**Deployment and Execution Guide for Performance Review Application**

**Overview:**

This guide provides clear and simple steps to deploy and execute the Performance Review Application. Follow these steps to ensure a smooth deployment and proper functioning of the application.

**Prerequisites:**

1. Java Development Kit (JDK):

Install JDK 11 or later.

Set the `JAVA\_HOME` environment variable.

2. Database:

Install MySQL or PostgreSQL (as per your application configuration).

Create the required database for the application.

Update the database connection details (username, password, URL) in the application configuration file.

3. Maven:

Install Apache Maven for dependency management and build . you can use SpringBoot for easy development and debug .

4. Application Configuration:

- Ensure the **application.properties** file contains the correct configurations, including:

->Database connection.

->Port number. (by default port number is 8080 in this application )

->Other environment-specific settings.

**Steps to Deploy:**

1. Clone the Repository

Clone the application repository from your version control system (e.g., GitHub, GitLab):

git clone <repository>

**use the repository name : https://github.com/WHITEDEVILTWO/EmployeeAnnualRatingAPI.git**

2. Build the Application

Use Maven to clean, compile, and package the application:

**mvn clean install**

This will generate a `.jar` file in the `target` directory.

3. Set Up the Database

1. Run the provided SQL scripts to set up the required tables and initial data:

- For example, use the SQL file to create the SelfReview, ManagerReview, and PerformanceSummary tables.

**CREATE TABLE SelfReview (**

**id BIGINT PRIMARY KEY AUTO\_INCREMENT,**

**employee\_id BIGINT NOT NULL,**

**review\_text TEXT NOT NULL,**

**timestamp DATETIME NOT NULL**

**);**

**CREATE TABLE ManagerReview (**

**id BIGINT PRIMARY KEY AUTO\_INCREMENT,**

**employee\_id BIGINT NOT NULL,**

**review\_text TEXT NOT NULL,**

**rating DECIMAL(3,2) NOT NULL,**

**timestamp DATETIME NOT NULL**

**);**

**CREATE TABLE PerformanceSummary (**

**id BIGINT PRIMARY KEY AUTO\_INCREMENT,**

**employee\_id BIGINT NOT NULL,**

**self\_review\_text TEXT NOT NULL,**

**self\_review\_rating DECIMAL(3,2) NOT NULL,**

**self\_review\_timestamp DATETIME NOT NULL,**

**manager\_review\_text TEXT NOT NULL**,

**manager\_rating DECIMAL(3,2) NOT NULL,**

**performance\_score DECIMAL(3,2) NOT NULL,**

**manager\_review\_timestamp DATETIME NOT NULL**

**);**

2. Verify the database connection by testing with a sample query.

3. Run the Application

4. Execute the `.jar` file with the following command:

**java -jar target/<application-name>.jar**

Replace `<application-name>` with the `.jar` file.

**JAR file : EmployeeRating-0.0.1-SNAPSHOT.jar**

**Testing the Application:**

**Test with Postman :**

- Import the provided Postman collection.

Use api-docs end point and copy the JSON format out from the end point , then got to post man

Go to file->import->select raw test-> then enter .

It will filter the all available API end points and gives us an clear view to use them in the post man .

- Use the configured requests to test API endpoints such as:

**- Add Self Review (`POST /api/self`)**

**- Add Manager Review (`POST /api/manager`)**

**- Get Performance Summary (`GET /api/{employeeId}`)**

**Note : Please visit the github repository for API end points results from postman .**

**GitHub repository Link :**

**Actuator and API-Docs Integration :**

1. Actuator for Health Monitoring:

- Use Actuator endpoints to monitor application health.

- Example: `http://localhost:<port>/actuator/health` to check application health status.

2. API-Docs for Endpoint Details:

- Access JSON API documentation at: `http://localhost:<port>/v3/api-docs`.

- Copy the JSON text and import it into Postman to filter and test API endpoints.

**Troubleshooting:**

1. Database Connection Issues:

- Verify the database is running and accessible.

- Check the connection details in the configuration file.

2. Application Fails to Start:

- Review the logs for specific error messages.

- Ensure all required dependencies are included in the build.

3. Endpoint Errors:

- Validate the input data for correctness.

- Ensure database migrations have been applied.

**Summary:**

By following this guide, you can deploy and execute the Performance Review Application successfully.