TECHNICAL INTERVIEW QUESTIONS

Backend Developer.

Welcome to ABC LTD, a Fintech company which provides banking solutions such as card management (virtual or real), accounts, payments, transfers and expense management to other financial institutions.

Scenario: You are tasked with developing a simple Spring Boot application that manages a collection of books. Each book has an ID, title, author, and publication year. You need to create a RESTful API to perform CRUD (Create, Read, Update, Delete) operations on this collection of books. The data should be stored in an H2 in-memory database.

Requirements:

1. **Project Setup:**
   - Create a new Spring Boot project using Spring Initializr with the following dependencies:
     - Spring Web
     - Spring Data JPA
     - H2 Database
2. **Entity Class:**
   - Create an `Book` entity with the following fields:
     - `id` (Long, primary key)
     - `title` (String)
     - `author` (String)
     - `year` (int)
3. **Repository Interface:**
   - Create a `BookRepository` interface that extends `JpaRepository<Book, Long>`.
4. **Service Layer:**
   - Create a `BookService` class that provides methods for CRUD operations.
5. **Controller Layer:**

    - Create a `BookController` class with the following REST endpoints:
        - `POST /books` to create a new book
        - `GET /books` to get a list of all books
        - `GET /books/{id}` to get details of a specific book by ID
        - `PUT /books/{id}` to update details of a specific book by ID
        - `DELETE /books/{id}` to delete a specific book by ID
6. **Database Configuration:**
    - Configure the application to use the H2 in-memory database.
7. **Security Configuration**
    - Configure the application to use basic Authentication

8. **Data Validation:**
    - Add basic validation to the `Book` entity (e.g., title and author should not be empty).
9. **Testing:**
    - Write unit tests for the service layer and integration tests for the controller layer.
10. **Swagger Document:**
    - Add swagger documentation to your project.

Notes from the exercise:

– Come up with a High-Level Design and Low-Level Designs.

– Create a new application using maven and SOLID principles to solve above problem

– Use any fake data that you want, there must be at least two different account ids and two different client ids.

– You can mock data

– Use Java version 8+

– Keep everything simple. Don't implement or use any third party library without an explanation

– Explain everything you think it's important to take into consideration in a file named Readme.md

Extras Marks:

Run the code inside a container

Use a DB, preferably mongoDB, MariaDB, H2

Implement Unit Tests

Implement Code Coverage

Swagger Definitions