|  |  |
| --- | --- |
|  | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HỒ CHÍ MINH** |

**ĐỒ ÁN MÔN HỌC**

**<WEBSITE ĐỌC TRUYỆN>**

Ngành: Công Nghệ Thông Tin

Môn : Lập Trình Web

Giảng viên hướng dẫn : Nguyễn Huy Cường

Sinh viên: Đặng Khánh Duy MSSV: 2080600935 Lớp: 21DTHC1

Sinh viên: Huỳnh Lưu Thiên Tân MSSV: 2180602903 Lớp: 21DTHC1

Sinh viên:Võ Hoàng Duy MSSV: 2180605888 Lớp: 21DTHC1

Sinh viên:Huỳnh Nhật Minh MSSV: 2180605888 Lớp: 21DTHC1

TP. Hồ Chí Minh, 2024

# Lời cam đoan

Chúng em xin cam đoan đề tài: “ Website đọc truyện” là một công trình nghiên cứu độc lập dưới sự hướng dẫn của giảng viên hướng dẫn: Nguyễn Huy Cường. Đề tài, nội dung báo cáo là sản phẩm mà chúng em đã nỗ lực nghiên cứu trong quá trình học tập tại trường cũng như nghiên cứu thực tế. Các số liệu, kết quả trình bày trong báo cáo là hoàn toàn trung thực, chúng em xin chịu hoàn toàn trách nhiệm, kỷ luật của bộ môn và nhà trường đề ra nếu như có vấn đề xảy ra.

**MỤC LỤC**

[Lời cam đoan 2](#_Toc164347258)

[CHƯƠNG 1. TỔNG QUAN 4](#_Toc164347259)

[1.1 Mô tả đề tài 4](#_Toc164347260)

[1.2 Lý do chọn đề tài 4](#_Toc164347261)

[1.3. Nhiệm vụ đồ án 5](#_Toc164347262)

[CHƯƠNG 2. LÝ THUYẾT CƠ SỞ 7](#_Toc164347263)

[2.1 Tổng quan về SQL 7](#_Toc164347264)

[2.2 Tổng quan về Visual Studio 8](#_Toc164347265)

[2.3 Tổng quan về MVC 9](#_Toc164347266)

[2.5 Tổng quan về html, css, javascript 10](#_Toc164347267)

[2.5 Tổng quan về AJAX 14](#_Toc164347268)

[CHƯƠNG 3 : KẾT QUẢ THỰC NGHIỆM: 15](#_Toc164347269)

[3.1. Database diagram 15](#_Toc164347270)

[3.2.Xây dựng mô hình chức năng(Usecase Diagram) 15](#_Toc164347271)

[*3.2.1 Danh sách actor* 15](#_Toc164347272)

[*3.2.2 Danh sách Usecase* 15](#_Toc164347273)

[3.3. Đặc tả Use-case chức năng 16](#_Toc164347274)

[*3.3.1 Đặc tả use-case “Đăng nhập”* 16](#_Toc164347275)

[*3.3.2 Đặc tả use-case “Thêm truyện”* 17](#_Toc164347276)

[*3.3.3 Đặc tả use-case “Sửa truyện”* 18](#_Toc164347277)

[*3.3.4 Đặc tả use-case “xóa truyện”* 19](#_Toc164347278)

[*3.3.5 Đặc tả use-case “tìm kiếm”* 20](#_Toc164347279)

[*3.3.6 Đặc tả use-case “đọc truyện”* 21](#_Toc164347280)

[3.4.Giao diện 22](#_Toc164347281)

[*3.4.1. Đăng nhập* 22](#_Toc164347282)

[*3.4.2. Admin* 22](#_Toc164347283)

[*3.4.3. Trang đọc truyện:* 26](#_Toc164347284)

[CHƯƠNG 4 : TỔNG KẾT 30](#_Toc164347285)

# CHƯƠNG 1. TỔNG QUAN

## 1.1 Mô tả đề tài

Đề tài lập trình Website đọc truyện là một chủ đề phổ biến trong lĩnh vực công nghệ thông tin. Dự án này nhằm xây dựng một trang web cho phép quản trị viên (admin) đăng tải và quản lý các câu chuyện trực tuyến một cách dễ dàng. Trang web sẽ cung cấp một nền tảng cho admin để nhập liệu và quản lý các truyện từ nhiều thể loại khác nhau, đồng thời cho phép người đọc dễ dàng trải nghiệm các tác phẩm này.

Chức năng

* Quản lý truyện: Giao diện dành cho admin để đăng tải, chỉnh sửa và xóa các truyện. Admin có thể nhập thông tin về tiêu đề, tác giả (nếu có), thể loại và nội dung của truyện.
* Thư viện truyện đa dạng: Hiển thị một thư viện truyện phong phú với nhiều thể loại khác nhau như tiểu thuyết, trinh thám, lãng mạn, kinh dị, hài hước, và nhiều thể loại khác.
* Tìm kiếm và lọc truyện: Cho phép người dùng tìm kiếm truyện theo tiêu đề, thể loại và từ khóa khác.
* Giao diện thân thiện với người dùng: Thiết kế giao diện đẹp mắt, dễ sử dụng trên cả máy tính và thiết bị di động.
* Quản lý tài khoản: Cho phép người dùng đăng ký tài khoản, quản lý thông tin cá nhân

## 1.2 Lý do chọn đề tài

Việc chọn đề tài xây dựng một trang web đọc truyện có thể được lựa chọn vì một số lý do sau:

* Sở thích cá nhân: Nếu bạn là một người đam mê văn hóa đọc và muốn tạo ra một nền tảng cho cộng đồng đọc truyện trực tuyến, việc xây dựng một trang web đọc truyện sẽ là một sự lựa chọn lý tưởng.
* Nhu cầu thị trường: Văn hóa đọc trực tuyến đang trở thành một xu hướng phổ biến, đặc biệt là trong thời đại số hóa ngày nay. Người đọc thường tìm kiếm các nền tảng trực tuyến để truy cập và đọc các tác phẩm một cách thuận tiện từ bất kỳ thiết bị nào.
* Tiềm năng kinh doanh: Nếu bạn nhìn vào khía cạnh kinh doanh, việc xây dựng một trang web đọc truyện có thể mang lại cơ hội kinh doanh đáng kể. Bạn có thể tạo ra các mô hình kinh doanh như quảng cáo, đăng ký thành viên, hoặc bán sách điện tử để tạo ra doanh thu.
* Thiết kế sản phẩm sáng tạo: Xây dựng một trang web đọc truyện cũng có thể là cơ hội để thể hiện sự sáng tạo và kỹ năng kỹ thuật của bạn. Bạn có thể thách thức bản thân trong việc tạo ra một giao diện đẹp mắt, tích hợp các tính năng tiện ích, và cung cấp trải nghiệm đọc tuyệt vời cho người dùng.
* Thúc đẩy văn hóa đọc: Bằng cách cung cấp một nền tảng trực tuyến cho việc đọc truyện, bạn có thể đóng góp vào việc thúc đẩy văn hóa đọc và giáo dục trong cộng đồng. Điều này có thể góp phần làm tăng cường kiến thức và tạo ra một xã hội đọc sách tích cực hơn.

Tóm lại, việc chọn đề tài xây dựng một trang web đọc truyện có thể được hỗ trợ bởi sở thích cá nhân, nhu cầu thị trường, tiềm năng kinh doanh, mong muốn thiết kế sản phẩm sáng tạo, và mục tiêu thúc đẩy văn hóa đọc trong cộng đồng.

## 1.3. Nhiệm vụ đồ án

Dưới đây là các chức năng chính của Website đọc truyện:  
+ Quản lý truyện: Cho phép thêm, xóa, sửa, tìm kiếm nội dung của truyện.   
+ Quản lý tác giả: Cho phép thêm, xóa, sửa, tìm kiếm thông tin của tác giả.   
+ Quản lý thể loại: Cho phép thêm, xóa, sửa, tìm kiếm thông tin của thể loại.

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **NỘI DUNG** | **MÔ TẢ CHI TIẾT** | **QUYỀN CHO PHÉP** |
| 1 | đăng nhập | đăng nhập sử dụng giao diện | Admin |
| 2 | Tính năng trang admin | Thêm, xóa, sửa, cập nhật thông tin về truyện , chương, tác giả, thể loại. | Admin |
| 3 | Phân quyền đăng nhập | Phân quyền cho Admin và các người dùng. | Admin |
| 4 | Tính năng trang thống tin truyện | Cập nhật tên , thể loại, nội dung truyện. | Admin |

**Bảng 1**: Tổng hợp chức năng

# CHƯƠNG 2. LÝ THUYẾT CƠ SỞ

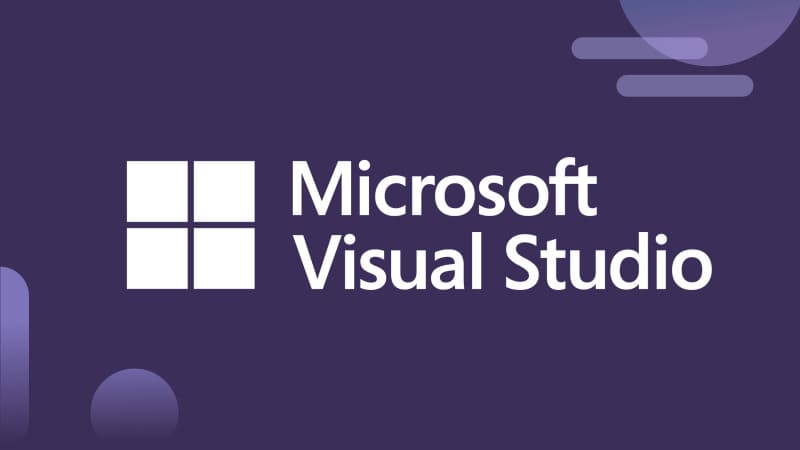
## 2.1 Tổng quan về SQL



SQL (Structured Query Language) là một ngôn ngữ lập trình sử dụng để quản lý và truy vấn cơ sở dữ liệu quan hệ (RDBMS - Relational Database Management System). SQL được sử dụng để tạo, thêm, xóa, sửa đổi cấu trúc cơ sở dữ liệu và truy xuất dữ liệu có trong cơ sở dữ liệu. SQL là một ngôn ngữ mạnh mẽ và phổ biến được sử dụng trong hầu hết các hệ thống quản lý cơ sở dữ liệu hiện đại. SQL có thể được sử dụng để truy vấn dữ liệu, tạo bảng, index, views, procedures và triggers để quản lý và tổ chức dữ liệu.là một ngôn ngữ lập trình dùng để quản lý và truy xuất các hệ thống quản trị cơ sở dữ liệu (DBMS).

SQL được sử dụng để tạo, sửa đổi và truy xuất dữ liệu trong các hệ thống quản trị cơ sở dữ liệu phổ biến như MySQL, Oracle, SQL Server, PostgreSQL, và SQLite.SQL được phát triển để giúp người dùng có thể tương tác với các cơ sở dữ liệu một cách chuyên nghiệp và an toàn. SQL cho phép người dùng thực hiện các thao tác đơn giản như tìm kiếm dữ liệu, thêm, sửa đổi hoặc xóa dữ liệu cũng như thực hiện các thao tác phức tạp hơn như tạo bảng, quản lý quyền truy cập và tối ưu hóa cơ sở dữ liệu.

## 2.2 Tổng quan về Visual Studio



Visual Studio là một môi trường phát triển tích hợp (Integrated Development Environment

- IDE) được phát triển bởi Microsoft. Được ra mắt lần đầu vào năm 1997, Visual Studio đã trở thành một trong những công cụ phát triển phổ biến và mạnh mẽ nhất trên thị trường phần mềm.

Dưới đây là một số điểm nổi bật về Visual Studio:

1. Hỗ trợ nhiều ngôn ngữ: Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau bao gồm C#, C++, F#, Python, và nhiều ngôn ngữ khác. Điều này cho phép nhà phát triển lựa chọn ngôn ngữ phù hợp với dự án của mình.

2. Hỗ trợ đa nền tảng: Visual Studio hỗ trợ phát triển ứng dụng chạy trên nhiều nền tảng, bao gồm Windows, Linux và macOS. Điều này đặc biệt quan trọng với sự phát triển của .NET Core và ứng dụng đa nền tảng.

3. IntelliSense: Visual Studio cung cấp tính năng IntelliSense, giúp người lập trình tăng tốc độ viết mã bằng cách tự động hoàn thiện cú pháp, gợi ý từ khóa và đề xuất các phương thức và thuộc tính.

4. Gỡ lỗi mạnh mẽ: IDE cung cấp một bộ công cụ gỡ lỗi mạnh mẽ, cho phép người lập trình dễ dàng tìm và sửa lỗi trong mã nguồn.

5. Quản lý mã nguồn: Visual Studio tích hợp công cụ quản lý phiên bản (version control) cho phép nhóm phát triển làm việc cùng nhau trên cùng một dự án và theo dõi sự thay đổi của mã nguồn.

6. Thiết kế giao diện người dùng (UI Design): IDE cung cấp các công cụ thiết kế giao diện người dùng dễ sử dụng, cho phép người lập trình kéo và thả các thành phần để tạo ra giao diện ứng dụng.

7. Hỗ trợ đa dạng cho các loại ứng dụng: Visual Studio hỗ trợ phát triển nhiều loại ứng dụng, bao gồm ứng dụng máy tính, ứng dụng di động (sử dụng Xamarin), ứng dụng web (sử dụng ASP.NET), trò chơi và nhiều loại ứng dụng khác.

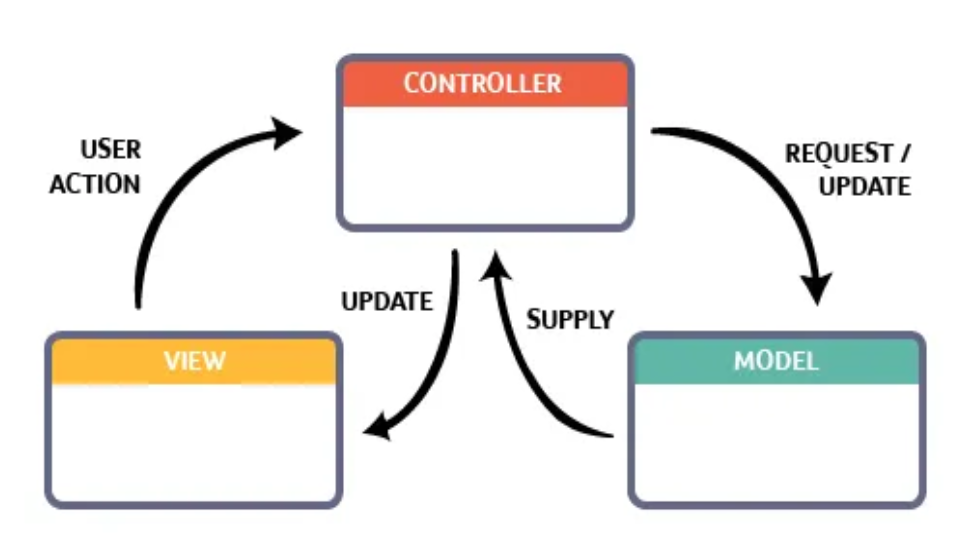
Một cộng đồng mạnh mẽ: Visual Studio có một cộng đồng lớn và đa dạng, với nhiều tài liệu, diễn đàn và nguồn tài nguyên hướng dẫn để hỗ trợ người lập trình.

Tóm lại, Visual Studio là một IDE mạnh mẽ và đa chức năng, cung cấp mọi thứ mà một nhà phát triển cần để xây dựng ứng dụng phức tạp và đa dạng trên nhiều nền tảng.

## 2.3 Tổng quan về MVC

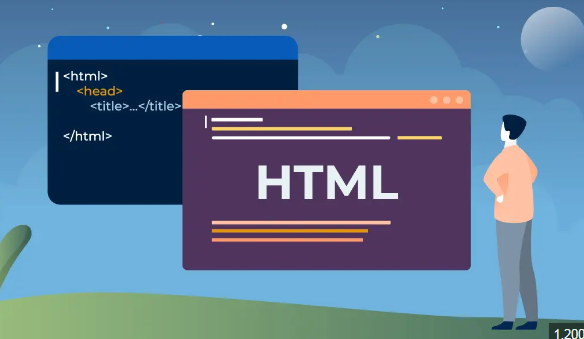
MVC là viết tắt của cụm từ “Model-View-Controller“. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác. Tên gọi 3 thành phần:

* Model (dữ liệu): Quản lí xử lí các dữ liệu.
* View (giao diện): Nới hiển thị dữ liệu cho người dùng.
* Controller (bộ điều khiển): Điều khiển sự tương tác của hai thành phần Model và View.



**Mô hình MVC (MVC pattern)** thường được dùng để **phát triển giao diện**người dùng. Nó cung cấp các thành phần cơ bản để thiết kế một chương trình cho **máy tính** hoặc **điện thoại di động**, cũng như là các **ứng dụng web.**

## 2.5 Tổng quan về html, css, javascript



* HTML (HyperText Markup Language) là ngôn ngữ đánh dấu được sử dụng để tạo ra cấu trúc nội dung của một trang web. Nó là một phần không thể thiếu trong việc xây dựng các trang web và đóng vai trò quan trọng trong việc xác định cách mà các phần tử trên trang web được tổ chức và hiển thị.

+Cấu trúc cơ bản của HTML:

1.Thẻ (Tag): HTML sử dụng các thẻ để đánh dấu và xác định các phần tử trên trang web. Mỗi thẻ bắt đầu bằng ký tự < và kết thúc bằng ký tự >. Ví dụ:

<html>: Định nghĩa khối chứa toàn bộ nội dung của trang web.

<head>: Chứa các thông tin meta về trang web như tiêu đề, định dạng, và các tập tin CSS và JavaScript.

<body>: Chứa nội dung hiển thị của trang web.

2.Phần tử (Element): Một phần tử HTML được tạo ra bằng cách sử dụng một cặp thẻ mở và thẻ đóng, bao gồm nội dung bên trong. Ví dụ:

<h1>: Định nghĩa tiêu đề lớn nhất.

<p>: Định nghĩa một đoạn văn bản.

<img>: Hiển thị một hình ảnh trên trang web.

<a>: Định nghĩa một liên kết.

3.Thuộc tính (Attribute): Mỗi thẻ HTML có thể chứa các thuộc tính để cung cấp thông tin bổ sung về phần tử đó. Các thuộc tính được đặt trong cặp dấu ngoặc kép và được đặt trong thẻ mở. Ví dụ:

<img src="image.jpg" alt="Ảnh mô tả">: Thuộc tính src chỉ định đường dẫn đến hình ảnh, và thuộc tính alt cung cấp văn bản mô tả cho hình ảnh (cho trường hợp hình ảnh không thể hiện được).

<a href="https://www.example.com">Link đến trang ví dụ</a>: Thuộc tính href chỉ định địa chỉ URL mà liên kết sẽ dẫn đến.

HTML cung cấp một cách đơn giản và cấu trúc để xây dựng trang web, và là nền tảng cho việc thêm CSS và JavaScript để cải thiện giao diện và chức năng của trang.



* CSS (Cascading Style Sheets) là ngôn ngữ được sử dụng để kiểm soát giao diện và trang trí của một trang web. Nó cho phép bạn tùy chỉnh cách mà các phần tử HTML được hiển thị trên trình duyệt, bao gồm màu sắc, font chữ, kích thước, đường viền, khoảng cách và vị trí.

Dưới đây là một số khái niệm cơ bản trong CSS:

-Lựa chọn phần tử (Selectors): CSS sử dụng các lựa chọn phần tử để xác định phần tử HTML mà bạn muốn áp dụng các quy tắc CSS. Các lựa chọn phổ biến bao gồm tên thẻ, lớp (class), ID và các thuộc tính khác. Ví dụ:

Tên thẻ: p { color: blue; } - Áp dụng màu xanh cho tất cả các đoạn văn bản <p>.

-Lớp: .header { font-size: 24px; } - Áp dụng font chữ có kích thước 24px cho tất cả các phần tử có lớp là "header".

ID: #main-content { background-color: #f0f0f0; } - Đặt màu nền cho phần tử có ID là "main-content".

-Quy tắc (Rules): Mỗi quy tắc trong CSS bao gồm một lựa chọn phần tử và một khối khai báo thuộc tính và giá trị muốn áp dụng cho phần tử đó. Ví dụ:

Css

h1 {

color: red;

font-size: 36px;

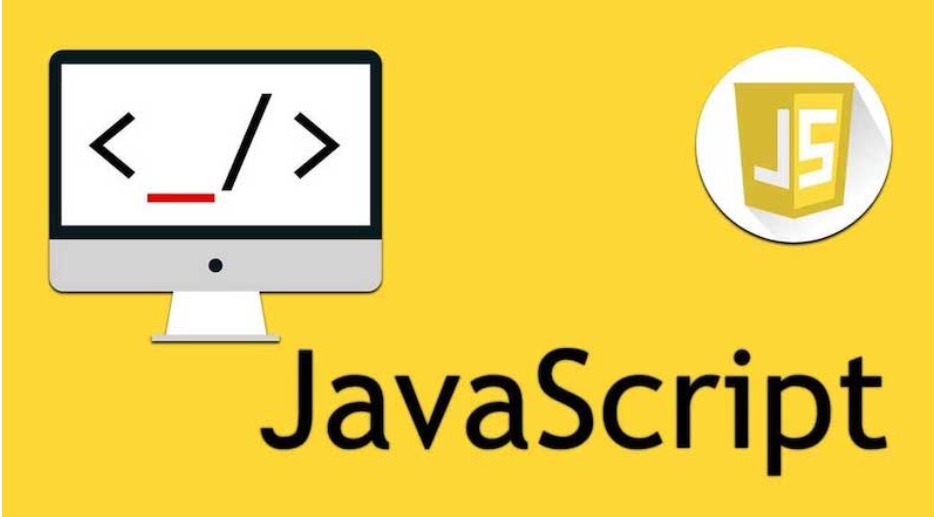
}

-Kích thước và vị trí (Size and Position): CSS cho phép bạn kiểm soát kích thước và vị trí của các phần tử trên trang web bằng cách sử dụng thuộc tính như width, height, margin, padding, position, và top, left, right, bottom.

-Trang trí (Decoration): Bạn có thể trang trí các phần tử bằng cách sử dụng thuộc tính như background-color, border, box-shadow, text-decoration, và text-transform.

-Responsive Design: CSS cũng hỗ trợ responsive design, cho phép bạn tạo ra các trang web linh hoạt và đáp ứng với kích thước màn hình của các thiết bị khác nhau, bằng cách sử dụng media queries và các kỹ thuật khác nhau.

-CSS là một phần quan trọng trong việc xây dựng trang web, giúp bạn tạo ra giao diện đẹp và linh hoạt cho người dùng.



* JavaScript là một ngôn ngữ lập trình phổ biến trong lập trình web. Nó được sử dụng để tạo ra các chức năng tương tác trên trang web, xử lý sự kiện, thay đổi nội dung động và tương tác với người dùng. Dưới đây là một số khái niệm cơ bản về JavaScript:

-Sự kiện (Events): JavaScript cho phép bạn xử lý sự kiện xảy ra trên trang web, như click chuột, gõ phím, hover chuột, và nhiều sự kiện khác. Bằng cách sử dụng các sự kiện này, bạn có thể tạo ra các chức năng tương tác để phản ứng với hành động của người dùng.

-DOM (Document Object Model): DOM là một biểu diễn cấu trúc của trang web dưới dạng một cây các nút, trong đó mỗi phần tử HTML trên trang web được đại diện bởi một nút trong cây. JavaScript cho phép bạn truy cập và thay đổi DOM, giúp bạn thay đổi nội dung và cấu trúc của trang web.

-Biến (Variables): JavaScript sử dụng biến để lưu trữ và đại diện cho dữ liệu. Bạn có thể khai báo biến bằng từ khóa var, let, hoặc const, và gán giá trị cho biến bằng dấu = .

Ví dụ:

javascript

var message = "Hello, world!";

-Hàm (Functions): Hàm là một khối mã được đặt tên và có thể được gọi lại để thực thi một tập hợp các câu lệnh. JavaScript cung cấp cú pháp để khai báo hàm, truyền tham số cho hàm, và trả về giá trị từ hàm. Ví dụ:

javascript

function greet(name) {

return "Hello, " + name + "!";

}

-Điều kiện và vòng lặp (Conditionals and Loops): JavaScript cung cấp các cấu trúc điều kiện như if, else if, else và các vòng lặp như for, while, để kiểm soát luồng thực thi của mã.

-Đối tượng và phương thức (Objects and Methods): JavaScript là một ngôn ngữ dựa trên đối tượng, có nghĩa là mọi thứ trong JavaScript đều là đối tượng hoặc có thể được xem xét là đối tượng. Bạn có thể tạo ra các đối tượng và phương thức để tổ chức mã của mình một cách cấu trúc hơn.

-JavaScript là một công cụ mạnh mẽ trong việc phát triển trang web động và tương tác, và nó đóng vai trò quan trọng trong việc cung cấp trải nghiệm người dùng tốt hơn trên web.

## 2.5 Tổng quan về AJAX

AJAX (Asynchronous JavaScript and XML): giúp người dùng trải nghiệm tốt hơn. Khi cần

một thay đổi nhỏ thì sẽ không cần load lại cả trang web, làm trang web phải tải lại nhiều

thứ không cần thiết.

Ví dụ: Tính năng Gợi ý của Google

⚫ Cách thức hoạt động của AJAX:

❑ Giao tiếp bất đồng bộ với server

❑ Dữ liệu truyền nhận: JSON / XML/ Javascript, HTML ...

❑ Lập trình phía client: JavaScript

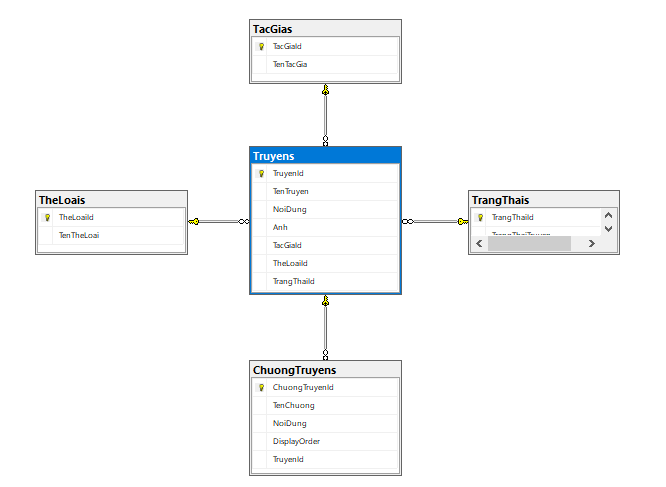
⚫ AJAX jQuery là jQuery cung cấp các phương thức để thực

hiện các chức năng của AJAX. Từ đó, người dùng có thể thực hiện yêu cầu text, HTML, JSON

hay XML hoặc dễ dàng lấy dữ liệu từ bên ngoài vào trong phần tử được chọn.

# CHƯƠNG 3 : KẾT QUẢ THỰC NGHIỆM:

## 3.1. Database diagram



Hình 3.1: Database diagram

## 3.2.Xây dựng mô hình chức năng(Usecase Diagram)

### *3.2.1 Danh sách actor*

+ Admin

+ Người dùng

### *3.2.2 Danh sách Usecase*

+ Quản lý thông tin truyện

+ Quản lý thông tin tác giả

+ Chi tiết thể loại

+Đăng nhập

## 3.3. Đặc tả Use-case chức năng

### *3.3.1 Đặc tả use-case “Đăng nhập”*

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use-case** | UC-Login | | |
| **Tên Use-case** | Use-case Login | | |
| **Tác nhân** | Admin + user | | |
| **Mô tả** | Cho phép admin hoặc user truy cập vào hệ thống | | |
| **Luồng sự kiện** |  | **Tác nhân** | **Hệ thống phản hồi** |
| 1 | Truy cập vào website |  |
| 2 |  | Đưa ra giao diện đăng nhập |
| 3 | Nhập thông tin |  |
| 4 | Nhấn nút đăng nhập |  |
| 5 |  | website nhận thông tin, tiếp nhập thông tin và gửi Database để kiểm tra. Nếu hợp lệ thì hiển thị giao diện đọc truyện |
| **Luồng thay thế** |  |  | Nếu dữ liệu không hợp lệ thì hiển thị thông báo không hợp lệ |
| **Điều kiện sau** | Đăng nhập vào website | | |
| **Điều kiện thoát** | - Khi chức năng thực hiện thành công  - Khi người dùng nhấn đăng xuất | | |

**Bảng 3.1:** “Đăng nhập”

### *3.3.2 Đặc tả use-case “Thêm truyện”*

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use-case** | UC-themtruyen | | |
| **Tên Use-case** | Use-case thêm truyện | | |
| **Tác nhân** | Admin(quản lý) | | |
| **Mô tả** | Cho phép admin cập nhật hệ thống và thêm truyện | | |
| **Luồng sự kiện** |  | **Tác nhân** | **Hệ thống phản hồi** |
| 1 | Nhấn vào nút Admin |  |
| 2 |  | Hiện giao diện chứa danh sách truyện |
| 3 | Nhấn thêm truyện |  |
| 3 | Nhập thông tin truyện |  |
| 4 | Nhấn nút thêm |  |
| 5 |  | Gửi thông tin xuống Database sau đó lưu thông tin database |
| 6 |  | Load lại danh sách truyện |
| **Luồng thay thế** |  |  | Nếu nhập thiếu thông tin không thể lưu vào database |
| **Điều kiện sau** | Thêm truyện thành công và lưu truyện vào database | | |
| **Điều kiện thoát** | Khi chức năng thực hiện thành công  Khi người dùng chọn thoát | | |

**Bảng 3.2:** “Thêm món ăn”

### *3.3.3 Đặc tả use-case “Sửa truyện”*

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use-case** | UC-suatruyen | | |
| **Tên Use-case** | Use-case sửa truyện | | |
| **Tác nhân** | Admin(quản lý) | | |
| **Mô tả** | Cho phép admin cập nhật hệ thống và sửa truyện | | |
| **Luồng sự kiện** |  | **Tác nhân** | **Hệ thống phản hồi** |
| 1 | Nhấn vào nút Admin |  |
| 2 |  | Hiện giao diện chứa danh sách truyện |
| 3 | Nhấn nút sửa |  |
| 3 | Nhập thông tin truyện |  |
| 4 | Nhấn nút sửa |  |
| 5 |  | Gửi thông tin xuống Database sau đó lưu thông tin database |
| 6 |  | Load lại danh sách truyện |
| **Luồng thay thế** |  |  | Nếu nhập thiếu thông tin không thể lưu vào database |
| **Điều kiện sau** | Sửa truyện ăn thành công và lưu truyện vào database | | |
| **Điều kiện thoát** | Khi chức năng thực hiện thành công  Khi người dùng chọn thoát | | |

**Bảng 3.3:** “Sửa truyện”

### *3.3.4 Đặc tả use-case “xóa truyện”*

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use-case** | UC-xoatruyen | | |
| **Tên Use-case** | Use-case xóa truyện | | |
| **Tác nhân** | Admin(quản lý) | | |
| **Mô tả** | Cho phép admin cập nhật hệ thống và xóa truyện | | |
| **Luồng sự kiện** |  | **Tác nhân** | **Hệ thống phản hồi** |
| 1 | Nhấn vào nút admin |  |
| 2 |  | Hiện giao diện chứa danh sách truyện |
| 3 | Nhấn nút xóa |  |
| 4 |  | Gửi thông tin xuống Database sau đó lưu thông tin database |
| 5 |  | Load lại danh sách truyện |
| **Luồng thay thế** |  |  | Nếu truyện không tồn tại hiển thị lỗi |
| **Điều kiện sau** | xóa truyện thành công và lưu vào database | | |
| **Điều kiện thoát** | Khi chức năng thực hiện thành công  Khi người dùng chọn thoát | | |

**Bảng 3.4:** “Xóa truyện”

### *3.3.5 Đặc tả use-case “tìm kiếm”*

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use-case** | UC-timkiem | | |
| **Tên Use-case** | Use-case tìm kiếm | | |
| **Tác nhân** | Admin + người dùng | | |
| **Mô tả** | Cho phép người dùng tìm kiếm | | |
| **Luồng sự kiện** |  | **Tác nhân** | **Hệ thống phản hồi** |
| 1 | Điền thông tin cần tìm |  |
| 2 | Nhấn nút tìm kiếm |  |
| 3 |  | Hiện danh sách truyện cần tìm |
|  |  |  |
| **Luồng thay thế** |  |  | Nếu không tìm được truyện thì không hiển thị lên danh sách |
| **Điều kiện sau** | Hiển thị danh sách truyện cần tìm | | |
| **Điều kiện thoát** | Khi chức năng thực hiện thành công  Khi người dùng chọn thoát | | |

**Bảng 3.5:** “tìm kiếm”

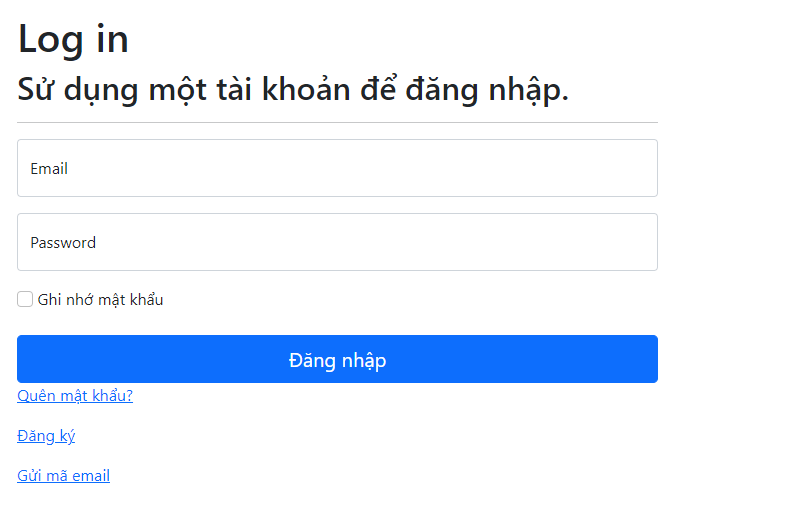
### *3.3.6 Đặc tả use-case “đọc truyện”*

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use-case** | UC-doctruyen | | |
| **Tên Use-case** | Use-case đọc truyện | | |
| **Tác nhân** | Admin + user | | |
| **Mô tả** | Cho phép admin và user xem nội dung truyện | | |
| **Luồng sự kiện** |  | **Tác nhân** | **Hệ thống phản hồi** |
| 1 | Chọn truyện |  |
| 2 |  | Hiển thị thông tin chi tiết truyện |
| 3 | Chọn chương truyện cần đọc |  |
| 4 |  | Hiện thị nội dung chương truyện |
| **Luồng thay thế** |  |  |  |
| **Điều kiện sau** |  | | |
| **Điều kiện thoát** | Khi chức năng thực hiện thành công  Khi người dùng chọn thoát | | |

**Bảng 3.6:** “Đọc truyện”

## 3.4.Giao diện

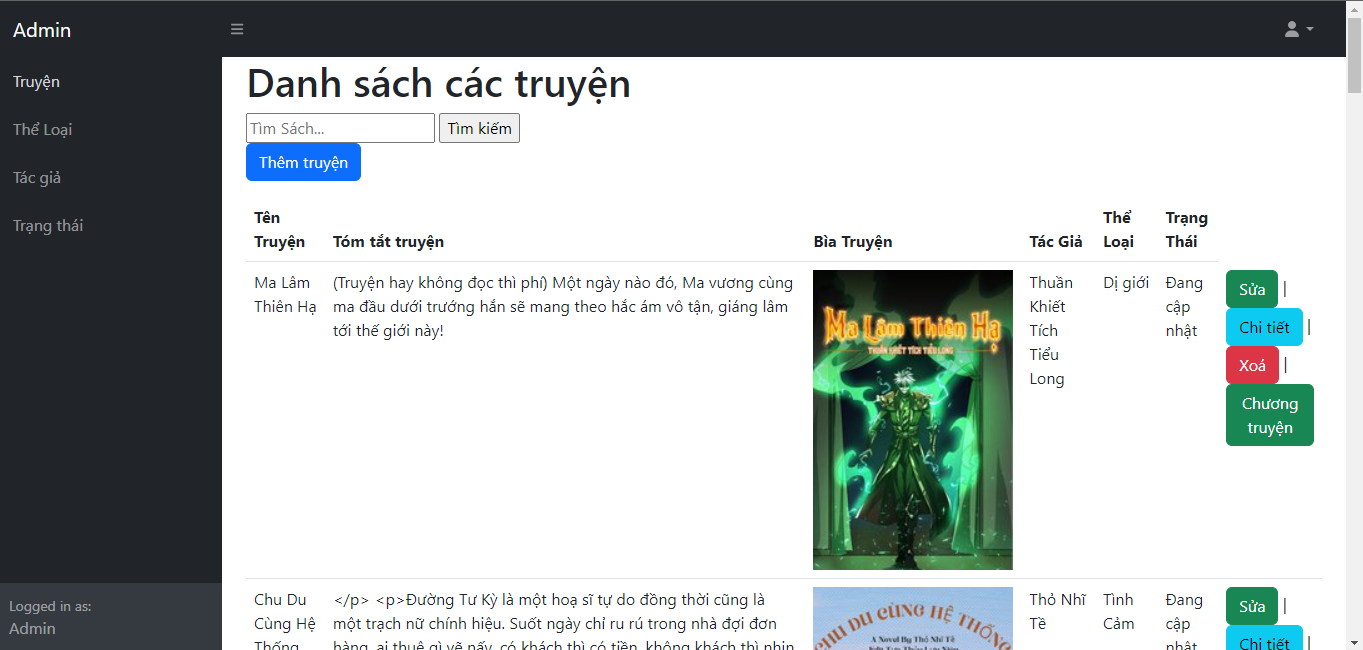
### *3.4.1. Đăng nhập*



### *3.4.2. Admin*

#### 3.4.2.1. Giao diện danh sách truyện:

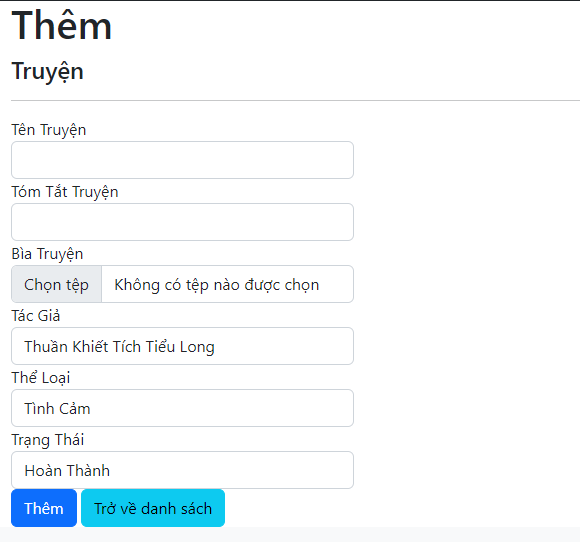
- Admin có thể quản lí những truyện đã được thêm ở đây



Hình 3.2: Giao diện danh sách truyện.

3.4.2.2. Giao diện thêm truyện:

-Admin có thể thêm truyện mới ở giao diện này



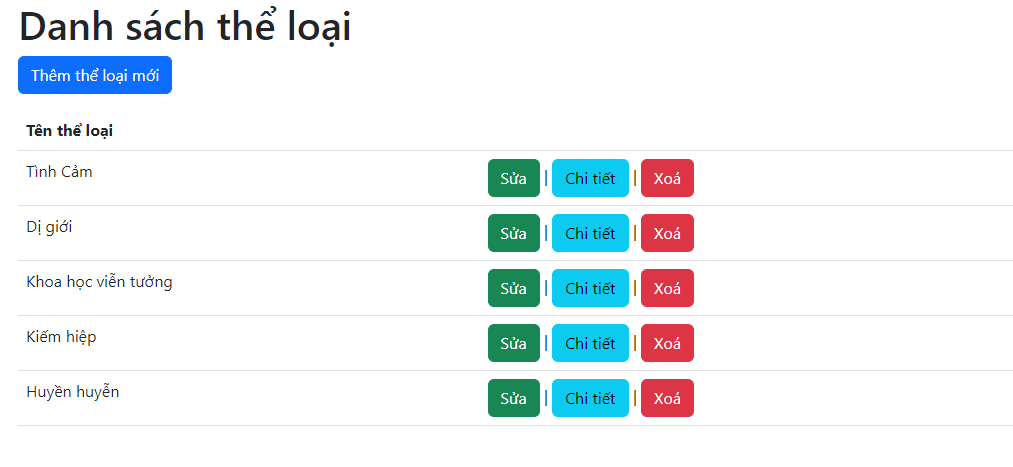
Hình 3.3: Giao diện thêm truyện.

3.4.2.3. Quản lý tác giả:



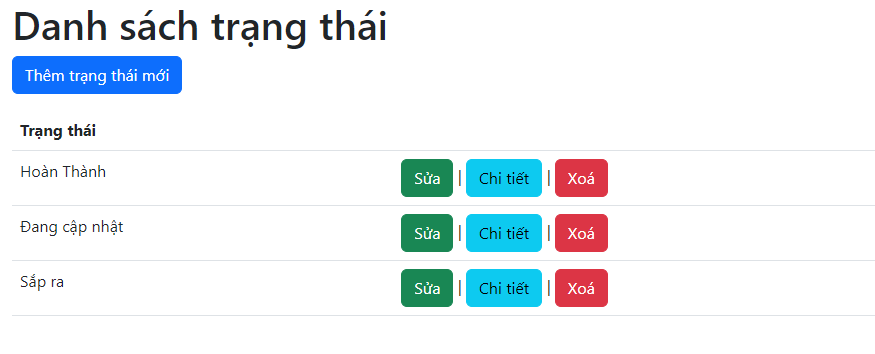
Hình 3.4: Quản lý tác giả

3.4.2.4. Quản lý thể loại:



Hình 3.5: Quản lý thể loại

3.4.2.5. Quản lý trạng thái:

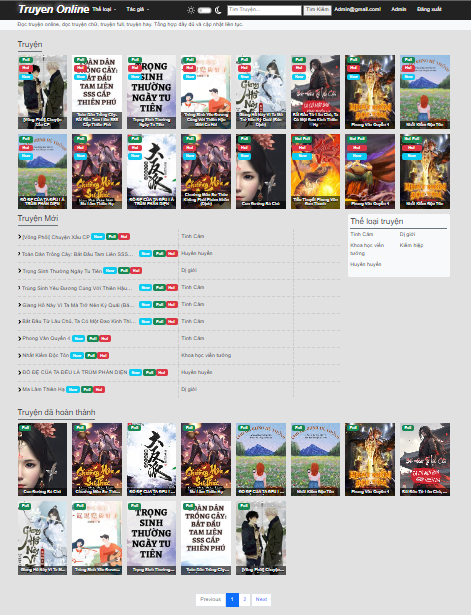


Hình 3.6: Quản lý trạng thái

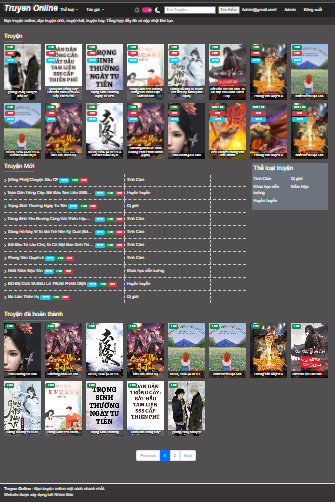
*3.4.3. Trang đọc truyện:*

3.4.3.1. Giao diện chính .

- Ở giao diện này người dùng có thể chọn một bộ truyện để đọc và có thể tìm kiếm truyện trên thanh tìm kiếm

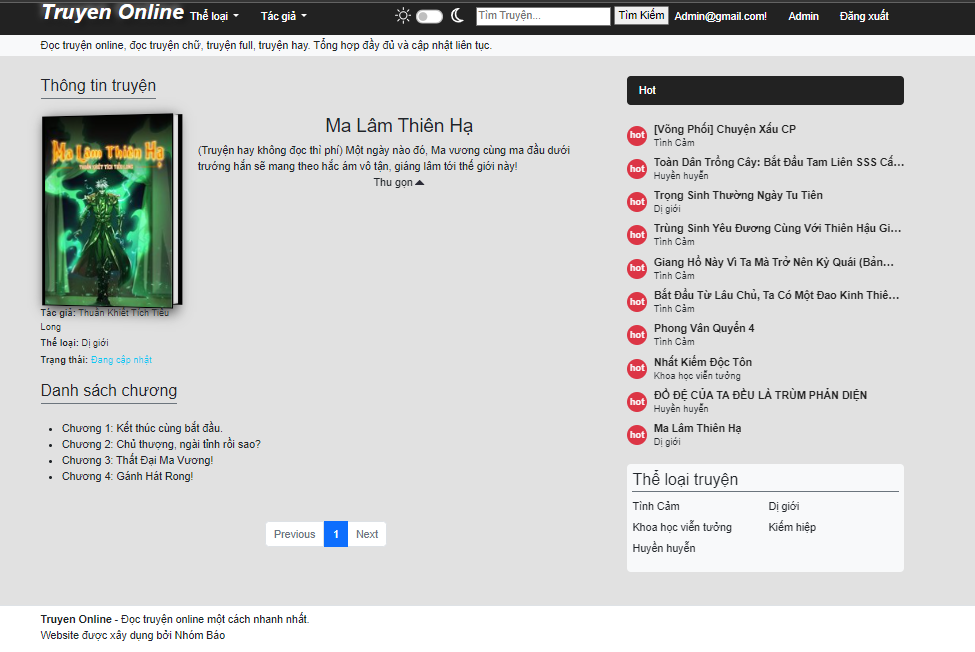


Hình 3.7: Giao diện chính website (sáng)

****

Hình 3.8: Giao diện chính website (tối)

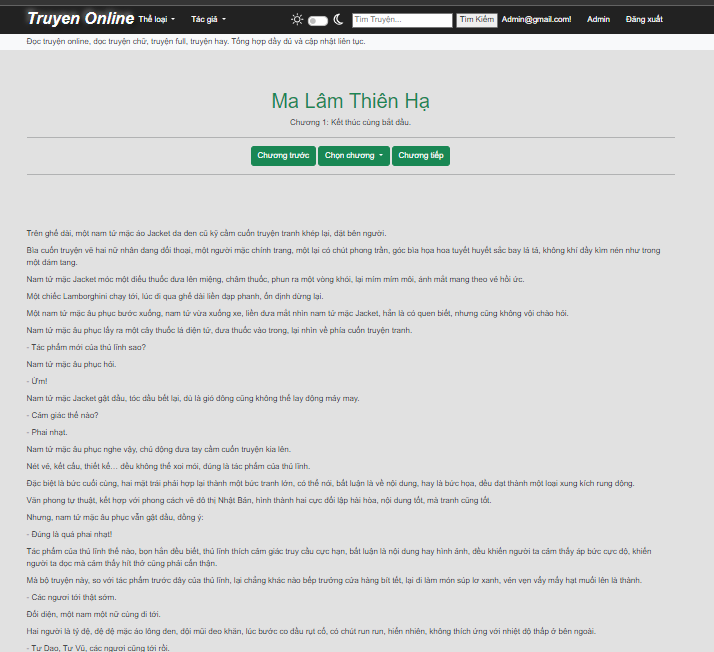
3.4.3.2. Giao diện thông tin truyện:



Hình 3.9: Giao diện thông tin truyện

3.4.3.4. Giao diện đọc truyện:

-Người dùng có thể đọc truyện ở giao diện này



Hình 3.10: Giao diện đọc truyện

# CHƯƠNG 4 : TỔNG KẾT

**4.1 Những vấn đề đã giải quyết**

+ Quản lý thông tin truyện

+ Quản lý thông tin tác giả

+ Quản lý thể loại

+ Quản lý trạng thái

+ Quản lý tài khoản

+ Đăng nhập

**4.2 Những vấn đề chưa giải quyết**

Cơ sở dữ liệu chưa tối ưu

Giao diện chưa tối ưu

**4.3 Hướng phát triển trên tương lai**

+Phát triển thêm về giao diện và chức năng các chức năng chưa dc tối ưu, chưa hoàn thành.

+Tối ưu hóa hình thức đăng nhập.