

**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 Tài liệu học tập + báo cáo + đồ án**

**GVHD :Nguyễn Đức Tấn
SVTH :Nguyễn Ngọc Vỹ
Mã số SV : 2301010045
Khóa học : 2024 - 2025**

Đà Lạt, tháng 6- 2025

PHẦN NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm:

Ngày ... tháng ... năm...

Ký và ghi rõ họ tên

MỤC LỤC

1. Chương 1 Tìm hiểu về lập trình Web MVC (Cơ sở lý thuyết).....	7
1.1 Khái niệm	7
1.2 Đặc điểm.....	8
1.3 Nguyên lý hoạt động	9
1.4 Công nghệ triển khai ASP.NET CoreGiới thiệu về ASP.NET Core	11
Chương 2 Xây dựng ứng dụng Web (Triển khai thực tế).....	16
2.1 Phát biểu bài toán ứng dụng	16
2.1.1 Bài toán đặt ra	16
2.1.2 Mục tiêu của ứng dụng web	16
2.1.3 Vai trò và giá trị mang lại	16
2.1.4. ý nghĩa thực tiễn	17
2.2 Phân tích yêu cầu của ứng dụng	18
2.2.1. Mô tả chức năng ứng dụng	18
2.2.2 Yêu cầu phi chức năng (Non-Functional Requirements).....	19
2.2.3.Biểu đồ Use-Case	20
2.3 Thiết kế cơ sở dữ liệu:	22
2.4 Thiết kế các thành phần MVC:.....	30
2.4.1. Model.....	30
2.4.2.View	33
2.5 Triển khai và cài đặt	39
Chương 3 Kết quả chương trình	40
3.1 Người dùng(User)	40
3.2 Quản trị(Admin)	47
Chương 4 : Kết luận.....	50
4.1 Tổng kiến thức đạt được.....	50
4.2 Điểm tồn tại	51
4.3 Hướng mở rộng đề tài	51
Tài liệu tham khảo.....	52
PHỤ LỤC.....	53

DANH MỤC HÌNH

Hình ảnh	Mô tả	Trang
Hình 1.1	Mô tả nguyên lý hoạt động website	3
Hình 1.2	Nguyên lý hoạt động	10
Hình 2.1	Sơ đồ UseCase mô tả chức năng của admin và user	21
Hình 2.2	Sơ đồ mô tả tương tác giữa Người dùng(User) và quản trị (Admin)	22
Hình 2.3.1	Phần hệ thống người dùng (Identity)	23
Hình 2.3.2	Hình dữ liệu nghiệp vụ(domain data)	23
Hình 2.2.3	Mô hình Cơ sở dữ liệu	24
Hình 2.4.1	Mô tả chi tiết bảng 3	30
Hình 2.4.2	Mô tả chi tiết bảng 4	31
Hình 2.4.3	Mô tả chi tiết bảng 5	31
Hình 2.4.4	Mô tả chi tiết bảng 6	32
Hình 2.4.5	Mô tả chi tiết bảng 2	32
Hình 2.4.6	Mô tả chi tiết bảng 1	32
Hình 3.1	Đăng kí	40
Hình 3.2	Đăng nhập	41
Hình 3.3	Trang chủ	42,43
Hình 3.4	Chi tiết sản phẩm	44
Hình 3.5	Tham khảo	44
Hình 3.6	Chi tiết sản phẩm	45
Hình 3.7	Giỏ hàng trống	45

Hình 3.8	Giỏ hàng có sản phẩm	45
Hình 3.9	Sản phẩm	46
Hình 3.10	Liên hệ	46
Hình 3.11	Hồ sơ người dùng	47
Hình 3.12	Đổi mật khẩu	47
Hình 3.13	Bảng điều khiển	48
Hình 3.14	Quản lí danh mục	48
Hình 3.15	Quản lí sản phẩm	49
Hình 3.16	Quản lí người dùng	49
Hình 3.17	Quản lí người tác giả	50

LỜI NÓI ĐẦU

Trong thời đại công nghệ số bùng nổ 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 kho tài nguyên phong phú cho người dùng ở mọi lĩnh vực, đặc biệt là trong lĩnh vực giáo dục. Nhu cầu tìm kiếm, trao đổi các tài liệu học tập, báo cáo, đồ án... ngày càng gia tăng, nhất là đối với học sinh, sinh viên và các đối tượng nghiên cứu chuyên sâu.

Xuất phát từ thực tế đó, việc lựa chọn đề tài **“Xây dựng website thương mại điện tử kinh doanh sản phẩm Tài liệu học tập, Báo cáo và Đồ án”** nhằm phát triển một nền tảng trực tuyến hỗ trợ người dùng dễ dàng tra cứu, mua bán và chia sẻ các tài liệu học tập chất lượng. Website không chỉ tập trung vào việc cung cấp các chức năng quản lý sản phẩm, giỏ hàng và thanh toán, mà còn hướng đến trải nghiệm người dùng thân thiện, dễ sử dụng và đảm bảo tính bảo mật thông tin.

Hệ thống được xây dựng theo mô hình **MVC** kết hợp với phương pháp **Code First** trong **Entity Framework**, cho phép thiết kế và quản lý cơ sở dữ liệu một cách linh hoạt thông qua các lớp mô hình. Dự án sử dụng các công nghệ tiêu biểu như **ASP.NET MVC**, **Entity Framework (Code First)**, **HTML**, **CSS**, **JavaScript** và **SQL Server** để triển khai đầy đủ các chức năng cần thiết của một hệ thống thương mại điện tử hiện đại: từ đăng ký – đăng nhập, quản lý sản phẩm, giỏ hàng, thanh toán, tìm kiếm theo từ khóa hoặc danh mục, đến hệ thống phân quyền người dùng.

Báo cáo này sẽ trình bày chi tiết toàn bộ quy trình xây dựng hệ thống – từ phân tích yêu cầu, thiết kế mô hình dữ liệu, hiện thực hóa các chức năng chính, đến giai đoạn kiểm thử và hoàn thiện sản phẩm. Thông qua đề tài này, em mong muốn thể hiện được tư duy logic, khả năng lập trình hướng đối tượng, cùng kỹ năng thiết kế và triển khai phần mềm theo xu hướng công nghệ hiện nay.

Cuối cùng, em xin được bày tỏ lòng biết ơn sâu sắc đến **thầy Nguyễn Đức Tấn** – người đã tận tình hướng dẫn, hỗ trợ và truyền cảm hứng cho em trong suốt quá trình thực hiện đề tài. Sự đồng hành của thầy là động lực to lớn giúp em hoàn thành tốt đồ án này.

Em xin chân thành cảm ơn!

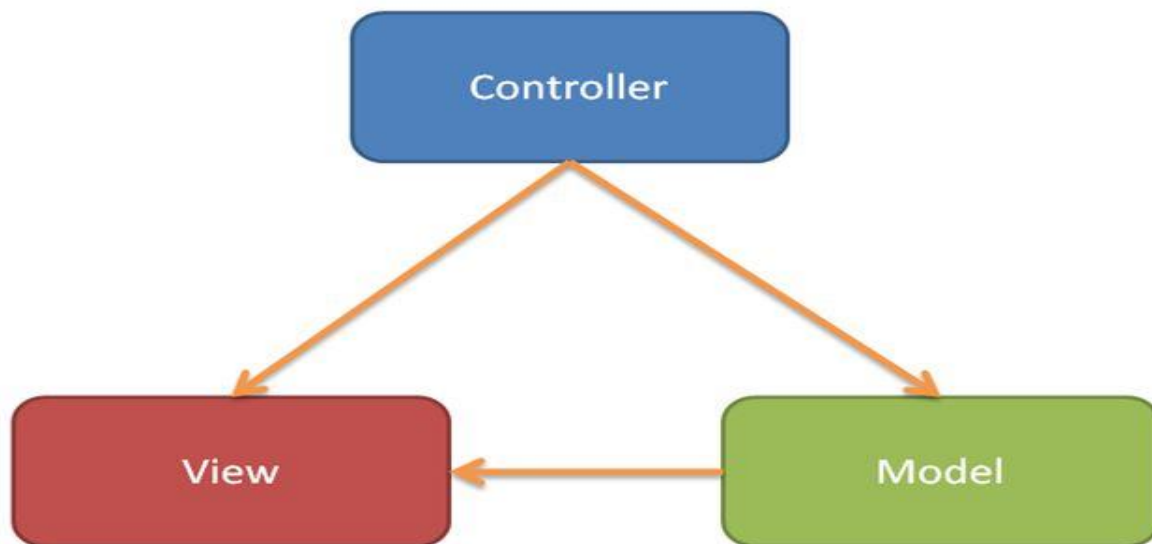
1. Chương 1 Tìm hiểu về lập trình Web MVC (Cơ sở lý thuyết)

1.1 Khái niệm

Mô hình MVC (Model - View - Controller) là một kiến trúc phần mềm được sử dụng phổ biến trong phát triển ứng dụng web và desktop hiện đại, đặc biệt phù hợp với những dự án có quy mô lớn, yêu cầu khả năng bảo trì, mở rộng và phân chia công việc rõ ràng giữa các nhóm phát triển. MVC giúp tổ chức mã nguồn khoa học và logic hơn thông qua việc chia hệ thống thành ba phần riêng biệt, mỗi phần đảm nhiệm một vai trò khác nhau:

- **Model (Dữ liệu và logic nghiệp vụ):** Là nơi định nghĩa các lớp đối tượng, mối quan hệ và các thao tác liên quan đến dữ liệu (truy vấn, cập nhật, xóa, tính toán...). Model không phụ thuộc vào View hay Controller, giúp dữ liệu có thể được tái sử dụng trong nhiều bối cảnh khác nhau mà không cần viết lại.
 - **View (Giao diện người dùng):** Là phần chịu trách nhiệm hiển thị thông tin cho người dùng và tiếp nhận các thao tác từ họ. View không chứa logic nghiệp vụ mà chỉ tập trung vào hiển thị dữ liệu được cung cấp bởi Controller hoặc Model. Điều này giúp nhà thiết kế giao diện có thể làm việc độc lập, cải tiến UI/UX mà không lo ảnh hưởng đến logic hệ thống.
 - **Controller (Xử lý nghiệp vụ):** Là cầu nối giữa View và Model. Controller tiếp nhận các yêu cầu từ người dùng thông qua View (ví dụ: bấm nút, điền form...), xử lý logic điều hướng, tương tác với dữ liệu thông qua Model, sau đó cập nhật kết quả hiển thị lên View. Việc xử lý logic được tập trung tại đây giúp code trở nên rõ ràng, dễ kiểm soát và bảo trì.
- 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.



Hình 1.1: Mô hình MVC

1.2 Đặc điểm

- **Phân tách rõ ràng vai trò:** Nhờ việc tách riêng từng thành phần, lập trình viên dễ dàng xác định chức năng và phạm vi xử lý của mỗi phần. Điều này giúp giảm thiểu tình trạng phụ thuộc lẫn nhau trong mã nguồn, tránh lỗi không mong muốn khi sửa đổi hoặc cập nhật hệ thống.
- **Tăng tính tái sử dụng:** Các thành phần trong mô hình MVC có thể được tái sử dụng độc lập. Ví dụ, một Model dùng để truy xuất thông tin tài liệu học tập có thể được dùng lại cho nhiều View khác nhau mà không cần viết lại logic dữ liệu. Tính tái sử dụng này góp phần rút ngắn thời gian phát triển và kiểm thử hệ thống.
- **Dễ bảo trì và mở rộng:** Khi dự án phát triển dài hạn, việc mở rộng thêm tính năng hoặc điều chỉnh logic nghiệp vụ là điều tất yếu. Với MVC, do mỗi module được quản lý riêng biệt, lập trình viên chỉ cần tập trung vào phần cần sửa mà không làm ảnh hưởng đến toàn bộ hệ thống. Điều này không chỉ tiết kiệm thời gian mà còn giúp hạn chế rủi ro phát sinh lỗi khi chỉnh sửa.
- **Hỗ trợ phát triển theo nhóm:** Mô hình MVC đặc biệt phù hợp với môi trường phát triển theo nhóm. Nhà thiết kế giao diện có thể tập trung xây dựng View mà không cần quan tâm đến logic xử lý, trong khi lập trình viên backend sẽ làm việc với Controller và Model. Việc phân chia này giúp tăng hiệu suất làm việc và giảm xung đột trong quá trình phát triển.
- **Thân thiện với SEO:** Một trong những điểm mạnh ít được chú ý của MVC là khả năng kiểm soát URL rõ ràng, có cấu trúc và thân thiện với công cụ tìm kiếm

(SEO). Điều này giúp các sản phẩm thương mại điện tử như website bán tài liệu học tập dễ dàng được tìm kiếm và tiếp cận hơn trên các công cụ như Google, Bing,...

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.

1.3 Nguyên lý hoạt động

Mô hình **MVC (Model – View – Controller)** hoạt động dựa trên nguyên tắc phân tách nhiệm vụ rõ ràng giữa ba thành phần chính, từ đó tối ưu hóa quá trình xử lý yêu cầu của người dùng và phản hồi dữ liệu một cách hiệu quả. Quy trình hoạt động tổng quát được mô tả như sau:

1. Người dùng gửi yêu cầu (HTTP Request) thông qua trình duyệt web, thường là khi họ truy cập một đường dẫn URL, nhấn vào nút hoặc gửi form dữ liệu.

2. Routing Middleware trong ASP.NET Core sẽ tiếp nhận yêu cầu và phân tích URL, từ đó ánh xạ tới Controller và Action phù hợp đã được định nghĩa trong hệ thống.

3. Controller đóng vai trò trung gian, chịu trách nhiệm xử lý yêu cầu:

- Nếu yêu cầu cần truy xuất hoặc xử lý dữ liệu, Controller sẽ gọi đến Model để thực hiện các thao tác như truy vấn, cập nhật hoặc xử lý nghiệp vụ.
- Sau khi có dữ liệu (hoặc không), Controller sẽ chọn một View tương ứng để hiển thị thông tin cho người dùng.

4. Model thực hiện các logic nghiệp vụ, xử lý dữ liệu và tương tác với cơ sở dữ liệu (CSDL), đảm bảo dữ liệu được truy xuất hoặc cập nhật một cách chính xác và an toàn.

5. View nhận dữ liệu từ Controller và thực hiện nhiệm vụ hiển thị giao diện phù hợp với nội dung cần hiển thị, tạo ra trang HTML để người dùng xem được trên trình duyệt.

6. Phản hồi (HTTP Response) chứa nội dung đã xử lý được gửi trả lại cho trình duyệt của người dùng.

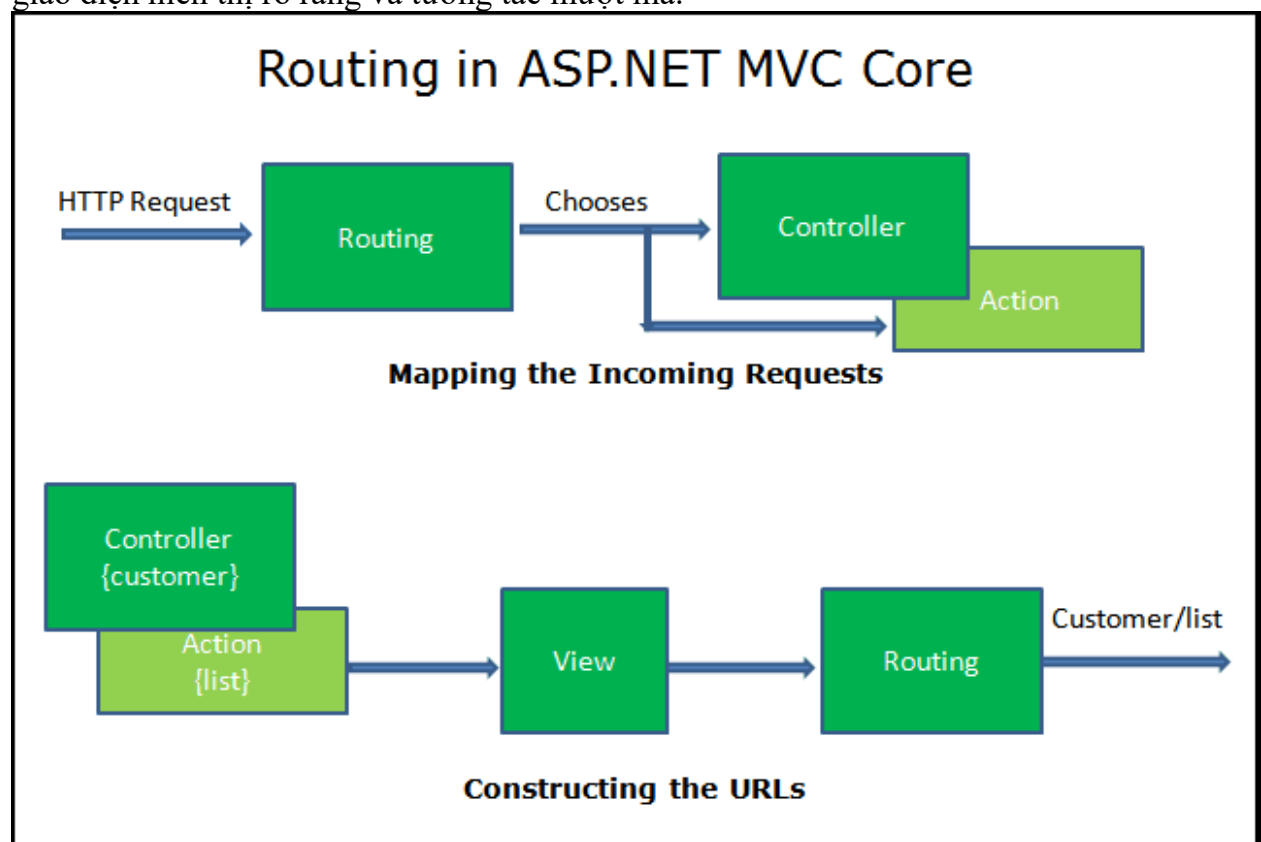
Tổng quan luồng xử lý trong mô hình MVC:

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...).

Trong thực tế triển khai, mô hình MVC không chỉ giúp tổ chức mã nguồn một cách khoa học mà còn:

- Tăng hiệu năng xử lý: Giảm thiểu số lần truy vấn cơ sở dữ liệu nhờ khả năng kiểm soát logic tại Controller và sử dụng bộ nhớ đệm hợp lý.
- Tối ưu hiển thị trên nhiều thiết bị: View có thể được thiết kế linh hoạt để hiển thị phù hợp trên nhiều nền tảng như máy tính để bàn, máy tính bảng và điện thoại di động.
- Nâng cao trải nghiệm người dùng: Thông tin được xử lý nhanh chóng, giao diện hiển thị rõ ràng và tương tác mượt mà.



Hình:1.2 Nguyên lý hoạt động

1.4 Công nghệ triển khai ASP.NET CoreGiới thiệu về ASP.NET Core

Khái niệm:

ASP.NET Core là một framework mã nguồn mở, đa nền tảng (cross-platform) do Microsoft phát triển, được sử dụng để xây dựng các ứng dụng web hiện đại, dịch vụ web (Web Services) và API. Đây là phiên bản cải tiến và được tái cấu trúc hoàn toàn từ nền tảng **ASP.NET truyền thống**, nhằm đáp ứng nhu cầu ngày càng cao về hiệu suất, khả năng mở rộng và sự linh hoạt trong triển khai.

ASP.NET Core có thể chạy trên nhiều hệ điều hành khác nhau như **Windows, Linux và macOS**, cho phép lập trình viên lựa chọn môi trường phát triển và triển khai phù hợp với hệ thống thực tế. Với kiến trúc hiện đại, ASP.NET Core loại bỏ những ràng buộc cũ của ASP.NET, cung cấp hiệu năng vượt trội và khả năng tích hợp mạnh mẽ với các công nghệ mới như **Docker, Kubernetes**, hay các nền tảng điện toán đám mây như **Azure, AWS**.

Đặc biệt, ASP.NET Core hỗ trợ nhiều mô hình lập trình khác nhau như **MVC (Model - View - Controller)**, **Razor Pages**, **Blazor** và **Web API**, tạo điều kiện linh hoạt cho việc xây dựng ứng dụng theo nhu cầu thực tế.

Tính đến thời điểm hiện tại, ASP.NET Core được xem là công nghệ chủ lực trong hệ sinh thái phát triển ứng dụng web của Microsoft, được sử dụng rộng rãi trong các dự án từ quy mô nhỏ đến lớn trên toàn cầu.

Đặc điểm nổi bật:

ASP.NET Core sở hữu nhiều đặc điểm nổi bật giúp nó trở thành lựa chọn hàng đầu trong phát triển ứng dụng web hiện đại:

Đa nền tảng (Cross-platform): Hỗ trợ chạy trên nhiều hệ điều hành khác nhau như Windows, Linux và macOS, tạo sự linh hoạt trong phát triển và triển khai hệ thống.

Hiệu năng cao: Sử dụng **Kestrel**, một web server nhẹ và tốc độ cao được tích hợp sẵn, giúp tối ưu hóa thời gian phản hồi và khả năng xử lý của ứng dụng.

Hỗ trợ Dependency Injection (DI) tích hợp sẵn: Cơ chế tiêm phụ thuộc giúp tách biệt các thành phần trong hệ thống, nâng cao khả năng kiểm thử, mở rộng và bảo trì.

Tích hợp tốt với Entity Framework Core (Code First): Cho phép xây dựng và quản lý cơ sở dữ liệu thông qua các lớp mô hình trong mã nguồn, giúp giảm thiểu lỗi và tăng hiệu quả phát triển.

Kiến trúc dạng module (Modular Architecture): ASP.NET Core tổ chức luồng xử lý thông qua hệ thống middleware có thể cấu hình tùy ý. Các middleware

n như UseRouting, UseAuthentication, UseAuthorization... giúp kiểm soát chi tiết từng bước trong quá trình xử lý yêu cầu từ client đến server.

Hỗ trợ tốt cho RESTful API và WebSocket: Cho phép xây dựng các dịch vụ API hiện đại, phục vụ nhu cầu trao đổi dữ liệu giữa client và server theo thời gian thực hoặc qua giao tiếp HTTP tiêu chuẩn.

Bảo mật mạnh mẽ: Tích hợp sẵn các cơ chế **xác thực (Authentication)** và **phân quyền (Authorization)**, hỗ trợ đa dạng các phương thức bảo mật như cookie, JWT, OAuth2, và xác thực bên ngoài (Google, Facebook, Microsoft...).

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.

Cấu trúc một ứng dụng ASP.NET Core MVC

1. /Controllers

Thư mục chứa các lớp *Controller* – thành phần chịu trách nhiệm tiếp nhận yêu cầu từ người dùng (HTTP Request), xử lý logic điều hướng và tương tác với Model hoặc View tương ứng.

2. /Models

Chứa các lớp mô hình đại diện cho dữ liệu và logic nghiệp vụ. Các lớp trong thư mục này thường được sử dụng để ánh xạ với cơ sở dữ liệu thông qua Entity Framework Core (Code First), hoặc định nghĩa các đối tượng trung gian cho việc truyền/nhận dữ liệu.

3. /Views

Thư mục chứa các *giao diện người dùng* dưới dạng các tệp .cshtml (Razor View). Mỗi View tương ứng với một Action trong Controller. View có thể hiển thị dữ liệu động và hỗ trợ cú pháp Razor để tương tác với dữ liệu từ Controller.

4. /wwwroot

Thư mục tĩnh chứa các tài nguyên phía client như:

- HTML, CSS, JavaScript, ảnh, fonts,...
- Đây là thư mục duy nhất được phép truy cập công khai từ phía trình duyệt.

5. appsettings.json

Tệp cấu hình ứng dụng, dùng để lưu trữ các giá trị như:

- Chuỗi kết nối cơ sở dữ liệu (ConnectionStrings)
- Cấu hình môi trường, logging, email, bảo mật...
- Hỗ trợ phân tách theo môi trường qua appsettings.Development.json, appsettings.Production.json,...

6. Program.cs

Điểm khởi chạy của ứng dụng (entry point). Trong phiên bản .NET 6 trở đi, cấu hình dịch vụ (DI), middleware và các thiết lập chạy ứng dụng thường được viết trực tiếp trong tệp này.

7. Startup.cs (từ phiên bản .NET)

Trong các phiên bản ASP.NET Core trước .NET 6 (như .NET Core 3.1, .NET 5), tệp Startup.cs là nơi định nghĩa các phương thức:

- ConfigureServices() – đăng ký các dịch vụ và cấu hình DI container.
- Configure() – thiết lập pipeline xử lý HTTP request thông qua middleware.

Mô hình MVC trong ASP.NET Core

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.

Nguyên lý hoạt động

- Người dùng gửi yêu cầu qua URL.
- **Routing Middleware** phân tích và chuyển yêu cầu đến Controller phù hợp.
- Controller xử lý logic, có thể gọi Model.
- Dữ liệu được trả về View để hiển thị.
- 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.

Ví dụ minh họa

1. Model

```
public class Order
{
    public int Id { get; set; }
    public string Email { get; set; }
```

```

    public decimal OrderTotal { get; set; }
    public DateTime OrderPlaced { get; set; }
    public List<OrderDetail>? OrderDetails { get; set; }
}
3.2. Controller
namespace Documentt.Controllers
{
    public class DocumentsController : Controller
    {
        private static List<Document> _documents = new List<Document>
        {
            new Document { Id = 1, Name = "Báo cáo thực tập công ty ABC", Price =
25000M, Description = "Báo cáo đầy đủ." },
            new Document { Id = 2, Name = "Đồ án quản lý bán hàng WinForms", Price
= 75000M, Description = "Đồ án có mã nguồn và báo cáo chi tiết." },
            new Document { Id = 3, Name = "Tài liệu ôn tập Nguyên lý kế toán", Price
= 18000M, Description = "Tổng hợp kiến thức trọng tâm và bài tập có lời giải." }
        };

        public IActionResult Index()
        {
            return View(_documents);
        }
    }
}

```

2. View

```
@model IEnumerable<Documentt.Models.Document>
```

```
@{
    ViewData["Title"] = "Danh sách tài liệu";
}
```

```
<h2 class="text-2xl font-semibold mb-4">Danh sách tài liệu học tập, báo cáo, đồ án</h2>
```

```
<table class="table-auto border border-collapse w-full text-sm">
    <thead class="bg-gray-100">
        <tr>
            <th class="border px-4 py-2 text-left">Tiêu đề</th>
            <th class="border px-4 py-2 text-left">Chuyên mục</th>
            <th class="border px-4 py-2 text-left">Ngày đăng</th>
            <th class="border px-4 py-2 text-left">Giá</th>

```

```

        <th class="border px-4 py-2 text-left">Mô tả</th>
        <th class="border px-4 py-2 text-left">Thao tác</th>
    </tr>
</thead>
<tbody>
    @foreach (var item in Model)
    {
        <tr class="hover:bg-gray-50">
            <td class="border px-4 py-2">@item.Title</td>
            <td class="border px-4 py-2">@item.Category</td>
            <td class="border px-4 py-
2">@item.UploadDate.ToShortDateString()</td>
            <td class="border px-4 py-2">@item.Price.ToString("N0") đ</td>
            <td class="border px-4 py-2">@item.Description</td>
            <td class="border px-4 py-2">
                <a asp-action="Details" asp-route-id="@item.Id" class="text-blue-600
hover:underline">Xem chi tiết</a>
            </td>
        </tr>
    }
</tbody>
</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 phát triển mạnh mẽ, việc học tập không còn chỉ giới hạn trong lớp học hay thư viện truyền thống. Sinh viên, học viên và giảng viên ngày càng có nhu cầu tiếp cận các tài liệu học tập một cách nhanh chóng, thuận tiện và chất lượng hơn. Tuy nhiên, hiện nay vẫn còn nhiều vấn đề tồn tại:
- Tài liệu học tập bị phân tán trên nhiều nền tảng khác nhau, gây khó khăn trong việc tìm kiếm.
- Nhiều tài liệu không được kiểm duyệt về chất lượng, dễ gây nhiễu thông tin.
- Sinh viên thường phải tìm kiếm tài liệu qua bạn bè, nhóm học tập hoặc các trang mạng xã hội, dẫn đến hiệu quả thấp.
- Những người sở hữu tài liệu (giảng viên, sinh viên có kinh nghiệm) chưa có môi trường phù hợp để chia sẻ và thương mại hóa tri thức của mình một cách dễ dàng và hợp pháp.
- Từ thực tế đó, bài toán đặt ra là cần xây dựng một nền tảng website thương mại điện tử chuyên về tài liệu học tập – nơi mà người dùng có thể mua bán, chia sẻ và quản lý tài liệu học tập một cách thuận tiện, bảo mật và hiệu quả.

2.1.2 Mục tiêu của ứng dụng web

Ứng dụng web StudyDocs được xây dựng nhằm đáp ứng những yêu cầu trên. Cụ thể:

- Cung cấp giao diện thân thiện cho người dùng dễ dàng đăng nhập, tìm kiếm và mua bán tài liệu học tập.
- Cho phép người dùng tạo tài khoản cá nhân, đăng tải tài liệu, phân loại theo danh mục như: Toán học, Lập trình, Kinh tế, Kỹ thuật, Ngôn ngữ học, v.v.
- Tích hợp các chức năng của một website thương mại điện tử như: giỏ hàng, bộ lọc theo danh mục, tìm kiếm nâng cao, đánh giá sản phẩm, thanh toán và tải tài liệu.
- Hỗ trợ người dùng quản lý hồ sơ cá nhân, lịch sử giao dịch, và nạp tiền để mua tài liệu.
- Góp phần kết nối cộng đồng học thuật, tạo môi trường học tập mở và năng động.

2.1.3 Vai trò và giá trị mang lại

- Dự án “Xây dựng website thương mại điện tử kinh doanh sản phẩm tài liệu học tập, báo cáo, đồ án” được triển khai nhằm đáp ứng nhu cầu ngày càng cao trong việc tìm kiếm, chia sẻ và mua bán các tài liệu học tập chất lượng

- của sinh viên, học sinh và các cá nhân hoạt động trong lĩnh vực giáo dục – đào tạo.
- Website đóng vai trò như một nền tảng trung gian kết nối giữa người đăng tải (người bán) và người học (người mua) thông qua môi trường trực tuyến hiện đại, thuận tiện, an toàn và tối ưu trải nghiệm.
 - Dự án được xây dựng với vai trò là một nền tảng kết nối giữa người chia sẻ (người bán) và người học (người mua), nơi cung cấp môi trường thuận tiện để trao đổi, mua bán và chia sẻ các tài liệu học tập, bao gồm: tài liệu lý thuyết, bài giảng, báo cáo môn học, đồ án tốt nghiệp, v.v.
 - Trình bày danh mục tài liệu theo chuyên ngành, thể loại và trình độ học tập
 - Cho phép người dùng dễ dàng tìm kiếm tài liệu phù hợp với nhu cầu, ví dụ: Công nghệ thông tin, Kế toán, Báo cáo thực tập, Đồ án tốt nghiệp,...
 - Hệ thống tìm kiếm và lọc nâng cao
 - Tìm kiếm tài liệu theo từ khóa, tiêu đề, người đăng, chuyên mục, giá, ngày đăng, mức độ phổ biến hoặc đánh giá từ cộng đồng.
 - Giỏ hàng và thanh toán tài liệu
 - Người dùng có thể thêm tài liệu vào giỏ hàng, thực hiện thanh toán bằng số dư tài khoản. Sau khi thanh toán thành công, tài liệu sẽ được cho phép tải xuống.
 - Đăng ký / Đăng nhập tài khoản người dùng
 - Cho phép người học tạo tài khoản, đăng nhập và quản lý hồ sơ cá nhân.
 - Định hướng cộng đồng chia sẻ tri thức
 - Bên cạnh chức năng bán hàng, hệ thống còn khuyến khích chia sẻ tài liệu miễn phí hoặc có phí thấp, góp phần xây dựng một cộng đồng học thuật tích cực, nơi mọi người cùng đóng góp và tiếp cận tri thức dễ dàng hơn.

2.1.4. ý nghĩa thực tiễn

- Dự án StudyDocs – Website thương mại điện tử mua bán tài liệu học tập không chỉ mang lại giá trị ứng dụng thực tế mà còn có ý nghĩa học thuật sâu sắc đối với sinh viên ngành công nghệ thông tin. Việc triển khai dự án bằng mô hình Code First của Entity Framework giúp sinh viên nắm vững và áp

dụng một cách thực tiễn kiến thức lập trình hướng đối tượng (OOP), quản lý cơ sở dữ liệu, và phát triển ứng dụng web hiện đại.

- Giải quyết bài toán thực tế trong giáo dục: Dự án đáp ứng nhu cầu tìm kiếm, chia sẻ và thương mại hóa tài liệu học tập – một nhu cầu có thật trong môi trường đại học hiện nay. Điều này giúp sinh viên thấy rõ việc học lập trình không chỉ là lý thuyết mà hoàn toàn có thể áp dụng để giải quyết vấn đề thực tế.
- Gần với mô hình doanh nghiệp: Với các tính năng như quản lý người dùng, đăng nhập, phân quyền, thêm sản phẩm, giỏ hàng, thanh toán và lịch sử mua hàng, dự án mô phỏng gần giống hệ thống e-commerce ngoài thực tế. Điều này giúp sinh viên có thêm kinh nghiệm làm việc với các mô hình dữ liệu phức tạp và trải nghiệm quy trình phát triển phần mềm giống môi trường công ty.
- Tăng khả năng làm việc nhóm và quản lý dự án: Nếu dự án được triển khai theo nhóm, sinh viên còn học được cách sử dụng Git, chia công việc, quản lý tiến độ, làm tài liệu – những kỹ năng cực kỳ cần thiết khi bước vào môi trường làm việc chuyên nghiệp.

2.2 Phân tích yêu cầu của ứng dụng

2.2.1. Mô tả chức năng ứng dụng

- Mục tiêu chính của đề tài là xây dựng một hệ thống website thương mại điện tử hoạt động trên môi trường web, nhằm hỗ trợ người dùng trong việc mua bán, chia sẻ tài liệu học tập, báo cáo, đề án một cách dễ dàng, nhanh chóng và hiệu quả.
- Hệ thống không chỉ giúp người quản lý kiểm soát tốt tài liệu và đơn hàng mà còn mang đến trải nghiệm thân thiện cho người học khi cần tìm kiếm và tải về tài liệu cần thiết mà không cần đến tận nơi hay phụ thuộc vào thời gian.

Chức năng dành cho người quản trị (Admin):

Quản lý danh mục tài liệu:

- Thêm, sửa, xóa các danh mục như: CNTT, Kế toán, Marketing, Ngôn ngữ Anh,...

Quản lý loại tài liệu:

- Phân loại theo mục đích sử dụng như: Bài giảng, Đề cương, Báo cáo thực tập, Đồ án tốt nghiệp,...

Quản lý sản phẩm (tài liệu):

- Thêm mới, chỉnh sửa, cập nhật mô tả, giá bán và trạng thái hiển thị của từng tài liệu.

Cập nhật giá bán tài liệu:

- Có thể điều chỉnh giá tài liệu dựa trên nhu cầu hoặc chương trình khuyến mãi.(điều chỉnh ở phần quản lý sản phẩm)

Quản lý người dùng:

- Theo dõi danh sách người dùng, phân quyền truy cập, khóa/mở tài khoản.

Chức năng dành cho khách hàng (người dùng):

- Khách hàng truy cập website có thể sử dụng các tính năng sau:
 - Tìm kiếm tài liệu:

Theo từ khóa, tên tài liệu, người đăng, danh mục, loại tài liệu...

Xem danh sách tài liệu:

- **Danh sách tài liệu mới nhất, phổ biến, miễn phí hoặc có phí.**

Xem chi tiết tài liệu:

- **Gồm mô tả, người đăng, ngày đăng, giá bán, lượt tải, đánh giá, hình ảnh...**

Thêm tài liệu vào giỏ hàng:

- **Cho phép lưu nhiều tài liệu để thanh toán sau.**

Quản lý giỏ hàng:

- **Thêm, xóa, cập nhật hoặc xóa toàn bộ tài liệu khỏi giỏ.**

Đăng ký tài khoản:

- **Đăng ký với họ tên, email, mật khẩu để trở thành thành viên.**

Đăng nhập/Đăng xuất:

- **Truy cập vào tài khoản cá nhân để sử dụng các tính năng nâng cao.**

Liên hệ với ban quản trị:

- **Gửi thắc mắc, góp ý, yêu cầu hỗ trợ qua trang liên hệ.**

2.2.2 Yêu cầu phi chức năng (Non-Functional Requirements)

Ngoài các yêu cầu chức năng, hệ thống cần đáp ứng một số yêu cầu phi chức năng quan trọng để đảm bảo hiệu năng, độ tin cậy và khả năng mở rộng lâu dài. Những yêu cầu này được chia thành các nhóm cụ thể như sau:

2.2.3. Hiệu suất (Performance)

Ứng dụng cần đảm bảo tốc độ phản hồi nhanh, thời gian tải toàn bộ nội dung trên mỗi trang không vượt quá 2 giây trong điều kiện mạng ổn định.

Giao dịch thanh toán phải được xử lý nhanh chóng, toàn bộ quá trình từ khi người dùng nhấn “thanh toán” đến khi nhận được xác nhận không vượt quá 5 giây.

Các hình ảnh tài liệu và ảnh xem trước cần được nén và tối ưu hóa để tránh gây chậm trễ trong việc hiển thị và tải trang.

2.2.4. Khả năng sử dụng

Giao diện hệ thống cần được thiết kế trực quan, thân thiện, dễ thao tác đối với mọi đối tượng người dùng, kể cả người không rành về công nghệ.

Website phải tương thích với các trình duyệt phổ biến hiện nay như Google Chrome, Firefox, Microsoft Edge, Safari,...

Hệ thống cần hỗ trợ hiển thị tốt trên nhiều thiết bị như máy tính để bàn, máy tính xách tay, máy tính bảng và điện thoại thông minh.

Giao diện cần có khả năng tự điều chỉnh linh hoạt với kích thước màn hình (responsive design) nhằm tối ưu trải nghiệm người dùng.

Các thông báo hệ thống như thành công, lỗi, cảnh báo,... cần được hiển thị rõ ràng, dễ hiểu và nhất quán.

2.2.5. Bảo mật

Hệ thống sử dụng ASP.NET Core Identity để quản lý người dùng, thực hiện xác thực, phân quyền truy cập.

Mật khẩu người dùng phải được mã hóa bằng thuật toán Hash hiện đại, không lưu trữ dưới dạng văn bản thuần.

Phân quyền được thực hiện rõ ràng giữa người dùng thông thường, người đăng bán tài liệu và quản trị viên, đảm bảo người dùng chỉ có quyền thao tác trong phạm vi cho phép.

2.2.6. Khả năng bảo trì

Hệ thống được xây dựng theo mô hình kiến trúc ASP.NET MVC, giúp tổ chức mã nguồn rõ ràng và dễ quản lý.

Mỗi chức năng được phân tách hợp lý theo từng Controller, Service và Interface, thuận tiện trong việc mở rộng, chỉnh sửa hoặc nâng cấp hệ thống.

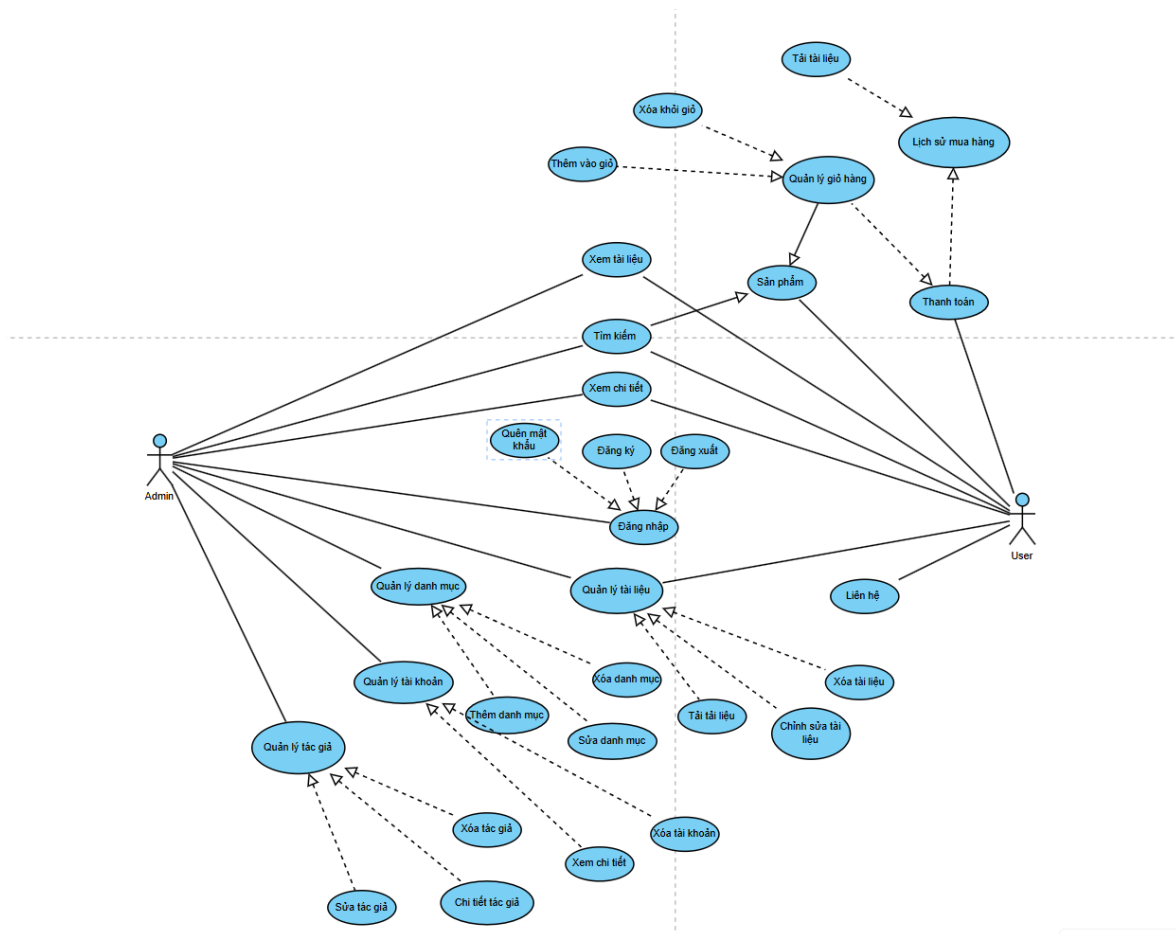
2.2.7. Độ tin cậy

Trong trường hợp xảy ra sự cố, hệ thống cần có cơ chế tự động xử lý lỗi, đảm bảo không làm mất dữ liệu người dùng và giảm thiểu thời gian gián đoạn.

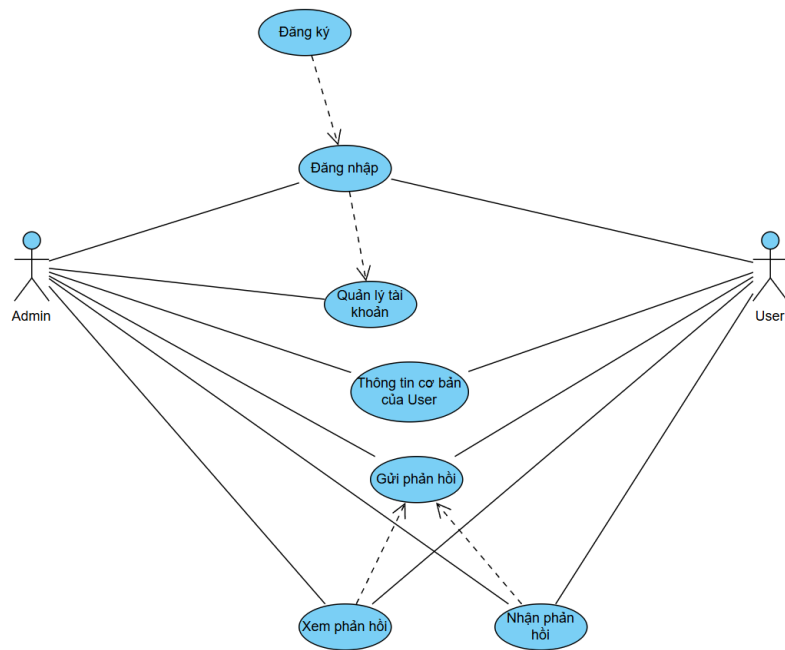
Dữ liệu quan trọng như tài khoản người dùng, đơn hàng, tài liệu đăng bán cần được sao lưu định kỳ và có kế hoạch phục hồi trong các tình huống khẩn cấp.

Hệ thống cần được kiểm thử kỹ lưỡng trước khi triển khai để phát hiện và khắc phục kịp thời các lỗi tiềm ẩn về logic, hiệu năng và bảo mật.

2.2.3. Biểu đồ Use-Case



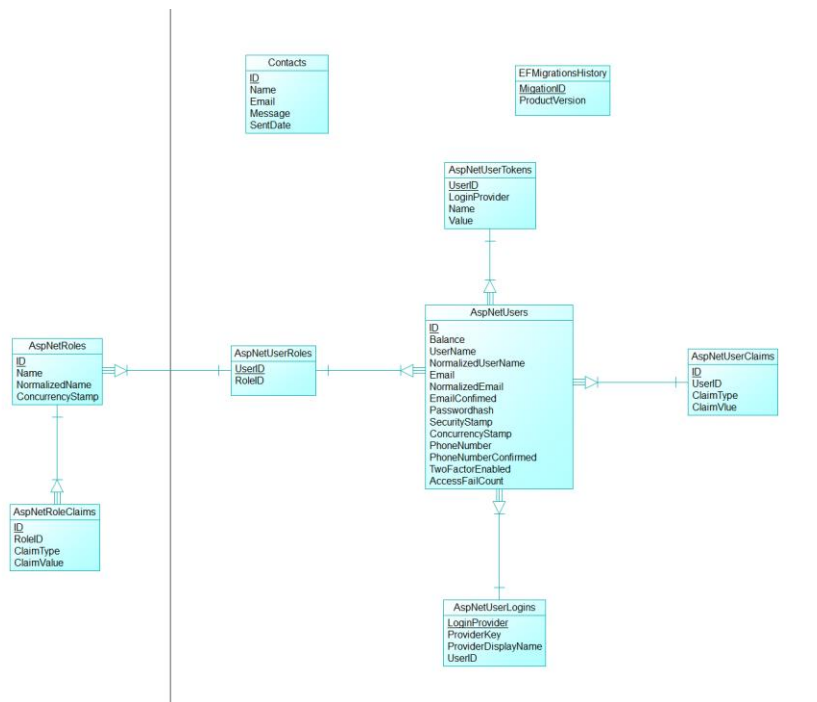
Hình 2.1: Sơ đồ UseCase mô tả chức năng của admin và user



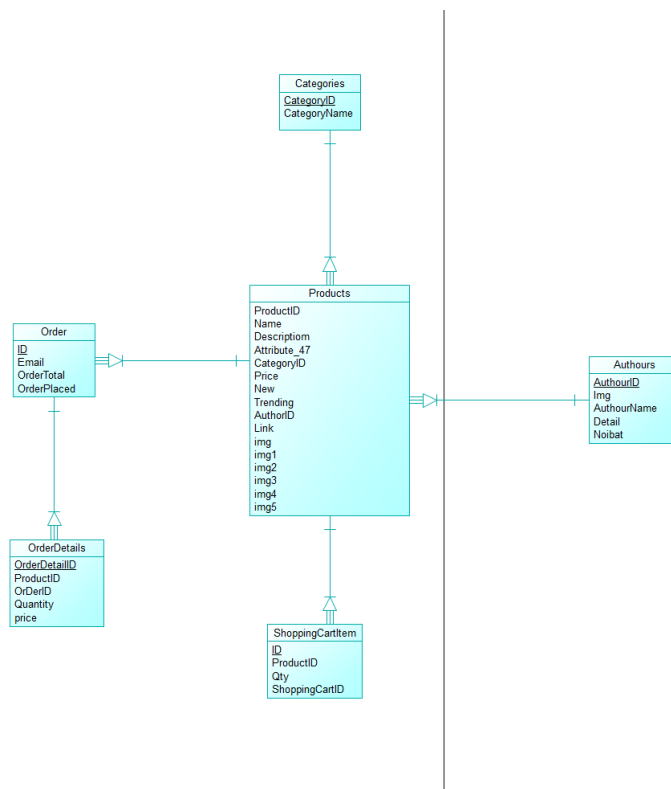
Hình 2.2: Sơ đồ UseCase mô tả sự tương tác giữa người dùng(User) với người quản trị(Admin)

2.3 Thiết kế cơ sở dữ liệu:

Thiết kế logic ERD:

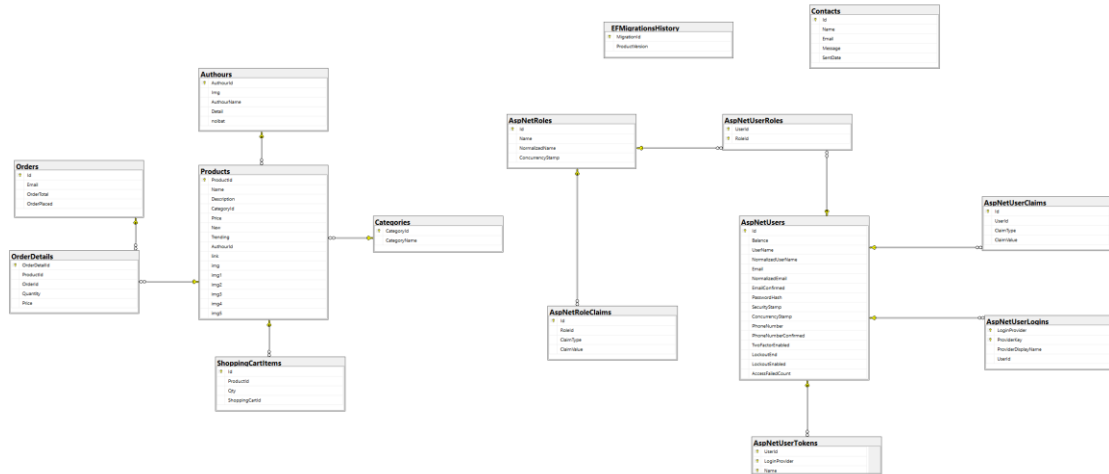


Hình 2.3.1: Phân hệ thống người dùng (Identity)



Hình 2.3.2: Hình dữ liệu nghiệp vụ(domain data)

Mô hình vật lý



Hình 2.2.3: Mô hình Cơ sở dữ liệu

- Mô tả chi tiết các bảng
- **Bảng dữ liệu nghiệp vụ (domain data)**

Tên cột	Kiểu Dữ liệu	Mô tả
CategoryID	Int	Khóa chính, là mã định danh duy nhất của từng danh mục sản phẩm.
CategoryName	String	Tên của danh mục sản phẩm.

Bảng 1: Categories

Tên cột	Kiểu Dữ liệu	Mô tả
AuthorID	Int	Khóa chính, định danh duy nhất cho mỗi tác giả.
Img	String	Đường dẫn đến ảnh đại diện của tác giả.
AuthorName	String	Tên đầy đủ của tác giả.
Detail	String	Mô tả chi tiết về tác giả (tiểu sử, sự nghiệp, tác phẩm nổi bật...).
Noibat	Bool	Cờ đánh dấu cho biết tác giả có phải là tác giả nổi bật hay không (true nếu nổi bật).

Bảng 2: Authors

Tên cột	Kiểu Dữ liệu	Mô tả
ProductID	Int	Khóa chính, định danh duy nhất cho từng sản phẩm.
Name	String	Tên của sản phẩm
Description	String	Mô tả chi tiết nội dung hoặc đặc điểm sản phẩm.
CategoryID	Int	Trường thuộc tính phụ, có thể dùng để lưu thông tin mở rộng.
Category	Category	Khóa ngoại liên kết đến bảng Categories, xác định sản phẩm thuộc danh mục nào.
Price	Decimal	Giá bán của sản phẩm.
New	Bool	Đánh dấu sản phẩm mới phát hành.

Trending	Bool	Cờ đánh dấu sản phẩm đang được quan tâm nhiều.
AuthorID	Int	Khóa ngoại liên kết đến bảng Authors, xác định tác giả của sản phẩm.
Author	Author	Xác thực
Link	String	Đường dẫn đến trang chi tiết hoặc nơi mua sản phẩm.
Img	String	Các đường dẫn ảnh minh họa cho sản phẩm.
Img1	String	Phục vụ hiển thị đa dạng hình ảnh
Img2	String	Phục vụ hiển thị đa dạng hình ảnh
Img3	String	Phục vụ hiển thị đa dạng hình ảnh
Img4	String	Phục vụ hiển thị đa dạng hình ảnh
Img5	String	Phục vụ hiển thị đa dạng hình ảnh

Bảng 3: Products

Tên cột	Kiểu Dữ liệu	Mô tả
Id	Int	Khóa chính, định danh duy nhất cho từng đơn hàng.
Email	String	Địa chỉ email của khách hàng đã đặt hàng.
OrderTotal	Decimal	Tổng giá trị đơn hàng tại thời điểm đặt.
OrderPlaced	Datetime	Thời gian mà khách hàng đã thực hiện đặt hàng.

Bảng 4: Order

Tên cột	Kiểu Dữ liệu	Mô tả
OrderDetailID	Int	Khóa chính, định danh duy nhất cho từng chi tiết đơn hàng.
ProductID	Int	Khóa ngoại liên kết đến bảng Products, xác định sản phẩm trong đơn hàng.
OrderID	Int	Khóa ngoại liên kết đến bảng Order, xác định đơn hàng chứa sản phẩm.
Quantity	Int	Số lượng sản phẩm được đặt trong chi tiết đơn hàng.
Price	Decimal	Giá của sản phẩm tại thời điểm đặt hàng (có thể khác với giá hiện tại).

Bảng 5: OrderDetails

Tên cột	Kiểu Dữ liệu	Mô tả
Id	Int	Khóa chính, định danh duy nhất cho từng mục trong giỏ hàng.
ProductID	Int	Khóa ngoại đến bảng Products, cho biết sản phẩm được thêm vào giỏ hàng.
Qty	Int	Số lượng sản phẩm trong giỏ hàng.
ShoppingCartID	String	Mã định danh cho giỏ hàng

Bảng 6: ShoppingCartItem

Bảng dữ liệu Identity

.AspNetUsers

- Đây là bảng chính lưu trữ thông tin người dùng trong hệ thống. Identity tạo ra bảng này với các cột mặc định như UserName, Email, PasswordHash, PhoneNumber, SecurityStamp... để phục vụ cho chức năng đăng nhập và xác thực người dùng. Trong hệ thống này, bảng AspNetUsers đã được mở rộng thêm một số trường như:
 - FullName: Họ tên đầy đủ của người dùng, phục vụ việc hiển thị và xử lý thông tin trong các chức năng liên quan.
 - Balance: Số dư tài khoản của người dùng (ví dụ trong trường hợp người dùng có thể mua tài liệu hoặc sản phẩm trực tuyến).

.AspNetRoles

Bảng này lưu trữ các vai trò (quyền) trong hệ thống, ví dụ như:

- Admin: Người quản trị hệ thống, có quyền chỉnh sửa dữ liệu, phê duyệt sản phẩm, thống kê.
- Khách hàng: Người dùng phổ thông, có thể đăng nhập, xem và đặt mua tài liệu.
- Các vai trò khác có thể được thêm tùy theo nhu cầu hệ thống như: Editor, Manager, Staff, v.v

.AspNetUserRoles

- Bảng này dùng để **liên kết nhiều-nhiều** giữa người dùng và vai trò. Một người dùng có thể mang nhiều vai trò, và một vai trò có thể được gán cho nhiều người dùng. Đây là bảng trung gian giữa AspNetUsers và AspNetRoles.

.AspNetUserClaims

- Bảng này lưu trữ các claims (yêu cầu mở rộng) được gán cho từng người dùng. Claims là những thông tin bổ sung có thể dùng để kiểm soát quyền truy cập linh hoạt hơn, ví dụ như:
 - "IsPremiumUser": true
 - "Department": "Marketing"
 - "AccessLevel": "ReadOnly"

Các bảng hỗ trợ khác như AspNetUserClaims, AspNetUserLogins, AspNetUserTokens phục vụ cho các chức năng xác thực và phân quyền nâng cao.

2.4 Thiết kế các thành phần MVC:

2.4.1. Model

- Tầng Controller đóng vai trò trung gian giữa người dùng và hệ thống. Đây là nơi tiếp nhận các yêu cầu từ phía client (trình duyệt), xử lý logic cần thiết, sau đó gọi tới tầng Model để thao tác dữ liệu, và cuối cùng trả về View tương ứng để hiển thị kết quả.
- Mỗi controller đại diện cho một nhóm chức năng riêng biệt trong hệ thống. Các phương thức bên trong controller (gọi là action) sẽ xử lý các thao tác như hiển thị danh sách, tạo mới, cập nhật hoặc xóa dữ liệu. Controller giúp đảm bảo quá trình luân chuyển dữ liệu diễn ra mạch lạc và rõ ràng giữa các tầng.
- **Class Product**

```
public class Product
{
    41 references
    public int ProductId { get; set; }
    34 references
    public string Name { get; set; }
    16 references
    public string Description { get; set; }
    20 references
    public int CategoryId { get; set; }
    18 references
    public Category? Category { get; set; }
    27 references
    public decimal Price { get; set; }
    15 references
    public bool New { get; set; }
    15 references
    public bool Trending { get; set; }
    18 references
    public int AuthourId { get; set; }
    7 references
    public Authour? Authour { get; set; }
    16 references
    public string link { get; set; }
    29 references
    public string img { get; set; }
    15 references
    public string? img1 { get; set; }
    15 references
    public string? img2 { get; set; }
    15 references
    public string? img3 { get; set; }
    15 references
    public string? img4 { get; set; }
    15 references
    public string? img5 { get; set; }
}
```

Mô tả chi tiết bảng 3

Class Order

```
0 references
public class Order
{
    1 reference
    public int Id { get; set; }
    3 references
    public string Email { get; set; }
    2 references
    public decimal OrderTotal { get; set; }
    3 references
    public DateTime OrderPlaced { get; set; }
    2 references
    public List<OrderDetail>? OrderDetails { get; set; }
}
```

Mô tả chi tiết bảng 4

Class OrderDetail

```
public class OrderDetail
{
    0 references
    public int OrderDetailId { get; set; }
    2 references
    public int ProductId { get; set; }
    4 references
    public Product? Product { get; set; }
    1 reference
    public int OrderId { get; set; }
    3 references
    public Order? Order { get; set; }
    2 references
    public int Quantity { get; set; }
    2 references
    public decimal Price { get; set; }
}
```

Mô tả chi tiết bảng 5

Class ShopingcartItem

```
public class ShopingcartItem
{
    0 references
    public int Id { get; set; }

    19 references
    public Product? Product { get; set; }

    4 references
    public int Qty { get; set; }

    6 references
    public string? ShoppingCartId { get; set; }
}
```

Mô tả chi tiết bảng 6

Class Authour

```
37 references
public class Authour
{
    18 references
    public int AuthourId { get; set; }
    22 references
    public string Img { get; set; }
    24 references
    public string AuthourName { get; set; }
    19 references
    public string Detail { get; set; }
    15 references
    public bool noibat { get; set; }
    7 references
    public List<Product>? Products { get; set; }
}
```

Mô tả chi tiết bảng 2

Class Category

```
28 references
public class Category
{
    21 references
    public int CategoryId { get; set; }

    [Required(ErrorMessage = "Tên danh mục là bắt buộc.")]
    [StringLength(100, ErrorMessage = "Tên danh mục không được vượt quá 100 ký tự.")]
    29 references
    public string CategoryName { get; set; } = string.Empty;

    1 reference
    public List<Product>? Products { get; set; }
}
```


Mô tả chi tiết bảng 1

2.4.2.View

- Tầng View chịu trách nhiệm trình bày dữ liệu ra giao diện người dùng. Nó được xây dựng bằng Razor (.cshtml) – một công cụ kết hợp giữa cú pháp HTML và C#, cho phép hiển thị dữ liệu động một cách linh hoạt.
- View không xử lý logic nghiệp vụ mà chỉ tập trung vào việc trình bày thông tin. Dữ liệu cần hiển thị sẽ được truyền từ Controller sang View thông qua Model hoặc ViewModel. Tầng View đóng vai trò quan trọng trong việc tạo ra trải nghiệm người dùng thân thiện, dễ sử dụng và phù hợp với chức năng hệ thống.

Home

```
<section id="home" class="hero-section h-96 flex items-center justify-center text-white bg-cover bg-center bg-fixed pt-16" style="background-image: url('h...>
  <div class="text-center bg-black bg-opacity-50 p-8 rounded-lg">
    <h2 class="text-4xl font-bold mb-4">Khám phá tài liệu học tập chất lượng</h2>
    <p class="text-lg mb-6">Hàng ngàn tài liệu từ sinh viên và chuyên gia trên toàn quốc, sẵn sàng hỗ trợ bạn học tập hiệu quả.</p>
    <a asp-controller="Product" asp-action="Index" class="bg-blue-600 text-white py-2 px-6 rounded-lg hover:bg-blue-700 transition">Khám phá ngay</a>
  </div>
</section>
```

- Giao diện bao gồm hai phần chính: phần banner sử dụng hình nền toàn màn hình với tiêu đề nổi bật, đoạn mô tả ngắn và nút điều hướng đến trang sản phẩm. Mục tiêu mang lại cái nhìn tổng quát, tạo ấn tượng ban đầu và điều hướng người dùng đến các chức năng chính của hệ thống một cách rõ ràng, trực quan.

Products

```
<!-- Discount Banner -->
<section class="relative bg-gradient-to-r from-blue-700 to-blue-500 text-white py-16 mb-10 overflow-hidden rounded-b-3xl shadow-lg">
  <div class="absolute inset-0 bg-black opacity-40"></div>
  <div class="relative z-10 container mx-auto px-6 text-center">
    <h2 class="text-4xl md:text-5xl font-extrabold mb-6 animate-pulse">🔥 Giảm giá 20% cho tất cả tài liệu!</h2>
    <a href="#products-list" class="inline-flex items-center bg-white text-blue-600 font-semibold px-6 py-3 rounded-full shadow hover:bg-gray-100 transition">
      <i class="fas fa-shopping-cart mr-2"></i> Mua ngay!
    </a>
  </div>
</section>

<!-- Main Container -->
<div class="container mx-auto px-6">
  <!-- Header -->
  <section>
    <div class="flex flex-col md:flex-row justify-between items-center mb-8">
      <h2 class="text-3xl font-bold text-gray-800">📚 Tất cả sản phẩm</h2>
      <div class="flex gap-4 mt-4 md:mt-0">
        <select id="sort-select" class="px-4 py-2 border-gray-300 rounded-full focus:ring-2 focus:ring-blue-500 transition" onchange="handleSortChange()">
          <option value="default">Mặc định</option>
          <option value="price-asc">Giá: Thấp đến cao</option>
          <option value="price-desc">Giá: Cao đến thấp</option>
        </select>
      </div>
    </div>
  </section>
</div>
```

- Ở phần này sẽ có phân loại danh mục và chỉnh giá từ thấp đến cao cho người tham khảo các tài liệu về chuyên của bản thân hoặc sinh có nhu cầu tìm kiếm tài liệu ôn tập

ShoppingCart

```

else
{
    <section class="py-16 bg-white">
        <div class="container mx-auto px-4">
            <h2 class="text-3xl font-bold mb-6 text-blue-600">Giỏ hàng</h2>
            <div id="cart-empty" class="text-center text-gray-600">
                <i class="fas fa-shopping-cart text-4xl mb-2"></i>
                <p>Giỏ hàng của bạn đang trống.</p>
                <a asp-controller="Home" asp-action="Index" class="btn bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700 transition mt-4 inline-b
            </div>
        </div>
    </section>
}

```

- Đoạn này cho biết giỏ hàng đang bị trống sử dụng câu lệnh if,else để phân biệt

```

<div class="flex flex-col sm:flex-row justify-between items-start sm:items-center gap-4">
    <div>
        <p id="cart-total" class="text-xl font-bold text-blue-600 mb-2">Tổng cộng: @totalCart.ToString("N0") VNĐ</p>

        @* HIỂN THỊ SỐ DƯ CỦA NGƯỜI DÙNG *@
        <p id="user-balance" class="text-lg font-semibold">
            Số dư của bạn:
            <span class="font-bold @(userBalance < totalCart ? "text-red-600" : "text-green-600")">@userBalance.ToString("N0") VNĐ</span>
        </p>
        @if (userBalance < totalCart)
        {
            <p class="text-red-600 text-sm mt-1">Số dư không đủ để thanh toán. Vui lòng nạp thêm tiền.</p>
        }
    </div>
    <div class="flex gap-4 mt-4 sm:mt-0">
        <a asp-controller="ShoppingCart" asp-action="ClearShoppingCart" id="clear-cart-btn"
            class="btn bg-red-600 text-white px-4 py-2 rounded-lg hover:bg-red-700 transition" aria-label="Xóa toàn bộ giỏ hàng">
            Xóa giỏ hàng
        </a>

        @* NÚT THANH TOÁN (FORM POST) *@
        <form asp-controller="ShoppingCart" asp-action="Checkout" method="post">
            <button type="submit" id="checkout-btn"
                class="btn bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700 transition"
                @(userBalance < totalCart ? "disabled" : "")>
                @* Vô hiệu hóa nếu số dư không đủ *@
                Thanh toán
            </button>
        </form>
    </div>
</div>

```

- Đoạn này cho biết số dư của người dùng và không thể thanh toán nếu số dư không đủ

```

@if (Model != null && Model.Any())
{
    <section class="py-16 bg-white">
        <div class="container mx-auto px-4">
            <div id="cart-items" class="space-y-4 mb-6">
                @foreach (var item in Model)
                {
                    <div class="cart-item bg-gray-50 p-4 rounded-lg flex flex-col sm:flex-row items-start sm:items-center gap-4 border" data-product-id="@item.Product.ProductId">
                        
                        <div class="flex-1">
                            <h3 class="text-lg font-semibold">@item.Product.Name</h3>
                            <p class="text-sm text-gray-600">@item.Product.Category.CategoryName</p>
                            <p class="text-red-500 font-medium">@item.Product.Price.ToString("N0") VNĐ</p>
                        </div>
                        <a asp-controller="ShoppingCart" asp-action="RemoveFromShoppingCart" asp-route-pid="@item.Product.ProductId"
                            class="btn bg-red-600 text-white px-2 py-1 rounded hover:bg-red-700 remove-item" aria-label="Xóa sản phẩm">
                            <i class="fas fa-trash"></i>
                        </a>
                    </div>
                }
            </div>
        </div>
    </section>
}

```

- Đây là đoạn có sản phẩm trong giỏ hàng

Profile

```

@* Hiển thị thông báo thành công/thất bại *@
@if (TempData["SuccessMessage"] != null)
{
    <div class="bg-green-100 border border-green-400 text-green-700 px-4 py-3 rounded relative mb-4" role="alert">
        <span class="block sm:inline">@TempData["SuccessMessage"]</span>
    </div>
}
@if (TempData["ErrorMessage"] != null)
{
    <div class="bg-red-100 border border-red-400 text-red-700 px-4 py-3 rounded relative mb-4" role="alert">
        <span class="block sm:inline">@TempData["ErrorMessage"]</span>
    </div>
}

```

- Khi thanh toán hay nạp tiền thì đều sẽ chuyển qua phần profile để nhận được thông báo là thành công hay là thất bại

```
@* LIÊN KẾT NÚT NẠP TIỀN VỚI ACTION TopUp *@
<a asp-action="TopUp" class="bg-green-600 text-white py-2 px-4 rounded-lg hover:bg-green-700 transition">Nạp tiền</a>
<a asp-action="ChangePassword" class="bg-blue-600 text-white py-2 px-4 rounded-lg hover:bg-blue-700 transition">Đổi mật khẩu</a>
<form asp-controller="Profile" asp-action="Logout" method="post" class="inline">
  <button type="submit" id="logout-btn" class="bg-red-600 text-white py-2 px-4 rounded-lg hover:bg-red-700 transition">Đăng xuất</button>
</form>
```

- Phần nạp tiền được thông qua trang profile

```
@foreach (var item in Model.PurchaseHistory)
{
  <li class="bg-gray-50 p-3 rounded-lg flex items-center justify-between shadow-sm">
    @if (!string.IsNullOrEmpty(item.ProductLink))
    {
      <a href="@item.ProductLink" target="_blank" rel="noopener noreferrer" class="text-blue-600 hover:underline font-medium">
        @item.ProductName
        <i class="fas fa-download ml-2 text-blue-500"></i>
      </a>
    }
    else
    {
      <span class="text-gray-800">@item.ProductName</span>
    }
  </li>
}
else
{
  <li class="text-gray-600 italic">Bạn chưa mua sản phẩm nào.</li>
}
```

- Khi mua thanh toán thành công sẽ chuyển đến trong profile trong đây sẽ có phần lịch sử mua hàng ảnh chính là đoạn code về phần lịch sử mua khi click vào link sẽ dẫn đến trang tài liệu

```
<a asp-action="TopUp" class="bg-green-600 text-white py-2 px-4 rounded-lg hover:bg-green-700 transition">Nạp tiền</a>
<a asp-action="ChangePassword" class="bg-blue-600 text-white py-2 px-4 rounded-lg hover:bg-blue-700 transition">Đổi mật khẩu</a>
<div class="col-md-4">
  <form asp-action="ChangePassword" method="post">
    <div asp-validation-summary="ModelOnly" class="text-danger"></div>
    <div class="form-group">
      <label asp-for="OldPassword"></label>
      <input asp-for="OldPassword" class="form-control" />
      <span asp-validation-for="OldPassword" class="text-danger"></span>
    </div>
    <div class="form-group">
      <label asp-for="NewPassword"></label>
      <input asp-for="NewPassword" class="form-control" />
      <span asp-validation-for="NewPassword" class="text-danger"></span>
    </div>
    <div class="form-group">
      <label asp-for="ConfirmNewPassword"></label>
      <input asp-for="ConfirmNewPassword" class="form-control" />
      <span asp-validation-for="ConfirmNewPassword" class="text-danger"></span>
    </div>
    <button type="submit" class="btn btn-primary mt-3">Đổi mật khẩu</button>
  </form>
</div>
```

- Đây là phần đổi mật khẩu sử dụng thẻ a để chuyển đến trang khác để thay đổi mật khẩu

```

section id="all-authors" class="py-16 bg-gray-100">
  <div class="container mx-auto px-4">
    <h2 class="text-3xl font-bold text-center mb-8 text-blue-600">Tất Cả Tác Giả</h2>
    <div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-6" id="all-authors-list">
      @foreach (var item in Model)
      {
        <div class="author-card bg-white rounded-lg shadow-md overflow-hidden transition transform hover:-translate-y-2 hover:shadow-lg">
          
          <div class="p-4">
            <h3 class="text-lg font-semibold truncate">@item.AuthorName</h3>
            <p class="text-sm text-gray-600 mb-2">@item.Detail</p>
            <a asp-controller="Author" asp-action="tacgia" asp-route-aid="@item.AuthorId" class="text-blue-600 hover:underline text-sm">Xem hồ sơ</a>
          </div>
        </div>
      }
    </div>
  </div>
</section>

```

- Đây là phần contact để người tương tác qua lại với admin

Controller

- Người dùng và admin khi vào trang sẽ thấy được phần tài liệu nổi bật, tài liệu mới và tác giả nổi bật được lấy trong database

```

3 references
public IEnumerable<Product> GetAllProducts()
{
    return db.Products;
}

2 references
public IEnumerable<Authour> noibat()
{
    return db.Authours.Where(p => p.noibat);
}

2 references
public IEnumerable<Product> Trending()
{
    return db.Products
        .Where(p => p.Trending)
        .Include(p => p.Category)
        .ToList();
}

2 references
public IEnumerable<Product> New()
{
    return db.Products
        .Where(p => p.New)
        .Include(p => p.Category)
        .ToList();
}

```

- Thêm và xóa giỏ hàng khi người dùng cần thêm hoặc xóa sản phẩm

```

0 references
public IActionResult AddToShoppingCart(int pId)
{
    var product = productRepository.GetAllProducts().FirstOrDefault(p => p.ProductId == pId);
    if (product != null)
    {
        shoppingCartRepository.AddToCart(product);
        int cartCount = shoppingCartRepository.GetAllShoppingCartItems().Count();
        HttpContext.Session.SetInt32("CartCount", cartCount);
    }

    // Chuyển hướng người dùng về trang trước đó
    if (Request.Headers["Referer"].Count > 0)
    {
        return Redirect(Request.Headers["Referer"].ToString());
    }
    return RedirectToAction("Index", "Home"); // Fallback
}

0 references
public RedirectToActionResult RemoveFromShoppingCart(int pId)
{
    var product = productRepository.GetAllProducts().FirstOrDefault(p => p.ProductId == pId);
    if (product != null)
    {
        shoppingCartRepository.RemoveFromCart(product);
        int cartCount = shoppingCartRepository.GetAllShoppingCartItems().Count();
        HttpContext.Session.SetInt32("CartCount", cartCount);
    }
    return RedirectToAction("Index");
}

```

- Người dùng cần đăng nhập trước khi thanh toán để nhận được thanh toán lấy tài liệu

```

// NEW: Action để xử lý thanh toán
[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Checkout()
{
    var user = await _userManager.GetUserAsync(User) as ApplicationUser;
    if (user == null)
    {
        // Nếu người dùng không đăng nhập, chuyển hướng đến trang đăng nhập
        return RedirectToPage("/Account/Login", new { area = "Identity" });
    }

    var shoppingCartItems = shoppingCartRepository.GetAllShoppingCartItems();
    if (!shoppingCartItems.Any())
    {
        TempData["ErrorMessage"] = "Giỏ hàng của bạn đang trống!";
        return RedirectToAction("Index");
    }

    decimal totalCart = shoppingCartRepository.GetShoppingCartTotal();
}

```

- Người dùng khi cần liên hệ hoặc trao đổi với admin thì thông qua liên hệ được lưu trữ trong database khi admin đăng nhập sẽ nhận được tin nhắn người gửi

```

// references
public async Task<IActionResult> Send(Contact model)
{
    if (ModelState.IsValid)
    {
        var contactMessage = new Contact
        {
            Name = model.Name,
            Email = model.Email,
            Message = model.Message,
        };

        db.Contacts.Add(contactMessage);
        await db.SaveChangesAsync();

        ViewBag.SuccessMessage = "Tin nhắn của bạn đã được gửi thành công. Chúng tôi sẽ liên hệ lại với bạn sớm nhất!";

        ModelState.Clear();
        return View("Contact", new Contact());
    }

    ViewBag.ErrorMessage = "Vui lòng kiểm tra lại thông tin bạn đã nhập.";
    return View("Contact", model);
}

```

- Người dùng cần xem trước tài liệu của sản phẩm để xem có phù hợp với bản thân không thì click vào phần xem chi tiết thì bên trong có thông tin của sản phẩm và xem trước tài liệu sẽ có 5 slide để cho người dùng xem trước

```

// references
public IActionResult Detail(int id)
{
    var product = contextDB.Products
        .Include(p => p.Category)
        .FirstOrDefault(p => p.ProductId == id);

    if (product == null)
    {
        return NotFound();
    }

    return View(product);
}

```

- Người dùng muốn đổi mật khẩu để bảo mật thông tin cá nhân

```

// references
public class ChangePasswordViewModel
{
    [Required(ErrorMessage = "Mật khẩu cũ là bắt buộc.")]
    [DataType(DataType.Password)]
    [Display(Name = "Mật khẩu cũ")]
    4 references
    public string OldPassword { get; set; } // This is correct

    [Required(ErrorMessage = "Mật khẩu mới là bắt buộc.")]
    [StringLength(100, ErrorMessage = "{0} phải dài ít nhất {2} và tối đa {1} ký tự.", MinimumLength = 6)]
    [DataType(DataType.Password)]
    [Display(Name = "Mật khẩu mới")]
    4 references
    public string NewPassword { get; set; } // This is correct

    [DataType(DataType.Password)]
    [Display(Name = "Xác nhận mật khẩu mới")]
    [Compare("NewPassword", ErrorMessage = "Mật khẩu mới và mật khẩu xác nhận không khớp.")]
    3 references
    public string ConfirmNewPassword { get; set; } // This is correct
}

```

- Phân quyền chỉ có admin mới có quyền truy cập để thêm,sửa,xóa các mục

```
[Area("Admin")]  
[Authorize(Roles = "Admin")] // Chỉ Admin mới có quyền truy cập
```

2.5 Triển khai và cài đặt

Cấu hình máy phát triển:

- **Hệ điều hành:** Windows 11 Pro 64-bit
- **Bộ xử lý (CPU):** Intel Core i5-12400
- **Bộ nhớ RAM:** 16 GB DDR4
- **Card đồ họa (GPU):** NVIDIA GeForce GTX 1650
- **Công cụ phát triển:** Visual Studio 2022 Community
- **Trình duyệt kiểm thử:** Google Chrome, Microsoft Edge

Môi trường phát triển:

- **Ngôn ngữ lập trình:** C#
- **Kiến trúc ứng dụng:** ASP.NET Core MVC
- **Giao diện người dùng:** Razor View Engine
- **Quản lý cơ sở dữ liệu:** Entity Framework Core (Code First)
- **Hệ quản trị CSDL:** Microsoft SQL Server LocalDB
- **Xác thực & phân quyền:** ASP.NET Core Identity
- **Thiết kế giao diện:** Bootstrap, Tailwind CSS
- **Biểu tượng giao diện:** Font Awesomes
- **Hiệu ứng & tương tác:** CSS, JavaScript
- **Bộ SDK:** .NET 8.0 SDK

Quy trình triển khai hệ thống bao gồm các bước:

1. **Khởi tạo project:** Tạo mới project ASP.NET Core MVC trong Visual Studio 2022, lựa chọn giao diện Razor và tích hợp ASP.NET Core Identity.
2. **Quản lý gói:** Thêm các thư viện cần thiết qua NuGet như Microsoft.EntityFrameworkCore.SqlServer, Microsoft.AspNetCore.Identity.EntityFrameworkCore, Microsoft.EntityFrameworkCore.Tools,...
3. **Tạo mô hình dữ liệu:** Xây dựng các lớp model đại diện cho thực thể nghiệp vụ như sản phẩm, loại tài liệu, tác giả, đơn hàng,...
4. **Cấu hình DbContext:** Định nghĩa lớp ApplicationDbContext và đăng ký trong Program.cs, đồng thời cấu hình chuỗi kết nối trong appsettings.json.
5. **Tạo cơ sở dữ liệu:** Sử dụng lệnh Add-Migration để tạo các migration cần thiết và Update-Database để khởi tạo CSDL theo mô hình Code First.
6. **Viết logic nghiệp vụ:** Tạo các controller phục vụ cho việc hiển thị danh sách sản phẩm, xử lý giỏ hàng, đặt hàng, quản trị,...

7. **Xây dựng giao diện:** Sử dụng Razor kết hợp Tailwind CSS và Bootstrap để xây dựng các view tương ứng với từng chức năng.
8. **Chạy và kiểm thử:** Sử dụng Visual Studio để chạy ứng dụng và kiểm thử toàn bộ chức năng trên trình duyệt.

Chương 3 Kết quả chương trình

3.1 Người dùng(User)

3.1.1 Người dùng chưa có tài khoản

Đăng ký

Tên người dùng

Email

Mật khẩu

Xác nhận mật khẩu

[Đã có tài khoản? Đăng nhập](#)

Đăng ký

Hình 3.1: Đăng kí

3.1.2 Đăng nhập

Đăng nhập vào StudyDocs

Email



Email của bạn

The Email field is required.

Mật khẩu



Mật khẩu của bạn

The Password field is required.

☐

Ghi nhớ đăng nhập

[Quên mật khẩu?](#)

Đăng nhập

[Chưa có tài khoản? Đăng ký ngay](#)

Hình 3.2: Đăng nhập

3.1.3 Trang chủ



Về StudyDocs

Sứ mệnh

Cung cấp nền tảng chia sẻ tài liệu học tập chất lượng, giúp sinh viên và giáo viên tiếp cận kiến thức dễ dàng.

Tầm nhìn

Trở thành nền tảng học tập trực tuyến hàng đầu tại Việt Nam, kết nối cộng đồng học thuật toàn quốc.

Giá trị cốt lõi

- Chất lượng tài liệu được đảm bảo.
- Hỗ trợ cộng đồng học tập 24/7.
- Giá cả hợp lý, thanh toán linh hoạt.

Danh mục tài liệu



Toán học

Giải tích, Đại số, Xác suất...

[Xem tài liệu](#)



Lập trình

Python, Java, C++...

[Xem tài liệu](#)



Kinh tế

Quản trị, Kinh tế vi mô...

[Xem tài liệu](#)



Kỹ thuật

Cơ khí, Điện tử...

[Xem tài liệu](#)



Ngôn ngữ học

Ngữ pháp, tweet pháp, Ngữ nghĩa học...

[Xem tài liệu](#)



Quản trị kinh doanh

Chiến lược, Marketing...

[Xem tài liệu](#)

Tài liệu mới nhất



Giáo trình Lập trình Python ...
750,000 đ
Lập trình
Thêm vào giỏ
Xem chi tiết

Combo ôn thi đại học khối ...
1,000,000 đ
Combo thi THPT
Thêm vào giỏ
Xem chi tiết

Đồ án Thiết kế giao diện ng...
100,000 đ
Lập trình
Thêm vào giỏ
Xem chi tiết

Giáo trình học tiếng Hàn sơ...
250,000 đ
Ngôn ngữ học
Thêm vào giỏ
Xem chi tiết

Xem tất cả tài liệu

Tài liệu nổi bật

Combo ôn thi đại học khối ...
1,000,000 đ
Combo thi THPT
Thêm vào giỏ
Xem chi tiết

Đồ án Thiết kế giao diện ng...
100,000 đ
Lập trình
Thêm vào giỏ
Xem chi tiết

Giáo trình học tiếng Hàn sơ...
250,000 đ
Ngôn ngữ học
Thêm vào giỏ
Xem chi tiết

Marketing căn bản
160,000 đ
Marketing
Thêm vào giỏ
Xem chi tiết

Tác giả nổi bật

Nguyễn Ngọc Vy
Chuyên gia kinh tế với 10 năm kinh nghiệm giảng dạy.
Tài liệu: ?
[Xem hồ sơ](#)

Anh Ngọc Vy
Giảng viên lập trình, tác giả nhiều tài liệu về JavaScript..
Tài liệu: ?
[Xem hồ sơ](#)

Vy DZ
Tiến sĩ Toán học, chuyên cung cấp tài liệu Toán cao cấp.
Tài liệu: ?
[Xem hồ sơ](#)

Soài Ca Cam Ranh
Kỹ sư cơ khí, tác giả tài liệu kỹ thuật nổi tiếng.
Tài liệu: ?
[Xem hồ sơ](#)

[Xem tất cả tác giả](#)

StudyDocs
Nền tảng chia sẻ tài liệu học tập chất lượng cao, hỗ trợ sinh viên và giảng viên trên toàn quốc.
Email: support@studycdocs.vn
Hotline: +84 122 456 789

Liên kết nhanh
Trang chủ
Sản phẩm
Liên hệ

Hỗ trợ
Câu hỏi thường gặp
Chính sách bảo mật
Điều khoản sử dụng
Hướng dẫn thanh toán

Kết nối với chúng tôi

Hình 3.3: Trang chủ

3.1.4 Chi tiết tài liệu



**Combo ôn thi đại học khối A:
Toán – Lý – Hóa**

1,000,000 VNĐ

Danh mục: Combo thi THPT

Đề thi thử, tài liệu công thức, bí kíp luyện đề, sơ đồ tư duy.

[Thêm vào giỏ](#) [Mua ngay](#)

Hình 3.4: Chi tiết sản phẩm

- Dưới chi tiết tài liệu có phần xem trước tài liệu để người dùng có thể tham khảo trước

Xem trước tài liệu

UX/UI là viết tắt của User Experience/User Interface, là hai lĩnh vực thuật yếu trong việc phát triển sản phẩm kỹ thuật số, tập trung vào việc tạo ra trải nghiệm người dùng (UX) tích cực và giao diện người dùng (UI) đẹp mắt, dễ sử dụng.

1. UX (Trải nghiệm người dùng)

- Tập trung vào mặt chức năng của sản phẩm, đảm bảo sản phẩm đáp ứng nhu cầu và mong muốn của người dùng một cách hiệu quả, dễ dàng và thú vị.
- Các công việc chính của UX Designer bao gồm:
 - Nghiên cứu người dùng: Khảo sát, phỏng vấn, thu thập dữ liệu để hiểu hành vi, nhu cầu và mong muốn của người dùng.
 - Lập bản đồ hành trình người dùng: Xác định các bước người dùng thực hiện khi sử dụng sản phẩm và xác định điểm cần cải thiện.
 - Thiết kế kiến trúc thông tin: Sắp xếp và phân loại nội dung sản phẩm một cách logic để người dùng dễ dàng tìm kiếm và truy cập.
 - Thiết kế tương tác: Thiết kế các yếu tố giao diện mà người dùng tương tác, đảm bảo sự trực quan và dễ sử dụng.
 - Kiểm tra khả năng sử dụng: Thử nghiệm sản phẩm với người dùng thực tế để xác định và sửa lỗi.

2. UI (Giao diện người dùng)

- Tập trung vào mặt thẩm mỹ của sản phẩm, tạo ra giao diện đẹp mắt, thu hút và truyền tải thông tin một cách rõ ràng.
- Các công việc chính của UI Designer bao gồm:
 - Thiết kế giao diện: Tạo ra các yếu tố hình ảnh như bố cục, màu sắc, typography, hình ảnh, biểu tượng, ... đảm bảo tính thẩm mỹ và nhất quán.
 - Thiết kế hệ thống thiết kế: Tạo ra bộ hướng dẫn thiết kế quy định các nguyên tắc sử dụng màu sắc, typography, bố cục, ... để đảm bảo tính nhất quán cho toàn bộ sản phẩm.
 - Tạo nguyên mẫu: Tạo ra các phiên bản mô phỏng của sản phẩm để thử nghiệm và thu thập phản hồi từ người dùng.

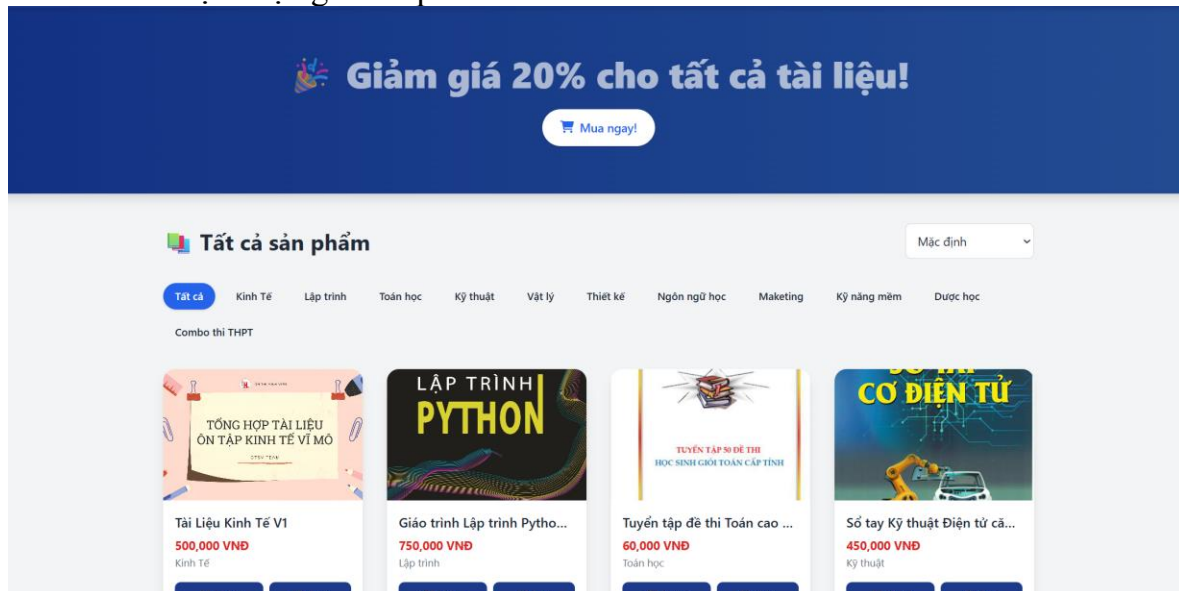
3. Kỹ năng cần thiết cho ngành UX/UI

a/ Kỹ năng mềm:

- Kỹ năng giao tiếp: Giao tiếp hiệu quả với các bên liên quan như kỹ sư, nhà phát triển sản phẩm, và người dùng.
- Kỹ năng tư duy phân tích: Phân tích vấn đề, đưa ra giải pháp sáng tạo và tối ưu.
- Kỹ năng làm việc nhóm: Hợp tác hiệu quả với các thành viên trong nhóm để hoàn thành dự án.

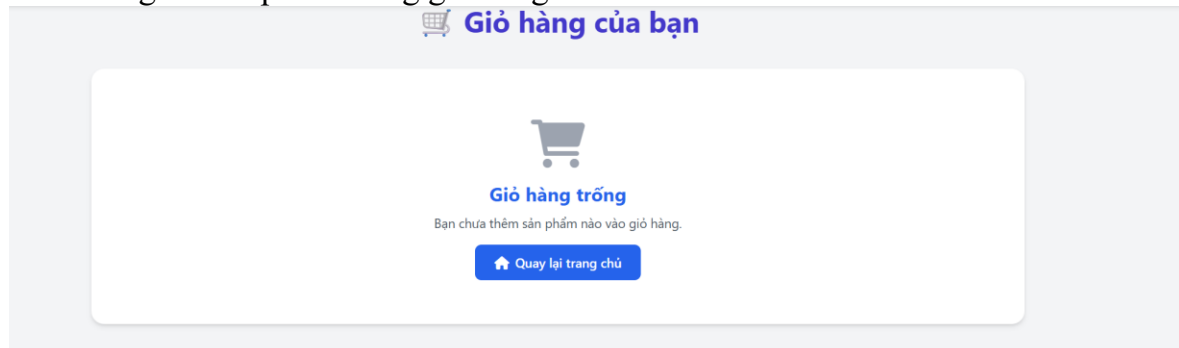
Hình 3.5: Tham khảo

3.1.5 Danh mục/ Lọc giá sản phẩm



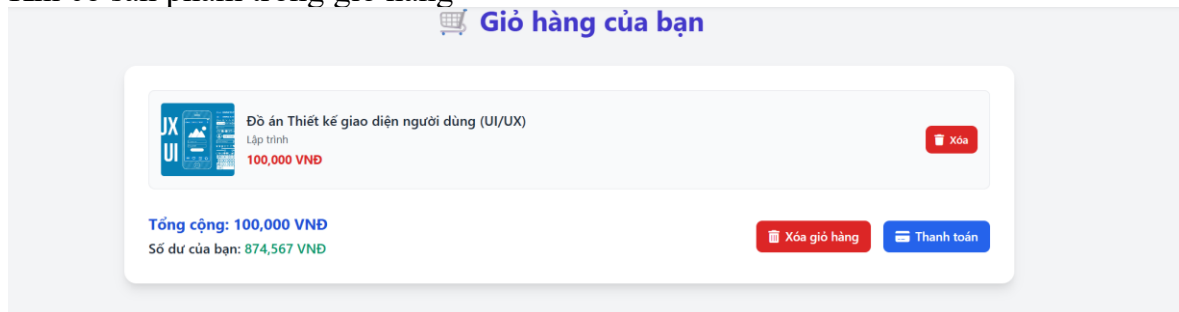
3.6 Giỏ hàng

Khi không có sản phẩm trong giỏ hàng



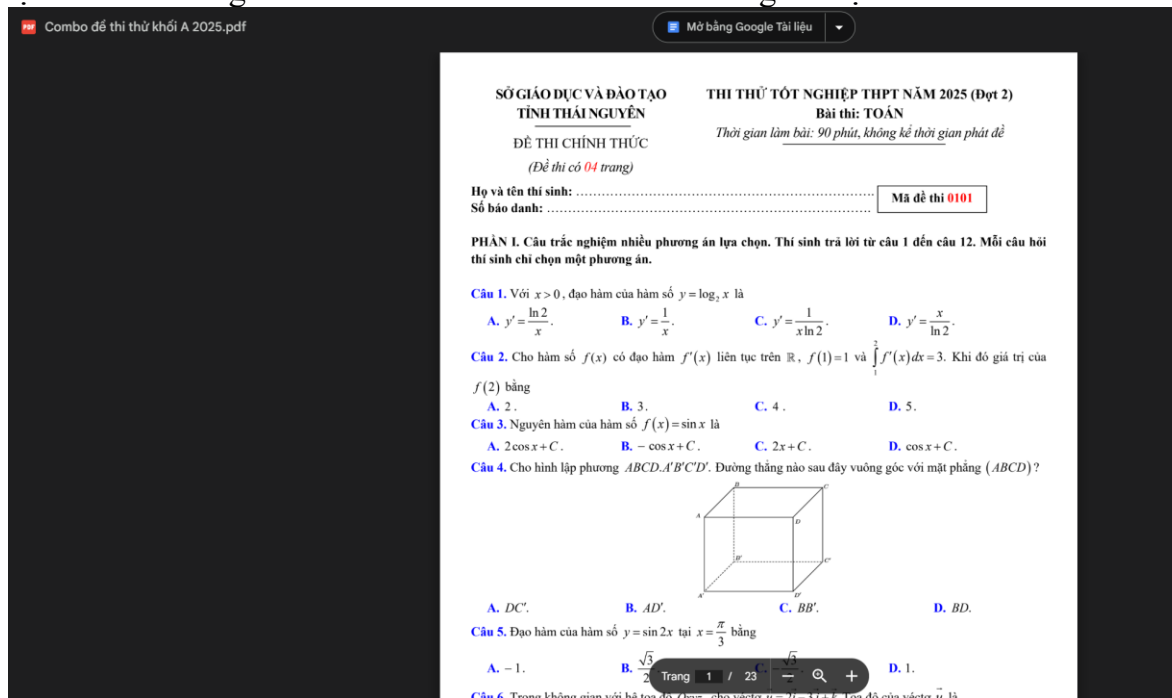
Hình 3.7: Giỏ hàng trống

Khi có sản phẩm trong giỏ hàng



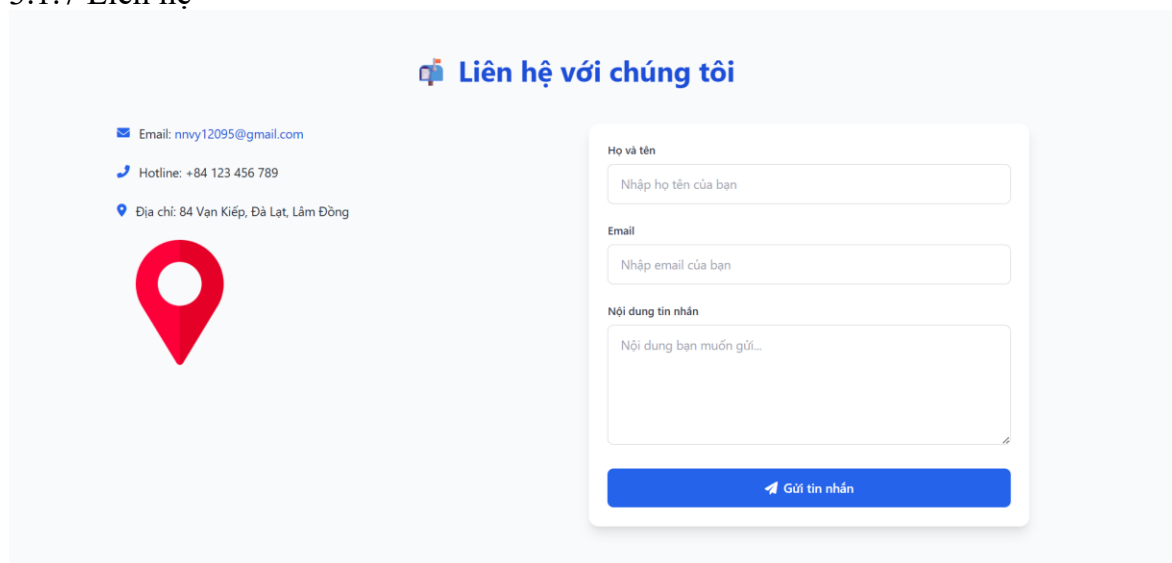
Hình 3.8: Giỏ hàng có sản phẩm

Sau khi thanh toán thành công sẽ dẫn đến trang hồ sơ người dùng trong đó có phần lịch sử mua hàng khi click vào link thì sẽ dẫn đến trang tài liệu



Hình 3.9: Sản phẩm

3.1.7 Liên hệ



Hình 3.10: Liên hệ

3.1.8 Profile

The screenshot shows a user profile page with a header 'Hồ sơ người dùng' and a user icon. The main content is divided into two columns. The left column, titled 'Thông tin cá nhân', displays the user's email '12@gmail.com' and a balance of '874,567 VND'. It includes two buttons: 'Nạp tiền' (Deposit) and 'Đăng xuất' (Logout). The right column, titled 'Lịch sử mua hàng' (Purchase History), lists four items: 'Tuyển tập đề thi Toán cao cấp A1', 'Đề cương ôn tập Cơ sở dữ liệu', 'Combo ôn thi đại học khối A: Toán – Lý – Hóa', and 'Đồ án Thiết kế giao diện người dùng (UI/UX)'. Each item has a download icon.

Hình 3.11: Hồ sơ người dùng

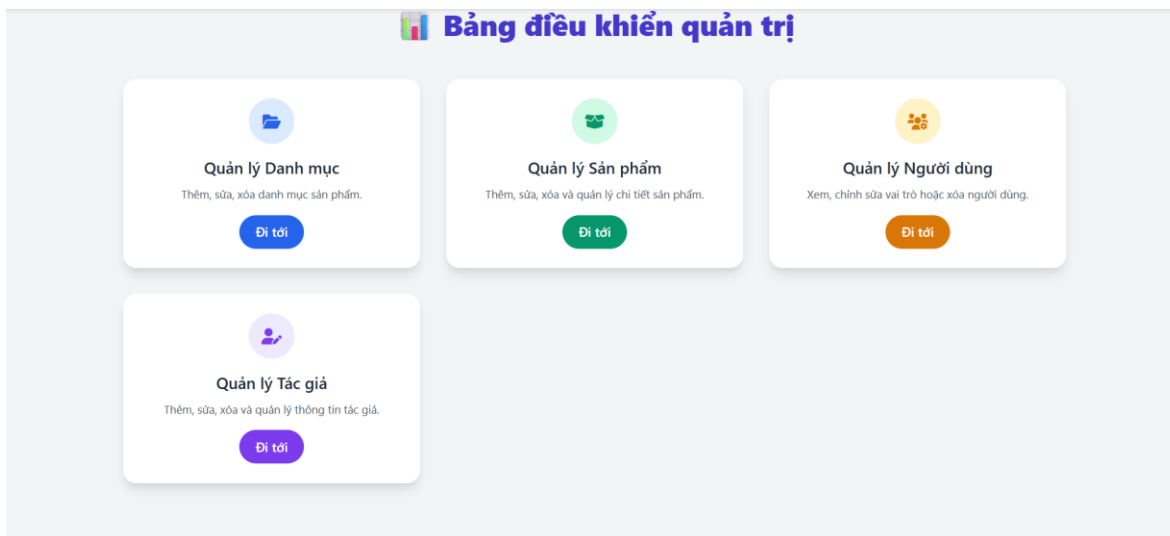
3.1.9 Đổi mật khẩu

The screenshot shows a password change form titled 'Đổi mật khẩu'. It contains three input fields: 'Mật khẩu cũ' (Old Password), 'Mật khẩu mới' (New Password), and 'Xác nhận mật khẩu' (Confirm Password). Each field has a red error message below it: 'Mật khẩu cũ là bắt buộc.' (Old Password is required.), 'Mật khẩu mới là bắt buộc.' (New Password is required.), and 'The Xác nhận mật khẩu mới field is required.' (The Confirm New Password field is required.). A blue button labeled 'Cập nhật mật khẩu' (Update Password) is at the bottom.

Hình 3.12: Đổi mật khẩu

3.2 Quản trị(Admin)

3.2.1 Dashboard



Hình 3.13: Bảng điều khiển

3.2.2 Quản lý danh mục

<div> Quản lý Danh mục </div> <div> + Thêm Danh mục mới </div>	
TÊN DANH MỤC	THAO TÁC
Combo thi THPT	Sửa Xóa
Dược học	Sửa Xóa
Kinh Tế	Sửa Xóa
Kỹ năng mềm	Sửa Xóa
Kỹ thuật	Sửa Xóa
Lập trình	Sửa Xóa
Marketing	Sửa Xóa
Ngôn ngữ học	Sửa Xóa
Thiết kế	Sửa Xóa
Toán học	Sửa Xóa

Hình 3.14: Quản lí danh mục

3.2.3 Quản lý sản phẩm

Quản lý Tài liệu							
Tổng cộng: 22 tài liệu				Thêm Tài liệu			
ẢNH	TÊN TÀI LIỆU	DANH MỤC	TÁC GIẢ	GIÁ	MỚI	NỔI BẬT	HÀNH ĐỘNG
	Tài Liệu Kinh Tế V1	Kinh Tế	Nguyễn Ngọc Vỹ	500.000 đ	Không	Không	Chi tiết Sửa Xóa
	Giáo trình Lập trình Python cơ bản	Lập trình	Anh Ngọc Vỹ	750.000 đ	Có	Không	Chi tiết Sửa Xóa
	Tuyển tập đề thi Toán cao cấp A1	Toán học	Vỹ DZ	60.000 đ	Không	Không	Chi tiết Sửa Xóa
	Sổ tay Kỹ thuật Điện tử căn bản	Kỹ thuật	Soái Ca Cam Ranh	450.000 đ	Không	Không	Chi tiết Sửa Xóa
	Bài tập Vật lý Đại cương A	Vật lý	Nguyễn Ngọc Vỹ	550.000 đ	Không	Không	Chi tiết Sửa Xóa
	Bộ đề thi thử THPT Quốc Gia 2024 môn Toán	Toán học	Soái Ca Cam Ranh	200.000 đ	Không	Không	Chi tiết Sửa Xóa
	Tổng hợp công thức Vật lý 12	Vật lý	Anh Ngọc Vỹ	1.000.000 đ	Không	Không	Chi tiết Sửa Xóa

Hình 3.15: Quản lí sản phẩm

3.2.4 Quản lý người dùng

Quản lý Người dùng			
TÊN ĐĂNG NHẬP	EMAIL	VAI TRÒ	THAO TÁC
admin@gmail.com	admin@gmail.com	Admin	Chi tiết Xóa
12@gmail.com	12@gmail.com		Chi tiết Xóa

Hình 3.16: Quản lí người dùng

3.2.5 Quản lý tác giả

<div>  Quản lý Tác giả </div>			
<div> <div>+</div> Thêm Tác giả mới </div>			
ẢNH	TÊN TÁC GIẢ	NỔI BẬT	HÀNH ĐỘNG
	Nguyễn Ngọc Vỹ	✓ Có	Chi tiết Sửa Xóa
	Anh Ngọc Vỹ	✓ Có	Chi tiết Sửa Xóa
	Vỹ DZ	✓ Có	Chi tiết Sửa Xóa
	Soái Ca Cam Ranh	✓ Có	Chi tiết Sửa Xóa

Hình 3.17: Quản lý tác giả

Chương 4 : Kết luận

4.1 Tổng kiến thức đạt được

- Hiểu và vận dụng mô hình MVC (Model – View – Controller) để tổ chức cấu trúc ứng dụng rõ ràng, dễ bảo trì và mở rộng.

- Thành thạo ASP.NET Core MVC trong việc xây dựng website động, từ định tuyến (routing), xử lý logic cho đến hiển thị dữ liệu bằng Razor View.
- Áp dụng Entity Framework Core (Code First) để tạo và quản lý cơ sở dữ liệu, từ đó thao tác dữ liệu hiệu quả thông qua các lớp mô hình (model).
- Tích hợp ASP.NET Core Identity nhằm xử lý chức năng đăng ký, đăng nhập và phân quyền người dùng một cách bảo mật và hiệu quả.
- Sử dụng Bootstrap, Tailwind CSS, JavaScript và AJAX để xây dựng giao diện đẹp mắt, responsive và cải thiện trải nghiệm người dùng.
- Làm việc thành thạo với Visual Studio và SQL Server, đồng thời rèn luyện kỹ năng debug, tổ chức dự án, viết mã sạch và dễ hiểu.
- Tư duy thiết kế hệ thống và tổ chức mã nguồn được cải thiện rõ rệt qua việc phân chia lớp, cấu trúc thư mục và xử lý logic theo luồng dữ liệu thực tế.

Ngoài ra, đồ án còn giúp mình nâng cao khả năng tự học, tìm kiếm và ứng dụng tài liệu từ nhiều nguồn khác nhau. Từ việc xử lý lỗi, cải tiến giao diện đến tối ưu chức năng, mình đã dần hình thành tư duy giải quyết vấn đề như một lập trình viên chuyên nghiệp. Đây là bước đệm quan trọng để mình phát triển hơn trong các dự án thực tế sau này.

4.2 Điểm tồn tại

Mặc dù đồ án đã hoàn thành với nhiều chức năng cơ bản và giao diện thân thiện, nhưng vẫn còn tồn tại một số điểm hạn chế cần cải thiện trong tương lai, cụ thể như sau:

- Phần quản trị còn đơn giản, chưa có tính năng phân quyền chi tiết hoặc thống kê dữ liệu nâng cao (số lượt tải, doanh thu, tài liệu phổ biến...).
- Chưa tích hợp lưu trữ thực file tài liệu (DOCX...), mà mới chỉ lưu đường dẫn hoặc mô tả, gây hạn chế trong việc bảo mật hoặc kiểm soát truy cập.
- Giao diện tuy hiện đại nhưng chưa tối ưu hoàn toàn cho thiết bị di động và trình duyệt nhỏ.
- Chưa có chức năng thống kê doanh thu, số lượt tải hoặc tổng số tài liệu theo thời gian cho admin.
- Chưa tích hợp hệ thống chat hỗ trợ khách hàng hoặc phản hồi trực tuyến.
- Hệ thống email chưa được tích hợp để gửi thông báo khi tài liệu được duyệt, khi người dùng đăng ký hoặc mua tài liệu.

4.3 Hướng mở rộng đề tài

- Tích hợp các cổng thanh toán điện tử qua nhiều ngân hàng hoặc ví điện tử (VNPay, Momo, ZaloPay...) để hỗ trợ giao dịch trực tuyến nhanh chóng và tiện lợi.
- Tự động gửi email thông báo sau khi người dùng đăng ký, đặt hàng hoặc có thay đổi trạng thái tài khoản, nhằm nâng cao trải nghiệm và kết nối người dùng.
- Tối ưu giao diện cho thiết bị di động, đảm bảo hiển thị mượt mà trên các loại màn hình nhỏ và hỗ trợ người dùng thao tác dễ dàng ở mọi nơi.
- Xây dựng hệ thống thống kê doanh thu trực quan, giúp quản trị viên theo dõi doanh số, tài liệu bán chạy và tình hình hoạt động của hệ thống.
- Phân quyền người dùng nâng cao: Mở rộng phân quyền chi tiết hơn như: người dùng thường, tác giả tài liệu, cộng tác viên, kiểm duyệt viên... để kiểm soát nội dung và nâng cao tính minh bạch, an toàn trên nền tảng.
- Hệ thống đánh giá và phản hồi: Cho phép người mua để lại nhận xét, đánh giá chất lượng tài liệu sau khi tải về. Tính năng này góp phần tạo ra hệ sinh thái minh bạch, thúc đẩy người bán cải thiện chất lượng sản phẩm.
- Ứng dụng trí tuệ nhân tạo (AI): Đề xuất tài liệu phù hợp cho người dùng dựa trên hành vi tìm kiếm và lịch sử mua hàng, giúp cá nhân hóa trải nghiệm mua sắm và tăng khả năng bán hàng.

Tài liệu tham khảo

- [1] Microsoft Docs – *ASP.NET Core MVC overview* – Microsoft – 2024.
<https://learn.microsoft.com/en-us/aspnet/core/mvc/overview>
- [2] Adam Freeman – *Pro ASP.NET Core MVC 2* – Apress – 2017.
<https://link.springer.com/book/10.1007/978-1-4842-3243-2>
- [3] Microsoft Learn – *Entity Framework Core Documentation* – Microsoft – 2023.
<https://learn.microsoft.com/en-us/ef/core/>
- [4] Xây dựng website với ASP.NET Core MVC – Ths. Nguyễn Đức Tấn – 2025 Đại học Yersin (Bài giảng nội bộ)
- [5] Bootstrap – *Official Bootstrap 5 Documentation* – getbootstrap.com – 2023.
<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

PHỤ LỤC