**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TPHCM**

**KHOA ĐÀO TẠO QUỐC TẾ**

🙠🙞🕮🙜🙢

**MÔN HỌC: INTRODUCTION TO INFORMATION TECHNOLOGY**

**BÀI BÁO CÁO**

**NGHIÊN CỨU, PHÁT TRIỂN, XÂY DỰNG TRANG WEB HỖ TRỢ KHÁCH DU LỊCH TẠI THÀNH PHỐ HỒ CHÍ MINH**

**Giáo viên hướng dẫn:** Nguyễn Đăng Quang

**Nhóm thực hiện:** 4

**Sinh viên thực hiện:**

1. Lê Ngọc Hải 24110089

2. Nguyễn Quốc Bảo Khang 24110099

3. Trần Gia Kiệt 24110103

4. Nguyễn Đăk Lộc 24110107

5. Đặng Tấn Phúc Thịnh 24110135

**Mã lớp học:** INIT130185E - Nhóm 02 FIE

Thành phố Hồ Chí Minh, tháng 12 năm 2024

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

.....................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

Giáo viên hướng dẫn

Nguyễn Đăng Quang

**MỤC LỤC**

[**PHẦN MỞ ĐẦU** 1](#_tyjcwt)

[1. Nội dung đề tài 1](#_3dy6vkm)

[2. Lí do chọn đề tài 1](#_1t3h5sf)

[3. Đối tượng nghiên cứu 1](#_4d34og8)

[4. Mục tiêu nghiên cứu 1](#_2s8eyo1)

[**PHẦN NỘI DUNG** 2](#_17dp8vu)

[**CHƯƠNG I.** **LÝ THUYẾT** 2](#_3rdcrjn)

[1.](#_26in1rg)1. Giới thiệu về website2

1.1.1. Khái niệm về website 2

1.1.2. Các thành phần của giao diện website 3

1.1.3. Tầm quan trọng của website 4

1.2 Công nghệ và ngôn ngữ lập trình website 4

1.2.1. Frontend: HTML, CSS, JavaScript 4

1.3. Quy trình phát triển website 19

1.4. Bảo mật trong phát triển website 20

1.5. Công cụ kiểm thử và tối ưu website 20

[**CHƯƠNG II.** **ỨNG DỤNG**](#_37m2jsg) 21

[2.](#_1mrcu09)1. [Lên ý tưởng](#_1mrcu09) 21

[2.2. Thiết kế thông qua phần mềm thứ 3 (Adobe Photoshop)](#_46r0co2) 21

[2.](#_2zbgiuw)3. [Phát triển ý tưởng](#_2zbgiuw) 24

[2.4](#_3cqmetx). [Cấu hình và kiểm thử 24](#_3cqmetx)

[2.5. Cài đặt ban đầu 24](#_1rvwp1q)

[**PHẦN KẾT LUẬN**](#_43ky6rz) 25

[1. Kết quả đạt được](#_2iq8gzs) 25

[2. Thuận lợi](#_xvir7l) 25

[3. Khó khăn](#_3hv69ve) 25

[4. Nhận xét và đánh giá tổng quan](#_1x0gk37) 25

[5. Đánh giá mức độ hoàn thành của thành viên](#_2w5ecyt) 28

[**TÀI LIỆU THAM KHẢO**](#_1baon6m) 29

**PHẦN MỞ ĐẦU**

1. **Nội dung đề tài:**

Thiết kế website cung cấp thông tin tham quan, du lịch ở TP.HCM.

## Lý do chọn đề tài:

Lý do chọn đề tài "Xây dựng Website Du lịch" xuất phát từ sự phát triển mạnh mẽ của ngành du lịch trực tuyến, khi ngày càng nhiều khách hàng tìm kiếm sự tiện lợi, linh hoạt và cá nhân hóa trong việc đặt dịch vụ du lịch. Với sự gia tăng của người dùng internet và các thiết bị di động, việc xây dựng một nền tảng trực tuyến giúp đáp ứng nhu cầu của khách du lịch trong và ngoài nước, đồng thời tận dụng tiềm năng thị trường du lịch lớn, đặc biệt tại Việt Nam. Công nghệ phát triển mạnh mẽ, cùng với sự gia tăng của thương mại điện tử, tạo cơ hội lớn để các website du lịch không chỉ cung cấp dịch vụ mà còn sáng tạo thêm những tính năng mới, nâng cao trải nghiệm người dùng.

## Đối tượng nghiên cứu:

Khách hàng tiềm năng

Các đối tác kinh doanh

Ứng dụng và tích hợp công nghệ

## Mục tiêu nghiên cứu:

Mục tiêu nghiên cứu khi phát triển website du lịch là xác định rõ các yếu tố cần thiết để tạo ra một nền tảng du lịch trực tuyến hiệu quả, đáp ứng nhu cầu của khách hàng, tối ưu hóa trải nghiệm người dùng, đảm bảo bảo mật và tuân thủ pháp lý. Từ đó, xây dựng một website có khả năng cạnh tranh và phát triển bền vững trong ngành du lịch trực tuyến.

# PHẦN NỘI DUNG

1. **LÝ THUYẾT**
   1. **Giới thiệu về website**

**1.1.1. Khái niệm về website**

Website là một tập hợp các trang web liên kết với nhau, được lưu trữ trên một máy chủ web và có thể truy cập thông qua mạng Internet. Nó là một không gian trực tuyến cho các tổ chức, doanh nghiệp, cá nhân hoặc tổ chức phi lợi nhuận để chia sẻ thông tin, sản phẩm, dịch vụ hoặc nội dung với người dùng trên khắp thế giới. Mỗi trang web trong một website có thể chứa nội dung đa dạng như văn bản, hình ảnh, video, âm thanh, biểu đồ và các phương tiện truyền thông khác. Người dùng có thể truy cập vào website bằng cách sử dụng một trình duyệt web như Google Chrome, Mozilla Firefox,… và nhập địa chỉ của website (tên miền) vào thanh địa chỉ.

*Một số khái niệm khác:*

*Domain name (Tên miền):* Là địa chỉ duy nhất để truy cập vào website. Tên miền thường được mua từ các công ty đăng ký tên miền và được cung cấp trong thời hạn thuê. Ví dụ: google.com, facebook.com

*DNS server (Domain Name System server*): Là một hệ thống phân giải tên miền sang địa chỉ IP. Khi người dùng gõ một tên miền vào trình duyệt, trình duyệt sẽ gửi yêu cầu đến máy chủ DNS để lấy địa chỉ IP tương ứng với tên miền đó.

*URL (Uniform Resource Locator*): Là địa chỉ cụ thể của một tài nguyên trên Internet. Nó bao gồm ba phần chính: giao thức (ví dụ: "http://" hoặc "https://"), tên miền (ví dụ: "google.com") và đường dẫn tới tài nguyên cụ thể trên trang web (ví dụ: "/search"). URL định danh chính xác một tài nguyên cụ thể trên mạng.

*Hosting:* Là dịch vụ cung cấp không gian lưu trữ trên máy chủ web để lưu trữ các tệp tin và dữ liệu của một website. Website của bạn được lưu trữ trên máy chủ web đó, và khi người dùng truy cập vào tên miền của bạn, máy chủ web sẽ cung cấp nội dung của website cho người dùng.

*Phân loại website:*

*Website tĩnh:* website có các website được lưu trữ trên máy chủ ở định dạng được gửi đến trình duyệt web của khách hàng. Nó chủ yếu được mã hóa bằng [Hypertext Markup Language](https://vi.wikipedia.org/wiki/HTML) (HTML); [Cascading Style Sheets](https://vi.wikipedia.org/wiki/CSS) (CSS) được sử dụng để kiểm soát giao diện ngoài HTML cơ bản. Hình ảnh thường được sử dụng để tạo ra sự xuất hiện mong muốn và là một phần của nội dung chính. Âm thanh hoặc video cũng có thể được coi là nội dung "tĩnh" nếu nó phát tự động hoặc nói chung là không tương tác. Loại website này thường hiển thị cùng một thông tin cho tất cả khách truy cập.

*Website động:* website tự động thay đổi hoặc tùy chỉnh thường xuyên và tự động. Các trang động phía máy chủ được tạo "nhanh chóng" bởi [mã máy](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_m%C3%A1y) tính tạo ra HTML (CSS chịu trách nhiệm về giao diện và do đó, là các tệp tĩnh). Có một loạt các hệ thống phần mềm, chẳng hạn như [CGI](https://vi.wikipedia.org/w/index.php?title=Common_Gateway_Interface&action=edit&redlink=1), [Java Servlet](https://vi.wikipedia.org/wiki/Java_Servlet) và [Java Server Pages](https://vi.wikipedia.org/wiki/JSP) (JSP), [Active Server Pages](https://vi.wikipedia.org/wiki/Active_Server_Pages) và [ColdFusion](https://vi.wikipedia.org/w/index.php?title=ColdFusion&action=edit&redlink=1) (CFML) có sẵn để tạo [hệ thống web động và website động](https://vi.wikipedia.org/w/index.php?title=Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh_%C4%91%C6%B0%E1%BB%A3c_s%E1%BB%AD_d%E1%BB%A5ng_trong_h%E1%BA%A7u_h%E1%BA%BFt_c%C3%A1c_website_ph%E1%BB%95_bi%E1%BA%BFn&action=edit&redlink=1). Các [khung ứng dụng web](https://vi.wikipedia.org/w/index.php?title=Khung_%E1%BB%A9ng_d%E1%BB%A5ng_web&action=edit&redlink=1) và [hệ thống mẫu web](https://vi.wikipedia.org/w/index.php?title=H%E1%BB%87_th%E1%BB%91ng_m%E1%BA%ABu_web&action=edit&redlink=1) khác nhau có sẵn cho [các ngôn ngữ lập trình thông](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) dụng như [Perl](https://vi.wikipedia.org/wiki/Perl), [PHP](https://vi.wikipedia.org/wiki/PHP), [Python](https://vi.wikipedia.org/wiki/Python_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) và [Ruby](https://vi.wikipedia.org/wiki/Ruby_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) để giúp tạo các website động phức tạp nhanh hơn và dễ dàng hơn.

**1.1.2. Các thành phần của giao diện website**

Header Header là phần đầu trang mà người dùng sẽ nhìn thấy đầu tiên khi truy cập vào website. Nó thường chứa logo, tên thương hiệu và thanh điều hướng. Thanh điều hướng giúp người dùng dễ dàng chuyển đến các trang khác nhau trên website, tạo điều kiện thuận lợi cho việc tìm kiếm thông tin.

Slider/Carousel : Slider hay Carousel là một phần quan trọng trong giao diện website, cho phép hiển thị hình ảnh hoặc thông điệp nổi bật theo dạng trượt. Thành phần này có thể bao gồm các nút kêu gọi hành động (CTA) hoặc các thanh dẫn hướng người dùng chủ động xem các ảnh tiếp theo về sản phẩm, dịch vụ… Slider hay Carousel giúp thu hút sự chú ý của người dùng ngay từ cái nhìn đầu tiên và thường được sử dụng để quảng bá sản phẩm, dịch vụ hoặc các chương trình khuyến mãi.

Content Area là khu vực chính của website, nơi trình bày nội dung chính như văn bản, hình ảnh, video, link và thông tin liên quan. Nội dung ở đây thường được sắp xếp một cách hợp lý và dễ tiếp cận, nhằm cung cấp thông tin giá trị cho người đọc và thúc đẩy họ tương tác với website.

Sidebar là cột bên cạnh nội dung chính của website, thường chứa các widget, liên kết đến các bài viết liên quan, quảng cáo hoặc thông tin bổ sung. Sidebar giúp người dùng dễ dàng truy cập các thông tin khác mà không cần phải rời khỏi trang hiện tại, giúp tăng cường trải nghiệm người dùng.

Footer Footer nằm ở cuối trang và cung cấp thông tin bổ sung như liên hệ, bản quyền, các liên kết chính sách và điều khoản sử dụng. Nó không chỉ đóng vai trò cung cấp thông tin mà còn giúp người dùng tìm kiếm các liên kết quan trọng khác mà họ có thể cần.

**1.1.3. Tầm quan trọng của website**

Website đóng vai trò thiết yếu trong việc xây dựng và phát triển thương hiệu, đặc biệt trong thời đại số ngày nay. Một website chuyên nghiệp giúp doanh nghiệp tạo dựng ấn tượng đầu tiên mạnh mẽ với khách hàng. Đây là nơi khách hàng có thể tìm hiểu thông tin về sản phẩm, dịch vụ và giá trị của thương hiệu, từ đó nâng cao độ tin cậy và sự uy tín của doanh nghiệp.

Website chính là kênh tiếp cận khách hàng rất hiệu quả. Với một website tối ưu hóa tốt, doanh nghiệp có thể thu hút lượng truy cập lớn thông qua các công cụ tìm kiếm. Điều này giúp mở rộng thị trường và gia tăng cơ hội bán hàng, không chỉ giới hạn trong khu vực địa lý mà còn trên toàn cầu.

Website cho phép doanh nghiệp cung cấp dịch vụ và hỗ trợ khách hàng 24/7. Khách hàng có thể tìm kiếm thông tin, đặt hàng hoặc liên hệ hỗ trợ bất cứ lúc nào, tạo ra sự thuận tiện và nâng cao trải nghiệm người dùng, giúp tăng trưởng doanh số đồng thời tăng cường sự trung thành của khách hàng.

Ngoài ra, website còn là công cụ quan trọng để theo dõi và phân tích hành vi của người dùng. Các công cụ phân tích cho phép doanh nghiệp hiểu rõ hơn về nhu cầu và thói quen của khách hàng, từ đó điều chỉnh chiến lược kinh doanh một cách linh hoạt và hiệu quả hơn.

* 1. **Công nghệ và ngôn ngữ lập trình website**

**1.2.1. Frontend: HTML, CSS, JavaScript**

**1.2.1.1. HTML là gì ?**

HTML là một ngôn ngữ đánh dấu được thiết kế để xây dựng trang web, nó là một ngôn ngữ mở cho phép tất cả các trang web có thể được hiển thị chính xác trên các trình duyệt mà không bị phụ thuộc về bản quyền.

HTML là viết tắt của Hypertext Markup Language , dịch là “ngôn ngữ đánh dấu siêu văn bản”. Trong đó:

+ Hypertext: dịch là siêu văn bản, là loại văn bản tích hợp nhiều dạng dữ liệu khác nhau như văn bản, âm thanh, hình ảnh … và các siêu liên kết (là các đường link) đến các siêu văn bản khác. Người dùng có thể chuyển đến các siêu văn bản khác một cách dễ dàng, do đó nó rất thích hợp khi làm việc với số lượng văn bản lớn, các bộ sách nhiều tập, từ điển bách khoa …

+ Markup Language: dịch là ngôn ngữ đánh dấu, là ngôn ngữ sử dụng các “thẻ” để cấu trúc văn bản (bạn sẽ biết khái niệm thẻ ở phần sau). Qua thời gian dài phát triển HTML hiện cung cấp hơn 80 loại thẻ khác nhau.

Chú ý: HTML không phải ngôn ngữ lập trình mà là một dạng ngôn ngữ để biểu diễn tài liệu web. Mọi trang web có thể được xây dựng bằng các công nghệ phức tạp nhưng cuối cùng đều sẽ trả về trang web dưới dạng HTML để hiển thị tại trình duyệt của người dùng

Một vài khái niệm liên quan:

+ XML: là một ngôn ngữ đánh dấu khác cũng được dùng trong lĩnh vực web, XML khác với html ở chỗ nó không định nghĩa sẵn thẻ nào cả, tất cả các thẻ đều do người dùng định nghĩa

+ XHTML: viết tắt của Extensible HyperText Markup Language , là một ngôn ngữ đánh dấu mở rộng của HTML, người ta xem XHTML là thế hệ kế tiếp của HTML, nó có cấu trúc chặt chẽ hơn, kế thừa XML, vẫn sử dụng được các thẻ của HTML

+ HTML5: là phiên bản mới nhất của HTML (ra đời năm 2008), cung cấp rất nhiều tính năng mới, khả năng hoạt động tốt trên mobile, loại bỏ nhiều thành phần trước đây html không có mà phải dùng của bên thứ 3.

Ở thời kỳ đầu một trang web thường được viết thuần túy bằng HTML, nhưng đến hiện tại, các trang web thường là kết hợp của 3 thành phần: HTML + CSS + Javascript.

**1.2.1.2. Cấu trúc 1 văn bản html**

Ví dụ về 1 văn bản html:

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>This is a Heading</h1>

<p>This is a paragraph.</p>

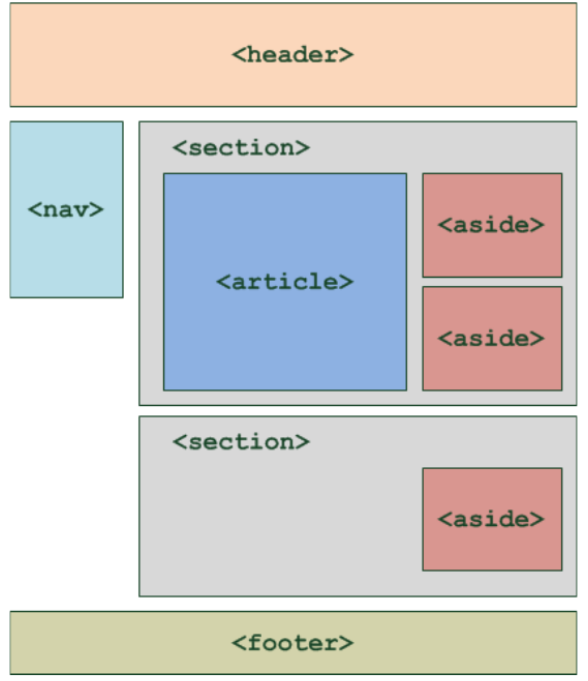
</body> </html>

Văn bản html chứa hàng loạt các thẻ (html element), mỗi thẻ đều có đều được viết trong cặp dấu “< >” và thường đi theo cặp gọi là thẻ mở và thẻ đóng để mô tả phạm vi của thẻ. Ví dụ:

<h1>This is a Heading</h1>

Trong ví dụ trên “h1” là tên thẻ, thẻ mở là <h1>, thẻ đóng là </h1>. Chú ý là trường hợp giữa thẻ đóng và thẻ mở không có nội dung gì thì người ta có thể viết tắt như sau: <br></br> sẽ chuyển thành <br /> mà không gây lỗi ngữ pháp.

Đa số các thẻ html có thể chứa các thẻ khác ở bên trong theo kiểu lồng nhau, chỉ có một vài thẻ không được chứa các thẻ con (cố tình viết các thẻ con vào thì trình duyệt vẫn không hiển thị ra).

Cấu trúc một trang html theo quy phạm cũ thường chia trang web thành nhiều thành phần với các thẻ đặc trưng như <header>, <footer>, <nav>, <section>, <article>, <aside>.

Tuy nhiên, sau này việc bố trí vị trí các thành phần trên trang web thường được làm bằng mã CSS có tính đa dạng và linh hoạt hơn rất nhiều so với việc sử dụng các thẻ html đặc biệt. Vì vậy hiện tại chúng ta có thể coi bố cục một trang html cơ bản đơn giản hơn như sau:

<!Doctype>: Phần khai báo chuẩn của html hay xhtml. phần này không bắt buộc nhưng nên có

<head></head>: Phần khai báo ban đầu, khai báo về meta, title, css, javascript… là các thông tin liên quan đến trang nếu cần có, trường hợp không cần thì thẻ này rỗng cũng được.

<body></body>: Phần chứa nội dung của trang web, nơi hiển thị nội dung.

Chú ý:

Toàn bộ trang html (trừ phần doctype) tài liệu phải được bao bởi cặp thẻ <html> </html>

Các thẻ không nhất thiết phải viết lùi đầu dòng nhưng nên viết lùi đầu dòng thích hợp để dễ nhìn cấu trúc lồng nhau của các thẻ

Trong phần này sẽ chỉ tập trung vào việc nắm các thẻ cơ bản, hay dùng nhất của html. Vấn đề bố cục các khối trên trang sẽ hướng dẫn ở bài CSS sau.

Cách tạo file html: dùng bất kì text editor hỗ trợ viết code tạo một file mới và lưu nó dưới tên file có đuôi .html (bạn có thể dùng notepad, gedit, notepad ++, sublime text ... nhưng không nên dùng Microsoft Word). Sau đó copy đoạn nội dung trong ví dụ trên vào nội dung file, lưu file sau đó mở file này bằng một trình duyệt bất kỳ để xem kết quả.

**1.2.1.3. Phân loại các thẻ HTML**

Tham khảo: https://hocwebchuan.com/tutorial/tut\_html.php

Html hiện cung cấp hơn 80 thẻ khác nhau.

**1.2.1.4. Giới thiệu 1 số thẻ**

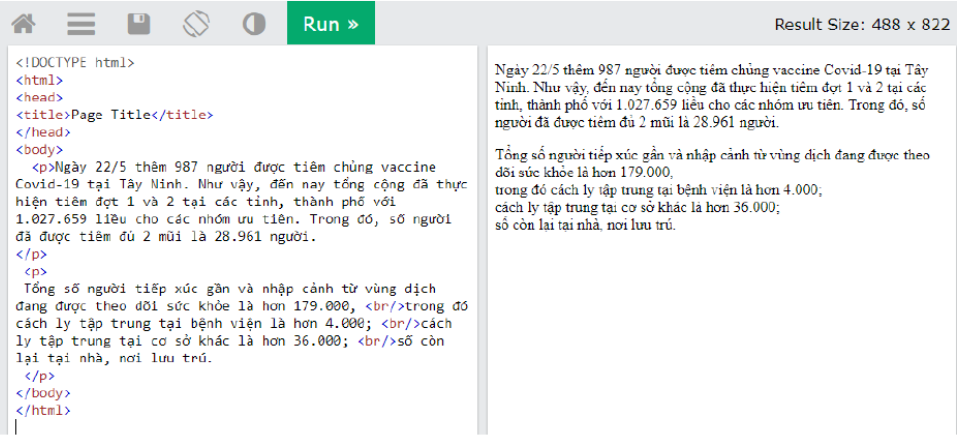
**1.2.1.4.1. Thẻ <p>**

Thẻ <p> là viết tắt của từ “paragraphs” nghĩa là đoạn văn. Khi bắt đầu một đoạn văn thì trình duyệt sẽ hiển thị đoạn văn ở một dòng mới. Việc xuống dòng trong đoạn văn có 3 trường hợp như sau:

Xuống dòng do nội dung bên trong thẻ <p> </p> dài đến kịch bên phải cửa sổ hiển thị thì nội dung còn lại sẽ tự động ngắt dòng và hiển thị tiếp ở dòng dưới

Xuống dòng chủ động trong đoạn văn bằng cách dùng thẻ <br />, thẻ này sẽ khiến văn bản ngắt dòng ngay tại vị trí thẻ <br />. Khoảng cách giữa 2 dòng sẽ tương đương cách ngắt dòng tự động ở trên.

Xuống dòng bằng cách mở đoạn văn mới bằng thẻ <p> khác. Cách này cũng sẽ tạo ra ngắt dòng nhưng khoảng cách giữa 2 dòng sẽ lớn hơn 2 cách ngắt dòng ở trên đó đây là phân biệt giữa 2 đoạn văn khác nhau- Người ta khuyến nghị khi viết mã html thì thẻ <p> chỉ nên chứa các thẻ inline ở bên trong, không nên chứa các thẻ block

Chú ý: việc xuống dòng trong văn bản mã html không làm cho đoạn văn lúc hiển thị trên trình duyệt sẽ xuống dòng, khi hiển thị trên trình duyệt văn bản chỉ xuống dòng theo 3 cách trên.

**1.2.1.4.2 Thẻ <div>**

Thẻ <div></div> viết tắt của từ "division" có nghĩa là phân chia, ta có thể hiểu đây là sự phân chia khu vực hay vùng, sự phân chia này sẽ giúp cho trình duyệt hiểu rõ những vùng được bố cục trên trang web. Thông thường thẻ <div> hay được kết hợp với mã CSS để dàn trang trên trình duyệt.

Về mặt hiển thị, khi không chỉ định CSS thì thẻ div gần giống thẻ <p> nhưng khác ở chỗ việc ngắt dòng do tạo thẻ <div> mới không tạo ra sự giãn dòng nhiều như giãn dòng giữa 2 đoạn văn.

Thẻ <div> có thể chứa hầu hết các thẻ khác ở bên trong trừ <html>, <meta>, <body>, <title>, <link>. Các thẻ <style>, <script> cũng không được khuyến khích viết bên trong thẻ <div>

Thẻ <div> thường dùng để bao trùm các khối nhất định phục vụ mục đích dàn trang cho nên người ta không khuyến khích dùng thẻ <div> cho các mục nhỏ mặc dù về mặt hiển thị trang web vẫn hiển thị tốt.

Ví dụ:

<div>đây là một đoạn văn<div>

Nên viết thành

<p> đây là một đoạn văn </p>

**1.2.1.4.3. Thẻ <ul>,<li>**

Để định nghĩa một danh sách không có thứ tự (các lựa chọn không được đánh số thứ tự) ta dùng thẻ <ul> kết hợp với các thẻ <li>.

<ul> là viết tắt của chữ: unordered list có nghĩa là danh sách không có thứ tự.

<li> viết tắt của chữ: list item có nghĩa là mục của danh sách.

Có thể dùng các thẻ khác kết hợp CSS vẫn cho ra được hiển thị như trên nhưng khuyến khích dùng đúng thẻ sẽ tốt hơn vì nó giúp các trình phân tích ngữ nghĩa trang web hoạt động chính xác, có lợi cho SEO (tối ưu trang web để có thể được hiển thị ưu tiên hơn trên các trang tìm kiếm).

**1.2.1.4.4. Thẻ <ol>,<li>**

Để thể hiện danh sách có thứ tự ta sử dụng cặp thẻ: <ol></ol> và <li></li>, trong đó:

<ol> là viết tắt của chữ "ordered list" có nghĩa là danh sách có thứ tự.

<li> viết tắt của chữ "list item" có nghĩa là mục của danh sách.

Mặc định thì thẻ <ol> sẽ đánh số theo kiểu số bắt đầu từ 1, ta có thể can thiệp cách đánh số bằng cách thêm các “thuộc tính” cho thẻ <ol> như sau:

Reversed: có thuộc tính này thì sẽ đánh số theo chiều ngược lại từ số lớn nhất về bé nhất.

Start: chỉ định số bắt đầu (nếu đánh số theo kiểu số).

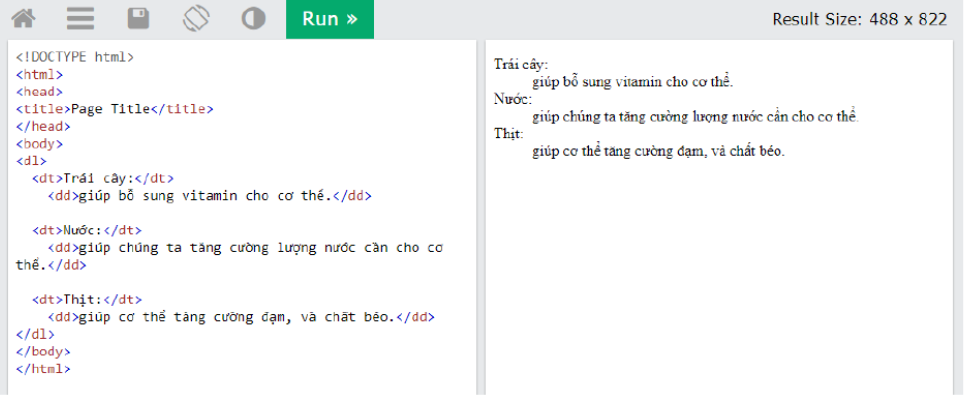
Type: chỉ định cách đánh số, có các giá trị hợp lệ như sau: [ 1, A, a, I, i]

**1.2.1.4.5 Thẻ<dl>,<dt>,<dd>**

Thẻ <dl>, <dt>, <dd> cũng dùng để định nghĩa danh sách nhưng khác ở chỗ mỗi phần tử có thêm phần chú thích. Loại danh sách này thường ít gặp hơn.

<dl> viết tắt của chữ "definition list" có nghĩa là sự xác định (hay định nghĩa) danh sách.

<dt> viết tắt của chữ "defines an item" có nghĩa là xác định (hay định nghĩa) một mục.

<dd> viết tắt của chữ "defines describe an item" có nghĩa là xác định (hay định nghĩa) một mô tả của một mục.

**1.2.1.4.6. Thẻ <nav>**

Thẻ <nav> định nghĩa liên kết điều hướng (navigation).

Thẻ <nav> thường sử dụng cho global menu, local link, topic path, pager link,...

Nội dung bên trong <nav> thường là một danh sách <ul> hoặc <ol> hoặc các thẻ <a>.

**1.2.1.4.7. Thẻ <header>,<main>,<footer>**

Theo quy tắc truyền thống thì một trang web (bên trong thẻ <body>) được phân làm 3 phần là:

<header>: phần đầu trang trang, hiển thị ở phía trên cùng, chứa thông logo công ty, menu chính, giới thiệu chung …

<main>: thẻ main được dùng bao xung quanh phần thân chính của trang web.

<footer>: phần dưới cùng của trang, thường chứa thông tin liên hệ, email, điện thoại …

Cả 3 thẻ trên thực ra không đưa ra định dạng hiển thị gì đặc biệt, nó chỉ như một thẻ div bình thường và thực tế là bạn có thể không tuân theo quy phạm trên vẫn không gây ra lỗi gì. Tuy nhiên người ta vẫn khuyến khích dùng các thẻ trên vì nó có lợi cho các trình phân tích ngữ nghĩa trang web, tối ưu cho SEO và giúp đọc mã dễ dàng hơn.

**1.2.1.4.8. Thẻ <section>**

Tag <section> định nghĩa một khu vực (vùng bao) trong văn bản HTML. Thẻ này thường chứa các thông tin liên quan cùng 1 chủ đề, do đóa thẻ <section> bên trong thường chứa một thẻ tiêu đề <hx> và nội dung.

Về mặt hiển thị thì nó giống như một thẻ div bình thường, việc dùng <section> mà không dùng <div> là lý do về ngữ nghĩa chứ phương diện hiển thị hay ngữ pháp thì không có khác biệt.

**1.2.1.4.9. Thẻ <article>,<aside>**

Giống với thẻ <section>, thẻ <article> được dùng vì mục đích ngữ nghĩa hơn là ngữ pháp và hiển thị. Người ta dùng thẻ này để mô tả một khối nội dung độc lập.

Ví dụ các mẩu tin trên báo, các bình luận của bài viết, các bài đăng trong danh sách bài của forum, …- về mặt hiển thị nó giống một thẻ div.

Thẻ < aside > được đặt ở bên trái hoặc phải hoặc cả 2 bên của phần nội dung chính. Nó thường chứa các thông tin liên quan, các chức năng hay dùng, link đến các sản phẩm khác …. Về mặt hiển thị nó giống một thẻ div.

**1.2.1.4.10. Thẻ <figure>,<figcaption>**

Thẻ <figure> thường dùng để bọc bên ngoài các ảnh đồ thị, sơ đồ, ảnh minh họa … có liên quan đến luồng chính tuy nhiên có tính độc lập nhất định (nếu bị xóa đi thì người đọc vẫn hiểu luồng chính nói gì)

Thẻ < figcaption> thường đi kèm để chú thích cho ảnh.

**1.2.1.5. CSS là gì**

CSS (Cascading Style Sheets) là một ngôn ngữ được sử dụng để định dạng và làm đẹp cho các trang web. Nó giúp chúng ta kiểm soát cách các yếu tố trên trang web được hiển thị, như:

Cấu trúc: Cách các phần tử sắp xếp trên trang.

Font chữ: Loại và kích thước chữ.

Màu sắc: Màu nền, màu chữ, màu viền,...

Bố cục: Cách bố trí các phần tử trên trang.

**1.2.1.6. Tại sao sử dụng CSS?**

Kiểm soát tốt hơn: cho phép kiểm soát chi tiết hơn về giao diện của trang web so với HTML.

Dễ bảo trì: do tất cả các định dạng được tập trung vào một hoặc một số ít tệp CSS.

Truy cập tốt hơn: trang web sử dụng CSS thường thân thiện với người dùng hơn.

Linh hoạt: có thể áp dụng cho nhiều loại tài liệu khác nhau, không chỉ HTML.

Đa nền tảng: cho phép bạn tạo các kiểu hiển thị khác nhau tùy thuộc vào thiết bị (màn hình, máy in, thiết bị di động,...).

**1.2.1.7. Cú pháp CSS**

Được cấu thành từ 3 thành phần chính:

Bộ chọn (selector): Xác định phần tử nào trong HTML sẽ bị ảnh hưởng bởi các định dạng. Ví dụ: body, p, h1, h2,...

Thuộc tính (property): Chỉ ra tính chất định dạng. Ví dụ: color, font-size,text-align,...

Giá trị (value): Giá trị cụ thể cho thuộc tính đó và đi kèm với dấu hai chấm. Vídụ: red, 16px, center.

**1.2.1.8. Khai báo sử dụng CSS trong HTML**

**Inline CSS (CSS nội tuyến)**

Cách thức: Định dạng CSS trực tiếp trong thẻ HTML, sử dụng thuộc tính style.

Ưu điểm: Linh hoạt, áp dụng cho một phần tử cụ thể.

Nhược điểm: Làm cho code HTML trở nên rối mắt, khó quản lý khi có nhiều định dạng.

Ví dụ: HTML<h1 style="color:red; text-align:right;">Khoa THQL</h1>

**Internal CSS (CSS nội bộ)**

Cách thức: Định dạng CSS trong phần <head> của tài liệu HTML, bên trong thẻ <style>.

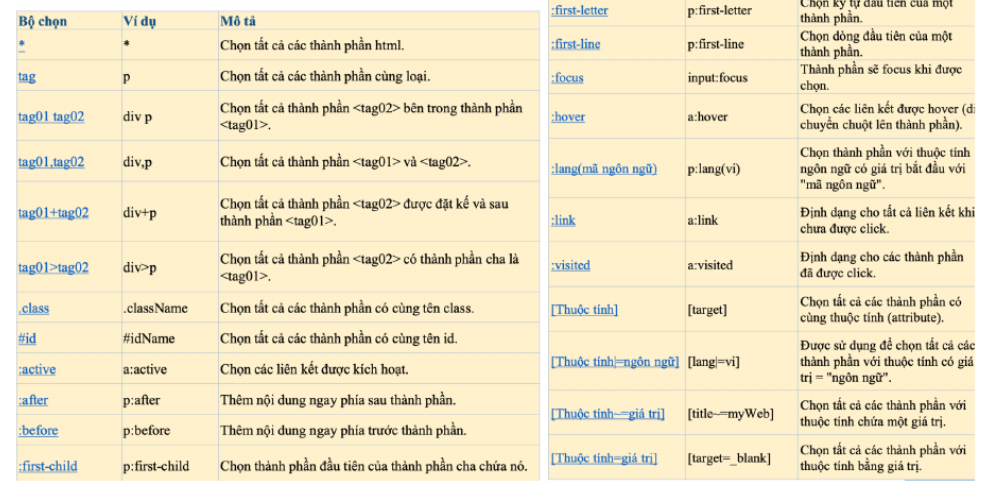
Ưu điểm: Quản lý tập trung hơn inline CSS, áp dụng cho nhiều phần tử trong cùng một trang.

Nhược điểm: Nếu có nhiều trang, phải khai báo lại CSS cho từng trang.

**External CSS (CSS bên ngoài)**

Cách thức: Tạo một file CSS riêng (.css) và liên kết nó vào file HTML bằng thẻ <link>.

Ưu điểm: Tách biệt hoàn toàn phần nội dung và phần định dạng, dễ quản lý, dễ bảo trì, có thể sử dụng chung cho nhiều trang.

Nhược điểm: Cần thêm một bước tạo file CSS và liên kết.

**1.2.1.9. Giá trị trong CSS:**

Tất cả các giá trị trong CSS đều được xem là chuỗi: Ngay cả các giá trị số như 14 px cũng được hiểu là một chuỗi đại diện cho kích thước 14 pixel.

Màu sắc:

* Được định nghĩa theo mô hình màu RGB (đỏ, xanh lá, xanh dương).
* Có thể sử dụng định dạng hex (ví dụ: #44f1e1) hoặc định dạng thập phân (ví dụ: rgb(68, 241, 255)).

Các giá trị kích thước:

* Khi đặt các giá trị cho chiều rộng, chiều cao, cỡ chữ,... có thể sử dụng nhiều đơn vị đo khác nhau:
* Pixel (px): Đơn vị đo tuyệt đối, dựa trên độ phân giải màn hình.
* Em: Đơn vị tương đối, dựa trên kích thước font chữ của phần tử cha.
* Rem: Tương tự em, nhưng dựa trên kích thước font chữ của gốc tài liệu (root element).

Pt, pc, in, cm, mm: Các đơn vị đo in ấn.

%: Phần trăm, so với kích thước của vùng chứa hoặc cỡ chữ.○Số 0: Có thể sử dụng mà không cần đơn vị, ví dụ: border: 0;.

**1.2.1.10. Các thuộc tính cơ bản trong CSS**

**1.2.1.10.1. Định dạng văn bản**

font-size: kích thước font chữ. Có thể sử dụng các giá trị tuyệt đối (px), tương đối (em, rem) hoặc các giá trị từ nhỏ đến lớn (xx-small, x-small, small, medium, large, x-large, xx-large).

font-family: font chữ sử dụng. Có thể liệt kê nhiều font, trình duyệt sẽ chọn font đầu tiên có sẵn.

font-weight: độ đậm nhạt của font chữ. Có thể sử dụng các từ khóa (normal, bold, bolder, lighter) hoặc giá trị số từ 100 đến 900.9.

*Các thuộc tính khác để định dạng văn bản:*

font-style: Tạo kiểu cho font chữ (normal, italic, oblique).

text-decoration: hiệu ứng trang trí cho văn bản (none, underline, line-through,overline, blink).

text-align: căn chỉnh văn bản (left, right, center, justify).

**1.2.1.10.2.Thuộc tính overflow**

Cách xử lý nội dung khi nó vượt quá kích thước của phần tử chứa.

Các giá trị:

* visible: Nội dung bị tràn sẽ hiển thị ra ngoài.
* hidden: Nội dung bị tràn sẽ bị ẩn đi.
* scroll: Hiển thị thanh cuộn khi nội dung bị tràn.
* auto: Tự động hiển thị thanh cuộn nếu nội dung bị tràn.

**1.2.1.10.3.Thuộc tính text-overflow**:

Chức năng: Xác định cách xử lý khi văn bản tràn ra ngoài phần tử chứa.

Cú pháp: text-overflow: <value>;

Các giá trị:○ellipsis: Hiển thị dấu ba chấm (...) để biểu thị phần văn bản bị cắt =>cách phổ biến nhất○clip: Cắt bỏ phần văn bản thừa, không hiển thị gì cả. Đây là giá trị mặc định.

**1.2.1.10.4.Thuộc tính word-wrap:**

Chức năng: Cho phép các từ dài được ngắt dòng để tránh bị tràn ra ngoài phần tử chứa.

Cú pháp: word-wrap: <value>;

Các giá trị:

* normal: Các từ dài sẽ không bị ngắt, có thể gây ra tình trạng tràn dòng.
* break-word: Cho phép các từ dài được ngắt ở giữa để tránh tràn dòng.

**1.2.1.10.5. Thuộc tính width và height**

Chức năng: Xác định chiều rộng và chiều cao của một phần tử.

Các giá trị:

* Số: Có thể là số điểm ảnh (px), centimet (cm), em (tương đối với kích thước font chữ của phần tử cha) hoặc phần trăm (so với kích thước của phần tử cha).

- min-width, max-width, min-height, max-height: Giới hạn chiều rộng và chiều cao tối thiểu và tối đa của phần tử.

**1.2.1.10.6.Thuộc tính display**:

Chức năng: Kiểm soát cách hiển thị của một phần tử, nó có tạo ra một vùng hộp chứa hay không, cách các phần tử khác tương tác…

Các giá trị:

* none: Phần tử không được hiển thị và không chiếm không gian.
* inline: Phần tử được hiển thị nội tuyến, không có ngắt dòng trước và sau.
* block: Phần tử được hiển thị như một khối, có ngắt dòng trước và sau.
* inline-block: Kết hợp giữa inline và block.
* list-item: Hiển thị phần tử như một mục trong danh sách.
* table, table-row, table-cell: Sử dụng cho các phần tử bảng.

**1.2.1.10.7.Margin (Lề) và Padding (Đệm)**

Margin: Tạo khoảng cách giữa phần tử hiện tại và các phần tử khác xung quanh nó.

Padding: Tạo khoảng cách giữa nội dung và viền Các thuộc tính margin

margin-top, margin-right, margin-bottom, margin-left: Điều chỉnh lề tương ứng cho các cạnh trên, phải, dưới và trái của phần tử.

Các giá trị:○Độ dài: Có thể là số âm (ví dụ: 1cm, -5pt).○Phần trăm: Tham chiếu đến chiều rộng của khối chứa (ví dụ: 10%).○auto: Trình duyệt tự động quyết định.Cách viết tắt cho margin Để đặt giá trị cho cả bốn cạnh của một phần tử, bạn có thể sử dụng các cách viết tắt sau:

margin: 5px; : Đặt tất cả bốn cạnh có lề 5px.

margin: 10px 20px; : Trên và dưới là 10px, trái và phải là 20px.

margin: 5px 3px 8px; : Trên là 5px, trái và phải là 3px, dưới là 8px.

margin: 1px 3px 5px 7px; : Theo thứ tự trên, phải, dưới, trái.

**1.2.1.10.8.Thuộc tính position**

Position: relative

●Chức năng: Định vị một phần tử tương đối so với vị trí mặc định của nó.

●Hoạt động: Phần tử dịch chuyển theo các giá trị top, bottom, left, right mà bạn khai báo. Các giá trị này sẽ xác định khoảng cách phần tử di chuyển so với vị trí ban đầu của nó.Position: fixed

●Chức năng: Định vị một phần tử cố định trên màn hình.

●Hoạt động: Phần tử sẽ luôn ở một vị trí cố định trên màn hình, bất kể người dùng có cuộn trang hay thay đổi kích thước cửa sổ trình duyệt.

**1.2.1.10.9.Thuộc tính float**

Để điều chỉnh vị trí của một phần tử so với các phần tử khác xung quanh nó trong dòng Các giá trị chính của thuộc tính float:

●none: hiển thị ở vị trí mặc định của nó trong dòng văn bản

●left: di chuyển sang bên trái của phần tử cha cho đến khi gặp một phần tử khác hoặc đến cuối dòng.

●right: di chuyển sang bên phải của phần tử cha cho đến khi gặp một phần tử khác hoặc đến cuối dòng.

**1.2.1.10.10.Thuộc tính clear**

●Vấn đề: Khi áp dụng thuộc tính float cho một phần tử, các phần tử tiếp theo trong HTML thường sẽ "trôi" lên phía trên phần tử nổi đó.

●Giải pháp: Sử dụng thuộc tính clear để ngăn chặn các phần tử khác bị trôi lên.

●Cách hoạt động:

clear: left: Phần tử hiện tại sẽ không xuất hiện bên trái của phần tử nổi.

clear: right: Phần tử hiện tại sẽ không xuất hiện bên phải của phần tử nổi.

clear: both: Phần tử hiện tại sẽ không xuất hiện ở cả bên trái và bên phải của phần tử nổi.

**1.2.1.10.11.Thuộc tính z-index:**

●Vấn đề: Khi nhiều phần tử chồng lên nhau, trình duyệt sẽ hiển thị phần tử nào trước?

●Giải pháp: Sử dụng thuộc tính z-index để xác định thứ tự chồng xếp của các phần tử.

●Cách hoạt động:

z-index: Giá trị càng cao, phần tử càng ở trên.

z-index: auto: Giá trị mặc định, phần tử sẽ được xếp theo thứ tự xuất hiện trong mã HTML.

z-index: negative: Phần tử sẽ được xếp phía sau các phần tử khác.

**1.2.1.10.12.CSS Effects**

●CSS Effects tập hợp các kỹ thuật trong CSS tạo ra các hiệu ứng động và tương tác trên các trang web Transforms

●Transforms để thay đổi kích thước, vị trí và hình dạng của các phần tử HTML.

●Các hiệu ứng biến đổi: di chuyển, xoay, biến đổi kích thước, biến đổi 3D.

●Các hàm biến đổi:

Perspective(): Tạo hiệu ứng phối cảnh 3D.

Rotate3d(): Xoay một phần tử quanh một trục 3D.

Scale3d(): Thay đổi kích thước một phần tử theo 3 chiều.

Translate3d(): Di chuyển một phần tử theo 3 chiều.Filters (Bộ lọc)

●Chức năng: Thay đổi cách hiển thị của hình ảnh

●Cách hoạt động: Áp dụng các hàm lọc lên hình ảnh để tạo ra các hiệu ứng như làm mờ, thay đổi màu sắc, độ tương phản, v.v.

●#someImage { filter: grayscale(100%); }Transitions (Chuyển đổi)

●Chức năng: Tạo ra các hiệu ứng chuyển đổi mượt mà giữa các trạng thái của một phần tử.

●Các thuộc tính:

transition-delay: Thời gian trì hoãn trước khi hiệu ứng bắt đầu.

transition-duration: Thời gian kéo dài của hiệu ứng.

transition-property: Thuộc tính CSS mà hiệu ứng sẽ tác động lên.

transition-timing-function: Hàm xác định cách thức thay đổi của thuộc tính trong quá trình chuyển đổi.

**Animations**: Cũng tạo ra các chuyển đổi giữa các trạng thái, cho phép định nghĩa chi tiết các khung hình trung gian => tạo ra các hiệu ứng động phức tạp hơn và có thể tùy chỉnh

Các thuộc tính chính của Animations:

●animation-iteration-count: Số lần lặp lại của hiệu ứng.

●animation-name: Tên của hiệu ứng, được định nghĩa trước đó bằng@keyframes.

●animation-play-state: Khởi động hoặc tạm dừng hiệu ứng.

●animation-duration: Thời gian kéo dài của một chu kỳ hiệu ứng.

●animation-timing-function: Hàm xác định cách thức thay đổi của thuộc tính trong quá trình chuyển đổi.

●animation-delay: Thời gian trì hoãn trước khi hiệu ứng bắt đầu.

**1.2.1.11. JavaScript**

JavaScript là một ngôn ngữ chương trình thông dịch, nhẹ. Nó được thiết kế để tạo các ứng dụng mạng trung tâm. Nó là ngôn ngữ tích hợp với Java, được lấy tên là JavaScript vì thời đó Java là một hiện tượng, nhưng trên thực tế JavaScript không được phát triển dựa trên Java. JavaScript rất dễ dàng để thực hiện bởi vì nó được tích hợp với HTML. Đây là ngôn ngữ chương trình mở và đa nền tảng.

JavaScript là một ngôn ngữ lập trình của HTML và WEB. Nó là nhẹ và được sử dụng phổ biến nhất như là một phần của các trang web, mà sự thi hành của chúng cho phép Client-Side script tương tác với người sử dụng và tạo các trang web động. Nó là một ngôn ngữ chương trình thông dịch với các khả năng hướng đối tượng.

Các file JavaScript thường được lưu dưới đuôi js và được nhúng vào các trang HTML. Do đó, để hiểu tác dụng của JavaScript thì điều kiện tiền đề là bạn cần phải biết cách HTML hoạt động. Nếu bạn còn cảm thấy mới mẻ với HTML, bạn có thể tham khảo tham khảo loạt bài hướng dẫn HTML cơ bản và nâng cao ngay trên trang của chúng mình.

**1.3. Quy trình phát triển website**

Quy trình phát triển website bao gồm các bước cơ bản:

* Lên ý tưởng và phân tích yêu cầu: Xác định mục tiêu, đối tượng sử dụng, và tính năng cần thiết.
* Thiết kế giao diện (UI/UX): Phác thảo bố cục và thiết kế trải nghiệm người dùng.
* Phát triển website: Lập trình Frontend và Backend dựa trên các thiết kế đã thống nhất.
* Kiểm thử: Thực hiện kiểm tra chức năng, bảo mật, hiệu suất, và khả năng tương thích.
* Triển khai: Đưa website lên máy chủ và cấu hình tên miền.
* Bảo trì và cập nhật: Theo dõi, khắc phục lỗi và cập nhật nội dung định kỳ.

**1.4.Bảo mật trong phát triển web**

Đảm bảo an ninh là yếu tố quan trọng trong phát triển web, gồm:

* Bảo vệ dữ liệu người dùng: Thông qua mã hóa (HTTPS, SSL) và xác thực đa yếu tố.
* Ngăn ngừa tấn công: Bảo vệ trước các lỗ hổng như SQL Injection, Cross-Site Scripting (XSS).
* Quản lý quyền truy cập: Hạn chế truy cập dữ liệu nhạy cảm và thiết lập thời gian hết hạn cho phiên đăng nhập.
* Sao lưu dữ liệu: Tạo các bản sao lưu thường xuyên để khôi phục khi cần thiết

**1.5.Công cụ kiểm thử và tối ưu web**

Các công cụ thường dùng:

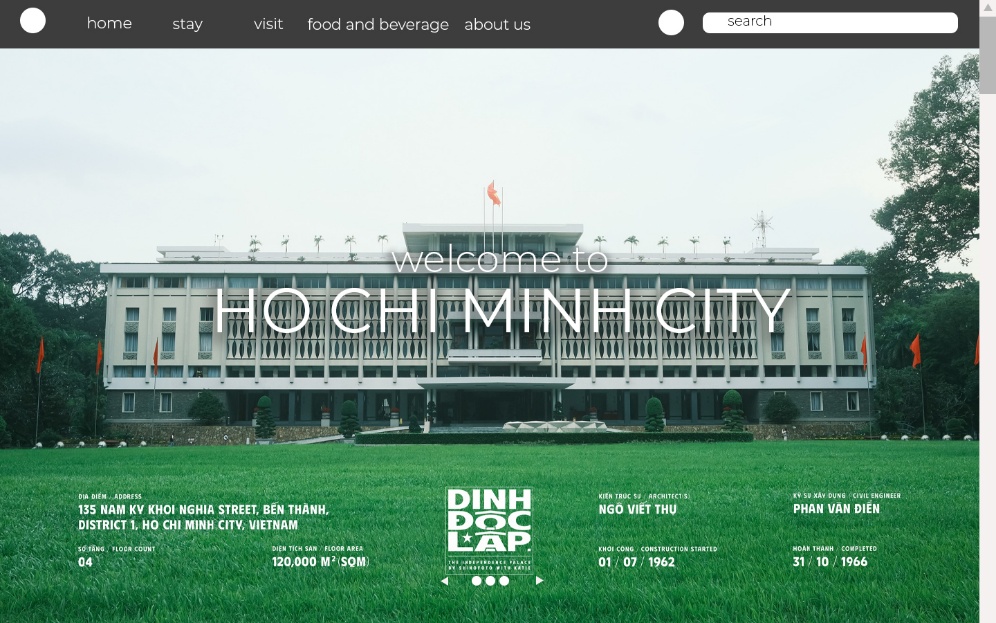
* Kiểm thử tự động: Selenium, Katalon Studio.
* Kiểm tra hiệu suất: Google PageSpeed Insights, GTmetrix.
* Kiểm tra bảo mật: Burp Suite, OWASP ZAP.
* Phân tích dữ liệu: Google Analytics để theo dõi hiệu suất và hành vi người dùng.

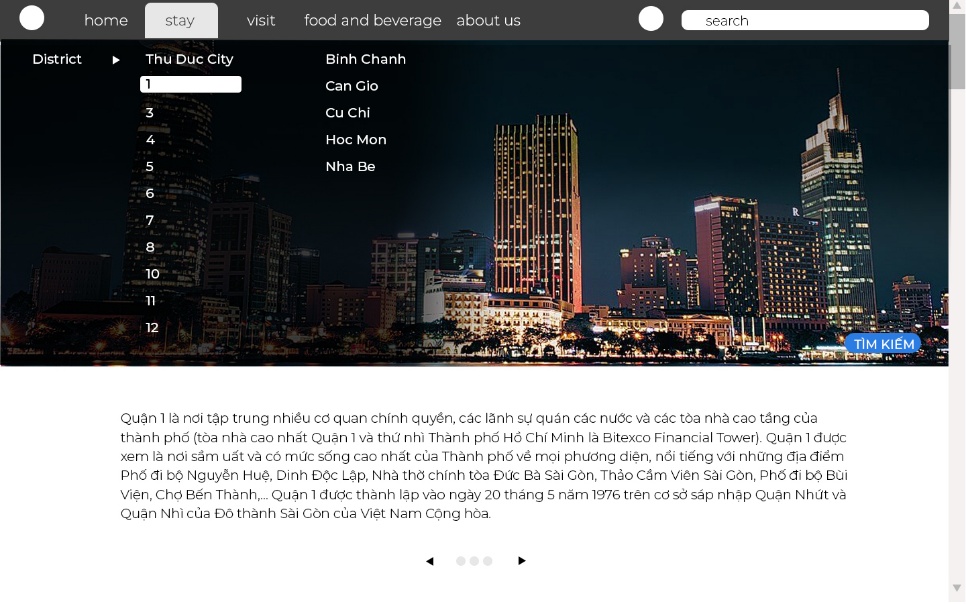
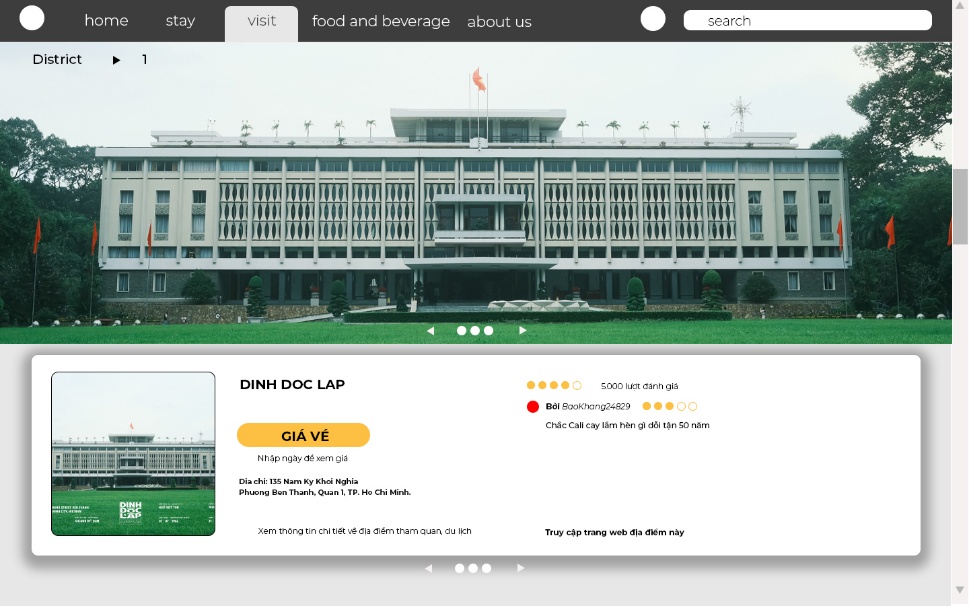
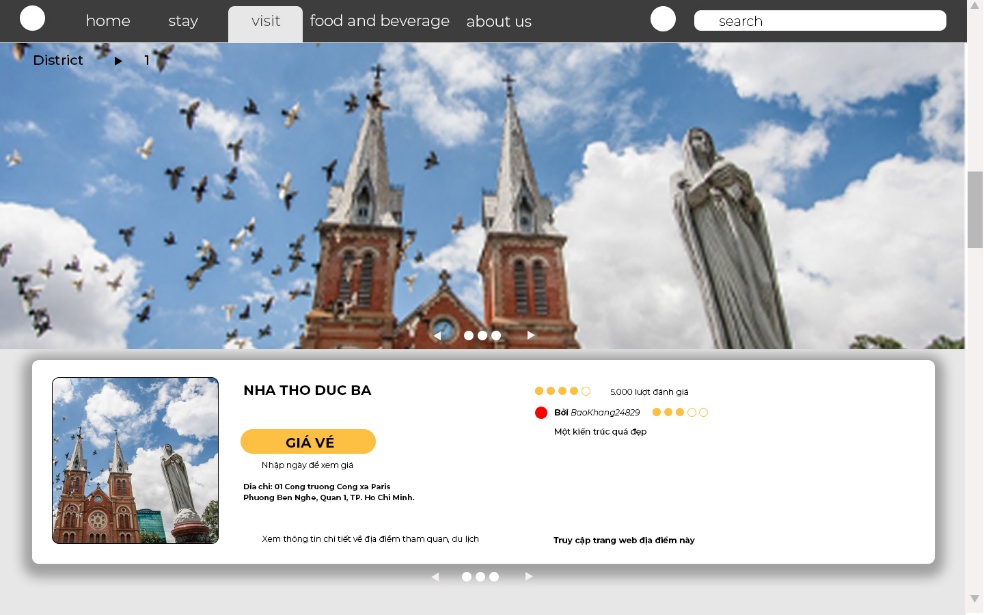
1. **ỨNG DỤNG**

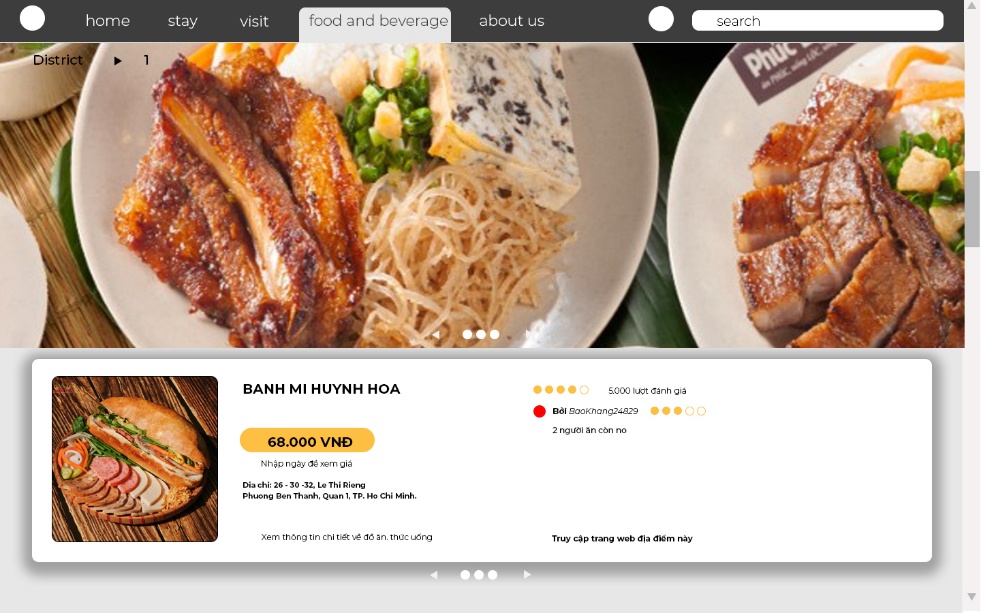
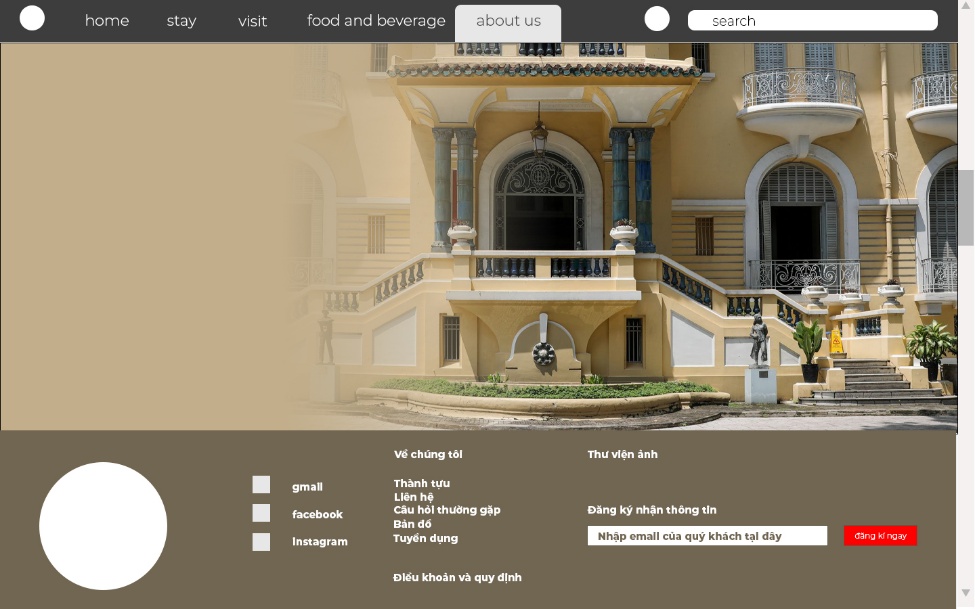
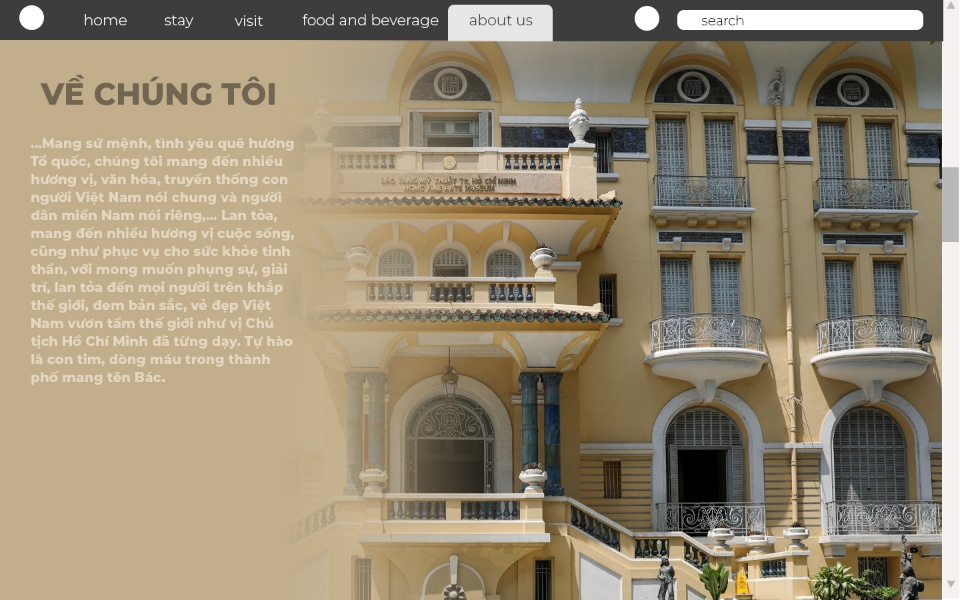
## Lên ý tưởng

Ban đầu, khi nhận đề tài, thì rất có nhiều ý tưởng được đặt ra, nhưng cuối cùng chốt đề tài du lịch, và bắt đầu xây dựng phác thảo đề tài thông qua các phần mềm thứ 3 hỗ trợ sáng tạo, thiết kế. Cũng như qua các cuộc họp offline và online đánh giá tính khách quan, tính khả thi của dự án, cũng như khả năng hoàn thiện dự án trong thời gian ngắn

### Thiết kế ý tưởng thông qua phần mềm thứ 3 (Adobe Photoshop)

Trải qua đóng góp chỉnh sửa, bản thiết kế cơ bản hoàn thành, dưới đây là một số thiết kế ban đầu của trang web:





## Phát triển ý tưởng

Ý tưởng ban đầu được nhóm phác thảo thông qua các bản thiết kế giao diện trên Photoshop, từ đó hình thành bố cục trực quan và thể hiện rõ nét mục tiêu mà sản phẩm hướng đến. Sau giai đoạn này, nhóm bắt đầu chuyển hóa các ý tưởng thành mã front-end bằng cách sử dụng HTML, CSS, và JavaScript để tạo ra một giao diện tương tác tinh gọn, trực quan, đồng thời đảm bảo sự nhất quán về màu sắc, bố cục, và trải nghiệm người dùng. Tiếp theo, nhóm nỗ lực tìm hiểu, nghiên cứu thêm nhiều tài liệu, khóa học và hướng dẫn trực tuyến để từng bước bổ sung mã back-end, nâng cao tính linh hoạt cũng như hiệu năng cho sản phẩm, giúp trang web không chỉ đẹp về mặt hình thức mà còn hoạt động trơn tru, đáp ứng nhanh các yêu cầu từ phía người dùng. Trong quá trình đó, nhờ sự chia sẻ kiến thức giữa các thành viên và việc không ngừng học hỏi từ nhiều nguồn, nhóm dần nhận thấy tiềm năng cải thiện hơn nữa giao diện đã có, làm cho sản phẩm hoàn thiện, bóng bẩy, dễ sử dụng, đồng thời mở rộng khả năng ứng dụng cho nhiều mục đích khác nhau.

## Cấu hình và kiểm thử

Về phần giao diện được thiết kế bởi các mã front-end đã có thể hoạt động trực quan và các tính năng được cấu hình lên từ ý tưởng cũng đã dần hoàn thiện. Nhóm tối ưu hiệu năng, cải thiện tương tác, bổ sung hiệu ứng chuyển cảnh mượt mà, đồng thời tiến hành kiểm thử kiểm thử trang web nhiều lần.

#### Cài đặt ban đầu

Cài đặt ban đầu của trang web khá cơ bản, tuy nhiên, sau quá trình học hỏi và phát triển liên tục, giao diện và hiệu năng đã được cải thiện đáng kể, mang lại trải nghiệm mượt mà hơn cho người dùng.

# 

# PHẦN KẾT LUẬN

* 1. **Kết quả đạt được**
* Tạo ra được mã front-end có giao diện trực quan và bắt mắt.
* Đưa được trang web lên internet thông qua nền tảng GITHUB.
* Trang web có thể được cập nhật nhờ những người được cấp quyền cộng tác viên dự án.

## Thuận lợi

* Dễ dàng trong việc xây dựng các mã front-end
* Lượng kiến thức về các ngôn ngữ như HTML, CSS, JavaScript không quá nhiều nên việc vừa học và vừa thực hành không quá khó.
* Các nguồn kiến thức đa dạng phong phú.
* Có nhiều nguồn tham khảo làm tăng tính phong phú và thẩm mĩ của trang web.

## Khó khăn

* Bị sức ép về mặt thời gian khi bắt đầu lấn sang học về cơ sở dữ liệu.
* Các kiến thức về cơ sở dữ liệu quá rộng khiến cho việc học khó nắm bắt đúng hướng phát triển và tốn quá nhiều thời gian.

## Nhận xét và đánh giá tổng quan

Gia Kiệt nhận xét đánh giá tổng quan:

Tự cá nhân đánh giá bản thân: Bản thân có chủ động làm việc nhóm, lắng nghe chia sẻ ý kiến đến mọi người, mong muốn qua dự án đầu tay, định hướng hướng đi làm việc nhóm cho mọi người, chịu tìm tòi, học hỏi những cái mới, thế mạnh của bản thân là thẩm mỹ, nghệ thuật, sáng tạo, nên có giúp mọi người khái quát chân thật về đề tài, nhưng sáng tạo nghệ thuật là một chuyện, đến khi thành viên nhóm thực hiện đề tài thì lại câu chuyện khác cảm thấy khá khó so với mọi người. Bên cạnh đó đôi khi bản thân còn trì trệ việc nhóm từ các yếu tố tác động bên ngoài, năng lực bản thân vẫn chưa hoàn thành tốt những ý kiến, mục tiêu ban đầu đặt ra. Đánh giá phiến diện đề tại đang thực hiện một cách chủ quan

Đánh giá chung cả nhóm: Cả nhóm đã phối hợp, thực hiện một cách rất chặt chẽ, hoàn thành rất tốt nội dung phần mục của mình đưa ra, một số thành viên nhóm có một số vấn đề riêng về sức khỏe,... hoặc các yếu tố khách quan khác nhưng vẫn cố gắng hoàn thành rất tốt công việc của mình không đùn đẩy cho ai, điều đó em rất đánh giá cao tinh thần cố gắng, học tập vượt khó của các bạn, không để việc riêng ảnh hưởng việc chung cả nhóm. Hoặc có thành viên có nghiên cứu sâu thử các tài liệu về dữ liệu để tối ưu đề tài phát triển dự án hơn nữa, nhưng nỗ lực không đạt như mong muốn, nếu có thời gian các bạn sẽ phát huy rất tốt điều này, rất trân trọng sự đóng góp của các bạn vào dự án đầu tay, cũng như khả năng vận dụng tốt kiến thức nghiên cứu, lần đầu thực hiện khía cạnh mới.

Ngọc Hải nhận xét đánh giá tổng quan:

Tự cá nhân đánh giá bản thân: Bản thân chủ động và có trách nhiệm trong các công việc nhóm, chịu khó tìm tòi, học hỏi các mảng kiến thức mới lạ. Nỗ lực trong các thử nghiệm tạo ra các mã front-end thông qua các cơ sở dữ liệu như MySQL, MongoDB, WordPress,... Một nỗ lực khác là việc đẩy trang web này lên trên internet, bản thân đã cố gắng thử nhiều cách như lấy mã nguồn tạm thời thông qua nền tảng Google Cloud SDK nhưng nhận thấy tính thiếu linh hoạt nên đã loại bỏ. Bản thân vẫn tiếp tục nỗ lực và đã tìm ra được một giải pháp có thể khắc phục cả 2 nhược điểm trên đó là đẩy dự án lên GITHUB. Song song với các nỗ lực đó thì bản thân vẫn còn các hạn chế như chưa có khả năng quản lí thời gian, thiếu các kĩ năng trong việc tiếp thu tài liệu trên mạng.

Đánh giá chung cả nhóm: Các thành viên trong nhóm làm việc rất tốt, mọi người đã hoàn thành các công việc được giao, trong quá trình làm việc cũng có 1 số khó khăn nhất định khiến công việc bị đẩy lùi nhưng không vì đó mà các thành viên bỏ cuộc hay thiếu trách nhiệm với phần công việc mà bản thân được giao. Có thể một số dự định về trang Web chưa được thực hiện do áp lực thời gian và lượng kiến thức khổng lồ. Nhưng nhìn chung các thành viên cũng đã rất nỗ lực, rất trân trọng sự đóng góp và có trách nhiệm của mỗi thành viên trong nhóm mặc dù đây là một lĩnh vực hoàn toàn mới.

Đăk Lộc nhận xét đánh giá tổng quan:

Tự cá nhân đánh giá bản thân: Bản thân có chủ động tìm hiểu học hỏi về đồ án này, chủ động làm việc nhóm, dù có chút khó khăn trong quá trình tiếp thu kiến thức một cách đột ngột và nhiều thứ cùng lúc, với cả việc sức khỏe không ổn định dù nhập viện nhưng bản thân đã cố hết sức để hoàn thành nhiệm vụ của bản thân và giúp đỡ nhóm để hoàn thành đồ án này. Đã bỏ công sức ra để học backend đặc biệt là MySQL,... nhưng do không có thời gian và còn thiếu nhiều kinh nghiệm nên chưa làm được gì nhiều.Song thấy bản thân còn yếu kém trong nhiều việc.

Đánh giá chung cả nhóm: Tất cả các thành viên nhóm làm rất tốt nhiệm vụ được giao, cả team đã cố gắng hết sức để hoàn thành.

Bảo Khang nhận xét đánh giá tổng quan:

Tự cá nhân đánh giá bản thân: Thời gian đầu em có những việc riêng từ gia đình và những dự định trong tương lai nên em đã bị xao nhãng và đã không chú tâm vào dự án của cả nhóm, nhưng nhờ các bạn ủng hộ và góp ý nên em đã hoàn thành phần việc của mình và đã tự vực dậy bản thân và kết quả là em đã không làm ảnh hưởng đến tiến độ của cả nhóm. Qua quãng thời gian trên em đã học hỏi rất nhiều và phần mà em khó học nhất đó là phần Javascript. Bản thân cảm thấy vẫn chưa đủ và cần học hỏi nhiều hơn nữa để hoàn thiện bản thân hơn. Cần bức phá bản thân để bước tới một cấp bậc mới trong ngành mang tính đào thải cao này.

Đánh giá chung cả nhóm: Cả nhóm đã rất đoàn kết và có năng lực, dù có bạn bị vấn đề về sức khỏe nhưng bạn vẫn có trách nhiệm với cả nhóm và đã cùng nhau cố gắng để cho ra sản phẩm. Tuy sản phẩm vẫn chưa được như ý mong muốn nhưng đây là nỗ lực, là những đêm thức trắng của cả nhóm, một điều quả thật rất đáng quý. Từng thành viên đều có trách nhiệm với nhau và đã ngồi lại cùng nhau nhiều ngày để bàn và họp. Dự án này đều là dự án đầu tiên của từng cá nhân nên em rất trân trọng sự nỗ lực của các bạn.

Phúc Thịnh nhận xét đánh giá tổng quan:

Tự cá nhân đánh giá bản thân: Đây được xem như là dự án đầu tiên nên em dành nhiều tâm huyết cho nó. Bản thân em còn yếu trong việc code và còn gặp nhiều khó khăn nhưng nhờ có sự hỗ trợ, góp ý của các bạn giúp em hoàn thiện. Qua dự án này, em học được thêm nhiều về HTML,CSS và em sẽ cố gắng phát triển bản thân để những dự án sau hoàn thiện hơn nữa.

Đánh giá chung cả nhóm: Cả nhóm đoàn kết, lắng nghe, giúp đỡ nhau và mỗi người đều có trách nhiệm, ý thức trong công việc. Dù chưa có kinh nghiệm về làm web nhưng mọi người đã dành ra nhiều thời gian, công sức để thực hiện cũng như góp ý cho nhau. Đây là một trong những dự án lớn đầu tiên cả nhóm không thể tránh những sai sót. Nhìn chung mọi người đã cố gắng tìm tòi học rất nhiều và cố gắng hoàn thành tốt nhất mục tiêu đề ra.

## Đánh giá mức độ hoàn thành của thành viên

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **HỌ VÀ TÊN** | **MSSV** | **MỨC ĐỘ HOÀN THÀNH** |
| 1 | Lê Ngọc Hải | 24110089 | 100% |
| 2 | Nguyễn Quốc Bảo Khang | 24110099 | 100% |
| 3 | Trần Gia Kiệt | 24110103 | 100% |
| 4 | Nguyễn Đăk Lộc | 24110107 | 100% |
| 5 | Đặng Tấn Phúc Thịnh | 24110135 | 100% |

# TÀI LIỆU THAM KHẢO

1.<https://vi.wikipedia.org/wiki/Website>

2.BKWEB:<https://www.bkweb.vn/website-la-gi/#Phan_biet_website_va_trang_web>

3.STUDOCU:

<https://www.studocu.com/vn/document/truong-dai-hoc-thang-long/mang-may-tinh/ly-thuyet-html/94798108?origin=home-recent-1>

<https://www.studocu.com/vn/document/truong-dai-hoc-kinh-te-thanh-pho-ho-chi-minh/kiem-toan/buoi-3-notebook/112458940>

4.

<https://hub.whitehub.net/cac-quy-trinh-chuan-kiem-tra-mot-ung-dung-website/>

https://daotaotester.vn/testing-website/

https://daotaotester.vn/testing-website/