Công cụ sử dụng trong hệ thống

# Thiết bị gửi số thứ tự đến Server: NodeMCU



NodeMCU là 1 board mạch được phát triển dựa trên chip ESP8266. Thế mạnh của các board mạch phát triển trên chip ESP8266 là khả năng kết nối wifi, kết nối với các vi điều khiển.

Điều này dẫn đến NodeMCU là sự lựa chọn hoàn hảo cho các ứng dụng IoT (thiết bị được kết nối qua wifi có thể kết nối với các thiết bị khác thông qua Internet.)

NodeMCU được thiết kế và lập trình bằng ngôn ngữ chính yếu là LUA, tuy nhiên chúng ta vẫn có thể dùng Arduino IDE hoặc PlatformIO IDE để lập trình bằng ngôn ngữ C như Arduino.

Một số thông số kỹ thuật board NodeMCU được sử dụng trong dự án:

* Nguồn: 3.3 VDC thông qua cổng Micro-USB hoặc chân Vin/GND.
* Chân điều khiển: EN, RST
* Chân GPIO: 16 chân (GPIO01~GPIO16)
* Chân Analog: 1 chân (A0)
* Chân SPI: 4 chân
* Chân I2C: 1 chân
* Flash/SRAM: 4MB/64KB

Trong đề tài này, nhóm nghiên cứu chọn NodeMCU ESP8266 làm thiết bị điều khiển mạch nhảy số tự động và truyền gửi dữ liệu về cho Máy chủ. Mạch NodeMCU cần cung cấp SSID (tên wifi) và Password để thực hiện kết nối.

# Máy chủ: PHP

PHP là một ngôn ngữ lập trình kịch bản chủ yếu được dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở, dùng cho mục đích tổng quát. Nó rất thích hợp với web và có thể dễ dàng nhúng vào trang HTML. PHP được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP trở thành một ngôn ngữ lập trình web phổ biến nhất thế giới.

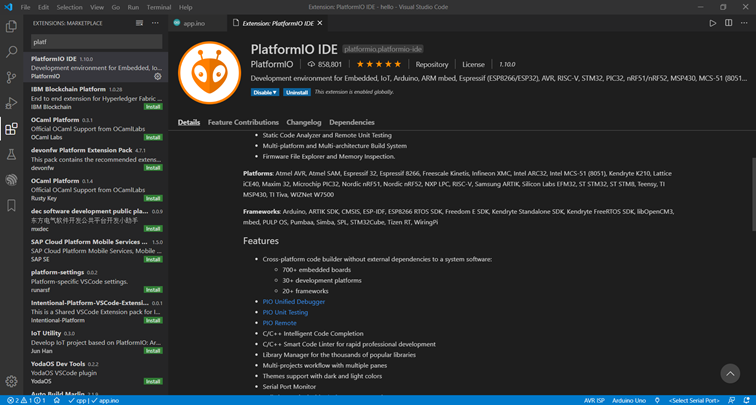
Nhóm phát triển cân nhắc chọn PHP để xây dựng hệ thống máy chủ vì khả năng phát triển nhanh và nhiều nhà cung cấp hosting PHP ở VN với chi phí hợp lý.

# Công cụ phát triển: VSCode + PlatformIO

Arduino IDE là IDE đầu tiên mà hầu như chúng đều sử dụng khi tiếp cận phát triển Arduino lần đầu tiên. Đây là công cụ đơn giản, ổn định nhất, cung cấp tất cả các tính năng cần thiết để viết, biên dịch và tải mã lên các board mạch Arduino.

Hạn chế lớn nhất của ArduinoIDE đối với dự án mà hệ thống có cả Server, Web, Mobile thì một mình IDE này không thể phát triển cho đa nền tảng.

Để giải quyết vấn đề này, nhóm phát triển đã chọn công cụ thay thế là VSCode kết hợp với PlatformIO.



**VSCode** là một trong những Code Editor (công cụ soạn thảo code) mã nguồn rất phổ biến được các lập trình viên sử dụng. VSCode hỗ trợ đa nền tảng, có dung nhẹ, giao diện gần tương tự với các IDE phổ biến khác nên việc tiếp cận với công cụ này khá dễ dàng, đặc biệt nhất là số lượng plugin khổng lồ giúp VSCode mở rộng tính năng.

Một trong những plugin của VSCode có thể kể đến là PlatformIO.

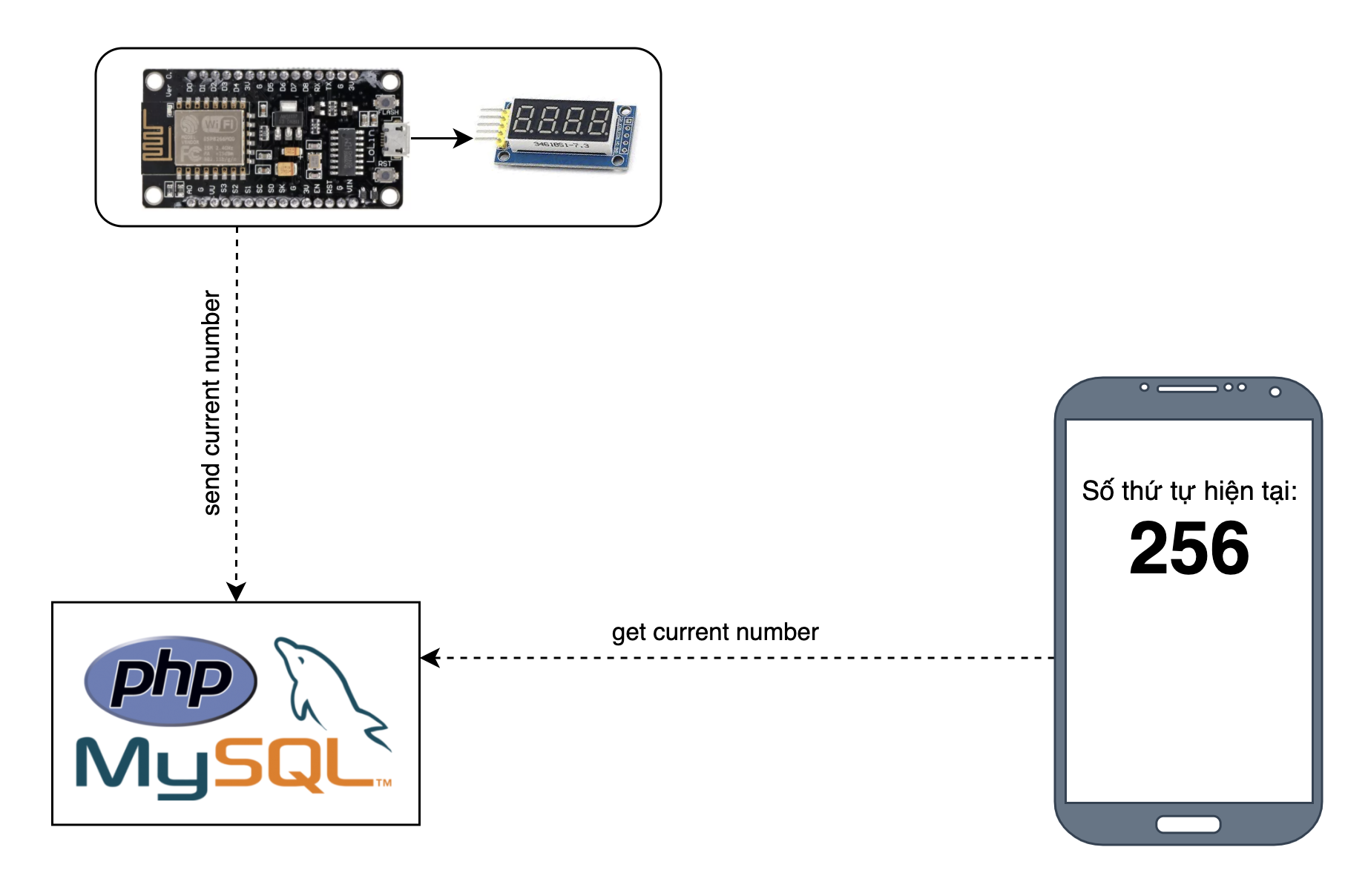
**PlatformIO** là một plugin đa nền tảng, được viết trên Python để phát triển các ứng dụng IoT, đi kèm với trình quản lý cấu hình, quản lý library cho các nền tảng như Arduino hay MBED. Nhờ các trình quản lý này, project được setup trên PlatformIO dễ dàng được chia sẽ cho các collaborator với ít chi phí và ít xảy ra vấn đề về cấu hình hơn ArduinoIDE.

Giống như ArduinoIDE PlatformIO hỗ trợ nhiều platform, framework, board như Arduino, ESP32, ESP8266 (trong đó có NodeMCU) và đi kèm với một số ví dụ và thư viện.

Sau khi cài plugin PlatformIO trên VSCode thì VSCode có thể code và upload code vào board Arduino, Wemos, NodeMCU. Đồng thời trên VSCode, nhóm nghiên cứu cũng có thể xây dựng ứng dụng Backend (PHP) và Web (HTML, Javascript) dễ dàng.

Phương Pháp Xây Dựng Hệ THống

# Các thành phần trong hệ thống



A -**Thiết bị điều khiển bộ nhảy số thứ tự** - Hiển thị và gửi thông tin số thứ tự đến Máy chủ.

B - **Máy Chủ** - Lưu data từ Thiết bị điều khiển bộ nhảy số, Cung cấp data cho WebSite

C - **WebSite** – Hiển thị số thứ tự cho người dùng.

# A - Thiết bị điều khiển bộ nhảy số thứ tự

## Yêu cầu

1. Bộ điều khiển có khả năng tự kết nối internet khi được cấp nguồn
2. Bộ điều khiển có thể hiển thị số trên các led 7 đoạn.
3. Bộ điều khiển có thể gửi số thứ tự hiện tại đến máy chủ.

## Trình tự thực hiện

### Kết nối NodeMCU với wifi (yêu cầu 1 & 3)

Khai báo thông tin board mạch trong PlatformIO là espressif8266 (espressif8266 là vi mạch wifi nền tảng của NodeMCU), PlatformIO sẽ tự liên kết đến các library chức các tập lệnh có thể điều khiển vi mạch, trong số đó có 1 số library chức kết nối wifi như: ESP8266WiFi.h, ESP8266HTTPClient.h, WiFiClient.h, ESP8266WiFiMulti.h.

Tham khảo:

### Hiển thị số trên các Led 7 đoạn (yêu cầu 2)

**Led 7 đoạn là gì?**

Led 7 đoạn là thiết bị hiển thị điện tử để hiển thị số, được sử dụng rộng rãi trong đồng hồ số, máy tính. Khi mỗi đoạn chiếu sáng thì một phần của chữ số sẽ được hiển thị.

Có 2 loại led 7 đoạn là: Anode chung và Cathode chung.

|  |  |
| --- | --- |
|  | LED 7 đoạn cơ bản có 10 chân, trong đó:   * 7 chân được nối với LED A, B, C, D, E, F, G, và DP (Decimal Point). * 2 chân COM tùy loại led sẽ được nối tương ứng với cực Anode hoặc Cathode. |

Ta thấy, tùy loại (anode chung, cathode chung) sẽ có cách điều khiển đèn khác nhau, nhưng luôn cần có ít nhất 7 chân tín hiệu 0-1 và 1 chân nối Anode hoặc Cathode.

Thông thường số thứ tự trong bệnh viện thường hiển thị 4 chữ số (0000 – 9999), nếu ta chỉ đơn thuần sử dụng các chân tín hiệu 0-1 thì cần (8 x 4) 32 chân. Các board NodeMCU và Arduino phổ biến sẽ không đáp ứng đủ số chân cho 4 đèn led 7 đoạn.

Do đó trên thực tế, để điều khiển led 7 đoạn, người ta không điều khiển trực tiếp mà sử dụng các IC để hỗ trợ như 74HC595, TM1637, MC14511B,…

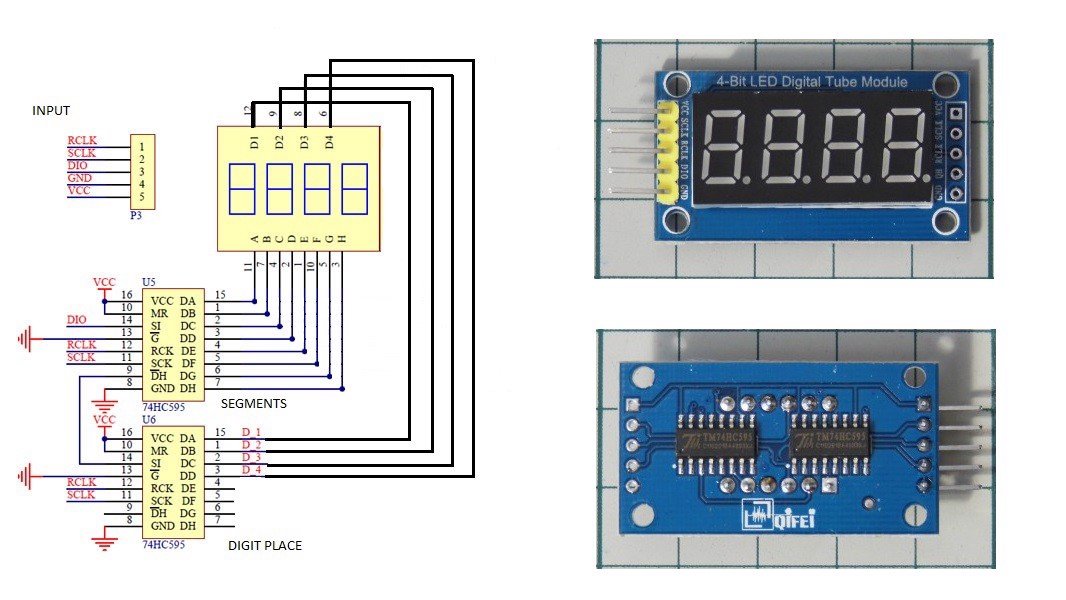
**74HC595** là thanh ghi dịch (register), có chức năng nhận và giải mã các xung dịch và output ra các giá trị 0-1 cho 8 chân tín hiệu ngõ ra của IC. Để tạo ra các xung 74HC595 cần vi điều khiển (Arduino, NodeMCU) điều khiển bằng 3 chân sau:

* Data pin (còn gọi là DS – Data Serial) & Lock pin: gửi tín hiệu cùng thời điểm với nhau, nhưng Lock Pin thì gửi 0/1 xen kẽ tạo thành 1 xung đều đặn, còn Data thì gửi 0/1 ứng với dữ liệu cần gửi đến IC.
* Latch pin: 0 để bắt đầu và 1 để kết thúc 1 dãy bit cần gửi.

|  |  |
| --- | --- |
| [(Link tham khảo)](https://lastminuteengineers.b-cdn.net/wp-content/uploads/arduino/74HC595-Shift-Register-Working.gif) |  |

Nếu Data pin truyền hơn 8 bit thì phần còn lại sẽ theo Q7’ truyền đến chân DS của IC 74HC595 mắc nối tiếp. Như vậy bằng cách mắc nối tiếp N lần IC 74HC595 lại với nhau chúng ta có thể điều khiển N đèn led 7 đoạn.

Trong đề tài lần này, nhóm nghiên cứu sử dụng mạch 3461BS1, đây là mạch gồm 4 led 7 đoạn và 2 chíp dán 74HC595 mắc nối tiếp để 1 IC thực hiện điều khiển led và 1 IC thực hiện chọn led cần điều khiển.



# B - Máy chủ

## Yêu cầu

1. Máy chủ có thể lưu số thứ tự cho nhiều phòng khám cùng lúc.
2. Máy chủ có thể phân biệt được những số thứ tự ngày hôm trước

## Trình tự thực hiện

### Thiết kế database

Để đáp ứng được cả 2 yêu cầu trên. Database cần lưu giữ các thông tin **tối thiểu** sau:

* room\_id: Phòng đang phát số.
* number: Số thứ tự đang phát.
* alert\_time: Thời điểm số được phát ra.

### Thiết kế API

API là viết tắt của Application Programming Interface – phương thức trung gian kết nối các ứng dụng và thư viện khác nhau. Nó cung cấp khả năng truy xuất đến một tập các hàm hay dùng, từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

Ở hệ thống này, chúng ta có 3 bên ứng dụng trao đổi data với nhau: NodeMCU, Máy Chủ, WebSite.

1. NodeMCU 🡪 Máy Chủ: Gửi data số thứ tự đến Máy Chủ
2. WebSite 🡪 Máy Chủ: Yêu cầu Máy Chủ cung cấp thông tin về số thứ tự hiện tại.

Ứng với các xử lý trên, Máy Chủ cần cung cấp 2 API sau:

|  |  |  |
| --- | --- | --- |
| Chức năng | Ứng dụng | Đặc tả API |
| Nhận data từ bộ phận điều khiển nhảy số NodeMCU | NodeMCU | * Url: */postNumber.php* * Method: POST * Request Body:   {  "room\_id": "a986c39e-3ce2-11ed-b878-0242ac120002",  "number": 256  }   * Response: Không có |
| Cung cấp data cho WebSite tra số thứ tự | WebSite | * Url: */getNumber.php?room\_id=YYYY* * Method: GET * Response:   {  "room\_id": "a986c39e-3ce2-11ed-b878-0242ac120002",  "number": 256,  "alert\_time": "2022-09-24 09:12:55"  } |

# C- WebSite

## Yêu cầu

1. Ít tốn tài nguyên Máy Chủ.
2. Nhận data từ Máy Chủ để cập nhật thông tin cho người dùng.
3. Hiện thông báo cho người dùng khi không mở điện thoại

## Trình tự thực hiện

### Để đáp ứng yêu cầu 1 & 2

website không nên thực hiện render giao diện web nhiều lần từ Máy Chủ, sẽ tăng gánh nặng cho và thời gian load trang chậm.

Thay vào đó, nhóm nghiên cứu sẽ thực hiện request data từ Máy Chủ dưới dạng JSON (định dạng giúp lưu trữ các thông tin có cấu trúc và nó chủ yếu được dùng để truyền tải dữ liệu giữa server và client), chỉ cập nhật những vùng giao diện cần thiết.

Nhóm nghiên cứu sử dụng thư viện Axios.js để thực hiện request data và thư viện JQuery để thực cập nhật data lên giao diện web.

|  |
| --- |
| Ví dụ: |
| function update() {  axios.get("https://dauden.cloud/number/get.php")  .then(function (response) {  var num = response.data.number;  if(num == null) {  $("#mynumtext").html("--");  } else {  $("#mynumtext").html(num);  }  }).catch(error => {  $("#mynumtext").html("--");  console.log('Error: ', error)  });  } |

### Để “Hiện thông báo cho người dùng khi không mở điện thoại”

Trước đây không lâu là chức năng độc quyền của ứng dụng Native và không có trên các trang web trên smartphone. Giới hạn này được giải phóng khi công nghệ web giới thiệu kỹ thuật Service Worker, cho phép website thực thi các xử lý chạy ngầm mà không cần giao diện.

Nhằm đảm bảo vấn đề bảo mật và riêng tư của người dùng, ứng dụng web cần sự đồng ý của người dùng cho phép hiển thị Notification.

|  |
| --- |
| function requestNotificationPermission() {  if (!("Notification" in window)) return alert("Trình duyệt của bạn không hỗ trợ Notification")  if (Notification.permission === "denied" || Notification.permission === "default") {  Notification.requestPermission().then((perm) => {  if (perm === "granted") registerSW();  })  } else {  registerSW();  }  } |
| Sau khi đảm bảo website đã được người dùng cấp quyền hiển thị Notification. Ta tiến hành đăng ký một Service Worker chạy ngầm và định nghĩa các phương thức cần thiết để show notification khi số thứ tự của người dùng đến lượt phục vụ. |
| var myRegistration = null;  function registerSW() {  navigator.serviceWorker  .register('sw.js')  .then(function (registration) {  myRegistration = registration;  });  }  /// Refer: https://www.javascripttutorial.net/web-apis/javascript-notification/  /// https://developer.mozilla.org/en-US/docs/Web/API/ServiceWorkerRegistration/showNotification  function showNoti(message, fullMessage) {  const isGranted = Notification.permission === "granted"  if (!isGranted) return;  myRegistration.showNotification(message, {  body: fullMessage,  icon: './assets/icon.png',  image: './assets/noti-bg.jpeg',  sound: './assets/bells-02.ogg',  silent: false,  vibrate: [300, 100, 400],  tag: 'blah-blah-blah'  });  } |

# Tham khảo:

* [Visual Studio Code là gì? Tính năng của Visual Studio Code - Fptshop.com.vn](https://fptshop.com.vn/tin-tuc/danh-gia/visual-studio-code-la-gi-cac-tinh-nang-noi-bat-cua-visual-studio-code-146213)
* [ESP8266WiFi library — ESP8266 Arduino Core documentation (arduino-esp8266.readthedocs.io)](https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html)
* [PlatformIO là gì? Lập trình Arduino với PlatformIO IDE (dientuviet.com)](https://dientuviet.com/lap-trinh-arduino-voi-platformio/)
* [Hướng dẫn lập trình Arduino với PlatformIO IDE (nha-xanh.com)](https://nha-xanh.com/bai-viet/huong-dan-lap-trinh-arduino-voi-platformio-ide/)
* [Bệnh viện (who.int)](https://www.who.int/vietnam/vi/health-topics/hospitals#:~:text=B%E1%BB%87nh%20vi%E1%BB%87n%20c%C3%B4ng%20%E1%BB%9F%20Vi%E1%BB%87t,%E1%BB%9F%20khu%20v%E1%BB%B1c%20th%C3%A0nh%20th%E1%BB%8B.)
* [JSON là gì? Hướng dẫn cách dùng cho người mới bắt đầu (itnavi.com.vn)](https://itnavi.com.vn/blog/json-la-gi)
* [Tìm hiểu về service worker phần 1 (viblo.asia)](https://viblo.asia/p/tim-hieu-ve-service-worker-phan-1-maGK7myOlj2)