**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

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

**TÊN ĐỀ TÀI: THIẾT KẾ WEB BÁN THIẾT BỊ Y TẾ**

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

**Sinh viên thực hiện:**  **Lê Đức Thọ**

**Hà Nội, 2024**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

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

**TÊN ĐỀ TÀI THIẾT KẾ WEB BÁN THIẾT BỊ Y TẾ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
|  | 1771020651 | Lê Đức Thọ | 26/01/2005 |  |  |

### 

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

**Hà Nội, năm 2024**

**LỜI NÓI ĐẦU**

**Thiết kế web** là một lĩnh vực đầy sáng tạo và năng động, nơi nghệ thuật và kỹ thuật hòa quyện để tạo ra những trải nghiệm trực tuyến độc đáo. Là một nhà thiết kế web, tôi luôn say mê với việc biến những ý tưởng thành hiện thực, mang đến cho người dùng những trang web đẹp mắt, hiệu quả và dễ sử dụng.

Thiết kế web là một học phần không thể thiếu trong ngành công nghệ thông tin. Học phần Thiết kế web giúp chúng ta biết được cách tạo lập một trang web, cách sử dụng CSS làm cho web cùa chúng ta trông có màu sắc, nổi bật hơn. Hiểu được thêm về hệ thống các thẻ trong web.

Với đề tài là thiết kế trang web giới thiệu du lịch vùng Tây Bắc do thầy Lê Văn Phong đưa ra, em xin trình bày quy trình các bước thực hiện để từ đó mọi người có thể biết đến các thao tác làm web, biết đến các thẻ được dùng trong web, web được dùng với mục đích gì một cách dễ dàng hơn, thuận tiện mà dễ hiểu.

Cũng với đề tài trình bày thiết kế website quản lý điểm sinh viên, em xin khái quát qua các chương của bài tập lớn.

**Chương 1**: **GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM**

Ở chương này em xin giới thiệu về thiết kế web là gì, cùng với những triển khai hệ thống phần mềm máy tính.

**Chương 2:** **HTML &AMP; HTML5 (HYPER TEXT MARKUP LANGUAGE)**

Chương 2 em xin giới thiệu về HTML &amp; HTML5 và những dư kiện liên quan đến HTML.

**Chương 3: CSS VÀ CSS3 (CASCADING STYLE SHEETS)**

Về chương này, em xin trình bày cơ bản CSS,CSS3 và về các phần liên quan đến CSS và CSS3.

**Chương 4:** **THIẾT KẾ TRANG WEB BÁN THIẾT BỊ Y TẾ**

Chương 4 : ý tưởng, cách xây dựng bố cục web, thiết kế trang web bằng HTML và CSS và kết quả đạt được.

Bài tập lớn này sẽ là tài liệu bổ ích dành cho những ai đang cần tìm hiểu đề tài thiết kế trang web bán thiết bị y tế ; nó cũng là tài liệu học tập hoặc tài liệu tham khảo cho học sinh trung cấp tin học cũng như tân sinh viên ngành công nghệ thông tin.

Em cũng xin chân thành cảm ơn vì có sự hỗ trợ rất lớn từ thầy Lê Văn Phong – Giảng viên Hệ thống máy tính, Khoa Công nghệ thông tin để em có được những định hướng đúng đắn quan trọng .

Vì là lần đầu làm web nên chắc chắn sẽ không tránh khỏi những sai sót , hy vọng Thầy và các bạn góp ý để em rút kinh nghiệm những lần sau .

**MỤC LỤC**

[**LỜI NÓI ĐẦU**..........................................................................................................3](#_Toc161284692)

[**MỤC LỤC** 5](#_Toc161284701)

[**CHƯƠNG I: GIỚI THIỆU VỀ THIẾT KẾ WEB VÀ TRIỂN KHAI HỆ THỐNG PHẦN MỀM** 7](#_Toc161284702)

[**1.1. Các khái niệm cơ bản** 7](#_Toc161284703)

[**1.2. Phân biệt Web tĩnh, Web động** 7](#_Toc161284704)

[**1.3. Một số thuật ngữ (Hosting, Tên miền-Domain Name-Web Server Name, HomePage, Web Site, WebServer, URL-Uniform Resource Location, Browser, …)** 8](#_Toc161284705)

[**1.4. Một số công cụ dùng thiết kế Web (Visual Studio code, Adobe Dreamweaver, …)** 12](#_Toc161284706)

[**Kết luận chương 1** 15](#_Toc161284707)

[**CHƯƠNG II: HTML & HTML5** 16](#_Toc161284708)

[**2.1. Tổng quan về HTML** 16](#_Toc161284709)

[**2.2. Cấu trúc tổng quát trang HTML** 19](#_Toc161284710)

[**2.3. Các thẻ HTML thông dụng** 20](#_Toc161284711)

[**2.4. Các thẻ tạo biểu mẫu (form)** 28](#_Toc161284714)

[**2.5. Một số thẻ HTML đặc biệt** 30](#_Toc161284715)

[**2.6. HTML5** 31](#_Toc161284716)

[**Kết luận chương 2** 36](#_Toc161284717)

[**CHƯƠNG III: CSS VÀ CSS3 (CASCADING STYLE SHEETS)** 38](#_Toc161284718)

[**3.1. CSS là gì?** 38](#_Toc161284719)

[**3.2. Cú pháp CSS** 38](#_Toc161284720)

[**3.3. Áp dụng CSS vào trang HTML** 39](#_Toc161284722)

[**3.4. Selector** 41](#_Toc161284723)

[***3.4.1. Universal selector*** 41](#_Toc161284724)

[***3.4.2. Type selector*** 41](#_Toc161284725)

[***3.4.3. Identity selector*** 42](#_Toc161284726)

[***3.4.4. Class selector*** 43](#_Toc161284727)

[***3.4.5. Descendant selector*** 43](#_Toc161284728)

[***3.4.6. Child selector*** 44](#_Toc161284729)

[***3.4.7. Adjacent sibling selector*** 44](#_Toc161284730)

[***3.4.8. Attribute selector*** 44](#_Toc161284731)

[***3.4.9. Pseudo class selector*** 45](#_Toc161284732)

[***3.4.10. Group selector*** 46](#_Toc161284733)

[**3.5. Đơn vị đo lường CSS** 47](#_Toc161284734)

[**3.6. Kế thừa thuộc tính** 48](#_Toc161284735)

[**3.7. Các nhóm thuộc tính trong CSS** 49](#_Toc161284736)

[**3.8. Float &amp; Clear** 50](#_Toc161284737)

[**3.9. Flex** 51](#_Toc161284738)

[**3.10. Grid** 51](#_Toc161284739)

[**3.11. CSS3** 52](#_Toc161284740)

[**3.12. SCSS** 57](#_Toc161284741)

[**3.13. SASS** 57](#_Toc161284742)

[**Kết luận chương 3** 58](#_Toc161284743)

[**4.1.** **Ý tưởng của Website** 59](#_Toc161284744)

[**4.2. Xây dựng bố cục của trang Web** 59](#_Toc161284745)

[**4.4**. **Kết quả đạt được** 62](#_Toc161284746)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 65](#_Toc161284749)

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

**1.1. Các khái niệm cơ bản**

*Web-www(Word Wide Web)*

Được gọi là web, là mạng lưới nguồn thông tin cho phép mọi người khai thác thông tin qua một số công cụ hoặc là chương trình hoạt động dưới các giao thức mạng. Word Wide Web là một trong những số các dịch vụ của Internet nhằm giúp cho việc trao đổi thông tin trở nên thuận tiện và dễ dàng hơn.

*Website*

Là tập hợp các tài liệu web (webpage,file định dạng CSS, các script, hình ảnh,âm thanh...) có liên quan với nhau, được đặt tại ít nhất một máy phục vụ web (web sever).

Các thành phần cơ bản của web:

-Siêu văn bản (Hyper Text)

-Ngôn ngữ đánh giấu siêu văn bản (HTML)

-Giao thức truyên tin siêu văn bản (HTTP)

-Trình duyệt web (web browser)

-URL (Uniform Resource Locator)

**1.2. Phân biệt Web tĩnh, Web động**

*Web tĩnh*

Tương tác yếu với người dùng, nội dung của trang web không bị thay đổi theo thời gian cho đến khi người lập trình thay đổi code của trang web.Trang Web tĩnh thường là các tài liệu [HTML](https://vi.wikipedia.org/wiki/HTML) được lưu dưới dạng tệp trong [hệ thống tệp](https://vi.wikipedia.org/wiki/H%E1%BB%87_th%E1%BB%91ng_t%E1%BA%ADp_tin) và được máy chủ web cung cấp qua [HTTP](https://vi.wikipedia.org/wiki/Hypertext_Transfer_Protocol) (tuy nhiên [các URL](https://vi.wikipedia.org/wiki/URL) kết thúc bằng ".html" không phải lúc nào cũng tĩnh). Tuy nhiên, việc giải thích thuật ngữ lỏng lẻo có thể bao gồm các trang web được lưu trữ trong [cơ sở dữ liệu](https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u) và thậm chí có thể bao gồm các trang được định dạng bằng mẫu và được phân phối qua máy chủ ứng dụng, miễn là trang được cung cấp không thay đổi và được trình bày về cơ bản là giống như khi nó được lưu trữ.

*Web động*

Tương tác mạnh với người dùng, nội dung trang web bị thay đổi tùy theo ngữ cảnh người dùng yêu cầu.Trang Web động được xây dựng bằng HTML kết hợp với các ngôn ngữ văn bản dành cho sever.Trang Web động thường có yêu cầu truy xuất dữ liệu từ database.

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

*Hosting*

Hosting (tên gọi khác là web hosting) là dịch vụ lưu trữ và duy trì website. Không gian lưu trữ này được chia nhỏ trên máy chủ nhằm giúp người dùng có thể chia sẻ, lưu trữ trực tuyến thông qua kết nối mạng. Hiểu đơn giản, website muốn hoạt động cần được lưu trữ trên một máy chủ (gọi là sever) được kết nối mạng Internet và hoạt động liên tục, không ngắt quãng.

Ví website là nhà thì hosting chính là mảnh đất để xây lên ngôi nhà đó.Hiện nay, có 4 hình thức cho thuê Hosting, mỗi hình thức như vậy sẽ phù hợp với 4 đối tượng cũng như các mục đích sử dụng khác nhau.

Share Hosting: là gói hosting chia sẻ, có chi phí thuê rẻ nhất trong số các dịch vụ cho thuê hosting. Thông thường những website có lượng truy cập thấp, không cần sử dụng nhiều tài nguyên thì sẽ thuê share hosting.

***Collocated Hosting*** (thuê chỗ đặt máy chủ): đây là gói hosting dành cho những công ty, doanh nghiệp, website của họ có tổng sổ lượng truy cập cao mỗi ngày. Đây là dạng hosting mà khách hàng có quyền tự cài đặt và cấu hình máy chủ.

***Dedicated Server*** (máy chủ dành riêng): hosting dạng này giống gói Collocated Hosting, khách hàng hoàn toàn có quyền điều khiển máy chủ của mình, đây là loại hositng có giá đắt nhất trong tổng sổ các loại hosting thường dùng.

***Virtual Private Server*** (VPS): hay còn gọi là máy chủ ảo riêng, nhà cung cấp sẽ chia một máy chủ vật lý thành nhiều máy chủ ảo khác nhau, trên mỗi máy chủ áo sẽ được cài một hệ điều hành riêng biệt như làm một máy chủ thật. Hiện nay số lượng website sử dụng vps tăng khá cao, nhất là những website lớn, có số lượng truy cập khủng.

*Tên miền-Domain Name-Web sever Name*

Domain được dịch sang Tiếng Việt nghĩa là Tên miền website. Như chúng ta vẫn thấy nó là dòng chữ không dấu có cấu trúc gồm từ 2 đến 3 phần được cách nhau bởi dấu chấm (.) trên thanh địa chỉ của trình duyệt web. Hiểu đơn giản thì tên miền chính là địa chỉ nhà của chúng ta . Mỗi ngôi nhà có một địa chỉ riêng và không trùng lặp, website cũng vậy.

Để trao đổi với nhau một cách dễ dàng, các thiết bị trong mạng kết nối Internet được đặt tên theo quy ước ngôn ngữ của chúng. Hiện nay, quy ước này được đặt tên có dạng một dãy số gồm 4 phần, mỗi phần gồm 3 chữ số từ 0 đến 9 và được cách nhau bởi dấu chấm (.). Ví dụ, tên website trên hệ thống máy tính được ghi nhận là một địa chỉ ip có dạng 103.63.213.148. Không chỉ là cách đặt tên, mà cách các máy tính truyền tin, giao tiếp với nhau cũng dùng thứ ngôn ngữ thập phân này.

Nhưng có hàng triệu website trên thế giớ này, và nếu truy cập website theo địa chỉ ip thì rất khó nhớ, nên người ta đã chuyển ip đó thành domain (tên miền) thông qua DNS, đó là lý do tên miền ra đời.

*Cấu tạo của tên miền*

Chi tiết hơn, một tên miền sẽ được tạo thành bởi hai phần chính, phần đầu là tên thương hiệu, phần sau tên loại miền.

Thành phần sau dấu chấm cuối cùng của tên miền được gọi là tên miền cao cấp nhất (Top Level Domain Name). Trên thị trường cung cấp [***Domain***](https://z.com/vn/) trên thế giới dùng chung các tên miền cao cấp sau:

* .com (thương mại)
* .net (mạng lưới)
* .org (các tổ chức)
* .info (thông tin)
* .edu (giáo dục)
* .mobi (điện thoại di động)

Ngoài ra, nếu chúng ta muốn có một tên miền độc đáo mang dấu ấn riêng của mình, chúng ta có thể mua những tên miền cấp cao riêng như mona.media chẳng hạn.

Đối với những doanh nghiệp, công ty, cá nhân chuyên kinh doanh online thì việc lựa chọn tên miền rất quan trọng, vì nó có tác động nhất định đến hoạt động marketing online, nhất là đối với việc seo website.

Khâu chọn lựa tên miền rất quan trọng, có ba dạng tên miền thường gặp, thứ nhất là tên miền thương hiệu, ví dụ sikido.vn, vnexpress.net… tên miền chỉ gắn liền với thương hiệu, tên gọi của công ty, doanh nghiệp. Thứ hai là tên miền nữa thương hiệu nữa ngành nghề, ví dụ freelancervietnam.com. Thứ ba là tên miền chỉ mô tả chuyên ngành, dịch vụ, ví dụ thietkewebre.vn, thietkewebdep.vn…

*Homepage*

Homepage hay còn được gọi là trang chủ của website, giao diện hiện lên đầu tiên khi người dùng truy cập vào một địa chỉ trang web nào đó. Từ trang chủ này, chúng ta sẽ có thanh điều hướng, một số thông tin quan trọng (về doanh nghiệp, sản phẩm, dịch vụ) và danh mục chi tiết để tìm đến nội dung của các trang con.

Trang chủ nằm trong thư mục gốc của trang web. Nhiều trang chủ hoạt động như một thư mục ảo cho trang web đó. Trang chủ đóng vai trò quan trọng với tất cả các website và hầu hết, website nào cũng có một homepage. Từ trang chủ, người dùng có thể dễ dàng truy cập và đi sâu hơn vào các khu vực khác trong website, bao quát được toàn website.

Ví dụ: Một trang web điển hình có trang chủ và các mục menu như:

* *Giới thiệu*
* *Dịch vụ*
* *Sản phẩm*
* *Liên hệ*
* *Tin tức*

Ngoài ra, trang chủ còn đóng vai trò để định hướng khách truy cập. Nó cung cấp cho người dùng các tiêu đề trang, hình ảnh, sơ đồ trang. Nhờ đó, người dùng có thể dễ dàng biết các thông tin về các đơn vị chủ quản của web qua tên doanh nghiệp, hình ảnh liên quan đến doanh nghiệp đó, các thông tin về số điện thoại, email,...

Không có bố cục trang chủ tiêu chuẩn nào được đặt ra. Nhưng hầu hết các trang chủ của web đều được thiết kế đẹp mắt, nổi bật nhất so với các trang con bên trong để thu hút người truy cập, gây ấn tượng đầu tiên cho khách truy cập.

Chức năng của trang chủ cũng bao gồm các thanh điều hướng thông minh cung cấp liên kết đến các phần khác nhau trong trang web. Bao gồm thanh tìm kiếm, thông tin về trang web và tin tức hoặc cập nhật gần đây.

***Web Site***

**Website**còn gọi là trang mạng hoặc trang web, là một tên miền (domain) chứa nội dung có liên quan và được phát hành thông qua ít nhất một máy chủ web (web server). Ví dụ nổi bật về website là [mikotech.vn](https://mikotech.vn/), [wikipedia.org](https://vi.wikipedia.org/wiki/Wikipedia), [google.com](https://vi.wikipedia.org/wiki/Google) và amazon.com.

Website là tập hợp các trang có chứa văn bản, hình ảnh, video, âm thanh, v.v… Website thường chỉ nằm trong một tên miền (domain name) hoặc tên miền phụ (subdomain) và chỉ có thể truy cập khi có Internet.

Một website là tập tin [**HTML**](https://mikotech.vn/html-la-gi/) hoặc XHTML có thể truy nhập dùng giao thức HTTP hoặc [**HTTPS**](https://mikotech.vn/https-la-gi/). Website có thể được xây dựng từ các tệp tin HTML (website tĩnh) hoặc vận hành bằng các [CMS](https://mikotech.vn/cms-la-gi/) chạy trên máy chủ (website động). Website có thể được xây dựng bằng nhiều ngôn ngữ lập trình khác nhau ([**PHP**](https://mikotech.vn/ngon-ngu-lap-trinh-php/)**,**[**.NET**](https://mikotech.vn/net-la-gi/), Java, Ruby on Rails,…).

***Web sever***

**Web server** là máy chủ cài đặt các chương trình phục vụ các ứng dụng web. Webserver có khả năng tiếp nhận request từ các trình duyệt web và gửi phản hồi đến client thông qua giao thức HTTP hoặc các giao thức khác. Có nhiều web server khác nhau như: Apache, [Nginx](https://topdev.vn/blog/nginx-la-gi/), IIS, … Web server thông dụng nhất hiện nay.

Bất cứ khi nào chúng ta xem một trang web trên internet, có nghĩa là chúng ta đang yêu cầu trang đó từ một web server. Khi chúng ta nhập URL trên trình duyệt của mình (ví dụ: [https://topdev.vn](https://topdev.vn/)) nó sẽ tiến hành các bước sau để gửi lại phản hồi cho chúng ta.

*Trình duyệt phân giải tên miền thành địa chỉ IP*

Trình duyệt web của chúng ta trước tiên cần phải xác định địa chỉ IP nào mà tên miền topdev.vn trỏ về. Trình duyệt sẽ yêu cầu thông tin từ một hoặc nhiều máy chủ DNS (thông qua internet). Máy chủ DNS sẽ cho trình duyệt biết địa chỉ IP nào tên miền sẽ trỏ đến cũng là nơi đặt trang web.Lúc này trình duyệt web đã biết địa chỉ IP của trang web, nó có thể yêu cầu URL đầy đủ từ webserver.

***Webserver gửi lại client Trang được yêu cầu***

Web server phản hồi bằng cách gửi lại những thông tin client yêu cầu… Nếu trang không tồn tại hoặc có lỗi khác xảy ra, nó sẽ gửi lại thông báo lỗi thích hợp.

***Trình duyệt hiển thị trang web***

Trình duyệt web của chúng ta nhận lại được các tập tin html css (nhiều file khác)… và render hiển thị trang theo yêu cầu.

*URL-Uniform Resouse Location*

Uniform Resource Locator (viết tắt là URL, nghĩa tiếng Việt: Hệ thống định vị tài nguyên thống nhất; được gọi một cách thông thường là một địa chỉ web ) là một tham chiếu đến [tài nguyên web](https://vi.wikipedia.org/wiki/T%C3%A0i_nguy%C3%AAn_web) chỉ định vị trí của nó trên một [mạng máy tính](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_m%C3%A1y_t%C3%ADnh) và cơ chế để truy xuất nó. URL là một loại [Mã nhận dạng tài nguyên thống nhất](https://vi.wikipedia.org/wiki/URI) (Uniform Resource Identifier).

Hầu hết [các trình duyệt web](https://vi.wikipedia.org/wiki/Tr%C3%ACnh_duy%E1%BB%87t_web) hiển thị URL của một trang web phía trên trang trong một [thanh địa chỉ](https://vi.wikipedia.org/wiki/Thanh_%C4%91%E1%BB%8Ba_ch%E1%BB%89). Một URL thông thường có thể có dạng http: www.exemple.com\index.html cho biết một giao thức (HTTP), [tên máy chủ](https://vi.wikipedia.org/wiki/M%C3%A1y_ch%E1%BB%A7) www.exemple.com và tên file index.html.

*Browser*

Browser là gì? Web Browser hay còn gọi là Browser được dịch ra tiếng Việt là trình duyệt web.Công dụng của trình duyệt web chính là giúp chúng ta có thể truy cập các trang web mà chúng ta muốn và thực hiện các hoạt động trên đó như: Tìm kiếm thông tin, xem phim, đọc sách báo, chơi game, đăng nhập tài khoản thuộc trang web,…

Các trang web đều có hình ảnh, video, văn bản,… được xác định bằng URL riêng biệt cho phép trình duyệt web nhận biết, truy xuất và hiển thị chúng trên thiết bị của người dùng.

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

*Web-www(Word Wide Web)*

Được gọi là web, là mạng lưới nguồn thông tin cho phép mọi người khai thác thông tin qua một số công cụ hoặc là chương trình hoạt động dưới các giao thức mạng. Word Wide Web là một trong những số các dịch vụ của Internet nhằm giúp cho việc trao đổi thông tin trở nên thuận tiện và dễ dàng hơn.

*Website*

Là tập hợp các tài liệu web (webpage,file định dạng CSS, các script, hình ảnh,âm thanh...) có liên quan với nhau, được đặt tại ít nhất một máy phục vụ web (web sever).

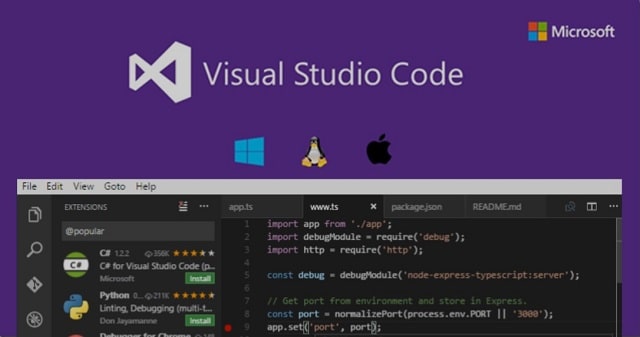
Các thành phần cơ bản của web:

* Siêu văn bản (Hyper Text)
* Ngôn ngữ đánh giấu siêu văn bản (HTML)
* Giao thức truyên tin siêu văn bản (HTTP)
* Trình duyệt web (web browser)
* URL (Uniform Resource Locator)

*Visua Studio Code*

Ưu điểm:

* Có thể tùy chỉnh
* Nhiều tích hợp
* Các công cụ kiểm soát nguồn



*Hình 5.Visual Studio Code*

Visual Studio Code (hay VScode) đã trở nên phổ biến trong số các nhà phát triển – những người không thích bởi thực tế rằng đó là một sản phẩm của Microsoft. Nó cung cấp nhiều gói và tiện ích mở rộng miễn phí có thể được tải xuống từ thị trường của nó để thêm các tính năng bổ sung – và bản thân trình chỉnh sửa mã có thể được tùy chỉnh.

Mặc dù nhiều người có thể bị ám ảnh bởi ý tưởng viết mã nếu họ không có kinh nghiệm lập trình, nhưng tại một số thời điểm trong quá trình thiết kế web, điều quan trọng là có thể chỉnh sửa mã trực tiếp. Điều này đặc biệt hữu ích nếu chúng ta muốn tùy chỉnh mã cho các tiện ích bổ sung phần mềm hiện có để có được các tính năng chúng ta muốn cho trang web của mình. Visual Studio Code có thể giúp quá trình này bớt khó khăn hơn bằng cách cung cấp một nền tảng chuyên dụng để viết mã.

Visual Studio Code có thiết bị đầu cuối và trình gỡ lỗi của riêng nó, hỗ trợ linting và tích hợp với tất cả các cách thức của các công cụ kiểm soát nguồn. Nó đặc biệt cung cấp hỗ trợ tốt cho Javascript và Python, làm cho nó có khả năng hữu ích không chỉ để mã hóa giao diện người dùng của một trang web mà còn có khả năng là một số chức năng phụ trợ.

*Adobe Dreamweaver*

Ưu điểm:

* Tùy chỉnh thiết kế của chúng ta
* Xây dựng một thiết kế đáp ứng
* Hỗ trợ HTML5

Nhược điểm:

* Giá khá cao



*Hình 6.Adobe Dreamweaver CC*

Adobe Dreamweaver  là một ứng dụng lâu đời cho phép chúng ta viết mã thiết kế trang web của mình một cách trực tiếp mà không cần phải biết quá nhiều về lập trình. Phần mềm hoạt động thông qua sự kết hợp giữa chỉnh sửa trực quan và chỉnh sửa HTML, có nghĩa là nó không nên có đường cong học tập quá dốc đối với hầu hết người dùng.

Ngoài ra, mặc dù việc viết mã thiết kế trang web của riêng chúng ta đòi hỏi chúng ta phải nỗ lực nhiều hơn là chỉ sử dụng một mẫu đã tạo sẵn hiện có, nhưng ít nhất chúng ta có cơ hội đảm bảo rằng chúng ta có được giao diện thực sự mong muốn, thay vì cố gắng làm việc với các thông số kỹ thuật thiết kế của người khác .

Một tính năng đặc biệt tốt của Dreamweaver là nó cho phép chúng ta tạo ra một thiết kế đáp ứng, có nghĩa là trang web của chúng ta có thể được tối ưu hóa để hiển thị trên máy tính để bàn cũng như thiết bị di động mà không giới hạn trải nghiệm người dùng.

Tuy nhiên, nếu chúng ta là người dùng cao cấp hơn, chúng ta có thể hài lòng khi lưu ý rằng Dreamweaver cung cấp hỗ trợ đa màn hình cho Windows. Nó cũng hỗ trợ Khung nhúng Chromium (CEF) để làm việc với các phần tử HTML5 và lưới CSS. Ngoài ra, Git cũng hỗ trợ cho phép chúng ta chỉnh sửa mã nguồn trực tiếp từ bên trong Dreamweaver.

Dreamweaver có sẵn như một phần của dịch vụ đăng ký ứng dụng đóng gói tầm trung của Adobe, Creative Cloud, cũng bao gồm Photoshop.

Và còn vô số các ứng dụng khác dùng để thiết kế web như Wix,WordPress,...

**Kết luận chương 1**

Thiết kế web và triển khai hệ thống phần mềm là hai lĩnh vực có mối quan hệ mật thiết với nhau. Thiết kế web tạo ra giao diện người dùng trực quan và thu hút, trong khi triển khai hệ thống phần mềm cung cấp chức năng và tính năng cho website.

Thiết kế web ảnh hưởng trực tiếp đến trải nghiệm người dùng (UX) của website. Một website được thiết kế tốt sẽ dễ sử dụng, thu hút và mang lại trải nghiệm tích cực cho người dùng.

Triển khai hệ thống phần mềm quyết định chức năng và tính năng của website. Hệ thống phần mềm cần được thiết kế và triển khai cẩn thận để đáp ứng nhu cầu của người dùng và mục tiêu kinh doanh của website.

Thiết kế web và triển khai hệ thống phần mềm cần được phối hợp chặt chẽ với nhau để tạo ra website thành công. Hai lĩnh vực này cần làm việc cùng nhau để đảm bảo rằng website có giao diện đẹp mắt, chức năng tốt và đáp ứng nhu cầu của người dùng.

Từ chương 1, em giới thiệu tới bạn đọc 1 số khái niệm cơ bản như Web tĩnh , web động ,1 số thuật ngữ và 1 số công cụ thiết kế . Đây là cơ sở để các bạn có thể hiểu rõ hơn về các phần sau .

# **CHƯƠNG II: HTML & HTML5**

## **2.1. Tổng quan về HTML**

[**HTML**](https://topdev.vn/viec-lam-it/html-kt75)là viết tắt của cụm từ **Hypertext Markup Language**(tạm dịch là Ngôn ngữ đánh dấu [siêu văn bản](https://vi.wikipedia.org/wiki/Si%C3%AAu_v%C4%83n_b%E1%BA%A3n)). **HTML** được sử dụng để tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, titles, blockquotes… và **HTML** không phải là ngôn ngữ lập trình.

Một tài liệu HTML được hình thành bởi các phần tử HTML (HTML Elements) được quy định bằng các cặp thẻ (tag và attributes). Các cặp thẻ này được bao bọc bởi một dấu ngoặc ngọn (ví dụ <html>) và thường là sẽ được khai báo thành một cặp, bao gồm thẻ mở và thẻ đóng. Ví dụ, chúng ta có thể tạo một đoạn văn bằng cách đặt văn bản vào trong cặp tag mở và đóng văn bản ***<p>*** và ***</p> :***

<p>Đây là cách chúng ta thêm đoạn văn trong HTML.</p>

Nhưng một số thẻ đặc biệt lại không có thẻ đóng và dữ liệu được khai báo sẽ nằm trong các thuộc tính (ví dụ như thẻ <img>).

Cha đẻ của HTML là **Tim Berners-Lee**, cũng là người khai sinh ra World Wide Web và chủ tịch của **World Wide Web Consortium** (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet). Các thiết lập và cấu trúc HTML được vận hành và phát triển bởi World Wide Web Consortium (W3C). Chúng ta có thể kiểm tra tình trạng mới nhất của ngôn ngữ này bất kỳ lúc nào trên trang [W3C’s website](https://www.w3.org/).



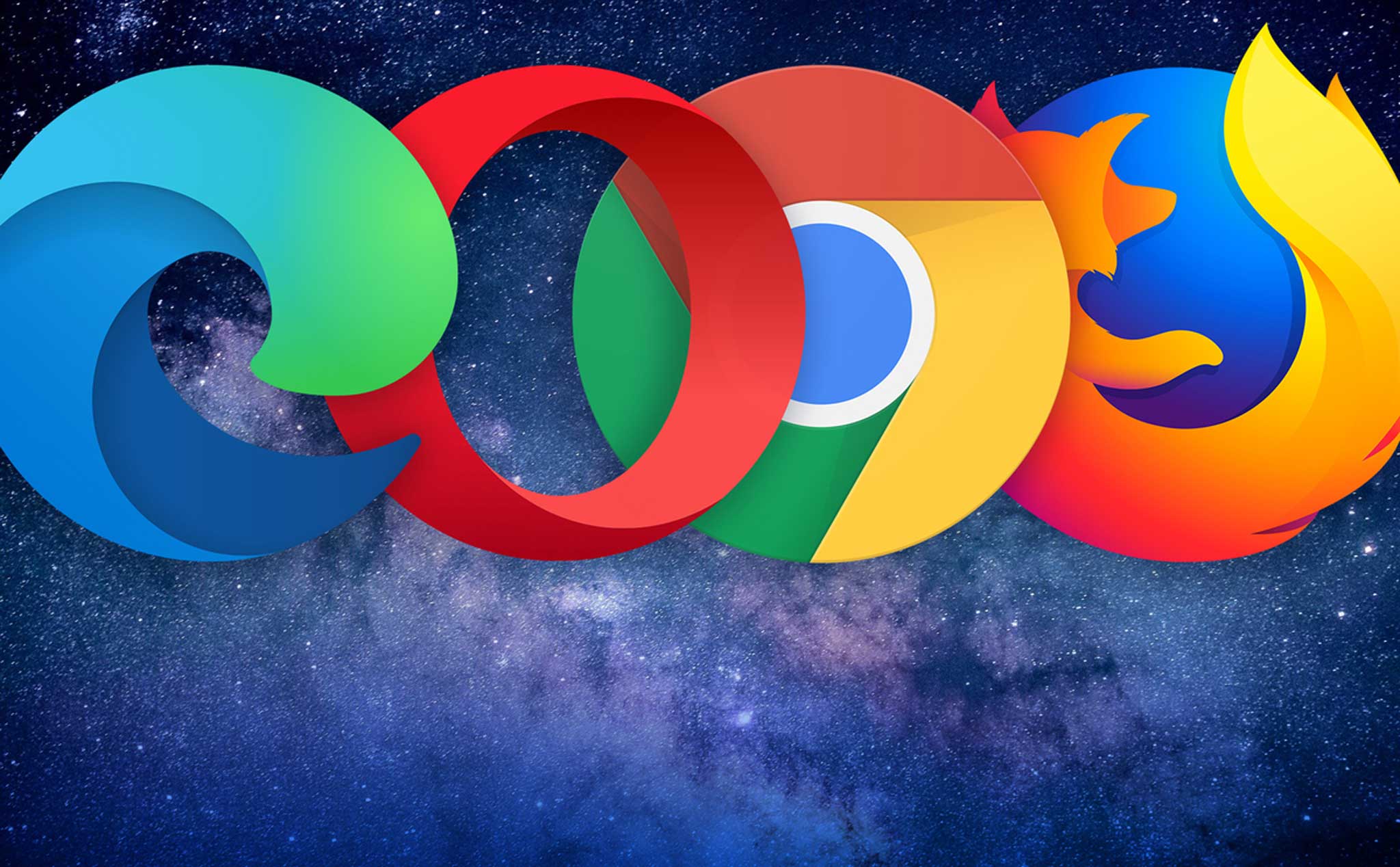
Hình 7.Trình duyệt web đọc file HTML và hiển thị chúng dưới dạng visual để người dùng có thể hiểu được

**HTML document** có đuôi file dạng .html hoặc htm. Chúng ta có thể xem chúng bằng các trình duyệt web hiện hành như Google Chrome, Firefox, Safari,… Nhiệm vụ của trình duyệt là đọc những file HTML này và “biến đổi” chúng thành một dạng nội dung visual trên Internet sao cho người dùng có thể xem và hiểu được chúng.

Thông thường, một Website sẽ có nhiều **HTML document** (ví dụ: trang chủ, trang blog, trang liên hệ,…) và mỗi trang con như vậy sẽ có một tệp HTML riêng. Mỗi tài liệu HTML bao gồm 1 bộ tag (hay còn gọi là element). Nó tạo ra một cấu trúc tương tự như cây thư mục với các heading, section, paragraph,… và một số khối nội dung khác. Hầu hết tất cả các HTML element đều có một tag mở và một tag đóng với cấu trúc <tag></tag>.

* Ưu điểm và Nhược điểm của HTML

HTML có khá nhiều những ưu điểm. Tuy nhiên nó vẫn tồn tại một số những điểm trừ. Hãy cùng Mắt Bão tìm hiểu chi tiết về ưu, nhược điểm của ngôn ngữ lập trình này.



Hình 8.HTML có thể hoạt động hiệu quả trên hầu hết các trình duyệt hiện nay

* **Ưu điểm của HTML là gì?**

**HTML**được sử dụng để tạo bố cục, cấu trúc trang web. Nó có một số ưu điểm sau:

* Có nhiều tài nguyên hỗ trợ với cộng đồng người dùng vô cùng lớn
* Có thể hoạt động mượt mà trên hầu hết mọi trình duyệt hiện nay
* Học HTML khá đơn giản
* Các markup sử dụng trong HTML thường ngắn gọn, có độ đồng nhất cao
* Sử dụng mã nguồn mở, hoàn toàn miễn phí
* HTML là chuẩn web được vận hành bởi W3C
* Dễ dàng để tích hợp với các loại ngôn ngữ backend (ví dụ như: PHP, Node.js,…)
* Nhược điểm của HTML là gì?

Bên cạnh ưu điểm, **HTML**cũng có các nhược điểm nhất định. Cụ thể như sau:

* Chỉ được áp dụng chủ yếu cho web tĩnh. Nếu muốn tạo các tính năng động, lập trình viên phải dùng thêm JavaScript hoặc ngôn ngữ backend của bên thứ 3 (ví dụ như: PHP)
* Mỗi trang **HTML** cần được tạo riêng biệt, ngay có khi có nhiều yếu tố trùng lặp như header, footer.
* Khó để kiểm soát cách đọc và hiển thị file HTML của trình duyệt (ví dụ, một số trình duyệt cũ không render được tag mới. Do đó, dù trong HTML document có sử dụng các tag này thì trình duyệt cũng không đọc được).
* Một vài trình duyệt còn chậm cập nhật để hỗ trợ tính năng mới của HTML

**HTML** là một loại ngôn ngữ đánh dấu siêu văn bản. Như tôi đã đề cập ở trên, nó giúp cấu thành các cấu trúc cơ bản của một Website, làm cho trang Web trở thành một hệ thống hoàn chỉnh. Cụ thể, ngôn ngữ đánh dấu siêu văn bản này giúp bố cục, chia khung sườn các thành phần trang Web. Đồng thời, nó còn hỗ trợ khai báo các File kỹ thuật số như nhạc, Video, hình ảnh,…

Nếu muốn Website có cấu trúc tốt, sử dụng nhiều loại yếu tố trong văn bản, chúng ta sẽ cần đến HTML. Theo quan điểm của tôi, lập trình viên có thể lựa chọn ngôn ngữ lập trình riêng cho Website tùy vào mục đích sử dụng. Nhưng HTML thực chất chứa những yếu tố cần thiết cho mọi thể loại Website. Trang Web của chúng ta sẽ cần đến ngôn ngữ HTML để hiển thị nội dung cho người truy cập. Điều này đúng dù trang của chúng ta xây dựng trên bất kỳ nền tảng nào, giao tiếp với bất kỳ ngôn ngữ lập trình nào để xử lý dữ liệu.

Trước khi bắt tay vào thiết kế, làm việc trên một trang Web thì nhà phát triển Web, lập trình viên đều phải học HTML như một loại ngôn ngữ cơ bản.

* Các đặc điểm của HTML

HTML có nhiều thẻ định dạng, do đó chúng ta có thể trình bày trang Web dễ dàng, hiệu quả với ngôn ngữ đánh dấu siêu văn bản này. Với tôi, nó là một ngôn ngữ đánh dấu dễ dàng và đơn giản để sử dụng. Chúng ta có thể sử dụng nó để thiết kế trang Web cùng với văn bản một cách linh hoạt.

Một đặc điểm theo tôi khá thú vị là HTML có thể liên kết đến các trang Web khác. Nhờ ngôn ngữ đánh dấu này, chúng ta có thể thêm các Video, hình ảnh, âm thanh vào để các Website hấp dẫn, đẹp mắt và dễ tương tác hơn.

Đặc biệt, HTML có thể hiển thị trên bất kỳ nền tảng nào khác như Linux, Windows, và Max vì nó là một nền tảng độc lập.

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

Để tạo được cấu trúc của HTML đầu tiên chúng ta phải:

* Lưu file vào thư chứa Web
* Đặt tên file, có phần mở rộng “.html”

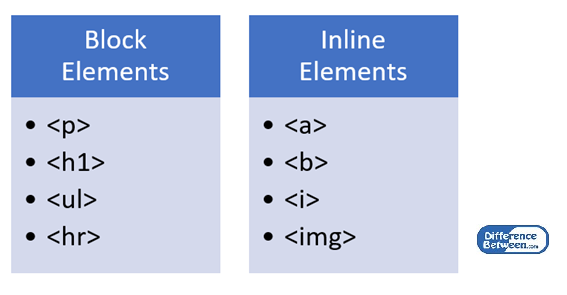
HTML là ngôn ngữ đánh dấu siêu văn bản, được sử dụng để tạo ra các trang web. Cấu trúc tổng quát của HTML gồm có:

* <!DOCTYPE html>: Khai báo chuẩn HTML5 cho trình duyệt.
* <html>: Thẻ bắt đầu và kết thúc của tài liệu HTML.
* <head>: Chứa các thông tin khai báo, như tiêu đề, mã hóa, liên kết CSS, JavaScript, v.v.
* <body>: Chứa nội dung hiển thị của trang web, bao gồm các thẻ HTML khác như <h1>, <p>, <div>, v.v.
* <title>: Thẻ con của <head>, chứa tiêu đề của trang web, hiển thị trên thanh tiêu đề của trình duyệt.
* <meta>: Thẻ con của <head>, chứa các thông tin siêu dữ liệu, như mã hóa, tác giả, mô tả, từ khóa, v.v.

Một ví dụ cơ bản của cấu trúc HTML như sau:



*Hình 9.Ví dụ minh họa về cấu trúc HTMl*



Hình 10. Có 2 loại tag được dùng phổ biến trong HTML là block-level tags và inline tags

tag được sử dụng chủ yếu là block-level tags và inline tags.

* Block-level tags

Đây là loại tag cấp cao, sẽ chiếm toàn bộ không gian của trang web và luôn bắt đầu dòng mới của trang. 3 block-level tags mà tất cả các trang HTML đầu cần có đó là: <html></html>, <head></head> và <body></body>. Ý nghĩa và chức năng của những tag này đã được đề cập ở trên nên trong phần này em sẽ không nhắc lại nữa.

* Inline tags

Loại tag này chỉ chiếm một phần nhỏ trong không gian Website và cũng không bắt đầu dòng mới của trang. Inline tags thường được dùng để định dạng, tạo bố cục cho nội dung bên trong của block-level tags.

HTML có nhiều dạng inline tags, ví dụ như: cặp tag *<strong></strong>*dùng để định dạng chữ in đậm, còn cặp tag *<em></em>*dùng để định dạng chữ in nghiêng.

Nếu muốn gắn hyperlinks vào trang, chúng ta có thể sử dụng cặp tag <a></a> với attributes href để xác định đường link cụ thể. Đoạn code lúc này có dạng:

<a href="https://example.com/">Click me!</a>

## **2.3. Các thẻ HTML thông dụng**

* *Các thẻ xử lý băn bản*

HTML cũng định nghĩa các**thẻ** đặc biệt để định dạng văn bản nhằm nhấn mạnh **ý nghĩa**nào đó.Các phần tử định dạng được thiết kế để hiển thị **các kiểu text**đặc biệt:

* Chữ đậm
* Chữ quan trọng
* Chữ nghiêng
* Chữ nhấn mạnh
* Chữ đánh dấu
* Chữ nhỏ
* Chữ gạch ngang
* Chữ gạch chân dưới
* Chữ viết thấp xuống
* Chữ viết cao lên

**HTML document** có đuôi file dạng .html hoặc htm. Chúng ta có thể xem chúng bằng các trình duyệt web hiện hành như Google Chrome, Firefox, Safari,… Nhiệm vụ của trình duyệt là đọc những file HTML này và “biến đổi” chúng thành một dạng nội dung visual trên Internet sao cho người dùng có thể xem và hiểu được chúng.

* *Định dạng chữ đậm và chữ quan trọng trong HTML*

Thẻ**<b>** định dạng chữ đậm trong HTML.

**Ví dụ:**

<p>This text is normal.</p>

<p><b>This text is bold</b>.</p>

Thẻ**<strong>** dùng để nhấn mạnh sự quan trọng của một đoạn văn bản, trình duyệt web thể hiện kết quả giống như thẻ <b>, nhưng về mặt ý nghĩa thì khác nhau. Thẻ <b> đơn giản là làm đậm phần văn bản, còn thẻ <strong> nhấn mạnh tầm quan trọng (mức ý nghĩa) của văn bản đó.

Ví dụ:

<p>This text is normal.</p>

<p><strong>This text is strong</strong>.</p>

* *Định dạng chữ nghiêng và chữ nhấn mạnh trong HTML*

Thẻ **<i>** trong HTML định dạng in nghiêng văn bản.

Ví dụ:

<p>This text is normal.</p>

<p><i>This text is italic</i>.</p>

Thẻ**<em>** dùng để nhấn mạnh đoạn văn bản. Trình duyệt web hiển thị kết quả giống như thẻ <i>. Nhưng về mặt ngữ nghĩa là khác nhau: thẻ <i> chỉ đơn thuần hiển thị chữ nghiêng, còn thẻ <em> ngoài việc hiển thị chữ nghiêng, nó còn nhấn mạnh ngữ nghĩa của đoạn văn bản mà chúng ta đánh dấu, ví dụ như một chương trình có thể đọc nội dung của một trang, nó đọc đến phần nội dung được đánh dấu bởi thẻ <em> thì nó sẽ nhấn mạnh đoạn đó, còn với thẻ <i> thì không.

Ví dụ:

<p>This text is normal.</p>

Trình duyệt hiển thị thẻ <strong> giống như thẻ <b>, và thẻ <em> giống như thẻ <i>.Tuy nhiên có sự khác nhau trong ý nghĩa của thẻ <b> định dạng chữ đậm và thẻ <i> định dạng chữ in nghiêng trong văn bản, nhưng thẻ <strong> và thẻ <em> nhấn mạnh tầm quan trọng về ngữ nghĩa trong văn bản.

* *Định dạng chữ đánh dấu trong HTML*

Thẻ**<mark>** dùng để định dạng**đánh dấu**hoặc**bôi màu** văn bản**.**

**Ví dụ:**

<h2>HTML <mark>Marked</mark> Formatting</h2>

* *Định dạng chữ gạch ngang trong HTML*

Thẻ **<del>**dùng để định dạng**chữ gạch ngang** trong văn bản.

Ví dụ:

<p>My favorite color is <del>blue</del> red.</p>

* *Định dạng chữ gạch ngang dưới trong HTML*

Thẻ**<ins>**trong HTML  định dạng chèn thêm văn bản.

### Ví dụ:

<p>My favorite <ins>color</ins> is red.</p>

* *Định dạng chữ viết thấp xuống trong HTML*

Thẻ **<sub>** trong HTML định dạng **chữ viết thấp xuống** so với chữ bình thường trong văn bản.

Ví dụ:

<p>This is <sub>subscripted</sub> text.</p>

* *Định dạng chữ viết cao lên trong HTML*

Thẻ **<sup>** định dạng chữ viết cao lên so với chữ bình thường trong văn bản.

### Ví dụ:

<p>This is <sup>superscripted</sup> text.</p>

* Các thẻ tạo bảng

**Tạo bảng trong HTML** là trường hợp rất hay gặp khi chúng ta muốn tạo bảng dữ liệu gồm cột và hàng. Với **thẻ Table trong HTML** chúng ta có thể tạo bảng để chèn thêm văn bản, hình ảnh hoặc 1 bảng khác vào mỗi hàng, mỗi cột.

Cách **tạo bảng trong HTML** như thế nào thì trong bài viết này chúng ta sẽ đi tìm hiểu về các [**thẻ tạo bảng trong HTML**](https://nguyenhung.net/the-table-trong-html-the-tao-bang-trong-html.html). Hướng dẫn dùng thẻ Table trong HTML – Các thẻ tạo bảng trong HTML. Để tạo bảng trong HTML chúng ta sẽ sử dụng các thẻ **table, tr, th, td, tbody, thead, tfoot.** Chúng ta sẽ tìm hiểu về các thẻ tạo bảng chi tiết ở dưới đây.

* Nhóm thẻ table -tr-th-td

Để tạo một bảng đơn giản với hàng và cột chúng ta viết đoạn code HTML như sau:

A computer code with many letters

Description automatically generated with medium confidence

*Hình 11. Ảnh mô tả hàng và cột*

Trong đó:

* Thẻ <table> là thẻ khai báo bảng trong HTML
* Thẻ <tr> là thẻ khai báo hàng trong HTML
* Thẻ <th> thường sử dụng trong hàng đầu tiên của mỗi cột để khai báo tiêu đề của bảng
* Thẻ <td> là thẻ khai báo cột trong HTML
* Thuộc tính *border=”1″* giúp khai báo đường viền cho bảng
* Nhóm thẻ thead-tbody-tfoot

Nhóm thẻ này giúp sắp xếp bố cục của bảng theo 3 phần lần lượt là phần đầu (thead), phần thân (tbody), phần cuối (tfoot).

Chúng ta chạy code lên sẽ thấy bố cục sắp xếp của bảng vẫn hiển thị như bình thường không có gì khác.Chúng ta chạy đoạn code đó lên thì thấy nó hiển thị cũng không khác gì so với code trên đúng không nào. Vậy chúng ta có thể thấy các thẻ <thead>, <tbody>, <tfoot> giúp định dạng bố cục của bảng theo quy luật là <thead> – <tbody> – <tfoot> cho dù chúng ta có đặt vị trí của nó ở đâu trong code.

* Thuộc tính colspan và rowspan trong bảng HTML

Thuộc tính Colspan: Dùng để gộp 2 hoặc nhiều cột gần nhau tính từ cột dặt thuộc tính colspan tiến sang bên phải.

Ví dụ:

A screen shot of a computer program

Description automatically generated

*Hình 13.Ví dụ về thuộc tính colspan*

**Thuộc tính Rowspan:**Dùng để **gộp 2 hoặc nhiều hàng gần nhau** tính từ hàng đặt thuộc tính **rowspan** xuống bên dưới.

Ví dụ:

A screen shot of a computer program

Description automatically generated

*Hình 14. Ví dụ về thuộc tính Rowspan*

* Thẻ liên kết

Thẻ liên kết là một thẻ HTML cơ bản bao gồm thẻ mở, thẻ đóng và các thuộc tính bên trong nó. Nó được sử dụng để liên kết, chuyển hướng đến một website, webpage, nội dung hay file nào đó,...

Cú pháp của thẻ liên kết trong HTML

<a href=“url” target=””title=””>Nội dung</a>

Trong đó:

* Href: Chỉ định trang, liên kết, đối tượng sẽ được chuyển đến khi ấn vào thẻ liên kết.
* Target:\_blank->Quy định về chuyển trang
* Title: Mô tả chi tiết về thẻ liên kết

Ví dụ:

**A computer screen with text

Description automatically generated**

*Hình 15. Ví dụ về thẻ liên kết*

* Các thẻ đa phương tiện

Các thẻ đa phương tiện là các thẻ có thể chèn hình ảnh, video, âm nhạc vào bên trong web.

* Thẻ Images

Để chèn ảnh vào HTML, chúng ta sẽ sử dụng thẻ <img> với các tham số bắt buộc, và thẻ này không có thẻ đóng. Ví dụ:

A screen shot of a computer program

Description automatically generated

*Hình 16. Ví dụ về thẻ Images*

Giải thích các thuộc tính:

* src: Đường dẫn đến tập tin hình ảnh.
* title: Tiêu đề của hình ảnh.
* alt: Tên định danh của hình ảnh.
* width: độ rộng của hình ảnh
* height: độ dài của hình ảnh
* br: xuống dòng (thẻ này không có thẻ đóng)
* Thẻ Audio

Chúng ta sẽ sử dụng một thẻ trong HTML để chèn audio nhanh chóng đó là cặp thẻ <audio></audio> như sau:

A computer screen shot of a black screen

Description automatically generated

*Hình 17. Ví dụ về thẻ Audio*

Nghĩa là trong cặp thẻ <video> </video> sẽ có một thẻ <source> với các thuộc tính nhằm khai báo đường dẫn tập tin video và loại tập tin. Các thuộc tính bên trong cặp thẻ này có vẻ không có gì là khó hiểu vì chúng ta đã sử dụng rồi. “autoplay controls” chính là bảng để chúng ta có thể điều khiển video chạy hoặc dừng hoặc tua nhanh, chậm tùy người sử dụng.

* Thẻ Video

Giống như thẻ <video>, để chèn âm thanh vào tài liệu HTML thì chúng ta có thể sử dụng thẻ <audio> và thẻ này cũng là HTML5.

A computer code on a black background

Description automatically generated

*Hình 18. Ví dụ về thẻ video*

* Chèn đối tượng kĩ thuật số với thẻ

Ngoài các thẻ đặc trưng cho từng loại tập tin ở trên thì chúng ta còn có một cách khác để chèn các đối tượng kỹ thuật số vào tài liệu HTML đó là dùng thẻ <object>, đây là một thẻ có thể giúp chúng ta chèn các loại đối tượng kỹ thuật số như Flash, Java, Audio, Video, PDF, ActiveX. Nhưng thông thường thì các loại mã nhúng của một số website cho phép sử dụng mã nhúng như Youtube họ sẽ dùng thẻ này để chúng ta chèn đối tượng vào web.

Nhúng tài liệu HTML vào web

Nếu chúng ta có nhu cầu nhúng thẳng một trang nào đó vào tài liệu HTML của chúng ta thì có thể sử dụng thẻ <iframe> như sau:

A black background with text

Description automatically generated

*Hình 19. Ví dụ về thẻ iframe*

Trong đó, thuộc tính  src  là đường dẫn của trang chúng ta muốn nhúng vào, width và height là kích thước của khung hiển thị (gọi là frame). Ngoài ra, có một cách nữa để chèn một liên kết và ép nó mở bằng frame trên website đó là sử dụng thuộc tính name trong thẻ <iframe> và thuộc tính target trong thẻ <a>. Trong đó, giá trị của thuộc tính target trong thẻ <a> phải trùng với giá trị trong thuộc tính name của thẻ <iframe>.

* Thẻ tạo khung

Thẻ iFrame HTML là những inline frames được sử dụng để chèn một HTML Document bên trong một HTML Document khác. IFrame được sử dụng rộng rãi trong quá trình thiết kế website để chèn nội dung trực tiếp vào trang web từ các nguồn khác như nội dung quảng cáo, các control panel điều khiển,…  
Hầu hết các nhà thiết kế web sử dụng iFrame HTML để thể hiện các ứng dụng tương tác trên trang các trang web. Điều này có thể thực hiện được bằng cách sử dụng JavaScript hoặc bằng cách sử dụng thuộc tính “target” trong HTML.

Mục đích chính của iFrame HTML là dùng để hiển thị một website bên trong một website khác thông qua việc sử dụng thẻ có tên gọi <iframe>. Nó hoạt động như một block hình chữ nhật được chèn trong website của chúng ta và browser sẽ hiển thị nội dung của website khác trong đó, bao gồm cả phần thanh trượt (scrollbar) để người dùng có thể tương tác với website được chèn vào

A screenshot of a computer

Description automatically generated

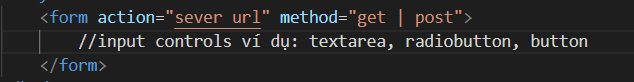
*Hình 20. Bảng mô tả <frame>*

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

Một HTML Form là một phần của trang web có chứa các điều khiển như các trường văn bản, các trường mật khẩu, các checkbox, các radio button, submit button, các menu...HTML Form là phương tiện cho người dùng nhập dữ liệu được gửi đến máy chủ để xử lý.

HTML Form được sử dụng khi chúng ta muốn thu thập một số dữ liệu của người truy cập trang web.Ví dụ: Nếu người dùng muốn mua một số mặt hàng trên online, họ phải điền vào form các thông tin như địa chỉ giao hàng và chi tiết thẻ tín dụng / thẻ ghi nợ để có thể gửi hàng đến địa chỉ đó.

* Khai báo HTML:

****

*Hình 21. Ví dụ về khai báo HTML*

Các thẻ HTML sử dụng trong Form

* <form>: Định nghĩa một HTML Form để nhập dữ liệu từ phía người dùng.
* <input>: Định nghĩa một điểu khiển một nhập dữ liệu vào thuộc tính type của nó.
* <textarea>: Định nghĩa ô nhập văn bản có thể nhập nhiều dòng.
* <label>: Định nghĩa nhãn cho các input.
* <fieldset>: Nhóm các phàn tử liên quan.
* <lengend>: Định nghĩa mô tả cho thẻ<fieldset>
* <select>: Định nghĩa một danh sách drop-down.
* <optgroup>: Định nghĩa một nhóm các tùy chọn lien quan thành một danh sách drop-down.
* <option>: Định nghĩa một option trong một danh sach drop-down.
* Trường HTML sử dụng trong Form

Thuộc tính type="text" của thẻ input tạo ra một textfield là trường nhập văn bản trong một dòng. Thuộc tính name là tùy chọn, nhưng nó là bắt buộc khi phía server là JSP, ASP, PHP...

A black background with text

Description automatically generated

*Hình 23. Ảnh minh họa thuộc tính type*

Đây được xem là một cách tốt hơn để hiển thị nhãn trong form. Vì nó tạo được trải nghiệm thân thiện hơn cho người dùng/trình duyệt/bộ phân tích mã nguồn.

Khi chúng ta click vào thẻ label, nó sẽ focus vào ô nhập text tương ứng. Để làm được như vậy, chúng ta cần có thuộc tính for trong thẻ label phải giống với thuộc tính id trong thẻ input.

* Trường HTML Password trong Form.

Password sẽ không được hiện thị trong trường password.

A computer screen with text

Description automatically generated

*Hình 25. Ảnh minh họa password trong HTML*

* Trường HTML 5 Email trong form

Trường email là một trường mới trong HTML 5. Nó sẽ kiểm tra tính hợp lệ của một địa chỉ email. Chúng ta phải sử dụng @ và dấu chấm (.) bên trong trường này.

**A computer screen shot of a program

Description automatically generated**

*Hình 26. Ảnh minh họa HTML 5*

* Ra dioButton trong form

Radio button được sử dụng để chọn chỉ một trong nhiều lựa chọn. Nó được sử dụng để lựu chọn giới tính, câu hỏi trắc nghiệm...

Sử dụng name giống nhau cho tất cả các radio button, thì chỉ duy nhất một radio button được chọn tại một thời điểm.

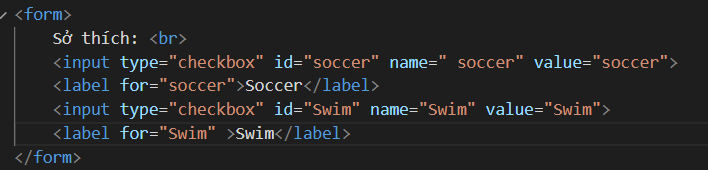
A screenshot of a computer code

Description automatically generated

*Hình 27. Ảnh minh họa Radio Button*

* Checkbox trong form

Checkbox được sử dụng để có thể chọn nhiều lựa chọn từ các checkbox đã cho.

****

*Hình 28. Ảnh minh họa checkbox*

## **2.5. Một số thẻ HTML đặc biệt**

Giả sử chúng ta có nhiều thông tin hoặc nội dung không muốn làm choáng ngợp người đọc ngay lập tức. Thẻ **<details>** và **<summary>** sẽ giúp chúng ta .Những thẻ này hoạt động cùng nhau để tạo ra một phần có thể mở rộng với tiêu đề hoặc phần tóm tắt mà người dùng web có thể click vào để hiển thị chúng. Mặc định, nội dung trong thành phần chi tiết sẽ không hiện, giữ cho trang của chúng ta luôn gọn gàng và có tổ chức.Khách truy cập có thể dễ dàng click vào phần tóm tắt để truy cập thông tin bị ẩn.

Với những thẻ này, chúng ta có thể ẩn các phần lớn của text, code hoặc thông tin khác, cung cấp trải nghiệm thân thiện người dùng trong khi vẫn duy trì thiết kế gọn gàng. Thậm chí chúng có thể giúp chúng ta cải thiện kỹ năng như một nhà thiết kế giao diện.

Thẻ**<mark>**cho phép chúng ta highlight các phần cụ thể của nội dung, làm chúng nổi bật. Khi dùng thành phần **<mark>,** các trình duyệt thường áp dùng màu nền vàng cho text bên trong nhằm thu hút sự chú ý của người đọc.

Tính năng này đặc biệt hữu ích khi chúng ta muốn nhấn mạnh từ khóa, các cụm từ quan trọng hoặc kết quả trên trang web.

Ví dụ, nếu web của chúng ta có chức năng tìm kiếm, chúng ta có thể dùng thẻ <mark> để highlight truy vấn tìm kiếm phù hợp trong kết quả, giúp người dùng tìm thấy thông tin liên quan dễ dàng hơn.

Nhập thẻ **<s>**. Nó là chữ viết tắt của strikethrough (gạch ngang) và hiển thị bất kỳ nội dung bên trong thành phần này với một đường thẳng ở giữa.

A black background with white text

Description automatically generated

*Hình 31. Hình minh họa về thẻ <s>*

Ở ví dụ này, text **out-of-stock**sẽ hiện với một đường kẻ đi qua nó, cho biết rằng trạng thái hàng trong kho đã thay đổi.

Khi muốn thêm ngữ nghĩa cho ngày tháng và thời gian trong nội dung của chúng ta, thẻ <time> sẽ rất hữu ích.

Dùng thuộc tính **datetime**, chúng ta có thể xác định phiên bản ngày tháng hoặc thời gian mà máy có thể đọc, giúp trình duyệt, công cụ tìm kiếm và trình đọc màn hình hiểu nội dung.

Bằng cách dùng thẻ **<time>,** chúng ta cho nội dung có cấu trúc hơn và giúp nhiều người dễ truy cập nó hơn, bao gồm cả người khuyết tật dùng trình đọc màn hình.

Quản lý văn bản ở nhiều ngôn ngữ trên trang web đôi khi thật sự là một thử thách, nhất là trong trường hợp mỗi phần lại yêu cầu định dạng khác nhau.Thẻ **<bdi>** sẽ “cứu” chúng ta trong trường hợp này bằng cách cô lập một phần văn bản mà trình duyệt sẽ xử lý khác do những yêu cầu khác biệt về ngôn ngữ.

A black background with white text

Description automatically generated

*Hình 33. Hình minh họa thẻ<bdi>*

Ở ví dụ này, thành phần **<bdi>**bao gồm số **5,000**. Điều này đảm bảo nếu text xung quanh ở một ngôn ngữ khác hay yêu cầu định dạng khác, nó sẽ không ảnh hưởng tới số.

## **2.6. HTML5**

HTML 5 là phiên bản lớn tiếp theo của HTML thay thế cho HTML 4.01, XHTML 1.0, và XHTML 1.1. HTML5 là một chuẩn để cấu trúc và hiện diện nội dung trên www.

Chuẩn mới này kết hợp các tính năng như Video Playback và drag-and-drop mà trước đây đã phụ thuộc vào plug-ins trình duyệt thứ 3 như Adobe Flash, Microsoft Silverlight, và Google Gears.

HTML5 là kết quả của sự hợp tác giữa W3C và WHATWG (viết tắt của Web Hypertext Application Technology Working Group).

* Cú pháp HTML5

Ngôn ngữ HTML5 có một cú pháp "custom" mà tương thích với tài liệu HTML4 và XHTML1 đã công bố trên Web, nhưng nó không tương thích với các tính năng riêng SGML của HTML4.

HTML5 không có quy tắc cú pháp chung như XHTML, như chúng ta cần các tên thẻ là chữ thường, trích dẫn các thuộc tính của nó, một thuộc tính phải có một giá trị và để đóng tất cả các phần tử trống.

Nhưng HTML5 có nhiều sự linh động hơn và sẽ hỗ trợ những điều sau:

* Các tên thẻ chữ hoa.
* Trích dẫn là tùy ý cho các thuộc tính.
* Giá trị của thuộc tính là tùy ý.
* Đóng các phần tử trống là tùy ý.
* Khai báo DOCTYPE trong HTML 5

Khai báo DOCTYPE trong HTML 5 là một phần quan trọng trong việc khai báo tệp HTML. Nó có chức năng thông báo cho trình duyệt về loại tệp và phiên bản HTML được sử dụng trong tệp đó. Cú pháp khai báo thường được đặt ngay trước thẻ và sau thẻ . Đối với HTML5, chúng ta chỉ cần khai báo để đánh dấu tệp HTML hiện đại. Việc khai báo đảm bảo rằng trình duyệt hiểu và hiển thị nội dung HTML một cách chính xác và nhất quán.Khai báo không phải là một thẻ HTML. Nó là một “thông tin” giúp trình duyệt biết loại tài liệu mà nó mong đợi.Trong HTML 5, cú pháp khai báo rất đơn giản.

* Thẻ ngữ nghĩa

Có nhiều web site chứa mã HTML dạng: <div id="nav"> <div class="header"> <div id="footer"> để chỉ ra thanh điều hướng,header, và footer.HTML5 mang đến các phần tử ngữ nghĩa mới để xác định các phần khác nhau của một trang web:

* <article>
* <aside>
* <details>
* <figcaption>
* <figure>
* <footer>
* <header>
* <main>
* <mark>
* <nav>
* <section>
* <summary>
* <time>
* **Phần tử <section> HTML5**

Phần tử <section> định nghĩa một phần của tài liệu.Theo tài liệu về HTML5 của W3C: “Một section là một nhóm chủ đề nội dung, thường có một tiêu đề.”Một trang chủ thường được chia thành các phần cho giới thiệu, nội dung, và thông tin liên lạc.

Ví dụ

A black background with white text

Description automatically generated

*Hình 35. Ảnh minh họa thẻ <selection>*

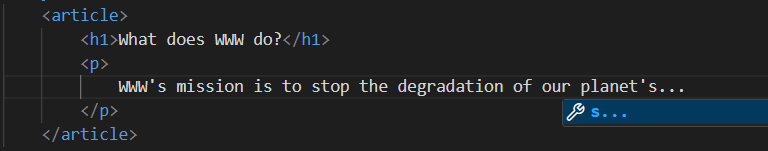
* **Phần tử <article> HTML5**

Phần tử <article> (bài viết) chỉ ra nội dung độc lập. Một bài viết nên có nội dung riêng và có thể đọc độc lập với phần còn lại của trang web.

Ví dụ nơi phần tử <article> có thể được dùng:

* Forum post
* Blog post
* Bài báo

Ví dụ



*Hình 36. Hình minh họa <article>*

* **Lồng phần tử <article> trong <section> hoặc ngược lại.**

Phần tử <article> chỉ ra phần tử có nội dung độc lập.Phần tử <section> xác định một phần của tài liệu.Chúng ta không thể dùng định nghĩa để quyết định cách lồng những phần tử này.Vậy nên, trên internet, chúng ta sẽ thấy những trang với phần tử <section> chứa phần tử <article>, và phần tử <article> chứa phần tử <section>Ví dụ: Trong một bài báo, bài viết(<article>) trong mục(<section>) thể thao, có thể có nhiều mục(<section>) trong bài(<article>).

* **Phần tử <header> HTML5**

Phần tử <header> chỉ ra một tiêu đề cho trang hoặc một section.Phần tử <header> nên được dùng như bộ chứa cho nội dung mở đầu.Có thể có nhiều phần tử <header> trong một trang.

Ví dụ sau xác định một tiêu đề cho một bài viết

A screenshot of a computer screen

Description automatically generated

*Hình 37. Ảnh minh họa <header>*

* **Phần tử <footer> HTML5**

Phần tử <footer> chỉ ra phần chân trang cho tài liệu hoặc một section. Phần tử <footer> nên chứa thông tin về phần tử chứa nó. Một footer thường bao gồm tác giả của tài liệu, thông tin bản quyền, liên kết tới điều khoản sử dụng, thông tin liên lạc,v.v...Có thể có nhiều phần tử <footer> trong một trang.

* **Phần tử <nav> HTML5**

Phần tử <nav> xác định tập hợp các liên kết điều hướng. Lưu ý: Không phải mọi liên kết trong tài liều đều nên đặt bên trong phần tử <nav>. Phần tử <nav> thường dành cho các khối lớn hoặc các liên kết điều hướng.

Ví dụ

A computer code with text

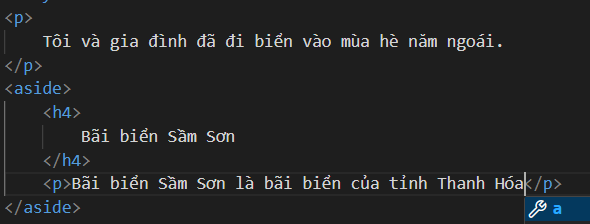
Description automatically generated with medium confidence

*Hình 39. Ảnh minh họa <nav>*

* **Phần tử <aside> HTML5**

Phần tử <aside> xác định một số nội dung bên cạnh nội dung chính (gần giống thanh bên).Nội dung <aside> nên liên quan tới các nội dung xung quanh.

Ví dụ



*Hình 40. Ảnh minh họa <aside>*

* **Phần tử <figure> và <figcaption> HTML5**

Mục đích của figure caption là để thêm một diễn tả một hình ảnh. Trong HTML5, một hình ảnh và tiêu đề có thể được nhóm chung trong một phần tử <figure>.

Với HTML4, các lập trình viên đã sử dụng tên id/class của chúng để style các phần tử: header, top, bottom, footer, menu, navigation, main, container, content, article, sidebar, topnav, v.v...

Điều này giúp các bộ tìm kiếm có khả năng xác định chính xác nội dung trang web. Với các phần tử mới của HTML5 (<header> <footer> <nav> <section> <article>), việc này trở nên dễ dàng hơn. Theo W3C, một Web ngữ nghĩa: “Cho phép dữ liệu được chia sẻ và tái sử dụng giữa các ứng dụng, doanh nghiệp, và cộng đồng.”

Webform là giải pháp chạy trên nền tảng web chỉ cần một yếu tố duy nhất để thực hiện đó là có kết nối internet. Vì vậy, nhắc đến webform thì bất cứ ai trong chúng ta cũng đều cảm thấy rất quen thuộc bởi chúng ta được tiếp xúc với nó hàng ngày.

Khi chúng ta đăng nhập vào tài khoản gmail,facebook, nhập thông tin mua hàng online, đăng ký form hay bất kì thao tác nào trên internet,… Tất cả chúng đều thông qua Webform.

Webform là mô hình lập trình rất hiện đại, đơn giản, rõ ràng, linh động, hoạt động độc lập,… Là một trong những công nghệ độc đáo của Microsoft.

* Ưu điểm về Webform:
* Tính bảo mật tuyệt đối, tốc độ xử lý dữ liệu nhanh, bảo trì linh hoạt…
* Người dùng không cần cài phần mềm về máy tính cá nhân mà chỉ cần được cấp tài khoản sử dụng, họ có thể đăng nhập vào phần mềm mọi lúc, mọi nơi ngay cả trên các thiết bị như máy tính bảng, Smartphone,…
* Phần mềm được cài đặt trực tiếp trên máy chủ của đơn vị mua phần mềm hoặc máy cloudserver của đơn vị cung cấp phần mềm … nên đảm bảo hoat động ổn định ngay cả khi không có kết nối Internet.
* Việc quản lý sẽ đơn giản dù chúng ta có nhiều đơn vị hoạt động ở các nơi cách xa nhau về mặt địa lí
* Thao tác với giải pháp phần mềm trên nền webform đơn giản, nhanh chóng, tiện lợi.

**Kết luận chương 2**

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu siêu văn bản, đóng vai trò như bộ khung cơ bản để xây dựng trang web. Nó sử dụng các thẻ để định nghĩa cấu trúc và nội dung của trang, bao gồm văn bản, hình ảnh, video, liên kết,...

HTML5 là phiên bản mới nhất của HTML, được phát hành vào năm 2014. Nó mang đến nhiều cải tiến và tính năng mới so với các phiên bản trước, giúp cho việc phát triển web trở nên dễ dàng và hiệu quả hơn.

Kết thúc chương II, em muốn đưa các chúng ta đọc đến với từng chi tiết cụ thể trong HTML và HTML 5. Các chúng ta đọc có thể nắm rõ các cấu trúc tổng quát, các thẻ HTML thông dụng, các thẻ HTML đặc biệt, cú pháp HTML 5, các thẻ ngữ nghĩa cũng như hiểu sâu hơn về webform trong HTML.

**CHƯƠNG III: CSS VÀ CSS3 (CASCADING STYLE SHEETS)**

**3.1. CSS là gì?**

**CSS** là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để **tìm và định dạng** lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu ([HTML](https://topdev.vn/blog/html-la-gi/)). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Chúng ta có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc…

CSS được phát triển bởi **W3C** ([World Wide Web Consortium](https://www.w3.org/)) vào năm 1996, vì HTML không được thiết kế để gắn tag để giúp định dạng trang web.

Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.

**3.2. Cú pháp CSS**

CSS được tạo thành từ các Style Rule. Các rule này được thông dịch bởi trình duyệt và sau đó được áp dụng tới các phần tử tương ứng trong tài liệu của chúng ta . Style Rule bao gồm ba phần:

* Selector: (hiểu nôm na là bộ chọn phần tử) Một selector là một thẻ HTML mà tại đó một chúng ta áp dụng một style cho nó. Đó có thể là bất kỳ một thẻ HTML nào, như thẻ div, thẻ span, thẻ p, …
* Property: (thuộc tính) Là property của một kiểu thuộc tính của thẻ HTML. Nói một cách đơn giản thì tất cả các thuộc tính trong HTML được chuyển đổi thành các CSS property. Đó có thể là *color*, *border*....
* Value: Là các giá trị được gán cho các property. Ví dụ, *color* có thể có các giá trị là *red* *#F1F1F1* etc.

Trong ví dụ trên, selector trỏ tới phần tử HTML là h1. Đây là phần tử mà mình muốn tạo style cho nó.Khối khai báo Declaration có thể bao gồm một hoặc nhiều khối và được đặt phân biệt nhau bởi dấu chấm phảy.

Mỗi khối khai báo này bao gồm một tên của CSS property (trong ví dụ là color và font-size) và một giá trị (trong ví dụ là blue và 12px) được gán cho property đó.

Chúng ta cần chú ý một điều rằng, một khai báo CSS luôn luôn kết thúc với một dấu chấm phảy và phần khai báo này sẽ được bao quanh bởi cặp dấu ngoặc ôm.

Trong ví dụ tiếp theo, tất cả các phần tử sẽ được căn chỉnh theo lề trái và có màu là màu đỏ:

A screenshot of a computer

Description automatically generated

*Hình 44. Ảnh minh họa*

**3.3. Áp dụng CSS vào trang HTML**

Có ba cách để sử dụng CSS: Inline, Internal, External

* Inline CSS

Ta đặt thuộc tính style vào thẻ mở của phần tử HTML, giá trị của thuộc tính style là các cặp thuộc tính định dạng CSS

* Internal CSS

Ta đặt các cặp thuộc tính định dạng CSS vào bên trong cặp thẻ <style type="text/css"></style>.

Cặp thẻ <style type="text/css"></style> thì được đặt bên trong cặp thẻ <head></head>

Ví dụ:

A screen shot of a computer

Description automatically generated

*Hình* *45. Ảnh minh họa <style>*

External CSS

Với Internal CSS:

* Ta đặt các thuộc tính định dạng vào bên trong *cặp thẻ <style type="text/css"></style>.*
* Rồi cặp thẻ <style type="text/css"></style> thì đặt bên trong cặp thẻ <head></head>.

Với External CSS:

* Ta đặt các thuộc tính định dạng vào bên trong *tập tin CSS.*
* Trong cặp thẻ <head></head> của tập tin HTML, ta dùng thẻ <link rel="stylesheet" type="text/css" href="đường dẫn đến tập tin CSS"> để nhúng tập tin CSS vào trang web.

Ví dụ:

Chúng ta có một tập tin dinhdang.css (cùng cấp với tập tin HTML) có nội dung là:

A screen shot of a computer

Description automatically generated

*Hình 46. Ảnh mô tả CSS*

Một số thuộc tính định dạng CSS quan trọng:

* color: Xác định kiểu chữ cho phần tử.

Giá trị của thuộc tính color có thể xác định theo:

* Tên màu (ví dụ: green, blue, red, yellow, pink, black, white, …)
* Giá trị Hex (ví dụ: #00FF00, #99FF66, #66CC99, …)
* Giá trị RGB (ví dụ: rgb(255,0,0), rgb(0,120,0), …)
* background-color: Thiết lập màu nền cho phần tử.

Giá trị của thuộc tính background-color cũng có thể xác định theo tên màu, giá trị Hex hoặc giá trị RGB giống như thuộc tính color.

* font-size: Điều chỉnh kích cỡ chữ cho phần tử. Giá trị của thuộc tính font-size có thể xác định bằng px hoặc em (1em = 16px).
* font-family: Xác định kiểu chữ cho phần tử.

Có một số kiểu chữ phổ biến như Times New Roman, Arial Black, Lucida Sans Unicode, và Palatino Linotype.

* text-align: Canh lề cho nội dung của phần tử.

Thuộc tính này có ba giá trị:

* left: Canh chữ nội dung nằm bên trái.
* center: Canh chữ nội dung ở giữa.
* right: Canh chữ nội dung nằm bên phải.

border: Tạo đường viền bao xung quanh phần tử.

Giá trị của thuộc tính border có 3 tham số:

* Tham số thứ nhất là độ dày của đường viền.
* Tham số thứ hai là kiểu đường viền (ví dụ: groove, ridge, inset, outset, none, hidden, dotted, dashed, solid, double, …).
* Tham số thứ ba là màu của đường viền.
* width, height: Thiết lập chiều rộng và chiều cao của phần tử.

**3.4. Selector**

***3.4.1. Universal selector***

Nếu chúng ta muốn áp dụng một style nào đó cho tất cả các thẻ HTML trong tài liệu, chúng ta có thể sử dụng cú pháp giống như dưới đây.

A screen shot of a computer

Description automatically generated

*Hình 48. Ảnh minh họa*

***3.4.2. Type selector***

Cũng giống như trong ví dụ trên, ví dụ sau sẽ áp dụng màu cho thẻ h1.

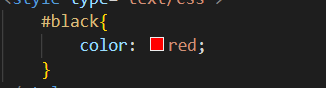
*A black screen with white text

Description automatically generated*

*Hình 49. Ảnh minh họa áp dụng màu cho thẻ h1*

***3.4.3. Identity selector***

ID Selector sử dụng thuộc tính id của một phần tử HTML để chọn một phần tử cụ thể. Trong một webpage, mỗi phần tử nên có một id duy nhất, vì thế ID Selector được sử dụng để lựa chọn một phần tử duy nhất.



*Hình 50.Ảnh minh họa*

Để lựa chọn một phần tử với một id cụ thể, chúng ta viết một ký tự # (đọc là ký tự hash) ở trước id của phần tử đó.

Ví dụ dưới đây sẽ áp dụng style rule cho phần tử có id= "para1";

A black background with white text

Description automatically generated

*Hình 51. Ảnh minh họa #paral*

Style Rule này sẽ chỉ áp dụng cho các phần tử div có id là para1.Ngoài ra, nếu chúng ta muốn áp dụng style rule cho một phần tử mà là con của (hay nằm trong) phần tử khác, chúng ta có thể sử dụng:

A black screen with white text

Description automatically generated

Hình 52. Ảnh minh họa

Style Rule này sẽ áp dụng cho phần tử div là phần tử con của phần tử có id là para1.

***3.4.4. Class selector***

Tương tự như ID Selector, chúng ta cũng có thể áp dụng các Style Rule cho các phần tử dựa trên thuộc tính class của phần tử đó.Để chọn các phần tử với một class cụ thể, chúng ta viết một dấu chấm (.) ở trước tên class của phần tử đó.

Ví dụ dưới đây sẽ định nghĩa Style Rule cho tất cả phần tử HTML với class= "center":

*A black background with white text

Description automatically generated*

*Hình 52. Ảnh minh họa center*

Với trường hợp có nhiều phần tử với cùng một class, nếu chúng ta chỉ muốn áp dụng style rule cho một phần tử nào đó, chúng ta viết tên phần tử đó trước cú pháp trên, như sau:

A screen shot of a computer

Description automatically generated

*Hình 53. Ảnh minh họa*

Các phần tử HTML có thể tham chiếu tới nhiều class. Do đó, chúng ta có thể áp dụng nhiều Style Rule cho một phần tử (chẳng hạn div) với các class = "center" (như trên) và class = "black" (để xác định màu đen chẳng hạn), như sau:

**A black screen with white text

Description automatically generated**

*Hình 54. Ảnh minh họa center black*

***3.4.5. Descendant selector***

Giả sử chúng ta chỉ muốn áp dụng style này cho một phần tử mà là con của phần tử khác (phần tử con là phần tử mà nằm bên trong phần tử khác). Ví dụ sau sẽ áp dụng style cho phần tử chỉ khi nào nó nằm bên trong thẻ <div>:

**A screenshot of a computer

Description automatically generated**

*Hình 55. Ảnh minh họa descendant seclector*

***3.4.6. Child selector***

Phần trên chúng ta đã tìm hiểu về Descendant Selector. Ngoài ra, còn có một kiểu Selector nữa mà khá giống với kiểu này.

A screenshot of a computer

Description automatically generated

*Hình 56. Ví dụ về child selector*

Style Rule này sẽ được áp dụng cho tất cả các phần tử p mà trực tiếp là con của phần tử body. Các phần tử p khác nằm bên trong các phần tử như div hoặc td sẽ không bị tác động.

***3.4.7. Adjacent sibling selector***

Adjacent sibling selector là một phần của CSS (Cascading Style Sheets) được sử dụng để chọn một phần tử nằm ngay sau một phần tử cụ thể khác. [Điều kiện là các phần tử này phải có cùng một phần tử cha (parent element) và liền kề, tức là phần tử thứ hai nằm ngay sau phần tử thứ nhất](https://timoday.edu.vn/bai-25-vung-chon-trong-css/). Ví dụ, để chọn phần tử <p> đầu tiên được đặt ngay sau phần tử <div>, chúng ta có thể sử dụng cú pháp sau:

**CSS**

div + p

{

background-color: yellow;

}

[Nhớ rằng Adjacent sibling selector chỉ áp dụng cho các phần tử liền kề, không có phần tử nào ở giữa chúng](https://timoday.edu.vn/bai-25-vung-chon-trong-css/). Nó giúp chúng ta tạo các quy tắc CSS dựa trên vị trí tương đối của các phần tử trong cùng một cấu trúc HTML.

***3.4.8. Attribute selector***

Chúng ta cũng có thể áp dụng style cho các phần tử HTML có các thuộc tính cụ thể nào đó. Ví dụ sau sẽ áp dụng style rule cho tất cả phần tử input nào mà có thuộc tính type= "text".

A black screen with white text and blue squares

Description automatically generated

*Hình 57. Ví dụ về attribute selector*

Phương thức này đem lại lợi thế: phần tử <input type = "submit" /> sẽ không bị tác động bởi style rule trên, và color sẽ chỉ được áp dụng cho trường text.

Sau đây là ví dụ về một số rule áp dụng cho Attribute Selector:

* p[lang] : lựa chọn tất cả phần tử p mà có thuộc tính *lang*.
* p[lang="fr"] : lựa chọn tất cả phần tử p mà có thuộc tính *lang* có giá trị chính xác là fr.
* p[lang~="fr"]: lựa chọn tất cả phần tử p mà có thuộc tính *lang* có giá trị có chứa từ fr.
* p[lang|="en"] : lựa chọn tất cả phần tử p mà có thuộc tính *lang* có giá trị chính xác là en hoặc bắt đầu với en.

***3.4.9. Pseudo class selector***

Pseudo class selector trong CSS là một phần của ngôn ngữ Cascading Style Sheets (CSS) được sử dụng để xác định một trạng thái đặc biệt của một phần tử cụ thể. Các trạng thái này thường liên quan đến tương tác của người dùng với phần tử đó. Dưới đây là một số điểm quan trọng về pseudo class selector:

Pseudo class được dùng để xác định trạng thái đặc biệt của một phần tử.Ví dụ phổ biến là :hover (khi rê chuột qua phần tử), :active (khi phần tử đang được nhấn), và :focus (khi phần tử đang được chọn).

Để viết một pseudo class selector, chúng ta sử dụng cú pháp sau:

selector:pseudo-class

{

property: value;

}

Danh sách một số pseudo classes phổ biến:

* :link: Chọn tất cả các liên kết chưa được click.
* :hover: Thay đổi trạng thái khi rê chuột qua phần tử.
* :active: Thay đổi trạng thái của phần tử khi click vào nó.
* :visited: Chọn tất cả các liên kết đã truy cập.
* :first-child: Chọn tất cả các phần tử là con đầu tiên của phần tử cha.
* :last-child: Chọn tất cả các phần tử là con cuối cùng của phần tử cha.

Pseudo class selector giúp chúng ta tạo các quy tắc CSS dựa trên tương tác của người dùng với phần tử trong cùng một cấu trúc HTML.

***3.4.10. Group selector***

Có một tình huống xảy ra là giả sử chúng ta muốn áp dụng cùng một style rule cho nhiều phần tử khác nhau. Nếu theo những gì đã học ở trên, chúng ta phải viết lần lượt từng style rule này cho từng phần tử. Như vậy cũng được, nhưng hãy tưởng tượng nếu số lượng phần tử là lớn và việc lặp đi lặp lại này có thể sẽ tốn khá nhiều thời gian. Có cách nào tốt hơn không. Câu trả lời là có.

Chúng ta hãy nhóm tất cả các phần tử này lại, và sau đó áp dụng style rule chung mà chúng ta muốn áp dụng cho chúng. Đó cũng chính là ý nghĩa của Grouping Selector trong CSS. Chúng ta theo dõi ví dụ sau:

A screen shot of a computer program

Description automatically generated

*Hình 58. Ví dụ về group selector*

Style rule này sẽ áp dụng cho các phần tử h1, h2, và h3. Chúng ta cần chú ý là giữa các phần tử này phải có dấu phảy (,) nhé. Đó là cú pháp chung. Sử dụng Grouping Selector sẽ giúp chúng ta tiết kiệm được thời gian và tối thiểu hóa được lượng code mà chúng ta cần viết.

Tương tự, nếu chúng ta muốn áp dụng cho nhiều class, thì chúng ta có thể sử dụng nhiều Class Selector trong CSS như ví dụ dưới đây:

**A black screen with white text

Description automatically generated**

*Hình 59. Ví dụ về style rule*

**3.5. Đơn vị đo lường CSS**

Trong CSS, có nhiều đơn vị đo lường để biểu thị chiều dài. Nhiều thuộc tính CSS sử dụng giá trị “độ dài”, chẳng hạn như chiều rộng, lề, độ dày chữ, v.v. Độ dài được biểu thị bằng một số kèm theo đơn vị, ví dụ: 10px, 2em, v.v. Dưới đây là một số đơn vị đo lường phổ biến trong CSS:

* Đơn vị độ dài tuyệt đối (Absolute Lengths):

Các đơn vị độ dài tuyệt đối có giá trị cố định và sẽ hiển thị với kích thước chính xác.Tuy nhiên, chúng không được khuyến nghị sử dụng trên màn hình vì kích thước màn hình thay đổi quá nhiều.

Các đơn vị tuyệt đối bao gồm:

* cm: centimet (ví dụ: 1cm).
* mm: millimet (ví dụ: 10mm).
* in: inches (1 inch = 96px = 2.54cm).
* px: pixels (1px = 1/96 của 1 inch).
* pt: points (1pt = 1/72 của 1 inch).
* pc: picas (1pc = 12pt).
* Đơn vị độ dài tương đối (Relative Lengths):

Các đơn vị độ dài tương đối xác định kích thước liên quan đến một thuộc tính độ dài khác.Chúng tỷ lệ tốt hơn giữa các phương tiện hiển thị khác nhau.

Các đơn vị tương đối bao gồm:

* em: Tương đối với kích thước font của phần tử (2em tương đương với 2 lần kích thước font hiện tại).
* rem: Tương đối với kích thước font của phần tử gốc.
* vw: Tương đối với 1% chiều rộng của viewport (cửa sổ trình duyệt).
* vh: Tương đối với 1% chiều cao của viewport.
* vmin: Tương đối với 1% chiều nhỏ hơn của viewport.
* vmax: Tương đối với 1% chiều lớn hơn của viewport.

Nhớ rằng em và rem thường được sử dụng để tạo giao diện linh hoạt và có thể tỷ lệ hoàn hảo.

**3.6. Kế thừa thuộc tính**

Trong CSS, kế thừa (inheritance) quyết định điều gì sẽ xảy ra khi không có giá trị được chỉ định cho một thuộc tính trên một phần tử. Các thuộc tính CSS có thể được phân loại thành hai loại:

* Thuộc tính kế thừa (Inherited properties):

Các thuộc tính kế thừa mặc định được thiết lập bằng giá trị tính toán của phần tử cha.Ví dụ về thuộc tính kế thừa là thuộc tính color (màu chữ). Khi không có giá trị cho thuộc tính này được chỉ định trên một phần tử con, phần tử con sẽ kế thừa giá trị đã tính toán từ phần tử cha.

Ví dụ:

CSS

p {

color: green;

}

<p>This paragraph has <em>emphasized text</em> in it.</p>

Các từ “emphasized text” sẽ hiển thị màu xanh (được kế thừa từ phần tử <p>).

* Thuộc tính không kế thừa (Non-inherited properties):

Các thuộc tính không kế thừa mặc định được thiết lập bằng giá trị khởi tạo của thuộc tính đó.

Ví dụ về thuộc tính không kế thừa là thuộc tính border. Khi không có giá trị cho thuộc tính này được chỉ định trên một phần tử con, phần tử con sẽ nhận giá trị khởi tạo của thuộc tính.

Ví dụ:

CSS

p {

border: medium solid;

}

Các từ “emphasized text” sẽ không có đường viền bổ sung (vì giá trị khởi tạo của border-style là none).

Từ khóa inherit cho phép tác giả chỉ định rõ việc kế thừa. Nó hoạt động cho cả thuộc tính kế thừa và không kế thừa. Chúng ta cũng có thể kiểm soát việc kế thừa cho tất cả các thuộc tính cùng một lúc bằng cách sử dụng thuộc tính ngắn gọn all, áp dụng giá trị của nó cho tất cả các thuộc tính.

Ví dụ:

CSS

p {

all: revert;

font-size: 200%;

font-weight: bold;

}

Đoạn mã trên sẽ đặt lại kiểu chữ của đoạn văn thành kiểu mặc định của trình duyệt, trừ khi đã có một bộ kiểu người dùng, trong trường hợp đó sẽ sử dụng bộ kiểu đó. Sau đó, nó tăng kích thước font lên gấp đôi và áp dụng kiểu chữ đậm.

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

* Thuộc tính định dạng font chữ, văn bản (font,text):
* font-family: Xác định kiểu font chữ.
* font-size: Đặt kích thước font chữ.
* font-weight: Xác định độ đậm của font chữ.
* color: Định dạng màu chữ.
* text-align: Căn chỉnh văn bản (trái, phải, giữa).
* Thuộc tính định dạng nền (background):
* background-color: Đặt màu nền.
* background-image: Đặt hình nền.
* background-repeat: Xác định cách lặp lại hình nền.
* background-position: Đặt vị trí hình nền.
* Mô hình hộp (box model):
* width và height: Đặt chiều rộng và chiều cao của phần tử.
* margin: Khoảng cách từ phần tử đến các phần tử khác.
* padding: Khoảng cách từ nội dung của phần tử đến viền của nó.
* border: Đặt đường viền cho phần tử.
* Các thuộc tính định margin, padding,border:
* margin-top, margin-right, margin-bottom, margin-left: Đặt khoảng cách từ viền ngoài của phần tử đến các phần tử khác.
* padding-top, padding-right, padding-bottom, padding-left: Đặt khoảng cách từ nội dung của phần tử đến viền của nó.
* border-width, border-style, border-color: Đặt độ dày, kiểu và màu của đường viền.
* Thuộc tính định dạng cách hiển thị (Display):
* display: Xác định cách hiển thị của phần tử (block, inline, inline-block, none, v.v.).
* Thuộc tính xác định vị trí (Position):
* position: Xác định cách phần tử được định vị (static, relative, absolute, fixed).
* top, right, bottom, left: Đặt vị trí tương ứng của phần tử.

## **3.8. Float &amp; Clear**

Float và Clear là hai thuộc tính quan trọng trong CSS để kiểm soát vị trí và bố trí của các phần tử trên trang web. Hãy cùng tìm hiểu chi tiết về chúng.

* Float

Thuộc tính float được sử dụng để định vị và định dạng nội dung, ví dụ như cho phép hình ảnh nổi bên trái hoặc bên phải của văn bản trong một khung.

Có ba giá trị chính cho thuộc tính float:

* left: Phần tử nổi bên trái của khung cha.
* right: Phần tử nổi bên phải của khung cha.
* none: Phần tử không nổi (hiển thị ở vị trí mà nó xuất hiện trong văn bản).

Ví dụ:

CSS

img

{

float: right;

}

* Clear

Thuộc tính clear xác định phần tử nào không được phép nổi bên cạnh phần tử đã được đặt thuộc tính float.

Có ba giá trị chính cho thuộc tính clear:

* left: Ngăn các phần tử nổi bên trái của nó.
* right: Ngăn các phần tử nổi bên phải của nó.
* both: Ngăn cả hai bên (trái và phải) của nó.

**3.9. Flex**

Thẻ flex trong CSS là một thuộc tính giúp sắp xếp và bố cục các thành phần khối một cách dễ dàng và linh hoạt hơn. Trước đây, chúng ta thường phải sử dụng JavaScript để hỗ trợ việc xử lý cấu trúc phức tạp, nhưng giờ đây display: flex giúp ta làm điều đó một cách dễ dàng hơn.

A screen shot of a computer code

Description automatically generated

*Hình 60. Ví dụ về flexx*

**3.10. Grid**

Thẻ grid trong CSS là một công cụ mạnh mẽ cho phép tạo bố cục hai chiều trên web. Nó giúp chúng ta dễ dàng xây dựng các trang web mà không cần sử dụng floats và positioning. Hãy cùng tìm hiểu về CSS Grid:

CSS Grid Layout (hay còn gọi là “Grid” hoặc “CSS Grid”) là một hệ thống bố cục dựa trên lưới, với hàng và cột, giúp thiết kế trang web một cách linh hoạt.

Khi một phần tử HTML có thuộc tính display được đặt thành grid hoặc inline-grid, nó trở thành một container của lưới.

Tất cả các phần tử con trực tiếp của container lưới tự động trở thành các phần tử lưới (grid items).

* Các thuộc tính quan trọng:
* Cột và hàng: Các đường dọc của phần tử lưới được gọi là cột, còn các đường ngang được gọi là hàng.
* Khoảng cách giữa cột và hàng: Các khoảng trống giữa các cột và hàng được gọi là gaps. Chúng ta có thể điều chỉnh kích thước gap bằng cách sử dụng các thuộc tính column-gap, row-gap, hoặc gap.

**3.11. CSS3**

CSS3 là phiên bản thứ 3 và cũng là mới nhất của CSS (Cascading Style Sheets). CSS3 được bổ sung thêm nhiều tính năng mới tiện lợi hơn CSS cho người dùng. Nó thừa hưởng tất cả những gì có trong phiên bản trước và bổ sung các tính năng mới, làm cho CSS3 hiện đang rất được ưa chuộng trong thiết kế website.

Một số tính năng mới của CSS3 bao gồm:

* Bo tròn góc (Border Radius): Cho phép làm tròn góc của các phần tử, tạo hiệu ứng mềm mại.
* Tạo độ bóng (Box Shadow): Thêm bóng đổ cho các phần tử, giúp chúng nổi bật hơn.
* Gradient: Tạo hiệu ứng màu sắc chuyển đổi từ một màu sang màu khác.
* Transitions và Animations: Cho phép tạo hiệu ứng chuyển động mượt mà.
* Multiple-column Layout: Tạo cấu trúc đa cột cho văn bản.
* Flexbox và Grid Layout: Cung cấp cách linh hoạt để sắp xếp các phần tử trên trang web.

CSS3 Selectors là các mẫu được sử dụng để chọn các phần tử HTML mà chúng ta muốn định dạng. Chúng giúp chúng ta kiểm soát cách các phần tử hiển thị trên trang web. Dưới đây là một số bổ sung cho CSS3.

CSS3 Rounded Corners (góc bo tròn trong CSS3) là một tính năng cho phép chúng ta làm tròn các góc của các phần tử trên trang web. Bằng cách sử dụng thuộc tính border-radius, chúng ta có thể thêm hiệu ứng góc bo tròn cho các phần tử HTML.

CSS3 Border Images là một tính năng trong CSS3 cho phép chúng ta sử dụng hình ảnh thay thế cho viền thông thường xung quanh một phần tử. Thay vì sử dụng viền đơn giản, chúng ta có thể áp dụng hình ảnh làm viền cho các phần tử trên trang web. Dưới đây là một số thuộc tính liên quan đến CSS3 Border Images:

* border-image-source: Thuộc tính này xác định đường dẫn tới hình ảnh được sử dụng làm viền.
* border-image-slice: Xác định cách lát cắt hình ảnh để tạo viền. Chúng ta có thể chỉ định từ 1 đến 4 giá trị để lát cắt theo các phần tử của viền.
* border-image-width: Định rõ bề rộng của hình ảnh viền. Cũng có thể chỉ định từ 1 đến 4 giá trị tương ứng với các phần tử của viền.
* border-image-outset: Xác định giá trị hình ảnh vượt ra ngoài vùng giới hạn của viền. Cũng có thể chỉ định từ 1 đến 4 giá trị tương ứng với các phần tử của viền.
* border-image-repeat: Xác định cách hình ảnh viền được lặp lại. Có ba giá trị: repeat, round, và stretch.

CSS3 Backgrounds là một tập hợp các thuộc tính trong CSS3 được sử dụng để định dạng nền (background) cho các phần tử trên trang web. Dưới đây là một số thuộc tính liên quan đến CSS3 Backgrounds:

* background-clip: Xác định vùng background được cắt bớt, giới hạn lại theo vùng chứa padding (không bao gồm border) hoặc border hoặc nội dung.
* background-origin: Xác định giá trị tương đối của background giới hạn theo vùng chứa padding (không bao gồm border) hoặc border hoặc nội dung.
* background-size: Xác định kích thước của vùng background. Chúng ta có thể chỉ định kích thước theo đơn vị (ví dụ: pixels), tỷ lệ phần trăm, hoặc sử dụng các giá trị như cover (tự chia tỷ lệ tới kích thước lớn nhất) hoặc contain (tự chia tỷ lệ tới kích thước nhỏ nhất).

CSS Gradients là một tính năng trong CSS cho phép chúng ta tạo ra hiệu ứng màu sắc chuyển đổi từ một màu sang màu khác. Khi sử dụng gradient, chúng ta sẽ thấy được sự mượt mà khi giao thoa của hai hoặc nhiều màu sắc.

Có hai loại chính của CSS Gradients:

* Linear Gradients: Đây là kiểu gradient trải dài theo một hướng cụ thể. Chúng ta có thể xác định hướng từ trên xuống dưới, từ dưới lên trên, từ trái qua phải hoặc từ phải qua trái.
* Radial Gradients: Đây là kiểu gradient tập trung về một điểm trung tâm. Nó tạo ra hiệu ứng từ một điểm phát tán ra.

CSS3 Shadows là thuộc tính có trong CSS3, cho phép tạo và điều chỉnh bóng đổ (shadow) cho các phần tử trên trang web một cách đơn giản và linh hoạt. Trước khi có CSS3 Shadows, để tạo shadow, ta thường phải sử dụng background-image và xử lý độ trong suốt (transparency) phức tạp.

A screen shot of a computer code

Description automatically generated

*Hình 61. Ví dụ về CSS3 Fonts*

Trong ví dụ trên:

* font-family xác định kiểu chữ được sử dụng (trước là Arial, sau là các font dự phòng).
* font-size xác định kích thước chữ.
* font-weight xác định độ đậm của chữ.
* color xác định màu chữ.

CSS3 2D Transforms là một tập hợp các thuộc tính trong CSS được sử dụng để xử lý hiệu ứng di chuyển 2D (không gian hai chiều) của các phần tử trên trang web. Điều này cho phép chúng ta thay đổi vị trí, hình dáng và xoay các phần tử theo nhiều hướng khác nhau.

Với thuộc tính transform có thể sử dụng các phương thức sau để biến đổi đối tượng:

* translate(): dịch chuyển phần tử từ vị trí hiện tại theo chiều X, Y
* rotate(): xoay phần tử theo chiều kim đồng hồ (đơn vị deg)
* scaleX(): tăng hoặc giảm chiều rộng của đối tượng
* scale(): tăng hoặc giảm kích thước của đối tượng
* matrix(): tập hợp tất cả các phép chuyển đổi đối tượng trong một phương thức
* matrix(scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY()).

CSS3 3D Transforms là một phần của CSS mở rộng chức năng của CSS Transforms, cho phép chúng ta biến đổi các phần tử được hiển thị bằng CSS trong không gian ba chiều. Điều này cho phép chúng ta tạo ra các hiệu ứng 3D thú vị trên trang web của mình.

Dưới đây là một số phương pháp biến đổi 3D mà chúng ta có thể sử dụng với thuộc tính transform:

* rotateX(): Xoay một phần tử quanh trục X của nó với một góc cụ thể. Ví dụ: transform: rotateX(150deg);.
* rotateY(): Xoay một phần tử quanh trục Y của nó với một góc cụ thể. Ví dụ: transform: rotateY(150deg);.
* rotateZ(): Xoay một phần tử quanh trục Z của nó với một góc cụ thể. Ví dụ: transform: rotateZ(90deg);.

Ngoài ra, còn có các phương pháp biến đổi khác như translate3d(), scale3d(), và perspective() để tạo ra các hiệu ứng 3D phong phú hơn.

CSS3 Transitions là một tính năng trong CSS cho phép chúng ta tạo hiệu ứng mượt mà và thanh lịch khi thay đổi kiểu dáng của một phần tử. Chúng dễ dàng triển khai và tương thích với hầu hết các trình duyệt. Chúng ta cũng có thể kết hợp chúng với CSS animations để tạo ra kết quả động đa dạng hơn.

Một số thuộc tính của CSS transition:

* transition: transition: <property> <duration> <timing-function> <delay>;
* transition-delay: khoảng thời gian chờ giữa thời gian thuộc tính được thay đổi và chuyển tiếp xảy ra
* transition-duration: định nghĩa khoảng thời gian chuyển tiếp xảy ra
* transition-property: định nghĩa thuộc tính CSS sẽ áp dụng transition
* transition-timing-function: định nghĩa phương thức xác định các giá trị tức thời của thuộc tính được tính toán

CSS3 Multiple Columns (hoặc còn gọi là CSS Columns) là một tính năng trong CSS cho phép chúng ta chia nội dung thành nhiều cột. Ý tưởng cơ bản của Multi-column là chúng ta có thể lấy một đoạn nội dung và chuyển nó thành nhiều cột giống như trong một tờ báo. Điều này hữu ích khi chúng ta muốn hiển thị văn bản hoặc nội dung khác thành các cột song song.

Có hai thuộc tính chính trong CSS Columns:

* column-count: Xác định số cột mà chúng ta muốn nội dung hiển thị.
* column-width: Xác định chiều rộng của mỗi cột.

Khi sử dụng CSS Columns, nội dung vẫn hiển thị theo trình tự bình thường và được chia thành các cột với độ rộng xác định. Điều này khác với các phương pháp dựng Layout khác như Flexbox hay CSS Grid, vì chúng thay đổi cách hiển thị các phần tử bên trong container.

Một số điểm lưu ý về CSS Columns:

* Chúng ta không thể style riêng lẻ từng cột, nhưng có thể thêm rule giữa chúng sử dụng thuộc tính column-rule.
* Khoảng cách giữa các cột có thể kiểm soát bằng thuộc tính column-gap.

CSS3 Animations là công nghệ được giới thiệu trong phiên bản CSS3. Nó cho phép chúng ta tạo hiệu ứng chuyển động mà không cần sử dụng Javascript hay Flash. Trong CSS3 Animations, chúng ta có thể tạo các hiệu ứng chuyển động cho các phần tử trên trang web.

Các thuộc tính CSS hỗ trợ tạo hoạt cảnh:

* animation-delay: thời gian trễ trước khi animation bắt đầu
* animation-interation-count: số lần animation chạy, giá trị infinite là vô hạn lần
* animation-direction: chiều của hoạt cảnh qua các khung hình với các giá trị normal (tiến),reverse (lùi), alternate (trước tiến sau đó lùi), alternate-reverse (trước lùi rồi tiến)
* animation-timing-function: định nghĩa đường cong tốc độ của hoạt cảnh: ease (bắt đầu chậm sau đó nhanh và kết thúc chậm), linear, ease-in, ease-out, ease-in-out, cubic-bezier, ...
* animation-fill-mode: hoạt cảnh không tác động lên đối tượng trước và sau hoạt cảnh có thể chỉnh bằng các giá trị none (mặc định), forwards (giữa lại giá trị style cuối cùng trong hoạt cảnh), backwards (giữ lại style đầu tiên trong hoạt cảnh), both (giữ lại cả 2)
* Animation: thuộc tính gộp theo thứ tự animation-name, animation-duration, animation-timing-function, animation-delay, animation-iteration-count, animation-direction

3

**3.12. SCSS**

**SCSS** (Syntactically Awesome Style Sheets) là một chương trình tiền xử lý CSS (CSS preprocessor). [Nó giúp chúng ta viết CSS theo cách của một ngôn ngữ lập trình, có cấu trúc rõ ràng, rành mạch, dễ phát triển và bảo trì code hơn](https://topdev.vn/blog/sass-scss-la-gi/).

SCSS ra đời sau SASS và có cú pháp viết tương tự như cách viết CSS. Cú pháp này được tạo ra nhằm thu hẹp khoảng cách giữa SASS và CSS bằng cách mang lại một thứ gì đó thân thiện với CSS. Trong SCSS, chúng ta có thể sử dụng cú pháp giống với Ruby (vì đơn giản nó được thiết kế bởi các lập trình viên Ruby). [Điều này giúp chúng ta viết code CSS một cách dễ dàng và đơn giản hơn](https://viblo.asia/p/sassscss-la-gi-bWrZnzy9Zxw).

**3.13. SASS**

SASS (Syntactically Awesome Style Sheets) là một phần mở rộng hay bộ tiền xử lý, ngôn ngữ lập trình được sử dụng để hỗ trợ cho nền tảng CSS. [Nó giúp chúng ta viết CSS theo cách của một ngôn ngữ lập trình, có cấu trúc rõ ràng, dễ phát triển và bảo trì code hơn](https://topdev.vn/blog/sass-la-gi/).

Dưới đây là một vài tính năng của SASS:

* Xếp chồng (Nested Rules): Xếp chồng là tính năng khá rất hay và đặc biệt của sass. Nó được sử dụng một cách rất thường xuyên khi một project viết CSS bằng SASS.
* Biến (variable): Biến là tính năng được sử dụng vô cùng cơ bản trong Sass. Chúng ta chỉ cần bắt đầu bằng chữ $ khi đặt tên và khai báo cho biến. Biến có chứa tất cả những giá trị mà chúng ta đã sử dụng nhiều lần, ví dụ như font chữ, màu sắc, kiểu chữ,...
* Quy tắc Mixin: Mixin là một cơ chế khá phổ biến trong Sass giúp tạo ra các hàm và chúng ta hoàn toàn có thể sử dụng nó bằng cách truyền các tham số vào bên trong. Công dụng của quy tắc mixin chính là mang lại những thuộc tính mà chúng ta đã quy ước sẵn trong một mix trước đó và include nó vào một thành phần bất kỳ và chúng ta sẽ không cần phải viết lại những thuộc tính đó.
* Kế thừa (Extends): Khi nghe đến tính năng kế thừa hay còn gọi là extends thì nó sẽ tương tự như lập trình hướng đối tượng OOP. Cách thực hiện tính kế thừa đơn giản chỉ là chúng ta sẽ phải định nghĩa ra một class và @extends những tag cần sử dụng là xong.

**Kết luận chương 3**

CSS và CSS3 đóng vai trò quan trọng trong việc tạo ra các trang web đẹp mắt, thu hút và dễ sử dụng. CSS cung cấp các công cụ cơ bản để định dạng giao diện trang web, bao gồm màu sắc, phông chữ, bố cục, và hiệu ứng. CSS3 là phiên bản nâng cao của CSS, bổ sung nhiều tính năng mới giúp bạn tạo ra các trang web với giao diện hiện đại, đáp ứng và tương tác tốt hơn..

**CHƯƠNG IV: THIẾT KẾ WEB BÁN THIẾT BỊ Y TẾ**

## **4.1.** **Ý tưởng của Website**

Em có ý tưởng là phát triển một dự án thiết kế trang web và ứng dụng di động dành cho thiết bị y tế. Trang web và ứng dụng sẽ cung cấp một nền tảng toàn diện cho người dùng để tìm kiếm, tìm hiểu và mua sắm các sản phẩm y tế một cách thuận tiện và đáng tin cậy.

Trong dự án này, em đề xuất tích hợp một phần mở rộng trên trang web cho phép người dùng thảo luận, chia sẻ kinh nghiệm và kiến thức về các vấn đề y tế. Điều này sẽ tạo ra một cộng đồng trực tuyến đồng lòng trong việc chăm sóc sức khỏe, đồng thời nâng cao giá trị của trang web.

Ngoài ra, em cũng đề xuất phát triển một ứng dụng di động đi kèm với trang web, giúp người dùng dễ dàng truy cập thông tin và mua sắm sản phẩm y tế từ bất kỳ nơi đâu và bất kỳ thiết bị nào. Ứng dụng này sẽ cung cấp các tính năng tiện ích như quét mã vạch để tra cứu thông tin sản phẩm, thông báo về các ưu đãi đặc biệt và nhắc nhở về lịch trình chăm sóc sức khỏe cá nhân.

Dự án này không chỉ nhằm mục đích cải thiện trải nghiệm mua sắm và tiếp cận thông tin y tế, mà còn mong muốn xây dựng một cộng đồng y tế trực tuyến tích cực và hữu ích cho mọi người..

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

Để xây dựng được bố cục của trang web, trước tiên chúng t cần phải tạo thư mục chứa các file liên quan đến trang web, chẳng hạn như các file ảnh, video, code HTML.



*Hình 62. Ảnh ninh họa thư mục*

Đầu tiên chúng ta tạo thư mục trong phần ổ cứng của máy tính chúng ta, sau đó vào ta mở ứng dụng thực hiện code và tìm chọn mở thư mục mà ta mới tạo. Chọn thư mục xong thì tiếp theo ta tạo file mới. File này sẽ là file code chính đầu tiên của chúng ta và khi chúng ta tạo file thì bao giờ ở cuối tên file sẽ có thêm đuôi .html để file code của chúng ta có thể thực hiện khởi chạy chương trình code.

A screenshot of a computer

Description automatically generated

*Hình 63. Ảnh minh họa tạo file index.html*

Tạo file và lưu vào thư mục xog, khi đó, ta sẽ đến giao diện của file đã tạo. Ở dòng thứ nhất, chúng ta nhấn phím “!” và phím Enter để file tự khởi tạo ra cấu trục cơ bản của HTML.

Cấu trúc cơ bản của HTML gồm có các phần:

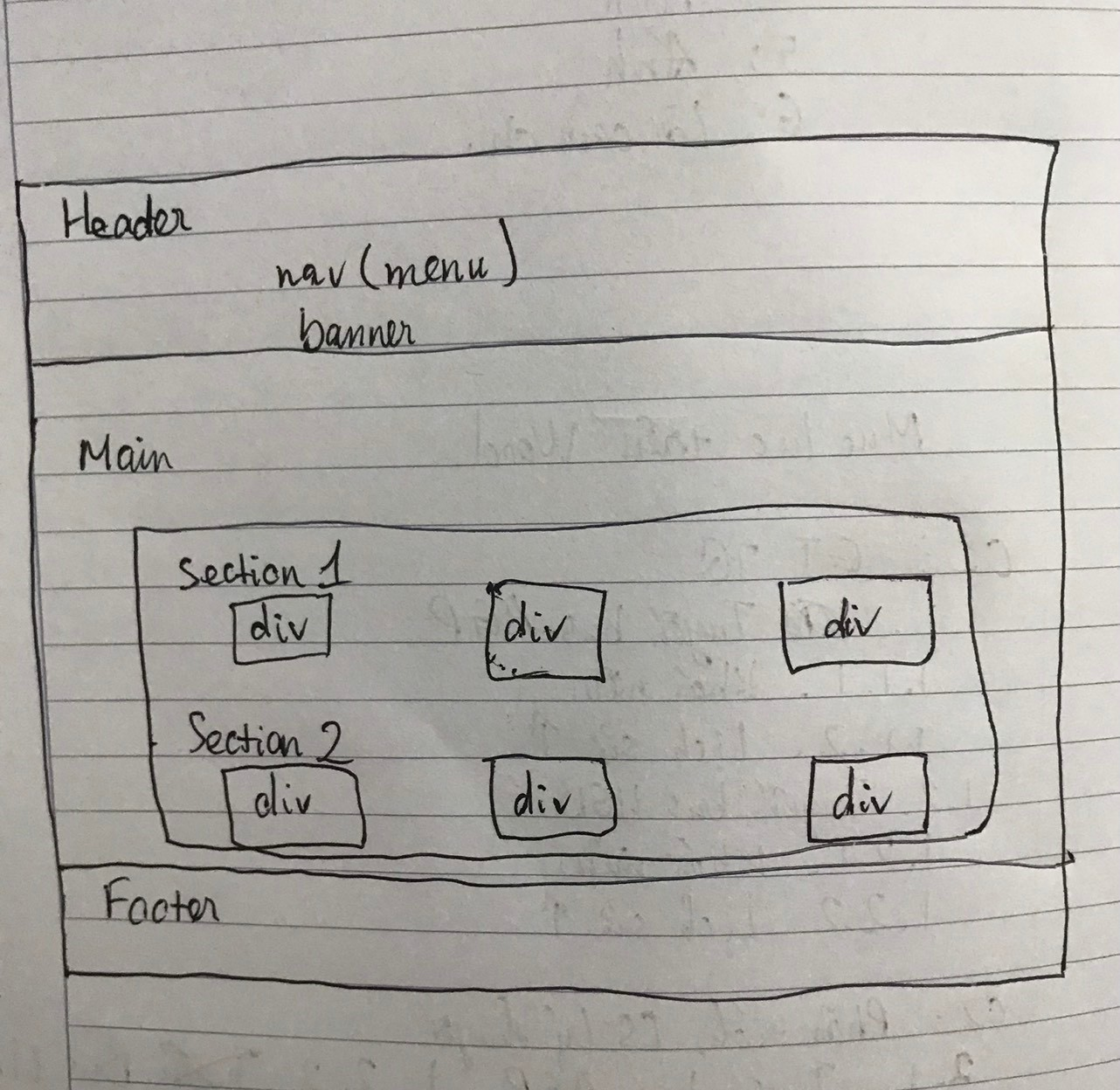
A screen shot of a computer code

Description automatically generated

*Hình 64. Cấu trúc cơ bản của html*

Trong đó:

* <!DOCTYPE html>: Đây là khai báo doctype HTML5, nó cho trình duyệt biết về phiên bản HTML mà trang web của bạn sử dụng.
* <html lang="en">: Thẻ mở đầu của trang web, đánh dấu bắt đầu của tài liệu HTML. Thuộc tính “lang” định rõ ngôn ngữ sử dụng (ở đây là tiếng Anh).
* <head>: Phần này chứa các siêu thông tin về trang web, chẳng hạn như kí tự mà trình duyệt nên sử dụng, title của trang, và liên kết với các tệp CSS.
* <meta charset="UTF-8">: Thẻ meta định rõ bảng mã sử dụng cho trang web (ở đây là “UTF-8”, tiêu chuẩn cho nhiều trang web ngày nay).
* <meta name="viewport" content="width=device-width, initial-scale=1.0">: Thẻ meta này cung cấp thông tin về cách trang web sẽ được hiển thị trên các thiết bị di động.
* <title>Trang Web Của Bạn</title>: Đây là nơi bạn đặt tiêu đề của trang web, nó sẽ xuất hiện trên thanh tiêu đề của trình duyệt.
* <link rel="stylesheet" href="styles.css">: Liên kết đến tệp CSS (nếu có) để tùy chỉnh giao diện của trang web.
* <body>: Phần này chứa nội dung chính của trang web, bao gồm tiêu đề, thanh điều hướng, nội dung chính, và chân trang.
* <header>...</header>: Khu vực đầu trang thường chứa tiêu đề chính của trang web hoặc logo.
* <nav>...</nav>: Khu vực này chứa thanh điều hướng, với các liên kết đến các trang khác của trang web.
* <section>...</section>: Đây là phần chính chứa nội dung của trang web. Bạn có thể có nhiều phần `section` tùy thuộc vào cấu trúc của trang web.
* <footer>...</footer>:Khu vực cuối trang thường chứa thông tin chung như bản quyền hoặc liên kết đến các trang quản lý của trang web.



*Hình 65 . Ảnh ninh họa*

Về bài của em, em đã chia bài web của em thành ba phần. Phần 1 là header, header là một phần của trang web được sử dụng để chứa thông tin mà bạn muốn hiển thị ở đầu trang, thường là tiêu đề, menu điều hướng, hoặc các phần tử khác như thông tin liên hệ. Header thường là một phần quan trọng của cấu trúc trang web và có thể chứa các phần tử khác.

Trong phần này, em có thẻ <nav> dùng để nhóm các đối tượng thành các đề mục là “Giới thiệu”, “Thiết bị xét nghiệm”, “Thiết bị y tế gia đình”, “Thiết bị y tế bệnh viện”, “Kiến thức”, “Liên hệ” và rồi dùng thẻ <ul>, <li> để điều chỉnh sao cho khoảng cách giữa các đề mục đều cân đối, bằng nhau.



*Hình 66 . Ảnh ninh họa*

Đối với các đề mục “Thiết bị xét nghiệm”, “Thiết bị y tế gia đình”, “Thiết bị y tế bệnh viện”, “Kiến thức” thì trong đó em có các đề mục nhỏ hơn, gồm từ 3-4 các đề mục nhỏ, phân chia chi tiết hơn về các thiết bị cụ thể, các thông tin văn bản...

A blue and white rectangular box with text

Description automatically generated with medium confidence

*Hình 67 . Ảnh ninh họa*

Cũng trong phần header này, em để banner là dòng chữ “Shop bán thiết bị y tế” có thể di chuyển từ trái qua phải nhờ vào thẻ <marquee>.

A blue sign with white letters

Description automatically generated

*Hình 68 . Ảnh ninh họa*

Phần 2 là phần main-phần chính giữa trang web-em có để trong đó các thẻ <section> dùng để chia thành các bảng chứa trong đó các thẻ <div>.

Phần <main> trong HTML là một phần tử dùng để đánh dấu nội dung chính của trang web. Thường thì phần này chứa nội dung cốt lõi của trang, bao gồm các đoạn văn bản, hình ảnh, video, bảng, hoặc bất kỳ loại nội dung nào khác mà trang web cần hiển thị cho người dùng.



Hình 67*. Ảnh ninh họa*

Sử dụng thẻ <main> giúp cải thiện sự cấu trúc của trang web và làm cho việc đọc mã nguồn HTML dễ dàng hơn. Đồng thời, nó cũng giúp trình duyệt hiểu được rõ ràng rằng phần nào của trang chứa nội dung chính, điều này có thể hữu ích cho các công cụ hỗ trợ đọc cho người khiếm thị hoặc các công cụ tìm kiếm.

A black background with white text

Description automatically generated

*Hình 69 : . Ảnh ninh họa*

Trong thẻ <section > em có để các thẻ <form> ,thẻ <Label>,thẻ <input> và thẻ <button> trong đó:

Thẻ <form> được sử dụng để tạo một mẫu biểu (form), nơi mà người dùng có thể nhập dữ liệu và gửi nó đến một máy chủ để xử lý. Thẻ <form> đóng vai trò quan trọng trong việc thu thập thông tin từ người dùng, như tạo tài khoản, gửi tin nhắn, đăng nhập, hoặc thực hiện các thao tác khác tùy thuộc vào mục đích của trang web.



Hình : *. Ảnh ninh họa*

Thẻ <label> được sử dụng để tạo ra một nhãn cho một phần tử nhập liệu (input) trong một mẫu biểu (form). Mục đích chính của thẻ <label> là cung cấp một nhãn mô tả cho ô nhập liệu, giúp người dùng hiểu được ý nghĩa của trường dữ liệu đó.



Hình *. Ảnh ninh họa*

Thẻ <input> được sử dụng để tạo ra các ô nhập liệu (input fields), nơi mà người dùng có thể nhập dữ liệu trực tiếp vào trang web. Thẻ <input> có nhiều loại (type) khác nhau tùy thuộc vào mục đích sử dụng.



Hình : *. Ảnh ninh họa*

Thẻ <button> được sử dụng để tạo ra một nút (button) trên trang web, có thể được sử dụng để thực hiện các hành động hoặc chức năng cụ thể khi được nhấn. Thẻ <button> có thể được sử dụng để thực hiện các hành động như gửi một mẫu biểu (form), mở một liên kết, hoặc kích hoạt các hành động JavaScript.

A computer screen shot of text

Description automatically generated

*Hình 70 . Ảnh ninh họa*

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

Ở bài code của em, em gồm có 6 file code và 12 file code nhỏ. Trong đó có một file tạo đầu tiên và là file tổng, nó có tên là index.html. Khí chúng ta khởi chạy chương trình code thì file này sẽ là file chạy đầu tiên và nó đưa ta đến giao diện đầu tiên của web.

A screenshot of a computer

Description automatically generated

*Hình 71: . Ảnh ninh họa*

Khi ta vào giao diện đầu của web, sẽ có các mục ở phía bên trên cùng của màn hình., đó là phần header, nó giúp ta đưa các mục lên phía bên trên của màn hình. Nó cũng cho chúng ta chèn các kiểu chữ, dạng văn bản trên đó.



*Hình 72 . Ảnh ninh họa*

Trong header em có các thuộc tính như <nav>, các thẻ như <ul> , <li>,<a>,<p>.

Thuộc tính <nav> được sử dụng để đánh dấu một phần của trang web chứa các liên kết điều hướng hoặc menu. Thẻ <nav> không có bất kỳ thuộc tính đặc biệt nào, nhưng nó được sử dụng để chỉ định rõ rằng phần nội dung trong đó chứa các liên kết điều hướng.

Thẻ <ul>,<li> dùng để tạo ra một danh sách để chứa các mục hiển thị trên trang web. Kèm với thẻ <ul>,<li> thì em có thẻ <a> dùng để gắn các đường liên kết cho một từ hoặc một cụm từ nhất định.

A black background with white text

Description automatically generated

*Hình 73 . Ảnh ninh họa*

Với các thẻ <a>, em đã gắn vào từng mục một đường link dẫn đến các file của mục đó. Ở trong các mục đó cũng sẽ chứa tương tự các phần như ở file index.html nhưng có một số file chỉ gồm các đoạn văn bản, thay vì gồm các lệnh code. Các file đó là các file sẽ đưa ra những thông tin ví dụ như giới thiệu về web, các văn bản thông tin về đời sống, thông báo tin tức.

A screen shot of a computer code

Description automatically generated

*Hình 74 . Ảnh ninh họa*

Như trong hình, thẻ <div> em dùng để nhóm những mục nhỏ của các mục lớn và đặt tên cho chúng để khi ta sang file index.css thì chúng ta sẽ dễ đặt bố cục và đặt cho các mục đó vào khuôn hợp lệ hơn,

Về phần thứ hai là phần main-phần bố cục ở giữa màn hình- trong đó có chứa các khối tập lệnh đối với các file dùng chạy tập lệnh, còn đối với các file chứa thông tin thì em sẽ để các đoạn văn bản vào trong đó.

A computer screen shot of a program code

Description automatically generated

*Hình 75 . Ảnh ninh họa*

Trong các file dùng để chạy tập lệnh, em đã đưa vào đó một số các thuộc tính và một số thẻ thông dụng, rất hay được sử dụng quá trình làm web.

Trong HTML, thẻ <select> được sử dụng để tạo ra một danh sách dropdown chứa các lựa chọn. Ở đây em chứa các khối thẻ <div> và để đặt cho mỗi thẻ <div> có một định dạng chung như nhau.

A black background with white text

Description automatically generated

*Hình 76 . Ảnh ninh họa*

Thẻ <div> là một thẻ phổ biến được sử dụng để tạo ra một khu vực hay một phần trong trang web có thể chứa nhiều loại nội dung khác nhau. Thẻ <div> không mang lại bất kỳ ý nghĩa ngữ cảnh nào mà chỉ đơn giản là một hộp container mà bạn có thể sử dụng để nhóm các phần tử HTML lại với nhau và áp dụng các kiểu dáng hoặc thuộc tính CSS cho toàn bộ nhóm đó.



Hình *. Ảnh ninh họa*

Thẻ <img> là thẻ được sử dụng để hiển thị một hình ảnh trên trang web. Thẻ <img> không có thẻ đóng (</img>), và sử dụng thuộc tính src để chỉ định đường dẫn tới tập tin hình ảnh cần hiển thị.



Hình *. Ảnh ninh họa*

Trong HTML, thẻ <h1> được sử dụng để định dạng và hiển thị tiêu đề cấp độ 1 trên trang web. Tiêu đề cấp độ 1 thường được sử dụng để làm tiêu đề chính của trang hoặc một phần quan trọng của nội dung.

Trong HTML, chúng ta có thể sử dụng thuộc tính onclick để gắn một hàm JavaScript hoặc một biểu thức JavaScript vào một sự kiện click của một phần tử HTML, chẳng hạn như một thẻ <button>.



Hình *. Ảnh ninh họa*

Đến với phần footer. Footer là phần ngược lại so với phần header khi nó là phần chỉ định để chứa các mục ở dưới cùng của trang web.Ở phần này, chúng ta có thể đưa vào đó các đường link, ảnh hoặc các thông tin, địa chỉ cụ thể.



*Hình 77 . Ảnh ninh họa*

Cuối cùng đến với phần index.css.Tại đây em đã đưa ra những thông số cụ thể về từng phần,từ header cho đến main và footer. Mọi người có thể tham khảo qua các anhnr sau đây.

A screen shot of a computer program

Description automatically generated

Hình *. Ảnh ninh họa*

A screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

Hình *. Ảnh ninh họa*

## **4.4**. **Kết quả đạt được**

Dự án thiết kế trang web dành cho thiết bị y tế đã đạt được những kết quả đáng kinh ngạc, mang lại những lợi ích đáng kể cho cả người dùng và nhà cung cấp sản phẩm y tế. Trang web này đã được thiết kế với mục tiêu cung cấp thông tin chính xác và dễ tiếp cận về các sản phẩm, dịch vụ y tế, giúp người dùng dễ dàng tìm kiếm và lựa chọn sản phẩm phù hợp nhất cho nhu cầu của mình.

Một số kết quả đáng chú ý mà dự án này đã đạt được bao gồm:

Nâng cao trải nghiệm người dùng: Trang web được thiết kế một cách thân thiện với người dùng, tạo ra một giao diện dễ sử dụng và dễ tìm kiếm thông tin. Điều này đã tăng cường trải nghiệm của người dùng khi tìm kiếm thông tin về các thiết bị y tế.

**A screenshot of a computer

Description automatically generated**

*Hình 78 :Giao diện bên trên của web*

Tăng cường tương tác giữa người dùng và nhà cung cấp: Trang web cung cấp các tính năng như hỗ trợ trực tuyến, đánh giá sản phẩm, và hệ thống phản hồi, giúp người dùng dễ dàng tương tác với nhà cung cấp, đặt câu hỏi và chia sẻ ý kiến.

**A screenshot of a computer

Description automatically generated**

*Hình 79 :Giao diện bên dưới của web.*

Tăng cường hiệu suất kinh doanh cho nhà cung cấp: Trang web không chỉ là một nền tảng để giới thiệu sản phẩm, mà còn là một công cụ quản lý đơn hàng, dữ liệu khách hàng và tiếp thị hiệu quả. Điều này đã giúp tăng cường hiệu suất kinh doanh và tạo ra cơ hội kinh doanh mới.

A screenshot of a computer

Description automatically generated

*Hình 80 :Giao diện khi mở mục nhỏ*

Cung cấp thông tin chính xác và đáng tin cậy: Trang web được thiết kế với sự chú trọng vào việc cung cấp thông tin chính xác và đáng tin cậy về các sản phẩm y tế. Điều này giúp người dùng cảm thấy yên tâm và tin tưởng khi tìm kiếm và mua sắm trên trang web.

A screenshot of a computer

Description automatically generated

Hình *. Ảnh ninh họa*

Tóm lại, dự án thiết kế trang web cho thiết bị y tế đã đạt được những kết quả tích cực, mang lại lợi ích đáng kể cho cả người dùng và nhà cung cấp sản phẩm y tế, đồng thời nâng cao trải nghiệm mua sắm và tiếp thị trong ngành y tế.

**Kết luận chương 4**

Kết luận chương 4 cho dự án thiết kế trang web dành cho thiết bị y tế:

Trong chương này, chúng ta đã thảo luận về quá trình thiết kế và triển khai trang web dành cho thiết bị y tế. Dự án đã mang lại những kết quả đáng chú ý và có ảnh hưởng tích cực đối với người dùng cũng như nhà cung cấp trong lĩnh vực y tế. Dưới đây là những kết luận chính từ quá trình nghiên cứu:

Tạo ra một trải nghiệm người dùng tốt hơn: Trang web được thiết kế với một giao diện thân thiện và dễ sử dụng, tạo điều kiện thuận lợi cho người dùng tìm kiếm và tìm hiểu về các sản phẩm y tế. Việc tăng cường tương tác thông qua các tính năng như hỗ trợ trực tuyến và đánh giá sản phẩm cũng đã giúp cải thiện trải nghiệm của họ.

Tăng cường quản lý kinh doanh: Trang web không chỉ là một công cụ giới thiệu sản phẩm mà còn là một hệ thống quản lý đơn hàng, dữ liệu khách hàng và tiếp thị hiệu quả. Điều này đã giúp nhà cung cấp tăng cường hiệu suất kinh doanh và tạo ra cơ hội kinh doanh mới.

Xây dựng uy tín và lòng tin từ người dùng: Việc cung cấp thông tin chính xác và đáng tin cậy về các sản phẩm y tế đã giúp xây dựng sự tin tưởng từ phía người dùng. Điều này quan trọng đặc biệt trong ngành y tế, nơi sự tin cậy và uy tín là yếu tố quyết định.

Mở ra cơ hội mở rộng thị trường: Trang web đã mở ra cơ hội cho nhà cung cấp tiếp cận khách hàng mới và mở rộng thị trường. Việc tăng cường hiệu quả tiếp thị trên nền tảng trực tuyến đã giúp tạo ra sự nhận biết và tăng cường vị thế cạnh tranh của họ.

Tổng quát, dự án thiết kế trang web dành cho thiết bị y tế không chỉ là một thành công về mặt kỹ thuật mà còn là một bước tiến quan trọng trong việc cải thiện trải nghiệm mua sắm và tiếp thị trong ngành y tế. Sự kết hợp giữa công nghệ và y tế đã tạo ra những giá trị đáng kể cho cả người dùng và doanh nghiệp trong lĩnh vực này.

**KẾT LUẬN**

Dự án thiết kế trang web cho thiết bị y tế đã đem lại những kết quả quan trọng và mang tính cách mạng trong ngành y tế. Dưới đây là ba ý kết luận về ưu điểm, nhược điểm và hướng phát triển của đề tài:

* Ưu điểm:

Cải thiện trải nghiệm người dùng: Trang web cung cấp một giao diện thân thiện và dễ sử dụng, giúp người dùng dễ dàng tìm kiếm và tìm hiểu về các sản phẩm y tế.

Tăng cường tương tác: Tính năng như hỗ trợ trực tuyến và đánh giá sản phẩm đã tạo ra một môi trường tương tác tích cực giữa người dùng và nhà cung cấp.

Quản lý kinh doanh hiệu quả: Trang web không chỉ là một công cụ giới thiệu sản phẩm mà còn là một hệ thống quản lý đơn hàng và dữ liệu khách hàng, giúp tăng cường hiệu suất kinh doanh.

* Nhược điểm:

Yêu cầu kiến thức chuyên môn cao: Việc tích hợp thông tin y tế đòi hỏi sự hiểu biết sâu rộng về lĩnh vực này, đôi khi gặp khó khăn trong việc cập nhật thông tin mới nhất và đảm bảo tính chính xác.

Rủi ro về bảo mật thông tin: Vì tính chất nhạy cảm của thông tin y tế, trang web cần phải đảm bảo các biện pháp bảo mật cao để bảo vệ thông tin cá nhân của người dùng.

Khả năng hạn chế tiếp cận: Trang web có thể không phản ánh được tất cả nhu cầu và mong muốn của mọi người dùng do hạn chế về phạm vi và tính chất của sản phẩm.

* Hướng phát triển chủ đề:

Nâng cao chất lượng dịch vụ và sản phẩm: Tiếp tục phát triển và cập nhật thông tin y tế mới nhất để đảm bảo trang web luôn cung cấp những sản phẩm chất lượng và dịch vụ tốt nhất cho người dùng.

Mở rộng phạm vi và đối tượng người dùng: Tăng cường tiếp cận đến đối tượng người dùng rộng lớn hơn, bao gồm cả bệnh nhân và nhà cung cấp y tế, để đáp ứng được nhiều nhu cầu khác nhau trong ngành y tế.

Tối ưu hóa bảo mật thông tin: Đầu tư vào các biện pháp bảo mật tiên tiến để đảm bảo an toàn và bảo vệ thông tin cá nhân của người dùng một cách toàn diện.

Tổng quát, dự án thiết kế trang web cho thiết bị y tế đã đạt được những thành tựu đáng kể, nhưng vẫn còn nhiều tiềm năng để phát triển và cải thiện trong tương lai. Qua việc tập trung vào những ưu điểm, giải quyết nhược điểm và phát triển hướng đi mới, chúng ta có thể xây dựng một hệ thống trang web y tế ngày càng hoàn thiện và hữu ích đối với cộng đồng.

**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.

James F. Kurose and Keith W. Ross (2013), *Computer Networking: A top-down approach sixth Edition*, Pearson Education