**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN MÔN HỌC: GIỚI THIỆU VỀ QUANG CẢNH CỦA CHÙA HƯƠNG**

**TÊN ĐỀ TÀI: : GIỚI THIỆU VỀ QUANG CẢNH CỦA CHÙA HƯƠNG**

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

**Sinh viên thực hiện:**  **Phan Tự Nghiệp**

**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 SHOP BÁN GIÀY**

**TÊN ĐỀ TÀI: THIẾT KẾ WEB SHOP BÁN GIÀY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020506 | Phan Tự Nghiệp | 29/08/2005 |  |  |

### 

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

**Hà Nội, 2024**

**LỜI NÓI ĐẦU**

Chào mừng đến với trang web của chúng tôi!

Ở đây, chúng tôi tự hào giới thiệu một không gian trực tuyến đầy màu sắc và đa dạng, nơi bạn có thể khám phá những thông tin hữu ích, trải nghiệm mới lạ và nhận được những lời khuyên chân thành về nhiều chủ đề khác nhau.

Trang web của chúng tôi được xây dựng với mục tiêu mang lại cho bạn những trải nghiệm tuyệt vời nhất khi bạn đặt chân vào thế giới trực tuyến. Từ việc cung cấp thông tin bổ ích về du lịch, ẩm thực, văn hóa, đến việc chia sẻ kiến thức và kỹ năng trong các lĩnh vực đa dạng như nghệ thuật, khoa học, công nghệ và nhiều hơn nữa.

Chúng tôi hy vọng rằng trang web của chúng tôi sẽ trở thành điểm đến tin cậy của bạn, nơi bạn có thể tham khảo thông tin, chia sẻ ý kiến và kết nối với cộng đồng trực tuyến rộng lớn.

Hãy cùng nhau khàm phấ và trải nghiệm trang wed của chúng tôi nha

Xin cảm ơn!

MỤC LỤC

[**CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM** 11](#_Toc10053)

[1.1. Các khái niệm cơ bản 11](#_Toc20650)

[*1.1.1. Thiết kế Web (Web Design)* 11](#_Toc31129)

[*1.1.2. Triển khai Hệ thống Phần mềm* 11](#_Toc2695)

[*1.1.3. Các công nghệ và ngôn ngữ phổ biến* 12](#_Toc24834)

[1.2. Phân biệt Web tĩnh, Web động 12](#_Toc3569)

[*1.2.1. Web tĩnh (Static Web)* 12](#_Toc20851)

[*1.2.2. Web động (Dynamic Web)* 13](#_Toc21933)

[*1.2.3. Sự phân biệt:* 13](#_Toc5037)

[1.3. Một số thuật ngữ (Hosting, Tên miền-Domain Name-Web Server Name, HomePage, Web Site, WebServer, URL-Uniform Resource Location, Browser, …) 14](#_Toc5335)

[1.4. Một số công cụ dùng thiết kế Web (Visual Studio code, Adobe Dreamweaver, …) 15](#_Toc19929)

[**CHƯƠNG II: HTML & HTML5** 17](#_Toc15272)

[2.1. Tổng quan về HTML 17](#_Toc28137)

[2.2. Cấu trúc tổng quát trang HTML 17](#_Toc18570)

[2.3. Các thẻ HTML thông dụng 17](#_Toc14265)

[2.4. Các thẻ tạo biểu mẫu (form) 18](#_Toc14580)

[2.5. Một số thẻ HTML đặc biệt 19](#_Toc13298)

[2.6. HTML5 20](#_Toc12442)

[**CHƯƠNG III: CSS và CSS3** 24](#_Toc29863)

[3.1. CSS là gì? 24](#_Toc12708)

[3.2. Cú pháp CSS 24](#_Toc31153)

[3.3. Áp dụng CSS vào trang HTML 24](#_Toc5421)

[3.4. Selectors 24](#_Toc21951)

[*3.4.1. Universal selector* 27](#_Toc20770)

[*3.4.2. Type selector* 27](#_Toc1311)

[*3.4.3. Identity selector* 27](#_Toc7782)

[*3.4.4. Class selector* 27](#_Toc31689)

[*3.4.5. Descendant selector* 27](#_Toc19754)

[*3.4.6. Child selector* 27](#_Toc11482)

[*3.4.7. Adjacent sibling selector* 27](#_Toc8446)

[*3.4.8. Attribute selector* 28](#_Toc7912)

[*3.4.9. Pseudo class selector* 28](#_Toc6751)

[*3.4.10. Group selector* 28](#_Toc31687)

[3.5. Đơn vị đo lường CSS 28](#_Toc471)

[3.6. Kế thừa thuộc tính 29](#_Toc16321)

[3.7. Các nhóm thuộc tính trong CSS 29](#_Toc26850)

[3.8. Float & Clear 30](#_Toc2899)

[3.9. Flex 31](#_Toc14133)

[3.10. Grid 32](#_Toc29663)

[3.11. CSS3 33](#_Toc20258)

[3.12. SCSS 34](#_Toc21247)

[3.13. SASS 35](#_Toc27178)

[**CHƯƠNG 4: Thiết kế trang Web giới thiệu về Chùa Hương** 38](#_Toc20822)

[4.1. Ý tưởng về trang wed 38](#_Toc5878)

[4.2. Xây dựng bố cục của trang Web 39](#_Toc4379)

[4.3. Thiết kế trang Web bằng HTML và CSS 40](#_Toc19538)

[4.4. Kết quả đạt được 47](#_Toc15783)

[**KẾT LUẬN** 53](#_Toc24809)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 54](#_Toc12497)

**MỤC LỤC HÌNH ẢNH**

**(Nếu có)**

**MỤC LỤC BẢNG**

**(Nếu có)**

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

**1.1. Các khái niệm cơ bản**

Trong lĩnh vực thiết kế web và triển khai hệ thống phần mềm, có một số khái niệm cơ bản quan trọng mà các chuyên gia và nhà phát triển cần hiểu rõ. Dưới đây là một số khái niệm quan trọng:

***1.1.1. Thiết kế Web (Web Design)***

- Thiết kế giao diện người dùng (UI): Là quá trình tạo ra giao diện mà người dùng sẽ tương tác với trang web hoặc ứng dụng. Nó bao gồm việc sắp xếp các yếu tố như nút, menu, và hình ảnh để tối ưu hóa trải nghiệm người dùng.

- Trải nghiệm người dùng (UX): Là khái niệm liên quan đến cảm nhận toàn diện của người dùng khi sử dụng trang web hoặc ứng dụng. Nó bao gồm việc đảm bảo sự thuận tiện, dễ sử dụng, và đáp ứng đối với người dùng.

- Responsive Design: Là phong cách thiết kế mà đảm bảo trang web có thể hiển thị đúng trên nhiều thiết bị và kích thước màn hình khác nhau, từ máy tính đến điện thoại di động.

***1.1.2. Triển khai Hệ thống Phần mềm***

- Phần mềm: Là tập hợp các chương trình máy tính, dữ liệu, và tài nguyên hệ thống cần thiết để thực hiện các nhiệm vụ cụ thể.

- Triển khai: Là quá trình triển khai phần mềm từ môi trường phát triển sang môi trường thực tế để sử dụng. Nó bao gồm cả việc cài đặt, cấu hình, và kiểm thử để đảm bảo tính ổn định và hiệu suất của hệ thống.

- Hệ thống: Bao gồm phần mềm, phần cứng, và các thành phần khác tương tác để thực hiện các chức năng cụ thể.

***1.1.3. Các công nghệ và ngôn ngữ phổ biến***

- HTML (HyperText Markup Language): Ngôn ngữ đánh dấu được sử dụng để xây dựng cấu trúc nội dung của trang web.

- CSS (Cascading Style Sheets): Ngôn ngữ quy định cách mà nội dung HTML sẽ được hiển thị trên trình duyệt, bao gồm màu sắc, font chữ, và bố cục.

- JavaScript: Ngôn ngữ lập trình dùng để thêm tính năng tương tác động vào trang web.

- Databases (Cơ sở dữ liệu): Là nơi lưu trữ và quản lý dữ liệu, giúp hệ thống có thể truy xuất và xử lý thông tin một cách hiệu quả.

**1.2. Phân biệt Web tĩnh, Web động**

***1.2.1. Web tĩnh (Static Web)***

* **Đặc điểm chính:**

- Trang web tĩnh hiển thị thông tin cố định mà không có sự thay đổi đáng kể dựa trên hành động của người dùng.

- Nội dung của trang web tĩnh thường được xây dựng trước và không thay đổi cho mỗi người dùng hay mỗi lượt truy cập.

* **Ưu điểm:**

- Tốc độ tải trang nhanh vì nội dung đã được tạo trước và được lưu trữ ổn định.

- Dễ quản lý và triển khai vì không có sự phức tạp trong xử lý động.

* **Nhược điểm:**

-Thiếu tính tương tác và cá nhân hóa do nội dung không thay đổi.

- Khó duy trì trang web lớn với nhiều nội dung đa dạng.

***1.2.2. Web động (Dynamic Web)***

* **Đặc điểm chính:**

- Trang web động có khả năng thay đổi nội dung dựa trên hành động của người dùng hoặc dữ liệu mới.

- Sử dụng các ngôn ngữ lập trình như PHP, Python, Ruby, hoặc ASP.NET để tạo và hiển thị nội dung động.

* **Ưu điểm:**

-Cung cấp trải nghiệm người dùng tương tác và cá nhân hóa.

- Dễ dàng quản lý nội dung đa dạng và thay đổi.

* **Nhược điểm:**

- Tăng độ phức tạp trong việc phát triển và triển khai.

- Tốc độ tải trang có thể chậm hơn so với trang web tĩnh do cần xử lý động.

***1.2.3. Sự phân biệt:***

* **Tính tương tác:**

- Web tĩnh: Thường không có tính tương tác, nội dung không thay đổi theo người dùng.

- Web động: Có tính tương tác, nội dung có thể thay đổi dựa trên hành động của người dùng.

* **Quản lý nội dung:**

- Web tĩnh: Dễ quản lý cho các trang web nhỏ với nội dung tĩnh.

- Web động: Phù hợp cho các trang web có nội dung phức tạp và thay đổi thường xuyên.

* **Hiệu suất:**

- Web tĩnh: Tải trang nhanh, đặc biệt là cho người dùng mới.

- Web động: Có thể có thời gian tải trang lâu hơn do cần xử lý động trước khi hiển thị.

**1.3. Một số thuật ngữ (Hosting, Tên miền-Domain Name-Web Server Name, HomePage, Web Site, WebServer, URL-Uniform Resource Location, Browser, …)**

1. **Hosting:**

- Hosting:Dịch vụ cung cấp không gian lưu trữ và tài nguyên trên máy chủ để lưu trữ trang web hoặc ứng dụng web. Người dùng có thể mua dịch vụ hosting từ các nhà cung cấp để đưa trang web của họ lên Internet.

1. **Tên miền (Domain Name) và Web Server Name:**

- Tên miền (Domain Name): Địa chỉ dễ nhớ được sử dụng để truy cập trang web. Ví dụ: "example.com".

- Web Server Name: Là tên của máy chủ mà trang web hoặc ứng dụng web đang chạy trên đó. Thường được liên kết với tên miền để tạo ra URL đầy đủ.

3. **HomePage:**

- HomePage**:** Trang chính của một trang web, thường là trang mà người dùng thấy khi truy cập vào tên miền của trang web đó.

4. **Web Site:**

- Web Site:Tập hợp các trang web liên kết với nhau và thường là một không gian thông tin trực tuyến với một chủ đề hay mục đích cụ thể.

5. **Web Server:**

- Web Server: Máy chủ chịu trách nhiệm xử lý các yêu cầu từ trình duyệt web và cung cấp nội dung trang web cho người dùng. Apache, Nginx là những ví dụ về phần mềm web server.

6. **URL (Uniform Resource Locator):**

- URL (Uniform Resource Locator): Địa chỉ web hoặc địa chỉ tài nguyên trên Internet. Ví dụ: "https://www.example.com/index.html". Nó bao gồm giao thức (https), tên miền ([www.example.com](http://www.example.com/)), và đường dẫn tài nguyên (index.html).

7. **Trình duyệt (Browser):**

- Trình duyệt (Browser): Phần mềm cho phép người dùng duyệt và tương tác với các trang web. Ví dụ: Google Chrome, Mozilla Firefox, Microsoft Edge.

8. **SSL (Secure Sockets Layer) và HTTPS:**

- SSL (Secure Sockets Layer): Chuẩn bảo mật để tạo một kết nối an toàn giữa máy chủ và trình duyệt.

- HTTPS (Hypertext Transfer Protocol Secure): Phiên bản an toàn của giao thức HTTP sử dụng SSL để mã hóa dữ liệu truyền tải giữa trình duyệt và máy chủ.

9. **Responsive Design:**

- Responsive Design: Phong cách thiết kế đảm bảo trang web có thể hiển thị đúng trên nhiều thiết bị và kích thước màn hình khác nhau.

10. **DNS (Domain Name System):**

- DNS (Domain Name System): Hệ thống giúp chuyển đổi tên miền thành địa chỉ IP, cho phép máy tính xác định được vị trí của máy chủ.

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

1. **Visual Studio Code:**

- Visual Studio Code (VS Code): Một trình soạn thảo mã nguồn mở và miễn phí, được phát triển bởi Microsoft. Nó hỗ trợ nhiều ngôn ngữ lập trình, cung cấp tính năng mở rộng và là một công cụ phổ biến cho phát triển web.

2. **Adobe Dreamweaver:**

- Adobe Dreamweaver: Một ứng dụng của Adobe Systems, là một môi trường phát triển tích hợp (IDE) cho thiết kế web. Dreamweaver cung cấp giao diện WYSIWYG (What You See Is What You Get) cùng với chế độ chỉnh sửa mã nguồn.

3. **Sublime Text:**

- Sublime Text: Một trình soạn thảo văn bản và mã nguồn với giao diện sạch sẽ và nhanh chóng. Nó hỗ trợ nhiều ngôn ngữ lập trình và có khả năng mở rộng mạnh mẽ.

Các công cụ trên giúp nhà phát triển và nhà thiết kế tạo ra và quản lý các dự án web một cách hiệu quả, từ quá trình thiết kế đến triển khai và duy trì. Lựa chọn công cụ phụ thuộc vào nhu cầu cụ thể của dự án và sở thích cá nhân của người làm việc.

**Kết luận chương 1:**

Chương 1 đã giới thiệu về lĩnh vực Thiết kế Web và Triển khai Hệ thống Phần mềm, tập trung vào các khái niệm cơ bản và một số thuật ngữ quan trọng. Dưới đây là kết luận tổng quan về những điểm chính đã được đề cập:

1. **Thiết kế Web:**

- Bao gồm Thiết kế Giao diện Người dùng (UI) và Trải nghiệm Người dùng (UX).

- Responsive Design đảm bảo trang web hoạt động trên nhiều thiết bị và kích thước màn hình.

1. **Triển khai Hệ thống Phần mềm:**

- Bao gồm việc cài đặt, cấu hình, và kiểm thử để đảm bảo tính ổn định và hiệu suất của hệ thống.

1. **Web tĩnh và Web động:**

- Web tĩnh không có tính tương tác, nội dung cố định trước.

- Web động có tính tương tác, nội dung thay đổi dựa trên hành động của người dùng.

1. **Thuật ngữ Quan trọng:**

- Đã giới thiệu về Hosting, Tên miền, Web Server Name, HomePage, Web Site, Web Server, URL, Trình duyệt, SSL, và nhiều khái niệm khác.

1. **Công cụ Thiết kế Web:**

- Một số công cụ như Visual Studio Code, Adobe Dreamweaver và Sublime Text đã được đề cập.

Chương 1 cung cấp nền tảng kiến thức cơ bản để hiểu và tiếp cận lĩnh vực này. Việc nắm vững những khái niệm và công cụ này là quan trọng để xây dựng và duy trì các dự án web và hệ thống phần mềm một cách hiệu quả.

**CHƯƠNG II: HTML & HTML5**

* 1. **Tổng quan về HTML**

HTML là viết tắt của "HyperText Markup Language" (Ngôn ngữ Đánh dấu Siêu văn bản). Đây là một ngôn ngữ lập trình dùng để tạo và thiết kế trang web. HTML cung cấp cấu trúc cơ bản cho trang web bằng cách sử dụng các thẻ và các phần tử để mô tả nội dung.

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

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

- **<!DOCTYPE html>**: Đây là khai báo DOCTYPE, thông báo cho trình duyệt về phiên bản HTML mà trang web sử dụng. Ví dụ, <!DOCTYPE html> là DOCTYPE cho HTML5.

- **<html>**: Thẻ mở đầu cho tài liệu HTML. Tất cả các phần của trang web sẽ nằm trong thẻ này.

- **<head>**: Khu vực này chứa các thông tin meta về trang web, tiêu đề của trang (hiển thị trên thanh tiêu đề trình duyệt), liên kết đến các tài nguyên bên ngoài như CSS hoặc JavaScript.

- **<meta charset="UTF-8">**: Điều này khai báo bảng mã ký tự sử dụng trong tài liệu là UTF-8, giúp hỗ trợ hiển thị các ký tự đa ngôn ngữ.

- **<title>**: Đây là nơi để đặt tiêu đề của trang web. Tiêu đề này sẽ hiển thị trên thanh tiêu đề của trình duyệt.

- **<body>**: Khu vực này chứa nội dung thực sự của trang web. Tất cả các phần tử như văn bản, hình ảnh, liên kết, biểu đồ sẽ được đặt trong thẻ này.

* 1. **Các thẻ HTML thông dụng**

- Các thẻ xử lý văn bản:

* **<p>**: Định nghĩa đoạn văn bản.
* **<h1>** đến **<h6>**: Đặt tiêu đề với các cấp độ khác nhau (h1 là lớn nhất, h6 là nhỏ nhất).
* **<strong>** hoặc **<b>**: In đậm văn bản.

- Các thẻ tạo bảng:

* **<table>**: Định nghĩa một bảng.
* **<tr>**: Định nghĩa một dòng trong bảng.
* <**th>**: Định nghĩa ô tiêu đề trong bảng.
* **<td>**: Định nghĩa ô dữ liệu trong bảng.

- Thẻ liên kết:

* **<a>**: Định nghĩa liên kết (hyperlink).
* **<img>**: Chèn hình ảnh.
* **<figure>** và **<figcaption>**: Để nhóm và mô tả hình ảnh.

- Các thẻ đa phương tiện:

* **<audio>**: Chèn phương tiện âm thanh.
* **<video>**: Chèn phương tiện video.

- Thẻ tạo khung:

* **<div>**: thường được sử dụng để tạo ra các container hoặc nhóm các phần tử có cùng một kiểu dáng hoặc thuộc tính.
  1. **Các thẻ tạo biểu mẫu (form)**

Một số thẻ quan trọng để tạo biểu mẫu:

- <**form>:**

* Thẻ <form> định nghĩa một biểu mẫu trên trang web. Tất cả các phần tử nhập liệu và các phần tử liên quan đến biểu mẫu nằm bên trong thẻ này.
* Thuộc tính action xác định URL mà dữ liệu biểu mẫu sẽ được gửi đến khi người dùng nhấn nút gửi, và thuộc tính method xác định phương thức HTTP sẽ được sử dụng (thường là "get" hoặc "post").

- **<input>:**

* Thẻ <input> được sử dụng để tạo các ô nhập liệu, nút radio, nút checkbox, và nút gửi trong biểu mẫu. Loại của ô nhập liệu được xác định thông qua thuộc tính type.

- **<textarea>:**

* Thẻ <textarea> tạo một ô nhập văn bản lớn, thường được sử dụng để nhận phản hồi dài hoặc các đoạn văn bản lớn.

- **<select>** và **<option>**:

* Thẻ <select> tạo một danh sách thả xuống, trong khi thẻ <option> định nghĩa các mục trong danh sách.

- **<label>:**

* Thẻ <label> được sử dụng để mô tả ô nhập liệu và giúp cải thiện khả năng tương tác của người dùng.

- **<button>:**

* Thẻ <button> tạo một nút, thường được sử dụng để gửi biểu mẫu khi được nhấn.
  1. **Một số thẻ HTML đặc biệt**

Một số thẻ HTML đặc biệt:

**- <header>:**

* Thẻ <header> được sử dụng để xác định phần đầu của một trang hoặc một phần của trang web. Thường chứa tiêu đề, logo, hoặc các phần tử đầu khác.

- **<footer>:**

* Thẻ <footer> xác định phần chân của một trang hoặc một phần của trang web. Thường chứa thông tin liên hệ, bản quyền, và các liên kết khác.

- **<nav>:**

* Thẻ <nav> được sử dụng để đánh dấu một khu vực chứa các liên kết điều hướng, thường xuất hiện như một menu.

- **<article>:**

* Thẻ <article> xác định một phần tử chứa nội dung độc lập, thường là một bài viết hoặc một phần nội dung tự đủ để đứng một mình.

- **<section>:**

* Thẻ <section> xác định một phần hoặc một khu vực trong tài liệu. Nó thường được sử dụng để nhóm các nội dung liên quan với nhau.

- **<aside>:**

* Thẻ <aside> được sử dụng để xác định một khu vực bên lề, thường chứa thông tin bổ sung không quan trọng đối với nội dung chính.

- **<time>:**

* Thẻ <time> được sử dụng để đánh dấu một thông tin thời gian hoặc ngày trong tài liệu.

Những thẻ này giúp tăng cường ý nghĩa semantical của trang web, cũng như hỗ trợ trong việc đọc màn hình và tối ưu hóa SEO. Đối với HTML5, việc sử dụng các thẻ semantical như trên được khuyến khích để cải thiện khả năng hiểu và tương tác của trang web.

**2.6. HTML5**

* HTML5 là gì?

HTML5 là phiên bản mới nhất của ngôn ngữ đánh dấu HTML, được thiết kế để cung cấp một nền tảng mạnh mẽ và linh hoạt cho việc phát triển các ứng dụng web và trang web hiện đại. HTML5 không chỉ là một phiên bản cập nhật của HTML4, mà còn là một tiêu chuẩn mới với nhiều tính năng và cải tiến.

* Cú pháp HTML5

Cú pháp HTML5 giữ nguyên đối với cú pháp chung của HTML, nhưng với thêm vào đó một số phần tử mới và thuộc tính. Dưới đây là một ví dụ về cú pháp cơ bản của HTML5:

* **<!DOCTYPE html>**: Khai báo phiên bản HTML và loại tài liệu.
* **<html lang="en">**: Định nghĩa ngôn ngữ của trang web (ở đây là tiếng Anh).
* **<head>**: Chứa các thông tin đầu trang như charset, viewport, và tiêu đề trang.
* **<meta charset="UTF-8">**: Khai báo bảng mã ký tự là UTF-8.
* **<meta name="viewport" content="width=device-width, initial-scale=1.0">**: Cấu hình viewport cho thiết bị di động.
* **<title>**: Đặt tiêu đề của trang web.
* **<header>**, **<section>**, **<article>**, **<footer>**: Các phần tử semantic mới của HTML5.
* **<nav>**: Một phần tử mới để đánh dấu một khu vực chứa các liên kết điều hướng.
* Khai báo DOCTYPE trong HTML5:

Trong HTML5, không còn cần khai báo DTD (Document Type Definition) như các phiên bản HTML trước đó. Thay vào đó, bạn có thể sử dụng một đoạn mã đơn giản như sau:

**<!DOCTYPE html>**

Dòng mã này đủ để xác định rằng trang web của bạn sử dụng HTML5. Trong HTML5, DOCTYPE không còn đòi hỏi phải chỉ định DTD cụ thể như trước đây, và đoạn mã ngắn trên là đủ để trình duyệt hiểu rằng trang web sử dụng tiêu chuẩn HTML5.

* Thẻ ngữ nghĩa:

Trong HTML5, các thẻ ngữ nghĩa (semantic tags) là những thẻ được thiết kế để mô tả ý nghĩa và chức năng của nội dung chúng bao quanh. Sử dụng các thẻ ngữ nghĩa giúp trang web trở nên có ý nghĩa hơn và dễ đọc hơn cho cả máy đọc màn hình và công cụ tìm kiếm. Dưới đây là một số thẻ ngữ nghĩa trong HTML5:

**<header>:**

Được sử dụng để xác định một phần đầu của trang hoặc một phần của trang web. Thường chứa tiêu đề, logo, hoặc các phần tử đầu khác.

**<nav>:**

Được sử dụng để xác định một khu vực chứa các liên kết điều hướng, thường xuất hiện như một menu.

**<section>:**

Được sử dụng để xác định một phần hoặc một khu vực trong tài liệu. Nó thường được sử dụng để nhóm các nội dung liên quan với nhau.

**<article>:**

Được sử dụng để xác định một phần tử chứa nội dung độc lập, thường là một bài viết hoặc một phần nội dung tự đủ để đứng một mình.

**<aside>:**

Được sử dụng để xác định một khu vực bên lề, thường chứa thông tin bổ sung không quan trọng đối với nội dung chính.

**<footer>:**

Được sử dụng để xác định phần chân của một trang hoặc một phần của trang web. Thường chứa thông tin liên hệ, bản quyền, và các liên kết khác.

- Các thẻ ngữ nghĩa giúp cải thiện ý nghĩa semantical của trang web, giúp máy đọc màn hình hiểu được cấu trúc trang và hỗ trợ tối ưu hóa cho công cụ tìm kiếm.

* Webform:

"Web form" (biểu mẫu web) là một thành phần quan trọng trong HTML, cho phép người dùng nhập dữ liệu và tương tác với trang web. Biểu mẫu thường được sử dụng để thu thập thông tin từ người dùng, đăng ký tài khoản, thực hiện tìm kiếm, và nhiều mục đích khác. Trong HTML, các phần tử sau được sử dụng để tạo biểu mẫu:

1. **<form>**
2. **<input>**
3. **<textarea>**
4. **<select> và <option>**
5. **<label>**
6. **<button>**

**Kết luận chương 2**

Trong chương 2 về HTML và HTML5, chúng ta đã đi qua một số khái niệm cơ bản và các thành phần quan trọng của ngôn ngữ đánh dấu này. Dưới đây là một kết luận về những điểm chính đã được thảo luận:

1. HTML và HTML5:

* HTML (Hyper Text Markup Language) là ngôn ngữ đánh dấu được sử dụng để xây dựng cấu trúc và hiển thị nội dung trên trang web.
* HTML5 là phiên bản mới nhất của HTML, mang lại nhiều tính năng và cải tiến, bao gồm hỗ trợ multimedia, canvas, localStorage, và nhiều tính năng khác.

1. Cấu trúc cơ bản của trang HTML:

* Một trang HTML thường bắt đầu với khai báo DOCTYPE để xác định phiên bản HTML.
* Phần <html> chứa toàn bộ nội dung trang, và có thể có thuộc tính ngôn ngữ (lang).
* Phần <head> chứa các thông tin đầu trang như charset, viewport, và tiêu đề.
* Phần <body> chứa nội dung hiển thị trên trang web.

1. Thẻ và thuộc tính HTML:

* HTML sử dụng các thẻ để đánh dấu các phần khác nhau của nội dung.
* Thuộc tính được sử dụng để cung cấp thông tin bổ sung về các thẻ.

1. Thẻ ngữ nghĩa trong HTML5:

* HTML5 giới thiệu nhiều thẻ ngữ nghĩa như <header>, <nav>, <section>, <article>, <aside>, <footer> để cải thiện ý nghĩa semantical của trang web.

1. Biểu mẫu web trong HTML:

* Thẻ <form> định nghĩa một biểu mẫu trên trang web.
* Các thẻ như <input>, <textarea>, <select>, <label>, <button> được sử dụng để tạo các phần tử nhập liệu và tương tác với người dùng.

1. HTML5:

* HTML5 mang lại nhiều tính năng mới như hỗ trợ multimedia, canvas, Web Storage, form và input cải tiến, và các thẻ semantic mới.
* Cú pháp HTML5 đơn giản hóa việc khai báo DOCTYPE và giúp xây dựng trang web hiện đại và linh hoạt hơn.

Chương 2 đã giúp hiểu rõ hơn về cấu trúc và các thành phần cơ bản của HTML và HTML5, tạo nền tảng cho việc phát triển và thiết kế trang web.

**CHƯƠNG III: CSS và CSS3**

 (Cascading Style Sheets)

**3.1. CSS là gì?**

CSS là viết tắt của "Cascading Style Sheets," trong tiếng Việt dịch là "Bảng điều khiển kiểu dáng lan truyền." CSS là một ngôn ngữ lập trình được sử dụng để mô tả cách hiển thị các trang web được viết bằng HTML hoặc XML. Mục tiêu chính của CSS là tách biệt nội dung (HTML) và kiểu dáng (CSS), giúp nâng cao tính tương thích và duyệt web.

**3.2. Cú pháp CSS**

Cú pháp của CSS được sử dụng để đặt ra các quy tắc kiểu dáng, xác định cách mà các phần tử HTML sẽ được hiển thị trên trình duyệt. Mỗi quy tắc kiểu dáng bao gồm một bộ chọn (selector) và một khối khai báo (declaration block).

**3.3. Áp dụng CSS vào trang HTML**

Để áp dụng CSS vào trang HTML, bạn cần liên kết (link) tệp CSS vào tệp HTML hoặc sử dụng các quy tắc kiểu dáng trực tiếp trong tệp HTML. Dưới đây là cách bạn có thể thực hiện điều này:

- Liên kết Tệp CSS Bên Ngoài:

1. Tạo một Tệp CSS: Tạo một tệp CSS riêng (ví dụ: styles.css) chứa các quy tắc kiểu dáng của bạn.
2. Liên kết Tệp CSS vào Tệp HTML: Trong phần <head> của tệp HTML, thêm một thẻ <link> để liên kết tệp CSS với tệp HTML.

- Sử Dụng Quy Tắc Kiểu Dáng Trực Tiếp:

Sử dụng quy tắc kiểu dáng trực tiếp trong phần <style> của tệp HTML.

**3.4. Selectors**

Selectors (Bộ chọn) trong CSS là cách xác định các phần tử HTML mà bạn muốn áp dụng kiểu dáng. Dưới đây là một số loại bộ chọn phổ biến:

1. Bộ chọn theo tên thẻ:

p {

color: #333;

}

Áp dụng kiểu dáng cho tất cả các thẻ **<p>**.

1. Bộ chọn theo lớp (Class):

.button {

background-color: #0066cc;

color: #fff;

}

Áp dụng kiểu dáng cho phần tử có ID là "header" **(<div id="header">)**

1. Bộ chọn theo ID:

#header {

font-size: 20px;

color: #333;

}

Áp dụng kiểu dáng cho phần tử có ID là "header" **(<div id="header">).**

1. Bộ chọn con:

nav ul {

list-style-type: none;

}

Áp dụng kiểu dáng cho tất cả các thẻ **<ul>** nằm trong phần tử **<nav>.**

1. Bộ chọn con trực tiếp:

article > p {

font-style: italic;

}

Áp dụng kiểu dáng cho tất cả các thẻ **<p>** là con trực tiếp của phần tử **<article>.**

1. Bộ chọn theo thuộc tính:

input[type="text"] {

border: 1px solid #ccc;

}

Áp dụng kiểu dáng cho tất cả các phần tử **<input>** có thuộc tính **type** là "text".

1. Bộ chọn kết hợp:

section.container {

background-color: #f5f5f5;

}

Áp dụng kiểu dáng cho tất cả các phần tử **<section>** có lớp là "container".

1. Bộ chọn pseudo-classes:

a:hover {

color: #ff0000;

}

Áp dụng kiểu dáng cho thẻ **<a>** khi chuột đưa vào nó.

1. Bộ chọn pseudo-elements:

p::first-line {

font-weight: bold;

}

Áp dụng kiểu dáng cho dòng đầu tiên của mỗi thẻ **<p>.**

***3.4.1. Universal selector***

Universal Selector là một loại bộ chọn trong CSS được biểu diễn bằng dấu sao (\*). Nó được sử dụng để áp dụng một đối tượng CSS cho tất cả các phần tử trong trang web. Cụ thể, nó chấp nhận tất cả các phần tử HTML trên trang và áp dụng các quy tắc được xác định cho chúng.

***3.4.2. Type selector***

Type Selector, hay còn được gọi là Element Selector, là một loại bộ chọn trong CSS cho phép bạn chọn tất cả các thẻ HTML của một loại cụ thể và áp dụng các quy tắc CSS cho chúng. Để sử dụng Type Selector, bạn chỉ cần ghi tên của loại phần tử mà bạn muốn chọn.

***3.4.3. Identity selector***

ID Selector là một loại bộ chọn trong CSS được biểu diễn bằng ký tự thăng (#), theo sau là tên ID của phần tử mà bạn muốn chọn. ID là một đặc điểm độc nhất được gán cho một phần tử trong HTML, và mỗi ID chỉ nên xuất hiện duy nhất trên trang web.

***3.4.4. Class selector***

Class Selector là một loại bộ chọn trong CSS được biểu diễn bằng dấu chấm (.), theo sau là tên lớp mà bạn muốn chọn. Lớp là một cách để gán một nhóm các phần tử có cùng kiểu hoặc tính năng và một phần tử có thể thuộc nhiều lớp khác nhau.

***3.4.5. Descendant selector***

Descendant Selector là một loại bộ chọn trong CSS, cho phép bạn chọn tất cả các phần tử con của một phần tử khác, không phụ thuộc vào cấp độ con cái đó. Bộ chọn này được biểu diễn bằng dấu cách (khoảng trắng) giữa các tên phần tử.

***3.4.6. Child selector***

Child Selector là một loại bộ chọn trong CSS, giúp bạn chọn tất cả các phần tử con trực tiếp của một phần tử cha, không bao gồm các phần tử con của các phần tử con đó. Bộ chọn này được biểu diễn bằng dấu "greater than" (>).

***3.4.7. Adjacent sibling selector***

Adjacent Sibling Selector là một loại bộ chọn trong CSS, cho phép bạn chọn tất cả các phần tử anh em ngay kế cạnh của một phần tử khác, có cùng một phần tử cha. Bộ chọn này được biểu diễn bằng dấu "plus" (+).

***3.4.8. Attribute selector***

Attribute Selector là một loại bộ chọn trong CSS, cho phép bạn chọn các phần tử dựa trên giá trị của thuộc tính của chúng. Bộ chọn này sử dụng cú pháp **[attribute=value]**, nơi **attribute** là tên của thuộc tính và **value** là giá trị cụ thể mà bạn muốn chọn.

***3.4.9. Pseudo class selector***

Pseudo-class selector là một loại bộ chọn trong CSS, cho phép bạn chọn các phần tử dựa trên trạng thái hoặc vị trí của chúng trong DOM. Pseudo-classes được biểu diễn bằng hai dấu hai chấm (::) hoặc một dấu hai chấm (:).

***3.4.10. Group selector***

Group Selector trong CSS cho phép bạn kết hợp nhiều bộ chọn lại với nhau để áp dụng cùng một tập hợp các quy tắc CSS cho nhiều loại phần tử hoặc các phần tử cụ thể. Bộ chọn nhóm được biểu diễn bằng cách sử dụng dấu phẩy (,).

**3.5. Đơn vị đo lường CSS**

Trong CSS, có nhiều đơn vị đo lường được sử dụng để xác định kích thước và khoảng cách của các phần tử trên trang web. Dưới đây là một số đơn vị đo lường phổ biến:

1. Đơn vị Tuyệt đối:

* px (pixel): Là đơn vị đo lường tuyệt đối, một pixel tương đương với một điểm ảnh trên màn hình.
* pt (point): Thường được sử dụng trong in ấn, một điểm bằng 1/72 inch.

1. Đơn vị Tương đối:

* % (phần trăm): Đo lường dựa trên phần trăm của một giá trị gốc. Ví dụ, width: 50% sẽ chiếm 50% của phần tử cha.
* em: Là đơn vị tương đối, thường được sử dụng cho kích thước font. 1em tương đương với kích thước font của phần tử hiện tại.

1. Đơn vị Thương của Chiều Dài:

* vw (viewport width): 1vw tương đương với 1% của chiều rộng của viewport.
* vh (viewport height): 1vh tương đương với 1% của chiều cao của viewport.
* vmin và vmax: Sử dụng để xác định giá trị tương đối dựa trên chiều nhỏ nhất hoặc lớn nhất của viewport.

1. Đơn vị Thời gian:

* s (seconds): Đơn vị đo thời gian, được sử dụng chủ yếu trong các thuộc tính animation và transition.

**3.6. Kế thừa thuộc tính**

Trong CSS, kế thừa thuộc tính đề cập đến khả năng của một phần tử con kế thừa giá trị của một số thuộc tính từ phần tử cha của nó. Điều này giúp giảm bớt việc phải đặt lại các giá trị của các thuộc tính cho từng phần tử con một cách tường minh.

Tuy nhiên, không phải tất cả các thuộc tính đều có thể kế thừa. Các thuộc tính nền như màu nền, hình nền, v.v., thường được kế thừa, trong khi các thuộc tính như kích thước và định dạng văn bản không thể kế thừa.

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

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

* font-family: Xác định font chữ được sử dụng.
* font-size: Đặt kích thước font.
* font-weight: Xác định độ đậm của font.
* font-style: Xác định kiểu chữ (in nghiêng, bình thường).
* line-height: Xác định độ cao dòng chữ.
* text-align: Đặt căn chỉnh văn bản.
* color: Xác định màu chữ.

- Thuộc tính định dạng nền (background)

* background-color: Xác định màu nền.
* background-image: Đặt hình ảnh nền.
* background-repeat: Xác định cách lặp lại hình ảnh nền.
* background-position: Xác định vị trí của hình ảnh nền.
* background-size: Xác định kích thước của hình ảnh nền.

- Mô hình hộp (box model)

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

- Các thuộc tính định margin, padding, border

* margin-top, margin-right, margin-bottom, margin-left: Xác định các giá trị margin cho từng hướng.
* padding-top, padding-right, padding-bottom, padding-left: Xác định các giá trị padding cho từng hướng.
* border-width, border-style, border-color: Xác định kích thước, kiểu và màu sắc của đường viền.

- Thuộc tính định dạng cách hiển thị (display)

* display: Xác định cách mà phần tử hiển thị trên trang (block, inline, inline-block, none, etc.).

- Thuộc tính xác định vị trí (position)

* position: Xác định cách một phần tử được định vị trong layout (relative, absolute, fixed, static).
* top, right, bottom, left: Xác định vị trí chính xác của phần tử (khi position không phải là static).

**3.8. Float & Clear**

***-*** Float(Nổi bọt):

* Là thuộc tính cho phép một phần tử nổi lên và dịch chuyển về bên trái hoặc bên phải của phần tử cha.
* Các phần tử khác có thể nổi theo theo chiều ngang của phần tử đã float.
* Sử dụng float để tạo cấu trúc đa cột và định hình phần tử trong layout.

- Clean(Xóa):

* clear là thuộc tính được sử dụng để xác định liệu một phần tử sẽ nổi theo cạnh trái, cạnh phải, hoặc cả hai cạnh của các phần tử đã nổi trước đó.
* Sử dụng clear để đảm bảo rằng phần tử không sẽ nổi theo cả hai bên của phần tử đã nổi trước đó.
* Điều này giúp tránh gặp vấn đề khi các phần tử nổi chồng lấn lên nhau.

**3.9. Flex**

Các Thuộc Tính Chính của Flexbox:

- display: flex; hoặc display: inline-flex;

* display: flex; chuyển container thành một flex container, nơi các phần tử con của nó trở thành các phần tử flex.
* display: inline-flex; thực hiện chức năng tương tự nhưng phần tử sẽ hiển thị như một phần tử inline.

- flex-direction: row|row-reverse|column|column-reverse;

* Xác định hướng của trục chính (main axis). Mặc định là row (trái sang phải).

- justify-content: flex-start|flex-end|center|space-between|space-around|space-evenly;

* Dùng để căn chỉnh các phần tử theo trục chính.

- align-items: stretch|flex-start|flex-end|center|baseline;

* Dùng để căn chỉnh các phần tử theo trục phụ.

- align-self: auto|flex-start|flex-end|center|baseline|stretch;

* Dùng để đặt căn chỉnh cho một phần tử cụ thể, bất kể giá trị align-items của container.

- flex-grow: number;

* Xác định tỷ lệ mà một phần tử sẽ mở rộng so với các phần tử flex khác trong container.

- flex-shrink: number;

* Xác định tỷ lệ mà một phần tử sẽ co lại nếu cần thiết.

- flex-basis: length|percentage|auto;

* Xác định kích thước cơ bản của một phần tử trước khi các thuộc tính flex-grow và flex-shrink được áp dụng.

- flex: none|auto|initial|number;

* Tổng hợp các thuộc tính flex-grow, flex-shrink, và flex-basis.

**3.10. Grid**

Các Thuộc Tính Chính của CSS Grid:

- display: grid;

* Xác định một phần tử là một container grid.

- grid-template-rows và grid-template-columns

* Xác định số lượng và kích thước của các hàng và cột trong lưới.

- grid-gap hoặc row-gap và column-gap

* Xác định khoảng cách giữa các hàng và cột trong lưới.

- grid-template-areas

* Xác định khu vực của các ô trong lưới.

- grid-auto-rows và grid-auto-columns

* Xác định kích thước mặc định của các hàng và cột không được xác định trước.

- grid-auto-flow

* Xác định hướng sắp xếp các phần tử mới trong lưới (theo hàng hoặc cột).

- grid-column và grid-row

* Xác định vị trí của một phần tử trong lưới dựa trên các hàng và cột.

- justify-items và align-items

* Xác định cách căn chỉnh các phần tử con bên trong ô của lưới.

- justify-content và align-content

* Xác định cách căn chỉnh lưới bên trong khung của nó.

**3.11. CSS3**

- CSS3 là gì?

CSS3 là phiên bản thứ ba của Cascading Style Sheets (CSS), một ngôn ngữ định dạng và trình bày cho các trang web. CSS3 bổ sung nhiều tính năng mới và cải thiện so với CSS2, giúp thiết kế web trở nên linh hoạt và đẹp mắt hơn.

- CSS3 Selectors

CSS3 Selector mở rộng các khả năng chọn lựa phần tử trong trang. Các selector mới bao gồm:

* Attribute Selectors: Chọn phần tử dựa trên thuộc tính và giá trị của nó.
* nth-child: Chọn phần tử con thứ n trong một phần tử cha.

- CSS3 Rounded Corners:

* Thuộc tính border-radius cho phép tạo các góc bo tròn cho các phần tử, tạo hiệu ứng thẩm mỹ.

- CSS3 Border Images:

* border-image cho phép sử dụng hình ảnh làm đường viền của phần tử.

- CSS3 Backgrounds

CSS3 cung cấp nhiều cải tiến cho nền trang web, bao gồm:

* Background Size: Điều chỉnh kích thước của hình nền.
* Background Origin: Xác định điểm bắt đầu vẽ hình nền.

- CSS Gradients

* linear-gradient và radial-gradient cho phép tạo hiệu ứng chuyển động màu sắc trên nền của phần tử.

- CSS3 Shadows

* box-shadow cho phép thêm bóng cho phần tử, tạo chiều sâu và độ chia cách.

- CSS3 Fonts

* CSS3 Fonts cung cấp tính năng nâng cao về kiểu chữ và định dạng văn bản.

- CSS3 2D Transforms

* transform cho phép biến đổi phần tử theo chiều ngang và chiều dọc.

- CSS3 3D Transforms

* transform mở rộng để hỗ trợ biến đổi phần tử trong không gian 3D.

- CSS3 Transitions

* transition tạo hiệu ứng mượt mà khi giá trị thuộc tính thay đổi.

- CSS3 Multiple Columns

* column-count và column-width cho phép tạo nhiều cột văn bản trong một phần tử.

- CSS3 Animations

* @keyframes và animation cho phép tạo hiệu ứng chuyển động linh hoạt.

**3.12. SCSS**

SCSS là viết tắt của "Sassy CSS," và nó là một dạng của Sass (Syntactically Awesome Stylesheets), một ngôn ngữ mở rộng của CSS. SCSS giữ lại cú pháp của CSS và bổ sung thêm các tính năng của Sass, giúp làm cho mã nguồn CSS trở nên dễ đọc hơn và quản lý dễ dàng hơn.

Dưới đây là một số điểm quan trọng về SCSS:

- Cú Pháp:

* SCSS sử dụng cú pháp giống với CSS, vì vậy bạn có thể bắt đầu sử dụng SCSS mà không cần thay đổi mã nguồn CSS hiện tại.

- Biến:

* SCSS hỗ trợ khai báo biến, giúp lưu trữ và tái sử dụng các giá trị nhất định như màu sắc, kích thước font, v.v.

- Nesting:

* SCSS cho phép lồng các quy tắc CSS trong nhau, tạo ra một cấu trúc lồng nhau thể hiện sự quan hệ giữa các phần tử.

- Mixins:

* Mixins là một cách để tái sử dụng các đoạn mã CSS. Bạn có thể định nghĩa một đoạn mã CSS và sử dụng nó trong nhiều địa điểm khác nhau.

- Extend/Inheritance:

* SCSS hỗ trợ kế thừa, giúp mở rộng một quy tắc CSS từ một quy tắc khác.

- Import:

* Bạn có thể chia nhỏ mã nguồn SCSS thành các file nhỏ và import chúng vào file chính.

- Operators:

* SCSS hỗ trợ các toán tử, giúp thực hiện các phép toán trong mã nguồn CSS.

- Control Directives:

* SCSS hỗ trợ các chỉ thị kiểm soát như @if, @else, @for, giúp điều khiển luồng thực thi của mã nguồn.

- Function:

* SCSS cho phép định nghĩa và sử dụng các hàm trong mã nguồn CSS.

- Responsive Design:

* SCSS giúp dễ dàng quản lý và duy trì các kiểu CSS phục vụ cho các kích thước màn hình khác nhau trong thiết kế đáp ứng.

**3.13. SASS**

SASS là một ngôn ngữ mở rộng của CSS (Cascading Style Sheets), giúp làm cho việc viết mã CSS trở nên linh hoạt và hiệu quả hơn. Dưới đây là một số điểm quan trọng về SASS:

- Cú Pháp:

* SASS có cú pháp riêng của mình, giúp tạo ra mã nguồn CSS dễ đọc hơn và có cấu trúc hơn.
* SASS hỗ trợ cả cú pháp có dạng có dấu ngoặc **{}** giống với CSS và cú pháp có dạng viết tắt với các khoảng trắng và dấu nhấn.

- Biến:

* SASS cho phép khai báo và sử dụng biến, giúp quản lý các giá trị nhất định như màu sắc, kích thước font, v.v.

- Nesting:

* SASS hỗ trợ lồng các quy tắc CSS bên trong nhau, giúp tạo ra một cấu trúc lồng nhau thể hiện sự quan hệ giữa các phần tử.

- Mixins:

* Mixins trong SASS là một cách để tái sử dụng các đoạn mã CSS. Bạn có thể định nghĩa một mixin và gọi nó ở nhiều nơi khác nhau.

- Extend/Inheritance:

* SASS hỗ trợ kế thừa, cho phép một quy tắc mở rộng từ một quy tắc khác.

- Import:

* Bạn có thể chia nhỏ mã nguồn SASS thành các file nhỏ và import chúng vào file chính.

- Operators:

* SASS hỗ trợ các toán tử, giúp thực hiện các phép toán trong mã nguồn CSS.

- Control Directives:

* SASS hỗ trợ các chỉ thị kiểm soát như **@if**, **@else**, **@for**, giúp điều khiển luồng thực thi của mã nguồn.

- Function:

* SASS cho phép định nghĩa và sử dụng các hàm trong mã nguồn CSS.

- Responsive Design:

* SASS giúp dễ dàng quản lý và duy trì các kiểu CSS phục vụ cho các kích thước màn hình khác nhau trong thiết kế đáp ứng.

Kết luận chương 3

Chương 3 về CSS và các phiên bản mở rộng như CSS3, SCSS, và SASS mang lại những kiến thức quan trọng về thiết kế và trình bày trong phát triển web. Dưới đây là một số điểm kết luận quan trọng:

- CSS:

* CSS (Cascading Style Sheets) là ngôn ngữ được sử dụng để định dạng và trình bày các phần tử trên trang web.
* Các thuộc tính và giá trị trong CSS quyết định các đặc tính như màu sắc, kích thước, và vị trí của các phần tử.

- CSS3:

* CSS3 là phiên bản mở rộng của CSS, mang lại nhiều tính năng mới như góc bo tròn, hình nền động, gradient, và hiệu ứng chuyển động.
* CSS3 giúp tạo ra giao diện web đẹp mắt, động độ và thú vị hơn.

- SCSS và SASS:

* SCSS và SASS là hai ngôn ngữ mở rộng của CSS, giúp làm cho việc viết mã CSS trở nên linh hoạt và hiệu quả hơn.
* Chúng hỗ trợ biến, lồng, mixins, và nhiều tính năng khác giúp quản lý và duy trì mã nguồn CSS dễ dàng hơn.

**CHƯƠNG 4: Thiết kế trang Web shop bán giày**

* 1. **Ý tưởng về trang wed**

**Tên trang wed:** Shop bán giày

**Mô tả:** Chào mừng bạn đến với [Tên Shop] điểm đến lý tưởng cho những đam mê giày dép và phong cách!

Chúng tôi tự hào là một trong những điểm đến hàng đầu dành cho những người yêu thời trang và đam mê giày dép. Với một bộ sưu tập đa dạng và phong phú từ những thương hiệu hàng đầu trên thế giới, chúng tôi cam kết mang đến cho bạn sự lựa chọn tuyệt vời nhất.

Tại [Tên Shop], chúng tôi hiểu rằng mỗi đôi giày không chỉ là sản phẩm, mà còn là biểu tượng cá nhân của phong cách và cá tính của bạn. Vì vậy, chúng tôi luôn cập nhật những xu hướng mới nhất trong ngành thời trang để đảm bảo bạn luôn tự tin và phong cách trong mọi bước đi.

Dù bạn đang tìm kiếm một đôi sneakers thoải mái cho cuộc sống hàng ngày, một đôi giày công sở lịch lãm hay một đôi boot đậm chất cá nhân, chúng tôi đều có mọi thứ bạn cần.

Hãy ghé thăm cửa hàng của chúng tôi hoặc trải nghiệm mua sắm trực tuyến thông qua trang web của chúng tôi để khám phá thêm về bộ sưu tập và dịch vụ của chúng tôi. Chúng tôi cam kết mang đến cho bạn trải nghiệm mua sắm tuyệt vời nhất và sự hài lòng tuyệt đối.

Hãy để [Tên Shop] là điểm đến của bạn cho sự phong phú và đa dạng trong thế giới của giày dép và phong cách!Top of Form

* 1. **Xây dựng bố cục của trang Web**

**Heder tiêu đề:**

**Logo của trang web**

Thẻ điều hướng (menu) chứa các liên kết đến các phần chính của trang web như "Giới Thiệu", "Sản Phẩm ", "Đăng Nhập ", "Liên Hệ",’’Đăng Ký’’.

Công cụ tìm kiếm (nếu cần thiết).

**Banner (Biểu ngữ)**

Giới thiệu nhiều loại giày

Tiêu đề chào mừng ngắn gọn và thu hút.

**Phần chính:**

**Giới thiệu:**

Gới thiệu nhiều loại giày phân loại

**Giày thể thao :**

Thư viện hình ảnh và video hiển thị Giày giá tiền

Mô tả chi tiết về các điểm đến đáng chú ý

Mô tả giá tiền lượt mua lượt bán

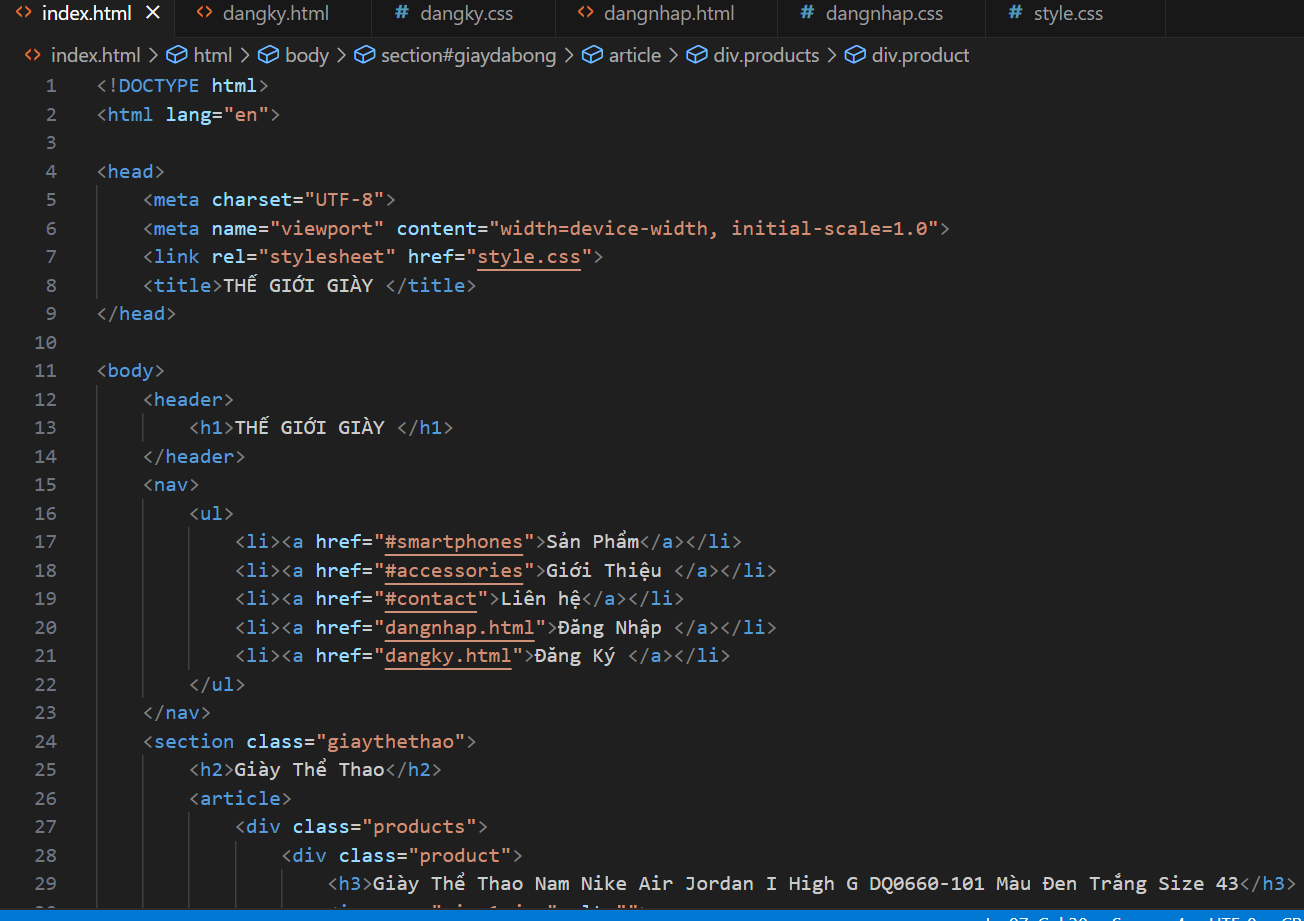
**Footer(chân trang)**

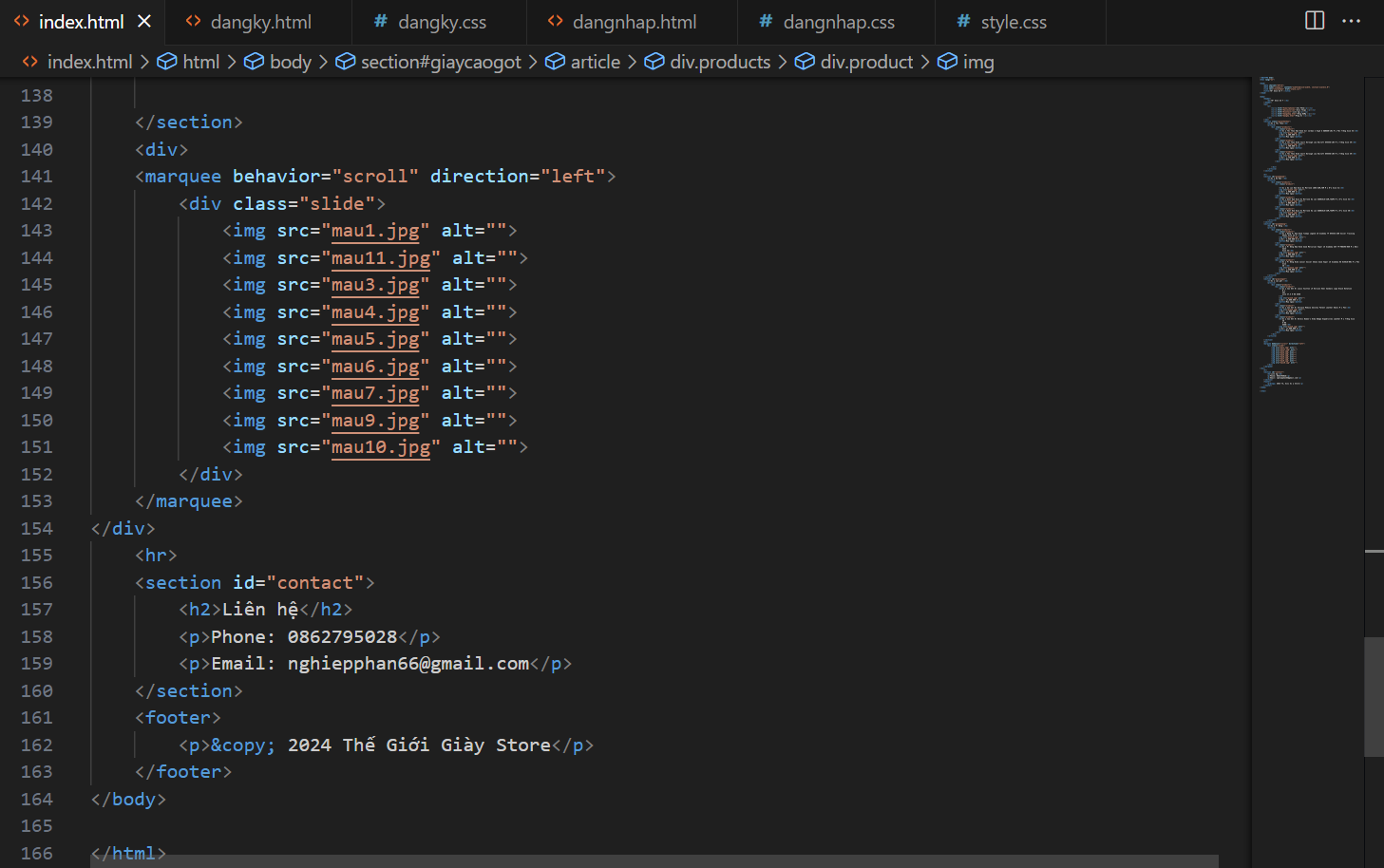
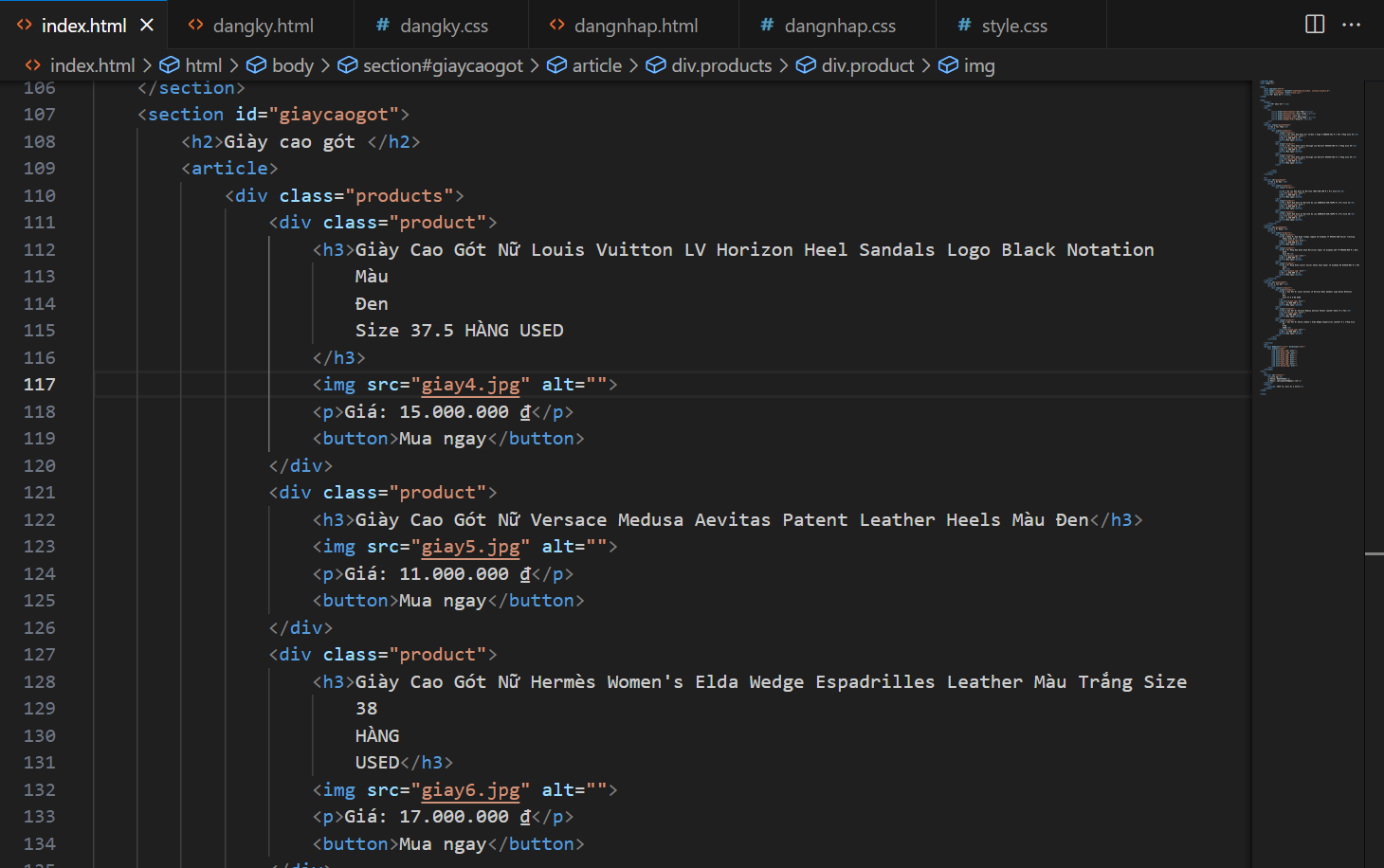
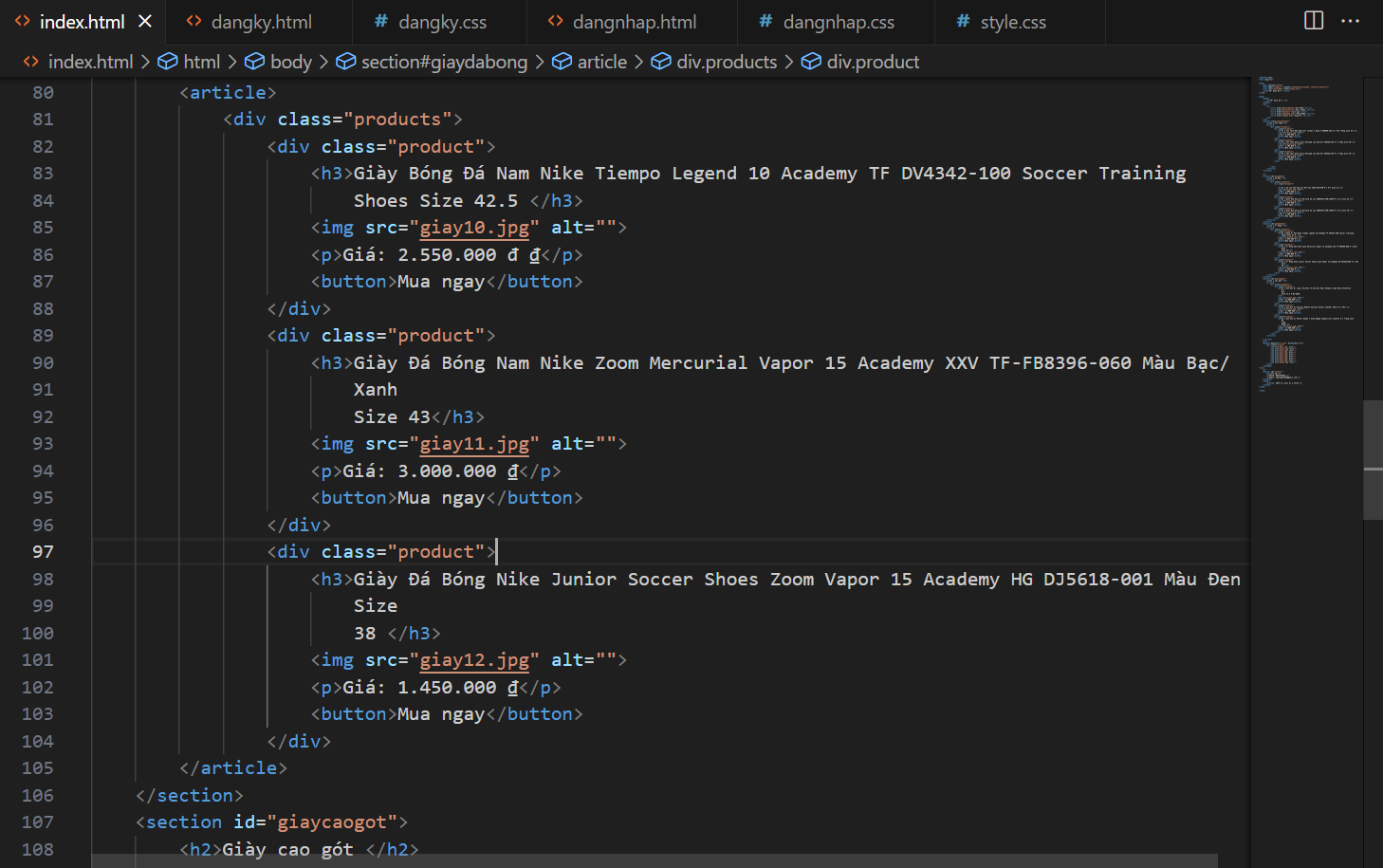
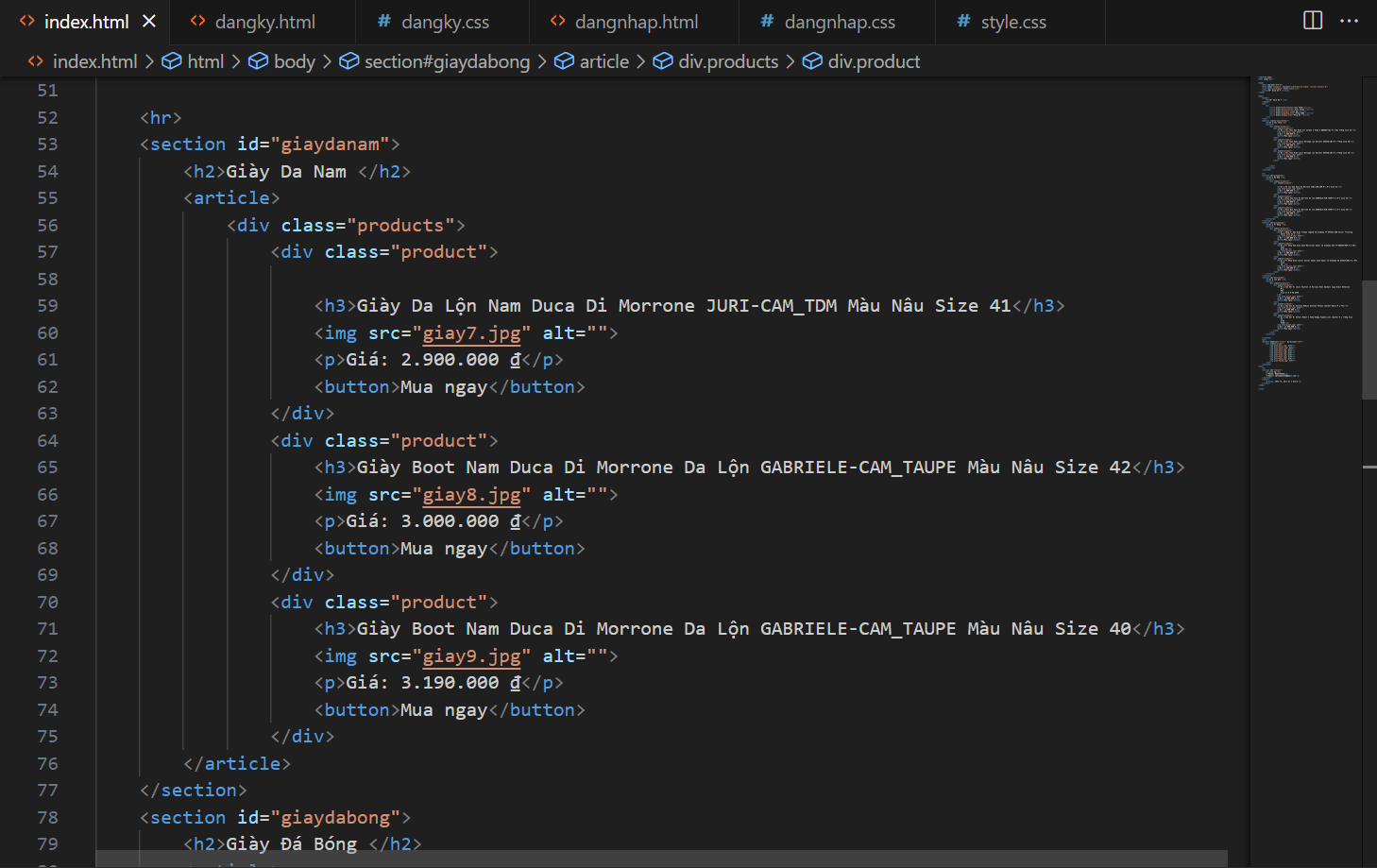
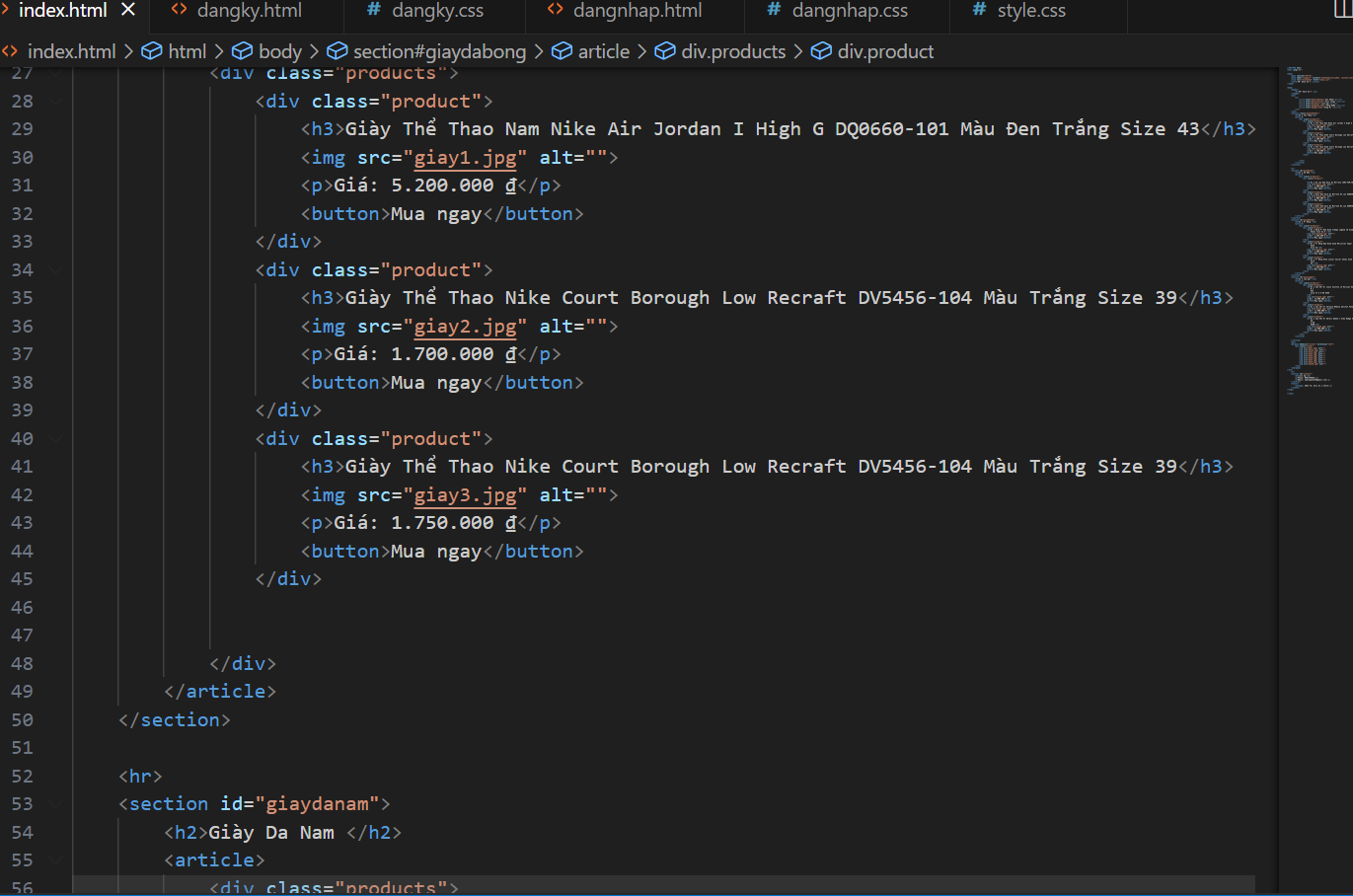
Thông tin liên hệ như địa chỉ, số điện thoại, địa chỉ email.

Liên kết đến các trang mạng xã hội của trang web.

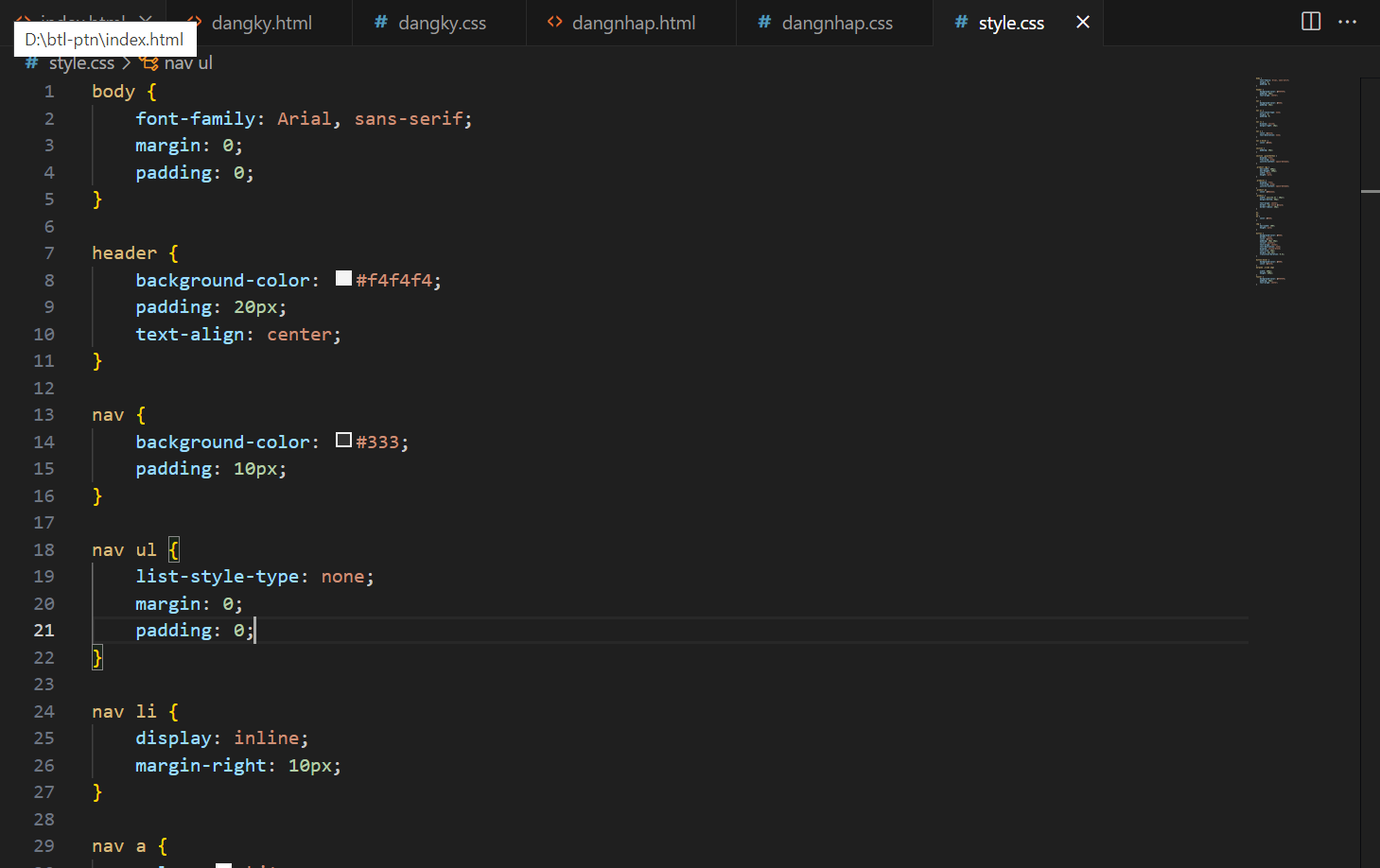
Thông tin bản quyền và chính sách bảo mật

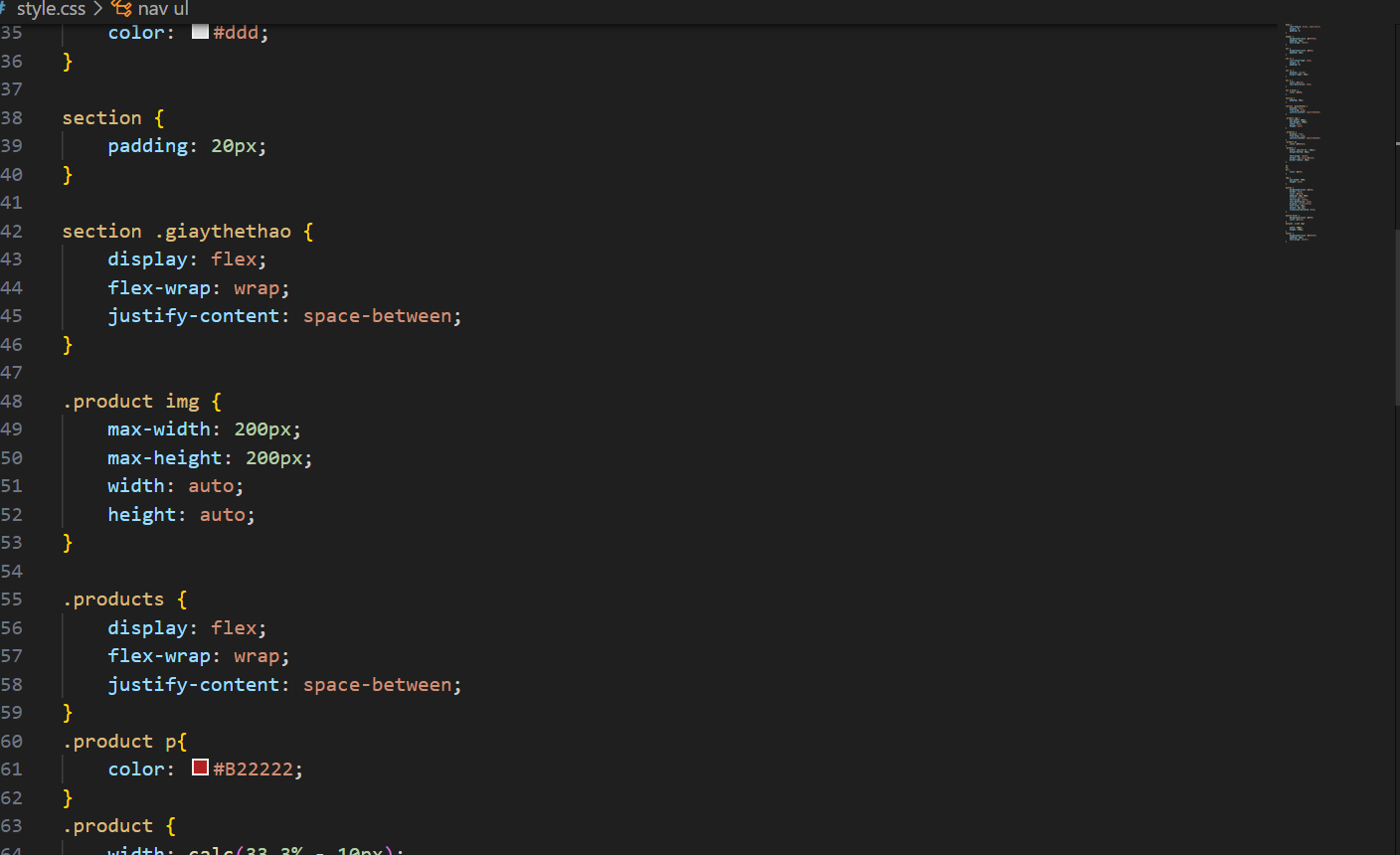
* 1. **Thiết kế trang Web bằng HTML và CSS**

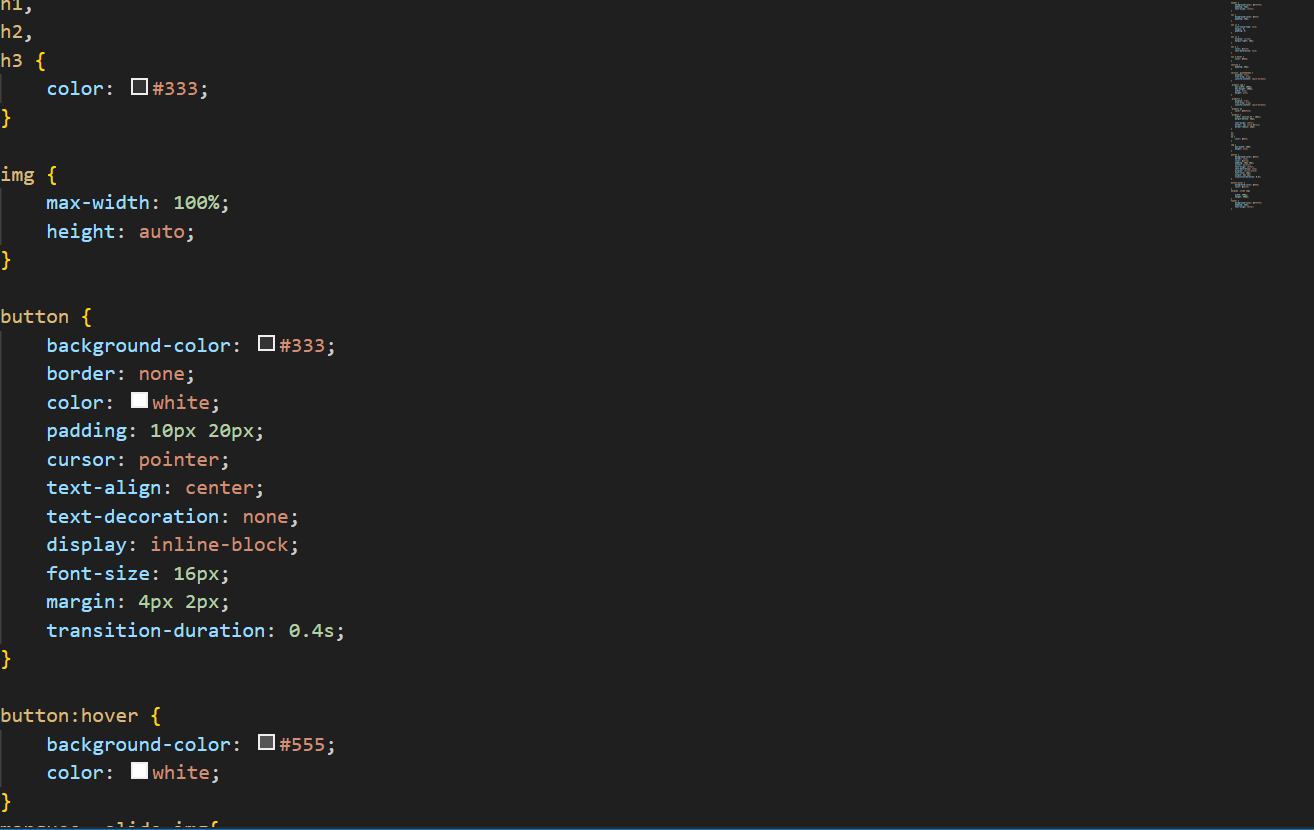
**HTML: **

****

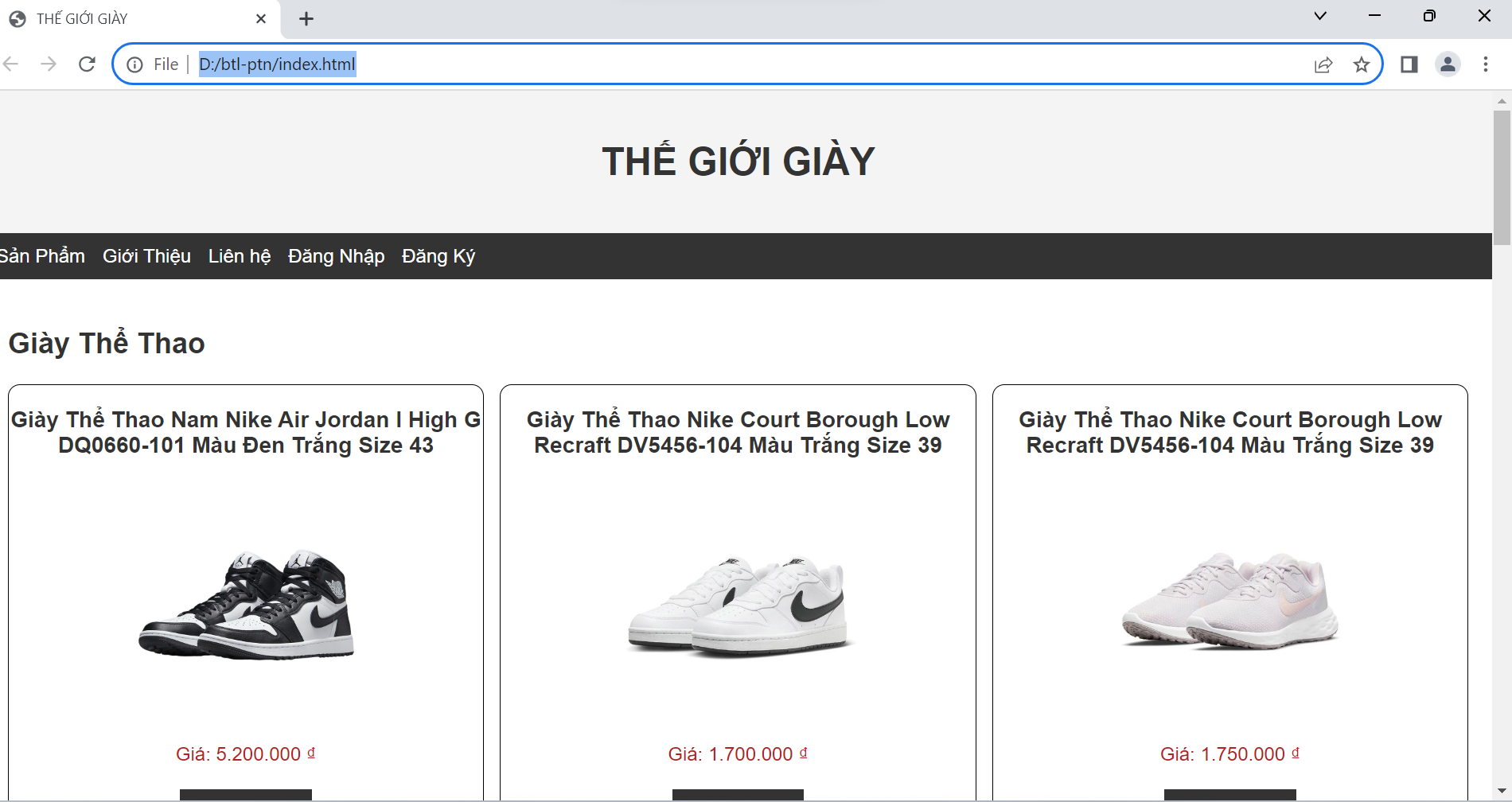
**CSS:**

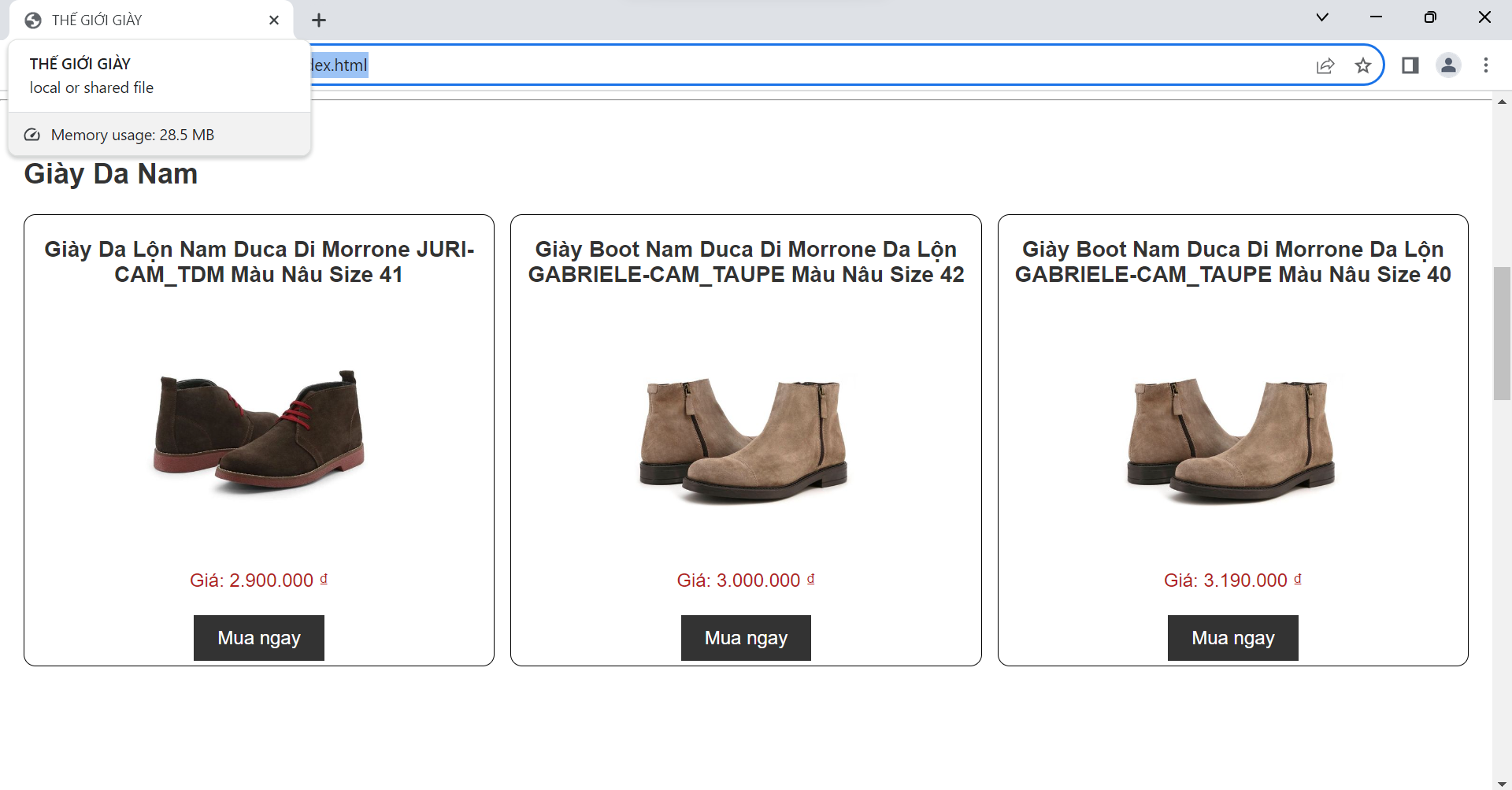
****

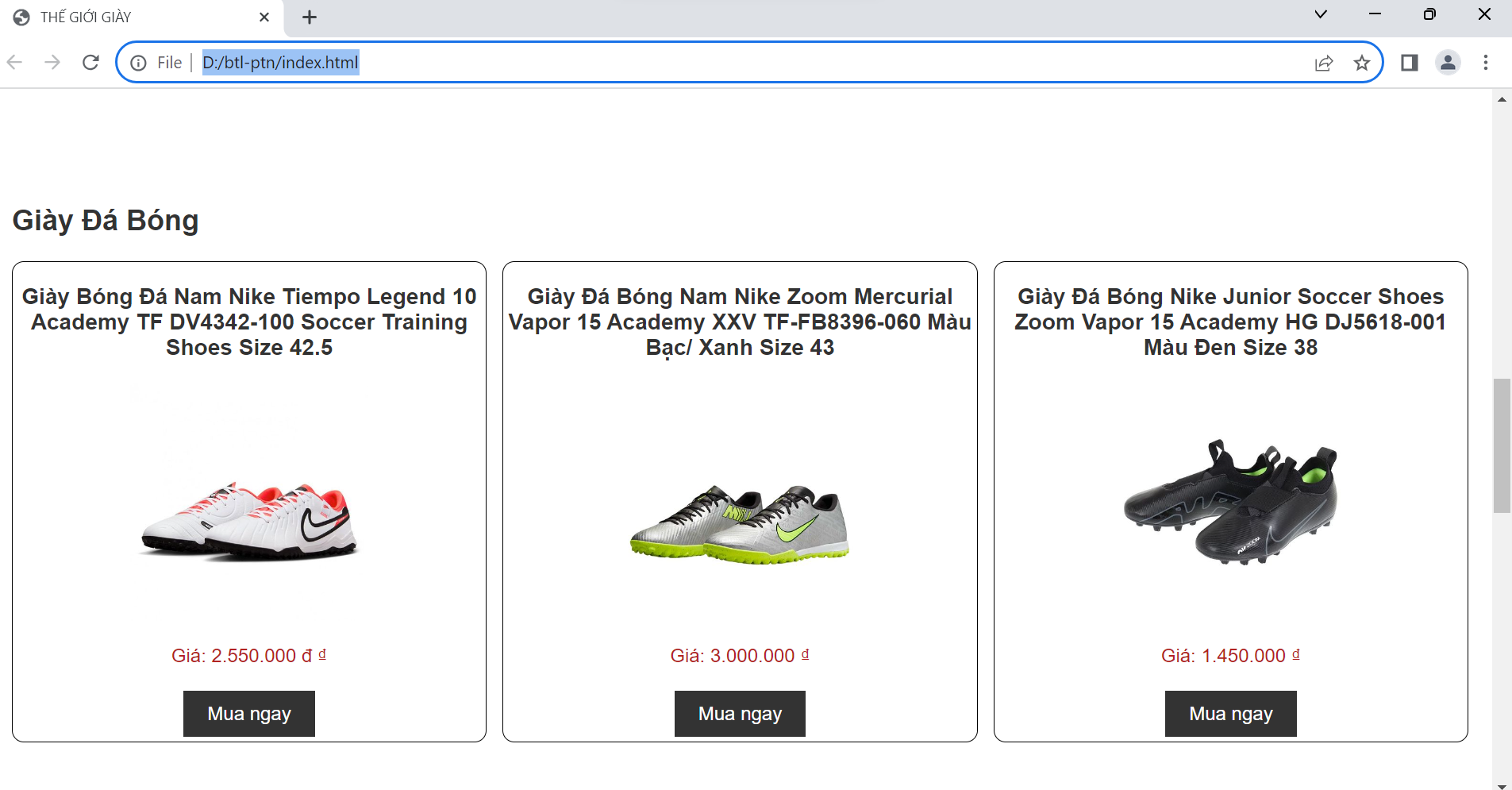
****

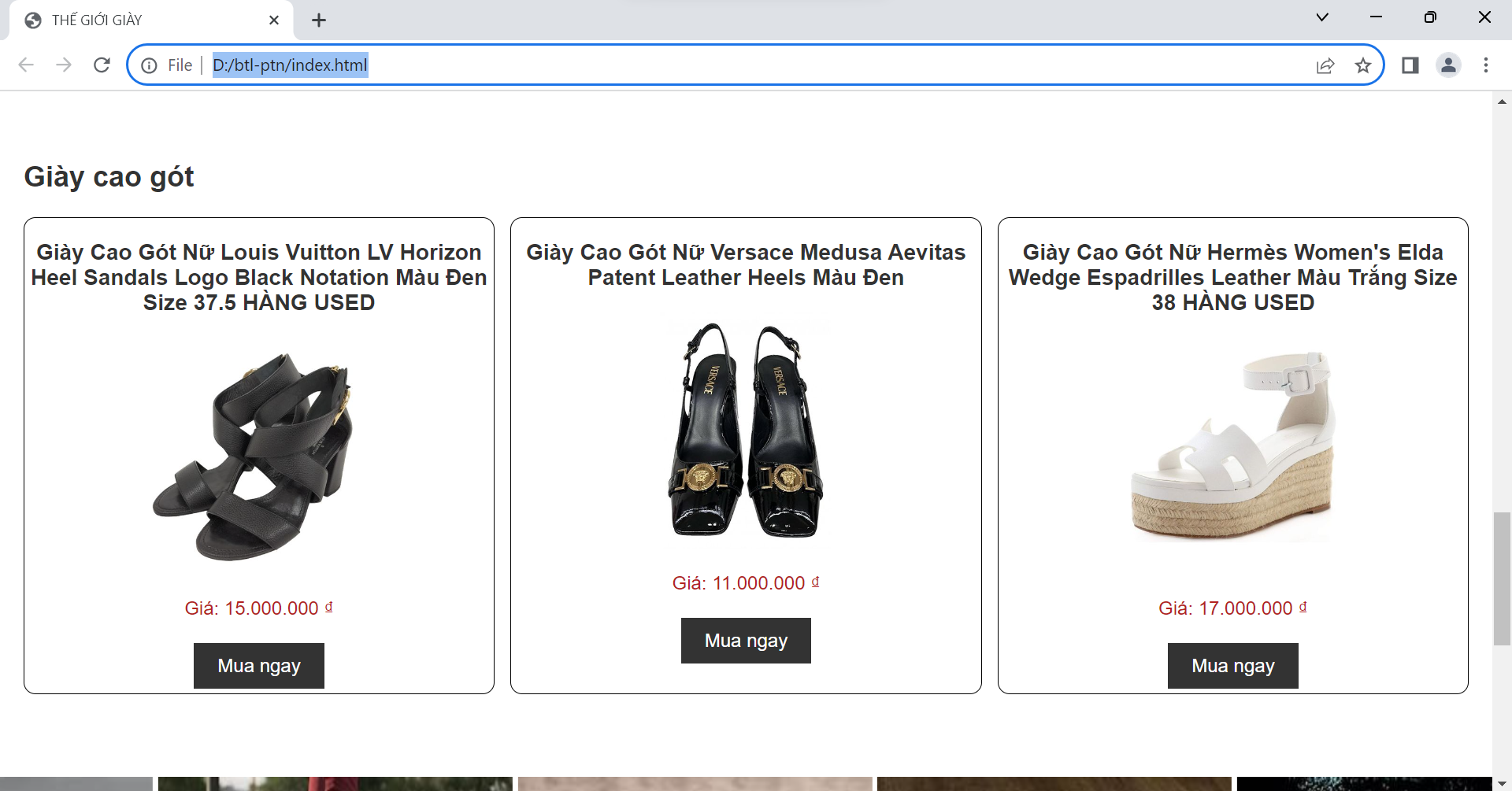
****

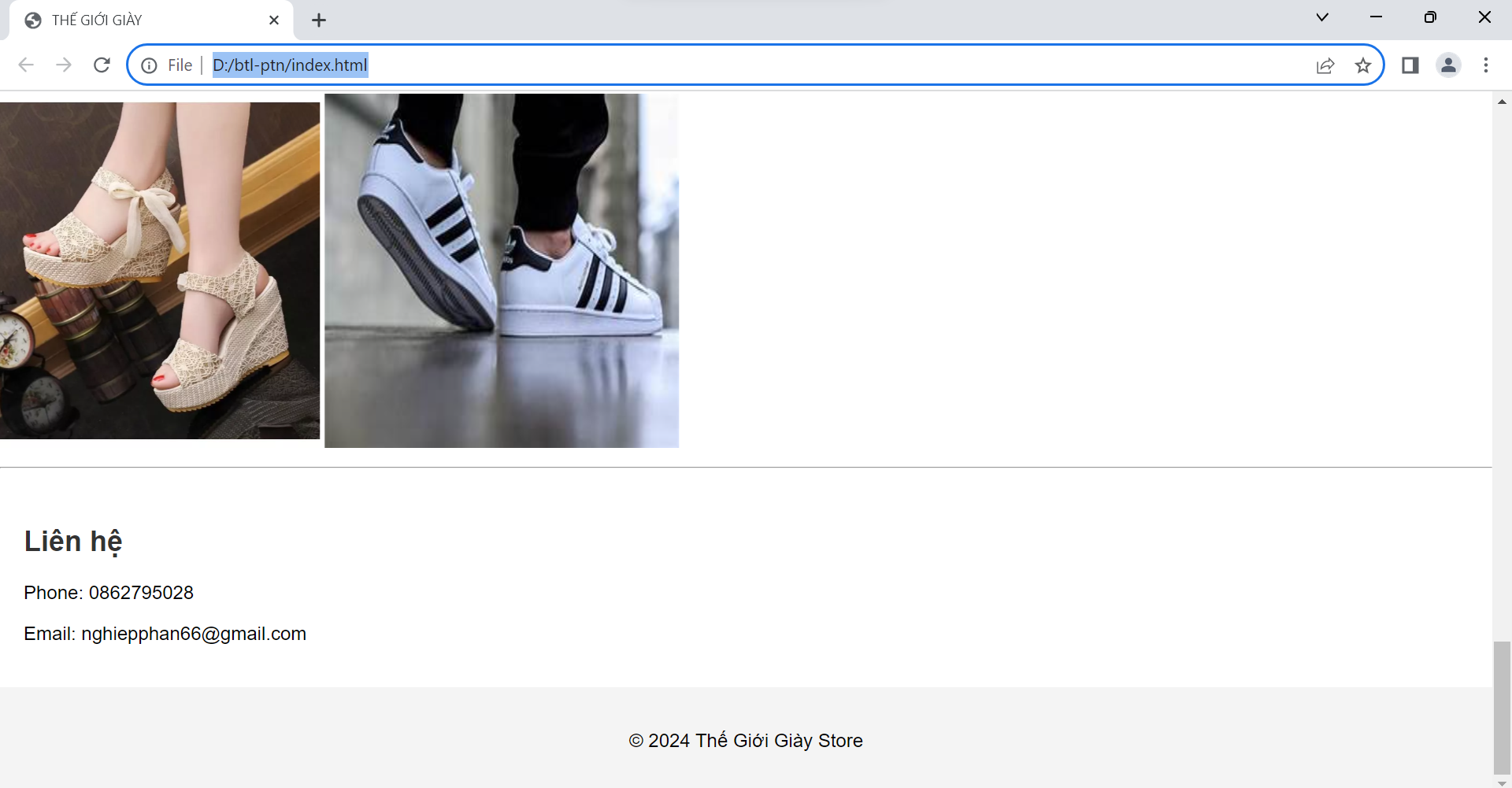
* 1. **Kết quả đạt được**

****

****

****

****

****

**Kết luận chương 4**

Trong ví dụ trên, chúng ta đã thấy cách sử dụng HTML và CSS để tạo ra một trang web giới thiệu và bán hàng. Bằng cách này, chúng ta có thể tạo ra một giao diện đơn giản nhưng hiệu quả, giúp người dùng dễ dàng tìm hiểu về nhiều loại giày và các thông tin liên quan.

Đương nhiên! Dưới đây là một ví dụ về cách giới thiệu một cửa hàng giày:

Chào mừng đến với Giày Chất Lượng -Nơi Sự Phong Phú Gặp Gỡ Phong Cách

Tại Giày Chất Lượng, chúng tôi tự hào là địa chỉ tin cậy cho những người yêu thích giày dép và mong muốn thể hiện phong cách cá nhân của mình. Với một loạt sản phẩm đa dạng và chất lượng từ những thương hiệu hàng đầu, chúng tôi cam kết đem đến cho bạn trải nghiệm mua sắm tốt nhất.

Chúng tôi không chỉ cung cấp những đôi giày đẹp mắt, mà còn chú trọng vào chất lượng và sự thoải mái. Bạn có thể tin tưởng rằng mỗi sản phẩm trong cửa hàng của chúng tôi đều được lựa chọn kỹ lưỡng và kiểm tra chất lượng một cách cẩn thận.

Tại Giày Chất Lượng, chúng tôi có mọi thứ từ các loại giày thể thao phong cách đến giày công sở thanh lịch và giày dép thoải mái cho mọi dịp. Dù bạn đang tìm kiếm một đôi giày cho các hoạt động thể thao, một đôi giày đi làm sang trọng, hay đơn giản là một đôi dép xỏ đi chơi cùng bạn bè, chúng tôi đều có sản phẩm phù hợp với nhu cầu của bạn.

Đặc biệt, dịch vụ khách hàng luôn là ưu tiên hàng đầu của chúng tôi. Đội ngũ nhân viên chuyên nghiệp và thân thiện sẽ luôn sẵn lòng hỗ trợ bạn trong quá trình mua sắm và đảm bảo bạn có trải nghiệm mua sắm thoải mái và thú vị nhất.

Hãy đến và khám phá bộ sưu tập đa dạng của chúng tôi tại cửa hàng hoặc truy cập trang web của chúng tôi ngay bây giờ để tìm kiếm những đôi giày hoàn hảo cho phong cách của bạn!

Hy vọng bạn thấy hữu ích! Nếu bạn cần thêm thông tin hoặc có bất kỳ câu hỏi nào, đừng ngần ngại để lại cho tôi biết.

**KẾT LUẬN**

***(Trình bày thành 3 đoạn văn nêu Ưu điểm, nhược điểm và hướng phát triển chủ đề)***

**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.