TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

Tel. (84-236) 3736949, Fax. (84-236) 3842771

Website: http://dut.udn.vn/khoacntt, E-mail: cntt@dut.udn.vn



BÁO CÁO THỰC TẬP CÔNG NHÂN

PHẦN THIẾT KẾ WEBSITE

**ĐỀ TÀI :**

Xây dựng trang web review sách

Đà Nẵng, 12/201

**Mục lục**

[MỞ ĐẦU 1](#_Toc69407696)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 2](#_Toc69407697)

[1.1. Website 2](#_Toc69407698)

[1.1.1. Khái nệm 2](#_Toc69407699)

[1.1.2. Phân loại website 2](#_Toc69407700)

[1.2. Ngôn ngữ javascript 2](#_Toc69407701)

[1.2.1. Tổng quan về Javascript 2](#_Toc69407702)

[1.2.2. Javascript là gì 2](#_Toc69407703)

[1.3. Nodejs 3](#_Toc69407704)

[1.3.1. Nodejs là gì ? 3](#_Toc69407705)

[1.3.2. Ứng dụng của Nodejs 4](#_Toc69407706)

[1.3.3. Nhược điểm của Nodejs 4](#_Toc69407707)

[1.3.4. Ưu điểm của Nodejs 4](#_Toc69407708)

[1.4. Express JS Framework 5](#_Toc69407709)

[1.4.1. ExpressJS là gì 5](#_Toc69407710)

[1.4.2. Cấu trúc của ExpressJS 6](#_Toc69407711)

[1.4.3. Khái niệm Router trong ExpressJS: 6](#_Toc69407712)

[1.4.4. Tìm hiểu các method all của router 7](#_Toc69407713)

[1.4.5. Tìm hiểu về router.METHOD() 8](#_Toc69407714)

[1.5. Phân tích yêu cầu 9](#_Toc69407715)

[1.5.1. Yêu cầu về quản lý website 9](#_Toc69407716)

[1.5.2. Yêu cầu về giao diện, chức năng đối với người dùng 9](#_Toc69407717)

[CHƯƠNG 2: PHÂN TÍCH, THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG 9](#_Toc69407718)

[2.1. Phân tích các chức năng của hệ thống 10](#_Toc69407719)

[2.1.1. Admin 10](#_Toc69407720)

[2.1.2. User 10](#_Toc69407721)

[2.2. Thiết kế cơ sở dữ liệu 10](#_Toc69407722)

[2.3. Tổ chức chương trình 11](#_Toc69407723)

[2.3.1. Tổ chức thư mục 11](#_Toc69407724)

[2.3.2. Tập tin public 12](#_Toc69407725)

[2.3.3. Tập tin routes 12](#_Toc69407726)

[2.3.4. Tập tin views 12](#_Toc69407727)

[CHƯƠNG 3: TRIỂN KHAI 13](#_Toc69407728)

[3.1. Môi trường cài đặt 13](#_Toc69407729)

[3.2. Kết quả triển khai 13](#_Toc69407730)

[3.3. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 16](#_Toc69407731)

[3.3.1. KẾT QUẢ ĐẠT ĐƯỢC 16](#_Toc69407732)

[3.3.2. HƯỚNG PHÁT TRIỂN 16](#_Toc69407733)

[CHƯƠNG 4: TÀI LIỆU THAM KHẢO 17](#_Toc69407734)

MỞ ĐẦU

**1.**  **Tổng quan về đề tài**

Hiện nay, đời sống ngày càng cao, nhu cầu mua xe ô tô càng nhiều. Vì vậy, em xây dựng trang web bán xe cũng như quán lý bán xe ô tô, hơn nữa củng cố các kiến thức đã học và cũng như tự học thêm những công nghệ mới.

**2. Mục đích và ý nghĩa của đề tài**

**2.1. Mục đích**

* Ôn tập, củng cố các kiến thức đã học ở môn Chuyên đề Mạng.
* Tìm hiểu, nghiên cứu các công nghệ mới

**2.2. Ý nghĩa**

* Có thể tự hoàn thiện một website hoàn chỉnh từ khâu phân tích yêu cầu và code, qua đó củng cố kiến thức về lập trình webs.

**3. Phương pháp thực hiện**

Phương pháp phân tích tổng hợp từ tài liệu: đọc các tài liệu có liên quan đến Nodejs và Express để có đầy đủ kiến thức lập trình.

Phương pháp thống kê, điều tra: xem về 3 trang web có nội dung tương tự.

Phương pháp phân tích thiết kế hệ thống: tiến hành phân tích, thiết kế database.

Phương pháp thử nghiệm, đánh giá kết quả: tiến hành triển khai hệ thống review sách.

**4. Bố cục của đề tài**

Báo cáo đề tài bao gồm các nội dung sau:

*Mở đầu*

*Chương 1: Cơ sở lý thuyết*

*Chương 2: Phân tích và triển khai hệ thống*

*Chương 3: Triển khai*

*Kết luận và hướng phát triển.*

# CƠ SỞ LÝ THUYẾT

## Website

### Khái nệm

Website là một tập hợp các trang web (web pages) bao gồm văn bản, hình ảnh, video, flash v.v... thường chỉ nằm trong một [tên miền (domain name)](http://www.vietsol.net/ten-mien/khai-niem-ten-mien/) hoặc tên miền phụ (subdomain). Trang web được lưu trữ (web hosting) trên máy chủ web (web server) có thể truy cập thông qua Internet.

Website đóng vai trò là một văn phòng hay một cửa hàng trên mạng [Internet](http://www.vietsol.net/tags/internet)– nơi giới thiệu thông tin về doanh nghiệp, sản phẩm hoặc dịch vụ do doanh nghiệp cung cấp… Có thể coi website chính là bộ mặt của doanh nghiệp, là nơi để đón tiếp và giao dịch với các khách hàng, đối tác trên Internet.

### Phân loại website

Có thể là công việc của một cá nhân, một doanh nghiệp hoặc các tổ chức, và thường dành riêng cho một số chủ đề cụ thể hoặc mục đích. Bất kỳ trang web có thể chứa một siêu liên kết vào bất kỳ trang web khác, do đó, phân biệt các trang web cá nhân, như cảm nhận của người sử dụng. Tạm thời phân loại như sau:

* Trang web cá nhân
* Trang web thương mại
* Trang web của chính phủ
* Trang web tổ chức phi lợi nhuận

## Ngôn ngữ javascript

### Tổng quan về Javascript

Javascript là một ngôn ngữ chương trình máy tính động. Javascript được nhúng hoặc tích hợp vào tập tin HTML, dùng để tạo các script ở máy client và máy server. Các script ở client được thực thi tại trình duyệt và các script ở server được thực hiện trên server. JavaScript và Java là hai ngôn ngữ hoàn toàn khác nhau, cả về khái niệm và thiết kế.

JavaScript được phát minh bởi Brendan Eich vào năm 1995, và trở thành một tiêu chuẩn ECMA năm 1997.

ECMA-262 là tên chính thức. ECMAScript 2016 (tháng 6 năm 2016) là phiên bản mới nhất của JavaScript.

### Javascript là gì

JavaScript là một ngôn ngữ lập trình đa nền tảng (cross-platform), ngôn ngữ lập trình kịch bản, hướng đối tượng. JavaScript là một ngôn ngữ nhỏ và nhẹ (small and lightweight). Khi nằm bên trong một môi trường (host environment), JavaScript có thể kết nối tới các object của môi trường đó và cung cấp các cách quản lý chúng (object).

JavaScript chứa các thư viện tiêu chuẩn cho các object, ví dụ như:  Array, Date, và Math, và các yếu tố cốt lõi của ngôn ngữ lập trình như: toán tử (operators), cấu trúc điều khiển (control structures), và câu lệnh. JavaScript có thể được mở rộng cho nhiều mục đích bằng việc bổ sung thêm các object; ví dụ:

* Client-side JavaScript - JavaScript phía máy khách, JavaScript được mở rộng bằng cách cung cấp các object để quản lý trình duyệt và Document Object Model (DOM) của nó. Ví dụ, phần mở rộng phía máy khách cho phép một ứng dụng tác động tới các yếu tố trên một trang HTML và phản hồi giống các tác động của người dùng như click chuột, nhập form, và chuyển trang.
* Server-side JavaScript - JavaScript phía Server, JavaScript được mở rộng bằng cách cung cấp thêm các đối tượng cần thiết để để chạy JavaScript trên máy chủ. Ví dụ, phần mở rộng phía server này cho phép ứng dụng kết nối với cơ sở dữ liệu (database), cung cấp thông tin một cách liên tục từ một yêu cầu tới phần khác của ứng dụng, hoặc thực hiện thao tác với các tập tin trên máy chủ.

## Nodejs

### Nodejs là gì ?

Node.js là một mã nguồn mở, một môi trường cho các máy chủ và ứng dụng mạng.

Node.js sử dụng Google V8 JavaScript engine để thực thi mã, và một tỷ lệ lớn các mô-đun cơ bản được viết bằng JavaScript. Các ứng dụng node.js thì được viết bằn JavaScript.

Node.js chứa một thư viện built-in cho phép các ứng dụng hoạt động như một Webserver mà không cần phần mềm như Nginx, Apache HTTP Server hoặc IIS.

Node.js cung cấp kiến trúc hướng sự kiện (event-driven) và non-blocking I/O API, tối ưu hóa thông lượng của ứng dụng và có khả năng mở rộng cao

Mọi hàm trong Node.js là không đồng bộ (asynchronous). Do đó, các tác vụ đều được xử lý và thực thi ở chế độ nền (background processing).

### Ứng dụng của Nodejs

* Xây dựng websocket server (Chat server)
* Hệ thống Notification (Giống như facebook hayTwitter)
* Ứng dụng upload file trên client
* Các máy chủ quảng cáo
* Các ứng dụng dữ liệu thời gian thực khác.

### Nhược điểm của Nodejs

* Ứng dụng nặng tốn tài nguyên Nếu bạn cần xử lý các ứng dụng tốn tài nguyên CPU như encoding video, convert file, decoding encryption… hoặc các ứng dụng tương tự như vậy thì không nên dùng NodeJS (Lý do: NodeJS được viết bằng C++ & Javascript, nên phải thông qua thêm 1 trình biên dịch của NodeJS sẽ lâu hơn 1 chút). Trường hợp này bạn hãy viết 1 Addon C++ để tích hợp với NodeJS để tăng hiệu suất tối đa!
* NodeJS và ngôn ngữ khác NodeJS, PHP, Ruby, Python .NET …thì việc cuối cùng là phát triển các App Web. NodeJS mới sơ khai như các ngôn ngữ lập trình khác. Vậy nên bạn đừng hi vọng NodeJS sẽ không hơn PHP,Ruby,Python… ở thời điểm này. Nhưng với NodeJS bạn có thể có 1 ứng dụng như mong đợi, điều đó là chắc chắn!

### Ưu điểm của Nodejs

Đặc điểm nổi bật của Node.js là nó nhận và xử lý nhiều kết nối chỉ với một single-thread. Điều này giúp hệ thống tốn ít RAM nhất và chạy nhanh nhất khi không phải tạo thread mới cho mỗi truy vấn giống PHP. Ngoài ra, tận dụng ưu điểm non-blocking I/O của Javascript mà Node.js tận dụng tối đa tài nguyên của server mà không tạo ra độ trễ như PHP

JSON APIs Với cơ chế event-driven, non-blocking I/O(Input/Output) và mô hình kết hợp với Javascript là sự lựa chọn tuyệt vời cho các dịch vụ Webs làm bằng JSON.

Ứng dụng trên 1 trang (Single page Application) Nếu bạn định viết 1 ứng dụng thể hiện trên 1 trang (Gmail?) NodeJS rất phù hợp để làm. Với khả năng xử lý nhiều Request/s đồng thời thời gian phản hồi nhanh. Các ứng dụng bạn định viết không muốn nó tải lại trang, gồm rất nhiều request từ người dùng cần sự hoạt động nhanh để thể hiện sự chuyên nghiệp thì NodeJS sẽ là sự lựa chọn của bạn.

Shelling tools unix NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn Process và trả ra 1 luồng khiến cho hiệu xuất hoạt động đạt mức tối đa nhất và tuyệt vời nhất.

Streamming Data (Luồng dữ liệu) Các web thông thường gửi HTTP request và nhận phản hồi lại (Luồng dữ liệu). Giả xử sẽ cần xử lý 1 luồng giữ liệu cực lớn, NodeJS sẽ xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.

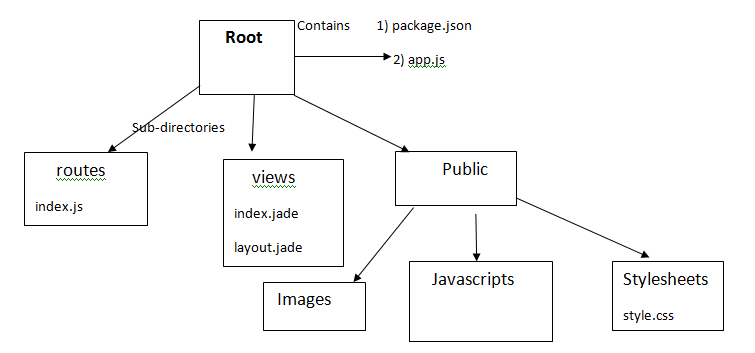
Ứng dụng Web thời gian thực Với sự ra đời của các ứng dụng di động & HTML 5 nên Node.js rất hiệu quả khi xây dựng những ứng dụng thời gian thực (real-time applications) như ứng dụng chat, các dịch vụ mạng xã hội như Facebook, Twitter…

## Express JS Framework

### ExpressJS là gì

* Express js là một Framework nhỏ, nhưng linh hoạt được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile
* Về các package hỗ trợ: Expressjs có vô số các package hỗ trợ nên các bạn không phải lo lắng khi làm việc với Framework này.
* Về performance: Express cung cấp thêm về các tính năng (feature) để dev lập trình tốt hơn. Chứ không làm giảm tốc độ của NodeJS.
* Và hơn hết, các Framework nổi tiếng của NodeJS hiện nay đều sử dụng ExpressJS như một core function, chẳng hạn: SailsJS, MEAN, ....

### Cấu trúc của ExpressJS

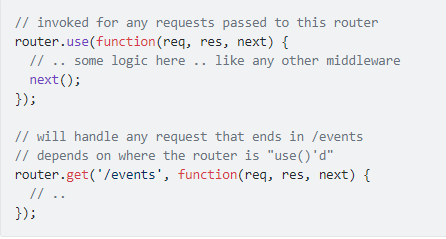
Để hiểu hơn về cấu trúc Expressjs thì hãy xem ảnh dưới đây:

*Hình 1: Cấu trúc của ExpressJS*

Có thể thấy cấu trúc của express js vô cùng đơn giản:

* Root:
* app.js chứa các thông tin về cấu hình, khai báo, các định nghĩa,... để ứng dụng của chúng ta chạy ok.
* package.json chứa các package cho ứng dụng chạy. Nếu bạn nào làm với PHP hoặc RoR rồi thì file này có chức năng tương tự như composer.json hoặc Gemfile
* Folder routes: chứa các route có trong ứng dụng
* Folder view: chứa view/template cho ứng dụng
* Folder public chứa các file css, js, images, ... cho ứng dụng

### Khái niệm Router trong ExpressJS:

* Router là một Object (khác Routing nhé), nó là một instance riêng của middleware và routes (Hai khái niệm này là gì thì chúng ta sẽ tìm hiểu sau nhé). Chính vì nó là một instance của middleware và route nên nó có các chức năng của cả hai. Chúng ta có thể gọi nó là một mini-application
* Các Application dùng ExpressJS làm core đều có phần Router được tích hợp sẵn trong đó.
* Router hoạt động như một middleware nên chúng ta có thể dùng nó như một arguments. Hoặc dùng nó như một arguments cho route khác. Nghe có vẻ khó hiểu đúng không nào. Ví dụ nhé:

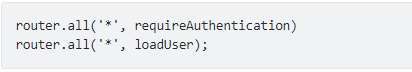
*Hình 2: Ví dụ về Router*

* Chúng ta cũng có thể sử dụng Router để chia route. Chẳng hạn:

### Tìm hiểu các method all của router

router.all(). Method này phù hợp với việc định nghĩa mang tính chất toàn cục cho các prefix

*Hình 3: Ví dụ 1 method all*

Nếu ta đặt route này trên cùng (top) thì nó yêu cầu tất cả các route bên dưới phải được requireAuthentication. Có nghĩa là xác thực trước khi thực hiện một hành động hay một task nào đó tiếp theo, ví dụ là loadUser.

*Hình 4: Ví dụ 2 method all*

Khác với ví dụ trên. Ở ví dụ này ta có một prefix đã được xác định là /api/ thay vì dùng \* . Nghĩa là trước khi request vào các route bên trong API thì phải qua một thao tác xác thực requireAuthentication.

*Hình 5: Ví dụ 3 method all*

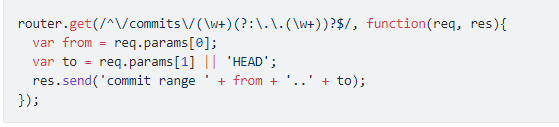
### Tìm hiểu về router.METHOD()

Router.METHOD() cung cấp cho chúng ta chức năng Routing trong ExpressJS. Cụ thể METHOD() ở đây là các HTTP method mà chúng ta thường xuyên sử dụng. Chẳng hạn GET, POST, PUT,...

Lưu ý là tên method phải được viết thường (lowercase)

Ví dụ:

*Hình 6: Ví dụ 1 router.METHOD()*

Nếu muốn bảo mật hơn thì có thể sử dụng Regex để bắt các Endpoint. Ví dụ:

*Hình 7: Ví dụ Regex*

## Phân tích yêu cầu

### Yêu cầu về quản lý website

* Xây dựng trang quản lý có giao diện dễ nhìn, đơn giản, dễ sử dụng.
* Trang quản lý có đầy đủ chức năng tìm kiếm, thêm, sửa, xóa,…
* Trang quản lý phải ổn định và có độ tin cậy cao với người dùng

### Yêu cầu về giao diện, chức năng đối với người dùng

* Giao diện trực quan, dễ sử dụng, tính năng đa dạng,..
* Thu thập thông tin để dự đoán xu hướng của người dùng

# PHÂN TÍCH, THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG

## Phân tích các chức năng của hệ thống

Website được xây dựng phục vụ 2 đối tượng chính là Admin (Quản trị viên) và Khách hàng.

### Admin

* + Đăng ký tài khoản
  + Đăng nhập Website
  + Đăng xuất
  + Quản lý danh mục
  + Quản lý xe
  + Quản lý thông tin cá nhân

### User

* + Xem thông tin xe
  + Xem bài viết
  + Đăng nhập, đăng ký, đăng xuất
  + Viết bài đánh giá
  + Đánh giá

## Thiết kế cơ sở dữ liệu

Bảng Sản phẩm:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Key** |
| 1 | **STT** | int(5) | No | Primary key |
| 2 | **Name** | varchar(225) | No |  |
| 3 | **Image** | varchar(225) | Yes |  |
| 4 | **Status** | varchar(225) | No |  |
| 5 | **Amount** | int(11) | No |  |
| 6 | **Price** | varchar(50) | No |  |

Bảng User:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Key** |
| 1 | **username** | varchar(20) | No | Primary key |
| 2 | **password** | varchar(100) | No |  |
| 3 | **Email** | varchar(50) | Yes |  |
| 4 | **Phonenumber** | varchar(13) | Yes |  |

## Tổ chức chương trình

### Tổ chức thư mục

**|---- VU\_GK**

**|---- node-modules**

**|---- public**

**|---- assets**

**|---- img**

**|---- css**

**|---- images**

**|---- routes**

**|---- database.js**

**|---- index.js**

**|---- product.js**

**|---- user.js**

**|---- validate**

**|---- user.validate.js**

**|---- views**

**|---- partials**

**|---- header.ejs**

**|---- index.ejs**

**|---- add-player.ejs**

**|---- edit-player.ejs**

**|---- login.ejs**

**|---- app.js**

**|---- database.sql**

**|---- package-lock.json**

**|---- package.json**

### Tập tin public

Lưu trữ những tập tin .png, .jpg được đăng tải trên website về máy.

### Tập tin routes

Xử lý những yêu cầu thực thi từ người sử dụng với sản phẩm như thêm/xoá/sửa và đăng nhập/ đăng xuất.

Product.js xử lý những tác vụ liên quan đến sản phẩm.

User.js xử lý tác vụ đăng nhập, đăng xuất.

### Tập tin views

Chữa những biểu mẫu hiển thị trên website như:

* Bảng hiển thị danh sách sản phẩm.
* Biểu mẫu thêm sản phẩm vào CSDL.
* Biểu mẫu chỉnh sửa sản phẩm vào CSDL
* Biểu mẫu tìm kiếm sản phẩm vào CSDL

# TRIỂN KHAI

## Môi trường cài đặt

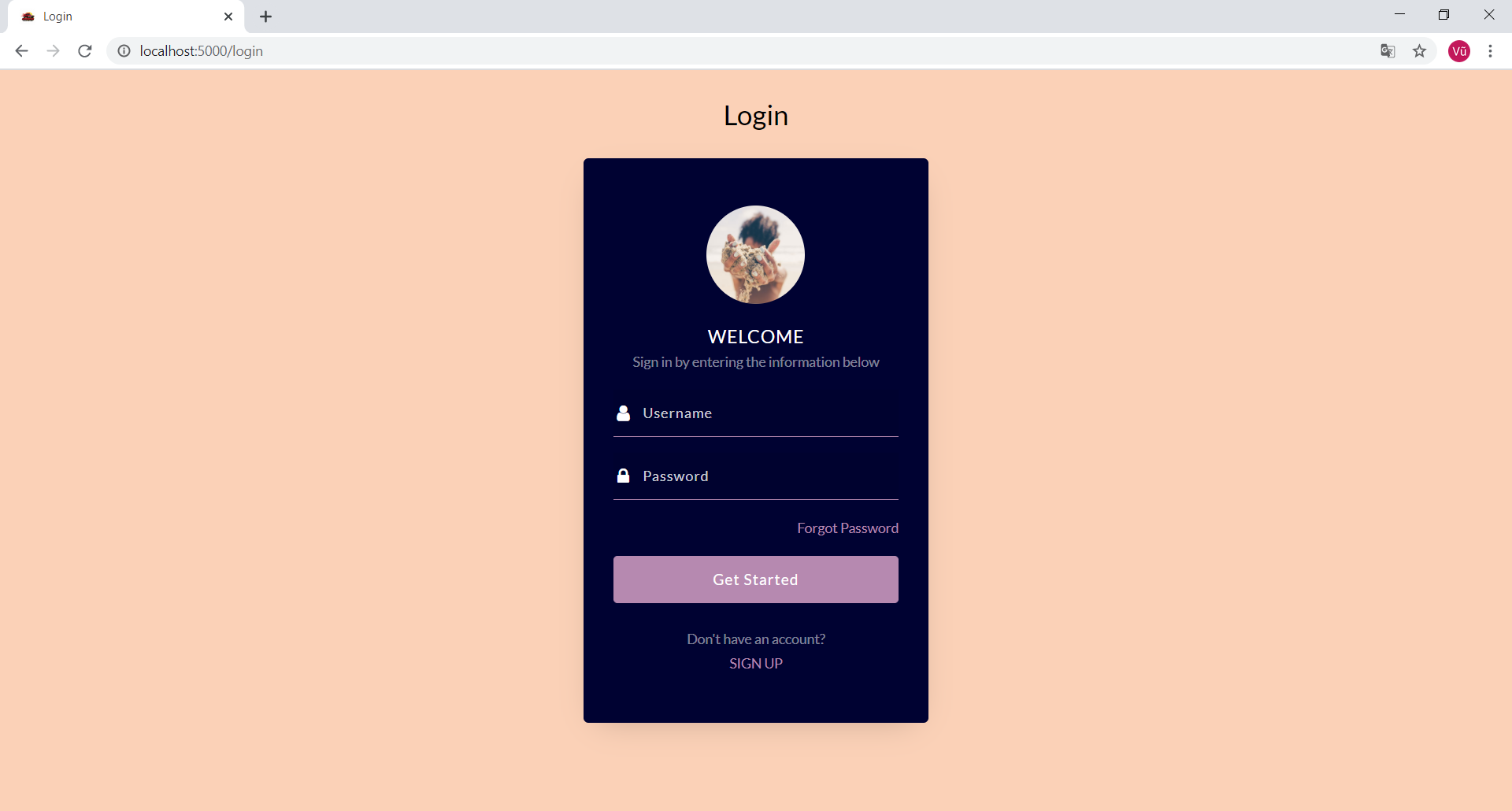
* Web site được xây dựng trên nền tảng :
  + Back-end là nodejs sử dụng framework ExpressJS
  + Front-end: HTML, CSS, Javascript, Jquery, Ajax.
  + Hệ quản trị cơ sở dữ liệu: MySQL
  + Quản lý source code: Github
* Các bước thực hiện đề tài:
  + Lập đề cương chi tiết cho đề tài
  + Tìm hiểu một số trang web bán xe ôtô
  + Lên danh sách công việc cần lầm, phân bổ công việc
  + Xây dựng database
  + Xây dựng web
  + Hoàn thành báo cáo

## Kết quả triển khai

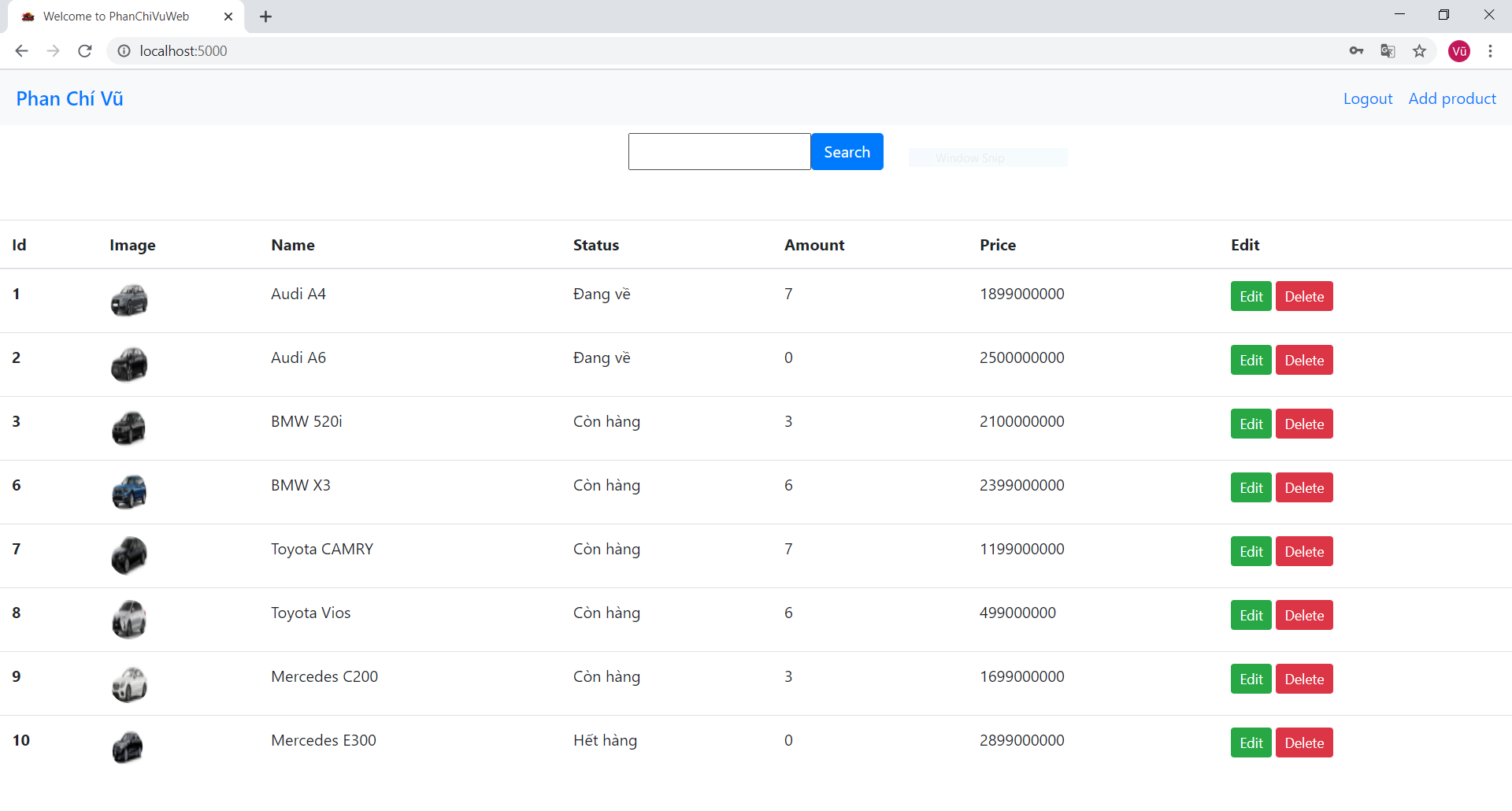
* Giao diện đăng nhập:

Đăng nhập với username: phanchivu

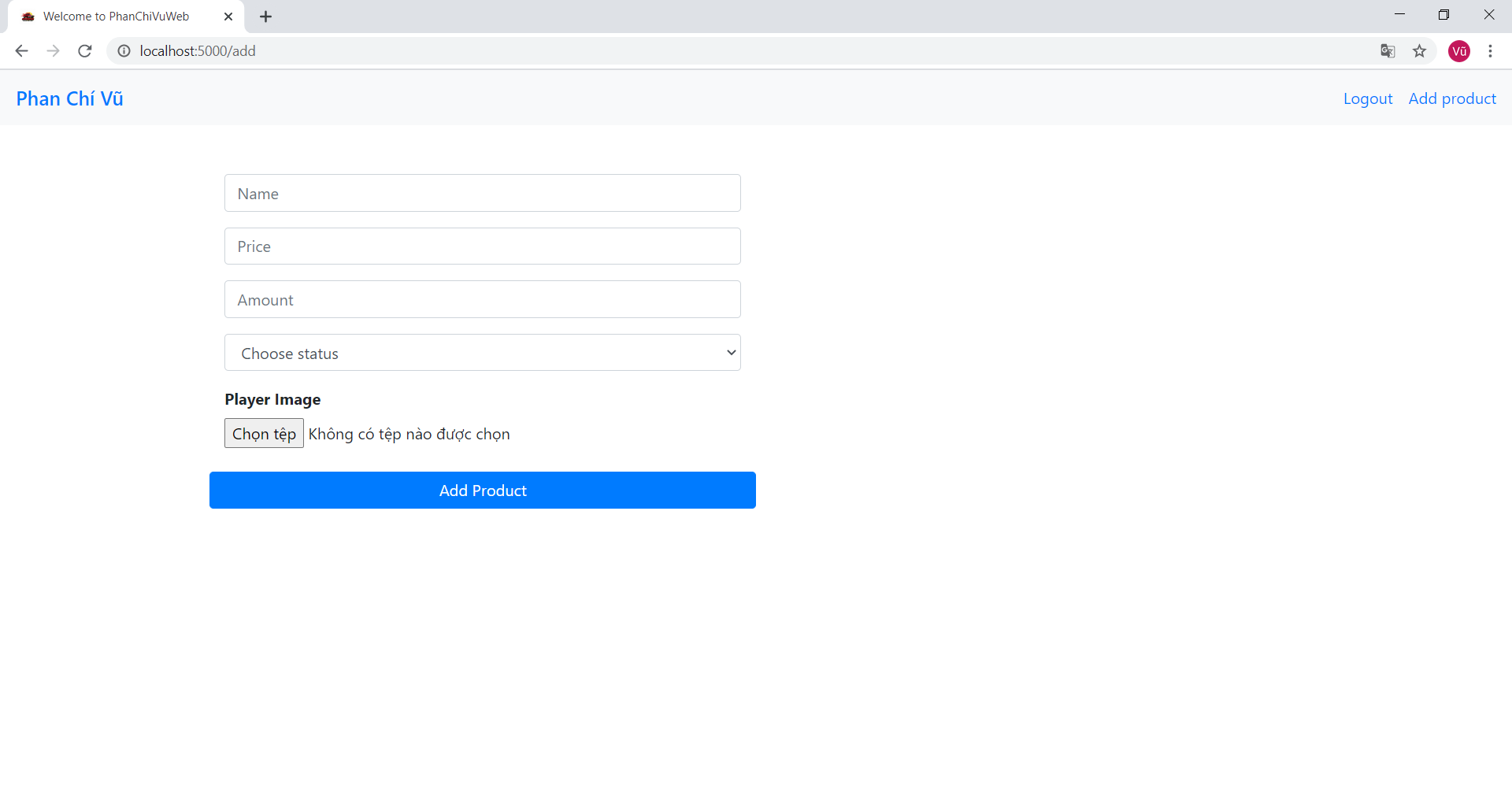
Password: 123



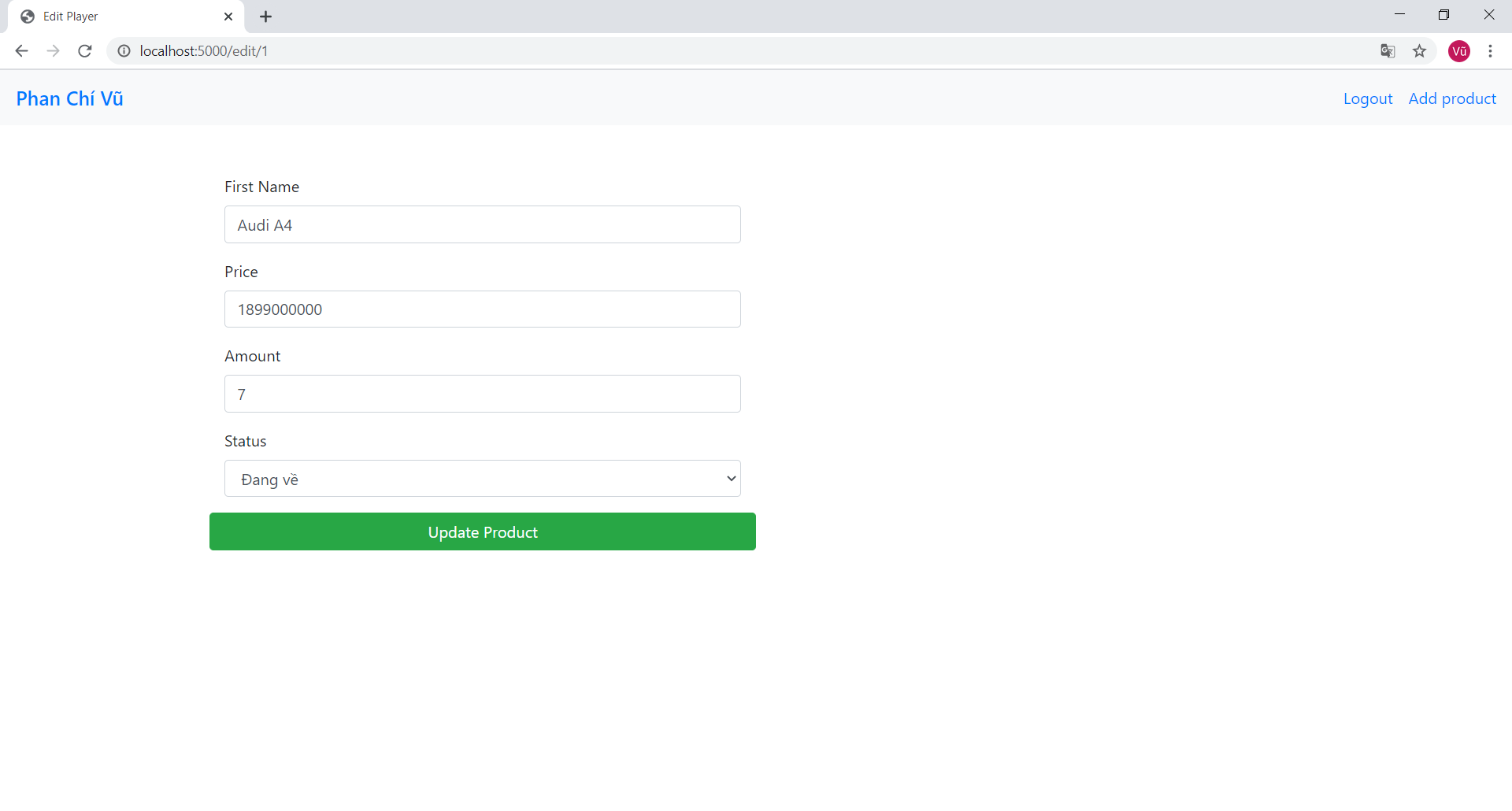
* Giao diện trang chủ khi đăng nhập:



* Giao diện thêm sản phẩm:



* Giao diện chỉnh sửa sản phẩm:



## KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### KẾT QUẢ ĐẠT ĐƯỢC

Trong thời gian tìm hiểu, nghiên cứu cơ sở lý thuyết và triển khai ứng dụng công nghệ, đề tài đã đạt được những kết quả sau:

*Về mặt lý thuyết,* đề tài đã đạt được:

* + - Ứng dụng các kiến thức về lập trình Website như Nodejs, ExrpessJS, HTML5, CSS3, Ajax, Jquery, … dể xây dựng web site.
    - Ứng dụng kiến thức về cấu trúc dữ liệu, cơ sở dữ liệu, phân tích thiết kế hệ thống thông tin, và các hệ quản trị cơ sở dữ liệu như MySQL vào đề tài.

*Về mặt thực tiễn ứng dụng*, đề tài đã đạt được:

* + - Tạo được một bán và quản lý xe ô tô vận hành khá tốt trên môi trường Internet.
    - Đáp ứng được nhu cầu người sử dụng.

Tuy nhiên, đề tài còn tồn tại các vấn đề như sau:

* Quá trình load dữ liệu chưa được tối ưu
* Một số thiếu sót do kiến thức về NodeJS và ExpressJS là mới đối với sinh viên.

### HƯỚNG PHÁT TRIỂN

Một số số hướng nghiên cứu và phát triển của đề tài như sau:

* Nghiên cứu chỉnh sửa design cho đúng tiêu chuẩn UX, giúp người dùng dễ dàng sử dụng
* Nghiên cứu phát triển thêm một vài tính năng mới
* Tìm hiểu tăng cường bảo mật website

# TÀI LIỆU THAM KHẢO

**Tiếng Việt**

1. Phạm Hữu Đức (2005), *Cơ sở dữ liệu và hệ thống thông tin địa lý GIS*, Nhà xuất bản Xây dựng.

**Tiếng Anh**

1. Nodejs Application Developer’s Guide – MarkLogic.
2. Web Development with Node and Express – Ethan Brown.

**Internet**

1. *https://o7planning.org/vi/11931/huong-dan-nodejs-cho-nguoi-moi-bat-dau*
2. *http://* [*www.­vre.­cse.­hcmut.­edu.­vn*](http://www.vre.cse.hcmut.edu.vn/)