Maintenance Guide

Introduction

The purpose of this guide is to inform team members of the steps needed to update and maintain the application for error resolution and system improvement. Below is a list of system requirements and actions required to redeploy the updated version of the application to AWS.

Requirements

- Java Development Kit (JDK) 11 or newer
- Apache Maven
- Any Java IDE

Project Setup

1. Create a new project using your IDE from version control (VCS).
2. Supply the IDE with the GitLab repository link.
3. After the project is successfully cloned, navigate to the root directory in the terminal and run '**mvn clean install**".
4. Next, run '**mvn spring-boot:run**'. This will enable you to access the application vial **localhost:8080**.

Running Tests

1. The application uses JUnit for testing. To run all tests, navigate to the project's root directory and, in the terminal, run '**mvn test**'.
2. Alternatively, individual unit tests can be performed by navigating to the **test** folder of the project, right-click the unit test you wish to perform, and click '**Run'.**

Updating the Application

1. The application uses Docker for image/container management.
2. After completing all necessary changes, navigate to the project's root directory and run '**mvn package'** in the terminal to generate a new .jar file.
3. Ensure that the new name of the .jar matches that which is named in the Dockefile.
4. Next, build a new Docker image by running '**docker build -t perfectpens:latest**'.
5. Log in to the ECR repository using the AWS CLI: '**aws ecr get-login-password --region us-east-1 | docker login --username AWS –'**
6. After the new image is built, push the updated image to AWS ECR '**docker push public.ecr.aws/p2i9t3d5/perfectpens:latest**'.
7. Navigate to Elastic Beanstalk from AWS console.
8. Locate the perfectpens environment and click '**Upload and Deploy**'.
9. Reupload the Dockerrun.aws.json file, and Elastic Beanstalk will handle the rest.