**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: LÀM WEB BÁN HÀNG ĐIỆN THOẠI DI ĐỘNG**

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

**Sinh viên thực hiện: Lâm Duy Khánh**

**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: LÀM WEB BÁN HÀNG ĐIỆN THOẠI DI ĐỘNG**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 01 | 1771020378 | Lâm Duy Khánh | 02/09/2005 |  |  |

### 

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

**Hà Nội, 2024**

**LỜI NÓI Đ**

**Muc Lục**

[**CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM** 10](#_Toc161691601)

[**1.2. Phân biệt Web tĩnh và Web động:** 10](#_Toc161691602)

[**1.3. Một số thuật ngữ:** 10](#_Toc161691603)

[**1.4. Một số công cụ dùng thiết kế Web:** 10](#_Toc161691604)

[**CHƯƠNG II: HTML & HTML5** 11](#_Toc161691605)

[**(**Hyper Text Markup Language**)** 11](#_Toc161691606)

[**2.1. Tổng quan về HTML** 11](#_Toc161691607)

[**2.2. Cấu trúc tổng quát trang HTML** 11](#_Toc161691608)

[**2.3. Các thẻ HTML thông dụng** 11](#_Toc161691609)

[**2.4. Các thẻ tạo biểu mẫu (form)** 12](#_Toc161691610)

[**2.5. Một số thẻ HTML đặc biệt** 12](#_Toc161691611)

[**2.6. HTML5** 13](#_Toc161691612)

[2.6.1. HTML5 là gì? 13](#_Toc161691613)

[2.6.2. Cú pháp HTML5 13](#_Toc161691614)

[2.6.3. Khai báo DOCTYPE trong HTML5 13](#_Toc161691615)

[2.6.4. Thẻ ngữ nghĩa 13](#_Toc161691616)

[2.6.5. Webform 14](#_Toc161691617)

[**CHƯƠNG III: CSS và CSS3** 15](#_Toc161691618)

[**Phần 1: CSS và CSS3** 15](#_Toc161691619)

[**1.1. CSS là gì?** 15](#_Toc161691620)

[**1.2. Cú pháp CSS** 15](#_Toc161691621)

[**1.3. Áp dụng CSS vào trang HTML** 16](#_Toc161691622)

[**1.4. Selectors** 17](#_Toc161691623)

[1.4.1. Universal selector 18](#_Toc161691624)

[1.4.2. Type selector 18](#_Toc161691625)

[1.4.3. Identity selector 19](#_Toc161691626)

[1.4.4. Class selector 20](#_Toc161691627)

[1.4.5. Descendant selector 20](#_Toc161691628)

[1.4.6. Child selector 21](#_Toc161691629)

[1.4.7. Adjacent sibling selector 22](#_Toc161691630)

[1.4.8. Attribute selector 23](#_Toc161691631)

[1.4.9. Pseudo class selector 24](#_Toc161691632)

[1.4.10. Group selector 25](#_Toc161691633)

[**1.5. Đơn vị đo lường CSS** 26](#_Toc161691634)

[**1.6. Các nhóm thuộc tính trong CSS** 27](#_Toc161691635)

[**1.7. Float & Clear** 28](#_Toc161691636)

[**1.8. Flex** 29](#_Toc161691637)

[**1.9. Grid** 30](#_Toc161691638)

[**CSS3** 31](#_Toc161691639)

[1. CSS3 Selectors: 31](#_Toc161691640)

[2. CSS3 Rounded Corners: 32](#_Toc161691641)

[3. CSS3 Border Images: 32](#_Toc161691642)

[4. CSS3 Backgrounds: 32](#_Toc161691643)

[5. CSS Gradients: 32](#_Toc161691644)

[6. CSS3 Shadows: 32](#_Toc161691645)

[7. CSS3 Fonts: 32](#_Toc161691646)

[8. CSS3 2D Transforms: 33](#_Toc161691647)

[9. CSS3 3D Transforms: 33](#_Toc161691648)

[10. CSS3 Transitions: 33](#_Toc161691649)

[11. CSS3 Multiple Columns: 33](#_Toc161691650)

[12. CSS3 Animations: 33](#_Toc161691651)

[13. SCSS và Sass: 33](#_Toc161691652)

[**CHƯƠNG 4: THIẾT KẾ WEBISTE THEO ĐỀ TÀI BẠN CHỌN** 36](#_Toc161691653)

[**4.1. Ý tưởng của Website** 36](#_Toc161691654)

[**4.2. Xây dựng bố cục của trang Web** 36](#_Toc161691655)

[**4.3. Thiết kế trang Web bằng HTML và CSS** 36](#_Toc161691656)

[**KẾT LUẬN** 46](#_Toc161691657)

[**Ưu điểm:** 46](#_Toc161691658)

[1. Giao diện thân thiện với người dùng: 46](#_Toc161691659)

[ Tính tương tác cao: 46](#_Toc161691660)

[ Tối ưu hóa cho thiết bị di động: 46](#_Toc161691661)

[ Quản lý sản phẩm hiệu quả: 46](#_Toc161691662)

[**Nhược điểm:** 46](#_Toc161691663)

[ Tính bảo mật: 46](#_Toc161691664)

[ Tốc độ tải trang: 46](#_Toc161691665)

[ Dịch vụ khách hàng: 46](#_Toc161691666)

[**Hướng phát triển chủ đề:** 46](#_Toc161691667)

[ Mở rộng danh mục sản phẩm: 46](#_Toc161691668)

[ Tích hợp công nghệ mới: 47](#_Toc161691669)

[ Phát triển chiến lược marketing: 47](#_Toc161691670)

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

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

* Web tĩnh: là các trang web chỉ chứa thông tin cố định mà không thay đổi theo người sử dụng. Các trang web tĩnh thường được viết bằng HTML và CSS.
* Web động: là các trang web có khả năng tương tác với người dùng, dựa trên dữ liệu được lưu trữ trong cơ sở dữ liệu. Các trang web động thường sử dụng ngôn ngữ lập trình như PHP, ASP.NET, Java để tạo ra các chức năng động như đăng nhập, tìm kiếm, đặt hàng, ...

## **1.3. Một số thuật ngữ:**

* Hosting: dịch vụ lưu trữ và duy trì trang web trên máy chủ internet để trang web có thể truy cập được từ mọi nơi trên thế giới.
* Tên miền (Domain Name): là địa chỉ duy nhất xác định một trang web trên internet, được sử dụng để truy cập vào trang web.
* Web Server Name: là tên của máy chủ chứa trang web.
* HomePage: trang chính của một trang web, thường là trang đầu tiên mà người dùng thấy khi truy cập vào trang web.
* Web Site: tổng thể của các trang web, hình ảnh và các tài nguyên khác tạo nên một trang web hoàn chỉnh.
* WebServer: là máy chủ chứa trang web và cung cấp nội dung cho người dùng truy cập.
* URL (Uniform Resource Location): địa chỉ dùng để xác định một tài nguyên trên internet, bao gồm tên miền, giao thức truy cập và đường dẫn đến tài nguyên đó.
* Browser: phần mềm dùng để duyệt trang web, hiển thị nội dung web cho người dùng.

## **1.4. Một số công cụ dùng thiết kế Web:**

* Visual Studio code: một trình biên soạn mã nguồn mã nguồn mở, hỗ trợ lập trình web và phát triển ứng dụng.
* Adobe Dreamweaver: một công cụ thiết kế trang web chuyên nghiệp, cho phép tạo và chỉnh sửa trang web một cách dễ dàng.

# **CHƯƠNG II: HTML & HTML5**

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

## **2.1. Tổng quan về HTML**

HTML (HyperText Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo web page. HTML giúp định dạng văn bản, chèn hình ảnh, video, âm thanh,... và liên kết các trang web với nhau.

HTML là một ngôn ngữ đánh dấu, không phải ngôn ngữ lập trình. Điều này có nghĩa là HTML không thể thực hiện các phép tính, vòng lặp, hoặc các cấu trúc điều khiển khác như các ngôn ngữ lập trình.

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

Một trang HTML được cấu trúc bởi hai phần chính:

Phần đầu (head): Phần đầu chứa các thông tin về trang web, chẳng hạn như tiêu đề trang, mô tả trang, và các thẻ meta.

Phần thân (body): Phần thân chứa nội dung chính của trang web, bao gồm văn bản, hình ảnh, video, âm thanh,...

## **2.3. Các thẻ HTML thông dụng**

HTML có rất nhiều thẻ khác nhau, mỗi thẻ có một chức năng riêng. Dưới đây là một số thẻ HTML thông dụng:

Các thẻ xử lý văn bản: Các thẻ này được sử dụng để định dạng văn bản, chẳng hạn như in đậm, in nghiêng, gạch chân,...

<b>: Định dạng văn bản in đậm

<i>: Định dạng văn bản in nghiêng

<u>: Định dạng văn bản gạch chân

<strong>: Định dạng văn bản đậm và in đậm

<em>: Định dạng văn bản nghiêng và in nghiêng

<sub>: Định dạng văn bản in nhỏ

<sup>: Định dạng văn bản in trên

Các thẻ tạo bảng: Các thẻ này được sử dụng để tạo bảng, chẳng hạn như chèn hàng, chèn cột, chèn dữ liệu,...

<table>: Tạo bảng

<tr>: Tạo hàng

<th>: Tạo tiêu đề cột

<td>: Tạo dữ liệu trong ô

Thẻ liên kết: Thẻ này được sử dụng để tạo liên kết giữa các trang web.

<a>: Tạo liên kết

href: Thuộc tính của thẻ a để chỉ định liên kết đến trang web nào

Các thẻ đa phương tiện: Các thẻ này được sử dụng để chèn hình ảnh, video, âm thanh,... vào trang web.

<img>: Chèn hình ảnh

<audio>: Chèn âm thanh

<video>: Chèn video

Thẻ tạo khung: Các thẻ này được sử dụng để chia trang web thành nhiều khung.

<frameset>: Tạo khung

<frame>: Tạo một khung trong khung

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

Các thẻ này được sử dụng để tạo biểu mẫu, chẳng hạn như nhập dữ liệu, gửi dữ liệu,...

<form>`: Tạo biểu mẫu

<input>`: Chèn trường đầu vào

<label>`: Tạo nhãn cho trường đầu vào

<textarea>`: Chèn vùng văn bản

<select>`: Chèn danh sách thả xuống

<button>`: Tạo nút

## **2.5. Một số thẻ HTML đặc biệt**

Ngoài các thẻ HTML thông dụng đã nêu trên, còn có một số thẻ HTML đặc biệt khác, chẳng hạn như:

<header>: Tạo phần đầu trang

<footer>: Tạo phần cuối trang

<nav>: Tạo phần điều hướng trang

<section>: Tạo phần nội dung trang

<aside>: Tạo phần nội dung phụ trợ

## **2.6. HTML5**

### 2.6.1. HTML5 là gì?

HTML5 là phiên bản mới nhất của ngôn ngữ HTML, được phát triển bởi World Wide Web Consortium (W3C). HTML5 được giới thiệu lần đầu tiên vào năm 2014 và hiện đang là phiên bản HTML được sử dụng phổ biến nhất.

HTML5 mang đến nhiều cải tiến so với các phiên bản HTML trước đây, bao gồm:

Hỗ trợ tốt hơn cho các phương tiện đa phương tiện, chẳng hạn như video, âm thanh, canvas,...

Hỗ trợ tốt hơn cho các thiết bị di động, chẳng hạn như điện thoại thông minh và máy tính bảng.

Hỗ trợ tốt hơn cho các thiết bị trợ năng, chẳng hạn như màn hình đọc màn hình.

### 2.6.2. Cú pháp HTML5

Cú pháp HTML5 tương tự như cú pháp HTML4, nhưng có một số điểm khác biệt đáng chú ý, chẳng hạn như:

Sử dụng thẻ DOCTYPE bắt buộc.

Sử dụng thẻ ngữ nghĩa thay thế cho thẻ div và span.

Sử dụng các thuộc tính mới, chẳng hạn như thuộc tính placeholder cho thẻ input.

### 2.6.3. Khai báo DOCTYPE trong HTML5

Trong HTML5, khai báo DOCTYPE là bắt buộc. Khai báo DOCTYPE giúp trình duyệt biết cách giải thích mã HTML.

Khai báo DOCTYPE cho HTML5 có dạng như sau:

HTML

<!DOCTYPE html>

### 2.6.4. Thẻ ngữ nghĩa

Thẻ ngữ nghĩa là các thẻ HTML được sử dụng để cung cấp thông tin về nội dung của phần tử HTML đó. Thẻ ngữ nghĩa giúp trình duyệt hiểu rõ hơn về nội dung của trang web, từ đó có thể trình bày trang web một cách hợp lý hơn.

Một số thẻ ngữ nghĩa phổ biến trong HTML5 bao gồm:

Thẻ header để chứa tiêu đề của trang web.

Thẻ nav để chứa menu điều hướng.

Thẻ section để chứa một phần nội dung của trang web.

Thẻ article để chứa một bài viết hoặc một nội dung độc lập.

Thẻ aside để chứa nội dung phụ trợ.

Thẻ footer để chứa thông tin về trang web.

### 2.6.5. Webform

Webform là một trang web được sử dụng để thu thập thông tin từ người dùng. Webform thường được sử dụng để đăng ký, đăng nhập, gửi phản hồi,...

HTML5 cung cấp một số thẻ mới để hỗ trợ việc tạo webform, chẳng hạn như:

Thẻ input để tạo các trường nhập liệu.

Thẻ select để tạo danh sách thả xuống.

Thẻ textarea để tạo vùng văn bản.

Thẻ button để tạo nút.

### Kết luận chương 2

Chương 2 đã giới thiệu về ngôn ngữ HTML và HTML5. HTML là ngôn ngữ cơ bản nhất để tạo web page. HTML5 là phiên bản mới nhất của HTML, mang đến nhiều cải tiến về khả năng hỗ trợ các phương tiện đa phương tiện, thiết bị di động và thiết bị trợ năng.

**CHƯƠNG III: CSS và CSS3**

(Cascading Style Sheets)

# **CSS và CSS3**

(Cascading Style Sheets)

## **CSS là gì?**

CSS (Cascading Style Sheets) là một ngôn ngữ đánh dấu được sử dụng để định dạng và trình bày các phần tử trên trang web. Nó là một phần quan trọng trong việc thiết kế giao diện và trực quan hóa trang web.

CSS cho phép bạn xác định các quy tắc và thuộc tính để áp dụng cho các phần tử HTML trên trang web. Thông qua CSS, bạn có thể kiểm soát màu sắc, kích thước, khoảng cách, định dạng văn bản, hình ảnh nền, đường viền, hiển thị ẩn, và nhiều thuộc tính khác của các phần tử trên trang web.

CSS sử dụng các bộ chọn (selectors) để xác định phạm vi áp dụng quy tắc CSS. Ví dụ, bạn có thể áp dụng quy tắc CSS cho một lớp cụ thể, một phần tử HTML cụ thể, hoặc một nhóm phần tử.

Tổng quan, CSS là một ngôn ngữ đánh dấu cho phép bạn định dạng và trình bày các phần tử trên trang web. Nó cung cấp khả năng tùy chỉnh và kiểm soát cao về giao diện và trực quan hóa của trang web.

### Cú pháp CSS

Cú pháp CSS (Cascading Style Sheets) bao gồm các quy tắc và thuộc tính để định dạng và trình bày các phần tử trên trang web. Dưới đây là cú pháp cơ bản của CSS:

1. Bộ chọn (Selector):

* Bộ chọn xác định các phần tử HTML mà bạn muốn áp dụng các quy tắc CSS.
* Bộ chọn có thể là tên thẻ HTML, lớp (class), ID, thuộc tính, hoặc các bộ chọn kết hợp.
* Ví dụ: p (chọn tất cả các phần tử <p>), .class (chọn các phần tử có lớp class), #id (chọn phần tử có ID).

1. Quy tắc (Rule):

* Quy tắc CSS bao gồm một bộ chọn và một khối khai báo thuộc tính.
* Ví dụ: p { color: blue; } (áp dụng màu xanh cho tất cả các phần tử <p>).

1. Thuộc tính (Property) và giá trị (Value):

* Thuộc tính xác định các đặc điểm của phần tử HTML mà bạn muốn thay đổi.
* Giá trị là giá trị cụ thể được gán cho thuộc tính.
* Ví dụ: color: blue; (thay đổi màu chữ thành màu xanh).

1. Khối khai báo (Declaration Block):

* Khối khai báo là nơi chứa một hoặc nhiều khai báo thuộc tính.
* Khối khai báo được đặt trong dấu ngoặc nhọn {}.

1. Bình luận (Comment):

* Bình luận trong CSS được sử dụng để thêm lời giải thích hoặc ghi chú trong mã.
* Bình luận bắt đầu bằng /\* và kết thúc bằng \*/.
* Ví dụ: /\* Đây là một bình luận \*/.

1. Lớp (Class) và ID:

* Lớp và ID là các thuộc tính được sử dụng để xác định các phần tử HTML cụ thể.
* Lớp được xác định bằng ký tự . và ID được xác định bằng ký tự #.
* Ví dụ: .class { color: blue; } (áp dụng màu xanh cho các phần tử có lớp class), #id { color: red; } (áp dụng màu đỏ cho phần tử có ID).

1. Kết hợp các bộ chọn:

* Bạn có thể kết hợp các bộ chọn để xác định các phần tử cụ thể.
* Ví dụ: p.class { color: blue; } (áp dụng màu xanh cho các phần tử <p> có lớp class).
* Đây chỉ là một cái nhìn tổng quan về cú pháp CSS cơ bản. CSS cung cấp nhiều thuộc tính và khả năng phức tạp hơn để tùy chỉnh và trình bày các phần tử trên trang web.

### Áp dụng CSS vào trang HTML

Để áp dụng một tệp CSS bên ngoài vào trang HTML, bạn cần làm như sau:

1. Tạo một tệp CSS: Đầu tiên, tạo một tệp có đuôi .css chứa mã CSS của bạn. Ví dụ, hãy tạo một tệp có tên là "styles.css" và đặt nó trong cùng thư mục với tệp HTML của bạn.
2. Liên kết tệp CSS: Trong phần <head> của tệp HTML, thêm một thẻ <link> để liên kết tệp CSS với tệp HTML. Thẻ <link> sẽ có các thuộc tính sau:

* rel: Xác định mối quan hệ giữa tệp HTML và tệp CSS, trong trường hợp này, sử dụng giá trị "stylesheet".
* type: Xác định loại nội dung của tệp liên kết, trong trường hợp này, sử dụng giá trị "text/css".
* href: Xác định đường dẫn tới tệp CSS (ví dụ: "styles.css").

1. Lưu và tải lại trang: Lưu tệp HTML và tệp CSS của bạn, và tải lại trang HTML trong trình duyệt. Trình duyệt sẽ tải tệp CSS và áp dụng các quy tắc CSS trong tệp đó vào trang HTML.

Bằng cách thêm thẻ <link> và liên kết tệp CSS, bạn có thể áp dụng CSS từ tệp bên ngoài vào trang HTML của bạn. Điều này cho phép bạn tách biệt mã CSS và mã HTML, giúp quản lý và chỉnh sửa giao diện trang web dễ dàng hơn.

### Selectors

Trong CSS, bộ chọn (selector) được sử dụng để xác định các phần tử HTML mà bạn muốn áp dụng các quy tắc CSS. Dưới đây là một số bộ chọn phổ biến:

1. Bộ chọn thẻ HTML:

* Chọn tất cả các phần tử của một thẻ HTML cụ thể.
* Ví dụ: p chọn tất cả các phần tử <p>.

1. Bộ chọn lớp (class):

* Chọn tất cả các phần tử có cùng lớp (class) được xác định.
* Sử dụng dấu chấm . trước tên lớp.
* Ví dụ: .my-class chọn tất cả các phần tử có lớp my-class.

1. Bộ chọn ID:

* Chọn phần tử có cùng ID được xác định.
* Sử dụng ký tự # trước tên ID.
* Ví dụ: #my-id chọn phần tử có ID my-id.

1. Bộ chọn thuộc tính:

* Chọn các phần tử dựa trên giá trị của thuộc tính.
* Ví dụ: input[type="text"] chọn tất cả các phần tử input có thuộc tính type là "text".

1. Bộ chọn kết hợp:

* Kết hợp các bộ chọn để xác định phần tử một cách chi tiết hơn.
* Ví dụ: div.container p chọn tất cả các phần tử <p> nằm bên trong phần tử <div> có lớp container.

1. Bộ chọn con:

* Chọn các phần tử con của một phần tử đã xác định.
* Ví dụ: ul li chọn tất cả các phần tử <li> nằm bên trong một phần tử <ul>.

1. Bộ chọn trạng thái:

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

CSS cung cấp nhiều bộ chọn khác nhau, cho phép bạn xác định các phần tử cần áp dụng quy tắc CSS một cách linh hoạt và chính xác.

Universal selector

Universal selector (\*) là một bộ chọn trong CSS cho phép bạn chọn tất cả các phần tử trên trang. Khi sử dụng universal selector, các quy tắc CSS sẽ được áp dụng cho mọi phần tử trong tài liệu HTML.

Ví dụ sử dụng universal selector:

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

Trong ví dụ trên, em sử dụng universal selector để áp dụng các quy tắc cho tất cả các phần tử trên trang. Quy tắc CSS này sẽ đặt giá trị margin, padding và box-sizing cho tất cả các phần tử thành 0.

Type selector

Type selector (còn được gọi là element selector) là một bộ chọn trong CSS cho phép người dùng chọn tất cả các phần tử của một loại cụ thể trên trang. Bạn có thể sử dụng tên thẻ HTML như là một bộ chọn để áp dụng các quy tắc CSS cho các phần tử thuộc loại đó.

Ví dụ sử dụng type selector:

p {

color: blue;

font-size: 16px;

}

Trong ví dụ trên, em sử dụng type selector p để áp dụng các quy tắc CSS cho tất cả các phần tử <p> trên trang. Quy tắc CSS này sẽ đặt màu chữ (color) thành màu xanh (blue) và kích thước chữ (font-size) thành 16px cho tất cả các đoạn văn bản trong các phần tử <p>.

Có thể sử dụng type selector cho bất kỳ thẻ HTML nào, ví dụ: h1, div, span, ul, li, vv. Điều này cho phép người dùng áp dụng các quy tắc CSS cho các loại phần tử khác nhau trên trang web của bạn một cách dễ dàng.

Identity selector

CSS cung cấp các loại bộ chọn khác nhau cho phép bạn chọn các phần tử dựa trên các tiêu chí khác nhau như ID, class, thuộc tính hoặc mối quan hệ phân cấp. Dưới đây là một số ví dụ về các bộ chọn phổ biến:

1. Bộ chọn ID: Chọn một phần tử dựa trên ID cụ thể.

Ví dụ: #my-element

1. Bộ chọn class: Chọn các phần tử dựa trên class cụ thể.

Ví dụ: .my-class

1. Bộ chọn thuộc tính: Chọn các phần tử dựa trên sự hiện diện hoặc giá trị của thuộc tính cụ thể.

Ví dụ: input[type="text"]

1. Bộ chọn con cháu: Chọn các phần tử là con cháu của một phần tử khác.

Ví dụ: div p

1. Bộ chọn con: Chọn các phần tử là con trực tiếp của một phần tử khác.

Ví dụ: ul > li

1. Bộ chọn giả định: Chọn các phần tử dựa trên trạng thái hoặc hành vi cụ thể.

Ví dụ: a:hover

Đây chỉ là một số ví dụ về các bộ chọn trong CSS. Mỗi bộ chọn có cú pháp và mục đích riêng, cho phép bạn chọn các phần tử cụ thể trong HTML và áp dụng các kiểu dáng tương ứng.

Class selector

Bộ chọn lớp (Class selector) trong CSS được sử dụng để chọn các phần tử dựa trên lớp (class) của chúng. Bạn có thể đặt tên lớp cho các phần tử trong HTML và sau đó sử dụng bộ chọn lớp để áp dụng các quy tắc CSS cho các phần tử đó.

Để sử dụng bộ chọn lớp, bạn cần đặt tên lớp cho các phần tử HTML bằng thuộc tính class.

Ví dụ:

<p class="highlight">Đây là đoạn văn bản được tô sáng.</p>

Trong ví dụ trên, chúng ta đã đặt lớp highlight cho đoạn văn bản <p>. Bây giờ, chúng ta có thể sử dụng bộ chọn lớp để áp dụng các quy tắc CSS cho phần tử đó.

Ví dụ:

.highlight {

background-color: yellow;

font-weight: bold;

}

Trong đoạn CSS trên, em sử dụng bộ chọn lớp .highlight để áp dụng các quy tắc cho các phần tử có lớp highlight. Quy tắc CSS này sẽ đặt màu nền (background-color) thành màu vàng và đặt độ đậm (font-weight) là đậm cho các phần tử đó.

Bộ chọn lớp cũng cho phép người dùng chọn nhiều phần tử cùng một lúc. Bạn có thể gán cùng một lớp cho nhiều phần tử và sau đó sử dụng bộ chọn lớp để áp dụng các quy tắc CSS cho tất cả các phần tử đó.

Descendant selector

Bộ chọn con cháu (Descendant selector) trong CSS được sử dụng để chọn các phần tử con cháu bên trong một phần tử cha cụ thể. Bộ chọn này cho phép bạn áp dụng các quy tắc CSS cho các phần tử con cháu trong một cấu trúc phân cấp.

Cú pháp của bộ chọn con cháu là ghi tên phần tử cha, sau đó là khoảng trắng, và cuối cùng là tên phần tử con cháu.

Ví dụ:

.container p {

color: blue;

}

Trong ví dụ trên, bộ chọn container p sẽ chọn tất cả các phần tử <p> nằm bên trong phần tử có lớp container. Các quy tắc CSS được áp dụng sẽ là màu chữ (color) là màu xanh (blue).

Bộ chọn con cháu có thể có nhiều cấp độ phân cấp.

Ví dụ:

.container div p {

font-weight: bold;

}

Trong ví dụ này, bộ chọn container div p sẽ chọn tất cả các phần tử <p> nằm bên trong các phần tử <div> nằm bên trong phần tử có lớp container. Quy tắc CSS sẽ đặt độ đậm (font-weight) là đậm cho các phần tử đó.

Bộ chọn con cháu là một công cụ mạnh mẽ trong CSS để chọn các phần tử con cháu cụ thể trong cấu trúc HTML phân cấp. Nó cho phép bạn áp dụng kiểu dáng cho các phần tử trong một phạm vi cụ thể của trang web.

Child selector

Bộ chọn con (Child selector) trong CSS được sử dụng để chọn các phần tử con trực tiếp của một phần tử cha cụ thể. Bộ chọn này chỉ chọn các phần tử con trực tiếp, không bao gồm các phần tử cháu hay các phần tử khác nằm sâu hơn trong cấu trúc phân cấp.

Cú pháp của bộ chọn con là đặt tên phần tử cha, sau đó là dấu > và cuối cùng là tên phần tử con.

Ví dụ:

.container > p {

color: blue;

}

Trong ví dụ trên, bộ chọn container > p sẽ chọn các phần tử <p> là con trực tiếp của phần tử có lớp container. Các quy tắc CSS được áp dụng sẽ là màu chữ (color) là màu xanh (blue).

Bộ chọn con chỉ áp dụng cho các phần tử con trực tiếp mà không bao gồm các phần tử cháu hay các phần tử khác sâu hơn trong cấu trúc phân cấp. Điều này giúp bạn lựa chọn và áp dụng kiểu dáng cho các phần tử cụ thể trong một cấu trúc phân cấp phức tạp hơn.

Ví dụ:

<div class="container">

<p>Đây là một đoạn văn bản</p>

<div>

<p>Đoạn văn bản này không bị chọn bởi bộ chọn con</p>

</div>

</div>

Trong ví dụ trên, chỉ có phần tử <p> đầu tiên sẽ được áp dụng quy tắc CSS do nó là con trực tiếp của phần tử <div> có lớp container. Phần tử <p> trong phần tử <div> con không được áp dụng quy tắc CSS vì nó không là con trực tiếp của <div> có lớp container.

Adjacent sibling selector

Bộ chọn anh em kế tiếp (Adjacent sibling selector) trong CSS được sử dụng để chọn các phần tử anh em kế tiếp của một phần tử cụ thể. Bộ chọn này chỉ chọn các phần tử ngay sau phần tử được chọn và có cùng phần tử cha.

Cú pháp của bộ chọn anh em kế tiếp là đặt tên phần tử đầu tiên, sau đó là dấu +, và cuối cùng là tên phần tử anh em kế tiếp.

Ví dụ:

h2 + p {

font-weight: bold;

}

Trong ví dụ trên, bộ chọn h2 + p sẽ chọn các phần tử <p> ngay sau các phần tử <h2> trong cùng phần tử cha. Các quy tắc CSS được áp dụng sẽ đặt độ đậm (font-weight) là đậm cho các phần tử đó.

Điều quan trọng cần lưu ý là bộ chọn anh em kế tiếp chỉ chọn các phần tử ngay sau phần tử được chọn và có cùng phần tử cha. Nó không áp dụng cho các phần tử anh em kế tiếp sâu hơn hay các phần tử khác không có cùng phần tử cha.

Attribute selector

Bộ chọn thuộc tính (Attribute selector) trong CSS được sử dụng để chọn các phần tử dựa trên các thuộc tính hoặc giá trị của thuộc tính của chúng. Bộ chọn này cho phép người dùng áp dụng các quy tắc CSS cho các phần tử có các thuộc tính cụ thể hoặc giá trị thuộc tính tương ứng.

Có hai loại bộ chọn thuộc tính: bộ chọn thuộc tính tồn tại (Existence attribute selector) và bộ chọn thuộc tính giá trị (Value attribute selector).

Bộ chọn thuộc tính tồn tại: Được sử dụng để chọn các phần tử có một thuộc tính cụ thể, bất kể giá trị của thuộc tính đó là gì. Cú pháp của bộ chọn này là [attribute].

Ví dụ:

a[href] {

color: blue;

}

Trong ví dụ trên, bộ chọn [href] sẽ chọn tất cả các phần tử <a> có thuộc tính href, bất kể giá trị của thuộc tính đó là gì. Các quy tắc CSS được áp dụng sẽ đặt màu chữ (color) là màu xanh (blue).

Bộ chọn thuộc tính giá trị: Được sử dụng để chọn các phần tử có một thuộc tính cụ thể và giá trị của thuộc tính đó. Cú pháp của bộ chọn này là [attribute=value].

Ví dụ:

input[type="text"] {

border: 1px solid black;

}

Trong ví dụ trên, bộ chọn [type="text"] sẽ chọn tất cả các phần tử <input> có thuộc tính type và giá trị của thuộc tính đó là "text". Các quy tắc CSS được áp dụng sẽ đặt đường viền (border) là một đường viền đen (1px solid black).

Bộ chọn thuộc tính trong CSS cho phép bạn lựa chọn và áp dụng kiểu dáng cho các phần tử dựa trên thuộc tính hoặc giá trị thuộc tính của chúng.

Pseudo class selector

Bộ chọn giả tưởng (Pseudo-class selector) trong CSS được sử dụng để chọn các trạng thái hoặc vị trí cụ thể của các phần tử. Bộ chọn này cho phép người dùng áp dụng các quy tắc CSS cho các phần tử trong các trạng thái hoặc vị trí đặc biệt.

Cú pháp của bộ chọn giả tưởng là đặt dấu hai chấm (:) sau tên phần tử hoặc bộ chọn và theo sau là tên của giả tưởng.

Ví dụ:

a:hover {

color: red;

}

Trong ví dụ trên, bộ chọn :hover sẽ chọn các phần tử <a> khi trỏ chuột lên phần tử đó. Các quy tắc CSS được áp dụng sẽ đặt màu chữ (color) là màu đỏ (red).

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

* :hover: Chọn phần tử khi trỏ chuột lên nó.
* :active: Chọn phần tử khi nó đang được nhấn xuống.
* :focus: Chọn phần tử khi nó đang được tập trung (như khi một phần tử input đang được chọn để nhập liệu).
* :first-child: Chọn phần tử là con đầu tiên của phần tử cha.
* :last-child: Chọn phần tử là con cuối cùng của phần tử cha.
* :nth-child(n): Chọn phần tử là con thứ n của phần tử cha.
* :nth-of-type(n): Chọn phần tử có kiểu cụ thể là phần tử thứ n của phần tử cha.

Bộ chọn giả tưởng trong CSS cho phép bạn lựa chọn và áp dụng kiểu dáng cho các trạng thái hoặc vị trí đặc biệt của các phần tử.

Group selector

Bộ chọn nhóm (Group selector) trong CSS cho phép người dùng kết hợp nhiều bộ chọn lại với nhau để áp dụng các quy tắc CSS cho nhiều phần tử cùng một lúc. Bằng cách sử dụng bộ chọn nhóm, bạn có thể viết một tập hợp các bộ chọn phân tách bằng dấu phẩy (,) để chọn nhiều phần tử cùng một kiểu dáng.

Cú pháp của bộ chọn nhóm là viết các bộ chọn cần nhóm lại và phân tách chúng bằng dấu phẩy.

Ví dụ:

h1, h2, h3 {

color: blue;

font-weight: bold;

}

Trong ví dụ trên, bộ chọn nhóm h1, h2, h3 sẽ áp dụng các quy tắc CSS được liệt kê cho tất cả các phần tử <h1>, <h2>, và <h3>. Các quy tắc CSS được áp dụng sẽ đặt màu chữ (color) là màu xanh (blue) và độ đậm (font-weight) là đậm.

Bằng cách sử dụng bộ chọn nhóm, bạn có thể ngắn gọn và hiệu quả áp dụng các kiểu dáng chung cho nhiều phần tử cùng một lúc.

### Đơn vị đo lường CSS

CSS hỗ trợ nhiều đơn vị đo lường khác nhau để xác định kích thước, khoảng cách và các giá trị liên quan khác trong kiểu dáng của một phần tử. Dưới đây là một số đơn vị đo lường phổ biến trong CSS:

1. Đơn vị đo tuyệt đối:

* px (pixel): Đơn vị tương đối với mật độ điểm ảnh của màn hình. Ví dụ: width: 200px;.
* cm (centimeter): Đơn vị đo theo centimet. Ví dụ: margin-top: 2cm;.
* mm (millimeter): Đơn vị đo theo millimet. Ví dụ: font-size: 5mm;.
* in (inch): Đơn vị đo theo inch. Ví dụ: width: 3in;.
* pt (point): Đơn vị đo theo điểm. Một điểm bằng 1/72 inch. Ví dụ: font-size: 12pt;.
* pc (pica): Đơn vị đo theo pica. Một pica bằng 12 điểm. Ví dụ: line-height: 1.5pc;.

1. Đơn vị đo tương đối:

* % (percent): Đơn vị tương đối theo phần trăm của phần tử cha hoặc phần tử có thuộc tính liên quan. Ví dụ: width: 50%;.
* em: Đơn vị tương đối theo kích thước phông chữ của phần tử cha gần nhất. Ví dụ: font-size: 1.2em;.
* rem: Đơn vị tương đối theo kích thước phông chữ của phần tử gốc (root). Ví dụ: font-size: 1.5rem;.
* vw (viewport width): Đơn vị tương đối theo chiều rộng của viewport (không gian hiển thị của trình duyệt). Ví dụ: width: 50vw;.
* vh (viewport height): Đơn vị tương đối theo chiều cao của viewport. Ví dụ: height: 75vh;.

1. Đơn vị đo khác:

* ex: Đơn vị đo theo chiều cao của ký tự 'x' trong phông chữ hiện tại. Ví dụ: line-height: 1.5ex;.
* ch: Đơn vị đo theo chiều rộng của ký tự '0' trong phông chữ hiện tại. Ví dụ: width: 20ch;.
* vmin (viewport minimum): Đơn vị tương đối theo chiều nhỏ nhất của viewport (chiều ngắn nhất giữa chiều rộng và chiều cao). Ví dụ: font-size: 3vmin;.
* vmax (viewport maximum): Đơn vị tương đối theo chiều lớn nhất của viewport (chiều dài nhất giữa chiều rộng và chiều cao). Ví dụ: height: 80vmax;.

Các đơn vị đo lường trong CSS cho phép bạn xác định kích thước và khoảng cách theo nhiều cách khác nhau, tùy thuộc vào yêu cầu và tình huống sử dụng.

### Các nhóm thuộc tính trong CSS

Trong CSS, các thuộc tính có thể được nhóm lại thành các nhóm chung dựa trên chức năng hoặc mục đích sử dụng của chúng. Các nhóm thuộc tính thông thường trong CSS bao gồm:

1. Thuộc tính liên quan đến kích thước và khoảng cách:

* width: Xác định chiều rộng của một phần tử.
* height: Xác định chiều cao của một phần tử.
* margin: Xác định khoảng cách bên ngoài của một phần tử.
* padding: Xác định khoảng cách bên trong của một phần tử.
* border: Xác định đường viền của một phần tử.

1. Thuộc tính liên quan đến màu sắc và nền:

* color: Xác định màu chữ của một phần tử.
* background-color: Xác định màu nền của một phần tử.
* background-image: Xác định hình ảnh nền của một phần tử.

1. Thuộc tính liên quan đến phông chữ và kiểu chữ:

* font-family: Xác định kiểu phông chữ của một phần tử.
* font-size: Xác định kích thước phông chữ của một phần tử.
* font-weight: Xác định độ đậm của phông chữ.
* text-align: Xác định căn chỉnh văn bản bên trong một phần tử.

1. Thuộc tính liên quan đến hiển thị và bố cục:

* display: Xác định cách mà phần tử được hiển thị trên trang.
* position: Xác định vị trí của một phần tử trong bố cục trang.
* float: Xác định cách mà phần tử nổi (floating) trong bố cục.
* flexbox và grid: Cung cấp các thuộc tính để tạo bố cục linh hoạt và đáp ứng.

1. Thuộc tính liên quan đến hiệu ứng và chuyển động:

* transition: Xác định các hiệu ứng chuyển đổi cho phần tử.
* animation: Xác định các hoạt hình cho phần tử.

1. Thuộc tính liên quan đến dạng và mức độ hiển thị:

* opacity: Xác định độ mờ của một phần tử.
* visibility: Xác định tính hiển thị hoặc ẩn của một phần tử.

Đây chỉ là một số nhóm thuộc tính thông thường trong CSS. CSS cung cấp rất nhiều thuộc tính khác nhau để tùy chỉnh và kiểm soát kiểu dáng của các phần tử trên trang web.

### Float & Clear

Trong CSS, float và clear là hai thuộc tính được sử dụng để tạo bố cục dạng nổi (float layout) và xử lý xung đột giữa các phần tử đang nổi. Dưới đây là giải thích về hai thuộc tính này:

1. Float (Nổi):

* Thuộc tính float trong CSS được sử dụng để định nghĩa cách mà một phần tử được đặt trong bố cục dạng nổi.
* Khi một phần tử được áp dụng thuộc tính float, nó sẽ di chuyển sang bên trái hoặc bên phải của phần tử cha. Các phần tử khác sẽ điều chỉnh vị trí của chúng để lấp đầy không gian còn lại.
* Các giá trị phổ biến cho thuộc tính float là left (nổi về bên trái) và right (nổi về bên phải).

1. Clear (Xóa):

* Thuộc tính clear trong CSS được sử dụng để xác định cách xử lý xung đột giữa các phần tử đang nổi và phần tử trước đó.
* Khi một phần tử được áp dụng thuộc tính clear, nó sẽ xác định rằng phần tử không được phép nổi bên cạnh các phần tử đang nổi trước đó.
* Các giá trị phổ biến cho thuộc tính clear là left (xóa các phần tử nổi bên trái), right (xóa các phần tử nổi bên phải) và both (xóa các phần tử nổi cả bên trái và bên phải).

### Flex

Flex là một khái niệm trong CSS để tạo bố cục linh hoạt và đáp ứng (responsive layout). Nó sử dụng thuộc tính display: flex để áp dụng cho một container (element chứa các phần tử con).

Khi một container được thiết lập với display: flex, các phần tử con bên trong nó sẽ trở thành các item linh hoạt (flex items). Thuộc tính flex và các thuộc tính liên quan được sử dụng để quy định cách các flex item sẽ phân phối và tự điều chỉnh kích thước.

Dưới đây là một số thuộc tính phổ biến trong flexbox:

1. flex-direction: Xác định hướng của các flex items trong container. Các giá trị phổ biến là row (mặc định, theo chiều ngang), column (theo chiều dọc), row-reverse (ngược lại chiều ngang) và column-reverse (ngược lại chiều dọc).
2. justify-content: Xác định cách các flex items được căn chỉnh theo chiều ngang. Các giá trị phổ biến bao gồm flex-start (căn trái), flex-end (căn phải), center (căn giữa), space-between (phân chia đều khoảng cách giữa các item), space-around (phân chia đều khoảng cách xung quanh các item) và space-evenly (phân chia đều khoảng cách xung quanh và giữa các item).
3. align-items: Xác định cách các flex items được căn chỉnh theo chiều dọc. Các giá trị phổ biến bao gồm flex-start (căn trên cùng), flex-end (căn dưới cùng), center (căn giữa), baseline (căn theo dòng cơ sở) và stretch (giãn nở để điền đầy container).
4. flex-wrap: Xác định cách các flex items được xếp hàng khi không còn đủ không gian trên một hàng duy nhất. Các giá trị phổ biến là nowrap (mặc định, không xếp hàng mới), wrap (xếp hàng mới) và wrap-reverse (xếp hàng mới theo chiều ngược lại).
5. align-content: Xác định cách các hàng của flex items được căn chỉnh theo chiều dọc khi sử dụng nhiều hàng. Các giá trị phổ biến bao gồm flex-start (căn trên cùng), flex-end (căn dưới cùng), center (căn giữa), space-between (phân chia đều khoảng cách giữa các hàng), space-around (phân chia đều khoảng cách xung quanh các hàng) và stretch (giãn nở để điền đầy container).
6. flex: Xác định tỉ lệ phân chia không gian và kích thước của các flex items trong container. Ví dụ: flex: 1 cho tất cả các item để chia đều không gian, hoặc flex: 0 0 auto cho mỗi item giữ nguyên kích thước ban đầu.

Flexbox cung cấp rất nhiều thuộc tính và tính năng khác nhau để tạo bố cục linh hoạt và điều chỉnh kích thước của các phần tử trong một container.

### Grid

Grid là một khái niệm trong CSS để tạo bố cục dạng lưới. Nó cho phép người dùng xây dựng một hệ thống lưới hai chiều với các hàng (rows) và cột (columns) để sắp xếp và điều chỉnh các phần tử trên trang web.

Để sử dụng Grid, bạn cần thiết lập một container cha (element chứa các phần tử con) với thuộc tính display: grid. Sau đó, bạn có thể xác định các hàng và cột bên trong container bằng cách sử dụng các thuộc tính như grid-template-rows, grid-template-columns, và grid-template-areas.

Dưới đây là một số thuộc tính phổ biến trong Grid:

* grid-template-rows và grid-template-columns: Xác định kích thước và chiều rộng của các hàng và cột trong grid. Bạn có thể sử dụng các giá trị như px, %, fr (đơn vị tỷ lệ linh hoạt), hoặc auto (tự động điều chỉnh kích thước).
* grid-template-areas: Xác định các khu vực (areas) của grid bằng cách gán tên cho các ô trong lưới. Bạn có thể sử dụng tên này để xác định vị trí của các phần tử con bên trong grid sử dụng thuộc tính grid-area.
* grid-gap hoặc gap: Xác định khoảng cách giữa các hàng và cột trong grid. Bạn có thể sử dụng các giá trị như px, %, hoặc em.
* grid-row và grid-column: Xác định vị trí của một phần tử trong grid bằng cách xác định số hàng và cột mà phần tử đó chiếm. Ví dụ: grid-row: 1 / 3 để phần tử chiếm từ hàng 1 đến hàng 3.
* justify-items và align-items: Xác định cách các phần tử trong grid được căn chỉnh theo chiều ngang (justify-items) và chiều dọc (align-items). Các giá trị phổ biến bao gồm start, end, center, và stretch.
* justify-content và align-content: Xác định cách các hàng và cột trong grid được căn chỉnh theo chiều ngang (justify-content) và chiều dọc (align-content) trong trường hợp không điền đầy toàn bộ kích thước của grid. Các giá trị phổ biến bao gồm start, end, center, space-between, space-around, và stretch.

Grid cung cấp khả năng mạnh mẽ để tạo bố cục đa dạng, điều chỉnh vị trí và kích thước của các phần tử trên trang web.

## **CSS3**

CSS3 là phiên bản thứ ba của CSS (Cascading Style Sheets), một ngôn ngữ dùng để mô tả và định dạng giao diện và bố cục của trang web. CSS3 là một tiêu chuẩn được phát triển bởi W3C (World Wide Web Consortium) và bổ sung nhiều tính năng mới so với phiên bản trước đó là CSS2.

1. CSS3 Selectors:

CSS3 cung cấp nhiều bộ chọn mới để lựa chọn các phần tử dựa trên các tiêu chí phức tạp hơn. Ví dụ, bộ chọn :nth-child() cho phép lựa chọn phần tử con thứ n trong một phần tử cha.

1. CSS3 Rounded Corners:

CSS3 cung cấp thuộc tính border-radius để tạo viền bo tròn cho các phần tử. Bằng cách sử dụng thuộc tính này, bạn có thể làm cho các góc của một phần tử trở nên cong.

1. CSS3 Border Images:

CSS3 cho phép sử dụng hình ảnh để tạo viền cho các phần tử bằng thuộc tính border-image. Bằng cách sử dụng thuộc tính này, bạn có thể tạo các hiệu ứng viền độc đáo cho phần tử.

1. CSS3 Backgrounds:

CSS3 cung cấp nhiều tính năng mới để tùy chỉnh nền của các phần tử. Ví dụ, thuộc tính background-size cho phép điều chỉnh kích thước hình nền, và thuộc tính background-origin cho phép xác định vị trí bắt đầu của hình nền.

1. CSS Gradients:

CSS3 hỗ trợ gradient, cho phép bạn tạo các hiệu ứng màu chuyển đổi mịn từ một màu sang một màu khác hoặc từ nhiều màu khác nhau. Bạn có thể sử dụng linear gradients hoặc radial gradients.

1. CSS3 Shadows:

CSS3 cung cấp thuộc tính box-shadow để tạo hiệu ứng đổ bóng cho các phần tử. Bằng cách sử dụng thuộc tính này, bạn có thể tạo ra các hiệu ứng đổ bóng 2D cho các phần tử.

1. CSS3 Fonts:

CSS3 cung cấp nhiều tính năng mới liên quan đến kiểu chữ. Ví dụ, @font-face cho phép bạn nhúng các phông chữ tùy chỉnh vào trang web và sử dụng chúng. Thuộc tính font-size-adjust có thể được sử dụng để điều chỉnh kích thước của phông chữ theo tỷ lệ xem.

1. CSS3 2D Transforms:

CSS3 hỗ trợ các biến đổi 2D, cho phép bạn thay đổi vị trí, kích thước và hình dạng của các phần tử. Bạn có thể sử dụng các thuộc tính như translate, rotate, scale, và skew để biến đổi phần tử theo ý muốn.

1. CSS3 3D Transforms:

CSS3 cung cấp cả các biến đổi 3D, cho phép bạn tạo ra các hiệu ứng và biến đổi ba chiều trên các phần tử. Bạn có thể sử dụng các thuộc tính như rotateX, rotateY, rotateZ, và perspective để thực hiện các biến đổi 3D.

1. CSS3 Transitions:

CSS3 cho phép bạn tạo hiệu ứng chuyển đổi mượt mà khi thuộc tính của một phần tử thay đổi. Bằng cách sử dụng thuộc tính transition, bạn có thể xác định thời gian và kiểu chuyển đổi (ví dụ: linear, ease-in, ease-out) cho các thuộc tính cụ thể.

1. CSS3 Multiple Columns:

CSS3 cho phép bạn chia nội dung văn bản thành nhiều cột. Bằng cách sử dụng thuộc tính column-count và column-width, bạn có thể xác định số lượng cột và kích thước của mỗi cột.

1. CSS3 Animations:

CSS3 cung cấp khả năng tạo ra các hoạt hình trên các phần tử. Bằng cách sử dụng @keyframes và thuộc tính animation, bạn có thể xác định các bước và thời gian chuyển đổi để tạo ra các hiệu ứng hoạt hình phức tạp trên trang web.

1. SCSS và Sass:

SCSS (Sassy CSS) và Sass (Syntactically Awesome StyleSheets) là các ngôn ngữ mở rộng của CSS. Chúng cung cấp cú pháp mạnh mẽ và linh hoạt hơn để viết mã CSS. SCSS và Sass cho phép bạn sử dụng biểu thức, biến, mixin, và nhiều tính năng khác để tạo ra mã CSS dễ đọc, tái sử dụng và dễ bảo trì hơn. SCSS là một phiên bản cú pháp tương thích ngược với CSS, trong khi Sass có một cú pháp ngắn gọn hơn mà không tương thích ngược với CSS.

Các tính năng CSS3 và ngôn ngữ Sass (SCSS và Sass). Chúng cho phép bạn tạo giao diện trang web phong phú, linh hoạt và độc đáo hơn bằng cách sử dụng các hiệu ứng, biến đổi và cú pháp tiện ích.

##### **KẾT LUẬN CHƯƠNG 3**

CSS (Cascading Style Sheets) là một ngôn ngữ được sử dụng để định dạng và tùy chỉnh giao diện của các trang web. Nó được sử dụng để xác định các quy tắc cho các phần tử HTML, bao gồm màu sắc, kích thước, khoảng cách, vị trí và các hiệu ứng khác. CSS giúp tách biệt phần nội dung (HTML) và phần trình bày (CSS), mang lại tính tái sử dụng và bảo trì dễ dàng cho các trang web.

CSS3 là phiên bản mới nhất của CSS, đưa ra nhiều tính năng mới và cải tiến so với CSS2. CSS3 giúp làm cho trang web trở nên đẹp mắt, linh hoạt và tương tác hơn. Dưới đây là một số tính năng cụ thể của CSS3:

1. CSS3 Rounded Corners: Cho phép tạo viền bo tròn cho các phần tử.
2. CSS3 Border Images: Cho phép sử dụng hình ảnh để tạo viền cho các phần tử.
3. CSS3 Backgrounds: Cung cấp nhiều tính năng mới để tùy chỉnh nền của các phần tử, ví dụ như xác định kích thước hình nền và vị trí bắt đầu của nó.
4. CSS Gradients: Cho phép tạo gradient màu chuyển đổi mịn từ một màu sang một màu khác hoặc từ nhiều màu khác nhau.
5. CSS3 Shadows: Cung cấp thuộc tính box-shadow để tạo hiệu ứng đổ bóng cho các phần tử.
6. CSS3 Fonts: Cung cấp các tính năng mới liên quan đến kiểu chữ, bao gồm khả năng nhúng phông chữ tùy chỉnh và điều chỉnh kích thước phông chữ theo tỷ lệ xem.
7. CSS3 2D Transforms: Hỗ trợ biến đổi 2D, cho phép thay đổi vị trí, kích thước và hình dạng ủa các phần tử.
8. CSS3 3D Transforms: Hỗ trợ biến đổi 3D, cho phép tạo hiệu ứng và biến đổi ba chiều trên các phần tử.
9. CSS3 Transitions: Cho phép tạo hiệu ứng chuyển đổi mượt mà khi thuộc tính của một phần tử thay đổi.
10. CSS3 Multiple Columns: Cho phép chia nội dung văn bản thành nhiều cột.
11. CSS3 Animations: Cho phép tạo các hoạt hình trên các phần tử bằng cách xác định các bước và thời gian chuyển đổi.

CSS3 là một bản cải tiến quan trọng của CSS, nó mở rộng khả năng và linh hoạt của CSS, mang lại nhiều lợi ích cho việc thiết kế và phát triển trang web hiện đại và hấp dẫn.

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

## **4.1. Ý tưởng của Website**

Thiết kế Web bán hàng điện thoại di động

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

- Bố cục web gồm :

+ Tiêu Đề

+ Sản Phẩm

+ Mệnh Giá

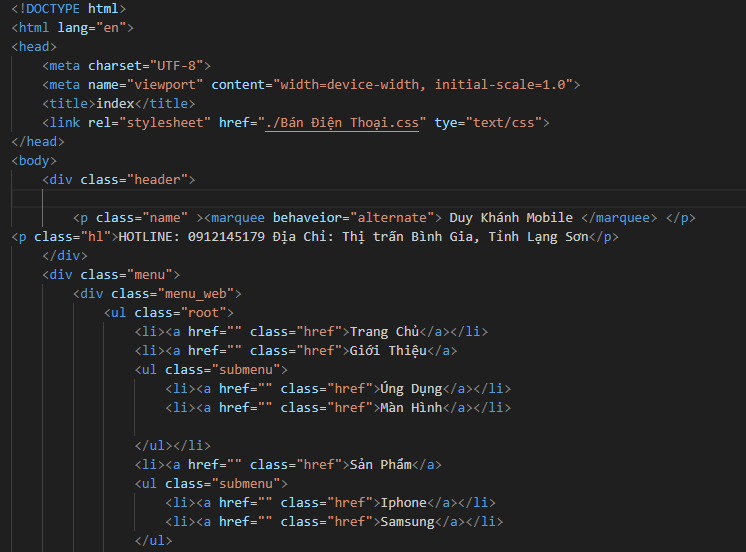
+ Địa Chỉ Liên Hệ

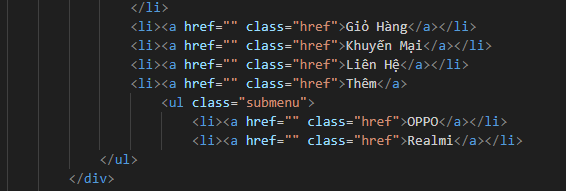
+ Các Loại Sản Phẩm

+ Giá trị từng mặt hàng

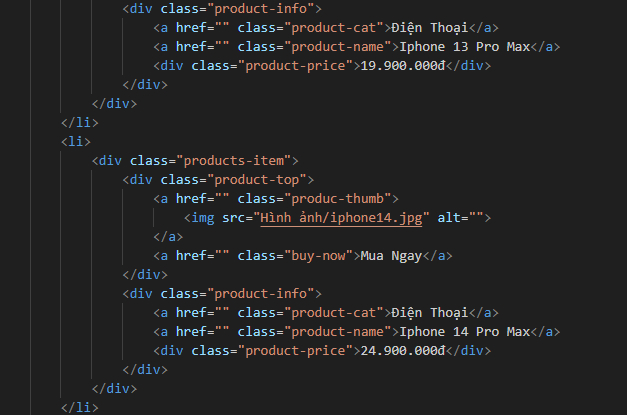
## **4.3. Thiết kế trang Web bằng HTML và CSS**

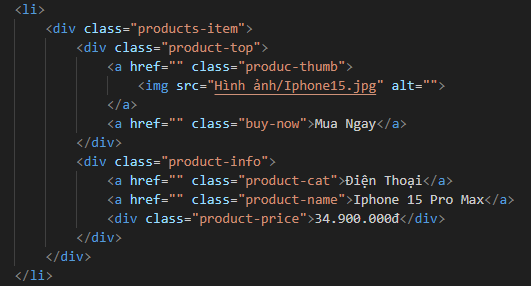
- Về HTML

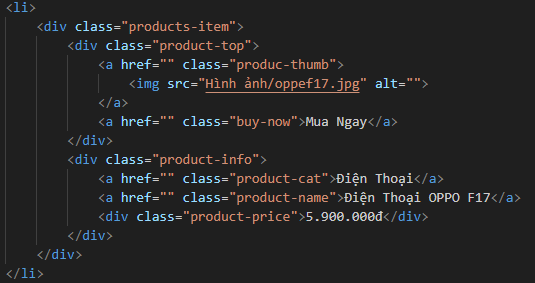


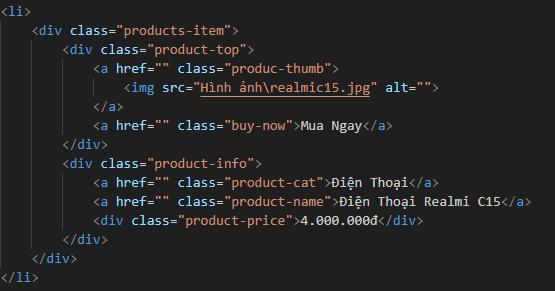


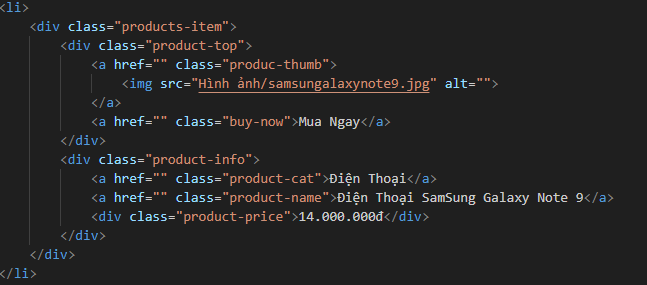


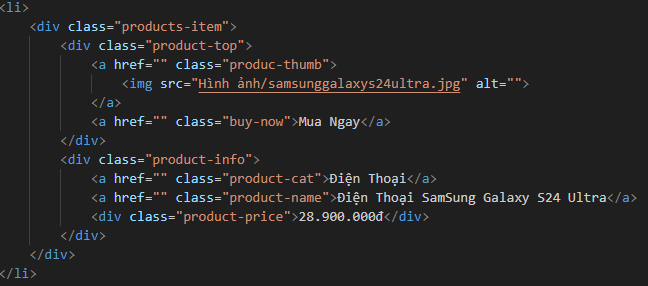


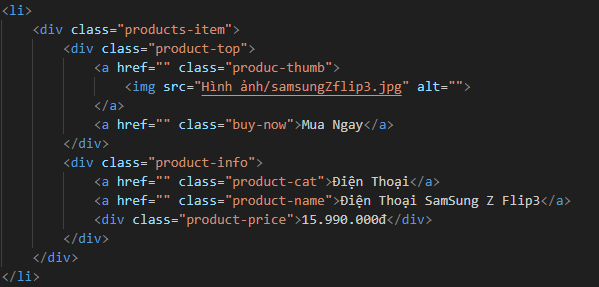


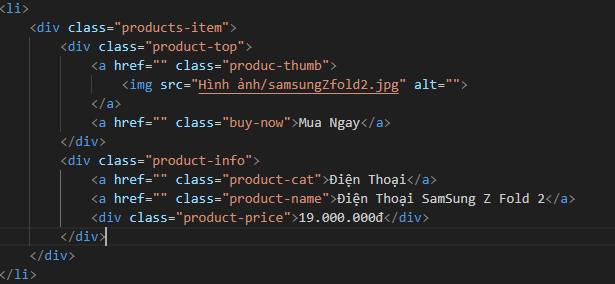


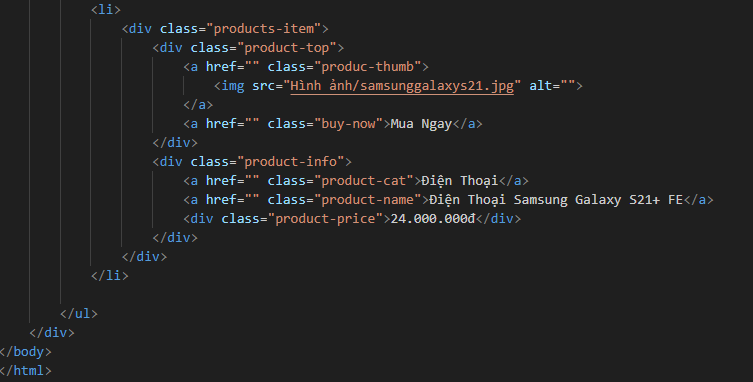




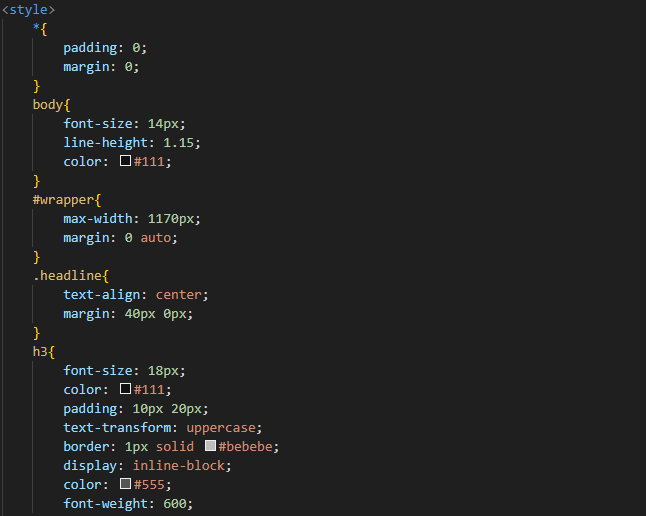


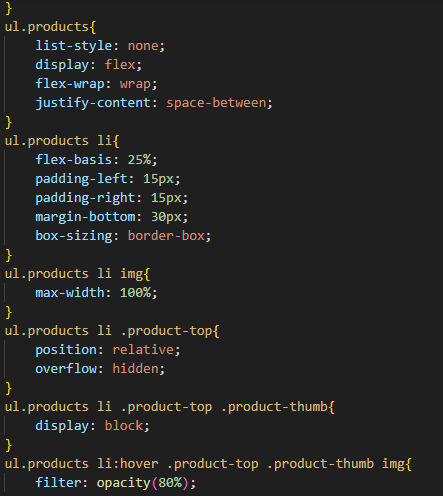
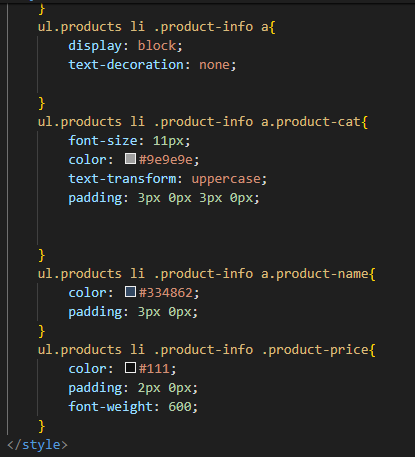


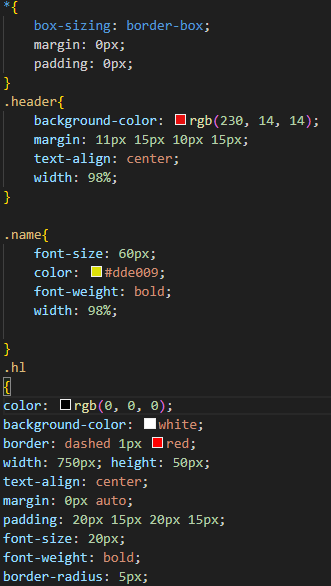
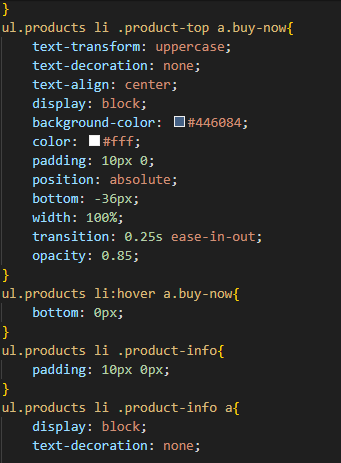


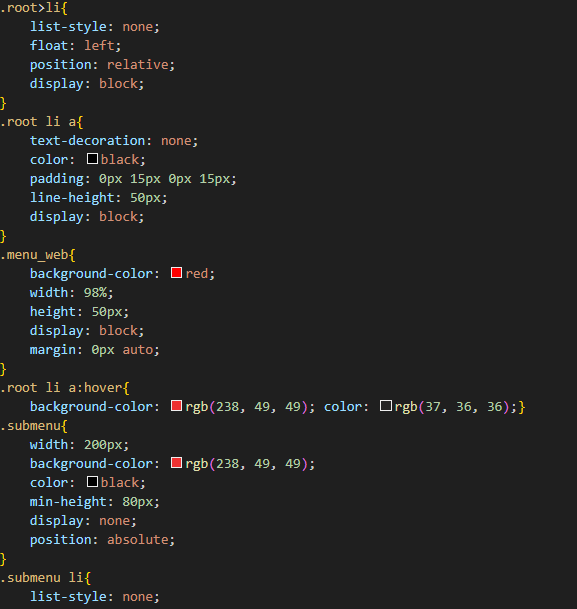
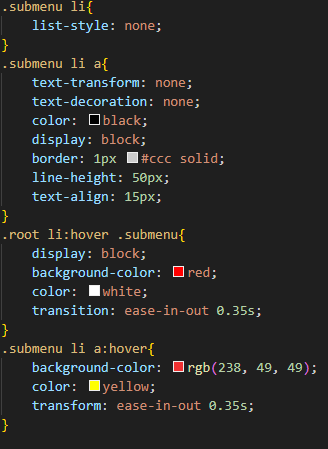


* + Về CSS

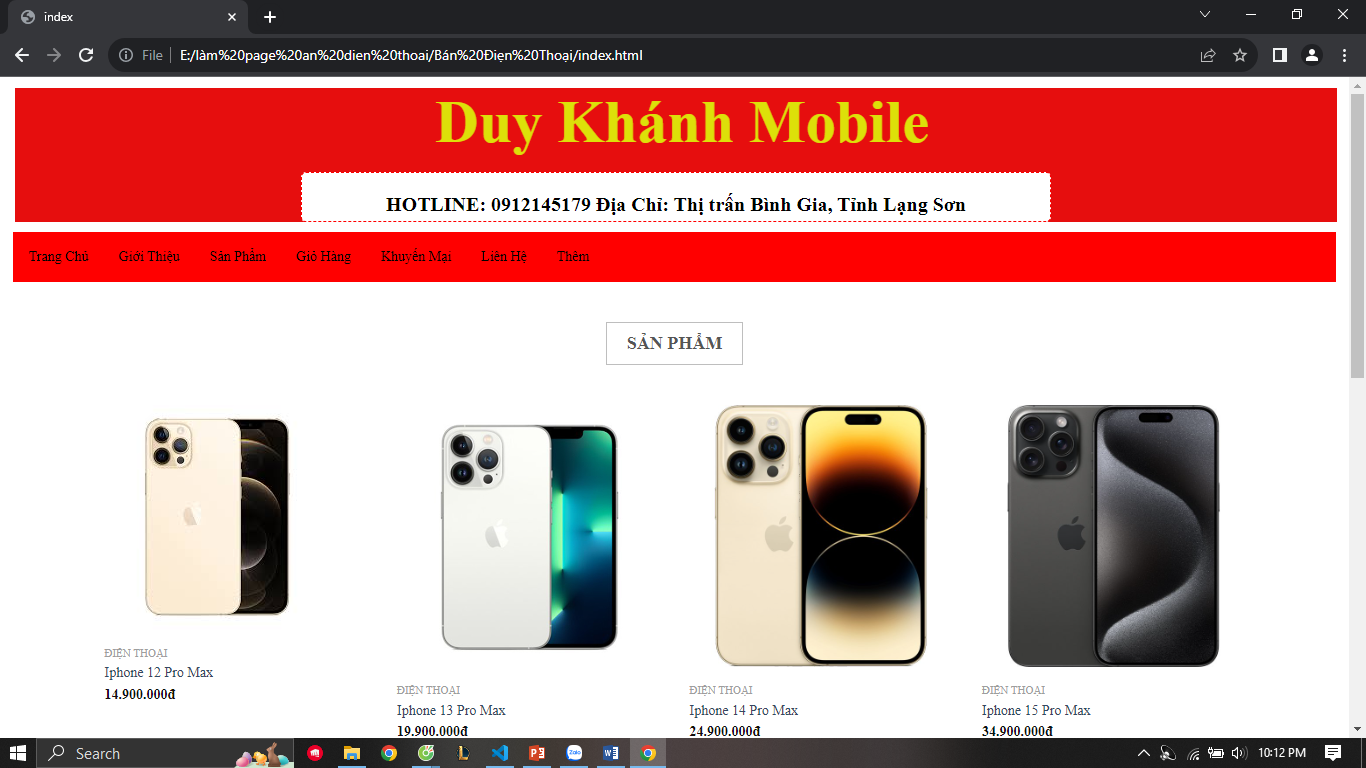


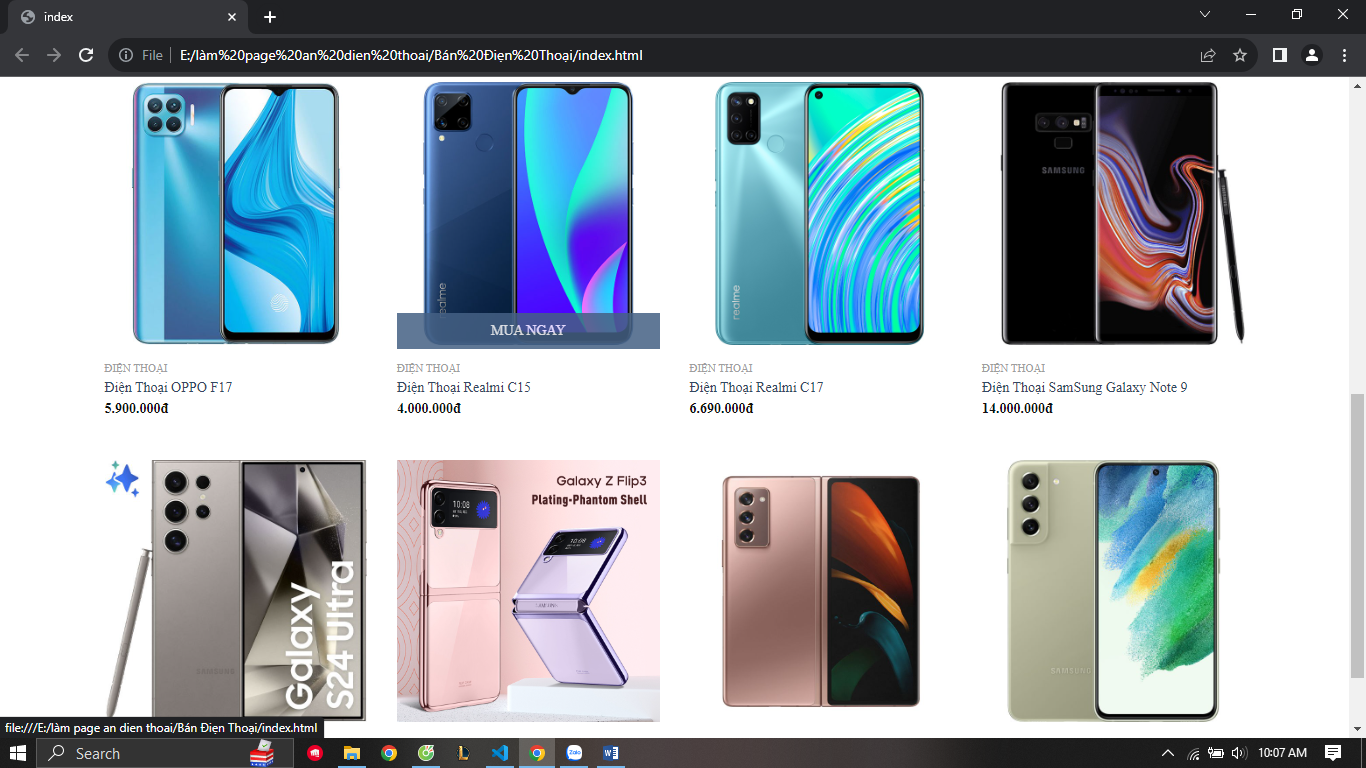






**4.5. Kết quả đạt được**





**Kết luận chương 4**

Sau khi làm ta có thể biết được, làm ra 1 cái web bán hàng cụ thể nó chình là web bán hàng điện thoại di động và biết được mục đích làm cho ta biết nhiều thêm về úng dụng html và css.

Trang web bán hàng điện thoại được thiết kế với một loạt các ưu điểm, bao gồm giao diện thân thiện với người dùng, tính tương tác cao và tối ưu hóa cho thiết bị di động. Nhờ vào những ưu điểm này, trang web đã tạo ra một trải nghiệm mua sắm thuận lợi và hấp dẫn cho người dùng.

Tuy nhiên, vẫn còn những điểm yếu cần được cải thiện, bao gồm vấn đề về tính bảo mật, tốc độ tải trang và dịch vụ khách hàng. Việc giải quyết các vấn đề này sẽ cần sự chú ý đặc biệt để đảm bảo rằng trang web hoạt động một cách trơn tru và đáp ứng được mọi nhu cầu của khách hàng.

Để phát triển chủ đề, có thể tập trung vào việc mở rộng danh mục sản phẩm, tích hợp công nghệ mới và cải thiện chiến lược marketing. Những cải tiến này sẽ giúp trang web không chỉ thu hút được nhiều khách hàng hơn mà còn nâng cao trải nghiệm mua sắm và tăng doanh số bán hàng.

Top of Form

**KẾT LUẬN**

## **Ưu điểm:**

### Giao diện thân thiện với người dùng:

Thiết kế trang web bán hàng điện thoại đảm bảo trải nghiệm người dùng thuận lợi và dễ sử dụng. Giao diện được tối ưu hóa để người dùng dễ dàng tìm kiếm, xem sản phẩm và hoàn tất giao dịch một cách nhanh chóng.

### Tính tương tác cao:

Trang web cung cấp các tính năng tương tác như bình luận, đánh giá sản phẩm, và chia sẻ trên các mạng xã hội. Điều này giúp tạo ra một cộng đồng trực tuyến sôi động và tăng tính tương tác với khách hàng.

### Tối ưu hóa cho thiết bị di động:

Thiết kế linh hoạt và đáp ứng cho phép trang web hiển thị đẹp mắt và dễ dàng trên các thiết bị di động, từ điện thoại thông minh đến máy tính bảng.

### Quản lý sản phẩm hiệu quả:

Hệ thống quản lý sản phẩm được tích hợp giúp quản lý thông tin sản phẩm, tồn kho và đơn hàng một cách hiệu quả.

## **Nhược điểm:**

### Tính bảo mật:

Cần tăng cường các biện pháp bảo mật như mã hóa dữ liệu và bảo vệ chống lại các cuộc tấn công mạng để đảm bảo thông tin cá nhân của khách hàng được bảo vệ tốt nhất.

### Tốc độ tải trang:

Đôi khi có thể gặp trở ngại với tốc độ tải trang, đặc biệt trên các thiết bị di động. Cần tối ưu hóa trang web để giảm thiểu thời gian tải và cải thiện trải nghiệm người dùng.

### Dịch vụ khách hàng:

Cần cải thiện dịch vụ khách hàng để đảm bảo phản hồi nhanh chóng và hiệu quả đối với các câu hỏi và khiếu nại từ khách hàng.

## **Hướng phát triển chủ đề:**

### Mở rộng danh mục sản phẩm:

Ngoài việc bán điện thoại, có thể mở rộng danh mục sản phẩm bao gồm phụ kiện điện thoại, đồng hồ thông minh, tai nghe và các sản phẩm công nghệ khác để tăng lượng khách hàng và doanh số bán hàng.

### Tích hợp công nghệ mới:

Liên tục cập nhật và tích hợp các công nghệ mới như trí tuệ nhân tạo (AI) và thực tế ảo (AR) vào trang web để cải thiện trải nghiệm mua sắm và tăng tính tương tác với khách hàng.

### Phát triển chiến lược marketing:

Tăng cường chiến lược marketing trực tuyến và nâng cao hiệu quả của chiến dịch quảng cáo trên các nền tảng mạng xã hội và công cụ tìm kiếm để thu hút thêm lượng khách hàng tiềm năng.

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