

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



YERSIN UNIVERSITY

**BÁO CÁO MÔN HỌC
LẬP TRÌNH WEB 1**

**Thiết Kế Website Bán Quần
Áo Thời Trang**

GVHD	: Nguyễn Đức Tấn
SVTH	: Ksor Phuk
Mã số SV	: 2301010014
Khóa học	: 2024 - 2025

Đà Lạt, tháng 06- 2025

PHẦN NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm:

Ngày ... tháng ... năm...

Ký và ghi rõ họ tên

MỤC LỤC

LỜI NÓI ĐẦU	6
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 Core	9
CHƯƠNG 2: XÂY DỰNG ỨNG DỤNG WEBSITE BÁN QUẦN ÁO THỜI TRANG ...	11
2.1 Phát biểu bài toán ứng dụng	11
2.1.1 Giới thiệu dự án	11
2.1.2 Vai trò của dự án.....	11
2.1.3 Ý nghĩa của dự án	12
2.2 Phân tích yêu cầu của ứng dụng	12
2.2.1 Yêu cầu chức năng	12
2.2.2 Yêu cầu phi chức năng	13
2.3 Thiết kế cơ sở dữ liệu.....	14
2.3.1 Mô tả các bảng dữ liệu	14
2.3.2 Các mối quan hệ.....	18
2.3.3 Mô hình ERD	19
2.4 Thiết kế giao diện người dùng:.....	20
2.5 Thiết kế các thành phần MVC:	24
2.6 Triển khai và cài đặt.....	28
2.6.1 Cấu hình máy phát triển	28
2.6.2 Môi trường phát triển	28
2.6.3 Quy trình cài đặt và triển khai	29
CHƯƠNG 3: KẾT QUẢ CHƯƠNG TRÌNH	30
Kết luận.....	35
Tài liệu tham khảo	36

DANH MỤC HÌNH

Hình ảnh	Mô tả	Trang
Hình 1.1	Mô hình MVC	8
Hình 2.1	Sơ đồ user case	13
Hình 2.2	Mô hình ERD cơ sở dữ liệu	19
Hình 2.3	Bản wireframe trang chủ	21
Hình 2.4	Bản wireframe cửa hàng	22
Hình 2.5	Bản wireframe cửa hàng	23
Hình 2.6	Bản wireframe bài báo	24
Hình 3.1	Giao diện trang chủ	31-33
Hình 3.2	Giao diện cửa hàng	33
Hình 3.3	Giao diện bài báo	34
Hình 3.4	Giao diện giỏ hàng	34

DANH MỤC TỪ NGỮ VIẾT TẮT

STT	Tên viết tắt	Tên đầy đủ	Dịch ra tiếng Việt (Nếu là tiếng nước ngoài)
1	MVC	Model – View - Controller	
2	ASP.NET	Active Server Pages .NET	
3	HTML	HyperText Markup Language	
4	CSS	Cascading Style Sheets	
5	HTTP	HyperText Transfer Protocol	
6	JS	JavaScript	
7	ID	Identifier	Định danh
8	API	Application Programming Interface	
9	INT	Interger	
10	SQL	Structured Query Language	

LỜI NÓI ĐẦU

Trong bối cảnh công nghệ thông tin phát triển không ngừng, thương mại điện tử đã trở thành một phần không thể thiếu trong đời sống hiện đại, đặc biệt trong lĩnh vực thời trang – nơi nhu cầu mua sắm trực tuyến ngày càng gia tăng. Quần áo thời trang không chỉ đáp ứng nhu cầu cơ bản mà còn là cách để mỗi cá nhân khẳng định phong cách và bản sắc riêng. Tuy nhiên, nhiều nền tảng bán hàng trực tuyến hiện nay vẫn còn hạn chế về tính thân thiện, trải nghiệm người dùng và sự đa dạng trong sản phẩm. Vì vậy, em đã lựa chọn đề tài “Thiết kế website bán quần áo thời trang” nhằm xây dựng một hệ thống trực tuyến hiện đại, tiện lợi và đáp ứng tốt nhu cầu của người dùng.

Dự án được phát triển dựa trên ASP.NET Core MVC – một nền tảng mạnh mẽ của Microsoft, nổi bật với cấu trúc Model – View – Controller, giúp mã nguồn rõ ràng, dễ dàng mở rộng và bảo trì. Trong quá trình thực hiện, em tập trung vào việc tạo ra một giao diện người dùng tối giản, trực quan và mang tính thẩm mỹ cao, phù hợp với xu hướng hiện đại. Các tính năng cốt lõi như tìm kiếm sản phẩm, phân loại theo danh mục, quản lý giỏ hàng, thanh toán trực tuyến và theo dõi đơn hàng được thiết kế để tối ưu hóa trải nghiệm, đảm bảo người dùng có thể thao tác dễ dàng trên mọi thiết bị, từ máy tính đến điện thoại di động. Việc tích hợp các thư viện như Bootstrap và thiết kế responsive giúp website thích nghi tốt với nhiều độ phân giải màn hình, mang lại sự tiện lợi và nhất quán.

Để hoàn thiện dự án, em đã áp dụng các kiến thức về lập trình web, quản lý cơ sở dữ liệu với SQL Server, thiết kế giao diện và xử lý logic nghiệp vụ, đồng thời tìm hiểu thêm các công cụ hỗ trợ để nâng cao chất lượng hệ thống. Báo cáo này sẽ trình bày chi tiết các giai đoạn phát triển, từ phân tích yêu cầu, thiết kế, đến triển khai và kiểm thử, đồng thời chia sẻ những thách thức em đã đối mặt và cách giải quyết. Em kỳ vọng website này không chỉ là một sản phẩm học thuật mà còn mang giá trị thực tiễn, góp phần cung cấp một giải pháp mua sắm trực tuyến hiệu quả. Em xin chân thành cảm ơn sự hướng dẫn tận tình của quý thầy cô, những người đã hỗ trợ em hoàn thành đề tài này một cách trọn vẹn.

CHƯƠNG 1: TÌM HIỂU VỀ LẬP TRÌNH WEB MVC (CƠ SỞ LÝ THUYẾT)

1.1 Khái niệm

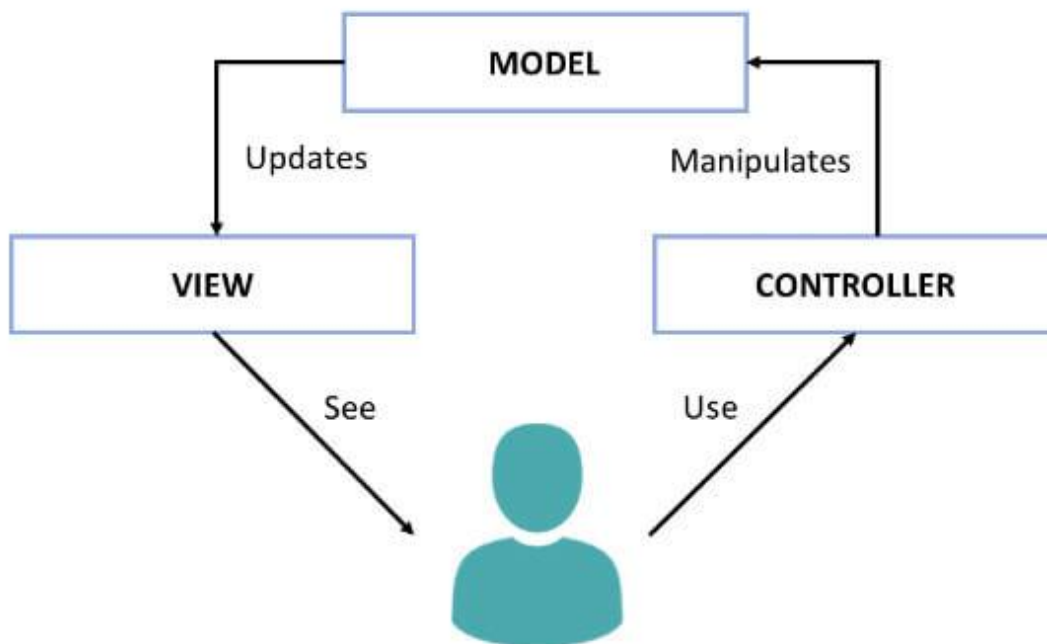
Trong lĩnh vực phát triển phần mềm, đặc biệt là lập trình web, mô hình kiến trúc đóng vai trò rất quan trọng trong việc tổ chức cấu trúc ứng dụng, phân chia nhiệm vụ và quản lý mã nguồn hiệu quả. Một trong những mô hình phổ biến và được ưa chuộng hiện nay là mô hình MVC – viết tắt của Model, View và Controller. Đây là mô hình hướng cấu trúc, có nhiệm vụ tách biệt ba phần chính trong một ứng dụng web: dữ liệu, giao diện và điều khiển logic.

Model: đại diện cho tầng dữ liệu và logic nghiệp vụ của hệ thống. Nó chứa các lớp xử lý dữ liệu, truy xuất cơ sở dữ liệu, các phương thức thao tác với dữ liệu như thêm, sửa, xóa và tìm kiếm. Model là nơi xử lý các quy tắc nghiệp vụ và không liên quan đến cách dữ liệu được hiển thị.

View: là tầng giao diện, có nhiệm vụ hiển thị dữ liệu lên trình duyệt cho người dùng cuối. View nhận dữ liệu từ Controller và hiển thị chúng theo đúng định dạng giao diện đã thiết kế. Trong ứng dụng ASP.NET, View thường được viết bằng cú pháp Razor trong các file.cshtml.

Controller: đóng vai trò là trung gian giữa Model và View. Khi người dùng gửi yêu cầu thông qua giao diện (ví dụ nhấp nút hoặc nhập dữ liệu), Controller sẽ tiếp nhận yêu cầu, xử lý logic, gọi đến Model nếu cần, sau đó truyền dữ liệu đến View để hiển thị kết quả.

Mô hình MVC được thiết kế nhằm tạo ra một kiến trúc phân tầng rõ ràng, giúp lập trình viên dễ dàng quản lý, phát triển, bảo trì và mở rộng ứng dụng web, đặc biệt là những dự án có quy mô lớn hoặc làm việc theo nhóm.



1.2 Đặc điểm

Mô hình MVC sở hữu nhiều đặc điểm nổi bật khiến nó trở thành một lựa chọn tối ưu trong phát triển ứng dụng web hiện đại:

Tách biệt mối quan hệ giữa giao diện và logic xử lý: Việc chia rõ ba phần giúp nhà phát triển dễ dàng chỉnh sửa hoặc nâng cấp từng phần mà không ảnh hưởng đến phần còn lại. Ví dụ, có thể thay đổi giao diện View mà không cần sửa lại phần xử lý dữ liệu.

Dễ bảo trì, dễ mở rộng: Khi ứng dụng cần nâng cấp hoặc thêm chức năng mới, lập trình viên chỉ cần can thiệp vào thành phần tương ứng. Điều này làm giảm đáng kể nguy cơ gây lỗi và giúp tiết kiệm thời gian phát triển.

Hỗ trợ phát triển theo nhóm hiệu quả: Mỗi thành viên có thể đảm nhận một vai trò cụ thể, ví dụ người thiết kế giao diện sẽ làm View, người xử lý nghiệp vụ làm Model, và người điều hướng tổng thể làm Controller.

Tăng khả năng tái sử dụng mã nguồn: Các thành phần có thể được tái sử dụng ở nhiều nơi, ví dụ như Model có thể dùng chung cho nhiều View hoặc nhiều Controller.

Phù hợp với các công nghệ và framework hiện đại: MVC là nền tảng của nhiều công nghệ lập trình web phổ biến hiện nay như ASP.NET MVC, Django (Python), Laravel (PHP), Ruby on Rails (Ruby), Spring MVC (Java)...

1.3 Nguyên lý hoạt động

Nguyên lý hoạt động của mô hình MVC tuân theo một chu trình xử lý yêu cầu từ phía người dùng và phản hồi kết quả theo cách phân tách vai trò rõ ràng giữa các thành phần. Quy trình này có thể mô tả cụ thể như sau:

Người dùng tương tác với giao diện web thông qua trình duyệt (chẳng hạn như nhấn vào một nút “Mua ngay”, nhập từ khóa tìm kiếm hoặc gửi một biểu mẫu).

Giao diện (View) gửi yêu cầu HTTP tới Controller thông qua hệ thống định tuyến (Routing).

Controller tiếp nhận yêu cầu, phân tích và xử lý. Nếu cần dữ liệu, nó sẽ gọi tới Model để thực hiện các tác vụ như truy vấn, kiểm tra logic hoặc xử lý dữ liệu.

Model thực hiện các tác vụ xử lý nghiệp vụ hoặc tương tác với cơ sở dữ liệu, sau đó trả kết quả về cho Controller.

Controller sau khi nhận dữ liệu sẽ chọn View phù hợp, truyền dữ liệu cần thiết cho View.

View hiển thị thông tin lên màn hình cho người dùng, hoàn tất chu trình xử lý yêu cầu.

1.4 Công nghệ triển khai ASP.NET Core

ASP.NET Core là một nền tảng phát triển ứng dụng web mã nguồn mở, đa nền tảng do Microsoft phát triển, kế thừa và nâng cấp từ ASP.NET truyền thống. Một trong những điểm mạnh của ASP.NET Core là hỗ trợ mạnh mẽ mô hình lập trình MVC, cho phép xây dựng ứng dụng web có cấu trúc chặt chẽ, dễ mở rộng và hiệu suất cao.

Trong ASP.NET Core, mô hình MVC được hiện thực hóa một cách rõ ràng thông qua các thư mục riêng biệt như Controllers, Models, Views. Việc tổ chức thư mục theo mô hình chuẩn giúp nhà phát triển dễ dàng quản lý và truy xuất mã nguồn. Công cụ Razor được sử dụng để xây dựng View, giúp kết hợp linh hoạt giữa HTML và mã C#, từ đó tạo ra các trang web động, phản hồi dữ liệu theo thời gian thực.

ASP.NET Core MVC còn hỗ trợ rất tốt các tính năng hiện đại như Dependency Injection (tiêm phụ thuộc), Middleware (thành phần trung gian xử lý request/response), hệ thống định tuyến linh hoạt, hỗ trợ xây dựng RESTful API, quản lý người dùng qua ASP.NET Identity và các kỹ thuật bảo mật như xác thực (Authentication), phân quyền (Authorization).

Đặc biệt, ASP.NET Core hoạt động đa nền tảng, cho phép chạy trên cả Windows, Linux và macOS, cùng với khả năng tích hợp với các công nghệ điện toán đám mây, giúp lập trình viên triển khai hệ thống lên môi trường thực tế một cách linh hoạt và hiệu quả.

Việc lựa chọn ASP.NET Core để xây dựng website không những giúp tối ưu hiệu suất, mà còn giúp sinh viên tiếp cận một công nghệ hiện đại, được sử dụng rộng rãi trong doanh nghiệp hiện nay. Điều này không chỉ phù hợp với yêu cầu môn học mà còn giúp rèn luyện kỹ năng thực tiễn, đáp ứng tốt các yêu cầu của thị trường công nghệ.

CHƯƠNG 2: XÂY DỰNG ỨNG DỤNG WEBSITE BÁN QUẦN ÁO THỜI TRANG

2.1 Phát biểu bài toán ứng dụng

2.1.1 Giới thiệu dự án

Trong bối cảnh công nghệ số phát triển mạnh mẽ, thương mại điện tử đã trở thành một kênh quan trọng, thay đổi cách người dùng tiếp cận và mua sắm các sản phẩm thời trang. Quần áo không chỉ đáp ứng nhu cầu cơ bản mà còn là công cụ để thể hiện phong cách cá nhân và xu hướng thời trang hiện đại. Tuy nhiên, nhiều nền tảng thương mại điện tử tại Việt Nam vẫn còn hạn chế, bao gồm giao diện phức tạp, tốc độ tải trang chưa tối ưu, hoặc thiếu các tính năng hỗ trợ trải nghiệm mua sắm như gợi ý sản phẩm thông minh và khả năng tương thích đa thiết bị.

Dự án “Thiết kế website bán quần áo thời trang” ra đời nhằm giải quyết các vấn đề này, với mục tiêu tạo ra một nền tảng trực tuyến thân thiện, hiệu quả và đáp ứng tốt nhu cầu mua sắm. Website được xây dựng để cung cấp các tính năng cốt lõi như tìm kiếm sản phẩm theo danh mục hoặc từ khóa, xem chi tiết sản phẩm, quản lý giỏ hàng, thanh toán trực tuyến và theo dõi đơn hàng. Ngoài ra, hệ thống được thiết kế để hỗ trợ tích hợp các tính năng nâng cao như đánh giá sản phẩm, gợi ý theo xu hướng và chia sẻ trên mạng xã hội, mang lại trải nghiệm mua sắm toàn diện.

Sử dụng ASP.NET Core MVC – một nền tảng mạnh mẽ của Microsoft, dự án đảm bảo cấu trúc mã nguồn rõ ràng, dễ bảo trì và khả năng mở rộng cao. Website không chỉ là một sản phẩm học thuật mà còn có tiềm năng ứng dụng thực tiễn trong lĩnh vực thương mại điện tử, hỗ trợ các doanh nghiệp thời trang nâng cao khả năng tiếp cận khách hàng.

2.1.2 Vai trò của dự án

Dự án “Thiết kế website bán quần áo thời trang” đóng vai trò như một giải pháp thương mại điện tử hiện đại, mang lại giá trị thiết thực cho cả người dùng và nhà phát triển. Cụ thể, dự án có các vai trò sau:

Tăng cường trải nghiệm mua sắm: Cung cấp một nền tảng trực tuyến tiện lợi, giúp người dùng dễ dàng khám phá, lựa chọn và mua sắm sản phẩm thời trang với giao diện thân thiện và tốc độ nhanh.

Ứng dụng công nghệ tiên tiến: Là cơ hội để áp dụng các công nghệ hiện đại như ASP.NET Core MVC, HTML, CSS, JavaScript và SQL Server vào việc xây dựng một hệ thống web thực tế.

Hỗ trợ kinh doanh thời trang: Giúp các doanh nghiệp, đặc biệt là các thương hiệu nhỏ, tiếp cận khách hàng hiệu quả thông qua một kênh bán hàng trực tuyến chuyên nghiệp.

Nâng cao kỹ năng phát triển: Cho phép người thực hiện rèn luyện quy trình phát triển ứng dụng web từ phân tích yêu cầu, thiết kế, đến triển khai và kiểm thử.

2.1.3 Ý nghĩa của dự án

Về mặt công nghệ và học thuật:

Dự án giúp củng cố kiến thức về lập trình web, từ kiến trúc MVC, quản lý cơ sở dữ liệu, đến thiết kế giao diện người dùng đáp ứng các tiêu chuẩn hiện đại.

Mở ra cơ hội nghiên cứu và tích hợp các công nghệ mới như API, hệ thống thanh toán trực tuyến, hoặc các thuật toán gợi ý sản phẩm dựa trên hành vi người dùng.

Về mặt thực tiễn và xã hội:

Mang đến một nền tảng mua sắm trực tuyến tối ưu, giúp người dùng tiếp cận các sản phẩm thời trang một cách dễ dàng, nhanh chóng và thẩm mỹ.

Góp phần thúc đẩy thị trường thương mại điện tử thời trang tại Việt Nam, hỗ trợ phân phối sản phẩm hợp pháp và nâng cao trải nghiệm tiêu dùng số.

Là ý tưởng khởi điểm cho các dự án khởi nghiệp trong lĩnh vực công nghệ và thời trang, tạo cơ hội phát triển các giải pháp sáng tạo trong ngành thương mại điện tử.

2.2 Phân tích yêu cầu của ứng dụng

2.2.1 Yêu cầu chức năng

Đăng nhập/đăng xuất người dùng.

Hiển thị danh sách sản phẩm theo danh mục (ví dụ: áo thun, quần jeans, váy, phụ kiện).

Quản lý danh mục sản phẩm (thêm, sửa, xóa danh mục, dành cho quản trị viên).

Xem chi tiết sản phẩm, bao gồm tên, mô tả, hình ảnh, giá bán và kích cỡ.

Cho phép người dùng thêm sản phẩm vào giỏ hàng.

Cho phép người dùng chỉnh sửa số lượng hoặc xóa sản phẩm khỏi giỏ hàng.

Tiến hành đặt hàng và xác nhận thanh toán trực tuyến.

Tìm kiếm sản phẩm theo từ khóa (tên sản phẩm, danh mục).

Quản lý đơn hàng (xem lịch sử đơn hàng, trạng thái đơn hàng).

Áp dụng mã giảm giá.

Xem bài báo về thời trang (xu hướng, mẹo phối đồ, tin tức) và bình luận trong bài báo.

2.2.2 Yêu cầu phi chức năng

Tốc độ tải trang dưới 3 giây đối với mọi chức năng cơ bản.

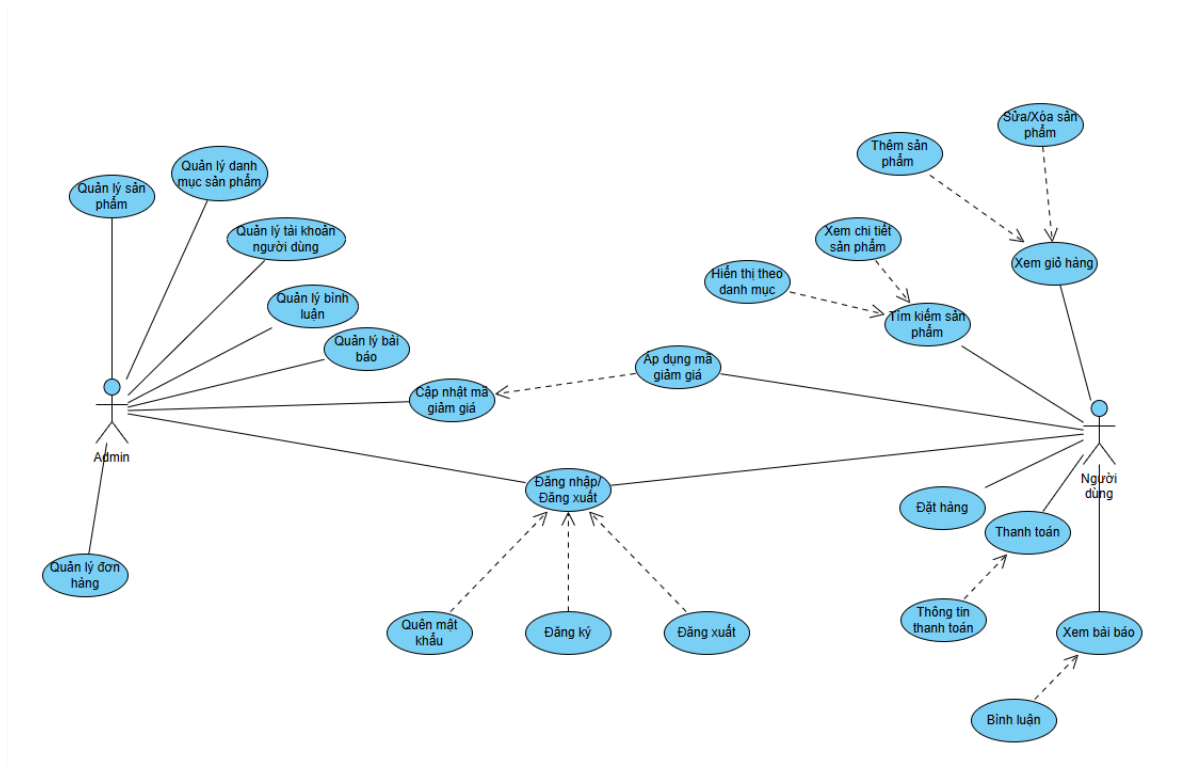
Giao diện trực quan, thiết kế responsive tương thích với các thiết bị (máy tính, điện thoại, máy tính bảng).

Hệ thống hoạt động ổn định, thời gian phản hồi nhanh, truy xuất dữ liệu hiệu quả.

Bảo mật thông tin người dùng, bao gồm xác thực tài khoản và phân quyền truy cập (người dùng/quản trị viên).

Hệ thống dễ mở rộng để tích hợp các tính năng như đánh giá sản phẩm, gợi ý theo xu hướng hoặc kết nối mạng xã hội.

Sử dụng các công nghệ phổ biến (ASP.NET Core MVC, SQL Server, Bootstrap) để dễ dàng duy trì và bảo trì lâu dài.



Hình 2.1: Sơ đồ user case

2.3 Thiết kế cơ sở dữ liệu

2.3.1 Mô tả các bảng dữ liệu

Bảng 1: Orders

Tên cột	Kiểu dữ liệu	Mô tả
OrderId	int	Khóa chính, mã định danh đơn hàng
UserId	String	Khóa ngoại, liên kết người dùng
FirstName	string	Tên của người đặt hàng
LastName	string	Họ của người đặt hàng
Country	string	Quốc gia giao hàng
Address	string	Địa chỉ giao hàng
City	string	Thành phố giao hàng
State	string	Tỉnh thành
Phone	string	Số điện thoại liên lạc
Email	string	Email người đặt hàng
OrderNotes	string	Ghi chú bổ sung cho đơn hàng
OrderDate	datetime	Ngày đặt hàng
Total	decimal	Tổng tiền đơn hàng
PaymentMethod	string	Phương thức thanh toán

Bảng 2:

Tên cột	Kiểu dữ liệu	Mô tả
OrderItemId	int	Khóa chính, mã định danh mục hàng
OrderId	int	Khóa ngoại, liên kết với đơn hàng
ProductId	int	Khóa ngoại, liên kết với sản phẩm
Quantity	int	Số lượng sản phẩm trong mục hàng
Price	decimal	Giá trị của mỗi đơn vị sản phẩm

Bảng 3: Products

Tên cột	Kiểu dữ liệu	Mô tả
ProductId	int	Khóa chính, mã định danh sản phẩm
Name	string	Tên sản phẩm
Description	string	Mô tả chi tiết sản phẩm
Price	decimal	Giá bán của sản phẩm
ImageUrl	string	Hình ảnh sản phẩm
Size	string	Kích cỡ của sản phẩm
Color	string	Màu sắc của sản phẩm
IsNew	bool	Sản phẩm mới
IsOnSale	bool	Sản phẩm đang giảm giá
Rating	int	Đánh giá của sản phẩm

Bảng 4: Category

Tên cột	Kiểu dữ liệu	Mô tả
CategoryId	int	Khóa chính, mã định danh danh mục
Name	string	Tên danh mục sản phẩm

Bảng 5: BlogPost

Tên cột	Kiểu dữ liệu	Mô tả
BlogPostId	int	Khóa chính, mã định danh bài viết
Title	string	Tiêu đề bài viết
Content	string	Nội dung bài viết
DatePosted	datetime	Ngày đăng bài viết
Author	string	Tác giả bài viết
Quote	string	Trích dẫn trong bài viết
QuoteAuthor	string	Tác giả của trích dẫn
Tags	string	Các thẻ tác liên quan
ImageUrl	string	Hình ảnh bài viết
PreviousPostId	int	Khóa ngoại, liên kết bài viết trước
NextPostId	int	Khóa ngoại, liên kết bài viết sau

Bảng 6: Comment

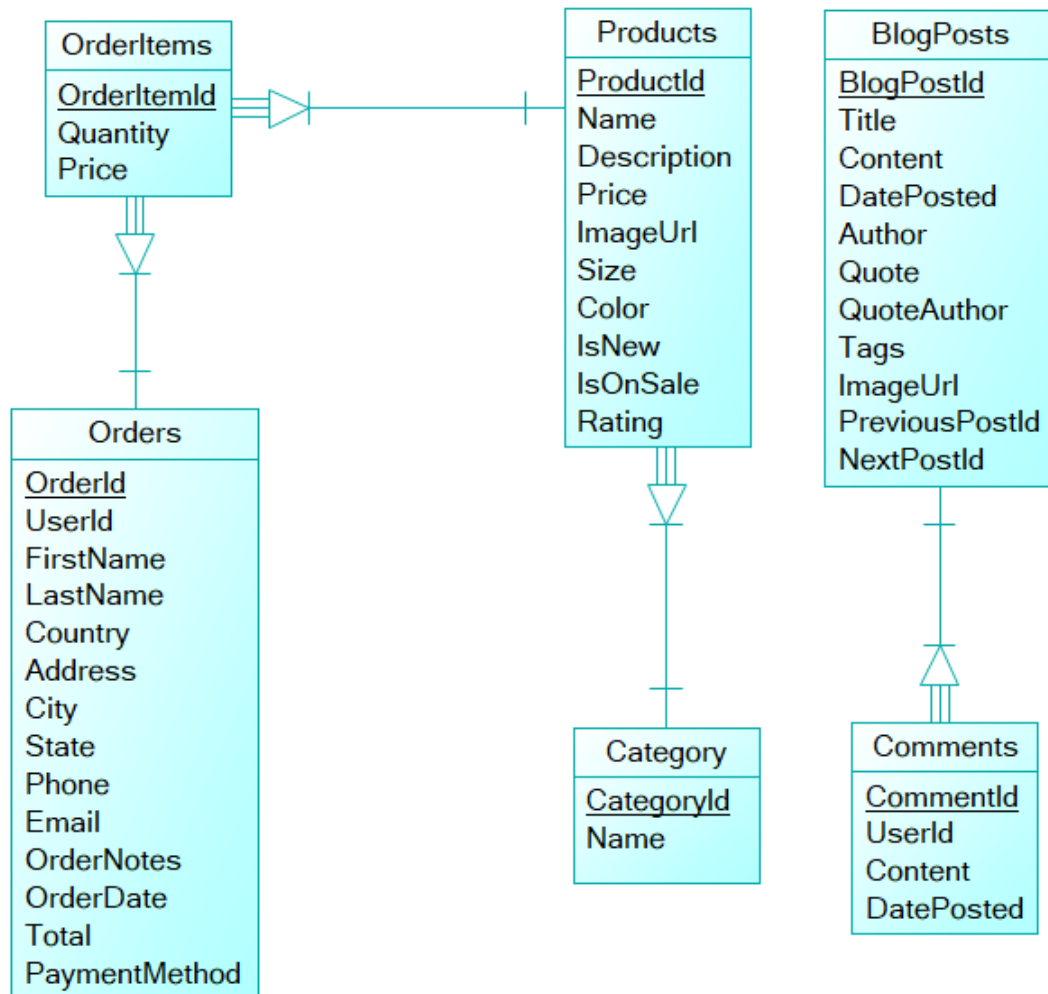
Tên cột	Kiểu dữ liệu	Mô tả
CommentId	int	Khóa chính, mã định danh bình luận
BlogPostId	int	Khóa ngoại, mã định danh bài viết
UserId	string	Mã định danh người dùng
Content	string	Nội dung bình luận
DatePosted	datetime	Ngày đăng bình luận

2.3.2 Các mối quan hệ

1-n: Một Order có nhiều OrderItems; một Product thuộc về một Category; một BlogPost có nhiều Comments; một User (được đại diện qua UserId) có thể có nhiều Orders và Comments.

n-n: Không áp dụng trong thiết kế này (có thể mở rộng nếu thêm tính năng như "Sản phẩm yêu thích" hoặc "Tag liên quan").

2.3.3 Mô hình ERD

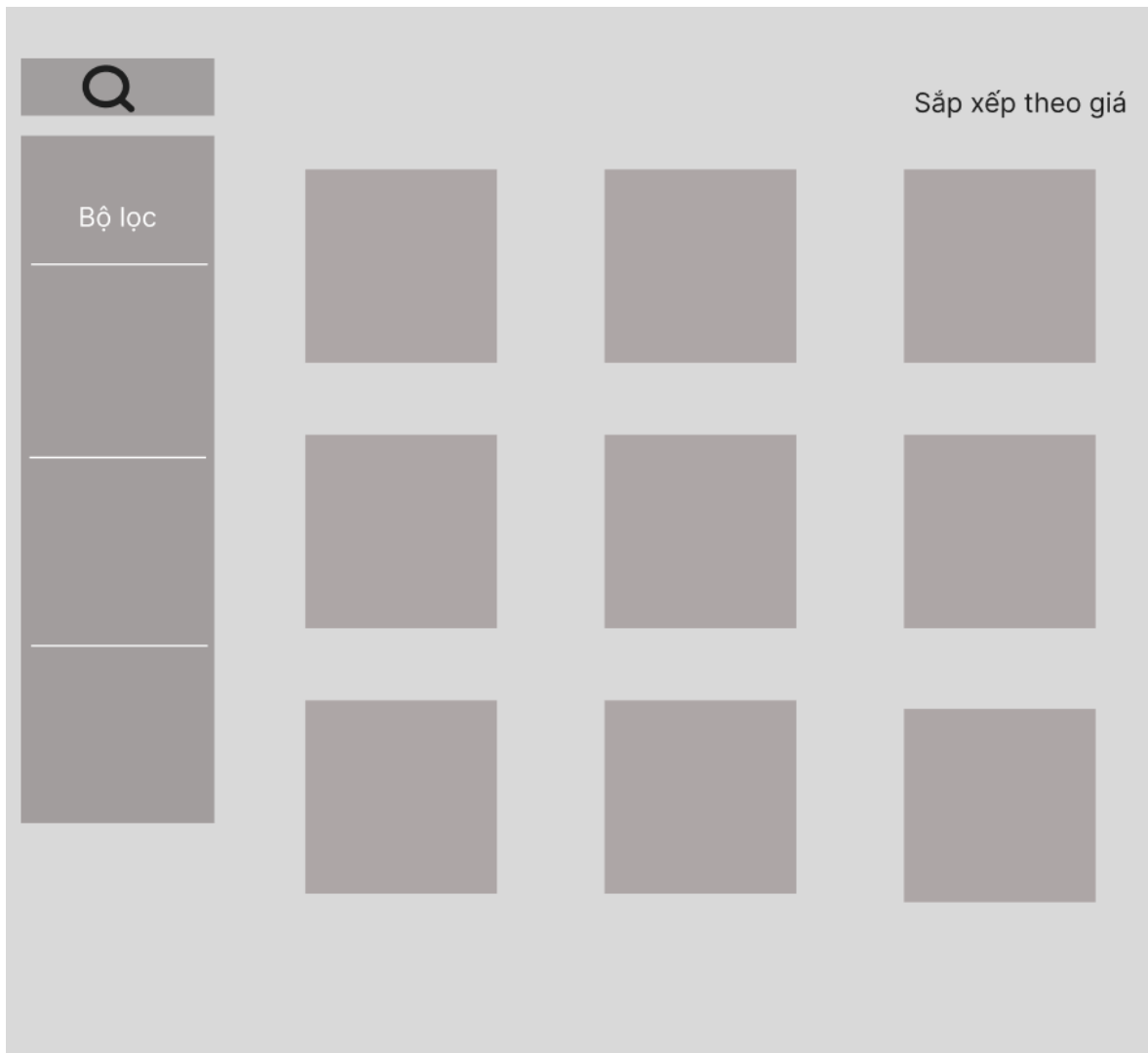


Hình 2.2: Mô hình ERD cơ sở dữ liệu

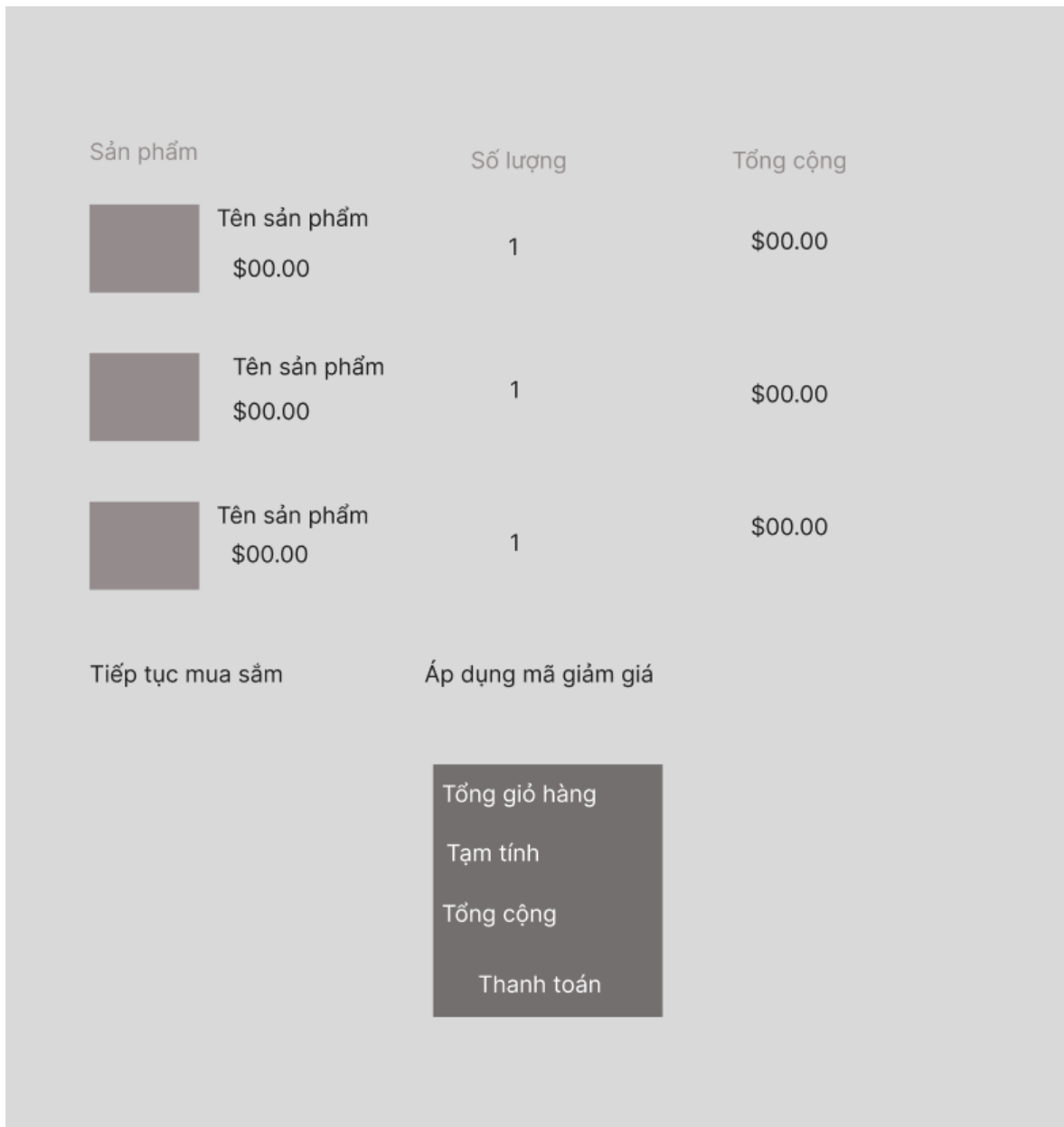
2.4 Thiết kế giao diện người dùng:



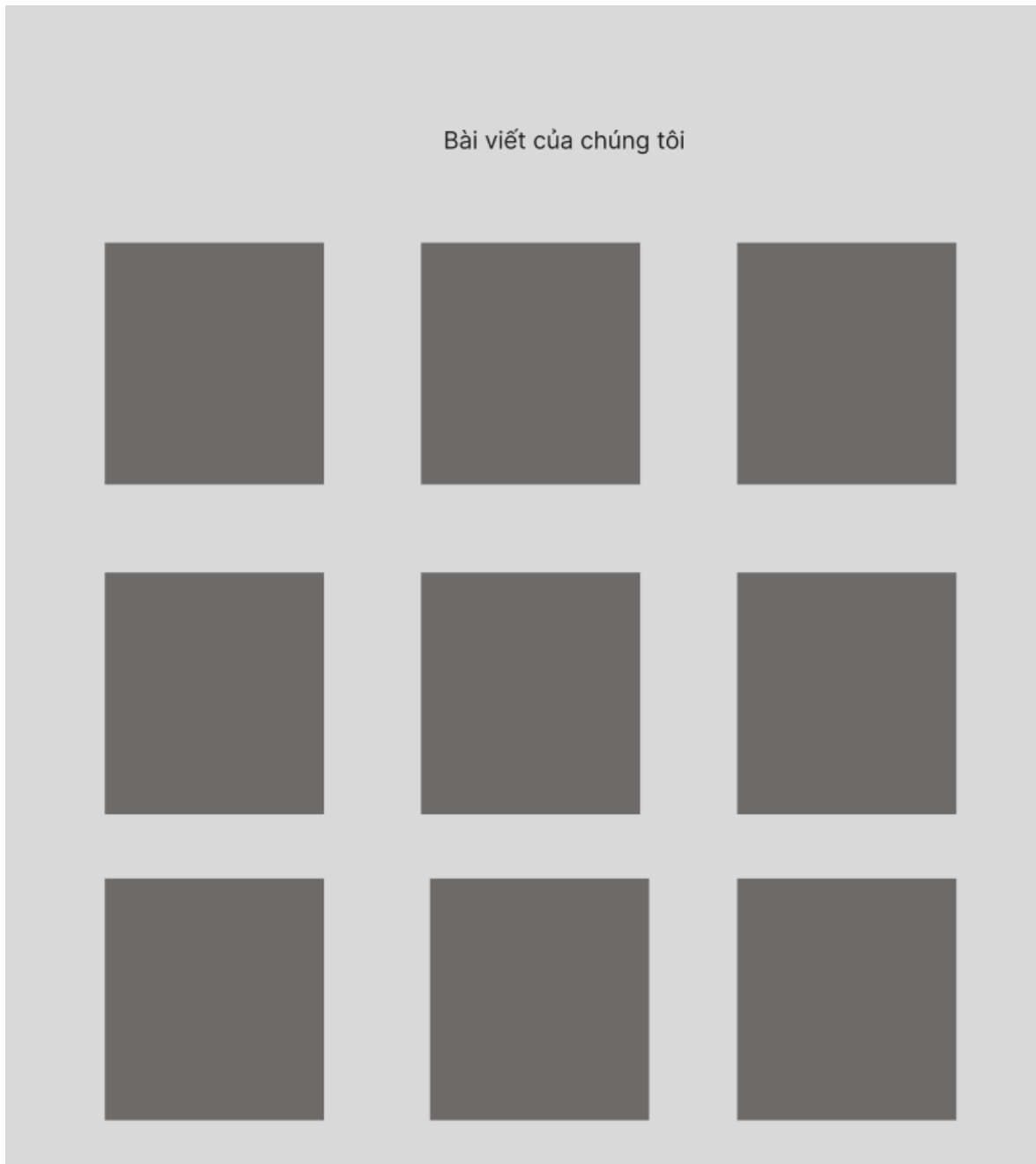
Hình 2.3: Bản wireframe trang chủ



Hình 2.4: Bản wireframe cửa hàng



Hình 2.5: Bản wireframe giỏ hàng



Hình 2.6: Bản wireframe bài báo

2.5 Thiết kế các thành phần MVC:

Model: Định nghĩa các lớp quản lý dữ liệu

Class Products:

```
public class Product
{
    public int Id { get; set; }
    public string Name { get; set; }
```



```

    public string Description { get; set; }
    public decimal Price { get; set; }
    public string ImageUrl { get; set; }
    public int CategoryId { get; set; }
    public Category Category { get; set; }
    public string Size { get; set; }
    [Required]
    public string Color { get; set; }
    public bool IsNew { get; set; }
    public bool IsOnSale { get; set; }
    public int Rating { get; set; }
}

```

Class Orders:

```

public class Order
{
    public int Id { get; set; }
    public string UserId { get; set; }

    [Required(ErrorMessage = "Họ là bắt buộc")]
    public string FirstName { get; set; }

    [Required(ErrorMessage = "Tên là bắt buộc")]
    public string LastName { get; set; }

    [Required(ErrorMessage = "Quốc gia là bắt buộc")]
    public string Country { get; set; }

    [Required(ErrorMessage = "Địa chỉ là bắt buộc")]
    public string Address { get; set; }

    [Required(ErrorMessage = "Thành phố là bắt buộc")]
    public string City { get; set; }

    [Required(ErrorMessage = "Quốc gia/State là bắt buộc")]
    public string State { get; set; }

    [Required(ErrorMessage = "Mã bưu điện là bắt buộc")]
    public string PostalCode { get; set; }

    [Required(ErrorMessage = "Số điện thoại là bắt buộc")]
    [Phone(ErrorMessage = "Số điện thoại không hợp lệ")]
    public string Phone { get; set; }
}

```

```

[Required(ErrorMessage = "Email là bắt buộc")]
[EmailAddress(ErrorMessage = "Email không hợp lệ")]
public string Email { get; set; }

public string OrderNotes { get; set; }
public DateTime OrderDate { get; set; }
public decimal Total { get; set; }
public string PaymentMethod { get; set; }
public List<OrderItem> OrderItems { get; set; } = new List<OrderItem>();
}

```

Class OrderItems:

```

public class OrderItem
{
    public int Id { get; set; }
    public int OrderId { get; set; }
    public int ProductId { get; set; }
    public int Quantity { get; set; }
    public decimal Price { get; set; }
    public Product Product { get; set; }
    public Order Order { get; set; }
}

```

Class Category:

```

public class Category
{
    public int Id { get; set; }
    public string Name { get; set; }
    public List<Product> Products { get; set; }
}

```

Class BlogPost:

```

public class BlogPost
{
    public int Id { get; set; }
    [Required]
    public string Title { get; set; }
    [Required]
    public string Content { get; set; }
    public DateTime DatePosted { get; set; }
    [Required]
    public string Author { get; set; }
}

```

```

public string Quote { get; set; }
public string QuoteAuthor { get; set; }
public string Tags { get; set; }
public string ImageUrl { get; set; }
public int? PreviousPostId { get; set; }
public int? NextPostId { get; set; }
public List<Comment> Comments { get; set; }
}

```

Class Comment:

```

public class Comment
{
    public int Id { get; set; }
    public int BlogPostId { get; set; }
    public string UserId { get; set; }
    public string Content { get; set; }
    public DateTime DatePosted { get; set; }
    public BlogPost BlogPost { get; set; }
}

```

View:

View chịu trách nhiệm trình bày dữ liệu từ Model một cách trực quan, tạo điều kiện cho người dùng tương tác với hệ thống thông qua các giao diện dễ sử dụng. Trong mô hình MVC, View chỉ tập trung vào hiển thị thông tin mà không xử lý logic nghiệp vụ. Trang chủ (Index.cshtml) được thiết kế sử dụng Bootstrap kết hợp với Razor syntax để hiển thị dữ liệu động từ Model, đảm bảo giao diện responsive và thẩm mỹ, đồng thời tích hợp các thư viện như Owl Carousel, Font Awesome, Elegant Icons, Magnific Popup, Nice Select, và Slicknav để nâng cao trải nghiệm người dùng.

Controller

HomeController:

Index: Lấy 8 sản phẩm mới nhất và 3 bài viết blog mới nhất từ cơ sở dữ liệu, truyền vào View.

About, Contact: Hiển thị các trang tĩnh, xử lý form liên hệ với xác thực dữ liệu đầu vào.

AccountController:

Login: Xử lý đăng nhập với xác thực qua SignInManager, ghi log lỗi và thông báo.

Register: Xử lý đăng ký tài khoản mới với UserManager, kiểm tra lỗi và thông báo.

Logout: Thực hiện đăng xuất và chuyển hướng về trang chủ.

ShopController:

Index: Lấy danh sách sản phẩm và danh mục, hỗ trợ lọc theo categoryId.
Details: Lấy thông tin sản phẩm chi tiết và sản phẩm liên quan theo categoryId.

ShoppingCartController:

Index: Lấy giỏ hàng từ session, hiển thị danh sách sản phẩm và tổng giá.
AddToShoppingCart: Thêm sản phẩm vào giỏ hàng qua ICartService.
UpdateCart: Cập nhật số lượng hoặc xóa sản phẩm trong giỏ hàng.
ApplyCoupon: Xử lý áp dụng mã giảm giá (chưa đầy đủ logic).
RemoveFromCart: Xóa sản phẩm khỏi giỏ hàng.

OrdersController:

Checkout: Hiển thị form thanh toán với thông tin giỏ hàng.
CheckoutComplete: Hiển thị thông tin đơn hàng sau khi hoàn tất.

BlogController:

Index: Lấy danh sách bài viết blog.
Details: Hiển thị chi tiết bài viết và danh sách bình luận.
PostComment: Xử lý thêm bình luận mới.

2.6 Triển khai và cài đặt

2.6.1 Cấu hình máy phát triển

Dự án được triển khai trên máy tính cá nhân với cấu hình mạnh mẽ, đảm bảo quá trình lập trình, biên dịch và kiểm thử diễn ra mượt mà:

Hệ điều hành: Windows 11 64-bit
Bộ vi xử lý: Intel Core i7-11800H
Bộ nhớ RAM: 16 GB DDR4
Card đồ họa: NVIDIA RTX 3050
Môi trường phát triển (IDE): Visual Studio 2022 Community Edition
Trình duyệt kiểm thử: Google Chrome, Microsoft Edge

2.6.2 Môi trường phát triển

Dự án website bán quần áo thời trang được xây dựng theo mô hình ASP.NET Core MVC với các công nghệ chính như sau:

Ngôn ngữ lập trình: C#

Kiến trúc: Model – View – Controller

Razor View Engine: công cụ tạo giao diện động, tích hợp HTML và C#

Entity Framework Core (Code First): thao tác cơ sở dữ liệu bằng đối tượng

Microsoft SQL Server LocalDB: hệ quản trị cơ sở dữ liệu nhẹ và tiện lợi

ASP.NET Core Identity: xử lý đăng ký, đăng nhập, phân quyền

Bootstrap: xây dựng giao diện web responsive, thẩm mỹ

Font Awesome: thư viện biểu tượng để tăng tính trực quan

Elegant Icons: bộ sưu tập biểu tượng tinh tế và hiện đại

Magnific Popup: plugin popup/lightbox cho hình ảnh và video

Nice Select: plugin tùy chỉnh giao diện dropdown select

Owl Carousel: plugin carousel cho slider hình ảnh

Slicknav: plugin menu điều hướng responsive

CSS & JavaScript: cải thiện hiệu ứng và trải nghiệm người dùng

.NET SDK: sử dụng phiên bản mới nhất .NET 8.0 cho hiệu suất tối ưu

2.6.3 Quy trình cài đặt và triển khai

Quá trình thiết lập và chạy ứng dụng được thực hiện theo các bước sau:

Tạo project: Mở Visual Studio 2022, chọn template ASP.NET Core Web App (Model-View-Controller) và khởi tạo dự án.

Cài đặt thư viện: Cài các package cần thiết như Microsoft.EntityFrameworkCore, Microsoft.AspNetCore.Identity.EntityFrameworkCore,... thông qua NuGet.

Xây dựng Model và lớp DataContext: Tạo các lớp POCO đại diện cho bảng dữ liệu (sản phẩm, người dùng, đơn hàng...).

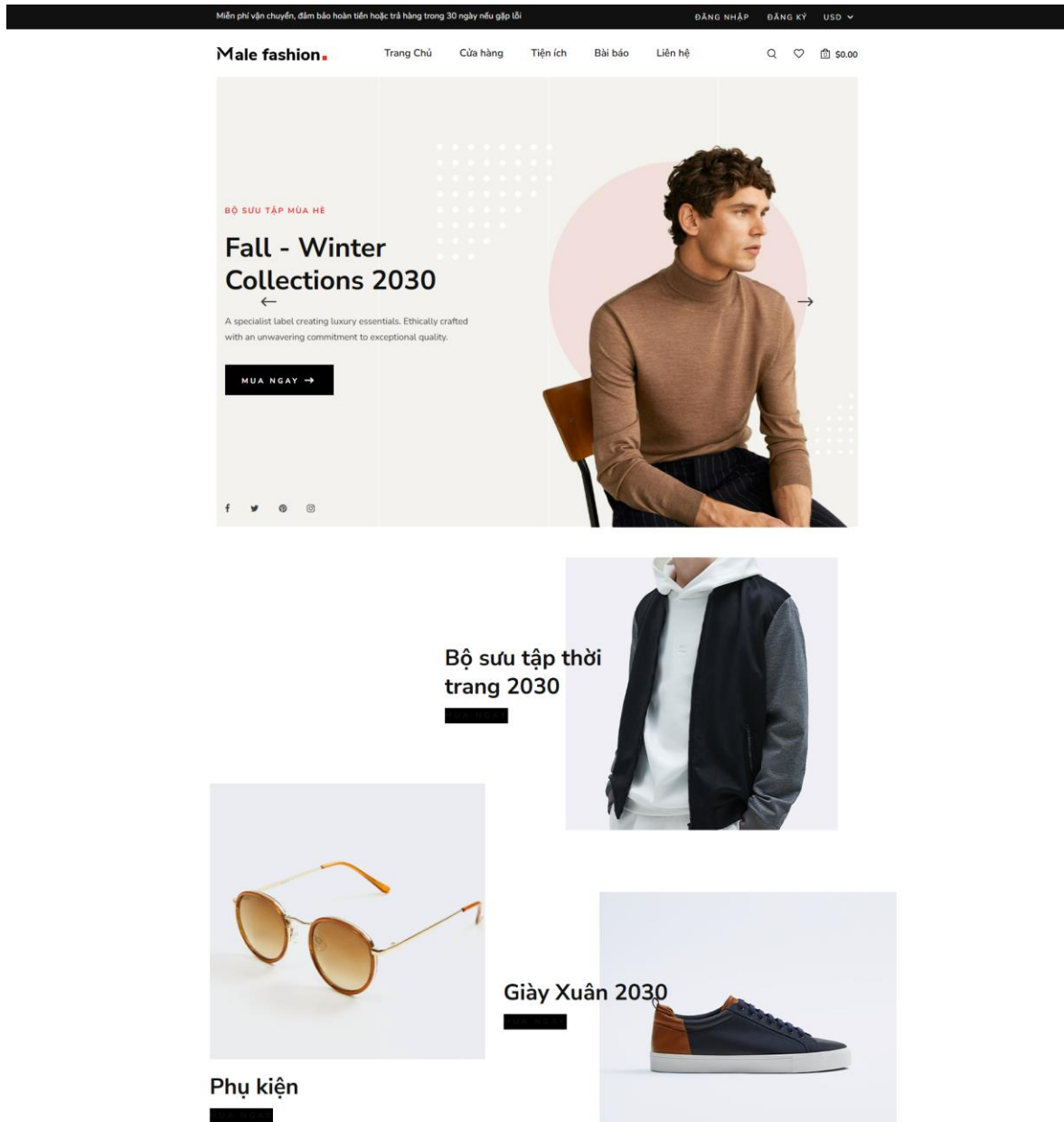
Cấu hình hệ thống: Thêm chuỗi kết nối trong appsettings.json và cấu hình DI, Identity trong Program.cs.

Tạo cơ sở dữ liệu: Sử dụng lệnh Add-Migration Initial và Update-Database để khởi tạo database theo mô hình Code First.

Viết Controller và View: Xây dựng logic xử lý trong Controller và tạo View để hiển thị (trang chủ, danh sách truyện, giỏ hàng, thanh toán,...).

Chạy ứng dụng: Nhấn Run hoặc Ctrl + F5 để chạy website và kiểm thử trên trình duyệt.

CHƯƠNG 3: KẾT QUẢ CHƯƠNG TRÌNH



Bán chạy nhất

Hàng mới về

Khuyến mãi hot



Ankle Boots

+ Thêm vào giỏ

★★★★☆

\$98.49



Basic Flowing Scarf

+ Thêm vào giỏ

☆☆☆☆☆

\$26.28



Canvas Tote Bag

+ Thêm vào giỏ

★★★★☆

\$39.99

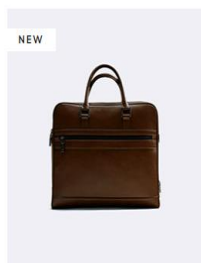


Diagonal Textured Cap

+ Thêm vào giỏ

☆☆☆☆☆

\$60.90



Hooded Sweatshirt

+ Thêm vào giỏ

☆☆☆☆☆

\$45.00



Leather Sneakers

+ Thêm vào giỏ

★★★★☆

\$89.99



Lether Backpack

+ Thêm vào giỏ

☆☆☆☆☆

\$31.37



Multi-pocket Chest Bag

+ Thêm vào giỏ


★★★★☆

\$43.48

Clothings Hot

Shoe Collection

Phụ kiện



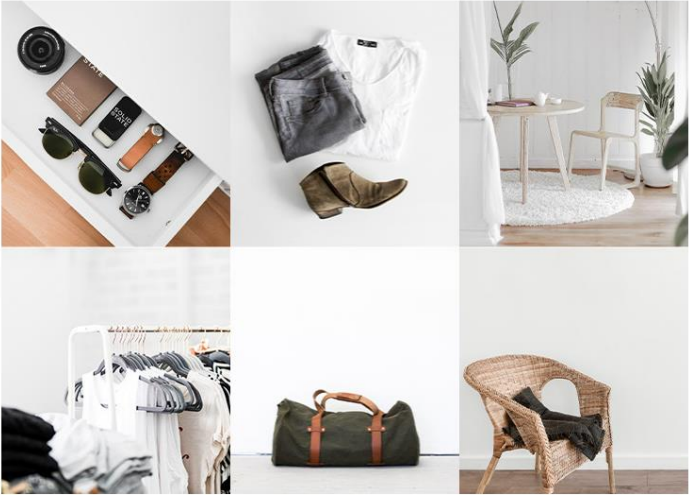
ƯU ĐÃI TRONG TUẦN

Multi-pocket Chest Bag Black

29 : 02 : 40 : 21

Days Hours Minutes Seconds

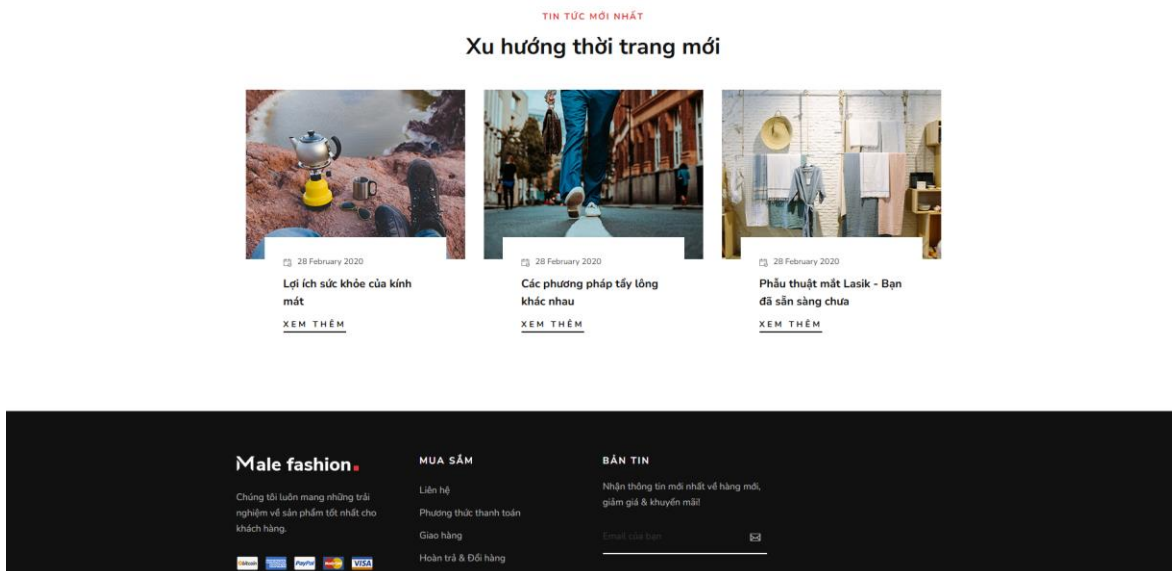
MUA NGAY



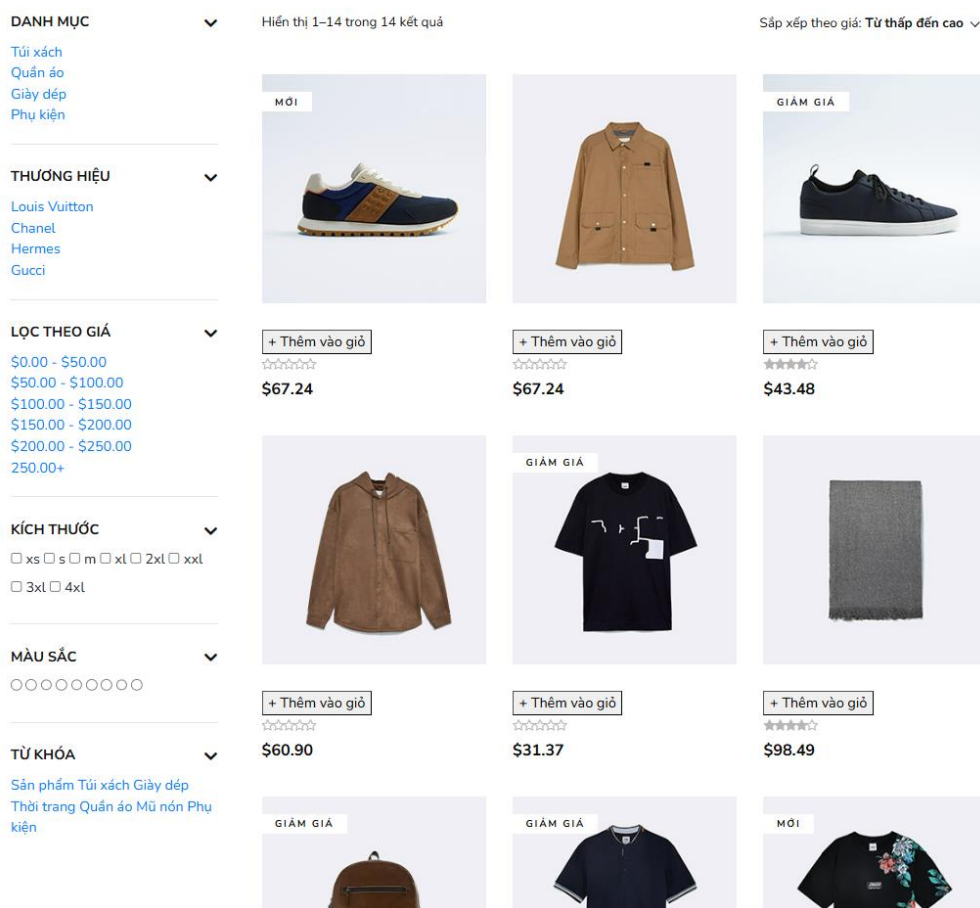
Instagram

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

#Male_Fashion



Hình 3.1: giao diện trang chủ



Hình 3.2: giao diện cửa hàng

Bài viết của chúng tôi



16 February 2020

Những loại máy uốn tóc
nào tốt nhất

[XEM THÊM](#)



21 February 2020

Nhãn Eternity bền mãi mãi

[XEM THÊM](#)



28 February 2020

Lợi ích sức khỏe của kính
mắt

[XEM THÊM](#)



16 February 2020

Phẫu thuật nâng ngực -
nâng cao sắc đẹp

[XEM THÊM](#)



21 February 2020

Nhãn cưới - Món quà cả đời

[XEM THÊM](#)



28 February 2020

Các phương pháp tẩy lông
khác nhau

[XEM THÊM](#)

Hình 3.3: giao diện bài báo

SẢN PHẨM

SỐ LƯỢNG

TỔNG CỘNG



Piqué Biker Jacket
\$67.24

< 1 >

\$67.24

x

TIẾP TỤC MUA SẮM

MÃ GIẢM GIÁ

Mã giảm giá

ÁP DỤNG

TỔNG GIỎ HÀNG

Tạm tính \$67.24

Tổng cộng \$67.24

TIẾN HÀNH THANH TOÁN

Hình 3.4: giao diện giỏ hàng

Kết luận

Kiến thức đặt được:

Qua quá trình phát triển dự án "Thiết kế website bán quần áo thời trang", các kiến thức và kỹ năng quan trọng đã được tích lũy, bao gồm:

Kiến trúc MVC: Hiểu rõ cách tổ chức mã nguồn theo mô hình Model-View-Controller, phân tách logic nghiệp vụ (Controller), dữ liệu (Model), và giao diện (View) để đảm bảo tính dễ bảo trì và mở rộng.

ASP.NET Core: Thành thạo sử dụng framework ASP.NET Core 6.0/8.0 để xây dựng ứng dụng web hiện đại, bao gồm quản lý request, routing, và middleware.

Entity Framework Core: Làm quen với cách sử dụng Code First để thiết kế và quản lý cơ sở dữ liệu (SQL Server LocalDB), từ việc tạo migration đến tương tác với các bảng như Orders, Products, và BlogPosts.

ASP.NET Core Identity: Nắm vững cách triển khai hệ thống đăng nhập, đăng ký, và phân quyền cho người dùng và quản trị viên.

Front-end Development: Áp dụng Bootstrap, Font Awesome, Elegant Icons, Magnific Popup, Nice Select, Owl Carousel, và Slicknav để xây dựng giao diện responsive và tương tác, cùng với CSS tùy chỉnh để đáp ứng thiết kế độc đáo.

Dependency Injection (DI): Hiểu cách cấu hình DI trong Program.cs để quản lý các dịch vụ như DbContext, Identity, và các repository/service tùy chỉnh, cải thiện khả năng kiểm thử và mô-đun hóa.

Kỹ năng kiểm thử và triển khai: Thực hành chạy ứng dụng trên localhost, kiểm tra trên các trình duyệt (Google Chrome, Microsoft Edge), và áp dụng quy trình Code First để khởi tạo cơ sở dữ liệu.

Quản lý lỗi và log: Sử dụng ILogger để ghi lại thông tin và xử lý ngoại lệ, đảm bảo ứng dụng ổn định trong quá trình phát triển.

Hạn chế

Dự án vẫn còn một số hạn chế cần khắc phục để hoàn thiện hơn:

Chưa hoàn thiện chức năng thanh toán: Chưa hiển thị thông báo "Thanh toán thành công" và không lưu dữ liệu đơn hàng sau khi thanh toán.

Lỗi định dạng form ở trang chi tiết bài báo: Giao diện form trên trang này chưa được hiển thị đúng, ảnh hưởng đến trải nghiệm người dùng.

Chưa hoàn thiện trang Contact: Chức năng liên hệ chưa được triển khai đầy đủ, thiếu các yếu tố như gửi email hoặc lưu thông tin người dùng.

Chưa tạo form quên mật khẩu: Chưa tích hợp chức năng khôi phục mật khẩu cho người dùng, hạn chế khả năng hỗ trợ người dùng quên thông tin đăng nhập.

Chưa làm trang Admin cho web: Chưa phát triển giao diện quản trị viên để quản lý danh mục, sản phẩm, đơn hàng, và người dùng, gây khó khăn trong việc vận hành hệ thống.

Hướng phát triển trong tương lai

Dựa trên nền tảng đã xây dựng, dự án có tiềm năng phát triển theo các hướng sau để nâng cao giá trị và khả năng cạnh tranh:

Tích hợp API: Phát triển API RESTful để hỗ trợ ứng dụng di động hoặc tích hợp với các nền tảng bên thứ ba (ví dụ: thanh toán trực tuyến như MoMo, ZaloPay).

Hệ thống gợi ý thông minh: Sử dụng thuật toán học máy (machine learning) hoặc AI để phân tích hành vi người dùng và đề xuất sản phẩm phù hợp, tăng trải nghiệm mua sắm.

Tối ưu hiệu suất: Áp dụng caching (ví dụ: Redis) và CDN để giảm thời gian tải trang, đáp ứng yêu cầu phi chức năng về tốc độ dưới 3 giây.

Tích hợp mạng xã hội: Thêm tính năng chia sẻ sản phẩm lên Facebook, Instagram, hoặc tích hợp đăng nhập qua mạng xã hội để thu hút người dùng.

Quản lý đa ngôn ngữ: Xây dựng hỗ trợ nhiều ngôn ngữ (ví dụ: tiếng Việt, tiếng Anh) để mở rộng thị trường quốc tế.

Bảo mật nâng cao: Triển khai mã hóa dữ liệu (SSL/TLS), bảo vệ chống tấn công CSRF/XSS, và tích hợp 2FA (xác thực hai yếu tố) để tăng cường an toàn thông tin.

Phân tích dữ liệu: Sử dụng công cụ như Power BI hoặc Google Analytics để theo dõi hành vi người dùng, từ đó tối ưu hóa chiến lược kinh doanh.

Tài liệu tham khảo

[1] ASP.NET Core MVC – Microsoft – Microsoft Docs – 2023

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

[2] ASP.NET Core Identity Overview – Microsoft – Microsoft Docs – 2023

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