ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA FAST

BÁO CÁO MÔN HỌC

CHUYÊN ĐỀ MẠNG

**ĐỀ TÀI :**

WEBSITE QUẢN LÝ HỒ SƠ BỆNH ÁN

SINH VIÊN : Trần Hữu Sơn

MÃ SINH VIÊN : 122170046

LỚP : 17PFIEV2

CBHD : Nguyễn Tấn Khôi

Đà Nẵng, 04/2021

MỤC LỤC

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 3](#_Toc69154669)

[1.1. CƠ SỞ LÝ THUYẾT 3](#_Toc69154670)

[1.2. PHÁT BIỂU BÀI TOÁN 6](#_Toc69154671)

[1.3. KẾT CHƯƠNG 6](#_Toc69154672)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 7](#_Toc69154673)

[2.1. PHÂN TÍCH YÊU CẦU 7](#_Toc69154674)

[2.2. PHÂN TÍCH CHỨC NĂNG 7](#_Toc69154675)

[2.2.1. Đối tượng sử dụng 7](#_Toc69154676)

[2.2.2. Xem thông tin hồ sơ 7](#_Toc69154677)

[2.2.3. Quản lí thông tin hồ sơ 7](#_Toc69154678)

[2.2.4. Tìm kiếm thông tin hồ sơ. 7](#_Toc69154679)

[2.2.5. Công nghệ sử dụng 8](#_Toc69154680)

[2.3. THIẾT KẾ CƠ SỞ DỮ LIỆU 8](#_Toc69154681)

[2.4. TỔ CHỨC CHƯƠNG TRÌNH 8](#_Toc69154682)

[2.4.1. Tổ chức thư mục 8](#_Toc69154683)

[2.4.2. Tập tin public 9](#_Toc69154684)

[2.4.3. Tập tin routes 9](#_Toc69154685)

[2.4.4. Tập tin views 9](#_Toc69154686)

[2.5. KẾT CHƯƠNG 10](#_Toc69154687)

[CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ 11](#_Toc69154688)

[3.1. MÔ HÌNH TRIỂN KHAI 11](#_Toc69154689)

[3.1.1. Mô hình triển khai 11](#_Toc69154690)

[3.1.2. Các công cụ sử dụng 12](#_Toc69154691)

[3.2. KẾT QUẢ THỰC NGHIỆM 12](#_Toc69154692)

[3.2.1. Kịch bản 1 – Đăng nhập hệ thống 12](#_Toc69154693)

[3.2.2. Kịch bản 2 – Chức năng xem/sửa dữ liệu 13](#_Toc69154694)

[3.2.3. Kịch bản 3 – Chức năng tìm kiếm 14](#_Toc69154695)

[3.2.4. Kịch bản 4 – Chức năng thêm sản phẩm 15](#_Toc69154696)

[3.3. NHẬN XÉT ĐÁNH GIÁ KẾT QUẢ 15](#_Toc69154697)

[3.4. KẾT CHƯƠNG 16](#_Toc69154698)

**DANH SÁCH HÌNH ẢNH**

[*Hình 3.1 Giao diện trang chủ* 11](#_Toc69284737)

[*Hình 3.2 Giao diện chức năng thêm hồ sơ* 11](#_Toc69284738)

[*Hình 3.3 Giao diện “Chỉnh sửa hồ sơ”* 12](#_Toc69284739)

[*Hình 3.4 Giao diện đăng nhập WebSite* 13](#_Toc69284740)

[*Hình 3.5 Thông tin bệnh án* 14](#_Toc69284741)

[*Hình 3.6 Thông tin sản phẩm khi chưa chỉnh sửa* 14](#_Toc69284742)

[*Hình 3.7 Thông tin sản phẩm sau khi chỉnh sửa* 14](#_Toc69284743)

[*Hình 3.8 Giao diện trang chủ khi chưa xoá bệnh án* 15](#_Toc69284744)

[*Hình 3.9 Giao diện trang chủ sau khi xoá bệnh án* 15](#_Toc69284745)

[*Hình 3.10 Ô tìm kiếm trên Website* 15](#_Toc69284746)

[*Hình 3.11 Kết quả tìm kiếm* 16](#_Toc69284747)

[*Hình 3.12 Giao diện trang chủ khi chưa thêm hồ sơ* 16](#_Toc69284748)

[*Hình 3.13 Giao diện chức năng thêm hồ sơ* 17](#_Toc69284749)

DANH SÁCH TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Diễn giải** |
| CSDL | Cơ sở dữ liệu |
| Req | Request |
| Res | Response |
| npm | Node package manager |
| Ejs | Embedded JavaScript templating |
| Div | Division |

MỞ ĐẦU

**1.**  **Tổng quan về đề tài**

Công nghệ thông tin (CNTT) đang dần chứng tỏ tầm ảnh hưởng rất lớn đến mọi mặt của đời sống xã hội. Đối với hoạt động của ngành y tế, có thể thấy rằng, CNTT ngày càng đóng vai trò quan trọng, không chỉ “bà đỡ” cho quá trình cải cách hành chính trong công tác quản lý, điều hành của cơ quan quản lý mà còn “đỡ đầu” cho việc triển khai và ứng dụng thành công các kỹ thuật cao trong công tác khám chữa bệnh.

Cùng với sự phát triển của công nghệ nói chung, sự hoàn thiện cơ sở pháp lý về ứng dụng công nghệ thông tin trong đời sống xã hội, để đẩy mạnh ứng dụng công nghệ thông tin trong ngành y tế, góp phần hiện đại hóa công tác bảo vệ, chăm sóc sức khỏe nhân dân. Vì vậy, em đã thực hiện đề tài “ XÂY DỰNG WEBSITE QUẢN LÍ HỒ SƠ BỆNH ÁN ” cho các đơn vị y tế khám chữa bệnh, góp phần quản lí hiệu quả thông tin người bệnh.

**2. Mục đích và ý nghĩa của đề tài**

***2.1. Mục đích***

Xây dựng website quản lý quản lý hồ sơ bệnh án giúp người bệnh không phải lưu trữ tất cả loại giấy gờ khi đi khám, chữa bệnh; Tránh được các chỉ định cận lâm sàng (siêu âm, xét nghiệm…) trùng lặp. Đồng thời, giúp bác sĩ dễ dàng tìm lại hồ sơ bệnh án của bệnh nhân,  giảm thời gian thăm khám, hỗ trợ điều trị kịp thời, giúp nâng cao chất lượng chẩn đoán và điều trị; Cung cấp thông tin, dữ liệu lâm sàng nhanh chóng, kịp thời cho các cơ sở cung cấp dịch vụ chăm sóc sức khỏe khác.; Khi thông tin về khám chữa bệnh của người bệnh thông suốt, minh bạch, giúp cho việc quản lý chi phí khám, chữa bệnh bảo hiểm y tế dễ dàng hơn, góp phần hạn chế việc lạm dụng thuốc và xét nghiệm nếu có.

***2.2. Ý nghĩa***

* Triển khai bệnh án điện tử giúp việc cập nhật thông tin sức khỏe cá nhân trong hồ sơ sức khỏe điện tử được dễ dàng, thuận lợi hơn.
* Các bác sĩ và bệnh nhân có thể truy cập hồ sơ bệnh án điện tử từ bất kỳ nơi nào trên thế giới chỉ cần có đường truyền internet.

**3. Phương pháp thực hiện**

**3.1 Bố cục của đồ án**

Đồ án bao gồm các nội dung sau:

Mở đầu

Chương 1: trình bày cơ sở lý thuyết.

Chương 2: phân tích thiết kế hệ thống.

Chương 3: triển khai và đánh giá kết quả.

Kết luận và hướng phát triển.

# CƠ SỞ LÝ THUYẾT

## CƠ SỞ LÝ THUYẾT

* + 1. **NodeJS**

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới. NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ WIndow cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

* + 1. **Các đặc tính của NodeJS**
* Không đồng bộ: Tất cả các API của NodeJS đều không đồng bộ (none-blocking), nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).
* Chạy rất nhanh: NodeJ được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
* Đơn luồng nhưng khả năng mở rộng cao: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lặp. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo đề hạn chế để xử lý yêu cầu. Node.js sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.
* Không đệm:  
  NodeJS không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.
* Có giấy phép: NodeJS đã được cấp giấy phép bởi MIT License.
  + 1. **Cách làm việc của NodeJS**

Ý tưởng chính của Node js là sử dụng non-blocking, hướng sự vào ra dữ liệu thông qua các tác vụ thời gian thực một cách nhanh chóng. Bởi vì, Node js có khả năng mở rộng nhanh chóng, khả năng xử lý một số lượng lớn các kết nối đồng thời bằng thông lượng cao. Nếu như các ứng dụng web truyền thống, các request tạo ra một luồng xử lý yêu cầu mới và chiếm RAM của hệ thống thì việc tài nguyên của hệ thống sẽ được sử dụng không hiệu quả. Chính vì lẽ đó giải pháp mà Node js đưa ra là sử dụng luồng đơn (Single-Threaded), kết hợp với non-blocking I/O để thực thi các request, cho phép hỗ trợ hàng chục ngàn kết nối đồng thời

* + 1. **Cài đặt các thư viện**
* ***npm init***: Câu lệnh trên đơn giản là sẽ tạo ra 1 file có tên là package.json – thành phần này được gọi là Local Package Database, lưu trữ thông tin (tên package, phiên bản, các dependencies) mà project của bạn sử dụng.
* ***npm install express***: Express là một framework giành cho nodejs. Nó cung cấp nhiều tính năng mạnh mẽ trên nền tảng web cũng như trên các ứng dụng di động. Express hỗ rợ các phương thức HTTP và midleware tạo ra môt API vô cùng mạnh mẽ và dễ sử dụng.
* ***express*** : dùng để tạo định tuyến xử lý và xử lý các yêu cầu từ máy khách.
* ***express-fileupload*** : Phần mềm trung gian tải lên tệp nhanh đơn giản bao bọc xung quanh busboy.
* ***body-parser*** : dùng để phân tích cú pháp yêu cầu đến từ máy khách.
* ***mysql*** : Trình điều khiển Node JS cho MySQL.
* ***ejs*** : công cụ tạo khuôn mẫu để hiển thị các trang html cho ứng dụng.
* ***req-flash*** : được sử dụng để gửi tin nhắn flash đến chế độ xem
* ***nodemon*** : Được cài đặt trên toàn cầu. Nó được sử dụng để theo dõi các thay đổi đối với tệp và tự động khởi động lại máy chủ.
  + 1. **HTML/CSS**

HTML là viết tắt của Hypertext Markup Language. Là ngôn ngữ giúp người dùng tạo ra và cấu trúc các thành phần của trang web, phân chia trang web thành các phần nhỏ dễ quản lí như header, body, footer,…

CSS là viết tắt của Cascading Style Sheet language là ngôn ngữ tạo phong cách cho trang web, cùng với HTML,CSS giúp chúng ta thay đổi phong cách của các đoạn văn, bảng, tiêu đề.. được viết bằng HTML như thay đổi bố cục, màu sắc, font chữ, cỡ font, thay đổi cấu trúc, .. giúp cho trang web trở nên bắt mắt và thân thiện với người dùng hơn.

Mối tương quan giữa HTML và CSS rất mật thiết.HTML là ngôn ngữ markup, và CSS định hình phong cách hay tất cả những gì tạo nên giao diện website, chúng là không thể tách rời.

* + 1. **Boostrap**

Boostrap là một framework bao gồm các HTML,CSS, và JavaScript template dùng để phát triển website theo chuẩn responsive – một phong cách thiết kế trang web phù hợp với các thiết bị khác nhau và các độ phân giải màn hình khác nhau.

* + 1. **Node.js Module**

Node.js Module là một thư viện JavaScript, là một tập hợp các hàm (function), đối tượng và các biến,.. mà bạn có thể đưa vào ứng dụng của bạn để sử dụng và điều này đơn giản hơn việc viết code rất nhiều.Thông thường mooic module sẽ được viết trong một tập tin riêng rẽ.

* + 1. **NPM**

NPM hay node package manager là một công cụ tạo và quản lí các thư viện lập trình JavaScript cho Node.js. NPM cung cấp khoảng 800.000 thư viện dựng sẵn cũng như các công cụ cho lạp tình viên và tăng tốc quá trình phát triển ứng dụng

* + 1. **JSON**

Cơ sở dữ liệu (Database) là một tập hợp có tổ chức các thông tin có cấu trúc hoặc dữ liệu, thường được lưu trữ trực tuyến trong một hệ thống máy tính. Một cơ sở dữ liệu thường được kiểm soát bởi hệ thống quản lí cơ sở dữ liệu (DBMS).Dữ liệu trong các loại cơ sở dữ liệu phổ biến nhất hiện nay thường được mô hình hóa theo hàng và cột trong một loạt các bảng để giúp xử lí và truy vấn hiệu quả.Dữ liệu sau đó có thể dễ dàng truy cập , quản lí, sửa đổi cập nhật, kiểm soát và tổ chức, ..

Một số cơ sở dữ liệu phổ biến hiện nay như MySQL, MongoDB, Oracle, CouchDB, ..

* + 1. **Cơ Sở Dữ Liệu**

Cơ sở dữ liệu (Database) là một tập hợp có tổ chức các thông tin có cấu trúc hoặc dữ liệu, thường được lưu trữ trực tuyến trong một hệ thống máy tính. Một cơ sở dữ liệu thường được kiểm soát bởi hệ thống quản lí cơ sở dữ liệu (DBMS).Dữ liệu trong các loại cơ sở dữ liệu phổ biến nhất hiện nay thường được mô hình hóa theo hàng và cột trong một loạt các bảng để giúp xử lí và truy vấn hiệu quả.Dữ liệu sau đó có thể dễ dàng truy cập , quản lí, sửa đổi cập nhật, kiểm soát và tổ chức, ..

Một số cơ sở dữ liệu phổ biến hiện nay như MySQL, MongoDB, Oracle, CouchDB, ..

* + 1. **XAMPP**

XAMPP là một loại ứng dụng phần mềm khá phổ biến thường được sử dụng để xây dựng và phát triển các dự án website theo ngôn ngữ PHP, XAMPP thường được sử dụng để tạo máy chủ web được tích hợp sẵn Apache, PHP, MySQL, FTP Sever, Mail Server và các công cụ như phpMyAdmin

* + 1. **Các thẻ HTML**
* Thẻ ***<div>*** : Sử dụng để tạo ra một khu vực kiểu block nào đó trên website hay có thể hiểu là gom nhóm tập hợp các phần tử trên website vào một khu vực với thẻ <div>.
* Thẻ ***<table>*** : Được sử dụng để hiển thị dữ liệu ở dạng bảng (hàng \* cột). Có thể có nhiều cột trong một hàng.

Các thẻ định nghĩa bảng trong HTML

|  |  |
| --- | --- |
| **Thẻ** | **Mô tả** |
| <table> | Định nghĩa bảng. |
| <tr> | Định nghĩa một hàng trong một bảng. |
| <th> | Định nghĩa phần header (dòng đầu tiên) của bảng. |
| <td> | Định nghĩa 1 ô của bảng. |
| <caption> | Định nghĩa phụ đề bảng. |
| <colgroup> | Chỉ định một nhóm của một hoặc nhiều cột trong một bảng để định dạng. |
| <col> | Được sử dụng với thẻ <colgroup> để chỉ định thuộc tính cột cho mỗi cột. |
| <tbody> | Được sử dụng để nhóm nội dung body trong một bảng. |
| <thead> | Được sử dụng để nhóm nội dung header trong một bảng. |
| <tfooter> | Được sử dụng để nhóm nội dung footer trong một bảng. |

* Thuộc tính ***class*** : dùng để xác định một hoặc nhiều tên lớp cho phần tử HTML. Tên lớp có thể dùng trong CSS hoặc JavaScript để thực hiện một số tác vụ nhất định cho phần tử có tên lớp đó.
* HTML ***<form>*** : Là một phần của trang web có chứa các điều khiển như các trường văn bản, các trường mật khẩu, các checkbox, các radio button, submit button, các menu... là phương tiện cho người dùng nhập dữ liệu được gửi đến máy chủ để xử lý.
  + ***action*** là thuộc tính chỉ ra địa chỉ URL mà dữ liệu của form gửi đến. Thuộc tính ***method*** có giá trị bằng get hoặc post.
  + Sử dụng ***method="get"*** thì khi submit dữ liệu được biểu diễn thông qua URL. Sử dụng ***method="post"*** thì khi submit biểu diễn trong nội dung của Request gửi đến Server và là ẩn với người dùng, sử dụng post an toàn hơn.
  + Các phần tử trong Form:
    - Thẻ ***<input>*** : thẻ này có thể hiện thị giao diện nhập liệu nhiều cách tùy vào thuộc tính type.
    - Thẻ ***<label>*** : thẻ label để cho biết tiêu đề của một thẻ input
* Tag ***<select>*** : được dùng để tạo một danh sách chọn lựa (danh sách thả xuống).
* Tag ***<option>*** : bên trong thành phần select, định nghĩa giá trị tùy chọn trong danh sách.
* The ***<button>*** : được sử dụng để tạo các nút có thể click bên trong form trên trang web.
  + Loại nút:
    - ***submit***: Nút gửi dữ liệu biểu mẫu đến máy chủ. Đây là mặc định nếu thuộc tính không được chỉ định hoặc nếu thuộc tính được thay đổi động thành một giá trị trống hoặc không hợp lệ.
    - ***reset***: Nút đặt lại tất cả các điều khiển về giá trị ban đầu của chúng.
    - ***button***: Nút không có hành vi mặc định. Nó có thể có các tập lệnh phía máy khách được liên kết với các sự kiện của phần tử, được kích hoạt khi các sự kiện xảy ra.

## PHÁT BIỂU BÀI TOÁN

Xây dựng WebSite quản lý. SV lựa chọn một trong các bài toán quản lý : Quản lý sinh viên, Quản lý vật tư, Quản lý bệnh nhân khám bệnh, Quản lý của hàng thiết bị điện, Quản lý cửa hàng thuốc, …

## KẾT CHƯƠNG

Chương này trình bày các nội dung cơ sở lý thuyết chính liên quan đến nội dung của đồ án. Nội dung cơ sở lý thuyết này sẽ được sử dụng trong phần phân tích và triển khai chương trình. Phần phát biểu bài toán sẽ mô tả nội dung và các vấn đề đặt ra. Ngoài ra cũng liệt kê và đánh giá các giải pháp đã có.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## PHÂN TÍCH YÊU CẦU

* Lập trình Node.js và khai thác các module của Node.js
* Kết nối CSDL MySQL của xampp
* Xây dựng các table CSDL
* Lập trình chức năng login vào Website theo tài khoản trong CSDL
* Giao diện trang chủ (Home) có hỉnh ảnh và link đến các trang khác
* Giao diện nhập dữ liệu
* Giao diện thống kê, tìm kiếm, liệt kê dữ liệu
* Các chức năng khác …

## PHÂN TÍCH CHỨC NĂNG

### Đối tượng sử dụng

Dành cho những bác sĩ, điều dương : người có nhiệm vụ khám chữa bệnh cho bệnh nhân. Người này được cấp username và password để đăng nhập hệ thống thực hiện chức năng của mình :

* Chức năng cập nhật, sửa, xoá các hồ sơ…

### Xem thông tin hồ sơ

* Cho phép xem các thông tin liên quan đến hồ sơ bệnh án: ngày nhập viện, tình trạng sức khỏe bệnh nhân, thông tin các nhân, khoa điều trị

### Quản lí thông tin hồ sơ bệnh án

* Người quản trị có thể cập nhật, thêm, sửa, xoá các thông tin các hồ sơ bệnh án

### Tìm kiếm thông tin hồ sơ bệnh án

Đây là chức năng dành cho mọi người dùng. Cả người dùng đơn thuần lẫn người QL đều không cần phải đăng nhập khi sử dụng chức năng này. Khi họ muốn tìm kiếm một bệnh nhân nào đó với đầy đủ thông tin hoặc chỉ một vài thông tin hệ thống sẽ liệt kê ra các bệnh với đầy đủ thuộc tính phù hợp với thông tin mà người dùng đã nhập vào và các thông tin chi tiết kèm theo. Người dung có thể tìm kiếm đơn thuộc tính và đa thuộc tính.

## THIẾT KẾ CƠ SỞ DỮ LIỆU

* Bệnh nhân

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Key** |
| 1 | **ID** | int(5) | No | Primary key |
| 2 | **Hovaten** | varchar(225) | No |  |
| 3 | **Tuoi** | varchar(225) | No |  |
| 4 | **Gioitinh** | varchar(225) | No |  |
| 5 | **Ngaynhapvien** | int(11) | No |  |
| 6 | **Khoa** | varchar(50) | No |  |
| 7 | **Image** | Varchar(255) | No |  |
| 8 | **Ghichu** | Varchar(255) | No |  |

*Bảng 2.3.1 Thông tin sản phẩm*

* User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Key** |
| 1 | **username** | varchar(20) | No | Primary key |
| 2 | **password** | varchar(30) | No |  |
| 3 | **email** | varchar(20) | Yes |  |
| 4 | **phonenumber** | varchar(20) | Yes |  |

*Bảng 2.3.1 Thông tin người dùng*

## TỔ CHỨC CHƯƠNG TRÌNH

### Tổ chức thư mục

**|---- GK-SonTran**

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

**|---- public**

**|---- assets**

**|---- img**

**|---- routes**

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

**|---- player.js**

**|---- login.js**

**|---- logout.js**

**|---- validate**

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

**|---- views**

**|---- partials**

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

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

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

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

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

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

### 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.

### 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
* …

## KẾT CHƯƠNG

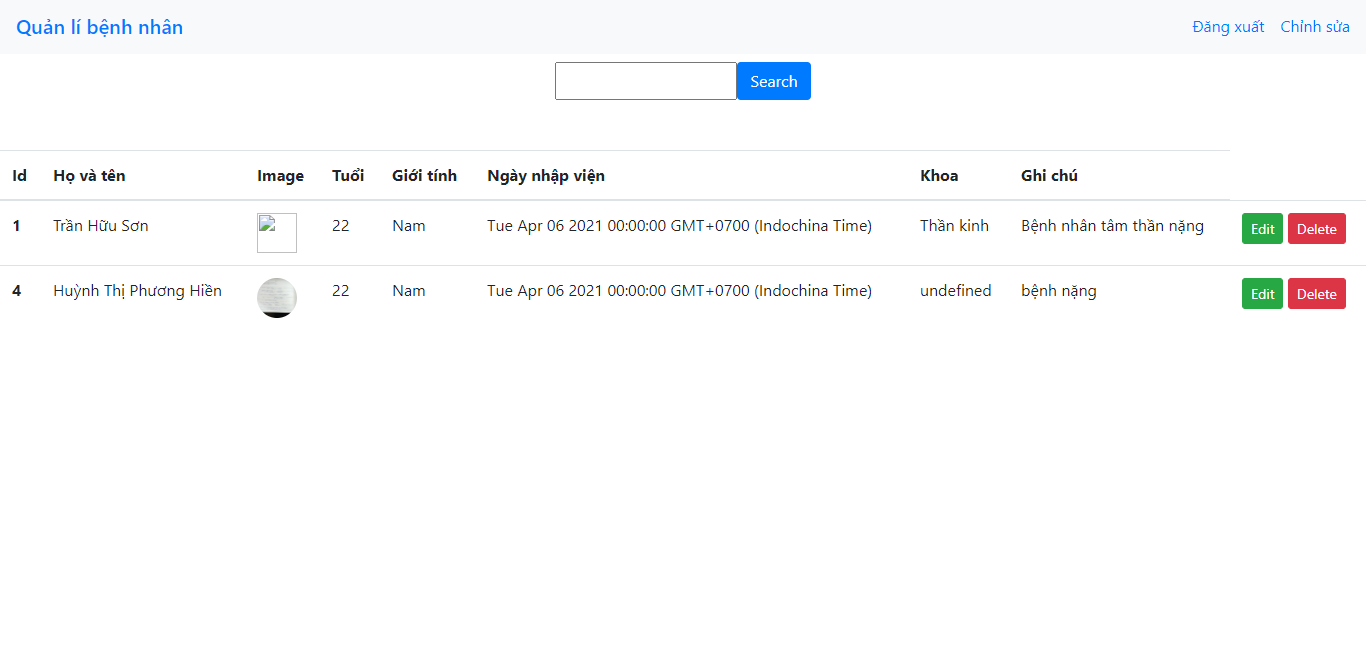
Chương này trình bày về phân tích và triển khai hệ thống, bao gồm phân tích hiện trạng, phân tích yêu cầu của bài toán.

# TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

## MÔ HÌNH TRIỂN KHAI

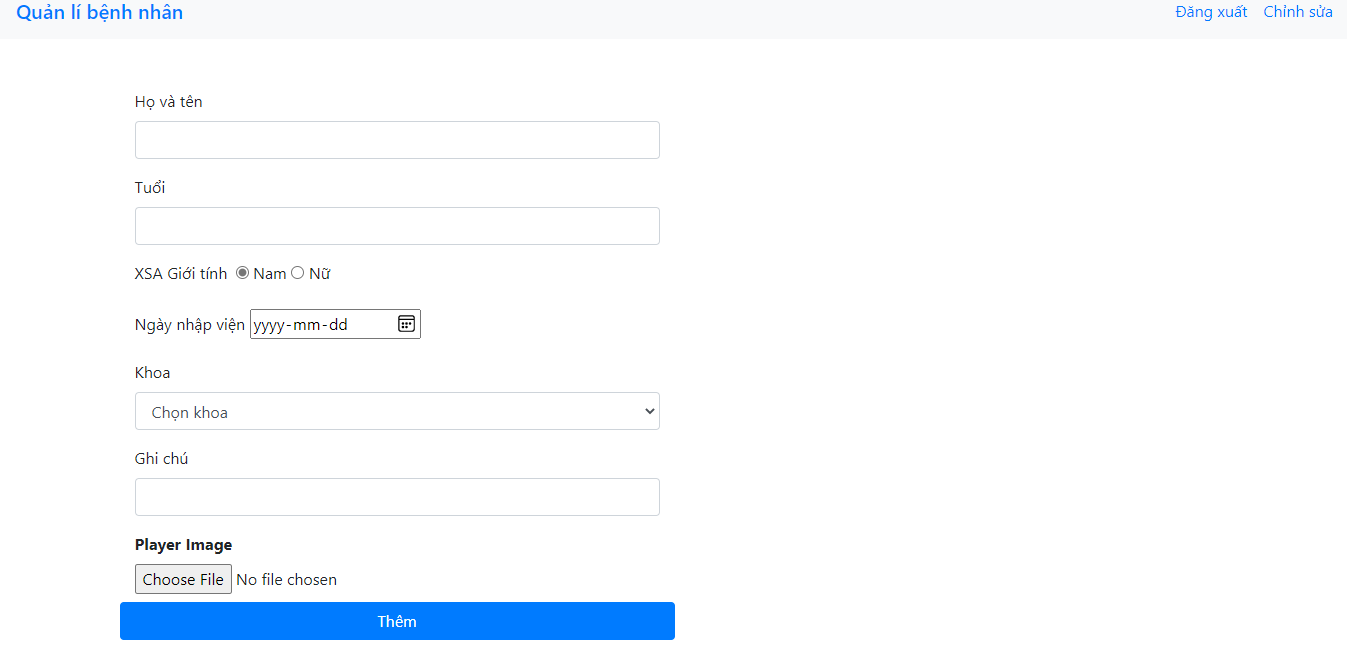
### Mô hình triển khai

* Giao diện trang chủ:



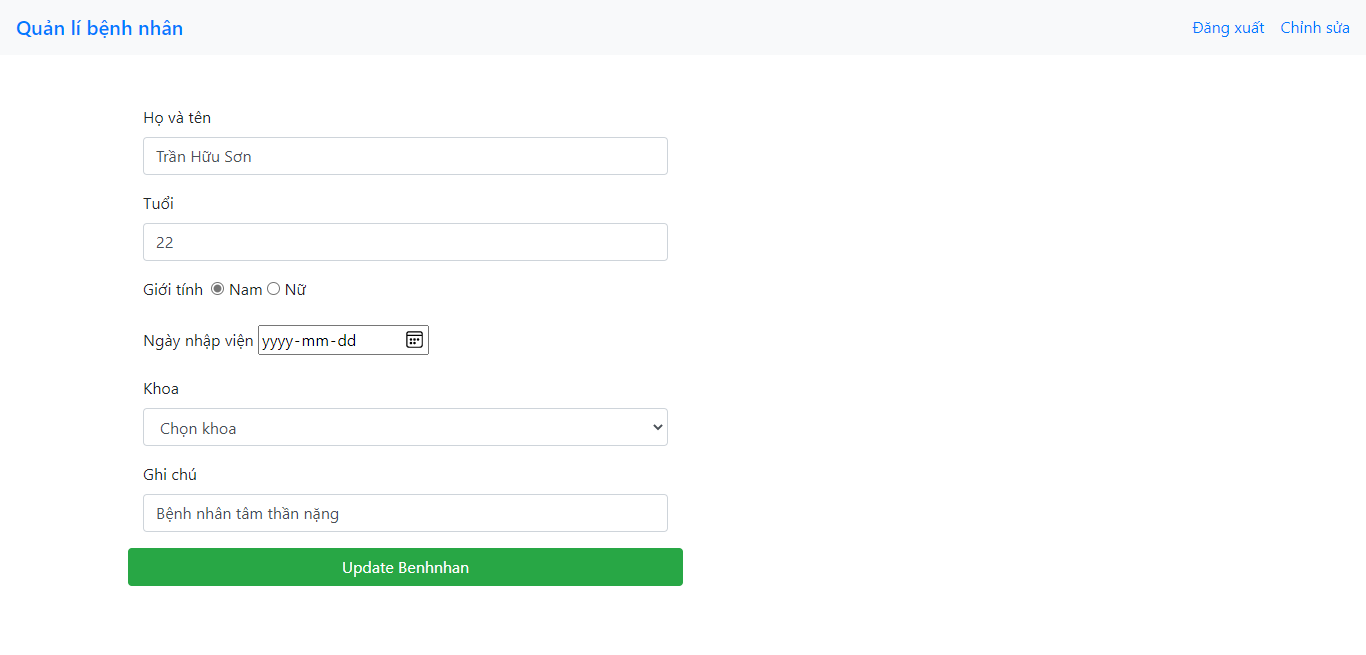
*Hình 3.1 Giao diện trang chủ*

* Thêm hồ sơ bệnh án



*Hình 3.2 Giao diện chức năng thêm hồ sơ*

* Chỉnh sửa hồ sơ



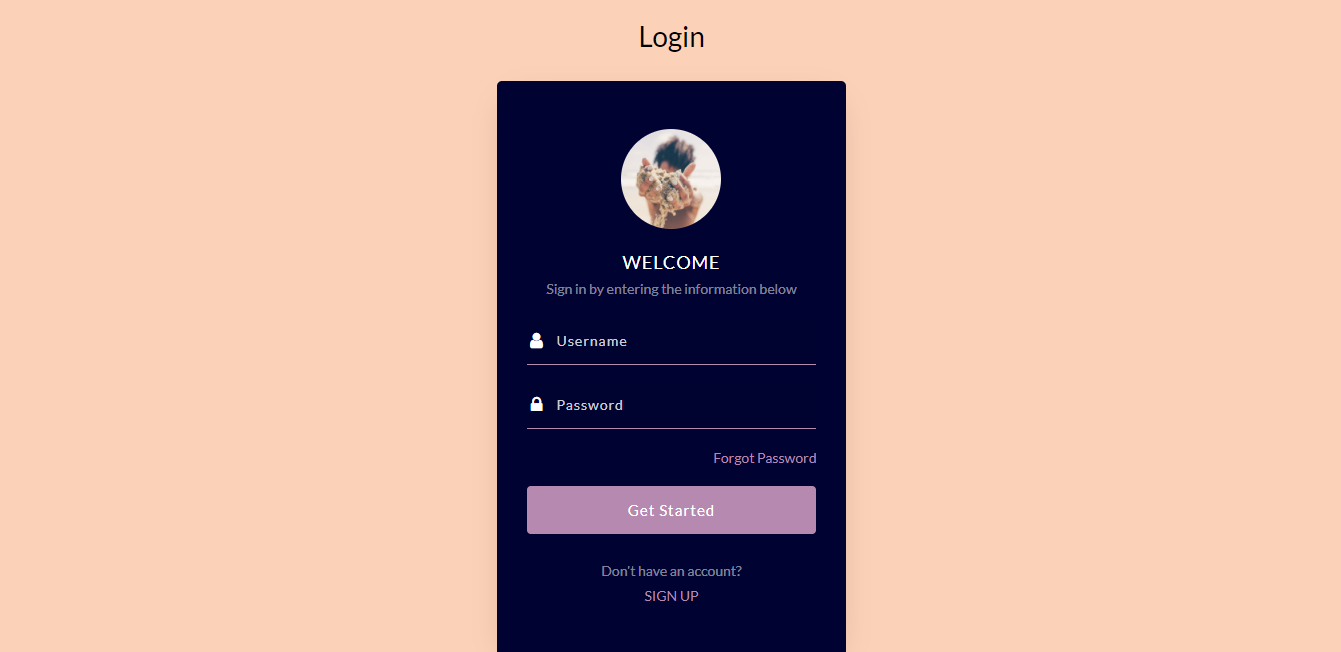
*Hình 3.3 Giao diện “Chỉnh sửa hồ sơ”*

### Các công cụ sử dụng

WebSite sử dụng công cụ quản lý dữ liệu MySQL trên xampp

## KẾT QUẢ THỰC NGHIỆM

### Kịch bản 1 – Đăng nhập hệ thống



*Hình 3.4 Giao diện đăng nhập WebSite*

Người dùng đăng nhập WebSite với tài khoản mặc định được cấp sẵn:

Username: trung

Password: 123

* TH1: Người dùng đăng nhập đúng:

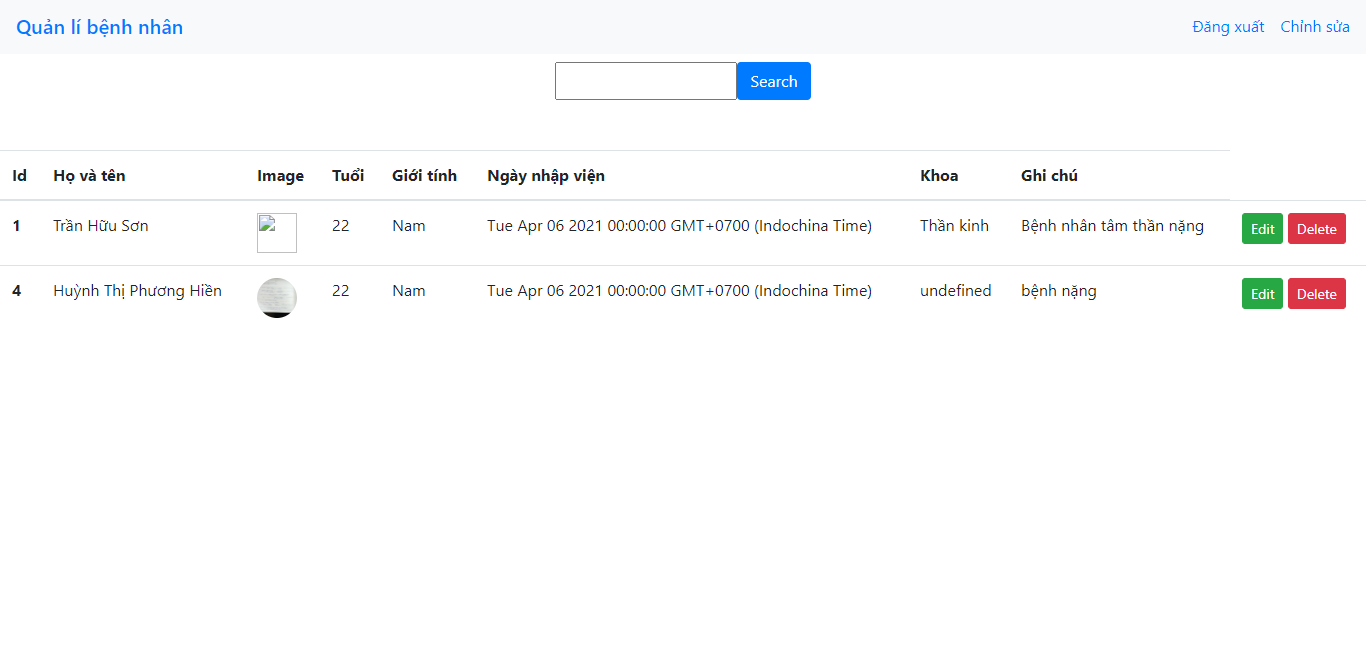
Website sẽ chuyển đến trang chủ hiển thị thông tin sản phẩm.

* TH2: Người dùng nhập sai username hay password:

Website sẽ không chuyển đến trang chủ cho đến khi người dùng đăng nhập đúng cú pháp.

### Kịch bản 2 – Chức năng xem/sửa dữ liệu

* Ở giao diện này người dùng có thể xem thông tin hồ sơ bệnh án



*Hình 3.5 Thông tin bệnh án*

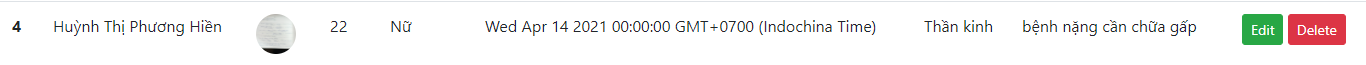
* Người dùng có thể thực hiện thao tác sửa/xoá thông tin bệnh án
* Người dùng click nút  để đi đến giao diện chỉnh sửa thông tin bệnh án

Ví dụ: Lúc chưa chỉnh sửa



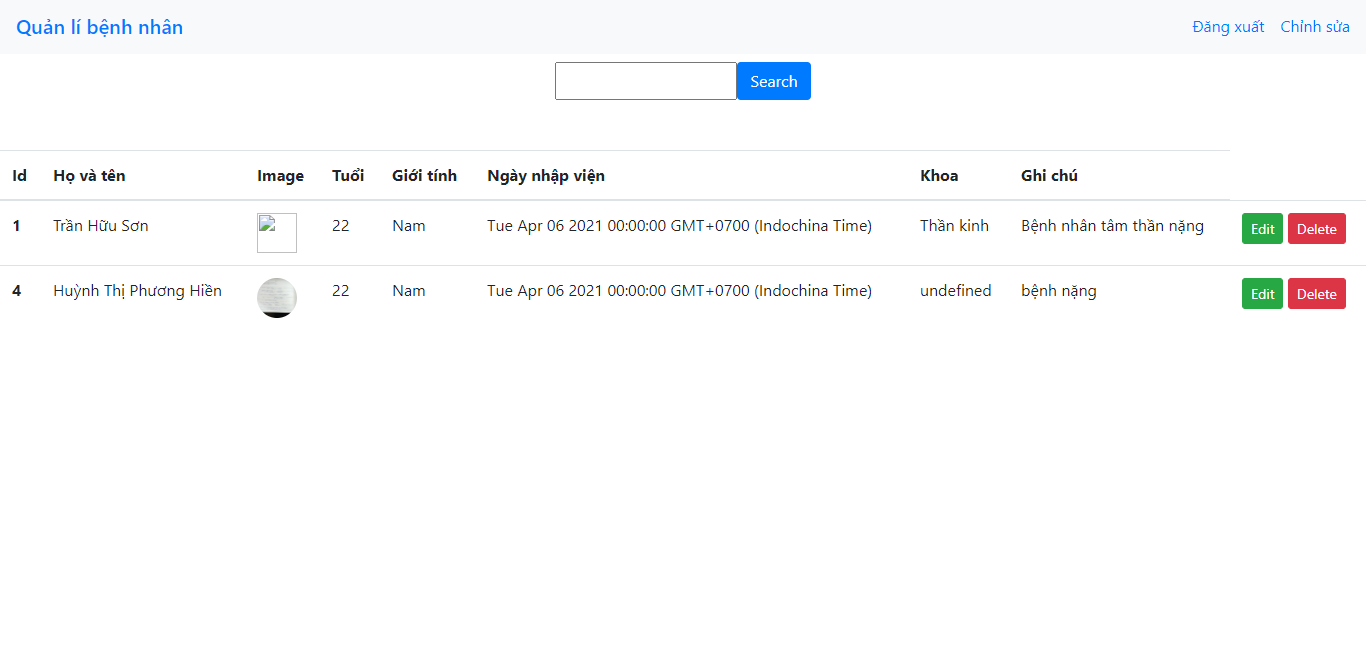
*Hình 3.6 Thông tin sản phẩm khi chưa chỉnh sửa*

Sau khi chỉnh sửa



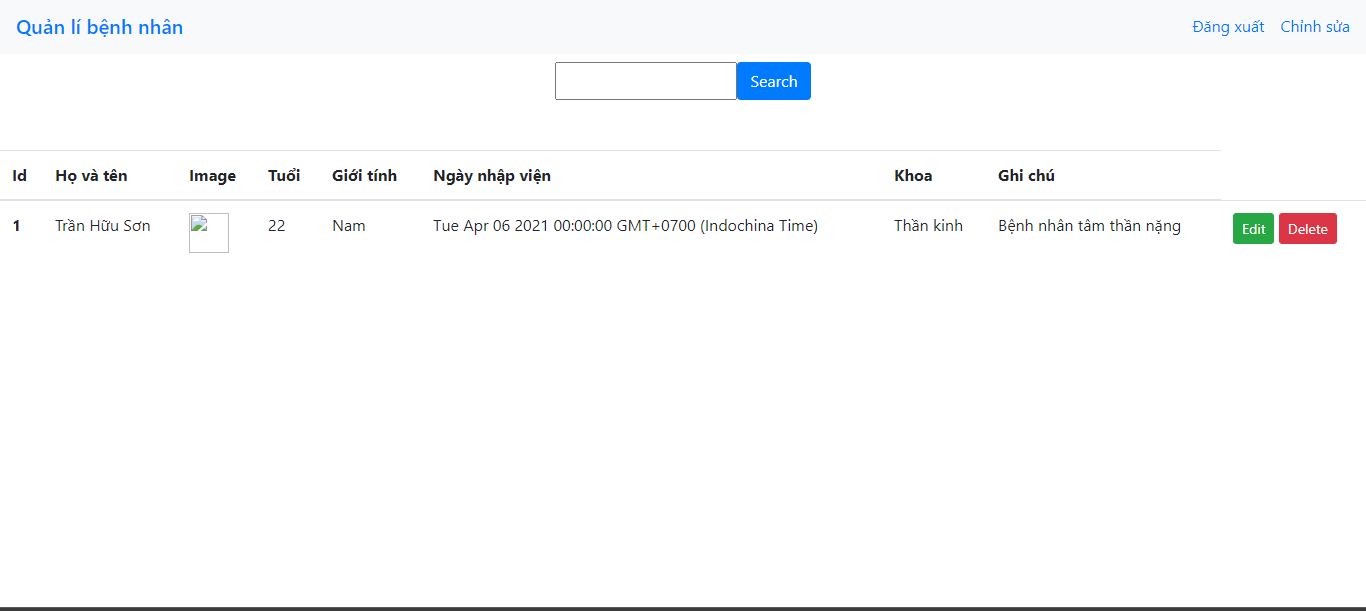
*Hình 3.7 Thông tin sản phẩm sau khi chỉnh sửa*

* + - **Ghi chú:** bệnh nặng -> bệnh nặng cần chữa gấp
    - **Khoa**: underfined -> Thần kinh
    - **Giới tính**: Nam -> Nữ
  + Người dùng click nút  để xoá sản phẩm



*Hình 3.8 Giao diện trang chủ khi chưa xoá bệnh án*

* Sau khi xoá:

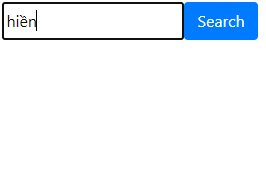


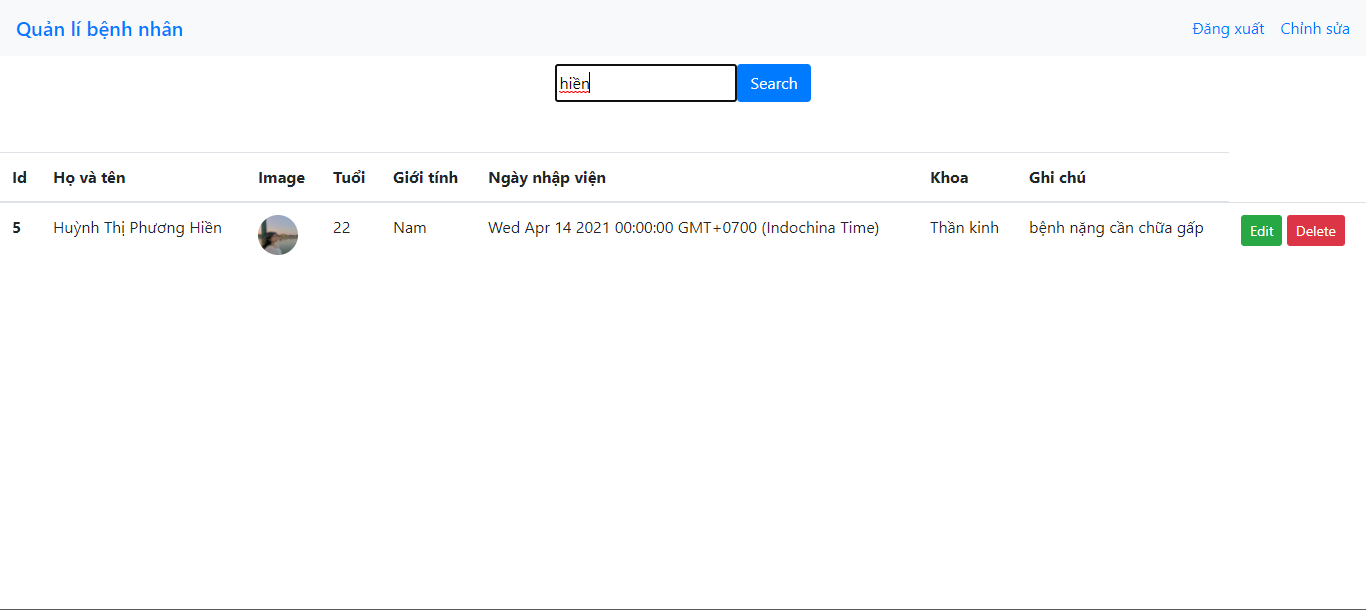
*Hình 3.9 Giao diện trang chủ sau khi xoá bệnh án*

### Kịch bản 3 – Chức năng tìm kiếm



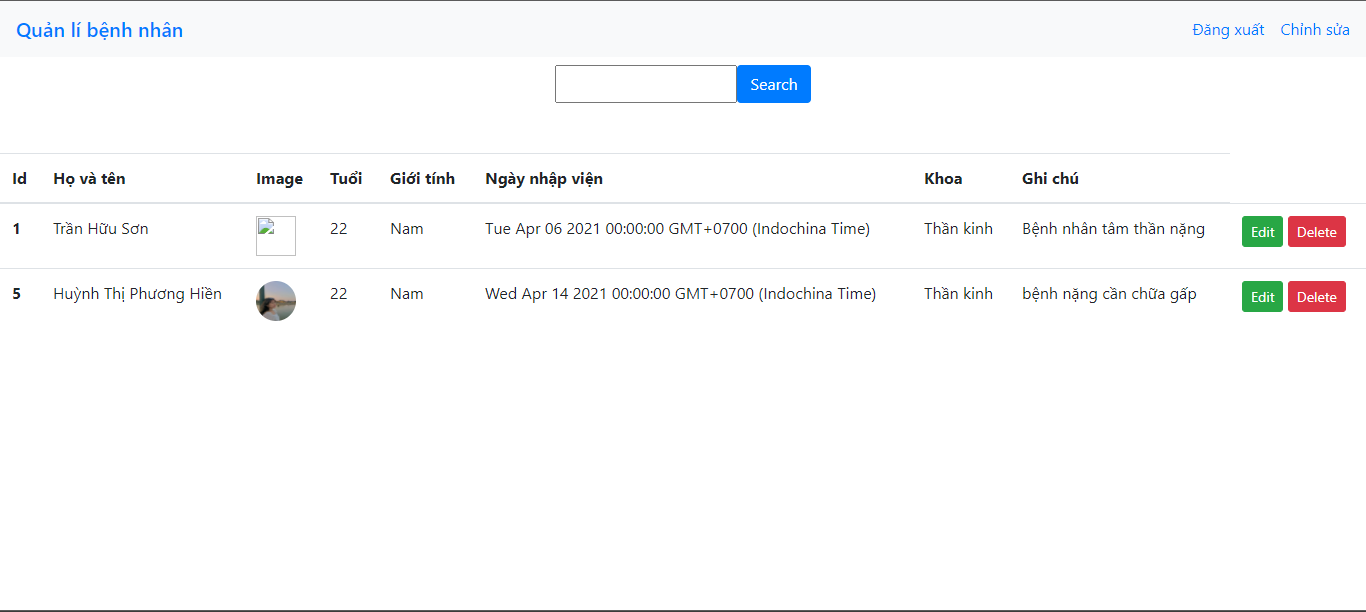
*Hình 3.10 Ô tìm kiếm trên Website*

* Tìm kiếm  ta được kết quả:



*Hình 3.11 Kết quả tìm kiếm*

### Kịch bản 4 – Chức năng thêm hồ sơ bệnh án



*Hình 3.12 Giao diện trang chủ khi chưa thêm hồ sơ*

Sau khi thêm hồ sơ với các giá trị:

Họ và tên: Nguyễn Cao Văn Trung

Tuổi: 22

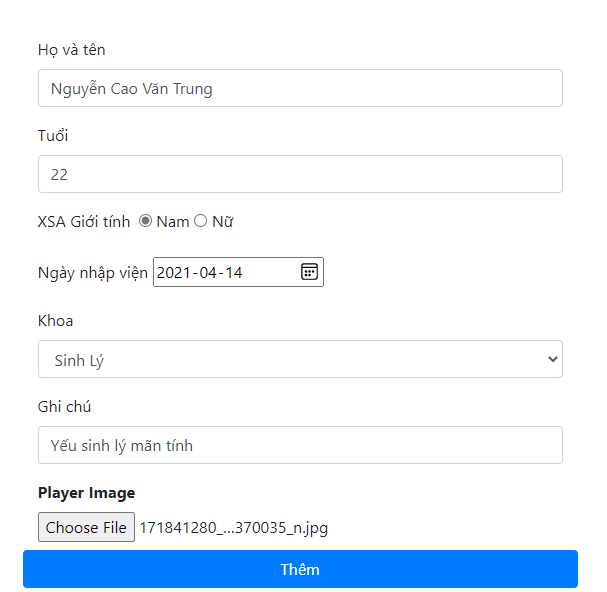
Ngày nhập viện: Chọn “Today”

Giới tính: Nam

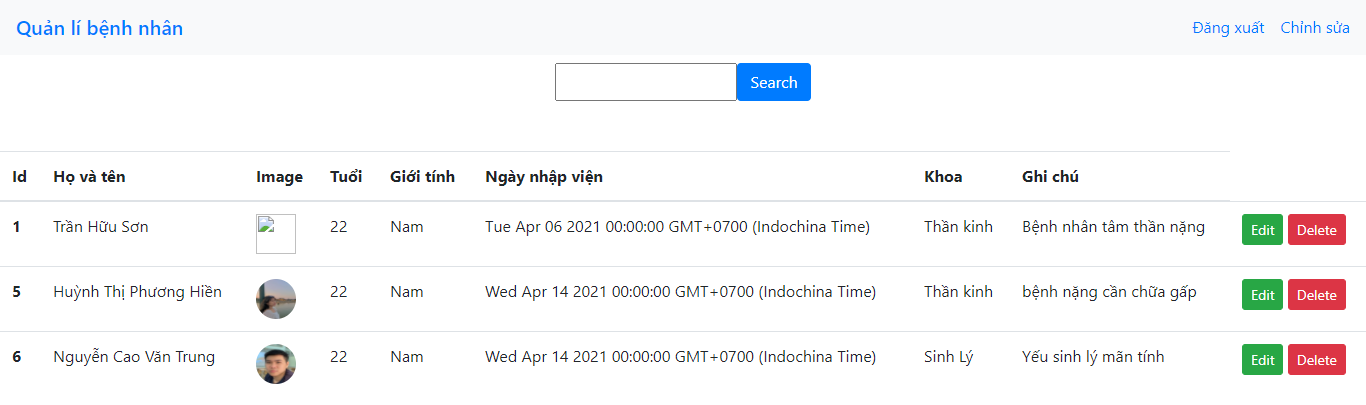
Khoa: Sinh lý

Ghi chú: yếu sinh lý mãn tính

Chọn ảnh:



*Hình 3.13 Giao diện chức năng thêm hồ sơ*



Hình 3.14 Giao diện sau khi thêm hồ sơ

## NHẬN XÉT ĐÁNH GIÁ KẾT QUẢ

Qua kết quả thực nghiệm, ta có những nhận xét như sau:

* Ưu điểm:

+ Cơ bản nắm được các bước xây dựng một website quản lý hồ sơ bệnh án trực tuyến, thực hiện đúng quy trình.

+ Xây dựng được bố cục trang web hợp lí, bước đầu thực hiện được nghiệp vụ của hệ thống.

* Hạn chế:

+ WebSite nhỏ, mang tính chất mô phỏng.

+ Cơ sở dữ liệu nhỏ.

+ Chức năng chưa tối ưu.

## KẾT CHƯƠNG

Chương này trình bày về kết quả triển khai hệ thống, cấu hình hệ thống và các thành phần chức năng. Kết quả được đánh giá thông qua các kịch bản thực nghiệm khác nhau nhằm thể hiện các ưu/nhược của giải pháp đề xuất.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. **KẾT QUẢ ĐẠT ĐƯỢC**

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

*Về mặt lý thuyết,* đồ án đã đạt được những yêu cầu cơ bản về lý thuyết cần có để xây dựng một WebSite.

Về mặt thực tiễn ứng dụng, đồ án đã đạt được cơ bản các chức năng đề ra của một Website quản lý hàng hoá.

Kết quả đóng góp của đồ án được thể hiện như sau:

* Tìm hiểu hệ thống kỹ càng, chính xác. Phân tích rõ được:
  + Yêu cầu bài toán đặt ra.
  + Các chức năng cần có của hệ thống.
  + Ưu/ nhược điểm của hệ thống.
* Thiết kế các chức năng của hệ thống đáp ứng yêu cầu đặt ra.
* Giao diện thân thiện, dễ sử dụng.

1. **KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN**

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

* Xây dựng Website sử dụng được đáp ứng đầy đủ yêu cầu nghiệp vụ của hệ thống
* Bổ sung và hoàn thiện một số chức năng của hệ thống.
* Đánh giá hiệu năng trên các môi trường khác nhau.
* Kiểm thử các chức năng của chương trình.
* Bổ sung các giải pháp bảo mật và an toàn cho hệ thống.

TÀI LIỆU THAM KHẢO

**Internet**

1. [*https://dev.to/achowba/build-a-simple-app-using-node-js-and-mysql-19me*](https://dev.to/achowba/build-a-simple-app-using-node-js-and-mysql-19me)
2. [*https://dev.to/achowba/build-a-simple-app-using-node-js-and-mysql-19me*](https://dev.to/achowba/build-a-simple-app-using-node-js-and-mysql-19me)
3. [*https://toidicode.com/*](https://toidicode.com/)
4. [*https://anonystick.com/*](https://anonystick.com/)
5. [*https://topdev.vn/*](https://topdev.vn/)
6. *http://* [*www.­vre.­cse.­hcmut.­edu.­vn*](http://www.vre.cse.hcmut.edu.vn/)