Introduction to DEVOPS (Merged- CSIZG514/SEZG514) (S1-25)

Assignment- 1

Objective

This assignment aims to provide students with hands-on experience in fundamental DevOps practices. By completing this assignment, students will gain proficiency in version control with Git and GitHub, containerization using Docker, and implementing continuous integration and continuous delivery (CI/CD) pipelines with GitHub Actions.

Problem Statement

You are tasked with developing a robust and automated application for **ACEest_Fitness and Gym**, a burgeoning startup. As a Junior DevOps Engineer, your primary responsibility is to establish a streamlined development and deployment workflow that ensures code quality, consistency, and efficient delivery. Your solution should encompass the following key phases:

1. **Application Development:** Create a foundational **Flask web application** that demonstrates core functionalities pertinent to a fitness and gym management system. A basic Python Flask file will be provided to initiate this development phase.

2. **Version Control System (VCS) Implementation:** Initialize a **Git repository** for the project and establish its remote counterpart on **GitHub**. All development iterations, including new features, bug fixes, and infrastructure changes, must be meticulously tracked and managed using Git.

3. **Unit Testing Framework Integration:** Develop comprehensive **unit test cases** using the **Pytest framework** to validate the integrity and expected behavior of the Flask application's core functionalities. These tests are crucial for ensuring the reliability of the application's components.

4. **Automated Testing Configuration:** Configure the project to automatically execute all developed **Pytest unit tests**. This automation ensures that code changes are thoroughly validated before proceeding to subsequent stages of the pipeline.

5. **Containerization with Docker:** Package the Flask web application and all its dependencies into a lightweight and portable **Docker container image**. This step ensures environmental consistency across development, testing, and deployment environments.

6. **CI/CD Pipeline with GitHub Actions:** Design and implement a fully automated **Continuous Integration/Continuous Delivery (CI/CD) pipeline** using **GitHub Actions**. This pipeline must trigger automatically upon every code push to the GitHub repository, performing the following critical operations:

   Automated Build: Successfully build the Docker image for the application.

Automated Testing: Execute the Pytest unit tests within the built Docker image to confirm application functionality and stability.

Deliverables

- A publicly accessible GitHub repository containing:
  - The Flask web application files (.py, requirements.txt, etc.).
  - Pytest unit test files.
  - Dockerfile for containerization.
  - GitHub Actions workflow file (.github/workflows/main.yml or similar).
- A clear README.md file in the GitHub repository explaining:
  - How to set up and run the application locally (if applicable).
  - How to execute the tests locally.
  - A brief overview of the GitHub Actions pipeline.

Submission Guidelines

- Ensure your GitHub repository is public and accessible for review.
- Submit the URL of your GitHub repository by the specified deadline.
- The GitHub Actions workflow should demonstrate successful runs for recent commits.

Evaluation Criteria

Your assignment will be evaluated based on the following:

- **Completeness of Application:** Functionality of the Flask application as per the problem statement.
- **Git & GitHub Usage:** Proper use of Git for version control (meaningful commits, branch management if applicable).
- **Test Case Quality:** Coverage and effectiveness of Pytest unit tests.
- **Docker Containerization:** Correctness and efficiency of the Dockerfile.
- **GitHub Actions CI/CD Pipeline:**
  - Successful execution of automated build and test stages on push.
  - Correct configuration of the workflow.
  - Reliability and robustness of the pipeline.
- **Documentation:** Clarity and thoroughness of the README.md file.