

4

CÔNG NGHỆ  
INTERNET OF THINGS  
HIỆN ĐẠI

# Xây dựng giao diện ứng dụng để tương tác với mô hình IoT

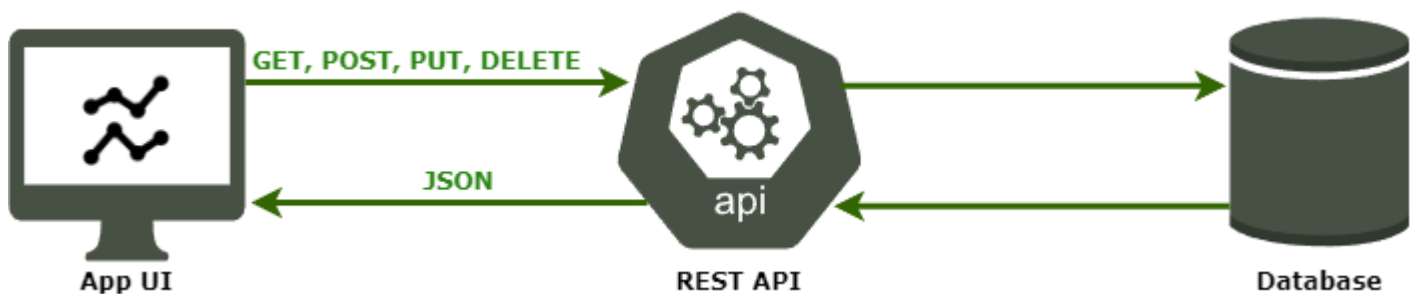
Lưu hành nội bộ

## A. MỤC TIÊU

- Giới thiệu tổng quan về lập trình ứng dụng cơ bản để hiển thị dữ liệu IoT.
- Giới thiệu các thành phần App UI, API, Database trong mô hình IoT.
- Tổng quan về việc xây dựng, kết hợp các thành phần để tạo nên một mô hình IoT cơ bản hoàn chỉnh.

## B. GIỚI THIỆU

### 1. Tổng quan mô hình thực hiện



Hình 1. Mô hình tổng quan

Tại bài thực hành này, sinh viên xây dựng ứng dụng để hiển thị và điều khiển trong mô hình IoT gồm có 3 thành phần chính:

- **App UI:** thành phần dùng để thể hiện các giá trị cảm biến, thực hiện các tác vụ liên quan đến điều khiển thiết bị và hiển thị logs cho người quản trị theo dõi.
- **API:** tương tự như ở bài thực hành trước đã giới thiệu, đây là thành phần trung gian để người dùng có thể tương tác các thành phần ở layer cao hơn như như cơ sở dữ liệu, trung tâm xử lý các tác vụ tính toán,...
- **Database:** cơ sở dữ liệu của hệ thống, giúp cấu trúc các dữ liệu, lưu trữ để phục vụ nhu cầu khai thác thông tin của nhiều người sử dụng từ một hoặc nhiều ứng dụng cùng lúc với nhiều mục đích khác nhau.

## 2. App/Web UI

UI là viết tắt của từ User Interface có nghĩa là giao diện người dùng. Hiểu một cách đơn giản nhất thì UI bao gồm tất cả những gì người dùng có thể nhìn thấy như: màu sắc ứng dụng, bố cục sắp xếp như thế nào, fonts chữ, hình ảnh, biểu đồ,...

**IoT platform<sup>1</sup>** là một dạng phần mềm trung gian, kết nối giữa phần cứng và các ứng dụng. IoT platform có nhiệm vụ thu thập và trao đổi dữ liệu qua các giao thức, cấu trúc mạng liên kết khác nhau. IoT platform được sử dụng để quản lý và kiểm soát các thiết bị kết nối IoT, thu thập dữ liệu từ chúng, và thực hiện các hoạt động tự động hóa dựa trên dữ liệu thu thập. Một số chức năng chính của các nền tảng IoT:

- **Quản lý thiết bị:** Các nền tảng IoT cho phép quản lý, theo dõi và điều khiển các thiết bị IoT từ xa. Có thể thêm mới thiết bị, xóa thiết bị, cập nhật phần mềm của thiết bị, và theo dõi tình trạng của chúng.
- **Thu thập dữ liệu:** Các nền tảng IoT có khả năng thu thập dữ liệu từ các thiết bị IoT. Dữ liệu này có thể bao gồm thông tin về nhiệt độ, độ ẩm, áp suất, vị trí, tình trạng hoạt động, và nhiều thông tin khác.
- **Trực quan hóa dữ liệu:** Các nền tảng IoT cung cấp công cụ để biểu đồ, biểu đồ và hiển thị dữ liệu thu thập dưới dạng giao diện trực quan. Điều này giúp người dùng dễ dàng theo dõi và phân tích dữ liệu IoT.
- **Bảo mật:** Các nền tảng IoT cung cấp các tính năng bảo mật để bảo vệ dữ liệu và thiết bị khỏi các mối đe dọa trực tuyến. Điều này bao gồm xác thực, mã hóa dữ liệu và kiểm soát quyền truy cập.

<sup>1</sup> <https://smartindustry.vn/technology/internet-of-things/iot-platform-la-gi-co-nhung-loai-iot-platform-nao/>

- **Hỗ trợ giao thức:** Các nền tảng IoT hỗ trợ nhiều giao thức kết nối, bao gồm MQTT, CoAP, HTTP, và nhiều giao thức khác để giao tiếp với các thiết bị IoT.

Một số IoT platform mã nguồn mở:



### 3. API

**API** là viết tắt của Application Programming Interface – phương thức trung gian kết nối các ứng dụng và thư viện khác nhau. **REST** (**RE**presentational **S**tate **T**ransfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API.

**RESTful API** là một tiêu chuẩn dùng trong việc thiết kế các API cho các website hoặc ứng dụng để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các phương thức HTTP (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web, app để quản các resource<sup>2</sup>.

### 4. Database

#### **Giới thiệu cơ sở dữ liệu SQL**

**SQL** là viết tắt của **Structured Query Language** có nghĩa là ngôn ngữ truy vấn có cấu trúc, là một ngôn ngữ máy tính để lưu trữ, thao tác và truy xuất dữ liệu được lưu trữ trong một cơ sở dữ liệu quan hệ.

<sup>2</sup> Xem thêm tại: <https://topdev.vn/blog/restful-api-la-gi/>

SQL là ngôn ngữ chuẩn cho hệ cơ sở dữ liệu quan hệ. Tất cả các hệ thống quản lý cơ sở dữ liệu quan hệ (RDMS) như MySQL, MS Access, Oracle, Sybase, Informix, Postgres và SQL Server đều sử dụng SQL làm ngôn ngữ cơ sở dữ liệu chuẩn.

### Giới thiệu cơ sở dữ liệu NoSQL

NoSQL là một hệ thống quản lý dữ liệu không quan hệ (**non-relational Data Management System**) để dàng mở rộng. Mục đích chính của việc sử dụng cơ sở dữ liệu NoSQL là dành cho các kho dữ liệu phân tán với nhu cầu lưu trữ dữ liệu lớn. NoSQL được sử dụng cho Dữ liệu lớn và ứng dụng web thời gian thực.

Hiện nay, trên thị trường có khá nhiều NoSQL Database Management System: MongoDB, RavenDB, Neo4j,... Trong đó nổi bật có 4 loại như sau:

- Key-Value Database
- Document Database
- Column-Family Database
- Graph Database

## C. YÊU CẦU & NỘI BÀI

### 1. Yêu cầu

*Lựa chọn 1 nền tảng IoT platform mã nguồn mở và thực hiện xây dựng ứng dụng IoT cơ bản đầy đủ các thành phần và chức năng, bao gồm:*

1. **Device:** sinh viên thực hiện sử dụng 2 loại thiết bị cảm biến DHT22/DHT11 và BH-1750 để lấy 3 loại dữ liệu cảm biến là nhiệt độ, độ ẩm và ánh sáng. Trên Wemos D1 sử dụng cảm biến BH-1750, đồng thời lập trình để gửi các giá trị cảm biến đến hệ thống thông qua giao thức HTTP/MQTT. Trên Raspberry Pi sử dụng DHT22/DHT11, đồng thời lập trình để gửi các giá trị cảm biến đến hệ thống thông qua giao thức HTTP/MQTT. Các dữ liệu này cũng được lưu trữ tại hệ thống IoT platform. Giao thức tùy vào IoT platform cung cấp.
2. **Dashboard:** hiển thị tình trạng thiết bị đang kết nối đến hệ thống (Wemos D1, Raspberry Pi,...).

**Lưu ý:** số lượng thiết bị kết nối đến hệ thống phải từ 2 trở lên (2 Wemos D1,...).

**3. Màn hình chính:** dùng để thể hiện các tương tác với hệ thống, bao gồm: thể hiện các giá trị cảm biến ở thời điểm hiện tại (tên giá trị, giá trị của cảm biến), chức năng điều khiển 2 đèn LED trên Wemos D1 thông qua giao diện ứng dụng. Chức năng này phải thể hiện thời gian thực.

**Gợi ý:** sử dụng MQTT thông qua IoT platform để điều khiển đèn.

**4. Biểu đồ:** dùng để thể hiện các giá trị cảm biến thu thập được từ phần thiết bị một cách trực quan bằng biểu đồ, bảng biểu. Cụ thể như sau: các giá trị cảm biến bao gồm 3 loại (nhiệt độ, độ ẩm, ánh sáng). Các biểu đồ này thể hiện dưới dạng thời gian thực.

**5. Logs:** thể hiện các log của hệ thống để người quản trị có thể dễ dàng tương tác, quản lý đối với hệ thống. Cụ thể như sau: ghi lại các thông tin của hệ thống, giá trị cảm biến từ phần thiết bị gửi đến, các trường dữ liệu cần phải có: ID của thiết bị, tên thiết bị, tên giá trị, giá trị cảm biến, thời gian nhận dữ liệu,...

**6. (Nâng cao)** thực hiện triển khai IoT platform đó ở local.

**7. (Nâng cao)** Người dùng có thể lọc lại các dữ liệu mà họ mong muốn hoặc download toàn bộ dữ liệu dưới các định dạng như csv, excel,...

## 2. Yêu cầu nộp bài

- Sinh viên tìm hiểu và thực hành theo hướng dẫn. Thực hiện **nhóm**.
- Sinh viên báo cáo kết quả thực hiện và nộp bài bằng file. Trong đó:
  - Trình bày chi tiết quá trình thực hành và trả lời các câu hỏi nếu có (kèm theo các ảnh chụp màn hình tương ứng).
  - Giải thích các kết quả đạt được.
  - Tải mẫu báo cáo thực hành và trình bày theo mẫu được cung cấp.

- Các thành viên trong nhóm tiến hành nộp source code thông qua GitHub bằng việc tập làm quen quản lý repository với git đã được thống nhất tại buổi thực hành số 1.

Nén tất cả các file và đặt tên file theo định dạng theo mẫu:

**NhomY-LabX\_MSSV1\_MSSV2**

Ví dụ: Nhom1-Lab04\_25520001\_25520002

- Nộp báo cáo trên theo thời gian đã thống nhất tại website môn học.
- Không “gánh team”, copy bài của người / nhóm khác. Trong trường hợp có tham khảo các nội dung trên Internet thì trích dẫn nguồn vào báo cáo. Nếu vi phạm các hành vi này sẽ bị xử lý tùy theo mức độ.
- Các bài nộp không tuân theo yêu cầu sẽ **KHÔNG** được chấm điểm.

**HẾT**

**Chúc các bạn hoàn thành tốt bài thực hành!**