

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

Ảnh có chứa biểu tượng, Phông chữ, Đồ họa, Xanh điện

Mô tả được tạo tự động

BÀI TẬP LỚN

*MÔN HỌC :* ***Công nghệ WEB***

*ĐỀ TÀI:* **Xây dựng Web bán sách**

Giảng viên hướng dẫn: **GV. Nguyễn Thị Hạnh**

Nhóm thực hiện: **Nhóm 2.**

Ảnh có chứa ảnh chụp màn hình, ánh sáng, công nghệ

Mô tả được tạo tự độngẢnh có chứa ảnh chụp màn hình, vòng tròn, thiết kế, tác phẩm nghệ thuật

Mô tả được tạo tự độngLớp: **20CN5**

Thành viên: **2055010054 - Trịnh Đình Đại,**

**2055010168 - Hoàng Kông Minh,**

**2055010168 – Bùi Đức Năng**

*Hà Nội, Ngày … tháng 9 năm 2023.*

**

Mục lục

[**1.** **Giới thiệu về HTML** 3](#_Toc146352723)

[**1.1 HTML Cơ bản** 3](#_Toc146352724)

[**1.2** **Hướng dẫn tạo một tệp HTML cơ bản:** 3](#_Toc146352725)

[**1.3** **Hướng dẫn cú pháp và các phần tử HTML cơ bản:** 3](#_Toc146352726)

[**1.3.1** **Các phần tử chính của phần tử như sau** 4](#_Toc146352727)

[**1.3.2** **Các phần tử có thể có các thuộc tính giống như sau** 4](#_Toc146352728)

[**1.3.3 Phần tử trống** 4](#_Toc146352729)

[**1.3.4 Giải phẫu tài liệu HTML** 5](#_Toc146352730)

[**1.4 Tạo trang Web với tiêu đề, đoạn văn bản, hình ảnh và siêu liên kết** 6](#_Toc146352731)

[**2.** **Cơ bản về CSS** 9](#_Toc146352732)

[**2.1 CSS – Thêm kiểu cho web** 9](#_Toc146352733)

[**2.1.1 Inline CSS (CSS trực tiếp trong phần tử HTML – Nội tuyến)** 9](#_Toc146352734)

[**2.1.2 Internal CSS (CSS trong tệp HTML – Nội bộ)** 9](#_Toc146352735)

[**2.1.3 External CSS (CSS bên ngoài)** 10](#_Toc146352736)

[**2.2 Tạo các lớp và áp dụng kiểu bằng cách sử dụng lựa chọn** 11](#_Toc146352737)

[**2.3. Thử nghiệm với các thuộc tính CSS cơ bản như font chữ, màu sắc, độ rộng và chiều cao** 12](#_Toc146352738)

[**2.3.1. Font chữ** 12](#_Toc146352739)

[**2.3.2. Màu sắc** 12](#_Toc146352740)

[**2.3.3. Độ rộng và chiều cao** 12](#_Toc146352741)

[**3.** **JavaScript cơ bản** 14](#_Toc146352742)

[**3.1** **Giới thiệu cú pháp JavaScript cơ bản** 14](#_Toc146352743)

[**3.2** **Tạo biến, hàm và sử dụng sự kiện JavaScript** 14](#_Toc146352744)

[**3.2.1** **Tạo biến (Variables)** 14](#_Toc146352745)

[**3.2.3** **Sử dụng sự kiện (Event)** 15](#_Toc146352746)

[**3.3. Tạo các ví dụ đơn giản như hiển thị thông báo khi nút được nhấn** 16](#_Toc146352747)

[**4.** **HTML Forms và Input Elements** 17](#_Toc146352748)

[**4.1 Tạo biểu mẫu HTML** 17](#_Toc146352749)

[**4.1.1 Input Elements (Phần tử đầu vào)** 18](#_Toc146352750)

[**4.1.2 Text Fields (Trường văn bản)** 18](#_Toc146352751)

[**4.1.3Phần tử <label>** 18](#_Toc146352752)

[**4.1.4 Radio Buttons (Nút Radio)** 19](#_Toc146352753)

[**4.1.5 Check box** 19](#_Toc146352754)

[**4.2** **CSS Forms (Biểu mẫu CSS)** 19](#_Toc146352755)

[**4.2.1 Các trường nhập liệu kiểu dáng (Styling Input Fields)** 19](#_Toc146352756)

[**4.3** **Sử dụng JavaScript** 20](#_Toc146352757)

[**4.3.1** **Xác thực biểu mẫu JavaScript** 20](#_Toc146352758)

[**4.3.2** **Xác thực biểu mẫu HTML tự động** 20](#_Toc146352759)

[**4.3.3** **Xác thực dữ liệu** 21](#_Toc146352760)

[**4.3.4** **Xác thực ràng buộc HTML** 21](#_Toc146352761)

[**4.3.5** **Xác thực ràng buộc thuộc tính đầu vào HTML** 21](#_Toc146352762)

[**4.3.6** **Xác thực ràng buộc bộ chọn giả CSS** 22](#_Toc146352763)

[**5.** **Quy trình phát triển Web** 22](#_Toc146352764)

[**5.1 Quản lý tiến độ phát triển website** 22](#_Toc146352765)

[**5.2** **Quy trình phát triển website** 23](#_Toc146352766)

[**5.2.1** **Thu thập thông tin, tiếp nhận yêu cầu của khách hàng** 23](#_Toc146352767)

[**5.2.2** **Lập kế hoạch** 24](#_Toc146352768)

[**5.2.3** **Lựa chọn tên miền website và hosting** 25](#_Toc146352769)

[**5.2.4** **Thiết kế giao diện** 25](#_Toc146352770)

[**5.2.5** **Xây dựng tính năng và nội dung website** 27](#_Toc146352771)

[**5.2.6** **Kiểm tra và chỉnh sửa** 27](#_Toc146352772)

[**5.2.7** **Bàn giao và hướng dẫn sử dụng** 28](#_Toc146352773)

[**5.2.8** **Bảo trì** 29](#_Toc146352774)

**NHIỆM VỤ NHÓM**

1. **Giới thiệu về HTML**

## **1.1 HTML Cơ bản**

* **HTML (HyperText Markup Language):** Ngôn ngữ đánh dấu siêu văn bản là khối xây dựng cơ bản nhất của web. Nó xác định ý nghĩa của nội dung Web. Các công nghệ khác ngoài HTML thường được sử dụng để mô tả sự xuất hiện / trình bày của trang Web(CSS) hoặc chức năng / hành vi (JavaScript).
* “Siêu văn bản” đề cập đến các liên kết kết nối các trang Web với nhau, trong một trang Web duy nhất hoặc giữa các trang. Liên kết là một khía cạnh cơ bản của Web. Bằng cách tải nội dung lên Internet và liên kết nó với các trang do người khác tạo, bạn trở thành người tham gia tích cực vào World Wide Web.
  1. **Hướng dẫn tạo một tệp HTML cơ bản:**
  2. **Hướng dẫn cú pháp và các phần tử HTML cơ bản:**
* **HTML:** là ngôn ngữ đánh dấu được xác định cấu trúc nội dung của bạn. HTML bao gồm loạt các phần tử mà bạn sử dụng để đính kèm hoặc bọc các phần khác nhau của nội dung để làm cho nó xuất hiện theo một cách nhất định hoặc hành động theo một cách nhất định. Các thẻ kèm theo có thể tạo siêu liên kết từ hoặc hình ảnh đến một nơi khác, có thể in nghiêng các từ, có thể làm cho phông chữ lơn hơn hoặc nhỏ hơn. Ví dụ lấy dòng nội dung sau:

***Hình 1.1 ví dụ khởi đầu***

* **Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

  Mô tả được tạo tự động**Nếu chúng ta muốn dòng đứng một mình, chúng ta có thể chỉ định rằng đó là một đoạn văn bằng cách đặt nó trong các thẻ đoan:

***Hình 1.2 thẻ p trong HTM***

* **Cấu trúc của một phần tử HTML:** Hãy khám phá phần tử đoạn văn này xa hơn một chút.

**Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, hàng

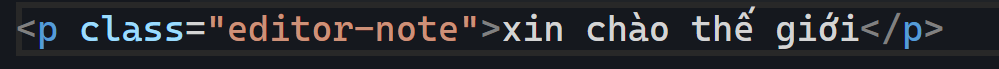
Mô tả được tạo tự động**

***Hình 1.3 Cấu trúc phần tử HTML***

* + 1. **Các phần tử chính của phần tử như sau**

1. Thẻ mở đầu (Opening tag): bao gồm tên của phần tử (trong trường hợp này là p), được bọc trong dấu ngoặc góc mở và đóng. Điều này cho biết nơi phần tử bắt đầu hoặc bắt đầu có hiệu lực - trong trường hợp này là nơi đoạn văn bắt đầu.
2. Thẻ đóng (Closing tag): Giống như thẻ mở, ngoại trừ việc nó bao gồm dấu gạch chéo về phía trước tên phần tử. Điều này cho biết nơi phần tử kết thúc – trong trường hợp này là nơi đoạn văn kết thúc. Không thêm thẻ đóng là một trong những lỗi tiêu chuẩn cho người mới bắt đầu và có thể dẫn đến kết quả lạ.
3. Nội dung (The Content): Đây là nội dung của phần tử, trong trường hợp nàym chỉ là băn bản.
4. Các phần tử (The Element): Thẻ mở, thẻ đóng và nội dung cùng nhau tạo thành phần tử

### **Các phần tử có thể có các thuộc tính giống như sau**

****Thuộc tính chứa thông tin bổ sung về phần tử mà bạn không muốn xuất hiện trong nội dung thực tế. Ở đây, là tên thuộc tính (attribute name) và giá trị thuộc tính (attribute value). Thuộc tính cho phép bạn cung cấp cho phần tử một mã định danh không duy nhất có thể sử dụng để nhắm mục tiêu nó (và bất kỳ phần tử nào khác có cùng giá trị) với thông tin kiểu và những thứ khác. Một số thuộc tính không có giá trị, chẳng hạn như required. “class=“editor-note””.

***Hình 1.4 Thuộc tính attribute***

**Các thuộc tính đặt giá trị luôn có:**

1. Khoảng trăng giữa nó và tên phần tử (hoặc thuộc tính trước, nếu phần tử đã có một hoặc nhiều thuộc tính).
2. Tên thuộc tính theo sau là dấu bằng.
3. Giá trị thuộc tính được bọc bằng dấu ngoặc kép mở và đóng.

### **1.3.3 Phần tử trống**

Một số phần tử không có nội dung và đượ gọi là phần tử trống (void elements). Lấy phần tử <img> mà chúng ta đã có trong trang HTML của chúng ta:

***Hình 1.5 Thẻ images trong HTML***

Điều này chứa hai thuộc tính, nhưng không có thẻ đóng và không có nội dung bên trong.

Điều này là do một số yếu tố hình ảnh không bao bọc nội dung để ảnh hưởng đến nó. Mục đích của nó là nhúng một hình ảnh vào trong HTML ở nơi nó xuất hiện.

### **1.3.4 Giải phẫu tài liệu HTML**

**Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động**Điều đó kết thúc những điều cơ bản của các phần tử HTML riêng lẻ, nhưng chúng không tiện dụng. Chúng ta sẽ xem xét cách các phần tử riêng lẻ được kết hợp để tạo thành toàn bộ trang HTML. Hãy xem lại mã ở ví dụ dưới đây:

***Hình 1.6 Cấu trúc đầy đủ HTML***

Ở đây, chúng ta có những thứ sau:

* **<!DOCTYPE html>** : Là một mở đầu bắt buộc. Khi HTML mới xuất hiên (1991-1992), Doctype có nghĩa là hoạt động như các liên kết đến một tập hợp các quy tắc mà trang HTML phải tuân theo để được coi là HTML tốt, có thể kiểm tra lỗi tự động và những hữu ích khác. Tuy nhiên, ngày nay chúng không làm được những điều gì và về cơ bản chỉ cần thiết để đảm bảo tài liệu của bạn hoạt động chính xác.
* **<html></html>**(The <html> element) Phần tử này bao bọc hết tất cả nội dung trên toàn bộ trang và đôi khi gọi là phần tử gốc. Nó cũng bao gồm thuộc tính, thiết lập ngôn ngữ chính xác của tài liệu.
* **<head></head>**(The <head> element): Phần tử này hoạt động như một vùng chứa cho tất cả nội dung bạn muốn đưa vào trang HTML không phải alf nội dung bạn đang hiển thị cho người xem trang của mình. Điều này bao gồm những thứ như keyword và mô tả trang mà bạn muốn xuất hiện trong kết quả tìm kiếm, CSS để tạo ra kiểu cho nội dung, khai báo bộ ký tự và hơn thế nữa.
* **<meta charset=“utf-8”>**: Phần tử này đặt bộ ký tự mà tài liệu của bạn nên sự dụng thành UTF-8, bao gồm gần hết các ký tự từ phần lớn các ngôn ngữ viết.
* **<meta name= “viewport” content= “witdth=device-width”>** (Viewport element): Phần tử này đảm bảo trang hiển thị ở độ rộng của khung nhìn, ngăn trình duyệt di động hiển thị các trang rộng hơn chế độ xem và sau đó thu nhỏ chúng xuống.
* **<title></<title>** (The title element): Phần tử này đặt tiêu đề trang của bạn, là tiêu đề xuất hiện trong tab trình duyệt mà trang được tải. Nó cũng được sử dụng để mô tả trang khi bạn đánh dấu yêu thích nó.
* **<body></body>** (The body element): Phần tử chứa tất cả nội dung mà bạn muốn hiện thị cho người dùng web khi họ muốn truy cập vào trang của bạn, cho dù đó là văn bản, hình ảnh, video, trò chơi, bản âm thanh có thể phá hay bất cứ thứ gì khác.

## **1.4 Tạo trang Web với tiêu đề, đoạn văn bản, hình ảnh và siêu liên kết**

* **Tiêu đề:**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, phần mềm

Mô tả được tạo tự độngCác phần tử tiêu đề (heading elements) cho phép bạn chỉ định rằng một số phần nhất định trong nội dung của bạn là tiêu đề hoặc tiêu đề phụ. Theo cùng một cách mà một cuốn sách có tiêu đề chính, tiêu đề chương và phụ đề, một tài liệu HTML cũng có thể. HTML chứa 6 cấp độ tiêu đề, từ <h1> đến <h6>, mặc dù bạn thường chỉ sử dụng tối đa 3 đến 4.

***Hình 1.7 Các thẻ tiêu đề HTML***

* **Đoạn văn bản:**

Ảnh có chứa ảnh chụp màn hình, văn bản, Phông chữ

Mô tả được tạo tự độngCác phần tử <p> là để chứa các đoạn văn bản. Bạn sẽ xử dụng chúng thường xuyên khi đánh dấu nội dung văn bản thông thường.

***Hình 1.8 Thẻ p trong HTML***

* **Danh sách:**

Rất nhiều nội dung của web là danh sách và HTML các yếu tố đặc biệt cho những điều này. Danh sách đánh dấu luôn bao gồm ít nhất 2 yếu tố. Các loại danh sách phổ biến nhất là danh sách thứ tự và không có thứ tự

* **Danh sách không có thứ tự** dành cho các danh sách mà thứ tự của các mặt hàng không quan trọng, chẳng hạn như danh sách mua sắm. Chúng được bọc trong một thẻ <ul>.
* **Danh sách có thứ tự** dành cho các danh sách mà thứ tự của các mặt hàng quan trọng, chẳng hạn như công thức. Chúng được bọc trong thẻ <ol>.

Mỗi mục bên trong danh sách được đặt bên trong thẻ <li>.

Ảnh có chứa ảnh chụp màn hình, văn bản, Phông chữ

Mô tả được tạo tự độngVí dụ: chúng ta muốn biến một phần của đoạn văn sau thành một danh sách

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự độngChúng ta có thể sửa đổi đánh dấu thành điều này:

***Hình 1.9 Thẻ danh sách không có thứ tự***

* **Liên kết:**

Liên kết rất quan trọng, chúng là những gì làm cho web trở thành một trang web! Để thêm một liên kết, chúng ta cần sử dụng một thẻ đơn giản <a>. “a” là viết tắt của “anchor”. Để đặt văn bản trong đoạn văn của bạn thành một liên kết, hãy làm như sau:

* Chọn một số văn bản. Chọn văn bản “ Ngôn ngữ lập trình phổ biến”.
* Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

  Mô tả được tạo tự độngNgắt dòng văn bản trong phần tử <a> như hình dưới đây:

***Hình 1.10 Thẻ a trong HTML***

* Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, phần mềm

  Mô tả được tạo tự độngCung cấp cho phần tử <a> một thuộc tính:

***Hình 1.11 thuộc tính trong thẻ a***

* Điền giá trị của thuộc tính này bằng địa chỉ web mà bạn muốn liên kết đến:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, phần mềm

Mô tả được tạo tự động

***Hình 1.12 Gắn địa chỉ vào thẻ a***

Bạn có thể nhận được kết quả không mong muốn nếu bạn bỏ qua giao thức, ở đầu địa chỉ web. Sau khi tạo liên kết, hãy nhấp vào liên kết đó để đảm bảo rằng nó đang gửi cho bạn nơi bạn muốn.

1. **Cơ bản về CSS**

## **2.1 CSS – Thêm kiểu cho web**

**CSS (Cascading Style Sheets) :** Là công nghệ đầu tiên bạn nên bắt đầu học sau HTML. Trong khi HTML được sử dụng để xác định cấu trúc và ngữ nghĩa của nội dung thì CSS được sử dụng để tạo kiểu và bố trí nó. Ví dụ: bạn có thể sử dụng CSS để thay đổi phông chữ, màu sắc, kích thước và khoảng cách của nội dung, chia nó thành nhiều cột hoặc thêm đoạn ảnh và các tính năng trang trí khác.

### **2.1.1 Inline CSS (CSS trực tiếp trong phần tử HTML – Nội tuyến)**

Phương pháp này, bạn thêm kiểu CSS trực tiếp vào một phần tử HTML cụ thể bằng thuộc tính **`style`.** Điều này hữu ích khi bạn muốn tùy chỉnh một phần tử duy nhất. Thuộc tính style có thể chứa bất kì thuộc tính CSS nào.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động***Hình 2.1 CSS Nội tuyến***

Tránh sử dụng CSS theo cách này, khi có thể. Nó trái ngược với một thực tiễn tốt nhất. Đầu tiên, nó là việc triển khai CSS kém hiệu quả nhất để bảo trì. Một thay đổi kiểu dáng có thể yêu cầu nhiều chỉnh sửa trong một trang Web. Thứ hai, CSS nội tuyến cũng kết hợp mã trình bày (CSS) với HTML và nội dung, làm cho mọi thứ khó đọc và hiểu hơn. Tách mã và nội dung giúp bảo trì dễ dàng hơn cho tất cả những người làm việc trên trang Web.

Có một vài trường hợp mà kiểu nội tuyến phổ biến hơn. Bạn có thể phải sử dựng các kiểu nội tuyến nếu môi trường làm việc của bạn rất hạn chế.

### **2.1.2 Internal CSS (CSS trong tệp HTML – Nội bộ)**

Phương pháp này, bạn đặt mã CSS trực tiếp trong thẻ **`<style>`** của tệp HTML. Điều này thích hợp cho việc tạo kiểu cho một trang Web cụ thể mà bạn không muốn chia sẻ kiểu đó với các trang khác.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động HTML cho kiểu nội bộ có thể trông như thế này:

***Hình 2.2 CSS nội bộ***

Trong một số trường hợp, kiểu nội bộ có thể hữu ích. Ví Dụ: có lẽ bạn đang làm việc với một hệ thống quản lý nội dung nơi bạn bị chặn sửa đổi các tệp CSS bên ngoài.

Nhưng đối với các trang Web có nhiều hơn một trang, kiểu nội bộ trở thành một các làm việc kém hiệu quả hơn. Để áp dụng kiểu CSS thống nhất cho nhiều trang bằng cách sử dụng kiểu nội bộ, bạn phải có một kiểu nội bộ trong mỗi trang web. Với CSS trong kiểu nội bộ, có nguy cơ ngay cả một thay đổi kiểu đơn giản cũng có thể yêu cầu chính sửa nhiều trang Web.

### **2.1.3 External CSS (CSS bên ngoài)**

Trong phương pháp này, bạn tạo một tệp CSS riêng biệt (Ví dụ: **`style.css`**) chứa tất cả các kiểu của bạn và sau đó liên kết tệp CSS này với trang web của bạn bằng thẻ **`<link>`** trong phần **`<head>`** của tệp HTML.

Tham chiếu một kiểu CSS bên ngoài từ thẻ HTML **`<linik>`** :

***Hình****Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động* ***2.3 CSS bên ngoại***

Ảnh có chứa văn bản, Phần mềm đa phương tiện, phần mềm, Phần mềm đồ họa

Mô tả được tạo tự động Tệp biểu định kiểu CSS có thể trông như này:

***Hình 2.4 tệp biểu định kiểu CSS***

Thuộc tính của thẻ **<link>** cần thamn chiếu đến một tệo trên hệ thống của bạn. Trong ví dụ trên, tệp CSS nằm trong vùng thư mục với tài liệu HTML, nhưng bạn có thể đặt nó ở một nơi khác và điều chỉnh đường dẫn.

Đây là phương pháp phổ biến và hữu ích nhất để đưa CSS vào tài liệu. bạn có thể liên kết một tệp CSS duy nhất với nhiều trang Web, tạo kiểu cho tất cả chúng bằng cùng một biểu định CSS.

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

* Đầu tiên, trong tệp tin CSS riêng hoặc phần `<style>` của tệp tin HTML, tạo một lớp bằng cách sử dụng dấu chấm "." trước tên lớp của bạn. Ví dụ:

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, màu đen

Mô tả được tạo tự động**

***Hình 2.5 Kiểu của lớp***

* Sau đó, trong phần tử HTML mà bạn muốn áp dụng kiểu cho, thêm thuộc tính "class" với giá trị là tên lớp mà bạn đã tạo. Ví dụ:

***Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động***

***Hình 2.6 Ví dụ thêm Class để áp dụng CSS***

* Bây giờ, trong phần CSS có thể sử dụng lựa chọn `.` kết hợp với tên lớp để áp dụng kiểu cho các phần tử có lớp tương ứng. Ví dụ:

**Ảnh có chứa văn bản, Phần mềm đa phương tiện, phần mềm, Phần mềm đồ họa

Mô tả được tạo tự động**

***Hình 2.7 Lựa chọn CSS để áp dụng kiểu***

* Trong ví dụ trên, chúng ta đã tạo một lớp có tên là "my-class" và áp dụng kiểu màu chữ, kích thước chữ, phông chữ và kiểu phông chữ cho mọi phần tử có lớp đó.
* Có thể áp dụng một lớp cho nhiều phần tử khác nhau trong tệp HTML bằng cách thêm
* <p class="my-class">Đây là một đoạn văn bản</p>
* <h1 class="my-class">Xin chào đây là trang web của tôi</h1>

Như vậy, tất cả đoạn văn bản và tiêu đề có lớp "my-class" sẽ được áp dụng vào cả 2 kết quả như sau

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

Để thử nghiệm các thuộc tính CSS cơ bản như font chữ, màu sắc, độ rộng và chiều cao, bạn có thể làm như sau:

### **2.3.1. Font chữ**

* Sử dụng thuộc tính `font-family` để đặt font chữ cho một phần tử. Ví dụ: `font-family: Arial, sans-serif;`
* Sử dụng thuộc tính `font-size` để đặt kích thước font chữ. Ví dụ: `font-size: 16px; `

### **2.3.2. Màu sắc**

* Sử dụng thuộc tính `color` để đặt màu chữ cho một phần tử. Ví dụ:`color: blue;`
* Sử dụng thuộc tính `background-color` để đặt màu nền cho một phần tử. Ví dụ: `background-color: yellow;`

**2.3.3. Độ rộng và chiều cao**

* Sử dụng thuộc tính `width` để đặt độ rộng cho một phần tử. Ví dụ: `width: 300px;`
* Sử dụng thuộc tính `height` để đặt chiều cao cho một phần tử. Ví dụ: `height: 50px;`
* Thử nghiệm các thuộc tính này bằng cách thêm chúng vào phần CSS của trang web của bạn. Ví dụ:

**Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động**

***Hình 2.8 Thử nghiệm các thuộc tính CSS***

Ngoài ra, CSS có các thuộc tính khác như **`padding`, `margin`, `border`**, và nhiều thuộc tính khác để tùy chỉnh kiểu cho phần tử trong trang web của bạn

**`padding`**: Được sử dụng để tạo khoảng cách giữa nội dung bên trong phần tử và viền của phần tử.

thể đặt giá trị padding cho các cạnh riêng biệt hoặc sử dụng thuộc tính **`padding-top`, `padding-right`, `padding-bottom`**và **`padding-left`** để chỉ định giá trị cho từng cạnh một cách riêng lẻ.

**`margin`**: Được sử dụng để tạo khoảng cách giữa các phần tử.

Khác với `padding`, **`margin`** không ảnh hưởng đến nội dung bên trong phần tử. Cũng giống như `padding`, có thể đặt giá trị margin cho các cạnh riêng biệt hoặc sử dụng thuộc tính **`margin-top`, `margin-right`**, **`margin-bottom`** và **`margin-left`** để chỉ định giá trị cho từng cạnh một cách riêng lẻ.

**`border`**: Được sử dụng để tạo đường viền xung quanh một phần tử. Có thể thiết lập độ dày, kiểu và màu sắc của đường viền bằng cách sử dụng các thuộc tính **`border-width`, `border-style`**và**`border-color`**.

Cũng giống như `padding` và `margin`, có thể thiết lập giá trị cho từng cạnh bằng cách sử dụng các thuộc tính có tiền tố là **`border-top`, `border-right`, `border-bottom`**và**`border-left`.**

1. **JavaScript cơ bản**
   1. **Giới thiệu cú pháp JavaScript cơ bản**

Cú pháp JavaScript là tập hợp các quy tắc, các chương trình JavaScript được xây dựng:

*// How to create variables:*

*var x;*

*let y;*

*// How to use variables:*

*x = 5;*

*y = 6;*

*let z = x + y;*

**Giá trị JavaScript (JavaScript Values)**

Cú pháp JavaScript định nghĩa hai loại giá trị:

* Giá trị cố định
* Giá trị biến

Giá trị cố định được gọi là **chữ viết (Literals)**

Giá trị biến được gọi là **biến (Variables)**

* **JavaScript literals :** Hai quy tắc cú pháp quan trọng nhất cho các giá trị cố định là:
* **Number (Số) :** Được viết dưới dạng số thập phân hoặc không thập phân
* **String (Chuỗi)**: Là văn bản được viết trong dấu ngoặc nháy kép hoặc nháy đơn.
* **JavaScript Variables:** trong ngôn ngữ lập trình, các biến được sử dụng để lưu trữ các giá trị dữ liệu. JavaScaript sử dụng các tư khóa, và để khai báo biến : var, let, const. Dấu bằng (=) được sử dụng để gán giá trị cho biến.
  1. **Tạo biến, hàm và sử dụng sự kiện JavaScript**
     1. **Tạo biến (Variables)**

Biến trong JavaScript được tạo bằng cách sử dụng các từ khóa ‘var’, ‘let’, ‘const’.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự độngVí dụ : Sử dụng biến ‘var’ hoặc ‘let’( cho biến có thể thay đổi giá trị)

***Hình 3.1 Sử dụng biến trong JavaScript***

Ảnh có chứa văn bản, Phông chữ, Phần mềm đa phương tiện, phần mềm

Mô tả được tạo tự độngVí dụ: sử dụng ‘const’ (cho biến không thay đổi giá trị)

***Hình 3.2 Sử dụng biến const trong JS***

* + 1. **Tạo hàm (Function)**

Hàm trong JavaScript được tạo bằng từ khóa ‘function’

Ảnh có chứa văn bản, Phông chữ, Phần mềm đa phương tiện, phần mềm

Mô tả được tạo tự độngVí dụ: Khai báo hàm không tham số:

***Hình 3.3 Khai báo hàm không tham số***

Ảnh có chứa văn bản, Phần mềm đa phương tiện, phần mềm, Phần mềm đồ họa

Mô tả được tạo tự độngVí dụ: Khai báo hàm có tham số và giá trị trả về

***Hình 3.4 Khai báo hàm có tham số***

* + 1. **Sử dụng sự kiện (Event)**

Sự kiện trong JavaScript thường được sử dụng để phản hồi khi người dùng tương tác với trang web.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự độngVí dụ: Gán sự kiện cho một phần tử HTML:

*Ảnh có chứa văn bản, phần mềm, Phần mềm đa phương tiện, Phông chữ

Mô tả được tạo tự động****Hình 3.5 Gán sự kiện cho một phần tử***

Trong ví dụ này, khi người nhấp vào nút, hàm được gọi và thông báo “Button clicked!’ sẽ hiển thị.

## **3.3. Tạo các ví dụ đơn giản như hiển thị thông báo khi nút được nhấn**

* Sử dụng sự kiện:
* Sử dụng phương thức `addEventListener()` để thêm một chức năng xử lý cho một sự kiện cụ thể của một phần tử HTML. Ví dụ:

const button = document.querySelector('button');

button.addEventListener('click', function() {

// Mã lệnh xử lý sự kiện

});

* Trong phần tham số thứ hai của `addEventListener()`, bạn có thể tạo một hàm xử lý sự kiện riêng, hoặc sử dụng một hàm đã được định nghĩa trước đó.
* Ví dụ đơn giản hiển thị thông báo khi nút được nhấn:
* HTML:

<button id="myButton">Nhấn vào đây</button>

* JavaScript:

const button = document.querySelector('#myButton');

button.addEventListener('click', function() {

alert('Bạn đã nhấn nút!');

});

Trong ví dụ trên, chúng ta đã sử dụng JavaScript để thêm một hàm xử lý sự kiện cho nút. Khi nút được nhấn, sự kiện `click` sẽ được kích hoạt và thông báo "Bạn đã nhấn nút!" sẽ hiển thị. Có thể sửa đổi thông báo, thay đổi phần tử mà sự kiện áp dụng và tùy chỉnh hành vi cho các sự kiện khác nhau.

1. **HTML Forms và Input Elements**

**4.1 Tạo biểu mẫu HTML**

Biểu mẫu HTML (HTML Forms) được sử dụng để thu thập thông tin đầu vào của người dùng. Đầu vào của người dùng là thường được gửi đến một máy chủ để xử lý.

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động**

***Hình 4.1 Biểu mẫu HTML***

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự độngPhần tử <form> trong HTML được sử dụng để tạo biểu mẫu HTML cho người dùng nhập:

***Hình 4.2 Phần tử Form trong HTML***

Phần tử là một thùng chứa cho các loại phần tử đầu vào khác nhau, chăng hạn như: Trường văn bản, hộp kiểm, radio nút, nút gửi,….

Tất cả phần tử biểu mẫu khác nhau được để cập trong phần này:

### **4.1.1 Input Elements (Phần tử đầu vào)**

Phần tử đầu vào (Input elements) được sử dụng nhiều nhất là phần tử <input>

Ảnh có chứa văn bản, Phông chữ, số, hàng

Mô tả được tạo tự độngMột phần tử có thể được hiển thị trong nhiều cách, tùy thuộc vào thuộc tính, dưới đau là một số ví dụ:

***Hình 4.2 Một số ví dụ các form trong HTML***

### **4.1.2 Text Fields (Trường văn bản)**

Định nghĩa một trường đầu vào một dòng cho văn bản <input type= “text”>

Ví dụ: Biểu mẫu có các trường nhập liệu cho văn bản:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

***Hình 4.3 Biểu mẫu có các trường nhập liệu cơ bản***

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, Phông chữ

Mô tả được tạo tự động Đây là cách mã HTML ở trên sẽ được hiển thị trên trình duyệt:

***Hình 4.4 Kết quả hiển thị trên trình duyệt***

### **4.1.3Phần tử <label>**

Lưu ý việc sử dụng phần tử trong ví dụ ở trên

Thẻ xác định nhãn cho nhiều yếu tố hình thức

Phần tử này hữu ích cho người dùng trình đọc màn hình vì trình đọc màn hình sẽ đọc to nhãn khi người dùng tập trung vào phần tử đầu vào.

Yếu tố này cũng giúp người dùng khó nhấo vào các vùng rất nhỏ (chẳng hạn như nút Radio hoặc check box) bở vì khi người dùng nhập vào văn bản trong phần tử, nó sẽ chuyển đổi nút radio/check box

Thuộc tính của thẻ nên bằng thuộc tính của phần tử để liên kết chúng lại với nhau.

### **4.1.4 Radio Buttons (Nút Radio)**

Nút Radio cho phép người dùng chọn một trong số các lựa chọn giới hạn

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động Ví dụ:

***Hình 4.5 Ví dụ về Radio buttons***

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, màu trắng

Mô tả được tạo tự động Đây là cách mã HTML trên sẽ được hiển thị trong trình duyệt:

***Hình 4.6 Mã HTML hiển thị trên trình duyệt***

### **4.1.5 Check box**

Cũng giống như Nút Radio, Check box cho phép người dùng chọn tùy chọn ZERO hoặc MORE của một số lựa chọn giới hạn.

* 1. **CSS Forms (Biểu mẫu CSS)**

Giao diện của một biểu mẫu HTML có thể được cải thiện rất nhiều với CSS**.**

### **4.2.1 Các trường nhập liệu kiểu dáng (Styling Input Fields)**

Sử dụng thuộc tính để xác định chiều rộng của trường nhập: width

Nếu bạn chỉ muốn tạo kiểu cho một loại đầu vào cụ thể, bạn có thể sử dụng bộ chọn thuộc tính:

* input[type=text] : sẽ chỉ chọn các trường văn bản
* input[type=password] : sẽ chỉ chọn các trường mật khẩu
* input[type=number] : sẽ chỉ chọn các trường số …
  1. **Sử dụng JavaScript** 
     1. **Xác thực biểu mẫu JavaScript**

Xác thực biểu mẫu HTML có thể được thực hiện tự động bởi trình duyệt

Nếu trường biểu mẫu (fname) trống, hàm này sẽ cảnh báo thư và trả về false, để ngăn cản không cho biểu mẫu được gửi:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự độngVí dụ về JavaScript

***Hình 4.7 Ví dụ về JavaScript***

Hàm có thể được gọi khi biểu mẫu được gửi:

Ảnh có chứa văn bản, Phông chữ, Phần mềm đa phương tiện, ảnh chụp màn hình

Mô tả được tạo tự độngVí dụ về biểu mẫu HTML:

***Hình 4.8 ví dụ về biểu mẫu HTML***

* + 1. **Xác thực biểu mẫu HTML tự động**

Xác thực biểu mẫu HTML có thể được thực hiện tự động bởi trình duyệt:

Nếu trường biểu mẫu (fname) trống, thuộc tính sẽ ngắn biểu mẫu này gửi: required

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự độngVí dụ về biểu mẫu HTML:

***Hình 4.9 Ví dụ về biểu mẫu HTML***

* + 1. **Xác thực dữ liệu**

Xác thực dữ liệu là quá trình đảm bảo rằng đầu vào của người dùng sạch sẽ, chính xác và hữu ích

Các nhiệm vụ xác thực điển hình là:

* Người dùng đã điền vào tất cả các trường bắt buộc chưa?
* Người dùng đã nhập ngày hợp lệ chưa?
* Người dùng đã nhập văn bản vào trường số chưa?

Thông thường, mục đích của xác thực dữ liệu là để đảm bảo đầu vào chính xác của người dùng.

Xác nhận có thể được xác định bằng nhiều phương pháp khác nhau và được triển khai trong nhiều phương pháp những cách khác nhau.

Xác thực phía máy chủ được thực hiện bởi một máy chủ web, sau khi đầu vào được gửi đến máy chủ.

Xác thực phía máy khách được thực hiện bỏi trình duyệt web, trước khi đầu vào được gửi đến máy chủ web.

* + 1. **Xác thực ràng buộc HTML**

HTML5 giới thiệu một khái niệm xác thực HTML mới đc gọi là constraint xác nhận.

Xác thực ràng buộc HTML dựa trên:

* Xác thực ràng buộc thuộc tính đầu vào HTML
* Xác thực ràng buộc bộ chọn giả CSS
* Xác thực ràng buộc thuộc tính và phương thức DOM
  + 1. **Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

       Mô tả được tạo tự độngXác thực ràng buộc thuộc tính đầu vào HTML**

***Hình 4.10 một số xác thực ràng buộc đầu vào HTML***

* + 1. Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

       Mô tả được tạo tự động**Xác thực ràng buộc bộ chọn giả CSS**

***Hình 4.11 Một số xác thực ràng buộc chọn bộ giả CSS***

1. **Quy trình phát triển Web**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự độngChúng ta thường cho rằng việc viết mã nguồn (coding) là yếu tố chính và là toàn bộ công việc mà lập trình viên cần làm khi phát triển Website. Tuy nhiên, để một dự án thành công, giai đoạn thu tập thông tin, lập kế hoạch chi tiết, bảo trì hệ thống sau khi ra mắt sản phẩm là những phần không thể thiếu. Tất cả tạo nên một **quy trình phát triển website** toàn diện.

***Hình 5.1 Quy trình phát triển Web***

Trước khi giới thiệu về từng bước, cùng tim hiểu qua về cách quản lý tiến độ của một dự án phát triển websie

## **5.1 Quản lý tiến độ phát triển website**

Khi nghĩ tới việc xây dựng website, bạn cần quan tâm đến 2 vấn đề chính: chi phí và thời gian. Hai giá trị này phụ thuộc phần lớn vào quy mô và phạm vi của dự án.

Để phác thảo tổng quan thời gian của toàn bộ quá trình phát triển, bạn có thể tại một workapace để quản lý tiến trình, quản lý công việc và thiết lập các mốc thời gian quan trọng cho dự án. Đó là cách tốt nhất để theo dõi tiến độ dự án và để đảm bảo dự án đúng thời hạn.

* 1. **Quy trình phát triển website**
     1. **Thu thập thông tin, tiếp nhận yêu cầu của khách hàng**

Thực tế, không phải ai cũng có nhiều kiến thức chuyên môn về lĩnh vực **thiết kế trang web.** Do đó, trong đơn đặt hàng **dịch vụ thiết kế web** có thể tồn tại nhiều yêu cầu bất cập, có thể khiến quá trình xây dựng website gặp nhiều khó khăn.

Vì vậy, bước đầu tiên trong quy trình Phát triển website luôn là Phân tích yêu cầu và thông tin của khách hàng. Sau khi xem xét tính khả thi của từng chi tiết, các lập trình viên sẽ tư vấn và thảo luận với khách hàng về các bước xây dựng website.

Trong bước đầu tiên này, nhiệm vụ cần xác định rõ các yêu cầu sau:

* **Lĩnh vực hoạt động:**

Trong từng lĩnh vực, yêu cầu và xu hướng về thiết kế website luôn khác nhau. Mặc dù không bắt buộc, tuy nhiên để đạt được hiệu quả tối đa, các website nên cẩn thận chọn lọc và tuân thủ những xu hướng và quy tắc của lĩnh vực của mình.

**Ví dụ:** trong trường hợp thiết kế website bán hàng trực tuyến, việc sử dụng bố cục dạng lưới để hiển thị các ảnh sản phẩm thu nhỏ là một lựa chọn tốt. Điều này giúp tăng số lượng sản phẩm hiển thị trên một trang và hạn chế việc phải cuộn chuột để có thể xem qua toàn bộ sản phẩm.

Không chỉ sự thiêt kế, mà cả các tính năng cốt lõi của một trang web cũng thay đổi tùy theo lĩnh vực hoạt động. Mỗi loại trang web sẽ có những bộ tính năng chủ yếu độc đáo không giống nhau.

**Ví dụ:** Trang web bán hàng online hay website thương mại điện tử sẽ phải có các tính năng thiết yếu như đăng nhập và quản lý tài khoản, quản lý giỏ hàng, quản lý coupon/voucher/mã giảm giá, định vị khách hàng để tính vận chuyển hoặc tìm đại lý/cửa hàng gần nhất, v.v…

* **Đối tượng website hướng tới**

Dù cùng 1 lĩnh vực, các website khác nhau có thể nhắm tới những nhóm đối tượng khác nhau. Vì thế, phong các thiết kế được sử dụng trên web site cũng từ đó mà phải lựa chọn sao cho phù hợp với từng đối tượng khác nhau.

Các công ty thiết kế website cao cấp sẽ luôn cần hiểu rõ nhóm đối tượng chính mà website muốn nhắm tới để có được hướng xâu dựng và phát triển website chính xác.

* **Yêu cầu kỹ thuật**

Yêu cầu kỹ thuật bao gồm nhiều yếu tố khác nhau, trong đó có yêu cầu về tên miền, hosting hay lập trình code web,… Khách hàng được phép yêu cầu những chi tiết thiết kế hoặc những tính năng tùy chọn sao cho phù hợp với lĩnh vực và hướng phát triển của doanh nghiệp.

Vì vậy, đội ngũ thiết kế website chuyên nghiệp sẽ xem xét, phân tích những yêu cầu tùy chọn của khách hàng. Sau đó, sẽ tiến hành tư vấn và thảo luận lại với khách hàng để có thể thống nhất được sản phẩm cuối cùng sẽ ra sao.

***Hình 5.2 Thu thập thông tin và yêu cầu khách hàng***

Một yêu cầu kỹ thuật bắt buộc khác trong quá trình này là bảo mật website. Bạn cần phải lựa chọn các chứng chỉ bảo mật phù hợp để đảm bảo toàn bộ dữ liệu và thông tin trên website an toàn tuyệt đối.

* + 1. **Lập kế hoạch**

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự độngXây dựng website cũng giống như xây dựng một ngôi nhà, trước hết bạn cần có bản vẽ thiết kế. Vì vậy bước phác thảo cấu trúc website là vô cùng quan trọng. Để tránh đi lạc hướng trong quá trình thiết kế website, trước tuên lập trình viên sẽ phác thảo sơ đồ (sitemap) website

***Hình 5.3 Lập kế hoạch***

**Sitemap** là xương sống của một website, là kim chỉ nam trong quá trình thiết thế website. Thiết kế sitemap sẽ giúp người dùng phân chia, lên kế hoạch công việc thiết kế một cách khoa học và hiệu quả.

Bên cạnh sitemap, thiết kế Menu điều hướng chính cũng là một việc vô cùng quan trọng.

Người thiết kế web chuyên nghiệp cần định hình được menu chúng nên gồm có những mục gì, thứ tự các mục ra sao… Bởi vì chính menu chính là phương tiện chính để khách hàng ghé thăm tiếp cận toàn bộ các mục thiết kế của website.

* + 1. **Lựa chọn tên miền website và hosting**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biểu tượng

Mô tả được tạo tự độngTên miền là địa chỉ website và cũng là công cụ định vị rất quan trọng cho website. Tuy nhiên, bạn cần phải đảm bảo các tên miền đó chưa được ai sử dụng và bạn có thể mua nó.

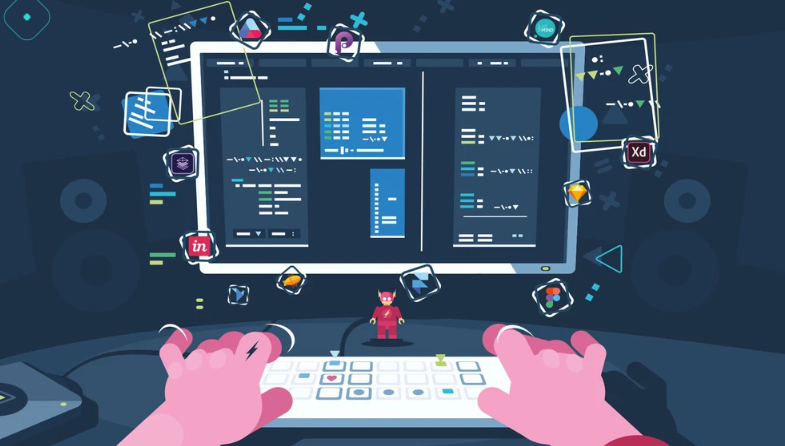
***Hình 5.4 Lựa chọn tên miền và hosting***

Về hosting, đây là nơi vận hành và lưu trữ mọi dữ liêu của website. Cho nên, việc lựa chọn gói hosting lớn hay nhỏ là phụ thuộc vào quy mô website của khách hàng. Khách ghé thăm nhiều hơn sẽ tiêu tốn nhiều tài nguyên hệ thống hơn và đòi hỏi một cấu hình hosting cao hơn.

* + 1. **Thiết kế giao diện**

**Thiết kế giao diện** là bước tiếp theo để khách hàng và người thiết kế thống nhất toàn bộ phần hiển thị website. Có vô số giao diện website, tuy nhiên chúng thường được chia làm 2 nhóm: giao diện có sẵn theo mẫu (template) và giao diện được thiết kế độc quyền.

Dù là giao diện nào, khi bắt đầu thiết kế cũng cần lưu ý:

* Tông màu của logo và bộ nhận diện thương hiệu của doanh nghiệp
* Ý nghĩa, thông điệp mà website muốn thể hiện
* Đối tượng khách hàng có phù hợp với tông màu không
* Chọn tông màu phù hợp với lĩnh vực hoạt động

***Hình 5.5 Thiết kế giao diện***

**Giao diện có sẵn** là nhưng mẫu website được thiết kế sẵn. Giao diện theo mẫu cũng phân chia theo chủ đề với mỗi chủ đề có một số phong cách thiết kế tương tự nhau

Ưu điểm của loại giao diện theo mẫu là có thể nhanh chóng áp dụng vào website, không tốn quá nhiều thời gian và chi phí. Tuy nhiên, sẽ dễ dàng trùng lặp với các website khách và không mang lại nhiều hiệu quả cho hoạt động kinh doanh.

**Giao diện độc quyền** sẽ được các lập trình viên thiết kế riêng cho duy nhất một website

**Giao diện độc quyền** có thể làm nổi bật sự độc đáo của website và thương hiệu hoặc chứa những tính năng, hiệu ứng độc đáo, khó có thể tìm thấy ở đại đa số website khác. Nhờ đó, giao diện độc quyền có hiệu quả về mặt định vị thương hiệu website tốt hơn.

Đội ngũ thiết kế phải bỏ ra nhiều công sức hơn để sáng tạo nên giao diện độc quyền. Vì vậy, giá thành của chúng sẽ cao hơn nhiều có với loại giao diện có sẵn theo mẫu.

Giao diện độc quyền chỉ phù hợp với những doanh nghiệp có tiềm năng kinh tế và tham vọng xây dựng, định vị thương hiệu mạn mẽ.

* + 1. **Xây dựng tính năng và nội dung website**

Ảnh có chứa văn bản, máy tính, phương tiện truyền thông, ảnh chụp màn hình

Mô tả được tạo tự độngĐây là bước cực kỳ quan trọng trong quá trình thiết kế website vởi các tính năng này sẽ quyết định đến hoạt động của trang web sau này.

***Hình 5.6 Xây dựng tính năng và nội dung***

Đầu tiên là các tính năng, chức năng cơ bản, cần thiết để website hoạt động một cách hiệu quả. Tùy theo yêu cầu và lĩnh vực về website khác nhau sẽ có các tính năng cơ bản và nâng cao khác nhau.

Ngoài ra, theo xu hướng và nhu cầu hiện đại, hầu hết các website cần có thiết kế Responsive giúp website hiển thị và hoạt động ổn định trên các thiết bị có kích cỡ màn hìng khác nhau.

Song song với việc lập trình các tính năng cần thiết cho website, đội ngũ phát triển website cũng cần phải xây dựng nội dung cho website. Để đi vào hoạt động, website cần phải được cập nhật nội dung liên tục để khách hàng có thể ghé thăm.

Tùy vào lĩnh vực hoạt động mà website sẽ có lượng bài viết ít hay nhiều. Những nội dung này khách hàng ghé có lý do để truy cập trang, cũng như gia tăng hiệu quả SEO cho trang web.

* + 1. **Kiểm tra và chỉnh sửa**

Kiểm tra và chỉnh sửa là một trong những bước quan trọng nhất trong quá trình xây dựng website. Bởi vì nó quyết định ý tưởng và yêu cầu khách hàng có được hiện thực hóa hay không? Và phát triển wensite của bạn có thành công hay không?

Ảnh có chứa ảnh chụp màn hình, văn bản, thiết kế, internet

Mô tả được tạo tự độngViệc kiểm tra sau khi website được hình thành là điều bắt buộc phải làm. Không thể chắc chắn được rằng website của bạn không có bất kỳ lỗi nào trong quá trình thiết kế. Nếu như không kiểm tra lại website thì bạn có thể gặp phải những phản hồi tiêu cực từ phía khách hàng và dễ đánh mất lòng tin của họ.

***Hình 5.7 Kiểm tra và chỉnh sửa***

Khi tiến hành công việc kiểm tra và chỉnh sửa, nên chú ý các vấn đề sau:

* Tối ưu hóa website cho các thiết bị di động
* Đảm bảo các chức năng của website hoạt động bình thường
* Kiểm tra trang chủ và công cụ tìm kiếm
* Hạn chế tối đa các lỗi mà website có thể gặp phải, đảm bảo độ bảo mật và an toàn.
  + 1. Ảnh có chứa người, máy tính, trang phục, máy tính xách tay

       Mô tả được tạo tự động**Bàn giao và hướng dẫn sử dụng**

***Hình 5.8 Bàn giao sản phẩm***

Sau khi đã cơ bản hoàn thành dịch vụ phát triển website, bên phát triển website sẽ liên hệ để khách hàng có thể kiểm tra, tự mình trải nghiệm thành phẩm. Đội ngũ tư vấn cũng sẽ hướng dẫn khách hàng sử dụng các tính năng, công cụ quán trị website một cách cặn ké.

Nếu có trải nghiệm, chi tiết nào chưa vừa ý hoặc không đúng yêu cầu, khách hàng có thể yêu cầu công ty thiết kế và chỉnh sửa lại.

* + 1. **Bảo trì**

Ảnh có chứa ảnh chụp màn hình, máy tính, thiết kế, internet

Mô tả được tạo tự độngBạn cũng nên lựa chọn các công ty có dịch vụ phát triển web doanh nghiệp uy tín để được bảo trì, hỗ trợ lâu dài và hiệu quả nhất.

***Hình 5.9 Bảo trì***

Trên đây là những thông tin về tầm quan trọng và những tiêu chí, điều kiện cần có khi thiết kế website.

Bên cạnh đó, quy trình phát triển website chuẩn 2023 cũng được trình bày cụ thể trong bài viết. Hi vọng những thông tin này giúp bạn hiểu kỹ hơn về thiết kế website.

Nếu bạn đọc có nhu cầu sử dụng website để phục vụ hoạt động kinh doanh cũng như Marketing thì có thể tham khảo dịch vụ thiết kế Website tại các công ty uy tín.