Interview Questions and Suggested Answers for Raju Ileni

Q: Can you explain the differences between Spring Boot and traditional Spring applications?

A: Spring Boot simplifies the setup of Spring applications by eliminating boilerplate configuration. Unlike traditional Spring, which requires manual setup of dependencies and XML-based configuration, Spring Boot provides embedded servers, auto-configuration, and starter dependencies, which accelerate development.

Q: How have you used Spring Batch in your recent projects?

A: I developed a standalone project using Spring Batch to extract data from SHA-256 encrypted files and load it into a master table. I handled job configuration, step processing, and error handling to ensure reliable and secure batch data processing.

Q: Describe how you handle exception management in a Java backend service.

A: I typically use @ControllerAdvice and @ExceptionHandler in Spring Boot to globally handle exceptions. This allows me to log errors consistently and return meaningful error messages to the client, improving both debugging and user experience.

Q: What are the key differences between JPA and Hibernate?

A: JPA is a specification, and Hibernate is an implementation of that specification. JPA provides a standard interface for ORM, while Hibernate offers additional features like caching and custom SQL support. I use JPA for portability and Hibernate when I need advanced ORM capabilities.

Q: What design patterns have you used in your projects and why?

A: Ive used Singleton for shared services, Factory for object creation, and Strategy for interchangeable algorithms, particularly in batch processing. These patterns help improve code maintainability and scalability.

Q: How have you implemented microservices in your projects?

A: In the Citi Bank and Organic 42 projects, I developed modular microservices using Spring Boot. Each service had its own database and handled a specific business function. Services communicated via REST APIs.

Q: How do you manage communication between microservices?

A: Primarily through RESTful APIs. We used standard HTTP methods and JSON for communication. In some projects, we used a service registry and config server setup for better orchestration, though not mentioned in this resume.

Q: Explain how you tested your REST APIs.

A: I used Postman and Swagger to test API functionality and ensure contract adherence. I also wrote JUnit and Mockito test cases to validate logic and simulate dependencies.

Q: How did you decrypt and process SHA-256 encrypted file data securely?

A: SHA-256 is a hashing algorithm, not reversible encryption. The system I developed likely involved verifying file integrity with hashes or processing already decrypted files. I used Spring Batch to process the data, ensuring secure access and proper exception handling.

Q: How do you handle authentication and authorization?

A: I use Spring Security for securing endpoints. For authentication, I configure token-based security (like JWT), and for authorization, I use role-based access controls with annotations like @PreAuthorize.