

**TRƯỜNG ĐẠI HỌC YERSIN ĐÀ LẠT  
KHOA CÔNG NGHỆ THÔNG TIN**



**YERSIN UNIVERSITY**

**BÁO CÁO MÔN HỌC  
LẬP TRÌNH WEB**

**WEB VĂN PHÒNG PHẨM**

**GVHD : Nguyễn Đức Tấn**

**SVTH : Trà Duy Bảo**

**Mã số SV : 2301010046**

**Khóa học : 2024 - 2025**

**Đà Lạt, tháng 06 - 2025**

## PHẦN NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

**Điểm:**.....

Ngày ... tháng ... năm...

*Ký và ghi rõ họ tên*

# MỤC LỤC

<b>DANH MỤC HÌNH .....</b>	<b>4</b>
<b>LỜI NÓI ĐẦU .....</b>	<b>7</b>
<b>1.1 Khái niệm .....</b>	<b>8</b>
<b>1.2 Đặc điểm .....</b>	<b>8</b>
<b>1.3 Nguyên lý hoạt động .....</b>	<b>9</b>
<b>1.4 Công nghệ triển khai ASP.NET Core .....</b>	<b>10</b>
1.4.1. Framework-Dependent Deployment (FDD): .....	10
1.4.2. Internet Information Services (IIS): .....	10
<b>2.1 Phát biểu bài toán ứng dụng .....</b>	<b>11</b>
a. Đối với Khách hàng: .....	11
b. Đối với Doanh nghiệp (Admin): .....	12
<b>2.2 Phân tích yêu cầu của ứng dụng .....</b>	<b>12</b>
<b>2.3 Thiết kế cơ sở dữ liệu .....</b>	<b>14</b>
<b>2.4 Thiết kế giao diện người dùng: .....</b>	<b>18</b>
2.4.1. Wireframe - Trang Chủ .....	18
2.4.2. Wireframe - Chi tiết sản phẩm .....	18
2.4.3. Wireframe - Sản phẩm .....	19
2.4.4. Wireframe - Giỏ hàng .....	20
2.4.6. Wireframe - Thông tin đặt hàng thành công .....	20
2.4.7. Wireframe - Liên hệ .....	21
2.4.8. Wireframe - Gửi thông tin liên hệ thành công .....	21
2.4.9. Wireframe - Đăng ký .....	22
2.4.10. Wireframe - Đăng nhập .....	22
2.4.11. Wireframe - Tài khoản đăng ký/nhập thành công .....	22
2.4.12. Wireframe - Đăng nhập tài khoản admin thành công .....	23
2.4.13. Wireframe - Quản lý sản phẩm .....	23
2.4.14. Wireframe - Thêm sản phẩm .....	23
2.4.15. Wireframe - Chỉnh sửa sản phẩm .....	24
2.4.16. Wireframe - Xóa sản phẩm .....	24
<b>2.5 Thiết kế các thành phần MVC: .....</b>	<b>25</b>
<b>2.6 Triển khai và cài đặt .....</b>	<b>30</b>
2.6.1. Môi Trường Triển Khai Thực Tế .....	30
2.6.2. Thư Viện và Framework Cần Cài Đặt .....	31
<b>Chương 3 Sơ lược về kết quả chương trình .....</b>	<b>31</b>
<b>3.1. Giao diện và Chức năng chính .....</b>	<b>31</b>
<b>3.2. Chức năng Mua hàng và Quản lý Giỏ hàng .....</b>	<b>32</b>
<b>3.3. Chức năng Xác thực và Phân quyền .....</b>	<b>34</b>
<b>3.4. Chức năng quản trị .....</b>	<b>36</b>
<b>Kết luận .....</b>	<b>36</b>
<b>Tài liệu tham khảo .....</b>	<b>37</b>

## DANH MỤC HÌNH

Hình ảnh	Mô tả	Trang
Hình 1	Sơ đồ Usecase	13
Hình 2.1	Sơ đồ CSDL	14
Hình 2.2	Sơ đồ ERD	16
Hình 2.4.1	Wireframe trang chủ	18
Hình 2.4.2	Wireframe trang chi tiết sản phẩm	18
Hình 2.4.3	Wireframe trang sản phẩm	19
Hình 2.4.4	Wireframe giỏ hàng	19
Hình 2.4.5	Wireframe thông tin giao hàng	20
Hình 2.4.6	Wireframe thông tin đặt hàng thành công	20
Hình 2.4.7	Wireframe trang liên hệ	21
Hình 2.4.8	Wireframe gửi thông tin liên hệ thành công	21
Hình 2.4.9	Wireframe trang đăng ký	22
Hình 2.4.10	Wireframe trang đăng nhập	22
Hình 2.4.11	Wireframe trang tài khoản đăng ký/nhập thành công	22
Hình 2.4.12	Wireframe đăng nhập tài khoản admin thành công	23
Hình 2.4.13	Wireframe trang quản lý sản phẩm	23
Hình 2.4.14	Wireframe trang thêm sản phẩm	23
Hình 2.4.15	Wireframe trang chỉnh sửa sản phẩm	24
Hình 2.4.16	Wireframe trang xóa sản phẩm	24
Hình 3.1	Giao diện trang chủ với các sản phẩm bán chạy	31
Hình 3.2	Giao diện danh sách sản phẩm	32
Hình 3.3	Giao diện chi tiết một sản phẩm	32
Hình 3.4	Giao diện trang giỏ hàng với danh sách sản phẩm, số lượng và tổng tiền.	33

Hình 3.5	Giao diện của form thanh toán nơi người dùng nhập thông tin giao hàng.	33
Hình 3.6	Giao diện thông báo đặt hàng thành công	34
Hình 3.6.1	Sơ đồ về cách hoạt động của trang web	34
Hình 3.7	Giao diện trang đăng ký	35
Hình 3.8	Giao diện trang đăng nhập	35
Hình 3.9	Giao diện trang Quản lý sản phẩm	36
Hình 3.10	Giao diện trang Quản lý tài khoản	36

### DANH MỤC TỪ NGỮ VIẾT TẮT

STT	Tên viết tắt	Tên đầy đủ	Dịch ra tiếng Việt (Nếu là tiếng nước ngoài)
1	CMF	Concurrent Max Flow	Lưu lượng tối đa đồng thời
2	CSDL	Cơ sở dữ liệu	
3	DBMS	Database Management System	Hệ quản trị cơ sở dữ liệu
4	MVC	Model - View - Controller	Mô hình - Giao diện - Bộ điều khiển
5	SSMS	Server Management Studio	Công cụ quản lý máy chủ
6	IIS	Internet Information Services	Dịch vụ thông tin mạng
7	HTTP	Hypertext Transfer Protocol	Giao thức truyền tải văn bản
8	FDD	Framework-Dependent Deployment	Triển khai phụ thuộc framework
9	DOM	Document Object Model	Mô hình đối tượng tài liệu
10	SSD	Solid State Drive	Phương tiện lưu trữ dữ liệu

## LỜI NÓI ĐẦU

Trong bối cảnh kỷ nguyên công nghệ thông tin đang phát triển mạnh mẽ, Internet và các ứng dụng web đã trở thành một phần thiết yếu trong đời sống xã hội và hoạt động kinh doanh. Việc xây dựng và phát triển website không chỉ đơn thuần là xu hướng mà còn là yêu cầu tất yếu để các tổ chức, doanh nghiệp tiếp cận hiệu quả với khách hàng và quảng bá sản phẩm, dịch vụ. Môn học Lập trình Web đã cung cấp cho em nền tảng kiến thức về C#, HTML, CSS và JavaScript – những công cụ chủ chốt để xây dựng giao diện người dùng cho các trang web. Nhằm củng cố kiến thức đã học và phát triển kỹ năng thực hành, em đã thực hiện đề án với chủ đề “Xây dựng Website bán Văn phòng phẩm”. Website được thiết kế với tiêu chí chính là mang lại một giao diện thân thiện, trực quan, giúp người dùng dễ dàng tìm kiếm và xem các thông tin chi tiết về sản phẩm văn phòng phẩm. Trong suốt quá trình thực hiện đề tài, em đã có cơ hội ứng dụng lý thuyết vào thực tế, từ việc xây dựng ý tưởng, thiết kế giao diện đến triển khai các chức năng cơ bản của trang web. Dù đã nỗ lực hoàn thiện sản phẩm trong thời gian và kiến thức của môn học đầu tiên về lập trình web, em hiểu rằng sản phẩm vẫn còn tồn tại những hạn chế nhất định. Vì vậy, em rất mong nhận được những góp ý quý báu từ thầy để có thể cải thiện dự án, đồng thời tích lũy thêm kinh nghiệm cho các dự án sau này.

Em xin gửi lời cảm ơn chân thành đến thầy [Nguyễn Đức Tấn] đã tận tình hướng dẫn và truyền đạt kiến thức bổ ích trong suốt học phần vừa qua.

# Chương 1 Tìm hiểu về lập trình Web MVC (Cơ sở lý thuyết)

## 1.1 Khái niệm

Mô hình MVC là một mẫu kiến trúc phần mềm được sử dụng để phát triển các ứng dụng, đặc biệt là ứng dụng web. MVC được tách biệt thành 3 thành phần chính, độc lập tương đối với nhau:

- Model :

- Đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng.
- Chịu trách nhiệm quản lý, lưu trữ, truy xuất và xử lý dữ liệu.
- Thường tương tác trực tiếp với cơ sở dữ liệu hoặc các nguồn dữ liệu khác.
- Không phụ thuộc vào View hay Controller, nó chỉ cung cấp dữ liệu và các phương thức để thao tác với dữ liệu đó.
- Trong ASP.NET Core, Model thường là các lớp C# mô tả cấu trúc dữ liệu và các quy tắc nghiệp vụ.

- View :

- Chịu trách nhiệm hiển thị dữ liệu cho người dùng và nhận tương tác từ người dùng.
- Là thành phần giao diện người dùng.
- Không chứa logic nghiệp vụ phức tạp, chỉ đơn thuần hiển thị dữ liệu được cung cấp bởi Controller thông qua Model.
- Trong ASP.NET Core, View thường được tạo bằng Razor trong các tệp .cshtml, kết hợp HTML, CSS và JavaScript để tạo ra trang web động.

- Controller :

- Đóng vai trò trung gian, điều phối sự tương tác giữa Model và View.
- Tiếp nhận yêu cầu từ người dùng .
- Xử lý yêu cầu, tương tác với Model để lấy hoặc cập nhật dữ liệu.
- Chọn một View thích hợp để hiển thị kết quả cho người dùng, và truyền dữ liệu từ Model sang View.
- Trong ASP.NET Core, Controller là các lớp C# kế thừa từ lớp Controller cơ sở, chứa các phương thức gọi là "Action methods" để xử lý các yêu cầu cụ thể.

## 1.2 Đặc điểm

Mô hình MVC có những đặc điểm nổi bật sau:

- Phân tách rõ ràng các mối quan tâm: Là đặc điểm cốt lõi. Mỗi thành phần (Model, View, Controller) có một vai trò và trách nhiệm riêng biệt, giúp mã nguồn trở nên có tổ chức, dễ quản lý hơn.



- Dễ bảo trì: Do sự phân tách, việc thay đổi một thành phần ít ảnh hưởng đến các thành phần khác. Ví dụ, thay đổi giao diện không nhất thiết phải sửa đổi logic nghiệp vụ hoặc cách xử lý yêu cầu.
- Dễ mở rộng: Khi cần thêm tính năng mới, có thể dễ dàng thêm hoặc sửa đổi các Model, View, Controller liên quan mà không làm xáo trộn toàn bộ hệ thống.
- Thân thiện với người dùng: Nhiều framework MVC hỗ trợ tạo ra các URL thân thiện, dễ dàng cho các công cụ tìm kiếm lập chỉ mục.

### 1.3 Nguyên lý hoạt động

Luồng hoạt động chung của một ứng dụng web theo mô hình MVC diễn ra như sau:

- Người dùng gửi yêu cầu: Người dùng tương tác với ứng dụng. Trình duyệt gửi một HTTP Request đến máy chủ web.
- Định tuyến: Hệ thống định tuyến của framework phân tích URL và các thông tin khác trong HTTP Request để xác định Controller nào và Action method nào trong Controller đó sẽ chịu trách nhiệm xử lý yêu cầu này.
- Controller xử lý yêu cầu: Action method được chọn trong Controller sẽ được thực thi. Controller có thể nhận dữ liệu từ request. Controller tương tác với Model để thực hiện các thao tác nghiệp vụ.
- Nếu cần đọc dữ liệu: Controller yêu cầu Model truy xuất dữ liệu từ CSDL.
- Nếu cần ghi/cập nhật dữ liệu: Controller gửi dữ liệu đến Model để xử lý và lưu trữ. Model thực hiện logic nghiệp vụ và trả kết quả về cho Controller.
- Controller chọn View và truyền dữ liệu: Dựa trên kết quả xử lý và yêu cầu, Controller chọn một View thích hợp để hiển thị thông tin cho người dùng. Controller truyền dữ liệu sang cho View đã chọn.
- View hiển thị dữ liệu: View nhận dữ liệu từ Controller. View sử dụng dữ liệu này để tạo ra nội dung HTML. View không chứa logic nghiệp vụ mà chỉ tập trung vào việc trình bày dữ liệu.
- Phản hồi về cho người dùng: Nội dung HTML được tạo bởi View sẽ được đóng gói và gửi trở lại trình duyệt của người dùng. Trình duyệt sẽ hiển thị nội dung này.

## 1.4 Công nghệ triển khai ASP.NET Core

- ASP.NET Core là một framework mã nguồn mở, đa nền tảng và hiệu suất cao do Microsoft phát triển để xây dựng các ứng dụng web và dịch vụ hiện đại. Đây là phiên bản được thiết kế lại hoàn toàn từ ASP.NET Framework truyền thống, mang lại nhiều cải tiến vượt trội.

### 1.4.1. Framework-Dependent Deployment (FDD):

Ứng dụng khi xuất bản chỉ chứa mã nguồn (StationeryShop.dll), tạo ra một gói triển khai nhỏ gọn.

Nó phụ thuộc và yêu cầu máy chủ phải được cài đặt sẵn .NET Framework tương ứng để có thể chạy.

### 1.4.2. Internet Information Services (IIS):

Là một phần mềm máy chủ web của Microsoft, có nhiệm vụ tiếp nhận các yêu cầu từ người dùng

IIS sẽ chuyển tiếp các yêu cầu này đến ứng dụng ASP.NET Core để xử lý. Nó hoạt động giúp quản lý, bảo mật cho ứng dụng mà không cần phải tự lập trình.

## Chương 2 Xây dựng ứng dụng Web

### 2.1 Phát biểu bài toán ứng dụng

#### GIỚI THIỆU: XÂY DỰNG ỨNG DỤNG WEB BÁN VĂN PHÒNG PHẨM

**2.1.1. Bối cảnh và Tính cấp thiết:** Trong bối cảnh cuộc cách mạng công nghiệp 4.0 và sự bùng nổ của thương mại điện tử, hành vi mua sắm của người tiêu dùng đã thay đổi một cách sâu sắc. Mọi người có xu hướng tìm kiếm sự tiện lợi, nhanh chóng và đa dạng khi mua sắm, và văn phòng phẩm cũng không phải là ngoại lệ. Các cửa hàng văn phòng phẩm truyền thống đang đối mặt với nhiều thách thức: chi phí mặt bằng, giới hạn về không gian trưng bày sản phẩm, phạm vi tiếp cận khách hàng bó hẹp trong một khu vực địa lý nhất định và khó khăn trong việc quản lý kho hàng, đơn hàng một cách hiệu quả. Do đó, bài toán đặt ra là xây dựng một website thương mại điện tử chuyên kinh doanh các mặt hàng văn phòng phẩm. Ứng dụng này sẽ giải quyết các vấn đề của mô hình kinh doanh truyền thống, đồng thời mang lại trải nghiệm mua sắm tiện lợi cho khách hàng.

**2.1.2. Mục tiêu của dự án:** Xây dựng một kênh bán hàng trực tuyến, hoạt động 24/7. Cung cấp một nền tảng cho khách hàng có thể dễ dàng tìm kiếm, xem thông tin và đặt mua sản phẩm. Tối ưu hóa quy trình quản lý sản phẩm, đơn hàng, và khách hàng cho người quản trị. Tiếp cận được nhiều đối tượng khách hàng hơn không bị giới hạn bởi địa lý. Nâng cao hình ảnh thương hiệu và tăng lợi thế cạnh tranh trên thị trường.

**2.1.3. Đối tượng người dùng:** Hệ thống sẽ phục vụ hai nhóm đối tượng chính, tương ứng với các tác nhân trong sơ đồ use case:

+) Khách Hàng: Bao gồm cá nhân và khách hàng tổ chức có nhu cầu mua văn phòng phẩm.

+) Admin: Là chủ cửa hàng hoặc nhân viên được ủy quyền, chịu trách nhiệm quản lý toàn bộ hoạt động của website.

#### 2.1.4. Vai trò của Website

Website bán văn phòng phẩm đóng vai trò như một cầu nối đa chức năng giữa doanh nghiệp và khách hàng.

a. Đối với Khách hàng:

- Là một kênh mua sắm tiện lợi: Khách hàng có thể truy cập, lựa chọn và mua sắm mọi lúc, mọi nơi chỉ với một thiết bị có kết nối internet, thay vì phải đến cửa hàng trực tiếp.

- Là một nguồn thông tin sản phẩm dồi dào: Cung cấp đầy đủ thông tin chi tiết về sản phẩm như hình ảnh, mô tả, giá cả, thông số kỹ thuật, đánh giá từ người dùng khác, giúp khách hàng dễ dàng so sánh và đưa ra quyết định mua hàng tốt hơn.

- Là một công cụ quản lý cá nhân: Cho phép khách hàng tạo tài khoản, quản lý thông tin cá nhân, theo dõi lịch sử đơn hàng.

b. Đối với Doanh nghiệp (Admin):

- Là một "cửa hàng ảo" hoạt động 24/7: Giúp trưng bày không giới hạn số lượng sản phẩm mà không tốn chi phí mặt bằng, tiếp cận khách hàng liên tục.

- Là một công cụ quản lý bán hàng hiệu quả: Tự động hóa các quy trình từ nhận đơn hàng, xác nhận, đến quản lý thanh toán. Chức năng Quản lý sản phẩm (CRUD) cho phép Admin dễ dàng thêm, xóa, sửa sản phẩm một cách nhanh chóng.

- Là một hệ thống quản lý tập trung: Tất cả dữ liệu về sản phẩm, đơn hàng, khách hàng đều được lưu trữ và quản lý tại một nơi duy nhất, giúp việc thống kê, báo cáo trở nên dễ dàng và chính xác.

#### **2.1.5. Ý nghĩa của việc xây dựng Website**

Việc xây dựng website mang lại nhiều giá trị và ý nghĩa to lớn.

Ý nghĩa về Kinh tế:

- Tăng doanh thu: Mở rộng thị trường và tiếp cận lượng khách hàng tiềm năng không lồ trên internet.

- Giảm chi phí vận hành: Tiết kiệm chi phí thuê mặt bằng, nhân viên bán hàng và các chi phí cố định khác.

- Ý nghĩa về Vận hành: Tự động hóa quy trình, giảm thiểu sai sót do thao tác thủ công.

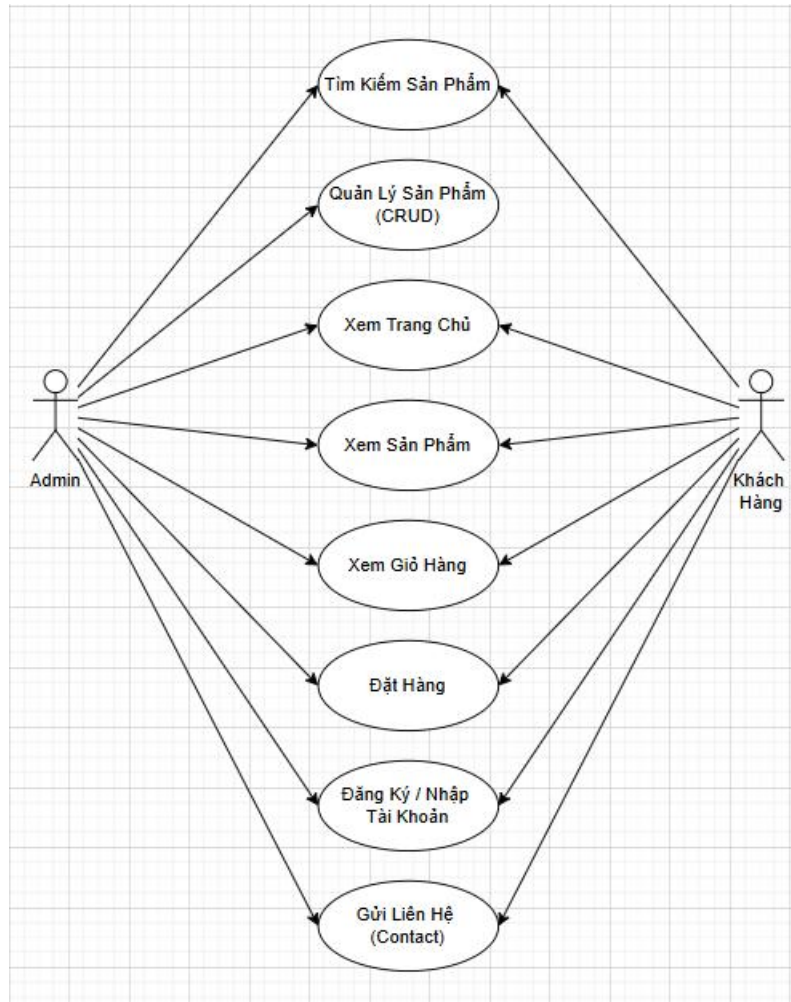
- Nâng cao hiệu quả quản lý: Admin có thể theo dõi tình hình kinh doanh mọi lúc, từ đó đưa ra các quyết định kịp thời.

- Ý nghĩa đối với Khách hàng: Tiết kiệm thời gian và công sức, mang lại trải nghiệm mua sắm thoải mái và hiệu quả.

- Tăng sự hài lòng: Khách hàng được phục vụ nhanh chóng, thông tin minh bạch và có nhiều lựa chọn hơn.

## **2.2 Phân tích yêu cầu của ứng dụng**

Sơ đồ usecase :



Hình.1. Sơ đồ Usecase

Phân tích các chức năng :

- Tìm kiếm sản phẩm : Tìm một sản phẩm cụ thể trên trang web (ví dụ tìm theo tên sản phẩm ).

- Chức năng quản lý CRUD : Cho phép quản lý các sản phẩm bao gồm tạo thêm sản phẩm cho cửa hàng, xem các sản phẩm, cập nhật thông tin sản phẩm và xóa sản phẩm.

- Xem trang chủ : Truy cập và xem nội dung trên trang chủ của website.

- Xem sản phẩm : Xem thông tin chi tiết của một hay nhiều sản phẩm.

- Xem giỏ hàng : Kiểm tra các sản phẩm đã được thêm vào giỏ hàng hay chưa.

- Đặt hàng : Thực hiện quy trình thanh toán để mua các sản phẩm trong giỏ hàng.

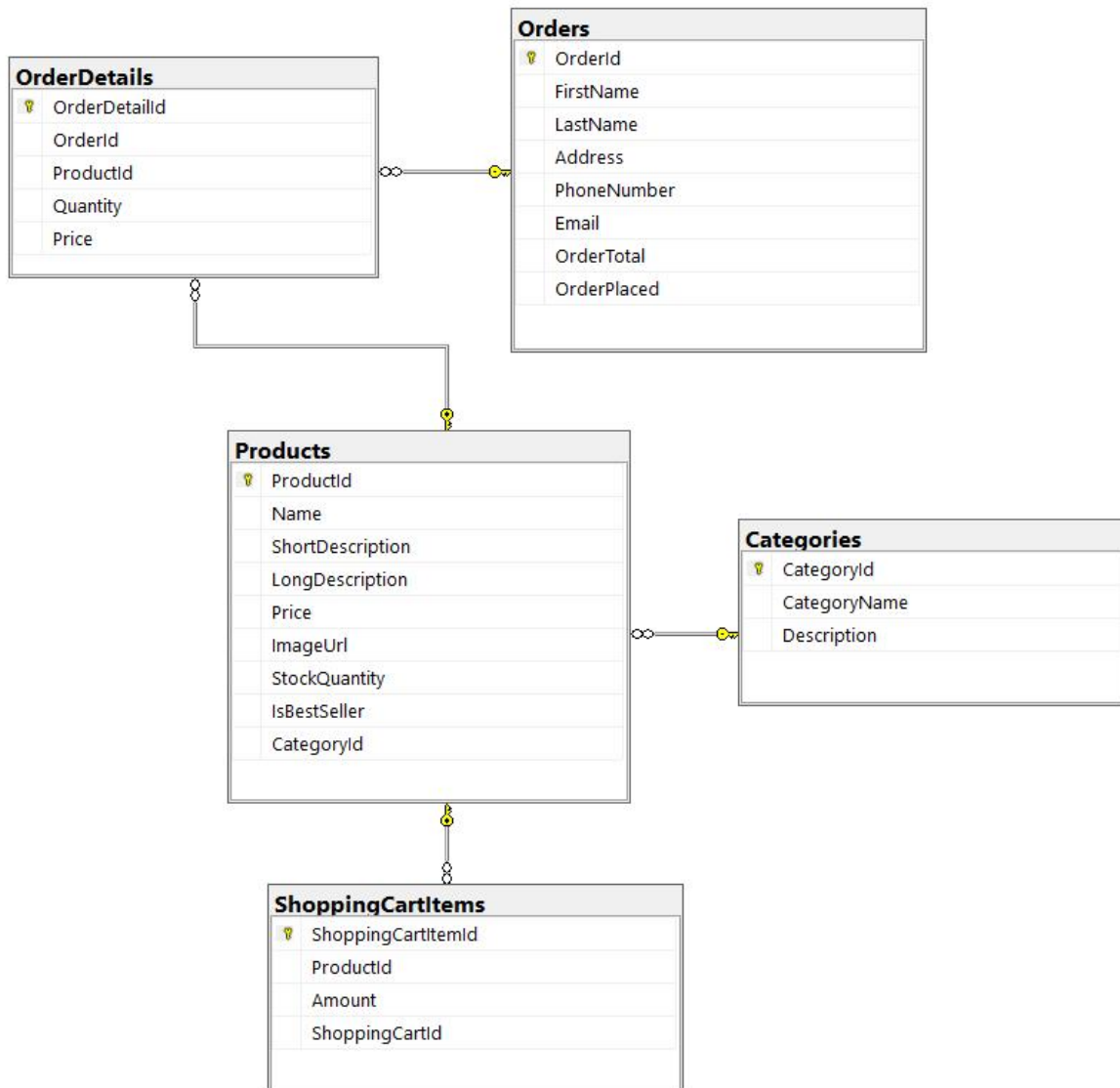
- Đăng ký/Nhập tài khoản : Tạo một tài khoản mới hoặc đăng nhập vào tài khoản đã có.

- Gửi liên hệ : Gửi tin nhắn , câu hỏi hoặc phản hồi cho quản trị viên thông qua một biểu mẫu liên hệ.

Như sơ đồ Usecase cho ta thấy tài khoản Admin có toàn quyền trong trang Web giúp Admin có thể theo dõi tình hình kinh doanh mọi lúc, cập nhật hàng hóa kịp thời còn tài khoản khách sẽ bị giới hạn quyền chức năng quản lý sản phẩm, phân chia các quyền hạn rõ ràng, minh bạch.

## 2.3 Thiết kế cơ sở dữ liệu

2.3.1. Mô hình bảng CSDL quản lý các chức năng cơ bản của một trang web thương mại :



Hình.2.1. Sơ đồ CSDL

Sơ đồ trên mô tả cách một trang web lưu trữ thông tin về Sản phẩm, phân loại chúng vào các Danh mục, cho phép người dùng thêm sản phẩm vào Giỏ hàng, và cuối cùng là đặt Đơn hàng.

Chi tiết chức năng của các bảng :

### ❖ **Categories (Danh mục sản phẩm)**

- CategoryId: Khóa chính, định danh duy nhất cho mỗi danh mục.
- CategoryName: Tên danh mục (ví dụ: “Bút”, “Giấy”, ”Sách”,...).
- Description: Mô tả chi tiết về danh mục.
- Mục đích: Dùng để phân nhóm các sản phẩm.

### ❖ **Products (Sản phẩm)**

- ProductId: Khóa chính, định danh duy nhất cho mỗi sản phẩm.
- Name, ShortDescription, LongDescription: Thông tin mô tả về sản phẩm.
- Price: Giá của sản phẩm.
- ImageUrl: Đường dẫn đến hình ảnh sản phẩm.
- StockQuantity: Số lượng sản phẩm còn trong kho.
- IsBestSeller: Cờ (boolean) để đánh dấu sản phẩm bán chạy.
- CategoryId: Khóa ngoại, liên kết đến bảng Categories. Điều này cho biết sản phẩm thuộc danh mục nào.
- Mỗi quan hệ: Một Category có thể có nhiều Products , quan hệ 1-N

### ❖ **ShoppingCartItems (Các mặt hàng trong giỏ hàng)**

- ShoppingCartItemId: Khóa chính.
- ProductId: Khóa ngoại, liên kết đến sản phẩm đã được thêm vào giỏ.
- Amount: Số lượng của sản phẩm đó trong giỏ hàng.
- ShoppingCartId: Một mã định danh cho giỏ hàng của một người dùng cụ thể. Mã này thường được lưu trong session hoặc cookie của người dùng.
- Mục đích: Lưu trữ trạng thái giỏ hàng tạm thời của người dùng trước khi họ thanh toán.

### ❖ **Orders (Đơn hàng)**

- OrderId: Khóa chính, định danh duy nhất cho mỗi đơn hàng.
- FirstName, LastName, Address, PhoneNumber, Email: Thông tin của người đặt hàng.
- OrderTotal: Tổng giá trị của đơn hàng.
- OrderPlaced: Thời gian đơn hàng được tạo.
- Mục đích: Lưu thông tin chung về một đơn hàng đã được khách hàng xác nhận.

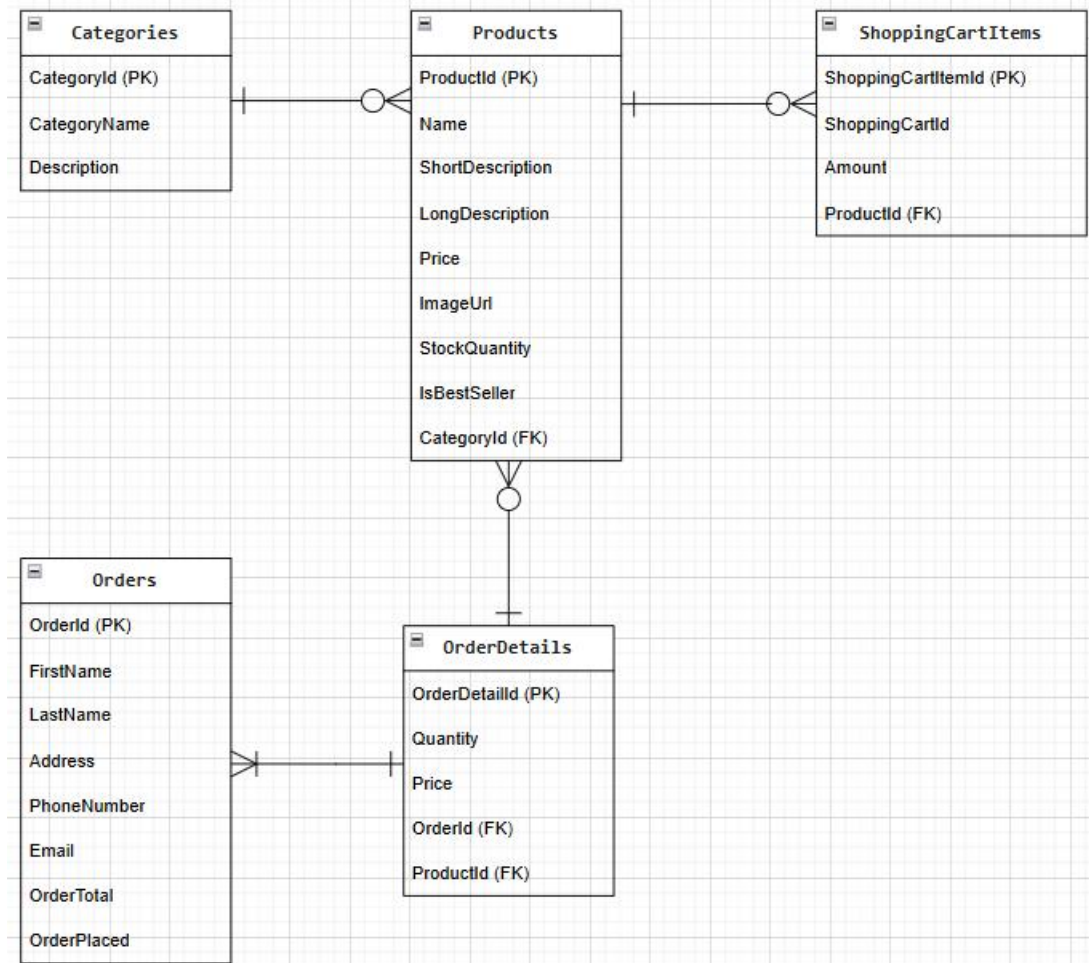
### ❖ **OrderDetails (Chi tiết đơn hàng)**

- OrderDetailId: Khóa chính.
- OrderId: Khóa ngoại, liên kết đến bảng Orders. Cho biết chi tiết này thuộc về đơn hàng nào.
- ProductId: Khóa ngoại, liên kết đến bảng Products. Cho biết sản phẩm nào đã được mua.
- Quantity: Số lượng sản phẩm đã mua.

- Price: Giá của sản phẩm tại thời điểm mua. Đây là một trường rất quan trọng, vì giá trong bảng Products có thể thay đổi sau này, nhưng giá trong đơn hàng đã đặt thì không.

→ Mục đích: Đây là một bảng trung gian, tạo mối quan hệ nhiều-nhiều giữa Orders và Products. Một đơn hàng có thể có nhiều sản phẩm, và một sản phẩm có thể xuất hiện trong nhiều đơn hàng.

### 2.3.2. Biểu đồ ERD được xây dựng dựa trên sơ đồ Database :



Hình.2.2. Sơ đồ ERD

Sơ đồ Quan hệ Thực thể (ERD) trên mô tả cấu trúc logic của cơ sở dữ liệu được thiết kế cho một hệ thống thương mại điện tử. Mục tiêu chính của mô hình này là lưu trữ, quản lý và truy xuất dữ liệu một cách hiệu quả liên quan đến các hoạt động của website, bao gồm: quản lý sản phẩm, phân loại sản phẩm, giỏ hàng của người dùng và xử lý đơn đặt hàng.

Thiết kế này tuân thủ các nguyên tắc chuẩn của cơ sở dữ liệu, giúp giảm thiểu sự trùng lặp dữ liệu, đảm bảo tính toàn vẹn và tạo ra một nền tảng vững chắc, dễ bảo trì và mở rộng trong tương lai.



### **Các mối quan hệ giữa các thực thể**

#### **- Quan hệ Categories và Products (Một-Nhiều)**

+ Một Category có thể có nhiều Products, nhưng một Product chỉ thuộc về duy nhất một Category.

+ Thực thi: Mối quan hệ này được thiết lập thông qua khóa ngoại CategoryId trong bảng Products, tham chiếu đến khóa chính của bảng Categories.

#### **- Quan hệ giữa Orders và Products (Nhiều-Nhiều)**

+ Một Order có thể bao gồm nhiều Products, và một Product có thể xuất hiện trong nhiều Orders khác nhau.

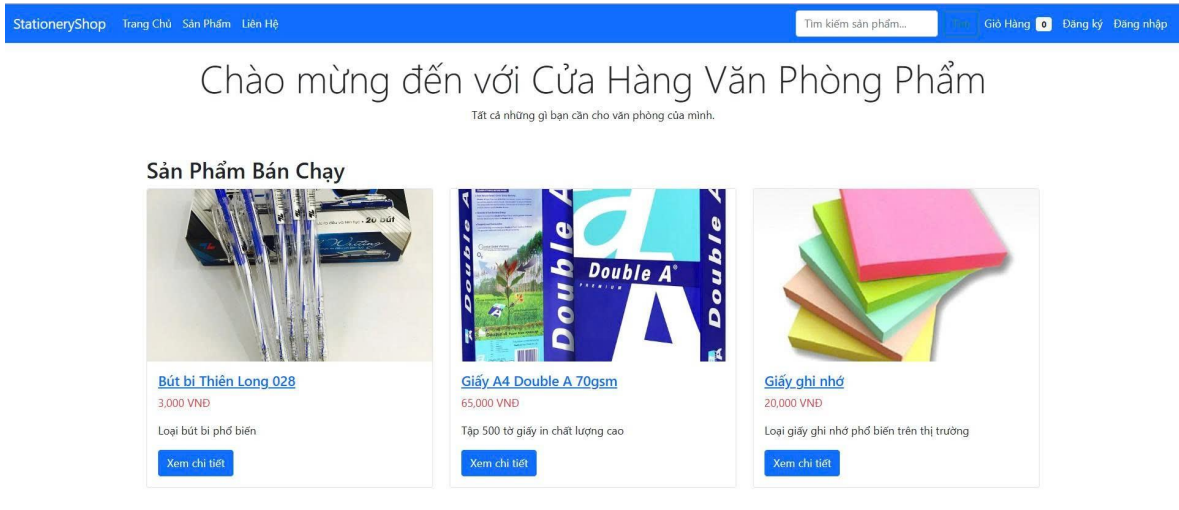
+ Thực thi: Mối quan hệ Nhiều-Nhiều này được giải quyết thành hai mối quan hệ Một-Nhiều bằng cách sử dụng bảng trung gian OrderDetails: Orders và OrderDetails (Một-Nhiều): Một đơn hàng phải có một hoặc nhiều chi tiết đơn hàng.

#### **- Products và OrderDetails (Một-Nhiều):**

+ Một sản phẩm có thể có mặt trong không hoặc nhiều chi tiết đơn hàng

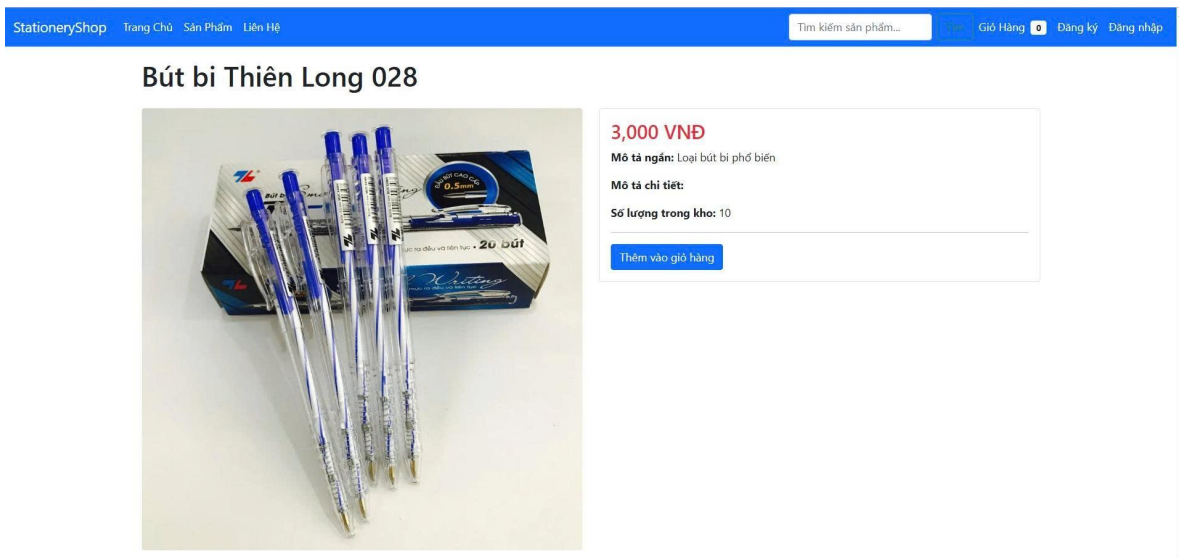
+ Thực thi: Được thiết lập chi tiết thông qua khóa ngoại là ProductId trong bảng ShoppingCartItems.

## 2.4 Thiết kế giao diện người dùng:



© 2025 - StationeryShop - Privacy

### 2.4.1. Wireframe - Trang Chủ



### 2.4.2. Wireframe - Chi tiết sản phẩm

StationeryShop

[Trang Chủ](#) [Sân Phẩm](#) [Liên Hệ](#)

Tìm kiếm sản phẩm...

Giỏ Hàng

0

Đăng ký

Đăng nhập

Giỏ Hàng Của Bạn

Số lượng	Sản phẩm	Giá	Thành tiền	
1	Giấy A4 Double A 70gsm	65,000 VNĐ	65,000 VNĐ	Xóa
1	Sổ tay Klong A5	25,000 VNĐ	25,000 VNĐ	Xóa
1	Bút bi Thiên Long 028	3,000 VNĐ	3,000 VNĐ	Xóa
Tổng cộng:			93,000 VNĐ	

Tiến hành thanh toán

© 2025 - StationeryShop - [Privacy](#)

## 19

StationeryShop

Trang ChủSản PhẩmLiên Hệ

Tìm kiếm sản phẩm...

Giỏ Hàng3

Đăng ký

Đăng nhập

Thông Tin Giao Hàng

Vui lòng điền đầy đủ thông tin để chúng tôi có thể giao hàng cho bạn.

Họ

Tên

Địa chỉ

Số điện thoại

Email

Hoàn tất đơn hàng

© 2025 - StationeryShop - [Privacy](#)

#### 2.4.5. Wireframe - Thông tin giao hàng

StationeryShop

Trang ChủSản PhẩmLiên Hệ

Tìm kiếm sản phẩm...

Giỏ Hàng0

Đăng ký

Đăng nhập

Cảm ơn bạn đã đặt hàng! Đơn hàng của bạn sẽ sớm được xử lý.

Cảm ơn bạn đã tin tưởng và mua sắm tại cửa hàng của chúng tôi.

Quay về Trang Chủ

© 2025 - StationeryShop - [Privacy](#)

Start

#### 2.4.6. Wireframe - Thông tin đặt hàng thành công

StationeryShop

Trang ChủSản PhẩmLiên Hệ

Tìm kiếm sản phẩm...

TiếpGiỏ Hàng0Đăng kýĐăng nhập

## Liên Hệ Với Chúng Tôi

Nếu bạn có bất kỳ câu hỏi nào, đừng ngần ngại gửi tin nhắn cho chúng tôi!

Tên của bạn

Email

Nội dung tin nhắn

Gửi Tin Nhắn

Địa chỉ

123 Đường ABC, Quận 1, TP. Hồ Chí Minh

Điện thoại

(028) 38 123 456

Email

contact@stationeryshop.com

© 2025 - StationeryShop - [Privacy](#)

#### 2.4.7. Wireframe - Liên hệ

StationeryShop

Trang ChủSản PhẩmLiên Hệ

Tìm kiếm sản phẩm...

TiếpGiỏ Hàng0Đăng kýĐăng nhập

## Cảm ơn bạn đã liên hệ! Chúng tôi sẽ phản hồi sớm nhất có thể.

Chúc bạn một ngày tốt lành!

Quay về Trang Chủ

© 2025 - StationeryShop - [Privacy](#)

#### 2.4.8. Wireframe - Gửi thông tin liên hệ thành công

StationeryShop

Trang ChủSản PhẩmLiên Hệ

Tìm kiếm sản phẩm...

Tài

Giỏ Hàng

Đăng ký

Đăng nhập

Đăng Ký Tài Khoản Mới

Tên tài khoản

Mật khẩu

Xác nhận mật khẩu

Đăng ký

© 2025 - StationeryShop - [Privacy](#)

#### 2.4.9. Wireframe - Đăng ký

StationeryShop

Trang ChủSản PhẩmLiên Hệ

Tìm kiếm sản phẩm...

Tài

Giỏ Hàng

Đăng ký

Đăng nhập

Đăng Nhập

Tên tài khoản

Mật khẩu

☐ Ghi nhớ đăng nhập?

Đăng nhập

© 2025 - StationeryShop - [Privacy](#)

#### 2.4.10. Wireframe - Đăng nhập

StationeryShop

Trang ChủSản PhẩmLiên Hệ

Tìm kiếm sản phẩm...

Tài

Giỏ Hàng

Xin chào trabao!

Đăng xuất

Chào mừng đến với Cửa Hàng Văn Phòng Phẩm

Tất cả những gì bạn cần cho văn phòng của mình.

#### 2.4.11. Wireframe - Tài khoản đăng ký/nhập thành công

StationeryShop

Trang ChủSản PhẩmLiên Hệ

Tìm kiếm sản phẩm...

Tài

Giỏ Hàng

Quản Trị

Xin chào admin@shop.com!

Đăng xuất

Chào mừng đến với Cửa Hàng Văn Phòng Phẩm

Tất cả những gì bạn cần cho văn phòng của mình.

#### 2.4.12. Wireframe - Đăng nhập tài khoản admin thành công

Quản Trị

Quản lý Sản phẩm

Về Trang Chủ

Thêm Sản Phẩm Mới

Tên Sản Phẩm	Giá	Số lượng tồn	Danh mục	
Bút bi Thiên Long 028	3,000	10	Bút Viết	<div>Sửa</div> <div>Xóa</div>
Giấy A4 Double A 70gsm	65,000	50	Sổ Tay & Giấy	<div>Sửa</div> <div>Xóa</div>
Sổ tay Klong A5	25,000	75	Sổ Tay & Giấy	<div>Sửa</div> <div>Xóa</div>
Kéo văn phòng Deli	15,000	80	Dụng Cụ Văn Phòng	<div>Sửa</div> <div>Xóa</div>
Băng keo trong 4.8 cm	12,000	200	Dụng Cụ Văn Phòng	<div>Sửa</div> <div>Xóa</div>
Giấy ghi nhớ	20,000	50	Sổ Tay & Giấy	<div>Sửa</div> <div>Xóa</div>

#### 2.4.13. Wireframe - Quản lý sản phẩm

Quản Trị

Quản lý Sản phẩm

Về Trang Chủ

Thêm Sản Phẩm Mới

Name

ShortDescription

Price

ImageUrl

StockQuantity

☐ IsBestSeller

CategoryId

Bút Viết

Tạo mới

Quay lại danh sách

#### 2.4.14. Wireframe - Thêm sản phẩm

Quản Trị

Quản lý Sản phẩm

Về Trang Chủ

## Chỉnh Sửa Sản Phẩm: Bút bi Thiên Long 028

Name

Bút bi Thiên Long 028

Mô tả ngắn

Loại bút bi phổ biến

Mô tả chi tiết

Giá

3000.00

Số lượng tồn kho

10

Đường dẫn ảnh

/images/products/but-bi-thien-long.jpg

Danh mục

Bút Viết

☒ Bán chạy nhất?

Lưu thay đổi

Quay lại danh sách

#### 2.4.15. Wireframe - Chỉnh sửa sản phẩm

Quản Trị

Quản lý Sản phẩm

Về Trang Chủ

## Xác Nhận Xóa

Bạn có chắc chắn muốn xóa sản phẩm này?

Giấy ghi nhớ

Giá

20,000

Danh mục

Sổ Tay & Giấy

Xóa

Quay lại danh sách

#### 2.4.16. Wireframe - Xóa sản phẩm



## 2.5 Thiết kế các thành phần MVC:

### 2.5.1. Model:

Mục đích: Model là bản thiết kế cho dữ liệu của ứng dụng. Nó định nghĩa các đối tượng kinh doanh, các thuộc tính của chúng và mối quan hệ giữa chúng. Trong Entity Framework Code-First, các lớp Model này sẽ được dùng để tự động tạo ra các bảng tương ứng trong cơ sở dữ liệu.

Ví dụ 1 ( Product.cs )

Mô tả: Đây là lớp quan trọng nhất, đại diện cho một mặt hàng văn phòng phẩm cụ thể được bán trên trang web.

```
namespace StationeryShop.Models
{
    public class Product
    {
        public int ProductId { get; set; }
        public string Name { get; set; } = string.Empty;
        public string? ShortDescription { get; set; }
        public string? LongDescription { get; set; }
        public decimal Price { get; set; }
        public string? ImageUrl { get; set; }
        public int StockQuantity { get; set; }
        public bool IsBestSeller { get; set; }
        public int CategoryId { get; set; }
        public Category? Category { get; set; }
    }
}
```

### Chức năng chính:

- ❖ Định nghĩa dữ liệu: Nó quy định tất cả các thông tin cần có của một sản phẩm, như ProductId, Name, Price, ImageUrl, và nó thuộc về Category nào.
- ❖ Tạo bảng trong Database: Entity Framework sẽ đọc lớp này và tự động tạo ra một bảng tên là Products trong cơ sở dữ liệu. Mỗi thuộc tính trong lớp sẽ trở thành một cột trong bảng đó.

Ví dụ 2 ( Category.cs )

Mô tả: Lớp này dùng để gom nhóm các sản phẩm có cùng tính chất như Bút viết, Dụng cụ cắt, Giấy các loại. Việc này giúp người dùng dễ dàng tìm kiếm và duyệt sản phẩm.

```
namespace StationeryShop.Models
{
    public class Category
    {
        public int CategoryId { get; set; }
        public string CategoryName { get; set; } = string.Empty;
        public string? Description { get; set; }
        public List<Product>? Products { get; set; }
    }
}
```

### Chức năng chính:

- ❖ Định nghĩa Danh mục: Nó xác định cấu trúc của một danh mục sản phẩm, bao gồm CategoryId, CategoryName, và Description.
- ❖ Tạo bảng trong Database: Entity Framework sẽ dựa vào lớp này để tự động tạo ra một bảng Categories trong cơ sở dữ liệu.
- ❖ Tạo mối quan hệ Một-Nhiều: Thông qua thuộc tính List<Product> Products, nó thiết lập một liên kết quan trọng, một danh mục có thể chứa nhiều sản phẩm. Điều này giúp tổ chức dữ liệu một cách logic và cho phép dễ dàng truy vấn tất cả sản phẩm thuộc về một danh mục cụ thể.

### 2.5.2. View

Mục đích: View chịu trách nhiệm hiển thị dữ liệu cho người dùng và thu thập thông tin người dùng nhập vào. Nó không chứa logic xử lý nghiệp vụ. Chúng ta sử dụng Bootstrap 5 để xây dựng giao diện responsive một cách nhanh chóng và Razor Syntax (@) để nhúng code C# vào HTML.

Ví dụ 3 ( Views/Product/List.cshtml )

Mô tả: Trang này hiển thị một lưới các sản phẩm cho người dùng xem.

```
@model IEnumerable<StationeryShop.Models.Product>

@{
    ViewData["Title"] = "Tất cả sản phẩm";
}

<h1>@ViewData["Title"]</h1>
@if (ViewData["CurrentFilter"] != null
    && !String.IsNullOrEmpty(ViewData["CurrentFilter"].ToString()))
{
    <h4>Kết quả tìm kiếm cho: "@ViewData["CurrentFilter"]"</h4>
}
else
{
    <p>Khám phá tất cả các sản phẩm tuyệt vời của chúng tôi!</p>
}

<div class="row">
    @foreach (var product in Model)
    {
        <div class="col-md-4 col-sm-6">
            <div class="card mb-4">
                
                <div class="card-body">
                    <h5 class="card-title">
                        <a asp-controller="Product" asp-action="Detail" asp-
route-id="@product.ProductId">@product.Name</a>
                    </h5>
                    <p class="card-text text-danger font-weight-
bold">@product.Price.ToString("N0") VNĐ</p>
                    <p class="card-text">@product.ShortDescription</p>
                </div>
            </div>
        </div>
    }
</div>
```

```

        </div>
    </div>
</div>
}
</div>

```

### Chức năng chính:

- ❖ Nhận dữ liệu: Dòng `@model IEnumerable<Product>` khai báo rằng View này được thiết kế để nhận một danh sách các đối tượng Product từ Controller.
- ❖ Lặp và Hiển thị: Sử dụng vòng lặp `@foreach`, nó duyệt qua từng sản phẩm trong danh sách dữ liệu nhận được.
- ❖ Trình bày dữ liệu: Với mỗi sản phẩm, để trình bày sản phẩm dưới dạng các thẻ đẹp mắt, gọn gàng và tự động sắp xếp trên lưới. Hiển thị các thông tin quan trọng của sản phẩm ra màn hình, bao gồm hình ảnh, tên, mô tả ngắn và giá.
- ❖ Tạo tương tác: Sử dụng Tag Helper như `asp-controller`, `asp-action`, nó tạo ra một link động cho mỗi tên sản phẩm, cho phép người dùng nhấp vào để di chuyển đến trang chi tiết của đúng sản phẩm đó.

Ví dụ 4 ( Views/Shared/ Layout.cshtml )

Mô tả: Là file template "master" chứa các thành phần chung như menu, footer.

Các View khác sẽ được nhúng vào bên trong nó.

```

@inject IHttpContextAccessor HttpContextAccessor
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"] - Cửa Hàng Văn Phòng Phẩm</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light
border-bottom box-shadow mb-3">
            <div class="container-fluid">
                <a class="navbar-brand" asp-area="" asp-controller="Home" asp-
action="Index">StationeryShop</a>
                <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target=".navbar-collapse" aria-
controls="navbarSupportedContent"
                    aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="navbar-collapse collapse d-sm-inline-flex justify-
content-between">
                    <ul class="navbar-nav flex-grow-1">
                        <li class="nav-item">
                            <a class="nav-link" asp-area="" asp-
controller="Home" asp-action="Index">Trang Chủ</a>
                        </li>
                        <li class="nav-item">

```

```

        <a class="nav-link" asp-area="" asp-
controller="Product" asp-action="List">Sản Phẩm</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" asp-area="" asp-
controller="Home" asp-action="Contact">Liên Hệ</a>
    </li>
</ul>
<form class="d-flex" asp-controller="Product" asp-
action="List" method="get">
    <input class="form-control me-2" type="search"
placeholder="Tìm kiếm sản phẩm..." aria-label="Search" name="searchString">
    <button class="btn btn-outline-success"
type="submit">Tìm</button>
</form>
<div class="d-flex align-items-center">
    <ul class="navbar-nav">
        <li class="nav-item">
            <a class="nav-link" asp-
controller="ShoppingCart" asp-action="Index">
                Giỏ Hàng
                <span class="badge bg-light text-dark ms-
1">
@(&HttpContextAccessor.HttpContext?.Session.GetInt32("CartItem&Count") ?? 0)
                </span>
            </a>
        </li>
    </ul>
    <partial name="_LoginPartial" />
</div>
</div>
</nav>
</header>

<div class="container">
    <main role="main" class="pb-3">
        @RenderBody()
    </main>
</div>

<footer class="footer border-top text-muted">
    <div class="container">
        © @DateTime.Now.Year - StationeryShop - <a asp-controller="Home"
asp-action="Privacy">Privacy</a>
    </div>
</footer>

<script src="~/lib/jquery/dist/jquery.min.js"></script>
<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
<script src="~/js/site.js" asp-append-version="true"></script>

    @await RenderSectionAsync("Scripts", required: false)
</body>
</html>

```

**Chức năng chính:**

- ❖ Định nghĩa Cấu trúc HTML cơ bản: Nó tạo ra bộ khung HTML và nhúng các file CSS và JavaScript dùng chung cho toàn bộ trang web.
- ❖ Tạo Header và Menu Điều Hướng:
  - Nó xây dựng một thanh menu cố định ở đầu trang.
  - Menu này chứa logo, các link điều hướng chính như Trang Chủ, Sản Phẩm, Liên Hệ, chức năng Tìm kiếm sản phẩm, và các chức năng người dùng như Giỏ hàng và Đăng nhập/Đăng ký.
- ❖ Tạo Footer: Nó tạo ra một phần chân trang cố định chứa thông tin bản quyền.
- ❖ Hiện thị Nội dung động:
  - Khi người dùng truy cập một trang cụ thể, nội dung của file List.cshtml sẽ được nhúng vào đúng vị trí này.
  - Điều này giúp tránh việc phải lặp lại code menu và footer trên mọi trang, làm cho code gọn gàng và dễ bảo trì.

### 2.5.3. Controller

Mục đích: Controller đóng vai trò trung gian. Nó nhận các yêu cầu HTTP từ trình duyệt, gọi đến các Model thông qua Repository để thực hiện các thao tác dữ liệu, sau đó chọn một View phù hợp và truyền dữ liệu cho View đó để trả về cho người dùng.

Ví dụ 5 ( ProductController.cs )

Mô tả: Controller này chịu trách nhiệm cho tất cả các yêu cầu liên quan đến sản phẩm.

```
using Microsoft.AspNetCore.Mvc;
using StationeryShop.Models;
using StationeryShop.Repositories.Interfaces;

namespace StationeryShop.Controllers
{
    public class ProductController : Controller
    {
        private readonly IProductRepository _productRepository;
        private readonly ICategoryRepository _categoryRepository;

        public ProductController(IProductRepository productRepository,
            ICategoryRepository categoryRepository)
        {
            _productRepository = productRepository;
            _categoryRepository = categoryRepository;
        }

        public IActionResult List(string? searchString)
        {
            ViewData["CurrentFilter"] = searchString;
            var products = _productRepository.AllProducts;
            if (!String.IsNullOrEmpty(searchString))
            {
                products = products.Where(p =>
```

```

        p.Name.ToLower().Contains(searchString.ToLower()));
    }
    return View(products.ToList());
}
public IActionResult Detail(int id)
{
    var product = _productRepository.GetProductById(id);
    if (product == null)
        return NotFound();

    return View(product);
}
}
}

```

Chức năng chính:

- ❖ Quản lý và điều phối các yêu cầu liên quan đến sản phẩm.
- ❖ Hiện thị Danh sách & Tìm kiếm: Lấy tất cả sản phẩm từ Repository, lọc theo từ khóa tìm kiếm, và gửi kết quả cho View List.cshtml để hiển thị.
- ❖ Hiện thị Chi tiết: Nhận một ID sản phẩm, tìm đúng sản phẩm đó trong Repository, và gửi thông tin chi tiết của nó cho View Detail.cshtml.

## 2.6 Triển khai và cài đặt

Để ứng dụng web "Cửa Hàng Văn Phòng Phẩm" có thể hoạt động trong môi trường thực tế, cần thiết nhằm đáp ứng các yêu cầu về phần cứng và phần mềm.

### 2.6.1. Môi Trường Triển Khai Thực Tế

Ứng dụng được thiết kế để triển khai trên một máy chủ chuyên dụng với các cấu hình đề xuất như:

#### 1. Cấu hình Máy chủ:

- CPU: Tối thiểu 2-Core. Cấu hình này đủ để xử lý các yêu cầu HTTP đồng thời và các tác vụ nền của ứng dụng.
- RAM: Tối thiểu 4 GB. Lượng RAM này đảm bảo cho hệ điều hành, máy chủ web, hệ quản trị cơ sở dữ liệu và chính ứng dụng ASP.NET Core có thể hoạt động ổn định.
- Ổ cứng: Tối thiểu 50 GB SSD. Sử dụng ổ SSD sẽ cải thiện đáng kể tốc độ đọc/ghi dữ liệu của cơ sở dữ liệu và tốc độ tải ứng dụng.

#### 2. Hệ Điều Hành:

- Windows Server 2019 hoặc mới hơn: Cung cấp sự tích hợp liền mạch với máy chủ web IIS và hệ quản trị cơ sở dữ liệu Microsoft SQL Server. Việc quản trị và cấu hình trên Windows Server thường trực quan và có nhiều tài liệu hướng dẫn.
- Hoặc sử dụng Linux ,ví dụ: Ubuntu: Chi phí bản quyền thấp hơn, tính ổn định cao và hiệu suất tốt.

### 3. Phần Mềm Phía Máy Chủ:

- Máy chủ web IIS: Là máy chủ web mặc định, mạnh mẽ và được tích hợp sâu vào Windows.
- Hệ quản trị Cơ sở dữ liệu : Tương thích với Entity Framework Core, cung cấp hiệu suất cao, bảo mật và các công cụ quản trị như SQL Server Management Studio.

#### 2.6.2. Thư Viện và Framework Cần Cài Đặt

Để ứng dụng có thể được thực thi, máy chủ cần được cài đặt các framework và runtime sau:

##### ❖ Các thư viện phía Client:

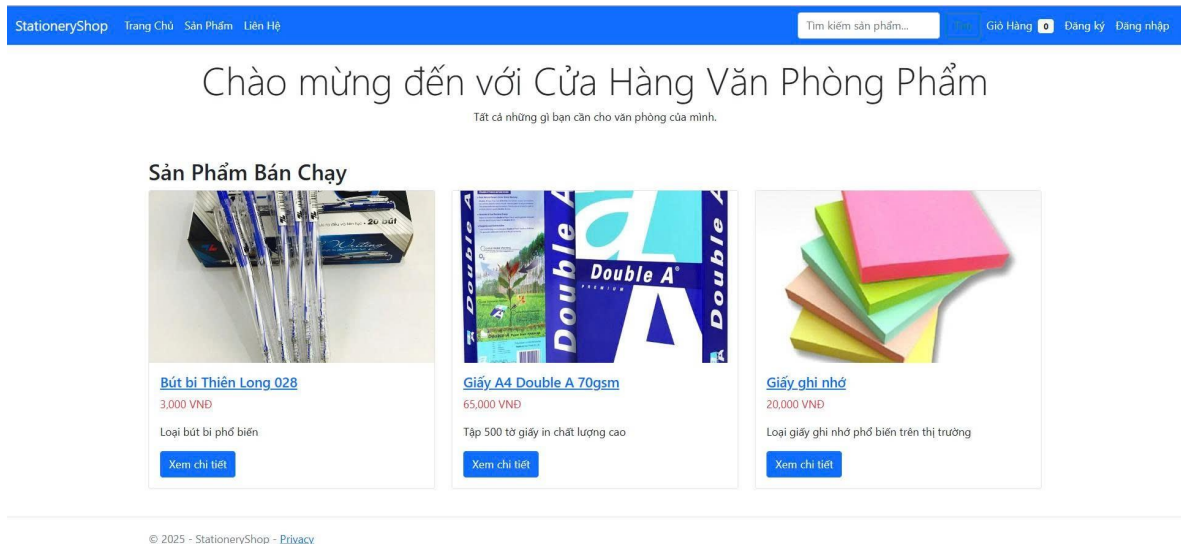
Các thư viện này đã được đóng gói sẵn trong thư mục wwwroot của ứng dụng và không cần cài đặt riêng trên máy chủ, bao gồm:

1. Bootstrap 5: Framework CSS/JS để xây dựng giao diện người dùng.
2. jQuery: Thư viện JavaScript để đơn giản hóa các thao tác DOM và xử lý sự kiện.

## Chương 3 Sơ lược về kết quả chương trình

### 3.1. Giao diện và Chức năng chính

Hình ảnh:

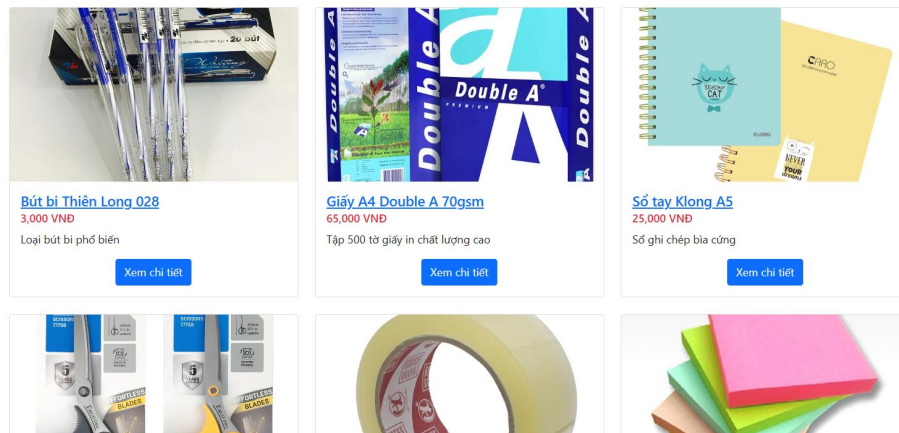


Hình 3.1: Giao diện trang chủ với các sản phẩm bán chạy.



## Danh Sách Sản Phẩm

Khám phá tất cả các sản phẩm tuyệt vời của chúng tôi!



Hình 3.2: Giao diện trang danh sách sản phẩm

### Bút bi Thiên Long 028



**3,000 VND**

Mô tả ngắn: Loại bút bi phổ biến

Mô tả chi tiết:

Số lượng trong kho: 10

[Thêm vào giỏ hàng](#)

Hình 3.3: Giao diện trang chi tiết một sản phẩm.

Nhận xét:

Ứng dụng có giao diện được thiết kế sạch sẽ, đảm bảo hiển thị tốt trên các thiết bị. Các chức năng duyệt sản phẩm, xem chi tiết và tìm kiếm hoạt động chính xác, giúp người dùng dễ dàng tương tác và tìm kiếm thông tin.

### 3.2. Chức năng Mua hàng và Quản lý Giỏ hàng

Hình ảnh:



StationeryShop

Trang ChủSản PhẩmLiên Hệ

Tìm kiếm sản phẩm...

Xóa

Giỏ Hàng3

Đăng ký

Đăng nhập

Giỏ Hàng Của Bạn

Số lượng	Sản phẩm	Giá	Thành tiền	
1	Giấy A4 Double A 70gsm	65,000 VNĐ	65,000 VNĐ	Xóa
1	Sổ tay Klong A5	25,000 VNĐ	25,000 VNĐ	Xóa
1	Bút bi Thiên Long 028	3,000 VNĐ	3,000 VNĐ	Xóa
Tổng cộng:			93,000 VNĐ	

Tiến hành thanh toán

© 2025 - StationeryShop - [Privacy](#)

Hình 3.4: Trang giỏ hàng với danh sách sản phẩm, số lượng và tổng tiền.

StationeryShop

Trang ChủSản PhẩmLiên Hệ

Tìm kiếm sản phẩm...

Xóa

Giỏ Hàng3

Đăng ký

Đăng nhập

Thông Tin Giao Hàng

Vui lòng điền đầy đủ thông tin để chúng tôi có thể giao hàng cho bạn.

Họ

Tên

Địa chỉ

Số điện thoại

Email

Hoàn tất đơn hàng

© 2025 - StationeryShop - [Privacy](#)

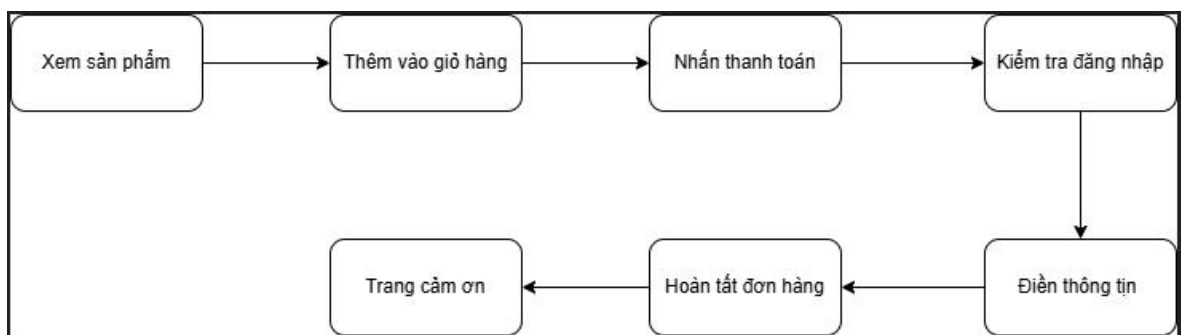
Hình 3.5: Form thanh toán nơi người dùng nhập thông tin giao hàng.



[Start]

Hình 3.6: Trang thông báo đặt hàng thành công.

Biểu đồ:



Hình 3.6.1 Sơ đồ về cách hoạt động của trang web

Nhận xét:

Xây dựng thành công luồng mua hàng hoàn chỉnh, từ việc thêm sản phẩm vào giỏ cho đến khi đặt hàng.

Hệ thống sử dụng Session để quản lý giỏ hàng hiệu quả, đảm bảo dữ liệu của mỗi người dùng là riêng biệt. Dữ liệu đơn hàng được lưu trữ thành công vào cơ sở dữ liệu.

### 3.3. Chức năng Xác thực và Phân quyền

Hình ảnh:

StationeryShop Trang Chủ Sản Phẩm Liên Hệ

Tìm kiếm sản phẩm... Trang Giỏ Hàng Đăng ký Đăng nhập

### Đăng Ký Tài Khoản Mới

Tên tài khoản

Mật khẩu

Xác nhận mật khẩu

Đăng ký

© 2025 - StationeryShop - [Privacy](#)

Hình 3.7: Giao diện trang Đăng ký

StationeryShop Trang Chủ Sản Phẩm Liên Hệ

Tìm kiếm sản phẩm... Trang Giỏ Hàng Đăng ký Đăng nhập

### Đăng Nhập

Tên tài khoản

Mật khẩu

☐ Ghi nhớ đăng nhập?

Đăng nhập

© 2025 - StationeryShop - [Privacy](#)

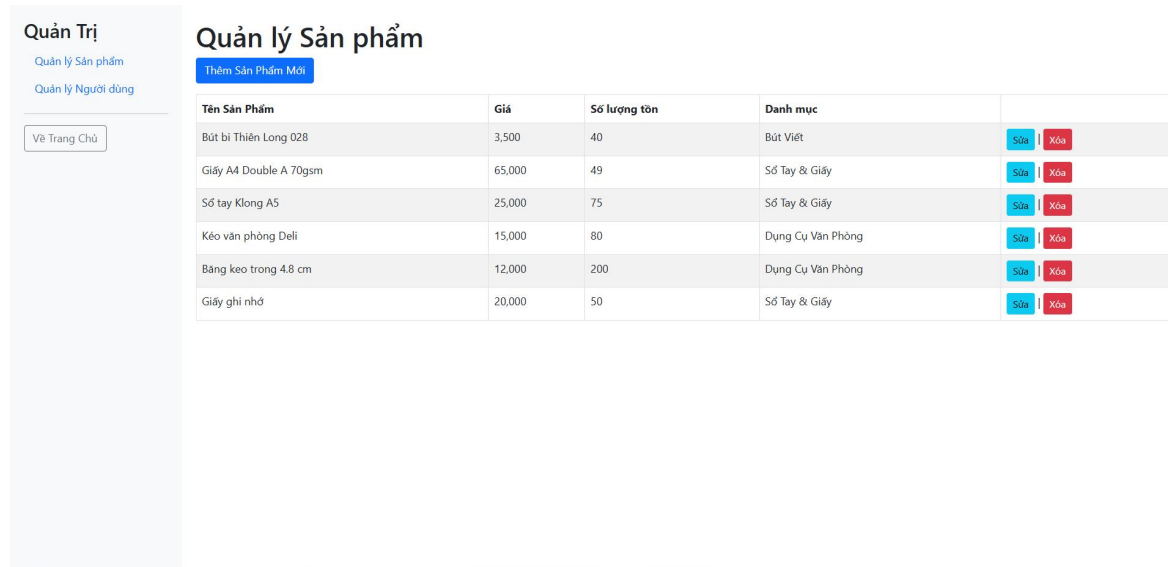
Hình 3.8: Giao diện trang Đăng nhập

Nhận xét:

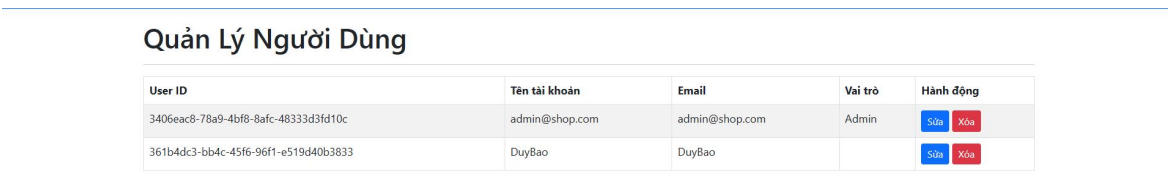
Tích hợp thành công hệ thống ASP.NET Core Identity, cho phép người dùng đăng ký, đăng nhập, đăng xuất tài khoản.

Chức năng phân quyền ([Authorize]) đảm bảo tính bảo mật và toàn vẹn dữ liệu.

### 3.4. Chức năng quản trị:



Hình 3.9: Giao diện trang Quản lý Sản phẩm



© 2025 - StationeryShop - [Privacy](#)

Hình 3.10: Giao diện trang Quản lý Tài khoản

Nhận xét:

Giao diện quản trị dễ nhìn, phân quyền hệ thống toàn diện cho admin

Cập nhật nhanh chóng trạng thái sản phẩm của cửa hàng, xem các tài khoản tồn tại trong cửa hàng

### Kết luận

❖ Tổng kết kiến thức đạt được:

### Về mặt lý thuyết:

- Qua quá trình thực hiện đề tài, em đã hiểu được cơ bản về mô hình kiến trúc MVC, phân biệt vai trò của từng thành phần.

- Hiểu sơ lược được về Entity Framework Core theo phương pháp Code-First, bao gồm cách định nghĩa các Model, thiết lập mối quan hệ giữa chúng, và sử dụng Migrations để tự động tạo và cập nhật cấu trúc cơ sở dữ liệu từ code.

- Hiểu được cơ chế hoạt động của ASP.NET Core Identity để xây dựng các chức năng xác thực và phân quyền một cách an toàn và hiệu quả.

### **Về mặt thực tiễn:**

- Đã có khả năng xây dựng một ứng dụng web ASP.NET Core MVC, thiết lập được project, kết nối cơ sở dữ liệu cho đến triển khai các chức năng phức tạp như giỏ hàng và thanh toán.

- Biết sử dụng các công cụ như Visual Studio, SQL Server Management Studio , và các lệnh trong Package Manager Console như add-migration, update-database.

❖ Điểm tồn tại:

Khuyết điểm:

### **Về giao diện người dùng:**

- Giao diện của trang web hiện tại vẫn còn khá đơn giản, chưa có nhiều hiệu ứng và tương tác phức tạp để tăng trải nghiệm người dùng, chỉ dừng lại ở mức cơ bản.

Về chức năng:

- Chức năng tìm kiếm sản phẩm mới chỉ dừng lại ở việc tìm theo tên, chưa hỗ trợ lọc nâng cao theo khoảng giá, theo danh mục hoặc sắp xếp kết quả.

- Chức năng thanh toán chưa tích hợp các cổng thanh toán trực tuyến ,mới chỉ dừng lại ở việc ghi nhận đơn hàng.

### **Về kỹ thuật:**

- Code chưa được tối ưu hóa về mặt hiệu năng ở một số truy vấn phức tạp, có thể gây chậm khi dữ liệu lớn lên.

- Ứng dụng chưa xử lý các trường hợp ngoại lệ một cách toàn diện.

Hướng mở rộng của đề tài:

- Thêm chức năng đánh giá và bình luận cho sản phẩm, chỉ những người dùng đã mua hàng mới có thể thực hiện.

- Tích hợp API và dịch vụ bên thứ ba: Tích hợp các cổng thanh toán trực tuyến phổ biến như MoMo, ZaloPay, VNPAY để đa dạng hóa phương thức thanh toán.

- Xây dựng chức năng gợi ý sản phẩm liên quan trên trang chi tiết sản phẩm.

### **Tài liệu tham khảo**

[1] Tên tài liệu: s1-s8-LTWeb.pdf – Nguyễn Đức Tấn – 2025

[2] Tên tài liệu: Youtube – TEDU Học Lập Trình Trực Tuyến – 2015

<https://www.youtube.com/@teduvn>

[3] Tên tài liệu: Google – Microsoft Docs ASP.NET – Core MVC

<https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-9.0>