**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 TRANG WEB BÁN HÀNG THỜI TRANG**

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

**Sinh viên thực hiện:**  **Nguyễn Mạnh Hùng**

**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 TRANG WEB BÁN HÀNG THỜI TRANG**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Mã Sinh Viên** | **Họ và Tên** | **Ngày Sinh** | **Điểm** | |
| **Bằng Số** | **Bằng Chữ** |
|  | **1771020316** | **Nguyễn Mạnh Hùng** | **14/09/2005** |  |  |

**CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2**

**Hà Nội, 2024**

**LỜI NÓI ĐẦU**

MỤC LỤC

**CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM** 8

**1.1.** **Các khái niệm cơ bản** 8

*Thiết kế web là gì?* 8

*Vai trò của một nhà thiết kế web* 8

**1.2.** **Phân biệt Web tĩnh, Web động** 8

**Web tĩnh là gì?** 8

Web động là gì? 10

**1.3.** **Một số thuật ngữ** 10

**1.4.** **Một số công cụ dùng thiết kế Web** 11

**CHƯƠNG II: HTML & HTML5** 13

**2.1. Tổng quan về HTML** 13

HTML là gì? 13

Ưu và nhược điểm của HTML 14

HTML hoạt động như thế nào? 14

**2.2. Cấu trúc tổng quát trang HTML** 15

**2.3. Các thẻ HTML thông dụng** 16

- Các thẻ xử lý văn bản 16

- Các thẻ tạo bảng 17

- Thẻ liên kết 18

- Các thẻ đa phương tiện 18

- Thẻ tạo khung 19

**2.4. Các thẻ tạo biểu mẫu (form)** 20

**2.5. Một số thẻ HTML đặc biệt** 20

**2.6. HTML5** 21

- **HTML5 là gì?** 21

- **Cú pháp HTML5** 21

- **Khai báo DOCTYPE trong HTML5** 21

- **Thẻ ngữ nghĩa** 21

- **Webform** 22

**CHƯƠNG III: CSS và CSS3** 24

**3.1. CSS là gì?** 24

**3.2. Cú pháp CSS** 24

**3.3. Áp dụng CSS vào trang HTML** 25

**3.4. Selectors** 25

3.4.1. Universal selector 25

3.4.2. Type selector 25

3.4.3. Identity selector 25

3.4.4. Class selector 25

3.4.5. Descendant selector 25

3.4.6. Child selector 25

3.4.7. Adjacent sibling selector 26

3.4.8. Attribute selector 26

3.4.9. Pseudo class selector 26

3.4.10. Group selector 26

**3.5. Đơn vị đo lường CSS** 26

**3.6. Kế thừa thuộc tính** 27

**3.7. Các nhóm thuộc tính trong CSS** 27

**3.8. Float & Clear** 28

**3.9. Flex** 28

**3.10. Grid** 29

**3.11. CSS3** 30

**3.12. SCSS** 31

**3.13. SASS** 32

**CHƯƠNG 4: THIẾT KẾ WEBISTE THEO ĐỀ TÀI BẠN CHỌN** 33

**4.1. Ý tưởng của Website:** 33

**4.2. Xây dựng bố cục của trang Web** 33

**4.3. Thiết kế trang Web bằng HTML và CSS** 35

**4.4. Kết quả đạt được** 36

**KẾT LUẬN** 37

**DANH MỤC TÀI LIỆU THAM KHẢO** 39

**BẢNG CÁC TỪ VIẾT TẮT**

**(Nếu có)**

|  |  |  |
| --- | --- | --- |
| **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. **Các khái niệm cơ bản**

### Thiết kế web là gì?

     Thiết kế web là tất cả các yếu tố trực quan như màu sắc, kiểu chữ, hình ảnh và các nút điều hướng liên quan đến một trang web. Nhưng đó không đơn giản chỉ là ngoại hình thiết kế đẹp. Thiết kế web giải quyết cách bố cục một trang web, đảm bảo trang web đều hiển thị đẹp ở các kích thước màn hình khác nhau, giữ cho việc điều hướng web trở nên đơn giản và dễ dàng, đồng thời đảm bảo khả năng sử dụng của trang web trôi chảy.

### Vai trò của một nhà thiết kế web

     Nếu coi nhà phát triển web là kỹ sư xây dựng một ngôi nhà thì nhà thiết kế web là kiến trúc sư hoặc người thiết kế nội thất. Nhà thiết kế web được coi là nghệ sĩ sáng tạo cho trang web hoặc ứng dụng, họ có trách nhiệm tập hợp các nội dung sáng tạo mang tính thương hiệu, thẩm mỹ và cung cấp giá trị cho người dùng.

     Thiết kế web và nhà thiết kế web từng là một thuật ngữ chung cho những người thiết kế trang web. Nhưng trong những năm gần đây, nhiều vai trò chuyên biệt hơn trong thiết kế web đã xuất hiện. Những vai trò này là người thiết kế UX (trải nghiệm người dùng) và UI (giao diện người dùng).



Minh họa bởi OrangeCrush

* 1. **Phân biệt Web tĩnh, Web động**

**Web tĩnh là gì?**

Web tĩnh là những website không có hệ thống quản lý nội dung và người dùng không thể chỉnh sửa được hoặc có thể chỉnh sửa, thay đổi dữ liệu được. Hiện nay web tĩnh hầu như ít tồn tại hoặc chỉ tồn tại với những công ty chuyên về thiết kế website, họ sử dụng web tĩnh bởi họ có kiến thức và dễ dàng chỉnh sửa nội dung khi cần.

**Web tĩnh được làm từ gì?**

Về kiến thức cơ bản thì web tĩnh thường được xây dựng từ CSS, HTML, JAVASCRIPT (DHTML), hiện nay có thêm công nghệ HTML5 & CSS3. Có thể bạn nhìn một web nào đó nhìn rất lung linh, đẹp và nội dung có thể thay đổi nhưng chưa chắc nó là web động bởi vì người ta có thể sử dụng DHTML để thay đổi nội dung (tại client).

Nếu bạn chạy trên máy tính của bạn thì với web tĩnh bạn có thể đặt tại đâu cũng chạy được vì bản chất nó chỉ là một file bình thường.

**Ưu điểm của web tĩnh:**

* Về giao diện Designer có thẻ thiết kế theo kiểu mới lạ
* Tốc độ truy cập nhanh bởi nó chỉ là những file HTML
* Chi phí đầu tư thấp bởi bạn không phải trả tiền nhiều cho Coder
* Thân thiện với bộ máy tìm kiếm bởi bạn có thể đặt tên file tùy ý (ten-file.html, tieu-de-tin-tuc.html)

**Nhược điểm của web tĩnh:**

* Khó quản lý nội dung
* Khó nâng cấp bảo trì
* Mỗi khi thay đổi phải vào file HTML, CSS hoặc JAVASCRIPT để chỉnh sửa

**Khi nào nên sử dụng web tĩnh?**

* Nếu bạn là doanh nghiệp muốn tự mình làm website thì bạn có thể học các kiến thức căn bản và tự làm một Web tĩnh cho mình
* Nội dung website ít khi cập nhật và ít nên bạn muốn tiết kiệm chi phí
* Website bạn nhỏ và bạn thuê luôn người chuyên về web để quản trị

Trên thực tế thì mình thấy hiện nay chỉ có các công ty thiết kế web họ mới làm web tĩnh bởi vì họ có khả năng chỉnh sửa được chứ khách hàng thì họ không có khả năng đó. Hiện nay có các trang web ONE PAGE có các hiệu ứng slide khá độc nên cũng có thể họ sử dụng web tĩnh hoặc tĩnh một phần trong đó.

### Web động là gì?

Ngược lại với web tĩnh, web động là những website có hệ thống quản lý nội dung và người dùng có thể chỉnh sửa nội dung được. Ví dụ như trang vnexpress.net hay 24h.com.vn thì đó là những web động bởi vì họ có thể đăng tin mới, chỉnh sửa tin mới và chỉnh sửa danh mục menu, ...

**Web động được làm từ gì?**

Web động sử dụng các công nghệ như HTML,  CSS, HTML5, CSS3, JAVASCRIPT, ... và điều đặc biệt là có sử dụng một ngôn ngữ lập trình server như PHP, một hệ quản trị cơ sở dữ liệu như MySQL, vì vậy web động phải chạy trong máy chủ. Nếu đặt trên máy tính của bạn thì bạn phải cài các Web server ảo như vertrigo, Xampp, wampp, ...

**Ưu điểm của web động:**

* Dễ dàng quản lý nội dung
* Dễ dàng nâng cấp và bảo trì
* Có thể xây dụng được web lớn
* Thường sử dụng tương tác với người dùng cao
* Hiện nay vẫn SEO tốt bởi ta có thể sử dụng chức năng Rewrite URL để chuyên URL thân thiện

**Nhược điểm của web động:**

* Chi phí xây dựng cao
* Nếu web lớn có thể cần thêm nhân sự chuyên ngành

**Khi nào nên sử dụng web động?**

* Bạn làm website thương mại điện tử bán hàng
* Bạn làm web giới thiệu sản phảm công ty
* Bạn làm web tin tức, blog cá nhân
* Web bạn tầm cỡ lớn
  1. **Một số thuật ngữ**

Dưới đây là một số thuật ngữ bạn phổ biến hiện nay:

1. **Hosting**: Hosting là dịch vụ lưu trữ trang web trên máy chủ kết nối với Internet. Khi một người dùng truy cập trang web của bạn, máy chủ web sẽ gửi các tệp tin liên quan đến trang web đó cho trình duyệt web của người dùng.
2. **Tên miền (Domain Name)**: Tên miền là địa chỉ của một trang web hoạt động trên Internet. Tên miền được sử dụng thay cho địa chỉ IP dài và khó nhớ, giúp người dùng tìm kiếm và truy cập trang web một cách dễ dàng hơn.
3. **Web Server Name**: Web Server Name, còn được biết đến với tên gọi khác là Domain NameServer, hoặc DNS Server. Chúng có vai trò giúp kết nối URL với địa chỉ IP máy chủ một cách thân thiện nhất và phù hợp với người dùng.
4. **HomePage**: HomePage, còn được gọi là trang chủ, là trang web mà người dùng thấy đầu tiên khi họ truy cập vào một website.
5. **Web Site**: Một trang web (Web Site) là một tập hợp các trang web và nội dung liên quan được xác định bởi một tên miền chung và được xuất bản trên ít nhất một máy chủ web.
6. **WebServer**: Một máy chủ web là một máy tính chứa phần mềm máy chủ web và các tệp tin thành phần của một trang web (ví dụ: tài liệu HTML, hình ảnh, bảng kiểu CSS và tệp JavaScript).
7. **URL-Uniform Resource Location**: URL (Uniform Resource Locator), còn được gọi là địa chỉ web, là một loại URI (Uniform Resource Identifier) chỉ định vị trí của một tài nguyên trên mạng và cơ chế để lấy lại nó.
8. **Browser**: Trình duyệt web là một ứng dụng cho phép người dùng truy cập thông tin trên World Wide Web. Một số trình duyệt web phổ biến bao gồm Google Chrome, Mozilla Firefox, Safari, và Microsoft Edge.
   1. **Một số công cụ dùng thiết kế Web**
9. **Mobirise Website Builder**: Công cụ này miễn phí cho tất cả người dùng, cả mục đích cá nhân và thương mại. Nó có hơn 800 website blocks và template theo xu hướng của thế giới[1](https://topdev.vn/blog/top-8-cong-cu-thiet-ke-website-va-page/).
10. **Visual Composer Website Builder**: Được thiết kế bởi WPBakery, Visual Composer Builder không giống công cụ WPBakery Builder. Trình thiết kế này không yêu cầu kĩ năng code và tương thích với mọi theme của WordPress.
11. **Portfoliobox**: Công cụ này cho phép bạn thiết kế theo bất kỳ phong cách nào với mọi trang web.
12. **Elementor**: Một công cụ thiết kế page chỉ cần kéo và thả. Word Press plugin này cho phép bạn thiết kế và xây dựng mọi thứ, mà lại hoàn toàn miễn phí.
13. **Notepad++**: Đây là một phần mềm soạn thảo rất mạnh, độ tùy biến cao và hoàn toàn miễn phí.
14. **Intype**: Đây là một phần mềm soạn thảo rất nhẹ và miễn phí nhưng cũng cung cấp cho bạn đầy đủ tính năng để sử dụng cho việc lập trình website.
15. **PHPdesigner**: Công cụ này có giao diện rất thân thiện và nhiều tính năng hay như: xem trước trang web trên trình duyệt, code snippets, báo lỗi cú pháp khi lập trình với PHP, autocomplete.
16. **Dreamweaver**: Đây là phần mềm được dùng rộng rãi cho người mới bắt đầu lập trình hay các lập trình viên chuyên nghiệp.
17. **E-Texteditor**: Với phần mềm e-Texteditor chuyên nghiệp, chúng ta có thể viết html một cách dễ dàng nhờ tình năng view trực tiếp trên màn hình.
18. **Komodo Edit**: Komodo Edit là một trong các trình soạn thảo XML miễn phí tốt nhất hiện nay.
19. **WordPress**: WordPress là công cụ thiết kế website phổ biến, được dùng nhiều hiện nay.

Nắm được bản chất của thiết kế web so với phát triển web có thể giúp bạn đánh giá tốt hơn những gì bạn cần, ai có thể đáp ứng nhu cầu đó và khi nào bạn cần chúng. Thêm vào đó là kiến ​​thức về những điều cần tìm khi tuyển dụng mang lại cho bạn lợi thế trong việc thành lập một nhóm để triển khai trang web hoạt động hiệu quả.

Mục đích cuối cùng? Một trang web hoạt động mượt mà, trông đẹp mắt và chuyển đổi người truy cập thành người mua hàng.

Cho dù đó là để xây dựng thương hiệu cá nhân hay doanh nghiệp nhỏ, hãy tham gia cùng nhiều người đã thành công trong việc mở rộng phạm vi thị trường và tăng kênh doanh thu của họ với các trang web có chức năng và hình ảnh đẹp mắt.

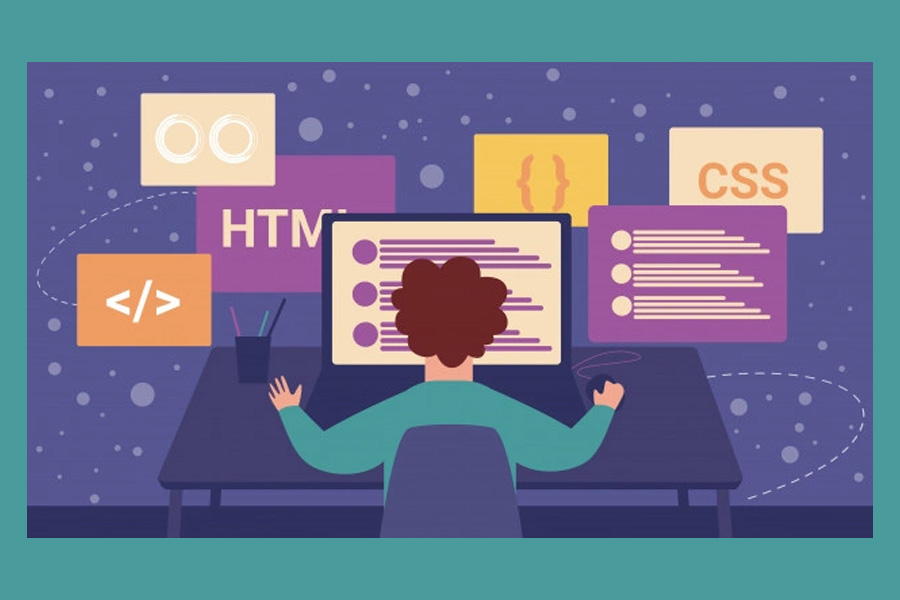
# **CHƯƠNG II: HTML & HTML5**

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

## **2.1. Tổng quan về HTML**

### HTML là gì?

**HTML (Hypertext Markup Language)** là một dạng ngôn ngữ đánh dấu văn bản giúp người dùng tạo và cấu trúc nên các thành phần trong ứng dụng hay website. Và phân chia các đoạn văn, links, heading, blockquote...

[](https://bizflyportal.mediacdn.vn/bizflyportal/images/htm16157917934978.jpg)

*HTML là gì?*

HTML không phải là một ngôn ngữ lập trình, vì vậy, nó không thể tạo ra các chức năng “động" được. Mục đích sử dụng của nó tương tự như Microsoft Word, chỉ dùng để xây dựng bố cục và định dạng văn bản.

Lịch sử phát triển HTML

**HTML** được tạo ra bởi Tim Berners-Lee, một nhà vật lý học của trung tâm nghiên cứu CERN ở Thụy Sĩ. Hiện nay, HTML đã trở thành một chuẩn Internet được tổ chức W3C (World Wide Web Consortium) vận hành và phát triển. Bạn có thể tự tìm kiếm tình trạng mới nhất của HTML tại bất kỳ thời điểm nào trên Website của W3C.

Phiên bản đầu tiên của **HTML** xuất hiện năm 1991, gồm 18 tag HTML. Phiên bản HTML 4.01 được xuất bản năm 1999. Sau đó, các nhà phát triển đã thay thế HTML bằng XHTML vào năm 2000.

Đến năm 2014, HTML được nâng cấp lên chuẩn HTML5 với nhiều tag được thêm vào markup, mục đích là để xác định rõ nội dung thuộc loại là gì (ví dụ như: <article>, <header>, <footer>,…).

**Theo Mozilla Developer Network**thì**HTML Element Reference** hiện nay có khoảng **hơn 140 tag**. Tuy nhiên một vài tag trong số đó đã bị tạm ngưng (do không được hỗ trợ bởi các trình duyệt hiện hành).

### Ưu và nhược điểm của HTML

Bất kể một ứng dụng hay một loại ngôn ngữ nào thì cũng đều tồn tại những điểm mạnh và điểm yếu nhất định. Và để hiểu hơn HTML là gì, Bizfly sẽ đưa ra những ưu, nhược điểm của HTML dưới đây.

*Ưu điểm:*

* Không giới hạn tài nguyên hỗ trợ và cộng đồng người dùng.
* Khả năng hoạt động mượt mà trên hầu hết các trình duyệt web phổ biến hiện nay.
* Đơn giản trong việc đọc HTML.
* Các markup sử dụng trong HTML thường có độ đồng nhất cao, ngắn gọn, dễ nhìn, dễ hiểu.
* Các mã nguồn mở được sử dụng hoàn toàn miễn phí.
* Được vận hành bởi W3C, HTML chuẩn web.
* Tích hợp các loại ngôn ngữ backend khác như PHP, Node.js,... một cách dễ dàng.

*Nhược điểm:*

* HTML chỉ được sử dụng cho những trang web tĩnh. Nếu muốn tạo thêm những tính năng động thì lập trình viên phải sử dụng thêm các [**ngôn ngữ front end**](https://bizfly.vn/techblog/frontend-la-gi.html) khác như JavaScript, CSS.
* Có khá nhiều các yếu tố trùng lặp như header và footer nếu trang HTML không được tạo riêng biệt.
* Cách đọc và hiển thị file HTML của trình duyệt khó kiểm soát.
* Tồn tại một số trình duyệt cập nhật chậm, không có khả năng hỗ trợ những tính năng mới của HTML.

### HTML hoạt động như thế nào?

Tìm hiểu thêm về nguyên lý, cách HTML hoạt động như thế nào sẽ giúp bạn dễ dàng theo dõi và khắc phục nếu xảy ra sự cố.

[](https://bizflyportal.mediacdn.vn/bizflyportal/images/htm16157919239459.jpg)

HTML hoạt động như thế nào?

Bạn có thể xem các HTML document có đuôi file dạng .html hoặc .htm bằng các trình duyệt web hiện nay như Google Chrome, Cốc cốc, FireFox, Safari,... Các trình duyệt này có nhiệm vụ đọc các file HTML này và biến đổi chúng thành một dạng thông thường để khách hàng có thể đọc và hiểu được chúng.

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

Cấu trúc tổng quát của một trang HTML bao gồm các phần sau:

1. **<!DOCTYPE html>**: Khai báo loại tài liệu, giúp trình duyệt hiểu được phiên bản HTML được sử dụng.
2. **<html>**: Thẻ bắt đầu và kết thúc cho toàn bộ tài liệu HTML.
3. **<head>**: Chứa thông tin meta, tiêu đề trang (title), liên kết đến CSS, JavaScript, và các thông tin khác không hiển thị trực tiếp trên trang web.
4. **<body>**: Chứa nội dung hiển thị trên trang web, bao gồm văn bản, hình ảnh, video, và các thành phần khác.

Một ví dụ cơ bản về cấu trúc HTML:

<!DOCTYPE html>

<html>

<head>

<title>Tiêu đề trang web</title>

</head>

<body>

<h1>Đây là tiêu đề cấp 1</h1>

<p>Đây là đoạn văn bản trong một đoạn <p>.</p>

</body>

</html>

## **2.3. Các thẻ HTML thông dụng**

Dưới đây là một số thẻ HTML thông dụng mà bạn có thể sử dụng khi thiết kế trang web:

1. **<!DOCTYPE html>**: Khai báo phiên bản HTML sử dụng cho trang web.
2. **<html>**: Thẻ bắt đầu cho một tài liệu HTML.
3. **<head>**: Chứa thông tin về tài liệu, như tiêu đề và liên kết đến CSS.
4. **<title>**: Đặt tiêu đề cho trang web.
5. **<body>**: Chứa nội dung hiển thị trên trang web.
6. **<h1> đến <h6>**: Định nghĩa các tiêu đề từ cấp độ 1 đến 6.
7. **<p>**: Định nghĩa một đoạn văn.
8. **<a>**: Tạo liên kết.
9. **<img>**: Chèn hình ảnh.
10. **<ul>**, **<ol>**, **<li>**: Tạo danh sách không thứ tự và có thứ tự.
11. **<table>**, **<tr>**, **<th>**, **<td>**: Tạo bảng và các thành phần của bảng.
12. **<div>**: Định nghĩa một phần tử chứa nội dung.
13. **<span>**: Định nghĩa một phần tử inline.
14. **<form>**: Tạo một biểu mẫu để thu thập dữ liệu từ người dùng.

### - Các thẻ xử lý văn bản

Các thẻ HTML dùng để xử lý văn bản bao gồm:

1. **<p>**: Định nghĩa một đoạn văn bản.
2. **<b>**: In đậm chữ viết.
3. **<strong>**: Nhấn mạnh sự quan trọng của văn bản.
4. **<i>**: In nghiêng chữ viết.
5. **<em>**: Nhấn mạnh ngữ nghĩa của đoạn văn bản.
6. **<u>**: Gạch chân chữ viết.
7. **<strike>**: Gạch ngang chữ viết.
8. **<sub>**: Chữ viết thấp xuống.
9. **<sup>**: Chữ viết cao lên.
10. **<mark>**: Tô sáng chữ viết.
11. **<code>**: Định dạng cho một đoạn mã nào đó.
12. **<hr>**: Thước kẻ ngang trên tài liệu.

Ví dụ sử dụng các thẻ này trong HTML:

<p>Đây là <b>chữ đậm</b> và đây là <strong>chữ quan trọng</strong>.</p>

<p>Đây là <i>chữ nghiêng</i> và đây là <em>chữ nhấn mạnh</em>.</p>

<p>Đây là <u>chữ gạch chân</u> và đây là <strike>chữ gạch ngang</strike>.</p>

<p>Đây là <sub>chữ viết thấp xuống</sub> và đây là <sup>chữ viết cao lên</sup>.</p>

<p>Đây là <mark>chữ tô sáng</mark> và đây là <code>đoạn mã</code>.</p>

<hr>

### - Các thẻ tạo bảng

Các thẻ HTML dùng để tạo bảng bao gồm:

1. **<table>**: Định nghĩa một bảng.
2. **<tr>**: Định nghĩa một hàng trong bảng.
3. **<td>**: Định nghĩa một ô trong bảng.
4. **<th>**: Định nghĩa một ô tiêu đề trong bảng.
5. **<caption>**: Định nghĩa tiêu đề cho bảng.
6. **<colgroup>**: Nhóm một hoặc nhiều cột trong bảng để định dạng.
7. **<col>**: Định nghĩa các thuộc tính cho mỗi cột trong một nhóm cột.
8. **<thead>**: Nhóm nội dung tiêu đề trong bảng.
9. **<tbody>**: Nhóm nội dung chính trong bảng.
10. **<tfoot>**: Nhóm nội dung cuối bảng.

Ví dụ về cách sử dụng các thẻ này để tạo một bảng đơn giản:

<table>

<caption>Thống kê doanh số</caption>

<thead>

<tr>

<th>Tháng</th>

<th>Doanh số</th>

</tr>

</thead>

<tbody>

<tr>

<td>Tháng 1</td>

<td>$100</td>

</tr>

<tr>

<td>Tháng 2</td>

<td>$150</td>

</tr>

</tbody>

<tfoot>

<tr>

<td>Tổng cộng</td>

<td>$250</td>

</tr>

</tfoot>

</table>

### - Thẻ liên kết

Thẻ liên kết trong HTML, được biểu diễn bằng thẻ <a>, cho phép bạn tạo ra một siêu liên kết đến một trang web khác, một file, địa chỉ email, hoặc bất kỳ loại địa chỉ URL nào. Dưới đây là cấu trúc cơ bản và một số thuộc tính của thẻ <a>:

* **Cấu trúc cơ bản**: <a href="URL">nội dung đại diện cho cái liên kết</a>
  + href: Đường dẫn đến tài liệu mà bạn muốn người dùng được chuyển đến.
  + Nội dung đại diện: Có thể là văn bản hoặc hình ảnh.
* **Thuộc tính target**: Xác định cách thức mở một trang web.
  + \_self: Mở liên kết trong tab hiện tại (mặc định).
  + \_blank: Mở liên kết trong một tab mới.
  + \_parent: Mở liên kết trong frame chứa frame của trang web hiện tại (nếu có).
* **Thuộc tính rel**: Định nghĩa mối quan hệ giữa trang hiện tại và trang liên kết.
  + nofollow: Báo cho các công cụ tìm kiếm biết rằng không nên theo liên kết này.

Ví dụ sử dụng thẻ <a>:

<a href="https://www.example.com">Trang chủ</a>

<a href="mailto:example@example.com">Gửi Email</a>

<a href="tel:+1234567890">Gọi điện thoại</a>

<a href="https://www.example.com" target="\_blank">Mở trong tab mới</a>

### - Các thẻ đa phương tiện

Các thẻ HTML đa phương tiện cho phép bạn nhúng video, âm thanh, và các loại tệp đa phương tiện khác vào trang web của bạn. Dưới đây là một số thẻ thông dụng:

1. **<audio>**: Dùng để nhúng nội dung âm thanh.
2. <audio controls>
3. <source src="audio.mp3" type="audio/mpeg">
4. Trình duyệt của bạn không hỗ trợ thẻ audio.
5. </audio>
6. **<video>**: Dùng để nhúng nội dung video.
7. <video width="320" height="240" controls>
8. <source src="movie.mp4" type="video/mp4">
9. Trình duyệt của bạn không hỗ trợ thẻ video.
10. </video>
11. **<embed>**: Dùng để nhúng nội dung đa phương tiện như Flash, PDF, hoặc các plugin khác.
12. <embed src="file.swf" width="500" height="300">
13. **<object>**: Cũng dùng để nhúng đa phương tiện nhưng cung cấp thêm tính năng như thẻ <param> để định cấu hình.
14. <object data="file.swf" width="500" height="300">
15. <param name="movie" value="file.swf">
16. <param name="quality" value="high">
17. </object>
18. **<source>**: Dùng trong thẻ <audio> hoặc <video> để chỉ định nhiều nguồn cho một loại đa phương tiện, giúp tương thích với nhiều trình duyệt.
19. **<track>**: Dùng với thẻ <video> để thêm phụ đề.

### - Thẻ tạo khung

Trong HTML, thẻ <frameset> và <frame> từng được sử dụng để tạo khung (frames), cho phép chia cửa sổ trình duyệt thành nhiều phần và mỗi phần có thể chứa một trang web riêng biệt. Tuy nhiên, cả hai thẻ này đều không được hỗ trợ trong HTML5 và đã bị loại bỏ do các vấn đề về thiết kế và bảo mật.

Thay vào đó, bạn có thể sử dụng thẻ <iframe> để nhúng một trang web hoặc tài liệu vào trong một trang web khác. Dưới đây là cách sử dụng thẻ <iframe>:

<iframe src="url\_của\_trang\_web" width="chiều\_rộng" height="chiều\_cao"></iframe>

* src: Đường dẫn URL của trang web bạn muốn nhúng.
* width và height: Đặt chiều rộng và chiều cao của khung.

Ví dụ:

<iframe src="https://www.example.com" width="600" height="400"></iframe>

Thẻ <iframe> cũng có thêm các thuộc tính như frameborder để xác định viền của khung, scrolling để kiểm soát thanh cuộn, và allowfullscreen để cho phép xem toàn màn hình.

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

Các thẻ HTML dùng để tạo biểu mẫu (form) cho phép thu thập dữ liệu từ người dùng và gửi chúng đến máy chủ. Dưới đây là một số thẻ cơ bản:

1. **<form>**: Định nghĩa một biểu mẫu và bao bọc các phần tử khác.
2. **<input>**: Tạo một trường nhập dữ liệu. Có nhiều loại như text, radio, checkbox, submit, v.v.
3. **<label>**: Gán nhãn cho các phần tử input để cải thiện khả năng truy cập.
4. **<textarea>**: Tạo một trường nhập văn bản nhiều dòng.
5. **<button>**: Tạo một nút có thể tùy chỉnh.
6. **<select>**: Tạo một danh sách lựa chọn thả xuống.
7. **<option>**: Định nghĩa các lựa chọn trong thẻ select.
8. **<fieldset>**: Nhóm các phần tử liên quan trong một biểu mẫu.
9. **<legend>**: Đặt tiêu đề cho nhóm fieldset.

Ví dụ về một biểu mẫu đơn giản:

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

<label for="name">Tên:</label>

<input type="text" id="name" name="user\_name">

<label for="email">Email:</label>

<input type="email" id="email" name="user\_email">

<input type="submit" value="Gửi">

</form>

## **2.5. Một số thẻ HTML đặc biệt**

* Dưới đây là một số thẻ HTML đặc biệt và quan trọng mà bạn có thể gặp trong việc xây dựng trang web:
* Thẻ <meta>: Thẻ này được sử dụng để cung cấp các siêu dữ liệu (metadata) về tài liệu HTML, như mô tả, từ khóa, trang tác giả, mã ngôn ngữ, và các thông tin khác mà trình duyệt hoặc công cụ tìm kiếm có thể sử dụng.
* Thẻ <link>: Thẻ này được sử dụng để liên kết với các tài nguyên bên ngoài như các tập tin CSS, biểu mẫu RSS, và các favicon.
* Thẻ <script>: Thẻ này được sử dụng để nhúng mã JavaScript vào trong tài liệu HTML. Mã này có thể được sử dụng để tạo ra các tương tác động trên trang web.
* Thẻ <style>: Thẻ này được sử dụng để định dạng CSS trực tiếp trong tài liệu HTML. Điều này hữu ích khi bạn muốn áp dụng một số kiểu CSS cụ thể cho một phần của trang web.
* Thẻ <base>: Thẻ này được sử dụng để thiết lập một URL cơ sở cho toàn bộ các liên kết trong tài liệu HTML.
* Thẻ <noscript>: Thẻ này được sử dụng để cung cấp nội dung dự phòng cho trình duyệt không hỗ trợ JavaScript hoặc khi JavaScript bị vô hiệu hóa.

## **2.6. HTML5**

* **HTML5 là gì?**

HTML5 là phiên bản mới nhất của ngôn ngữ đánh dấu Hypertext Markup Language (HTML), được sử dụng để tạo các trang web. Nó cung cấp các thẻ mới và tính năng mới như tương tác, đồ họa, và video, giúp cho việc tạo trang web trở nên dễ dàng hơn. HTML5 cũng hỗ trợ cho việc xây dựng ứng dụng web và trò chơi trên các thiết bị di động.

* **Cú pháp HTML5**

Cú pháp HTML5 được thiết kế để tương thích với các phiên bản HTML trước đây và cung cấp nhiều sự linh hoạt hơn. Dưới đây là một số điểm cơ bản trong cú pháp HTML5:

1. **DOCTYPE**: Khai báo loại tài liệu HTML5 đơn giản chỉ cần <!DOCTYPE html>[1](https://viettuts.vn/html5/cu-phap-html5-co-ban).
2. **Mã hóa ký tự**: Sử dụng <meta charset="UTF-8"> để đặt mã hóa ký tự cho tài liệu[1](https://viettuts.vn/html5/cu-phap-html5-co-ban).
3. **Thẻ <script>**: Không cần thuộc tính type khi nhúng JavaScript, chỉ cần <script src="scriptfile.js"></script>[1](https://viettuts.vn/html5/cu-phap-html5-co-ban).
4. **Thẻ <link>**: Đối với CSS, không cần thuộc tính type, chỉ cần <link rel="stylesheet" href="stylefile.css">[1](https://viettuts.vn/html5/cu-phap-html5-co-ban).
5. **Các phần tử HTML5**: Các phần tử được đánh dấu bằng thẻ mở và thẻ đóng, ví dụ <p>...</p>. Các tên thẻ không phân biệt chữ hoa hay chữ thường[1](https://viettuts.vn/html5/cu-phap-html5-co-ban).
6. **Thuộc tính**: Các thuộc tính có thể không cần trích dẫn nếu không chứa khoảng trắng hoặc ký tự đặc biệt, ví dụ <div class=example>...</div>[1](https://viettuts.vn/html5/cu-phap-html5-co-ban).

* **Khai báo DOCTYPE trong HTML5**
* Trong HTML5, khai báo DOCTYPE rất đơn giản và được thực hiện ngay tại dòng đầu tiên của văn bản HTML, trước thẻ <html>. Cú pháp cho khai báo DOCTYPE trong HTML5 là:
* <!DOCTYPE html>
* Khai báo này thông báo cho trình duyệt rằng tài liệu nên được hiểu và hiển thị theo chuẩn HTML5. Đây là một phần quan trọng để đảm bảo rằng trang web của bạn hoạt động chính xác trên các trình duyệt khác nhau và tuân theo các tiêu chuẩn web hiện đại.
* **Thẻ ngữ nghĩa**

Trong HTML5, các thẻ ngữ nghĩa (semantic tags) được sử dụng để mô tả ý nghĩa cấu trúc của nội dung trang web, giúp cho cả trình duyệt và người phát triển hiểu rõ hơn về các phần của trang web. Dưới đây là một số thẻ ngữ nghĩa thông dụng:

* **<article>**: Định nghĩa một nội dung độc lập hoặc tự chứa, như bài viết, bài đăng diễn đàn, hoặc bài báo.
* **<aside>**: Chứa nội dung liên quan nhưng không chính yếu so với nội dung xung quanh nó, thường là thanh bên hoặc ghi chú.
* **<details>**: Tạo một widget mà người dùng có thể mở ra để xem thêm thông tin chi tiết.
* **<figcaption>**: Định nghĩa chú thích cho một phần tử <figure>.
* **<figure>**: Định nghĩa nội dung tự chứa, thường là hình ảnh, biểu đồ, mã code, mà có thể được di chuyển trong tài liệu mà không ảnh hưởng đến dòng chảy chính.
* **<footer>**: Định nghĩa chân trang của một phần hoặc toàn bộ tài liệu.
* **<header>**: Định nghĩa đầu trang của một phần hoặc toàn bộ tài liệu.
* **<main>**: Chứa nội dung chính của tài liệu hoặc ứng dụng.
* **<mark>**: Đánh dấu hoặc làm nổi bật một phần của văn bản.
* **<nav>**: Định nghĩa tập hợp các liên kết điều hướng.
* **<section>**: Định nghĩa một phần của tài liệu, thường có một tiêu đề và liên quan đến một chủ đề hoặc chức năng cụ thể.
* **<summary>**: Định nghĩa tiêu đề cho phần tử <details>.
* **<time>**: Định nghĩa ngày tháng hoặc thời gian.

Sử dụng các thẻ ngữ nghĩa giúp cải thiện khả năng truy cập của trang web, tối ưu hóa công cụ tìm kiếm (SEO), và làm cho mã nguồn dễ đọc hơn cho các nhà phát triển.

* **Webform**

Webform, hay còn gọi là biểu mẫu web, là một phần của trang web cho phép người dùng nhập dữ liệu hoặc thông tin cá nhân, sau đó được gửi đến máy chủ để xử lý. Trong ASP.NET, Web Forms là một phần của khung ứng dụng web ASP.NET và được bao gồm trong Visual Studio. Nó là một trong bốn mô hình lập trình bạn có thể sử dụng để tạo ứng dụng web ASP.NET. Các Web Forms là các trang mà người dùng yêu cầu thông qua trình duyệt của họ. Những trang này có thể được viết bằng sự kết hợp của HTML, mã client, điều khiển máy chủ, và mã máy chủ. Khi người dùng yêu cầu một trang, nó được biên dịch và thực thi trên máy chủ bởi khung ứng dụng, và sau đó khung ứng dụng tạo ra mã HTML mà trình duyệt có thể hiển thị.

**Kết luận:** HTML5 đã đánh dấu một bước tiến quan trọng trong lịch sử của ngôn ngữ đánh dấu siêu văn bản. Với sự giới thiệu của các thẻ mới và cú pháp linh hoạt hơn, HTML5 không chỉ tăng cường khả năng tương thích với các trình duyệt và thiết bị khác nhau mà còn mở rộng khả năng biểu đạt nội dung web phong phú và đa dạng. Các tính năng như Web Storage, Canvas, và WebSockets đã mở ra cánh cửa cho các ứng dụng web tương tác và đa phương tiện, trong khi đảm bảo tính bảo mật và hiệu suất cao. HTML5 không chỉ là một bước tiến trong công nghệ web mà còn là một nền tảng vững chắc cho tương lai của lập trình web.

# **CHƯƠNG III: CSS và CSS3**

 (Cascading Style Sheets)

## **3.1. CSS là gì?**

CSS là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (HTML). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc…

CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, vì HTML không được thiết kế để gắn tag để giúp định dạng trang web.

Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.

## **3.2. Cú pháp CSS**

CSS được tạo thành từ các Style Rule. Các rule này được thông dịch bởi trình duyệt và sau đó được áp dụng tới các phần tử tương ứng trong tài liệu của bạn. Style Rule bao gồm ba phần:

Selector: (hiểu nôm na là bộ chọn phần tử) Một selector là một thẻ HTML mà tại đó một bạn áp dụng một style cho nó. Đó có thể là bất kỳ một thẻ HTML nào, như thẻ div, thẻ span, thẻ p, …

Property: (thuộc tính) Là property của một kiểu thuộc tính của thẻ HTML. Nói một cách đơn giản thì tất cả các thuộc tính trong HTML được chuyển đổi thành các CSS property. Đó có thể là color, border....

Value: Là các giá trị được gán cho các property. Ví dụ, color có thể có các giá trị là red #F1F1F1 etc.

Dưới đây là cú pháp CSS chung:



Trong ví dụ trên, selector trỏ tới phần tử HTML là h1. Đây là phần tử mà mình muốn tạo style cho nó.

Khối khai báo Declaration có thể bao gồm một hoặc nhiều khối và được đặt phân biệt nhau bởi dấu chấm phảy.

Mỗi khối khai báo này bao gồm một tên của CSS property (trong ví dụ là color và font-size) và một giá trị (trong ví dụ là blue và 12px) được gán cho property đó.

Bạn cần chú ý một điều rằng, một khai báo CSS luôn luôn kết thúc với một dấu chấm phảy và phần khai báo này sẽ được bao quanh bởi cặp dấu ngoặc ôm.

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

## **3.4. Selectors**

Bộ chọn (Selector): là mẫu để chọn phần tử HTML mà bạn muốn định nghĩa phong cách. Các selector được áp dụng cho các trường hợp sau:

Tất cả phần tử theo một dạng cụ thể nào đó, ví dụ phần tử tiêu đề h1.

Thuộc tính id và class của các phần tử.

Các phần tử dựa vào mối liên quan với các phần tử khác trong cây phân cấp tài liệu.

### 3.4.1. Universal selector

Sử dụng dấu '\*' để chọn tất cả các phần tử trong tài liệu HTML.

### 3.4.2. Type selector

Chọn tất cả các phần tử của một loại cụ thể, ví dụ như chọn tất cả các phần tử **<p>**.

### 3.4.3. Identity selector

Sử dụng id của một phần tử để chọn nó. Đây là một lựa chọn độc nhất vì mỗi id phải là duy nhất trong một tài liệu.

### 3.4.4. Class selector

Sử dụng id của một phần tử để chọn nó. Đây là một lựa chọn độc nhất vì mỗi id phải là duy nhất trong một tài liệu.

### 3.4.5. Descendant selector

Chọn các phần tử con của một phần tử cụ thể. Ví dụ, **div p** sẽ chọn tất cả các phần tử **<p>** nằm trong các phần tử **<div>**.

### 3.4.6. Child selector

Tương tự như descendant selector, nhưng chỉ chọn các phần tử con trực tiếp của một phần tử cụ thể.

### 3.4.7. Adjacent sibling selector

Chọn phần tử kế tiếp cùng một cấp cha. Ví dụ, **h2 + p** sẽ chọn tất cả các phần tử **<p>** được kế tiếp ngay sau phần tử **<h2>**.

### 3.4.8. Attribute selector

Chọn các phần tử dựa trên thuộc tính của chúng. Ví dụ, **[href]** sẽ chọn tất cả các phần tử có thuộc tính **href**.

### 3.4.9. Pseudo class selector

Chọn các phần tử dựa trên trạng thái hoặc hành vi của chúng. Ví dụ, **:hover** sẽ chọn phần tử khi con trỏ chuột di chuyển qua phần tử đó.

### 3.4.10. Group selector

Cho phép bạn kết hợp nhiều lựa chọn lại với nhau bằng cách sử dụng dấu phẩy ','. Ví dụ, **h1, h2, h3** sẽ áp dụng các quy tắc cho tất cả các phần tử **<h1>**, **<h2>**, và **<h3>**.

## **3.5. Đơn vị đo lường CSS**

Chắc các bạn đã rất quen thuộc với những đơn vị tính như px, pt để đặt kích thước cho các phần tử trên giao diện website như width, height, margin, padding, ... Vậy cụ thể về các đơn vị tính trong CSS như thế nào, các bạn cùng mình tìm hiểu trong bài viết này nhé.

Những đơn vị đo trong CSS.

Theo những tìm hiểu của mình và trong quá trình làm việc thực tế ở các dự án thì có các đơn vị tính trong CSS: px, pt, in, pc, mm, cm, em, ex, ch, rem, vw, vh, vmin, vmax, %...

Quy định về đơn vị trong CSS:

Giá trị số rồi đến đơn vị, VD: 10px, 2em, 50%, ...

Tuyệt đối không có dấu cách giữa số và đơn vị tính.

Nếu giá trị số là 0, có thể bỏ qua đơn vị tính.

Một số đơn vị cho phép sử dụng số âm. VD: padding-top:-20px; margin-top: -20%; Nhưng phần lớn các đơn vị khác là số dương.

Trong khuân khổ bài chia sẻ này mình sẽ cùng các bạn tìm hiểu những đơn vị tính thường sử dụng trong CSS. Mặc dù sẽ không đề cập hết nhưng không có nghĩa là những đơn vị đó sẽ không được sử dụng, các bạn nên tìm hiểu qua để biết và lựa chọn đơn vị tính phù hợp.

Trong CSS có hai loại đơn vị là Đơn vị tương đối (Relative Units) và Đơn vị tuyệt đối (Absolute Units). Trong đó:

**Đơn vị tương đối** là đơn vị được tính một cách tương đối so với phần tử mẹ hoặc các phần tử khác. Đơn vị tương đối cơ động và hoạt động tốt hơn cho những thiết bị, những màn hình có kích thước và độ phân giải khác nhau.

Khác với đơn vị tương đối, như tên gọi, đơn vị tuyệt đối sẽ có những giá trị cố định không thay đổi. Đơn vị tuyệt đối không được khuyến khích sử dụng cho màn hình, vì có rất nhiều các kích thước màn hình khác nhau. Nó chỉ nên dùng cho những trường hợp biết chắc kích thước của giao diện, như xuất ra máy in chẳng hạn.

## **3.6. Kế thừa thuộc tính**

 Thuộc tính kế thừa (inherited properties): là những thuộc tính ảnh hưởng tới thuộc tính con của nó. Tất cả các thuộc tính ảnh hưởng tới text đều là thuộc tính kế thừa. Ví dụ, nếu chúng ta chọn font-size cho một phần tử HTML, giá trị đó sẽ được áp dụng cho tất cả các phần tử con của nó.

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

- Thuộc tính định dạng font chữ, văn bản (font, text)

Cung cấp các thuộc tính để điều chỉnh font chữ, kích thước chữ, màu sắc, định dạng văn bản và căn chỉnh.

Ví dụ: font-family, font-size, color, text-align, line-height.

- Thuộc tính định dạng nền (background)

Cho phép tùy chỉnh nền của các phần tử HTML, bao gồm màu nền, hình nền và hiệu ứng nền.

Ví dụ: background-color, background-image, background-repeat, background-size.

- Mô hình hộp (box model)

Xác định cách mà các phần tử HTML được căn chỉnh và xếp chồng lên nhau.

Bao gồm các thuộc tính như width, height, margin, padding, border.

Mô hình hộp bao gồm phần nội dung, padding, border và margin.

- Các thuộc tính định margin, padding, border

margin: Định nghĩa khoảng cách giữa các phần tử.

padding: Định nghĩa khoảng cách giữa nội dung và ranh giới của phần tử.

border: Định nghĩa đường viền của phần tử.

Các thuộc tính liên quan: margin-top, margin-bottom, padding-left, border-width.

- Thuộc tính định dạng cách hiển thị (display)

Xác định cách mà một phần tử HTML được hiển thị trong luồng tài liệu.

Ví dụ: display: block, display: inline, display: inline-block.

- Thuộc tính xác định vị trí (position)

Xác định cách một phần tử HTML được định vị trong văn bản.

Các giá trị có thể bao gồm static, relative, absolute, fixed.

Ví dụ: position: static, position: relative, position: absolute.

Các nhóm thuộc tính này cung cấp các công cụ quan trọng để điều chỉnh giao diện của trang web và tạo ra các trải nghiệm người dùng tốt hơn.

## **3.8. Float & Clear**

Thuộc tính float cho phép bạn di chuyển một phần tử ra khỏi luồng bình thường của tài liệu và căn chỉnh nó sang một bên (trái hoặc phải) của phần tử chứa nó.

Khi một phần tử được float, các phần tử khác có thể "bơi" xung quanh nó. Điều này thường được sử dụng để tạo bố cục đa cột hoặc để làm cho hình ảnh "bơi" trong văn bản.

Thuộc tính clear được sử dụng để quyết định phần tử nào sẽ không được cho phép "bơi" xung quanh các phần tử được float.

Điều này thường được sử dụng để giữ các phần tử sau các phần tử float, để tránh việc chồng chéo.

Các giá trị thông thường của thuộc tính clear bao gồm left, right, both, và none.

Float và Clear thường được sử dụng cùng nhau để tạo ra các bố cục đa cột và xử lý vị trí của các phần tử trong giao diện web. Tuy nhiên, chúng cũng có thể gây ra một số vấn đề như việc "mất cân bằng" các phần tử hoặc tạo ra các khoảng trống không mong muốn nếu không được sử dụng đúng cách.

## **3.9. Flex**

Flexbox là một phần của CSS3 giúp quản lý bố cục và sắp xếp các phần tử trên một trang web một cách linh hoạt và dễ dàng. Nó cung cấp một cách mạnh mẽ để điều chỉnh và tự động căn chỉnh các phần tử trong một container theo một cách linh hoạt.

Dưới đây là một số thuộc tính quan trọng của Flexbox:

display: flex;

Khi một container được thiết lập với display: flex;, nó trở thành một container flex, và các phần tử con bên trong nó sẽ tuân theo các quy tắc của flexbox.

justify-content:

Thuộc tính này được sử dụng để căn chỉnh các phần tử con theo hướng chính của container flex (theo chiều ngang, từ trái sang phải hoặc từ phải sang trái).

Ví dụ: justify-content: center; để căn chỉnh các phần tử con vào giữa container.

align-items:

Thuộc tính này được sử dụng để căn chỉnh các phần tử con theo hướng phụ của container flex (theo chiều dọc, từ trên xuống hoặc từ dưới lên).

Ví dụ: align-items: center; để căn chỉnh các phần tử con vào giữa container.

flex-direction:

Xác định hướng các phần tử con được xếp sắp trong container flex.

Các giá trị có thể bao gồm: row, row-reverse, column, column-reverse.

flex-wrap:

Xác định liệu các phần tử con có được phép xuống dòng nếu không đủ không gian trong container hay không.

Các giá trị có thể bao gồm: nowrap, wrap, wrap-reverse.

flex-grow:

Xác định tỷ lệ mà các phần tử con sẽ mở rộng để điền vào không gian trống trong container flex.

Giá trị mặc định là 0, tức là các phần tử con không mở rộng.

Flexbox là một công cụ mạnh mẽ để tạo ra các bố cục linh hoạt và đáp ứng trên các trang web. Nó giúp giải quyết nhiều vấn đề về bố cục mà trước đây thường gặp phải khi sử dụng CSS thông thường.

## **3.10. Grid**

CSS Grid là một phần của CSS3, cung cấp một cách mạnh mẽ để tạo ra các bố cục hai chiều (dọc và ngang) trên trang web. Nó cho phép bạn chia các phần tử trên trang thành lưới đa chiều, giúp bạn quản lý bố cục một cách linh hoạt và hiệu quả.

Dưới đây là một số khái niệm cơ bản và thuộc tính quan trọng của CSS Grid:

Grid Container và Grid Item:

Trong CSS Grid, các phần tử được chia thành hai loại chính: Grid Container (bên ngoài) và Grid Item (bên trong).

Grid Container là phần tử cha chứa các Grid Item, được thiết lập với thuộc tính display: grid; hoặc display: inline-grid;.

Grid Item là các phần tử con của Grid Container.

Grid Lines:

Các đường grid lines là các đường ảo dọc và ngang tạo thành lưới của Grid.

Có thể định nghĩa các grid lines bằng cách sử dụng các giá trị trong thuộc tính grid-template-rows và grid-template-columns.

Grid Rows và Grid Columns:

Grid Rows là các phần tử nằm giữa hai grid lines dọc.

Grid Columns là các phần tử nằm giữa hai grid lines ngang.

Bạn có thể xác định số lượng và kích thước của các hàng và cột bằng cách sử dụng các giá trị trong grid-template-rows và grid-template-columns.

Grid Template Areas:

Cho phép bạn xác định các khu vực của lưới bằng cách sử dụng các tên cho các phần tử grid.

Các tên được sử dụng để xác định khu vực của các Grid Item trong thuộc tính grid-template-areas.

Grid Gap:

Là khoảng cách giữa các Grid Item trong lưới.

Có thể được đặt bằng thuộc tính grid-row-gap và grid-column-gap, hoặc sử dụng thuộc tính rút gọn grid-gap.

CSS Grid cung cấp một cách tiếp cận mạnh mẽ để tạo ra các bố cục phức tạp và linh hoạt trên trang web, và là một công cụ hiệu quả cho việc quản lý bố cục trên các trình duyệt hiện đại.

## **3.11. CSS3**

- CSS3 là gì?

CSS3 là phiên bản tiếp theo của ngôn ngữ mô tả kiểu dạng trang web - Cascading Style Sheets (CSS), với nhiều tính năng mới và cải tiến so với các phiên bản trước đó.

- CSS3 Selectors

CSS3 mở rộng các lựa chọn của các phần tử trong HTML bằng cách cung cấp các lựa chọn mới như lựa chọn thuộc tính, lựa chọn ngôn ngữ và lựa chọn con trực tiếp.

- CSS3 Rounded Corners

CSS3 cho phép bạn tạo các góc bo tròn cho các phần tử bằng cách sử dụng thuộc tính border-radius.

- CSS3 Border Images

Với CSS3, bạn có thể sử dụng hình ảnh làm nền cho các đường viền của phần tử bằng cách sử dụng thuộc tính border-image.

- CSS3 Backgrounds

CSS3 cung cấp các tính năng nâng cao cho việc quản lý hình ảnh và màu sắc nền của các phần tử, bao gồm sử dụng multiple background images và background size.

- CSS Gradients

CSS3 cho phép tạo ra gradient màu sắc một cách linh hoạt và dễ dàng bằng cách sử dụng thuộc tính linear-gradient và radial-gradient.

- CSS3 Shadows

CSS3 cung cấp các tính năng để tạo ra đổ bóng cho các phần tử bằng cách sử dụng thuộc tính box-shadow và text-shadow.

- CSS3 Fonts

CSS3 cung cấp các tính năng mới cho việc quản lý và định dạng font chữ, bao gồm sử dụng web fonts và custom font faces.

- CSS3 2D Transforms

CSS3 cung cấp các tính năng để biến đổi các phần tử trong không gian 2D, bao gồm di chuyển, quay, scale và skew.

- CSS3 3D Transforms

Ngoài các biến đổi 2D, CSS3 cũng hỗ trợ biến đổi 3D cho các phần tử, cho phép bạn tạo ra các hiệu ứng 3D phức tạp.

- CSS3 Transitions

CSS3 cho phép tạo ra các hiệu ứng chuyển đổi mềm mại khi trạng thái của các phần tử thay đổi, như hover hoặc focus, bằng cách sử dụng thuộc tính transition.

- CSS3 Multiple Columns

CSS3 cho phép chia nội dung của một phần tử thành nhiều cột bằng cách sử dụng thuộc tính column-count và column-width.

- CSS3 Animations

CSS3 cung cấp các tính năng để tạo ra các hoạt hình động một cách linh hoạt và mạnh mẽ bằng cách sử dụng @keyframes và animation định nghĩa.

## **3.12. SCSS**

SCSS (Sassy CSS) là một tiền xử lý CSS, có nghĩa là nó mở rộng và cải thiện cú pháp của CSS thông thường. SCSS được phát triển từ ngôn ngữ Sass (Syntactically Awesome Stylesheets) và có cú pháp tương tự như CSS, nhưng với nhiều tính năng tiện ích hơn.

Dưới đây là một số tính năng chính của SCSS:

Biến:

SCSS cho phép bạn định nghĩa và sử dụng biến, giúp tái sử dụng mã CSS và dễ dàng thay đổi các giá trị một cách hiệu quả.

Nesting:

Bạn có thể lồng các quy tắc CSS bên trong nhau trong SCSS, giúp làm cho mã nguồn dễ đọc hơn và giảm bớt sự lặp lại.

Mixins:

Mixins là các khối mã CSS có thể được sử dụng lại nhiều lần trong tài liệu SCSS. Chúng giúp giảm sự lặp lại và tăng tính tái sử dụng mã nguồn.

Định nghĩa hàm:

SCSS cho phép bạn định nghĩa các hàm CSS, giúp bạn tái sử dụng mã và thực hiện các phép tính phức tạp.

Import:

Bạn có thể chia nhỏ mã SCSS thành các tệp nhỏ hơn và import chúng vào tệp chính, giúp quản lý dự án dễ dàng hơn.

Điều kiện và vòng lặp:

SCSS hỗ trợ các cấu trúc điều kiện và vòng lặp, giúp bạn tạo ra mã CSS động dựa trên các điều kiện và thực hiện các hoạt động lặp lại.

SCSS là một công cụ mạnh mẽ cho việc viết mã CSS hiệu quả và dễ bảo trì. Nó giúp giảm thời gian và công sức khi phát triển và duy trì các dự án CSS lớn.

## **3.13. SASS**

SASS (Syntactically Awesome Stylesheets) là một ngôn ngữ tiền xử lý CSS, giúp mở rộng và cải thiện cú pháp của CSS thông thường. SASS cho phép bạn viết mã CSS một cách súc tích và dễ đọc hơn, với các tính năng như biến, nesting, mixins, và nhiều tính năng khác.

Dưới đây là một số tính năng chính của SASS:

Biến:

SASS cho phép bạn định nghĩa và sử dụng biến, giúp tái sử dụng mã CSS và dễ dàng thay đổi các giá trị một cách hiệu quả.

Nesting:

Bạn có thể lồng các quy tắc CSS bên trong nhau trong SASS, giúp làm cho mã nguồn dễ đọc hơn và giảm bớt sự lặp lại.

Mixins:

Mixins là các khối mã CSS có thể được sử dụng lại nhiều lần trong tài liệu SASS. Chúng giúp giảm sự lặp lại và tăng tính tái sử dụng mã nguồn.

Định nghĩa hàm:

SASS cho phép bạn định nghĩa các hàm CSS, giúp bạn tái sử dụng mã và thực hiện các phép tính phức tạp.

Import:

Bạn có thể chia nhỏ mã SASS thành các tệp nhỏ hơn và import chúng vào tệp chính, giúp quản lý dự án dễ dàng hơn.

Điều kiện và vòng lặp:

SASS hỗ trợ các cấu trúc điều kiện và vòng lặp, giúp bạn tạo ra mã CSS động dựa trên các điều kiện và thực hiện các hoạt động lặp lại.

SASS giúp tăng hiệu suất và linh hoạt trong việc viết và quản lý mã CSS. Nó là một công cụ mạnh mẽ cho phát triển web và được sử dụng rộng rãi trong cộng đồng phát triển web.

Kết luận: CSS là ngôn ngữ định dạng dòng được sử dụng rộng rãi trong phát triển web để kiểm soát giao diện và bố cục của trang. Nó cung cấp các quy tắc và thuộc tính để tùy chỉnh các phần tử HTML và làm cho trang web trở nên hấp dẫn và thú vị. CSS3 là phiên bản tiếp theo của CSS, mở rộng các khả năng của nó bằng cách thêm các tính năng mới như định dạng văn bản nâng cao, hiệu ứng, và các tính năng thiết kế linh hoạt.

CSS và CSS3 cung cấp một nền tảng mạnh mẽ cho việc thiết kế và phát triển giao diện web hiện đại và linh hoạt. Sự tiếp tục phát triển của chúng mang lại nhiều cơ hội mới cho các nhà phát triển và thiết kế web để tạo ra trải nghiệm người dùng tốt nhất có thể.

# **CHƯƠNG 4: THIẾT KẾ WEBISTE THEO ĐỀ TÀI BẠN CHỌN**

## **4.1. Ý tưởng của Website:**

Thiết kế web bán quần áo là 1 cơ hội tuyệt vời để đưa ra sự trải nghiệm và mua sắm hợp lí.Dưới đây là 1 số ý tưởng về web bạn có thể tham khảo và xem xét:

Giao diện và bố trí:giao diện bất mắt ,bố trí hợp lí.Giúp người dùng quen thuộc, dễ sử dụng

Thiết kế Responsive: Đảm bảo trang web phản hồi tốt trên các thiết bị di động và desktop, để người dùng có thể trải nghiệm mua sắm một cách thuận tiện từ mọi nơi.

Hình ảnh chất lượng cao: Sử dụng hình ảnh chất lượng cao để hiển thị sản phẩm, cung cấp nhiều bộ quần Áo đa dạng

Giỏ hàng tiện lợi: Tạo ra một giao diện giỏ hàng thuận tiện và dễ sử dụng, cho phép người dùng xem và chỉnh sửa đơn hàng của họ trước khi thanh toán.

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

***Header (Phần Đầu Trang):***

-Logo của thương hiệu hoặc trang web.

-Thanh điều hướng chính để người dùng dễ dàng di chuyển đến các trang quan trọng như danh mục sản phẩm, trang chủ, giỏ hàng và trang thanh toán.

-Công cụ tìm kiếm để người dùng có thể tìm kiếm sản phẩm một cách nhanh chóng.

-Liên kết đến trang đăng nhập hoặc đăng ký tài khoản.

***Banner hoặc Slideshow:***

-Hiển thị hình ảnh chất lượng cao của các sản phẩm giày dép nổi bật hoặc các chương trình khuyến mãi đặc biệt.

-Có thể bao gồm các thông điệp quảng cáo để thu hút sự chú ý của khách hàng.

***Danh Mục Sản Phẩm:***

-Hiển thị các danh mục sản phẩm chính như giày nam, giày nữ, giày thể thao, giày cao gót, và các loại khác.

Một menu dọc hoặc ngang để người dùng dễ dàng lựa chọn danh mục sản phẩm mình -quan tâm.

***Sản Phẩm Nổi Bật Hoặc Mới Nhất:***

-Hiển thị một số sản phẩm nổi bật hoặc mới nhất để thu hút sự chú ý của khách hàng.

-Cung cấp các liên kết để điều hướng người dùng đến các trang sản phẩm cụ thể.

***Thông Tin Thương Hiệu:***

-Một phần giới thiệu ngắn về thương hiệu, bao gồm câu chuyện của thương hiệu, cam kết sản phẩm và các giải thưởng hoặc đối tác đã được công nhận.

***Phần Đánh Giá và Nhận Xét:***

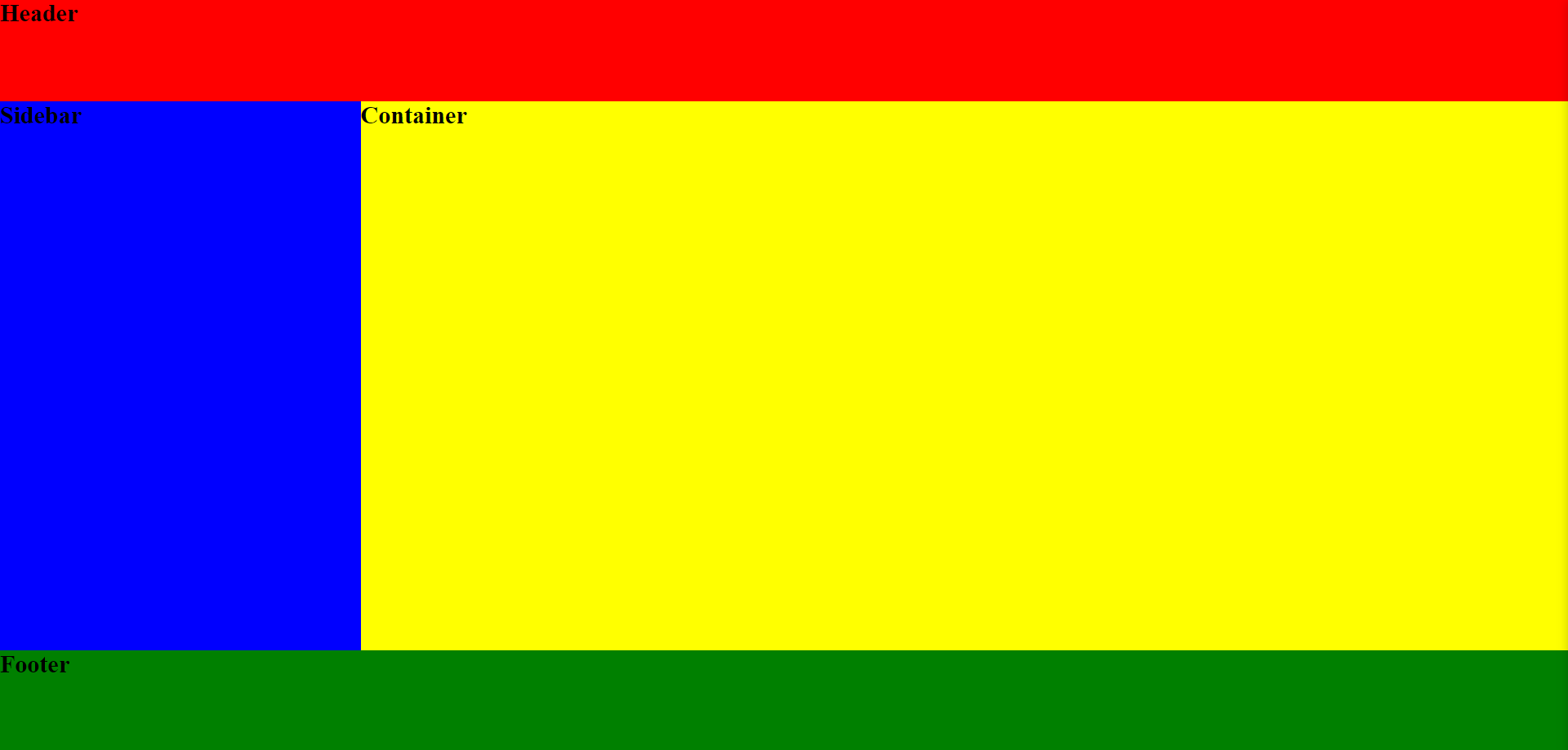
-Hiển thị các đánh giá và nhận xét từ khách hàng trước đó về sản phẩm hoặc dịch vụ.

-Cung cấp một nút hoặc liên kết để khuyến khích người dùng viết đánh giá của riêng họ.

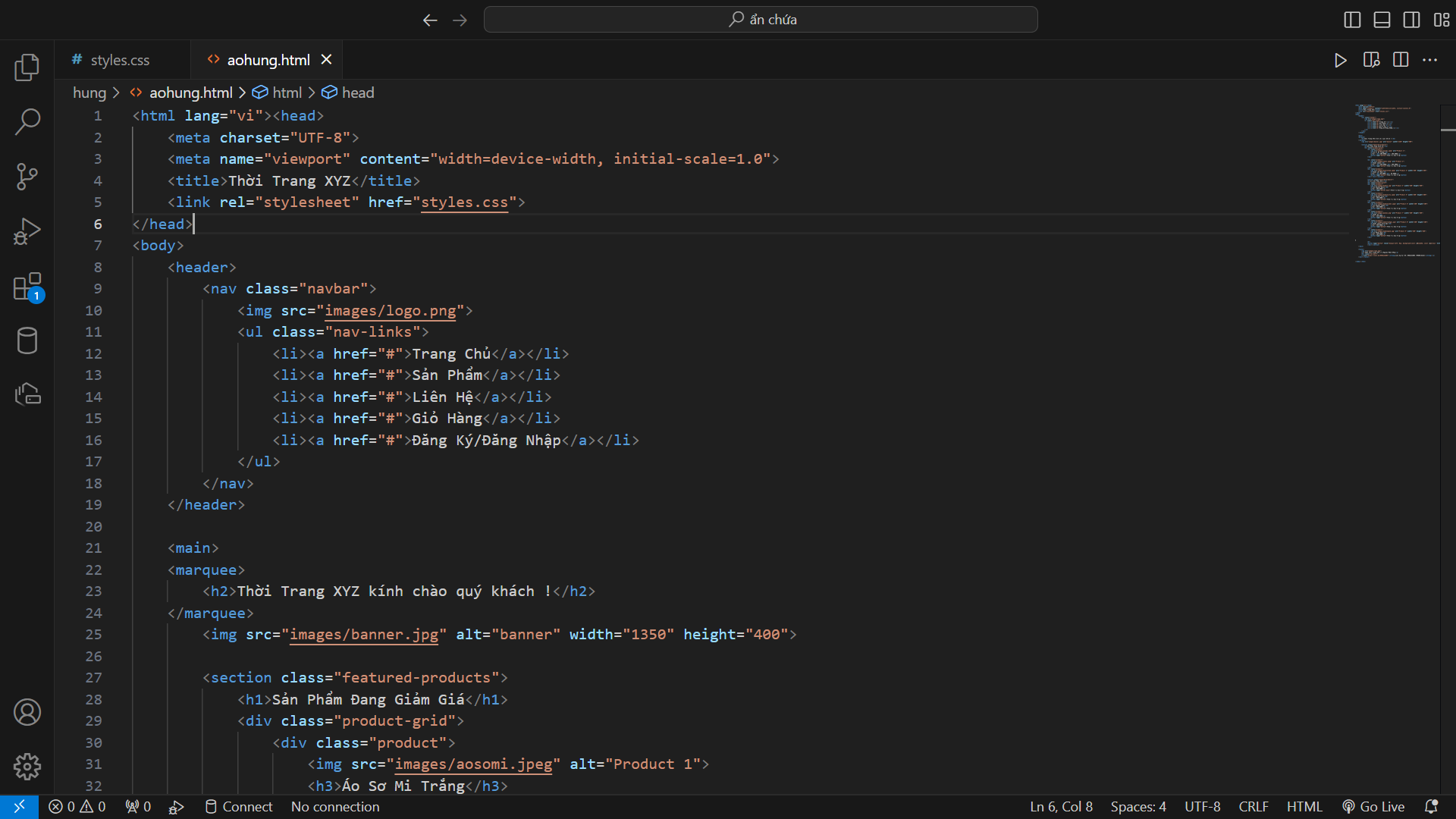
***Footer (Phần Cuối Trang):***

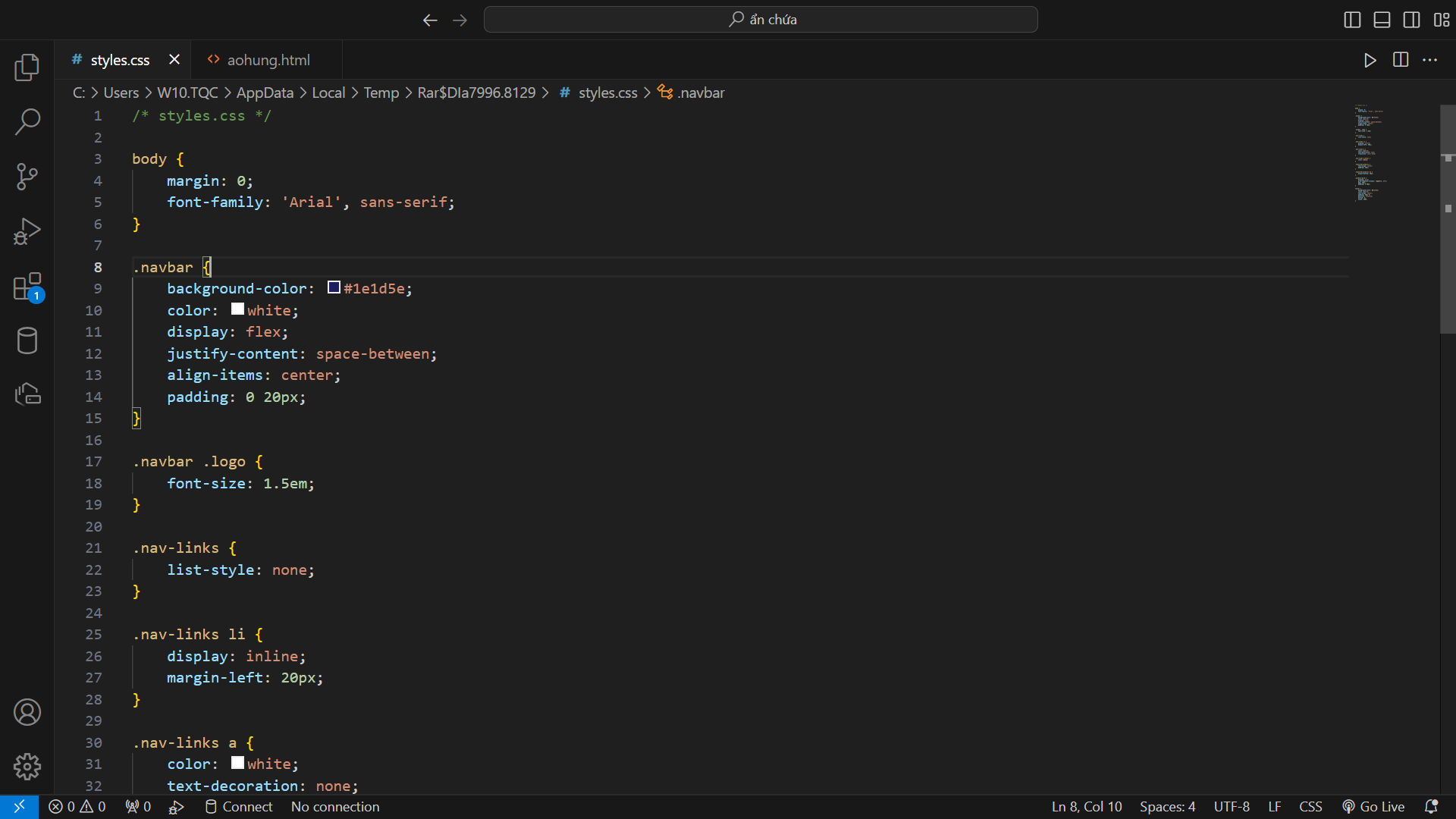
-Liên kết đến các trang quan trọng như về chúng tôi, chính sách bảo mật, điều khoản sử dụng và trợ giúp.

-Thông tin liên hệ như địa chỉ email, số điện thoại và địa chỉ văn phòng.

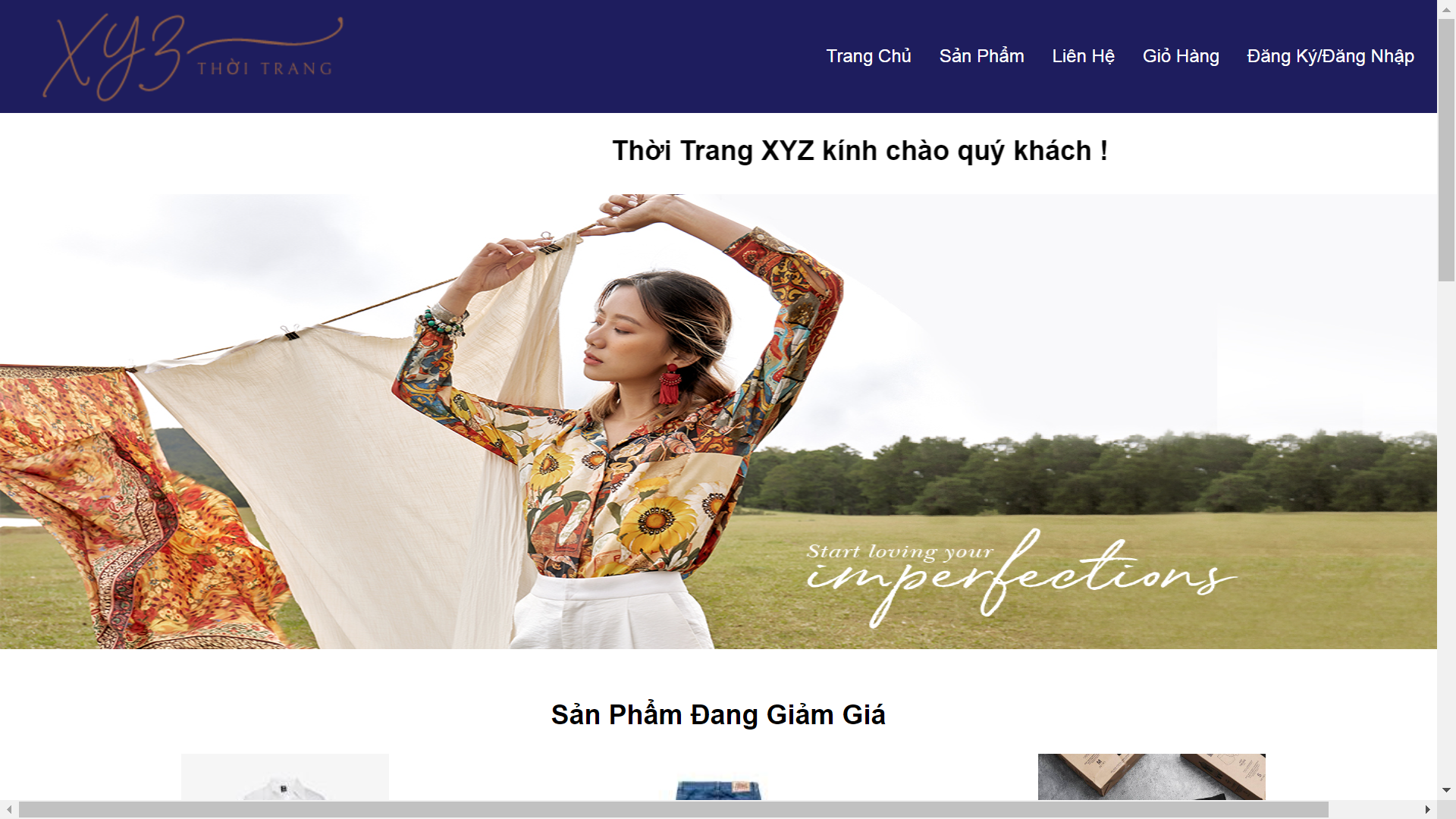


## **4.3. Thiết kế trang Web bằng HTML và CSS**





## **4.4. Kết quả đạt được**



Kết luận chương 4

-Chương 4 cung cấp cái nhìn tổng quan về quy trình thiết kế một trang web bán quần áo từ đầu đến cuối. Bằng cách sử dụng HTML và CSS, chúng tôi đã tạo ra một giao diện đơn giản nhưng hiệu quả cho phép người dùng dễ dàng duyệt qua sản phẩm, tìm thông tin và thực hiện các giao dịch mua sắm.

-Qua chương này, chúng ta đã: Sử dụng HTML để xây dựng cấu trúc cơ bản của một trang web, bao gồm header, banner, sản phẩm nổi bật và footer.

- Thiết kế giao diện trực quan, thân thiện với người dùng bằng CSS, từ việc chọn màu sắc, font chữ cho đến sắp xếp các thành phần trên trang.

-Tích hợp các chức năng cơ bản như thanh điều hướng, tìm kiếm sản phẩm, liên kết tới các trang thông tin khác.

-Nhưng việc xây dựng một trang web không dừng lại ở đó. Bạn có thể tiếp tục mở rộng và phát triển trang web của mình bằng cách thêm các tính năng mới như chức năng đăng nhập, giỏ hàng, thanh toán trực tuyến, đánh giá sản phẩm, v.v. để cải thiện trải nghiệm mua hàng. Khách hàng mua sắm.

# **KẾT LUẬN**

***\* Ưu điểm của thiết kế website bán quần áo:***

***-***Thuận tiện cho khách hàng: Thiết kế website bán quần áo mang đến sự tiện lợi cho khách hàng, có thể dễ dàng duyệt tìm sản phẩm, tìm kiếm thông tin và thực hiện giao dịch mua sắm mọi lúc mọi nơi.

-Độ phủ rộng: Một website có thể tiếp cận lượng lớn người dùng trên toàn thế giới, thay vì chỉ giới hạn ở một khu vực cụ thể như cửa hàng truyền thống, giúp tăng cơ hội bán hàng và phát triển kinh doanh.

-Tính năng tùy chỉnh và linh hoạt: Thiết kế web cho phép tích hợp các tính năng tùy chỉnh như tìm kiếm sản phẩm, xếp hạng và đánh giá, cũng như gửi thông báo khuyến mại đến khách hàng, tạo trải nghiệm mua sắm được cá nhân hóa giúp đơn giản hóa và nâng cao tương tác với khách hàng.

**\* Nhược điểm của thiết kế website bán quần áo:**

Cạnh tranh khốc liệt: Thị trường bán giày trực tuyến ngày càng cạnh tranh với nhiều cửa hàng trực tuyến khác nhau mọc lên. Điều này tạo ra thách thức trong việc thu hút và giữ chân khách hàng.

Vấn đề bảo mật: Trong môi trường trực tuyến, việc bảo vệ thông tin cá nhân và tài khoản của khách hàng là một vấn đề quan trọng. Thiết kế web không an toàn có thể tạo ra nguy cơ mất thông tin cá nhân hoặc gian lận tài khoản.

Phải được duy trì và cập nhật thường xuyên: Việc duy trì và cập nhật trang web để đáp ứng nhu cầu thị trường và nâng cao trải nghiệm người dùng đòi hỏi sự đầu tư về thời gian và nguồn lực.

**\*Hướng phát triển thiết kế web bán hàng quần áo:**

Tối ưu hóa trải nghiệm người dùng: Nâng cao trải nghiệm người dùng bằng cách cải thiện giao diện, tối ưu tốc độ tải trang và tăng tương tác với khách hàng thông qua các tính năng mới.

Mở rộng dịch vụ và sản phẩm: Mở rộng danh mục sản phẩm và cung cấp các dịch vụ mới, như chính sách giao hàng nhanh, chính sách hoàn trả linh hoạt và dịch vụ khách hàng tốt hơn.

Tăng cường an ninh, bảo mật: Đảm bảo website luôn giữ thông tin cá nhân, tài khoản của khách hàng an toàn, bảo mật, đồng thời cập nhật các biện pháp ngăn chặn gian lận và tấn công mạng.

Tích hợp với các kênh bán hàng khác: Tích hợp website bán quần áo của bạn với các kênh bán hàng khác như cửa hàng truyền thống, mạng xã hội, chợ trực tuyến để tăng hiệu quả bán hàng và phát triển doanh nghiệp của bạn.

# **DANH MỤC TÀI LIỆU THAM KHẢO**

1. Nguyễn Hồng Sơn (2007), *Giáo trình hệ thống Mạng máy tính CCNA* (Semester 1), NXB Lao động xã hội.
2. Phạm Quốc Hùng (2017), *Đề cương bài giảng Mạng máy tính*, Đại học SPKT Hưng Yên.
3. James F. Kurose and Keith W. Ross (2013), *Computer Networking: A top-down approach sixth Edition*, Pearson Education.