**BỘ GIÁO DỤC VÀ ĐÀO TẠO TRƯỜNG ĐẠI HỌC ĐẠI NAM**



BÀI TẬP LỚN

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

**ĐỀ TÀI: THIẾT KẾ VÀ XÂY DỰNG WEBSITE DẠY HỌC LỊCH SỬ - ĐỊA LÝ ĐỊA PHƯƠNG**

**Giáo viên hướng dẫn: ThS. Phạm Văn Tiệp Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sinh viên** | **Họ và tên** | **Lớp** |
| 1 | 1671020360 | Đinh Công Vỹ | CNTT 16– 06 |

**Hà Nội, năm 2024**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO TRƯỜNG ĐẠI HỌC ĐẠI NAM**  Logo, company name  Description automatically generated  **BÀI TẬP LỚN**  **TÊN HỌC PHẦN: THIẾT KẾ, LẬP TRÌNH BACK-END ĐỀ TÀI: THIẾT KẾ VÀ XÂY DỰNG WEBSITE DẠY HỌC LỊCH SỬ - ĐỊA LÝ ĐỊA PHƯƠNG** | | | | | | |
|  | **STT** | **Mã Sinh Viên** | **Họ và Tên** | **Ngày Sinh** | **Điểm** | |
| **Bằng Số** | **Bằng Chữ** |
| **1** | **1671020360** | **Đinh Công Vỹ** | **24/08/2004** |  |  |
| **CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2**  **Hà Nội, năm 2024** | | | | | | |

# LỜI NÓI ĐẦU

Trong thời đại công nghệ số hiện nay, việc xây dựng và phát triển các trang web giáo dục đóng một vai trò vô cùng quan trọng trong việc cung cấp kiến thức và kết nối người học với nguồn tài nguyên học tập. Đặc biệt, các trang web dạy học trực tuyến không chỉ giúp người học tiếp cận dễ dàng với các tài liệu học tập mà còn mang lại trải nghiệm học tập tương tác và hiệu quả hơn.

Trong khuôn khổ môn học "Thiết Kế Lập Trình Back-End bằng ASP.NET Core," em đã chọn chủ đề "Thiết Kế và Xây Dựng Website Dạy Học Lịch Sử - Địa Lý Địa Phương" để thực hiện dự án cuối khóa.

Đây là một cơ hội để em áp dụng những kiến thức và kỹ năng đã học được về lập trình back-end vào thực tế, đồng thời phát triển khả năng giải quyết các vấn đề kỹ thuật phức tạp liên quan đến giáo dục trực tuyến.

Báo cáo này sẽ trình bày quá trình thiết kế và xây dựng một trang web dạy học lịch sử - địa lý địa phương từ giai đoạn lên ý tưởng, thiết kế hệ thống, triển khai mã nguồn, đến kiểm thử và đánh giá sản phẩm cuối cùng. Em sẽ đi sâu vào các kỹ thuật và công nghệ sử dụng trong quá trình phát triển, bao gồm ASP.NET Core - một framework mạnh mẽ và phổ biến cho lập trình web back-end, cũng như các công cụ hỗ trợ khác như Entity Framework Core, SQL Server, và các API liên quan.

Em hy vọng rằng qua báo cáo này, không chỉ bản thân mà cả những người đọc sẽ có được cái nhìn tổng quan và chi tiết về quy trình phát triển một website giáo dục hiện đại và hiệu quả. Em rất mong nhận được những góp ý và phản hồi quý báu từ giảng viên và các bạn đồng học để có thể hoàn thiện hơn sản phẩm của mình và nâng cao kiến thức chuyên môn.

Xin chân thành cảm ơn!

# MỤC LỤC

[LỜI NÓI ĐẦU 3](#_Toc169123526)

[MỤC LỤC 4](#_Toc169123527)

[CHƯƠNG 1. MỞ ĐẦU 7](#_Toc169123528)

[1.1. Giới thiệu chung 7](#_Toc169123529)

[1.2. Mục tiêu của website 7](#_Toc169123530)

[1.3. Phạm vi và đối tượng sử dụng 7](#_Toc169123531)

[1.4. Lợi ích của website 9](#_Toc169123532)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 10](#_Toc169123533)

[2.1. Cơ sở lý thuyết về SQL Server 10](#_Toc169123534)

[\* Kiến trúc của SQL Server: 10](#_Toc169123535)

[\* Ngôn ngữ SQL: 10](#_Toc169123536)

[\* Tính năng nổi bật của SQL Server: 11](#_Toc169123537)

[2.2. Cơ sở lý thuyết về ASP.NET Core 11](#_Toc169123538)

[\* Kiến trúc của ASP.NET Core: 11](#_Toc169123539)

[\* Các thành phần chính: 11](#_Toc169123540)

[\* Tính năng nổi bật của ASP.NET Core: 12](#_Toc169123541)

[2.3. Cơ sở lý thuyết về Mô Hình MVC 12](#_Toc169123542)

[\* Lợi ích của mô hình MVC: 13](#_Toc169123543)

[2.4. Cơ sở lý thuyết về lập trình Back-End 13](#_Toc169123544)

[\* Chức năng của Back-End: 13](#_Toc169123545)

[\* Công nghệ và công cụ Back-End phổ biến: 14](#_Toc169123546)

[\* Tính chất của lập trình Back-End: 14](#_Toc169123547)

[CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 15](#_Toc169123548)

[3.1. Phân Tích Về Người Dùng 15](#_Toc169123549)

[\* Khách hàng cá nhân: 15](#_Toc169123550)

[\* Doanh nghiệp: 15](#_Toc169123551)

[\* Người đam mê công nghệ: 16](#_Toc169123552)

[3.2. Phân Tích Về Hệ Thống 17](#_Toc169123553)

[\* Chức năng của hệ thống: 17](#_Toc169123554)

[\* Kiến trúc hệ thống: 17](#_Toc169123555)

[\* Use Case Diagram 19](#_Toc169123556)

[Khách hàng: 19](#_Toc169123557)

[CHƯƠNG 4. THIẾT KẾ CƠ SỞ DỮ LIỆU 20](#_Toc169123558)

[4.1. Thiết Kế Các Bảng Dữ Liệu 20](#_Toc169123559)

[4.1.1. Bảng Manufacturer (Nhà sản xuất) 20](#_Toc169123560)

[4.1.2. Bảng Category (Loại sản phẩm) 20](#_Toc169123561)

[4.1.3. Bảng Image (Hình ảnh của sản phẩm) 20](#_Toc169123562)

[4.1.4. Bảng Product (Sản phẩm) 21](#_Toc169123563)

[4.1.5. Bảng Specification (Thông số kỹ thuật của sản phẩm) 21](#_Toc169123564)

[4.1.6. Bảng Role (Vai trò người dùng) 22](#_Toc169123565)

[4.1.7. Bảng User (Người dùng) 22](#_Toc169123566)

[4.2. Mô Hình ERD (Entity-Relationship Diagram) 23](#_Toc169123567)

[4.3. Ràng Buộc Dữ Liệu và Toàn Vẹn 24](#_Toc169123568)

[CHƯƠNG 5. CÁC CHỨC NĂNG CỦA WEBSITE 27](#_Toc169123569)

[5.1. Chức Năng Đăng Ký 27](#_Toc169123570)

[Bước 1: Tạo file UserController.cs 27](#_Toc169123571)

[Bước 2: Tạo 1 ViewModels LoginVM.cs 28](#_Toc169123572)

[5.2. Chức Năng Đăng Nhập 31](#_Toc169123573)

[Bước 1: Tạo file UserController.cs 31](#_Toc169123574)

[Bước 2: Tạo 1 ViewModels LoginVM.cs 31](#_Toc169123575)

[Bước 3: Tạo View DangNhap.cshtml 32](#_Toc169123576)

[5.3. Chức Năng Lọc Sản Phẩm Theo Loại Và Theo Hãng 33](#_Toc169123577)

[Bước 1: Tạo Controller ShopController.cs 33](#_Toc169123578)

[Bước 3 : Tạo View 36](#_Toc169123579)

[5.4. Chức Năng Xem Chi Tiết Sản Phẩm 38](#_Toc169123580)

[Bước 1: Tạo Controller : ShopController 38](#_Toc169123581)

[Bước 2: Tạo View : Detail.cshtml 38](#_Toc169123582)

[5.5. Chức Năng Thêm Sản Phẩm Vào Giỏ Hàng 40](#_Toc169123583)

[Bước 1 : Tạo Controller : CartController 40](#_Toc169123584)

[Bước 2 : Tạo View : Index.cshtml 42](#_Toc169123585)

[CHƯƠNG 6. GIAO DIỆN NGƯỜI DÙNG 44](#_Toc169123586)

[6.1. Phần HOME ( Trang chủ ) 46](#_Toc169123587)

[6.2. Phần SHOP ( Hiển thị các sản phẩm ) 46](#_Toc169123588)

[6.3. Phần CART ( Giỏ hàng ) 49](#_Toc169123589)

[Ưu điểm: 52](#_Toc169123590)

[Nhược điểm: 52](#_Toc169123591)

[Hướng phát triển: 52](#_Toc169123592)

[DANH MỤC TÀI LIỆU THAM KHẢO 53](#_Toc169123593)

# CHƯƠNG 1. MỞ ĐẦU

## Giới thiệu chung

Trong bối cảnh công nghệ phát triển mạnh mẽ như hiện nay, việc học tập và giảng dạy trực tuyến đã trở thành một phần không thể thiếu trong đời sống của nhiều người. Các trang web giáo dục trực tuyến cung cấp một nền tảng tiện lợi để học sinh, giáo viên và những người đam mê lịch sử, địa lý có thể dễ dàng tiếp cận, tìm kiếm, và học hỏi kiến thức một cách nhanh chóng và hiệu quả. Trong số các lĩnh vực giáo dục, lịch sử và địa lý địa phương là những môn học có tầm quan trọng đặc biệt, giúp học sinh hiểu rõ hơn về quê hương mình, cũng như phát triển lòng tự hào dân tộc và ý thức bảo vệ di sản văn hóa, môi trường.

## Mục tiêu của website

Website bán laptop được thiết kế và xây dựng nhằm đạt được những mục tiêu sau:

* + - Cung cấp thông tin chi tiết về sản phẩm: Giới thiệu đầy đủ và chi tiết về các loại laptop bao gồm thông số kỹ thuật, tính năng nổi bật, hình ảnh, và giá cả.
    - Tạo trải nghiệm mua sắm thuận tiện: Cung cấp giao diện người dùng thân thiện, dễ dàng điều hướng và tìm kiếm sản phẩm, hỗ trợ khách hàng trong quá trình mua sắm.
    - Tăng cường tương tác với khách hàng: Hỗ trợ các chức năng đánh giá, nhận xét sản phẩm, cũng như hệ thống tư vấn và hỗ trợ khách hàng trực tuyến.
    - Quản lý bán hàng hiệu quả: Hỗ trợ các công cụ quản lý hàng tồn kho, đơn hàng, và thông tin khách hàng một cách hiệu quả.

## Phạm vi và đối tượng sử dụng

Website dạy học lịch sử - địa lý địa phương hướng đến đối tượng người dùng đa dạng:

- Học sinh: Những em học sinh cần tìm hiểu và học tập về lịch sử và địa lý địa phương như một phần của chương trình học.

- Giáo viên: Các thầy cô giáo cần tài liệu giảng dạy và tài nguyên hỗ trợ để nâng cao chất lượng bài giảng về lịch sử và địa lý địa phương.

- Người yêu thích lịch sử và địa lý: Những người quan tâm và muốn tìm hiểu sâu hơn về lịch sử và địa lý của địa phương mình.

- Nhà nghiên cứu: Các nhà nghiên cứu và chuyên gia muốn thu thập thông tin và tài liệu nghiên cứu liên quan đến lịch sử và địa lý địa phương.

## 1.4 Lợi ích của website

Việc triển khai website dạy học lịch sử - địa lý địa phương mang lại nhiều lợi ích cho cả giáo viên, học sinh và người yêu thích lịch sử, địa lý:

- Tiết kiệm thời gian và chi phí: Học sinh và giáo viên có thể dễ dàng truy cập và sử dụng tài liệu học tập, tài nguyên giảng dạy mà không cần phải di chuyển, giúp tiết kiệm thời gian và chi phí đi lại.

- Mở rộng phạm vi tiếp cận: Thông tin về lịch sử và địa lý địa phương có thể tiếp cận được nhiều người hơn, không bị giới hạn bởi khoảng cách địa lý.

- Cập nhật thông tin nhanh chóng: Người dùng có thể nhanh chóng cập nhật các thông tin mới về lịch sử, địa lý địa phương, bao gồm các sự kiện lịch sử quan trọng, địa danh nổi bật và các nghiên cứu mới nhất.

- Cải thiện trải nghiệm học tập: Hỗ trợ học sinh và giáo viên mọi lúc, mọi nơi thông qua hệ thống học trực tuyến, các kênh liên lạc và hỗ trợ khác. Điều này giúp nâng cao chất lượng dạy và học, tạo ra môi trường học tập linh hoạt và hiệu quả hơn.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## Cơ sở lý thuyết về SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft, được sử dụng rộng rãi trong các ứng dụng doanh nghiệp để quản lý và lưu trữ dữ liệu. SQL Server cung cấp một môi trường mạnh mẽ và an toàn cho việc quản lý cơ sở dữ liệu với nhiều tính năng tiên tiến.

### Kiến trúc của SQL Server:

* Client-Server Architecture: SQL Server hoạt động dựa trên kiến trúc client-server, nơi các ứng dụng khách gửi yêu cầu SQL tới máy chủ SQL Server để truy cập và quản lý dữ liệu.
* Database Engine: Đây là thành phần cốt lõi của SQL Server, chịu trách nhiệm xử lý các lệnh SQL, thực hiện các hoạt động CRUD (Create, Read, Update, Delete) và quản lý giao dịch.
* SQL Server Management Studio (SSMS): Công cụ quản lý đồ họa cho phép người dùng tương tác với SQL Server, thực hiện các truy vấn, quản lý cơ sở dữ liệu và thực hiện các tác vụ quản trị.
* SQL Server Agent: Dịch vụ giúp lập lịch và quản lý các tác vụ tự động như sao lưu, bảo trì và thực thi các script SQL.

### Ngôn ngữ SQL:

* Data Definition Language (DDL): Sử dụng để định nghĩa và quản lý cấu trúc cơ sở dữ liệu, bao gồm các lệnh như CREATE, ALTER, DROP.
* Data Manipulation Language (DML): Sử dụng để thao tác dữ liệu trong cơ sở dữ liệu, bao gồm các lệnh như SELECT, INSERT, UPDATE, DELETE.
* Data Control Language (DCL): Quản lý quyền truy cập và bảo mật cơ sở dữ liệu, bao gồm các lệnh GRANT và REVOKE.
* Transaction Control Language (TCL): Quản lý giao dịch trong cơ sở dữ liệu, bao gồm các lệnh COMMIT, ROLLBACK.

### Tính năng nổi bật của SQL Server:

* Bảo mật: SQL Server cung cấp các cơ chế bảo mật tiên tiến như xác thực người dùng, phân quyền truy cập, mã hóa dữ liệu và bảo vệ chống các tấn công SQL injection.
* Tính toàn vẹn dữ liệu: Hỗ trợ các ràng buộc dữ liệu, khóa ngoại và các quy tắc nghiệp vụ để đảm bảo tính toàn vẹn và nhất quán của dữ liệu.
* Hiệu suất cao: Tối ưu hóa việc xử lý các truy vấn, sử dụng các chỉ mục, bộ đệm và các kỹ thuật tối ưu hóa khác để cải thiện hiệu suất.
* Khả năng mở rộng: Hỗ trợ xử lý và lưu trữ khối lượng dữ liệu lớn, phù hợp cho các ứng dụng có quy mô lớn và phức tạp.
* Sao lưu và phục hồi: Cung cấp các tùy chọn sao lưu và phục hồi dữ liệu để đảm bảo tính liên tục và an toàn của hệ thống.

## Cơ sở lý thuyết về ASP.NET Core

ASP.NET Core là một framework mã nguồn mở, đa nền tảng do Microsoft phát triển, được thiết kế để xây dựng các ứng dụng web hiện đại, dịch vụ web và API hiệu suất cao.

### Kiến trúc của ASP.NET Core:

* Middleware: ASP.NET Core sử dụng pipeline của các middleware để xử lý các yêu cầu HTTP. Middleware là các thành phần nhỏ được xâu chuỗi với nhau để xử lý các yêu cầu và phản hồi.
* Dependency Injection (DI): Tích hợp sẵn DI giúp quản lý các đối tượng và dịch vụ, giúp cải thiện khả năng mở rộng và bảo trì mã nguồn.
* Hosting: ASP.NET Core có thể được host trên nhiều nền tảng khác nhau như Kestrel, IIS, Nginx, Apache, hoặc tự host trong một ứng dụng console.

### Các thành phần chính:

* Razor Pages: Một mô hình lập trình trang đơn giản cho phép kết hợp giữa mã C# và HTML để tạo ra các trang động.
* MVC (Model-View-Controller): Mô hình thiết kế phân chia ứng dụng thành ba phần: Model (quản lý dữ liệu và logic nghiệp vụ), View (giao diện người dùng), và Controller (xử lý yêu cầu và điều hướng).
* Blazor: Một framework cho phép xây dựng các ứng dụng web tương tác với C# và HTML, có thể chạy trên máy chủ hoặc trên trình duyệt thông qua WebAssembly.

### Tính năng nổi bật của ASP.NET Core:

* Hiệu suất cao: Được thiết kế để đạt hiệu suất tối ưu, sử dụng ít tài nguyên hệ thống và có khả năng xử lý nhiều yêu cầu đồng thời.
* Bảo mật: Cung cấp các tính năng bảo mật tiên tiến như xác thực, ủy quyền, bảo vệ chống tấn công XSS (Cross-Site Scripting) và CSRF (Cross-Site Request Forgery).
* Cross-platform: Chạy được trên nhiều hệ điều hành khác nhau như Windows, Linux và macOS, hỗ trợ việc triển khai ứng dụng trên nhiều môi trường khác nhau.
* Kết hợp với các công nghệ hiện đại: Hỗ trợ tích hợp với Docker, Kubernetes, và các dịch vụ cloud như Azure.

## Cơ sở lý thuyết về Mô Hình MVC

Mô hình MVC (Model-View-Controller) là một mẫu kiến trúc phần mềm phổ biến được sử dụng trong phát triển ứng dụng web, giúp tách biệt các thành phần của ứng dụng để cải thiện khả năng quản lý và bảo trì.

* **Model:** Đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng. Model chịu trách nhiệm truy xuất dữ liệu từ cơ sở dữ liệu và thực hiện các quy tắc nghiệp vụ. Các thao tác CRUD (Create, Read, Update, Delete) thường được thực hiện tại đây.
* **View:** Đại diện cho giao diện người dùng. View hiển thị dữ liệu từ Model và cung cấp giao diện cho người dùng tương tác. View thường là các tệp HTML, kết hợp với mã Razor hoặc các template engine khác để render dữ liệu động.
* **Controller:** Điều khiển luồng dữ liệu giữa Model và View. Controller nhận các yêu cầu từ người dùng, xử lý chúng bằng cách tương tác với Model, và sau đó trả về View thích hợp để hiển thị kết quả.

### \* Lợi ích của mô hình MVC:

* Tách biệt trách nhiệm: Giúp mã nguồn trở nên rõ ràng và dễ quản lý hơn, mỗi thành phần có một nhiệm vụ cụ thể.
* Khả năng mở rộng: Dễ dàng mở rộng từng phần của ứng dụng mà không ảnh hưởng đến các phần khác.
* Tái sử dụng mã nguồn: Các thành phần của MVC có thể được tái sử dụng trong các ứng dụng khác nhau, giúp tiết kiệm thời gian và công sức phát triển.
* Dễ kiểm thử: Mô hình MVC cho phép kiểm thử từng thành phần một cách độc lập, giúp đảm bảo chất lượng phần mềm.

## Cơ sở lý thuyết về lập trình Back-End

Lập trình Back-End là một phần quan trọng của phát triển ứng dụng web, chịu trách nhiệm xử lý logic nghiệp vụ, quản lý cơ sở dữ liệu, xác thực người dùng, và cung cấp API cho Front-End.

### Chức năng của Back-End:

* Quản lý cơ sở dữ liệu: Thực hiện các thao tác với cơ sở dữ liệu như truy vấn, cập nhật, xóa và chèn dữ liệu. Các thao tác này thường được thực hiện bằng cách sử dụng SQL hoặc các ORM (Object-Relational Mapping) như Entity Framework.
* Xử lý logic nghiệp vụ: Thực hiện các quy tắc và quy trình nghiệp vụ, đảm bảo rằng dữ liệu và các hành động của người dùng được xử lý một cách đúng đắn.
* Xác thực và ủy quyền: Quản lý quyền truy cập của người dùng, bảo vệ dữ liệu và tài nguyên của hệ thống. Điều này bao gồm việc xác thực người dùng (authentication) và xác định quyền hạn của họ (authorization).
* Cung cấp API: Tạo các dịch vụ web (RESTful API, GraphQL) để giao tiếp với Front-End và các hệ thống khác. API giúp tách biệt logic phía máy chủ và phía khách hàng, đồng thời cho phép ứng dụng dễ dàng mở rộng và tích hợp.

### Công nghệ và công cụ Back-End phổ biến:

* Ngôn ngữ lập trình: Các ngôn ngữ phổ biến bao gồm C#, Java, Python, PHP, Node.js và Ruby.
* Framework: Các framework phổ biến bao gồm ASP.NET Core (C#), Spring Boot (Java), Django (Python), Express.js (Node.js), Laravel (PHP) và Ruby on Rails (Ruby).
* Cơ sở dữ liệu: Các hệ quản trị cơ sở dữ liệu phổ biến bao gồm SQL Server, MySQL, PostgreSQL, MongoDB và Oracle.
* Công cụ quản lý: Docker (containerization), Kubernetes (orchestration), Git (version control), Jenkins (CI/CD) và các công cụ khác để quản lý và triển khai ứng dụng.

### Tính chất của lập trình Back-End:

* Hiệu suất: Back-End phải xử lý các yêu cầu và truy vấn dữ liệu một cách nhanh chóng, tối ưu hóa tài nguyên hệ thống để đảm bảo hiệu suất cao.
* Bảo mật: Đảm bảo an toàn dữ liệu và bảo vệ khỏi các tấn công mạng. Điều này bao gồm mã hóa dữ liệu, sử dụng HTTPS, và các biện pháp bảo mật khác.
* Ổn định: Hệ thống phải hoạt động ổn định và xử lý lỗi một cách hiệu quả để đảm bảo rằng dịch vụ không bị gián đoạn.
* Khả năng mở rộng: Back-End phải hỗ trợ tăng trưởng về số lượng người dùng và khối lượng dữ liệu mà không làm giảm hiệu suất. Điều này có thể đạt được thông qua việc thiết kế hệ thống phân tán, cân bằng tải và các chiến lược mở rộng khác.

# CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Phân Tích Về Người Dùng

Việc phân tích người dùng là bước quan trọng nhằm hiểu rõ nhu cầu và mong muốn của các đối tượng sử dụng website bán laptop. Dưới đây là các nhóm người dùng chính và yêu cầu của họ:

### Khách hàng cá nhân:

* Nhu cầu: Tìm kiếm, mua các sản phẩm laptop phù hợp với nhu cầu cá nhân như học tập, làm việc, giải trí.

Yêu cầu:

* + Giao diện dễ sử dụng, dễ điều hướng.
  + Thông tin sản phẩm chi tiết, bao gồm thông số kỹ thuật, hình ảnh, đánh giá và giá cả.
  + Tính năng tìm kiếm và lọc sản phẩm theo nhiều tiêu chí (hãng, cấu hình, v.v.).
  + Hỗ trợ tư vấn trực tuyến và chính sách bảo hành rõ ràng.
  + Quy trình mua hàng đơn giản và an toàn.

### Doanh nghiệp:

* Nhu cầu: Mua sắm laptop số lượng lớn cho nhân viên, trang bị văn phòng với chi phí tối ưu.

Yêu cầu:

* + Tùy chọn mua hàng số lượng lớn và chiết khấu.
  + Hỗ trợ báo giá và hợp đồng mua bán.
  + Quản lý đơn hàng và theo dõi giao hàng.
  + Hỗ trợ sau bán hàng và dịch vụ bảo hành tốt.

### Người đam mê công nghệ:

* Nhu cầu: Cập nhật các sản phẩm laptop mới nhất, công nghệ tiên tiến và các xu hướng thị trường.

Yêu cầu:

* + Thông tin chi tiết về các sản phẩm mới, đánh giá chuyên sâu và bài viết về công nghệ.
  + Tính năng so sánh sản phẩm và đánh giá từ người dùng khác.
  + Cập nhật thường xuyên về các chương trình khuyến mãi và giảm giá.

## Phân Tích Về Hệ Thống

Phân tích hệ thống nhằm xác định các yêu cầu kỹ thuật và thiết kế kiến trúc cho website bán laptop. Các yếu tố quan trọng cần xem xét bao gồm:

### Chức năng của hệ thống:

* Quản lý sản phẩm: Hệ thống cần cho phép quản trị viên thêm, sửa, xóa và quản lý thông tin sản phẩm.
* Quản lý người dùng: Bao gồm quản lý thông tin tài khoản khách hàng, đăng ký, đăng nhập, và quyền hạn người dùng.
* Giỏ hàng và thanh toán: Khách hàng có thể thêm sản phẩm vào giỏ hàng, tiến hành thanh toán trực tuyến an toàn và theo dõi đơn hàng.
* Tìm kiếm và lọc sản phẩm: Tính năng tìm kiếm nâng cao, lọc sản phẩm theo nhiều tiêu chí để giúp người dùng dễ dàng tìm thấy sản phẩm mong muốn.
* Đánh giá và nhận xét sản phẩm: Khách hàng có thể đánh giá và nhận xét về sản phẩm, giúp người dùng khác có cái nhìn tổng quan.
* Hỗ trợ trực tuyến: Tính năng chat trực tuyến để hỗ trợ khách hàng nhanh chóng.

### Kiến trúc hệ thống:

* Mô hình MVC (Model-View-Controller): Phân tách ứng dụng thành ba phần chính để dễ dàng quản lý và mở rộng.

**Model:** Quản lý dữ liệu và logic nghiệp vụ.

**View:** Hiển thị giao diện người dùng.

**Controller:** Xử lý các yêu cầu từ người dùng và điều hướng giữa Model và View.

* Cơ sở dữ liệu: Sử dụng SQL Server để quản lý dữ liệu sản phẩm, người dùng, đơn hàng, v.v.
* API: Xây dựng các API để giao tiếp giữa Front-End và Back-End, cũng như hỗ trợ tích hợp với các hệ thống khác.
* Bảo mật: Đảm bảo an toàn dữ liệu, xác thực người dùng và bảo vệ khỏi các tấn công mạng (XSS, CSRF, SQL Injection).
* Hiệu suất và khả năng mở rộng: Tối ưu hóa hiệu suất xử lý và thiết kế hệ thống có khả năng
* mở rộng để đáp ứng nhu cầu tăng trưởng.

### Use Case Diagram

Sơ đồ Use Case giúp xác định các chức năng của hệ thống và mối quan hệ giữa các chức năng này với người dùng.

Actors (Tác nhân):

**Khách hàng:** Đăng ký, đăng nhập, đăng xuất, tìm kiếm sản phẩm, lọc sản phẩm theo loại, hãng, thêm vào giỏ hàng, thanh toán, đánh giá sản phẩm.

**Quản trị viên:** Quản lý sản phẩm, quản lý người dùng, quản lý đơn hàng…..

Use Cases (Trường hợp sử dụng):

## Khách hàng:

* Đăng ký tài khoản
* Đăng nhập/Đăng xuất
* Tìm kiếm và lọc sản phẩm
* Xem chi tiết sản phẩm
* Thêm sản phẩm vào giỏ hàng

# CHƯƠNG 4. THIẾT KẾ CƠ SỞ DỮ LIỆU

Trong chương này, em sẽ trình bày chi tiết về thiết kế cơ sở dữ liệu cho website bán laptop. Thiết kế này sẽ bao gồm các bảng chính, các mối quan hệ giữa các bảng, và các ràng buộc dữ liệu để đảm bảo tính toàn vẹn của hệ thống.

## Thiết Kế Các Bảng Dữ Liệu

### Bảng Manufacturer (Nhà sản xuất)

Bảng Manufacturer lưu trữ thông tin về các nhà sản xuất laptop.

create table Manufacturer (

id int IDENTITY(1,1) primary key, name nvarchar(200) not null

);

### Bảng Category (Loại sản phẩm)

Bảng Category lưu trữ các loại sản phẩm khác nhau của laptop.

create table Category (

id int IDENTITY(1,1) primary key, name nvarchar(200) not null

);

### Bảng Image (Hình ảnh của sản phẩm)

Bảng Image lưu trữ các hình ảnh liên quan đến sản phẩm.

create table Image (

id int IDENTITY(1,1) primary key, url nvarchar(500),

product\_id int foreign key references Product(id)

);

### Bảng Product (Sản phẩm)

Bảng Product lưu trữ thông tin về các sản phẩm laptop, bao gồm tên, giá, số lượng, giảm giá, mô tả, và các mối quan hệ với nhà sản xuất và loại sản phẩm.

CREATE TABLE [dbo].[HangHoa](

[MaHH] [int] IDENTITY(1,1) NOT NULL,

[TenHH] [nvarchar](50) NOT NULL,

[TenAlias] [nvarchar](50) NULL,

[MaLoai] [int] NOT NULL,

[MoTaDonVi] [nvarchar](50) NULL,

[DonGia] [float] NULL,

[Hinh] [nvarchar](50) NULL,

[NgaySX] [datetime] NOT NULL,

[GiamGia] [float] NOT NULL,

[SoLanXem] [int] NOT NULL,

[MoTa] [nvarchar](max) NULL,

[MaNCC] [nvarchar](50) NOT NULL,

### Bảng Specification (Thông số kỹ thuật của sản phẩm)

Bảng Specification lưu trữ các thông số kỹ thuật chi tiết của từng sản phẩm laptop.

create table Specification (

id int IDENTITY(1,1) primary key, cpu nvarchar(100),

ram nvarchar(100), graphics\_card nvarchar(100), operating\_system nvarchar(100),

dimensions nvarchar(100), weight nvarchar(100),

product\_id int foreign key references Product(id)

);

### Bảng Role (Vai trò người dùng)

Bảng Role lưu trữ các vai trò của người dùng trong hệ thống.

create table Role (

id int IDENTITY(1,1) primary key, name nvarchar(50)

);

### Bảng User (Người dùng)

Bảng User lưu trữ thông tin về người dùng, bao gồm tên đầy đủ, tên đăng nhập, mật khẩu, email, số điện thoại, địa chỉ, trạng thái và vai trò.

create table [User] (

id int IDENTITY(1,1) primary key, fullname nvarchar(100),

username nvarchar(50) not null, password nvarchar(255),

email nvarchar(100), phonenumber nvarchar(20), address nvarchar(500), status int,

role\_id int foreign key references Role(id)

);

## Mô Hình ERD (Entity-Relationship Diagram)

Mô hình ERD dưới đây mô tả các thực thể (entities) và mối quan hệ (relationships) giữa chúng trong cơ sở dữ liệu của hệ thống bán laptop.

* Thực thể **Manufacturer**: Bao gồm các thuộc tính id và name.
* Thực thể **Category**: Bao gồm các thuộc tính id và name

- Thực thể **Product**: Bao gồm các thuộc tính id, name, price, quantity, discount, description, thumbnail, status, create\_at, update\_at, manufacturer\_id, category\_id.

* Thực thể **Image**: Bao gồm các thuộc tính id, url, product\_id.
* Thực thể **Specification**: Bao gồm các thuộc tính id, cpu, ram, graphics\_card, operating\_system, dimensions, weight, product\_id.
* Thực thể **Role**: Bao gồm các thuộc tính id và name.
* Thực thể **User**: Bao gồm các thuộc tính id, fullname, username, password, email, phonenumber, address, status, role\_id.
* Thực thể **Order**: Bao gồm các thuộc tính id, fullname, phonenumber, email, note, address, orderdate, status, total, user\_id.
* Thực thể **OrderDetail**: Bao gồm các thuộc tính id, price, quantity, total, status, product\_id, order\_id.
* Thực thể **WishList**: Bao gồm các thuộc tính id, product\_id, user\_id.
* Thực thể **Feedback**: Bao gồm các thuộc tính id, name, email, subject, message, created\_at, status, user\_id.

Các thực thể này được kết nối với nhau thông qua các khóa ngoại (foreign keys) để tạo ra các mối quan hệ giữa các bảng.

## Ràng Buộc Dữ Liệu và Toàn Vẹn

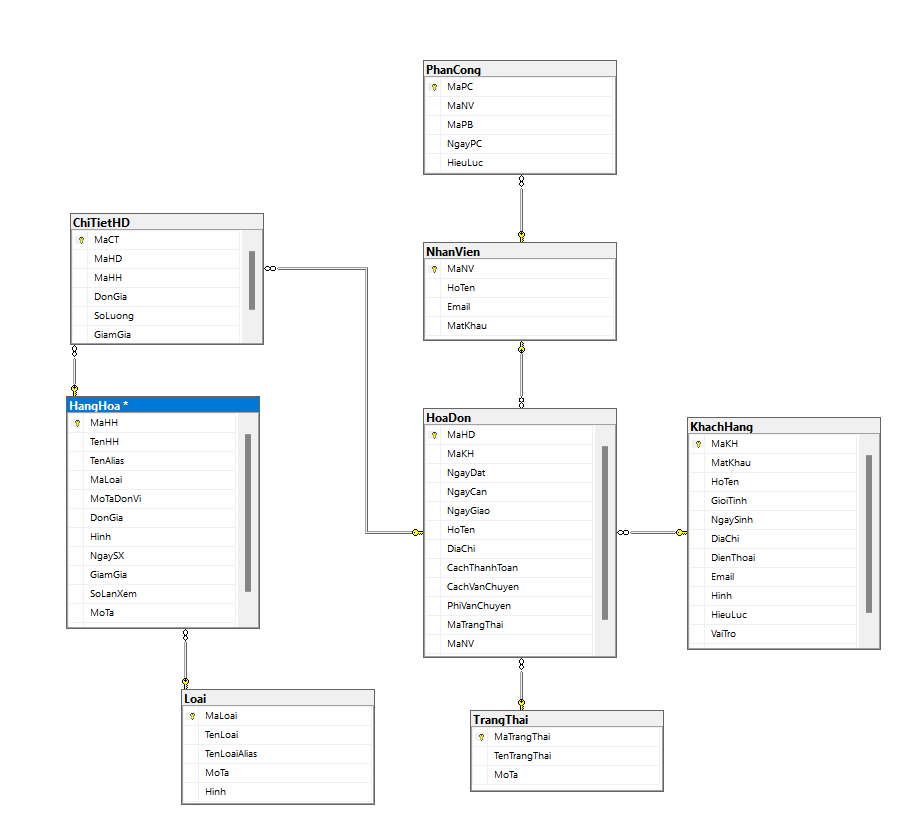
Khóa chính **(Primary Key):** Mỗi bảng có một khóa chính duy nhất để định danh các bản ghi, ví dụ id trong các bảng Product, User, Order, v.v.

Khóa ngoại **(Foreign Key):** Được sử dụng để tạo mối quan hệ giữa các bảng, đảm bảo tính toàn vẹn dữ liệu. Ví dụ, manufacturer\_id trong bảng Product là khóa ngoại tham chiếu đến id trong bảng Manufacturer.

Ràng buộc **NOT NULL:** Đảm bảo các trường bắt buộc phải có giá trị, ví dụ name trong bảng Product.

Ràng buộc **UNIQUE:** Đảm bảo các giá trị trong cột phải là duy nhất, ví dụ username trong bảng User.

Thiết kế cơ sở dữ liệu này giúp đảm bảo hệ thống hoạt động hiệu quả, dữ liệu được tổ chức khoa học và có thể mở rộng trong tương lai. Các bảng và mối quan hệ đã được tối ưu hóa để hỗ trợ các chức năng cần thiết của website bán laptop.

**

Hình 1: Cơ sở dữ liệu

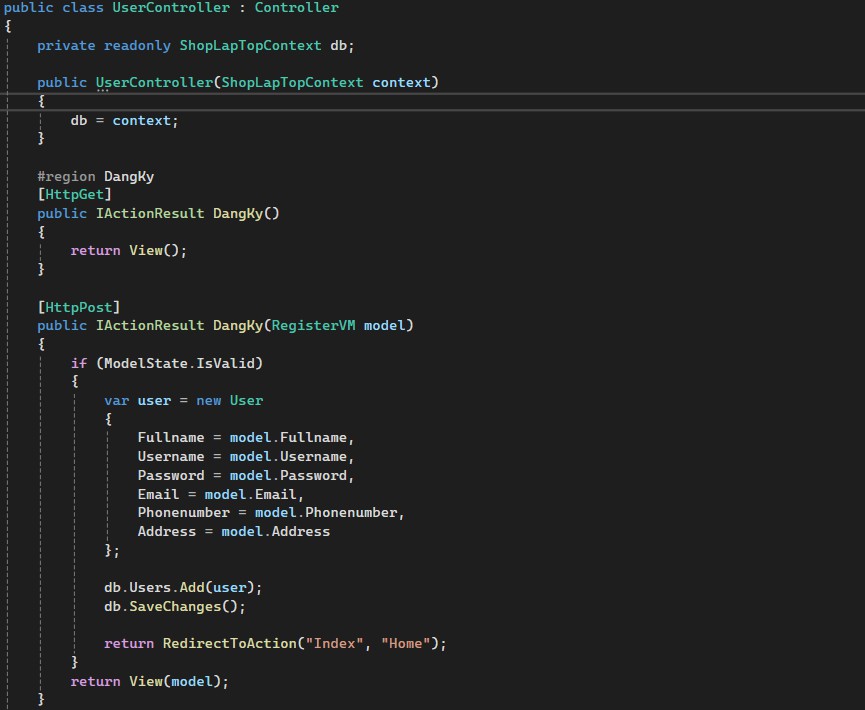
# CHƯƠNG 5. CÁC CHỨC NĂNG CỦA WEBSITE

## Chức Năng Đăng Ký

Chức năng đăng ký cho phép người dùng tạo tài khoản mới trên website. Quy trình làm chức năng đăng ký bao gồm các bước sau:

### Bước 1: Tạo file UserController.cs

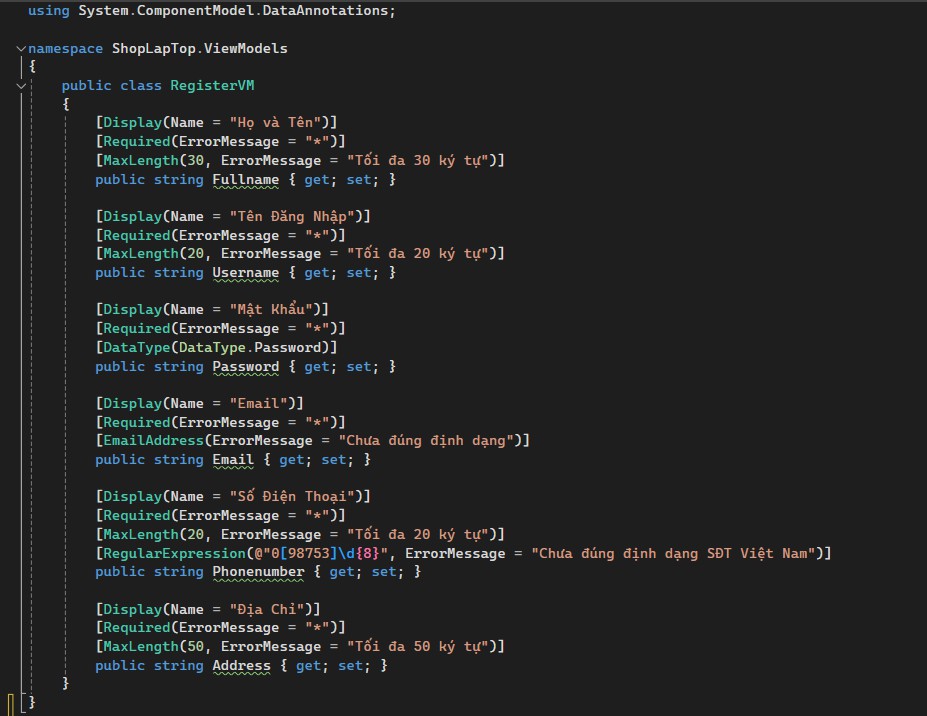
Trong file này, em sử dụng ASP.NET Core để xử lý các yêu cầu từ người dùng, bao gồm cả việc xác thực và xác autorization.



Hình 2: Đăng ký

### Bước 2: Tạo 1 ViewModels LoginVM.cs

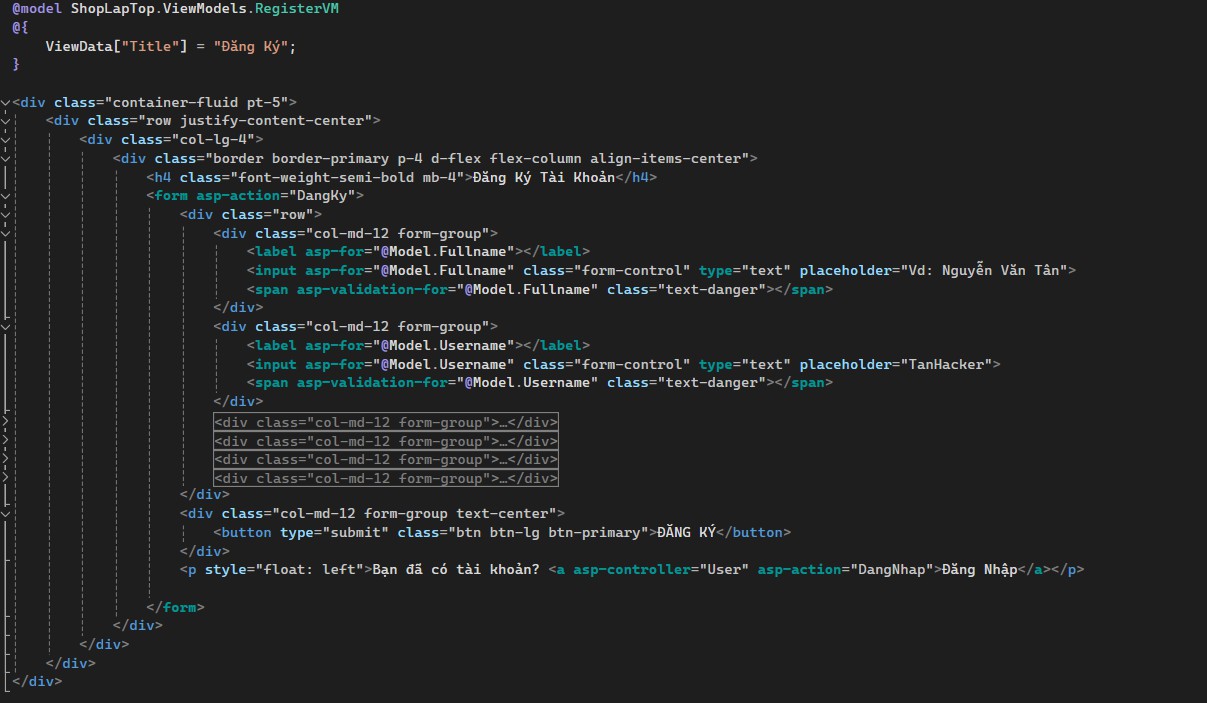
ViewModel này định nghĩa các thuộc tính cần thiết để nhận thông tin đăng nhập từ người dùng.



Hình 3: Định nghĩa đăng nhập

Bước 3: Tạo View DangKy.cshtml

View này hiển thị giao diện đăng ký cho người dùng, bao gồm các trường nhập thông tin như Họ và tên, Tên đăng nhập, Mật khẩu, Email, Số điện thoại, và Địa chỉ. Người dùng sau đó điền thông tin vào các trường và gửi yêu cầu đến server để đăng ký tài khoản.



Hình 4: View Đăng ký

Chức năng đăng ký được xây dựng trên nền tảng ASP.NET Core, sử dụng các công cụ như Razor Pages, ViewModels, và Entity Framework Core để xử lý dữ liệu và giao diện. Quy trình đăng ký được thực hiện một cách an toàn và hiệu quả, đảm bảo tính toàn vẹn của dữ liệu người dùng và bảo mật thông tin cá nhân.

Việc tạo tài khoản mới cho phép người dùng truy cập vào các chức năng và tính năng khác của website, như xem sản phẩm, thêm vào giỏ hàng, và đặt hàng. Điều này giúp tăng trải nghiệm người dùng và tính tương tác của website, từ đó thúc đẩy doanh số bán hàng và tăng cường mối quan hệ khách hàng.



Hình 5: Giao diện đăng ký

## Chức Năng Đăng Nhập

Chức năng đăng nhập cho phép người dùng đăng nhập vào hệ thống bằng tài khoản đã đăng ký trước đó. Quy trình đăng nhập bao gồm các bước sau:

### Bước 1: Tạo file UserController.cs

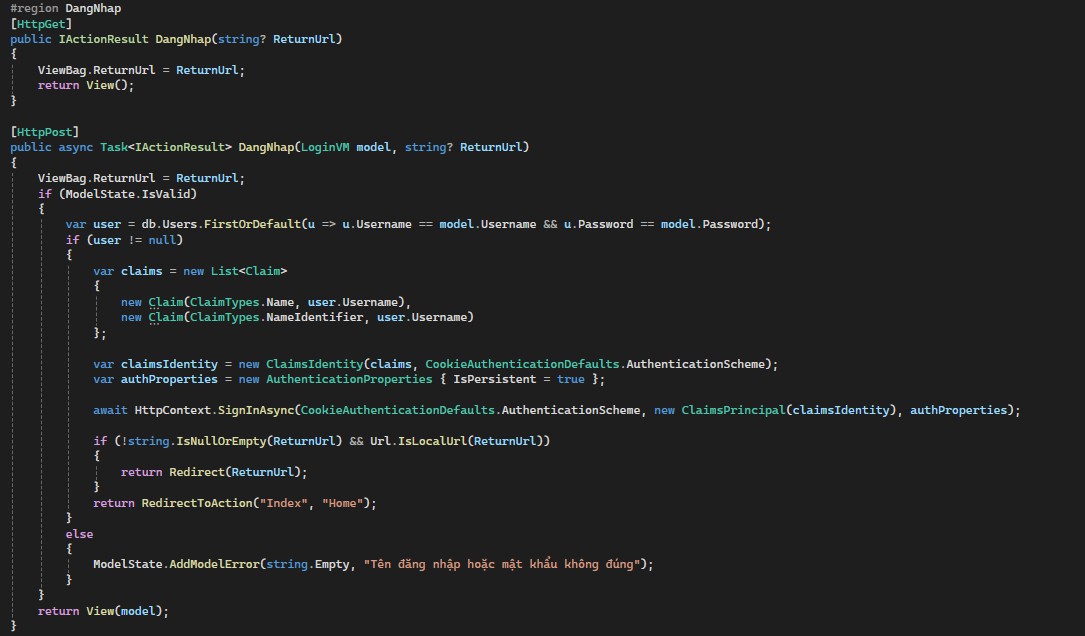
Trong file này, em đã xây dựng các hàm xử lý việc đăng nhập và đăng ký người dùng. Hàm DangNhap và DangNhap POST được sử dụng để xử lý yêu cầu đăng nhập từ phía người dùng. Nếu thông tin đăng nhập hợp lệ, người dùng sẽ được chuyển hướng đến trang chủ của website.



Hình 6: Định nghĩa đăng nhập

### Bước 2: Tạo 1 ViewModels LoginVM.cs

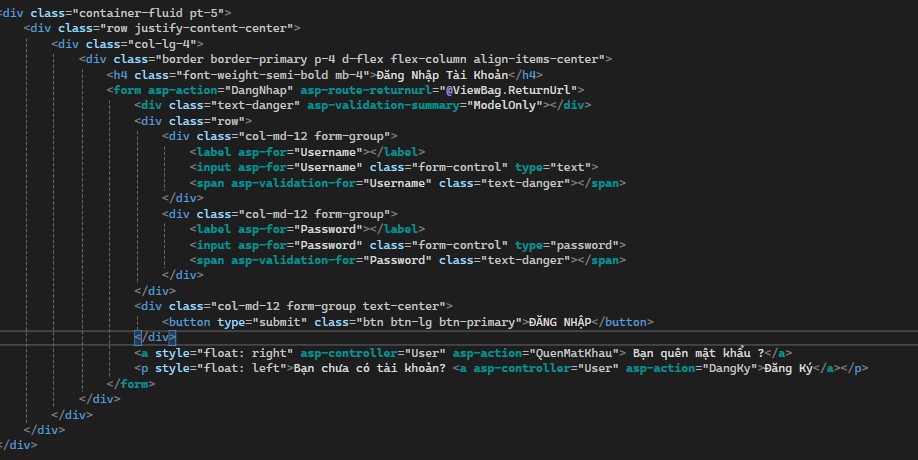
ViewModel này định nghĩa các thuộc tính cần thiết để nhận thông tin đăng nhập từ người dùng, bao gồm Tên đăng nhập và Mật khẩu.



Hình 7: Tạo View Model đăng nhập

### Bước 3: Tạo View DangNhap.cshtml

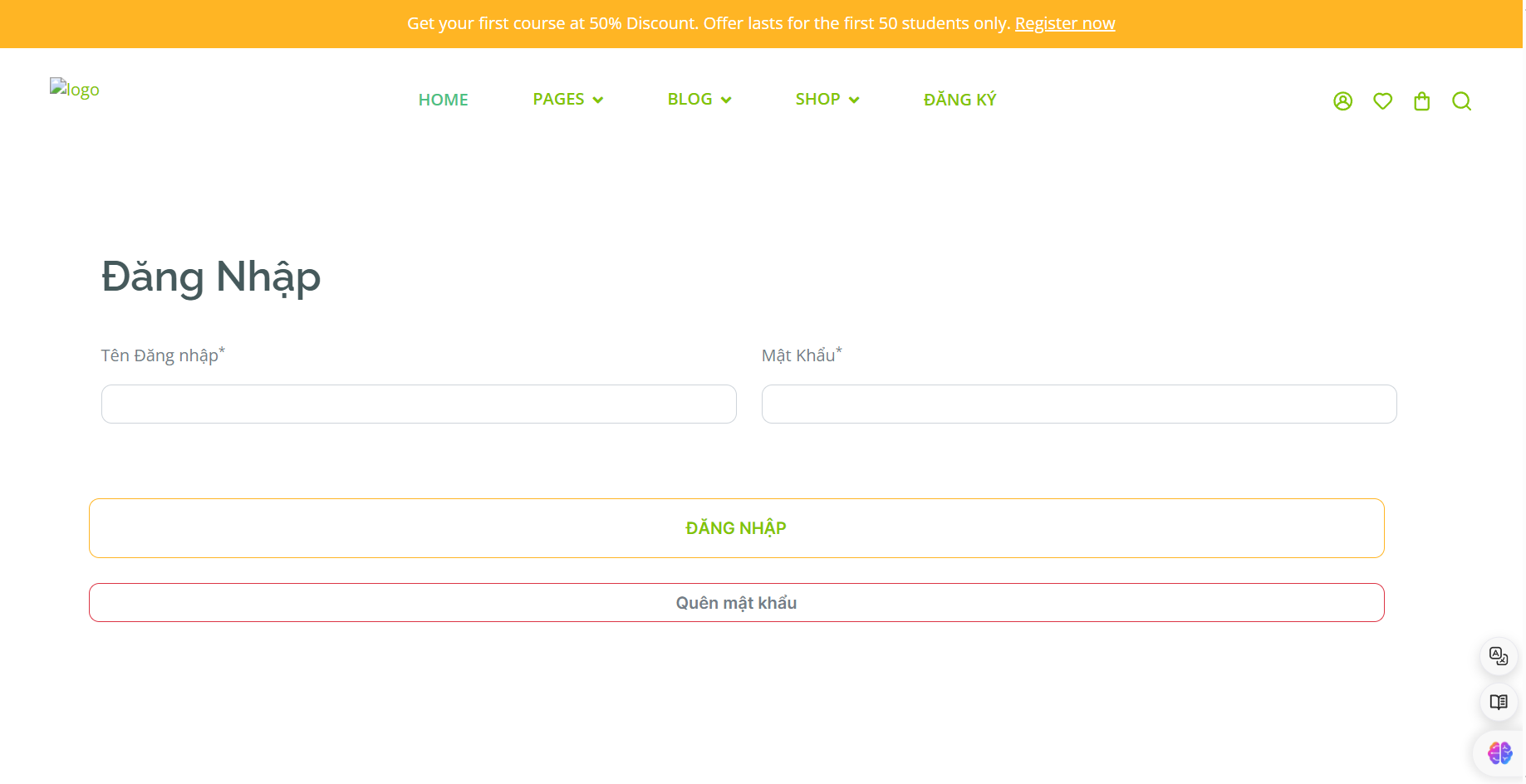
View này hiển thị giao diện đăng nhập cho người dùng, bao gồm các trường nhập thông tin như Tên đăng nhập và Mật khẩu. Người dùng sau đó điền thông tin vào các trường và gửi yêu cầu đến server để đăng nhập.



Hình 8: Tạo View đăng nhập

Chức năng đăng nhập được triển khai một cách an toàn và linh hoạt, sử dụng các công nghệ như ASP.NET Core và Entity Framework Core để xử lý dữ liệu và xác thực người dùng. Quy trình đăng nhập đảm bảo tính bảo mật của thông tin cá nhân và giúp người dùng trải nghiệm dịch vụ một cách thuận tiện và dễ dàng.

Việc đăng nhập vào hệ thống cho phép người dùng truy cập vào các chức năng và tính năng của website, như xem thông tin sản phẩm, thêm vào giỏ hàng, và thực hiện đơn hàng. Điều này giúp tăng cường trải nghiệm người dùng và tạo ra một môi trường mua sắm trực tuyến an toàn và tiện lợi.



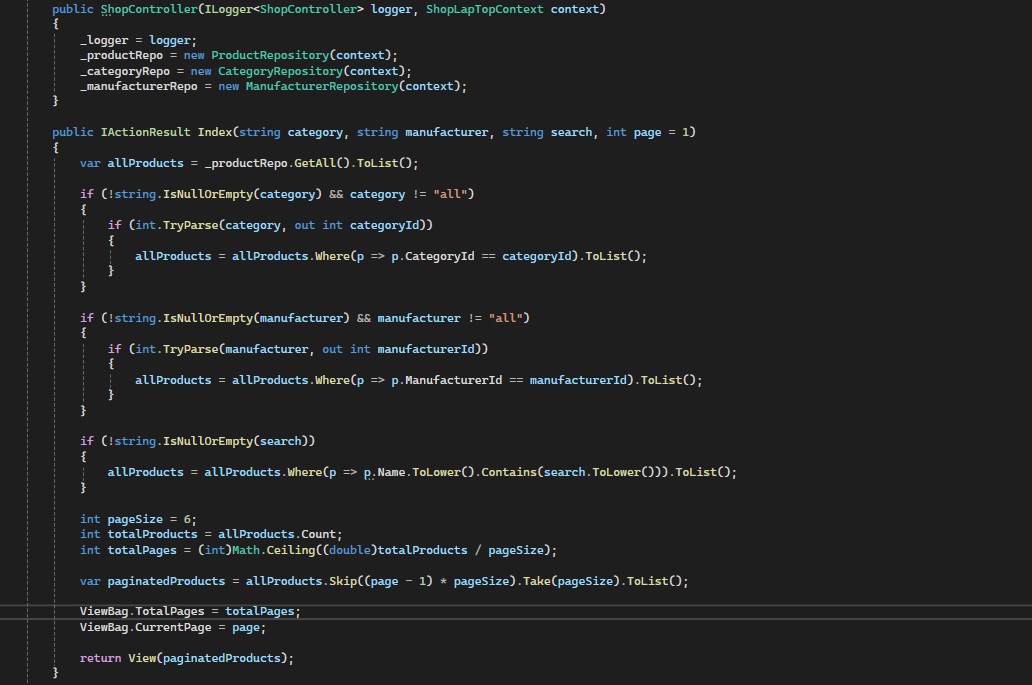
Hình 9: Giao diện đăng nhập

## Chức Năng Lọc Sản Phẩm Theo Loại Và Theo Hãng

Trong website bán laptop, người dùng có thể lọc sản phẩm theo loại và hãng để tìm kiếm những sản phẩm phù hợp với nhu cầu của mình. Quy trình này được thực hiện thông qua các menu loại và nhà sản xuất.

### Bước 1: Tạo Controller ShopController.cs

Trong controller này, hai phương thức được xây dựng là Index và Detail. Trong phương thức Index, người dùng có thể lọc sản phẩm theo loại và hãng thông qua các tham số truyền vào. Danh sách sản phẩm sau đó được hiển thị dựa trên các tiêu chí lọc này.

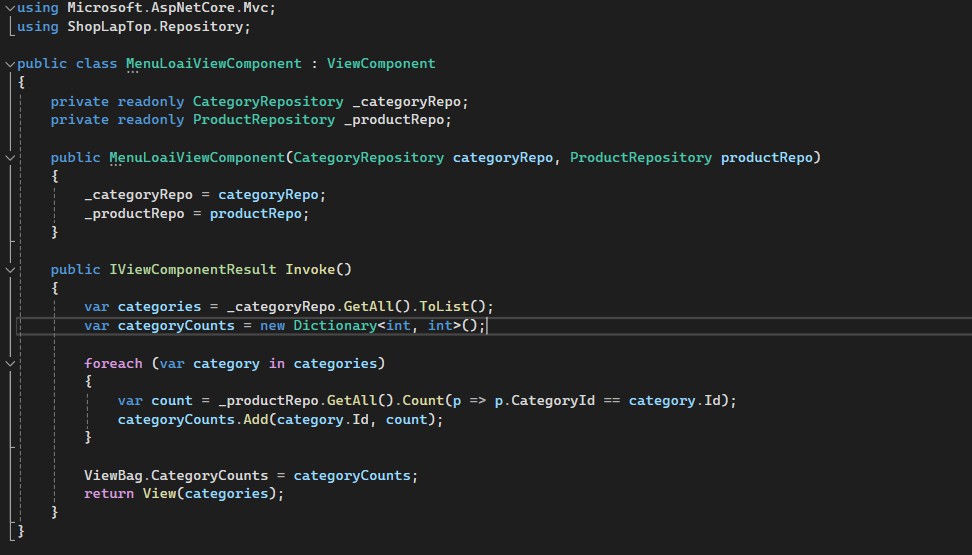


Hình 10: Tạo Controller

Bước 2 : Tạo Menu View Component

* “MenuLoaiViewComponent.cs”
* “MenuNhaSanXuatViewComponent.cs”

Hai View Component này được sử dụng để tạo menu lọc sản phẩm theo loại và nhà sản xuất. Trong đó, số lượng sản phẩm tương ứng với mỗi loại và nhà sản xuất được đếm và hiển thị.

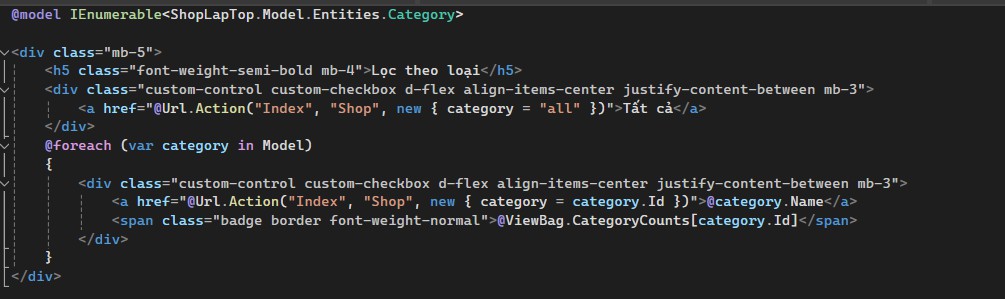


Hình 11: MenuLoai View

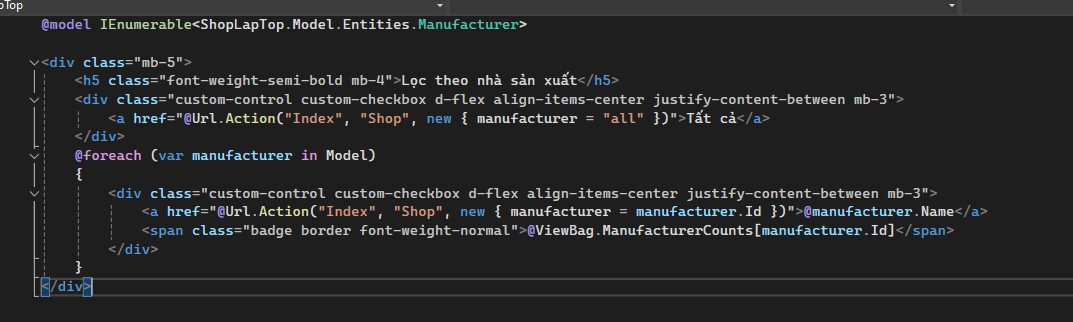
### Bước 3 : Tạo View

* + “MenuLoai/Default.cshtml”
  + “MenuNhaSanXuat/Default.cshtml”

Trong các file này, danh sách loại sản phẩm và nhà sản xuất được hiển thị dưới dạng các liên kết. Khi người dùng nhấp vào một loại hoặc hãng cụ thể, họ sẽ được chuyển hướng đến trang Index của ShopController với tham số tương ứng.



Hình 12: MenuLoai/Default



Hình 13:MenuKhoahoc/Default

Chức năng lọc sản phẩm theo loại và nhà sản xuất giúp người dùng dễ dàng tìm kiếm và chọn lựa những sản phẩm mà họ quan tâm. Quy trình này tạo ra một trải nghiệm mua sắm thuận tiện và hiệu quả cho người dùng, từ đó tăng cường sự hài lòng và trải nghiệm trên website.



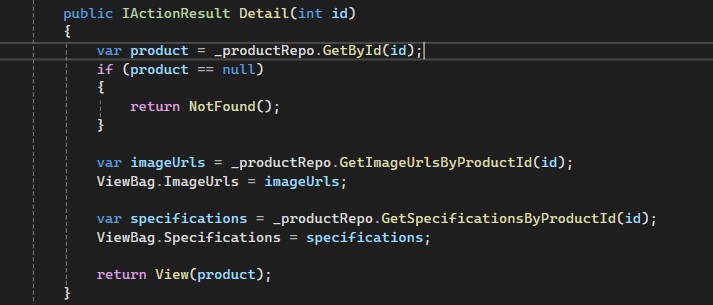
Hình 14: Giao diện loại các khóa học

## Chức Năng Xem Chi Tiết Sản Phẩm

Trong website bán laptop, chức năng xem chi tiết sản phẩm cho phép người dùng xem thông tin chi tiết về một sản phẩm cụ thể, bao gồm hình ảnh, mô tả, thông số kỹ thuật và giá cả.

### Bước 1: Tạo Controller : ShopController

Như Trong ShopController ở trên, phương thức Detail được tạo ra để xử lý yêu cầu xem chi tiết sản phẩm. Nó nhận ID của sản phẩm làm tham số và trả về view chi tiết tương ứng nếu sản phẩm tồn tại. Nếu không, nó trả về mã trạng thái 404 (Not Found).

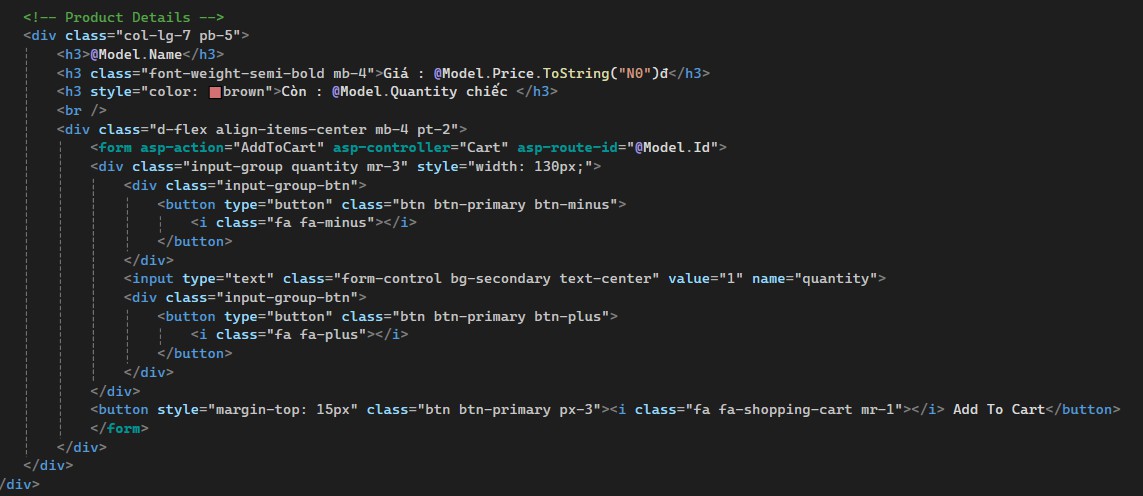


Hình 15: Controller chi tiết sản phẩm

### Bước 2: Tạo View : Detail.cshtml

Trong view này, thông tin chi tiết của sản phẩm được hiển thị bao gồm tên, giá, số lượng còn lại, mô tả, thông số kỹ thuật và một số ảnh của sản phẩm. Nó cũng bao gồm một tab để người dùng chuyển đến các phần khác nhau của sản phẩm như mô tả, thông tin kỹ thuật và đánh giá.

* Thông tin Chi Tiết Sản Phẩm: Người dùng có thể dễ dàng xem thông tin chi tiết về sản phẩm một cách rõ ràng và chi tiết.
* Hình Ảnh Sản Phẩm: Hình ảnh của sản phẩm được hiển thị trong một carousel, cho phép người dùng xem sản phẩm từ nhiều góc độ khác nhau.
* Tab Navigation: Giao diện tab cho phép người dùng dễ dàng chuyển đổi giữa các phần thông tin về sản phẩm.

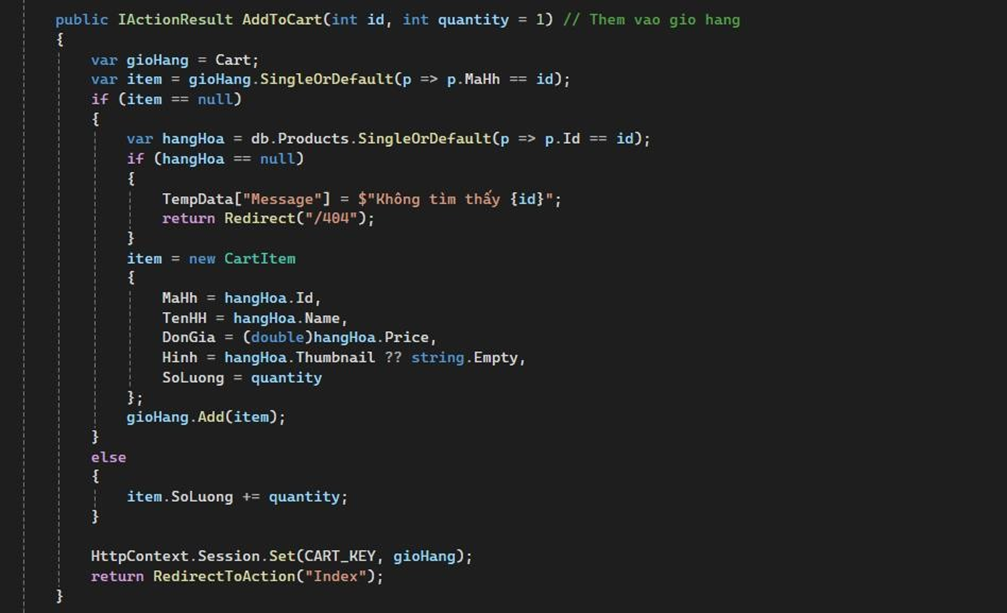


Hình 16: Tạo View

## Chức Năng Thêm Sản Phẩm Vào Giỏ Hàng

### Bước 1 : Tạo Controller : CartController

* + - Điều khiển các hành động liên quan đến giỏ hàng như thêm sản phẩm, xóa sản phẩm, giảm số lượng sản phẩm và thanh toán.



Hình 17: Tạo Controller

* + - Sử dụng các phương thức như AddToCart, RemoveFromCart, DecreaseQuantity để thực hiện các hành động tương ứng.
    - AddToCart Action:

+ Được gọi khi người dùng muốn thêm một sản phẩm vào giỏ hàng.

+ Nhận vào hai tham số: id của sản phẩm cần thêm và quantity (số lượng mặc định là 1 nếu không được cung cấp).

+ Kiểm tra xem sản phẩm đã tồn tại trong giỏ hàng chưa.

+ Nếu sản phẩm chưa tồn tại trong giỏ hàng:

+ Tìm sản phẩm trong cơ sở dữ liệu bằng id.

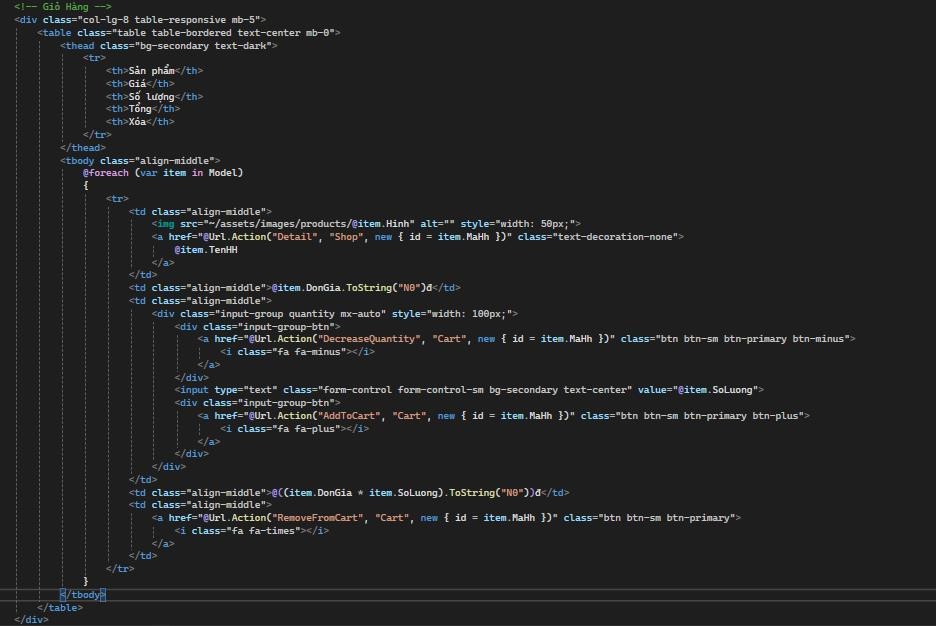
+ Tạo một đối tượng CartItem mới đại diện cho sản phẩm và thêm vào giỏ hàng.

+ Nếu sản phẩm đã tồn tại trong giỏ hàng, chỉ tăng số lượng của sản phẩm đó.

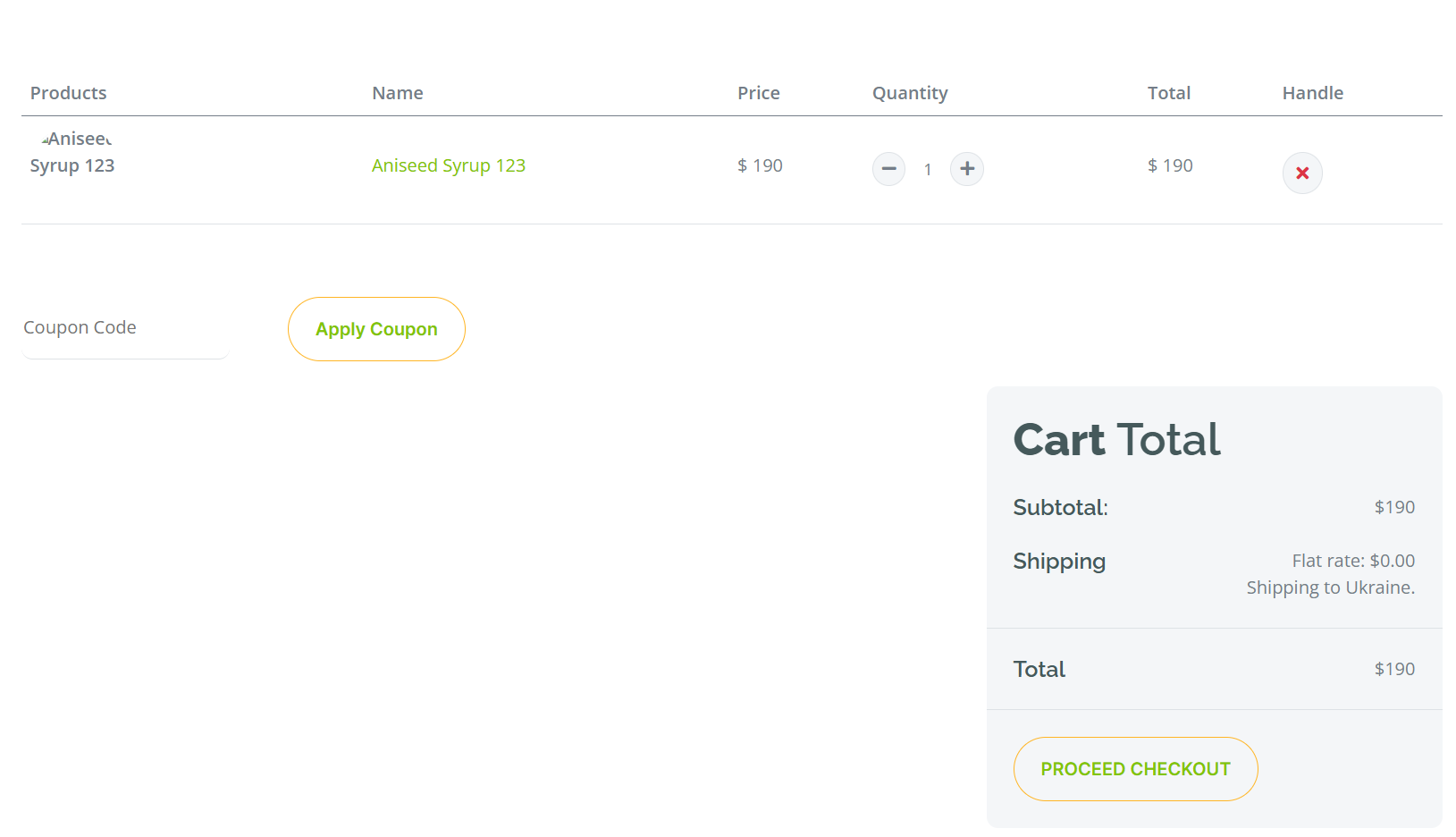
+ Lưu giỏ hàng vào Session và sau đó chuyển hướng người dùng đến trang Index của giỏ hàng.

### Bước 2 : Tạo View : Index.cshtml

* + - Hiển thị danh sách sản phẩm trong giỏ hàng và cho phép người dùng tương tác với chúng.
    - Sử dụng vòng lặp foreach để liệt kê các sản phẩm và các nút điều khiển tương ứng (tăng số lượng, giảm số lượng, xóa).

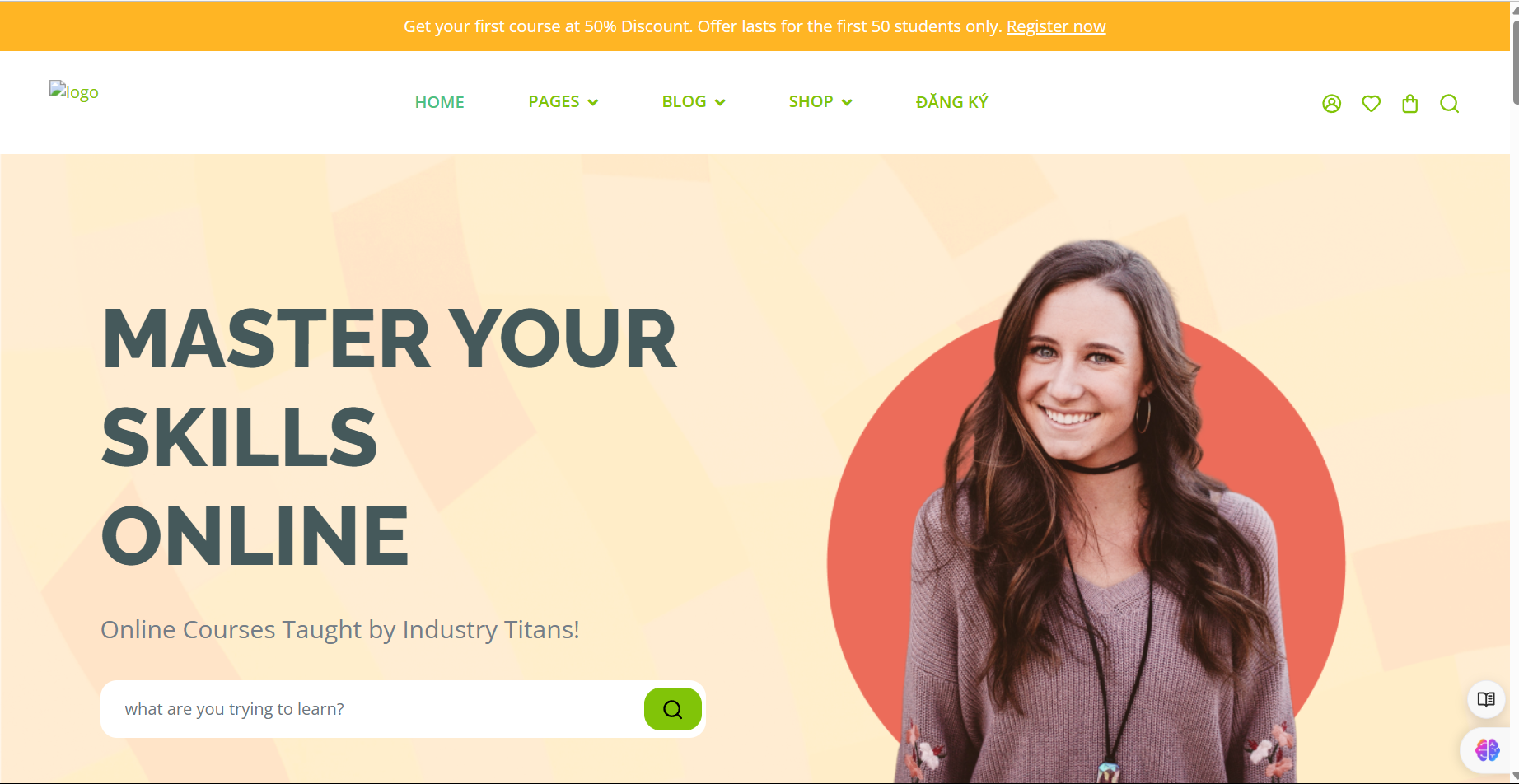


Hình 18: Tạo View

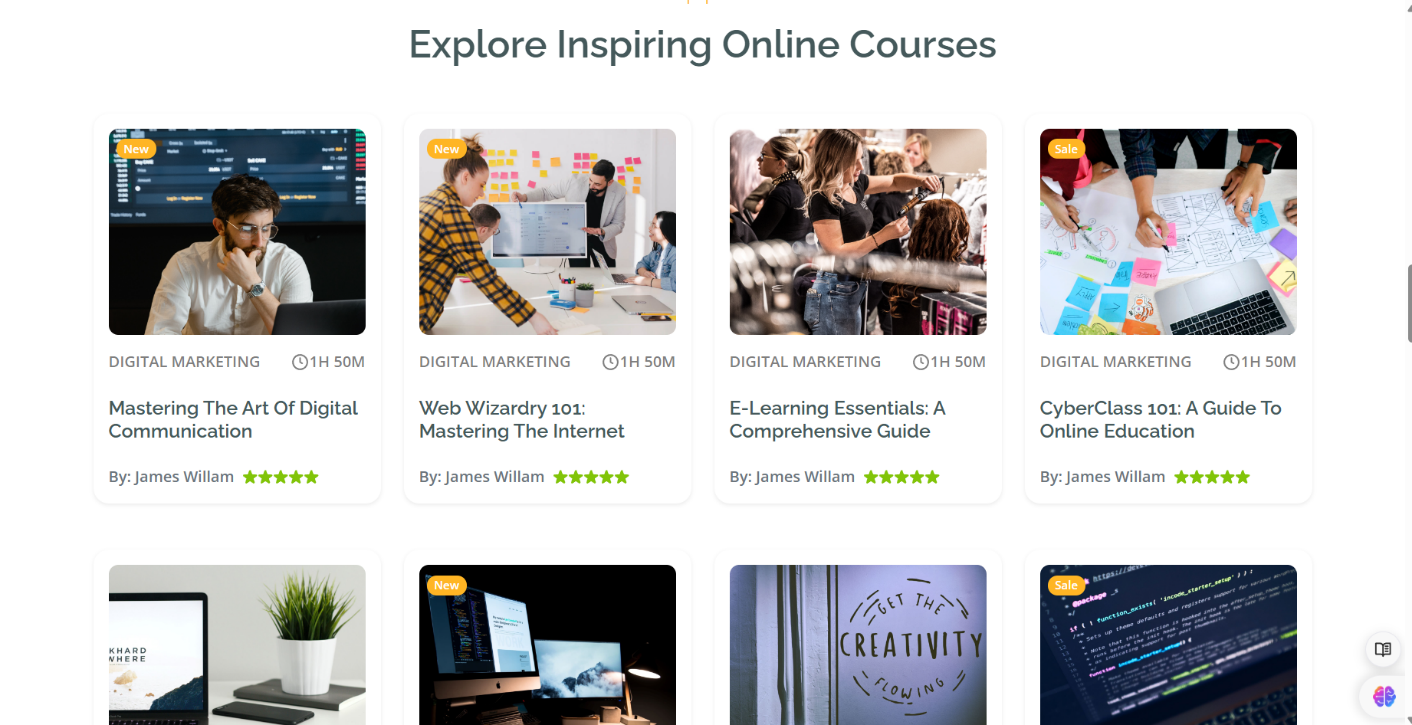


Hình 19: Giao diện giỏ hàng

# CHƯƠNG 6. GIAO DIỆN NGƯỜI DÙNG

****

Hình 20: View Home



Hình 21: Bảng thông tin

## Phần HOME ( Trang chủ )

Về phần này em sẽ phân tích chủ yếu về phía Back-End nhiều hơn. Trang chủ của website là điểm đầu tiên mà người dùng thường truy cập khi truy cập vào trang web.

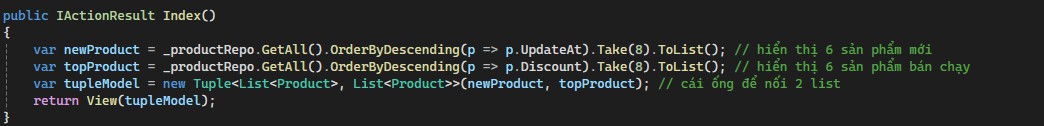
* + - Chức Năng Hiển Thị Sản Phẩm Mới và Sản Phẩm Bán Chạy

Trong phương thức Index(), sử dụng đối tượng ProductRepository để lấy danh sách sản phẩm mới nhất và sản phẩm bán chạy nhất.

Sản phẩm mới được lấy bằng cách sắp xếp danh sách tất cả sản phẩm theo thời gian cập nhật, sau đó lấy 8 sản phẩm đầu tiên.

Sản phẩm bán chạy được lấy bằng cách sắp xếp danh sách tất cả sản phẩm theo mức giảm giá, sau đó lấy 8 sản phẩm đầu tiên.

Sử dụng Tuple để nối hai danh sách sản phẩm này lại với nhau trước khi truyền vào view.



Hình 22: Sản phẩm bán chạy

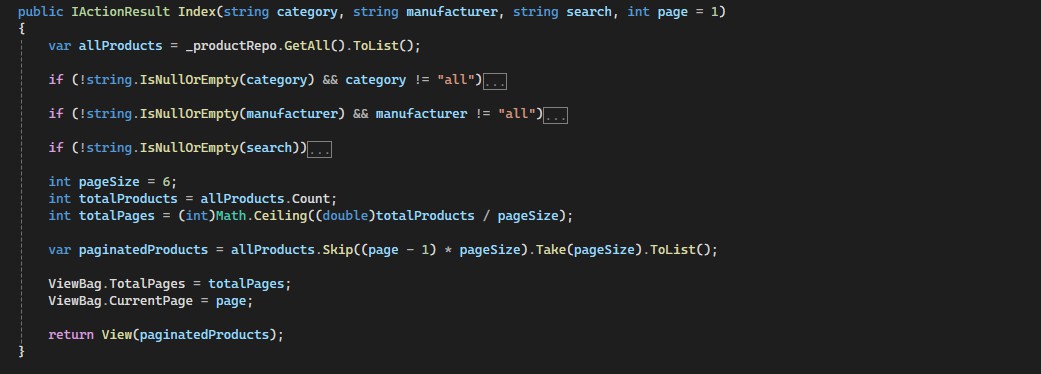
## Phần SHOP ( Hiển thị các sản phẩm )

Phần Shop của website là nơi mà người dùng có thể duyệt và tìm kiếm các sản phẩm

* + - Phương Thức Index()

Phương thức này trả về view chứa danh sách sản phẩm dựa trên các tham số như loại, nhà sản xuất, và từ khóa tìm kiếm.

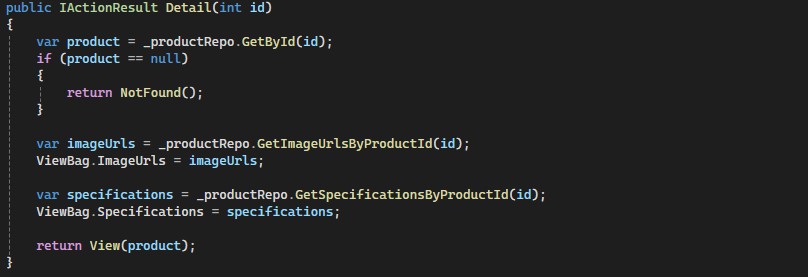
Dùng các tham số category, manufacturer, và search để lọc sản phẩm. Sử dụng page để phân trang, hiển thị một số sản phẩm trên mỗi trang.



Hình 23: Hiển thị các sản phẩm

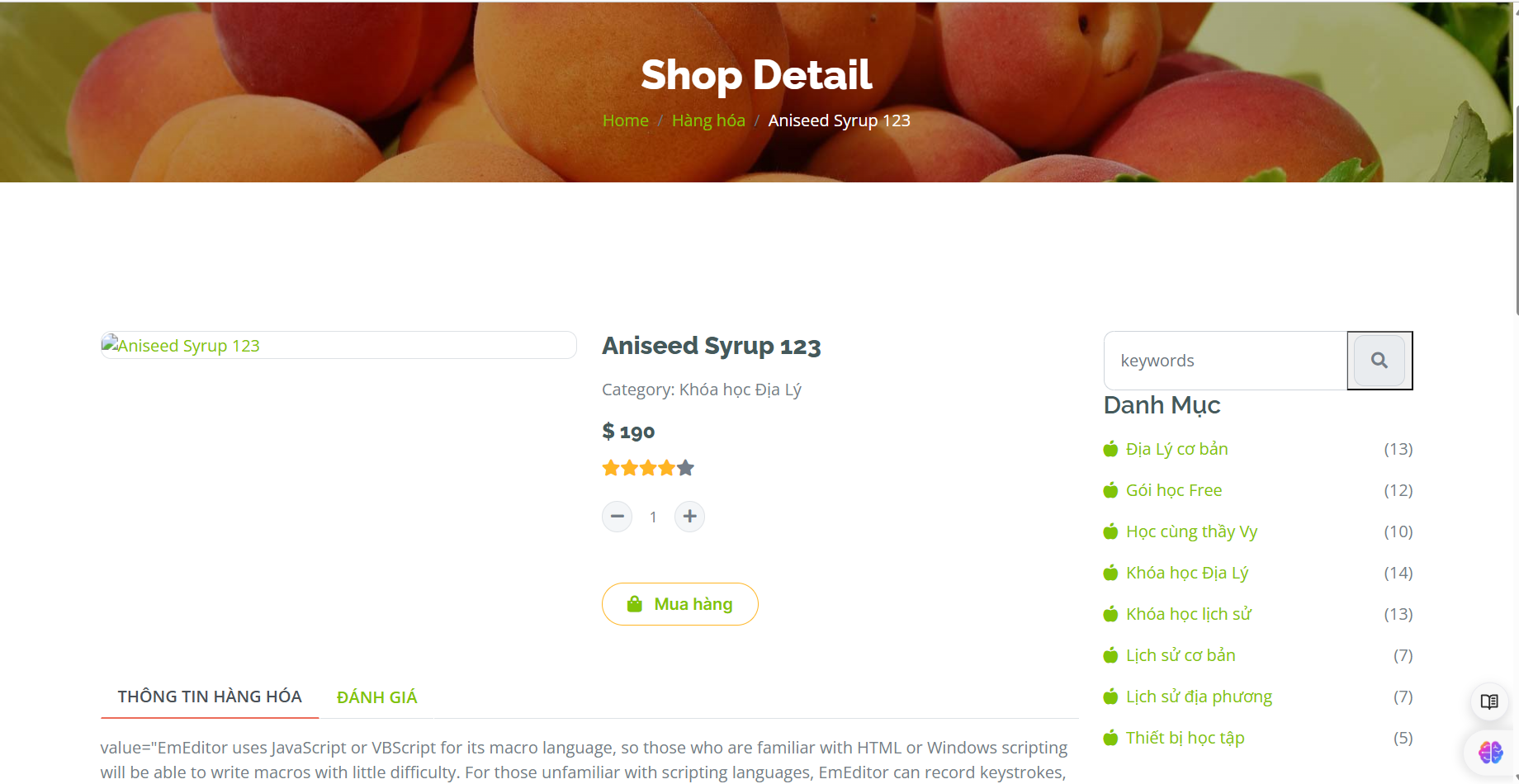
* + - Phương Thức Detail()

Phương thức này trả về view hiển thị chi tiết sản phẩm dựa trên id của sản phẩm. Hiển thị thông tin chi tiết của sản phẩm như hình ảnh, mô tả, thông số kỹ thuật.

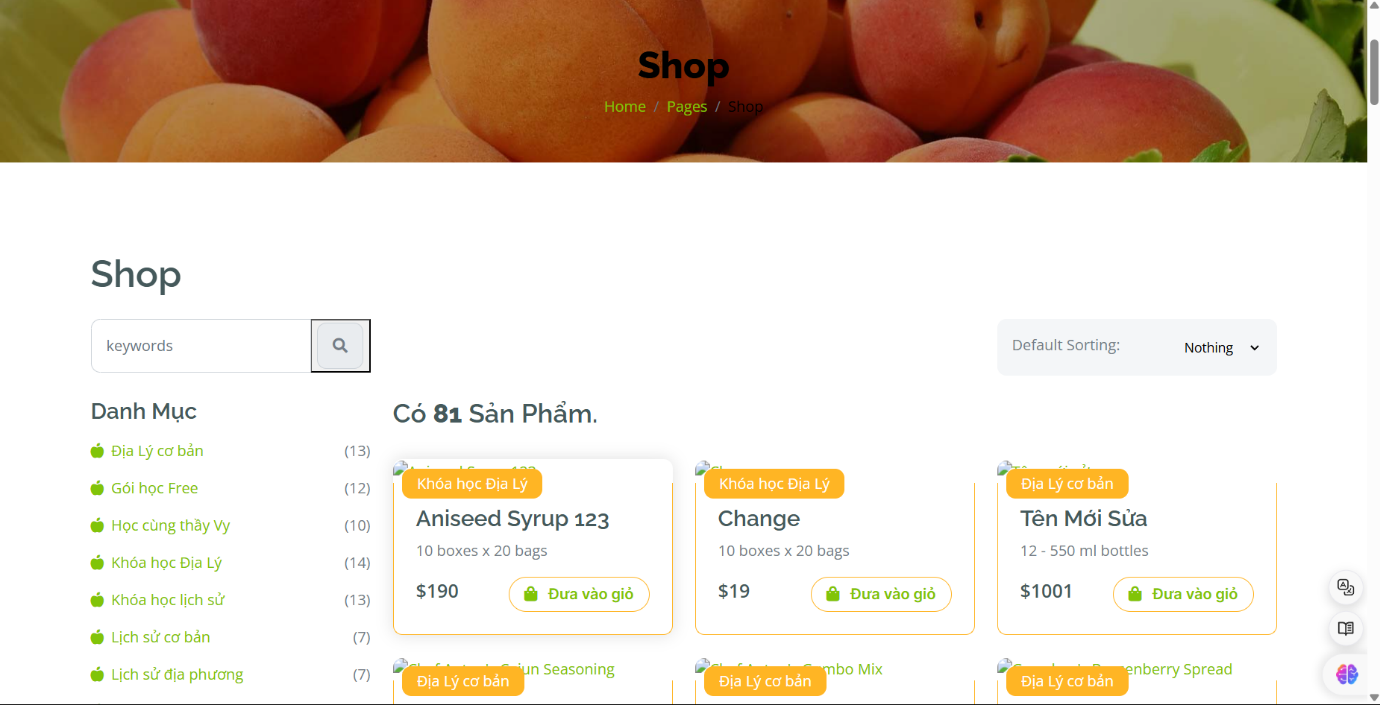


Hình 24: Phương thức Detail

Phần Shop của website cung cấp một giao diện dễ sử dụng và linh hoạt, giúp người dùng dễ dàng tìm kiếm và mua sắm sản phẩm mong muốn. Tuy nhiên, cần tiếp tục cải thiện và bổ sung tính năng để nâng cao trải nghiệm của người dùng.



Hình 25Giao diện sản phẩm



Hình 26: Giao diện shop

## Phần CART ( Giỏ hàng )

Phần CART của website là nơi mà người dùng có thể quản lý và kiểm soát các sản phẩm đã được thêm vào giỏ hàng và thanh toán.

* + - Cấu Trúc Dữ Liệu

Sử dụng session để lưu trữ thông tin giỏ hàng của người dùng.

Mỗi mục trong giỏ hàng được biểu diễn bằng đối tượng CartItem, bao gồm các thuộc tính như MaHh (mã hàng hóa), TenHH (tên hàng hóa), DonGia (đơn giá), Hinh (hình ảnh), và SoLuong (số lượng).

- Các Phương Thức Chính

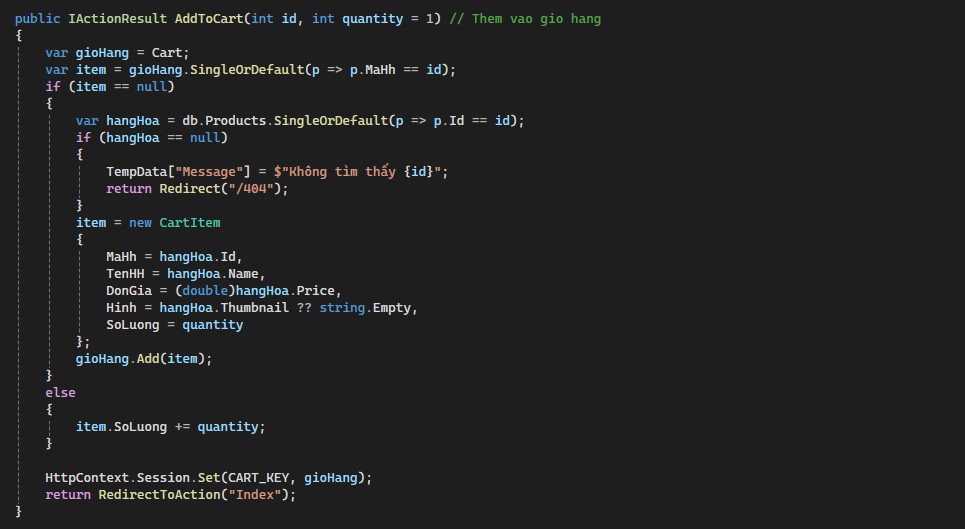
Index(): Hiển thị giỏ hàng của người dùng.

AddToCart(int id, int quantity): Thêm sản phẩm vào giỏ hàng hoặc cập nhật số lượng nếu sản phẩm đã tồn tại trong giỏ hàng.

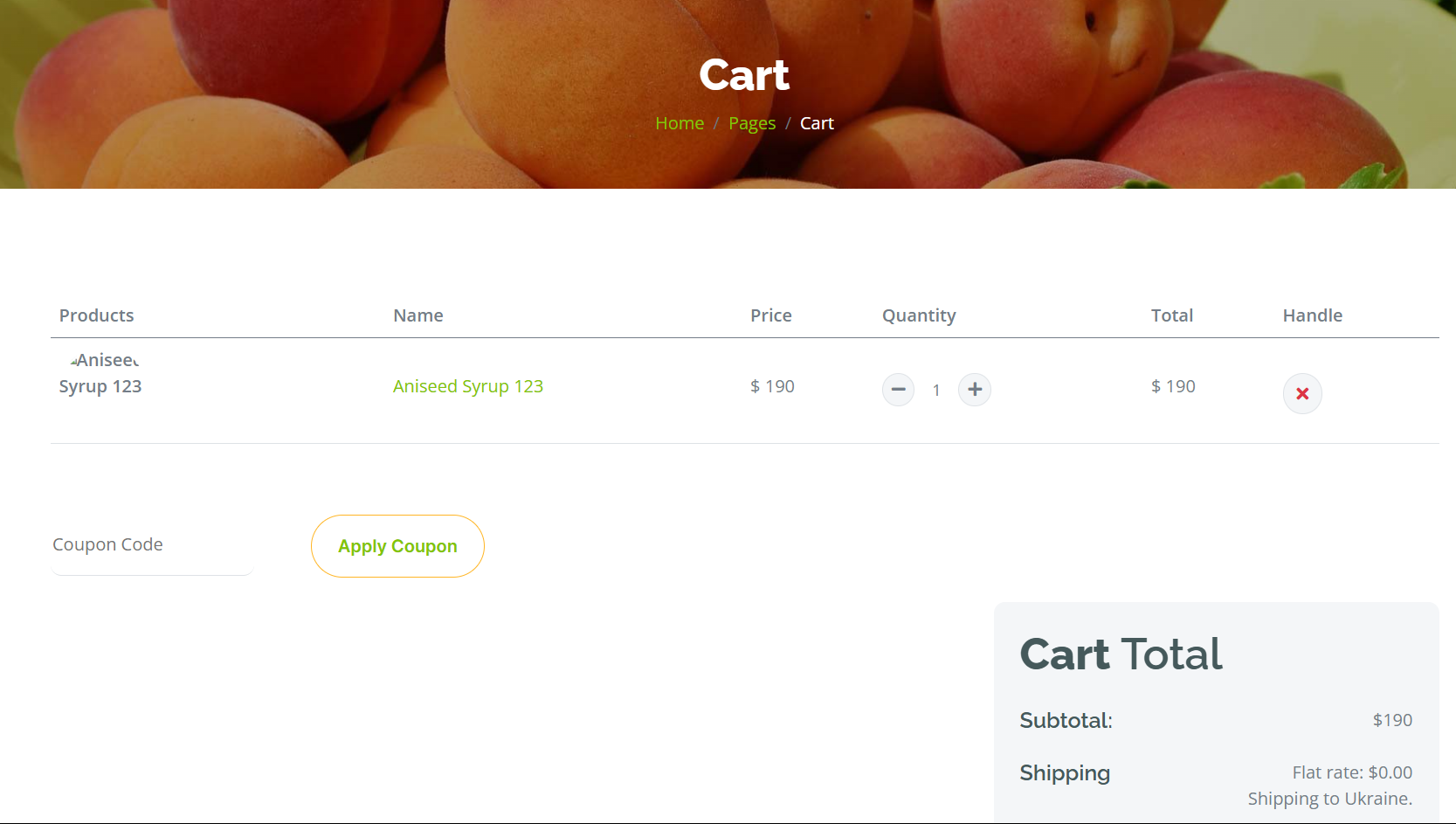
RemoveFromCart(int id): Xóa sản phẩm khỏi giỏ hàng.

DecreaseQuantity(int id): Giảm số lượng của một sản phẩm trong giỏ hàng.

CartSummary(): Hiển thị tổng số sản phẩm trong giỏ hàng, thường được sử dụng để cập nhật biểu tượng giỏ hàng trên giao diện.

Phần CART của website cung cấp một cách tiếp cận linh hoạt và tiện lợi cho người dùng trong việc quản lý và thanh toán các sản phẩm trong giỏ hàng. Tuy nhiên, cần tiếp tục cải thiện để nâng cao trải nghiệm của người dùng và tính bảo mật của hệ thống.

Hình 27: Thêm vào giỏ hàng



Hình 28: Giao diện giỏ hàng

# KẾT LUẬN

## Ưu điểm:

ASP.NET Core Back-End mang lại nhiều lợi ích cho việc phát triển website dạy học lịch sử - địa lý địa phương. Với sự linh hoạt và hiệu suất cao, nền tảng này cho phép xây dựng các ứng dụng web mạnh mẽ, đồng thời tối ưu hóa hiệu suất hoạt động. Khả năng đa nhiệm và tương thích đa nền tảng của ASP.NET Core cũng là một ưu điểm lớn, giúp đảm bảo rằng website có thể hoạt động mượt mà trên nhiều thiết bị và trình duyệt khác nhau..

## Nhược điểm:

Mặc dù ASP.NET Core Back-End mang lại nhiều ưu điểm, nhưng cũng tồn tại một số hạn chế. Trong đó, việc học và sử dụng ASP.NET Core có thể đòi hỏi một quá trình học tập và thích nghi, đặc biệt đối với những nhà phát triển mới. Ngoài ra, việc triển khai và quản lý cũng có thể đòi hỏi một số kỹ năng kỹ thuật cao.

## Hướng phát triển:

Để phát triển chủ đề website dạy học lịch sử - địa lý địa phương sử dụng ASP.NET Core Back-End, có thể tập trung vào việc tối ưu hóa trải nghiệm người dùng, cải thiện tính tương tác và tích hợp các tính năng mới như hệ thống đánh giá, tương tác xã hội, và cải thiện tính bảo mật. Ngoài ra, việc nâng cao hiệu suất và tối ưu hóa cơ sở hạ tầng cũng là những điểm cần được chú ý để đảm bảo website hoạt động ổn định và mượt mà, phục vụ tốt cho mục đích giáo dục trong môi trường học tập ngày càng phát triển.

## DANH MỤC TÀI LIỆU THAM KHẢO

1. Entity Framework. (2021). Entity Framework Core Documentation. Retrieved from https://docs.microsoft.com/en-us/ef/core/
2. W3Schools. (2021). HTML Tutorial. Retrieved from htt[ps://www.w3schools.com/html/](http://www.w3schools.com/html/)
3. W3Schools. (2021). CSS Tutorial. Retrieved from https:[//www.w3schools.com/css/](http://www.w3schools.com/css/)
4. W3Schools. (2021). JavaScript Tutorial. Retrieved from https://[www.w3schools.com/js/](http://www.w3schools.com/js/)
5. W3Schools. (2021). SQL Tutorial. Retrieved from https:[//www.w3schools.com/sql/](http://www.w3schools.com/sql/)
6. Nguyen, A. (2024). Thiết kế website nhà hàng hiệu quả. Hanoi: Hanoi Publishing House.
7. Tran, B. (2024). Lập trình web với ASP.NET Core và SQL Server. Ho Chi Minh City: Saigon Tech Press.
8. Le, C. (2024). Cơ sở dữ liệu cho các ứng dụng web. Da Nang: Danang University Press.
9. Chat GPT
10. Microsoft. https://dotnet.microsoft.com/en-us/apps/aspnet