TRƯỜNG ĐẠI HỌC YERSIN ĐÀ LẠT

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO MÔN HỌC**

**LẬP TRÌNH WEB 2**

WEB QUẢN LÍ CHO THUÊ XE

***Giáo Viên Hướng Dẫn*** : Nguyễn Đức Tấn

***Lớp*** : Công nghệ thông tin K19

***Sinh viên thực hiện*** : Nguyễn Minh Đức – 12201032

Trần Nhã Phương – 12201072

Võ Minh Trí – 12201064

***Đà Lạt, năm 2024***

**PHẦN NHẬN XÉT CỦA GIẢNG VIÊN**

**Điểm:**

Ngày … tháng … năm …

*Ký ghi rõ họ tên*

**BÁO CÁO TIẾN ĐỘ**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nội dung công việc** | **Phân công** | **T1** | **T2** | **T3** | **T4** | **T5** | **T6** | **T7** |
| Phân công công việc | Cả nhóm | **x** |  |  |  |  |  |  |
| Tìm hiểu lý thuyết và viết nội dung | Minh Trí | **x** | **x** |  |  |  |  |  |
| Tổng hợp nội dung và viết báo cáo | Minh Trí, Nhã Phương |  |  | **x** | **x** |  |  |  |
| Chuẩn bị nội dung slide thuyết trình | Phương |  |  |  |  | **x** | **x** |  |
| Thống nhất nội dung thuyết trình | Cả nhóm |  |  | **x** |  |  |  |  |
| Thuyết trình | Minh Đức |  |  |  |  |  |  | **x** |
| Tổng hợp và chỉnh sửa báo cáo | Trần Nhã Phương |  |  |  |  | **x** | **x** |  |
| Xây dựng dữ liệu database cho trang web | Trần Nhã Phương |  |  |  | **x** |  |  |  |
| Xây dựng code backend(API) ứng dụng Web | Minh Đức, Nhã Phương |  | **x** | **x** | **x** | **x** | **x** |  |
| Xây dựng và kết nối View | Minh Đức |  |  |  |  | **x** | **x** |  |
| Thu thập hình ảnh và kết quả ứng dụng Web | Minh Đức |  |  |  |  |  | **x** |  |
| Demo lab | Đức |  |  |  |  |  |  | **x** |

**NHÓM TRƯỞNG ĐÁNH GIÁ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu Chí** | **Hiếm khi (0 - 1)** | **Đôi khi (2 - 3)** | **Thường xuyên (4 - 5)** |
| Đóng góp ý kiến |  | 3 |  |
| Lắng nghe và tôn trọng ý kiến người khác |  |  | 4 |
| Bảo đảm nội dung và tiến độ công việc được giao (deadline) |  | 3 |  |
| Tham gia các buổi họp nhóm | 0 |  |  |
| Kỹ năng giao tiếp với các thành viên |  |  | 4 |
| Chia sẻ với nhóm về công việc đang thực hiện |  |  | 5 |
| **Tổng Điểm:** | 19 | | |

**LỜI NÓI ĐẦU**

Với tình hình cuộc sống đang dần phát triển của đất nước hiện nay và đây là thời đại của công nghệ 4.0 bắt đầu trở thành một xu hướng của toàn thế giới không chỉ ở Việt Nam. Trên thế giới công nghệ hiện đại đang ngày càng phát triển mạnh mẽ. Ở Việt Nam, công nghệ cũng đang dần dần phát triển không kém cạnh các nước trên thế giới. Và công nghệ thông tin nó không thể thiếu trong mọi lĩnh vực của công việc.

Và Internet đóng vai trò quan trọng không thể phủ nhận trong cuộc sống hàng ngày của chúng ta. Việc truy cập thông tin, giao tiếp và thậm chí là quản lý công việc đã trở nên thuận tiện hơn bao giờ hết, đặc biệt nhờ vào sự phát triển của các ứng dụng web.

Trong hoạt động sản xuất, kinh doanh, giờ đây, các dịch vụ cũng ngày càng phát triển. Đối với Việt Nam là một nước khá phát triển về du lịch, là nơi mà nhiều quốc gia chọn làm nơi du lịch, và một trong những thứ quan trọng để trải nghiệm chuyến đi là phương tiện đi lại.Vì vậy, nhóm em đã chọn đề tài Hệ Thống API Cho Dịch Vụ Thuê Xe để thực hiện.

Trang web sẽ giúp người dùng có thể quản lí các dịch vụ cho thuê xe, quản lí được khách hàng, tạo ra một dịch vụ khá cần thiết ở một đất nước phát triển về linh vực du lịch như nước ta.

Mục Lục

[Chương 1: Tìm hiểu về Web Service (Cơ sở lý thuyết) 1](#_Toc169248128)

[**I.Khái niệm cơ bản về Web Service:** 1](#_Toc169248129)

[**1. Web Service là gì?** 1](#_Toc169248130)

[**2. Các loại Web Service** 1](#_Toc169248131)

[**3. API (Application Programming Interface)** 2](#_Toc169248132)

[**II.**  **Các thành phần chính của Web API và Những điểm nổi bật của Web API** 2](#_Toc169248133)

[**1. Các thành phần của Web API** 2](#_Toc169248134)

[**2. Điểm nổi bật** 3](#_Toc169248135)

[**III. Nguyên lý hoạt động của API (Working Principles)** 3](#_Toc169248136)

[**1.Nguyên lý hoạt động** 3](#_Toc169248137)

[**2.** **Ưu và nhược điểm của Web API** 4](#_Toc169248138)

[**IV. Kiểu dữ liệu JSON, XML** 4](#_Toc169248139)

[**1. Định nghĩa JSON (JavaScript Object Notation)** 4](#_Toc169248140)

[**2. Định nghĩa XML (Extensible Markup Language)** 5](#_Toc169248141)

[**V. Triển khai Web API dựa trên công nghệ ASP.NET Core** 5](#_Toc169248142)

[**1. ASP.NET Core là gì?** 5](#_Toc169248143)

[**2. Cấu trúc của một Web API ASP.NET Core** 5](#_Toc169248144)

**[3. Triển khai một Web API ASP.NET Core](#_Toc169248145)** [5](#_Toc169248145)

**[4. Kết nối với SQL Server (khi cần)](#_Toc169248146)** [5](#_Toc169248146)

**[VI. Các vấn đề bảo mật (Security Considerations)](#_Toc169248147)** [5](#_Toc169248147)

**[1. Định nghĩa Authentication](#_Toc169248148)** [5](#_Toc169248148)

[**2. Định nghĩa Authorization** 6](#_Toc169248149)

[**VII. Mã trạng thái (Status Code)** 6](#_Toc169248150)

[**1. Định nghĩa mã trạng thái (Status Code)** 6](#_Toc169248151)

[**2. Một số mã trạng thái thường gặp** 6](#_Toc169248152)

[Chương 2: Xây dựng ứng dụng Web 6](#_Toc169248153)

[**1. Bài toán ứng dụng** 6](#_Toc169248154)

[**1.1 Phát biểu bài toán ứng dụng** 6](#_Toc169248155)

[**1.2 Xây dựng yêu cầu** 7](#_Toc169248156)

[**1.3 Xây dựng mô hình ER** 7](#_Toc169248157)

[**1.4 Sơ đồ Use Case** 8](#_Toc169248158)

[**1.5. Xây dựng cơ sở dữ liệu** 9](#_Toc169248159)

[Chương 3 Kết quả chương trình 10](#_Toc169248161)

[**1. Register, Login và Logout** 10](#_Toc169248162)

[**1.1 Register** 10](#_Toc169248163)

[**1.2 Login** 10](#_Toc169248164)

[**1.3 Logout** 11](#_Toc169248165)

[**2. Giao diện người dùng** 11](#_Toc169248166)

[**2.1 Trang danh mục xe** 11](#_Toc169248167)

[**2.2 Trang khách hàng** 13](#_Toc169248168)

[**2.3 Trang nhân viên** 15](#_Toc169248169)

[**2.4 Trang hợp đồng thuê xe** 17](#_Toc169248170)

[**2.5 Trang thanh toán** 18](#_Toc169248171)

[KẾT LUẬN 20](#_Toc169248172)

[DANH MỤC TỪ NGỮ VIẾT TẮT 21](#_Toc169248174)

[Nguồn Tham Khảo 21](#_Toc169248175)

# **Chương 1: Tìm hiểu về Web Service (Cơ sở lý thuyết)**

## **I.Khái niệm cơ bản về Web Service:**

### **1. Web Service là gì?**

Web Service là một ứng dụng phần mềm có khả năng tương tác qua mạng thông qua giao thức HTTP hoặc HTTPS. Nó cho phép các hệ thống khác nhau giao tiếp và chia sẻ dữ liệu và chức năng với nhau qua internet.

**1.1. Đặc điểm của Web Service**

**- Tính độc lập nền tảng**: Web service có thể hoạt động trên nhiều hệ điều hành và môi trường khác nhau.

**- Giao tiếp qua mạng**: Sử dụng các giao thức tiêu chuẩn như HTTP, HTTPS, SMTP, hoặc FTP để trao đổi dữ liệu.

**- Mã hóa dữ liệu:** Thường sử dụng XML hoặc JSON để mã hóa dữ liệu được gửi và nhận.

**- Mô hình yêu cầu-đáp ứng**: Hoạt động theo mô hình client-server, trong đó client gửi yêu cầu và server trả lời.

**1.2. Lợi ích của Web Service**

- Tính tương thích: Có thể tương tác với bất kỳ ứng dụng nào mà không cần quan tâm đến ngôn ngữ lập trình của nó.

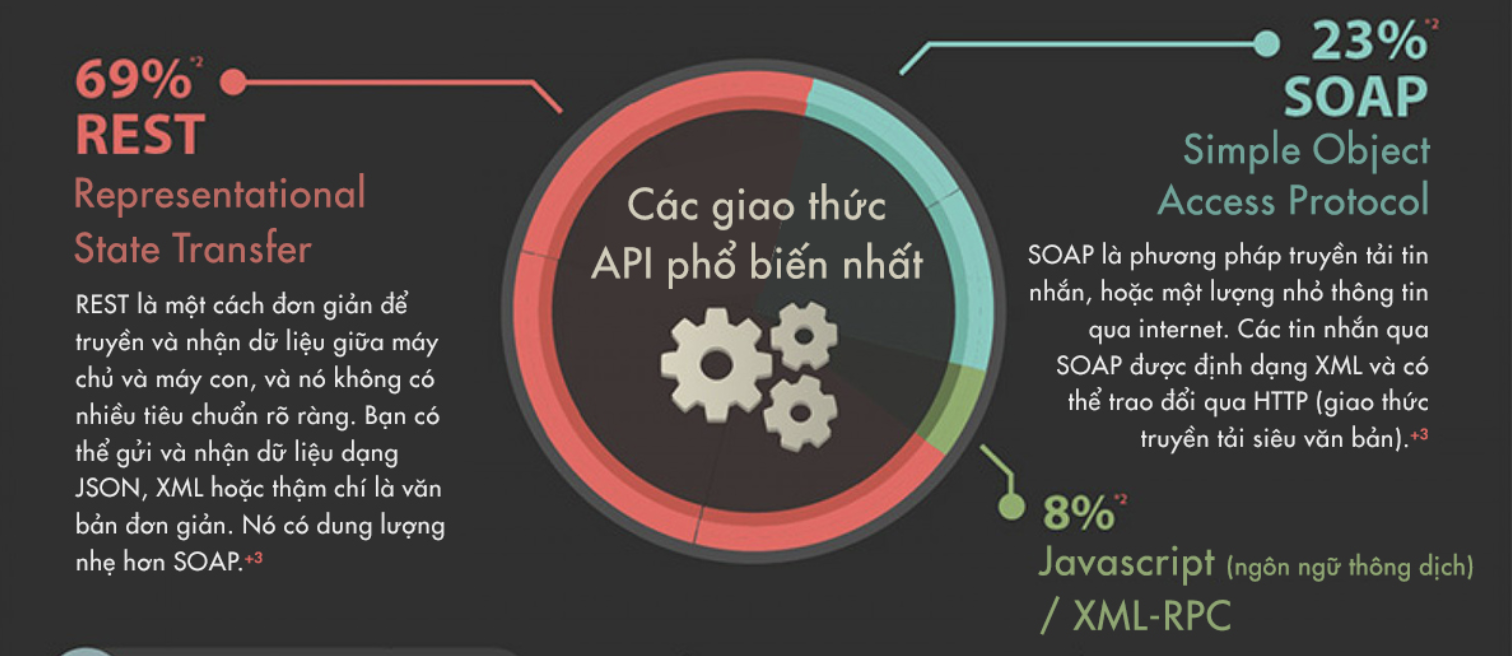
- Khả năng mở rộng: Dễ dàng tích hợp và mở rộng với các hệ thống khác.

- Tái sử dụng: Các dịch vụ có thể được tái sử dụng trong nhiều ứng dụng khác nhau.

- Tính linh hoạt: Có thể được triển khai và quản lý trên nhiều nền tảng và môi trường khác nhau.

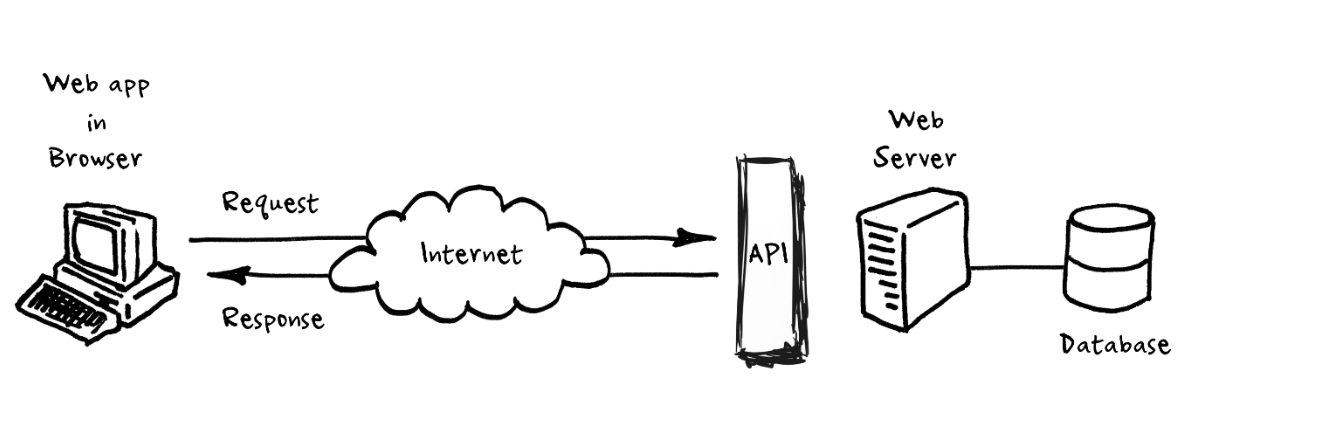
### **2. Các loại Web Service**

SOAP (Simple Object Access Protocol) và REST (Representational State Transfer) là hai phong cách phổ biến cho việc triển khai Web Service. REST thường được ưa chuộng hơn vì đơn giản, linh hoạt và hiệu quả.



### **3. API (Application Programming Interface)**

API chính là Interface của các đối tượng phần mềm. Nó được sử dụng rộng rãi khi nói về mô hình Client-Server và khi nói về cách sử dụng các library (các public method cũng được coi là API).



Nói đơn giản hơn, API là cái cầu nối giữa client và server. Client ở đây có thể là máy tính, điện thoại sử dụng hệ điều hành khác nhau và được viết bằng những ngôn ngữ khác nhau, ví dụ như Swift, Nodejs, Java. Tương tự, server back-end cũng được viết bằng các ngôn ngữ khác nhau. Để client và server có thể giao tiếp được với nhau chúng phải nói cùng 1 “ngôn ngữ”. Ngôn ngữ ấy chính là API.

**II.**  **Các thành phần chính của Web API và Những điểm nổi bật của Web API**

### **1. Các thành phần của Web API**

Web API hỗ trợ restful đầy đủ các phương thức: Get/Post/Put/Delete dữ liệu. Nó giúp bạn xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content format.

**1.1. Methods (HTTP Methods)**

**- Định nghĩa:** Các hành động được thực hiện thông qua API.

- Các loại chính:

GET: Lấy dữ liệu từ server.

POST: dữ liệu đến server.

PUT: Cập nhật dữ liệu trên server.

DELETE: Xóa dữ liệu từ server.

**1.2. Request**

**- Định nghĩa**: Yêu cầu từ client (người sử dụng API) gửi đến server chứa API để thực hiện một hành động hoặc lấy dữ liệu.

**- Thành phần**:

**HTTP Method**: GET (lấy dữ liệu), POST (gửi dữ liệu), PUT (cập nhật dữ liệu), DELETE (xóa dữ liệu).

**Headers**: Metadata gửi kèm theo request, ví dụ như Authorization (cho bảo mật), Content-Type (định dạng dữ liệu).

**Body**: Dữ liệu gửi kèm theo request, thường là JSON hoặc XML, đặc biệt quan trọng với các phương thức POST và PUT.

**1.3. Response**

**- Định nghĩa**: Phản hồi từ server gửi về client sau khi xử lý request.

**- Thành phần**:

**- Status Code**: Mã trạng thái HTTP phản ánh kết quả của request, ví dụ 200 (OK), 404 (Not Found), 500 (Internal Server Error).

**- Headers**: Metadata của response, như Content-Type (định dạng dữ liệu trả về).

**- Body**: Dữ liệu thực trả về, thường là JSON hoặc XML.

### **2. Điểm nổi bật**

- Tự động hóa sản phẩm : Với **web API**, chúng ta sẽ tự động hóa quản lý công việc, cập nhật luồng công việc, giúp tăng năng suất và tạo hiệu quả công việc cao hơn.

- Khả năng tích hợp linh động: API cho phép lấy nội dung từ bất kỳ website hoặc ứng dụng nào một cách dễ dàng nếu được cho phép, tăng trải nghiệm người dùng. API hoạt động như một chiếc cổng, cho phép các công ty chia sẻ thông tin được chọn nhưng vẫn tránh được những yêu cầu không mong muốn.

- Cập nhật thông tin thời gian thực: API có chức năng thay đổi và cập nhật thay đổi theo thời gian thực. Với công nghệ này, dữ liệu sẽ được truyền đi tốt hơn, thông tin chính xác hơn, dịch vụ cung cấp linh hoạt hơn.

- Có tiêu chuẩn chung dễ sử dụng: Bất kỳ người dùng, công ty nào sử dụng cũng có thể điều chỉnh nội dung, dịch vụ mà họ sử dụng. Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, [dependency injection](https://topdev.vn/blog/dependency-injection-la-gi/), unit test.

**III. Nguyên lý hoạt động của API (Working Principles)**

**1.Nguyên lý hoạt động**



Hình 1.1 Nguyên Lý Họat Động

- Yêu cầu (Request): Một ứng dụng khách (client) gửi một yêu cầu đến máy chủ (server) thông qua một URL API. Yêu cầu này có thể bao gồm thông tin về những gì client muốn thực hiện (ví dụ: lấy dữ liệu người dùng, gửi dữ liệu đơn hàng).

- Xử lý (Processing): Máy chủ nhận yêu cầu, xử lý nó và thực hiện các tác vụ cần thiết, chẳng hạn như truy xuất dữ liệu từ cơ sở dữ liệu hoặc thực hiện các tính toán. Phản hồi (Response): Sau khi xử lý, máy chủ gửi lại phản hồi cho client.

- Phản hồi này thường ở dạng JSON hoặc XML, chứa dữ liệu mà client yêu cầu hoặc thông báo về kết quả của yêu cầu (ví dụ: dữ liệu người dùng, thông báo thành công/ thất bại).

### **2.** **Ưu và nhược điểm của Web API**

**2.1 Ưu điểm :**

- Web API được sử dụng hầu hết trên các ứng dụng desktop, ứng dụng mobile và ứng dụng website.

- Linh hoạt với các định dạng dữ liệu khi trả về client: Json, XML hay định dạng khác.

- Nhanh chóng xây dựng HTTP service: URI, request/response headers, caching, versioning, content formats và có thể host trong ứng dụng hoặc trên IIS.

- Mã nguồn mở, hỗ trợ chức năng RESTful đầy đủ, sử dụng bởi bất kì client nào hỗ trợ XML, Json.

- Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.

- Giao tiếp hai chiều được xác nhận trong các giao dịch, đảm bảo độ tin cậy cao.

**2.2.Nhược Điểm :**

Do web API còn khá mới nên chưa thể đánh giá nhiều về nhược điểm.Tuy nhiên, có hai nhược điểm dễ dàng nhận thấy:

. Web API chưa hoàn toàn phải là RESTful service, mới chỉ hỗ trợ mặc định GET, POST

. Để sử dụng hiệu quả cần có kiến thức chuyên sâu, có kinh nghiệm backend tốt

. Tốn thời gian và chi phí cho việc phát triển, nâng cấp và vận hành

. Có thể gặp vấn đề về bảo mật khi hệ thống bị tấn công nếu không giới hạn điều kiện kỹ.

## **IV. Kiểu dữ liệu JSON, XML**

### **1. Định nghĩa JSON (JavaScript Object Notation)**

- JSON là một kiểu định dạng dữ liệu nhẹ, tuân theo một quy luật nhất định. Nó dựa trên cú pháp của JavaScript, nhưng độc lập với ngôn ngữ.

- JSON biểu diễn dữ liệu dưới dạng các cặp key-value. Nó hỗ trợ các cấu trúc dữ liệu như đối tượng và mảng

- JSON có ưu điểm là dễ đọc, Hỗ trợ tốt trong các ngôn ngữ lập trình hiện đại, kích thước nhỏ, truyền tải nhanh

### **2. Định nghĩa XML (Extensible Markup Language)**

**-** XML là một ngôn ngữ đánh dấu mở rộng, được thiết kế để mô tả và truyền tải dữ liệu.

- XML sử dụng các thẻ (tags) để đánh dấu dữ liệu.

-XML có ưu điểm là hỗ trợ tốt việc xác thực và định nghĩa cấu trúc dữ liệu thông qua DTD và XML Schema, mô tả dữ liệu phức tạp và có cấu trúc rõ ràng.

## **V. Triển khai Web API dựa trên công nghệ ASP.NET Core**

### **1. ASP.NET Core là gì?**

ASP.NET Core là một framework phát triển ứng dụng web và dịch vụ web trên nền tảng .NET Core. Nó cung cấp các công cụ và thư viện cho việc xây dựng và triển khai các ứng dụng web hiệu quả và linh hoạt.

### **2. Cấu trúc của một Web API ASP.NET Core**

Một Web API ASP.NET Core thường bao gồm các Controller, Route, Model và Middleware.

- **Controller**: Là các class chịu trách nhiệm xử lý các yêu cầu và phản hồi từ client.

- **Route**: Xác định cách các yêu cầu HTTP sẽ được định tuyến đến các controller và action tương ứng.

- **Model**: Đại diện cho dữ liệu được truyền giữa client và server.

- **Middleware**: Là các thành phần cho phép xử lý các yêu cầu HTTP trước khi chúng đến các endpoint.

### **3. Triển khai một Web API ASP.NET Core**

**Bước 1: Tạo một dự án Web API**: Sử dụng Visual Studio hoặc command line để tạo một dự án Web API ASP.NET Core mới.

**Bước 2: Định nghĩa các Controller và Routes**: Tạo các Controller để xử lý các yêu cầu từ client và định nghĩa các route để định tuyến các yêu cầu đến các Controller tương ứng.

**Bước 3: Xử lý yêu cầu và phản hồi**: Viết mã logic trong các Controller để xử lý các yêu cầu và trả về phản hồi cho client.

**Bước 4: Kiểm thử và triển khai**: Kiểm thử API của bạn và sau đó triển khai nó lên một máy chủ hoặc dịch vụ đám mây.

### **4. Kết nối với SQL Server (khi cần)**

Để kết nối với SQL Server ta cần sử dụng Entity Framework Core (EF Core)

## **VI. Các vấn đề bảo mật (Security Considerations)**

### **1. Định nghĩa Authentication**

**Authentication (Xác thực)** là quá trình xác định danh tính người dùng. Ví dụ: một người A đăng nhập bằng tên người dùng và mật khẩu mà anh ấy đăng ký trước đó và máy chủ sử dụng tên người dùng và mật khẩu của anh ấy cung cấp để xác thực danh tính người đăng nhập là người dùng A

### **2. Định nghĩa Authorization**

**Authorization (Ủy quyền)** là quá trình quyết định xem người dùng được xác thực có được phép thực hiện một hành động trên một tài nguyên cụ thể (Tài nguyên API Web) hay không? Ví dụ: User A (người dùng được xác thực) có quyền nhận tài nguyên (GET) nhưng không có quyền tạo mới tài nguyên (POST)

## **VII. Mã trạng thái (Status Code)**

### **1. Định nghĩa mã trạng thái (Status Code)**

**Mã trạng thái (Status Code)** là một phần quan trọng trong các phản hồi HTTP mà máy chủ gửi lại cho máy khách sau mỗi yêu cầu HTTP. Mã trạng thái cung cấp thông tin về kết quả của yêu cầu, giúp máy khách hiểu được liệu yêu cầu đã được xử lý thành công hay không, và nếu không thì vấn đề là gì.

### **2. Một số mã trạng thái thường gặp**

**200 OK:** Yêu cầu thành công. Máy chủ đã trả về thông tin yêu cầu.

**201 Created:** Yêu cầu đã tạo một tài nguyên mới thành công.

**204 No Content:** Yêu cầu đã được xử lý thành công, nhưng không có dữ liệu nào được trả về.

**400 Bad Request**: Yêu cầu không hợp lệ do cú pháp không đúng hoặc thiếu thông tin cần thiết.

**401 Unauthorized**: Người dùng cần phải được xác thực trước khi tiếp tục yêu cầu.**403 Forbidden:** Máy chủ đã hiểu yêu cầu, nhưng từ chối phản hồi do nguyên nhân nào đó. Thường là do quyền truy cập bị từ chối.

**404 Not Found:** Máy chủ không tìm thấy tài nguyên được yêu cầu.

**405 Method Not Allowed:** Phương thức không cho phép với user hiện tại.

**500 Internal Server Error:** Là mã lỗi phổ biến nhất, được sử dụng khi máy chủ gặp phải một lỗi không xử lý được khi xử lý yêu cầu.

**503 Service Unavailable:** Máy chủ tạm thời không thể xử lý yêu cầu vì quá tải hoặc bảo trì.

# Chương 2: Xây dựng ứng dụng Web

**1. Bài toán ứng dụng**

**1.1 Phát biểu bài toán ứng dụng**

Xây dựng trang web quản lí cho thuê xe

Trong dự án này, chúng tôi sẽ phát triển một trang web quản lí cho thuê xe để cung cấp một nên tảng thuận tiện và linh hoạt cho người dùng. Mục tiêu chính của dự án là tạo ra một website đặt xe trực tuyến dễ dàng và thoải mái, đồng đảm bảo tính linh hoạt và tiện ích cho cả người dùng và người quản trị

**1.2 Xây dựng yêu cầu**

- Giao diện người dùng: Hiển thị các danh mục xe để người dùng lựa chọn, hiển thị khách hàng, và hiện thị các hợp đồng thuê xe để người quản trị quản lí.

- Chức năng đặt xe

Cho phép người quản trị thêm, sửa, xóa thông tin xe, khách hàng, hợp đồng thuê xe…

Thực hiện chức năng thanh toán

Xe thông tin Khách hàng, hợp đồng thuê và thông tin xe

- Chức năng quản trị

Quản lí danh mục xe

Quản lí nhân viên

Quản lí khách hàng

Quản lí hợp đồng thuê

Quản lí thanh toán của khách hàng

- Công nghệ và ngôn ngữ:

Sử dụng ngôn ngữ lập trình là C# và mô hình MVC

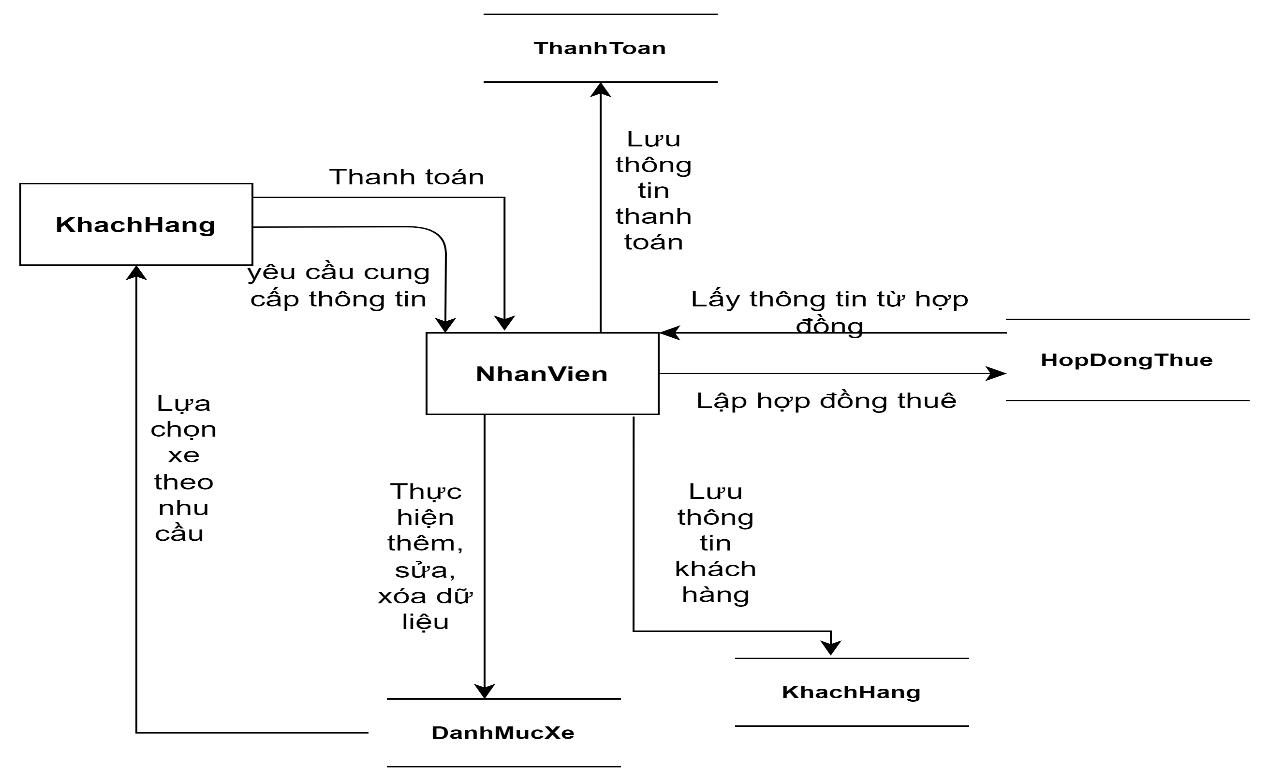
Sử dụng cơ sở dữ liệu như MySQL

Sử dụng HTML, CSS, và JavaScript tạo giao diện trang web.

**1.3 Xây dựng mô hình ER**

****

Hình 1.2 Sơ đồ ER

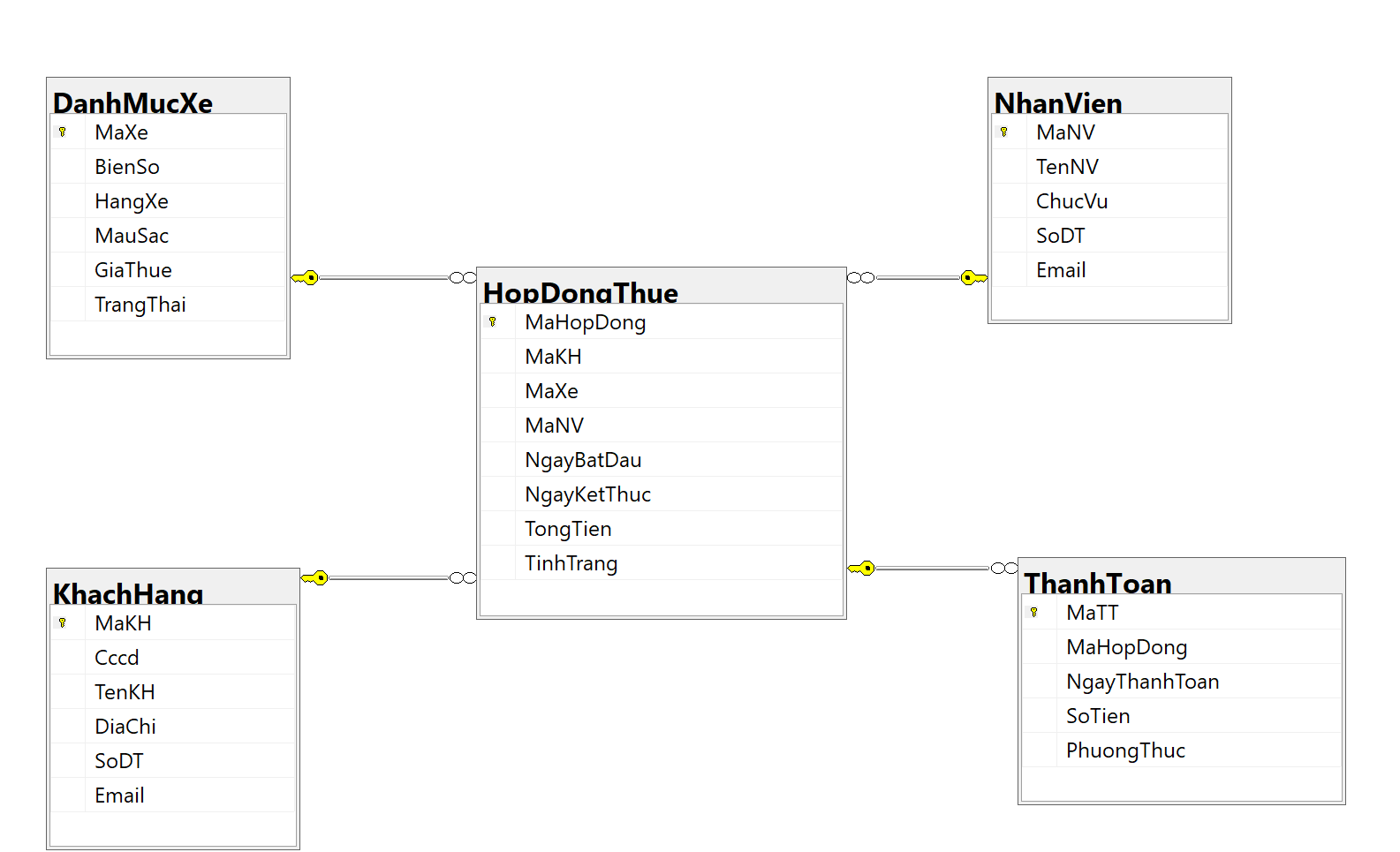


Hình 1.3 Sơ đồ thể hiện công việc

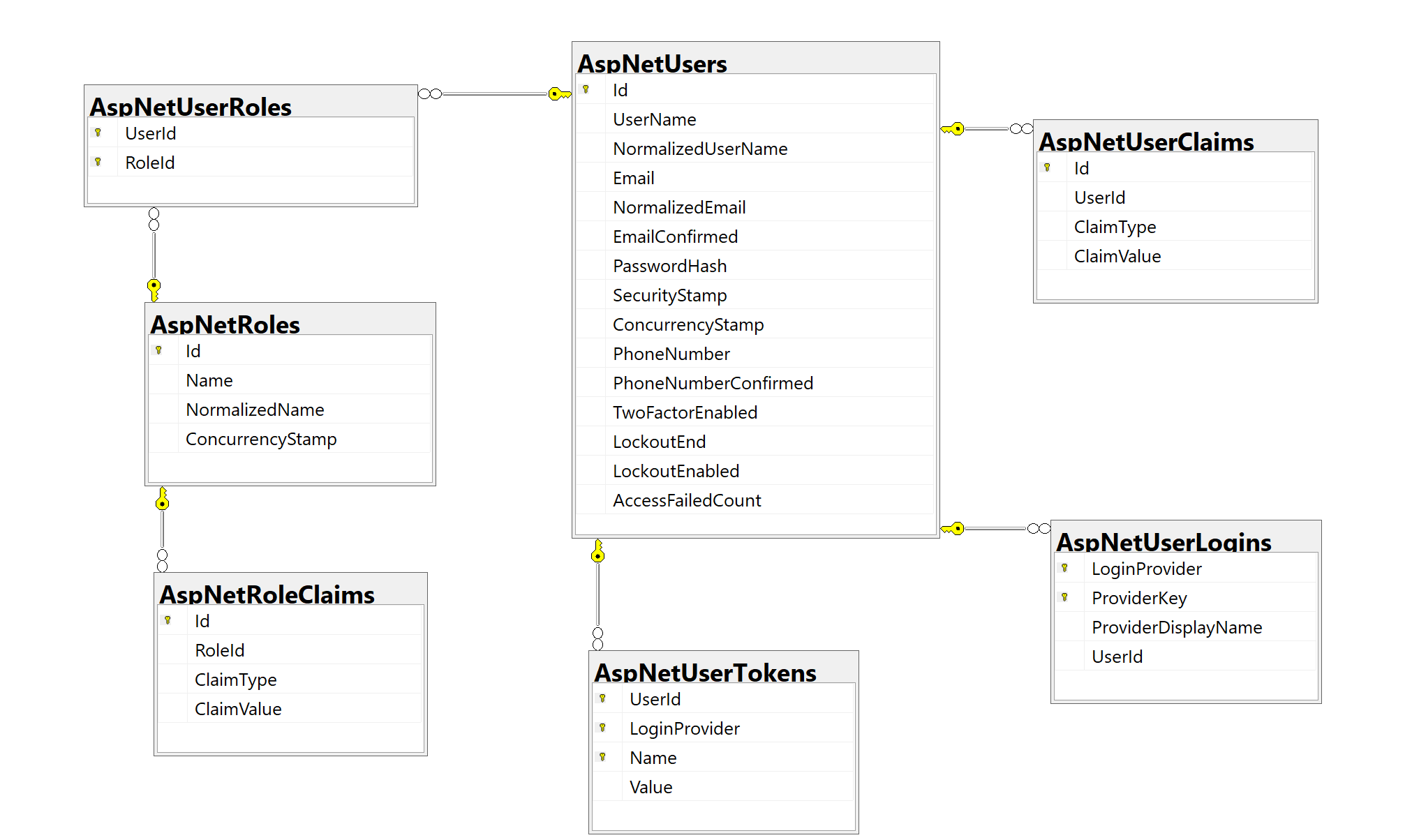
### **1.4 Sơ đồ Use Case**

Hình 1.4 Sơ đồ Case

**1.5. Xây dựng cơ sở dữ liệu**

- Sử dụng mô hình Code-First để xây dựng dữ liệu

Hình 1.4 Sơ đồ cơ sở dữ liệu

****

Hình 1.5 Hình sơ đồ cơ sở dữ liệu về Account

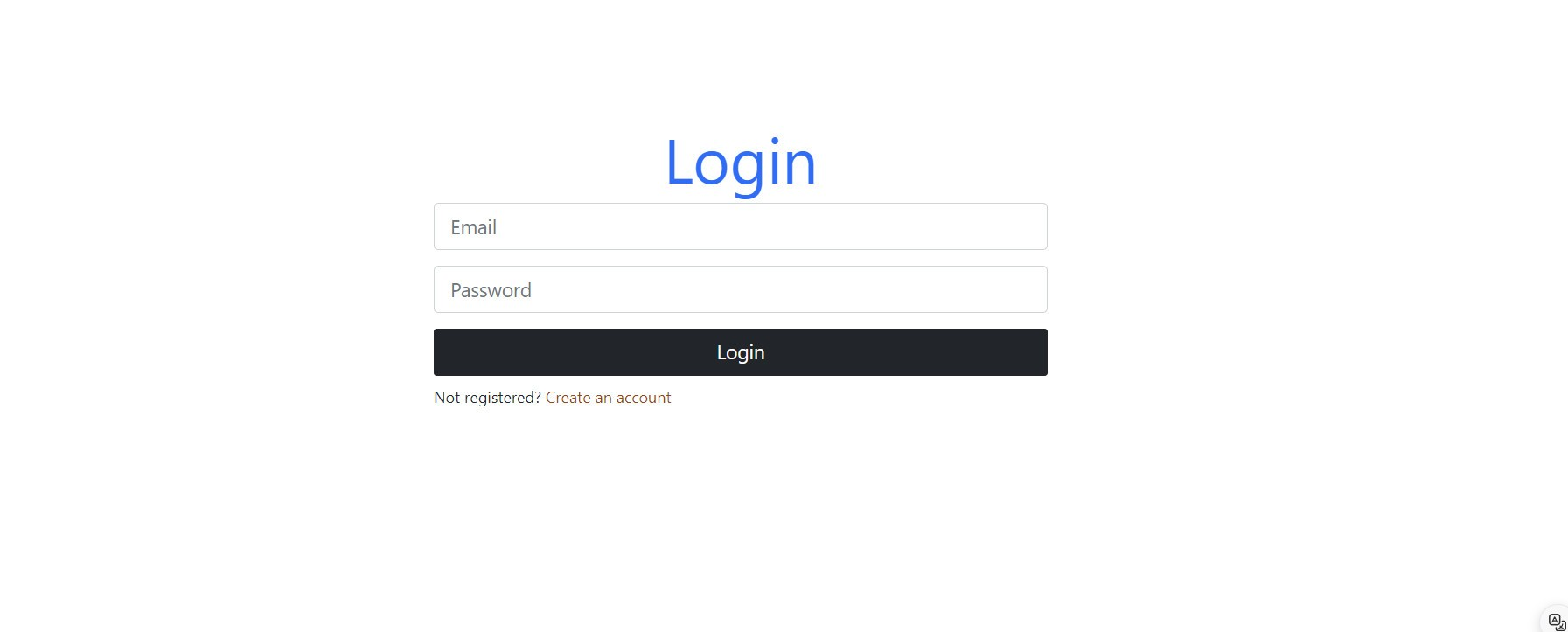
# Chương 3 Kết quả chương trình

## **1. Register, Login và Logout**

### **1.1 Register**

Người dùng cần phải đăng kí tài khoản trước khi login vào trang web, việc đăng kí yêu cầu Username và Password

### **1.2 Login**

****

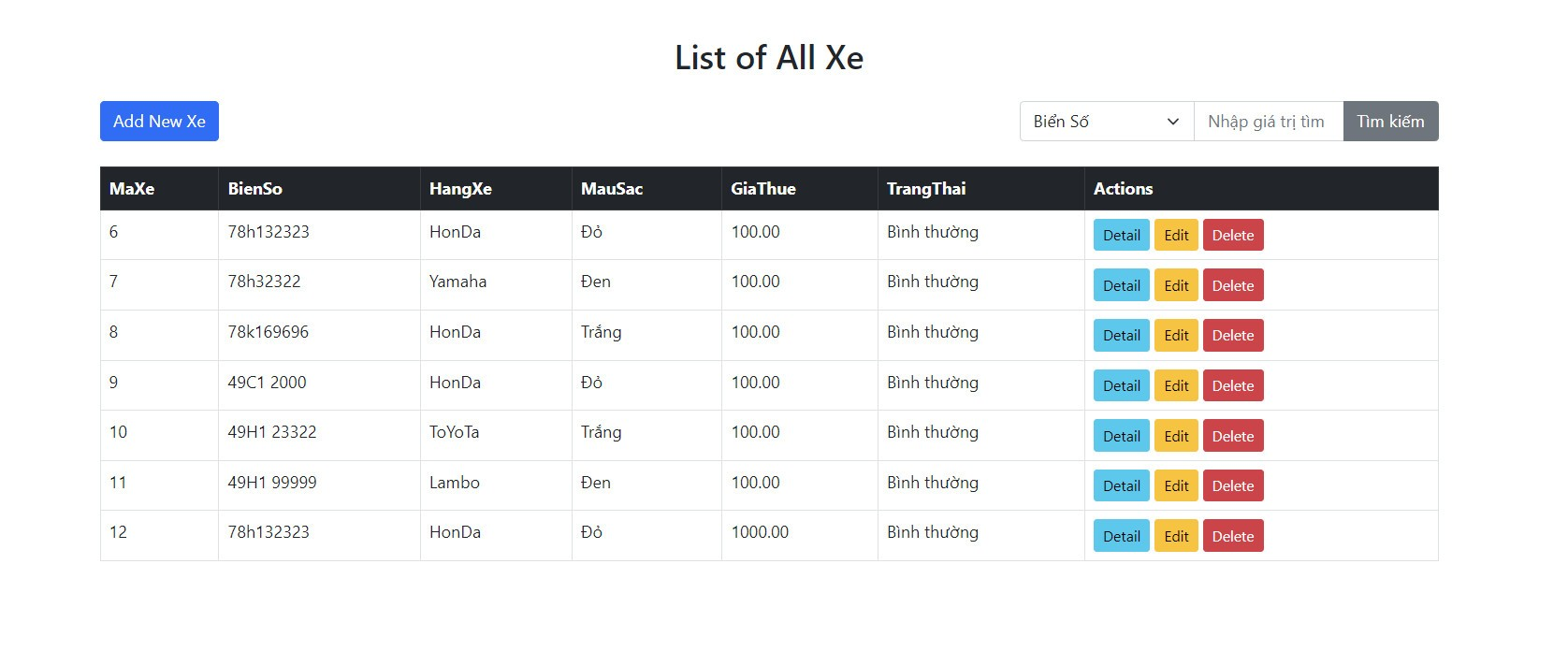
Khi đã có tài khoản thì tiến hành login vào, ở đây nó sẽ kiểm tra xem tài khoản của bạn có tồn tại hay chưa, hoặc bạn nhập thông tin đăng nhập đung chưa

### **1.3 Logout**

Khi mà người dùng không muốn sử dụng nữa thì ta có thể logout tài khoản

## **2. Giao diện người dùng**

### **2.1 Trang danh mục xe**

**2.1.1 Giao diện chính của trang danh mục xe**

Trong trang Danh Mục Xe:

Hiển thị tất cả các thông tin bao gồm mã xe, biển số, hãng xe, màu sắc, giá thuê, trạng thái của từng đối tượng xe

Người dùng cũng có thể tìm kiếm Xe theo biển số

Hiển thị các chức năng mà người dùng có thể thực hiện (Detail, Edit, Delete)

Khi nhấn delete thì đối tượng đó sẽ bị xóa mất

**2.1.2 Chức năng Add Xe**



Tạo mới Xe theo các thông tin BienSo, HangXe, MauSac, GiaThue, TrangThai, Image

**2.1.3 Chức năng Detail trong danh mục xe**

Chức năng này cho phép người dùng xem đầy đủ thông tin của một đối tượng xe mà mình muốn xem

**2.1.4 Chức năng Edit trong danh mục xe**



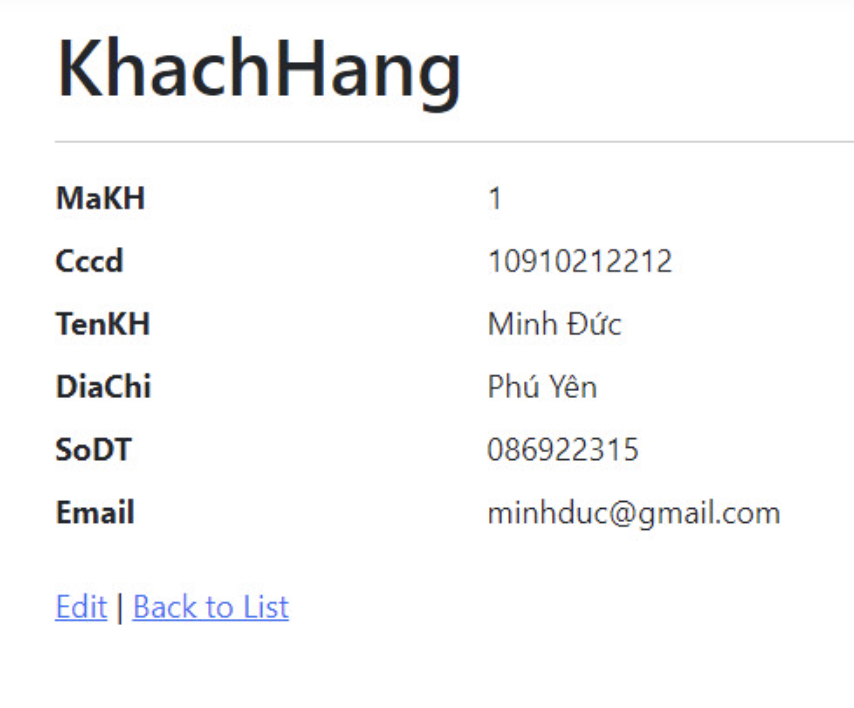
Chức năng edit cho phép người dùng thay đổi, sửa chữa thông tin của từng đối tượng xe

### **2.2 Trang khách hàng**

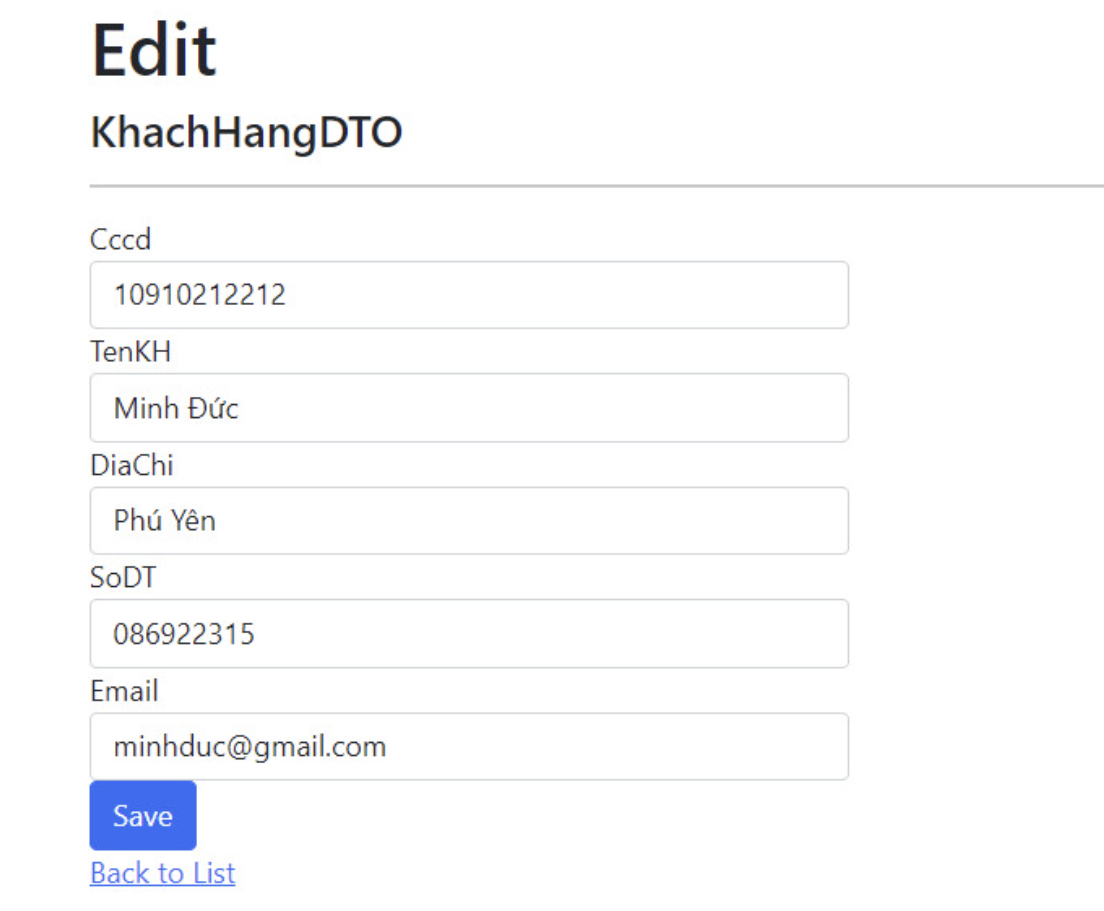
**2.2.1 Giao diện chính của trang khách hàng**

Hiển thị tất cả thông tin của từng Khách hàng

Bấm Delete để xóa

**2.2.2 Chức năng Detail**

Chức năng này cho phép người dùng xem đầy đủ thông tin của một đối tượng khách hàng mà mình muốn xem

**2.2.3 Chức năng Edit**

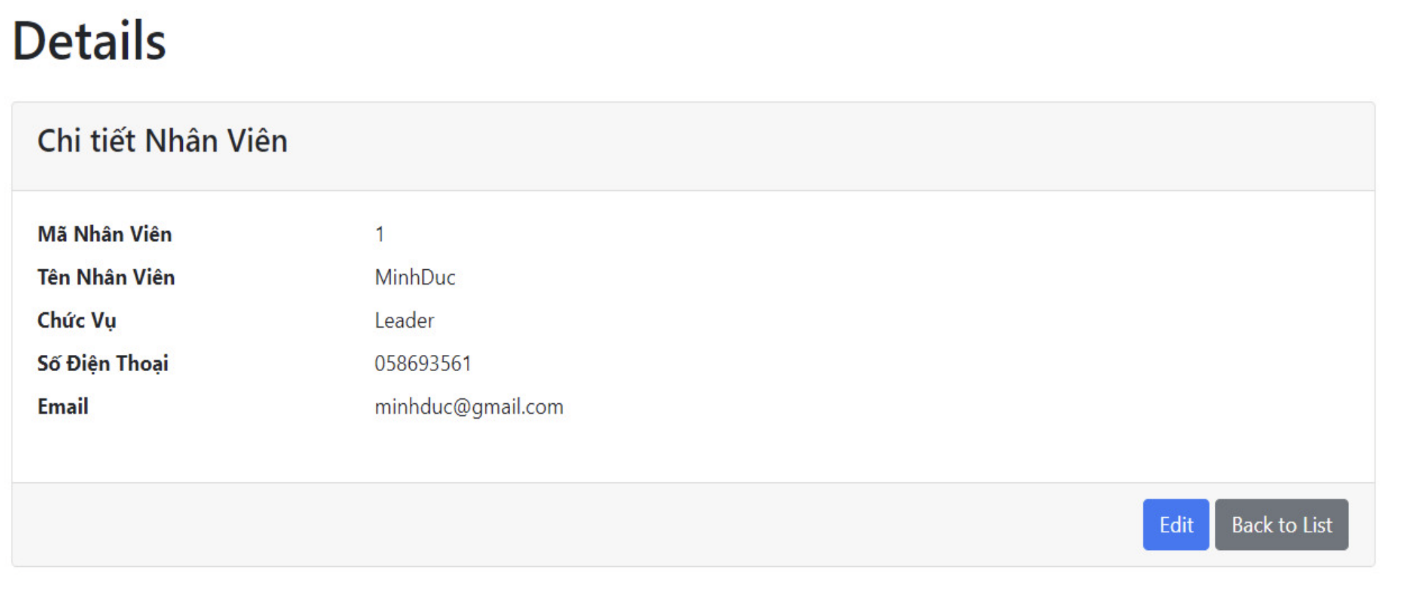
Chức năng edit cho phép người dùng thay đổi, sửa chữa thông tin của từng đối tượng khách hàng

### **2.3 Trang nhân viên**

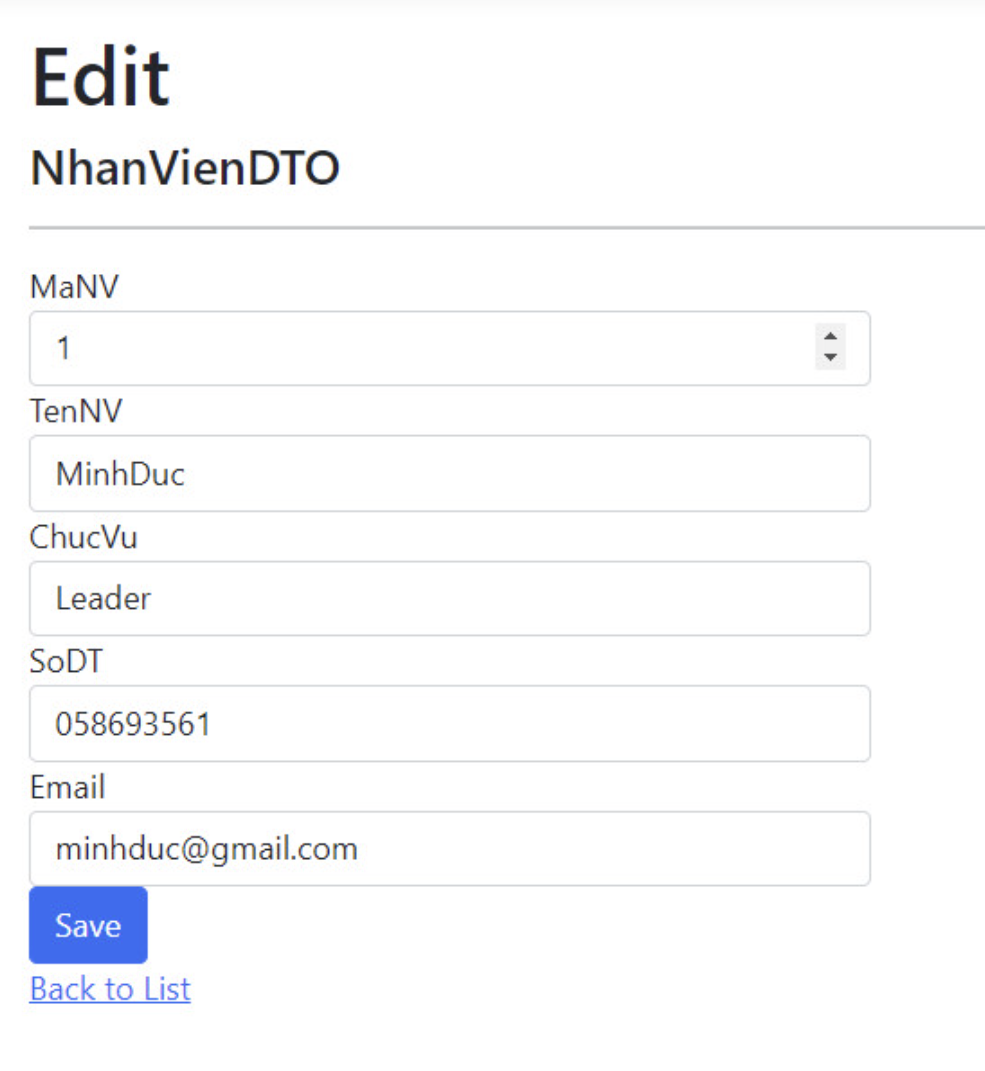
**2.3.1 Giao diện chính của trang nhân viên**

Hiển thị tất cả thông tin của từng Nhân viên

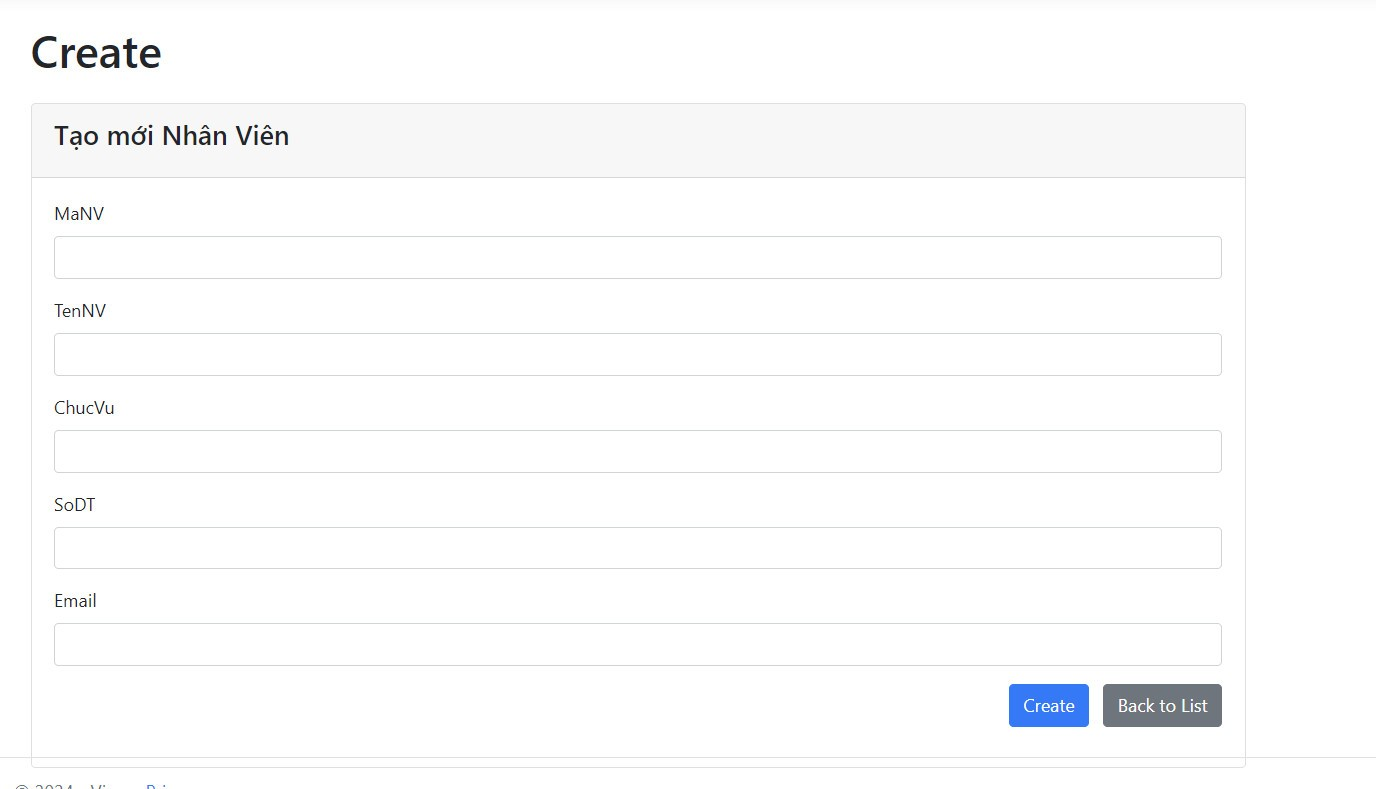
Bấm Delete để xóa

**2.3.2 Chức năng Detail**

Chức năng này cho phép người dùng xem đầy đủ thông tin của một đối tượng nhân viên mà mình muốn xem

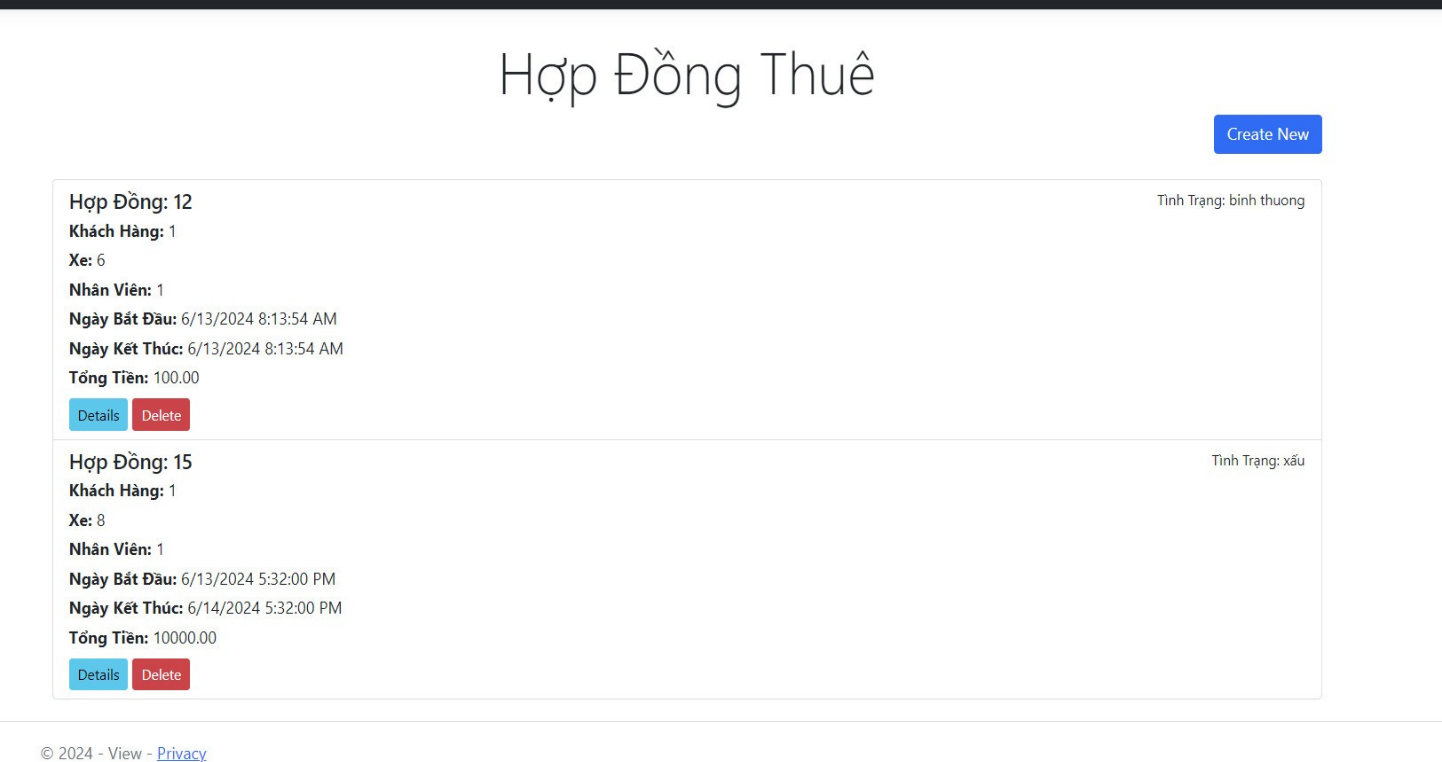
** 2.3.3 Chức năng Edit**

Chức năng edit cho phép người dùng thay đổi, sửa chữa thông tin của từng đối tượng nhân viên

**** **2.3.4 Chức năng Add nhân viên**

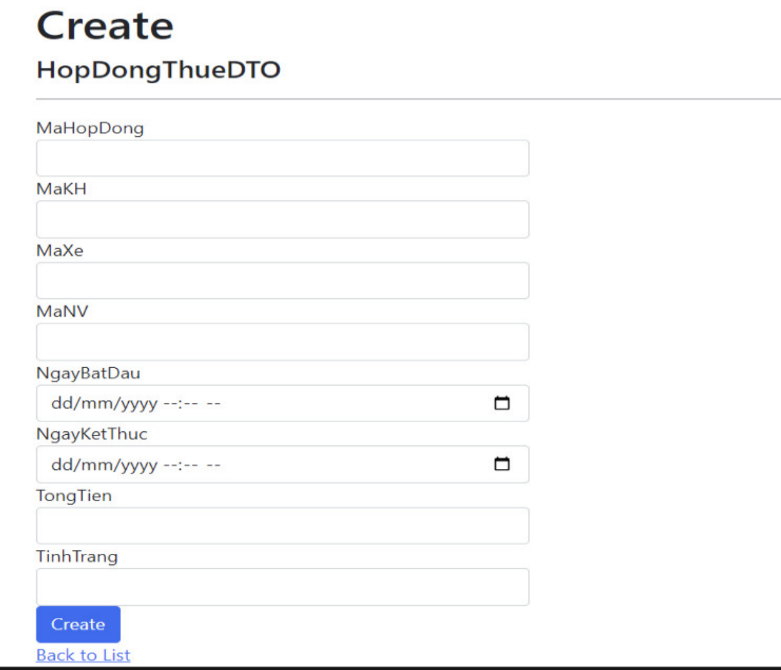
Thêm nhân viên mới theo các thông tin MaNV, TenNV, ChucVu, SoDT, Email

### **2.4 Trang hợp đồng thuê xe**

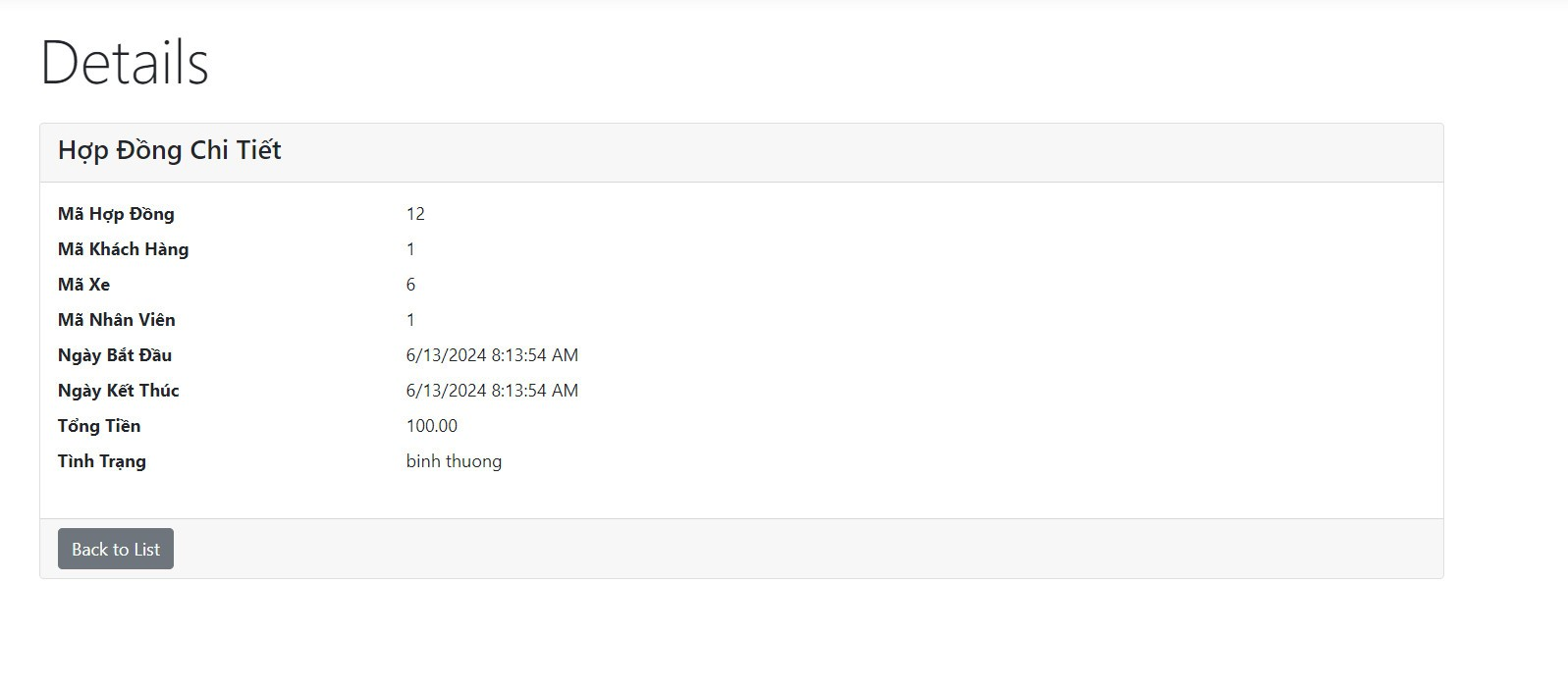
**2.4.1 Giao diện chính của trang hợp đồng thuê**

Hiển thị tất cả thông tin của từng Hợp đồng

Bấm Delete để xóa

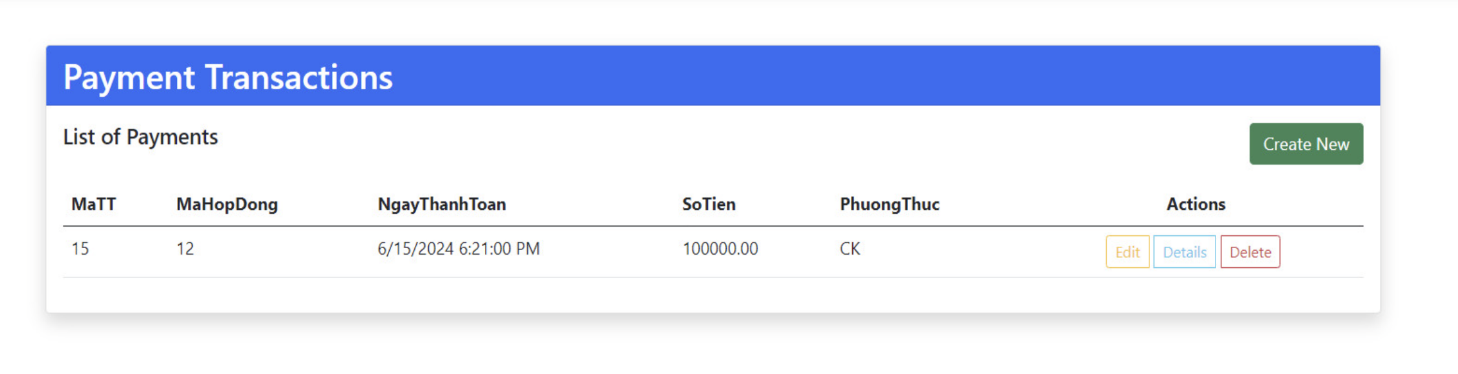
**2.4.2 Chức năng Add**

Tạo một hợp đồng mới

**2.4.3 Chức năng Detail**

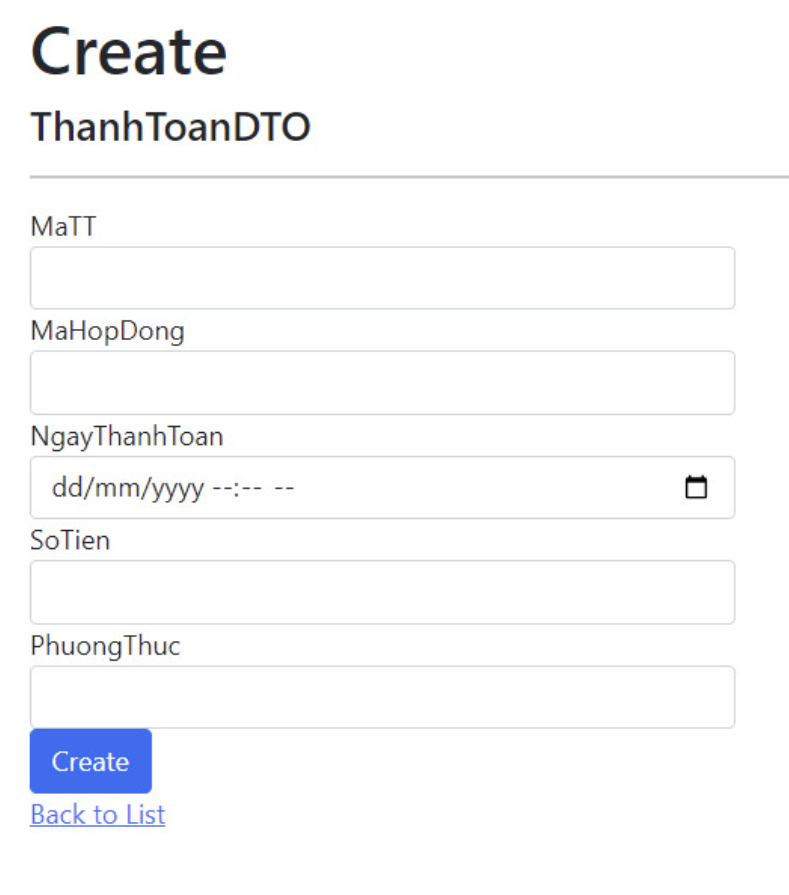
Chức năng này cho phép người dùng xem đầy đủ thông tin của một đối tượng nhân viên mà mình muốn xem

### **2.5 Trang thanh toán**

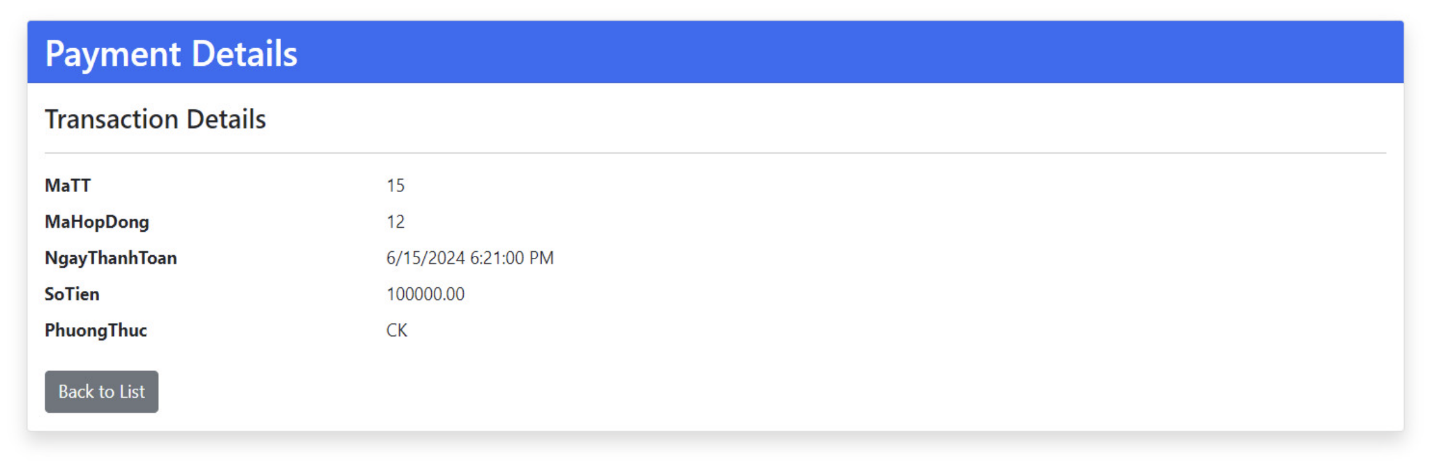
**2.5.1 Giao diện chinh của trang thanh toán**

Hiển thị tất cả thông tin của từng thanh toan

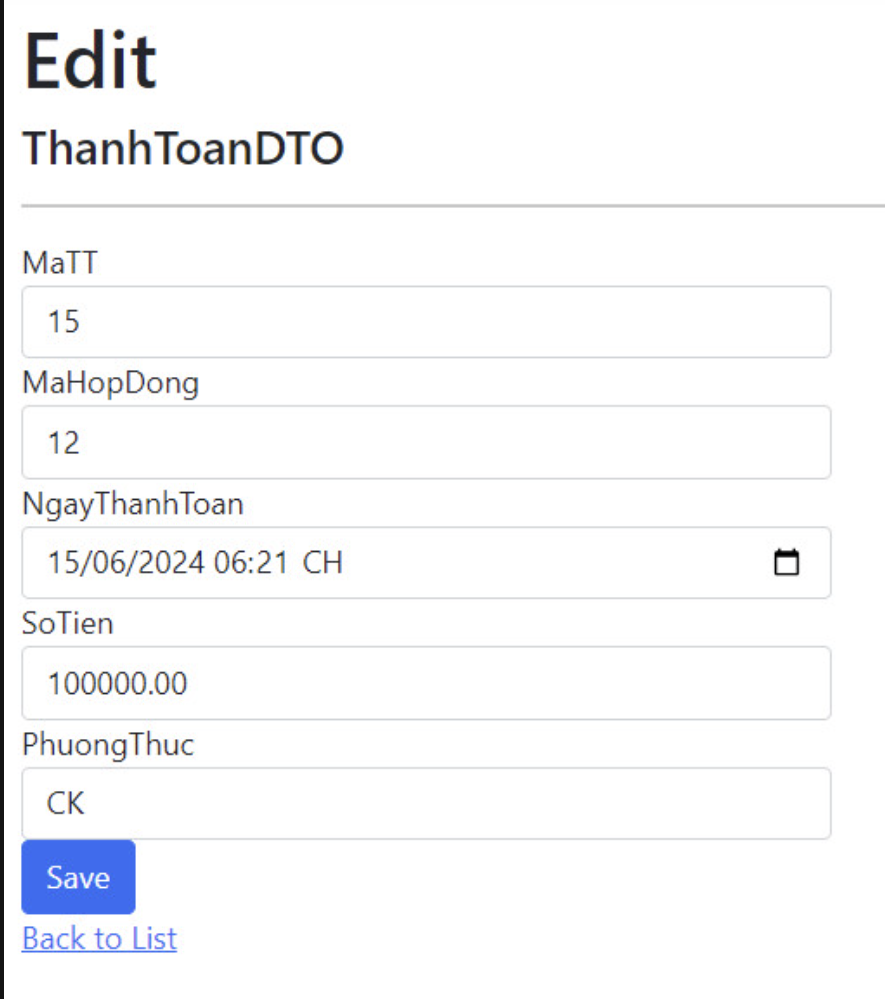
Bấm Delete để xóa

**2.5.2 Chức năng Add**

Tạo một thanh toán mới

**2.5.3 Chức năng Detail**

Chức năng này cho phép người dùng xem đầy đủ thông tin của một thanh toan mà mình muốn xem

**2.5.4 Chức năng Edit**

Chức năng edit cho phép người dùng thay đổi, sửa chữa thông tin của từng thanh toan

# KẾT LUẬN

Trong báo cáo này, chúng tôi đã đi sâu vào việc phát triển và triển khai một Web API sử dụng công nghệ ASP.Net Core . Qua quá trình nghiên cứu và thực hành, chúng tôi đã đạt được những kết quả sau:

**1.** T**ìm hiểu công nghệ và công cụ phát triển Web API:** Chúng tôi đã nghiên cứu về các công nghệ và công cụ phổ biến trong việc phát triển Web API như RESTful, JSON,..

2. **Thiết kế và xây dựng Web API:** Báo cáo đã trình bày quá trình thiết kế và xây dựng một Web API hoàn chỉnh, từ việc xác định yêu cầu hệ thống, thiết kế cấu trúc dữ liệu, đến việc triển khai các endpoint API.

**3. Đánh giá và rút kinh nghiệm:** Qua quá trình phát triển, chúng tôi đã rút ra được nhiều kinh nghiệm quý báu trong việc lập trình Web API

# DANH MỤC TỪ NGỮ VIẾT TẮT

|  |  |
| --- | --- |
| **VIẾT TẮT** | **NỘI DUNG** |
| SMTP | Simple Mail Transfer Protocol |
| HTTP | Hypertext Transfer Protocol |
| SQL | Structured Query Language |
| HTTPS | Hypertext Transfer Protocol Secure |

# Nguồn Tham Khảo

[1] Sách: ASP.NET Web API Build RESTful web applications and services on the .NET framework

[2] Sách: API testing book from A to Y – for Vietnamese testers

[3] Giáo trình môn Lập Trình Web 2 – Nguyễn Đức Tấn