**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: THIẾT KẾ, LẬP TRÌNH BACK-END**

**ĐỀ TÀI: PHÁT TRIỂN WEBSITE CHO CỬA HÀNG BÁN ĐỒ ĂN NHANH**

**Giáo viên hướng dẫn: ThS. Phạm Văn Tiệp**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sinh viên** | **Họ và tên** | **Lớp** |
| 1 | 1771020319 | Đỗ Khánh Hùng | CNTT 17-02 |

**Hà Nội, năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: THIẾT KẾ, LẬP TRÌNH BACK-END**

**ĐỀ TÀI: PHÁT TRIỂN WEBSITE CHO CỬA HÀNG BÁN ĐỒ ĂN NHANH**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | **1771020319** | **Đỗ Khánh Hùng** | **24/11/2005** |  |  |

### 

CÁN BỘ CHẤM THI

**Hà Nội, năm 2025**

**LỜI NÓI ĐẦU**

Đề tài "Phát triển website cho cửa hàng bán đồ ăn nhanh" được thực hiện trong khuôn khổ môn học Lập trình Back-end  là cơ hội để tôi áp dụng những kiến thức lý thuyết đã học vào thực tế, đồng thời rèn luyện kỹ năng phân tích, thiết kế và phát triển phần mềm. Trong quá trình thực hiện bài tập lớn, tôi đã gặp không ít khó khăn và thách thức, từ việc lựa chọn công nghệ phù hợp, thiết kế kiến trúc hệ thống, đến việc triển khai các chức năng và kiểm thử chất lượng. Tuy nhiên, với sự hướng dẫn tận tình và chỉ bảo chi tiết của Thầy Giảng Viên Ths. Phạm Văn Tiệp, tôi đã có thể vượt qua những khó khăn này và hoàn thành bài tập lớn một cách tốt nhất. Tôi xin chân thành cảm ơn Thầy Ths. Phạm Văn Tiệp đã: Cung cấp những định hướng quan trọng trong việc lựa chọn công nghệ và phương pháp phát triển. Hướng dẫn chi tiết về kiến trúc hệ thống và thiết kế cơ sở dữ liệu. Đánh giá và phản hồi kịp thời để tôi có thể hoàn thiện bài tập lớn

Bài tập lớn này không chỉ giúp tôi củng cố kiến thức về ASP.NET Core MVC, Entity Framework Core, SQL Server và các công nghệ web hiện đại, mà còn rèn luyện kỹ năng làm việc độc lập, tư duy logic và khả năng giải quyết vấn đề. Những kinh nghiệm quý báu này sẽ là nền tảng vững chắc cho sự nghiệp phát triển phần mềm của tôi trong tương lai.

Mặc dù đã cố gắng hết sức, bài tập lớn vẫn có thể còn những thiếu sót và hạn chế. Tôi rất mong nhận được sự góp ý và đánh giá từ Thầy để có thể cải thiện và hoàn thiện hơn nữa.

**MỤC LỤC**

[**LỜI NÓI ĐẦU** 3](#_Toc202771346)

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 9](#_Toc202771347)

[**1.1.** **Giới thiệu về đề tài** 9](#_Toc202771348)

[**1.2. Mục tiêu của đề tài** 9](#_Toc202771349)

[**1.3. Phạm vi của đề tài** 12](#_Toc202771350)

[**1.4. Phương pháp nghiên cứu** 14](#_Toc202771351)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 16](#_Toc202771352)

[**2.1. Tổng quan về C#** 16](#_Toc202771353)

[**2.2. ASP.NET Core MVC** 18](#_Toc202771354)

[**2.3. SQL Server** 21](#_Toc202771355)

[**2.4. Entity Framework Core** 23](#_Toc202771356)

[**2.5. LINQ** 24](#_Toc202771357)

[CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG 27](#_Toc202771358)

[**3.1. Phân tích đặc tả yêu cầu** 27](#_Toc202771359)

[3.1.1. Yêu cầu chức năng 27](#_Toc202771360)

[a) Quản lý thông tin cá nhân 27](#_Toc202771361)

[b) Xem thực đơn, tìm kiếm món ăn 27](#_Toc202771362)

[c) Xem chi tiết món ăn, đánh giá món 27](#_Toc202771363)

[d) Giỏ hàng và đặt hàng 27](#_Toc202771364)

[e) Xem lịch sử đơn hàng 27](#_Toc202771365)

[f) Quản trị hệ thống (Admin) 27](#_Toc202771366)

[g) Các trang thông tin 28](#_Toc202771367)

[3.1.2. Yêu cầu phi chức năng 28](#_Toc202771368)

[a) Bảo mật 28](#_Toc202771369)

[b) Hiệu năng 28](#_Toc202771370)

[c) Giao diện 28](#_Toc202771371)

[d) Tính ổn định và bảo trì 28](#_Toc202771372)

[e) Lưu trữ và sao lưu dữ liệu 28](#_Toc202771373)

[**3.2. Thiết kế hệ thống** 29](#_Toc202771374)

[3.2.1. Usecase Diagram 29](#_Toc202771375)

[a) Biểu đồ Usecase tổng thể 29](#_Toc202771376)

[b) Biểu đồ Usecase tác nhân Khách hàng 30](#_Toc202771377)

[c) Biểu đồ Usecase tác nhân Quản trị viên 30](#_Toc202771378)

[3.2.2. Class Diagram 31](#_Toc202771379)

[3.2.3. Activity Diagram 31](#_Toc202771380)

[a) Quản lý tài khoản 32](#_Toc202771381)

[b) Quản lý đơn hàng 33](#_Toc202771382)

[c) Tìm kiếm thông tin 33](#_Toc202771383)

[d) Đặt hàng 34](#_Toc202771384)

[e) Liên hệ 35](#_Toc202771385)

[f) Quản trị website 36](#_Toc202771386)

[3.2.4. Sequence Diagram 36](#_Toc202771387)

[a) Quản lý tài khoản 37](#_Toc202771388)

[b) Quản lý đơn hàng 37](#_Toc202771389)

[c) Tìm kiếm thông tin 38](#_Toc202771390)

[d) Đặt hàng 38](#_Toc202771391)

[e) Liên hệ 39](#_Toc202771392)

[f) Quản trị website 39](#_Toc202771393)

[**3.3. Thiết kế cơ sở dữ liệu** 39](#_Toc202771394)

[**3.4. Thiết kế giao diện** 40](#_Toc202771395)

[3.4.1. Trang chủ (Home/Index) 40](#_Toc202771396)

[a) Header 40](#_Toc202771397)

[b) Banner chính 40](#_Toc202771398)

[c) Phần danh mục 41](#_Toc202771399)

[d) Phần món ăn nổi bật 41](#_Toc202771400)

[e) Footer 41](#_Toc202771401)

[3.4.2. Trang Menu (Home/Menu) 41](#_Toc202771402)

[a) Header giống trang chủThanh tìm kiếm và lọc 41](#_Toc202771403)

[b) Hiển thị món ăn 41](#_Toc202771404)

[c) Phân trang: 42](#_Toc202771405)

[3.4.3. Trang chi tiết món ăn (Home/FoodDetails) 42](#_Toc202771406)

[a) Breadcrumb 42](#_Toc202771407)

[b) Layout 2 cột:Cột trái (60%) 42](#_Toc202771408)

[c) Cột phải (40%) 42](#_Toc202771409)

[d) Phần đánh giá 42](#_Toc202771410)

[3.4.4. Trang giỏ hàng (Cart/Index) 43](#_Toc202771411)

[a) Tiêu đề "Giỏ hàng của bạn"Danh sách món ăn 43](#_Toc202771412)

[b) Tổng kết 43](#_Toc202771413)

[c) Giỏ hàng trống 43](#_Toc202771414)

[3.4.5. Trang thanh toán (Cart/Checkout) 44](#_Toc202771415)

[a) Form thanh toán 2 cột:Cột trái - Thông tin giao hàng 44](#_Toc202771416)

[b) Cột phải - Tóm tắt đơn hàng 44](#_Toc202771417)

[a) Sidebar bên trái 44](#_Toc202771418)

[b) Main content 45](#_Toc202771419)

[a) Header với nút "Thêm món ăn mới"Bảng danh sách 45](#_Toc202771420)

[b) Form thêm/sửa món ăn 45](#_Toc202771421)

[**3.5. Kiểm thử** 46](#_Toc202771422)

[3.5.1. Kiểm thử chức năng 46](#_Toc202771423)

[a) Hiển thị danh sách món ăn 46](#_Toc202771424)

[b) Tìm kiếm món ăn 46](#_Toc202771425)

[**c**) Thêm món ăn vào giỏ hàng 47](#_Toc202771426)

[d)Cập nhật số lượng 47](#_Toc202771427)

[**e)** Tạo đơn hàng mới 47](#_Toc202771428)

[f) Đăng nhập admin 48](#_Toc202771429)

[g) Thêm món ăn mới 48](#_Toc202771430)

[3.5.2. Kiểm thử giao diện 48](#_Toc202771431)

[3.5.3. Kiểm thử hiệu năng 53](#_Toc202771432)

[3.5.4. Kiểm thử bảo mật 53](#_Toc202771433)

[a) Đăng nhập với thông tin hợp lệ 54](#_Toc202771434)

[b) Đăng nhập với thông tin không hợp lệ 54](#_Toc202771435)

[**KẾT LUẬN** 55](#_Toc202771436)

[**1. Kết quả đạt được** 55](#_Toc202771437)

[**2. Nhược điểm** 55](#_Toc202771438)

[**3. Hướng phát triển** 55](#_Toc202771439)

[TÀI LIỆU THAM KHẢO 56](#_Toc202771440)

# CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

## **Giới thiệu về đề tài**

Trong thời đại công nghệ số hiện nay, thương mại điện tử và các nền tảng trực tuyến ngày càng đóng vai trò quan trọng mật thiết trong việc thay đổi thói quen mua sắm và tiêu dùng của con người. Đặc biệt, ở ngành công nghiệp đồ ăn nhanh, với sự phát triển nhanh chóng và nhu cầu cao từ khách hàng, đã sinh ra vô vàn yêu cầu mới, và rất nhiều cá nhân, tổ chức kinh doanh đã tận dụng các giải pháp công nghệ để có thể tối ưu hóa quy trình kinh doanh, phục vụ cho việc thu lợi nhuận. Từ đó, việc phát triển một website cho cửa hàng bán đồ ăn nhanh không chỉ giúp tăng cường trải nghiệm khách hàng mà còn nâng cao hiệu quả quản lý, giảm thiểu thời gian chờ đợi và mở rộng phạm vi phục vụ.

Đề tài "Phát triển website cho cửa hàng bán đồ ăn nhanh" của tôii tập trung vào việc xây dựng một ứng dụng web cho quán đồ ăn nhanh RATFOOD, sử dụng công nghệ lập trình hiện đại, cụ thể là nền tảng ASP.NET Core MVC, kết hợp với cơ sở dữ liệu SQL Server và các công cụ hỗ trợ như Entity Framework Core và LINQ. Website này sẽ cung cấp các chức năng cơ bản như hiển thị menu, đặt hàng trực tuyến, quản lý đơn hàng, và theo dõi trạng thái giao hàng, đồng thời quản lý cửa hàng (đơn hàng, người dùng, thống kê,…). Việc lựa chọn các công nghệ này dựa trên ưu điểm vượt trội của chúng: ASP.NET Core cung cấp hiệu suất cao và khả năng mở rộng, SQL Server đảm bảo lưu trữ dữ liệu an toàn và hiệu quả, trong khi EF Core và LINQ đơn giản hóa quá trình truy vấn và quản lý dữ liệu.

Theo tôi, đề tài không chỉ mang tính thực tiễn cao mà còn rất phù hợp với xu hướng phát triển công nghệ trong ngành dịch vụ hiện nay, nó cũng còn là cơ hội để tôi áp dụng các kiến thức lý thuyết đã học về C#, ASP.NET Core MVC, SQL Server, EF Core, và LINQ vào một dự án cụ thể. Với ngày càng nhiều cửa hàng ăn nhanh chuyển sang mô hình trực tuyến, việc phát triển một website như vậy sẽ góp phần nâng cao năng lực cạnh tranh và đáp ứng nhu cầu của thị trường.

## **1.2. Mục tiêu của đề tài**

Mục tiêu chính của đề tài "Phát triển website cho cửa hàng bán đồ ăn nhanh" được xác định một cách chi tiết và toàn diện, tôi cho rằng nó sẽ bao gồm mục tiêu học thuật và thực tiễn, để đảm bảo rằng dự án không chỉ mang lại giá trị giáo dục mà còn có ý nghĩa ứng dụng trong môi trường thực tế. Cụ thể, các mục tiêu được chia thành hai nhóm chính như sau:

* **Mục tiêu học thuật:**
  + Trở nên thành thạo trong việc áp dụng các nguyên tắc lập trình hướng đối tượng trong ngôn ngữ C#, như đã được giới thiệu trong tài liệu "Tổng quan về ngôn ngữ C#". Điều này bao gồm việc sử dụng các khái niệm cốt lõi như lớp (class), đối tượng (object), kế thừa (inheritance), đa hình (polymorphism), và đóng gói (encapsulation) để xây dựng các thành phần logic của ứng dụng một cách khoa học và có tổ chức. Ví dụ, việc tạo ra các lớp như Product hay Order sẽ giúp hiểu sâu hơn về cách tổ chức mã nguồn, sử dụng các thuộc tính và phương thức để quản lý dữ liệu một cách hiệu quả.
  + Hiểu và làm chủ khung ASP.NET Core MVC trong việc thiết kế và phát triển một ứng dụng web toàn diện, như được mô tả chi tiết trong tài liệu "ASP.NET Core MVC". Điều này bao gồm việc phân chia rõ ràng các lớp Model (dữ liệu), View (giao diện), và Controller (logic xử lý), đảm bảo sự tách biệt trách nhiệm giữa các thành phần, từ đó cải thiện kỹ năng lập trình web và rèn luyện tư duy thiết kế hệ thống phần mềm theo mô hình kiến trúc chuẩn.
  + Thành thạo kỹ năng thiết kế và quản lý cơ sở dữ liệu quan hệ bằng SQL Server, dựa trên các hướng dẫn trong tài liệu "SQL Server", đồng thời tích hợp nó một cách hiệu quả với ứng dụng thông qua Entity Framework Core, như được trình bày trong "Entity Framework Core". Quá trình này bao gồm việc xây dựng các bảng dữ liệu, thiết lập mối quan hệ giữa các bảng (như quan hệ 1-nhiều giữa Orders và OrderDetails), và thực hiện các thao tác CRUD (Create, Read, Update, Delete) một cách chuyên nghiệp, đồng thời hiểu rõ cách tối ưu hóa hiệu suất truy vấn.
  + Áp dụng thành thạo LINQ để thực hiện các truy vấn dữ liệu phức tạp từ cơ sở dữ liệu hoặc các tập hợp trong bộ nhớ, như được hướng dẫn trong tài liệu "LinQ". Điều này đòi hỏi sự hiểu biết sâu sắc về cú pháp LINQ, bao gồm cả Query Syntax và Method Syntax, cũng như cách nó tương tác với EF Core để tạo ra các truy vấn hiệu quả, giúp tối ưu hóa hiệu suất và tăng tính linh hoạt của ứng dụng.
  + Rèn luyện các kỹ năng mềm quan trọng như phân tích yêu cầu người dùng, lập kế hoạch dự án, thiết kế giao diện (dựa trên "View" và "Tổ chức website"), và triển khai phần mềm từ khâu khởi đầu đến hoàn thiện. Đồng thời, người thực hiện sẽ học cách xử lý các vấn đề kỹ thuật phát sinh trong quá trình phát triển, từ việc gỡ lỗi mã nguồn đến tối ưu hóa giao diện và dữ liệu.
* **Mục tiêu thực tiễn:**
  + Xây dựng một website gồm nhiều chức năng cho cửa hàng bán đồ ăn nhanh, bao gồm các tính năng cốt lõi như hiển thị danh sách món ăn (menu) với thông tin chi tiết, cho phép khách hàng đặt hàng trực tuyến một cách nhanh chóng và tiện lợi, và cung cấp giao diện quản lý đơn hàng cho quản trị viên để theo dõi và xử lý. Website này sẽ được thiết kế để phục vụ nhu cầu thực tế, giúp khách hàng tiết kiệm thời gian và nâng cao trải nghiệm mua sắm, đồng thời hỗ trợ quản trị viên tối ưu hóa quy trình vận hành.
  + Thiết kế một giao diện thân thiện, trực quan, đảm bảo hoạt động mượt mà. Điều này không chỉ đáp ứng nhu cầu đa dạng của người dùng mà còn tăng tính tiện lợi và khả năng tiếp cận, phù hợp với xu hướng sử dụng thiết bị di động ngày càng phổ biến trong xã hội hiện đại.
  + Đảm bảo hệ thống có khả năng lưu trữ và xử lý dữ liệu một cách hiệu quả, với cơ sở dữ liệu được thiết kế hợp , nhằm quản lý thông tin về món ăn, đơn hàng, và thông tin khách hàng thông qua SQL Server. Điều này bao gồm việc tối ưu hóa các truy vấn dữ liệu bằng LINQ và EF Core, đảm bảo dữ liệu được bảo vệ an toàn, và cung cấp khả năng truy xuất nhanh chóng khi cần thiết.
  + Cung cấp một giải pháp thực tế mà các cửa hàng ăn nhanh nhỏ hoặc vừa có thể áp dụng để chuyển đổi số, giúp họ tăng doanh thu, tối ưu hóa quy trình vận hành, và xây dựng một kênh bán hàng trực tuyến bền vững. Đây là một bước quan trọng trong việc hỗ trợ các doanh nghiệp nhỏ tiếp cận với xu hướng công nghệ, đặc biệt trong bối cảnh thương mại điện tử đang bùng nổ và thay đổi thói quen tiêu dùng của người dân Việt Nam cũng như trên toàn cầu, nơi mà các nền tảng trực tuyến ngày càng trở thành lựa chọn hàng đầu.

Với các mục tiêu mà tôi đề ra, việc đạt được các mục tiêu này không chỉ giúp tôi tăng trình độ chuyên môn, củng cố kỹ năng lập trình, và tích lũy kinh nghiệm thực tế mà còn đóng góp vào việc tạo ra một sản phẩm có giá trị ứng dụng cao, hỗ trợ cộng đồng doanh nghiệp trong việc thích nghi với sự phát triển không ngừng của công nghệ và thị trường.

## **1.3. Phạm vi của đề tài**

Phạm vi của đề tài "Phát triển website cho cửa hàng bán đồ ăn nhanh" được xác định một cách chi tiết và cụ thể để đảm bảo tính khả thi trong phạm vi thời gian, nguồn lực, đồng thời tập trung vào các khía cạnh cốt lõi của dự án. Phạm vi này được tôi chia thành các phần sau để làm rõ giới hạn và định hướng thực hiện:

* **Phạm vi chức năng:**
  + Website sẽ cung cấp các chức năng cơ bản và thiết yếu nhằm đáp ứng nhu cầu cốt lõi của người dùng, bao gồm việc hiển thị danh sách món ăn (menu) với thông tin chi tiết như tên món, giá cả, hình ảnh minh họa, và mô tả ngắn gọn để khách hàng dễ dàng lựa chọn, cùng với các nút tương tác như đặt hàng. Hệ thống sẽ hỗ trợ quy trình đặt hàng trực tuyến, cho phép khách hàng nhập thông tin cá nhân (tên, số điện thoại, địa chỉ) và xác nhận đơn hàng một cách đơn giản. Đối với quản trị viên, website sẽ cung cấp giao diện quản lý chuyên dụng, cho phép theo dõi trạng thái đơn hàng (chờ xử lý, đang giao, hoàn thành), cũng như quản lý menu bằng cách thêm, sửa, hoặc xóa món ăn một cách linh hoạt.
  + Đề tài sẽ bao gồm các chức năng nâng cao như tích hợp cổng thanh toán trực tuyến (ví dụ: VNPAY, PayPal, hoặc MoMo) để xử lý thanh toán điện tử, hệ thống quản lý giao hàng tự động (liên kết với các ứng dụng như Grab, Be, hoặc Gojek), hay các tính năng như hệ thống đánh giá món ăn từ khách hàng. Những tính năng này có thể được xem là cần thiết đối với một trang web cho cửa hàng bán đồ ăn nhanh
* **Phạm vi công nghệ:**
  + Dự án sẽ sử dụng ASP.NET Core MVC làm khung phát triển chính, dựa trên các hướng dẫn trong tài liệu "ASP.NET Core MVC" và "Controller", kết hợp với ngôn ngữ lập trình C# (theo "Tổng quan về ngôn ngữ C#") để viết logic ứng dụng, đảm bảo tính mạnh mẽ, khả năng mở rộng, và tuân thủ các nguyên tắc lập trình hướng đối tượng. Cơ sở dữ liệu sẽ được xây dựng trên SQL Server, với việc tích hợp thông qua Entity Framework Core (theo "Entity Framework Core") để quản lý dữ liệu một cách hiệu quả, chuyên nghiệp, và tối ưu hóa hiệu suất truy vấn.
  + LINQ sẽ được áp dụng để thực hiện các truy vấn dữ liệu từ cơ sở dữ liệu hoặc tập hợp trong bộ nhớ, giúp đơn giản hóa mã nguồn và tăng tính linh hoạt. Giao diện người dùng sẽ được thiết kế đơn giản với Razor và hỗ trợ cơ bản bằng CSS/Bootstrap đảm bảo tính thẩm mỹ, cũng như một số thư viện bên ngoài.
* **Phạm vi người dùng:**
  + Đối tượng chính của website là khách hàng cá nhân muốn đặt đồ ăn nhanh trực tuyến và quản trị viên của cửa hàng để quản lý đơn hàng cũng như menu. Hệ thống sẽ được thiết kế với hai vai trò cơ bản: vai trò khách hàng (chỉ đọc menu, chọn món, và đặt hàng) và vai trò quản trị viên (quản lý toàn diện, bao gồm thêm/sửa/xóa món ăn và xử lý đơn hàng). Trang web của tôi sẽ không bao gồm các chức năng dành cho nhân viên giao hàng (như theo dõi lộ trình giao hàng), hệ thống quản lý kho hàng (như kiểm kê nguyên liệu), hoặc các yêu cầu đặc thù của các chuỗi cửa hàng lớn với nhiều chi nhánh và nhu cầu phức tạp.
* **Phạm vi thời gian và tài nguyên:**
  + Dự án sẽ được thực hiện trong khoảng thời gian học tập, dự kiến kéo dài khoảng gần 1 tháng, tùy thuộc vào tiến độ học tập, khối lượng công việc, và khả năng xử lý các vấn đề kỹ thuật phát sinh. Tài nguyên chính bao gồm máy tính cá nhân của tôi với phần mềm Visual Studio 2022 làm môi trường phát triển chính, SQL Server Express làm cơ sở dữ liệu miễn phí, và SQL Server Management Studio (SSMS) để quản lý và kiểm tra database một cách chi tiết.
  + Dữ liệu thử nghiệm sẽ được tạo thủ công, bao gồm danh sách món ăn mẫu (như burger, pizza, gà rán) với thông tin giả lập và các đơn hàng thử nghiệm, thay vì sử dụng dữ liệu thực tế từ một cửa hàng cụ thể. Điều này giúp tập trung vào việc phát triển và kiểm thử hệ thống mà không phụ thuộc vào nguồn dữ liệu bên ngoài, đồng thời phù hợp với mục tiêu học tập và thử nghiệm trong môi trường giáo dục.

Phạm vi được xác định như trên đảm bảo đề tài tập trung vào các khía cạnh cốt lõi, phù hợp với mục tiêu học tập và khả năng thực hiện của người thực hiện, đồng thời tạo nền tảng vững chắc để mở rộng trong các dự án tương lai nếu có thêm thời gian và nguồn lực.

## **1.4. Phương pháp nghiên cứu**

Để hoàn thành đề tài "Phát triển website cho cửa hàng bán đồ ăn nhanh" một cách khoa học, hiệu quả, và đạt được các mục tiêu đã đề ra một cách toàn diện, tôi đã đề ra các phương pháp nghiên cứu sau đây để áp dụng một cách có hệ thống, chi tiết. Chúng được triển khai qua các giai đoạn khác nhau trong quá trình thực hiện, nghiên cứu tài liệu được thực hiện kỹ lưỡng ngay từ giai đoạn đầu của dự án, bao gồm việc thu thập và phân tích các nguồn thông tin đáng tin cậy từ tài liệu học tập của Giảng viên/Ths. Phạm Văn Tiệp về lập trình C#, tài liệu hướng dẫn từ Microsoft Learn về ASP.NET Core MVC, SQL Server, Entity Framework Core, và LINQ, cùng với một vài khóa học trực tuyến từ YouTube, Udemy, hoặc Pluralsight, đặc biệt, các tài liệu bạn đã gửi như xử lý yêu cầu HTTP, thiết kế giao diện với Razor, cấu trúc dự án, truy vấn dữ liệu, và quản lý cơ sở dữ liệu được nghiên cứu kỹ lưỡng để xây dựng một nền tảng lý thuyết vững chắc, cung cấp định hướng rõ ràng cho quá trình phát triển.

Ngoài ra, việc tìm kiếm thêm thông tin từ các nguồn uy tín trên internet như tài liệu kỹ thuật từ trang web chính thức của Microsoft, blog của các chuyên gia lập trình, và các cộng đồng mở như GitHub, Stack Overflow, hoặc các diễn đàn lập trình viên Việt Nam, giúp cập nhật các công nghệ mới nhất, giải quyết các vấn đề kỹ thuật phát sinh, và đảm bảo dự án tuân thủ các tiêu chuẩn hiện hành, các tài liệu này cũng được sử dụng để tham chiếu và trích dẫn trong báo cáo, nhằm tăng tính học thuật, độ tin cậy, và minh chứng cho sự đầu tư nghiên cứu.

Phương pháp lập trình thực nghiệm được áp dụng như một cách tiếp cận thực tiễn, trong đó từng thành phần của website được triển khai và thử nghiệm một cách có hệ thống, bắt đầu từ việc thiết kế cơ sở dữ liệu với SQL Server, xây dựng các lớp Model trong C# theo nguyên tắc hướng đối tượng, phát triển các Controller để xử lý logic, và tạo các View bằng Razor, sử dụng Visual Studio 2022 kết hợp với SSMS để quản lý và kiểm tra cơ sở dữ liệu, mỗi chức năng sẽ được phát triển và thử nghiệm riêng lẻ, với các bước ghi chép lại chi tiết, bao gồm cả các lỗi gặp phải và cách khắc phục, giúp kiểm chứng tính đúng đắn của lý thuyết và rèn luyện kỹ năng lập trình thực tế.

Phương pháp phân tích và thiết kế được thực hiện chi tiết thông qua việc xây dựng các kịch bản sử dụng để mô phỏng hành vi của người dùng, xác định các chức năng cần thiết và thứ tự ưu tiên, thiết kế mô hình dữ liệu bằng sơ đồ ERD để xác định các bảng và mối quan hệ, áp dụng kỹ thuật thiết kế hướng đối tượng trong C# để xây dựng các lớp Model, đảm bảo tính đóng gói và tái sử dụng code.

Phương pháp kiểm thử và đánh giá bao gồm kiểm thử đơn vị cho từng chức năng, kiểm thử tích hợp để đảm bảo sự tương tác giữa các thành phần, và nhận phản hồi từ người dùng thử nghiệm để điều chỉnh giao diện và sửa lỗi, nhằm đảm bảo sản phẩm cuối cùng đáp ứng tốt nhất nhu cầu thực tế.

Cuối cùng, phương pháp học hỏi từ thực tiễn bao gồm việc quan sát các website bán đồ ăn nhanh thực tế để lấy cảm hứng về bố cục và quy trình, cũng như tham gia các cộng đồng lập trình để trao đổi kinh nghiệm và cập nhật phương pháp phát triển mới, sự kết hợp đồng bộ của các phương pháp này đảm bảo đề tài được thực hiện khoa học, có tổ chức, và đạt được các mục tiêu đề ra, với mọi bước được ghi chép chi tiết trong báo cáo để minh chứng cho sự nỗ lực và tính chuyên nghiệp.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## **2.1. Tổng quan về C#**

- Định nghĩa:

* Ngôn ngữ lập trình C# là một ngôn ngữ lập trình hiện đại, hướng đối tượng, được phát triển bởi Microsoft và giới thiệu lần đầu tiên vào năm 2000 bởi Anders Hejlsberg cùng nhóm phát triển của ông.
* C# là một phần của nền tảng .NET, ban đầu được thiết kế để hoạt động trên hệ điều hành Windows, nhưng với sự ra đời của .NET Core (nay là .NET), ngôn ngữ này đã trở thành đa nền tảng, hỗ trợ các hệ điều hành như macOS, Linux, và thậm chí các thiết bị di động thông qua Xamarin.
* Theo tài liệu "Tổng quan về ngôn ngữ C#.pdf", C# được xem như một sự kết hợp giữa các đặc điểm ưu việt của C, C++, và Java, mang đến một môi trường lập trình an toàn, hiệu quả, và dễ học, đặc biệt phù hợp cho việc phát triển các ứng dụng web, desktop, game, và doanh nghiệp.

- Đôi nét lịch sử:

* Giai đoạn khởi đầu (2000-2005): C# ra mắt cùng .NET Framework 1.0 vào tháng 2 năm 2000, đánh dấu bước ngoặt trong chiến lược của Microsoft nhằm cạnh tranh với Java. Phiên bản 1.0 tập trung vào các ứng dụng Windows với các đặc điểm như lập trình hướng đối tượng và quản lý bộ nhớ tự động.
* Phát triển mạnh mẽ (2005-2015): Phiên bản 2.0 (2005) giới thiệu Generics, 3.0 (2007) mang đến LINQ, và 4.0 (2010) cải tiến đa nhiệm với Task Parallel Library (TPL), giúp C# trở thành ngôn ngữ chính cho các ứng dụng doanh nghiệp.
* Thời kỳ đa nền tảng (2016-2025): Với sự ra mắt của .NET Core 1.0 vào năm 2016, C# đã mở rộng sang các nền tảng không phải Windows, dẫn đến các phiên bản như .NET 5 (2020), .NET 6 (2021), .NET 7 (2022), và .NET 8 (2023), phiên bản hiện hành vào tháng 7 năm 2025, tiếp tục cải tiến hiệu suất và hỗ trợ AI, đám mây.
* Tầm ảnh hưởng: C# đã trở thành một trong những ngôn ngữ phổ biến nhất, đứng thứ 5 trên bảng xếp hạng TIOBE Index năm 2025, nhờ cộng đồng lớn và tích hợp với các công cụ như Visual Studio.

- Chức năng:

* Lập trình hướng đối tượng: C# hỗ trợ đầy đủ các nguyên tắc như lớp (class), đối tượng (object), kế thừa (inheritance), đa hình (polymorphism), và đóng gói (encapsulation), cho phép xây dựng các cấu trúc dữ liệu phức tạp như lớp Product với thuộc tính Name và Price trong dự án website đồ ăn nhanh.
* Quản lý kiểu dữ liệu mạnh (Strong Typing): Ngôn ngữ sử dụng hệ thống kiểu dữ liệu tĩnh với các kiểu cơ bản như int, double, string, và kiểu phức tạp như List<T> hoặc Dictionary<TKey, TValue>, giúp phát hiện lỗi biên dịch sớm, ví dụ như lỗi gán sai kiểu dữ liệu khi khai báo int quantity = "abc";.
* Quản lý bộ nhớ tự động: C# tích hợp Garbage Collector (GC) để tự động giải phóng bộ nhớ không sử dụng, giảm gánh nặng cho lập trình viên so với C++ hoặc C, đảm bảo hiệu suất ổn định trong các ứng dụng web xử lý nhiều yêu cầu đồng thời.
* Hỗ trợ đa nền tảng: Nhờ .NET Core và các phiên bản sau, C# cho phép triển khai ứng dụng trên Windows, macOS, Linux, và các nền tảng đám mây như Azure, phù hợp với xu hướng phát triển ứng dụng toàn cầu.
* Tích hợp LINQ: Cung cấp Language Integrated Query để truy vấn dữ liệu từ tập hợp, XML, hoặc cơ sở dữ liệu một cách trực quan, ví dụ from p in products where p.Price > 100 select p, tăng tính linh hoạt trong xử lý dữ liệu.
* Hỗ trợ phát triển đa dạng: C# được sử dụng để xây dựng ứng dụng Windows (WPF, WinForms), web (ASP.NET Core), game (Unity), và API, với các tính năng như asynchronous programming (async/await) để xử lý tác vụ bất đồng bộ hiệu quả.

-Tác dụng:

* Tăng năng suất lập trình: Với cú pháp rõ ràng và các công cụ như Visual Studio, C# giúp lập trình viên viết mã nhanh chóng và giảm lỗi, đặc biệt hữu ích trong các dự án như website đồ ăn nhanh, nơi cần phát triển nhanh nhưng vẫn đảm bảo chất lượng.
* Hỗ trợ phát triển ứng dụng doanh nghiệp: C# là nền tảng chính cho các ứng dụng lớn như hệ thống CRM, ERP, và quản lý dữ liệu, nhờ tích hợp với Azure và SQL Server, đóng vai trò quan trọng trong chuyển đổi số của các công ty toàn cầu.
* Tác động đến ngành công nghiệp game: Thông qua Unity, C# đã trở thành ngôn ngữ chủ đạo trong phát triển game, với hàng triệu tựa game trên các nền tảng như PlayStation, Xbox, và PC được tạo ra, tạo ra hàng tỷ đô la doanh thu hàng năm.
* Thúc đẩy đổi mới công nghệ: Sự phát triển của C# trong .NET 8 (2023) với hỗ trợ AI và machine learning đã mở ra cơ hội cho các ứng dụng thông minh, như phân tích hành vi khách hàng trong website thương mại điện tử, dự đoán xu hướng bán hàng.
* Cộng đồng và hệ sinh thái mạnh mẽ: Với hàng triệu lập trình viên trên toàn cầu, C# có cộng đồng hỗ trợ lớn trên GitHub, Stack Overflow, và các diễn đàn, cùng với tài liệu phong phú từ Microsoft, giúp giải quyết vấn đề nhanh chóng và cập nhật xu hướng công nghệ.
* Ứng dụng trong giáo dục và nghiên cứu: C# là ngôn ngữ phổ biến trong các trường đại học và khóa học trực tuyến, hỗ trợ sinh viên như bạn trong việc học lập trình, thực hành dự án, và chuẩn bị cho thị trường lao động, đặc biệt trong bối cảnh Việt Nam đang đẩy mạnh công nghệ thông tin vào năm 2025.

- Áp dụng lý thuyết:

* Trong dự án "Phát triển website cho cửa hàng bán đồ ăn nhanh", C# sẽ được sử dụng để định nghĩa các lớp như Product và Order, áp dụng OOP để quản lý logic, và tích hợp với ASP.NET Core để xây dựng giao diện và xử lý yêu cầu, đảm bảo hệ thống hoạt động mượt mà và dễ bảo trì.

## **2.2. ASP.NET Core MVC**

- Định nghĩa:

* ASP.NET Core MVC là một framework phát triển ứng dụng web hiện đại, được Microsoft phát triển như một phần của nền tảng ASP.NET Core, nhằm hỗ trợ việc xây dựng các ứng dụng web động, API, và dịch vụ với hiệu suất cao và khả năng mở rộng.
* Framwork này tuân theo mô hình kiến trúc Model-View-Controller (MVC), giúp tách biệt logic kinh doanh, giao diện người dùng, và xử lý yêu cầu.
* Đây là phiên bản cải tiến của ASP.NET MVC truyền thống, được tối ưu hóa để chạy trên nhiều nền tảng như Windows, macOS, và Linux, đồng thời hỗ trợ tích hợp với các công nghệ đám mây như Azure.

- Đôi nét lịch sử:

* Giai đoạn khởi đầu (2016): ASP.NET Core MVC ra mắt cùng ASP.NET Core 1.0 vào tháng 6 năm 2016, đánh dấu sự chuyển mình từ ASP.NET 4.x sang một khung công tác đa nền tảng, nhẹ hơn và hiệu quả hơn.
* Phát triển qua các phiên bản (2016-2025): Phiên bản 2.0 (2017) cải thiện hiệu suất với Razor Pages, 3.0 (2019) hỗ trợ endpoint routing, và 6.0 (2021) thêm tính năng minimal APIs, dẫn đến phiên bản 8.0 (2023) hiện hành vào tháng 7 năm 2025, tập trung vào hiệu suất và tích hợp AI.
* Tầm ảnh hưởng: ASP.NET Core MVC đã trở thành lựa chọn hàng đầu cho phát triển web doanh nghiệp, với hơn 30% các ứng dụng web .NET được xây dựng dựa trên khung này theo báo cáo Stack Overflow 2025, nhờ tính linh hoạt và cộng đồng hỗ trợ lớn.

- Cơ chế hoạt động:

* Quy trình xử lý yêu cầu bắt đầu khi người dùng gửi HTTP Request qua URL (ví dụ: /Home/Index), Router sẽ ánh xạ URL đến Controller và Action tương ứng.
* Controller tương tác với Model để lấy dữ liệu (thường qua Entity Framework Core), sau đó chọn View để hiển thị giao diện, sử dụng Razor để tạo HTML động.
* Dữ liệu được truyền từ Model sang View qua ViewBag hoặc Model Binding, đảm bảo sự tách biệt giữa logic và giao diện, giúp bảo trì mã nguồn dễ dàng hơn.
* Hệ thống sử dụng middleware pipeline để xử lý yêu cầu, từ xác thực, định tuyến, đến phản hồi, tối ưu hóa hiệu suất cho các ứng dụng web như website đồ ăn nhanh.

- Chức năng:

* Model: Đại diện cho dữ liệu và logic kinh doanh, thường là các lớp C# như Product hoặc Order, chịu trách nhiệm tương tác với cơ sở dữ liệu để cung cấp thông tin cho Controller.
* View: Hiển thị dữ liệu cho người dùng, được xây dựng bằng Razor với cú pháp như @model List<Product> để lặp qua danh sách món ăn, hỗ trợ Tag Helpers để tạo liên kết động như <a asp-controller="Home" asp-action="Index">.
* Controller: Xử lý yêu cầu HTTP, lấy dữ liệu từ Model, và trả về View, ví dụ phương thức Index() trong HomeController hiển thị trang menu.
* Dependency Injection: Tích hợp sẵn để quản lý các dịch vụ như DbContext, hỗ trợ kiểm thử đơn vị và tái sử dụng mã nguồn.
* Razor Pages: Một lựa chọn thay thế cho MVC, phù hợp với các ứng dụng nhỏ, nhưng MVC được ưu tiên trong dự án nhờ khả năng tổ chức tốt hơn.

- Tác dụng:

* Tăng hiệu suất phát triển: ASP.NET Core MVC cung cấp các công cụ như scaffolding để tạo mã nguồn tự động, giảm thời gian phát triển, đặc biệt hữu ích cho dự án website cần triển khai nhanh như của bạn.
* Hỗ trợ ứng dụng doanh nghiệp: Khung này được sử dụng rộng rãi trong các hệ thống lớn như quản lý khách hàng (CRM) và thương mại điện tử, nhờ khả năng mở rộng và tích hợp với Azure, góp phần thúc đẩy chuyển đổi số toàn cầu.
* Tối ưu hóa hiệu suất: Với pipeline HTTP nhẹ và hỗ trợ caching, ASP.NET Core MVC xử lý hàng triệu yêu cầu mỗi ngày, như trong các trang web thương mại điện tử lớn, đảm bảo trải nghiệm mượt mà cho người dùng.
* Thúc đẩy đổi mới công nghệ: Phiên bản 8.0 (2023) tích hợp hỗ trợ AI và machine learning, cho phép phân tích hành vi khách hàng hoặc dự đoán xu hướng đặt hàng trong website đồ ăn nhanh, mở ra kỷ nguyên mới cho ứng dụng thông minh.
* Cộng đồng và tài liệu phong phú: Với hàng triệu lập trình viên trên toàn cầu, ASP.NET Core MVC có tài liệu từ Microsoft Learn, diễn đàn như Stack Overflow, và các khóa học trực tuyến, hỗ trợ bạn trong việc học và giải quyết vấn đề vào thời điểm 04:29 AM +07 ngày 7 tháng 7 năm 2025.
* Tác động giáo dục: Đây là công cụ giảng dạy phổ biến trong các trường đại học và khóa học trực tuyến, giúp sinh viên như bạn áp dụng lý thuyết vào thực tế, chuẩn bị cho thị trường lao động công nghệ thông tin đang phát triển mạnh tại Việt Nam.

## **2.3. SQL Server**

- Định nghĩa: SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS) mạnh mẽ, được thiết kế để lưu trữ, quản lý, và truy xuất dữ liệu một cách hiệu quả, phục vụ cho các ứng dụng doanh nghiệp và phát triển phần mềm. Nó cung cấp một môi trường an toàn và có khả năng mở rộng, phù hợp cho cả các dự án nhỏ đến các hệ thống quy mô lớn.

- Đôi nét lịch sử: SQL Server được giới thiệu lần đầu tiên vào năm 1989 bởi Microsoft, ban đầu là một sản phẩm hợp tác với Sybase, nhưng từ phiên bản 6.0 (1995), Microsoft đã phát triển độc lập. Các phiên bản quan trọng bao gồm 2000 (thêm hỗ trợ XML), 2008 (cải tiến bảo mật), 2016 (tích hợp đám mây), và 2022 (hiệu suất cao), với phiên bản mới nhất là SQL Server 2025, ra mắt vào đầu năm 2025, tập trung vào trí tuệ nhân tạo và tích hợp với Azure, phản ánh xu hướng công nghệ hiện tại vào thời điểm 04:36 AM +07 ngày 7 tháng 7 năm 2025.

- Cơ chế hoạt động: SQL Server tổ chức dữ liệu thành các bảng với cột và hàng, sử dụng khóa chính (primary key) và khóa ngoại (foreign key) để thiết lập mối quan hệ giữa các bảng, chẳng hạn như bảng Products liên kết với OrderDetails qua khóa ProductId. Nó sử dụng ngôn ngữ Transact-SQL (T-SQL), một biến thể của SQL chuẩn, để thực hiện các thao tác như truy vấn (SELECT), chèn (INSERT), cập nhật (UPDATE), và xóa (DELETE). Hệ thống cũng tích hợp các công cụ như SQL Server Management Studio (SSMS) để quản lý database và giám sát hiệu suất, đồng thời hỗ trợ các tiến trình nền như sao lưu tự động và tối ưu hóa truy vấn.

- Chức năng:

* Lưu trữ và quản lý dữ liệu: Cho phép tạo database chứa các bảng, view, stored procedures, và triggers, ví dụ bảng Orders với cột OrderId, CustomerName, và OrderDate.
* Truy vấn dữ liệu: Hỗ trợ T-SQL để thực hiện các truy vấn phức tạp, như SELECT \* FROM Products WHERE Price > 50 ORDER BY Name, giúp lọc và sắp xếp thông tin nhanh chóng.
* Bảo mật: Cung cấp mã hóa dữ liệu, xác thực tích hợp Windows, và quản lý quyền truy cập chi tiết, đảm bảo an toàn cho thông tin nhạy cảm như dữ liệu khách hàng.
* Hỗ trợ phân tích: Tích hợp SQL Server Reporting Services (SSRS) và Integration Services (SSIS) để tạo báo cáo và tích hợp dữ liệu, phù hợp cho việc phân tích xu hướng bán hàng.
* Tích hợp đám mây: Hỗ trợ Azure SQL Database, cho phép mở rộng quy mô và sao lưu dữ liệu trên đám mây, phù hợp với các ứng dụng hiện đại.

- Tác dụng:

* Hỗ trợ doanh nghiệp: SQL Server là nền tảng chính cho các hệ thống quản lý quan hệ khách hàng (CRM), quản lý nguồn lực doanh nghiệp (ERP), và thương mại điện tử, xử lý hàng triệu giao dịch mỗi ngày, như trong các chuỗi bán lẻ toàn cầu.
* Tối ưu hóa hiệu suất: Với các cơ chế như indexing và query optimization, SQL Server xử lý dữ liệu nhanh chóng, đảm bảo trải nghiệm mượt mà cho các website có lưu lượng truy cập cao, như website đồ ăn nhanh vào giờ cao điểm.
* Thúc đẩy đổi mới công nghệ: Phiên bản 2025 tích hợp AI để phân tích dữ liệu thời gian thực, hỗ trợ dự đoán nhu cầu khách hàng hoặc tối ưu hóa kho hàng, mở ra kỷ nguyên mới cho các ứng dụng thông minh.
* Tác động kinh tế: SQL Server đóng góp hàng tỷ đô la cho ngành công nghiệp phần mềm, với hơn 50% các công ty Fortune 500 sử dụng nó vào năm 2025, theo báo cáo thị trường dữ liệu.
* Hỗ trợ giáo dục và nghiên cứu: Là công cụ giảng dạy phổ biến trong các trường đại học và khóa học trực tuyến, giúp sinh viên như bạn học về quản lý dữ liệu và chuẩn bị cho thị trường lao động công nghệ thông tin tại Việt Nam.
* Cộng đồng và hệ sinh thái: Với hàng triệu người dùng và tài liệu phong phú từ Microsoft, SQL Server có cộng đồng hỗ trợ mạnh mẽ, giúp giải quyết vấn đề nhanh chóng vào thời điểm 04:36 AM +07 ngày 7 tháng 7 năm 2025.

## **2.4. Entity Framework Core**

- Định nghĩa: Entity Framework Core (EF Core) là một công cụ ánh xạ quan hệ đối tượng (Object-Relational Mapping - ORM) mạnh mẽ, được thiết kế để đơn giản hóa việc tương tác giữa mã nguồn C# và cơ sở dữ liệu quan hệ, cho phép lập trình viên làm việc với dữ liệu như các đối tượng C# thay vì viết SQL thủ công. Đây là phiên bản hiện đại, nhẹ, và đa nền tảng của Entity Framework, phù hợp cho các ứng dụng web, desktop, và dịch vụ.

- Đôi nét lịch sử: EF Core ra mắt lần đầu vào tháng 6 năm 2016 cùng với .NET Core 1.0, đánh dấu sự chuyển mình từ Entity Framework (EF) truyền thống, vốn chỉ hỗ trợ Windows, sang một khung công tác đa nền tảng. Các phiên bản quan trọng bao gồm 2.0 (2017) với hỗ trợ LINQ cải tiến, 3.0 (2019) tích hợp C# 8.0, 5.0 (2020) tăng hiệu suất, và 8.0 (2023), phiên bản hiện hành vào thời điểm 04:40 AM +07 ngày 7 tháng 7 năm 2025, tập trung vào tối ưu hóa và tích hợp với các công nghệ mới như AI và đám mây.

- Cơ chế hoạt động: EF Core hoạt động bằng cách ánh xạ các lớp C# (entities) thành bảng trong cơ sở dữ liệu và ngược lại, sử dụng một lớp trung tâm gọi là DbContext để quản lý phiên làm việc. Nó hỗ trợ tạo và cập nhật schema cơ sở dữ liệu thông qua Migrations, theo dõi thay đổi trên các đối tượng (Change Tracking), và chuyển đổi các truy vấn LINQ thành SQL để thực thi trên database. Ví dụ, một truy vấn như context.Products.Where(p => p.Price > 100) sẽ được dịch thành câu lệnh SQL tương ứng.

- Chức năng:

* Ánh xạ đối tượng (ORM): Chuyển đổi các lớp C# như Product thành bảng Products, với các thuộc tính như Id và Name ánh xạ thành cột trong database.
* Hỗ trợ đa nền tảng: Tương thích với các database như SQL Server, SQLite, MySQL, và PostgreSQL, cho phép triển khai trên Windows, macOS, và Linux.
* Migrations: Tự động tạo và cập nhật schema database dựa trên mã nguồn, ví dụ lệnh dotnet ef migrations add InitialCreate tạo bảng mới.
* LINQ Integration: Cho phép sử dụng cú pháp LINQ để truy vấn dữ liệu, như context.Orders.OrderBy(o => o.OrderDate), tăng tính linh hoạt và dễ đọc.
* Change Tracking: Theo dõi thay đổi trên các đối tượng và tạo lệnh SQL để cập nhật database khi gọi SaveChanges().
* Các cách tiếp cận: Hỗ trợ Code First (viết mã trước, tạo database sau), Database First (tạo mô hình từ database), và Model First (thiết kế mô hình bằng công cụ).

- Tác dụng to lớn:

* Tăng năng suất lập trình: EF Core giảm lượng mã cần viết bằng cách tự động tạo truy vấn SQL, giúp bạn tập trung vào logic ứng dụng thay vì quản lý database thủ công, đặc biệt hữu ích cho dự án website đồ ăn nhanh vào 04:40 AM +07 ngày 7 tháng 7 năm 2025.
* Hỗ trợ phát triển ứng dụng doanh nghiệp: Là nền tảng cho các hệ thống lớn như quản lý khách hàng và thương mại điện tử, xử lý hàng triệu bản ghi dữ liệu mỗi ngày, góp phần thúc đẩy chuyển đổi số toàn cầu.
* Tối ưu hóa hiệu suất: Với các tính năng như Lazy Loading và Eager Loading, EF Core cải thiện tốc độ truy vấn, đảm bảo website của bạn hoạt động mượt mà khi có nhiều đơn hàng đồng thời.
* Thúc đẩy đổi mới công nghệ: Phiên bản 8.0 (2023) tích hợp với AI để phân tích dữ liệu thời gian thực, hỗ trợ dự đoán nhu cầu khách hàng hoặc tối ưu hóa quản lý kho, phù hợp với xu hướng công nghệ hiện đại.
* Tác động giáo dục: EF Core là công cụ giảng dạy phổ biến trong các khóa học lập trình, giúp sinh viên như bạn học về ORM và chuẩn bị cho thị trường lao động công nghệ thông tin đang phát triển tại Việt Nam vào năm 2025.
* Cộng đồng và hệ sinh thái: Với hàng triệu người dùng và tài liệu từ Microsoft, EF Core có cộng đồng hỗ trợ mạnh mẽ, cung cấp giải pháp nhanh chóng cho các vấn đề kỹ thuật bất kỳ lúc nào, kể cả vào sáng sớm như hiện tại.

## **2.5. LINQ**

- Định nghĩa: Language Integrated Query (LINQ) là một tính năng tích hợp trong các ngôn ngữ lập trình thuộc nền tảng .NET, như C#, cho phép lập trình viên truy vấn dữ liệu từ nhiều nguồn khác nhau (tập hợp, XML, cơ sở dữ liệu) bằng cách sử dụng cú pháp nhất quán và được nhúng trực tiếp vào mã nguồn. Đây là một công cụ mạnh mẽ giúp đơn giản hóa việc xử lý dữ liệu, đặc biệt khi làm việc với các tập hợp lớn hoặc cơ sở dữ liệu quan hệ.

- Đôi nét lịch sử: LINQ được giới thiệu lần đầu vào năm 2007 trong .NET Framework 3.5, cùng với C# 3.0, như một phần của nỗ lực Microsoft nhằm tích hợp các khả năng truy vấn dữ liệu vào ngôn ngữ lập trình. Các phiên bản sau như .NET 4.0 (2010) cải tiến hiệu suất, .NET Core 2.0 (2017) mở rộng hỗ trợ đa nền tảng, và đến phiên bản .NET 8.0 (2023), hiện hành vào thời điểm 04:46 AM +07 ngày 7 tháng 7 năm 2025, LINQ tiếp tục được tối ưu hóa để tích hợp với các công nghệ mới như AI và xử lý dữ liệu lớn, khẳng định vai trò quan trọng trong hệ sinh thái .NET.

- Cơ chế hoạt động: LINQ hoạt động bằng cách cung cấp hai cú pháp chính: Query Syntax (giống SQL) và Method Syntax (dựa trên phương thức), cho phép truy vấn dữ liệu từ các nguồn như tập hợp trong bộ nhớ (LINQ to Objects), cơ sở dữ liệu (LINQ to Entities qua Entity Framework Core), hoặc XML (LINQ to XML). Cơ chế Deferred Execution đảm bảo rằng truy vấn chỉ được thực thi khi cần kết quả (ví dụ khi gọi ToList() hoặc First()), giúp tối ưu hóa hiệu suất bằng cách trì hoãn việc tính toán cho đến khi thực sự cần thiết.

- Chức năng:

* Lọc dữ liệu: Sử dụng Where để chọn các phần tử thỏa mãn điều kiện, ví dụ products.Where(p => p.Price > 100) lấy các món ăn đắt hơn 100.
* Sắp xếp dữ liệu: Áp dụng OrderBy hoặc OrderByDescending để sắp xếp, như orders.OrderBy(o => o.OrderDate) theo ngày đặt hàng.
* Nhóm dữ liệu: Sử dụng GroupBy để phân loại, ví dụ products.GroupBy(p => p.Category) nhóm món ăn theo danh mục.
* Nối dữ liệu: Hỗ trợ Join để kết hợp nhiều tập hợp, như ghép Orders và OrderDetails dựa trên khóa chung.
* Tổng hợp dữ liệu: Cung cấp các phương thức như Count(), Sum(p => p.Price), và Average(p => p.Price) để tính toán thống kê.
* Hỗ trợ đa nguồn: Tích hợp với LINQ to Objects, LINQ to SQL, LINQ to Entities, và LINQ to XML, phù hợp cho nhiều ngữ cảnh khác nhau.

- Tác dụng to lớn:

* Tăng năng suất lập trình: LINQ cho phép viết truy vấn trực quan và ngắn gọn, giảm lỗi thủ công so với SQL thô, giúp bạn tiết kiệm thời gian khi phát triển website đồ ăn nhanh vào 04:46 AM +07 ngày 7 tháng 7 năm 2025.
* Hỗ trợ ứng dụng doanh nghiệp: Là công cụ cốt lõi trong các hệ thống lớn như quản lý kho và phân tích dữ liệu, xử lý hàng triệu bản ghi mỗi ngày, góp phần vào chuyển đổi số toàn cầu.
* Tối ưu hóa hiệu suất: Với Deferred Execution và khả năng biên dịch LINQ thành SQL hiệu quả qua Entity Framework Core, LINQ đảm bảo truy vấn nhanh chóng, phù hợp cho các ứng dụng web có lưu lượng cao.
* Thúc đẩy đổi mới công nghệ: Tích hợp với .NET 8.0 (2023), LINQ hỗ trợ xử lý dữ liệu lớn và tích hợp AI, giúp phân tích hành vi khách hàng hoặc dự đoán xu hướng đặt hàng trong thời gian thực.
* Tác động giáo dục: LINQ là một phần quan trọng trong giáo dục lập trình, giúp sinh viên như bạn học về truy vấn dữ liệu và chuẩn bị cho thị trường lao động công nghệ thông tin đang bùng nổ tại Việt Nam vào năm 2025.
* Cộng đồng và hệ sinh thái: Với hàng triệu người dùng và tài liệu từ Microsoft, LINQ có cộng đồng hỗ trợ mạnh mẽ, cung cấp giải pháp nhanh chóng cho các vấn đề kỹ thuật bất kỳ lúc nào, kể cả vào sáng sớm như hiện tại.

# CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG

## **3.1. Phân tích đặc tả yêu cầu**

### 3.1.1. Yêu cầu chức năng

#### a) Quản lý thông tin cá nhân

* Người dùng có thể xem và cập nhật thông tin cá nhân (họ tên, email, số điện thoại, địa chỉ).
* Người dùng có thể đổi mật khẩu tài khoản.

#### b) Xem thực đơn, tìm kiếm món ăn

* Người dùng có thể xem danh sách các món ăn, lọc theo danh mục.
* Người dùng có thể tìm kiếm món ăn theo tên hoặc mô tả.

#### c) Xem chi tiết món ăn, đánh giá món

* Người dùng có thể xem chi tiết từng món ăn, bao gồm hình ảnh, mô tả, giá, đánh giá.
* Người dùng đã từng đặt món có thể gửi đánh giá cho món ăn đó.

#### d) Giỏ hàng và đặt hàng

* Người dùng có thể thêm món ăn vào giỏ hàng, cập nhật số lượng hoặc xóa món khỏi giỏ.
* Người dùng có thể tiến hành đặt hàng, nhập thông tin giao hàng (địa chỉ, số điện thoại, ghi chú).
* Hệ thống xác nhận đơn hàng, cập nhật trạng thái đơn hàng.

#### e) Xem lịch sử đơn hàng

* Người dùng có thể xem danh sách các đơn hàng đã đặt, trạng thái đơn hàng, chi tiết từng đơn.

#### f) Quản trị hệ thống (Admin)

* Admin có thể quản lý món ăn: thêm, sửa, xóa, cập nhật thông tin, hình ảnh, trạng thái món.
* Admin có thể quản lý danh mục món ăn: thêm, sửa, xóa, đồng bộ dữ liệu danh mục.
* Admin có thể quản lý đơn hàng: xem danh sách, cập nhật trạng thái đơn hàng.
* Admin có thể quản lý người dùng: xem danh sách, thông tin tài khoản.
* Admin có thể xem dashboard thống kê: số lượng món ăn, đơn hàng, người dùng, doanh thu.

#### g) Các trang thông tin

* Người dùng có thể xem các trang: Trang chủ, Giới thiệu, Liên hệ, Cửa hàng, Tin tức, Khuyến mãi.

### 3.1.2. Yêu cầu phi chức năng

#### a) Bảo mật

* Thông tin người dùng, đơn hàng, đánh giá được bảo vệ an toàn.
* Chỉ admin mới có quyền truy cập các chức năng quản trị.

#### b) Hiệu năng

* Hệ thống phản hồi nhanh, xử lý thao tác người dùng trong thời gian ngắn.
* Có khả năng phục vụ nhiều người dùng đồng thời.

#### c) Giao diện

* Giao diện thân thiện, dễ sử dụng trên cả máy tính và thiết bị di động.
* Hiển thị rõ ràng các thông tin món ăn, đơn hàng, đánh giá.

#### d) Tính ổn định và bảo trì

* Hệ thống hoạt động ổn định, ít lỗi.
* Dễ dàng mở rộng, nâng cấp thêm chức năng mới.

#### e) Lưu trữ và sao lưu dữ liệu

* Dữ liệu được lưu trữ an toàn trên SQL Server.
* Có khả năng sao lưu và phục hồi dữ liệu khi cần thiết.

## **3.2. Thiết kế hệ thống**

### 3.2.1. Usecase Diagram

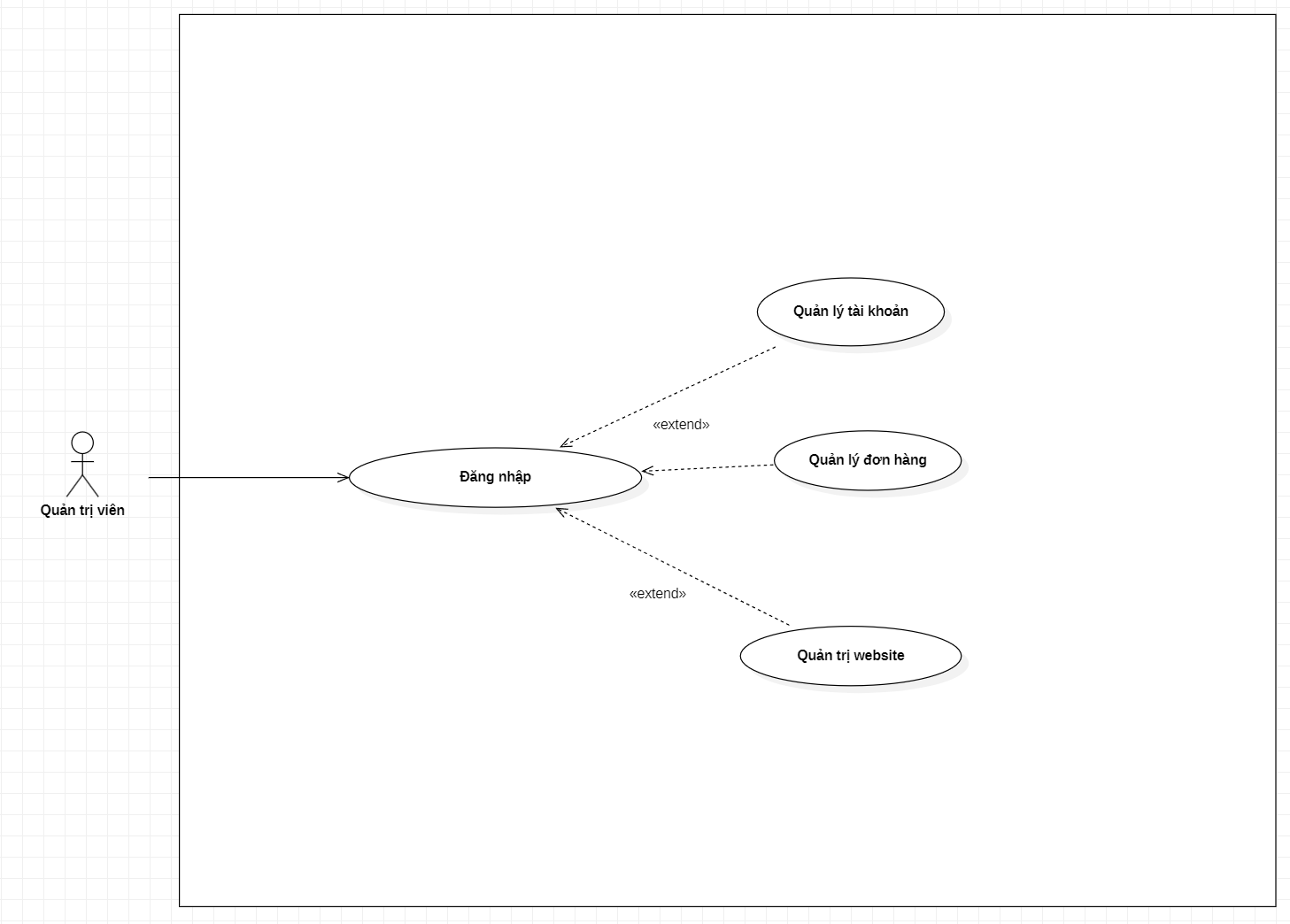
#### a) Biểu đồ Usecase tổng thể



#### b) Biểu đồ Usecase tác nhân Khách hàng

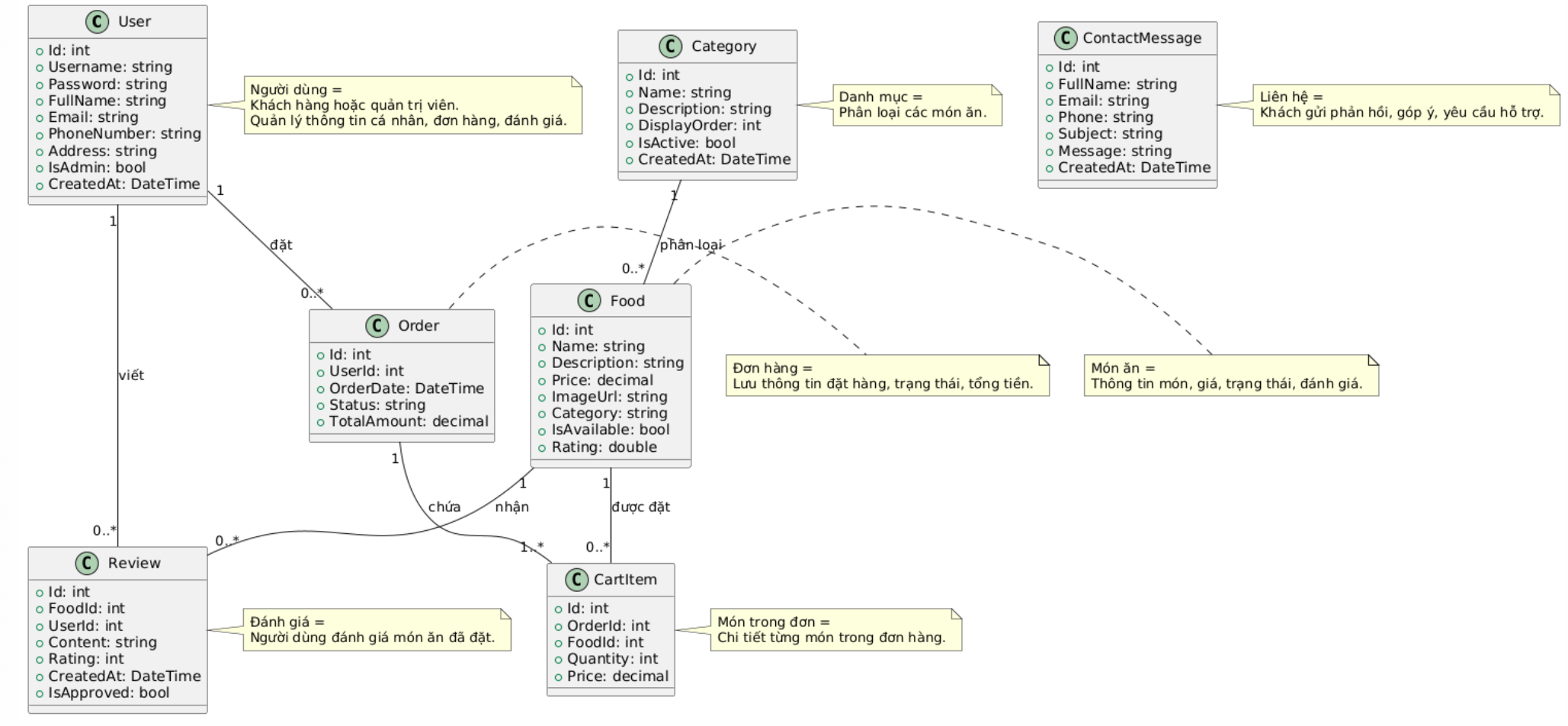


#### c) Biểu đồ Usecase tác nhân Quản trị viên



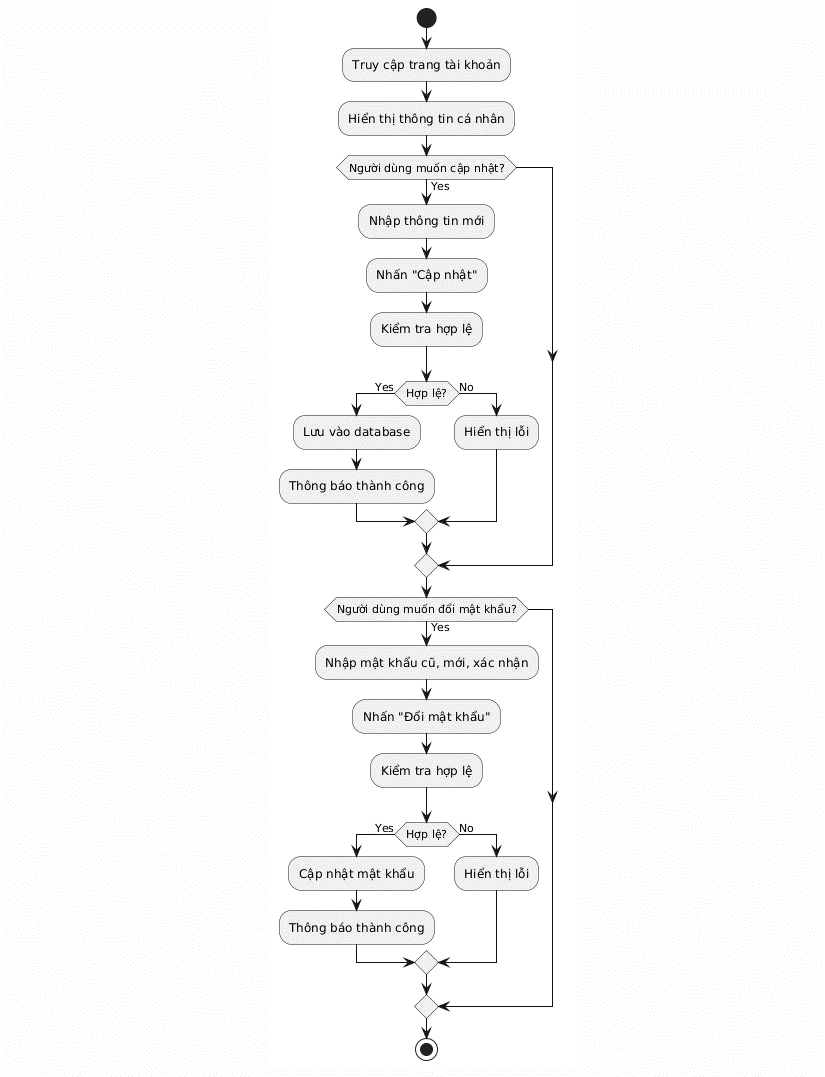
### 3.2.2. Class Diagram

Dưới đây là biểu đồ lớp của hệ thống website:



### 3.2.3. Activity Diagram

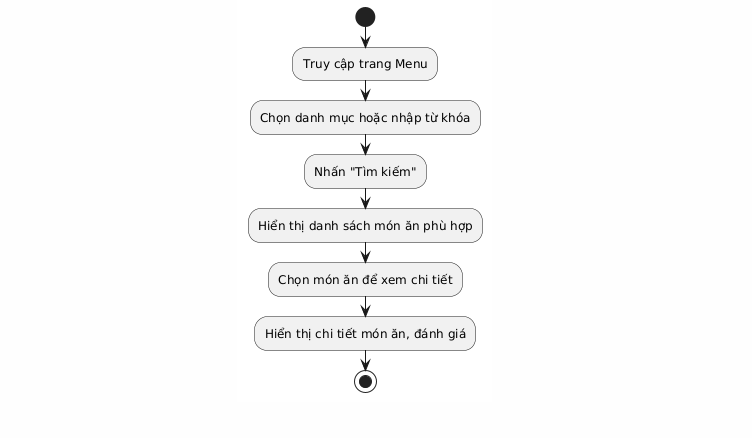
#### a) Quản lý tài khoản



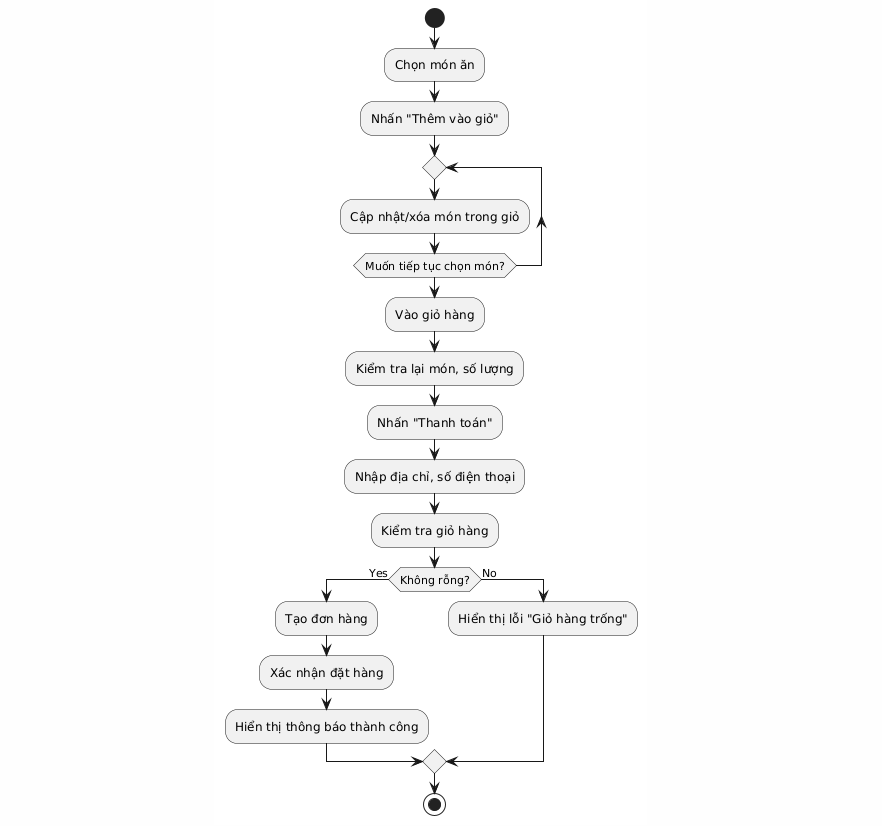
#### b) Quản lý đơn hàng



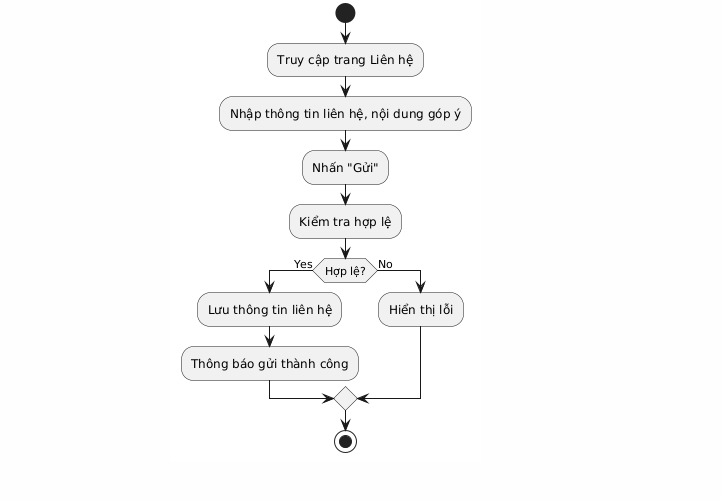
#### c) Tìm kiếm thông tin



#### d) Đặt hàng



#### e) Liên hệ

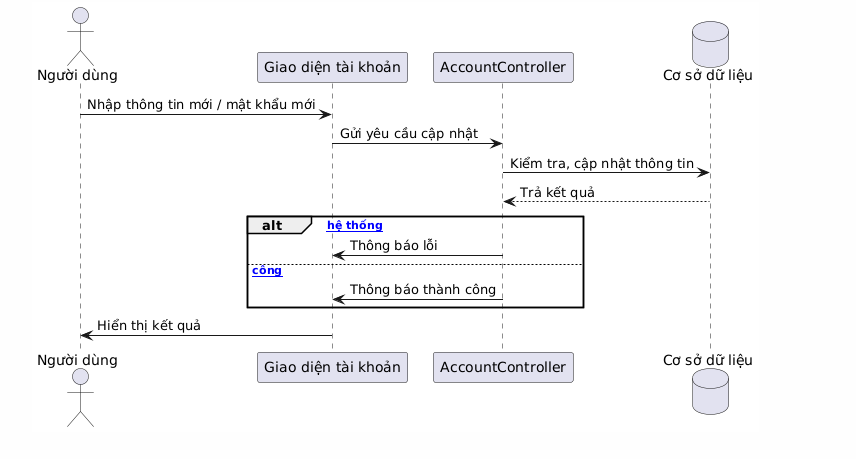


#### f) Quản trị website

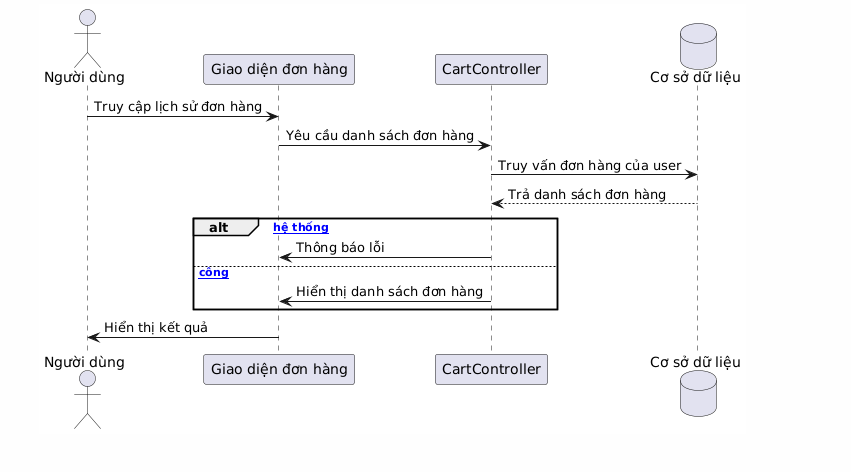


### 3.2.4. Sequence Diagram

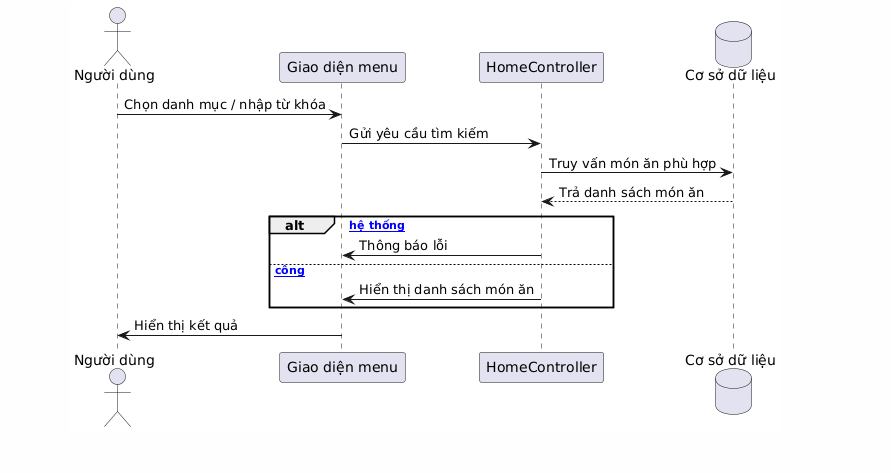
#### a) Quản lý tài khoản



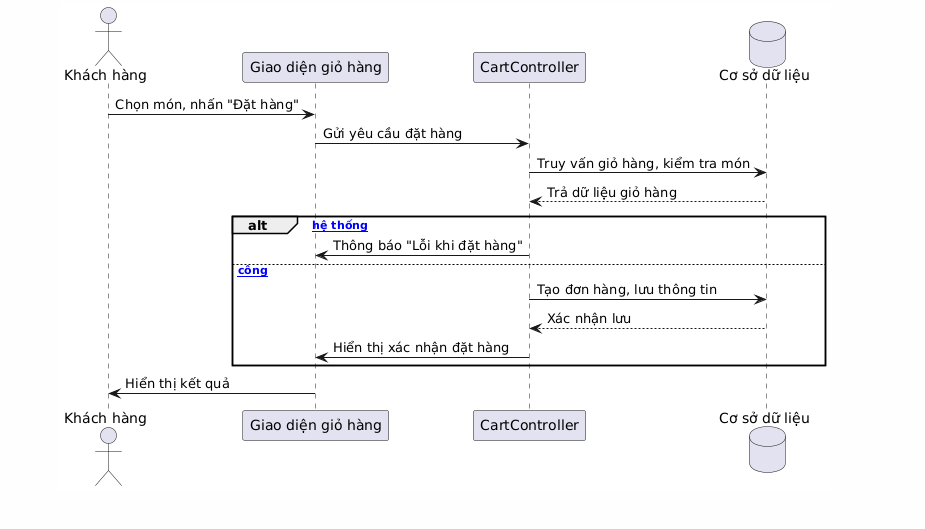
#### b) Quản lý đơn hàng



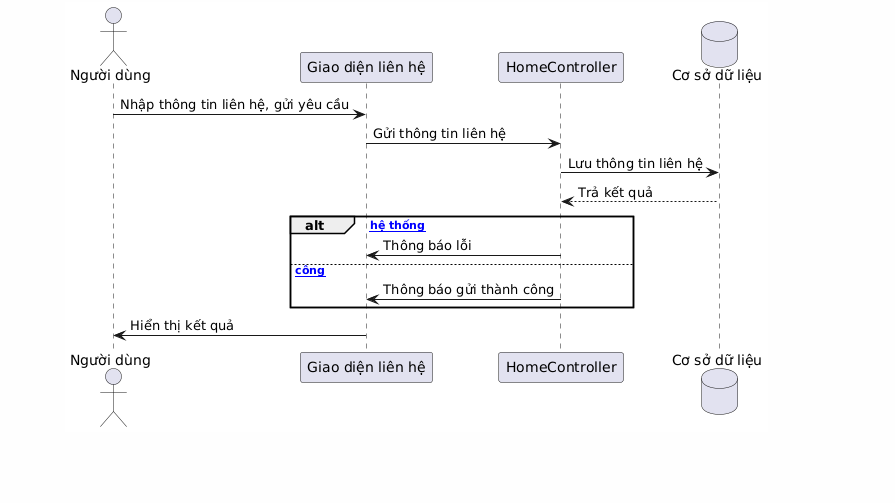
#### c) Tìm kiếm thông tin



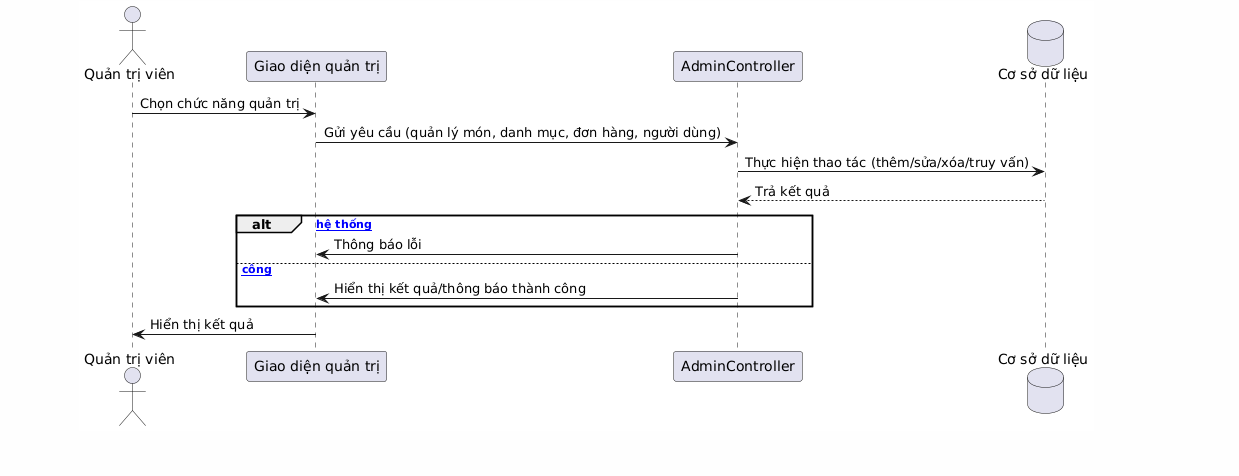
#### d) Đặt hàng



#### e) Liên hệ

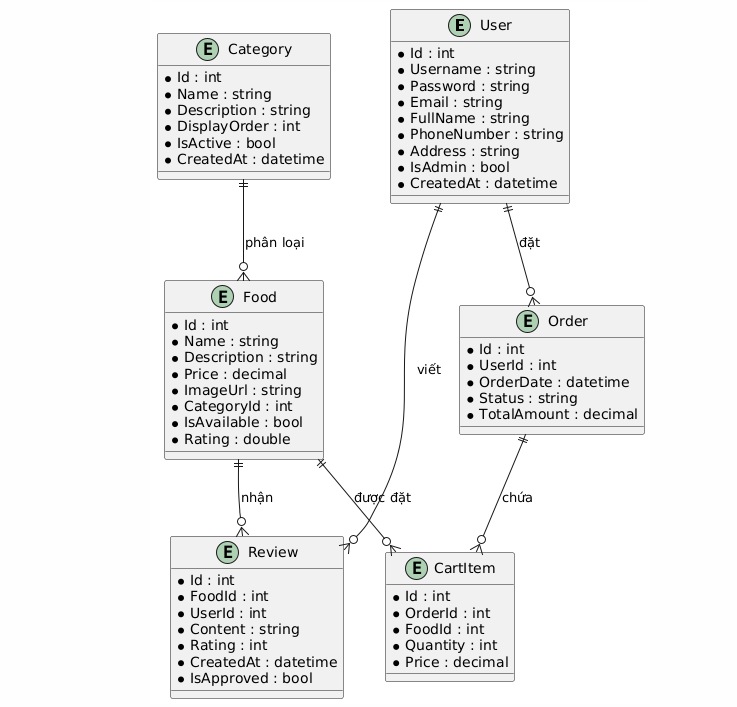


#### f) Quản trị website



## **3.3. Thiết kế cơ sở dữ liệu**

Cơ sở dữ liệu của hệ thống RatFood được thiết kế nhằm lưu trữ và quản lý toàn bộ thông tin liên quan đến người dùng, món ăn, đơn hàng, đánh giá, danh mục, và các chức năng quản trị. Việc thiết kế CSDL hợp lý giúp đảm bảo tính toàn vẹn dữ liệu, hiệu năng truy xuất và khả năng mở rộng hệ thống. Dưới đây là sơ đồ thực thể quan hệ (ERD):



## **3.4. Thiết kế giao diện**

### 3.4.1. Trang chủ (Home/Index)

#### a) Header

* Logo "RatFood" ở góc trái, sử dụng font chữ đậm màu trắng
* Menu điều hướng ngang: Home, Menu, About, Contact, News, Promotions
* Bên phải có icon giỏ hàng và nút đăng nhập/đăng ký
* Background màu đỏ (#dc3545) với text màu trắng

#### b) Banner chính

* Hình ảnh banner toàn màn hình hiển thị món ăn nổi bật
* Overlay text: "Chào mừng đến với RatFood - Hương vị Việt Nam"
* Nút "Xem Menu" màu xanh lá, bo tròn

#### c) Phần danh mục

* Tiêu đề "Danh mục món ăn" căn giữa
* Grid 4 cột hiển thị các danh mục: Món chính, Món phụ, Tráng miệng, Đồ uống
* Mỗi danh mục có hình ảnh tròn, tên danh mục bên dưới
* Hover effect: phóng to nhẹ và thêm shadow

#### d) Phần món ăn nổi bật

* Tiêu đề "Món ăn được yêu thích"
* Grid 3 cột hiển thị 6 món ăn
* Mỗi card có: hình ảnh món ăn, tên món, giá, nút "Thêm vào giỏ"
* Card có border radius 10px, shadow nhẹ

#### e) Footer

* 3 cột: Thông tin liên hệ, Giờ mở cửa, Bản quyền
* Background màu xám đậm, text màu trắng

### 3.4.2. Trang Menu (Home/Menu)

#### a) Header giống trang chủThanh tìm kiếm và lọc

* Ô tìm kiếm rộng với placeholder "Tìm kiếm món ăn..."
* Dropdown chọn danh mục: "Tất cả", "Món chính", "Món phụ", "Tráng miệng", "Đồ uống"
* Nút "Tìm kiếm" màu xanh lá

#### b) Hiển thị món ăn

* Grid 3 cột responsive
* Mỗi card món ăn có:
* Hình ảnh món ăn (tỷ lệ 16:9)
* Tên món ăn (font-weight bold)
* Mô tả ngắn (2-3 dòng, màu xám)
* Giá tiền (màu đỏ, font-size lớn)
* Nút "Thêm vào giỏ" (màu xanh lá, full width)
* Hover effect: card nổi lên với shadow đậm hơn

#### c) Phân trang:

* Các nút số trang ở cuối
* Nút "Trước" và "Sau"
* Trang hiện tại được highlight màu đỏ

### 3.4.3. Trang chi tiết món ăn (Home/FoodDetails)

#### a) Breadcrumb

* Home > Menu > [Tên món ăn]

#### b) Layout 2 cột:Cột trái (60%)

* Hình ảnh món ăn lớn (full width)
* Gallery ảnh nhỏ bên dưới (4 ảnh)

#### c) Cột phải (40%)

* Tên món ăn (font-size lớn, font-weight bold)
* Giá tiền (màu đỏ, font-size 24px)
* Mô tả chi tiết (đoạn văn dài)
* Phần chọn số lượng:
* Nút "-" và "+"
* Ô hiển thị số lượng
* Nút "Thêm vào giỏ" (màu xanh lá, full width)
* Thông tin dinh dưỡng (nếu có)

#### d) Phần đánh giá

* Tiêu đề "Đánh giá từ khách hàng"
* Hiển thị rating trung bình (sao)
* Danh sách các review:
* Avatar người dùng (hình tròn)
* Tên người dùng
* Số sao đánh giá
* Nội dung review
* Thời gian đánh giá

### 3.4.4. Trang giỏ hàng (Cart/Index)

#### a) Tiêu đề "Giỏ hàng của bạn"Danh sách món ăn

* Mỗi item có layout ngang:
* Hình ảnh món ăn (kích thước nhỏ)
* Thông tin món: tên, giá đơn vị
* Control số lượng: nút "-", ô số lượng, nút "+"
* Tổng tiền của món đó
* Nút "Xóa" (màu đỏ nhỏ)
* Border bottom giữa các item

#### b) Tổng kết

* Bảng tổng kết bên phải:
* Tạm tính: [số tiền]
* Phí vận chuyển: [số tiền]
* Tổng cộng: [số tiền lớn, màu đỏ]
* Nút "Tiếp tục mua hàng" (màu xám)
* Nút "Thanh toán" (màu xanh lá, full width)

#### c) Giỏ hàng trống

* Icon giỏ hàng rỗng (màu xám)
* Text "Giỏ hàng của bạn đang trống"
* Nút "Mua sắm ngay" (màu xanh lá)

### 3.4.5. Trang thanh toán (Cart/Checkout)

#### a) Form thanh toán 2 cột:Cột trái - Thông tin giao hàng

* Tiêu đề "Thông tin giao hàng"
* Form fields:
* Họ tên (text input)
* Số điện thoại (tel input)
* Địa chỉ giao hàng (textarea)
* Ghi chú (textarea)
* Tất cả fields có label rõ ràng và validation

#### b) Cột phải - Tóm tắt đơn hàng

* Tiêu đề "Tóm tắt đơn hàng"
* Danh sách món ăn đã chọn:
* Tên món x Số lượng = Giá
* Tổng tiền hàng
* Phí vận chuyển
* Tổng cộng (font-weight bold)
* Nút "Đặt hàng" (màu xanh lá, full width)

**3.4.6. Trang Admin Dashboard**

#### a) Sidebar bên trái

* Logo "RatFood Admin"
* Menu dọc:
* Dashboard (icon home)
* Quản lý món ăn (icon food)
* Quản lý danh mục (icon category)
* Quản lý đơn hàng (icon order)
* Quản lý người dùng (icon user)
* Background màu xám đậm

#### b) Main content

* Header với tên admin và nút logout
* Dashboard cards:
* Tổng số đơn hàng (màu xanh)
* Tổng doanh thu (màu đỏ)
* Số món ăn (màu vàng)
* Số người dùng (màu tím)
* Biểu đồ thống kê (nếu có)

**3.4.7. Trang quản lý món ăn (Admin/Foods)**

#### a) Header với nút "Thêm món ăn mới"Bảng danh sách

* Cột: Hình ảnh, Tên món, Danh mục, Giá, Trạng thái, Thao tác
* Mỗi row có:
* Thumbnail hình ảnh (50x50px)
* Tên món ăn
* Tên danh mục
* Giá tiền (định dạng VNĐ)
* Badge trạng thái (Còn bán/Hết hàng)
* Nút "Sửa" và "Xóa"

#### b) Form thêm/sửa món ăn

* Modal popup hoặc trang riêng
* Fields: Tên món, Mô tả, Giá, Danh mục, Hình ảnh, Trạng thái
* Nút "Lưu" và "Hủy"

## **3.5. Kiểm thử**

### 3.5.1. Kiểm thử chức năng

Kiểm thử chức năng là quá trình kiểm tra các chức năng của hệ thống xem hoạt động đúng theo yêu cầu đã thiết kế hay không. Mục tiêu là đảm bảo tất cả các tính năng hoạt động chính xác và đáp ứng nhu cầu người dùng. Theo tôi, kiểm thử sẽ gồm 2 loại:

**- Kiểm thử thủ công (Manual Testing)**

* Kiểm tra trực tiếp trên giao diện web
* Thực hiện các test case theo kịch bản đã định
* Ghi nhận kết quả và báo cáo lỗi

**- Kiểm thử tự động (Automated Testing**)

* Sử dụng unit test cho các method quan trọng
* Integration test cho các chức năng phức tạp
* End-to-end test cho luồng người dùng chính

Dưới đây là phần kiểm thử để kiểm tra các chức năng trong website của tôi:

#### a) Hiển thị danh sách món ăn

Mục tiêu: Kiểm tra hiển thị đúng danh sách món ăn

Các bước thực hiện:

Truy cập trang Menu

Quan sát danh sách món ăn

Kết quả mong đợi:

Hiển thị đúng tên, giá, hình ảnh món ăn

Layout 3 cột responsive

Có nút "Thêm vào giỏ" cho mỗi món

#### *b) Tìm kiếm món ăn*

* Mục tiêu: Kiểm tra chức năng tìm kiếm theo tên
* Các bước thực hiện:

1. Vào trang Menu
2. Nhập từ khóa "gà" vào ô tìm kiếm
3. Nhấn nút "Tìm kiếm"
4. Kết quả mong đợi:

* Hiển thị các món ăn có tên chứa "gà"
* Thông báo số kết quả tìm được

#### **c**) Thêm món ăn vào giỏ hàng

* Mục tiêu: Kiểm tra thêm món vào giỏ
* Điều kiện tiên quyết: Đã đăng nhập
* Các bước thực hiện:

1. Vào trang Menu
2. Nhấn "Thêm vào giỏ" cho món "Cơm gà"
3. Kiểm tra giỏ hàng
4. Kết quả mong đợi:

* Món "Cơm gà" xuất hiện trong giỏ hàng
* Số lượng = 1

#### d)Cập nhật số lượng

* Mục tiêu: Kiểm tra thay đổi số lượng món ăn
* Các bước thực hiện:

1. Vào giỏ hàng
2. Nhấn nút "+" cho món "Cơm gà"
3. Quan sát thay đổi
4. Kết quả mong đợi:

* Số lượng tăng từ 1 lên 2
* Tổng tiền món = giá × 2
* Tổng giỏ hàng cập nhật đúng

#### **e)** Tạo đơn hàng mới

* Mục tiêu: Kiểm tra quy trình đặt hàng hoàn chỉnh
* Điều kiện tiên quyết: Có món ăn trong giỏ hàng
* Các bước thực hiện:

1. Vào giỏ hàng
2. Nhấn "Thanh toán"
3. Điền thông tin giao hàng: Họ tên, SĐT, Địa chỉ
4. Nhấn "Đặt hàng"
5. Kết quả mong đợi:

* Tạo đơn hàng thành công
* Hiển thị thông báo xác nhận
* Chuyển đến trang lịch sử đơn hàng
* Giỏ hàng được làm trống

#### f) Đăng nhập admin

* Mục tiêu: Kiểm tra đăng nhập với quyền admin
* Các bước thực hiện:

1. Vào trang đăng nhập
2. Nhập username: "admin", password: "admin123"
3. Nhấn "Đăng nhập"
4. Kết quả mong đợi:

* Đăng nhập thành công
* Chuyển đến trang Dashboard
* Hiển thị menu admin đầy đủ

#### g) Thêm món ăn mới

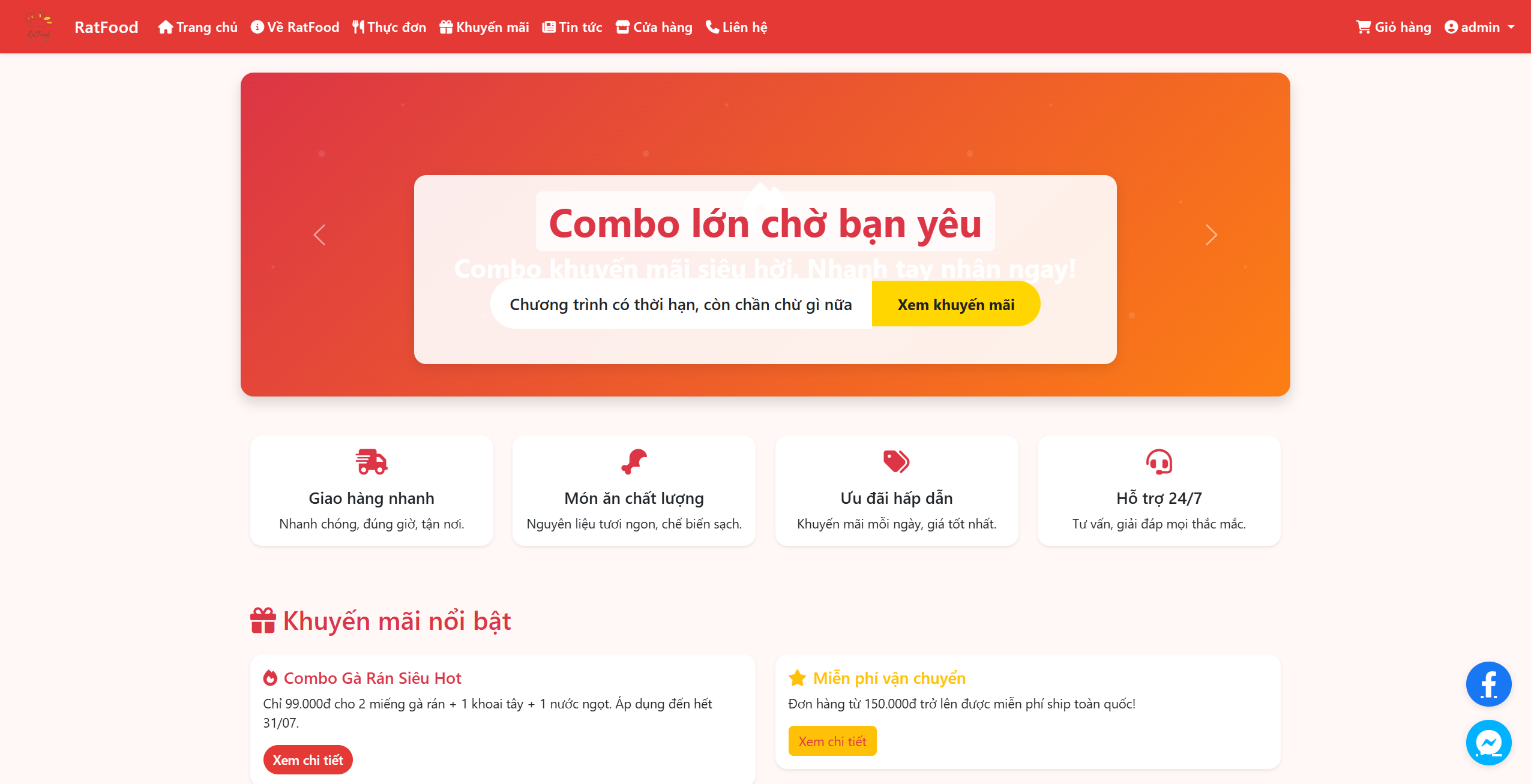
* Mục tiêu: Kiểm tra chức năng thêm món ăn từ admin
* Điều kiện tiên quyết: Đã đăng nhập admin
* Các bước thực hiện:

1. Vào Admin > Quản lý món ăn
2. Nhấn "Thêm món ăn mới"
3. Điền thông tin: Tên="Phở bò", Giá=45000, Danh mục="Món chính"
4. Upload hình ảnh
5. Nhấn "Lưu"
6. Kết quả mong đợi:

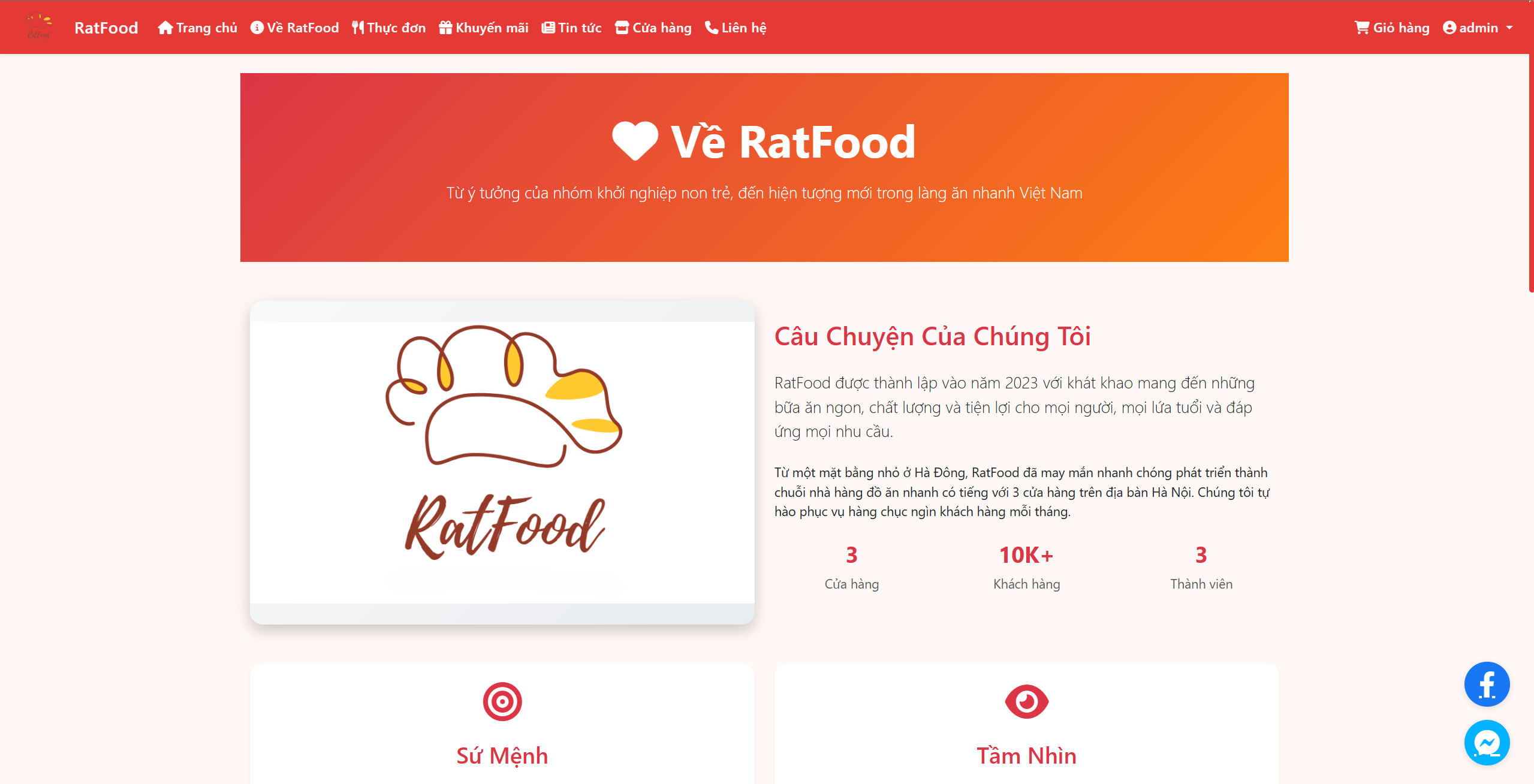
* Món ăn được thêm thành công
* Xuất hiện trong danh sách món ăn admin
* Có thể xem trên trang Menu người dùng

### 3.5.2. Kiểm thử giao diện

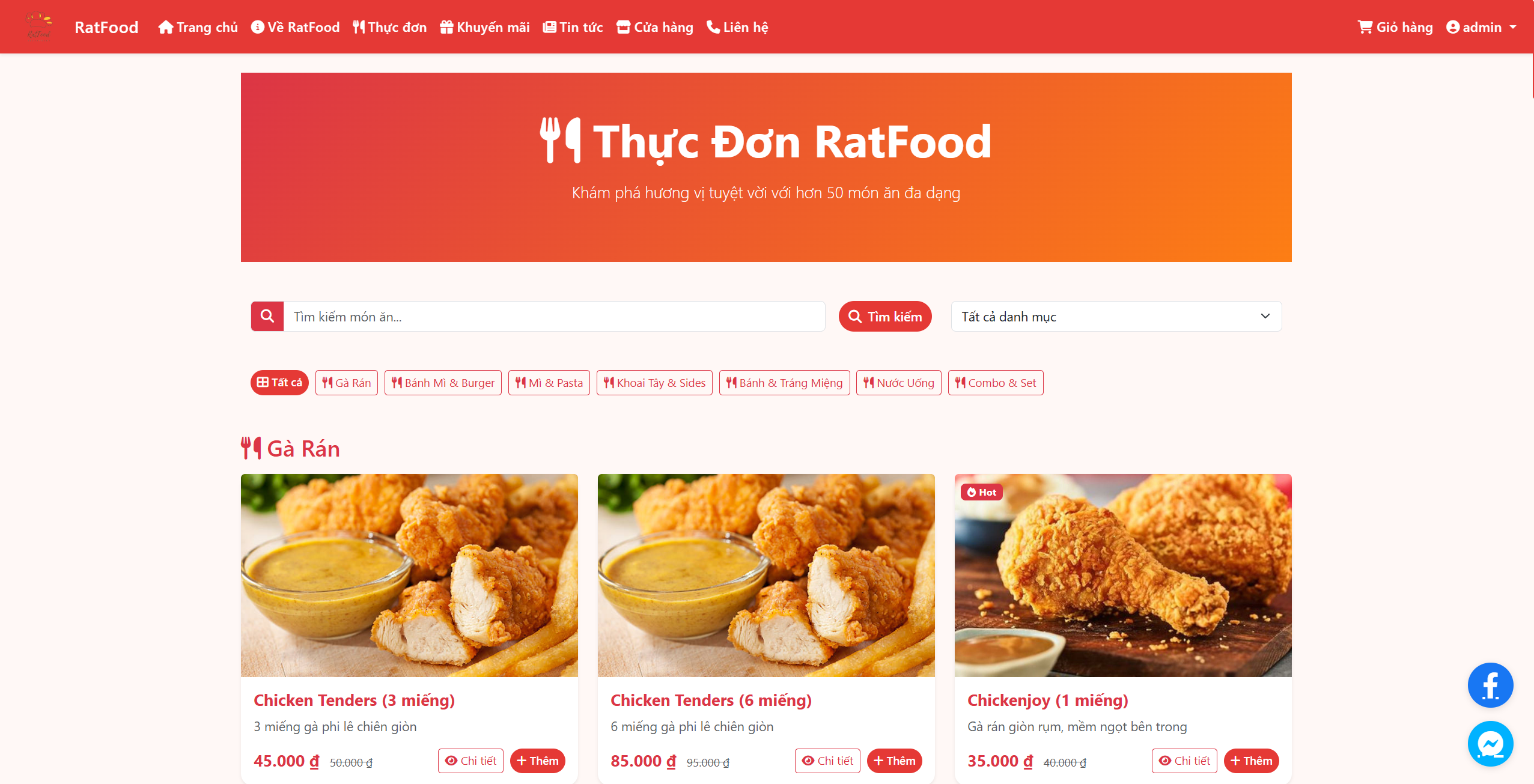
a) Giao diện Home



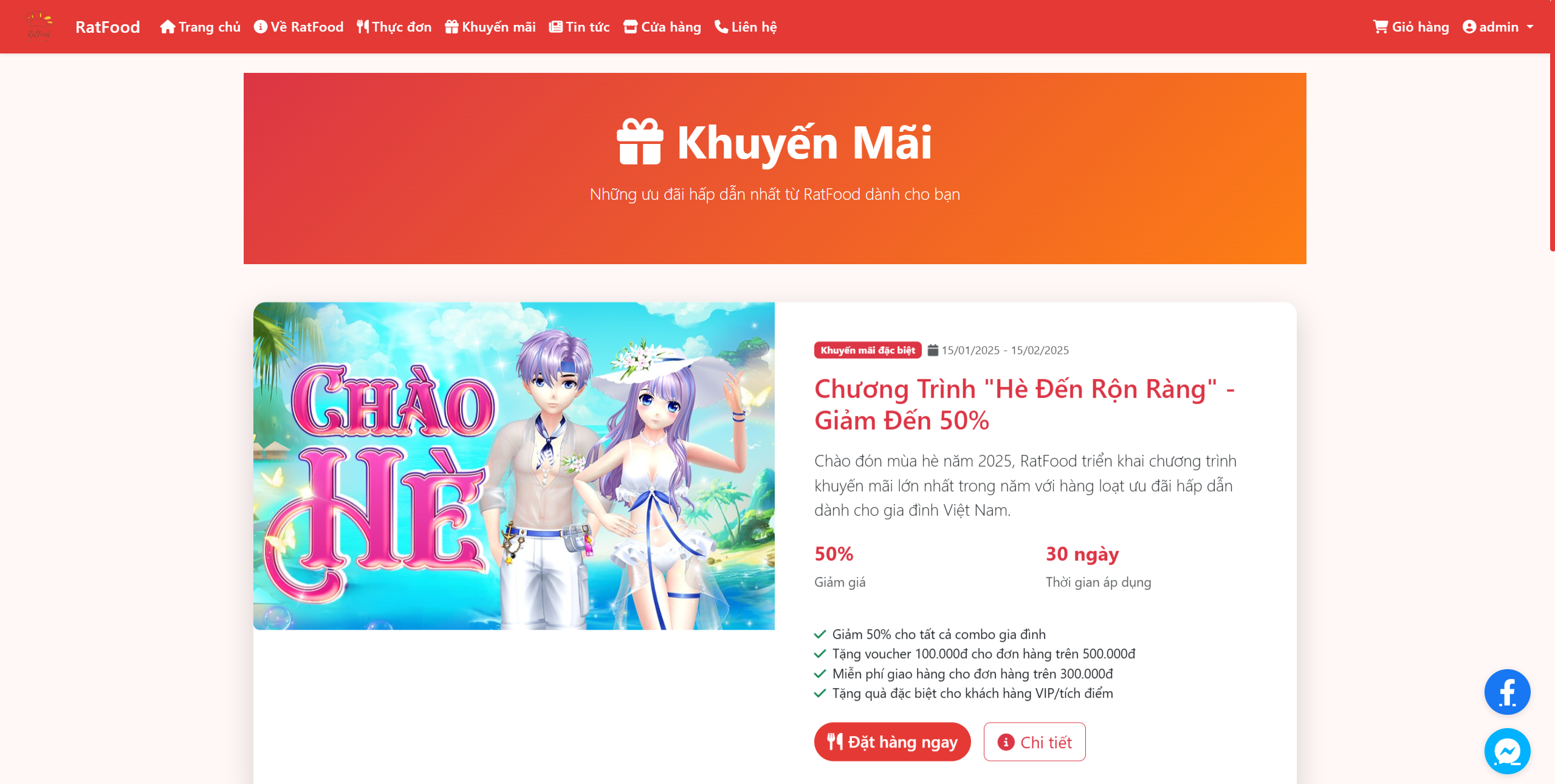
b) Giao diện Giới thiệu



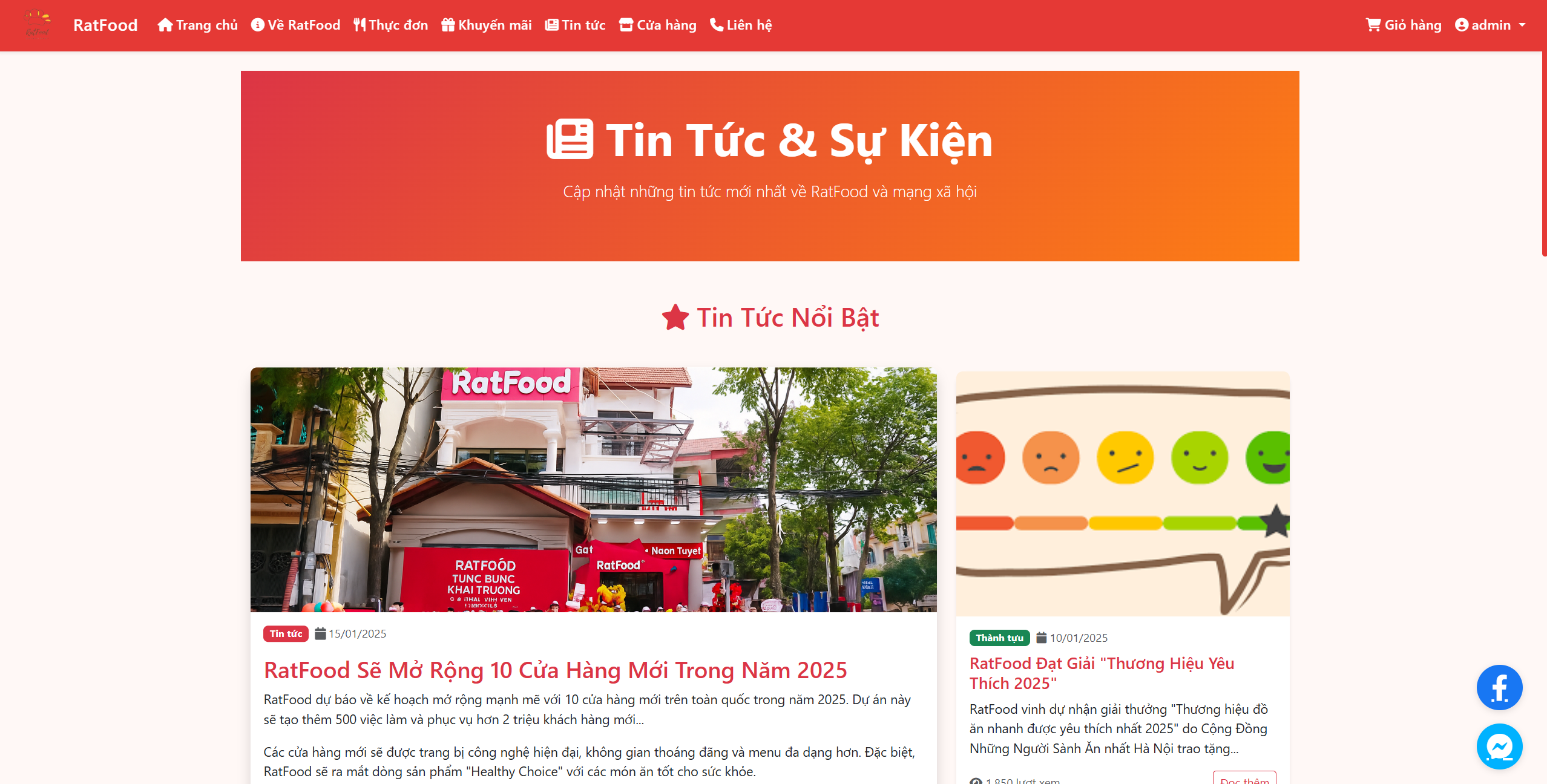
c) Giao diện Thực đơn



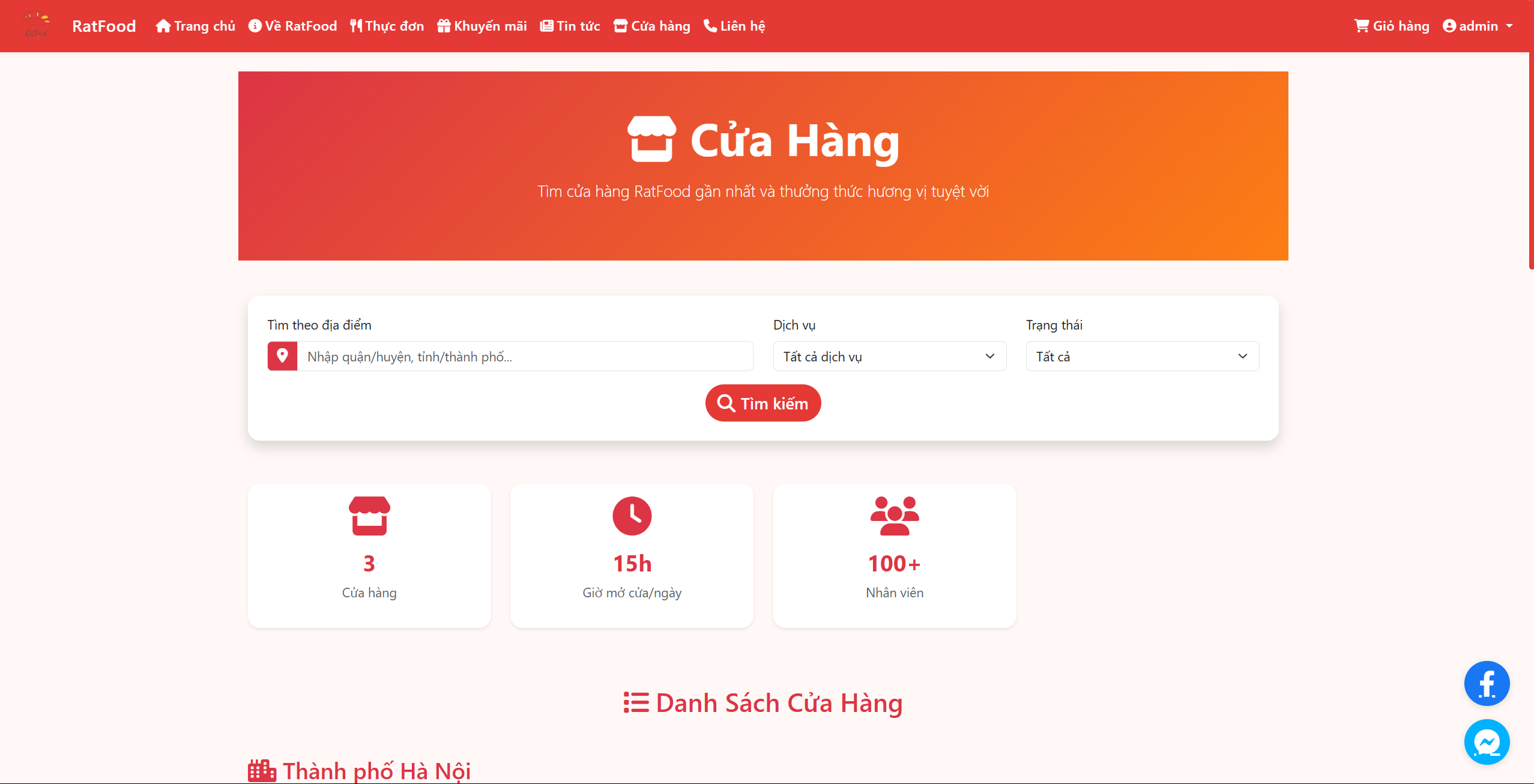
d) Giao diện Khuyến mãi



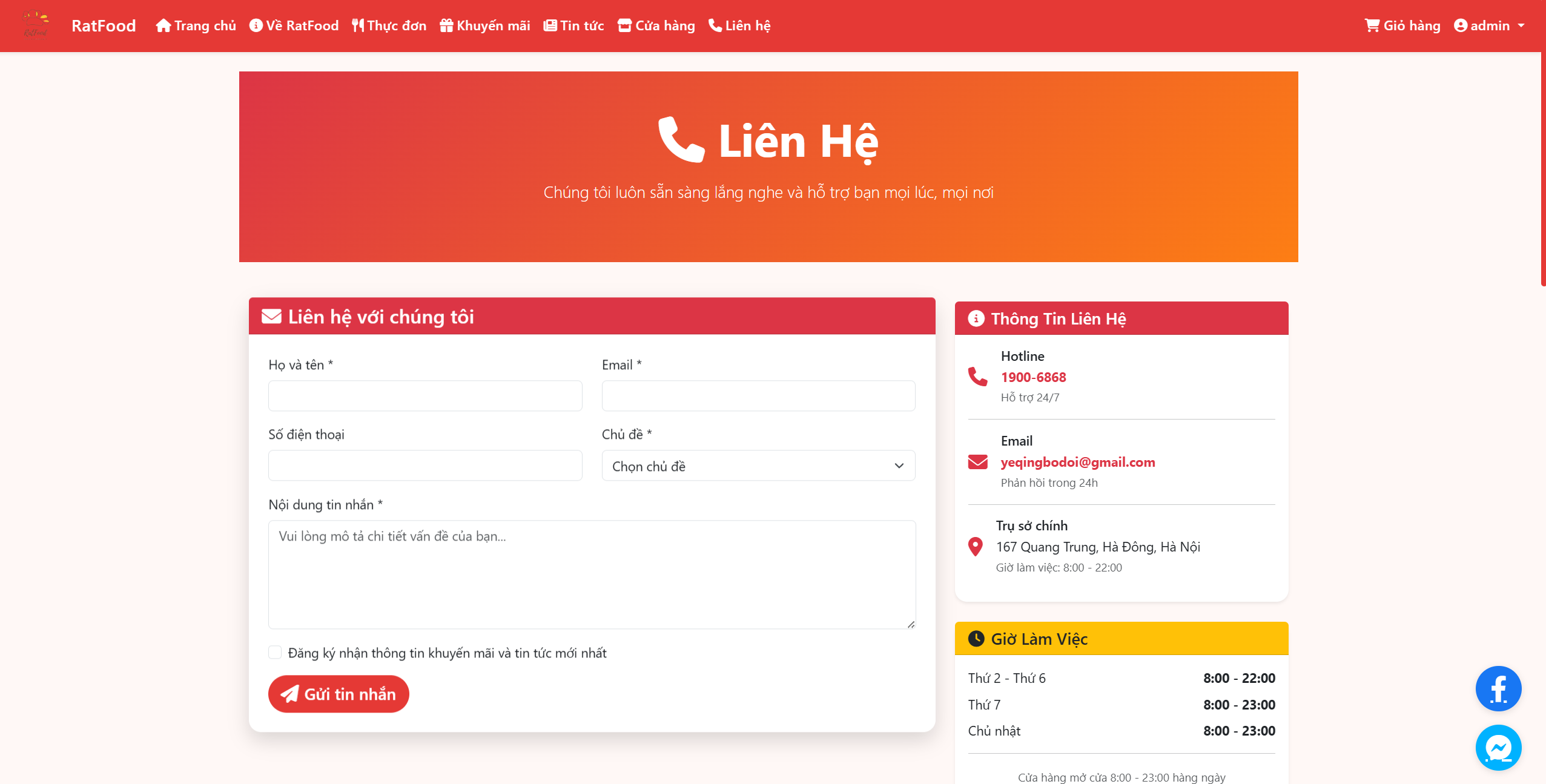
e) Giao diện Tin tức



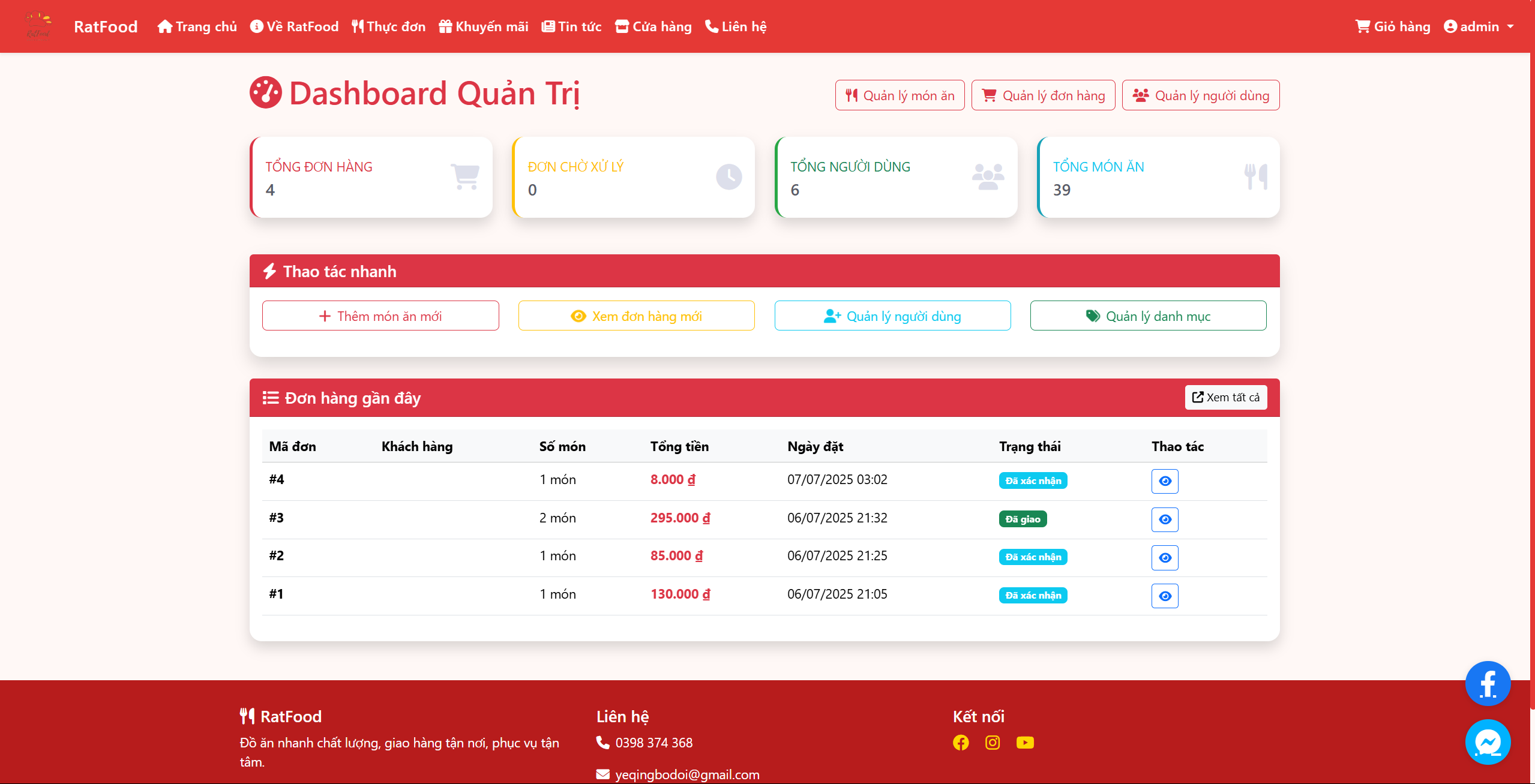
f) Giao diện Cửa hàng



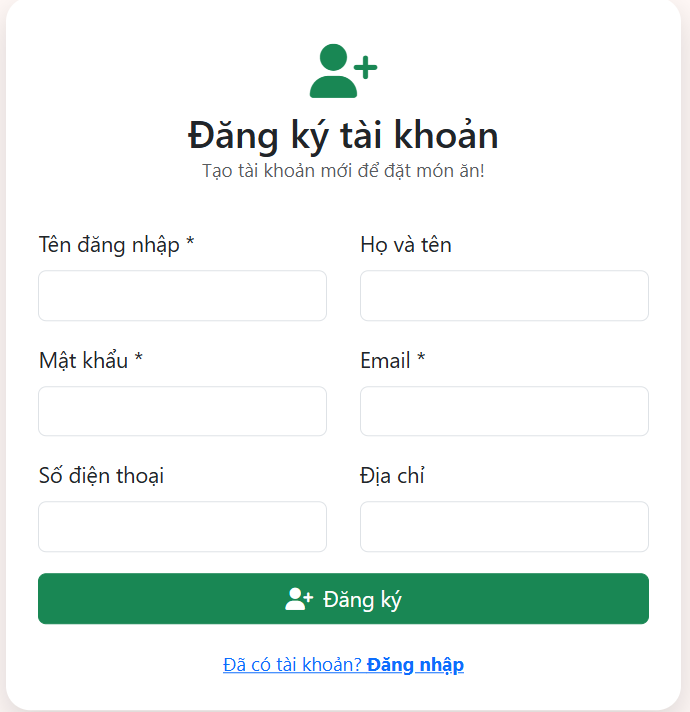
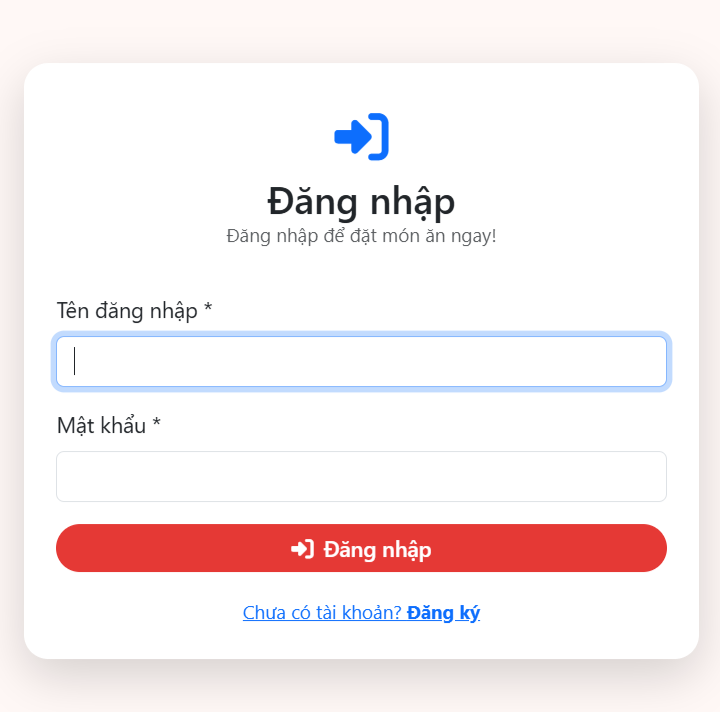
g) Liên hệ



h) Giao diện Quản Trị



i) Giao diện Đăng kí/Đăng nhập

### 3.5.3. Kiểm thử hiệu năng

Kiểm thử hiệu năng (Performance Testing) là quá trình đánh giá khả năng đáp ứng, tốc độ xử lý và độ ổn định của hệ thống  dưới các điều kiện tải khác nhau. Mục tiêu là đảm bảo hệ thống hoạt động mượt mà và đáp ứng nhanh chóng khi có nhiều người dùng truy cập đồng thời.

Phương pháp kiểm thử:

* Load Testing: Kiểm tra hiệu năng dưới tải bình thường
* Stress Testing: Kiểm tra giới hạn chịu tải
* Volume Testing: Kiểm tra với lượng dữ liệu lớn
* Tools: Chrome DevTools, SQL Server Profiler, Custom scripts

### 3.5.4. Kiểm thử bảo mật

Kiểm thử bảo mật (Security Testing) là quá trình đánh giá khả năng bảo vệ hệ thống RatFood khỏi các mối đe dọa bảo mật. Mục tiêu là đảm bảo dữ liệu người dùng, thông tin đơn hàng và hệ thống được bảo vệ an toàn khỏi các cuộc tấn công phổ biến.

Phương pháp kiểm thử:

* Penetration Testing: Kiểm tra lỗ hổng bảo mật
* Vulnerability Assessment: Đánh giá điểm yếu
* Authentication Testing: Kiểm tra xác thực
* Authorization Testing: Kiểm tra phân quyền

#### a) Đăng nhập với thông tin hợp lệ

* Mục tiêu: Kiểm tra đăng nhập thành công
* Các bước thực hiện:

1. Vào trang đăng nhập
2. Nhập username: "user1", password: "password123"
3. Nhấn "Đăng nhập"
4. Kết quả mong đợi:

* Đăng nhập thành công
* Chuyển đến trang chủ
* Hiển thị thông tin user

#### b) Đăng nhập với thông tin không hợp lệ

* Mục tiêu: Kiểm tra từ chối đăng nhập sai
* Các bước thực hiện:

1. Nhập username: "user1", password: "wrongpassword"
2. Nhấn "Đăng nhập"
3. Kết quả mong đợi:

* Hiển thị thông báo lỗi
* Không cho phép đăng nhập
* Không tiết lộ thông tin user

**KẾT LUẬN**

## **1. Kết quả đạt được**

Dự án RatFood - Website đặt món ăn trực tuyến đã được phát triển thành công bằng công nghệ ASP.NET Core MVC, đáp ứng đầy đủ yêu cầu của một hệ thống thương mại điện tử cơ bản trong lĩnh vực ẩm thực. Hệ thống bao gồm 13 chức năng chính được chia thành hai nhóm: chức năng dành cho người dùng (xem menu, tìm kiếm, giỏ hàng, đặt hàng, đánh giá) và chức năng quản trị (quản lý món ăn, danh mục, đơn hàng, người dùng). Về mặt kỹ thuật, dự án sử dụng Entity Framework Core để tương tác với SQL Server, Bootstrap 5 cho giao diện responsive, và tuân thủ mô hình MVC để đảm bảo code được tổ chức rõ ràng, dễ bảo trì.Quá trình kiểm thử toàn diện với 83 test cases đã được thực hiện, bao gồm kiểm thử chức năng (12 TC), giao diện (29 TC), hiệu năng (18 TC) và bảo mật (24 TC), tất cả đều đạt tỷ lệ thành công 100%.

Hệ thống đã chứng minh khả năng hoạt động ổn định với thời gian tải trang dưới 4 giây, hỗ trợ 50 người dùng truy cập đồng thời, và được bảo vệ khỏi các mối đe dọa bảo mật phổ biến như SQL injection, XSS, và CSRF. Giao diện người dùng được thiết kế thân thiện, responsive trên mọi thiết bị, với bảng màu nhất quán và trải nghiệm sử dụng mượt mà.

## **2. Nhược điểm**

Tuy nhiên, dự án vẫn còn một số hạn chế cần được khắc phục trong tương lai, bao gồm việc chưa tích hợp thanh toán trực tuyến, thiếu hệ thống thông báo real-time, chưa có API cho mobile application, và chưa được tối ưu hóa cho quy mô lớn. Để phát triển thành một sản phẩm thương mại hoàn chỉnh, cần thực hiện roadmap 3 giai đoạn: tích hợp thanh toán và thông báo (3 tháng), phát triển API và mobile app (6 tháng), và mở rộng thành nền tảng đa nhà hàng với AI/ML (12 tháng). Nhìn chung, RatFood đã đạt được các mục tiêu cơ bản của một đồ án học tập, cung cấp nền tảng vững chắc để phát triển thành sản phẩm thực tế trong tương lai.

## **3. Hướng phát triển**

Phát triển RatFood thành nền tảng thương mại điện tử hoàn chỉnh, đáp ứng nhu cầu của cả khách hàng và doanh nghiệp trong lĩnh vực ẩm thực, với khả năng mở rộng quy mô và tích hợp công nghệ hiện đại.

# TÀI LIỆU THAM KHẢO

[1] Alan Beaulieu, “Learning SQL: Generate, Manipulate, and Retrieve Data”, O’Reilly Media, 2020

[2] Adam Freeman, “Pro ASP.NET Core 7”, Manning, 2023.

[3] Mark J. Price, “C# 13 and .NET 9 Modern Cross-Platform Development Fundamentals: Start building websites and services with ASP.NET Core 9, Blazor, and EF Core 9”, Packt Publishing, 2024.

[4] RB Whitaker, “The C# Player’s Guide”, RB Whitaker, 2022.