**BÁO CÁO CUỐI KỲ MÔN IOT PHẦN WEBSITE**

1. **Ngôn ngữ HTML, CSS và JS để thực hiện tạo website** 
   1. **HTML – HyperTextMarkup Language**
   2. **HTML là gì?**

HTML là từ viết tắt của **Hyper Text Markup Language** được sử dụng để tạo các trang web và ứng dụng web.

* **Hyper Text:** Hay siêu văn bản đơn giản có nghĩa là “Văn bản trong văn bản”. Một văn bản có một liên kết bên trong nó, là một siêu văn bản. Bất cứ khi nào bạn nhấp vào một liên kết đưa bạn đến một trang web mới, bạn đã nhấp vào một siêu văn bản. Siêu văn bản là một cách để liên kết hai hoặc nhiều trang web (Tài liệu HTML) với nhau.
* **Markup Language:** hay ngôn ngữ đánh dấu là ngôn ngữ máy tính được sử dụng để áp dụng các quy ước về bố cục và định dạng cho tài liệu văn bản. Ngôn ngữ đánh dấu làm cho văn bản tương tác và năng động hơn. Nó có thể biến văn bản thành hình ảnh, bảng biểu, liên kết.
* **Trang web:** Trang web là một tài liệu thường được viết bằng HTML và được dịch bởi trình duyệt web. Một trang web có thể được xác định bằng cách nhập một URL. Một trang Web có thể thuộc loại tĩnh hoặc động. Chỉ với sự trợ giúp của HTML, chúng ta có thể tạo các trang web tĩnh.

Do đó, HTML là ngôn ngữ đánh dấu được sử dụng để tạo các trang web hấp dẫn với sự trợ giúp của kiểu dáng và có định dạng đẹp trên trình duyệt web. Một tài liệu HTML được tạo thành từ nhiều thẻ HTML và mỗi thẻ HTML chứa nội dung khác nhau.

A screenshot of a computer program

Description automatically generated

*Hình 1.1. Ví dụ về code HTML trong trang web của nhóm “Nhà thông minh”*

* 1. **Cách hoạt động của HTML**

Để xây dựng một trang HTML, người coder phải tạo một file với đuôi .html trong phần mềm VSCODE để thực hiện tạo một trang web. Sau đó, tệp này được tải lên Trang web Toàn cầu, trang này sẽ hiển thị giao diện trực tiếp của tệp.

Để mở tệp HTML từ máy tính của mình, hãy truy cập vào File Explorer, nhấp chuột phải vào tệp và chọn “Mở bằng” từ trình đơn để khởi động trong trình duyệt ưa thích của mình, ví dụ như: Google Chrome, Cốc cốc, MicroSoft Edge. Ngoài ra cũng có thể khởi động trình duyệt của mình, nhấn Ctrl + O để mở menu “mở” và nhấp đúp chuột vào tệp html của mình.

Các trình duyệt sử dụng các phần tử HTML để giải thích nội dung của trang và hiển thị chúng một cách chính xác trên màn hình theo cách gọn gàng và có định dạng.

A screenshot of a computer

Description automatically generated

*Hình 1.2. Sau khi mở trang HTML ở trên* ***ứng dụng Microsoft Edge***

* 1. **Cấu trúc của HTML**

Các thành phần chính của trang HTML là các thẻ và phần tử. Các thẻ bắt đầu và kết thúc bằng dấu ngoặc nhọn hoặc dấu “nhỏ hơn” và “lớn hơn”. Các chữ cái giữa chính được gọi là nội dung phần tử.

Các thành phần dưới đây tạo nên cấu trúc cơ bản của một trang HTML (theo thứ tự xuất hiện của chúng)

* **Khai báo loại tài liệu (DTD):** ***<!DOCTYPE html>*** xuất hiện ở đầu hoặc trên cùng của trang HTML. Nó cho trình duyệt biết phiên bản HTML nào đã được sử dụng để tạo trang.
* **Phần tử gốc HTML: *<html>***, được viết bên dưới DTD, hoạt động giống như “thùng chứa chính” chứa tất cả các phần tử khác. Nó có thể chỉ định ngôn ngữ của tài liệu HTML, ví dụ <html lang=” en-US”> có nghĩa là trang được viết bằng tiếng Anh – Mỹ
* **Head:** Vị trí nằm ở giữa cấu trúc HTML và chứa siêu dữ liệu mô tả thông tin về trang. Head bao gồm:
* **<title>:** hoặc chủ đề tổng thể của trang web. Nó tách biệt nhưng phải khớp với thẻ tiêu đề xuất hiện trong phần nội dung
* **<style>:** Xác định cách các thành phần sẽ hiển thị trong trình duyệt. Điều này bao gồm màu của tiêu đề, căn chỉnh văn bản, màu nền nội dung …
* **<link>:** Cho biết các tài nguyên (tức là một trang web khác hoặc biểu định kiểu bên ngoài) được liên kết với trang HTML
* **<meta>:** chứa các từ khoá, tác giả và mô tả trang
* **<base>:** đề cập đến URL mặc định
* **Body:** Là phần chính của HTML chứa thông tin mà trình duyệt hiển thị trên màn hình. Body bao gồm:
* Tiêu đề trang web, logo, điều hướng chính và thanh tìm kiếm.
* Nội dung chính bao gồm tiêu đề hoặc tiêu đề bài viết, nội dung bài viết, ngày đăng, tác giả…
* Các thanh hiển thị tiện ích con và điều hướng thứu cấp, chẳng hạn nhưu lưu trữ theo thể loại hoặc ngày
* Chân trang cung cấp thông tin liên hệ, liên kết xã hội, bản quyền và điều hướng cấp ba.

A screen shot of a computer program

Description automatically generated

*Hình 1.3. Hình ảnh cấu trúc một HTML*

* 1. **Vai trò của HTML**

HTML có nhiệm vụ xây dựng cấu trúc siêu văn bản trên một website, hay khai báo các tập tin kỹ thuật số như hình ảnh, video, nhạc.

Ngoài ra, nó còn được dùng để phân chia các đoạn văn bản, heading, links, blockquotes, …

Đồng thời thực hiện các công việc sau:

* Thêm tiêu đề, định dạng văn bản, ngắt dòng điều khiển.
* Tạo danh sách, nhấn mạnh văn bản, tạo ký tự đặc biệt, chèn hình ảnh và tạo liên kết
* Góp phần xây dựng bảng, điều khiển một số kiểu mẫu

1. **CSS – Cascading Style Sheets**
   1. **CSS là gì?**

CSS là viết tẳt của cụm từ Cascading Style Sheeets, đây là ngôn ngữ để tìm và định dạng miêu tả lại các phần tử được tạo ra bởi ngôn ngữ Markup cấu trúc HTML. Để nói một cách ngắn gọn, CSS là ngôn ngữ dùng để tạo nên màu sắc, phong cách cho trang web.

Chúng ta có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử, cấu trúc trên website như việc tạo ra các đoạn văn bản, các tiêu đề, … thì ngôn ngữ CSS sẽ giúp chúng ta có thể định dạng “style” cho các phần tử HTML đó như thay đổi bố cục, màu sắc trang, màu chữ, font chữ, cấu trúc …

* 1. **Cách hoạt động của CSS**

**Bước 1:** Trình duyệt sẽ tải HTML đã tạo bởi file .html

**Bước 2:** Trình duyệt sẽ chuyển đổi HTML thành DOM (Mô hình đối tượng tài liệu). DOM đại diện cho tài liệu trong bộ nhớ máy tính

**Bước 3:** Sau đó, trình duyệt sẽ tìm nạp hầu hết các tài nguyên được liên kết với tài liệu HTML, chẳng hạn như hình ảnh và video được nhúng và CSS được liên kết với JavaScript được xử lý sau trong quá trình này.

**Bước 4:** Trình duyệt phân tích cú pháp CSS đã tìm nạp và sắp xếp các quy tắc khác nhau theo loại của chúng thành các nhóm các nhau. Dựa trên các bộ chọn mà nó tìm thấy, trình duyệt sẽ tìm ra các quy tắc nên được áp dụng cho các nodes trong DOM và đính kèm kiểu cho chúng theo yêu cầu

**Bước 5**: Cây kết xuất được bố trí trong cấu trúc mà nó sẽ xuất hiện sau khi các quy tắc đã được áp dụng cho nó

**Bước 6:** Sau tất cả, hiển thị trực quan của trang sẽ được hiển thị trên màn hình

A diagram of a diagram

Description automatically generated

*Hình 1.4. Sơ đồ hoạt động của CSS*

* 1. **Cấu trúc của CSS**

Bố cục của một đoạn CSS thường dựa trên các hình hộp, trong đó mỗi hộp chiếm không gian trên trang và các các thuộc tính như:

* Padding (Vùng đệm): Gồm không gian xung quanh nội dung.
* Border (Đường viền): Là đường viền nằm ngay bên ngoài phần đệm.
* Margin (Lề): Là khoảng cách xung quanh bên ngoài của phần tử.

A diagram of a rectangular object with a rectangular object in the middle

Description automatically generated with medium confidence

*Hình 1.5. Hình ảnh về bố cục của CSS*

Về cơ bản cấu trúc của một CSS bao gồm các phần tử sau:

* Phần thứ nhất: Vùng chọn {
* Phần thứ hai: Thuộc tính:
* Phần thứ ba: Giá trị;
* Phần thứ tư: } (Được nằm trong cặp dấu {})

Các phần này có ý nghĩa như sau:

**Vùng chọn là Selector:** Tên của phần tử HTML kích hoạt bộ quy tắc và thực hiện lựa chọn phần tử đã được tạo kiểu. Từ đó, bạn có thể tạo kiểu cho các phần tử khác chỉ bẳng cách thay đổi bộ chọn

**Xác định thuộc tính là Declaration:** Xác định các thuộc tính của phần tử để định dạng.

**Thuộc tính:** Là cách mà coder tạo kiểu cho phần tử HTML

**Giá trị thuộc tính:** Ở bên phải của thuộc tính sau dấu hai chấm (:) là giá trị thuộc tính và các lựa chọn xuất hiện nhiều làn để cho phép bạn chỉ định một thuộc tính cụ thể.

A screenshot of a computer program

Description automatically generated

*Hình 1.6. Hình ảnh cấu trúc một CSS*

* 1. **Vai trò của CSS**

CSS sẽ giúp xây dựng nên “bộ mặt” hoàn hảo cho một website, đồng thời cũng mang đến cho nó phần giao diện “ưa nhìn” và trông lôi cuốn hơn trong quá trình giao tiếp.

Ngoài ra, CSS giúp webmaster xác định được phong cách và định nghĩa nội dung web

Thông qua đó, coder có thể thực hiện nhanh chóng các công việc sau:

* Tạo style và định dạng cho những nội dung được thể hiện dưới dạng văn bản của HTML
* Tiết kiệm thời gian và công sức của coder nhờ khả năng điều khiển định dạng
* Phân biết cách hiển thị của trang web và nội dung của trang qua việc căn chỉnh bố cục màu sắc cũng như font chữ.

1. **JS – JavaScript** 
   1. **JS là gì?**

JavaScript là ngôn ngữ lập trình website phổ biến hiện nay, nó được tích hợp và nhúng vào HTML giúp website trở nên sống động hơn. JavaScript đóng vai trò như là một phần của trang web, thực thi cho phép Client-side script từ phía người dùng cũng như phía máy chủ (Nodejs) tạo ra các trang web động.

* 1. **Cách hoạt động của JS**

**Đối với JavaScript phía máy khách (Client – Side)**

Công cụ JavaScript được tích hợp sẵn bên trong hầu hết các trình duyệt web phổ biến.

* Trình duyệt thực hiện tải trang web khi có người truy cập.
* Trình duyệt chuyển đổi trang web đó và tất cả các phần tử của nó (button, nhãn dán, hộp thả xuống,) thành dạng cấu trúc dữ liệu gọi là DOM (tạm dịch: Mô hình Đối tượng Tài liệu).
* Công cụ JavaScript của trình duyệt thực hiện chuyển mã JavaScript thành mã byte.
* Khi có sự kiện xảy ra, ví dụ như nhấp vào button, việc thực thi code block JavaScript liên kết sẽ được kích hoạt.
* Công cụ JavaScript diễn giải mã byte và tạo thay đổi đối với DOM.
* Trình duyệt hiển thị DOM mới.

**Đối với JavaScript phía máy chủ (Server – Side)**

Công cụ JavaScript nằm trực tiếp trên server. JavaScript phía máy chủ có thể thực hiện:

* Truy cập dữ liệu
* Thực hiện các tính toán logic khác nhau
* Phản hồi các sự kiện khác nhau do hệ điều hành của máy chủ kích hoạt
  1. **Cấu trúc của JS**

Cấu trúc hàm cơ bản:

**Function functionName([parameter1], [parameter2]){**

**statement1;**

**statement2;**

**}**

Trong đó: function là từ khoá bắt buộc

functionName là tên của hàm mà mình muốn đặt

([parameters1], [parameters2]): là danh sách các tham số của hàm

Thân hàm bao gồm các câu lệnh nằm trong cặp cấu ngoặc {}

A screen shot of a computer code

Description automatically generated

*Hình 1.7. Hình ảnh cấu trúc một JS*

* 1. **Vai trò của JS**

JavaScript còn được biết đến là một ngôn ngữ lập trình cho phép các nhà phát triển web thiết kế ra các trang web có khả năng tương tác.

Nhờ có sự hỗ trợ của JavaScript, mà các công việc trở nên đơn giản và dễ dàng hơn khi thực hiện như:

* Bắt đầu chỉnh sửa với thư viện ảnh, nhằm mục đích tạo nên một bố cục chuyển động.
* Hỗ trợ xây dựng các trò chơi, hoạt hình 2D hoặc 3D, ứng dụng cơ sở dữ liệu toàn diện để hoàn thiện website.
* Tăng cường các hành vi người dùng và kiểm soát mặc định trình duyệt.

1. **Mối liên hệ giữa HTML, CSS và JS để hoàn thiện một trang Website**



*Hình 1.8. Hình ảnh về ngôn ngữ HTML, CSS và JS để thực hiện xây dựng một trang web*

Tóm lại, vai trò của HTML – CSS – JavaScript được phân định như sau:

* HTML: xác định cấu trúc và nội dung của trang web
* CCS: đặt cách trình bày và thiết kế giao diện web
* JavaScript bổ sung tính tương tác cho trang web, tạo ra các ứng dụng web phong phú hơn

1. **Thiết kế Website “Smart Home”**
   1. **Thiết kế hệ thống bảo mật thông tin**
   2. **Thiết kế Website đăng nhập và lưu trữ thông tin**
      1. **Sử dụng HTML CSS và JS để thiết kế Website đăng nhập**

* ***Code HTML – Hyper Text Markup Language cho hệ thống đăng nhập.***

***A screen shot of a computer program

Description automatically generated***

*Hình 1.9. Hình ảnh về code của HTML cho phần trang Website đăng nhập*

* ***Sử dụng một số thuộc tính như:***
* *<title> Login System </title>:*

**Mục đích:** Để hiển thị tên của dự án vào trong trang web

* *<script src="https://www.gstatic.com/firebasejs/8.4.1/firebase-app.js"></script>*
* *<script src="https://www.gstatic.com/firebasejs/8.4.1/firebase-database.js"></script>*
* *<script src ="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script*

**Mục đích:** Để gọi API kết nối Firebase

* **Các đoạn code nằm trong body có các chức năng như:**
* Dùng để tạo nên bố cục đăng nhập bao gồm như: Tên Username, một ô trống để ghi tên Username. Tương tự như Password, một ô trống để nhập Password
* Dùng để tạo một nút nhấn để đăng nhập vào hệ thống và một nút nhấn để đăng nhập vào hệ thống thay đổi mật khẩu.
* Ngoài ra thì có các đoạn lệnh script để kết nối với file code js và css để thực hiện các thao tác và trang trí trang web được nằm trong lệnh <script></script>
* ***Code CSS - Cascading Style Sheeets cho hệ thống đăng nhập.***

A screenshot of a computer screen

Description automatically generated

*Hình 1.10. Hình ảnh về code của CSS cho phần trang Website đăng nhập*

* ***Sử dụng một số thuộc tính như:***
* Background – color: Dùng để thay đổi màu nền của trang web.
* Text – align: Dùng để căn chỉnh văn bản trong trang web.
* Padding, margin, left, right: Dùng để căn chỉnh vị trí của các compenent trong trang web.
* Font – size: Dùng để điều chỉnh kích thước văn bản cũng như kích thước các nút nhấn có trong trang web.
* Width, height: Dùng để căn chỉnh chiều cao, chiều rộng của các logo, hình ảnh có trong trang web
* Ngoài ra còn các có thuộc tính như border-radius, transistion … để tạo hiệu ứng như mờ chữ khi đưa chuột vào ô gõ password …
* ***Mục đích của CSS trong Website đăng nhập:***
* Dùng để trang trí trang web, căn chỉnh các layout, để định dạng các compenent có trong trang web
* **Code JS cho hệ thống đăng nhập.**

*A screenshot of a computer program

Description automatically generated*

*Hình 1.11. Hình ảnh về code của JS cho phần trang Website đăng nhập*

* ***Sử dụng một số hàm như:***
* Khai báo các thông số Firebase để kết nối như apiKey, aythDomain, databaseURL, projectID …
* Khởi tạo Firebase bằng phương thức initializeAPP (FirebaseConfig)
* Viết một hàm với trang đăng nhập với tài khoản và password được lưu trữ tại Firebase. Nếu như tài khoản và password nhập đúng thì cho đến trang chính còn nếu không thì phải cho đăng nhập lại
* ***Mục đích của JS trong Website đăng nhập***
* Dùng để thực hiện các thao tác như đăng nhập, kết nối Firebase, lấy địa chỉ API của Firebase, lấy các ID của Username, Password để thực hiện chức năng đăng nhập, gọi đến trang chính khi đăng nhập thành công ngược lại bắt người dùng phải đăng nhập lại nếu mật khẩu sai. Và gọi đến trang thay đổi mật khẩu
  + 1. **Kết nối Firebase cho Website để lưu trữ thông tin**
* Để kết nối Firebase với Website đăng nhập ta sử dụng các API và khai báo các tham số apiKey, aythDomain, databaseURL, projectID … để kết nối Firebase

**A computer screen shot of a program code

Description automatically generated**

*Hình 1.12. Hình ảnh về các tham số để kết nối Firebase*

* Để bảo mật được thông tin trang web thì ta cần lưu trữ các dữ liệu của chúng ta ở đám mây, cụ thể là Firebase với cấu trúc dữ liệu như hình bên dưới:

**A screenshot of a computer screen

Description automatically generated A close-up of a password

Description automatically generated**

*Hình 1.13. Hình ảnh về các cấu trúc dữ liệu*

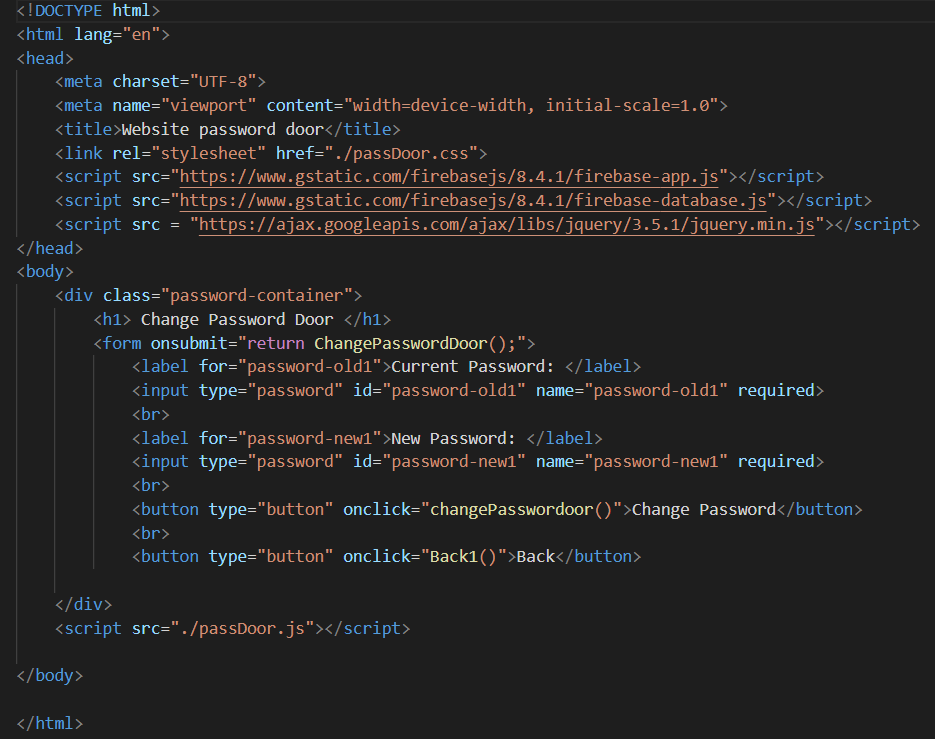
* Với tài khoản đăng nhập là: admin và password: 1234 nếu ta thêm tài khoản và mật khẩu nó sẽ cập nhật qua đây, ngoài ra còn có về password của cửa khoá số là 6789.
* **Mục đích:** Dùng để lưu trữ dữ liệu tránh bị mất mát và lộ thông tin ra bên ngoài để bị mất dữ liệu
  + 1. **Hình ảnh kết quả sau khi code trang Website đăng nhập**

**A screenshot of a login screen

Description automatically generated**

*Hình 1.14 Giao diện Website của trang đăng nhập hệ thống*

* 1. **Thiết kế hệ thống thay đổi mật khẩu**
     1. **Sử dụng HTML CSS và JS để thiết kế Website thay đổi mât khẩu**
* ***Code HTML – Hyper Text Markup Language cho hệ thống thay đổi mật khẩu.***

******

*Hình 1.15 Code HTML cho hệ thống thay đổi mật khẩu*

* Cũng sử dụng các thuộc tính như label, input, button, link. Script … để tạo nên một website cho phần hệ thống đăng nhập. Ngoài ra còn có liên kết Firebase để cập nhật username hay password khi thay đổi từ trang web thay đổi qua
* **Mục đích:**
* Tạo nên trang website đăng nhập sử dụng các thuộc tính trên
* Kết nối Firebase để update Username và Password khi có sự thay đổi từ bên trang web.
* ***Code CSS - Cascading Style Sheeets cho hệ thống thay đổi mật khẩu.***

A screenshot of a computer screen

Description automatically generated

*Hình 1.16 Code CSS cho hệ thống thay đổi mật khẩu*

* Cũng sử dụng các thuộc tính như Background – color, padding, margin, posittion, top, left … để căn chỉnh các compenent có trong hệ thống để định dạng cho hệ thống được đẹp mắt hơn.
* **Mục đích:** Gọi các ID từ bên HTML ra để trang trí cho các layout có trong hệ thống giúp hệ thống được bắt mắt hơn, dễ nhìn và gọn gàng trang web hơn.
* ***Code JS cho hệ thống thay đổi mật khẩu.***

**A screen shot of a computer program

Description automatically generated**

*Hình 1.16 Code JS cho hệ thống thay đổi mật khẩu*

* **Mục đích:** Sử dụng các APIKey, AuthDomain. databaseURL … để lấy API firebase và kết nối Firebase, tạo một hàm để update username hoặc password từ bên Website đưa qua. Nếu đổi mật khẩu thành công thì cho về lại trang đăng nhập để đăng nhập vào hệ thống
  + 1. **Hình ảnh về Website thay đổi mật khẩu**

**A screenshot of a login screen

Description automatically generated**

*Hình 1.17 Hình ảnh về trang web thay đổi mật khẩu cho Website*

1. **Thiết kế giao diện chính cho hệ thống “Smart Home”** 
   1. **Thiết kế Website phần thay đổi mật khẩu**
   2. **Giao diện màn hình chính**
   3. **Giao diện màn hình thông tin cá nhân**
2. **Kết nối website với nền tảng đám mây Firebase**
3. **Điều khiển thiết bị thông qua nút nhấn và nhận dữ liệu**
4. **Nhận diện giọng nói để điều khiển các thiết bị**