⚡️BÁO CÁO TIẾN ĐỘ THỰC TẬP⚡️

⚡️Phạm Hoàng Tuấn⚡️

## ✏️ ***Câu 1: Mô hình Microservice ?***

* Một ứng dụng được chia thành nhiều “thành phần” độc lập về mặt dữ liệu, đủ nhỏ về kích cỡ và đủ độ ảnh hưởng trong hệ thống.
* Mỗi microservice có khả năng độc lập, đảm báo lắp vào hoặc tháo ra khỏi hệ thống không làm ảnh hưởng đến các thành phần khác
* Có 4 đặc trưng: micro-service, tính độc lập, tính chuyên biệt, phòng chống lỗi.

## ✏️ ***Câu 2: So sánh kiến trúc Ứng dụng nguyên khối và Microservice ?***

| Nguyên khối | Microservice |
| --- | --- |
| Toàn bộ ứng dụng là một khối | Toàn bộ ứng dụng được chia thành nhiều thành phần |
| Triển khai và phát tiển ứng dụng khá đơn giản | Triển khai ứng dụng phức tạp do có nhiều thành phần |
| Các lập trình viên phải tuân thủ các quy tắc và tiêu chuẩn chung | Lập trình viên tuân thủ các quy tắc và tiêu chuẩn theo mỗi thành phần |
| Tốn kém thời gian sửa chữa, bảo trì trên một hệ thống lớn | Tiết kiệm thời gian chỉ cần sửa chữa, bảo trì trên một thành phần |

## ✏️ ***Câu 3: So sánh SQL Database và NoSQL Database***

| SQL Database | NoSQL Database |
| --- | --- |
| Dữ liệu chuẩn hóa thành các bảng, bảng ghi(hàng), thuộc tính(cột) | Dữ liệu được chuẩn hóa thành JSON |
| Có sự ràng buộc giữa các các bảng(khóa chính, khóa ngoại), điều kiện của dữ liệu | Không có sự ràng buộc về mặt dữ liệu |
| Phù hợp với các hệ thống yêu cầu chặt chẽ về mặt dữ liệu | Phù hợp với các hệ thống chỉ yêu cầu lưu trữ dữ liệu |

## ✏️ ***Câu 4: Tìm hiểu về Angular***

[**Angular**](https://angular.io/) là một Javascript Framework dùng để viết giao diện Web, xây dựng các Single Page Application(SPA)

### [Ưu điểm của Angular](https://angular.io/)

* Cú pháp mã lệnh đơn giản hơn Javascript thuần
* Phân chia layout thành các component, tăng khả năng tái sử dụng và sửa lỗi
* Tạo ra các Single Page Application (SPA)