# Task Manager API Documentation

## Instructions to Run the Application Locally

### Prerequisites:

1. Java 17: Ensure Java 17 is installed.  
 To check: java -version  
 Download Java 17 if not installed: https://www.oracle.com/java/technologies/javase-jdk17-downloads.html

2. Maven: Make sure Maven is installed and available in your system's PATH.  
 To check: mvn -version

3. PostgreSQL: Set up PostgreSQL locally.  
 Database: taskdb  
 Username: user  
 Password: password

4. Docker: Install Docker and Docker Compose if you plan to run the application via containers.  
 Install Docker: https://docs.docker.com/get-docker/  
 Install Docker Compose: https://docs.docker.com/compose/install/

## Running the Application

1. Clone the Repository:  
 git clone https://github.com/GH-ShivamTiwari/GKAssignment.git  
 cd GKAssignment

2. Configure the Database:  
 Update the src/main/resources/application.properties file with your PostgreSQL configurations:  
 spring.datasource.url=jdbc:postgresql://localhost:5432/taskdb  
 spring.datasource.username=user  
 spring.datasource.password=password

3. Build and Run:  
 mvn clean install  
 mvn spring-boot:run

4. Docker Compose:  
 If you prefer running with Docker:  
 docker-compose up --build

5. Access the Application:  
 The application will be running on http://localhost:8080.

## Write-Up and Assumptions

### Approach:

1. User Authentication and Registration:  
 Implemented with Spring Security and JWT for stateless authentication.  
 Passwords are securely hashed using BCryptPasswordEncoder.

2. Task Management:  
 CRUD operations for tasks allow users to create, read, update, and delete their own tasks.  
 Tasks can be filtered by status, priority, and due date, and can be searched by title or description.

3. Database:  
 PostgreSQL is used as the relational database.  
 Spring Data JPA is used to interact with the database.

4. Dockerization:  
 A Dockerfile is provided to containerize the application.  
 docker-compose.yml sets up both the application and a PostgreSQL database.

5. Security:  
 JWT tokens are used to secure all API endpoints except for user registration and login.  
 Role-based access control is implemented for additional points.

6. Bonus Features:  
 Pagination is available for task retrieval to optimize the API for large datasets.  
 Unit and integration tests are set up for key API endpoints.  
 A CI/CD pipeline can be set up using GitHub Actions for continuous integration and delivery.

## API Documentation

Postman Collection: https://www.postman.com/crimson-moon-634689/publicws/collection/a02vh2a/task-manager-api?action=share&creator=19085251  
Click the link to view the collection in Postman. From there, you can fork it to your workspace or download it to use locally.

Instructions on how to use the collection:  
1. Click the link to open the collection in Postman.  
2. Fork it into your Postman workspace.  
3. Use the pre-configured requests to interact with the API endpoints.  
Make sure to provide a valid JWT token in the authorization header for protected routes (tasks-related requests).