

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC YERSIN ĐÀ LẠT**



**BÁO CÁO MÔN HỌC LẬP TRÌNH WEB**

**XÂY DỰNG WEB SITE THƯƠNG MẠI ĐIỆN TỬ KINH DOANH XE GẮN  
MÁY**

Môn học : Lập Trình Web

Giảng viên : Nguyễn Đức Tấn

Sinh viên thực hiện : Nguyễn Thị Hồng Vần

MSSV: 2301010073

Khóa học: 2023-2026

## **LỜI NÓI ĐẦU**

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép em được bày tỏ lòng biết ơn đến giảng viên và nhà trường đã tạo điều kiện hỗ trợ, giúp đỡ em trong suốt quá trình học tập và thực hiện đề tài này. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô và bạn bè.

Với lòng biết ơn sâu sắc nhất, em xin gửi đến Thầy Nguyễn Đức Tấn đã truyền đạt vốn kiến thức quý báu cho chúng em trong suốt quá trình học tập. Nhờ có những lời hướng dẫn, dạy bảo của các thầy cô nên đề tài nghiên cứu của em mới có thể hoàn thiện tốt đẹp.

Một lần nữa, em xin chân thành cảm ơn thầy người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn em hoàn thành tốt bài báo cáo này trong thời gian qua.

Bài báo cáo thực hiện trong khoảng thời gian dài. Bước đầu đi vào thực tế của em còn hạn chế và còn nhiều bất ngờ nên không tránh khỏi những thiếu sót, em rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy Cô để kiến thức của em trong lĩnh vực này được hoàn thiện hơn đồng thời có điều kiện bổ sung, nâng cao ý thức của mình.

Em xin chân thành cảm ơn!

## **MỤC LỤC**

<b>LỜI MỞ ĐẦU .....</b>	<b>.....</b>
-------------------------	--------------

## **CHƯƠNG 1: CƠ SỞ LÝ THUYẾT .....**

<b>1.1.1 Phân biệt giữa Website và Ứng dụng Website.....</b>	<b>1</b>
<b>1.1.2 Thành phần cấu tạo nên một Website.....</b>	<b>2</b>
<b>1.1. 3 Phân loại Web .....</b>	<b>3</b>
<b>1.1.4 Hoạt động của Ứng dụng Website.....</b>	<b>4</b>
<b>1.2.1 Giới thiệu mô hình MVC .....</b>	<b>5</b>
<b>1.2.2 Ưu điểm và hạn chế của MVC .....</b>	<b>6</b>
<b>1.2.3 Giới thiệu về ASP.NET Core .....</b>	<b>7</b>
<b>1.2.4 Nguyên lý hoạt động của MVC .....</b>	<b>8</b>

## **CHƯƠNG 2: XÂY DỰNG ỨNG DỤNG .....**

<b>2.1. Phát biểu bài toán .....</b>	<b>9</b>
<b>2.2. Phân tích yêu cầu .....</b>	<b>10</b>
<b>2.3. Phân tích dữ liệu .....</b>	<b>11</b>
<b>2.4. Thiết kế giao diện .....</b>	<b>12</b>
<b>2.5. Thiết kế hệ thống theo mô hình MVC .....</b>	<b>13</b>
<b>2.6. Triển khai và cài đặt .....</b>	<b>14</b>

## **CHƯƠNG 3: KẾT QUẢ CHƯƠNG TRÌNH .....**

<b>3.1. Demo chức năng chính .....</b>	<b>15</b>
<b>3.2. Hình ảnh minh họa .....</b>	<b>16</b>
<b>3.3. Nhận xét và đánh giá .....</b>	<b>17</b>

<b>KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>18</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>19</b>
<b>PHỤ LỤC .....</b>	<b>20</b>

## **CHƯƠNG 1 CƠ SỞ LÝ THUYẾT**

### **1.1.1. Phân biệt giữa Website và Ứng dụng Website**

Website là một hệ thống các trang thông tin được xây dựng và lưu trữ trên máy chủ web, cho phép người dùng truy cập thông qua trình duyệt bằng địa chỉ URL. Nội dung trên website thường được tổ chức theo dạng văn bản, hình ảnh, âm thanh, hoặc các phương tiện truyền thông khác, phục vụ mục đích cung cấp thông tin cho người truy cập.

Ứng dụng website là một dạng phần mềm chạy trên nền tảng web, cho phép người dùng tương tác trực tiếp và thực hiện các tác vụ cụ thể thông qua giao diện trình duyệt. Ứng dụng website không chỉ hiển thị thông tin mà còn xử lý dữ liệu, xác thực người dùng, thực thi nghiệp vụ và phản hồi kết quả tương ứng theo thời gian thực. Việc vận hành của ứng dụng website thường phụ thuộc vào cả phía trình duyệt và máy chủ.

### **1.1.2. Thành phần cấu tạo nên một Website**

Một website được hình thành từ nhiều thành phần hợp thành với chức năng và vai trò riêng biệt:

- **Giao diện người dùng:** Là phần trực quan hiển thị trên trình duyệt mà người dùng tương tác trực tiếp. Được xây dựng chủ yếu bằng các ngôn ngữ đánh dấu và lập trình phía trình duyệt như HTML, CSS và JavaScript.
- **Máy chủ web:** Là nơi lưu trữ toàn bộ mã nguồn và tài nguyên của website. Máy chủ chịu trách nhiệm tiếp nhận và xử lý các yêu cầu từ trình duyệt người dùng.
- **Cơ sở dữ liệu:** Là nơi lưu trữ và quản lý dữ liệu phát sinh trong quá trình vận hành website, bao gồm thông tin người dùng, nội dung bài viết, cấu hình hệ thống và các dữ liệu nghiệp vụ.
- **Xử lý phía máy chủ:** Bao gồm các đoạn mã chương trình chạy trên máy chủ để thực hiện các thao tác xử lý logic, truy xuất cơ sở dữ liệu, điều hướng luồng thông tin và phản hồi kết quả về cho người dùng.

- **Tên miền và hệ thống phân giải tên miền:** Là địa chỉ định danh website trên môi trường Internet, cho phép người dùng truy cập dễ dàng thay vì phải nhớ địa chỉ IP của máy chủ.

### 1.1.3. Phân loại Website

Website có thể được phân loại dựa trên nhiều tiêu chí khác nhau tùy vào cấu trúc, chức năng, phạm vi sử dụng hoặc mục đích triển khai.

- Theo mức độ tương tác: bao gồm website tĩnh và website động. Website tĩnh có nội dung cố định trong khi website động có khả năng hiển thị nội dung được xử lý từ cơ sở dữ liệu hoặc từ yêu cầu người dùng.
- Theo mục đích sử dụng: bao gồm website thương mại, giáo dục, giải trí, tin tức, cá nhân, và tổ chức.
- Theo phạm vi truy cập: bao gồm các website công cộng (hoạt động trên Internet) và các website nội bộ (hoạt động trong mạng nội bộ hoặc Intranet).

Việc phân loại website giúp định hướng đúng trong quá trình thiết kế, phát triển và triển khai hệ thống phù hợp với nhu cầu thực tế.

### 1.1.4. Hoạt động của Ứng dụng Website

Quy trình hoạt động của một ứng dụng website tuân theo mô hình client-server. Trình duyệt người dùng đóng vai trò là client, gửi yêu cầu đến máy chủ thông qua giao thức HTTP hoặc HTTPS. Máy chủ web tiếp nhận yêu cầu, chuyển đến thành phần xử lý logic để thực hiện các tác vụ nghiệp vụ, truy xuất dữ liệu nếu cần thiết, sau đó phản hồi kết quả trở lại cho trình duyệt.

Dữ liệu từ người dùng được xử lý phía máy chủ có thể trải qua các bước kiểm tra, xác thực, lưu trữ và chuyển đổi định dạng trước khi được trả về. Trình duyệt sau đó sẽ hiển thị kết quả dưới dạng HTML, JSON hoặc định dạng phù hợp với ngữ cảnh sử dụng.

Ứng dụng website hiện đại thường sử dụng các công nghệ hỗ trợ như AJAX, RESTful API hoặc WebSocket để nâng cao khả năng tương tác và tốc độ phản hồi, từ đó tối ưu trải nghiệm người dùng.

### 1.2.1. Giới thiệu mô hình MVC

Mô hình MVC (Model - View - Controller) là một mô hình kiến trúc phần mềm được sử dụng phổ biến trong việc phát triển các hệ thống ứng dụng có giao diện người dùng. Mục tiêu của mô hình là phân tách rõ ràng ba thành phần chính của hệ thống: dữ liệu, giao

diện và xử lý logic, nhằm nâng cao tính tổ chức, khả năng bảo trì và mở rộng của phần mềm.

- **Model** là thành phần chịu trách nhiệm quản lý dữ liệu và các quy tắc nghiệp vụ liên quan đến dữ liệu đó. Model có thể bao gồm các lớp tương tác trực tiếp với cơ sở dữ liệu hoặc thực hiện các phép biến đổi dữ liệu.
- **View** là lớp hiển thị, đảm nhận nhiệm vụ trình bày dữ liệu từ Model đến người dùng cuối. View không chứa logic xử lý nghiệp vụ mà chỉ tập trung vào giao diện và cách trình bày thông tin.
- **Controller** là thành phần trung gian giữa Model và View. Nó tiếp nhận yêu cầu từ người dùng thông qua giao diện, xử lý các logic cần thiết và cập nhật dữ liệu cho Model hoặc View tùy theo ngữ cảnh.

Việc áp dụng mô hình MVC cho phép phát triển phần mềm theo hướng phân lớp chức năng, giúp dễ dàng quản lý mã nguồn, hỗ trợ kiểm thử và cải tiến hệ thống trong tương lai.

### 1.2.2. Ưu điểm và hạn chế của mô hình MVC

Mô hình MVC mang lại một số ưu điểm đáng kể trong phát triển phần mềm:

- Cho phép tách biệt rõ ràng giữa giao diện người dùng và logic xử lý, giúp lập trình viên có thể phát triển độc lập từng thành phần.
- Hỗ trợ tổ chức mã nguồn theo hướng mô-đun, tăng tính tái sử dụng và khả năng bảo trì.
- Thích hợp với các dự án có quy mô lớn hoặc cần mở rộng, nhờ cấu trúc rõ ràng và linh hoạt.
- Tạo điều kiện thuận lợi cho việc kiểm thử từng lớp chức năng riêng biệt, đặc biệt là lớp xử lý nghiệp vụ và dữ liệu.

Tuy nhiên, mô hình MVC cũng tồn tại một số hạn chế:

- Tăng độ phức tạp trong giai đoạn thiết kế và triển khai, đặc biệt đối với những dự án nhỏ hoặc yêu cầu đơn giản.
- Việc phối hợp giữa ba thành phần Model, View và Controller đòi hỏi lập trình viên phải có kiến thức nền tảng vững và tuân thủ chặt chẽ nguyên tắc kiến trúc.

### 1.2.3. Nguyên lý hoạt động của mô hình MVC

Mô hình MVC vận hành theo nguyên tắc phân chia ba thành phần độc lập, trong đó mỗi thành phần đảm nhiệm một vai trò riêng biệt trong quá trình xử lý yêu cầu từ người dùng đến khi hiển thị kết quả.

Khi người dùng tương tác với ứng dụng (chẳng hạn gửi yêu cầu qua trình duyệt), yêu cầu đó sẽ được tiếp nhận và xử lý theo các bước sau:

**1. Controller tiếp nhận yêu cầu:**

Controller là thành phần đầu tiên tiếp nhận các yêu cầu từ giao diện người dùng. Dựa trên nội dung của yêu cầu (thường là URL hoặc dữ liệu gửi qua form), Controller xác định logic cần thực hiện.

**2. Xử lý nghiệp vụ và tương tác với Model:**

Controller thực hiện các tác vụ xử lý nghiệp vụ cần thiết. Nếu yêu cầu liên quan đến dữ liệu, Controller sẽ gọi các phương thức của Model để truy xuất, cập nhật hoặc xử lý dữ liệu từ cơ sở dữ liệu.

**3. Cập nhật View và trả kết quả:**

Sau khi nhận kết quả từ Model, Controller sẽ truyền dữ liệu đến View tương ứng. View có nhiệm vụ hiển thị dữ liệu dưới dạng giao diện HTML hoặc một định dạng phù hợp, và kết quả được trả về cho người dùng thông qua trình duyệt.

**4. Hiển thị kết quả cho người dùng:**

Người dùng nhận được trang kết quả được sinh ra từ View, có thể tiếp tục tương tác và gửi các yêu cầu mới.

### 1.2.4. Giới thiệu về ASP.NET Core

ASP.NET Core là nền tảng phát triển ứng dụng web mã nguồn mở do Microsoft phát triển, là bước cải tiến toàn diện so với ASP.NET Framework truyền thống. ASP.NET Core được xây dựng lại hoàn toàn nhằm mục tiêu đa nền tảng, hiệu năng cao, cấu trúc nhẹ và dễ bảo trì.

Nền tảng này hỗ trợ xây dựng các loại ứng dụng hiện đại như:

- Ứng dụng web động (Web App),
- Dịch vụ RESTful API,
- Ứng dụng thời gian thực (Real-time apps với SignalR),
- Và ứng dụng đa nền tảng (cross-platform) triển khai được trên Windows, Linux và macOS.

Một số đặc điểm nổi bật của ASP.NET Core bao gồm:

- **Kiến trúc modular:** Tách thành từng gói nhỏ, chỉ tải những phần cần thiết, giúp tối ưu hiệu năng và giảm kích thước ứng dụng.
- **Tích hợp Dependency Injection (DI):** Hỗ trợ mặc định mô hình tiêm phụ thuộc, giúp tăng khả năng mở rộng và kiểm thử.
- **Middleware Pipeline:** Cho phép kiểm soát và xử lý các request thông qua chuỗi middleware linh hoạt.
- **Hệ thống định tuyến mạnh mẽ:** Hỗ trợ route truyền thống và route attribute-based.
- **Hỗ trợ Razor Pages và MVC cùng lúc:** Cho phép lựa chọn kiến trúc phù hợp tùy vào đặc thù dự án.

Ngoài ra, ASP.NET Core còn tích hợp chặt chẽ với Entity Framework Core – một ORM hiện đại, giúp truy vấn dữ liệu theo mô hình hướng đối tượng, giảm thiểu thao tác SQL thuần và nâng cao hiệu quả lập trình.

Nhờ khả năng mở rộng linh hoạt, độ ổn định cao và cộng đồng hỗ trợ mạnh, ASP.NET Core hiện đang là lựa chọn phổ biến trong các dự án xây dựng hệ thống web quy mô vừa và lớn, đặc biệt trong lĩnh vực thương mại điện tử, quản lý nội bộ và dịch vụ công trực tuyến.

## CHƯƠNG 2



## 2.2. Phân tích yêu cầu

### 2.2.1. Yêu cầu chức năng

Hệ thống website kinh doanh xe gắn máy cần đáp ứng các chức năng nghiệp vụ cơ bản như sau:

- **Quản lý sản phẩm:** Hệ thống cho phép quản trị viên thực hiện các thao tác thêm mới, chỉnh sửa, xóa và tra cứu thông tin sản phẩm xe gắn máy. Thông tin sản phẩm bao gồm mã xe, tên xe, hãng sản xuất, phân khối, đơn giá, mô tả và hình ảnh minh họa.
- **Quản lý danh mục và hãng xe:** Hệ thống hỗ trợ tổ chức sản phẩm theo danh mục và hãng sản xuất nhằm phục vụ việc phân loại, hiển thị và tìm kiếm hiệu quả.
- **Quản lý khách hàng:** Cho phép lưu trữ thông tin khách hàng đặt hàng hoặc gửi yêu cầu liên hệ. Thông tin bao gồm họ tên, số điện thoại, địa chỉ, email và nội dung yêu cầu.
- **Quản lý đơn hàng:** Ghi nhận các đơn đặt hàng từ phía người dùng, theo dõi trạng thái xử lý và thông tin sản phẩm được đặt mua. Mỗi đơn hàng liên kết với một khách hàng cụ thể và danh sách sản phẩm tương ứng.
- **Tìm kiếm và lọc dữ liệu:** Cung cấp chức năng tìm kiếm xe theo tên, hãng sản xuất hoặc mức giá. Ngoài ra, hỗ trợ lọc theo danh mục hoặc thông số kỹ thuật nhằm nâng cao trải nghiệm người dùng.
- **Giao diện người dùng:** Cung cấp giao diện trình bày thông tin sản phẩm, danh sách xe, chi tiết sản phẩm, biểu mẫu đặt hàng và thông báo kết quả đặt hàng sau khi gửi.

### 2.2.2. Yêu cầu phi chức năng

Ngoài các chức năng chính, hệ thống cần đảm bảo các yêu cầu phi chức năng sau:

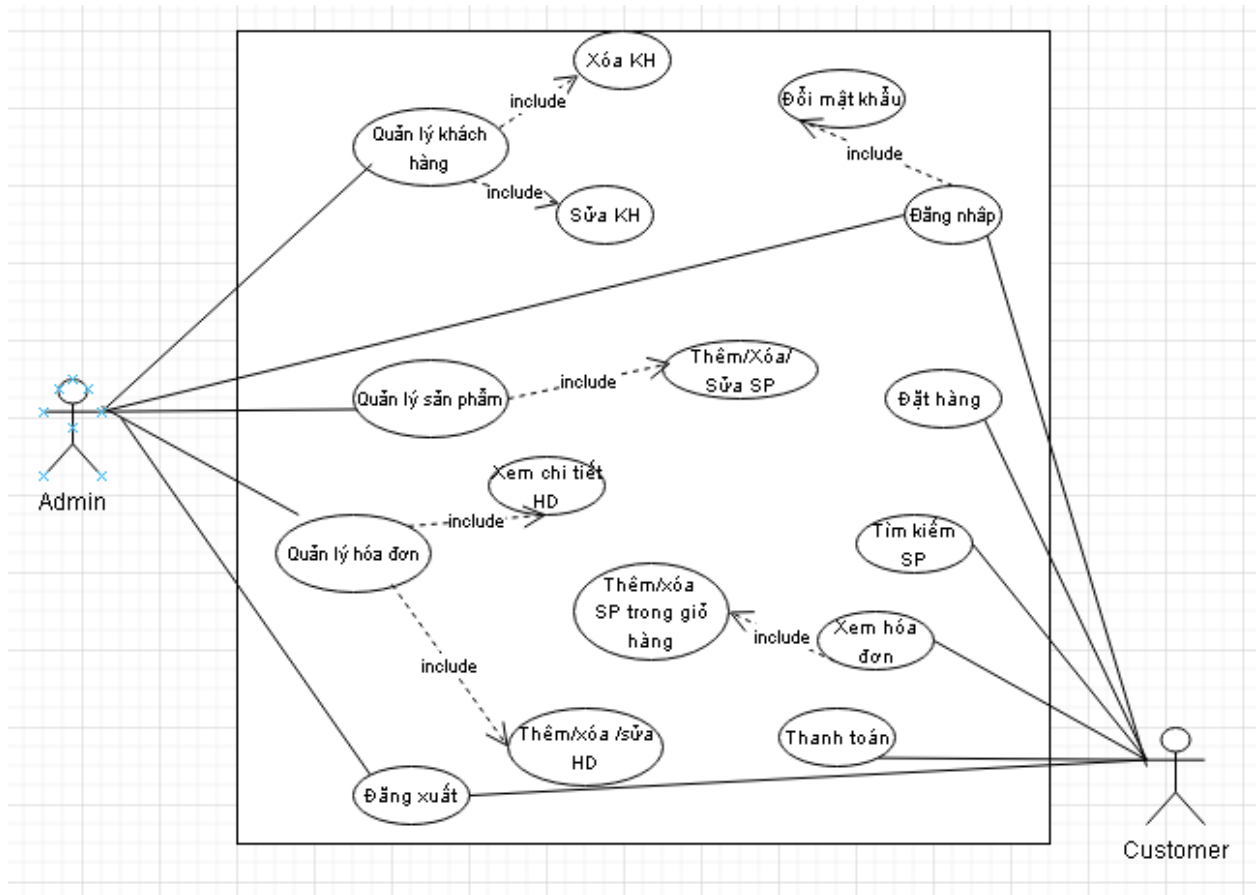
- **Hiệu năng:** Thời gian phản hồi của hệ thống đối với các thao tác thông thường (xem danh sách sản phẩm, gửi đơn hàng, tìm kiếm) không vượt quá 3 giây trong điều kiện mạng ổn định.
- **Bảo mật:** Chức năng quản trị chỉ được truy cập sau khi thực hiện xác thực người dùng. Hệ thống phải đảm bảo dữ liệu nhạy cảm được bảo vệ khỏi truy cập trái phép và không rò rỉ thông tin khách hàng.

- **Khả năng mở rộng:** Hệ thống được thiết kế theo hướng mở, cho phép dễ dàng bổ sung thêm các danh mục sản phẩm, hãng xe mới hoặc tích hợp các chức năng nâng cao như thanh toán trực tuyến, đánh giá sản phẩm, phân quyền người dùng.
- **Tính khả dụng:** Hệ thống phải hoạt động ổn định và sẵn sàng phục vụ người dùng liên tục, trừ những khoảng thời gian bảo trì định kỳ có thông báo trước.
- **Tính tương thích:** Website đảm bảo khả năng hiển thị đồng đều và chính xác trên các trình duyệt phổ biến như Google Chrome, Mozilla Firefox, Microsoft Edge cũng như trên các thiết bị có độ phân giải màn hình khác nhau, bao gồm cả thiết bị di động.
- **Tính thân thiện:** Giao diện người dùng phải đảm bảo rõ ràng, dễ thao tác, trực quan và phù hợp với thói quen sử dụng của người dùng phổ thông, không yêu cầu kỹ năng kỹ thuật chuyên sâu.

### 2.2.3. Sơ đồ Use Case

Hệ thống có hai tác nhân chính, bao gồm:

- **Khách hàng:** Là người truy cập website, có thể thực hiện các hành vi như xem thông tin sản phẩm, tìm kiếm xe theo tiêu chí cụ thể, gửi yêu cầu liên hệ và thực hiện đặt hàng.
- **Quản trị viên:** Là người quản lý nội dung website, thực hiện các chức năng thêm, sửa, xóa sản phẩm; quản lý danh mục, hãng xe, thông tin khách hàng và đơn hàng. Tác nhân này phải đăng nhập hệ thống để được cấp quyền thao tác.



## 2.3. Phân tích dữ liệu

### 2.3.1. Lược đồ thực thể

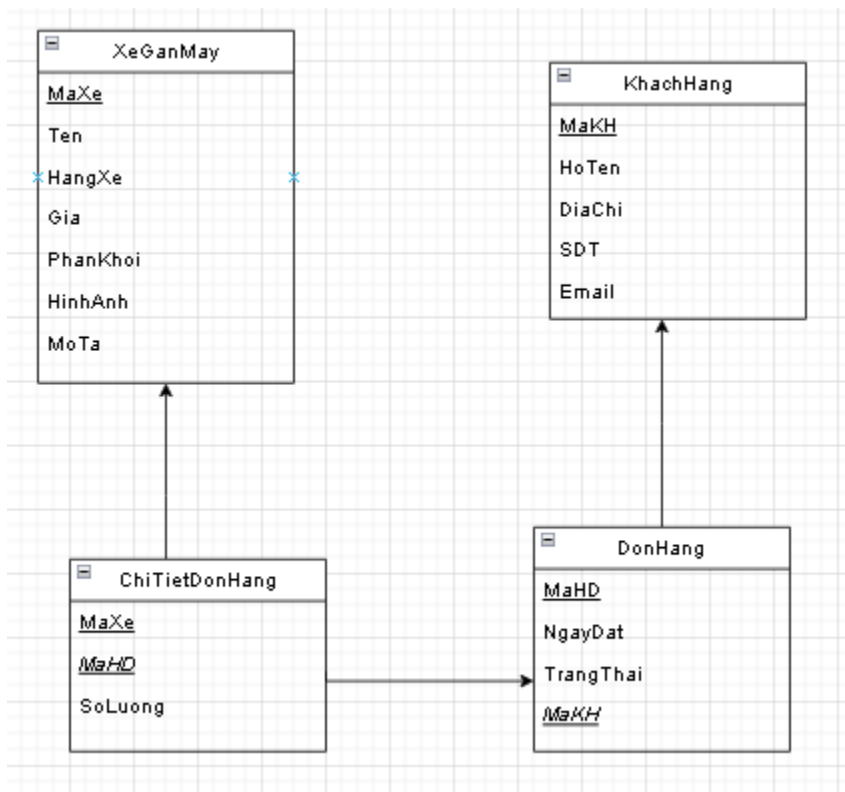
Cơ sở dữ liệu của hệ thống được thiết kế nhằm phục vụ các chức năng quản lý sản phẩm, khách hàng và đơn hàng. Các thực thể chính bao gồm:

- **XeMay**(MaXe, TenXe, Hang, Gia, PhanKhoi, MoTa, HinhAnh, MaDM)
- **KhachHang**(MaKH, HoTen, Email, DienThoai, DiaChi)
- **DonHang**(MaDH, MaKH, NgayDat, TrangThai)
- **ChiTietDonHang**(MaDH, MaXe, SoLuong)

### 2.3.2. Mô hình quan hệ dữ liệu

- Mỗi **KhachHang** có thể thực hiện nhiều **DonHang** → quan hệ 1-N.
- Mỗi **DonHang** gồm nhiều **ChiTietDonHang** → quan hệ 1-N.
- Mỗi **ChiTietDonHang** liên kết với một sản phẩm **XeMay** cụ thể → quan hệ N-1.

### 2.3.3. Sơ đồ thực thể - quan hệ (ERD)



### 2.4. Thiết kế giao diện

Giao diện người dùng của hệ thống được thiết kế theo nguyên tắc trực quan, dễ sử dụng, tuân thủ các tiêu chuẩn về trải nghiệm người dùng và tính tương thích trình duyệt. Các trang chức năng chính bao gồm:

- **Trang danh sách sản phẩm:** Hiển thị danh sách xe gắn máy dưới dạng lưới hoặc danh sách, có hỗ trợ phân trang và lọc theo hãng, phân khối hoặc danh mục.
- **Trang chi tiết sản phẩm:** Hiển thị thông tin chi tiết về một xe cụ thể, bao gồm hình ảnh, mô tả kỹ thuật, giá bán và nút đặt hàng.
- **Trang thêm/sửa sản phẩm (dành cho quản trị viên):** Cho phép thêm mới hoặc cập nhật thông tin sản phẩm, với các trường kiểm tra hợp lệ đầu vào.
- **Trang quản lý khách hàng:** Hiển thị danh sách khách hàng đã gửi đơn hàng hoặc yêu cầu liên hệ.
- **Trang quản lý đơn hàng:** Theo dõi các đơn hàng, trạng thái xử lý và chi tiết từng đơn.

## 2.5. Thiết kế hệ thống theo mô hình MVC

Hệ thống được thiết kế và xây dựng dựa trên mô hình kiến trúc **MVC (Model – View – Controller)**, trong đó:

- **Model:** Gồm các lớp như XeMay.cs, DanhMuc.cs, KhachHang.cs, DonHang.cs, phản ánh các bảng dữ liệu và chứa các thuộc tính tương ứng với cột trong cơ sở dữ liệu. Model có thể tích hợp các annotation để kiểm tra dữ liệu đầu vào.
- **View:** Sử dụng Razor Pages để xây dựng các giao diện tương tác, bao gồm các trang Index (danh sách), Create (thêm mới), Edit (chỉnh sửa), Details (chi tiết). Giao diện được phát triển bằng HTML kết hợp với Bootstrap và Razor syntax để tạo cấu trúc động.
- **Controller:** Gồm các lớp điều khiển như XeMayController, KhachHangController, chịu trách nhiệm tiếp nhận yêu cầu từ giao diện, xử lý logic nghiệp vụ, gọi các phương thức tương tác với cơ sở dữ liệu và trả kết quả cho người dùng. Mỗi controller tương ứng với một nhóm chức năng.

## 2.6. Triển khai và cài đặt

### 2.6.1. Môi trường phát triển

- **Hệ điều hành:** Windows 10 hoặc cao hơn
- **IDE:** Visual Studio 2022
- **Cơ sở dữ liệu:** SQL Server
- **Framework:** .NET 6 (ASP.NET Core MVC)
- **ORM:** Entity Framework Core

### 2.6.2. Thư viện và công cụ sử dụng

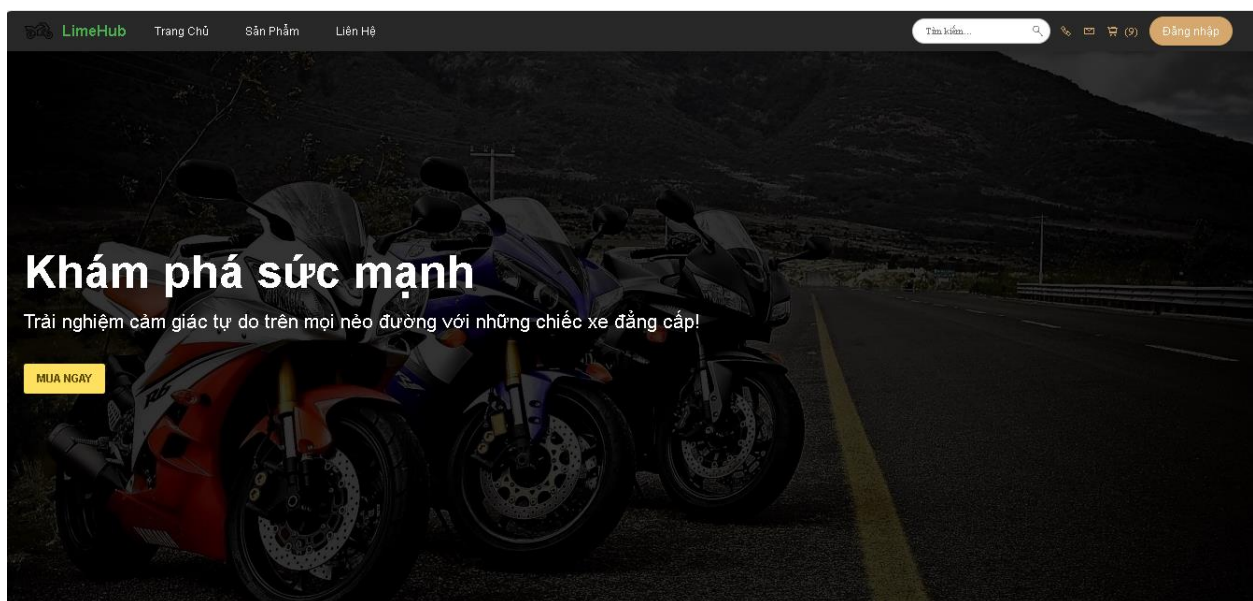
- **Entity Framework Core:** Quản lý dữ liệu và tương tác với cơ sở dữ liệu thông qua mô hình ORM.
- **Bootstrap:** Tạo giao diện người dùng hiện đại, hỗ trợ responsive trên thiết bị di động.
- **Data Annotation:** Kiểm tra tính hợp lệ của dữ liệu đầu vào ở tầng Model.


### 2.6.3. Các bước triển khai hệ thống

1. **Khởi tạo dự án:** Tạo mới project ASP.NET Core MVC trên Visual Studio.
2. **Cấu hình kết nối cơ sở dữ liệu:** Thiết lập chuỗi kết nối trong appsettings.json.




3. **Xây dựng các lớp Model:** Tạo các lớp thực thể phản ánh các bảng dữ liệu.
4. **Tạo migration và cập nhật cơ sở dữ liệu:** Sử dụng lệnh Add-Migration và Update-Database.
5. **Xây dựng Controller và View:** Tạo các chức năng thêm, sửa, xóa, xem chi tiết.
6. **Seed dữ liệu mẫu:** Tạo các bản ghi mẫu để kiểm thử tính năng.
7. **Kiểm thử và chạy ứng dụng:** Kiểm tra toàn bộ chức năng trên môi trường localhost và tinh chỉnh giao diện.

## CHƯƠNG 3





[Trang Chủ](#)
[Sản Phẩm](#)
[Liên Hệ](#)

Tìm kiếm...




[Đăng nhập](#)

## Xe Moto nổi bật


Những mẫu xe được yêu thích nhất



**LimeHub Offer!**

Giảm giá lên tới 30% cho các sản phẩm chọn lọc. Đừng bỏ lỡ!


[Xem ngay](#)



**Featured Item!**

Khám phá những mẫu xe và phụ kiện mới nhất.

[Tìm hiểu thêm](#)



**Về chúng tôi**


- Câu chuyện của chúng tôi
- Liên hệ
- Cửa hàng
- Đánh giá

**Cửa hàng**




- Xe Moto
- Phụ kiện
- Bảo dưỡng
- Ưu đãi


**Liên hệ**

- Điện thoại: +1 (0) 123 456 7890
- Email: info@limehub.com
- Địa chỉ: 3200 Yonge Street


[Trang Chủ](#)
[Sản Phẩm](#)
[Liên Hệ](#)


Tìm kiếm...




[Đăng nhập](#)



# Our Motorbikes


Ride your dream, every time...



**BMW R 1250 GS**

\$35,000.00


[Thêm vào giỏ hàng](#)



**Ducati Monster**

\$30,000.00

[Thêm vào giỏ hàng](#)



**Harley-Davidson Iron 883**

\$25,000.00

[Thêm vào giỏ hàng](#)



## BMW R 1250 GS

\$35,000.00

ID:5



### Mô tả chi tiết

Mô tả sản phẩm Xe adventure touring hàng đầu, phù hợp mọi địa hình.

Năm sản xuất

Nhiên liệu / Hộp số

Động cơ / phân khối

Màu sắc

Khu vực

Xem xe trực tiếp Liên hệ người bán

Thêm vào giỏ hàng

### Your cart is empty

Before proceed to checkout you must add some product to your shopping cart. You will find a lot of interesting product on our "Shop" page.

Start Shopping

### 3.3. Nhận xét và đánh giá kết quả

Hệ thống website kinh doanh xe gắn máy đã được xây dựng và triển khai thành công theo đúng định hướng ban đầu. Các chức năng chính như quản lý sản phẩm, danh mục xe,



khách hàng, đơn hàng và đặt hàng trực tuyến đều đã được hiện thực hóa. Giao diện người dùng đơn giản, dễ thao tác, hỗ trợ tìm kiếm và lọc dữ liệu hiệu quả.

Tuy nhiên, hệ thống vẫn còn tồn tại một số hạn chế đáng lưu ý. Trong phiên bản hiện tại, người dùng có thể thực hiện đặt hàng và hoàn tất quá trình gửi thông tin mua xe mà không cần đăng ký hoặc đăng nhập. Điều này tiềm ẩn rủi ro về xác thực thông tin người dùng và khó khăn trong việc quản lý lịch sử đơn hàng theo từng khách hàng.

Việc không kiểm soát danh tính người dùng có thể dẫn đến:

- Thông tin đơn hàng không chính xác hoặc trùng lặp.
- Khó đối chiếu khi xử lý hậu mãi, bảo hành.
- Không có cơ chế phân biệt giữa người dùng và quản trị viên.

Nhìn chung, hệ thống đã đạt được mục tiêu cốt lõi về mặt chức năng và kỹ thuật, nhưng cần được nâng cấp ở các khía cạnh bảo mật, xác thực và trải nghiệm người dùng nhằm đảm bảo tính toàn vẹn và tin cậy của quá trình giao dịch trực tuyến.

## KẾT LUẬN

Trong khuôn khổ đề tài, đã tiến hành nghiên cứu, thiết kế và xây dựng thành công hệ thống website kinh doanh xe gắn máy dựa trên nền tảng ASP.NET Core MVC. Ứng dụng đáp ứng đầy đủ các chức năng cơ bản như quản lý sản phẩm, danh mục xe, khách hàng, đơn hàng cũng như các tính năng tìm kiếm, đặt hàng trực tuyến.

Thông qua việc áp dụng mô hình kiến trúc MVC kết hợp với Entity Framework Core, hệ thống đảm bảo tính tách biệt giữa giao diện, dữ liệu và xử lý nghiệp vụ. Điều này không những nâng cao tính dễ bảo trì mà còn tạo nền tảng vững chắc cho việc mở rộng hệ thống trong tương lai.

Giao diện người dùng được thiết kế thân thiện, hỗ trợ phân trang và hoạt động ổn định trên nhiều thiết bị. Quá trình kiểm thử cho thấy ứng dụng vận hành hiệu quả, đáp ứng đúng các yêu cầu chức năng đặt ra.

Tuy nhiên, trong phạm vi thời gian thực hiện hạn chế, hệ thống vẫn còn một số điểm có thể cải tiến:

- Chưa tích hợp phân quyền người dùng giữa quản trị viên và khách hàng.

- Chưa hỗ trợ thanh toán trực tuyến qua cổng ngân hàng hoặc ví điện tử.
- Thiếu chức năng gửi email xác nhận khi đặt hàng thành công.

### **Hướng phát triển trong tương lai:**

- Bổ sung chức năng đăng nhập, phân quyền và quản lý tài khoản người dùng.
- Tích hợp các hình thức thanh toán trực tuyến thông qua cổng thanh toán nội địa và quốc tế.
- Cải thiện giao diện người dùng theo hướng responsive và hỗ trợ đa ngôn ngữ.
- Triển khai hệ thống thực tế trên môi trường cloud (Azure, AWS) để phục vụ người dùng thật sự.

### **TÀI LIỆU THAM KHẢO**

[1] Microsoft.(2023). ASP.NET Core Documentation.

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

[2]Microsoft.(2023).EntityFrameworkCoreDocumentation.

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

[3] Nguyễn Đức Hoàng. (2021). Lập trình ASP.NET Core MVC. NXB Khoa học & Kỹ thuật.

[4] Okxe.vn - Sàn giao dịch xe máy. <https://www.okxe.vn>