ĐẠI HỌC NHA TRANG

Lớp: CC23TTH   
Khoa: Công nghệ thông tin

Nhóm 6 - Thành viên bao gồm:

23TH2520

Huỳnh Lê Minh Thuận

23TH2508

Trần Nhật Hào

GV hướng dẫn: ThS Bùi Chí Thành

Năm 2025

Học phần: Phát triển ứng dụng web

DỰ ÁN THI CUỐI KỲ

MỤC LỤC

[MỤC LỤC i](#_Toc186651216)

[DANH MỤC HÌNH ẢNH iv](#_Toc186651217)

[MỞ ĐẦU 1](#_Toc186651218)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 2](#_Toc186651219)

[1.1. Mục tiêu về tư duy: 2](#_Toc186651220)

[1.2. Mục tiêu kỹ thuật: 2](#_Toc186651221)

[1.3. Mục tiêu cụ thể của dự án: 2](#_Toc186651222)

[1.4. Công cụ và ngôn ngữ được sử dụng trong dự án: 2](#_Toc186651223)

[1.4.1. Frontend bao gồm: HTML, CSS, JS, Bootstrap, Jquery 2](#_Toc186651224)

[1.4.2. Backend bao gồm: .NET, SQL Server, C# 2](#_Toc186651225)

[1.4.3. Công cụ hỗ trợ thêm: Git, Github, Visual Studio, Resharper, Entity… 2](#_Toc186651226)

[1.5. Lý do chọn đề tài: 3](#_Toc186651227)

[1.6. Tính khả thi: 3](#_Toc186651228)

[1.7. Khảo sát hiện trạng: 3](#_Toc186651229)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 4](#_Toc186651230)

[2.1. Tổng quan về ngôn ngữ Asp.NET MVC: 4](#_Toc186651231)

[2.2. Giới thiệu Asp.NET MVC framework: 4](#_Toc186651232)

[2.2.2. Model: 5](#_Toc186651233)

[2.2.3. View: 5](#_Toc186651234)

[2.2.4. Controller: 5](#_Toc186651235)

[2.3. Tìm hiểu sâu về MVC: 5](#_Toc186651236)

[2.3.1. Model: 5](#_Toc186651237)

[2.3.2. Tìm hiểu về View: 6](#_Toc186651238)

[2.3.3. Tìm hiểu về Controller: 6](#_Toc186651239)

[2.3.4. Tại sao MVC quan trọng? 7](#_Toc186651240)

[2.4. Lý do chọn Asp.NET MVC: 8](#_Toc186651241)

[2.5. Tổng quan về SQL Server: 9](#_Toc186651242)

[2.5.1. Giới thiệu về SQL Server: 9](#_Toc186651243)

[2.5.2. Đặc điểm của SQL Server và đối tượng làm việc: 10](#_Toc186651244)

[CHƯƠNG 3: NỘI DUNG THỰC HIỆN 12](#_Toc186651245)

[3.1. Đối tượng sử dụng website: 12](#_Toc186651246)

[3.1.1. Đối với nhân viên: 12](#_Toc186651247)

[3.1.2. Đối với khách hàng: 12](#_Toc186651248)

[3.2. Các chức năng: 12](#_Toc186651249)

[3.2.1. Chức năng đối với nhân viên Admin: 12](#_Toc186651250)

[3.3. Phân Tích hệ thống: 14](#_Toc186651251)

[3.3.1. Các yêu cầu chức năng chính của hệ thống: 14](#_Toc186651252)

[3.4. Thiết kế cơ sở dữ liệu: 15](#_Toc186651253)

[3.4.1. Mô hình cơ sở dữ liệu: 15](#_Toc186651254)

[3.4.2. Danh sách các đối tượng: 16](#_Toc186651255)

[3.5. Thiết kế giao diện: 18](#_Toc186651256)

[3.5.1. Giao diện trang chủ: 18](#_Toc186651257)

[3.5.2. Giao diện đăng nhập: 19](#_Toc186651258)

[3.5.3. Trang danh sách sản phẩm: 20](#_Toc186651259)

[3.5.4. Trang Hóa Đơn: 21](#_Toc186651260)

[3.6. Giao diện trang Nhân Viên: 24](#_Toc186651261)

[3.6.1. Trang chủ quản lý nhân viên: 24](#_Toc186651262)

[3.7. Giao diện trang quản lý khách hàng: 27](#_Toc186651263)

[3.7.1. Mục đích của trang danh sách khách hàng: 27](#_Toc186651264)

[3.7.2. Lý do nên có một ID khách hàng mang tên "Default": 28](#_Toc186651265)

[3.8. Những trang được tự động hóa: 29](#_Toc186651266)

[3.8.1. Trang Create của trang Hóa Đơn: 29](#_Toc186651267)

[3.8.2. Trang Create của trang Chi tiết hóa đơn: 32](#_Toc186651268)

[CHƯƠNG 4: KẾT LUẬN 37](#_Toc186651269)

[4.1. Kết quả đạt được của đề tài: 37](#_Toc186651270)

[4.1.1. Tóm tắt kết quả nghiên cứu: 37](#_Toc186651271)

[4.1.2. Phân tích chi tiết kết quả: 37](#_Toc186651272)

[4.1.3. Thảo luận và so sánh: 37](#_Toc186651273)

[4.1.4. Kết luận và đề xuất: 37](#_Toc186651274)

[4.1.5. Ứng dụng thực tiễn: 37](#_Toc186651275)

[4.1.6. Đóng góp cho cộng đồng nghiên cứu: 38](#_Toc186651276)

[4.1.7. Hướng phát triển thêm: 38](#_Toc186651277)

[4.1.8. Phản hồi và cải tiến: 38](#_Toc186651278)

[4.2. Hạn chế của đề tài: 38](#_Toc186651279)

[4.2.1. Giao diện: 38](#_Toc186651280)

[4.2.2. Chức năng thống kê: 39](#_Toc186651281)

[4.2.3. Phân quyền người dùng, đăng ký: 39](#_Toc186651282)

[4.3. Kết luận: 39](#_Toc186651283)

[KẾT LUẬN CHUNG 40](#_Toc186651284)

[TÀI LIỆU THAM KHẢO 40](#_Toc186651285)

DANH MỤC HÌNH ẢNH

[Hình 2.1: Mẫu Model -View – Controller 4](#_Toc186651288)

[Hình 2.2: Logo Asp.NET Framework 9](#_Toc186651289)

[Hình 2.3: Logo SQL Server 10](#_Toc186651290)

[Hình 3.1: Biểu đồ dữ liệu quan hệ 15](#_Toc186651291)

[Hình 3.2: Giao diện trang chủ 18](#_Toc186651292)

[Hình 3.3: Giao diện trang đăng nhập 19](#_Toc186651293)

[Hình 3.4: Giao diện trang danh sách sản phẩm 20](#_Toc186651294)

[Hình 3.5: Giao diện trang chi tiết sản phẩm 21](#_Toc186651295)

[Hình 3.6: Giao diện trang Hóa Đơn 21](#_Toc186651296)

[Hình 3.7: Giao diện chi tiết hóa đơn 23](#_Toc186651297)

[Hình 3.8: Giao diện trang quản lý nhân viên 24](#_Toc186651298)

[Hình 3.9: Giao diện trang chi tiết nhân viên 26](#_Toc186651299)

[Hình 3.10: Giao diện trang quản lý khách hàng 27](#_Toc186651300)

[Hình 3.11: Trang Create của trang quản lý Hóa Đơn 29](#_Toc186651301)

[Hình 3.12: Múi thời gian trong máy tính dùng để lấy mốc tự động hóa cho trang Create Hoa Don 29](#_Toc186651302)

[Hình 3.13: Đoạn code Create trong file Controller của trang Hóa Đơn 30](#_Toc186651303)

[Hình 3.14: Code Javascripts tự động điền thời gian và ngày lập 31](#_Toc186651304)

[Hình 3.15: Trang Create của trang Chi tiết hóa đơn 32](#_Toc186651305)

[Hình 3.16: Đoạn code Javascripts giúp tự động điền cho mục Thành Tiền của Chi Tiết Hóa đơn 33](#_Toc186651306)

[Hình 3.17: Đoạn code trả về JSON 34](#_Toc186651307)

[Hình 3.18: Logo Postman 34](#_Toc186651308)

[Hình 3.19: Ảnh kết quả sau khi test bằng Postman 35](#_Toc186651309)

MỞ ĐẦU

Trong thời đại công nghệ số hiện nay, các hệ thống quản lý đóng vai trò quan trọng trong việc nâng cao hiệu suất và hiệu quả của doanh nghiệp. Đặc biệt, việc quản lý các hoạt động kinh doanh của quán cà phê đòi hỏi một hệ thống quản lý chuyên nghiệp và hiệu quả.

Báo cáo này trình bày quá trình xây dựng và triển khai hệ thống quản lý quán cà phê sử dụng công nghệ .NET ASP. Mục tiêu của hệ thống là cung cấp các chức năng quản lý hóa đơn, khách hàng, nhân viên, và sản phẩm một cách hiệu quả và tiện lợi. Báo cáo cũng nêu lên các bước triển khai, phương pháp kiểm thử và đánh giá kết quả của hệ thống.

# TỔNG QUAN ĐỀ TÀI

## Mục tiêu về tư duy:

Học hỏi và nắm bắt được kỹ năng khảo sát, trình bày, phân tích những mong muốn cũng như tầm nhìn của các doanh nghiệp để có thể đưa ra những phương án kỹ thuật số tốt nhất, nhằm nâng cao về chất lượng hệ thống quản lý dân sự.

Bên cạnh đó thì rất là mong muốn thông qua dự án cuối kỳ này thì nó sẽ rèn luyện thêm cho chúng ta tư duy làm việc nhóm đoàn kết hơn trong một môi trường mà rất cần sự đoàn kết và chung tay như thế này.

## Mục tiêu kỹ thuật:

Xây dựng hệ thống quản lý quán cà phê với các chức năng cơ bản như quản lý hóa đơn, khách hàng, nhân viên, và sản phẩm.

## Mục tiêu cụ thể của dự án:

Quản lý hóa đơn: Tạo, chỉnh sửa, xóa, và xem chi tiết hóa đơn.

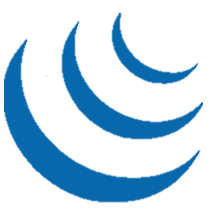
Quản lý khách hàng: Thêm mới, chỉnh sửa, xóa, và xem danh sách khách hàng.

Quản lý nhân viên: Thêm mới, chỉnh sửa, xóa, và xem danh sách nhân viên.

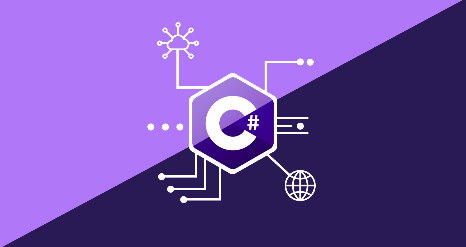
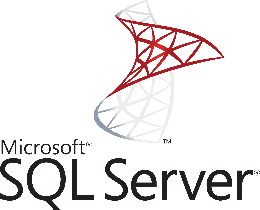
Quản lý sản phẩm: Thêm mới, chỉnh sửa, xóa, và xem danh sách sản phẩm.

## Công cụ và ngôn ngữ được sử dụng trong dự án:

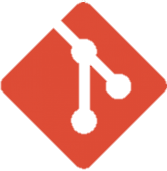
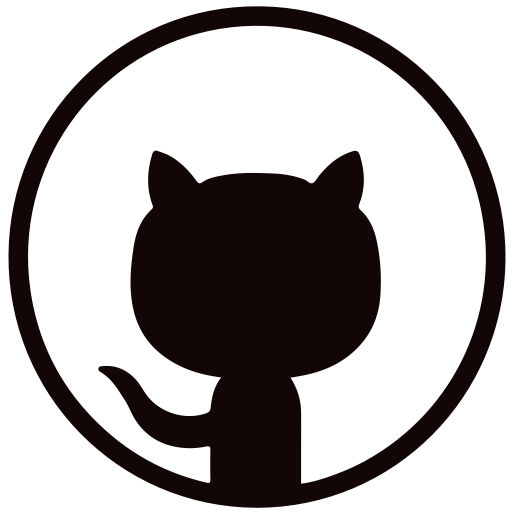
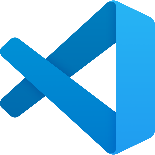
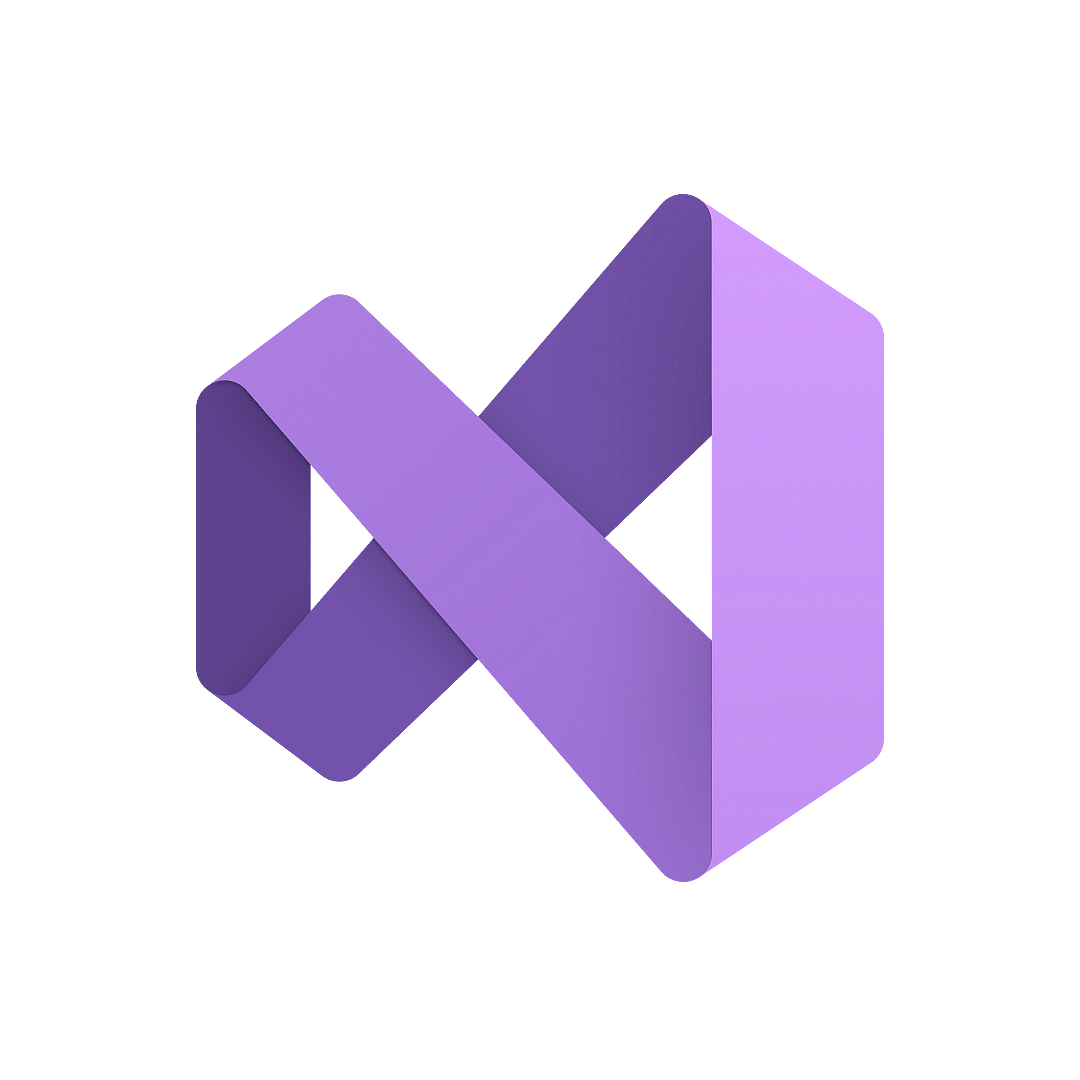
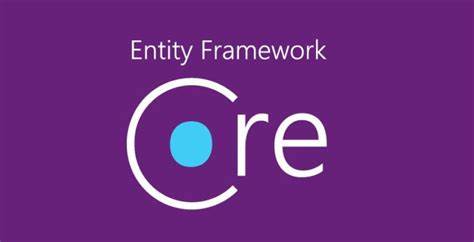
### Logo Bootstrap – Logos PNGFrontend bao gồm: HTML, CSS, JS, Bootstrap, Jquery



### Backend bao gồm: .NET, SQL Server, C#



### Công cụ hỗ trợ thêm: Git, Github, Visual Studio, Resharper, Entity…



## Lý do chọn đề tài:

Chúng mình chọn đề tài “Xây dựng website quản lý quán cafe” vì chúng ta đang sống trong thời đại của cách mạng công nghiệp 4.0, nơi mọi thứ đều hướng tới công nghiệp hóa và hiện đại hóa. Các trang thương mại điện tử và các nền tảng trực tuyến ngày càng trở nên quan trọng trong việc kinh doanh và mua sắm. Bằng cách tận dụng lợi thế của những nền tảng này, chúng mình muốn tạo ra một hệ thống quản lý quán cafe hiệu quả và tiện lợi nhất. Hy vọng rằng dự án này sẽ giúp quán cafe của mình tiếp cận được nhiều khách hàng hơn, tối ưu hóa quy trình quản lý và phát triển mạnh mẽ trong tương lai.

## Tính khả thi:

Dự án “Xây dựng website quản lý quán cafe” rất khả thi vì hiện nay nhu cầu thị trường cao và công nghệ phát triển cho phép triển khai dễ dàng. Với đội ngũ chuyên môn và hiệu quả kinh tế dự kiến, dự án này có thể cải thiện quy trình quản lý và thu hút nhiều khách hàng hơn.

## Khảo sát hiện trạng:

Trong những năm gần đây, thị trường cà phê Việt Nam đã chứng kiến sự phát triển mạnh mẽ, không chỉ tại các thành phố lớn mà còn ở các khu vực đô thị hóa. Nhu cầu của người tiêu dùng đối với quán cafe không chỉ dừng lại ở việc thưởng thức đồ uống mà còn tìm kiếm một không gian thư giãn, làm việc và gặp gỡ bạn bè. Đặc biệt, với xu hướng làm việc từ xa và học tập trực tuyến ngày càng phổ biến, các quán cafe đang trở thành lựa chọn ưu tiên cho nhiều người muốn có một môi trường làm việc thoải mái, sáng tạo.

Mặc dù có sự phát triển, nhưng thị trường cũng đang đối mặt với sự cạnh tranh khốc liệt. Rất nhiều quán cafe mới mở ra, từ những chuỗi cafe lớn đến các quán cafe độc lập với phong cách riêng biệt. Điều này đòi hỏi các quán cafe phải không ngừng nâng cao chất lượng dịch vụ và tìm cách thu hút khách hàng bằng những trải nghiệm độc đáo và tiện lợi.

Một khảo sát gần đây cho thấy, hơn 60% khách hàng mong muốn các quán cafe cung cấp dịch vụ đặt chỗ trước và thanh toán trực tuyến để tiết kiệm thời gian. Đồng thời, việc quản lý hiệu quả và đồng bộ giữa các khâu như đặt hàng, thanh toán, quản lý kho hàng và chăm sóc khách hàng cũng là yếu tố quyết định đến sự thành công của một quán cafe. Vì vậy, một website quản lý quán cafe hiện đại và tiện ích sẽ là giải pháp lý tưởng để giải quyết các thách thức này, giúp các quán cafe đáp ứng nhu cầu của khách hàng một cách tốt nhất và nâng cao lợi thế cạnh tranh.

Từ những yếu tố trên, có thể thấy rằng việc xây dựng một website quản lý quán cafe không chỉ phù hợp với xu hướng phát triển của thị trường mà còn là một bước đi cần thiết để nâng cao hiệu quả kinh doanh và đáp ứng kỳ vọng của khách hàng.

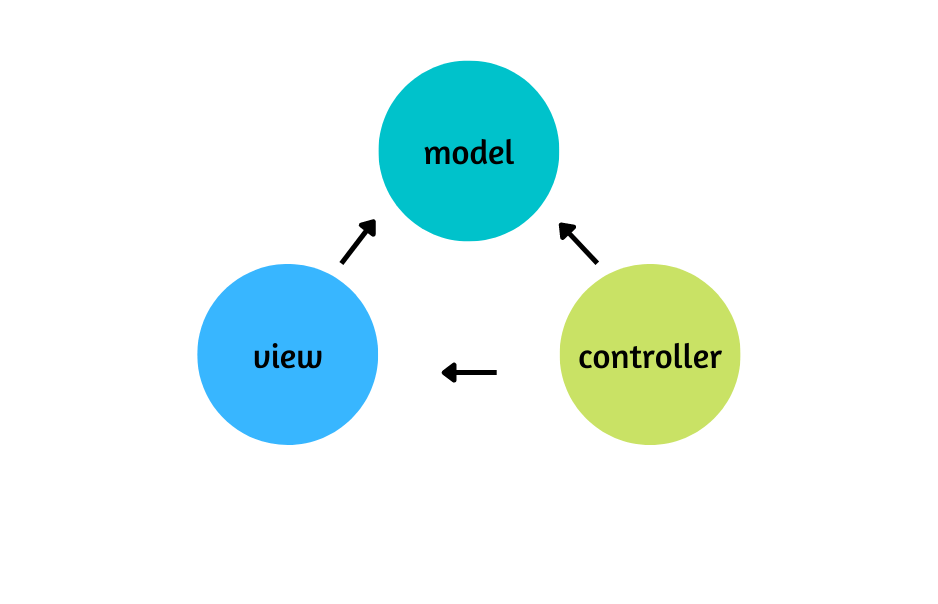
# CƠ SỞ LÝ THUYẾT

## Tổng quan về ngôn ngữ Asp.NET MVC:

ASP.NET MVC là một framework phát triển ứng dụng web mạnh mẽ và phổ biến, được phát triển bởi Microsoft. MVC, viết tắt của Model-View-Controller, là một mô hình kiến trúc phần mềm chia ứng dụng thành ba thành phần chính: Model, View và Controller. Model đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng, xử lý các tác vụ như truy xuất và lưu trữ dữ liệu từ cơ sở dữ liệu, kiểm tra và xác thực dữ liệu. View đại diện cho giao diện người dùng, nơi hiển thị dữ liệu từ Model và chịu trách nhiệm trình bày dữ liệu dưới dạng HTML và các định dạng khác mà người dùng có thể tương tác. Controller điều khiển luồng xử lý và tương tác giữa Model và View, nhận các yêu cầu từ người dùng, xử lý qua Model và trả kết quả cho View để hiển thị. ASP.NET MVC cung cấp nhiều ưu điểm như tính linh hoạt, tái sử dụng mã dễ dàng, kiểm thử độc lập các thành phần, và tối ưu hóa hiệu suất qua cơ chế caching và tối ưu hóa HTML, CSS và JavaScript. Với những ưu điểm trên, ASP.NET MVC là lựa chọn tuyệt vời cho việc phát triển các ứng dụng web hiện đại và phức tạp.

## Giới thiệu Asp.NET MVC framework:

ASP.NET MVC là một framework phát triển ứng dụng web hiện đại và mạnh mẽ do Microsoft phát triển, được xây dựng trên nền tảng ASP.NET. MVC viết tắt của Model-View-Controller, một mô hình kiến trúc phần mềm phân chia ứng dụng thành ba thành phần chính: Model (dữ liệu và logic nghiệp vụ), View (giao diện người dùng) và Controller (điều khiển luồng xử lý). Việc tách biệt này giúp phát triển ứng dụng linh hoạt, dễ dàng bảo trì và nâng cấp. ASP.NET MVC cung cấp khả năng tối ưu hóa hiệu suất, hỗ trợ kiểm thử và tái sử dụng mã, cùng với các công cụ mạnh mẽ và thư viện phong phú, giúp lập trình viên tạo ra các ứng dụng web phức tạp và hiệu quả. Framework này đặc biệt phù hợp cho việc phát triển các ứng dụng yêu cầu tính tương tác cao và quản lý dữ liệu phức tạp.



Mẫu Model -View – Controller

### Model:

Model là phần chịu trách nhiệm quản lý dữ liệu và logic nghiệp vụ của ứng dụng. Nó tương tác trực tiếp với cơ sở dữ liệu để lưu trữ và truy xuất dữ liệu. Model cũng chứa các quy tắc và logic nghiệp vụ, như tính toán, kiểm tra tính hợp lệ và xử lý dữ liệu. Trong ASP.NET MVC, Model thường được triển khai dưới dạng các lớp C# (hoặc VB.NET) và sử dụng các công nghệ như Entity Framework để làm việc với cơ sở dữ liệu.

### View:

View là phần chịu trách nhiệm hiển thị dữ liệu và tương tác với người dùng. View sử dụng các dữ liệu từ Model và trình bày chúng dưới dạng HTML, CSS và JavaScript. Trong ASP.NET MVC, View thường được xây dựng bằng các tệp .cshtml (Razor) hoặc .aspx. View chỉ nên chứa mã liên quan đến việc hiển thị dữ liệu và giao diện người dùng, không nên chứa logic nghiệp vụ phức tạp.

### Controller:

Controller là phần điều khiển luồng dữ liệu giữa Model và View. Nó nhận các yêu cầu từ người dùng thông qua các URL, xử lý chúng bằng cách gọi Model và sau đó chọn View phù hợp để trả về cho người dùng. Controller cũng có thể thực hiện các tác vụ như xác thực, xử lý form, và điều hướng. Trong ASP.NET MVC, Controller được triển khai dưới dạng các lớp kế thừa từ lớp Controller, và mỗi hành động (action) trong Controller là một phương thức có thể được gọi thông qua URL.

## Tìm hiểu sâu về MVC:

### Model:

**Model** là thành phần chịu trách nhiệm quản lý dữ liệu và logic nghiệp vụ của ứng dụng. Nó bao gồm các đối tượng đại diện cho dữ liệu trong ứng dụng, chẳng hạn như các lớp đối tượng (Entity) và các quy trình xử lý dữ liệu (Business Logic). Các nhiệm vụ chính của Model bao gồm:

#### Quản lý dữ liệu:

Model tương tác trực tiếp với cơ sở dữ liệu hoặc các nguồn dữ liệu khác để thực hiện các thao tác như truy xuất, lưu trữ, cập nhật và xóa dữ liệu.

#### Kiểm tra tính hợp lệ của dữ liệu:

Model chứa các quy tắc và logic để kiểm tra tính hợp lệ của dữ liệu, đảm bảo rằng dữ liệu nhập vào tuân thủ các yêu cầu và quy định của ứng dụng.

#### Thực hiện các tính toán nghiệp vụ:

Model xử lý các logic nghiệp vụ, chẳng hạn như tính toán giá trị, xử lý đơn hàng, và các tác vụ khác liên quan đến dữ liệu.

Trong ASP.NET MVC, Model thường được triển khai bằng các lớp C# hoặc VB.NET. Các lớp này có thể sử dụng Entity Framework hoặc các ORM (Object-Relational Mapping) khác để làm việc với cơ sở dữ liệu một cách hiệu quả. Entity Framework giúp đơn giản hóa quá trình truy vấn và thao tác dữ liệu bằng cách sử dụng các đối tượng và truy vấn ngôn ngữ LINQ (Language Integrated Query).

### Tìm hiểu về View:

**View** là thành phần chịu trách nhiệm hiển thị dữ liệu và giao diện người dùng của ứng dụng. Nó nhận dữ liệu từ Model và trình bày chúng dưới dạng HTML, CSS và JavaScript để người dùng có thể tương tác. Các nhiệm vụ chính của View bao gồm:

#### Hiển thị dữ liệu:

View hiển thị dữ liệu từ Model cho người dùng thông qua giao diện người dùng. Nó có thể bao gồm các thành phần như bảng, biểu đồ, biểu mẫu và các phần tử giao diện khác.

#### Tạo giao diện người dùng:

View chịu trách nhiệm tạo ra giao diện người dùng thân thiện và dễ sử dụng. Nó sử dụng các tệp .cshtml (Razor) hoặc .aspx để kết hợp mã C# với HTML, giúp trình bày dữ liệu một cách linh hoạt và hiệu quả.

#### Tương tác với người dùng:

View cung cấp các thành phần giao diện cho phép người dùng tương tác với ứng dụng, chẳng hạn như các nút, trường nhập liệu, và các liên kết.

Razor là một engine template mạnh mẽ được sử dụng trong ASP.NET MVC để tạo ra các tệp View. Razor cho phép lập trình viên kết hợp mã C# với HTML một cách dễ dàng và trực quan, giúp việc tạo ra các giao diện động trở nên đơn giản hơn.

### Tìm hiểu về Controller:

**Controller** là thành phần điều khiển luồng dữ liệu giữa Model và View, nhận các yêu cầu từ người dùng và xử lý chúng. Controller chịu trách nhiệm điều hướng và thực hiện các tác vụ logic ứng dụng. Các nhiệm vụ chính của Controller bao gồm:

#### Nhận và xử lý yêu cầu:

Controller nhận các yêu cầu từ người dùng thông qua các URL. Mỗi hành động (action) trong Controller là một phương thức có thể được gọi thông qua URL. Controller xác định các yêu cầu và gọi Model để xử lý dữ liệu.

#### Xử lý form và xác thực:

Controller xử lý các form nhập liệu từ người dùng, xác thực dữ liệu nhập vào và chuyển dữ liệu đến Model để lưu trữ hoặc xử lý.

#### Điều hướng và phản hồi:

Sau khi xử lý dữ liệu, Controller chọn View phù hợp để trả về cho người dùng. Nó cũng có thể điều hướng người dùng đến các trang khác hoặc thực hiện các tác vụ khác tùy theo logic ứng dụng.

Trong ASP.NET MVC, Controller được triển khai dưới dạng các lớp kế thừa từ lớp **Controller**. Mỗi phương thức trong Controller đại diện cho một hành động cụ thể và có thể được gọi thông qua URL. Controller đóng vai trò cầu nối giữa Model và View, đảm bảo rằng dữ liệu từ Model được hiển thị đúng cách và các yêu cầu từ người dùng được xử lý một cách hiệu quả.

### Tại sao MVC quan trọng?

Việc sử dụng mô hình MVC trong ASP.NET đem lại nhiều lợi ích đáng kể, bao gồm:

#### Tách biệt rõ ràng giữa các thành phần:

MVC giúp tách biệt giao diện người dùng, logic nghiệp vụ và dữ liệu, giúp việc phát triển, bảo trì và mở rộng ứng dụng trở nên dễ dàng và hiệu quả hơn.

#### Tái sử dụng mã:

MVC giúp tách biệt giao diện người dùng, logic nghiệp vụ và dữ liệu, giúp việc phát triển, bảo trì và mở rộng ứng dụng trở nên dễ dàng và hiệu quả hơn.

#### Kiểm thử dễ dàng:

Việc tách biệt các thành phần trong mô hình MVC giúp mã dễ dàng được tái sử dụng, giảm bớt công việc lặp lại và tăng hiệu quả phát triển.

#### Hiệu suất cao:

MVC cho phép kiểm thử độc lập các thành phần riêng lẻ, nâng cao chất lượng ứng dụng và đảm bảo rằng các tính năng hoạt động đúng như mong đợi.

#### Linh hoạt và mở rộng:

MVC cho phép dễ dàng tích hợp với các công nghệ và framework khác, phù hợp cho các dự án lớn và phức tạp, đáp ứng được nhu cầu đa dạng của người dùng.

Với những ưu điểm trên, mô hình MVC trong ASP.NET là lựa chọn lý tưởng cho việc phát triển các ứng dụng web hiện đại và phức tạp, giúp tối ưu hóa quy trình làm việc và nâng cao chất lượng sản phẩm.

## Lý do chọn Asp.NET MVC:

Lý do để chọn ASP.NET MVC cho việc phát triển ứng dụng web rất nhiều và mang lại nhiều lợi ích đáng kể. Đầu tiên, ASP.NET MVC cung cấp một môi trường phát triển ứng dụng mạnh mẽ và linh hoạt. MVC (Model-View-Controller) giúp tách biệt rõ ràng giữa các thành phần của ứng dụng, bao gồm giao diện người dùng (View), logic nghiệp vụ (Controller) và dữ liệu (Model). Việc tách biệt này giúp lập trình viên dễ dàng quản lý và bảo trì mã nguồn, đồng thời tăng cường khả năng tái sử dụng mã. Mỗi thành phần có thể được phát triển và kiểm thử độc lập, giảm thiểu sự phụ thuộc và lỗi phát sinh khi cập nhật hoặc mở rộng ứng dụng.

ASP.NET MVC dễ dàng tích hợp với nhiều công nghệ hiện đại khác nhau, từ cơ sở dữ liệu đến các dịch vụ web và API. Các thư viện và công cụ hỗ trợ như Entity Framework, LINQ, và Dependency Injection giúp lập trình viên xây dựng các ứng dụng phức tạp một cách nhanh chóng và hiệu quả. Ngoài ra, ASP.NET MVC hỗ trợ các chuẩn HTML5, CSS3 và JavaScript, cho phép tạo ra các giao diện người dùng hiện đại và tương tác.

Một trong những ưu điểm nổi bật của ASP.NET MVC là khả năng tối ưu hóa hiệu suất. Hệ thống caching mạnh mẽ giúp giảm tải cho máy chủ và tăng tốc độ phản hồi của ứng dụng. ASP.NET MVC cũng hỗ trợ tối ưu hóa SEO (Search Engine Optimization) bằng cách cho phép lập trình viên kiểm soát các URL và meta tags, giúp cải thiện thứ hạng của trang web trên các công cụ tìm kiếm. Hơn nữa, ASP.NET MVC có khả năng mở rộng cao, phù hợp với các ứng dụng từ nhỏ đến lớn, đáp ứng được nhu cầu của các doanh nghiệp và tổ chức. Kiểm thử là một phần quan trọng trong quá trình phát triển ứng dụng, và ASP.NET MVC cung cấp môi trường thuận lợi để thực hiện điều này. Với việc tách biệt các thành phần MVC, các lập trình viên có thể dễ dàng viết các test case cho từng phần riêng lẻ. Các công cụ và framework kiểm thử như NUnit và MSTest giúp thực hiện kiểm thử đơn vị (unit testing) và kiểm thử tích hợp (integration testing) một cách hiệu quả.

Ngoài ra, ASP.NET MVC hỗ trợ phát triển theo nhóm, cho phép các thành viên trong nhóm làm việc trên các phần khác nhau của ứng dụng mà không gây ra xung đột mã nguồn. Là một sản phẩm của Microsoft, ASP.NET MVC được hỗ trợ mạnh mẽ từ cộng đồng lập trình viên và các tài liệu phong phú. Các diễn đàn, blog, và trang web học thuật cung cấp nhiều thông tin và hướng dẫn chi tiết, giúp lập trình viên nhanh chóng làm quen và áp dụng framework này vào dự án của mình. Ngoài ra, Microsoft liên tục cập nhật và cải tiến ASP.NET MVC, đảm bảo rằng framework luôn đáp ứng được các yêu cầu và thách thức mới nhất trong phát triển ứng dụng web. ASP.NET MVC cho phép lập trình viên tùy chỉnh và mở rộng các thành phần của framework theo nhu cầu cụ thể của dự án. Các tính năng như routing, filters, và custom helpers cho phép linh hoạt điều chỉnh hành vi và giao diện của ứng dụng. Điều này giúp các lập trình viên tạo ra những ứng dụng độc đáo và phù hợp với mục tiêu kinh doanh của doanh nghiệp. Bảo mật là một yếu tố quan trọng trong phát triển ứng dụng web, và ASP.NET MVC cung cấp nhiều tính năng bảo mật tiên tiến.

Các cơ chế như authentication, authorization, và data validation giúp bảo vệ ứng dụng khỏi các cuộc tấn công và lỗ hổng bảo mật. Ngoài ra, ASP.NET MVC hỗ trợ các tiêu chuẩn bảo mật như HTTPS, XSS (Cross-Site Scripting) protection, và CSRF (Cross-Site Request Forgery) protection, đảm bảo rằng ứng dụng của bạn luôn được bảo vệ an toàn. Tóm lại, việc chọn ASP.NET MVC làm framework phát triển ứng dụng web mang lại nhiều lợi ích vượt trội về hiệu quả phát triển, tích hợp công nghệ hiện đại, tối ưu hóa hiệu suất, hỗ trợ kiểm thử, tài liệu phong phú, khả năng tùy chỉnh và bảo mật cao. Đây là một lựa chọn lý tưởng cho các dự án phát triển ứng dụng web hiện đại và phức tạp, giúp các doanh nghiệp nâng cao hiệu quả kinh doanh và đáp ứng nhu cầu ngày càng cao của người dùng.



Logo Asp.NET Framework

## Tổng quan về SQL Server:

### Giới thiệu về SQL Server:

Giống như ASP.NET thì SQL Server cũng là một đứa "con cưng" của Microsoft, được phát triển và duy trì bởi tập đoàn công nghệ hàng đầu này. SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS - Relational Database Management System) mạnh mẽ và toàn diện, được thiết kế để hỗ trợ quản lý và phân tích dữ liệu trong các ứng dụng doanh nghiệp từ nhỏ đến lớn.

SQL Server cung cấp một loạt các công cụ và dịch vụ để quản lý dữ liệu, bao gồm lưu trữ, truy vấn, và phân tích dữ liệu. Với khả năng xử lý dữ liệu lớn và tích hợp với nhiều công nghệ hiện đại, SQL Server đáp ứng nhu cầu đa dạng của các doanh nghiệp, từ việc quản lý cơ sở dữ liệu truyền thống đến các giải pháp phân tích dữ liệu phức tạp.

Một trong những ưu điểm nổi bật của SQL Server là tính bảo mật cao, với các cơ chế bảo vệ dữ liệu như mã hóa, xác thực người dùng và kiểm soát truy cập. Điều này giúp đảm bảo rằng dữ liệu của doanh nghiệp luôn được bảo vệ an toàn trước các mối đe dọa tiềm tàng.



Logo SQL Server

SQL Server còn hỗ trợ tích hợp với nhiều công cụ phát triển và phân tích dữ liệu khác của Microsoft, như Power BI, Azure, và Visual Studio, giúp tạo ra một hệ sinh thái dữ liệu mạnh mẽ và linh hoạt. Khả năng tương thích cao với các ứng dụng và dịch vụ khác của Microsoft cũng giúp việc triển khai và quản lý SQL Server trở nên dễ dàng và hiệu quả.

Với những tính năng vượt trội và khả năng mở rộng linh hoạt, SQL Server đã và đang là lựa chọn ưu tiên của nhiều doanh nghiệp trên toàn thế giới trong việc quản lý và phân tích dữ liệu. Dù bạn là một doanh nghiệp nhỏ đang tìm kiếm giải pháp quản lý dữ liệu hiệu quả, hay một tập đoàn lớn cần một hệ thống phân tích dữ liệu mạnh mẽ, SQL Server đều có thể đáp ứng mọi nhu cầu của bạn một cách tốt nhất.

### Đặc điểm của SQL Server và đối tượng làm việc:

Server, hệ quản trị cơ sở dữ liệu của Microsoft, nổi bật với nhiều đặc điểm ưu việt. Một trong những đặc điểm chính là tính bảo mật cao, với các cơ chế mã hóa dữ liệu, kiểm soát truy cập và xác thực người dùng. Điều này giúp bảo vệ dữ liệu khỏi các mối đe dọa tiềm tàng và đảm bảo an toàn cho thông tin của doanh nghiệp. Khả năng quản lý dữ liệu lớn là một đặc điểm quan trọng khác, cho phép SQL Server xử lý và phân tích khối lượng dữ liệu khổng lồ một cách hiệu quả.

SQL Server cũng nổi tiếng với hiệu suất cao, nhờ vào các cơ chế tối ưu hóa truy vấn và khả năng caching mạnh mẽ. Điều này giúp tăng tốc độ truy xuất dữ liệu và giảm tải cho hệ thống, đảm bảo hoạt động mượt mà. Khả năng tích hợp linh hoạt của SQL Server với các công cụ và dịch vụ khác của Microsoft, như Azure, Power BI và Visual Studio, giúp tạo ra một hệ sinh thái dữ liệu mạnh mẽ và linh hoạt.

Đối tượng làm việc của SQL Server rất đa dạng, bao gồm các doanh nghiệp từ nhỏ đến lớn, các tổ chức tài chính, y tế, giáo dục và nhiều lĩnh vực khác. Các nhà phát triển, quản trị viên cơ sở dữ liệu (DBA), nhà phân tích dữ liệu và quản lý dự án đều có thể sử dụng SQL Server để xây dựng, quản lý và phân tích dữ liệu. Với các công cụ hỗ trợ như SQL Server Management Studio (SSMS), SQL Server Data Tools (SSDT) và các dịch vụ phân tích dữ liệu tiên tiến, SQL Server đáp ứng mọi nhu cầu từ phát triển ứng dụng, quản lý cơ sở dữ liệu đến phân tích dữ liệu chuyên sâu.

# NỘI DUNG THỰC HIỆN

## Đối tượng sử dụng website:

### Đối với nhân viên:

Nhân viên admin: Có thể xem, thêm, sửa, tìm kiếm sản phẩm. Có thể xem danh sách và chi tiết khách hàng. Có thể xem danh sách và chi tiết các nhân viên. Có thể xem, tìm kiếm các hóa đơn. Có thể nhận và xác nhận những hóa đơn mà khách hàng đặt. Có thể đăng ký tài khoản cho các nhân viên khác. Và có chức năng thống kê doanh thu

Nhân viên bình thường: Có thể xem, thêm, sửa, tìm kiếm sản phẩm. Có thể xem danh sách và chi tiết khách hàng. Có thể xem danh sách và chi tiết các nhân viên. Có thể xem, tìm kiếm các hóa đơn. Có thể nhận giao những đơn hàng mà admin đã xác nhận và xác nhận đã giao hàng khi giao hàng thành công

### Đối với khách hàng:

Có thể nói thì hiện tại website trong dự án lần này nhóm chúng em chỉ thiết kế giao diện dành cho Nhân viên quản lý mà thôi, chúng em vẫn chưa kịp thiết kế giao diện cho khách hàng. Tuy nhiên thì nhóm chúng em sẽ cố gắng hoàn thiện thêm phần này trong tương lai, dù là chưa hoàn hảo vẫn mong nhận được sự thông cảm của giảng viên hướng dẫn cũng như là những người đang đọc dòng chữ này.

## Các chức năng:

### Chức năng đối với nhân viên Admin:

#### Quản lý tài khoản:

Quản trị viên đăng nhập vào hệ thống bằng cách cung cấp địa chỉ tên đăng nhập, password vào form đăng nhập được xây dựng sẵn. Kiểm tra thông tin vào có hợp lệ hay không. Nếu có chuyển đến trang quản trị ngược lại thông báo lỗi và yêu cầu nhập lại. Quản trị viên có thể đăng ký tài khoản cho các nhân viên khác. Khi đó admin cần nhập các thông tin: họ tên, ngày sinh, địa chỉ, sđt, email, tên đăng nhập, password và cofirm pasword của nhận viên cần đăng ký tài khoản, cùng với đó là nhân viên đó có phải là admin hay không.

#### Quản lý sản phẩm:

Quản trị viên có quyền xem, thêm, sửa và tìm kiếm sản phẩm:

Thêm: Kiểm tra tính hợp lệ của thông tin sản phẩm mới. Kiểm tra có tồn tại hay chưa? Nếu chưa thêm mới sản phẩm vào CSDL.

Sửa: Kiểm tra tính hợp lệ của thông tin nhưng thuộc tính cần sửa- Nếu hợp lệ thì cập nhật thông tin sản phẩm vào CSDL.

Xem: Lấy từ CSDL ra tất cả sản phẩm, và có thể xem chi tiết những sản phẩm đó.

Tìm kiếm: Nhận những thông tin muốn tìm kiếm: mã sản phẩm, tên sản phẩm, loại sản phẩm. Hiện thị từ CSDL những sản phẩm khớp với những yêu cầu tìm kiếm.

#### Quản lý hóa đơn:

Quản trị viên có thể xem, tìm kiếm:

Xem: Hiển thị tất cả các hóa đơn, có thể xem chi tiết từng hóa đơn.

Tìm kiếm: Nhận những thông tin về đơn hàng cần tìm kiếm: mã hóa đơn, mã nhân viên gọi món, thời gian gọi món.

#### Quản lý nhân viên:

Quản trị viên có quyền xem danh sách các nhân viên và chi tiết các nhân viên đó được lấy ra từ CSDL.

#### Thống kê doanh thu:

Quản trị viên sẽ nhập vào khoảng thời gian muốn thống kê doanh thu, từ ngày nào đến ngày nào, và từ CSDL sẽ tra về nhưng hóa đơn có trong khoảng thời gian đó, cùng với tổng doanh thu.

## Phân Tích hệ thống:

### Các yêu cầu chức năng chính của hệ thống:

#### Yêu cầu lưu trữ:

Hệ thống cần lưu trữ các thông tin liên quan đến sản phẩm, loại sản phẩm, thông tin đăng nhập quản trị viên, nhân viên, các hóa đơn gọi món.

Phần quản trị của website cần đảm bảo tính bảo mật do có thể truy cập tới toàn bộ thông tin về sản phẩm và hoạt động kinh doanh:

Thông tin NhanVien (NhanVienID, HoTen, NgaySinh, GioiTinh, SoDienThoai, DiaChi, AnhNV)

Thông tin KhachHang (KhachHangID, HoTen, SoDienThoai, DiaChi)

Thông tin SanPham (SanPhamID, TenSanPham, Gia, MoTa, HinhAnh)

Thông tin HoaDon (HoaDonID, NgayLap, ThoiGianLap, NhanVienID, KhachHangID, TongTien)

Thông tin ChiTietHoaDon (ChiTietHoaDonID, HoaDonID, SanPhamID, SoLuong, DonGia, ThanhTien)

Thông tin Users (UserID, Username, Password, Role, NhanVienID)

Thông tin Permissions (PermissionID, UserID, CanAddProduct, CanOrder, CanCreateInvoice, CanViewInvoiceDetails)

Thông tin QuanTri (Email, Admin, HoTen, Password)

#### Yêu cầu về tự động hóa:

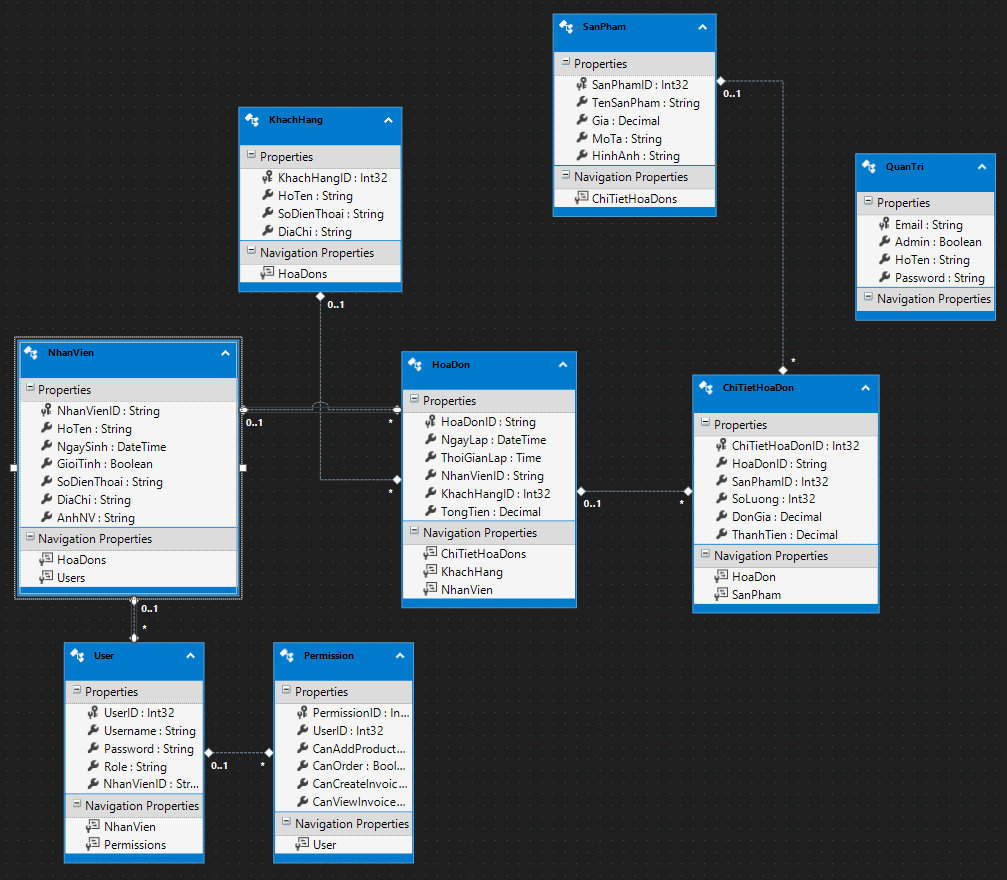
Tính năng tự động hóa như là: tự động điền, tính toán dự trên thông tin sản phẩm, ngày giờ, thời gian của hóa đơn được tự động điền, hay là mã số của nhân viên, mã hóa đơn. Có thể nói tính năng tự động hóa này sẽ là một tính năng cần thiết cho một website quản lý như thế này.

#### Tìm kiếm và tra cứu:

Chọn thông tin tra cứu về hóa đơn. Kết quả sau khi tra cứu: Danh sách các hóa đơn (nếu có) theo tiêu chuẩn đã được quản trị viên đưa ra.

## Thiết kế cơ sở dữ liệu:

### Mô hình cơ sở dữ liệu:



Biểu đồ dữ liệu quan hệ

### Danh sách các đối tượng:

#### Lớp nhân viên:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | NhanVienID | NVARCHAR(10) | Khóa chính | Mã nhân viên |
| 2 | HoTen | NVARCHAR(100) | NOT NULL | Họ và tên nhân viên |
| 3 | NgaySinh | DATE |  | Ngày sinh của nhân viên |
| 4 | GioiTinh | BIT |  | Giới tính của nhân viên |
| 5 | SoDienThoai | NVARCHAR(15) |  | Số điện thoại của nhân viên |
| 6 | DiaChi | NVARCHAR(255) |  | Địa chỉ của nhân viên |
| 7 | AnhNV | NVARCHAR(50) |  | Hình ảnh nhân viên |

#### Lớp khách hàng:

#### Lớp sản phẩm:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | KhachHangID | NVARCHAR(10) | Khóa chính | Mã khách hàng |
| 2 | HoTen | NVARCHAR(100) |  | Họ và tên khách hàng |
| 3 | SoDienThoai | NVARCHAR(15) |  | Số điện thoại của khách hàng |
| 4 | DiaChi | NVARCHAR(255) |  | Địa chỉ của khách hàng |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | SanPhamID | INT(10) | Khóa chính | Mã sản phẩm |
| 2 | TenSanPham | NVARCHAR(100) | NOT NULL | Tên sản phẩm |
| 3 | Gia | DECIMAL(18, 2) | NOT NULL | Giá sản phẩm |
| 4 | MoTa | NVARCHAR(255) |  | Mô tả sản phẩm |
| 5 | HinhAnh | NVARCHAR(255) |  | Hình ảnh sản phẩm |

#### Lớp Chi tiết hóa đơn:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | ChiTietHoaDonID | INT | Khóa chính | Mã chi tiết hóa đơn |
| 2 | HoaDonID | NVARCHAR(10) | Khóa ngoại | Mã hóa đơn |
| 3 | SanPhamID | INT | Khóa ngoại | Mã sản phẩm |
| 4 | SoLuong | INT | NOT NULL | Số lượng sản phẩm |
| 5 | DonGia | DECIMAL(18, 2) | NOT NULL | Đơn giá sản phẩm |
| 6 | ThanhTien | DECIMAL(18, 2) | NOT NULL | Thành tiền |

#### Lớp User:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | UserID | INT | Khóa chính | Mã người dùng |
| 2 | Username | NVARCHAR(50) | NOT NULL, UNIQUE | Tên đăng nhập |
| 3 | Password | NVARCHAR(255) | NOT NULL | Mật khẩu |
| 4 | Role | NVARCHAR(50) | NOT NULL | Vai trò của người dùng |
| 5 | NhanVienID | NVARCHAR(10) | NOT NULL | Mã nhân viên |

#### Lớp quyền hạn:

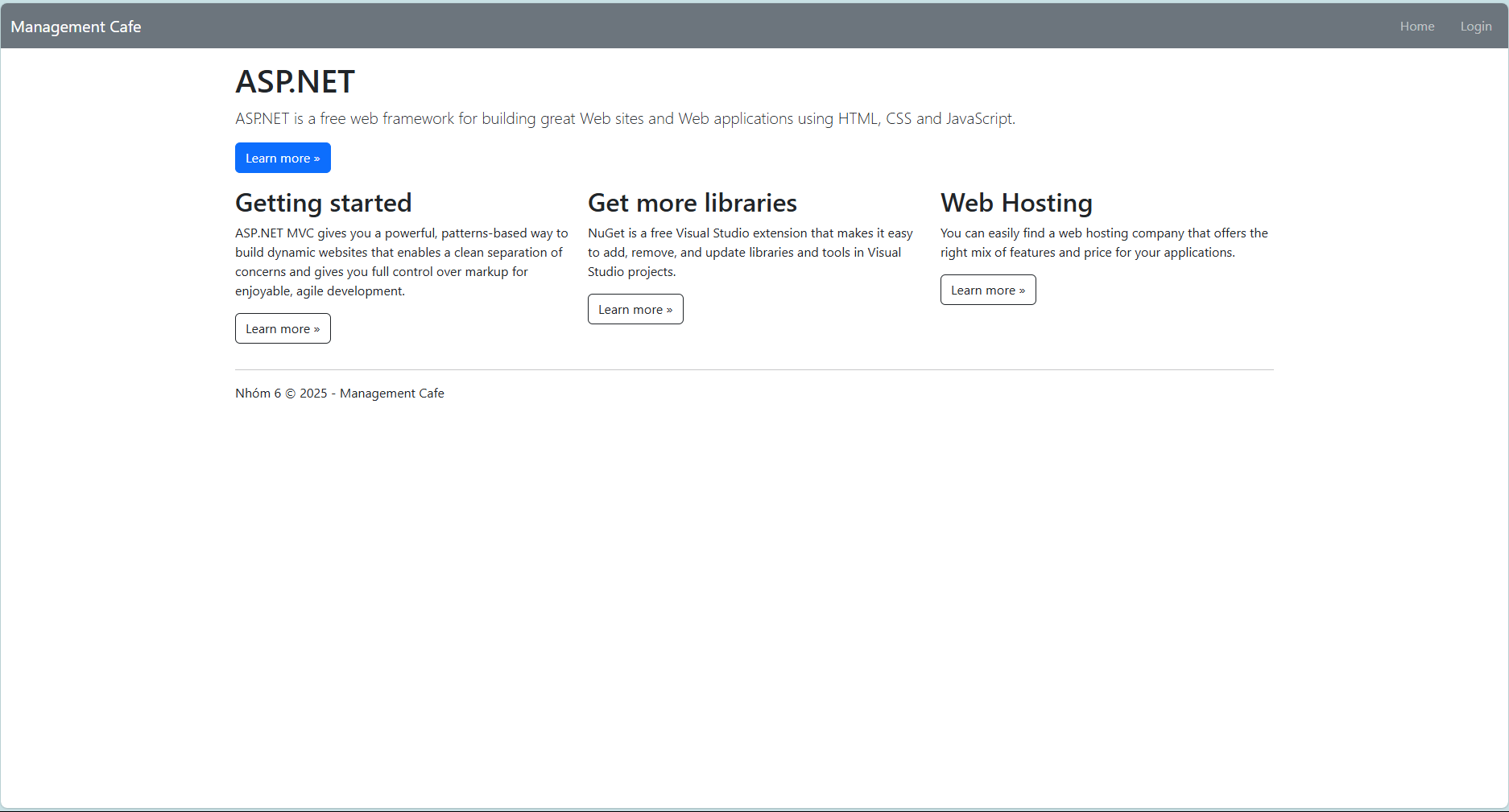
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | PermissionID | INT | Khóa chính | Mã quyền |
| 2 | UserID | INT | Khóa ngoại | Mã người dùng |
| 3 | CanAddProduct | BIT | DEFAULT 0 | Quyền thêm sản phẩm |
| 4 | CanOrder | BIT | DEFAULT 0 | Quyền đặt hàng |
| 5 | CanCreateInvoice | BIT | DEFAULT 0 | Quyền tạo hóa đơn |
| 6 | CanViewInvoiceDetails | BIT | DEFAULT 0 | Quyền xem chi tiết hóa đơn |

#### Lớp quản trị

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Ý nghĩa |
| 1 | Email | VARCHAR(50) | Khóa chính | Email quản trị viên |
| 2 | Admin | BIT |  | Quyền quản trị |
| 3 | HoTen | NVARCHAR(50) |  | Họ và tên quản trị viên |
| 4 | Password | NVARCHAR(50) |  | Mật khẩu quản trị viên |

## Thiết kế giao diện:

### Giao diện trang chủ:



Giao diện trang chủ

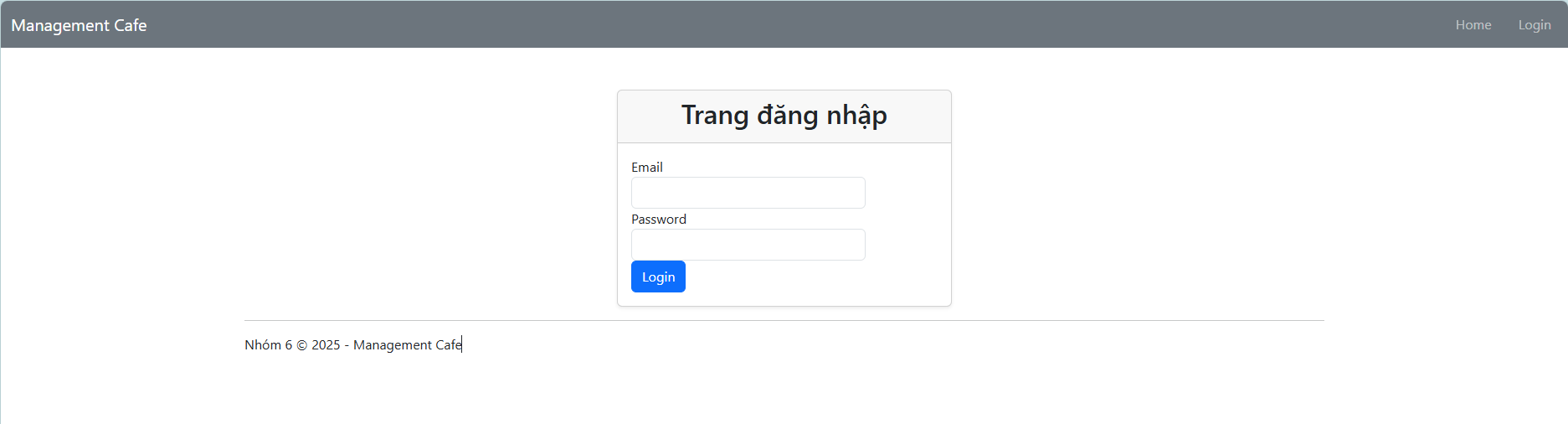
Trên là hình ảnh giao diện trang chủ của nhóm chúng em, có thể rằng là thầy cô hay mọi người sẽ thắc mắc tại sao lại là giao diện mặc định mà không phải một giao diện được cá nhân hóa, thậm chí trong hình ảnh trên chắc chỉ có thứ duy nhất được thay đổi là màu sắc và chữ “ApplicationName” bên góc trái trên cùng thành “ManagementCafe”, để nói về điều này thì có thể nói như sau:

Giao diện web trang chủ mặc định của ASP.NET là một template cơ bản, được thiết kế để giúp các lập trình viên khởi đầu dự án một cách nhanh chóng và dễ dàng. Giao diện này thường bao gồm một trang chủ đơn giản với các thành phần cơ bản như thanh điều hướng, phần giới thiệu, và các liên kết tới các trang khác như trang đăng nhập, đăng ký và về chúng tôi.

Trang chủ mặc định thường có màu sắc trung tính, bố cục rõ ràng và dễ đọc, giúp người dùng dễ dàng điều hướng và tìm kiếm thông tin. Bố cục này thường tuân theo các chuẩn thiết kế web hiện đại, với việc sử dụng CSS và JavaScript để tạo ra một giao diện người dùng thân thiện và tương tác.

### Giao diện đăng nhập:

#### Tại sao lại phải đăng nhập?



Giao diện trang đăng nhập

**Bảo mật:** Đăng nhập giúp bảo vệ thông tin cá nhân và dữ liệu quan trọng của người dùng khỏi sự truy cập trái phép. Chỉ những người dùng đã xác thực mới có quyền truy cập vào các tài nguyên và chức năng của hệ thống, đảm bảo rằng thông tin nhạy cảm luôn được an toàn.

**Tùy chỉnh trải nghiệm:** Việc đăng nhập cho phép hệ thống nhận diện người dùng và cung cấp trải nghiệm tùy chỉnh dựa trên thông tin và sở thích của họ. Ví dụ, sau khi đăng nhập, người dùng có thể thấy các thông báo cá nhân, đơn hàng của họ, và các tùy chọn thiết lập cá nhân.

**Quản lý người dùng:** Đối với quản trị viên và những người quản lý hệ thống, việc người dùng phải đăng nhập giúp họ theo dõi và quản lý hoạt động của người dùng. Điều này có thể bao gồm việc kiểm soát quyền truy cập, theo dõi hành vi người dùng, và đảm bảo rằng tất cả các hoạt động đều được ghi nhận và quản lý một cách minh bạch.

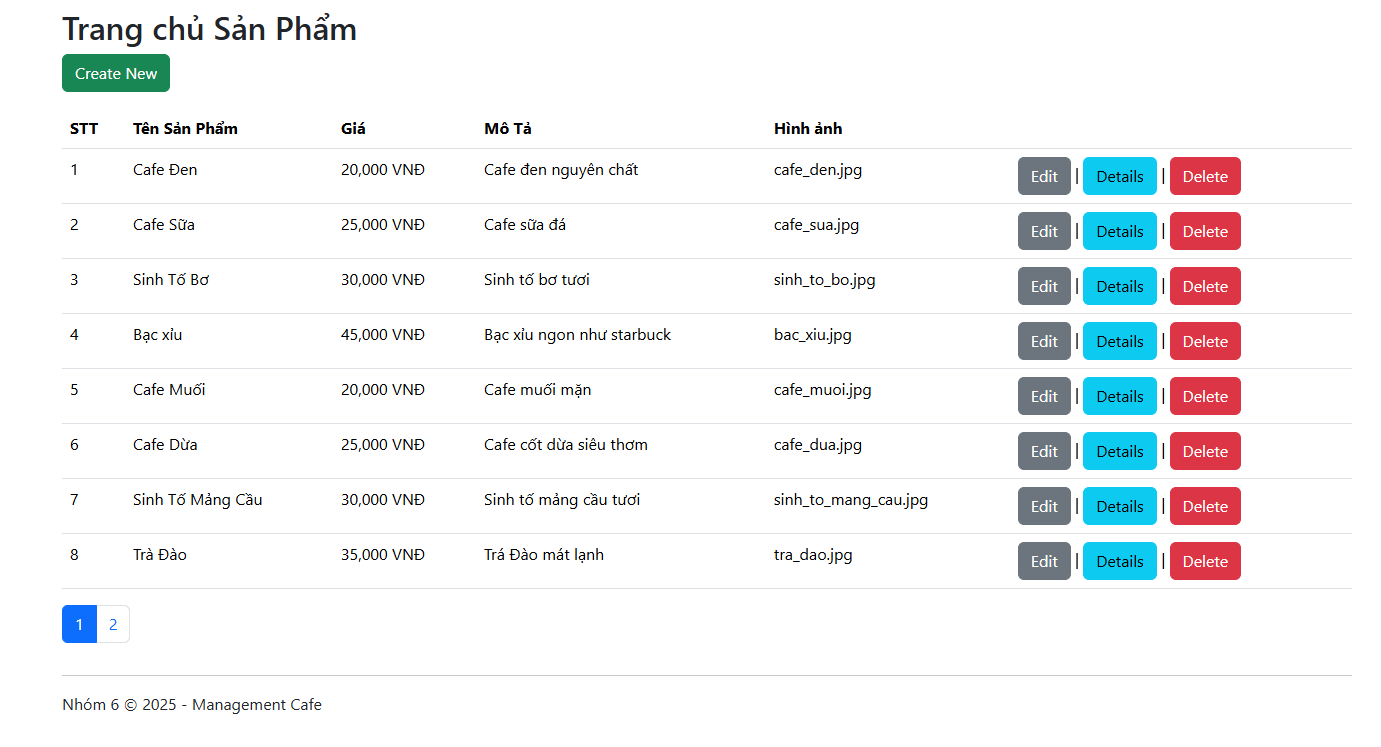
**Bảo vệ khỏi các cuộc tấn công:** Đăng nhập giúp giảm thiểu nguy cơ bị tấn công từ các kẻ xâm nhập. Hệ thống có thể triển khai các biện pháp bảo mật như xác thực hai yếu tố (2FA), mã hóa dữ liệu, và các cơ chế bảo vệ khác để đảm bảo rằng chỉ những người dùng hợp lệ mới có thể truy cập vào tài khoản của họ.

**Cải thiện dịch vụ:** Thông qua việc đăng nhập, các doanh nghiệp có thể thu thập dữ liệu và phân tích hành vi người dùng để cải thiện dịch vụ và phát triển các tính năng mới. Điều này giúp nâng cao chất lượng sản phẩm và đáp ứng tốt hơn nhu cầu của người dùng.

#### Trang đăng ký:

Thực sự rất tiếc khi phải nói rằng là nhóm chúng em vẫn chưa hoàn thành kịp chức năng cũng như giao diện trang đăng ký. Rất mong thầy cô và mọi người sẽ hiểu và thông cảm cho nhóm chúng em.

### Trang danh sách sản phẩm:



Giao diện trang danh sách sản phẩm

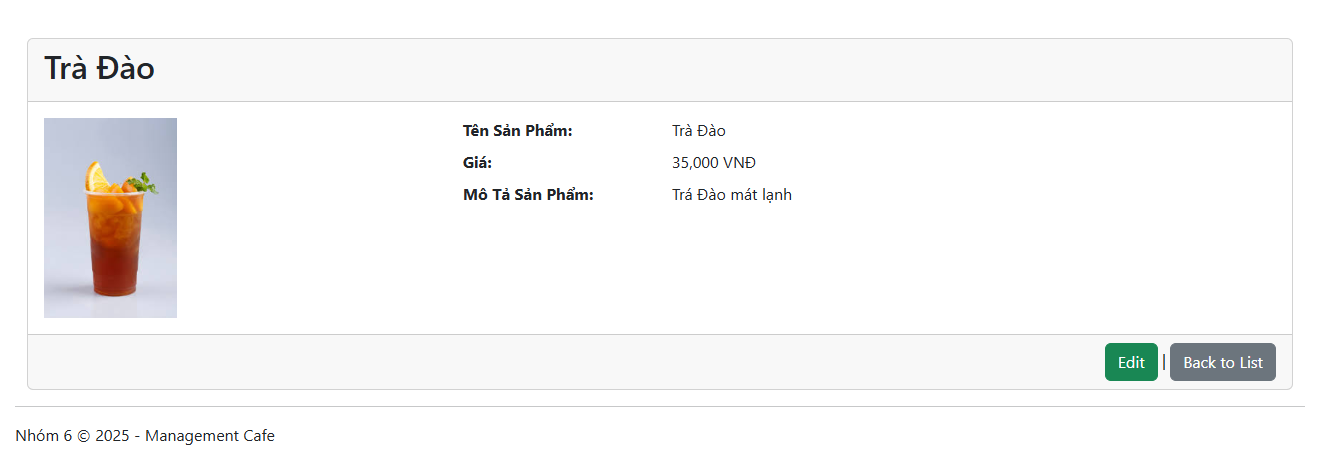
Giao diện trang list sản phẩm trong hình trên là một bảng quản lý sản phẩm của một quán cafe. Bảng này bao gồm các cột: STT (số thứ tự), Tên Sản Phẩm, Giá, Mô Tả, Hình Ảnh, và các nút chức năng (Edit, Details, Delete) cho mỗi sản phẩm.

Phía trên cùng của bảng có tiêu đề "Trang chủ Sản Phẩm" và một nút "Create New" màu xanh lá cây để thêm sản phẩm mới.

Mỗi sản phẩm có các nút chức năng "Edit" (chỉnh sửa), "Details" (chi tiết), và "Delete" (xóa) để quản lý sản phẩm.

Phía dưới bảng có phân trang với số trang hiện tại là 1 và có thể chuyển sang trang 2.

**Mục đích**: Trang list sản phẩm được thiết kế nhằm mục đích cung cấp cho người quản lý một cái nhìn tổng quan về toàn bộ các sản phẩm đang được kinh doanh. Qua đó, người quản lý có thể dễ dàng theo dõi, kiểm tra và cập nhật thông tin về sản phẩm một cách nhanh chóng và hiệu quả.

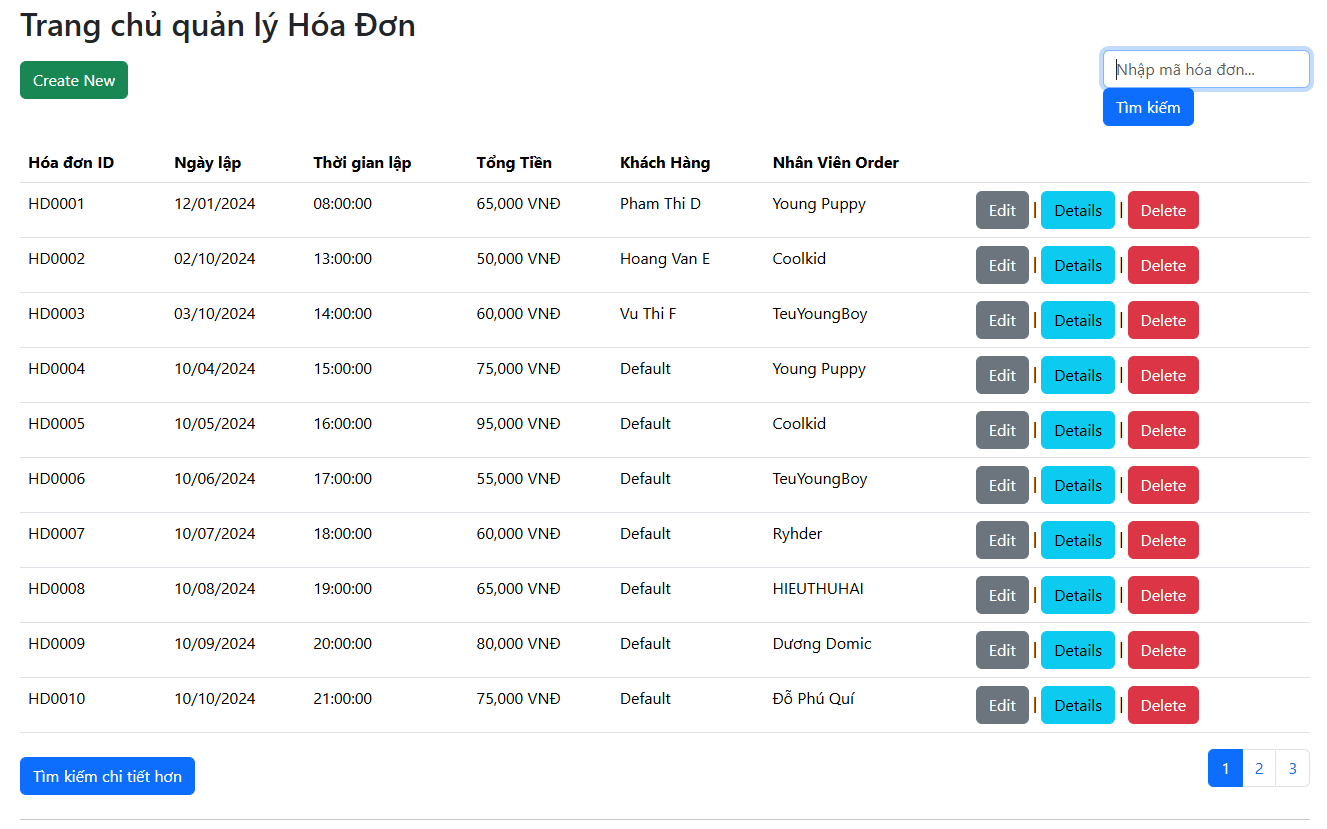


Giao diện trang chi tiết sản phẩm

**Mục đích**: hiện thị thông tin chi tiết của một sản phẩm như: mô tả, cấu tạo, công dụng, giá bán, …

Khi muốn xem chi tiết thông tin sản phẩm bạn chọn chi tiết sẽ có các thông tin liên quan đến sản phẩm.

### Trang Hóa Đơn:



Giao diện trang Hóa Đơn

Trang hóa đơn trong ảnh là một trang quản lý hóa đơn, có mục đích giúp người dùng theo dõi và quản lý các hóa đơn đã lập. Trang này cho phép người dùng tạo mới, chỉnh sửa, xem chi tiết và xóa các hóa đơn.

Ngoài ra, trang còn có các nút chức năng như "Edit" (Chỉnh sửa), "Details" (Chi tiết), và "Delete" (Xóa) để người dùng có thể thực hiện các thao tác quản lý hóa đơn một cách dễ dàng. Trang cũng có chức năng tìm kiếm hóa đơn theo mã và nút "Create New" để tạo hóa đơn mới.

**Mục đích**:

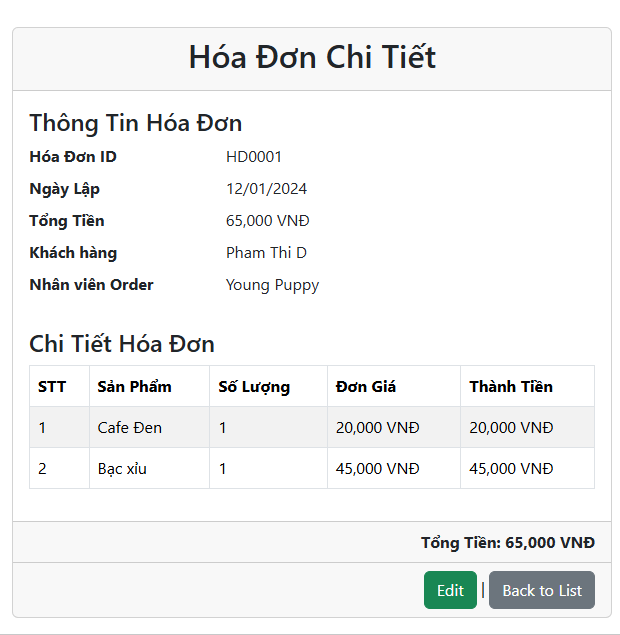
 **Quản lý hóa đơn:** Giúp người dùng dễ dàng theo dõi, kiểm tra và cập nhật thông tin về các hóa đơn đã lập. Điều này giúp đảm bảo rằng tất cả các giao dịch được ghi nhận và quản lý một cách chính xác.

 **Tối ưu hóa quy trình làm việc:** Cung cấp các chức năng tạo mới, chỉnh sửa và xóa hóa đơn, giúp người dùng quản lý các hóa đơn một cách hiệu quả và nhanh chóng. Điều này giảm thiểu thời gian và công sức trong việc xử lý hóa đơn.

 **Cải thiện trải nghiệm người dùng:** Giao diện trực quan và dễ sử dụng giúp người dùng dễ dàng điều hướng và thực hiện các thao tác cần thiết. Các thông tin quan trọng được hiển thị rõ ràng, giúp người dùng nắm bắt nhanh chóng tình hình hóa đơn.

 **Báo cáo và phân tích:** Tính năng tìm kiếm và lọc hóa đơn giúp người dùng dễ dàng tìm kiếm và phân tích các hóa đơn theo nhiều tiêu chí khác nhau. Điều này hỗ trợ việc báo cáo và đưa ra các quyết định kinh doanh dựa trên dữ liệu thực tế.

Tóm lại, trang hóa đơn không chỉ giúp quản lý và theo dõi các hóa đơn một cách hiệu quả mà còn tối ưu hóa quy trình làm việc và cải thiện trải nghiệm người dùng. Điều này đóng vai trò quan trọng trong việc quản lý tài chính và hỗ trợ các quyết định kinh doanh của quán cafe.



Giao diện chi tiết hóa đơn

Trang chi tiết hóa đơn trong ảnh hiển thị thông tin chi tiết về một hóa đơn cụ thể. Mục đích của trang này là cung cấp cho người dùng thông tin chi tiết về giao dịch mua hàng, bao gồm các mục đã mua, số lượng, đơn giá và tổng tiền.

Trang này cũng có các nút chức năng "Edit" và "Back to List" để chỉnh sửa hóa đơn hoặc quay lại danh sách hóa đơn.

Mục đích của trang chi tiết hóa đơn:

**Cung cấp thông tin chi tiết:** Giúp người dùng xem các thông tin chi tiết về hóa đơn, bao gồm các sản phẩm đã mua, số lượng, đơn giá và tổng tiền. Điều này giúp người dùng nắm rõ nội dung của mỗi giao dịch mua hàng.

**Xác minh và kiểm tra:** Cho phép người dùng xác minh các chi tiết giao dịch, đảm bảo rằng tất cả các mục trong hóa đơn là chính xác và đúng theo yêu cầu. Điều này cũng giúp phát hiện và khắc phục các sai sót nếu có.

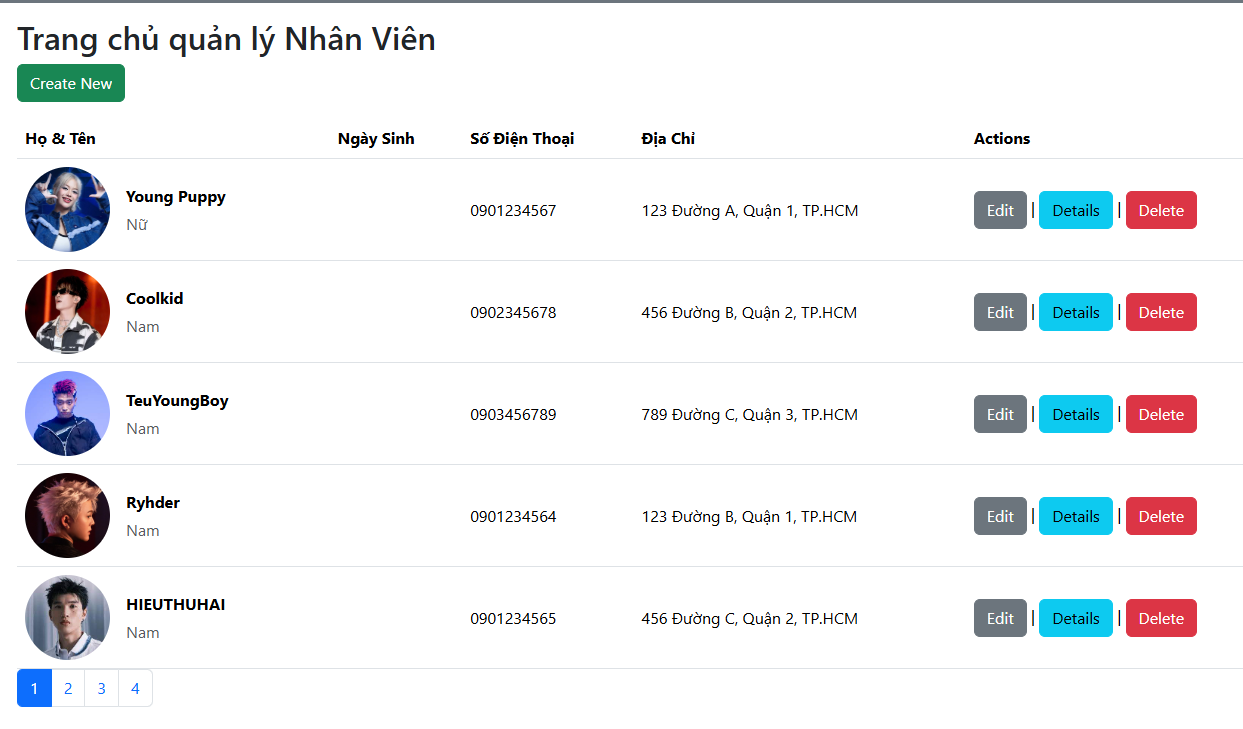
**Quản lý và theo dõi:** Cung cấp các công cụ để người dùng chỉnh sửa hoặc cập nhật thông tin hóa đơn, đảm bảo rằng tất cả các thông tin luôn được cập nhật và chính xác.

**Hỗ trợ báo cáo và phân tích:** Cung cấp thông tin chi tiết để người dùng có thể thực hiện các báo cáo và phân tích về các giao dịch mua hàng, từ đó đưa ra các quyết định kinh doanh dựa trên dữ liệu thực tế.

Tóm lại, trang chi tiết hóa đơn không chỉ cung cấp thông tin chi tiết về các giao dịch mua hàng mà còn hỗ trợ quản lý, kiểm tra và báo cáo, giúp nâng cao hiệu quả và độ chính xác trong việc quản lý tài chính của quán cafe.

## Giao diện trang Nhân Viên:

### Trang chủ quản lý nhân viên:



Giao diện trang quản lý nhân viên

Trang danh sách nhân viên trong ảnh là một giao diện quản lý nhân viên. Trang này hiển thị thông tin cơ bản của các nhân viên bao gồm họ và tên, ngày sinh, số điện thoại, địa chỉ và các hành động có thể thực hiện như chỉnh sửa, xem chi tiết và xóa. Mỗi nhân viên được liệt kê trong một hàng riêng biệt với các nút chức năng tương ứng. Trang này giúp người quản lý dễ dàng theo dõi và quản lý thông tin nhân viên một cách hiệu quả.

Mục đích của trang danh sách nhân viên:

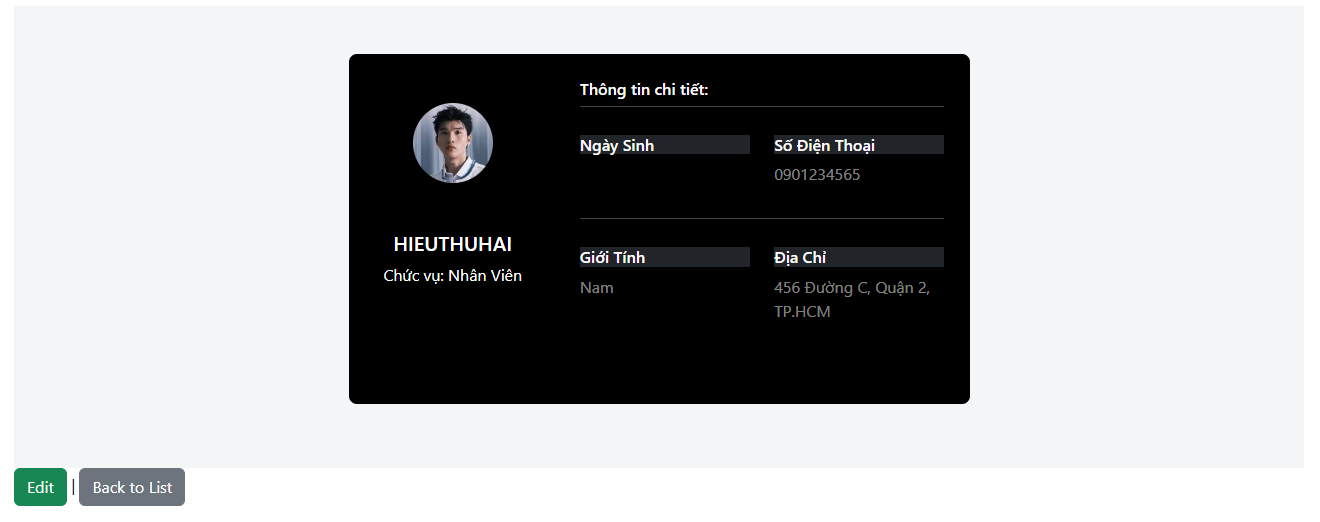
**Quản lý thông tin nhân viên:** Trang này cho phép người quản lý xem toàn bộ danh sách nhân viên, từ đó dễ dàng theo dõi và cập nhật thông tin cá nhân của từng nhân viên.

**Thực hiện các hành động quản lý:** Cung cấp các nút chức năng như chỉnh sửa, xem chi tiết và xóa, giúp người quản lý thực hiện các tác vụ quản lý nhân viên một cách nhanh chóng và hiệu quả.

**Tối ưu hóa quy trình làm việc:** Giúp người quản lý tiết kiệm thời gian và công sức trong việc tìm kiếm và quản lý thông tin nhân viên, từ đó nâng cao hiệu suất làm việc.

**Cải thiện trải nghiệm người dùng:** Giao diện trực quan, dễ sử dụng giúp người quản lý dễ dàng điều hướng và thực hiện các thao tác cần thiết một cách thuận tiện.

Tóm lại, trang danh sách nhân viên không chỉ cung cấp một cái nhìn tổng quan về tất cả các nhân viên mà còn hỗ trợ người quản lý trong việc quản lý thông tin và thực hiện các tác vụ liên quan một cách hiệu quả và thuận tiện.



Giao diện trang chi tiết nhân viên

Trang danh sách nhân viên trong ảnh hiển thị thông tin chi tiết của một nhân viên cụ thể. Trang này bao gồm ảnh đại diện của nhân viên, tên, chức vụ, và các thông tin cá nhân như ngày sinh, số điện thoại, giới tính và địa chỉ. Có hai nút chức năng ở phía dưới trang: "Edit" để chỉnh sửa thông tin và "Back to List" để quay lại danh sách nhân viên.

Mục đích của trang danh sách nhân viên:

**Quản lý thông tin nhân viên:** Trang này giúp người quản lý dễ dàng theo dõi và cập nhật thông tin cá nhân của từng nhân viên, từ đó đảm bảo dữ liệu luôn được cập nhật và chính xác.

**Xác minh và kiểm tra:** Cho phép người quản lý xác minh và kiểm tra các chi tiết nhân viên, đảm bảo rằng thông tin được nhập đúng và đầy đủ.

**Thực hiện các tác vụ quản lý:** Cung cấp các chức năng như chỉnh sửa và quay lại danh sách, giúp người quản lý thực hiện các tác vụ quản lý nhân viên một cách nhanh chóng và hiệu quả.

**Cải thiện trải nghiệm người dùng:** Giao diện trực quan và dễ sử dụng giúp người quản lý tiết kiệm thời gian và công sức trong việc quản lý nhân viên, nâng cao hiệu suất làm việc.

Tóm lại, trang danh sách nhân viên không chỉ cung cấp một cái nhìn tổng quan về thông tin cá nhân của từng nhân viên mà còn hỗ trợ người quản lý trong việc quản lý, kiểm tra và cập nhật thông tin một cách hiệu quả và thuận tiện.

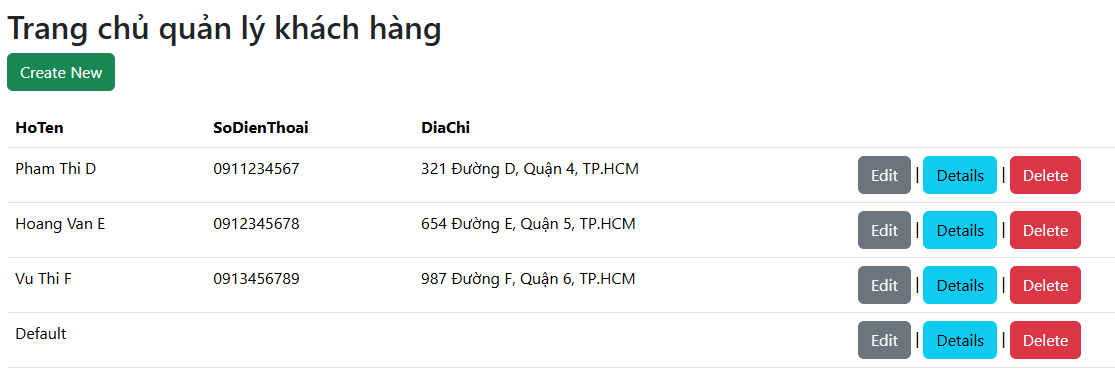
## Giao diện trang quản lý khách hàng:

Trang danh sách khách hàng trong ảnh hiển thị một bảng quản lý khách hàng với các cột bao gồm "HoTen" (Họ Tên), "SoDienThoai" (Số Điện Thoại), và "DiaChi" (Địa Chỉ). Mỗi hàng trong bảng đại diện cho một khách hàng với các thông tin tương ứng. Ngoài ra, mỗi hàng còn có các nút "Edit" (Chỉnh sửa), "Details" (Chi tiết), và "Delete" (Xóa) để quản lý thông tin khách hàng.

### Mục đích của trang danh sách khách hàng:

#### Quản lý thông tin khách hàng:

Giúp người quản lý dễ dàng theo dõi và cập nhật thông tin cá nhân của khách hàng. Từ đó, đảm bảo dữ liệu luôn được cập nhật và chính xác.



Giao diện trang quản lý khách hàng

#### Thực hiện các tác vụ quản lý:

Cung cấp các chức năng như chỉnh sửa, xem chi tiết và xóa, giúp người quản lý thực hiện các tác vụ quản lý khách hàng một cách nhanh chóng và hiệu quả.

#### Dễ dàng tìm kiếm và lọc thông tin:

Bảng dữ liệu cho phép người quản lý dễ dàng tìm kiếm và lọc thông tin khách hàng theo các tiêu chí khác nhau, giúp việc quản lý trở nên tiện lợi hơn.

#### Cải thiện trải nghiệm người dùng:

Giao diện trực quan, dễ sử dụng giúp người quản lý tiết kiệm thời gian và công sức trong việc quản lý khách hàng, nâng cao hiệu suất làm việc.

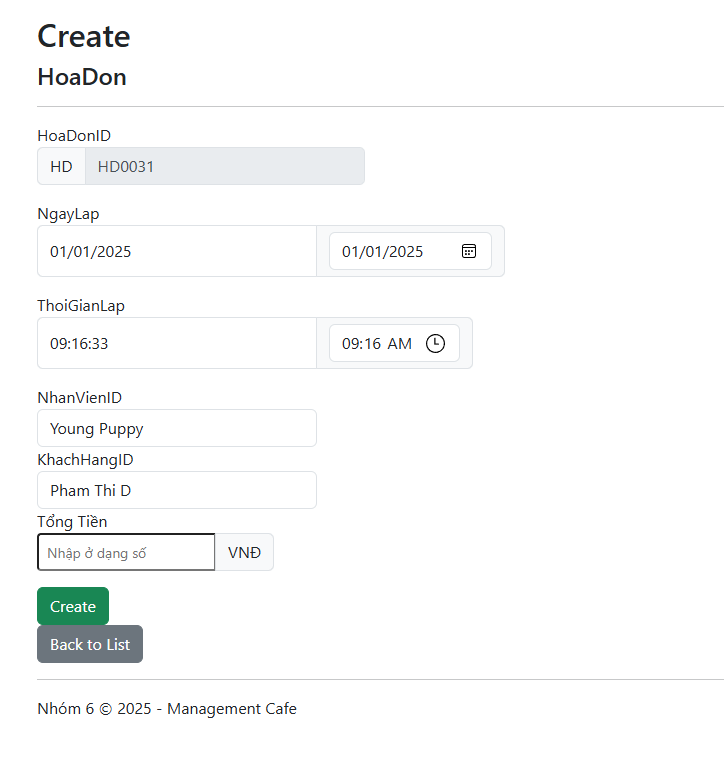
### Lý do nên có một ID khách hàng mang tên "Default":

Việc có một ID khách hàng mang tên "Default" là cần thiết vì nó có thể được sử dụng như một bản ghi tạm thời hoặc mặc định khi thông tin khách hàng chưa được xác định hoặc khi cần tạo hóa đơn mà không có thông tin khách hàng cụ thể. Sử dụng tên khách hàng "Default" cho mọi hóa đơn giúp đảm bảo rằng hệ thống không bị gián đoạn và có thể tiếp tục hoạt động mà không cần phải chờ đợi thông tin khách hàng đầy đủ. Điều này cũng giúp dễ dàng theo dõi và quản lý các giao dịch tạm thời hoặc chưa hoàn tất.

Tóm lại, trang danh sách khách hàng không chỉ giúp quản lý thông tin khách hàng một cách hiệu quả mà còn hỗ trợ các tác vụ quản lý, tìm kiếm và cập nhật thông tin một cách thuận tiện. Việc sử dụng ID khách hàng "Default" giúp đảm bảo tính liên tục và hiệu quả trong việc quản lý hóa đơn và giao dịch.

## Những trang được tự động hóa:

### Trang Create của trang Hóa Đơn:



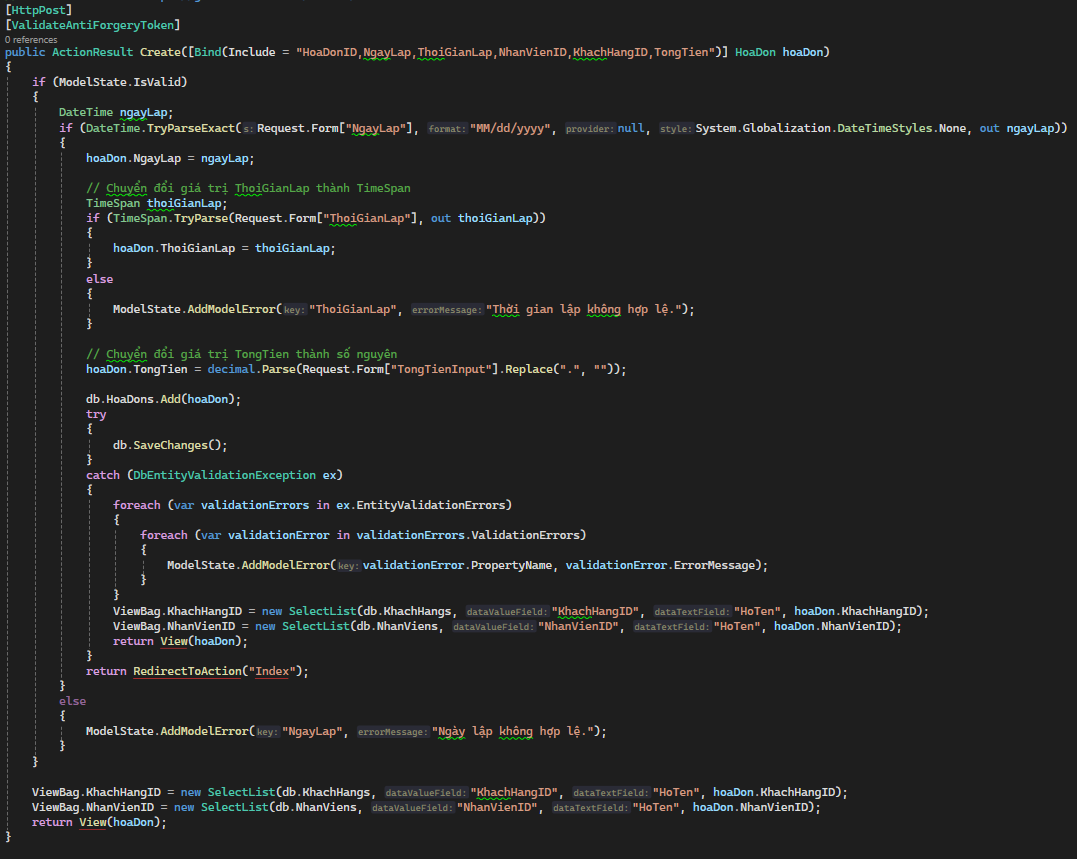
Trang Create của trang quản lý Hóa Đơn

Như bạn có thể thấy như trên thì trang Create thường sẽ được tự động hóa như những phần ID, phần ngày lập và thời gian lập sẽ được lấy tự động theo múi giờ của máy tính dưới đây.

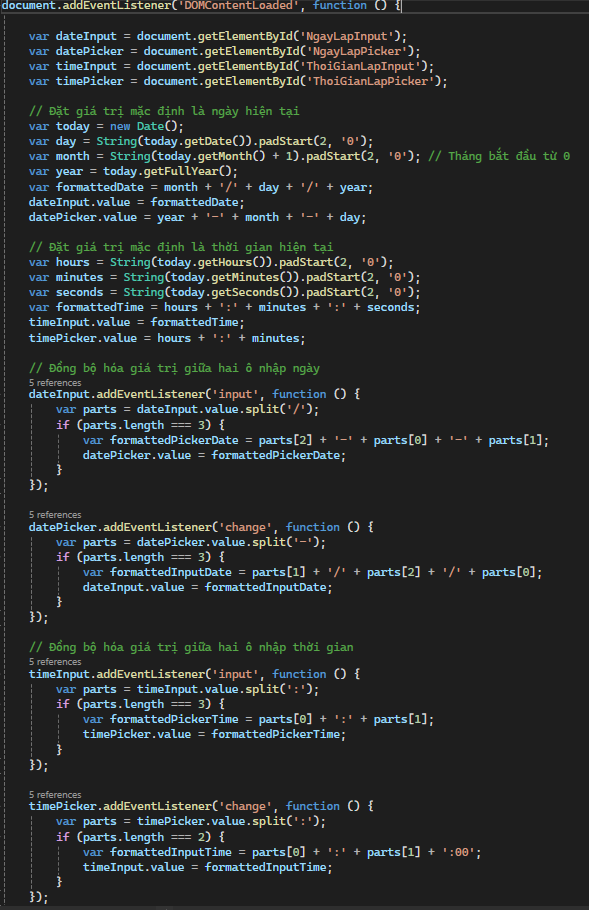


Múi thời gian trong máy tính dùng để lấy mốc tự động hóa cho trang Create Hoa Don

Và làm thì cơ bản chúng ta cần kết hợp cả Controller của ngôn ngữ C# với ngôn ngữ của frontend là Javascripts để có thể lấy được mốc thời gian trong máy tính và chuyển nó để file Controller để xử lý:

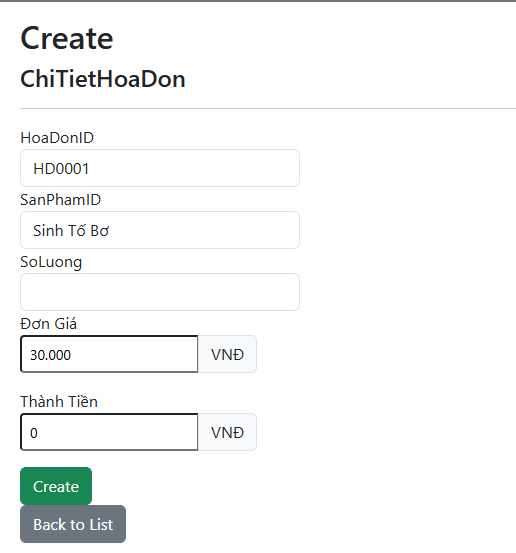
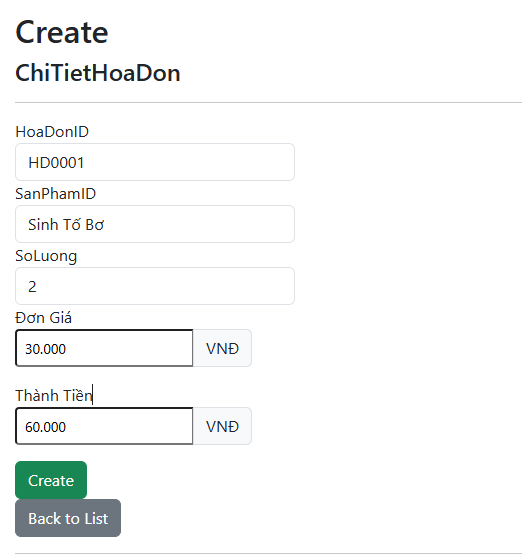


Đoạn code Create trong file Controller của trang Hóa Đơn



Code Javascripts tự động điền thời gian và ngày lập

### Trang Create của trang Chi tiết hóa đơn:



Trang Create của trang Chi tiết hóa đơn

Như trong ảnh trên thì có thể thấy rằng khi chúng ta chọn sản phẩm xong và nhập số lượng vào thì hệ thống sẽ tự động điền giúp chúng ta mục Thành tiền theo công thức “ Đơn giá x số lượng = Thành tiền ”.

Và để làm được chức năng như trên thì chúng ta sẽ cần sự hỗ trợ đến từ ngôn ngữ Javascripts cũng như là JSON của C#:



Đoạn code Javascripts giúp tự động điền cho  
mục Thành Tiền của Chi Tiết Hóa đơn

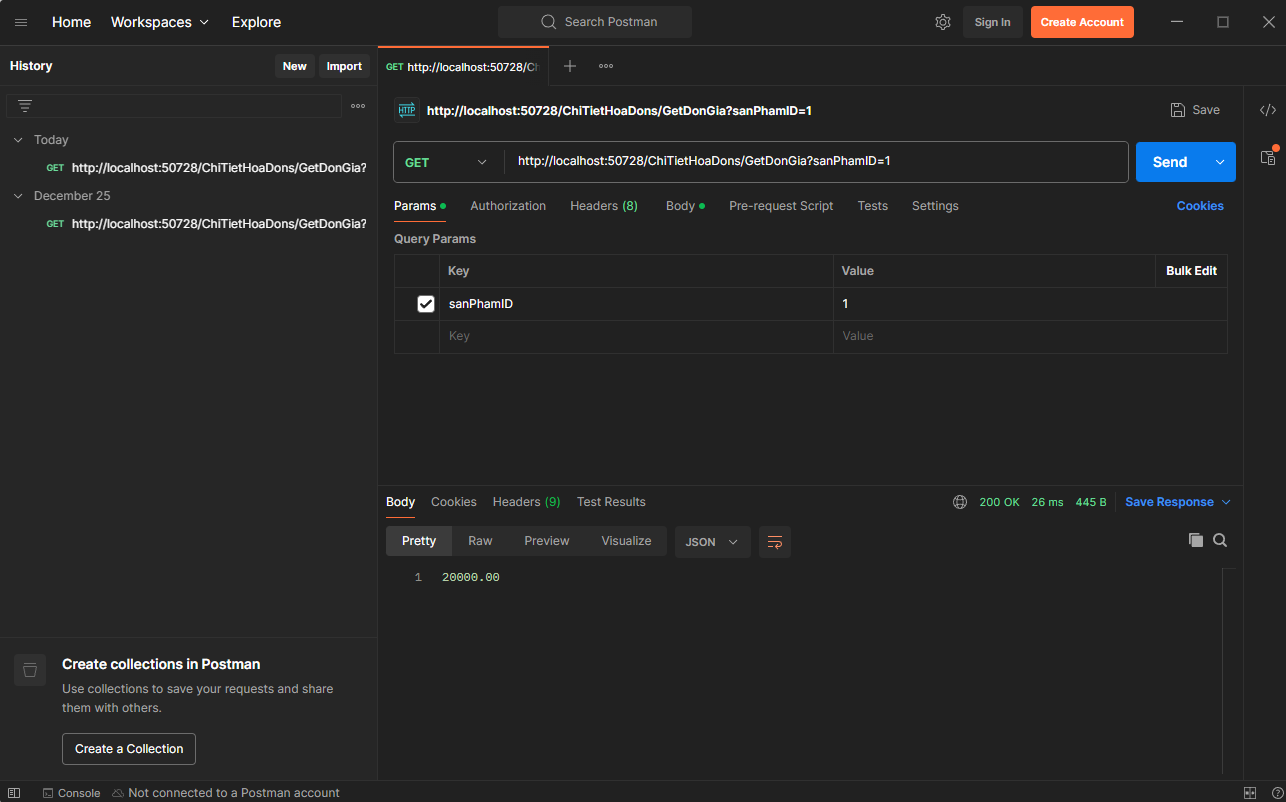


Đoạn code trả về JSON

Ở đây thì đoạn code trên sẽ trả về giá trị JSON để chúng ta có thể tính toán ra giá trị Thành tiền, nếu bạn muốn chắc chắn rằng là code của chúng ta đã đúng thì các bạn có thể chạy code và sử dụng công cụ mạnh mẽ để thử nghiệm mang tên Postman:



Logo Postman



Ảnh kết quả sau khi test bằng Postman

Như các bạn có thấy thì kết quả đã trả ra đúng như trong CSDL của chúng ta.

**Lý do mà chúng ta nên sử dụng JSON trong trường hợp này**:

**Không cần tải lại trang**: JSON có thể được sử dụng để truyền dữ liệu giữa máy chủ và trình duyệt mà không cần tải lại toàn bộ trang. Điều này giúp tối ưu hoá trải nghiệm người dùng và cải thiện hiệu suất.

**Cấu trúc dữ liệu đơn giản**: JSON có cấu trúc rất đơn giản và dễ đọc, phù hợp cho cả máy và con người. Dữ liệu được tổ chức thành các cặp "key-value", tương tự như các đối tượng trong JavaScript.

**Tương thích nhiều ngôn ngữ**: JSON không chỉ sử dụng trong JavaScript mà còn tương thích với nhiều ngôn ngữ lập trình khác như Python, Java, Ruby, C#, và nhiều ngôn ngữ khác.

**Dễ dàng phân tích và tạo dữ liệu**: Các thư viện JSON trong hầu hết các ngôn ngữ lập trình đều cung cấp các hàm tiện ích để phân tích (parse) và tạo (generate) dữ liệu JSON một cách dễ dàng.

**Nhẹ và nhanh**: JSON nhẹ hơn và nhanh hơn XML vì nó có cú pháp gọn gàng và không có các thẻ đóng mở phức tạp.

**Tương thích với AJAX**: JSON là định dạng lý tưởng để làm việc với AJAX (Asynchronous JavaScript and XML), giúp tải và hiển thị dữ liệu từ máy chủ một cách động mà không ảnh hưởng đến phần còn lại của trang web.

JSON thật sự hữu ích và dễ sử dụng, đặc biệt trong các ứng dụng web hiện đại! 🌟

# KẾT LUẬN

## Kết quả đạt được của đề tài:

Đề tài nghiên cứu là một hành trình khám phá, tìm hiểu và phân tích về một chủ đề cụ thể, thường dựa trên các phương pháp khoa học và logic chặt chẽ. Kết quả của một đề tài không chỉ phản ánh nỗ lực và sự cống hiến của người thực hiện, mà còn mang lại những giá trị to lớn cho cộng đồng nghiên cứu và ứng dụng thực tiễn.

### Tóm tắt kết quả nghiên cứu:

Phần này trình bày ngắn gọn những phát hiện chính của đề tài, cung cấp cái nhìn tổng quan về những gì đã được khám phá và những kết luận quan trọng rút ra từ nghiên cứu. Ví dụ, nếu đề tài nghiên cứu về hiệu quả của một phương pháp điều trị y học mới, tóm tắt kết quả sẽ nêu rõ những lợi ích và hạn chế chính của phương pháp này so với các phương pháp truyền thống.

### Phân tích chi tiết kết quả:

Phần này đi sâu vào phân tích các dữ liệu thu thập được trong quá trình nghiên cứu. Các biểu đồ, đồ thị và bảng số liệu được sử dụng để minh họa và giải thích các kết quả cụ thể. Việc phân tích chi tiết giúp làm rõ các mối quan hệ, xu hướng và những phát hiện đáng chú ý từ dữ liệu.

### Thảo luận và so sánh:

Trong phần thảo luận, kết quả của đề tài được so sánh với các nghiên cứu trước đó để xác định tính nhất quán hoặc mâu thuẫn. Điều này giúp đánh giá độ tin cậy và giá trị của kết quả nghiên cứu. Phần thảo luận cũng có thể nêu ra những yếu tố hạn chế của nghiên cứu và đề xuất hướng nghiên cứu tiếp theo.

### Kết luận và đề xuất:

Phần kết luận tổng hợp lại những kết quả quan trọng nhất của đề tài, khẳng định những đóng góp chính của nghiên cứu. Đồng thời, phần này cũng đưa ra những đề xuất cụ thể dựa trên kết quả nghiên cứu, nhằm ứng dụng vào thực tiễn hoặc tiếp tục phát triển nghiên cứu trong tương lai.

### Ứng dụng thực tiễn:

Một trong những yếu tố quan trọng của kết quả đề tài là khả năng ứng dụng vào thực tiễn. Ví dụ, nếu nghiên cứu về một giải pháp công nghệ mới, phần này sẽ trình bày các ứng dụng cụ thể và tiềm năng phát triển của giải pháp đó trong thực tế. Khả năng ứng dụng thực tiễn không chỉ tăng cường giá trị của nghiên cứu mà còn mở ra cơ hội cho sự đổi mới và phát triển.

### Đóng góp cho cộng đồng nghiên cứu:

Kết quả của đề tài không chỉ mang lại giá trị cho người thực hiện mà còn đóng góp vào kho tàng tri thức chung của cộng đồng nghiên cứu. Những phát hiện mới, phương pháp tiếp cận sáng tạo và dữ liệu nghiên cứu có thể được chia sẻ và sử dụng bởi các nhà nghiên cứu khác, góp phần thúc đẩy sự tiến bộ của khoa học và công nghệ.

### Hướng phát triển thêm:

Không có nghiên cứu nào là hoàn hảo từ đầu cả, việc nhìn nhận lại đề tài và tìm kiếm thêm những hướng phát triển trong tương lai sẽ là một điều cần thiết cho đề tài như thế này.

### Phản hồi và cải tiến:

Cuối cùng, kết quả của đề tài cũng cần được đánh giá và phản hồi từ cộng đồng chuyên gia và người sử dụng. Phản hồi tích cực và tiêu cực đều quan trọng để cải tiến và hoàn thiện nghiên cứu. Quá trình này không chỉ giúp nâng cao chất lượng của đề tài mà còn tạo cơ sở cho các nghiên cứu tiếp theo.

Tóm lại, kết quả của đề tài nghiên cứu không chỉ dừng lại ở những con số và dữ liệu, mà còn mở ra nhiều cơ hội và thách thức mới. Nó phản ánh quá trình làm việc chăm chỉ, sáng tạo và nhiệt huyết của người thực hiện, đồng thời mang lại những giá trị to lớn cho cộng đồng và ứng dụng thực tiễn.

## Hạn chế của đề tài:

Sẽ không có bất kỳ đề tài nào hoàn hảo từ đầu cả mà không có cải tiến, trong đề tài lần này cũng không phải ngoại lệ, sau đây sẽ là những hạn chế, thiếu sót trong dự án mà chúng ta sẽ phải bắt gặp trong quá trình trải nghiệm

### Giao diện:

Giao diện còn khá sơ xài, thiếu tính sáng tạo, chưa được bắt mắt và thu hút như những hệ thống khác, điều này có thể nói rằng là do thời gian dự án cũng như là hạn chế về năng lực của thành viên nhóm dẫn đến thiếu sót này. Tuy nhiên thì đây là một thiếu sót mà mình sẽ rằng là chúng ta có thể học được và sửa đổi trong tương lai, đây chỉ là vấn đề thời gian mà thôi, trình độ về con người, nhân sự cũng không phải là tất cả đối với vấn đề này.

### Chức năng thống kê:

Nếu các bạn để ý kỹ thì có thể thấy chức năng thống kê, như là thống kế doanh thu, lợi nhuận..v.v…Thì dự án lần này không có, để giải thích cho chuyện này thì chúng ta cần phải quay lại về vấn đề nhân sự của nhóm, có thể nói nhân sự nhóm vẫn chưa có đủ trải nghiệm cũng như kiến thức để làm được chức năng này. Tuy nhiên thì mình vẫn nghĩ là vấn đề này hoàn toàn có thể khắc phục được trong tương lai không xa.

### Phân quyền người dùng, đăng ký:

Về chức năng phần quyền người dùng và đăng ký thì thực tế ra chức năng này không khó đối với nhân sự của nhóm mình, tuy vậy nhưng thời gian làm dự án thì lại có hạn, vì vậy mà bạn có thể thấy chức năng đăng ký người dùng cũng như là phân quyền người dùng chỉ có thể ngừng lại ở mức độ là Cơ sở dữ liệu chứ chưa có giao diện hay là mã lệnh xử lý ở phía server. Điều cũng hoàn toàn dễ hiểu vì đây cũng là một chức năng mà nhóm dự định sẽ để cuối cùng của dự án, nhưng vẫn là không kịp hoàn thành được chức năng.

## Kết luận:

Trong suốt quá trình nghiên cứu và ứng dụng các kiến thức đã được học, chúng em đã tương đối hoàn thành việc phát triển website “Quản lý quán cafe”. Website này không chỉ nhằm mục đích hỗ trợ hoạt động kinh doanh mà còn là cơ hội để chúng em củng cố và mở rộng những kiến thức đã học. Chúng em hy vọng rằng, sau dự án này, chúng em sẽ có nền tảng vững chắc để tiếp tục phát triển kỹ năng và chất lượng phần mềm trong tương lai.

Qua quá trình làm việc nhóm và phát triển website, chúng em đã học được rất nhiều về quy trình phát triển phần mềm và các kỹ năng quan trọng. Tuy nhiên, do kinh nghiệm thực tế chưa nhiều, chắc chắn báo cáo và sản phẩm của chúng em không tránh khỏi những sai sót. Chúng em rất mong nhận được sự góp ý và hướng dẫn của thầy để có thể hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

KẾT LUẬN CHUNG

Nếu bỏ qua những thiếu sót nhỏ trên thì theo góc nhìn tổng quát có thể nói rằng là hệ thống đã hoàn thành và hoạt động tốt, đáp ứng tương đối là đầy đủ các yêu cầu đề ra. Dù gì thì đây cũng là một chủ đề khó đối với nhóm chúng em, dựa trên năng lực của nhóm thì có thể nói rằng là như vậy.

Và dù là chưa hoàn hảo thì cũng rất mong quý thầy cô cũng như mọi người sẽ hiểu và thông cảm cho nhóm chúng em. Một lần nữa

Chúng em xin chân thành cảm ơn thầy cô và mọi người.

TÀI LIỆU THAM KHẢO

[Github của project cuối kì.](https://github.com/ThuanPC-vn/Final_Project)

[ASP.NET MVC Documentation](https://learn.microsoft.com/en-us/aspnet/mvc/)

[Entity Framework Documentation](https://learn.microsoft.com/en-us/ef/)

[Microsoft SQL Server Documentation](https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16)