNhap] = useState("");

  const [matkhau, setMatKhau] = useState("");

  const [error, setError] = useState("");

  const [showPassword, setShowPassword] = useState(false);

  const [isLoading, setIsLoading] = useState(false);

  const navigate = useNavigate();

  const { setUser } = useAuth();

  const handleSubmit = async (e) => {

    e.preventDefault();

    if (!ten\_dangnhap.trim()) {

      setError("Vui lòng nhập tên đăng nhập");

      return;

    }

    if (!matkhau) {

      setError("Vui lòng nhập mật khẩu");

      return;

    }

    setError("");

    setIsLoading(true);

    try {

      // 🔐 GỌI API LOGIN

      const res = await login(ten\_dangnhap, matkhau);

      localStorage.setItem("token", res.token);

      localStorage.setItem("user", JSON.stringify(res.user));

      setUser(res.user);

      alert("Đăng nhập thành công!");

      // 🔀 PHÂN QUYỀN

      if (res.user.vai\_tro === "admin" || res.user.vai\_tro === "member") {

        navigate("/admin");

      } else {

        navigate("/");

      }

    } catch (err) {

      setError(

        err.response?.data?.error ||

          err.response?.data?.message ||

          "Tên đăng nhập hoặc mật khẩu không đúng"

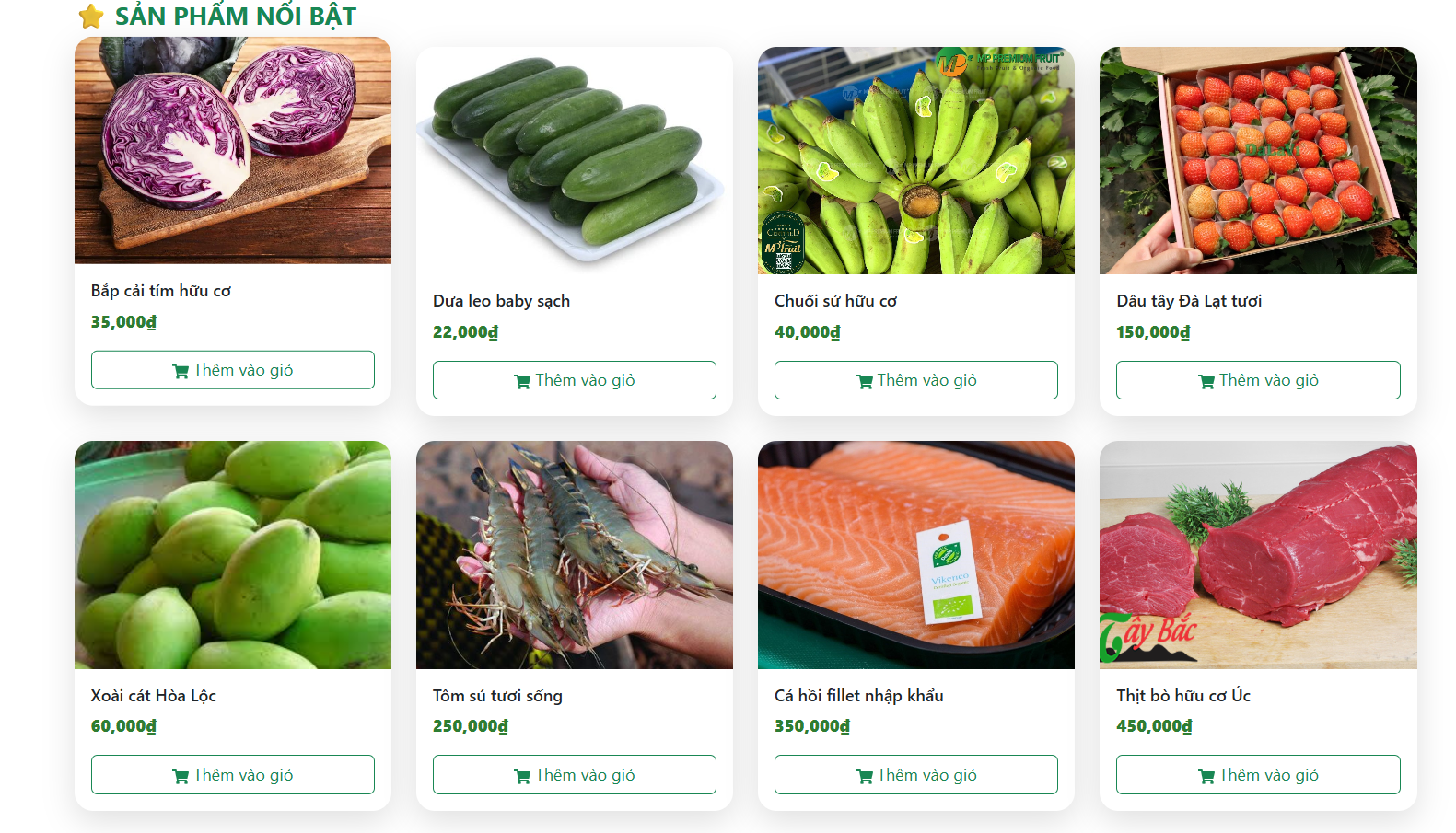
      );

    } finally {

      setIsLoading(false);

    }

  };



1. Danh sách sản phẩm

\* Mã nguồn sản phẩm

import { useEffect, useState } from "react";

import { getAllSanPham } from "../api/sanpham.js";

import { Link, useLocation, useNavigate } from "react-router-dom";

import Header from "../components/Header";

import {

  Container,

  Row,

  Col,

  Card,

  Button,

  Form,

  InputGroup,

} from "react-bootstrap";

import { FaHome, FaShoppingCart } from "react-icons/fa";

import { useAuth } from "../context/AuthContext.jsx";

import { useCart } from "../context/CartContext.jsx";

function Products() {

  const [products, setProducts] = useState([]);

  const [filteredProducts, setFilteredProducts] = useState([]);

  const [loading, setLoading] = useState(true);

  const [error, setError] = useState(null);

  const [searchTerm, setSearchTerm] = useState("");

  const location = useLocation();

  const searchParams = new URLSearchParams(location.search);

  const categoryQuery = searchParams.get("category");

  const searchQuery = searchParams.get("search");

  const categoryTitles = {

    "rau-cu": "Rau Củ Sạch",

    "hoa-qua": "Hoa Quả Tươi",

    "hai-san": "Hải Sản Tươi Sống",

    "do-kho": "Đồ Khô Hữu Cơ",

    "theo-mua": "Thực Phẩm Theo Mùa",

    "thit": "Thịt Sạch",

  };

  const currentCategoryTitle = categoryTitles[categoryQuery] || "Tất Cả Sản Phẩm";

  const { user } = useAuth();

  const navigate = useNavigate();

  const { addToCart } = useCart();

  useEffect(() => {

    const fetchProducts = async () => {

      try {

        const data = await getAllSanPham();

        setProducts(data);

      } catch (err) {

        console.error(err);

        setError("Không thể tải sản phẩm");

      } finally {

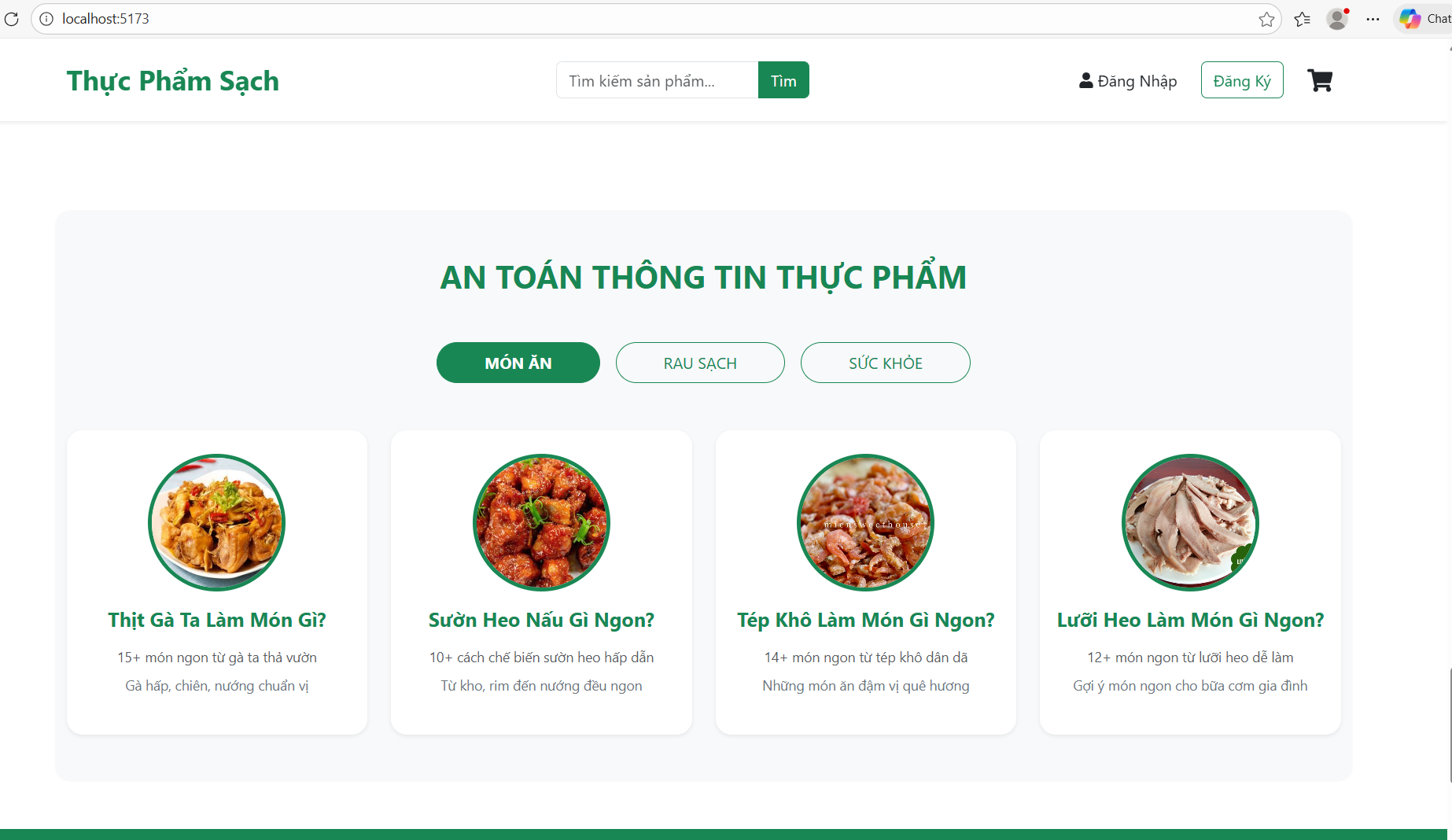
        setLoading(false);

      }

    };

    fetchProducts();

  }, []);



1. Giao diện Blog

\* Mã nguồn blog

import { useEffect, useState } from "react";

import { useParams } from "react-router-dom";

import { getBlogDetail } from "../api/blog.js";

function BlogDetail() {

  const { id } = useParams();

  const [blog, setBlog] = useState(null);

  useEffect(() => {

    async function fetchData() {

      try {

        const data = await getBlogDetail(id);

        setBlog(data);

      } catch (error) {

        console.error(error);

      }

    }

    fetchData();

  }, [id]);

  if (!blog) return <div className="container mt-5">Đang tải bài viết...</div>;

  return (

    <div className="container py-5">

      <h1 className="fw-bold text-success mb-4">{blog.title}</h1>

      <img

        src={blog.img}

        alt={blog.title}

        className="img-fluid rounded shadow mb-4"

        style={{ maxHeight: "400px", objectFit: "cover" }}

      />

      <p className="fs-5 text-muted">{blog.desc1}</p>

      <p className="fs-6 mb-4">{blog.desc2}</p>

      {/\* Nội dung chi tiết \*/}

      <div

        className="fs-5"

        dangerouslySetInnerHTML={{ \_\_html: blog.content }}

      ></div>

    </div>

  );

}

export default BlogDetail;

## 3.3. Kiểm thử hệ thống

**3.3.1. Kiểm thử chức năng**

Kiểm thử Đăng ký, Đăng nhập khách hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Mô tả** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| TC01 | Đăng ký tài khoản khách hàng với thông tin hợp lệ | Tài khoản được tạo thành công | Thông báo đăng ký thành công | PASS |
| TC02 | Đăng ký tài khoản với email đã tồn tại | Hệ thống thông báo lỗi “Email đã tồn tại” | Hiển thị thông báo lỗi | PASS |
| TC03 | Đăng nhập với tài khoản hợp lệ | Đăng nhập thành công, chuyển về trang chủ | Đăng nhập thành công | PASS |
| TC04 | Đăng nhập với mật khẩu sai | Hệ thống thông báo lỗi “Sai mật khẩu” | Hiển thị thông báo lỗi | PASS |

Kiểm thử chức năng giỏ hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Mô tả** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| TC05 | Khách hàng thêm sản phẩm hợp lệ vào giỏ hàng | Sản phẩm được thêm vào giỏ hàng | Thêm thành công | PASS |
| TC06 | Khách hàng thêm sản phẩm khi chưa đăng nhập | Hệ thống yêu cầu đăng nhập | Hiển thị thông báo yêu cầu đăng nhập | PASS |
| TC07 | Khách hàng cập nhật số lượng sản phẩm trong giỏ hàng | Số lượng sản phẩm được cập nhật | Cập nhật thành công | PASS |
| TC08 | Khách hàng xóa sản phẩm khỏi giỏ hàng | Sản phẩm bị xóa khỏi giỏ hàng | Xóa thành công | PASS |

c) Kiểm thử chức năng đặt hàng và thanh toán

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Mô tả** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| TC09 | Khách hàng đặt hàng với đầy đủ thông tin hợp lệ | Đơn hàng được tạo thành công | Tạo đơn hàng thành công | PASS |
| TC10 | Khách hàng đặt hàng khi giỏ hàng trống | Hệ thống thông báo lỗi “Giỏ hàng trống” | Hiển thị thông báo lỗi | PASS |
| TC11 | Khách hàng thanh toán thành công | Hệ thống cập nhật trạng thái đơn hàng | Thanh toán thành công | PASS |
| TC12 | Khách hàng thanh toán thất bại | Hệ thống thông báo lỗi thanh toán | Hiển thị thông báo lỗi | PASS |

**3.3.2.Kiểm thử hiệu suất**

Kiểm thử hiệu suất chức năng tải trang và hiển thị sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Mô tả** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| TC07 | Truy cập trang chủ website | Trang chủ tải trong ≤ 3 giây | Trang chủ tải nhanh, hiển thị đầy đủ nội dung | PASS |
| TC08 | Truy cập trang danh sách sản phẩm | Danh sách sản phẩm hiển thị trong ≤ 3 giây | Sản phẩm hiển thị nhanh, không lỗi | PASS |
| TC09 | Xem chi tiết một sản phẩm | Trang chi tiết sản phẩm tải trong ≤ 2 giây | Hiển thị đầy đủ thông tin sản phẩm | PASS |

Kiểm thử hiệu suất chức năng tìm kiếm sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Mô tả** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| TC10 | Tìm kiếm sản phẩm với từ khóa hợp lệ | Kết quả trả về trong ≤ 2 giây | Kết quả hiển thị nhanh | PASS |
| TC11 | Tìm kiếm với từ khóa không tồn tại | Hệ thống trả về thông báo không có sản phẩm | Hiển thị đúng thông báo | PASS |
| TC12 | Tìm kiếm liên tục nhiều lần | Hệ thống phản hồi ổn định, không treo | Hoạt động ổn định | PASS |

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* Kết luận

Sau quá trình nghiên cứu, phân tích, thiết kế và triển khai, đề tài **“Xây dựng website bán thực phẩm sạch HTFood sử dụng Node.js, React.js, Mysql”** đã được hoàn thành và đạt được các mục tiêu đề ra. Website được xây dựng theo mô hình thương mại điện tử hiện đại, đáp ứng các nhu cầu cơ bản của người dùng trong việc mua sắm thực phẩm sạch trực tuyến.

Hệ thống cho phép khách hàng thực hiện các chức năng như: xem và tìm kiếm sản phẩm theo danh mục, xem chi tiết sản phẩm, thêm sản phẩm vào giỏ hàng, đặt hàng và theo dõi trạng thái đơn hàng. Bên cạnh đó, website hỗ trợ đăng ký, đăng nhập và quản lý tài khoản người dùng, góp phần nâng cao trải nghiệm và đảm bảo an toàn thông tin.

Đối với phía quản trị, hệ thống cung cấp các chức năng quản lý sản phẩm, đơn hàng, khách hàng, blog và xử lý liên hệ. Việc phân quyền rõ ràng giữa các vai trò **admin, member và customer** giúp tăng tính bảo mật và nâng cao hiệu quả quản lý hệ thống.

Trong quá trình thực hiện đề tài, sinh viên đã vận dụng được các kiến thức về phân tích hệ thống, thiết kế cơ sở dữ liệu, lập trình backend với Node.js và Express.js, cũng như xây dựng giao diện frontend bằng React.js. Đề tài có ý nghĩa cả về mặt học thuật và thực tiễn, phù hợp với xu hướng phát triển của các website thương mại điện tử hiện nay. Tuy nhiên, do thời gian và kinh nghiệm còn hạn chế, hệ thống vẫn cần được tiếp tục hoàn thiện và phát triển thêm trong tương lai.

* Hướng phát triển

Trong thời gian tới, website có thể được mở rộng bằng việc tích hợp các hình thức **thanh toán trực** tuyến như VNPay, MoMo hoặc ZaloPay nhằm tăng sự tiện lợi cho khách hàng. Ngoài ra, hệ thống **đánh giá và bình luận sản phẩm** có thể được bổ sung để nâng cao mức độ tin cậy. Việc phát triển **ứng dụng di động** cũng là một hướng đi tiềm năng, giúp người dùng mua sắm dễ dàng và thuận tiện hơn trên các thiết bị di động.

Ngoài ra, hệ thống **báo cáo và thống kê thông minh** cho quản trị viên như thống kê doanh thu, số lượng đơn hàng, sản phẩm bán chạy và hành vi người dùng cũng là một hướng phát triển quan trọng. Kết hợp với đó là các chức năng **gợi ý sản phẩm** dựa trên lịch sử mua hàng, giúp nâng cao hiệu quả kinh doanh và cá nhân hóa trải nghiệm người dùng.

Với những hướng phát triển trên, website bán thực phẩm sạch hoàn toàn có khả năng trở thành một hệ thống thương mại điện tử hoàn chỉnh, đáp ứng tốt hơn nhu cầu thực tế của người dùng và phù hợp với xu hướng phát triển công nghệ trong tương lai.

# TÀI LIỆU THAM KHẢO

[1] Chen, P. (1976).  
The Entity–Relationship Model — Toward a Unified View of Data.  
ACM Transactions on Database Systems.  
Mô tả: Tài liệu nền tảng giới thiệu mô hình thực thể – quan hệ (ER Model), cách biểu diễn thực thể, thuộc tính và mối quan hệ trong thiết kế cơ sở dữ liệu.  
Truy cập: 10/01/2026  
🔗 [https://en.wikipedia.org/wiki/Entity%E2%80%93relationship\_model](https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model?utm_source=chatgpt.com)

[2] Sharma, A. K. (2015).  
Database Design for Real-World E-Commerce Systems.  
Academia.edu.  
Mô tả: Nghiên cứu thiết kế cơ sở dữ liệu cho hệ thống thương mại điện tử, phân tích các bảng: sản phẩm, đơn hàng, khách hàng, thanh toán, giỏ hàng.  
Truy cập: 21/12/2026  
🔗<https://www.academia.edu/27825091/Database_design_for_real_world_e_commerce_systems>

[3] Công ty TNHH Nông Sản Dũng Hà. (2026).  
Website Nông Sản Dũng Hà.  
Mô tả: Website thương mại điện tử thực phẩm sạch dùng làm tài liệu tham khảo về chức năng: xem sản phẩm, tìm kiếm, giỏ hàng, đặt hàng, blog, liên hệ.  
Truy cập: 20/01/2026  
🔗 <https://nongsandungha.com>

[4] Sommerville, I. (2016).  
Software Engineering (10th Edition).  
Pearson Education Limited.  
Mô tả: Tài liệu chuẩn về phân tích yêu cầu, Use Case, UML, thiết kế và phát triển hệ thống phần mềm.  
Truy cập: 11/01/2026

🔗 https://www.pearson.com/en-us/subject-catalog/p/software-engineering/P200000003352

[5] Dennis, A., Wixom, B. H., & Roth, R. M. (2015).  
Systems Analysis and Design (6th Edition).  
John Wiley & Sons.  
Mô tả: Hướng dẫn xây dựng Use Case, biểu đồ hoạt động, biểu đồ lớp và thiết kế cơ sở dữ liệu cho hệ thống thông tin.  
Truy cập: 13/01/2026

🔗 https://www.wiley.com/en-us/Systems+Analysis+and+Design-p-9781119346355

[6] Oracle Corporation. (2025).  
MySQL 8.0 Reference Manual.  
Oracle.  
Mô tả: Tài liệu chính thức về thiết kế bảng, khóa chính, khóa ngoại, chỉ mục và tối ưu truy vấn trong MySQL.  
Truy cập: 12/01/2026  
🔗 <https://dev.mysql.com/doc/refman/8.0/en/>

[7] Lucid Software Inc. (2025).  
UML Diagram Tutorial.  
Lucidchart.  
Mô tả: Hướng dẫn vẽ Use Case Diagram, Activity Diagram, Class Diagram.  
Truy cập: 03/01/2026  
🔗 <https://www.lucidchart.com/pages/uml-diagram>

[8] Draw.io Team. (2025).  
Diagrams for Software Design.  
Draw.io.  
Mô tả: Công cụ vẽ sơ đồ UML, ERD, Use Case cho đồ án phần mềm.  
Truy cập: 04/01/2026  
🔗 <https://www.drawio.com>

[9] GeeksforGeeks. (2025).  
E-Commerce Website Database Design.  
GeeksforGeeks.org.  
Mô tả: Ví dụ thiết kế cơ sở dữ liệu cho website bán hàng gồm sản phẩm, khách hàng, đơn hàng, thanh toán và giỏ hàng.  
Truy cập: 15/01/2026  
🔗 <https://www.geeksforgeeks.org/e-commerce-website-database-design/>