Scenario 1 – Containerize and existing legacy application

In this exercise, you will take on the role of a DevOps engineer tasked with dockerizing a legacy application. The application has been running on a dedicated server for five years, and the goal is to transition it into a Docker container to leverage the benefits of containerization, such as scalability, consistency, and isolation.

Requirements

A legacy todo application is available on GitHub at https://github.com/tdevsin/docker-mastery. This application has been running for five years and now needs to be dockerized. You need to write a Dockerfile and ensure the application runs properly locally.

To successfully complete this exercise, you need to:

- Create a Dockerfile that encapsulates all the dependencies and runtime environment needed for the application.
- Ensure the application runs correctly inside the container without any changes to the legacy code.
- Implement network settings that allow the containerized application to communicate with external systems as it did in its non-containerized state.

Resources

- Docker Official Documentation
- Dockerfile Reference
- Docker Networking Overview
- Project GitHub repository: https://github.com/tdevsin/docker-mastery.git

Acceptance Criteria

1. **Dockerfile Created**: A Dockerfile must be created that successfully builds an image capable of running the application.
2. **Application Runs**: The application must run without errors inside the Docker container.

3. **Network Connectivity**: The container must maintain its required network connections with external systems, such as a database.

Hints

- Clone the application locally and set up a new MySQL database. Make sure you create a new schema as mentioned in the project's README.
- Carefully read the project's README and follow all the steps to run the application locally first.
- Once the application is up and running, you'll have a better understanding of what needs to be done.
- Write a Dockerfile at the root of the project and add the necessary code to it.
- Run the Docker container locally and test the application.