**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN MÔN HỌC: THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM**

**TÊN ĐỀ TÀI: THIẾT KẾ WEDSIDE XEM PHIM TRỰC TUYẾN MIỄN PHÍ**

**Giảng viên hướng dẫn: ThS. Lê Văn Phong**

**Sinh viên thực hiện:**  **Đinh Văn Hoàng**

**Hà Nội, 2024**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN MÔN HỌC: THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM**

**TÊN ĐỀ TÀI: THIẾT KẾ WEDSIDE XEM PHIM TRỰC TUYẾN MIỄN PHÍ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020299 | Đinh Văn Hoàng | 09/11/2005 |  |  |

### 

### CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2

**Hà Nội, 2024**

**LỜI NÓI ĐẦU**

Ngày nay ứng dụng công nghệ thông tin và việc tin học hoá được xem là một trong những yếu tố mang tính quyết định trong hoạt động của các nước, tổ chức,...Nó đóng vai trò quan trọng và có thể tạo ra những bước đột phá mạnh mã. Cùng với sự phát triển không ngừng về kĩ thuật máy tính vàn mạng điện tử, công nghệ thông tin cũng được những công nghệ có đẳng cấp cao và lần lượt chinh phục hết đỉnh cao này đến đỉnh cao khác. Mạng internet là một trong những sản phẩm có giá trị hết sức lớn lao và ngày càng trở thành 1 công cụ không thể thiếu, là nền tảng chính cho sự truyền tải, trao đổi thông tin trên toàn cầu cũng như về nhu cầu giải trí của mọi người khá cao ở thời điểm hiện tại . Bằng internet, chúng ta đã thực thiện được nhiều công việc với tốc độ nhanh hơn và chi phí thấp hơn nhiều so với cách thức truyền thống. Chính điều này, nên em đã thực hiện dự án “THIẾT KẾ WEDSIDE XEM PHIM TRỰC TUYẾN MIỄN PHÍ” để giúp cho mọi người có những bộ phim để xem, được thư giãn, giải trí sau những giờ làm việc và học tập mệt mỏi.

MỤC LỤC

[CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM 10](#_Toc161745010)

[**1.1. Các khái niệm cơ bản 10**](#_Toc161745011)

[**1.2. Phân biệt Web tĩnh, Web động 10**](#_Toc161745012)

[**1.3. Một số thuật ngữ (Hosting, Tên miền-Domain Name-Web Server Name, HomePage, Web Site, WebServer, URL-Uniform Resource Location, Browser, …) 11**](#_Toc161745013)

[**1.4. Một số công cụ dùng thiết kế Web (Visual Studio code, Adobe Dreamweaver, …) 12**](#_Toc161745014)

[**Kết luận chương 1 12**](#_Toc161745015)

[CHƯƠNG II: HTML & HTML5 13](#_Toc161745016)

[**2.1. Tổng quan về HTML 13**](#_Toc161745017)

[**2.2. Cấu trúc tổng quát trang HTML 13**](#_Toc161745018)

[**2.3. Các thẻ HTML thông dụng 15**](#_Toc161745020)

[**2.4. Các thẻ tạo biểu mẫu (form) 18**](#_Toc161745021)

[**2.5. Một số thẻ HTML đặc biệt 18**](#_Toc161745022)

[**2.6. HTML5 19**](#_Toc161745023)

[**Kết luận chương 2 20**](#_Toc161745024)

[**3.1. CSS là gì? 21**](#_Toc161745025)

[**3.2. Cú pháp CSS 21**](#_Toc161745026)

[**3.3. Áp dụng CSS vào trang HTML 21**](#_Toc161745027)

[**3.4. Selectors 22**](#_Toc161745030)

[***3.4.1. Universal selector* 23**](#_Toc161745031)

[***3.4.2. Type selector* 25**](#_Toc161745035)

[***3.4.3. Identity selector* 25**](#_Toc161745036)

[***3.4.4. Class selector* 26**](#_Toc161745037)

[***3.4.5. Descendant selector* 26**](#_Toc161745038)

[***3.4.6. Child selector* 27**](#_Toc161745039)

[***3.4.7. Adjacent sibling selector* 27**](#_Toc161745040)

[***3.4.8. Attribute selector* 28**](#_Toc161745041)

[***3.4.9. Pseudo class selector* 29**](#_Toc161745042)

[***3.4.10. Group selector* 30**](#_Toc161745043)

[**3.5. Đơn vị đo lường CSS 31**](#_Toc161745044)

[**3.6. Kế thừa thuộc tính 31**](#_Toc161745045)

[**3.7. Các nhóm thuộc tính trong CSS 32**](#_Toc161745046)

[**3.8. Float & Clear 34**](#_Toc161745047)

[**3.9. Flex 35**](#_Toc161745048)

[**3.10. Grid 35**](#_Toc161745049)

[***3.11. CSS3* 36**](#_Toc161745050)

[**3.12. SCSS 42**](#_Toc161745051)

[**3.13. SASS 43**](#_Toc161745052)

[**Kết luận chương 3 44**](#_Toc161745053)

[CHƯƠNG 4: THIẾT KẾ WEDSIDE XEM PHIM TRỰC TUYẾN MIỄN PHÍ 45](#_Toc161745054)

[**4.1. Ý tưởng của Website 45**](#_Toc161745055)

[**4.2. Xây dựng bố cục của trang Web 45**](#_Toc161745056)

[**4.3. Thiết kế trang Web bằng HTML và CSS 46**](#_Toc161745057)

[**4.5. Kết quả đạt được 59**](#_Toc161745058)

[**Kết luận chương 4 61**](#_Toc161745059)

[**KẾT LUẬN 62**](#_Toc161745060)

**MỤC LỤC HÌNH ẢNH**

[Hình 0 14](#_Toc161291621)

[Hình 1 21](#_Toc161291622)

[Hình 2 22](#_Toc161291623)

[Hình 3 23](#_Toc161291624)

[Hình 4 24](#_Toc161291625)

[Hình 5 24](#_Toc161291626)

[Hình 6 24](#_Toc161291627)

[Hình 7 25](#_Toc161291628)

[Hình 8 25](#_Toc161291629)

[Hình 9 25](#_Toc161291630)

[Hình 10 26](#_Toc161291631)

[Hình 11 26](#_Toc161291632)

[Hình 12 26](#_Toc161291633)

[Hình 13 27](#_Toc161291634)

[Hình 14 27](#_Toc161291635)

[Hình 15 27](#_Toc161291636)

[Hình 16 28](#_Toc161291637)

[Hình 17 28](#_Toc161291638)

[Hình 18 29](#_Toc161291639)

[Hình 19 29](#_Toc161291640)

[Hình 20 30](#_Toc161291641)

[Hình 21 30](#_Toc161291642)

[Hình 22 31](#_Toc161291643)

[Hình 23 36](#_Toc161291644)

[Hình 24 37](#_Toc161291645)

[Hình 25 38](#_Toc161291646)

[Hình 26 38](#_Toc161291647)

[Hình 27 39](#_Toc161291648)

[Hình 28 41](#_Toc161291649)

[Hình 29 41](#_Toc161291650)

[Hình 30 42](#_Toc161291651)

[Hình 31 43](#_Toc161291652)

B**ẢNG CÁC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **TỪ VIẾT TẮT** | **VIẾT ĐẦY ĐỦ** |
| 1 | HTML | Hyper Text Markup Language |
| 2 | **CSS** | Cascading Style Sheets |

# CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM

## 1.1. Các khái niệm cơ bản

* Website:
* Mục đích: cung cấp thông tin, quảng bá thương hiệu, bán hàng, kết nối,...
* Phân loại: website tĩnh, website động, website bán hàng,...
* Là tập hợp các trang web được liên kết với nhau, truy cập thông qua Internet.
* Giao diện web:
* Là phần người dùng nhìn thấy và tương tác trên website.
* Bao gồm: bố cục, màu sắc, hình ảnh, typography,...
* Yêu cầu: đẹp mắt, dễ sử dụng, phù hợp với mục đích website.
* Trải nghiệm người dùng (UX):
* Là cảm nhận của người dùng khi sử dụng website.
* Mục tiêu: mang đến trải nghiệm tốt nhất, thu hút người dùng quay lại.
* Yếu tố ảnh hưởng: bố cục, nội dung, tốc độ tải trang,...
* Tối ưu hóa công cụ tìm kiếm (SEO):
* Là kỹ thuật giúp website hiển thị cao hơn trong kết quả tìm kiếm.
* Mục tiêu: thu hút nhiều truy cập tự nhiên hơn từ Google, Bing,...
* Kỹ thuật SEO: tối ưu hóa nội dung, cấu trúc website, backlink,...
* Một số nguyên tắc thiết kế web cơ bản:
* Xác định mục tiêu và đối tượng mục tiêu của website.
* Lên kế hoạch và thiết kế giao diện bắt mắt, dễ sử dụng.
* Sử dụng nội dung chất lượng, hữu ích cho người dùng.
* Đảm bảo website tối ưu hóa cho SEO và tốc độ tải trang nhanh.
* Thường xuyên cập nhật và bảo trì website.

## 1.2. Phân biệt Web tĩnh, Web động

* Web tĩnh (static website) được hiểu là một website được tạo ra hoàn toàn bằng việc sử dụng ngôn ngữ HTML (đường linh website thường có đuôi HTML). Sau khi download nội dung của những trang web tĩnh, trình duyệt sẽ tự động biên dịch và hiển thị nội dung, người dùng hoàn toàn không thể tương tác được với loại website này.
* Trái ngược với web tĩnh, web động (dynamic website) được xem là những website có khả năng tương tác với người dùng khá cao, có hệ thống quản lý nội dung, tập hợp các dữ liệu số hóa được tổ chức thành hệ cơ sở dữ liệu, được thể hiện dưới dạng văn bản, âm thanh, hình ảnh và có thể truy xuất dữ liệu.

## 1.3. Một số thuật ngữ (Hosting, Tên miền-Domain Name-Web Server Name, HomePage, Web Site, WebServer, URL-Uniform Resource Location, Browser, …)

* Hosting: Hosting (đưa lên web) là dịch vụ lưu trữ tất cả các tệp tin của trang web, chẳng hạn như tệp HTML, hình ảnh, video và mã JavaScript, trên một máy tính được kết nối với internet 24/7. Máy tính này được gọi là máy chủ web và được các công ty hosting cung cấp.
* Tên miền (Domain Name, Web Server Name): Tên miền là một địa chỉ web thân thiện với con người, giống như tên đường dẫn đến 1 nơi nào đó, nó dễ nhớ và dễ gõ hơn so với địa chỉ IP (dãy số) của máy chủ web.
* Trang chủ (Home Page): Trang chủ là trang đầu tiên người dùng nhìn thấy khi truy cập vào một trang web, thường chứa nội dung tóm tắt về trang web, các liên kết đến các phần khác của trang web và thông tin quan trọng khác.
* Trang web (Website): Trang web là một tập hợp các trang HTML được liên kết với nhau và truy cập được thông qua internet, có thể chứa văn bản, hình ảnh, video, âm nhạc, biểu mẫu và các yếu tố tương tác khác.
* Máy chủ web (Web Server): Máy chủ web là một máy tính được kết nối với internet và được cấu hình để lưu trữ các tệp tin của trang web và phục vụ chúng cho người dùng khi họ yêu cầu.
* URL (Uniform Resource Locator): URL là địa chỉ duy nhất của một tài nguyên trên internet, chẳng hạn như trang web, hình ảnh, video hoặc tệp tin khác, bao gồm tên miền, đường dẫn đến tệp tin và giao thức truyền thông (ví dụ: [đã xoá URL không hợp lệ]).
* Trình duyệt (Browser): Trình duyệt là một phần mềm cho phép người dùng truy cập và tương tác với các trang web, tải xuống các tệp tin HTML, CSS và JavaScript của trang web từ máy chủ web và hiển thị chúng trên màn hình của người dùng. Các trình duyệt phổ biến bao gồm Chrome, Firefox, Safari, Edge và Opera.

## 1.4. Một số công cụ dùng thiết kế Web (Visual Studio code, Adobe Dreamweaver, …)

* Các công cụ thiết kế web:
* Adobe Photoshop: phần mềm chỉnh sửa ảnh chuyên nghiệp.
* Visual Sutudio code:
* Adobe Illustrator: phần mềm thiết kế vector.
* Sketch: phần mềm thiết kế giao diện web.
* WordPress: hệ thống quản trị nội dung phổ biến.
* Bootstrap: framework CSS giúp xây dựng website nhanh chóng.

## Kết luận chương 1

Thiết kế web là một lĩnh vực rộng lớn bao gồm nhiều khái niệm quan trọng, từ giao diện người dùng và trải nghiệm người dùng đến bố cục, màu sắc và các yếu tố kỹ thuật như HTML, CSS. Đây cũng là lĩnh vực phức tạp nhưng cũng rất thú vị, việc nắm vững các khái niệm cơ bản của thiết kế web sẽ giúp tạo ra những website đẹp mắt, dễ sử dụng và hiệu quả.

# CHƯƠNG II: HTML & HTML5

**(**Hyper Text Markup Language**)**

## 2.1. Tổng quan về HTML

* HTML là viết tắt của HyperText Markup Language, là ngôn ngữ đánh dấu siêu văn bản được sử dụng để tạo ra các trang web, là nền tảng cơ bản của World Wide Web, cho phép định nghĩa cấu trúc và nội dung của các trang web.

## 2.2. Cấu trúc tổng quát trang HTML

* Cách HTML mô tả cấu trúc của một website:

<html>

<head>

﻿ <title>Đây là nhan đề của một bài viết</title>

</head>﻿

<body>

  <h1>Đây là tiêu đề chính</h1>

  <p>Đây là đoạn văn để giới thiệu nội dung phần còn lại của trang, nếu nội dung dài thì có thể chia thành nhiều tiêu đề phụ.<p>

  <h2>Đây là tiêu đề phụ</h2>

  <p>Một bài viết nên có 1 vài tiêu đề phụ, điều đó giúp co cấu trúc bài viết được rõ ràng, người đọc dễ hiểu hơn.</p>

  <h2>Đây là một tiêu đề phục khác cùng cấp với tiêu đề phụ bên trên</h2>

</body>

﻿</html>

* Mã HTML được tạo thành từ các ký tự nằm bên trong các dấu ngoặc nhọn <>. Chúng được gọi là các phần tử HTML (HTML elements). Các phần tử này thường được tạo thành từ hai thẻ: thẻ mở và thẻ đóng. (Thẻ đóng có thêm một dấu gạch chéo trong đó ví dụ thẻ </p>). Cũng có thẻ không có đầy đủ cả thẻ mở và thẻ đóng ví dụ: <img /> có / trước >.
* Element Trong HTML :



Hình 0

* Các thẻ ở đây hoạt động giống như các vùng chứa, nó chứa thông tin nằm giữa 2 thẻ mở và đóng.
* Các ký tự trong ngoặc cho ta biết loại thẻ, mục đích của thẻ. Ví dụ: trong các thẻ ở trên p là viết tắt của đoạn văn (paragraph).
* Thẻ mở <p> được cấu tạo bởi dấu nhỏ hơn nằm bên trái, tiếp đến là kí tự "p" và cuối cùng là dấu lớn hơn.
* Thẻ đóng </p> được cấu tạo cơ bản giống thẻ mở của nó nhưng nó thêm "/" vào trước kí tự "p".
* Body, Head & Title :

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Tiêu đề/Nhan đề bài viết </title>

</head>

<body>

Nội dung của trang sẽ nằm ở đây

</body>

</html>

* <!DOCTYPE html>  Khai báo đây là một file có định dạng là HTML5 để giúp trình duyệt biết đang dùng phiên bản html bao nhiêu, có cũng được, không có cũng được nhưng sẽ bị hạn chế nhiều thẻ mới, thuộc tính trong thẻ mà trình duyệt không phát hiện được.
* Thẻ <html> Cấu trúc file html.
* Thẻ <body> Nằm sau thẻ  <head>, có thẻ đóng </body> nằm trước </html>. Body là nơi chứa những gì mà mọi người sẽ thấy trên trang như thanh menu, banner, quảng cáo.
* Thẻ <head> Nằm sau thẻ <html>, có thẻ đóng </head>, nó chứa những thẻ khai báo thông tin cho trang như tiêu đề, mô tả, bảng mã ký tự. head không hiển thị nội dung những gì nó chứa đựng ra ngoài.
* Thuộc tính (attribute) charset nằm trong thẻ meta có nhiệm vụ khai báo bảng mã, ngôn ngữ tiếng việt nên dùng UTF-8.
* Thẻ <title> dùng để khai báo tiêu đề của trang, và gần như đây là bắt buộc phải có trong một file html.

## 2.3. Các thẻ HTML thông dụng

* Các thẻ xử lý văn bản :
* Thẻ heading (<h1> - <h6>): Các thẻ này được sử dụng để định nghĩa tiêu đề của một trang web hoặc phần của trang web. Số càng nhỏ thì cấp độ tiêu đề càng cao.
* Thẻ đoạn văn (<p>): Thẻ này được sử dụng để định nghĩa một đoạn văn bản.
* Thẻ dòng mới (<br>): Thẻ này được sử dụng để tạo một dòng mới trong đoạn văn bản.
* Thẻ đường kẻ ngang (<hr>): Thẻ này được sử dụng để tạo một đường kẻ ngang trong đoạn văn bản.
* Thẻ gạch đầu dòng (<ul> - <li>): Các thẻ này được sử dụng để tạo danh sách gạch đầu dòng. Thẻ <ul> định nghĩa danh sách, thẻ <li> định nghĩa mỗi mục trong danh sách.
* Thẻ số thứ tự (<ol> - <li>): Các thẻ này được sử dụng để tạo danh sách số thứ tự. Thẻ <ol> định nghĩa danh sách, thẻ <li> định nghĩa mỗi mục trong danh sách.
* Thẻ định dạng văn bản (<b>, <i>, <u>, <strong>, <em>): Các thẻ này được sử dụng để định dạng văn bản. Thẻ <b> được sử dụng để tạo văn bản in đậm, thẻ <i> được sử dụng để tạo văn bản in nghiêng, thẻ <u> được sử dụng để tạo văn bản gạch chân, thẻ <strong> được sử dụng để tạo văn bản nhấn mạnh, thẻ <em> được sử dụng để tạo văn bản nhấn mạnh.
* Các thẻ tạo bảng:
* Thẻ bảng (<table>): Thẻ này được sử dụng để xác định một bảng.
* Thẻ hàng (<tr>): Thẻ này được sử dụng để xác định một hàng trong bảng.
* Thẻ ô (<td>): Thẻ này được sử dụng để xác định một ô trong bảng.
* Thẻ tiêu đề cột (<th>): Thẻ này được sử dụng để xác định một ô tiêu đề trong bảng.
* Thẻ liên kết :
* Thẻ <a>: Thẻ liên kết (href là thuộc tính bắt buộc, xác định đường dẫn đến liên kết).
* Các thẻ đa phương tiện :
* Thẻ hình ảnh (<img>): Thẻ này được sử dụng để chèn hình ảnh vào trang web.
* src: Xác định đường dẫn đến hình ảnh.
* alt: Xác định văn bản thay thế cho hình ảnh. Văn bản này sẽ được hiển thị nếu trình duyệt không thể tải hình ảnh.
* width: Chiều rộng của hình ảnh.
* height: Chiều cao của hình ảnh.
* Thẻ âm thanh (<audio>): Thẻ này được sử dụng để chèn âm thanh vào trang web.
* src: Xác định đường dẫn đến âm thanh.
* controls: Xác định xem trình duyệt có hiển thị các điều khiển âm thanh hay không.
* autoplay: Xác định liệu âm thanh có tự động phát khi trang web được tải hay không.
* loop: Xác định liệu âm thanh có lặp lại sau khi phát xong hay không.
* Thẻ video (<video>): Thẻ này được sử dụng để chèn video vào trang web.
* src: Xác định đường dẫn đến video.
* controls: Xác định xem trình duyệt có hiển thị các điều khiển video hay không.
* autoplay: Xác định liệu video có tự động phát khi trang web được tải hay không.
* loop: Xác định liệu video có lặp lại sau khi phát xong hay không.

## 2.4. Các thẻ tạo biểu mẫu (form)

* Thẻ <form> là thẻ bắt buộc để tạo một biểu mẫu.
* action: Xác định URL của trang web xử lý dữ liệu từ biểu mẫu.
* method: Xác định phương thức HTTP được sử dụng để gửi dữ liệu từ biểu mẫu. Các giá trị có thể có là GET hoặc POST.
* enctype: Xác định cách mã hóa dữ liệu được gửi từ biểu mẫu. Giá trị mặc định là application/x-www-form-urlencoded.
* Thẻ <input> được sử dụng để tạo các phần tử đầu vào cho biểu mẫu. Thẻ này có nhiều loại khác nhau, mỗi loại được sử dụng để tạo một loại phần tử đầu vào khác nhau.
* text: Tạo một hộp văn bản để người dùng nhập văn bản.
* password: Tạo một hộp văn bản ẩn văn bản được nhập bởi người dùng.
* checkbox: Tạo một hộp kiểm cho phép người dùng chọn hoặc bỏ chọn một tùy chọn.
* radio: Tạo một nút radio cho phép người dùng chọn một tùy chọn duy nhất.
* submit: Tạo một nút gửi để người dùng gửi dữ liệu từ biểu mẫu.
* reset: Tạo một nút đặt lại để người dùng đặt lại dữ liệu trong biểu mẫu về trạng thái ban đầu.
* Thẻ <label> được sử dụng để liên kết một phần tử đầu vào với một nhãn. Nhãn sẽ giúp người dùng hiểu rõ hơn về mục đích của phần tử đầu vào.
* Thẻ <select> được sử dụng để tạo một danh sách thả xuống cho phép người dùng chọn một tùy chọn.
* Thẻ <option> được sử dụng để tạo một tùy chọn trong danh sách thả xuống.
* Thẻ <textarea> được sử dụng để tạo một hộp văn bản có thể cuộn để người dùng nhập văn bản dài.
* Thẻ <button> được sử dụng để tạo một nút cho phép người dùng thực hiện một hành động.

## 2.5. Một số thẻ HTML đặc biệt

* Thẻ DOCTYPE: Xác định phiên bản HTML của trang web.
* Thẻ html: Thẻ bao bọc toàn bộ nội dung của trang web.
* Thẻ head : Chứa thông tin về trang web, như tiêu đề, charset, và liên kết đến các tập tin CSS.
* Thẻ body: chứa nội dung chính của trang web, như văn bản, hình ảnh,...
* Thẻ meta: cung cấp bổ sung về web, như charset, author,...
* Thẻ title:Xác định tiêu đề của trang web.

## 2.6. HTML5

* HTML5 là phiên bản thứ 5 của ngôn ngữ HTML (HyperText Markup Language) được sử dụng để cấu trúc và trình bày nội dung trên World Wide Web (WWW). Đây là công nghệ cốt lõi cho các trang web hiện đại, mang đến nhiều cải tiến so với các phiên bản HTML trước đây.
* Cú pháp HTML5 :

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

<title>Tiêu đề trang web</title>

</head>

<body>

Nội dung trang web

</body>

</html>

<!DOCTYPE html>: Khai báo loại tài liệu là HTML5.

<html>: Thẻ bắt đầu của trang HTML.

* lang="vi": Thuộc tính xác định ngôn ngữ của trang web (tiếng Việt trong ví dụ này).
* <head>: Khu vực chứa thông tin về trang web, bao gồm tiêu đề, charset, v.v.
* <meta charset="UTF-8">: Xác định mã hóa ký tự cho trang web (UTF-8 là phổ biến nhất).
* <title>: Tiêu đề của trang web.
* </head>: Thẻ kết thúc khu vực head.
* <body>: Khu vực chứa nội dung chính của trang web.
* Nội dung trang web: Toàn bộ nội dung hiển thị trên trang web, bao gồm văn bản, hình ảnh, video, v.v.
* </body>: Thẻ kết thúc khu vực body.
* </html>: Thẻ kết thúc trang HTML.
* Khai báo DOCTYPE trong HTML5 : <!DOCTYPE html>
* Thẻ ngữ nghĩa trong HTML5 là những thẻ được sử dụng để xác định rõ ràng ý nghĩa của nội dung bên trong thẻ. Việc sử dụng thẻ ngữ nghĩa giúp cải thiện khả năng tiếp cận, khả năng đọc hiểu mã và hiệu suất của trang web.
* Một số thẻ ngữ nghĩa phổ biến:
* <header>: Phần đầu trang web, thường chứa logo, tiêu đề và menu.
* <main>: Phần nội dung chính của trang web.
* <section>: Một phần nội dung riêng biệt trong trang web.
* <article>: Một bài viết hoặc mục tin.
* <aside>: Thanh bên trang web, thường chứa nội dung phụ như quảng cáo hoặc liên kết.
* <footer>: Phần cuối trang web, thường chứa thông tin liên hệ và bản quyền.
* <h1>, <h2>, ..., <h6>: Thẻ tiêu đề
* <p>: Thẻ đoạn văn
* <img>: Thẻ hình ảnh
* <a>: Thẻ liên kết
* <strong>, <em>: Thẻ định dạng văn bản
* <ul>, <ol>: Thẻ danh sách
* <table>: Thẻ bảng
* <video>: Thẻ video
* <audio>: Thẻ âm thanh
* <form>: Thẻ biểu mẫu
* <input>: Thẻ nhập liệu
* <button>: Thẻ nút
* Web form là một phần quan trọng của nhiều trang web, cho phép người dùng tương tác với trang web bằng cách nhập dữ liệu. HTML5 cung cấp một số tính năng mới để tạo web form mạnh mẽ và dễ sử dụng hơn.

## Kết luận chương 2

HTML là ngôn ngữ lập trình cơ bản để tạo trang web. Nó được sử dụng để định cấu trúc nội dung trang web, bao gồm văn bản, hình ảnh, video và các yếu tố khác. HTML sử dụng các thẻ để xác định các phần khác nhau của trang web và cách chúng được hiển thị. Là một phần quan trọng trong việc học lập trình web, nó cung cấp kiến thức nền tảng về cách thức hoạt động của HTML và cách sử dụng nó để tạo trang web.

**CHƯƠNG III: CSS và CSS3**

 (Cascading Style Sheets)

## 3.1. CSS là gì?

* là viết tắt của Cascading Style Sheets, là một ngôn ngữ lập trình được sử dụng để định dạng và tạo kiểu cho các trang web. CSS giúp kiểm soát cách hiển thị các yếu tố HTML trên màn hình, bao gồm phông chữ, màu sắc, bố cục, v.v...

## 3.2. Cú pháp CSS

* Chọn bộ chọn (Selector):
* Xác định phần tử HTML mà bạn muốn định dạng.

Ví dụ: h1, p, .class, #id.

* Khai báo thuộc tính (Property):
* Xác định thuộc tính mà bạn muốn thay đổi.

Ví dụ: color, font-size, background-color.

* Gán giá trị (Value):
* Xác định giá trị cho thuộc tính.

Ví dụ: red, 24px, #ffffff

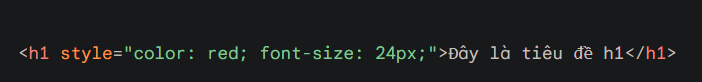
selector {

property: value;

}

## 3.3. Áp dụng CSS vào trang HTML

* Inline CSS:
* Thêm style trực tiếp vào thẻ HTML bằng thuộc tính style.
* Ưu điểm: Đơn giản, nhanh chóng.
* Nhược điểm: Khó quản lý, không phù hợp cho việc áp dụng style cho nhiều phần tử.
* Ví dụ:



Hình 1

* Internal CSS:
* Thêm style vào file HTML trong thẻ <style>.
* Ưu điểm: Dễ quản lý, phù hợp cho việc áp dụng style cho nhiều phần tử trong cùng một file HTML.
* Nhược điểm: Không phù hợp cho việc áp dụng style cho nhiều file HTML.
* Ví dụ :



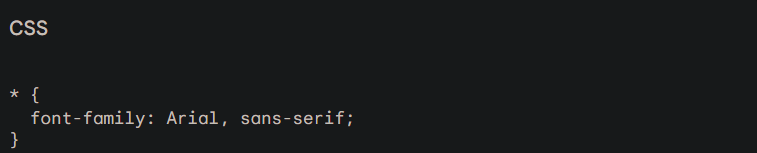
Hình 2

## 3.4. Selectors

* Selectors (bộ chọn) là một phần quan trọng của CSS, đóng vai trò xác định các phần tử HTML để áp dụng các kiểu dáng cho chúng. Có nhiều loại selectors khác nhau, mỗi loại có cách sử dụng và ưu nhược điểm riêng.
* Selectors đơn giản:
* Thẻ tag: Chọn dựa trên tên thẻ HTML, ví dụ: h1, p, div.
* Class: Chọn dựa trên class của phần tử, ví dụ: .class-name.
* ID: Chọn dựa trên ID duy nhất của phần tử, ví dụ: #element-id.
* Selectors kết hợp:
* Hậu duệ: Chọn các phần tử nằm bên trong một phần tử khác, ví dụ: div p (chọn tất cả đoạn văn <p> nằm bên trong thẻ <div>).
* Con: Chọn các phần tử trực tiếp là con của một phần tử khác, ví dụ: header > nav (chọn thẻ <nav> là con trực tiếp của thẻ <header>).
* Anh chị em: Chọn các phần tử cùng cấp với một phần tử khác, ví dụ: ul + li (chọn tất cả thẻ <li> nằm sau thẻ <ul> cùng cấp).
* Pseudo-classes:
* Thêm các kiểu dáng dựa trên trạng thái của phần tử, ví dụ: :hover (kiểu khi di chuột qua), :active (kiểu khi click giữ).
* Pseudo-elements:
* Thêm các kiểu dáng cho các phần cụ thể của một phần tử, ví dụ: ::before (thêm nội dung trước), ::after (thêm nội dung sau).
* Selectors thuộc tính:
* họn dựa trên giá trị của thuộc tính HTML, ví dụ: [href$=".pdf"] (chọn các liên kết có đuôi là ".pdf").

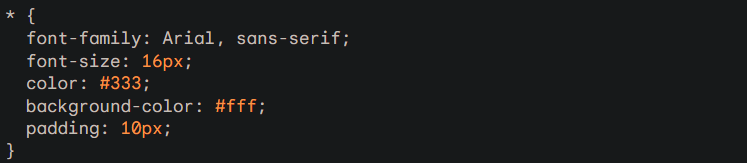
### *3.4.1. Universal selector*

* Universal selector (ký hiệu: \*) là một trong những bộ chọn đơn giản nhất trong CSS. Nó chỉ định tất cả các phần tử HTML trên trang web và áp dụng các kiểu dáng được khai báo cho chúng.
* Sử dụng Universal Selector:
* Mặc dù có thể sử dụng \* để áp dụng style cho mọi phần tử, thực tế không khuyến khích sử dụng thường xuyên.
* Lý do là nó có thể gây ra khó khăn trong việc duy trì và quản lý code CSS. Áp dụng style cho tất cả phần tử có thể dẫn đến các xung đột style không mong muốn, làm cho việc sửa lỗi và cập nhật trở nên khó khăn.
* Sử dụng \* hợp lý:
* Kiểm tra nhanh các kiểu dáng trên toàn bộ trang web.
* Áp dụng các thiết lập mặc định cơ bản cho tất cả phần tử (như font mặc định, màu nền mặc định).
* Kết hợp với các pseudo-classes và pseudo-elements để tăng tính cụ thể.



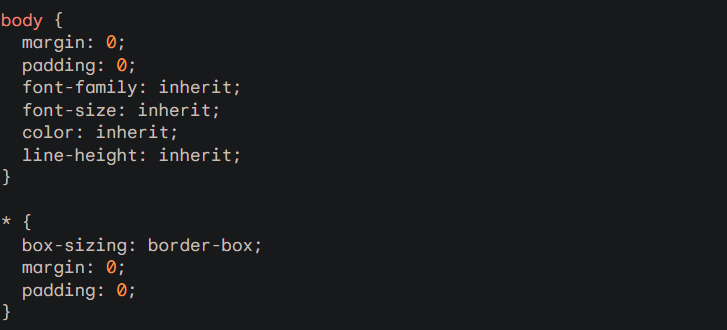
Hình 3

* Áp dụng các thiết lập mặc định cơ bản cho tất cả phần tử (như font mặc định, màu nền mặc định).
* Sử dụng Universal Selector: Cách đơn giản nhất để áp dụng thiết lập mặc định cho tất cả phần tử là sử dụng Universal selector (\*).



Hình 4

* Sử dụng thuộc tính reset: Bạn có thể sử dụng thuộc tính reset trong CSS để khôi phục các thiết lập mặc định của trình duyệt cho các phần tử.



Hình 5

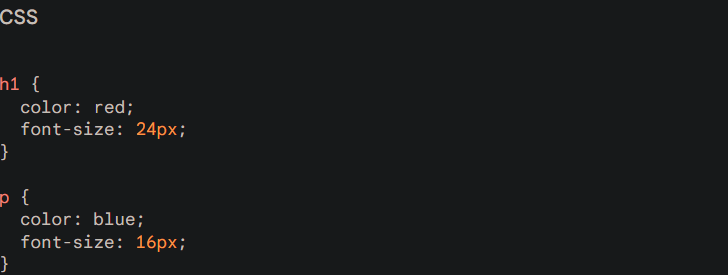
* Sử dụng CSS preprocessors: Có thể sử dụng CSS preprocessors như Sass hoặc Less để viết code CSS dễ dàng và hiệu quả hơn.
* Ví dụ, có thể sử dụng mixins để tạo các thiết lập mặc định và áp dụng cho các phần tử khác nhau:



Hình 6

### *3.4.2. Type selector*

* Type selector là một bộ chọn đơn giản trong CSS dùng để chọn các phần tử HTML dựa trên tên thẻ của chúng.



Hình 7

* Trong ví dụ trên:
* h1 chọn tất cả các thẻ tiêu đề h1 và áp dụng màu đỏ và kích thước font 24px.
* p chọn tất cả các đoạn văn và áp dụng màu xanh và kích thước font 16px.

### *3.4.3. Identity selector*

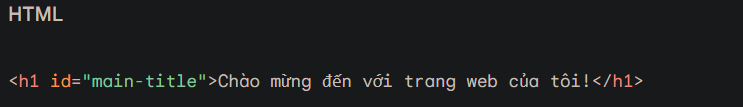
* Identity selector, còn được gọi là ID selector, là một bộ chọn cho phép xác định và áp dụng style cho một phần tử HTML duy nhất trên trang web.
* Cú pháp :

CSS



Hình 8

* Ví dụ :





Hình 9

### *3.4.4. Class selector*

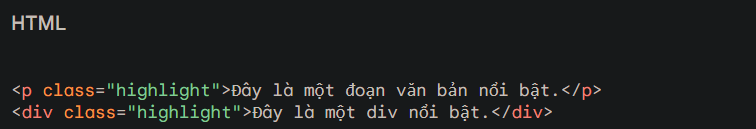
* Class selector là một bộ chọn linh hoạt trong CSS, cho phép xác định và áp dụng style cho một nhóm phần tử HTML có cùng class.
* Cú pháp:

CSS



Hình 10

* Ví dụ:



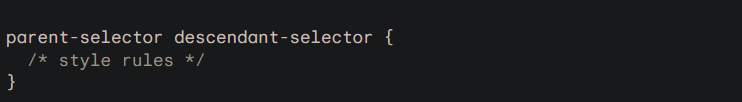


Hình 11

### *3.4.5. Descendant selector*

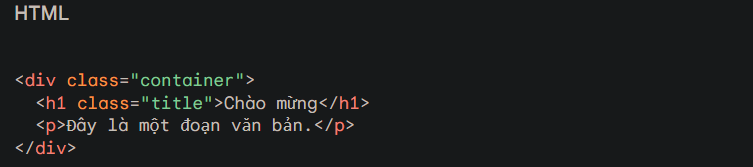
* Descendant selector, còn được gọi là space combinator, là một bộ chọn cho phép xác định và áp dụng style cho các phần tử HTML là hậu duệ trực tiếp hoặc gián tiếp của một phần tử khác.
* Cú pháp:

CSS



Hình 12

* Ví dụ:



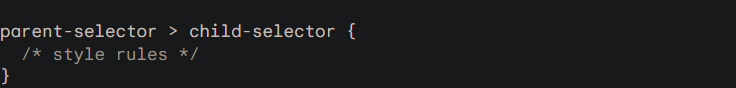


Hình 13

### *3.4.6. Child selector*

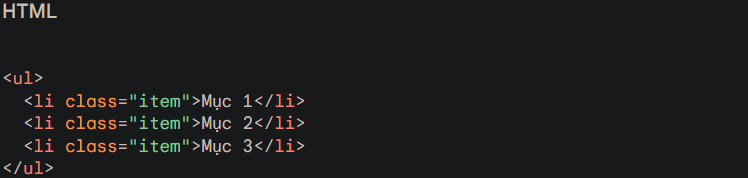
* Child selector, còn được gọi là direct descendant selector, là một bộ chọn dùng để xác định và áp dụng style cho các phần tử HTML là con trực tiếp của một phần tử khác.
* Cú pháp:

CSS



Hình 14

* Ví dụ:



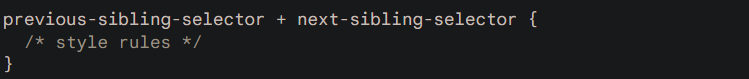


Hình 15

### *3.4.7. Adjacent sibling selector*

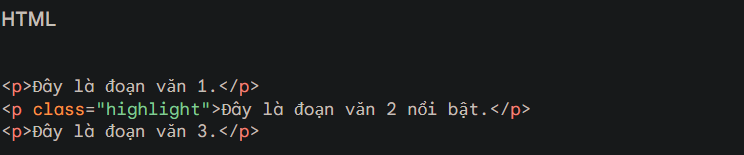
* Adjacent sibling selector, còn được gọi là plus (+) selector, là một bộ chọn cho phép xác định và áp dụng style cho một phần tử HTML là anh chị em liền kề của một phần tử khác.
* Cú pháp:

CSS



Hình 16

* Ví dụ:





Hình 17

### *3.4.8. Attribute selector*

* Attribute selector là một bộ chọn mạnh mẽ trong CSS cho phép xác định và áp dụng style cho các phần tử HTML dựa trên các thuộc tính của chúng. Thuộc tính là những thông tin bổ sung được gắn vào thẻ HTML để cung cấp thêm dữ liệu về phần tử đó.
* Cú pháp: [attribute][operator][value]
* attribute: tên của thuộc tính HTML.
* operator: tùy chọn, dùng để so sánh giá trị của thuộc tính. Các toán tử thường dùng:
* =: giá trị chính xác.
* ^=: bắt đầu bằng giá trị.
* $=: kết thúc bằng giá trị.
* \*=: chứa giá trị.
* |=: giá trị là phần tách biệt bằng dấu gạch ngang.
* value: giá trị của thuộc tính được so sánh.
* Ví dụ:





Hình 18

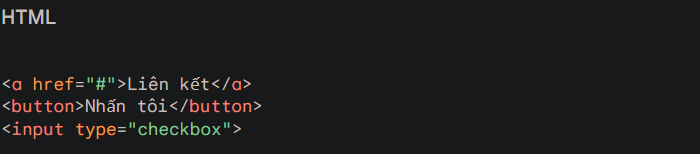
### *3.4.9. Pseudo class selector*

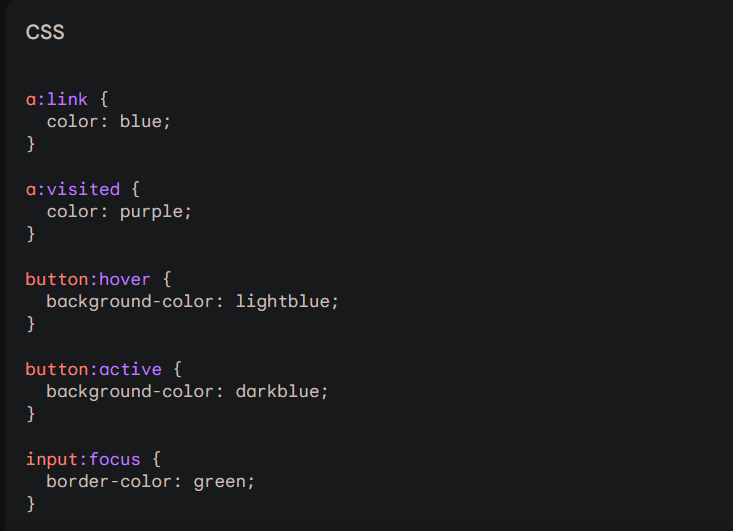
* Pseudo-class selector là một bộ chọn đặc biệt trong CSS cho phép xác định và áp dụng style cho các phần tử HTML dựa trên trạng thái của chúng. Trạng thái này có thể là trạng thái tương tác người dùng (hover, click, focus), trạng thái của phần tử (checked, disabled, first-child), hoặc trạng thái tùy chỉnh được định nghĩa bằng JavaScript.
* Cú pháp:



Hình 19

* Ví dụ:





Hình 20

### *3.4.10. Group selector*

* **Group Selector** không phải là một bộ chọn riêng biệt, mà là một cách kết hợp nhiều bộ chọn khác nhau bằng cách dùng dấu phẩy (,) để tách chúng ra. Điều này cho phép áp dụng cùng một style cho nhiều phần tử HTML khác nhau.
* Ví dụ:



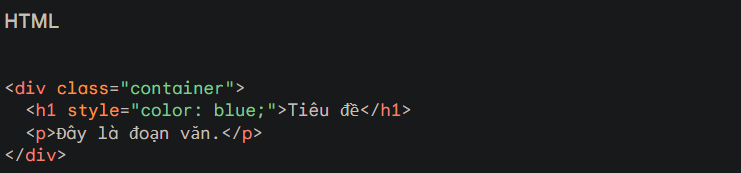
Hình 21

## 3.5. Đơn vị đo lường CSS

* Đơn vị tuyệt đối:
* px (pixel): Đơn vị pixel là đơn vị cơ bản trong CSS, tương ứng với một điểm ảnh trên màn hình.
* pt (point): 1pt bằng 1/72 inch.
* in (inch): Đơn vị inch tương ứng với 1 inch thực tế.
* cm (centimeter): Đơn vị cm tương ứng với 1 cm thực tế.
* mm (millimeter): Đơn vị mm tương ứng với 1 mm thực tế.
* Đơn vị tương đối:
* % (percent): Đơn vị phần trăm là đơn vị tương đối so với kích thước của phần tử cha.
* em: 1em bằng kích thước font của phần tử hiện tại.
* rem: 1rem bằng kích thước font của phần tử gốc (html).

## 3.6. Kế thừa thuộc tính

* Kế thừa thuộc tính là một tính năng quan trọng trong CSS cho phép các phần tử HTML con kế thừa các thuộc tính style từ phần tử HTML cha. Điều này giúp giảm thiểu việc lặp lại code và tạo bố cục trang web thống nhất.
* Cách thức hoạt động:
* Các thuộc tính style được kế thừa từ phần tử cha sang con theo cấu trúc DOM.
* Các phần tử con có thể ghi đè các thuộc tính được kế thừa bằng cách khai báo các thuộc tính riêng.
* Có một số thuộc tính không được kế thừa, ví dụ như position, display, float, clear, visibility, cursor, outline, z-index, content, page-break-before, page-break-after, page-break-inside, orphans, widows, pointer-events, mix-blend-mode, isolation, filter, backdrop-filter, transform, transform-origin, transform-style, transition, animation, will-change.
* Ví dụ:





Hình 22

## 3.7. Các nhóm thuộc tính trong CSS

* Thuộc tính định dạng font chữ, văn bản (font, text): CSS cung cấp nhiều thuộc tính để định dạng font chữ và văn bản, giúp kiểm soát giao diện của trang web một cách chi tiết.
* Dưới đây là một số thuộc tính phổ biến nhất:
* Font family:
* Xác định kiểu font chữ sử dụng cho phần tử.
* Có thể sử dụng tên font có sẵn trên hệ thống hoặc font web được tải lên.
* Ví dụ: font-family: Arial, sans-serif;
* Font size:
* Xác định kích thước font chữ.
* Có thể sử dụng đơn vị tuyệt đối (px, pt, in, cm, mm) hoặc đơn vị tương đối (em, rem).
* Ví dụ: font-size: 16px;
* Font weight:
* Xác định độ dày của font chữ.
* Có các giá trị thường dùng như normal, bold, bolder, lighter.
* Ví dụ: font-weight: bold;
* Font style:
* Xác định kiểu chữ in nghiêng hoặc bình thường.
* Có các giá trị normal, italic, oblique.
* Ví dụ: font-style: italic;
* Text color:
* Xác định màu sắc của văn bản.
* Có thể sử dụng tên màu, mã màu RGB hoặc HEX.
* Ví dụ: color: red;
* Text decoration:
* Xác định các kiểu trang trí cho văn bản như underline, overline, line-through.
* Có thể sử dụng một hoặc kết hợp nhiều giá trị.
* Ví dụ: text-decoration: underline;
* Text align:
* Xác định vị trí canh lề của văn bản.
* Có các giá trị left, center, right, justify.
* Ví dụ: text-align: center;
* Text transform:
* Xác định kiểu biến đổi chữ hoa, chữ thường hoặc viết hoa toàn bộ.
* Có các giá trị uppercase, lowercase, capitalize.
* Ví dụ: text-transform: uppercase;
* Letter spacing:
* Xác định khoảng cách giữa các ký tự trong văn bản.
* Có thể sử dụng đơn vị tuyệt đối hoặc đơn vị tương đối.
* Ví dụ: letter-spacing: 2px;
* Line height:
* Xác định khoảng cách giữa các dòng văn bản.
* Có thể sử dụng đơn vị tuyệt đối hoặc đơn vị tương đối.
* Ví dụ: line-height: 1.5;
* Word spacing:
* Xác định khoảng cách giữa các từ trong văn bản.
* Có thể sử dụng đơn vị tuyệt đối hoặc đơn vị tương đối.
* Ví dụ: word-spacing: 2px;
* Text shadow:
* Tạo hiệu ứng bóng đổ cho văn bản.
* Có thể sử dụng các thuộc tính để xác định vị trí, độ mờ, màu sắc của bóng đổ.
* Ví dụ: text-shadow: 2px 2px 4px rgba(0,0,0,0.5);
* text-indent: thụt lề đầu dòng.
* white-space: kiểm soát cách hiển thị các khoảng trắng.
* word-break: xác định cách ngắt từ khi cần thiết.
* overflow-wrap: xác định cách xử lý văn bản khi vượt quá chiều rộng của phần tử.
* Thuộc tính định dạng nền (background):
* Màu sắc: Sử dụng background-color để đặt màu nền đơn giản.
* Hình ảnh: Sử dụng background-image để đặt một hình ảnh làm nền.
* Vị trí: Sử dụng background-position để xác định vị trí của hình ảnh nền.
* Lặp lại: Sử dụng background-repeat để xác định cách hình ảnh nền được lặp lại.
* Kích thước: Sử dụng background-size để xác định kích thước của hình ảnh nền.
* Đính kèm: Sử dụng background-attachment để xác định cách hình ảnh nền di chuyển khi cuộn trang.
* Mô hình hộp (box model) là Mô hình hộp là một khái niệm quan trọng trong CSS giúp cho hiểu cách các phần tử HTML được hiển thị trên trang web. Mô hình này mô tả mỗi phần tử như một hộp hình chữ nhật.
* Các thuộc tính định margin, padding, border:
* Nội dung (Content): Là phần nội dung thực tế của phần tử, bao gồm văn bản, hình ảnh, video, v.v.
* Đệm (Padding): Là khoảng trống giữa nội dung và viền của phần tử.
* Viền (Border): Là đường viền bao quanh phần tử.
* Lề (Margin): Là khoảng trống giữa phần tử và các phần tử khác
* Thuộc tính định dạng cách hiển thị (display):
* block: Hiển thị phần tử như một khối độc lập, chiếm toàn bộ chiều rộng của phần tử cha.
* inline: Hiển thị phần tử như một phần nội tuyến, nằm trên cùng dòng với các phần tử khác.
* inline-block: Hiển thị phần tử như một khối nội tuyến, nhưng cho phép thiết lập chiều rộng và chiều cao.
* none: Ẩn phần tử khỏi trang web.
* flex: Hiển thị phần tử theo bố cục flexbox, cho phép bố trí các phần tử con một cách linh hoạt.
* grid: Hiển thị phần tử theo bố cục grid, cho phép tạo bố cục dạng lưới với các cột và hàng.
* Thuộc tính xác định vị trí (position):
* static: Vị trí mặc định của phần tử, được xác định bởi bố cục trang web.
* relative: Vị trí của phần tử được xác định so với vị trí ban đầu của nó.
* absolute: Vị trí của phần tử được xác định so với phần tử cha gần nhất có vị trí relative hoặc absolute.
* fixed: Vị trí của phần tử được xác định so với khung nhìn của trình duyệt.

## 3.8. Float & Clear

* Float giúp di chuyển các phần tử (như hình ảnh, đoạn văn) sang trái hoặc phải, cho phép chúng trôi nổi bên cạnh nội dung khác. Tuy nhiên, điều này có thể gây ra một số vấn đề về bố cục, đặc biệt là khi các phần tử khác bị "mắc kẹt" bên dưới phần tử nổi.
* Thuộc tính float có các giá trị chính:
* left: Cho phần tử nổi sang trái.
* right: Cho phần tử nổi sang phải.
* none: (Mặc định) Giữ phần tử ở vị trí ban đầu.
* Clear ra đời để giải cứu! Nó cho phép xóa các phần tử nổi khỏi dòng văn bản bình thường, giải phóng không gian cho các phần tử khác hiển thị đúng vị trí.
* Thuộc tính clear có các giá trị chính:
* left: Xóa các phần tử nổi bên trái phần tử hiện tại.
* right: Xóa các phần tử nổi bên phải phần tử hiện tại.
* both: Xóa cả hai bên.
* none: (Mặc định) Không xóa bất kỳ phần tử nổi nào.

## 3.9. Flex

* Flex là một mô hình bố cục CSS mới mẻ và mạnh mẽ giúp tạo ra các giao diện web linh hoạt và đáp ứng. Nó giải quyết nhiều vấn đề bố cục truyền thống và giúp thiết kế trang web dễ dàng hơn bao giờ hết.
* Các thuộc tính của Flex:
* display: flex: Chuyển đổi một phần tử thành container Flexbox.
* flex-direction: Xác định hướng bố trí các phần tử con (row, column, row-reverse, column-reverse).
* justify-content: Căn chỉnh các phần tử con theo chiều ngang (flex-start, flex-end, center, space-between, space-around).
* align-items: Căn chỉnh các phần tử con theo chiều dọc (flex-start, flex-end, center, baseline, stretch).
* flex: Thuộc tính shorthand để thiết lập flex-grow, flex-shrink, và flex-basis cùng lúc.

## 3.10. Grid

* Grid là một hệ thống bố cục 2 chiều mới trong CSS, mang đến nhiều ưu điểm vượt trội so với các phương pháp bố cục truyền thống. Grid giúp tạo ra các giao diện web phức tạp một cách dễ dàng và hiệu quả, đồng thời đảm bảo khả năng đáp ứng trên mọi thiết bị.
* Các thuộc tính của Grid:
* display: grid: Chuyển đổi một phần tử thành container Grid.
* grid-template-columns: Xác định số lượng và kích thước của các cột.
* grid-template-rows: Xác định số lượng và kích thước của các hàng.
* grid-gap: Thiết lập khoảng cách giữa các hàng và cột.
* justify-content: Căn chỉnh các phần tử con theo chiều ngang.
* align-items: Căn chỉnh các phần tử con theo chiều dọc.
* place-items: Căn chỉnh các phần tử con theo cả chiều ngang và chiều dọc

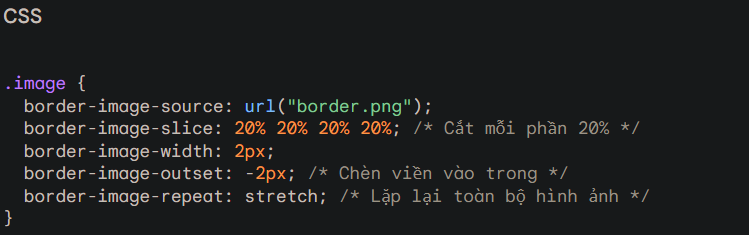
## *3.11. CSS3*

* CSS3 là phiên bản nâng cấp mới nhất của Cascading Style Sheets (CSS), ngôn ngữ được sử dụng để định dạng và trình bày trang web. CSS3 mang đến nhiều tính năng mới mạnh mẽ giúp tạo ra các giao diện web đẹp mắt, hiện đại và tương tác tốt hơn.
* CSS3 Selectors mở rộng khả năng chọn lọc và định dạng các thành phần HTML trong trang web so với các phiên bản CSS trước đó.
* Pseudo-classes: Chọn phần tử dựa trên trạng thái của nó (ví dụ: :hover, :active, :checked).
* Pseudo-elements: Chọn các phần cụ thể của một phần tử (ví dụ: ::before, ::after).
* Attribute selectors: Chọn phần tử dựa trên giá trị của thuộc tính (ví dụ: [href], [data-type]).
* Combinators: Kết hợp các selector để tạo ra các lựa chọn phức tạp hơn (ví dụ: div p, .class:hover).
* Ví dụ:



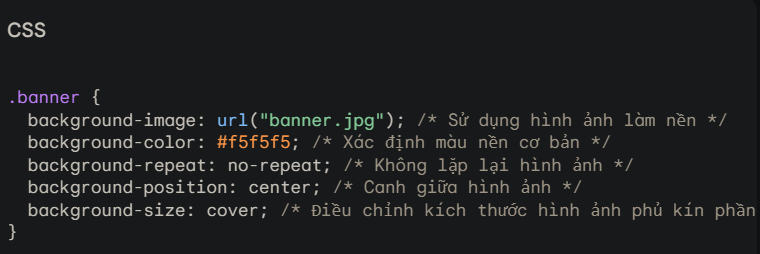
Hình 23

* CSS3 Rounded Corners
* CSS3 Border Images là một tính năng mạnh mẽ cho phép sử dụng hình ảnh làm viền cho các phần tử HTML. Nó giúp tạo ra các hiệu ứng viền độc đáo và linh hoạt, vượt xa những đường viền đơn điệu truyền thống.
* Các thuộc tính chính:
* border-image-source: Chỉ định URL của hình ảnh viền.
* border-image-slice: Xác định cách chia hình ảnh viền thành chín phần.
* border-image-width: Thiết lập độ rộng của viền.
* border-image-outset: Xác định khoảng cách giữa viền và nội dung phần tử.
* border-image-repeat: Kiểm soát cách lặp lại các phần của hình ảnh viền.
* Ví dụ:



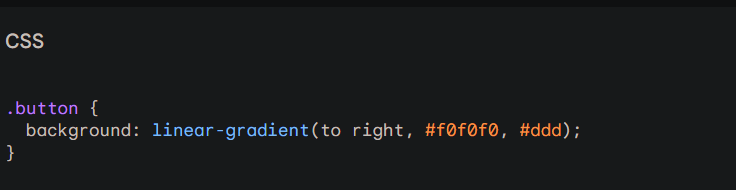
Hình 24

* CSS3 Backgrounds là tạo ra những nền giao diện đẹp mắt và linh hoạt. Nó cung cấp cho nhiều công cụ vượt xa màu nền đơn điệu, cho phép thêm hình ảnh, gradient, video, thậm chí cả các họa tiết phức tạp.
* Các thuộc tính chính:
* background-image: Cho phép sử dụng hình ảnh làm nền, tạo ra hiệu ứng thị giác phong phú.
* background-color: Xác định màu nền cơ bản, thường phối hợp với hình ảnh hoặc gradient.
* background-repeat: Kiểm soát cách lặp lại hình ảnh hoặc gradient trên nền, tránh tình trạng méo mó hoặc trống trải.
* background-position: Xác định vị trí của hình ảnh hoặc gradient trên nền, cho phép căn chỉnh chính xác.
* background-size: Điều chỉnh kích thước của hình ảnh hoặc gradient trên nền, đảm bảo phù hợp với kích thước phần tử.
* background-clip: Xác định phần tử nào nhận được hiệu ứng nền, giúp tạo ra các hiệu ứng chồng lớp thú vị.
* Gradients: Sử dụng nhiều màu sắc để tạo ra hiệu ứng chuyển màu mượt mà trên nền, tăng thêm tính thẩm mỹ.
* Videos: Thêm video làm nền, tạo ra trải nghiệm tương tác và năng động hơn.
* Ví dụ:



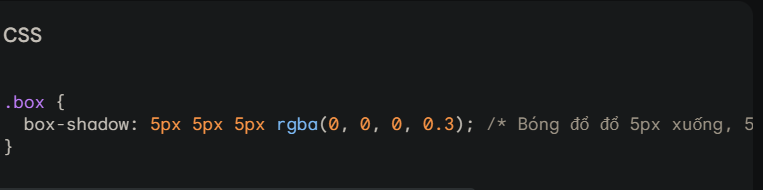
Hình 25

* CSS Gradients là tạo ra những nền giao diện đẹp mắt và linh hoạt. Nó cho phép tạo ra các hiệu ứng chuyển màu mượt mà, tăng thêm tính thẩm mỹ và chiều sâu cho các thành phần .
* Những loại chính:
* Linear Gradients: Chuyển màu theo một đường thẳng, từ điểm bắt đầu đến điểm kết thúc.
* Radial Gradients: Chuyển màu từ một điểm trung tâm ra ngoài theo hình tròn hoặc elip.
* Những thành phần chính:
* Color Stops: Xác định các màu sắc và vị trí của chúng trong gradient.
* Hướng: Xác định hướng chuyển màu của gradient (đối với linear gradients).
* Ví dụ:



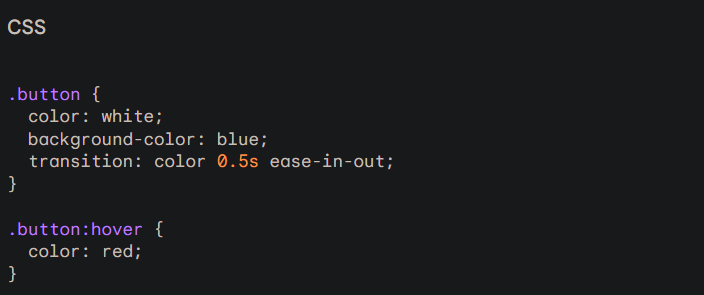
Hình 26

* CSS3 Shadows là tạo chiều sâu, tách biệt các thành phần và tăng tính thẩm mỹ cho giao diện web.
* Những loại chính:
* Drop Shadow: Bóng đổ rơi xuống dưới và sang phải của phần tử, tạo hiệu ứng 3D nhẹ nhàng.
* Inner Shadow: Bóng đổ xuất hiện bên trong phần tử, tạo cảm giác sâu hơn.
* Blur Shadow: Bóng đổ được làm mờ, tạo hiệu ứng mềm mại hơn.
* Spread Shadow: Bóng đổ lan tỏa rộng hơn ra ngoài, tạo hiệu ứng mạnh mẽ hơn.
* Thuộc tính chính:
* box-shadow: Xác định các thuộc tính của bóng đổ, bao gồm offset (dấu dịch), blur (độ mờ), spread (độ lan tỏa) và màu sắc.
* Ví dụ:



Hình 27

* CSS3 Fonts cho phép các nhà thiết kế thoát khỏi giới hạn của các font chữ "web-safe" và áp dụng nhiều lựa chọn typographical hơn.
* Tính năng chính:
* @font-face: Quy tắc này cho phép nhúng trực tiếp font chữ tùy chỉnh vào trang web của mình. Có thể xác định tên font chữ, chỉ định đường dẫn tệp (cục bộ hoặc được lưu trữ) và thậm chí bao gồm nhiều định dạng để tối ưu hóa khả năng tương thích với trình duyệt.
* Đa dạng định dạng font chữ: Nhờ CSS3, có thể cung cấp font chữ của mình ở nhiều định dạng khác nhau như WOFF, WOFF2, TTF và OTF. Trình duyệt sẽ ưu tiên các định dạng nhỏ hơn để tải nhanh hơn, đảm bảo trải nghiệm người dùng mượt mà.
* Họ font và biến thể: CSS3 cho phép chỉ định nhiều font chữ trong một họ, cho phép trình duyệt chọn tùy chọn tốt nhất dựa trên khả năng của nó. Cũng có thể xác định các kiểu font chữ như in nghiêng, đậm và chữ hoa nhỏ để kiểm soát linh hoạt.
* Dịch vụ font web: Các dịch vụ như Google Fonts và Adobe Typekit lưu trữ thư viện font chữ khổng lồ và cung cấp cách triển khai dễ dàng thông qua URL, đơn giản hóa việc tích hợp font chữ và giảm tải dung lượng tệp trên máy chủ.
* CSS3 2D Transforms là một tính năng mạnh mẽ cho phép biến đổi, di chuyển, xoay và định dạng các phần tử HTML theo chiều ngang và chiều dọc. Nó cung cấp nhiều khả năng sáng tạo và hiệu ứng động cho trang web.
* Các tính năng chính:
* translate(x, y): Di chuyển một phần tử theo trục X và Y.
* rotate(angle): Xoay một phần tử theo một góc nhất định.
* scale(x, y): Thay đổi kích thước của một phần tử theo chiều ngang và chiều dọc.
* skewX(angle): Nghiêng một phần tử theo trục X.
* skewY(angle): Nghiêng một phần tử theo trục Y.
* matrix(a, b, c, d, e, f): Áp dụng một ma trận chuyển đổi phức tạp hơn cho một phần tử.
* CSS3 3D Transforms mang đến khả năng biến đổi và định dạng các phần tử HTML theo cả chiều sâu, mở ra cánh cửa cho các hiệu ứng 3D ấn tượng và phức tạp hơn.
* Các thuộc tính chính:
* rotateX(angle): Xoay một phần tử quanh trục X (trục ngang).
* rotateY(angle): Xoay một phần tử quanh trục Y (trục dọc).
* rotateZ(angle): Xoay một phần tử quanh trục Z (trục đi vào màn hình).
* perspective(distance): Xác định điểm nhìn của người dùng, tạo cảm giác xa gần cho các phần tử 3D.
* translate3d(x, y, z): Di chuyển một phần tử trong không gian 3D theo các trục X, Y và Z.
* scaleX(x), scaleY(y), scaleZ(z): Thay đổi kích thước của một phần tử theo các hướng X, Y và Z.
* CSS3 Transitions cho phép tạo hiệu ứng chuyển đổi mượt mà giữa các trạng thái khác nhau của một phần tử HTML. Khi thuộc tính của một phần tử thay đổi (ví dụ: màu sắc, vị trí, kích thước), có thể sử dụng Transitions để tạo hiệu ứng chuyển đổi thay vì thay đổi đột ngột.
* Cách sử dụng :
* Xác định thuộc tính cần chuyển đổi: Sử dụng thuộc tính transition trong CSS để xác định các thuộc tính muốn áp dụng hiệu ứng chuyển đổi.
* Thiết lập thời gian chuyển đổi: Sử dụng thuộc tính transition-duration để xác định thời gian (tính bằng giây hoặc mili giây) của hiệu ứng chuyển đổi.
* Chọn kiểu chuyển đổi: Sử dụng thuộc tính transition-timing-function để chọn kiểu chuyển đổi mong muốn (ví dụ: tuyến tính, ease-in, ease-out, v.v.).
* Thêm độ trễ (tùy chọn): Sử dụng thuộc tính transition-delay để thêm độ trễ (tính bằng giây hoặc mili giây) trước khi bắt đầu hiệu ứng chuyển đổi.
* Ví dụ:



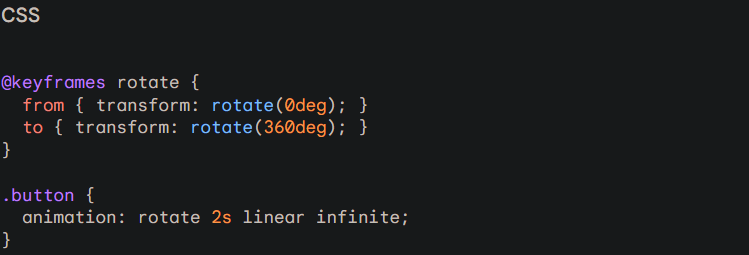
Hình 28

* CSS3 Multiple Columns là khả năng tạo bố cục đa cột vẫn được hỗ trợ thông qua thuộc tính column-count. Đây là một phương pháp linh hoạt và tiện lợi để sắp xếp nội dung thành nhiều cột trên trang web.
* Cách sử dụng:
* Xác định phần tử đa cột: Chọn phần tử HTML muốn chia thành nhiều cột. Thường thì đó sẽ là một container hoặc một đoạn văn bản dài.
* Áp dụng thuộc tính column-count: Sử dụng thuộc tính này trong CSS để xác định số cột mong muốn. Ví dụ: column-count: 3; sẽ chia phần tử thành 3 cột.
* Thiết lập các thuộc tính bổ sung (tùy chọn): có thể tùy chỉnh thêm các thuộc tính như:
* column-width: Xác định độ rộng của mỗi cột.
* column-gap: Xác định khoảng cách giữa các cột.
* column-rule: Tạo đường viền giữa các cột.
* Ví dụ:



Hình 29

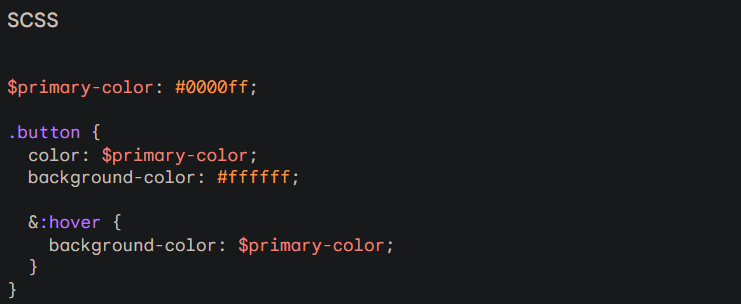
* CSS3 Animations tạo hiệu ứng chuyển động mượt mà và sống động cho các phần tử HTML. Khả năng này giúp nâng cao trải nghiệm người dùng, thu hút sự chú ý và tăng tính tương tác cho trang web.
* Cách sử dụng:
* Xác định keyframes: Sử dụng quy tắc @keyframes để định nghĩa các trạng thái (keyframes) của animation. Mỗi keyframe xác định thuộc tính CSS và giá trị tương ứng tại thời điểm cụ thể trong animation.
* Áp dụng animation cho phần tử: Sử dụng thuộc tính animation trong CSS để liên kết keyframes với phần tử HTML mong muốn. Có thể xác định các thuộc tính như:
* animation-name: Tên của keyframe animation.
* animation-duration: Thời lượng của animation.
* animation-timing-function: Kiểu chuyển động của animation (linear, ease-in, ease-out, v.v.).
* animation-iteration-count: Số lần lặp lại animation.
* Ví dụ:



Hình 30

## 3.12. SCSS

* SCSS là một ngôn ngữ tiền xử lý mạnh mẽ được xây dựng trên nền tảng CSS. Nó cung cấp nhiều tính năng nâng cao giúp viết CSS hiệu quả và dễ dàng hơn.
* Lợi ích của SCSS:
* Biến: Giúp bạn lưu trữ và tái sử dụng giá trị, tạo code dễ hiểu và bảo trì hơn.
* Nesting: Cho phép lồng các quy tắc CSS vào nhau, giúp code gọn gàng và trực quan hơn.
* Mixin: Giúp tạo các khối code có thể sử dụng lại nhiều lần, tiết kiệm thời gian và công sức.
* Functions: Giúp thực hiện các thao tác logic và toán học trong CSS, tăng tính linh hoạt cho code.
* Operators: Cung cấp các toán tử để thao tác với giá trị, giúp code ngắn gọn và hiệu quả hơn.
* Ví dụ:



Hình 31

## 3.13. SASS

* **Là** 1 ngôn ngữ mở rộng có mục đích giúp lập trình viên viết mã css hiệu quả hơn và dễ dàng hơn. SASS cung cấp các tính năng mạnh mẽ như biến, lồng ghép, mixin, và nhiều hơn nữa, giúp tạo ra mã CSS rõ ràng, để bảo trì và tái sử dụng.
* Một số tính năng chính của SASS:
* Biến: cho phép định nghĩa các giá trị có thể sử dụng trong cả mã SASS và CSS kết quả.
* Lồng: ghép cho phép nhóm các quy tắc CSS bênn trong nhau, giúp cải thiện tổ chức và đọc mã.
* Mixin: cho phép tái sử dụng các khối mã CSS.
* Biểu thức toán học và hàm: cho phép sử dụng biểu thức toán học và hàm trong mã SASS.
* Import: cho phép chí nhỏ mã SCC thành các phần nhỏ của import vào 1 tệp duy nhất.

## Kết luận chương 3

Chương CSS cung cấp cho bạn kiến thức nền tảng về cách thức hoạt động của CSS và cách sử dụng nó để tạo kiểu cho trang web. Với cú pháp đơn giản và mạnh mẽ, CSS cho phép lập trình viên xác định các phong cách, bố cục và hiệu ứng trực quan cho các phần tử HTML trên trang web.

CSS3 cải tiến hơn CSS , đem lại nhiều tính năng mới . CSS3 cung cấp một loạt các thuộc tính mới có việc tạo ra hiệu đồ họa và tính năng tương tác phức tạp. Điều này đã mở ra cánh cửa cho việc thiết kế web đa dạng, linh hoạt và phong phú hơn, giúp tạo ra trải nghiệm người dùng độc đáo và thu hút.

# CHƯƠNG 4: THIẾT KẾ WEDSIDE XEM PHIM TRỰC TUYẾN MIỄN PHÍ

## 4.1. Ý tưởng của Website

* Giao diện người dùng đơn giản: Thiết kế giao diện người dùng đơn giản và dễ hiểu để người dùng dễ dàng tìm kiếm và xem phim. Sử dụng CSS để tạo giao diện thân thiện với người dùng.
* Thiết kế nút điều khiển: Tạo các nút điều khiển như play, pause, stop và các nút điều hướng khác để người dùng có thể điều khiển video một cách thuận tiện.
* Thiết kế danh sách phim: tạo danh sách các bộ phim với hình ảnh minh họa, tiêu đề và mô tả ngắn gọn. Đảm bảo rằng các mục trong danh sách được sắp xếp một cách hợp lý và dễ nhìn.
* Trình phát video tùy chỉnh: Sử dụng các thẻ video của HTML5 để nhúng video và tạo giao diện trình phát video tùy chỉnh bằng CSS. Điều này bao gồm trình phát video, thanh tiến độ, nút điều khiển và các tính năng khác.
* Tích hợp biểu mẫu tìm kiếm: Tạo một biểu mẫu tìm kiếm đơn giản để người dùng có thể tìm kiếm phim theo tiêu đề, thể loại hoặc từ khóa khác.
* Tùy chỉnh hình nền và màu sắc: Sử dụng CSS để tùy chỉnh hình nền, màu sắc và phông chữ của trang web để tạo ra một trải nghiệm thú vị và phù hợp với chủ đề của trang web.
* Responsive design: Đảm bảo trang web của bạn có responsive design để nó có thể hiển thị đúng trên các thiết bị di động và máy tính bảng.
* Hiển thị thông tin chi tiết: Khi người dùng nhấp vào một phim, hiển thị thông tin chi tiết như đạo diễn, diễn viên, thể loại, năm sản xuất và mô tả phim.

## 

## 4.2. Xây dựng bố cục của trang Web

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | | | | | | | | | | | | | |
| Logo | Lịch chiếu | Phim | Tin tức phim | | | | Ưu đãi | Thành viên | Đăng kí | | | | Đăng nhập | Tìm kiếm |
|  |
| PHIM OMEGA ONLINE | | | | | | | | | | | | | | |
| Header | | | | | | | | | | | | | | |
| Main | | | | | | | | | | | | | | |
| PHIM MỚI | | | | | | | | | | | | | | |
| Phim 1 | | | | Phim 2 | | | | | | Phim 3 | | | | |
| PHIM NHIỀU NGƯỜI XEM | | | | | | | | | | | | | | |
| Phim 1 | | | | | Phim 2 | | | | | | Phim 3 | | | |
| PHIM SẮP CHIẾU RẠP | | | | | | | | | | | | | | |
| Phim 1 | | | | | | Phim 2 | | | | | | Phim 3 | | |
| Liên hệ mua vé | | | | | | | | | | | | | | |
| Họ và tên | | | | | | | | | | | | | | |
| Email | | | | | | | | | | | | | | |
| Nội dung | | | | | | | | | | | | | | |
| Main | | | | | | | | | | | | | | |
| Flooter | | | | | | | | | | | | | | |
| Copy by PHIM OMEGA ONLINE | | | | | | | | | | | | | | |
| Flooter | | | | | | | | | | | | | | |

* Đầu trang <header>:
* Chứa thông tin như logo của trang web, thanh điều hướng (menu), và các liên kết quan trọng như đăng nhập, đăng kí,...
* chính <main>:
* Là phần bao quát toàn bộ nội dung chính của trang web, chứa thông tin chủ đề, hình ảnh, văn bản.
* Các thành phần được chia nhỏ ra thành các đoạn hoặc ô để dễ quản lý và đọc.
* Chân trang <footer>:
* Chứa thông tin bản quyền và nguồn lấy những video khác.

## 4.3. Thiết kế trang Web bằng HTML và CSS

* File trang chủ-HTML:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Trang chủ</title>

<link rel="stylesheet" href="./buoi10.css">

</head>

<body>

<header><nav><div><ul>

<li><a href=""><img class="logo" src="./pngtree-letter-h-logo-template-design-png-image\_3580314.jpg"></a></li>

<li><a href="">LỊCH CHIẾU</a></li>

<li><a href="">PHIM</a></li>

<li><a href="">TIN TỨC PHIM</a></li>

<li><a href="">ƯU ĐÃI</a></li>

<li><a href="">THÀNH VIÊN</a></li>

<li><a href="d:\thietkeweb\dangky1.html">ĐĂNG KÝ</a></li>

<li><a href="d:\thietkeweb\dangnhap.html">ĐĂNG NHẬP</a></li>

</ul></div>

<div class="timkiem">

<input type="text" placeholder="tim kiem...">

</div></nav>

<h1 class="hn1">PHIM OMEGA ONLINE</h1>

</header>

Css:

\*{

margin: 0;

padding: 0;

box-sizing: border-box;

}

/\*tạo css cho header\*/

header{

background-color: black;

color: aliceblue;

padding: 20px;

width: 100%;

justify-content: space-around;

}

nav ul{

list-style: none;

display: flex;

justify-content: space-between;font-weight: bolder;

}

nav li{margin-right:20px ;}

nav li:hover{background-color: rgb(43, 186, 169);padding: 10px;}

.logo{width: 50px;height: 40px;}

nav .a{

color: aqua;

text-decoration: none;

}

header nav{

width: 95%;

height: 100%;

/\* border: 1px solid #fff; \*/

margin: auto;

display: flex;

align-items: center;

justify-content: space-between;

}

header nav .hn1{

color: #fff;

font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;

display: flex;

align-items: flex;

justify-content: flex-start;

}

header nav ul{

width: 100%;

height: 200%;

/\* border: 1px solid #fff; \*/

display: flex;

align-items: flex-end;

justify-content: center;

}

header nav ul li{

list-style-type: none;

padding: 3px 10px

}

header nav ul li a{

color: #fff;

text-decoration: none;

font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;

font-size: 19px;

font-weight: bold;

transition: all .3s ease-in-out;

}

header nav ul li a:hover{

color:gray;

}

header nav .timkiem{

width: 20%;

height: 100%;

display: flex;

align-items: flex-end;

justify-content: flex-end;

}

header nav .timkiem input{

width: 90%;

padding: 5px 0px 3px 15px;

border-radius: 20px;

border: none;

outline: none;

}

**4.5. kết quả đạt được**



<main>

<section class="section">

<h2>PHIM MỚI</h2>

<div class="flim">

<div class="phim">

<img class="anh" src="./phim-dien-anh-avengers-endgame.jpg">

<h3>Phim1</h3>

<h4>Phim Chuẩn Bị Chiếu</h4>

<button onclick="location.href='xemphim.html'">XEM NGAY</button>

</div><br>

<div class="phim">

<img class="anh" src="./578692Fa75ba9ce4c5ab.jpg">

<h3>Phim2</h3>

<h4>Phim Chuẩn Bị Chiếu</h4>

<button onclick="location.href='xemphim.html'">XEM NGAY</button>

</div><br>

<div class="phim">

<img class="anh" src="./avata-1576594156841526843627.webp">

<h3>Phim3</h3>

<h4>Phim Chuẩn Bị Chiếu</h4>

<button onclick="location.href='xemphim.html'">XEM NGAY</button>

</div>

</div>

</section>

<section class="section">

Css:

h1{margin: 20px 15px 10px ;

text-align: center;

text-transform: uppercase;

font-size: 40px;}

main {padding: 50px;}

.section{margin-bottom: 50px;}

.section p{

text-align: center;

}

.section p,h2,h3{

text-align: center;

}

.section p,h4{

text-align: center;

}

.section h2{margin: 0 0 20px;}

.flim{display:flex;

flex-wrap: wrap;

justify-content: space-between;

}

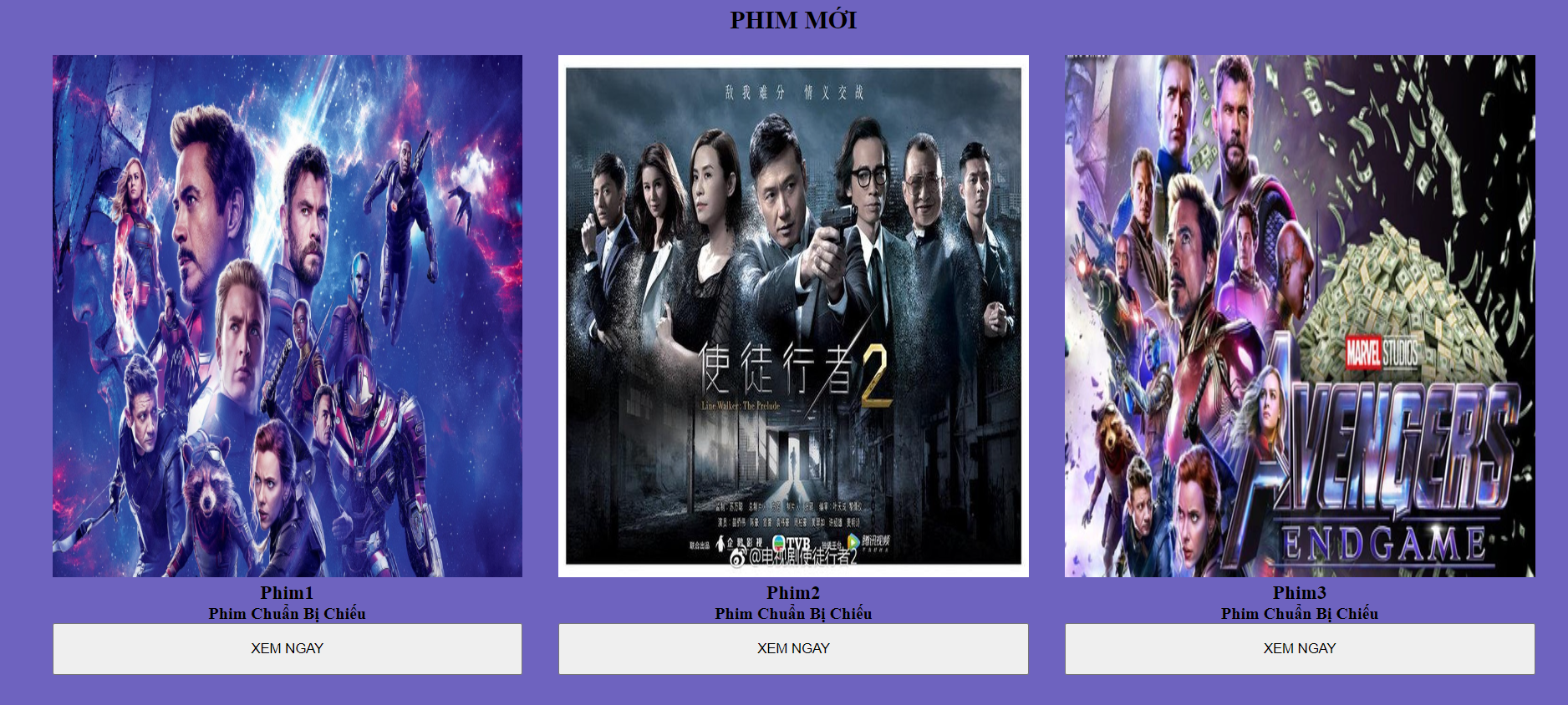
.anh{width: 450px; height: 500px;}

.section button{

margin: 0 auto;

width: 450px;

**4.5. kết quả đạt được**



<h2>PHIM NHIỀU NGƯỜI XEM</h2>

<div class="flim">

<div class="phim">

<img class="anh" src="./hqdefault.jpg">

<h3>Phim1</h3>

<h4>Đang Diễn Ra</h4>

<button onclick="location.href='xemphim.html'">XEM NGAY</button>

</div><br>

<div class="phim">

<img class="anh" src="./phim-canh-sat-hinh-su-tvb-1.jpg">

<h3>Phim2</h3>

<h4>Đang Diễn Ra</h4>

<button onclick="location.href='xemphim.html'">XEM NGAY</button>

</div><br>

<div class="phim">

<img class="anh" src="./0a91nq6a9i681694335589466.jpg">

<h3>Phim3</h3>

<h4>Đang Diễn Ra</h4>

<button onclick="location.href='xemphim.html'">XEM NGAY</button>

</div>

</div>

</section>

<section class="section">

Css:

h1{margin: 20px 15px 10px ;

text-align: center;

text-transform: uppercase;

font-size: 40px;}

main {padding: 50px;}

.section{margin-bottom: 50px;}

.section p{

text-align: center;

}

.section p,h2,h3{

text-align: center;

}

.section p,h4{

text-align: center;

}

.section h2{margin: 0 0 20px;}

.flim{display:flex;

flex-wrap: wrap;

justify-content: space-between;

}

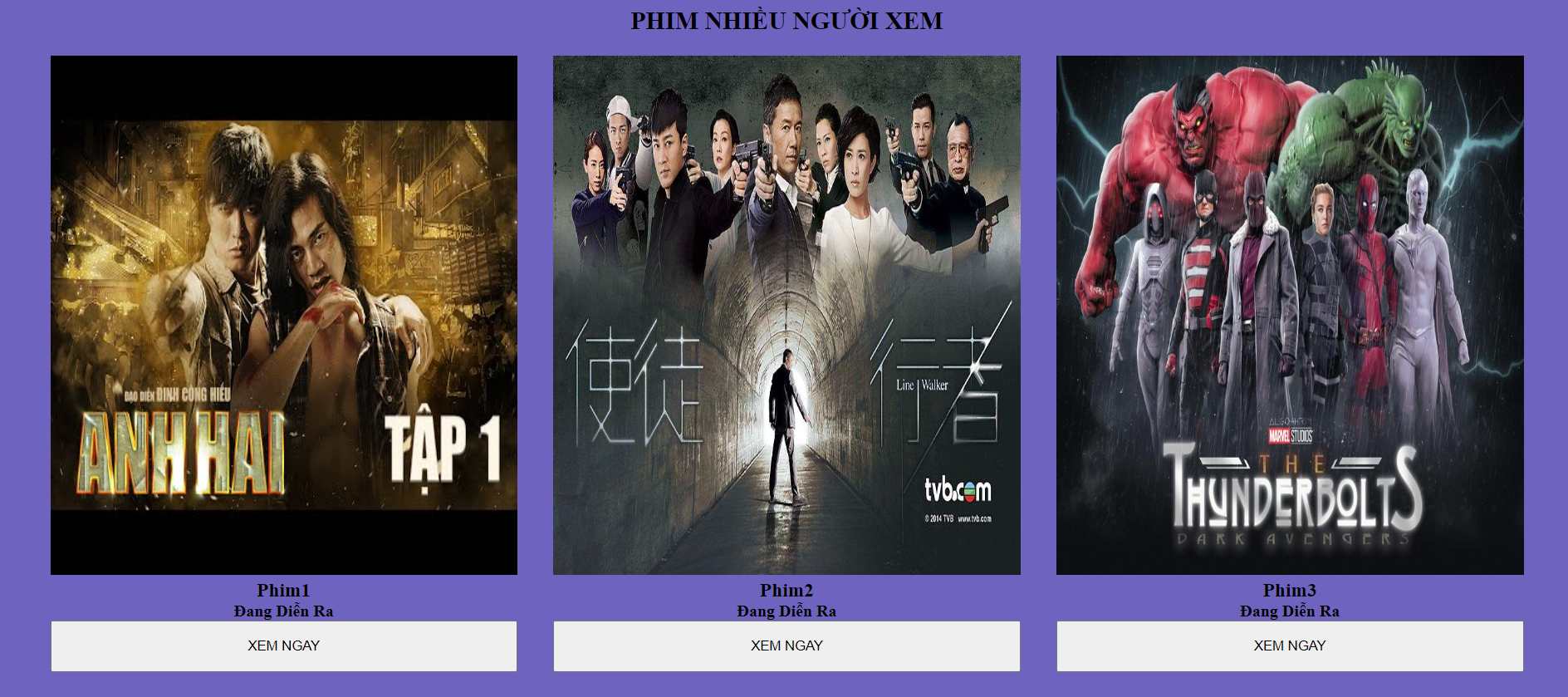
.anh{width: 450px; height: 500px;}

.section button{

margin: 0 auto;

width: 450px;

**4.5. kết quả đạt được**



<h2>PHIM SẮP CHIẾU RAP</h2>

<div class="flim">

<div class="phim">

<img class="anh" src="./0t5jre5hyh79169433541650

6.jpg">

<h3>Phim1</h3>

<h4>1vé / 70k</h4>

<button onclick="location.href='xemphim.html'">XEM NGAY</button>

</div><br>

<div class="phim">

<img class="anh" src="./578692Fa75ba9ce4c5ab.jpg">

<h3>Phim2</h3>

<h4>1vé / 70k</h4>

<button onclick="location.href='xemphim.html'">XEM NGAY

</button>

</div><br>

<div class="phim">

<img class="anh" src="./phim-truyen-hinh-viet-nam.jpg">

<h3>Phim3</h3>

<h4>1vé / 70k</h4>

<button onclick="location.href='xemphim.html'">XEM NGAY</button>

</div>

</div>

</section>

<section class="section">

Css:

h1{margin: 20px 15px 10px ;

text-align: center;

text-transform: uppercase;

font-size: 40px;}

main {padding: 50px;}

.section{margin-bottom: 50px;}

.section p{

text-align: center;

}

.section p,h2,h3{

text-align: center;

}

.section p,h4{

text-align: center;

}

.section h2{margin: 0 0 20px;}

.flim{display:flex;

flex-wrap: wrap;

justify-content: space-between;

}

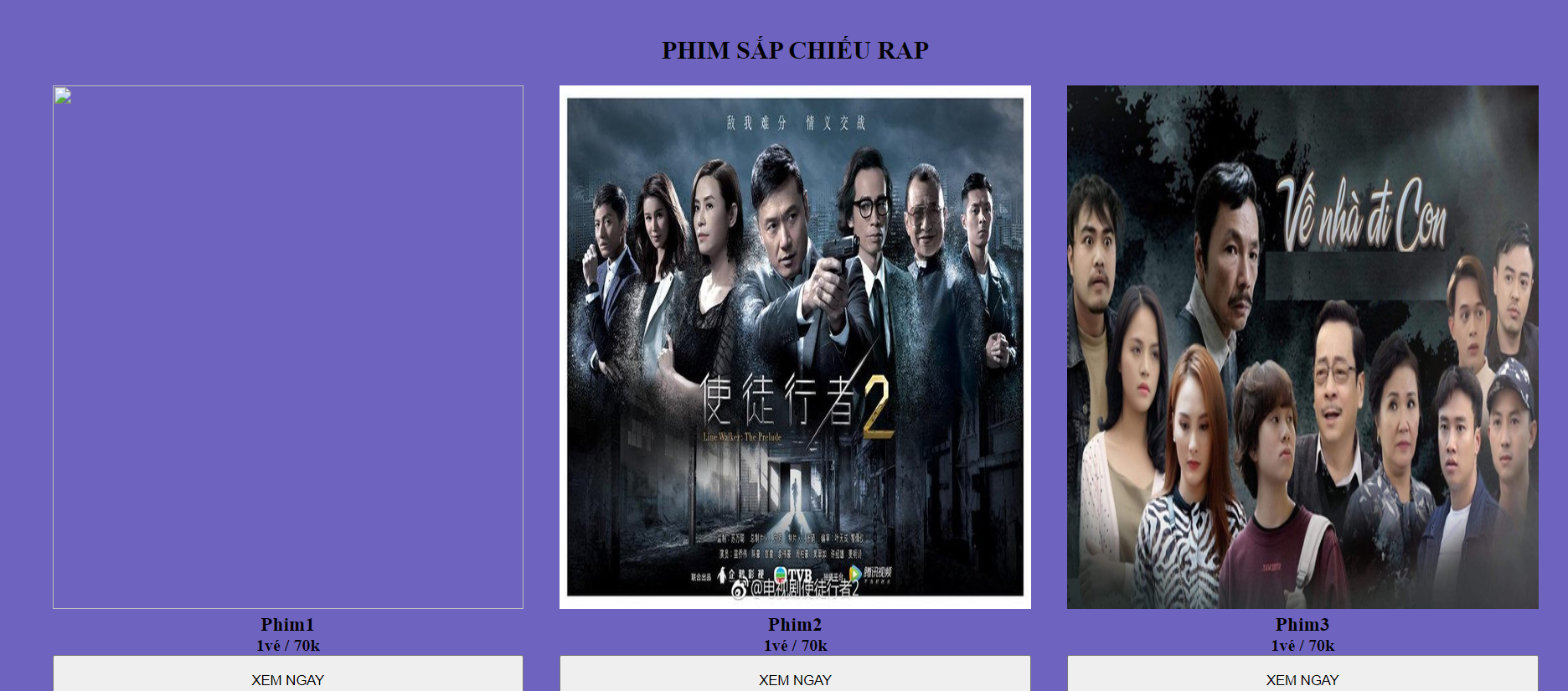
.anh{width: 450px; height: 500px;}

.section button{

margin: 0 auto;

width: 450px;

**4.5. kết quả đạt được**



<h2>Liên hệ mua vé</h2>

<form>

<label>Họ và tên</label>

<input type="text" id="text" name="name"><br>

<label>Email</label>

<input type="text" id="Email" name="Email"><br>

<label>Nội dung</label>

<input type="text" id="messeger" name="noidung">

</form>

</section>

<button type="muave">Mua Vé</button>

</main>

Css:

section form{

margin-left: 35%;

}

section form input{

margin-left: 50px;

width: 300px;

height: 30px;

margin-top: 20px;

}

#Email{

margin-left: 74px;

}

#messeger{

margin-left: 52px;

}

main button{

width: 100px;

height: 50px;

margin-left: 633px;

}

**4.5. kết quả đạt được**



<footer><p>&copy;Copy by PHIM OMEGA ONLINE</p></footer>

</body>

</html>

Css:

footer{

color:aliceblue;

background-color: red;

padding: 20px;

font-size: 30px;

text-align: center;

}

## Kết luận chương 4

Bằng cách sử dụng HTML và CSS, em đã thiết kế một giao diện người dùng mạnh mẽ và dễ sử dụng. Người dùng có thể dễ dàng tìm kiếm, duyệt và xem các bộ phim một cách thuận tiện. Trang web của em đã được tối ưu hóa để hiển thị đẹp trên nhiều thiết bị và kích thước màn hình khác nhau. Điều này đảm bảo rằng người dùng có thể truy cập và xem phim mọi lúc, mọi nơi từ các thiết bị di động đến máy tính để bàn.

## KẾT LUẬN

**Ưu điểm**: trang web xem phim có thể hoạt động trên hầu hết các trình duyệt web hiện đại mà không cần các phần mềm bổ sung, là ngôn ngữ dễ hiểu và dễ bảo trì, điều này có nghĩa là việc thay đổi hoặc cập nhật trang web xem phim trở nên dễ dàng hơn. Với sự phát triển của thiết bị di động, việc sử dụng HTML và CSS cho phép trang web xem phim hiển thị một cách mượt mà và tương thích trên các thiết bị di động.

**Nhược điểm**: có giới hạn trong việc tạo ra các tính năng tương tác phức tạp, như tìm kiếm nâng cao, gợi ý phim dựa trên lịch sử xem, hoặc xây dựng các chức năng đăng nhập và đăng ký. Khi trang web phát triển, việc mở rộng chức năng và tính năng có thể trở nên khó khăn với HTML và CSS đơn thuần, và có thể yêu cầu sự tích hợp của JavaScript . không thể quản lý dữ liệu lớn một cách hiệu quả như các cơ sở dữ liệu, điều này có thể gây ra vấn đề với việc quản lý nội dung phim đa dạng và lớn.

**Hướng phát triển**: Phát triển giao diện người dùng đáp ứng tốt trên các thiết bị di động, cung cấp trải nghiệm xem phim mượt mà và dễ dàng trên điện thoại di động và máy tính bảng. Kết hợp với các API hoặc dịch vụ dữ liệu phim để cung cấp thông tin chi tiết về các bộ phim, bao gồm hình ảnh, mô tả, diễn viên, và đánh giá từ các nguồn đáng tin cậy, sử dụng các kỹ thuật tối ưu hóa để giảm thời gian tải trang và tăng tốc độ trang web, bao gồm tải trước dữ liệu, tối ưu hóa hình ảnh và CSS, và sử dụng bộ nhớ đệm để giảm tải lại trang. Tiếp tục tìm kiếm và triển khai các tính năng mới để cải thiện trải nghiệm người dùng, bao gồm chức năng xem phim ngoại lệ, cải thiện tìm kiếm và gợi ý phim, và cung cấp tùy chọn tùy chỉnh và cá nhân hóa.

**DANH MỤC TÀI LIỆU THAM KHẢO**

1. Nguyễn Bá Đạt (2014), *giáo trình HTML5 và CSS3.*
2. W3Schools: <https://www.w3schools.com/>
3. <https://www.slideshare.net/toason91/bi-ging-kin-trc-my-tnh>