

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



**YERSIN UNIVERSITY**

## **BÁO CÁO MÔN HỌC LẬP TRÌNH WEB 1**

**Tên đề tài: Xây dựng website thương mại điện tử kinh doanh sản phẩm Thú cưng + báo cáo + đồ án**

<b>GVHD</b>	<b>:Nguyễn Đức Tấn</b>
<b>SVTH</b>	<b>: Hà Gia Huy</b>
<b>Mã số SV</b>	<b>: 2301010035</b>
<b>Khóa học</b>	<b>: 2024 - 2025</b>

**Đà Lạt, tháng 6- 2025**

# MỤC LỤC

MỤC LỤC .....	2
Lời nói đầu.....	5
Chương 1 Tìm hiểu về lập trình web MVC.....	6
1.1 Giới thiệu về ASP.NET Core.....	6
1.2 Khái niệm.....	6
1.3 Đặc điểm.....	7
1.4 Nguyên lý hoạt động.....	7
1.5 Công nghệ triển khai ASP.NET Core .....	8
1.5.1 Khái niệm .....	8
1.5.2 Đặc Điểm Nổi Bật .....	8
1.5.3 Cấu trúc một ứng dụng ASP.NET core MVC .....	9
1.6 Mô hình MVC trong ASP.NET Core .....	9
1.6.1 Khái niệm MVC .....	9
1.6.2 Nguyên lý hoạt động .....	9
1.7 Ví dụ minh họa .....	10
1.7.1 Model.....	10
1.7.2 Controller.....	10
1.7.3 View.....	11
Chương 2 XÂY DỰNG ỨNG DỤNG WEB (TRIỂN KHAI THỰC TẾ).....	12
2.1 Phát biểu bài toán ứng dụng .....	12
2.1.1 Bài toán đặt ra.....	12
2.1.2 Mục tiêu của ứng dụng web .....	12
2.1.3 Vai trò giá trị mang lại.....	13
2.1.4 Ý nghĩa thực tiễn .....	13
2.2 Phân tích yêu cầu của ứng dụng .....	13
2.2.1 Yêu cầu chức năng .....	13
2.2.2 Yêu cầu phi chức năng .....	14
2.2.3 Hiệu suất:.....	14
2.2.4 Bảo mật.....	14
2.2.5 Khả dụng: .....	14
2.2.6 Mở rộng .....	14
2.2.7 Bảo trì .....	14

2.3 Biểu đồ sơ đồ Use-Case .....	15
2.3.1 Dành cho User .....	15
2.4 Thiết kế cơ sở dữ liệu.....	16
2.4.1 Mô tả chi tiết các bảng.....	16
2.4.2 Bảng dữ liệu Identity .....	19
2.5 Thiết kế thành phần MVC: .....	19
2.5.1 Model.....	19
2.5.2 View.....	22
2.5.3 Home .....	22
2.5.4 Chi tiết sản phẩm .....	22
2.5.5 Giỏ hàng .....	23
2.5.6 Danh mục thẻ loại.....	24
2.6 Controller .....	24
2.6.1 Hiện thị danh sách .....	25
2.6.2 Hiện thị chi tiết sản phẩm .....	25
2.6.3 Tìm kiếm.....	25
2.6.4 Thêm/Xóa giỏ hàng .....	26
2.7 Triển khai và cài đặt.....	27
2.7.1 Cấu hình máy phát triển .....	27
2.7.2 Môi trường phát triển.....	27
2.7.3 Cài đặt.....	28
Chương 3 KẾT QUẢ CHƯƠNG TRÌNH .....	29
3.1 User .....	29
3.1.1 Trang chủ.....	29
3.1.2 Chi tiết sản phẩm .....	30
3.1.3 Đăng nhập.....	30
3.1.4 Đăng ký.....	31
3.1.5 Giỏ hàng .....	32
3.1.6 Thanh toán .....	32
3.1.7 Liên Hệ .....	34
3.1.8 Dịch vụ .....	34
3.1.9 Đặt lịch .....	34
Kết Luận .....	35
3.1.10 Tài liệu tham khảo: .....	36



## Lời nói đầu

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, thương mại điện tử đã và đang trở thành một phần không thể thiếu trong đời sống hiện đại. Việc mua sắm trực tuyến không chỉ mang lại sự tiện lợi, tiết kiệm thời gian mà còn mở ra cơ hội tiếp cận nguồn tài nguyên đa dạng cho người dùng, đặc biệt là trong lĩnh vực giáo dục. Nhu cầu tìm kiếm và trao đổi tài liệu học tập, báo cáo, và đồ án ngày càng tăng cao, đặc biệt đối với học sinh – sinh viên và các đối tượng nghiên cứu.

Xuất phát từ thực tế đó, việc "Xây dựng website thương mại điện tử kinh doanh sản phẩm Tài liệu học tập + Báo cáo + Đồ án" nhằm xây dựng một nền tảng trực tuyến giúp người dùng dễ dàng tìm kiếm, mua bán, chia sẻ các tài liệu học tập chất lượng. Website không chỉ cung cấp hệ thống quản lý sản phẩm, giỏ hàng, thanh toán, mà còn hướng đến trải nghiệm người dùng thân thiện, dễ sử dụng và bảo mật.

Website được xây dựng theo mô hình MVC kết hợp với phương pháp Code First, cho phép thiết kế cơ sở dữ liệu trực tiếp từ các lớp mô hình trong mã nguồn, giúp dễ dàng quản lý, thay đổi và mở rộng hệ thống theo yêu cầu.

Dự án sử dụng các công nghệ như ASP.NET MVC, Entity Framework (Code First), HTML, CSS, JavaScript và SQL Server. Website cung cấp các chức năng cơ bản của một hệ thống thương mại điện tử như: đăng ký – đăng nhập, quản lý sản phẩm, giỏ hàng, thanh toán, tìm kiếm tài liệu theo từ khóa hoặc danh mục, và hệ thống phân quyền người dùng.

Báo cáo này sẽ trình bày chi tiết quy trình xây dựng hệ thống từ khâu phân tích yêu cầu, thiết kế mô hình dữ liệu bằng Code First, đến triển khai và kiểm thử. Qua đó, em mong muốn thể hiện khả năng tư duy hệ thống, lập trình hướng đối tượng và thiết kế phần mềm theo chuẩn hiện đại.

Em xin chân thành cảm ơn thầy đã tận tình hướng dẫn và hỗ trợ trong suốt quá trình thực hiện đề tài.

# Chương 1 Tìm hiểu về lập trình web MVC

## 1.1 Giới thiệu về ASP.NET Core

ASP.NET Core là một framework mã nguồn mở, đa nền tảng được phát triển bởi Microsoft, dùng để xây dựng các ứng dụng web hiện đại như web APIs, web ứng dụng, dịch vụ microservice, và các ứng dụng thời gian thực. Đây là phiên bản cải tiến và tái thiết kế từ nền tảng ASP.NET truyền thống.

ASP.NET Core cung cấp hiệu năng cao, khả năng mở rộng tốt, và hỗ trợ triển khai trên nhiều hệ điều hành như Windows, Linux, macOS. Ngoài ra, nó cũng tích hợp chặt chẽ với **Entity Framework Core**, hỗ trợ phát triển theo kiến trúc MVC (Model - View - Controller), giúp việc tổ chức mã nguồn rõ ràng và dễ bảo trì.

## 1.2 Khái niệm

Mô hình MVC (viết tắt của Model – View – Controller) là một kiến trúc phần mềm giúp phân chia ứng dụng web thành ba thành phần riêng biệt:

- **Model:** Xử lý logic nghiệp vụ và thao tác dữ liệu. Đại diện cho dữ liệu và các quy tắc liên quan đến dữ liệu, chẳng hạn như các thao tác CRUD (Create, Read, Update, Delete).
  - **View:** Giao diện người dùng (UI), hiển thị thông tin mà Model cung cấp. View không chứa logic xử lý mà chỉ đảm nhận việc trình bày dữ liệu.
  - **Controller:** Thành phần trung gian giữa View và Model. Nó tiếp nhận yêu cầu từ người dùng, xử lý logic, tương tác với Model và chọn View phù hợp để hiển thị kết quả.
- Mô hình này giúp tách biệt rõ ràng giữa dữ liệu, giao diện và điều khiển luồng xử lý, từ đó dễ dàng bảo trì, mở rộng và phát triển ứng dụng.

Ngoài ra, MVC được xem như một chuẩn thiết kế phổ biến trong các framework web hiện đại như ASP.NET Core, Laravel (PHP), Ruby on Rails (Ruby), Spring MVC (Java)... giúp rút ngắn thời gian phát triển và tăng độ ổn định cho sản phẩm phần mềm.

### **1.3 Đặc điểm**

- Phân tách rõ vai trò: Giữa giao diện người dùng (View), xử lý nghiệp vụ (Controller), và dữ liệu (Model).
- Tăng tính tái sử dụng: Các thành phần có thể được tái sử dụng độc lập.
- Dễ bảo trì và mở rộng: Nhờ việc phân tách module rõ ràng, lập trình viên có thể dễ dàng chỉnh sửa hoặc thêm tính năng mới.
- Hỗ trợ phát triển theo nhóm: Nhà thiết kế làm việc với View, lập trình viên backend làm việc với Model và Controller mà không ảnh hưởng lẫn nhau.
- Thân thiện với SEO: Vì mô hình MVC cho phép kiểm soát URL tốt hơn so với mô hình truyền thống

Nhờ đặc điểm này, MVC rất phù hợp với các dự án lớn hoặc những hệ thống cần mở rộng lâu dài, nơi mà khả năng cộng tác nhóm và bảo trì hệ thống là rất quan trọng.

### **1.4 Nguyên lý hoạt động**

- Nguyên tắc hoạt động của MVC được mô tả theo quy trình sau:
  1. Người dùng gửi yêu cầu từ trình duyệt (HTTP Request).
  2. Routing Middleware trong ASP.NET Core phân tích URL và ánh xạ đến Controller và Action phù hợp.
  3. Controller tiếp nhận yêu cầu:
    - Nếu cần xử lý dữ liệu: Gọi tới Model để thực hiện các thao tác (truy vấn, cập nhật...).
    - Sau đó, chọn một View để hiển thị dữ liệu cho người dùng.

4. Model thực hiện xử lý nghiệp vụ và truy xuất dữ liệu từ CSDL.
5. View nhận dữ liệu từ Controller và hiển thị trên giao diện người dùng.
6. Phản hồi (HTTP Response) được gửi ngược về trình duyệt.

- Luồng xử lý tổng quát:

Client Request → Routing → Controller → Model (nếu cần) → View → Client Response

Trong thực tế triển khai, mô hình MVC giúp cải thiện hiệu năng ứng dụng bằng cách giảm số lần truy cập cơ sở dữ liệu không cần thiết và tối ưu hóa việc hiển thị nội dung theo từng loại thiết bị (PC, Mobile...).

## 1.5 Công nghệ triển khai ASP.NET Core

### 1.5.1 Khái niệm

ASP.NET Core là một framework mã nguồn mở, đa nền tảng, do Microsoft phát triển, dùng để xây dựng các ứng dụng web hiện đại, dịch vụ web và API. Nó là phiên bản cải tiến và tái cấu trúc của ASP.NET truyền thống.

Hoạt động được trên Windows, Linux và macOS

Hỗ trợ xây dựng ứng dụng hiệu suất cao, dễ mở rộng và bảo trì

Cho phép triển khai theo mô hình MVC, Razor Pages, Blazor, hoặc Web API

ASP.NET Core được thiết kế lại hoàn toàn để loại bỏ các ràng buộc cũ, tối ưu hóa hiệu suất và khả năng mở rộng. Đây là công nghệ chủ lực của Microsoft hiện nay trong phát triển ứng dụng web hiện đại.

### 1.5.2 Đặc Điểm Nổi Bật

- Đa nền tảng (Cross-platform): Chạy trên nhiều hệ điều hành.
- Hiệu năng cao: Sử dụng Kestrel – web server nhẹ, tốc độ cao.
- Hỗ trợ Dependency Injection (DI) tích hợp sẵn.
- Tích hợp tốt với Entity Framework Core theo hướng Code First.
- Modular architecture: Cho phép cấu hình Middleware dễ dàng.
- Hỗ trợ tốt cho RESTful API và WebSocket.



- Bảo mật mạnh mẽ với hệ thống xác thực (Authentication) và phân quyền (Authorization)

Với kiến trúc dạng module, lập trình viên có thể kiểm soát chi tiết luồng xử lý yêu cầu từ client thông qua các middleware như UseRouting, UseAuthentication, UseAuthorization, v.v. Điều này giúp ứng dụng có tính tùy biến và tối ưu hóa cao.

### 1.5.3 Cấu trúc một ứng dụng ASP.NET core MVC

/Controllers

/Models

/Views

/wwwroot

appsettings.json

Program.cs

Startup.cs (tùy phiên bản .NET)

## 1.6 Mô hình MVC trong ASP.NET Core

### 1.6.1 Khái niệm MVC

MVC là viết tắt của Model – View – Controller, là kiến trúc thiết kế phân chia ứng dụng thành 3 thành phần:

Model: Quản lý dữ liệu và logic nghiệp vụ.

View: Giao diện hiển thị dữ liệu cho người dùng.

Controller: Xử lý yêu cầu từ người dùng, điều phối Model và View.

→ Mục tiêu chính là tách biệt giao diện, xử lý logic và dữ liệu, giúp bảo trì và phát triển dễ dàng hơn.

MVC là mô hình nổi bật trong phát triển web vì giúp hạn chế sự phụ thuộc giữa các thành phần, tăng khả năng kiểm thử và quản lý mã nguồn hiệu quả. Khi ứng dụng mở rộng, việc bảo trì hoặc chỉnh sửa chỉ ảnh hưởng đến thành phần liên quan thay vì toàn bộ hệ thống.

### 1.6.2 Nguyên lý hoạt động

- 1 Người dùng gửi yêu cầu qua URL.
- 2 Routing Middleware phân tích và chuyển yêu cầu đến Controller phù hợp.

- 3 Controller xử lý logic, có thể gọi Model.
- 4 Dữ liệu được trả về View để hiển thị.
- 5 Gửi phản hồi về cho người dùng.

Luồng:

Client → Controller → Model (nếu cần) → View → Client

Middleware định tuyến (`UseRouting`) trong ASP.NET Core đóng vai trò ánh xạ yêu cầu HTTP đến Controller và Action phù hợp. Cơ chế routing này có thể thiết lập theo Convention-based Routing hoặc Attribute Routing để xử lý chính xác các URL theo yêu cầu nghiệp vụ.

## 1.7 Ví dụ minh họa

### 1.7.1 Model

```
public class Document
{
    public int Id { get; set; }
    public string Title { get; set; }
    public string Category { get; set; }
    public DateTime UploadDate { get; set; }
}
```

### 1.7.2 Controller

```
public class DocumentController : Controller
{
    public IActionResult Index()
    {
        var documents = new List<Document>
        {
            new Document { Id = 1, Title = "Đồ án C#", Category = "Lập trình", UploadDate = DateTime.Now },
            new Document { Id = 2, Title = "Báo cáo Web", Category = "Web", UploadDate = DateTime.Now }
        }
    }
}
```

```

    };
    return View(documents);
}
}

```

### 1.7.3 View

@model IEnumerable<Document>

<h2>Danh sách tài liệu</h2>

<table>

<tr>

<th>Tiêu đề</th>

<th>Chuyên mục</th>

<th>Ngày đăng</th>

</tr>

@foreach (var item in Model)

{

<tr>

<td>@item.Title</td>

<td>@item.Category</td>

<td>@item.UploadDate.ToShortDateString()</td>

</tr>

}

</table>

## **Chương 2 XÂY DỰNG ỨNG DỤNG WEB (TRIỂN KHAI THỰC TẾ)**

### **2.1 Phát biểu bài toán ứng dụng**

#### **2.1.1 Bài toán đặt ra**

Trong thời đại công nghệ số, việc chăm sóc thú cưng ngày càng được quan tâm hơn, không chỉ là cho ăn mà còn bao gồm các hoạt động như spa, huấn luyện, tiêm ngừa, khám sức khỏe định kỳ. Tuy nhiên, người nuôi thú cưng hiện nay còn gặp nhiều khó khăn:

Sản phẩm, phụ kiện cho thú cưng bị phân tán trên nhiều nền tảng, gây khó khăn trong tìm kiếm và lựa chọn.

Các dịch vụ chăm sóc thú cưng chưa được số hóa, việc đặt lịch spa, huấn luyện thường qua mạng xã hội hoặc gọi điện thủ công.

Thiếu kênh thông tin rõ ràng và tập trung, gây nhiễu loạn thông tin khi chọn mua hàng. Các cửa hàng, người chăm sóc thú cưng chuyên nghiệp chưa có nền tảng số hóa phù hợp để tiếp cận khách hàng dễ dàng.

Bài toán đặt ra: Cần xây dựng một nền tảng web thương mại điện tử chuyên cung cấp sản phẩm và dịch vụ cho thú cưng – nơi người dùng có thể mua sắm, đặt dịch vụ và quản lý thú cưng một cách hiện đại, dễ sử dụng, an toàn và hiệu quả.

#### **2.1.2 Mục tiêu của ứng dụng web**

- Website Pet Store ra đời nhằm đáp ứng các nhu cầu nói trên, với mục tiêu:
- Cung cấp giao diện hiện đại, dễ sử dụng cho người dùng tìm kiếm, mua sắm sản phẩm cho thú cưng.
- Cho phép đặt lịch các dịch vụ như: spa, huấn luyện, tiêm phòng, tắm gội,...
- Có hệ thống đăng ký/đăng nhập, giỏ hàng, thanh toán
- Hỗ trợ người bán quản lý sản phẩm, dịch vụ và đơn hàng.
- Tạo môi trường chăm sóc thú cưng chuyên nghiệp, hiện đại.

### 2.1.3 Vai trò giá trị mang lại

- Đối với người nuôi thú cưng:
- Dễ dàng tìm kiếm sản phẩm phù hợp cho từng loại thú cưng.
- Đặt lịch chăm sóc thú cưng nhanh chóng, tiện lợi.
- Tiết kiệm thời gian, đảm bảo chất lượng thông qua đánh giá và hình ảnh minh họa.
- Đối với người cung cấp sản phẩm/dịch vụ:
- Có nền tảng chuyên biệt để quản lý hàng hóa, dịch vụ.
- Dễ dàng tiếp cận khách hàng tiềm năng.
- Góp phần nâng cao chất lượng chăm sóc thú cưng tại Việt Nam.

### 2.1.4 Ý nghĩa thực tiễn

- Dự án Pet Store giúp sinh viên:
- Hiểu rõ cách triển khai ứng dụng web thực tế từ mô hình thương mại điện tử.
- Vận dụng được các kiến thức như: lập trình hướng đối tượng (OOP), thiết kế cơ sở dữ liệu, Entity Framework Code First, HTML – CSS – JavaScript, mô hình MVC,...
- Rèn luyện tư duy hệ thống, kỹ năng quản lý sản phẩm, quản lý dữ liệu thực tế.
- Gắn gũi với mô hình làm việc tại doanh nghiệp khi triển khai hệ thống có chức năng giỏ hàng, phân quyền, quản trị backend.
- Nếu thực hiện nhóm: nâng cao khả năng teamwork, sử dụng Git, chia task, quản lý tiến độ.

## 2.2 Phân tích yêu cầu của ứng dụng

### 2.2.1 Yêu cầu chức năng

- Người dùng chưa đăng nhập:
- [FR1] Xem danh sách sản phẩm/dịch vụ trên trang chủ.
- [FR2] Tìm kiếm sản phẩm theo từ khóa hoặc danh mục.
- [FR3] Xem chi tiết từng sản phẩm.
- [FR4] Đăng ký tài khoản mới.
- [FR5] Đăng nhập vào hệ thống.
- [FR6] Người dùng đã đăng nhập:

- [FR7] Thêm sản phẩm vào giỏ hàng.
- [FR8] Đặt lịch dịch vụ cho thú cưng.hả
- [FR9] Thực hiện thanh toán.
- [FR10] Xem và tải hóa đơn đơn hàng.
- [FR12] Đăng xuất.

#### 2.2.2 Yêu cầu phi chức năng

##### 2.2.3 Hiệu suất:

- Phản hồi nhanh, tải trang < 3 giây.
- Muợt mà trên mọi thiết bị.

##### 2.2.4 Bảo mật

- Bảo vệ thông tin người dùng, giỏ hàng, thanh toán.
- Chỉ người mua mới được tải hóa đơn/dịch vụ.

##### 2.2.5 Khả dụng:

- Giao diện dễ dùng, có thông báo lỗi rõ ràng.

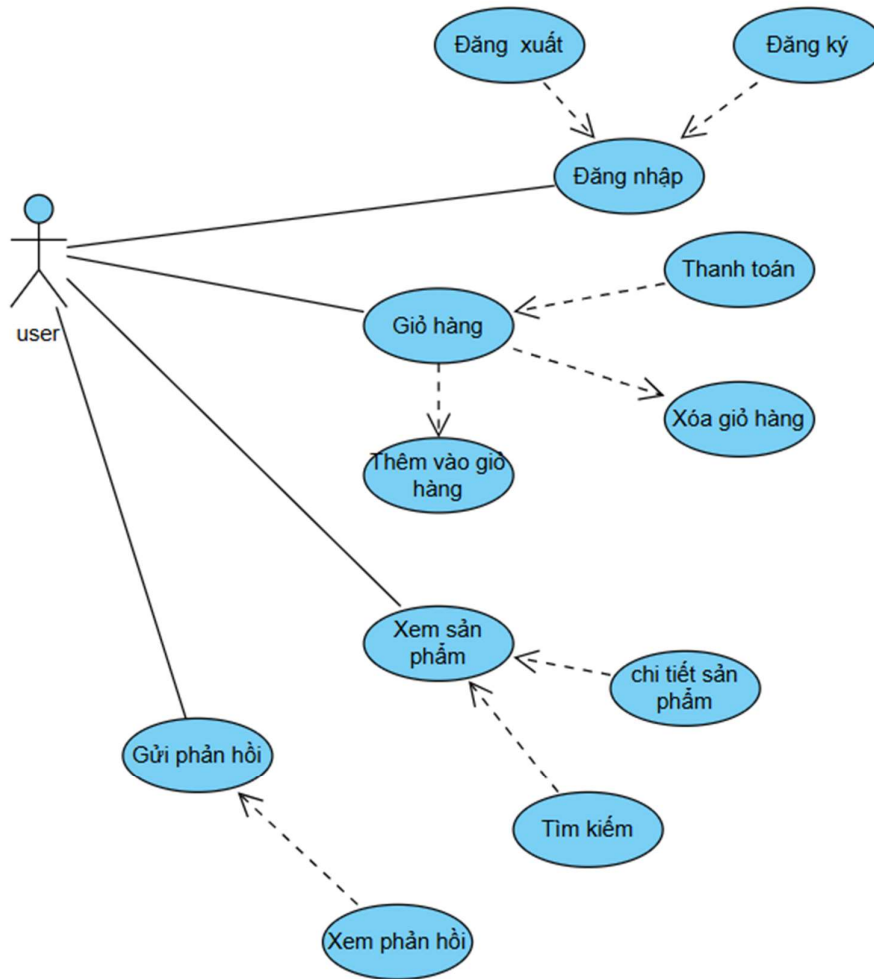
##### 2.2.6 Mở rộng

- Có thể thêm danh mục sản phẩm/dịch vụ mới.
- Hỗ trợ tích hợp thanh toán online sau này.

##### 2.2.7 Bảo trì

- Sử dụng mô hình MVC rõ ràng.
- Dễ dàng cập nhật thông qua Entity Framework Code First.

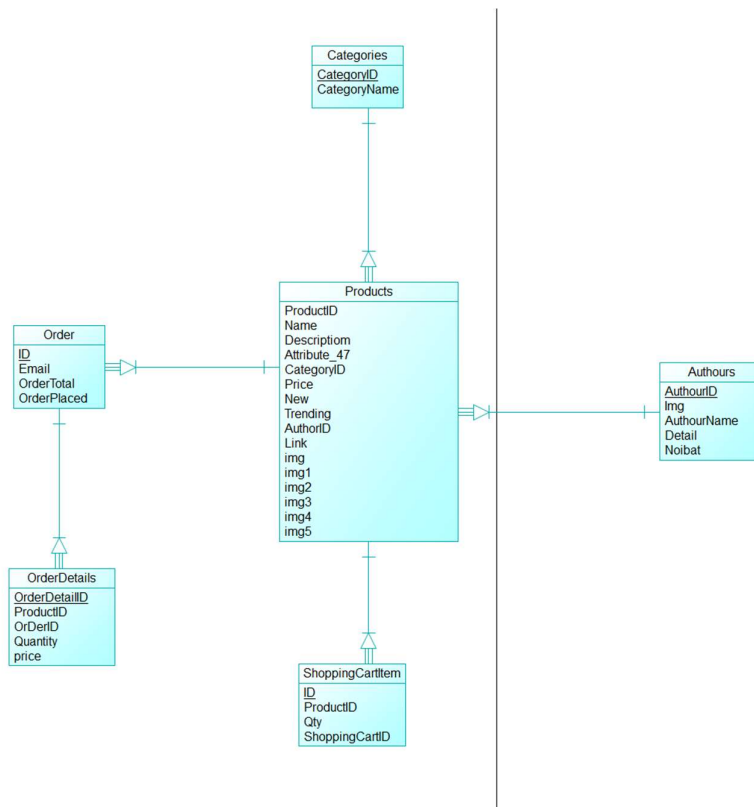
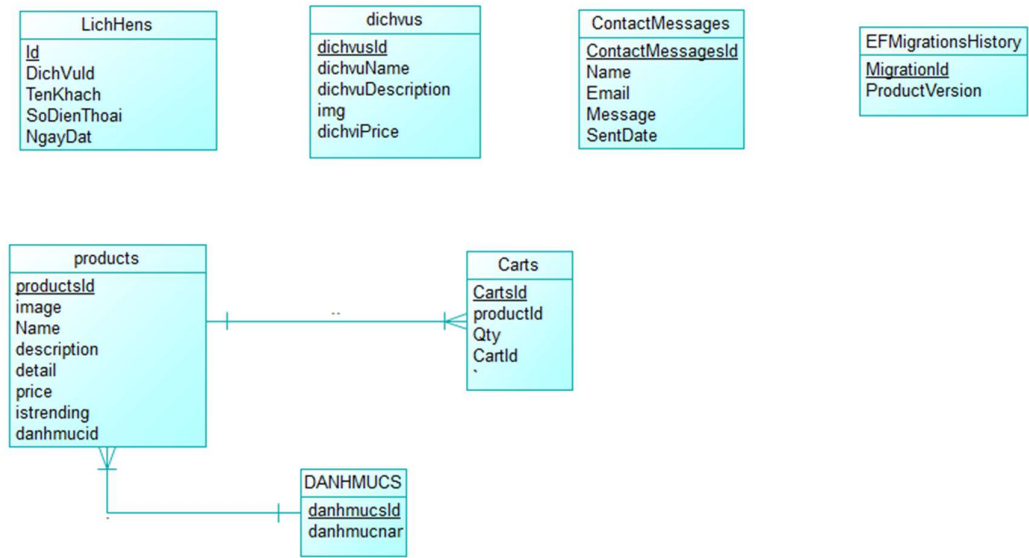
## 2.3 Biểu đồ sơ đồ Use-Case



### 2.3.1 Dành cho User

- Đăng ký / Đăng nhập.
- Xem/sửa tài khoản.
- Xem sản phẩm, thêm vào giỏ.
- Quản lý giỏ hàng.
- Đặt dịch vụ.
- Thanh toán đơn hàng.
- Xem lịch sử giao dịch.
- Gửi phản hồi/liên hệ

## 2.4 Thiết kế cơ sở dữ liệu



### 2.4.1 Mô tả chi tiết các bảng

Bảng 1: :Products



products			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	<input type="checkbox"/>
	image	nvarchar(MAX)	<input type="checkbox"/>
	name	nvarchar(MAX)	<input type="checkbox"/>
	description	nvarchar(MAX)	<input type="checkbox"/>
	detail	nvarchar(MAX)	<input type="checkbox"/>
	price	decimal(18, 2)	<input type="checkbox"/>
	istrending	bit	<input type="checkbox"/>
	danhmucid	int	<input type="checkbox"/>
	products		<input type="checkbox"/>

Bảng 2: Carts

Carts			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	<input type="checkbox"/>
	productId	int	<input checked="" type="checkbox"/>
	Qty	int	<input type="checkbox"/>
	CartId	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Bảng 3: danhmuocs

danhmuocs			
	Column Name	Data Type	Allow Nulls
🔑	danhmucid	int	<input type="checkbox"/>
	danhmucname	nvarchar(MAX)	<input type="checkbox"/>
			<input type="checkbox"/>

Bảng 4: LịchHẹn

LichHens			
	Column Name	Data Type	Allow Nulls
▶	Id	int	<input type="checkbox"/>
	DichVuId	int	<input type="checkbox"/>
	TenKhach	nvarchar(MAX)	<input type="checkbox"/>
	SoDienThoai	nvarchar(MAX)	<input type="checkbox"/>
	NgayDat	datetime2(7)	<input type="checkbox"/>
			<input type="checkbox"/>

Bảng 5: ContactMessages

ContactMessages			
	Column Name	Data Type	Allow Nulls
▶	Id	int	<input type="checkbox"/>
	Name	nvarchar(100)	<input type="checkbox"/>
	Email	nvarchar(100)	<input type="checkbox"/>
	Message	nvarchar(1000)	<input type="checkbox"/>
	SentDate	datetime2(7)	<input type="checkbox"/>
			<input type="checkbox"/>

Bảng 6: dichvus

dichvus			
	Column Name	Data Type	Allow Nulls
▶	Id	int	<input type="checkbox"/>
	dichvuName	nvarchar(MAX)	<input type="checkbox"/>
	dichvuDescription	nvarchar(MAX)	<input type="checkbox"/>
	img	nvarchar(MAX)	<input type="checkbox"/>
	dichviPrice	decimal(18, 2)	<input type="checkbox"/>
			<input type="checkbox"/>

### 2.4.2 Bảng dữ liệu Identity

Khác với các bảng nghiệp vụ do mình tự tạo, các bảng trong hệ thống Identity được tạo tự động bằng thư viện ASP.NET Core Identity. Thư viện này hỗ trợ sẵn các chức năng như đăng nhập, đăng ký và phân quyền người dùng. Khi chạy lệnh migration, hệ thống sẽ tạo ra các bảng sau:

AspNetUsers: chứa thông tin người dùng như tên đăng nhập, email, mật khẩu đã mã hóa... Trong dự án này, bảng được mở rộng thêm các cột như FullName (họ tên) và Balance (số dư) để phục vụ cho nhu cầu nghiệp vụ.

AspNetRoles: lưu các vai trò của người dùng, ví dụ như Admin, Khách hàng, v.v.

AspNetUserRoles: bảng trung gian ánh xạ mối quan hệ nhiều-nhiều giữa người dùng và vai trò.

Ngoài ra còn có các bảng hỗ trợ khác như:

- AspNetUserClaims,
- AspNetUserLogins,
- AspNetUserTokens

## 2.5 Thiết kế thành phần MVC:

Tầng Model đóng vai trò quan trọng trong việc quản lý dữ liệu của ứng dụng. Nó định nghĩa cấu trúc dữ liệu và chứa các quy tắc nghiệp vụ liên quan đến dữ liệu đó. Model là nơi trực tiếp làm việc với cơ sở dữ liệu, bao gồm các thao tác như lấy dữ liệu, lưu trữ, cập nhật và xóa.

Trong ứng dụng, tầng Model được xây dựng bằng các lớp C# thuần túy (POCO – Plain Old CLR Objects). Những lớp này được kết hợp với Entity Framework Core, giúp ánh xạ các đối tượng trong code với các bảng trong cơ sở dữ liệu một cách tự động và hiệu quả.

### 2.5.1 Model

Bảng 1: Cart

Tên cột	Kiểu Dữ liệu	Mô tả
---------	--------------	-------

Id	Int	Khóa chính, định danh duy nhất cho mỗi dòng trong giỏ hàng.
product	Product (object)	Thông tin sản phẩm được thêm vào giỏ hàng. Đây là quan hệ đến bảng Product.
Qty	Int	Số lượng sản phẩm trong giỏ hàng.
CartId	String (nullable)	Mã định danh giỏ hàng (ví dụ: theo người dùng, session,...).

Bảng 2: Contact

Tên cột	Kiểu Dữ liệu	Mô tả
Id	Int	Khóa chính, định danh duy nhất cho mỗi liên hệ.
Name	String (tối đa 100 ký tự)	Tên người gửi. Bắt buộc nhập.
Email	String (tối đa 100 ký tự)	Địa chỉ email người gửi. Bắt buộc và đúng định dạng email.
Message	String (tối đa 1000 ký tự)	Nội dung tin nhắn liên hệ. Bắt buộc nhập.
SentDate	DateTime	Ngày gửi tin nhắn. Mặc định là thời điểm hiện tại.

Bảng 3: danhmuc

Tên cột	Kiểu Dữ liệu	Mô tả
danhmucid	Int	Khóa chính, định danh duy nhất cho mỗi danh mục.
danhmucname	String	Tên danh mục sản phẩm.
products	List<product>	Danh sách sản phẩm thuộc danh mục này (quan hệ 1-n).

Bảng 4: dichvu

Tên cột	Kiểu Dữ liệu	Mô tả
Id	Int	Khóa chính, định danh duy nhất cho mỗi dịch vụ.
dichvuName	String	Tên của dịch vụ.

dichvuDescription	String	Mô tả chi tiết về dịch vụ.
img	String	Đường dẫn hoặc tên file hình ảnh đại diện cho dịch vụ.
dichvuPrice	Decimal	Giá của dịch vụ.

Bảng 5 LichHen

Tên cột	Kiểu Dữ liệu	Mô tả
Id	Int	Khóa chính, định danh duy nhất cho mỗi lịch hẹn.
DichVuId	Int	Mã dịch vụ được khách đặt (liên kết với bảng Dịch vụ).
TenKhach	String	Họ tên khách hàng đặt lịch.
SoDienThoai	String	Số điện thoại liên hệ của khách hàng.
NgayDat	DateTime	Ngày giờ khách đặt lịch hẹn.

Bảng 6: product

Tên cột	Kiểu Dữ liệu	Mô tả
Id	Int	Khóa chính, định danh duy nhất cho mỗi sản phẩm.
image	String	Đường dẫn hình ảnh đại diện cho sản phẩm.
name	String	Tên sản phẩm.
description	String	Mô tả ngắn gọn về sản phẩm.
detail	String	Chi tiết đầy đủ về sản phẩm.
price	Decimal	Giá bán của sản phẩm.
istrending	Bool	Sản phẩm đang thịnh hành (true/false).
danhmucid	Int	Khóa ngoại liên kết với danh mục chứa sản phẩm.
danhmucs	Danhmuc (object)	Đối tượng danh mục liên kết (quan hệ 1-n).

### 2.5.2 View

View là phần giao diện hiển thị dữ liệu từ model và cho phép người dùng tương tác với hệ thống. Trong dự án này các View được mình xây dựng bằng công nghệ HTML/CSS kết hợp với Razor và các framework giao diện như Bootstrap để đảm bảo tính thẩm mỹ, phản hồi nhanh và thân thiện với người dùng.

### 2.5.3 Home

Giao diện danh sách sản phẩm được xây dựng bằng cách sử dụng vòng lặp `@foreach` trong Razor để duyệt qua danh sách sản phẩm truyền từ Model. Mỗi sản phẩm sẽ được hiển thị dưới dạng một thẻ riêng (product card), có giao diện đẹp, hiện đại nhờ sử dụng các lớp CSS của Tailwind.

```
@foreach(var item in Model)
{
    <div class="product-card bg-white rounded-lg shadow p-4">
        <a asp-controller="Home" asp-action="Detail" asp-route-id="@item.Id"></a>
        <h3 class="text-lg font-semibold">@item.name</h3>
        <p class="text-gray-600">@item.price.ToString("N0")</p>
        <form asp-controller="Cart" asp-action="Add" method="post">
            <input type="hidden" name="id" value="@item.Id" />
            <input type="hidden" name="quantity" value="1" />
            <button type="submit" class="bg-blue-600 text-white px-6 py-3 rounded hover:bg-blue-700">Thêm vào giỏ</button>
        </form>
    </div>
}
</div>
</section>
```

### 2.5.4 Chi tiết sản phẩm

Giao diện trang chi tiết sản phẩm được xây dựng nhằm cung cấp đầy đủ thông tin cho người dùng trước khi quyết định mua hàng. Dữ liệu được truyền từ controller xuống dưới dạng một đối tượng duy nhất (`@model petshop.Models.product`), chứa các thuộc tính như: tên, ảnh, giá, mô tả và chi tiết sản phẩm.

```
@model petshop.Models.product
<section class="container mx-auto px-4 py-12">
    <div class="grid grid-cols-1 md:grid-cols-2 gap-8">
        <div>
            
        </div>
        <div>
            <h2 class="text-3xl font-bold mb-4">@Model.name</h2>
            <p class="text-gray-600 text-2xl mb-4">@Model.price.ToString("N0")</p>
            <p class="text-gray-700 mb-6">@Model.description</p>
            <div class="flex items-center mb-6">
                <label for="quantity" class="mr-4">Số lượng:</label>
                <input type="number" id="quantity" min="1" value="1" class="border rounded w-16 px-2 py-1">
            </div>
            <a asp-controller="Cart" asp-action="Add" asp-route-id="@Model.Id">
                <button class="bg-blue-600 text-white px-6 py-3 rounded hover:bg-blue-700">Thêm vào giỏ</button>
            </a>
        </div>
    </div>
    <!-- Additional Info -->
    <div class="mt-12">
        <h3 class="text-2xl font-semibold mb-4">Thông tin chi tiết</h3>
        <p class="text-gray-600">@Model.detail</p>
    </div>
</section>
```

### 2.5.5 Giỏ hàng

Trang giỏ hàng được thiết kế để hiển thị danh sách sản phẩm mà người dùng đã thêm vào giỏ, đồng thời cho phép thao tác tăng/giảm số lượng hoặc xóa từng sản phẩm. Dữ liệu được truyền từ controller về dưới dạng Model (danh sách các dòng giỏ hàng) và ViewBag.Total để hiển thị tổng tiền.

```
@if (Model != null && Model.Count > 0)
{
    foreach (var item in Model)
    {
        <div class="cart-item flex items-center border-b py-4">
            
            <div class="flex-1">
                <h3 class="text-lg font-semibold">@item.product.name</h3>
                <p class="text-gray-600">@item.product.price.ToString("N0") VND x @item.Qty</p>
            </div>
            <div class="flex items-center space-x-2">
                <form method="post" asp-action="Decrease" asp-controller="Cart">
                    <input type="hidden" name="id" value="@item.product.Id" />
                    <button class="decrease-qty px-2 py-1 bg-gray-200 rounded"></button>
                </form>
                <input type="number" value="@item.Qty" min="1" class="w-12 text-center border rounded" readonly />
                <form method="post" asp-action="Increase" asp-controller="Cart">
                    <input type="hidden" name="id" value="@item.product.Id" />
                    <button class="increase-qty px-2 py-1 bg-gray-200 rounded"></button>
                </form>
                <a asp-action="Remove" asp-controller="Cart" asp-route-id="@item.product.Id" class="remove-item text-red-600">Xóa</a>
            </div>
        </div>

        <div class="flex justify-between items-center bg-white rounded-lg shadow p-6 mt-6">
            <h3 class="text-xl font-semibold">Tổng cộng: <span>@ViewBag.Total</span></h3>
            <a asp-action="Checkout" class="bg-blue-600 text-white px-6 py-3 rounded hover:bg-blue-700">Thanh toán</a>
        </div>

        <div class="mt-4 text-right">
            <a asp-action="Clear" class="text-red-600 hover:underline">Xóa toàn bộ giỏ hàng</a>
        </div>
    }
    else
    {
        <p class="text-center text-gray-600">Không có sản phẩm nào trong giỏ hàng.</p>
    }
}
</section>
```

### 2.5.6 Danh mục thể loại

```
<!-- Bộ lọc -->
<form method="get" class="flex justify-between mb-8">
  <div class="flex space-x-4">
    <!-- Danh mục -->
    <select name="categoryId" class="border rounded px-4 py-2" onchange="this.form.submit()">
      <option value="">Tất cả danh mục</option>
      @foreach (var dm in Model.Danhmucs)
      {
        <option value="@dm.danhmucid" selected="@((Model.SelectedCategoryId == dm.danhmucid))">
          @dm.danhmucname
        </option>
      }
    </select>

    <!-- Giá -->
    <select name="sortOrder" class="border rounded px-4 py-2" onchange="this.form.submit()">
      <option value="">Sắp xếp</option>
      <option value="price_asc" selected="@((Model.SortOrder == "price_asc"))">Giá: thấp đến cao</option>
      <option value="price_desc" selected="@((Model.SortOrder == "price_desc"))">Giá: cao đến thấp</option>
    </select>
  </div>

  <!-- Tìm kiếm -->
  <input type="text" name="searchTerm" value="@Model.SearchTerm" placeholder="Tìm kiếm sản phẩm..."
    class="border rounded px-4 py-2 w-64" />
  <button type="submit" class="ml-2 px-4 py-2 bg-blue-600 text-white rounded hover:bg-blue-700">Tìm</button>
</form>

<!-- Danh sách sản phẩm -->
<div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-6">
  @foreach (var item in Model.Products)
  {
    <div class="bg-white shadow rounded p-4">
      <a asp-controller="Home" asp-action="Detail" asp-route-id="@item.Id">
        
      </a>

      <h3 class="text-lg font-semibold">@item.name</h3>
      <p class="text-gray-600">@item.price.ToString("N0") VNĐ</p>
      <a asp-controller="Cart" asp-action="Add" asp-route-id="@item.Id">
        <button class="mt-4 w-full bg-blue-600 text-white py-2 rounded hover:bg-blue-700">
          Thêm vào giỏ
        </button>
      </a>
    </div>
  }
</div>
</section>
```

Đây là giao diện danh sách sản phẩm trong ứng dụng web được xây dựng bằng ASP.NET Core MVC với Razor View. Giao diện này cho phép người dùng thực hiện các chức năng tìm kiếm, lọc và xem thông tin sản phẩm một cách trực quan.

## 2.6 Controller

Controller đóng vai trò là thành phần trung gian giữa Model và View. Nó chịu trách nhiệm tiếp nhận các yêu cầu (request) từ phía người dùng, thực hiện xử lý logic cần thiết bằng cách tương tác với dữ liệu trong Model, sau đó trả về kết quả cho người dùng thông qua View.

Controller đảm nhận việc quản lý luồng điều khiển của ứng dụng và xác định các hành động (action) mà người dùng có thể thực hiện. Trong một hệ thống, mỗi Controller thường được thiết kế để phục vụ một nhóm chức năng cụ thể, chẳng hạn như: quản lý người dùng, sản phẩm, bài viết hoặc đơn hàng.



### 2.6.1 Hiện thị danh sách

```
0 references
public IActionResult Index()
{
    return View(productRepository.Gettrending());
}

2 references
public IEnumerable<product> Gettrending()
{
    return db.products.Where(p => p.istrending);
}
```

Giao diện này nằm trên trang chủ (Index.cshtml), được thiết kế để hiển thị các sản phẩm đang nổi bật (trending) trong cửa hàng. Các sản phẩm này được truy vấn từ cơ sở dữ liệu thông qua phương thức Gettrending() và hiển thị bằng vòng lặp @foreach.

### 2.6.2 Hiện thị chi tiết sản phẩm

Chức năng này cho phép người dùng xem thông tin chi tiết của một sản phẩm khi họ bấm vào tên hoặc ảnh sản phẩm từ danh sách.

```
2 references
public product? GetProductDetail(int id)
{
    return db.products.FirstOrDefault(p => p.Id == id);
}

0 references
public IActionResult Detail(int id)
{
    var product = productRepository.GetProductDetail(id);
    if (product != null)
    {
        return View(product);
    }
    return NotFound();
}
```

### 2.6.3 Tìm kiếm

Đoạn mã thuộc về action Index() trong controller xử lý việc hiển thị danh sách sản phẩm. Đây là một phiên bản nâng cao của trang danh sách sản phẩm, hỗ trợ:

- Tìm kiếm theo tên sản phẩm
- Lọc theo danh mục
- Sắp xếp theo giá tăng/giảm

```

public async Task<IActionResult> Index(
    string sortOrder = "",
    string searchTerm = "",
    int? categoryId = null)
{
    var query = _context.products.Include(p => p.danhmucs).AsQueryable();

    if (!string.IsNullOrEmpty(searchTerm))
        query = query.Where(p => p.name.Contains(searchTerm));

    if (categoryId.HasValue)
        query = query.Where(p => p.danhmucid == categoryId.Value);

    query = sortOrder switch
    {
        "price_asc" => query.OrderBy(p => p.price),
        "price_desc" => query.OrderByDescending(p => p.price),
        _ => query.OrderBy(p => p.name),
    };

    var products = await query.ToListAsync();

    var viewModel = new ProductListViewModel
    {
        Products = products,
        Danhmucs = await _context.danhmucs.ToListAsync(),
        SelectedCategoryId = categoryId,
        SearchTerm = searchTerm,
        SortOrder = sortOrder,

        // Bỏ PagingInfo
        // PagingInfo = null
    };

    return View(viewModel);
}

```

#### 2.6.4 Thêm/Xóa giỏ hàng

Controller CartController có nhiệm vụ quản lý toàn bộ hoạt động liên quan đến giỏ hàng của người dùng, bao gồm: xem giỏ, thêm sản phẩm, tăng/giảm số lượng và xóa sản phẩm.

```

0 references
public IActionResult Index()
{
    var cartItems = _cartRepository.GetAllCartItems();
    var cartTotal = _cartRepository.GetCartTotal();

    ViewBag.Total = cartTotal.ToString("N0") + " VNĐ"; // định dạng tiền

    UpdateCartCount(); // Cập nhật số lượng sản phẩm trong session

    return View(cartItems);
}

// Thêm sản phẩm vào giỏ - chỉ yêu cầu login nếu chưa có
0 references
public IActionResult Add(int id, string returnUrl = null)
{
    if (!User.Identity.IsAuthenticated)
    {
        var redirectUrl = Url.Action("Add", "Cart", new { id = id, returnUrl = returnUrl ?? Url.Action("Index", "Cart") });
        return RedirectToPage("/Account/Login", new { area = "Identity", returnUrl = redirectUrl });
    }

    var product = _cartRepository.GetProductById(id);
    if (product == null)
        return NotFound();

    _cartRepository.AddCart(product);
    UpdateCartCount();

    if (!string.IsNullOrEmpty(returnUrl))
        return LocalRedirect(returnUrl);

    return RedirectToAction("Index");
}

// Tăng số lượng sản phẩm
[HttpPost]
0 references
public async Task<IActionResult> Increase(int id)
{
    await _cartRepository.IncreaseQuantity(id.ToString());
    return RedirectToAction("Index");
}

// Giảm số lượng sản phẩm
[HttpPost]
0 references
public async Task<IActionResult> Decrease(int id)
{
    await _cartRepository.DecreaseQuantity(id.ToString());
    return RedirectToAction("Index");
}

// Xóa sản phẩm khỏi giỏ
0 references
public IActionResult Remove(int id)
{
    var product = _cartRepository.GetProductById(id);
    if (product != null)
    {
        _cartRepository.RemoveCart(product);
    }

    UpdateCartCount();
    return RedirectToAction("Index");
}

```

## 2.7 Triển khai và cài đặt

### 2.7.1 Cấu hình máy phát triển

Hệ điều hành: Windows 11 64-bit

- CPU: Intel Core i5-11400H
- RAM: 16 GB DDR4
- GPU: GTX 1650 GDDR6
- IDE: Visual Studio 2022 Community
- Trình duyệt kiểm thử: Google Chrome, Microsoft Edge

### 2.7.2 Môi trường phát triển

Ngôn ngữ: C#

- Kiến trúc: ASP.NET MVC (Model – View – Controller)

- Razor View Engine: để xây dựng giao diện người dung
- Entity Framework Core (Code First): để làm việc với cơ sở dữ liệu

52

- Microsoft SQL Server (LocalDB): hệ quản trị cơ sở dữ liệu
- ASP.NET Core Identity: để xử lý đăng nhập, phân quyền, quản lý người dùng
- Bootstrap : xây dựng giao diện responsive, thân thiện
- Font Awesome: sử dụng icon
- CSS & JavaScript : để tạo hiệu ứng, cải thiện trải nghiệm giao diện
- .NET SDK: Phiên bản .NET 8.0

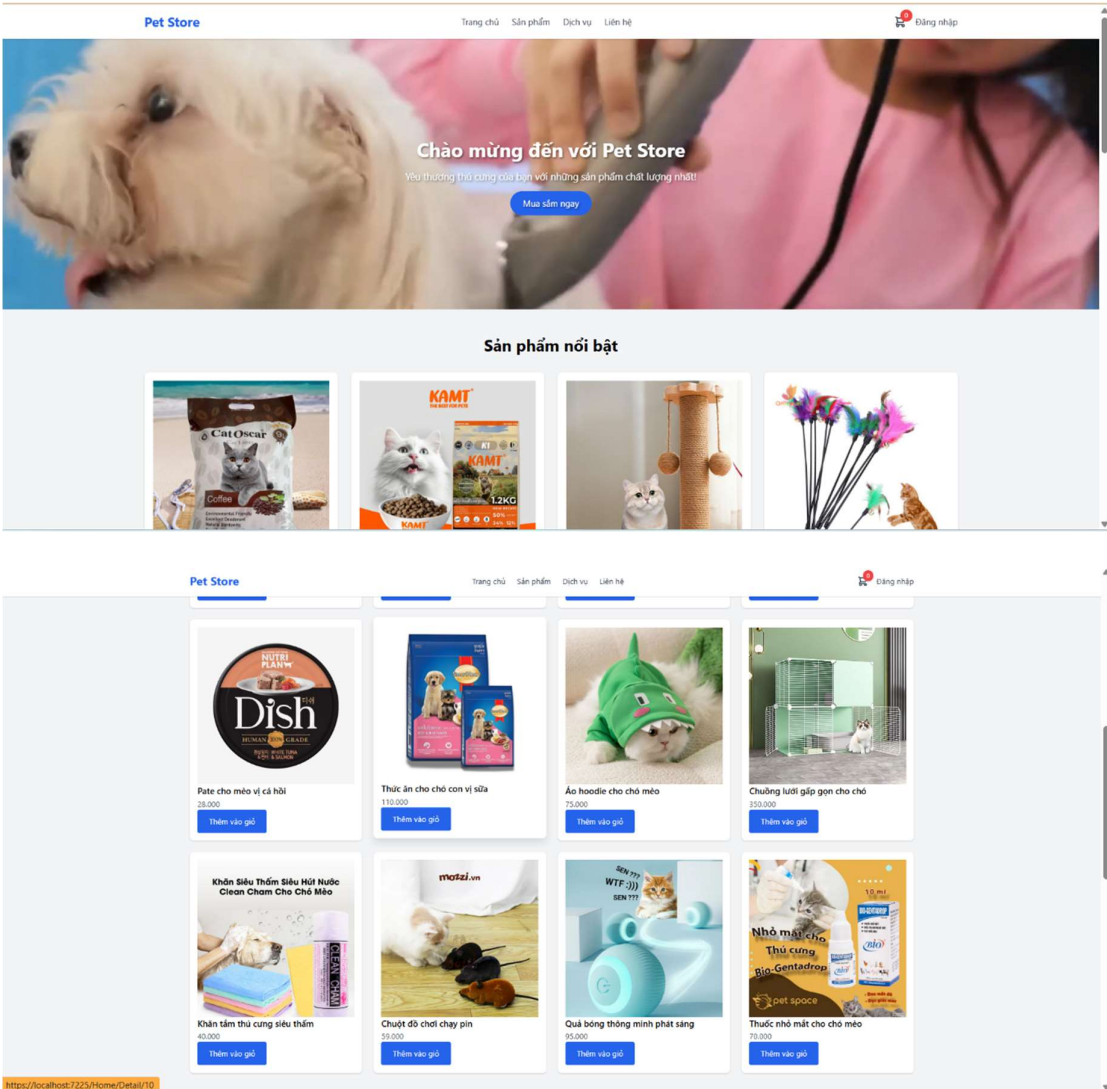
### 2.7.3 Cài đặt

1. Mở Visual studio 2022 tạo project ASP.Net Core (MVC)
2. Cài bộ thư viện EF và Identity
3. Khởi Tạo Model và Data
4. Cấu hình appsetting.json và program
5. Chạy lệnh add-migration và update-database để tạo CSDL theo CodeFisrt
6. Thực hiện viết các controller xử lý logic và view để hiển thị
7. Nhấn Run/Build để chạy

# Chương 3 KẾT QUẢ CHƯƠNG TRÌNH

## 3.1 User

### 3.1.1 Trang chủ



Pet Store

[Trang chủ](#)
[Sản phẩm](#)
[Dịch vụ](#)
[Liên hệ](#)

Đăng nhập

Đánh giá từ khách hàng

"Sản phẩm chất lượng, giao hàng nhanh, thú cưng của tôi rất thích!"

Nguyễn An

"Dịch vụ chăm sóc khách hàng tuyệt vời, sẽ tiếp tục ủng hộ!"

Trần Bình

"Giá cả hợp lý, sản phẩm đa dạng, rất đáng để mua!"

Lê Thị Cẩm

Tin tức & Mẹo chăm sóc thú cưng

Cách chăm sóc mèo con mới sinh

Hướng dẫn chi tiết để chăm sóc mèo con khỏe mạnh.

Chọn thức ăn phù hợp cho chó

Những điều cần biết khi chọn thức ăn cho thú cưng.

Huấn luyện thú cưng cơ bản

Mẹo huấn luyện chó và mèo hiệu quả tại nhà.

Pet Store

Cung cấp các sản phẩm và dịch vụ tốt nhất cho thú cưng của bạn.

Liên kết nhanh

[Trang chủ](#)
[Sản phẩm](#)
[Liên hệ](#)

Liên hệ

Email: [support@petstore.vn](mailto:support@petstore.vn)

Hotline: 0123 456 789

Địa chỉ: 123 Đường Pet, TP.HCM

© 2023 Pet Store. All rights reserved.

### 3.1.2 Chi tiết sản phẩm

Pet Store

[Trang chủ](#)
[Sản phẩm](#)
[Dịch vụ](#)
[Liên hệ](#)

hghuy@gmail.com

Bánh thưởng vị gà cho chó

45.000

Bánh mềm thơm ngon, phù hợp huấn luyện.

Số lượng:

1

Thêm vào giỏ

Thông tin chi tiết

Khối lượng: 100g

Hình dạng nhỏ gọn

Không chất bảo quản

### 3.1.3 Đăng nhập

30

## Đăng nhập

Email

hghuy@gmail.com

Mật khẩu

.....

☐ Ghi nhớ đăng nhập

[Quên mật khẩu?](#)

[Đăng ký](#)

Đăng nhập

### 3.1.4 Đăng ký

## Đăng ký

Họ tên

Email

hghuy@gmail.com

Mật khẩu

Xác nhận mật khẩu

[Đã có tài khoản? Đăng nhập](#)

Đăng ký

### 3.1.5 Giỏ hàng

#### Giỏ hàng của bạn

	<b>Bánh thưởng vị gà cho chó</b> 45.000 VNĐ x 1	- 1 + <a href="#">Xóa</a>
	<b>Cat Foods</b> 320.000 VNĐ x 1	- 1 + <a href="#">Xóa</a>

**Tổng cộng: 365.000 VNĐ**[Thanh toán](#)

[Xóa toàn bộ giỏ hàng](#)

### 3.1.6 Thanh toán



## Thông Tin Thanh Toán

HỌ VÀ TÊN

SỐ ĐIỆN THOẠI

ĐỊA CHỈ GIAO HÀNG

**Xác Nhận Đặt Hàng**

[← Quay Lại Giỏ Hàng](#)

### 3.1.7 Liên Hệ

#### Liên hệ với chúng tôi

**Thông tin liên hệ**  
Email: support@petstore.vn  
Hotline: 0123 456 789  
Địa chỉ: 123 Đường Pet, TP.HCM  
Chúng tôi luôn sẵn sàng hỗ trợ bạn với mọi thắc mắc về sản phẩm và dịch vụ!

#### Gửi tin nhắn

Họ tên


Email

Tin nhắn


[Gửi](#)

### 3.1.8 Dịch vụ


#### Dịch vụ của chúng tôi




**Chăm sóc thú cưng**  
Dịch vụ chăm sóc toàn diện bao gồm tắm rửa, cắt tỉa lông, và kiểm tra sức khỏe cơ bản.  
Từ 200.000 VND  
[Đặt lịch](#)




**Spa thú cưng thư giãn**  
Dịch vụ spa cao cấp cho thú cưng gồm tắm nước ấm, massage nhẹ nhàng và chăm sóc lông mềm mượt.  
Từ 250.000 VND  
[Đặt lịch](#)




**Tắm rửa và vệ sinh tai**  
Dịch vụ tắm gội bằng sữa tắm chuyên dụng, kết hợp vệ sinh tai sạch sẽ cho thú cưng.  
Từ 150.000 VND  
[Đặt lịch](#)



**Cắt móng và làm sạch bàn chân**  
Dịch vụ cắt móng tay, móng chân và làm sạch bàn chân chuyên nghiệp cho chó mèo.  
Từ 100.000 VND  
[Đặt lịch](#)



**Cạo lông vệ sinh vùng kín**  
Dịch vụ cạo lông vùng kín giúp thú cưng sạch sẽ, hạn chế nhiễm khuẩn và mùi hôi.  
Từ 120.000 VND  
[Đặt lịch](#)



**Tư vấn dinh dưỡng và sức khỏe cơ bản**  
Hỗ trợ chủ nuôi xây dựng chế độ ăn và chăm sóc cơ bản phù hợp với từng giống thú cưng.  
Từ 80.000 VND  
[Đặt lịch](#)

### 3.1.9 Đặt lịch

#### Đặt lịch: Spa thú cưng thư giãn

Họ tên

Số điện thoại

[Gửi yêu cầu](#)

## Kết Luận

Website không chỉ đáp ứng nhu cầu người dùng trong việc mua sắm sản phẩm cho thú cưng, mà còn hỗ trợ đặt lịch các dịch vụ chăm sóc như spa, huấn luyện,... Đây là một bước tiến trong việc số hóa lĩnh vực chăm sóc thú cưng – một ngành đang phát triển mạnh mẽ.

### 1. Kiến thức và kỹ năng đạt được

- Lập trình web với HTML, CSS (Tailwind), JavaScript và tích hợp với mô hình MVC.
- Quản lý cơ sở dữ liệu bằng Entity Framework Code First.
- Tổ chức mã nguồn rõ ràng, phân tách theo từng vai trò: Model, View, Controller.
- Thiết kế giao diện người dùng thân thiện, responsive, phù hợp nhiều thiết bị.

### 2. Những điểm còn hạn chế

- Chưa tích hợp hệ thống phân quyền giữa người dùng và admin.
- Tính năng thanh toán chỉ mô phỏng, chưa kết nối với cổng thanh toán thật.
- Hệ thống chưa có quản lý tồn kho hoặc đánh giá, bình luận sản phẩm.

### 3. Hướng phát triển mở rộng

- Tích hợp thanh toán online qua Momo, VNPay hoặc PayPal.
- Bổ sung phân quyền giữa người quản trị và người dùng bình thường.
- Xây dựng trang quản trị riêng cho admin theo dõi thống kê, doanh thu.
- Tạo hệ thống khuyến mãi, giảm giá, đăng ký thành viên VIP cho khách hàng thân thiết.
- Phát triển thành SPA (Single Page Application) hoặc chuyển sang sử dụng framework như React, Vue hoặc Blazor.

### 4. Tổng kết

Trong suốt quá trình thực hiện đề tài “Xây dựng website Pet Store – hệ thống bán hàng và dịch vụ cho thú cưng”, em đã có cơ hội trải nghiệm thực tế toàn bộ quy trình phát triển một ứng dụng web, từ khâu phân tích yêu cầu, thiết kế giao diện, xây dựng chức năng, cho đến kiểm tra và đánh giá kết quả.

Điều đầu tiên em nhận thấy là việc học đi đôi với hành thật sự rất quan trọng. Những kiến thức tưởng như lý thuyết đơn thuần trên lớp như: mô hình MVC, lập trình hướng đối tượng (OOP), Entity Framework Code First hay cách tổ chức dữ liệu,... khi áp dụng vào dự án thực tế lại giúp em hiểu sâu hơn và nhớ lâu hơn. Thay vì chỉ biết lý thuyết, em đã thực sự biết cách ứng dụng để tạo ra một sản phẩm cụ thể có thể chạy được.

Quá trình làm dự án cũng giúp em rèn luyện nhiều kỹ năng mới. Em học được cách tổ chức mã nguồn sao cho rõ ràng, sử dụng Tailwind CSS để tạo giao diện hiện đại, và đặc biệt là cách làm việc với dữ liệu qua mô hình Code First trong Entity Framework. Em cũng đã viết script xử lý giỏ hàng bằng JavaScript, thiết kế các giao diện người dùng như: đăng nhập, sản phẩm, giỏ hàng, dịch vụ,... với bộ cục hợp lý, trực quan.

Bên cạnh đó, em cũng gặp không ít khó khăn như lỗi style giao diện không hiển thị đúng giữa các trình duyệt, phân tách chức năng chưa hợp lý lúc đầu, hay làm sao để logic xử lý giỏ hàng linh hoạt hơn. Tuy nhiên, nhờ kiên trì tìm kiếm tài liệu, tham khảo nguồn mở, nhờ thầy cô hướng dẫn, em đã từng bước giải quyết và cải thiện sản phẩm.

Dự án tuy vẫn còn một số hạn chế như chưa tích hợp chức năng phân quyền (admin – user), chưa có hệ thống thanh toán online thực tế, hay việc chưa có backend hoàn chỉnh để lưu trữ dữ liệu động. Nhưng xét về mặt mục tiêu và yêu cầu đặt ra ban đầu, em cảm thấy bản thân đã hoàn thành khá đầy đủ và có thể tự tin phát triển tiếp trong tương lai.

Ngoài kỹ năng kỹ thuật, đề tài này còn giúp em hiểu hơn về quy trình làm dự án: từ lên ý tưởng, xác định yêu cầu, viết báo cáo, kiểm thử, chỉnh sửa,... Đây là những trải nghiệm quý báu để sau này em có thể tham gia vào các dự án lớn hơn trong học tập cũng như khi đi làm thực tế.

Tóm lại, sau khi hoàn thành dự án này, em cảm thấy bản thân đã trưởng thành hơn rất nhiều cả về kiến thức chuyên môn lẫn kỹ năng làm việc. Em hiểu rõ hơn vai trò của một lập trình viên web trong việc tạo ra những sản phẩm có giá trị ứng dụng, đồng thời cũng nhận thấy rằng việc học công nghệ thông tin không chỉ nằm ở việc viết được đoạn mã, mà quan trọng hơn là viết ra giải pháp cho một vấn đề thực tế.

#### 3.1.10 Tài liệu tham khảo:

[1] Microsoft Docs – ASP.NET Core MVC – Microsoft – Online – 2023  
<https://learn.microsoft.com/en-us/aspnet/core/mvc>

[2] Entity Framework Core Documentation – Microsoft – Online – 2023

<https://learn.microsoft.com/en-us/ef/core/>

[3] ASP.NET Core Identity Overview – Microsoft – Online – 2023

<https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity>

[4] Xây dựng website với ASP.NET Core MVC – Ths. Nguyễn Đức Tấn – 2025

Đại học Yersin (Bài giảng nội bộ)

