TRƯỜNG ĐẠI HỌC TRÀ VINH

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN MÔN HỌC**

**THIẾT KẾ GIAO DIỆN**

**WEBSITE KHO SÁCH GIÁO KHOA**

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn:**  Nguyễn Ngọc Đan Thanh | **Sinh viên thực hiện:**  Mã số sinh viên: 110122163  Họ tên: Châu Thanh Thiện  Lớp: DA22TTC |

***Trà Vinh, 21 tháng 5 năm 2024***

TRƯỜNG ĐẠI HỌC TRÀ VINH

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN MÔN HỌC**

**THIẾT KẾ GIAO DIỆN**

**WEBSITE KHO SÁCH GIÁO KHOA**

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn:**  Nguyễn Ngọc Đan Thanh | **Sinh viên thực hiện:**  Mã số sinh viên: 110122163  Họ tên: Châu Thanh Thiện  Lớp: DA22TTC |

***Trà Vinh, 21 tháng 4 năm 2024***

**LỜI CẢM ƠN**

Em xin chân thành cảm ơn cô Nguyễn Ngọc Đan Thanh. Cảm ơn cô đã cung cấp những kiến thức và thông tin quan trọng để hướng dẫn em hoàn thành thực hiện đề tài này trong môi trường tốt nhất.

Em đã cố gắng hết sức để tìm hiểu và thực hiện báo cáo này, tuy nhiên em thực hiện đề tài này sẽ không tránh được những sai sót. Em rất mong có thể nhận được những góp ý từ những cô và cũng mong cô sẽ thông cảm tạo điều kiện để em rút kinh nghiệm và hoàn thiện hơn.

Một lần nữa em xin chân thành cảm ơn.

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**….……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**MỤC LỤC**

[CHƯƠNG 1. Tổng quan về nội dung nghiên cứu 1](#_Toc30374)

[1.1. Quy trình phát triển website 1](#_Toc9874)

[1.2. Ngôn ngữ HTML 2](#_Toc32436)

[1.2.1. Giới thiệu, đặc điểm, vai trò và cấu trúc của HTML 2](#_Toc7439)

[1.2.2. Các thẻ để định dạng đoạn văn bản HTML. 4](#_Toc29289)

[1.2.3. Các thẻ để định dạng đoạn danh sách HTML. 5](#_Toc13064)

[1.2.4. Các thẻ để tổ chức nội dung trang HTML. 5](#_Toc31060)

[1.2.5. Các thẻ để định dạng văn bản HTML 6](#_Toc28517)

[1.2.6 Các thẻ xác định phần tử chung của HTML. 7](#_Toc19976)

[1.2.7 Liên kết trong HTML. 7](#_Toc23690)

[1.2.8 Bảng trong HTML. 7](#_Toc10992)

[1.2.9 Forms trong HTML. 8](#_Toc9838)

[1.2.10 Phương tiện trong HTML. 11](#_Toc17041)

[1.3. Bảng định kiểu CSS 12](#_Toc14913)

[1.3.1. Giới thiệu, ưu điểm, nhược điểm, cấu trúc CSS. 12](#_Toc18200)

[1.3.2. Bộ chọn CSS 14](#_Toc31823)

[1.3.3. Nền trong CSS. 15](#_Toc25385)

[1.3.4. Đường viền CSS. 15](#_Toc3595)

[1.3.5. Lề trong CSS. 17](#_Toc22086)

[1.3.6. Phông chữ, cỡ chữ CSS. 18](#_Toc11621)

[1.3.7. Liên kết trong CSS. 20](#_Toc14662)

[1.4. Ngôn ngữ JavaScript 21](#_Toc16344)

[1.4.1. Giới thiệu, ưu điểm, nhược điểm, nhiệm vụ, cấu trúc, JavaScript. 21](#_Toc21430)

[1.4.2. Các thành phần cơ bản trong JavaScript. 24](#_Toc28366)

[1.5. Kết chương 25](#_Toc28958)

[CHƯƠNG 2. Phân tích thiết kế hệ thống 26](#_Toc10324)

[2.1. Mô tả bài toán 26](#_Toc23927)

[2.1.1. Giới thiệu 26](#_Toc11820)

[2.1.2. Phân tích yêu cầu 26](#_Toc10847)

*[2.1.2.1](#_Toc28358)* [Chức năng: 26](#_Toc28358)

*[2.1.2.2](#_Toc10947)* [Giao diện: 27](#_Toc10947)

[2.1.3. Thiết kế và triển khai 27](#_Toc32421)

*[2.1.3.1](#_Toc16672)* [Header: 27](#_Toc16672)

*[2.1.3.2](#_Toc32311)* [Danh sách các cấp, lớp và sách giáo khoa: 27](#_Toc32311)

*[2.1.3.3](#_Toc15500)* [Footer: 28](#_Toc15500)

[2.1.4. Cải tiến 28](#_Toc22319)

[2.1.5. Kết luận 28](#_Toc8983)

[2.2. Đặc tả các yêu cầu 28](#_Toc21976)

[2.2.1. Yêu cầu chức năng 29](#_Toc31831)

*[2.2.1.1](#_Toc1546)* [Hiển thị các danh sách cấp trường, lớp và từng loại sách. 29](#_Toc1546)

*[2.2.1.2](#_Toc1293)* [Trang thông tin sách giáo khoa: 30](#_Toc1293)

*[2.2.1.3](#_Toc14494)* [Thanh điều hướng: 31](#_Toc14494)

*[2.2.1.4](#_Toc15693)* [Tìm kiếm: 32](#_Toc15693)

*[2.2.1.5](#_Toc32141)* [Đăng nhập 32](#_Toc32141)

*[2.2.1.6](#_Toc3216)* [Đăng ký tài khoản 33](#_Toc3216)

*[2.2.1.7](#_Toc47)* [Khách 35](#_Toc47)

[2.2.2. Yêu cầu phi chức năng 35](#_Toc21391)

*[2.2.2.1](#_Toc24102)* [Tính thẩm mỹ và trải nghiệm người dùng: 35](#_Toc24102)

*[2.2.2.2](#_Toc16872)* [Menu điều hướng: 36](#_Toc16872)

*[2.2.2.3](#_Toc5889)* [Hiển thị danh sách sách giáo khoa: 36](#_Toc5889)

[2.3. Thiết kế dữ liệu 39](#_Toc13771)

[2.3.1. Mô hình dữ liệu 39](#_Toc24989)

[2.3.2. Danh sách các thực thể và mối kết hợp 41](#_Toc24140)

[2.3.3. Chi tiết thực thể và mối kết hợp 41](#_Toc24000)

[2.4. Thiết kế xử lý 44](#_Toc23821)

[2.4.1. Sơ đồ DFD mức ngữ cảnh 44](#_Toc2968)

[2.4.2. Sơ đồ DFD mức 1 44](#_Toc19029)

[2.4.3. Sơ đồ DFD mức 2 (Theo từng chức năng) 44](#_Toc32753)

[2.5. Thiết kế giao diện 44](#_Toc22231)

[2.5.1. Sơ đồ hệ thống 45](#_Toc17252)

[2.5.2. Giao diện trang chủ người dùng 46](#_Toc24754)

*[2.5.2.1](#_Toc31702)* [Giao diện chức năng đăng kí 46](#_Toc31702)

*[2.5.2.2](#_Toc15731)* [Giao diện chức năng đăng nhặp 47](#_Toc15731)

[2.5.3. Kết chương 47](#_Toc15090)

[CHƯƠNG 3. Cài đặt thực nghiệm 48](#_Toc18718)

[3.1. Bộ dữ liệu thử nghiệm 48](#_Toc24977)

[3.2. Kết quả thực nghiệm 48](#_Toc2696)

[3.2.1. Chức năng 1 48](#_Toc18208)

[3.2.2. Chức năng 2 49](#_Toc32655)

[3.2.3. Chức năng 3 49](#_Toc3467)

[3.3. Kết chương 49](#_Toc30700)

[CHƯƠNG 4. Kết luận và hướng phát triển 51](#_Toc3386)

[4.1. Kết luận 51](#_Toc23472)

[4.2. Hướng phát triển 52](#_Toc25111)

**DANH MỤC HÌNH ẢNH**

[Hình 2.1 Sơ đồ hệ thống 45](#_Toc19514)

[Hình 2.2 Giao diện trang chủ của người dùng 46](#_Toc13398)

[Hình 2.3 Giao diện chức năng đăng kí 46](#_Toc4675)

[Hình 2.4 Giao diện chức năng đăng nhập 47](#_Toc7828)

**DANH MỤC BẢNG BIỂU**

[Bảng 2.1 Danh sách các thực thể và mối kết hợp 41](#_Toc30521)

[Bảng 2.2 Chi tiết thực thể User 41](#_Toc15042)

[Bảng 2.2 Chi tiết thực thể Đăng Ký 43](#_Toc1788)

[Bảng 2.2 Chi tiết thực thể Đăng Nhập 43](#_Toc17755)

# Tổng quan về nội dung nghiên cứu

# 

## Quy trình phát triển website

- Thu thập thông tin và phân tích yêu cầu: Trước hết cần hiểu rõ mục

tiêu của dự án, đối tượng sử dụng, yêu cầu kĩ thuật và cái yếu tố cần thiết khác.

- Lập kế hoạch (Planning): Đây là giai đoạn để xác định phạm vi dự án, tạo

lập lịch trình công việc, đặc tả chức năng và xác định nguồn lực cần thiết.

- Thiết kế (Design):

+ Wireframing: Tạo ra bản vẽ tĩnh hoặc dự án về cấu trúc của trang web mà không quan tâm đến thiết kế chi tiết.

+ UI/UX Design: Phát triển giao diện người dùng (UI) và trải nghiệm người dùng (UX) thông qua các bản thiết kế, màu sắc, font chữ, hình ảnh và các yếu tố khác

- Phát triển (Development):

+ Front-end Development: Xây dựng giao diện người dùng sử dụng các ngôn ngữ như HTML, CSS và JavaScript.

+ Back-end Development: Xây dựng phần mềm server và cơ sở dữ liệu sử dụng các công nghệ như PHP, Python, Node.js và MySQL, MongoDB,….

- Kiểm thử (Testing): Kiểm tra tính năng, tương tác và hiệu suất của trang

web trên nhiều thiết bị và trình duyệt khác nhau để đảm bảo chất lượng.

- Triển khai (Deployment): Đưa trang web lên môi trường sản phẩm, bao

gồm cài đặt và cấu hình các máy chủ, domain và các công việc liên quan.

- Bảo trì và nâng cấp (Maintenance and Upgrade): Liên tục duy trì và cập

nhật trang web để đảm bảo hoạt động mượt mà và phù hợp với yêu cầu của người dùng và công nghệ mới.

## Ngôn ngữ HTML

### Giới thiệu, đặc điểm, vai trò và cấu trúc của HTML

- Giới thiệu: HTML là viết tắt của HyperText Markup Language, hay còn gọi là Ngôn ngữ Đánh dấu Siêu văn bản. Đây là ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web trên World Wide Web. Nó hoạt động bằng cách sử dụng các thẻ để định dạng nội dung và cấu trúc của trang web.

- Đặc điểm của HTML:

+ Dễ học và sử dụng: HTML có cú pháp đơn giản và dễ hiểu, ngay cả đối với người mới bắt đầu.

+ Giao diện linh hoạt: HTML có thể được sử dụng để tạo ra các trang web với nhiều kiểu dáng và bố cục khác nhau.

+ Hỗ trợ đa nền tảng: HTML có thể được hiển thị trên tất cả các trình duyệt web và thiết bị khác nhau.

Phổ biến: HTML là ngôn ngữ lập trình web được sử dụng phổ biến nhất trên thế giới.

- Vai trò của HTML:

+ Xác định cấu trúc: HTML giúp xác định cấu trúc của trang web bằng cách sử dụng các thẻ (tags) như <html>, <head>, <body>, <header>, <footer>, <nav>, <section>, <article>, vv. Những thẻ này định nghĩa các phần khác nhau của trang web và giúp trình duyệt hiểu cách hiển thị nội dung.

+ Đánh dấu nội dung: HTML cho phép đánh dấu và phân loại nội dung trên trang web bằng cách sử dụng các thẻ như <p> (đoạn văn), <h1>, <h2>, ..., <h6> (tiêu đề), <ul>, <ol>, <li> (danh sách), <a> (liên kết), <img> (hình ảnh), v,v.

+ Tạo liên kết và tương tác: HTML cho phép tạo các liên kết (hyperlinks) bằng thẻ <a> và các tương tác người dùng khác như nút bấm, biểu mẫu, vv., thông qua các thẻ như <button>, <input>, <form>, vv.

+ Cung cấp metadata: HTML cho phép định nghĩa các metadata của trang web, bao gồm tiêu đề, mô tả, từ khóa, và thông tin về tác giả, bằng cách sử dụng thẻ <title>, <meta>, <link>, vv.

+ Hỗ trợ tạo nên cấu trúc sematic: HTML5 cung cấp nhiều thẻ sematic như <header>, <footer>, <article>, <section>, <nav>,... giúp tạo ra cấu trúc rõ ràng và dễ hiểu cho cả máy tính và con người.

- HTML là một phần quan trọng trong việc xây dựng trang web và là cơ sở cho việc sử dụng các ngôn ngữ và công nghệ web khác như CSS (Cascading Style Sheets) và JavaScript.

- Cấu trúc của HTML:

<!DOCTYPE html>

<html>

<head>

<title> Tiêu đề trang </title>

<meta charset = “utf-8”>

</head>

<body>

<p> </p>

</body>

</html>

<!DOCTYPE html> Xác định tài liệu là HTML5.

<html></html> Phần tử gốc của trang HTML.

<head></head> Chứa thông tin meta về trang HTML.

<meta> Khai báo văn bản được viết theo bộ kí tự utf-8.

<title></title> Chỉ định tiêu đề cho trang HTML.

<body></body> Xác định nội dung của tài liệu và là nới chứa tất cả nội dung hiểu thị.

### 1.2.2. Các thẻ để định dạng đoạn văn bản HTML.

- Thẻ <p>…</p> (Paragraphs) dùng để xác định một đoạn văn.

- Thẻ <hN>...</hN> (Headings) dùng để xác định tiêu đề quan trọng. N là số nguyên từ 1 đến 6.

- Thẻ <hr> (Horizontal Rule) dùng để xác định điểm ngắt theo chủ đề dưới dạng quy tắc ngang.

- Thẻ <br> (Line breaks) dùng để xác định ngắt dòng mà không bắt đầu một đoạn mới.

- Thẻ <wbr> (Word breaks) dùng để định nghĩa một điểm ngắt dòng tiềm năng nếu cần khi không đủ chổ hiển thị hết trên một dòng.

- Thẻ <blockquote>…</blockquote> (Long Quotations) dùng để xác định đoạn trích dẫn dài hoặc một phần sao chép từ nguồn khác thường được thụt lề.

- Thẻ <pre>…</pre> (Preformatted Text) dùng để xác định văn bản được định dạng sẵn. Nó giữ nguyên cả dấu cách và ngắt dòng.

- Thẻ <figure>…</figure> (Figures) dùng để xác định nội dung minh họa có thể là hình, video hoặc đoạn mã phải được xử lý, tham chiếu như một đơn vị độc lập. Thẻ <figcaption>…</figcaption> dùng để chú thích cho nội dung minh họa trên.

### 1.2.3. Các thẻ để định dạng đoạn danh sách HTML.

- Danh sách không có thứ tự (Unordered Lists) Thẻ <ul>...</ul> dùng để tạo danh sách, mỗi mục danh sách bắt đầu bằng thẻ <li>...</li>. Có thuộc tính type để định dạng ký hiệu mục danh sách (dise/ square/ circle).

- Danh sách có thứ tự (Ordered Lists) Thẻ <ol>...</ol> dùng để tạo danh sách, mỗi mục danh sách bắt đầu bằng thẻ <li>...</li>. Có thuộc tính type để định dạng ký hiệu mục danh sách (1/ I/ a/ A/ i) và thứ tự bắt đầu mục danh sách là một số N.

- Danh sách mô tả (Description Lists)

Thẻ <dl>...</dl> dùng để xác định danh sách mô tả.

Thẻ <dt>...</dt> dùng để xác định thuật ngữ (tên).

Thẻ <dd>...</dd> dùng để mô tả từng thuật ngữ.

### 1.2.4. Các thẻ để tổ chức nội dung trang HTML.

- Thẻ <main>…</main> là nơi chứa nội dung chính của trang hoặc app.

- Thẻ <header>...</header> dùng để xác định phần tiêu đề cho một tài liệu, một phần hoặc một trang.

- Thẻ <footer>...</footer> dùng để xác định phần chân cho một tài liệu, một phần hoặc một trang.

- Thẻ <section>...</section> dùng để xác định một hoặc một nhóm phần chính, nội dung chuyên đề.

- Thẻ <article>…</article> dùng để xác định thành phần khép kín, độc lập có thể tái sử dụng.

- Thẻ <aside>…</aside> dùng để xác định nội dung tách biệt có liên quan (như thanh bên).

- Thẻ <nav>…</nav> dùng để xác định tập hợp danh sách các liên kết điều hướng.

- Thẻ <address>…</address> dùng để xác định thông tin liên hệ của tác giả hoặc người bảo trì tài liệu.

### 1.2.5. Các thẻ để định dạng văn bản HTML

- Thẻ <em>…</em> (Emphasized Text) và <i>...</i> (Italic) dùng để nhấn mạnh (in nghiêng) văn bản.

- Thẻ <strong>...</strong> (Important text) và <b>...</b> (Bold) dùng để xác định văn bản quan trọng (in đậm).

- Thẻ <s>…</s> (Strike-Through) dùng để định dạng văn bản không chính xác được gạch giữa.

- Thẻ <u>...</u> (Underline) dùng để gạch chân văn bản.

- Thẻ <small>…</small> (Smaller Type Size) dùng để định dạng văn bản nhỏ hơn.

- Thẻ <q>...</q> (Short Quotations) dùng để định dạng trích dẫn ngắn.

- Thẻ <dfn>…</dfn> (Defining Terms) dùng để xác định một thuật ngữ có dạng in nghiêng.

- Thẻ <sub>…</sub> (Subscript) dùng để định dạng văn bản chỉ số dưới.

- Thẻ <sup>...</sup> (Superscript) dùng để định dạng văn bản chỉ số trên.

- Thẻ <data>…</data> (Machine-readable information) dùng để xác định văn bản thông tin mà máy tính có thể đọc được, có thuộc tính value.

- Thẻ <ins>…</ins> (Inserted Text) dùng để xác định văn bản đã bị xóa khỏi tài liệu có dạng gạch giữa.

- Thẻ <del>…</del> (Deleted Text) dùng để xác định văn bản đã được chèn vào tài liệu có dạng gạch dưới.

### 1.2.6 Các thẻ xác định phần tử chung của HTML.

- Thẻ <div>…</div> (Generic block-level element) được sử dụng làm vùng chứa cho các phần tử HTML khác, nhóm các phần của trang web lại với nhau. Tạo kiểu cho các khối nội dung và không có thuộc tính bắt buộc nhưng các thuộc tính style, class và id là phổ biến.

- Thẻ <span>…</span> (Generic inline element) được sử dụng làm một vùng chứa nội tuyến được sử dụng để đánh dấu một phần của văn bản hoặc một phần của tài liệu. Tạo kiểu cho các phần của văn bản và không có thuộc tính bắt buộc nhưng các thuộc tính style, class và id là phổ biến.

### 1.2.7 Liên kết trong HTML.

- Thẻ <a>…</a> (Hypertext Link) xác định một siêu liên kết, có thuộc tính href cho biết đích đến của liên kết.

- Thẻ <img> (Adds an inline image) sử dụng để nhúng hình ảnh vào trang web, có hai thuộc tính bắt buộc là src để chỉ định đường dẫn đến hình ảnh và alt để chỉ định văn bản thay thế cho hình ảnh. Ngoài ra, có thể thêm thuộc tính style hoặc width và height để chỉ định chiều rộng và chiều cao của hình ảnh.

- Thẻ <svg> (Scalable Vector Graphics) sử dụng để tự tạo hình ảnh đơn giản bằng lệnh.

### 1.2.8 Bảng trong HTML.

- Thẻ <table>…</table> (Tabular Content) dùng để tạo bảng.

- Thẻ <tr>…</tr> (Table Row) dùng để xác định một hàng của bảng.

- Thẻ <th>…</th> (Table Header) dùng để xác định tiêu đề của bảng.

- Thẻ <td>…</td> (Table Cell Data) dùng để xác định ô của bảng

- Thẻ <caption>…</caption> (Describing Table Content) dùng để làm chú thích (tiêu đề) cho bảng.

\*Lưu ý: Để thêm các đường viền cho bảng, tại thẻ <head>...</head> thêm thẻ <style>...</style> để định dạng các thẻ table, td, th bằng thuộc tính border.

- Thẻ <thead>…</thead> (Table Header Row Group) dùng để nhóm một hàng phần tiêu đề của bảng.

- Thẻ <tbody>…</tbody> (Table Body Row Group) dùng để nhóm một hàng phần nội dung của bảng.

- Thẻ <tfoot>…</tfoot> (Table Footer Row Group) dùng để nhóm một hàng phần chân của bảng.

- Thẻ <colgroup>…</colgroup> (A Semantically Related Group Of Columns) dùng để nhóm một cột liên quan với nhau về mặt ngữ nghĩa.

- Thẻ <col>…</col> (One Column In A Column Group) dùng để nhóm một cột của bảng.

### 1.2.9 Forms trong HTML.

- Thẻ <form>…</form> (Interactive Form) dùng để tạo biểu mẫu HTML cho người dùng nhập, có hai thuộc tính là action dùng để cung cấp vị trí (URL) của ứng dụng hoặc tập lệnh sẽ được sử dụng để xử lý biểu mẫu, phần đuôi có kiểu định dạng là .php và method chỉ định cách gửi thông tin đến máy chủ với hai phương thức là GET (mặc định) và POST.

- Thẻ <label> (Attaches Information To Form Controls) dùng để hiển thị nội dung yêu cầu nhập (nhãn) của biểu mẫu. Thuộc tính for để liên kết với thuộc tính id của thẻ <input>, giá trị của 2 thuộc tính liên kết này phải bằng nhau.

- Thẻ <input> (Text-Entry Controls) dùng để nhập thông tin vào biểu mẫu. Có ba thuộc tính là type để xác định kiểu dữ liệu đầu vào, id để liên kết với thuộc tính for của thẻ <label> và name để nhận giá trị được gửi đi.

+ Thẻ <input type="text"> (Single-line text-entry control) dùng để nhập thông tin văn bản một dòng.

+ Thẻ <input type="password"> (Password text control) dùng để nhập thông tin mật khẩu.

+ Thẻ <input type="submit"> (Submits the form data to the server) xác định nút để gửi dữ liệu biểu mẫu tới trình xử lý biểu mẫu.

+ Thẻ <input type="reset"> (Resets the form controls to their default settings) xác định nút đặt lại sẽ đặt lại tất cả các giá trị của biểu mẫu về giá trị mặc định của chúng.

+ Thẻ <input type="radio"> (Radio button) xác định một nút radio, cho phép người dùng chỉ chọn một trong số các lựa chọn giới hạn.

+ Thẻ <input type="checkbox"> (Checkbox button) xác định một hộp đánh dấu, cho phép người dùng không chọn hoặc chọn nhiều tùy chọn trong một số lựa chọn giới hạn.

+ Thẻ <input type="button"> (Custom input button) dùng để định nghĩa một nút.

+ <input type="image"> (Image buttons) xác định một hình ảnh là một nút gửi

+ Thẻ <input type="search"> (Search field) dùng để nhập một từ khóa tìm kiếm.

+ Thẻ <input type="email"> (Email address) dùng để nhập một địa chỉ email.

+ Thẻ <input type="tel"> (Telephone number) dùng để nhập một số điện thoại.

+ Thẻ <input type="url"> (Location (URL)) dùng để nhập một địa chỉ URL.

+ Thẻ <input type="color"> (Color picker) dùng để nhập màu được chọn.

+ Thẻ <input type="file"> (File selection field) xác định phần chọn tệp và một nút để tải tệp lên.

+ Thẻ <input type="hidden"> (Hidden control field) xác định giá trị đầu vào ẩn (không hiển thị cho người dùng).

+ Thẻ <input type="number"> (Number input) dùng để nhập số.

+ Thẻ <input type="range"> (Slider input) xác định một điều khiển để nhập một số có giá trị chính xác không quan trọng (như điều khiển thanh trượt).

+ Thẻ <input type="date"> (Date input control) cho phép người dùng nhập ngày.

+ Thẻ <input type="time"> (Time input control) cho phép người dùng chọn thời gian.

+ Thẻ <input type="datetime-local"> (Date/time control) cho phép người dùng nhập ngày và giờ.

+ Thẻ <input type="month"> (Specifies a month in a year) cho phép người dùng chọn một tháng và năm.

+ Thẻ <input type="week"> (Specifies a particular week in a year) cho phép người dùng chọn một tuần và năm.

- Thẻ <textarea>…</textarea> (Multiline Text-Entry Control) dùng để nhập thông tin một đoạn văn bản, có thuộc tính placeholder chỉ định một gợi ý ngắn mô tả giá trị mong đợi.

- Thẻ <select>…</select> (Menu Control) xác định danh sách thả xuống.

- Thẻ <option>…</option> (An Option Within A Menu) xác định một tùy chọn có thể được chọn trong danh sách thả xuống.

- Thẻ <optgroup>…</optgroup> (A Logical Grouping Of Options Within A Menu) dùng để tạo các nhóm tùy chọn mang tính khái niệm hợp lí, có thuộc tính label để đặt tên cho nhóm.

- Thẻ <fieldset>...</fieldset> (Groups Related Controls And Labels) dùng để nhóm dữ liệu liên quan trong một biểu mẫu.

- Thẻ <legend>…</legend> (Assigns A Caption To A Fieldset) xác định chú thích cho thẻ <fieldset>...</fieldset>.

- Thẻ <datalist>…</datalist> (Drop-down menu input) chỉ định danh sách các tùy chọn được xác định trước cho, người dùng sẽ thấy danh sách thả xuống gồm các tùy chọn được xác định trước khi họ nhập dữ liệu.

### 1.2.10 Phương tiện trong HTML.

- Thẻ <iframe>....</iframe> (A Nested Browsing Window) chỉ định khung nội tuyến. Khung nội tuyến được sử dụng để nhúng tài liệu khác trong tài liệu HTML hiện tại. Có thuộc tính src để chọn tài liệu để nhúng. Có thể thêm thuộc tính width và height để chọn kích cỡ khung nội tuyến.

- Thẻ <object>…</object> (Represents External Resource) dùng để nhúng một "tài liệu" nào đó vào trang web nhưng không dùng để nhúng các tập tin audio, video, trang web. Có hai thuộc tính là data để chỉ định tài liệu và type để cung cấp loại MIME của nó.

- Thẻ <param> (Parameters Of An Object) được sử dụng để xác định tham số cho thẻ <object>…</object>. Có hai thuộc tính là name để chỉ định tên tham số và value xác định giá trị của tham số.

- Thẻ <video>....</video> (Adds A Video Player To The Page) được sử dụng để nhúng một đoạn video trên trang web. Có hai thuộc tính thường có là control dùng để điều khiển video (phát, dừng, âm lượng) và thuộc tính autoplay để video tự động phát, thêm muted vào sau nếu muốn tắt tiếng. Có thể thêm các thuộc tính như width và height để chọn kích cỡ khung video hay poster để hiển thị hình ảnh trước phát video.

- Thẻ <source>....</source> (Source Media) dùng để chỉ định nguồn video cần nhúng vào tài liệu. Có hai thuộc tính là src để chọn đường dẫn nguồn và type để chọn loại MIME của nó.

- Thẻ <audio>…</audio> (Adds An Audio File To The Page) được sử dụng để nhúng một đoạn âm thanh trên trang web. và thuộc tính autoplay để audio tự động phát, thêm muted vào sau nếu muốn tắt tiếng. (Tương tự như video)

- Thẻ <track> (Adds Synchronized Text To Embedded Media) được sử dụng để cung cấp thêm văn bản được đồng bộ hóa với dòng thời gian của đoạn video hoặc âm thanh (phụ đề).

- Thẻ <canvas>...</canvas> (Adds A 2-D Dynamic Drawing Area) dùng để vẽ đồ họa một cách nhanh chóng thông qua JavaScript. Canvas là một vùng hình chữ nhật trên trang HTML. Theo mặc định, canvas không có đường viền và không có nội dung. Sau khi tạo vùng canvas hình chữ nhật, cần phải thêm JavaScript để thực hiện vẽ.

## Bảng định kiểu CSS

### Giới thiệu, ưu điểm, nhược điểm, cấu trúc CSS.

- Giới thiệu: CSS là viết tắt của Cascading Style Sheets, mô tả cách các phần tử HTML được hiển thị trên màn hình, giấy hoặc trên các phương tiện khác, CSS tiết kiệm rất nhiều công việc, nó có thể kiểm soát bố cục của nhiều trang web cùng một lúc. Các bảng định kiểu bên ngoài được lưu trữ trong các tệp CSS

- Ưu điểm:

+ Khả năng tiết kiệm thời gian: CSS cho phép bạn viết lần đầu và sử dụng lại nó trong các trang HTML tiếp theo. Ngoài ra, bạn hoàn toàn có thể định nghĩa phong cách của từng phần tử HTML và áp dụng nó cho nhiều trang web khác nếu cần.

+ CSS giúp khả năng tải trang nhanh chóng: Đặc điểm của CSS khi sử dụng là ít mã code hơn. Do đó, thời gian tải xuống nhanh chóng và tiết kiệm rất nhiều thời gian. Sau khi viết quy tắc CSS cho một thẻ cụ thể, bạn có thể áp dụng quy tắc đó cho mọi lần xuất hiện sau này của thẻ đó. Bạn hoàn toàn không cần thay đổi các thuộc tính của thẻ HTML.

+ Dễ dàng khi thực hiện bảo trì: Bạn chỉ cần thay đổi một phong cách và từng thành phần trên mọi trang web khi cần và CSS sẽ hỗ trợ cập nhật tự động.

+ Khả năng tương thích tốt: CSS cho khả năng tương thích với nhiều thiết bị, cùng với việc sử dụng cùng một tài liệu HTML song với nhiều cách hiển thị các phiên bản khác nhau của trang web trên mọi thiết bị di động đều được hỗ trợ tốt và có tính tương thích cao.

- Nhược điểm:

+ CSS hoạt động khác biệt cho từng trình duyệt: Khi các thay đổi được thực hiện đòi hỏi người thực hiện   phải kiểm tra tính tương thích trước khi CSS sẽ hiển thị cùng một hiệu ứng thay đổi trong tất cả các trình duyệt khác nhau. Điều này là do CSS hoạt động khác nhau trong từng trình duyệt cụ thể.

+ Khá khó khăn đối với người mới bắt đầu: Cùng với sự phát triển của công nghệ thì các cấp độ của CSS ngày càng trở nên khó nên việc tìm hiểu và nắm bắt để sử dụng càng trở nên khó khăn hơn.

+ Định dạng của web có khả năng rủi ro: CSS là một hệ thống mở, dựa trên văn bản nên rất dễ để truy cập. Điều này làm cho toàn bộ định dạng của trang web hoàn toàn có thể chịu tổn thương và gặp phải gián đoạn khi hành động hoặc xảy ra tai nạn với tệp. Lúc này hệ thống sẽ yêu cầu truy cập đọc hoặc ghi vào web dự định để có thể ghi đè được lên các thay đổI.

- Cú pháp: Bộ chọn trỏ đến phần tử HTML mà bạn muốn tạo kiểu. Khối khai báo chứa một hoặc nhiều khai báo được phân tách bằng dấu chấm phẩy. Mỗi khai báo bao gồm tên thuộc tính CSS và một giá trị, được phân tách bằng dấu hai chấm. Nhiều khai báo CSS được phân tách bằng dấu chấm phẩy và các khối khai báo được bao quanh bởi dấu ngoặc nhọn.

VD:

p {  
  color: red;  
  text-align: center;  
}

+ p là một bộ chọn trong CSS (nó trỏ đến phần tử HTML mà bạn muốn tạo kiểu: <p>).

+ color là một property và red là property value.

+ text-align là một property value và center là property value.

### Bộ chọn CSS

- Bộ chọn CSS được sử dụng để "tìm" (hoặc chọn) các thành phần HTML mà bạn muốn tạo kiểu. Chúng ta có thể chia bộ chọn CSS thành năm loại:

+ Bộ chọn đơn giản (chọn các thành phần dựa trên tên, id, lớp).

+ [Bộ chọn kết hợp](https://www.w3schools.com/css/css_combinators.asp) (chọn các phần tử dựa trên mối quan hệ cụ thể giữa chúng).

+ [Bộ chọn lớp giả](https://www.w3schools.com/css/css_pseudo_classes.asp) (chọn các phần tử dựa trên một trạng thái nhất định).

+ [Bộ chọn phần tử giả](https://www.w3schools.com/css/css_pseudo_elements.asp) (chọn và định kiểu một phần của phần tử).

+ [Bộ chọn thuộc tính](https://www.w3schools.com/css/css_attribute_selectors.asp) (chọn các thành phần dựa trên thuộc tính hoặc giá trị thuộc tính).

VD: Trong ví dụ này, phần tử <p> sẽ được tạo kiểu theo class="center" và class="large":

<p class="center large">This paragraph refers to two classes.</p>

### Nền trong CSS.

- Hình nền: Thuộc background-colortính chỉ định màu nền của một phần tử.

VD: Màu nền của trang được đặt như sau:

body {  
  background-color: lightblue;  
}

- Hình nền: Thuộc background-imagetính chỉ định một hình ảnh để sử dụng làm nền của một phần tử. Theo mặc định, hình ảnh được lặp lại để bao phủ toàn bộ phần tử.

VD: Đặt hình nền cho trang:

body {  
  background-image: url("paper.gif");  
}

### Đường viền CSS.

- Thuộc border-styletính chỉ định loại đường viền sẽ hiển thị. Các giá trị sau được cho phép:

+ dotted: Xác định đường viền chấm.

+ dashed: Xác định đường viền nét đứt.

+ solid: Xác định một đường viền vững chắc.

+ double: Xác định đường viền kép.

+ groove: Xác định đường viền có rãnh 3D. Hiệu ứng phụ thuộc vào giá trị màu viền.

+ ridge: Xác định đường viền có gờ 3D. Hiệu ứng phụ thuộc vào giá trị màu viền.

+ inset: Xác định đường viền chèn 3D. Hiệu ứng phụ thuộc vào giá trị màu viền.

+ outset: Xác định đường viền đầu 3D. Hiệu ứng phụ thuộc vào giá trị màu viền.

+ none: Xác định không có đường viền.

+ hidden: Xác định đường viền ẩn.

+ Thuộc border-style tính có thể có từ một đến bốn giá trị (cho viền trên, viền phải, viền dưới và viền trái).

VD: Trình diễn các kiểu đường viền khác nhau:

p.dotted {border-style: dotted;}  
 p.dashed {border-style: dashed;}  
 p.solid {border-style: solid;}  
 p.double {border-style: double;}  
 p.groove {border-style: groove;}  
 p.ridge {border-style: ridge;}  
 p.inset {border-style: inset;}  
 p.outset {border-style: outset;}  
 p.none {border-style: none;}  
 p.hidden {border-style: hidden;}  
 p.mix {border-style: dotted dashed solid double;}

### Lề trong CSS.

- Thuộc tính CSS marginđược sử dụng để tạo không gian xung quanh các phần tử, bên ngoài bất kỳ đường viền được xác định nào. Với CSS, bạn có toàn quyền kiểm soát lề. Có các thuộc tính để đặt lề cho mỗi cạnh của một phần tử (trên, phải, dưới và trái).

- CSS có các thuộc tính để xác định lề cho mỗi cạnh của một phần tử:

+ margin-top

+ margin-right

+ margin-bottom

+ margin-left

- Tất cả các thuộc tính lề có thể có các giá trị sau:

+ auto: trình duyệt tính toán mức ký quỹ

+ length: chỉ định lề bằng px, pt, cm, v.v.

+ %: chỉ định lề tính bằng % chiều rộng của phần tử chứa

+ inherit: chỉ định rằng lề phải được kế thừa từ phần tử cha

VD: Đặt các lề khác nhau cho cả bốn cạnh của phần tử <p>:

p {  
  margin-top: 100px;  
  margin-bottom: 100px;  
  margin-right: 150px;  
  margin-left: 80px;  
}

### Phông chữ, cỡ chữ CSS.

\* Phông chữ:

- Lựa chọn phông chữ là quan trọng, việc chọn đúng phông chữ có tác động rất lớn đến cách người đọc trải nghiệm một trang web. Phông chữ phù hợp có thể tạo ra bản sắc mạnh mẽ cho thương hiệu của bạn. Sử dụng phông chữ dễ đọc là điều quan trọng. Phông chữ thêm giá trị cho văn bản của bạn. Điều quan trọng nữa là chọn đúng màu sắc và kích thước văn bản cho phông chữ.

- Họ phông chữ chung: Trong CSS có năm họ phông chữ chung:

+ Phông chữ Serif có một nét nhỏ ở rìa mỗi chữ cái. Chúng tạo cảm giác trang trọng và sang trọng.

+ Phông chữ Sans-serif có đường nét rõ ràng (không có nét nhỏ kèm theo). Chúng tạo ra một cái nhìn hiện đại và tối giản.

+ Phông chữ Monospace ở đây tất cả các chữ cái đều có cùng chiều rộng cố định. Họ tạo ra một cái nhìn máy móc.

+ Phông chữ Cursive bắt chước chữ viết tay của con người.

+ Phông chữ Fantasy là phông chữ trang trí/vui tươi.

- Tất cả các tên phông chữ khác nhau đều thuộc về một trong các họ phông chữ chung.

VD: Chỉ định một số phông chữ khác nhau cho ba đoạn văn:

.p1 {  
  font-family: "Times New Roman", Times, serif;  
}  
  
.p2 {  
  font-family: Arial, Helvetica, sans-serif;  
}  
.p3 {  
  font-family: "Lucida Console", "Courier New", monospace;  
}

\* Cỡ chữ:

- Thuộc font-sizetính thiết lập kích thước của văn bản. Khả năng quản lý kích thước văn bản rất quan trọng trong thiết kế web. Tuy nhiên, bạn không nên sử dụng chức năng điều chỉnh cỡ chữ để đoạn văn trông giống tiêu đề, hoặc tiêu đề giống đoạn văn. Luôn sử dụng các thẻ HTML thích hợp, như <h1> - <h6> cho tiêu đề và <p> cho các đoạn văn. Giá trị kích thước phông chữ có thể là kích thước tuyệt đối hoặc tương đối.

- Kích thước tuyệt đối:

+ Đặt văn bản theo kích thước được chỉ định

+ Không cho phép người dùng thay đổi kích thước văn bản trong tất cả các trình duyệt (không tốt vì lý do khả năng truy cập)

+ Kích thước tuyệt đối rất hữu ích khi biết kích thước vật lý của đầu ra

- Kích thước tương đối:

+ Đặt kích thước tương ứng với các phần tử xung quanh

+ Cho phép người dùng thay đổi kích thước văn bản trong trình duyệt

- Lưu ý: Nếu bạn không chỉ định cỡ chữ, kích thước mặc định cho văn bản thông thường, như đoạn văn, là 16px (16px=1em).

VD: Đặt kích thước cho các đoạn văn bản sau:

h1 {  
  font-size: 40px;  
}

h2 {  
  font-size: 30px;  
}  
p {  
  font-size: 14px;  
}

### Liên kết trong CSS.

- Các liên kết có thể được tạo kiểu bằng bất kỳ thuộc tính CSS nào (ví dụ: color, font-family, background, v.v.).

VD:

a {  
  color: hotpink;  
}

- Ngoài ra, các liên kết có thể được tạo kiểu khác nhau tùy thuộc vào trạng thái của chúng. Bốn trạng thái liên kết là:

+ a:link- một liên kết bình thường, chưa được truy cập

+ a:visited- một liên kết người dùng đã truy cập

+ a:hover- một liên kết khi người dùng di chuột qua nó

+ a:active- một liên kết ngay khi nó được nhấp vào

- Khi đặt kiểu cho một số trạng thái liên kết, có một số quy tắc thứ tự:

+ a:hover PHẢI đứng sau a:link và a:visited

+ a:active PHẢI xuất hiện sau a:hover

VD:

/\* selected link \*/  
a:active {  
  color: blue;  
}

- Thuộc tính này text-decorationchủ yếu được sử dụng để xóa phần gạch chân khỏi các liên kết

VD:

a:link {  
  text-decoration: none;  
}

- Thuộc background-colortính có thể được sử dụng để chỉ định màu nền cho các liên kết.

VD:

a:link {  
  background-color: yellow;  
}

## Ngôn ngữ JavaScript

### Giới thiệu, ưu điểm, nhược điểm, nhiệm vụ, cấu trúc, JavaScript.

- Giới thiệu: JS là viết tắt của JavaScript, khi có JS bạn sẽ hiểu đó đang nói đến JavaScript. JavaScript là ngôn ngữ lập trình website phổ biến hiện nay, nó được tích hợp và nhúng vào HTML giúp website trở nên sống động hơn. JavaScript đóng vai trò như là một phần của trang web, thực thi cho phép Client-side script từ phía người dùng cũng như phía máy chủ (Nodejs) tạo ra các trang web động.

- Ưu điểm: Một số ưu điểm nổi bật của ngôn ngữ lập trình JS như sau:

Chương trình rất dễ học.

+ Những lỗi Javascript rất dễ để phát hiện, từ đó giúp bạn sửa lỗi một cách nhanh chóng hơn.

+ Những trình duyệt web có thể dịch thông qua HTML mà không cần sử dụng đến một compiler.

+ JS có thể hoạt động ở trên nhiều nền tảng và các trình duyệt web khác nhau.

+ Được các chuyên gia đánh giá là một loại ngôn ngữ lập trình nhẹ và nhanh hơn nhiều so với các ngôn ngữ lập trình khác.

+ JS còn có thể được gắn trên một số các element hoặc những events của các trang web.

+ Những website có sử dụng JS thì chúng sẽ giúp cho trang web đó có sự tương tác cũng như tăng thêm nhiều trải nghiệm mới cho người dùng.

+ Người dùng cũng có thể tận dụng JS với mục đích là để kiểm tra những input thay vì cách kiểm tra thủ công thông qua hoạt động truy xuất database.

+ Giao diện của ứng dụng phong phú với nhiều thành phần như Drag and Drop, Slider để cung cấp đến cho người dùng một Rich Interface (giao diện giàu tính năng).

+ Giúp thao tác với người dùng phía Client và tách biệt giữa các Client với nhau.

- Nhược điểm: Bên cạnh những ưu điểm kể trên thì JS vẫn có những nhược điểm riêng tương tự như các ngôn ngữ lập trình khác hiện nay. Cụ thể:

+ JS Code Snippet khá lớn.

+ JS dễ bị các hacker và scammer khai thác hơn.

+ JS cũng không có khả năng đa luồng hoặc đa dạng xử lý.

+ Có thể được dùng để thực thi những mã độc ở trên máy tính của người sử dụng.

+ Những thiết bị khác nhau có thể sẽ thực hiện JS khác nhau, từ đó dẫn đến sự không đồng nhất.

+ Vì tính bảo mật và an toàn nên các Client-Side JavaScript sẽ không cho phép đọc hoặc ghi các file.

+ JS không được hỗ trợ khi bạn sử dụng ở trong tình trạng thiết bị được kết nối mạng.

- Nhiệm vụ: xử lý những đối tượng HTML trên trình duyệt. Nó có thể can thiệp với các hành động như thêm, xóa, sửa các thuộc tính CSS và các thẻ HTML một cách dễ dàng. Hay nói cách khác, JavaScript là một ngôn ngữ lập trình trên trình duyệt ở phía client. Tuy nhiên, hiện nay với sự xuất hiện của NodeJS đã giúp cho JavaScript có thể làm việc ở backend. Bạn thử truy cập vào một số website trên internet thì sẽ thấy có những hiệu ứng slide, menu xổ xuống, các hình ảnh chạy qua chạy lại rất đẹp. Trong những năm gần đây, sự xuất hiện của các framework như NodeJS (chuyên code backend), ExpressJS (NodeJS framework), và nhiều thư viện frontend khác như Angular, jQuery, RactJS ra đời, giúp tạo ra một cơn sốt với từ khóa JavaScript Fullstack.

- Cấu trúc: JavaScript bao gồm các khối cơ bản sau:

+ Biến (Variables): Biến trong JavaScript được sử dụng để lưu trữ và tham chiếu đến dữ liệu. Các biến có thể là số, chuỗi, đối tượng hoặc các giá trị khác.

+ Hàm (Functions): Hàm là một khối mã được đặt tên và được sử dụng để thực thi một tập hợp các lệnh cụ thể. JavaScript hỗ trợ khai báo hàm và hàm vô danh.

+ Đối tượng (Objects): Trong JavaScript, đối tượng là một tập hợp của các thuộc tính (properties) và phương thức (methods). Các đối tượng giúp tổ chức và quản lý mã và dữ liệu một cách có tổ chức.

+ Điều kiện (Conditional Statements): Các câu lệnh điều kiện trong JavaScript cho phép bạn thực hiện mã khác nhau dựa trên điều kiện được xác định. Các câu lệnh điều kiện bao gồm if, else, else if, switch.

+ Vòng lặp (Loops): Vòng lặp cho phép bạn lặp lại một khối mã nhiều lần. JavaScript hỗ trợ các vòng lặp như for, while, do...while.

### Các thành phần cơ bản trong JavaScript.

- Biến (Variable)

var message = "Hello, world!";

=> Biến ‘message’ được khai báo để lưu trữ chuỗi “Hello, world”

- Hàm (Function)

function greet(name) {

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

}

=> Hàm ‘greet’ nhận một tham số ‘name’ và trả về chuỗi ‘Hello’

- Đối tượng (Objects)

var person = {

name: "Thien",

age: 20,

greet: function() {

return "Hello, my name is " + this.name + ".";

}

};

=> Đối tượng ‘person’ có thuộc tính ‘name’ và ‘age’, cùng một phương thức ‘greet’ để trả về ‘Hello, my name is”.

- Điều kiện (Conditinal Statements)

var x = 20;

if (x > 10) {

console.log("x is greater than 10");

} else {

console.log("x is less than or equal to 10");

}

=> Kiểm tra giá trị của biến ‘x’ và in ra thông báo tườn úng.

- Vòng lặp (Loops)

for (var i = 0; i < 10; i++) {

console.log("The value of i is: " + i);

}

=> Sử dụng vòng lặp ‘for’ để in ra giá trị của biến ‘i’ từ 0 đến 9.

## Kết chương

- Qua chương I ta đã tìm hiểu được ba loại ngôn ngữ thường được dùng để thiết kế một trang web: HTML, CSS, Javascript.

- Mỗi loại ngôn ngữ đều có những đặc điểm, nhiệm vụ, ưu điểm, nhược điểm và cấu trúc riêng.

- Để thiết kế một trang web ta có thể sử dụng một loại ngôn ngữ hoặc linh hoạt phối hợp nhiều loại ngôn ngữ với nhau làm cho việc thiết kế web trở nên đơn giản và nhanh chóng hơn.

# Phân tích thiết kế hệ thống

## Mô tả bài toán

Báo cáo: Xây dựng trang web kho sách giáo khoa.

### Giới thiệu

Trang web kho sách giáo khoa được xây dựng để học sinh, giáo viên, phụ huynh được tiếp cận với sách giáo khoa thuận lợi, trang web bao gồm những sách giáo khoa hiện hành do Nhà xuất bản Giáo dục Việt Nam xuất bản. Thầy cô, các bậc phụ huynh và các em học sinh có thể truy cập website “Kho sách giáo khoa” để xem sách giáo khoa miễn phí mà không cần đăng nhập hay mở tài khoản.

### Phân tích yêu cầu

#### Chức năng:

- Hiển thị sách giáo khoa hiện hành với ảnh bìa sách, tên sách và nhà xuất bản.

- Cho phép người dùng xem sách giáo khoa ở các lớp của từng cấp bằng cách nhấp chuột.

- Cung cấp liên kết đến các trang khác như Trang Chủ, Tìm Kiếm, Đăng Ký và Đăng Nhập.

#### Giao diện:

- Thiết kế giao diện đẹp mắt, gần gũi, dễ dàng sử dụng chỉ bằng cách nhấp chuột, thân thiện với mọi người dùng.

- Sử dụng ngôn ngữ HTML và CSS đảm bảo trang web hiển thị một cách thân thiện với nhiều loại thiết bị khác nhau.

### Thiết kế và triển khai

#### Header:

- Sử dụng thẻ <nav> để tạo các liên kết.

- Các liên kết bao gồm Trang Chủ, Tìm Kiếm, Đăng Ký, Đăng Nhập và một liên kết dành cho khách.

#### Danh sách các cấp, lớp và sách giáo khoa:

- Tạo các liên kết trang bằng cách nhấp chuột vào hình ảnh hoặc chữ bằng thẻ <a> để di chuyển đến trang kế tiếp hoặc quay về trang trước.

- Sử dụng bảng<table> để chia cột và hiển thị danh sách các các sách giáo khoa và nội dung giới thiệu.

- Mỗi ảnh bìa sách và nội dung giới thiệu được đặt trong thẻ <p>.

#### Footer:

- Tương tự như header nhưng có một số thay đổi về màu sắc và chứa thông tin liên hệ.

### Cải tiến

Bên cạnh những tính năng cơ bản, có thể thực hiện một số cải tiến sau:

- Tối ưu hóa hiệu suất: Đảm bảo trang web hoạt động mượt mà và nhanh chóng trên mọi thiết bị.

- Tính năng tìm kiếm và phân loại: Cho phép người dùng tìm kiếm sách giáo khoa theo tên, lớp hoặc cấp...

- Cung cấp các thông tin cần thiết về sách giáo khoa cho học sinh, giáo viên và phụ huynh.

- Đăng ký và đăng nhập: Cho phép người dùng tạo tài khoản để nhân thông báo về các thay đổi của sách giáo khoa một cách nhanh chóng.

### Kết luận

Trang web kho sách giáo khoa là trong web cần thiết gần gũi với tất cả mọi người dùng từ học sinh giáo viên đến các bậc phụ huynh các cấp. Trang web cung cấp đầy đủ các thông tin cơ bản cần thiết về sách giáo khoa cập nhật liên tục đảm bảo cung cấp cho người dùng những thay đổi về sách giáo khoa.

## Đặc tả các yêu cầu

### Yêu cầu chức năng

#### Hiển thị các danh sách cấp trường, lớp và từng loại sách.

- Trang web hiển thị danh sách các cấp trường, lớp và từng loại sách giáo khoa.

- Để tìm hiểu rõ hơn người dùng phải nhấp chuột vào hình ảnh hoặc chữ để đi và những liên kết trang nếu muốn tìm hiểu thông tin của từng loại sách.

- Ví dụ, trong đoạn mã HTML dưới đây, hiển thị các cấp trường người dùng có thể nhấp chuột vào hình ảnh hoặc chữ đề di chuyển vào trang kế tiếp.

<h2> <a href="SGKI.html">Sách giáo khoa cấp 1 </a></h2>

<p><a href="SGKI.html"><img src="img/1.png" alt="không tìm thấy" width="200" height="200"></a></p>

<h2><a href="SGKII.html">Sách giáo khoa cấp 2 </a></h2>

<p><a href="SGKII.html"><img src="img/2.png" alt="không tìm thấy" width="200" height="200"></a><br></p>

<h2> <a href="SGKIII.html">Sách giáo khoa cấp 3 </a></h2>

<p><a href="SGKIII.html"><img src="img/3.png" alt="không tìm thấy" width="200" height="200"></a><br></p>

#### Trang thông tin sách giáo khoa:

- Trang chi tiết thông tin sách giáo khoa cần cung cấp cho người dùng bao gồm tên sách, ảnh bìa, nhà xuất bản.

- Ví dụ, trong đoạn mã HTML dưới đây, các sách giáo khoa và nội dung được hiển thị trong một bảng (table). Mỗi cột chứa một ảnh bìa và thông tin sách.

<table>

<h1> Bộ Sách Giáo Khoa Lớp 1 </h1>

</tr>

<tr>

<td>

<img src="lop1/tiengviet1\_1.png" alt="Sách tiếng việt lớp 1 tập 1" width="400" height="500"></p>

</td>

<td>

<p>

Sách tiếng việt lớp 1 tập 1<br>

Nhà Xuất Bản Bộ Giáo Dục Và Đào Tạo

</p>

</td>

</tr>

<tr>

<td>

<p><img src="lop1/tiengviet1\_2.png" alt="Sách tiếng việt lớp 1 tập 2" width="400" height="500"></p>

</td>

<td>

<p>

Sách tiếng việt lớp 1 tập 2 <br>

Nhà Xuất Bản Bộ Giáo Dục Và Đào Tạo

</p>

</td>

</tr>

<tr>

</table>

#### Thanh điều hướng:

- Thanh điều hướng (nav) cần chứa các liên kết đến các trang khác như: Trang Chủ, Tìm Kiếm, Đăng Ký, và Đăng Nhập.

- Liên kết phải dẫn đến các trang tương ứng để người dùng có thể trải nghiệm tính năng một cách dễ dàng.

- Ví dụ, trong đoạn mã HTML dưới đây, các sách giáo khoa và nội dung được hiển thị trong một bảng (table). Mỗi cột chứa một ảnh bìa và thông tin sách.

<nav>

<ul>

<li><a href="khosachgiaokhoa.html">Trang Chủ</a> </li>

<li><a href="#">Tìm Kiếm</a></li>

<li><a href="dangky.html">Đăng Ký</a></li>

<li><a href="dangnhap.html">Đăng Nhập</a></li>

</ul>

</nav>

#### Tìm kiếm:

- Trang web cung cấp một ô tìm kiếm để người dùng có thể tìm kiếm tên sách, lớp hoặc cấp trường.

- Tìm kiếm nhanh chóng kết quả hiển thị rõ ràng.

Ví dụ: trong đoạn mã HTML dưới đây, cung cấp một ô tìm kiếm cho người dùng dễ dàng sử dụng.

<form action="/search" method="get">

<input type="text" placeholder="Tìm kiếm..." name="search">

<button type="submit">Tìm</button>

</form>

#### Đăng nhập

- Trang web hỗ trợ đăng nhập cho phép người dùng xác thực danh tính của mình trước khi truy cập vào một hệ thống hoặc ứng dụng. Quá trình này thường bao gồm việc cung cấp thông tin đăng nhập như tên người dùng và mật khẩu, sau đó hệ thống sẽ kiểm tra thông tin này để xác định xem người dùng có quyền truy cập hay không.

- Ví dụ:

<div id="wrapper">

<form action="" id="form-login">

<h1 class="form-heading"> Đăng nhập</h1>

<div class="form-group">

<i class="far fa-user"></i>

<input type="text" class="form-input" placeholder="Tên đăng nhập">

</div>

<div class="form-group">

<i class="fas fa-key"></i>

<input type="password" class="form-input" placeholder="Mật khẩu">

<div id="eye">

<i class="far fa-eye"></i>

</div>

</div>

<input type="submit" value="Đăng nhập" class="form-submit">

</form>

</div>

#### Đăng ký tài khoản

- Trang web hỗ trợ đăng ký tài khoản cho phép người dùng tạo hồ sơ cá nhân của mình trong trang web, ứng dụng hoặc hệ thống. Quá trình này thường bao gồm việc cung cấp thông tin cá nhân như tên, địa chỉ email, tài khoản, mật khẩu, và có thể là các thông tin khác như địa chỉ, số điện thoại, ngày sinh, v.v.

- Ví dụ:

<div id="wrapper">

<form action="" id="form-login">

<h1 class="form-heading"> Đăng ký</h1>

<div class="form-group">

<i class="far fa-user"></i>

<input type="text" class="form-input" placeholder="Tên tài khoản ">

</div>

<div class="form-group">

<i class="fas fa-key"></i>

<input type="password" class="form-input" placeholder="Mật khẩu">

<div id="eye">

<i class="far fa-eye"></i>

</div>

</div>

<div class="form-group">

<i class="fas fa-key"></i>

<input type="password" class="form-input" placeholder=" Xác nhận mật khẩu">

<div id="eye">

<i class="far fa-eye"></i>

</div>

</div>

<input type="submit" value="Đăng ký tài khoản " class="form-submit">

</form>

</div>

#### Khách

- Trang web còn hỗ trợ tính năng "khách" cho phép người dùng truy cập và trang web ở chế độ xem hoặc trạng thái mà người dùng không cần phải đăng nhập mà vẫn có thể truy cập nội dung hoặc chức năng cụ thể. Tính năng này cho phép người dùng truy cập các phần của trang web mà không cần tạo tài khoản hoặc đăng nhập.

- Ví dụ: trang web ở trang web kho sách giáo khoa người dùng có thể truy cập, tim kiếm, xem thông tin về các loại sách giáo, nhưng để tạo các lịch sử sách đã xem hoặc thêm bình luận, nhận xét, người dùng cần phải tạo tài khoản và đăng nhập tài khoản của mình khi truy cập trang web.

### Yêu cầu phi chức năng

Tính yêu cầu phi chức năng của website này bao gồm:

#### Tính thẩm mỹ và trải nghiệm người dùng:

- Sử dụng màu sắc đơn giản, hài hòa, phông chữ dễ nhìn.

- Giao diện được thiết kế sao cho dễ nhìn và thân thiện với người dùng.

- Ví dụ:

body {

background-color: #faf2e4;

margin: 0 10%;

font-family: sans-serif;

text-align: center;

}

#### Menu điều hướng:

- Menu điều hướng dễ dàng sử dụng bằng cách nhấp chuột vào các chữ (trang chủ, tìm kiếm, đăng kí, đăng nhập).

- Ví dụ:

<nav>

<ul>

<li><a href="khosachgiaokhoa.html">Trang Chủ</a> </li>

<li><a href="#">Tìm Kiếm</a></li>

<li><a href="dangky.html">Đăng Ký</a></li>

<li><a href="dangnhap.html">Đăng Nhập</a></li>

</ul>

</nav>

#### Hiển thị danh sách sách giáo khoa:

- Sắp xếp các các ảnh bìa sách giáo khoa và nội dung giới thiệu sách thành một bản có kích thước đồng đều.

- Ví dụ:

<table border="1" width="100%" height="100%">

<tr>

<nav>

<ul>

<li><a href="../khosachgiaokhoa.html">Trang Chủ</a> </li>

<li><a href="#">Tìm Kiếm</a></li>

</ul>

</nav>

</tr>

<tr>

<h1> Bộ Sách Giáo Khoa Lớp 1 </h1>

</tr>

<tr>

<td>

<img src="lop1/tiengviet1\_1.png" alt="Sách tiếng việt lớp 1 tập 1" width="400" height="500"></p>

</td>

<td>

<p>

Sách tiếng việt lớp 1 tập 1<br>

Nhà Xuất Bản Bộ Giáo Dục Và Đào Tạo

</p>

</td>

</tr>

<tr>

<td>

<p><img src="lop1/tiengviet1\_2.png" alt="Sách tiếng việt lớp 1 tập 2" width="400" height="500"></p>

</td>

<td>

<p>

Sách tiếng việt lớp 1 tập 2 <br>

Nhà Xuất Bản Bộ Giáo Dục Và Đào Tạo

</p>

</td>

</tr>

<tr>

<td>

<p><img src="lop1/toan1.png" alt="Sách toán lớp 1" width="400" height="500"></p>

</td>

<td>

<p>

Sách toán lớp 1 <br>

Nhà Xuất Bản Bộ Giáo Dục Và Đào Tạo

</p>

</td>

</tr>

<tr>

<td>

<p><img src="lop1/tienganh1.png" alt="Sách tiếng anh lớp 1" width="400" height="500"></p>

</td>

<td>

<p>

Sách tiếng anh lớp 1 <br>

Nhà Xuất Bản Bộ Giáo Dục Và Đào Tạo

</p>

</td>

</tr>

<tr>

<td>

<p><img src="lop1/amnhac1.png" alt="Sách âm nhạc lớp 1" width="400" height="500"></p>

</td>

<td>

<p>

Sách âm nhạc lớp 1 <br>

Nhà Xuất Bản Bộ Giáo Dục Và Đào Tạo

</p>

</td>

</tr>

<tr>

<td>

<p><img src="lop1/mithuat1.png" alt="Sách mĩ thuật lớp 1" width="400" height="500"></p>

</td>

<td>

<p>

Sách mĩ thuật lớp 1 <br>

Nhà Xuất Bản Bộ Giáo Dục Và Đào Tạo

</p>

</td>

</tr>

</table>

## Thiết kế dữ liệu

### Mô hình dữ liệu

- Khái niệm về mô hình ERD: Entity Relationship Diagram được biết đến là một dạng sơ đồ trực quan thực thể, đóng vai trò quan trọng trong việc quản lý và thiết kế cơ sở dữ liệu quan hệ

- Dưới đây là một số khái niệm cơ bản về ERD:

+ Thực thể (Entity): Đại diện cho một nhóm các đối tượng có các thuộc tính chung. Thực thể có thể là người, đồ vật, sự kiện hoặc khái niệm trừu tượng khác. Ví dụ: Người dùng, Cấp, Lớp, Sách giáo khoa.

+ Thuộc tính (Attribute): Là các đặc điểm hoặc thông tin mô tả về một thực thể. Ví dụ: ID, Tên, Email là các thuộc tính của thực thể Người Dùng.

+ Mối quan hệ (Relationship): Mô tả cách mà các thực thể liên kết với nhau trong cơ sở dữ liệu. Có ba loại mối quan hệ chính là 1-n (một-nhiều), n-n (nhiều-nhiều) và 1-1 (một-một).

+ Ví dụ: Mối quan hệ giữa Cấp và Lớp, mỗi cấp có thể có nhiều lớp (mối quan hệ 1-n).

+ Khóa (Key): Là một thuộc tính hoặc tập hợp các thuộc tính được sử dụng để định danh duy nhất cho mỗi bản ghi trong một thực thể. Khóa chính đại diện cho một cách duy nhất để xác định một bản ghi. Ví dụ: ID thường được sử dụng làm khóa chính cho các thực thể.

- Mô hình ERD: Là biểu đồ đồ họa mô tả các thực thể, thuộc tính và mối quan hệ giữa chúng trong cơ sở dữ liệu. Mô hình ERD giúp hiểu rõ cấu trúc của cơ sở dữ liệu và quan hệ giữa các thành phần của nó.

*\*\* Mô hình ERD*

### Danh sách các thực thể và mối kết hợp

Bảng 2.1 Danh sách các thực thể và mối kết hợp

|  |  |  |  |
| --- | --- | --- | --- |
| *STT* | *Tên thực thể/Mối kết hợp* | *Diễn giải* | *Ghi chú* |
| 1 | Người dùng | Thực thể đại diện cho người dùng của hệ thống | Định danh người dùng trên trang web |
| 2 | Sách giáo khoa | Thực thể đại diện cho sách giáo khoa | Các loại sách Giáo Khoa |
| 3 | Tìm kiếm | Mối kết hợp giữa người dùng và sách giáo khoa, thể hiện việc người dùng tìm kiếm sách giáo khoa | có thể bao gồm các thực thể phụ trợ như từ khóa tìm kiếm, kết quả tìm kiếm |
| 4 | Đăng ký | Mối kết hợp giữa người dùng và hệ thống, thể hiện quá trình đăng ký tài khoản | có thể bao gồm các thực thể phụ trợ như thông tin người dùng, quản lý tài khoản, v.v. |
| 5 | Đăng nhập | Mối kết hợp giữa người dùng và hệ thống, thể hiện quá trình đăng nhập vào hệ thống | có thể bao gồm các thực thể phụ trợ như thông tin người dùng, quản lý tài khoản, v.v. |

### Chi tiết thực thể và mối kết hợp

Thực thể

Mô tả

Chi tiết thực thể

Bảng 2.2 Chi tiết thực thể User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***STT*** | ***Tên thuộc tính*** | ***Diễn giải*** | ***Kiểu dữ liệu*** | ***Ràng buộc toàn vẹn*** |
| 1 | ID | Mã người dùng | INT | PRIMARY KEY |
| 2 | Username | Tên đăng nhập | VARCHAR(50) | NOT NULL |
| 3 | Password | Mật khẩu | VARCHAR(50) | NOT NULL |
| 4 | Email | Email | VARCHAR(100) | UNIQUE |
| 5 | Fullname | Họ và tên | VARCHAR(100) |  |
| 6 | Phone | Số điện thoại | VARCHAR(15) |  |

Bảng 2.3 Chi tiết thực thể Sách Giáo Khoa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***STT*** | ***Tên thuộc tính*** | ***Diễn giải*** | ***Kiểu dữ liệu*** | ***Ràng buộc toàn vẹn*** |
| 1 | ID | Mã sách | INT | PRIMARY KEY |
| 2 | Title | Tiêu đề | VARCHAR(255) | NOT NULL |
| 3 | Author | Tác giả | VARCHAR(100) |  |
| 4 | Publisher | Nhà xuất bản | VARCHAR(100) |  |
| 5 | Year | Năm xuất bản | INT |  |
| 6 | ISBN | Mã số quốc tế sách | VARCHAR(20) | UNIQUE |

Bảng 2.3 Chi tiết thực thể Tìm Kiếm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***STT*** | ***Tên thuộc tính*** | ***Diễn giải*** | ***Kiểu dữ liệu*** | ***Ràng buộc toàn vẹn*** |
| 1 | ID | Mã tìm kiếm | INT | PRIMARY KEY |
| 2 | User\_ID | Mã người dùng | INT | FOREIGN KEY |
| 3 | Keyword | Từ khóa tìm kiếm | VARCHAR(255) |  |
| 4 | Date | Ngày tìm kiếm | DATE |  |

Bảng 2.2 Chi tiết thực thể Đăng Ký

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***STT*** | ***Tên thuộc tính*** | ***Diễn giải*** | ***Kiểu dữ liệu*** | ***Ràng buộc toàn vẹn*** |
| 1 | ID | Mã đăng ký | INT | PRIMARY KEY |
| 2 | User\_ID | Mã người dùng | INT | FOREIGN KEY |
| 3 | Date | Ngày đăng ký | DATE |  |

Bảng 2.2 Chi tiết thực thể Đăng Nhập

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***STT*** | ***Tên thuộc tính*** | ***Diễn giải*** | ***Kiểu dữ liệu*** | ***Ràng buộc toàn vẹn*** |
| 1 | ID | Mã đăng nhập | INT | PRIMARY KEY |
| 2 | User\_ID | Mã người dùng | INT | FOREIGN KEY |
| 3 | Date | Ngày đăng nhập | DATE |  |

## Thiết kế xử lý

### Sơ đồ DFD mức ngữ cảnh

### Sơ đồ DFD mức 1

### Sơ đồ DFD mức 2 (Theo từng chức năng)

Diễn giải từng bước:

(1)

(2)

(3)

(4)

## Thiết kế giao diện

### Sơ đồ hệ thống

Kho sách giáo khoa

Trang chủ

Sách cấp 1

Tìm kiếm

Đăng kí

Sách cấp 2

Sách cấp 3

Đăng nhập

Sách lớp 10

Sách lớp 6

Sách lớp 1

Sách lớp 11

Sách lớp 7

Sách lớp 2

Sách lớp 12

Sách lớp 8

Sách lớp 3

Sách lớp 9

Sách lớp 4

Sách lớp 5

Hình 2.1 Sơ đồ hệ thống

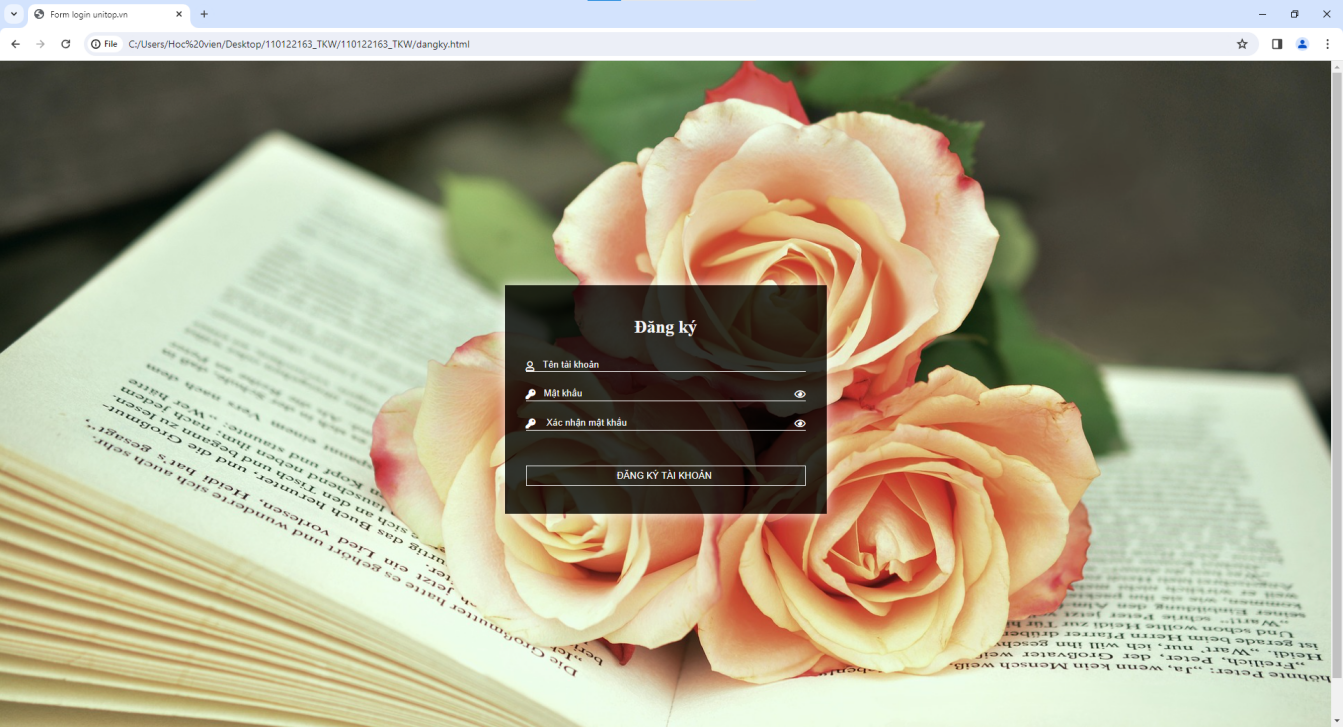
### Giao diện trang chủ người dùng



Hình 2.2 Giao diện trang chủ của người dùng

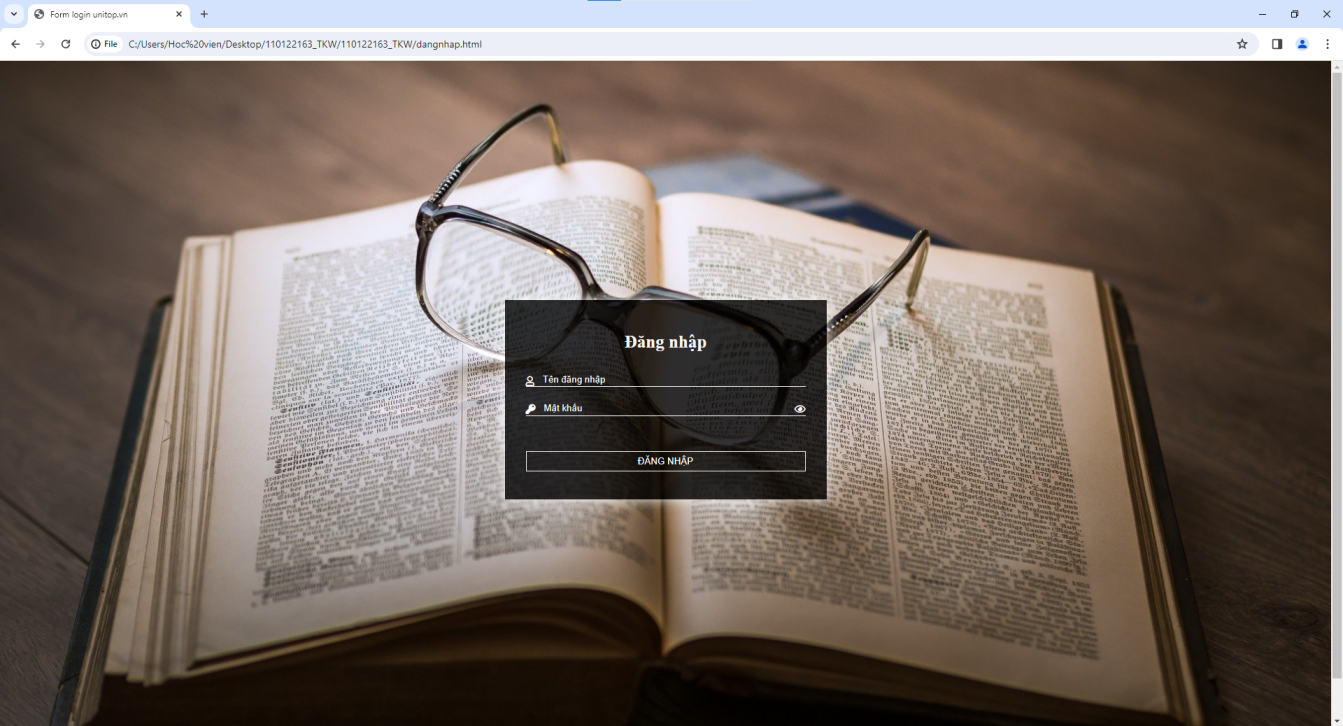
Mô tả chi tiết về màn hình bao gồm các thành phần và cách bố cục

#### Giao diện chức năng đăng kí



Hình 2.3 Giao diện chức năng đăng kí

#### Giao diện chức năng đăng nhặp



Hình 2.4 Giao diện chức năng đăng nhập

### Kết chương

-Tóm tắt kết quả: Phần thiết kế giao diện đã hoàn thành với các giao diện người dùng và quản trị rõ ràng và dễ sử dụng.

-Ý nghĩa: Thiết kế giao diện dễ sử dụng giúp cải thiện trải nghiệm người dùng và hiệu quả quản trị.

-Kết luận: Giao diện được thiết kế đảm bảo tính trực quan và hiệu quả trong việc thao tác và quản lý hệ thống, hỗ trợ mạnh mẽ cho việc triển khai hệ thống trong môi trường thực tế.

# Cài đặt thực nghiệm

## Bộ dữ liệu thử nghiệm

-Nguồn dữ liệu: Dữ liệu được thu thập từ các nguồn XYZ, bao gồm các tập dữ liệu ABC.

-Thuộc tính dữ liệu: Bộ dữ liệu gồm 1000 mẫu, với các thuộc tính như ID, tên, loại, thời gian, và giá trị.

-Tiền xử lý: Các bước tiền xử lý bao gồm làm sạch dữ liệu, chuẩn hóa giá trị, và phân chia dữ liệu thành các tập huấn luyện và kiểm tra.

## Kết quả thực nghiệm

### Chức năng 1

- Mô tả chức năng: Chức năng 1 là phân loại dữ liệu dựa trên thuật toán XYZ.

- Kết quả:

+ Độ chính xác: Đạt được độ chính xác 95% trên tập kiểm tra.

+ Biểu đồ: (Đính kèm biểu đồ ROC, Precision-Recall, v.v.)

+ Phân tích: Kết quả cho thấy mô hình có hiệu suất cao trong việc phân loại chính xác các mẫu dữ liệu.

### Chức năng 2

- Mô tả chức năng: Chức năng 2 là dự đoán giá trị dựa trên mô hình hồi quy.

- Kết quả:

+ Sai số trung bình: Sai số trung bình là 3.5%.

+ Biểu đồ: (Đính kèm biểu đồ dự đoán so với thực tế)

+ Phân tích: Mô hình hồi quy đã dự đoán gần đúng với giá trị thực tế, với sai số nhỏ.

### Chức năng 3

- Mô tả chức năng: Chức năng 3 là phân cụm dữ liệu.

- Kết quả:

+ Số cụm tối ưu: Xác định được 4 cụm chính.

+ Biểu đồ: (Đính kèm biểu đồ phân cụm)

+ Phân tích: Các cụm dữ liệu được phân chia rõ ràng, cho thấy sự khác biệt rõ rệt giữa các nhóm

## Kết chương

- Tóm tắt kết quả: Các thử nghiệm đã chứng minh hiệu quả của các chức năng được phát triển.

- Ý nghĩa: Kết quả cho thấy khả năng ứng dụng cao của hệ thống trong việc xử lý và phân tích dữ liệu.

- Kết luận: Những kết quả này hỗ trợ mạnh mẽ cho việc triển khai hệ thống trong môi trường thực tế.

# Kết luận và hướng phát triển

## Kết luận

Nghiên cứu này đã tập trung vào việc [mô tả ngắn gọn mục tiêu nghiên cứu], với mục tiêu chính là [nêu rõ mục tiêu nghiên cứu]. Qua quá trình nghiên cứu và phân tích, chúng tôi đã đạt được những kết quả quan trọng như sau:

- Kết quả chính: Nghiên cứu đã phát hiện ra rằng [mô tả các kết quả chính]. Các dữ liệu thu thập được chỉ ra rằng [mô tả chi tiết về dữ liệu và kết quả].

- Ý nghĩa của kết quả: Những kết quả này không chỉ khẳng định [một lý thuyết nào đó] mà còn mở ra những hướng đi mới trong lĩnh vực [lĩnh vực nghiên cứu]. Chúng cung cấp bằng chứng thuyết phục về [một hiện tượng, xu hướng hoặc giải pháp nào đó], từ đó giúp ích cho [ứng dụng cụ thể hoặc ngành công nghiệp cụ thể].

- Hạn chế của nghiên cứu: Tuy nhiên, nghiên cứu cũng gặp phải một số hạn chế như [mô tả hạn chế]. Những hạn chế này có thể ảnh hưởng đến [tính chính xác, độ tin cậy, hoặc phạm vi ứng dụng của kết quả], và do đó, cần được xem xét khi áp dụng kết quả vào thực tế.

Những kết luận này cho thấy rằng [nêu rõ ràng và súc tích những kết luận chính], từ đó xác định tầm quan trọng và đóng góp của nghiên cứu vào lĩnh vực [lĩnh vực nghiên cứu].

## Hướng phát triển

Để tiếp tục phát triển những kết quả đã đạt được, nghiên cứu tương lai có thể tập trung vào những hướng sau:

- Hướng nghiên cứu tiếp theo: Các nghiên cứu tương lai có thể mở rộng và đi sâu vào [một khía cạnh cụ thể nào đó] của đề tài này. Ví dụ, [nêu cụ thể một số nghiên cứu tiềm năng hoặc thí nghiệm bổ sung].

- Ứng dụng thực tiễn: Kết quả của nghiên cứu có thể được áp dụng vào [mô tả ứng dụng thực tiễn]. Chẳng hạn, trong ngành công nghiệp [ngành cụ thể], các phát hiện của chúng tôi có thể được sử dụng để [nêu ứng dụng cụ thể].

- Cải tiến phương pháp: Các phương pháp nghiên cứu hiện tại có thể được cải tiến bằng cách [mô tả cải tiến cụ thể]. Điều này sẽ giúp [mô tả lợi ích của cải tiến], nâng cao độ chính xác và độ tin cậy của các kết quả nghiên cứu tương lai.

- Phát triển công cụ hoặc tài liệu: Một hướng phát triển khác là nâng cấp và mở rộng các công cụ hoặc tài liệu đã được phát triển trong nghiên cứu này. Việc tạo ra các phiên bản tiếp theo với [mô tả cải tiến hoặc bổ sung] sẽ tăng cường khả năng ứng dụng và hiệu quả.

- Khả năng hợp tác liên ngành: Nghiên cứu này cũng mở ra cơ hội hợp tác với các lĩnh vực nghiên cứu khác như [lĩnh vực cụ thể] và ngành công nghiệp [ngành cụ thể]. Sự kết hợp này có thể tạo ra những giải pháp đa dạng và toàn diện hơn cho [mô tả vấn đề hoặc nhu cầu cụ thể].

Những hướng phát triển này không chỉ giúp tiếp tục mở rộng và hoàn thiện nghiên cứu hiện tại, mà còn tạo ra những giá trị mới cho lĩnh vực [lĩnh vực nghiên cứu] và các ứng dụng thực tiễn trong tương lai.

**DANH MỤC TÀI LIỆU THAM KHẢO**

[1] T. Berners-Lee, "Information Management: A Proposal," CERN, Geneva, Switzerland,1989.[Online].Available:[https://www.w3.org/History/1989/proposal.html](https://www.w3.org/History/1989/proposal.html" \t "https://chatgpt.com/c/_new)

[2] I. Sommerville, Software Engineering, 10th ed. Boston, MA, USA: Pearson, 2015.

[3] W3C, "HTML5: A vocabulary and associated APIs for HTML and XHTML," W3C Recommendation, Oct. 2014. [Online]. Available: [https://www.w3.org/TR/html5/](https://www.w3.org/TR/html5/" \t "https://chatgpt.com/c/_new)

[4] M. D. Hansen, S. D. Gray, and H. F. Wong, "CSS: The Definitive Guide," 4th ed. Sebastopol, CA, USA: O'Reilly Media, 2017.

[5] D. Flanagan, JavaScript: The Definitive Guide, 7th ed. Sebastopol, CA, USA: O'Reilly Media, 2020.

[6] M. Fowler, Patterns of Enterprise Application Architecture, Addison-Wesley Professional, 2002.

[7] R. Pressman and B. Maxim, Software Engineering: A Practitioner's Approach, 8th ed. New York, NY, USA: McGraw-Hill Education, 2014.

[8] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional, 1994.

[9] P. J. Deitel and H. M. Deitel, JavaScript for Programmers, Prentice Hall, 2009.

[10] J. Duckett, HTML & CSS: Design and Build Websites, John Wiley & Sons, 2011.

[11] ISO/IEC 27001:2013, "Information technology — Security techniques — Information security management systems — Requirements," International Organization for Standardization, 2013.

[12] K. Beck, Extreme Programming Explained: Embrace Change, 2nd ed. Addison-Wesley Professional, 2004.

[13] B. W. Boehm, "A Spiral Model of Software Development and Enhancement," ACM SIGSOFT Software Engineering Notes, vol. 11, no. 4, pp. 14-24, Aug. 1986.

[14] D. Crockford, JavaScript: The Good Parts, O'Reilly Media, 2008.

[15] H. M. Delete and P. J. Deitel, Internet & World Wide Web: How to Program, 5th ed. Pearson, 2011.

[16] A. T. S. Chan, W. N. Chan, and W. H. Ip, "A formal approach for mobile agent-oriented software engineering," IEEE Transactions on Software Engineering, vol. 29, no. 5, pp. 337-350, May 2003.

[17] P. C. Jorgensen, Software Testing: A Craftsman's Approach, 4th ed. CRC Press, 2013.

[18] ISO/IEC 9126-1:2001, "Software engineering — Product quality — Part 1: Quality model," International Organization for Standardization, 2001.

[19] W3Schools. (n.d.). Bootstrap Introduction. [Online]. Available: https://www.w3schools.com/bootstrap/bootstrap\_get\_started.asp.