# BÁO CÁO THỰC TẬP TUẦN 1

## Nguyễn Thị Hồng Đào

## MSSV: B1809115

### Câu 1: Angular là gì? Lợi ích của Angular so với javascript thuần

* Angular được xe là một open source (mã nguồn mở) hay frameworks miễn phí chuyên dụng cho công việc thiết kế giao diện web. Angular được phát triển từ những năm 2009 và được duy trì bởi Google. Frameworks này được xem là frameworks frontend mạnh mẽ nhất chuyên dụng bởi các lập trình viên cắt HTML cao cấp.
* Lợi ích của Angular:
  + Angular được “chống lưng” bởi Google, giúp cho Developer có cảm giác được đảm bảo. Mặc nhiên, họ sẽ ám thị rằng framework này khó mà bị “khai tử”, vì vậy cứ yên tâm sử dụng.
  + Cộng đồng người dùng lớn nên nếu có thắc mắc gì cũng sẽ nhanh chóng được giải đáp.
  + Giúp phát triển Ứng dụng trang đơn (Single-page Application). Đây là ứng dụng chạy trên browser mà không bắt buộc phải tải lại trang khi sử dụng.

### Câu 2: So sánh CSDL quan hệ và CSDL không quan hệ

| **Tiêu chí** | **Cơ sở dữ liệu quan hệ** | **Cơ sở dữ liệu không quan hệ** |
| --- | --- | --- |
| Ngôn ngữ Query | Structured Query Language | Không có ngôn ngữ Query |
| Loại hình | SQL databases là cơ sở dữ liệu dựa trên bảng | NoSQL databases có thể dựa trên tài liệu, cặp khóa-giá trị, cơ sở dữ liệu biểu đồ |
| Khả năng mở rộng | Có thể mở rộng theo chiều dọc | Có thể mở rộng theo chiều ngang |
| Lưu trữ dữ liệu phân cấp | Không thích hợp cho việc lưu trữ dữ liệu phân cấp. | Phù hợp cho kho lưu trữ dữ liệu phân cấp vì nó hỗ trợ phương thức cặp khóa-giá trị |
| Mục đích sử dụng | Được thiết kế dành cho các ứng dụng xử lý giao dịch trực tuyến trong giao dịch có độ ổn định cao và thích hợp để xử lí phân tích trực tuyến | Được thiết kế để phục vụ phân tích dữ liệu có cấu trúc chưa hoàn chỉnh. |
| Lựa chọn sử dụng | Dự án đã có yêu cầu dữ liệu rõ ràng xác định quan hệ logic có thể được xác định trước | Dự án yêu cầu dữ liệu không liên quan, khó xác định, đơn giản mềm dẻo khi đang phát triển |
| Mã nguồn | Một sự kết hợp của mã nguồn mở như Postgres & MySQL, và thương mại như Oracle Database. | Open-source |

### Câu 3: Cơ sở dữ liệu phân tán

Cơ sở dữ liệu phân tán là một tập hợp dữ liệu có liên quan với nhau về mặt logic được dùng chung nhưng được lưu trữ phân tán về mặt vật lý với nhau.

* Một hệ QTCSDL phân tán là một hệ thống phần mềm cho phép quản trị CSDL phân tán và làm cho người sử dụng không nhận thấy sự phân tán về lưu trữ dữ liệu.
* Người dùng truy cập vào cơ sở dữ liệu phân tán thông qua chương trình ứng dụng. Có 2 loại:
  + Chương trình không yêu cầu dữ liệu từ nơi khác,
  + Chương trình có yêu cầu dữ liệu từ nơi khác.
* Có thể chia các hệ cơ sở dữ liệu phân tán thành 2 loại chính:
  + ***Hệ CSDL phân tán thuần nhất:*** các nút trên mạng đều cùng một hệ QTCSDL.
  + ***Hệ CSDL phân tán hỗn hợp:*** các nút trên mạng có thể dùng các hệ QTCSDL khác nhau.

### Câu 4: So sánh kiến trúc Micro service và kiến trúc ứng dụng nguyên khối (Monolithic application)

Microservice là một loại kiến trúc phần mềm hướng dịch vụ, tập trung vào việc xây dựng một loạt các thành phần tự quản lý tạo nên ứng dụng. Không giống như các ứng dụng nguyên khối được xây dựng dưới dạng một đơn vị không thể chia tách, các ứng dụng microservice bao gồm nhiều thành phần độc lập output ra các API.

|  | **Microservices Architecture** | **Monolithic Architecture** |
| --- | --- | --- |
| **Tổng quát** | Microservice chia rất nhỏ, dịch vụ độc lập kết hợp lỏng lẻo và có tính năng phát triển liên tục nhanh chóng. | Gồm các đơn vị phụ thuộc lẫn nhau, không thể phân chia và có tốc độ phát triển rất thấp |
| **Ưu điểm** | - Một microservice có thể được phát triển bởi một team nhỏ. Do vậy việc quản lý sẽ dễ dàng hơn.  - Các microservice khởi động nhanh giúp quá trình phát triển, kiểm thử cũng nhanh hơn.  - Dễ dàng mở rộng và tích hợp với các dịch vụ của bên thứ ba.  - Cô lập lỗi tốt hơn, khi một microservice bị lỗi và ngừng hoạt động thì các microservice khác vẫn có thể hoạt động bình thường.  - Khi cần thay đổi một thành phần, thì chỉ cần sửa đổi, cập nhật và triển khai lại thành phần đó chứ không cần triển khai lại toàn bộ hệ thống. | - Dễ phát triển, yêu cầu cơ sở hạ tầng đơn giản, nhân lực nhỏ và có thể chia sẻ ít nhiều về kỹ năng.  - Việc phát triển và triển khai ứng dụng với kiến trúc này khá đơn giản khi mà các IDE hỗ trợ rất tốt việc kiểm tra và chạy ứng dụng với chỉ một cú click chuột hay một phím tắt.  - Kiến trúc nguyên khối có thể mở rộng bằng cách thêm một load balancer (thiết bị cân bằng tải) và triển khai thêm các khối ứng dụng nữa nếu cần. |
| **Nhược điểm** | - Việc triển khai hệ thống microservice phức tạp hơn nhiều so với việc triển khai hệ thống nguyên khối.  - Các lập trình viên phải tốn nhiều công sức hơn để thực hiện phần giao tiếp giữa các microservice.  - Cần tính toán kích cỡ của một microservice. Nếu một microservice quá lớn, bản thân nó trở thành một ứng dụng theo kiến trúc nguyên khối. Nếu một microservice quá nhỏ thì độ phức tạp của hệ thống tăng lên rất nhiều. | - Với mô hình nguyên khối, một lỗi nhỏ có thể làm cả hệ thống ngừng hoạt động.  - Khi ứng dụng nguyên khối ngày một lớn thì quá trình maintain (duy trì) và sửa lỗi ứng dụng cũng trở thành ác mộng vì thời gian build và khởi động lại ứng dụng rất lớn.  - Áp dụng công nghệ mới khó khăn vì toàn bộ ứng dụng phải thay đổi. Do đó nhiều ứng dụng một khối thường phụ thuộc một công nghệ cũ và lỗi thời. |

### Câu 5: Tìm hiểu về Spring Boot

* Để phát triển một ứng dụng web cơ bản HelloWorld sử dụng Spring framework bạn sẽ cần ít nhất 5 công đoạn sau:
  + Tạo một project sử dụng Maven với các dependency cần thiết của Spring MVC và Servlet API.
  + Một tập tin web.xml để khai báo DispatcherServlet của Spring MVC.
  + Một tập tin cấu hình của Spring MVC.
  + Một class Controller trả về một trang “Hello World” khi có request đến.
  + Cuối cùng là phải có một web server dùng để triển khai ứng dụng lên chạy.

**Spring Boot** là một dự án phát triển bởi **JAV** (ngôn ngữ java) trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng.

* Với Spring Boot chúng ta có thể triển khai dự án Spring một cách nhanh chóng và cấu hình đơn giản với các tính năng nổi bật:
  + Tạo các ứng dụng Spring độc lập
  + Nhúng trực tiếp Tomcat, Jetty hoặc Undertow (không cần phải deploy ra file WAR)
  + Các starter dependency giúp việc cấu hình Maven đơn giản hơn
  + Tự động cấu hình Spring khi cần thiết
  + Không sinh code cấu hình và không yêu cầu phải cấu hình bằng XML …