

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

**Logo

Description automatically generated**

**MÔN HỌC: CÔNG NGHỆ WEB**

**ĐỀ TÀI: THIẾT KẾ TRANG WEB BÁN QUẦN ÁO**

|  |  |
| --- | --- |
| **GIẢNG VIÊN HƯỚNG DẪN:** | **Nguyễn Thị Hạnh** |
| **SINH VIÊN THỰC HIỆN:** | 1. Đào Phương Anh 2. Vũ Thị Tính 3. Đào Quang Tuấn |

Hà Nội ,Tháng 10 năm 2023

**MỤC LỤC**

[**I.TỔNG QUAN VỀ CÔNG NGHÊ WEB** 1](#_Toc149292737)

[**1.Giới thiệu về HTML** 1](#_Toc149292738)

[**1.1.Hướng dẫn tạo một tệp HTML cơ bản** 1](#_Toc149292739)

[**1.2.Tạo trang web với tiêu đề, đoạn văn bản, hình ảnh và siêu liên kết.** 3](#_Toc149292740)

[**1.3.Cú pháp cơ bản** 5](#_Toc149292741)

[**2. Cơ bản về CSS** 10](#_Toc149292742)

[**2.1.Hướng dẫn cách thêm kiểu cho trang web bằng CSS** 10](#_Toc149292743)

[**2.2. Tạo các lớp và áp dụng kiểu bằng cách sử dụng lựa chọn** 10](#_Toc149292744)

[**2.3. Thử nghiệm với các thuộc tính CSS cơ bản như font chữ, màu sắc, độ rộng và chiều cao** 11](#_Toc149292745)

[**3.JavaScript Cơ bản** 13](#_Toc149292746)

[**3.1.Giới thiệu cú pháp Javascript cơ bản** 13](#_Toc149292747)

[**3.2.Tạo biến, hàm và sử dụng sự kiện** 13](#_Toc149292748)

[**3.3. Ví dụ** 14](#_Toc149292749)

[**4.TML Forms và Input Elements** 16](#_Toc149292750)

[**4.1.Hướng dẫn cách tạo biểu mẫu HTML đơn giản với các phần tử nhập liệu như ô văn bản, ô chọn, và nút gửi** 16](#_Toc149292751)

[**4.2. Tạo kiểu cho biểu mẫu bằng CSS** 17](#_Toc149292752)

[**4.3.Sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu** 18](#_Toc149292753)

[**5.Quy trình phát triển web** 21](#_Toc149292754)

[**5.1. Giới thiệu về quy trình phát triển web từ lập kế hoạch đến triển khai trang web** 21](#_Toc149292755)

[**5.2. Xác định các bước quan trọng như phân tích yêu cầu, thiết kế giao diện, phát triển, kiểm thử, và triển khai.** 23](#_Toc149292756)

[**5.3. Hướng dẫn về vai trò và trách nhiệm của từng thành viên trong nhóm, bao gồm quản lý dự án và phát triển.** 24](#_Toc149292757)

[**6. Thực hành xây dựng trang web đơn giản** 24](#_Toc149292758)

[**6.1 Xây dựng trang web đơn giản từ đầu bằng việc kết hợp HTML, CSS và JavaScript** 24](#_Toc149292759)

[**6.2. Xây dựng 1 website hoàn thiện (Web động có kết nối CSDL)** 29](#_Toc149292760)

[**7. Responsive Web Design** 32](#_Toc149292761)

[**7.1.Giới thiệu về cách tạo trang web** 32](#_Toc149292762)

[**7.2. Sử dụng các kỹ thuật CSS media queries để điều chỉnh giao diện cho các màn hình khác nhau.** 32](#_Toc149292763)

[**7.3 Hướng dẫn cách tạo bố cục đáng linh hoạt cho các thiết bị khác nhau** 33](#_Toc149292764)

[**8. HTML5 và CSS3** 36](#_Toc149292765)

[**8.1.Giới thiệu về các tính năng mới trong HTML5 và CSS3** 36](#_Toc149292766)

[**8.2 Hướng dẫn cách sử dụng các phần tử HTML5 như video, âm thanh, canvas, và đồ họa vector.** 37](#_Toc149292767)

[**8.3 Sử dụng CSS3 để thêm các hiệu ứng và chuyển động vào trang web, bao gồm cả CSS transitions và animations** 38](#_Toc149292768)

[**9. Thiết kế trang web thân thiện với SEO** 39](#_Toc149292769)

[**9.1.Giới thiệu về SEO** 39](#_Toc149292770)

[**9.2.Hướng dẫn cách tối ưu hóa tiêu đề trang, thẻ meta, và nội dung để cải thiện thứ hạng tìm kiếm** 41](#_Toc149292771)

[**9.3. Sử dụng công cụ tối ưu hóa SEO để theo dõi hiệu suất và tối ưu hóa trang web.** 42](#_Toc149292772)

[**10. Quản lý phiên bản và Git** 43](#_Toc149292773)

[**10.1. Giới thiệu về hệ thống quản lý mã nguồn và Git** 43](#_Toc149292774)

[**10.2. Hướng dẫn tạo kho lưu trữ (repository) Git và làm việc với các lệnh cơ bản như git clone, git pull, git push, và git commit** 45](#_Toc149292775)

[**10.3. Thảo luận về việc làm việc cùng nhóm trong môi trường Git và quản lý phiên bản dự án** 46](#_Toc149292776)

# **I.TỔNG QUAN VỀ CÔNG NGHÊ WEB**

Công nghệ web là một thuật ngữ chung đề cập đến nhiều ngôn ngữ và gói đa phương tiện được sử dụng kết hợp với nhau, để tạo ra các trang web. Mỗi công nghệ có chức năng riêng biệt và cần yêu cầu sử dụng kép ít nhất thêm một công nghệ khác. Do đó, có thể rằng tất cả các thành phần tạo nên một trang web phụ thuộc lẫn nhau.

## **1.Giới thiệu về HTML**

* HTML là chữ viết tắt của HyperText Markup Language, có nghĩa là ngôn ngữ đánh dấu siêu văn bản
* Nó dùng để định dạng bố cục, các thuộc tính liên quan đến cách hiển thị của một đoạn text và được hiển thị trên một chương trình đặc biệt ta gọi là Brower
* Hiện nay có khá nhiều Browser như Firefox, Chorme, Cốc Cốc,… Tất cả Browser đều có điểm chung là giúp người dùng thao tác với website và nó đều có khả năng biên dịch những đoạn mã HTML, CSS và Javascrip.

### **1.1.Hướng dẫn tạo một tệp HTML cơ bản**

Để tạo một tệp HTML cơ bản, bạn có thể làm theo các bước sau:

* Cách 1:
* Bước 1: Mở trình chỉnh sửa văn bản Bạn có thể sử dụng bất kỳ trình chỉnh sửa văn bản nào để tạo tệp HTML, ví dụ như Notepad trên Windows, TextEdit trên macOS hoặc gedit trên Linux.
* Bước 2: Tạo tập tin mới Trong trình chỉnh sửa văn bản, tạo một tệp mới và lưu nó với phần mở rộng ".html". Ví dụ: "index.html".
* Bước 3: Khai báo loại tài liệu Thêm đoạn mã sau vào đầu tệp HTML để khai báo loại tài liệu:

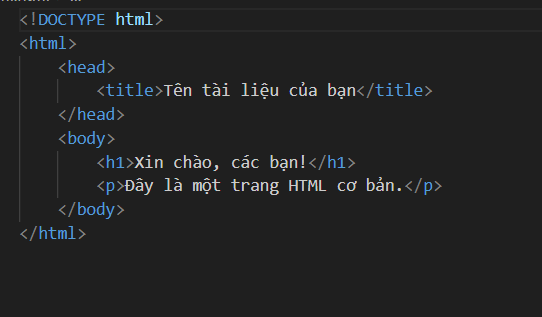
htmlSao chép mã

<!DOCTYPE html>

Đoạn mã trên cho biết trình duyệt hiểu rằng đây là một tài liệu HTML.

* Bước 4: Tạo phần tử cơ bản Thêm đoạn mã sau giữa cặp thẻ <html> và </html> để tạo một phần tử cơ bản trong tài liệu HTML:

htmlSao chép mã



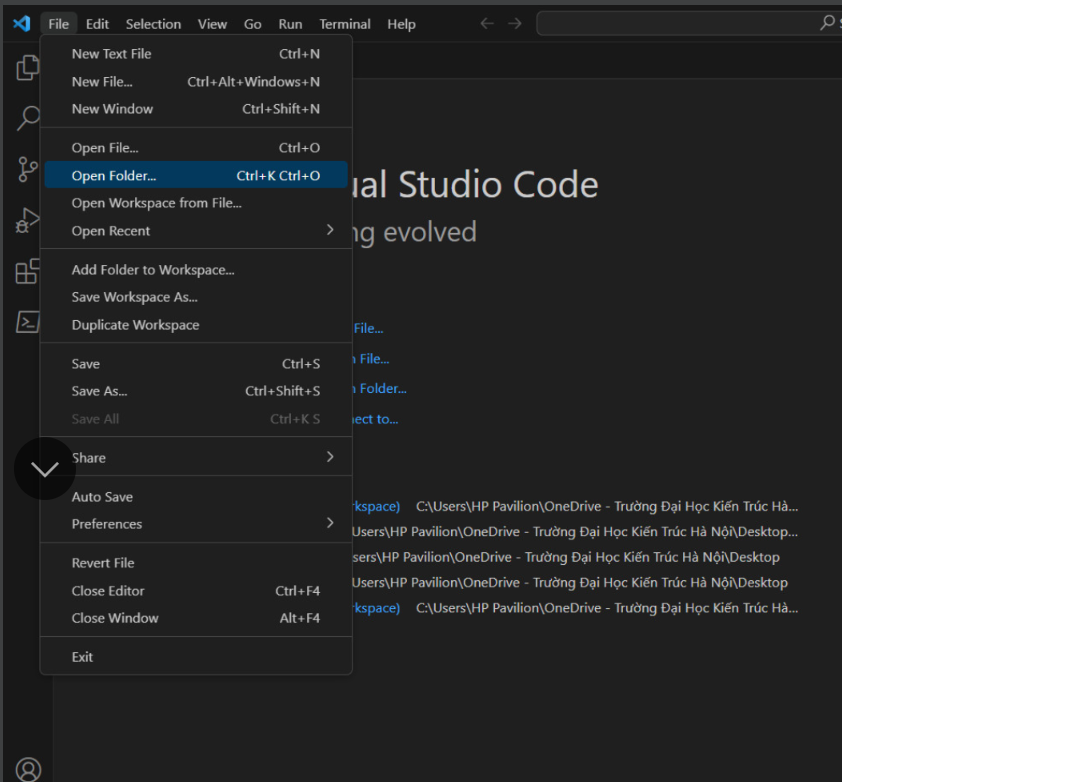
Trong ví dụ trên, chúng ta tạo một tiêu đề cấp 1 <h1> và một đoạn văn bản <p>.

* Bước 5: Lưu và mở tệp HTML Lưu tệp HTML và mở nó trong trình duyệt web để xem kết quả. Bạn có thể nhấp đúp vào tệp HTML hoặc kéo và thả tệp vào trình duyệt.

Chính sách an toàn đãi ngộ! Bây giờ bạn đã tạo thành công một tập tin HTML cơ bản.

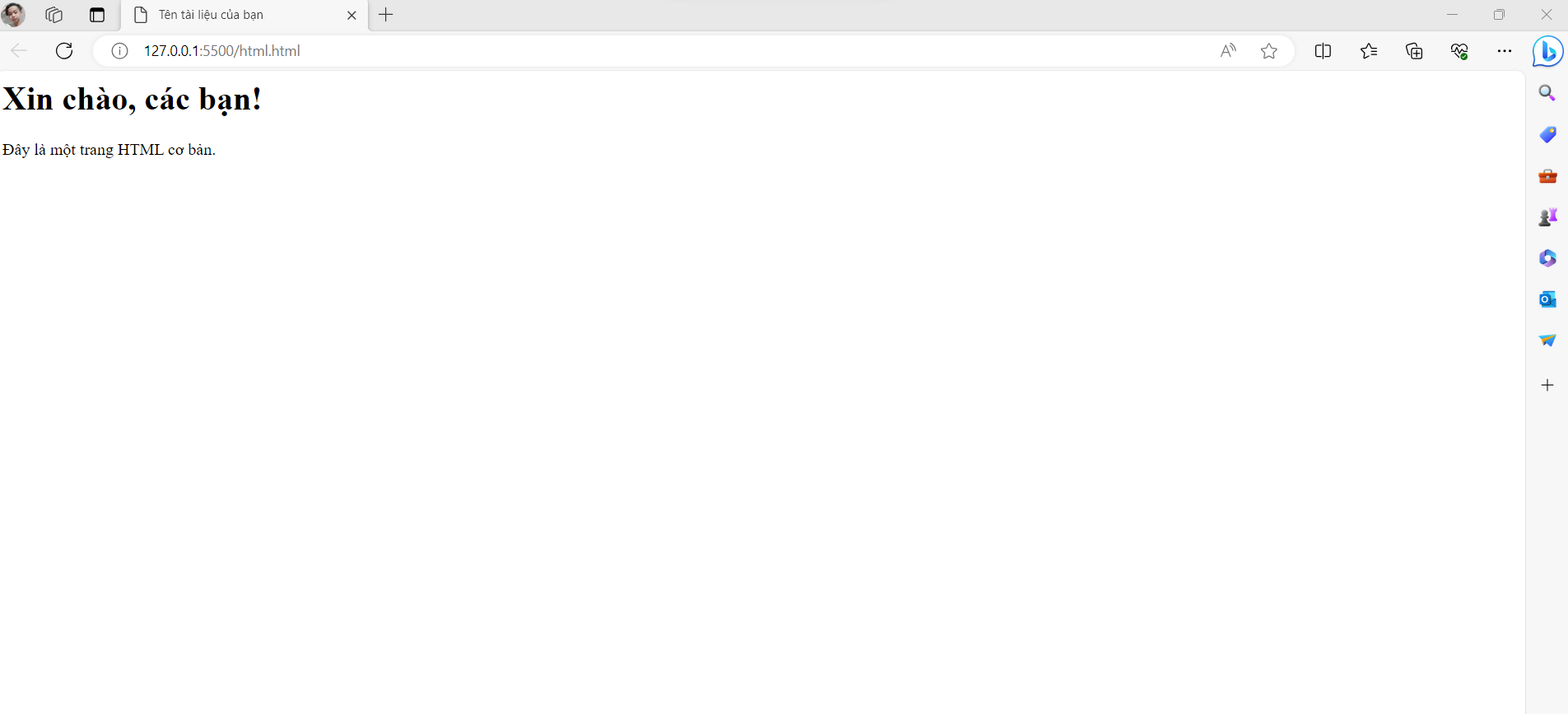
Chính sách an toàn đãi ngộ

* Cách 2:
* Bước 1: tạo 1 folder rỗng
* Bước 2 : vào visual studio code ->file->open folder



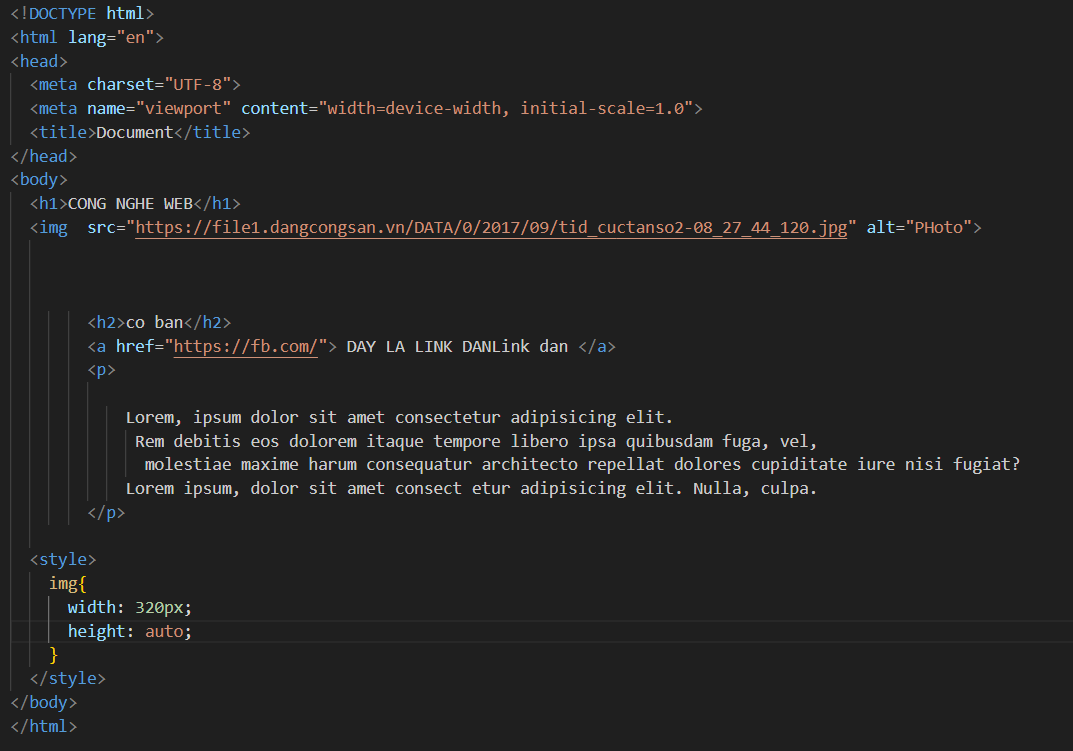
* Bước 3,Bước 4 ,Bước 5 tương tự cách 1

Chạy thử :

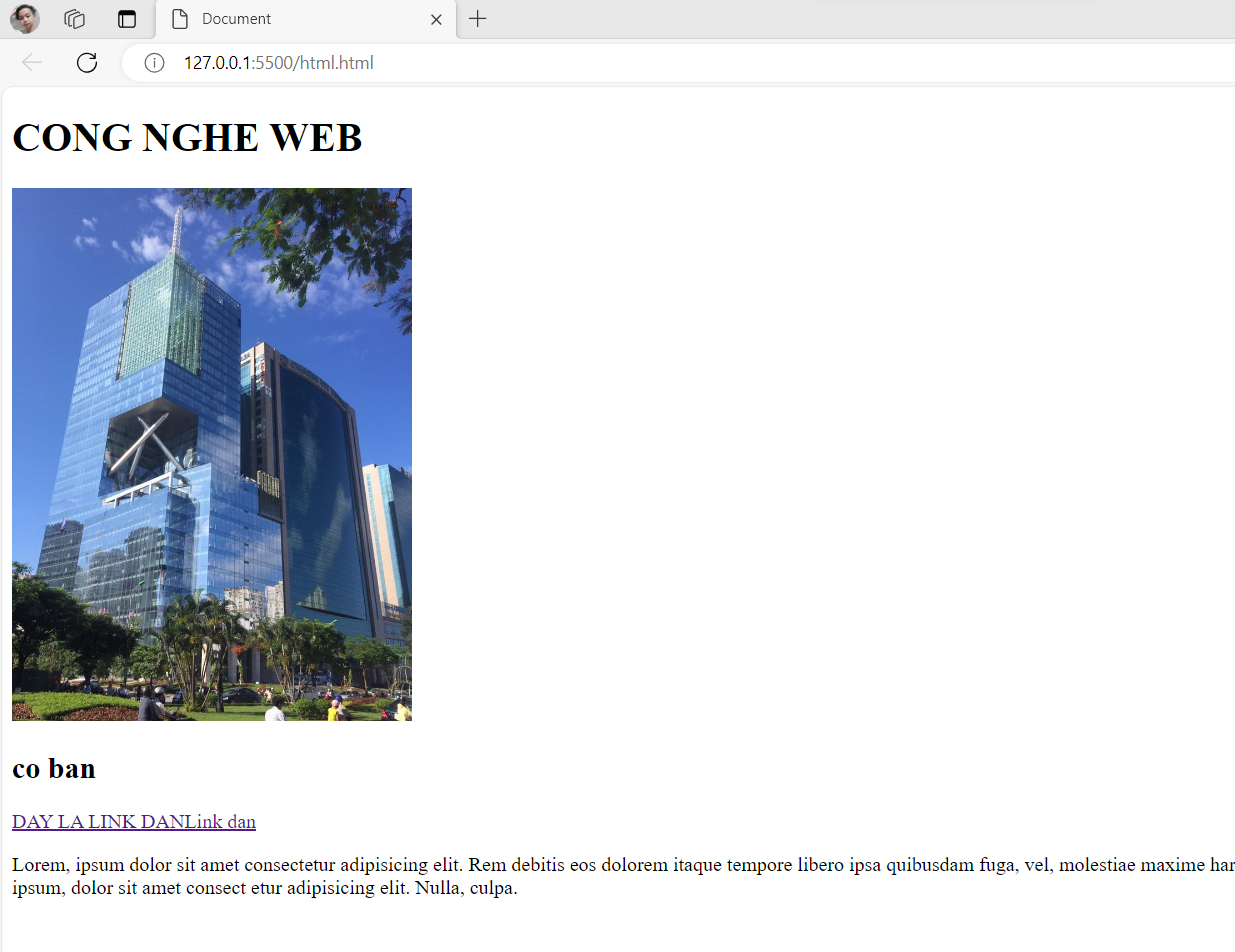


### **1.2.Tạo trang web với tiêu đề, đoạn văn bản, hình ảnh và siêu liên kết.**

+)Phần code:

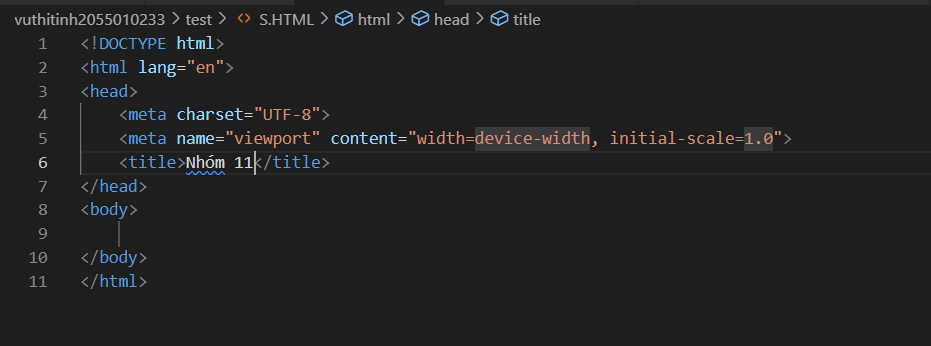


+)Chạy thử:



### **1.3.Cú pháp cơ bản**

Cấu trúc của file HTML cơ bản



Cấu trúc chung của html gồm có thẻ <!DOCTYPE html> dùng để xác định rằng code bạn đang dùng là html

* Cặp thẻ đóng mở <html> </html> chứa toàn bộ các dòng lệnh html bên trong (sau này còn là code của Javascript và CSS khi chèn vào)
* Tiếp đến là 2 cặp thẻ <head></head> chứa thông tin của webpage và đặc trưng bởi thẻ <title></title>, mà mặc định sẽ không được hiển thị trên web
* Thẻ <body></body> chứa toàn bộ nội dung html sẽ hiển thị lên trên trang web của chúng ta.

Danh sách tất cả các phần tử trong HTML

1.Basic

|  |  |
| --- | --- |
| Thẻ | Mô tả chức năng |
| [<!DOCTYPE>](https://webcoban.vn/html/the-doctype-trong-html.html) | Xác định cho trình duyệt biết phiên bản HTML mà bạn đang sử dụng |
| [<html>](https://webcoban.vn/html/the-html-trong-html.html) | Xác định một tài liệu HTML |
| [<head>](https://webcoban.vn/html/the-head-trong-html.html) | Xác định phần đầu của tài liệu HTML *(chứa các thẻ cung cấp thông tin cho trang web)* |
| [<title>](https://webcoban.vn/html/the-title-trong-html-tao-tieu-de-cho-trang-web.html) | Xác định tiêu đề của trang web |
| [<body>](https://webcoban.vn/html/the-body-trong-html.html) | Xác định phần thân của tài liệu HTML *(chứa những phần tử sẽ được hiển thị lên màn hình trình duyệt)* |
| [<h1> - <h6>](https://webcoban.vn/html/the-heading-h1-h6-trong-html.html) | Tạo những đề mục quan trọng trong trang web |
| [<p>](https://webcoban.vn/html/the-p-trong-html.html) | Xác định một đoạn văn bản |
| [<br>](https://webcoban.vn/html/the-br-trong-html-ngat-xuong-dong.html) | Chèn một ngắt xuống dòng |
| [<hr>](https://webcoban.vn/html/the-hr-trong-html.html) | Tạo một đường kẻ phân cách nằm ngang |

2.Formatting

|  |  |
| --- | --- |
| [<abbr>](https://webcoban.vn/html/the-abbr-trong-html.html) | Định nghĩa một từ viết tắt |
| [<address>](https://webcoban.vn/html/the-address-trong-html.html) | Xác định thông tin liên hệ của tác giả *(hoặc chủ sở hữu)* trang web |
| [<b>](https://webcoban.vn/html/the-b-trong-html-in-dam-van-ban.html) | Xác định đoạn văn bản được in đậm |
| [<s>](https://webcoban.vn/html/the-s-trong-html.html) | Tạo một đường kẻ ngang lên văn bản |
| [<strong>](https://webcoban.vn/html/the-strong-trong-html-in-dam-van-ban.html) | Xác định đoạn văn bản được in đậm |
| <time> | Đánh dấu những phần văn bản là: *thời gian, ngày tháng, ngày lễ, ....* |

3.Frames

|  |  |
| --- | --- |
| Thẻ | Mô tả chức năng |
| [<frame>](https://webcoban.vn/html/the-frame-trong-html.html) | Xác định một khung trong một bộ khung |
| [<iframe>](https://webcoban.vn/html/the-iframe-trong-html-nhung-mot-trang-web-khac-vao-trang-web-hien-tai.html) | Nhúng một trang web khác vào trang web hiện tại |
| [<frameset>](https://webcoban.vn/html/the-frameset-trong-html.html) | Xác định một bộ khung |
| [<noframes>](https://webcoban.vn/html/the-noframes-trong-html.html) | Xác định một nội dung sẽ được hiển thị khi trình duyệt không hỗ trợ thẻ <frame> |

4.Images

|  |  |
| --- | --- |
| Thẻ | Mô tả chức năng |
| [<img>](https://webcoban.vn/html/the-img-trong-html-chen-hinh-anh-vao-trang-web.html) | Chèn hình ảnh vào trang web |
| [<map>](https://webcoban.vn/html/the-map-va-area-trong-html-tao-mot-ban-do-hinh-anh.html) & [<area>](https://webcoban.vn/html/the-map-va-area-trong-html-tao-mot-ban-do-hinh-anh.html) | Tạo một bản đồ ảnh |
| [<figcaption>](https://webcoban.vn/html/the-figcaption-trong-html.html) | Tạo một tiêu đề cho nội dung được đặt bên trong phần tử <figure> |
| [<figure>](https://webcoban.vn/html/the-figure-trong-html.html) | Xác định một nội dung cần được tách biệt rõ ràng |

5.Audio / Video

|  |  |
| --- | --- |
| Thẻ | Mô tả chức năng |
| [<audio>](https://webcoban.vn/html/the-audio-trong-html-tao-mot-trinh-nghe-nhac-cho-trang-web.html) | Tạo một "trình phát nhạc" cho trang web |
| [<source>](https://webcoban.vn/html/the-source-trong-html.html) | Chỉ định tài nguyên cho trình nghe nhạc hoặc trình xem phim |
| [<track>](https://webcoban.vn/html/the-track-trong-html-tao-phu-de-cho-video.html) | Chèn một bản phụ đề vào video |
| [<video>](https://webcoban.vn/html/the-video-trong-html-tao-mot-trinh-xem-phim-cho-trang-web.html) | Tạo một "trình xem phim" cho trang web |

6.Links

|  |  |
| --- | --- |
| Thẻ | Mô tả chức năng |
| [<a>](https://webcoban.vn/html/the-a-trong-html-tao-mot-lien-ket.html) | Tạo một liên kết đến một tài liệu nào đó *(khi người dùng bấm vào liên kết thì sẽ được chuyển đến tài liệu đó)* |
| [<nav>](https://webcoban.vn/html/the-nav-trong-html.html) | Xác định một tập hợp các liên kết & thường được sử dụng kết hợp với CSS để tạo một thanh menu |

7.Lists

|  |  |
| --- | --- |
| Thẻ | Mô tả chức năng |
| <[ul](https://webcoban.vn/html/the-audio-trong-html-tao-mot-trinh-nghe-nhac-cho-trang-web.html)> | Xác định một danh sách không có thứ tự |
| <[ol](https://webcoban.vn/html/the-source-trong-html.html)> | Xác định một danh sách có thứ tự |
| <[li>](https://webcoban.vn/html/the-track-trong-html-tao-phu-de-cho-video.html) | Xác định một *"danh mục"* trong danh sách |

8.Tables

|  |  |
| --- | --- |
| Thẻ | Mô tả chức năng |
| [<table>](https://webcoban.vn/html/the-table-trong-html-cach-tao-bang.html) | Xác định phần tử là một cái bảng |
| [<caption>](https://webcoban.vn/html/the-caption-trong-html-tao-tieu-de-cho-bang.html) | Tạo tiêu đề cho bảng |
| [<th>](https://webcoban.vn/html/the-th-trong-html.html) | Xác định phần tử là một ô tiêu đề trong hàng |
| [<tr>](https://webcoban.vn/html/the-tr-trong-html-tao-mot-hang-trong-bang.html) | Xác định phần tử là một hàng trong bảng |
| [<td>](https://webcoban.vn/html/the-td-trong-html.html) | Xác định phần tử là một ô trong hàng |
| [<thead>](https://webcoban.vn/html/the-thead-trong-html.html) | Xác định những dòng nào thuộc *"phần đầu"* của bảng |
| [<tbody>](https://webcoban.vn/html/the-tbody-trong-html.html) | Xác định những dòng nào thuộc *"phần thân"* của bảng |

9.Styles and Semantics

|  |  |
| --- | --- |
| Thẻ | Mô tả chức năng |
| [<style>](https://webcoban.vn/html/the-style-trong-html.html) | Dùng để làm thùng chứa cho các đoạn mã CSS |
| [<div>](https://webcoban.vn/html/the-div-trong-html.html) | Nhóm các phần tử lại với nhau để tiện cho việc định dạng cũng như thiết kế bố cục của trang |
| [<header>](https://webcoban.vn/html/the-header-trong-html.html) | Xác định phần đầu của trang web |
| [<footer>](https://webcoban.vn/html/the-footer-trong-html.html) | Xác định phần chân của trang web |
| [<span>](https://webcoban.vn/html/the-span-trong-html.html) | Nhóm các phần tử nội tuyến lại với nhau để tiện cho việc định dạng CSS |

10.Meta Info

|  |  |
| --- | --- |
| Thẻ | Mô tả chức năng |
| [<head>](https://webcoban.vn/html/the-head-trong-html.html) | Xác định phần đầu của tài liệu HTML *(chứa các thẻ cung cấp thông tin cho trang web)* |
| [<meta>](https://webcoban.vn/html/the-meta-trong-html.html) | Cung cấp thêm *"thông tin về trang web"* cho trình duyệt và các công cụ tìm kiếm |
| [<base>](https://webcoban.vn/html/the-base-trong-html.html) | Xác định *"đường dẫn cơ sở"* trong trang web và kiểu mở liên kết mặc định |

11.Programming

|  |  |
| --- | --- |
| Thẻ | Mô tả chức năng |
| [<script>](https://webcoban.vn/html/the-script-trong-html.html) | Dùng để làm thùng chứa cho các câu lệnh JavaScript |
| [<noscript>](https://webcoban.vn/html/the-noscript-trong-html.html) | Xác định một nội dung HTML sẽ được hiển thị khi trình duyệt không hỗ trợ JavaScript hoặc đã tắt JavaScript |
| [<embed>](https://webcoban.vn/html/the-embed-trong-html.html) | Dùng để nhúng một *"tài liệu"* nào đó vào trang web |
| [<object>](https://webcoban.vn/html/the-object-trong-html.html) | Dùng để nhúng một *"tài liệu"* nào đó vào trang web |

## **2. Cơ bản về CSS**

### **2.1.Hướng dẫn cách thêm kiểu cho trang web bằng CSS**

* Cách 1: ở phần body trong file html ta them thẻ style

Với câu lệnh <style> câu lệnh css </style>

* Cách 2:ở folder ta tạo them 1 file css rồi ở phần head trong file html ta sử dụng câu lệnh

<link rel="stylesheet" type="text/css" href="TÊN FILE CSS">

**Cách 3: ghi thuộc tính css vào thẻ html (inline)**

****

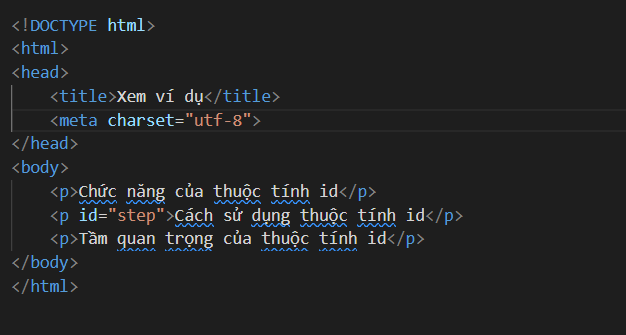
### **2.2. Tạo các lớp và áp dụng kiểu bằng cách sử dụng lựa chọn**

Trong ngôn ngữ HTML, thuộc tính ID và thuộc tính CLASS được sử dụng để đặt tên (phân loại) các phần tử, mục đích là để thuận tiện cho việc quản lí, định dạng các phần tử sau này.

1.Thuộc tính Class trong HTML   
 B1.Đặt tên cho lớp CSS  
 Chọn một tên đại diện cho tiêu chí hoặc chức năng của lớp CSS.  
 Sử dụng các từ ngữ có ý nghĩa để dễ hiểu và rõ ràng.  
B2.Viết mã CSS  
 Mở một khối CSS bằng cách sử dụng các dấu ngoặc nhọn ({}).  
 Trong khối CSS, áp dụng các thuộc tính và giá trị để định dạng các yếu tố HTML được gắn với lớp CSS.   
 Ví dụ: .my-class { color: blue; font-size: 16px; }  
B3.Áp dụng lớp CSS thông qua lựa chọn  
 Lựa chọn các phần tử HTML mà bạn muốn áp dụng lớp CSS.  
 Sử dụng cú pháp lựa chọn CSS để xác định các phần tử cần áp dụng lớp.  
 Ví dụ: h1.my-class { color: blue; } sẽ áp dụng lớp CSS .my-class cho tất cả các thẻ h1.  
B4.Gắn kết lớp CSS với phần tử HTML  
 Mở thẻ HTML mà bạn muốn gắn kết lớp CSS.  
 Sử dụng thuộc tính class và gán giá trị là tên của lớp CSS.  
 Ví dụ: <h1 class="my-class">Tiêu đề</h1>  
 Lưu ý rằng bạn có thể áp dụng một lớp CSS cho nhiều phần tử HTML bằng cách sử dụng cùng một tên lớp trong thuộc tính class. Điều này giúp tái sử dụng và quản lý kiểu dáng một cách hiệu quả.

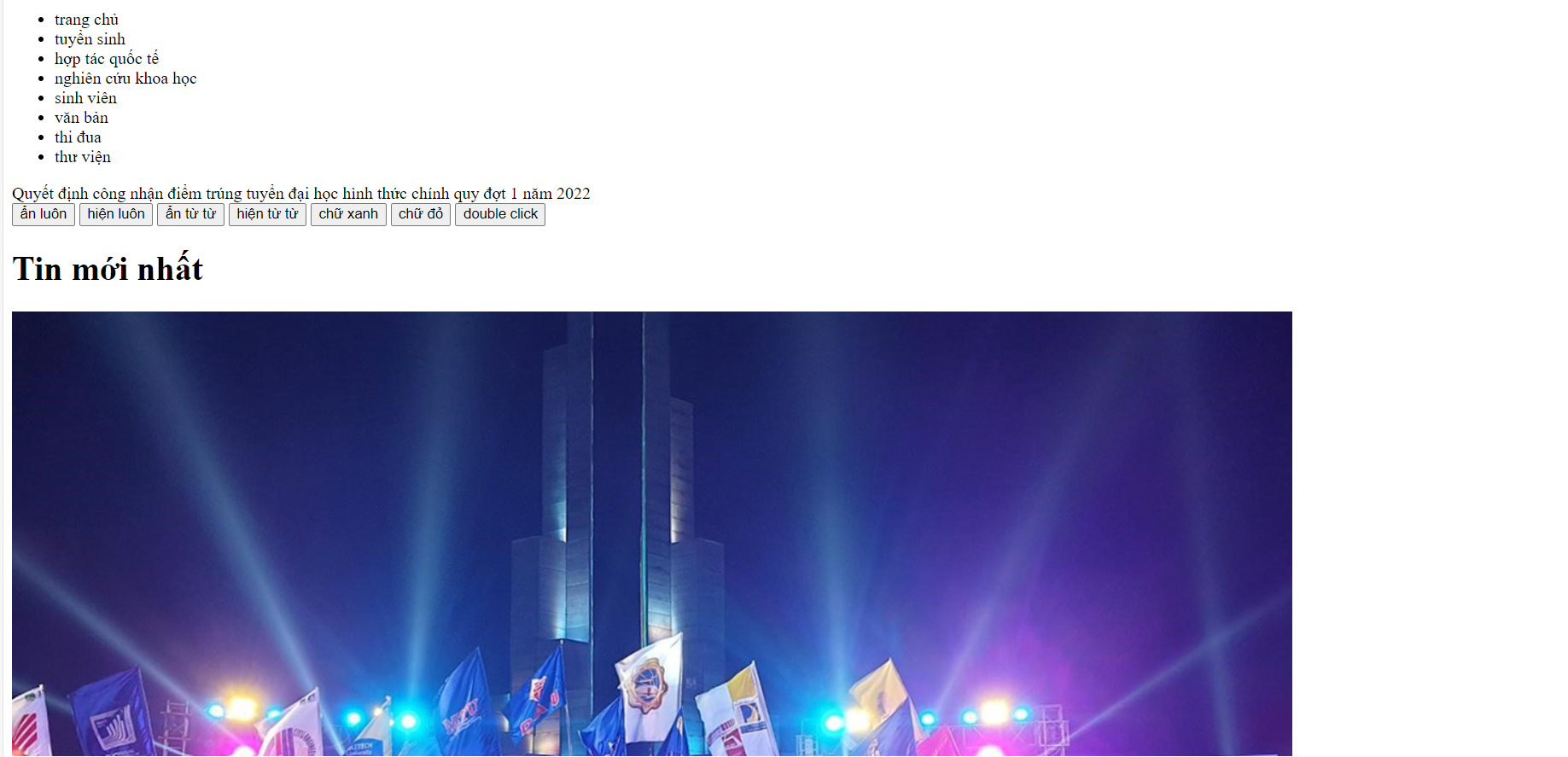
2.Thuộc tính ID trong HTML

* Thuộc tính id dùng để đặt tên cho phần tử, tên này phải là duy nhất, không có trường hợp tên id của các phần tử bị trùng nhau (nếu so sánh một phần tử trong trang web giống như một người công dân Việt Nam thì tên id cũng giống như số CMND, nó dùng để xác định danh tính của phần tử)
* Để khai báo id cho một phần tử thì chúng ta đặt thuộc tính id vào bên trong thẻ mở của phần tử đó với cú pháp id="tên id". Sau khi đã khai báo, nếu muốn định dạng cho phần tử thì chúng ta chỉ cần gọi thẳng tên id của nó với cú pháp #tên id



### **2.3. Thử nghiệm với các thuộc tính CSS cơ bản như font chữ, màu sắc, độ rộng và chiều cao**

* Không có css

****

* Có css

****

## **3.JavaScript Cơ bản**

### **3.1.Giới thiệu cú pháp Javascript cơ bản**

**1.Cú pháp Javascript**

* **Cú pháp của JavaScript** là tập hợp các quy tắc xác định chương trình JavaScript có cấu trúc chính xác.
* Một JavaScript bao gồm các câu lệnh JavaScript được đặt trong các thẻ HTML **<script> </script>** trong một trang web hoặc trong tệp JavaScript bên ngoài có phần mở rộng .js.

**2.Viết hoa - thường trong Javascript**

* **JavaScript phân biệt chữ hoa chữ thường.** Điều này có nghĩa là các biến, từ khóa ngôn ngữ, tên hàm và các mã định danh khác phải luôn được nhập với chữ viết nhất quán.
* Đặt tên theo quy tắc con lạc đà, chữ đầu viết thường chữ sau viết hoa

**3.Comment trong Javascript**

* Một comment (nhận xét) chỉ đơn giản là một dòng văn bản hoàn toàn bị bỏ qua bởi trình thông dịch JavaScript. (Có nghĩa là Comment đoạn code thì đoạn code đó sẽ không hoạt động nữa)
* Comment thường được thêm vào với mục đích cung cấp thêm thông tin liên quan để người đọc có thể hiểu code có ý nghĩa gì.
* Nó sẽ không chỉ giúp bạn hiểu code của bạn khi bạn làm việc với chúng trong một khoảng thời gian mà còn giúp các lập trình viên khác có thể hiểu bạn đang viết gì để phối hợp, sửa chữa, tiếp nhận.
* JavaScript hỗ trợ **comment dòng đơn** cũng như các **comment nhiều dòng. Comment một dòng** bắt đầu bằng dấu gạch chéo kép (//)

### **3.2.Tạo biến, hàm và sử dụng sự kiện**

1.Tạo biến

Để khai báo một biến ta sử dụng từ khóa const, let, var.

Ví dụ:

var foo = 50; //<= biến toàn cục

function test() {

var bar = 100; //<= biến cục bộ

}

2.Tạo hàm

Khai báo 1 hàm với từ khóa **function**, theo sau nó là tên của hàm đó cùng với 1 hoặc nhiều tham số (**parameters**) được tập hợp lại trong dấu ngoặc tròn (), cuối cùng là phần thân hàm sẽ được nhận biết thông qua cặp ngoặc nhọn {}.

Ví dụ:

function isEven(num) {

return num % 2 === 0;

}

isEven(24); // => true

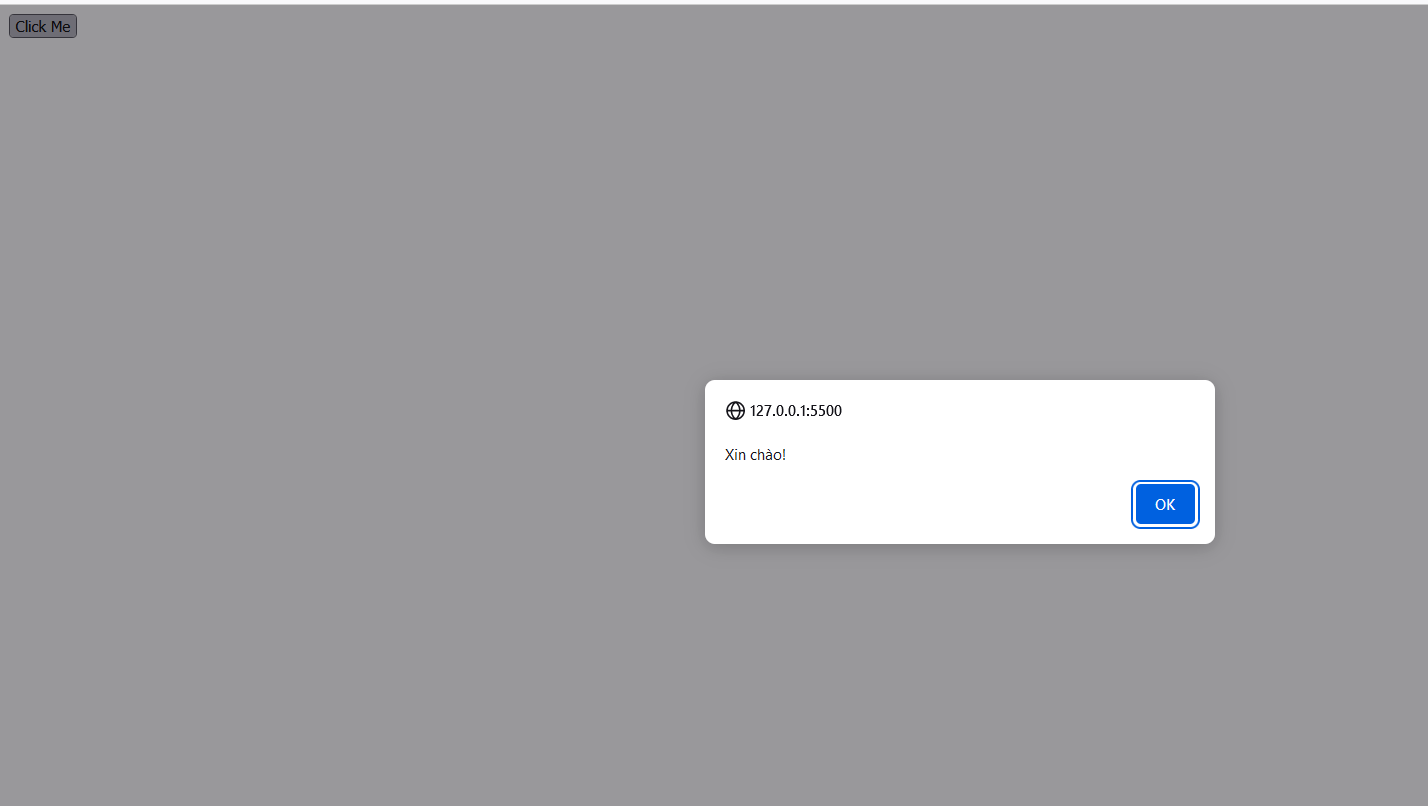
isEven(11); // => false

### **3.3. Ví dụ**

1.Phần code

****

2.Chạy thử

****

## **4.TML Forms và Input Elements**

### **4.1.Hướng dẫn cách tạo biểu mẫu HTML đơn giản với các phần tử nhập liệu như ô văn bản, ô chọn, và nút gửi**

Để tạo ra HTML Form thì dùng đến thẻ <form>, sau đó nội dung trong thẻ trình bày các HTML và các phần tử là điều khiển (control) có trong form.

<form action="http://xuanthulab.net/" method="post">

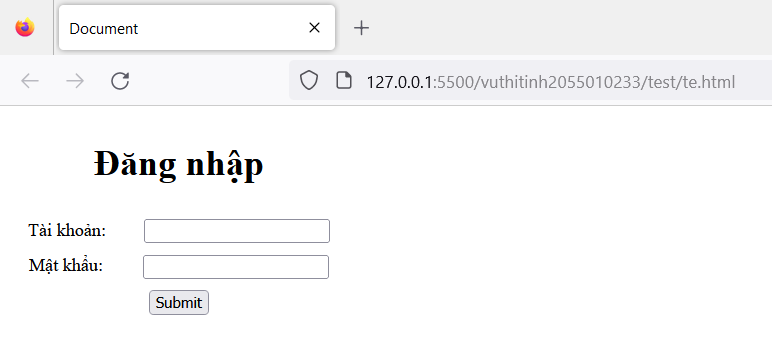
<!--Các mã HTML, các phần tử trong form -->

</form>

Thẻ <form> cơ bản có hai thuộc tính cần lưu tâm là action và method:

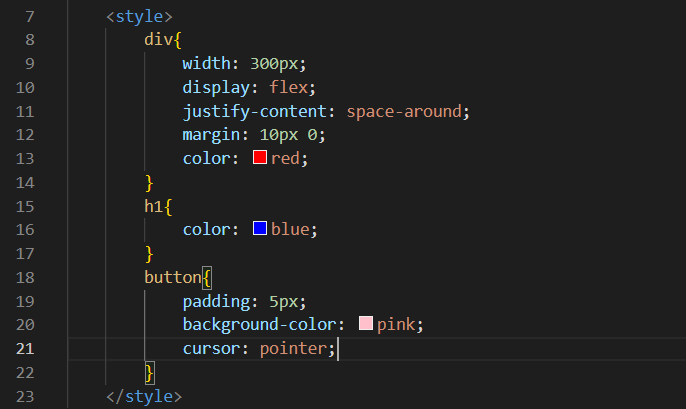
* Thuộc tính action trong form: thuộc tính để thiết lập URL sẽ nhận dữ liệu, là địa chỉ mà dữ liệu của form gửi đến (submit đến, post đến). Thiếu tham số này thì action bằng URL đang truy cập (tức gửi thông tin form đến server theo địa chỉ đang truy cập). Web server nhận được dữ liệu, xử lý và trả về nội dung nào đó.
* Thuộc tính method trong form: thuộc tính để thiết lập HTTP Method, xem thêm [HTTP Request Message](https://xuanthulab.net/giao-thuc-http-va-cau-truc-co-ban-cua-http-message.html#httprequest) thường có giá trị bằng get hoặc post. Nếu không viết thuộc tính này thì method mặc định của form là get
* Sử dụng method="get" thì khi submit dữ liệu được biểu diễn (encoding) thông qua URL (người dùng thấy rõ dữ liệu trên thanh địa chỉ trình duyệt khi nó gửi đi). Sử dụng method="post" thì khi submit biểu diễn trong nội dung của Http Request gửi đến Server và là ẩn với người dùng. Sử dụng post an toàn hơn.

Ví dụ:  
****

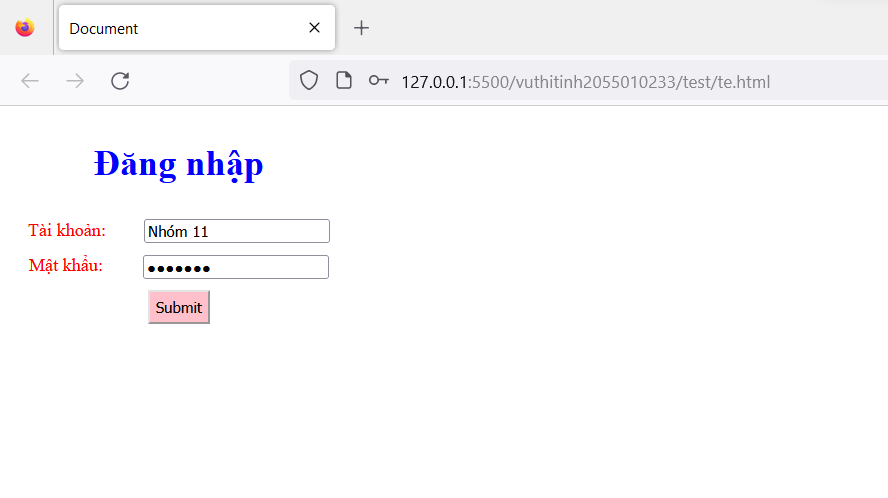
****

### **4.2. Tạo kiểu cho biểu mẫu bằng CSS**

1.Code

****

2.Chạy thử

****

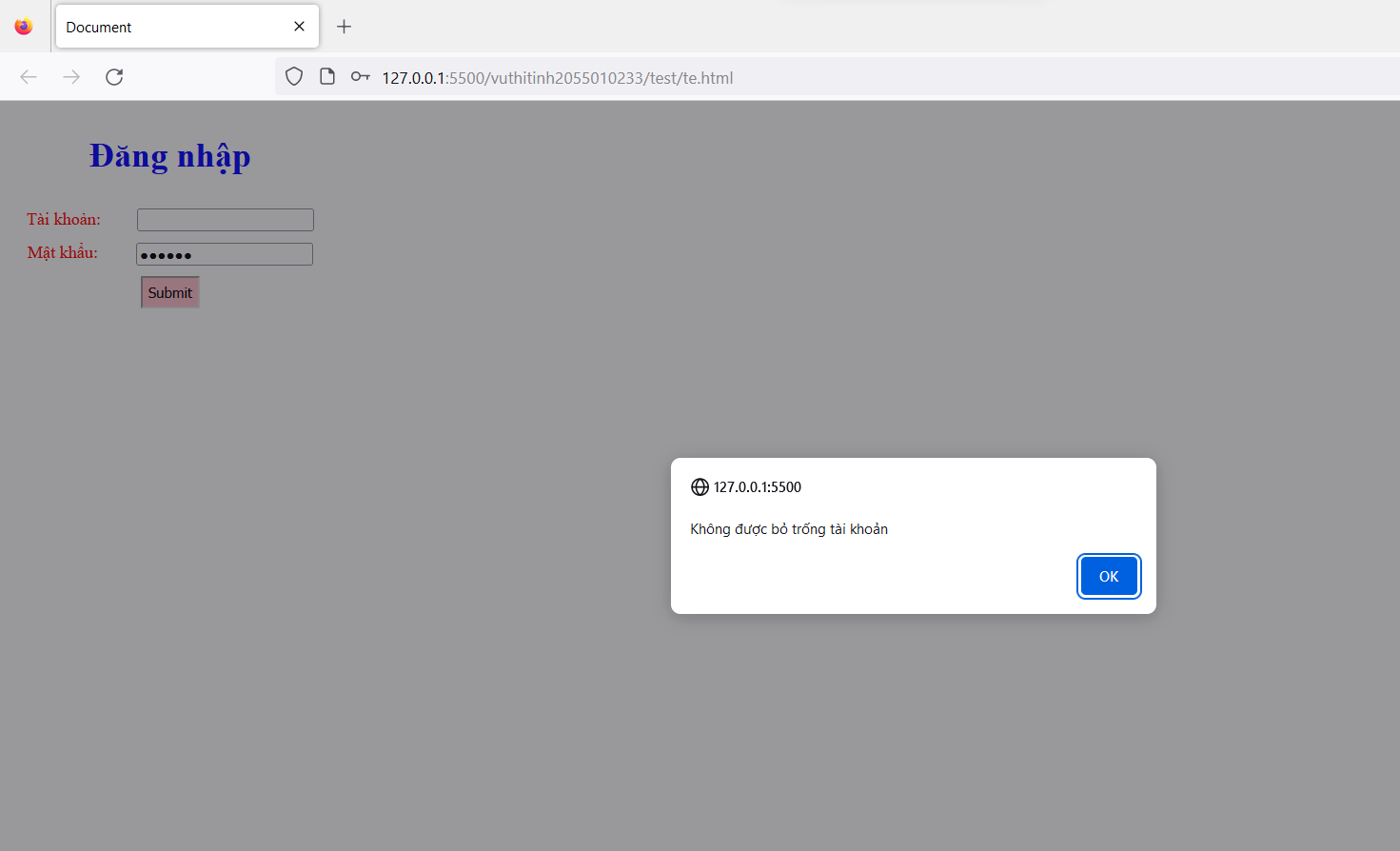
### **4.3.Sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu**

VD: tạo phần đăng nhập , Không được bỏ trống tài khoản và mật khẩu. Nếu mục nào bỏ trống sẽ đưa ra thông báo không được bỏ trống tài khoản hoặc không được bỏ trống mật khẩu ,đủ thông tin đăng nhập thành công

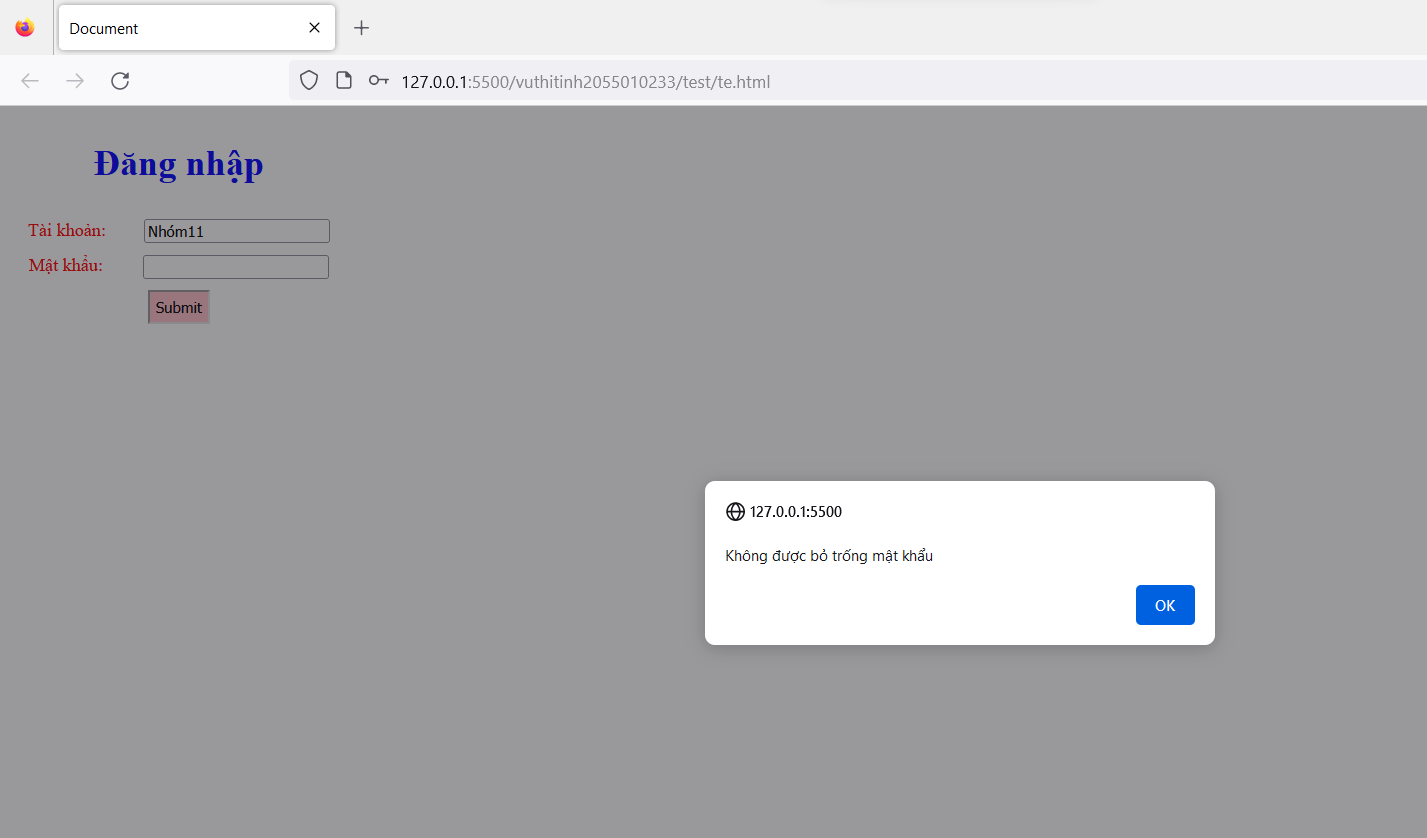
Phần code

Chạy thử

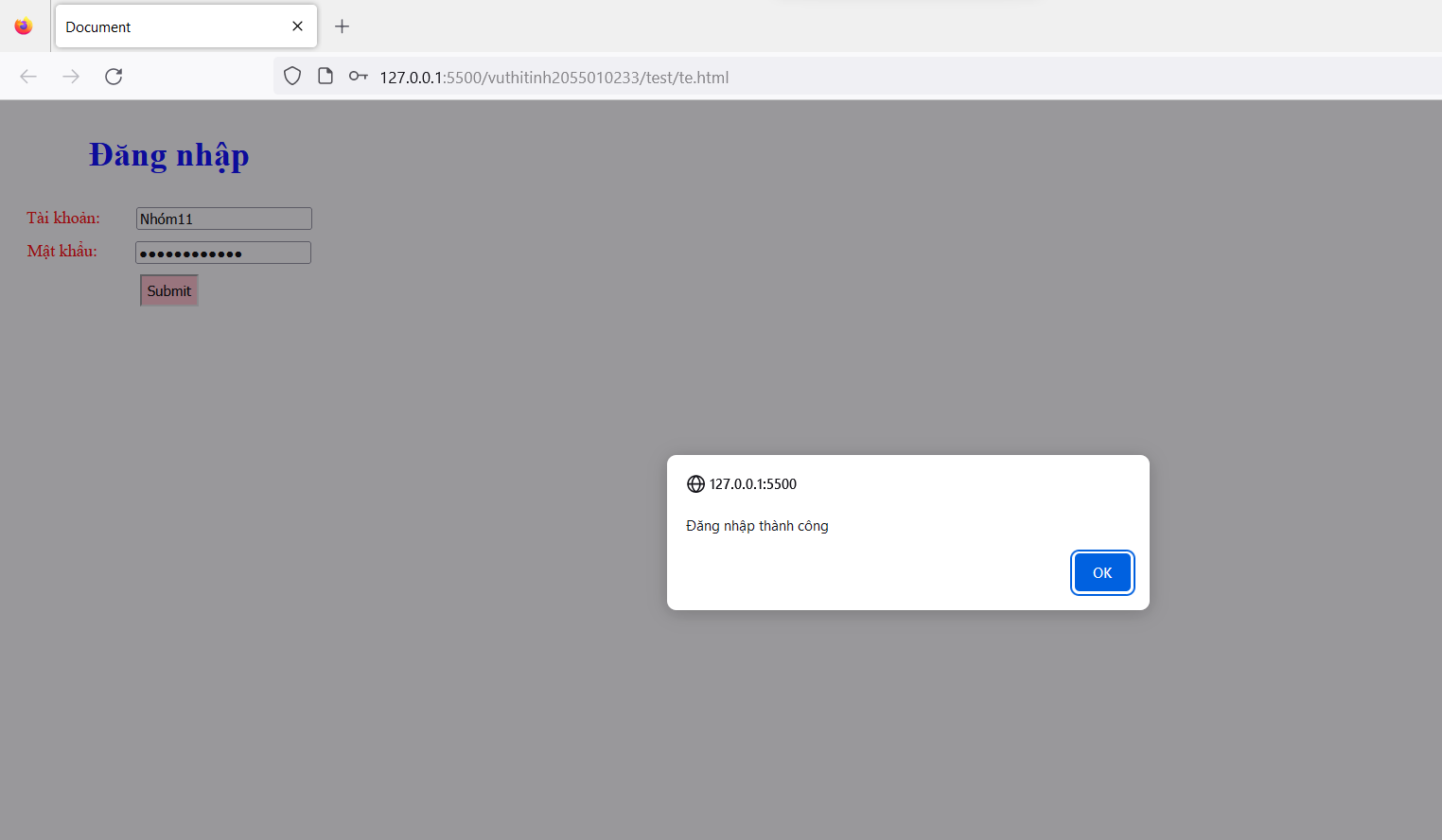
* Không nhập tài khoản

****

* Không nhập mật khẩu

****

* Nhập đủ thông tin

****

## **5.Quy trình phát triển web**

### **5.1. Giới thiệu về quy trình phát triển web từ lập kế hoạch đến triển khai trang web**

Quy trình phát triển web từ lập kế hoạch đến triển khai trang web bao gồm các bước sau:

B1.Lập kế hoạch:

+)Xác định mục tiêu và yêu cầu của dự án.

+)Nắm vững nhu cầu người dùng và đối tượng mục tiêu.

+)Phân tích và thiết kế giao diện, chức năng cần có trên trang web.

+)Xây dựng lịch trình và ngân sách cho quá trình phát triển.

B2.Thu thập thông tin:

+)Thu thập thông tin về nội dung cần hiển thị trên trang web.

+)Tìm hiểu về công nghệ và xu hướng sử dụng trong việc phát triển trang web.

+)Đặt các yêu cầu kỹ thuật như hệ điều hành, cơ sở dữ liệu, ngôn ngữ lập trình,...

B3.Thiết kế:

+)Tạo wireframe hoặc mockup để minh họa cấu trúc và bố cục của trang web.

+)Xác định các yếu tố thiết kế như màu sắc, font chữ, hình ảnh,...

+)Xây dựng giao diện người dùng (UI) và trải nghiệm người dùng (UX).

B4.Phát triển:

+)Chọn một ngôn ngữ lập trình phù hợp và framework để viết mã.

+)Xây dựng cơ sở dữ liệu và kết nối với trang web.

+)Phát triển tính năng, chức năng và các module cần thiết.

+)Kiểm thử, sửa lỗi và tối ưu hoá mã nguồn.

B5.Kiểm thử:

+)Thử nghiệm toàn diện các tính năng và chức năng của trang web.

+)Kiểm tra tích hợp và tương tác giữa các hệ thống.

+)Sửa lỗi và đảm bảo trang web hoạt động một cách ổn định trên nhiều trình duyệt và +)thiết bị khác nhau.

B6.Triển khai:

+)Đưa trang web lên môi trường sản phẩm thực tế (production environment).

+)Cấu hình máy chủ, tên miền và các yêu cầu hệ thống khác.

+)Đăng ký và xác minh SSL (Secure Sockets Layer) nếu cần thiết.

+)Backup và lưu trữ mã nguồn và cơ sở dữ liệu.

+)Sau quá trình triển khai, trang web đã sẵn sàng để được sử dụng và tiếp tục quản lý, bảo trì trong tương lai.

### **5.2. Xác định các bước quan trọng như phân tích yêu cầu, thiết kế giao diện, phát triển, kiểm thử, và triển khai.**

Các bước quan trọng trong quy trình phát triển web từ lập kế hoạch đến triển khai trang web gồm:  
B1.Phân tích yêu cầu:  
 • Xác định mục tiêu và yêu cầu của dự án.  
 • Thu thập thông tin về nhu cầu người dùng và đối tượng mục tiêu.  
 • Đặt các yêu cầu kỹ thuật như hệ điều hành, cơ sở dữ liệu, ngôn ngữ lập trình,...  
B2.Thiết kế giao diện:  
 • Tạo wireframe hoặc mockup để minh họa cấu trúc và bố cục của trang web.  
 • Xác định các yếu tố thiết kế như màu sắc, font chữ, hình ảnh,...  
 • Xây dựng giao diện người dùng (UI) và trải nghiệm người dùng (UX).  
B3.Phát triển:  
 • Chọn một ngôn ngữ lập trình phù hợp và framework để viết mã.  
 • Xây dựng cơ sở dữ liệu và kết nối với trang web.  
 • Phát triển tính năng, chức năng và các module cần thiết.  
 • Kiểm thử, sửa lỗi và tối ưu hoá mã nguồn.  
B4.Kiểm thử:  
 • Thử nghiệm toàn diện các tính năng và chức năng của trang web.  
 • Kiểm tra tích hợp và tương tác giữa các hệ thống.  
 • Sửa lỗi và đảm bảo trang web hoạt động một cách ổn định trên nhiều trình duyệt và thiết bị khác nhau.  
B5.Triển khai:  
 • Đưa trang web lên môi trường sản phẩm thực tế (production environment).  
 • Cấu hình máy chủ, tên miền và các yêu cầu hệ thống khác.  
 • Đăng ký và xác minh SSL (Secure Sockets Layer) nếu cần thiết.  
 • Backup và lưu trữ mã nguồn và cơ sở dữ liệu.  
Các bước này là quan trọng trong việc đảm bảo rằng trang web được phát triển theo yêu cầu, có giao diện hấp dẫn và chất lượng, hoạt động một cách ổn định và an toàn khi triển khai.

### **5.3. Hướng dẫn về vai trò và trách nhiệm của từng thành viên trong nhóm, bao gồm quản lý dự án và phát triển.**

Vai trò và trách nhiệm của từng thành viên:

1.Đào Phương Anh:

2. Vũ Thị Tính

3.Đào Quang Tuấn

## **6. Thực hành xây dựng trang web đơn giản**

### **6.1 Xây dựng trang web đơn giản từ đầu bằng việc kết hợp HTML, CSS và JavaScript**

**6.1.1.HTML**

**Hình ảnh khi có html k có css**

****

Bước đầu tiên để xây dựng một trang web đơn giản là xác định cấu trúc trang và các thành phần cơ bản như head, menu, content và footer. Dưới đây là một ví dụ về cách bạn có thể cấu trúc trang web :

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Tên trang web của bạn</title>

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

<script src="script.js"></script>

</head>

<body>

<header>

<!-- Phần đầu trang, có thể chứa logo và menu -->

<h1>Tên trang web của bạn</h1>

<nav>

<ul>

<li><a href="#">Trang chủ</a></li>

<li><a href="#">Giới thiệu</a></li>

<li><a href="#">Liên hệ</a></li>

</ul>

</nav>

</header>

<main>

<!-- Nội dung chính của trang web -->

<section>

<h2>Tiêu đề phần nội dung 1</h2>

<p>Nội dung phần 1</p>

</section>

<section>

<h2>Tiêu đề phần nội dung 2</h2>

<p>Nội dung phần 2</p>

</section>

</main>

<footer>

<!-- Phần chân trang, có thể chứa thông tin bản quyền và liên kết liên quan -->

<p>Bản quyền © 2023 Tên trang web của bạn</p>

</footer>

</body>

</html>

Trong ví dụ trên, ta sử dụng các thẻ HTML như `<header>`, `<nav>`, `<main>`, `<section>`, và `<footer>` để định nghĩa cấu trúc trang web. Bạn cũng có thể sử dụng các thẻ khác theo nhu cầu của riêng bạn.

Lưu ý rằng các tệp CSS và JavaScript cần được liên kết trong thẻ `<head>` để áp dụng kiểu và tương tác cho trang web. Bạn cần tạo các tệp CSS và JavaScript tương ứng và liên kết chúng trong trang HTML.

Sau khi xác định cấu trúc trang và các thành phần cơ bản như vậy, bạn có thể sử dụng CSS và JavaScript để thiết kế và tạo tính năng cho các thành phần này.

Áp dụng vào trang web bán hàng:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Lele Shop</title>

<link rel="shortcut icon" href="./Pictures/Lele.png">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css" integrity="sha512-z3gLpd7yknf1YoNbCzqRKc4qyor8gaKU1qmn+CShxbuBusANI9QpRohGBreCFkKxLhei6S9CQXFEbbKuqLg0DA==" crossorigin="anonymous" referrerpolicy="no-referrer" />

<link rel="stylesheet" href="./Leleshop.css">

</head>

<body>

<div class="header">

Gồm Các thẻ I kết nối với facebook Instagram twitter

</div>

<div class="header-main">

Gồm Logo ,Thanh tìm kiếm ,Đăng nhập ,Giỏ hàng

</div>

<div class="navigation">

Thanh menu sử dụng

</div>

<div class="advertisement">

Hiển thị Hình ảnh quảng cáo

</div>

<div class="category">

Hiển thị Tổng hợp số lượng hàng

</div>

<div class="product">

Hiển thị các sản phẩm

</div>

<div class="detail">

Hiển thị các sản phẩm giảm giá

</div>

<div class="testimonial">

Hiển thị bình luận của khách hàng

</div>

<div class="trend">

Hiển thị các tin tức thời trang

</div>

<div class="footer-nav">

Hiển thị menu dịch vụ thông tin liên hê

</div>

<div class="footer-bottom">

bản quyền  và tên trang web

</div>

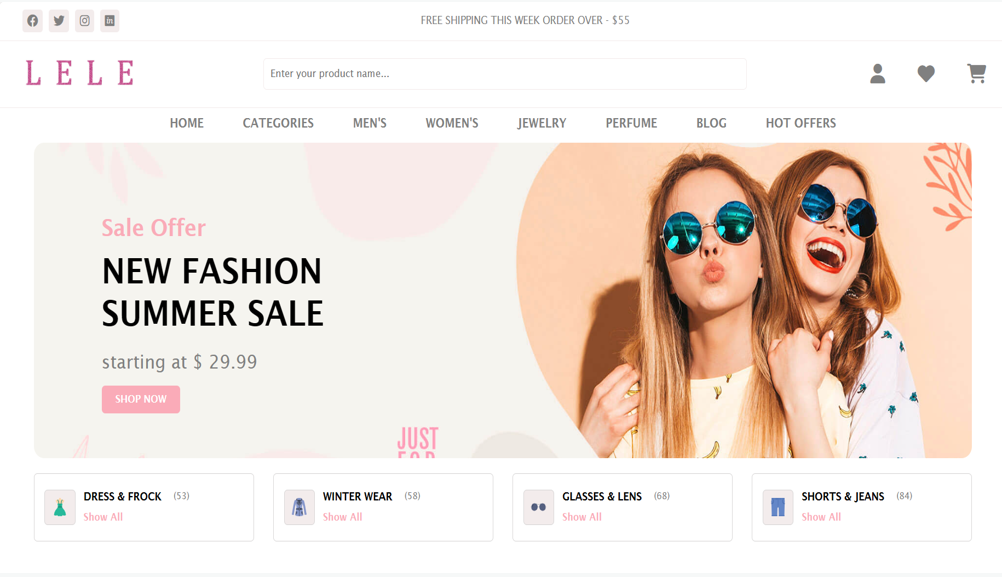
</body>

</html>

**6.1.2.CSS**

Chọn bố cục (layout) phù hợp: Sử dụng CSS để xác định bố cục của trang web, bao gồm kích thước, vị trí, và căn chỉnh của các phần tử. Đảm bảo bố cục hợp lý và dễ dùng cho người dùng.

Màu sắc và hình ảnh: Sử dụng CSS để áp dụng màu sắc hài hòa, tương phản và thích hợp cho trang web của bạn. Hãy cân nhắc việc sử dụng hình ảnh và biểu đồ để làm cho trang web của bạn thêm sinh động và hấp dẫn.

Sử dụng các hiệu ứng và chuyển động: CSS cung cấp nhiều hiệu ứng và chuyển động như hover, transition và animation. Sử dụng chúng để tạo hiệu ứng ấn tượng và làm cho trang web của bạn trở nên bắt mắt hơn.

Responsive Design: Đảm bảo trang web của bạn đáp ứng tốt trên các thiết bị khác nhau như điện thoại di động, máy tính bảng và máy tính để bàn. Sử dụng CSS media queries để điều chỉnh giao diện cho từng kích thước màn hình khác nhau.

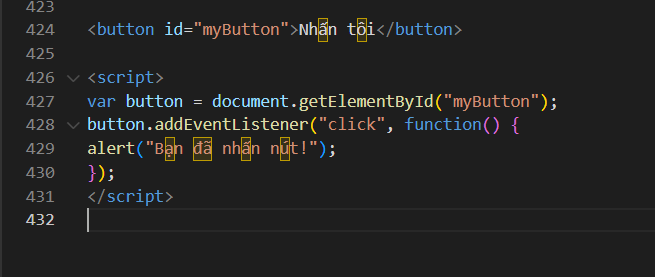
Đảm bảo đọc dễ dàng: Sử dụng CSS để tăng cường khả năng đọc và đảm bảo rằng nội dung trên trang web của bạn dễ nhìn và dễ đọc. Chú ý đến kích thước và kiểu chữ, khoảng cách và độ tương phản của văn bản.

Thử nghiệm và tinh chỉnh: Luôn thử nghiệm trang web của bạn trên các trình duyệt và thiết bị khác nhau để đảm bảo rằng giao diện vẫn giữ được tính thống nhất và hấp dẫn trên mọi nền tảng.

6.1.3 JAVASCRIP:

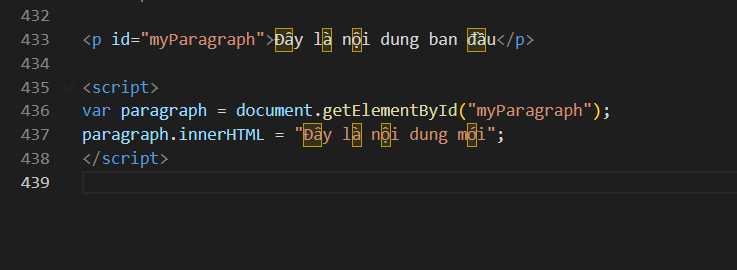
xử lý sự kiện nhấn nút, thay đổi nội dung động, và kiểm tra dữ liệu nhập từ người dung:

Xử lý sự kiện khi nhấn nút:



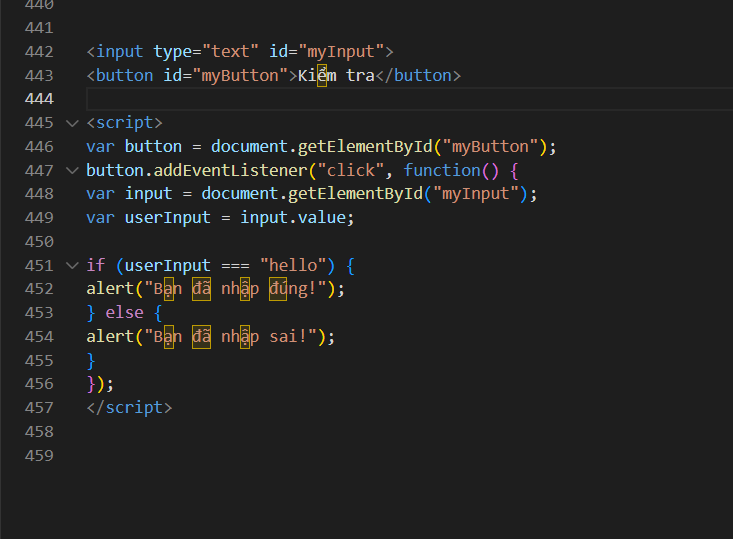
Trong ví dụ trên, chúng ta sử dụng JavaScript để lắng nghe sự kiện khi người dùng nhấn nút và xuất hiện thông báo.

Thay đổi nội dung động:



Trong ví dụ này, chúng ta sử dụng JavaScript để thay đổi nội dung của một phần tử HTML theo mong muốn.

Kiểm tra dữ liệu nhập từ người dùng:



### **6.2. Xây dựng 1 website hoàn thiện (Web động có kết nối CSDL)**

**MONGODB:**



MongoDB là một chương trình cơ sở dữ liệu NoSQL hướng tài liệu, mã nguồn mở phổ biến. Nó sử dụng các tài liệu giống JSON với các lược đồ tùy chọn và được phân loại là cơ sở dữ liệu NoSQL vì nó không dựa vào cấu trúc cơ sở dữ liệu quan hệ dựa trên bảng truyền thống.

Các tính năng chính của MongoDB bao gồm:

* Định hướng tài liệu: MongoDB lưu trữ dữ liệu trong các tài liệu linh hoạt, giống JSON.
* Truy vấn đặc biệt: MongoDB hỗ trợ truy vấn động trên tài liệu bằng ngôn ngữ truy vấn dựa trên tài liệu mạnh gần bằng SQL.
* Lập chỉ mục:Bất kỳ trường nào trong tài liệu MongoDB đều có thể được lập chỉ mục.
* Sao chép:MongoDB cung cấp tính sẵn sàng cao với các bộ bản sao.
* Cân bằng tải:MongoDB chia tỷ lệ theo chiều ngang bằng cách sử dụng sharding.

Tập hợp: MongoDB cung cấp một khung tổng hợp dựa trên khái niệm về quy trình xử lý dữ liệu.

* Lưu trữ tệp:Nó có thể được sử dụng để lưu trữ các tệp lớn bằng GridFS.

Kiến trúc MongoDB bao gồm các cơ sở dữ liệu chứa các bộ sưu tập, từ đó chứa các tài liệu. Mỗi tài liệu có thể khác nhau và cấu trúc dữ liệu có thể được thay đổi theo thời gian. Tính linh hoạt này là một trong những lợi thế chính của việc sử dụng MongoDB, đặc biệt đối với các ứng dụng yêu cầu khả năng mở rộng và quản lý dữ liệu theo thời gian thực.

MongoDB được nhiều tổ chức sử dụng cho nhiều ứng dụng khác nhau, bao gồm lưu trữ dữ liệu cho hệ thống quản lý nội dung, ứng dụng di động, phân tích thời gian thực, v.v. Tính linh hoạt, khả năng mở rộng và dễ sử dụng của nó khiến nó trở thành lựa chọn phổ biến của nhiều nhà phát triển và tổ chức.

**NODE.JS:**



Node.js là môi trường thời gian chạy đa nền tảng, mã nguồn mở, cho phép bạn chạy mã JavaScript bên ngoài trình duyệt web. Nó cho phép các nhà phát triển sử dụng JavaScript để viết kịch bản phía máy chủ, từ đó cho phép họ tạo nội dung trang web động trước khi trang được gửi đến trình duyệt web của người dùng. Node.js sử dụng mô hình I/O không chặn, hướng sự kiện, giúp nó nhẹ và hiệu quả cho các ứng dụng thời gian thực.

Một số tính năng chính của Node.js bao gồm:

* Không đồng bộ và hướng sự kiện: Node.js hoạt động trên vòng lặp sự kiện một luồng, cho phép nó xử lý nhiều kết nối đồng thời mà không cần tạo các luồng riêng biệt cho mỗi yêu cầu. Điều này làm cho nó rất phù hợp cho các ứng dụng yêu cầu tương tác thời gian thực hoặc truyền dữ liệu.
* Khả năng mở rộng: Node.js được biết đến với khả năng xử lý số lượng lớn kết nối đồng thời một cách hiệu quả, khiến nó trở thành lựa chọn lý tưởng cho các ứng dụng yêu cầu khả năng mở rộng.
* Công cụ JavaScript V8: Node.js được xây dựng trên thời gian chạy JavaScript V8, cùng một công cụ hỗ trợ Google Chrome. Điều này cho phép Node.js thực thi mã JavaScript với hiệu suất cao.
* NPM (Trình quản lý gói nút): NPM là trình quản lý gói mạnh mẽ đi kèm với Node.js. Nó cung cấp quyền truy cập vào hệ sinh thái rộng lớn gồm các gói nguồn mở, cho phép các nhà phát triển dễ dàng tích hợp và sử dụng các thư viện và công cụ khác nhau trong ứng dụng của họ.
* Đa nền tảng: Node.js tương thích với nhiều hệ điều hành khác nhau, bao gồm Windows, macOS và Linux, khiến nó trở thành một lựa chọn linh hoạt cho các nhà phát triển làm việc trên các nền tảng khác nhau.

Node.js thường được sử dụng để xây dựng nhiều loại ứng dụng khác nhau, bao gồm máy chủ web, ứng dụng web thời gian thực, công cụ mạng và ứng dụng dòng lệnh. Nó đã trở nên phổ biến đáng kể trong cộng đồng phát triển nhờ tính hiệu quả, khả năng mở rộng và hệ sinh thái sôi động do trình quản lý gói NPM cung cấp.

## **7. Responsive Web Design**

### **7.1.Giới thiệu về cách tạo trang web**

Responsive web design (RWD) là một phương pháp thiết kế trang web nhằm tạo ra một trải nghiệm người dùng tốt nhất trên nhiều thiết bị và kích thước màn hình khác nhau, bao gồm máy tính để bàn, máy tính xách tay, tablet và điện thoại di động.

Cấu trúc và đặc điểm của một trang web phản hồi bao gồm:

1. Layout mạch linh hoạt: Sử dụng các đơn vị đo không phụ thuộc vào pixel, như phần trăm (%) hoặc đơn vị EM, REM, hoạc VH, VW để xây dựng giao diện. Điều này cho phép giao diện thay đổi kích thước và diện mạo dựa trên kích thước màn hình hiển thị.

2. Media queries: Sử dụng media queries, một tính nănng CSS, để áp dụng các quy tắc CSS khác nhau dựa trên kích thước màn hình và đặc điểm thiết bị. Điều này cho phép điều chỉnh kích thước, vị trí, font chữ và bố cục của các phần tử trên trang.

3. Ẩn/hiện phần tử: Sử dụng CSS để ẩn/hiện các phần tử không phù hợp với kích thước màn hình nhất định. Bằng cách này, bạn có thể cung cấp một trải nghiệm người dùng thông qua giao diện gọn gàng và dễ đọc hơn.

4. Ảnh phản hồi: Sử dụng các phương pháp như thu nhỏ, tạo phiên bản nhạy cảm và tải hình ảnh dựa trên kích thước màn hình của thiết bị. Dùng thuộc tính CSS "max-width: 100%" để đảm bảo các hình ảnh không tràn ra khỏi kích thước khung hiển thị.

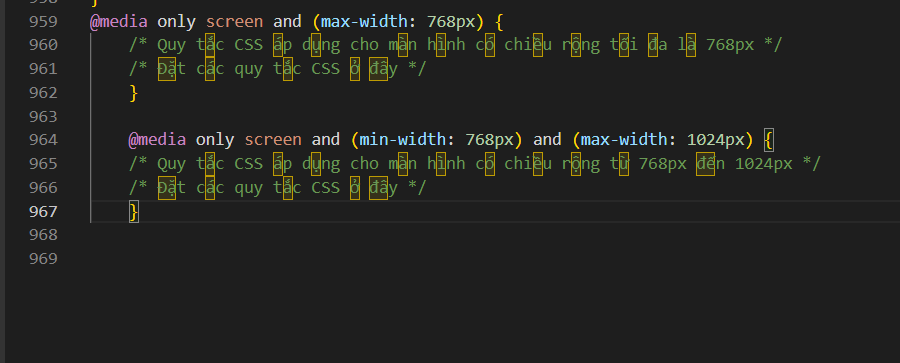
5. Điều hướng và tương tác: Thiết kế điều hướng và tương tác tự nhiên và dễ sử dụng trên mọi thiết bị. Sử dụng các cấu trúc như menu rời và nút "hamburger" để tìm kiếm và điều hướng dễ dàng trên màn hình nhỏ.

Với responsive web design, một trang web có thể thay đổi cấu trúc và hiển thị tùy thuộc vào kích thước màn hình và thiết bị của người dùng. Điều này mang lại một trải nghiệm tốt hơn cho người dùng và đảm bảo rằng trang web của bạn hiển thị đẹp và dễ sử dụng trên mọi thiết bị di động.

### **7.2. Sử dụng các kỹ thuật CSS media queries để điều chỉnh giao diện cho các màn hình khác nhau.**

Media queries là một kỹ thuật quan trọng trong responsive web design để điều chỉnh giao diện cho các màn hình khác nhau. Khi sử dụng media queries, bạn có thể áp dụng các quy tắc CSS khác nhau dựa trên các điều kiện như kích thước màn hình, tỷ lệ khung hình, hoặc các thiết bị đặc biệt.

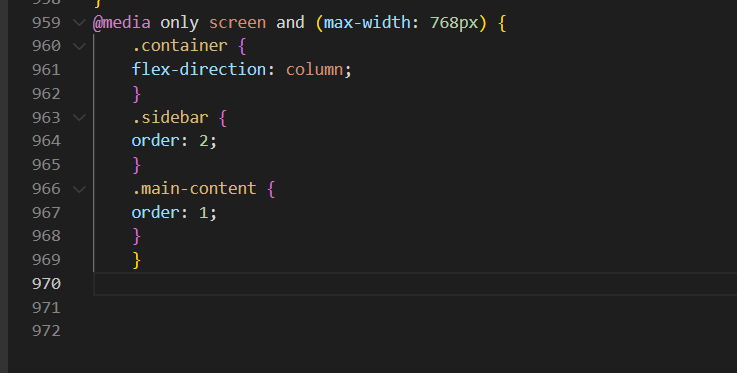
Cú pháp sử dụng media queries trong CSS như sau:



/\* thêm nhiều media queries khác nếu cần \*/

Dựa vào các query media, bạn có thể định nghĩa các quy tắc CSS khác nhau để thay đổi kích thước, vị trí, phông chữ, bố cục, màu sắc và các thuộc tính khác của các phần tử trên trang web.

Ví dụ, bạn có thể áp dụng các quy tắc CSS sau để điều chỉnh bố cục khi chiều rộng của màn hình nhỏ hơn 768px:

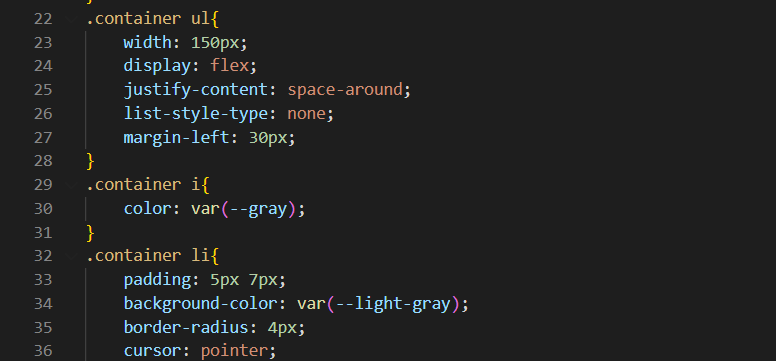


### **7.3 Hướng dẫn cách tạo bố cục đáng linh hoạt cho các thiết bị khác nhau**

Để tạo bố cục đáng linh hoạt cho các thiết bị khác nhau, bạn có thể sử dụng các kỹ thuật và cấu trúc HTML/CSS sau đây:

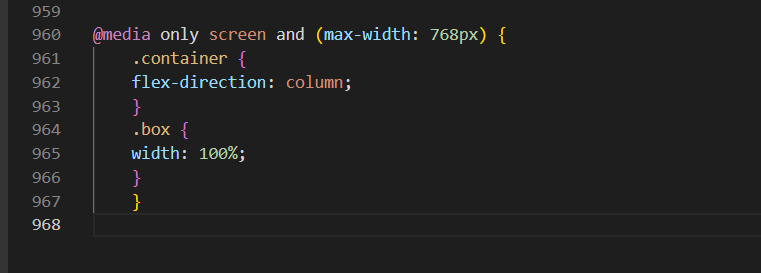
1. Sử dụng đơn vị đo không phụ thuộc vào pixel: Thay vì sử dụng pixel để định kích thước và vị trí của các phần tử, hãy sử dụng phần trăm (%) hoặc đơn vị EM, REM, VH (viewport height) và VW (viewport width). Điều này cho phép các phần tử thay đổi kích thước phù hợp với kích thước màn hình.

Ví dụ:



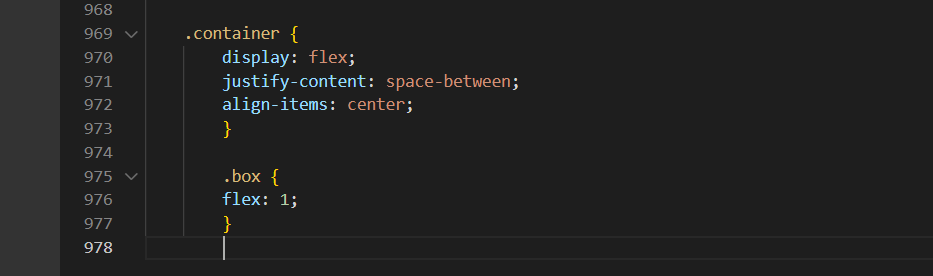
2. Sử dụng Media Queries: Sử dụng media queries để áp dụng các quy tắc CSS khác nhau dựa trên kích thước màn hình. Bằng cách này, bạn có thể điều chỉnh bố cục và kiểu dáng của các phần tử trên các thiết bị khác nhau.

Ví dụ:



3. Sử dụng CSS Flexbox: Flexbox là một công cụ mạnh mẽ cho việc xây dựng bố cục đáng linh hoạt. Bạn có thể sử dụng các thuộc tính như `display: flex`, `flex-direction`, `justify-content`, và `align-items` để điều chỉnh và sắp xếp các phần tử trên trang web.

Ví dụ:



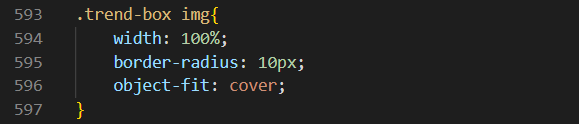
4. Sử dụng CSS Grid: Grid cung cấp khả năng chia lưới linh hoạt và tiện ích hơn cho việc tạo bố cục đáng linh hoạt. Sử dụng các thuộc tính như `display: grid`, `grid-template-columns`, và `grid-template-rows` để tạo các ô và hàng cho các phần tử trong bố cục.

Ví dụ:



5. Sử dụng max-width cho hình ảnh và phần tử khác: Để đảm bảo rằng hình ảnh và các phần tử không tràn ra khỏi kích thước màn hình, sử dụng thuộc tính `max-width: 100%` cho các hình ảnh và phần tử khác.

Ví dụ:

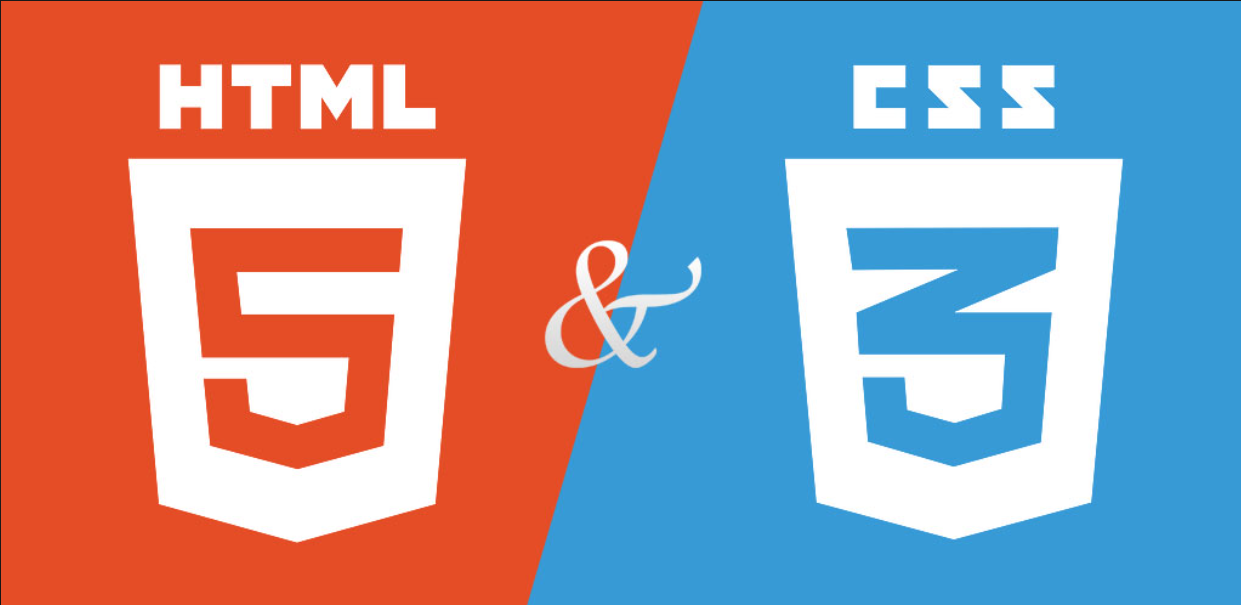


Bằng cách sử dụng các kỹ thuật và cấu trúc này, bạn có thể tạo bố cục đáng linh hoạt để trang web của bạn tự động thích ứng với các thiết bị khác nhau. Điều này sẽ mang lại trải nghiệm người dùng tốt hơn và cho phép trang web của bạn được hiển thị đẹp trên mọi thiết bị.

## **8. HTML5 và CSS3**

### **8.1.Giới thiệu về các tính năng mới trong HTML5 và CSS3**

HTML5 và CSS3 là hai phiên bản mới nhất của ngôn ngữ đánh dấu và ngôn ngữ định dạng trang web.



**HTML5:**

* Semantic Elements: HTML5 giới thiệu nhiều phần tử ngữ nghĩa mới như `<header>`, `<nav>`, `<article>`, `<section>`, `<aside>`, `<footer>`,... Giúp việc chia nhỏ và xác định cấu trúc của trang web trở nên dễ dàng hơn và tăng cường tính tương thích với các công cụ tìm kiếm và trình đọc màn hình.
* Multimedia support: HTML5 cung cấp các phần tử như `<video>` và `<audio>` để nhúng nội dung multimedia trực tiếp vào trang web, loại bỏ nhu cầu sử dụng plugin bên ngoài như Flash để phát video hoặc audio.
* Form enhancements: HTML5 đưa ra nhiều tính năng mới cho các biểu mẫu như autocomplete, date picker, email validation, slider, và pattern matching, giúp tăng cường khả năng xử lý và kiểm soát dữ liệu người dùng nhập vào.
* Canvas and SVG: HTML5 cung cấp cả phần tử `<canvas>` và hỗ trợ SVG (Scalable Vector Graphics), cho phép tạo đồ họa và hoạt ảnh phức tạp trên trang web mà không cần sử dụng plugin bên ngoài.
* CSS3:
* Flexible Box Layout (Flexbox): Flexbox giúp dễ dàng tạo bố cục linh hoạt và điều chỉnh vị trí của các phần tử trong một container, cho phép tạo các mẫu layout phức tạp một cách linh hoạt hơn.
* Grid Layout: CSS3 Grid Layout là một công cụ mạnh mẽ để tạo các lưới nhiều hàng và cột cho trang web. Nó cho phép xác định kích thước và vị trí của các phần tử trên trang.
* Transitions và Animations: CSS3 cung cấp tính năng transitions và animations, cho phép tạo hiệu ứng chuyển động mượt mà và tạo các hoạt ảnh trên trang mà không cần sử dụng JavaScript hoặc Flash.
* Nhiều thuộc tính mới: CSS3 cung cấp các thuộc tính mới như box-shadow, border-radius, gradients, transforms, và nth-child selector. Các thuộc tính này giúp tạo hiệu ứng đẹp mắt và tăng khả năng kiểm soát giao diện trang web.

Sự kết hợp giữa HTML5 và CSS3 mang lại nhiều cải tiến đáng kể trong phát triển trang web, từ cấu trúc, giao diện người dùng, đồ họa, đến hiệu ứng tự động và tương tác. Việc sử dụng những tính năng mới này giúp tạo ra trải nghiệm người dùng dễ dàng hơn và nâng cao khả năng tương tác trang web.

### **8.2 Hướng dẫn cách sử dụng các phần tử HTML5 như video, âm thanh, canvas, và đồ họa vector.**



Để sử dụng các phần tử HTML5 như video, âm thanh, canvas và đồ họa vector, bạn có thể làm theo các bước sau:

* Phần tử video: - Sử dụng thẻ <video> để nhúng video vào trang web của bạn. - Định rõ nguồn video bằng thuộc tính src. - Bạn có thể thêm các thuộc tính bổ sung như controls để hiển thị các nút điều khiển video, width và height để định kích thước video. Ví dụ:

<video src="video.mp4" controls width="400" height="300"></video>

* Phần tử âm thanh: - Sử dụng thẻ <audio> để nhúng âm thanh vào trang web. - Định rõ nguồn âm thanh bằng thuộc tính src. - Tương tự như video, bạn có thể thêm thuộc tính controls để hiển thị các nút điều khiển âm thanh. Ví dụ:

<audio src="audio.mp3" controls></audio>

* Phần tử canvas: - Sử dụng thẻ <canvas> để tạo một vùng vẽ trên trang web. - Định rõ kích thước của canvas bằng thuộc tính width và height. - Sử dụng JavaScript để vẽ hoặc thao tác với các hình vẽ trên canvas. Ví dụ:

<canvas id="myCanvas" width="500" height="300"></canvas>

* Đồ họa vector: - Sử dụng thẻ <svg> để tạo đồ họa vector trên trang web. - Sử dụng các phần tử SVG như <rect>, <circle>, <path> để tạo các hình vẽ. - Định rõ thuộc tính như x, y, width, height, cx, cy, r, d để chỉnh sửa hình dạng và vị trí của các phần tử. Ví dụ:

<svg width="400" height="300">

<rect x="50" y="50" width="200" height="100" fill="blue" />

<circle cx="300" cy="150" r="50" fill="red" />

<path d="M100,200 L200,100 L300,200 Z" fill="green" />

</svg>

### **8.3 Sử dụng CSS3 để thêm các hiệu ứng và chuyển động vào trang web, bao gồm cả CSS transitions và animations**



Để sử dụng CSS3 để thêm hiệu ứng và chuyển động vào trang web, bạn có thể sử dụng CSS transitions và animations. 1. CSS Transitions: CSS transitions cho phép bạn tạo ra hiệu ứng chuyển đổi mượt mà khi một thuộc tính của phần tử thay đổi. Để áp dụng một transition, bạn cần định nghĩa thuộc tính transition cho phần tử và xác định thuộc tính thay đổi cùng với thời gian và kiểu chuyển đổi. Ví dụ:

.element { transition: property duration timing-function delay; }

Trong đó:

- property: là thuộc tính của phần tử (ví dụ: background, color, opacity, etc.)

- duration: thời gian diễn ra transition (ví dụ: 0.5s, 1s, etc.)

- timing-function: kiểu chuyển đổi (ví dụ: ease, linear, ease-in-out, etc.)

- delay: thời gian trễ trước khi transition bắt đầu (ví dụ: 0s, 0.5s, etc.) 2. CSS Animations: CSS animations cho phép bạn tạo ra các chuyển động phức tạp và điều khiển chúng bằng cách xác định các keyframes. Bạn cần định nghĩa animation bằng cách tạo ra một tên cho animation, đặt thuộc tính animation trên phần tử và xác định các keyframes. Ví dụ:

@keyframes animation-name {

0% { property: value; }

50% { property: value; }

100% { property: value; }

}

.element {

animation: animation-name duration timing-function delay iteration-count direction;

}

Trong đó:

- animation-name: là tên của animation mà bạn đã định nghĩa

- duration: thời gian diễn ra animation (ví dụ: 0.5s, 1s, etc.)

- timing-function: kiểu chuyển đổi (ví dụ: ease, linear, ease-in-out, etc.)

- delay: thời gian trễ trước khi animation bắt đầu (ví dụ: 0s, 0.5s, etc.)

- iteration-count: số lần lặp lại animation (ví dụ: infinite, 3, etc.)

- direction: hướng chuyển đổi của animation (ví dụ: normal, alternate, etc.)

Bằng cách sử dụng CSS transitions và animations, bạn có thể thêm các hiệu ứng và chuyển động thú vị vào các phần tử trên trang web của bạn. Hãy thử tạo ra các transition và animation mà bạn muốn và tuỳ chỉnh thuộc tính và giá trị của chúng để tạo ra hiệu ứng độc đáo.

## **9. Thiết kế trang web thân thiện với SEO**

### **9.1.Giới thiệu về SEO**

SEO là viết tắt của Search Engine Optimization, một chiến lược tối ưu hóa để cải thiện vị trí của một trang web trên các kết quả tìm kiếm tự nhiên (không phải quảng cáo) trên các công cụ tìm kiếm như Google, Bing, Yahoo, và nhiều hệ thống tìm kiếm khác.



Mục tiêu chính của SEO là tăng lượng truy cập trang web và tăng cường sự hiện diện trực tuyến bằng cách đạt được vị trí cao trên trang kết quả tìm kiếm cho các từ khóa cụ thể hoặc các cụm từ liên quan đến nội dung của trang web đó.

SEO đóng vai trò quan trọng đối với một trang web vì nó mang lại nhiều lợi ích quan trọng như sau:

* Tăng lượng truy cập: Tối ưu hóa SEO giúp tăng cơ hội trang web xuất hiện trên các kết quả tìm kiếm hàng đầu, từ đó tăng lượng truy cập của người dùng.
* Tăng uy tín và tăng doanh số: Vị trí cao trên trang kết quả tìm kiếm cũng tạo niềm tin đối với người dùng, giúp tăng cơ hội chuyển đổi từ lượng truy cập thành doanh số.
* Tiết kiệm chi phí quảng cáo: Với việc xuất hiện tự nhiên trên các kết quả tìm kiếm, trang web không cần phải chi tiêu lớn cho quảng cáo trực tuyến.
* Tăng tính cạnh tranh: Trong môi trường kinh doanh trực tuyến cạnh tranh, việc thực hiện SEO đúng đắn giúp trang web của bạn tiếp cận được đến nhiều khách hàng tiềm năng hơn, từ đó tạo ra lợi thế cạnh tranh.
* Tối ưu hóa trải nghiệm người dùng: Việc tối ưu hóa trang web để tăng vị trí trên các công cụ tìm kiếm thường đi kèm với việc cải thiện trải nghiệm người dùng, từ tốc độ tải trang đến cấu trúc trang web và nội dung hấp dẫn.
* Xác định khách hàng tiềm năng: SEO giúp tập trung vào việc tìm kiếm từ khóa mục tiêu mà khách hàng tiềm năng của bạn thường tìm kiếm trên internet, từ đó giúp bạn định hình chiến lược tiếp thị mục tiêu một cách hiệu quả hơn.
* Với mục tiêu tăng cường sự hiện diện trực tuyến và tối ưu hóa hiệu suất trang web, SEO trở thành một phần quan trọng của chiến lược tiếp thị kỹ thuật số cho các doanh nghiệp và tổ chức.

### **9.2.Hướng dẫn cách tối ưu hóa tiêu đề trang, thẻ meta, và nội dung để cải thiện thứ hạng tìm kiếm**

Tối ưu hóa SEO (Search Engine Optimization) là một trong những yếu tố quan trọng để cải thiện thứ hạng trang web của bạn trên các công cụ tìm kiếm. Đây là một số hướng dẫn cơ bản để tối ưu hóa tiêu đề trang, thẻ meta, và nội dung:

Tiêu đề Trang (Page Title):

* Hãy chắc chắn rằng tiêu đề trang chứa từ khóa chính mà bạn muốn trang web của mình được tìm kiếm.
* Sử dụng từ khóa ở phần đầu của tiêu đề để tăng tính quan trọng của nó.
* Giữ độ dài tiêu đề dưới 60 ký tự để đảm bảo nó không bị cắt ngắn trên kết quả tìm kiếm.

Thẻ Meta Description:

* Tạo mô tả hấp dẫn và liên quan, khuyến khích người dùng nhấp chuột vào trang web của bạn.
* Sử dụng từ khóa trong mô tả meta, đặc biệt là ở phần đầu.
* Giữ độ dài mô tả dưới 160 ký tự để đảm bảo nó không bị cắt ngắn trên kết quả tìm kiếm.

Nội Dung:

* Tạo nội dung chất lượng và độc đáo mà người dùng sẽ thấy hữu ích và chia sẻ.
* Tối ưu hóa từ khóa mục tiêu trong nội dung, bao gồm cả tiêu đề, đoạn văn, và thẻ hình ảnh.
* Sử dụng tiêu đề phù hợp, các đoạn văn ngắn, hình ảnh, video, và định dạng văn bản hợp lý để tăng tính hấp dẫn và dễ đọc của nội dung.

URLs thân thiện với SEO:

* Sử dụng URLs chứa từ khóa mục tiêu của bạn.
* Tránh sử dụng các ký tự đặc biệt và số ID không cần thiết trong URL của bạn.

Thẻ Heading (H1, H2, H3, v.v.):

* Sử dụng thẻ heading một cách hợp lý để phân đoạn nội dung và làm nổi bật các từ khóa chính.
* Sử dụng thẻ H1 cho tiêu đề chính của trang và sử dụng thẻ H2 và H3 để đánh dấu các phần con quan trọng.

Liên kết nội bộ:

* Xây dựng các liên kết nội bộ giữa các trang trong trang web của bạn để cải thiện sự điều hướng và tăng cường sự liên kết giữa các trang.

Tốc độ tải trang:

* Tối ưu hóa tốc độ tải trang để cải thiện trải nghiệm người dùng và thứ hạng tìm kiếm.
* Nén hình ảnh, sử dụng bộ nhớ đệm và một máy chủ nhanh để tăng tốc độ tải trang.

Phản hồi di động:

* Đảm bảo trang web của bạn được tối ưu hóa cho trải nghiệm di động để đáp ứng yêu cầu của Google và cải thiện thứ hạng tìm kiếm trên các thiết bị di động.

Nhớ rằng, SEO là một quá trình liên tục và đòi hỏi thời gian để thấy kết quả. Bạn nên liên tục theo dõi, đánh giá và điều chỉnh chiến lược SEO của mình để đảm bảo rằng bạn đang đạt được kết quả tốt nhất có thể.

### **9.3. Sử dụng công cụ tối ưu hóa SEO để theo dõi hiệu suất và tối ưu hóa trang web.**



Tối ưu hóa SEO (Search Engine Optimization) là một phần quan trọng của việc xây dựng và duy trì một trang web. Để theo dõi hiệu suất và tối ưu hóa trang web của bạn, có nhiều công cụ tối ưu hóa SEO mạnh mẽ có thể giúp bạn đạt được mục tiêu này. Dưới đây là một số công cụ tối ưu hóa SEO phổ biến mà bạn có thể sử dụng:

* Google Analytics: Đây là một công cụ phân tích web mạnh mẽ cung cấp thông tin chi tiết về lưu lượng truy cập trang web, nguồn lưu lượng, bài viết phổ biến và nhiều thông tin khác. Nó cũng cung cấp thông tin về từ khóa mà người dùng sử dụng để tìm kiếm trang web của bạn.
* Google Search Console: Cung cấp thông tin chi tiết về cách Google xem trang web của bạn và cung cấp thông tin về tình trạng lập chỉ mục, lỗi trang, truy cập từ khóa, và nhiều thông tin khác. Nó giúp bạn hiểu rõ hơn về cách trang web của bạn hiển thị trên kết quả tìm kiếm Google.
* SEMrush: Là một công cụ phổ biến cung cấp thông tin chi tiết về từ khóa, cạnh tranh từ khóa, kiểm tra backlink, và nhiều yếu tố tối ưu hóa trang web khác. Nó cũng cung cấp báo cáo đầy đủ về hiệu suất SEO và gợi ý cải thiện.
* Ahrefs: Cung cấp thông tin chi tiết về từ khóa, cạnh tranh từ khóa, backlink, và phân tích đối thủ mạnh mẽ. Nó cung cấp các công cụ giúp bạn tối ưu hóa trang web của mình theo cách hiệu quả và đạt được thứ hạng cao hơn trên kết quả tìm kiếm.
* Moz Pro: Cung cấp các công cụ tối ưu hóa mạnh mẽ bao gồm kiểm tra từ khóa, phân tích liên kết, phân tích cạnh tranh, và nhiều tính năng khác giúp bạn tối ưu hóa trang web và theo dõi hiệu suất của nó.

Khi sử dụng các công cụ này, hãy đảm bảo bạn hiểu rõ về cách sử dụng chúng và phân tích kỹ lưỡng thông tin mà chúng cung cấp. Bằng cách kết hợp các công cụ này với chiến lược tối ưu hóa SEO đúng đắn, bạn có thể tối ưu hóa trang web của mình để đạt được hiệu suất tốt nhất trên các kết quả tìm kiếm.

## **10. Quản lý phiên bản và Git**

### **10.1. Giới thiệu về hệ thống quản lý mã nguồn và Git**

Hệ thống quản lý mã nguồn (Source Control Management System - SCM) là một phần quan trọng của quá trình phát triển phần mềm. Nó giúp quản lý sự thay đổi của mã nguồn trong quá trình phát triển phần mềm, cho phép các nhà phát triển làm việc cùng nhau trên cùng một dự án mà không xảy ra xung đột, cũng như theo dõi lịch sử của mã nguồn. Git là một trong những hệ thống quản lý mã nguồn phổ biến nhất hiện nay.



Git là một hệ thống quản lý phiên bản phân tán được tạo ra bởi Linus Torvalds vào năm 2005. Nó được thiết kế để quản lý mã nguồn một cách hiệu quả, linh hoạt và an toàn. Một số đặc điểm quan trọng của Git bao gồm:

* Phiên bản phân tán: Mỗi người dùng có một bản sao đầy đủ của toàn bộ kho lưu trữ. Điều này giúp giảm thiểu rủi ro mất dữ liệu do mất kết nối với máy chủ trung tâm.
* Hiệu suất cao: Git thực hiện các hoạt động cục bộ mà không cần kết nối với máy chủ, do đó giúp tăng tốc độ xử lý.
* Nhánh linh hoạt: Git cho phép tạo và quản lý nhiều nhánh một cách dễ dàng, giúp phát triển song song và thử nghiệm các tính năng mới mà không làm ảnh hưởng đến nhánh chính.
* Khả năng theo dõi lịch sử: Git ghi lại toàn bộ lịch sử thay đổi của mã nguồn, cho phép người dùng theo dõi, đánh giá và phục hồi lại bất kỳ phiên bản nào từ quá khứ.
* Hỗ trợ cho nhiều dịch vụ lưu trữ: Git hỗ trợ nhiều dịch vụ lưu trữ trực tuyến như GitHub, GitLab và Bitbucket, cung cấp cho người dùng nhiều tùy chọn để lưu trữ mã nguồn của họ và tương tác với các nhà phát triển khác.
* Git cung cấp một loạt các lệnh dòng lệnh mạnh mẽ để quản lý kho lưu trữ, nhánh, merge, và theo dõi lịch sử thay đổi. Nó cũng có một giao diện đồ họa, điển hình như GitHub, giúp người dùng quản lý kho lưu trữ một cách trực quan và thuận tiện. Hơn nữa, Git đã trở thành một tiêu chuẩn công nghiệp và được sử dụng rộng rãi trong cộng đồng phát triển phần mềm.

### **10.2. Hướng dẫn tạo kho lưu trữ (repository) Git và làm việc với các lệnh cơ bản như git clone, git pull, git push, và git commit**

Để bắt đầu làm việc với Git, bạn cần cài đặt Git trên máy tính của mình trước. Sau đó, bạn có thể tạo một kho lưu trữ (repository) trên máy tính hoặc trên một dịch vụ lưu trữ trực tuyến như GitHub, GitLab, hoặc Bitbucket. Dưới đây là hướng dẫn cơ bản cho việc tạo kho lưu trữ và làm việc với các lệnh cơ bản trong Git.

**Bước 1: Cài đặt Git**

Truy cập trang chủ Git tại git-scm.com và tải xuống phiên bản phù hợp với hệ điều hành của bạn.

Tiến hành cài đặt Git trên máy tính của bạn.

**Bước 2: Tạo kho lưu trữ (repository) trên máy tính hoặc trên dịch vụ lưu trữ trực tuyến**

Tạo kho lưu trữ trên máy tính của bạn:

Mở Terminal hoặc Command Prompt trên máy tính của bạn.

Di chuyển đến thư mục mà bạn muốn tạo kho lưu trữ. Sử dụng lệnh cd <đường\_dẫn\_đến\_thư\_mục>.

Gõ lệnh sau để tạo một kho lưu trữ mới:



Tạo kho lưu trữ trên dịch vụ lưu trữ trực tuyến:

Đăng nhập vào tài khoản của bạn trên GitHub, GitLab, hoặc Bitbucket.

Tạo một kho lưu trữ mới từ giao diện người dùng.

**Bước 3: Làm việc với các lệnh cơ bản**

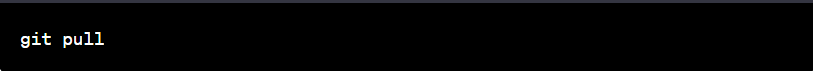
Git Clone:

Để sao chép một kho lưu trữ từ một URL cụ thể, sử dụng lệnh sau:

****

Git Pull:

Để tải xuống và hợp nhất các thay đổi từ kho lưu trữ từ xa, sử dụng lệnh sau:

****

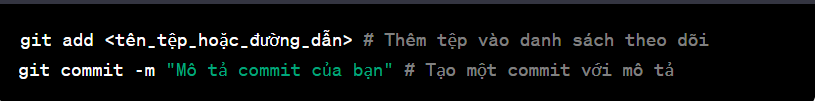
Git Push:

Để đẩy các thay đổi từ kho lưu trữ cục bộ của bạn lên kho lưu trữ từ xa, sử dụng lệnh sau:

****

Git Commit:

Để tạo một lần commit mới, sử dụng lệnh sau:



Hãy nhớ thay thế <tên\_tệp\_hoặc\_đường\_dẫn> bằng tên tệp hoặc đường dẫn thích hợp.

Với hướng dẫn trên, bạn có thể bắt đầu làm việc với Git và làm quen với các lệnh cơ bản như git clone, git pull, git push và git commit.

### **10.3. Thảo luận về việc làm việc cùng nhóm trong môi trường Git và quản lý phiên bản dự án**

Làm việc cùng nhóm trong môi trường Git và quản lý phiên bản dự án là một phần quan trọng trong quá trình phát triển phần mềm. Điều này đảm bảo rằng tất cả các thành viên trong nhóm có thể làm việc đồng thời trên cùng một dự án mà không gây xung đột và mất mát dữ liệu. Dưới đây là một số điểm cần xem xét khi làm việc cùng nhóm trong môi trường Git và quản lý phiên bản dự án:

* Branching strategy: Quyết định chiến lược nhánh là cần thiết để phân chia công việc và tránh xung đột giữa các thay đổi. Có thể sử dụng các chiến lược nhánh như GitFlow, Feature Branch Workflow hoặc Trunk-Based Development tùy thuộc vào quy mô và yêu cầu cụ thể của dự án.
* Regular updates and pull requests: Đảm bảo rằng tất cả các thành viên trong nhóm thường xuyên cập nhật mã nguồn của mình từ nhánh chính (trunk) hoặc nhánh cha để tránh xung đột khi hợp nhất (merge) code. Pull requests nên được sử dụng để kiểm tra và xác nhận tính ổn định của mã nguồn trước khi hợp nhất vào nhánh chính.
* Commit standards: Thiết lập tiêu chuẩn về cách thức commit và message để giúp cho việc theo dõi và đánh giá lịch sử thay đổi dễ dàng hơn. Mỗi commit nên tập trung vào một nhiệm vụ cụ thể và có message rõ ràng để mọi người trong nhóm có thể dễ dàng hiểu.
* Code review process: Xác định quá trình xem xét mã nguồn của các thành viên trong nhóm. Quá trình này giúp cải thiện chất lượng mã nguồn và đảm bảo rằng mã nguồn đáp ứng các tiêu chuẩn và yêu cầu đã đặt ra trước khi được hợp nhất vào nhánh chính.
* Issue tracking and management: Sử dụng các công cụ quản lý dự án để theo dõi và quản lý vấn đề, yêu cầu và nhiệm vụ. Điều này giúp cho việc phân công công việc, theo dõi tiến độ và quản lý ưu tiên trong dự án.
* Backup và restore: Đảm bảo rằng dữ liệu của dự án được sao lưu định kỳ và có khả năng khôi phục khi cần thiết để tránh mất mát dữ liệu không mong muốn.
* Documentation: Đảm bảo rằng tất cả các thay đổi và quyết định trong quá trình phát triển đều được ghi lại đầy đủ. Tài liệu về cấu trúc dự án, tiêu chuẩn mã hóa, quy trình kiểm tra, và các quyết định quan trọng khác cần được ghi lại một cách rõ ràng để tất cả thành viên trong nhóm đều có thể hiểu và tham khảo.

Những điểm trên chỉ là một số gợi ý cơ bản và không đầy đủ về cách quản lý phiên bản dự án trong môi trường Git. Tùy thuộc vào quy mô và yêu cầu cụ thể của dự án, có thể cần phải điều chỉnh hoặc mở rộng thêm các phương pháp và tiêu chuẩn quản lý khác.