# Project 1 – CI/CD Demo Project – Documentation

This project demonstrates a complete CI/CD (Continuous Integration and Continuous Deployment) pipeline using GitHub Actions and Docker. The application is a simple Node.js based web application containerized and deployed using Docker. The project automates the build, test, and deployment process to ensure continuous delivery with minimal manual effort.

**Tools & Technologies Used: -**

GitHub Actions: For automation of CI/CD pipeline.

Docker: To containerize and run the Node.js application.

Node.js: Backend runtime environment for the sample app.

Git & GitHub: Version control and code repository.

**Workflow Explanation:-**

1. Code is pushed to the GitHub repository.
2. GitHub Actions workflow (ci.yml) is triggered automatically.
3. The workflow logs into Docker Hub using username and access token.
4. A Docker image is built from the source code and pushed to Docker Hub.
5. The Docker container is run locally or on a server using the pulled image.

**Challenges Faced & Fixes:-**

**Docker Hub Authentication Error:** Resolved by creating a Docker Hub access token and storing it as a GitHub secret.

**Port Mismatch Issue:** The app was running on port 3000 internally, but exposed on port 8080.

Fixed by mapping ports correctly (-p 3000:3000).

**Localhost Access Error:** Verified using docker logs to confirm the running port and updated the run command accordingly.

**Conclusion:-**

This project helped in understanding the automation of software delivery using GitHub Actions and Docker. It demonstrates how a developer can push code changes to GitHub and automatically trigger builds, tests, and deployments. It also provided hands on experience in resolving issues related to authentication, port mapping, and container management.