

**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 XE ÔTÔ**

**GVHD: NGUYỄN ĐỨC TẤN**

**SVTH : Trần Thiện Em - 12201094**

**Khóa học : 20... - 20...**

**Đà Lạt, tháng 6 năm 2024**

## PHẦN NHẬN XÉT CỦA GIÁNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

**Điểm:**.....

Ngày ... tháng ... năm...

*Ký và ghi rõ họ tên*

## MỤC LỤC

DANH MỤC HÌNH .....	III
DANH MỤC TỪ NGỮ VIẾT TẮT .....	V
LỜI NÓI ĐẦU .....	VI
Chương 1 Tìm hiểu về lập trình Web (Cơ sở lý thuyết) .....	- 1 -
1.1 Khái niệm .....	- 1 -
1.2 Đặc điểm.....	- 3 -
1.3 Ưu điểm .....	- 4 -
1.4 Nhược điểm .....	- 4 -
1.5 Nguyên lý hoạt động .....	- 5 -
1.6 Phân tích MVC trong ASP.NET Core.....	- 7 -
1.7 So sánh MVC với các mô hình khác. ....	- 9 -
Chương 2 Xây dựng ứng dụng Web (Triển khai thực tế) .....	- 9 -
2.1 Bài toán ứng dụng.....	- 9 -
2.2 Xây dựng chương trình.....	- 17 -
Chương 3 Kết quả chương trình .....	- 19 -
Kết luận.....	- 25 -
1. Kết quả đạt được.....	- 25 -
2. Hướng nghiên cứu .....	- 25 -
Tài liệu tham khảo .....	- 26 -
Tài liệu chính: .....	- 26 -
Tài liệu kham khảo: .....	- 27 -

## DANH MỤC HÌNH

Hình ảnh	Mô tả	Trang
Hình 1	Sơ đồ hoạt động của mô hình MVC	5
Hình 2	Biểu đồ Use Case	10
Hình 3	Bảng __EFMigrationsHistory	11
Hình 4	BảngAspNetRoleClaims	11
Hình 5	BảngAspNetRoles	11
Hình 6	BảngAspNetUserClaims	12
Hình 7	BảngAspNetUserLogins	12
Hình 8	BảngAspNetUserRoles	12
Hình 9	BảngAspNetUsers	13
Hình 10	BảngAspNetUserTokens	14
Hình 11	Bảng Cars	14
Hình 12	Bảng Categories	15
Hình 13	Bảng Contacts	15
Hình 14	Bảng Orders	16
Hình 15	Bảng Products	16
Hình 16	Sơ đồ thực thể liên kết	17
Hình 17	Trang chủ	19
Hình 18	Sản phẩm	19
Hình 19	Liên hệ	20
Hình 20	Giới thiệu	20
Hình 21	Đăng nhập	21
Hình 22	So sánh sản phẩm	21
Hình 23	Đăng ký	22

Hình 24	Đặt hàng	23
Hình 25	Chi tiết hóa đơn	23
Hình 26	Thông tin đơn hàng	24
Hình 27	Tìm kiếm	24

## DANH MỤC TỪ NGỮ VIẾT TẮT

VIẾT TẮT	NỘI DUNG
MVC	Model-View-Controller
ASP.NET.Core	Active Server Pages .NET.Core
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
CSS	Cascading Style Sheets
SQL	Structured Query Language
JS	JavaScript
ORM	Object-Relational Mapping
URL	Uniform Resource Locator
CSDL	Cơ sở dữ liệu

## LỜI NÓI ĐẦU

Đầu tiên chúng em xin gửi lời cảm ơn chân thành đến thầy **Nguyễn Đức Tấn** giảng viên môn Lập trình web 1 khoa CNTT, trường Đại học Yersin Đà Lạt. Cảm ơn thầy giúp đỡ và trang bị cho chúng em những kiến thức cơ bản, nâng cao và kỹ năng cần thiết để chúng em hoàn thành đồ án **Xây dựng website thương mại điện tử kinh doanh xe oto.**

Tuy chúng em đã cố gắng rất nhiều trong quá trình làm đồ án nhưng do kiến thức chuyên ngành và kinh nghiệm còn hạn chế nên không thể tránh khỏi một vài thiếu sót cơ bản. Chúng em rất mong nhận được sự góp ý, lời khuyên cũng như sự đánh giá của thầy để có thể hoàn thiện đồ án tốt hơn.

Cuối cùng chúng em kính chúc thầy có nhiều sức khỏe và thành công hơn trong sự nghiệp giảng dạy của mình. Đồng thời kính chúc các thầy, cô trong trường luôn dồi dào sức khỏe và đạt được nhiều thành công tốt đẹp trong công việc. Chúng em trân trọng cảm ơn.

# **Chương 1 Tìm hiểu về lập trình Web (Cơ sở lý thuyết)**

## **1.1 Khái niệm**

### **1.1.1 MVC**

Mô hình Model-View-Controller (MVC) là một mẫu kiến trúc phân tách một ứng dụng thành ba thành phần logic chính Model, View và Controller. Do đó viết tắt MVC. Mỗi thành phần kiến trúc được xây dựng để xử lý khía cạnh phát triển cụ thể của một ứng dụng. MVC tách lớp logic nghiệp vụ và lớp hiển thị ra riêng biệt. Ngày nay, kiến trúc MVC đã trở nên phổ biến để thiết kế các ứng dụng web cũng như ứng dụng di động.

#### **MVC quan trọng bao gồm:**

- Model: Nó bao gồm tất cả dữ liệu và logic liên quan của nó.
- View: Trình bày dữ liệu cho người dùng hoặc xử lý tương tác của người dùng.
- Controller: Là phần quan trọng nhất trong mô hình, nó liên kết phần Model và View.

#### **View**

- View là một phần của ứng dụng đại diện cho việc trình bày dữ liệu.
- View được tạo bởi các dữ liệu mà chúng ta lấy từ dữ liệu trong model. Một view yêu cầu model cung cấp đầy đủ dữ liệu để nó hiển thị đầu ra cho người dùng.
- View chính là nơi chứa những giao diện như một nút bấm, khung nhập, menu, hình ảnh... nó đảm nhiệm nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống.



## **Controller**

- Controller là một phần của ứng dụng xử lý tương tác của người dùng. Bộ điều khiển diễn giải đầu vào chuột và bàn phím từ người dùng, thông báo cho model và view để thay đổi khi thích hợp.
- Controller là nơi tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó sẽ gồm những class/ function xử lý nhiều nghiệp vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ các nghiệp vụ lớp Model cung cấp và hiển thị dữ liệu đó ra cho người dùng nhờ lớp View.
- Controller gửi các lệnh đến model để làm thay đổi trạng thái của nó (Ví dụ: ta thêm mới 1 user hoặc cập nhật tên 1 user). Controller cũng gửi các lệnh đến view liên quan của nó để thay đổi cách hiển thị của view (Ví dụ: xem thông tin 1 user).

## **Model**

- Thành phần model lưu trữ dữ liệu và logic liên quan của nó. Bao gồm các class function xử lý các tác vụ như truy vấn, thêm, sửa hoặc xóa dữ liệu. Ví dụ, một đối tượng Controller sẽ lấy thông tin khách hàng từ cơ sở dữ liệu. Nó thao tác dữ liệu và gửi trở lại cơ sở dữ liệu hoặc sử dụng nó để hiển thị dữ liệu.

## **Sự tương tác giữa các thành phần**

- Controller tương tác với qua lại với View.
- Controller tương tác qua lại với Model.
- Model và View không có sự tương tác với nhau trực tiếp mà nó tương tác với nhau thông qua Controller.

**Ví dụ cho sự tương tác:** Khi người dùng ấn đăng nhập từ view thì request sẽ được gửi từ trình duyệt đến controller, controller sẽ gọi đến model xử lý logic và trả lại kết quả đó cho user thông qua view .

## **1.2 Đặc điểm**

- Phân tách rõ ràng ba thành phần: Model, View, Controller giúp tổ chức mã nguồn khoa học.
- Dễ mở rộng và dễ bảo trì do từng phần có thể thay đổi độc lập.
- Thường được sử dụng trong các website quy mô vừa và lớn, đặc biệt là các website thương mại điện tử như bán xe ô tô, giúp dễ dàng quản lý sản phẩm, khách hàng, đơn hàng.
- Hỗ trợ các công nghệ hiện đại như: Entity Framework, Web API, Razor Pages trong ASP.NET Core.

### **1.3 Ưu điểm**

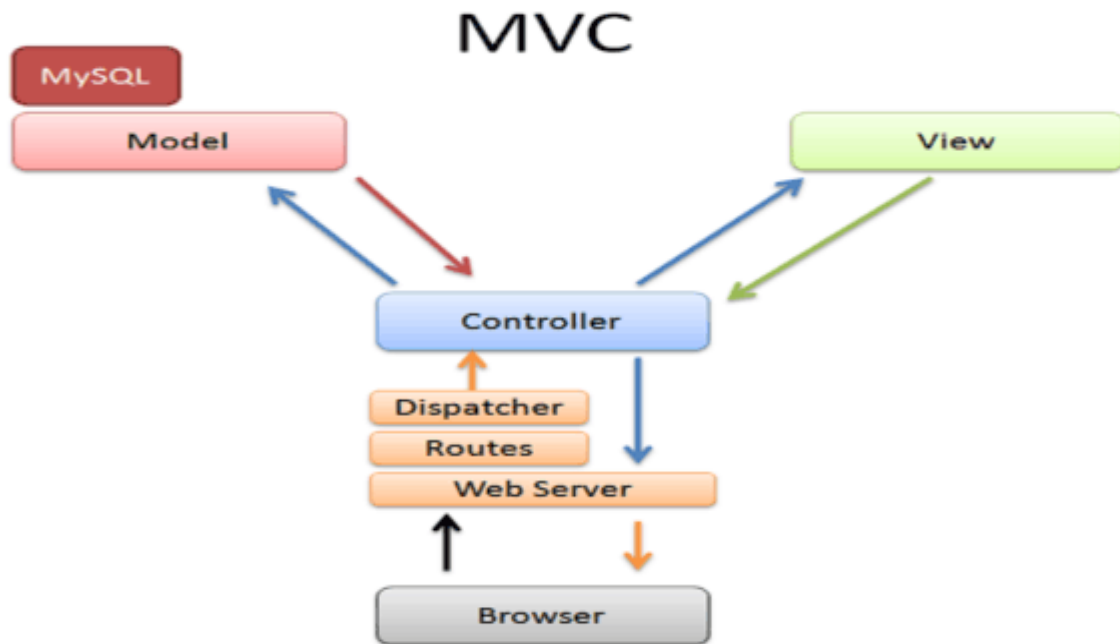
- Tăng khả năng mở rộng: Dễ dàng thêm mới tính năng (ví dụ: lọc xe theo giá, thương hiệu).
- Tách biệt giao diện và xử lý dữ liệu: Frontend (View) và Backend (Model, Controller) có thể phát triển song song.
- Hỗ trợ kiểm thử tốt: Có thể kiểm thử riêng Controller, Model mà không cần giao diện View.
- Tái sử dụng mã nguồn: Một Model hoặc Controller có thể phục vụ nhiều View khác nhau (ví dụ: cả website và mobile app).
- Tối ưu hiệu suất phát triển nhóm: Nhiều nhóm lập trình có thể làm việc trên các thành phần riêng biệt mà không gây xung đột mã nguồn.

### **1.4 Nhược điểm**

- Việc tách thành ba phần riêng biệt (Model, View, Controller) khiến dự án nhiều file, nhiều lớp hơn, gây dư thừa cho các ứng dụng nhỏ hoặc đơn giản.
- Khi một thành phần thay đổi (như Model), các phần còn lại có thể phải điều chỉnh theo, dễ dẫn đến lỗi hoặc thiếu đồng nhất nếu không kiểm soát tốt.
- Controller dễ bị “béo” (Fat Controller)  
Nếu lập trình viên đưa quá nhiều logic vào Controller thay vì Model, Controller trở nên cồng kềnh, khó bảo trì, đi ngược lại mục tiêu phân tách ban đầu.
- Do phải qua nhiều lớp trung gian, việc tối ưu tốc độ xử lý hoặc cache phức tạp hơn, yêu cầu kỹ thuật cao để đảm bảo hiệu quả.
- Dự án lớn với nhiều Model, View, Controller đòi hỏi bảo trì đồng bộ, dễ phát sinh lỗi ngầm và cần quy ước chặt chẽ trong nhóm phát triển.
- MVC không lý tưởng cho các ứng dụng yêu cầu xử lý thời gian thực (chat, game online) mà phải kết hợp thêm mô hình hoặc công nghệ khác.

## 1.5 Nguyên lý hoạt động

Mô hình MVC (Model - View - Controller) là một kiến trúc phần mềm phổ biến giúp tách biệt ba thành phần chính trong ứng dụng: Dữ liệu (Model), Giao diện (View) và Xử lý điều khiển (Controller). Sự phân tách này nhằm tăng khả năng bảo trì, mở rộng, kiểm thử và tái sử dụng mã nguồn.



Hình 1. Sơ đồ hoạt động của mô hình MVC

### 1. Thành phần Model (Mô hình dữ liệu)

**Vai trò:** Chịu trách nhiệm quản lý dữ liệu, trạng thái và các logic nghiệp vụ cốt lõi của ứng dụng.

**Chức năng:**

- Truy xuất và cập nhật dữ liệu từ/đến cơ sở dữ liệu.
- Xử lý các quy tắc nghiệp vụ (business logic).
- Không quan tâm đến cách dữ liệu được hiển thị cho người dùng.

## **2. Thành phần View (Giao diện hiển thị)**

**Vai trò:** Chịu trách nhiệm hiển thị dữ liệu cho người dùng dưới dạng giao diện trực quan.

### **Chức năng:**

- Nhận dữ liệu từ Model (thông qua Controller) để trình bày.
- Không xử lý nghiệp vụ hay dữ liệu trực tiếp.
- Không tự cập nhật dữ liệu – luôn phụ thuộc vào Controller.

## **3. Thành phần Controller (Điều khiển)**

**Vai trò:** Là cầu nối giữa View và Model; tiếp nhận yêu cầu (request) từ người dùng và quyết định hành động xử lý.

### **Chức năng:**

- Nhận và phân tích các yêu cầu từ người dùng (thông qua View).
- Tương tác với Model để xử lý dữ liệu.
- Lựa chọn View phù hợp để trả kết quả về cho người dùng.

## **1.6 Phân tích MVC trong ASP.NET Core**

### **1.6.1 Giới thiệu về ASP.NET Core**

ASP.NET Core là một nền tảng phát triển ứng dụng web mã nguồn mở, đa nền tảng (cross-platform), do Microsoft phát triển. Đây là phiên bản cải tiến của ASP.NET trước đây, hỗ trợ xây dựng các ứng dụng web hiện đại, API dịch vụ, và ứng dụng real-time với hiệu năng cao. ASP.NET Core tích hợp mô hình MVC để giúp tách biệt rõ ràng các thành phần dữ liệu, giao diện và xử lý logic, giúp dự án dễ bảo trì, mở rộng và kiểm thử.

### **1.6.2 Cấu trúc và vai trò của Model trong ASP.NET Core MVC**

#### **Cấu trúc:**

- Thường đặt trong thư mục Models/
- Là các lớp C# đại diện cho dữ liệu, như: thực thể cơ sở dữ liệu (Entity), lớp ViewModel hoặc lớp dữ liệu trung gian.

#### **Vai trò**

- Quản lý dữ liệu và logic nghiệp vụ.
- Xử lý các quy tắc nghiệp vụ (validation, tính toán...).
- Làm việc với cơ sở dữ liệu thông qua ORM (Entity Framework Core).
- Không chứa mã điều khiển hay mã giao diện.

### **1.6.3 Cấu trúc và vai trò của View trong ASP.NET Core MVC**

#### **Cấu trúc:**

- Thường nằm trong thư mục Views/, mỗi Controller có một thư mục View riêng.
- Sử dụng Razor (.cshtml) để kết hợp HTML và C#.

#### **Vai trò:**

- Hiển thị dữ liệu lên giao diện cho người dùng.
- Không xử lý logic nghiệp vụ hoặc truy xuất cơ sở dữ liệu.
- Nhận dữ liệu từ Controller (qua ViewModel hoặc Model).

#### **1.6.4 Cấu trúc và vai trò của Controller trong ASP.NET Core MVC**

##### **Cấu trúc:**

- Nằm trong thư mục Controllers/.
- Kế thừa từ lớp Controller hoặc ControllerBase.

##### **Vai trò:**

- Tiếp nhận và xử lý các yêu cầu HTTP từ người dùng.
- Tương tác với Model để lấy hoặc cập nhật dữ liệu.
- Trả về View (hoặc dữ liệu JSON đối với API).
- Điều hướng luồng dữ liệu giữa View và Model.

#### **1.6.5 Luồng dữ liệu và tương tác giữa các thành phần trong ASP.NET Core MVC**

- Người dùng gửi yêu cầu (HTTP Request) đến ứng dụng (ví dụ: nhập URL hoặc nhấn nút).
- Controller tiếp nhận yêu cầu, xử lý tham số, tương tác với Model để lấy hoặc cập nhật dữ liệu.
- Model thực hiện xử lý dữ liệu (truy vấn cơ sở dữ liệu hoặc áp dụng logic nghiệp vụ).
- Controller nhận dữ liệu từ Model, chọn View phù hợp và truyền dữ liệu cần hiển thị.
- View nhận dữ liệu, tạo HTML động và trả về trình duyệt để hiển thị cho người dùng (HTTP Response).

### **1.7 So sánh MVC với các mô hình khác.**

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

### **2.1 Bài toán ứng dụng**

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

Trong bối cảnh nền kinh tế ngày càng phát triển, thu nhập và nhu cầu sở hữu xe ô tô của người dân không ngừng gia tăng, thị trường mua bán xe ô tô đang trở nên sôi động và cạnh tranh hơn bao giờ hết. Đặc biệt, xu hướng tiêu dùng hiện đại đã dần dịch chuyển từ hình thức mua bán truyền thống sang môi trường trực tuyến.

Người tiêu dùng hiện nay có xu hướng tìm kiếm, so sánh, tham khảo thông tin và thậm chí ra quyết định mua xe ngay trên các nền tảng thương mại điện tử thay vì trực tiếp đến showroom. Điều này đòi hỏi các doanh nghiệp kinh doanh xe ô tô cần đầu tư nghiêm túc vào các kênh bán hàng trực tuyến để mở rộng thị trường, tiếp cận khách hàng tiềm năng và nâng cao năng lực cạnh tranh.

Xuất phát từ thực tiễn đó, bài toán đặt ra là xây dựng một hệ thống website thương mại điện tử chuyên biệt cho lĩnh vực kinh doanh xe ô tô với đầy đủ các chức năng đáp ứng nhu cầu của cả khách hàng và nhà quản trị hệ thống.

#### **2.1.2 Xác định yêu cầu của ứng dụng**

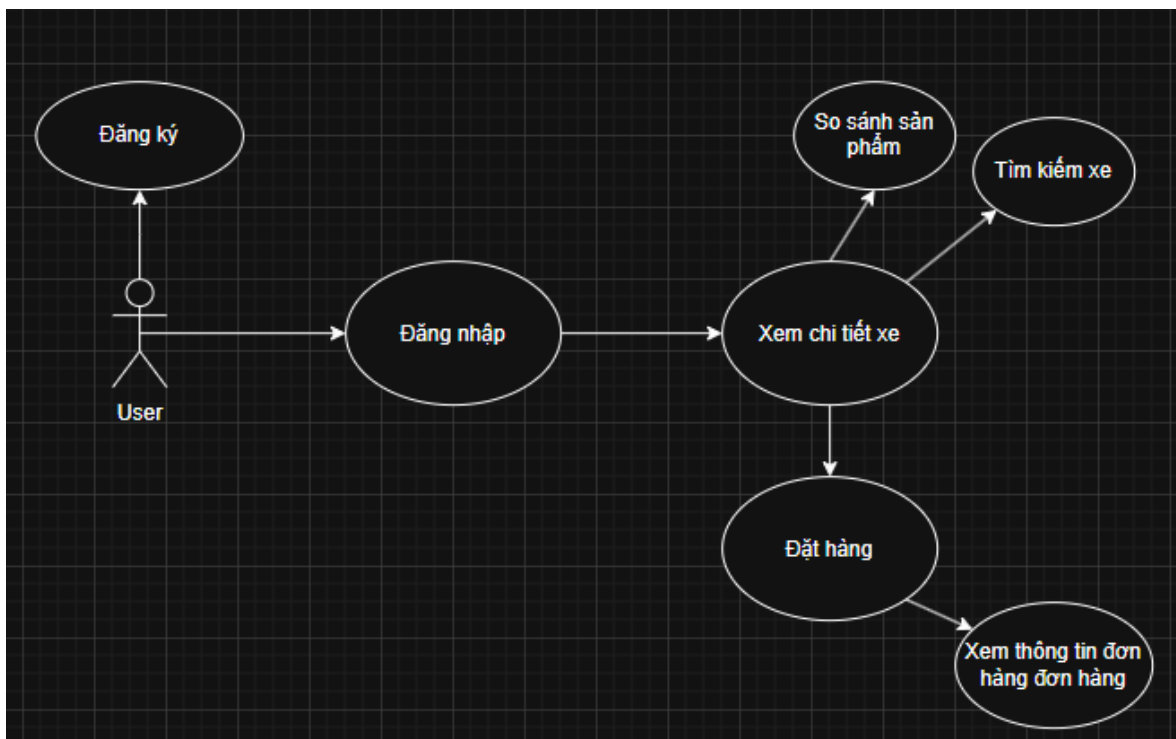
Về yêu cầu của ứng dụng thì có rất là nhiều, như là :

- Chức năng Đăng ký tài khoản Cho phép người dùng mới tạo tài khoản trên hệ thống bằng cách cung cấp các thông tin cần thiết như tên đăng nhập, email, mật khẩu. Sau khi đăng ký thành công, người dùng có thể sử dụng tài khoản để truy cập các tính năng khác.
- Chức năng Đăng nhập hệ thống Cung cấp giao diện để người dùng đã có tài khoản nhập tên đăng nhập (hoặc email) và mật khẩu để truy cập vào các tính năng dành riêng cho người dùng đã đăng nhập, như đặt hàng, xem lịch sử đơn hàng.



- Chức năng Đặt hàng Cho phép người dùng chọn xe ô tô, cấu hình các tùy chọn (nếu có), và tiến hành đặt mua trực tuyến. Hệ thống sẽ xác nhận đơn hàng và lưu thông tin vào cơ sở dữ liệu.
- Chức năng Chi tiết đơn hàng Hiển thị đầy đủ thông tin của một đơn hàng cụ thể, bao gồm thông tin xe, các tùy chọn đã chọn, tổng giá trị đơn hàng, trạng thái xử lý đơn hàng, và ngày đặt hàng.
- Chức năng Danh sách đơn hàng Cung cấp cái nhìn tổng quan về tất cả các đơn hàng mà người dùng đã đặt hàng. Cho phép xem nhanh trạng thái và các thông tin cơ bản của từng đơn hàng.
- Chức năng So sánh sản phẩm Cho phép người dùng chọn hai hoặc nhiều mẫu xe ô tô để đặt cạnh nhau, hiển thị các thông số kỹ thuật, tính năng, giá cả, và hình ảnh để người dùng dễ dàng đối chiếu và đưa ra quyết định mua sắm.

### 2.1.3 Biểu đồ Use Case



Hình 2. Biểu đồ Use Case

### 2.1.4 Xây dựng Cơ sở dữ liệu

- EFMigrationsHistory: Giúp Entity Framework Core theo dõi và quản lý các lần **migrate** cơ sở dữ liệu. Bảng này được tạo tự động và không cần chỉnh sửa thủ công, nhằm đảm bảo cơ sở dữ liệu luôn đồng bộ với mô hình dữ liệu trong ứng dụng.

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

**Hình 3. Bảng \_\_EFMigrationsHistory**

- AspNetRoleClaims: Lưu trữ các claim (quyền đặc biệt) được gán cho từng vai trò (role) trong hệ thống. Bảng này là một phần trong ASP.NET Core Identity, giúp mở rộng khả năng phân quyền người dùng theo vai trò chi tiết hơn.

TranThienEm.CarSh....AspNetRoleClaims			
	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"/>


**Hình 4. Bảng AspNetRoleClaims**

- AspNetRoles: Lưu trữ thông tin về các vai trò (roles) trong hệ thống người dùng. Bảng này thuộc hệ thống ASP.NET Core Identity, cho phép gán quyền truy cập theo từng nhóm vai trò như "Admin", "User", "Manager".

TranThienEm.CarSh...- dbo.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"/>




**Hình 5. Bảng AspNetRoles**

- **AspNetUserClaims:** Lưu trữ các claim cá nhân được gán cho từng người dùng cụ thể. Đây là bảng thuộc hệ thống ASP.NET Core Identity, giúp mở rộng khả năng phân quyền chi tiết hơn ngoài vai trò.

TranThienEm.CarSh....AspNetUserClaims		TranThienEm.CarSh...- dbo.AspNetRoles	
	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"/>



**Hình 6. Bảng AspNetUserClaims**

- **AspNetUserLogins:** Lưu trữ thông tin đăng nhập bên ngoài (external login) của người dùng, như đăng nhập qua Google, Facebook, Microsoft, Bảng này thuộc hệ thống ASP.NET Core Identity.

TranThienEm.CarSh....AspNetUserLogins		TranThienEm.CarSh....AspNetUserClaims	
	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 7. Bảng AspNetUserLogins**

- **AspNetUserRoles:** Lưu trữ mối quan hệ giữa người dùng và vai trò. Đây là bảng liên kết n-n giữa bảng AspNetUsers và AspNetRoles trong hệ thống ASP.NET Core Identity.

TranThienEm.CarSh...o.AspNetUserRoles		TranThienEm.CarSh....AspNetUserLogins	
	Column Name	Data Type	Allow Nulls
	UserId	nvarchar(450)	<input type="checkbox"/>
	RoleId	nvarchar(450)	<input type="checkbox"/>

**Hình 8. Bảng AspNetUserRoles**

- **AspNetUsers**: Lưu trữ toàn bộ thông tin người dùng trong hệ thống. Đây là bảng trung tâm trong ASP.NET Core Identity, phục vụ cho chức năng xác thực và phân quyền người dùng..

TranThienEm.CarSh...- dbo.AspNetUsers		TranThienEm.CarSh...o.AspNetUserRoles	
	Column Name	Data Type	Allow Nulls
	Id	nvarchar(450)	<input type="checkbox"/>
	FullName	nvarchar(100)	<input checked="" type="checkbox"/>
	Address	nvarchar(200)	<input checked="" type="checkbox"/>
	CreatedDate	datetime2(7)	<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"/>

**Hình 9. Bảng AspNetUsers**

- AspNetUserTokens: Lưu trữ các token xác thực hoặc thông tin bổ sung liên quan đến người dùng, thường dùng trong các chức năng như "Remember Me", xác thực 2 bước, hoặc token bên ngoài. Đây là một phần của hệ thống ASP.NET Core Identity.

TranThienEm.CarSh....AspNetUserTokens				TranThienEm.CarSh...- dbo.AspNetUsers			
	Column Name	Data Type	Allow Nulls		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"/>				


**Hình 10. Bảng AspNetUserTokens**

- Cars: Lưu trữ thông tin chi tiết về các xe ô tô đang được rao bán trên hệ thống. Đây là bảng chính trong cơ sở dữ liệu của website bán ô tô, phục vụ cho việc tìm kiếm, hiển thị và quản lý sản phẩm.

TranThienEm.CarShopDB - dbo.Cars				TranThienEm.CarShopDB - dbo.Table_1			
	Column Name	Data Type	Allow Nulls		Column Name	Data Type	Allow Nulls
🔑	<b>Id</b>	int	<input type="checkbox"/>				
	Name	nvarchar(100)	<input type="checkbox"/>				
	BrandId	int	<input type="checkbox"/>				
	CategoryId	int	<input type="checkbox"/>				
	Year	int	<input type="checkbox"/>				
	Price	decimal(18, 2)	<input type="checkbox"/>				
	Mileage	int	<input type="checkbox"/>				
	FuelType	nvarchar(50)	<input checked="" type="checkbox"/>				
	Transmission	nvarchar(50)	<input checked="" type="checkbox"/>				
	Color	nvarchar(50)	<input checked="" type="checkbox"/>				
	Description	nvarchar(1000)	<input checked="" type="checkbox"/>				
	ImageUrl	nvarchar(500)	<input checked="" type="checkbox"/>				
	IsAvailable	bit	<input type="checkbox"/>				
	CreatedDate	datetime2(7)	<input type="checkbox"/>				
	Location	nvarchar(50)	<input checked="" type="checkbox"/>				
	ContactPhone	nvarchar(20)	<input checked="" type="checkbox"/>				
	IsFeatured	bit	<input type="checkbox"/>				
			<input type="checkbox"/>				


**Hình 11. Bảng Cars**

- Categories: Đây là bảng dùng để lưu trữ thông tin về các danh mục (Categories) của xe ô tô trong hệ thống. Mỗi danh mục đại diện cho một loại xe nhất định, giúp phân loại và tổ chức các sản phẩm xe một cách có hệ thống, phục vụ cho việc tìm kiếm và lọc xe trên trang web.

TranThienEm.CarSh...B - dbo.Categories		TranThienEm.CarShopDB - dbo.Cars	
	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Name	nvarchar(50)	<input type="checkbox"/>
	Description	nvarchar(200)	<input type="checkbox"/>

**Hình 12. Bảng Categories**

- Contact: Đây là bảng dùng để lưu trữ thông tin liên hệ được gửi từ các biểu mẫu liên hệ trên trang web. Bảng này giúp hệ thống ghi nhận các yêu cầu, thắc mắc hoặc phản hồi từ người dùng, từ đó đội ngũ hỗ trợ có thể liên hệ lại và giải quyết.

TranThienEm.CarShopDB - dbo.Contacts		TranThienEm.CarSh...B - dbo.Categories	
	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Name	nvarchar(MAX)	<input type="checkbox"/>
	Email	nvarchar(MAX)	<input type="checkbox"/>
	Phone	nvarchar(MAX)	<input type="checkbox"/>
	Message	nvarchar(MAX)	<input type="checkbox"/>
	CreatedAt	datetime2(7)	<input type="checkbox"/>

**Hình 13. Bảng Contact**

- Orders: Đây là bảng trung tâm dùng để lưu trữ thông tin về các đơn đặt hàng xe ô tô của khách hàng trên hệ thống. Bảng này ghi nhận chi tiết mỗi giao dịch đặt xe, bao gồm thông tin khách hàng, sản phẩm được đặt, thời gian và trạng thái của đơn hàng.

TranThienEm.CarShopDB - dbo.Orders		TranThienEm.CarShopDB - dbo.Orders	
	Column Name	Data Type	Allow Nulls
🔑	Id	int	<input type="checkbox"/>
	CarId	int	<input type="checkbox"/>
	FullName	nvarchar(MAX)	<input type="checkbox"/>
	Phone	nvarchar(MAX)	<input type="checkbox"/>
	Email	nvarchar(MAX)	<input type="checkbox"/>
	Address	nvarchar(MAX)	<input type="checkbox"/>
	Note	nvarchar(MAX)	<input checked="" type="checkbox"/>
	OrderDate	datetime2(7)	<input type="checkbox"/>
	Status	int	<input type="checkbox"/>
	CancelledDate	datetime2(7)	<input checked="" type="checkbox"/>
	CancelReason	nvarchar(MAX)	<input checked="" type="checkbox"/>

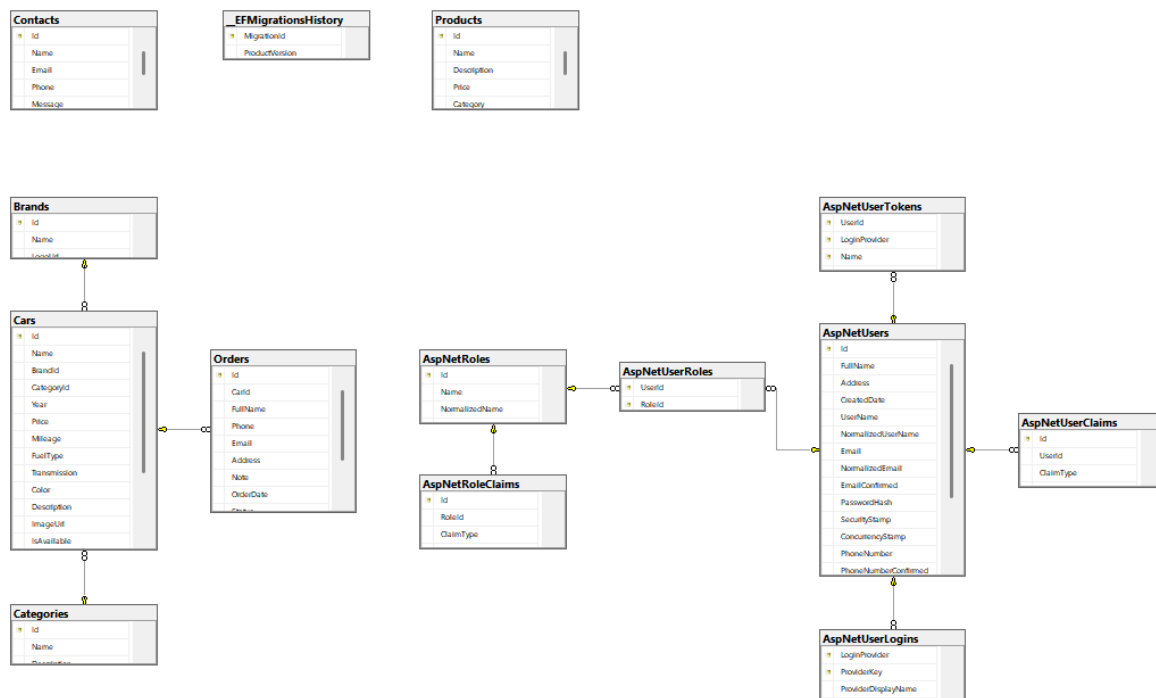
**Hình 14. Bảng Orders**

- Products: Đây là bảng dùng để lưu trữ thông tin chi tiết về các sản phẩm (xe ô tô) được bán trên trang thương mại điện tử. Bảng này chứa các thông số cơ bản giúp mô tả mỗi chiếc xe một cách đầy đủ, phục vụ cho việc hiển thị sản phẩm, tìm kiếm và quản lý kho hàng.

TranThienEm.CarShopDB - dbo.Products		TranThienEm.CarShopDB - dbo.Orders	
	Column Name	Data Type	Allow Nulls
🔑	Id	int	<input type="checkbox"/>
	Name	nvarchar(100)	<input type="checkbox"/>
	Description	nvarchar(200)	<input type="checkbox"/>
	Price	decimal(18, 2)	<input type="checkbox"/>
	Category	nvarchar(100)	<input type="checkbox"/>
	ImageUrl	nvarchar(MAX)	<input type="checkbox"/>
▶			<input type="checkbox"/>

**Hình 15. Bảng Products**

Sơ đồ thực thể liên kết:



Hình 16. Sơ đồ thực thể liên kết

## 2.2 Xây dựng chương trình

Chúng em xây dựng chương trình dựa trên ngôn ngữ C#, SQL Server, framework ASP.NET CORE ,...

C# là một ngôn ngữ lập trình phát triển mạnh về hướng đối tượng của Microsoft. Với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc phát triển ứng dụng Web, ứng dụng Windows Forms, Game hay Mobile một cách dễ dàng.

ASP.NET MVC là một framework phát triển web trong .NET Framework, được thiết kế để xây dựng các ứng dụng web động và mạnh mẽ. Nó tuân thủ kiến trúc Model-View-Controller (MVC), giúp tách biệt rõ ràng ba thành phần chính của ứng dụng:

- Model: Đại diện cho dữ liệu và logic nghiệp vụ.
- View: Chịu trách nhiệm hiển thị giao diện người dùng.
- Controller: Xử lý yêu cầu của người dùng, tương tác với Model và chọn View phù hợp để hiển thị.

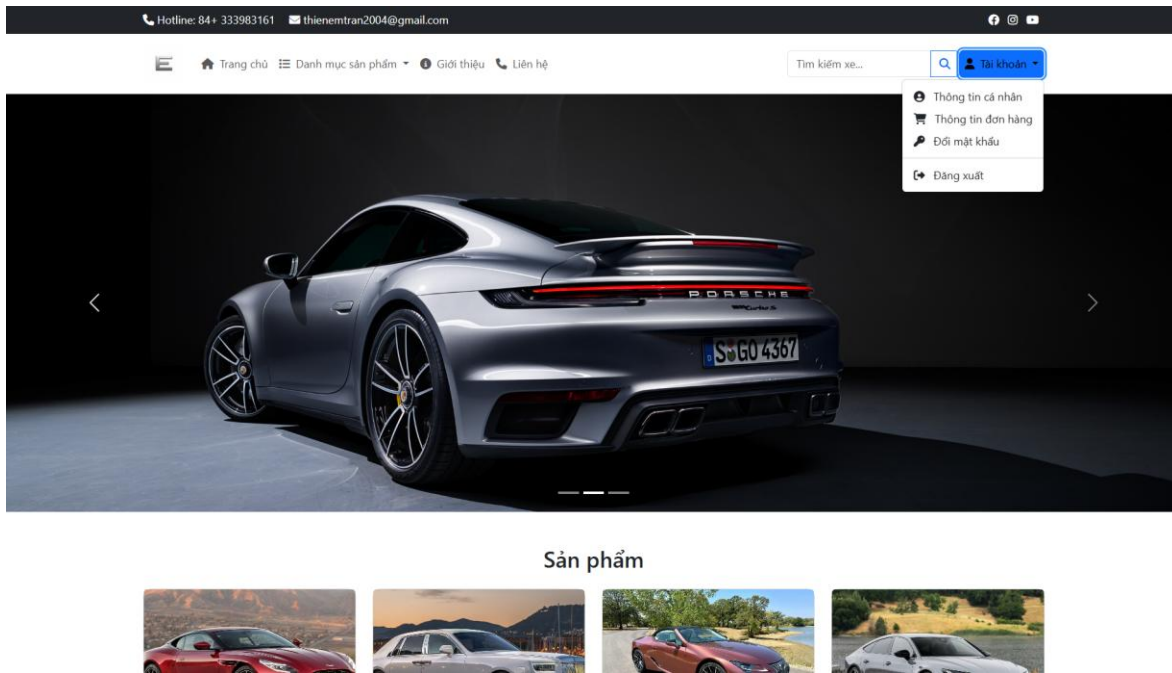


SQL Server là phần mềm ứng dụng quản lý cơ sở dữ liệu. Nó được dùng để lưu trữ, quản lý và truy vấn một cách nhanh chóng và tiện lợi. Vì nó là sản phẩm của Microsoft nên rất dễ dàng để liên kết với C# để truy vấn, thêm, xóa, sửa dữ liệu.

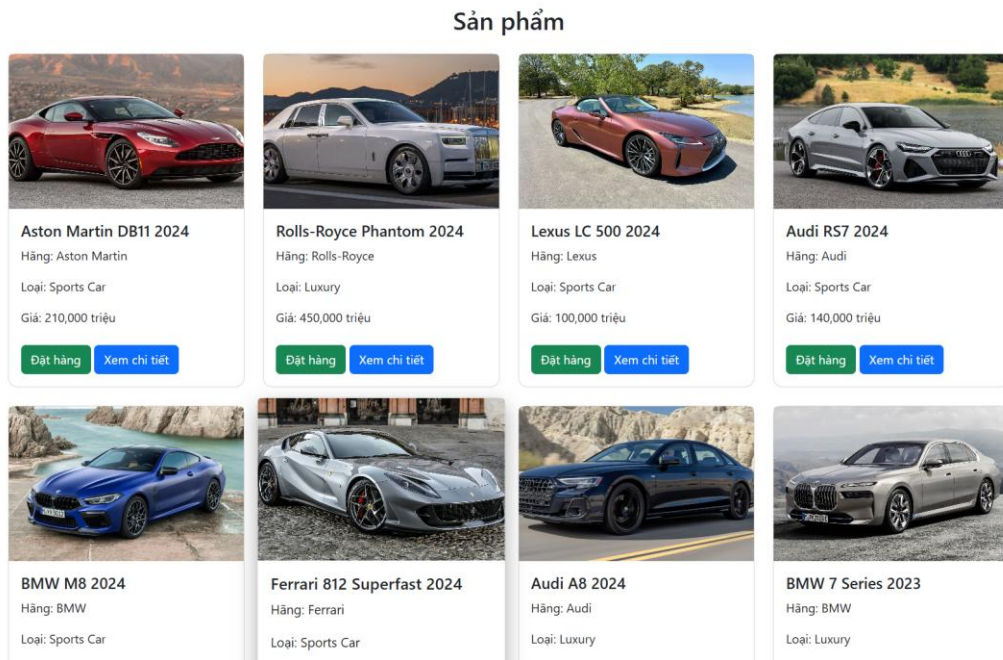
Trong chương trình này, chúng em có sử dụng một số thư viện như:

- Microsoft.AspNetCore.Identity.EntityFrameworkCore
- Microsoft.AspNetCore.Identity.UI
- Microsoft.EntityFrameworkCore.Design
- Microsoft.EntityFrameworkCore.SqlServer
- Microsoft.EntityFrameworkCore.Tools

## Chương 3 Kết quả chương trình



Hình 17: Trang chủ



Hình 18: Sản phẩm

## Liên hệ với chúng tôi

Chúng tôi luôn sẵn lòng lắng nghe và hỗ trợ bạn. Vui lòng liên hệ với chúng tôi qua các thông tin dưới đây hoặc điền vào biểu mẫu để được tư vấn nhanh chóng.

### Thông tin liên hệ

Địa chỉ: 123 Đường Thái Phiên, Phường 12, TP Đà Lạt, Lâm Đồng

Hotline: **84+ 333983161**

Email: **thienemtran2004@gmail.com**

Giờ làm việc: Thứ 2 - Thứ 7 (8:00 - 18:00)

### Theo dõi chúng tôi

Kết nối với chúng tôi qua các kênh mạng xã hội để cập nhật tin tức và ưu đãi mới nhất!



### Gửi tin nhắn cho chúng tôi

Họ và tên \*

Email \*

Tiêu đề

Nội dung tin nhắn \*



## Hình 19: Liên hệ

## Giới thiệu về E-Car

Chào mừng bạn đến với E-Car – nơi chúng tôi biến ước mơ sở hữu xế hộp của bạn thành hiện thực với sự tin cậy và chuyên nghiệp.

### Tầm nhìn của chúng tôi

E-Car ra đời với tầm nhìn trở thành đơn vị hàng đầu trong lĩnh vực mua bán xe ô tô đã qua sử dụng, mang đến những giá trị vượt trội cho khách hàng và đối tác. Chúng tôi không chỉ bán xe, mà còn xây dựng niềm tin và sự hài lòng dài lâu.

Chúng tôi cam kết cung cấp các dòng xe chất lượng cao, minh bạch về nguồn gốc và tình trạng, đồng thời mang đến trải nghiệm dịch vụ chuyên nghiệp, tận tâm từ khâu tư vấn đến hậu mãi.



### Giá trị cốt lõi

- Chất lượng:** Mỗi chiếc xe đều được kiểm tra kỹ lưỡng, đảm bảo tiêu chuẩn cao nhất.
- Uy tín:** Minh bạch trong mọi giao dịch, xây dựng lòng tin với khách hàng.
- Chuyên nghiệp:** Đội ngũ nhân viên tận tâm, am hiểu, mang đến dịch vụ hoàn hảo.
- Tận tâm:** Luôn đặt khách hàng làm trung tâm, lắng nghe và đáp ứng mọi nhu cầu.

## Hình 20: Giới thiệu

## Đăng nhập

Email

thienemtran2004@gmail.com

Mật khẩu

.....

☐ Ghi nhớ đăng nhập

→ Đăng Nhập

Quên mật khẩu? [Lấy lại mật khẩu](#)

Chưa có tài khoản? [Đăng ký ngay](#)

Hình 21: Đăng nhập

Hotline: 84+ 333983161 | thienemtran2004@gmail.com

Trang chủ

Danh mục sản phẩm

Giới thiệu


Liên hệ

Tìm kiếm xe...


Tài khoản

### So sánh xe

Trở về trang sản phẩm



**Aston Martin DB11 2024**  
Hãng: Aston Martin  
Loại: Sports Car  
Năm: 2024 Mới nhất  
Giá: 210,000 triệu Cao nhất  
Km đã đi: 600 Km Ít nhất  
Nhiên liệu: Xăng  
Hộp số: Tự động  
Màu: Đỏ rượu vang  
Vị trí: Hà Nội



**Maserati Ghibli 2023**  
Hãng: Maserati  
Loại: Luxury  
Năm: 2023  
Giá: 90,000 triệu  
Km đã đi: 1,500 Km  
Nhiên liệu: Xăng  
Hộp số: Tự động  
Màu: Xám  
Vị trí: TP.HCM

Hình 22: So sánh sản phẩm

## Đăng ký tài khoản

Họ và tên

Email

Số điện thoại

Địa chỉ

Mật khẩu

Nhập lại mật khẩu

 Đăng ký

Đã có tài khoản? [Đăng nhập](#)


**Hình 23: Đăng ký**

Hotline: 84+ 333983161
thienemtran2004@gmail.com

Trang chủ
Danh mục sản phẩm
Giới thiệu
Liên hệ

Tìm kiếm xe...
Tài khoản

### Đặt mua xe: Aston Martin DB11 2024



**Aston Martin DB11 2024**

**Hãng:** Aston Martin

**Loại:** Sports Car

**Giá:** 210,000 VNĐ

**Mô tả:** Aston Martin DB11 xe thể thao hạng sang, đẳng cấp và hiệu suất cao.

#### Thông tin khách hàng

Họ và tên

Số điện thoại

Email

Địa chỉ

Ghi chú

Xác nhận đặt hàng


Hình 24: Đặt hàng

Trang chủ
Danh mục sản phẩm
Giới thiệu
Liên hệ

Tìm kiếm xe...
Tài khoản

### Chi tiết đơn hàng #000005

#### Thông tin xe



**Aston Martin DB11 2024**

**Hãng:** Aston Martin

**Loại:** Sports Car

**Giá:** 210,000 VNĐ

**Mô tả:** Aston Martin DB11 xe thể thao hạng sang, đẳng cấp và hiệu suất cao.

#### Thông tin khách hàng

**Họ tên:** Trần Thiện Em

**Số điện thoại:** 0924325653

**Email:** thienemtran2004@gmail.com

**Địa chỉ:** An Hữu , Cái Bè, Tiền Giang

**Ngày đặt:** 25/06/2025 00:19

**Trạng thái:** Đang chờ xử lý

Xác nhận đơn hàng

Về trang chủ
Danh sách đơn hàng

Hình 25: Chi tiết đơn hàng



Tìm theo email, SĐT, tên...

### Hình 26: Thông tin đơn hàng

Bộ lọc tìm kiếm

Từ khóa:

Hãng xe:

Loại xe:

Tên xe, hãng xe, mô tả...

-- Chọn hãng xe --

-- Chọn loại xe --

Năm từ:

Năm đến:

Giá từ (triệu):

Giá đến (triệu):

Loại nhiên liệu:

Hộp số:

Địa điểm:

-- Chọn nhiên liệu --

-- Chọn hộp số --

Tỉnh/Thành phố

🔍 Tìm kiếm

🗑️ Xóa bộ lọc

## Hình 27: Tìm kiếm

## **Kết luận**

### **1. Kết quả đạt được**

Hiện nay, thị trường đang phát triển nhiều trang website thương mại điện tử kinh doanh mặt hàng oto , và việc nắm vững tổng quan về chúng sẽ giúp chúng ta xây dựng dự án của mình một cách chi tiết hơn.

Qua môn học này, giúp chúng em hiểu biết hơn về Web MVC, cách nó hoạt động và cách để tạo và sử dụng nó cho trang web của mình.

Mặc dù đã cố gắng rất nhiều, tìm hiểu những kiến thức mới và tham khảo các website quản lý khóa học. Nhưng do còn hạn chế về mặt thời gian cũng như kinh nghiệm thực tiễn nên em không thể tránh khỏi những thiếu sót nhất định. Chúng em mong nhận được những lời góp ý từ thầy để ngày càng hoàn thiện sản phẩm hơn.

Và hiện tại đồ án của chúng em chỉ dừng lại ở việc xây dựng Models-Views-Controller, chúng em mong muốn trong tương lai sẽ phát triển thêm phần giao diện và chức năng của đồ án này.

### **2. Hướng nghiên cứu**

Với những hạn chế nêu trên, hướng nghiên cứu của em sau này là: Xây dựng giao diện và chức năng, tìm hiểu sâu hơn về WEB MVC và các framework hay để phát triển đồ án của mình.



## Tài liệu tham khảo

### **Tài liệu chính:**

#### **1. ASP.NET Core in Action**

Tác giả: Andrew Lock

Link online: <https://www.manning.com/books/asp-net-core-in-action-second-edition>

Link: <https://www.amazon.com/ASP-NET-Core-Action-Andrew-Lock/dp/1617298301>

#### **2. Pro ASP.NET Core MVC**

Tác giả: Adam Freeman

Link online: Thường có trên Apress.com hoặc các nền tảng sách điện tử khác.

Link: <https://www.amazon.com/Pro-ASP-NET-Core-MVC-9th/dp/1484288768/>

#### **3. Architecting Modern Web Applications with ASP.NET Core and Azure (eBook chính thức của Microsoft)**

Tác giả: Dino Esposito, Andrea Saltarello, và các cộng tác viên

Link online: <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/>

Link: ( Đây là eBook miễn phí trên trang của Microsoft, không bán trên Amazon như sách vật lý truyền thống )

#### **4. Learning ASP.NET Core MVC Programming**

Tác giả: Juergen Gutsch

Link online: Thường có trên Packtpub.com hoặc các nền tảng sách điện tử khác.

Link: <https://www.amazon.com/Learning-ASP-NET-Core-MVC-Programming/dp/183882772X>

## **Tài liệu tham khảo:**

### **1. Hướng dẫn cơ bản ASP.NET MVC**

iHocLapTrinh.com. Link online: <https://ihoclaptrinh.com/series/huong-dan-lap-trinh-asp-net-mvc-co-ban> (Truy cập: 25 tháng 6 năm 2025).

### **2. Get started with ASP.NET Core MVC and Visual Studio, Microsoft Learn.**

Link online: <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-8.0&tabs=visual-studio>