## Mô hình hoạt động của Website:

Khi người dùng muốn xem một trang web, họ sẽ nhập địa chỉ website (URL) vào trình duyệt. Lúc này, website chưa trực tiếp hoạt động. Trình duyệt sẽ gửi yêu cầu tới DNS Server, đóng vai trò như một người "phiên dịch" địa chỉ web sang địa chỉ IP, để xác định chính xác vị trí máy chủ của website trên internet. Sau khi có địa chỉ IP, trình duyệt sẻ kết nối tới máy chủ web (Web Server), nơi website "cư trú" và lưu trữ tất cả dữ liệu. Máy chủ web này sẽ tìm và gửi các file nội dung website (HTML, CSS, JavaScript, hình ảnh,...) về cho trình duyệt. Cuối cùng, trình duyệt sẽ nhận các file này, xử lý để tạo giao diện trang web và hiển thị cho người dùng, tạo nên trải nghiệm website mà chúng ta thấy hàng ngày. Như vậy, website đóng vai trò là một hệ thống, nơi kết nối các thành phần để cung cấp nội dung và dịch vụ cho người dùng.

## Cách hoạt động của Browser:

Khi người dùng muốn xem một trang web, họ nhập địa chỉ (URL) hoặc nhấp vào một liên kết trên trình duyệt. Lúc này, browser đóng vai trò là "cánh cửa" để người dùng tiếp cận web. Trình duyệt sẽ gửi một yêu cầu HTTP (HTTP request) đến máy chủ web, như một người "gọi món" thông báo rằng nó muốn xem trang web đó. Máy chủ web sẽ phản hồi bằng một phản hồi HTTP (HTTP response), thường chứa file HTML, như "thực đơn" của trang web. Trình duyệt bắt đầu phân tích (parsing) file HTML để hiểu cấu trúc trang web, đóng vai trò "người đọc hiểu" ngôn ngữ web. Nếu có các liên kết đến tài nguyên khác như file CSS, JavaScript, hình ảnh, video, trình duyệt sẽ gửi các yêu cầu HTTP riêng cho từng tài nguyên đó, như "yêu cầu thêm" các nguyên liệu cho món ăn. Máy chủ sẽ gửi trả lại các tài nguyên này. Cuối cùng, trình duyệt kết hợp tất cả các tài nguyên nhận được, như "đầu bếp" tạo nên món ăn hoàn chỉnh, để xây dựng và hiển thị trang web hoàn chỉnh trên màn hình của người dùng.

## Cơ chế hoạt động của DNS (Domain Name System)

Khi người dùng nhập địa chỉ website (URL) vào trình duyệt, bước đầu tiên và quan trọng là trình duyệt cần biết địa chỉ IP (Internet Protocol) của máy chủ web tương ứng. Để thực hiện điều này, trình duyệt sẽ gửi một yêu cầu đến DNS server. DNS server, đóng vai trò như một "danh bạ" của internet, có chức năng tìm kiếm và trả về địa chỉ IP của máy chủ web dựa trên địa chỉ URL mà trình duyệt cung cấp. Sau khi có được địa chỉ IP, trình duyệt sẽ thiết lập một kết nối TCP đến máy chủ web để bắt đầu quá trình truyền tải dữ liệu, tạo "đường đi" cho dữ liệu. Tiếp theo, trình duyệt sẽ gửi các yêu cầu HTTP đến máy chủ, yêu cầu các tài nguyên cần thiết để hiển thị trang web. Máy chủ web sẽ phản hồi bằng các phản hồi HTTP, chứa nội dung của trang web. Như vậy, DNS đóng vai trò như một bước khởi đầu cần thiết, một "người định vị" quan trọng, giúp trình duyệt có thể tìm đến đúng địa chỉ máy chủ web mà người dùng muốn truy cập.

## Các khái niệm

**Các khái niệm cơ bản về Internet và Web:**

1. **Internet:** Mạng lưới toàn cầu kết nối các máy tính và thiết bị trên khắp thế giới, cho phép chúng trao đổi dữ liệu với nhau.
2. **World Wide Web (WWW) hay Web:** Hệ thống thông tin toàn cầu được xây dựng trên nền tảng internet, cho phép truy cập các trang web, hình ảnh, video, và các loại tài liệu khác thông qua trình duyệt.
3. **Website:** Một tập hợp các trang web (webpage) được liên kết với nhau và có chung một tên miền, được lưu trữ trên máy chủ web và cung cấp thông tin, dịch vụ cho người dùng.
4. **Webpage:** Một trang đơn lẻ trong một website, được hiển thị trên trình duyệt web.
5. **URL (Uniform Resource Locator):** Địa chỉ duy nhất của một trang web hoặc tài nguyên trên internet (ví dụ: <https://www.example.com/page1>).
6. **Domain name (Tên miền):** Tên dễ nhớ của một website trên internet, thay thế cho địa chỉ IP dạng số (ví dụ: google.com, youtube.com).
7. **IP address (Địa chỉ IP):** Địa chỉ số duy nhất của một máy tính hoặc thiết bị kết nối vào mạng internet (ví dụ: 192.168.1.1).
8. **DNS Server (Máy chủ DNS):** Máy chủ chịu trách nhiệm chuyển đổi tên miền (domain name) thành địa chỉ IP để trình duyệt có thể tìm đến máy chủ web.

**Các khái niệm về Browser (Trình duyệt):**

1. **Web browser:** Phần mềm dùng để xem các trang web (ví dụ: Chrome, Firefox, Safari, Edge).
2. **HTTP (Hypertext Transfer Protocol):** Giao thức ứng dụng dùng để truyền tải dữ liệu trên web, bao gồm các yêu cầu (request) và phản hồi (response).
3. **HTTP request:** Yêu cầu mà trình duyệt gửi đến máy chủ web.
4. **HTTP response:** Phản hồi của máy chủ web trả lại cho trình duyệt.
5. **HTML (Hypertext Markup Language):** Ngôn ngữ đánh dấu dùng để tạo cấu trúc của trang web.
6. **CSS (Cascading Style Sheets):** Ngôn ngữ định kiểu, dùng để tạo kiểu dáng và màu sắc cho trang web.
7. **JavaScript:** Ngôn ngữ lập trình dùng để thêm chức năng và tương tác cho trang web.

**Các khái niệm về Web Server (Máy chủ web):**

1. **Web server:** Máy tính hoặc phần mềm lưu trữ website và đáp ứng các yêu cầu từ trình duyệt.
2. **Website content:** Nội dung của website bao gồm HTML, CSS, JavaScript, hình ảnh, video và các file khác.
3. **Web application:** Ứng dụng chạy trên máy chủ web, thường xử lý các tương tác phức tạp của người dùng (ví dụ: đăng nhập, thanh toán,...).
4. **Database (Cơ sở dữ liệu):** Nơi lưu trữ dữ liệu của website (ví dụ: thông tin người dùng, bài viết, sản phẩm), thường sử dụng MySQL, PostgreSQL,...
5. **Operating System:** Hệ điều hành chạy trên cả máy tính cá nhân và máy chủ web (ví dụ: Windows, Linux, macOS).

**Các khái niệm về Hosting và Domain:**

1. **Hosting:** Dịch vụ lưu trữ web, cung cấp không gian lưu trữ và các tài nguyên máy chủ để website hoạt động trên internet.
2. **Shared hosting:** Loại hosting mà nhiều website cùng chia sẻ một máy chủ.
3. **VPS (Virtual Private Server):** Máy chủ ảo riêng, cung cấp tài nguyên riêng biệt hơn shared hosting.
4. **Dedicated server:** Máy chủ vật lý riêng, cung cấp toàn bộ tài nguyên cho một website.
5. **Disk space (Dung lượng lưu trữ):** Lượng không gian lưu trữ trên máy chủ dành cho website.
6. **Bandwidth (Băng thông):** Lượng dữ liệu tối đa mà website được truyền tải trong một khoảng thời gian nhất định.
7. **Sub-domain (Tên miền con):** Tên miền phụ được tạo ra từ một tên miền chính, để tổ chức và phân chia nội dung hoặc chức năng của một trang web (ví dụ: blog.example.com phần đứng trước tên miền example).

**Các khái niệm khác:**

1. **FTP (File Transfer Protocol):** Giao thức truyền tải file giữa máy tính cá nhân và máy chủ.
2. **SEO (Search Engine Optimization):** Quá trình tối ưu hóa website để dễ dàng xuất hiện trên kết quả tìm kiếm của Google.
3. **CMS (Content Management System):** Hệ thống quản lý nội dung website (ví dụ: WordPress, Joomla, Drupal).

## Câu hỏi dự đoán

**Câu 1.** Tại sao một website cần nhiều hơn một địa chỉ IP? Hãy giải thích các tình huống sử dụng nhiều địa chỉ IP và lợi ích của việc này.

**Trả lời:** Một website cần nhiều hơn một địa chỉ IP để phân tải, dự phòng, và phân tán địa lý. Phân tải giúp website có thể có nhiều máy chủ để đáp ứng số lượng lớn người truy cập. Dự phòng đảm bảo website vẫn hoạt động nếu một máy chủ gặp sự cố, DNS sẽ định tuyến yêu cầu đến các địa chỉ IP đang hoạt động. Phân tán địa lý giúp người dùng ở gần có tốc độ truy cập tốt hơn nhờ vào việc truy cập các máy chủ gần nhất. Việc dùng nhiều IP giúp tăng khả năng đáp ứng, đảm bảo tính liên tục, cải thiện tốc độ và tối ưu chi phí.

**Câu 2.** Giải thích sự khác biệt cơ bản giữa HTTP và HTTPS. Tại sao HTTPS lại quan trọng trong việc bảo mật thông tin khi duyệt web?

**Trả lời:** HTTP là giao thức truyền dữ liệu trên web ở dạng văn bản thuần túy, dễ bị đánh cắp hoặc thay đổi, còn HTTPS là phiên bản bảo mật hơn, sử dụng mã hóa SSL/TLS để bảo vệ dữ liệu truyền tải giữa trình duyệt và máy chủ, đảm bảo tính bảo mật và toàn vẹn của thông tin cá nhân, ngăn chặn các cuộc tấn công kiểu "man-in-the-middle", và nâng cao uy tín của website.

**Câu 3.** Trong quá trình phát triển web, tại sao cần phải có quá trình tối ưu hóa SEO (Search Engine Optimization)? Hãy giải thích một vài kỹ thuật SEO cơ bản.

**Trả lời:** Tối ưu hóa SEO là cần thiết để giúp website có thứ hạng cao trên kết quả tìm kiếm của Google, giúp người dùng dễ dàng tìm thấy website khi tìm kiếm thông tin liên quan, tăng lượng truy cập tự nhiên, tiếp cận nhiều đối tượng khách hàng tiềm năng. Một số kỹ thuật SEO cơ bản gồm: tối ưu từ khóa, tối ưu nội dung, xây dựng liên kết, và tối ưu tốc độ tải trang.

**Câu 4** Giải thích khái niệm về API (Application Programming Interface) và vai trò của nó trong các ứng dụng web và di động hiện đại.

**Trả lời:** API là tập hợp các quy tắc và giao thức cho phép các ứng dụng phần mềm khác nhau giao tiếp và chia sẻ dữ liệu với nhau, đóng vai trò như "giao diện" giúp các ứng dụng kết nối và trao đổi thông tin, cho phép frontend giao tiếp backend, ứng dụng di động giao tiếp máy chủ, hoặc ứng dụng bên thứ ba tích hợp vào ứng dụng chính, giúp xây dựng các hệ thống phức tạp, linh hoạt và tái sử dụng được các thành phần phần mềm.

**Câu 5.** Giải thích vai trò của CDN (Content Delivery Network) trong việc cải thiện tốc độ tải trang web. Nếu không có CDN, quá trình tải trang web sẽ khác biệt như thế nào?

**Trả lời:** CDN (Content Delivery Network) giúp cải thiện tốc độ tải trang web bằng cách lưu trữ các bản sao của nội dung website trên nhiều máy chủ phân tán trên toàn cầu. Khi người dùng truy cập, CDN sẽ tự động định tuyến yêu cầu đến máy chủ gần nhất, giúp giảm độ trễ và tăng tốc độ tải trang. Nếu không có CDN, trình duyệt của người dùng sẽ phải tải nội dung trực tiếp từ máy chủ gốc, có thể gây chậm trễ đặc biệt khi người dùng ở xa máy chủ hoặc website có lượng truy cập lớn. CDN giúp website hoạt động nhanh hơn, ổn định hơn, và mang lại trải nghiệm tốt hơn cho người dùng.

**Câu 6. Tại sao chúng ta cần quan tâm đến responsive design khi thiết kế website? Các kỹ thuật chính để tạo ra một trang web responsive là gì?**

**Trả lời:** Responsive design là cần thiết để website có thể hiển thị tốt trên nhiều kích thước màn hình khác nhau. Vì số lượng người dùng truy cập web bằng thiết bị di động ngày càng tăng, website cần hiển thị tốt trên mọi thiết bị để đảm bảo trải nghiệm tốt nhất cho người dùng. Các kỹ thuật chính để tạo responsive website gồm: sử dụng flexible layout, sử dụng media queries để điều chỉnh style cho từng kích thước màn hình, và sử dụng hình ảnh responsive.

**Câu 7.** Giải thích tầm quan trọng của việc bảo mật website (web security). Các biện pháp cơ bản để bảo vệ một website khỏi các cuộc tấn công là gì?

**Trả lời:** Bảo mật website rất quan trọng để bảo vệ dữ liệu người dùng, thông tin nhạy cảm, và danh tiếng của website. Các cuộc tấn công mạng có thể gây ra mất dữ liệu, gián đoạn dịch vụ, hoặc thiệt hại về tài chính. Các biện pháp bảo mật cơ bản gồm: sử dụng HTTPS, sử dụng các biện pháp xác thực, chống các cuộc tấn công SQL injection và XSS, thường xuyên cập nhật phần mềm, và sao lưu dữ liệu.

**Câu 8.** Giải thích sự khác biệt giữa website tĩnh và website động. Cho ví dụ minh họa về từng loại.

**Trả lời:** Website tĩnh là website có nội dung cố định, không thay đổi khi người dùng truy cập. Website động là website có nội dung thay đổi tùy theo tương tác của người dùng, hoặc theo dữ liệu được cập nhật từ cơ sở dữ liệu. Ví dụ: website giới thiệu công ty là website tĩnh, còn website thương mại điện tử, mạng xã hội là website động. Website tĩnh dễ tạo nhưng ít tương tác, website động phức tạp hơn nhưng linh hoạt và tương tác tốt hơn.

**Đề 9:** Giải thích khái niệm caching (bộ nhớ đệm) và tại sao nó lại quan trọng trong việc cải thiện hiệu suất của website?

**Trả lời:** Caching là kỹ thuật lưu trữ tạm thời dữ liệu thường xuyên được truy cập ở bộ nhớ có tốc độ truy xuất nhanh, giúp giảm số lần truy xuất đến máy chủ gốc, nhờ đó cải thiện tốc độ tải trang web. Caching có thể được thực hiện ở nhiều cấp độ, giúp giảm tải cho máy chủ, giảm băng thông, và mang lại trải nghiệm tốt hơn cho người dùng.

**Đề 10:**Giải thích vai trò của Load Balancer (bộ cân bằng tải) trong hệ thống web có lượng truy cập lớn.

**Trả lời:** Load Balancer là thành phần phân bổ các yêu cầu của người dùng đến các máy chủ web khác nhau, giúp các máy chủ không bị quá tải, đảm bảo website hoạt động ổn định và nhanh chóng. Load balancer đóng vai trò quan trọng trong việc đảm bảo tính sẵn sàng và khả năng chịu tải của website.

**Đề 11:** Giải thích mối quan hệ giữa HTML, CSS, và JavaScript trong việc xây dựng một trang web. Nếu thiếu một trong ba thành phần này, trang web sẽ như thế nào?

**Trả lời:** HTML xây dựng cấu trúc và nội dung, CSS định kiểu dáng và màu sắc, còn JavaScript thêm chức năng và tương tác cho trang web. Nếu thiếu HTML, trang web không có nội dung. Nếu thiếu CSS, trang web sẽ đơn điệu và khó nhìn. Nếu thiếu JavaScript, trang web sẽ thiếu tương tác và không sống động. Ba thành phần này kết hợp để tạo nên một trang web hoàn chỉnh.

**Đề 12:** Giải thích tầm quan trọng của việc thiết kế giao diện người dùng (UI) và trải nghiệm người dùng (UX) trong việc phát triển một website thành công.

**Trả lời:** Thiết kế UI tập trung vào vẻ đẹp của giao diện, đảm bảo giao diện trực quan và dễ nhìn. Thiết kế UX tập trung vào trải nghiệm tổng thể của người dùng, đảm bảo website dễ sử dụng, thân thiện, và mang lại giá trị. Cả UI và UX đều rất quan trọng trong việc thu hút và giữ chân người dùng, tăng tỷ lệ chuyển đổi và tạo sự thành công cho website.

**Đề 13:** Bạn hiểu như thế nào về khái niệm “tối ưu hóa hình ảnh” cho website? Các kỹ thuật tối ưu hóa hình ảnh cơ bản là gì?

**Trả lời:** Tối ưu hóa hình ảnh là quá trình giảm dung lượng file ảnh mà vẫn giữ được chất lượng, giúp website tải nhanh hơn và cải thiện trải nghiệm người dùng. Các kỹ thuật tối ưu hóa hình ảnh gồm: lựa chọn định dạng ảnh phù hợp (JPEG cho ảnh chụp, PNG cho đồ họa, WebP cho cả hai), nén ảnh để giảm kích thước, thay đổi kích thước ảnh cho phù hợp với kích thước hiển thị, và sử dụng lazy loading để chỉ tải ảnh khi chúng cần hiển thị.

## Một số cú pháp thẻ cần lưu ý

<a href="" target="" title="">

<img src="" alt="" width="" height="">

<form action="" method="" enctype="">

<input type="" name="" value="" placeholder="" required readonly disabled min="" max="" pattern="">

<textarea name="" rows="" cols="">

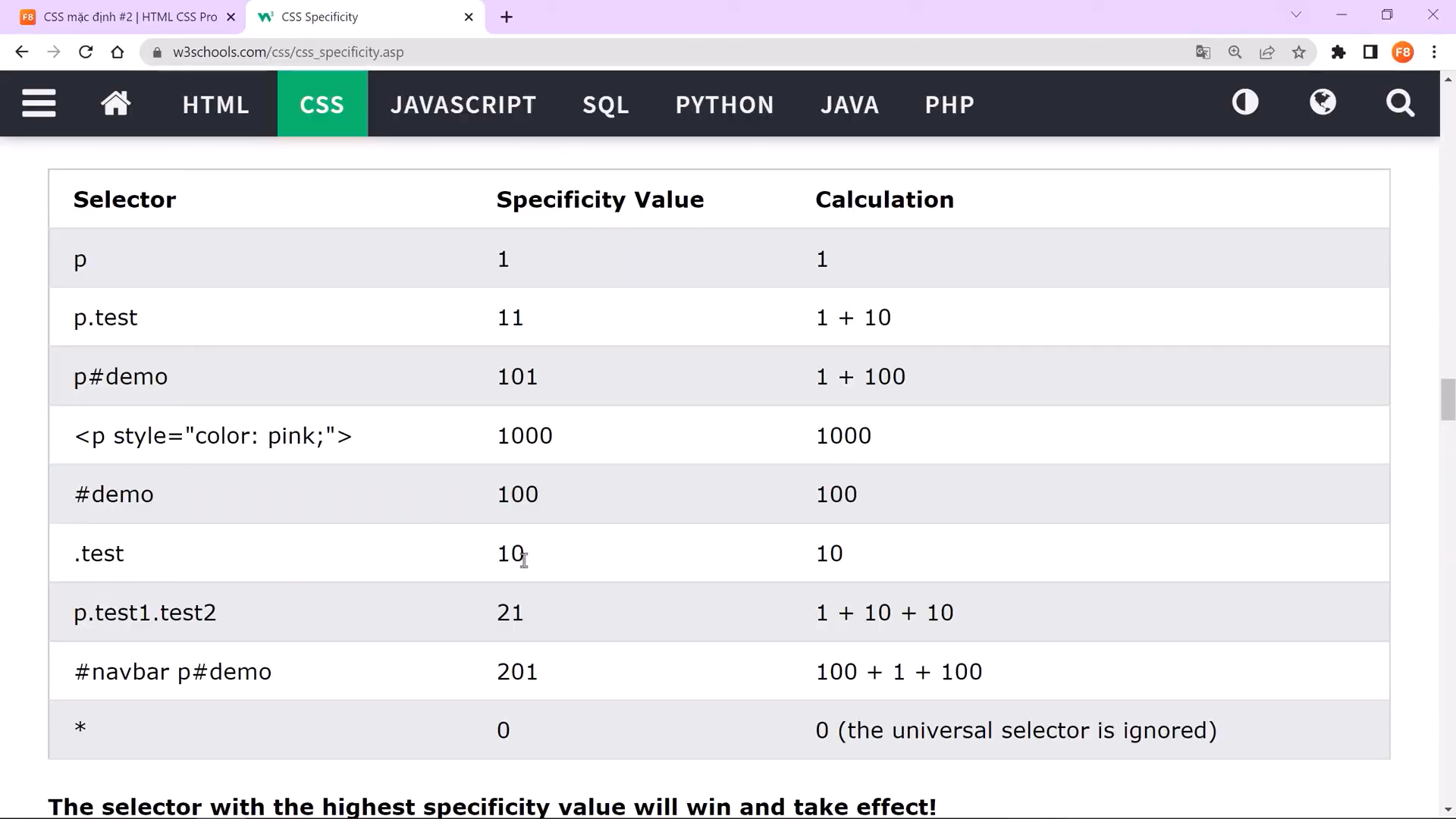
<select name="">

<option value="">

<button type="" name="" value="">

<label for="">

## CSS specificity (độ ưu tiên selectors)



## Quan điểm khi thiết kế giao diện website:

Khi thiết kế giao diện website, chúng ta cần phải đặt người dùng ở vị trí trung tâm, và luôn nhớ rằng giao diện không chỉ là về tính thẩm mỹ, mà còn về trải nghiệm người dùng. Một số quan điểm quan trọng khi thiết kế giao diện website bao gồm:

1. **Ưu tiên sự đơn giản (Simplicity):**
   * Giao diện phải rõ ràng, dễ hiểu, không quá phức tạp hay rối mắt.
   * Giảm thiểu các yếu tố không cần thiết, tập trung vào những nội dung và chức năng quan trọng.
   * Sử dụng bố cục rõ ràng, dễ theo dõi.
2. **Tính nhất quán (Consistency):**
   * Sử dụng các màu sắc, font chữ, kiểu nút, và bố cục đồng nhất trên toàn bộ website.
   * Giữ cho các thành phần tương tác có cùng hành vi (ví dụ: các nút bấm có cùng kiểu).
   * Tạo ra một trải nghiệm quen thuộc, dễ dự đoán cho người dùng.
3. **Tính trực quan (Intuitive):**
   * Giao diện phải dễ sử dụng, người dùng có thể tìm thấy những gì họ cần một cách nhanh chóng và dễ dàng.
   * Sử dụng các biểu tượng, hình ảnh, và màu sắc phù hợp để hướng dẫn người dùng.
   * Tránh sử dụng các thuật ngữ chuyên môn khó hiểu, mà sử dụng ngôn ngữ đơn giản, dễ tiếp cận.
4. **Tính phản hồi (Responsiveness):**
   * Website phải hiển thị tốt trên mọi loại thiết bị (ví dụ: máy tính, điện thoại, máy tính bảng).
   * Giao diện phải co giãn và điều chỉnh kích thước phù hợp với từng loại màn hình.
   * Đảm bảo trải nghiệm người dùng nhất quán trên mọi thiết bị.
5. **Tính dễ tiếp cận (Accessibility):**
   * Website phải dễ sử dụng cho tất cả mọi người, bao gồm cả những người khuyết tật.
   * Sử dụng màu sắc tương phản đủ tốt, cung cấp văn bản thay thế cho hình ảnh (alt text), và hỗ trợ bàn phím.
   * Tuân thủ các tiêu chuẩn về khả năng tiếp cận web (WCAG).
6. **Tập trung vào người dùng (User-centric):**
   * Đặt nhu cầu và mong muốn của người dùng lên hàng đầu trong quá trình thiết kế.
   * Thường xuyên thu thập phản hồi từ người dùng và cải thiện website dựa trên những phản hồi đó.
   * Thiết kế hướng đến mục tiêu cụ thể của người dùng khi truy cập website.
7. **Tính thẩm mỹ (Aesthetics):**
   * Giao diện phải đẹp mắt, phù hợp với thương hiệu và đối tượng mục tiêu.
   * Sử dụng màu sắc, font chữ, hình ảnh, và bố cục một cách hài hòa.
   * Tạo ra một giao diện chuyên nghiệp và đáng tin cậy.

## 

## Nguyên tắc C.R.A.P:

Nguyên tắc C.R.A.P là một nguyên tắc thiết kế trực quan cơ bản, được sử dụng để tạo ra các thiết kế rõ ràng, dễ đọc và có tổ chức. C.R.A.P là viết tắt của:

* **C - Contrast (Tương phản):**
  + Tạo sự khác biệt rõ ràng giữa các yếu tố trong thiết kế để chúng nổi bật lên.
  + Sử dụng màu sắc, kích thước, và font chữ tương phản để thu hút sự chú ý của người dùng.
  + Ví dụ: sử dụng chữ màu sáng trên nền màu tối, hoặc chữ đậm và chữ nhạt để tạo điểm nhấn.
* **R - Repetition (Sự lặp lại):**
  + Sử dụng các yếu tố thiết kế lặp đi lặp lại để tạo ra sự thống nhất và nhất quán.
  + Lặp lại màu sắc, font chữ, hình dạng, bố cục, ... trên toàn bộ thiết kế.
  + Giúp người dùng dễ dàng nhận diện và ghi nhớ các thành phần của trang web.
* **A - Alignment (Sự căn chỉnh):**
  + Sắp xếp các yếu tố trong thiết kế một cách có trật tự và ngăn nắp.
  + Căn chỉnh các yếu tố theo chiều ngang, chiều dọc, hoặc theo một grid.
  + Giúp tạo ra một giao diện rõ ràng, dễ đọc và có tổ chức.
* **P - Proximity (Sự gần gũi):**
  + Gộp các yếu tố có liên quan đến nhau lại gần nhau, và tách các yếu tố không liên quan ra xa nhau.
  + Giúp người dùng dễ dàng nhận biết mối quan hệ giữa các thành phần trong thiết kế.
  + Tạo ra một bố cục logic và dễ hiểu.

## Quá trình xây dựng website

1. **Requirements (Xác định yêu cầu):** Đây là giai đoạn đầu tiên và quan trọng nhất, tập trung vào việc **thu thập và phân tích** các yêu cầu của website. Giai đoạn này bao gồm việc xác định rõ **mục đích, đối tượng người dùng, các chức năng (yêu cầu chức năng), nội dung, và các đặc tính (yêu cầu phi chức năng)** như hiệu suất, bảo mật. Kết quả của giai đoạn này là một tài liệu đặc tả yêu cầu làm cơ sở cho các bước tiếp theo.
2. **Design (Thiết kế):** Dựa trên các yêu cầu đã xác định, giai đoạn này tập trung vào việc thiết kế **giao diện người dùng (UI)** và **trải nghiệm người dùng (UX)** của website. Các hoạt động chính bao gồm: xác định cấu trúc website, bố cục các trang, thiết kế **wireframe (bản phác thảo)**, **mockup (bản thiết kế chi tiết)**, và **prototype (bản demo có thể tương tác)**. Đồng thời, cũng lựa chọn màu sắc, font chữ, hình ảnh, và các yếu tố thiết kế khác sao cho phù hợp.
3. **Implementation & assembly (Xây dựng và lắp ráp):** Đây là giai đoạn **triển khai code** cho website, bao gồm cả **frontend (HTML, CSS, JavaScript)** để xây dựng giao diện người dùng, và **backend (ví dụ: PHP, Python, Node.js)** để xử lý logic nghiệp vụ, kết nối với **database (cơ sở dữ liệu)**, và tích hợp các **API (Application Programming Interface)** cần thiết. Giai đoạn này cũng bao gồm việc tạo nội dung cho website.
4. **Quality control & verification (Kiểm soát chất lượng và xác minh):** Ở giai đoạn này, website sẽ được **kiểm tra kỹ lưỡng** về chức năng, nội dung, giao diện, và hiệu suất. Các hoạt động chính bao gồm: **phát hiện và sửa lỗi (bug)**, thực hiện các bài kiểm thử như **unit test (kiểm thử đơn vị)**, **integration test (kiểm thử tích hợp)**, và **UAT (user acceptance test – kiểm thử chấp nhận)**. Mục tiêu là đảm bảo website hoạt động đúng như mong đợi.
5. **Site launch (Ra mắt website):** Sau khi đã kiểm tra và xác nhận chất lượng, website sẽ được **triển khai lên máy chủ (hosting)**, **cấu hình DNS (Domain Name System)** và đưa vào hoạt động chính thức trên internet. Cần phải theo dõi và đảm bảo website hoạt động ổn định sau khi ra mắt.
6. **Maintenance (Bảo trì):** Đây là giai đoạn sau khi website đã được ra mắt, bao gồm các hoạt động **duy trì và bảo trì** website. Các hoạt động chính bao gồm: sửa lỗi phát sinh, cập nhật nội dung, nâng cấp tính năng, theo dõi hiệu suất của website, và đảm bảo website luôn hoạt động tốt và đáp ứng nhu cầu của người dùng.