Java Backend Engineer Assignment

Use Case: Library Book Reservation System

Objective

You are tasked with designing and implementing a **Library Book Reservation Service** for a modern digital library system. The service should allow users to reserve books, manage their reservations, and ensure proper business rules are followed.

Requirements

Reservation Service API - Develop a RESTful API that allows users to:

- Reserve a book.
- Retrieve an existing reservation by ID.
- Retrieve all reservations for a specific user.
- Cancel a reservation.

Constraints**:**

- A user cannot reserve more than 3 books at a time.
- A book can only be reserved if it is available.

Database Schema - Design a relational database schema to store reservation details.

- **Reservations** should keep track of:
  - The user making the reservation.
  - The book being reserved.
  - When the reservation was created.
  - The current status of the reservation (e.g., active, canceled, expired).
- **Books** should keep track of:
  - The book’s title, author, and ISBN.
  - The number of copies available for reservation.

Business Logic

- Prevent a user from exceeding the max reservation limit.
- Update the book’s available copies when a reservation is made or canceled.

Tech Stack

- Java (Spring Boot)
- ORM or no ORM, your choice
- Relation database of your choice

Bonus

Logical enhancements

- Reservations expire after 7 days if not picked up
- User can see history of reservations incl. expired ones

Technical enhancements

- Implement caching using Redis to optimize frequently accessed data.
- Add unit tests for service and repository layers.
- Dockerize the application.
- Implement a scheduled job to expire reservations after 7 days.

Evaluation Criteria

- **Code Quality**: Readability, structure, and maintainability
- **Database Design**: Proper schema design and indexing.
- **Microservices Best Practices**: Clean architecture and separation of concerns.
- **API Design**: RESTful principles, error handling, and response structure.
- **Performance Considerations**: Efficient queries and data handling.
- **Testing**: Presence of unit tests.

Submission Guidelines

- After the task is completed, we kindly ask you to provide the project in a ZIP file to the Talent Acquisition team via email attachment. Be sure to only include your work and not related libraries or compiled binaries, which will prevent the attachment from being successfully transmitted.