



DEPT OF COMPUTER AND COMMUNICATION ENGINEERING
TT KIẾN TRÚC VÀ GIAO THỨC IOT
ITAL328264

Group: Phan Nhị Hào – 22139019

Mai Ngọc Hoàng – 22139023

Trương Lý Minh Hoàng - 22139025

HTML (tag, features)

1. HTML Headings

- Tag**

Cú pháp: <h1>Nội dung tiêu đề</h1>

- Features**

HTML headings là tiêu đề hoặc phụ đề mà bạn muốn hiển thị trên trang web.

HTML headings được xác định bằng các thẻ từ <h1> đến <h6>. Trong đó <h1> định nghĩa tiêu đề quan trọng nhất, <h6> định nghĩa tiêu đề ít quan trọng nhất.

Mỗi thẻ heading sẽ hiển thị văn bản với kích thước và độ đậm khác nhau để nhấn mạnh mức độ quan trọng của tiêu đề.

- Code and Result**

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1>Heading 1</h1> <h2>Heading 2</h2> <h3>Heading 3</h3> <h4>Heading 4</h4> <h5>Heading 5</h5> <h6>Heading 6</h6> </body></pre>	Heading 1 Heading 2 Heading 3 Heading 4 Heading 5 Heading 6

</html>	
---------	--

Mỗi thẻ heading sẽ hiển thị văn bản với kích thước và độ đậm khác nhau để nhấn mạnh mức độ quan trọng của tiêu đề.

2. HTML Paragraphs.

- **Tag**

< p > Nội dung đoạn văn < /p >

- **Features**

Thẻ < p > giúp chia nội dung thành các đoạn văn bản riêng biệt. Mỗi đoạn văn được bao quanh bởi một cặp thẻ mở < p > và thẻ đóng < /p >.

Các đoạn văn bản trong thẻ < p > thường có khoảng cách trên và dưới mặc định, tạo ra sự tách biệt rõ ràng giữa các đoạn văn.

Khi sử dụng thẻ < p >, trình duyệt sẽ tự động hiểu rằng kết thúc một đoạn văn là bắt đầu một đoạn văn khác, và sẽ xuống dòng cho đoạn văn mới.

Bên trong thẻ < p >, bạn có thể thêm các phần tử inline khác như thẻ < a >, < strong >, < em >, v.v.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>This is a paragraph.</p> <p>This is a paragraph.</p> <p>Đây là một đoạn văn với một phần chữ đậm và một liên kết.</p> </body> </html></pre>	<p>This is a paragraph.</p> <p>This is a paragraph.</p> <p>Đây là một đoạn văn với một phần chữ đậm và một liên kết.</p>

Bên trong thẻ < p >, bạn có thể thêm các phần tử inline khác như thẻ < a >, < strong >, < em >, v.v.

Code	Result
<pre data-bbox="115 198 817 783"><!DOCTYPE html> <html> <body> <p>Đây là một đoạn văn với một phần chữ đậm và một liên kết.</p> </body> </html></pre>	<p data-bbox="817 198 1530 783">Đây là một đoạn văn với một phần chữ đậm và một liên kết.</p>

Đoạn văn này chứa rất nhiều khoảng trắng trong mã nguồn, nhưng trình duyệt bỏ qua. Số dòng trong một đoạn văn phụ thuộc vào kích thước của cửa sổ trình duyệt. Nếu bạn thay đổi kích thước cửa sổ trình duyệt, số dòng trong đoạn văn này sẽ thay đổi.

Code	Result
<pre data-bbox="115 1032 817 1926"><!DOCTYPE html> <html> <body> <p> Một hai ba bốn </p> <p> Một hai ba bốn năm sáu bảy tám chín mười. Một hai ba bốn năm sáu bảy tám chín mười. Một hai ba bốn năm sáu bảy tám chín mười. Một hai ba bốn năm sáu bảy tám chín mười.</p> </body></pre>	<p data-bbox="817 1032 1530 1453">Một hai ba bốn Một hai ba bốn năm sáu bảy tám chín mười. Một hai ba bốn năm sáu bảy tám chín mười. Một hai ba bốn năm sáu bảy tám chín mười.</p>

```
</html>
```

❖ HTML Horizontal Rules

• Tag

```
<hr>
```

• Features

Thẻ `<hr>` xác định ngắt chủ đề trong trang HTML và thường được hiện thị dưới dạng một đường ngang.

Thẻ `<hr>` được sử dụng để phân tách nội dung (hoặc xác định thay đổi) trong trang HTML.

• Code and Result

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1>This is heading 1</h1> <p>This is some text.</p> <hr> <h2>This is heading 2</h2> <p>This is some other text.</p> <hr> <h2>This is heading 2</h2> <p>This is some other text.</p> </body> </html></pre>	<p>This is heading 1</p> <p>This is some text.</p> <hr/> <p>This is heading 2</p> <p>This is some other text.</p> <hr/> <p>This is heading 2</p> <p>This is some other text.</p>

HTML Line Breaks

• Tag

```
<br>
```

- **Features**

Sử dụng thẻ
 nếu bạn muốn ngắt dòng (xuống dòng mới) mà không cần bắt đầu đoạn văn mới.

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>This is
a paragraph
with line breaks.</p> </body> </html></pre>	<p>This is a paragraph with line breaks.</p>

3. HTML Styles

- **Tag**

<tagname style="property:value;">

- **Features**

HTML Style được sử dụng để thêm kiểu cho một phần tử, chẳng hạn như màu sắc, phông chữ, kích thước,...

Thẻ <style> giúp dễ dàng quản lý và áp dụng các quy tắc CSS cho trang web trực tiếp trong HTML, đặc biệt hữu ích khi bạn đang làm việc trên các dự án nhỏ hoặc cần áp dụng style một cách nhanh chóng mà không cần phải tạo file CSS riêng.

- **Code and Result**

Code	Result
------	--------

```

<!DOCTYPE html>
<html>
<body>

<p>I am normal</p>
<p style="color:red;">I am red</p>
<p style="color:blue;">I am blue</p>
<p style="font-size:50px;">I am big</p>

</body>
</html>

```

I am normal

I am red

I am blue

I am big

4. HTML Formatting

- **Tag**

**** <i> <mark> <small> <ins> <sub> <sup>

- **Description**

Các thành phần định dạng được thiết kế để hiển thị các loại văn bản đặc biệt:

****- Văn bản in đậm

****- Văn bản quan trọng

<i>- Văn bản in nghiêng

****- Văn bản được nhấn mạnh

<mark>- Văn bản được đánh dấu

<small>- Văn bản nhỏ hơn

****- Văn bản đã xóa

<ins>- Đã chèn văn bản

<sub>- Văn bản chỉ số dưới

<sup>- Văn bản chữ mũ

- **Features**

HTML Formatting thường hỗ trợ các thuộc tính cơ bản để tùy chỉnh và điều khiển cách thức hiển thị của nội dung trên trang web.

HTML Formatting chứa một số thành phần để xác định văn bản có ý nghĩa đặc biệt.

- **Code and result**

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>Văn bản này được in đậm.</p> <p>Văn bản này quan trọng. </p> <p><i>Văn bản in nghiêng.</i></p> <p>Văn bản được nhấn mạnh.</p> <p><small>Văn bản nhỏ hơn.</small></p> <p>Bôi màu kí tự <mark>HeHe.</mark></p> <p>Gạch ngang kí tự HuHu</p> <p>Gạch chân kí tự <ins>HiHi.</ins></p> <p>Chỉ kí tự <sub>xuống dưới</sub></p> <p>Chỉ kí tự <sup>mũ</sup></p> </body> </html></pre>	<p>Văn bản này được in đậm.</p> <p>Văn bản này quan trọng.</p> <p>Văn bản in nghiêng.</p> <p>Văn bản được nhấn mạnh.</p> <p>Văn bản nhỏ hơn.</p> <p>Bôi màu kí tự HeHe.</p> <p>Gạch ngang kí tự HuHu</p> <p>Gạch chân kí tự HiHi.</p> <p>Chi kí tự xuống dưới</p> <p>Chi kí tự mũ</p>

5. HTML Quotation

- **Tag**

`<blockquote>, <q>, <abbr>, <address>, <cite>, và <bdo>` HTML.

- ❖ **HTML <blockquote> for Quotations**

- **Tag**

`<blockquote>`

- **Description**

Phần tử HTML `<blockquote>` định nghĩa một phần được trích dẫn từ một nguồn khác.

Trình duyệt thường thụt lề phần tử `<blockquote>`.

- **Features**

Cite: được sử dụng để chỉ định URL của nguồn gốc đoạn trích.

Global Attributes: Bao gồm các thuộc tính như class, id, style, title, v.v. Những thuộc tính này giúp bạn tùy chỉnh và định danh thẻ <blockquote> giống như các phần tử HTML khác.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>Dưới đây là trích dẫn Dám nghĩ lớn - Đột phá tư duy, thay đổi cuộc đời trong trang web:</p> <blockquote cite="https://trichdanhay.com/dam-nghi-lon/"> Nếu chỉ để mọi ý tưởng, kế hoạch ở trong đầu mà không bắt tay thực hiện thì cả đời này chúng ta khó mà gặt hái được thành công. Một bản kế hoạch, ý tưởng hoàn hảo phải đi đôi với sự thực tế. "Nghĩ lớn" có thể là cụm từ mà rất nhiều chúng ta không dám ứng dụng vào cuộc sống, hầu như chúng ta có xu hướng ngại thay đổi và sợ những thử thách, mạo hiểm. </blockquote> <blockquote style="border-left: 4px solid #ccc; padding-left: 10px; color: #555;"> "Đây là một đoạn trích với kiểu dáng tùy chỉnh." </blockquote></pre>	<p>Dưới đây là trích dẫn Dám nghĩ lớn - Đột phá tư duy, thay đổi cuộc đời trong trang web:</p> <p>Nếu chỉ để mọi ý tưởng, kế hoạch ở trong đầu mà không bắt tay thực hiện thì cả đời này chúng ta khó mà gặt hái được thành công. Một bản kế hoạch, ý tưởng hoàn hảo phải đi đôi với sự thực tế. "Nghĩ lớn" có thể là cụm từ mà rất nhiều chúng ta không dám ứng dụng vào cuộc sống, hầu như chúng ta có xu hướng ngại thay đổi và sợ những thử thách, mạo hiểm.</p> <p> "Đây là một đoạn trích với kiểu dáng tùy chỉnh."</p>

```
</body>  
</html>
```

❖ HTML <q> for Short Quotations

• Tag

<q>

• Description

Thẻ HTML <q> là một trích dẫn ngắn.

Trình duyệt thường chèn dấu ngoặc kép quanh trích dẫn.

• Features

Cite: được sử dụng để chỉ định URL của nguồn gốc đoạn trích dẫn. Tuy nhiên, thuộc tính này không hiển thị trực tiếp trong giao diện người dùng, mà chỉ cung cấp thông tin về nguồn cho các công cụ tìm kiếm hoặc trình duyệt.

Global Attributes: hỗ trợ các thuộc tính toàn cục như:

- class: Để chỉ định một hoặc nhiều lớp CSS.
- id: Để đặt một định danh duy nhất cho phần tử.
- style: Để thêm các kiểu CSS inline cho phần tử.
- title: Để cung cấp thêm thông tin khi người dùng di chuột qua phần tử.

• Code and result

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>I said: <q>Let's go!</q></p> <p>Đây là một ví dụ khác: <q class="highlight-quote" style="font- weight:bold;">Trích dẫn nổi bật trong văn bản.</q></p></pre>	<p>I said: “Let's go!”</p> <p>Đây là một ví dụ khác: “Trích dẫn nổi bật trong văn bản.”</p>

```
</body>  
</html>
```

❖ HTML <abbr> for Abbreviations

• Tag

<abbr>

• Description

Thẻ HTML <abbr> định nghĩa một từ viết tắt hoặc từ viết tắt, như "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM".

Đánh dấu các từ viết tắt có thể cung cấp thông tin hữu ích cho trình duyệt, hệ thống dịch thuật và công cụ tìm kiếm. Khi di chuột tới sẽ hiện rõ nghĩa của từ được viết tắt.

• Features

Title: được sử dụng để cung cấp thông tin đầy đủ hoặc ý nghĩa của từ viết tắt. Khi người dùng di chuột qua từ viết tắt, nội dung của thuộc tính title sẽ được hiển thị dưới dạng một chú giải (tooltip).

Global Attributes: Thẻ <abbr> cũng hỗ trợ các thuộc tính toàn cục như:

- class: Để chỉ định một hoặc nhiều lớp CSS.
- id: Để đặt một định danh duy nhất cho phần tử.
- style: Để thêm các kiểu CSS inline cho phần tử.
- lang: Để xác định ngôn ngữ của nội dung trong thẻ.

• Code and Result

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>Định dạng <abbr title="Cascading Style Sheets">CSS</abbr> giúp cải thiện giao diện của trang web.</p></pre>	<p>Định dạng CSS giúp cải thiện giao diện của trang web.</p> <p>Tổ chức WHO đóng vai trò quan trọng trong lĩnh vực y tế toàn cầu.</p> <p>World Health Organization</p>

```

<p> Tổ chức <abbr id="who" class="highlight"
style="text-decoration: underline;" title="World
Health Organization">WHO</abbr> đóng vai
trò quan trọng trong lĩnh vực y tế toàn cầu.</p>

</body>
</html>

```

❖ HTML <address> for Contact Information

- **Tag**

<address>

- **Description**

Thẻ <address> xác định thông tin liên hệ của tác giả/chủ sở hữu của một tài liệu hoặc bài viết.

Thông tin liên lạc có thể là địa chỉ email, URL, địa chỉ thực, số điện thoại, tên người dùng trên mạng xã hội, v.v.

- **Features**

Văn bản trong <address> phần tử thường được hiển thị *nghiêng* và trình duyệt sẽ luôn thêm ngắt dòng trước và sau phần tử <address>.

- **Code and Result**

Code	Result
<pre> <!DOCTYPE html> <html> <body> <p>The HTML address element defines contact information (author/owner) of a document or article.</p> <address> Written by John Doe.
 Visit us at:
 </pre>	<p>The HTML address element defines contact information (author/owner) of a document or article.</p> <p><i>Written by John Doe.</i> <i>Visit us at:</i> <i>Example.com</i> <i>Box 564, Disneyland</i> <i>USA</i></p>

Example.com

Box 564, Disneyland

USA

</address>

</body>

</html>

❖ HTML <cite> for Work Title

- **Tag**

<cite>

- **Description**

Thẻ HTML <cite> định nghĩa tiêu đề của một tác phẩm sáng tạo (ví dụ: một cuốn sách, một bài thơ, một bài hát, một bộ phim, một bức tranh, một tác phẩm điêu khắc, v.v.).

Lưu ý: Tên của một người không phải là tiêu đề của một tác phẩm.

- **Features**

Văn bản trong phần tử <cite> thường được hiển thị theo dạng in nghiêng.

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>“Tôi đã yêu em lặng thầm, không hy vọng,</p> <p>Lúc rụt rè, khi hậm hực lòng ghen,</p> <p>Tôi yêu em, yêu chân thành đầm thắm,</p> <p>Cầu em được người tình như tôi đã yêu em”</p> <p><cite>Tôi Yêu Em</cite>, A.X.Pu-Skin, năm 1829.</p></pre>	<p>“Tôi đã yêu em lặng thầm, không hy vọng, Lúc rụt rè, khi hậm hực lòng ghen, Tôi yêu em, yêu chân thành đầm thắm, Cầu em được người tình như tôi đã yêu em” <i>Tôi Yêu Em</i>, A.X.Pu-Skin, năm 1829.</p>

```
</body>  
</html>
```

❖ HTML <bdo> for Bi-Directional Override

- **Tag**

<bdo>

- **Description**

BDO là viết tắt của Bi-Directional Override.

- **Features**

Thẻ HTML <bdo> được sử dụng để ghi đè hướng văn bản hiện tại.

Code	Result
<pre><!DOCTYPE html> <html> <body> <p>Hệ thống nhúng và IoT.</p> <p><bdo dir="rtl">Hệ thống nhúng và IoT.</bdo></p> </body> </html></pre>	Hệ thống nhúng và IoT. .ToI àv gnúhn gnóht êH

6. HMTL Colors

- **Tag**

< background-color> <color> <border-Color>

- **Description**

Màu HTML được chỉ định bằng tên màu được xác định trước hoặc bằng giá trị RGB, HEX, HSL, RGBA hoặc HSLA.

- **Features**

Color: thuộc tính này thay đổi màu chữ của nội dung bên trong thẻ HTML.

Background Color: thuộc tính này thay đổi màu nền của phần tử HTML.

Border-color: thuộc tính này thay đổi màu sắc của viền xung quanh phần tử.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1 style="color:Tomato;">Hello Baby</h1> <p style="color:DodgerBlue;">Học thật là vui.</p> <h2 style="background-color:DodgerBlue;">I love you</h2> <p style="background-color:Tomato;">Hòn hoang vắng nàng...</p> <h3 style="border: 2px solid Tomato;">Hello World</h3> <h3 style="border: 2px solid Violet;">Hello World</h3> </body> </html></pre>	<p>Hello Baby</p> <p>Học thật là vui.</p> <p>I love you</p> <p>Hòn hoang vắng nàng...</p> <p>Hello World</p> <p>Hello World</p>

7. HTML CSS

CSS là viết tắt của Cascading Style Sheets.

CSS tiết kiệm rất nhiều công sức. Nó có thể kiểm soát bố cục của nhiều trang web cùng một lúc.

CSS có thể được thêm vào tài liệu HTML theo 3 cách:

Inline: bằng cách sử dụng **style** thuộc tính bên trong các phần tử HTML.

Internal: bằng cách sử dụng một **<style>** phần tử trong **<head>** phần.

External: bằng cách sử dụng một **<link>** phần tử để liên kết đến một tệp CSS bên ngoài.

- ❖ **Inline CSS**

- **Tag**

<tagname style="property: value; property: value;"> Nội dung phần tử</tagname>

- **Description**

CSS nội tuyến được sử dụng để áp dụng một kiểu duy nhất cho một phần tử HTML duy nhất.

CSS nội tuyến sử dụng thuộc tính style của phần tử HTML.

- **Features**

Thay đổi màu sắc văn bản.

Thay đổi màu nền.

Thay đổi kích thước và kiểu chữ.

Thay đổi khoảng cách và căn lề.

Thay đổi kích thước phần tử.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html> <body> <p style="color: blue;">Đoạn văn này có màu xanh dương.</p> <div style="background-color: yellow;">Đây là một phần tử với nền màu vàng.</div> <h1 style="font-size: 24px; font-family: Arial, sans-serif;">Tiêu đề lớn với kiểu chữ Arial</h1> <p style="margin: 20px; padding: 10px; text- align: center;">Đoạn văn này có khoảng cách và căn lề được tùy chỉnh.</p> <button style="width: 150px; height: 50px;">Nút có kích thước cụ thể</button></pre>	<p>Đoạn văn này có màu xanh dương.</p> <p>Đây là một phần tử với nền màu vàng.</p> <p>Tiêu đề lớn với kiểu chữ Arial</p> <p>Đoạn văn này có khoảng cách và căn lề được tùy chỉnh.</p> <p>Nút có kích thước cụ thể</p>

```
</body>  
</html>
```

❖ Internal CSS

• Tag

Cú pháp: **<head>**

```
<style> /*Internal CSS bắt đầu từ đây*/ </style>
```

• Description

Internal CSS được sử dụng để định nghĩa kiểu cho một trang HTML duy nhất.

Internal CSS được định nghĩa trong phần **<head>** của trang HTML, bên trong phần tử **<style>**.

• Features

Thuộc tính định dạng văn bản: color, font-size, font-family, font-weight,...

Thuộc tính định dạng nền: background-color, background-image, background-size,...

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body {background-color: powderblue;} h1 {color: blue;} p {color: red;} </style> </head> <body> <h1>This is a heading</h1> <p>This is a paragraph.</p> </body> </html></pre>	

❖ External CSS

• Tag

Cú pháp: Tạo một tệp CSS (ví dụ: styles.css) chứa các quy tắc CSS. Sử dụng thẻ `<link>` trong phần `<head>` của tài liệu HTML để liên kết với tệp CSS.

- **Description**

Bảng định kiểu bên ngoài được sử dụng để xác định kiểu cho nhiều trang HTML.

Để sử dụng bảng định kiểu bên ngoài, hãy thêm liên kết đến bảng đó trong `<head>` phần của mỗi trang HTML.

- **Features**

External CSS không có thuộc tính riêng biệt so với CSS nội bộ hay inline CSS. Thay vào đó, các thuộc tính của CSS trong tệp bên ngoài hoạt động tương tự như các thuộc tính trong các phương pháp khác của CSS. Tuy nhiên, để sử dụng CSS bên ngoài, bạn cần hiểu cách liên kết tệp CSS với tài liệu HTML.

- **Code and Result**

Tạo một tệp CSS, ví dụ: styles.css, và lưu trữ các quy tắc CSS bên trong:

Code

```
/* styles.css */  
/* Định dạng cho body */  
body {  
    background-color: #f0f0f0;  
    color: #333;  
    font-family: Arial, sans-serif;  
}  
/* Định dạng cho tiêu đề h1 */  
h1 {  
    color: darkblue;  
    text-align: center;  
    font-size: 2em;  
}  
/* Định dạng cho đoạn văn p */  
p {  
    font-size: 16px;
```

```

line-height: 1.5;
margin: 10px 0;
}

/* Định dạng cho phần tử với lớp highlight */

.highlight {
background-color: yellow;
border: 1px solid black;
padding: 5px;
}

/* Định dạng cho phần tử với ID unique-element */

#unique-element {
color: red;
font-weight: bold;
}

/* Định dạng cho phần tử với lớp transition */

.transition {
transition: background-color 0.5s ease;
}

.transition:hover {
background-color: lightblue;
}

```

Trong tài liệu HTML của bạn, sử dụng thẻ <link> để liên kết với tệp CSS:

Code	Result
<pre> <!DOCTYPE html> <html lang="vi"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial- scale=1.0"> </pre>	<p style="text-align: center;">Tiêu đề với External CSS</p> <div style="background-color: yellow; padding: 5px; margin-bottom: 10px;"> Đoạn văn này có nền màu vàng và viền đen. </div> <div style="background-color: #f0f0f0; padding: 10px;"> Đoạn văn này có màu đỏ và chữ đậm. Di chuột qua đây để thấy hiệu ứng chuyển đổi. </div>

```

<title>Ví dụ về External CSS</title>
<link rel="stylesheet" href="styles.css"> <!--
Liên kết với tệp CSS -->
</head>
<body>
<h1>Tiêu đề với External CSS</h1>
<p class="highlight">Đoạn văn này có nền màu
vàng và viền đen.</p>
<p id="unique-element">Đoạn văn này có màu
đỏ và chữ đậm.</p>
<div class="transition">Di chuột qua đây để
thấy hiệu ứng chuyển đổi.</div>
</body>
</html>

```

8. HTML Links

- **Tag**

`<a>`

- **Description**

Liên kết có ở hầu hết các trang web. Liên kết cho phép người dùng nhấp chuột để chuyển từ trang này sang trang khác.

Liên kết HTML là siêu liên kết. Bạn có thể nhấp vào liên kết và chuyển đến tài liệu khác. Khi bạn di chuyển chuột qua một liên kết, mũi tên chuột sẽ biến thành hình bàn tay nhỏ.

- ❖ **HTML Links - Syntax**

- **Tag**

`link text`

- **Features**

Thuộc tính quan trọng nhất của phần tử `<a>` là thuộc tính `href`, cho biết đích đến của liên kết.

Văn bản liên kết là phần mà người đọc sẽ nhìn thấy. Nhấp vào văn bản liên kết sẽ đưa người đọc đến địa chỉ URL đã chỉ định.

- **Code and Result**

Code	Result
<pre data-bbox="115 198 816 901"><!DOCTYPE html> <html> <body> <h1>HTML Links</h1> <p>Visit W3Schools.com!</p> </body> </html></pre>	<p data-bbox="873 397 1388 475">HTML Links</p> <p data-bbox="873 551 1286 593">Visit W3Schools.com!</p>

❖ HTML Links - The target Attribute

- Tag

link text

- Description

Theo mặc định, trang được liên kết sẽ được hiển thị trong cửa sổ trình duyệt hiện tại. Để thay đổi điều này, bạn phải chỉ định một mục tiêu khác cho liên kết.

- Features

Thuộc tính mục tiêu chỉ định nơi mở tài liệu được liên kết.

Thuộc tính mục tiêu có thể có một trong các giá trị sau:

_self - Mặc định. Mở tài liệu trong cùng cửa sổ/tab khi nhấp vào.

_blank - Mở tài liệu trong cửa sổ hoặc tab mới.

_parent - Mở tài liệu trong frame parent.

_top - Mở tài liệu trong toàn bộ phần thân của cửa sổ.

- Code and Result

Code	Result
------	--------

```

<!DOCTYPE html>
<html>
<body>

<h2>The target Attribute</h2>

<a href="https://www.w3schools.com/">
target=_self>Visit W3Schools!</a>

<p>nếu target="_blank", đường dẫn sẽ mở
trong trình duyệt mới hoặc tab mới.</p>

</body>
</html>

```

Trước khi nhấn vào link:

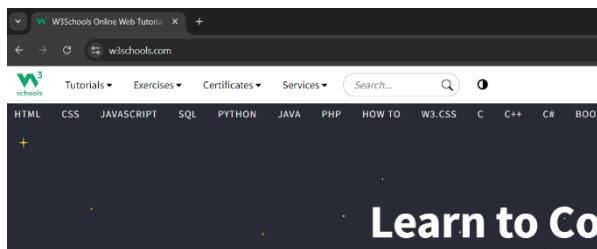


The target Attribute

[Visit W3Schools!](#)

nếu target="_blank", đường dẫn sẽ mở trong trình duyệt mới hoặc tab mới.

Sau khi nhấn vào link:



Learn to Co

❖ Absolute URLs vs. Relative URLs

• Features

Cả hai ví dụ đều sử dụng URL tuyệt đối (một địa chỉ web đầy đủ) trong thuộc tính href.

Liên kết cục bộ (liên kết đến một trang trong cùng một trang web) được chỉ định bằng URL tương đối (không có phần "https://www").

• Code and Result

Code	Result
<pre> <!DOCTYPE html> <html> <body> <p>Địa chỉ tuyệt đối</p> <p>Youtube</p> <p>Địa chỉ tương đối</p> </pre>	<p>Địa chỉ tuyệt đối</p> <p>Youtube</p> <p>Địa chỉ tương đối</p> <p>Nhạc trong Youtube</p>

```

<p><a href="https://www.youtube.com/watch?v=f9P7_qWrf38">Nhạc trong Youtube</a></p>
</body>
</html>

```

❖ HTML Links - Use an Image as a Link

- **Tag**

- **Features**

Sử dụng hình ảnh làm liên kết.

- **Code and Result**

Code	Result
<pre> <!DOCTYPE html> <html> <body> <p>Link dưới đây được biểu diễn dạng hình ảnh:</p> </body> </html> </pre>	<p>Link dưới đây được biểu diễn dạng hình ảnh</p> 

❖ Link to an Email Address

- **Tag**

 Gửi email cho tôi

- **Features**

Sử dụng mailto: bên trong thuộc tính href để tạo liên kết mở chương trình email của người dùng (để cho phép họ gửi email mới)

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1>My email:</h1> <p>Se nd email</p> </body> </html></pre>	<p>My email:</p> <p>Send email</p>

- ❖ **Button as a Link**

- **Tag**

`<button type="button">Nút Button</button>`

- **Features**

Thuộc tính tiêu đề chỉ định thông tin bổ sung về một phần tử. Thông tin thường được hiển thị dưới dạng văn bản chú giải công cụ khi chuột di chuyển qua phần tử.

Để sử dụng nút HTML làm liên kết, bạn phải thêm một số mã JavaScript.

JavaScript cho phép bạn chỉ định những gì xảy ra tại các sự kiện nhất định, chẳng hạn như nhấp vào nút.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html></pre>	Trước khi nhấn:

```

<body>

<h2>Button as a Links</h2>


Click the button to go to the HTML tutorial.


<button
    onclick="document.location='default.asp'">HTML Tutorial</button>

</body>
</html>

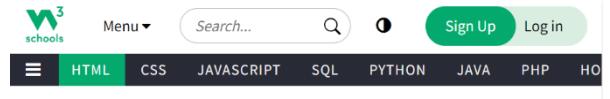
```

Button as a Links

Click the button to go to the HTML tutorial.

[HTML Tutorial](#)

Sau khi nhấn:



❖ Link Titles

- Tag

<title>

- Features

Thẻ <title> phải nằm trong phần <head> của tài liệu HTML.

Thuộc tính **title** chỉ định thông tin bổ sung về một phần tử. Thông tin thường được hiển thị dưới dạng văn bản chú giải công cụ khi chuột di chuyển qua phần tử.

- Code and Result

Code	Result
<pre> <!DOCTYPE html> <html> <body> <h1>Nhấn vào link:</h1> </pre>	<p>Nhấn vào link:</p> <p>Link</p>

```

<p><a
    href="https://www.youtube.com/watch?v=yhWCh5IVE
    04" title="Bad Bye">Link</a></p>

</body>
</html>

```

9. HTML Images

- **Tag**

 <src> <alt>

- **Description**

Thẻ HTML được sử dụng để nhúng hình ảnh vào trang web.

Về mặt kỹ thuật, hình ảnh không được chèn vào trang web; hình ảnh được liên kết đến các trang web. Thẻ tạo ra một không gian lưu trữ cho hình ảnh được tham chiếu.

- **Features**

Thẻ trống, chỉ chứa các thuộc tính và không có thẻ đóng.

Thẻ có hai thuộc tính bắt buộc:

src - Chỉ định đường dẫn đến hình ảnh

alt - Chỉ định văn bản thay thế cho hình ảnh

- ❖ **The src Attribute, alt Attribute, Size**

- **Features**

Thuộc tính src bắt buộc chỉ định đường dẫn (URL) đến hình ảnh.

Thuộc tính alt bắt buộc cung cấp văn bản thay thế cho hình ảnh, nếu vì lý do nào đó người dùng không thể xem hình ảnh (do kết nối chậm, lỗi trong thuộc tính src hoặc nếu người dùng sử dụng trình đọc màn hình).

Thuộc tính width và height luôn xác định chiều rộng và chiều cao của hình ảnh tính bằng pixel.

- **Code and Result**

Code	Result
------	--------

```

<!DOCTYPE html>
<html>
<body>

<p>This is the tiger</p>


</body>
</html>

```

This is the tiger



❖ Images in Another Folder

- Features

Nếu bạn có hình ảnh trong một thư mục con, bạn phải đưa tên thư mục vào thuộc tính src.

- Code and Result

Code	Result
<pre> <!DOCTYPE html> <html> <body> <p>This is the rhino</p> </body> </html> </pre>	<p>This is the rhino</p>

❖ Images on Another Server/Website

- **Features**

Một số trang web trả đến hình ảnh trên máy chủ khác.

Để trả đến hình ảnh trên máy chủ khác, bạn phải chỉ định URL tuyệt đối (đầy đủ) trong thuộc tính src:

Code	Result
<pre><!DOCTYPE html> <html> <body> <h2>Images on Another Server</h2> </body> </html></pre>	<p>Images on Another Server</p> 

❖ **Animated Images**

- **Feature**

HTML cho phép sử dụng ảnh GIF động.

- **Code and Result**

Code	Result

```

<!DOCTYPE html>
<html>
<body>



</body>
</html>

```



❖ Image Floating

- **Feature**

Sử dụng thuộc tính float của CSS để cho hình ảnh nổi sang bên phải hoặc bên trái của văn bản.

- **Code and Result**

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> p { padding: 20px; } </pre>	

```

</style>
</head>
<body>

<p>Tiger is in left</p>

<p>Tiger is in right</p>

</body>
</html>

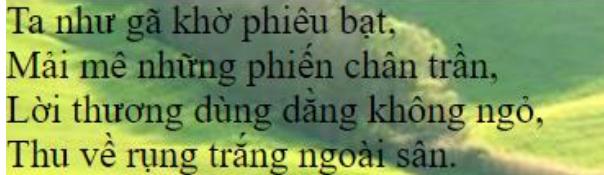
```

❖ HTML Background Images

- **Featuer**

Thuộc tính này có thể được sử dụng với nhiều thuộc tính khác nhau để kiểm soát giao diện và hành vi của hình nền.

- **Code and Result**

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> p { background-image: url('phongcanh.jpg'); } </style> </pre>	 <p>Ta như gã khờ phiêu bạt, Mái mê những phiến chân trần, Lời thương dùng dằng không ngỏ, Thu về rụng tráng ngoài sân.</p>

```

</head>
<body>
<p>Ta như gã khờ phiêu bạt,<br>
Mãi mê những phiên chân trần,<br>
Lời thương dùng dăng không ngoổ,<br>
Thu về rụng trăng ngoài sân.<br>
</p>

</body>
</html>

```

10. HTML Tables

- **Tag**

<table> <th> <tr> <td> <caption> < colgroup> <col> <thead> <tbody> <tfoot>

- **Description**

<table> - Định nghĩa một bảng.

<th> - Xác định một ô tiêu đề trong một bảng.

<tr> - Xác định một hàng trong bảng.

<td> - Xác định một ô trong bảng.

<caption> - Xác định chú thích bảng.

< colgroup> - Chỉ định một nhóm gồm một hoặc nhiều cột trong một bảng để định dạng.

<col> - Chỉ định các thuộc tính cột cho mỗi cột trong phần tử <colgroup>.

<thead> - Nhóm nội dung tiêu đề trong một bảng.

<tbody> - Nhóm nội dung phần thân trong một bảng.

<tfoot> - Nhóm nội dung chân trang trong một bảng.

- **Features**

HTML table có nhiều thuộc tính để tùy chỉnh và định dạng các bảng, bao gồm cả cấu trúc bảng, các hàng và các ô.

- **Code and Result**

Code	Result								
<pre> <!DOCTYPE html> <html> <style> table, th, td { border:1px solid black; } </style> <body> <h2>Thông tin cơ bản</h2> <table style="width:100%"> <tr> <th>Họ tên</th> <th>Năm sinh</th> </tr> <tr> <td>Trương Lý Minh Hoàng</td> <td>2004</td> </tr> <tr> <td>Phan Nhị Hào</td> <td>2004</td> </tr> <tr> <td>Mai Ngọc Hoàng</td> <td>2004</td> </pre>	<p>Thông tin cơ bản</p> <table border="1"> <thead> <tr> <th data-bbox="866 1064 1302 1094">Họ tên</th><th data-bbox="1302 1064 1465 1094">Năm sinh</th></tr> </thead> <tbody> <tr> <td data-bbox="866 1094 1302 1121">Trương Lý Minh Hoàng</td><td data-bbox="1302 1094 1465 1121">2004</td></tr> <tr> <td data-bbox="866 1121 1302 1148">Phan Nhị Hào</td><td data-bbox="1302 1121 1465 1148">2004</td></tr> <tr> <td data-bbox="866 1148 1302 1176">Mai Ngọc Hoàng</td><td data-bbox="1302 1148 1465 1176">2004</td></tr> </tbody> </table>	Họ tên	Năm sinh	Trương Lý Minh Hoàng	2004	Phan Nhị Hào	2004	Mai Ngọc Hoàng	2004
Họ tên	Năm sinh								
Trương Lý Minh Hoàng	2004								
Phan Nhị Hào	2004								
Mai Ngọc Hoàng	2004								

```

</tr>
</table>

</body>
</html>

```

❖ Collapsed Table Borders

- Tag

```

table {
    border-collapse: collapse;
}

```

- Features

Để tránh có đường viền kép, đặt thuộc tính border-collapse của CSS thành collapse. Thao tác này sẽ khiến các đường viền thu gọn thành một đường viền duy nhất.

- Code and Result

Code	Result								
<pre> <!DOCTYPE html> <html> <style> table { border-collapse: collapse; width: 100%; } table, th, td { border: 1px solid black; } </style> <body> </pre>	<p>Thông tin cơ bản</p> <table border="1"> <thead> <tr> <th>Họ tên</th><th>Năm sinh</th></tr> </thead> <tbody> <tr> <td>Truong Ly Minh Hoang</td><td>2004</td></tr> <tr> <td>Phan Nhị Hào</td><td>2004</td></tr> <tr> <td>Mai Ngoc Hoang</td><td>2004</td></tr> </tbody> </table>	Họ tên	Năm sinh	Truong Ly Minh Hoang	2004	Phan Nhị Hào	2004	Mai Ngoc Hoang	2004
Họ tên	Năm sinh								
Truong Ly Minh Hoang	2004								
Phan Nhị Hào	2004								
Mai Ngoc Hoang	2004								

```

<h2>Thông tin cơ bản</h2>



| Họ tên               | Năm sinh |
|----------------------|----------|
| Trương Lý Minh Hoàng | 2004     |
| Phan Nhị Hào         | 2004     |
| Mai Ngọc Hoàng       | 2004     |



</body>
</html>

```

❖ Style Table Borders

- **Feature**

Có thể thiết lập màu nền cho mỗi ô và tô màu cho đường viền.

- **Code and Result**

Code	Result								
<pre> <!DOCTYPE html> <html> <style> table { border-collapse: collapse; width: 100%; } th, td { border: 2px solid #4CAF50; padding: 12px; text-align: center; } th { background-color: tomato; color: purple; } td { background-color: powderblue; /* Màu nền của ô dữ liệu */ } table, th, td { border:1px solid black; } </style> <body> </pre>	<p style="text-align: center;">Thông tin cơ bản</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #ff9999;"> <th style="text-align: center;">Họ tên</th> <th style="text-align: center;">Năm sinh</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Trương Lý Minh Hoàng</td> <td style="text-align: center;">2004</td> </tr> <tr> <td style="text-align: center;">Phan Nhị Hào</td> <td style="text-align: center;">2004</td> </tr> <tr> <td style="text-align: center;">Mai Ngọc Hoàng</td> <td style="text-align: center;">2004</td> </tr> </tbody> </table>	Họ tên	Năm sinh	Trương Lý Minh Hoàng	2004	Phan Nhị Hào	2004	Mai Ngọc Hoàng	2004
Họ tên	Năm sinh								
Trương Lý Minh Hoàng	2004								
Phan Nhị Hào	2004								
Mai Ngọc Hoàng	2004								

```
<h2>Thông tin cơ bản</h2>
```

```
<table style="width:100%">
<tr>
<th>Họ tên</th>
<th>Năm sinh</th>
</tr>
<tr>
<td>Trương Lý Minh Hoàng</td>
<td>2004</td>
</tr>
<tr>
<td>Phan Nhị Hào</td>
<td>2004</td>
</tr>
<tr>
<td>Mai Ngọc Hoàng</td>
<td>2004</td>
</tr>
</table>

</body>
</html>
```

❖ Round Table Borders

- Tag

```
<border-radius>
```

- Features

border-radius sẽ bo tròn các cạnh ngoài của bảng, nhưng không thể áp dụng trực tiếp cho các đường viền giữa các ô.

Nếu bạn muốn các ô riêng lẻ có đường viền tròn, bạn cần phải sử dụng thuộc tính border-radius cho các ô cụ thể (ví dụ: thẻ <th>, <td>), nhưng kết quả có thể không hoàn hảo vì các cạnh của ô có thể không khớp chính xác với nhau.

- **Code and Result**

Code	Result								
<pre><!DOCTYPE html> <html> <style> table { border-radius: 12px; border-collapse: separate; border-spacing: 0; overflow: hidden; } th, td { border: 2px solid #4CAF50; padding: 12px; text-align: center; } th { background-color: tomato; color: purple; } td { background-color: powderblue; } table, th, td {</pre>	<p>Thông tin cơ bản</p> <table border="1"><thead><tr><th>Họ tên</th><th>Năm sinh</th></tr></thead><tbody><tr><td>Trương Lý Minh Hoàng</td><td>2004</td></tr><tr><td>Phan Nhị Hào</td><td>2004</td></tr><tr><td>Mai Ngọc Hoàng</td><td>2004</td></tr></tbody></table>	Họ tên	Năm sinh	Trương Lý Minh Hoàng	2004	Phan Nhị Hào	2004	Mai Ngọc Hoàng	2004
Họ tên	Năm sinh								
Trương Lý Minh Hoàng	2004								
Phan Nhị Hào	2004								
Mai Ngọc Hoàng	2004								

```
border:1px solid black;  
}  
</style>  
<body>  
  
<h2>Thông tin cơ bản</h2>  
  
<table style="width:100%">  
<tr>  
<th>Họ tên</th>  
<th>Năm sinh</th>  
</tr>  
<tr>  
<td>Trương Lý Minh Hoàng</td>  
<td>2004</td>  
</tr>  
<tr>  
<td>Phan Nhị Hào</td>  
<td>2004</td>  
</tr>  
<tr>  
<td>Mai Ngọc Hoàng</td>  
<td>2004</td>  
</tr>  
</table>  
  
</body>  
</html>
```

❖ Dotted Table Borders

- **Tag**

<dotted> <dashed> <solid> <double> <groove> <ridge> <inset> <outset> <none> <hidden>

Cú pháp:

```
table, th, td {  
    border: 2px Kiểu đường viền;  
}
```

- **Description**

Các giá trị sau đây được phép:

- **dotted** 
- **dashed** 
- **solid** 
- **double** 
- **groove** 
- **ridge** 
- **inset** 
- **outset** 
- **none** 
- **hidden** 

- **Feature**

Với thuộc tính border-style, bạn có thể thiết lập giao diện của đường viền.

- **Code and Result**

Code	Result								
<pre><!DOCTYPE html> <html> <style> table {</pre>	<p>Thông tin cơ bản</p> <table border="1"><thead><tr><th>Họ tên</th><th>Năm sinh</th></tr></thead><tbody><tr><td>Trương Lý Minh Hoàng</td><td>2004</td></tr><tr><td>Phan Nhị Hào</td><td>2004</td></tr><tr><td>Mai Ngọc Hoàng</td><td>2004</td></tr></tbody></table>	Họ tên	Năm sinh	Trương Lý Minh Hoàng	2004	Phan Nhị Hào	2004	Mai Ngọc Hoàng	2004
Họ tên	Năm sinh								
Trương Lý Minh Hoàng	2004								
Phan Nhị Hào	2004								
Mai Ngọc Hoàng	2004								

```
border-collapse: collapse;  
width: 100%;  
}  
  
th, td {  
border: 2px dashed #4CAF50;  
padding: 12px;  
text-align: center;  
}  
th {  
background-color: tomato;  
color: purple;  
}  
td {  
background-color: powderblue; /* Màu nền của  
ô dữ liệu */  
}  
table, th, td {  
border: 1px dashed black;  
}  
</style>  
<body>  
  
<h2>Thông tin cơ bản</h2>  
  
<table style="width:100%">  
<tr>  
<th>Họ tên</th>  
<th>Năm sinh</th>  
</tr>
```

```

<tr>
<td>Trương Lý Minh Hoàng</td>
<td>2004</td>
</tr>
<tr>
<td>Phan Nhị Hào</td>
<td>2004</td>
</tr>
<tr>
<td>Mai Ngọc Hoàng</td>
<td>2004</td>
</tr>
</table>

</body>
</html>

```

❖ Table Sizes

- **Features**

HTML Table Width: có thể thiết lập chiều rộng của bảng, bằng cách thêm thuộc tính style vào phần tử `<table>`.

HTML Table Column Width: có thiết lập kích thước của một cột cụ thể, bằng cách thêm thuộc tính style vào phần tử `<th>` hoặc `<td>`.

HTML Table Row Height: có thiết lập chiều cao của một hàng cụ thể, bằng cách thêm thuộc tính style vào phần tử hàng của bảng.

Code	Result
------	--------

```

<!DOCTYPE html>
<html>
<style>
table, th, td {
border:1px solid black;
border-collapse: collapse;
}
</style>

<body>

<h1>Bảng thông tin cơ bản</h1>

<table style="width: 70%;">

<thead>
<tr style="height: 50px;">
<th style="width: 10%;">STT</th>
<th>Họ tên</th>
<th style="width: 40%;">Học phí</th>
</tr>
</thead>

<tbody>
<tr>
<tr style="height: 50px;">
<td>1</td> <td>Mai Ngọc Hoàng</td>
<td>19.500.000</td>
</tr>
<tr>

```

Bảng thông tin cơ bản

STT	Họ tên	Học phí
1	Mai Ngọc Hoàng	19.500.000
2	Trương Lý Minh Hoàng	21.000.000
3	Phan Nhị Hào	20.000.000
Tổng học phí		60,500,000

```

<tr style="height: 30px;">
<td style="width: %;">2</td> <td>Trương Lý
Minh Hoàng</td>
<td>21.000.000</td>
</tr>
<tr>
<td style="height: 100px;">
<td>3</td> <td>Phan Nhị Hào</td>
<td>20.000.000</td>
</tr>
</tbody>

<tfoot>
<tr>
<th colspan="2">Tổng học phí</th>
<th>60,500,000</th>
</tr>
</tfoot>

</table>
</body>
</html>

```

❖ HTML Table Headers

- **Features**

Align Table Headers: theo mặc định, tiêu đề bảng được in đậm và căn giữa. Để căn trái tiêu đề bảng, sử dụng thuộc tính text-align của CSS.

Table Caption: có thể thêm chú thích đóng vai trò là tiêu đề cho toàn bộ bảng, để thêm chú thích vào bảng, sử dụng thẻ <caption>.

Code	Result
------	--------

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
table, th, tr, td {
```

```
border: 1px solid black;
```

```
border-collapse: collapse;
```

```
}
```

```
th {
```

```
text-align: left;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<table style="width: 40%;">
```

```
<caption>Phí chi tiêu hàng tháng</caption>
```

```
<thead>
```

```
<tr>
```

```
<th>Tháng</th>
```

```
<th>Phí chi tiêu</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<td>Tháng 8</td>
```

Chi phí chi tiêu hàng tháng

Tháng	Phí chi tiêu
Tháng 8	1000\$
Tháng 9	2000\$

```

<td>1000$</td>
</tr>
<tr>
<td>Tháng 9</td>
<td>2000$</td>
</tr>
</tbody>

</table>
</body>
</html>

```

❖ HTML Table Padding & Spacing

- **Features**

HTML Table - Cell Padding: để thêm phần đệm vào các ô trong bảng, sử dụng thuộc tính padding của CSS. Các thuộc tính có thể dùng là padding-top, padding-bottom, padding-left, and padding-right.

HTML Table - Cell Spacing: khoảng cách giữa các ô là khoảng cách giữa mỗi ô. Để thay đổi khoảng cách giữa các ô trong bảng, sử dụng thuộc tính border-spacing của CSS trên phần tử bảng.

Code	Result								
<pre> <!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; } th, td, tr { </pre>	<table border="1"> <thead> <tr> <th colspan="2">Chi phí chi tiêu hàng tháng</th> </tr> <tr> <th>Tháng</th> <th>Phi chi tiêu</th> </tr> </thead> <tbody> <tr> <td>Tháng 8</td> <td>1000\$</td> </tr> <tr> <td>Tháng 9</td> <td>2000\$</td> </tr> </tbody> </table>	Chi phí chi tiêu hàng tháng		Tháng	Phi chi tiêu	Tháng 8	1000\$	Tháng 9	2000\$
Chi phí chi tiêu hàng tháng									
Tháng	Phi chi tiêu								
Tháng 8	1000\$								
Tháng 9	2000\$								

```
padding-top: 10px;  
padding-left: 40px;  
padding-bottom: 30px;  
padding-right: 40px;  
}  
  
table {  
border-spacing: 30px;  
}  
  
</style>  
</head>  
<body>  
<table style="width: 70%;">  
<caption>Chi phí chi tiêu hàng tháng</caption>  
<thead>  
<tr>  
<th>Tháng</th>  
<th>Phí chi tiêu</th>  
</tr>  
</thead>  
<tbody>  
<tr>  
<td>Tháng 8</td>  
<td>1000$</td>  
</tr>  
<tr>  
<td>Tháng 9</td>  
<td>2000$</td>  
</tr>  
</tbody>  
</table>
```

```
</body>  
</html>
```

❖ HTML Table Colspan & Rowspan

• Features

HTML Table – Colspan: để tạo một ô trải dài trên nhiều cột, hãy sử dụng thuộc tính colspan.

HTML Table – Rowspan: để tạo một ô trải dài trên nhiều hàng, hãy sử dụng thuộc tính rowspan.

Code	Result											
<pre><!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; border-collapse: collapse; } </style> </head> <body> <table style="width: 40%;"> <caption>Chi phí chi tiêu hàng tháng</caption> <thead> <tr> <th>Tháng</th> <th>Phí chi tiêu</th> <th>Tổng</th> </tr> </thead></pre>	<p>Chi phí chi tiêu hàng tháng</p> <table border="1"><thead><tr><th>Tháng</th><th>Phí chi tiêu</th><th>Tổng</th></tr></thead><tbody><tr><td>Tháng 8</td><td>1000\$</td><td rowspan="2">3000\$</td></tr><tr><td>Tháng 9</td><td>2000\$</td></tr><tr><td>Tổng chi phí</td><td>3000\$</td><td></td></tr></tbody></table>	Tháng	Phí chi tiêu	Tổng	Tháng 8	1000\$	3000\$	Tháng 9	2000\$	Tổng chi phí	3000\$	
Tháng	Phí chi tiêu	Tổng										
Tháng 8	1000\$	3000\$										
Tháng 9	2000\$											
Tổng chi phí	3000\$											

```

</thead>
<tbody>
<tr>
<td>Tháng 8</td>
<td>1000$</td>
<td rowspan="2" style="text-align: center;">3000$</td>
</tr>
<tr>
<td>Tháng 9</td>
<td>2000$</td>
</tr>
<tr>
<td>Tổng chi phí</td>
<td colspan="2" style="text-align: center;">3000$</td>
</tr>
</tbody>
</table>
</body>
</html>

```

11. HTML Lists

HTML Lists dùng nhóm các mục liên quan với nhau trong một cấu trúc rõ ràng. Có ba loại danh sách chính trong HTML: Ordered Lists, Unordered Lists và Description Lists.

❖ Ordered Lists

- **Tag**

Danh sách có thứ tự bắt đầu bằng thẻ ****. Mỗi mục danh sách bắt đầu bằng thẻ ****.

- **Description**

Được sử dụng để tạo danh sách trong đó thứ tự các mục rất quan trọng, thường được hiển thị bằng số.

- **Features**

Thuộc tính **type** của thẻ **** xác định loại đánh dấu mục danh sách:

type= “1”: Các mục danh sách sẽ được đánh số bằng số (mặc định);

type= “A”: Các mục danh sách sẽ được đánh số bằng chữ in hoa;

type= “a”: Các mục danh sách sẽ được đánh số bằng chữ thường;

type= “I”: Các mục danh sách sẽ được đánh số bằng số La Mã in hoa;

type= “i”: Các mục danh sách sẽ được đánh số bằng số La Mã in thường.

- **Code and Result**

Các mục trong danh sách sẽ được đánh dấu bằng số theo mặc định:

Code	Result
<pre><html> <body> <ol type="1"> Coffee Tea Milk </body> </html></pre>	<p>1. Coffee 2. Tea 3. Milk</p>

- ❖ **Unordered Lists**

- **Tag**

Danh sách có thứ tự bắt đầu bằng thẻ ****. Mỗi mục danh sách bắt đầu bằng thẻ ****.

- **Description**

Được sử dụng để tạo danh sách trong đó thứ tự các mục không quan trọng, thường được hiển thị bằng dấu đầu dòng.

- **Features**

Bullet Type (list-style-type in CSS): Thay đổi loại dấu đầu dòng (ví dụ: disc, circle, square);

Nested List: Tạo danh sách con trong các mục danh sách;

Horizontal List with CSS: Danh sách HTML có thể được định dạng theo nhiều cách khác nhau bằng CSS.

- **Code and Result**

Thuộc tính CSS list-style-type được sử dụng để xác định kiểu của mục đánh dấu danh sách:

Code	Result
<pre><html> <body> <ul style="list-style-type:disc;"> Coffee Tea Milk </body> </html></pre>	<ul style="list-style-type: none">• Coffee• Tea• Milk

Thuộc tính Nested List tạo danh sách mà nó có thể được lồng nhau (danh sách bên trong danh sách)

Code	Result
<pre><html> <body> <h2>A Nested List</h2> <p>Lists can be nested (list inside list):</p></pre>	<ul style="list-style-type: none">• Coffee• Tea<ul style="list-style-type: none">◦ Black tea◦ Green tea• Milk

```

<ul>
<li>Coffee</li>
<li>Tea
<ul>
<li>Black tea</li>
<li>Green tea</li>
</ul>
</li>
<li>Milk</li>
</ul>

</body>
</html>

```

Thuộc tính Horizontal List with CSS định dạng danh sách theo chiều ngang để tạo menu điều hướng:

Code	Result
<pre> <html> <head> <style> ul { list-style-type: none; margin: 0; padding: 0; overflow: hidden; background-color: #333333; } li { float: left; </pre>	<p>Navigation Menu</p> <p>In this example, we use CSS to style the list horizontally, to create a navigation menu:</p> 

```
}
```



```
li a {
```

```
    display: block;
```

```
    color: white;
```

```
    text-align: center;
```

```
    padding: 16px;
```

```
    text-decoration: none;
```

```
}
```

```
li a:hover {
```

```
    background-color: #111111;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

<h2>Navigation Menu</h2>

<p>In this example, we use CSS to style the list horizontally, to create a navigation menu:</p>

```
<ul>
```

```
<li><a href="#home">Home</a></li>
```

```
<li><a href="#news">News</a></li>
```

```
<li><a href="#contact">Contact</a></li>
```

```
<li><a href="#about">About</a></li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

❖ Description Lists

• Tag

Thẻ `<dl>` định nghĩa danh sách mô tả, thẻ `<dt>` định nghĩa thuật ngữ (tên) và thẻ `<dd>` mô tả từng thuật ngữ.

• Description

Được sử dụng để tạo danh sách các thuật ngữ có mô tả tương ứng.

• Features

Flexible Pairing: có thể có nhiều mô tả (`<dd>`) cho một thuật ngữ (`<dt>`) hoặc bạn có thể liệt kê nhiều thuật ngữ theo sau là mô tả của chúng;

Nested Description Lists: có thể lồng danh sách mô tả trong `<dd>` để tạo ra các cấu trúc phức tạp hơn.

• Code and Result

Flexible Pairing

Code	Result
<pre><html> <body> <dl> <dt>Term 1</dt> <dd>Description for term 1</dd> <dd>Additional information for term 1</dd> <dt>Term 2</dt> <dt>Term 3</dt> <dd>Description for term 2 and term 3</dd> </dl> <html> <body></pre>	<hr/> <p>Term 1</p> <p>Description for term 1</p> <p>Additional information for term 1</p> <p>Term 2</p> <p>Term 3</p> <p>Description for term 2 and term 3</p>

Nested Description Lists

Code	Result
<pre><html> <body> <dl> <dt>Fruit</dt> <dd> A sweet, edible part of a plant. <dl> <dt>Apple</dt> <dd>A round fruit with red or green skin.</dd> <dt>Banana</dt> <dd>A long curved fruit with a yellow skin.</dd> </dl> </dd> <dt>Vegetable</dt> <dd>An edible plant or part of a plant, other than a fruit or seed.</dd> </dl> <html> <body></pre>	<p>Fruit A sweet, edible part of a plant.</p> <hr/> <p>Apple A round fruit with red or green skin.</p> <p>Banana A long curved fruit with a yellow skin.</p> <p>Vegetable An edible plant or part of a plant, other than a fruit or seed.</p>

12. HTML Block &Inline Elements

Mỗi phần tử HTML đều có giá trị hiển thị mặc định, tùy thuộc vào loại phần tử đó. Hai giá trị hiển thị phổ biến nhất là block và inline.

❖ Block-Level Elements

• Tag

[`<address>`](#) [`<article>`](#) [`<aside>`](#) [`<blockquote>`](#) [`<canvas>`](#) [`<dd>`](#) [`<div>`](#) [`<dl>`](#) [`<dt>`](#) [`<fieldset>`](#)
[`<figcaption>`](#) [`<figure>`](#) [`<footer>`](#) [`<form>`](#) [`<h1>-<h6>`](#) [`<header>`](#) [`<hr>`](#) [``](#) [`<main>`](#) [`<nav>`](#)
[`<noscript>`](#) [``](#) [`<p>`](#) [`<pre>`](#) [`<section>`](#) [`<table>`](#) [`<tfoot>`](#) [``](#) [`<video>`](#)

Hai phần tử khôi thường được sử dụng là: `<p>` and `<div>`.

Phần tử `<p>` xác định một đoạn văn trong tài liệu HTML.

Phần tử `<div>` xác định một phần hoặc một phần trong tài liệu HTML.

• Description

Một phần tử cấp khôi luôn bắt đầu trên một dòng mới và trình duyệt tự động thêm một số khoảng trắng (lè) trước và sau phần tử;

Một phần tử cấp khôi luôn chiếm toàn bộ chiều rộng có sẵn (kéo dài sang trái và phải xa nhất có thể).

• Features

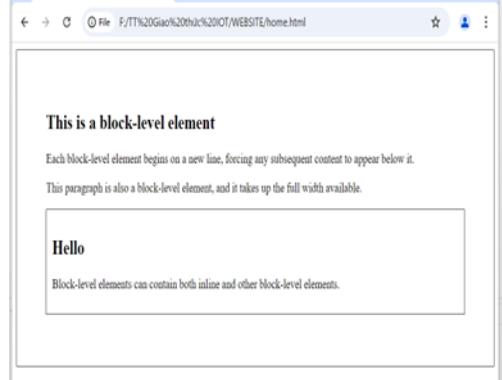
Occupies Full Width: Theo mặc định, phần tử cấp khôi trải dài đến toàn bộ chiều rộng của vùng chứa chính của nó;

Starts on a New Line: Mỗi phần tử cấp khôi bắt đầu trên một dòng mới, buộc mọi nội dung tiếp theo phải xuất hiện bên dưới nó;

Contains Other Elements: Phần tử cấp khôi có thể chứa cả phần tử nội tuyến và phần tử cấp khôi khác;

Margin and Padding: Có thể dễ dàng áp dụng lè và khoảng đệm cho các phần tử cấp khôi, ảnh hưởng đến không gian xung quanh và bên trong chúng.

• Code and Result

Code	Result
<pre> <div style="border: 1px solid black; padding: 50px; margin: 1px 0;"> <h2>This is a block-level element</h2> <p>Each block-level element begins on a new line, forcing any subsequent content to appear below it.</p> <p>This paragraph is also a block-level element, and it takes up the full width available.</p> </pre>	

```

<div style="border: 1px solid black; padding: 10px; margin: 10px 0;">
<h2>Hello</h2>
<p>Block-level elements can contain both inline and other block-level elements.</p>
</div>
</div>

```

❖ Inline Elements

- **Tag**

- **Description**

Một Inline element không bắt đầu trên một dòng mới;

Inline elements chỉ chiếm chiều rộng nếu cần thiết.

- **Features**

Occupies Only Necessary Width: Các phần tử nội tuyến chỉ chiếm chiều rộng cần thiết để hiển thị nội dung của chúng;

Stays Inline with Text: Các phần tử này vẫn nằm trong luồng nội dung xung quanh và không buộc nội dung khác phải chuyển sang một dòng mới;

Cannot Contain Block-Level Elements: Các phần tử nội tuyến chỉ có thể chứa các phần tử nội tuyến hoặc văn bản khác.

- **Code and Result**

Code	Result
<pre> <html> <body> <p>This is an inline span Hello </pre>	<p>This is an inline span Hello WorldHello HEHE element inside a paragraph.</p> <p>The SPAN element is an inline element, and will not start on a new line and only takes up as much width as necessary.</p> <p>Inline elements can only contain other inline elements or text.</p>

```
World<span style="border: 1px solid blue">Hello HEHE</span></span>  
element inside a paragraph.</p>
```

```
<p>The SPAN element is an inline element, and will not start on a new line and only takes up as much width as necessary.</p>
```

```
<p>Inline elements can only contain other inline elements or text.</p>
```

```
</body>  
</html>
```

13. HTML Iframes

- **Tag**

<iframe>

- **Description**

HTML Iframes được sử dụng để hiển thị một trang web trong một trang web.

- **Features**

Embed External Content: Tải nội dung từ trang web hoặc tài liệu khác trực tiếp trong trang của bạn;

Responsive Display: Kích thước của iframe có thể được điều chỉnh bằng CSS để làm cho nó phản hồi.

- **Code and Result**

Code	Result
------	--------

```

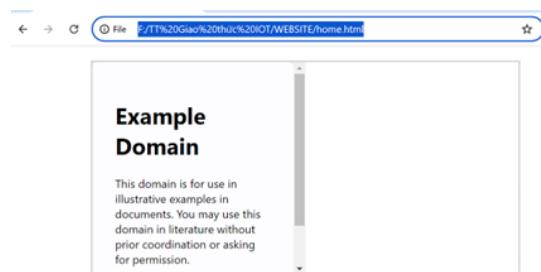
<html lang="en">
<head>
<style>
.iframe-container {
width: 100%;
max-width: 600px;
margin: 20px auto;
border: 2px solid #ccc;
}

iframe {
width: 50%;
height: 300px;
border: 0;
}
</style>
</head>
<body>

<div class="iframe-container">
<iframe src="https://www.example.com"
title="Example Iframe"></iframe>
</div>

</body>
</html>

```



14. HTML Javascripts

- Tag

<script>

- Description

Thẻ HTML `<script>` được sử dụng để định nghĩa một tập lệnh phía máy khách (JavaScript);
Phần tử `<script>` chứa các câu lệnh tập lệnh hoặc trỏ đến một tệp tập lệnh bên ngoài thông qua thuộc tính `src`.

- **Features**

Event Handling: Có thể xử lý các tương tác của người dùng như nhấp chuột, gửi biểu mẫu và nhấn phím để kích hoạt các hành động cụ thể. Animations and Effects: Có thể tạo hoạt ảnh và hiệu ứng hình ảnh để nâng cao giao diện người dùng.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>JavaScript Animation Example</title> <style> #box { width: 100px; height: 100px; background-color: red; position: absolute; top: 0; left: 0; } </style> </head> <body> <h1>JavaScript Animation Example</h1></pre>	<p>JavaScript Animation Example</p> <p>Start Animation</p> 

```

<button onclick="startAnimation()">Start
Animation</button>
<div id="box"></div>

<script>
function startAnimation() {
let box = document.getElementById('box');
let position = 0;
let interval = setInterval(function() {
if (position >= 500) {
clearInterval(interval);
} else {
position++;
box.style.left = position + 'px';
}
}, 5); // Move every 5 milliseconds
}
</script>

</body>
</html>

```

15. HTML Head

- **Tag**

<head>

- **Description**

Phần tử **<head>** của HTML là nơi chứa các phần tử sau: **<title>**, **<style>**, **<meta>**, **<link>**, **<script>** và **<base>**;

Nội dung bên trong thẻ <head> không được hiển thị trên chính trang web, nhưng nó rất cần thiết để thiết lập trang, liên kết đến bảng định kiểu, tập lệnh và cung cấp siêu dữ liệu mà công cụ tìm kiếm và trình duyệt sử dụng.

- **Features**

Title of the Document: Thẻ <head> chứa thẻ <title> đặt tiêu đề cho trang web, hiển thị trong tab trình duyệt;

Metadata: Sử dụng thẻ <meta>, <head> cung cấp siêu dữ liệu như mã hóa ký tự, tác giả, mô tả, từ khóa, cài đặt chế độ xem cho thiết kế đáp ứng, v.v;

Stylesheets and Scripts: <head> có thể liên kết với các bảng định kiểu CSS và tệp JavaScript bên ngoài bằng cách sử dụng thẻ <link> và <script>.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <meta name="description" content="A simple webpage example"> <meta name="author" content="Your Name"> <title>My Webpage</title> <link rel="stylesheet" href="styles.css"> <script src="script.js" defer></script> <link rel="icon" href="favicon.ico" type="image/x-icon"> </head> <body> <h1>Welcome to My Webpage</h1></pre>	<p>Welcome to My Webpage</p> <p>This is an example of a simple webpage.</p>

```
<p>This is an example of a simple webpage.</p>
```

```
</body>
```

```
</html>
```

16. HTML Layout

- **Tag**

Một số tags phổ biến cho Layout:

<header> - Định nghĩa tiêu đề cho một tài liệu hoặc một phần;

<nav> - Định nghĩa một tập hợp các liên kết điều hướng;

<section> - Định nghĩa một phần trong tài liệu;

<article> - Định nghĩa một nội dung độc lập, khép kín;

<aside> - Định nghĩa nội dung ngoài nội dung (như thanh bên);

<footer> - Định nghĩa footer cho một tài liệu hoặc một phần;

<details> - Định nghĩa các chi tiết bổ sung mà người dùng có thể mở và đóng theo yêu cầu;

<summary> - Định nghĩa tiêu đề cho phần tử **<details>**.

- **Description**

Mục đích chính của các thẻ Layout là cấu trúc và sắp xếp nội dung của trang web theo cách có ý nghĩa, vừa về mặt bố cục trực quan vừa cải thiện khả năng truy cập.

- **Features**

Semantic Structure: Các thẻ như **<header>**, **<footer>**, **<section>** và **<nav>** cung cấp ý nghĩa và ngữ cảnh cho các phần khác nhau của trang, cải thiện khả năng đọc và SEO.

Flexibility: Các thẻ này có thể được kết hợp với CSS (bao gồm Flexbox, Grid và các kỹ thuật bố cục khác) để tạo bố cục phức tạp.

- **Code and Result**

Code	Result
------	--------

```

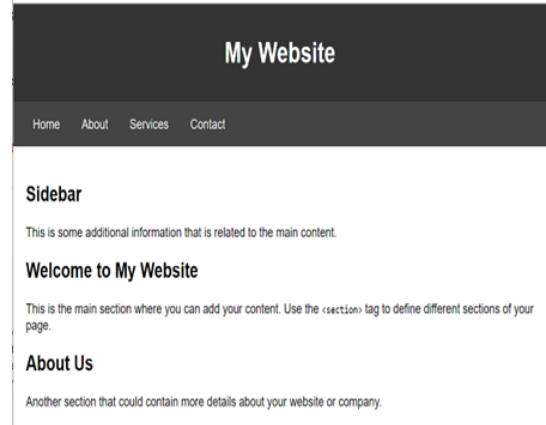
<!DOCTYPE html>

<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Simple HTML Layout</title>
<style>
body {
font-family: Arial, sans-serif;
margin: 0;
padding: 0;
display: flex;
flex-direction: column;
min-height: 100vh;
}

header, footer {
background-color: #333;
color: white;
text-align: center;
padding: 1em;
}

nav {
background-color: #444;
color: white;
padding: 1em;
}

```



```
nav a {  
    color: white;  
    margin: 0 15px;  
    text-decoration: none;  
}  
  
main {  
    flex: 1;  
    padding: 20px;  
}  
  
section {  
    margin-bottom: 20px;  
}  
  
footer {  
    background-color: #222;  
}  
</style>  
</head>  
<body>  
  
<header>  
<h1>My Website</h1>  
</header>  
  
<nav>  
<a href="#home">Home</a>  
<a href="#about">About</a>  
<a href="#services">Services</a>
```

```
<a href="#contact">Contact</a>
</nav>

<main>
<aside>
<h2>Sidebar</h2>
<p>This is some additional information that is
related to the main content.</p>
</aside>

<section>
<h2>Welcome to My Website</h2>
<p>This is the main section where you can add
your content. Use the
<code>&lt;section&gt;</code> tag to define
different sections of your page.</p>
</section>

<section>
<h2>About Us</h2>
<p>Another section that could contain more
details about your website or company.</p>
</section>
</main>

<footer>
<p>&copy; 2024 My Website. All rights
reserved.</p>
</footer>
```

```
</body>  
</html>
```

17. HTML Symbols

- **Tag**

Không có thẻ cụ thể cho các ký hiệu trong HTML. Thay vào đó, các ký hiệu được chèn bằng các thực thể ký tự hoặc Unicode.

- **Description**

Hiển thị các ký tự và biểu tượng đặc biệt không có sẵn trực tiếp trên bàn phím.

- **Features**

Character Entities: Đây là các mã đặc biệt đại diện cho các ký tự. Chúng bắt đầu bằng ký hiệu & (&), sau là tên mã hoặc số và kết thúc bằng dấu chấm phẩy (;);

Unicode: Các ký tự Unicode có thể được biểu diễn bằng HTML bằng mã số.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1>Greek Letter Example</h1> <h2>The capital alpha letter by character entities : &Alpha;</h2> <h2>The capital alpha letter by number: &#913;</h2> </body> </html></pre>	<p>Greek Letter Example</p> <p>The capital alpha letter by character entities : A</p> <p>The capital alpha letter by number: A</p>

18. HTML Emojis

- **Tag**

Không có thẻ cụ thể cho biểu tượng cảm xúc trong HTML.

- **Description**

Để thêm các yếu tố trực quan thể hiện cảm xúc, thể hiện đồ vật hoặc nâng cao trải nghiệm người dùng tổng thể trên trang web.

- **Features**

Wide Range of Emojis: Có hàng nghìn biểu tượng cảm xúc, bao gồm các danh mục như cảm xúc, đồ vật, biểu tượng, động vật, thức ăn, v.v.

Accessibility: Biểu tượng cảm xúc có thể cải thiện khả năng đọc và tính biểu cảm của văn bản, khiến văn bản trở nên hấp dẫn hơn.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1>HTML Emoji Example</h1> <h2>😊</h2> <h2>🚀</h2> </body> </html></pre>	<p>HTML Emoji Example</p> <p>😊</p> <p>🚀</p>

19. HTML URL Encode

- **Tag**

Không có thẻ HTML cụ thể cho URL Encode. Mã hóa URL được áp dụng cho chuỗi URL, thường là trong các thuộc tính như **href**, **action** hoặc **src**.

- **Description**

Để mã hóa các ký tự đặc biệt trong URL để chúng có thể được truyền qua internet một cách an toàn. Điều này bao gồm khoảng trắng, ký tự dành riêng và ký tự không phải ASCII.

- **Features**

Safe Transmission: Đảm bảo rằng các URL được truyền và giải thích chính xác bởi các máy chủ web và trình duyệt;

Special Characters: Mã hóa các ký tự có ý nghĩa đặc biệt trong URL, chẳng hạn như dấu cách, ký hiệu, dấu chấm hỏi và các ký tự khác.

Non-ASCII Characters: Mã hóa các ký tự nằm ngoài phạm vi ASCII bằng mã hóa UTF-8.

Uniformity: Cung cấp cách mã hóa URL tiêu chuẩn để đảm bảo khả năng tương thích trên các hệ thống và nền tảng khác nhau.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>HTML URL Encoding Example</title> </head> <body> <h1>URL Encoding Examples</h1> <p>Original URL: Search for HTML URL encoding</p></pre>	<p>URL Encoding Examples</p> <p>Original URL: Search for HTML URL encoding</p> <p>Encoded URL: Search for HTML URL encoding</p> <p>Encoded URL with Special Characters: Search with special characters</p>

```
<p> Encoded URL: <a  
href="https://example.com/search?query=HTML  
%20URL%20encoding">Search for HTML URL  
encoding</a></p>
```

```
<p> Encoded URL with Special Characters: <a  
href="https://example.com/search?query=HTML  
%20URL%20encoding%20&%20more=details">  
Search with special characters</a></p>
```

```
</body>  
</html>
```

20. HTML Forms

- **Tag**

<form>

- **Description**

Một HTML form được sử dụng để thu thập thông tin đầu vào của người dùng.

- **Features**

Input Elements: Có thể bao gồm nhiều loại thành phần đầu vào khác nhau, chẳng hạn như trường văn bản, trường mật khẩu, hộp kiểm, nút radio và danh sách thả xuống.

Form Controls: Bao gồm các nút gửi, nút đặt lại và các điều khiển khác tương tác với đầu vào của người dùng.

- **Code and Result**

Code	Result
------	--------

```

<!DOCTYPE html>
<html>
<body>

<h2>Text input fields</h2>

<form>
<label for="fname">First
name:</label><br>
<input type="text" id="fname"
name="fname" value="John"><br>
</form>

<p>Choose your favorite Web
language:</p>

<form>
<input type="radio" id="html"
name="fav_language" value="HTML">
<label for="html">HTML</label><br>
<input type="radio" id="css"
name="fav_language" value="CSS">
<label for="css">CSS</label><br>

</form>

</body>
</html>

```

Text input fields

First name:

Choose your favorite Web language:

- HTML
- CSS

21. HTML Graphics

HTML Graphics đề cập đến việc sử dụng các kỹ thuật và công nghệ HTML5 để tạo ra và xử lý đồ họa và hình ảnh trên trang web. Các công cụ chính cho việc này là thẻ `<canvas>` và `<svg>` (Scalable Vector Graphics).

❖ HTML Canvas

• Tag

`<canvas>`

Cú pháp cơ bản: `<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">`

Các thuộc tính:

- + id: Đặt một id để có thể truy cập phần tử này bằng JavaScript.
- + width và height: Xác định kích thước của canvas.
- + style: Có thể sử dụng thuộc tính CSS để thêm viền hoặc các kiểu khác cho canvas.

• Description

HTML Canvas cung cấp một loạt các tính năng để tạo và thao tác với đồ họa trên trang web. Từ việc vẽ các hình dạng cơ bản, văn bản, và hình ảnh đến việc sử dụng gradient, pattern, clipping và các hiệu ứng chuyển đổi, Canvas là một công cụ mạnh mẽ cho việc tạo các ứng dụng đồ họa phong phú và tương tác trực tiếp trên trình duyệt.

• Features

Vẽ Hình Dạng Cơ Bản: Hình chữ nhật, hình tròn, đường thẳng, **hình đa giác**.

Vẽ văn bản

Vẽ hình ảnh

Gradient: Gradient tuyến tính, gradient radial.

• Code and Result

Vẽ Hình Dạng Cơ Bản: Hình chữ nhật

Code	Result

```

<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="170"
height="120" style="border:1px solid
#d3d3d3;"></canvas>
<script>
var c = document.getElementById("myCanvas");
//Lấy phần tử canvas bằng id và lưu vào biến c.
var ctx = c.getContext("2d");//Lấy ngữ cảnh 2D
của canvas và lưu vào biến ctx.
ctx.fillStyle = 'blue'; // Màu nền
ctx.strokeStyle = 'red'; // Màu viền
ctx.fillRect(10, 10, 150, 100); // Vẽ hình chữ nhật
đầy màu
ctx.strokeRect(10, 10, 150, 100); // Vẽ viền hình
chữ nhật
</script>
</body>
</html>

```



Vẽ Hình Dạng Cơ Bản: Hình tròn

Code	Result
------	--------

```

<!DOCTYPE html>

<html>
<body>
<canvas id="myCanvas" width="120"
height="120" style="border:1px solid
#d3d3d3;"></canvas>

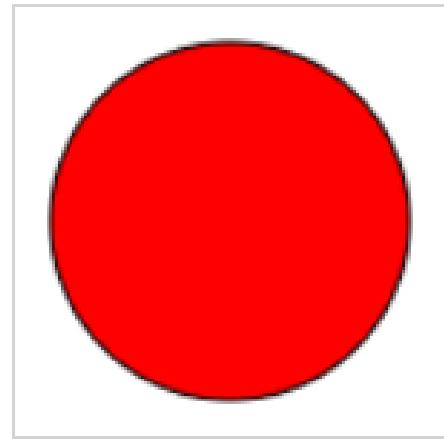
<script>
var c = document.getElementById("myCanvas");
//Lấy phần tử canvas bằng id và lưu vào biến c.
var ctx = c.getContext("2d"); //Lấy ngữ cảnh 2D của
canvas và lưu vào biến ctx.

ctx.beginPath(); // Bắt đầu một đường vẽ mới
ctx.arc(60, 60, 50, 0, Math.PI * 2); // x, y, bán kính,
góc bắt đầu, góc kết thúc

ctx.fillStyle = 'red';

ctx.fill(); // Tô màu hình tròn
ctx.stroke(); // Vẽ viền hình tròn
</script>
</body>
</html>

```



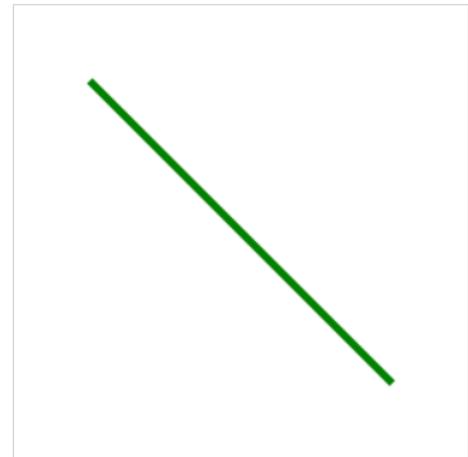
Vẽ Hình Dạng Cơ Bản: Đường thẳng

Code	Result
------	--------

```

<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="300"
height="300" style="border:1px solid
#d3d3d3;"></canvas>
<script>
var c = document.getElementById("myCanvas");
//Lấy phần tử canvas bằng id và lưu vào biến c.
var ctx = c.getContext("2d"); //Lấy ngữ cảnh 2D của
canvas và lưu vào biến ctx.
ctx.beginPath(); // Bắt đầu một đường vẽ mới
ctx.moveTo(50, 50); // Di đến điểm bắt đầu
ctx.lineTo(250, 250); // Vẽ đến điểm kết thúc
ctx.strokeStyle = 'green'; // Màu đường
ctx.lineWidth = 5; // Độ dày đường
ctx.stroke(); // Vẽ đường
</script>
</body>
</html>

```



Vẽ Hình Dạng Cơ Bản: Hình Đa Giác

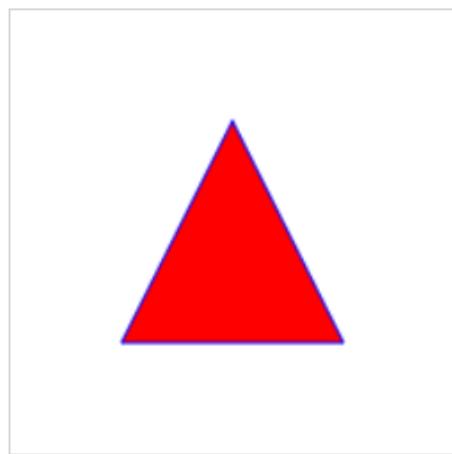
Code	Result
------	--------

```

<!DOCTYPE html>

<html>
<body>
<canvas id="myCanvas" width="200"
height="200" style="border:1px solid
#d3d3d3;"></canvas>
<script>
var c = document.getElementById("myCanvas");
//Lấy phần tử canvas bằng id và lưu vào biến c.
var ctx = c.getContext("2d"); //Lấy ngữ cảnh 2D của
canvas và lưu vào biến ctx.
ctx.beginPath(); // Bắt đầu một đường vẽ mới
ctx.moveTo(50, 150); // Di đến điểm bắt đầu
ctx.lineTo(150, 150); // Di đến điểm tiếp theo
ctx.lineTo(100, 50); // Vẽ đến điểm kết thúc
ctx.closePath(); // Đóng đường dẫn
ctx.fillStyle = 'red'; // Màu nền
ctx.strokeStyle = 'blue'; // Màu viền
ctx.fill(); // Tô màu
ctx.stroke(); // Vẽ viền
</script>
</body>
</html>

```



Vẽ Văn Bản

Code	Result
------	--------

```

<!DOCTYPE html>

<html>
<body>
<canvas id="myCanvas" width="400"
height="200" style="border:1px solid
#d3d3d3;"></canvas>
<script>
var c = document.getElementById("myCanvas");
//Lấy phần tử canvas bằng id và lưu vào biến c.
var ctx = c.getContext("2d"); //Lấy ngữ cảnh 2D của
canvas và lưu vào biến ctx.

ctx.font = '30px Arial'; // Kích thước và kiểu chữ
ctx.fillStyle = 'black'; // Màu chữ
ctx.strokeStyle = 'blue'; // Màu viền
ctx.fillText('Hello Canvas!', 100, 70); // Vẽ văn bản
ctx.strokeText('Hello Canvas!', 100, 130); // Vẽ viền
văn bản
</script>
</body>
</html>

```

Hello Canvas!

Hello Canvas!

Vẽ Hình Ảnh

Code	Result
------	--------

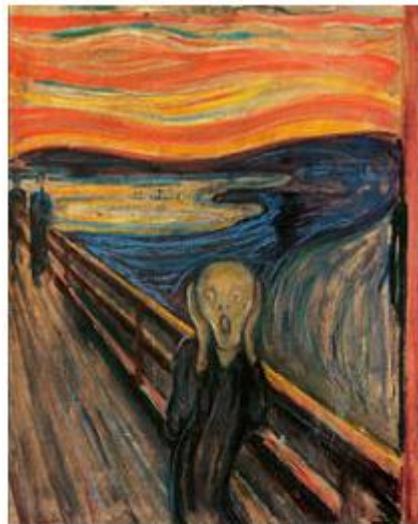
```

<!DOCTYPE html>
<html>
<body>
<p>Image to use:</p>

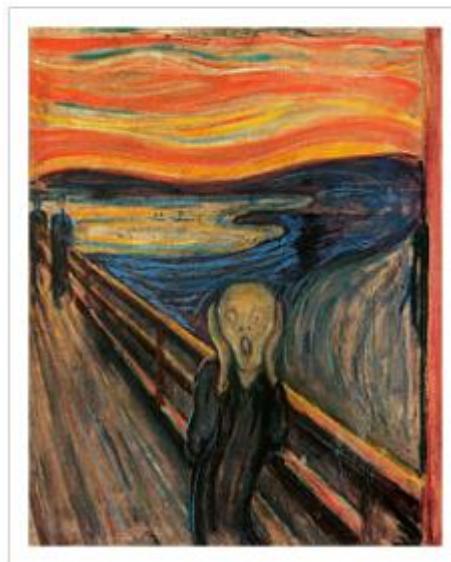

</>
<p>Canvas to fill:</p>
<canvas id="myCanvas" width="240"
height="297" style="border:1px solid #d3d3d3;">
<!-- Thẻ canvas tạo một khu vực vẽ đồ họa trên
trang --&gt;
&lt;/canvas&gt;
&lt;p&gt;&lt;button onclick="myCanvas()"&gt;Try
it&lt;/button&gt;&lt;/p&gt;
<!-- `onclick` gọi hàm `myCanvas()` khi người
dùng nhấp vào nút --&gt;
&lt;script&gt;
function myCanvas() {
  var c = document.getElementById("myCanvas");
  // Lấy đối tượng canvas từ DOM bằng id
  "myCanvas"
  var ctx = c.getContext("2d");
  // Lấy đối tượng context 2D từ canvas. Context
  này cho phép vẽ đồ họa 2D lên canvas
  var img = document.getElementById("scream");
  // Lấy đối tượng hình ảnh từ DOM bằng id
  "scream"
  ctx.drawImage(img, 10, 10);
}
</pre>

```

Image to use:



Canvas to fill:



[Try it](#)

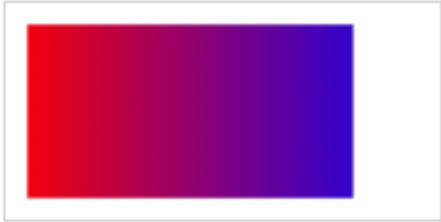
```

// Vẽ hình ảnh lên canvas tại vị trí (10, 10) tính từ
góc trên bên trái của canvas
}

</script>
</body>
</html>

```

Gradient Tuyến Tính

Code	Result
<pre> <!DOCTYPE html> <html> <body> <canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;"></canvas> <script> var c = document.getElementById("myCanvas"); //Lấy phần tử canvas bằng id và lưu vào biến c. var ctx = c.getContext("2d"); //Lấy ngữ cảnh 2D của canvas và lưu vào biến ctx. var grd = ctx.createLinearGradient(0,0,200,0); // Tạo một gradient tuyến tính bắt đầu từ tọa độ (0, 0) đến (200, 0). grd.addColorStop(0, 'red'); // Điểm bắt đầu của gradient grd.addColorStop(1, 'blue'); // Điểm kết thúc của gradient ctx.fillStyle = grd; </pre>	

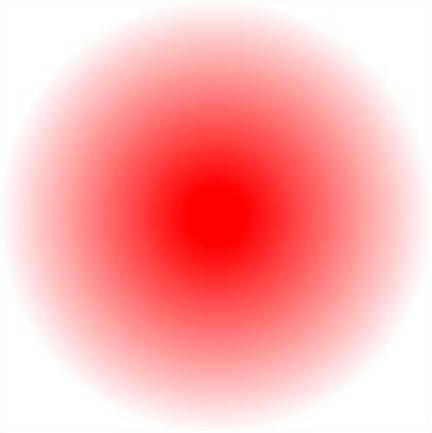
```

ctx.fillRect(10,10,150,80); //Vẽ một hình chữ nhật
bắt đầu từ tọa độ (10, 10) với chiều rộng 150 và
chiều cao 80, và tô màu bằng gradient.

</script>
</body>
</html>

```

Gradient Radial

Code	Result
<pre> <!DOCTYPE html> <html> <body> <canvas id="myCanvas" width="300" height="300" style="border:1px solid #d3d3d3;"></canvas> <script> var c = document.getElementById("myCanvas"); //Lấy phần tử canvas bằng id và lưu vào biến c. var ctx = c.getContext("2d"); //Lấy ngữ cảnh 2D của canvas và lưu vào biến ctx. var grd = ctx.createRadialGradient(150, 150, 20, 150, 150, 150); // Tạo một gradient hình tròn với tâm đầu tiên tại tọa độ (150, 150) và bán kính 20, tâm thứ hai tại tọa độ (150, 150) và bán kính 150. grd.addColorStop(0, 'red'); // Màu ở trung tâm grd.addColorStop(1, 'white'); // Màu ở ngoài cùng ctx.fillStyle = grd; // Đặt gradient vừa tạo làm màu tô cho các hình vẽ. ctx.fillRect(0, 0, 300, 300); // Vẽ và tô màu với gradient radial </pre>	

```
</script>  
</body>  
</html>
```

❖ HTML SVG

• Tag

```
<svg>
```

Cú pháp cơ bản:

```
<svg width="200" height="200" viewBox="0 0 100 100">  
    <!-- Các phần tử SVG khác sẽ nằm trong thẻ này -->  
</svg>
```

Các thuộc tính:

- + width: Chiều rộng của khung SVG trong đơn vị pixel.
- + height: Chiều cao của khung SVG trong đơn vị pixel.
- + viewBox: Xác định hệ tọa độ nội bộ của SVG. Thuộc tính này có bốn giá trị: min-x, min-y, width, và height.

• Description

SVG (Scalable Vector Graphics) là một tiêu chuẩn của W3C cho đồ họa vector được sử dụng trên web. SVG cho phép bạn tạo ra đồ họa và hình ảnh có thể thay đổi kích thước mà không bị mất chất lượng, điều này rất quan trọng cho các thiết bị có độ phân giải cao và các thiết kế đáp ứng (responsive design).

• Features

Định dạng vector: SVG là định dạng đồ họa vector, có nghĩa là hình ảnh không bị mất chất lượng khi phóng to hoặc thu nhỏ.

Tương tác với JavaScript: Bạn có thể sử dụng JavaScript để tương tác và thay đổi các phần tử SVG.
Hỗ trợ các hình dạng cơ bản: SVG hỗ trợ nhiều hình dạng cơ bản như đường thẳng, hình chữ nhật, hình tròn, hình elip, đa giác, và đường cong.

• Code and Result

Vẽ Hình Dạng Cơ Bản: Hình tròn

Code	Result
------	--------

```

<!DOCTYPE html>
<html>
<body>


Không sử dụng viewBox

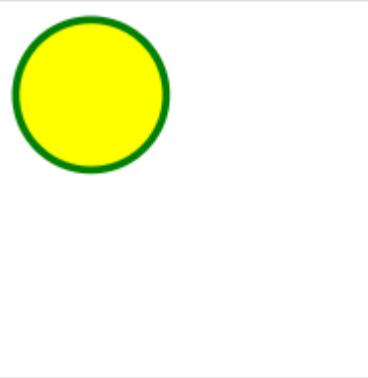

<svg width="200" height="200" style="border:1px solid #d3d3d3;">
<circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
<!--
<circle>: Vẽ một hình tròn.
- cx="50" và cy="50": Tọa độ tâm của hình tròn.
- r="40": Bán kính của hình tròn.
- stroke="green": Màu viền của hình tròn.
- stroke-width="4": Độ dày của viền.
- fill="yellow": Màu nền của hình tròn.
-->
</svg>


Sử dụng viewBox

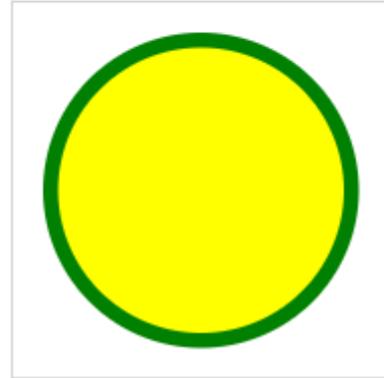

<!--viewBox="0 0 100 100": Xác định hệ tọa độ nội bộ của SVG. Điều này có nghĩa là tất cả các phần tử bên trong SVG sẽ được vẽ trong một không gian tọa độ từ (0,0) đến (100,100). Hệ tọa độ này sẽ được co giãn để phù hợp với kích thước hiển thị 200x200 pixel mà không làm thay đổi chất lượng hình ảnh.-->
<svg width="200" height="200" viewBox="0 0 100 100" style="border:1px solid #d3d3d3;">
<circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>

```

Không sử dụng viewBox



Sử dụng viewBox



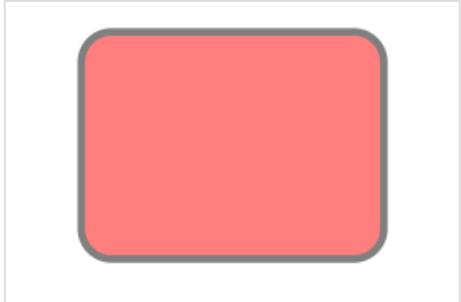
```
</body>  
</html>
```

Vẽ Hình Dạng Cơ Bản: Hình chữ nhật

Code	Result
<pre><!DOCTYPE html> <html> <body> <svg width="400" height="120"> <rect x="10" y="10" width="200" height="100" stroke="red" stroke-width="6" fill="blue" /> <!-- <rect>: Vẽ một hình chữ nhật.</pre>	

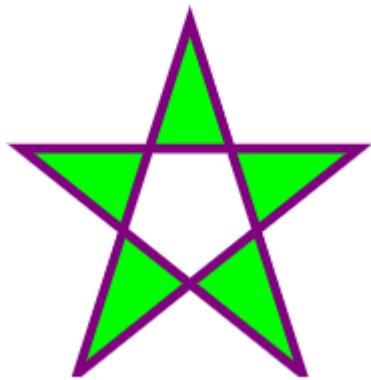
- `x="10" và y="10"`: Tọa độ của góc trên bên trái của hình chữ nhật (10, 10).
 - `width="200" và height="100"`: Chiều rộng 200 và chiều cao 100 của hình chữ nhật.
 - `stroke="red"`: Màu viền của hình chữ nhật.
 - `stroke-width="6"`: Độ dày của viền.
 - `fill="blue"`: Màu nền của hình chữ nhật.
`-->`
`</svg>`
`</body>`
`</html>`

Vẽ Hình Dạng Cơ Bản: Hình chữ nhật với Opacity và Rounded Corners

Code	Result
<pre> <!DOCTYPE html> <html> <body> <svg width="300" height="200" style="border:1px solid #d3d3d3;"> <rect x="50" y="20" rx="20" ry="20" width="200" height="150" style="fill:red;stroke:black;stroke-width:5;opacity:0.5" /> <!--<rect>: Vẽ một hình chữ nhật. - x="50" và y="20": Tọa độ của góc trên bên trái của hình chữ nhật (50, 20). - rx="20" và ry="20": Bán kính bo tròn của các góc hình chữ nhật. </pre>	

- width="200" và height="150": Chiều rộng 150 và chiều cao 150 của hình chữ nhật.
 - style="fill:red;stroke:black;stroke-width:5;opacity:0.5": Các thuộc tính kiểu dáng:
 + fill="red": Màu nền của hình chữ nhật là màu đỏ.
 + stroke="black": Màu viền của hình chữ nhật là màu đen.
 + stroke-width="5": Độ dày của viền là 5.
 + opacity="0.5": Độ mờ của hình chữ nhật là 0.5 (50% trong suốt).
 -->
 </svg>
 </body>
 </html>

Vẽ Hình Dạng Cơ Bản: Ngôi sao

Code	Result
<pre> <!DOCTYPE html> <html> <body> <svg width="300" height="200"> <polygon points="100,10 40,198 190,78 10,78 160,198" style="fill:lime;stroke:purple;stroke-width:5;fill- rule:evenodd;" /> <!-- <polygon>: Vẽ một đa giác. - points="100,10 40,198 190,78 10,78 160,198": Danh sách các tọa độ của các điểm đỉnh của đa giác. - style="fill:lime;stroke:purple;stroke-width:5;fill- rule:evenodd;": Các thuộc tính kiểu dáng: </pre>	

+ fill="lime": Màu nền của đa giác là màu xanh chanh.
 + stroke="purple": Màu viền của đa giác là màu tím.
 + stroke-width="5": Độ dày của viền là 5.
 + fill-rule="evenodd": Quy tắc điền màu cho đa giác, trong trường hợp này là “evenodd” (chẵn lẻ).
 -->
 </svg>
 </body>
 </html>

Vẽ Hình Dạng Cơ Bản: Gradient Ellipse và văn bản

Code	Result
<pre> <!DOCTYPE html> <html> <body> <svg height="130" width="500"> <defs> <linearGradient id="grad1"> <stop offset="0%" stop-color="yellow" /> <stop offset="100%" stop-color="red" /> </linearGradient> </defs> <ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" /> <text fill="#ffffff" font-size="45" font- family="Verdana" x="50" y="86">SVG</text> <!-- <defs>: Thể định nghĩa các phần tử đồ họa mà có thể được sử dụng lại trong SVG. </pre>	

```

- <linearGradient id="grad1">: Định nghĩa một
gradient tuyến tính với id là “grad1”.
+ <stop offset="0%" stop-color="yellow" />: Đặt
màu bắt đầu của gradient là màu vàng tại điểm 0%.
+ <stop offset="100%" stop-color="red" />: Đặt
màu kết thúc của gradient là màu đỏ tại điểm 100%.
<ellipse>: Vẽ một hình ellipse.
- cx="100" và cy="70": Tọa độ tâm của ellipse (100,
70).
- rx="85" và ry="55": Bán kính ngang 85 và bán
kính dọc 55 của ellipse.
- fill="url(#grad1)": Tô màu nền của ellipse bằng
gradient “grad1”.
<text>: Thêm văn bản vào SVG.
- fill="#ffffff": Màu chữ là màu trắng.
- font-size="45": Kích thước chữ là 45.
- font-family="Verdana": Font chữ là Verdana.
- x="50" và y="86": Tọa độ của điểm bắt đầu văn
bản (50, 86).

-->
</svg>
</body>
</html>

```

So sánh SVG và Canvas

Đặc điểm	SVG	Canvas
Định dạng	Vector (dựa trên XML)	Raster (dựa trên điểm ảnh)
Khả năng mở rộng	Tốt, không mất chất lượng	Kém, dễ mất chất lượng
Hiệu suất	Tốt với ít đối tượng hoặc bề mặt lớn	Tốt với nhiều đối tượng hoặc bề mặt nhỏ

Tương tác	CSS và JavaScript	JavaScript
DOM	Là một phần của cây DOM	Không phải là một phần của cây DOM
Ứng dụng	Biểu đồ, đồ thị, logo, hình ảnh chi tiết	Trò chơi, hoạt hình, đồ họa động phức tạp

22. HTML Video

- **Tag**

`<video>`

Cú pháp cơ bản:

```
<video width="320" height="240" controls>
    <source src="movie.mp4" type="video/mp4">
</video>
```

Các thuộc tính:

- src: Đường dẫn đến file video. Thông thường thuộc tính này được đặt trong thẻ `<source>`.
- controls: Hiển thị các điều khiển phát lại (play, pause, volume, etc.) cho video.
- autoplay: Video sẽ tự động phát khi tải trang. Lưu ý rằng thuộc tính này thường bị vô hiệu hóa trên nhiều trình duyệt để tránh việc phát âm thanh không mong muốn.
- loop: Video sẽ tự động lặp lại khi kết thúc.
- muted: Video sẽ được phát mà không có âm thanh. Đây là thuộc tính hữu ích khi kết hợp với autoplay.
- poster: Đường dẫn đến hình ảnh được hiển thị trước khi video phát hoặc khi video chưa tải.
- preload: Định nghĩa cách trình duyệt tải video khi trang web được tải. Các giá trị có thể là:
 - + auto: Trình duyệt sẽ tải video khi trang tải.
 - + metadata: Chỉ tải metadata (như chiều dài video, kích thước, etc.).
 - + none: Không tải video trước khi người dùng bấm play.
- width và height: Định rõ chiều rộng và chiều cao của khung video.

- **Description**

Thẻ `<video>` được sử dụng để tạo một “trình xem phim” trên trang web.

- **Features**

Thẻ <video> được sử dụng để nhúng video trên trang web. Thẻ này phải bao gồm một hoặc nhiều thẻ <source> để chỉ định các nguồn video với các định dạng khác nhau nhằm đảm bảo tính tương thích trên tất cả các trình duyệt.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html> <body> <video width="320" height="240" controls autoplay loop> <source src="movie.mp4" type="video/mp4"> </video> </body> </html></pre>	

23. HTML Id

- **Tag**

- **Description**

Thuộc tính id được sử dụng để chỉ định một định danh duy nhất cho một phần tử HTML.

- **Features**

Định danh duy nhất: Giá trị của thuộc tính id phải là duy nhất trong toàn bộ tài liệu HTML. Điều này có nghĩa là không thể có hai phần tử cùng một id trong cùng một trang.

Liên kết với CSS: Thuộc tính id thường được sử dụng để liên kết với các khai báo kiểu trong CSS. Bạn có thể sử dụng id để áp dụng các kiểu cụ thể cho một phần tử duy nhất.

Tương tác với JavaScript: JavaScript có thể truy cập và thao tác với phần tử có id cụ thể bằng phương thức getElementById().

Tạo điểm đánh dấu (Bookmarks): Thuộc tính id có thể được sử dụng để tạo các điểm đánh dấu trong tài liệu, cho phép người dùng nhảy đến các phần cụ thể của trang.

- **Code and Result**

Liên kết với CSS: Thuộc tính id có thể được sử dụng trong CSS để tạo kiểu cho một phần tử cụ thể. Để chọn phần tử bằng id, bạn sử dụng ký tự # trước giá trị id.

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> #myHeader { background-color: lightblue; color: black; padding: 40px; text-align: center; } </style> </head> <body> <h2>The id Attribute</h2> <p>Use CSS to style an element with the id "myHeader":</p> <h1 id="myHeader">My Header</h1> </body> </html></pre>	<p>The id Attribute</p> <p>Use CSS to style an element with the id "myHeader":</p> 

Tương tác với JavaScript

Code	Result
<pre><!DOCTYPE html> <html> <body> <h1 id="myHeader">Đây là tiêu đề</h1></pre>	<p>Đây là tiêu đề</p> <p><input type="button" value="Thay đổi văn bản"/></p> 

```

<button onclick="changeText()">Thay đổi văn bản</button>
<script>
function changeText() {
document.getElementById("myHeader").innerHTML
= "Tiêu đề đã thay đổi!";
//getElementById("myHeader"): Truy cập phần tử h1
có id là "myHeader" và thay đổi nội dung của nó khi
nhấn nút.
}
</script>
</body>
</html>

```

Tiêu đề đã thay đổi!

Thay đổi văn bản

Tạo điểm đánh dấu (Bookmarks): Thuộc tính id có thể được sử dụng để tạo liên kết nội bộ trong cùng một trang. Khi sử dụng thẻ [<a>](#) với thuộc tính href có giá trị là # theo sau là giá trị id, trình duyệt sẽ cuộn đến phần tử có id đó.

Code	Result
<pre> <!DOCTYPE html> <html> <head> <title>Dấu trang HTML</title> </head> <body> <h1>Trang Web của Tôi</h1> <p>Phần 1</p> <p>Phần 2</p> <p>Phần 3</p> <p>Phần 4</p> <p>Phần 5</p> </pre>	

```

<p><a href="#Sect6">Phần 6</a></p>
<p><a href="#Sect7">Phần 7</a></p>
<p><a href="#Sect8">Phần 8</a></p>
<p><a href="#Sect9">Phần 9</a></p>
<p><a href="#Sect10">Phần 10</a></p>
<h1 id="Sect1">Phần 1</h1>
<p>Nội dung của phần 1.</p>
<h1 id="Sect2">Phần 2</h1>
<p>Nội dung của phần 2.</p>
<h1 id="Sect3">Phần 3</h1>
<p>Nội dung của phần 3.</p>
<h1 id="Sect4">Phần 4</h1>
<p>Nội dung của phần 4.</p>
<h1 id="Sect5">Phần 5</h1>
<p>Nội dung của phần 5.</p>
<h1 id="Sect6">Phần 6</h1>
<p>Nội dung của phần 6.</p>
<h1 id="Sect7">Phần 7</h1>
<p>Nội dung của phần 7.</p>
<h1 id="Sect8">Phần 8</h1>
<p>Nội dung của phần 8.</p>
<h1 id="Sect9">Phần 9</h1>
<p>Nội dung của phần 9.</p>
<h1 id="Sect10">Phần 10</h1>
<p>Nội dung của phần 10.</p>
</body>
</html>

```

Trang Web của Tôi

[Phần 1](#)

[Phần 2](#)

[Phần 3](#)

[Phần 4](#)

[Phần 5](#)

[Phần 6](#)

[Phần 7](#)

[Phần 8](#)

[Phần 9](#)

[Phần 10](#)

Phần 1

Nội dung của phần 1.

Phần 2

Nội dung của phần 2.

Phần 3

Nội dung của phần 3.

24. HTML Plug-ins

Khái niệm "plugin" thường liên quan đến việc nhúng nội dung từ bên ngoài, như video, âm thanh, tài liệu, hoặc các loại nội dung khác thông qua việc sử dụng các thẻ và thuộc tính cho phép tích hợp các

ứng dụng hoặc tiện ích bên ngoài vào trang web. Các "plugin" phổ biến trong HTML: Thẻ <object>, Thẻ <embed>,...

❖ Thẻ <object>

• Tag

<object>

Cú pháp cơ bản: <object width="100%" height="500px" data="snippet.html"></object>

Các thuộc tính:

- + **data**: URL của tệp nhúng (như video, âm thanh, PDF, etc.).
- + **width** và **height**: Chiều rộng và chiều cao của đối tượng.

• Features

Thẻ <object> được sử dụng để nhúng một tài liệu, ứng dụng hoặc tài nguyên bên ngoài vào trang web.

• Code and Result

Code	Result																																																			
<pre><!DOCTYPE html> <html> <body> <object width="100%" height="400px" data="snippet.html"></object> <object width="50%" height="50%" data="audi.jpeg"></object> </body> </html></pre>	<table><tbody><tr><td>Alfreds Futterkiste</td><td>Berlin</td><td>Germany</td></tr><tr><td>Berglunds snabbköp</td><td>Luleå</td><td>Sweden</td></tr><tr><td>Centro comercial Moctezuma</td><td>México D.F.</td><td>Mexico</td></tr><tr><td>Ernst Handel</td><td>Graz</td><td>Austria</td></tr><tr><td>FISSA Fabrica Inter. Salchichas S.A.</td><td>Madrid</td><td>Spain</td></tr><tr><td>Galeria del gastrónomo</td><td>Barcelona</td><td>Spain</td></tr><tr><td>Island Trading</td><td>Cowes</td><td>UK</td></tr><tr><td>Königlich Essen</td><td>Brandenburg</td><td>Germany</td></tr><tr><td>Laughing Bacchus Wine Cellars</td><td>Vancouver</td><td>Canada</td></tr><tr><td>Magazzini Alimentari Riuniti</td><td>Bergamo</td><td>Italy</td></tr><tr><td>North/South</td><td>London</td><td>UK</td></tr><tr><td>Paris spécialités</td><td>Paris</td><td>France</td></tr><tr><td>Rattlesnake Canyon Grocery</td><td>Albuquerque</td><td>USA</td></tr><tr><td>Simons bistro</td><td>København</td><td>Denmark</td></tr><tr><td>The Big Cheese</td><td>Portland</td><td>USA</td></tr><tr><td>Vaffeljernet</td><td>Århus</td><td>Denmark</td></tr><tr><td>Wolski Zajazd</td><td>Warszawa</td><td>Poland</td></tr></tbody></table> 	Alfreds Futterkiste	Berlin	Germany	Berglunds snabbköp	Luleå	Sweden	Centro comercial Moctezuma	México D.F.	Mexico	Ernst Handel	Graz	Austria	FISSA Fabrica Inter. Salchichas S.A.	Madrid	Spain	Galeria del gastrónomo	Barcelona	Spain	Island Trading	Cowes	UK	Königlich Essen	Brandenburg	Germany	Laughing Bacchus Wine Cellars	Vancouver	Canada	Magazzini Alimentari Riuniti	Bergamo	Italy	North/South	London	UK	Paris spécialités	Paris	France	Rattlesnake Canyon Grocery	Albuquerque	USA	Simons bistro	København	Denmark	The Big Cheese	Portland	USA	Vaffeljernet	Århus	Denmark	Wolski Zajazd	Warszawa	Poland
Alfreds Futterkiste	Berlin	Germany																																																		
Berglunds snabbköp	Luleå	Sweden																																																		
Centro comercial Moctezuma	México D.F.	Mexico																																																		
Ernst Handel	Graz	Austria																																																		
FISSA Fabrica Inter. Salchichas S.A.	Madrid	Spain																																																		
Galeria del gastrónomo	Barcelona	Spain																																																		
Island Trading	Cowes	UK																																																		
Königlich Essen	Brandenburg	Germany																																																		
Laughing Bacchus Wine Cellars	Vancouver	Canada																																																		
Magazzini Alimentari Riuniti	Bergamo	Italy																																																		
North/South	London	UK																																																		
Paris spécialités	Paris	France																																																		
Rattlesnake Canyon Grocery	Albuquerque	USA																																																		
Simons bistro	København	Denmark																																																		
The Big Cheese	Portland	USA																																																		
Vaffeljernet	Århus	Denmark																																																		
Wolski Zajazd	Warszawa	Poland																																																		

❖ Thẻ <embed>

• Tag

<embed>

Cú pháp cơ bản: <embed width="100%" height="500px" src="snippet.html">

Các thuộc tính:

- + **src**: URL của nội dung hoặc ứng dụng được nhúng.
- + **width** và **height**: Chiều rộng và chiều cao của đối tượng.

• Features

Thẻ <embed> được sử dụng để nhúng nội dung hoặc ứng dụng từ các nguồn bên ngoài như video, âm thanh, và flash (lỗi thời) vào trang HTML. Không giống như <object>, <embed> không yêu cầu phần tử kết thúc và nội dung không được hiển thị nếu trình duyệt không hỗ trợ.

• Code and Result

Code	Result																																																			
<pre><!DOCTYPE html> <html> <body> <embed width="100%" height="400px" src="snippet.html"> <embed width="50%" height="50%" src="audi.jpeg"> </body> </html></pre>	<table><tbody><tr><td>Alfreds Futterkiste</td><td>Berlin</td><td>Germany</td></tr><tr><td>Berglunds snabbköp</td><td>Luleå</td><td>Sweden</td></tr><tr><td>Centro comercial Moctezuma</td><td>México D.F.</td><td>Mexico</td></tr><tr><td>Ernst Handel</td><td>Graz</td><td>Austria</td></tr><tr><td>FIESA Fábrica Inter. Salchichas S.A.</td><td>Madrid</td><td>Spain</td></tr><tr><td>Galería del gastrónomo</td><td>Barcelona</td><td>Spain</td></tr><tr><td>Island Trading</td><td>Cowes</td><td>UK</td></tr><tr><td>Königlich Essen</td><td>Brandenburg</td><td>Germany</td></tr><tr><td>Laughing Bacchus Wine Cellars</td><td>Vancouver</td><td>Canada</td></tr><tr><td>Magazzini Alimentari Riuniti</td><td>Bergamo</td><td>Italy</td></tr><tr><td>North/South</td><td>London</td><td>UK</td></tr><tr><td>Paris spécialités</td><td>Paris</td><td>France</td></tr><tr><td>Rattlesnake Canyon Grocery</td><td>Albuquerque</td><td>USA</td></tr><tr><td>Simons bistro</td><td>København</td><td>Denmark</td></tr><tr><td>The Big Cheese</td><td>Portland</td><td>USA</td></tr><tr><td>Vaffeljernet</td><td>Århus</td><td>Denmark</td></tr><tr><td>Wolski Zajazd</td><td>Warszawa</td><td>Poland</td></tr></tbody></table> 	Alfreds Futterkiste	Berlin	Germany	Berglunds snabbköp	Luleå	Sweden	Centro comercial Moctezuma	México D.F.	Mexico	Ernst Handel	Graz	Austria	FIESA Fábrica Inter. Salchichas S.A.	Madrid	Spain	Galería del gastrónomo	Barcelona	Spain	Island Trading	Cowes	UK	Königlich Essen	Brandenburg	Germany	Laughing Bacchus Wine Cellars	Vancouver	Canada	Magazzini Alimentari Riuniti	Bergamo	Italy	North/South	London	UK	Paris spécialités	Paris	France	Rattlesnake Canyon Grocery	Albuquerque	USA	Simons bistro	København	Denmark	The Big Cheese	Portland	USA	Vaffeljernet	Århus	Denmark	Wolski Zajazd	Warszawa	Poland
Alfreds Futterkiste	Berlin	Germany																																																		
Berglunds snabbköp	Luleå	Sweden																																																		
Centro comercial Moctezuma	México D.F.	Mexico																																																		
Ernst Handel	Graz	Austria																																																		
FIESA Fábrica Inter. Salchichas S.A.	Madrid	Spain																																																		
Galería del gastrónomo	Barcelona	Spain																																																		
Island Trading	Cowes	UK																																																		
Königlich Essen	Brandenburg	Germany																																																		
Laughing Bacchus Wine Cellars	Vancouver	Canada																																																		
Magazzini Alimentari Riuniti	Bergamo	Italy																																																		
North/South	London	UK																																																		
Paris spécialités	Paris	France																																																		
Rattlesnake Canyon Grocery	Albuquerque	USA																																																		
Simons bistro	København	Denmark																																																		
The Big Cheese	Portland	USA																																																		
Vaffeljernet	Århus	Denmark																																																		
Wolski Zajazd	Warszawa	Poland																																																		

25. HTML Youtube

• Tag

<iframe>

Cú pháp cơ bản: <iframe width="560" height="315" src="https://www.youtube.com/embed/VIDEO_ID" frameborder="0" allow="accelerometer;

```
autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture"
allowfullscreen></iframe>
```

Các thuộc tính:

- + **width** và **height**: Định rõ chiều rộng và chiều cao của khung video YouTube nhúng.
- + **src**: Đường dẫn URL đến video YouTube bạn muốn nhúng. Định dạng của URL là "https://www.youtube.com/embed/VIDEO_ID", trong đó VIDEO_ID là mã định danh duy nhất của video YouTube.
- + **frameborder**: Đặt viền của khung video (thường là 0 để không có viền).
- + **allow**: Xác định các tính năng được phép sử dụng trong khung, như autoplay, encrypted-media, v.v.
- + **allowfullscreen**: Cho phép video hiển thị ở chế độ toàn màn hình.

• Description

Thẻ <iframe> sẽ nhúng trình phát video YouTube vào trang web.

• Features

Tự động phát (Autoplay): Bạn có thể thêm autoplay=1 vào URL để video tự động phát khi tải trang. Tuy nhiên, tính năng này có thể bị hạn chế trên một số trình duyệt.

Tắt âm thanh (Muted Autoplay): Để video tự động phát mà không có âm thanh, bạn thêm autoplay=1&mute=1 vào URL.

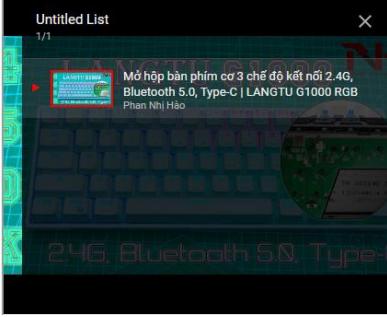
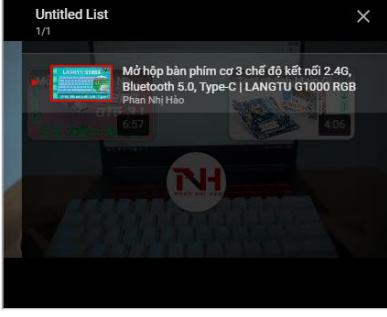
Danh sách phát (Playlist): Bạn có thể tạo danh sách phát bằng cách thêm playlist=VIDEO_ID vào URL.

Lặp lại video (Loop): Để video lặp lại liên tục, thêm loop=1 và playlist=VIDEO_ID vào URL.

Ẩn điều khiển (Hide Controls): Để ẩn các điều khiển phát lại, thêm controls=0 vào URL.

• Code and Result

Features	Code	Result
Tự động phát (Autoplay)	<!DOCTYPE html> <html> <body> <iframe width="420" height="345" src="https://www.youtube.com/embed/ip8FHCFhyd4?" autoplay=1>	

	<pre></iframe> </body> </html></pre>	
Tắt âm thanh (Muted Autoplay)	<pre><!DOCTYPE html> <html> <body> <iframe width="420" height="345" src="https://www.youtube.com/embed/ip8FHCPhyd4? autoplay=1&mute=1"> </iframe> </body> </html></pre>	
Danh sách phát (Playlist)	<pre><!DOCTYPE html> <html> <body> <iframe width="420" height="345" src="https://www.youtube.com/embed/xaYOXxiuK-0?playlist=xaYOXxiuK-0"> </iframe> </body> </html></pre>	
Lặp lại video (Loop)	<pre><!DOCTYPE html> <html> <body> <iframe width="420" height="345" src="https://www.youtube.com/embed/xaYOXxiuK-0?playlist=xaYOXxiuK-0&loop=1"> </iframe> </body> </html></pre>	

Ẩn điều khiển (Hide Controls)	<pre><!DOCTYPE html> <html> <body> <iframe width="420" height="345" src="https://www.youtube.com/embed/xaYOXxiuK-0?controls=0 "> </iframe> </body> </html></pre>	
-------------------------------	--	---

26. HTML Geolocation

- **Tag**
- **Description**

Geolocation API trong HTML cho phép các trang web và ứng dụng web truy cập vào vị trí địa lý của người dùng. Tính năng này được thực hiện thông qua JavaScript và rất hữu ích cho các ứng dụng yêu cầu biết vị trí người dùng, như bản đồ, dịch vụ tìm kiếm địa phương, hoặc dịch vụ cung cấp thông tin theo vị trí.

- **Features**

Lấy vị trí hiện tại: Sử dụng phương thức `getCurrentPosition()` để lấy vị trí hiện tại của người dùng. Phương thức này trả về một đối tượng vị trí chứa tọa độ (latitude và longitude) của người dùng.

Xử lý lỗi: Geolocation API cung cấp các mã lỗi để xử lý các tình huống như người dùng từ chối yêu cầu vị trí, thông tin vị trí không khả dụng, yêu cầu bị hết thời gian, và các lỗi không xác định.

Theo dõi vị trí: Sử dụng phương thức `watchPosition()` để theo dõi vị trí của người dùng và cập nhật liên tục khi vị trí thay đổi.

Hủy theo dõi vị trí: Sử dụng phương thức `clearWatch()` để hủy theo dõi vị trí.

- **Code and Result**

Lấy vị trí hiện tại của người dùng

Code	Result
------	--------

```

<!DOCTYPE html>
<html>
<body>
<h1>HTML Geolocation</h1>
<p>Click the button to get your coordinates.</p>
<button onclick="getLocation()">Try It</button>
<p id="demo"></p>
<script>
const x = document.getElementById("demo");
navigator.geolocation.getCurrentPosition(showPosition);
function getLocation() {
  if (navigator.geolocation) {

} else {
  x.innerHTML = "Geolocation is not supported by this
browser.";
}

}
function showPosition(position) {
  x.innerHTML = "Latitude: " + position.coords.latitude +
  "<br>Longitude: " + position.coords.longitude;
}
</script>
</body>
</html>

```

HTML Geolocation

Click the button to get your coordinates.

[Try It](#)

Latitude: 10.8027
Longitude: 106.6966

Xử lý lỗi

Code	Result
------	--------

```

<!DOCTYPE html>
<html>
<body>
<h1>HTML Geolocation</h1>
<p>Click the button to get your coordinates.</p>
<button onclick="getLocation()">Try It</button>
<p id="demo"></p>
<script>
const x = document.getElementById("demo");

function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition,
showError);
    } else {
        x.innerHTML = "Geolocation is not supported by this
browser.";
    }
}

function showPosition(position) {
    x.innerHTML = "Latitude: " + position.coords.latitude +
    "<br>Longitude: " + position.coords.longitude;
}

function showError(error) {
    switch(error.code) {
        case error.PERMISSION_DENIED:
            x.innerHTML = "User denied the request for
Geolocation."
            break;
        case error.POSITION_UNAVAILABLE:
            x.innerHTML = "Location information is unavailable."
    }
}

```

HTML Geolocation

Click the button to get your coordinates.

[Try It](#)

User denied the request for Geolocation.

```

break;

case error.TIMEOUT:
    x.innerHTML = "The request to get user location timed
out."

break;

case error.UNKNOWN_ERROR:
    x.innerHTML = "An unknown error occurred."
break;

}

}

</script>
</body>
</html>

```

Theo dõi vị trí

Code	Result
<pre> <!DOCTYPE html> <html> <body> <h1>HTML Geolocation</h1> <p>Click the button to get your coordinates.</p> <button onclick="getLocation()">Try It</button> <p id="demo"></p> <script> const x = document.getElementById("demo"); function getLocation() { if (navigator.geolocation) { navigator.geolocation.watchPosition(showPosition); } else { </pre>	<p>HTML Geolocation</p> <p>Click the button to get your coordinates.</p> <p>Try It</p> <p>Latitude: 10.8027 Longitude: 106.6966</p>

```

x.innerHTML = "Geolocation is not supported by this
browser.";

}

}

function showPosition(position) {
    x.innerHTML="Latitude: " + position.coords.latitude +
    "<br>Longitude: " + position.coords.longitude;
}
</script>
</body>
</html>

```

27. HTML Div

- **Tag**

`<div>`

Cú pháp cơ bản:

`<div>`

`<!-- Nội dung ở đây -->`

`</div>`

- **Description**

Thẻ `<div>` trong HTML là một thẻ dùng để định nghĩa một "division" hoặc một phần khói trong tài liệu HTML. Thẻ `<div>` không có ngữ nghĩa cụ thể mà chỉ đơn giản là một phần tử khói (block-level element) được sử dụng để nhóm các phần tử HTML lại với nhau, nhằm mục đích tạo cấu trúc hoặc bố cục cho trang web.

- **Features**

Block-level Element: Thẻ `<div>` là một phần tử cấp khói (block-level), nghĩa là nó chiếm toàn bộ chiều rộng của phần tử cha và bắt đầu trên một dòng mới.

Khối chứa (Container): Thẻ `<div>` tạo ra một khói chứa cho các phần tử khác, giúp nhóm chúng lại với nhau.

Tùy chỉnh bằng CSS: Thủ `<div>` có thể dễ dàng được tùy chỉnh bằng CSS thông qua các thuộc tính class và id.

- **Code and Result**

Block-level Element

Code	Result
<pre><!DOCTYPE html> <html> <style> div { background-color: #FFF4A3; } </style> <body> <h1>HTML DIV Example</h1> <h2>IoT <div>K22</div> is the best</h2> <p>Phần có nền vàng là DIV element.</p> </body> </html></pre>	<p>HTML DIV Example</p> <p>IoT K22 is the best</p> <p>Phần có nền vàng là DIV element.</p>

Khối chứa (Container)

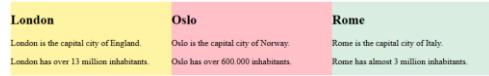
Code	Result
<pre><!DOCTYPE html> <html> <style> div { background-color: #FFF4A3; } </style> <body> <h1>HTML DIV Example</h1></pre>	<p>HTML DIV Example</p> <p>IoT K22 Is the best.</p>

```

<div>
  <h2>IoT</h2>
  <p>K22</p>
  <p>Is the best.</p>
</div>
</body>
</html>

```

Tùy chỉnh bằng CSS

Code	Result
<pre> <!DOCTYPE html> <html> <style> div.mycontainer { width:100%; overflow:auto; } div.mycontainer div { width:33%; float:left; } </style> <body> <div class="mycontainer"> <div style="background-color:#FFF4A3;"> <h2>London</h2> <p>London is the capital city of England.</p> <p>London has over 13 million inhabitants.</p> </div> </pre>	 <p>The screenshot displays three colored boxes side-by-side, each containing a city name and a brief description. The first box is yellow and contains 'London' and 'London is the capital city of England. London has over 13 million inhabitants.'. The second box is pink and contains 'Oslo' and 'Oslo is the capital city of Norway. Oslo has over 650,000 inhabitants.'. The third box is green and contains 'Rome' and 'Rome is the capital city of Italy. Rome has almost 3 million inhabitants.'</p>

```

<div style="background-color:#FFC0C7;">
    <h2>Oslo</h2>
    <p>Oslo is the capital city of Norway.</p>
    <p>Oslo has over 600.000 inhabitants.</p>
</div>
<div style="background-color:#D9EEE1;">
    <h2>Rome</h2>
    <p>Rome is the capital city of Italy.</p>
    <p>Rome has almost 3 million inhabitants.</p>
</div>
</div>
</body>
</html>

```

28. HTML Computer Code

- **Tag**

<code><kbd><samp><var>

- **Description**

HTML cung cấp một số thẻ để định dạng và hiển thị mã máy tính, đầu vào từ bàn phím, và các biến số.

- **Features**

Thẻ <code> được sử dụng để định dạng một đoạn mã ngắn hoặc một phần mã trong văn bản. Nó hiển thị nội dung bên trong nó với phông chữ đơn cách (monospace), giúp mã trở nên dễ đọc hơn. Thẻ <samp> (sample output) được sử dụng để hiển thị đầu ra từ một chương trình máy tính hoặc hệ thống. Thẻ này thường được sử dụng để biểu thị kết quả mà người dùng sẽ thấy sau khi thực hiện mã.

Thẻ <kbd> (keyboard input) được sử dụng để biểu thị đầu vào từ bàn phím. Nó thường được sử dụng khi hiển thị các phím hoặc tổ hợp phím mà người dùng nên nhấn.

Thẻ <var> (variable) được sử dụng để biểu thị tên biến trong mã. Thẻ này làm nổi bật tên biến, giúp phân biệt nó với phần còn lại của văn bản.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html> <head> <title>HTML Computer Code Elements</title> </head> <body> <h2>Ví dụ về các thẻ mã máy tính trong HTML</h2> <p>Đoạn mã:</p> <code> x = 5; y = 6; z = x + y; printf("Hello World"); </code> <p>Đầu ra mẫu:</p> <p>Thông báo từ máy tính: <samp>File not found. Press F1 to continue.</samp></p> <p>Đầu vào từ bàn phím:</p> <p>Nhấn <kbd>Ctrl</kbd> + <kbd>S</kbd> để lưu tài liệu.</p> <p>Biến số:</p> <p>Diện tích của tam giác là: 1/2 x <var>b</var> x <var>h</var>, trong đó <var>b</var> là đáy và <var>h</var> là chiều cao.</p> </body> </html></pre>	<p>Ví dụ về các thẻ mã máy tính trong HTML</p> <p>Đoạn mã:</p> <pre>x = 5; y = 6; z = x + y; printf("Hello World");</pre> <p>Đầu ra mẫu:</p> <p>Thông báo từ máy tính: <code>File not found. Press F1 to continue.</code></p> <p>Đầu vào từ bàn phím:</p> <p>Nhấn <code>Ctrl + S</code> để lưu tài liệu.</p> <p>Biến số:</p> <p>Diện tích của tam giác là: $1/2 \times b \times h$, trong đó b là đáy và h là chiều cao.</p>

29. HTML Semantics

- Tag

<article> <aside> <details> <figcaption> <figure> <footer> <header> <main> <mark> <nav> <section> <summary> <time>

Một số thẻ được sử dụng là:

<header>: Biểu thị nội dung giới thiệu, thường chứa các liên kết điều hướng hoặc thương hiệu trang web;

<footer>: Biểu thị chân trang của tài liệu hoặc phần, thường chứa thông tin bản quyền, liên kết hoặc thông tin liên hệ.

<article>: Biểu thị một thành phần độc lập có thể được phân phối độc lập, chẳng hạn như bài đăng trên blog hoặc bài báo tin tức.

<section>: Biểu thị một phần nội dung được nhóm theo chủ đề.

<nav>: Biểu thị một phần của trang dành cho các liên kết điều hướng.

<aside>: Biểu thị nội dung có liên quan gián tiếp đến nội dung chính, thường được hiển thị dưới dạng thanh bên.

<main>: Biểu thị nội dung chủ đạo của <body> trong tài liệu, không bao gồm nội dung như thanh bên, điều hướng, v.v.

- **Description**

HTML Semantics xác định ý nghĩa và cấu trúc của nội dung theo cách có ý nghĩa đối với cả con người và máy móc.

- **Features**

Improved Accessibility: Thẻ ngữ nghĩa cung cấp khả năng truy cập tốt hơn cho các công nghệ hỗ trợ, cho phép trình đọc màn hình điều hướng nội dung hiệu quả hơn.

Enhanced Readability: Thẻ ngữ nghĩa giúp mã HTML dễ đọc hơn và dễ bảo trì hơn bằng cách xác định rõ ràng cấu trúc và mục đích của nội dung.

Consistent Structure: Sử dụng thẻ ngữ nghĩa thúc đẩy cấu trúc nhất quán trên các trang web, cải thiện trải nghiệm người dùng và quản lý nội dung.

- **Code and Result**

Code	Result
------	--------

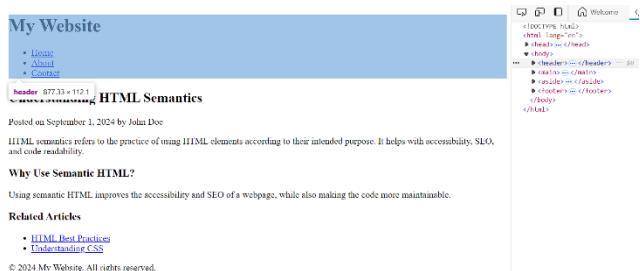
```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport"
content="width=device-width, initial-
scale=1.0">
<title>HTML Semantics Example</title>
</head>
<body>

<header>
<h1>My Website</h1>
<nav>
<ul>
<li><a href="#home">Home</a></li>
<li><a href="#about">About</a></li>
<li><a href="#contact">Contact</a></li>
</ul>
</nav>
</header>

<main>
<article>
<header>
<h2>Understanding HTML
Semantics</h2>
<p>Posted on September 1, 2024 by John
Doe</p>
</header>

```



```
<p>HTML semantics refers to the  
practice of using HTML elements  
according to their intended purpose. It  
helps with accessibility, SEO, and code  
readability.</p>  
</article>  
  
<section>  
<h3>Why Use Semantic HTML?</h3>  
<p>Using semantic HTML improves the  
accessibility and SEO of a webpage,  
while also making the code more  
maintainable.</p>  
</section>  
</main>  
  
<aside>  
<h3>Related Articles</h3>  
<ul>  
<li><a href="#article1">HTML Best  
Practices</a></li>  
<li><a href="#article2">Understanding  
CSS</a></li>  
</ul>  
</aside>  
<footer>  
<p>&copy; 2024 My Website. All rights  
reserved.</p>  
</footer>  
</body>
```

</html>

30. HTML Input Types

- **Tag**

<input>

- **Description**

HTML Input Types cho phép người dùng nhập dữ liệu vào biểu mẫu. Thuộc tính type chỉ định loại dữ liệu mà dữ liệu đầu vào sẽ chấp nhận.

Những input types thường được sử dụng trong HTML:

<input type="text">: Nhập văn bản một dòng tiêu chuẩn;

<input type="checkbox">: Cho phép người dùng chọn hoặc bỏ chọn một giá trị (đúng/sai);

<input type="color">: Cung cấp công cụ chọn màu để chọn màu;

<input type="date">: Cung cấp công cụ chọn ngày để chọn ngày;

<input type="datetime-local">

<input type="email">: Xác thực thông tin đầu vào là địa chỉ email hợp lệ;

<input type="file">: Cho phép người dùng tải tập tin lên;

<input type="number">: Cho phép nhập giá trị số, với tùy chọn min/max;

<input type="password">: Làm mờ văn bản nhập cho mật khẩu;

<input type="radio">: Cho phép người dùng chọn một tùy chọn từ một nhóm;

<input type="range">: Cung cấp thanh trượt để điều khiển việc chọn giá trị trong một phạm vi;

<input type="search">: Cung cấp trường tìm kiếm với nút rõ ràng

<input type="url">: Xác thực rằng đầu vào là một URL hợp lệ.

- **Features**

Various Input Types: giới thiệu nhiều loại đầu vào khác nhau giúp cải thiện trải nghiệm của người dùng và cung cấp các trường đầu vào cụ thể cho các loại dữ liệu khác nhau.

Client-Side Validation: Một số loại đầu vào như email, url và số cung cấp xác thực phía máy khách tích hợp.

User-Friendly: Các loại đầu vào như ngày, phạm vi và màu sắc đi kèm với các điều khiển UI chuyên biệt giúp tăng cường khả năng sử dụng.

- **Code and Result**

Code	Result
<pre data-bbox="127 264 693 1900"><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial- scale=1.0"> <title>HTML Input Types Example</title> </head> <body> <h1>HTML Input Types Example</h1> <form> <label for="username">Text Input:</label> <input type="text" id="username" name="username" placeholder="Enter your name">

 <label for="password">Password Input:</label> <input type="password" id="password" name="password" placeholder="Enter your password">

</pre>	<h2 data-bbox="752 234 1312 283">HTML Input Types Example</h2> <p data-bbox="752 312 1090 340">Text Input: <input data-bbox="866 312 1090 340" type="text" value="Enter your name"/></p> <p data-bbox="752 369 1139 397">Password Input: <input data-bbox="866 369 1139 397" type="password" value="Enter your password"/></p> <p data-bbox="752 426 1106 454">Email Input: <input data-bbox="866 426 1106 454" type="email" value="Enter your email"/></p> <p data-bbox="752 483 992 511">Number Input: <input data-bbox="866 483 992 511" type="number"/></p> <p data-bbox="752 540 1095 568">URL Input: <input data-bbox="866 540 1095 568" type="url" value="https://example.com"/></p> <p data-bbox="752 597 936 625">Color Input: <input data-bbox="866 597 936 625" type="color" value="#FF0000"/></p> <p data-bbox="752 654 1152 682">Checkbox Input: <input checked="" data-bbox="866 654 887 682" type="checkbox"/> Subscribe to newsletter</p> <p data-bbox="752 711 1062 739">Radio Input: <input type="radio"/> Male <input type="radio"/> Female</p>

```
<label for="email">Email Input:</label>
<input type="email" id="email"
name="email" placeholder="Enter your
email">
<br><br>

<label for="age">Number Input:</label>
<input type="number" id="age"
name="age" min="1" max="100">
<br><br>

<label for="website">URL Input:</label>
<input type="url" id="website"
name="website"
placeholder="https://example.com">
<br><br>

<label for="favcolor">Color
Input:</label>
<input type="color" id="favcolor"
name="favcolor" value="#ff0000">
<br><br>

<label>Checkbox Input:</label>
<input type="checkbox" id="subscribe"
name="subscribe" checked> Subscribe to
newsletter
<br><br>
```

```

<label>Radio Input:</label>
<input type="radio" id="male"
name="gender" value="male"> Male
<input type="radio" id="female"
name="gender" value="female"> Female
<br><br>
</form>
</body>
</html>

```

CSS (style)

1. CSS Color

Colors được chỉ định bằng cách sử dụng tên màu được xác định trước hoặc RGB, HEX, HSL và HSLA

- **Description**

Trong CSS, có thể chỉ định màu bằng cách sử dụng tên màu được xác định trước như: tomato, orange, DodgerBlue, MediumSeaGreen, Gray, SlateBlue, Violet, LightGray.

Trong CSS, có thể thiết lập màu cho text (văn bản), background (nền), borders (đường viền) và một số thuộc tính khác.

Syntax: selector { color: value; }

- Color có thể là: color (văn bản), background-color (nền), border: <line-width> || <line-style> || <color> (đường viền)
- Value có thể là: name color (e.g., red, blue, green), hexadecimal value (e.g., #ff0000 for red), RGB value (e.g., rgb(255,0,0) for red)

- **Features**

Text and Background Color: Sử dụng thuộc tính color cho văn bản và background-color cho nền.

Nhiều định dạng cho màu sắc:

Multiple Formats for Colors:

- Named color: CSS hỗ trợ hơn 140 màu được xác định trước như đỏ, xanh lá cây và xanh lam.

- Hexadecimal Colors: Được định nghĩa là #RRGGBB, trong đó RR, GG và BB là các giá trị thập lục phân.
- RGB (Đỏ, Xanh lá cây, Xanh lam): Sử dụng các giá trị RGB như rgb(255, 0, 0) cho màu đỏ.
- RGBA (Đỏ, Xanh lá cây, Xanh lam, Alpha): Thêm kênh alpha để tạo độ trong suốt, như rgba(255, 0, 0, 0,5) cho màu đỏ bán trong suốt.
- HSL (Sắc thái, Độ bão hòa, Độ sáng): Xác định màu theo sắc thái, độ bão hòa và độ sáng, như hsl(0, 100%, 50%) cho màu đỏ.
- HSLA (Sắc thái, Độ bão hòa, Độ sáng, Alpha): Thêm độ trong suốt vào các giá trị HSL, như hsla(0, 100%, 50%, 0,5) cho màu đỏ bán trong suốt.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8" /> <meta name="viewport" content="width=device-width, initial-scale=1.0" /> <title>CSS Colors Example</title> <style> body { background-color: #f0f0f0; /* Light gray background */ } h1 {</pre>	<p style="text-align: center;">CSS Colors Examples</p> <p>This text is blue using a named color.</p> <p>This text is red using RGB.</p> <p>This text is semi-transparent green using RGBA.</p> <p>This text is blue using HSL.</p> <p>This text is semi-transparent purple using HSLA.</p>

```
color: #ff5733; /* Hexadecimal color
(orange-red) */

}

p.named-color {
    color: blue; /* Named color */
}

p.rgb {
    color: rgb(255, 0, 0); /* RGB color (red) */
}

p.rgba {
    color: rgba(0, 128, 0, 0.5); /* RGBA color
(semi-transparent green) */
}

p.hsl {
    color: hsl(240, 100%, 50%); /* HSL color
(blue) */
}

p.hsla {
    color: hsla(
        300,
        100%,
        50%,
        0.5
    ); /* HSLA color (semi-transparent purple)
*/
```

```
}

</style>

</head>

<body>
<h1>CSS Colors Examples</h1>

<p class="named-color">This text is blue
using a named color.</p>

<p class="rgb">This text is red using
RGB.</p>

<p class="rgba">This text is semi-transparent
green using RGBA.</p>

<p class="hsl">This text is blue using
HSL.</p>

<p class="hsla">This text is semi-transparent
purple using HSLA.</p>

</body>
</html>
```

2. CSS Background

Thuộc tính nền CSS được sử dụng để thêm hiệu ứng nền cho các thành phần.

Cho phép bạn chỉ định nhiều thuộc tính nền khác nhau, chẳng hạn như màu nền, hình ảnh, vị trí, kích thước, v.v. Bằng cách sử dụng thuộc tính này, bạn có thể tăng cường sức hấp dẫn trực quan cho trang web của mình.

- **Description**

Syntax: selector {background: color;}

background có thể là: background-color, background-image, background-repeat, background-position, background-size, background-attachment, background-origin.

color có thể là: name color (e.g., red, blue, green), hexadecimal value (e.g., #ff0000 for red), RGB value (e.g., rgb(255,0,0) for red)

- **Features**

background-color: Chỉ định màu nền.

background-image: Chỉ định hình ảnh nền.

background-repeat: Chỉ định hình ảnh nền có lặp lại hay không/như thế nào.

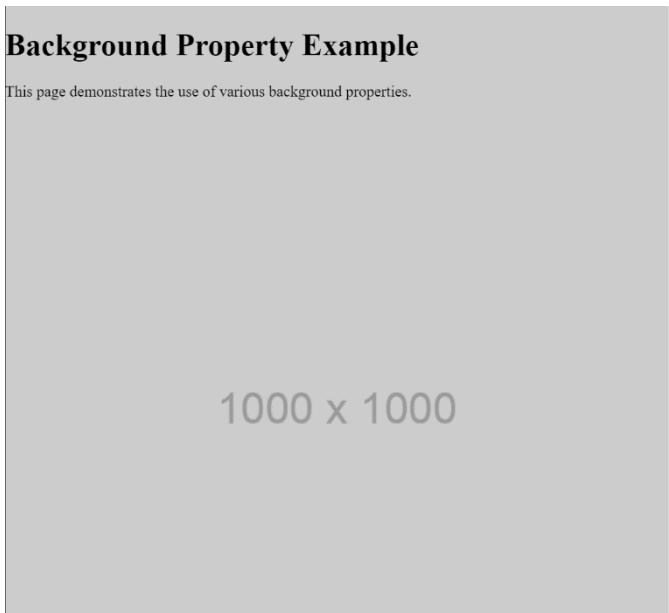
background-position: Chỉ định vị trí ban đầu của hình ảnh nền.

background-size: Chỉ định kích thước của hình ảnh nền.

background-attachment: Chỉ định hình ảnh nền có cố định hay cuộn theo trang không.

background-origin: Chỉ định vùng tô nền

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8" /> <meta name="viewport" content="width=device-width, initial-scale=1.0" /> <title>Background Property Example</title> <style> body { background-color: lightgray; width: 1000px; height: 1000px; } </style> </head> <body> </body> </html></pre>	<p>Background Property Example</p> <p>This page demonstrates the use of various background properties.</p>  <p>1000 x 1000</p>

```
background-color: #f0f0f0; /* Màu nền  
xám nhạt */  
  
background-image:  
url("https://via.placeholder.com/1000"); /* Ảnh  
giữ chỗ */  
  
background-repeat: no-repeat; /* Ngăn  
chặn việc lặp lại ảnh nền */  
  
background-position: center; /* Căn giữa  
ảnh nền */  
  
background-size: cover; /* Thay đổi tỷ lệ  
ảnh nền để bao phủ toàn bộ phần tử */  
  
background-attachment: fixed; /* Giữ  
nguyên ảnh nền khi cuộn */  
  
margin: 0; /* Xóa lề mặc định */  
  
height: 100vh; /* Đảm bảo phần thân chiếm  
toute bộ chiều cao của khung nhìn */  
}  
</style>  
</head>  
  
<body>  
<h1>Background Property Example</h1>
```

```

<p>This page demonstrates the use of various
background properties.</p>
</body>
</html>

```

3. CSS Border

Thuộc tính border của CSS được sử dụng để thiết lập đường viền của một phần tử. Nó cho phép bạn xác định chiều rộng, kiểu dáng và màu sắc của đường viền

- **Description**

Thuộc tính border là cách viết tắt để thiết lập các thuộc tính border riêng lẻ sau:

- border-width: Chỉ định chiều rộng của border.
- border-style: Chỉ định kiểu của border (ví dụ: liền, đứt nét, chấm).
- border-color: Chỉ định màu của border.

- **Features**

Border Width: Xác định độ dày hoặc mỏng của đường viền.

Border Style: Xác định giao diện của đường viền (ví dụ: liền, đứt nét, chấm bi, v.v.).

Border Color: Đặt màu của đường viền.

Individual Properties: Bạn có thể đặt độ rộng, kiểu và màu đường viền riêng biệt nếu cần.

Corner Radius: Sử dụng border-radius để tạo các góc bo tròn.

- **Code and Result**

Code	Result
<pre> <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8" /> <meta name="viewport" content="width=device-width, initial-scale=1.0" /> </pre>	<div style="border: 3px solid #3498db; width: 100px; height: 50px; margin: auto;"></div> <p>Div này có đường viền solid với chiều rộng là 3px và màu là #3498db (màu xanh lam).</p>

```

<title>Ví dụ về đường viền CSS</title>

<style>
    .border-example {
        border: 3px solid #3498db; /* Chiều rộng,
        kiểu và màu đường viền */
        padding: 20px; /* Thêm khoảng cách giữa
        đường viền và nội dung */
        margin: 20px; /* Thêm khoảng cách bên
        ngoài đường viền */
        border-radius: 10px; /* Tùy chọn: Làm tròn
        các góc của đường viền */
    }
</style>
</head>
<body>
    <div class="border-example">
        Div này có đường viền solid với chiều rộng
        là 3px và màu là #3498db (màu
        xanh lam).
    </div>
</body>
</html>

```

4. CSS Margin

Margin được sử dụng để tạo khoảng trống xung quanh các thành phần, bên ngoài bất kỳ đường viền nào được xác định.

- **Description**

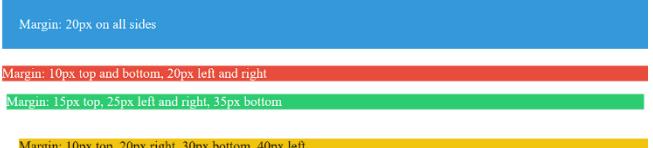
Thuộc tính margin chỉ định khoảng cách bên ngoài đường viền của phần tử. Thuộc tính này có thể được thiết lập riêng cho từng cạnh của phần tử hoặc trong thuộc tính viết tắt để thiết lập tất cả các lề cùng một lúc.

- **Features**

Uniform Margins: Đặt cùng một lề trên tất cả các mặt bằng một giá trị duy nhất.

Individual Margins: Chỉ định các lề khác nhau cho mỗi mặt bằng cách sử dụng tối đa bốn giá trị.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial- scale=1.0"> <title>CSS Margin Example</title> <style> .box { background-color: #3498db; /* Nền xanh đè dẽ nhìn */ color: white; padding: 20px; /* Thêm khoảng trống bên trong hộp */ margin: 20px; /* Thêm khoảng trống bên ngoài hộp ở tất cả các phía */ } .box-two-values { background-color: #e74c3c; /* Nền đỏ */ color: white; margin: 10px 20px; /* 10px trên và dưới, 20px trái và phải */ } </pre>	

```
.box-three-values {  
    background-color: #2ecc71; /* Nền xanh  
*/  
    color: white;  
    margin: 15px 25px 35px; /* 15px trên  
cùng, 25px bên trái và bên phải, 35px dưới cùng  
*/  
}  
  
.box-four-values {  
    background-color: #f1c40f; /* Nền vàng  
*/  
    color: black;  
    margin: 10px 20px 30px 40px; /* 10px  
trên cùng, 20px bên phải, 30px dưới cùng, 40px  
bên trái */  
}  
</style>  
</head>  
<body>  
  
<div class="box">  
    Margin: 20px on all sides  
</div>  
  
<div class="box-two-values">  
    Margin: 10px top and bottom, 20px left and  
right  
</div>
```

```

<div class="box-three-values">
    Margin: 15px top, 25px left and right, 35px
    bottom
</div>

<div class="box-four-values">
    Margin: 10px top, 20px right, 30px bottom,
    40px left
</div>

</body>
</html>

```

5. CSS Padding

Thuộc tính CSS padding được sử dụng để tạo khoảng trống bên trong một phần tử, giữa nội dung của phần tử và đường viền của nó.

- **Description**

Thuộc tính padding xác định khoảng cách giữa nội dung của một phần tử và đường viền của nó.

Nó có thể được thiết lập đồng đều ở tất cả các bên hoặc với các giá trị khác nhau cho mỗi bên.

- **Features**

Uniform Padding: Áp dụng cùng một đệm trên cả bốn cạnh của một phần tử.

Individual Padding: Chỉ định các giá trị đệm khác nhau cho các cạnh trên, phải, dưới và trái.

- **Code and Result**

Code	Result
<pre> <!DOCTYPE html> <html lang="en"> <head> </pre>	

```
<meta charset="UTF-8">
<meta name="viewport"
content="width=device-width, initial-
scale=1.0">
<title>CSS Padding Example</title>
<style>
.uniform-padding {
    background-color: #3498db; /* Nền xanh
đè dẽ nhìn */
    color: white;
    padding: 20px; /* Đệm 20px ở tất cả các
cạnh */
    margin: 10px; /* Thêm khoảng trống bên
ngoài hộp */
}
.two-values-padding {
    background-color: #e74c3c; /* Nền đỏ */
    color: white;
    padding: 10px 20px; /* 10px trên và
dưới, 20px trái và phải */
    margin: 10px;
}
</style>
</head>
<body>

<div class="uniform-padding">
    Padding: 20px on all sides
</div>
```

```

<div class="two-values-padding">
    Padding: 10px top and bottom, 20px left
    and right
</div>

</body>
</html>

```

6. CSS Height

Thuộc tính height của CSS chỉ định chiều cao của một phần tử.

- **Description**

Thuộc tính height thiết lập chiều cao của một phần tử. Thuộc tính này có thể được áp dụng cho các phần tử block-level, các phần tử inline-block và các phần tử container khác. Chiều cao có thể được xác định bằng các đơn vị tuyệt đối (như pixel) hoặc các đơn vị tương đối (như phần trăm).

- **Features**

Absolute Units: Đặt chiều cao bằng các đơn vị cố định như pixel (px), điểm (pt) hoặc cm (cm).

Relative Units: Sử dụng phần trăm (%) để đặt chiều cao tương đối với chiều cao của phần tử cha.

Auto Height: Giá trị mặc định tự động cho phép phần tử điều chỉnh chiều cao dựa trên nội dung của nó.

Min/Max Height: Sử dụng min-height và max-height để đặt các ràng buộc về chiều cao tối thiểu và tối đa.

- **Code and Result**

Code	Result
<pre> <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> </pre>	

```

<meta name="viewport"
      content="width=device-width, initial-
      scale=1.0">

<title>CSS Height Example</title>

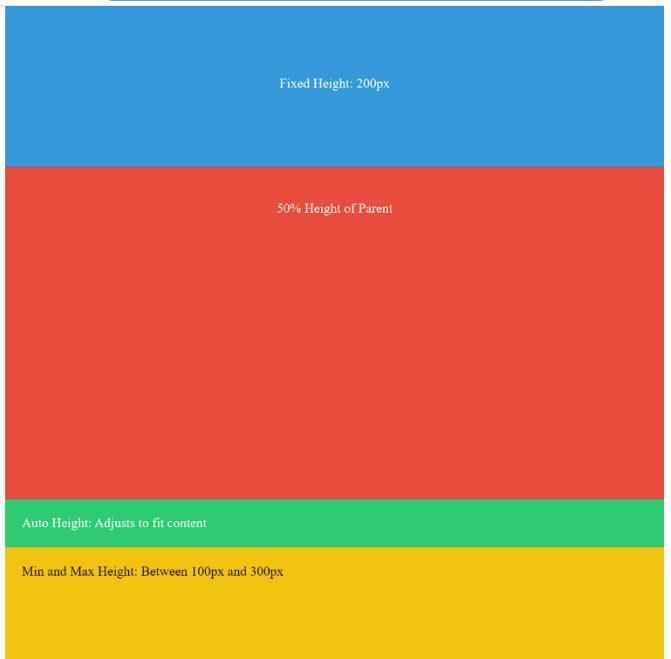
<style>

    .fixed-height {
        height: 200px; /* Chiều cao cố định là
        200 pixel */
        background-color: #3498db; /* Nền xanh
        để dễ nhìn */
        color: white;
        text-align: center;
        line-height: 200px; /* Căn giữa văn bản
        theo chiều đọc */
    }

    .percentage-height {
        height: 50%; /* 50% chiều cao của phần
        tử cha */
        background-color: #e74c3c; /* Nền đỏ */
        color: white;
        text-align: center;
        line-height: 100px; /* Điều chỉnh để căn
        giữa văn bản nếu cần */
    }

    .auto-height {
        height: auto; /* Chiều cao điều chỉnh cho
        phù hợp với nội dung */
    }

```



```
background-color: #2ecc71; /* Nền xanh
*/
color: white;
padding: 20px;
}

.min-max-height {
    min-height: 100px; /* Chiều cao tối thiểu
là 100 pixel */
    max-height: 300px; /* Chiều cao tối đa
300 pixel */
    background-color: #f1c40f; /* Nền vàng
*/
color: black;
padding: 20px;
overflow: auto; /* Thêm thanh cuộn nếu
nội dung vượt quá chiều cao tối đa */
}

</style>
</head>
<body>

<div class="fixed-height">
    Fixed Height: 200px
</div>

<div class="percentage-height" style="height:
50vh;">
    50% Height of Parent
</div>
```

```
<div class="auto-height">  
    Auto Height: Adjusts to fit content  
</div>  
  
<div class="min-max-height">  
    Min and Max Height: Between 100px and  
300px  
</div>  
  
</body>  
</html>
```

7. CSS Outline

Thuộc tính outline của CSS được sử dụng để tạo một đường bao quanh một phần tử, bên ngoài đường viền của phần tử đó.

- **Description**

Thuộc tính outline cung cấp một cách để vẽ một đường xung quanh một phần tử, thường là để làm nổi bật mục đích. Không giống như đường viền, outline không chiếm không gian hoặc ảnh hưởng đến kích thước của phần tử.

Syntax: selector { outline: [outline-width] [outline-style] [outline-color]; }

- outline-width: Specifies the width of the outline. (e.g., thin, medium, thick, or a specific size like 2px)
- outline-style: Specifies the style of the outline. (e.g., solid, dashed, dotted, double)
- outline-color: Specifies the color of the outline. (e.g., red, #ff0000, rgba(255,0,0,0.5))

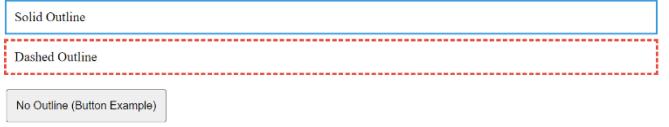
- **Features**

Non-Impact on Layout: Outline không ảnh hưởng đến mô hình hộp hoặc bố cục của phần tử; chúng được vẽ bên ngoài đường viền.

Customizable: Bạn có thể đặt màu, kiểu và chiều rộng của đường viền.

No Space Occupation: Không giống như đường viền, đường viền không chiếm thêm không gian hoặc ảnh hưởng đến bố cục của các phần tử xung quanh.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial- scale=1.0"> <title>CSS Outline Example</title> <style> .solid-outline { outline: 2px solid #3498db; /* Đường viền màu xanh lam đặc */ padding: 10px; margin: 10px; } .dashed-outline { outline: 3px dashed #e74c3c; /* Đường viền màu đỏ đứt nét */ padding: 10px; margin: 10px; } .outline-none { outline: none; /* Không có đường viền */ padding: 10px; } </style> </head> <body> <div>Solid Outline</div> <div>Dashed Outline</div> <div>No Outline (Button Example)</div> </body> </html></pre>	

```
margin: 10px;  
}  
</style>  
</head>  
<body>  
  
<div class="solid-outline">  
    Solid Outline  
</div>  
  
<div class="dashed-outline">  
    Dashed Outline  
</div>  
  
<button class="outline-none">  
    No Outline (Button Example)  
</button>  
  
</body>  
</html>
```

8. CSS Text

- **Description**

CSS cung cấp nhiều thuộc tính để định dạng và điều chỉnh văn bản trên trang web. Các thuộc tính sẽ giúp bạn kiểm soát cách văn bản hiển thị, từ màu sắc, phông chữ, kích thước, đến khoảng cách giữa các dòng và ký tự.

- ❖ **CSS Text color**

- **Features**

Color: được sử dụng để đặt màu sắc cho văn bản trên trang web. Thuộc tính này có thể áp dụng cho bất kỳ phần tử HTML nào có văn bản, từ đoạn văn, tiêu đề đến các liên kết và các phần tử văn bản khác.

Background-color: dùng để thiết lập màu nền cho các phần tử HTML. Màu nền có thể áp dụng cho toàn bộ trang web, một phần tử cụ thể, hoặc một nhóm phần tử, giúp tạo ra sự tương phản hoặc nhấn mạnh nội dung.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> body { background-color: lightgrey; color: green; } h1 { background-color: white; color: black; } div { background-color: pink; color: brown; } </style> </head> <body> <h1>Hello Everybody</h1> <p>Hôm nay thật là vui.</p></pre>	

```

<div>Đây là cái khung.</div>

</body>
</html>

```

❖ CSS Text Alignment

- Features

text-align: được sử dụng để căn chỉnh theo chiều ngang của văn bản. Một văn bản có thể được căn trái hoặc phải, căn giữa hoặc căn đều.

text-align-last: chỉ định cách căn chỉnh dòng cuối cùng của văn bản.

vertical-align: thiết lập căn chỉnh theo chiều dọc của một phần tử.

direction: chỉ định hướng văn bản/hướng viết.

unicode-bidi: được sử dụng cùng với thuộc tính direction để đặt hoặc trả về liệu văn bản có nên được ghi đè để hỗ trợ nhiều ngôn ngữ trong cùng một tài liệu.

- Code and Result

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> h1 { text-align: center; } img { vertical-align: text-top; } p.a { text-align-last: right; } div { border: 1px solid red; } </pre>	<p style="text-align: center;">Hello Baby</p> <p>Đây là con hổ 🐻</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p> <div style="border: 1px solid red; padding: 10px; margin: 10px auto; width: fit-content;"> <p>In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'</p> </div> <p>Hệ thống nhúng và IoT.</p> <p>.ToI àv gnúhn gnóht �H</p>

```
padding: 10px;  
width: 200px;  
height: 200px;  
text-align: justify;  
}  
  
p.ex1 {  
direction: rtl;  
unicode-bidi: bidi-override;  
}  
  
</style>  
</head>  
<body>  
  
<h1>Hello Baby</h1>  
  
<p>Đây là con hổ </p>  
  
<p class="a">Lorem ipsum dolor sit amet,  
consectetur adipiscing elit. Etiam semper diam  
at erat pulvinar, at pulvinar felis blandit.  
Vestibulum volutpat tellus diam, consequat  
gravida libero rhoncus ut.</p>  
  
<div>In my younger and more vulnerable years  
my father gave me some advice that I've been  
turning over in my mind ever since. 'Whenever  
you feel like criticizing anyone,' he told me, 'just  
remember that all the people in this world
```

```
haven't had the advantages that you've  
had.'</div>
```

```
<p>Hệ thống nhúng và IoT.</p>  
<p class="ex1">Hệ thống nhúng và IoT.</p>  
  
</body>  
</html>
```

❖ CSS Text Decoration

• Features

text-decoration-line: được, sử dụng để thêm đường trang trí vào văn bản. Có thể kết hợp nhiều giá trị, như gạch chân và gạch trên để hiển thị các dòng bên trên và bên dưới văn bản.

text-decoration-color: được sử dụng để thiết lập màu sắc của đường trang trí.

text-decoration-style: được sử dụng để thiết lập kiểu dáng của đường trang trí.

text-decoration-thickness: được sử dụng để thiết lập độ dày của đường trang trí.

• Code and Result

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> h1 { text-decoration: overline; text-decoration-color: red; text-decoration-style: dashed; } h2 { text-decoration-line: overline underline; text-decoration-style: wavy;</pre>	<p>Hôm nay là sinh nhật của em.</p> <p>Em thấy dui trong lòng.</p> <p>Mà em phải đi học.</p>

```

}

h3 {
    text-decoration-line: underline;
    text-decoration-color: purple;
    text-decoration-style: double;
    text-decoration-thickness: 5px;

}

</style>
</head>
<body>

<h1>Hôm nay là sinh nhật của em.</h1>
<h2>Em thấy dui trong lòng.</h2>
<h3>Mà em phải đi học.</h3>

</body>
</html>

```

❖ CSS Text Transformation

- Feature

text-transform: được sử dụng để chỉ định chữ hoa và chữ thường trong văn bản. Có thể sử dụng nó để chuyển mọi thứ thành chữ hoa hoặc chữ thường, hoặc viết hoa chữ cái đầu tiên của mỗi từ.

- Code and Result

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.uppercase { text-transform: uppercase; }</pre>	<p>Using the text-transform property</p> <p>THIS TEXT IS TRANSFORMED TO UPPERCASE. this text is transformed to lowercase. This Text Is Capitalized.</p>

```

p.lowercase {
    text-transform: lowercase;
}

p.capitalize {
    text-transform: capitalize;
}

</style>
</head>
<body>

<h1>Using the text-transform property</h1>

<p class="uppercase">This text is transformed
to uppercase.</p>
<p class="lowercase">This text is transformed
to lowercase.</p>
<p class="capitalize">This text is
capitalized.</p>

</body>
</html>

```

❖ CSS Text Spacing

- **Features**

text-indent: được sử dụng để chỉ định thụt lề cho dòng đầu tiên của văn bản.

letter-spacing: được sử dụng để xác định khoảng cách giữa các ký tự trong văn bản.

line-height: được sử dụng để chỉ định khoảng cách giữa các dòng.

word-spacing: được sử dụng để xác định khoảng cách giữa các từ trong văn bản

white-space: chỉ rõ cách xử lý khoảng trắng bên trong một phần tử.

- **Code and Result**

Code	Result
------	--------

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.indent {
```

```
    text-indent: 50px;
```

```
}
```

```
h1 {
```

```
    letter-spacing: 8px;
```

```
}
```

```
p.small {
```

```
    line-height: 0.7;
```

```
}
```

```
p.word {
```

```
    word-spacing: -2px;
```

```
}
```

```
p.nowrap {
```

```
    white-space: nowrap;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>CSS Text Spacing</h1>
```

```
<p class="indent">In my younger and more  
vulnerable years my father gave me some advice  
that I've been turning over in my mind ever  
since. 'Whenever you feel like criticizing
```

CSS Text Spacing

In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'

This is a paragraph with a smaller line-height.

This is a paragraph with smaller word spacing.

This is some text that will not wrap. This is some text that will not wrap. This is some text that will not wrap.

anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'</p>

```
<p class="small">  
This is a paragraph with a smaller line-height.<br>
```

```
This is a paragraph with a smaller line-height.<br>
```

```
</p>
```

```
<p class="word">This is a paragraph with  
smaller word spacing.</p>
```

```
<p class="nowrap">  
This is some text that will not wrap.  
This is some text that will not wrap.  
This is some text that will not wrap.  
</p>
```

```
</body>  
</html>
```

❖ CSS Text Shadow

• Description

CSS Text Shadow là một thuộc tính trong CSS được sử dụng để thêm bóng cho văn bản.

Cú pháp: **text-shadow: offset-x offset-y blur-radius color;**

• Features

Thuộc tính này cho phép bạn tạo ra các hiệu ứng đổ bóng, giúp văn bản trở nên nổi bật và hấp dẫn hơn.

• Code and Result

Code	Result
<pre data-bbox="115 192 817 1085"><!DOCTYPE html> <html> <head> <style> h1 { text-shadow: 2px 2px 5px red; } </style> </head> <body> <h1>Text-shadow effect!</h1> </body> </html></pre>	<p data-bbox="833 608 1465 671">Text-shadow effect!</p>

9. CSS Font

- **Description**

Font là kiểu chữ, mẫu mã của chữ, việc định dạng font chữ rất quan trọng để tạo ra giao diện đẹp mắt và dễ đọc, việc chọn đúng phông chữ có tác động rất lớn đến cách người đọc trải nghiệm một trang web và tạo thêm giá trị cho văn bản.

Phông chữ phù hợp có thể tạo nên bản sắc mạnh mẽ cho thương hiệu của bạn.

Sử dụng phông chữ dễ đọc là rất quan trọng. Phông chữ làm tăng giá trị cho văn bản của bạn.

Điều quan trọng nữa là chọn đúng màu sắc và kích thước chữ cho phông chữ.

- ❖ **Font Family**

- **Features**

font-family: định nghĩa kiểu chữ cho văn bản, có thể liệt kê nhiều font, cách nhau bởi dấu phẩy, để trình duyệt chọn font khả dụng.

Fallback Fonts: không có phông chữ nào an toàn 100% trên web. Luôn có khả năng phông chữ không được tìm thấy hoặc không được cài đặt đúng cách. Do đó, việc luôn sử dụng phông chữ dự phòng là rất quan trọng.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> .p1 { font-family: "Times New Roman", Times, serif; } .p2 { font-family: Arial, Helvetica, sans-serif; } .p3 { font-family: "Lucida Console", "Courier New", monospace; } </style> </head> <body> <h1>CSS font-family</h1> <p class="p1">This is a paragraph, shown in the Times New Roman font.</p> <p class="p2">This is a paragraph, shown in the Arial font.</p></pre>	<p>CSS font-family</p> <p>This is a paragraph, shown in the Times New Roman font.</p> <p>This is a paragraph, shown in the Arial font.</p> <p>This is a paragraph, shown in the Lucida Console font.</p>

```
<p class="p3">This is a paragraph, shown in the  
Lucida Console font.</p>
```

```
</body>  
</html>
```

❖ Font Style

• Features

font-style này có ba giá trị: bình thường, in nghiêng, xiên (xiên rất giống với nghiêng, nhưng ít được hỗ trợ hơn).

font-style: chủ yếu được sử dụng để chỉ định văn bản in nghiêng.

font-weight: chỉ định độ đậm của phông chữ.

font-variant: chỉ rõ liệu văn bản có được hiển thị bằng phông chữ in thường hay không.

• Code and Result

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> p.normal { font-style: normal; } p.italic { font-style: italic; font-weight: bold; } p.oblique { font-style: oblique; font-variant: small-caps; } </style></pre>	<h2>The font-style property</h2> <p>This is a paragraph in normal style.</p> <p><i>This is a paragraph in italic style.</i></p> <p><i>THIS IS A PARAGRAPH IN OBLIQUE STYLE.</i></p>

```

</head>
<body>

<h1>The font-style property</h1>

<p class="normal">This is a paragraph in
normal style.</p>
<p class="italic">This is a paragraph in italic
style.</p>
<p class="oblique">This is a paragraph in
oblique style.</p>

</body>
</html>

```

❖ Font Size

- **Features**

font-size: thiết lập kích thước văn bản. Có thể sử dụng đơn vị như px, em, %, ...

- **Code and Result**

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> p.a { font-size: 30px; } p.b { font-size: 1.875em; /* 30px/16=1.875em */ } p.c { </pre>	Sử dụng đơn vị px.
	Sử dụng đơn vị em.
	Sử dụng đơn vị %.

```

font-size: 200%;

}

</style>

</head>

<body>

<p class="a">Sử dụng đơn vị px.</p>
<p class="b">Sử dụng đơn vị em.</p>
<p class="c">Sử dụng đơn vị %.</p>

</body>
</html>

```

❖ Font Pairings

- Feature

Font pairings trong CSS là việc kết hợp hai hoặc nhiều font khác nhau để tạo ra một thiết kế hài hòa và dễ đọc.

- Code and Result

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> body { background-color: black; font-family: Verdana, sans-serif; font-size: 16px; color: gray; } h1 { </pre>	<p style="text-align: center;">Beautiful Norway</p> <p>Norway has a total area of 385,252 square kilometers and a population of 5,438,657 (December 2020). Norway is bordered by Sweden, Finland and Russia to the north-east, and the Skagerrak to the south, with Denmark on the other side.</p> <p>Norway has beautiful mountains, glaciers and stunning fjords. Oslo, the capital, is a city of green spaces and museums. Bergen, with colorful wooden houses, is the starting point for cruises to the dramatic Sognefjord. Norway is also known for fishing, hiking and skiing.</p>

```
font-family: Georgia, serif;  
font-size: 60px;  
color: white;  
}  
</style>  
</head>  
<body>  
  
<h1>Beautiful Norway</h1>  
  
<p>Norway has a total area of 385,252 square  
kilometers and a population of 5,438,657  
(December 2020). Norway is bordered by  
Sweden, Finland and Russia to the north-east,  
and the Skagerrak to the south, with Denmark  
on the other side.</p>  
  
<p>Norway has beautiful mountains, glaciers  
and stunning fjords. Oslo, the capital, is a city of  
green spaces and museums. Bergen, with  
colorful wooden houses, is the starting point for  
cruises to the dramatic Sognefjord. Norway is  
also known for fishing, hiking and skiing.</p>  
  
</body>  
</html>
```

10. CSS Icons

- **Description**

Trong CSS, bạn có thể sử dụng các biểu tượng (icons) để làm cho giao diện trang web trở nên sinh động và dễ hiểu hơn. Các biểu tượng có thể được thêm vào bằng nhiều cách khác nhau, chẳng hạn như sử dụng hình ảnh, font icon, SVG, hoặc các thư viện icon phổ biến.

Để thêm icons vào trang web, sử dụng thư viện icon :

Font Awesome: Một trong những thư viện phổ biến nhất, chỉ cần thêm đoạn mã sau vào phần <head> của trang HTML :

```
<script src="https://kit.fontawesome.com/a076d05399.js" crossorigin="anonymous"></script>
```

Bootstrap Icons: Nếu bạn đã sử dụng Bootstrap, có thể thêm các glyphicons bằng cách thêm dòng sau vào phần <head>:

```
<link rel="stylesheet"  
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

Google Material Icons: Google cung cấp một bộ icon phong phú, có thể thêm dòng sau vào phần <head>: <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">

• Features

Font Awesome Icons: là một trong những thư viện biểu tượng phổ biến nhất cho các nhà phát triển web, nhờ vào sự tiện lợi và tính linh hoạt của nó.

Bootstrap Icons: là một thư viện biểu tượng mã nguồn mở được thiết kế đặc biệt để tích hợp với Bootstrap framework, nhưng cũng có thể được sử dụng độc lập.

Google Icons: tuân theo ngôn ngữ thiết kế Material Design của Google, mang lại giao diện nhất quán, hiện đại và dễ sử dụng. Biểu tượng có đường nét đơn giản, rõ ràng và thân thiện với người dùng.

• Code and Result

Code	Result
<pre><!DOCTYPE html> <html> <head> <script src="https://kit.fontawesome.com/3f51c74bcf.j s" crossorigin="anonymous"></script></pre>	<p>Some Font Awesome icons:</p> <p>🎵 🌄 ❤️</p> <p>Some Bootstrap icons:</p> <p>☁️ ✖️ 🚶</p> <p>Some Google icons:</p> <p>⌚ 📱 ⚡</p>

```

<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<link rel="stylesheet"
      href="https://fonts.googleapis.com/icon?family=Material+Icons">
</head>
<body>

<h3>Some Font Awesome icons:</h3>
<i class="fa-solid fa-music"></i>
<i class="fas fa-cloud"></i>
<i class="fas fa-heart"></i>

<h3>Some Bootstrap icons:</h3>
<i class="glyphicon glyphicon-cloud"></i>
<i class="glyphicon glyphicon-remove"></i>
<i class="glyphicon glyphicon-user"></i>

<h3>Some Google icons:</h3>
<i class="material-icons">attachment</i>
<i class="material-icons">computer</i>
<i class="material-icons">traffic</i>

</body>
</html>

```

11. CSS Lists

- **Description**

Trong HTML, có hai loại danh sách chính:

- Danh sách không có thứ tự (``) - các mục danh sách được đánh dấu bằng dấu đầu dòng
- Danh sách có thứ tự (``) - các mục trong danh sách được đánh dấu bằng số hoặc chữ cái

Thuộc tính danh sách CSS cho phép bạn:

- Đặt các dấu hiệu mục danh sách khác nhau cho danh sách có thứ tự.
- Đặt các dấu hiệu mục danh sách khác nhau cho danh sách không có thứ tự.
- Đặt hình ảnh làm điểm đánh dấu mục danh sách.
- Thêm màu nền vào danh sách và các mục danh sách.

• Features

list-style-type: Xác định kiểu của các dấu đầu dòng hoặc số thứ tự.

list-style-image: thuộc tính chỉ định một hình ảnh làm điểm đánh dấu mục danh sách.

list-style-position: Xác định vị trí của các dấu đầu dòng hoặc số thứ tự.

list-style: Thuộc tính rút gọn để thiết lập tất cả các thuộc tính trên cùng một lúc.

• Code and Result

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> ul.a { list-style-type: circle; } ol.b { list-style-type: upper-roman; } ul.c { list-style-image: url('sqpurple.gif'); } ul.d { list-style-position: outside; }</pre>	<p>The list-style-type Property</p> <p>Example of unordered lists:</p> <ul style="list-style-type: none"> ◦ Coffee ◦ Tea ◦ Coca Cola <p>Example of ordered lists:</p> <ul style="list-style-type: none"> I. Coffee II. Tea III. Coca Cola <p>The list-style-image Property</p> <p>The list-style-image property specifies an image as the list item marker:</p> <ul style="list-style-type: none"> ■ Coffee ■ Tea ■ Coca Cola <p>List-style-position: outside (default):</p> <ul style="list-style-type: none"> • Coffee - A brewed drink prepared from roasted coffee beans, which are the seeds of berries from the Coffea plant <p>List-style-position: inside:</p> <ul style="list-style-type: none"> • Coffee - A brewed drink prepared from roasted coffee beans, which are the seeds of berries from the Coffea plant <p>The list-style Property</p> <p>The list-style property is a shorthand property, which is used to set all the list properties in one declaration.</p> <ul style="list-style-type: none"> ■ Coffee ■ Tea ■ Coca Cola

```
}

ul.e {
    list-style-position: inside;
}

ul.f {
    list-style: square inside url("sqpurple.gif");
}

</style>
</head>
<body>

<h2>The list-style-type Property</h2>
<p>Example of unordered lists:</p>
<ul class="a">
    <li>Coffee</li>
    <li>Tea</li>
    <li>Coca Cola</li>
</ul>

<p>Example of ordered lists:</p>
<ol class="b">
    <li>Coffee</li>
    <li>Tea</li>
    <li>Coca Cola</li>
</ol>

<h2>The list-style-image Property</h2>
<p>The list-style-image property specifies an
image as the list item marker:</p>
<ul class="c">
    <li>Coffee</li>
```

```
<li>Tea</li>
<li>Coca Cola</li>
</ul>

<h2>List-style-position: outside (default):</h2>
<ul class="d">
<li>Coffee - A brewed drink prepared from
roasted coffee beans, which are the seeds of
berries from the Coffea plant</li>
</ul>

<h2>List-style-position: inside:</h2>
<ul class="e">
<li>Coffee - A brewed drink prepared from
roasted coffee beans, which are the seeds of
berries from the Coffea plant</li>
</ul>

<h2>The list-style Property</h2>
<p>The list-style property is a shorthand
property, which is used to set all the list
properties in one declaration.</p>
<ul class="f">
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>

</body>
</html>
```

12. CSS Table

- **Description**

Table CSS là một bảng trong CSS được sử dụng để áp dụng các thuộc tính khác nhau cho các phần tử của bảng HTML. Các thuộc tính này được dùng để sắp xếp dữ liệu theo hàng và cột hoặc theo những cấu trúc phức tạp hơn.

Trong CSS, có thể định dạng bảng (table) để làm cho chúng trông đẹp mắt và dễ đọc hơn.

- ❖ **Table Borders**

- **Features**

border: Tạo đường viền cho bảng và các ô bên trong bảng.

border-collapse: Gộp các đường viền lân cận lại với nhau.

- **Code and Result**

Code	Result								
<pre><!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid; } table { width: 50%; border-collapse: collapse; } </style> </head> <body> <h2>Name</h2> <table> <tr></pre>	<p>Name</p> <table border="1"><thead><tr><th>Firstname</th><th>Lastname</th></tr></thead><tbody><tr><td>Hoang</td><td>Truong</td></tr><tr><td>Hao</td><td>Phan</td></tr><tr><td>Hoang</td><td>Mai</td></tr></tbody></table>	Firstname	Lastname	Hoang	Truong	Hao	Phan	Hoang	Mai
Firstname	Lastname								
Hoang	Truong								
Hao	Phan								
Hoang	Mai								

```

<th>Firstname</th>
<th>Lastname</th>
</tr>
<tr>
<td>Hoang</td>
<td>Truong</td>
</tr>
<tr>
<td>Hao</td>
<td>Phan</td>
</tr>
<tr>
<td>Hoang</td>
<td>Mai</td>
</tr>
</table>

</body>
</html>

```

❖ Table Size

- **Feature**

Chiều rộng và chiều cao của bảng được xác định bởi thuộc tính **width** và **height**.

- **Code and Result**

Code	Result															
<pre> <!DOCTYPE html> <html> <head> <style> table, td, th { </pre>	<p>The width and height Properties</p> <p>Set the width of the table, and the height of the table header row:</p> <table border="1"> <thead> <tr> <th>Firstname</th><th>Lastname</th><th>Savings</th></tr> </thead> <tbody> <tr> <td>Peter</td><td>Griffin</td><td>\$100</td></tr> <tr> <td>Lois</td><td>Griffin</td><td>\$150</td></tr> <tr> <td>Joe</td><td>Swanson</td><td>\$300</td></tr> <tr> <td>Cleveland</td><td>Brown</td><td>\$250</td></tr> </tbody> </table>	Firstname	Lastname	Savings	Peter	Griffin	\$100	Lois	Griffin	\$150	Joe	Swanson	\$300	Cleveland	Brown	\$250
Firstname	Lastname	Savings														
Peter	Griffin	\$100														
Lois	Griffin	\$150														
Joe	Swanson	\$300														
Cleveland	Brown	\$250														

```
border: 1px solid black;  
}
```

```
table {  
border-collapse: collapse;  
width: 100%;  
}
```

```
th {  
height: 70px;  
}  
</style>  
</head>  
<body>
```

<h2>The width and height Properties</h2>

<p>Set the width of the table, and the height of the table header row:</p>

```
<table>  
<tr>  
<th>Firstname</th>  
<th>Lastname</th>  
<th>Savings</th>  
</tr>  
<tr>  
<td>Peter</td>  
<td>Griffin</td>  
<td>$100</td>
```

```

</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
<td>$150</td>
</tr>
<tr>
<td>Joe</td>
<td>Swanson</td>
<td>$300</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Brown</td>
<td>$250</td>
</tr>
</table>

</body>
</html>

```

❖ Table Alignment

- **Features**

text-align: đặt căn chỉnh văn bản theo chiều ngang trong các ô (như phải , trái hoặc giữa).

vertical-align: đặt căn chỉnh văn bản theo chiều dọc (như trên, dưới hoặc giữa).

- **Code and Result**

Code	Result									
<!DOCTYPE html> <html> <head> <style>	<p>The text-align Property</p> <p>This property sets the horizontal alignment (like left, right, or center) of the content in th or td.</p> <table border="1"> <thead> <tr> <th>Firstname</th><th>Lastname</th><th>Savings</th></tr> </thead> <tbody> <tr> <td>Peter</td><td>Griffin</td><td>\$100</td></tr> <tr> <td>Lois</td><td>Griffin</td><td>\$150</td></tr> </tbody> </table>	Firstname	Lastname	Savings	Peter	Griffin	\$100	Lois	Griffin	\$150
Firstname	Lastname	Savings								
Peter	Griffin	\$100								
Lois	Griffin	\$150								

```
table, td, th {  
    border: 1px solid black;  
}  
  
table {  
    border-collapse: collapse;  
    width: 100%;  
}  
  
td {  
    text-align: center;  
}  
  
</style>  
</head>  
<body>
```

<h2>The text-align Property</h2>

<p>This property sets the horizontal alignment (like left, right, or center) of the content in th or td.</p>

```
<table>  
    <tr>  
        <th>Firstname</th>  
        <th>Lastname</th>  
        <th>Savings</th>  
    </tr>  
    <tr>  
        <td>Peter</td>  
        <td>Griffin</td>  
        <td>$100</td>  
    </tr>  
    <tr>
```

```
<td>Lois</td>  
<td>Griffin</td>  
<td>$150</td>  
</table>
```

```
</body>  
</html>
```

```
<!DOCTYPE html>
```

```
<html>  
<head>  
<style>  
table, td, th {  
    border: 1px solid black;  
}
```

```
table {  
    border-collapse: collapse;  
    width: 100%;  
}  
td {  
    height: 50px;  
    vertical-align: bottom;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>The vertical-align Property</h2>
```

```
<p>This property sets the vertical alignment  
(like top, bottom, or middle) of the content in th  
or td.</p>
```

The vertical-align Property

This property sets the vertical alignment (like top, bottom, or middle) of the content in th or td.

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150

```

<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
</table>

</body>
</html>

```

❖ Table Style

- **Features**

Để kiểm soát khoảng cách giữa đường viền và nội dung, sử dụng thuộc tính **padding** để thêm khoảng cách bên trong các ô trên các phần tử **<td>** và **<th>**.

border-bottom: Thêm thuộc tính này vào **<th>** và **<td>** cho dải phân cách ngang.

Code	Result
------	--------

```

<!DOCTYPE html>
<html>
<head>
<style>
table {
    border-collapse: collapse;
    width: 100%;
}

th, td {
    padding: 8px;
    text-align: left;
    border-bottom: 2px solid #ddd;
}

</style>
</head>
<body>

<h2>Table</h2>

```

```

<table>
<tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
</tr>
<tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
</tr>

```

Table

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

```

<tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
</tr>
<tr>
    <td>Joe</td>
    <td>Swanson</td>
    <td>$300</td>
</tr>
<tr>
    <td>Cleveland</td>
    <td>Brown</td>
    <td>$250</td>
</tr>
</table>

</body>
</html>

```

13. CSS Navigation Bar

- **Description**

Thanh điều hướng (navigation bar) là một phần quan trọng trong thiết kế giao diện trang web, giúp người dùng dễ dàng di chuyển giữa các trang hoặc tìm kiếm và truy cập vào các phần khác nhau của trang web.

Navigation bar là một tập hợp các liên kết hoặc mục menu, thường xuất hiện ở phía trên hoặc dưới đầu trang web.

Lợi ích của navigation bar : Dễ dàng điều hướng, tăng trải nghiệm người dùng, phân loại nội dung, tăng khả năng tương tác...

- ❖ **Navigation Bars**

- **Feature**

Navigation bar = List of links: thanh điều hướng về cơ bản là một danh sách các liên kết. Do đó việc sử dụng các phần tử `` và `` là hoàn toàn hợp lý

- **Code and Result**

Code	Result
<pre> <!DOCTYPE html> <html> <body> Home News Contact About <p>Note: We use href="#" for test links. In a real web site this would be URLs.</p> </body> </html> </pre>	<p style="text-align: center;"> Result </p> <ul style="list-style-type: none"> • Home • News • Contact • About <p>Note: We use href="#" for test links. In a real web site this would be URLs.</p>

- ❖ **Vertical navigation bar**

- **Feature**

Là thanh điều hướng dọc , thường được đặt bên trái hoặc phải của trang. Để tạo thanh điều hướng dọc, có thể tạo kiểu cho các phần tử `<a>` bên trong danh sách, ngoài mã từ trang trước.

- **Code and Result**

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> </pre>	

```

ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}

li a {
    display: block;
    width: 60px;
    background-color: #dddddd;
}

</style>
</head>
<body>

<ul>
<li><a href="#home">Home</a></li>
<li><a href="#news">News</a></li>
<li><a href="#contact">Contact</a></li>
<li><a href="#about">About</a></li>
</ul>

<p>A background color is added to the links to
show the link area.</p>
<p>Notice that the whole link area is clickable,
not just the text.</p>

</body>
</html>

```

Tạo thanh điều hướng dọc cơ bản với màu nền xám và thay đổi màu nền của các liên kết khi người dùng di chuyển chuột qua chúng.

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> ul { list-style-type: none; margin: 0; padding: 0; width: 200px; background-color: #f1f1f1; } li a { display: block; color: #000; padding: 8px 16px; text-decoration: none; } /* Change the link color on hover */ li a:hover { background-color: #555; color: white; } </style> </head> <body> <h2>Vertical Navigation Bar</h2> </pre>	<p style="text-align: center;">Vertical Navigation Bar</p>  <ul style="list-style-type: none"> Home News Contact About

```

<ul>
<li><a href="#home">Home</a></li>
<li><a href="#news">News</a></li>
<li><a href="#contact">Contact</a></li>
<li><a href="#about">About</a></li>
</ul>

</body>
</html>

```

Thêm một lớp hoạt động vào liên kết hiện tại để người dùng biết họ đang ở trang nào.

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> ul { list-style-type: none; margin: 0; padding: 0; width: 200px; background-color: #f1f1f1; } li a { display: block; color: #000; padding: 8px 16px; text-decoration: none; } </pre>	<p>Vertical Navigation Bar</p> <p>In this example, we create an "active" class with a green background color and a white text. The class is added to the "Home" link.</p> 

```

li a.active {
    background-color: #04AA6D;
    color: white;
}

li a:hover:not(.active) {
    background-color: #555;
    color: white;
}

```

</style>

</head>

<body>

<h2>Vertical Navigation Bar</h2>

<p>In this example, we create an "active" class with a green background color and a white text. The class is added to the "Home" link.</p>

- Home
- News
- Contact
- About

</body>

</html>

❖ Horizontal navigation bar

• Features

Là thanh điều hướng ngang, thường đặt ở đầu hàng.

Inline List Items: Một cách để xây dựng thanh điều hướng ngang là chỉ định các phần tử `` là nội tuyến.

Floating List Items: Một cách khác để tạo thanh điều hướng ngang là làm nổi các phần tử `` và chỉ định bối cảnh cho các liên kết điều hướng.

- **Code and Result**

Code	Result
<p style="text-align: center;">Inline list items</p> <pre><!DOCTYPE html> <html> <head> <style> ul { list-style-type: none; margin: 0; padding: 0; } li { display: inline; } </style> </head> <body> Home News Contact About </pre>	<p style="text-align: center;">Home News Contact About</p>

```
</body>  
</html>
```

Floating list items

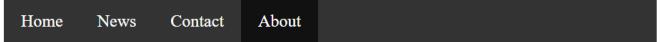
```
<!DOCTYPE html>  
  
<html>  
<head>  
<style>  
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    overflow: hidden;  
}  
  
li {  
    float: left;  
}  
  
li a {  
    display: block;  
    padding: 8px;  
    background-color: #dddddd;  
}  
  
</style>  
</head>  
<body>  
  
<ul>  
    <li><a href="#home">Home</a></li>  
    <li><a href="#news">News</a></li>  
    <li><a href="#contact">Contact</a></li>
```

```

<li><a href="#about">About</a></li>
</ul>
<p>A background color is added to the links to
show the link area. The whole link area is
clickable, not just the text.</p>
<p>
</body>
</html>

```

Tạo thanh điều hướng ngang cơ bản với màu nền tối và thay đổi màu nền của các liên kết khi người dùng di chuyển chuột qua chúng.

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> ul { list-style-type: none; margin: 0; padding: 0; overflow: hidden; background-color: #333; } li { float: left; } li a { display: block; color: white; } </pre>	

```

text-align: center;
padding: 14px 16px;
text-decoration: none;
}

li a:hover {
background-color: #111;
}

</style>
</head>
<body>

<ul>
<li><a class="active"
href="#home">Home</a></li>
<li><a href="#news">News</a></li>
<li><a href="#contact">Contact</a></li>
<li><a href="#about">About</a></li>
</ul>

</body>
</html>

```

Đặt thanh điều hướng ở đầu hoặc cuối trang, ngay cả khi cuộn trang.

Code	Result

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {margin:0;}
```

```
ul {
```

```
list-style-type: none;
```

```
margin: 0;
```

```
padding: 0;
```

```
overflow: hidden;
```

```
background-color: #333;
```

```
position: fixed;
```

```
top: 0;
```

```
width: 100%;
```

```
}
```

```
li {
```

```
float: left;
```

```
}
```

```
li a {
```

```
display: block;
```

```
color: white;
```

```
text-align: center;
```

```
padding: 14px 16px;
```

```
text-decoration: none;
```

```
}
```

```
li a:hover:not(.active) {
```



```
background-color: #111;  
}  
  
.active {  
background-color: #04AA6D;  
}  
/style>  
/head>  
<body>  
  
<ul>  
<li><a class="active"  
href="#home">Home</a></li>  
<li><a href="#news">News</a></li>  
<li><a href="#contact">Contact</a></li>  
<li><a href="#about">About</a></li>  
</ul>  
  
<div style="padding:20px;margin-  
top:30px;background-  
color:#1abc9c;height:1500px;">  
<h1>Fixed Top Navigation Bar</h1>  
<h2>Scroll this page to see the effect</h2>  
<h2>The navigation bar will stay at the top of  
the page while scrolling</h2>  
  
<p>Some text some text some text some  
text..</p>  
<p>Some text some text some text some  
text..</p>
```

<p>Some text some text some text..</p>

```

<p>Some text some text some text..</p>
<p>Some text some text some text some text..</p>
<p>Some text some text some text some text..</p>
</div>

</body>
</html>

```

14. CSS Image Gallery

- **Description**

Image gallery (bộ sưu tập hình ảnh) là một phần quan trọng trong thiết kế web, giúp hiển thị nhiều hình ảnh một cách có tổ chức và thẩm mỹ, tập hợp các hình ảnh được sắp xếp theo một cấu trúc nhất định thường được sử dụng để hiển thị danh mục sản phẩm, bộ sưu tập ảnh hoặc các dự án trong portfolio.

- **Code and Result**

Code	Result
<!DOCTYPE html>	
<html>	
<head>	
<style>	
div.gallery {	Add a description of the image here
margin: 5px;	
border: 1px solid #ccc;	
float: left;	
width: 180px;	
}	Add a description of the image here
div.gallery:hover {	
	Add a description of the image here

```
border: 1px solid #777;  
}  
  
div.gallery img {  
    width: 100%;  
    height: auto;  
}  
  
div.desc {  
    padding: 15px;  
    text-align: center;  
}  
  
</style>  
</head>  
<body>  
  
<div class="gallery">  
    <a target="_blank" href="img_5terre.jpg">  
          
    </a>  
    <div class="desc">Add a description of the  
image here</div>  
</div>  
  
<div class="gallery">  
    <a target="_blank" href="img_forest.jpg">  
          
    </a>  
    <div class="desc">Add a description of the  
image here</div>
```

```

</div>

<div class="gallery">
  <a target="_blank" href="img_lights.jpg">
    
  </a>
  <div class="desc">Add a description of the
image here</div>
</div>

<div class="gallery">
  <a target="_blank"
href="img_mountains.jpg">
    
  </a>
  <div class="desc">Add a description of the
image here</div>
</div>

</body>
</html>

```

15. CSS Website Layout

Website layout (bố cục trang web) là sự sắp xếp của tất cả các phần tử trực quan trên trang Web. Có rất nhiều thiết kế bố cục khác nhau để lựa chọn. Tuy nhiên, để tạo ra một website layout hiệu quả, bạn cần xem xét và tích hợp các yếu tố sau: Header, Navigation, Main Content, Footer,...

- ❖ **Header**

- **Description**

Header thường nằm ở đầu trang web (hoặc ngay bên dưới menu điều hướng trên cùng). Nó thường chứa logo hoặc tên trang web.

- **Features**

Nhận diện thương hiệu: Header thường chứa logo, tên thương hiệu, và slogan, giúp người dùng nhận diện trang web ngay lập tức.

Thông tin liên hệ: Header có thể chứa thông tin liên hệ như số điện thoại, email, và liên kết đến các mạng xã hội.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html lang="vi"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial- scale=1.0"> <title>Header Example</title> <style> body { font-family: Arial, sans-serif; margin: 0; padding: 0; background-color: #f3f3f3; } /* Style the header */ .header { background-color: #ffffff; padding: 20px; display: flex; align-items: center; }</pre>	

```
justify-content: space-between;  
border-bottom: 1px solid #ddd;  
}  
  
.header img {  
height: 50px;  
}  
  
.header p {  
margin: 0;  
font-size: 24px;  
font-weight: bold;  
position: absolute;  
left: 50%;  
transform: translateX(-50%);  
}  
  
</style>  
</head>  
<body>  
<header class="header">  
  
<p>Trang của tôi</p>  
</header>  
</body>  
</html>
```

❖ Navigation Bar

• Description

Thanh điều hướng chứa danh sách các liên kết để giúp khách truy cập điều hướng qua trang web của bạn.

- **Features**

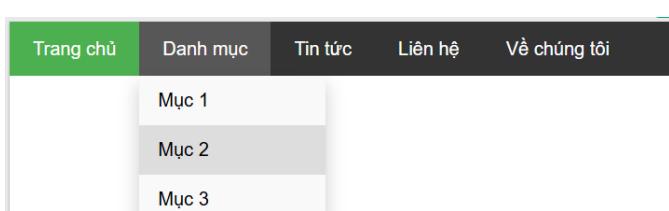
Điều hướng: Cung cấp liên kết nhanh đến các trang hoặc phần quan trọng của trang web (ví dụ: Trang chủ, Giới thiệu, Liên hệ, Dịch vụ).

Tăng trải nghiệm người dùng: Giúp người dùng dễ dàng tìm thấy thông tin họ cần mà không phải cuộn hay tìm kiếm quá nhiều.

Phân loại nội dung: Thanh NAV thường giúp tổ chức nội dung theo từng danh mục hoặc chủ đề, giúp người dùng hiểu rõ cấu trúc trang web.

Tìm kiếm: Nhiều trang web tích hợp thanh tìm kiếm trong thanh điều hướng để người dùng dễ dàng tìm kiếm nội dung trên trang.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html lang="vi"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial- scale=1.0"> <title>Thanh Điều Hướng</title> <style> body { margin: 0; font-family: Arial, sans-serif; } nav { background-color: #333; overflow: hidden; }</pre>	

```
nav a {  
    float: left;  
    display: block;  
    color: white;  
    text-align: center;  
    padding: 14px 20px;  
    text-decoration: none;  
}  
  
nav a:hover {  
    background-color: #575757;  
}  
  
nav a.active {  
    background-color: #4CAF50;  
    color: white;  
}  
  
.dropdown {  
    float: left;  
    overflow: hidden;  
}  
  
.dropdown .dropbtn {  
    font-size: 16px;  
    border: none;  
    outline: none;  
    color: white;  
    padding: 14px 20px;  
    background-color: inherit;  
    font-family: inherit;  
    margin: 0;  
}  
  
.dropdown-content {
```

```
display: none;  
position: absolute;  
background-color: #f9f9f9;  
min-width: 160px;  
box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);  
z-index: 1;  
}  
  
.dropdown-content a {  
float: none;  
color: black;  
padding: 12px 16px;  
text-decoration: none;  
display: block;  
text-align: left;  
}  
  
.dropdown-content a:hover {  
background-color: #ddd;  
}  
  
.dropdown:hover .dropdown-content {  
display: block;  
}  
  
.dropdown:hover .dropbtn {  
background-color: #575757;  
}  
}</style>  
</head>  
<body>  
<nav>  
<a href="#" class="active">Trang chủ</a>  
<div class="dropdown">
```

```

<button class="dropbtn">Danh mục
<i class="fa fa-caret-down"></i>
</button>

<div class="dropdown-content">
<a href="#">Mục 1</a>
<a href="#">Mục 2</a>
<a href="#">Mục 3</a>
</div>
</div>

<a href="#">Tin tức</a>
<a href="#">Liên hệ</a>
<a href="#">Về chúng tôi</a>
</nav>
</body>
</html>

```

❖ Content

- **Description**

Bộ cục của nội dung thường phụ thuộc vào mục đích của người dùng. Một số bộ cục phổ biến:

- 1 cột (thường được sử dụng cho trình duyệt di động)
- 2 cột (thường được sử dụng cho máy tính bảng và máy tính xách tay)
- Bộ cục 3 cột (chỉ được sử dụng cho máy tính để bàn)

- **Features**

Chứa và hiển thị các nội dung chính như bài viết, hình ảnh, và thông tin quan trọng.

Giúp tổ chức và hiển thị các nội dung theo cách khoa học, dễ dàng tiếp cận người dùng.

- **Code and Result**

Bộ cục nội dung 3 cột bằng nhau

Code	Result
------	--------

```

<!DOCTYPE html>
<html lang="vi">

    <head>
        <meta charset="UTF-8">
        <meta name="viewport"
            content="width=device-width, initial-
            scale=1.0">
        <style>
            * {
                box-sizing: border-box;
            }

            body {
                font-family: Arial, sans-serif;
                margin: 0;
                padding: 0;
            }

            /* Create three equal columns that floats next to
            each other */
            .column {
                float: left;
                width: 33.33%;
                padding: 15px;
            }

            /* Clear floats after the columns */
            .row::after {
                content: "";

```

Cột 1	Cột 2	Cột 3
Đây là nội dung của cột 1. Bạn có thể đặt văn bản, hình ảnh hoặc các phần tử khác vào đây.	Đây là nội dung của cột 2. Bạn có thể tùy chỉnh nội dung để phù hợp với yêu cầu của bạn.	Đây là nội dung của cột 3. Nội dung ở đây có thể là bất kỳ thông tin nào mà bạn muốn hiển thị.

```
display: table;
clear: both;
}

/* Responsive layout - makes the three columns
stack on top of each other instead of next to each
other */

@media screen and (max-width:600px) {
.column {
width: 100%;

}
}

</style>
</head>

<body>

<div class="row">
<div class="column">
<h2>Cột 1</h2>
<p>Đây là nội dung của cột 1. Bạn có thể đặt
văn bản, hình ảnh hoặc các phần tử khác vào
đây.</p>
</div>
<div class="column">
<h2>Cột 2</h2>
<p>Đây là nội dung của cột 2. Bạn có thể tùy
chỉnh nội dung để phù hợp với yêu cầu của
bạn.</p>
</div>
```

```

<div class="column">
<h2>Cột 3</h2>
<p>Đây là nội dung của cột 3. Nội dung ở đây
có thể là bất kỳ thông tin nào mà bạn muốn hiển
thi.</p>
</div>
</div>

</body>

</html>

```

Bố cục nội dung 3 cột không bằng nhau

Code	Result
<pre> <!DOCTYPE html> <html lang="vi"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial- scale=1.0"> <link rel="icon" href="https://yt3.googleusercontent.com/yc/AId ro_msxLLWr4lzT0Y82EROcPk0WPc3ycjD1p6 fgMVZ7RWcZwg=s160-c-k-c0x00fffff-no-rj"> <title>My Web</title> <style> * { </pre>	<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"> Cột 1 <p>Đây là nội dung của cột 1. Bạn có thể đặt văn bản, hình ảnh hoặc các phần tử khác vào đây.</p> </div> <div style="width: 30%;"> Cột 2 <p>Đây là nội dung của cột 2. Bạn có thể tùy chỉnh nội dung chính để phù hợp với yêu cầu của bạn.</p> </div> <div style="width: 30%;"> Cột 3 <p>Đây là nội dung của cột 3. Nội dung ở đây có thể là bất kỳ thông tin nào mà bạn muốn hiển thị.</p> </div> </div>

```
}

body {
font-family: Arial, sans-serif;
margin: 0;
padding: 0;
background-color: #ffffff;
}

/* Create three unequal columns that floats next
to each other */
.column {
float: left;
padding: 10px;
}

/* Left and right column */
.column.side {
width: 25%;
}

/* Middle column */
.column.middle {
width: 50%;
}

/* Clear floats after the columns */
.row::after {
content: "";
display: table;
clear: both;
}
```

```
/* Responsive layout - makes the three columns
stack on top of each other instead of next to each
other */

@media screen and (max-width: 600px) {

.column.side,
.column.middle {
width: 100%;
}

}

</style>
</head>
<body>
<!-- Content -->
<div class="row">
<div class="column side">
<h2>Cột 1</h2>
<p>Đây là nội dung của cột 1. Bạn có thể đặt
văn bản, hình ảnh hoặc các phần tử khác vào
đây.</p>
</div>
<div class="column middle">
<h2>Cột 2</h2>
<p>Đây là nội dung của cột 2. Bạn có thể tùy
chỉnh nội dung chính để phù hợp với yêu cầu
của bạn.</p>
</div>
<div class="column side">
<h2>Cột 3</h2>
```

```

<p>Đây là nội dung của cột 3. Nội dung ở đây  

có thể là bất kỳ thông tin nào mà bạn muốn hiển  

thị.</p>  

</div>  

</div>  

</body>  

</html>

```

❖ Footer

• Description

Footer được đặt ở cuối trang của bạn. Nó thường chứa Thông tin như bản quyền và thông tin liên hệ.

• Features

Thông tin liên hệ: Footer thường chứa thông tin liên hệ như địa chỉ email, số điện thoại, và địa chỉ văn phòng.

Liên kết quan trọng: Đây là nơi để đặt các liên kết đến các trang quan trọng khác trên website như trang chủ, giới thiệu, dịch vụ, và chính sách bảo mật.

Bản quyền: Footer thường chứa thông tin về bản quyền và điều khoản sử dụng.

Biểu tượng mạng xã hội: Các biểu tượng liên kết đến các trang mạng xã hội của doanh nghiệp cũng thường được đặt ở đây.

Thông tin bổ sung: Có thể bao gồm các thông tin như giờ làm việc, thông tin tuyển dụng, và các chi nhánh của công ty.

• Code and Result

Code	Result
<pre> <!DOCTYPE html> <html lang="vi"> <head> <meta charset="UTF-8"> </pre>	 <p>My website</p> <p>Slogan</p> <p>Chia sẻ những điều hay.</p> <p>Hướng dẫn chi tiết.</p> <p>THÔNG TIN CHUNG</p> <p>Trang chủ</p> <p>Danh mục</p> <p>Tin tức</p> <p>Liên hệ</p> <p>Về chúng tôi</p> <p>SOCIAL</p> <p>YOUTUBE</p> <p>FANPAGE</p> <p>FACEBOOK</p> <p>© Copyright HAO4 2024-2025. Designers by Phan Nhị Hào.</p>

```
<meta name="viewport"
      content="width=device-width, initial-
      scale=1.0">
<link
      href="https://fonts.googleapis.com/css?family=
      Baumans|Open+Sans" rel="stylesheet"
      type="text/css">
<title>Footer Example</title>
<style>
  * {
    box-sizing: border-box;
  }
  body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
  }
  a {
    text-decoration: none;
    color: inherit;
  }
  p {
    margin: 5px 0;
    line-height: 25px;
  }
  /*Footer*/
  footer {
    /* width: auto;
    height: auto; */
    background: #272727;
```

```
font-family: Open sans, Sans-serif;  
}  
  
footer #footer {  
width: auto;  
height: auto;  
max-width: 900px;  
margin: 0 auto;  
padding: 10px 0 15px 0;  
}  
  
footer #footer #column {  
width: 280px;  
display: inline-table;  
margin-top: 15px;  
}  
  
/*Column 1*/  
  
footer #footer #column #column1 {  
padding-left: 80px;  
}  
  
footer #footer #column #column1 h1 {  
font: inherit;  
font-family: Baumans;  
font-size: 30px;  
color: #ff0000cf;  
margin: auto;  
  
}  
  
footer #footer #column #column1 h2 {  
font-size: 16px;  
color: #05ccfbdf;  
padding-bottom: 5px;
```

```
line-height: 25px;  
}  
  
footer #footer #column #column1 p {  
font-size: 14px;  
color: #fff;  
line-height: 25px;  
}  
  
/*Column 2*/  
  
footer #footer #column #column2 {  
padding-left: 80px;  
}  
  
footer #footer #column #column2 h1 {  
font-size: 16px;  
color: #05ccfbef;  
text-transform: uppercase;  
padding-bottom: 5px;  
margin: auto;  
}  
  
footer #footer #column #column2 p {  
font-size: 14px;  
color: #fff;  
line-height: 15px;  
}  
  
footer #footer #column #column2 p a:hover {  
color: #9e9e9e;  
transition: color 0.3s;  
}  
  
/*Column 3*/  
  
footer #footer #column #column3 {  
padding-left: 80px;
```

```
}

footer #footer #column #column3 h1 {
    font-size: 16px;
    color: #05ccfbef;
    text-transform: uppercase;
    padding-bottom: 5px;
    margin: auto;
}

footer #footer #column #column3 p {
    font-size: 14px;
    color: #fff;
    line-height: 25px;
    text-transform: uppercase;
}

footer #footer #column #column3 p img {
    height: 13px;
    padding-right: 3px;
}

footer #footer #column #column3 a:hover {
    color: #9e9e9e;
    transition: color 0.3s;
}

/*Copyright*/
#copyright {
    background: #101010;
    padding: 5px 10px;
}

#copyright p {
    text-align: center;
    color: #9e9e9e;
```

```
}

#copyright p a {
color: #fff;
}

#copyright p a:hover {
color: #9e9e9e;
transition: color 0.3s;
}

</style>
</head>

<body>
<div>
<!--Begin - Footer-->
<footer>
<div id="footer">
<div id="column">
<div id="column1">
<h1><a href="#">My website</a></h1>
<h2>Slogan</h2>
<p>Chia sẻ những điều hay.</p>
<p>Hướng dẫn chi tiết.</p>
</div>
</div>
<div id="column">
<div id="column2">
<h1>Thông tin chung</h1>
<p><a href="#">Trang chủ</a></p>
<p><a href="#">Danh mục</a></p>
<p><a href="#">Tin tức</a></p>
```

```
<p><a href="#">Liên hệ</a></p>
<p><a href="#">Về chúng tôi</a></p>
</div>
</div>
<div id="column">
<div id="column3">
<h1>Social</h1>
<p><a
href="https://www.youtube.com/c/PhanNh%E1
%BB%8BH%C3%A0o">Youtube</a></p>
<p><a href="#">Fanpage</a></p>
<p><a
href="https://www.facebook.com/10001287601
3996">Facebook</a></p>
</div>
</div>
</footer>
<!--End - Footer-->
<!--Begin - Copyright-->
<div id="copyright">
<p>© Copyright <a
href="https://www.youtube.com/c/PhanNh%E1
%BB%8BH%C3%A0o">HAO04</a> 2024-
2025.
Designers by <a
href="https://www.facebook.com/10001287601
3996">Phan Nhị Hào</a>.</p>
```

```

</div>
<!--End - Copyright-->
</div>
</div>
</body>
</html>

```

❖ Responsive Website Layout

• Description

Bố cục của trang Web sẽ thay đổi tùy thuộc vào chiều rộng màn hình.

• Features

Tự động điều chỉnh kích thước và bố cục: Trang web sẽ tự động thay đổi kích thước và bố cục để phù hợp với kích thước màn hình của thiết bị, từ máy tính để bàn, máy tính xách tay, máy tính bảng đến điện thoại di động.

Cải thiện trải nghiệm người dùng: Responsive design giúp người dùng có trải nghiệm tốt hơn khi truy cập trang web trên các thiết bị khác nhau, không cần phải phóng to hoặc thu nhỏ để đọc nội dung.

Tối ưu hóa tốc độ tải trang: Bằng cách sử dụng các kỹ thuật như tải hình ảnh theo kích thước màn hình, responsive design giúp tối ưu hóa tốc độ tải trang, đặc biệt là trên các thiết bị di động có băng thông hạn chế.

• Code and Result

Code	Result
<pre> <!DOCTYPE html> <html> <head> <style> * { box-sizing: border-box; } </pre>	 <p>The screenshot shows a mobile-friendly website layout. At the top, there's a navigation bar with three dark blue buttons labeled 'Liên kết'. Below it is a large title 'TIÊU ĐỀ TIÊU ĐỀ' with a smaller subtitle 'Mô tả tiêu đề, Dec 7, 2017'. Underneath the title is a placeholder box with the word 'Ảnh'. To the right, there's a sidebar with a section titled 'Về tôi' containing a small profile picture labeled 'Ảnh' and a bio snippet: 'Một số văn bản về tôi trong culpa qui officia deserunt mollit anim..'. A blue downward arrow points from the bottom right of the screenshot towards the bottom of the page.</p>

```
body {  
    font-family: Arial;  
    padding: 10px;  
    background: #f1f1f1;  
}  
  
/* Header/Blog Title */  
  
.header {  
    padding: 30px;  
    text-align: center;  
    background: white;  
}  
  
.header h1 {  
    font-size: 50px;  
}  
  
/* Style the top navigation bar */  
  
.topnav {  
    overflow: hidden;  
    background-color: #333;  
}  
  
/* Style the topnav links */  
  
.topnav a {  
    float: left;  
    display: block;  
    color: #f2f2f2;  
    text-align: center;  
    padding: 14px 16px;
```

Trang web của tôi

Thay đổi kích thước cửa sổ trình duyệt để xem hiệu ứng.

Liên kết

Liên kết

Liên kết

Liên kết

TIÊU ĐỀ TIÊU ĐỀ

Mô tả tiêu đề, Dec 7, 2017

Ảnh

```
text-decoration: none;  
}  
  
/* Change color on hover */  
.topnav a:hover {  
background-color: #ddd;  
color: black;  
}  
  
/* Create two unequal columns that floats next  
to each other */  
/* Left column */  
.leftcolumn {  
float: left;  
width: 75%;  
}  
  

```

```
}

/* Add a card effect for articles */

.card {
background-color: white;
padding: 20px;
margin-top: 20px;
}

/* Clear floats after the columns */

.row::after {
content: "";
display: table;
clear: both;
}

/* Footer */

.footer {
padding: 20px;
text-align: center;
background: #ddd;
margin-top: 20px;
}

/* Responsive layout - when the screen is less
than 800px wide, make the two columns stack
on top of each other instead of next to each other
*/

@media screen and (max-width: 800px) {
.leftcolumn, .rightcolumn {
```

```
width: 100%;  
padding: 0;  
}  
}  
  
/* Responsive layout - when the screen is less  
than 400px wide, make the navigation links  
stack on top of each other instead of next to each  
other */  
@media screen and (max-width: 400px) {  
.topnav a {  
float: none;  
width: 100%;  
}  
}  
  
</style>  
</head>  
<body>  
<div class="header">  
<h1>Trang Web Của Tôi</h1>  
<p>Thay đổi kích thước cửa sổ trình duyệt để  
xem hiệu ứng.</p>  
</div>  
<div class="topnav">  
<a href="#">Liên kết</a>  
<a href="#">Liên kết</a>  
<a href="#">Liên kết</a>  
<a href="#" style="float:right">Liên kết</a>  
</div>  
<div class="row">
```

```
<div class="leftcolumn">  
<div class="card">  
<h2>TIÊU ĐỀ CHÍNH</h2>  
<h5>Mô tả tiêu đề, 7 Tháng 12, 2017</h5>  
<div class="fakeimg"  
style="height:200px;">Hình ảnh</div>  
<p>Một số văn bản..</p>  
<p>Sunt in culpa qui officia deserunt mollit  
anim id est laborum consectetur adipisciing elit,  
sed do eiusmod tempor incididunt ut labore et  
dolore magna aliqua. Ut enim ad minim veniam,  
quis nostrud exercitation ullamco.</p>  
</div>  
<div class="card">  
<h2>TIÊU ĐỀ CHÍNH</h2>  
<h5>Mô tả tiêu đề, 2 Tháng 9, 2017</h5>  
<div class="fakeimg"  
style="height:200px;">Hình ảnh</div>  
<p>Một số văn bản..</p>  
<p>Sunt in culpa qui officia deserunt mollit  
anim id est laborum consectetur adipisciing elit,  
sed do eiusmod tempor incididunt ut labore et  
dolore magna aliqua. Ut enim ad minim veniam,  
quis nostrud exercitation ullamco.</p>  
</div>  
</div>  
<div class="rightcolumn">  
<div class="card">  
<h2>Về Tôi</h2>
```

```

<div class="fakeimg"
style="height:100px;">Hình ảnh</div>
<p>Một số văn bản về tôi in culpa qui officia
deserunt mollit anim..</p>
</div>
<div class="card">
<h3>Bài Viết Phổ Biến</h3>
<div class="fakeimg"><p>Hình ảnh</p></div>
<div class="fakeimg"><p>Hình ảnh</p></div>
<div class="fakeimg"><p>Hình ảnh</p></div>
</div>
<div class="card">
<h3>Theo Dõi Tôi</h3>
<p>Một số văn bản..</p>
</div>
</div>
</div>
<div class="footer">
<h2>Chân Trang</h2>
</div>
</body>
</html>

```

16. CSS Rounded Corners

- **Description**

CSS Rounded Corners là một kỹ thuật trong CSS sử dụng thuộc tính border-radius để bo tròn các góc của phần tử.

- **Features**

Tạo giao diện mềm mại: Giúp các phần tử trên trang web có các góc bo tròn, tạo cảm giác mềm mại và thân thiện hơn so với các góc vuông cứng nhắc.

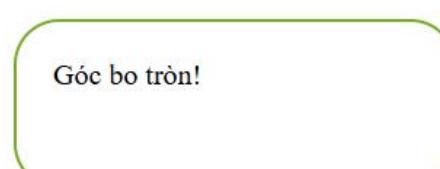
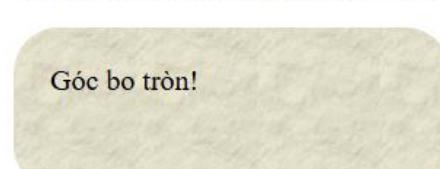
Tăng tính thẩm mỹ: Các góc bo tròn làm cho trang web trông hiện đại và chuyên nghiệp hơn, cải thiện trải nghiệm người dùng.

Linh hoạt trong thiết kế: Bạn có thể áp dụng border-radius cho bất kỳ phần tử nào, từ các khối văn bản, hình ảnh, đến các nút bấm, giúp tăng tính linh hoạt trong thiết kế.

Hỗ trợ nhiều giá trị: border-radius cho phép bạn chỉ định bán kính cho từng góc riêng biệt hoặc tất cả các góc cùng một lúc, thậm chí tạo các góc bo tròn theo hình elip.

• Code and Result

Bo tròn tất cả các góc

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> #rcorners1 { border-radius: 25px; background: #73AD21; padding: 20px; width: 200px; height: 50px; } #rcorners2 { border-radius: 25px; border: 2px solid #73AD21; padding: 20px; width: 200px; height: 50px; } #rcorners3 {</pre>	<p>Thuộc tính border-radius</p> <p>Góc bo tròn cho một phần tử có màu nền được chỉ định:</p>  <p>Góc bo tròn cho một phần tử có đường viền:</p>  <p>Góc bo tròn cho một phần tử có hình nền:</p> 

```

border-radius: 25px;
background: url(paper.gif);
background-position: left top;
background-repeat: repeat;
padding: 20px;
width: 200px;
height: 50px;
}

</style>
</head>
<body>

<h1>Thuộc tính border-radius</h1>

<p>Góc bo tròn cho một phần tử có màu nền  
được chỉ định:</p>
<p id="rcorners1">Góc bo tròn!</p>
<p>Góc bo tròn cho một phần tử có đường  
viền:</p>
<p id="rcorners2">Góc bo tròn!</p>
<p>Góc bo tròn cho một phần tử có hình  
nền:</p>
<p id="rcorners3">Góc bo tròn!</p>

</body>
</html>

```

Bo tròn từng góc riêng biệt

Code	Result
<!DOCTYPE html>	

```

<html>
<head>
<style>

#rcorners1 {
    border-radius: 15px 50px 30px 5px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 50px;
}

#rcorners2 {
    border-radius: 15px 50px 30px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 50px;
}

#rcorners3 {
    border-radius: 15px 50px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 50px;
}

#rcorners4 {
    border-radius: 15px;
    background: #73AD21;
}

```

Thuộc tính border-radius

Bốn giá trị - border-radius: 15px 50px 30px 5px:



Ba giá trị - border-radius: 15px 50px 30px:



Hai giá trị - border-radius: 15px 50px:



Một giá trị - border-radius: 15px:



```

padding: 20px;
width: 200px;
height: 50px;
}

</style>
</head>
<body>

<h1>Thuộc tính border-radius</h1>
<p>Bốn giá trị - border-radius: 15px 50px 30px
5px;</p>
<p id="rcorners1"></p>
<p>Ba giá trị - border-radius: 15px 50px
30px;</p>
<p id="rcorners2"></p>
<p>Hai giá trị - border-radius: 15px 50px;</p>
<p id="rcorners3"></p>
<p>Một giá trị - border-radius: 15px;</p>
<p id="rcorners4"></p>
</body>
</html>

```

17. CSS Multiple Columns

- **Description**

CSS Multiple Columns là một kỹ thuật trong CSS cho phép bạn chia nội dung thành nhiều cột, giống như cách bố trí văn bản trong các tờ báo.

- **Features**

Chia nội dung thành nhiều cột: Sử dụng thuộc tính column-count để xác định số lượng cột mà bạn muốn chia nội dung.

Khoảng cách giữa các cột: Sử dụng thuộc tính column-gap để xác định khoảng cách giữa các cột.

Đường viền giữa các cột: Sử dụng thuộc tính column-rule để đặt đường viền giữa các cột, bao gồm độ rộng, kiểu và màu sắc.

Trải dài qua nhiều cột: Sử dụng thuộc tính column-span để xác định một phần tử sẽ trải dài qua bao nhiêu cột.

Xác định chiều rộng cột: Sử dụng thuộc tính column-width để xác định chiều rộng tối thiểu của mỗi cột. Trình duyệt sẽ tự động điều chỉnh số lượng cột dựa trên chiều rộng này.

- **Code and Result**

Code	Result
<pre><!DOCTYPE html> <html> <head> <style> /* Định nghĩa lớp .new1 với 3 cột, khoảng cách giữa các cột là 100px và đường viền giữa các cột là 2px màu xanh nhạt */ .new1 { column-count: 3; column-gap: 100px; column-rule: 2px solid lightblue; } /* Định nghĩa tiêu đề h2 sẽ trải dài qua tất cả các cột */ h2 { column-span: all; } /* Định nghĩa lớp .new2 với chiều rộng cột là 100px và đường viền giữa các cột là 2px màu xanh nhạt */ .new2 { column-width: 100px; column-rule: 2px solid lightblue; }</pre>	<p>Tiêu Đề Trải Dài Qua Tất Cả Các Cột</p> <p>Các cột được chỉ định chiều rộng 100px</p> <p>The screenshot shows a web page with a header 'Tiêu Đề Trải Dài Qua Tất Cả Các Cột'. Below the header, there are three columns of text. The first column contains: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliqua.' The second column contains: 'erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.' The third column contains: 'nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.'</p>

```
</style>
</head>
<body>
<!-- Phần tử div với lớp .new1, chứa tiêu đề và
đoạn văn bản được chia thành 3 cột --&gt;
&lt;div class="new1"&gt;
&lt;h2&gt;Tiêu Đề Trái Dài Qua Tất Cả Các
Cột&lt;/h2&gt;
<p> Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh
euismod tincidunt ut laoreet dolore magna
aliquam erat volutpat. Ut wisi enim ad minim
veniam, quis nostrud exerci tation ullamcorper
suscipit lobortis nisl ut aliquip ex ea commodo
consequat.
</div>
<!-- Tiêu đề h2 không nằm trong bộ cục nhiều
cột --&gt;
&lt;h2&gt;Các cột được chỉ định chiều rộng
100px&lt;/h2&gt;
<!-- Phần tử div với lớp .new2, chứa đoạn văn
bản được chia thành các cột có chiều rộng 100px
--&gt;
&lt;div class="new2"&gt;
<p> Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh
euismod tincidunt ut laoreet dolore magna
aliquam erat volutpat. Ut wisi enim ad minim
veniam, quis nostrud exerci tation ullamcorper
```

```

suscipit lobortis nisl ut aliquip ex ea commodo
consequat.

</div>
</body>
</html>

```

18. CSS Buttons

- **Description**

CSS Buttons là một thuật ngữ dùng để chỉ việc tạo và tùy chỉnh các nút (buttons) trên trang web bằng cách sử dụng Cascading Style Sheets (CSS).

- **Features**

Tạo giao diện đẹp mắt: CSS cho phép bạn tùy chỉnh màu sắc, viền, bóng đổ, và các hiệu ứng khác để tạo ra các nút đẹp mắt và thu hút người dùng.

Cải thiện trải nghiệm người dùng: Các nút được thiết kế tốt giúp người dùng dễ dàng tương tác với trang web, từ việc nhấn nút để gửi biểu mẫu đến điều hướng giữa các trang.

Hiệu ứng tương tác: CSS cho phép bạn thêm các hiệu ứng khi người dùng di chuột qua hoặc nhấp vào nút, như thay đổi màu sắc, kích thước, hoặc thêm bóng đổ, giúp người dùng nhận biết rằng nút đang hoạt động.

- **Code and Result**

Features	Code	Result
Basic Button Styling	<pre> <!DOCTYPE html> <html> <head> <style> * { box-sizing: border-box; } .button { background-color: #04AA6D; border: none; } </pre>	<p>CSS Buttons</p> <p>The result shows four buttons labeled "Default Button", "Link Button", "Button", and "Input Button". The "Default Button" is a light gray button with a thin black border. The other three buttons are green buttons with white text. The "Link Button" has a slight shadow effect at the bottom. The "Button" and "Input Button" also have a slight shadow effect at the bottom.</p>

```
color: white;  
padding: 15px 32px;  
text-align: center;  
text-decoration: none;  
display: grid;  
font-size: 16px;  
margin: 4px 2px;  
cursor: pointer;  
width: 200px;  
}  
</style>  
</head>  
<body>  
<h2>CSS Buttons</h2>  
<button>Default Button</button>  
<a href="#" class="button">Link  
Button</a>  
<button class="button">Button</button>  
<input type="button" class="button"  
value="Input Button">  
</body>  
</html>
```

Button Colors	<pre> <!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: grid; font-size: 16px; margin: 4px 2px; cursor: pointer; width: 100px; columns: 2; } .button2 {background-color: #008CBA;} /* Blue */ .button3 {background-color: #f44336;} /* Red */ .button4 {background-color: #e7e7e7; color: black;} /* Gray */ .button5 {background-color: #555555;} /* Black */ </style> </head> <body> <h2>Button Colors</h2> </pre>	<h2>Button Colors</h2> <p>Thay đổi màu nền của nút bằng thuộc tính background-color:</p> 
---------------	---	---

	<pre> <p>Thay đổi màu nền của nút bằng thuộc tính background-color:</p> <button class="button">Green</button> <button class="button button2">Blue</button> <button class="button button3">Red</button> <button class="button button4">Gray</button> <button class="button button5">Black</button> </body> </html> </pre>	
Button Sizes	<pre> <!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: grid; margin: 4px 2px; cursor: pointer; } p{ width: 150px; </pre>	<p>Button Sizes</p> <p>Thay đổi kích thước phông chữ của nút bằng thuộc tính font-size:</p> 

```
}

.button1 {font-size: 10px;}

.button2 {font-size: 12px;}

.button3 {font-size: 16px;}

.button4 {font-size: 20px;}

.button5 {font-size: 24px;}

</style>

</head>

<body>

<h2>Button Sizes</h2>

<p>Thay đổi kích thước phông chữ của nút  
bằng thuộc tính font-size:</p>

<button class="button  
button1">10px</button>

<button class="button  
button2">12px</button>

<button class="button  
button3">16px</button>

<button class="button  
button4">20px</button>

<button class="button  
button5">24px</button>

</body>

</html>
```

```

<!DOCTYPE html>
<html>
<head>
<style>
.button {
background-color: #04AA6D; /* Green */
border: none;
color: white;
text-align: center;
text-decoration: none;
display: grid;
font-size: 16px;
margin: 4px 2px;
cursor: pointer;
}
.button1 {padding: 10px 24px;}
.button2 {padding: 12px 28px;}
.button3 {padding: 14px 40px;}
.button4 {padding: 32px 16px;}
.button5 {padding: 16px;}
</style>
</head>
<body>
<h2>Button Padding</h2>
<p>Thay đổi padding của nút bằng thuộc tính padding:</p>
<button class="button button1">10px  
24px</button>
<button class="button button2">12px  
28px</button>
<button class="button button3">14px  
40px</button>
<button class="button button4">32px  
16px</button>
<button class="button button5">16px</button>

```

Button Padding

Thay đổi padding của nút bằng thuộc tính padding:

10px 24px

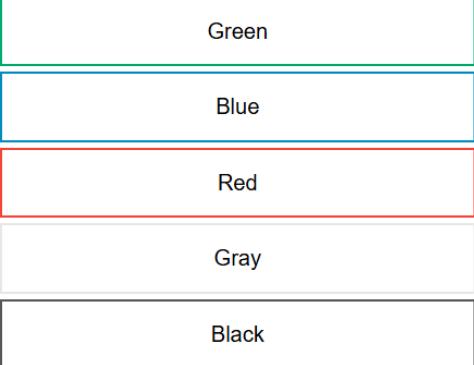
12px 28px

14px 40px

32px 16px

16px

	<pre> <button class="button button3">14px 40px</button> <button class="button button4">32px 16px</button> <button class="button button5">16px</button> </body> </html> </pre>	
Rounded Buttons	<pre> <!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 20px; text-align: center; text-decoration: none; display: grid; font-size: 16px; margin: 4px 2px; cursor: pointer; width: 200px; } .button1 {border-radius: 2px;} .button2 {border-radius: 4px;} .button3 {border-radius: 8px;} .button4 {border-radius: 16px;} .button5 {border-radius: 50%;} </pre>	<p>Rounded Buttons</p> <p>Thêm các góc bo tròn vào nút bằng thuộc tính border-radius:</p>  <p>The image shows five green rectangular buttons arranged vertically. From top to bottom, they have increasing amounts of rounded corners. The first button has a flat top and bottom (border-radius: 0px). The second has a slight curve (border-radius: 2px). The third has more rounded corners (border-radius: 4px). The fourth has almost full circular arcs (border-radius: 8px). The fifth has perfectly rounded corners (border-radius: 50%).</p>

	<pre> </style> </head> <body> <h2>Rounded Buttons</h2> <p>Thêm các góc bo tròn vào nút bằng thuộc tính border-radius:</p> <button class="button button1">2px</button> <button class="button button2">4px</button> <button class="button button3">8px</button> <button class="button button4">16px</button> <button class="button button5">50%</button> </body> </html> </pre>	
Colored Button Borders	<pre> <!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: grid; } </pre>	<p>Colored Button Borders</p> <p>Sử dụng thuộc tính border để thêm đường viền vào nút:</p>  <p>The image shows a vertical column of five rectangular boxes, each with a different colored border. From top to bottom, the colors are: Green, Blue, Red, Gray, and Black. Each box is a solid color with a thin border of the same color around its perimeter.</p>

```
font-size: 16px;  
margin: 4px 2px;  
cursor: pointer;  
width: 350px;  
}  
.button1 {  
background-color: white;  
color: black;  
border: 2px solid #04AA6D;  
}  
.button2 {  
background-color: white;  
color: black;  
border: 2px solid #008CBA;  
}  
.button3 {  
background-color: white;  
color: black;  
border: 2px solid #f44336;  
}  
.button4 {  
background-color: white;  
color: black;  
border: 2px solid #e7e7e7;  
}  
.button5 {  
background-color: white;  
color: black;  
border: 2px solid #555555;  
}
```

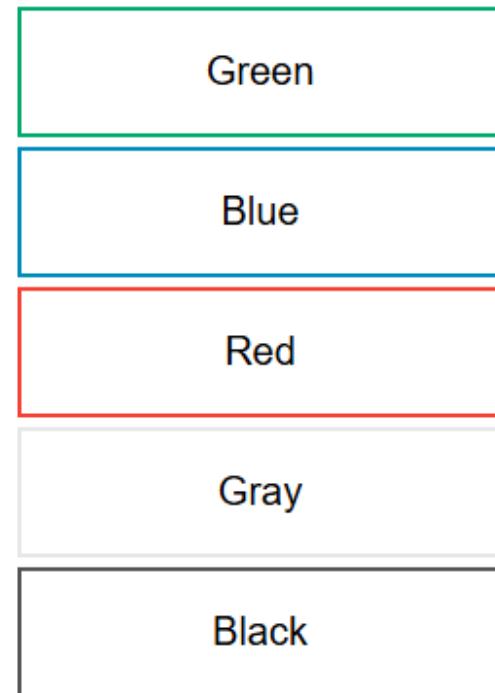
```
</style>
</head>
<body>
<h2>Colored Button Borders</h2>
<p>Sử dụng thuộc tính border để thêm
đường viền vào nút:</p>
<button class="button
button1">Green</button>
<button class="button
button2">Blue</button>
<button class="button
button3">Red</button>
<button class="button
button4">Gray</button>
<button class="button
button5">Black</button>
</body>
</html>
```

Hoverable Buttons	<pre> <!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 16px 32px; text-align: center; text-decoration: none; display: grid; font-size: 16px; margin: 4px 2px; transition-duration: 0.4s; cursor: pointer; width: 200px; } .button1 { background-color: white; color: black; border: 2px solid #04AA6D; } .button1:hover { background-color: #04AA6D; color: white; } .button2 { background-color: white; color: black; } </pre>
-------------------	--

Hoverable Buttons

Sử dụng bộ chọn :hover để thay đổi kiểu của nút khi bạn di chuyển chuột qua nút đó.

Mẹo: Sử dụng thuộc tính transition-duration để xác định tốc độ của hiệu ứng "hover":



```
border: 2px solid #008CBA;
}

.button2:hover {
background-color: #008CBA;
color: white;
}

.button3 {
background-color: white;
color: black;
border: 2px solid #f44336;
}

.button3:hover {
background-color: #f44336;
color: white;
}

.button4 {
background-color: white;
color: black;
border: 2px solid #e7e7e7;
}

.button4:hover {background-color:
#e7e7e7; }

.button5 {
background-color: white;
color: black;
border: 2px solid #555555;
}

.button5:hover {
background-color: #555555;
color: white;
```

```
}

</style>

</head>

<body>

<h2>Hoverable Buttons</h2>

<p>Sử dụng bộ chọn :hover để thay đổi  
kiểu của nút khi bạn di chuyển chuột qua  
nút đó.</p>

<p><strong>Mẹo:</strong> Sử dụng thuộc  
tính transition-duration để xác định tốc độ  
của hiệu ứng "hover":</p>

<button class="button  
button1">Green</button>

<button class="button  
button2">Blue</button>

<button class="button  
button3">Red</button>

<button class="button  
button4">Gray</button>

<button class="button  
button5">Black</button>

</body>

</html>
```

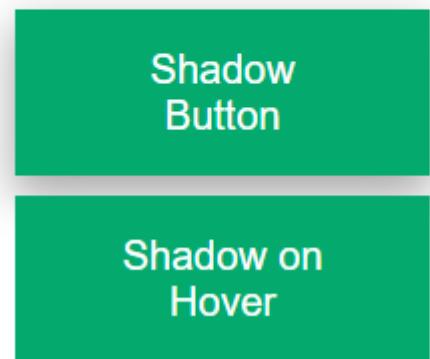
```

<!DOCTYPE html>
<html>
<head>
<style>
.button {
background-color: #04AA6D; /* Green */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
margin: 4px 2px;
cursor: pointer;
-webkit-transition-duration: 0.4s; /* Safari */
transition-duration: 0.4s;
}
.button1 {
box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2),
0 6px 20px 0 rgba(0,0,0,0.19);
}
.button2:hover {
box-shadow: 0 12px 16px 0
rgba(0,0,0,0.24),0 17px 50px 0
rgba(0,0,0,0.19);
}
</style>
</head>

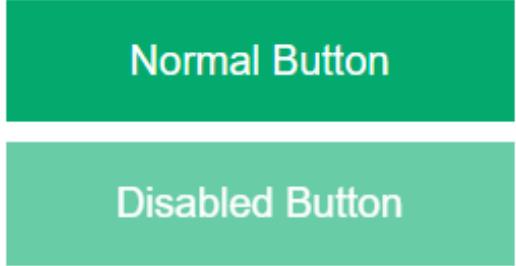
```

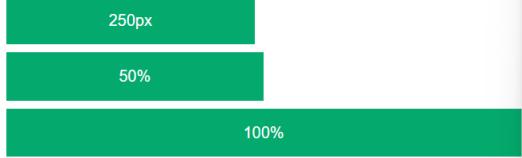
Shadow Buttons

Sử dụng thuộc tính box-shadow để thêm bóng cho nút:



Shadow
Buttons

	<pre> <body> <h2>Shadow Buttons</h2> <p>Sử dụng thuộc tính box-shadow để thêm bóng cho nút:</p> <button class="button button1">Shadow Button</button> <button class="button button2">Shadow on Hover</button> </body> </html> </pre>	
Disabled Buttons	<pre> <!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: inline-block; font-size: 16px; margin: 4px 2px; cursor: pointer; width: 200px; } .disabled { opacity: 0.6; cursor: not-allowed; } </pre>	<p>Disabled Button</p> <p>Sử dụng thuộc tính opacity để thêm độ trong suốt cho nút (làm cho nút trông giống như bị vô hiệu hóa):</p>  <p>Normal Button</p> <p>Disabled Button</p>

	<pre> } </style> </head> <body> <h2>Disabled Button</h2> <p>Sử dụng thuộc tính opacity để thêm độ trong suốt cho nút (làm cho nút trông giống như bị vô hiệu hóa):</p> <button class="button">Normal Button</button> <button class="button disabled">Disabled Button</button> </body> </html> </pre>	
Button Width	<pre> <!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: inline-block; font-size: 16px; margin: 4px 2px; cursor: pointer; } </pre>	<p>Set Button Widths</p> <p>Sử dụng thuộc tính width để thay đổi chiều rộng của nút:</p>  <p>Mẹo: Sử dụng pixel nếu bạn muốn đặt chiều rộng cố định và sử dụng phần trăm cho các nút phản hồi (ví dụ: 50% phần tử cha của nó). Thay đổi kích thước của trình duyệt để xem hiệu ứng.</p>

	<pre> .button1 {width: 250px;} .button2 {width: 50%;} .button3 {width: 100%;} </style> </head> <body> <h2>Set Button Widths</h2> <p>Sử dụng thuộc tính width để thay đổi chiều rộng của nút:</p> <button class="button button1">250px</button>
 <button class="button button2">50%</button>
 <button class="button button3">100%</button> <p>Mẹo: Sử dụng pixel nếu bạn muốn đặt chiều rộng cố định và sử dụng phần trăm cho các nút phản hồi (ví dụ: 50% phần tử cha của nó). Thay đổi kích thước cửa sổ trình duyệt để xem hiệu ứng.</p> </body> </html> </pre>	
Button Groups	<pre> <!DOCTYPE html> <html> <head> <style> .btn-group .button { background-color: #04AA6D; /* Green */ border: none; </pre>	<p>Button Groups</p> <p>Xóa lè và làm nổi các nút để tạo một nhóm nút:</p> <div style="display: flex; justify-content: space-around;"> Button Button Button Button </div> <p>Nhớ xóa các phần tử float sau đó, nếu không phần tử p này cũng sẽ float bên cạnh các nút.</p>

```
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
cursor: pointer;
float: left;
}
.btn-group .button:hover {
background-color: #3e8e41;
}
</style>
</head>
<body>
<h2>Button Groups</h2>
<p>Xóa lè và làm nổi các nút để tạo một nhóm nút:</p>
<div class="btn-group">
<button class="button">Button</button>
<button class="button">Button</button>
<button class="button">Button</button>
<button class="button">Button</button>
</div>
<p style="clear:both"><br>Nhớ xóa các phần tử float sau đó, nếu không phần tử này cũng sẽ float bên cạnh các nút.</p>
</body>
</html>
```

Bordered Button Group

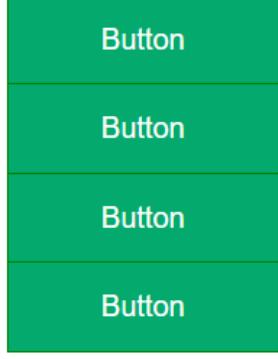
```
<!DOCTYPE html>
<html>
<head>
<style>
.btn-group .button {
background-color: #04AA6D; /* Green */
border: 1px solid green;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
cursor: pointer;
float: left;
}
.btn-group .button:not(:last-child) {
border-right: none; /* Prevent double
borders */
}
.btn-group .button:hover {
background-color: #3e8e41;
}
</style>
</head>
<body>
<h2>Bordered Button Group</h2>
<p>Thêm đường viền để tạo nhóm nút có
đường viền:</p>
<div class="btn-group">
```

Bordered Button Group

Thêm đường viền để tạo nhóm nút có đường viền:



Nhớ xóa các phần tử float sau đó, nếu không phần tử p này cũng sẽ
float bên cạnh các nút.

	<pre> <button class="button">Button</button> <button class="button">Button</button> <button class="button">Button</button> <button class="button">Button</button> </div> <p style="clear:both">
Nhớ xóa các phần tử float sau đó, nếu không phần tử p này cũng sẽ float bên cạnh các nút.</p> </body> </html> </pre>	
Vertical Button Group	<pre> <!DOCTYPE html> <html> <head> <style> .btn-group .button { background-color: #04AA6D; /* Green */ border: 1px solid green; color: white; padding: 15px 32px; text-align: center; text-decoration: none; font-size: 16px; cursor: pointer; width: 150px; display: block; } .btn-group .button:not(:last-child) { border-bottom: none; /* Prevent double borders */ } </pre>	<p>Vertical Button Group</p> 

```

.btn-group .button:hover {
background-color: #3e8e41;
}
</style>
</head>
<body>
<h2>Vertical Button Group</h2>
<div class="btn-group">
<button class="button">Button</button>
<button class="button">Button</button>
<button class="button">Button</button>
<button class="button">Button</button>
</div>
</body>
</html>

```

19. CSS The object-position Property

- **Description**

Thuộc tính object-position trong CSS được sử dụng để xác định vị trí của nội dung bên trong một phần tử thay thế (như hình ảnh hoặc video) trong hộp chứa của nó. Thuộc tính này thường được sử dụng cùng với object-fit để kiểm soát cách nội dung được căn chỉnh bên trong phần tử.

- **Features**

Căn chỉnh nội dung bên trong phần tử thay thế: object-position xác định vị trí của nội dung (như hình ảnh hoặc video) bên trong hộp chứa của phần tử thay thế.

Kiểm soát phần hiển thị của nội dung: Khi sử dụng cùng với object-fit, bạn có thể kiểm soát phần nào của nội dung sẽ được hiển thị trong hộp chứa. Ví dụ, bạn có thể căn chỉnh nội dung để hiển thị phần trung tâm, góc trên, hoặc bất kỳ vị trí nào khác.

Tạo hiệu ứng hình ảnh: object-position cho phép bạn tạo ra các hiệu ứng hình ảnh độc đáo bằng cách thay đổi vị trí của nội dung bên trong phần tử.

- **Code and Result**

Code	Result
<pre data-bbox="115 192 817 1928"><!DOCTYPE html> <html> <head> <style> div{ float: left; margin-right: 10px; max-width: 200px; text-align: center; } p{ height: 50px; margin: 0; } </style> </head> <body> <h2>CSS The object-position Property</h2> <div> <p>Đây là ảnh gốc</p> </div> <div> <p>Đây là ảnh sử dụng object-fit: cover</p> </div> <div></pre>	<p data-bbox="845 692 1379 734">CSS The object-position Property</p> <div data-bbox="845 760 1158 992" data-label="Image"> </div> <div data-bbox="922 998 1083 1028" data-label="Caption"> <p data-bbox="922 998 1083 1028">Đây là ảnh gốc</p> </div> <div data-bbox="1256 760 1379 992" data-label="Image"> </div> <div data-bbox="1166 998 1476 1056" data-label="Caption"> <p data-bbox="1166 998 1476 1056">Đây là ảnh sử dụng object-fit: cover</p> </div> <div data-bbox="938 1077 1068 1298" data-label="Image"> </div> <div data-bbox="848 1305 1155 1389" data-label="Caption"> <p data-bbox="848 1305 1155 1389">Đây là ảnh sử dụng object-position để hiển thị tòa nhà là chủ đề</p> </div> <div data-bbox="1256 1077 1379 1298" data-label="Image"> </div> <div data-bbox="1166 1305 1476 1389" data-label="Caption"> <p data-bbox="1166 1305 1476 1389">Đây là ảnh sử dụng object-position để hiển thị tháp Paris là chủ đề</p> </div>

```




Đây là ảnh sử dụng object-position để hiển
thị toà nhà là chủ đề


</div>
<div>



Đây là ảnh sử dụng object-position để hiển
thị tháp Paris là chủ đề


</div>
</body>
</html>

```

GIẢI THÍCH CODE SMART CITY.

```

<!-- Khai báo cho trình duyệt biết tệp đang sử dụng HTML5-->
<!DOCTYPE html>
<!-- Khai báo cho trình duyệt tệp đang sử dụng ngôn ngữ tiếng anh -->
<html lang="en">
<!-- Không hiện thị trực tiếp trên trình duyệt, chứa tiêu đề trang, tập tin css, javascript, bảng ký tự
cho trang web... -->
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Smart City</title>
<link rel="stylesheet" href="style.css" />
</head>
<body>

```

```
<!-- Khởi hiện thị tiêu đề Smart City -->
<div id="header">
    <ul>
        <li><b>Smart City</b></li>
        
    </ul>
</div>

<!-- Khởi hiện thị banner cùng với các nút ấn tương ứng với các quận để chuyển hướng -->
<div id="banner">
    <div class="tag">
        <button onclick="function_q1()"><b>Quận 1</b></button>
        <button onclick="function_q2()"><b>Quận 2</b></button>
        <button onclick="function_q3()"><b>Quận 3</b></button>
        <button onclick="function_q4()"><b>Quận 4</b></button>
    </div>
</div>

<!-- Khởi hiện thị chính của giao diện -->
<div id="main">
    <div id="top">
        <!-- Khởi hiện thị giờ hiện tại -->
        <div id="time"></div>
        <script>
            function dongho() {
                var time = new Date();
                var gio = time.getHours();
                var phut = time.getMinutes();
                var giay = time.getSeconds();
                if (gio < 10) gio = "0" + gio;
                if (phut < 10) phut = "0" + phut;
                if (giay < 10) giay = "0" + giay;
                document.getElementById("time").innerHTML =

```

```

gio + ":" + phut + ":" + giay;
setTimeout("dongho()", 1000);
}
dongho();
</script>
<!-- Load hình ảnh thang nhiệt độ lên giao diện --&gt;
&lt;div id="nd"&gt;
&lt;h1&gt;&lt;img src="image\nhietdo1.png" id="nd1" /&gt;&lt;/h1&gt;
&lt;h2&gt;Nhiệt độ&lt;/h2&gt;
&lt;/div&gt;
<!-- Load hình ảnh giọt nước tượng trưng cho độ ẩm lên giao diện --&gt;
&lt;div id="da"&gt;
&lt;h3&gt;&lt;img src="image\doam1.png" id="da1" /&gt;&lt;/h3&gt;
&lt;h4&gt;Độ ẩm&lt;/h4&gt;
&lt;/div&gt;
<!-- Load hình ảnh đám mây tượng trưng lượng mưa lên giao diện --&gt;
&lt;div id="lm"&gt;
&lt;h5&gt;&lt;img src="image\luongmua1.png" id="lm1" /&gt;&lt;/h5&gt;
&lt;h6&gt;Lượng mưa&lt;/h6&gt;
&lt;/div&gt;
<!-- Các thuộc tính hiện thị như nhiệt độ, độ ẩm, lượng mưa --&gt;
&lt;div class="thongso"&gt;
&lt;h7 id="nhietdo"&gt;&lt;/h7&gt;
&lt;h7 id="doam"&gt;&lt;/h7&gt;
&lt;h7 id="luongmua"&gt;&lt;/h7&gt;
&lt;/div&gt;
&lt;/div&gt;
<!-- Hiện thị khói phía dưới --&gt;
&lt;div id="bot"&gt;
<!-- Load ảnh điện thoại và thiết lập 2 nút ấn on, off --&gt;
&lt;h1&gt;
</pre>

```

```
<center></center>
<button type="button" id="btn1">On</button>
<button type="button" id="btn2">Off</button><br /><br />
</h1>
<!-- Load ảnh đèn và thiết lập 2 nút ấn on, off -->
<h1>
<center></center>
<button type="button" id="btn3">On</button>
<button type="button" id="btn4">Off</button><br /><br />
</h1>
<!-- Load ảnh loa và thiết lập 2 nút ấn on, off -->
<h1>
<center></center>
<button type="button" id="btn5">On</button>
<button type="button" id="btn6">Off</button><br /><br />
</h1>
<!-- Hiện thị tiêu đề phụ SOS, Đèn, Loa -->
<h2>SOS</h2>
<h3>Đèn</h3>
<h4>Loa</h4>
</div>
</div>
<div id="footer">
<ul>
<li>Mọi chi tiết xin liên hệ qua Hotline: 0337440783</li>
<li>Email: tinphambt49c@gmail.com</li>
</ul>
</div>
<script src="function.js"></script>
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/8.2.10.firebaseio-app.js"></script>
```

```

<script src="https://www.gstatic.com/firebasejs/8.2.10.firebaseio.js"></script>
<!-- TODO: Add SDKs for Firebase products that you want to use
https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="https://www.gstatic.com/firebasejs/8.2.10.firebaseio.analytics.js"></script>

<script>
  // Your web app's Firebase configuration
  // For Firebase JS SDK v7.20.0 and later, measurementId is optional
  var firebaseConfig = {
    apiKey: "AIzaSyCxgDsstpzYSu-9zvYPr23fy3kn0lHPYyg",
    authDomain: "projet1-9cf75.firebaseioapp.com",
    databaseURL: "https://projet1-9cf75-default.firebaseio.com",
    projectId: "projet1-9cf75",
    storageBucket: "projet1-9cf75.appspot.com",
    messagingSenderId: "73110806842",
    appId: "1:73110806842:web:48eb70b4127629a87c7003",
    measurementId: "G-QJM9N2V40P",
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
  firebase.analytics();
</script>
<!--Lấy dữ liệu về nhiệt độ, độ ẩm, và lượng mưa của các quận (quan1, quan2, quan3, quan4) từ
Firebase Realtime Database và liên kết các giá trị này với các phần tử HTML trên trang web để hiển
thị. -->
<script>
  //functions
  var nhietDo = document.getElementById("nhietdo");
  var dbRef = firebase.database().ref("quan1").child("Nhietdo");
  var doAm = document.getElementById("doam");
  var dbRef2 = firebase.database().ref("quan1").child("Doam");

```

```

var luongMua = document.getElementById("luongmua");
var dbRef3 = firebase.database().ref("quan1").child("Luongmua");

var nhietDo_q2 = document.getElementById("nhietdo");
var dbRef4 = firebase.database().ref("quan2").child("Nhietdo");
var doAm_q2 = document.getElementById("doam");
var dbRef5 = firebase.database().ref("quan2").child("Doam");
var luongMua_q2 = document.getElementById("luongmua");
var dbRef6 = firebase.database().ref("quan2").child("Luongmua");

var nhietDo_q3 = document.getElementById("nhietdo");
var dbRef7 = firebase.database().ref("quan3").child("Nhietdo");
var doAm_q3 = document.getElementById("doam");
var dbRef8 = firebase.database().ref("quan3").child("Doam");
var luongMua_q3 = document.getElementById("luongmua");
var dbRef9 = firebase.database().ref("quan3").child("Luongmua");

var nhietDo_q4 = document.getElementById("nhietdo");
var dbRef10 = firebase.database().ref("quan4").child("Nhietdo");
var doAm_q4 = document.getElementById("doam");
var dbRef11 = firebase.database().ref("quan4").child("Doam");
var luongMua_q4 = document.getElementById("luongmua");
var dbRef12 = firebase.database().ref("quan4").child("Luongmua");

```

</script>

<!-- Hàm JavaScript dùng để lấy dữ liệu từ Firebase Realtime Database cho khu vực quận 1 và cập nhật nội dung của các phần tử HTML tương ứng. -->

```

<script>

function function_q1() {
  dbRef.on("value", (snap) => (nhietDo.innerText = snap.val()));
  dbRef2.on("value", (snap) => (doAm.innerText = snap.val()));
  dbRef3.on("value", (snap) => (luongMua.innerText = snap.val()));
}

```

```
}

</script>

<!-- Hàm JavaScript dùng để lấy dữ liệu từ Firebase Realtime Database cho khu vực quận 2 và
cập nhật nội dung của các phần tử HTML tương ứng. --&gt;

&lt;script&gt;
    function function_q2() {
        dbRef4.on("value", (snap) =&gt; (nhietDo_q2.innerText = snap.val()));
        dbRef5.on("value", (snap) =&gt; (doAm_q2.innerText = snap.val()));
        dbRef6.on("value", (snap) =&gt; (luongMua_q2.innerText = snap.val()));

    }
&lt;/script&gt;

<!-- Hàm JavaScript dùng để lấy dữ liệu từ Firebase Realtime Database cho khu vực quận 3 và
cập nhật nội dung của các phần tử HTML tương ứng. --&gt;

&lt;script&gt;
    function function_q3() {
        dbRef7.on("value", (snap) =&gt; (nhietDo_q3.innerText = snap.val()));
        dbRef8.on("value", (snap) =&gt; (doAm_q3.innerText = snap.val()));
        dbRef9.on("value", (snap) =&gt; (luongMua_q3.innerText = snap.val()));

    }
&lt;/script&gt;

<!-- Hàm JavaScript dùng để lấy dữ liệu từ Firebase Realtime Database cho khu vực quận 4 và
cập nhật nội dung của các phần tử HTML tương ứng. --&gt;

&lt;script&gt;
    function function_q4() {
        dbRef10.on("value", (snap) =&gt; (nhietDo_q4.innerText = snap.val()));
        dbRef11.on("value", (snap) =&gt; (doAm_q4.innerText = snap.val()));
        dbRef12.on("value", (snap) =&gt; (luongMua_q4.innerText = snap.val()));

    }
&lt;/script&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

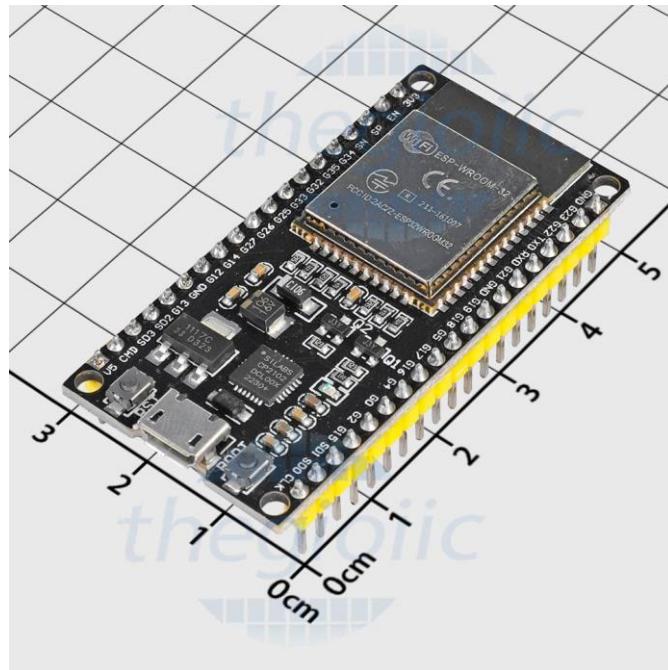
BÁO CÁO DASHBOARD SMART GARDEN SYSTEM

1. Giới thiệu

- Mục tiêu của Hệ thống:** Mục tiêu chính của hệ thống là cung cấp một giải pháp chăm sóc vườn tự động và tối ưu hóa điều kiện phát triển cho cây trồng. Thông qua giao diện đơn giản, người dùng có thể dễ dàng kiểm soát các thiết bị như quạt thông gió, đèn chiếu sáng, và hệ thống tưới nước từ xa. Những tính năng tự động hóa này giúp giảm thiểu thời gian và công sức chăm sóc vườn, đồng thời đảm bảo các điều kiện tốt nhất cho sự phát triển của cây.
- Ứng dụng của IoT trong Dự án:** SMART GARDEN SYSTEM là một ứng dụng IoT được thiết kế nhằm theo dõi và quản lý hiệu quả các yếu tố môi trường trong khu vườn. Dựa trên công nghệ IoT, hệ thống cho phép người dùng giám sát nhiệt độ, độ ẩm không khí, độ ẩm đất, và mức độ ánh sáng một cách chính xác và theo thời gian thực. Các dữ liệu môi trường này được lưu trữ và hiển thị thông qua kết nối với nền tảng Firebase, tạo ra trải nghiệm quản lý thuận tiện và tin cậy. Ngoài ra hệ thống còn liên kết với Thingspeak để trực quan các dữ liệu thành biểu đồ và hiển thị trực tiếp trên Dashboard để người dùng dễ dàng theo dõi khu vườn của mình.

2. Kiến trúc Hệ thống và Phần cứng

- Hệ thống "Giám sát Vườn Thông minh" là một ứng dụng IoT giúp theo dõi các yếu tố môi trường và quản lý các thiết bị trong vườn một cách hiệu quả và chính xác. Hệ thống bao gồm các thành phần phần cứng chính như ESP32, màn hình LCD 16x2, cảm biến nhiệt độ độ ẩm DHT22, cảm biến độ ẩm đất, cảm biến ánh sáng, cảm biến điện áp AC, cảm biến dòng AC và các nút nhấn điều khiển, hỗ trợ quản lý toàn diện môi trường vườn.
- Phần Cứng Sử Dụng:**
 - ESP32:** Đây là vi điều khiển trung tâm của hệ thống, chịu trách nhiệm thu thập dữ liệu từ các cảm biến và gửi lên Firebase để lưu trữ và xử lý. ESP32 cũng quản lý việc điều khiển các thiết bị trong vườn.



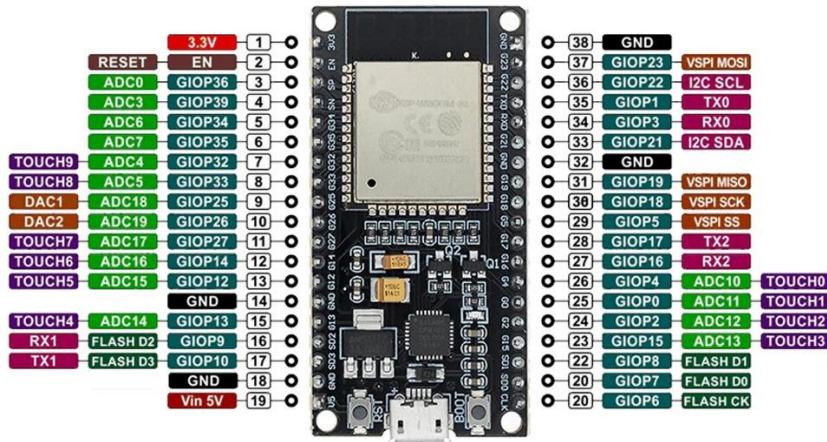
Hình 2.1: ESP32 NodeMCU LuaNode32 Module Thu Phát Wifi 38 Chân

Kit RF thu phát Wifi ESP32 NodeMCU LuaNode32 được phát triển trên nền module trung tâm là ESP32 với công nghệ Wifi, BLE và nhân ARM SoC tích hợp mới nhất hiện nay, kit có thiết kế phần cứng, firmware và cách sử dụng tương tự Kit NodeMCU ESP8266, với ưu điểm là cách sử dụng dễ dàng, ra chân đầy đủ.

Thông số kỹ thuật:

- Điện áp hoạt động: 5V (DC)
- Bộ nhớ: 520 KB SRAM, Flash 4MB (có thể mở rộng đến 16MB)
- Kết nối: Wi-Fi 802.11 b/g/n, Bluetooth v4.2
- Chân GPIO: Hỗ trợ 38 chân GPIO với khả năng PWM, ADC, DAC
- Kích thước: 54.9mm x 27.9mm
- Tiêu thụ năng lượng: Chế độ ngủ sâu khoảng 10 μ A, hoạt động bình thường khoảng 160mA

PINOUT ESP32 38 PINES ESP WROOM 32



Hình 2.2: Sơ đồ chân của ESP32 NodeMCU LuaNode32

- ❖ **LCD 16x2 (kết nối I2C):** Màn hình LCD 16x2 kết nối qua giao thức I2C giúp đơn giản hóa việc kết nối và sử dụng ít chân trên ESP32. Màn hình này hiển thị thông tin về các giá trị môi trường như nhiệt độ, độ ẩm không khí, độ ẩm đất, và ánh sáng, cho phép người dùng theo dõi trạng thái của hệ thống ngay tại vị trí lắp đặt.



Hình 2.3: LCD 1602 Kèm I2C Driver

Đây là LCD ký tự 16x2 chữ trắng nền xanh dương có IC điều khiển HD44780 Super Twisted Nematic (STN) LCD có tích hợp Module I2C để giảm số lượng chân giao tiếp. Giao tiếp dễ dàng với các loại vi điều khiển và board Arduino, ATMEL AVR hoặc PIC.

Thông số kỹ thuật:

- Điện áp hoạt động: Chủ yếu từ 4.5V đến 5.5V.
- Dòng tiêu thụ: Bình thường chỉ khoảng 1mA.
- Hiển thị: 16 dòng và 2 cột.
- Kích thước Module: 80 x 36 x 13.5mm
- Kích thước vùng nhìn: 66 x 16mm



Hình 2.4: Mạch Giao Tiếp LCD Ký Tự 1602 Sang I2C

Sơ đồ chân:

- SDA (Serial Data Line): Chân dữ liệu nối tiếp.
- SCL (Serial Clock Line): Chân đồng hồ nối tiếp.
- VCC: Chân nguồn, cung cấp điện áp 5V cho module.
- GND (Ground): Chân nối đất.

❖ **Cảm biến DHT22:** Đây là cảm biến đo nhiệt độ và độ ẩm không khí, cung cấp dữ liệu với độ chính xác cao, đặc biệt hữu ích cho việc kiểm soát điều kiện khí hậu trong vườn. Dữ liệu thu thập từ DHT22 được cập nhật lên Firebase và hiển thị trên cả dashboard trực tuyến và màn hình LCD 16x2.



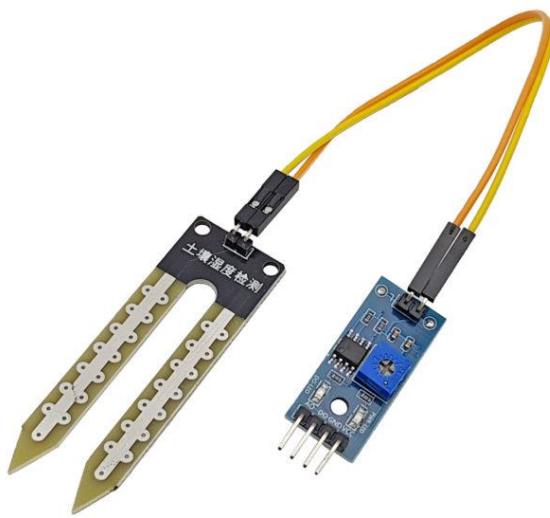
Hình 2.5: Cảm biến độ ẩm, nhiệt độ DHT22

Thông số kỹ thuật:

- Nguồn sử dụng: 3~5 VDC.
- Dòng sử dụng: 2.5mA max (khi truyền dữ liệu).
- Đo tốt ở độ ẩm 0 → 100%RH với sai số 2%.
- Đo tốt ở nhiệt độ -40 → 80°C sai số ±0.5°C.
- Tần số lấy mẫu tối đa 0.5Hz (2 giây 1 lần)
- Kích thước 27mm x 59mm x 13.5mm (1.05" x 2.32" x 0.53")

Sơ đồ chân:

- Chân + (VCC): Cung cấp nguồn điện cho cảm biến. Điện áp hoạt động thường từ 3.3V đến 5V.
 - Chân OUT (DATA): Chân dữ liệu, truyền tín hiệu nhiệt độ và độ ẩm từ cảm biến đến vi điều khiển.
 - Chân - (GND): Chân nối đất.
- ❖ **Cảm biến độ ẩm đất:** Cảm biến này đo mức độ ẩm trong đất, giúp hệ thống xác định khi nào cần kích hoạt hoặc tắt hệ thống tưới nước, đảm bảo rằng cây trồng luôn nhận được lượng nước thích hợp.



Hình 2.6: Cảm Biến Độ Âm Đất SMS-V1

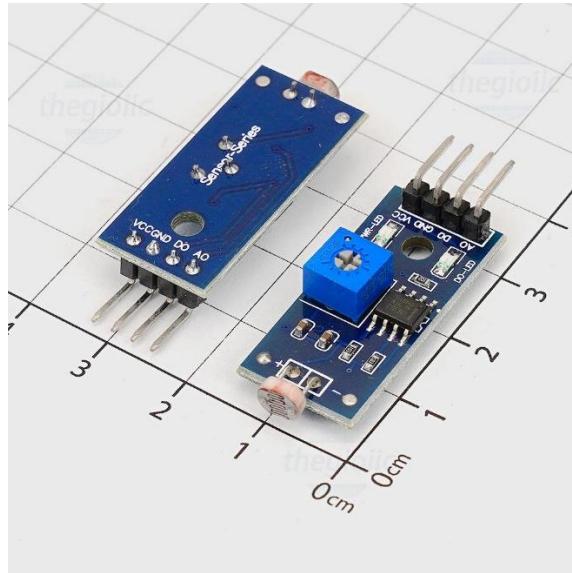
Thông số kỹ thuật:

- Điện áp hoạt động: 3.3~5VDC
- Tín hiệu đầu ra:
 - Analog: theo điện áp cấp nguồn tương ứng.
 - Digital: High hoặc Low, có thể điều chỉnh độ ẩm mong muốn bằng biến trờ thông qua mạch so sánh LM393 tích hợp.
- Kích thước: 3 x 1.6cm.

Sơ đồ chân:

- Chân VCC: Cung cấp nguồn điện cho cảm biến. Điện áp hoạt động thường từ 3.3V đến 5V.
- Chân GND: Chân nối đất.
- Chân DO: Đầu ra tín hiệu số (mức cao hoặc mức thấp)
- Chân AO: Đầu ra tín hiệu tương tự (Analog)

❖ **Cảm biến ánh sáng LM393:** Cảm biến ánh sáng giúp giám sát mức độ ánh sáng trong vườn. Thông qua dữ liệu này, người dùng có thể điều chỉnh hệ thống chiếu sáng, đảm bảo cây trồng nhận được đủ ánh sáng trong những thời điểm cần thiết.



Hình 2.7: Cảm biến ánh sáng quang trở CDS Light Sensor

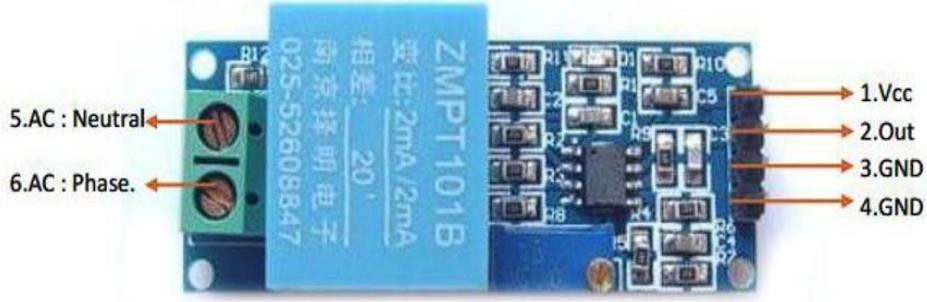
Thông số kỹ thuật:

- Sử dụng cảm biến photoresistor loại nhạy
- Dùng IC so sánh LM393 cho dòng ra lớn nhất 15mA.
- Nguồn cấp 3.3~5 VDC
- Đầu ra digital DO (0 và 1)
- Đầu ra điện áp analog AO
- Kích thước: 3.2 x 1.4cm

Sơ đồ chân:

- Chân VCC: Cung cấp nguồn điện cho cảm biến. Điện áp hoạt động thường từ 3.3V đến 5V.
- Chân GND: Chân nối đất.
- Chân DO: Đầu ra tín hiệu số (mức cao hoặc mức thấp)
- Chân AO: Đầu ra tín hiệu tương tự (Analog)

❖ **Cảm biến điện áp AC:** Cảm biến điện áp giúp theo dõi thông số điện năng trong hệ thống. Các thông số này hỗ trợ quản lý mức tiêu thụ điện và bảo vệ các thiết bị điện trong hệ thống.



Hình 2.8: Cảm biến điện áp AC Voltage Sensor ZMPT101B

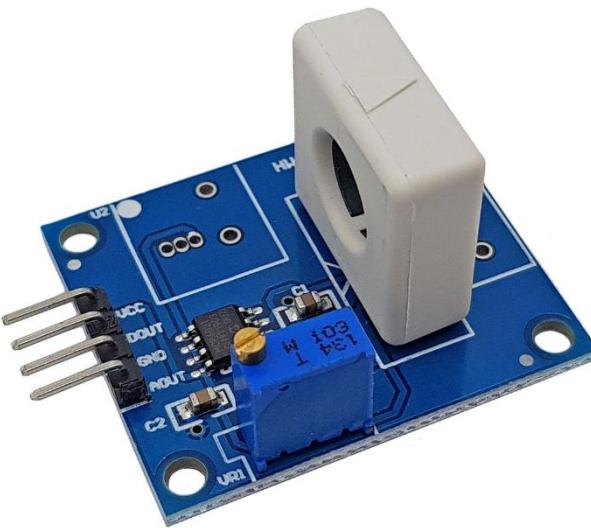
Thông Số Kỹ Thuật:

- Nguồn sử dụng: 5V đến 30V DC.
- Điện áp cảm biến tối đa: 250V AC.
- Tín hiệu đầu ra: Analog 0~5V DC (tương ứng với điện áp đầu vào).
- Biến áp chính xác trên bo mạch: Đảm bảo độ chính xác cao trong việc chuyển đổi tín hiệu.
- Chính được tín hiệu đầu ra: Có thể điều chỉnh giá trị đầu ra analog.
- Độ chính xác cao: Đảm bảo sự đồng nhất khi đo điện áp và công suất.
- Kích thước: 49.5 mm x 19.4 mm.
- Nhiệt độ hoạt động: Từ -40°C đến +70°C.

Sơ đồ chân:

- Chân VCC: Cung cấp nguồn điện cho cảm biến. Điện áp hoạt động thường từ 3.3V đến 5V.
- Chân GND: Chân nối đất.
- Chân OUT: Chân OUT sẽ cung cấp tín hiệu analog, bạn có thể đọc tín hiệu này từ vi điều khiển như Arduino hoặc Raspberry Pi thông qua cổng analog.

- ❖ **Cảm biến dòng AC:** Cảm biến dòng điện giúp theo dõi thông số điện năng trong hệ thống. Các thông số này hỗ trợ quản lý mức tiêu thụ điện và bảo vệ các thiết bị điện trong hệ thống.



Hình 2.9: Cảm biến dòng điện DC/AC Hall 35A WCS1800

Thông số kỹ thuật

- IC chính: LM393, WCS1800
- Điện áp làm việc: 5VDC
- Phạm vi phát hiện dòng điện:
- DC: +-35A
- AC: 25A
- Độ tuyến tính: $K = 60\text{mV / A}$
- Tín hiệu quá dòng có thể điều chỉnh bằng biến trở, độ phân giải cài đặt là: 1.5A
- Đèn báo tín hiệu quá dòng nhô ra
- Kích thước: 36.7 x 31.5 x 22mm
- Trọng lượng: 10g

Sơ đồ chân:

- VCC: Kết nối với nguồn 5V DC.
- GND: Kết nối với đất của hệ thống.
- AOUT: Đọc tín hiệu analog từ chân này. Tín hiệu đầu ra sẽ thay đổi tỷ lệ với dòng điện chạy qua cảm biến, cho phép bạn tính toán giá trị dòng điện.

- ❖ **Module relay:** Module relay được sử dụng như một công tắc điện tử để điều khiển các thiết bị điện áp cao (AC/DC) như quạt, đèn, bơm nước, hoặc các thiết bị khác.



Hình 2.10: Mạch 8 Relay Opto cách ly 5VDC

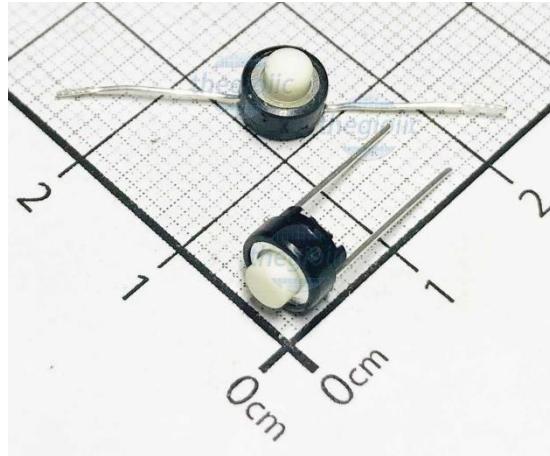
Thông số kỹ thuật:

- Điện áp sử dụng: 5VDC
- Tín hiệu kích: TTL 3.3~5VDC, mức thấp Low Relay đóng, mức cao High Relay ngắt.
- Mỗi Relay tiêu thụ dòng khoảng 80mA.
- Điện thế đóng ngắt tối đa: AC250V ~ 10A hoặc DC30V ~ 10A (Để an toàn nên dùng cho tải có công suất <100W).
- Tích hợp Opto cách ly, Diod chống nhiễu và đèn báo tín hiệu kích.
- Kích thước: 137 x 56 x 20mm

Sơ đồ chân:

- VCC → 5V nguồn.
- GND → Ground.
- IN → Tín hiệu 5V từ vi điều khiển để kích hoạt relay.
- NO, NC, COM → Kết nối với thiết bị điện cần điều khiển

- ❖ **Nút Nhấn Điều Khiển:** Nút nhấn cho phép người dùng điều khiển thiết bị trực tiếp tại khu vực vườn tạo sự linh hoạt cho người dùng trong việc điều chỉnh hoạt động của thiết bị.



Hình 2.11: Nút Nhấn D5 Cao 5mm 2 Chân Tròn Xuyên Lỗ

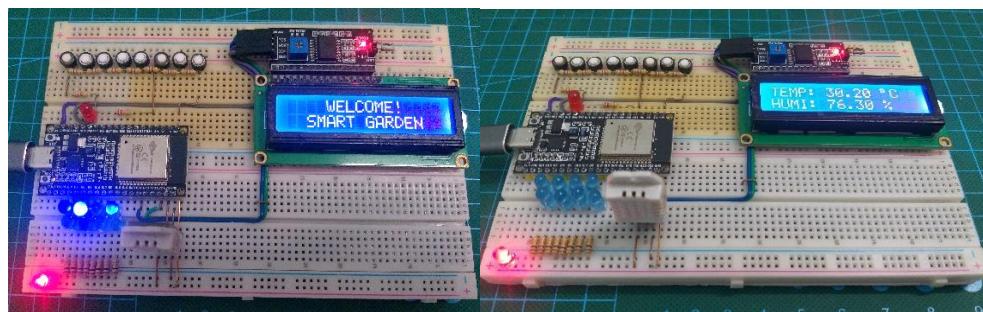
Thông Số Kỹ Thuật:

- Đường kính nút: 5mm
- Chiều cao: 5mm
- Loại: Nút nhấn 2 chân, tròn, xuyên lỗ
- Điện áp chịu tải: 12V
- Dòng điện chịu tải: 0.5A
- Chất liệu: Nhựa, bền, chịu va đập
- Màu sắc: Đen, đỏ (tùy nhà sản xuất)

Sơ đồ chân:

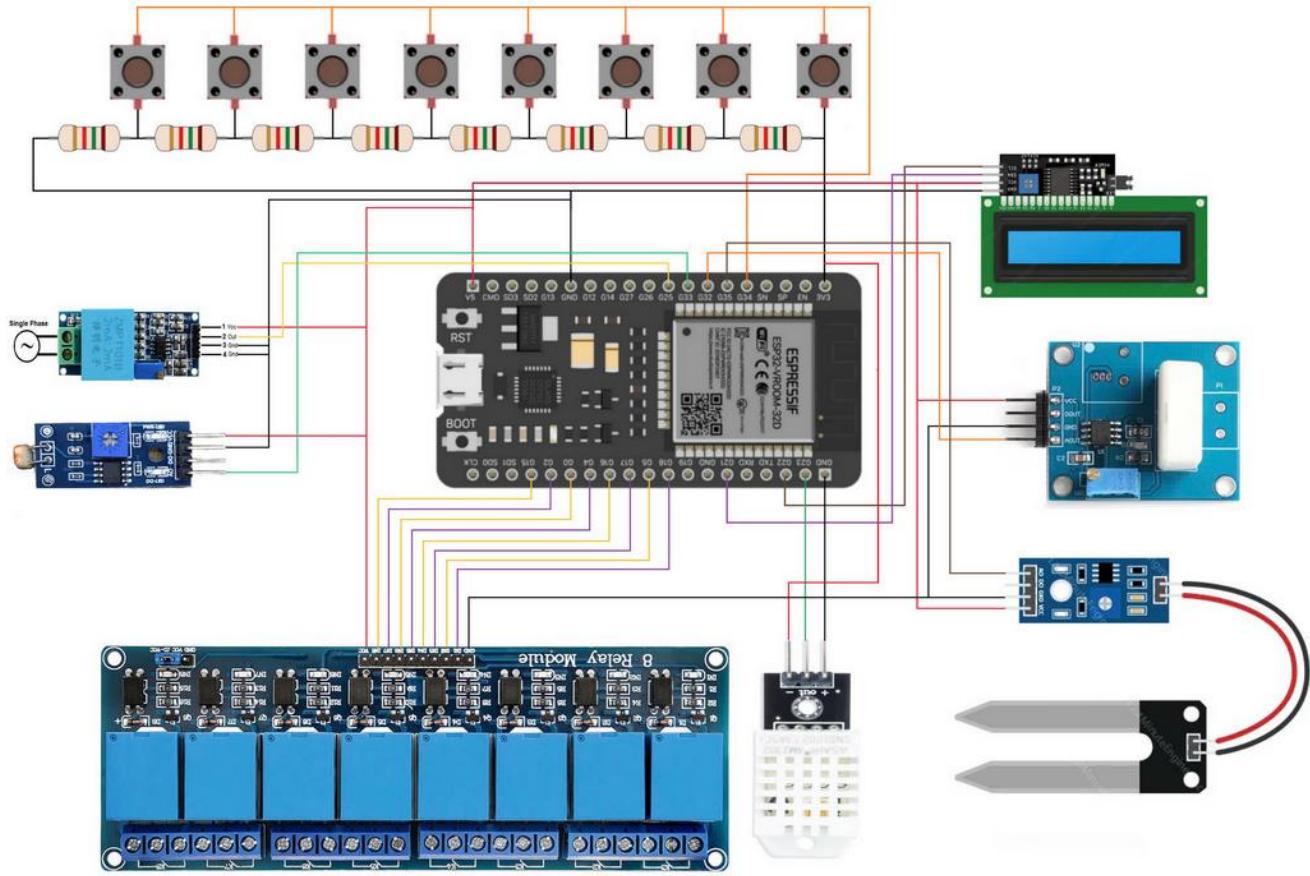
- Chân 1: Nối vào mạch điều khiển hoặc tín hiệu cần kích hoạt.
- Chân 2: Nối vào đất (GND) hoặc mạch điện cần điều khiển.

❖ Phản ứng ngoài thực tế



Hình 2.12: Phản ứng được thử nghiệm trên breadboard

- **Sơ đồ kết nối phản ứng:**



Hình 2.13: Sơ đồ kết nối các phần cứng

3. Chức Năng Của Hệ Thống

- Theo Dõi Môi Trường Thời Gian Thực:** Hệ thống liên tục thu thập và cập nhật các thông số môi trường từ các cảm biến được lắp đặt trong vườn. Người dùng có thể theo dõi dữ liệu này ngay trên trang web, được thể hiện qua các biểu đồ trực quan từ nền tảng Thingspeak. Các biểu đồ này cho phép quan sát biến động nhiệt độ, độ ẩm và mức ánh sáng trong thời gian thực, từ đó giúp người dùng dễ dàng đánh giá và đưa ra các quyết định kịp thời.
- Điều Khiển Từ Xa Các Thiết Bị:** Người dùng có thể điều khiển các thiết bị trong vườn qua trang web, bao gồm bật/tắt các quạt thông gió và đèn, hoặc kích hoạt hệ thống tưới nước. Hệ thống hỗ trợ thiết lập các ngưỡng tự động, nhờ đó các thiết bị sẽ tự động hoạt động dựa trên các điều kiện môi trường. Điều này không chỉ tối ưu hóa việc chăm sóc vườn mà còn giúp tiết kiệm năng lượng hiệu quả.
- Tự Động Hóa Theo Ngưỡng Thiết Lập:** Người dùng có thể thiết lập các giá trị ngưỡng và chế độ tự động trong trang "Settings". Khi các thông số môi trường vượt quá giá trị ngưỡng, hệ thống sẽ tự động bật/tắt các thiết bị để điều chỉnh các điều kiện trở về trạng thái mong muốn.

Chức năng này giúp đảm bảo môi trường phát triển lý tưởng cho cây trồng mà không cần can thiệp thủ công.

4. Thiết Kế Giao Diện và Trải Nghiệm Người Dùng

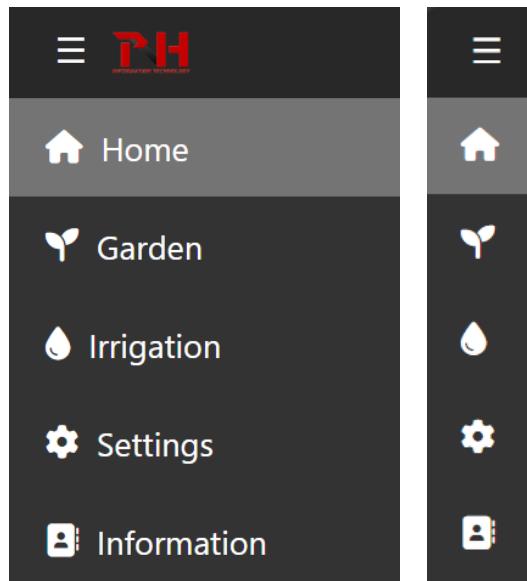
4.1. Bố Cục Dashboard



Hình 4.1.1: Bố cục tổng thể của Dashboard

- Bố cục chính của Dashboard gồm 3 phần:
 - **Header (Phần đầu trang):**
 - Vị trí: Nằm ở trên cùng của trang web và khoanh vùng màu đỏ trong hình 4.1.1
 - Nội dung: Chứa tiêu đề "SMART GARDEN SYSTEM" và logo bên trái.
 - Chức năng: Header giúp nhận diện hệ thống với tiêu đề rõ ràng, cung cấp cảm giác chuyên nghiệp và nhấn mạnh tên hệ thống.
 - **Sidebar (Thanh điều hướng):**
 - Vị trí: Nằm ở phía bên trái của trang web và khoanh vùng màu xanh lá trong hình 4.1.1
 - Nội dung: Menu điều hướng dọc với các tùy chọn:
 - + Home: Quay về trang chủ.
 - + Garden: Hiển thị thông tin và điều khiển cho khu vườn.
 - + Irrigation: Quản lý hệ thống tưới.
 - + Settings: Tùy chỉnh các cài đặt hệ thống.
 - + Information: Thông tin chung về thành viên.

- Chức năng: Thanh điều hướng cho phép người dùng di chuyển dễ dàng giữa các mục. Tính năng thu gọn giúp tiết kiệm không gian, khi mở rộng sẽ hiển thị tên từng mục, còn khi thu gọn sẽ chỉ hiển thị biểu tượng, làm cho giao diện gọn gàng hơn.



Hình 4.1.2: Thanh điều hướng mặc định và khi đã thu gọn

- **Main Content Area (Khu vực nội dung chính):**

- Vị trí: Chiếm phần trung tâm của trang web và khoanh vùng màu xanh dương trong hình 4.1.1
- Nội dung: Đây là khu vực chính hiển thị nội dung chi tiết tương ứng với mục được chọn trong Sidebar.
- Chức năng: Khu vực này thay đổi nội dung linh hoạt theo lựa chọn của người dùng, giúp tối ưu hóa không gian và tập trung vào thông tin quan trọng.

- ❖ **Phong cách thiết kế, màu sắc và cách giao diện hỗ trợ trải nghiệm người dùng:**

- Giao diện sử dụng tone màu tối, tạo cảm giác hiện đại và công nghệ cao. Font chữ dễ đọc, với các tiêu đề rõ ràng. Sidebar có màu nền tối, làm nổi bật các tùy chọn điều hướng bằng biểu tượng và chữ trắng. Phong cách thiết kế trực quan, giúp người dùng nhanh chóng nhận diện các chức năng chính và sử dụng dễ dàng. Việc tổ chức bố cục và màu sắc này hỗ trợ trải nghiệm người dùng bằng cách giảm thiểu sự phân tâm và giúp tập trung vào thông tin hiển thị.

4.2. Tương Tác Người Dùng và Điều Khiển

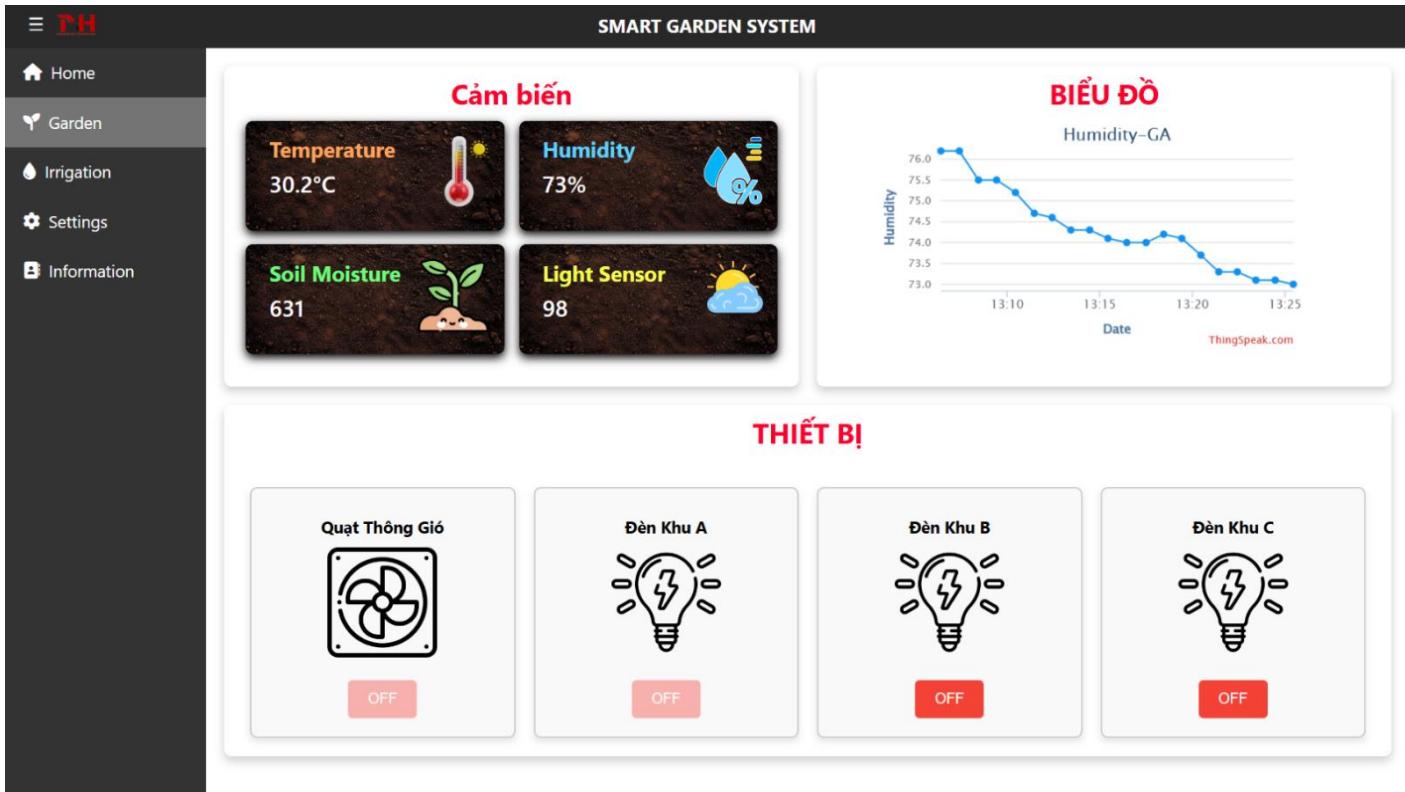
- ❖ **Home**



Hình 4.2.1: Trang giới thiệu đề tài

- Phần Trang chủ hiển thị một màn hình chào mừng người dùng đến với hệ thống giám sát vườn thông minh. Trang này cung cấp thông tin tổng quan về hệ thống và các lợi ích chính, giúp người dùng nhanh chóng hiểu được các chức năng mà hệ thống mang lại.
- Nội dung hiển thị rõ ràng với các phần như "Về Chúng Tôi" cùng mô tả về cách thức hoạt động của hệ thống. Mục đích chính của phần Trang chủ là giới thiệu người dùng vào hệ thống một cách dễ dàng và hấp dẫn.

❖ Garden



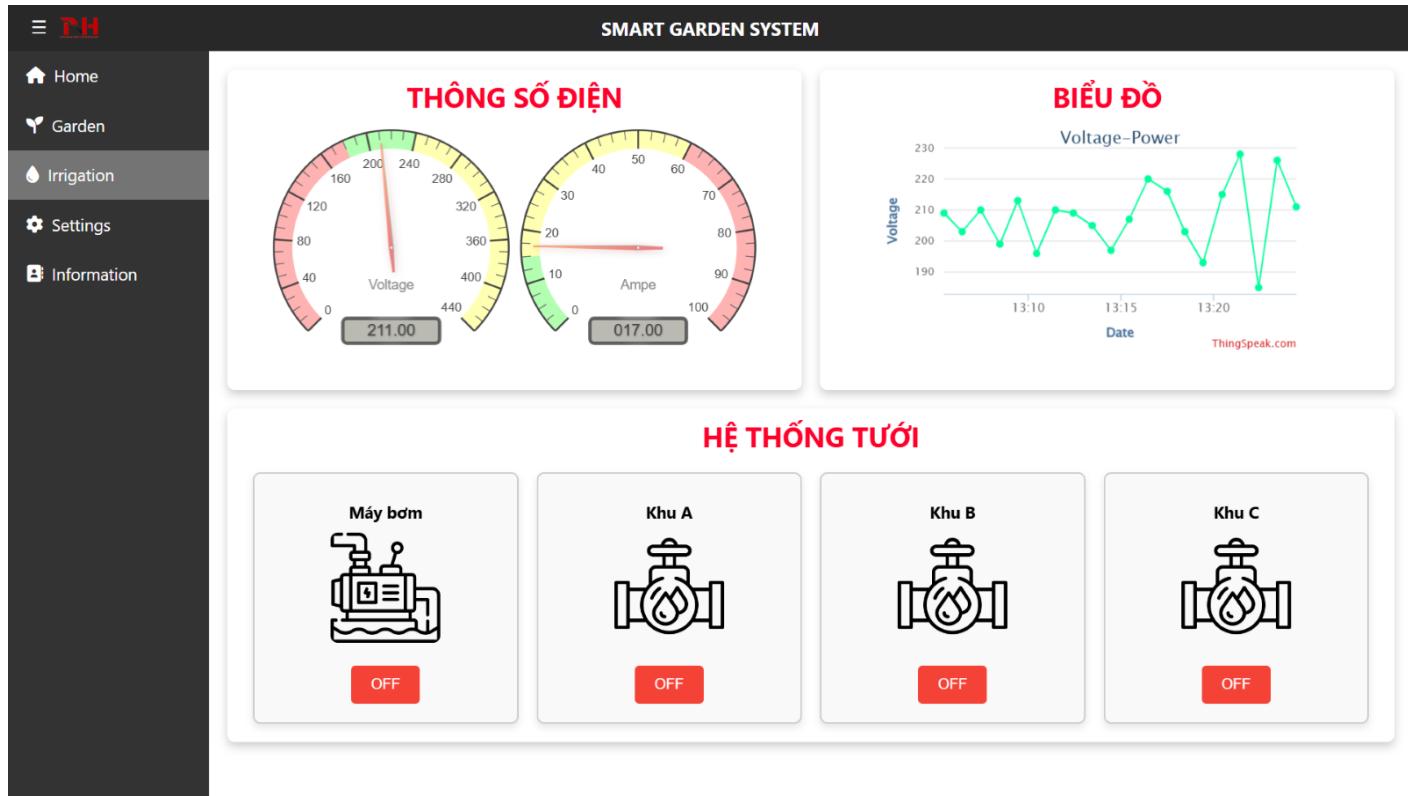
Hình 4.2.2: Trang quản lí khu vườn và thiết bị

- Phần Garden hiển thị dữ liệu từ các cảm biến trong khu vực vườn, bao gồm: nhiệt độ, độ ẩm, độ ẩm đất, và cảm biến ánh sáng. Các cảm biến được hiển thị bằng các khung hình riêng biệt, với biểu tượng minh họa trực quan giúp người dùng dễ dàng nhận biết loại thông số.
- Biểu đồ thời gian thực cho phép người dùng theo dõi sự thay đổi của các thông số môi trường theo thời gian. Dữ liệu từ các cảm biến được cập nhật liên tục để đảm bảo người dùng luôn có được thông tin mới nhất về khu vực Garden. Người dùng có thể thay đổi theo dõi biểu đồ của các thông số khác bằng cách nhấp vào các khung thông số mà người dùng muốn hiển thị.

• Chức năng điều khiển

- Hệ thống cung cấp giao diện người dùng để bật/tắt cho các thiết bị như quạt thông gió, đèn chiếu sáng trong khu vườn. Người dùng có thể kích hoạt chế độ tự động cho các thiết bị này, giúp tự động duy trì các điều kiện môi trường phù hợp cho cây trồng.
- Giao diện hiển thị trực quan giúp người dùng dễ dàng quan sát trạng thái thiết bị: bật/tắt. Khi thiết bị đang bật, màu sắc và trạng thái của nút thay đổi, đồng thời hình ảnh minh họa thiết bị cũng được cập nhật (ví dụ: hình ảnh quạt thông gió sáng khi đang được bật).

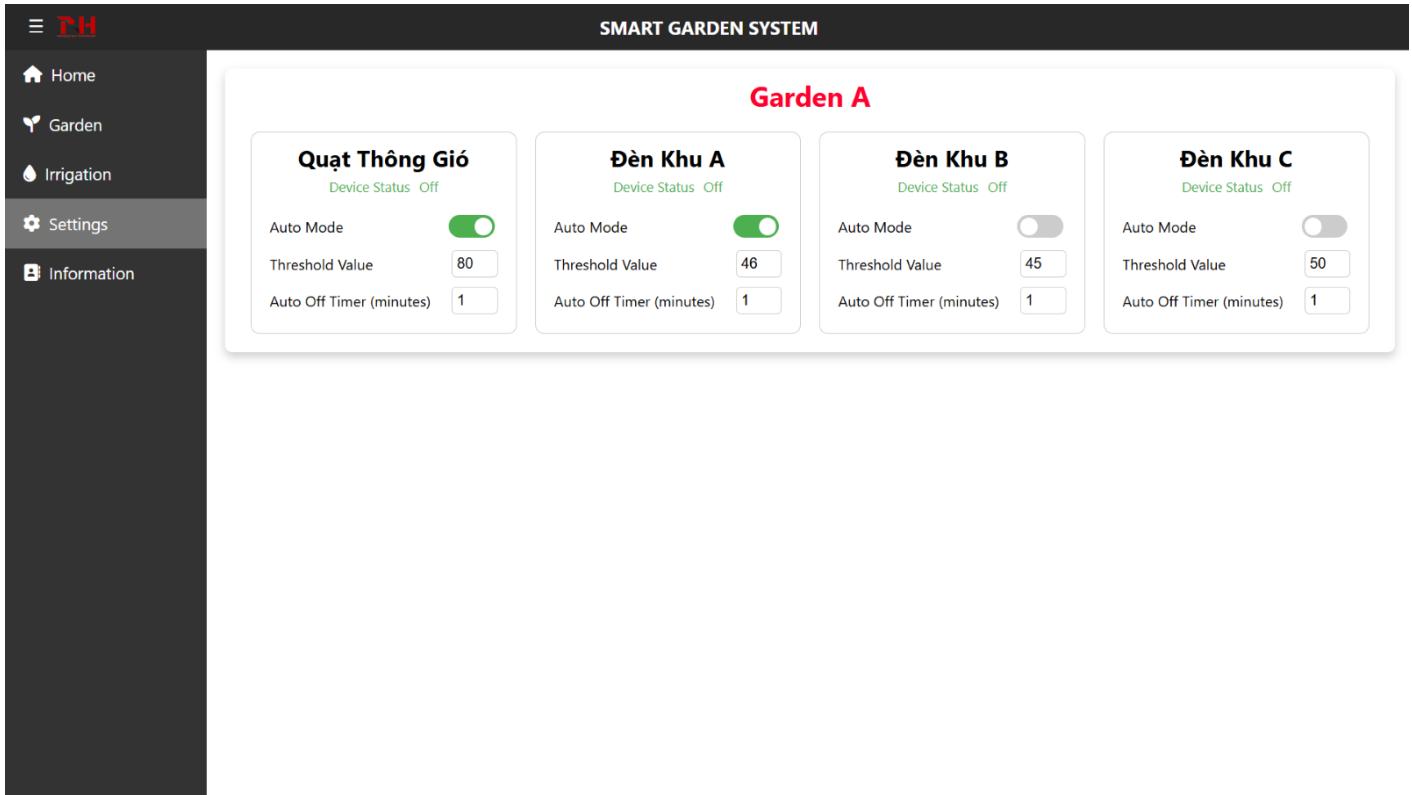
❖ Irrigation



Hình 4.2.2: Trang quản lí hệ thống máy bơm, van nước

- Phần Irrigation hiển thị thông tin về hệ thống tưới nước, bao gồm các thông số điện áp và cường độ dòng điện của máy bơm. Hai đồng hồ đo được sử dụng để hiển thị giá trị điện áp (Voltage) và cường độ dòng điện (Ampe), giúp người dùng dễ dàng theo dõi tình trạng hoạt động của hệ thống tưới.
- Biểu đồ thời gian thực cho phép người dùng theo dõi sự thay đổi của các thông số điện áp (Voltage) và cường độ dòng điện (Ampe) theo thời gian. Dữ liệu từ các cảm biến được cập nhật liên tục để đảm bảo người dùng luôn có được thông tin mới nhất về điện áp và cường độ dòng điện. Người dùng có thể thay đổi theo dõi biểu đồ của điện áp hay cường độ dòng điện bằng cách nhấn vào đồng hồ mà người dùng muốn hiện thị.
- Chức Năng Điều Khiển**
 - Hệ thống cung cấp giao diện người dùng để bật/tắt từng khu vực và các thiết bị máy bơm, vòi tưới, cho phép kiểm soát linh hoạt lượng nước cung cấp cho từng khu vực. Ngoài ra, việc hiện thị thông số điện áp và dòng điện giúp người dùng có thể nhận biết tình trạng của hệ thống và phát triển sự cống kề thời.
 - Các nút bật/tắt cho từng khu vực tưới (Khu A, Khu B, Khu C, và máy bơm) được hiển thị rõ ràng. Khi người dùng nhấn vào một nút, màu sắc và trạng thái của nút sẽ thay đổi để phản ánh tình trạng thực tế của thiết bị.

❖ Settings



Hình 4.2.3: Trang điều khiển chế độ tự động

- Chế độ tự động: Người dùng có thể cấu hình chế độ tự động cho các thiết bị bằng cách thiết lập giá trị ngưỡng cho các cảm biến môi trường. Chế độ tự động sẽ bật/tắt thiết bị dựa trên giá trị ngưỡng và thời gian thiết lập sẵn từ trước.
 - Ví dụ, quạt sẽ tự động bật khi độ ẩm không khí vượt quá một giá trị được chỉ định và tự động thiết lập thời gian. Khi hết thời gian mà giá trị vẫn còn vượt qua giá trị ngưỡng thì sẽ tự động đặt thêm thời gian và thiết bị vẫn tiếp tục bật, khi giá trị đã thỏa giá trị người dùng thì tự động tắt thiết bị.
- Thao tác người dùng có thể thực hiện: Người dùng có thể trực tiếp bật/tắt thiết bị từ giao diện hoặc chuyển đổi chế độ hoạt động của thiết bị sang chế độ tự động. Các thông số như ngưỡng kích hoạt và thời gian tự động tắt thiết bị cũng có thể được cấu hình từ giao diện, giúp hệ thống linh hoạt trong việc đáp ứng các điều kiện môi trường cụ thể.

❖ Information

The screenshot displays the 'Information' section of the Smart Garden System. It features three cards, each representing a member with their profile picture, name, ID number, date of birth, education level, email, and phone number.

- Trương Lý Minh Hoàng**
MSSV: 22139025
Ngày sinh: 05/08/2004
Ngành học: Hệ thống nhúng & IoT
Email: truonglyminhhoang2004@gmail.com
Điện thoại: 0832 525 579
- Phan Nhị Hào**
MSSV: 22139019
Ngày sinh: 29/03/2004
Ngành học: Hệ thống nhúng & IoT
Email: nhihao2903@gmail.com
Điện thoại: 0352 733 708
- Mai Ngọc Hoàng**
MSSV: 22139023
Ngày sinh: 22/01/2004
Ngành học: Hệ thống nhúng & IoT
Email: ngochoang2012004@gmail.com
Điện thoại: 0395 157 692

Thông tin giảng viên

Giảng viên: Trương Quang Phúc
Môn học: TT Kiến trúc và giao thức IoT
Email: phuctq@hcumute.edu.vn
Điện thoại: Oxxxxxxxxx

Hình 4.2.4: Trang giới thiệu thông tin thành viên

- Phần Information cung cấp thông tin về các thành viên tham gia phát triển hệ thống và giảng viên hướng dẫn. Các thông tin này bao gồm tên, mã số sinh viên, ngày sinh, ngành học, email, và số điện thoại.
- Giao diện được thiết kế đơn giản và rõ ràng để người dùng có thể dễ dàng tìm hiểu thêm về những người đã tham gia xây dựng hệ thống.

5. Kết Nối với Firebase và Giao Thức Truyền Dữ Liệu

5.1 Firebase

Hệ thống sử dụng Firebase Realtime Database để quản lý và lưu trữ dữ liệu từ các cảm biến trong thời gian thực. Việc này cho phép hệ thống theo dõi, lưu trữ, và xử lý thông tin môi trường từ các thiết bị và cảm biến khác nhau, sau đó cung cấp dữ liệu trực tiếp lên dashboard để người dùng giám sát và điều khiển hệ thống từ xa. Cấu hình Firebase trong hệ thống bao gồm:

- Cấu hình Firebase:**
 - ESP32 kết nối với Firebase để quản lý dữ liệu từ các cảm biến thông qua đoạn mã cấu hình kết nối với Firebase trong tập tin script.js.

```

//Firebase
const firebaseConfig = {
  apiKey: "AIzaSyAP32IhkHv3b61GgsKF-5WU_7td576j100",
  authDomain: "smart-garden-system-6b468.firebaseio.com",
  databaseURL: "https://smart-garden-system-6b468-default-rtdb.firebaseio.com",
  projectId: "smart-garden-system-6b468",
  storageBucket: "smart-garden-system-6b468.appspot.com",
  messagingSenderId: "60112766198",
  appId: "1:60112766198:web:a050921dfd50ed3d4858d7",
  measurementId: "G-5D8JB02RX0"
};

// Initialize Firebase
firebase.initializeApp(firebaseConfig);
firebase.analytics();
var database = firebase.database();

```

Hình 5.1.1: Các thiết lập cho Firebase trong Dashboard

- Sau khi cấu hình, Firebase sẽ hỗ trợ lưu trữ và cập nhật dữ liệu từ các cảm biến của hệ thống (như nhiệt độ, độ ẩm đất ...) theo thời gian thực.
- **Lưu và Truy Xuất Dữ Liệu trên Firebase:**
- Dữ liệu từ cảm biến được thu thập bởi ESP32 và cập nhật lên Firebase thông qua các hàm như Firebase.set().

```

Firebase.setFloat(firebaseData, "/GardenA/Temperature/", temperature);
Firebase.setFloat(firebaseData, "/GardenA/Humidity/", humidity);
Firebase.setFloat(firebaseData, "/GardenA/Light/", lightValue);
Firebase.setFloat(firebaseData, "/GardenA/Soil/", soilMoistureValue);
Firebase.setFloat(firebaseData, "/Power/Voltage/", Volt);
Firebase.setFloat(firebaseData, "/Power/Ampe/", Ampe);

```

Hình 5.1.2: Các hàm cập nhật giá trị của cảm biến lên Firebase

- Các hàm cập nhật dữ liệu từ Firebase đến giao diện người dùng, cho phép người dùng quan sát trực tiếp trạng thái cảm biến. Các hàm on("value") sẽ tự động cập nhật giá trị từ cơ sở dữ liệu Firebase và hiển thị theo thời gian thực trên dashboard.

5.2 Giao Thức Kết Nối:

- Hệ thống sử dụng các giao thức như HTTP để duy trì kết nối liên tục và đảm bảo các thiết bị hoạt động hiệu quả.
- Thư viện FirebaseESP32.h sẽ sử dụng giao thức HTTP hoặc HTTP/2 ngầm trong nền để thực hiện các thao tác với cơ sở dữ liệu và không yêu cầu lập trình viên trực tiếp khai báo giao thức HTTP trong mã nguồn khi thao tác với cơ sở dữ liệu.

5.3 Truyền Dữ Liệu giữa Các Thành Phần

Quá trình truyền dữ liệu giữa các thành phần trong hệ thống được thực hiện dựa trên nguyên tắc đồng bộ và theo dõi thời gian thực. Cảm biến sẽ gửi dữ liệu lên Firebase, sau đó dữ liệu này được xử lý và hiển thị trên dashboard.

- **Hiển Thị Dữ Liệu Cảm Biến trên Dashboard:**

- Các chỉ số như nhiệt độ, độ ẩm đất, và ánh sáng được hiển thị dưới dạng biểu đồ và khung thông tin trên giao diện người dùng. Việc cập nhật này giúp người dùng dễ dàng theo dõi tình trạng môi trường và tình hình khu vườn.

```
// Auto load Temperature
function setupTemperature(ValueId, imgId, firebasePath) {
    database.ref(firebasePath).child('Temperature').on("value", function (snapshot) {
        var Temperature = snapshot.val();
        document.getElementById(ValueId).innerHTML = Temperature;
        if (Temperature < 20) {
            document.getElementById(imgId).src = 'img/temperature-low.png';
        } else if (Temperature < 30) {
            document.getElementById(imgId).src = 'img/temperature-normal.png';
        } else {
            document.getElementById(imgId).src = 'img/temperature-high.png';
        }
    });
}
```

Hình 5.3.1: Funktion cập nhật giá trị và icon cho các khung thông tin

- **Điều Khiển Từ Xa Thiết Bị qua Firebase:**

- Hệ thống cho phép điều khiển từ xa các thiết bị như quạt thông gió, đèn, và máy bơm. Người dùng có thể thao tác bật/tắt từng thiết bị từ dashboard và theo dõi trạng thái trên giao diện thông qua Firebase:

```

// Hàm xử lý sự kiện click cho nút thiết bị
function setupButton(buttonId, imageOnId, imageOffId, firebasePath, stateRef) {
    const button = document.getElementById(buttonId);
    const imageOn = document.getElementById(imageOnId);
    const imageOff = document.getElementById(imageOffId);

    // Lắng nghe trạng thái nút
    firebase.database().ref(firebasePath).child(stateRef).on('value', function (snapshot) {
        const state = snapshot.val();
        updateDeviceState(button, imageOn, imageOff, state);
    });

    // Thêm sự kiện click cho nút
    button.addEventListener('click', function () {
        firebase.database().ref(firebasePath).child(stateRef).once('value').then(function (snapshot) {
            const currentState = snapshot.val();
            const newState = (currentState === 0) ? 1 : 0; // Chuyển đổi trạng thái
            firebase.database().ref(firebasePath).child(stateRef).set(newState);
        });
    });
}

```

Hình 5.3.3: Function cập nhật trạng thái của thiết bị

6. Xây dựng giao diện Mobile sử dụng Blynk IoT.

6.1. Giới thiệu về Blynk

Blynk là một nền tảng IoT (Internet of Things) phổ biến, giúp người dùng dễ dàng tạo ra ứng dụng điều khiển và giám sát thiết bị thông qua giao diện di động mà không cần kiến thức lập trình phức tạp.

Đặc điểm nổi bật của Blynk

- **Hỗ trợ đa dạng phần cứng:** Tương thích với nhiều loại vi điều khiển như Arduino, ESP8266, ESP32, Raspberry Pi, và các thiết bị khác.
- **Widget trực quan và dễ sử dụng:** Cung cấp các công cụ như nút nhấn, thanh trượt, đồ thị, và nhiều widget khác để xây dựng giao diện điều khiển một cách nhanh chóng.
- **Kết nối linh hoạt:** Hỗ trợ kết nối qua Wi-Fi, Ethernet, Bluetooth, và GSM/GPRS, giúp linh hoạt trong nhiều ứng dụng IoT.
- **Quản lý dự án từ xa:** Người dùng có thể điều khiển, giám sát thiết bị từ bất cứ đâu thông qua ứng dụng di động hoặc nền tảng web.
- **Tích hợp dễ dàng với mã nguồn:** Blynk cung cấp **thư viện Blynk** giúp lập trình giao tiếp giữa thiết bị và ứng dụng một cách đơn giản.

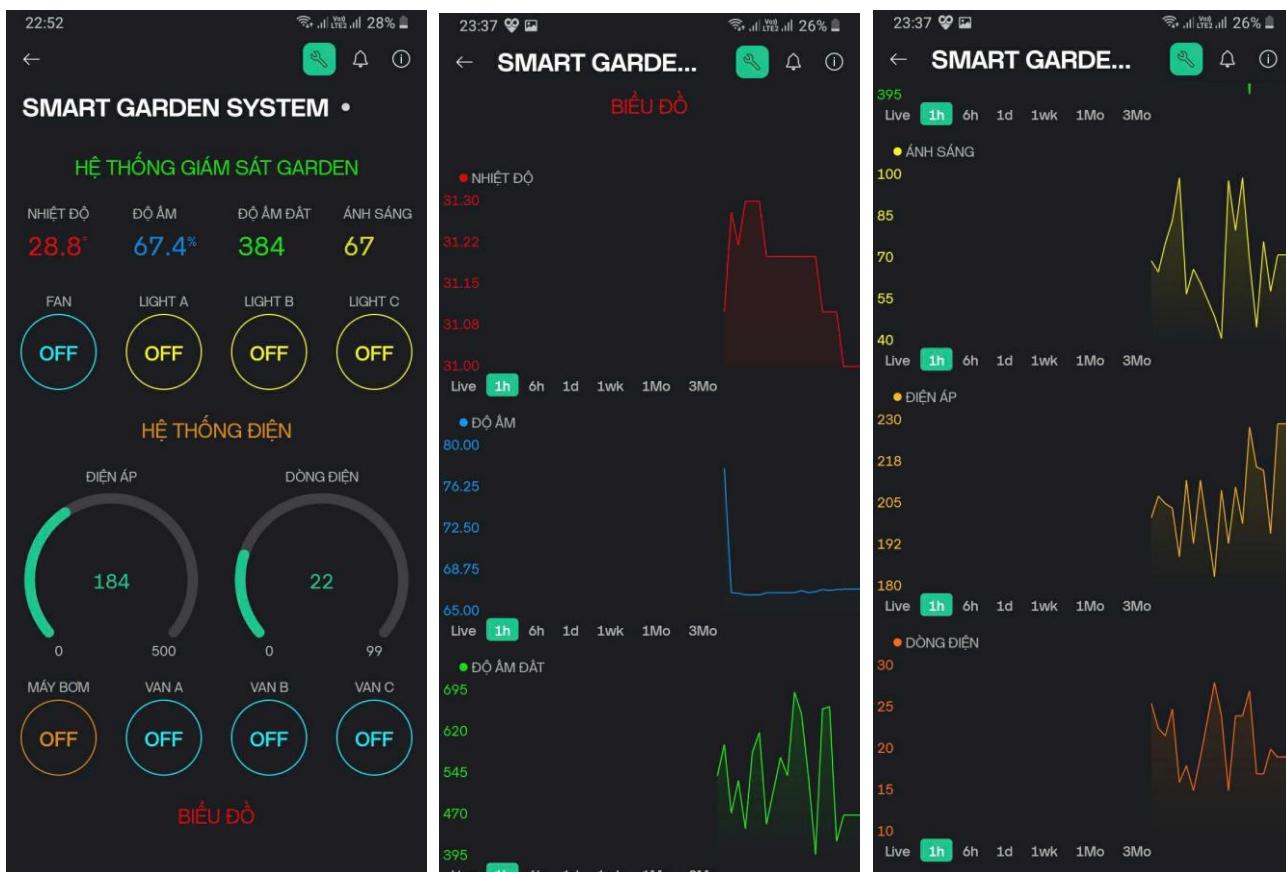
6.2. Thiết kế và triển khai

6.2.1. Giao diện Blynk

Giao diện của hệ thống "SMART GARDEN SYSTEM" được thiết kế trên ứng dụng Blynk nhằm mang lại trải nghiệm trực quan và dễ sử dụng cho người dùng. Ứng dụng Blynk cho phép người dùng tạo ra các bảng điều khiển tùy chỉnh để giám sát và điều khiển các thiết bị IoT từ xa.

Các bước thiết kế giao diện trên Blynk:

- Tạo tài khoản Blynk:** Người dùng cần tạo một tài khoản trên ứng dụng Blynk để bắt đầu thiết kế giao diện.
- Tạo dự án mới:** Tạo một dự án mới và chọn loại phần cứng sử dụng (như ESP32, ESP8266,...).
- Thêm và cấu hình widget:** Thêm các widget cần thiết vào giao diện như nút điều khiển, biểu đồ, và các hiển thị số liệu. Cấu hình các widget để liên kết chúng với các chân và biến trên vi điều khiển.
- Lập trình và kết nối:** Viết mã trên Arduino IDE để nhận và gửi dữ liệu từ cảm biến và thiết bị điều khiển đến ứng dụng Blynk. Kết nối phần cứng với mạng Wi-Fi hoặc Bluetooth theo cấu hình đã thiết lập.



Hình 6.2.1: Giao diện App của hệ thống

- Bộ cục chính của App gồm 3 phần: Hệ thống giám sát, hệ thống điện và biểu đồ.

• **Hệ thống giám sát**

Phần này của ứng dụng cung cấp một cái nhìn tổng quan về các thông số môi trường trong vườn, giúp người dùng dễ dàng theo dõi và quản lý.

Thông số môi trường:

Nhiệt độ: Hiển thị nhiệt độ hiện tại trong vườn (ví dụ: 28.8°C).

Độ ẩm không khí: Hiển thị độ ẩm hiện tại (ví dụ: 67.4%).

Độ ẩm đất: Hiển thị mức độ ẩm của đất (ví dụ: 384).

Ánh sáng: Hiển thị mức độ ánh sáng hiện tại (ví dụ: 67).

Nút điều khiển thiết bị:

Quạt (Fan): Nút bật/tắt quạt để điều chỉnh nhiệt độ.

Đèn chiếu sáng (Light A, Light B, Light C): Nút bật/tắt đèn chiếu sáng trong vườn.

• **Hệ thống điện**

Phần này giúp người dùng dễ dàng kiểm soát và điều khiển các thiết bị điện trong vườn, đảm bảo sự hoạt động ổn định và hiệu quả của hệ thống.

Thông số điện năng:

Điện áp (Voltage): Hiển thị điện áp hiện tại của hệ thống (ví dụ: 184V).

Dòng điện (Current): Hiển thị dòng điện hiện tại (ví dụ: 22A).

Nút điều khiển thiết bị:

Máy bơm nước và van (Pump, Valve A, Valve B, Valve C): Nút điều khiển hệ thống tưới nước, bao gồm bật/tắt máy bơm và các van nước.

• **Biểu đồ**

Phần này cung cấp các biểu đồ theo dõi thời gian thực, giúp người dùng có cái nhìn chi tiết và toàn diện về các thông số môi trường và điện năng trong các khoảng thời gian từ 1 giờ đến 3 tháng.

6.2.2. Truyền Dữ Liệu giữa Các Thành Phần

Trong hệ thống "SMART GARDEN SYSTEM", module ESP32 đóng vai trò quan trọng trong việc truyền dữ liệu giữa các cảm biến, vi điều khiển và ứng dụng Blynk. Việc giao tiếp giữa ESP và Blynk được thực hiện thông qua các bước sau:

Kết nối ESP với mạng Wi-Fi: Trước tiên, module ESP cần được kết nối với mạng Wi-Fi để có thể giao tiếp với máy chủ Blynk. Cấu hình Wi-Fi được thực hiện trong mã nguồn của vi điều khiển, với các thông tin như SSID và mật khẩu của mạng.

Gửi dữ liệu từ ESP đến Blynk: Sau khi kết nối thành công với mạng Wi-Fi và máy chủ Blynk, ESP sẽ thu thập dữ liệu từ các cảm biến và gửi dữ liệu này lên Blynk. Dữ liệu có thể là nhiệt độ, độ ẩm, độ ẩm đất, hoặc các thông số khác và trạng thái của các thiết bị.

```
Blynk.virtualWrite(V8, temperature);
Blynk.virtualWrite(V9, humidity);
Blynk.virtualWrite(V10, soilMoistureValue);
Blynk.virtualWrite(V11, lightValue);
Blynk.virtualWrite(V12, Volt);
Blynk.virtualWrite(V13, Ampe);
// Cập nhật trạng thái thiết bị lên Blynk
Blynk.virtualWrite(V0 + i, DeviceState[i]);
```

Hình 6.2.2: Các hàm cập nhật giá trị và trạng thái từ Esp lên Blynk

Nhận lệnh điều khiển từ Blynk: Người dùng có thể gửi lệnh điều khiển từ ứng dụng Blynk để điều chỉnh hoạt động của các thiết bị trong hệ thống (như bật/tắt quạt, đèn, hoặc hệ thống tưới nước). ESP sẽ nhận các lệnh này và thực hiện các tác vụ tương ứng.

```
// Các hàm BLYNK_WRITE cho từng Virtual Pin để nhận trạng thái từ Blynk
BLYNK_WRITE(V0) { updateDeviceState(0, param.asInt()); }
BLYNK_WRITE(V1) { updateDeviceState(1, param.asInt()); }
BLYNK_WRITE(V2) { updateDeviceState(2, param.asInt()); }
BLYNK_WRITE(V3) { updateDeviceState(3, param.asInt()); }
BLYNK_WRITE(V4) { updateDeviceState(4, param.asInt()); }
BLYNK_WRITE(V5) { updateDeviceState(5, param.asInt()); }
BLYNK_WRITE(V6) { updateDeviceState(6, param.asInt()); }
BLYNK_WRITE(V7) { updateDeviceState(7, param.asInt()); }

// Hàm cập nhật trạng thái thiết bị và lưu lên Firebase
void updateDeviceState(int deviceIndex, int state) {
    DeviceState[deviceIndex] = state;
    digitalWrite(DevicePins[deviceIndex], state ? HIGH : LOW);
    FirebaseDatabase.setInt(firebaseData, devicePaths[deviceIndex], state);
}
```

Hình 6.2.3: Các hàm cập nhật trạng thái từ Blynk về ESP và lưu lên Firebase

Việc giao tiếp hiệu quả giữa ESP và Blynk giúp hệ thống vườn thông minh có thể hoạt động đồng bộ, cho phép người dùng giám sát và điều khiển các thiết bị từ xa một cách dễ dàng và tin cậy.

7. Đánh Giá Hiệu Quả Hệ Thống

7.1. Ưu Điểm và Độ Ổn Định:

- ESP32:** ESP32 là vi điều khiển mạnh mẽ với khả năng kết nối Wi-Fi tích hợp, giúp hệ thống dễ dàng thu thập dữ liệu từ cảm biến và truyền tải lên Firebase. Với khả năng hỗ trợ nhiều châ

I/O, ESP32 đảm bảo xử lý các tác vụ như đọc dữ liệu cảm biến, điều khiển thiết bị, và cập nhật giao diện hiển thị. Tính ổn định của ESP32 giúp hệ thống duy trì hoạt động liên tục, đáng tin cậy.

- **Xử lý đa luồng trên ESP32:** Khả năng xử lý đa luồng của ESP32 cho phép nó thực hiện nhiều tác vụ đồng thời một cách hiệu quả. Điều này đặc biệt hữu ích trong các hệ thống giám sát và điều khiển như SMART GARDEN SYSTEM, nơi mà việc thu thập dữ liệu từ nhiều cảm biến và điều khiển nhiều thiết bị cần được xử lý song song, đảm bảo hệ thống hoạt động mượt mà và nhanh chóng.
- **Màn hình LCD 16x2 (I2C):** Màn hình LCD 16x2 qua giao thức I2C tiết kiệm chân kết nối và cung cấp một phương thức hiển thị trực quan ngay tại vị trí hệ thống, giúp người dùng dễ dàng theo dõi thông số môi trường như nhiệt độ và độ ẩm mà không cần truy cập vào giao diện trực tuyến. Đây là một lợi thế trong việc giám sát nhanh tình trạng hệ thống.
- **Tính năng tự động bật/tắt theo ngưỡng:** Hệ thống tự động điều chỉnh các thiết bị như quạt, đèn, và hệ thống tưới nước dựa trên các giá trị ngưỡng đã cài đặt trước cho các chỉ số môi trường (như nhiệt độ, độ ẩm). Điều này giúp hệ thống phản ứng linh hoạt mà không cần can thiệp thủ công, đảm bảo môi trường phát triển tối ưu cho cây trồng và tiết kiệm năng lượng. Đây là một tính năng quan trọng giúp hệ thống duy trì độ ổn định và phản ứng kịp thời trong các điều kiện môi trường thay đổi.
- **Firebase:** Firebase cung cấp giải pháp lưu trữ và truy xuất dữ liệu theo thời gian thực, giúp đồng bộ dữ liệu giữa các cảm biến và thiết bị điều khiển với dashboard. Điều này đảm bảo dữ liệu luôn được cập nhật mới nhất và người dùng có thể theo dõi trạng thái thiết bị và các chỉ số môi trường từ xa với độ tin cậy cao.
- **Độ ổn định và sự tin cậy:** Hệ thống tổng thể có độ ổn định tốt, phù hợp với các ứng dụng trong nông nghiệp hoặc giám sát môi trường nhờ khả năng phản hồi nhanh từ các thành phần phản ứng và kết nối Wi-Fi ổn định. Tính tự động hóa cao cùng khả năng giám sát trực quan giúp giảm thiểu sự can thiệp thủ công, mang lại hiệu quả quản lý cao.

7.2. Hạn Chế và Đề Xuất Cải Thiện:

- **Độ trễ trong phản hồi từ Firebase:** Firebase có thể gặp độ trễ trong việc cập nhật dữ liệu khi mạng chậm hoặc bị gián đoạn, làm ảnh hưởng đến thời gian phản hồi của các thiết bị trong hệ thống. Đề cải thiện, có thể cân nhắc sử dụng các giao thức truyền tải nhanh hơn, như MQTT, để giảm thiểu độ trễ và tăng tốc độ đồng bộ dữ liệu.

- **Chế độ tự động phụ thuộc vào Dashboard:** Một hạn chế là chế độ điều khiển tự động hiện tại chỉ hoạt động khi Dashboard đang hoạt động. Điều này có nghĩa là nếu Dashboard tắt hoặc ngắt kết nối, chế độ tự động sẽ không được kích hoạt. Để cải thiện, có thể tích hợp thêm chế độ tự động hoạt động trực tiếp trên ESP32, không phụ thuộc vào Dashboard, đảm bảo hệ thống vẫn vận hành tự động ngay cả khi không có kết nối đến giao diện.
- **Bảo mật dữ liệu:** Firebase cung cấp nhiều tính năng bảo mật, nhưng khi hệ thống được triển khai ở môi trường mở, việc nâng cấp cấu hình bảo mật như mã hóa dữ liệu hoặc xác thực người dùng qua OAuth có thể tăng tính an toàn cho dữ liệu truyền qua mạng, nhất là khi hệ thống có quy mô lớn hơn.
- **Hiển thị dữ liệu và giao diện người dùng:** Màn hình LCD 16x2 hiển thị thông tin giới hạn, có thể thay thế bằng màn hình OLED hoặc TFT để hiển thị nhiều thông tin hơn, đồng thời giúp người dùng dễ dàng quan sát trạng thái toàn bộ hệ thống. Điều này sẽ đặc biệt hữu ích khi hệ thống phát triển với nhiều loại thiết bị và thông số cần giám sát hơn.

8. Kết Luận và Hướng Phát Triển

Hệ thống IoT "Giám sát Vườn Thông minh" mang lại nhiều lợi ích rõ rệt trong việc quản lý và giám sát môi trường trồng trọt. Bằng cách thu thập và hiển thị thông tin thời gian thực về các yếu tố môi trường như nhiệt độ, độ ẩm, độ ẩm đất và ánh sáng, hệ thống giúp người dùng dễ dàng theo dõi và điều chỉnh các điều kiện để tối ưu hóa cho sự phát triển của cây trồng. Tính năng điều khiển từ xa và tự động hóa giúp tiết kiệm tài nguyên và thời gian, giảm thiểu công sức lao động, và mang lại hiệu quả cao trong việc chăm sóc vườn. Đặc biệt, với sự kết hợp của các thành phần như ESP32, LCD 16x2 và Firebase, hệ thống duy trì độ ổn định cao và đáp ứng nhanh, đáp ứng tốt nhu cầu của người dùng trong giám sát và quản lý từ xa.

Đề Xuất Phát Triển Thêm

- **Tích hợp thêm cảm biến mới:** Để mở rộng khả năng giám sát, hệ thống có thể tích hợp thêm các cảm biến như cảm biến pH đất để đo độ axit/kiềm của đất và cảm biến độ dẫn điện để đánh giá chất lượng đất. Những cảm biến này sẽ cung cấp thông tin chi tiết hơn về môi trường đất, giúp người dùng đưa ra các quyết định phù hợp trong việc bón phân và tưới tiêu.
- **Cảnh báo thông minh qua ứng dụng di động:** Triển khai hệ thống cảnh báo thông minh qua ứng dụng di động giúp người dùng nhận thông báo kịp thời khi các thông số môi trường vượt ngưỡng cho phép hoặc khi thiết bị gặp sự cố. Điều này sẽ tăng khả năng giám sát và phản ứng tức thì, đặc biệt hữu ích trong trường hợp khẩn cấp.

- **Tính năng tự động bật/tắt thiết bị:** Tích hợp thêm tính năng tự động bật/tắt thiết bị trực tiếp trên ESP32, không phụ thuộc vào Dashboard, giúp thiết bị tự động phản hồi theo các giá trị ngưỡng của cảm biến ngay cả khi không có kết nối mạng hoặc khi Dashboard không hoạt động. Tính năng này sẽ tăng cường tính linh hoạt và tính ổn định của hệ thống, đảm bảo các thiết bị trong vườn luôn hoạt động theo các điều kiện cài đặt sẵn, giúp tối ưu hóa quy trình chăm sóc vườn ngay cả trong tình huống kết nối gặp sự cố.

TRIỂN KHAI HỆ THỐNG SMART GARDEN SYSTEM TRÊN ADAFRUIT IO

1. Giới thiệu về Adafruit IO

Adafruit IO là một nền tảng đám mây được phát triển bởi Adafruit, giúp người dùng dễ dàng lưu trữ, truy xuất và tương tác với dữ liệu từ các thiết bị IoT (Internet of Things). Nền tảng này hỗ trợ nhiều loại phần cứng khác nhau, bao gồm Arduino, ESP8266, ESP32, Raspberry Pi và nhiều loại khác.

Một số tính năng nổi bật của Adafruit IO:

Lưu trữ dữ liệu: Adafruit IO cho phép người dùng lưu trữ dữ liệu từ các cảm biến và thiết bị IoT một cách an toàn và tin cậy. Dữ liệu được lưu trữ trong thời gian thực, giúp người dùng có thể truy cập và theo dõi mọi lúc, mọi nơi.

Giao diện người dùng: Nền tảng cung cấp một giao diện người dùng trực quan và dễ sử dụng, cho phép tạo ra các bảng điều khiển tùy chỉnh để hiển thị dữ liệu từ các thiết bị. Người dùng có thể dễ dàng kéo và thả các widget để hiển thị biểu đồ, số liệu, và nhiều thông tin khác.

Hỗ trợ giao thức MQTT: Adafruit IO sử dụng giao thức MQTT (Message Queuing Telemetry Transport), một giao thức truyền thông nhẹ và hiệu quả dành cho các thiết bị IoT. MQTT giúp truyền dữ liệu nhanh chóng và ổn định giữa các thiết bị và nền tảng đám mây.

Tích hợp API và thư viện: Adafruit IO cung cấp API và thư viện mạnh mẽ, giúp người dùng dễ dàng tích hợp và phát triển các ứng dụng IoT. Thư viện của Adafruit IO hỗ trợ nhiều ngôn ngữ lập trình và phần cứng khác nhau.

Bảo mật: Nền tảng này cung cấp các tính năng bảo mật như mã hóa dữ liệu và xác thực người dùng, đảm bảo rằng dữ liệu của bạn luôn được bảo vệ.

2. Giao thức MQTT

MQTT (Message Queuing Telemetry Transport) là một giao thức truyền thông nhẹ, dựa trên nguyên tắc xuất bản/đăng ký (publish/subscribe), được thiết kế đặc biệt cho các kết nối từ xa, nơi băng thông hạn chế hoặc yêu cầu tiết kiệm năng lượng. MQTT rất phù hợp cho các ứng dụng Internet of Things (IoT) với các đặc điểm nổi bật sau:

Cấu trúc nhẹ: MQTT sử dụng rất ít tài nguyên, lý tưởng cho các thiết bị có bộ nhớ hạn chế và kết nối không ổn định.

Hiệu quả: Giao thức này tối ưu hóa việc truyền tải dữ liệu, giúp giảm thiểu băng thông và tiết kiệm năng lượng.

Linh hoạt: Cho phép kết nối giữa nhiều loại thiết bị khác nhau và hỗ trợ nhiều ngôn ngữ lập trình.

Đáng tin cậy: Hỗ trợ các mức chất lượng dịch vụ (QoS) khác nhau để đảm bảo dữ liệu được truyền tải đúng và đầy đủ.

❖ Cách MQTT được tích hợp vào hệ thống

MQTT được tích hợp vào hệ thống SMART GARDEN SYSTEM để truyền dữ liệu giữa các cảm biến, thiết bị điều khiển và ứng dụng Adafruit IO. Dưới đây là các bước chính:

Kết nối đến broker MQTT: Adafruit IO đóng vai trò là broker MQTT, nơi các thiết bị có thể xuất bản và đăng ký các thông điệp.

```
#define IO_USERNAME "nhihao2903"
#define IO_KEY "aio_LbU194FMJKK3KLxv20z6127ZZdE2"
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, ssid, password);
```

Chủ đề (topics) và truyền dữ liệu: Các thiết bị sẽ xuất bản và đăng ký thông điệp qua các chủ đề cụ thể. Ví dụ, cảm biến nhiệt độ có thể xuất bản dữ liệu lên chủ đề "temperature", trong khi ứng dụng Adafruit IO có thể đăng ký chủ đề này để nhận dữ liệu.

```
AdafruitIO_Feed *BTN_FAN = io.feed("btn-fan");
AdafruitIO_Feed *BTN_LIGHTA = io.feed("btn-lighta");
AdafruitIO_Feed *BTN_LIGHTB = io.feed("btn-lightb");
AdafruitIO_Feed *BTN_LIGHTC = io.feed("btn-lightc");
AdafruitIO_Feed *BTN_MAYBOM = io.feed("btn-maybom");
AdafruitIO_Feed *BTN_VANA = io.feed("btn-vana");
AdafruitIO_Feed *BTN_VANB = io.feed("btn-vanb");
AdafruitIO_Feed *BTN_VANC = io.feed("btn-vanc");
AdafruitIO_Feed *HumidityFeed = io.feed("humidity");
AdafruitIO_Feed *TemperatureFeed = io.feed("temperature");
```

Xuất bản và đăng ký thông điệp:

- **Xuất bản (publish):** Thiết bị gửi dữ liệu cảm biến lên Adafruit IO.

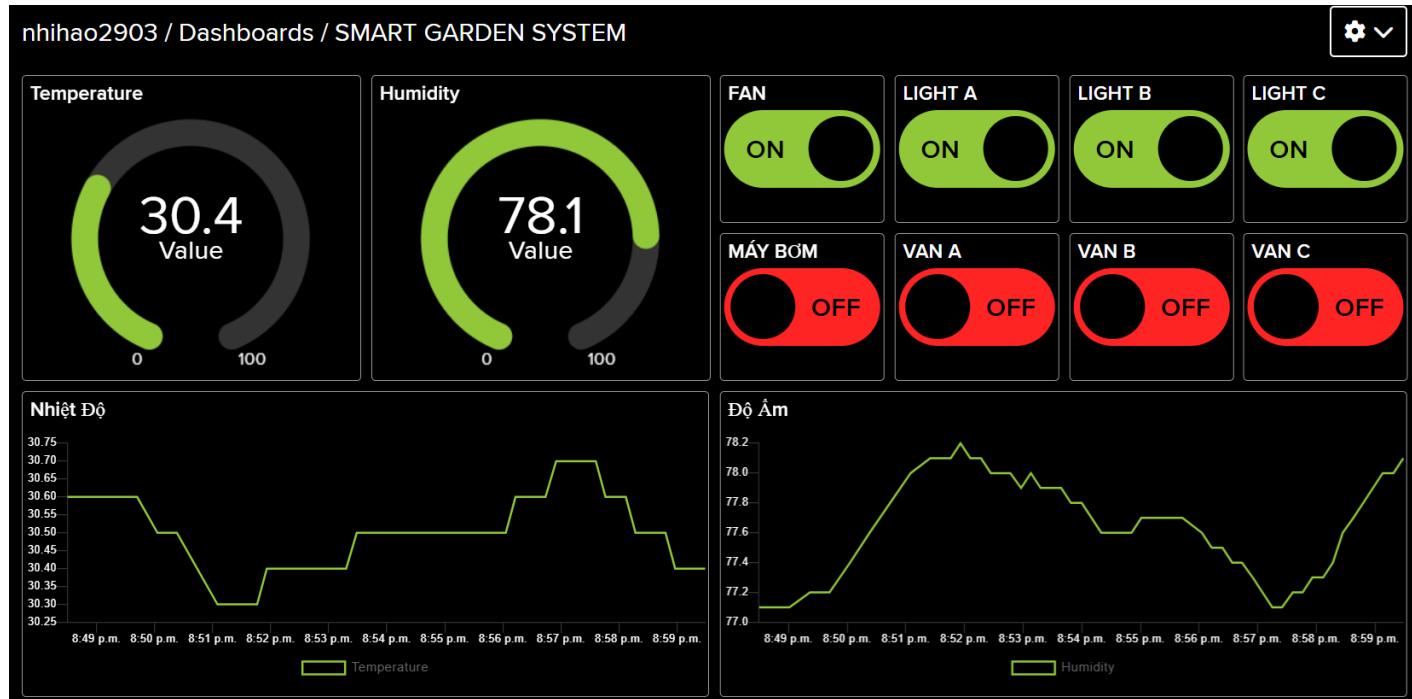
```
TemperatureFeed->save(temperature);
HumidityFeed->save(humidity);
```

- **Đăng ký (subscribe):** Thiết bị nhận lệnh điều khiển từ Adafruit IO.

```
BTN_FAN->onMessage(handleFanMessage);
BTN_LIGHTA->onMessage(handleLightAMessage);
BTN_LIGHTB->onMessage(handleLightBMessage);
BTN_LIGHTC->onMessage(handleLightCMessage);
BTN_MAYBOM->onMessage(handleMayBomMessage);
BTN_VANA->onMessage(handleVanAMessage);
BTN_VANB->onMessage(handleVanBMessage);
BTN_VANC->onMessage(handleVanCMessage);
```

Việc tích hợp MQTT vào hệ thống giúp đảm bảo dữ liệu từ các cảm biến được truyền tải một cách hiệu quả và tin cậy, đồng thời cho phép người dùng giám sát và điều khiển hệ thống từ xa thông qua Adafruit IO.

3. Thiết kế giao diện người dùng



Giao diện trên Adafruit IO cung cấp một cái nhìn tổng quan rõ ràng và trực quan về hệ thống giám sát và điều khiển, đặc biệt là trong lĩnh vực nông nghiệp hoặc nhà thông minh. Bộ cục được thiết kế khoa học, tập trung vào các thông số quan trọng và các nút điều khiển cần thiết, giúp người dùng dễ dàng theo dõi và tương tác với hệ thống.

❖ **Bộ cục giao diện trên Adafruit IO được chia thành các khu vực chính:**

- Khu vực hiển thị thông số môi trường: hệ thống giám sát.**

Nhiệt độ và độ ẩm: Hiển thị giá trị hiện tại của nhiệt độ và độ ẩm dưới dạng đồ thị tròn. Điều này giúp người dùng nhanh chóng nắm bắt được tình trạng môi trường hiện tại.

- Khu vực điều khiển thiết bị: hệ thống điện.**

Quạt, đèn, máy bơm, van: Các nút bật/tắt cho phép người dùng điều khiển trực tiếp các thiết bị trong hệ thống vườn. Điều này rất tiện lợi khi cần điều chỉnh các thiết bị theo nhu cầu.

- Khu vực theo dõi biến động các thông số:**

Biểu đồ theo dõi thời gian thực: Hiển thị biểu đồ đường biểu diễn sự thay đổi của nhiệt độ và độ ẩm theo thời gian. Điều này giúp người dùng theo dõi xu hướng thay đổi của các thông số này và phát hiện các bất thường.

4. Đánh giá hiệu quả hệ thống

❖ Ưu điểm và độ ổn định của hệ thống

- **Ưu điểm** của hệ thống: giao diện trực quan, dễ sử dụng, giúp người dùng nhanh chóng nắm bắt được tình trạng của hệ thống. Các thông số môi trường như nhiệt độ, độ ẩm được hiển thị rõ ràng và trực quan qua biểu đồ, cho phép người dùng theo dõi sự thay đổi của các yếu tố này theo thời gian. Hệ thống cũng cung cấp các công cụ điều khiển thiết bị trực quan, giúp người dùng dễ dàng điều chỉnh các yếu tố môi trường để phù hợp với nhu cầu của cây trồng.
- **Độ ổn định**: hệ thống cho thấy khả năng hoạt động liên tục và ổn định. Các biểu đồ nhiệt độ và độ ẩm cho thấy sự thay đổi mượt mà và không có dấu hiệu bất thường, chứng tỏ hệ thống hoạt động ổn định. Tuy nhiên, để đánh giá chính xác hơn về độ ổn định, cần thu thập thêm dữ liệu trong thời gian dài và trong các điều kiện môi trường khác nhau.

❖ Khả năng mở rộng

- Hệ thống này có tiềm năng mở rộng rất lớn. Nó có thể được tích hợp với các cảm biến khác để theo dõi thêm các yếu tố như ánh sáng, độ pH của đất, độ ẩm không khí,... Ngoài ra, hệ thống cũng có thể được kết nối với các thiết bị điều khiển tự động khác như van tưới, máy bơm,... để tạo thành một hệ thống nhà xanh hoàn toàn tự động.

❖ Ứng dụng thực tế

- Nông nghiệp: Giúp nông dân quản lý và điều khiển quá trình trồng trọt một cách hiệu quả, tăng năng suất và chất lượng sản phẩm.
- Nghiên cứu: Dùng để nghiên cứu về sinh trưởng của cây trồng, ảnh hưởng của các yếu tố môi trường đến sự phát triển của cây.
- Giáo dục: Là công cụ hỗ trợ giảng dạy về nông nghiệp, môi trường cho học sinh, sinh viên.

5. Kết luận

Việc ứng dụng MQTT từ Adafruit IO trong hệ thống "SMART GARDEN SYSTEM" đã đem lại nhiều lợi ích to lớn trong việc giám sát và điều khiển các thông số môi trường một cách hiệu quả. Adafruit IO, với khả năng hỗ trợ giao thức MQTT, đã chứng minh là một nền tảng lý tưởng cho các dự án IoT, cung cấp khả năng truyền tải dữ liệu nhanh chóng và ổn định.

Qua MQTT, hệ thống có thể đảm bảo rằng dữ liệu từ các cảm biến và thiết bị được truyền tải liên tục, chính xác và đồng bộ. Điều này giúp người dùng theo dõi và điều khiển hệ thống từ xa một cách dễ dàng và tiện lợi. Sử dụng MQTT, các thiết bị có thể xuất bản và đăng ký thông điệp qua các chủ đề (topics) cụ thể, giúp tối ưu hóa việc truyền tải dữ liệu và tiết kiệm băng thông.

Nhìn chung, việc tích hợp MQTT từ Adafruit IO đã giúp hệ thống hoạt động mượt mà, đáng tin cậy và mang lại trải nghiệm người dùng tốt hơn. Sự kết hợp này không chỉ đáp ứng được các yêu cầu của dự án mà còn mở ra nhiều tiềm năng phát triển và ứng dụng rộng rãi trong các lĩnh vực khác nhau.

Các hướng phát triển trong tương lai:

- Phát triển ứng dụng di động: Cho phép người dùng điều khiển hệ thống từ xa thông qua điện thoại di động.
- Tích hợp với các nền tảng IoT: Kết nối hệ thống với các nền tảng IoT khác để tạo thành một hệ sinh thái thông minh.

Link Demo: <https://youtu.be/ppCOFEvYFSQ>

Link Github: [nhihao2903/SMART-GARDEN-SYSTEM](https://github.com/nhihao2903/SMART-GARDEN-SYSTEM)

MỤC LỤC

HTML (tag, features)	1
1. HTML Headings	1
2. HTML Paragraphs.....	2
3. HTML Styles	5
4. HTML Formatting	6
5. HTML Quotation.....	7
6. HMTL Colors.....	13
7. HTML CSS.....	14
8. HTML Links	19
9. HTML Images.....	25
10. HTML Tables.....	30
11. HTML Lists.....	47
12. HTML Block &Inline Elements	53
13. HTML Iframes.....	56
14. HTML Javascripts.....	57
15. HTML Head	59
16. HTML Layout.....	61
17. HTML Symbols.....	65
18. HTML Emojis	66
19. HTML URL Encode	66
20. HTML Forms	68
21. HTML Graphics	70
22. HTML Video	86
23. HTML Id	87
24. HTML Plug-ins	90
25. HTML Youtube	92
26. HTML Geolocation.....	95
27. HTML Div.....	99
28. HTML Computer Code.....	102
29. HTML Semantics.....	103
30. HTML Input Types	107
CSS (style)	110
1. CSS Color	110
2. CSS Background.....	113

3. CSS Border	116
4. CSS Margin	117
5. CSS Padding	120
6. CSS Height	122
7. CSS Outline	125
8. CSS Text	127
9. CSS Font	136
10. CSS Icons	141
11. CSS Lists	143
12. CSS Table	147
13. CSS Navigation Bar	155
14. CSS Image Gallery	168
15. CSS Website Layout	170
16. CSS Rounded Corners	195
17. CSS Multiple Columns	199
18. CSS Buttons	202
19. CSS The object-position Property	222
GIẢI THÍCH CODE SMART CITY	224
BÁO CÁO DASHBOARD SMART GARDEN SYSTEM	231
1. Giới thiệu	231
2. Kiến trúc Hệ thống và Phần cứng	231
3. Chức Năng Của Hệ Thống	242
4. Thiết Kế Giao Diện và Trải Nghiệm Người Dùng	243
4.1. Bố Cục Dashboard	243
4.2. Tương Tác Người Dùng và Điều Khiển	244
5. Kết Nối với Firebase và Giao Thức Truyền Dữ Liệu	249
5.1 Firebase	249
5.2 Giao Thức Kết Nối:	250
5.3 Truyền Dữ Liệu giữa Các Thành Phần	251
6. Xây dựng giao diện Mobile sử dụng Blynk IoT.	252
6.1. Giới thiệu về Blynk	252
6.2. Thiết kế và triển khai	253
6.2.1. Giao diện Blynk	253
6.2.2. Truyền Dữ Liệu giữa Các Thành Phần	254
7. Đánh Giá Hiệu Quả Hệ Thống	255

7.1. Ưu Điểm và Độ Ổn Định:	255
7.2. Hạn Chế và Đề Xuất Cải Thiện:	256
8. Kết Luận và Hướng Phát Triển.....	257
TRIỂN KHAI HỆ THỐNG SMART GARDEN SYSTEM TRÊN ADAFRUIT IO	259
1. Giới thiệu về Adafruit IO	259
2. Giao thức MQTT	259
3. Thiết kế giao diện người dùng	261
4. Đánh giá hiệu quả hệ thống	262
5. Kết luận	262
Link Demo: https://youtu.be/ppCOFEvYFSQ	263
Link Github: nihao2903/SMART-GARDEN-SYSTEM	263