

TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN

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

TÊN ĐỀ TÀI: Tạo Website bán đồ điện tử sử dụng HTML5

Giảng viên hướng dẫn:

ThS. Lê Văn Phong

Sinh viên thực hiện:

Lê Hoàng Lâm

Hà Nội, 2025

TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN

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

TÊN ĐỀ TÀI: Tạo Website bán đồ điện tử sử dụng HTML5

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
				Bảng Số	Bảng Chữ
1	1971020273	Lê Hoàng Lâm	16/04/2007		

CÁN BỘ CHẤM THI 1

CÁN BỘ CHẤM THI 2

Hà Nội, 2025

LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ như hiện nay, Internet và các ứng dụng web ngày càng trở thành công cụ thiết yếu, hỗ trợ con người trong học tập, làm việc và giải trí. Các hệ thống phần mềm, đặc biệt là ứng dụng web, không chỉ giúp quản lý thông tin một cách hiệu quả mà còn tạo điều kiện thuận lợi cho việc kết nối giữa người dùng và dịch vụ. Việc thiết kế web và triển khai hệ thống phần mềm do đó trở thành một kỹ năng quan trọng, cần thiết cho sinh viên ngành công nghệ thông tin.

Bài tập lớn môn Thiết kế Web và Triển khai Hệ thống Phần mềm nhằm mục tiêu giúp sinh viên vận dụng các kiến thức lý thuyết đã học vào thực tiễn. Thông qua dự án này, sinh viên sẽ trải nghiệm toàn bộ quy trình phát triển một ứng dụng web hoàn chỉnh, từ việc phân tích yêu cầu, lập kế hoạch, thiết kế giao diện, lập trình chức năng, quản lý cơ sở dữ liệu cho đến triển khai và kiểm thử hệ thống. Quá trình này không chỉ giúp nâng cao kỹ năng thực hành mà còn rèn luyện tư duy logic, khả năng giải quyết vấn đề và làm việc nhóm.

Việc thực hiện bài tập lớn cũng giúp sinh viên nhận thức rõ hơn về tầm quan trọng của việc lập kế hoạch và triển khai hệ thống phần mềm một cách bài bản. Đồng thời, dự án cung cấp cơ hội để sinh viên áp dụng các công nghệ, công cụ và phương pháp thiết kế web hiện đại, từ đó chuẩn bị tốt cho công việc thực tế trong lĩnh vực công nghệ thông tin.

MỤC LỤC

CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM	9
1. Các khái niệm cơ bản	9
2. Phân biệt Web tĩnh, Web động	9
3. Một số thuật ngữ	9
4. Một số công cụ dùng thiết kế Web	11
5. Giới thiệu về Emmet	11
6. Kết luận chương 1	12
CHƯƠNG II: HTML & HTML5	13
1. Tổng quan về HTML	13
2. Tổng quát về HTML	14
3. Các thẻ HTML thông dụng	16
4. Các thẻ tạo biểu mẫu (form)	18
5. Một số thẻ HTML đặc biệt	19
6. HTML5	20
7. Kết luận.	23
CHƯƠNG III: CSS và CSS3	24
1. CSS là gì?	24
2. Cú pháp CSS	24
3. Áp dụng CSS vào trang HTML	24
4. Selectors	26
5. Đơn vị đo lường CSS	30
6. Kế thừa thuộc tính	31
7. Các nhóm thuộc tính trong CSS	32
8. Float & Clear	36
9. Flex	37
10. Grid	38
11. CSS3	39
12. SCSS	44
13. SASS	45
CHƯƠNG 4: THIẾT KẾ WEBISTE	48
1. Ý tưởng của Website	48
2. Xây dựng bố cục của trang Web	48
3. Thiết kế trang Web bằng HTML và CSS	49
4. Kết quả đạt được	56
5. Kết luận	59
KẾT LUẬN	60
TÀI LIỆU THAM KHẢO	62

MỤC LỤC HÌNH ẢNH

STT	Nội dung
1	Cấu trúc cơ bản của 1 trang HTML
2	Cấu trúc chuẩn của 1 tài liệu HTML
3	Cấu trúc cơ bản của 1 trang HTML5
4	Cú pháp CSS cơ bản
5	CSS nội dòng
6	CSS nội bộ
7	CSS ngoài
8	Cấu trúc mô hình hộp
9	CSS3 sử dụng Border Radius
10	CSS3 2D Transforms
11	CSS3 3D Transforms
12	CSS3 Animations
13	Bố cục trang web cơ bản
14	Thiết kế HTML phần <header>
15	Thiết kế CSS phần <header>
16	Kết quả phần <header>
17	Thiết kế HTML phần <nav>
18	Thiết kế CSS phần <nav>
19	Kết quả phần <nav>
20	Thiết kế HTML phần <section>
21	Thiết kế CSS phần <section>
22	Kết quả phần <section>
23	Thiết kế HTML phần <main>
24	Thiết kế CSS phần <main>
25	Thiết kế CSS phần product

26	Kết quả phần <main>
27	Thiết kế HTML phần <footer>
28	Thiết kế CSS phần <footer>
29	Kết quả phần <footer>
30	Giao diện trang chủ
31	Giao diện trang sản phẩm
32	Giao diện trang giới thiệu
33	Giao diện trang liên hệ

MỤC LỤC BẢNG

STT	Nội dung
1	Phân biệt web tĩnh, web động
2	Ví dụ về Emmet
3	Giải thích từng phần của HTML
4	Các thẻ tạo bảng trong HTML
5	Một số thuộc tính quan trọng của thẻ <a>
6	Các thuộc tính thường dùng để tạo khung
7	Các thuộc tính quan trọng của biểu mẫu
8	Một số thẻ HTML đặc biệt thường dùng
9	Ý nghĩa / Chức năng từng phần trong HTML5
10	Một số quy tắc, cú pháp trong HTML5
11	Các thẻ ngữ nghĩa phổ biến trong HTML5
12	Các phần tử thường dùng trong web form
13	Các phổ biến của bộ chọn thuộc tính
14	Ví dụ về Pseudo Class Selector
15	Các đơn vị tuyệt đối
16	Các đơn vị tương đối
17	Các thuộc tính có thể kế thừa
18	Các thuộc tính không kế thừa mặc định
19	Thuộc tính định dạng font chữ (Font Properties)
20	Thuộc tính định dạng văn bản (Text Properties)
21	Thuộc tính định dạng nền (Backgrounds)
22	Các thành phần của Box Model
23	Các giá trị thường dùng của Display
24	Các giá trị của Position

25	Giá trị của Float
26	Giá trị của Clear
27	Thuộc tính của phần tử cha (Container)
28	Thuộc tính của phần tử con (Item)
29	Thuộc tính của phần tử cha (Grid Container)
30	Thuộc tính của phần tử con (Grid Item)
31	Các tính năng nổi bật của SCSS
32	Các tính năng nổi bật của Sass

BẢNG CÁC TỪ VIẾT TẮT

STT	TỪ VIẾT TẮT	VIẾT ĐẦY ĐỦ
1	HTML	Hyper Text Markup Language
2	CSS	Cascading Style Sheets
3	JS	JavaScript
4	PHP	Hypertext Preprocessor
5	IDE	Integrated Development Environment
6	HTML5	HyperText Markup Language version 5
7	DOCTYPE	Document Type Declaration
8	CSS3	Cascading Style Sheets – Level 3
9	SCSS	Syntactically Awesome Style Sheets

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

1. Các khái niệm cơ bản

- Một trang trình diễn thông tin được viết bằng ngôn ngữ HTML.
- Được xem bằng trình duyệt (Web Browser).
- Một tập hợp các trang web đơn có sự liên kết, liên quan về mặt nội dung.

2. Phân biệt Web tĩnh, Web động

Bảng 1: Phân biệt Web tĩnh, Web động.

Web tĩnh	Web động
Nội dung có tần suất thay đổi ít	Nội dung được cập nhật thường xuyên
Người dùng không tương tác được	Người dùng có thể tương tác với trang, trang có hoạt cảnh hiệu ứng đẹp mắt
Để thay đổi nội dung cần có chuyên gia CNTT	Người sử dụng có thể thông qua giao diện trang để thực hiện cập nhật nội dung dễ dàng
Được viết bằng ngôn ngữ HTML & CSS	Được viết bằng ngôn ngữ HTML, CSS, JS, các ngôn ngữ lập trình backend PHP, Java, C#, Python, NodeJS, ...

3. Một số thuật ngữ

a. Hosting (Lưu trữ web):

- Là nơi lưu trữ dữ liệu của website (như hình ảnh, mã HTML, video, bài viết...)
- Khi người dùng truy cập web, dữ liệu sẽ được tải từ máy chủ hosting lên trình duyệt.
- Ví dụ: Hosting giống như “ổ cứng” của website.

b. Tên miền (Domain Name)

- Là địa chỉ của website trên Internet để người dùng dễ nhớ và truy cập.
- Dạng thường thấy: www.tenmien.com, www.truonghoc.edu.vn
- Ví dụ: www.google.com là tên miền của Google.

c. Web Server Name

- Là tên của máy chủ (server) nơi lưu trữ và chạy website.
- Có thể là tên kỹ thuật nội bộ (ví dụ: server1.hosting.com) hoặc dịch vụ web như Apache, Nginx, IIS...

d. Home Page (Trang chủ)

- Là trang đầu tiên của website khi người dùng truy cập.
- Nó thường chứa giới thiệu, menu, và liên kết đến các phần khác.
- Ví dụ: Khi vào www.facebook.com, trang đầu tiên hiện ra chính là Home Page.

e. Web Site (Trang web)

- Là tập hợp nhiều trang web con (web pages) liên kết với nhau, cùng thuộc một tên miền.
- Ví dụ: Website trường đại học có các trang: Giới thiệu, Tuyển sinh, Tin tức, Liên hệ...

f. Web Server

- Là máy tính chuyên dụng chạy phần mềm để lưu trữ và phục vụ website cho người dùng thông qua Internet.
- Các phần mềm web server phổ biến: Apache, Nginx, Microsoft IIS...

g. URL (Uniform Resource Locator)

- Là địa chỉ cụ thể để truy cập đến một trang hoặc tài nguyên trên website.
- Cấu trúc: <https://www.tenmien.com/thu-muc/trang.html>
- Ví dụ: <https://www.youtube.com/watch?v=abc123> là một URL.

h. Browser (Trình duyệt web)

- Là phần mềm dùng để truy cập và xem website.
- Các trình duyệt phổ biến: Google Chrome, Firefox, Edge, Safari...

4. Một số công cụ dùng thiết kế Web

- a. Visual Studio Code (VS Code)
 - Môi trường phát triển tích hợp hoàn chỉnh (IDE)
 - Hỗ trợ nhiều ngôn ngữ lập trình: HTML, Java, C#, PHP, ...
 - Miễn phí
 - Có nhiều bản mở rộng (extension) hỗ trợ lập trình viên
- b. Adobe Dreamweaver
 - Hỗ trợ thiết kế giao diện trang web một cách trực quan bằng cách kéo thả
- c. Sublime Text
 - Là trình soạn thảo mã gọn nhẹ, tốc độ cao, dễ dùng.
 - Có tô màu cú pháp (syntax highlighting) và tự động hoàn thành mã (auto-complete).
- d. Notepad ++
 - Là trình soạn thảo văn bản nâng cao, hỗ trợ nhiều ngôn ngữ lập trình.
 - Giao diện đơn giản, dễ học, thường dùng cho người mới bắt đầu.
- e. Figma / Adobe XD
 - Là công cụ thiết kế giao diện (UI/UX) cho website và ứng dụng.
 - Cho phép vẽ bố cục trang web, chọn màu sắc, font chữ và chia sẻ thiết kế với nhóm.
- f. Canva / Photoshop
 - Dùng để thiết kế hình ảnh, banner, logo cho website.
 - Canva: dễ dùng, trực tuyến, có sẵn mẫu.
 - Photoshop: mạnh mẽ, chuyên nghiệp hơn.

5. Giới thiệu về Emmet

Emmet là một tiện ích mở rộng (plugin) giúp viết mã HTML và CSS nhanh hơn.

Nó có sẵn trong các trình soạn thảo như Visual Studio Code, Sublime Text, Atom, Brackets, v.v.

Bảng 2: Ví dụ về Emmet

Emmet	Kết quả
html:5	Tạo khung mẫu HTML5 đầy đủ
ul>li*3	Tạo danh sách gồm 3 thẻ
div.container>h1{Hello}+p{Welcome to my site}	<div class="container"><h1>Hello</h1><p>Welcome to my site</p></div>
input:text	<input type="text">
.box	<div class="box"></div>

Lợi ích của Emmet:

- Tiết kiệm thời gian khi viết HTML/CSS.
- Giảm lỗi cú pháp vì các thẻ tự động đóng mở.
- Tăng năng suất cho lập trình viên web.
- Hỗ trợ hầu hết các IDE phổ biến (như VS Code).

6. Kết luận chương 1

Thiết kế web và triển khai hệ thống phần mềm là những bước quan trọng trong quá trình phát triển ứng dụng công nghệ thông tin.

Quá trình này không chỉ đòi hỏi kiến thức kỹ thuật (như HTML, CSS, JavaScript, cơ sở dữ liệu, hosting, domain...) mà còn cần tư duy sáng tạo, logic và khả năng phân tích yêu cầu người dùng.

Thiết kế web giúp xây dựng giao diện thân thiện, dễ sử dụng, thẩm mỹ và tối ưu trải nghiệm người dùng (UX/UI).

Trong khi đó, triển khai hệ thống phần mềm đảm bảo ứng dụng hoạt động ổn định, bảo mật và hiệu quả trong môi trường thực tế.

Việc kết hợp tốt giữa thiết kế giao diện, lập trình chức năng, kiểm thử và triển khai hệ thống sẽ giúp tạo ra sản phẩm phần mềm hoàn chỉnh, đáp ứng đúng mục tiêu, nhu cầu và hiệu quả kinh tế.

CHƯƠNG II: HTML & HTML5

(Hyper Text Markup Language)

Cách dùng các thẻ bằng emmet trong HTML

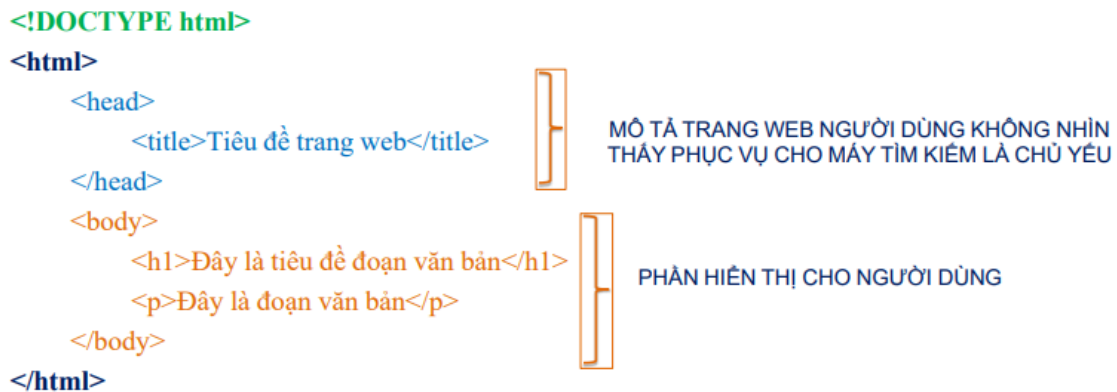
1. Tổng quan về HTML

1.1. Khái niệm

HTML (Hypertext Markup Language - Ngôn ngữ Đánh dấu Siêu văn bản) là một ngôn ngữ để mô tả cấu trúc của một trang web. Cấu trúc cơ bản của một trang HTML

1.2. Cấu trúc cơ bản của một trang HTML

Một tài liệu HTML thông thường có cấu trúc như sau:



Ảnh 1: Cấu trúc cơ bản của một trang HTML

1.3. Các thành phần chính trong HTML

- Khai báo **<!DOCTYPE html>** định nghĩa tài liệu này là tài liệu HTML5
- Thành phần **<html>** là thành phần gốc của trang HTML
- **<head>** chứa các thông tin bổ sung về trang HTML
- **<title>** định nghĩa tiêu đề trang web (được hiển thị ở thanh tiêu đề trên trình duyệt)
- **<body>** định nghĩa thân của tài liệu HTML chứa đựng các thành phần nhìn thấy trên trang web
- **<h1>**, **<p>** các thẻ định nghĩa dữ liệu của HTML giải thích nhanh

1.4. Vai trò của HTML trong thiết kế web

- Xây dựng cấu trúc và nội dung cho website.
- Kết hợp với:
 - CSS để trang trí, định dạng giao diện.

- JavaScript để tạo tương tác và hiệu ứng động.

- Là nền tảng cơ bản mà mọi lập trình viên web cần nắm vững.

1.5. Ưu điểm của HTML

- Dễ học, dễ hiểu.
- Tương thích với hầu hết các trình duyệt.
- Miễn phí, không cần cài đặt thêm.
- Là bước đầu tiên để học thiết kế và phát triển website.

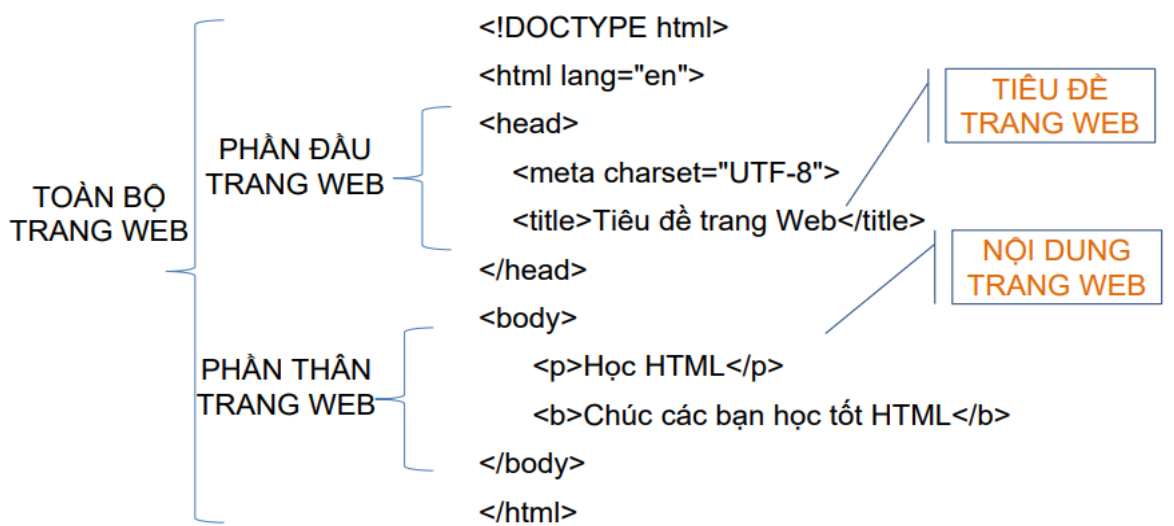
1.6. Kết luận.

HTML là ngôn ngữ nền tảng của web, giúp xác định cấu trúc và nội dung trang web.

Nắm vững HTML là bước khởi đầu quan trọng để tiến tới học CSS, JavaScript và các công nghệ web nâng cao khác.

2. Tổng quát về HTML

2.1. Cấu trúc chuẩn của một tài liệu HTML



Ảnh 2: Cấu trúc chuẩn của một tài liệu HTML

2.2. Giải thích từng phần

Bảng 3: Giải thích từng phần của HTML

Phần	Thẻ HTML	Chức năng / Ý nghĩa
Toàn bộ trang Web	<!DOCTYPE html>	Xác định trang sử dụng chuẩn HTML5.
Phần gốc của trang	<html> ... </html>	Bao toàn bộ nội dung của trang HTML.
Phần đầu trang Web	<head> ... </head>	Chứa thông tin cấu hình trang web như tiêu đề, kiểu chữ, từ khóa, CSS, biểu tượng...

Phần thân trang Web	<body> ... </body>	Chứa nội dung hiển thị cho người dùng như văn bản, hình ảnh, video, liên kết, biểu mẫu...
---------------------	--------------------	---

2.3. Một số thẻ thường gặp trong mỗi phần

- Trong <head>
 - <title>: Tiêu đề trang hiển thị trên tab trình duyệt.
 - <meta charset="UTF-8">: Thiết lập bảng mã hiển thị tiếng Việt.
 - <meta name="viewport" content="width=device-width, initial-scale=1.0">: Giúp trang web hiển thị tốt trên điện thoại.
 - <link>: Liên kết file CSS ngoài.
 - <script>: Chèn hoặc liên kết file JavaScript.
- Trong <body>:
 - <h1> → <h6>: Tiêu đề các cấp.
 - <p>: Đoạn văn bản.
 - : Hình ảnh.
 - <a>: Liên kết.
 - <div>, <section>, <footer>, <header>: Chia bố cục trang.

2.4. Tóm tắt sơ đồ cấu trúc HTML

```

<!DOCTYPE html>

|
└─ <html>
    └─ <head>
        |   └─ <meta> ...
        |   └─ <title>...</title>
        |   └─ (liên kết CSS, JS, favicon...)
        |
        └─ <body>
            └─ <header>...</header>
            └─ <main>...</main>
  
```


└── <section>...</section>
 └── <footer>...</footer>
 └── (nội dung trang web hiển thị)

Kết luận:

- Một trang HTML chuẩn gồm 3 phần chính:
 - Khai báo tài liệu (<!DOCTYPE html>)
 - Phần đầu (<head>) chứa thông tin về trang
 - Phần thân (<body>) chứa nội dung hiển thị
- Nắm vững cấu trúc này là bước đầu tiên quan trọng để xây dựng và phát triển website.

3. Các thẻ HTML thông dụng

- Các thẻ tạo bảng

Bảng 4: Các thẻ tạo bảng trong HTML

Thẻ	Ý nghĩa / Chức năng	Ví dụ minh họa
<table>	Thẻ bao toàn bộ bảng	<table>...</table>
<tr>	Tạo một hàng (row) trong bảng	<tr>...</tr>
<th>	Ô tiêu đề cột (header cell) – chữ in đậm, canh giữa	<th>Tên</th>
<td>	Ô dữ liệu (data cell) – chứa nội dung của bảng	<td>Nguyễn Văn A</td>
<caption>	Thêm tiêu đề cho bảng (hiển thị phía trên bảng)	<caption>BẢNG ĐIỂM</caption>

- Thẻ liên kết: Trong HTML, thẻ dùng để tạo liên kết là <a> (anchor tag).

Bảng 5: Một số thuộc tính quan trọng của thẻ <a>

Thuộc tính	Ý nghĩa	Ví dụ
href	Đường dẫn đến trang/tệp liên kết	Giới thiệu
target	Quy định mở liên kết ở đâu	Mở tab mới
title	Hiển thị chú thích khi rê chuột	Trang chủ
download	Cho phép tải tệp khi nhấp	Tải file PDF

mailto:	Gửi email trực tiếp	<code>Liên hệ</code>
tel:	Gọi điện (trên thiết bị di động)	<code>Gọi ngay</code>

- Các thẻ đa phương tiện

- Thẻ chèn hình ảnh

```

```

Giải thích:

- + src: Đường dẫn đến tệp ảnh (bắt buộc)
- + alt: Văn bản thay thế nếu ảnh không hiển thị
- + width, height: Kích thước ảnh (đơn vị px hoặc %)
- + title: Chú thích khi rê chuột lên ảnh

- Thẻ chèn âm thanh — `<audio>`

```
<audio controls>
```

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

Trình duyệt của bạn không hỗ trợ thẻ audio.

```
</audio>
```

Giải thích:

- + controls: Hiển thị thanh điều khiển (phát, dừng, âm lượng)
- + autoplay: Tự động phát khi tải trang
- + loop: Phát lại liên tục
- + muted: Tắt tiếng khi bắt đầu

- Thẻ chèn video — `<video>`

```
<video width="400" height="300" controls>
```

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

Trình duyệt của bạn không hỗ trợ thẻ video.

```
</video>
```

Giải thích:

- + controls: Hiển thị thanh điều khiển video
- + autoplay: Tự động phát video khi trang tải
- + loop: Phát lại liên tục

- + poster: Ảnh đại diện hiển thị trước khi phát
- + muted: Bắt đầu ở chế độ tắt tiếng
- Thẻ nhúng nội dung bên ngoài — <iframe>

```
<iframe src="https://www.youtube.com/embed/abcd1234" width="560"
height="315" allowfullscreen></iframe>
```

Giải thích:

- + src: Địa chỉ trang hoặc nội dung muốn nhúng
- + width, height: Kích thước khung
- + allowfullscreen: Cho phép hiển thị toàn màn hình

- Thẻ tạo khung

- Thẻ <iframe> – Khung nội tuyến (HTML5)

```
<iframe src="đường_dẫn" width="600" height="400" title="Tên khung"></iframe>
```

Các thuộc tính thường dùng:

Bảng 6: Các thuộc tính thường dùng để tạo khung

Thuộc tính	Ý nghĩa	Ví dụ
src	Đường dẫn đến trang hoặc tài liệu cần hiển thị	<iframe src="about.html">
width, height	Chiều rộng và chiều cao khung	<iframe width="800" height="600">
name	Đặt tên cho khung (để liên kết từ trang khác mở vào)	<iframe name="khungchinh">
title	Tên mô tả cho khung (hỗ trợ truy cập)	<iframe title="Khung chính">
frameborder	Ẩn/hiện đường viền (0 = ẩn)	<iframe frameborder="0">
allowfullscreen	Cho phép hiển thị toàn màn hình (khi nhúng video)	<iframe allowfullscreen>
scrolling	Cho phép cuộn nội dung (yes/no/auto)	<iframe scrolling="yes">

4. Các thẻ tạo biểu mẫu (form)

Thẻ chính của biểu mẫu — <form>

```
<form action="trang_xu_ly.php" method="post">
```

```
<!-- các phần tử trong form -->
```

```
</form>
```

Bảng 7: Các thuộc tính quan trọng của biểu mẫu

Thuộc tính	Ý nghĩa	Ví dụ
action	Xác định trang hoặc tệp xử lý dữ liệu sau khi gửi form	action="xuly.php"
method	Cách gửi dữ liệu: GET (hiển thị trên URL) hoặc POST (ẩn dữ liệu)	method="post"
name	Tên của form (phân biệt khi có nhiều form)	name="form1"
target	Nơi hiển thị kết quả (cửa sổ, tab...)	target="_blank"

5. Một số thẻ HTML đặc biệt

Bảng 8: Một số thẻ HTML đặc biệt thường dùng

Thẻ	Chức năng	Ví dụ minh họa
<marquee>	Tạo hiệu ứng chạy chữ (ngang hoặc dọc)	<marquee>Chào mừng bạn đến với website!</marquee>
<hr>	Tạo đường kẻ ngang để phân cách nội dung	<hr>
 	Xuống dòng (ngắt dòng) trong văn bản	Xin chào HTML!
<pre>	Hiển thị văn bản giữ nguyên định dạng (cách dòng, khoảng trắng)	<pre>Xin chào HTML</pre>
<sup>	Hiển thị chữ nhỏ nằm trên dòng (số mũ)	x² → x ²
<sub>	Hiển thị chữ nhỏ nằm dưới dòng (chỉ số)	H₂O → H ₂ O
<abbr>	Viết tắt, hiển thị chú thích khi di chuột	<abbr title="HyperText Markup Language">HTML</abbr>
<bdo>	Đảo ngược hướng văn bản (trái → phải hoặc phải → trái)	<bdo dir="rtl">Xin chào</bdo>
<code>	Hiển thị đoạn mã lập trình (kiểu monospace)	<code>print("Hello")</code>
<kbd>	Hiển thị phím người dùng nhập	Nhấn <kbd>Ctrl</kbd> + <kbd>S</kbd>
<samp>	Hiển thị kết quả đầu ra của chương trình	<samp>Hello World!</samp>
<cite>	Dùng để trích dẫn tiêu đề (tên sách, tác phẩm)	<cite>Truyện Kiều</cite>
<blockquote>	Trích dẫn đoạn văn dài	<blockquote>“Học, học nữa, học mãi.”</blockquote>

<address>	Hiển thị thông tin liên hệ (tên, email, địa chỉ)	<address>Liên hệ: abc@gmail.com</address>
<wbr>	Cho phép ngắt dòng khi cần thiết trong từ dài	Một chuỗi<wbr>dài không<wbr>ngắt dòng

6. HTML5

6.1. HTML5 là gì?

- Bản nâng cấp của HTML, cải tiến để website thân thiện hơn với người dùng
- Hỗ trợ tối đa cho ứng dụng đa phương tiện
- HTML5 sử dụng SQL database, Application Cache
- Hỗ trợ các ứng dụng SVG, Canvas, ...
- HTML 5 hỗ trợ tốt hơn JS, ứng dụng chạy nền với sự hỗ trợ Web Worker API
- Hỗ trợ đồ họa vector

6.2. Cú pháp HTML5

Cấu trúc cơ bản của một trang HTML5

```

<> a.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Document</title>
7  </head>
8  <body>
9
10 </body>
11 </html>

```

Ảnh 3: Cấu trúc cơ bản của một trang HTML5

Bảng 9: Ý nghĩa / Chức năng từng phần trong HTML5

Thành phần	Ý nghĩa / Chức năng
<!DOCTYPE html>	Khai báo phiên bản HTML5 – giúp trình duyệt hiểu rằng đây là trang HTML5 (không phân biệt chữ hoa/thường).
<html lang="vi">...</html>	Thẻ gốc của toàn bộ trang web, thuộc tính lang chỉ định ngôn ngữ của trang (ở đây là tiếng Việt).
<head>...</head>	Chứa thông tin mô tả của trang (metadata), không hiển thị trực tiếp lên trình duyệt.

<code><meta charset="UTF-8"></code>	Xác định bảng mã ký tự UTF-8, giúp hiển thị đúng tiếng Việt.
<code><title>...</title></code>	Tiêu đề của trang, hiển thị trên tab của trình duyệt.
<code><body>...</body></code>	Chứa nội dung chính hiển thị cho người dùng (văn bản, hình ảnh, video, bảng, biểu mẫu, v.v.).

Bảng 10: Một số quy tắc cú pháp trong HTML5

Quy tắc	Mô tả / Ví dụ
Không cần đóng thẻ tự kết thúc	Ví dụ: <code>
</code> , <code><hr></code> , <code></code> (không cần viết <code>
</code> như XHTML)
Không phân biệt chữ hoa, chữ thường	<code><BODY></code> và <code><body></code> đều hợp lệ, nhưng nên viết chữ thường để thống nhất
Có thể bỏ qua một số thẻ	Một số thẻ như <code><html></code> , <code><head></code> , <code><body></code> có thể được trình duyệt tự hiểu, nhưng nên viết đầy đủ để rõ ràng
Thuộc tính không cần ngoặc kép nếu không có khoảng trắng	<code><input type=text></code> hợp lệ, nhưng nên viết <code><input type="text"></code> để dễ đọc
Hỗ trợ ký tự đặc biệt	Dùng <code>&lt;</code> để hiển thị <code><</code> , <code>&gt;</code> cho <code>></code> , <code>&copy;</code> cho ký hiệu ©,...

6.3. Khai báo DOCTYPE trong HTML5

Khái niệm DOCTYPE: DOCTYPE (viết tắt của Document Type Declaration) là dòng khai báo đầu tiên trong tài liệu HTML, giúp trình duyệt hiểu và hiển thị đúng phiên bản HTML mà trang web đang sử dụng.

Cú pháp khai báo DOCTYPE trong HTML5: `<!DOCTYPE html>`

DOCTYPE trong HTML5 là dòng khai báo bắt buộc ở đầu trang web, giúp trình duyệt hiểu và hiển thị trang theo chuẩn HTML5.

Thẻ ngữ nghĩa

Khái niệm thẻ ngữ nghĩa: Thẻ ngữ nghĩa (Semantic Tags) là các thẻ HTML5 có ý nghĩa mô tả rõ ràng nội dung mà chúng bao bọc. Nói cách khác, thay vì chỉ định hình dạng hiển thị, thẻ ngữ nghĩa giúp trình duyệt và công cụ tìm kiếm (SEO) hiểu rõ cấu trúc nội dung của trang web.

Bảng 11: Các thẻ ngữ nghĩa phổ biến trong HTML5

Thẻ	Chức năng / Ý nghĩa
<code><header></code>	Phần đầu trang hoặc đầu phần nội dung (chứa logo, menu, tiêu đề)
<code><nav></code>	Khu vực liên kết điều hướng (menu, thanh điều hướng)
<code><section></code>	Một phần nội dung riêng biệt, thường chứa chủ đề cụ thể
<code><article></code>	Bài viết độc lập (tin tức, blog, bài đăng, sản phẩm,...)

<aside>	Nội dung phụ, thường là quảng cáo, liên kết, thanh bên
<main>	Phần nội dung chính của trang (mỗi trang chỉ có 1 <main>)
<footer>	Phần chân trang (bản quyền, thông tin liên hệ, liên kết)
<figure>	Chứa hình ảnh hoặc biểu đồ minh họa
<figcaption>	Chú thích cho hình ảnh hoặc biểu đồ trong <figure>
<mark>	Tô sáng văn bản, đánh dấu nội dung quan trọng
<time>	Hiển thị thời gian, ngày tháng theo chuẩn máy đọc hiểu

6.4. Webform

Khái niệm Web Form: Web Form (hay biểu mẫu web) là thành phần trong trang HTML cho phép người dùng nhập dữ liệu và gửi lên máy chủ (server) để xử lý.

Cấu trúc cơ bản của Web Form:

```
<form action="đường_dẫn_xử lý" method="phương_thức">
    <!-- Các phần tử nhập liệu -->
</form>
```

Giải thích:

- <form>: thẻ bao toàn bộ biểu mẫu.
- action: chỉ định địa chỉ (URL) của tệp hoặc trang xử lý dữ liệu (ví dụ: process.php).
- method: phương thức gửi dữ liệu, thường dùng:
- GET → gửi dữ liệu lên URL (thường dùng cho tìm kiếm, xem thông tin)
- POST → gửi dữ liệu ẩn, an toàn hơn (thường dùng cho đăng nhập, đăng ký, gửi form liên hệ)

Bảng 12: Các phần tử thường dùng trong Web Form

Thẻ	Chức năng
<input>	Tạo các ô nhập dữ liệu (text, password, email, number, radio, checkbox, button, file, ...)
<label>	Nhãn mô tả cho ô nhập liệu
<textarea>	Ô nhập nhiều dòng (dùng cho nội dung dài, ví dụ: ghi chú, bình luận)
<select> và <option>	Tạo danh sách chọn (menu xổ xuống)
<button>	Nút nhấn (gửi form hoặc thao tác khác)
<fieldset>	Nhóm các phần tử có liên quan
<legend>	Tiêu đề của nhóm <fieldset>
<input type="submit">	Nút gửi dữ liệu

<code><input type="reset"></code>	Nút xóa trắng các ô nhập
---	--------------------------

7. Kết luận.

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu chuẩn dùng để xây dựng và tổ chức nội dung của trang web.

Nó đóng vai trò là nền tảng cốt lõi trong phát triển web, giúp trình duyệt hiểu và hiển thị văn bản, hình ảnh, bảng, liên kết, biểu mẫu... trên Internet.

Sự ra đời của HTML5 đã nâng cấp đáng kể so với các phiên bản trước, mang lại nhiều thẻ ngữ nghĩa (semantic tags), hỗ trợ đa phương tiện (audio, video), tương thích di động tốt hơn, và tối ưu cho SEO và hiệu năng.

HTML5 cũng hỗ trợ tích hợp với CSS3 và JavaScript dễ dàng hơn, giúp việc thiết kế web trở nên hiện đại, tương tác cao và thân thiện với người dùng.

CHƯƠNG III: CSS và CSS3

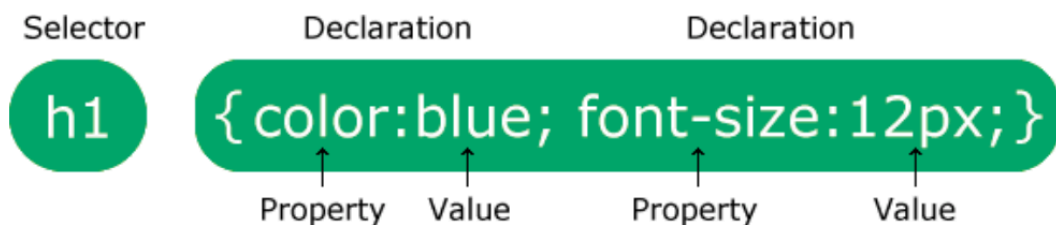
(Cascading Style Sheets)

Áp dụng cách dùng các thẻ bằng emmet trong CSS

1. CSS là gì?

CSS (viết tắt của Cascading Style Sheets) là ngôn ngữ dùng để định dạng và trình bày nội dung của trang web được viết bằng HTML.

2. Cú pháp CSS



Ảnh 4: Cú pháp CSS cơ bản

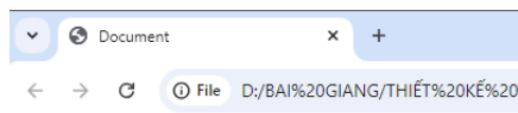
- Cú pháp CSS bao gồm bộ chọn và khai báo
- Bộ chọn chỉ ra các thành phần HTML bạn muốn áp dụng
- Khởi khai báo chứa một hoặc nhiều khai báo cách nhau bằng “;”
- Mỗi khai báo chứa thuộc tính và giá trị của thuộc tính cách nhau “:”

3. Áp dụng CSS vào trang HTML

Để CSS tác động đến HTML, ta cần liên kết (hoặc nhúng) các quy tắc CSS vào trang web. Có 3 cách áp dụng CSS vào trang HTML:

- 3.1. Inline CSS (CSS nội dòng):** CSS được viết trực tiếp trong thẻ HTML, thông qua thuộc tính style.

```
<p style="color: ■ red;font-weight: bold;">
    Nhúng trực tiếp CSS vào trong thẻ
</p>
```



Nhúng trực tiếp CSS vào trong thẻ

Ảnh 5: CSS nội dòng

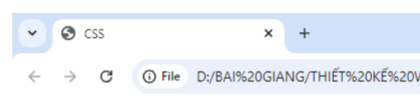
3.2. Internal CSS (CSS nội bộ): CSS được đặt trong phần <head> của trang HTML, bên trong cặp thẻ <style>.

```
<p>
    Nhúng CSS nội bộ vào HTML!
</p>
```

Nội dung tài liệu HTML

```
<style>
    p {
        color: ■ blue;
        font-weight: bold;
        font-size: 20px;
    }
</style>
```

Nội dung CSS



Nhúng CSS nội bộ vào HTML!

Ảnh 6: CSS nội bộ

3.3. External CSS (CSS ngoài): CSS được lưu trong một tệp riêng (.css) và liên kết với HTML qua thẻ <link> trong <head>.

Cú pháp:

```
<link rel="stylesheet" href="ten_file.css">
```

Tái sử dụng tệp CSS cho các tài liệu HTML khác tương tự.

Nội dung tài liệu HTML

```
<p>
    Nhúng CSS từ một File khác vào HTML!
</p>
<h1>Dùng thẻ Link để tham chiếu sang.</h1>
```

Nội dung CSS

```
h1 {
    font-size: 40px;
    color: blueviolet;
    background-color: aqua;
    margin: 20px;
}

p {
    font-size: 30px;
    color: chocolate;
    font-family: Arial, Helvetica, sans-serif;
    font-weight: bold;
}
```

```
<link rel="stylesheet" href="./style.css">
```

Tham chiếu CSS sang HTML



Nhúng CSS từ một File khác vào HTML!

Dùng thẻ Link để tham chiếu sang.

Ảnh 7: CSS ngoài

4. Selectors

- CSS Selector là gì?

Selector (bộ chọn) trong CSS là phần dùng để xác định phần tử HTML muốn áp dụng kiểu định dạng (style).

Có thể chia bộ chọn thành 5 loại:

- Bộ chọn đơn giản theo id, tên, lớp
- Bộ chọn kết hợp lựa chọn thành phần theo mối quan hệ giữa chúng
- Bộ chọn lớp giả lựa chọn thành phần theo trạng thái của nó
- Bộ chọn thành phần giả: lựa chọn và chỉnh phong cách từng phần
- Bộ chọn thuộc tính: lựa chọn theo thuộc tính hoặc giá trị thuộc tính

4.1. Universal selector

- Cú pháp:

```
* {
    property: value;
}
```

- Giải thích:

- Dấu * là ký hiệu của Universal selector.
- Nó áp dụng các quy tắc CSS cho mọi thẻ HTML trong trang, trừ khi có quy tắc cụ thể hơn ghi đè lại

4.2. Tag name selector

- Cú pháp

```
tên_thẻ {
    thuộc_tính: giá_trị;
```

}

- Tóm tắt:
 - Bộ chọn: Tag name selector
 - Mô tả: Chọn phần tử theo tên thẻ HTML
 - Ví dụ: p {color: red;}

4.3. Id selector

- Cú pháp

```
#id_name {  
    thuộc_tính: giá_trị;  
}
```

- Giải thích
 - Dấu # được dùng để chọn phần tử có id tương ứng.
 - Mỗi id trong một trang HTML nên là duy nhất, nghĩa là chỉ dùng cho một phần tử

4.4. Class selector

- Cú pháp

```
.tenlop {  
    thuoc_tinh: gia_tri;  
}
```

- Giải thích
 - Dấu chấm (.) đứng trước tên lớp.
 - Tên lớp phải trùng với giá trị của thuộc tính class trong thẻ HTML

4.5. Descendant selector

- Cú pháp

```
A B {  
    /* các thuộc tính CSS */  
}
```

- Giải thích
 - A: là phần tử cha (hoặc tổ tiên).
 - B: là phần tử con (hoặc hậu duệ) của A

4.6. Child selector

- Cú pháp

```
A > B {  
    thuộc_tính: giá_trị;
```

}

- Giải thích
 - A là phần tử cha.
 - B là phần tử con trực tiếp của A.
 - Dấu > thể hiện mối quan hệ cha – con trực tiếp, không chọn các phần tử cháu (nằm sâu hơn)

4.7. Adjacent selector

- Bộ chọn anh chị em liền kề (bộ chọn anh em tiếp giáp) trong CSS là bộ chọn dùng để chọn phần tử ngay sau một phần tử khác trong cùng một cấp (cùng cha)
- Cú pháp:

```
A + B {
    /* các thuộc tính CSS */
}
```

- Giải thích
 - A là phần tử trước đó
 - B là phần tử đứng sau A và cấp cao nhất trong HTML cây.

4.8. Attribute selector

- Bộ chọn thuộc tính (bộ chọn thuộc tính) trong CSS được sử dụng để chọn các phần tử HTML dựa trên thuộc tính của chúng (ví dụ như id, class, href, type, vv..)
- Cú pháp cơ bản:

```
[attribute] {
    property: value;
}
```

Bảng 13: Các phổ biến của Bộ chọn thuộc tính

Cú pháp	Ý nghĩa	Ví dụ	Giải thích
[attr]	Chọn phần tử có thuộc tính attr	[required]	chọn tất cả phần tử có thuộc tính required
[attr="value"]	Chọn phần tử có thuộc tính attr chính xác bằng value	[type="text"]	chọn input có type="text"
[attr~="value"]	Chọn phần tử có trong giá trị thuộc tính một từ là value	[class~="btn"]	chọn phần tử có class chứa từ btn

`[attr`	= "value"]`	Chọn phần tử có thuộc tính bắt đầu bằng value hoặc value-	`[lang`
[attr^="value"]	Chọn phần tử có giá trị thuộc tính bắt đầu bằng value	[href^="https"]	chọn liên kết bắt đầu bằng https
[attr\$="value"]	Chọn phần tử có giá trị thuộc tính kết thúc bằng value	[src\$=".jpg"]	chọn ảnh có đuôi .jpg
[attr*="value"]	Chọn phần tử có giá trị thuộc tính chứa value	[title*="học"]	chọn phần tử có từ “học” trong thuộc tính title

4.9. Pseudo class selector

- Bộ chọn lớp giả trong CSS là bộ chọn đặc biệt dùng để chọn các trạng thái cụ thể của phần tử - tức là trạng thái không thể xác định bằng HTML thông thường
- Cú pháp:

```
selector:pseudo-class {
    property: value;
}
```

Bảng 14: Ví dụ về Pseudo class selector

Pseudo-class	Ý nghĩa	Ví dụ
:hover	Khi con trỏ chuột di chuyển qua phần tử	a:hover { color: red; }
:active	Khi phần tử đang được nhấn	button:active { background: green; }
:focus	Khi phần tử (như input) được focus	input:focus { border-color: blue; }
:first-child	Chọn phần tử đầu tiên trong cùng cha	p:first-child { color: red; }
:last-child	Chọn phần tử cuối cùng trong cùng cha	li:last-child { font-weight: bold; }
:nth-child(n)	Chọn phần tử thứ n	tr:nth-child(2) { background: #eee; }
:checked	Áp dụng cho checkbox/radio được chọn	input:checked { accent-color: green; }
:disabled	Áp dụng cho phần tử bị vô hiệu hóa	button:disabled { opacity: 0.5; }

4.10. Group selector

- Bộ chọn nhóm trong CSS được sử dụng để nhóm nhiều bộ (bộ chọn) lại với nhau và áp dụng cùng một loại (kiểu) cho tất cả chúng.
- Cú pháp:

```
selector1, selector2, selector3 {  
    property: value;  
}
```

- Giải thích:
 - Các bộ lựa chọn có thể được giải phóng bằng comma (,)
 - Mỗi phần tử được chọn sẽ chia sẻ cùng một quy tắc CSS .
- Lợi ích:
 - Giúp rút gọn mã hóa , dễ đọc hơn.
 - Tránh lặp lại cùng một đoạn CSS cho nhiều phần tử

5. Đơn vị đo lường CSS

5.1. Khái niệm

Trong CSS, đơn vị đo lường (units) được sử dụng để tính, đặt vị trí, kích cỡ các phần tử trong website

Quy cách viết:

- Giá trị số rồi đến đơn vị
- Không có dấu cách giữa số và đơn vị
- Nếu giá trị là 0 có thể bỏ qua đơn vị tính
- Một số đơn vị có thể sử dụng số âm

5.2. Phân loại đơn vị trong CSS.

CSS có hai nhóm đơn vị chính:

5.2.1. Đơn vị tương đối (Relative units)

- Đặc điểm: Là đơn vị được tính tương đối so với phần tử mẹ hoặc các phần tử khác, tương đối cơ động và hoạt động tốt cho nhiều loại thiết bị, màn hình khác nhau

Bảng 15: Các đơn vị tuyệt đối

Đơn vị	Ý nghĩa
%	Kích thước % tương đối của phần tử con so với phần tử cha mẹ
em	Đơn vị tương đối so với kích thước font-size của phần tử cha mẹ
rem	Đơn vị tương đối dựa vào kích thước font-size của phần tử gốc html
vw	View width 1vw=1/100 chiều rộng khung nhìn
vh	View height 1vh=1/100 chiều cao khung nhìn

vmin	Giá trị nhỏ hơn trong 2 giá trị vw và vh
vmax	Giá trị lớn hơn trong 2 giá trị vw và vh

5.2.2. Đơn vị tuyệt đối (Absolute units)

- Đặc điểm: Là đơn vị có giá trị cố định không thay đổi, gặp vấn đề khi website được hiển thị ở nhiều loại màn hình khác nhau

Bảng 16: Các đơn vị tương đối

Đơn vị	Ý nghĩa
px	1px = 1 inch /96
pt	1pt = 1 inch / 72
pc	1pc = 12pt
cm	Centimeters
mm	Milimeters
in	Inches (1in = 96px = 2.54cm)

6. Kế thừa thuộc tính

- Khái niệm

Kế thừa thuộc tính (Inheritance) trong CSS là cơ chế cho phép các phần tử con tự động nhận (thừa hưởng) một số thuộc tính từ phần tử cha mà không cần khai báo lại.

Bảng 17: Các thuộc tính có thể kế thừa

Thuộc tính	Ý nghĩa
color	Màu chữ
font-family	Kiểu chữ
font-size	Cỡ chữ
font-style	Kiểu chữ (nghiêng, đậm, thường)
font-weight	Độ đậm chữ
line-height	Chiều cao dòng
text-align	Căn lề văn bản
visibility	Ẩn/hiện phần tử

Bảng 18: Các thuộc tính không kế thừa mặc định

Thuộc tính	Ý nghĩa
margin, padding	Khoảng cách
border	Viền
width, height	Chiều rộng và chiều cao
background	Nền
display	Kiểu hiển thị (block, inline, ...)

- Ép kế thừa bằng từ khóa inherit: Nếu muốn bất kỳ thuộc tính nào cũng được kế thừa, sử dụng từ khóa inherit:

```
p {
    border: inherit;
}
```

7. Các nhóm thuộc tính trong CSS

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

Bảng 19: Thuộc tính định dạng font chữ (Font Properties)

Thuộc tính	Ý nghĩa	Ví dụ
font-family	Chọn kiểu phông chữ	font-family: Arial, sans-serif;
font-size	Kích thước chữ	font-size: 16px;
font-style	Kiểu chữ: nghiêng, thường, đậm	font-style: italic;
font-weight	Độ đậm của chữ	font-weight: bold;
font-variant	Chuyển chữ thường thành chữ in hoa nhỏ (small-caps)	font-variant: small-caps;
font	Viết tắt cho tất cả các thuộc tính trên	font: italic bold 16px Arial;

Bảng 20: Thuộc tính định dạng văn bản (Text Properties)

Thuộc tính	Ý nghĩa	Ví dụ
color	Màu chữ	color: blue;
text-align	Căn lề văn bản: trái, phải, giữa, đều	text-align: center;
text-decoration	Trang trí chữ: gạch chân, gạch giữa...	text-decoration: underline;
text-transform	Biến đổi chữ hoa/thường	text-transform: uppercase;
text-indent	Thụt đầu dòng	text-indent: 30px;
line-height	Chiều cao dòng	line-height: 1.5;

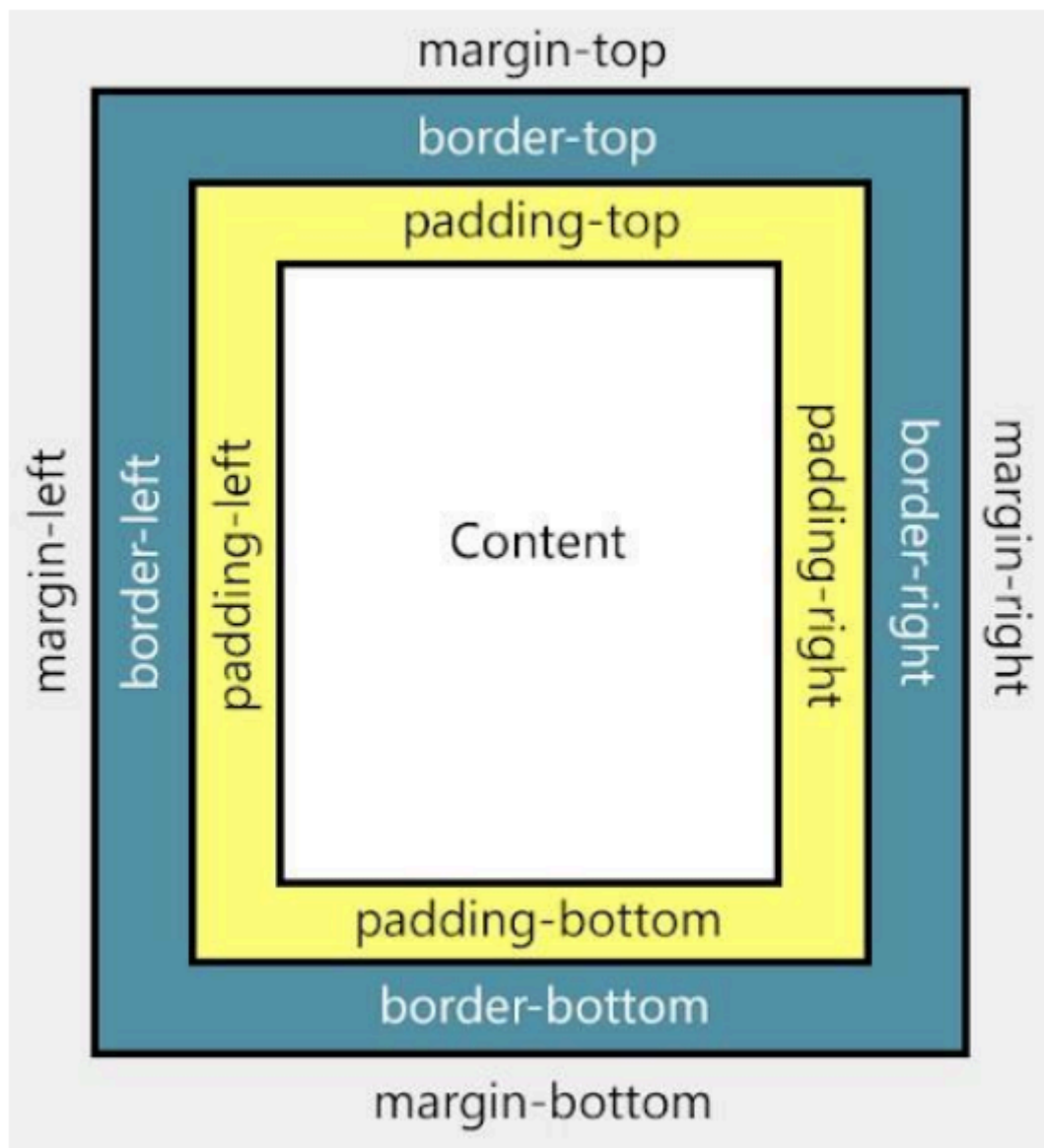
letter-spacing	Khoảng cách giữa các ký tự	letter-spacing: 2px;
word-spacing	Khoảng cách giữa các từ	word-spacing: 5px;
text-shadow	Đổ bóng cho văn bản	text-shadow: 2px 2px 5px gray;

- Thuộc tính định dạng nền (background)

Bảng 21: Thuộc tính định dạng nền (background)

Thuộc tính	Ý nghĩa	Ví dụ
background-color	Màu nền	background-color: lightblue;
background-image	Hình nền	background-image: url('hinhnen.jpg');
background-repeat	Cách lặp lại hình nền	background-repeat: no-repeat;
background-position	Vị trí hình nền	background-position: center top;
background-size	Kích thước hình nền	background-size: cover;
background-attachment	Cố định hay cuộn nền khi trang cuộn	background-attachment: fixed;
background	Viết tắt cho tất cả các thuộc tính trên	background: lightblue url('bg.jpg') no-repeat center/cover;

- Mô hình hộp (box model)
 - Cấu trúc mô hình hộp:



Ảnh 8: Cấu trúc mô hình hộp

Bảng 22: Các thành phần của Box Model

Thành phần	Mô tả	Ví dụ
Content	Vùng chứa nội dung (văn bản, hình ảnh, video,...)	width, height
Padding	Khoảng cách bên trong giữa nội dung và viền	padding: 10px;
Border	Đường viền bao quanh phần tử	border: 2px solid black;
Margin	Khoảng cách bên ngoài giữa các phần tử	margin: 20px;

- Thuộc tính định dạng cách hiển thị (display)

- Cú pháp:

```
selector {
    display: giá_trị;
}
```

Bảng 23: Các giá trị thường dùng của display

Giá trị	Mô tả	Đặc điểm
block	Phần tử hiển thị dạng khối (chiếm toàn bộ chiều ngang)	Bắt đầu trên dòng mới. Có thể đặt width, height, margin, padding.
inline	Hiển thị trên cùng một dòng với phần tử khác	Không xuống dòng. Không thể đặt width/height.
inline-block	Kết hợp inline và block	Trên cùng dòng nhưng vẫn đặt được width, height.
none	Ẩn phần tử hoàn toàn (không chiếm không gian)	Dùng để ẩn/hiện phần tử khi cần.
flex	Biến phần tử cha thành container linh hoạt	Dễ sắp xếp, căn giữa, chia cột hoặc hàng (Flexbox).
grid	Hiển thị phần tử dưới dạng lưới (grid layout)	Phù hợp cho bố cục phức tạp, chia ô.
list-item	Hiển thị như phần tử danh sách ()	Có dấu đầu dòng mặc định.

- Thuộc tính xác định vị trí (position)

- Cú pháp:

```
selector {
    position: value;
    top: ...px;
    left: ...px;
    right: ...px;
    bottom: ...px;
}
```

- Giải thích:

- + position: xác định kiểu định vị.
- + top, left, right, bottom: xác định khoảng cách so với vị trí gốc.

- Các giá trị của position

Bảng 24: Các giá trị của position

Giá trị	Ý nghĩa	Đặc điểm
static	Mặc định (phần tử nằm theo luồng bình thường của trang)	Không thể dùng top, left,...
relative	Định vị tương đối so với vị trí ban đầu của phần tử	Có thể di chuyển phần tử mà vẫn giữ khoảng trống cũ
absolute	Định vị tuyệt đối so với phần tử cha gần nhất có position khác static	Bỏ khỏi luồng, không giữ chỗ
fixed	Cố định vị trí theo cửa sổ trình duyệt	Luôn ở 1 vị trí, kể cả khi cuộn trang
sticky	Kết hợp relative + fixed	Ban đầu di chuyển theo trang, sau đó “dính” lại khi đạt vị trí nhất định

8. Float & Clear

8.1. Thuộc tính float là gì?

- Float trong CSS được dùng để đẩy phần tử sang bên trái hoặc phải trong vùng chứa của nó, cho phép các phần tử khác (như văn bản, hình ảnh) bao quanh (chảy quanh) nó.
- Cú pháp:

```
selector {
    float: left | right | none | inherit;
}
```

Bảng 25: Giá trị của float

Giá trị	Ý nghĩa
left	Đẩy phần tử sang bên trái, nội dung khác sẽ nằm bên phải
right	Đẩy phần tử sang bên phải, nội dung khác nằm bên trái
none	Không trôi sang bên nào (mặc định)
inherit	Kế thừa giá trị float từ phần tử cha

8.2. Thuộc tính clear là gì?

- Clear được dùng để ngăn không cho phần tử nằm cạnh (chảy quanh) các phần tử đã được float trước đó. Nói cách khác, clear giúp “ngắt dòng” — bắt đầu một phần tử bên dưới phần tử float.
- Cú pháp:

```
selector {  
    clear: left | right | both | none;  
}
```

Bảng 26: Giá trị của clear

Giá trị	Ý nghĩa
left	Không cho phần tử nằm cạnh phần tử float bên trái
right	Không cho phần tử nằm cạnh phần tử float bên phải
both	Không cho nằm cạnh bất kỳ phần tử float nào (trái hoặc phải)
none	Cho phép phần tử nằm cạnh float (mặc định)

9. Flex

9.1. Khái niệm

Flex (viết tắt của Flexible Box) là một mô hình bố cục (layout model) trong CSS giúp bạn sắp xếp, căn chỉnh và phân chia không gian giữa các phần tử một cách linh hoạt, hiện đại và dễ dàng hơn so với cách dùng float hoặc position.

9.2. Kích hoạt Flexbox

Đặt display: flex; cho phần tử cha

```
.container {  
    display: flex;  
}
```

Lúc này, các phần tử con trong .container sẽ trở thành flex items

9.3. Các thuộc tính quan trọng của Flex

Bảng 27: Thuộc tính của phần tử cha (container)

Thuộc tính	Chức năng	Ví dụ
display: flex;	Kích hoạt chế độ Flexbox	display: flex;
flex-direction	Xác định hướng sắp xếp các phần tử con	row (ngang), column (dọc)

justify-content	Căn chỉnh theo trục chính (hàng ngang mặc định)	center, space-between, flex-end
align-items	Căn chỉnh theo trục vuông góc (dọc)	center, flex-start, flex-end, stretch
flex-wrap	Cho phép các phần tử xuống hàng khi không đủ chỗ	wrap, nowrap
align-content	Căn chỉnh các hàng khi có nhiều dòng flex	center, space-around, ...

Bảng 28: Thuộc tính của phần tử con (item)

Thuộc tính	Chức năng	Ví dụ
flex	Gộp 3 thuộc tính flex-grow, flex-shrink, flex-basis	flex: 1; (chia đều không gian)
align-self	Căn chỉnh riêng cho từng phần tử	align-self: center;
order	Thay đổi thứ tự hiển thị	order: 2;

10. Grid

10.1. Khái niệm

CSS Grid là hệ thống bố cục (layout) mạnh mẽ trong CSS, cho phép bạn chia trang web thành hàng (rows) và cột (columns) một cách linh hoạt, chính xác và gọn gàng.

10.2. Cú pháp cơ bản

```
.container {
    display: grid; /* Kích hoạt chế độ Grid */
}
```

10.3. Các thuộc tính chính trong CSS Grid

Bảng 29: Thuộc tính của phần tử cha (Grid Container)

Thuộc tính	Mô tả	Ví dụ
display: grid;	Kích hoạt Grid layout	display: grid;
grid-template-columns	Xác định số cột và kích thước mỗi cột	grid-template-columns: 200px 1fr 2fr;
grid-template-rows	Xác định số hàng và kích thước	grid-template-rows: 100px auto;
gap	Khoảng cách giữa các ô (hàng & cột)	gap: 10px;

justify-content	Căn chỉnh toàn bộ grid theo chiều ngang	center, space-between
align-content	Căn chỉnh theo chiều dọc	center, stretch

Bảng 30: Thuộc tính của phần tử con (Grid Item)

Thuộc tính	Mô tả	Ví dụ
grid-column	Chỉ định phần tử chiếm bao nhiêu cột	grid-column: 1 / 3;
grid-row	Chỉ định phần tử chiếm bao nhiêu hàng	grid-row: 1 / 2;
justify-self	Căn ngang riêng cho từng ô	center, end
align-self	Căn dọc riêng cho từng ô	center, end

11. CSS3

11.1. Khái niệm

CSS3 (viết tắt của Cascading Style Sheets – Level 3) là phiên bản thứ 3 của ngôn ngữ CSS, được dùng để mô tả cách trình bày (giao diện, màu sắc, bố cục) của các trang web viết bằng HTML hoặc XHTML.

11.2. CSS3 Selectors

CSS3 Selectors mở rộng bộ chọn thuộc tính trong CSS2

Thêm 3 bộ chọn thuộc tính:

- [attribute^=value]: chọn các thành phần có thuộc tính attribute có giá trị bắt đầu value
- [attribute\$=value]: chọn các thành phần có thuộc tính attribute có giá trị kết thúc value
- [attribute*=value]: chọn các thành phần có thuộc tính attribute có giá trị chứa value

Ví dụ:

p[class^=doan1]: chọn các thẻ p có thuộc tính class bắt đầu bằng doan1

p[class\$=doan1]: chọn các thẻ p có thuộc tính class có giá trị kết thúc bằng doan1

p[class*=doan1]: chọn các thẻ p có thuộc tính class có giá trị chứa doan1

11.3. CSS3 Pseudo-classes

- Mở rộng các lớp bên cạnh các lớp: :hover, :active, :visited, :focus
- :root: chọn phần tử gốc của tài liệu
- :first-child, :last-child: chọn phần tử con đầu tiên hoặc cuối cùng của phần tử cha

- `:not(selector)`: bộ chọn phủ định

Ví dụ:

`p:first-child`: chọn các thẻ `p` là con đầu tiên của thẻ cha chứa `p`

`:not(div)`: chọn các thẻ không phải là thẻ `div`

11.4. CSS3 Borders

CSS3 sử dụng `border-radius` để tạo các góc được bo tròn

`border-radius:10px 20px 5px 40px;(top left, top right, bottom right, bottom left)`

Ảnh 9: CSS3 sử dụng border-radius

11.5. CSS3 border-image

`border-image` được sử dụng để thiết lập một ảnh làm đường bao quanh đối tượng

- Một số thuộc tính cụ thể:
 - `border-image-source`: đường dẫn ảnh làm đường bao sử dụng `url()`
 - `border-image-slice`
 - `border-image-width`: chiều rộng ảnh bao
 - `border-image-outset`: dung lượng phần tràn ra ngoài đường bao
 - `border-image-repeat`: xác định đường bao có lặp lại hay không

11.6. CSS3 Background

Trong CSS3 có thể thêm một hoặc nhiều hình ảnh nền cho phần tử

Các hình nền được phân tách bởi dấu `”,”`

Các hình ảnh được xếp chồng lên nhau trong đó phần tử đầu tiên sẽ gần với góc nhìn nhất

11.7. CSS3 Gradients

CSS 3 hỗ trợ đồ màu tuyến tính: Linear Gradient, Radical Gradient

- Linear Gradient: `background: linear-gradient(direction, color-stop1, color-stop2,...);`
- `linear-gradient()` tạo ra hình ảnh chuyển tiếp màu tuyến tính theo đường thẳng
- Radical Gradient: `background: radial-gradient(shape size at position, start-color,..., last-color);`

background: linear-gradient(45deg, red, blue);
radial-gradient(circle at center, red 0, blue, green 100%)

11.8. CSS3 Fonts

Cho phép chọn lựa kiểu font chữ hiển thị trên trang web

- Font được nhóm vào thành các họ font
- 5 họ font chữ cơ bản:
 - Serif: Times New Roman, Georgia, Garamond, ...
 - Sans-serif: Arial, Verdana, Helvetica, ...
 - Monospace: Courier New, Lucida Console, Monaco, ...
 - Cursive: Brush Script MT, Lucida Handwriting, ...
 - Fantasy: Cooperplate, Papyrus, ...

11.9. CSS3 2D Transforms

Thuộc tính transform cho phép biến đổi đối tượng như di chuyển, xoay, co giãn

- Với thuộc tính transform có thể sử dụng các phương thức sau để biến đổi đối tượng:
 - translate(): dịch chuyển phần tử từ vị trí hiện tại theo chiều X, Y
 - rotate(): xoay phần tử theo chiều kim đồng hồ (đơn vị deg)
 - scaleX(): tăng hoặc giảm chiều rộng của đối tượng
 - scale(): tăng hoặc giảm kích thước của đối tượng
 - skewX(): nghiêng đối tượng theo trục X (đơn vị deg)
 - skewY(): nghiêng đối tượng theo trục Y (đơn vị deg)
 - skew(): nghiêng đối tượng theo trục X và Y một góc cho trước
 - matrix(): tập hợp tất cả các phép chuyển đổi đối tượng trong một phương thức

matrix(scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY())

```
div {
  width: 300px;
  height: 120px;
  background-color: red;
  border: 1px solid black;
  padding: 5px;
  border-radius: 10px;
  text-align: center;
  color: white;
}
div:hover {
  transition-delay: 250ms;
  transform: rotate(180deg);
}
```

Xoay tròn nếu di chuột vào

Ảnh 10: CSS3 2D Transforms

11.10. CSS3 3D Transforms

- CSS hỗ trợ các phép biến đổi 3 chiều
- Hỗ trợ các phương thức biến đổi ba chiều để hỗ trợ thêm chiều sâu Z
- Thêm các phương thức biến đổi chiều sâu: rotate(), translate(), scaleZ(), ...
- Chuyển tiếp CSS cho phép thay đổi các giá trị thuộc tính CSS một cách trơn tru hơn trong một khoảng thời gian
- Một số thuộc tính của CSS transition:
 - transition: transition: <property> <duration> <timing-function> <delay>;
 - transition-delay: khoảng thời gian chờ giữa thời gian thuộc tính được thay đổi và chuyển tiếp xảy ra
 - transition-duration: định nghĩa khoảng thời gian chuyển tiếp xảy ra
 - transition-property: định nghĩa thuộc tính CSS sẽ áp dụng transition
 - transition-timing-function: định nghĩa phương thức xác định các giá trị tức thời của thuộc tính được tính toán

```
.box {
  border-style: solid;
  border-width: 1px;
  display: block;
  width: 200px;
  height: 100px;
  background-color: red;
  transition: width 2s, height 2s, background-color 2s, rotate 2s;
  color: white;
  text-align: center;
  padding: 5px;
  border-radius: 10px;
}
.box:hover {
  background-color: orange;
  width: 300px;
  height: 200px;
  rotate: 180deg;
}
```

Ví dụ transition

Ảnh 11: CSS3 3D Transforms

11.11. CSS3 Multiple Columns

- CSS3 cho phép định nghĩa layout nhiều cột cho trang web
- Một số thuộc tính của layout nhiều cột:
 - column-count: số cột của đối tượng
 - column-gap: khoảng cách các cột
 - column-rule-style: style của thước phân cách
 - column-rule-width: độ rộng của thước
 - column-rule-color: màu sắc của vạch phân cách
 - column-rule: thuộc tính tổng hợp của width, style, color
 - column-span: số cột mà đối tượng sẽ tràn ra
 - column-width: định nghĩa kích thước tối ưu của cột

11.12. CSS3 Animations

- CSS cho phép tạo hoạt cảnh của các đối tượng trên trang web mà không phải sử dụng Javascript hay Flash
- Các thuộc tính CSS hỗ trợ tạo hoạt cảnh:
 - @keyframes: khai báo một tập các frame hoạt cảnh được đặt một tên
 - animation-name: tên của @keyframes sẽ sử dụng cho đối tượng
 - animation-duration: thời gian animation kết thúc
 - animation-delay: thời gian trễ trước khi animation bắt đầu
 - animation-iteration-count: số lần animation chạy, giá trị infinite là vô hạn lần
 - animation-direction: chiều của hoạt cảnh qua các khung hình với các giá trị normal (tiến), reverse (lùi), alternate (trước tiến sau đó lùi), alternate-reverse (trước lùi rồi tiến)
 - animation-timing-function: định nghĩa đường cong tốc độ của hoạt cảnh: ease (bắt đầu chậm sau đó nhanh và kết thúc chậm), linear, ease-in, ease-out, ease-in-out, cubic-bezier, ...
- Các thuộc tính CSS hỗ trợ tạo hoạt cảnh:
 - animation-fill-mode: hoạt cảnh không tác động lên đối tượng trước và sau hoạt cảnh có thể chỉnh bằng các giá trị none (mặc định), forwards (giữ lại giá trị style cuối cùng trong hoạt cảnh), backwards (giữ lại style đầu tiên trong hoạt cảnh), both (giữ lại cả 2)
 - Animation: thuộc tính gộp theo thứ tự animation-name, animation-duration, animation-timing-function, animation-delay, animation-iteration-count, animation-direction
- @keyframes định nghĩa một hoạt cảnh trong đó có các trạng thái được chuyển đổi

- Khi định nghĩa các style trong một hoạt cảnh có thể dùng từ khoá from (0%) và to (100%) hoặc các giá trị % để khai báo các style, 0% là bắt đầu, 100% là kết thúc

```
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: dennhapnhay;
  animation-duration: 4s;
}

@keyframes dennhapnhay {
  0% {background-color: red;}
  25% {background-color: blue;}
  50% {background-color: green;}
  75% {background-color: yellow;}
  100% {background-color: orange;}
}
```

CSS Animation



Ảnh 12: CSS3 Animations

Giải thích:

- @keyframes blinkBox: Animation sẽ thay đổi màu nền (background-color) từ đỏ sang xanh dương và thay đổi độ mờ (opacity) từ 1 đến 0.5, sau đó quay lại trạng thái ban đầu
- .blinking-box: Class này tạo ra một hộp với kích thước 200x200 pixel, viền đen dày 5px. Nó sử dụng animation để áp dụng hiệu ứng nhấp nháy trong 2 giây và lặp lại vô hạn (infinite).
- Các thuộc tính khác: display: flex, justify-content, và align-items được dùng để căn giữa văn bản bên trong hộp.

11.13. CSS3 Shadow

- Hiệu ứng đổ bóng: box-shadow
- Xác định độ rộng, độ cao, blur, offset và màu cho hiệu ứng đổ bóng
 - box-shadow: none|h-offset v-offset blur spread color |inset|initial|inherit;
- Bóng văn bản: text-shadow
- Xác định vị trí trên bên trái, blur và màu sắc của bóng văn bản

12. SCSS

12.1. Khái niệm

SCSS là một ngôn ngữ mở rộng của CSS thuộc Sass (Syntactically Awesome Style Sheets).

Nó được thiết kế để viết CSS dễ quản lý, nhanh hơn và có cấu trúc hơn, đặc biệt với các dự án lớn.

Hiểu đơn giản: SCSS giúp CSS trở nên mạnh mẽ, có tính modular, và dễ bảo trì.

12.2. Các tính năng nổi bật của SCSS

Bảng 31: Các tính năng nổi bật của SCSS

Tính năng	Chức năng	Ví dụ
Biến (Variables)	Lưu giá trị để dùng nhiều lần	<code>\$primary-color: #ff4757;</code>
Nested (Lồng thẻ)	Viết CSS lồng nhau, giống cấu trúc HTML	<code>nav { ul { li { color: red; } } }</code>
Mixin	Tạo nhóm CSS dùng lại	<code>@mixin rounded { border-radius: 10px; }</code>
Inheritance	Kế thừa thuộc tính	<code>%button { padding: 10px; } .btn { @extend %button; }</code>
Operators	Tính toán giá trị CSS	<code>width: 100% / 3;</code>
Functions	Hàm xử lý giá trị	<code>lighten(\$color, 20%), darken(\$color, 10%)</code>
Control directives	Vòng lặp, điều kiện	<code>@if, @for, @each, @while</code>

12.3. Lợi ích khi dùng SCSS

- Tiết kiệm thời gian: Không cần viết đi viết lại các giá trị giống nhau.
- Dễ quản lý: Lồng thẻ, mixin, biến giúp CSS có cấu trúc hơn.
- Dễ mở rộng: Thêm tính năng mới mà không làm rối CSS.
- Tăng tính tái sử dụng: Sử dụng mixin, inheritance và biến cho nhiều component.
- Hỗ trợ dự án lớn: Dễ chia nhỏ file SCSS (partials) và import (@import / @use).

12.4. Kết luận

SCSS là công cụ mạnh mẽ giúp viết CSS hiệu quả, dễ quản lý và mở rộng, đặc biệt cho các dự án web lớn.

Khi kết hợp với CSS3 (Transitions, Animations, Transforms, Flexbox, Grid...) SCSS giúp tạo ra giao diện hiện đại, chuyên nghiệp và dễ bảo trì.

13. SASS

13.1. Khái niệm

Sass là một ngôn ngữ mở rộng của CSS, được thiết kế để viết CSS mạnh mẽ, có cấu trúc, dễ quản lý và tái sử dụng.

Nói cách khác, Sass là tiền thân của SCSS, cung cấp các tính năng như biến, mixin, nested rules, inheritance, vòng lặp và điều kiện.

Hiểu đơn giản: Sass giúp CSS sạch hơn, ngắn gọn hơn và dễ bảo trì, đặc biệt cho các dự án web lớn.

13.2. Hai cú pháp của Sass

- SCSS:

Cú pháp giống CSS thông thường, có thể dùng file .scss

Hỗ trợ tất cả tính năng Sass

Ví dụ:

```
$primary-color: #3498db;
body {
    background-color: $primary-color;
}
```

- Indented Sass (Sass cổ điển)

Dùng thụt đầu dòng thay cho {} và ;

File có đuôi .sass

Ví dụ:

```
$primary-color: #3498db
```

```
body
    background-color: $primary-color
```

13.3. Các tính năng nổi bật của Sass

Bảng 32: Các tính năng nổi bật của SASS

Tính năng	Chức năng	Ví dụ
Variables (Biến)	Lưu giá trị dùng nhiều lần	\$main-color: #ff4757;
Nesting (Lồng thẻ)	Lồng CSS theo cấu trúc HTML	nav { ul { li { color: red } } }
Mixin	Tạo nhóm CSS tái sử dụng	@mixin rounded { border-radius: 10px; }
Inheritance / Extend	Kế thừa CSS	%btn { padding: 10px; } .button { @extend %btn; }
Operators	Tính toán CSS	width: 100% / 3;
Functions	Hàm xử lý giá trị	lighten(\$color, 20%)
Control directives	Vòng lặp, điều kiện	@if, @for, @each, @while
Partials & Import	Chia file CSS nhỏ và import	_header.scss, _footer.scss và @imp

13.4. Lợi ích của Sass

- Lợi ích của Sass
- Tái sử dụng code: Biến, mixin, extend giúp không viết lại CSS.
- Quản lý dễ dàng: Nesting và partials giúp CSS có cấu trúc.
- Giảm lỗi CSS: Các hàm, operators và control directives giúp tính toán giá trị CSS.
- Hỗ trợ dự án lớn: Chia nhỏ file và import giúp dễ bảo trì.

13.5. Kết luận

Sass giúp CSS mạnh mẽ, linh hoạt, dễ quản lý và tái sử dụng, đặc biệt hữu ích cho các dự án web lớn và phức tạp.

Khi kết hợp với CSS3 (Transitions, Animations, Transforms, Flexbox, Grid), Sass giúp thiết kế web hiện đại, chuyên nghiệp và dễ bảo trì.

14. Kết luận

- CSS (Cascading Style Sheets)
 - Là ngôn ngữ dùng để định dạng, trình bày giao diện của các trang web HTML.
 - Giúp tách biệt nội dung và kiểu dáng, làm web dễ quản lý và bảo trì.
 - Các chức năng cơ bản: màu sắc, font chữ, khoảng cách, bố cục cơ bản.
- CSS3
 - Là phiên bản nâng cấp của CSS, được bổ sung nhiều tính năng mới mạnh mẽ hơn.
 - Các tính năng nổi bật:
 - + Border-radius: bo tròn góc.
 - + Box-shadow & text-shadow: tạo bóng cho khối và chữ.
 - + Gradients: tạo màu chuyển dần.
 - + Transform (2D & 3D): xoay, nghiêng, phóng to/thu nhỏ phần tử.
 - + Transition & Animation: tạo hiệu ứng chuyển động mượt mà.
 - + Media Queries: hỗ trợ thiết kế responsive (thích ứng mọi thiết bị).
 - CSS3 giúp web hiện đại, sinh động và tương tác tốt hơn so với CSS truyền thống.

CSS là nền tảng để trang trí web, CSS3 là phiên bản nâng cao, mang lại tính thẩm mỹ, tương tác và hiệu ứng phong phú cho trang web.

CSS3 vẫn dựa trên CSS cơ bản, nên học CSS trước rồi mới học CSS3 sẽ hiệu quả hơn.

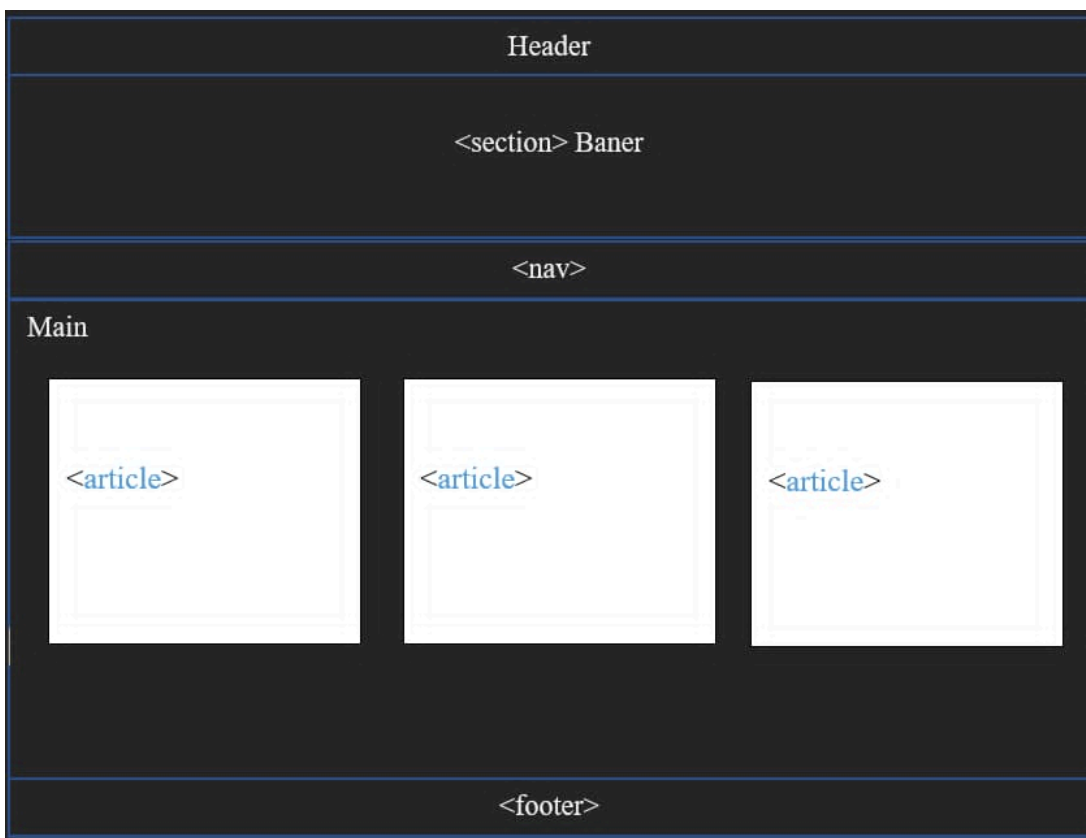
CHƯƠNG 4: THIẾT KẾ WEBISTE

1. Ý tưởng của Website

- Thiết kế Web bán hàng điện tử.
- Sản phẩm là Laptop, PC, điện thoại, các phụ kiện điện tử.
- Trang Web có những chức năng như: mua hàng, đặt hàng, thanh toán, đặt lịch bảo hành, đặt lịch tư vấn mua thiết bị.
- Chức năng đăng kí, đăng nhập,...

2. Xây dựng bố cục của trang Web

Trang Web được xây dựng thành các thành phần chia theo từng chức năng của trang Web như: Header, Banner, nav, Main, section,...



Ảnh 13: Bố cục trang web cơ bản

- Thẻ <haeder>: Gồm Đăng ký người dùng và Đăng nhập kèm theo Chọn ngôn ngữ
- Thẻ Baner: Hình ảnh đại diện và Logo, tên trang Web.
- NAV: Thanh điều hướng gồm: Trang chủ, sản phẩm, giới thiệu, liên hệ, đăng nhập, đăng ký
- Main: Các danh mục sản phẩm
- <footer>: <p>© 2025 ElectroShop | Chất lượng – Uy tín – Giá tốt</p>

3. Thiết kế trang Web bằng HTML và CSS

3.1. Thiết kế phần <header>

3.1.1. HTML

```
9      <header>
10      |   <h1>ElectroShop</h1>
11      |   <p>Cửa hàng thiết bị điện tử hiện đại và uy tín</p>
12      </header>
13
```

Ảnh 14: Thiết kế HTML phần <header>

- <header>: Là một thẻ chứa, dùng để nhóm các nội dung giới thiệu hoặc điều hướng ở đầu trang.
- <h1>ElectroShop</h1>: Tạo ra một tiêu đề chính, lớn nhất trên trang, với nội dung là "ElectroShop".
- <p>Cửa hàng...</p>: Tạo ra một đoạn văn bản mô tả ngắn gọn hoặc khẩu hiệu cho cửa hàng, nằm ngay dưới tiêu đề chính.

3.1.2. CSS

```
15  /* --- HEADER --- */
16  header {
17      position: fixed;
18      top: 0;
19      left: 0;
20      width: 100%;
21      background-color: #002b5c;
22      color: white;
23      text-align: center;
24      padding: 15px 0;
25      z-index: 1000;
26  }
27
```

Ảnh 15: Thiết kế CSS phần <header>

3.1.3. Kết quả



Ảnh 16: Kết quả phần <header>

3.2. Thiết kế thành phần <nav>

3.2.1. HTML

```
14     <nav>
15         <div class="nav-container">
16             <div class="nav-links">
17                 <a href="index.html">Trang chủ</a>
18                 <a href="products.html">Sản phẩm</a>
19                 <a href="about.html">Giới thiệu</a>
20                 <a href="contact.html">Liên hệ</a>
21             </div>
22             <div class="user-links">
23                 <a href="#">Đăng nhập</a>
24                 <a href="#">Đăng ký</a>
25             </div>
26         </div>
27     </nav>
28
```

Ảnh 17: Thiết kế HTML phần <nav>

- <nav>: Là thẻ chứa toàn bộ thanh điều hướng.
- Bên trong được chia làm hai khu vực chính:
 - <div class="nav-links">: Chứa các liên kết chính để di chuyển giữa các trang của website như "Trang chủ", "Sản phẩm", "Giới thiệu", và "Liên hệ".
 - <div class="user-links">: Chứa các liên kết dành cho người dùng như "Đăng nhập" và "Đăng ký".

3.2.2. CSS

```
28  /* --- NAV --- */
29  nav {
30      position: fixed;
31      top: 80px;
32      left: 0;
33      width: 100%;
34      background-color: #004080;
35      padding: 10px 0;
36      z-index: 999;
37  }
38
39  .nav-container {
40      width: 90%;
41      margin: auto;
42      display: flex;
43      justify-content: space-between;
44      align-items: center;
45  }
46
47  .nav-links {
48      display: flex;
49      gap: 20px;
50  }
51
52  .nav-links a {
53      color: white;
54      text-decoration: none;
55      font-weight: bold;
56  }
57
58  .nav-links a:hover {
59      color: #ffcc00;
60  }
61
```

Ảnh 18: Thiết kế CSS phần <nav>

3.2.3. Kết quả

Ảnh 19: Kết quả phần <nav>

3.3. Thiết kế thành phần <section>

3.3.1. HTML

```
29 <section class="banner">
30 | <marquee>🎉 Chào mừng bạn đến với ElectroShop - Giảm giá 15% cho tất cả sản phẩm trong tháng này! 🎉</marquee>
31 </section>
32
```

Ảnh 20: Thiết kế HTML phần <section>

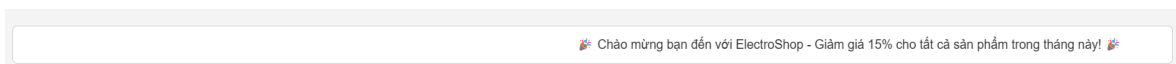
- <section class="banner">: Tạo một khu vực riêng biệt trên trang để chứa banner.
- <marquee>...</marquee>: Đây là một thẻ HTML cũ, có tác dụng làm cho nội dung văn bản bên trong nó di chuyển (chạy) từ bên này sang bên kia màn hình.

3.3.2. CSS

```
78 /* --- Banner --- */
79 .banner {
80     background-color: white;
81     text-align: center;
82     margin: 10px;
83     padding: 10px;
84     border-radius: 6px;
85     border: 1px solid #ccc;
86 }
87
```

Ảnh 21: Thiết kế CSS phần <section>

3.3.3. Kết quả



Ảnh 22: Kết quả phần <section>

3.4. Thiết kế thành phần <main>

3.4.1. HTML

```
33 <main>
34 <h2>Sản phẩm nổi bật</h2>
35 <div class="product-grid">
36
37 <!-- 9 sản phẩm -->
38 <div class="product-card">
39 Laptop Dell XPS 13</h3>
41 <p>Core i7, RAM 16GB</p>
42 <p><strong>25.900.000</strong></p>
43 <button>Mua ngay</button>
44 </div>
45 <div class="product-card">
46 iPhone 14 128GB</h3>
48 <p>OLED 6.1", Camera kép</p>
49 <p><strong>20.990.000</strong></p>
50 <button>Mua ngay</button>
51 </div>
52 <div class="product-card">
53 Sony WH-1000XM5</h3>
55 <p>Chống ồn chủ động</p>
56 <p><strong>8.490.000</strong></p>
57 <button>Mua ngay</button>
58 </div>
59 <div class="product-card">
60 Apple Watch SE</h3>
62 <p>Màn hình Retina, GPS</p>
63 <p><strong>6.990.000</strong></p>
64 <button>Mua ngay</button>
65 </div>
```

Ảnh 23: Thiết kế HTML phần <main>

- <main>: Là thẻ chứa toàn bộ nội dung chính của trang.
- <h2>Sản phẩm nổi bật</h2>: Tạo một tiêu đề lớn cho khu vực này.
- <div class="product-grid">: Là một hộp chứa lớn, dùng để sắp xếp tất cả các sản phẩm theo dạng lưới.
- <div class="product-card">: Mỗi khối này là một "thẻ sản phẩm", hiển thị thông tin cho một mặt hàng. Bên trong mỗi thẻ có:
 - : Hình ảnh của sản phẩm.
 - <h3>: Tên sản phẩm.
 - <p>: Các đoạn mô tả ngắn và giá tiền (được in đậm).
 - <button>: Nút "Mua ngay" để người dùng tương tác.

3.4.2. CSS

```
/* --- Main --- */
main {
  background-color: white;
  margin: 10px;
  padding: 15px;
  border-radius: 6px;
}
```

Ảnh 24: Thiết kế CSS phần <main>

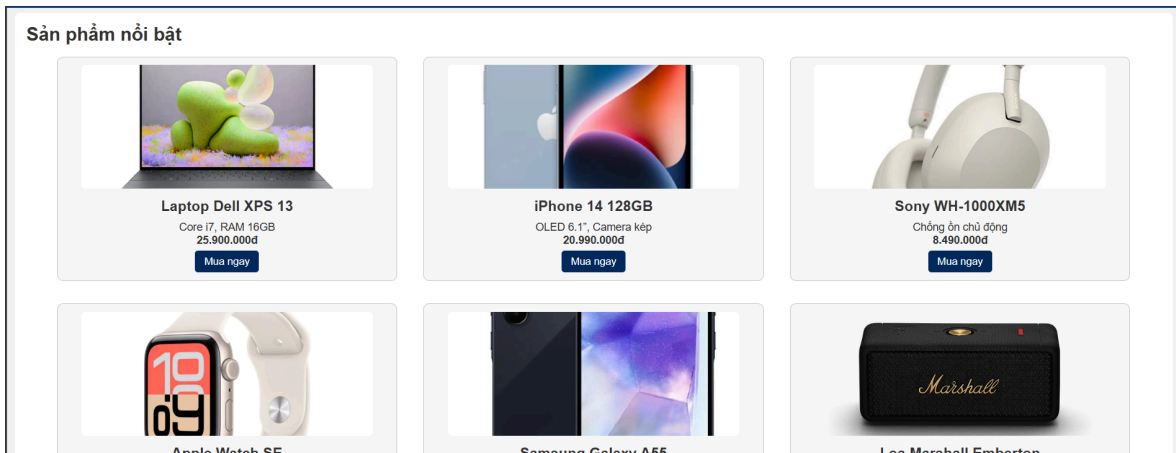
```

108  /* --- Card sản phẩm --- */
109  .product-grid {
110    text-align: center;
111  }
112
113  .product-card {
114    display: inline-block;
115    width: 30%;
116    margin: 1%;
117    background-color: #f9f9f9;
118    border: 1px solid #ccc;
119    border-radius: 8px;
120    padding: 10px;
121    vertical-align: top;
122  }
123
124  .product-card img {
125    width: 90%;
126    height: 160px;
127    object-fit: cover;
128    border-radius: 5px;
129  }
130
131  .product-card h3 {
132    margin: 8px 0;
133  }
134
135  .product-card p {
136    font-size: 14px;
137  }
138
139  .product-card button {
140    margin-top: 5px;
141    padding: 6px 12px;
142    background-color: #002b5c;
143    color: white;
144    border: none;

```

Ảnh 25: Thiết kế CSS phần Product

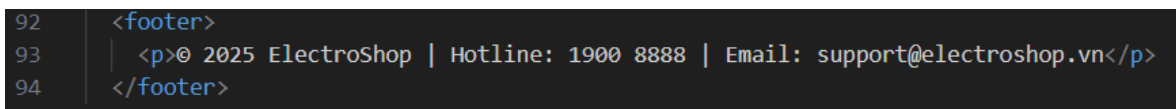
3.4.3. Kết quả



Ảnh 26: Kết quả phân <main>

3.5. Thiết kế thành phần <footer>

3.5.1. HTML



Ảnh 27: Thiết kế HTML phần <footer>

- <footer>: Là thẻ chứa các thông tin ở cuối cùng của trang.
- <p>...</p>: Tạo một dòng văn bản chứa các thông tin liên hệ và bản quyền, bao gồm:
 - Thông tin bản quyền © 2025 ElectroShop.
 - Số điện thoại hỗ trợ (Hotline).
 - Địa chỉ email liên hệ.

3.5.2. CSS

```
96 footer {
97     position: fixed;
98     bottom: 0;
99     left: 0;
100    width: 100%;
101    background-color: #002b5c;
102    color: white;
103    text-align: center;
104    padding: 10px;
105    font-size: 14px;
106 }
107
```

Ảnh 28: Thiết kế CSS phần <footer>

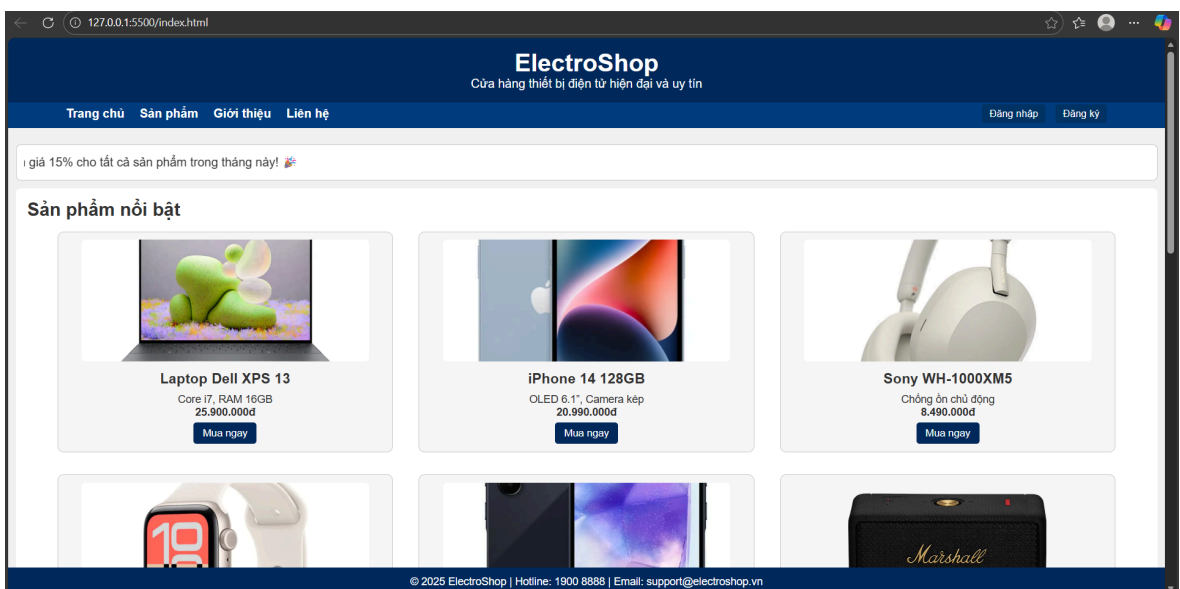
3.5.3. Kết quả



Ảnh 29: Kết quả phần <footer>

4. Kết quả đạt được

4.1. Trang chủ



Ảnh 30: Giao diện trang chủ

Đây là trang chính của website ElectroShop. Trong trang chính này, người dùng có thể đăng nhập (hoặc đăng ký khi chưa có tài khoản) ở góc trên bên phải

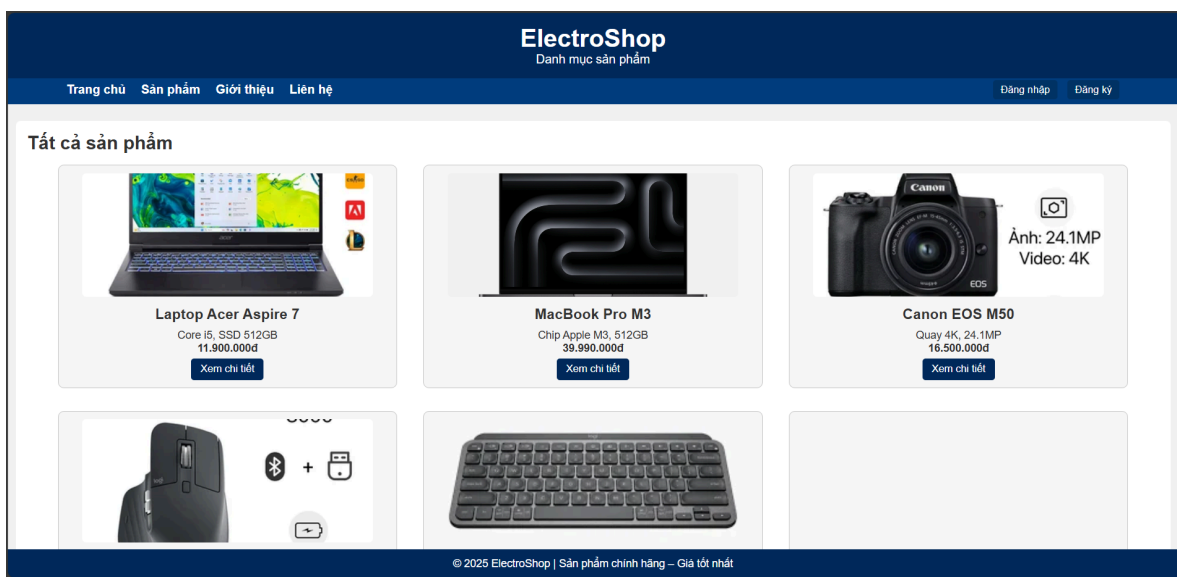
màn hình. Bên trái chính là các danh mục chính của web, bao gồm: Trang chủ, Sản phẩm, Giới thiệu, Liên hệ.

Ở phần banner có 1 câu chạy từ phải sang trái: “🎉 Chào mừng bạn đến với ElectroShop - Giảm giá 15% cho tất cả sản phẩm trong tháng này! 🎉” - hiển thị thông tin giảm giá của web

Bên dưới là các sản phẩm hiện có trong cửa hàng, bao gồm tên sản phẩm, giá tiền, nút mua hàng.

Cuối cùng là thông tin cơ bản của ElectroShop.

4.2. Trang sản phẩm



Ảnh 31: Giao diện trang sản phẩm

Đây là trang liệt kê tất cả sản phẩm hiện tại đang có trong ElectroShop. Mỗi sản phẩm được trình bày tương đương như ở Trang chủ, tuy nhiên thay vì mua hàng, ở đây có nút “Xem chi tiết” để xem chi tiết từng sản phẩm.

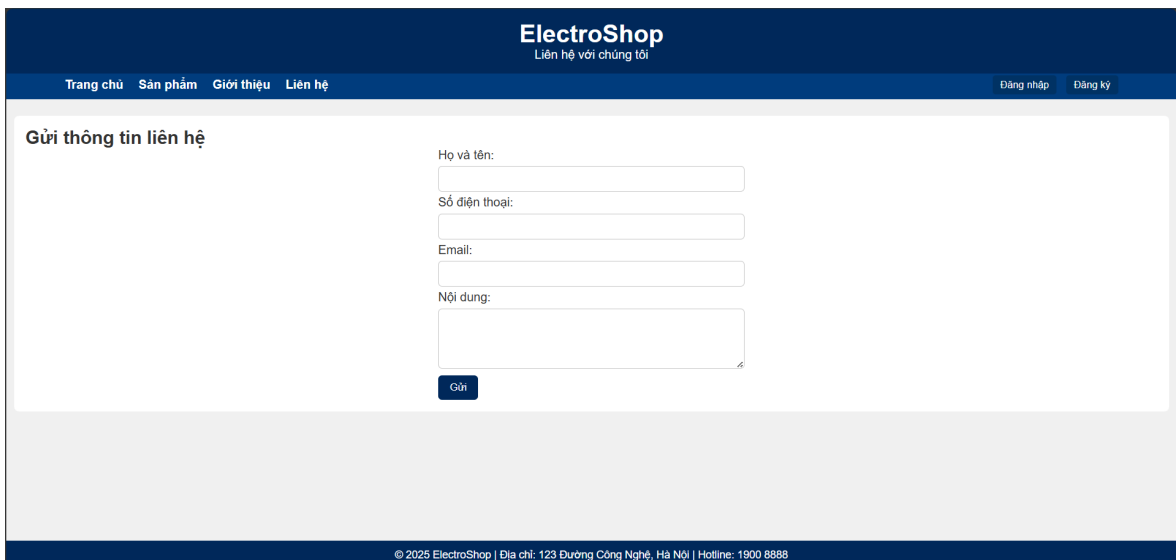
4.3. Trang giới thiệu



Ảnh 32: Giao diện trang giới thiệu

Đây là trang giới thiệu đơn giản về ElectroShop, bao gồm các thông tin cơ bản như: tầm nhìn, sứ mệnh, giá trị cốt lõi, đội ngũ và chính sách khách hàng.

4.4. Trang liên hệ



Ảnh 33: Giao diện trang liên hệ

Đây là trang đơn giản để người dùng gửi thông tin liên hệ về cho ElectroShop. Thông tin cần điền bao gồm: Họ và tên, Số điện thoại, Email, Nội dung.

5. Kết luận

Việc xây dựng website bán đồ điện tử sử dụng HTML5 đã giúp chúng tôi hiểu rõ cách tổ chức cấu trúc và trình bày nội dung trên web. Thông qua đề tài này, em đã áp dụng các kiến thức về HTML5, CSS3 và các thẻ mới để tạo ra một trang web gọn gàng, trực quan và thân thiện với người dùng.

Website được thiết kế với các tính năng cơ bản như:

- Trang chủ giới thiệu sản phẩm nổi bật.
- Danh mục sản phẩm phân loại rõ ràng, dễ tìm kiếm.
- Chi tiết sản phẩm cung cấp hình ảnh và thông tin đầy đủ.
- Giỏ hàng và form liên hệ cơ bản giúp trải nghiệm người dùng tốt hơn.

Ngoài ra, việc sử dụng HTML5 còn giúp website hỗ trợ đa phương tiện, responsive trên nhiều thiết bị, đồng thời tạo nền tảng vững chắc cho việc học nâng cao về JavaScript, PHP hoặc các framework hiện đại trong tương lai.

KẾT LUẬN

Việc lựa chọn HTML5 và CSS3 để xây dựng website bán đồ điện tử "ElectroShop" mang lại nhiều ưu điểm cơ bản và quan trọng. Đầu tiên, cấu trúc trang web trở nên gọn gàng, dễ đọc và dễ bảo trì nhờ vào việc sử dụng các thẻ ngữ nghĩa mới của HTML5 như `<header>`, `<nav>`, `<section>` và `<footer>`. Điều này không chỉ tối ưu hóa công cụ tìm kiếm (SEO) mà còn tạo nền tảng vững chắc cho việc phát triển về sau. Thứ hai, website đã được thiết kế để có khả năng Responsive (thân thiện với đa thiết bị), đảm bảo giao diện hiển thị trực quan và trải nghiệm người dùng tốt trên cả máy tính để bàn, máy tính bảng và điện thoại di động, đáp ứng xu hướng truy cập di động hiện nay. Cuối cùng, việc tập trung vào ngôn ngữ đánh dấu và tạo kiểu giúp đội ngũ phát triển nắm vững được cốt lõi của phát triển giao diện (Front-end), làm chủ được các kỹ thuật bố cục hiện đại, tạo tiền đề vững chắc cho việc tích hợp các công nghệ phức tạp hơn trong tương lai.

Bên cạnh những ưu điểm, đề tài vẫn còn tồn tại một số hạn chế do phạm vi và mục tiêu ban đầu của bài tập lớn. Hạn chế lớn nhất là website hiện tại mới chỉ là Web Tĩnh, thiếu khả năng xử lý động và tương tác phức tạp với người dùng. Các chức năng quan trọng như đăng nhập/dăng ký tài khoản, xử lý thanh toán, lưu trữ dữ liệu sản phẩm và giỏ hàng một cách liên tục (Persistent Storage), hay tìm kiếm nâng cao đều chưa được tích hợp do chưa sử dụng ngôn ngữ lập trình Back-end (như PHP, Node.js) và cơ sở dữ liệu (Database). Ngoài ra, mặc dù đã sử dụng CSS3, giao diện và hiệu ứng tương tác (animations/transitions) vẫn còn tương đối đơn giản, cần được trau chuốt và tối ưu hóa hơn nữa về mặt thẩm mỹ và trải nghiệm người dùng để cạnh tranh với các sàn thương mại điện tử chuyên nghiệp.

Để nâng cấp "ElectroShop" trở thành một hệ thống thương mại điện tử hoàn chỉnh, có ba hướng phát triển chính cần được ưu tiên. Thứ nhất, triển khai tầng logic động bằng cách tích hợp JavaScript/Framework (như React/Vue/Angular) cho Front-end và một ngôn ngữ Back-end mạnh mẽ (như Node.js hoặc Python/Django) kết hợp với Cơ sở dữ liệu (ví dụ: MongoDB hoặc Firestore). Điều này sẽ cho phép xử lý đơn hàng, quản lý người dùng và cập nhật sản phẩm theo thời gian thực. Thứ hai, tập trung vào việc cá nhân hóa trải nghiệm khách hàng thông qua việc xây dựng hệ thống gợi ý sản phẩm dựa trên lịch sử duyệt web và mua hàng. Thứ ba, mở rộng tích hợp các API thanh toán và giao hàng từ các đối tác thứ ba (ví dụ: PayPal, Momo, Viettel Post) để hoàn thiện quy trình mua hàng trực tuyến, biến website từ

một bản trình bày kỹ thuật thành một ứng dụng kinh doanh thực tế, sẵn sàng cho triển khai và vận hành.

TÀI LIỆU THAM KHẢO

- [1].ThS. Lê Văn Phong (2025), *Bài giảng Thiết kế Web và triển khai hệ thống phần*, Khoa CNTT – Trường Đại học Đại Nam.
- [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.