

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



YERSIN UNIVERSITY

**BÁO CÁO MÔN HỌC
LẬP TRÌNH WEB 1**

**Xây dựng website thương mại
điện tử kinh doanh sản phẩm
Máy tính**

**GVHD : Nguyễn Đức Tấn
SVTH : Nguyễn Công Thiện
Mã số SV : 2301010007
Khóa học : 2024 - 2025**

Đà Lạt, tháng 7- 2025

PHẦN NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm:

Ngày ... tháng ... năm...

Ký và ghi rõ họ tên

LỜI NÓI ĐẦU

Trước hết chúng em xin cảm ơn đến thầy Nguyễn Đức Tấn, người đã hướng dẫn chúng em rất nhiều trong suốt quá trình tìm hiểu nghiên cứu và hoàn thành đồ án này từ lý thuyết đến ứng dụng. Sự hướng dẫn của thầy đã giúp chúng em có thêm được những hiểu biết và cách thức tạo nên một website bán hàng trực tuyến cho cửa hàng bán PC. Do lĩnh vực của đề tài này còn tương đối rộng và kiến thức của nhóm chúng em còn nhiều hạn chế nên đồ án không thể tránh khỏi sai sót. Kính mong sự chỉ bảo thêm của các thầy và những sự góp ý của các bạn sinh viên trong lớp để nhóm có thể hoàn thiện đề tài nghiên cứu của mình hơn nữa.

MỤC LỤC

LỜI NÓI ĐẦU.....	3
MỤC LỤC.....	4
DANH SÁCH HÌNH ẢNH.....	5
Chương 1: Cơ sở lý thuyết.....	7
1.1 Khái niệm MVC.....	7
1.2 Đặc điểm MVC.....	7
1.3 Nguyên lý hoạt động MVC.....	8
1.4 Công nghệ triển khai ASP.NET Core.....	8
Chương 2. Xây dựng ứng dụng Web (triển khai thực tế).....	9
2.1 Triển khai ứng dụng thực tế.....	9
2.2 Phân tích yêu cầu ứng dụng	9
2.3 Thiết kế cơ sở dữ liệu.....	12
2.4 Thiết kế giao diện người dùng.....	19
2.5 Thiết kế các thành phần MVC.....	25
2.6 Triển khai và cài đặt.....	28
2.6.1 Môi trường triển khai.....	28
2.6.2 Môi trường cài đặt.....	29
Chương 3. Kết quả trưng trình.....	30
Kết luận.....	30
Tài liệu tham khảo.....	31

DANH SÁCH HÌNH ẢNH

Hình ảnh	Trang
Hình 1	12
Hình 2	12
Hình 3	13
Hình 4	14
Hình 5	14
Hình 6	15
Hình 7	15
Hình 8	15
Hình 9	16
Hình 10	16
Hình 11	16
Hình 12	17
Hình 13	17
Hình 14	18
Hình 15	18
Hình 16	19
Hình 17	20
Hình 18	20
Hình 19	21
Hình 20	22
Hình 21	23
Hình 22	23
Hình 23	24
Hình 24	24
Hình 25	25
Hình 26	25
Hình 27	26
Hình 28	28

Hình 29	29
Hình 30	30

Giới thiệu dự án

- Trong kỷ nguyên chuyển đổi số, việc số hóa hoạt động bán hàng và quản lý là yếu tố then chốt để các cửa hàng truyền thống thích nghi với nhu cầu thị trường hiện đại. Dự án Computer Shop ra đời nhằm hiện thực hóa điều đó tạo nên một nền tảng thương mại điện tử đơn giản, hiện đại, hỗ trợ việc mua bán linh kiện và thiết bị máy tính một cách nhanh chóng và thuận tiện.
- Đối với các doanh nghiệp kinh doanh máy tính, linh kiện và thiết bị công nghệ, việc xây dựng một hệ thống bán hàng trực tuyến không chỉ giúp nâng cao trải nghiệm khách hàng mà còn là giải pháp hiệu quả để mở rộng thị trường, gia tăng doanh thu và quản lý hoạt động kinh doanh.
- Một website thương mại điện tử được thiết kế bài bản sẽ cho phép khách hàng dễ dàng tìm kiếm, so sánh và lựa chọn các sản phẩm công nghệ phù hợp, đồng thời hỗ trợ việc đặt hàng nhanh chóng. Bên cạnh đó, hệ thống còn giúp doanh nghiệp thuận tiện trong việc quản lý sản phẩm, giỏ hàng, đơn hàng, thông tin khách hàng và xử lý đơn mua từ đó tối ưu hóa quy trình vận hành và từng bước thực hiện chuyển đổi số trong lĩnh vực kinh doanh công nghệ.
- Từ thực tiễn chuyển đổi số trong kinh doanh, em thực hiện đề tài “Xây dựng website thương mại điện tử bán máy tính và linh kiện” với mục tiêu phát triển một nền tảng trực tuyến thân thiện, dễ sử dụng cho cả khách hàng và quản trị viên. Website không chỉ hỗ trợ người dùng tra cứu, đặt hàng sản phẩm công nghệ nhanh chóng, mà còn cung cấp các chức năng quản lý sản phẩm, đơn hàng, khách hàng và thống kê doanh thu cho quản trị viên. Qua dự án này, em đã vận dụng kiến thức về lập trình web, cơ sở dữ liệu và thiết kế hệ thống, đồng thời rèn luyện kỹ năng phân tích yêu cầu, giải quyết bài toán thực tế và đóng góp một giải pháp hiệu quả nhằm nâng cao hiệu suất hoạt động cho doanh nghiệp trong thời đại số hóa.

Chương 1 Cơ sở lý thuyết.

1.1 Khái niệm MVC.

- **MVC (Model-View-Controller)** là một mẫu kiến trúc phần mềm được sử dụng rộng rãi trong việc phát triển ứng dụng để tạo ra các giao diện người dùng trực quan và có khả năng tương tác cao. MVC chia một ứng dụng thành 3 phần chính và mỗi phần có một vai trò riêng biệt:
 - + Model đại diện cho dữ liệu và quy tắc nghiệp vụ của ứng dụng.
 - + View chịu trách nhiệm hiển thị dữ liệu cho người dùng một cách trực quan và tương tác.
 - + Controller đóng vai trò là cầu nối giữa Model và View, xử lý các yêu cầu từ người dùng và cập nhật giao diện tương ứng.

1.2 Đặc điểm MVC.

- MVC sở hữu nhiều đặc điểm nổi bật giúp nó trở thành một lựa chọn ưu việt trong phát triển phần mềm. Mỗi thành phần (Model, View, Controller) đảm nhiệm một vai trò riêng biệt và độc lập, giảm thiểu sự phụ thuộc lẫn nhau, sự tách biệt này giúp tăng cường tính module hóa của ứng dụng, làm cho mã nguồn dễ hiểu, dễ quản lý và dễ bảo trì hơn rất nhiều.
- Ưu điểm của mô hình MVC.
 - + Tách biệt rõ ràng giữa logic xử lý, dữ liệu và giao diện.
 - + Dễ bảo trì, dễ mở rộng.
 - + Tái sử dụng code tốt (View có thể dùng lại, Model có thể dùng chung nhiều Controller).
 - + Hỗ trợ tốt cho phát triển theo nhóm.
- Bên cạnh những ưu điểm thì vẫn có một số ngược điểm sau:
 - + Với các dự án đơn giản hoặc quy mô nhỏ, việc tách riêng Model, View và Controller có thể khiến cấu trúc trở nên cồng kềnh, gây tốn thời gian
 - + Tăng độ phức tạp khi bảo trì .
 - + Không kiểm soát chặt luồng dữ liệu giữa Model và View.

1.3 Nguyên lý hoạt động MVC.

- Mô hình MVC (Model - View - Controller) hoạt động dựa trên sự phối hợp chặt chẽ giữa ba thành phần chính nhằm tổ chức việc xử lý dữ liệu và hiển thị thông tin một cách rõ ràng và hiệu quả. Khi người dùng tương tác với giao diện (View), yêu cầu đó sẽ được gửi đến Controller – thành phần chịu trách nhiệm tiếp nhận và phân tích hành động từ người dùng. Controller sau đó sẽ gọi đến Model để thực hiện các thao tác liên quan đến dữ liệu, chẳng hạn như truy xuất, cập nhật hoặc xử lý thông tin. Sau khi Model hoàn tất việc xử lý, kết quả sẽ được chuyển lại cho Controller và được truyền sang View để hiển thị lên giao diện người dùng. Chu trình này lặp lại mỗi khi có tương tác mới, tạo nên một quy trình xử lý mạch lạc, trong đó giao diện, logic xử lý và dữ liệu được tách biệt rõ ràng, giúp việc phát triển, bảo trì và mở rộng ứng dụng trở nên thuận tiện hơn.

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, được Microsoft xây dựng lại hoàn toàn nhằm mang đến hiệu suất cao, tính linh hoạt và khả năng mở rộng tốt hơn so với ASP.NET truyền thống. Đây là công nghệ hiện đại cho phép xây dựng các ứng dụng web, API, dịch vụ backend và ứng dụng thời gian thực.
- ASP.NET Core được thiết kế để chạy trên nhiều hệ điều hành như Windows, macOS và Linux, đồng thời hỗ trợ triển khai trên nền tảng đám mây như Azure hoặc máy chủ riêng. Với kiến trúc hướng modular, lập trình viên chỉ cần sử dụng các thành phần cần thiết, giúp giảm kích thước ứng dụng và tăng tốc độ xử lý. Ngoài ra, ASP.NET Core hỗ trợ Dependency Injection tích hợp, hệ thống routing mạnh mẽ, Razor Pages, SignalR và tích hợp dễ dàng với Entity Framework Core để quản lý dữ liệu.
- Với khả năng tích hợp tốt, hiệu năng cao và dễ bảo trì, ASP.NET Core đang là một trong những công nghệ hàng đầu được lựa chọn trong quá trình chuyển đổi số và phát triển các hệ thống thương mại điện tử, quản lý, và dịch vụ trực tuyến hiện đại.

Chương 2 Xây dựng ứng dụng Web (Triển khai thực tế)

2.1 Phát biểu bài toán ứng dụng

- Đặc biệt trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ. Ứng dụng web không chỉ đơn thuần là một trang thông tin, mà còn là công cụ tương tác giữa doanh nghiệp và khách hàng. Việc xây dựng một hệ thống web giúp tự động hóa các quy trình như bán hàng, đặt hàng, quản lý dữ liệu khách hàng, quản lý đơn hàng và thống kê doanh thu.
- Nhằm được những điều đó dự án xây dựng Web thương mại điện tử kinh doanh sản phẩm máy tính được sinh ra để giải quyết các bất lợi mà các cửa hàng truyền thống mang lại như giá cả, khoảng cách, v.v.
- Vai trò là cung cấp cho người dùng một nền tảng truy cập mọi lúc, mọi nơi chỉ cần có kết nối Internet. Với doanh nghiệp, ứng dụng web đóng vai trò là kênh bán hàng trực tuyến hiệu quả, tiết kiệm chi phí, đồng thời giúp mở rộng thị trường.
- Ý nghĩa của việc giải quyết bài toán này là góp phần nâng cao hiệu suất kinh doanh, cải thiện trải nghiệm khách hàng và bắt kịp xu hướng công nghệ hiện đại. Đây cũng là cơ hội để sinh viên, lập trình viên rèn luyện kỹ năng xây dựng hệ thống thực tế, áp dụng các công nghệ như ASP.NET Core, cơ sở dữ liệu, mô hình MVC,... để giải quyết một vấn đề cụ thể.

2.2 Phân tích yêu cầu của ứng dụng

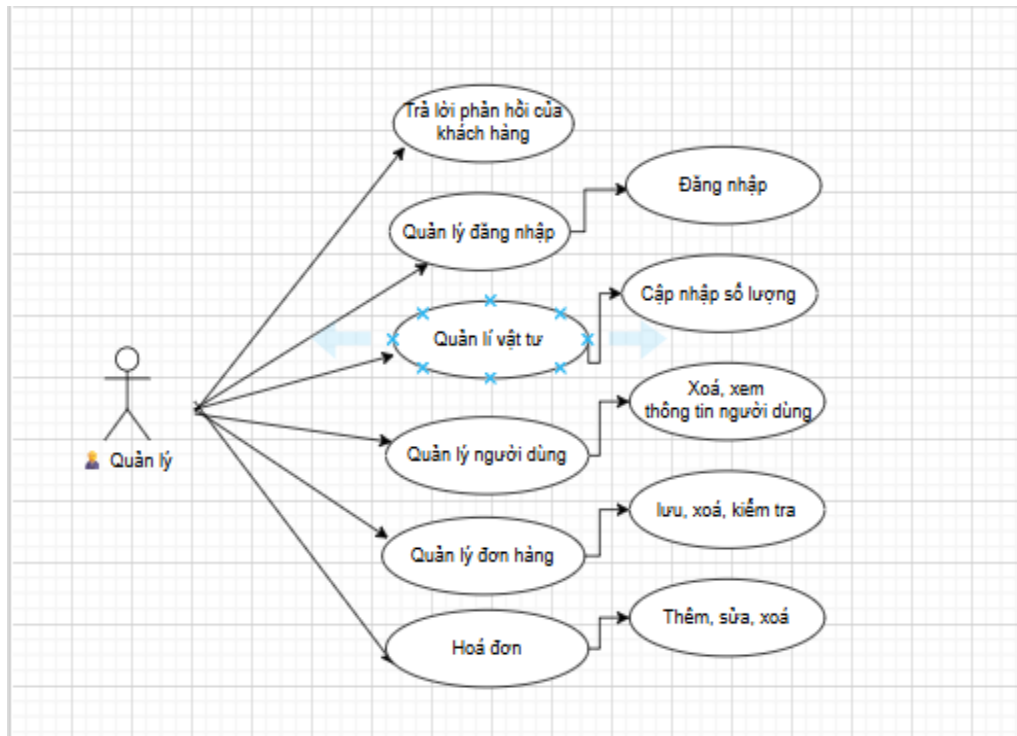
- Để đáp ứng nhu cầu quản lý và bán hàng trong thời đại chuyển đổi số, hệ thống website thương mại điện tử mà em xây dựng đã được thiết kế dựa theo mô hình Use-Case, giúp mô tả trực quan các chức năng chính của cả người dùng và quản trị viên. Thông qua sơ đồ Use-Case, các vai trò được phân định rõ ràng: Người dùng có thể thực hiện các thao tác như đăng ký tài khoản, xem và mua sản phẩm, quản lý đơn hàng; trong khi đó, Quản trị viên có thể giám sát toàn bộ hệ thống, từ quản lý người dùng, sản phẩm, đơn hàng cho đến trả lời phản hồi khách hàng. Mỗi chức năng trong hệ thống đều hướng đến mục tiêu nâng cao trải nghiệm người dùng, tối ưu quy trình quản lý và hỗ trợ doanh nghiệp vận hành hiệu quả hơn.

- Các yêu cầu chức năng dành cho quản trị viên:
 - + Đăng nhập hệ thống quản trị: chức năng này dành cho quản trị viên để truy cập các chức năng cục bộ.
 - + Quản lý vật tư: có các chức năng như thêm, sửa, xoá, cập nhập số lượng sản phẩm.
 - + Quản lý người dùng: xem, xoá thông tin người dùng.
 - + Quản lý đơn hàng: kiểm tra thông tin, lưu trạng thái và xoá các đơn hàng nếu cần thiết.
 - + Quản lý hoá đơn: tạo chỉnh sửa hoặc xoá các đơn hàng.
 - + Trả lời phản hồi của khách hàng: đọc và gửi phản hồi giải quyết các vấn đề mà khách hàng gặp phải.

- Các yêu cầu chức năng dành cho người dùng:
 - + Đăng ký, đăng nhập, đăng xuất tài khoản: người dùng có thể tạo tài khoản, đăng nhập vào hệ thống hoặc thoát.
 - + Xem danh sách sản phẩm: truy cập vào danh sách sản phẩm đang được bày bán.
 - + Xem chi tiết sản phẩm: hiển thị thông tin cụ thể về từng sản phẩm như mô tả, giá, hình ảnh, v.v.
 - + Thêm sản phẩm vào giỏ hàng: người dùng có thể chọn mua nhiều sản phẩm và lưu vào giỏ hàng.
 - + Kiểm tra giỏ hàng: xem danh sách sản phẩm đã chọn và hiển thị số lượng.
 - + Đặt hàng và kiểm tra đơn hàng: tiến hành đặt mua và theo dõi tình trạng đơn hàng.
 - + Thanh toán:

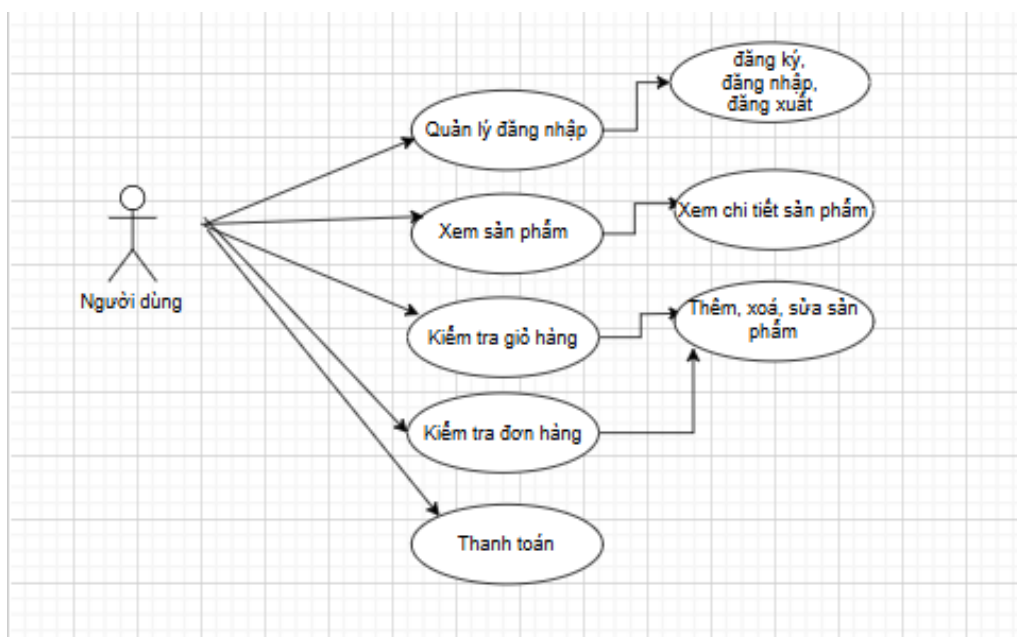
- Các yêu cầu phi chức năng
 - + Hiệu năng: có thể truy cập vào trang web nhanh và ổn định.
 - + Tính bảo mật: mọi dữ liệu đăng nhập, thanh toán phải được mã hoá, Chỉ người quản trị có quyền truy cập vào khu vực quản lý sản phẩm, đơn hàng. Các hành vi truy cập trái phép phải được cảnh báo.
 - + Khả năng mở rộng: hệ thống được thiết kế linh hoạt để có thể nâng cấp thêm tính năng mới.
 - + Tính ổn định: hệ thống phải hoạt động ổn định khi có lượt truy cập nhiều bất ngờ.
 - + Giao diện thân thiện: Giao diện người dùng thân thiện, rõ ràng, dễ thao tác với mọi độ tuổi và có khả năng hiển thị tốt trên cả máy tính và thiết bị di động.
 - + Khả năng bảo trì: mã lệnh phải được thiết kế rõ ràng, dễ mở rộng và bảo trì.

- Sơ đồ usecase của Quản Lý.



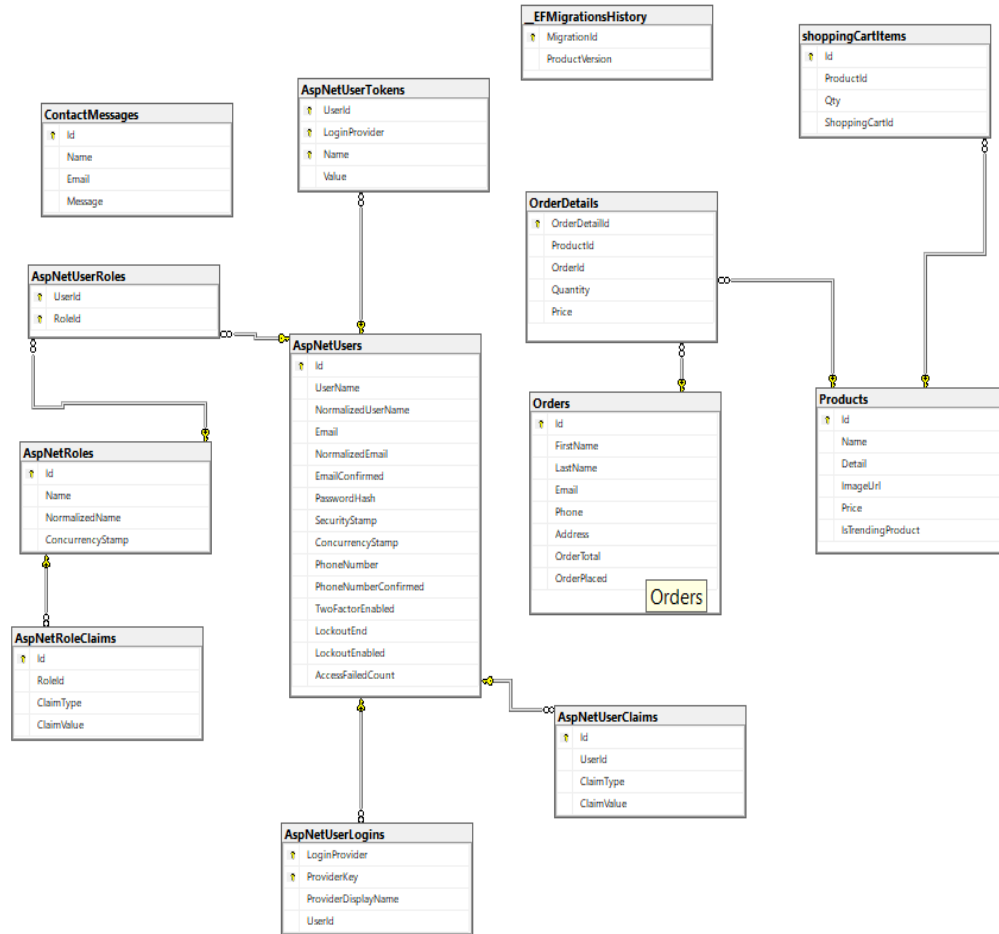
Hình 1. Sơ đồ usecase Quản lý.

- Sơ đồ usecase của Người dùng.



Hình 2. Sơ đồ usecase Người dùng.

2.3 Thiết kế cơ sở dữ liệu



Hình 3. Sơ đồ cơ sở dữ liệu

- Mô tả bảng AspNetUser.

	Column Name	Data Type	Allow Nulls
🔑	Id	nvarchar(450)	<input type="checkbox"/>
	UserName	nvarchar(256)	<input checked="" type="checkbox"/>
	NormalizedUserName	nvarchar(256)	<input checked="" type="checkbox"/>
	Email	nvarchar(256)	<input checked="" type="checkbox"/>
	NormalizedEmail	nvarchar(256)	<input checked="" type="checkbox"/>
	EmailConfirmed	bit	<input type="checkbox"/>
	PasswordHash	nvarchar(MAX)	<input checked="" type="checkbox"/>
	SecurityStamp	nvarchar(MAX)	<input checked="" type="checkbox"/>
	ConcurrencyStamp	nvarchar(MAX)	<input checked="" type="checkbox"/>
	PhoneNumber	nvarchar(MAX)	<input checked="" type="checkbox"/>
	PhoneNumberConfirmed	bit	<input type="checkbox"/>
	TwoFactorEnabled	bit	<input type="checkbox"/>
	LockoutEnd	datetimeoffset(7)	<input checked="" type="checkbox"/>
	LockoutEnabled	bit	<input type="checkbox"/>
	AccessFailedCount	int	<input type="checkbox"/>
			<input type="checkbox"/>

Hình 4. bảng AspNetUser

- Mô tả bảng __EFMigrationsHistory.

	Column Name	Data Type	Allow Nulls
🔑	MigrationId	nvarchar(150)	<input type="checkbox"/>
	ProductVersion	nvarchar(32)	<input type="checkbox"/>
			<input type="checkbox"/>

Hình 5. bảng __EFMigrationsHistory

- Mô tả bảngAspNetRoleClaims.

	Column Name	Data Type	Allow Nulls
▶ 🔑	Id	int	<input type="checkbox"/>
	RoleId	nvarchar(450)	<input type="checkbox"/>
	ClaimType	nvarchar(MAX)	<input checked="" type="checkbox"/>
	ClaimValue	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Hình 7. bảng AspNetRoleClaims.

- Mô tả bảng AspNetUserLogins.

	Column Name	Data Type	Allow Nulls
▶ 🔑	LoginProvider	nvarchar(128)	<input type="checkbox"/>
▶ 🔑	ProviderKey	nvarchar(128)	<input type="checkbox"/>
	ProviderDisplayName	nvarchar(MAX)	<input checked="" type="checkbox"/>
	UserId	nvarchar(450)	<input type="checkbox"/>
			<input type="checkbox"/>


Hình 8. Bảng AspNetUserLogins.

- Mô tả bảng AspNetRoles.

	Column Name	Data Type	Allow Nulls
▶ 🔑	Id	nvarchar(450)	<input type="checkbox"/>
	Name	nvarchar(256)	<input checked="" type="checkbox"/>
	NormalizedName	nvarchar(256)	<input checked="" type="checkbox"/>
	ConcurrencyStamp	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>



Hình 9. Bảng AspNetRoles.

- Mô tả bảngAspNetUserClaims.

	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	UserId	nvarchar(450)	<input type="checkbox"/>
	ClaimType	nvarchar(MAX)	<input checked="" type="checkbox"/>
	ClaimValue	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>




Hình 10. Bảng AspNetUserClaims.

- Mô tả bảng AspNetUserRoles.

	Column Name	Data Type	Allow Nulls
	UserId	nvarchar(450)	<input type="checkbox"/>
	RoleId	nvarchar(450)	<input type="checkbox"/>
			<input type="checkbox"/>


Hình 11. Bảng AspNetUserRoles.

- Bảng mô tả AspNetUserTokens.

	Column Name	Data Type	Allow Nulls
	UserId	nvarchar(450)	<input type="checkbox"/>
	LoginProvider	nvarchar(128)	<input type="checkbox"/>
	Name	nvarchar(128)	<input type="checkbox"/>
	Value	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>


Hình 12. Bảng AspNetUserTokens.

- Bảng mô tả ContactMessages.

	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Name	nvarchar(MAX)	<input type="checkbox"/>
	Email	nvarchar(MAX)	<input type="checkbox"/>
	Message	nvarchar(MAX)	<input type="checkbox"/>
			<input type="checkbox"/>


Hình 13. Bảng ContactMessages.

- Bảng mô tả OrderDetail.

	Column Name	Data Type	Allow Nulls
	OrderDetailId	int	<input type="checkbox"/>
	ProductId	int	<input type="checkbox"/>
	OrderId	int	<input type="checkbox"/>
	Quantity	int	<input type="checkbox"/>
	Price	decimal(18, 2)	<input type="checkbox"/>
			<input type="checkbox"/>


Hình 14. Bảng OrderDetail.

- Bảng mô tả Order.

	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	FirstName	nvarchar(MAX)	<input checked="" type="checkbox"/>
	LastName	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Email	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Phone	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Address	nvarchar(MAX)	<input checked="" type="checkbox"/>
	OrderTotal	decimal(18, 2)	<input type="checkbox"/>
	OrderPlaced	datetime2(7)	<input type="checkbox"/>
			<input type="checkbox"/>


Hình 15. Bảng Order.

- Bảng mô tả Product.

	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Detail	nvarchar(MAX)	<input checked="" type="checkbox"/>
	ImageUrl	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Price	decimal(18, 2)	<input type="checkbox"/>
	IsTrendingProduct	bit	<input type="checkbox"/>
			<input type="checkbox"/>

Hình 16. Bảng Product.

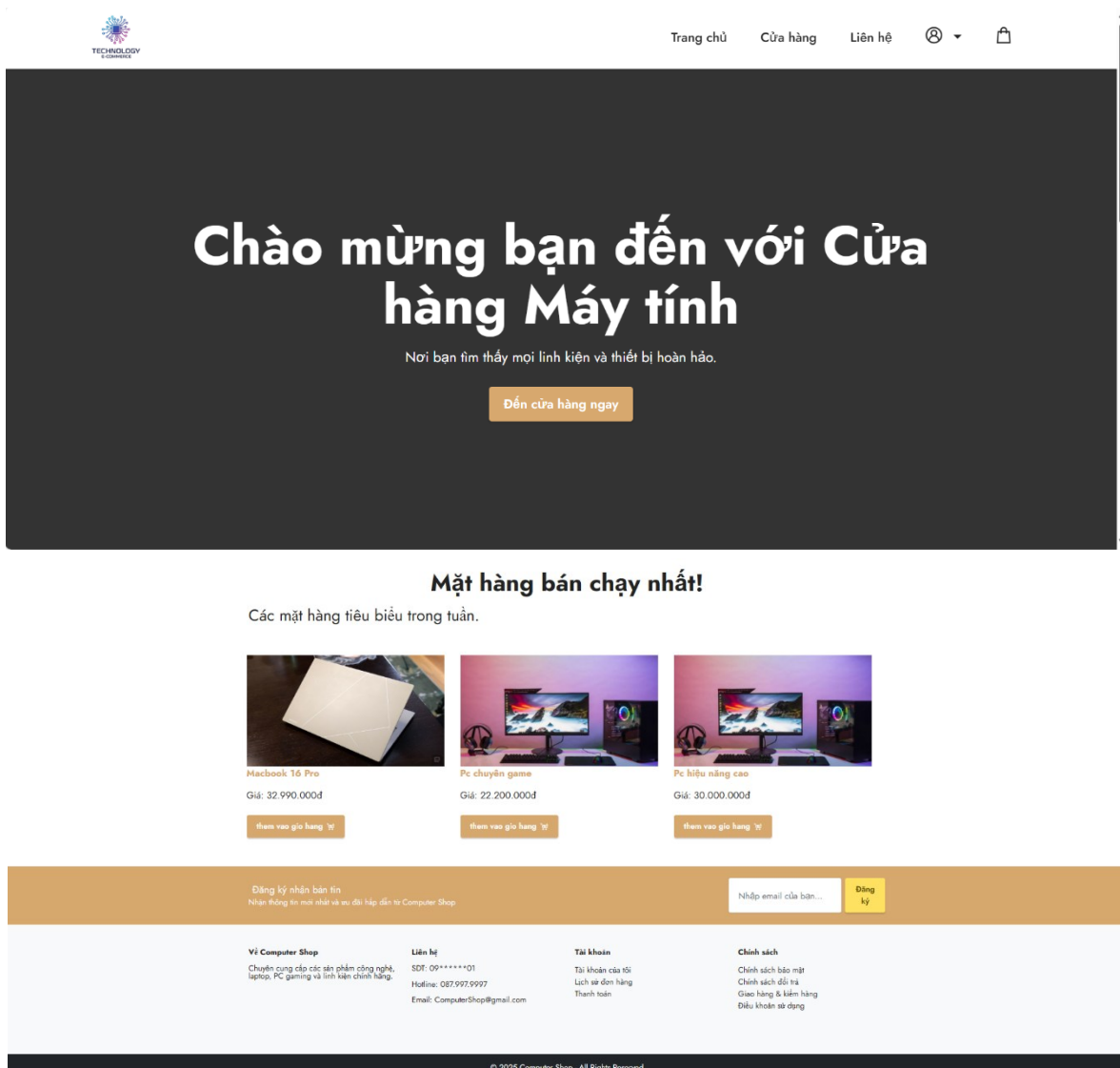
- Bảng mô tả ShoppingCartItems.

	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	ProductId	int	<input checked="" type="checkbox"/>
	Qty	int	<input type="checkbox"/>
	ShoppingCartId	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Hình 17. Bảng ShoppingCartItems.

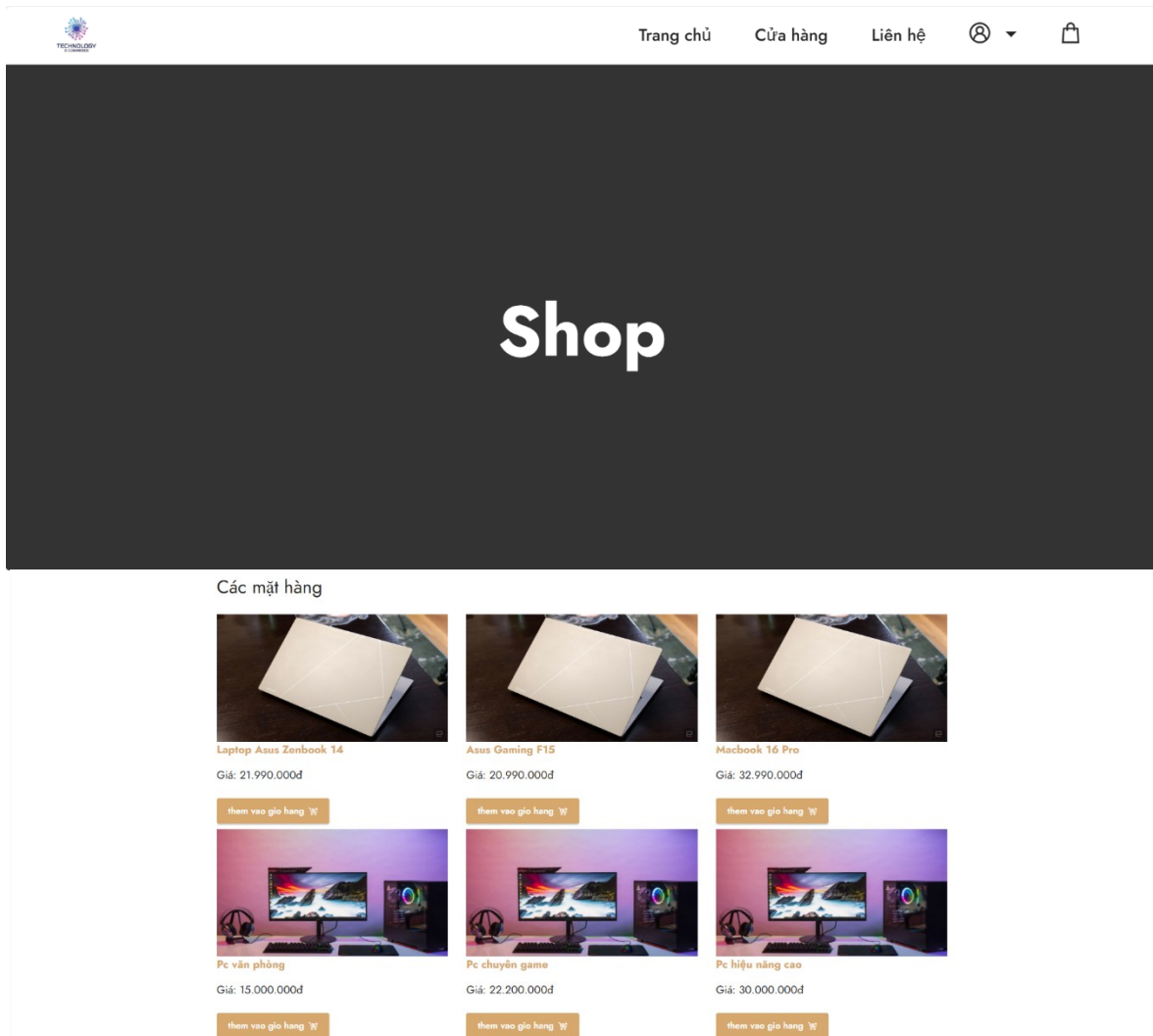
2.4 Thiết kế giao diện người dùng:

- Trang chủ.



Hình 18.Giao diện trang chủ.

- Trang Shop.



Hình 19. Giao diện trang shop.

- Trang liên hệ.

The screenshot shows a web page for 'Computer Shop' with a contact form. The header includes a logo, navigation links ('Trang chủ', 'Cửa hàng', 'Liên hệ'), and user information ('Hi thien@gmail.com'). The form is titled 'Liên hệ với Computer Shop' and contains the following elements:

- Họ và tên:** A text input field with the placeholder 'Nhập họ tên của bạn'.
- Email:** A text input field with the placeholder 'Nhập email của bạn'.
- Nội dung:** A text area with the placeholder 'Bạn muốn gửi gì?'.
- Gửi phản hồi:** A yellow button to submit the form.

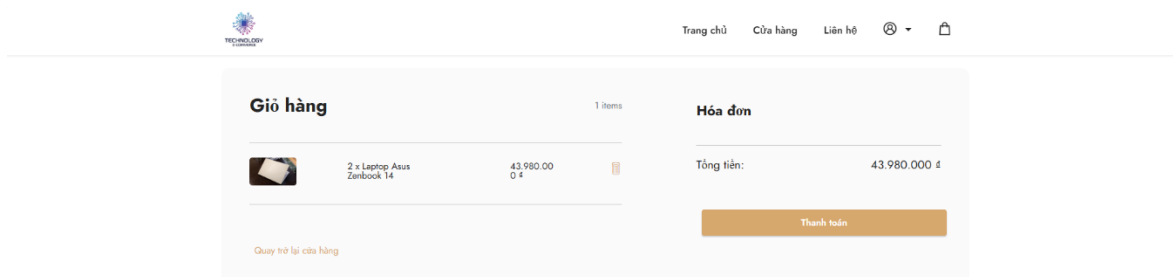
Hình 20. Giao diện trang liên hệ.

- Trang Mô tả.



Hình 21. Giao diện trang mô tả.

- Trang Giỏ hàng.



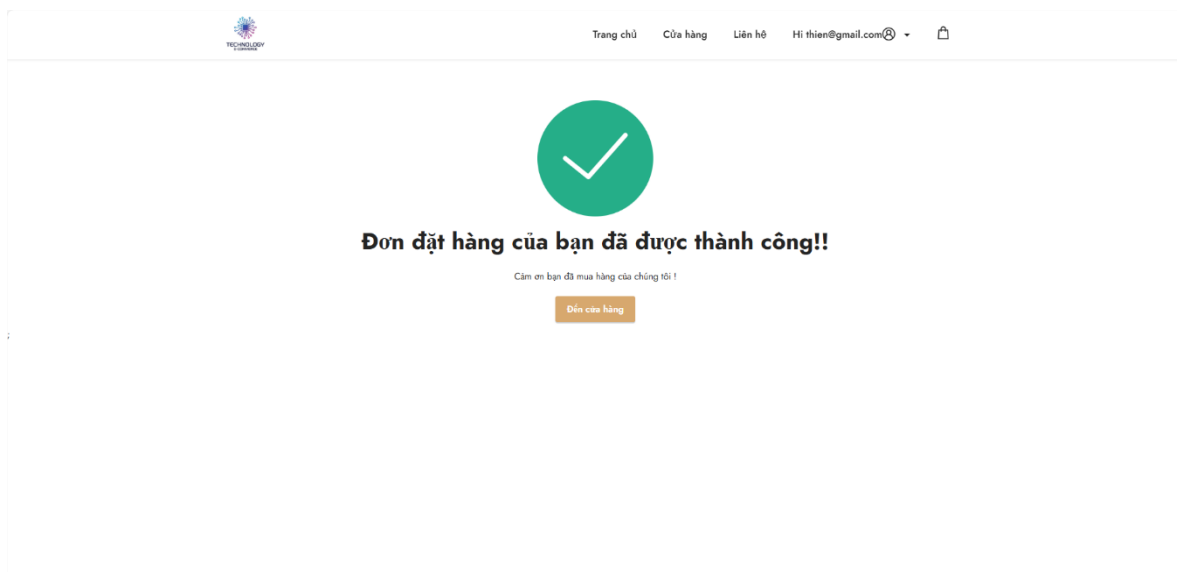
Hình 22. Giao diện trang giỏ hàng.

- Trang địa chỉ giao hàng.

The screenshot shows a web application header with a logo on the left and navigation links (Trang chủ, Cửa hàng, Liên hệ, Hi thien@gmail.com) and a shopping bag icon on the right. The main content area is titled 'Địa chỉ giao hàng' (Shipping Address). It contains a form with the following fields: 'Tên' (Name), 'Họ' (Surname), 'Email', 'SDT' (Phone Number), and 'Địa chỉ' (Address). The 'Địa chỉ' field is pre-filled with '123 Some Street Somewhere'. Below the form is an orange button labeled 'Đặt hàng' (Place Order).

Hình 23. Giao diện trang địa chỉ.

- Trang thanh toán thành công.



Hình 24. Giao diện trang thanh toán.

- Trang đăng nhập.

Trang chủ Cửa hàng Liên hệ

Log in

Sử dụng tài khoản bạn đã tạo để đăng nhập.

Email
thaco74605@gmail.com

Mật khẩu

☐ Remember me?

Đăng nhập

[Quên mật khẩu?](#)
[Tạo một tài khoản mới](#)
[Gửi lại mật khẩu qua email](#)

Hình 25. Giao diện trang đăng nhập.

- Trang đăng ký.

Trang chủ Cửa hàng Liên hệ

Register

Tạo một tài khoản mới tại đây.

Email

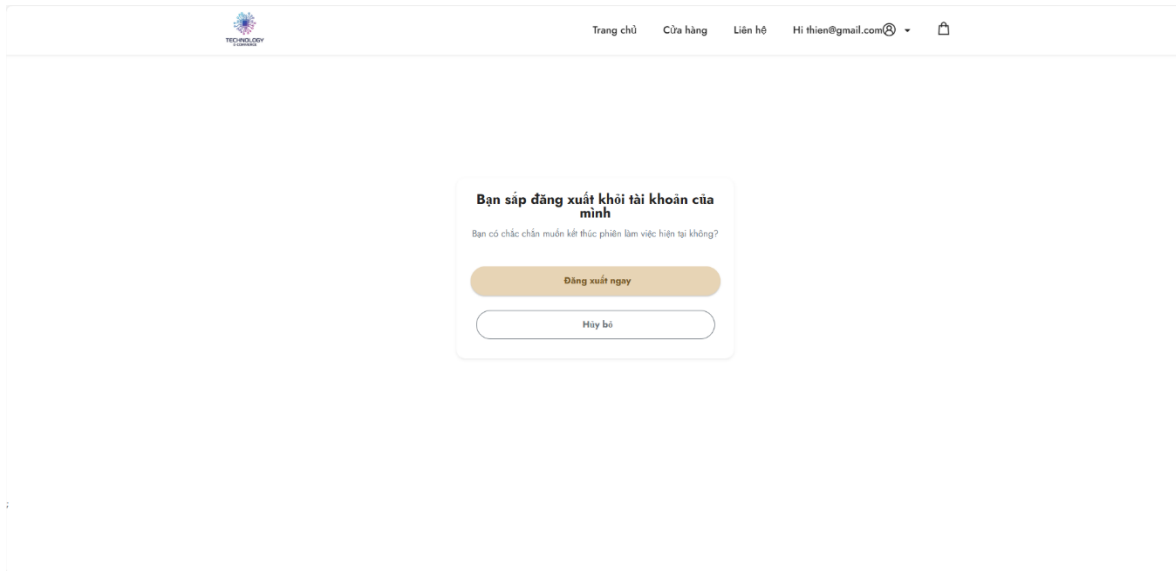
Mật khẩu

Nhập lại mật khẩu

Tạo

Hình 26. Giao diện trang đăng ký.

- Trang thoát khỏi đăng nhập.



Hình 27. Giao diện trang thoát.

2.5 Thiết kế các thành phần MVC:

- Model có nhiệm vụ đại diện cho dữ liệu sản phẩm và các mối quan hệ nghiệp vụ liên quan. Model được thiết kế để tương tác với cơ sở dữ liệu và cung cấp dữ liệu cho các thành phần Controller và View.

- Ví dụ Model:

```
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; }
}
```

- Lớp này đại diện cho sản phẩm với các thuộc tính cơ bản như:

- + Id: mã sản phẩm
- + Name: tên sản phẩm
- + Description: mô tả
- + Price: giá
- + ImageUrl: đường dẫn ảnh
- + CategoryId: khóa ngoại liên kết đến danh mục sản phẩm

- View đóng vai trò quan trọng trong việc hiển thị dữ liệu từ Model dưới dạng giao diện người dùng trực quan thông qua các công nghệ web như HTML, CSS và JavaScript. View không chứa bất kỳ logic nghiệp vụ phức tạp nào mà chỉ tập trung vào nhiệm vụ trình bày dữ liệu một cách rõ ràng và thân thiện với người dùng.
- Ví dụ cho View.

```

@model ComputerShopII.Models.ContactMessage
@{
    ViewData["Title"] = "Liên hệ";
}

<div class="container d-flex justify-content-center align-items-center my-5">
    <div class="col-md-8">
        <div class="text-center mb-4">
            <h2 class="fw-bold">Liên hệ với Computer Shop</h2>
            <p class="text-muted">Bạn có phản hồi, câu hỏi hoặc cần hỗ trợ? Hãy để lại tin nhắn cho chúng tôi!</p>
        </div>

        <div class="card shadow p-4">
            <form asp-action="ThankYou" asp-controller="Contact" method="post">
                <div class="mb-3">
                    <label asp-for="Name" class="form-label">Họ và tên</label>
                    <input asp-for="Name" class="form-control" placeholder="Nhập họ tên của bạn" />
                    <span asp-validation-for="Name" class="text-danger"></span>
                </div>

                <div class="mb-3">
                    <label asp-for="Email" class="form-label">Email</label>
                    <input asp-for="Email" type="email" class="form-control" placeholder="Nhập email của bạn" />
                    <span asp-validation-for="Email" class="text-danger"></span>
                </div>

                <div class="mb-3">
                    <label asp-for="Message" class="form-label">Nội dung</label>
                    <textarea asp-for="Message" class="form-control" rows="4" placeholder="Bạn muốn gửi gì?"></textarea>
                    <span asp-validation-for="Message" class="text-danger"></span>
                </div>

                <button type="submit" class="btn btn-warning w-100">Gửi phản hồi</button>
            </form>
        </div>
    </div>
</div>

```

Hình 28. View của trang liên hệ.

- Giao tiếp với Model
 - + Sử dụng `@model ComputerShopII.Models.ContactMessage` để kết nối với lớp dữ liệu `ContactMessage`, giúp ràng buộc dữ liệu form.
- Hiển thị giao diện người dùng
 - + Hiển thị các ô nhập liệu: họ tên, email, nội dung phản hồi.
 - + Các input có thuộc tính `asp-for="..."` nghĩa là chúng tự động liên kết với các thuộc tính trong model (`Name`, `Email`, `Message`).
- Xử lý và thông báo lỗi nhập liệu
 - + Các `asp-validation-for` hiển thị lỗi nếu người dùng không nhập đúng (ví dụ bỏ trống hoặc sai định dạng email).

- Controller đóng vai trò là bộ phận điều phối trung gian giữa View (giao diện người dùng) và Model (lớp xử lý dữ liệu). Sau khi nhận đầu vào từ người dùng (input từ view), controller sẽ xử lý yêu cầu được đưa ra từ phía người dùng, đó có thể bao gồm xác thực, chuyển đổi dữ liệu, sau đó gọi các phương thức thích hợp của Model để cập nhật trạng thái hoặc truy vấn dữ liệu, Chọn và truyền dữ liệu đến View để hiển thị kết quả tương ứng.
- Ví dụ về controller.

```

using ComputerShop11.Models.Interfaces;
using ComputerShop11.Models.Services;
using Microsoft.AspNetCore.Mvc;

namespace ComputerShop11.Controllers
{
    1 reference
    public class HomeController : Controller
    {
        private IProductRepository productRepository;
        0 references
        public HomeController(IProductRepository productRepository)
        {
            this.productRepository = productRepository;
        }
        0 references
        public IActionResult Index()
        {
            var randomProducts = productRepository.GetAllProducts()
                .OrderBy(x => Guid.NewGuid()) // Sắp xếp ngẫu nhiên
                .Take(3)                       // Lấy 3 sản phẩm
                .ToList();

            return View(randomProducts);
        }
    }
}

```

Hình 29. Controller của trang Home

- Controller sử dụng interface IProductRepository để lấy dữ liệu sản phẩm từ cơ sở dữ liệu mà không phụ thuộc vào cách hiện thực chi tiết.
- HomeController có nhiệm vụ chính là lấy ngẫu nhiên 3 sản phẩm từ kho dữ liệu thông qua IProductRepository, rồi truyền danh sách này sang View để người dùng thấy được trên trang chủ.

2.6 Triển khai và cài đặt

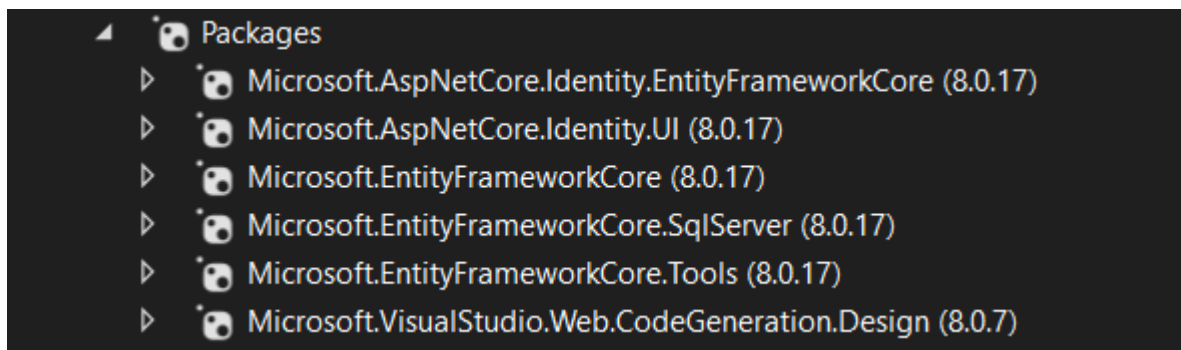
2.6.1 Môi trường triển khai.

- Hệ điều hành Window 11.
- Sử dụng Visual Studio 2022.
- NET SDK: .NET 8.0.
- SQL Server: SQL Server 2019.
- Giao diện người dùng:

- + HTML Cốt lõi để tạo cấu trúc các trang từ thanh menu, banner đến từng khối sản phẩm hay form.
- + CSS (Bootstrap 5) đảm nhiệm phần layout responsive, style chung.
- + JavaScript: Tạo các tương tác động.

2.6.2 Môi trường cài đặt.

- C# trên ASP.NET Core MVC:
 - + Controller tiếp nhận mọi HTTP request, điều phối giữa Model và View.
 - + Model định nghĩa các class (Product, Order, ContactMessage, v.v.) và ánh xạ qua Entity Framework Core đến SQL Server.
- Entity Framework Core: ORM giúp CRUD dữ liệu bằng LINQ, tự động sinh migration và đồng bộ schema.
- ASP.NET Core Identity: Quản lý đăng ký/đăng nhập/người dùng, với các bảng AspNetUsers, AspNetRoles, v.v.
- SQL Server 2019:
 - Lưu trữ bảng sản phẩm (Products), giỏ hàng (ShoppingCartItems), đơn hàng (Orders, OrderDetails) và bảng Identity.
- Đảm bảo tính toàn vẹn (foreign key), bảo mật (login, role) và hiệu năng truy vấn.
- Connection string cấu hình trong appsettings.json trỏ tới LocalDB hoặc instance riêng.
- Git & GitHub: Theo dõi lịch sử thay đổi, làm việc đa nhánh.
- Thư viện:



Hình 30. Các thư viện

Chương 3 Kết quả chương trình

Nhận xét và đánh giá.

- Quá trình xây dựng **ComputerShopII** đã thực sự là một hành trình học tập và ứng dụng thực tiễn mô hình MVC. Em đã:
 - + Xây dựng kiến trúc rõ ràng: Phân tách chặt chẽ giữa Controller (xử lý nghiệp vụ), Model (định nghĩa dữ liệu) và View (giao diện), giúp dự án dễ đọc, dễ mở rộng và bảo trì.
 - + Làm chủ EF Core: Sử dụng Entity Framework Core để thiết kế code-first, tự động tạo migration và thao tác CRUD, giúp rút ngắn thời gian phát triển và bảo đảm tính nhất quán của dữ liệu.
 - + Tích hợp ASP.NET Core Identity: Cung cấp tính năng đăng ký, đăng nhập, phân quyền cơ bản một cách an toàn, cho phép quản lý người dùng ngay từ đầu.
 - + Thiết kế giao diện responsive: Áp dụng Bootstrap và JavaScript để tạo trải nghiệm người dùng mượt mà trên đa thiết bị, từ desktop đến mobile.

Kết luận

1. Kết quả đạt được

- Trước hết, kiến trúc MVC được triển khai rất rõ ràng, tách biệt trách nhiệm giữa Model, View và Controller, giúp mã nguồn dễ đọc, bảo trì và mở rộng. Việc áp dụng Repository pattern qua các Interface/Service cũng minh bạch, thuận tiện cho việc mock và unit-test.
- Trên nền tảng ASP.NET Core kết hợp Entity Framework Core, dự án có hiệu năng cao với hỗ trợ async/await; đồng thời cơ chế Code-First và Migration tự động đảm bảo schema luôn đồng bộ và dễ dàng seed dữ liệu mẫu. Hệ thống xác thực ASP.NET Core Identity được tích hợp sẵn cho phép quản lý đăng ký, đăng nhập và phân quyền nhanh chóng, an toàn.
- Giao diện responsive dựa trên Bootstrap mang lại trải nghiệm mượt mà trên cả desktop lẫn di động, sử dụng thống nhất các component như navbar, card và form. Tính năng giỏ hàng và đặt hàng hoạt động thông qua session và repository, cho phép dễ dàng tùy chỉnh logic, đồng thời quy trình tính tổng và hiển thị hóa đơn rõ ràng.
- Cuối cùng, module phản hồi (Contact) với form liên hệ kèm trang “Thank You” đảm bảo trải nghiệm người dùng liền mạch, trong khi cơ chế Model validation ngăn chặn tình trạng bỏ trống thông tin.

2. Khuyết điểm, những điểm chưa thực hiện được.

Hiện tại dự án chỉ sử dụng lớp IdentityUser mặc định để quản lý người dùng, nhưng chưa có phân chia vai trò cụ thể là admin và user và thêm một số giao

diện khác có thể kể đến như bình luận của người dùng về sản phẩm, menu về từng loại sản phẩm, thanh tìm kiếm sản phẩm, và mục tự xây dựng PC.

3. Hướng mở rộng của đề tài

Trong thời gian tới em sẽ phải hoàn thiện dự án của mình hơn như bổ xung phân quyền người dùng thêm các chức cho người quản lý và người dùng, thêm các chức năng sản phẩm yêu thích, cho phép người dùng đăng bài đánh giá về sản phẩm và cải thiện khả năng đồng bộ hóa của website

Tài liệu tham khảo

1. Thiết kế trang web thương mại điện tử với MVC, Nguyễn Công Thiện, <https://github.com/thienstyle04/ComputerShop11.git>.
2. Tài liệu hướng dẫn về ASP.NET Core MVC, Nguyễn Đức Tấn.
3. Trang web tham khảo [TTGshop](#).

