**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 WEB BÁN THIẾT BỊ Y TẾ**

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

**Sinh viên thực hiện:**  **PHÙNG VĂN ĐẠT**

**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 WEB BÁN THIẾT BỊ Y TẾ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
|  | 1771020146 | PHÙNG VĂN ĐẠT | 21/09/2005 |  |  |

### 

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

**Hà Nội, 2024**

**LỜI NÓI ĐẦU**

**Thiết kế web** là một lĩnh vực đầy sáng tạo và năng động, nơi nghệ thuật và kỹ thuật hòa quyện để tạo ra những trải nghiệm trực tuyến độc đáo. Là một nhà thiết kế web, tôi luôn say mê với việc biến những ý tưởng thành hiện thực, mang đến cho người dùng những trang web đẹp mắt, hiệu quả và dễ sử dụng.

**MỤC LỤC**

[**CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM** 6](#_Toc160701018)

[**1.1.** **Các khái niệm cơ bản** 6](#_Toc160701019)

[**1.2. Phân biệt Web tĩnh, Web động** 6](#_Toc160701020)

[**1.3. Một số thuật ngữ (Hosting, Tên miền-Domain Name-Web Server Name, HomePage,** 7](#_Toc160701021)

[**Web Site, WebServer, URL-Uniform Resource Location, Browser, …)** 7](#_Toc160701022)

[**1.4. Một số công cụ dùng thiết kế Web (Visual Studio code, Adobe Dreamweaver, …)** 8](#_Toc160701023)

[**2.1. Tổng quan về HTML** 11](#_Toc160701024)

[**2.3. Các thẻ HTML thông dụng** 13](#_Toc160701025)

[2.4. Các thẻ tạo biểu mẫu (form) 14](#_Toc160701026)

[**2.5. Một số thẻ HTML đặc biệt** 16](#_Toc160701027)

[**2.6. HTML5** 17](#_Toc160701028)

[**3.1. CSS là gì?** 21](#_Toc160701029)

[**3.2. Cú pháp CSS** 22](#_Toc160701030)

[**3.3. Áp dụng CSS vào trang HTML** 24](#_Toc160701031)

[**3.4. Selectors** 26](#_Toc160701032)

[3.4.1. Universal selector 28](#_Toc160701033)

[3.4.2. Type selector 29](#_Toc160701034)

[3.4.3. Identity selector 30](#_Toc160701035)

[3.4.4. Class selector 30](#_Toc160701036)

[3.4.6. Child selector 31](#_Toc160701037)

[3.4.7. Adjacent sibling selector 32](#_Toc160701038)

[3.4.8. Attribute selector 33](#_Toc160701039)

[3.4.9. Pseudo class selector 33](#_Toc160701040)

[3.4.10. Group selector 34](#_Toc160701041)

[**3.5. Đơn vị đo lường CSS** 35](#_Toc160701042)

[**3.6. Kế thừa thuộc tính** 36](#_Toc160701043)

[**3.7. Các nhóm thuộc tính trong CSS** 37](#_Toc160701044)

[**3.8. Float & Clear** 39](#_Toc160701045)

[**3.9. Flex** 40](#_Toc160701046)

[**3.10. Grid** 41](#_Toc160701047)

[**3.11. CSS3** 42](#_Toc160701048)

[**3.12. SCSS** 44](#_Toc160701049)

[**3.13. SASS** 45](#_Toc160701050)

[**4.1.** **Ý tưởng của Website** 48](#_Toc160701051)

[**4.2. Xây dựng bố cục của trang Web** 48](#_Toc160701052)

[**4.4**. **Kết quả đạt được** 52](#_Toc160701053)

[Kết luận chương 4 53](#_Toc160701054)

[**KẾT LUẬN** 54](#_Toc160701055)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 55](#_Toc160701056)

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

## **Các khái niệm cơ bản**

Thiết kế web là quá trình tạo ra các trang web. Nó bao gồm nhiều khía cạnh khác nhau, bao gồm bố cục trang, đồ họa, kiểu chữ và khả năng sử dụng. Thiết kế web tốt sẽ vừa đẹp mắt vừa dễ sử dụng.

Triển khai phần mềm là quá trình đưa phần mềm vào sử dụng. Nó bao gồm nhiều bước khác nhau, chẳng hạn như cài đặt phần mềm, cấu hình phần mềm và đào tạo người dùng.

Các loại triển khai phần mềm

Có hai loại triển khai phần mềm chính:

**Triển khai tại chỗ:** Phần mềm được cài đặt trên máy tính hoặc máy chủ của tổ chức.

**Triển khai phần mềm dựa trên đám mây:** Phần mềm được lưu trữ trên máy chủ của bên cung cấp dịch vụ và được truy cập thông qua internet.

Triển khai phần mềm dựa trên đám mây

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

Web tĩnh (Static Web) là website có nội dung được lưu trữ dưới dạng các file HTML, CSS và JavaScript cố định. Khi người dùng truy cập, web server sẽ gửi trực tiếp các file này đến trình duyệt của họ.

Web động (Dynamic Web) là website có nội dung được tạo ra dựa trên yêu cầu của người dùng. Khi người dùng truy cập, web server sẽ xử lý các yêu cầu này bằng các ngôn ngữ lập trình như PHP, Python, Ruby, v.v. và truy xuất dữ liệu từ cơ sở dữ liệu để tạo ra trang web tương ứng.

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

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

**Hosting:** Dịch vụ cho thuê chỗ lưu trữ website của bạn trên máy chủ (server) được kết nối internet 24/7. Giống như thuê đất để xây nhà, hosting cung cấp nền tảng để đặt website và hoạt động trực tuyến.

**Tên miền (Domain Name):** Tên gọi độc nhất dùng để truy cập website, ví dụ như [đã xoá URL không hợp lệ] hoặc vnexpress.net. Giống như tên đường dẫn đến nhà bạn, tên miền giúp người dùng dễ dàng tìm thấy website.

**Web Server Name:** Tên máy chủ chứa website, thường trùng với tên miền nhưng đôi khi sử dụng tên riêng biệt cho mục đích bảo mật.

**Homepage:** Trang chủ, là trang mặc định xuất hiện đầu tiên khi truy cập website, giống như sảnh chính của ngôi nhà.

**Website:** Tổng thể các trang web được liên kết với nhau dưới một tên miền chung, đại diện cho tổ chức, cá nhân, doanh nghiệp, v.v.

**Web Server:** Phần mềm máy tính chạy trên máy chủ, xử lý yêu cầu truy cập website, gửi nội dung web đến trình duyệt. Giống như người quản lý nhà, web server đảm bảo mọi người đến website đều được phục vụ đúng cách.

**URL (Uniform Resource Locator):** Địa chỉ cụ thể của một tài nguyên trên internet, bao gồm tên miền, đường dẫn thư mục và tên file.

**Browser (Trình duyệt):** Phần mềm cho phép người dùng truy cập và tương tác với các website, ví dụ như Google Chrome, Firefox, Safari. Giống như phương tiện di chuyển đến ngôi nhà, trình duyệt giúp bạn xem nội dung website.

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

**1. Visual Studio Code:**

Là trình soạn thảo mã nguồn miễn phí và mã nguồn mở được phát triển bởi Microsoft.

Hỗ trợ nhiều ngôn ngữ lập trình và có nhiều tiện ích mở rộng giúp cho việc thiết kế web trở nên dễ dàng hơn.

Phù hợp cho cả người mới bắt đầu và người có kinh nghiệm.

**2. Adobe Dreamweaver:**

Là phần mềm thiết kế web chuyên nghiệp được phát triển bởi Adobe.

Cung cấp nhiều tính năng hỗ trợ cho việc thiết kế web, bao gồm:

Giao diện trực quan WYSIWYG (What You See Is What You Get) giúp dễ dàng chỉnh sửa giao diện web mà không cần biết mã.

Hỗ trợ nhiều ngôn ngữ lập trình web phổ biến như HTML, CSS, JavaScript.

Có nhiều công cụ hỗ trợ cho việc tối ưu hóa website cho SEO.

Phù hợp cho người có kinh nghiệm thiết kế web.

**Sublime Text:**

Là trình soạn thảo mã nguồn trả phí với nhiều tính năng mạnh mẽ.

Hỗ trợ nhiều ngôn ngữ lập trình và có nhiều tiện ích mở rộng giúp cho việc thiết kế web trở nên dễ dàng hơn.

Phù hợp cho người có kinh nghiệm thiết kế web.

**4. Notepad++:**

Là trình soạn thảo mã nguồn miễn phí và nhẹ.

Hỗ trợ nhiều ngôn ngữ lập trình và có một số tiện ích mở rộng.

Phù hợp cho người mới bắt đầu thiết kế web.

**5. WYSIWYG Web Editors:**

Là các phần mềm thiết kế web trực quan cho phép người dùng tạo website mà không cần biết mã.

Một số WYSIWYG Web Editors phổ biến bao gồm:

Wix

Squarespace

Weebly

Phù hợp cho người mới bắt đầu thiết kế web và muốn tạo website đơn giản.

Kết luận chương 1

Thiết kế web và triển khai hệ thống phần mềm là hai lĩnh vực có mối quan hệ mật thiết với nhau. Thiết kế web tạo ra giao diện người dùng trực quan và thu hút, trong khi triển khai hệ thống phần mềm cung cấp chức năng và tính năng cho website.

Thiết kế web ảnh hưởng trực tiếp đến trải nghiệm người dùng (UX) của website. Một website được thiết kế tốt sẽ dễ sử dụng, thu hút và mang lại trải nghiệm tích cực cho người dùng.

Triển khai hệ thống phần mềm quyết định chức năng và tính năng của website. Hệ thống phần mềm cần được thiết kế và triển khai cẩn thận để đáp ứng nhu cầu của người dùng và mục tiêu kinh doanh của website.

Thiết kế web và triển khai hệ thống phần mềm cần được phối hợp chặt chẽ với nhau để tạo ra website thành công. Hai lĩnh vực này cần làm việc cùng nhau để đảm bảo rằng website có giao diện đẹp mắt, chức năng tốt và đáp ứng nhu cầu của người dùng.

**CHƯƠNG II: HTML & HTML5**

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

## **2.1. Tổng quan về HTML**

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



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

**1. Phần khai báo (head):**

Nằm giữa thẻ <html> và </html>.

Chứa thông tin về trang web, không hiển thị trực tiếp trên trang.

Một số thẻ HTML phổ biến trong phần head:

<meta>: Cung cấp thông tin về trang web như charset, author, keywords,...

<title>: Định nghĩa tiêu đề trang web.

<link>: Liên kết đến các tập tin CSS và JavaScript.

<script>: Chèn mã JavaScript vào trang web.

**2. Phần nội dung (body):**

Nằm giữa thẻ <body> và </body>.

Chứa nội dung chính hiển thị trên trang web.

Một số thẻ HTML phổ biến trong phần body:

<p>: Định nghĩa đoạn văn bản.

<h1> đến <h6>: Định nghĩa heading (tiêu đề) từ cấp 1 đến cấp 6.

<img>: Chèn hình ảnh vào trang web.

<a>: Tạo liên kết đến một trang web khác.

<table>: Định nghĩa bảng.

<ul>: Định nghĩa danh sách không thứ tự.

<ol>: Định nghĩa danh sách có thứ tự.

## **2.3. Các thẻ HTML thông dụng**

1. Các thẻ xử lý văn bản:

Thẻ <p>: Định nghĩa đoạn văn bản.

Thẻ <h1> đến <h6>: Định nghĩa heading (tiêu đề) từ cấp 1 đến cấp 6.

Thẻ <br>: Tạo dòng mới.

Thẻ <pre>: Định nghĩa định dạng văn bản cố định.

Thẻ <strong>: Định nghĩa văn bản đậm.

Thẻ <em>: Định nghĩa văn bản in nghiêng.

Thẻ <mark>: Đánh dấu văn bản.

Thẻ <ins>: Chèn văn bản mới.

Thẻ <del>: Xóa văn bản.

2. Các thẻ tạo bảng:

Thẻ <table>: Định nghĩa bảng.

Thẻ <tr>: Định nghĩa hàng trong bảng.

Thẻ <td>: Định nghĩa ô trong bảng.

Thẻ <thead>: Định nghĩa phần đầu bảng.

Thẻ <tbody>: Định nghĩa phần thân bảng.

Thẻ <tfoot>: Định nghĩa phần chân bảng.

3. Thẻ liên kết:

Thẻ <a>: Tạo liên kết đến một trang web khác.

Thuộc tính href: Cung cấp địa chỉ URL của trang web liên kết.

Thuộc tính target: Xác định vị trí mở trang web liên kết.

4. Các thẻ đa phương tiện:

Thẻ <img>: Chèn hình ảnh vào trang web.

Thuộc tính src: Cung cấp địa chỉ URL của hình ảnh.

Thuộc tính alt: Cung cấp văn bản thay thế cho hình ảnh.

Thẻ <video>: Chèn video vào trang web.

Thuộc tính src: Cung cấp địa chỉ URL của video.

Thẻ <audio>: Chèn âm thanh vào trang web.

Thuộc tính src: Cung cấp địa chỉ URL của âm thanh.

5. Thẻ tạo khung

Thẻ <frameset>: Định nghĩa khung cho trang web.

Thẻ <frame>: Định nghĩa nội dung cho từng khung.

Thuộc tính src: Cung cấp địa chỉ URL của nội dung cho khung.

## 2.4. Các thẻ tạo biểu mẫu (form)

**Thẻ** <form>**:** Định nghĩa biểu mẫu.

**Thuộc tính**action**:** Cung cấp địa chỉ URL để xử lý dữ liệu biểu mẫu.

**Thuộc tính**method**:** Xác định phương thức gửi dữ liệu biểu mẫu.

**Thẻ** <input>**:** Tạo các trường nhập liệu trong biểu mẫu.

**Thuộc tính**type**:** Xác định loại trường nhập liệu.

text: Trường nhập văn bản.

password: Trường nhập mật khẩu.

radio: Nút radio.

checkbox: Hộp kiểm.

submit: Nút gửi.

reset: Nút đặt lại.

**Thuộc tính**name**:** Cung cấp tên cho trường nhập liệu.

**Thuộc tính**value**:** Cung cấp giá trị mặc định cho trường nhập liệu.

**Thẻ** <label>**:** Gắn nhãn cho các trường nhập liệu.

**Thuộc tính**for**:** Cung cấp ID của trường nhập liệu liên quan.

**Thẻ** <textarea>**:** Tạo trường nhập văn bản đa dòng.

**Thuộc tính**name**:** Cung cấp tên cho trường nhập liệu.

**Thuộc tính**rows**:** Xác định số lượng hàng của trường nhập liệu.

**Thuộc tính**cols**:** Xác định số lượng cột của trường nhập liệu.

**Thẻ** <select>**:** Tạo danh sách lựa chọn.

**Thuộc tính**name**:** Cung cấp tên cho danh sách lựa chọn.

**Thẻ**<option>**:** Định nghĩa các lựa chọn trong danh sách.

**Thuộc tính**value**:** Cung cấp giá trị cho lựa chọn.

**Thẻ** <button>**:** Tạo nút bấm.

**Thuộc tính**type**:** Xác định loại nút bấm.

submit: Nút gửi.

reset: Nút đặt lại.

**Thuộc tính**name**:** Cung cấp tên cho nút bấm.

**Thuộc tính**value**:** Cung cấp giá trị cho nút bấm.

## **2.5. Một số thẻ HTML đặc biệt**

DOCTYPE html> Đây là thẻ HTML đặc biệt xác định rằng tài liệu là tài liệu HTML.

<html> Đây là thẻ HTML đặc biệt xác định đầu trang HTML.

<head> Đây là thẻ HTML đặc biệt chứa thông tin về trang HTML, chẳng hạn như tiêu đề và tác giả.

<title> Đây là thẻ HTML đặc biệt xác định tiêu đề của trang HTML.

<meta> Đây là thẻ HTML đặc biệt cung cấp thông tin bổ sung về trang HTML, chẳng hạn như mô tả và từ khóa.

<body> Đây là thẻ HTML đặc biệt xác định phần thân của trang HTML, chứa nội dung chính của trang.

<h1> Đây là thẻ HTML đặc biệt xác định tiêu đề cấp 1.

<h2> Đây là thẻ HTML đặc biệt xác định tiêu đề cấp 2.

<h3> Đây là thẻ HTML đặc biệt xác định tiêu đề cấp 3.

<p> Đây là thẻ HTML đặc biệt xác định một đoạn văn bản.

<br> Đây là thẻ HTML đặc biệt xác định một ngắt dòng.

<img> Đây là thẻ HTML đặc biệt chèn hình ảnh vào trang HTML.

<table> Đây là thẻ HTML đặc biệt xác định bảng.

<tr> Đây là thẻ HTML đặc biệt xác định một hàng trong bảng.

<td> Đây là thẻ HTML đặc biệt xác định một ô trong bảng.

<a> Đây là thẻ HTML đặc biệt xác định một liên kết.

<form> Đây là thẻ HTML đặc biệt xác định một biểu mẫu.

<input> Đây là thẻ HTML đặc biệt xác định một trường nhập trong biểu mẫu.

<button> Đây là thẻ HTML đặc biệt xác định một nút trên biểu mẫu.

</body> Đây là thẻ HTML đặc biệt xác định phần cuối của phần thân trang HTML.

</html> Đây là thẻ HTML đặc biệt xác định phần cuối của trang HTML.

## **2.6. HTML5**

HTML5 là phiên bản thứ năm của ngôn ngữ đánh dấu siêu văn bản HTML được sử dụng để tạo trang web. Nó được phát triển bởi W3C (World Wide Web Consortium) và được hỗ trợ bởi hầu hết các trình duyệt web hiện đại.

HTML5 cung cấp nhiều tính năng mới và cải tiến so với các phiên bản HTML trước đây, bao gồm:

**Đa phương tiện:** HTML5 cung cấp khả năng nhúng video, âm thanh và đồ họa trực tiếp vào trang web mà không cần sử dụng trình cắm thêm.

**Lưu trữ web:** HTML5 cung cấp API lưu trữ web mới cho phép các trang web lưu trữ dữ liệu cục bộ trên máy tính của người dùng.

**Kết nối web:** HTML5 cung cấp các API mới cho phép các trang web kết nối với các thiết bị và dịch vụ khác, ví dụ như API Geolocation và API WebRTC.

Cú pháp HTML5

Cú pháp HTML5 dựa trên XML, có nghĩa là nó tuân theo các quy tắc cú pháp nghiêm ngặt hơn so với các phiên bản HTML trước đây. Dưới đây là một số quy tắc cú pháp cơ bản của HTML5:

Tất cả các thẻ phải được đóng,các thuộc tính thẻ phải được đặt trong dấu ngoặc kép,các giá trị thuộc tính phải được đặt trong dấu ngoặc kép hoặc đơn,các thẻ phải được viết bằng chữ thường.

Khai báo DOCTYPE trong HTML5

Khai báo DOCTYPE là dòng đầu tiên trong tài liệu HTML5. Nó cho trình duyệt biết phiên bản HTML nào đang được sử dụng. Dưới đây là khai báo DOCTYPE cho HTML5:



Thẻ ngữ nghĩa

Thẻ ngữ nghĩa được sử dụng để xác định rõ ràng ý nghĩa của nội dung trang web. HTML5 giới thiệu các thẻ ngữ nghĩa mới sau:

<article>: Dùng để xác định một bài báo hoặc một phần nội dung độc lập.

<section>: Dùng để xác định một phần của trang web.

<header>: Dùng để xác định phần đầu trang web.

<footer>: Dùng để xác định phần chân trang web.

<aside>: Dùng để xác định nội dung bổ sung, chẳng hạn như thanh bên.

<nav>: Dùng để xác định hệ thống điều hướng của trang web.

Webform

Webform là một tập hợp các thẻ HTML được sử dụng để tạo các biểu mẫu trên trang web. Các thẻ Webform phổ biến bao gồm:

<input>: Dùng để tạo các trường nhập liệu, ví dụ anhư hộp văn bản, hộp kiểm, nút radio, v.v.

<select>: Dùng để tạo menu thả xuống.

<textarea>: Dùng để tạo vùng nhập văn bản đa dòng.

<button>: Dùng để tạo nút.

Dưới đây là ví dụ về một Webform đơn giản:



Kết luận chương 2

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu siêu văn bản, đóng vai trò như bộ khung cơ bản để xây dựng trang web. Nó sử dụng các thẻ để định nghĩa cấu trúc và nội dung của trang, bao gồm văn bản, hình ảnh, video, liên kết,...

HTML5 là phiên bản mới nhất của HTML, được phát hành vào năm 2014. Nó mang đến nhiều cải tiến và tính năng mới so với các phiên bản trước, giúp cho việc phát triển web trở nên dễ dàng và hiệu quả hơn.

**CHƯƠNG III: CSS và CSS3**

 (Cascading Style Sheets)

## **3.1. CSS là gì?**

CSS là viết tắt của Cascading Style Sheets, là một ngôn ngữ lập trình được sử dụng để định dạng và tạo kiểu cho các trang web. Nó hoạt động song song với HTML, ngôn ngữ dùng để tạo cấu trúc và nội dung cho trang web.

Vai trò của CSS:

Kiểm soát giao diện: CSS cho phép bạn điều chỉnh màu sắc, phông chữ, kích thước, bố cục, hiệu ứng, v.v. của các yếu tố HTML trên trang web.

Tạo sự nhất quán: CSS giúp bạn tạo ra giao diện đồng nhất cho toàn bộ trang web hoặc cho các phần cụ thể của trang web.

Tăng khả năng truy cập: CSS giúp cải thiện khả năng truy cập của trang web bằng cách cho phép bạn điều chỉnh kích thước phông chữ, khoảng cách dòng và các yếu tố khác để dễ đọc hơn.

Giảm dung lượng tải trang: Việc sử dụng CSS giúp giảm dung lượng tải trang vì bạn có thể định dạng nhiều yếu tố HTML cùng một lúc.

## **3.2. Cú pháp CSS**

1. Bộ chọn (Selector)

Bộ chọn xác định các phần tử HTML mà bạn muốn định dạng. Có nhiều loại bộ chọn khác nhau, bao gồm:

Bộ chọn thẻ: Chọn các phần tử HTML dựa trên tên thẻ của chúng. Ví dụ: h1, p, div.

Bộ chọn lớp: Chọn các phần tử HTML dựa trên lớp được gán cho chúng. Ví dụ: .header, .content, .footer.

**Bộ chọn ID:** Chọn các phần tử HTML dựa trên ID được gán cho chúng. Ví dụ: #main, #sidebar, #footer.

**Bộ chọn con:** Chọn các phần tử HTML con của một phần tử khác. Ví dụ: div p, ul li.

**Bộ chọn chung:** Chọn tất cả các phần tử HTML trên trang.

2. Thuộc tính (Property)

Thuộc tính là các đặc điểm mà bạn muốn định dạng cho các phần tử HTML được chọn. Có rất nhiều thuộc tính CSS khác nhau, bao gồm:

Màu sắc: color

Phông chữ: font-family, font-size, font-weight

Kích thước: width, height

Vị trí: margin, padding, text-align

Hiệu ứng: border, background, box-shadow

3. Giá trị (Value)

Giá trị là các giá trị cụ thể mà bạn gán cho các thuộc tính. Ví dụ:

color: red;

font-family: Arial, sans-serif;

font-size: 16px;

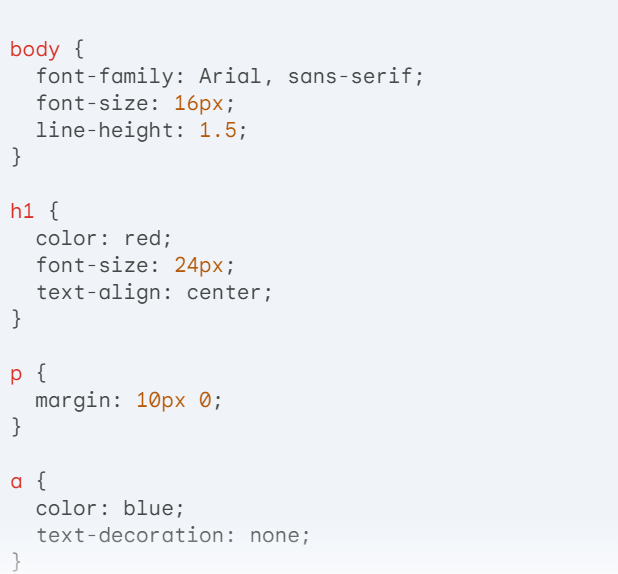
width: 500px;

margin: 10px;

Cấu trúc:

Một quy tắc CSS bao gồm bộ chọn, thuộc tính và giá trị được viết trong ngoặc vuông và được ngăn cách bởi dấu hai chấm.





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

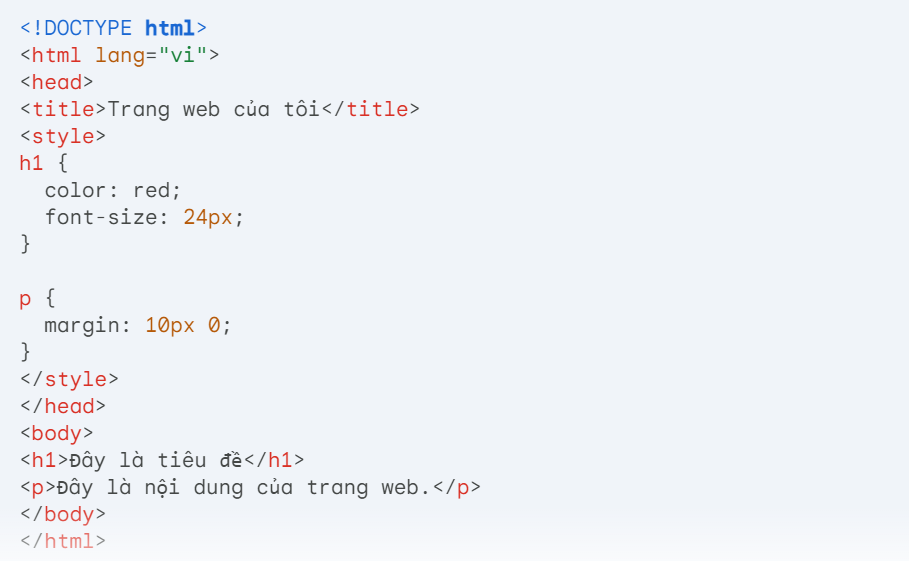
1. CSS nội tuyến (Inline CSS)

Với cách này, bạn thêm mã CSS trực tiếp vào thẻ HTML. Đây là cách đơn giản nhất để áp dụng CSS, nhưng nó không phù hợp cho việc sử dụng nhiều lần hoặc cho các trang web lớn.

\

2. CSS nhúng (Internal CSS)

Với cách này, bạn thêm mã CSS vào phần <head> của trang HTML. Đây là cách tốt để áp dụng CSS cho một trang HTML duy nhất.



3. CSS liên kết (External CSS)

Với cách này, bạn lưu trữ mã CSS trong một tệp riêng biệt với phần mở rộng .css và sau đó liên kết tệp đó với trang HTML. Đây là cách tốt nhất để áp dụng CSS cho nhiều trang web.





Lựa chọn phương pháp phù hợp:

Lựa chọn phương pháp áp dụng CSS phù hợp phụ thuộc vào nhu cầu cụ thể của bạn.

CSS nội tuyến: Sử dụng cho các thay đổi nhỏ và đơn giản.

CSS nhúng: Sử dụng cho các thay đổi áp dụng cho một trang HTML duy nhất.

CSS liên kết: Sử dụng cho các thay đổi áp dụng cho nhiều trang HTML.

Lời khuyên:

Nên sử dụng CSS liên kết để tách biệt nội dung HTML khỏi định dạng CSS.

Nên đặt tên tệp CSS theo cách mô tả nội dung của nó.

Nên sử dụng các công cụ dành cho nhà phát triển web để giúp bạn viết và kiểm tra mã CSS của bạn.

## **3.4. Selectors**

Bộ chọn CSS là mẫu để chọn phần tử HTML mà bạn muốn định dạng. Có nhiều loại bộ chọn khác nhau, bao gồm:

1. Bộ chọn thẻ (Type Selectors)

Chọn các phần tử HTML dựa trên tên thẻ của chúng. Ví dụ:

h1: Chọn tất cả các thẻ <h1>.

p: Chọn tất cả các thẻ <p>.

div: Chọn tất cả các thẻ <div>.

2. Bộ chọn lớp (Class Selectors)

Chọn các phần tử HTML dựa trên lớp được gán cho chúng. Ví dụ:

.header: Chọn tất cả các phần tử HTML có lớp header.

.content: Chọn tất cả các phần tử HTML có lớp content.

.footer: Chọn tất cả các phần tử HTML có lớp footer.

3. Bộ chọn ID (ID Selectors)

Chọn các phần tử HTML dựa trên ID được gán cho chúng. Ví dụ:

#main: Chọn phần tử HTML có ID main.

#sidebar: Chọn phần tử HTML có ID sidebar.

#footer: Chọn phần tử HTML có ID footer.

4. Bộ chọn con (Descendant Selectors)

Chọn các phần tử HTML con của một phần tử khác. Ví dụ:

div p: Chọn tất cả các thẻ <p> là con của thẻ <div>.

ul li: Chọn tất cả các thẻ <li> là con của thẻ <ul>.

table td: Chọn tất cả các thẻ <td> là con của thẻ <table>.

5. Bộ chọn chung (Universal Selector)

Chọn tất cả các phần tử HTML trên trang. Ví dụ:

\*: Chọn tất cả các phần tử HTML.

6. Bộ chọn kết hợp (Combinators)

Kết hợp các bộ chọn để chọn các phần tử HTML cụ thể hơn. Ví dụ:

h1.title: Chọn tất cả các thẻ <h1> có lớp title.

div p.content: Chọn tất cả các thẻ <p> có lớp content là con của thẻ <div>.

ul li:first-child: Chọn thẻ <li> đầu tiên là con của thẻ <ul>.

7. Bộ chọn thuộc tính (Attribute Selectors)

Chọn các phần tử HTML dựa trên thuộc tính của chúng. Ví dụ:

img[alt]: Chọn tất cả các thẻ <img> có thuộc tính alt.

a[href]: Chọn tất cả các thẻ <a> có thuộc tính href.

input[type="radio"]: Chọn tất cả các thẻ <input> có thuộc tính type bằng radio.

8. Bộ chọn giả (Pseudo-classes)

Chọn các phần tử HTML dựa trên trạng thái của chúng. Ví dụ:

a:hover: Chọn thẻ <a> khi con trỏ chuột di chuột qua.

li:active: Chọn thẻ <li> đang được kích hoạt.

input:focus: Chọn thẻ <input> đang được chọn.

9. Bộ chọn giả phần tử (Pseudo-elements)

Chọn các phần tử giả của một phần tử HTML. Ví dụ:

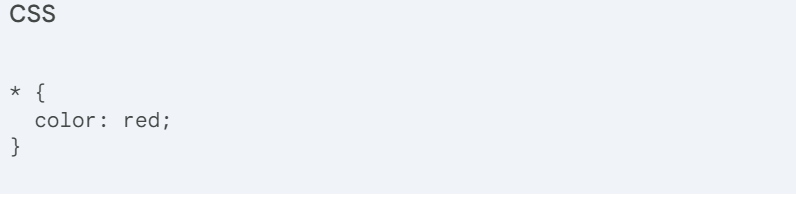
::first-letter: Chọn chữ cái đầu tiên của một phần tử.

::before: Chọn nội dung trước một phần tử.

::after: Chọn nội dung sau một phần tử.

### 3.4.1. Universal selector

**Universal selector**, hay còn gọi là **bộ chọn chung**, là một bộ chọn CSS dùng để chọn tất cả các phần tử HTML trên trang web. Nó được biểu thị bằng ký hiệu \*.



Lưu ý:

Universal selector nên được sử dụng một cách cẩn thận vì nó có thể ảnh hưởng đến tất cả các phần tử HTML trên trang web.

Nên sử dụng các bộ chọn cụ thể hơn để tránh ảnh hưởng đến các phần tử HTML không mong muốn.

Một số trường hợp sử dụng Universal selector:

Đặt lại kiểu dáng: Sử dụng universal selector để đặt lại tất cả các kiểu dáng được áp dụng cho các phần tử HTML.

Tạo hiệu ứng toàn trang: Sử dụng universal selector để tạo hiệu ứng ảnh hưởng đến tất cả các phần tử HTML trên trang web.

Chọn các phần tử không có lớp hoặc ID: Sử dụng universal selector để chọn các phần tử HTML không có lớp hoặc ID.

### 3.4.2. Type selector

Trong CSS, Type selector (bộ chọn theo loại) là một cách chọn các phần tử HTML dựa trên tên thẻ của chúng.

Đây là bộ chọn cơ bản và dễ sử dụng nhất, giúp bạn định dạng kiểu dáng cho nhiều phần tử cùng loại trên trang web một cách nhanh chóng.

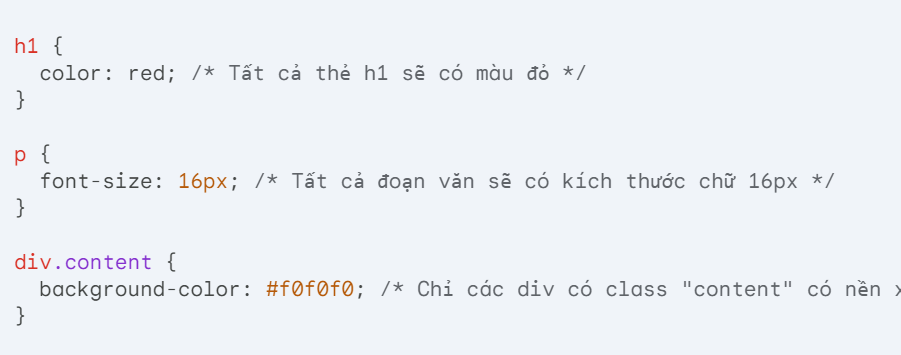
Cách sử dụng:

Chỉ cần viết tên thẻ bạn muốn chọn, không cần thêm bất kỳ ký hiệu đặc biệt nào. Ví dụ:

h1: Chọn tất cả các thẻ tiêu đề chính <h1>.

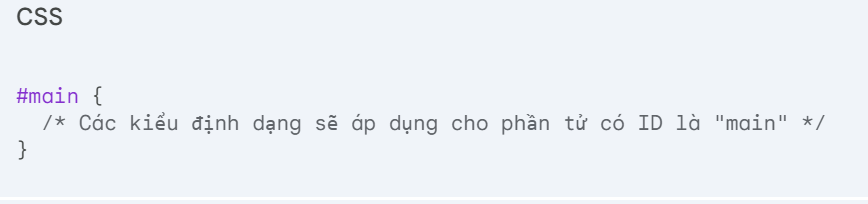
p: Chọn tất cả các đoạn văn <p>.

div: Chọn tất cả các vùng nội dung <div>.

****

### 3.4.3. Identity selector

Trong CSS, **Identity selector (bộ chọn ID)** là cách chọn một phần tử HTML duy nhất trên trang web dựa trên **ID** (định danh) được gán cho nó. ID phải là duy nhất trong toàn bộ tài liệu HTML và thường dùng để chọn các thành phần chính của trang.

****

Ưu điểm:

Chính xác cao, chỉ chọn được một phần tử duy nhất.

Dễ dàng định dạng các thành phần quan trọng của trang.

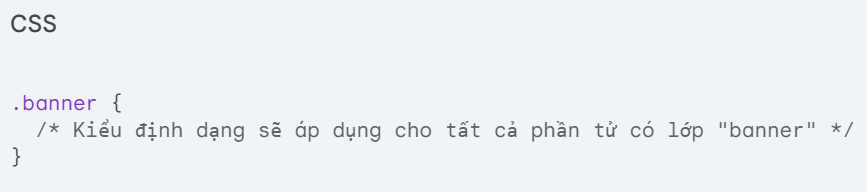
Thường dùng trong JavaScript để tương tác với các phần tử trên trang.

### 3.4.4. Class selector

Trong CSS, **Class selector (bộ chọn lớp)** là một cách chọn các phần tử HTML dựa trên **lớp (class)** được gán cho chúng. Lớp là một thuộc tính tùy chọn có thể gán cho nhiều phần tử khác nhau, giúp bạn định dạng kiểu dáng theo nhóm một cách linh hoạt.

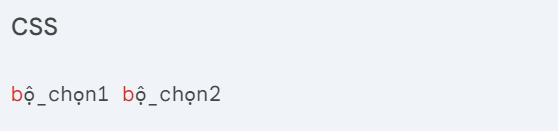
**Cách sử dụng:**

Để chọn các phần tử dựa trên lớp, hãy sử dụng dấu chấm (.) trước tên lớp.

****

Bộ chọn con cháu, được biểu thị bằng một khoảng trắng đơn, là một trong những bộ kết hợp cơ bản trong CSS. Nó cho phép bạn chọn các phần tử được lồng nhúng bất kỳ nơi nào bên trong một phần tử khác, bất kể độ sâu của chúng trong cây DOM.

**Cú pháp:**



bộ\_chọn1: Có thể là bất kỳ bộ chọn CSS hợp lệ nào (ví dụ: tên thẻ, tên lớp, ID).

bộ\_chọn2: Chọn bất kỳ phần tử nào được lồng nhúng bên trong bộ\_chọn1.

### 3.4.6. Child selector

Chọn con là một khái niệm quan trọng trong CSS, cho phép bạn chọn các phần tử dựa trên mối quan hệ của chúng với các phần tử khác trong DOM. Có hai khía cạnh chính cần nắm rõ:

1. Trình kết hợp con (>):

Ký hiệu này được sử dụng để chọn các phần tử là con trực tiếp của một phần tử khác. Ví dụ: div > p sẽ chọn tất cả các phần tử <p> là con trực tiếp của các phần tử <div>.

2. Tả bán giả :nth-child():

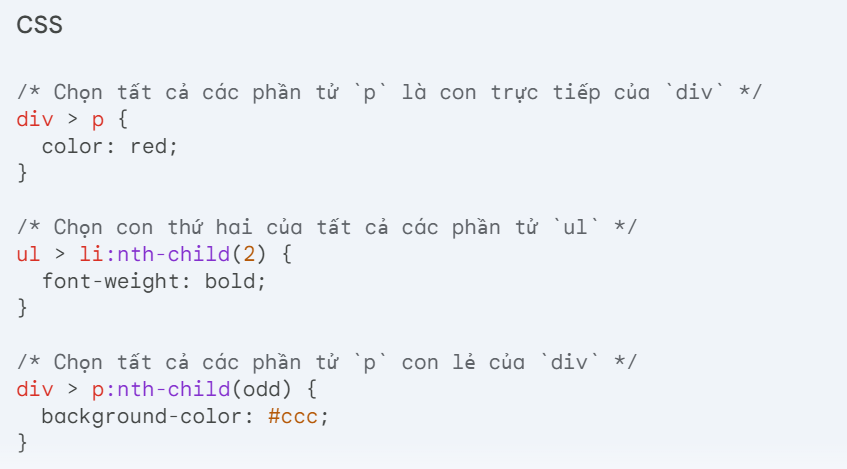
Tả bán giả này cho phép bạn chọn các phần tử dựa trên vị trí của chúng trong danh sách con của phần tử cha. Bạn có thể chỉ định vị trí chính xác (ví dụ: :nth-child(3) cho con thứ ba), sử dụng từ khóa như odd hoặc even, hoặc tạo công thức như an + b để phù hợp với các mẫu phức tạp hơn.

Lưu ý:

Trình kết hợp con khắt khe hơn trình kết hợp hậu duệ (), vốn chọn bất kỳ phần tử nào là hậu duệ của phần tử được chỉ định, bất kể mức độ sâu.

Khi sử dụng :nth-child(), hãy nhớ rằng chỉ mục của con đầu tiên là 1.

Cũng có các biến thể như :nth-of-type() chỉ xem xét các phần tử cùng loại (tên thẻ) trong phần tử cha.



### 3.4.7. Adjacent sibling selector

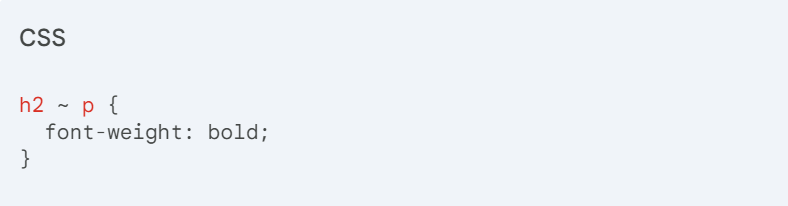
1. Trình kết hợp anh chị em liền kề (+):

Ký hiệu + cho phép bạn chọn phần tử anh chị em **liền kề ngay sau** phần tử được chỉ định. Cả hai phần tử phải cùng cấp và có cùng phần tử cha.



2. Trình chọn anh chị em tổng quát (~):

Ký hiệu ~ cho phép bạn chọn tất cả các phần tử anh chị em của phần tử được chỉ định, miễn là chúng cùng cấp và có cùng phần tử cha.



### 3.4.8. Attribute selector

Chọn phần tử dựa trên thuộc tính (Attribute selector) là một công cụ mạnh mẽ trong CSS cho phép bạn chọn các phần tử HTML dựa trên giá trị của các thuộc tính HTML. Dưới đây là một số điểm chính cần nắm rõ:

Cấu trúc:



### 3.4.9. Pseudo class selector

Bộ chọn lớp giả là một loại bộ chọn đặc biệt trong CSS cho phép bạn chọn các phần tử dựa trên trạng thái hoặc điều kiện hiện tại của chúng. Chúng không phải là một phần của HTML của phần tử mà được thêm vào bộ chọn phần tử bằng cách sử dụng dấu hai chấm (:).

Dưới đây là một số ví dụ phổ biến về bộ chọn lớp giả:

:hover: Bộ chọn lớp giả này chọn một phần tử khi người dùng di chuột qua nó. Nó thường được sử dụng để thay đổi giao diện của các nút, liên kết và các phần tử tương tác khác khi di chuột qua.

:active: Bộ chọn lớp giả này chọn một phần tử khi người dùng nhấp vào nó và giữ nút chuột trái. Nó thường được sử dụng để định kiểu các nút và các phần tử biểu mẫu khác trong khi chúng được nhấp.

:visited: Bộ chọn lớp giả này chọn các liên kết mà người dùng đã truy cập trước đây. Nó có thể được sử dụng để định kiểu các liên kết đã truy cập khác với các liên kết chưa truy cập.

:first-child: Bộ chọn lớp giả này chọn phần tử con đầu tiên của phần tử cha của nó. Nó có thể được sử dụng để định kiểu đoạn văn đầu tiên trong một phần khác với các đoạn văn khác, ví dụ.

:last-child: Bộ chọn lớp giả này chọn phần tử con cuối cùng của phần tử cha của nó. Nó có thể được sử dụng để định kiểu đoạn văn cuối cùng trong một phần khác với các đoạn văn khác, ví dụ.

Bộ chọn lớp giả có thể là một công cụ mạnh mẽ để thêm tính tương tác và hiệu ứng trực quan vào trang web của bạn. Chúng có thể được sử dụng để tạo các hiệu ứng như menu di chuột, chú thích công cụ, v.v.

### 3.4.10. Group selector

Trong CSS, trình chọn nhóm (group selector) là một cách hữu ích để chọn nhiều phần tử khác nhau và áp dụng cùng một style cho chúng.

Dấu phẩy (,):

Bạn có thể liệt kê các bộ chọn khác nhau, ngăn cách nhau bằng dấu phẩy, để chọn một nhóm phần tử. Mọi style được áp dụng sau dấu phẩy sẽ được áp dụng cho tất cả các phần tử được chọn trong nhóm.



## **3.5. Đơn vị đo lường CSS**

CSS sử dụng nhiều đơn vị đo lường khác nhau để xác định kích thước, vị trí, và các thuộc tính khác của các phần tử trên trang web. Việc lựa chọn đơn vị phù hợp là rất quan trọng để đảm bảo bố cục trang web hiển thị chính xác và phù hợp trên các thiết bị khác nhau.

Dưới đây là một số loại đơn vị đo lường phổ biến trong CSS:

1. Đơn vị tuyệt đối:

px (pixel): Đây là đơn vị được sử dụng phổ biến nhất, đại diện cho một điểm ảnh trên màn hình. Tuy nhiên, kích thước của pixel có thể khác nhau trên các thiết bị khác nhau, dẫn đến việc bố cục trang web có thể bị thay đổi.

pt (point): Đơn vị này tương đương với 1/72 inch, thường được sử dụng cho typography.

pc (pica): Đơn vị này tương đương với 12pt, cũng thường được sử dụng cho typography.

cm (centimeter): Đơn vị này dựa trên hệ thống đo lường quốc tế, tương đương với 1/100 mét.

mm (millimeter): Đơn vị này cũng dựa trên hệ thống đo lường quốc tế, tương đương với 1/1000 mét.

2. Đơn vị tương đối:

% (percent): Đơn vị này biểu thị tỷ lệ phần trăm so với một giá trị khác. Ví dụ, width: 50% có nghĩa là chiều rộng của phần tử sẽ bằng 50% chiều rộng của phần tử cha.

em: Đơn vị này tương đương với kích thước phông chữ hiện tại của phần tử. Ví dụ, font-size: 16px; margin: 1em; có nghĩa là lề của phần tử sẽ bằng 16px (kích thước phông chữ)

rem: Đơn vị này tương đương với kích thước phông chữ gốc của phần tử HTML. Ví dụ, font-size: 16px; margin: 1rem; có nghĩa là lề của phần tử sẽ bằng 16px (kích thước phông chữ gốc)

3. Đơn vị viewport:

vw (viewport width): Đơn vị này tương đương với 1% chiều rộng của khung nhìn (viewport). Ví dụ, width: 10vw; có nghĩa là chiều rộng của phần tử sẽ bằng 10% chiều rộng của khung nhìn.

vh (viewport height): Đơn vị này tương đương với 1% chiều cao của khung nhìn. Ví dụ, height: 10vh; có nghĩa là chiều cao của phần tử sẽ bằng 10% chiều cao của khung nhìn.

Ngoài ra, CSS còn có một số đơn vị khác như vmin (viewport minimum) và vmax (viewport maximum) để xác định kích thước dựa trên giá trị nhỏ nhất hoặc lớn nhất của khung nhìn.

## **3.6. Kế thừa thuộc tính**

Kế thừa thuộc tính là một tính năng quan trọng trong CSS cho phép các thuộc tính được truyền từ phần tử cha sang phần tử con. Nhờ vậy, bạn có thể định kiểu cho nhiều phần tử cùng lúc một cách dễ dàng và hiệu quả, giảm thiểu việc lặp lại code.

Cơ chế hoạt động:

Mọi thuộc tính CSS đều có khả năng kế thừa, trừ các thuộc tính border, margin, padding, float, position, clear, và display.

Khi một phần tử con không được định nghĩa một thuộc tính nào đó, nó sẽ kế thừa giá trị của thuộc tính đó từ phần tử cha.

Nếu phần tử cha cũng không định nghĩa thuộc tính đó, trình duyệt sẽ tiếp tục tìm kiếm giá trị kế thừa từ các phần tử cha cao hơn cho đến khi tìm thấy giá trị được định nghĩa.

Giá trị kế thừa có thể được ghi đè bằng cách định nghĩa trực tiếp thuộc tính cho phần tử con.

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

1. Thuộc tính định dạng font chữ, văn bản (font, text):

Thuộc tính font-family: Chọn kiểu font chữ.

Thuộc tính font-size: Chọn kích thước font chữ.

Thuộc tính font-weight: Chọn độ đậm nhạt của font chữ.

Thuộc tính font-style: Chọn kiểu chữ (thường, nghiêng, đậm).

Thuộc tính text-align: Căn chỉnh văn bản (trái, phải, giữa).

Thuộc tính text-decoration: Gạch chân, gạch ngang, bóng đổ cho văn bản.

Thuộc tính text-transform: Chuyển đổi hoa, thường, in hoa.

Thuộc tính letter-spacing: Khoảng cách giữa các chữ cái.

Thuộc tính line-height: Khoảng cách giữa các dòng văn bản.

2. Thuộc tính định dạng nền (background):

Thuộc tính background-color: Màu nền.

Thuộc tính background-image: Hình ảnh nền.

Thuộc tính background-position: Vị trí hình ảnh nền.

Thuộc tính background-repeat: Lặp lại hình ảnh nền.

Thuộc tính background-size: Kích thước hình ảnh nền.

Thuộc tính background-attachment: Cố định hình ảnh nền khi cuộn trang.

3. Mô hình hộp (box model):

Thuộc tính width: Chiều rộng của hộp.

Thuộc tính height: Chiều cao của hộp.

Thuộc tính margin: Khoảng cách giữa hộp và các phần tử xung quanh.

Thuộc tính padding: Khoảng cách giữa nội dung và viền hộp.

Thuộc tính border: Viền xung quanh hộp.

4. Các thuộc tính định margin, padding, border:

Thuộc tính margin-top: Khoảng cách trên của hộp.

Thuộc tính margin-right: Khoảng cách phải của hộp.

Thuộc tính margin-bottom: Khoảng cách dưới của hộp.

Thuộc tính margin-left: Khoảng cách trái của hộp.

Thuộc tính padding-top: Khoảng cách trên nội dung.

Thuộc tính padding-right: Khoảng cách phải nội dung.

Thuộc tính padding-bottom: Khoảng cách dưới nội dung.

Thuộc tính padding-left: Khoảng cách trái nội dung.

Thuộc tính border-width: Độ rộng viền.

Thuộc tính border-style: Kiểu viền (liền, đứt nét, chấm bi).

Thuộc tính border-color: Màu viền.

5. Thuộc tính định dạng cách hiển thị (display):

Thuộc tính display: Hiển thị phần tử như khối, nội tuyến, ẩn.

Thuộc tính float: Di chuyển phần tử sang trái hoặc phải.

Thuộc tính clear: Xóa phần tử nổi bên cạnh.

6. Thuộc tính xác định vị trí (position):

Thuộc tính position: Xác định vị trí phần tử (tĩnh, tương đối, cố định).

Thuộc tính top: Vị trí trên của phần tử.

Thuộc tính right: Vị trí phải của phần tử.

Thuộc tính bottom: Vị trí dưới của phần tử.

Thuộc tính left: Vị trí trái của phần tử.

# **3.8. Float & Clear**

Float:

Thuộc tính float cho phép di chuyển một phần tử sang trái hoặc phải của vùng chứa nó, nhưng phần tử đó vẫn nằm trong nội dung. Các phần tử khác trong nội dung sẽ "dòng chảy" xung quanh phần tử được float.

Giá trị của float:

left: Di chuyển phần tử sang trái.

right: Di chuyển phần tử sang phải.

none: Phần tử không được float (mặc định).

inherit: Phần tử thừa hưởng giá trị float của phần tử cha.

Clear:

Thuộc tính clear được sử dụng để ngăn các phần tử được float khác ảnh hưởng đến vị trí của phần tử hiện tại.

Giá trị của clear:

none: Không xóa bất kỳ phần tử float nào.

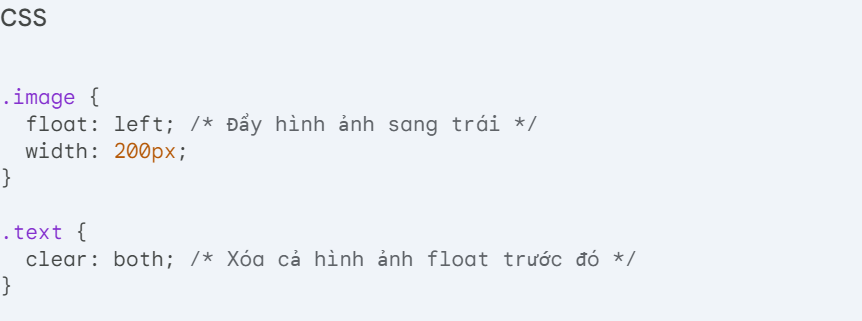
left: Xóa các phần tử float bên trái phần tử hiện tại.

right: Xóa các phần tử float bên phải phần tử hiện tại.

both: Xóa cả các phần tử float bên trái và phải phần tử hiện tại.

Sử dụng kết hợp:

Thường thì bạn sẽ sử dụng kết hợp float và clear để tạo bố cục trang web mong muốn.



## **3.9. Flex**

Flex là một tính năng mạnh mẽ trong CSS giúp đơn giản hóa việc tạo bố cục linh hoạt và đáp ứng với nhiều kích thước màn hình khác nhau. Dưới đây là một số điểm mấu chốt về Flex:

Khái niệm:

Flexbox bố trí các phần tử con bên trong một phần tử cha theo hàng dọc hoặc hàng ngang, tương tự như một hộp đựng đồ linh hoạt.

Các phần tử con được gọi là "flex items".

Bạn có thể điều khiển kích thước, vị trí và cách các flex items tương tác với nhau bằng các thuộc tính Flexbox.

Thuộc tính chính:

display: flex: Kích hoạt Flexbox cho phần tử cha.

flex-direction: Xác định hướng của flex items (row, column, row-reverse, column-reverse).

justify-content: Căn chỉnh flex items theo chiều ngang (flex-start, flex-end, center, space-between, space-around).

align-items: Căn chỉnh flex items theo chiều dọc (flex-start, flex-end, center, stretch).

flex-grow: Chỉ định mức độ flex items được phép lớn lên để lấp đầy không gian trống.

flex-shrink: Chỉ định mức độ flex items được phép nhỏ lại để vừa với không gian có sẵn.

flex-basis: Chiều rộng/chiều cao ban đầu của flex items trước khi tính toán flex-grow và flex-shrink.

Ưu điểm:

Bố cục linh hoạt và đáp ứng.

Dễ dàng tạo các bố cục phức tạp.

Cần ít code hơn so với các phương pháp bố cục khác.

Không cần dùng float và clear, tránh được các vấn đề liên quan.

Nhược điểm:

Ít được hỗ trợ trên các trình duyệt cũ.

Cần thời gian để học các thuộc tính mới.

## **3.10. Grid**

Grid Layout là một tính năng mạnh mẽ khác trong CSS, giúp bạn xây dựng bố cục trang web phức tạp một cách trực quan và dễ dàng. Nó cung cấp một hệ thống lưới vô hình gồm các hàng và cột, cho phép bạn đặt các phần tử con (grid items) chính xác vào vị trí mong muốn.

Khái niệm:

Một phần tử được kích hoạt Grid Layout bằng thuộc tính display: grid.

Các đường kẻ ngang và dọc chia phần tử cha thành các ô lưới (grid cells).

Bạn có thể đặt grid items vào các ô lưới này bằng các thuộc tính như grid-column, grid-row, grid-area.

Bạn cũng có thể điều khiển kích thước và khoảng cách giữa các ô lưới, cũng như cách các grid items tương tác với nhau.

Thuộc tính chính:

display: grid: Kích hoạt Grid Layout cho phần tử cha.

grid-template-columns: Định nghĩa số lượng và kích thước của các cột.

grid-template-rows: Định nghĩa số lượng và kích thước của các hàng.

grid-column-start: Xác định cột bắt đầu cho một grid item.

grid-row-start: Xác định hàng bắt đầu cho một grid item.

grid-column-end: Xác định cột kết thúc cho một grid item.

grid-row-end: Xác định hàng kết thúc cho một grid item.

grid-gap: Khoảng cách giữa các ô lưới.

justify-items: Căn chỉnh grid items theo chiều ngang trong một hàng/cột.

align-items: Căn chỉnh grid items theo chiều dọc trong một hàng/cột.

grid-auto-flow: Xác định cách lấp đầy các ô lưới trống.

## **3.11. CSS3**

1. CSS3 Selectors:

Descendant Selector ( > ): Chọn các phần tử con được lồng trực tiếp bên dưới một phần tử cha.

Child Selector ( ): Chọn các phần tử được đặt trong một phần tử cha, bao gồm cả các phần tử con lồng nhau.

Nth-Child Selector ( :nth-child(n) ): Chọn các phần tử con cụ thể (ví dụ: :nth-child(2) cho phần tử con thứ hai).

Nth-of-Type Selector ( :nth-of-type(n) ): Chọn các phần tử cụ thể của một kiểu cụ thể trong một phần tử cha (ví dụ: :nth-of-type(2) cho phần tử li thứ hai).

Sibling Selectors ( ~, + ): Chọn các phần tử liền kề với một phần tử cụ thể.

Universal Selector ( \* ): Chọn tất cả các phần tử (sử dụng cẩn thận vì lý do hiệu suất).

2. CSS3 Rounded Corners:

Thuộc tính border-radius: Tạo các góc bo tròn cho các phần tử. Xác định giá trị cho cả bốn góc hoặc từng góc riêng lẻ nếu cần (ví dụ: border-radius: 10px 20px;).

3. CSS3 Border Images:

Thuộc tính border-image: Thay thế đường viền liền bằng hình ảnh được cắt lát, cho phép tạo đường viền trang trí hoặc linh hoạt (ví dụ: border-image: url(border.png) 3 stretch;).

4. CSS3 Backgrounds:

Thuộc tính background: Cung cấp khả năng kiểm soát chi tiết đối với nền, bao gồm nhiều nền, hiệu ứng chuyển tiếp, vị trí và kích thước (ví dụ: background: linear-gradient(#f00, #fff) 0 0, url(image.jpg) 100px 50px no-repeat;).

5. CSS Gradients:

Thuộc tính linear-gradient và radial-gradient: Tạo chuyển tiếp màu mượt mà theo nhiều hướng và hình dạng khác nhau. Kết hợp màu sắc, hướng và điểm dừng để tùy chỉnh (ví dụ: background: linear-gradient(to right, blue 0%, green 50%, red 100%);).

6. CSS3 Shadows:

Thuộc tính box-shadow: Thêm bóng đổ, bóng đổ bên trong hoặc nhiều bóng đổ cho các phần tử (ví dụ: box-shadow: 10px 5px 5px rgba(0, 0, 0, 0.3);).

7. CSS3 Fonts:

Quy tắc font-face: Nhập phông chữ tùy chỉnh (ví dụ: @font-face { font-family: 'MyFont'; src: url(myfont.woff); }).

Thuộc tính font-size: Chấp nhận giá trị tương đối (ví dụ: em, rem) để định cỡ văn bản nhạy bén.

Thuộc tính font-weight: Mở rộng phạm vi để kiểm soát chi tiết hơn độ dày của văn bản (ví dụ: font-weight: 400, font-weight: bold).

8. CSS3 2D Transforms:

Thuộc tính transform: Dịch chuyển, xoay, thu phóng và làm lệch các phần tử để tạo hiệu ứng động (ví dụ: transform: translateX(50px) rotate(45deg);).

9. CSS3 3D Transforms:

Thuộc tính transform: Áp dụng biến đổi 3D, cho phép phối cảnh và xoay phức tạp hơn (ví dụ: transform: perspective(1000px) rotateX(30deg) rotateY(20deg);).

10. CSS3 Transitions:

Thuộc tính transition: Làm cho các thay đổi thuộc tính diễn ra mượt mà theo thời gian (ví dụ: transition: color 0.5s ease-in-out;).

11. CSS3 Multiple Columns:

Thuộc tính column-count và column-width: Tạo bố cục nhiều cột để cải thiện khả năng đọc trên màn hình rộng hơn (ví dụ: `column-count: 3; column-width: 200px

## **3.12. SCSS**

SCSS là một ngôn ngữ tiền xử lý CSS (CSS preprocessor) được viết trên cú pháp Ruby. Nó cho phép bạn viết mã CSS theo cách ngắn gọn, dễ đọc và dễ bảo trì hơn so với CSS thông thường.

Lợi ích của SCSS:

Biến: Giúp bạn lưu trữ và sử dụng lại các giá trị CSS, ví dụ như màu sắc, font chữ, kích thước.

Mixin: Giúp bạn tạo các nhóm thuộc tính CSS có thể sử dụng lại cho nhiều phần tử.

Nesting: Giúp bạn lồng các phần tử CSS vào nhau, tạo code dễ đọc và dễ quản lý hơn.

Functions: Giúp bạn thực hiện các phép toán và logic trong CSS, ví dụ như tính toán kích thước, tạo hiệu ứng.

Operators: Giúp bạn sử dụng các toán tử toán học như cộng, trừ, nhân, chia trong CSS.

Inheritance: Giúp bạn kế thừa các thuộc tính CSS từ phần tử cha sang phần tử con.

## **3.13. SASS**

SASS, viết tắt của Syntactically Awesome Style Sheets, là một preprocessor cho CSS, giúp bạn viết CSS dễ dàng hơn, duy trì và mở rộng hơn. Nó cung cấp cho bạn các tính năng như biến, mixin, nested rules (quy tắc lồng nhau), hàm, các phép toán và nhiều hơn nữa, giúp bạn viết CSS ngắn gọn, tổ chức và dễ đọc hơn.

Dưới đây là một số lợi ích chính của việc sử dụng SASS:

Học CSS dễ dàng hơn: SASS sử dụng cú pháp giống với cú pháp của nhiều ngôn ngữ lập trình khác, giúp người mới dễ dàng học CSS.

Mã CSS ngắn gọn và dễ đọc: Các tính năng như biến, mixin và nested rules giúp bạn viết mã CSS ngắn gọn và dễ hiểu hơn.

Tổ chức và duy trì mã tốt hơn: SASS giúp bạn tổ chức mã CSS thành các phần riêng biệt, dễ dàng duy trì và mở rộng.

Tính năng mạnh mẽ: SASS cung cấp các tính năng như hàm và các phép toán, giúp bạn viết CSS linh hoạt và mạnh mẽ hơn.

Kiểm soát phiên bản tốt hơn: Bạn có thể sử dụng các công cụ như Sass compilers để kiểm soát phiên bản của mã SASS và compile nó thành CSS tương thích với trình duyệt.

Tuy nhiên, SASS cũng có một số nhược điểm:

Phụ thuộc vào công cụ compile: Bạn cần sử dụng một công cụ compile để chuyển đổi SASS thành CSS trước khi hiển thị trên trình duyệt.

Có thể làm chậm quá trình phát triển: Quá trình compile SASS có thể làm chậm quá trình phát triển, đặc biệt là trên các dự án lớn.

Không phải trình duyệt nào cũng hỗ trợ trực tiếp: Trình duyệt không hiểu SASS, bạn cần compile nó thành CSS trước khi trình duyệt có thể hiển thị nội dung trang web.

**Tổng kết:**

SASS là một công cụ mạnh mẽ có thể giúp bạn viết CSS dễ dàng hơn, duy trì và mở rộng hơn. Nếu bạn đang tìm cách cải thiện khả năng viết CSS của mình, SASS là một lựa chọn tốt. Tuy nhiên, hãy cân nhắc những nhược điểm của nó trước khi sử dụng.

## 

Kết luận chương 3

CSS và CSS3 đóng vai trò quan trọng trong việc tạo ra các trang web đẹp mắt, thu hút và dễ sử dụng. CSS cung cấp các công cụ cơ bản để định dạng giao diện trang web, bao gồm màu sắc, phông chữ, bố cục, và hiệu ứng. CSS3 là phiên bản nâng cao của CSS, bổ sung nhiều tính năng mới giúp bạn tạo ra các trang web với giao diện hiện đại, đáp ứng và tương tác tốt hơn..

**CHƯƠNG 4: THIẾT KẾ WEB BÁN THIẾT BỊ Y TẾ**

## **4.1.** **Ý tưởng của Website**

1.Mua bán:

Cung cấp cho người dùng nhũng mặt hàng y tế cần thiết .

Giúp người dùng hiểu và sử dụng các mặt hàng 1 cách dễ dàng.

2.Tiện ích:

Có thể truy cập 1 cách nhanh chóng tiện lợi mọi lúc.

Nhiều mặt hàng và nhãn hàng để cho người dùng tiếp cận.

3. Kiếm tiền:

Có thể kiếm tiền từ quảng cáo, bán dụng cụ thiết bị y tế cho nhiều nhãn hàng.

Tạo ra đầu ra cho nhiều nhãn hàng kinh doanh online.

5. Quảng bá thương hiệu:

Có thể sử dụng web bán thiết bị y tế để quảng bá thương hiệu, sản phẩm hoặc dịch vụ của doanh nghiệp.

Tăng độ nhận diện thương hiệu và thu hút khách hàng tiềm năng.

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

1. Mục tiêu:

Xác định mục tiêu của website: thu hút người dùng mới, giữ chân người chơi cũ, quảng bá thương hiệu thiết bị , v.v.

Xác định đối tượng mục tiêu: trẻ em, thanh thiếu niên, người lớn, v.v.

2. Nội dung:

Xác định các nội dung chính của website: giới thiệu sản phẩm mới, sản phẩm đang có , tin tức, v.v.

Sắp xếp các nội dung theo thứ tự logic, dễ dàng tìm kiếm và truy cập.

3. Giao diện:

Sử dụng giao diện đơn giản, trực quan, phù hợp với đối tượng mục tiêu.

Sử dụng màu sắc, hình ảnh và đồ họa phù hợp với thể loại yêu cầu .

Đảm bảo website hiển thị tốt trên mọi thiết bị (PC, mobile, tablet).

4. Tính năng:

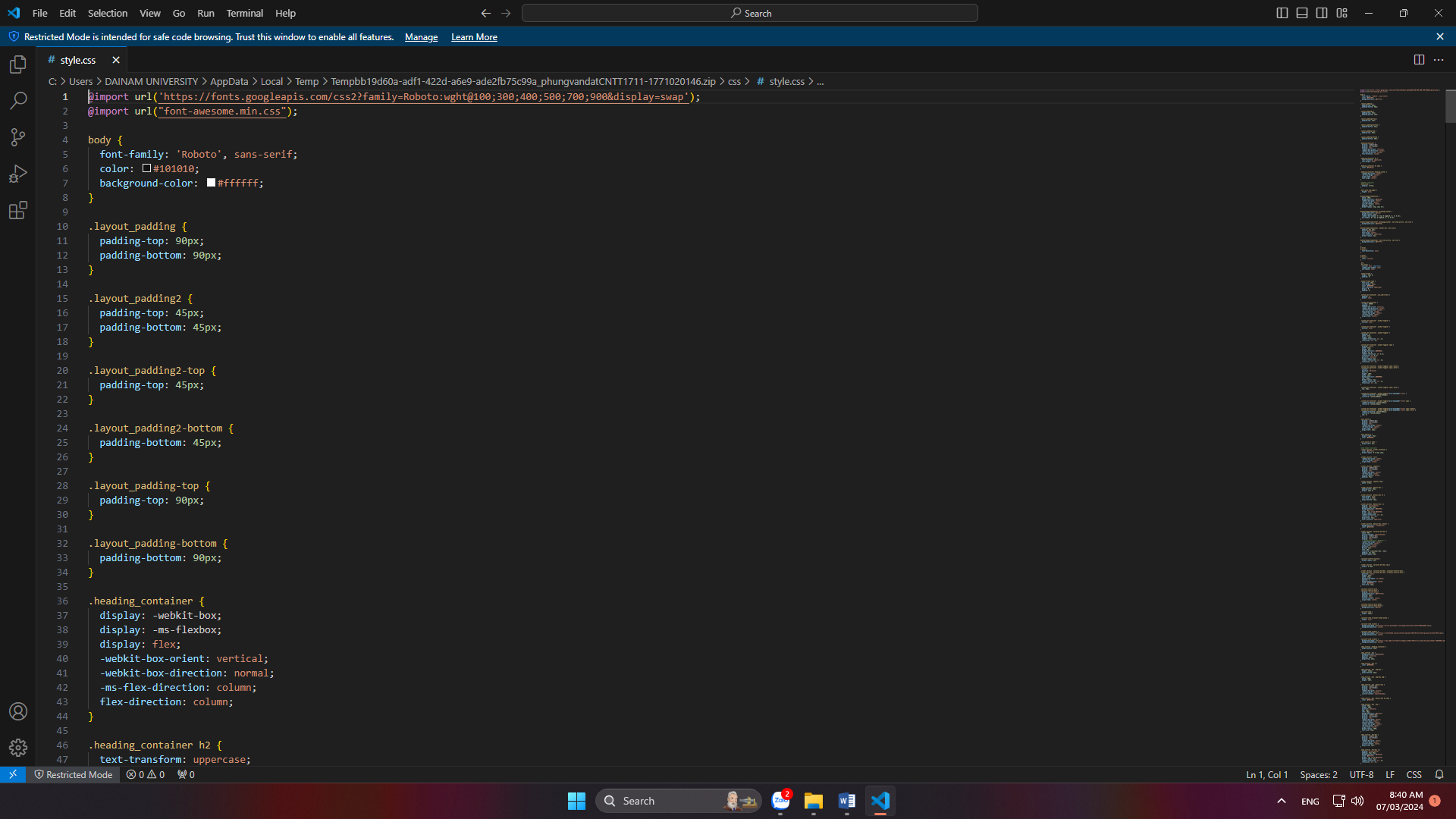
Cung cấp các tính năng cần thiết cho người dùng: đăng nhập, đăng ký, mua bán , thanh toán, v.v.

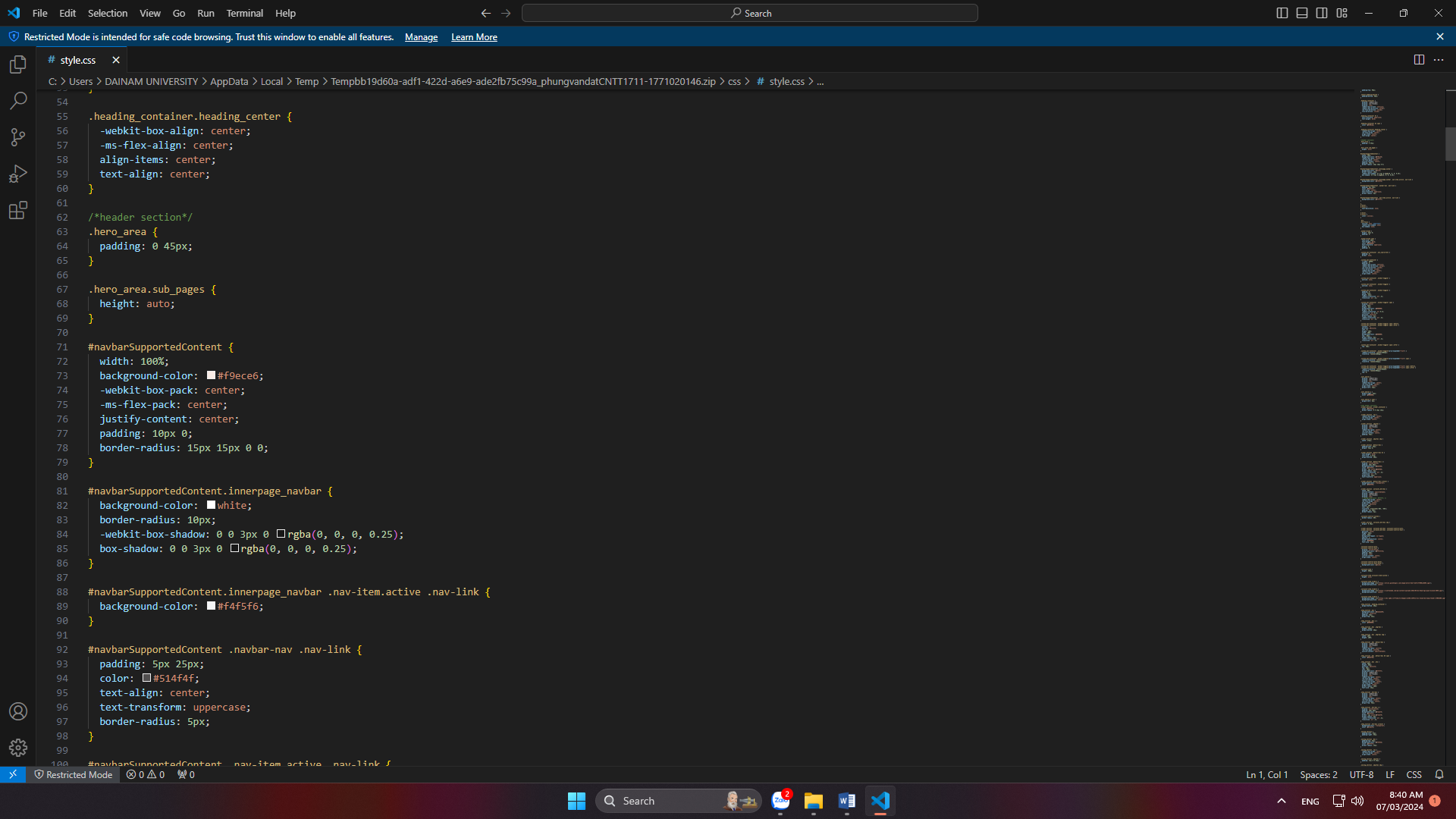
Tích hợp các tính năng mạng xã hội để chia sẻ, kết nối với bạn bè.

5. Marketing:

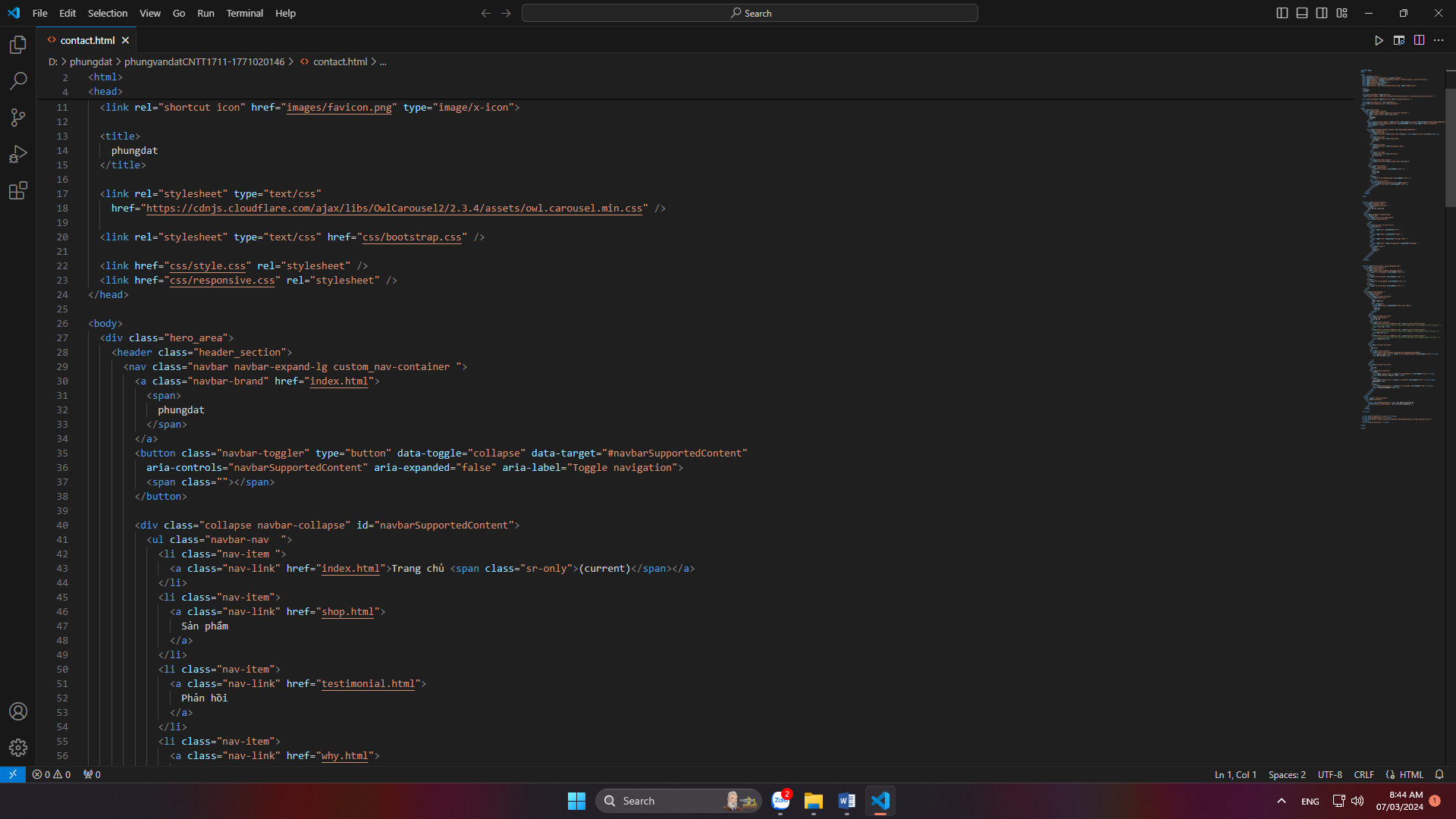
Quảng bá website trên các kênh online và offline phù hợp.

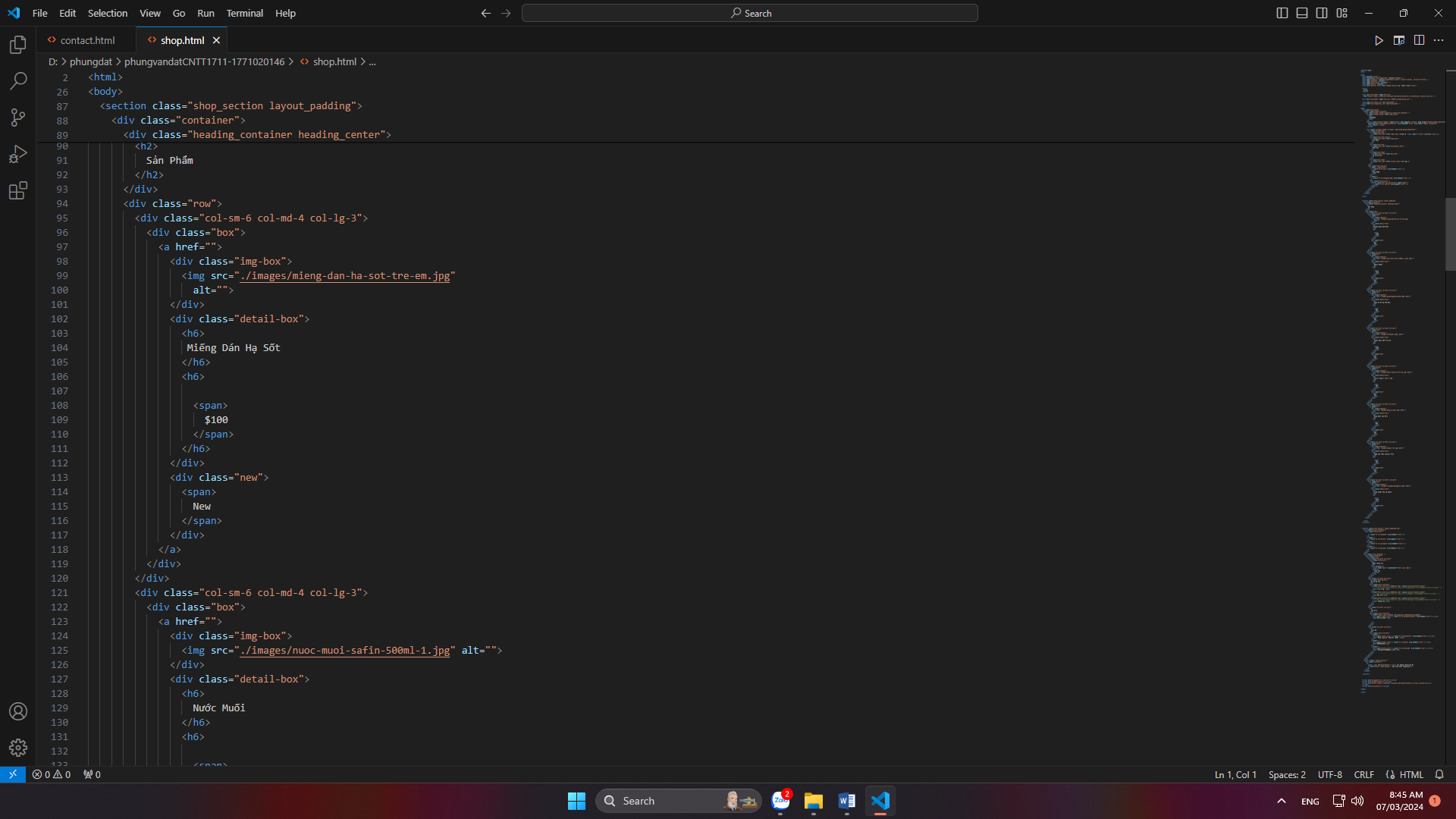
**4.3**. **Thiết kế trang Web bằng HTML và CSS**





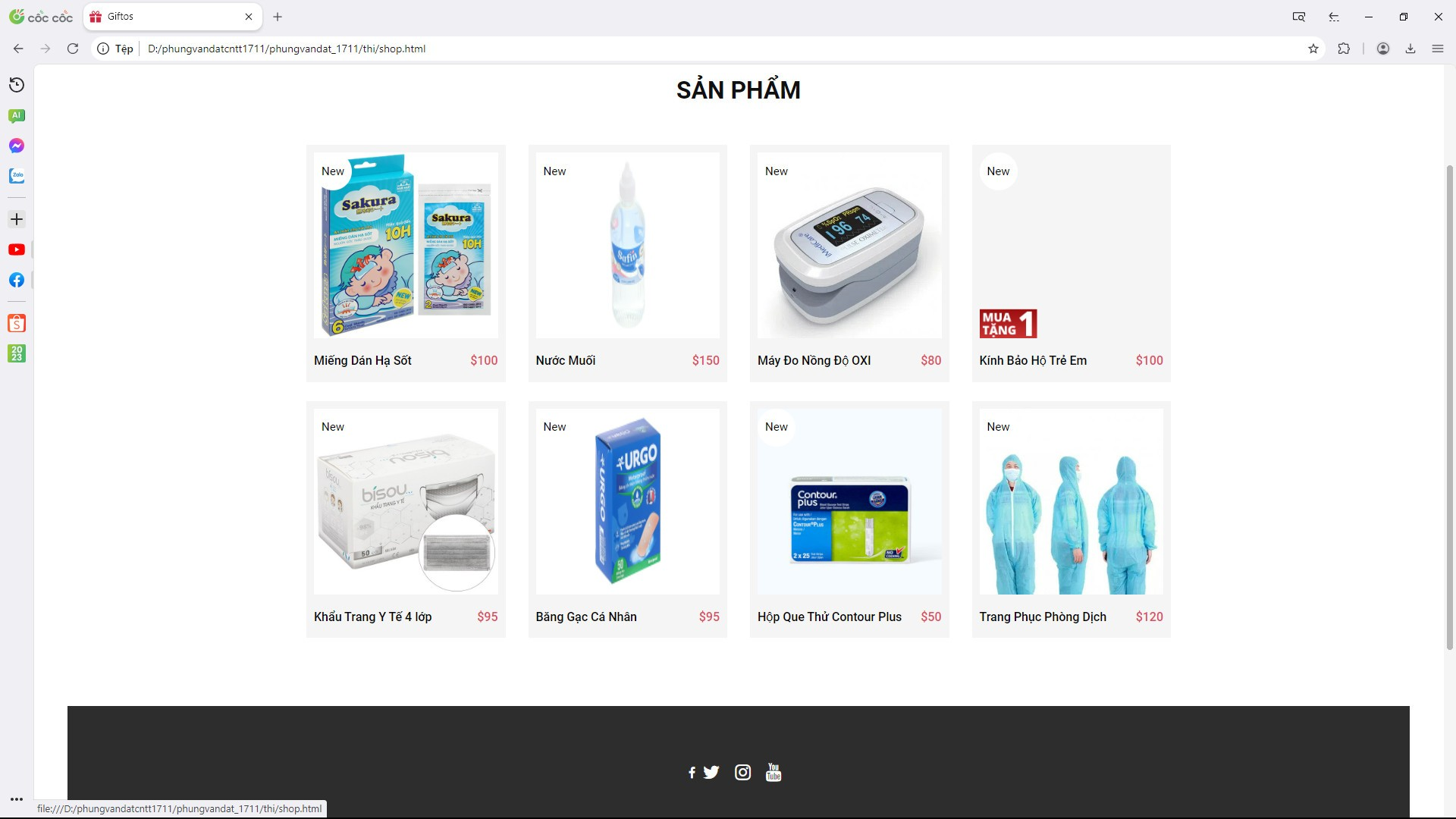
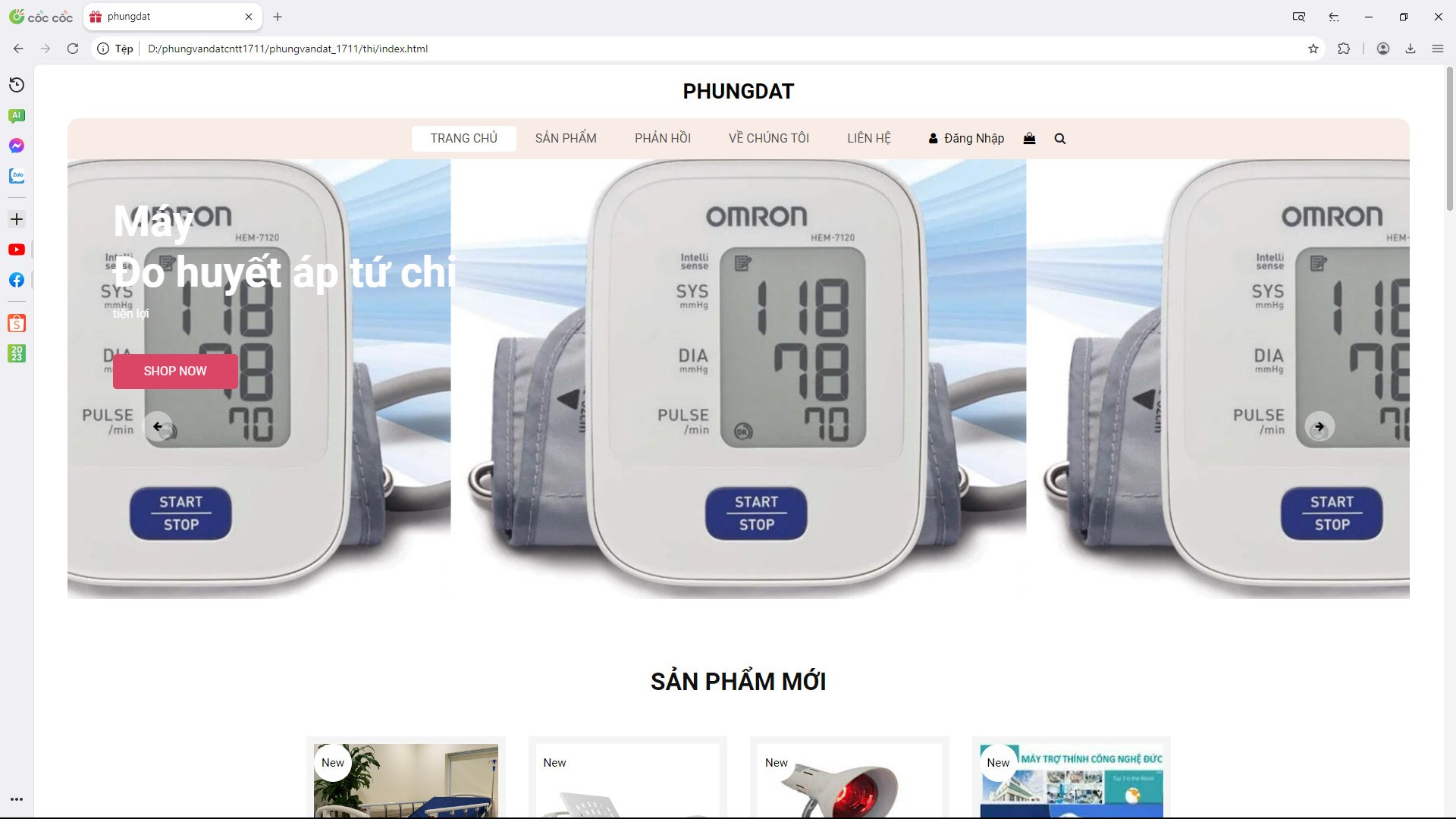
(CSS)





(HTML)

## **4.4**. **Kết quả đạt được**



## Kết luận chương 4

**KẾT LUẬN**

**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.