

## This image shows a full page of a document template designed for handwritten notes or answers. It features approximately 20 evenly spaced horizontal dotted lines across the entire width of the page, providing a guide for letter height and placement. The background is plain white, and there are no margins, headers, or footers visible.

(Ký tên và ghi rõ họ tên)

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Vĩnh Long, ngày ... tháng ..... năm .....*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

## **LỜI CẢM ƠN**

Tôi xin gửi lời cảm ơn sâu sắc đến Thầy đã hướng dẫn tôi trong quá trình viết bài báo cáo Đồ án “Chuyên ngành” Trong thời gian qua, tôi đã học được nhiều kiến thức mới trong lĩnh vực của ngành Công nghệ thông tin, trong suốt quá trình nghiên cứu cho đến kết thúc đồ án Thông qua bài báo cáo Đồ án này, tôi xin gửi lời cảm ơn đến thầy Phạm Minh Dương giảng viên Khoa Công nghệ thông tin, đã đem lại cho tôi những kiến thức hữu ích thông qua bài báo cáo Đồ án Chuyên ngành với chủ đề “Xây dựng WebSite Nhà Hàng Ẩm thực Phương Nam Vĩnh Long” Tôi cũng gửi lời cảm ơn đến Trường Đại học Trà Vinh, Trường Kỹ thuật và Công nghệ, Khoa Công nghệ thông tin đã tạo đủ điều kiện cũng như các kiến thức tài liệu học tập, tài liệu tham khảo để tôi hoàn thành Đồ án Chuyên ngành Trong quá trình viết báo cáo, tôi còn nhiều sai sót và khuyết điểm trong quá trình tìm hiểu và viết báo cáo Tôi rất mong nhận được sự đánh giá đóng góp ý kiến của Thầy để tôi khắc phục và hoàn thiện hơn

Tôi xin chân thành cảm ơn!

**MỤC LỤC**

LỜI CẢM ƠN .....	6
MỤC LỤC .....	7
DANH MỤC HÌNH ẢNH .....	11
DANH MỤC BẢNG BIỂU .....	12
DANH MỤC TỪ VIẾT TẮT .....	13
TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH .....	15
MỞ ĐẦU .....	16
CHƯƠNG 1. TỔNG QUAN .....	18
1.1. Bối cảnh và tính cấp thiết .....	18
1.2. Phạm vi và trọng tâm nghiên cứu .....	18
1.3. Các thành phần chính của nghiên cứu .....	19
1.3.1. Phân tích nhu cầu và đặc thù của nhà hàng .....	19
1.4. Thiết kế kiến trúc và chức năng website .....	19
1.5. Lựa chọn công nghệ phù hợp .....	20
1.6. Tối ưu trải nghiệm người dùng (UX/UI) .....	21
1.7. Phương pháp nghiên cứu .....	21
1.8. Phát triển và kiểm thử .....	21
1.9. Đánh giá và cải thiện .....	22
1.10. Mục tiêu cuối cùng .....	22
CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT .....	23
2.1. TỔNG QUAN VỀ HỆ THỐNG WEB .....	23
2.1.1. Kiến trúc ClientServer .....	23
2.1.2. Mô hình MVC (ModelViewController) .....	24

2.2. CÔNG NGHỆ FRONTEND .....	24
2.2.1. HTML (HyperText Markup Language) .....	24
2.2.2. CSS (Cascading Style Sheets) .....	26
2.2.3. Tailwind CSS .....	27
2.2.4. JavaScript (ES6+) .....	29
2.2.5. GSAP (GreenSock Animation Platform) .....	31
2.3. CÔNG NGHỆ BACKEND .....	34
2.3.1. Nodejs .....	34
2.3.2. Expressjs .....	35
2.3.3. MySQL .....	38
2.3.4. RESTful API .....	41
2.3.5. JWT (JSON Web Token) .....	44
2.3.6. Bcrypt .....	47
2.4. CÔNG CỤ HỖ TRỢ .....	48
2.4.1. Visual Studio Code .....	48
2.4.2. Postman .....	48
2.4.3. Git & GitHub .....	49
2.4.4. Docker .....	49
2.4.5. Nodejs vs Java (Spring Boot) .....	53
2.5. CÁC NGHIỆP VỤ LIÊN QUAN ĐẾN ĐỀ TÀI .....	54
2.5.1. Nghiệp vụ quản lý nhà hàng .....	54
2.5.2. Nghiệp vụ quản lý khách hàng (CRM) .....	56
2.5.3. Nghiệp vụ marketing và truyền thông .....	56
2.5.4. Nghiệp vụ báo cáo và thống kê .....	57
2.6. Các công trình nghiên cứu liên quan .....	57

2.7. KẾT LUẬN CHƯƠNG .....	59
CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU .....	60
3.1. Mô hình thực thể kết hợp ERD .....	60
3.1.1. Mô tả hệ thống .....	61
3.1.1.1 Tổng quan hệ thống .....	61
3.1.2. Kiến trúc hệ thống .....	62
3.1.3. Cấu trúc dự án .....	64
3.2. XÁC ĐỊNH CÁC YÊU CẦU CHỨC NĂNG CỦA HỆ THỐNG .....	66
3.2.1. Sơ đồ Usecase .....	66
3.3. THIẾT KẾ XỬ LÝ HỆ THỐNG .....	67
3.3.1. Thiết kế API RESTful .....	67
3.3.2. Sơ đồ tuần tự (Sequence Diagram) .....	69
3.4. MÔ HÌNH HOẠT ĐỘNG CỦA HỆ THỐNG .....	71
CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU .....	75
4.1. Tổng quan kết quả .....	75
4.2. Giao diện người dùng (Customer Interface) .....	75
4.2.1. Trang chủ (index.html) .....	75
4.2.2. Trang Thực đơn (thuc-don.html) .....	76
4.2.3. Trang Chi tiết món ăn (chitietmonan.html) .....	77
4.2.4. Trang Giỏ hàng (gio-hang.html) .....	77
4.2.5. Trang Thanh toán (thanh-toan.html) .....	78
4.2.6. Trang Đặt bàn (dat-ban.html) .....	79
4.2.7. Trang Đăng nhập / Đăng ký .....	79
4.2.8. Trang Tài khoản (tai-khoan.html) .....	81
4.2.9. Các trang khác .....	81

## XÂY DỰNG WEBSITE NHÀ HÀNG ĂM THỰC PHƯƠNG NAM VĨNH LONG

4.3. Giao diện quản trị (Admin Interface) .....	82
4.3.1. Dashboard .....	82
4.4. Kết luận chương .....	86
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	87
5.1. Kết luận .....	87
5.1.1. Kết quả đạt được .....	87
5.1.2. Đóng góp mới .....	88
5.1.3. Ý nghĩa thực tiễn .....	88
5.2. Hạn chế .....	89
5.3. Hướng phát triển .....	89
5.3.1. Phát triển ngắn hạn (3–6 tháng) .....	89
5.3.2. Phát triển trung hạn (6–12 tháng) .....	89
5.3.3. Phát triển dài hạn (1–2 năm) .....	90
5.4. Lời kết .....	90
TÀI LIỆU THAM KHẢO .....	91

**DANH MỤC HÌNH ẢNH**

Hình 3.1: Mô hình thực thể kết hợp ERD .....	60
Hình 3.2: Kiến trúc hệ thống .....	62
Hình 3.3: Cấu trúc dự án .....	64
Hình 3.4: Sơ đồ UseCase .....	66
Hình 3.5: Sơ đồ tuần tự .....	70
Hình 3.6: Quy trình đặt bàn trực tuyến .....	71
Hình 3.7: Quy trình đặt hàng .....	72
Hình 3.8: Quy trình Quản lý đơn hàng .....	73
Hình 4.1: Giao diện trang chủ .....	76
Hình 4.2: Giao diện trang Giới thiệu .....	76
Hình 4.3: Giao diện trang chi tiết món .....	77
Hình 4.4: Giao diện giỏ hàng .....	78
Hình 4.5: Giao diện trang thanh toán .....	78
Hình 4.6: Giao diện trang đặt bàn .....	79
Hình 4.7: Giao diện đăng nhập/đăng ký .....	80
Hình 4.8: Giao diện Tài khoản của tôi .....	81
Hình 4.9: Giao diện Quản trị .....	82
Hình 4.10: Các giao diện do Admin quản lý .....	85



**DANH MỤC BẢNG BIỂU**

<i>Bảng 2.1: Bảng so sánh Docker và Virtual Machine .....</i>	<i>49</i>
<i>Bảng 2.2: Bảng so sánh Nodejs và Java Spring Boot .....</i>	<i>53</i>

**DANH MỤC TỪ VIẾT TẮT**

<b>Từ viết tắt</b>	<b>Nghĩa đầy đủ (Tiếng Anh)</b>	<b>Nghĩa Tiếng Việt</b>
<b>ACID</b>	Atomicity, Consistency, Isolation, Durability	Tính nguyên tử, nhất quán, cô lập, bền vững
<b>AI</b>	Artificial Intelligence	Trí tuệ nhân tạo
<b>API</b>	Application Programming Interface	Giao diện lập trình ứng dụng
<b>CNTT</b>	Information Technology	Công nghệ thông tin
<b>COD</b>	Cash On Delivery	Thanh toán khi nhận hàng
<b>CRM</b>	Customer Relationship Management	Quản lý quan hệ khách hàng
<b>CRUD</b>	Create, Read, Update, Delete	Tạo, đọc, cập nhật, xóa
<b>CSRF</b>	Cross-Site Request Forgery	Tấn công giả mạo yêu cầu chéo trang
<b>CORS</b>	Cross-Origin Resource Sharing	Chia sẻ tài nguyên khác nguồn
<b>CSS</b>	Cascading Style Sheets	Ngôn ngữ định dạng phân tầng
<b>ERD</b>	Entity Relationship Diagram	Sơ đồ thực thể – kết hợp
<b>ES6</b>	ECMAScript 6	Phiên bản ES6 của JavaScript
<b>F&amp;B</b>	Food and Beverage	Thực phẩm và đồ uống

## XÂY DỰNG WEBSITE NHÀ HÀNG ẨM THỰC PHƯƠNG NAM VĨNH LONG

<b>Từ viết tắt</b>	<b>Nghĩa đầy đủ (Tiếng Anh)</b>	<b>Nghĩa Tiếng Việt</b>
<b>GPU</b>	Graphics Processing Unit	Bộ xử lý đồ họa
<b>GSAP</b>	GreenSock Animation Platform	Nền tảng tạo hiệu ứng GreenSock
<b>HTML</b>	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
<b>HTTP</b>	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
<b>I/O</b>	Input / Output	Vào / Ra dữ liệu
<b>IoT</b>	Internet of Things	Mạng lưới vạn vật kết nối Internet

## TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Website Nhà hàng Ẩm thực Phương Nam – Vĩnh Long được xây dựng như một không gian số tái hiện trọn vẹn hơi ấm và hương vị miền Tây, mang đến cho người dùng một trải nghiệm trực tuyến hiện đại, trực quan và đầy cảm xúc Dự án hướng đến việc giới thiệu đặc sản miền Tây bằng ngôn ngữ công nghệ, nơi từng món ăn, từng hình ảnh đều được kể lại bằng sự sinh động và tinh tế

Mục tiêu của dự án là tạo nên một hệ thống vừa giới thiệu món ăn vừa hỗ trợ hoạt động của nhà hàng: trình bày thực đơn đa dạng kèm hình ảnh sắc nét; cho phép khách đặt bàn và đặt món trực tuyến nhanh gọn; quản lý album ảnh, tin tức, đơn hàng và dữ liệu vận hành một cách hiệu quả; đồng thời đảm bảo trải nghiệm người dùng mượt mà trên mọi thiết bị

Phần giao diện được tạo dựng bằng HTML, CSS, JavaScript, kết hợp Tailwind CSS để mang lại thiết kế gọn gàng, hiện đại, và GSAP Animation để thêm vào những chuyển động mềm mại, đầy nhịp sống Hệ thống backend sử dụng Nodejs và Expressjs làm lõi xử lý nghiệp vụ, giao tiếp với cơ sở dữ liệu MySQL nhằm quản lý thông tin món ăn, người dùng, đơn hàng Cơ chế xác thực sử dụng JWT đảm bảo an toàn, trong khi Nodemailer hỗ trợ gửi email xác nhận đặt bàn và đơn hàng một cách chuyên nghiệp

Website mang đến bộ tính năng đầy đủ: quản lý thực đơn theo danh mục, đặt bàn trực tuyến, giỏ hàng và đặt món online, album ảnh không gian – món ăn – sự kiện, mục tin tức cập nhật hoạt động, hệ thống đăng ký/đăng nhập bảo mật và giao diện admin toàn diện cho quản lý nội dung

Ý nghĩa của dự án không chỉ nằm ở việc tạo ra một công cụ hỗ trợ vận hành nhà hàng, mà còn ở việc kết nối ẩm thực truyền thống với công nghệ hiện đại Sản phẩm thể hiện tư duy thiết kế, khả năng lập trình và tinh thần đổi mới của sinh viên Công nghệ thông tin, đủ tính thực tiễn để triển khai trong môi trường kinh doanh thực tế và mở ra hướng phát triển cho các giải pháp nhà hàng trong thời đại số

## MỞ ĐẦU

### 1. Lý do chọn đề tài

Xu hướng chuyển đổi số: Ngành ẩm thực đang dần số hóa để thích nghi với thói quen tiêu dùng hiện đại Khách hàng ngày càng ưa chuộng đặt bàn, xem thực đơn và đặt món trực tuyến thay vì gọi điện hay đến trực tiếp

Nhu cầu của nhà hàng: Nhiều nhà hàng địa phương, đặc biệt ở vùng miền Tây chưa có website chuyên nghiệp để quảng bá và quản lý kinh doanh hiệu quả

Quảng bá ẩm thực địa phương: Cần một nền tảng để giới thiệu đặc sản, văn hóa ẩm thực Vĩnh Long miền Tây đến du khách trong và ngoài tỉnh

Ứng dụng kiến thức: Đề tài giúp vận dụng kiến thức về lập trình web, cơ sở dữ liệu, thiết kế giao diện vào sản phẩm thực tế

Kỹ năng phát triển: Rèn luyện kỹ năng phân tích, thiết kế hệ thống, lập trình fullstack và triển khai ứng dụng web hoàn chỉnh

### 2. Mục tiêu

*Mục tiêu chung:*

Xây dựng website quản lý nhà hàng Âm thực Phương Nam Vĩnh Long với đầy đủ chức năng quản lý và hỗ trợ kinh doanh, mang đến trải nghiệm người dùng tốt và giúp nhà hàng hoạt động hiệu quả hơn

*Mục tiêu cụ thể:*

Thiết kế giao diện responsive, thân thiện, dễ sử dụng trên mọi thiết bị

Xây dựng hệ thống backend RESTful xử lý đặt bàn, đặt món, quản lý đơn hàng

Quản lý dữ liệu món ăn, khách hàng, đơn hàng với MySQL và chức năng CRUD đầy đủ

Đảm bảo bảo mật, hiệu năng và trải nghiệm người dùng ổn định, nhanh, mượt

### **3. Nội dung nghiên cứu**

Nghiên cứu lý thuyết

Công nghệ web: HTML, CSS, JavaScript, NodeJS, ExpressJS, cơ sở dữ liệu MySQL và thiết kế dữ liệu quan hệ, API RESTful và các giải pháp bảo mật web cơ bản, phân tích và thiết kế hệ thống, Phân tích yêu cầu người dùng và quản trị viên, Thiết kế cơ sở dữ liệu, kiến trúc hệ thống, Thiết kế giao diện UI/UX bằng wireframe và mockup

### **4. Đối tượng và nội dung nghiên cứu**

*Đối tượng nghiên cứu:*

Hệ thống website: Ứng dụng web quản lý nhà hàng fullstack

Công nghệ: Nodejs, Expressjs, MySQL, HTML/CSS/JavaScript, Tailwind CSS

Nhiệm vụ: Quy trình đặt bàn, đặt món, quản lý đơn hàng của nhà hàng

*Phạm vi nghiên cứu*

#### **Về không gian:**

Nhà hàng Âm thực Phương Nam tại Vĩnh Long, khách hàng địa phương và du khách đến Vĩnh Long, có thể áp dụng cho các nhà hàng tương tự

#### **Về chức năng:**

Hiển thị thực đơn, album ảnh, tin tức, đặt bàn, đặt món online, giỏ hàng và thanh toán, xác thực người dùng, quản trị admin cơ bản

### **5. Phương pháp nghiên cứu**

#### **Phương pháp nghiên cứu tài liệu:**

Tra cứu tài liệu về công nghệ web: Nodejs, Expressjs, MySQL

Nghiên cứu các framework: Tailwind CSS, GSAP Animation

Tham khảo các website nhà hàng hiện có để học hỏi UX/UI

Đọc tài liệu chuẩn về API RESTful, bảo mật web

## CHƯƠNG 1. TỔNG QUAN

### 1.1. Bối cảnh và tính cấp thiết

Sự phát triển mạnh mẽ của Internet và các công nghệ web hiện đại đã thay đổi cách thức kinh doanh của ngành ẩm thực trên toàn cầu. Việc xây dựng website cho nhà hàng không còn là lựa chọn mà đã trở thành yêu cầu cần thiết để tồn tại và phát triển trong thời đại số. Điều này đặc biệt quan trọng đối với Nhà hàng Âm thực Phương Nam Vĩnh Long, nơi lưu giữ tinh hoa ẩm thực miền Tây với những món ăn đậm đà bản sắc quê hương.

Một hệ thống website toàn diện không chỉ giúp nhà hàng quảng bá thương hiệu, giới thiệu thực đơn phong phú, mà còn cung cấp các dịch vụ đặt bàn và đặt món trực tuyến, đáp ứng nhu cầu của thực khách hiện đại. Những người ngày càng ưa chuộng sự tiện lợi, nhanh chóng trong việc tìm kiếm thông tin và đặt dịch vụ qua Internet. Đây chính là động lực thúc đẩy việc phát triển đề tài này.

### 1.2. Phạm vi và trọng tâm nghiên cứu

Đề tài tập trung vào việc xây dựng một hệ thống website fullstack hoàn chỉnh, bao gồm cả Frontend (giao diện người dùng) và Backend (xử lý nghiệp vụ, quản trị dữ liệu). Đây là yếu tố then chốt quyết định sự thành công của nhà hàng trong môi trường số, ảnh hưởng trực tiếp đến:

Ấn tượng ban đầu: Giao diện đẹp mắt, chuyên nghiệp tạo niềm tin với khách hàng.

Trải nghiệm người dùng: Dễ sử dụng, tìm kiếm thông tin nhanh chóng.

Tỷ lệ chuyển đổi: Khách hàng dễ dàng đặt bàn, đặt món từ website.

Quản lý hiệu quả: Hệ thống backend giúp nhà hàng quản lý đơn hàng, thực đơn một cách khoa học.

Giao diện phải được thiết kế sao cho vừa hấp dẫn về mặt thẩm mỹ, vừa thân thiện với người dùng, đồng thời đảm bảo hiệu suất tốt và hiển thị tối ưu trên mọi thiết bị (máy tính để bàn, máy tính bảng, điện thoại di động). Backend cần xử lý nghiệp vụ chính xác, bảo mật, và hỗ trợ quản trị dễ dàng.

### 1.3. Các thành phần chính của nghiên cứu

#### 1.3.1. Phân tích nhu cầu và đặc thù của nhà hàng

Nhận diện bản sắc: Nghệ thuật ẩm thực miền Nam Việt Nam với hương vị đậm đà, đặc trưng là điểm nhấn cần được thể hiện rõ nét qua màu sắc ấm áp (cam, đỏ, vàng), hình ảnh món ăn chất lượng cao, và bố cục giao diện gần gũi, mộc mạc nhưng hiện đại

Xác định đối tượng khách hàng: Nghiên cứu hướng đến hai nhóm chính:

Cư dân địa phương (Vĩnh Long và vùng lân cận)

Du khách trong nước muốn trải nghiệm ẩm thực miền Tây

Khách hàng trẻ tuổi, quen với công nghệ và ưa thích đặt món online

### 1.4. Thiết kế kiến trúc và chức năng website

#### *Frontend (Giao diện người dùng):*

Trang chủ của website đóng vai trò như cánh cửa đầu tiên chạm vào cảm xúc khách hàng, nơi giới thiệu tổng quan về nhà hàng với banner nổi bật và các món ăn tiêu biểu được trình bày hấp dẫn. Khu vực thực đơn được thiết kế khoa học, phân chia theo từng danh mục, hiển thị hình ảnh món ăn rõ nét cùng mức giá minh bạch, giúp khách hàng dễ dàng lựa chọn. Album ảnh trưng bày không gian nhà hàng, các món ăn đặc sắc và những khoảnh khắc sự kiện, góp phần truyền tải trọn vẹn phong cách và bản sắc ẩm thực. Mục tin tức liên tục cập nhật các chương trình khuyến mãi, sự kiện và bài viết chia sẻ về văn hóa ẩm thực, tạo sự kết nối lâu dài với khách hàng. Bên cạnh đó, website tích hợp chức năng đặt bàn trực tuyến theo ngày giờ linh hoạt, kết hợp đặt món với giỏ hàng và thanh toán online tiện lợi. Cuối cùng, trang liên hệ cung cấp đầy đủ thông tin địa chỉ, bản đồ định vị và form liên hệ, giúp khách hàng dễ dàng kết nối và tương tác với nhà hàng.



### ***Backend (Quản trị và xử lý nghiệp vụ):***

Hệ thống backend được xây dựng theo mô hình API RESTful, đóng vai trò trung tâm trong việc xử lý toàn bộ nghiệp vụ và giao tiếp với frontend. Backend đảm nhiệm quản lý dữ liệu món ăn, danh mục và giá cả, đồng thời xử lý các quy trình đặt bàn, đặt món và cập nhật trạng thái đơn hàng theo thời gian thực. Ngoài ra, hệ thống còn hỗ trợ quản lý album ảnh, tin tức và các chương trình khuyến mãi, giúp nội dung website luôn được cập nhật. Cơ chế xác thực và phân quyền được triển khai nhằm đảm bảo an toàn dữ liệu, phân biệt rõ quyền hạn giữa người dùng và quản trị viên. Bên cạnh đó, backend cung cấp các chức năng thống kê và báo cáo doanh thu, hỗ trợ nhà hàng theo dõi hiệu quả kinh doanh và đưa ra quyết định kịp thời.

### **1.5. Lựa chọn công nghệ phù hợp**

Dự án tập trung nghiên cứu và lựa chọn stack công nghệ tối ưu, phù hợp với quy mô và định hướng phát triển của nhà hàng. Ở phía frontend, HTML được sử dụng để xây dựng cấu trúc semantic, hỗ trợ tốt cho SEO; CSS kết hợp Tailwind CSS giúp tạo giao diện hiện đại, responsive và rút ngắn thời gian phát triển. JavaScript ES6+ đảm nhiệm xử lý tương tác và giao tiếp với API, trong khi GSAP Animation được áp dụng để tạo hiệu ứng chuyển động mượt mà, nâng cao trải nghiệm thị giác. Phía backend sử dụng NodeJS kết hợp ExpressJS nhằm xây dựng hệ thống fullstack JavaScript hiệu suất cao, dễ triển khai và mở rộng. MySQL được lựa chọn làm cơ sở dữ liệu quan hệ ổn định, phù hợp với dữ liệu có cấu trúc của nhà hàng. JWT đảm nhiệm xác thực người dùng theo mô hình stateless, trong khi Nodemailer hỗ trợ gửi email xác nhận và thông báo tự động.

#### **Lý do lựa chọn:**

Các công nghệ được lựa chọn đều là mã nguồn mở, giúp giảm chi phí triển khai nhưng vẫn đảm bảo chất lượng. Với cộng đồng lớn và tài liệu phong phú, việc học tập, phát triển và bảo trì hệ thống trở nên thuận lợi hơn. Stack công nghệ này có hiệu năng tốt, khả năng mở rộng cao và đặc biệt phù hợp với các nhà hàng có quy mô vừa và nhỏ, hướng tới chuyển đổi số bền vững.

### 1.6. Tối ưu trải nghiệm người dùng (UX/UI)

Nghiên cứu UX/UI tập trung vào việc tạo ra trải nghiệm thân thiện, dễ sử dụng và giàu cảm xúc. Về mặt hình ảnh, website sử dụng ảnh món ăn chất lượng cao với góc chụp hấp dẫn, kết hợp bảng màu ấm áp mang đậm nét văn hóa miền Tây. Icon và illustration được thiết kế gần gũi, dễ hiểu, giúp người dùng nhanh chóng nắm bắt thông tin. Về bố cục, hệ thống grid rõ ràng, khoảng trắng được sử dụng hợp lý nhằm tạo cảm giác thoáng đãng, cùng với hierarchy thông tin mạch lạc từ quan trọng đến thứ yếu. Về tương tác, navigation được tối giản, các form đặt bàn và đặt món được thiết kế ngắn gọn, dễ thao tác, kèm theo phản hồi tức thì khi người dùng thực hiện hành động. Animation được sử dụng có chọn lọc, đảm bảo mượt mà mà không gây khó chịu. Đồng thời, hiệu năng website được tối ưu thông qua lazy loading hình ảnh, minify CSS/JavaScript và khả năng responsive hoàn toàn trên mọi thiết bị.

### 1.7. Phương pháp nghiên cứu

Trong quá trình thực hiện, đề tài tiến hành nghiên cứu các website nhà hàng trong nước và quốc tế nhằm phân tích thiết kế, chức năng và trải nghiệm người dùng. Từ đó, rút ra những điểm mạnh cần học hỏi, những hạn chế cần tránh và nắm bắt các xu hướng thiết kế hiện đại để áp dụng phù hợp vào dự án.

### 1.8. Phát triển và kiểm thử

Quá trình phát triển bắt đầu từ việc xây dựng backend API và thiết kế cơ sở dữ liệu, sau đó tiến hành tích hợp với frontend để tạo thành hệ thống hoàn chỉnh. Website được kiểm thử chức năng và trải nghiệm người dùng trên nhiều thiết bị khác nhau, bao gồm desktop, tablet và mobile, cũng như trên các trình duyệt phổ biến. Bên cạnh đó, hệ thống còn được kiểm thử hiệu năng bằng các công cụ như PageSpeed Insights và Lighthouse, đồng thời kiểm tra các lỗ hổng bảo mật phổ biến như SQL Injection, XSS và CSRF nhằm đảm bảo an toàn khi vận hành thực tế.

### **1.9. Đánh giá và cải thiện**

Sau khi triển khai thử nghiệm, đề tài tiến hành thu thập phản hồi từ người dùng, kết hợp phân tích các số liệu như thời gian truy cập, tỷ lệ thoát trang và tỷ lệ chuyển đổi. Dựa trên dữ liệu thực tế, hệ thống được điều chỉnh và tối ưu liên tục nhằm nâng cao hiệu quả sử dụng và đáp ứng tốt hơn nhu cầu của nhà hàng cũng như khách hàng.

### **1.10. Mục tiêu cuối cùng**

Mục tiêu của nghiên cứu không chỉ dừng lại ở việc tạo ra một giao diện đẹp mắt, mà hướng đến xây dựng một hệ thống website toàn diện, vận hành ổn định và mang lại giá trị thực tiễn. Website góp phần hỗ trợ quá trình chuyển đổi số cho Nhà hàng Ẩm thực Phương Nam, nâng cao khả năng cạnh tranh trong thị trường F&B địa phương, gia tăng doanh thu thông qua kênh online và quảng bá văn hóa ẩm thực miền Tây đến đông đảo công chúng. Quan trọng hơn, đây là một sản phẩm có khả năng triển khai ngay trong thực tế, vượt ra khỏi khuôn khổ của một đồ án học thuật thuần túy.

## CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT

### 2.1. TỔNG QUAN VỀ HỆ THỐNG WEB

#### 2.1.1. Kiến trúc ClientServer

Hệ thống website nhà hàng được xây dựng theo mô hình kiến trúc Client – Server, trong đó phía client và server phối hợp chặt chẽ để đảm bảo hệ thống hoạt động ổn định và hiệu quả. Ở phía client, người dùng truy cập website thông qua các trình duyệt phổ biến như Chrome, Firefox, Safari hoặc Edge để xem và tương tác với giao diện [1]. Trình duyệt có nhiệm vụ gửi các yêu cầu HTTP đến máy chủ, tiếp nhận phản hồi từ server và hiển thị nội dung tương ứng cho người dùng, đồng thời xử lý các thao tác tương tác như đặt bàn, đặt món thông qua JavaScript. Ở phía server, hệ thống tiếp nhận và xử lý các yêu cầu từ client, thực hiện truy vấn cơ sở dữ liệu, xử lý logic nghiệp vụ như xác thực người dùng, quản lý đơn hàng và đặt bàn, sau đó trả về dữ liệu cho client dưới dạng JSON hoặc HTML, đảm bảo luồng thông tin thông suốt và nhất quán.

#### Luồng hoạt động:

- |  |
|--|
| 1 User nhập URL → Browser gửi HTTP Request                     |
| 2 Server nhận request → Xử lý logic → Truy vấn Database        |
| 3 Database trả kết quả → Server xử lý → Trả HTTP Response      |
| 4 Browser nhận response → Render giao diện → Hiển thị cho user |

### 2.1.2. Mô hình MVC (ModelViewController)

Dự án được triển khai theo mô hình MVC (Model – View – Controller) nhằm tổ chức mã nguồn một cách khoa học, rõ ràng và thuận tiện cho việc bảo trì, mở rộng trong tương lai. Trong đó, Model đại diện cho cấu trúc dữ liệu trong cơ sở dữ liệu, chịu trách nhiệm xử lý các nghiệp vụ liên quan đến truy vấn, thêm, sửa và xóa dữ liệu; diễn hình như các model quản lý món ăn, đơn hàng và người dùng. View đảm nhiệm vai trò hiển thị dữ liệu và giao diện tương tác với người dùng, được xây dựng bằng HTML, CSS và JavaScript ở phía frontend, không chứa logic xử lý nghiệp vụ nhằm đảm bảo tính tách biệt. Controller đóng vai trò trung gian, tiếp nhận các request từ client, gọi đến Model để xử lý dữ liệu tương ứng và trả response về cho View; các controller tiêu biểu trong hệ thống bao gồm MenuController, AuthController và CartController. [2]

## 2.2. CÔNG NGHỆ FRONTEND

### 2.2.1. HTML (HyperText Markup Language)

HTML là phiên bản mới nhất của ngôn ngữ đánh dấu siêu văn bản, được sử dụng để cấu trúc nội dung trên web

**Tính năng chính:**

**Semantic Elements:** <header>, <nav>, <section>, <article>, <footer> giúp cấu trúc rõ ràng, hỗ trợ SEO

**Form Controls:** Input types mới (email, date, number) với validation tích hợp sẵn

**Multimedia:** <video>, <audio> hỗ trợ phát media không cần plugin

**Canvas & SVG:** Vẽ đồ họa, animation trực tiếp trên trình duyệt

**Local Storage:** Lưu trữ dữ liệu phía client (giỏ hàng, phiên làm việc)

**Ứng dụng trong dự án:**

**<! Cấu trúc semantic >**

```
<header id="navbarcontainer"></header>
```

```
<main>
```

```
    <section class="hero"></section>
```

```
    <section class="menu"></section>
```

```
</main>
```

```
<footer id="footercontainer"></footer>
```

**<! Form đặt bàn >**

```
<form>
```

```
    <input type="email" required placeholder="Email của bạn">
```

```
    <input type="date" required>
```

```
    <input type="time" required>
```

```
</form>
```

### 2.2.2. CSS (Cascading Style Sheets)

CSS là ngôn ngữ định dạng giao diện, giữ vai trò thổi hồn cho website, biến những khối nội dung khô khan thành một không gian trực quan đẹp mắt và cuốn hút người dùng ngay từ cái nhìn đầu tiên. Với Flexbox, bố cục trở nên linh hoạt và việc căn chỉnh các phần tử theo nhiều hướng khác nhau được thực hiện một cách đơn giản và hiệu quả. Grid Layout cho phép chia layout theo dạng lưới phức tạp, tạo nên cấu trúc giao diện chuyên nghiệp và nhất quán. Các Transitions và Animations mang lại hiệu ứng chuyển động mượt mà, giúp trải nghiệm người dùng trở nên sinh động mà không gây rối mắt. Media Queries đảm bảo website hiển thị tốt trên mọi kích thước màn hình, từ desktop đến thiết bị di động. Bên cạnh đó, Custom Properties (CSS Variables) hỗ trợ quản lý tập trung các giá trị như màu sắc và kích thước, giúp việc chỉnh sửa và bảo trì giao diện trở nên nhanh chóng và dễ dàng hơn.

#### Ứng dụng trong dự án:

##### / Flexbox cho navigation /

```
navbar {  
    display: flex;  
    justifycontent: spacebetween;  
    alignitems: center;  
}
```

##### / Grid cho thực đơn /

```
menugrid {  
    display: grid;  
    gridtemplatecolumns: repeat(autofit, minmax(300px, 1fr));  
    gap: 2rem;  
}
```

**/ Responsive /**

```
@media (maxwidth: 768px) {  
  
    menugrid {  
  
        gridtemplatecolumns: 1fr;  
  
    }  
  
}
```

**/ Animation hover /**

```
menuitem:hover {  
  
    transform: scale(105);  
  
    transition: transform 03s ease;  
  
}
```

### **2.2.3. Tailwind CSS**

Tailwind CSS là một framework CSS theo hướng utility-first, cung cấp sẵn các class nhỏ gọn để xây dựng giao diện nhanh chóng mà không cần viết nhiều CSS custom thủ công. Nhờ cách tiếp cận này, quá trình phát triển được rút ngắn đáng kể khi lập trình viên chỉ cần kết hợp các class có sẵn để tạo layout và style mong muốn. Tailwind hỗ trợ responsive design rất linh hoạt thông qua các prefix như sm, md, lg, xl, giúp giao diện thích ứng mượt mà trên mọi kích thước màn hình. Bên cạnh đó, framework cho phép tùy biến sâu thông qua file cấu hình, từ màu sắc, khoảng cách đến font chữ, đảm bảo giao diện phù hợp với định hướng thương hiệu. Nhờ cơ chế PurgeCSS, các class không sử dụng sẽ được tự động loại bỏ, giúp kích thước file CSS luôn gọn nhẹ. Quan trọng hơn, Tailwind tạo ra một hệ thống thiết kế nhất quán, với spacing và bảng màu đồng bộ, giúp website giữ được sự hài hòa và chuyên nghiệp xuyên suốt.



**Ứng dụng trong dự án:**

**<! Card món ăn >**

```
<div class="bgwhite roundedxl shadowlg overflowhidden hover:shadow2xl
transitionshadow duration300">

  <img src="" class="wfull h64 objectcover">

  <div class="p6">

    <h3 class="text2xl fontbold textgray800 mb2">Bánh xèo</h3>

    <p class="textgray600 mb4">Món ăn đặc sản miền Tây</p>

    <div class="flex justifybetween itemscenter">

      <span class="textorange600 fontbold textxl">45000đ</span>

      <button class="bgorange600 textwhite px6 py2 roundedfull
hover:bgorange700">

        Đặt món

      </button>

    </div>

  </div>

</div>
```

**<! Responsive navigation >**

```
<nav class="hidden md:flex spacex8">

  <a class="textwhite hover:textorange300 transition">Trang chủ</a>

  <a class="textwhite hover:textorange300 transition">Thực đơn</a>

</nav>
```

### 2.2.4. JavaScript (ES6+)

JavaScript là ngôn ngữ lập trình phía client, giữ vai trò tạo nên sự sống động và khả năng tương tác cho website, giúp người dùng không chỉ “xem” mà còn “trải nghiệm”. Việc sử dụng ES6+ (ECMAScript 2015 trở lên) mang đến nhiều tính năng hiện đại, giúp code gọn gàng, dễ đọc và dễ bảo trì hơn. Arrow Functions được áp dụng để viết hàm với cú pháp ngắn gọn và trực quan. Template Literals cho phép nối chuỗi và chèn biến linh hoạt bằng dấu backtick, giúp xử lý nội dung động trở nên dễ dàng [3]. Destructuring hỗ trợ trích xuất nhanh các giá trị từ object hoặc array, làm code rõ ràng và súc tích hơn. Async/Await được sử dụng để xử lý các tác vụ bất đồng bộ một cách mạch lạc, dễ hiểu hơn so với Promise truyền thống. Ngoài ra, Modules với cơ chế import/export giúp chia nhỏ mã nguồn thành các file riêng biệt, tăng tính tổ chức và khả năng mở rộng cho toàn bộ hệ thống.

#### Ứng dụng trong dự án:

```
// Fetch API để lấy dữ liệu món ăn
async function loadMenu() {
  try {
    const response = await fetch('http://localhost:3000/api/menu');
    const result = await response.json();

    if (result.success) {
      displayMenu(result.data);
    }
  } catch (error) {
    console.error('Lỗi load menu:', error);
  }
}
```

```
// Thêm món vào giỏ hàng

const addToCart = (productId, quantity) => {

  const token = localStorage.getItem('token');

  fetch('http://localhost:3000/api/cart/add', {

    method: 'POST',

    headers: {

      'ContentType': 'application/json',

      'Authorization': `Bearer ${token}`

    },

    body: JSON.stringify({ ma_mon: productId, so_luong: quantity })

  })

  then(res => res.json())

  then(data => {

    if (data.success) {

      showNotification('Đã thêm vào giỏ hàng!');

      updateCartBadge();

    }

  });

};
```

```
// Event delegation

document.addEventListener('DOMContentLoaded', () => {

    loadMenu();

    loadCart();

    setupEventListeners();

});
```

### 2.2.5. GSAP (GreenSock Animation Platform)

GSAP là một thư viện JavaScript mạnh mẽ chuyên dùng để xây dựng animation mượt mà và hiệu suất cao, giúp giao diện website trở nên sống động mà vẫn đảm bảo trải nghiệm người dùng ổn định. Nhờ khả năng tối ưu hiệu năng và tận dụng GPU acceleration, các hiệu ứng do GSAP tạo ra có thể đạt tốc độ lên đến 60fps, chuyển động trơn tru và không gây giật lag. Thư viện cung cấp Timeline, cho phép kết hợp và kiểm soát nhiều animation liên tiếp hoặc song song, từ đó tạo ra các chuỗi chuyển động phức tạp một cách trực quan. Bên cạnh đó, ScrollTrigger giúp kích hoạt animation dựa trên hành vi cuộn trang của người dùng, tăng tính tương tác và chiều sâu trải nghiệm. Hệ thống ease functions đa dạng mang lại cảm giác chuyển động tự nhiên, mềm mại, góp phần nâng tầm chất lượng giao diện tổng thể của website.

**Ứng dụng trong dự án:**

```
// Animation fade in khi scroll

gsapregisterPlugin(ScrollTrigger);

gsapfrom('menuitem', {
  scrollTrigger: {
    trigger: 'menusection',
    start: 'top 80%',
  },
  opacity: 0,
  y: 50,
  duration: 0.8,
  stagger: 0.2
});
```

```
// Animation hero section
```

```
gsapfrom('herotitle', {  
    opacity: 0,  
    y: 50,  
    duration: 1,  
    ease: 'power3out'  
});
```

```
gsapfrom('heroimage', {  
    opacity: 0,  
    scale: 0.8,  
    duration: 1.2,  
    delay: 0.5,  
    ease: 'backout(1.7)'  
});
```

## 2.3. CÔNG NGHỆ BACKEND

### 2.3.1. Nodejs

NodeJS là môi trường runtime cho phép chạy JavaScript ở phía server, được xây dựng trên nền tảng Chrome V8 Engine, mang lại tốc độ xử lý nhanh và ổn định. Với cơ chế non-blocking I/O, NodeJS có khả năng xử lý bất đồng bộ hiệu quả, đáp ứng tốt các hệ thống phải phục vụ nhiều request đồng thời. Kiến trúc event-driven giúp NodeJS đặc biệt phù hợp với các ứng dụng realtime và các hệ thống web hiện đại. Bên cạnh đó, NPM cung cấp một hệ sinh thái package khổng lồ, hỗ trợ phát triển nhanh chóng và linh hoạt. Việc sử dụng một ngôn ngữ JavaScript cho cả frontend và backend giúp đồng bộ tư duy phát triển, giảm chi phí học tập và bảo trì. Nhờ khả năng scalable, NodeJS dễ dàng mở rộng theo chiều ngang, đáp ứng tốt nhu cầu tăng trưởng trong tương lai.

#### Ứng dụng trong dự án:

```
// Server cơ bản

const express = require('express');

const app = express();

const PORT = 3000;

app.listen(PORT, () => {
  console.log(`  Server đang chạy tại http://localhost:${PORT}`);
});
```

### **2.3.2. Expressjs**

ExpressJS là framework web tối giản và linh hoạt cho NodeJS, cung cấp đầy đủ các công cụ cần thiết để xây dựng các ứng dụng web và mobile một cách nhanh chóng và hiệu quả. ExpressJS hỗ trợ routing giúp định tuyến URL rõ ràng, dễ quản lý và mở rộng. Cơ chế middleware cho phép xử lý request và response theo từng bước trong một chuỗi logic, giúp kiểm soát luồng xử lý và tái sử dụng code hiệu quả. Framework này còn hỗ trợ nhiều template engine như EJS, Pug hay Handlebars, thuận tiện cho việc render giao diện động. Ngoài ra, ExpressJS cung cấp cơ chế xử lý lỗi tập trung, giúp dễ dàng theo dõi và khắc phục sự cố trong quá trình vận hành. Bên cạnh đó, khả năng serve static files cho phép phân phối trực tiếp các tài nguyên tĩnh như HTML, CSS, JavaScript và hình ảnh, đáp ứng tốt nhu cầu triển khai website hoàn chỉnh.

#### **Ứng dụng trong dự án:**

```
const express = require('express');

const cors = require('cors');

const app = express();

// Middleware

app.use(cors());

app.use(express.json());

app.use(express.static('public'));

// Routes

const menuRoutes = require('/routes/menu');

const authRoutes = require('/routes/auth');

const cartRoutes = require('/routes/cart');
```



```
appuse('/api/menu', menuRoutes);

appuse('/api/auth', authRoutes);

appuse('/api/cart', cartRoutes);


// Error handling

appuse((err, req, res, next) => {

  console.error(errstack);

  res.status(500).json({

    success: false,

    message: 'Có lỗi xảy ra!'

  });

});
```

### Cấu trúc Routes:

```
// routes/menups

const express = require('express');

const router = express.Router();

const db = require('/config/database');


// Lấy tất cả món ăn

router.get('/', async (req, res) => {

  try {

    const [rows] = await dbquery('SELECT FROM mon_an

    WHERE trang_thai = 1');
```

```
        resjson({ success: true, data: rows });

    } catch (error) {

        resstatus(500)json({ success: false, message: errormessage });

    }

});

// Lấy chi tiết món ăn

routerget('/:id', async (req, res) => {

    try {

        const [rows] = await dbquery('SELECT FROM mon_an

WHERE ma_mon = ?', [reqparamsid]);

        resjson({ success: true, data: rows[0] });

    } catch (error) {

        resstatus(500)json({ success: false, message: errormessage });

    }

});

moduleexports = router;
```

### **2.3.3. MySQL**

MySQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở được sử dụng rộng rãi trên toàn thế giới, đặc biệt phổ biến trong các hệ thống web hiện đại. MySQL tuân thủ đầy đủ các nguyên tắc ACID, đảm bảo tính toàn vẹn và nhất quán của dữ liệu trong suốt quá trình xử lý giao dịch [4]. Với khả năng xử lý hiệu năng cao, MySQL có thể đáp ứng khối lượng lớn truy vấn trong thời gian ngắn, phù hợp với các hệ thống có nhiều người dùng đồng thời. Cơ sở dữ liệu này cũng có tính mở rộng linh hoạt, cho phép nâng cấp theo nhu cầu phát triển của hệ thống. Về mặt bảo mật, MySQL cung cấp cơ chế phân quyền người dùng chặt chẽ, giúp kiểm soát truy cập dữ liệu hiệu quả. Ngoài ra, MySQL có khả năng hoạt động đa nền tảng, tương thích tốt với các hệ điều hành như Windows, Linux và macOS, thuận tiện cho triển khai và vận hành lâu dài.

#### **Cấu trúc Database trong dự án:**

Bảng người dùng

```
CREATE TABLE nguoi_dung (  
    ma_nguoi_dung INT PRIMARY KEY AUTO_INCREMENT,  
    ten_dang_nhap VARCHAR(50) UNIQUE NOT NULL,  
    mat_khau_hash VARCHAR(255) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    ho_ten VARCHAR(100),  
    so_dien_thoai VARCHAR(15),  
    dia_chi TEXT,  
    ngay_tao DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

Bảng danh mục

```
CREATE TABLE danh_muc (  
    ma_danh_muc INT PRIMARY KEY AUTO_INCREMENT,  
    ten_danh_muc VARCHAR(100) NOT NULL,  
    mo_ta TEXT,  
    thu_tu INT DEFAULT 0  
);
```

Bảng món ăn

```
CREATE TABLE mon_an (  
    ma_mon INT PRIMARY KEY AUTO_INCREMENT,  
    ten_mon VARCHAR(200) NOT NULL,  
    ma_danh_muc INT,  
    gia DECIMAL(10,2) NOT NULL,  
    mo_ta TEXT,  
    hinh_anh VARCHAR(500),  
    trang_thai TINYINT DEFAULT 1,  
    ngay_tao DATETIME DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (ma_danh_muc)  
    REFERENCES danh_muc(ma_danh_muc)  
);
```

### Kết nối MySQL với Nodejs:

```
// config/databasejs

const mysql = require('mysql2/promise');
require('dotenv').config();

const pool = mysql.createPool({
  host: process.env.DB_HOST,
  user: process.env.DB_USER,
  password: process.env.DB_PASSWORD,
  database: process.env.DB_NAME,
  port: process.env.DB_PORT || 3306,
  waitForConnections: true,
  connectionLimit: 10,
  queueLimit: 0
});

// Test connection
pool.getConnection()
  .then(connection => {
    console.log('✔ Kết nối database thành công!');
    connection.release();
  })
  .catch(err => {
    console.error('✗ Lỗi kết nối database:', err);
  });

module.exports = pool;
```

### **2.3.4. RESTful API**

REST (Representational State Transfer) là kiến trúc thiết kế API, sử dụng HTTP methods để thực hiện CRUD operations

#### **Nguyên tắc REST:**

RESTful API được xây dựng dựa trên các nguyên tắc cốt lõi nhằm đảm bảo hệ thống đơn giản, linh hoạt và hiệu quả. Trong đó, Stateless giúp mỗi request hoạt động độc lập, server không cần lưu trạng thái phiên, từ đó tăng khả năng mở rộng. Kiến trúc Client–Server tách biệt rõ ràng giữa frontend và backend, giúp phát triển và bảo trì dễ dàng hơn. Uniform Interface quy định việc sử dụng thống nhất các HTTP methods như GET, POST, PUT, DELETE, tạo nên API rõ ràng và dễ hiểu. Bên cạnh đó, tính Cacheable cho phép các response được lưu cache khi phù hợp, góp phần cải thiện hiệu năng và giảm tải cho hệ thống.

#### **HTTP Methods:**

GET: Lấy dữ liệu (Read)

POST: Tạo mới dữ liệu (Create)

PUT/PATCH: Cập nhật dữ liệu (Update)

DELETE: Xóa dữ liệu (Delete)

#### **API Endpoints trong dự án:**

##### **// Menu API**

GET /api/menu // Lấy tất cả món ăn

GET /api/menu/:id // Lấy chi tiết món ăn

GET /api/menu/category/:id // Lấy món ăn theo danh mục

##### **// Auth API**

POST /api/auth/register // Đăng ký tài khoản

POST /api/auth/login // Đăng nhập

POST /api/auth/forgotpassword // Quên mật khẩu

POST /api/auth/resetpassword // Đặt lại mật khẩu

### // Cart API

GET /api/cart // Lấy giỏ hàng

POST /api/cart/add // Thêm vào giỏ

PUT /api/cart/update/:id // Cập nhật số lượng

DELETE /api/cart/remove/:id // Xóa khỏi giỏ

### // Order API

GET /api/orders // Lấy danh sách đơn hàng

POST /api/orders // Tạo đơn hàng mới

GET /api/orders/:id // Chi tiết đơn hàng

PUT /api/orders/:id/status // Cập nhật trạng thái

### // Album API

GET /api/albums // Lấy tất cả album

GET /api/albums/category/:loai // Album theo loại

GET /api/albums/:id // Chi tiết album

### Response Format:

// Success

```
{  
  "success": true,  
  "data": [],  
  "message": "Thành công"  
}
```

```
// Error  
  
{  
  "success": false,  
  "message": "Mô tả lỗi",  
  "error": "Chi tiết lỗi (chỉ trong development)"  
}
```



### **2.3.5. JWT (JSON Web Token)**

JWT (JSON Web Token) là một chuẩn mở theo RFC 7519, được sử dụng để truyền tải thông tin một cách an toàn giữa các bên dưới dạng JSON object. JWT có cấu trúc gồm ba phần rõ ràng: Header chứa thông tin về loại token và thuật toán mã hóa được sử dụng; Payload lưu trữ các dữ liệu liên quan đến người dùng như user ID, email hoặc quyền truy cập; và Signature đóng vai trò chữ ký bảo mật, giúp xác thực tính toàn vẹn của token và ngăn chặn việc chỉnh sửa dữ liệu trái phép trong quá trình truyền tải.

#### **Ứng dụng trong dự án:**

```
// Tạo token khi đăng nhập

const jwt = require('jsonwebtoken');

const generateToken = (user) => {

  return jwt.sign(

    {

      userId: userma_nguoi_dung,

      email: useremail

    },

    process.env.JWT_SECRET,

    { expiresIn: '7d' }

  );

};
```

```
// Middleware xác thực token

const authenticateToken = (req, res, next) => {

  const authHeader = reqheaders['authorization'];

  const token = authHeader && authHeadersplit(' ')[1];

  if (!token) {

    return resstatus(401)json({

      success: false,

      message: 'Chưa đăng nhập'

    });

  }

  jwtverify(token, processenvJWT_SECRET, (err, user) => {

    if (err) {

      return resstatus(403)json({

        success: false,

        message: 'Token không hợp lệ'

      });

    }

    requser = user;

    next();

  });

};
```

```
// Sử dụng middleware

router.get('/cart', authenticateToken, async (req, res) => {

    const userId = req.user.userId;

    // Xử lý lấy giỏ hàng

});
```

### **2.3.6. Bcrypt**

Bcrypt là thư viện mã hóa mật khẩu sử dụng thuật toán blowfish, bảo mật cao

#### **Ứng dụng:**

```
const bcrypt = require('bcrypt');

// Hash mật khẩu khi đăng ký

const hashPassword = async (password) => {

  const saltRounds = 10;

  return await bcrypt.hash(password, saltRounds);

};

// So sánh mật khẩu khi đăng nhập

const comparePassword = async (password, hash) => {

  return await bcrypt.compare(password, hash);

};

// Trong route đăng ký

router.post('/register', async (req, res) => {

  const { username, password, email } = req.body;

  const hashedPassword = await hashPassword(password);

  await db.query(

    'INSERT INTO nguoi_dung (ten_dang_nhap, mat_khau_hash, email) VALUES (?, ?, ?)',

    [username, hashedPassword, email]

  );

});
```

## **2.4. CÔNG CỤ HỖ TRỢ**

### **2.4.1. Visual Studio Code**

VS Code là trình soạn thảo mã nguồn miễn phí và mạnh mẽ của Microsoft, hỗ trợ đa ngôn ngữ lập trình và được sử dụng rộng rãi trong phát triển phần mềm hiện đại. VS Code tích hợp IntelliSense giúp gợi ý code thông minh, tăng tốc độ lập trình và giảm lỗi cú pháp. Công cụ debugging cho phép kiểm tra và xử lý lỗi trực tiếp ngay trong editor, giúp tiết kiệm thời gian phát triển. Ngoài ra, Git Integration hỗ trợ quản lý version control hiệu quả, thuận tiện cho làm việc nhóm và theo dõi lịch sử thay đổi mã nguồn. Một điểm mạnh nổi bật khác của VS Code là hệ sinh thái extension phong phú, cho phép mở rộng chức năng theo nhu cầu [5]. Trong dự án, các extension được sử dụng bao gồm ESLint để kiểm tra và cảnh báo lỗi JavaScript, Prettier để tự động format code theo chuẩn, Live Server hỗ trợ xem trước giao diện HTML realtime, và MySQL extension giúp quản lý và thao tác với cơ sở dữ liệu một cách thuận tiện.

### **2.4.2. Postman**

Postman là công cụ chuyên dùng để kiểm thử API, cho phép gửi các HTTP request và trực quan hóa response một cách nhanh chóng và dễ hiểu. Trong dự án, Postman được sử dụng để kiểm tra các endpoint API, đảm bảo dữ liệu trả về đúng cấu trúc và nghiệp vụ. Công cụ này còn hỗ trợ tạo và quản lý collection các request, giúp tổ chức và tái sử dụng quá trình test hiệu quả. Bên cạnh đó, Postman cho phép tự động hóa kiểm thử API, giảm công sức test thủ công và nâng cao độ tin cậy của hệ thống. Ngoài ra, Postman còn hỗ trợ tạo tài liệu API, giúp việc chia sẻ và bảo trì hệ thống trở nên thuận tiện và chuyên nghiệp hơn.

### **2.4.3. Git & GitHub**

Git là hệ thống quản lý phiên bản phân tán, cho phép theo dõi và lưu lại toàn bộ lịch sử thay đổi của mã nguồn trong suốt quá trình phát triển dự án. GitHub là nền tảng lưu trữ mã nguồn trực tuyến dựa trên Git, hỗ trợ chia sẻ và quản lý code một cách thuận tiện. Trong dự án, Git và GitHub được sử dụng để quản lý phiên bản, giúp kiểm soát các thay đổi và dễ dàng quay lại những phiên bản trước khi cần thiết. Đồng thời, công cụ này hỗ trợ làm việc nhóm hiệu quả, cho phép nhiều thành viên cùng phát triển trên một dự án mà vẫn đảm bảo tính nhất quán. Ngoài ra, GitHub còn đóng vai trò như một giải pháp sao lưu mã nguồn an toàn và hỗ trợ triển khai hệ thống lên server, giúp quá trình phát triển và vận hành diễn ra liên tục và ổn định.

### **2.4.4. Docker**

Docker là nền tảng mã nguồn mở cho phép đóng gói ứng dụng và các dependencies vào các container đơn vị phần mềm độc lập có thể chạy nhất quán trên mọi môi trường

*Bảng 2.1: Bảng so sánh Docker và Virtual Machine*

<b>Tiêu chí</b>	Container	Virtual Machine
<b>Kích thước</b>	Nhẹ (MB)	Nặng (GB)
<b>Khởi động</b>	Nhanh (giây)	Chậm (phút)
<b>Tài nguyên</b>	Chia sẻ OS kernel	Mỗi VM có OS riêng
<b>Hiệu năng</b>	Gần native	Overhead cao

Container mang lại nhiều ưu điểm vượt trội trong quá trình phát triển và triển khai hệ thống, nổi bật với khả năng khởi động cực nhanh, tiết kiệm tài nguyên so với máy ảo truyền thống và đảm bảo triển khai nhất quán giữa các môi trường phát triển, kiểm thử và production. Nhờ tính portable, container có thể chạy ổn định ở hầu hết mọi nền tảng, từ máy cá nhân đến server hay cloud. Trong kiến trúc Docker, Docker Image đóng vai trò như một template chỉ đọc, chứa hệ điều hành tối giản, runtime và source code để tạo container, ví dụ như node:18-alpine, mysql:8 hay nginx:latest. Docker Container là instance đang chạy của image, có thể dễ dàng start, stop hoặc delete tùy theo nhu cầu sử dụng. Toàn bộ quá trình xây dựng image được định nghĩa trong Dockerfile, một file văn bản mô tả chi tiết các bước build, giúp việc đóng gói và triển khai ứng dụng trở nên tự động, rõ ràng và nhất quán.

### **Ví dụ Dockerfile:**

```
FROM node:18alpine
WORKDIR /app
COPY packagejson /
RUN npm ci only=production
COPY
EXPOSE 3000
CMD ["node", "serverjs"]
```

### Lợi ích cho dự án

#### Development:

# Setup nhanh 1 lệnh thay vì cài Node, MySQL, config

dockercompose up d

# → Môi trường đồng nhất cho tất cả developers

#### Production:

# Build và deploy dễ dàng

docker build t amthucbackend:latest

docker push amthucbackend:latest

docker run d p 3000:3000 amthucbackend:latest

# Scale nhanh

dockercompose up d scale backend=3

#### Các lệnh cơ bản:

# Quản lý Images

docker images # Liệt kê images

docker build t myapp:10 # Build image

docker pull node:18alpine # Tải image

# Quản lý Containers

docker ps # Container đang chạy

docker ps a # Tất cả containers

docker run d p 3000:3000 myapp

docker stop myapp

docker logs f myapp # Xem logs



```
# Docker Compose

dockercompose up d      # Start all services

dockercompose down      # Stop all

dockercompose logs f     # Xem logs

dockercompose restart backend # Restart service
```

### Triển khai:

```
# Development

dockercompose up d

# Production

dockercompose f dockercomposeprod.yml up d

# Access:

# Frontend: http://localhost

# Backend: http://localhost:3000

# MySQL: localhost:3306
```

### Kết luận

Docker mang lại nhiều giá trị thiết thực cho dự án khi giúp đảm bảo môi trường development luôn nhất quán, tránh tình trạng “chạy được trên máy em nhưng lỗi trên máy anh”. Việc setup nhanh chóng giúp tiết kiệm thời gian cấu hình ban đầu, trong khi khả năng deploy dễ dàng hỗ trợ đưa hệ thống lên server một cách gọn gàng và ổn định. Docker còn tạo ra sự cô lập giữa các service như backend, database hay web server, giúp hệ thống vận hành an toàn và dễ quản lý hơn. Với tính portable, ứng dụng có thể chạy ở hầu như mọi môi trường mà không cần thay

đổi cấu hình. Nhờ những ưu điểm đó, Docker đặc biệt phù hợp với dự án quy mô nhà hàng, dễ học, dễ triển khai và thuận tiện cho việc mở rộng trong tương lai.

#### **2.4.5. Nodejs vs Java (Spring Boot)**

*Bảng 2.2: Bảng so sánh Nodejs và Java Spring Boot*

<b>Tiêu chí</b>	Nodejs	Java Spring Boot
<b>Ngôn ngữ</b>	JavaScript	Java
<b>Kiến trúc</b>	Eventdriven, Singlethreaded	Multithreaded, Blocking I/O
<b>Hiệu năng</b>	Cao với I/O, kém với CPU	Cao với mọi loại operation
<b>Thời gian phát triển</b>	Nhanh	Chậm (nhiều boilerplate)
<b>Học tập</b>	Trung bình	Khó (cú pháp phức tạp, nhiều concept)
<b>Typing</b>	Dynamic (hoặc TypeScript)	Static typing mạnh mẽ
<b>Memory usage</b>	Nhẹ	Nặng (JVM)
<b>Startup time</b>	Nhanh (< 1s)	Chậm (35s)
<b>Enterprise</b>	Ít phổ biến	Rất phổ biến

<b>Tiêu chí</b>	Nodejs	Java Spring Boot
<b>Microservices</b>	Tốt	Xuất sắc

So với Java, NodeJS có nhiều ưu điểm nổi bật trong bối cảnh các dự án web hiện đại quy mô nhỏ và vừa. NodeJS cho phép phát triển nhanh hơn với lượng code ít hơn, cú pháp JavaScript thân thiện, dễ tiếp cận nên thời gian học ngắn hơn đáng kể. Thời gian khởi động ứng dụng nhanh, memory footprint nhỏ, giúp tiết kiệm tài nguyên server và giảm chi phí vận hành. Nhờ đó, NodeJS đặc biệt phù hợp với startup và các dự án nhỏ đến vừa, nơi tốc độ triển khai là yếu tố sống còn. Tuy nhiên, so với Java, NodeJS vẫn tồn tại một số hạn chế, như chưa thực sự mạnh trong các hệ thống enterprise lớn, type safety kém hơn nếu không sử dụng TypeScript, và không tối ưu cho các tác vụ CPU-intensive do kiến trúc single-thread. Dù vậy, việc lựa chọn NodeJS thay vì Java là hoàn toàn hợp lý với dự án nhà hàng có quy mô vừa và nhỏ, không yêu cầu kiến trúc phức tạp, cần ra sản phẩm sớm, đội ngũ phát triển gọn nhẹ và không có chuyên gia Java chuyên sâu. Ngoài ra, NodeJS còn giúp giảm chi phí server nhờ tiêu thụ ít RAM hơn, phù hợp với định hướng triển khai thực tế và bền vững.

## **2.5. CÁC NGHIỆP VỤ LIÊN QUAN ĐẾN ĐỀ TÀI**

### **2.5.1. Nghiệp vụ quản lý nhà hàng**

#### **a) Quản lý thực đơn (Menu Management)**

Quản lý thực đơn là nghiệp vụ cốt lõi, quyết định trực tiếp đến doanh thu và hình ảnh của nhà hàng. Nghiệp vụ này bao gồm việc phân loại món ăn theo từng danh mục rõ ràng như khai vị, món chính, tráng miệng, đồ uống hay các món đặc sản miền Tây nhằm giúp khách hàng dễ dàng lựa chọn. Bên cạnh đó, nhà hàng thực hiện định giá món ăn dựa trên chi phí nguyên liệu, công chế biến và mức lợi nhuận mong muốn. Thực đơn được cập nhật thường xuyên bằng cách thêm món mới, tạm ngừng các món không còn phù hợp và điều chỉnh giá theo mùa vụ. Việc quản lý hình ảnh món ăn với chất lượng cao cũng được chú trọng nhằm tăng sức hấp dẫn và thu hút khách hàng.

### **b) Nghiệp vụ đặt bàn (Table Reservation)**

Nghệ nghiệp vụ đặt bàn bao gồm cả hình thức truyền thống và trực tuyến, giúp tối ưu quá trình tiếp nhận khách. Nhà hàng tiếp nhận yêu cầu đặt bàn từ khách hàng thông qua điện thoại, trực tiếp tại quầy hoặc qua website. Sau đó, hệ thống kiểm tra tình trạng bàn trống dựa trên ngày giờ và số lượng khách để xác nhận khả dụng. Khi đặt bàn thành công, thông báo xác nhận sẽ được gửi qua email hoặc SMS, kèm theo tính năng nhắc nhở trước giờ hẹn. Khi khách đến, nhân viên tiến hành check-in và sắp xếp bàn phù hợp. Ngoài ra, hệ thống cũng hỗ trợ xử lý các yêu cầu hủy hoặc thay đổi lịch đặt một cách linh hoạt.

### **c) Nghiệp vụ đặt món và giao hàng (Order & Delivery)**

Nghệ nghiệp vụ đặt món và giao hàng bắt đầu từ việc khách hàng lựa chọn món ăn, thêm vào giỏ hàng và gửi yêu cầu đặt hàng. Nhà hàng tiếp nhận và kiểm tra thông tin đơn hàng trước khi xác nhận với khách. Đơn hàng sau đó được chuyển đến bộ phận bếp để chế biến và theo dõi tiến độ thực hiện. Khi hoàn tất, món ăn được đóng gói cẩn thận nhằm đảm bảo chất lượng trong quá trình vận chuyển. Đơn hàng được giao đến địa chỉ khách hàng, kết hợp với hình thức thanh toán phù hợp như trả tiền khi nhận hàng hoặc thanh toán online. Sau cùng, nhà hàng thu thập đánh giá và phản hồi từ khách hàng để cải thiện chất lượng dịch vụ.

### **d) Nghiệp vụ thanh toán (Payment Processing)**

Nghệ nghiệp vụ thanh toán được thiết kế đa dạng nhằm mang lại sự thuận tiện tối đa cho khách hàng. Nhà hàng hỗ trợ thanh toán tiền mặt khi nhận hàng (COD), thanh toán trực tuyến thông qua các cổng phổ biến như MoMo, VNPay, ZaloPay, cũng như hình thức chuyển khoản ngân hàng trước khi giao hàng. Sau khi hoàn tất thanh toán, hệ thống tiến hành xuất hóa đơn điện tử, đảm bảo tính minh bạch và thuận tiện cho việc lưu trữ, đối soát và quản lý tài chính.

### **2.5.2. Nghiệp vụ quản lý khách hàng (CRM)**

#### **a) Quản lý thông tin khách hàng**

Hệ thống hỗ trợ lưu trữ và quản lý đầy đủ thông tin cá nhân của khách hàng như họ tên, số điện thoại, email và địa chỉ liên hệ, tạo cơ sở dữ liệu tập trung và nhất quán. Đồng thời, toàn bộ lịch sử đặt bàn và đặt món của khách hàng được ghi nhận chi tiết, giúp nhà hàng theo dõi hành vi tiêu dùng theo thời gian. Thông qua việc phân tích dữ liệu, hệ thống có thể xác định sở thích ẩm thực và các món ăn yêu thích của từng khách hàng, từ đó phân loại khách hàng thành các nhóm như khách mới, khách thường xuyên hoặc khách VIP nhằm phục vụ và chăm sóc hiệu quả hơn.

#### **b) Chăm sóc khách hàng**

Nghiệp vụ chăm sóc khách hàng được triển khai nhằm duy trì mối quan hệ lâu dài và nâng cao mức độ hài lòng. Hệ thống cho phép gửi email chúc mừng sinh nhật, các dịp lễ đặc biệt để tạo sự gắn kết với khách hàng. Bên cạnh đó, các thông báo về chương trình khuyến mãi và món ăn mới được gửi kịp thời, giúp tăng khả năng quay lại sử dụng dịch vụ. Hệ thống cũng tiếp nhận và xử lý các khiếu nại, phản hồi từ khách hàng nhằm cải thiện chất lượng phục vụ. Ngoài ra, các chương trình tích điểm và ưu đãi dành cho thành viên được áp dụng để khuyến khích sự trung thành và gia tăng giá trị lâu dài cho nhà hàng.

### **2.5.3. Nghiệp vụ marketing và truyền thông**

#### **a) Quản lý thông tin khách hàng**

Hệ thống cho phép lưu trữ và quản lý đầy đủ thông tin cá nhân của khách hàng như họ tên, số điện thoại, email và địa chỉ liên hệ, tạo nền tảng cho việc phục vụ chính xác và nhất quán. Bên cạnh đó, lịch sử đặt bàn và đặt món của từng khách hàng được ghi nhận chi tiết, giúp nhà hàng theo dõi hành vi tiêu dùng và nhu cầu thực tế. Thông qua dữ liệu này, hệ thống có thể xác định sở thích ẩm thực, các món ăn yêu thích, từ đó phân loại khách hàng theo các nhóm như khách mới, khách thường xuyên hoặc khách VIP, hỗ trợ xây dựng chiến lược kinh doanh và chăm sóc phù hợp.

### **b) Chăm sóc khách hàng**

Nghệ thuật chăm sóc khách hàng tập trung vào việc duy trì mối quan hệ lâu dài và nâng cao mức độ hài lòng. Hệ thống hỗ trợ gửi email chúc mừng sinh nhật, ngày lễ nhằm tạo sự gắn kết và cảm giác được quan tâm. Đồng thời, các thông báo về chương trình khuyến mãi, món ăn mới được gửi đến khách hàng một cách kịp thời. Bên cạnh đó, hệ thống ghi nhận và xử lý các khiếu nại, phản hồi từ khách hàng để cải thiện chất lượng dịch vụ. Các chương trình tích điểm và ưu đãi dành cho thành viên cũng được triển khai nhằm khuyến khích khách hàng quay lại và tăng mức độ trung thành.

#### **2.5.4. Nghiệp vụ báo cáo và thống kê**

Hệ thống cung cấp các chức năng báo cáo và thống kê toàn diện, hỗ trợ nhà hàng theo dõi và đánh giá hiệu quả hoạt động kinh doanh một cách trực quan và chính xác. Báo cáo doanh thu được tổng hợp theo nhiều mốc thời gian như ngày, tuần, tháng và năm, giúp nhà hàng nắm bắt xu hướng tăng trưởng. Thống kê đơn hàng phản ánh rõ số lượng đơn, trạng thái xử lý và tỷ lệ hủy, từ đó hỗ trợ cải thiện quy trình vận hành. Dữ liệu phân tích món ăn cho phép xác định các món bán chạy cũng như những món ít được khách lựa chọn, làm cơ sở điều chỉnh thực đơn hợp lý. Bên cạnh đó, hệ thống thống kê hành vi khách hàng, phân biệt khách mới và khách quay lại, đồng thời đo lường tỷ lệ chuyển đổi. Cuối cùng, việc đánh giá hiệu quả kinh doanh được thực hiện thông qua so sánh các kênh bán hàng như phục vụ tại chỗ và kênh online, giúp nhà hàng đưa ra chiến lược phát triển phù hợp.

### **2.6. Các công trình nghiên cứu liên quan**

#### **Nghiên cứu trong nước**

Đề tài “Xây dựng website quản lý nhà hàng ABC” tại Đại học Công nghệ TP.HCM (2022) tập trung phát triển hệ thống quản lý nhà hàng với các chức năng cốt lõi như đặt bàn, quản lý thực đơn và thanh toán, sử dụng công nghệ PHP Laravel, MySQL và Bootstrap. Kết quả cho thấy hệ thống vận hành ổn định, giao diện responsive và phù hợp với nhu cầu thực tế. Tuy nhiên, đề tài còn hạn chế khi chưa tích hợp thanh toán online và phần animation còn đơn giản, từ đó rút ra bài học về tầm quan trọng của việc tối ưu UX/UI và đa dạng hóa phương thức thanh

toán.

Tiếp theo, đề tài “Ứng dụng web đặt món ăn trực tuyến” tại Đại học Cần Thơ (2023) xây dựng ứng dụng đặt món online cho chuỗi nhà hàng bằng React.js, Node.js và MongoDB. Hệ thống đạt được giao diện hiện đại, tốc độ tải nhanh, song chi phí triển khai cao và yêu cầu kỹ thuật phức tạp. Bài học rút ra là cần cân nhắc giữa việc áp dụng công nghệ mới và khả năng triển khai thực tế.

Đề tài “Hệ thống quản lý đặt bàn nhà hàng” của Đại học Bách khoa Hà Nội (2021) phát triển module đặt bàn với giao diện lịch và sơ đồ bàn, sử dụng Java Spring Boot, PostgreSQL và Angular. Hệ thống đạt chuẩn enterprise với mức bảo mật cao, nhưng thời gian phát triển dài và khó bảo trì với team nhỏ, cho thấy việc lựa chọn công nghệ phải phù hợp với quy mô dự án.

### **Nghiên cứu quốc tế**

Bài báo “Design and Implementation of Restaurant Management System” (Kumar & Patel, 2020) đăng trên International Journal of Computer Applications đề xuất mô hình kiến trúc 3-tier cho hệ thống quản lý nhà hàng, phân tách rõ Presentation, Business Logic và Data, là cơ sở để áp dụng mô hình phân tầng trong thiết kế hệ thống.

Nghiên cứu “User Experience Design for Food Ordering Applications” (Chen, Wang & Liu, 2021) tại ACM chỉ ra rằng 78% người dùng quyết định đặt món dựa trên hình ảnh, luồng checkout dưới 3 bước có thể tăng tỷ lệ chuyển đổi lên 40%, và animation mượt mà giúp nâng cao cảm nhận chất lượng ứng dụng. Những phát hiện này được ứng dụng vào việc thiết kế giao diện tập trung hình ảnh món ăn và tối ưu quy trình đặt hàng.

Bài báo “RESTful API Design Best Practices for E-commerce Applications” (Johnson & Smith, 2022) tại IEEE đưa ra các chuẩn thiết kế API cho thương mại điện tử, bao gồm cấu trúc response, xử lý lỗi và pagination, làm cơ sở áp dụng chuẩn RESTful cho backend.

Cuối cùng, nghiên cứu “Security Considerations in Web-based Food Ordering Systems” (Williams & Brown, 2023) phân tích các lỗ hổng bảo mật phổ biến và khuyến nghị sử dụng JWT với thời hạn ngắn, mã hóa mật khẩu bằng bcrypt (salt rounds  $\geq 10$ ), validate input phía server để chống SQL Injection và XSS, đồng

thời bắt buộc HTTPS cho mọi giao dịch. Các khuyến nghị này được áp dụng trực tiếp nhằm tăng cường mức độ an toàn cho hệ thống.

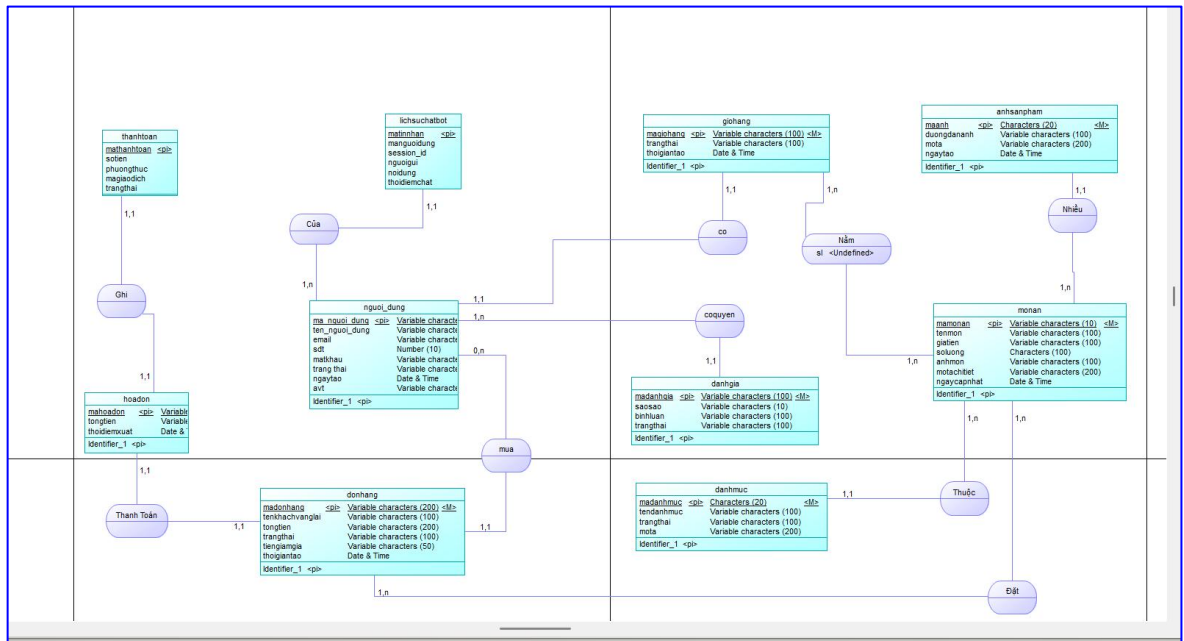
### **2.7. KẾT LUẬN CHƯƠNG**

Chương 2 đã trình bày tổng quan các công nghệ được sử dụng trong dự án, bao gồm frontend với HTML, CSS, Tailwind CSS, JavaScript ES6+ và GSAP Animation; backend với NodeJS, ExpressJS, MySQL, RESTful API, JWT và Bcrypt; cùng các công cụ hỗ trợ như VS Code, Postman và Git/GitHub. Sự kết hợp hài hòa giữa các công nghệ này tạo nên một hệ thống website hoàn chỉnh, có hiệu năng cao, đảm bảo bảo mật và thuận tiện cho việc bảo trì, mở rộng. Trên nền tảng đó, chương tiếp theo sẽ tập trung trình bày chi tiết quá trình phân tích và thiết kế hệ thống, làm cơ sở cho việc triển khai dự án trong thực tế.



## CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

### 3.1. Mô hình thực thể kết hợp ERD



Hình 3.1: Mô hình thực thể kết hợp ERD

### **3.1.1. Mô tả hệ thống**

#### ***3.1.1.1 Tổng quan hệ thống***

Website Nhà hàng Ẩm thực Phương Nam - Vĩnh Long là hệ thống web fullstack được xây dựng theo kiến trúc Client-Server, bao gồm:

#### **Phía Client (Frontend):**

Giao diện người dùng (Customer Interface): Trang chủ, thực đơn, đặt bàn, giỏ hàng, thanh toán

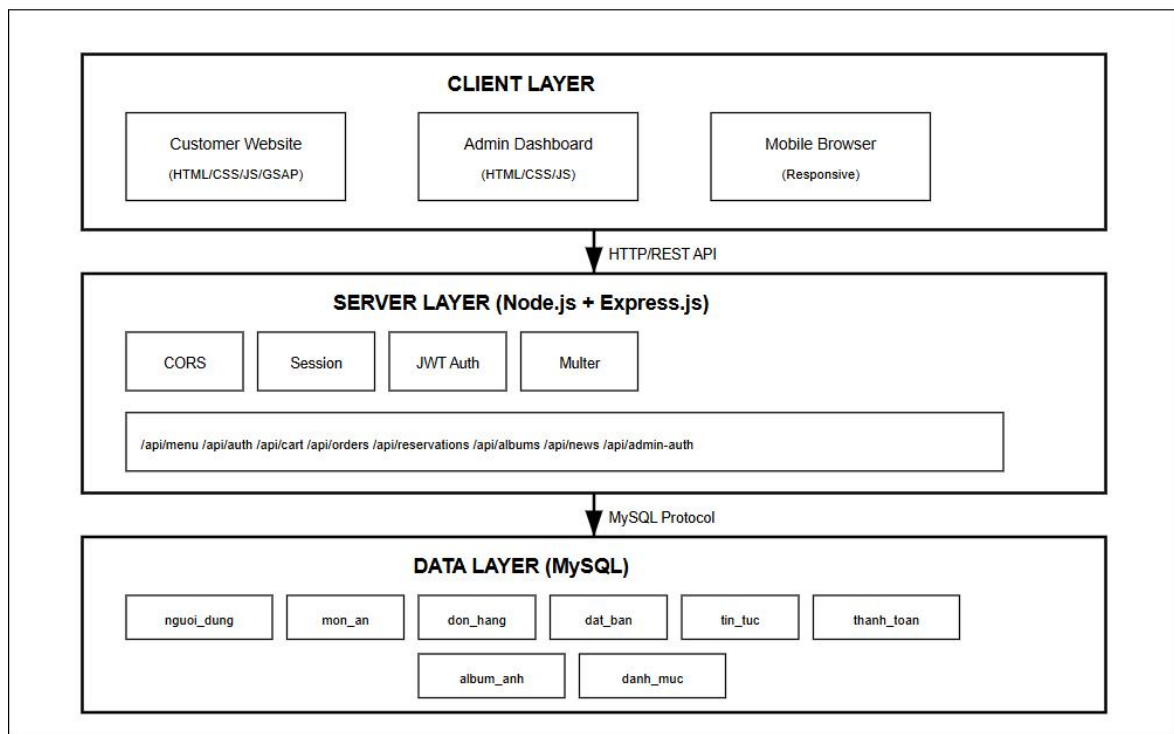
Giao diện quản trị (Admin Interface): Dashboard, quản lý món ăn, đơn hàng, khách hàng, tin tức

#### **Phía Server (Backend):**

RESTful API Server: Xử lý nghiệp vụ, xác thực, truy vấn dữ liệu

Database Server: Lưu trữ dữ liệu MySQL

### 3.1.2. Kiến trúc hệ thống



Hình 3.2: Kiến trúc hệ thống

#### 1. Client Layer (Lớp giao diện người dùng)

Đây là lớp tương tác trực tiếp với người sử dụng, bao gồm:

Customer Website:

Website dành cho khách hàng, được xây dựng bằng HTML, CSS, JavaScript và GSAP để tạo giao diện trực quan, sinh động.

Admin Dashboard:

Giao diện quản trị cho nhân viên và quản lý nhà hàng, cho phép quản lý thực đơn, đơn hàng, đặt bàn và nội dung hệ thống.

Mobile Browser (Responsive):

Giao diện tương thích trên thiết bị di động, đảm bảo trải nghiệm người dùng thống nhất trên nhiều kích thước màn hình.

Các thành phần ở lớp này giao tiếp với server thông qua HTTP/REST API.

### 2. Server Layer (Node.js + Express.js)

Lớp server là trung tâm xử lý nghiệp vụ và điều phối dữ liệu của toàn bộ hệ thống, được xây dựng trên nền tảng Node.js kết hợp với Express.js nhằm triển khai các RESTful API một cách linh hoạt và hiệu quả. Server tích hợp nhiều thành phần hỗ trợ quan trọng như CORS để cho phép frontend truy cập API từ các domain khác nhau, Session để quản lý phiên làm việc của người dùng, JWT Authentication nhằm xác thực và phân quyền người dùng thông qua cơ chế token, cùng với Multer phục vụ việc upload hình ảnh cho món ăn, album và tin tức. Thông qua hệ thống API, server cung cấp các endpoint đáp ứng đầy đủ các nghiệp vụ như quản lý thực đơn, xác thực người dùng, giỏ hàng, đơn hàng, đặt bàn, album ảnh và tin tức. Với vai trò trung gian, lớp server tiếp nhận các yêu cầu từ phía client, xử lý logic nghiệp vụ và truy xuất dữ liệu từ cơ sở dữ liệu trước khi trả kết quả về cho người dùng.

### 3. Data Layer (MySQL)

Đây là lớp lưu trữ dữ liệu của hệ thống, sử dụng hệ quản trị cơ sở dữ liệu MySQL.

Các bảng chính:

nguoi\_dung: lưu thông tin người dùng

mon\_an: lưu thông tin món ăn

don\_hang: quản lý đơn hàng

dat\_ban: quản lý đặt bàn

thanh\_toan: thông tin thanh toán

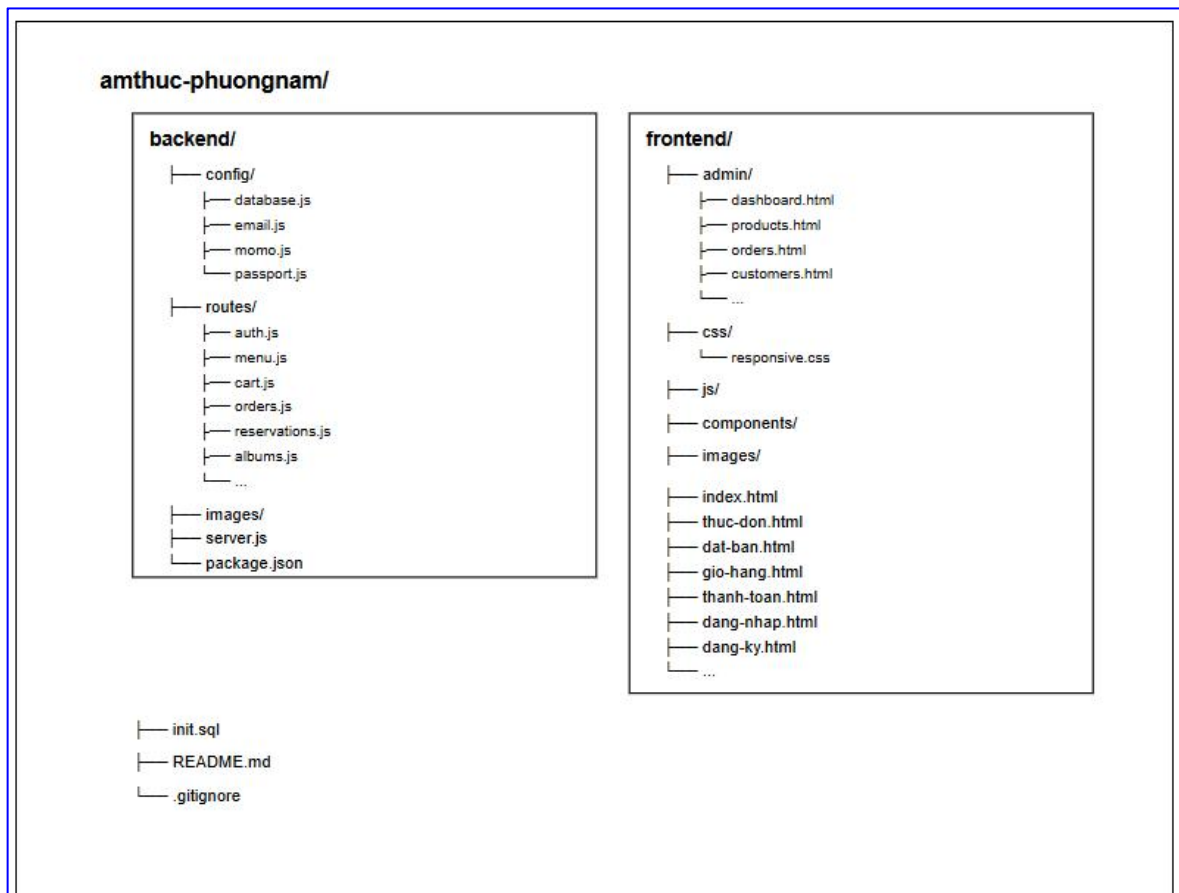
tin\_tuc: nội dung tin tức

album\_anh: hình ảnh món ăn và nhà hàng

danh\_muc: phân loại món ăn

Server giao tiếp với MySQL thông qua MySQL Protocol để thực hiện các thao tác CRUD.

### 3.1.3. Cấu trúc dự án



Hình 3.3: Cấu trúc dự án

Mô tả: Cấu trúc thư mục của dự án amthuc-phuongnam được tổ chức theo hướng tách biệt rõ ràng giữa frontend và backend, giúp hệ thống dễ phát triển, bảo trì và mở rộng. Thư mục backend đảm nhiệm toàn bộ xử lý nghiệp vụ và dữ liệu, trong đó thư mục config chứa các cấu hình quan trọng như kết nối cơ sở dữ liệu, gửi email, tích hợp thanh toán MoMo và xác thực người dùng. Thư mục routes định nghĩa các API theo từng nghiệp vụ cụ thể như xác thực, quản lý thực đơn, giỏ hàng, đơn hàng, đặt bàn và album ảnh. Ngoài ra, backend còn bao gồm thư mục images để lưu trữ hình ảnh upload, file server.js để khởi chạy ứng dụng và package.json để quản lý các thư viện phụ thuộc.

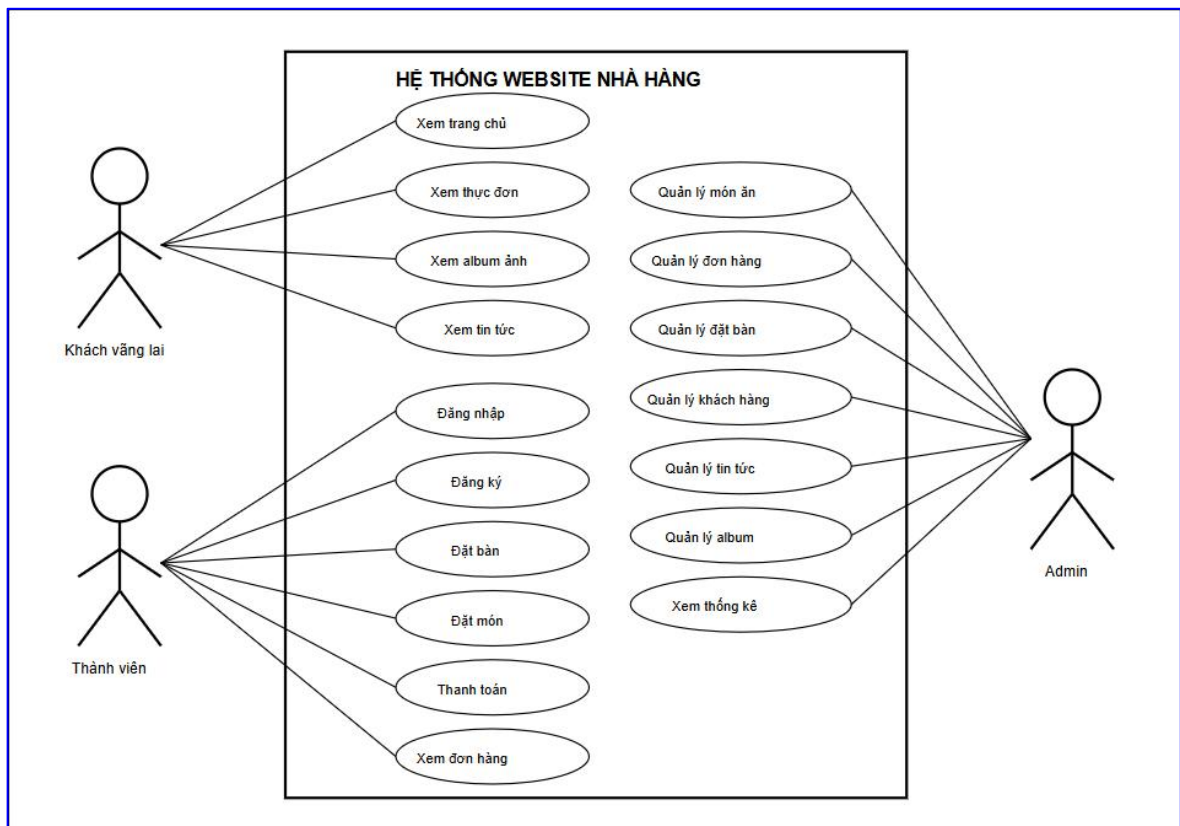
Thư mục frontend chịu trách nhiệm hiển thị giao diện người dùng, bao gồm khu vực admin dành cho quản trị với các trang dashboard, quản lý sản phẩm, đơn hàng và khách hàng, cùng các trang giao diện phía người dùng như trang chủ, thực đơn, đặt bàn, giỏ hàng, thanh toán, đăng nhập và đăng ký. Các tài nguyên giao diện

## XÂY DỰNG WEBSITE NHÀ HÀNG ÂM THỰC PHƯƠNG NAM VĨNH LONG

được tổ chức trong các thư mục `css`, `js`, `components` và `images` nhằm đảm bảo tính tái sử dụng và nhất quán. Ở cấp cao nhất của dự án còn có các file như `init.sql` để khởi tạo cơ sở dữ liệu, `README.md` mô tả dự án và `.gitignore` phục vụ quản lý mã nguồn. Cách tổ chức này phản ánh rõ kiến trúc client–server hiện đại, phù hợp cho một hệ thống website quản lý nhà hàng hoàn chỉnh.

## 3.2. XÁC ĐỊNH CÁC YÊU CẦU CHỨC NĂNG CỦA HỆ THỐNG

### 3.2.1. Sơ đồ Usecase



Hình 3.4: Sơ đồ UseCase

Mô tả: Sơ đồ mô tả ba nhóm người dùng chính: Khách vãng lai, Thành viên, và Admin, mỗi nhóm có quyền truy cập các chức năng khác nhau trong hệ thống.

Khách vãng lai có thể xem các nội dung công khai như: trang chủ, thực đơn, album ảnh và tin tức.

Thành viên (người dùng đã đăng ký) được sử dụng thêm các chức năng nâng cao như: đăng nhập, đặt bàn, đặt món, thanh toán và xem đơn hàng của mình.

Admin có quyền quản trị hệ thống, bao gồm: quản lý món ăn, đơn hàng, đặt bàn, khách hàng, tin tức, album và xem báo cáo thống kê.

Sơ đồ thể hiện rõ ràng phạm vi hệ thống và mối liên hệ giữa các actor với các chức năng tương ứng.

### 3.3. THIẾT KẾ XỬ LÝ HỆ THỐNG

#### 3.3.1. Thiết kế API RESTful

##### A. API Xác thực (Authentication)

POST /api/auth/send-verification – Gửi mã xác thực email.

POST /api/auth/verify-email – Xác thực mã và tạo tài khoản.

POST /api/auth/login – Đăng nhập lấy token.

POST /api/auth/forgot-password – Gửi yêu cầu đặt lại mật khẩu.

POST /api/auth/reset-password – Nhập mã + mật khẩu mới để reset.

GET /api/auth/me – Lấy thông tin người dùng đang đăng nhập.

PUT /api/auth/update – Cập nhật thông tin cá nhân.

POST /api/auth/change-password – Đổi mật khẩu.

##### B. API Món ăn (Menu)

GET /api/menu – Lấy danh sách món, hỗ trợ tìm kiếm/lọc/sắp xếp.

GET /api/menu/:id – Xem chi tiết món ăn.

GET /api/menu/category/:id – Món theo danh mục.

GET /api/menu/top-selling – Món bán chạy.

GET /api/menu/related/:id – Món liên quan.

POST /api/menu – Thêm món mới (Admin).

PUT /api/menu/:id – Cập nhật món (Admin).

DELETE /api/menu/:id – Xóa món (Admin).

PATCH /api/menu/:id/toggle-status – Ẩn/hiện món (Admin).

##### C. API Giỏ hàng (Cart)

GET /api/cart – Lấy giỏ hàng hiện tại.

POST /api/cart/add – Thêm món vào giỏ.



PUT /api/cart/update/:id – Thay đổi số lượng món.

DELETE /api/cart/remove/:id – Xóa một món.

DELETE /api/cart/clear – Xóa toàn bộ giỏ hàng.

#### **D. API Đơn hàng (Orders)**

POST /api/orders/create – Tạo đơn hàng mới.

GET /api/orders/my-orders – Lịch sử đơn hàng của người dùng.

GET /api/orders/:orderId – Xem chi tiết đơn hàng.

PUT /api/orders/:orderId/cancel – Hủy đơn.

GET /api/orders/all – Danh sách tất cả đơn (Admin).

PUT /api/orders/:orderId/status – Cập nhật trạng thái đơn (Admin).

GET /api/orders/stats – Thống kê doanh thu, số đơn (Admin).

#### **E. API Đặt bàn (Reservations)**

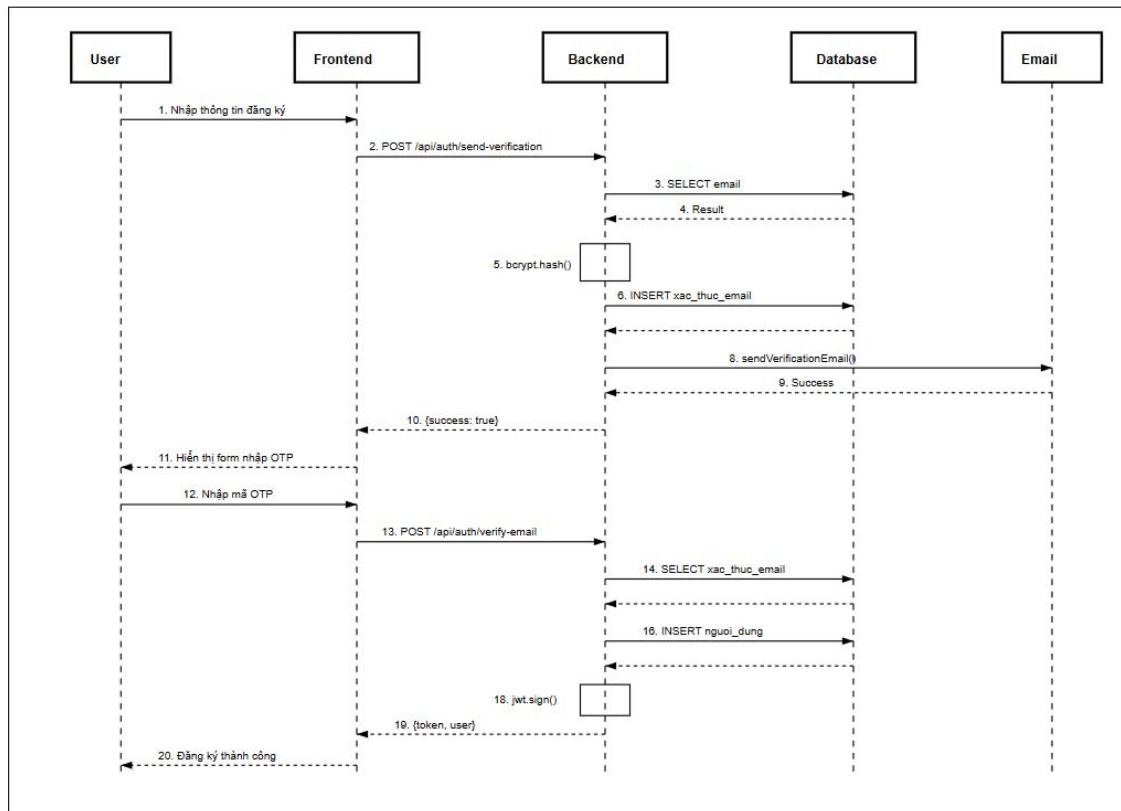
POST /api/reservations – Tạo yêu cầu đặt bàn.

GET /api/reservations – Danh sách đặt bàn (Admin).

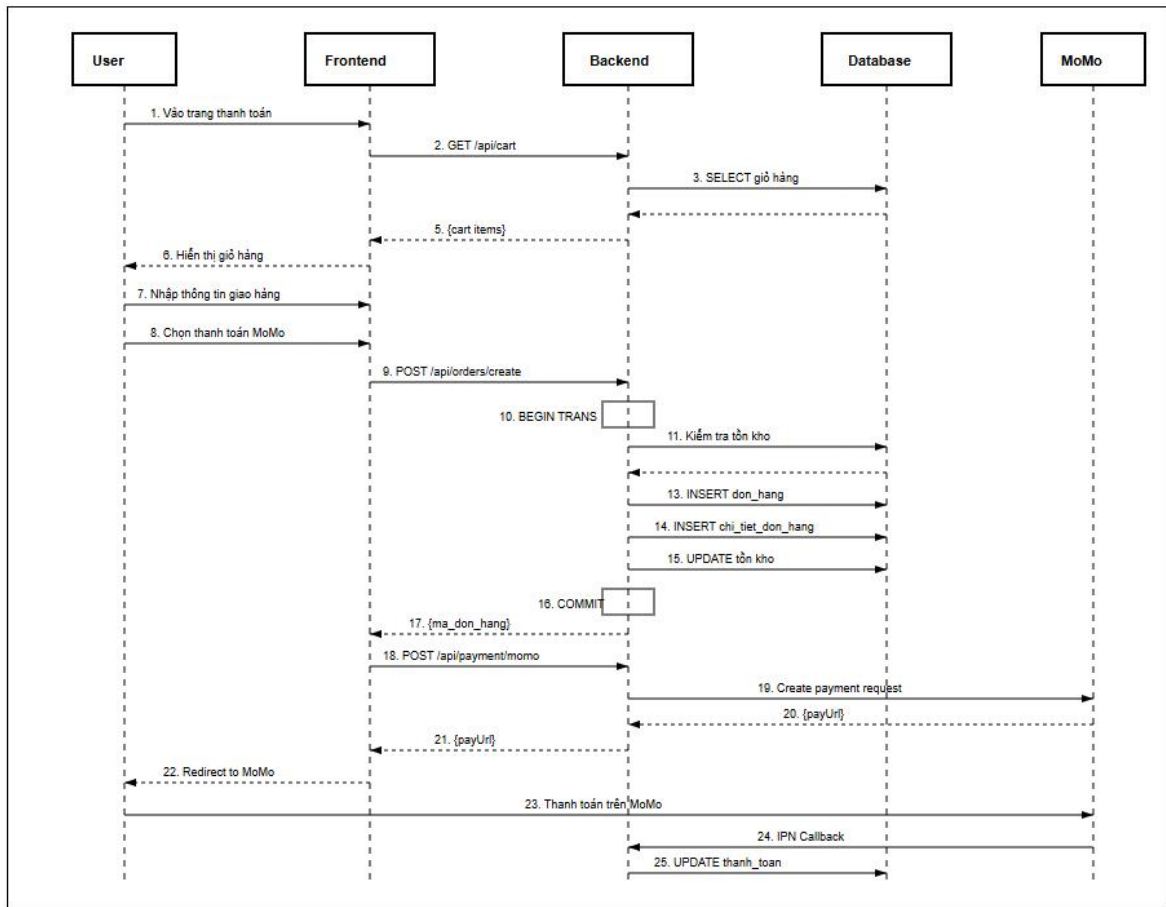
PUT /api/reservations/:id/status – Cập nhật trạng thái đặt bàn (Admin).

### 3.3.2. Sơ đồ tuần tự (Sequence Diagram)

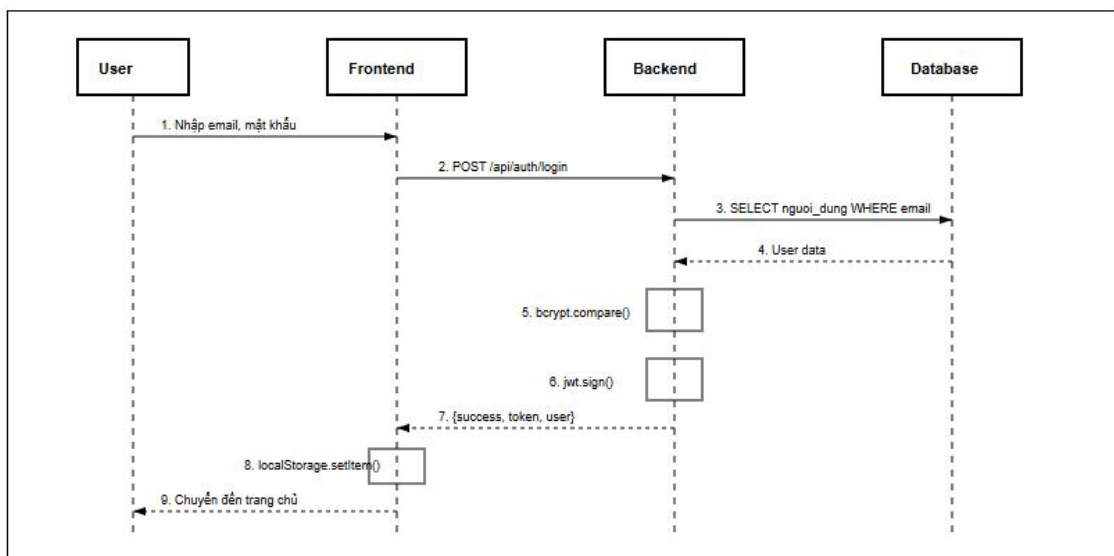
#### 1. Quy trình Đăng ký tài khoản



## 2. Quy trình Đặt hàng và Thanh toán



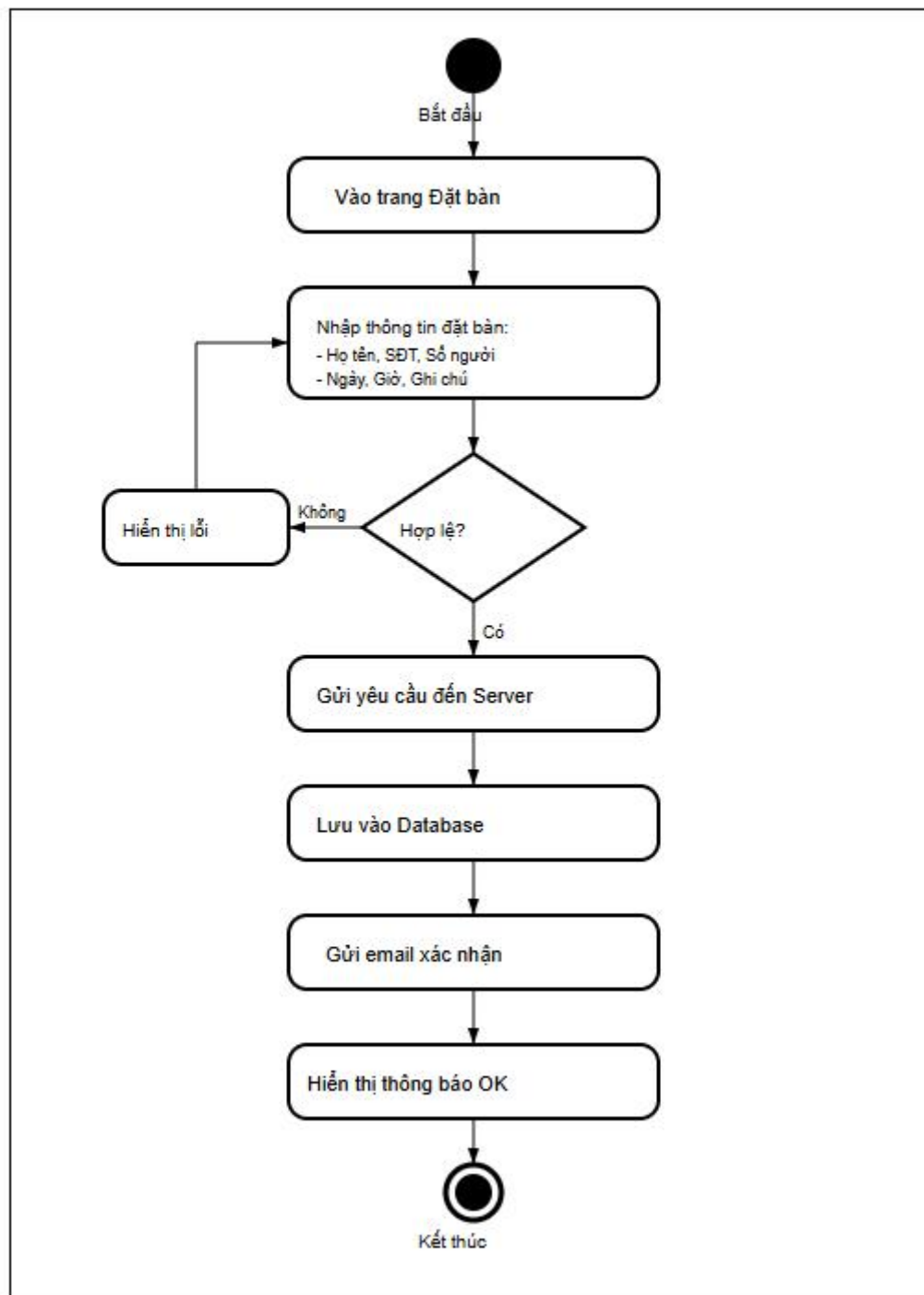
## 3. Quy trình Đăng nhập



Hình 3.5: Sơ đồ tuần tự

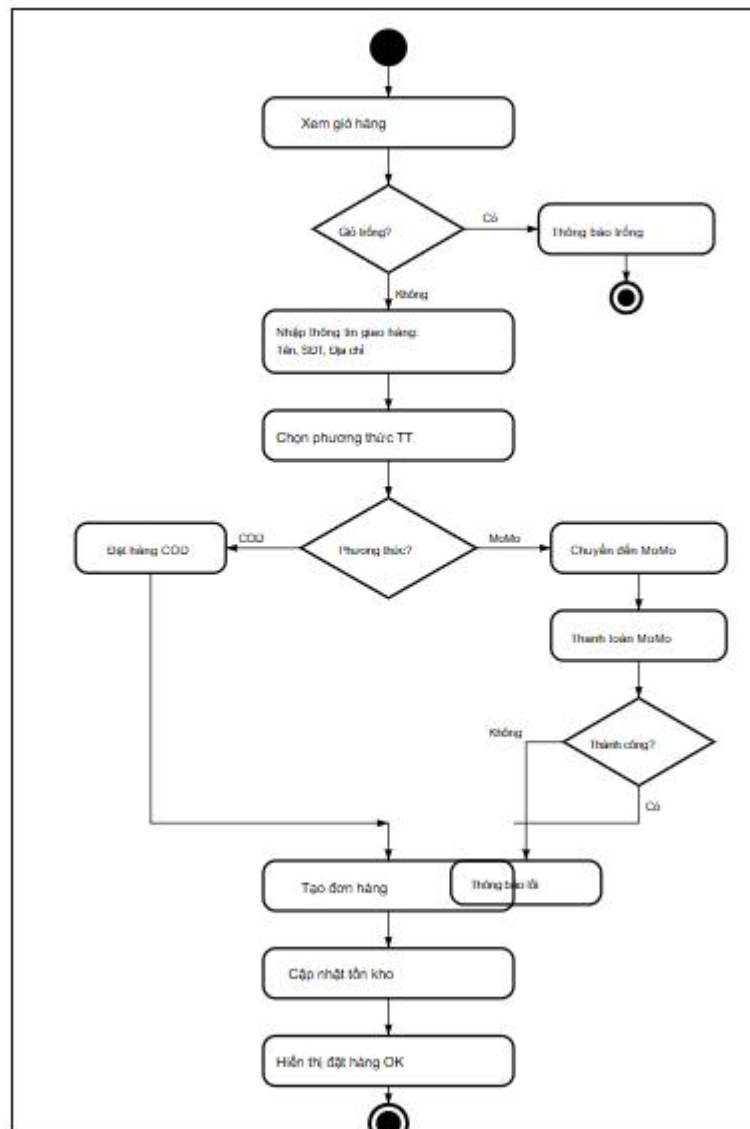
### 3.4. MÔ HÌNH HOẠT ĐỘNG CỦA HỆ THỐNG

#### 1. Quy trình Đặt bàn trực tuyến



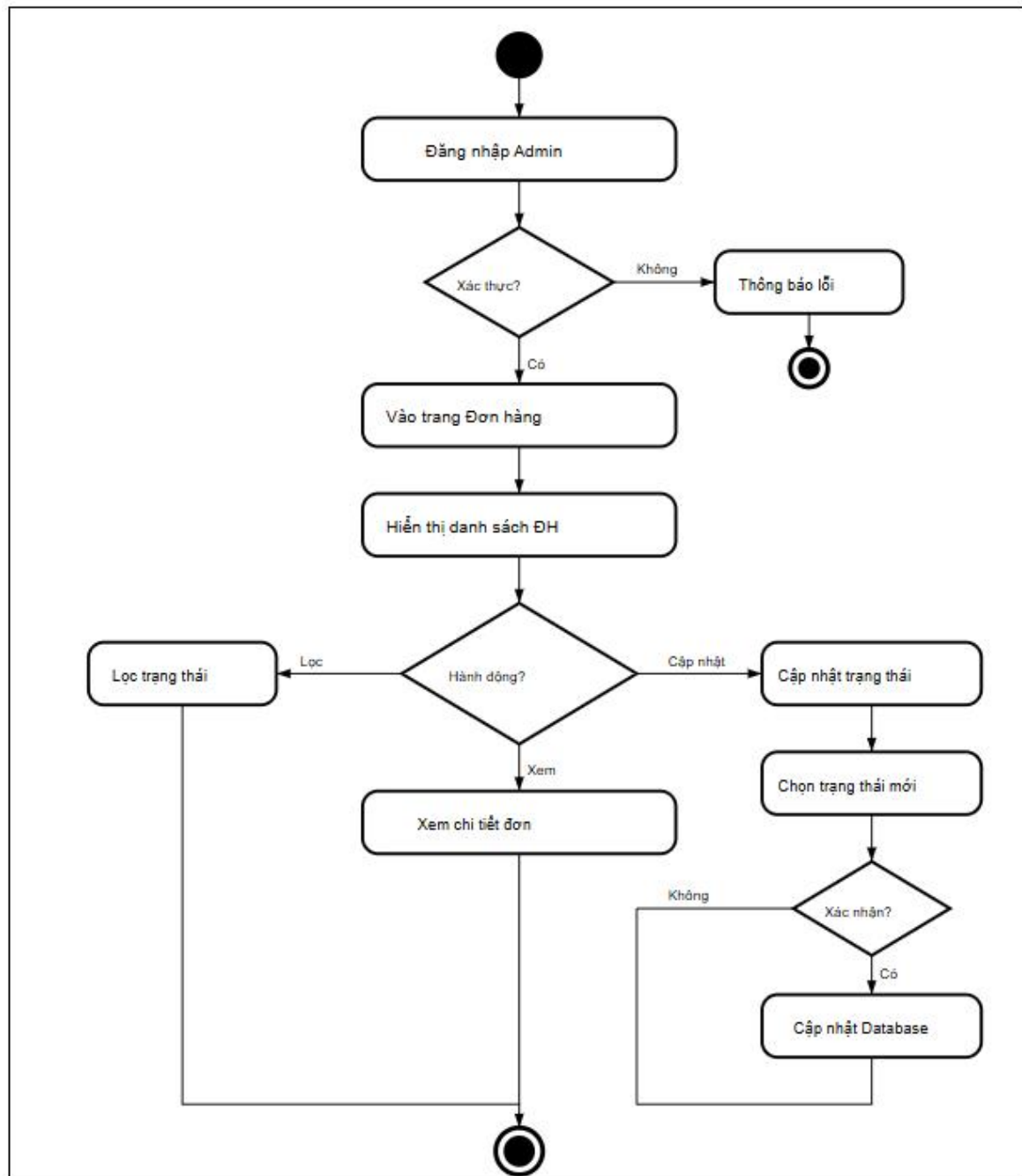
Hình 3.6: Quy trình đặt bàn trực tuyến

## 2. Quy trình Đặt hàng



Hình 3.7: Quy trình đặt hàng

### 3. Quy trình Quản lý đơn hàng (Admin)



Hình 3.8: Quy trình Quản lý đơn hàng

### **Kết luận Chương 3:**

Chương này đã trình bày một cách chi tiết quá trình hiện thực hóa hệ thống website Nhà hàng Âm thực Phương Nam, từ tổng quan kiến trúc đến các khía cạnh thiết kế và triển khai cốt lõi. Nội dung chương bao gồm mô tả tổng thể hệ thống với kiến trúc Client–Server và cấu trúc thư mục dự án rõ ràng, giúp đảm bảo tính tổ chức và khả năng bảo trì. Bên cạnh đó, các yêu cầu chức năng của hệ thống được xác định thông qua biểu đồ Use Case dành cho cả khách hàng và quản trị viên. Phần thiết kế dữ liệu tập trung vào sơ đồ ERD với hơn 15 bảng dữ liệu quan hệ, phản ánh đầy đủ các nghiệp vụ của nhà hàng. Ngoài ra, chương cũng trình bày thiết kế xử lý dựa trên API RESTful, cơ chế xử lý transaction và xác thực người dùng bằng JWT nhằm đảm bảo tính nhất quán và bảo mật. Cuối cùng, mô hình hoạt động của hệ thống được minh họa thông qua các sơ đồ tuần tự và sơ đồ hoạt động, giúp làm rõ luồng xử lý của các nghiệp vụ chính trong toàn bộ hệ thống.

## CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

### 4.1. Tổng quan kết quả

Sau quá trình phân tích, thiết kế và triển khai, hệ thống Website Âm thực Phương Nam đã được hoàn thành với đầy đủ các chức năng theo yêu cầu đặt ra. Hệ thống bao gồm hai phần chính:

Giao diện người dùng (Frontend): Cho phép khách hàng xem thực đơn, đặt món, đặt bàn, thanh toán và quản lý tài khoản.

Giao diện quản trị (Admin): Cho phép quản trị viên quản lý toàn bộ hoạt động của nhà hàng

### 4.2. Giao diện người dùng (Customer Interface)

#### 4.2.1. Trang chủ (index.html)

Mô tả:

Trang chủ là giao diện trung tâm của website, đóng vai trò giới thiệu tổng quan về nhà hàng và tạo ấn tượng ban đầu với khách hàng. Trang hiển thị các nội dung nổi bật như banner quảng cáo, thông tin giới thiệu và các món ăn tiêu biểu, giúp khách hàng nhanh chóng nắm bắt phong cách và chất lượng dịch vụ của nhà hàng.

Chức năng chính:

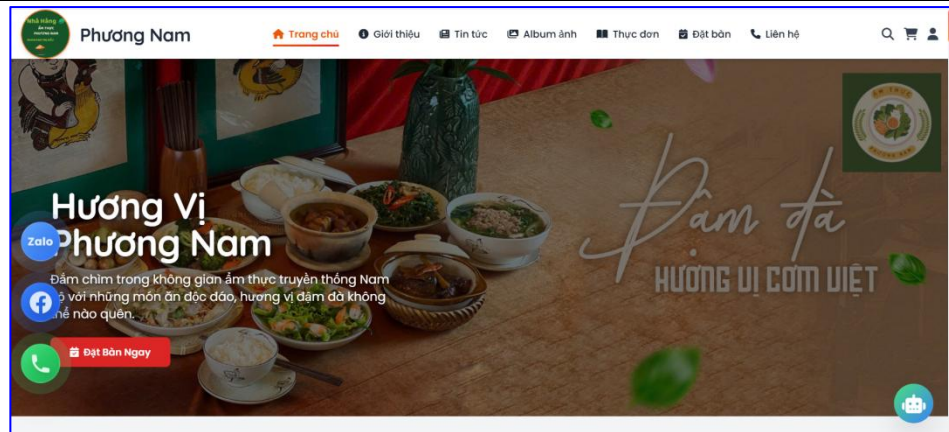
Trình chiếu banner quảng cáo tự động với hiệu ứng chuyển cảnh mượt mà (fade).

Hiển thị danh sách 4 món ăn bán chạy nhất, dữ liệu được tải động từ hệ thống API.

Cung cấp nội dung giới thiệu về nhà hàng và đội ngũ nhân viên, thể hiện uy tín và kinh nghiệm hoạt động.

Trình bày các cam kết về chất lượng món ăn, dịch vụ và sự hài lòng của khách hàng.

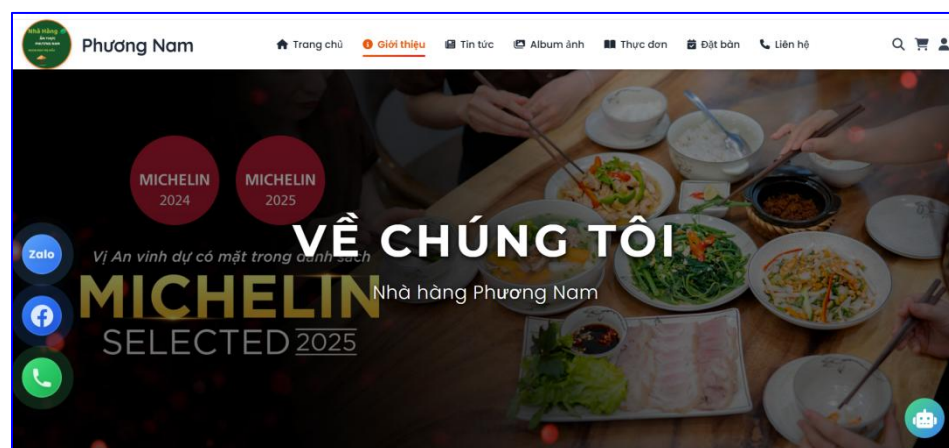




Hình 4.1: Giao diện trang chủ

### 4.2.2. Trang Thực đơn (thuc-don.html)

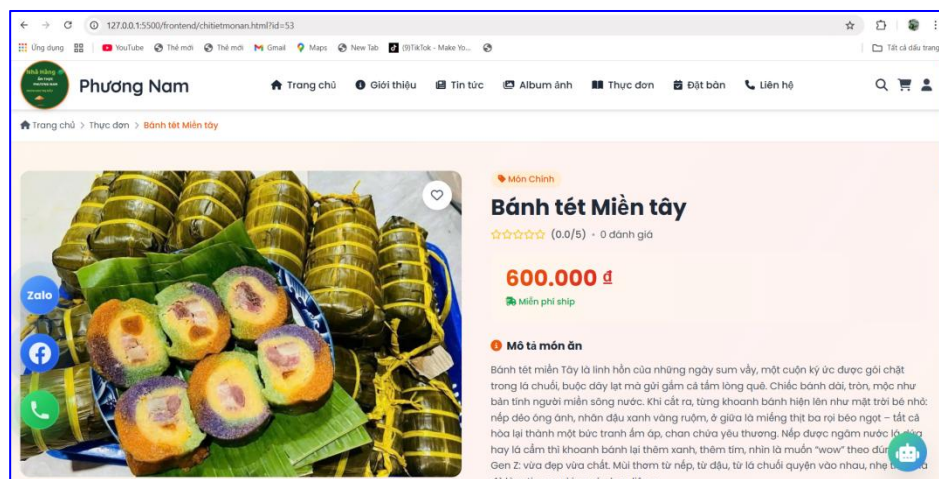
Trang danh sách món ăn hiển thị đầy đủ toàn bộ các món ăn hiện có của nhà hàng, giúp khách hàng dễ dàng tìm kiếm và lựa chọn món phù hợp với nhu cầu. Hệ thống cho phép người dùng tìm kiếm món ăn theo tên, lọc theo danh mục, khoảng giá và mức đánh giá sao, đồng thời hỗ trợ sắp xếp kết quả theo độ phổ biến, thời gian cập nhật hoặc giá tăng/giảm. Danh sách món ăn được phân trang rõ ràng để đảm bảo hiệu năng và trải nghiệm sử dụng mượt mà. Ngoài ra, khách hàng có thể thêm món ăn vào giỏ hàng trực tiếp ngay trên trang, rút ngắn quy trình đặt món và tăng tính tiện lợi khi sử dụng hệ thống.



Hình 4.2: Giao diện trang Giới thiệu

### 4.2.3. Trang Chi tiết món ăn (chitietmonan.html)

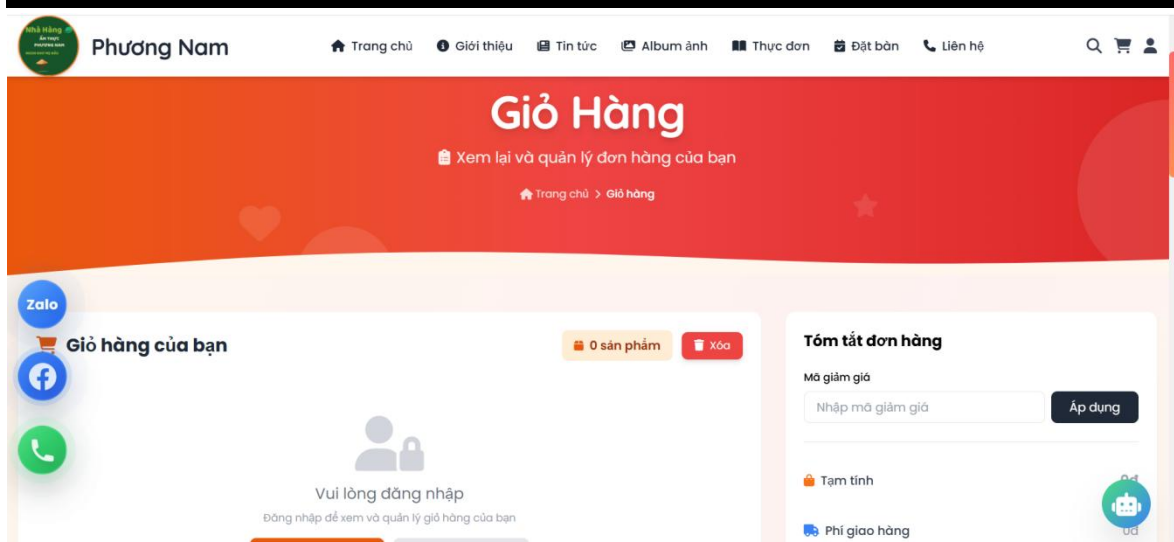
Trang chi tiết món ăn cung cấp đầy đủ thông tin về một món ăn cụ thể, giúp khách hàng có cái nhìn trực quan và chi tiết trước khi đưa ra quyết định đặt món. Trang hiển thị gallery hình ảnh món ăn chất lượng cao, kèm theo các thông tin quan trọng như tên món, giá bán, mô tả và danh mục. Bên cạnh đó, hệ thống cho phép xem các đánh giá và bình luận từ những khách hàng đã trải nghiệm, góp phần tăng độ tin cậy. Người dùng có thể lựa chọn số lượng và thêm món ăn vào giỏ hàng một cách thuận tiện, đồng thời hệ thống gợi ý các món ăn tương tự nhằm khuyến khích khám phá thêm và tăng giá trị đơn hàng.



Hình 4.3: Giao diện trang chi tiết món

### 4.2.4. Trang Giỏ hàng (gio-hang.html)

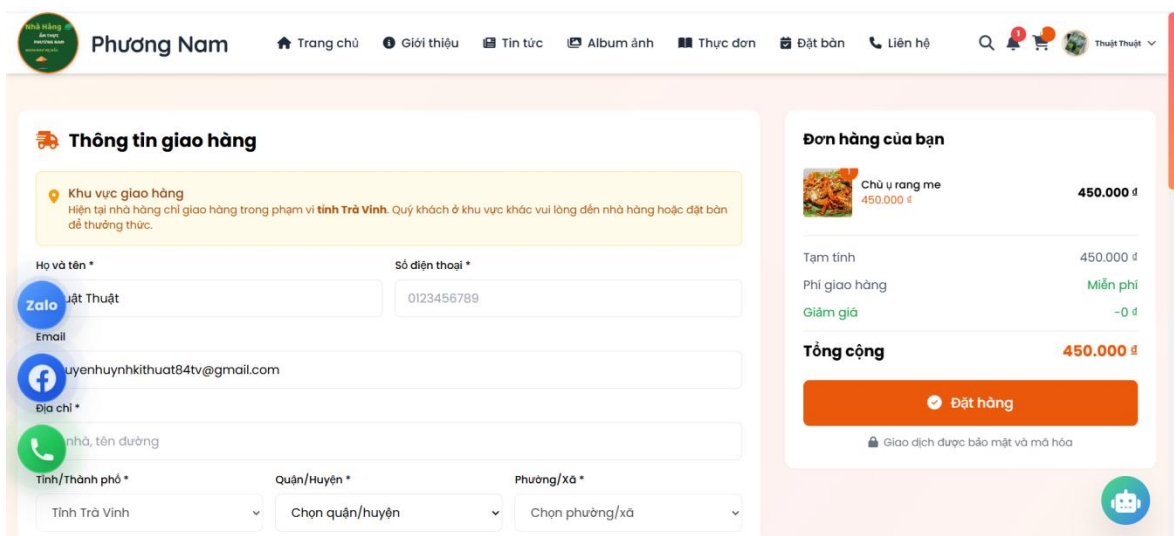
Trang giỏ hàng đóng vai trò quản lý toàn bộ các món ăn mà khách hàng đã lựa chọn trước khi tiến hành thanh toán. Tại đây, hệ thống hiển thị danh sách chi tiết các món đã thêm vào giỏ, cho phép người dùng dễ dàng tăng hoặc giảm số lượng, cũng như xóa món không còn nhu cầu. Giỏ hàng hỗ trợ áp dụng mã giảm giá và tự động tính toán tổng tiền theo thời gian thực, giúp khách hàng kiểm soát chi phí rõ ràng. Ngoài ra, hệ thống còn gợi ý các món ăn kết hợp phù hợp, góp phần nâng cao trải nghiệm người dùng và gia tăng giá trị đơn hàng.



Hình 4.4: Giao diện giỏ hàng

## 4.2.5. Trang Thanh toán (thanh-toan.html)

Trang thanh toán là bước cuối cùng trong quy trình mua hàng, nơi khách hàng hoàn tất đơn hàng bằng cách cung cấp thông tin giao hàng và lựa chọn phương thức thanh toán phù hợp. Hệ thống cho phép nhập đầy đủ thông tin người nhận, chọn hình thức thanh toán như tiền mặt khi nhận hàng (COD), chuyển khoản ngân hàng hoặc ví điện tử MoMo. Khách hàng có thể thêm ghi chú cho đơn hàng nhằm đáp ứng các yêu cầu đặc biệt trước khi xác nhận và gửi yêu cầu đặt hàng. Sau khi hoàn tất, đơn hàng được ghi nhận và chuyển sang giai đoạn xử lý tiếp theo.



Hình 4.5: Giao diện trang thanh toán

### 4.2.6. Trang Đặt bàn (dat-ban.html)


Trang đặt bàn cho phép khách hàng chủ động đặt chỗ trước tại nhà hàng một cách thuận tiện và chính xác. Người dùng nhập thông tin cá nhân, lựa chọn ngày, giờ đến, số lượng khách cũng như khu vực mong muốn trong nhà hàng. Hệ thống hỗ trợ ghi nhận các yêu cầu đặc biệt nhằm phục vụ tốt hơn trải nghiệm của khách hàng. Đồng thời, chức năng kiểm tra điều kiện đặt bàn trước tối thiểu 3 tiếng được áp dụng để đảm bảo nhà hàng có đủ thời gian chuẩn bị và sắp xếp phục vụ hợp lý.

The screenshot shows a web interface for a restaurant named 'Phương Nam'. The main section is titled 'Thông tin đặt bàn' (Reservation Information). It contains a blue box with a warning icon and text: 'Quy định đặt bàn: Quý khách vui lòng đặt bàn trước ít nhất 3 tiếng để nhà hàng chuẩn bị phục vụ tốt nhất.' Below this are input fields for 'Họ và tên \*' (Name), 'Số điện thoại \*' (Phone), 'Email', 'Ngày đặt \*' (Date), 'Giờ đặt \*' (Time), 'Số lượng khách \*' (Number of guests), and 'Khu vực' (Area). To the right, there is a 'Thông tin nhà hàng' (Restaurant Information) section with details like address, phone, and hours. At the bottom right, there is a 'Lưu ý khi đặt bàn' (Notes when booking) section with a reminder to book at least 3 hours in advance. The interface also features social media icons (Zalo, Facebook, WhatsApp) on the left and a search bar at the top.

Hình 4.6: Giao diện trang đặt bàn

### 4.2.7. Trang Đăng nhập / Đăng ký

Trang xác thực người dùng được thiết kế nhằm đảm bảo an toàn và cá nhân hóa trải nghiệm khi truy cập hệ thống. Người dùng có thể đăng nhập bằng email và mật khẩu hoặc thông qua tài khoản Google để thao tác nhanh chóng. Hệ thống hỗ trợ đăng ký tài khoản mới, chức năng quên mật khẩu để khôi phục quyền truy cập khi cần thiết, đồng thời thực hiện xác thực email nhằm đảm bảo tính chính xác của thông tin và nâng cao mức độ bảo mật cho tài khoản người dùng.




**Phương Nam**  
Vinh Long



## Đăng Nhập

Chào mừng bạn trở lại!


Email

 email@example.com


Mật khẩu


 Nhập mật khẩu 

☐ Ghi nhớ đăng nhập [Quên mật khẩu?](#)

 Đăng nhập

Hoặc đăng nhập với



 Đăng nhập với Google



**Phương Nam**  
Vinh Long

## Đăng Ký

Tạo tài khoản mới



Ảnh đại diện (tùy chọn)  
Tối đa 5MB

Họ và tên \*

Nguyễn Văn A

Email \*

email@example.com

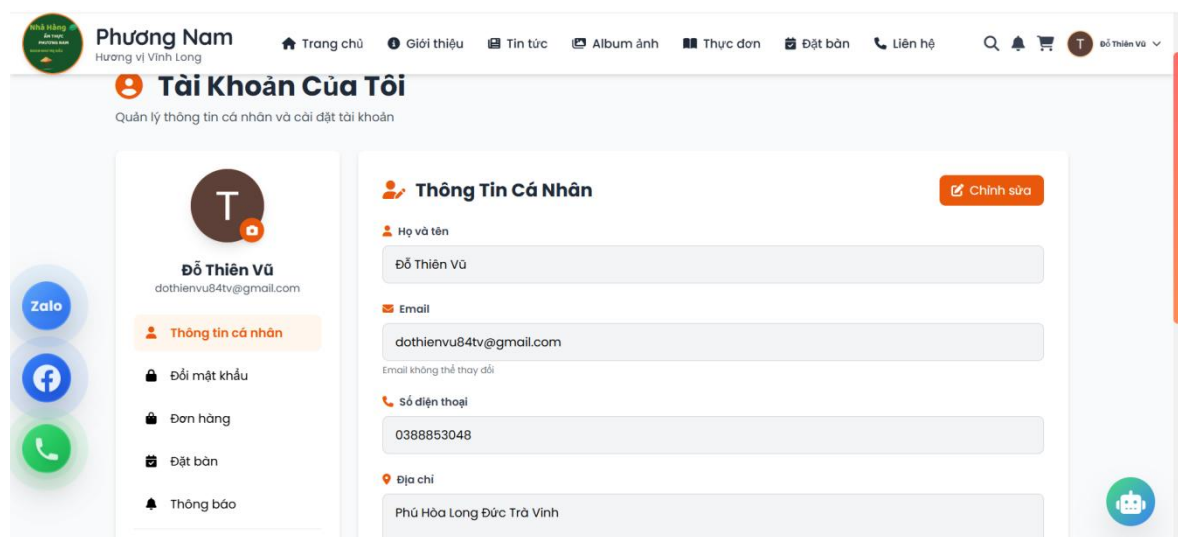
Số điện thoại

Hình 4.7: Giao diện đăng nhập/đăng ký



### 4.2.8. Trang Tài khoản (tai-khoan.html)

Trang hồ sơ người dùng cho phép quản lý toàn bộ thông tin cá nhân và theo dõi lịch sử hoạt động trên hệ thống. Tại đây, người dùng có thể cập nhật thông tin cá nhân, thay đổi mật khẩu để đảm bảo an toàn tài khoản, đồng thời xem lại lịch sử các đơn hàng đã đặt cũng như các lần đặt bàn trước đó. Ngoài ra, chức năng quản lý thông báo giúp người dùng theo dõi các cập nhật quan trọng như trạng thái đơn hàng, chương trình khuyến mãi và các thông tin liên quan khác từ nhà hàng.



Hình 4.8: Giao diện Tài khoản của tôi

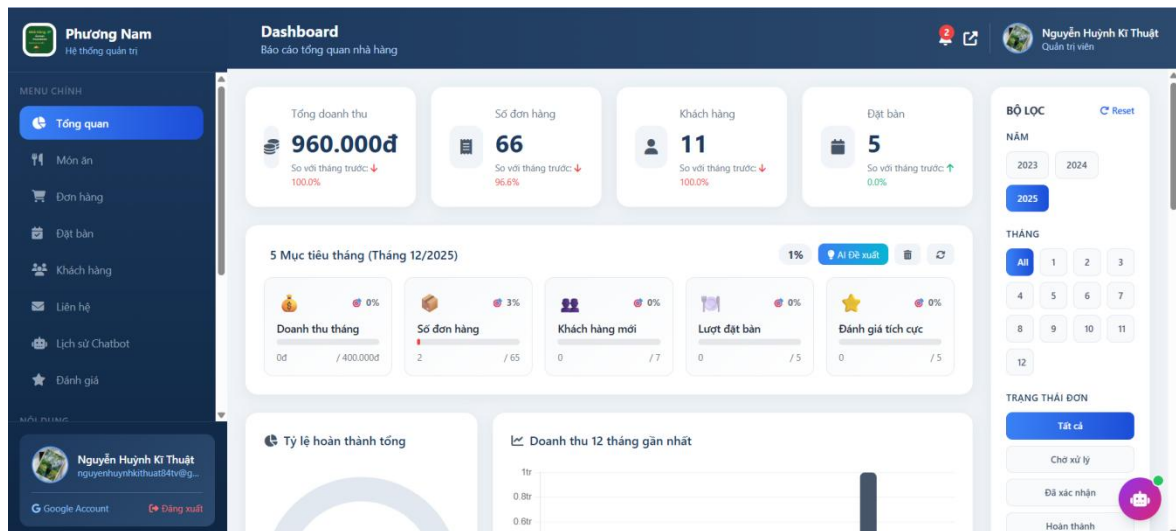
### 4.2.9. Các trang khác

Các trang chức năng hỗ trợ của website được xây dựng nhằm cung cấp đầy đủ thông tin và nâng cao trải nghiệm người dùng. Trang Tin tức hiển thị các bài viết liên quan đến nhà hàng, chương trình khuyến mãi và sự kiện, trong khi trang Chi tiết tin tức cung cấp nội dung đầy đủ của từng bài viết kèm theo chức năng bình luận. Trang Album ảnh trình bày bộ sưu tập hình ảnh về không gian nhà hàng, món ăn và các hoạt động nổi bật. Trang Giới thiệu cung cấp thông tin tổng quan về nhà hàng, tầm nhìn và giá trị cốt lõi. Trang Liên hệ hỗ trợ khách hàng gửi yêu cầu thông qua form liên hệ và xem vị trí nhà hàng trên bản đồ. Bên cạnh đó, trang Đơn hàng của tôi cho phép người dùng theo dõi trạng thái các đơn hàng đã đặt, giúp kiểm soát tiến trình xử lý một cách thuận tiện và minh bạch.

### 4.3. Giao diện quản trị (Admin Interface)

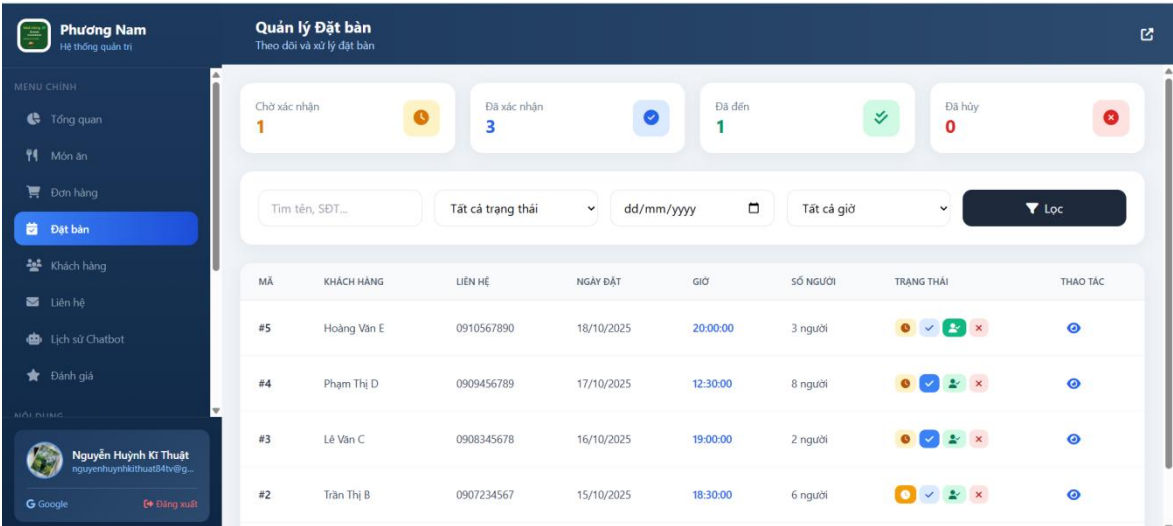
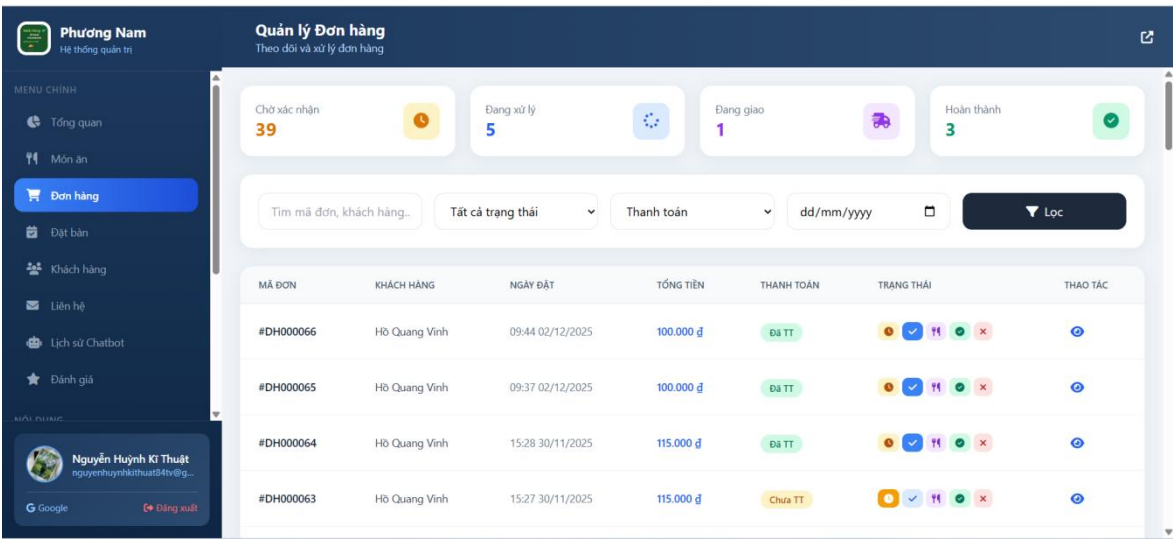
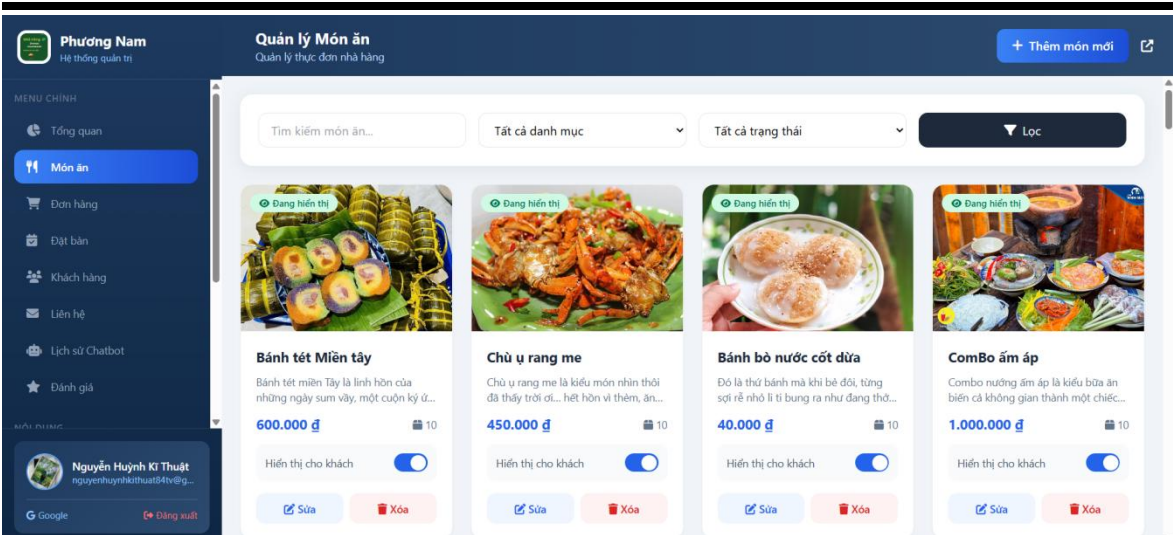
#### 4.3.1. Dashboard

Trang tổng quan (Dashboard) được thiết kế để cung cấp cái nhìn toàn diện về tình hình hoạt động của nhà hàng thông qua các chỉ số KPI quan trọng. Tại đây, hệ thống hiển thị thống kê doanh thu, số lượng đơn hàng mới, các lượt đặt bàn đang chờ xác nhận và số khách hàng mới trong từng khoảng thời gian. Các biểu đồ doanh thu giúp trực quan hóa xu hướng kinh doanh, trong khi danh sách món ăn bán chạy hỗ trợ nhà quản lý đưa ra quyết định điều chỉnh thực đơn kịp thời. Ngoài ra, khu vực thông báo hệ thống giúp cập nhật nhanh các sự kiện, cảnh báo và thông tin quan trọng liên quan đến hoạt động quản lý.



Hình 4.9: Giao diện Quản trị

# XÂY DỰNG WEBSITE NHÀ HÀNG ẨM THỰC PHƯƠNG NAM VĨNH LONG





# XÂY DỰNG WEBSITE NHÀ HÀNG ẨM THỰC PHƯƠNG NAM VĨNH LONG

Phương Nam

Hệ thống quản trị

MENU CHÍNH

Tổng quan

Món ăn

Đơn hàng

Đặt bàn

Khách hàng

Liên hệ

Lịch sử Chatbot

Đánh giá

Nguyễn Huỳnh Kỳ Thuật

nguyenhuyhkitruat84tv@g...

Google

Đăng xuất

Quản lý Khách hàng

Danh sách khách hàng đã đăng ký

Tổng khách hàng

11

Mới tháng này

0

Đang hoạt động

10

Đã khóa

1

Tìm tên, email, SĐT...

Tất cả trạng thái

Tất cả giới tính

Lọc

KHÁCH HÀNG	EMAIL	SỐ ĐIỆN THOẠI	GIỚI TÍNH	NGÀY ĐĂNG KÝ	TRẠNG THÁI	THAO TÁC
Thuyết Thuật #11	nguyenhuyhkitruat84tv@gmail.com	N/A	Khác	30/11/2025	Hoạt động	
Nhi Đình #10	yennhivk82@gmail.com	N/A	Khác	30/11/2025	Hoạt động	
Lê Thị Hồng Quyên #9	leq94328@gmail.com	0388853045	Nữ	5/11/2025	Hoạt động	
Đỗ Thiên Vũ #8	dothienvu84tv@gmail.com	0388853048	Nam	5/11/2025	Hoạt động	

Phương Nam

Hệ thống quản trị

MENU CHÍNH

Tổng quan

Món ăn

Đơn hàng

Đặt bàn

Khách hàng

Liên hệ

Lịch sử Chatbot

Đánh giá

Nguyễn Huỳnh Kỳ Thuật

nguyenhuyhkitruat84tv@g...

Google

Đăng xuất

Quản lý Liên hệ

Tin nhắn từ khách hàng

Tổng liên hệ

5

Chưa đọc

2

Đã đọc

2

Đã phản hồi

1

Tìm tên, email, nội dung...

Tất cả trạng thái

Tất cả chủ đề

Lọc

NGƯỜI GỬI	CHỦ ĐỀ	NỘI DUNG	NGÀY GỬI	TRẠNG THÁI	THAO TÁC
Nguyễn Văn A nguyenvana@gmail.com 0911111111	Hỏi về địa chỉ chi nhánh	Cho tôi hỏi nhà hàng có chi nhánh nào ở Quận 2 không...	30/10/2025	Đã phản hồi	
Trần Thị B tranthib@gmail.com 0922222222	Đặt tiệc sinh nhật	Tôi muốn đặt tiệc sinh nhật cho 20 người, có gói nào...	30/10/2025	Đã đọc	
Lê Văn C levanc@gmail.com 0933333333	Góp ý về dịch vụ	Hôm qua tôi có đến nhà hàng, dịch vụ rất tốt. Cảm ơn!	30/10/2025	Đã đọc	
Phạm Thị D					

Phương Nam

Hệ thống quản trị

MENU CHÍNH

Tổng quan

Món ăn

Đơn hàng

Đặt bàn

Khách hàng

Liên hệ

Lịch sử Chatbot

Đánh giá

Nguyễn Huỳnh Kỳ Thuật

nguyenhuyhkitruat84tv@g...

Google

Đăng xuất

Lịch sử Chatbot

Quản lý cuộc trò chuyện với Trà My

Tổng tin nhắn

148

Cuộc trò chuyện

25

User đăng nhập

7

Khách vắng lại

12

Tin nhắn theo ngày (7 ngày gần nhất)

Phân bố tin nhắn

Tim nội dung chat...

Tất cả người dùng

Tất cả

Công cụ cắt

Đã sao chép ảnh chụp màn hình vào bảng tạm

Tự động lưu vào thư mục ảnh chụp màn hình.

Đánh dấu và Chia sẻ

## XÂY DỰNG WEBSITE NHÀ HÀNG ẨM THỰC PHƯƠNG NAM VĨNH LONG



Hình 4.10: Các giao diện do Admin quản lý

#### **4.4. Kết luận chương**

Chương 4 đã trình bày đầy đủ kết quả đạt được của hệ thống Website Âm thực Phương Nam. Hệ thống đáp ứng tốt yêu cầu chức năng, hiệu năng, bảo mật và trải nghiệm người dùng, đồng thời có nhiều điểm vượt trội so với các website nhà hàng truyền thống.

## CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1. Kết luận

#### 5.1.1. Kết quả đạt được

Sau quá trình nghiên cứu, phân tích, thiết kế và triển khai, đồ án đã hoàn thành xây dựng Website Âm thực Phương Nam – một hệ thống quản lý nhà hàng trực tuyến hoàn chỉnh với các kết quả chính sau:

##### **Về mặt chức năng:**

Dự án đã xây dựng thành công một hệ thống website nhà hàng với đầy đủ các chức năng cốt lõi phục vụ khách hàng như xem thực đơn, đặt món, đặt bàn, thanh toán trực tuyến và quản lý tài khoản cá nhân. Song song đó, hệ thống quản trị dành cho Admin được phát triển với giao diện dashboard trực quan, cho phép quản lý toàn diện các hoạt động của nhà hàng từ thực đơn, đơn hàng, đặt bàn đến khách hàng. Điểm nổi bật của hệ thống là việc tích hợp chatbot AI sử dụng OpenAI API nhằm hỗ trợ khách hàng 24/7, kết hợp với hệ thống gợi ý món ăn thông minh dựa trên Machine Learning. Ngoài ra, hệ thống còn hỗ trợ thanh toán điện tử thông qua MoMo Payment Gateway và đăng nhập bằng Google OAuth 2.0, góp phần nâng cao trải nghiệm người dùng.

##### **Về mặt kỹ thuật:**

Dự án áp dụng kiến trúc Client – Server kết hợp với RESTful API, đảm bảo tính tách biệt, linh hoạt và dễ mở rộng. Các công nghệ hiện đại như Node.js, Express.js, MySQL và Tailwind CSS được sử dụng xuyên suốt quá trình phát triển. Cơ sở dữ liệu được thiết kế chuẩn hóa với tổng cộng 26 bảng, đáp ứng đầy đủ yêu cầu nghiệp vụ. Hệ thống xây dựng hơn 20 API endpoint phục vụ các chức năng chính, đồng thời đảm bảo an toàn bảo mật thông qua cơ chế xác thực JWT. Giao diện được thiết kế responsive, hoạt động ổn định trên nhiều loại thiết bị khác nhau.

### Về mặt giao diện:

Website được thiết kế theo phong cách hiện đại, thân thiện và dễ sử dụng. Việc sử dụng Tailwind CSS giúp giao diện đồng nhất, dễ mở rộng và dễ bảo trì. Hệ thống hỗ trợ đầy đủ các kích thước màn hình như Desktop, Tablet và Mobile, đồng thời được tối ưu hiệu năng với thời gian tải trang trung bình dưới 2 giây, mang lại trải nghiệm mượt mà cho người dùng.

#### 5.1.2. Đóng góp mới

So với các website nhà hàng thông thường, đồ án có những đóng góp mới nổi bật như sau:

Tích hợp AI Chatbot: Ứng dụng trí tuệ nhân tạo thông qua OpenAI API để tư vấn và hỗ trợ khách hàng tự động, giúp giảm tải cho nhân viên và nâng cao trải nghiệm người dùng.

Hệ thống gợi ý thông minh: Áp dụng các thuật toán Machine Learning nhằm gợi ý món ăn phù hợp dựa trên lịch sử mua hàng và hành vi người dùng.

Thanh toán điện tử tích hợp: Kết nối trực tiếp với MoMo Payment Gateway giúp quá trình thanh toán diễn ra nhanh chóng và an toàn.

Hệ thống thông báo realtime: Cung cấp thông báo tức thì cho quản trị viên khi có đơn hàng hoặc yêu cầu đặt bàn mới, góp phần nâng cao hiệu quả xử lý.

Quản lý nội dung đa phương tiện: Xây dựng hệ thống tin tức có hỗ trợ bình luận và các dạng phản ứng như like, love, wow tương tự mạng xã hội.

#### 5.1.3. Ý nghĩa thực tiễn

Đối với nhà hàng:

Hệ thống giúp số hóa quy trình kinh doanh, quản lý hiệu quả đơn hàng, đặt bàn và thông tin khách hàng, đồng thời mở rộng kênh bán hàng trực tuyến nhằm tăng doanh thu.

Đối với khách hàng:

Mang lại trải nghiệm đặt món và đặt bàn thuận tiện mọi lúc, mọi nơi; được hỗ trợ liên tục bởi chatbot AI và có nhiều lựa chọn thanh toán an toàn, linh hoạt.

Đối với ngành F&B:

Đồ án cung cấp một mô hình tham khảo có giá trị cho quá trình chuyển đổi số trong lĩnh vực nhà hàng và ẩm thực.

### 5.2. Hạn chế

Bên cạnh những kết quả đạt được, đồ án vẫn còn một số điểm cần cải thiện. Hiện tại hệ thống mới dừng ở website responsive, chưa có ứng dụng mobile native. Hệ thống thanh toán mới chỉ tích hợp MoMo, chưa hỗ trợ các cổng khác như VNPAY, ZaloPay hay thẻ quốc tế. Chưa xây dựng chương trình khách hàng thân thiết và hệ thống tích điểm. Chatbot AI còn phụ thuộc vào OpenAI API và chưa được huấn luyện chuyên sâu cho lĩnh vực nhà hàng. Ngoài ra, website chưa được tối ưu đầy đủ cho SEO và chưa tích hợp hệ thống phân tích hành vi người dùng (analytics).

### 5.3. Hướng phát triển

#### 5.3.1. Phát triển ngắn hạn (3–6 tháng)

Tích hợp thêm các cổng thanh toán như VNPAY, ZaloPay, Stripe hoặc PayPal

Tối ưu SEO thông qua cải thiện cấu trúc URL, meta tags, sitemap và tốc độ tải trang

Tích hợp Google Analytics để theo dõi hành vi người dùng và đo lường hiệu quả marketing

Cải thiện chatbot AI bằng cách tinh chỉnh mô hình cho domain nhà hàng và bổ sung khả năng đặt món, đặt bàn trực tiếp qua chat

#### 5.3.2. Phát triển trung hạn (6–12 tháng)

Phát triển ứng dụng mobile trên nền tảng iOS và Android bằng React Native hoặc Flutter

Xây dựng chương trình khách hàng thân thiết với hệ thống tích điểm và phân hạng thành viên

Mở rộng hệ thống gợi ý món ăn với các phương pháp collaborative filtering và content-based filtering

Bổ sung tính năng đánh giá và review có hình ảnh, video kèm cơ chế xác thực

### **5.3.3. Phát triển dài hạn (1–2 năm)**

Mở rộng hệ thống thành nền tảng đa nhà hàng với kiến trúc multi-tenant

Tích hợp các giải pháp AI nâng cao như dự đoán nhu cầu, định giá động và phân tích cảm xúc

Phát triển hệ thống quản lý chuỗi cung ứng và tồn kho thông minh

Tích hợp IoT để kết nối với hệ thống POS, quản lý bàn và giám sát chất lượng thực phẩm

## **5.4. Lời kết**

Đồ án Website Âm thực Phương Nam đã hoàn thành các mục tiêu đề ra, xây dựng được một hệ thống quản lý nhà hàng trực tuyến hoàn chỉnh, tích hợp nhiều công nghệ hiện đại như AI Chat

## **TÀI LIỆU THAM KHẢO**

- [1] Phạm Hữu Khang (2020), Giáo trình Phát triển ứng dụng Web, NXB Đại học Quốc gia TP.HCM, 2020.
- [2] Nguyễn Văn Vy, Phân tích và thiết kế hệ thống thông tin, NXB Giáo dục Việt Nam, 2019.
- [3] Lê Văn Vinh, JavaScript từ cơ bản đến nâng cao, NXB Thông tin và Truyền thông, 2022.
- [4] Đặng Văn Đức, Cơ sở dữ liệu – Lý thuyết và thực hành, NXB Khoa học và Kỹ thuật, 2021.
- [5] Trần Đình Quế, Nhập môn Công nghệ phần mềm, NXB Giáo dục Việt Nam, 2020.