1. Introduction

This project focuses on implementing a complete CI/CD pipeline using GitHub Actions and Docker. The goal is to automate the testing, building, and deployment of a web application inside a Docker container.

2. Abstract

The project automates the software delivery process through GitHub Actions, ensuring code pushed to the repository is automatically tested, containerized, and deployed using Docker. It provides a practical understanding of DevOps tools and real-world deployment practices.

3. Tools Used

- Git & GitHub
- GitHub Actions (CI/CD)
- Docker & Docker Hub
- HTML/CSS (Frontend)
- Linux (Ubuntu)
- EC2 Instance (Deployment server)

4. Steps Involved in Building the Project

1. Created a simple personal portfolio webpage using HTML & CSS
2. Added a `Dockerfile` to containerize the app
3. Wrote a `docker-compose.yml` to run the container
4. Created a GitHub Actions workflow (`.github/workflows/ci.yml`) to:

    - Run test script
    - Build Docker image
    - Push it to Docker Hub

5. Hosted the containerized app on EC2 and exposed via port 8080
6. Secrets like `DOCKER_USERNAME` and `DOCKER_PASSWORD` were added to GitHub for secure Docker Hub login

5. Conclusion

Through this project, I understood the full lifecycle of a CI/CD pipeline. I learned how GitHub Actions integrate with Docker and how automated workflows can be built for real-world deployments. This experience has strengthened my DevOps fundamentals and made me more confident in applying these practices in production environments.