TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Viện công nghệ thông tin và truyền thông

XÂY DỰNG HỆ THỐNG THEO DÕI VÀ VẬN HÀNH NHÀ KÍNH TRONG NÔNG NGHIỆP

Môn: Internet of Thing and Application

Nhóm: 11

Giáo viên hướng dẫn: TS. Nguyễn Đình Thuận

Nhóm sinh viên: 1. Lê Tường Khanh – 20180109

2. Nguyễn Phi Phúc – 20180155

3. Trần Hữu Hiếu –20180078

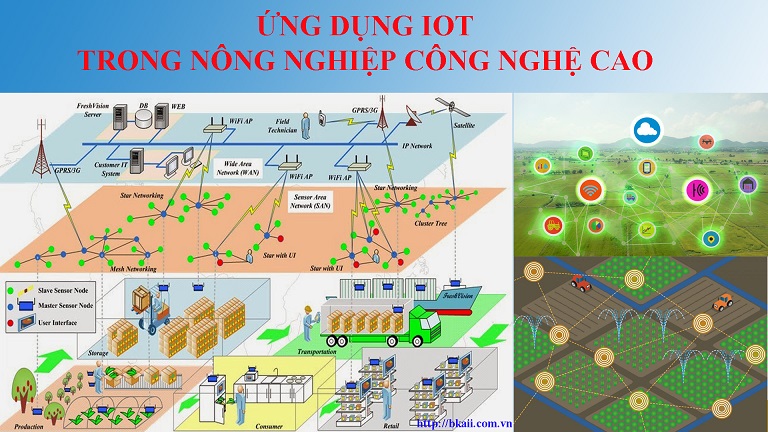
*Hà Nội, ngày 3 tháng 3 năm 2022*

# 

# Tổng quan đề tài

## 1.1 Giới thiệu về nông nghiệp thông mình vào IOT

***Nông nghiệp công nghệ cao , IoT,.... có lẽ chưa bao giờ hai từ khóa này lại được quan tâm nhiều đến như thế. Cũng phải thôi, nông nghiệp là sở trường của nước ta theo đúng nghĩa đen. Công nghệ thông tin, điện tử, viễn thông cũng đang là sở trường của các kỹ sư Việt Nam trên đấu trường quốc tế. Như vậy, nếu kết hợp được việc ứng dụng IoT vào trong nông nghiệp thì đây đúng là một hướng phát triển bền vững và mạnh mẽ của nước ta.***

******

Nông nghiệp - một khái niệm rộng lớn trải dài nhiều ngành nghề nhỏ: chăn nuôi, trồng trọt,.... Trong chăn nuôi lại có nuôi lợn, bò, gà, vịt, nuôi tôm, nuôi cá,... Trong trồng trọt thì là trồng hoa, quả, lúa, rau.....Có thể nói Nông nghiệp là "sở trường" theo đúng nghĩa đen của nước ta với các vị thế xuất khẩu lúa, gạo, hồ tiêu,.... thuộc top đầu thế giới. Vậy, nông nghiệp công nghệ cao nên được hiểu như thế nào? Theo Wikipedia tiếng Việt: "Nông nghiệp công nghệ cao là một nền nông nghiệp được ứng dụng kết hợp những công nghệ mới, tiên tiến để sản xuất, còn gọi là công nghệ cao nhằm nâng cao hiệu quả, tạo bước đột phá về năng suất, chất lượng nông sản, thỏa mãn nhu cầu ngày càng cao của xã hội và đảm bảo sự phát triển nông nghiệp bền vững". Như vậy, nói nôm na là: sản phẩm nông nghiệp cần chuyển đổi dần theo hướng: Năng suất -> Sạch -> Chất Lượng. Bấy lâu nay, người nông dân mới chỉ chú tâm nâng cao năng suất và thậm chí tìm mọi cách để nâng cao năng suất mà chưa quan tâm tới "sạch và chất lượng", tiêu biểu như: Phun thuốc trừ sâu, thuốc kích thích trong việc trồng rau - lúa, cho ăn cám tăng trọng trong việc nuôi lợn,.....

Như vậy, không cần nói xa xôi, trước mắt chúng ta cần định hướng toàn diện từ "năng suất" sang "sạch". Xu hướng này đã xuất hiện từ cách đây vài năm. Nếu để ý, chúng ta sẽ thấy xu hướng: tự cung - tự cấp, tức là các bà, mẹ, chị không còn tin tưởng với thực phẩm mua ngoài nên tự trồng rau, tự nuôi con gà, con lợn ...sạch để ăn. Đây là cách "phòng thủ" để bảo vệ phần nào sức khỏe gia đình của các bà mẹ nội trợ trước ma hồn trận thực phẩm không đảm bảo chất lương. Cũng đâu đó, đã xuất hiện các vườn rau hữu cơ - tức là không dùng thuốc bảo vệ thực vật, trồng hoàn toàn tự nhiên, dùng "thuốc trừ sâu sinh học" từ tỏi-rượu,.. rồi các trang trại nuôi lợn bằng thảo dược...rồi các mô hình nhà kính, trồng thủy canh. Sắp đến những ngày tết, giờ "mốt" không phải quà biếu tết là những chai rượu đắt tiền nữa, mà thay vào đó là "những con lợn quê nuôi tự nhiên hoàn toàn", " những con gà chạy bộ thả rông trên núi",...

Xong, coi như sản phẩm đã năng suất và sạch, giờ chúng ta bàn đến "Chất lượng", trong chúng ta khi ăn một miếng thịt lơn nuôi bằng thảo dược, sẽ thấy thịt thơm và có hương vị đặc trưng. Ừ, ăn vào và cảm nhận, thấy đúng thật. Nhưng đó mới chỉ là "cảm nhận" và nó mới chỉ là định tính, vậy người dùng có thể định lượng được không? Tức là có thể cân - đong - đo - đếm xem nó có đáp ứng tiêu chuẩn không, ăn vào có tốt không, có thừa hay thiếu các chất cần thiết không,.... Tất nhiên, việc này thì sẽ do doanh nghiệp và cơ quan nhà nước quản lý, bởi doanh nghiệp là đơn vị sản xuất, đầu ra là các sản phẩm sẽ được kiểm định bởi các cơ quan quản lý nhà nước. Khi được giám sát chặt chẽ và cấp phép ra thị trường tiêu thụ: chúc mừng các bạn, đã được sử dụng những sản phẩm: sạch và chất lượng theo đúng nghĩa đen.

Vậy chúng ta sẽ ứng dụng IOT vào trong nông nghiệp như thế nào ?

1. Áp dụng công nghệ khoa học vào toàn bộ quá trình trong nông nghiệp (khép kín): Cụ thể là áp dụng công nghệ thông tin, điện tử, viễn thông vào các quá trình từ chuẩn bị giống, đất,... -> trồng trọt - chăn nuôi -> thu hoạch -> chế biến -> bảo quản -> phân phối -> đến bàn ăn
2. Dữ liệu thu thập được phải tạo thành database ở quy mô lớn, để dần tự động hóa được cả quá trình ( tức là: loại bỏ dần "kinh nghiệm" của con người, chủ động nhận biết vấn đề và đề xuất cách giải quyết). Nôm na, có thể gọi là trí tuệ nhân tạo, thay con người đưa ra quyết định.

***Mô hình nhà kính - green house***



Nhà kính - hiện được sử dụng phổ biến ở nước ta, đặc biệt là ở Đà Lạt - nơi có lợi thế về khí hậu và thời tiết. Nhà kính ban đầu ra đời với mục đích giúp tách ly cây trồng với điều kiện thời tiết bên ngoài. Dần dần, được bổ xung thêm các hệ thống kiểm soát khí hậu bên trong nhà kính ( nhiệt độ, độ ẩm, ánh sáng,...) và hệ thống điều khiển tưới. Hiểu nôm na hai hệ thống như sau:

* Hệ thống điều khiển tưới: Hệ thống bao gồm các thiết bị: đầu tưới nhỏ giọt hoặc đầu tưới phun sương/mưa, bộ châm phân, bộ điều khiển tưới..... hệ thống giúp tưới nước/phân một cách tiết kiệm, hiệu quả và đạt năng suất cao. Giúp người nông dân giảm giá thành chi phí sản xuất.
* Hệ thống điều khiển vi khí hậu: Hệ thống bao gồm các cảm biến nhận biết nhiệt độ, độ ẩm bên trong ( và bên ngoài) nhà kính, hệ thống quạt thông gió để đối lưu không khí, hệ thống đèn chiếu sáng để có thể tăng cường ánh sáng khi cần thiết, trạm đo thời tiết để biết các thông số: cường độ bức xạ mặt trời, cảnh báo mưa, tốc độ gió, lưu lượng mưa,.... Mục đích giúp nhà kính duy trì ở điều kiện mong muốn.

## 

## 1.2 Đặt vấn đề

Trong nông nghiệp thì sự phát sinh, phát triển và lây lan các dịch bệnh do nhiều yếu tố ngoại cảnh như: độ ẩm đất, nhiệt độ, độ PH… , sự ảnh hưởng của những mầm bệnh lây lan từ nơi này đến nơi khác, làm hỏng cây trồng. Do đó cần phải có nhà kính, một nơi tối ưu cho việc cách ly vườn rau với các yếu tố ngoại cảnh, các tác nhân không tốt cho cây trồng và dịch bệnh, dễ dàng cho việc chăm sóng tưới tiêu, thu hoạch.

Từ những lợi ích đã rút ra từ những tính năng của hệ thống IOT trong nông nghiệp và trong nhà kính, nhóm sẽ kết hợp lại để tạo thành một project đơn giản cung cấp một giải pháp để vận hành và theo dõi nhà kính. Hệ thống bao gồm 3 phần:

* Client sẽ thực thi tưới tiêu, chăm sóc cây trồng khi phát hiện có sâu bệnh
* Các sensor để đo nhiệt độ, đổ ẩm không khí và độ ẩm đất.
* Web server để có track những dữ liệu trên.

Và quá trình này sẽ được thực hiện một cách tự động hoá để giúp nông dân tiết kiệm thời gian, công sức và tăng năng suất nông sản.

# 

# Thiết kế hệ thống

## 2.1. Các công nghệ sử dụng

### 2.1.1 WIFI

Wifi là **viết tắt của Wireless Fidelity** là hệ thống truy cập internet không dây. Loại sóng vô tuyến này tương tự như sóng điện thoại, truyền hình và radio.

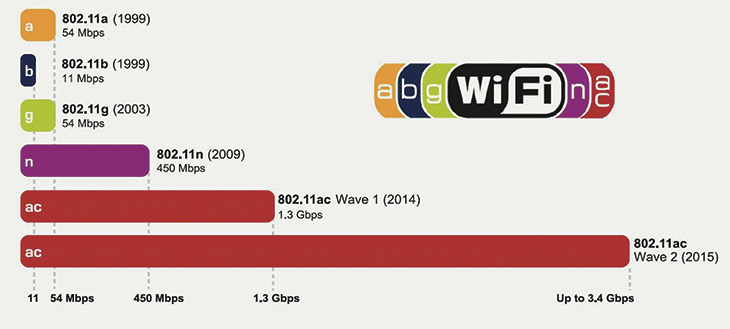
Wifi là công cụ kết nối không thể thiếu trên điện thoại, laptop, [máy tính bảng](https://www.dienmayxanh.com/may-tinh-bang) và một số thiết bị thông minh khác như [smartwatch](https://www.dienmayxanh.com/dong-ho-thong-minh).

### **Nguyên tắc hoạt động của mạng Wifi**

Để tạo được kết nối Wifi nhất thiết phải có [Router](https://www.dienmayxanh.com/thiet-bi-mang-router)**(bộ thu phát),** Router này lấy thông tin từ mạng Internet qua kết nối hữu tuyến rồi chuyển nó sang tín hiệu vô tuyến và gửi đi, bộ chuyển tín hiệu không dây **(adapter)** trên các thiết bị di động thu nhận tín hiệu này rồi giải mã nó sang những dữ liệu cần thiết.

Quá trình này có thể thực hiện ngược lại, Router nhận tín hiệu vô tuyến từ Adapter và giải mã chúng rồi gởi qua Internet.

### **Một số chuẩn kết nối Wifi hiện nay**



#### **Chuẩn 802.11**

Năm 1997, IEEE giới thiệu chuẩn mạng không dây đầu tiên và đặt tên nó là 802.11. Khi đó, tốc độ hỗ trợ tối đa của mạng này chỉ là **2 Mbps** với băng tầng **2.4GHz.**

#### **Chuẩn 802.11b**

Vào tháng 7/1999, chuẩn 802.11b ra đời và hỗ trợ **tốc độ lên đến 11Mbps**. Chuẩn này cũng hoạt động tại **băng tần 2.4GHz** nên cũng rất dễ bị nhiễu từ các thiết bị điện tử khác.

#### **Chuẩn 802.11a**

Song song với quá trình hình thành chuẩn b, chuẩn 802.11a phát ở tần số cao hơn là **5GHz**nhằm tránh bị nhiễu từ các thiết bị khác. Tốc độ xử lý của chuẩn đạt **54 Mbps** tuy nhiên chuẩn này khó xuyên qua các vách tường và giá cả của nó hơi cao.

#### **Chuẩn 802.11g**

Chuẩn 802.11g có phần hơn so với chuẩn b, tuy nhiên nó cũng hoạt động ở **tần số 2.4GHz** nên vẫn dễ nhiễu. Chuẩn này có thể xử lý tốc độ lên tới **54 Mbps.**

#### **Chuẩn 802.11n**

Ra mắt năm 2009 và là chuẩn phổ biến nhất hiện nay nhờ sự vượt trội hơn so với chuẩn b và g. Chuẩn kết nối 802.11n hỗ trợ tốc độ tối đa lên đến **300Mbps**, có thể hoạt động trên cả **băng tần 2,4 GHz và 5 GHz.**

Chuẩn kết nối này đã và đang dần thay thế chuẩn 802.11g với, phạm vi phát sóng lớn hơn, tốc độ cao hơn và giá hợp lý.

#### **Chuẩn 802.11ac**

Là chuẩn được IEEE giới thiệu vào đầu năm 2013, hoạt động ở **băng tầng 5 GHz**. Chuẩn ac có thể mang đến cho người dùng trải nghiệm tốc độ cao nhất lên đến **1730 Mpbs.**

Do vấn đề giá thành cao nên các thiết bị phát tín hiệu cho chuẩn này chưa phổ biến dẫn đến các thiết bị này sẽ bị hạn chế sự tối ưu do thiết bị phát.

#### **Chuẩn 802.11ad**

Được giới thiệu năm 2014, chuẩn wifi 802.11ad được hỗ trợ **băng thông lên đến 70 Gbps** và hoạt động ở **dải tần 60GHz**. Nhược điểm của chuẩn này là sóng tín hiệu khó có thể xuyên qua các bức tường, đồng nghĩa với việc chỉ cần Router khuất khỏi tầm mắt, thiết bị sẽ không còn kết nối tới Wifi được nữa.

#### **Chuẩn 802.11ax**

Wi-Fi 6 là bản cập nhật mới nhất cho chuẩn mạng không dây. **Wi-Fi 6** dựa trên tiêu chuẩn **IEEE 802.11ax**, với tốc độ **nhanh hơn, dung lượng lớn hơn** và hiệu suất năng lượng được cải thiện tốt hơn so với các kết nối không dây trước đây. Tên gọi mới Wifi 6 này sẽ chính thức được áp dụng từ năm 2019.

**Wifi Hotspot**

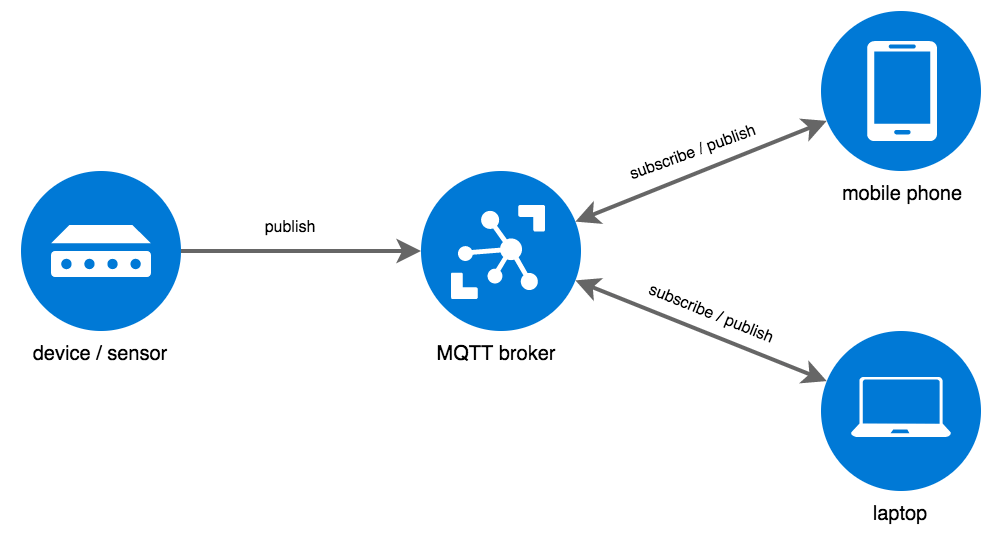
Ngoài những chuẩn kết nối kể trên, mỗi thiết bị di động có thể tự phát ra sóng Wifi cho những thiết bị khác. Nói cách khác, thiết bị di động có thể được xem như là một Router.

### 2.1.2 MQTT BROKER

Giao thức MQTT xác định hai dạng thực thể trong mạng: một message broker và những client.

Một MQTT broker là một máy chủ (server) nhận tất cả các tin nhắn (message) từ các client và sau đó định tuyến (route) các tin nhắn này đến các client phù hợp.[[18]](https://vi.wikipedia.org/wiki/MQTT#cite_note-message_broker-_client-18) Một MQTT client là bất kỳ thiết bị nào (từ vi điều khiển đến một máy chủ chính thức) chạy thư viện MQTT và kết nối tới MQTT broker qua mạng.[[19]](https://vi.wikipedia.org/wiki/MQTT#cite_note-MQTT_client_-_broker-19)

MQTT[[20]](https://vi.wikipedia.org/wiki/MQTT#cite_note-20) quản lý các thông tin - dữ liệu mà nó nhận được theo hệ thống cấp bậc của các topic (tạm dịch: chủ đề). Với cơ chế publish-subscribe của MQTT, khi một publisher có một dữ liệu muốn truyền đi, nó sẽ gởi một tin nhắn điều khiển (control message) với dữ liệu muốn truyền đi đó đến MQTT broker mà nó đã kết nối đến. Broker sau đó sẽ gởi các thông tin mà nó nhận được này đến client đã đăng ký (subscribe) vào topic đó. Các publisher không cần có bất cứ thông tin nào về số lượng hay vị trí của các subscriber, đồng thời, các subscriber cũng không cần phải được cấu hình để có bất kì thông tin gì về các publisher.



### **MQTT CLIENT**

Là bất cứ thiết bị nào sử dụng một thư viện mqtt để kết nối đến một MQTT Broker thông qua một mạng như cảm biến chạy pin, thiết bị gia dụng hay một máy tính.

MQTT Client thông thường đóng cả hai vai trò publisher/subscriber tuỳ vào việc chúng gửi hay nhận các thông điệp đến/từ MQTT Broker thông qua các topic.

## **MQTT CLIENTID, TOPIC, PAYLOAD, QOS VÀ RETAINED**

### **CLIENTID /TOPIC**

MQTT Broker quản lý các clients qua ClientID, vì vậy các ClientID này cần là duy nhất cho mỗi client. Client không biết về các clientid khác trên cùng mạng, cũng không biết về địa chỉ IP hay bất cứ định danh nào khác.

Client gửi và nhận – publish/subscribe thông điệp thông qua các *Topic* – chủ đề. Một Client có thể publish hay subscribe vào nhiều topic (mô hình 1-n). MQTT Broker sẽ tự động phân phối bất cứ thông điệp nào được publish – xuất bản vào một topic đến tất cả Client đã subscribe vào topic này.

*MQTT Topic* thực chất là một chuỗi kí tự (UTF-8) được sử dụng bởi Broker để phân loại thông điệp trước khi gửi đến Client. MQTT Topic có thể bao gồm nhiều mức – level, phân cách với nhau bởi dấu gạch ‘/’, ví dụ: mysweethome/livingroom/switch/main light/status. Mỗi topic phải có ít nhất một kí tự, cho phép khoảng trống – space và có phân biệt chữ hoa và thường; ‘/’ là một topic hợp lệ.

### **PAYLOAD / QOS**

*Payload*: là phần nội dung thực sự chứa dữ liệu của một thông điệp được Client gửi đến Broker hay được Broker phân phối lại cho các Client đã đăng ký. Đây cũng chính là phần dữ liệu các Client khác nhận được khi subscribe vào cùng topic.

QoS: quality of service, mức đảm bảo. Có ba mức Qos chính:

– *Qos 0*: đảm bảo thông điệp gửi đi sẽ đến đích nhiều nhất là 1 lần. Thông điệp có thể bị mất khi kết nối bị gián đoạn. Nhờ sự đảm bảo toàn vẹn dữ liệu của giao thức TCP/IP, mức Qos 0 là đủ dùng trong hầu hết trường hợp thông thường.

– *Qos 1*: thông điệp gửi đi đảm bảo sẽ đến đích nhưng có thể bị trùng lắp.

– *Qos 2*: thông điệp gửi đi đảm bảo đến đích và chỉ đúng 1 lần. Mức Qos 2 và Qos 1 thường được sử dụng trong các môi trường mà chỉ cơ cấu TCP/IP là không đủ đảm bảo khi mà các thiết bị kết nối qua kết nối kém ổn định như Vệ Tinh (hoặc Lora).

### **RETAINED**

**Retained** – lưu giữ: một thông điệp được đánh dấu Retained sẽ được MQTT Broker lưu trữ với đầy đủ payload và Qos. Khi đó, mọi client subscribe – đăng ký vào cùng topic sẽ nhận được thông điệp này ngay khi nó được publish và ngay khi subscribe thành công. Như vậy nếu một client liên tục bị ngắt kết nối rồi kết nối và subscribe vào topic, nó sẽ nhận được cùng một thông điệp này mọi lần subscribe thành công.

Trên mỗi topic, Broker chỉ giữ một thông điệp Retained duy nhất, thông điệp đánh dấu Retained sau sẽ thay thế cái trước. Các thông điệp không đánh dấu Retained sẽ được Broker phân phối bình thường nhưng không ảnh hưởng đến thông điệp Retained đang lưu trữ.

Một client có thể *xoá bỏ* thông điệp Retained trên một topic bằng cách đơn giản là publish một payload trống ” được đánh dấu *Retained* vào cùng topic.

## **BẢO MẬT MQTT**

Giao thức MQTT thực hiện xác thực thông qua cặp *username/password* – tên người dùng/mật khẩu. Với giao thức không mã hoá, thông tin này được gửi đi dưới dạng chữ – plain text thông thường.

### **BẢO MẬT KHI TRUYỀN TẢI**

*Bảo mật phương thức truyền tải – secure transport*: để tăng cường bảo mật, giao thức MQTTs áp dụng mã hoá TLS/SSL lên toàn bộ các thông tin được gửi đi (tương tự HTTPs) giữa MQTT Broker và Client, bao gồm cả tuỳ chọn yêu cầu client cung cấp chứng chỉ riêng để so sánh với chứng chỉ lưu tại Broker. Lưu ý rằng việc này sẽ gia tăng mức độ bảo mật nhưng đồng thời cũng làm tăng đáng kể dung lượng truyền tải, dẫn đến tăng thời gian phản hồi, tăng độ trễ, tăng khả năng mất tin và giảm thời lượng pin của các thiết bị.

Cổng – port mặc định của giao thức MQTT là 1883 trong khi 8883 là cổng mặc định cho giao thức MQTTs, tuy nhiên Broker có thể được cấu hình để thay đổi các cổng này.

### **BẢO MẬT NETWORK**

*Bảo mật mạng lưới – secure network*: thực hiện giao thức MQTT thông qua các kết nối bảo mật như VPN, [**SSH**](https://konnected.vn/the/openssh).

### **PHÂN QUYỀN ACL**

*Phân quyền Topic – MQTT ACL (access control list)*: MQTT *Broker* có thể được cấu hình để cho phép một Client chỉ được publish và subscribe vào những topic nhất định, dựa trên một danh sách gọi là ACL. Danh sách này xác định một ClientID hoặc username được quyền publish và subscribe vào những topic nào. Mặc định, ACL được thiết lập ở dạng không áp dụng hoặc cho phép tất cả trên hầu hết các MQTT Broker thông dụng.

\**Không* *bao giờ nên xây dựng/duy trì một mạng MQTT mở và hoàn toàn không xác thực.*

## **MQTT BRIDGE**

Bạn có thể mở rộng hoặc gia tăng bảo mật mạng MQTT hay kết nối nhiều mạng MQTT từ các vị trí cách xa nhau bằng cách xây dựng các MQTT Bridge, đóng vai trò vừa là MQTT Broker vừa là Client để chuyển tiếp các thông điệp. Một trường hợp sử dụng thường thấy là các Client trong một khu vực kết nối đến một MQTT Broker qua giao thức MQTT không mã hoá, sau đó MQTT Broker này đóng vai trò là MQTT Bridge để chuyển tiếp các thông điệp đến một Broker khác ở xa thông qua giao thức MQTTs.

Bằng cách này, bạn có thể xây dựng một mạng MQTT với nhiều vùng riêng biệt và mở rộng không giới hạn.

## **CÁC MQTT BROKER THÔNG DỤNG**

[***Mosquitto***](https://mosquitto.org/): miễn phí, nguồn mở. Là MQTT Broker thông dụng và rất gọn nhẹ.

[***Mosca***](https://github.com/moscajs/mosca): miễn phí, nguồn mở, viết bằng ngôn ngữ JavaScript. Thường được sử dụng trong Node.js tuy nhiên có thể hoạt động độc lập.

[***EMQ***](https://www.emqx.io/): miễn phí, nguồn mở, viết bằng ngôn ngữ [**Erlang**](https://www.erlang.org/)/OTP. Hỗ trợ MQTT 3.1, 3.1.1 và 5.0, hỗ trợ websocket – ws, MQTT-SN, CoAP, STOMP và SockJS.

[***VerneMQ***](https://vernemq.com/): miễn phí, nguồn mở, viết bằng ngôn ngữ Erlang/OTP. Hỗ trợ MQTT 3.1, 3.1.1 và 5.0, hỗ trợ websocket – ws.

Có sẵn cho nền tảng GNU/Linux và Mac OS.

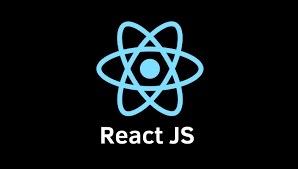
[***HiveMQ***](https://www.hivemq.com/): giấy phép thương mại sở hữu bởi dc-square GmbH, viết bằng ngôn ngữ Java. Hỗ trợ MQTT 3.1, 3.1.1, hỗ trợ websocket – ws. Nguồn mở công bố trên github, áp dụng giấy phép Apache 2.0.

### 2.1.3.Node.js



* [NodeJS](https://topdev.vn/blog/node-js-la-gi/) là một nền tảng được xây dựng trên “V8 Javascript engine” được viết bằng c++ và Javascript. Nền tảng này được phát triển bởi Ryan Lienhart Dahl vào năm 2009.
* Node.js ra đời khi các developer đời đầu của JavaScript mở rộng nó từ một thứ bạn chỉ chạy được trên trình duyệt thành một thứ bạn có thể chạy trên máy của mình dưới dạng ứng dụng độc lập.

### 2.1.4.ReactJS



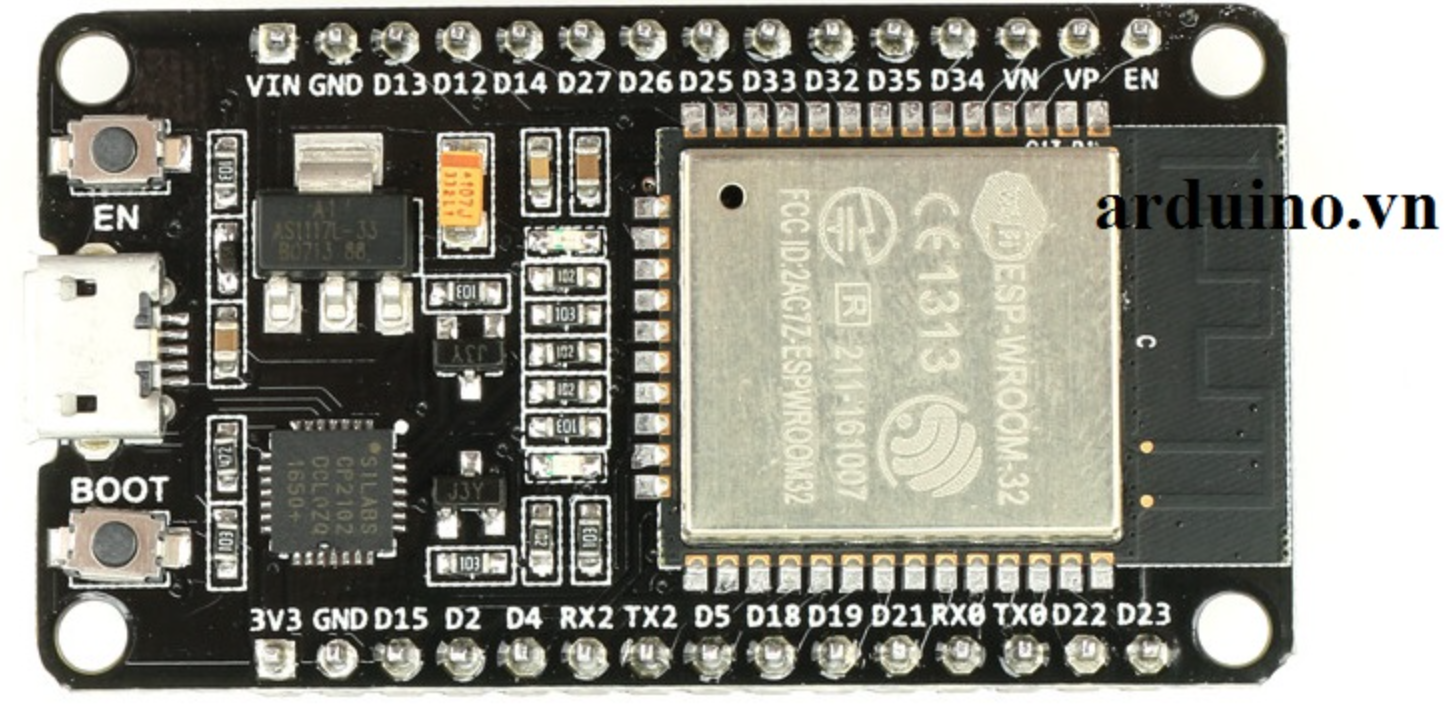
* ReactJS là một thư viện JavaScript mã nguồn mở được thiết kế bởi Facebook để tạo ra những ứng dụng web hấp dẫn, nhanh và hiệu quả với mã hóa tối thiểu. Mục đích cốt lõi của ReactJS không chỉ khiến cho trang web phải thật mượt mà còn phải nhanh, khả năng mở rộng cao và đơn giản.
* Sức mạnh của nó xuất phát từ việc tập trung vào các thành phần riêng lẻ. Chính vì vậy, thay vì làm việc trên toàn bộ ứng dụng web, ReactJS cho phép một developer có thể phá vỡ giao diện người dùng phức tạp thành các thành phần đơn giản hơn

## 2.2. Kiến trúc hệ thống

## 2.3. Các thành phần chính

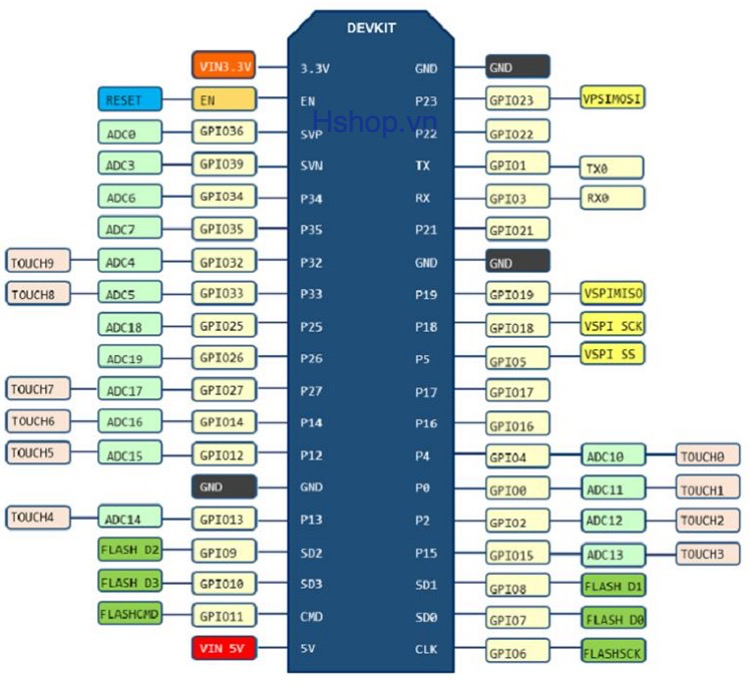
### 2.1.1.Kít phát triển ESP32 – NodeMCU-32S

Kit RF thu phát Wifi BLE ESP32 NodeMCU-32S CP2102 Ai-Thinker được phát triển trên nền Vi điều khiển trung tâm là ESP32 SoC với công nghệ Wifi, BLE và kiến trúc ARM mới nhất hiện nay, kit có thiết kế phần cứng, firmware và cách sử dụng tương tự Kit NodeMCU ESP8266, với ưu điểm là cách sử dụng dễ dàng, ra chân đầy đủ, tích hợp mạch nạp và giao tiếp UART CP2102, thích hợp với các nghiên cứu, ứng dụng về Wifi, BLE, IoT và điều khiển, thu thập dữ liệu qua mạng.



**Thông số kỹ thuật:**

* Nhà sản xuất: Ai-Thinker
* Module trung tâm: Ai-Thinker ESP32-S
* SPI Flash: 32Mbits
* Frequency Range: 2400~2483.5Mhz
* Bluetooth: BLE 4.2 BR/EDR
* Wifi: 802.11 b/g/n/e/i
* Support Interface: UART/SPI/SDIO/I2C/PWM/I2S/IR/ADC/DAC
* Nguồn sử dụng: 5VDC từ cổng Micro USB.
* Tích hợp mạch nạp và giao tiếp UART CP2102
* Chuẩn 38 chân cắm 2.54mm, ra chân đầy đủ module ESP32.
* Tích hợp Led Status, nút nhấn IO0 (BOOT) và ENABLE.
* Kích thước: 25.4 x 48.3mm



### 2.1.2. Sensor cảm biến độ ẩm – Nhiệt độ 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.

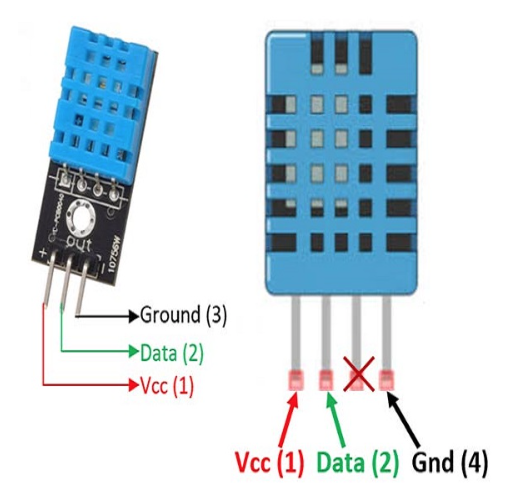
DHT11 là một cảm biến độ ẩm tương đối. Để đo không khí xung quanh, cảm biến này sử dụng một điện trở nhiệt và một cảm biến độ ẩm điện dung.

#### **Cấu tạo cảm biến nhiệt độ độ ẩm DHT11**

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 giữa chúng. Thay đổi giá trị điện dung xảy ra với sự thay đổi của các mức độ ẩm. IC đo, xử lý các giá trị điện trở đã thay đổi này và chuyển chúng thành dạng kỹ thuật số.

Để đo nhiệt độ, cảm biến này sử dụng một nhiệt điện trở có hệ số nhiệt độ âm, làm giảm giá trị điện trở của nó khi nhiệt độ tăng. Để có được giá trị điện trở lớn hơn ngay cả đối với sự thay đổi nhỏ nhất của nhiệt độ, cảm biến này thường được làm bằng gốm bán dẫn hoặc polymer.

| Số chân | Tên chân | Mô tả |
| --- | --- | --- |
| 1 | Vcc | Nguồn 3.5V đến 5.5V |
| 2 | Data | Truyền dữ liệu nhiệt đổ đổ ẩm thông qua cổng nối tiếp |
| 3 | Ground | Nối đất |



#### Thông số kỹ thuật DHT11

- Điện áp hoạt động: 3V - 5V DC

- Dòng điện tiêu thụ: 2.5mA

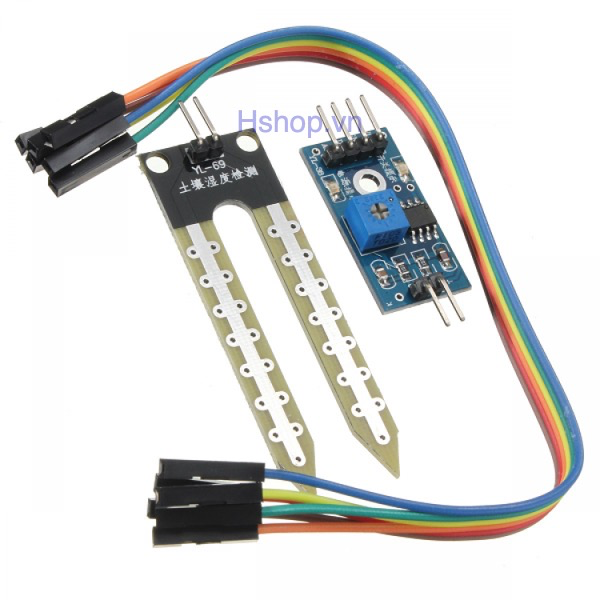
- Phạm vi cảm biến độ ẩm: 20% - 90% RH, sai số ±5%RH

- Phạm vi cảm biến nhiệt độ: 0°C ~ 50°C, sai số ±2°C

- Tần số lấy mẫu tối đa: 1Hz (1 giây 1 lần)

- Kích thước: 23 \* 12 \* 5 mm

### 2.1.3. Cảm biến độ ẩm đất



### Cảm biến độ ẩm đất Soil Moisture Sensor thường được sử dụng trong các mô hình tưới nước tự động, vườn thông minh,..., cảm biến giúp xác định độ ẩm của đất qua đầu dò và trả về giá trị Analog, Digital qua 2 chân tương ứng để giao tiếp với Vi điều khiển để thực hiện vô số các ứng dụng khác nhau.

**Nguyên lý hoạt động**

Cảm biến độ ẩm đất, trạng thái đầu ra mức thấp (0V), khi đất thiếu nước đầu ra sẽ là mức cao (5V), độ nhạy cao chúng ta có thể điều chỉnh được bằng biến trở. Phần đầu đo được cắm vào đất để phát hiện độ ẩm của đất, khi độ ầm của đất đạt ngưỡng thiết lập, đầu ra DO sẽ chuyển trạng thái từ mức thấp lên mức cao. Nhờ thế, các bạn có thể sử dụng Analog hoặc Digital của Arduino để đọc giá trị từ cảm biến.

**Cấu tạo cảm biến**

Cảm biến độ ẩm đất có 4 chân : Vcc, GND, 2 ngõ ra là D0 ( cho giá trị trả về mức logic 0 1) và A0 (giúp bạn có thể đọc được chính xác hơn độ ẩm của đất ). Bạn có thể dùng 1 trong 2 chân này...Ở đây đọc giá trị của cả 2 chân

| Số chân | Tên chân | Mô tả |
| --- | --- | --- |
| 1 | VCC | 3.3V~5V |
| 2 | GND | GND của nguồn ngoài |
| 3 | D0 | Đầu ra tín hiệu số( mức logic cao hoặc mức logic thấp) |
| 4 | A0 | Đầu ra tín hiệu tương tự |

**Thông số kỹ thuật:**

* Điện áp hoạt động: 3.3~5VDC
* Tín hiệu đầu ra:
  + Analog: theo điện áp cấp nguồn tương ứng.
  + Digital: High hoặc Low, có thể điều chỉnh độ ẩm mong muốn bằng biến trở thông qua mạch so sánh LM393 tích hợp.
* Kích thước: 3 x 1.6cm

## 2.4. Các thiết bị Client

### 2.4.1 LCD Text LCD1602 Xanh lá



Màn hình text LCD1602 xanh lá sử dụng driver HD44780, có khả năng hiển thị 2 dòng với mỗi dòng 16 ký tự, màn hình có độ bền cao, rất phổ biến, nhiều code mẫu và dễ sử dụng thích hợp cho những người mới học và làm dự án.

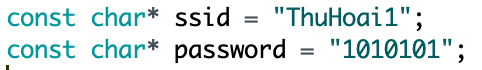
**Thông số kỹ thuật:**

* Điện áp hoạt động là 5 V.
* Kích thước: 80 x 36 x 12.5 mm
* Chữ đen, nền xanh lá
* Khoảng cách giữa hai chân kết nối là 0.1 inch tiện dụng khi kết nối với Breadboard.
* Tên các chân được ghi ở mặt sau của màn hình LCD hổ trợ việc kết nối, đi dây điện.
* Có đèn led nền, có thể dùng biến trở hoặc PWM điều chình độ sáng để sử dụng ít điện năng hơn.
* Có thể được điều khiển với 6 dây tín hiệu
* Có bộ ký tự được xây dựng hổ trợ tiếng Anh và tiếng Nhật, xem thêm HD44780 datasheet để biết thêm chi tiết.

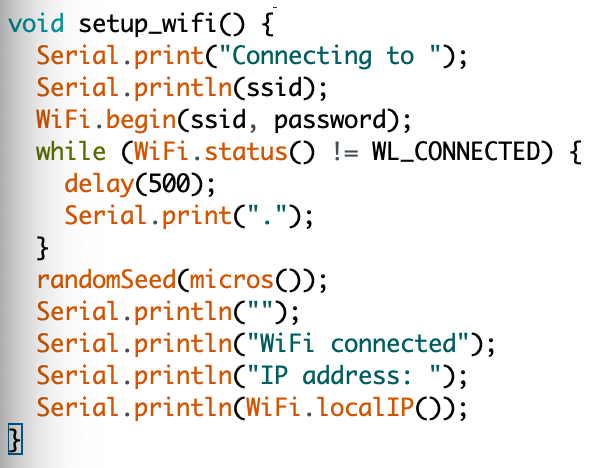
# Triển khai hệ thống

## 3.1 Kết nối Wifi

* Sử dụng thư viện ***#include <WiFi.h>***
* SSID and Password:



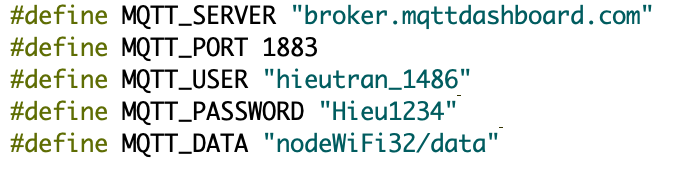
* Hàm Setup Wifi():



Đầu tiên ESP32 sẽ kết nối tới Router bằng phương thức WiFi.begin() với tham số truyền vào là SSID và Password. Tiếp theo sẽ có 1 vòng lặp pooling kiểm tra trạng thái kết nối. Nếu trạng thái kết nối ko rơi vào WL\_CONNECTED mà rơi vào 3 trạng thái còn lại là : WL\_CONNECT\_FAILED, WL\_CONNECTION\_LOST, WL\_DISCONNECTED thì chương trình sẽ trễ đi nửa giây và tiếp tục kiểm tra. Tiếp theo hàm sẽ hiện thị điạ chỉ IP của WFI.

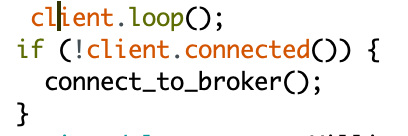
## 3.2 Kết nối đến MQTT

* Định nghĩa các thông số Server, port, topic

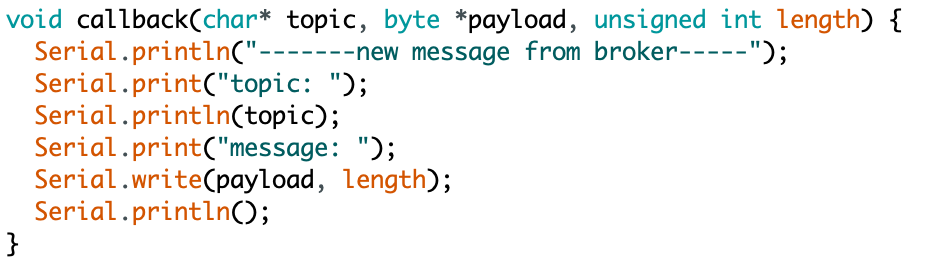


* Hàm kết nối tới Broker





* Hàm callback() là hàm được thực hiện khi ESP32 nhận được message từ topic mà ESP32 subscribe.

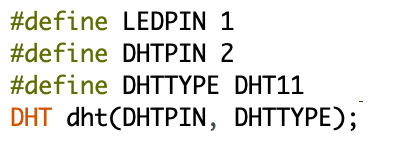




## 3.3 Đọc dữ liệu từ Sensor

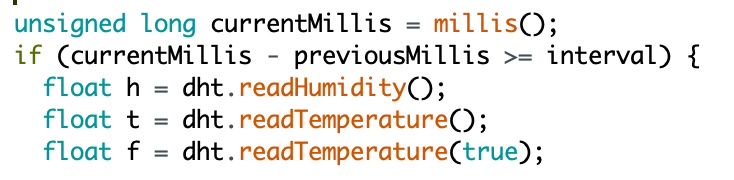
### 3.3.1 Đọc dữ liệu từ DHT11

* Sử dụng thư viện ***#include "DHT.h"***
* Định nghĩa các chân:



* Cứ 5000 ms, thì ESP32 sẽ đọc dữ liệu 1 lần từ sensor DHT11



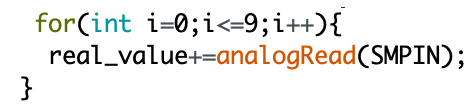


### 3.3.2 Đọc dữ liệu từ cảm biến độ ẩm đất

* Định nghĩa chân



* Chúng ta sẽ tạo một hàm for để đọc 10 lần giá trị cảm biến, sau đó lấy giá trị trung bình để được giá trị chính xác nhất.



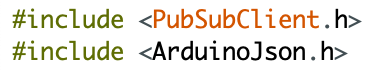
* Set giá thang giá trị đầu và giá trị cuối để đưa giá trị về thang từ 0-100.



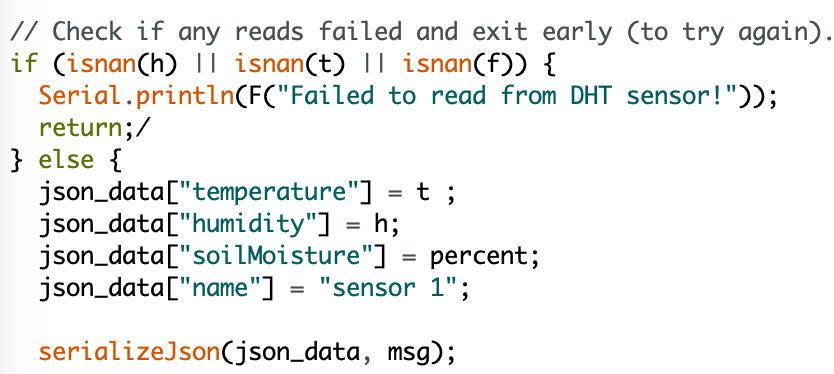
## 3.4 Publish dữ liệu đến MQTT Broker

3.4.1 Publish dữ liệu DHT11 với cảm biến độ ẩm đất lên MQTT Broker

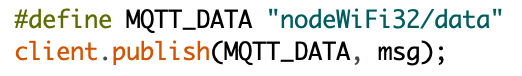
* Sử dụng 2 thư viện



* Đóng gói dữ liệu lại thành JSON

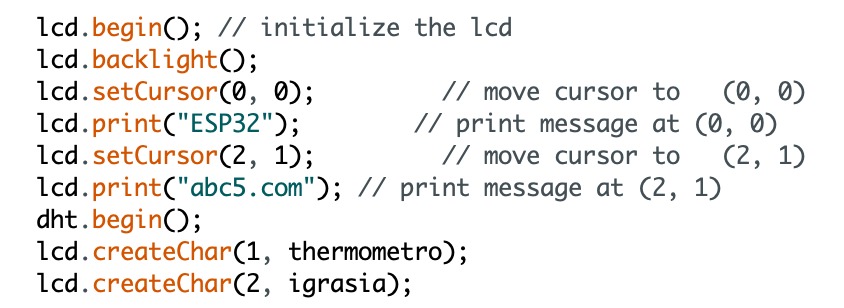


* Publish dữ liệu đến MQTT với TOPIC tương ứng.



## 3.4 Hiện thị dữ liệu ra LCD 1602

* Khởi tạo LCD



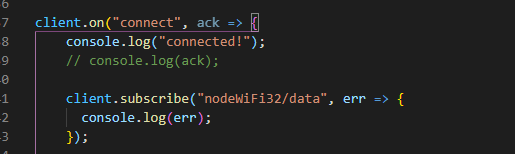
* Hiển thị đèn LCD



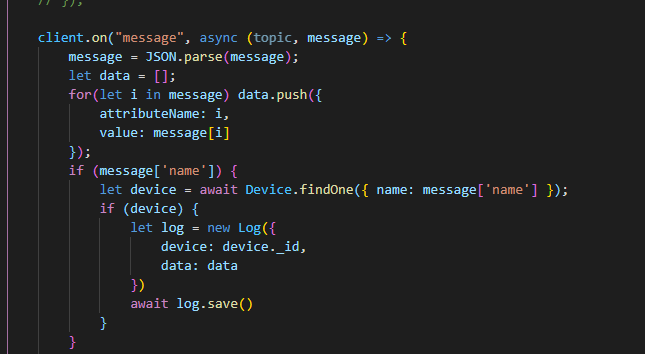
## 3.5. Xây dựng Web Server:

Nhóm sinh viên sử dụng framework NodeJs để xây dựng server cho hệ thống. Phía Server sẽ đảm nhiệm một số chức năng như:

* Subscribe topic trên MQTT để lấy dữ liệu của các sensors được gửi về từ chip ESP32, bao gồm dữ liệu về độ ẩm không khí, độ ẩm đất, nhiệt độ và tên sensor.



Dựa vào tên của sensor, dữ liệu sẽ được lưu vào cơ sở dữ liệu ứng với id của sensor đó.



Hệ CSDL mà nhóm sử dụng là MongoDB, và lưu trữ 4 document:

+) Document accounts liên quan tới tài khoản người dùng (gồm một số trường như email, name, gender, birthday, role, password, createdAt, verifiedAt…),

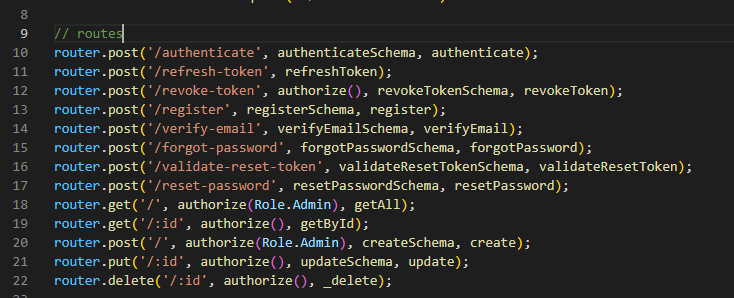
+) Document devices liên quan tới các thiết bị (sensor) được sử dụng (gồm các trường: name, code, topic, type)

+) Document logs lưu trữ lịch sử dữ liệu gửi về từ MQTT (gồm các trường: sensor id, temperature, humidity, soil mosture)

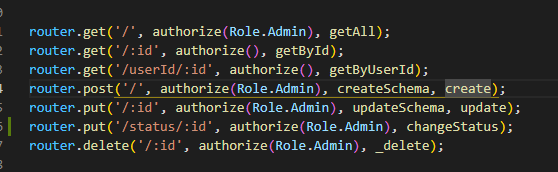
+) Document refreshTokens lưu trữ các token cho 1 phiên session của người dùng.

* Cung cấp một số API hỗ trợ cho phía Web Client như:

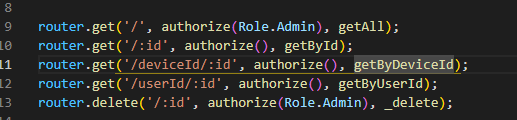
+) Các API về accounts: Tạo/Xoá/Chỉnh sửa/Xác nhận tài khoản…



+) Các API về device: Lấy thiết bị, lấy trạng thái thiết bị, xoá thiết bị…



+) Các API về logs: Lấy lịch sử log, xoá lịch sử log



## 3.6. Xây dựng Client

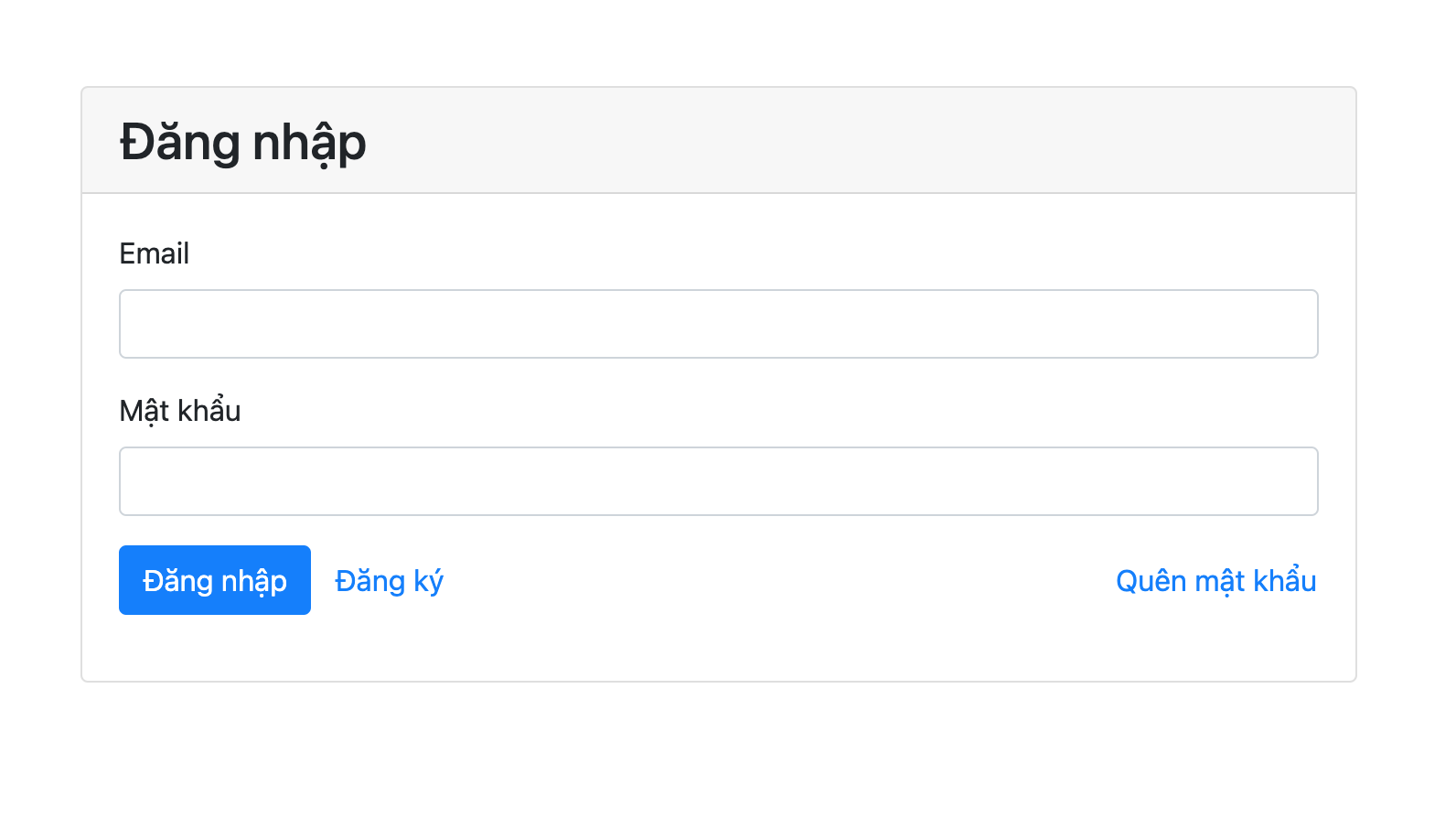
Nhóm sử dụng thư viện ReactJS cùng NPM ( viết tắt của Node package manager - là một công cụ tạo và quản lý các thư viện lập trình Javascript cho [Node.js](https://nodejs.org/) ) để xây dựng giao diện phía người dùng.

Tuy nhiên hệ thống phía Client mới sử dụng react mà không sử dụng redux khiến cho việc mở rộng các chức năng khó khăn => có thể refactor lại code phần client

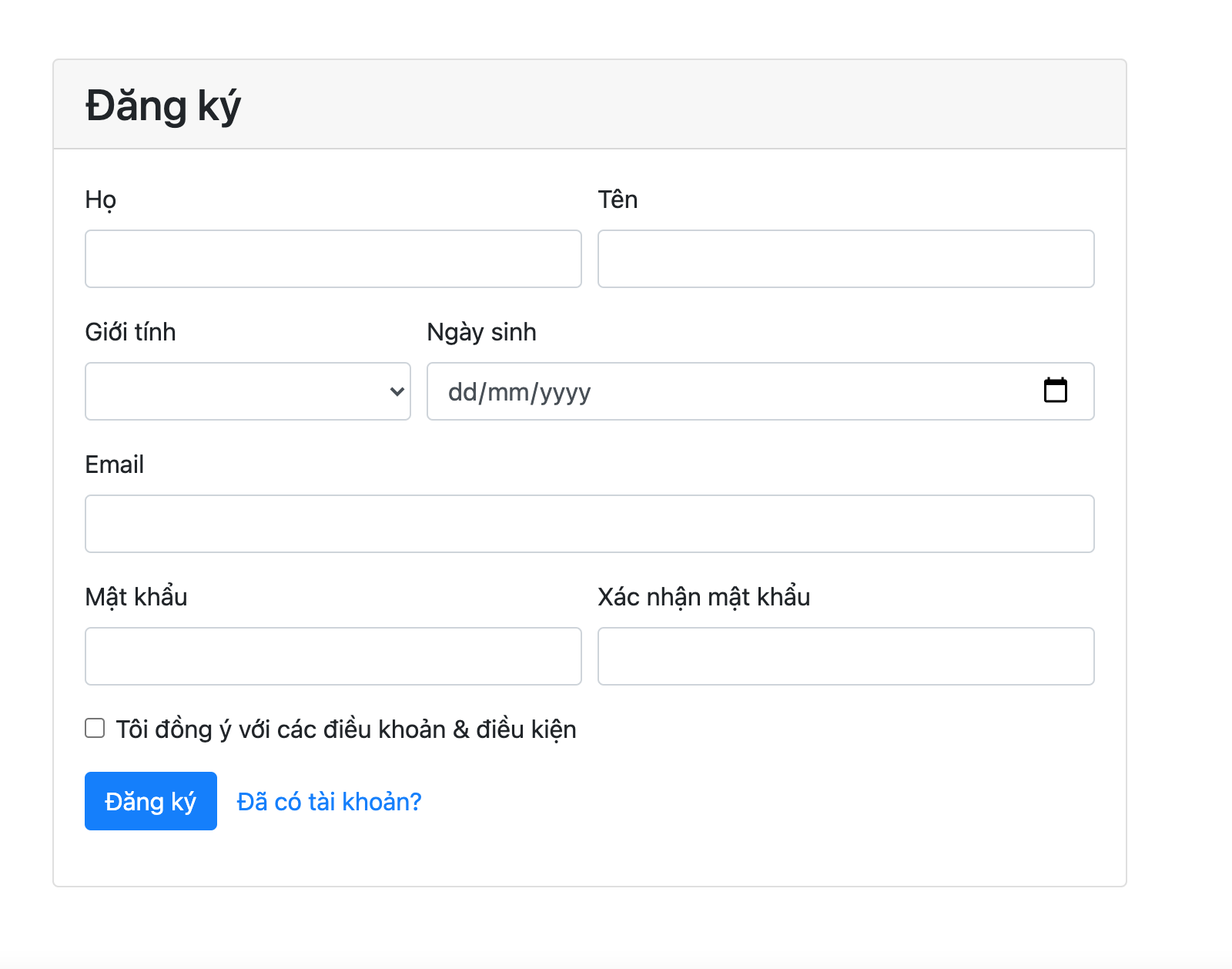
Client sẽ sử dụng các API như đã trình bày ở trên để lấy dữ liệu và thực hiện giao tiếp với server

Các chức năng chính của Client :

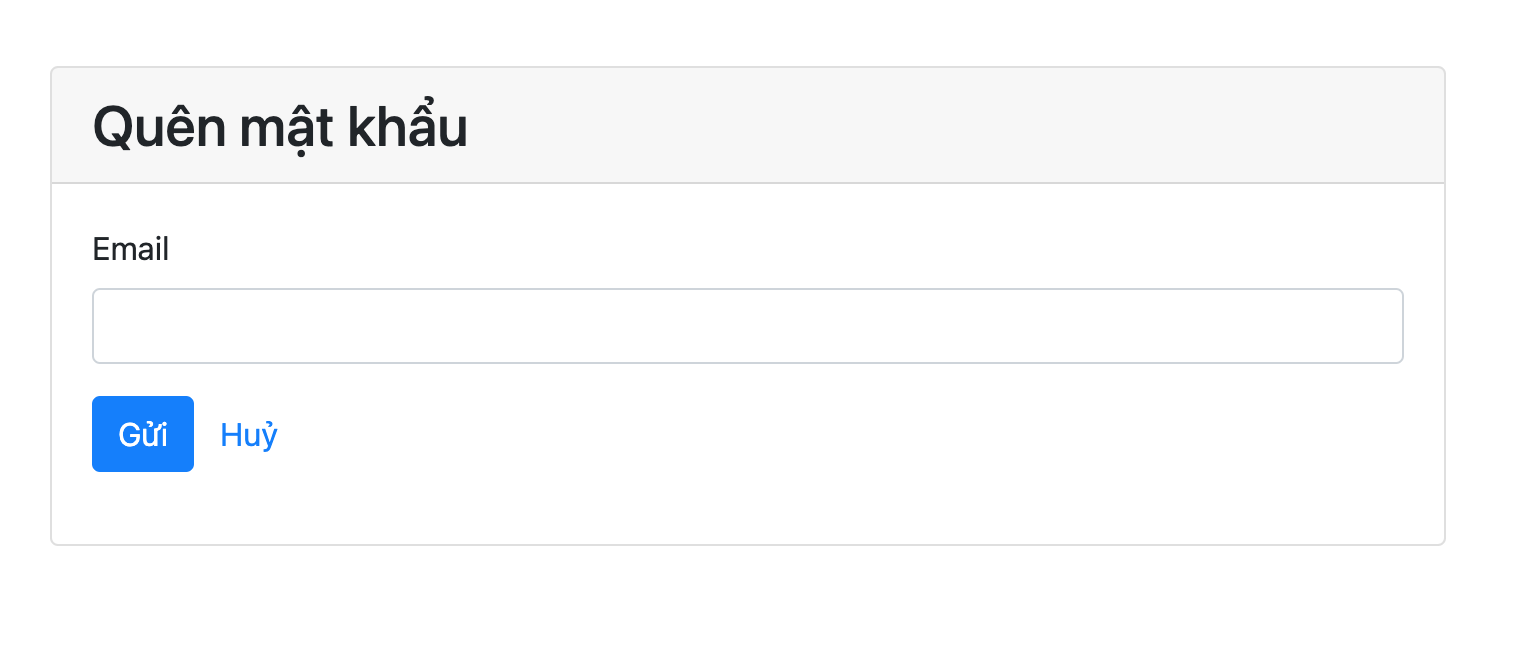
### 3.6.1. Đăng nhập



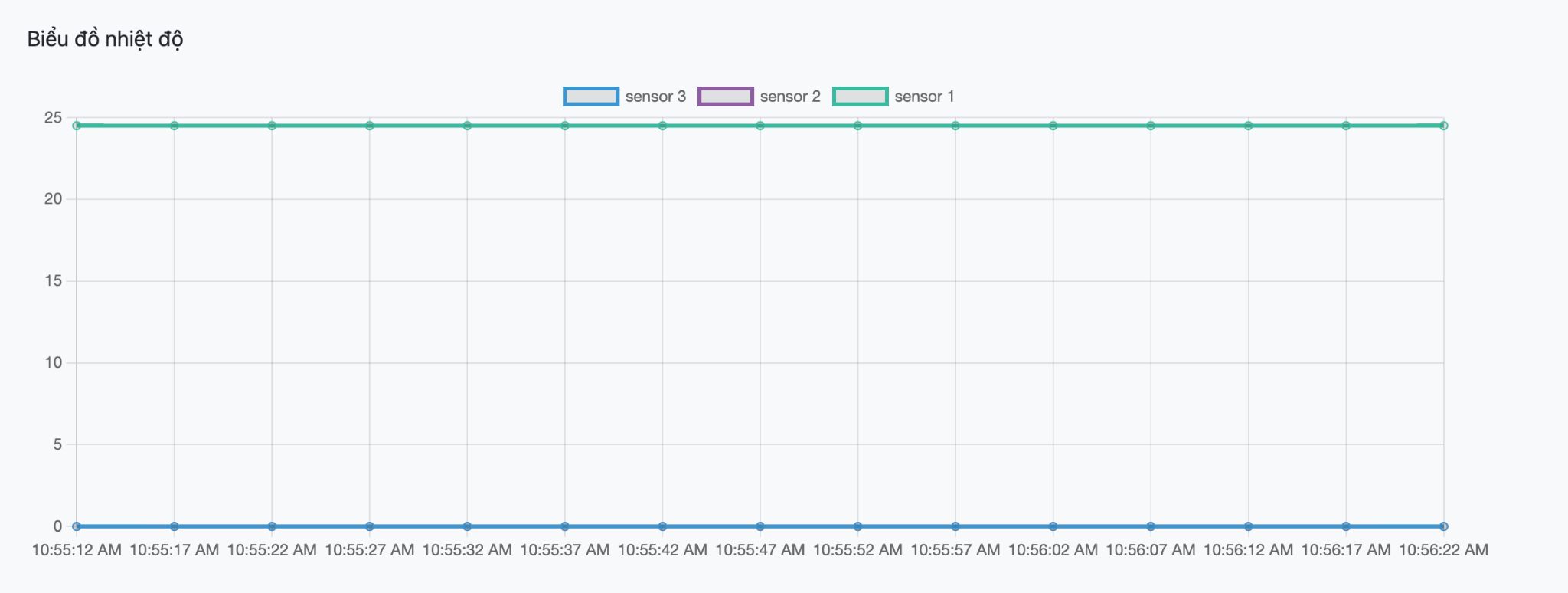
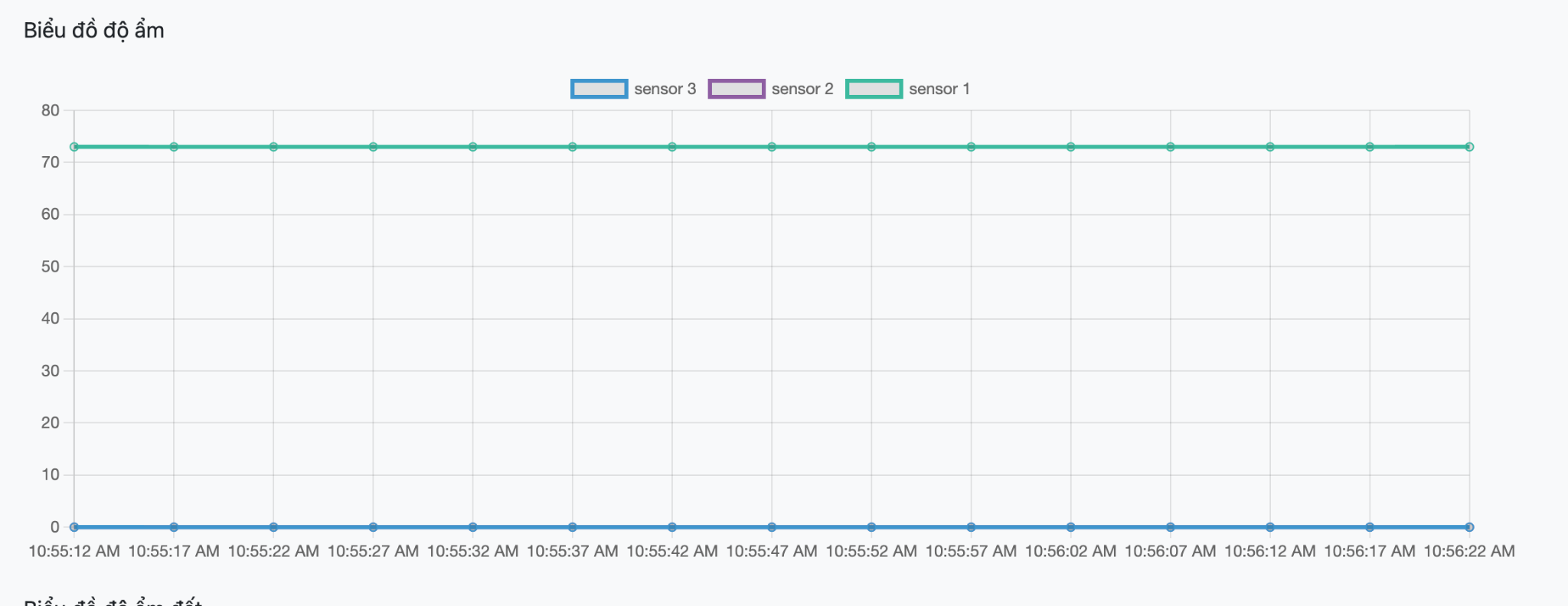
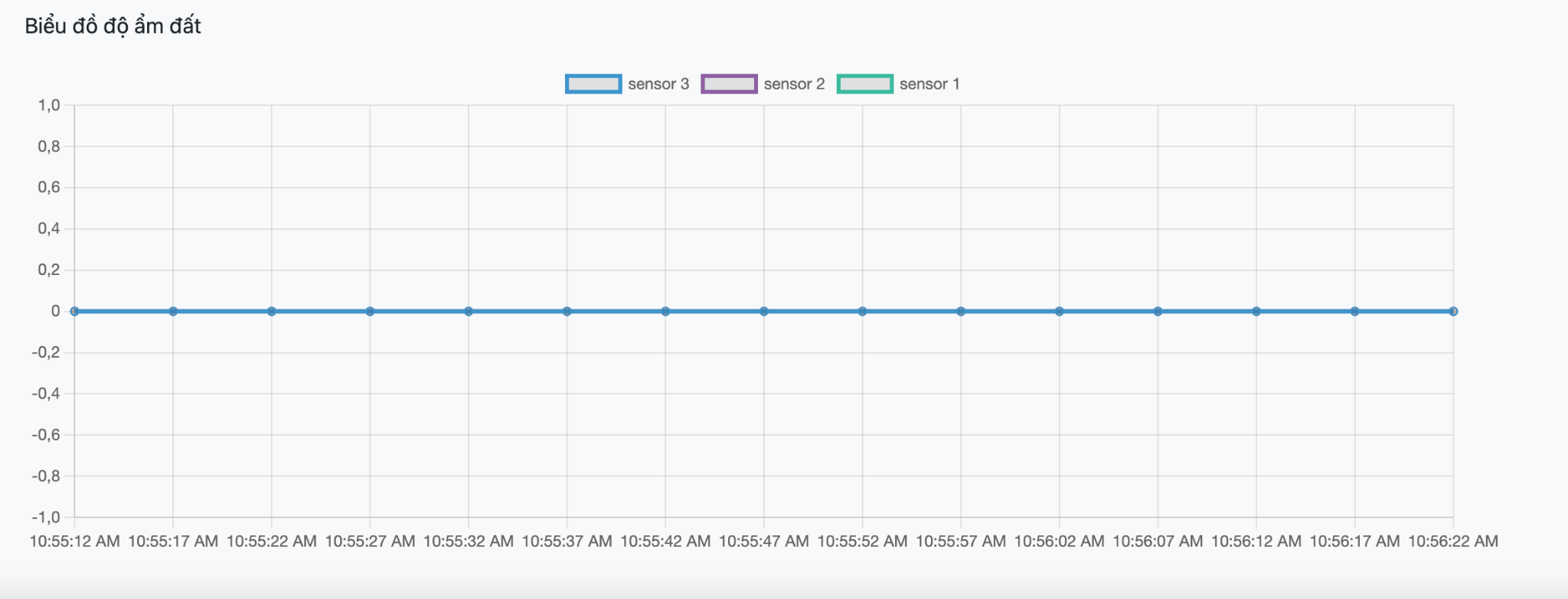
### 3.6.2.Đăng ký



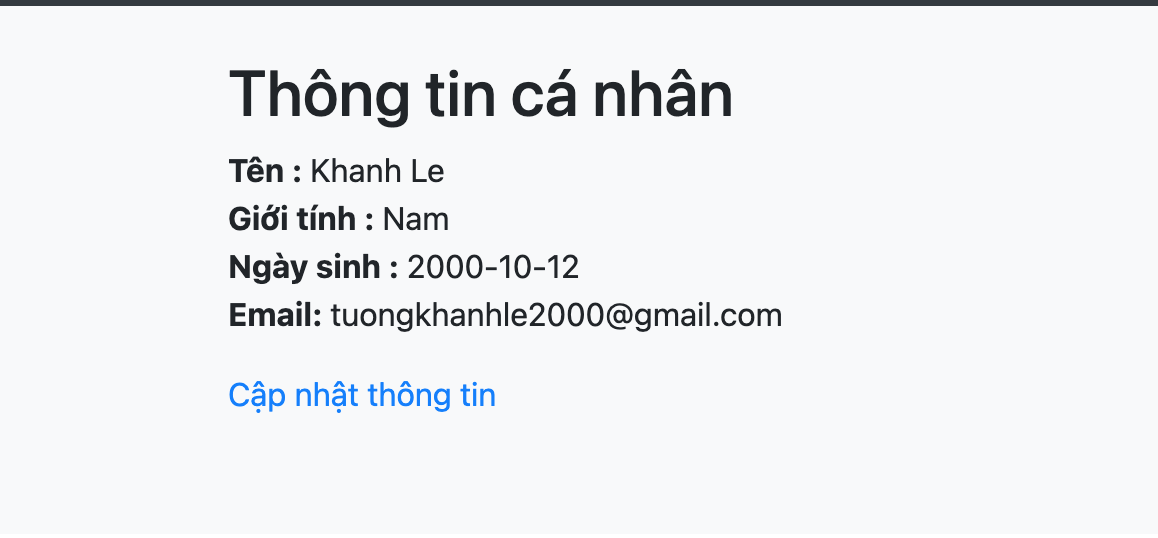
### 3.6.3.Lấy lại mật khẩu



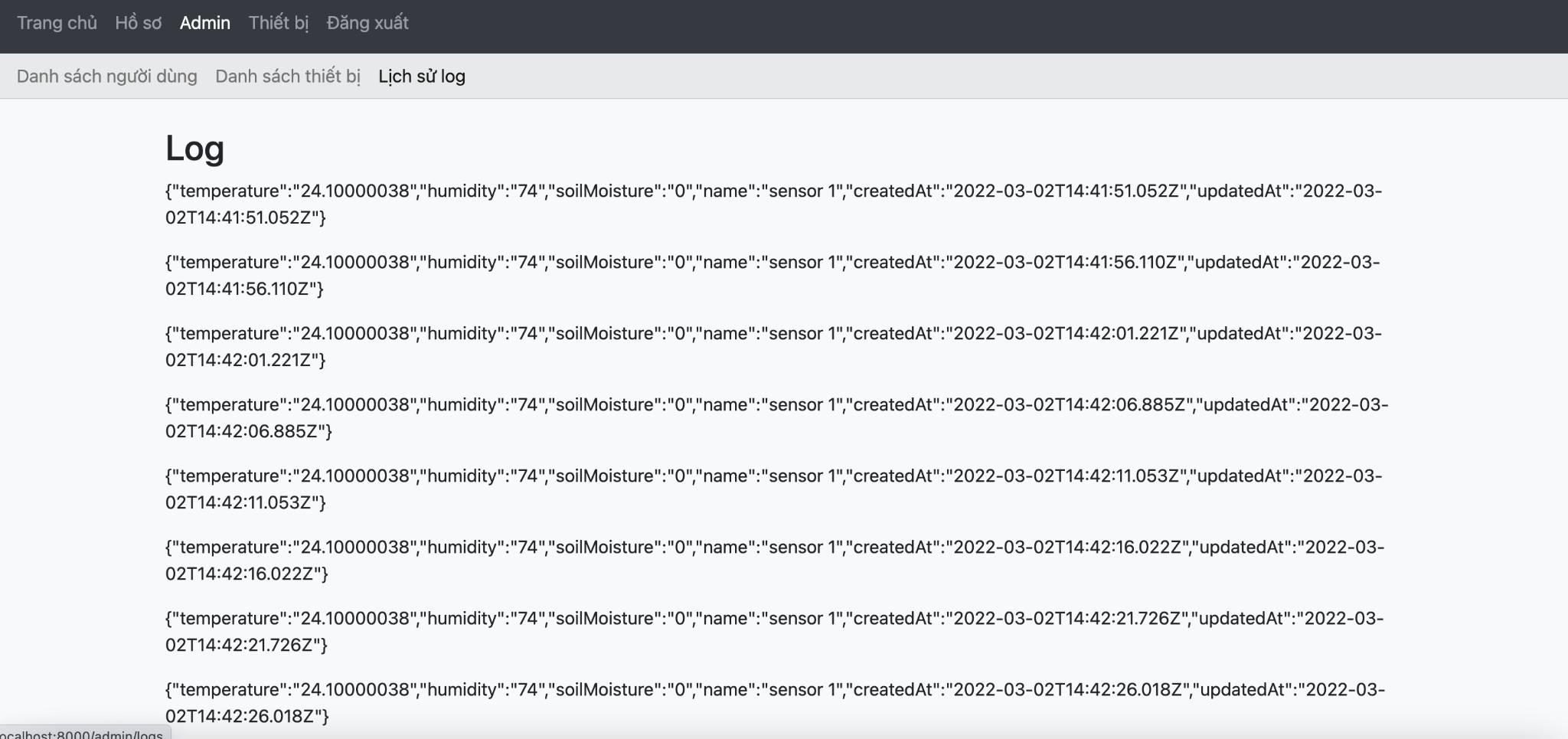
### 3.6.4.Xem biểu đồ về nhiệt độ, độ ẩm, độ ẩm đất, thông tin thời tiết

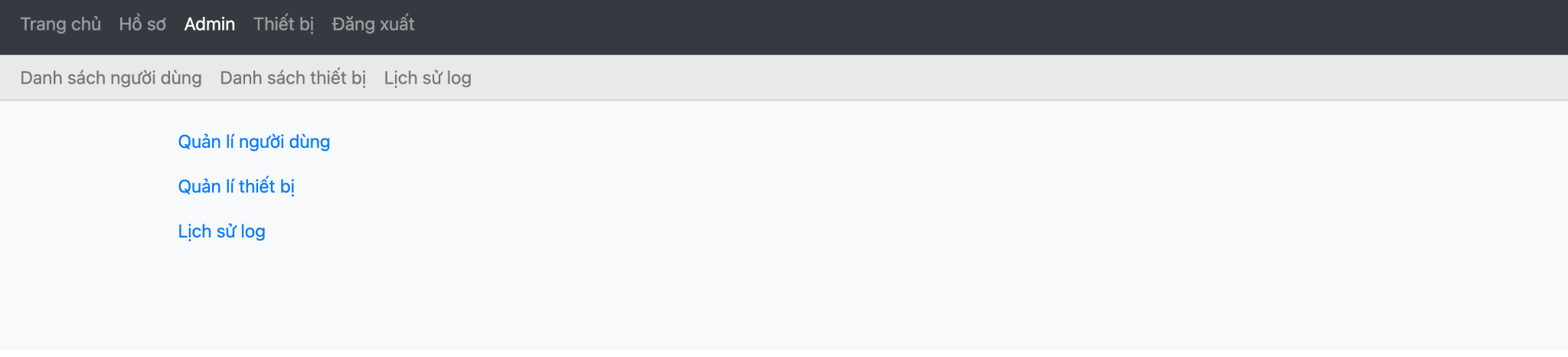


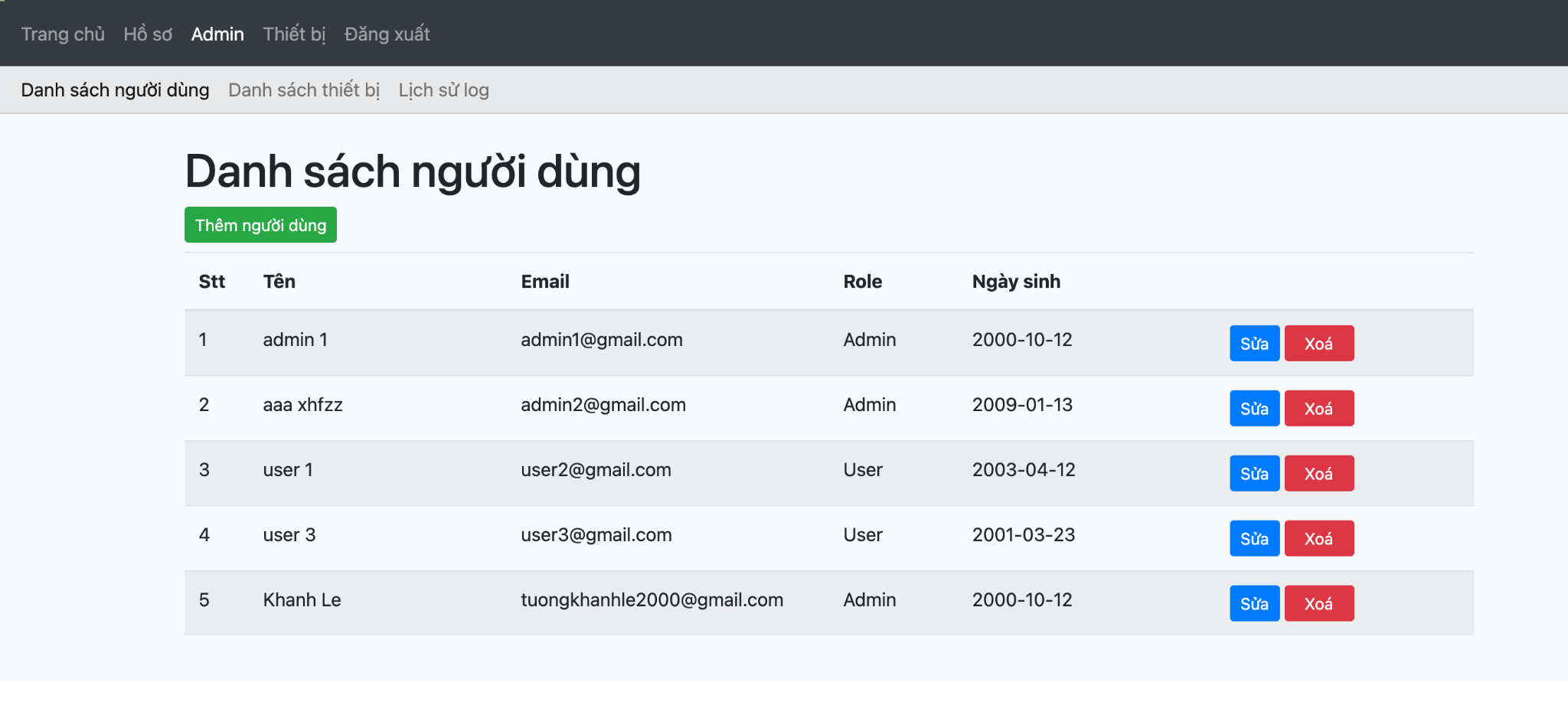
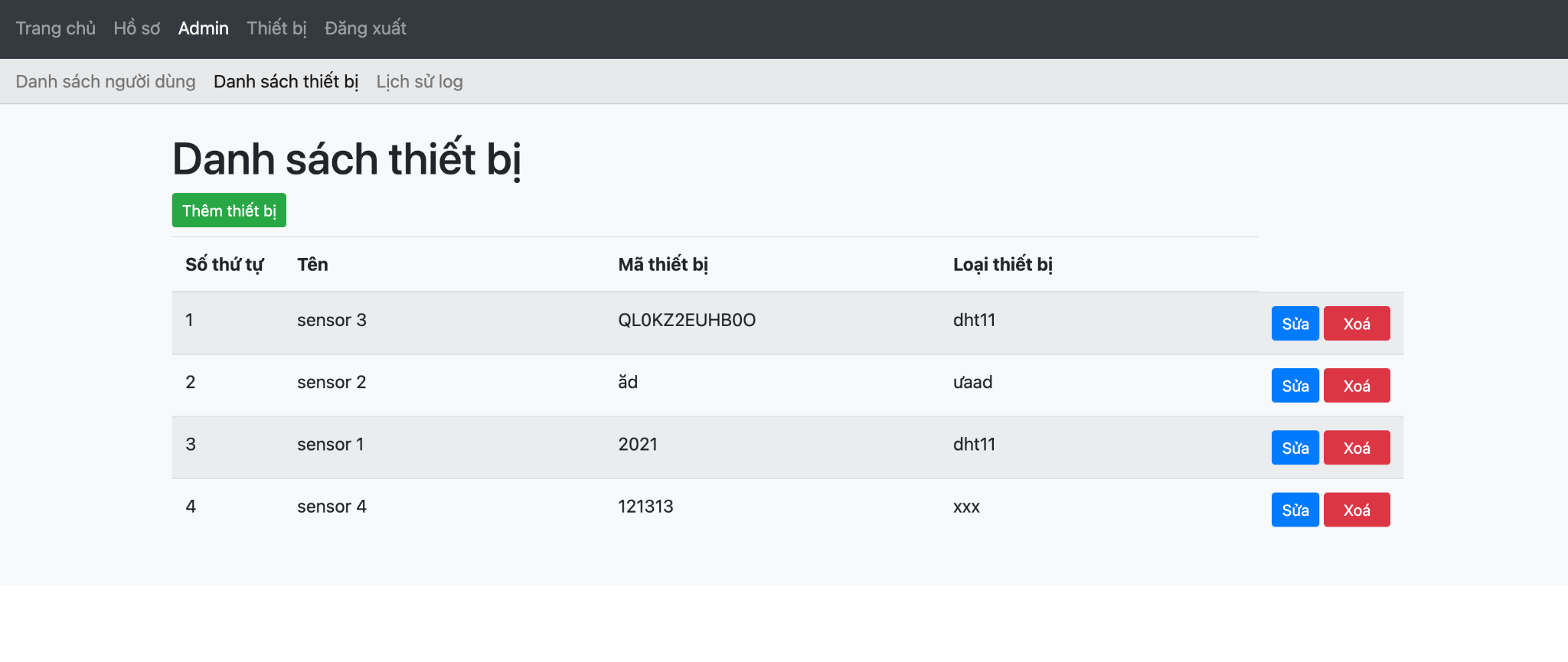
### 3.6.5.Xem thông tin cá nhân



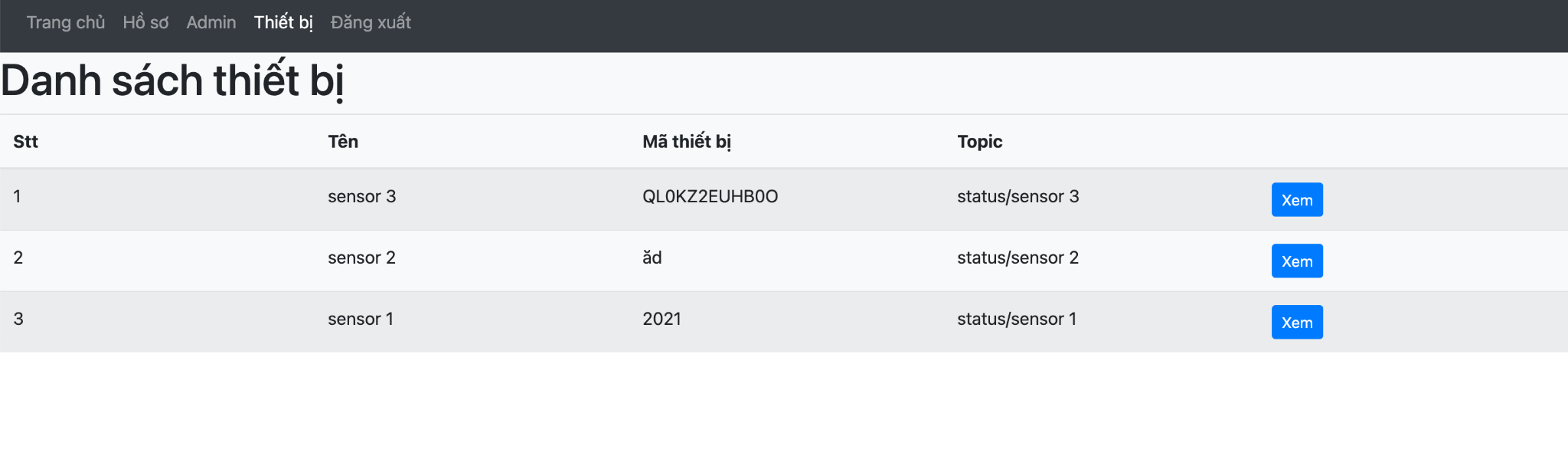
### 3.6.6.Quản lý người dùng, thiết bị và xem lịch sử toàn bộ log của các sensor (Chức năng chỉ dành cho admin) :

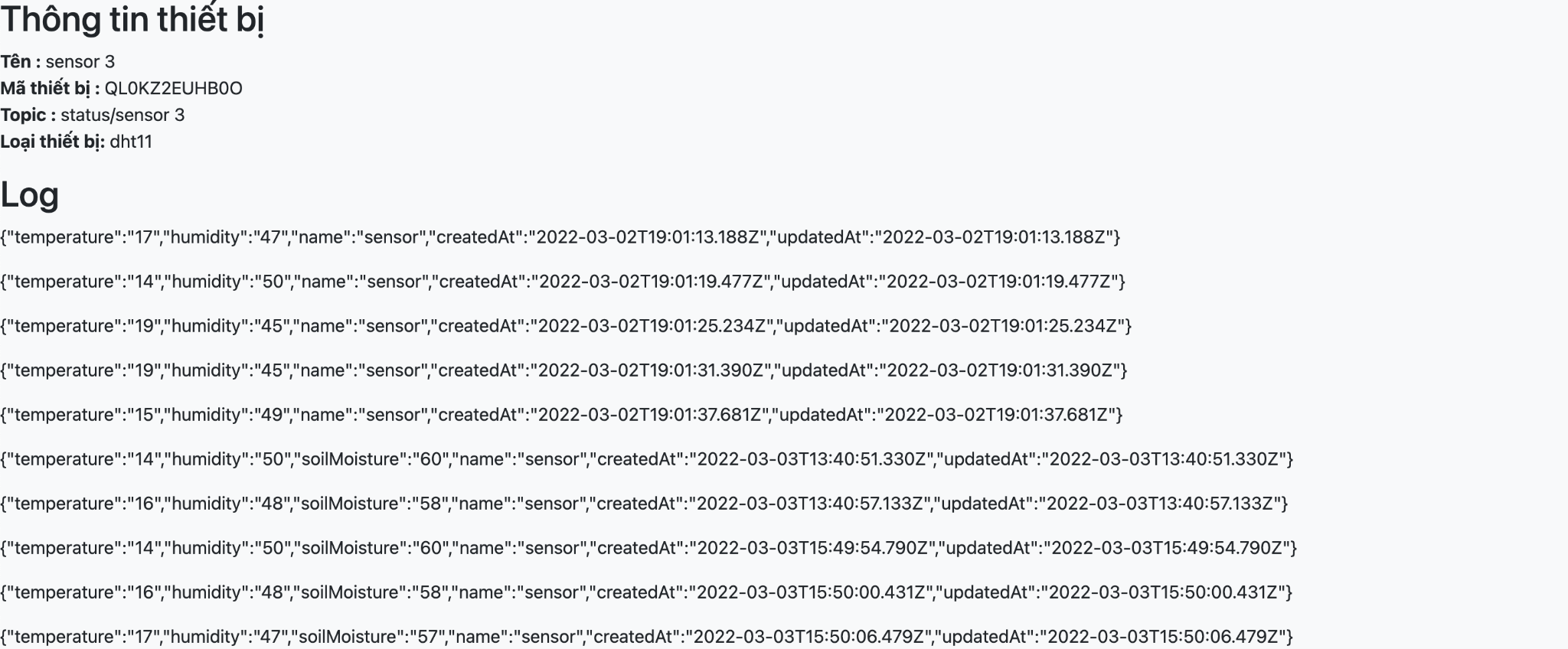






### 3.6.7.Xem thông tin và log của sensor mà người dùng được phép truy cập





# Tổng kết

## 4.1. Kết quả thu được:

Hệ thống IoT theo đề xuất giúp người nông dân có thể linh động hơn trong việc xử lý các vấn đề của nơi canh tác. Giờ đây, nhờ có hệ thống, các công việc như tưới tiêu, chăm bón hay điều khiển lưới bảo vệ có thể thực hiện được chỉ bằng các thao tác trên các thiết bị di động hoặc máy tính.

## 4.2. Hướng phát triển:

Tuỳ từng loại cây có các yêu cầu lượng nước, nhiệt độ khác nhau nhưng về cơ bản, hệ thống nông nghiệp thông minh bao gồm các thành phần được nêu trên, người dùng hoàn toàn có thể sử dụng mạng local để tăng tính bảo mật.

Còn về hướng phát triển tương lai thì hệ thống có thể bổ sung nhiều chức năng, nhận diện loại cây trồng, tăng độ chính xác của cảm biến, đưa ra các gợi ý và cảnh báo khi cây trồng có dấu hiệu bị sâu bệnh…

## 4.3. Kết luận:

Với việc áp dụng IoT vào nông nghiệp, người nông dân có thể tiết kiệm được thời gian công sức trong lao động sản xuất nông nghiệp.

Để tạo ra nền nông nghiệp bền vững, việc sử dụng IoT sẽ là trung tâm hàng đầu trong các hoạt động nông nghiệp. IoT sắp xếp hợp lý về cách làm việc từ sử dụng điện và nước, vận chuyển cây trồng, cảnh báo vận hành, bảo trì máy móc nông trại. IoT đã chứng tỏ một bước đột phá và tiếp tục thay đổi cách nhìn vào các hoạt động khác nhau, trên thế giới ước tính có hơn 75 triệu thiết bị dựa trên IoT hoạt động trong ngành nông nghiệp vào năm 2020. Trong tương lai, IoT có thể định hình bởi tiến bộ vượt bậc trong Mạng cảm biến không dây WSN và thế hệ thứ 5 của công nghệ thông tin di động 5G để cung cấp cho nông dân dữ liệu và thông tin theo thời gian thực mọi lúc mọi nơi trên đất của họ.s

# Tài liệu tham khảo

1. <https://123docz.net/document/5069402-ung-dung-iot-vao-he-thong-vuon-nha-kinh.html>.
2. <https://bkaii.com.vn/tin-tuc/168-ung-dung-iot-trong-nong-nghiep-cong-nghe-cao>.
3. <https://reactjs.org>
4. <https://nodejs.org/en/docs/>
5. https://www.emqx.com/en/blog/how-to-use-mqtt-in-nodejs