**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 WEBSITE CÁC TRÒ**

**CHƠI GAMES**

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

**Sinh viên thực hiện:**  **Lê Duy Hòa**

**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 WEBSITE CÁC TRÒ**

**CHƠI GAMES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
|  | 1771020238 | Lê Duy Hòa | 29/11/2005 |  |  |

### 

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

**Hà Nội, 2024**

**LỜI NÓI ĐẦU**

Xin chào thầy cô và các bạn, em tên là Lê Duy Hòa, học sinh lớp CNTT 17-01 ở trường đại học Đại Nam. Hôm nay, em xin phép được trình bày về bài tập lớn mà em đã hoàn thành ở môn Thiết kế Web.

Đầu tiên, em xin được phép trình bày về đề tài bài tập lớn mà em đã làm.

Đề tài mà em đã chọn là “thiết kế website các trò chơi games bằng HTML và CSS”.

Trong quá trình hoàn thành bài tập này, em đã sử dụng rất nhiều phương pháp nghiên cứu và nhiều công cụ tìm kiếm dữ liệu để có thể thu thập và xử lí dữ liệu. Em sẽ trình bày bài tập lớn một cách ngắn gọn và dễ hiểu.

Cuối cùng, em xin gửi lời cảm ơn đến thầy đã cung cấp cho em những kiến thức và định hướng để em có thể hoàn thành được bài tập lần này.

Mong rằng bài tập này sẽ đáp ứng được kì vọng của mọi người và nhận được đánh giá cao từ phía thầy cô và các bạn sinh viên. Xin cảm ơn!

**MỤC LỤC**

[**CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM** 10](#_Toc161688447)

[**1.1. Các khái niệm cơ bản:** 10](#_Toc161688448)

[**1.2. Phân biệt Web tĩnh, Web động:** 10](#_Toc161688449)

[*1.2.1. Web tĩnh:* 10](#_Toc161688450)

[*1.2.2. Web động:* 11](#_Toc161688451)

[**1.3. Một số thuật ngữ (Hosting, Tên miền-Domain Name-Web Server Name, HomePage,** 12](#_Toc161688452)

[**Web Site, WebServer, URL-Uniform Resource Location, Browser, …):** 12](#_Toc161688453)

[**1.4. Một số công cụ dùng thiết kế Web (Visual Studio code, Adobe Dreamweaver, …):** 13](#_Toc161688454)

[**Kết luận chương 1:** 14](#_Toc161688455)

[**CHƯƠNG II: HTML & HTML5** 15](#_Toc161688456)

[**2.1. Tổng quan về HTML:** 15](#_Toc161688457)

[**2.2. Cấu trúc tổng quát trang HTML:** 15](#_Toc161688458)

[**2.3. Các thẻ HTML thông dụng:** 16](#_Toc161688459)

[*2.3.1. Các thẻ xử lý văn bản:* 16](#_Toc161688460)

[*2.3.2. Các thẻ tạo bảng:* 17](#_Toc161688461)

[*2.3.3. Thẻ liên kết:* 18](#_Toc161688462)

[*2.3.4. Các thẻ đa phương tiện:* 18](#_Toc161688463)

[*2.3.5. Thẻ tạo khung:* 19](#_Toc161688464)

[**2.4. Các thẻ tạo biểu mẫu (form):** 19](#_Toc161688465)

[**2.5. Một số thẻ HTML đặc biệt:** 20](#_Toc161688466)

[**2.6. HTML5:** 22](#_Toc161688467)

[*2.6.1. HTML5 là gì?* 22](#_Toc161688468)

[*2.6.2. Cú pháp HTML5:* 22](#_Toc161688469)

[*2.6.3. Khai báo DOCTYPE trong HTML5:* 22](#_Toc161688470)

[*2.6.4. Thẻ ngữ nghĩa:* 23](#_Toc161688471)

[*2.6.5. Webform:* 23](#_Toc161688472)

[**Kết luận chương 2:** 23](#_Toc161688473)

[**CHƯƠNG III: CSS và CSS3** 25](#_Toc161688474)

[**3.1. CSS là gì?** 25](#_Toc161688475)

[**3.2. Cú pháp CSS:** 25](#_Toc161688476)

[**3.3. Áp dụng CSS vào trang HTML:** 25](#_Toc161688477)

[**3.4. Selectors:** 27](#_Toc161688478)

[*3.4.1. Universal selector:* 27](#_Toc161688479)

[*3.4.2. Type selector:* 28](#_Toc161688480)

[*3.4.3. Identity selector:* 28](#_Toc161688481)

[*3.4.4. Class selector:* 28](#_Toc161688482)

[*3.4.5. Descendant selector:* 28](#_Toc161688483)

[*3.4.6. Child selector:* 29](#_Toc161688484)

[*3.4.7. Adjacent sibling selector:* 29](#_Toc161688485)

[*3.4.8. Attribute selector:* 29](#_Toc161688486)

[*3.4.9. Pseudo class selector:* 30](#_Toc161688487)

[*3.4.10. Group selector:* 30](#_Toc161688488)

[**3.5. Đơn vị đo lường CSS:** 30](#_Toc161688489)

[**3.6. Kế thừa thuộc tính:** 31](#_Toc161688490)

[**3.7. Các nhóm thuộc tính trong CSS:** 31](#_Toc161688491)

[*3.7.1. Thuộc tính định dạng font chữ, văn bản (font, text):* 31](#_Toc161688492)

[*3.7.2. Thuộc tính định dạng nền (background):* 32](#_Toc161688493)

[*3.7.3. Mô hình hộp (box model):* 33](#_Toc161688494)

[*3.7.4. Các thuộc tính định margin, padding, border:* 33](#_Toc161688495)

[*3.7.5. Thuộc tính định dạng cách hiển thị (display):* 33](#_Toc161688496)

[*3.7.6. Thuộc tính xác định vị trí (position):* 34](#_Toc161688497)

[**3.8. Float & Clear:** 35](#_Toc161688498)

[**3.9. Flex:** 35](#_Toc161688499)

[**3.10. Grid:** 36](#_Toc161688500)

[**3.11. CSS3:** 36](#_Toc161688501)

[**3.12. SCSS:** 40](#_Toc161688502)

[**3.13. SASS:** 40](#_Toc161688503)

[**Kết luận chương 3:** 41](#_Toc161688504)

[**CHƯƠNG 4: THIẾT KẾ WEBSITE CÁC TRÒ CHƠI GAMES** 42](#_Toc161688505)

[**4.1. Ý tưởng của Website:** 42](#_Toc161688506)

[**4.2. Xây dựng bố cục của trang Web:** 42](#_Toc161688507)

[**4.3. Thiết kế trang Web bằng HTML và CSS:** 43](#_Toc161688508)

[**4.4. Kết quả đạt được:** 45](#_Toc161688509)

[**Kết luận chương 4:** 46](#_Toc161688510)

[**KẾT LUẬN** 48](#_Toc161688511)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 49](#_Toc161688512)

**BẢNG CÁC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **TỪ VIẾT TẮT** | **VIẾT ĐẦY ĐỦ** |
| **1** | **HTML** | **Hyper Text Markup Language** |
| **2** | **CSS** | **Cascading Style Sheets** |
| **3** | **URL** | **Uniform Resource Location** |
| **4** | **PHP** | **Hypertext Preprocesso** |
| **5** | **API** | **Application Programming Interface** |
| **6** | **TLD** | **Top Level Domain** |
| **7** | **WWW** | **World Wide Web** |
| **8** | **SEO** | **Search Engine Optimization** |

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

**1.1. Các khái niệm cơ bản:**

Thiết kế web là quá trình tạo ra giao diện người dùng của một trang web để cung cấp trải nghiệm tốt nhất cho người dùng cuối. Nó không chỉ đơn thuần là việc tạo ra một

giao diện đẹp mắt, mà còn liên quan đến việc cân nhắc về trải nghiệm người dùng, tối ưu hóa hiệu suất, và tương tác với nội dung.

Một số khía cạnh cơ bản của thiết kế web có thể kể đến như:

1. Phân tích yêu cầu: Hiểu rõ nhu cầu và mục tiêu của dự án, bao gồm đối tượng người dùng, mục đích của trang web, và các chức năng cần thiết.

2. Thiết kế giao diện người dùng: Xác định cách tổ chức và trình bày các yếu tố trên trang web để tạo ra một giao diện hấp dẫn và dễ sử dụng. Điều này bao gồm việc chọn màu sắc, font chữ, hình ảnh, và bố cục.

3. Trải nghiệm người dùng: Tạo ra một trải nghiệm người dùng tốt bằng cách đảm bảo rằng trang web dễ sử dụng, tương tác mượt mà và cung cấp giá trị cho người dùng.

4. Phát triển mã nguồn: Chuyển đổi thiết kế thành mã nguồn HTML, CSS và JavaScript để tạo ra giao diện trực quan và tương tác.

5. Phát triển chức năng: Nếu cần, phát triển các chức năng phức tạp hoặc tương tác với cơ sở dữ liệu bằng các ngôn ngữ lập trình như PHP, Python hoặc Node.js.

6. Kiểm thử và sửa lỗi: Kiểm tra kỹ lưỡng trang web trên nhiều trình duyệt và thiết bị khác nhau để đảm bảo tính tương thích và sửa lỗi nếu cần.

7. Triển khai và duy trì: Triển khai trang web và duy trì nó để đảm bảo hoạt động ổn định và cập nhật theo nhu cầu của người dùng.

8. Tối ưu hoá SEO: Đảm bảo rằng trang web được tối ưu hóa để có thứ hạng cao trên các công cụ tìm kiếm như Google.

## **1.2. Phân biệt Web tĩnh, Web động:**

### *1.2.1. Web tĩnh:*

Một trang web tĩnh là một trang web mà nội dung của nó không thay đổi dựa trên hành động của người dùng hoặc dữ liệu từ cơ sở dữ liệu hoặc các nguồn dữ liệu động khác. Trang web tĩnh được tạo ra bằng cách sử dụng các tập tin HTML, CSS và JavaScript mà không cần sử dụng các ngôn ngữ lập trình phía máy chủ như PHP, Ruby hoặc Python.

Các trang web tĩnh thường rất đơn giản và dễ dàng để triển khai. Chúng thích hợp cho các trang web nhỏ, cá nhân hoặc doanh nghiệp có nhu cầu hiển thị thông tin cơ bản, như trang giới thiệu, trang sản phẩm, trang liên hệ, và các trang web tĩnh khác.

Ưu điểm:

- Dễ dàng phát triển và triển khai.

- Hiệu suất cao và tải nhanh do không cần tương tác với máy chủ để lấy dữ liệu.

- Chi phí thấp hơn so với các trang web động.

- Dễ dàng để quản lí và bảo trì.

Tuy nhiên, các trang web tĩnh có hạn chế trong việc cung cấp tính năng tương tác phức tạp và không thể tự động hóa việc cập nhật nội dung một cách động. Do đó, khi cần

tính năng động và tương tác cao, thì việc sử dụng các framework và ngôn ngữ lập trình phía máy chủ sẽ là lựa chọn phù hợp hơn.

### *1.2.2. Web động:*

Web động là một trang web được tạo ra bằng cách sử dụng các ngôn ngữ lập trình như PHP, Python, Ruby, hoặc Node.js để tạo ra nội dung động và tương tác với người dùng. Trái ngược với trang web tĩnh (static website) - nơi mọi nội dung được tạo ra trước và chỉ thay đổi khi trang web được cập nhật bằng tay - trang web động cho phép tạo ra nội dung một cách tự động dựa trên yêu cầu của người dùng hoặc các điều kiện khác.

Các tính năng của web động:

- Nội dung động: Trang web có thể tạo ra nội dung mới dựa trên dữ liệu từ các nguồn khác nhau như cơ sở dữ liệu, API hoặc các tập tin dữ liệu.

- Tương tác người dùng: Trang web động thường có khả năng tương tác với người dùng, cho phép họ tương tác với các biểu mẫu, gửi dữ liệu, hoặc thực hiện các hành động khác như tải lên tệp tin.

- Quản lí dữ liệu: Trang web động thường có khả năng quản lý và xử lý dữ liệu từ người dùng hoặc từ các nguồn dữ liệu khác, thông qua việc sử dụng cơ sở dữ liệu hoặc các dịch vụ lưu trữ dữ liệu khác.

- Tuỳ biến linh hoạt: Do tính linh hoạt của mã nguồn, trang web động thường dễ dàng tùy chỉnh và mở rộng theo nhu cầu của dự án hoặc doanh nghiệp.

## **1.3. Một số thuật ngữ (Hosting, Tên miền-Domain Name-Web Server Name, HomePage,**

## **Web Site, WebServer, URL-Uniform Resource Location, Browser, …):**

Hosting: là dịch vụ cung cấp không gian lưu trữ tài nguyên trên máy chủ cho các trang web và ứng dụng web. Dịch vụ Hosting cho phép các trang web được truy cập và hiển thị trên Internet thông qua mạng lưới các máy chủ toàn cầu.

Tên miền- Domain Name- Web Server Name:

- Tên miền (Domain Name) là một chuỗi các ký tự được sử dụng để xác định một địa chỉ duy nhất trên Internet. Mỗi trang web có một tên miền duy nhất giúp người dùng truy cập vào trang web bằng cách nhập tên miền vào thanh địa chỉ trình duyệt. Ví dụ, trong tên miền "example.com", "example" là tên của trang web, còn ".com" là phần mở rộng tên miền (Top-Level Domain - TLD).

- Web Server Name là tên của máy chủ web hoặc máy chủ mạng mà trang web được lưu trữ và chạy trên đó. Đây là máy chủ mà các yêu cầu từ trình duyệt web được gửi đến và các trang web được truy cập thông qua nó. Tên máy chủ web thường không được hiển thị rõ ràng cho người dùng, mà thường được quản lý bởi nhà cung cấp dịch vụ hosting hoặc quản trị hệ thống.

HomePage: HomePage là trang chính hoặc trang đầu tiên mà người dùng thấy khi truy cập vào một trang web. Nó thường là trang mà nhà phát triển web hoặc chủ sở hữu trang web thiết lập làm điểm xuất phát cho trải nghiệm người dùng. HomePage thường chứa các thông tin quan trọng như thông tin giới thiệu, menu điều hướng, nội dung chính, …

Website: là một bộ sưu tập các trang điện tử liên kết với nhau và được lưu trữ trên Internet. Mỗi trang web có thể chứa nhiều loại nội dung, bao gồm văn bản, hình ảnh, video, âm thanh và các yếu tố tương tác.

WebServer: là một phần mềm hoặc máy chủ (server) máy tính có nhiệm vụ lưu trữ, xử lý và phục vụ các trang web và nội dung web cho các máy tính khác (client) thông qua mạng Internet hoặc mạng nội bộ.

URL-Uniform Resource Location: là một chuỗi ký tự được sử dụng để chỉ định vị trí và định danh của một tài nguyên trên Internet. URL thường được sử dụng để truy cập các trang web, tập tin, hình ảnh, video và các tài nguyên khác trên mạng.

Browser: là một phần mềm cho phép người dùng truy cập và duyệt các trang web trên Internet. Trình duyệt web hoạt động bằng cách gửi yêu cầu đến các máy chủ web, sau đó nhận phản hồi từ các máy chủ đó và hiển thị nội dung trang web trên màn hình của người dùng.

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

- Visual Studio Code: là một trình biên tập mã nguồn mở và miễn phí được phát triển bởi Microsoft. Nó là một ứng dụng chuyên nghiệp dành cho phát triển phần mềm và hỗ trợ nhiều ngôn ngữ lập trình và framework khác nhau.

Một số đặc điểm nổi bật:

+ Dễ sử dụng.

+ Hỗ trợ nhiều ngôn ngữ và framework.

+ Mở rộng bằng các Extention.

+ Hỗ trợ Debugging

+ Tích hợp Git.

- Adobe Dreamweaver: là một ứng dụng phần mềm dành cho phát triển và thiết kế trang web. Nó được phát triển và phân phối bởi Adobe Systems và là một trong những công cụ phổ biến nhất trong lĩnh vực phát triển web.

Một số đặc điểm nổi bật:

+ Trình biên tập mã linh hoạt.

+ Giao diện thân thiên với người dùng.

+ Hỗ trợ thiết kế giao diện WYSIWYG.

+ Tích hợp Adobe Creative Cloud.

+ Hỗ trợ thiết kế đa nền tảng và đáp ứng.

# **Kết luận chương 1:**

Trong cuộc cách mạng công nghệ hiện nay, việc thiết kế web và triển khai hệ thống phần mềm không chỉ đơn giản là việc tạo ra giao diện đẹp mắt hoặc phát triển các chức năng phức tạp. Đây là quá trình kết hợp nhiều yếu tố khác nhau như trải nghiệm người dùng, tối ưu hóa hiệu suất, bảo mật thông tin, và tích hợp các công nghệ mới.

Thiết kế web: Đây là quá trình tạo ra giao diện người dùng đẹp mắt, dễ sử dụng và có trải nghiệm người dùng tốt. Thiết kế web cũng bao gồm việc tối ưu hóa cho các thiết bị di động, tương thích trên nhiều trình duyệt và nền tảng khác nhau.

Triển khai hệ thống phần mềm: Sau khi thiết kế hoàn thành, việc triển khai hệ thống phần mềm là quá trình triển khai các thành phần của ứng dụng và đảm bảo rằng nó hoạt động một cách hiệu quả trên môi trường sản xuất. Quá trình này bao gồm việc lựa chọn hạ tầng máy chủ, cài đặt và cấu hình phần mềm, kiểm tra tích hợp và triển khai.

Kết hợp cả hai quá trình này yêu cầu sự hiểu biết sâu rộng về công nghệ, kỹ năng thiết kế và lập trình, cũng như khả năng làm việc trong nhóm và quản lý dự án. Đồng thời, việc duy trì và cập nhật hệ thống sau khi triển khai cũng rất quan trọng để đảm bảo rằng nó vẫn hoạt động ổn định và đáp ứng được nhu cầu của người dùng.

# **CHƯƠNG II: HTML & HTML5**

(Hyper Text Markup Language)

## **2.1. Tổng quan về HTML:**

HTML là viết tắt của Hyper Text Markup Language, là ngôn ngữ đánh dấu tiêu chuẩn được sử dụng để tạo cấu trúc và định dạng nội dung trên các trang web. HTML cung cấp các phần tử và cú pháp để biểu diễn thông tin trên trang web, từ văn bản và hình ảnh đến các liên kết và biểu mẫu.

## **2.2. Cấu trúc tổng quát trang HTML:**

Cấu trúc tổng quan của trang HTML gồm có những thành phần cơ bản như sau:

a) Khai báo DOCTYPE:  
 Đây là khai báo đầu tiên trong một tài liệu HTML và nó xác định phiên bản HTML mà trang sử dụng. Điều này giúp trình duyệt hiểu cách diễn giải mã HTML trong tài liệu.

VD: <!DOCTYPE html>

b) Thẻ <html>:

Thẻ <html> là thẻ gốc của mọi trang HTML và bao gồm toàn bộ nội dung của trang.

VD:

<html lang="en">

<!-- Nội dung trang -->

</html>

c) Thẻ <head>:

Thẻ <head> chứa các thông tin meta, tiêu đề trang, và liên kết tới các tập tin CSS hoặc Java Script.

VD;

<head>

<!-- Thông tin meta, tiêu đề trang, và liên kết tới các tập tin CSS/JavaScript -->

</head>

d) Thẻ <title>:

Thẻ <title> xác định tiêu đề của trang, mà sẽ hiển thị trên thanh tiêu đề của trình duyệt.

VD:  
 <title>Tiêu đề của trang</title>

e) Thẻ <body>:

Thẻ <body> chứa toàn bộ nội dung hiển thị trên trang web, bao gồm văn bản, hình ảnh, liên kết, biểu mẫu, …

VD:  
 <body>

<!-- Nội dung trang -->

</body>

f) Thẻ <footer>:

Thẻ <footer> chứa thông tin chân trang web, thường bao gồm các liên kết đến trang chủ, thông tin về tác giả, bản quyền, …

VD:

<footer>

<!-- Thông tin chân trang -->

</footer>

## **2.3. Các thẻ HTML thông dụng:**

### *2.3.1. Các thẻ xử lý văn bản:*

Dưới đây là một số thẻ xử lí văn bản thông dụng trong HTML:

a) <p> - Đoạn văn bản: Sử dụng để định dạng đoạn văn bản.

VD:

<p>Đoạn văn bản<p>

b) <h1>, <h2>, …, <h6> - Tiêu đề: Sử dụng để tạo các tiêu đề với cỡ chữ khác nhau.

VD:

<h1>Tiêu đề lớn nhất</h1>

<h2>Tiêu đề lớn hơn</h2>

c) <br> - Xuống dòng: Sử dụng để tạo một dòng mới.

VD:

Dòng 1.<br>

Dòng 2.

d) <b> - Chữ in đậm: Sử dụng để làm cho văn bản đậm.

VD:

<b>Văn bản in đậm</b>

e) <i> - In nghiêng: Sử dụng để làm cho văn bản in nghiêng.

VD:

<i>Văn bản in nghiêng</i>

f) <u> - gạch chân: Sử dụng để gạch chân văn bản.

VD:

<u>Văn bản gạch chân</u>

### *2.3.2. Các thẻ tạo bảng:*

Trong HTML, có một số thẻ được sử dụng để tạo và định dạng bảng trên trang web. Dưới đây là các thẻ thông dụng được sử dụng để tạo bảng:

a) <table> : Được sử dụng để bắt đầu và kết thúc một bảng.

VD:  
 <table>

<!-- Các hàng và cột trong bảng -->

</table>

b) <tr>: Đại diện cho một hàng trong bảng.

VD:

<tr>

<!-- Các ô trong hàng -->

</tr>

c) <th> : Được sử dụng để định dạng ô tiêu đề trong bảng. Thông thường được sử dụng cho các ô tiêu đề của các cột hoặc hàng.

VD:

<th>Tiêu đề cột</th>

d) <td>: Đại diện cho một ô trong bảng, được sử dụng cho dữ liệu trong các hàng và cột.

VD:  
 <td>Dữ liệu</td>

e) <caption> : Được sử dụng để thêm một tiêu đề cho bảng.

VD:

<caption>Tiêu đề bảng</caption>

### *2.3.3. Thẻ liên kết:*

Trong HTML, có một số thẻ được sử dụng để tạo các liên kết hoặc đường dẫn đến các trang web khác, tài nguyên trên cùng trang hoặc vị trí khác trong cùng một trang web. Dưới đây là các thẻ liên kết thông dụng:

a) <a> - Anchor (Liên kết): Được sử dụng để tạo một liên kết đến một URL, một tài nguyên trên cùng trang hoặc vị trí khác trong cùng một trang web.

VD:

<a href="https://www.google.com">Liên kết đến trang web</a>

b) <link> - Liên kết tài nguyên ngoại vi: Được sử dụng để liên kết với các tài nguyên bên ngoài như tập tin CSS hoặc biểu tượng trang web.

VD:  
 <link rel="stylesheet" href="styles.css">

c) <base> - Cơ sở URL: Được sử dụng để chỉ định một URL cơ sở cho tất cả các liên kết trong trang.

VD:  
 <base href="https://www.example.com/">

d) <script> - Kịch bản liên kết: Được sử dụng để liên kết đến một tập tin script hoặc mã kịch bản từ một URL bên ngoài.

VD:

<script src="script.js"></script>

### *2.3.4. Các thẻ đa phương tiện:*

Trong HTML, có một số thẻ được sử dụng để nhúng và điều khiển các phương tiện đa phương tiện như hình ảnh, video và âm thanh. Một số thẻ đa phương tiện cơ bản như:

a) <img> - Hình ảnh: Được sử dụng để hiển thị hình ảnh trên trang web.

VD:

<img src="image.jpg" alt="Mô tả hình ảnh">

b) <video> - Video: Được sử dụng để nhúng và phát video vào trang web.

VD:

<video controls>

<source src="video.mp4" type="video/mp4">

Your browser does not support the video tag.

</video>

c) <audio> - Âm thanh: Được sử dụng để nhúng và phát âm thanh trên trang web.

VD:

<audio controls>

<source src="audio.mp3" type="audio/mpeg">

Your browser does not support the audio tag.

</audio>

d) <iframe> - Inline Frame: Được sử dụng để nhúng một trang web hoặc tài liệu HTML khác vào trong trang web hiện tại.

VD:  
 <iframe src="page.html" width="600" height="400"></iframe>

### *2.3.5. Thẻ tạo khung:*

Trong HTML, để tạo một khung hoặc một vùng có đường viền để làm nổi bật hoặc phân chia nội dung, bạn có thể sử dụng thẻ <div> kết hợp với CSS để tạo khung

VD:

<div>

<p>Nội dung</p>

</div>

## **2.4. Các thẻ tạo biểu mẫu (form):**

Trong HTML, có một số thẻ được sử dụng để tạo biểu mẫu hoặc form để người dùng có thể nhập thông tin hoặc thực hiện các hành động trên trang web. Dưới đây là các thẻ tạo biểu mẫu thông dụng:

a) <form> - Biểu mẫu:

Thẻ <form> được sử dụng để tạo một biểu mẫu, trong đó chúng ta có thể đặt các trường nhập liệu hoặc các nút gửi.

VD:

<form action="/submit\_form" method="post">

<!-- Các trường nhập liệu và nút gửi -->

</form>

b) <input> - Trường nhập liệu:

Thẻ <input> được sử dụng để tạo các trường nhập liệu như ô văn bản, ô chọn, ô mật khẩu, …

VD:  
 <input type="text" name="username" placeholder="Tên người dùng">

c) <textarea> - Ô nhập văn bản:

Thẻ <textarea> được sử dụng để tạo ô nhập văn bản lớn hơn.

VD:  
 <textarea name="message" rows="4" cols="50">Nhập nội dung ở đây</textarea>

d) <select> và <option> - Ô chọn:

Thẻ <select> được sử dụng để tạo một ô chọn tronh khi thẻ <option> được sử dụng để các định các tuỳ chọn trong đó.

VD:

<select name="city">

<option value="hanoi">Hà Nội</option>

<option value="hochiminh">Hồ Chí Minh</option>

</select>

## **2.5. Một số thẻ HTML đặc biệt:**

Trong HTML có một số thẻ đặc biệt. Dưới đây là một số thẻ đặc biệt:

a) <meta> - Thẻ meta:

Thẻ <meta> được sử dụng để cung cấp thông tin metadata về trang web, chẳng hạn như charset, mô tả trang, từ khóa, và các dữ liệu khác cho máy chủ và trình duyệt web.

VD:

<meta charset="UTF-8">

<meta name="description" content="Mô tả trang web">

<meta name="keywords" content="từ khóa, liên quan">

b) <base> - Thẻ base:

Thẻ <base> được sử dụng để xác định một URL cơ sở cho tất cả các URL tương đối được sử dụng trong trang web.

VD:  
 <base href="https://www.example.com/">

c) <link> - Thẻ link:

Thẻ <link> được sử dụng để liên kết với các tài nguyên bên ngoài như tập tin CSS, biểu tượng trang web, …

VD:  
 <link rel="stylesheet" href="styles.css">

d) <link> - Thẻ link:

Thẻ <link> được sử dụng để liên kết với các tài nguyên bên ngoài như tập tin CSS, biểu tượng trang web, …

VD:  
 <link rel="stylesheet" href="styles.css">

e) <iframe> - Thẻ Inline Frame:

Thẻ <iframe> được sử dụng để nhúng một trang web hoặc một phần của một trang web khác vào trang web của bạn.

VD:  
 <iframe src="https://www.example.com"></iframe>

f) <details> và <summary> - Thẻ Details và Summary:

Thẻ <details> được sử dụng để tạo một phần chi tiết có thể được mở rộng hoặc thu gọn, trong khi <summary> được sử dụng để xác định tiêu đề của phần chi tiết đó.

VD:

<details>

<summary>Thông tin chi tiết</summary>

<p>Nội dung chi tiết ở đây</p>

## **2.6. HTML5:**

### *2.6.1. HTML5 là gì?*

**HTML5 là một ngôn ngữ lập trình được phát triển trên nền tảng ngôn ngữ HTML và**  **quan trọng nhất của World Wide Web (WWW).** Nó được sử dụng để thiết kế và cấu trúc các website, hỗ trợ cho đa phương tiện tối đa nhưng vẫn giúp cho website thân thiện với mọi người dùng và mọi thiết bị, các chương trình máy tính, trình duyệt web…

### *2.6.2. Cú pháp HTML5:*

Cú pháp cơ bản của HTML5 giống với các phiên bản HTML trước đó, nhưng HTML5 có một số cải tiến và thêm mới.

VD:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

</head>

<body>

<!-- Nội dung của trang web -->

</body>

</html>

### *2.6.3. Khai báo DOCTYPE trong HTML5:*

Để khai báo DOCTYPE cho HTML5, bạn chỉ cần sử dụng một dòng đơn giản như sau ở đầu tài liệu HTML:

VD:

<!DOCTYPE html>

Đoạn mã trên cung cấp cho trình duyệt thông tin rằng tài liệu HTML đang sử dụng định dạng HTML5. DOCTYPE HTML5 không cần bất kỳ thông số hoặc tùy chọn nào khác, và nó là một cách tiện lợi và ngắn gọn để bắt đầu viết mã HTML5.

### *2.6.4. Thẻ ngữ nghĩa:*

Trong HTML5, thẻ ngữ nghĩa được sử dụng để đánh dấu các phần tử trong trang web với ý nghĩa cụ thể hơn, giúp cho trình duyệt và các công cụ khác (như công cụ tìm kiếm) hiểu được nội dung của trang web một cách chính xác hơn. Điều này không chỉ giúp cải thiện SEO mà còn tạo ra một cấu trúc logic hơn cho trang web. Dưới đây là một số thẻ ngữ nghĩa thường được sử dụng trong HTML5:

- <header> : Đánh dấu phần tiêu đề của một phần tử hoặc trang web.

- <nav> : Đánh dấu phần điều hướng (menu) của trang web.

- <main> : Đánh dấu phần nội dung chính của trang web.

- <section> : Đánh dấu một phần của trang web, thường bao gồm một loại các nội dung liên quan với nhau.

- <article> : Đánh dấu một bài viết hoặc nội dung độc lập có thể đứng một mình.

- <aside> : Đánh dấu phần nội dung phụ, thường chứa các thông tin bổ sung như sidebar.

- <footer> : Đánh dấu phần cuối trang web, thường chứa thông tin về tác giả, bản quyền và các liên kết liên quan.

### *2.6.5. Webform:*

Trong HTML5, một webform (biểu mẫu web) là một phần tử cho phép người dùng nhập dữ liệu và gửi nó đến máy chủ để xử lý. Web form là một phần quan trọng của các trang web tương tác, cho phép người dùng tương tác với trang web bằng cách nhập thông tin, chẳng hạn như tên, email, mật khẩu, ô chọn, v.v. và sau đó gửi thông tin này đến máy chủ để xử lý.

Một webform có thể chứa các phần tử như ô văn bản ( <input type=”text”> ), ô email ( input type=”email”> ), ô mật khẩu ( <input type=”password”> ), nút gửi ( <input type=”submit”> ), ô chọn ( <select> hoặc <option> ), ô kiểm ( <input type=”checkbox”> ), ô radio ( <input type=”radio”> ), …

# **Kết luận chương 2:**

HTML và HTML5 đều là các ngôn ngữ đánh dấu quan trọng trong việc phát triển các trang web và ứng dụng web.

Trong khi HTML cung cấp một cơ sở cho việc phát triển web, HTML5 nâng cao và mở rộng các khả năng của nó để đáp ứng được các yêu cầu ngày càng phức tạp của web hiện đại. Sự phát triển của HTML5 đã đóng vai trò quan trọng trong việc tạo ra các trải nghiệm web đa dạng và tương tác cho người dùng.

**CHƯƠNG III: CSS và CSS3**

## **3.1. CSS là gì?**

CSS là viết tắt của "Cascading Style Sheets" (Bảng kiểu dáng nối tiếp), là một ngôn ngữ lập trình được sử dụng để định dạng và trang trí cho các trang web. CSS giúp bạn tạo ra các giao diện đẹp mắt và linh hoạt bằng cách xác định cách mà các phần tử HTML được hiển thị trên trình duyệt.

Với CSS, bạn có thể:

- Định dạng văn bản: Thay đổi kiểu, kích thước, màu sắc, căn chỉnh, v.v. của văn bản trên trang web.

- Định dạng màu sắc và hình ảnh: Điều chỉnh kích thước, màu sắc, độ trong suốt và các thuộc tính khác của hình ảnh và nền.

- Định dạng bố cục: Xác định vị trí, kích thước, và khoảng cách giữa các phần tử HTML trên trang web.

- Tạo hiệu ứng: Áp dụng các hiệu ứng như chuyển động, đổ bóng, viền, và chuyển đổi cho các phần tử HTML.

- Responsive design: Tạo ra các giao diện linh hoạt có thể thích ứng với kích thước và thiết bị của người dùng.

## **3.2. Cú pháp CSS:**

Cú pháp CSS được sử dụng để định dạng và trang trí các phần tử trên trang web. Dưới đây là một ví dụ về cú pháp cơ bản của CSS:

selector {

property: value;

}

Trong đó:

- selector: Là phần chọn các phần tử HTML mà bạn muốn áp dụng các quy tắc CSS.

- property: Là thuộc tính mà bạn muốn định dạng cho các phần tử được chọn.

- value: Là giá trị của thuộc tính đó.

## **3.3. Áp dụng CSS vào trang HTML:**

Để có thể áp dụng CSS và HTML, chúng ta có thể làm theo những cách sau:

a) Sử dụng thuộc tính <style> trong các phần tử HTML.

VD:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Tuỳ chọn</title>

</head>

<body>

<h1 style="color: red;">Tiêu đề màu đỏ</h1>

<p style="font-size: 16px;">Đây là một đoạn văn bản có kích thước 16px</p>

</body>

</html>

b) Sử dụng phần tử <style> trong phần <head> của trang HTML:

VD:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Tuỳ chọn</title>

<style>

h1 {

color: red;

}

p {

font-size: 16px;

}

</style>

</head>

<body>

<h1>Nguyễn Văn A</h1>

<p>Văn bản</p>

</body>

</html>

c) Sử dụng file CSS riên biệt và liên kết nó với trang HTML bằng thẻ <link>

VD:

File HTML:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Áp dụng CSS vào HTML</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<h1>Tiêu đề màu đỏ</h1>

<p>Đây là một đoạn văn bản có kích thước 16px</p>

</body>

</html>

Trong khi đó, file CSS (styles.css) sẽ là:

h1 {

color: red;

}

p {

font-size: 16px;

}

## **3.4. Selectors:**

Trong CSS, selectors là các phần tử mà CSS sử dụng để xác định các phần tử HTML nào sẽ áp dụng các quy tắc CSS.

Một số loại selector phổ biến như: Element selector, class selector, ID selector, …

### *3.4.1. Universal selector:*

Universal selector trong CSS được sử dụng để áp dụng các quy tắc CSS cho tất cả các phần tử trong tài liệu HTML. Khi sử dụng universal selector, mọi phần tử trong tài liệu HTML đều sẽ áp dụng các thuộc tính CSS được chỉ định.

VD:

\* {

margin: 0;

padding: 0;

}

Tuy nhiên, cần lưu ý rằng việc sử dụng universal selector có thể ảnh hưởng đến hiệu suất của trang web, đặc biệt là khi trang web có nhiều phần tử. Do đó, nên sử dụng universal selector một cách cẩn thận và chỉ khi thực sự cần thiết.

### *3.4.2. Type selector:*

Type selector trong CSS là một loại selector được sử dụng để chọn tất cả các phần tử có cùng loại (type). Điều này có nghĩa là selector sẽ áp dụng các quy tắc CSS cho tất cả các phần tử có tên (type) được chỉ định.

VD:

p {

color: blue;

}

Type selector là một trong những loại selector phổ biến nhất trong CSS và được sử dụng rộng rãi để định dạng văn bản, hình ảnh, và các phần tử khác trên trang web.

### *3.4.3. Identity selector:*

Identity selector được sử dụng để chọn một phần tử HTML duy nhất dựa trên giá trị của thuộc tính ID của nó. ID selector được biểu diễn bằng cách sử dụng ký tự # sau đó là giá trị của ID.

VD:

#header {

color: blue;

}

### *3.4.4. Class selector:*

Class selector được sử dụng để chọn tất cả các phần tử có cùng giá trị của thuộc tính class attribute. Class selector được biểu diễn bằng cách sử dụng dấu chấm “.” sau đó là giá trị của class.

VD:

.highlight {

background-color: yellow;

}

### *3.4.5. Descendant selector:*

Descendant selector trong CSS được sử dụng để chọn các phần tử con (descendants) của một phần tử khác. Điều này có nghĩa là các quy tắc CSS được áp dụng cho các phần tử con của một phần tử mẹ cụ thể.

Định dạng của Descendant selector là “parentElement childElement” trong đó “parentElement” là phần tử mẹ và “childElement” là phần tử con.

### *3.4.6. Child selector:*

Child selector trong CSS là một loại selector được sử dụng để chọn các phần tử con trực tiếp của một phần tử mẹ cụ thể. Điều này có nghĩa là chỉ các phần tử con ngay lập tức, không phải là các phần tử cháu hay tiếp tục lồng nhau sâu hơn.

### *3.4.7. Adjacent sibling selector:*

Adjacent sibling selector trong CSS là một loại selector được sử dụng để chọn các phần tử kế tiếp có cùng một phần tử anh em ngay sau đó trong cấu trúc HTML. Điều này có nghĩa là chỉ các phần tử ngay kế tiếp với phần tử được chọn sẽ áp dụng các quy tắc CSS.

Adjacent sibling selector được biểu diễn bằng dấu ‘+’ giữa các phần tử.

VD:

h1 + p {

font-weight: bold;

}

### *3.4.8. Attribute selector:*

Attribute selector trong CSS là một loại selector được sử dụng để chọn các phần tử dựa trên các thuộc tính (attributes) của chúng. Điều này cho phép bạn chọn các phần tử mà có thuộc tính cụ thể, hoặc thuộc tính có giá trị cụ thể.

Có 3 loại Attribute selector phổ biến:

- Attribute Exists Selector.

VD:

[target] {

color: blue;

}

- Attribute Equals Selector.

VD:

[type="text"] {

border: 1px solid gray;

}

- Attribute Contains Selector.

VD:

[href\*="example.com"] {

text-decoration: underline;

}

### *3.4.9. Pseudo class selector:*

Pseudo-class selector trong CSS là một loại selector được sử dụng để chọn các phần tử dựa trên trạng thái hoặc vị trí cụ thể của chúng trong tài liệu HTML. Pseudo-classes không phải là phần tử thực sự, nhưng chúng đại diện cho một trạng thái hoặc vị trí của phần tử.

### *3.4.10. Group selector:*

Group selector trong CSS là một cách để áp dụng cùng một tập hợp các quy tắc CSS cho nhiều phần tử cùng một lúc. Điều này giúp viết mã CSS ngắn gọn và dễ đọc hơn bằng cách gom các quy tắc CSS liên quan lại với nhau.

Để sử dụng group selector, bạn chỉ cần liệt kê các phần tử mà bạn muốn áp dụng các quy tắc CSS cho chúng, phân tách bằng dấu phẩy ‘,’.

VD:

h1, h2, h3 {

font-family: Arial, sans-serif;

color: blue;

}

## **3.5. Đơn vị đo lường CSS:**

Trong CSS, có một số đơn vị đo lường được sử dụng để xác định kích thước, khoảng cách, màu sắc và các thuộc tính khác.

Một số đơn vị đo lường tiêu biểu như:

- px (pixels): Đây là đơn vị đo lường tương đối, thường được sử dụng để xác định kích thước của phần tử và khoảng cách trên màn hình. Một pixel tương đương với một điểm ảnh trên màn hình.

- em: Đây là đơn vị đo lường tương đối, thường được sử dụng để xác định kích thước văn bản. Một em bằng kích thước của font hiện tại, nên nó có thể thay đổi tùy thuộc vào font được sử dụng.

- rem: Tương tự như em, nhưng thay vì dựa trên kích thước font của phần tử cha, rem dựa trên kích thước font của phần tử root (thường là phần tử ‘html’ ). Điều này làm cho rem trở nên dễ dàng để quản lý kích thước font trên toàn bộ trang.

- % (percentage): Đây là đơn vị đo lường tương đối, thường được sử dụng để xác định kích thước hoặc khoảng cách dựa trên một phần trăm của phần tử cha.

- vw, vh (viewpoint width, viewpoint height): Đây là đơn vị đo lường tương đối, thường được sử dụng để xác định kích thước hoặc khoảng cách dựa trên chiều rộng hoặc chiều cao của cửa sổ trình duyệt (viewport).

- cm (centimeters), mm (millimeters), in (inches), pt (points), pc (picas): Đây là các đơn vị đo lường tuyệt đối, thường được sử dụng khi cần xác định kích thước cụ thể trong đơn vị vật lý như centimeters, millimeters, inches, points, hoặc picas.

## **3.6. Kế thừa thuộc tính:**

Trong CSS, kế thừa thuộc tính là khả năng cho các phần tử con kế thừa các thuộc tính CSS từ phần tử cha của chúng. Nói cách khác, nếu một phần tử cha có một thuộc tính CSS được áp dụng cho nó, các phần tử con có thể kế thừa thuộc tính đó mà không cần phải khai báo lại trong CSS của chúng.

Tuy nhiên, không tất cả các thuộc tính CSS đều kế thừa. Các thuộc tính như ‘border’, ‘margin’, ‘padding’ và ‘background-color’ thường được kế thừa, trong khi các thuộc tính như ‘width’, ‘height’ và ‘display’ thường không được kế thừa.

## **3.7. Các nhóm thuộc tính trong CSS:**

### *3.7.1. Thuộc tính định dạng font chữ, văn bản (font, text):*

Trong CSS, có một số thuộc tính định dạng font chữ và văn bản được sử dụng để kiểm soát hiển thị của văn bản trên trang web. Một số thuộc tính phổ biến như:

- font-family: Xác định font chữ được sử dụng cho văn bản. Bạn có thể chỉ định một danh sách các font để sử dụng, với các font backup được liệt kê nếu font chính không có sẵn trên máy tính của người dùng.

- font-size: Xác định kích thước của font chữ. Có thể được chỉ định bằng các đơn vị như px, em, rem hoặc%.

- font-weight: Xác định độ đậm của font chữ. Các giá trị phổ biến bao gồm normal, bold, bolder, lighter và các giá trị số từ 100 đến 900.

- font-style: Xác định kiểu font chữ, như italic hoặc normal.

- font-variant: Xác định kiểu chữ in hoa hay chữ thường. Các giá trị phổ biến bao gồm normal và small-caps.

- line-height: Xác định khoảng cách giữa các dòng văn bản. Có thể được chỉ định bằng số, px, em, rem hoặc%.

- text-align: Xác định căn chỉnh văn bản trong một phần tử. Các giá trị phổ biến bao gồm left, right, center và justify.

- text-decoration: Xác định gạch chân, gạch ngang, hoặc gạch trên văn bản. Các giá trị phổ biến bao gồm underline, overline, và line-through.

- text-transform: Xác định kiểu viết chữ của văn bản. Các giá trị phổ biến bao gồm uppercase, lowercase và capitalize.

- text-shadow: Tạo hiệu ứng bóng cho văn bản bằng cách xác định một hoặc nhiều bóng cho văn bản.

### *3.7.2. Thuộc tính định dạng nền (background):*

Trong CSS, có một số thuộc tính được sử dụng để định dạng nền của phần tử HTML. Một số thuộc tính được sử dụng để định dạng nền.

- background-color: Xác định màu nền của phần tử.

- background-image: Xác định hình ảnh nền của phần tử. Bạn có thể chỉ định một đường dẫn đến hình ảnh hoặc một giá trị như none nếu không muốn sử dụng hình ảnh nền.

- background-repeat: Xác định cách lặp lại hình ảnh nền. Các giá trị phổ biến bao gồm repeat (lặp lại cả chiều ngang và dọc), repeat-x (chỉ lặp lại chiều ngang), repeat- y (chỉ lặp lại chiều dọc) và no-repeat (không lặp lại).

- background-position: Xác định vị trí ban đầu của hình ảnh nền. Bạn có thể sử dụng giá trị tương đối như left, center, right, top, bottom hoặc giá trị tuyệt đối như px hoặc%.

- background-size: Xác định kích thước của hình ảnh nền. Các giá trị phổ biến bao gồm contain (hình ảnh được thu nhỏ hoặc phóng to sao cho nó vừa khít bên trong phần tử) và cover (hình ảnh được phóng to hoặc thu nhỏ sao cho nó bao phủ phần tử).

- background-attachment: Xác định liệu hình ảnh nền có cuộn cùng với nội dung của trang web hay không. Các giá trị phổ biến bao gồm scroll (cuộn cùng nội dung) và fixed (không cuộn).

- background-origin: Xác định vị trí bắt đầu của background-image. Các giá trị phổ biến bao gồm border-box, padding-box và content-box.

- background-clip: Xác định phần nào của phần tử sẽ được hiển thị background. Các giá trị phổ biến bao gồm border-box, padding-box và content-box.

### *3.7.3. Mô hình hộp (box model):*

Mô hình hộp (box model) trong CSS là cách mà trình duyệt hiểu và xử lý kích thước và khoảng cách của mỗi phần tử trên trang web. Mỗi phần tử HTML được coi là một hộp, và mô hình hộp mô tả cách mà các thành phần của một hộp được bố trí và tổ chức.

Mô hình hộp cơ gồm các thành phần:

- Content (nội dung): Phần này chứa nội dung thực sự của phần tử, như văn bản, hình ảnh hoặc các phần tử con khác.

- Padding (đệm): Là khoảng cách giữa nội dung của phần tử và biên của nó. Padding tạo ra một khu vực trong phần tử mà không có nội dung, nhưng vẫn nằm trong ranh giới của phần tử.

- Border (biên): Là ranh giới bên ngoài của phần tử. Border được định dạng bằng một màu sắc và một kích thước cụ thể.

- Margin (phần dư): Là khoảng cách giữa biên của phần tử và các phần tử xung quanh nó. Margin tạo ra một khoảng cách giữa các phần tử trên trang web.

### *3.7.4. Các thuộc tính định margin, padding, border:*

a) Margin (phần dư):

Margin-top: Xác định khoảng cách từ phía trên của phần tử đến phần tử kế tiếp.

Margin-right: Xác định khoảng cách từ phía bên phải của phần tử đến phần tử kế tiếp.

Margin-bottom: Xác định khoảng cách từ phía dưới của phần tử đến phần tử kế tiếp.

Margin-left: Xác định khoảng cách từ phía bên trái của phần tử đến phần tử kế tiếp.

b) Padding (phần đệm):

Padding-top: Xác định khoảng cách từ phía trên của phần tử đến nội dung bên trong.

Padding-right: Xác định khoảng cách từ phía bên phải của phần tử đến nội dung bên trong.

Padding-bottom: Xác định khoảng cách từ phía dưới của phần tử đến nội dung bên trong.

Padding-left: Xác định khoảng cách từ phía bên trái của phần tử đến nội dung bên trong.

c) Border (biên):

Border-top: Định dạng phía trên của phần tử.

Border-right: Định dạng viền phía bên phải của phần tử.

Border-bottom: Định dạng viền phía dưới của phần tử.

Border-left: Định dạng viền phía bên trái của phần tử.

### *3.7.5. Thuộc tính định dạng cách hiển thị (display):*

Trong CSS, thuộc tính <display> được sử dụng để xác định cách mà một phần tử HTML sẽ hiển thị trên trang web. Các giá trị của thuộc tính <display> có thể là:

- Block: Phần tử được hiển thị như một khối, chiếm toàn bộ chiều rộng của phần tử cha và bắt đầu từ dòng mới.

- Inline: Phần tử được hiển thị trên cùng một dòng với phần tử khác. Chiều rộng và chiều cao của phần tử được xác định bởi nội dung của nó.

- Inline-block: Phần tử được hiển thị trên cùng một dòng với phần tử khác nhưng vẫn có thể được chỉnh kích thước và margin/padding.

- None: Phần tử không được hiển thị trên trang web và không chiếm không gian trong luồng tài liệu.

- Table: Phần tử được hiển thị như một bảng. Các phần tử con sẽ được hiển thị như các ô trong bảng.

- Flex: Phần tử trở thành một flex container và các phần tử con trở thành các flex item. Các phần tử trong flex container có thể được sắp xếp và điều chỉnh một cách linh hoạt. Đây là một phần của CSS Flexbox.

- Grid: Phần tử trở thành một grid container và các phần tử con trở thành các grid item. Các phần tử trong grid container có thể được sắp xếp thành lưới 2 chiều. Đây là một phần của CSS Grid Layout.

### *3.7.6. Thuộc tính xác định vị trí (position):*

Trong CSS, thuộc tính <position> được sử dụng để xác định cách một phần tử HTML sẽ được định vị trên trang web.

Các giá trị của position có thể là:

- Static: Giá trị mặc định. Phần tử sẽ theo luồng bình thường của tài liệu HTML và không bị ảnh hưởng bởi các thuộc tính ‘top’, ‘bottom’, ‘left’, ‘right’.

- Relative: Phần tử sẽ được định vị dựa trên vị trí ban đầu của nó trong luồng tài liệu, sau đó có thể được điều chỉnh bằng các thuộc tính ‘top’, ‘bottom’, ‘left’, ‘right’.

- Absolute: Phần tử sẽ được đặt tuyệt đối trong mối quan hệ với phần tử mẹ gần nhất có thuộc tính <position> không phải <static>. Nếu không có phần tử mẹ nào có thuộc tính <position> khác <static>, phần tử này sẽ được đặt tuyệt đối trong mối quan hệ với cửa sổ trình duyệt. Phần tử này sẽ không chiếm không gian trong luồng tài liệu và có thể chồng lên các phần tử khác.

- Fixed: Phần tử sẽ được đặt tuyệt đối trong mối quan hệ với cửa sổ trình duyệt, nghĩa là nó sẽ luôn hiển thị ở vị trí cố định trên màn hình ngay cả khi trang được cuộn. Phần tử này cũng không chiếm không gian trong luồng tài liệu.

- Sticky: Phần tử sẽ hoạt động như <relative> cho đến khi nó chạm đến ranh giới của phần tử mẹ, sau đó nó sẽ hoạt động như <fixed> và giữ ở vị trí cố định trên màn hình ngay cả khi trang được cuộn. Đây là một tính năng mới và không được hỗ trợ trên tất cả các trình duyệt.

## **3.8. Float & Clear:**

Trong CSS, <float> và <clear> là hai thuộc tính được sử dụng để xử lý vị trí của các phần tử trong layout của trang web, đặc biệt là trong việc tạo cấu trúc dạng cột và định vị các phần tử theo hàng.

a) Float (Độ nổi):

Thuộc tính <float> được sử dụng để đặt một phần tử ở bên trái hoặc bên phải của phần tử cha của nó, và cho phép các phần tử khác phần nào "bơi" xung quanh nó. Các giá trị thường được sử dụng là ‘left’ hoặc ‘right’.

Khi một phần tử được đặt float, nó sẽ bỏ khỏi luồng bình thường của tài liệu và các phần tử khác sẽ nằm xung quanh phần tử được float đó.

Để chấm dứt float và trở lại luồng bình thường, bạn có thể sử dụng thuộc tính <clear>.

b) Clear (Xoá):

Thuộc tính <clear> được sử dụng để xác định liệu một phần tử cụ thể có thể đặt bên cạnh phần tử được float hay không.

Các giá trị của <clear> bao gồm ‘left’, ‘right’, ‘both’, ‘none’.

Khi bạn muốn chắc chắn rằng một phần tử không bị ảnh hưởng bởi các phần tử float ở trước nó, bạn có thể sử dụng thuộc tính <clear> với giá trị tương ứng.

## **3.9. Flex:**

Trong CSS, flex được sử dụng để tạo ra một container linh hoạt (flex container) để điều chỉnh sắp xếp và căn chỉnh các phần tử con bên trong theo các trục khác nhau. Flexbox là một kỹ thuật mạnh mẽ để tạo ra layout linh hoạt và đáp ứng trên trang web. Một số điểm chính về flex:

- Flex Container: Phần tử được áp dụng trở thành một flex container, và các phần tử con của nó trở thành các flex item.

- Flex Direction: Thuộc tính này xác định hướng của các flex item trong container. Các giái trị phố biến bao gồm ‘row’, ‘row-reverse’, ‘column’, ‘column-reverse’.

- Flex Wrap: Thuộc tính này xác định liệu các flex item có thể xuống dòng (wrap) khi không còn đủ không gian trong container hay không. Các giá trị phổ biến bao gồm ‘nowrap’, ‘wrap’, ‘wrap-reverse’.

- Flex Justify Content: Thuộc tính này xác định cách flex items được căn chỉnh dọc theo trục chính của flex container. Các giá trị phổ biến bao gồm ‘flex-start’, ‘flex-end’, ‘center’, ‘sapce-between’, ‘space-around’.

- Flex Items: Các phần tử con của flex container là các flex item. Chúng sẽ tự động điều chỉnh kích thước của mình để phù hợp với không gian trong container dựa trên các thuộc tính flexbox.

## **3.10. Grid:**

Trong CSS, grid được sử dụng để tạo ra một lưới linh hoạt để điều chỉnh và sắp xếp các phần tử con trong một container. CSS Grid Layout cung cấp một cách tiện lợi để tạo ra các layout phức tạp và đáp ứng trên trang web. Dưới đây là một số điểm chính về thuộc tính <display: grid>.

- Grid Container: Phần tử được áp dụng trở thành một grid container và các phần tử con của nó trở thành các grid item.

- Grid Template Columns và Rows: Hai thuộc tính này được sử dụng để xác định kích thước và số lượng cột và hàng trong grid container. Bạn có thể chỉ định kích thước của mỗi cột/hàng bằng cách sử dụng các giá trị như pixels, phần trăm hoặc tỷ lệ.

- Grid Gap: Thuộc tính này được sử dụng để xác định khoảng cách giữa các cột và hàng trong grid. Các giá trị này có thể là pixels, phần trăm hoặc tỷ lệ.

- Grid Justify và Align: Tương tự như trong flex box, bạn có thể sử dụng các thuộc tính ‘justify-items’, ‘align-items’, ‘justify-content’, ‘align-content’ để căn chỉnh và căn lề các grid items trong grid container.

- Grid Template Areas: Thuộc tính này cho phép bạn định nghĩa các khu vực trong grid container bằng cách sử dụng tên của các khu vực. Điều này giúp tạo ra các layout phức tạp một cách dễ dàng.

- Grid Auto Flow: Thuộc tính này xác định cách các grid items được sắp xếp trong grid container, theo hàng hoặc theo cột.

- Grid Implicit và Explicit: Grid items có thể được định nghĩa một cách rõ ràng (explicit) hoặc tự động sinh ra (implicit) bằng cách sử dụng các thuộc tính ‘grid- column’ và ‘grid-row’.

## **3.11. CSS3:**

a) CSS3 là gì?

CSS3 là phiên bản tiếp theo của ngôn ngữ kiểu trang (CSS - Cascading Style Sheets), một ngôn ngữ sử dụng để định dạng và trang trí nội dung của các trang web. CSS3 đưa ra nhiều cải tiến so với các phiên bản trước đó của CSS, bao gồm cả các tính năng mới và cải thiện cho phép nhà phát triển web tạo ra các trang web có trải nghiệm người dùng tốt hơn và thú vị hơn.

b) CSS3 Selectors:

CSS3 cung cấp một loạt các selectors mới so với CSS2, mở ra khả năng lựa chọn phần tử HTML một cách linh hoạt hơn. Một số CSS3 Selectors phổ biến như:

- Element Selector: Chọn tất cả các phần tử HTML có tên được chỉ định.

- Class Selector: Chọn tất cả các phần tử HTML có class được chỉ định.

- ID Selector: Chọn một phần tử HTML có ID được xác định.

- Universal Selector: Chọn tất cả các phần tử HTML trong tài liệu

c) CSS3 Rounded Corners:

Trong CSS3, bạn có thể tạo các góc bo tròn cho các phần tử bằng cách sử dụng thuộc tính <border-radius>. Đây là một cách dễ dàng và mạnh mẽ để làm cho các góc của phần tử trở nên mềm mại và thân thiện với mắt.

VD:

border-radius: giá\_trị;

Giá trị có thể là một giá trị duy nhất để áp dụng cho tất cả các góc hoặc bốn giá trị riêng biệt để chỉ định góc cụ thể (trái trên, phải trên, phải dưới, trái dưới) theo thứ tự ngược chiều kim đồng hồ, hoặc hai giá trị đầu tiên áp dụng cho góc trái trên và phải dưới, và hai giá trị sau áp dụng cho góc phải trên và trái dưới.

VD:

/\* Góc bo tròn với bán kính 10px \*/

border-radius: 10px;

/\* Góc trái trên: 20px, góc phải dưới: 10px \*/

border-radius: 20px 0 0 10px;

/\* Góc trái trên và phải dưới: 15px, góc phải trên và trái dưới: 5px \*/

border-radius: 15px 5px;

d) CSS3 Border Images:

Trong CSS3, chúng ta có thể sử dụng thuộc tính này để áp dụng hình ảnh làm viền cho các phần tử HTML thay vì việc sử dụng viền thông thường bằng cách chỉ định màu sắc hoặc kiểu viền. Điều này cho phép bạn tạo ra các hiệu ứng viền phong phú và đa dạng hơn.

Cú pháp cơ bản của Border-Image như sau:

border-image: url(hình\_ảnh) giá\_trị;

e) CSS3 Backgrounds:

Trong CSS3, bạn có thể sử dụng nhiều tính năng mới để tạo nền cho các phần tử HTML một cách linh hoạt và phong phú hơn. Một số thuộc tính và tính năng trong CSS3 để điều chỉnh hình nền:

- Background Color: Đặt màu nền cho một phần tử.

- Background Image: Đặt hình ảnh nền cho một phần tử.

- Background Repeat: Xác định cách mà hình ảnh nền được lặp lại

f) CSS Gradients:

CSS Gradients là một tính năng trong CSS3 cho phép bạn tạo ra hiệu ứng gradient mà không cần sử dụng hình ảnh. Gradient có thể là một dãy màu chuyển dần từ một màu này sang màu khác hoặc từ màu này đến trong suốt. Một số cách để có thể sử dụng gradients trong CSS:

- Linear Gradients: Tạo gradients theo hướng tuyến tính, từ điểm này sang điểm khác.

- Radial Gradients: Tạo gradients theo hình tròn từ tâm ra ngoài.

- Diagonal Gradients: Tạo gradients theo đường chéo.

- Transparent Gradients: Sử dụng màu trong suốt trong gradient.

g) CSS3 Shadows:

Trong CSS3, bạn có thể sử dụng các thuộc tính đổ bóng (shadows) để thêm chiều sâu và hiệu ứng đồ họa cho các phần tử HTML. Có hai loại đổ bóng phổ biến: đổ bóng người xem (box-shadow) và đổ bóng văn bản (text-shadow).

h) CSS3 Fonts:

CSS3 cung cấp một số tính năng mới liên quan đến định dạng font chữ trên các trình duyệt web. Một số tính năng quan trọng của CSS3 Fonts có thể kể đến như:

- @font-face Rule: Cho phép bạn nhúng font chữ tùy chỉnh vào trang web của bạn, giúp trình duyệt hiển thị font đó mà không cần font được cài đặt trước đó trên hệ thống người dùng.

- Font Formats: CSS3 cho phép sử dụng các định dạng font chữ mới như WOFF (Web Open Font Format) và WOFF2 để tối ưu hóa việc nhúng font vào trang web.

- Font Stretching: CSS3 cho phép bạn chỉ định tỉ lệ co giãn của font chữ, cho phép tạo ra các hiệu ứng trình chiếu font chữ độc đáo.

- Font Synthesis: Cho phép trình duyệt tự động tạo ra các phông chữ đậm và nghiêng từ các phông chữ không có các biến thể đó.

- Font Variant: Cho phép bạn kiểm soát các biến thể của font chữ như viết hoa, viết thường, viết in hoặc viết nghiêng.

i) CSS3 2D Transforms:

CSS3 Transforms là một tính năng trong CSS3 cho phép bạn thay đổi hình dạng, kích thước và vị trí của các phần tử HTML một cách linh hoạt và mạnh mẽ. Có nhiều loại biến đổi, trong đó 2D Transforms là một trong những loại phổ biến nhất. Một số biến đổi 2D Transform có thể sử dụng:

- Translate: Di chuyển phần tử theo trục X và Y.

- Rotate: Xoay phần tử một góc nhất định.

- Scale: Phóng to và thu nhỏ phần tử.

- Skew: Nghiêng phần tử theo trục X và Y.

- Matrix: Chuyển đổi tổ hợp của các phép biến đổi.

k) CSS3 3D Transforms:

CSS3 Transforms cũng cung cấp các biến đổi 3D, cho phép bạn tạo ra các hiệu ứng đồ họa 3 chiều trên các phần tử HTML. Một số biến đổi bạn có thể sử dụng:

- RotateX, RotateY, RotateZ: Xoay phần tử quanh các trục 3 chiều.

- Rotate3d: Xoay phần tử theo 1 vector 3 chiều.

- Translate3d: Di chuyển phần tử theo 3 chiều.

l) CSS3 Transitions:

CSS3 Transitions là một tính năng cho phép bạn tạo ra các hiệu ứng chuyển động mềm mại và tương tác trên các thuộc tính CSS của các phần tử. Với CSS3 Transitions, bạn có thể làm cho các thuộc tính của phần tử thay đổi một cách mượt mà khi có sự kiện kích hoạt, chẳng hạn như hover, click, focus, và nhiều hơn nữa.

Để sử dụng CSS3 Transitions, bạn cần chỉ định thuộc tính <transition> trong CSS với cú pháp:

selector {

transition: property duration timing-function delay;

}

m) CSS3 Multiple Columns:

CSS3 Multiple Columns là một tính năng trong CSS3 cho phép bạn chia nội dung của một phần tử thành nhiều cột. Điều này hữu ích khi bạn muốn hiển thị nội dung dài trên một trang web mà không phải cuộn xuống quá nhiều.

Để sử dụng Multiple Columns, bạn có thể sử dụng thuộc tính <column-count> để xác định số lượng cột và <column-gap> để xác định khoảng cách giữa các cột. Cú pháp cơ bản là:

.selector {

column-count: value; /\* Số lượng cột \*/

column-gap: value; /\* Khoảng cách giữa các cột \*/

}

n) CSS3 Animations:

CSS3 Animations là một tính năng trong CSS3 cho phép bạn tạo ra các hiệu ứng động trên các phần tử HTML một cách linh hoạt và mạnh mẽ. Bằng cách sử dụng CSS3 Animations, bạn có thể tạo ra các hiệu ứng như di chuyển, xoay, phóng to, thu nhỏ và nhiều hơn nữa mà không cần sử dụng JavaScript.

Để tạo CSS3 Animations, bạn sử dụng các keyframes để xác định các bước trong quá trình animation. Dưới đây là cú pháp cơ bản:

@keyframes animationName {

0% {

/\* CSS properties at the start of the animation \*/

}

50% {

/\* CSS properties halfway through the animation \*/

}

100% {

/\* CSS properties at the end of the animation \*/

}

}

## **3.12. SCSS:**

SCSS (Sassy CSS) là một ngôn ngữ mở rộng của CSS, cung cấp các tính năng mạnh mẽ để viết mã CSS một cách dễ dàng và hiệu quả hơn. SCSS giữ nguyên cú pháp của CSS, nhưng bổ sung thêm các tính năng như biến, nesting, mixins, functions, và các module khác giúp tăng khả năng tái sử dụng và duy trì mã CSS.

Một số tính năng chính của SCSS:

- Biến: Cho phép bạn đặt tên cho các giá trị và sử dụng chúng trong toàn bộ stylesheet.

- Nesting: Cho phép nhóm các quy tắc CSS bên trong các phần tử cha, giúp tăng khả năng đọc và duy trì mã CSS.

- Mixins: Cho phép bạn tái sử dụng các đoạn mã CSS dễ dàng và hiệu quả hơn.

- Functions: Cho phép bạn tạo ra các hàm để tái sử dụng logic CSS.

## **3.13. SASS:**

SASS (Syntactically Awesome Style Sheets) là một ngôn ngữ mở rộng của CSS, được sử dụng để viết mã CSS một cách dễ dàng và hiệu quả hơn. SASS giữ cú pháp tương tự CSS nhưng bổ sung thêm các tính năng như biến, nesting, mixins, inheritance, và các module khác giúp tăng khả năng tái sử dụng và duy trì mã CSS.

Một số tính năng chính của SASS:

- Biến: Cho phép bạn đặt tên cho các giá trị và sử dụng chúng trong toàn bộ stylesheet.

- Nesting: Cho phép nhóm các quy tắc CSS bên trong các phần tử cha, giúp tăng khả năng đọc và duy trì mã CSS.

- Mixins: Cho phép bạn tái sử dụng các đoạn mã CSS dễ dàng và hiệu quả hơn.

- Functions: Cho phép bạn tạo ra các hàm để tái sử dụng logic CSS.

# **Kết luận chương 3:**

CSS và CSS3 là hai ngôn ngữ được sử dụng để tạo kiểu và trang trí cho các trang web.

CSS và CSS3 đóng vai trò quan trọng trong việc thiết kế và phát triển các trang web hiện đại. Sự tiến bộ từ CSS đến CSS3 mang lại nhiều cải tiến và tính năng mới, giúp tạo ra trải nghiệm người dùng tốt hơn và trang web đẹp mắt hơn.

# **CHƯƠNG 4: THIẾT KẾ WEBSITE CÁC TRÒ CHƠI GAMES**

## **4.1. Ý tưởng của Website:**

- Trang chủ đa dạng trò chơi: Tạo một trang chủ hiển thị các trò chơi phổ biến, mới nhất và được yêu thích. Người chơi có thể dễ dàng chọn lựa và trải nghiệm ngay từ trang chủ.  
  
- Hệ thống phân loại trò chơi: Xây dựng hệ thống phân loại trò chơi theo thể loại, độ khó, năm phát hành, hoặc theo đánh giá của người chơi để giúp họ tìm kiếm trò chơi dễ dàng hơn.

- Bảng xếp hạng: Hiển thị bảng xếp hạng của người chơi, các nhóm chơi, hoặc các giải đấu để tạo động lực cho người chơi cố gắng đạt thành tích cao hơn.  
  
- Tính năng xã hội: Tích hợp chia sẻ trên mạng xã hội, kết nối với tài khoản Facebook hoặc Twitter để người chơi có thể chia sẻ thành tích, mời bạn bè tham gia chơi cùng.

## **4.2. Xây dựng bố cục của trang Web:**

|  |
| --- |
| Header |
| Menu |
| Main |
| Footer |

4.2.1. Header (Đầu trang):  
 - Logo: Hiển thị logo của trang web hoặc của trò chơi.  
 - Menu điều hướng: Bao gồm các liên kết đến các trang chính, thể loại trò chơi, cộng đồng, sự kiện, v.v.  
 - Tìm kiếm: Cung cấp ô tìm kiếm để người dùng có thể dễ dàng tìm kiếm trò chơi yêu thích.  
  
4.2.2. Slider hoặc Banner:  
 - Hiển thị các trò chơi nổi bật, sự kiện hoặc thông báo quan trọng để thu hút sự chú ý của người dùng.  
  
4.2.3. Danh sách trò chơi:  
 - Hiển thị các trò chơi phổ biến, mới nhất, được yêu thích hoặc theo thể loại để người dùng dễ dàng lựa chọn.  
  
4.2.4. Phân loại trò chơi:  
 - Tạo các mục phân loại trò chơi theo thể loại, độ khó, năm phát hành để người dùng có thể lọc và chọn trò chơi theo sở thích của mình.  
  
  
4.2.5. Thông tin chi tiết trò chơi:  
 - Trang chi tiết của mỗi trò chơi nên bao gồm hình ảnh, mô tả, đánh giá, lượt chơi, nút chơi trực tiếp, v.v.  
  
4.2.6. Bảng xếp hạng và giải đấu:  
 - Hiển thị bảng xếp hạng người chơi, các giải đấu đang diễn ra, thông tin về giải thưởng để kích thích sự tham gia của người chơi.(Đang cập nhập…)  
  
4.2.7. Footer (Chân trang):  
 - Liên kết đến các trang quan trọng như liên hệ, thông tin về trang web, chính sách bảo mật, điều khoản sử dụng, v.v.  
 - Thông tin liên hệ và các biểu tượng mạng xã hội để người dùng có thể kết nối và liên lạc.

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

HTML:

<!DOCTYPE> để khai báo kiểu HTML đang sử dụng

<head>: chứa các thông tin meta, tiêu đề của trang web và liên kết đến tập tin CSS.

<body>: chứa toàn bộ nội dung của trang web

<title>: chứa tiêu đề của trang web, sau đó hiển thị trên thanh tiêu đề của trình duyệt.

<img>: Hiển thị hình ảnh liên quan đến chùa Hương và hiện lên trên trang web.

<h1>: Để tạo tiêu đề cho trang web.

<div>: Được sử dụng để nhóm các mục trong thanh menu đa cấp.

<ul>: Được sử dụng để đại diện cho danh sách các mục trong thanh menu đa cấp.

<li>: Được sử dụng để đại diện cho một mục của thanh menu đa cấp.

<br>: Thẻ <br> được sử dụng trong trang web để xuống dòng cho các đoạn trong trang web.

<b>: Được sử dụng trong web để đánh dấu những mục quan trọng trong web

<p>: Được sử dụng để ghi nội dung, những lời văn trong trang web

<center>: Để căn giữa một số hình ảnh, những chữ in nghiêng trong trang web.

<i>: Để in nghiêng chữ, trong trang web thì được dùng để ghi chú dưới nhũng bức ảnh trong trang web.

<footer>: Để chứa thông tin ở cuối trang web, thông tin về người làm web và cảnh cáo bản quyền của trang web.

CSS:

Trong CSS, em sử dụng những thẻ như:

Color: Được sử dụng để đổi màu sắc cho chữ trên menu đa cấp.

Box-sizing: Được sử dụng để tính kích thước của phần tử và xem thuộc tính “width” và “height” áp dụng cho kích thước nội dung của phần tử trong trang web.

Margin: Để đạt giá trị cho độ dày của vùng lề.

Padding: Để thiết lập khoảng cách giữa nội dung của phần tử và biên của phần tử đó.

Float: Được sử dụng để xác định vị trí của phần từ trên trang web.

List-style: Được sử dụng để kiểm soát kiểu đánh dấu cho các dnah sách trong HTML.

Display: Xác định cách để phần tử hiển thị trên trang web.

Position: Được sử dụng để xác định cách một phần tử HTML sẽ được định vị trên trang web.

Text-decoration: Được sử dụng để xác định kiểu trang trí cho văn bản.

Line-height: Được sử dụng để xác định độ cao của dòng trong một phần tử văn bản trong trang web.

Background-color: Được sử dụng để thiết lập màu nền cho phần tử HTML.

Width: Được sử dụng để xác định chiều rộng của một phần tử HTML.

Height: Được sử dụng để xác định chiều cao của một phần tử HTML.

Text-align: Được sử dụng để căn chỉnh văn bản trong một phần tử.

Text-Transform: Được sử dụng để chuyển đổi văn bản trong một phần tử thành kiểu chữ viết khác.

Border: Được sử dụng để thiết lập các đường viền xung quanh một phần tử trong trang web.

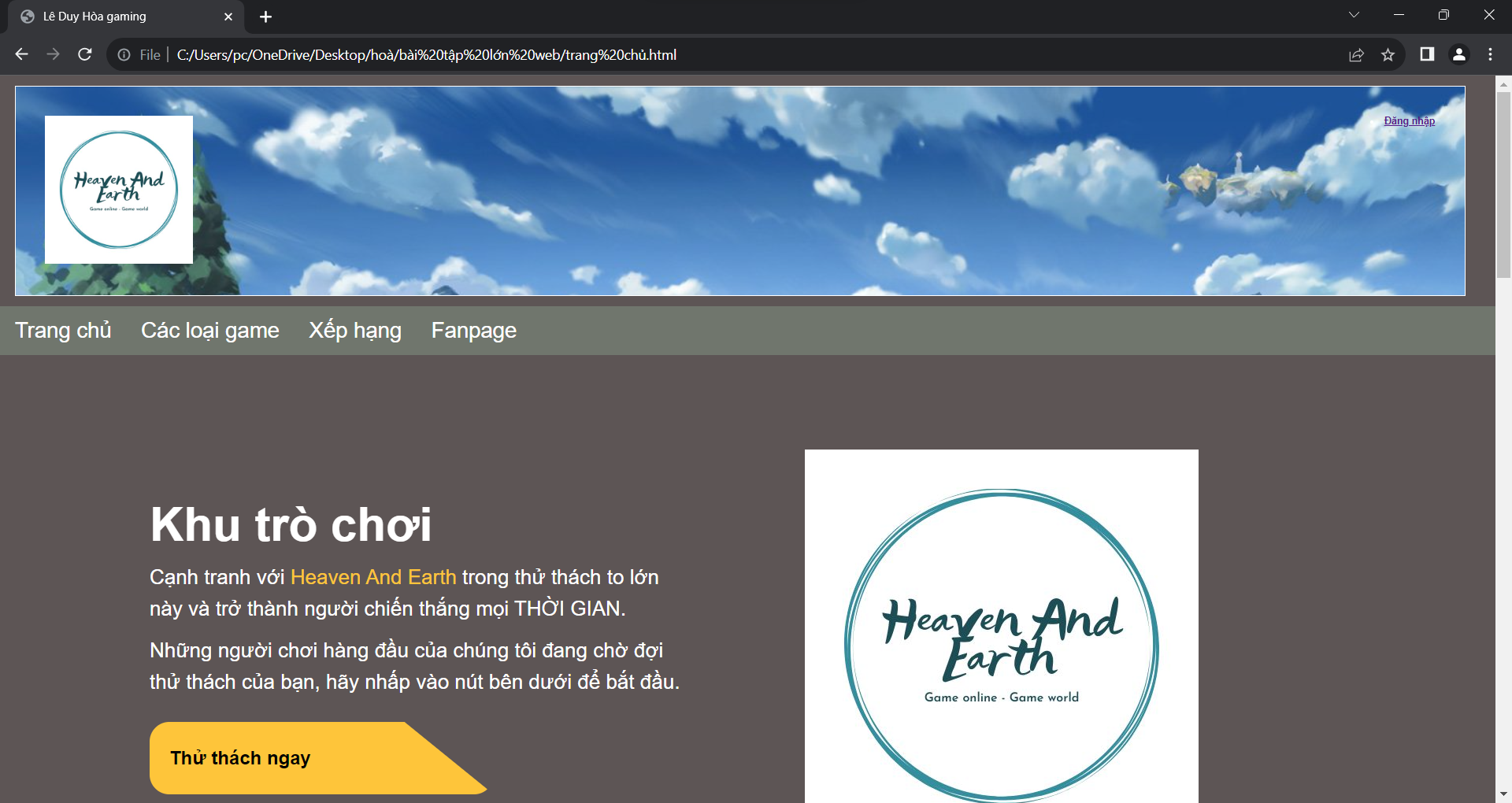
Max-width: Được sử dụng để xác định chiều rộng tối đa của một phần tử.

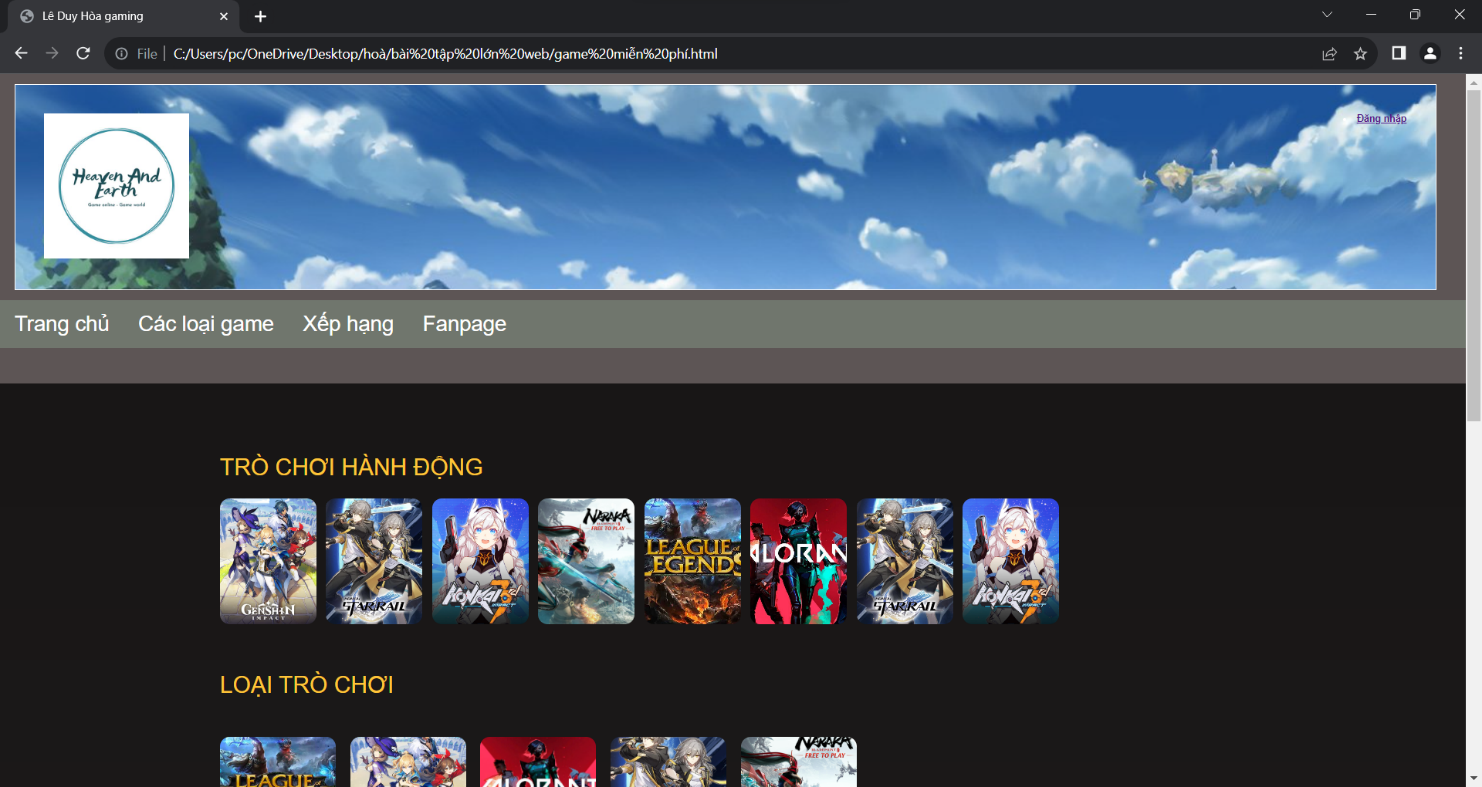
Justify-content: Được sử dụng để xác định cách mà các phần tử con của phần tử container sẽ được căn chỉnh theo chiều ngang.

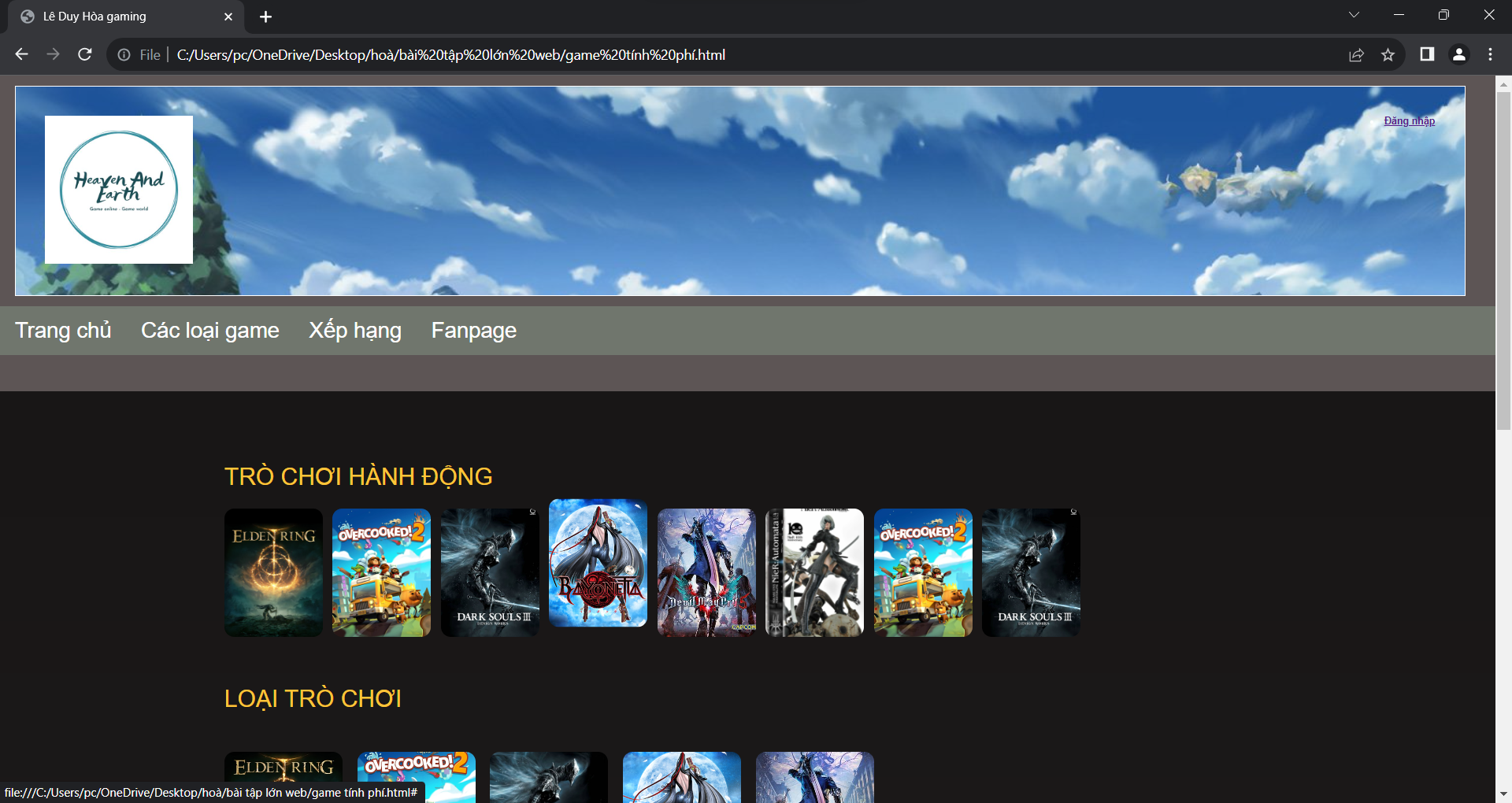
Align-items: Được sử dụng để xác định cách mà phần tử con của phần tử container sẽ được căn chỉnh theo chiều dọc.

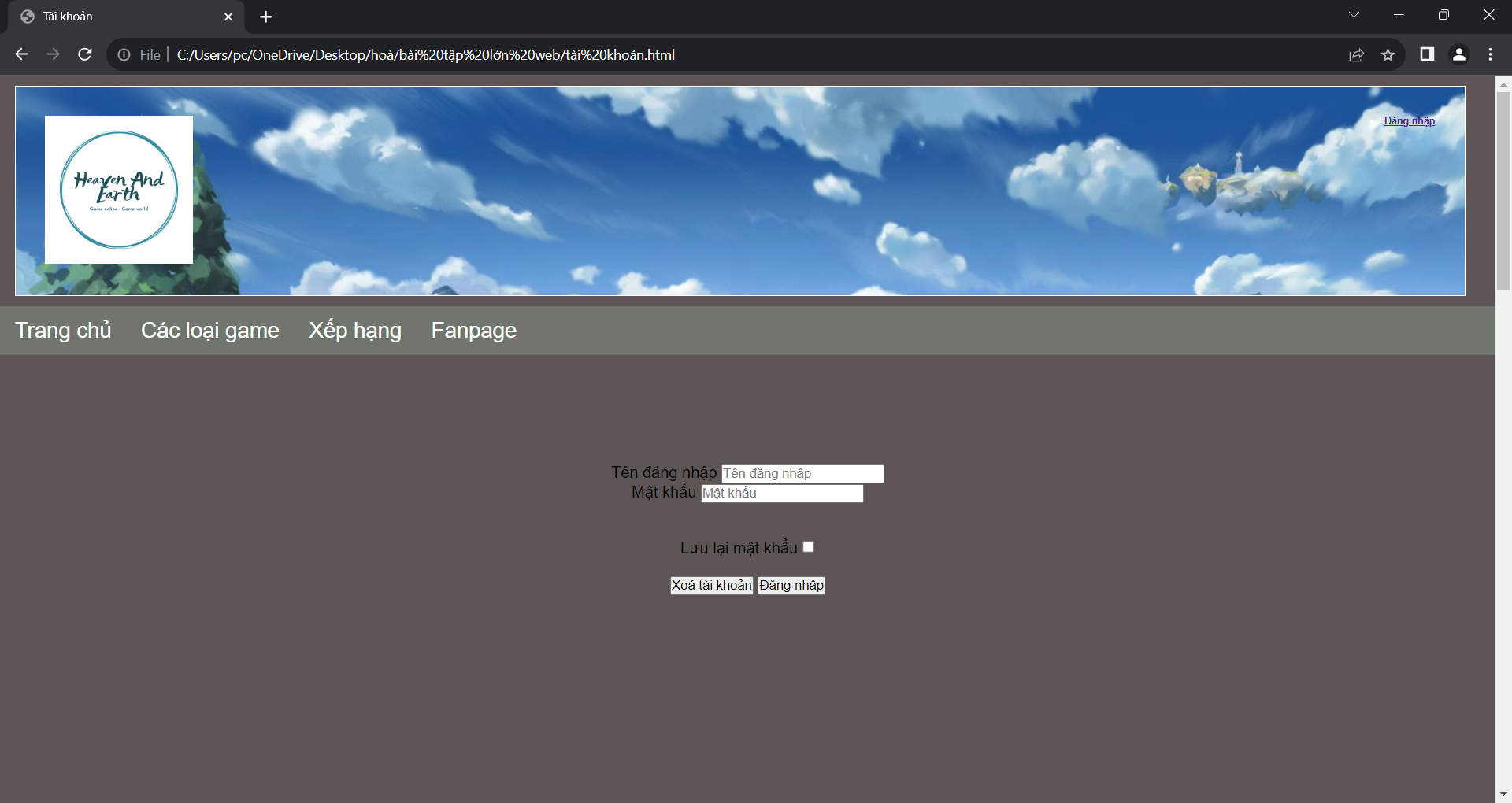
## **4.4. Kết quả đạt được:**

Kết quả đạt được là:

Trang chủ của web

Các loại game free

Các tựa game tính phí

Trang đăng nhập tài khoản

## **Kết luận chương 4:**

Như vậy trang web bên trên đã cung cấp những con game nổi tiếng ở Việt Nam.

Giao diện của trang web nên được thiết kế đơn giản, dễ sử dụng và thân thiện với người dùng.

**KẾT LUẬN**

Thiết kế website chuyên nghiệp và thú vị giúp cải thiện trải nghiệm người dùng, tạo sự hấp dẫn và thu hút người chơi. Website trò chơi games thường có tính tương tác cao, cho phép người chơi tham gia, giao tiếp và chơi cùng nhau trực tuyến. Thiết kế website có thể chứa nhiều trò chơi khác nhau, từ casual games đến games nhiều người chơi, đáp ứng nhu cầu của đa dạng đối tượng người chơi. Website trò chơi games có thể dễ dàng cập nhật và thêm mới các trò chơi, sự kiện để duy trì sự hấp dẫn và giữ chân người chơi. Thiết kế website đáp ứng tốt trên nhiều thiết bị và trình duyệt khác nhau, tạo điều kiện thuận lợi cho người chơi truy cập.

Thiết kế website trò chơi games đòi hỏi kiến thức kỹ thuật cao về lập trình, thiết kế đồ họa và kỹ năng xử lý dữ liệu. Xây dựng và duy trì một website trò chơi games chất lượng có thể đòi hỏi chi phí đầu tư lớn, từ việc mua sắm thiết bị đến chi phí phát triển và quảng cáo. Ngành công nghiệp trò chơi games rất cạnh tranh, vì vậy thiết kế website cần phải đáp ứng được yêu cầu cao về chất lượng và sáng tạo để cạnh tranh hiệu quả. Sự phức tạp về vấn đề bản quyền trong ngành game có thể ảnh hưởng đến việc thiết kế website và chọn lựa các trò chơi để tích hợp. Website trò chơi games cần có cơ chế bảo mật thông tin người chơi, tránh rủi ro về lạm dụng dữ liệu cá nhân hay tài khoản người dùng.

Hướng phát triển quan trọng là thiết kế website trò chơi games linh hoạt, tương thích trên nhiều thiết bị và nền tảng khác nhau như PC, điện thoại di động, máy tính bảng để tạo trải nghiệm người chơi mượt mà và thuận tiện. Xu hướng tăng cường tính tương tác giữa người chơi thông qua các tính năng như chế độ chơi đa người, giao tiếp trực tuyến, thách đấu và chia sẻ kết quả trên mạng xã hội. Sự phát triển về đồ họa và hiệu ứng đặc biệt giúp tạo ra trải nghiệm hấp dẫn và sống động hơn cho người chơi. Với sự phổ biến của thiết bị di động, việc phát triển trò chơi trên nền tảng di động và thiết kế website tương thích với thiết bị di động là xu hướng quan trọng. Sử dụng công nghệ mới như trí tuệ nhân tạo (AI), thực tế ảo (VR), thực tế tăng cường (AR) để tạo ra trải nghiệm chơi game độc đáo và hấp dẫn. Tăng cường tính năng xã hội, cho phép người chơi kết nối, tham gia cộng đồng, chia sẻ trải nghiệm và thông tin với nhau. Tăng cường bảo mật thông tin người dùng, quản lý dữ liệu cá nhân và tài khoản người chơi một cách an toàn và bảo mật.

Những hướng phát triển trên đều nhằm mục tiêu cải thiện trải nghiệm người chơi, tạo ra những trò chơi games đa dạng, hấp dẫn và tiện lợi trên mọi nền tảng.

# **DANH MỤC TÀI LIỆU THAM KHẢO**

1. "Don't Make Me Think" by Steve Krug - Cuốn sách này tập trung vào trải nghiệm người dùng và cung cấp hướng dẫn cụ thể về cách thiết kế website dễ hiểu và dễ sử dụng
2. "The Principles of Beautiful Web Design" by Jason Beaird - Cuốn sách này tập trung vào việc thiết kế web đẹp mắt và chuyên nghiệp, cung cấp hướng dẫn về cách sử dụng màu sắc, hình ảnh và layout.
3. "Web Design with HTML, CSS, JavaScript and jQuery Set" by Jon Duckett - Bộ sách này cung cấp kiến thức cơ bản về HTML, CSS, JavaScript và jQuery, giúp bạn hiểu rõ cách xây dựng website từ đầu.