**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA FAST**

**

**BÁO CÁO MÔN HỌC CHUYÊN ĐỀ MẠNG**

**ĐỀ TÀI: ỨNG DỤNG API LẬP TRÌNH XÂY DỰNG WEBSITE QUẢN LÝ SÁCH**

Người hướng dẫn: **PGS.TS. NGUYỄN TẤN KHÔI**

Sinh viên thực hiện: **LÊ TRẦN TRUNG TUẤN**

**TRƯƠNG ĐÌNH ĐẠT**

Nhóm**: 5**

Lớp: **18PFIEV2**

Ngành**: TIN HỌC CÔNG NGHIỆP**

Nội dung

[Chương 1: CƠ SỞ LÝ THUYẾT 2](#_Toc109466869)

[1.1. CƠ SỞ LÝ THUYẾT 2](#_Toc109466870)

[1.1.1. JavaScript 2](#_Toc109466871)

[1.1.2. HTML/CSS 2](#_Toc109466872)

[1.1.3. Node.js 2](#_Toc109466873)

[1.1.4. React.JS 3](#_Toc109466874)

[1.1.5. Cơ Sở Dữ Liệu 3](#_Toc109466875)

[1.1.6. Các thư viện chính của node.js 4](#_Toc109466876)

[1.1.7. Thẻ HTML dùng trong lập trình 5](#_Toc109466877)

[1.2. PHÁT BIỂU BÀI TOÁN 8](#_Toc109466878)

[1.3. KẾT CHƯƠNG 8](#_Toc109466879)

[Chương 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 9](#_Toc109466880)

[2.1. PHÂN TÍCH YÊU CẦU 9](#_Toc109466881)

[2.2. PHÂN TÍCH CHỨC NĂNG 9](#_Toc109466882)

[2.2.1. Đối tượng sử dụng 9](#_Toc109466883)

[2.2.2. Xem thông tin sách 9](#_Toc109466884)

[2.2.3. Quản lí thông tin sản phẩm 9](#_Toc109466885)

[2.2.4. Tìm kiếm thông tin sách. 9](#_Toc109466886)

[2.2.5. Công nghệ sử dụng 9](#_Toc109466887)

[2.3. TỔ CHỨC CHƯƠNG TRÌNH 10](#_Toc109466888)

[2.3.1. Tổ chức thư mục 10](#_Toc109466889)

[2.3.2. Thư mục controllers 11](#_Toc109466890)

[2.3.3. Thư mục public 11](#_Toc109466891)

[2.3.4. Thư mục routes 11](#_Toc109466892)

[2.3.5. Thư mục config 11](#_Toc109466893)

[2.3.6. Thư mục constants 11](#_Toc109466894)

[2.3.7. Thư mục middleware 11](#_Toc109466895)

[2.3.8. Thư mục migrations 11](#_Toc109466896)

[2.3.9. Thư mục models 11](#_Toc109466897)

[2.3.10. Thư mục node\_module 11](#_Toc109466898)

[2.4. KẾT CHƯƠNG 11](#_Toc109466899)

[Chương 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ 12](#_Toc109466900)

[3.1. TRIỂN KHAI 12](#_Toc109466901)

[Chạy chương trình 12](#_Toc109466902)

[3.2. KẾT QUẢ THỰC NGHIỆM 12](#_Toc109466903)

[3.2.1. Kịch bản 1 – Đăng nhập hệ thống 12](#_Toc109466904)

[3.2.2. Kịch bản 2 – Chức năng xem/sửa dữ liệu 13](#_Toc109466905)

[3.2.3. Kịch bản 3 – Chức năng tìm kiếm trên User page. 15](#_Toc109466906)

[3.2.4. Kịch bản 4 – Chức năng thêm sách 16](#_Toc109466907)

[3.3. NHẬN XÉT ĐÁNH GIÁ KẾT QUẢ 18](#_Toc109466908)

[3.4. KẾT CHƯƠNG 18](#_Toc109466909)

[**1.** **KẾT QUẢ ĐẠT ĐƯỢC** 19](#_Toc109466910)

[**2.** **KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN** 19](#_Toc109466911)

MỞ ĐẦU

**1.**  **Tổng quan về đề tài**

Ngàynay, ứng dụng công nghệ thông tin và việc tin học hoá được xem là một trong những yếu tố mang tính quyết định, có thể tạo ra những bước đột phá mạnh mẽ.

Vì vậy, em đã thực hiện đề tài “Xây dựng webside quản lý sách” cho các thư viện hay nhà sách có thể quản lý được sách của mình.

**2. Mục đích và ý nghĩa của đề tài**

***2.1. Mục đích***

Xây dựng website quản lý sách giúp người quản trị quản lý, thống kê sách.

***2.2. Ý nghĩa***

* Góp phần tối ưu hoá quá trình quản lý, loại bỏ những chi phí về nhân lực, sai sót trong tính toán số lượng sản phẩm.
* Trở thành đề tài, tài liệu nghiên cứu tham khảo cho những bạn có đam mê, yêu thích lập trình web.

**3. Phương pháp thực hiện**

Ứng dụng API để thiết kế xây dựng Website quản lý sách bằng các ngôn ngữ Node JS , React JS , HTML, CSS …

**4. Bố cục**

Báo cáo 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

Để xây dựng và vận hành một website ta sẽ cần sử dụng đến các công nghệ sau:

### JavaScript

JavaScript là một ngôn ngữ lập trình hoặc ngôn ngữ kịch bản cho phép tạo ra các trang web tương tác. JavaScript được tích hợp và nhúng trong HTML.JavaScript cho phép kiểm soát các hành vi của trang web tốt hơn chỉ đơn thuần HTML.

### 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.

### Node.js

Node.js là một nền tảng phía server được xây dựng dựa trên Google Engine V8 - Một JavaScript engine mã nguồn mở phát triển bởi The Chromium Project cho trình duyệt web Google Chrome và Chromium.Node.js giúp cho lập trình viên xây dựng phần backend của một hệ thống web application bằng JavaScript.Node.js tạo ra được các ứng dụng có tốc độ xử lí nhanh, realtime thời gian thực, áp dụng cho các sản phâm có lượng truy cập lớn, cần đổi mới nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

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.

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

### React.JS

Reacjs là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác.

Nếu như AngularJS là một framework cho phép người dùng nhúng code javascript trong code html thông qua các attribute như ng-model,ng-repeat thì với react là một thư viện cho nhúng code vào HTML trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng đoạn HTML vào JS. Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn

### 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, ..

### Các thư viện chính của node.js

Sử dụng lệnh ***npm install [package-name]*** để cài đặt các gói tin cần thiết

* ***npm init --yes:*** tạo ra file package.json mẫu bên trong bao gồm các thuộc tính:

***name :*** Tên gói thư viện

***version:*** Phiên bản gói

***description :*** Phần mô tả về gói thư viện

***author:*** Tác giả

***dependencies :*** Danh sách các gói phụ thuộc, tự động được cài theo

***repository:*** Loại repository và url của package thông thường hay là git

***main:***index.js

***keyword:*** Các từ khóa

* ***npm install express :*** Express cung cấp các tính năng để phát triển web hoặc mobile, hỗ trợ các các method HTTP và middleware tạo ra API dễ sử dụng và mạnh mẽ.Một số chức năng chính của express như : thiết lập các lớp trung gian để trả về HTTP request, Define router cho phép sử dụng với các hành động khác nhau dựa theo phương thức HTTP và URL, cho phép trả về các trang HTML dựa vào các tham số.
* ***npm install express-fileupload***: Một express middleware đơn giản để tải lên các tập tin
* ***npm install body-parser :*** Lấy được các dữ liệu nhập vào
* ***npm install mysql :*** Trình điều khiển node.js cho mysql
* ***npm install ejs :*** Một view engine được sử dụng để phân tích các tập tin ejs, và tạo ra HTML trả về cho client.
* ***npm install req-flash :***Phần mềm trung gian chưa được tối ưu hóa để tạo các tin nhắn flash thuộc mọi loại cho các ứng dụng express
* ***npm install nodemon:*** Một công cụ giúp phát triển các ứng dụng node.js bằng cách tự động khởi động lại ứng dụng node.js khi phát hiện các thay đổi tệp trong thư mục.
* ***npm install express-session:***
* ***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.

### Thẻ HTML dùng trong lập trình

* 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.
* ***req*** : biểu diễn HTTP Request và có các thuộc tính cho chuỗi truy vấn request, các tham số, body, HTTP Header, …
* ***res*** : Đối tượng res biểu diễn HTTP Response mà ứng dụng Express gửi phản hồi khi nhận một HTTP Request.
* ***res.render*** : Kết xuất một khung nhìn với một cuộc gọi lại đáp ứng với chuỗi kết xuất. Khi xảy ra lỗi next(err) được gọi bên trong. Khi một cuộc gọi lại được cung cấp cả lỗi có thể và chuỗi kết xuất được thông qua và không có phản hồi tự động nào được thực hiện.

## PHÁT BIỂU BÀI TOÁN

Ứng dụng API lập trình xây dựng website quản lý sách. Xây dựng thiết kế các tính năng cho Website chẳng hạn như : tìm kiếm thông tin sách đối với User , quản lý sách , thêm sách , xóa sách … đối với Admin.

## 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
* Lập trình chức năng login vào Website
* 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 người quản trị : người có quyền kiểm soát mọi hoạt động của hệ thống. 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 cuốn sách.

### Xem thông tin sách

* Cho phép xem các thông tin liên quan đến sách : Tên sách

### Quản lí thông tin sản phẩm

* Người quản trị có thể cập nhật, thêm, sửa, xoá các thông tin các cuốn sách, loại sách.

### Tìm kiếm thông tin sách.

Chức năng này cho phép người sử dụng có thể tìm kiếm tên sách, website sẽ hiển thị các thông tin, nếu như website tìm kiếm được thông tin phù hợp.

### Công nghệ sử dụng

* NODE JS : NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.
* MySQL : **MySQL** là [hệ quản trị cơ sở dữ liệu](https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u) [tự do nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F) phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet.

## TỔ CHỨC CHƯƠNG TRÌNH

### Tổ chức thư mục

Ảnh có chứa văn bản

Mô tả được tạo tự động

### Thư mục controllers

Lưu trữ các file controller.js để xử lý các yêu cầu của người dùng như thêm/xóa/sửa và đăng nhập/ đăng xuất.

### Thư mục public

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

### Thư mục routes

Xử lý các trang của website

### Thư mục config

Là nơi chứa các thông tin của cơ sở dữ liệu mà người dùng dùng, khi chạy project thì nó lấy thông tin đó ra kết nối cơ sở dữ liệu.

### Thư mục constants

Các hằng không thay đổi sử dụng trong cả project.

### Thư mục middleware

Chứa file auth.js, nhiệm vụ chặn bắt các request từ Front-end, giải mã xem có quyền truy cập tới controller không. nếu có quyền thì giải mã token của request rồi đẩy tiếp tới controller xử lí, nếu không có quyền thì chặn không cho thực hiện.

### Thư mục migrations

Các file trong thư mục này dùng để tạo các bảng cho cơ sở dữ liệu.

### Thư mục models

Các file thư mục này đại diện cho các thực thể như User,… trong project dùng nó để tương tác với cơ sở dữ liệu.

### Thư mục node\_module

Thư mục chứa các thư viện cần thiết cho dự án.

## 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 và công nghệ sử dụng.

# TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

## TRIỂN KHAI

### Chạy chương trình

**Bước 1 :** Khởi động webside, ta thực hiện lệnh node app.js ở folder chính:

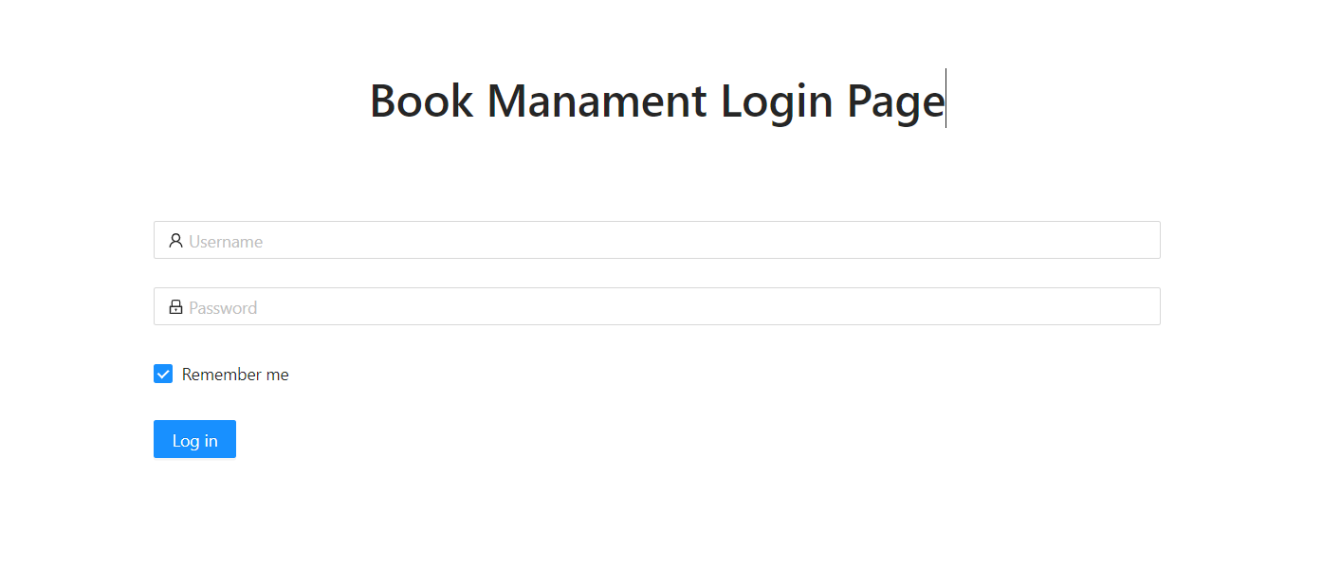
F:\DOANCHUYENDEMANG\Book-management-main\Book-management-main> node app.js

**Bước 2 :** Ta truy cập website cho admin bằng cách thực hiện lệnh *npm start* ở forder admin : F:\DOANCHUYENDEMANG\book-management-admin-main , trang đầu tiên sẽ là trang login, ta tiến hành nhập Username : admin và Password : admin (thông tin user và password đã được chèn vào bảng member ở phần tạo database )

Bước 3 : Ta truy cập website cho user bằng cách thực hiện lệnh *npm start* ở forder user : F:\DOANCHUYENDEMANG\book-management-client-main.

## KẾT QUẢ THỰC NGHIỆM

### Kịch bản 1 – Đăng nhập hệ thống



Hình 3.1 Giao diện đăng nhập

Username: dattruong@gmail.com

Password: 123456

* TH1: Người dùng đăng nhập đúng:

Website sẽ chuyển đến trang chủ hiển thị thông tin sách .

* TH2: Người dùng nhập sai username hay password:

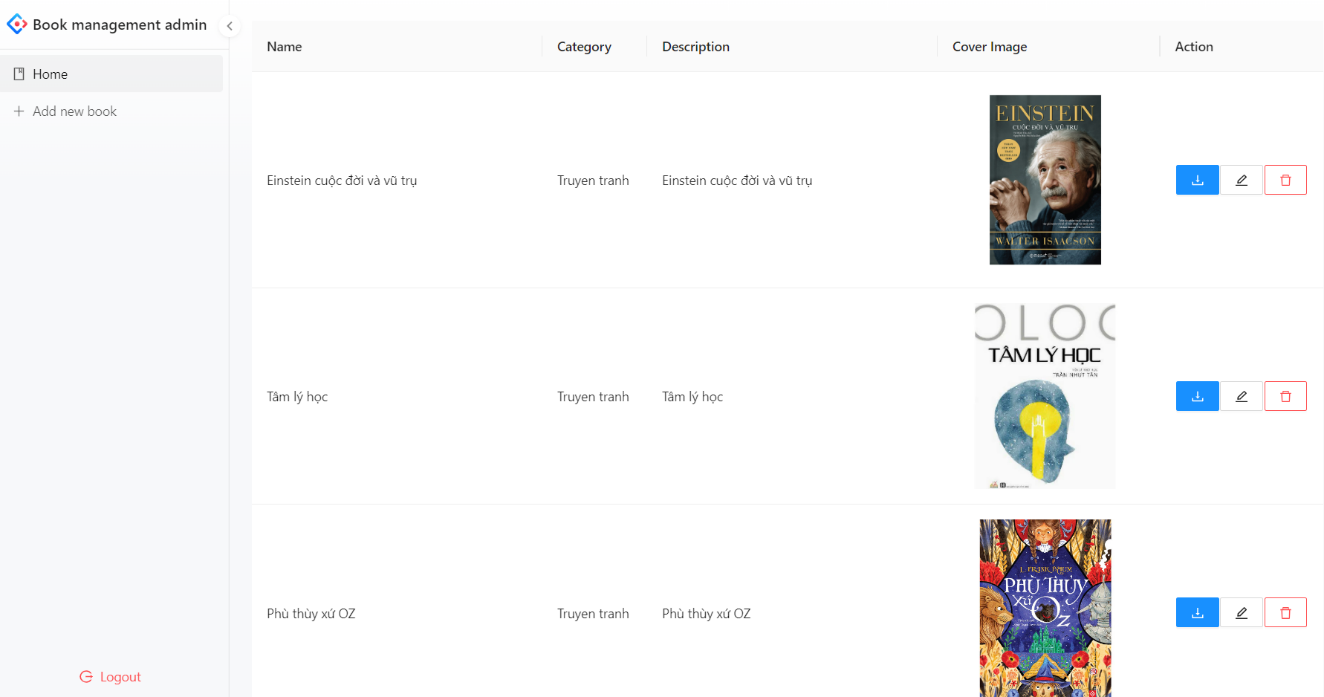
Website sẽ không chuyển đến trang chủ cho đến khi người dùng đăng nhập đúng cú pháp.

Ảnh có chứa văn bản

Mô tả được tạo tự động

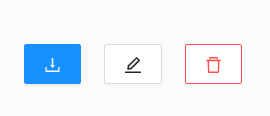
### Kịch bản 2 – Chức năng xem/sửa dữ liệu

* Ở giao diện này người dùng có thể xem thông tin sách.



Hình 3.2 Giao diện hiển thị thông tin sách

* Người quản trị có thể thực hiện thao tác sửa/xoá thông tin sách

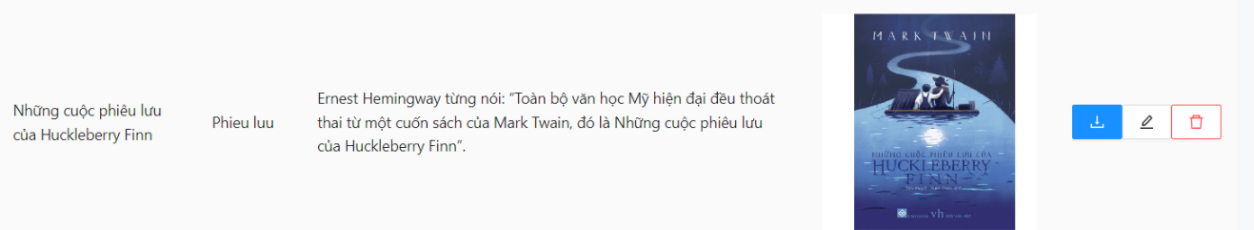


Ví dụ: Lúc chưa chỉnh sửa



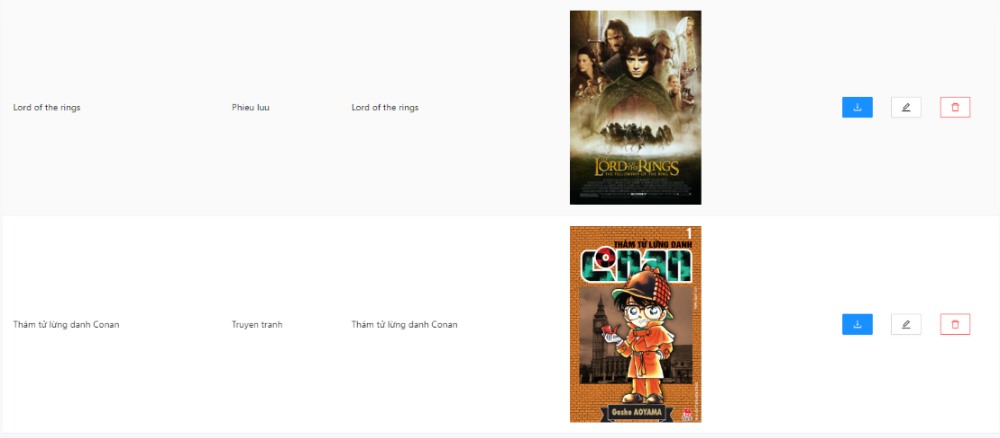
Hình 3.3 Thông tin sản phẩm khi chưa sửa

* Sau khi chỉnh sửa



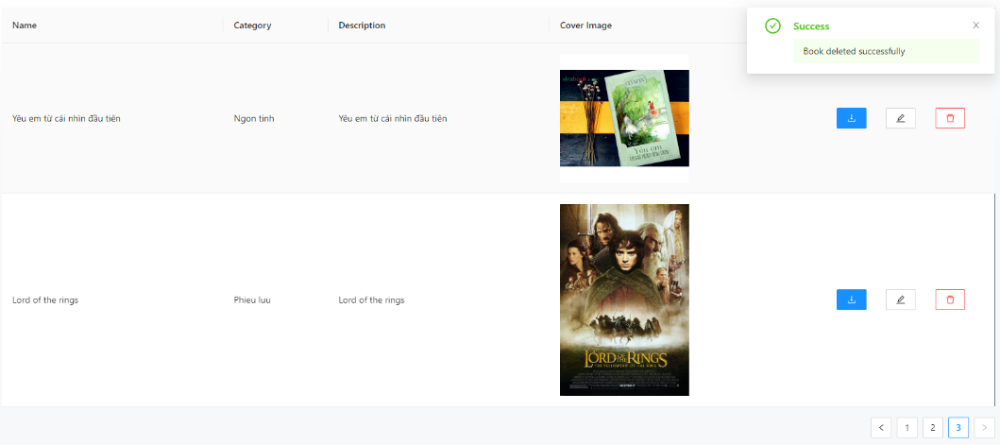
Hình 3.4 Thông tin sản phẩm sau khi chỉnh sửa

* Người quản trị click nút  để xoá sản phẩm sách Conan.



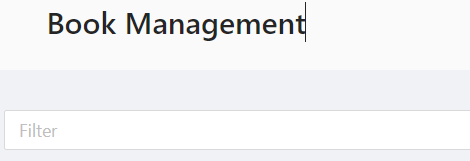
Hình 3.5 Giao diện trang chủ khi chưa xoá sản phẩm

* Sau khi xoá:

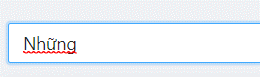


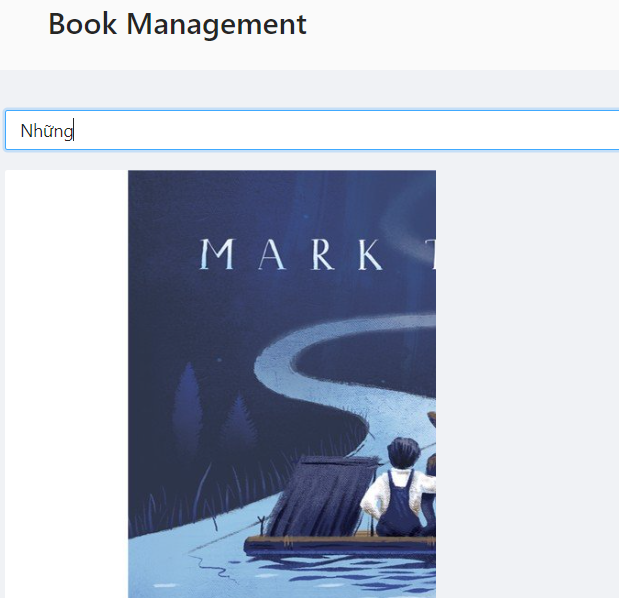
Hình 3.6 Giao diện trang chủ sau khi xoá sản phẩm

### Kịch bản 3 – Chức năng tìm kiếm trên User page.



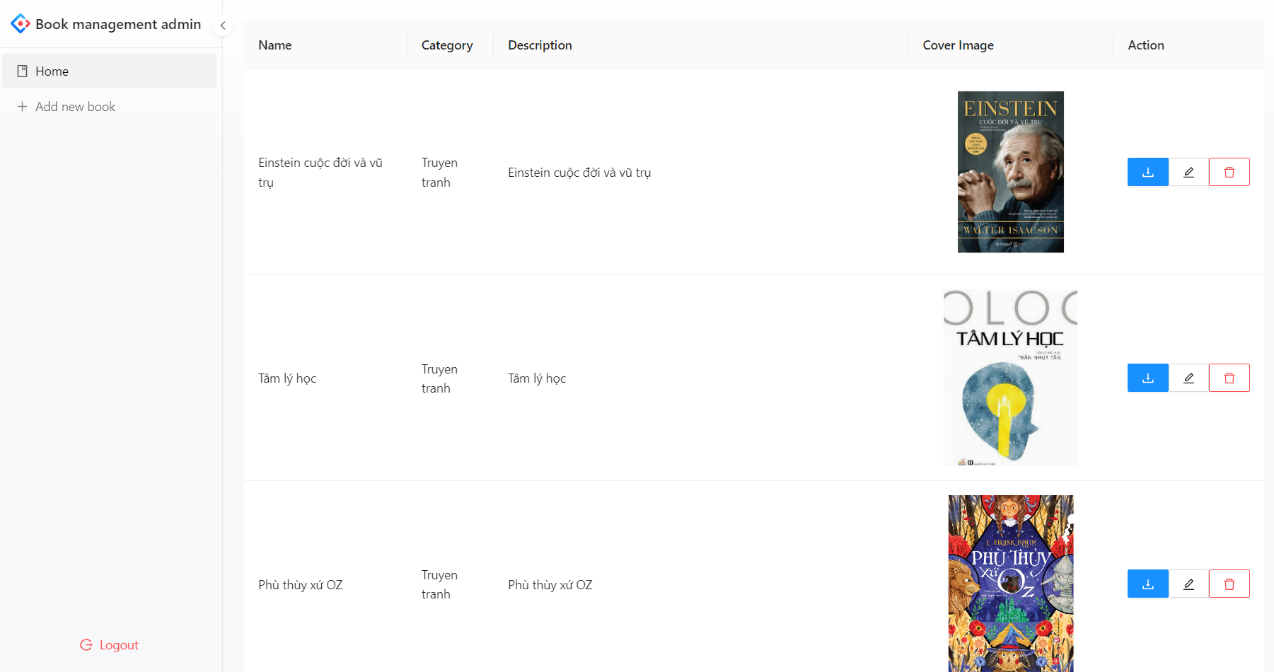
Hình 3.7 Ô tìm kiếm trên Website

* Tìm kiếm  ta được kết quả:



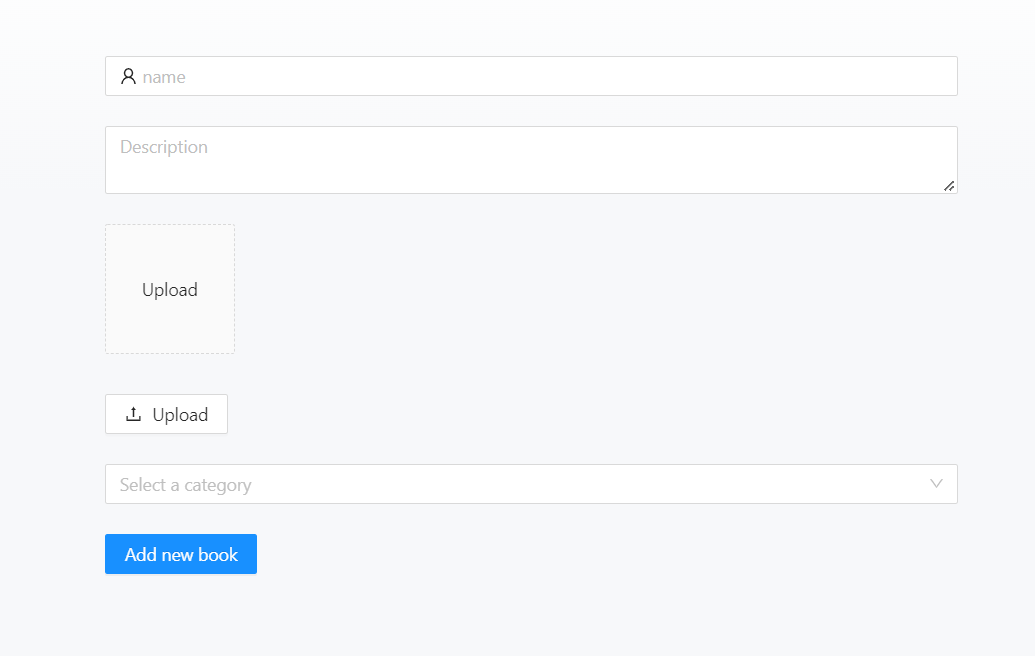
Hình 3.8 Kết quả tìm kiếm

### Kịch bản 4 – Chức năng thêm sách

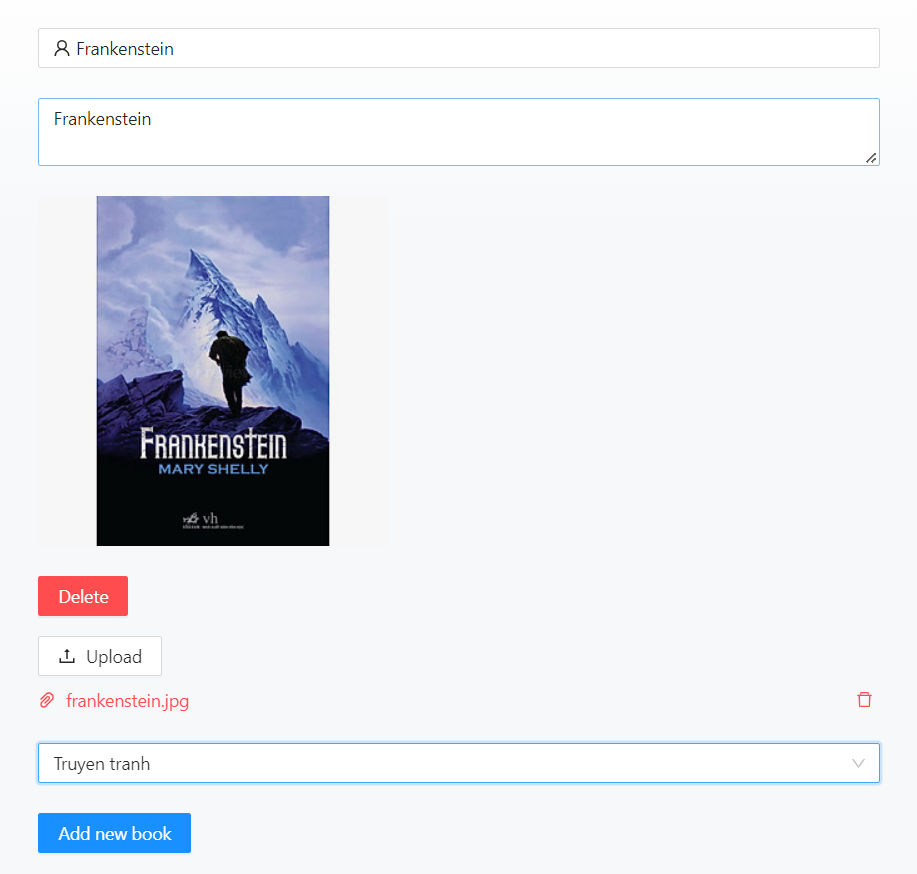


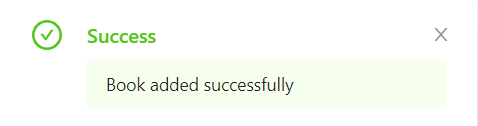
Hình 3.9 Giao diện trang chủ khi chưa thêm sách

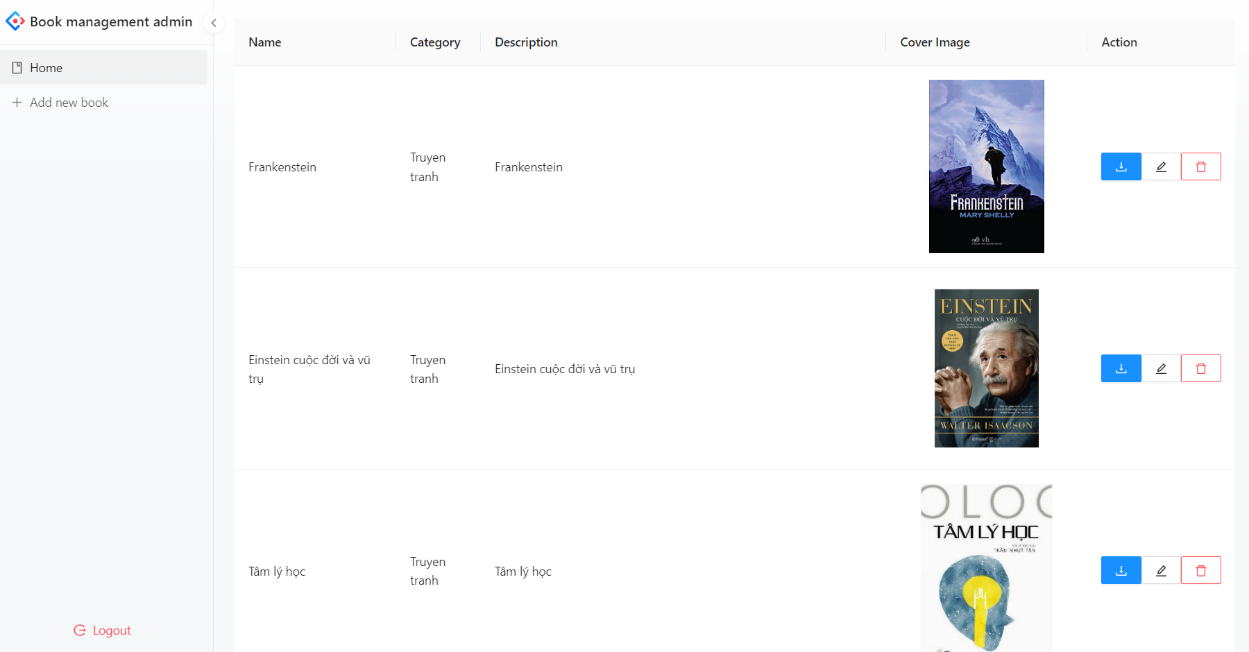
Ta thử thêm một cuốn sách khác , chọn Add new book . Giao diện như sau



Giả sử muốn thêm cuốn sách Frankenstein. Ta nhập tên sách , mô tả … , ta được như sau :



* Thông báo khi thêm sách thành công



Hình 3.10 Giao diện trang chủ sau khi thêm sản phẩm

## 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 sách , 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 sách

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**

*https://github.com/dattruong1225/book-management-main*