**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 | 1771020599 | Nguyễn Minh Sơn | CNTT 17-01 |

**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 | 1771020599 | Nguyễn Minh Sơn | 13/11/2005 |  |  |

**CÁN BỘ CHẤM THI**

**Hà Nội, năm 2025**

**LỜI NÓI ĐẦU**

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ trên toàn cầu, việc áp dụng công nghệ thông tin vào quản lý, vận hành và kinh doanh đang trở thành xu thế tất yếu ở hầu hết các lĩnh vực, trong đó có ngành dịch vụ ăn uống – một lĩnh vực có tính cạnh tranh cao và yêu cầu ngày càng khắt khe về tốc độ phục vụ cũng như trải nghiệm khách hàng. Với sự phát triển vượt bậc của thương mại điện tử, việc xây dựng một hệ thống website bán hàng trực tuyến không chỉ giúp doanh nghiệp tiếp cận người tiêu dùng hiệu quả mà còn góp phần nâng cao năng lực quản lý, tiết kiệm chi phí vận hành và tối ưu hóa nguồn lực.

Xuất phát từ thực tế đó, em lựa chọn thực hiện đề tài: “Phát triển Website cho cửa hàng bán đồ ăn nhanh sử dụng ASP.NET Core MVC” nhằm hiện thực hóa một mô hình hệ thống bán hàng trực tuyến đơn giản, hiệu quả và có tính ứng dụng cao. Thông qua website này, khách hàng có thể dễ dàng truy cập để xem thực đơn, chọn món ăn, đặt hàng và theo dõi đơn hàng một cách tiện lợi. Đồng thời, hệ thống cũng cung cấp giao diện quản trị giúp chủ cửa hàng quản lý sản phẩm, đơn hàng và người dùng một cách khoa học và nhanh chóng.

Quá trình thực hiện đề tài là một cơ hội để em củng cố và vận dụng kiến thức đã học trong chương trình đào tạo, đặc biệt là về lập trình web, thiết kế hệ thống, cơ sở dữ liệu và mô hình MVC. Em sử dụng ASP.NET Core MVC làm nền tảng phát triển ứng dụng với phương pháp Code First của Entity Framework nhằm tối ưu quy trình xây dựng cơ sở dữ liệu và đảm bảo tính linh hoạt cho việc mở rộng hệ thống về sau.

Bên cạnh những kết quả đạt được, trong quá trình thực hiện đề tài, em cũng gặp không ít khó khăn như việc thiết kế giao diện sao cho thân thiện và tối ưu trải nghiệm người dùng, xử lý logic đặt hàng - giỏ hàng, hoặc tối ưu truy vấn trong cơ sở dữ liệu. Tuy nhiên, nhờ sự hướng dẫn tận tình của quý thầy cô và quá trình tự nghiên cứu, thử nghiệm, em đã hoàn thiện đề tài với các chức năng cơ bản đáp ứng yêu cầu đặt ra ban đầu.

Mặc dù đã cố gắng hết mình trong quá trình thực hiện, nhưng do hạn chế về thời gian, kinh nghiệm thực tiễn và nguồn lực, đề tài chắc chắn vẫn còn những thiếu sót nhất định. Em rất mong nhận được sự góp ý, nhận xét chân thành từ quý thầy cô và hội đồng để em có thể tiếp tục hoàn thiện sản phẩm và tích lũy thêm kinh nghiệm cho bản thân.

**MỤC LỤC**

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 8](#_Toc202642625)

[1.1 Giới thiệu về đề tài: 8](#_Toc202642626)

[1.1.1 Bối cảnh nghiên cứu: 8](#_Toc202642627)

[1.1.2 Vấn đề cần giải quyết: 8](#_Toc202642628)

[1.2 Mục tiêu của đề tài: 8](#_Toc202642629)

[1.2.1 Mục tiêu tổng quát: 8](#_Toc202642630)

[1.2.2 Mục tiêu cụ thể: 9](#_Toc202642631)

[1.3 Phạm vi của đề tài: 9](#_Toc202642632)

[1.3.1 Về nội dung chức năng: 9](#_Toc202642633)

[1.3.2 Về công nghệ: 9](#_Toc202642634)

[1.4 Phương pháp nghiên cứu: 10](#_Toc202642635)

[1.4.1 Phương pháp thu thập dữ liệu: 10](#_Toc202642636)

[1.4.2 Phương pháp phân tích và thiết kế hệ thống: 10](#_Toc202642637)

[1.4.3 Phương pháp kiểm thử và đánh giá: 10](#_Toc202642638)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 11](#_Toc202642639)

[2.1 Tổng quan về C#: 11](#_Toc202642640)

[2.1.1 Khái niệm: 11](#_Toc202642641)

[2.1.2 Đặc điểm: 11](#_Toc202642642)

[2.1.3 Ưu điểm: 11](#_Toc202642643)

[2.2 ASP.NET Core MVC: 11](#_Toc202642644)

[2.2.1 Giới thiệu: 11](#_Toc202642645)

[2.2.2 Kiến trúc MVC: 12](#_Toc202642646)

[2.2.3 Ưu điểm và nhược điểm: 12](#_Toc202642647)

[2.3 SQL Server: 12](#_Toc202642648)

[2.3.1 Giới thiệu: 12](#_Toc202642649)

[2.3.2 Đặc trưng: 12](#_Toc202642650)

[2.3.3 Ưu điểm: 12](#_Toc202642651)

[2.4 Entity Framework Core: 13](#_Toc202642652)

[2.4.1 Giới thiệu: 13](#_Toc202642653)

[2.4.2 Tính năng nổi bật: 13](#_Toc202642654)

[2.4.3 Ưu điểm: 13](#_Toc202642655)

[2.5 LINQ: 13](#_Toc202642656)

[2.5.1 Giới thiệu: 13](#_Toc202642657)

[2.5.2 Cú pháp truy vấn: 13](#_Toc202642658)

[2.5.3 Ưu điểm: 14](#_Toc202642659)

[CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG 15](#_Toc202642660)

[3.1 Phân tích đặc tả yêu cầu: 15](#_Toc202642661)

[3.1.1 Yêu cầu chức năng: 15](#_Toc202642662)

[3.1.2 Yêu cầu phi chức năng: 15](#_Toc202642663)

[3.2 Thiết kế hệ thống: 15](#_Toc202642664)

[3.2.1 Usecase Diagram: 15](#_Toc202642665)

[3.2.2 Class Diagram: 16](#_Toc202642666)

[3.2.3 Activity Diagram: 17](#_Toc202642667)

[3.2.4 Sequence Diagram: 18](#_Toc202642668)

[3.3 Thiết kế cơ sở dữ liệu: 19](#_Toc202642669)

[3.3.1 Thực thể Order: 19](#_Toc202642670)

[3.3.2 Cấu hình trong OnModelCreating: 20](#_Toc202642671)

[3.3.3 Chiến lược Migration & Seed: 21](#_Toc202642672)

[3.4 Thiết kế giao diện: 21](#_Toc202642673)

[3.4.1 Giao diện người dùng: 21](#_Toc202642674)

[3.4.2 Giao diện Admin: 25](#_Toc202642675)

[3.5 Kiểm thử: 27](#_Toc202642676)

[3.5.1 Kiểm thử chức năng: 27](#_Toc202642677)

[3.5.2 Kiểm thử giao diện: 28](#_Toc202642678)

[3.5.3 Kiểm thử hiệu năng: 28](#_Toc202642679)

[3.5.4 Kiểm thử bảo mật: 28](#_Toc202642680)

**DANH MỤC HÌNH ẢNH**

[Hình 3.1. Use case Diagram 16](#_Toc202635021)

[Hình 3.2. Class Diagram 17](#_Toc202635022)

[Hình 3.3. Activity Diagram 18](#_Toc202635023)

[Hình 3.4. Sequence Diagram 19](#_Toc202635024)

[Hình 3.5. Trang chủ 22](#_Toc202635025)

[Hình 3.6. Đăng ký 22](#_Toc202635026)

[Hình 3.7. Đăng nhập 23](#_Toc202635027)

[Hình 3.8. Thực đơn 23](#_Toc202635028)

[Hình 3.9. Chi tiết món ăn 24](#_Toc202635029)

[Hình 3.10. Giỏ hàng 24](#_Toc202635030)

[Hình 3.11. Đặt hàng 25](#_Toc202635031)

[Hình 3.12. Bảng điều khiển 26](#_Toc202635032)

[Hình 3.13. Quản lý món ăn 26](#_Toc202635033)

[Hình 3.14. Quản lý đơn hàng 27](#_Toc202635034)

# TỔNG QUAN VỀ ĐỀ TÀI

## Giới thiệu về đề tài:

### Bối cảnh nghiên cứu:

Trong thời đại công nghệ số bùng nổ, nhu cầu mua sắm trực tuyến đang ngày càng gia tăng mạnh mẽ, đặc biệt là trong lĩnh vực ăn uống. Với nhịp sống bận rộn, người tiêu dùng có xu hướng lựa chọn các hình thức đặt món ăn nhanh qua website hoặc ứng dụng di động để tiết kiệm thời gian và công sức. Bên cạnh đó, đại dịch COVID-19 cũng đã góp phần thúc đẩy xu hướng tiêu dùng không tiếp xúc, trong đó đặt hàng và giao đồ ăn trực tuyến trở thành lựa chọn phổ biến.

Đối với các cửa hàng bán đồ ăn nhanh, việc xây dựng một website hiện đại không chỉ giúp tiếp cận khách hàng rộng rãi hơn, mà còn giúp tối ưu quy trình đặt hàng, thanh toán và quản lý đơn hàng một cách hiệu quả. Website trở thành kênh bán hàng quan trọng, song song với hình thức bán tại chỗ.

### Vấn đề cần giải quyết:

Hiện nay, nhiều cửa hàng ăn nhanh vẫn sử dụng phương pháp đặt hàng truyền thống như gọi điện thoại, nhắn tin hoặc đặt trực tiếp tại quầy. Những phương thức này tồn tại nhiều hạn chế như: dễ sai sót thông tin, mất thời gian xử lý đơn hàng, khó thống kê số liệu bán hàng theo thời gian thực. Ngoài ra, một số website hiện có chưa được tối ưu về giao diện người dùng, tốc độ xử lý hoặc thiếu các tính năng quản trị chuyên biệt cho mô hình kinh doanh ăn nhanh.

Chính vì vậy, đề tài “Phát triển website cho cửa hàng bán đồ ăn nhanh” được đề xuất nhằm xây dựng một hệ thống trực tuyến hiện đại, tiện lợi, đáp ứng tốt nhu cầu đặt món của khách hàng và hỗ trợ hiệu quả công việc quản lý cho chủ cửa hàng.

Website được phát triển bằng công nghệ ASP.NET Core MVC – nền tảng lập trình mạnh mẽ và tối ưu của Microsoft – giúp đảm bảo tính bảo mật, tốc độ xử lý cao, dễ mở rộng và bảo trì. Việc áp dụng mô hình MVC giúp tổ chức mã nguồn rõ ràng, tách biệt giữa giao diện, xử lý và dữ liệu.

## Mục tiêu của đề tài:

### Mục tiêu tổng quát:

Xây dựng website đặt món ăn nhanh chuyên nghiệp cho cửa hàng, sử dụng ASP.NET Core MVC để cung cấp trải nghiệm thân thiện cho người dùng, hỗ trợ quản trị đơn hàng hiệu quả, đồng thời đáp ứng nhu cầu mở rộng kinh doanh trực tuyến.

### Mục tiêu cụ thể:

#### Về chức năng:

* Hệ thống danh mục món ăn phân loại theo nhóm (đồ ăn, đồ uống, combo).
* Tính năng đặt hàng trực tuyến, giỏ hàng và thanh toán.
* Tích hợp theo dõi đơn hàng, đánh giá sản phẩm.
* Trang quản trị cho admin (quản lý món ăn, đơn hàng, khách hàng, thống kê doanh thu).

#### Về công nghệ:

* Sử dụng ASP.NET Core MVC để đảm bảo hiệu năng và bảo mật.
* Kết hợp Bootstrap 5, jQuery để xây dựng giao diện responsive.
* Cơ sở dữ liệu: SQL Server.

#### Về trải nghiệm người dùng:

* Giao diện dễ sử dụng, hỗ trợ mobile/tablet.
* Thời gian phản hồi nhanh, thao tác đặt món mượt mà.

## Phạm vi của đề tài:

### Về nội dung chức năng:

* Thiết kế và phát triển giao diện website có bố cục hợp lý, bắt mắt, dễ thao tác.
* Các chức năng chính bao gồm:
* Xem danh sách món ăn theo danh mục.
* Tìm kiếm món ăn theo tên hoặc từ khóa.
* Xem chi tiết món ăn (mô tả, giá, hình ảnh).
* Chức năng giỏ hàng và đặt món.
* Đăng ký, đăng nhập, quản lý tài khoản người dùng.
* Quản lý đơn hàng, theo dõi tình trạng đơn.
* Trang quản trị: thêm/sửa/xóa món ăn, quản lý người dùng, xem thống kê bán hàng.

### Về công nghệ:

* Sử dụng công nghệ ASP.NET Core MVC cho backend.
* Cơ sở dữ liệu SQL Server lưu trữ thông tin món ăn, đơn hàng, tài khoản người dùng.
* Giao diện được thiết kế bằng HTML, CSS, Bootstrap, JavaScript.
* Ứng dụng triển khai trên môi trường cục bộ (localhost), có thể mở rộng triển khai online.

## Phương pháp nghiên cứu:

### Phương pháp thu thập dữ liệu:

* Tìm hiểu xu hướng tiêu dùng và hành vi đặt món online của khách hàng hiện nay.
* Khảo sát các hệ thống đặt món online phổ biến (GrabFood, Baemin, ShopeeFood) để học hỏi tính năng và giao diện.
* Tham khảo tài liệu chính thức về ASP.NET Core MVC từ Microsoft Docs và các diễn đàn lập trình.

### Phương pháp phân tích và thiết kế hệ thống:

* Phân tích nhu cầu thực tế của cửa hàng bán đồ ăn nhanh, xác định các nghiệp vụ chính.
* Thiết kế sơ đồ luồng dữ liệu, use-case, giao diện mẫu (wireframe/mockup), cơ sở dữ liệu logic.
* Áp dụng UML để biểu diễn các thành phần hệ thống (class diagram, activity diagram...).

### Phương pháp kiểm thử và đánh giá:

* Kiểm thử chức năng người dùng (đặt món, thanh toán, xem đơn).
* Kiểm thử admin (quản lý món, đơn hàng, tài khoản).
* Kiểm thử giao diện và tốc độ phản hồi trên nhiều thiết bị.
* Thu thập phản hồi từ người dùng thử nghiệm để điều chỉnh UI/UX.

# CƠ SỞ LÝ THUYẾT

## Tổng quan về C#:

### Khái niệm:

C# (phát âm là "C-sharp") là một ngôn ngữ lập trình hiện đại, hướng đối tượng do Microsoft phát triển như một phần của nền tảng .NET. C# được thiết kế để đơn giản, mạnh mẽ, an toàn và hỗ trợ lập trình hướng đối tượng toàn diện.

Với cú pháp dễ tiếp cận, tương tự C/C++ và Java, C# rất phù hợp để xây dựng các ứng dụng Windows, web, mobile và các hệ thống lớn.

### Đặc điểm:

* Lập trình hướng đối tượng: Hỗ trợ đầy đủ các khái niệm như đóng gói, kế thừa, đa hình.
* Quản lý bộ nhớ tự động: Sử dụng cơ chế Garbage Collection để thu hồi bộ nhớ không còn sử dụng.
* An toàn và bảo mật: Có hệ thống kiểm tra kiểu mạnh, kiểm tra ràng buộc tại thời gian biên dịch.
* Tích hợp dễ dàng với các công nghệ Microsoft như .NET Core, WPF, ASP.NET, Entity Framework.

### Ưu điểm:

* Cú pháp dễ đọc, dễ học, gần giống với Java và C++.
* Phù hợp với phát triển hệ thống quy mô từ nhỏ đến lớn.
* Có cộng đồng phát triển mạnh và nhiều tài nguyên học tập.

## ASP.NET Core MVC:

### Giới thiệu:

ASP.NET Core là một Open-source mới và là Cross-Platform framework để

xây dựng các ứng dụng web. Các ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên .NET framework.

.NET CORE là một framework mã nguồn mở được xây dựng và phát triển bởi

Microsoft và cộng đồng .NET trên Github. Hỗ trợ phát triển ứng dụng đa nền tảng (hỗ trợ Windows, macOS và Linux) cho ứng dụng .NET

### Kiến trúc MVC:

* Controller:
* Nhận yêu cầu từ user.
* Xử lý và xây dựng model phù hợp.
* Chuyển model sang View để hiển thị.
* View: Nhận model từ Controller để tạo giao diện phù hợp và hiển thị.
* Model: Chứa dữ liệu chia sẻ giữa Controller và View.

### Ưu điểm và nhược điểm:

#### Ưu điểm:

* Thể hiện tính chuyên nghiệp trong lập trình, phân tích thiết kế.
* Phát triển ứng dụng nhanh, dễ dàng, dễ nâng cấp, bảo trì.

#### Nhược điểm:

Đối với project nhỏ, gây cồng kềnh, tốn thời gian trong quá trình phát triển, tốn thời gian trung chuyển dữ liệu của các thành phần.

## SQL Server:

### Giới thiệu:

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft, thường được sử dụng cho các ứng dụng doanh nghiệp, thương mại điện tử và hệ thống quản lý dữ liệu lớn.

### Đặc trưng:

* Lưu trữ dữ liệu dưới dạng bảng, hỗ trợ mối quan hệ giữa các bảng.
* Hỗ trợ thực hiện các truy vấn SQL nhanh chóng.
* Có thể tạo stored procedure, trigger, function để xử lý nghiệp vụ tại cơ sở dữ liệu.
* Đảm bảo tính toàn vẹn, bảo mật và hiệu năng khi truy cập đồng thời.

### Ưu điểm:

* Hỗ trợ nhiều kiểu dữ liệu, định dạng phong phú.
* Tích hợp tốt với .NET và Entity Framework.
* Giao diện quản lý dễ sử dụng (SSMS - SQL Server Management Studio).

## Entity Framework Core:

### Giới thiệu:

Là một phiên bản mã nguồn mở, nhỏ, nhẹ, có thể mở rộng và đa nền tảng nằm

trong bộ Entity Framework.

Là một framework ORM (ánh xạ quan hệ – đối tượng) cho phép người lập trình

.NET làm việc với dữ liệu quan hệ qua các đối tượng (object).

Sử dụng LINQ để truy vấn và cập nhật dữ liệu với sự hỗ trợ của ADO.NET Data

Provider.

### Tính năng nổi bật:

* Code First: Tạo bảng từ class C# và có thể migration để cập nhật dữ liệu.
* Database First: Tạo class từ database sẵn có.
* Migration: Quản lý thay đổi database thông qua dòng lệnh.
* Query bằng LINQ: Cho phép truy vấn dữ liệu một cách trực quan.

### Ưu điểm:

* Tự động hóa việc ánh xạ dữ liệu giữa class và bảng.
* Dễ bảo trì, nâng cấp cấu trúc dữ liệu.
* Tích hợp mạnh mẽ với ASP.NET Core.

## LINQ:

### Giới thiệu:

LINQ (Language Integrated Query) là cú pháp truy vấn thống nhất trong C# và

VB.NET để truy xuất dữ liệu từ các nguồn và định dạng khác nhau. Là một mô hình ORM (Object-relational mapping) hỗ trợ lập trình CSDL hướng đối tượng mạnh mẽ.

Cú pháp đơn giản, dễ hiểu. Có thể hiển thị, binding trực tiếp lên các giao diện.

### Cú pháp truy vấn:

* Method Syntax

var doDaiToiDas = dodais.Where( d => d.length > 10);

* Query Syntax

var doDaiToiDas = from d in dodais where d.length > 10;

### Ưu điểm:

* Ngắn gọn, dễ đọc và dễ bảo trì.
* Tương thích chặt chẽ với EF Core.
* Có thể kết hợp nhiều phép toán (join, group, select, orderby...).

# XÂY DỰNG ỨNG DỤNG

## Phân tích đặc tả yêu cầu:

Website bán đồ ăn nhanh là một ứng dụng thương mại điện tử với mục tiêu cung cấp cho khách hàng một nền tảng thuận tiện để lựa chọn và đặt món ăn trực tuyến, đồng thời hỗ trợ quản lý đơn hàng, thực đơn, người dùng và các thông tin liên quan.

### Yêu cầu chức năng:

#### Đối với người dùng (khách hàng):

* Đăng ký, đăng nhập, đăng xuất.
* Xem danh sách sản phẩm (thức ăn, combo, nước uống).
* Tìm kiếm sản phẩm.
* Thêm sản phẩm vào giỏ hàng.
* Đặt hàng và theo dõi tình trạng đơn hàng.
* Quản lý thông tin tài khoản cá nhân.

#### Đối với quản trị viên (admin):

* Quản lý sản phẩm (thêm, sửa, xóa).
* Quản lý danh mục món ăn.
* Quản lý người dùng.
* Quản lý đơn hàng.
* Xem báo cáo doanh thu.

### Yêu cầu phi chức năng:

* Giao diện thân thiện, dễ sử dụng, hỗ trợ nhiều thiết bị.
* Hệ thống đảm bảo bảo mật thông tin người dùng.
* Đảm bảo hiệu năng khi nhiều người dùng truy cập đồng thời.
* Khả năng mở rộng, bảo trì và nâng cấp dễ dàng.

## Thiết kế hệ thống:

### Usecase Diagram:

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3.1. Use case Diagram

### Class Diagram:

A diagram of a network

AI-generated content may be incorrect.

Hình 3.2. Class Diagram

### Activity Diagram:

Hoạt động chính: Đặt hàng

* Người dùng đăng nhập
* Xem danh sách món ăn
* Thêm món vào giỏ
* Đặt hàng
* Xem giỏ hàng
* Hệ thống xác nhận đặt hàng
* Cập nhật trạng thái đơn hàng.

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3.3. Activity Diagram

### Sequence Diagram:

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 3.4. Sequence Diagram

## Thiết kế cơ sở dữ liệu:

Sử dụng phương pháp Code First, từ các lớp C# sinh ra bảng SQL tương ứng.

### Thực thể Order:

public enum OrderStatus { Pending=0, Shipping=1, Completed=2, Cancelled=3 }

public class Order

{

public int Id { get; set; }

[Required] public string UserId { get; set; }

public AppUser User { get; set; }

public DateTime CreatedDate { get; set; } = DateTime.UtcNow;

public OrderStatus Status { get; set; } = OrderStatus.Pending;

[Required, MaxLength(255)] public string ShipAddress { get; set; }

[Column(TypeName = "decimal(12,2)")] public decimal TotalAmount { get; set; }

public ICollection<OrderDetail> OrderDetails { get; set; } = new();

}

### Cấu hình trong OnModelCreating:

protected override void OnModelCreating(ModelBuilder builder)

{

base.OnModelCreating(builder);

builder.Entity<Category>()

.HasIndex(c => c.Name).IsUnique();

builder.Entity<Product>()

.HasIndex(p => p.Name);

builder.Entity<Product>()

.Property(p => p.Price)

.HasColumnType("decimal(10,2)");

builder.Entity<OrderDetail>()

.HasKey(od => new { od.OrderId, od.ProductId });

builder.Entity<Order>()

.Property(o => o.TotalAmount)

.HasColumnType("decimal(12,2)");

// Seed some categories

builder.Entity<Category>().HasData(

new { Id = 1, Name = "Combo", Slug = "combo", ParentId = (int?)null },

new { Id = 2, Name = "Đồ uống", Slug = "do-uong", ParentId = (int?)null }

);

}

### Chiến lược Migration & Seed:

* Add-Migration InitSchema → sinh file *.cs* migration.
* Trong file migration, kiểm tra rằng mọi cột decimal, composite key đúng cấu hình.
* Update-Database để EF tạo schema và seed danh mục gốc.

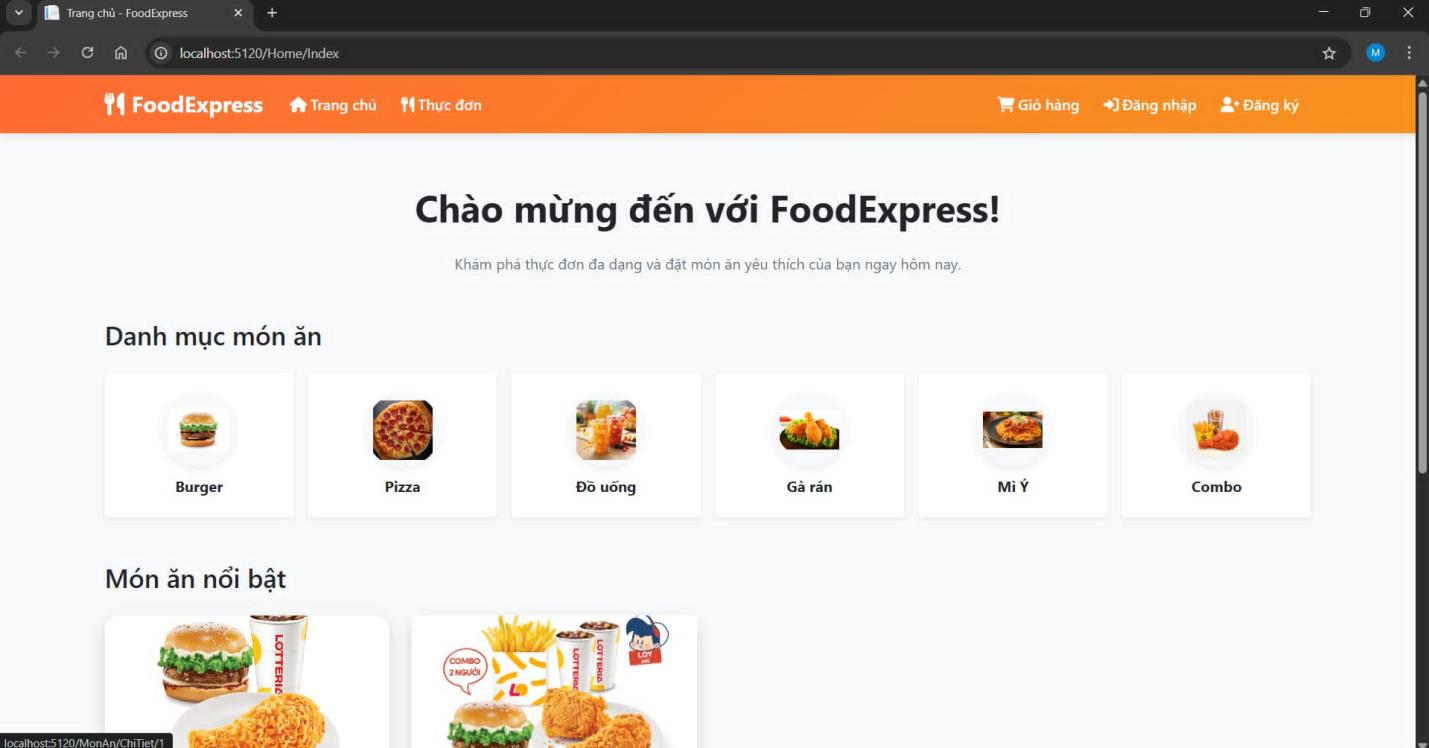
## Thiết kế giao diện:

**Link github: https://github.com/minhson11/WebFood\_BackEnd.git**

### Giao diện người dùng:

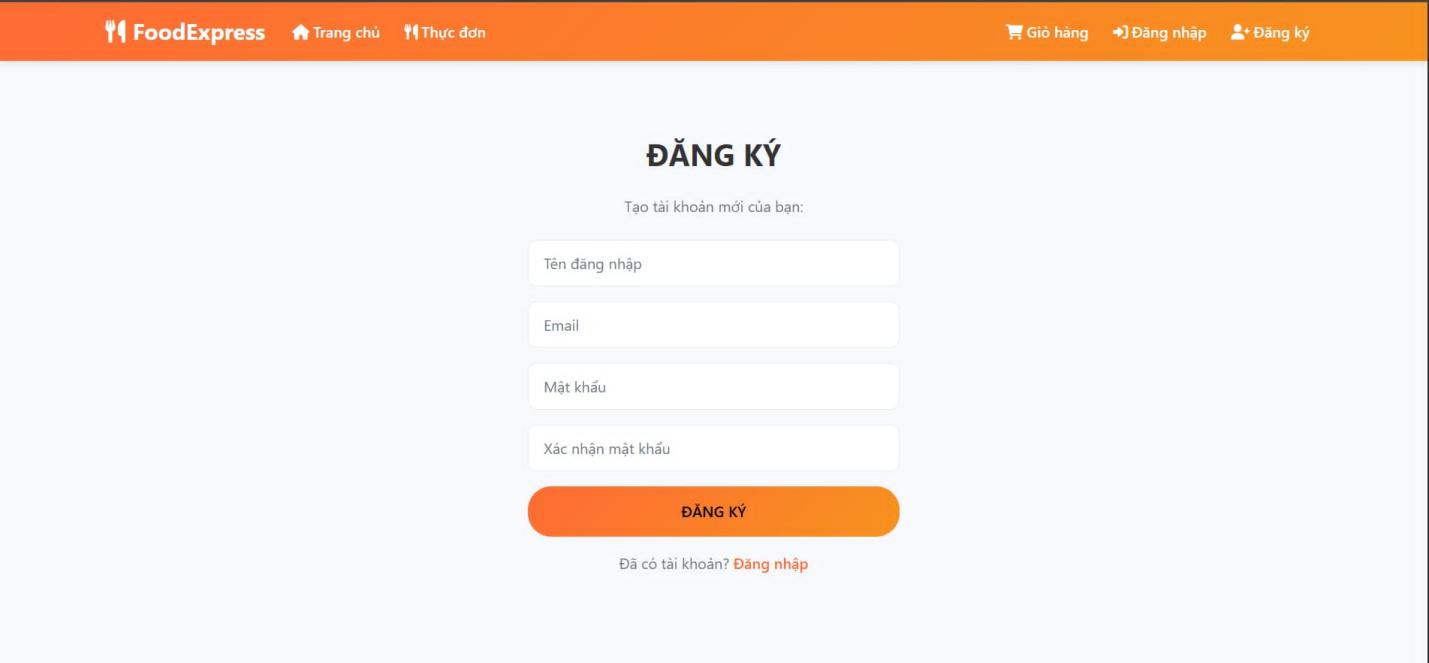
* Đăng ký/Đăng nhập
* Trang chủ: Banner + món nổi bật
* Thực đơn: Hiển thị món ăn, lọc theo danh mục
* Chi tiết món ăn: Mô tả, ảnh, thêm giỏ hàng
* Giỏ hàng: Sửa số lượng, xóa món
* Thanh toán: Nhập địa chỉ, xác nhận đơn
* Tài khoản: Cập nhật thông tin

#### Trang chủ:



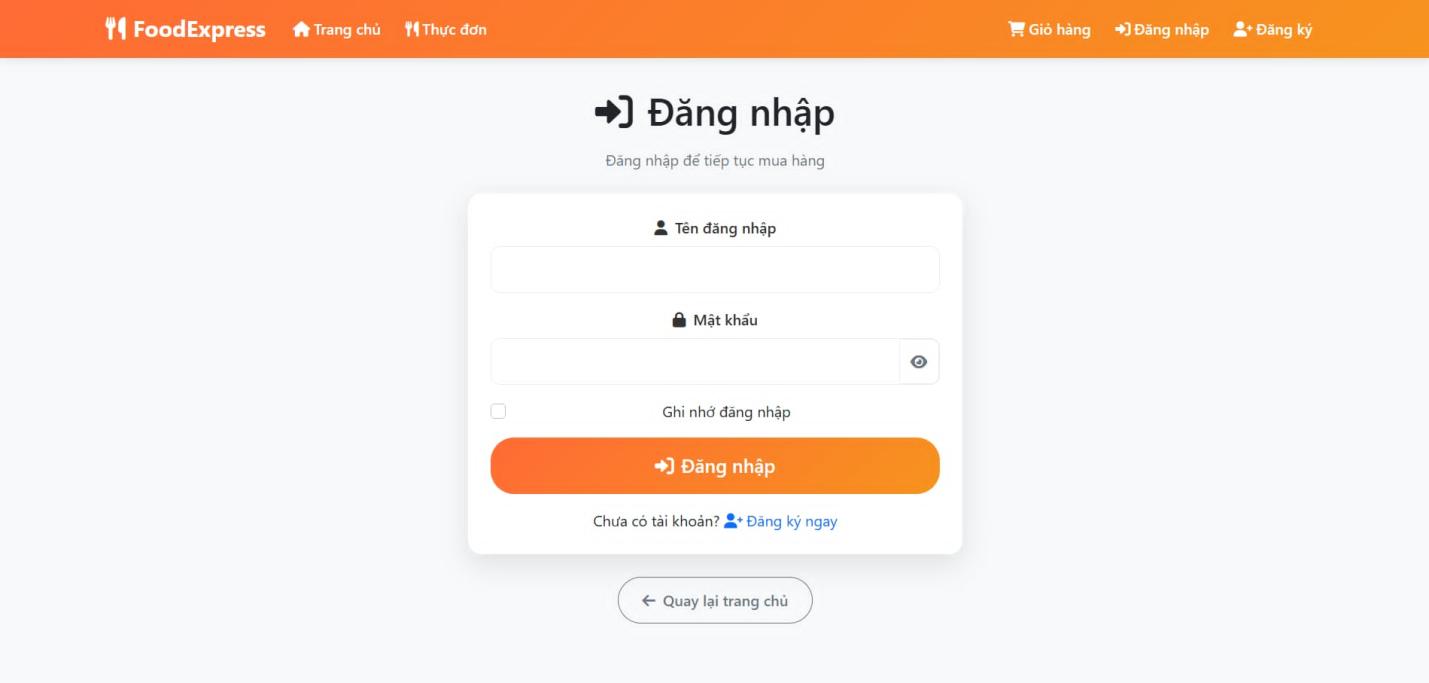
Hình 3.5. Trang chủ

#### Đăng ký:



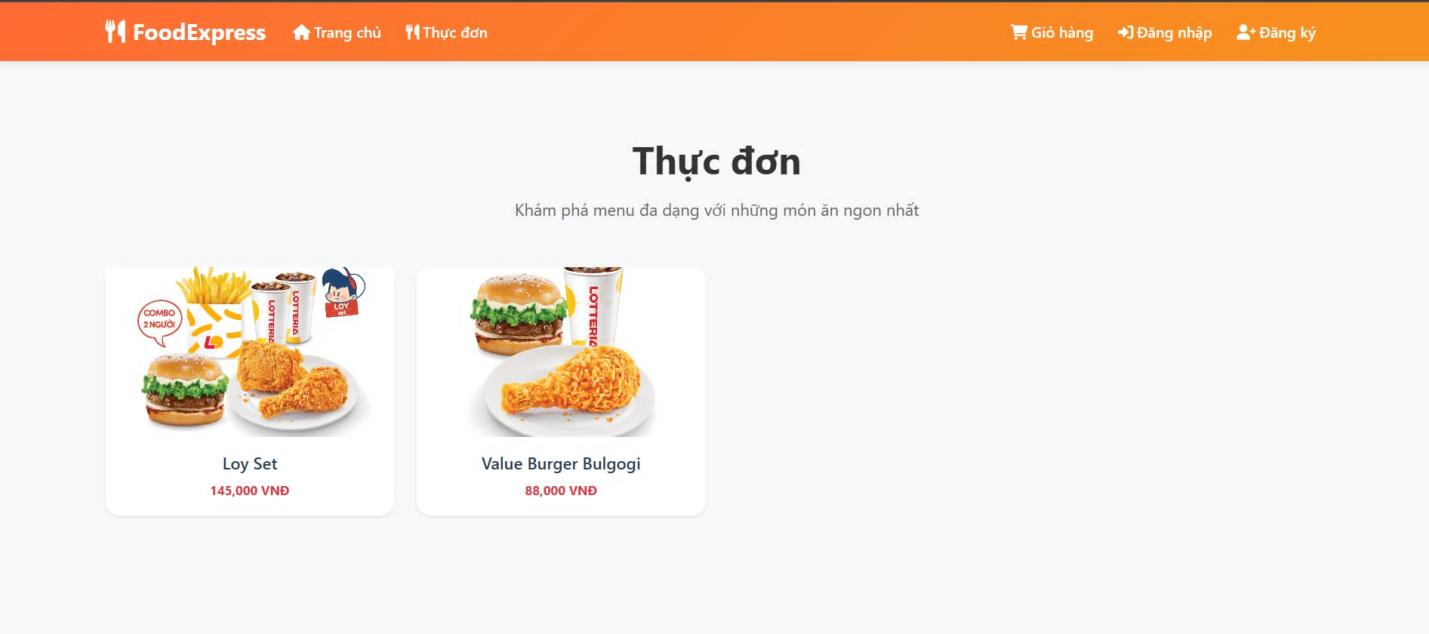
Hình 3.6. Đăng ký

#### Đăng nhập:



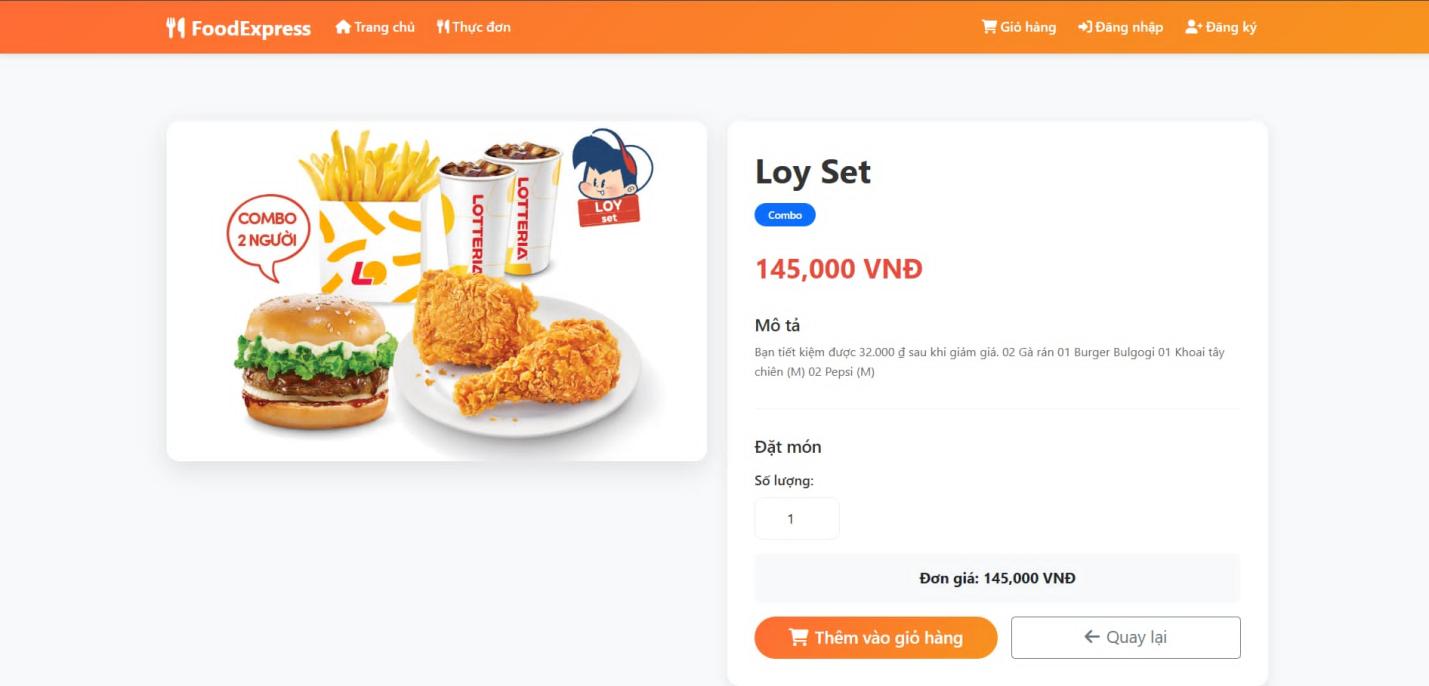
Hình 3.7. Đăng nhập

#### Thực đơn:



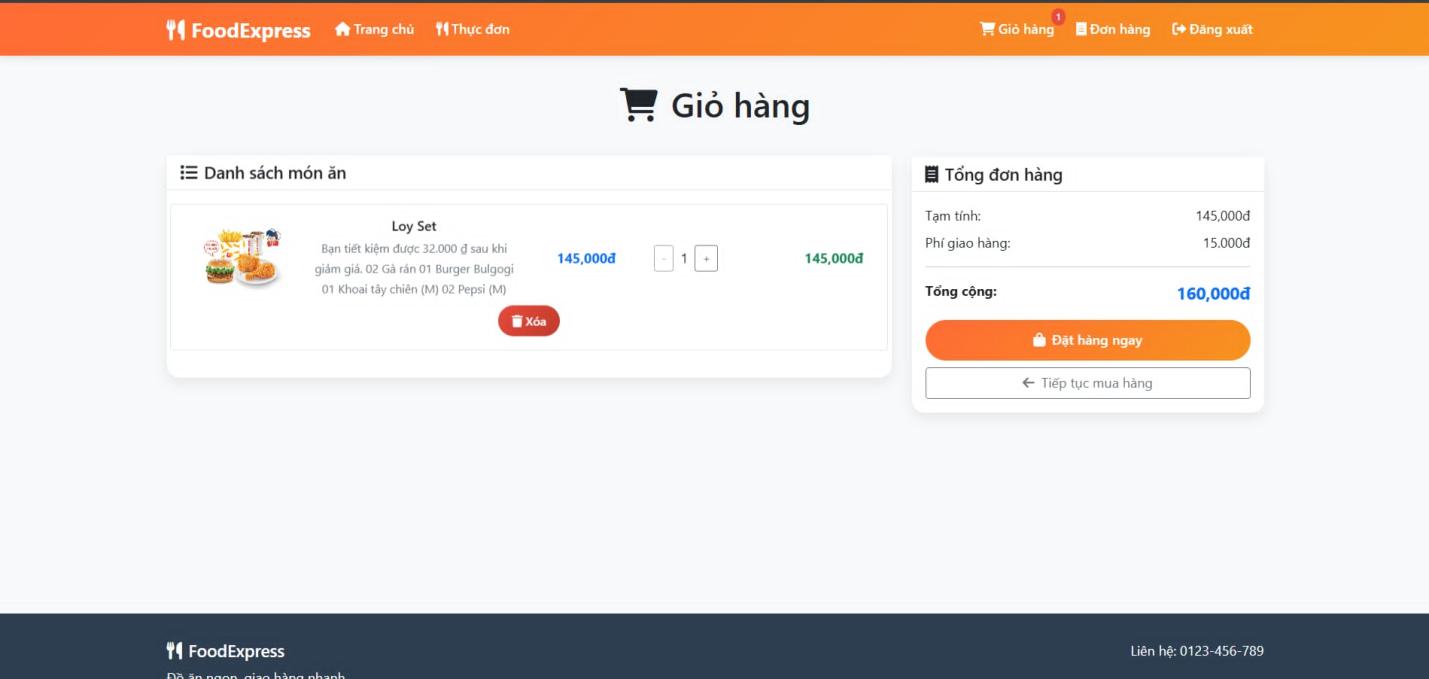
Hình 3.8. Thực đơn

#### Chi tiết món ăn:



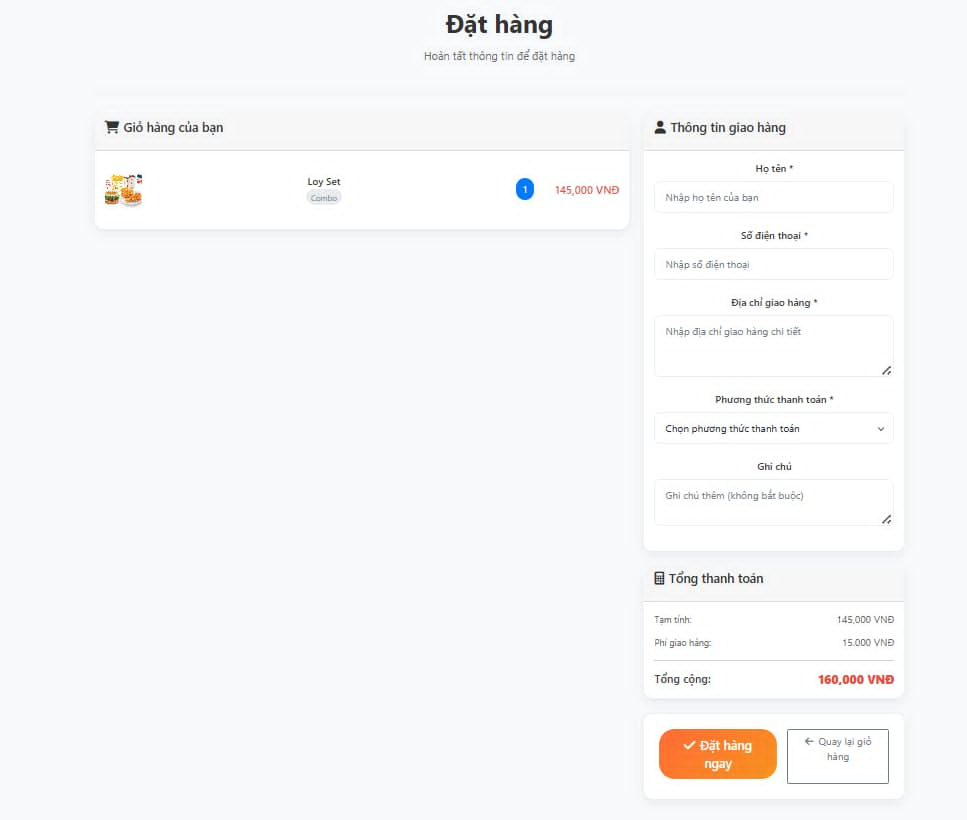
Hình 3.9. Chi tiết món ăn

#### Giỏ hàng:



Hình 3.10. Giỏ hàng

#### Đặt hàng:

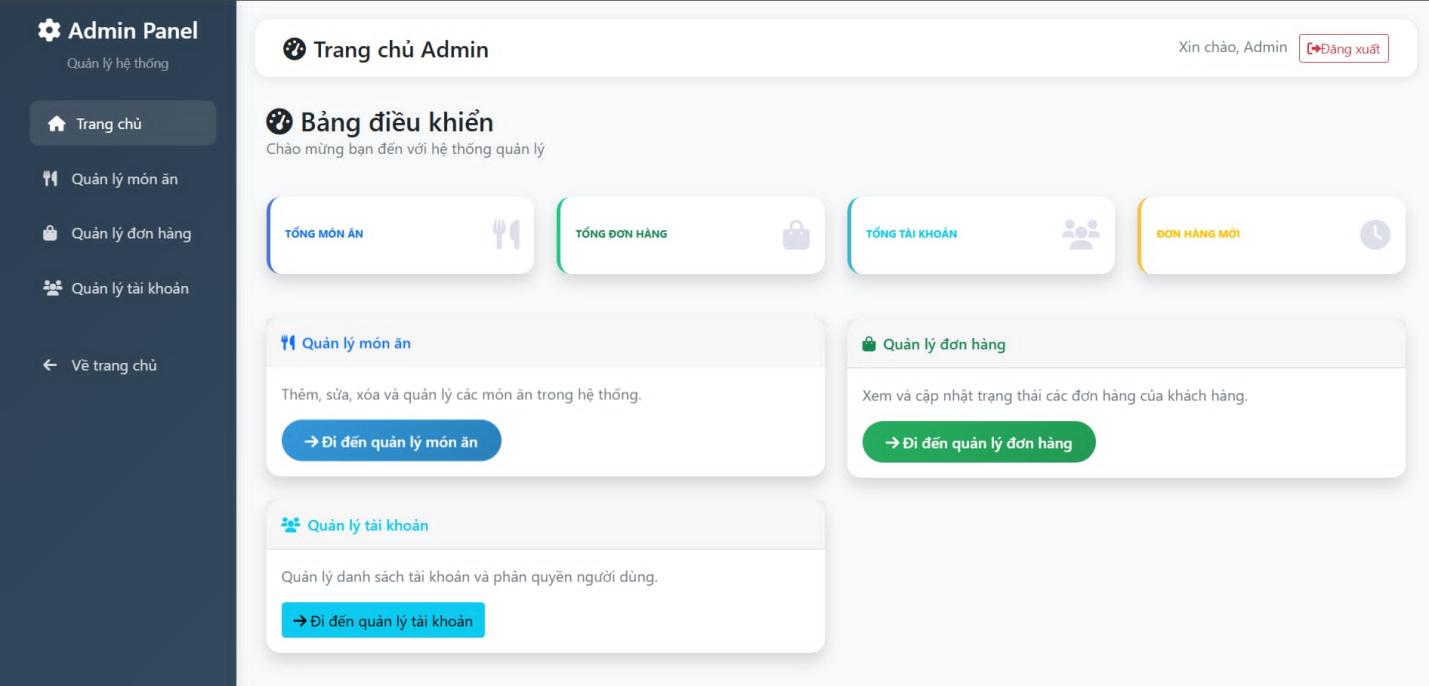


Hình 3.11. Đặt hàng

### Giao diện Admin:

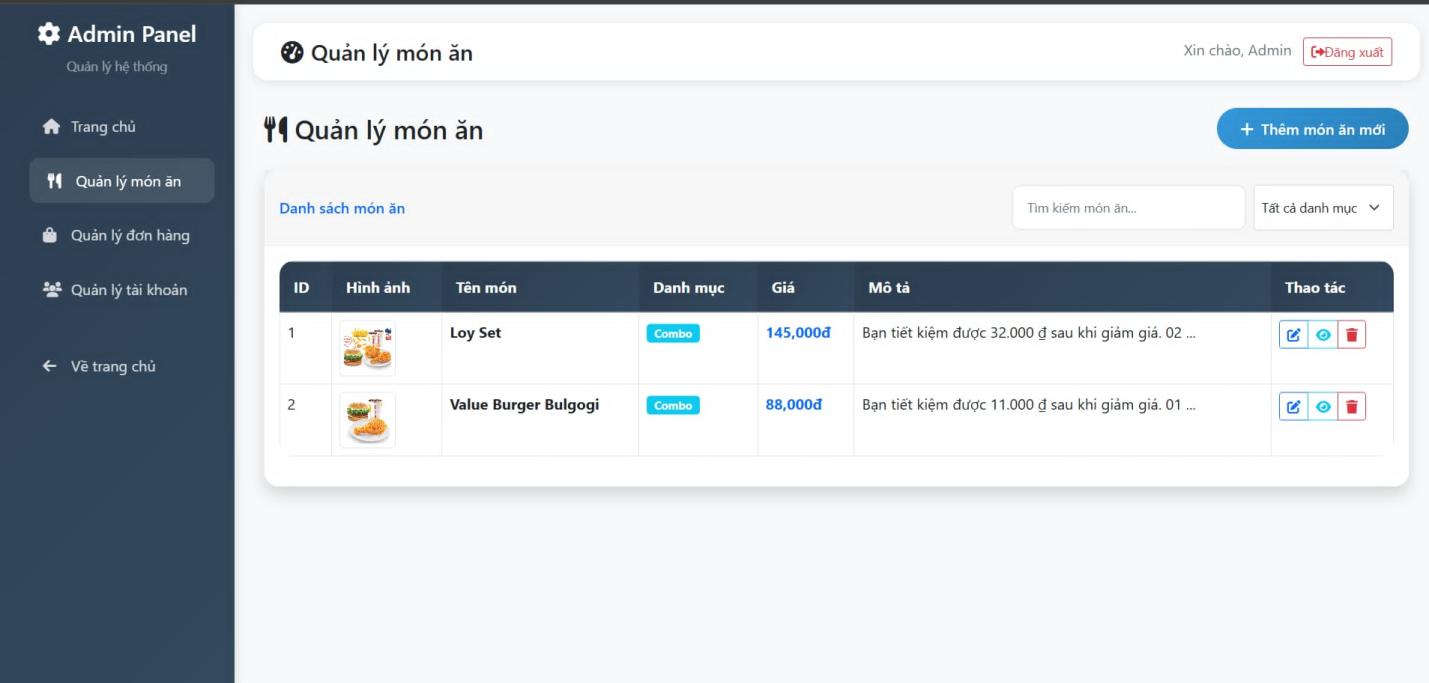
* Dashboard
* Quản lý món ăn
* Quản lý đơn hàng
* Quản lý người dùng
* Thống kê đơn hàng/tháng

#### Dashboard:



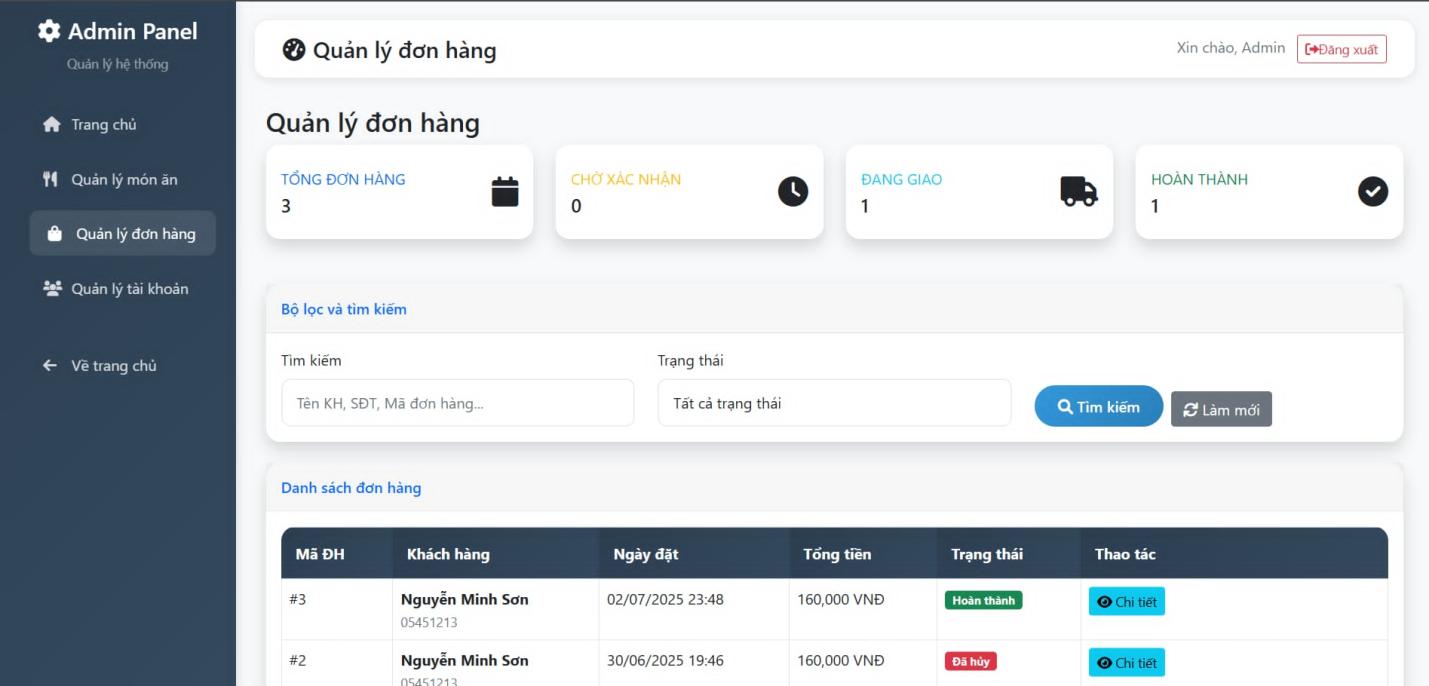
Hình 3.12. Bảng điều khiển

#### Quản lý món ăn:



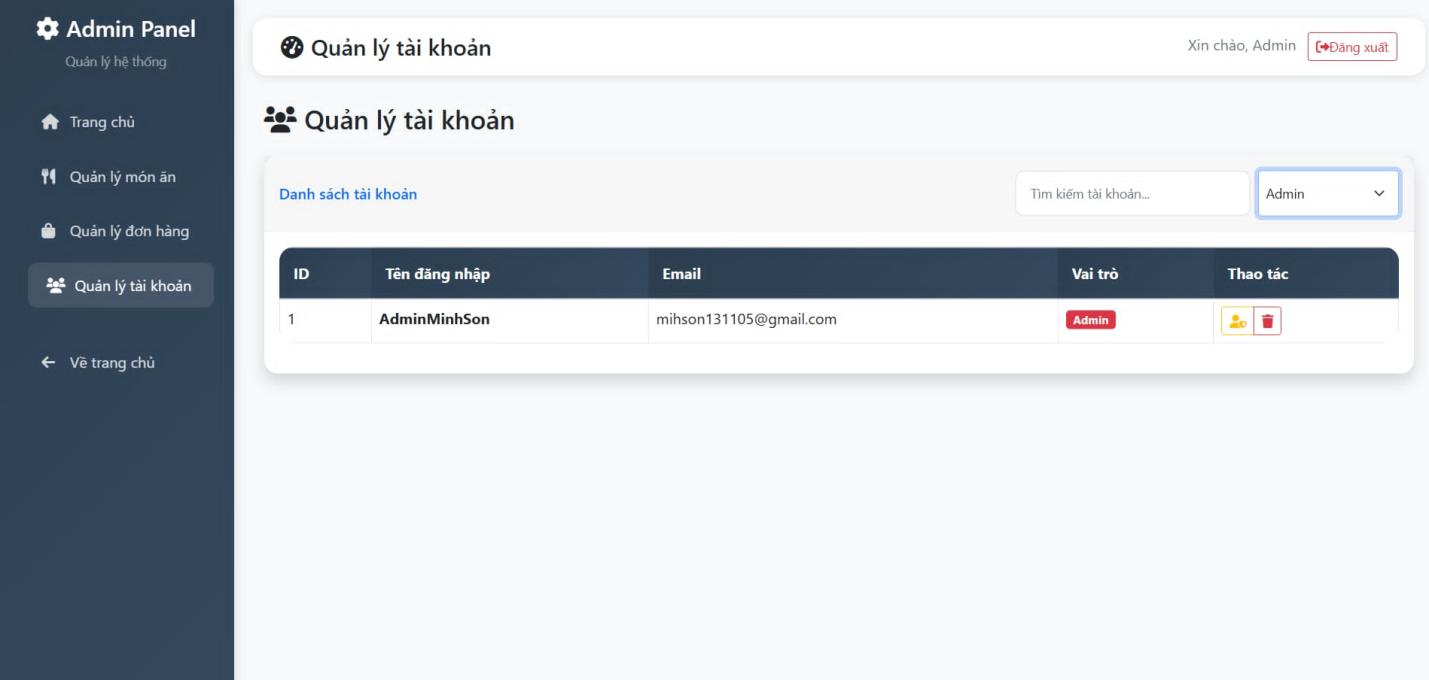
Hình 3.13. Quản lý món ăn

#### Quản lý đơn hàng:



Hình 3.14. Quản lý đơn hàng

#### Quản lý tài khoản:



Hình 3.15. Quản lý tài khoản

## Kiểm thử:

### Kiểm thử chức năng:

| **Tên chức năng** | **Dữ liệu đầu vào** | **Kết quả mong đợi** | **Thực tế** |
| --- | --- | --- | --- |
| Đăng ký | Họ tên, email, mật khẩu | Tạo tài khoản | Thành công |
| Đăng nhập | Email, mật khẩu đúng | Chuyển sang trang chủ | OK |
| Thêm món | Chọn sản phẩm | Giỏ hàng tăng | OK |
| Đặt hàng | Xác nhận giỏ hàng | Tạo đơn mới | OK |

### Kiểm thử giao diện:

| **Thiết bị** | **Tương thích** | **Hiển thị tốt** | **Ghi chú** |
| --- | --- | --- | --- |
| Laptop | ✔️ | ✔️ | --- |
| Điện thoại | ✔️ | ✔️ | Có responsive |
| Máy tính bảng | ✔️ | ✔️ | OK |

### Kiểm thử hiệu năng:

* Kiểm thử ảo 5 user truy cập đồng thời.
* Kết quả:
* Trang thực đơn tải < 1.8s.
* Trang thanh toán tải < 2.3s.
* CPU < 40%, RAM < 500MB.

### Kiểm thử bảo mật:

* Mật khẩu người dùng được hash bằng ASP.NET Identity.
* Session/Token bảo vệ bằng cookie middleware.
* Kiểm tra SQL Injection: Tất cả query dùng LINQ hoặc EF.
* Phân quyền: Dùng [Authorize(Roles = "Admin")] cho các route quản lý.

**KẾT LUẬN**

**1. Kết quả đạt được:**

Trong quá trình thực hiện đề tài, tôi đã áp dụng các kiến thức chuyên ngành liên quan đến thiết kế phần mềm, lập trình web, cơ sở dữ liệu, kiến trúc MVC cũng như kỹ năng làm việc cá nhân để hoàn thiện sản phẩm với các kết quả nổi bật như sau: Xây dựng được website hoàn chỉnh theo mô hình MVC, sử dụng ASP.NET Core MVC làm nền tảng và Entity Framework Code First để phát triển cơ sở dữ liệu một cách linh hoạt. Thiết kế giao diện người dùng thân thiện, hiện đại, hỗ trợ trên cả máy tính và điện thoại di động, đảm bảo trải nghiệm người dùng tốt. Tổ chức hệ thống phân quyền rõ ràng: gồm người dùng thông thường (khách hàng) và quản trị viên (Admin). Cung cấp đầy đủ các chức năng cần thiết như: đăng ký/đăng nhập, xem danh mục món ăn, thêm vào giỏ hàng, đặt hàng, quản lý đơn hàng, quản lý sản phẩm, thống kê doanh thu…Tối ưu hệ thống qua kiểm thử: kiểm thử chức năng, giao diện, hiệu năng và bảo mật đều cho kết quả tốt, đảm bảo hệ thống hoạt động ổn định.

**2. Nhược điểm:**

Bên cạnh những kết quả đạt được, đề tài vẫn còn tồn tại một số hạn chế nhất định, chủ yếu do thời gian và nguồn lực cá nhân có giới hạn: Giao diện chưa được tùy biến cao, chưa có nhiều hiệu ứng tương tác nâng cao như animation, AJAX động hoặc filter nâng cao. Chưa tích hợp thanh toán trực tuyến, người dùng hiện chỉ đặt hàng, chưa có cơ chế thanh toán bằng VNPay, Momo hoặc thẻ ngân hàng. Chưa có chức năng phản hồi đánh giá từ khách hàng đối với sản phẩm đã mua. Bảo mật ở mức cơ bản, chưa triển khai các kỹ thuật nâng cao như xác thực 2 lớp, mã hóa JWT, hạn chế brute force...Chưa hỗ trợ đa ngôn ngữ, điều này ảnh hưởng đến khả năng mở rộng với khách hàng nước ngoài.

**3. Hướng phát triển:**

Trong tương lai, tôi mong muốn tiếp tục phát triển và mở rộng hệ thống để hoàn thiện hơn và có thể ứng dụng thực tế trong các cửa hàng bán đồ ăn nhanh. Các hướng mở rộng cụ thể như sau: Tích hợp các cổng thanh toán online như VNPay, Momo, ZaloPay để nâng cao tính tiện lợi cho khách hàng. Bổ sung chức năng đánh giá sản phẩm, hiển thị bình luận và rating, giúp tạo tính tương tác và phản hồi hai chiều. Quản lý tồn kho thực tế, cảnh báo khi số lượng sản phẩm giảm xuống dưới mức tối thiểu. Nâng cấp module báo cáo thống kê, cho phép lọc doanh thu theo ngày, tháng, sản phẩm bán chạy…Tăng cường bảo mật hệ thống, như xác thực OTP khi đăng nhập, giám sát truy cập bất thường, mã hóa toàn bộ dữ liệu nhạy cảm. Phát triển ứng dụng mobile (Android/iOS) song song với website để tiếp cận người dùng di động dễ dàng hơn. Tự động gợi ý món ăn theo lịch sử đặt hàng bằng thuật toán phân tích hành vi người dùng (gợi ý món tương tự, combo hấp dẫn…).

**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.