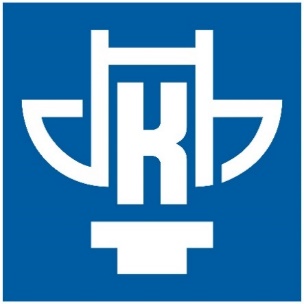
BỘ XÂY DỰNG VIỆT NAM

TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----



**BÀI NGHIÊN CỨU**

**NHÓM 6**

**ĐỀ TÀI:** **XÂY DỰNG WEBSITE**

**NGHE NHẠC**

**Giảng viên hướng dẫn : gv. NGUYỄN THỊ HẠNH**

**Thành viên nhóm : PHẠM TUẤN MINH**

**NGUYỄN THÀNH TRUNG**

**HOÀNG NGỌC THÀNH**

**LỚP : 20CN5**

**MỤC LỤC**

[1. Giới thiệu về HTML 3](#_Toc147431916)

[1.1. Hướng dẫn tạo một tệp HTML cơ bản. 3](#_Toc147431917)

[1.2. Tạo trang web với tiêu đề, đoạn văn bản, hình ảnh và siêu liên kết. 4](#_Toc147431918)

[1.3. Hướng dẫn về cú pháp và các phần tử HTML cơ bản. 5](#_Toc147431919)

[2. Cơ bản về CSS 5](#_Toc147431920)

[2.1. Hướng dẫn cách thêm kiểu cho trang web bằng CSS. 6](#_Toc147431921)

[2.2. Tạo các lớp và áp dụng kiểu bằng cách sử dụng lựa chọn. 7](#_Toc147431922)

[2.3. Thử nghiệm với các thuộc tính CSS cơ bản như font chữ, màu sắc, độ rộng và chiều cao. 8](#_Toc147431923)

[3. JavaScript cơ bản : 9](#_Toc147431924)

[3.1. Giới thiệu cú pháp JavaScript cơ bản. 9](#_Toc147431925)

[3.2. Hướng dẫn tạo biến, hàm, và sử dụng sự kiện. 9](#_Toc147431926)

[3.3. Tạo các ví dụ đơn giản như hiển thị thông báo khi nút được nhấn. 10](#_Toc147431927)

[4. HTML Forms và Input Elements 11](#_Toc147431928)

[4.1. Cách tạo biểu mẫu HTML đơn giản với các phần tử nhập liệu như ô văn bản, ô chọn, và nút gửi 11](#_Toc147431929)

[4.2. Tạo kiểu cho biểu mẫu bằng CSS 12](#_Toc147431930)

[4.3. Sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu 13](#_Toc147431931)

[5. Quy trình phát triển web 15](#_Toc147431932)

[5.1. Giới thiệu về quy trình phát triển web từ lập kế hoạch đến triển khai trang web. 15](#_Toc147431933)

[5.2. Xác định các bước quan trọng như phân tích yêu cầu, thiết kế giao diện, phát triển, kiểm thử, và triển khai. 15](#_Toc147431934)

[5.3. Hướng dẫn về vai trò và trách nhiệm của từng thành viên trong nhóm, bao gồm quản lý dự án và phát triển. 16](#_Toc147431935)

# 1. Giới thiệu về HTML

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu được sử dụng để xây dựng và cấu trúc nội dung trên các trang web. Nó sử dụng các thẻ và thưộc tính để định dạng và hiển thị nội dung trên trình duyệt web. Trong phần này, chúng ta sẽ tìm hiểu về HTML cơ bản và cách tạo trang web đơn giản.

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

Để tạo một tệp HTML cơ bản, bạn cần một trình soạn thảo văn bản như Notepad (Windows) hoặc Visual Studio Code . Dưới đây là các bước cơ bản:

* Mở trình soạn thảo văn bản và tạo tệp mới.
* Đặt phần mở rộng của tệp là .html để nó trở thành tệp HTML. Ví dụ: index.html.
* Bắt đầu viết mã HTML bên trong tệp.

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

<!DOCTYPE html>

<html>

<head>

    <title>Nhóm 6\_20CN5</title>

</head>

<body>

    <h1>Chào mừng đến với Nhóm 6\_20CN5</h1>

    <p>Đây là Nhóm 6\_20CN5.</p>

    <img src="hinh-anh.jpg" alt="Hình ảnh mẫu">

    <a href="https://www.nhom6.com"> Nhóm 6\_20CN5</a>

</body>

</html>

<html>: Thẻ gốc của tài liệu HTML.

<head>: Chứa thông tin về tài liệu như tiêu đề trang.

<title>: Đặt tiêu đề của trang web.

<body>: Nơi chứa nội dung hiển thị trên trang web.

<h1>: Tiêu đề cấp 1.

<p>: Đoạn văn bản.

<img>: Hiển thị hình ảnh với thuộc tính src là đường dẫn đến hình ảnh và alt là văn bản mô tả hình ảnh.

<a>: Tạo siêu liên kết với thuộc tính href là đường dẫn đến trang khác.

# 1.3. Hướng dẫn về cú pháp và các phần tử HTML cơ bản.

Dưới đây là một số cú pháp và các phần tử HTML cơ bản:

* Một tệp HTML bắt đầu với <!DOCTYPE html> để xác định phiên bản HTML và cấu trúc tệp.
* <html> là phần tử gốc của mọi tệp HTML.
* <head> chứa thông tin về trang web như tiêu đề và liên kết đến các tệp CSS hoặc JavaScript.
* <title> đặt tiêu đề của trang web.
* <body> chứa nội dung hiển thị trên trình duyệt web.
* <h1> đến <h6> được sử dụng để đặt tiêu đề với kích thước khác nhau.
* <p> để thêm đoạn văn bản.
* <img> để hiển thị hình ảnh.
* <a> là thẻ siêu liên kết để tạo các liên kết đến các trang web khác.

Các thẻ HTML thường đi kèm với các thuộc tính để tùy chỉnh và định dạng nội dung trang web.

# 2. Cơ bản về CSS

CSS (Cascading Style Sheets) là ngôn ngữ được sử dụng để kiểm soát cách trang web được hiển thị trên trình duyệt. Trong phần này, chúng ta sẽ tìm hiểu về cách thêm kiểu cho trang web bằng CSS, tạo các lớp để áp dụng kiểu và thử nghiệm các thuộc tính CSS cơ bản.

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

* Nhúng CSS trong tệp HTML:

Có thể sử dụng thẻ <style> trong phần <head> của tệp HTML để viết mã CSS ngay trong tệp HTML.

<!DOCTYPE html>

<html>

<head>

    <style>

        h1 {

            color: blue;

        }

    </style>

</head>

<body>

    <h1>Đây là tiêu đề được định dạng bằng CSS.</h1>

</body>

</html>

* Liên kết tệp CSS bên ngoài :

Có thể viết CSS trong một tệp riêng biệt và sau đó liên kết nó với tệp HTML bằng thẻ <link>.

<!DOCTYPE html>

<html>

<head>

    <link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

    <h1>Đây là tiêu đề được định dạng bằng CSS từ tệp ngoài.</h1>

</body>

</html>

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

* Định nghĩa một lớp trong CSS:

Trong tệp CSS, bạn có thể định nghĩa một lớp bằng cách sử dụng dấu chấm . trước tên lớp.

.mylayout {

    background-color: lightgray;

    padding: 10px;

    border: 1px solid black;

}

* Áp dụng lớp cho các phần tử HTML:

Trong tệp HTML, bạn có thể áp dụng lớp bằng thuộc tính class.

<div class="mylayout">

    Đây là một phần tử với lớp CSS "mylayout".

</div>

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

* Font chữ:

Sử dụng thuộc tính font-family để thay đổi font chữ.

.mylayout {

    font-family: Arial, sans-serif;

}

* Màu sắc:

Sử dụng thuộc tính color để thay đổi màu chữ.

.mylayout {

    color: red;

}

* Độ rộng và chiều cao:

Sử dụng thuộc tính width và height để điều chỉnh kích thước của phần tử.

.mylayout {

    width: 300px;

    height: 200px;

}

# 3. JavaScript cơ bản

JavaScript là ngôn ngữ lập trình phía máy khách (client-side) mạnh mẽ được sử dụng để làm cho trang web trở nên động và tương tác.

# 3.1. Giới thiệu cú pháp JavaScript cơ bản.

JavaScript sử dụng cú pháp đơn giản và có một số yếu tố cơ bản:

* Biến (var, let, const): Dùng để lưu trữ dữ liệu.
* Hàm (function): Dùng để đóng gói một khối mã để có thể tái sử dụng.
* Câu lệnh điều kiện (if, else): Dùng để kiểm tra điều kiện và thực hiện các hành động khác nhau.
* Vòng lặp (for, while): Dùng để lặp qua danh sách hoặc thực hiện các hành động lặp lại.
* Sự kiện: Dùng để phản hồi và tương tác với người dùng, ví dụ: xử lý sự kiện click.

# 3.2. Hướng dẫn tạo biến, hàm và sử dụng sự kiện.

* Tạo biến:

Sử dụng var, let, hoặc const để khai báo biến.

var age = 20;

let name = "Minh";

const PI = 3.14;

* Tạo hàm:

Sử dụng từ khóa function để định nghĩa hàm.

function greet(name) {

    console.log("Xin chào, " + tên + "!");

}

* Sử dụng sự kiện:

Để xử lý sự kiện, bạn có thể sử dụng sự kiện onclick trong HTML hoặc addEventListener trong JavaScript.

// Sử dụng sự kiện onclick trong HTML

<button onclick="myFunction()">Nhấn vào đây</button>

// Sử dụng addEventListener trong JavaScript

document.getElementById("myButton").addEventListener("click", myFunction);

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

<!DOCTYPE html>

<html>

<head>

    <title>Ví dụ JavaScript đơn giản</title>

</head>

<body>

    <button id="myButton">Thanh toán</button>

    <script>

        // Định nghĩa một hàm JavaScript

        function showMessage() {

            alert("Cảm ơn! Bạn đã mua thành công.");

        }

        // Gán hàm vào sự kiện click của nút

        document.getElementById("myButton").addEventListener("click", showMessage);

    </script>

</body>

</html>

Trong ví dụ này, khi nút được nhấn, hàm showMessage sẽ được gọi, và một thông báo sẽ xuất hiện trên trình duyệt thông báo "Cảm ơn! Bạn đã mua thành công."

# 4. HTML Forms và Input Elements

Biểu mẫu HTML là một phần quan trọng của trang web và cho phép người dùng tương tác với trang web bằng cách nhập thông tin và gửi nó đi. Trong phần này, chúng ta sẽ tìm hiểu về cách tạo biểu mẫu HTML đơn giản với các phần tử nhập liệu như ô văn bản, ô chọn, và nút gửi, cách tạo kiểu cho biểu mẫu bằng CSS và cách sử dụng JavaScript để xử lý dữ liệu từ biểu mẫu.

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

Để tạo một biểu mẫu HTML đơn giản, có thể sử dụng các phần tử như <form>, <input>, <select>, và <button>.

<!DOCTYPE html>

<html>

<head>

    <title>Biểu mẫu đơn giản</title>

</head>

<body>

    <h1>Biểu mẫu đơn giản</h1>

    <form>

        <label for="name">Tên:</label>

        <input type="text" id="name" name="name"><br><br>

        <label for="gender">Giới tính:</label>

        <select id="gender" name="gender">

            <option value="male">Nam</option>

            <option value="female">Nữ</option>

        </select><br><br>

        <input type="submit" value="Gửi">

    </form>

</body>

</html>

* <form> được sử dụng để bao quanh toàn bộ biểu mẫu.
* Các phần tử <input> được sử dụng để nhập dữ liệu và có các thuộc tính như type, id, và name.
* Phần tử <select> tạo một danh sách chọn với các tùy chọn.
* Phần tử <label> được sử dụng để đặt nhãn cho các phần tử nhập liệu.

# 4.2. Tạo kiểu cho biểu mẫu bằng CSS

form {

    width: 300px;

    margin: 0 auto;

}

label {

    display: block;

    margin-bottom: 5px;

}

input[type="text"],

select {

    width: 100%;

    padding: 5px;

    margin-bottom: 10px;

}

input[type="submit"] {

    background-color: #007bff;

    color: #fff;

    padding: 10px 20px;

    border: none;

    cursor: pointer;

}

* Đặt chiều rộng của biểu mẫu là 300px và căn giữa nó trên trang.
* Đặt các nhãn (labels) thành kiểu hiển thị "block" để chúng xuất hiện trên các dòng riêng biệt và thêm khoảng cách dưới chúng là 5px.
* Đặt chiều rộng của trường nhập liệu và hộp chọn là 100% của phần tử cha, thêm lớp padding, và khoảng cách dưới chúng là 10px.
* Định dạng nút gửi (submit button) bằng cách đặt màu nền là xanh, màu chữ là trắng, thêm padding, loại bỏ viền, và thay đổi con trỏ chuột thành biểu tượng "bàn tay" khi di chuyển qua nút gửi.

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

<!DOCTYPE html>

<html>

<head>

    <title>Biểu mẫu đơn giản</title>

</head>

<body>

    <h1>Biểu mẫu đơn giản</h1>

    <form id="myForm">

        <label for="name">Tên:</label>

        <input type="text" id="name" name="name"><br><br>

        <label for="gender">Giới tính:</label>

        <select id="gender" name="gender">

            <option value="male">Nam</option>

            <option value="female">Nữ</option>

        </select><br><br>

        <input type="submit" value="Gửi">

    </form>

    <script>

        document.getElementById("myForm").addEventListener("submit", function(event) {

            var name = document.getElementById("name").value;

            if (name === "") {

                alert("Vui lòng nhập tên.");

                event.preventDefault(); // Ngăn chặn việc gửi biểu mẫu

            }

        });

    </script>

</body>

</html>

* Một biểu mẫu <form> đơn giản với hai trường nhập liệu và một nút gửi (<input type="submit">). Mỗi trường nhập liệu có một id để chúng ta có thể truy cập chúng bằng JavaScript.
* Dưới phần JavaScript, chúng ta sử dụng document.getElementById("myForm") để lấy tham chiếu đến biểu mẫu bằng cách sử dụng ID "myForm".
* Sử dụng addEventListener("submit", function(event) {...}) để bắt sự kiện "submit" của biểu mẫu. Khi người dùng nhấn nút "Gửi", sự kiện này sẽ được kích hoạt.
* Trong hàm xử lý sự kiện, lấy giá trị từ trường nhập liệu "name" bằng cách sử dụng document.getElementById("name").value. Sau đó, chúng ta kiểm tra xem giá trị này có trống hay không (if (name === "")).
* Nếu giá trị trống, hiển thị một thông báo cảnh báo bằng alert("Vui lòng nhập tên.") và sử dụng event.preventDefault() để ngăn chặn hành động mặc định của việc gửi biểu mẫu. Điều này có nghĩa là nếu người dùng không nhập tên và nhấn "Gửi", thì biểu mẫu sẽ không được gửi đi và một cảnh báo sẽ hiển thị.

# 5. Quy trình phát triển web

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

Quy trình phát triển web là một loạt các bước được thực hiện để xây dựng một trang web từ ý tưởng ban đầu đến khi trang web đó được triển khai và hoạt động trên internet.

* Lập kế hoạch (Planning): Bước này bao gồm việc xác định mục tiêu, yêu cầu, và phân tích khách hàng cũng như việc lên kế hoạch cho dự án.
* Thiết kế (Design): Tạo thiết kế giao diện trang web, bố cục, và trải nghiệm người dùng (UI/UX). Đây là giai đoạn để vẽ ra hình ảnh trang web và xác định cách thức hoạt động.
* Phát triển (Development): Xây dựng trang web bằng lập trình và tạo cơ sở dữ liệu nếu cần. Các ngôn ngữ như HTML, CSS, JavaScript, và các framework thường được sử dụng ở đây.
* Kiểm thử (Testing): Kiểm tra và sửa lỗi trên trang web để đảm bảo rằng nó hoạt động đúng cách trên nhiều trình duyệt và thiết bị khác nhau.
* Triển khai (Deployment): Đưa trang web lên máy chủ và làm cho nó có thể truy cập trên internet. Điều này liên quan đến việc cấu hình máy chủ và DNS.
* Bảo trì và cập nhật (Maintenance and Updates): Liên tục duy trì, bảo trì, và cập nhật trang web để đảm bảo tính bảo mật và hiệu suất tốt.

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

Quy trình phát triển web đã tuân theo các bước quan trọng như sau:

- Phân tích yêu cầu: Trước khi bắt đầu phát triển, bọn em đã phân tích yêu cầu của dự án để xác định chức năng và giao diện người dùng cần thiết.

- Thiết kế giao diện: Sau khi hiểu rõ yêu cầu, thiết kế giao diện người dùng bằng cách sử dụng HTML và CSS để tạo trang web với các thành phần như biểu mẫu, navbar, và footer.

- Phát triển: đã sử dụng React và Express.js để phát triển cả phía máy khách và phía máy chủ của ứng dụng. Các component React đã được xây dựng để tạo trải nghiệm người dùng tốt.

- Kiểm thử: Sau khi hoàn thành phát triển, đã tiến hành kiểm thử để đảm bảo rằng ứng dụng hoạt động đúng và không có lỗi. Có các bài kiểm tra (test cases) sử dụng thư viện như `supertest` để kiểm tra tính năng của các API RESTful.

- Triển khai: Trong trường hợp này, có thể thấy một số cấu hình liên quan đến triển khai và sử dụng Service Worker để làm cho ứng dụng có khả năng hoạt động offline.

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

- Quản lý dự án: Người quản lý dự án (Project Manager) có nhiệm vụ lập kế hoạch, theo dõi tiến độ, và đảm bảo rằng dự án được hoàn thành đúng hạn. Họ cũng quản lý tài nguyên và xác định ưu tiên.

- Phát triển front-end: Nhóm phát triển front-end chịu trách nhiệm xây dựng giao diện người dùng, tạo các component, và xử lý trải nghiệm người dùng trên trình duyệt.

- Phát triển back-end: Nhóm phát triển back-end xây dựng server, xử lý dữ liệu, và triển khai các API cho phía máy khách sử dụng.

- Kiểm thử: Nhóm kiểm thử kiểm tra ứng dụng để đảm bảo tính đúng đắn và hiệu suất. Họ phát hiện và báo cáo lỗi, sau đó đảm bảo rằng các lỗi này được sửa.

- Triển khai và quản lí hệ thống : Nhóm triển khai và quản lý hệ thống đảm bảo rằng ứng dụng được triển khai một cách đáng tin cậy trên môi trường sản xuất. Họ cũng quản lý cơ sở hạ tầng và bảo trì hệ thống.