Level 3 - Code Challenge

You are expected to tackle two questions in preparation for this interview. For the first question, you are required to **write a complete program and save it on GitHub**. The completed program code will be shared on the day of the interview for review.

As for the second question, a written solution is not necessary. **Prepare to discuss any points that need clarification and possible approaches to solving the problem**.

Question 1 - Project Simple CRUD API

The goal of this project is to create a basic CRUD API for managing a collection of items using PostgreSQL as the database. Each item will have a unique identifier, a name, and a description.

Technologies

- Java
- Spring Boot
- Spring Data JPA (for data access)
- PostgreSQL (as the database)

Step-by-step guide:

Project Setup

- Create a new Spring Boot project using your preferred IDE or Spring Initializr (https://start.spring.io/).
- Add the necessary dependencies: Spring Web, Spring Data JPA, and PostgreSQL driver.

Configure Database Connection:

Open the application.properties (or application.yml) file and update the database connection details with your PostgreSQL configuration:

```
spring.datasource.url=jdbc:postgresql://localhost:5432/your_database_name
spring.datasource.username=your_username
spring.datasource.password=your_password
```

Data Model

Create a model class to represent your items. For example:

```
@Entity
public class Item {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;
    private String description;

    // Constructors, getters, setters, etc.
}
```

Repository

Create a repository interface that extends JpaRepository<Item, Long>. This interface will provide basic CRUD operations for your Item entity without requiring you to write the implementation.

```
public interface ItemRepository extends JpaRepository<Item, Long> {
}
```

Service Layer (Optional but recommended)

Create a service class to encapsulate business logic if needed. For this simple project, you might skip this step.

Controller

Create a controller class to handle HTTP requests and interact with the repository. The controller will define methods for creating, reading, updating, and deleting items.

```
@RestController
@RequestMapping("/api/items")
public class ItemController {
    @Autowired
    private ItemRepository itemRepository;

    // Endpoint for getting all items
    @GetMapping
    public List<Item> getAllItems() {
        // your code goes here
    }

    // Endpoint for getting a single item by its ID
    @GetMapping("/{id}")
    public ResponseEntity<Item> getItemById(@PathVariable Long id) {
        // your code goes here
    }

    // Endpoint for creating a new item
    @PostMapping
    public Item createItem(@RequestBody Item item) {
        // your code goes here
    }

    // Endpoint for updating an existing item
    @PutMapping("/{id}")
    public ResponseEntity<Item> updateItem(@PathVariable Long id, @RequestBody Item updatedItem) {
        // your code goes here
    }

    // Endpoint for deleting an item by its ID
    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteItem(@PathVariable Long id) {
        // your code goes here
    }
}
```

Run the Application

Run your Spring Boot application and test the API using tools like Postman or curl.
During the interview, be ready to explain your thought process behind each step and how you handle error cases and validations in the API. Best of luck!

Bonus

Create a user interface using a framework of your choice!

Question 2 - Flat Game Board with Rolling Ball

You are required to design a program that finds the shortest path from point A to point B on a flat game board. The board will have walls placed at runtime. The ball on the board can only roll orthogonally, and once it starts moving, it cannot stop until it encounters a wall. Your task is to implement an algorithm that efficiently calculates and displays the shortest path between points A and B under these given constraints.