Project Overview: Image and Slideshow Management System

This project aims to demonstrate expertise in Java Spring Boot technologies and modern development practices. The application allows users to manage a list of image URLs, add new images, create slideshows, and register events related to image transitions within slideshows. The application also handles error management and provides an efficient database structure using JPA and Spring Boot.

# Technical Overview

The system was built using Spring Boot with an emphasis on good software practices. The database was managed using SQL Server, and the application was developed using JDK 8. The system integrates several APIs to handle images and slideshows, providing the functionality required in the task.

Additionally, the system utilizes Spring Boot for building REST APIs and managing image and slideshow data using JPA for persistent data storage. Error handling is implemented using try-catch blocks to ensure the system's stability, with commit and rollback operations being applied in case of failures.

# Functional Requirements

The system meets the following core functional requirements:

1. APIs for adding, deleting, searching, and managing images and slideshows.

2. Proof-of-play events when images are transitioned in slideshows.

3. Data storage using MySQL or PostgreSQL for persistent storage.

4. Modern Java techniques such as error handling and unit testing using JUnit 5.

# Advanced Features

The application includes the following advanced features:

1. Containerization using Docker and docker-compose.yml for local environment setup.

2. Event-Driven Architecture with Spring EventPublisher for event logging related to image and slideshow management.

3. Non-blocking reactive programming techniques implemented with Spring WebFlux for handling requests efficiently.

# Testing

Unit tests were written using JUnit 5 to ensure the correctness of the application. These tests verify that the core functionalities, such as image addition, deletion, and slideshow management, work correctly.

# Code Submission and Quality

The project was submitted to a public GitHub repository, following best coding practices and ensuring a clean code structure. Meaningful comments were included throughout the code to enhance readability and maintainability. The code follows modern Java and Spring principles, adhering to high standards of software quality.

# Docker & JDK Compatibility

A Dockerfile was created to containerize the application, with a local environment configured via docker-compose.yml. The system is compatible with JDK 8, although it can be adapted to work with newer JDK versions by modifying the configuration if needed. Currently, the system is running with the older JDK for stability reasons.

# Conclusion

This project provides a complete solution for managing images and slideshows, implementing modern Java development practices and advanced features such as containerization, event-driven architecture, and non-blocking reactive programming. The code is clean, well-documented, and adheres to best practices in the industry.