Final Project Report

Introduction

This project demonstrates a complete CI/CD pipeline using GitHub Actions and Docker. The objective was to automate testing, building, pushing of Docker images, and deploying the application on a self-hosted runner (local VM). This pipeline ensures faster delivery, consistency, and reliability in the software lifecycle.

Abstract

The CI/CD pipeline was built with GitHub Actions to automate the software delivery workflow. Each code commit triggers unit tests, Docker image build, and image push to Docker Hub. The deployment is handled by Docker Compose on a self-hosted runner, with health checks to validate successful deployment. This setup demonstrates end-to-end DevOps practices without requiring cloud infrastructure.

Tools Used

- GitHub Actions – CI/CD workflow automation
- Docker – Containerization
- Docker Hub – Image registry
- Docker Compose – Multi-container orchestration
- Node.js – Application runtime
- Self-hosted Runner (Windows/VM) – Local deployment

Steps Involved in Building the Project

1. Configured GitHub Actions workflow (ci.yml) for testing, building, and deployment.
2. Implemented unit testing for the Node.js application.
3. Built Docker images and tagged them with latest and commit SHA.
4. Pushed Docker images to Docker Hub repository.
5. Set up a self-hosted runner to deploy containers using Docker Compose.
6. Verified application health via HTTP endpoint after deployment.

Conclusion

The project successfully implemented a CI/CD pipeline with GitHub Actions and Docker. The workflow automated testing, image building, and deployment on a local VM. By completing this project, hands-on experience was gained in containerization, automation, and DevOps practices that are highly relevant in the IT industry. The pipeline can be extended further for cloud-based deployments in the future.