TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

KHOA ĐIỆN – ĐIỆN TỬ

BỘ MÔN KỸ THUẬT MÁY TÍNH – VIỄN THÔNG

BÁO CÁO CUỐI KỲ

HỆ THỐNG NGÔI NHÀ THÔNG MINH

MÔN HỌC: THỰC TẬP CƠ SỞ VÀ ỨNG DỤNG IoT

NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

HỌC KỲ: II – NĂM HỌC: 2023 – 2024

GVHD: ThS. Trương Quang Phúc

Sinh viên: Võ Đức Hưng – 21119341

Đỗ Thái Dương – 21119058

Thành phố Hồ Chí Minh, tháng 5 năm 2024

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

****

BẢNG NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Họ tên	MSSV	Đánh giá	Điểm
Võ Đức Hưng	21119341		
Đỗ Thái Dương	21119058		

NHẬN XÉT CỦA GIÁO VIÊN

.....

.....

.....

.....

.....

.....

TP. Hồ Chí Minh, ngày 2 tháng 6 năm 2024

Giảng viên nhận xét

(Ký & ghi rõ họ tên)

MỤC LỤC

DANH MỤC HÌNH ẢNH 3
DANH MỤC CÁC BẢNG BIỂU 4
CHƯƠNG 1: GIỚI THIỆU 5
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 6
2.1. Ngôn ngữ đánh dấu siêu văn bản(HTML) 6
2.1.1. Giới thiệu về HTML 6
2.1.2 Cách hoạt động của HTML 7
2.1.3 Cấu trúc một đoạn HTML 8
2.1.4 Các Tags và Attributes thông dụng trong HTML 9
2.2. CSS 45
2.2.1. Giới thiệu về CSS 45
2.2.2. Bố cục và cấu trúc của một đoạn CSS 45
2.2.3. Các thuộc tính thông dụng trong CSS 47
2.3. Ngôn ngữ lập trình Javascript 84
2.3.1. Giới thiệu về Javascript 84
2.3.2 Tính năng chính của JavaScript: 84
2.3.3 Ưu và nhược điểm của JavaScript: 85
2.3.4 Ứng dụng phổ biến sử dụng JavaScript 86
2.4. Google Firebase 86
2.4.1. Giới thiệu về Google Firebase 86
2.4.2. Cách thức hoạt động 87
2.4.3. Ưu nhược điểm của Google Firebase 87
2.4.4. Ứng dụng phổ biến sử dụng Google Firebase 88
CHƯƠNG 3: YÊU CẦU THIẾT KẾ 90
3.1. Yêu cầu hệ thống 90
3.1.1. Yêu cầu người dùng 90
3.1.2. Yêu cầu kỹ thuật 90

3.1.3. Chức năng hệ thống 91
CHƯƠNG 4: PHẦN CỨNG 92
4.1 Sơ đồ đặc tả hệ thống 92
4.2 Sơ đồ kết nối 93
4.3 Giới thiệu phần cứng 94
4.3.1. Vi điều khiển ESP32 94
4.3.2. Cảm biến nhiệt độ, độ ẩm DHT11 95
4.3.3. Cảm biến khí Gas MQ-2 95
CHƯƠNG 5: PHẦN MỀM 97
5.1 Lưu đồ giải thuật 97
5.2 Giao diện kết quả 99
CHƯƠNG 6: KẾT QUẢ 100
6.1. Kết quả giao diện website 100
6.1.1. Giao diện web khi mới khởi tạo 100
6.1.2. Kết quả khi đọc các giá trị cảm biến 101
6.1.3. Kết quả khi sử dụng nút nhấn cho các thiết bị trong nhà 102
6.1.4. Kết quả giao diện trong trang thông tin “About” 104
CHƯƠNG 7: LINK YOUTUBE 105
TÀI LIỆU THAM KHẢO 106

DANH MỤC HÌNH ẢNH

Hình 4. 1 Sơ đồ đặc tả hệ thống 92
Hình 4. 2 Sơ đồ mạch nối dây 93
Hình 4. 3 Sơ đồ kết nối 93
Hình 4. 4 Sơ đồ chân ESP-32 94
Hình 4. 5 Sơ đồ chân của cảm biến nhiệt độ độ ẩm DHT11 95
Hình 4. 6 Sơ đồ chân của cảm biến khí gas MQ-2 95
Hình 5. 1 Lưu đồ giải thuật của hệ thống 97
Hình 6. 1 Slide giới thiệu về Smart Homes 100
Hình 6. 2 Các chức năng giám sát và điều khiển của hệ thống nhà thông minh ... 101
Hình 6. 3 Kết quả đọc được từ các cảm biến hiển thị lên website 101
Hình 6. 4 Kết quả đọc được từ các cảm biến hiển thị lên firebase 101
Hình 6. 5 Phần cứng khi chưa bật đèn 102
Hình 6. 6 Giao diện website khi chưa mở các thiết bị 102
Hình 6. 7 Phần cứng sau khi bật đèn 103
Hình 6. 8 Giao diện sau khi đã bật các thiết bị 103
Hình 6. 9 Thanh menu của giao diện 104
Hình 6. 10 Giới thiệu về nhóm ở trang giao diện About 104
Hình 7. 1 Hình ảnh nhóm 105

DANH MỤC CÁC BẢNG BIỂU

Bảng 2. 1 Các Tags và Attributes thông dụng trong HTML 9
Bảng 2. 2 Các thuộc tính thông dụng trong CSS 47

CHƯƠNG 1: GIỚI THIỆU

Ngày nay khoa học công nghệ phát triển nhằm đáp ứng nhu cầu ngày càng cao của con người. Việc nghiên cứu khoa học ngày càng được đầu tư để đáp ứng nhu cầu đó, các ngành công nghệ kỹ thuật điện tử đã có sự phát triển vượt bậc đưa khoa học vào kỷ nguyên mới. Kỹ thuật vi xử lý và điều khiển là một ứng dụng lớn của khoa học kỹ thuật vào cuộc sống phục vụ trực tiếp con người.

Nhà thông minh là ngôi nhà được trang bị các hệ thống tự động thông minh cùng với cách bố trí hợp lý, các hệ thống này có khả năng tự điều phối các hoạt động trong ngôi nhà theo thói quen sinh hoạt và nhu cầu cá nhân của gia chủ. Giải pháp nhà thông minh sẽ biến những món đồ điện tử bình thường trong ngôi nhà trở nên thông minh và gần gũi với người dùng hơn, chúng được kiểm soát thông qua các thiết bị truyền thông như điều khiển từ xa, điện thoại di động… Ngôi nhà thông minh đơn giản nhất có thể được hình dung bao gồm một mạng điều khiển liên kết một số lượng cố định các thiết bị điện, điện tử gia dụng trong ngôi nhà và chúng được điều khiển thông qua một chiếc điều khiển từ xa.

Ở đề tài "Thiết kế mô hình giám sát và điều khiển thông minh". Đây cũng là một trong những đề tài rất sát với thực tế, mang tính ứng dụng thực tiễn rất cao. Điều đó càng tạo động lực và cảm hứng cho sinh viên tìm tòi và nghiên cứu.

Trong bài báo cáo chắc hẳn còn nhiều sai sót, chúng em rất mong nhận được sự chỉ bảo, hướng dẫn của thầy cũng như sự đóng góp của các bạn sinh viên để đề tài hoàn thiện hơn.

Chúng em chân thành cảm ơn!

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Ngôn ngữ đánh dấu siêu văn bản(HTML)

2.1.1. Giới thiệu về HTML

HTML (HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản) là khối xây dựng cơ bản nhất của Web. Nó xác định ý nghĩa và cấu trúc của nội dung web. Các công nghệ khác ngoài HTML thường được sử dụng để mô tả giao diện/trình bày (CSS) hoặc chức năng/hành vi (JavaScript) của trang web.

"HyperText" dùng để chỉ các liên kết kết nối các trang web với nhau, trong một trang web hoặc giữa các trang web. Liên kết là một khía cạnh cơ bản của Web. Bằng cách tải nội dung lên Internet và liên kết nó với các trang do người khác tạo, bạn trở thành người tham gia tích cực vào World Wide Web.

HTML sử dụng "đánh dấu" để chú thích văn bản, hình ảnh và nội dung khác để hiển thị trong trình duyệt Web. Đánh dấu HTML bao gồm các "phần tử" đặc biệt như <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, , , < img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <search>, <output>, <progress>, <video>, <ul> , <ol>, <li> và nhiều thứ khác.

Một phần tử HTML được tách biệt khỏi văn bản khác trong tài liệu bằng "thẻ", bao gồm tên phần tử được bao quanh bởi "<" và ">". Tên của một phần tử bên trong thẻ không phân biệt chữ hoa chữ thường. Nghĩa là, nó có thể được viết bằng chữ hoa, chữ thường hoặc hỗn hợp. Ví dụ: thẻ <title> có thể được viết dưới dạng <Title>, <TITLE> hoặc theo bất kỳ cách nào khác. Tuy nhiên, quy ước và cách thực hành được khuyến nghị là viết thẻ bằng chữ thường.

2.1.2 Cách hoạt động của HTML

HTML là một tệp văn bản chứa cú pháp, tệp và quy ước đặt tên cụ thể giúp máy tính và máy chủ web biết rằng nó ở dạng HTML và nên được đọc như vậy. Bằng cách áp dụng các quy ước HTML này cho tệp văn bản trong hầu hết mọi trình soạn thảo văn bản, người dùng có thể viết và thiết kế một trang web cơ bản, sau đó tải nó lên internet.

Quy ước cơ bản nhất của HTML là đưa vào phần khai báo loại tài liệu ở đầu tệp văn bản. Phần này luôn xuất hiện đầu tiên trong tài liệu vì nó là phần thông báo chắc chắn cho máy tính rằng đây là tệp HTML. Tiêu đề tài liệu thường trông như thế này: <!DOCTYPE html>. Nó phải luôn được viết theo cách đó, không có bất kỳ nội dung nào bên trong hoặc chia nhỏ nó. Bất kỳ nội dung nào xuất hiện trước tuyên bố này sẽ không được máy tính công nhận là HTML.

Các loại tài liệu không chỉ được sử dụng cho HTML mà còn có thể áp dụng cho việc tạo bất kỳ tài liệu nào sử dụng SGML (Ngôn ngữ đánh dấu tổng quát tiêu chuẩn). SGML là một tiêu chuẩn để chỉ định ngôn ngữ đánh dấu cụ thể đang được sử dụng. HTML là một trong một số ngôn ngữ đánh dấu mà khai báo SGML và doctype áp dụng.

Yêu cầu quan trọng khác để tạo tệp HTML là lưu tệp đó với phần mở rộng tệp .html. Trong khi khai báo doctype báo hiệu HTML đến máy tính từ bên trong tệp, thì phần mở rộng tệp báo hiệu HTML đến máy tính từ bên ngoài tệp. Bằng cách có cả hai, máy tính có thể biết rằng đó là tệp HTML cho dù nó có đọc tệp hay không. Điều này trở nên đặc biệt quan trọng khi tải tệp lên web, vì máy chủ web cần biết phải làm gì với tệp trước khi có thể gửi chúng đến máy khách để đọc nội dung bên trong.

Sau khi viết loại tài liệu và lưu dưới dạng tệp HTML, người dùng có thể triển khai tất cả các công cụ cú pháp khác của HTML để tùy chỉnh trang web. Sau khi hoàn tất, họ có thể sẽ có một số tệp HTML tương ứng với các trang khác nhau của trang web. Điều quan trọng là người dùng phải tải các tệp này lên theo cùng thứ bậc mà họ đã lưu chúng, vì mỗi trang đều tham chiếu đường dẫn tệp cụ thể của các trang khác, cho phép

liên kết giữa chúng. Tải chúng lên theo thứ tự khác sẽ khiến liên kết bị hỏng và mất trang vì đường dẫn tệp được chỉ định sẽ không khớp với các trang.

2.1.3 Cấu trúc một đoạn HTML

Mỗi trang HTML chứa một bộ các tag (cũng được gọi là elements). Mỗi thẻ sẽ có những tác dụng nhất định, giúp xây dựng nên một cấu trúc hoàn chỉnh cho Website. Bạn có thể xem như là việc xây dựng từng khối của một trang web. Nó tạo thành cấu trúc cây thư mục bao gồm section, paragraph, heading, và những khối nội dung khác.

Hầu hết các HTML elements đều có tag mở và tag đóng với cấu trúc như <tag></tag>.

Để biết bố cục HTML của một trang web như thế nào, bạn có thể xem code ví dụ của một trang HTML được cấu trúc như thế nào:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>The Main Heading</h1>
    <h2>A catchy subheading</h2>
    <p>First paragraph</p>
  </body>
</html>
```

Trong đó:

- ***<!DOCTYPE html>**: **khai** báo kiểu dữ liệu hiển thị*
- ***<html>** và **</html>**: cặp thẻ bắt buộc, element cấp cao nhất, có nhiệm vụ đóng gói tất cả nội dung của trang HTML*

- **<head>** và **</head>**: khai báo các thông tin meta của trang web như: tiêu đề trang, charset
- **<title>** và **</title>**: cặp thẻ nằm bên trong thẻ <head>, dùng để khai báo tiêu đề của trang
- **<body>** và **</body>**: cặp thẻ dùng để đóng gói tất cả các nội dung sẽ hiển thị trên trang
- **<h1></h1>, <h2></h2>**: định dạng dữ liệu dạng heading. Thông thường có 6 cấp độ heading trong HTML, trải dài từ <h1> tới <h6>. Trong đó, <h1> là cấp độ heading cao nhất và <h6> là cấp độ heading thấp nhất.
- ** ** và ** **: cặp thẻ chứa các đoạn văn bản của trang web.

2.1.4 Các Tags và Attributes thông dụng trong HTML

Dưới đây là các Tags và Attributes trong HTML thường được sử dụng khi thiết kế một Website:

Bảng 2. 1 Các Tags và Attributes thông dụng trong HTML

STT	HTML	Lý thuyết	Code	Result
1	HTML Heading	HTML Heading là tiêu đề hoặc phụ đề mà bạn muốn hiển thị trên trang web. HTML heading được xác định bằng thẻ <h1> đến <h6>. Thẻ <h1> xác định tiêu đề quan trọng nhất. Thẻ <h6> xác định tiêu đề ít quan trọng nhất.	<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>Document</title> </head> <body> <h1>Heading 1</h1> <h2>Heading 2</h2> <h3>Heading 3</h3> <h4>Heading	**Heading 1** **Heading 2** **Heading 3** **Heading 4** **Heading 5** **Heading 6**

			4</h4> <h5>Heading 5</h5> <h6>Heading 6</h6> </body> </html>	
2	HTML Paragraphs	Đoạn văn luôn bắt đầu trên một dòng mới và thường là một khối văn bản. Thẻ HTML xác định một đoạn văn. Một đoạn văn luôn bắt đầu trên một dòng mới và trình duyệt sẽ tự động thêm một số khoảng trắng (lề) trước và sau một đoạn văn.	<!DOCTYPE html> <html lang="en">> <head> <meta charset="UTF-8">> <title>Document</title> </head> <body> This is a paragraph. This is another paragraph. </body> </html>	This is a paragraph. This is another paragraph.
3	HTML Styles	Thuộc tính Style được sử dụng để thêm kiểu cho một thành phần, chẳng hạn như màu sắc, phông chữ, kích thước, v.v. Cú pháp: <tagname style="property:value;"> với: Property là thuộc tính CSS, Value là giá trị CSS. Các thuộc tính CSS thường dùng trong Style: - Use background-color for background color. - Use color for text	<!DOCTYPE html> <html lang="en">> <head> <meta charset="UTF-8">> <title>Document</title> </head> <body> <h1 style="background-color:powderblue;">This is a heading</h1> <h1 style="color:blue;">T	This is a heading This is a heading This is a heading This is a heading Centered Heading

		colors. - Use font-family for text fonts. - Use font-size for text sizes - Use text-align for text alignment.	his is a heading</h1> <h1 style="font-family:verdana;">This is a heading</h1> <h1 style="font-size:60px;">This is a heading</h1> <h1 style="text-align:center;">Centered Heading</h1> </body> </html>	
4	HTML Formatting	Trong khi hầu hết các thẻ HTML được sử dụng để tạo các phần tử, HTML cũng cung cấp các thẻ định dạng trong văn bản để áp dụng các kiểu liên quan đến văn bản cụ thể cho các phần văn bản. Các phần tử định dạng được thiết kế để hiển thị các loại văn bản đặc biệt: - - Bold text - - Important text - - Italic text - - Emphasized text - <mark> - Marked text - <small> - Smaller text - <del> - Deleted text - <ins> - Inserted text - <sub> - Subscript text - <sup> - Superscript	<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8">* <title>Document</title> </head> <body> This text is bold This text is important! This text is italic This text is emphasized <small>This is some smaller text.</small> <p>Do not forget to buy	**This text is bold** **This text is important!** *This text is italic* *This text is emphasized* This is some smaller text. Do not forget to buy milk today. My favorite color is ~~blue~~ red. This is $_{\text{subscripted}}$ text. This is ^{superscripted} text.

		text	<mark>milk</mark> today. My favorite color is <del>blue</del> <ins>red</ins>. This is _{subscripted} text. This is ^{superscripted} text. </body> </html>	
5	HTML Quotations	In this chapter we will go through the <blockquote>,<q>, <abbr>, <address>, <cite>, and <bdo> HTML elements. HTML Quotation and Citation Elements Tag — Description <abbr> — Defines an abbreviation or acronym <address> — Defines contact information for the author/owner of a document <bdo> — Defines the text direction <blockquote> — Defines a section that is quoted from another source <cite> — Defines the title of a work <q> — Defines a short inline quotation	<!DOCTYPE html> <html> <body> Here is a quote from WWF's website: <blockquote cite="http://www.worldwildlife.org/who/index.html"> For 60 years, WWF has worked to help people and nature thrive. As the world's leading conservation organization, WWF works in nearly 100 countries. At every level, we collaborate with people around the world to develop and deliver innovative solutions that protect communities, wildlife,	Here is a quote from WWF's website: For 60 years, WWF has worked to help people and nature thrive. As the world's leading conservation organization, WWF works in nearly 100 countries. At every level, we collaborate with people around the world to develop and deliver innovative solutions that protect communities, wildlife, and the places in which they live.

			and the places in which they live. </blockquote> </body> </html>	
6	HTML Colors	HTML colors được chỉ định bằng tên màu được xác định trước hoặc bằng các giá trị RGB, HEX, HSL, RGBA hoặc HSLA.	<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>Document</title> </head> <body> <h1 style="color:Tomato;">Hello World</h1> <p style="color:DodgerBlue;">Lorem ipsum...</p> <h1 style="background-color:rgba(25, 199, 0, 0.5);">...</h1> <h1 style="background-color:hsla(9, 100%, 64%, 0.2);">...</h1> </body> </html>	Hello World Lorem ipsum... ... ...
7	HTML CSS	Cascading Style Sheets (CSS) được sử dụng để	<!DOCTYPE html> <html>	This is a heading This is a paragraph.

		định dạng bố cục của trang web.
		Với CSS, bạn có thể kiểm soát màu sắc, phông chữ, kích thước văn bản, khoảng cách giữa các thành phần, cách các thành phần được định vị và bố trí, hình nền hoặc màu nền nào sẽ được sử dụng, các hiển thị khác nhau cho các thiết bị và kích thước màn hình khác nhau, và nhiều hơn nữa!
		CSS có thể được thêm vào tài liệu HTML theo 3 cách:
		- Inline - bằng cách sử dụng thuộc tính style bên trong các phần tử HTML
		- Internal - bằng cách sử dụng phần tử <style> trong phần <head>
		- External - bằng cách sử dụng phần tử <link> để liên kết với phần bên ngoài tập tin CSS
			<head>
			<style>
			h1 {
			color: blue;
			font-family: verdana;
			font-size: 300%;
			}
			p {
			color: red;
			font-family: courier;
			font-size: 160%;
			}
			</style>
			</head>
			<body>
			
			<h1>This is a heading</h1>
			 This is a paragraph.
			
			</body>
			</html>	

8	HTML Links	HTML Links có thể được sử dụng để tạo dấu trang để người đọc có thể chuyển đến các phần cụ thể của trang web. Tạo dấu trang trong HTML Dấu trang có thể hữu ích nếu một trang web rất dài. Để tạo dấu trang - trước tiên hãy tạo dấu trang rồi thêm liên kết vào dấu trang đó. Khi nhấp vào liên kết, trang sẽ cuộn xuống hoặc cuộn lên vị trí có dấu trang.	<!DOCTYPE html> <html> <body> <a href="#C4">Jump to Chapter 4</a> <a href="#C10">Jump to Chapter 10</a> <h2>Chapter 1</h2> This chapter explains ba bla bla <h2>Chapter 2</h2> This chapter explains ba bla bla <h2>Chapter 3</h2> This chapter explains ba bla bla <h2 id="C4">Chapter 4</h2> This chapter explains ba bla bla <h2>Chapter 5</h2> This chapter explains ba bla bla <h2>Chapter 6</h2> This chapter explains ba bla bla	Jump to Chapter 4 Jump to Chapter 10 **Chapter 1** This chapter explains ba bla bla **Chapter 2** This chapter explains ba bla bla **Chapter 3** This chapter explains ba bla bla **Chapter 4** This chapter explains ba bla bla **Chapter 5** This chapter explains ba bla bla **Chapter 6** This chapter explains ba bla bla **Chapter 7** This chapter explains ba bla bla

			<h2>Chapter 7</h2> This chapter explains ba bla bla <h2>Chapter 8</h2> This chapter explains ba bla bla <h2>Chapter 9</h2> This chapter explains ba bla bla <h2 id="C10">Chapter 10</h2> This chapter explains ba bla bla <h2>Chapter 11</h2> This chapter explains ba bla bla <h2>Chapter 12</h2> This chapter explains ba bla bla <h2>Chapter 13</h2> This chapter explains ba bla bla <h2>Chapter 14</h2> This chapter explains ba bla bla	**Chapter 8** This chapter explains ba bla bla **Chapter 9** This chapter explains ba bla bla **Chapter 10** This chapter explains ba bla bla **Chapter 11** This chapter explains ba bla bla **Chapter 12** This chapter explains ba bla bla **Chapter 13** This chapter explains ba bla bla **Chapter 14** This chapter explains ba bla bla ***Nhận xét***: khi nhấn vào 2 dòng Jump to Chapter 4 và Jump to Chapter 10 nó sẽ nhảy tới chapter tương ứng

			</body> </html>	
9	HTML Images	Hình ảnh có thể cải thiện thiết kế và sự xuất hiện của một trang web.	<!DOCTYPE html> <html> <body> <h2>Cup 50</h2> Thuộc tính alt sẽ phản ánh nội dung hình ảnh, để người dùng không thể nhìn thấy hình ảnh sẽ hiểu được nội dung trong hình ảnh: <img src="https://i.pinimg.com/564x/8e/c9/21/8ec9214f2b666fc7e4c3a4a1a88da8ea.jpg" alt="Hình ảnh chiếc xe máy cup50" width="460" height="345"> </body>	**Cup 50** Thuộc tính alt sẽ phản ánh nội dung hình ảnh ***Nhận xét:*** *Khi ảnh bị lỗi thì sẽ có kết quả như sau* **Cup 50** Thuộc tính alt sẽ phản ánh nội dung hình ảnh Hình ảnh chiếc xe máy cup50

			</html>	

10	HTML Tables	HTML Tables cho phép các nhà phát triển web sắp xếp dữ liệu thành hàng và cột.	<!DOCTYPE html>
			<html>
			<style>
			table, th, td {
			  border:1px solid black;
			}
			</style>
			<body>
			
			<h2>A basic HTML table</h2>
			
			<table style="width:100%">
			  <tr>
			  <th>Company</th>
			  <th>Contact</th>
			  <th>Country</th>
			  </tr>
			  <tr>
			  <td>Alfreds Futterkiste</td>
			  <td>Maria Anders</td>
			  <td>Germany</td>
			  </tr>
			  <tr>
			  <td>Centro comercial Moctezuma</td>
			  <td>Francisco Chang</td>
			  <td>Mexico</td>
			  </tr>
			</table>
			
			<p>To understand the
				A basic HTML table
				Company | Contact | Country
				Alfreds Futterkiste | Maria Anders | Germany
				Centro comercial Moctezuma | Francisco Chang | Mexico
				To understand the example better, we have added borders to the table.

			example better, we have added borders to the table.</p> </body> </html>	
11	HTML Lists	HTML Lists cho phép các nhà phát triển web nhóm một tập hợp các mục liên quan vào danh sách. 1. HTML cũng hỗ trợ danh sách mô tả. Danh sách mô tả là danh sách các thuật ngữ, kèm theo mô tả cho từng thuật ngữ. Thẻ <dl> xác định danh sách mô tả, thẻ <dt> xác định thuật ngữ (tên) và thẻ <dd> mô tả từng thuật ngữ:	<!--HTML Description Lists--> <!DOCTYPE html> <html> <body> <h2>A Description List</h2> <dl> <dt>Coffee</dt> <dd>- black hot drink</dd> <dt>Milk</dt> <dd>- white cold drink</dd> </dl> </body> </html>	**A Description List** Coffee - black hot drink Milk - white cold drink
		2. Danh sách có thứ tự bắt đầu bằng thẻ <ol>. Mỗi mục danh sách bắt đầu bằng thẻ <li>. Các mục trong danh sách sẽ được đánh dấu bằng số theo mặc định:	<!--Ordered HTML List--> <!DOCTYPE html> <html> <body> <h2>An ordered HTML list</h2> <ol> <li>Duc Hung</li>	**An ordered HTML list** 1. Duc Hung 2. Thai Duong 3. Van A

			<li>Thai Duong</li> <li>Van A</li> </ol> </body> </html>	
		3. Danh sách không có thứ tự bắt đầu bằng thẻ <ul>. Mỗi mục danh sách bắt đầu bằng thẻ <li>. Các mục trong danh sách sẽ được đánh dấu bằng dấu đầu dòng (vòng tròn nhỏ màu đen) theo mặc định:	<!DOCTYPE html> <html> <body> <h2>An unordered HTML list</h2> <ul> <li>Duc Hung</li> <li>Thai Duong</li> <li>Van A</li> </ul> </body> </html>	**An unordered HTML list** • Duc Hung • Thai Duong • Van A
12	HTML Block and Inline	Mọi 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 and inline.. -Block-level Elements A block-level element luôn bắt đầu trên một	<!DOCTYPE html> <html> <body> Hello World Hello	. Hello World Hello World The P and the DIV elements are both block elements, and they will always start on a new line and take up the full width available (stretches out to the left and right as far as it can).

		dòng mới và trình duyệt sẽ tự động thêm một số khoảng trắng (lề) trước và sau phần tử. Phần tử cấp khối luôn chiếm toàn bộ chiều rộng có sẵn (kéo dài sang trái và phải hết mức có thể). Hai phần tử khối thường được sử dụng là: và . Phần tử xác định một đoạn văn trong tài liệu HTML. Phần tử xác định một phần hoặc một phần trong tài liệu HTML.	World The P and the DIV elements are both block elements, and they will always start on a new line and take up the full width available (stretches out to the left and right as far as it can). </body> </html>	
		An inline element không bắt đầu trên một dòng mới. An inline element chỉ chiếm nhiều chiều rộng nếu cần thiết. Đây là phần tử bên trong một đoạn văn.	<!DOCTYPE html> <html> <body> This is an inline span Hello World element inside a paragraph. The SPAN element is an inline element, and will not start on a new line and	This is an inline span Hello World element inside a paragraph. The SPAN element is an inline element, and will not start on a new line and only takes up as much width as necessary.

			only takes up as much width as necessary. </body> </html>	
13	HTML Iframes	HTML Iframes được sử dụng để hiển thị một trang web trong một trang web. **Cú pháp:** Thẻ HTML <iframe> chỉ định khung nội tuyến. Khung nội tuyến được sử dụng để nhúng tài liệu khác trong tài liệu HTML hiện tại. <iframe src="*url*" title="*description*"></iframe> Có thể sử dụng height và width attributes để chỉnh kích cỡ của iframe	<!DOCTYPE html> <html> <body> <h2>HTML Iframes</h2> You can use the height and width attributes to specify the size of the iframe: <iframe src="demo_iframe.htm" height="200" width="300" title="Iframe Example"></iframe> </body> </html>	**HTML Iframes** You can use the height and width attributes to specify the size of the iframe: **This page is displayed in an iframe**
14	HTML Javascripts	JavaScript làm cho các trang HTML năng động và tương tác hơn. JavaScript là ngôn ngữ lập trình linh hoạt thường được sử dụng để viết kịch bản phía máy khách trong phát triển web -The HTML <script> Tag Thẻ HTML <script> được sử dụng để xác	<!DOCTYPE html> <html> <body> <h2>Use JavaScript to Change Text</h2> This example writes "Hello JavaScript!" into an HTML element with id="demo": 	**Use JavaScript to Change Text** This example writes "Hello JavaScript!" into an HTML element with id="demo": Hello JavaScript!

		định tập lệnh phía máy khách (JavaScript). Phần tử <script> chứa các câu lệnh script hoặc nó trỏ đến tệp script bên ngoài thông qua thuộc tính src. Các cách sử dụng phổ biến của JavaScript là thao tác hình ảnh, xác thực biểu mẫu và thay đổi nội dung động. Để chọn một phần tử HTML, JavaScript thường sử dụng phương thức document.getElementById(). Ví dụ JavaScript này viết "Hello JavaScript!" thành một phần tử HTML có id="demo":	<script> document.getElementById("demo").innerHTML = "Hello JavaScript!"; </script> </body> </html>	
		A Taste of JavaScript	<!DOCTYPE html> <html> <body> <h1>My First JavaScript</h1> JavaScript can change the content of an HTML element: <button type="button" onclick="myFunction()">Click Me!</button> This is a demonstration.	**My First JavaScript** JavaScript can change the content of an HTML element: Click Me! This is a demonstration. Nhận xét: khi nhấn vào button có chữ Clock Me! thì dòng cuối cùng chuyển đổi từ “This is a demonstration” thành “Hello JavaScrip!” **My First JavaScript** JavaScript can change the content of an HTML element: Click Me! Hello JavaScript!

			<script>
			function myFunction()
			{
			
			document.getElementById("demo").innerHTML = "Hello JavaScript!";
			}
			</script>
			
			</body>
			</html>	

		The HTML <noscript> Tag: Thẻ HTML <noscript> xác định nội dung thay thế sẽ được hiển thị cho người dùng đã tắt tập lệnh trong trình duyệt của họ hoặc có trình duyệt không hỗ trợ tập lệnh:	<!DOCTYPE html> <html> <body> <script> document.getElementById("demo").innerHTML = "Hello JavaScript!"; </script> <noscript>Sorry, your browser does not support JavaScript!</noscript> A browser without support for JavaScript will show the text written inside the noscript element. </body> </html>	Hello JavaScript! A browser without support for JavaScript will show the text written inside the noscript element. *Nhận xét: Khi trình duyệt có hỗ trợ JavaScript, nội dung của phần tử có id "demo" sẽ được thay đổi thành "Hello JavaScript!". Trong trường hợp trình duyệt không hỗ trợ JavaScript, văn bản trong phần <noscript> sẽ được hiển thị:"A browser without support for JavaScript will show the text written inside the noscript element."*

<table>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr>
<td>15</td>
<td>HTML Head</td>
<td>Phần tử HTML <head> là nơi chứa các phần tử sau: <title>, <style>, <meta>, <link>, <script> và <base>.
HTML head Elements
<table><tr><th>Tag</th><th>Description</th></tr><tr><td><head></td><td>Defines information about the document</td></tr><tr><td><title></td><td>Defines the title of a document</td></tr><tr><td><base></td><td>Defines a default address or a default target for all links on a page</td></tr><tr><td><link></td><td>Defines the relationship between a document and an external resource</td></tr><tr><td><meta></td><td>Defines metadata about an HTML document</td></tr><tr><td><script></td><td>Defines a client-side script</td></tr><tr><td><style></td><td>Defines style information for a document</td></tr></table></td>
<td><!DOCTYPE html>
<html lang="en">
<head><br   <meta charset="UTF-8">

<title>Document</title><br   <script><br   function myFunction() {

document.getElementById("demo").innerHTML = "Hello JavaScript!";}<br   </script>
</head>
<body><br   <h1>My Web Page</h1><br   A Paragraph <br   <button type="button" onclick="myFunction()">Try it</button>
</body>
</html></td>
<td>My Web Page
A Paragraph
Try it

Click to button “try it”:
My Web Page
Hello JavaScript!
Try it</td>
</tr>
</table>

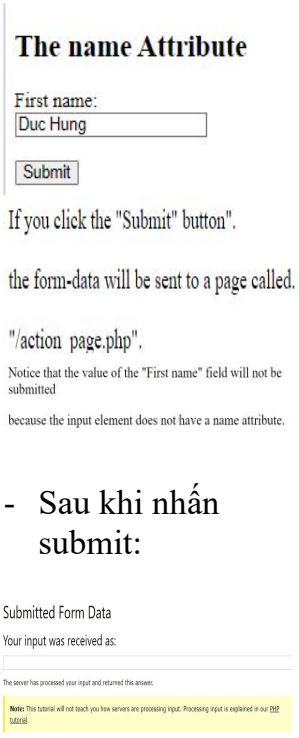
16	HTML Layout	Trang web thường hiển thị nội dung theo nhiều cột (như tạp chí hoặc báo).	<!DOCTYPE html>
			<html lang="en">
			<head>
			<title>CSS Template</title>
			<meta charset="utf-8">
			<meta name="viewport" content="width=device-width, initial-scale=1">
			<style>
			* {
			 box-sizing: border-box;
			}
			body {
			 font-family: Arial, Helvetica, sans-serif;
			}
			header {
			 background-color: #666;
			 padding: 5px;
			 text-align: center;
			 font-size: 20px;
			 color: white;
			}
			nav {
			 float: left;
			 width: 30%;
			 height: 200px;
			 background: #ccc;
			 padding: 20px;
			}
			
			nav ul {
			 list-style-type: none;
			 padding: 0;
			}
			article {
				A basic HTML table
				
				Company | Contact | Country
				Alfreds Futterkiste | Maria Anders | Germany
				Centro comercial Moctezuma | Francisco Chang | Mexico
				
				To understand the example better, we have added borders to the table.

			float: left;
			padding: 20px;
			width: 70%;
			background-color: #f1f1f1;
			height: 200px;
			}
			section::after {
			content: "";
			display: table;
			clear: both;
			}
			footer {
			background-color: #777;
			padding: 10px;
			text-align: center;
			color: white;
			}
			@media (max-width: 600px) {
			nav, article {
			width: 100%;
			height: auto;
			}
			}
			</style>
			</head>
			<body>
			<h2>CSS Layout Float</h2>
			<header>
			<h2>Cities</h2>
			</header>
			<section>
			<nav>
			<ul>
			<li><a href="#">London</a></li>
			</ul>
			</nav>	

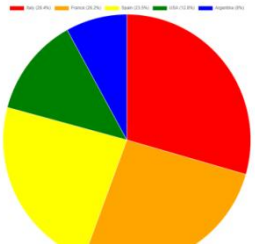
			<article> <h1>London</h1> London is the capital city of England... </article> </section> <footer> Footer </footer> </body> </html>	
17	HTML Symbols	Các ký hiệu hoặc chữ cái không có trên bàn phím của bạn có thể được thêm vào HTML bằng cách sử dụng các thực thể Nhiều ký hiệu toán học, kỹ thuật và tiền tệ không có trên bàn phím thông thường. Để thêm các ký hiệu như vậy vào trang HTML, bạn có thể sử dụng tên thực thể hoặc số thực thể (tham chiếu thập phân hoặc thập lục phân) cho ký hiệu:	<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8">\ <title>Document</title> </head> <body> I will display € I will display Α I will display ♠ I will display © I will display Δ I will display ₺ </body> </html>	I will display € I will display Α I will display ♠ I will display © I will display Δ I will display ₤

18	HTML Emojis	Biểu tượng cảm xúc trông giống như hình ảnh hoặc biểu tượng nhưng thực tế không phải vậy. Chúng là các chữ cái (ký tự) từ bộ ký tự UTF-8 (Unicode). UTF-8 bao gồm hầu hết các ký tự và biểu tượng trên thế giới.	<!DOCTYPE html> <html> <meta charset="UTF-8"> <body> <h1>Sized Emojis</h1> 😀 😄 😍 💗 </body> </html>	**Sized Emojis** 😀 😄 😍 💗
19	HTML URL Encode	URL là một từ khác chỉ địa chỉ web. URL có thể bao gồm các từ (ví dụ: w3schools.com) hoặc địa chỉ Giao thức Internet (IP) (ví dụ: 192.68.20.50). Hầu hết mọi người đều nhập tên khi lướt web, vì tên dễ nhớ hơn số. Địa chỉ web như https://www.w3schools.com/html/default.asp tuân theo các quy tắc cú pháp sau: `scheme://prefix.domain:port/path/filename` Giải thích: Scheme - xác định loại tiền tố dịch vụ Internet (phổ biến nhất là http hoặc https).	<!DOCTYPE html> <html> <head> <title>URL Encoding Example</title> </head> <body> <input type="text" id="originalUrl" placeholder="Enter URL"> <button onclick="encodeUrl()">Encode URL</button> Encoded URL:	https://www.w3schools.com/ Encode URL Encoded URL: https%3A%2F%2Fwww.w3schools.com%2Fhtml%2Ftryit.asp%3Ffilename%3Dtryhtml_noscript

		Prefix- xác định tiền tố tên miền.
		Domain - xác định tên miền Internet.
		Port - xác định số cổng tại đường dẫn máy chủ (mặc định cho http là 80) .
		Path - xác định đường dẫn tại máy chủ .
		Filename - xác định tên của tài liệu hoặc tài nguyên.
			<script>
			function encodeUrl() {
			var originalUrl = document.getElementById("originalUrl").value;
			var encodedUrl = encodeURIComponent(originalUrl);
			
			document.getElementById("encodedUrl").textContent= encodedUrl;
			}
			</script>
			
			</body>
			</html>	

20	HTML Froms	Một biểu mẫu HTML được sử dụng để thu thập thông tin đầu vào của người dùng. Đầu vào của người dùng thường được gửi đến máy chủ để xử lý.	<!DOCTYPE html> <html> <body> <h2>The name Attribute</h2> <form action="/action_page.php"> <label for="fname">First name:</label> <input type="text" id="fname" value="John"> <input type="submit" value="Submit">>	

			</form> If you click the "Submit" button". the form-data will be sent to a page called. "/action_page.php ". Notice that the value of the "First name" field will not be submitted because the input element does not have a name attribute. </body> </html>	

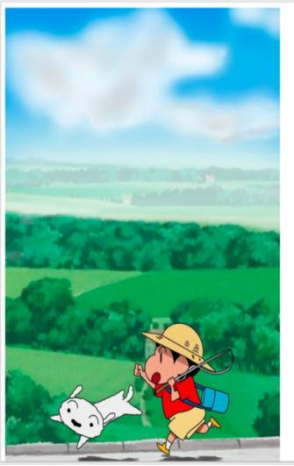
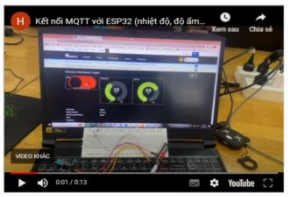
21	HTML Graphics	HTML Graphics đề cập đến việc sử dụng các thành phần và kỹ thuật HTML khác nhau để tạo và hiển thị nội dung đồ họa trên một trang web. Bản thân HTML cung cấp một số khả năng đồ họa cơ bản thông qua các phần tử như <img> để hiển thị hình ảnh và <canvas> để vẽ đồ họa động bằng JavaScript. Ngoài ra, CSS có thể được sử dụng để tạo kiểu và thao tác hình thức của các phần tử HTML nhằm tạo ra các thiết kế hấp dẫn về mặt hình ảnh.	<script
			src="https://cdn.jsdelivr.net/npm/chart.js@3.8.0/dist/chart.min.js">
			</script>
			<canvas id="myChart" width="400" height="400"></canvas>
			
			<script>
			var ctx = document.getElementById('myChart').getContext('2d');
			var chart = new Chart(ctx, {
			type: 'pie',
			data: {
			labels: ['Italy (29.4%)', 'France (26.2%)', 'Spain
				

			(23.5%)', 'USA (12.8%)', 'Argentina (8%)'],
			datasets: [{
			data: [29.4, 26.2, 23.5, 12.8, 8],
			backgroundColor: [
			'red',
			'orange',
			'yellow',
			'green',
			'blue'
			],
			borderWidth: 1,
			borderColor: '#fff'
			}]
			},
			options: {
			// Set aspect ratio of the chart to be slightly bigger than 1,
			// creating a 3D effect.
			aspectRatio: 1.2,
			// Hide legend
			legend: {
			display: false
			},
			// Add some padding around the chart
			layout: {
			padding: 20
			},
			// Add a shadow to give the chart a 3D feel
			plugins: {
			shadow: {
			enabled: true,
			color: 'rgba(0, 0,	

			0, 0.2)', blur: 10, } } } }); </script>	
22	HTML video	Phần tử HTML <video> được sử dụng để hiển thị video trên trang web.	<!DOCTYPE html> <html> <body> <button onclick="playPause()">Play/Pause</button> <button onclick="makeBig()">Big</button> <button onclick="makeSmall()">Small</button> <button onclick="makeNormal()">Normal</button> <video id="video1" width="420"> <source src="mov_bbb.mp4" type="video/mp4"> <source src="mov_bbb.ogg" type="video/ogg"> Your browser does not support HTML video. </video> 	

			<script>
			var myVideo = document.getElementById("video1");
			
			function playPause() {
			  if (myVideo.paused)
			  myVideo.play();
			  else
			  myVideo.pause();
			}
			
			function makeBig() {
			  myVideo.width = 560;
			}
			
			function makeSmall() {
			  myVideo.width = 320;
			}
			
			function makeNormal() {
			  myVideo.width = 420;
			}
			</script>
			
			 Video courtesy of <a href="https://www.bigbuckbunny.org/" target="_blank">Big Buck Bunny</a>.
			
			</body>
			</html>	

23	HTML id	Thuộc tính id HTML được sử dụng để chỉ định một id duy nhất cho một phần tử HTML.
		Thuộc tính id chỉ định một id duy nhất cho một phần tử HTML. Giá trị của thuộc tính id phải là duy nhất trong tài liệu HTML.
		Thuộc tính id được sử dụng để trỏ đến một khai báo kiểu cụ thể trong biểu định kiểu. Nó cũng được JavaScript sử dụng để truy cập và thao tác phần tử với id cụ thể.
		Cú pháp của id là: viết ký tự (#), theo sau là tên id. Sau đó, xác định thuộc tính CSS trong dấu ngoặc nhọn {}.
		Trong ví dụ sau, chúng ta có phần tử <h1> trỏ đến tên id "myHeader". Phần tử <h1> này sẽ được tạo kiểu theo định nghĩa kiểu #myHeader trong phần đầu:
			<!-- Using The id Attribute -->
			<!DOCTYPE html>
			<html>
			<head>
			<style>
			#myHeader {
			background-color: black;
			color: lightblue;
			padding: 40px;
			text-align: center;
			}
			</style>
			</head>
			<body>
			
			<h2>The id Attribute</h2>
			 Sử dụng CSS để tạo kiểu cho một phần tử có id "myHeader":
			
			<h1 id="myHeader">Report HTML For Iot</h1>
			
			</body>
			</html>	

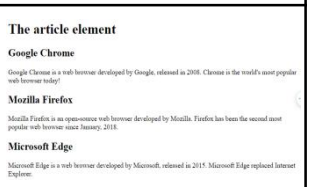
24	HTML Plug-ins	-HTML Plug-ins là các thành phần phần mềm bổ sung mở rộng chức năng của trình duyệt web. -Plug-in được thiết kế để sử dụng cho nhiều mục đích khác nhau: +Để chạy các ứng dụng Java +Để chạy các điều khiển +Microsoft ActiveX +Để hiển thị phim Flash +Để hiển thị bản đồ +Để quét virus +Để xác minh id ngân hàng	<!-- Ví dụ bên dưới là hiển thị hình ảnh --> <!DOCTYPE html> <html> <body> <object data="https://i.pinimg.com/564x/d5/9b/ed/d59bed442ae494da5d2751e100cc3416.jpg"></object> </body> </html>	
25	HTML YouTube	HTML YouTube là việc nhúng video từ YouTube vào một trang web bằng mã HTML. Điều này liên quan đến việc sử dụng phần tử <iframe> do YouTube cung cấp, cho phép bạn nhúng video một cách dễ dàng.	<!DOCTYPE html> <html> <body> <iframe width="560" height="400" src="https://www.youtube.com/embed/kvv1S1EmQBI?si=CPVeEIa6pGBiBsU1"> </iframe> </body> </html>	
26	HTML Geolocation API	-HTML Geolocation API được sử dụng để xác định vị trí của người dùng. Xác định vị trí của người dùng -HTML Geolocation API được sử dụng để có được vị trí địa lý của người dùng. Vì điều này có thể ảnh	<!-- Ví dụ bên dưới trả về vĩ độ và kinh độ vị trí của người dùng --> <!DOCTYPE html> <html> <body> <h1>HTML Geolocation</h1> <p>Click the button to get your	**HTML Geolocation** Click the button to get your coordinates. Try It Latitude: 10.8607871 Longitude: 106.7817171 Nhận xét: Ở kết quả trên khi nhấm Try it thì code sẽ lấy vĩ độ

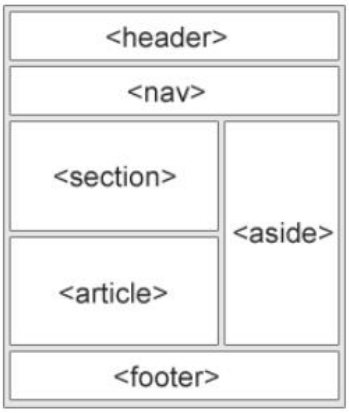
		hưởng đến quyền riêng tư nên vị trí này sẽ không có sẵn trừ khi người dùng chấp thuận.
		-Hàm getCurrentPosition() được sử dụng để trả về vị trí của người dùng.
		-Phương thức getCurrentPosition() - Trả về dữ liệu: Phương thức getCurrentPosition() trả về một đối tượng nếu thành công. Các thuộc tính vĩ độ, kinh độ và độ chính xác luôn được trả về. Các thuộc tính khác được trả về nếu có:
		ví dụ như:
		+ coords.latitude trả về Vĩ độ dưới dạng số thập phân (luôn được trả về)
		+ coords.longitude trả về Kinh độ dưới dạng số thập phân (luôn được trả về)
		+ coords.accuracy trả về Độ chính xác của vị trí (luôn được trả về)
			coordinates.
			
			<button onclick="getLocation()">Try It</button>
			
			<script>
			const x = document.getElementById("demo");
			
			function getLocation() {
			  if (navigator.geolocation) {
			
			navigator.geolocation.getCurrentPosition(showPosition);
			  } else {
			  x.innerHTML = "Geolocation is not supported by this browser.";
			  }
			}
			
			function showPosition(position) {
			  x.innerHTML = "Latitude: " + position.coords.latitude +
			  "
			Longitude: " + position.coords.longitude;
			}
			</script>
				và kinh độ của người dùng để hiển thị lên.

			</body> </html>	
27	HTML Div	Phần tử được sử dụng làm vùng chứa cho các phần tử HTML khác. Phần tử theo mặc định là phần tử khối, nghĩa là nó chiếm toàn bộ chiều rộng có sẵn và đi kèm với ngắt dòng trước và sau.	<!DOCTYPE html> <html> <body> <h1>Multiple DIV Elements</h1> <h2>London</h2> London is the capital city of England. London has over 13 million inhabitants. <h2>Oslo</h2> Oslo is the capital city of Norway. Oslo has over 600.000 inhabitants. <h2>Rome</h2> Rome is the capital city of Italy.	**Multiple DIV Elements** **London** London is the capital city of England. London has over 13 million inhabitants. **Oslo** Oslo is the capital city of Norway. Oslo has over 600.000 inhabitants. **Rome** Rome is the capital city of Italy. Rome has almost 3 million inhabitants.

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td><p>Rome has almost 3 million inhabitants.</p>
</div>
</body>
</html></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>HTML Computer code</td>
<td>HTML chứa một số phần tử để xác định đầu vào của người dùng và mã máy tính.
Phần tử <kbd> xác định đầu vào bàn phím.
Phần tử <samp> xác định đầu ra mẫu từ chương trình máy tính.
Phần tử <code> xác định một đoạn mã máy tính.
Phần tử <var> xác định một biến trong lập trình hoặc trong biểu thức toán học.
Phần tử <pre> phần tử xác định văn bản được định dạng sẵn.
HTML Computer Code Elements
<table><tr><th>Tag</th><th>Description</th></tr><tr><td><code></td><td>Defines programming code</td></tr><tr><td><kbd></td><td>Defines keyboard input</td></tr><tr><td><samp></td><td>Defines computer output</td></tr><tr><td><var></td><td>Defines a variable</td></tr><tr><td><pre></td><td>Defines preformatted text</td></tr></table></td>
<td><!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Document</title>
</head>
<body>
<p>Save the document by pressing <kbd>Ctrl + S</kbd></p>
<p><samp>File not found.
Press F1 to continue</samp></p>

<p>Programming code example:</p>
<pre>
<code>
x = 5;
y = 6;
z = x + y;
</code>
</pre>
<p>The area of a triangle is: 1/2 x <var>b</var> x <var>h</var>,
where <var>b</var> is the base,
and <var>h</var> is the</td>
<td>Save the document by pressing Ctrl + S
File not found.
Press F1 to continue
Programming code example:
x = 5;
y = 6;
z = x + y;
The area of a triangle is: 1/2 x b x h, where b is the base, and h is the vertical height.</td>
</tr>
</table>

			vertical height. </body> </html>	
29	HTML Semantics	Các yếu tố ngữ nghĩa là gì? Phần tử ngữ nghĩa mô tả rõ ràng ý nghĩa của nó đối với cả trình duyệt và nhà phát triển. ● Ví dụ về các phần tử phi ngữ nghĩa: và - Không cho biết gì về nội dung của nó. ● Ví dụ về các phần tử ngữ nghĩa: <form>, <table>, <article> - Xác định rõ ràng nội dung của nó. Các thành phần ngữ nghĩa trong HTML: Nhiều trang web chứa mã HTML như: để biểu thị điều hướng, đầu trang và chân trang. Trong HTML có một số thành phần ngữ nghĩa có thể được sử dụng để xác định các phần khác nhau của trang web:	Ví dụ về HTML <article> Element: <!DOCTYPE html> <html> <body> <h1>The article element</h1> <article> <h2>Google Chrome</h2> Google Chrome is a web browser developed by Google, released in 2008. Chrome is the world's most popular web browser today! </article> <article> <h2>Mozilla Firefox</h2> Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018. </article> <article> <h2>Microsoft Edge</h2> Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft Edge	 The article element Google Chrome Google Chrome is a web browser developed by Google, released in 2008. Chrome is the world's most popular web browser today! Mozilla Firefox Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018. Microsoft Edge Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft Edge replaced Internet Explorer.

		• <article> • <aside> • <details> • <figcaption> • <figure> • <footer> • <header> • <main> • <mark> • <nav> • <section> • <summary> • <time>  	replaced Internet Explorer. </article> </body> </html>	
30	HTML Input Types	Chương này mô tả các loại khác nhau của phần tử HTML <input>. Các loại HTML Input Types:	Ví dụ cho một số Input Types thường dùng: <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>Document</title> </head> <body> 1.Input types "text" <form>	

		• <input type="button"> • <input type="checkbox">> • <input type="color">> • <input type="date">> • <input type="datetime-local">> • <input type="email">> • <input type="file">> • <input type="hidden">> • <input type="image">> • <input type="month">> • <input type="number">> • <input type="password">> • <input type="radio">> • <input type="range">> • <input type="reset">> • <input type="search">> • <input type="submit">> • <input type="tel">> • <input type="text">> • <input type="time">> • <input type="url">> • <input type="week">>	<label for="fname">First name:</label> <input type="text" id="fname" name="fname"> <label for="lname">Last name:</label> <input type="text" id="lname" name="lname">> </form> 2.Input types "submit" <form action="/action_page.php">> <label for="fname">First name:</label> <input type="text" id="fname" name="fname" value="John"> <label for="lname">Last name:</label> <input type="text" id="lname" name="lname" value="Doe"> <input type="submit"	

			value="Submit"> </form> </div> <div> <p>3.Input types "checkbox"</p> <form> <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike"> <label for="vehicle1"> I have a bike</label> <input type="checkbox" id="vehicle2" name="vehicle2" value="Car"> <label for="vehicle2"> I have a car</label> <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat"> <label for="vehicle3"> I have a boat</label> </form> </div> <div> <p>4.Input types "button"</p> <input type="button" onclick="alert('Hello World!')"	

			value="Click Me!"> 5. The input type="url" is used for input fields that should contain a URL address: <form action="/action_page.php"> <label for="homepage">Add your homepage:</label> <input type="url" id="homepage" name="homepage"> <input type="submit" value="Submit">> </form> </body> </html>	

2.2. CSS

2.2.1. Giới thiệu về CSS

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

CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, vì HTML không được thiết kế để gắn tag để giúp định dạng trang web.

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

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

2.2.2. Bố cục và cấu trúc của một đoạn CSS

2.2.2.1. Bố cục của một đoạn CSS

Bố cục CSS thường chủ yếu dựa vào hình hộp và mỗi hộp đều chiếm những khoảng trống trên trang của bạn với các thuộc tính như:

- Padding: Gồm không gian xung quanh nội dung (ví dụ: xung quanh đoạn văn bản).
- Border: Là đường liền nằm ngay bên ngoài phần đệm.
- Margin: Là khoảng cách xung quanh bên ngoài của phần tử.

2.2.2.2. Cấu trúc của một đoạn CSS

Một đoạn CSS bao gồm các phần như thế này:

vùng chọn { thuộc tính : giá trị; thuộc tính: giá trị; .... }

Nghĩa là nó sẽ được khai báo bằng vùng chọn, sau đó các thuộc tính và giá trị sẽ nằm bên trong cặp dấu ngoặc nhọn {}. Mỗi thuộc tính sẽ luôn có một giá trị riêng, giá trị có thể là dạng số, hoặc các tên giá trị trong danh sách có sẵn của CSS. Phần giá trị và thuộc tính phải được cách nhau bằng dấu hai chấm, và mỗi một dòng khai báo thuộc tính sẽ luôn có dấu chấm phẩy ở cuối. Một vùng chọn có thể sử dụng không giới hạn thuộc tính.

Định nghĩa của các phần này như sau:

- Bộ chọn (Selector): là mẫu để chọn phần tử HTML mà bạn muốn định nghĩa phong cách. Các selector được áp dụng cho các trường hợp sau:
  - Tất cả phần tử theo một dạng cụ thể nào đó, ví dụ phần tử tiêu đề h1.
  - Thuộc tính id và class của các phần tử.
  - Các phần tử dựa vào mối liên quan với các phần tử khác trong cây phân cấp tài liệu.
- Khai báo (Declaration): Khối khai báo chứa một hoặc nhiều khai báo, phân tách với nhau bằng các dấu chấm phẩy. Mỗi khai báo gồm tên và giá trị đặc tính CSS, phân tách bằng dấu phẩy. Khai báo CSS luôn kết thúc bằng dấu chấm phẩy, khối khai báo nằm trong các dấu ngoặc móc. Trong ví dụ dưới đây, các phần tử <p> sẽ được căn giữa, chữ màu đỏ.

  p { color: red; text-align: center; }
- Thuộc tính (Properties): Những cách mà bạn có thể tạo kiểu cho một phần tử HTML. (Với trường hợp này thì color được xem là một trong những thuộc tính của phần tử p). Chính vì vậy, với CSS thì bạn chỉ cần lựa chọn thuộc tính mà chính bạn muốn tác động nhất trong bộ quy tắc của mình.

- Giá trị thuộc tính: Ở bên phải của thuộc tính sau dấu hai chấm(:), chúng ta sẽ sở hữu giá trị thuộc tính mà việc lựa chọn trong số đó sẽ xuất hiện nhiều lần để có thể cho một thuộc tính cụ thể nào đó.

2.2.3. Các thuộc tính thông dụng trong CSS

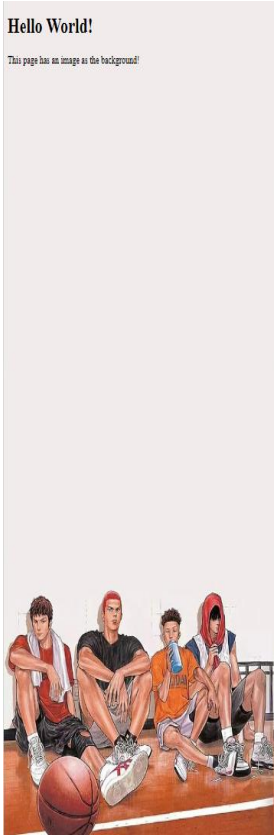
Dưới đây là các thuộc tính CSS được sử dụng nhiều khi thiết kế một Website:

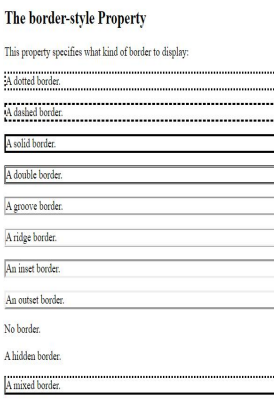
Bảng 2. 2 Các thuộc tính thông dụng trong CSS

STT	CSS	Lý thuyết	Code	Result

1	Color	Màu sắc được chỉ định bằng cách sử dụng tên màu được xác định trước hoặc các giá trị RGB, HEX, HSL, RGBA, HSLA.
		- Trong CSS, màu có thể được chỉ định bằng cách sử dụng tên màu được xác định trước
		- Trong CSS, màu sắc cũng có thể được chỉ định bằng giá trị RGB, giá trị HEX, giá trị HSL, giá trị RGBA và giá trị HSLA
			<!DOCTYPE html>
			<html>
			<body>
			 Ut wisi enim...
			<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
			
			 Same as color name "Tomato":
			
			<h1 style="background-color:rgb(255, 99, 71);">rgb(255, 99, 71)</h1>
			<h1 style="background-color:#ff6347;">#ff6347</h1>
			<h1 style="background-color:hsl(9, 100%, 64%);">hsl(9, 100%, 64%)</h1>
				Ut wisi enim...
				Hello World
				Same as color name "Tomato":
				rgb(255, 99, 71)
				#ff6347
				hsl(9, 100%, 64%)

			In addition to the predefined color names, colors can be specified using RGB, HEX, HSL, or even transparent colors using RGBA or HSLA color values. </body> </html>	

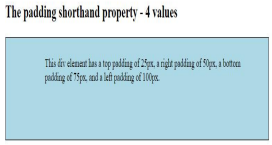
2	Background	Thuộc tính background-image chỉ định một hình ảnh để sử dụng làm nền của một phần tử. Theo mặc định, hình ảnh được lặp lại để bao phủ toàn bộ phần tử.	<!DOCTYPE html>
			<html>
			<head>
			<style>
			body {
			background-image: url("https://i.pinimg.com/564x/b8/70/34/b87034e3831eb30bf6b28b79e7aa4c8e.jpg");
			}
			</style>
			</head>
			<body>
			
			<h1>Hello World!</h1>
			
			 This page has an image as the background!
			
			</body>
			</html>
				

3	Border	Thuộc tính border CSS cho phép bạn chỉ định kiểu, chiều rộng và màu sắc của đường viền của một phần tử. Thuộc tính border-style chỉ định loại đường viền sẽ hiển thị. Cho phép các giá trị sau: dotted - Xác định đường viền có chấm. dashed - Xác định đường viền liền nét đứt. solid - Xác định đường viền liền nét kép. double - Xác định rãnh viền đôi. groove - Xác định đường viền có rãnh 3D. Hiệu ứng phụ thuộc vào giá trị đường viền màu của đường viền. ridge- Xác định đường viền có đường viền 3D. Hiệu ứng phụ thuộc vào giá trị màu đường viền. inset - Xác định đường viền chèn 3D. Hiệu ứng phụ thuộc vào giá trị màu đường viền. outset - Xác định đường viền bắt đầu 3D. Hiệu ứng phụ thuộc vào giá trị màu đường viền. none - Xác định không có đường viền. hidden - Xác định đường viền ẩn. Thuộc tính border-style có thể có từ một đến bốn giá trị (đối với đường viền trên, đường viền	<!DOCTYPE html> <html> <head> <style> p.dotted {border-style: dotted;} p.dashed {border-style: dashed;} p.solid {border-style: solid;} p.double {border-style: double;} p.groove {border-style: groove;} p.ridge {border-style: ridge;} p.inset {border-style: inset;} p.outset {border-style: outset;} p.none {border-style: none;} p.hidden {border-style: hidden;} p.mix {border-style: dotted dashed solid double;} </style> </head> <body> <h2>The border-style Property</h2> This property specifies what kind of border to display: A dotted border. A dashed border.	

		phải, đường viền dưới và đường viền bên trái).	A solid border. A double border. A groove border. A ridge border. An inset border. An outset border. No border. A hidden border. A mixed border. </body> </html>	

4	Margin	Margin được sử dụng để tạo không gian xung quanh các phần tử, bên ngoài bất kỳ đường viền xác định nào.
		Thuộc tính Margin được sử dụng để tạo khoảng trống xung quanh các phần tử, bên ngoài bất kỳ đường viền được xác định nào.
		Với CSS, bạn có toàn quyền kiểm soát lề. Có các thuộc tính để đặt lề cho mỗi cạnh của một phần tử (trên, phải, dưới và trái).
		-CSS có các thuộc tính để xác định lề cho mỗi cạnh của một phần tử:
		● margin-top
			<!DOCTYPE html>
			<html>
			<head>
			<style>
			div {
			border: 1px solid black;
			margin-top: 100px;
			margin-bottom: 100px;
			margin-right: 400px;
			margin-left: 200px;
			background-color: lightblue;
			}
			</style>
			</head>
			<body>
			
			<h2>Using individual margin
				Using individual margin properties
				Thuộc tính margin.

		• margin-right • margin-bottom • margin-left Trong đó - auto - trình duyệt tính toán mức ký quỹ - *length* - chỉ định lề bằng px, pt, cm, v.v. - % - chỉ định lề tính bằng % chiều rộng của phần tử chứa. - inherit - chỉ định rằng lề phải được kế thừa từ phần tử cha.	properties</h2> Thuộc tính margin. </body> </html>	

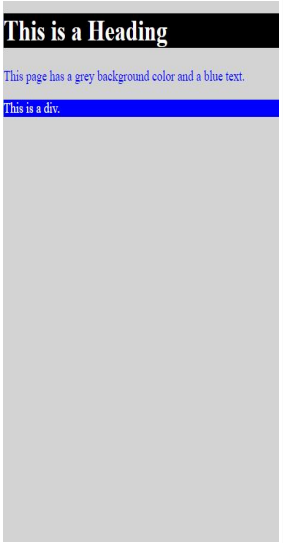
5	Padding	Thuộc tính Padding CSS được sử dụng để tạo khoảng trống xung quanh nội dung của phần tử, bên trong bất kỳ đường viền được xác định nào.
		Với CSS, bạn có toàn quyền kiểm soát phần đệm. Có các thuộc tính để đặt khoảng đệm cho mỗi cạnh của một phần tử (trên, phải, dưới và trái).
		• Padding - Individual Sides
		CSS has properties for specifying the padding for each side of an element:
		- padding-top
		- padding-right
		- padding-bottom
		- padding-left
		• Padding - Shorthand Property
		If the padding property has four values:
			<!DOCTYPE html>
			<html>
			<head>
			<style>
			div {
			border: 1px solid black;
			padding: 25px 50px 75px 100px;
			background-color: lightblue;
			}
			</style>
			</head>
			<body>
			
			<h2>The padding shorthand property - 4 values</h2>
			
			 This div element has a top padding of 25px, a right padding of 50px, a bottom padding of 75px, and
				

		padding: 25px 50px 75px 100px; - top padding is 25px - right padding is 50px - bottom padding is 75px - left padding is 100px	a left padding of 100px. </body> </html>	

6	Height	Thuộc tính height thiết lập chiều cao của một phần tử.
		Chiều cao của một phần tử không bao gồm phần đệm, đường viền hoặc lề!
		Nếu height: auto; phần tử sẽ tự động điều chỉnh chiều cao của nó để cho phép nội dung của nó được hiển thị chính xác.
		Nếu height được đặt thành một giá trị số (như pixel, (r) em, tỷ lệ phần trăm) thì nếu nội dung không vừa với chiều cao đã chỉ định, nó sẽ tràn. Cách vùng chứa xử lý nội dung tràn được xác định bởi thuộc tính tràn.
		Lưu ý: Thuộc tính chiều cao tối thiểu và chiều cao tối đa ghi đè thuộc tính chiều cao.
			<!DOCTYPE html>
			<html>
			<head>
			<style>
			div.a {
			height: auto;
			border: 1px solid black;
			}
			div.b {
			height: 50px;
			border: 1px solid black;
			}
			</style>
			</head>
			<body>
			<h1>The height Property</h1>
			
			<h2>height: auto (default)</h2>
			 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida
				The height Property
				height: auto (default)
				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. Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.
				height: 50px
				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. Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.
				ở đây ta có thể thấy đoạn text đầu tiên khi height: auto thì chiều cao của khung vừa với đoạn text còn đoạn text thứ 2 thì khung chỉ cao đúng 50px không phụ thuộc vào độ dài đoạn text.

			libero rhoncus ut. Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat. <h2>height: 50px</h2> 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. Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat. </body> </html>	

7	Outline	Outline là một đường được vẽ bên ngoài đường viền của phần tử.
		Outline là một đường được vẽ xung quanh các phần tử, BÊN NGOÀI đường viền, để làm cho phần tử "nổi bật".
		CSS có các thuộc tính Outline sau:
		• outline-style
		• outline-color
		• outline-width
		• outline-offset
		• outline
			<!DOCTYPE html>
			<html>
			<head>
			<style>
			p {
			border: 2px solid black;
			outline: #4CAF50 solid 10px;
			margin: auto;
			padding: 20px;
			text-align: center;
			}
			</style>
			</head>
			<body>
			<h2>CSS Outline</h2>
			 This element has a 2px black border and a green outline with a width of 10px.
			</body>
			</html>	

8	Text	CSS có rất nhiều thuộc tính để định dạng văn bản.
		Màu văn bản và màu nền
			<!DOCTYPE html>
			<html>
			<head>
			<style>
			body {
			background-color: lightgrey;
			color: blue;
			}
			
			h1 {
			background-color: black;
			color: white;
			}
				

			div {
			  background-color: blue;
			  color: white;
			}
			</style>
			</head>
			<body>
			
			<h1>This is a Heading</h1>
			 This page has a grey background color and a blue text.
			 This is a div.
			
			</body>
			</html>	

9	Font	Việc chọn đúng phông chữ có tác động rất lớn đến cách người đọc trải nghiệm một trang web.
		Phông chữ phù hợp có thể tạo ra bản sắc mạnh mẽ cho thương hiệu của bạn.
		Sử dụng phông chữ dễ đọc là điều quan trọng. Phông chữ thêm giá trị cho văn bản của bạn. Điều quan trọng nữa là chọn đúng màu sắc và kích thước văn bản cho phông chữ.
		Generic Font Families
		In CSS there are five generic font families:
		1. Serif fonts have a small stroke at the edges of each letter.
			<!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>
				CSS font-family
				This is a paragraph, shown in the Times New Roman font.
				This is a paragraph, shown in the Arial font.
				This is a paragraph, shown in the Lucida Console font.

		They create a sense of formality and elegance. 2. Sans-serif fonts have clean lines (no small strokes attached). They create a modern and minimalistic look. 3. Monospace fonts - here all the letters have the same fixed width. They create a mechanical look. 4. Cursive fonts imitate human handwriting. 5. Fantasy fonts are decorative/playful fonts.	<body> <h1>CSS font-family</h1> This is a paragraph, shown in the Times New Roman font. This is a paragraph, shown in the Arial font. This is a paragraph, shown in the Lucida Console font. </body> </html>	
10	Icon	Các biểu tượng có thể dễ dàng được thêm vào trang HTML của bạn bằng cách sử dụng thư viện biểu tượng. Cách đơn giản nhất để thêm biểu tượng vào trang HTML của bạn là sử dụng thư viện biểu tượng, chẳng hạn như Font Awesome. Thêm tên của lớp biểu tượng được chỉ định vào bất kỳ phần tử HTML nội tuyến nào (như <i> hoặc). Tất cả các biểu tượng trong thư viện biểu tượng bên dưới đều là các vectơ có thể mở rộng và có thể được tùy chỉnh bằng CSS	<!DOCTYPE html> <html> <head> <title>Font Awesome Icons</title> <meta name="viewport" content="width=device-width, initial-scale=1"> <script src="https://kit.fontawesome.com/a076d05399.js" crossorigin="anonymous"></script> <!--Get your own code at fontawesome.com--> </head> <body>	

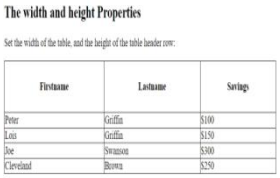
		(kích thước, màu sắc, bóng, v.v.)	<h1>Font Awesome icon library</h1> Some Font Awesome icons: Styled Font Awesome icons (size and color): </body> </html>	

11	List	HTML Lists and CSS List Properties
		Trong HTML, có hai loại danh sách chính: danh sách không có thứ tự (<ul>) - các mục danh sách được đánh dấu bằng dấu đầu dòng danh sách có thứ tự (<ol>) - các mục danh sách được đánh dấu bằng số hoặc chữ cái. Thuộc tính danh sách CSS cho phép bạn : Đặt các điểm đánh dấu mục danh sách khác nhau cho danh sách được sắp xếp Đặt các điểm đánh dấu 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 cho danh sách và mục danh sách
		Trong HTML, có hai loại danh sách chính:
		danh sách không có thứ tự (<ul>) - các mục danh sách được đánh dấu bằng dấu đầu dòng
		danh sách có thứ tự (<ol>) - các mục danh sách được đánh dấu bằng số hoặc chữ cái.
		
		Thuộc tính danh sách CSS cho phép bạn:
		- Đặt các điểm đánh dấu mục danh sách khác nhau cho danh sách được sắp xếp
		- Đặt các điểm đánh dấu
			<!DOCTYPE html>
			<html>
			<head>
			<style>
			ul.a {
			list-style-type: circle;
			}
			
			ul.b {
			list-style-type: square;
			}
			
			ol.c {
			list-style-type: upper-roman;
			}
			
			ol.d {
			list-style-type: lower-alpha;
			}
			</style>
			</head>
			<body>
			
			<h2>The list-style-type Property</h2>
			
			 Example of unordered lists:
			<ul class="a">
			  <li>Coffee</li>
			  <li>Tea</li>
			  <li>Coca Cola</li>
			</ul>
			
			<ul class="b">
			  <li>Coffee</li>
			  <li>Tea</li>
			  <li>Coca Cola</li>
				The list-style-type Property
				Example of unordered lists:
				◦ Coffee
				◦ Tea
				◦ Coca Cola
				▪ Coffee
				▪ Tea
				▪ Coca Cola
				Example of ordered lists:
				I. Coffee
				II. Tea
				III. Coca Cola
				a. Coffee
				b. Tea
				c. Coca Cola

		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 cho danh sách và mục danh sách	</ul> Example of ordered lists: <ol class="c"><li>Coffee</li> <li>Tea</li> <li>Coca Cola</li> </ol> <ol class="d"><li>Coffee</li> <li>Tea</li> <li>Coca Cola</li> </ol> </body> </html>	

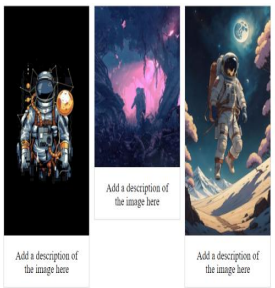
12	Table	**CSS Table Style**
		Để kiểm soát khoảng cách giữa đường viền và nội dung trong bảng, hãy sử dụng thuộc tính đệm trên các phần tử <td> và <th>
			<!-- Table Padding -->
			<!DOCTYPE html>
			<html>
			<head>
			<style>
			table, td, th {
			  border: 1px solid #ddd;
			  text-align: left;
			}
			table {
			  border-collapse: collapse;
			  width: 100%;
			}
			th, td {
			  padding: 15px;
			}
			</style>
			</head>
			<body>
			<h2>The padding Property</h2>
			
			 This property adds space between the border and the content in a table.
			
			<table>
			  <tr>
			
			<th>Firstname</th>
			
			<th>Lastname</th>
			  <th>Savings</th>
			  </tr>
			  <tr>
			  <td>Peter</td>
			  <td>Griffin</td>
				
				The padding Property
				This property adds space between the border and the content in a table.
				Firstname Lastname Savings
				Peter Griffin $100
				Lois Griffin $150
				Joe Swanson $300
				Cleveland Brown $250

			<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>	

		CSS Table Size
		Chiều rộng và chiều cao của bảng được xác định bởi thuộc tính chiều rộng và chiều cao.
			<!DOCTYPE html>
			<html>
			<head>
			<style>
			table, td, th {
			  border: 1px solid black;
			}
			
			table {
			  border-collapse: collapse;
			  width: 100%;
			}
			
			th {
			  height: 70px;
			}
			</style>
			</head>
			<body>
			
			<h2>The width and height Properties</h2>
			
			 Set the width of the table, and the height of the table header row:
			
			<table>
			  <tr>
			
			<th>Firstname</th>
			
			<th>Lastname</th>
			  <th>Savings</th>
			  </tr>
			  <tr>
			  <td>Peter</td>
			  <td>Griffin</td>
				
				The width and height Properties
				Set the width of the table, and the height of the table header row:
				Firstname Lastname Savings
				Peter Griffin $100
				Lois Griffin $150
				Joe Swanson $300
				Cleveland Brown $250

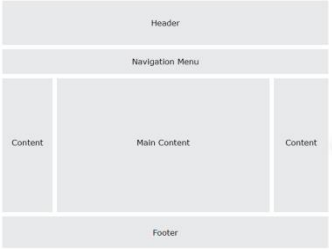
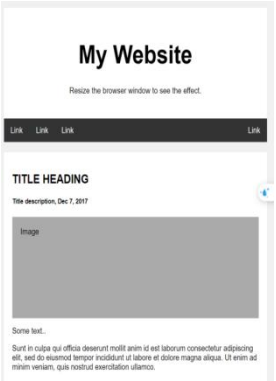
			<td>$100</td> </tr> <tr> <td>Lois</td> <td>Griffin</td> <td>$150</td> </tr> <tr> <td>Joe</td> <td>Swanson</td> <td>$300</td> </tr> <tr> <td>Cleveland</td> <td>Brown</td> <td>$250</td> </tr> </table> </body> </html>	
13	Navigation bar	Việc điều hướng dễ sử dụng là điều quan trọng đối với bất kỳ trang web nào. Với CSS, bạn có thể biến các menu HTML nhàm chán thành các thanh điều hướng đẹp mắt. **Navigation Bar = List of Links** Thanh điều hướng cần có HTML tiêu chuẩn làm cơ sở.	<!DOCTYPE html> <html> <body> <ul> <li><a href="#home">Home</a></li> <li><a href="#news">News</a></li> <li><a href="#contact">Contact</a></li>	• Home • News • Contact • About Note: We use href="#" for test links. In a real web site this would be URLs.

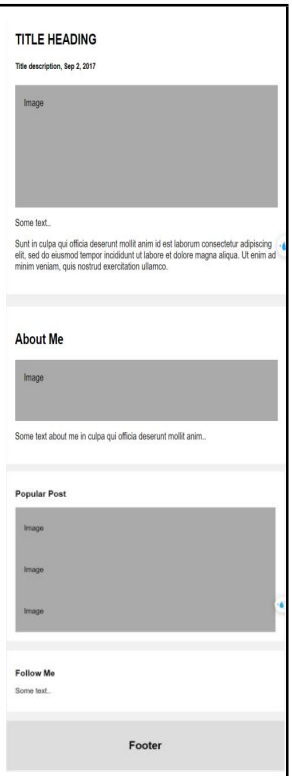
		Trong ví dụ của chúng tôi, chúng tôi sẽ xây dựng thanh điều hướng từ danh sách HTML tiêu chuẩn. Thanh điều hướng về cơ bản là một danh sách các liên kết, vì vậy việc sử dụng các phần tử <ul> và <li> là hoàn toàn hợp lý	<li><a href="#about">About </a></li> </ul> Note: We use href="#" for test links. In a real web site this would be URLs. </body> </html>	

14	Image Gallery	CSS có thể được sử dụng để tạo một thư viện hình ảnh.	<!DOCTYPE html>
			<html>
			<head>
			<style>
			div.gallery {
			 margin: 5px;
			 border: 1px solid #ccc;
			 float: left;
			 width: 180px;
			}
			div.gallery:hover {
			 border: 1px solid #777;
			}
			div.gallery img {
			 width: 100%;
			 height: auto;
			}
			div.desc {
			 padding: 15px;
			 text-align: center;
			}
			</style>
			</head>
			<body>
			
			 <a target="_blank"
				

			href="https://i.pinimg.com/564x/d0/a7/00/d0a7000ccd3b1a59fc7c1fd1b922182b.jpg"> alt="Cinque Terre" width="600" height="400"> </a> class="desc">Add a description of the image here target="_blank" href="https://i.pinimg.com/564x/02/01/cb/0201cb5a85b0bbf6783c5873c0b52acb.jpg"> alt="Forest" width="600" height="400"> </a> class="desc">Add a description of the image here target="_blank"	

			href="https://i.pinimg.com/564x/b1/e8/62/b1e862c3d269a012fc91278c85409f40.jpg"> <img src="https://i.pinimg.com/564x/b1/e8/62/b1e862c3d269a012fc91278c85409f40.jpg" alt="Northern Lights" width="600" height="400"> </a> <div class="desc">Add a description of the image here </body> </html>	

15	Website Layout	Một trang web thường được chia thành các phần đầu trang, menu, nội dung và phần chân trang:
		 
		
		Có rất nhiều thiết kế bố trí khác nhau để lựa chọn. Tuy nhiên, cấu trúc ở trên là một trong những cấu trúc phổ biến nhất và chúng ta sẽ xem xét kỹ hơn về nó trong hướng dẫn này.
			<!DOCTYPE html>
			<html>
			<head>
			<style>
			* {
			  box-sizing: border-box;
			}
			
			body {
			  font-family: Arial;
			  padding: 10px;
			  background: #f1f1f1;
			}
			
			/* Header/Blog Title */
			.header {
				

		Header
		Header thường nằm ở đầu trang web (hoặc ngay bên dưới menu điều hướng trên cùng). Nó thường chứa logo hoặc tên trang web
		
		Navigation Bar
		Thanh điều hướng chứa danh sách các liên kết để giúp khách truy cập điều hướng qua trang web của bạn
		
		Content
		Bố cục trong phần này thường phụ thuộc vào người dùng mục tiêu. Bố cục phổ biến nhất là một (hoặc kết hợp chúng) trong số các bố cục sau:
		1 cột: (thường được sử dụng cho trình duyệt trên thiết bị di động)
		2 cột: (thường được sử dụng cho máy tính bảng và máy tính xách tay)
		Bố cục 3 cột: (chỉ được sử dụng cho máy tính để bàn)
		
		Unequal Columns
		Nội dung chính là phần lớn nhất và quan trọng nhất trên trang web của bạn. Điều phổ biến là độ rộng cột không bằng nhau nên phần lớn không gian được dành cho nội dung chính. Nội dung phụ (nếu có) thường được sử dụng làm điều hướng thay thế
			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;
			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;
				

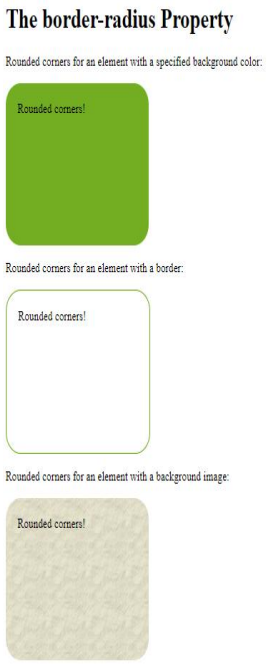
		hoặc để chỉ định thông tin liên quan đến nội dung chính. Thay đổi độ rộng theo ý muốn, chỉ nhớ rằng tổng chiều rộng phải lên tới 100%.
		
		Footer
		Chân trang được đặt ở cuối trang của bạn. Nó thường chứa thông tin như bản quyền và thông tin liên hệ.
			width: 75%;
			}
			/* Right column */
			.rightcolumn {
			float: left;
			width: 25%;
			background-color: #f1f1f1;
			padding-left: 20px;
			}
			
			/* Fake image */
			.fakeimg {
			background-color: #aaa;
			width: 100%;
			padding: 20px;
			}
			
			/* 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>
			
			  <h1>My Website</h1>
			  <p>Resize the browser window to	

			see the effect.</p> <a href="#">Link</a> <a href="#">Link</a> <a href="#">Link</a> <a href="#" style="float:right">Li nk</a> <div class="leftcolumn"><div class="card"><h2>TITLE HEADING</h2> <h5>Title description, Dec 7, 2017</h5> <div class="fakeimg" style="height:200px;" >Image <p>Some text..</p> <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>	

			   <h2>TITLE HEADING</h2> <h5>Title description, Sep 2, 2017</h5> Image Some text.. Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco. <div class="card"> <h2>About Me</h2> Image Some text about me in culpa qui officia deserunt mollit anim.. </div>	

			<h3>Popular Post</h3> Image Image Image <h3>Follow Me</h3> Some text.. <h2>Footer</h2> </body> </html>	

16	Rounded Corners	Với thuộc tính border-radius CSS, bạn có thể cung cấp cho bất kỳ phần tử nào "các góc tròn".	<!DOCTYPE html>
			<html>
			<head>
			<style>
			#rcorners1 {
			  border-radius: 25px;
			  background: #73AD21;
			  padding: 20px;
			  width: 200px;
			  height: 150px;
			}
			
			#rcorners2 {
			  border-radius: 25px;
			  border: 2px solid #73AD21;
			  padding: 20px;
			  width: 200px;
			  height: 150px;
			}
			#rcorners3 {
			  border-radius: 25px;
			  background: url(paper.gif);
			  background-position: left top;
			  background-repeat: repeat;
			  padding: 20px;
			  width: 200px;
			  height: 150px;
			}
			</style>
			</head>
			<body>
			
			<h1>The border-radius Property</h1>
			<p>Rounded corners for an element with a specified background
				

			color: Rounded corners! Rounded corners for an element with a border: Rounded corners! Rounded corners for an element with a background image: Rounded corners! </body> </html>	

17	Multiple Backgrounds	CSS cho phép bạn thêm nhiều hình nền cho một phần tử, thông qua thuộc tính hình nền.
		Các hình nền khác nhau được phân tách bằng dấu phẩy và các hình ảnh được xếp chồng lên nhau, trong đó hình ảnh đầu tiên ở gần người xem nhất.
		Ví dụ sau có hai ảnh nền, ảnh đầu tiên là một bông hoa (căn chỉnh ở dưới cùng bên phải) và ảnh thứ hai là nền giấy (căn chỉnh ở góc trên bên trái).
		
			<!DOCTYPE html>
			<html>
			<head>
			<style>
			#example1 {
			background-image: url(img_flwr.gif), url(paper.gif);
			background-position: right bottom, left top;
			background-repeat: no-repeat, repeat;
			padding: 15px;
			}
			</style>
			</head>
			<body>
			
			<h1>Multiple Backgrounds</h1>

CSS Advanced Background Properties

CSS Advanced Background Properties

Property	Description
background	A shorthand property for setting all the background properties in one declaration
background-clip	Specifies the painting area of the background
background-image	Specifies one or more background images for an element
background-origin	Specifies where the background image(s) is/are positioned
background-size	Specifies the size of the background image(s)

CSS Background Size

Thuộc tính CSS Background-size cho phép bạn chỉ định kích thước của hình nền.

Kích thước có thể được chỉ định theo độ dài, tỷ lệ phần trăm hoặc bằng cách sử dụng một trong hai từ khóa: chứa hoặc che.

Ví dụ sau thay đổi kích thước hình nền thành nhỏ hơn nhiều so với hình ảnh gốc (sử dụng pixel):

Lorem Ipsum Dolor

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.

CSS background-origin Property

Thuộc tính nguồn gốc nền CSS chỉ định vị trí của hình nền.

Thuộc tính nhận ba giá trị khác nhau:

border-box - hình nền bắt đầu từ góc trên bên trái của đường viền

padding-box - (mặc định) hình nền bắt đầu từ góc trên bên trái của cạnh đệm

content-box - hình nền

```
<p>The following div element has two background images:</p>

<div id="example1">
 <h1>Lorem Ipsum Dolor</h1>
 <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
 <p>Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
</div>

</body>
</html>
```

		bắt đầu từ góc trên bên trái của nội dung		
		CSS background-clip Property Thuộc tính background-clip chỉ định vùng vẽ của nền. Thuộc tính nhận ba giá trị khác nhau: border-box - (mặc định) nền được vẽ ra mép ngoài của đường viền padding-box - the background is painted to the outside edge of the padding content-box - nền được vẽ trong hộp nội dung		

18	Buttons	Chúng ta có thể tạo màu, hình dạng, viền … cho các nút	<!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>
			<body>	

			<h2>Shadow Buttons</h2> Use the box-shadow property to add shadows to the button: <button class="button button1">Shadow Button</button> <button class="button button2">Shadow on Hover</button> </body> </html>	
			<!DOCTYPE html> <html> <head> <style> .button { background-color: #04AA6D; /* Green */ border: none; color: white; padding: 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 16px; margin: 4px 2px; cursor: pointer;	**Rounded Buttons** Add rounded corners to a button with the border-radius property: 2px 4px 8px 12px 50%

			} .button1 {border-radius: 2px; padding: 16px;background-color: #04AA6D;} .button2 {border-radius: 4px; padding: 32px 16px;background-color: #008CBA;} .button3 {border-radius: 8px; padding: 14px 40px;background-color: #555555;} .button4 {border-radius: 12px;background-color: #e7e7e7; color: black;} .button5 {border-radius: 50%;padding: 10px 24px;background-color: #f44336;} </style> </head> <body> <h2>Rounded Buttons</h2> Add rounded corners to a button with the border-radius property: <button class="button button1">2px</button> <button class="button button2">4px</button> <button class="button	

			button3">8px</butto n> <button class="button button4">12px</butt on> <button class="button button5">50%</butto n> </body> </html>	

19	Object-position	Thuộc tính được sử dụng để chỉ định cách định vị <img> hoặc <video> trong vùng chứa của nó.
		Example:
		The Image
		Look at the following image from Paris, which is 400x300 pixels:
		
			<!DOCTYPE html>
			<html>
			<head>
			<style>
			
			#img1 img {
			width: 200px;
			height: 300px;
			object-fit: cover;
			}
			#img2 img {
			width: 200px;
			height: 300px;
			object-fit: cover;
			object-position: 80%
			100%;
			}
			</style>
			</head>
				Using object-fit: cover
				
				Using object-position
				Here we will use the object-position property to position the image so that the great old building is in center:

		a. Chúng tôi sử dụng object-fit: cover; để giữ tỷ lệ khung hình và điền vào kích thước nhất định. Tuy nhiên, hình ảnh sẽ được cắt bớt cho phù hợp, như sau: b. Giả sử phần hình ảnh được hiển thị không được định vị như chúng ta mong muốn. Để định vị hình ảnh, chúng ta sẽ sử dụng thuộc tính object-position. Ở đây chúng ta sẽ sử dụng thuộc tính object-position để định vị hình ảnh sao cho tòa nhà cổ vĩ đại nằm ở trung tâm:	<body> <h2>Using object-fit: cover</h2> <img src="paris.jpg" alt="Paris" width="400" height="300"> <h2>Using object-position</h2> Here we will use the object-position property to position the image so that the great old building is in center: <img src="paris.jpg" alt="Paris" width="400" height="300"> </body> </html>	

20	Multiple Columns	Bố cục nhiều cột CSS cho phép dễ dàng xác định nhiều cột văn bản
		Thuộc tính nhiều cột CSS có các thuộc tính nhiều cột sau:
		● column-count
		● column-gap
		● column-rule-style
		● column-rule-width
		● column-rule-color
		● column-rule
		● column-span
		● column-width
			<!DOCTYPE html>
			<html>
			<head>
			<style>
			.newspaper {
			column-count: 3;
			column-gap: 40px;
			column-rule: 1px solid lightblue;
			}
			</style>
			</head>
			<body>
			<h1>Use the column-rule Shorthand Property</h1>
			
			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
				
				Use the column-rule Shorthand Property
				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. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit
				praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

			ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. </body> </html>	

2.3. Ngôn ngữ lập trình Javascript

2.3.1. Giới thiệu về Javascript

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web.

Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web. Ví dụ: khi duyệt internet, bất cứ khi nào bạn thấy quảng cáo quay vòng dạng hình ảnh, menu thả xuống nhấp để hiển thị hoặc màu sắc phần tử thay đổi động trên trang web cũng chính là lúc bạn thấy các hiệu ứng của JavaScript.

2.3.2 Tính năng chính của JavaScript:

Đa nền tảng: JavaScript có thể chạy trên nhiều trình duyệt web khác nhau như Chrome, Firefox, Safari và Edge, cũng như trên nhiều hệ điều hành như Windows, macOS và Linux.

Ngôn ngữ dễ học: JavaScript có cú pháp gần giống với các ngôn ngữ lập trình khác như C và Java, làm cho việc học và sử dụng nó trở nên dễ dàng đối với các lập trình viên mới.

Cú pháp linh hoạt: JavaScript hỗ trợ nhiều phong cách lập trình từ lập trình hàm cho đến lập trình hướng đối tượng, cũng như các tính năng như cấu trúc điều khiển, hàm callback, và promise.

Tích hợp dễ dàng: JavaScript có thể tích hợp vào các trang web HTML và các ứng dụng web thông qua các tập tin script và thư viện bên ngoài như jQuery, React, Angular và Vue.js.

Xử lý sự kiện: JavaScript cho phép xử lý các sự kiện như click chuột, nhập liệu từ bàn phím và tương tác người dùng khác để tạo ra trải nghiệm tương tác trên web.

Xử lý dữ liệu động: JavaScript có thể tạo, đọc và thay đổi dữ liệu trên trang web một cách động, cho phép tạo ra các ứng dụng web phong phú và linh hoạt.

2.3.3 Ưu và nhược điểm của JavaScript:

Ưu điểm:

Dễ học và sử dụng: JavaScript có cú pháp đơn giản và dễ tiếp cận, làm cho việc học và phát triển ứng dụng trở nên dễ dàng hơn.

Thiết kế đa nền tảng: JavaScript có thể chạy trên nhiều nền tảng và trình duyệt khác nhau, giúp tăng khả năng phổ cập của ứng dụng.

Cộng đồng phát triển mạnh mẽ: JavaScript có một cộng đồng lập trình viên lớn và đa dạng, cung cấp nhiều tài nguyên, thư viện và framework hữu ích.

Phát triển nhanh chóng: JavaScript cho phép phát triển ứng dụng một cách nhanh chóng và linh hoạt, giúp giảm thời gian và chi phí phát triển.

Nhược điểm:

Bảo mật: JavaScript có thể dễ bị tấn công bởi các lỗ hổng bảo mật như cross-site scripting (XSS) và cross-site request forgery (CSRF) nếu không được lập trình cẩn thận.

Hiệu suất: JavaScript có thể ảnh hưởng đến hiệu suất của trang web nếu không được viết tối ưu, đặc biệt là khi xử lý các tác vụ nặng như xử lý dữ liệu lớn hoặc tải tài nguyên nhiều.

Phụ thuộc vào trình duyệt: Một số tính năng JavaScript có thể không hoạt động đồng nhất trên tất cả các trình duyệt, làm cho việc phát triển và kiểm tra ứng dụng trở nên phức tạp hơn.

2.3.4 Ứng dụng phổ biến sử dụng JavaScript

JavaScript đã trở thành một phần không thể thiếu trong phát triển web và có mặt trong nhiều ứng dụng phổ biến trên internet. Dưới đây là một số ứng dụng phổ biến sử dụng JavaScript:

Ứng dụng Web Động (Dynamic Web Applications): JavaScript là lựa chọn hàng đầu cho việc phát triển các ứng dụng web động. Từ các trang web tương tác đơn giản đến các ứng dụng web phức tạp như các trang mạng xã hội, JavaScript giúp tạo ra trải nghiệm tương tác mạnh mẽ và phong phú cho người dùng.

Single Page Applications (SPA): SPA là các ứng dụng web mà một trang web duy nhất được tải vào trình duyệt và các nội dung mới được tải bằng JavaScript khi người dùng tương tác. Các framework như React, Angular và Vue.js giúp phát triển SPA một cách dễ dàng và hiệu quả.

Ứng dụng di động và đám mây (Mobile and Cloud Applications): JavaScript không chỉ được sử dụng trên web mà còn trong phát triển ứng dụng di động và đám mây. Với các framework như React Native và Ionic, nhà phát triển có thể sử dụng JavaScript để xây dựng ứng dụng di động chạy trên nhiều nền tảng. Đồng thời, JavaScript cũng có thể được sử dụng trong việc phát triển các ứng dụng đám mây sử dụng các dịch vụ như AWS Lambda và Google Cloud Functions.

2.4. Google Firebase

2.4.1. Giới thiệu về Google Firebase

Google Firebase là một nền tảng phát triển ứng dụng được Google cung cấp, cung cấp một bộ công cụ và dịch vụ toàn diện giúp các nhà phát triển dễ dàng xây dựng, cải thiện và phát triển ứng dụng di động và web. Firebase cung cấp các dịch vụ bao gồm cơ sở dữ liệu, xác thực, phân tích, lưu trữ, và các tính năng khác giúp giảm bớt công việc liên quan đến backend và tập trung vào việc phát triển frontend và trải nghiệm người dùng.

2.4.2. Cách thức hoạt động

Google Firebase hoạt động như một nền tảng dịch vụ đám mây cung cấp các API và SDK cho các nhà phát triển tích hợp vào ứng dụng của họ. Dưới đây là một số thành phần chính và cách chúng hoạt động:

- Firebase Realtime Database: Một cơ sở dữ liệu NoSQL thời gian thực, cho phép dữ liệu được lưu trữ và đồng bộ hóa giữa người dùng trong thời gian thực.
- Cloud Firestore: Một cơ sở dữ liệu NoSQL linh hoạt và mở rộng được xây dựng để tự động đồng bộ hóa dữ liệu giữa người dùng và thiết bị.
- Firebase Authentication: Cung cấp các phương thức xác thực người dùng dễ dàng như email/password, Google, Facebook, Twitter, và nhiều nhà cung cấp khác.
- Firebase Analytics: Cung cấp các công cụ phân tích mạnh mẽ để theo dõi hành vi người dùng và hiểu rõ hơn về cách người dùng tương tác với ứng dụng.
- Firebase Cloud Messaging (FCM): Cho phép gửi thông báo và tin nhắn tới người dùng trên các nền tảng Android, iOS, và web.
- Firebase Storage: Lưu trữ và phục vụ nội dung người dùng như ảnh, video, và các file khác một cách an toàn và mở rộng.

Các dịch vụ Firebase tích hợp tốt với nhau và với các dịch vụ đám mây khác của Google như Google Cloud Platform, giúp các nhà phát triển xây dựng và quản lý ứng dụng một cách dễ dàng và hiệu quả.

2.4.3. Ưu nhược điểm của Google Firebase

Ưu điểm:

- Thời gian Thực: Firebase Realtime Database và Cloud Firestore cung cấp khả năng đồng bộ dữ liệu thời gian thực giữa các thiết bị và người dùng.
- Dễ sử dụng: Firebase cung cấp các SDK và API dễ dàng tích hợp vào ứng dụng, giúp giảm bớt công việc phát triển backend.

- Tính năng Toàn diện: Cung cấp một loạt các dịch vụ từ cơ sở dữ liệu, xác thực, phân tích, đến lưu trữ và tin nhắn, giúp nhà phát triển có mọi thứ cần thiết từ một nền tảng duy nhất.
- Mở rộng: Firebase dễ dàng mở rộng theo nhu cầu sử dụng của ứng dụng, hỗ trợ từ những ứng dụng nhỏ đến những hệ thống lớn.
- Hỗ trợ Đa nền tảng: Hỗ trợ các ứng dụng trên Android, iOS, và web, cho phép đồng bộ hóa dữ liệu và trải nghiệm người dùng liên tục trên các thiết bị khác nhau.

Nhược điểm:

- Giới hạn Miễn phí: Các dịch vụ miễn phí của Firebase có giới hạn về số lượng người dùng, dung lượng lưu trữ và số lượng yêu cầu, cần phải nâng cấp lên gói trả phí khi vượt qua giới hạn.
- Khả năng Tùy chỉnh: Một số dịch vụ của Firebase có thể không cung cấp khả năng tùy chỉnh và kiểm soát chi tiết như khi sử dụng các giải pháp backend tự xây dựng.
- Khóa Định nhà cung cấp (Vendor Lock-In): Việc sử dụng Firebase có thể gây ra tình trạng phụ thuộc vào Google cho các dịch vụ backend, khó khăn khi muốn chuyển đổi sang nền tảng khác.
- Hiệu năng với Dữ liệu Lớn: Với những ứng dụng có lượng dữ liệu rất lớn, việc truy vấn và quản lý dữ liệu có thể gặp khó khăn và không hiệu quả như các giải pháp cơ sở dữ liệu truyền thống.

2.4.4. Ứng dụng phổ biến sử dụng Google Firebase

Firebase đã được sử dụng rộng rãi trong nhiều ứng dụng phổ biến, bao gồm:

- Duolingo: Ứng dụng học ngôn ngữ sử dụng Firebase để quản lý dữ liệu người dùng và đồng bộ hóa nội dung học tập trên các thiết bị.

- Shazam: Ứng dụng nhận diện âm nhạc sử dụng Firebase để gửi thông báo đẩy và quản lý thông tin người dùng.
- The New York Times: Sử dụng Firebase để gửi thông báo tin tức thời gian thực đến người dùng.

CHƯƠNG 3: YÊU CẦU THIẾT KẾ

3.1. Yêu cầu hệ thống

3.1.1. Yêu cầu người dùng

- Điều khiển thiết bị và theo dõi từ xa
  + Người dùng có thể bật tắt các thiết bị như đèn, quạt, hoặc máy điều hòa.
  + Người dùng có thể theo dõi nhiệt độ và độ ẩm thay đổi theo thời gian bằng biểu đồ.
- Giao diện thân thiện và dễ sử dụng
  + Giao diện web phải thân thiện với người dùng và dễ dàng sử dụng trên cả máy tính và thiết bị di động.

3.1.2. Yêu cầu kỹ thuật

- **Thiết bị ESP32:**
  + Bộ vi xử lý dual-core với tốc độ tối thiểu 160 MHz.
  + Kết nối Wifi 2.4GHz, hỗ trợ chuẩn 802.11 b/g/n.
  + Cổng I/O: Ít nhất 10 cổng GPIO để kết nối với cảm biến và các module điều khiển (relay).
  + Nguồn điện: Được cấp nguồn ổn định 5V qua nguồn adapter hoặc cổng USB.
- **Cảm biến:**
  + Cảm biến Nhiệt độ và Độ ẩm: DHT11 hoặc DHT22, kết nối qua cổng GPIO của ESP32.
  + Cảm biến Khí gas: MQ-2 hoặc tương đương, kết nối qua cổng ADC của ESP32.

3.1.3. Chức năng hệ thống

- Hệ thống có các chức năng sau:

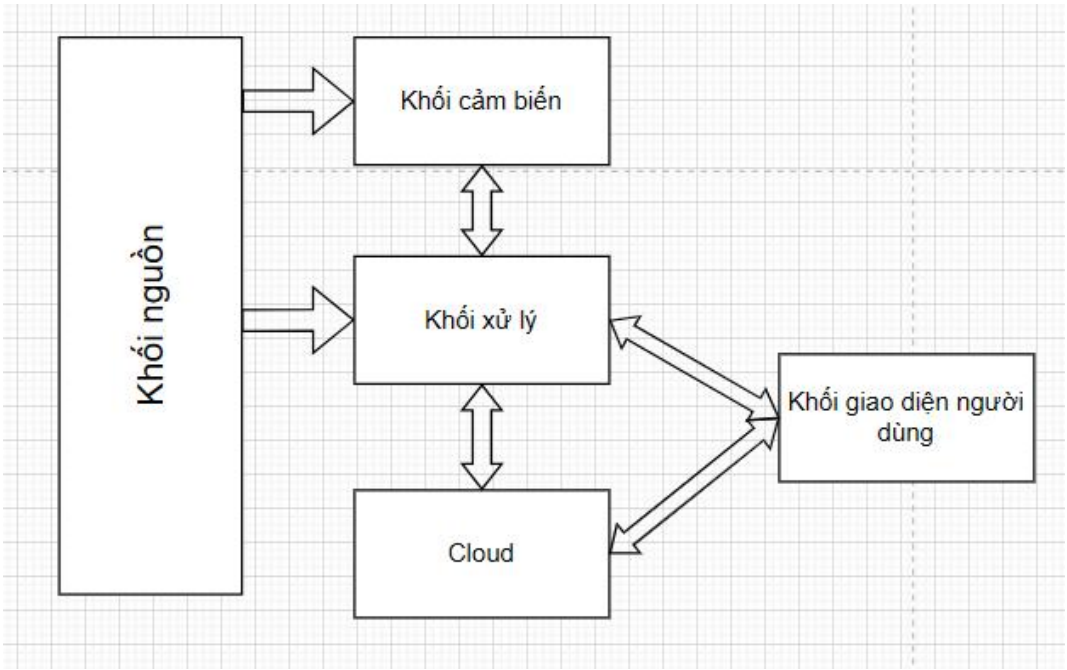
+ Hiển thị thông tin giới thiệu về Smart home và nhóm tác giả.

+ Hiển thị nhiệt độ, độ ẩm, khí gas lên website.

+ Bật tắt các thiết bị như đèn, quạt, điều hòa bằng website.

CHƯƠNG 4: PHẦN CỨNG

4.1 Sơ đồ đặc tả hệ thống



Hình 4. 1 Sơ đồ đặc tả hệ thống

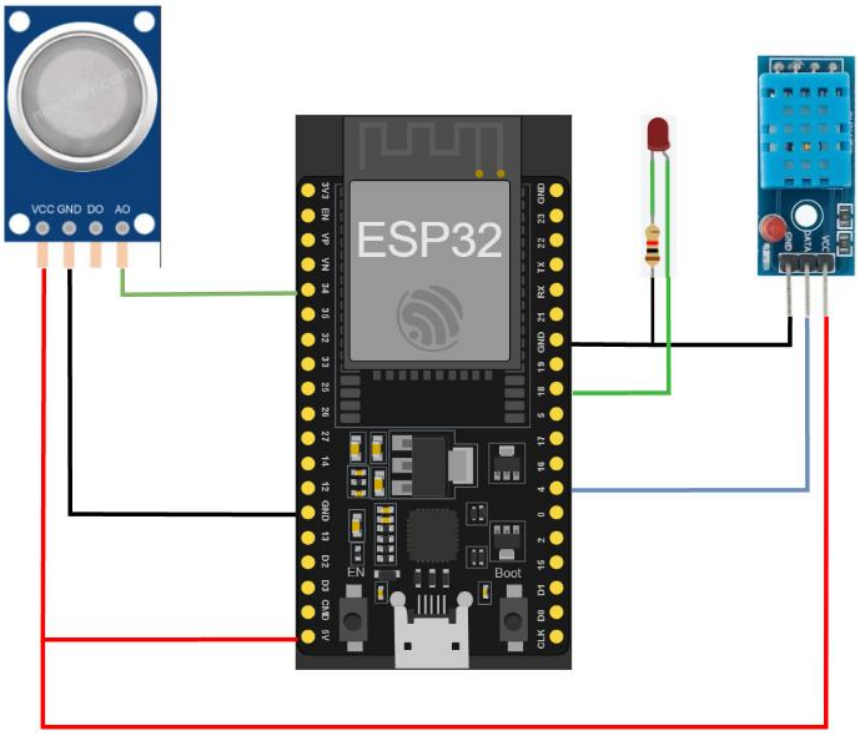
Khối nguồn: Cung cấp nguồn điện 5V và dòng điện tối thiểu 2A cho khối xử lý trung tâm và khối cảm biến.

Khối cảm biến: Chứa các thiết bị cảm biến dùng để đo lường các yếu tố môi trường như nhiệt độ, độ ẩm, và khí Gas

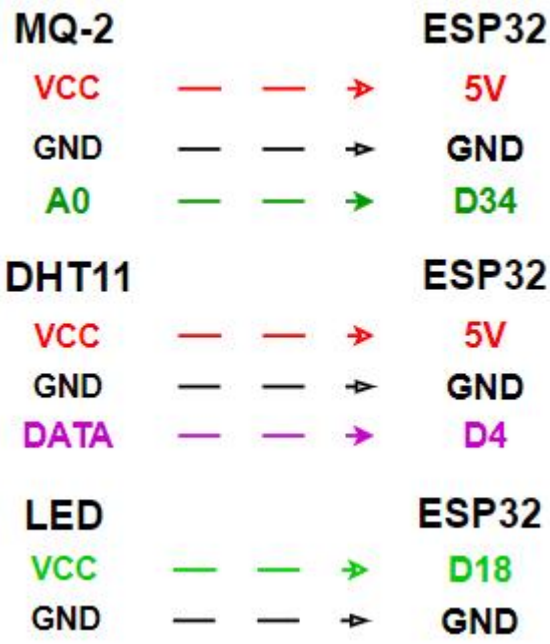
Khối xử lý trung tâm: Sử dụng vi điều khiển để xử lý dữ liệu cho phép thu thập thông tin về các giá trị về nhiệt độ, độ ẩm và khí gas trong nhà.

Khối giao diện người dùng: là giao diện mà người dùng sử dụng để tương tác với hệ thống của bạn thông qua Internet nhằm trực tiếp điều khiển các chế độ của khối vấn hành theo mong muốn.

4.2 Sơ đồ kết nối



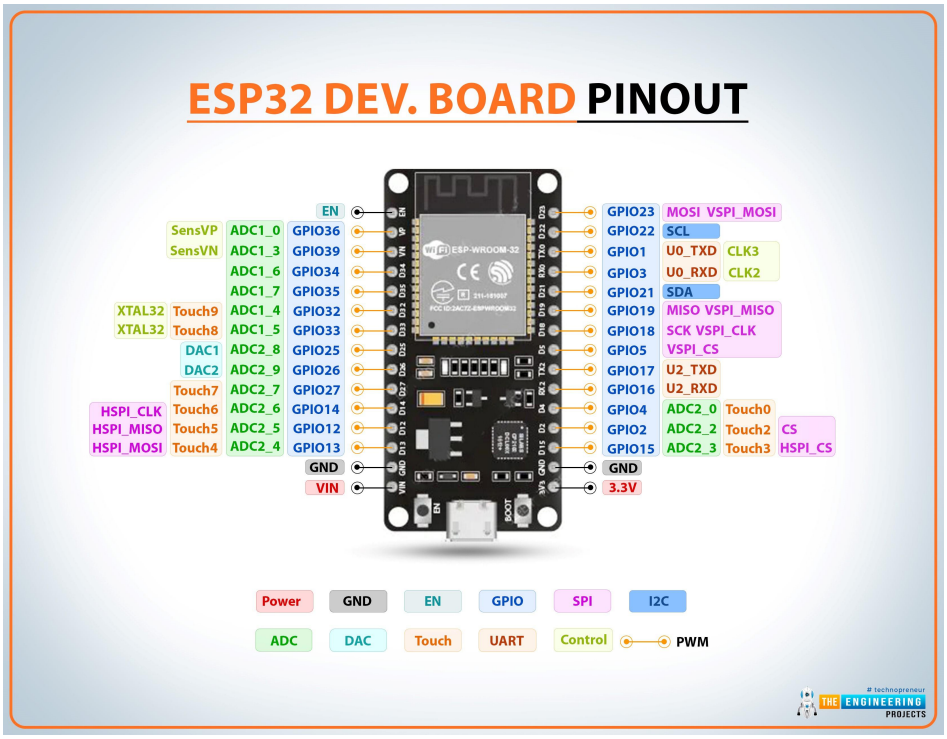
Hình 4. 2 Sơ đồ mạch nối dây



Hình 4. 3 Sơ đồ kết nối

4.3 Giới thiệu phần cứng

4.3.1. Vi điều khiển ESP32



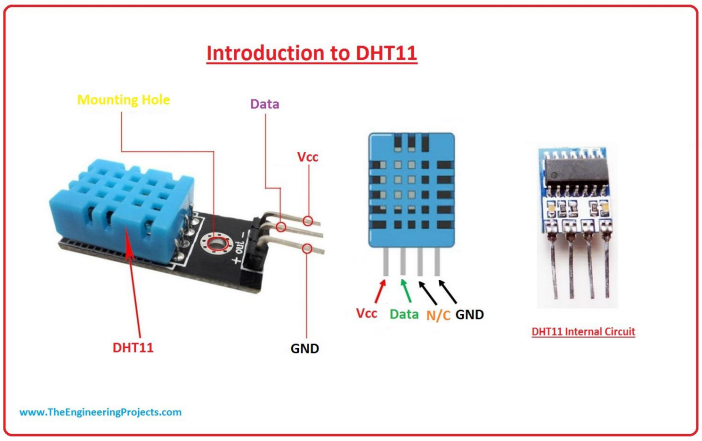
Hình 4. 4 Sơ đồ chân ESP-32

ESP32 là Bộ vi điều khiển Hệ thống trên Chip (SoC) chi phí thấp của Espressif Systems, nhà phát triển của SoC ESP8266 nổi tiếng. Nó là sản phẩm kế thừa của SoC ESP8266 và có cả biến thể lõi đơn và lõi kép của Bộ vi xử lý Xtensa LX6 32 bit của Tensilica với Wi-Fi và Bluetooth tích hợp.

Điểm hay của ESP32, giống như ESP8266 là các thành phần RF tích hợp như Bộ khuếch đại công suất, Bộ khuếch đại thu tiếng ồn thấp, Công tắc ăng-ten, Bộ lọc và RF Balun. Điều này làm cho việc thiết kế phần cứng xung quanh ESP32 trở nên rất dễ dàng vì bạn cần rất ít thành phần bên ngoài.

Một điều quan trọng khác cần biết về ESP32 là nó được sản xuất bằng công nghệ 40 nm năng lượng cực thấp của TSMC. Vì vậy, việc thiết kế các ứng dụng chạy bằng pin như thiết bị đeo, thiết bị âm thanh, thiết bị giám sát trẻ em, đồng hồ thông minh, v.v., sử dụng ESP32 sẽ rất dễ dàng.

4.3.2. Cảm biến nhiệt độ, độ ẩm DHT11

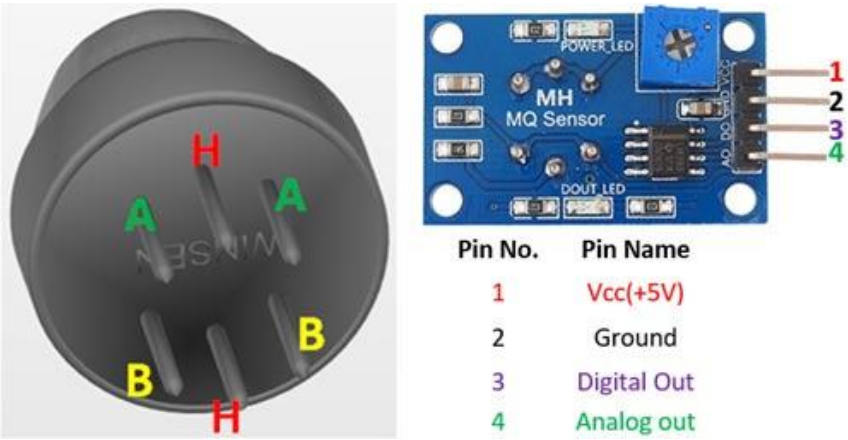


Hình 4. 5 Sơ đồ chân của cảm biến nhiệt độ độ ẩm DHT11

DHT11 là một cảm biến kỹ thuật số giá rẻ để cảm nhận nhiệt độ và độ ẩm. Cảm biến này có thể dễ dàng giao tiếp với bất kỳ bộ vi điều khiển vi nào như Arduino, Raspberry Pi, ... để đo độ ẩm và nhiệt độ ngay lập tức.

Cảm biến DHT11 bao gồm một phần tử cảm biến độ ẩm điện dung và một điện trở nhiệt để cảm nhận nhiệt độ. Tụ điện cảm biến độ ẩm có hai điện cực với chất nền giữ ẩm làm chất điện môi giữachúng. Thay đổi giá trị điện dung xảy ra với sự thay đổi của các mức độ ẩm. IC đo, xử lý các giá trịđiện trở đã thay đổi này và chuyển chúng thành dạng kỹ thuật số.

4.3.3. Cảm biến khí Gas MQ-2



Hình 4. 6 Sơ đồ chân của cảm biến khí gas MQ-2

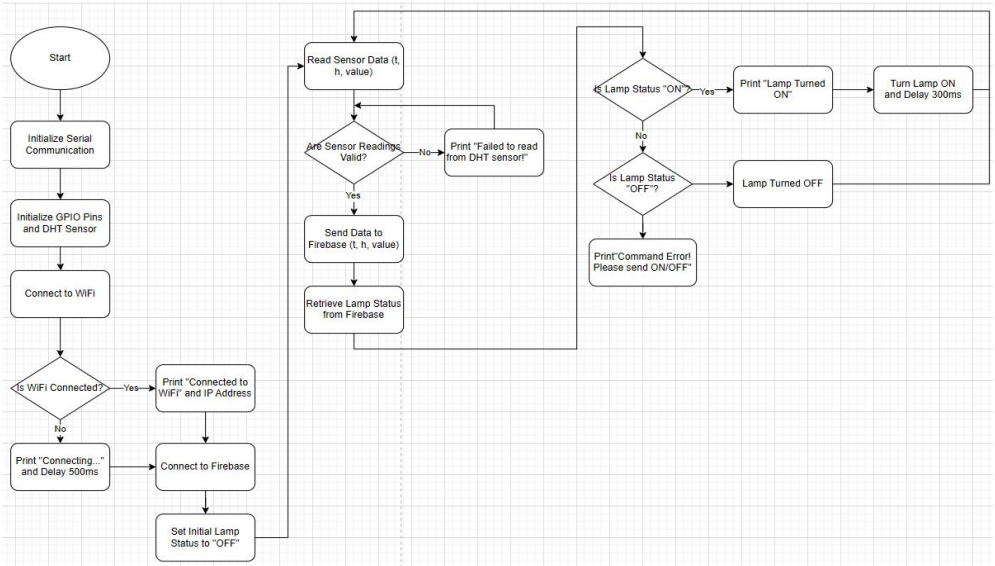
Cảm biến khí MQ2 là cảm biến điện tử dùng để cảm nhận nồng độ các loại khí trong không khí như LPG, propan, metan, hydro, rượu, khói và carbon monoxide.

Cảm biến khí MQ2 còn được gọi là cảm biến hóa học. Nó chứa một vật liệu cảm biến có điện trở thay đổi khi tiếp xúc với khí. Sự thay đổi giá trị điện trở này được sử dụng để phát hiện khí.

MQ2 là cảm biến khí loại bán dẫn oxit kim loại. Nồng độ khí trong khí được đo bằng mạng phân áp có trong cảm biến. Cảm biến này hoạt động ở điện áp DC 5V. Nó có thể phát hiện các loại khí có nồng độ trong khoảng từ 200 đến 10000ppm.

CHƯƠNG 5: PHẦN MỀM

5.1 Lưu đồ giải thuật



Hình 5. 1 Lưu đồ giải thuật của hệ thống

Ở lưu đồ trên có 2 giai đoạn đó là giai đoạn khởi tạo và giai đoạn vòng lặp:

Giai đoạn Khởi tạo

1. **Start**: Quá trình bắt đầu.

2. **Initialize Serial Communication**: Khởi tạo giao tiếp serial để phục vụ cho việc gỡ lỗi và theo dõi.

3. **Initialize GPIO Pins and DHT Sensor**: Thiết lập các chân GPIO và khởi tạo cảm biến DHT để đọc nhiệt độ và độ ẩm.

4. **Connect to WiFi**: Cố gắng kết nối với mạng WiFi được chỉ định.

   - **Is WiFi Connected?**

   - **No**: Nếu WiFi chưa kết nối, in ra "Connecting..." và chờ 500 mili giây trước khi thử lại.

   - **Yes**: Nếu WiFi đã kết nối, in ra "Connected to WiFi" và địa chỉ IP, sau đó kết nối với Firebase.

5. **Connect to Firebase**:

- Khởi tạo kết nối Firebase với thông tin host và auth.
- Thiết lập trạng thái ban đầu của đèn (Lamp) là "OFF" trên Firebase.

Giai đoạn Vòng lặp

1. **Read Sensor Data**:

- Đọc nhiệt độ (t) từ cảm biến DHT.
- Đọc độ ẩm (h) từ cảm biến DHT.
- Đọc mức khí gas (value) từ cảm biến MQ2.

2. **Check Sensor Readings**:

- **Are Sensor Readings Valid?**
- **No**: Nếu dữ liệu không hợp lệ (NaN), in ra "Failed to read from DHT sensor!" và kết thúc vòng lặp hiện tại.
- **Yes**: Nếu dữ liệu hợp lệ, gửi dữ liệu lên Firebase.

3. **Send Data to Firebase**:

- Gửi nhiệt độ (t) lên Firebase.
- Gửi độ ẩm (h) lên Firebase.
- Gửi mức khí gas (value) lên Firebase.

4. **Retrieve Lamp Status from Firebase**:

- Lấy trạng thái hiện tại của đèn (Lamp) từ Firebase.

5. **Control Lamp Based on Firebase Data**:

- **Is Lamp Status "ON"?**
  + **Yes**: Nếu trạng thái đèn là "ON", in ra "Lamp Turned ON" và bật đèn (Lamp ON).
  + **No**: Nếu trạng thái đèn không là "ON", kiểm tra tiếp.
- **Is Lamp Status "OFF"?**
  + Yes: Nếu trạng thái đèn là "OFF", in ra "Lamp Turned OFF" và tắt đèn (Lamp OFF).
  + No: Nếu trạng thái đèn không là "ON" hoặc "OFF", in ra "Command Error! Please send ON/OFF".

6. **Delay 300ms**: Đợi 300 mili giây trước khi bắt đầu vòng lặp tiếp theo.

5.2 Giao diện kết quả

Dưới đây là giao diện của trang web ngôi nhà thông minh bao gồm: giới thiệu ngôi nhà thông minh, thu thập và quản lý các thông số dữ liệu các thiết bị trong ngôi nhà.

- Website bao gồm:

+ Giao diện chính: Bao gồm phần Header, Slider, Nav.

+ Header: Có Logo, thanh điều hướng sang trang Home hoặc trang About.

+ Slider: Hình ảnh và thông tin về các lợi ích của Smart Home

+ Nav: biểu đồ hiển thị các chỉ số chất lượng không khí, và các nút nhấn bật tắt các vật dụng trong nhà.

+ Đồ thị giám sát: Dữ liệu được đẩy từ Firebase qua, giúp cập nhật các thông số của từng chỉ số.

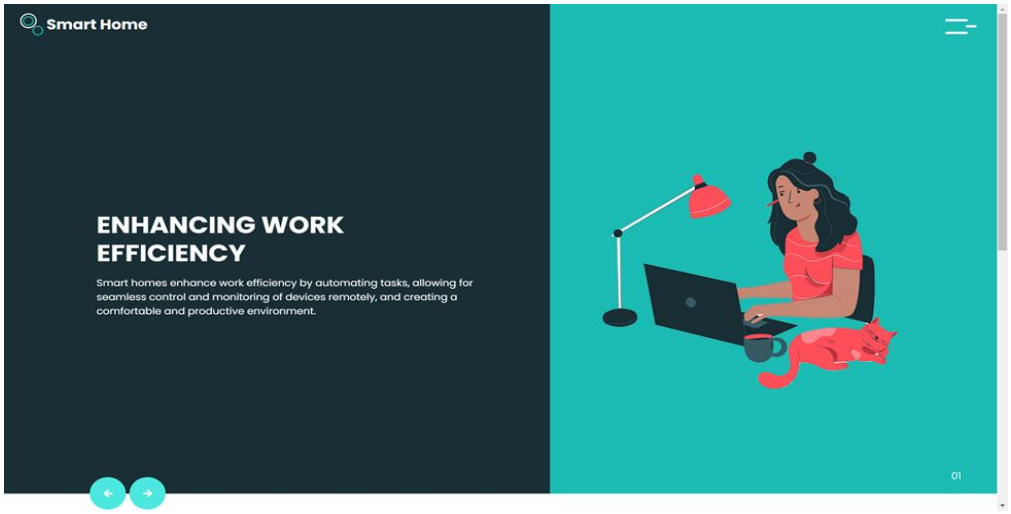
+ Trang điều hướng About: Giới thiệu về thông tin trường học và thông tin liên hệ của các thành viên.

CHƯƠNG 6: KẾT QUẢ

6.1. Kết quả giao diện website

6.1.1. Giao diện web khi mới khởi tạo

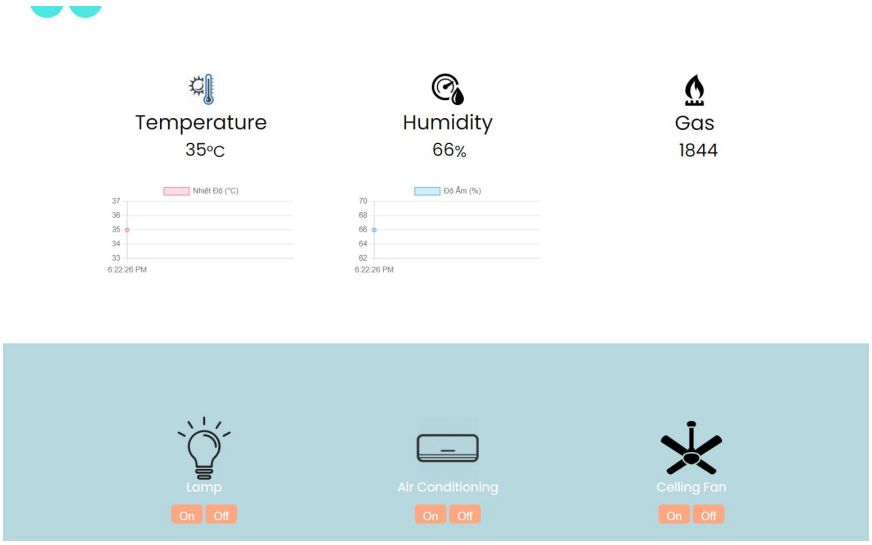
Đây là giao diện của trang web khi mới khởi tạo:



Hình 6. 1 Slide giới thiệu về Smart Homes

Hình bên trên là các slide giới thiệu về các lợi ích của nhà thông minh.

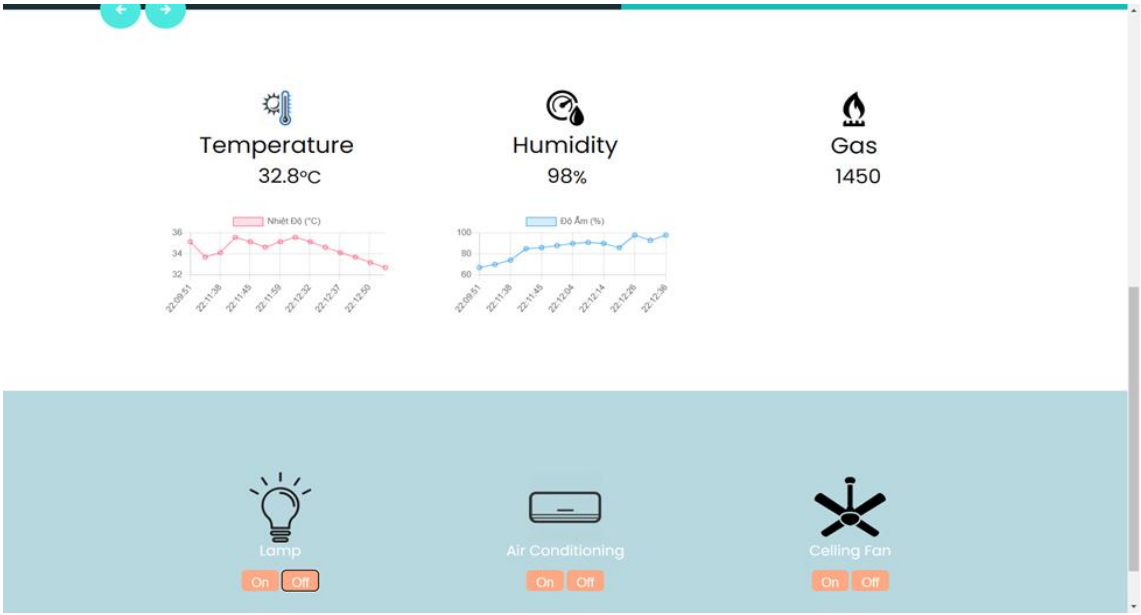
Ở bên dưới sẽ có các chức năng như giám sát nhiệt độ, độ ẩm, khí gas hay là bật tắt đèn, điều hòa và quạt từ xa.



Hình 6. 2 Các chức năng giám sát và điều khiển của hệ thống nhà thông minh

6.1.2. Kết quả khi đọc các giá trị cảm biến

Đây là kết quả cập nhật trên Website khi thu thập được từ cảm biến:



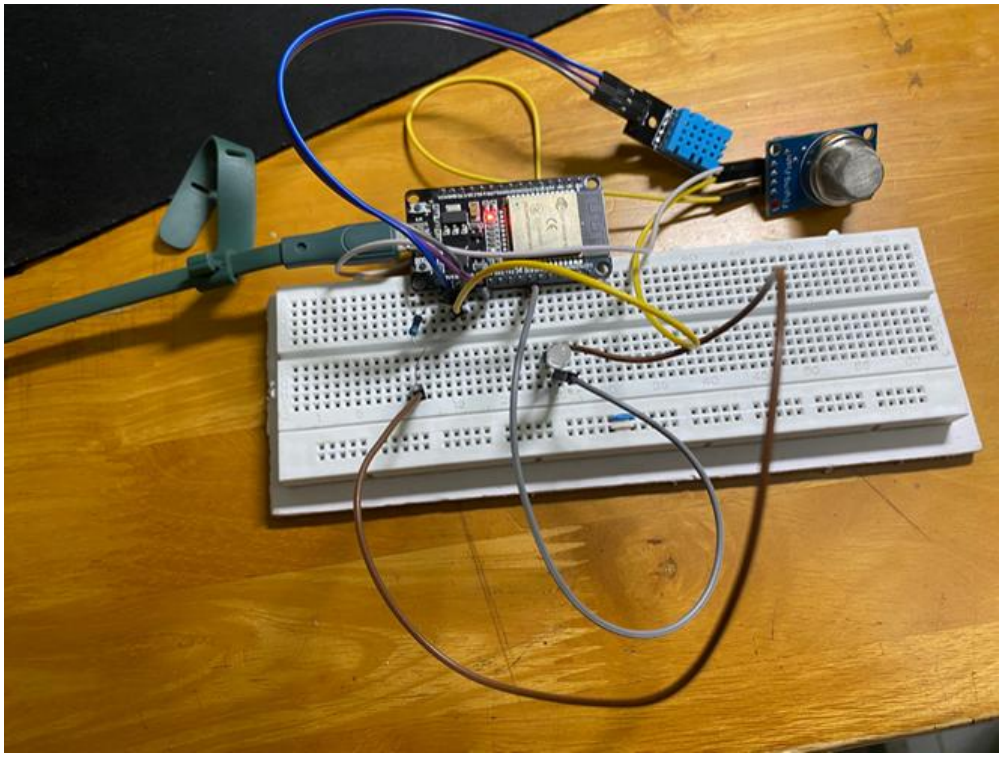
Hình 6. 3 Kết quả đọc được từ các cảm biến hiển thị lên website

Đây là kết quả cập nhật trên Firebase khi thu thập được từ cảm biến:

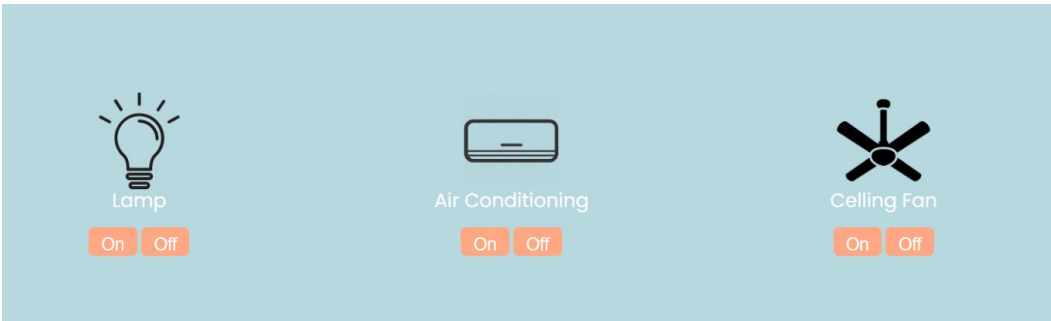
Hình 6. 4 Kết quả đọc được từ các cảm biến hiển thị lên firebase

6.1.3. Kết quả khi sử dụng nút nhấn cho các thiết bị trong nhà

Ban đầu đèn sẽ tắt như hình sau:

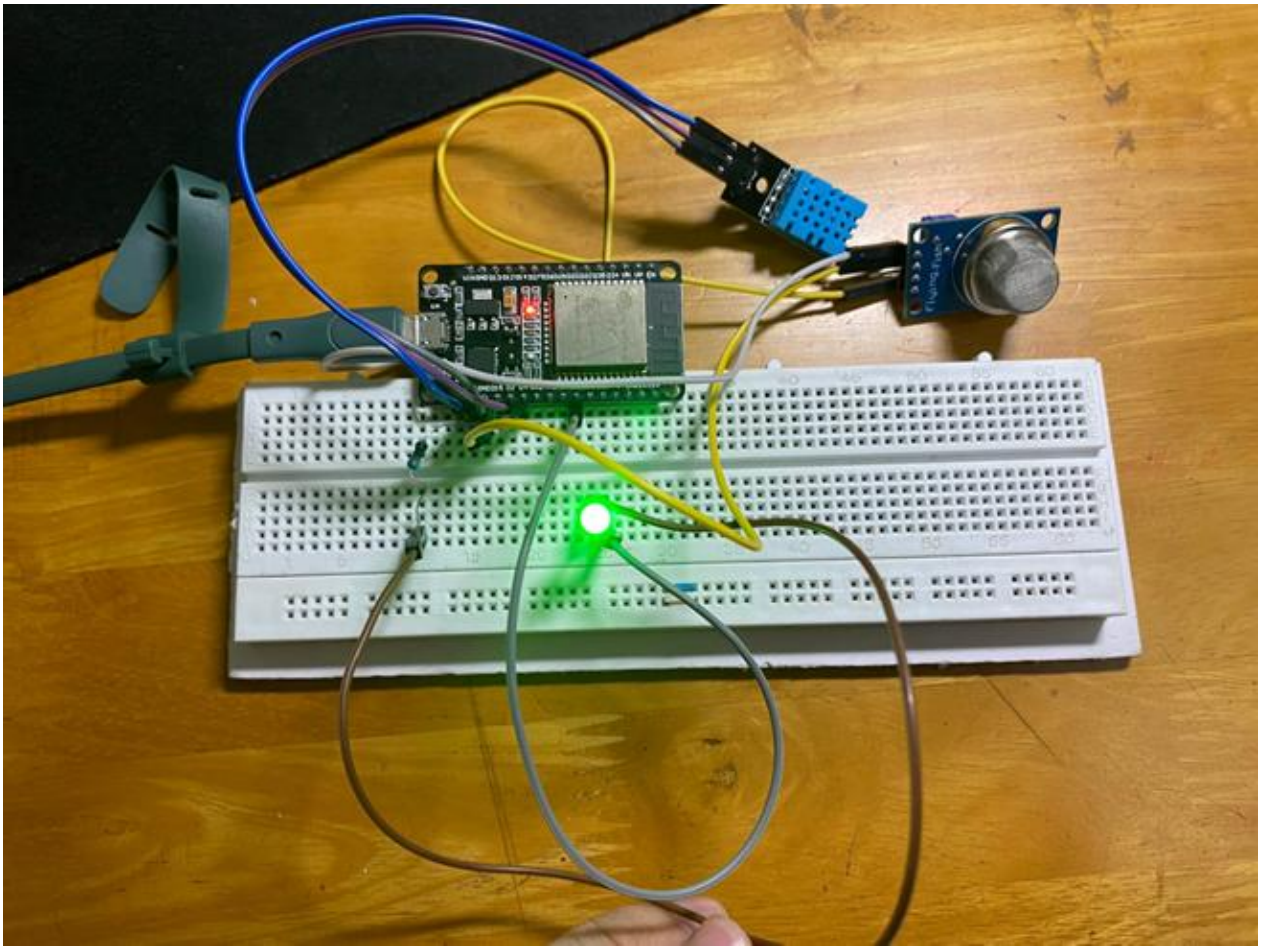


Hình 6. 5 Phần cứng khi chưa bật đèn



Hình 6. 6 Giao diện website khi chưa mở các thiết bị

Khi nhấn nút On ở bóng đèn trên giao diện thì phần đèn sẽ được bật ở thực tế:



Hình 6. 7 Phần cứng sau khi bật đèn

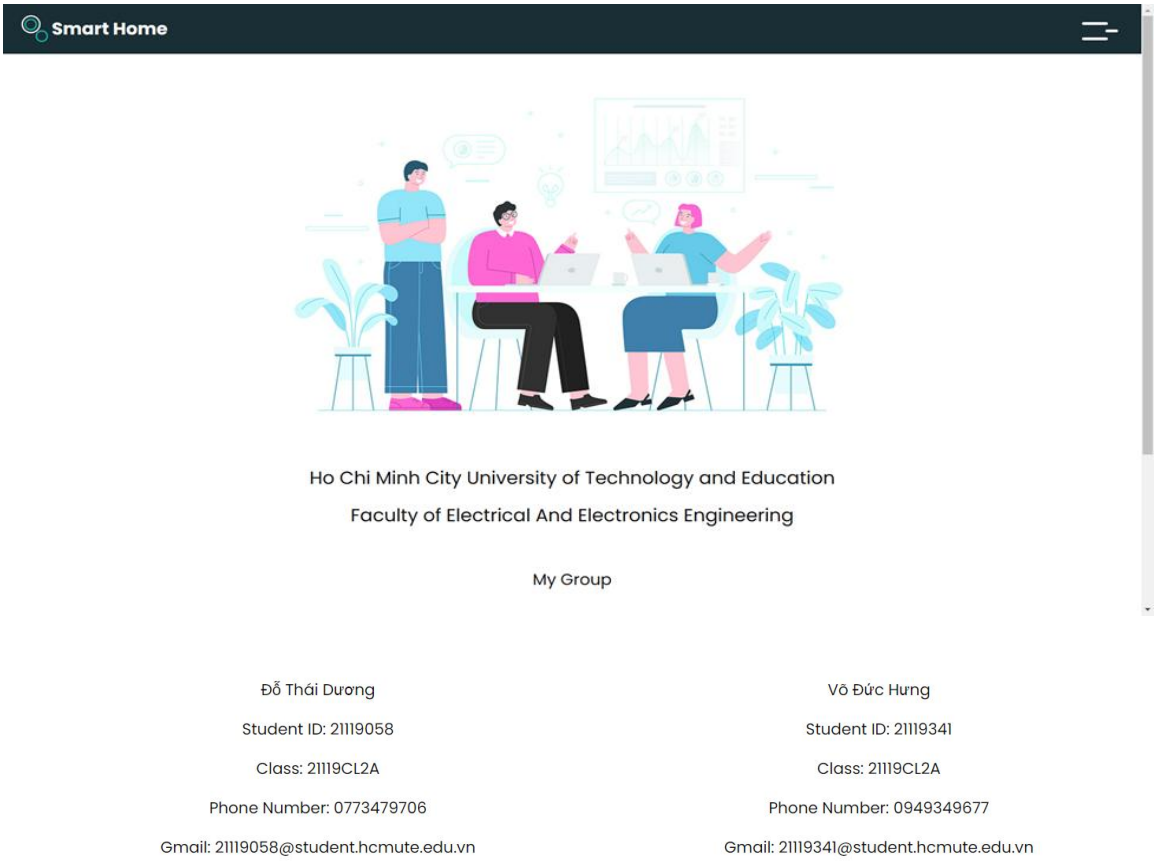
Hình 6. 8 Giao diện sau khi đã bật các thiết bị

6.1.4. Kết quả giao diện trong trang thông tin "About"

Ở đây web của nhóm tác giả có 1 thanh menu gồm Home, và About.

Hình 6. 9 Thanh menu của giao diện

Khi nhấn vào About thì giao diện sẽ chuyển tiếp như bên dưới:



Hình 6. 10 Giới thiệu về nhóm ở trang giao diện About

CHƯƠNG 7: LINK YOUTUBE

Link youtube:

Link dẫn code: Link

Hình ảnh nhóm trong quá trình làm việc:

Hình 7. 1 Hình ảnh nhóm

TÀI LIỆU THAM KHẢO

[1]	P. M. Tien, Ky nang dinh huong noi dung nghien cuu khoa hoc, Tp. HCM: ĐH SPKT Tp. HCM, 2021.
[2]	N. C. Đức, Kỹ năng lập trình C lần 2, Tp.HCM: NXB ĐH SPKT Tp.HCM, 2021.
[3]	Blog, T., & Blog, T. (2023, March 1). *CSS là gì?* TopDev. https://topdev.vn/blog/css-la-gi/
[4]	HTML: HyperText Markup Language | MDN. (2024, April 25). MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/HTML
[5]	Blog, T., & Blog, T. (2024, March 5). HTML là gì? TopDev. https://topdev.vn/blog/html-la-gi/#cau-truc-mot-doan-html
[6]	Mắt Bão. (2021, December 26). *Firebase là gì? Giải pháp lập trình không cần Backend từ Google*. Trung Tâm Hỗ Trợ Kỹ Thuật | MATBAO.NET. https://wiki.matbao.net/firebase-la-gi-giai-phap-lap-trinh-khong-can-backend-tu-google/