AI ASSISTED CODING

A MINI PROJECT REPORT

In partial fulfillment for the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

2403A51101	GUGGILLA ANUJA
2403A51103	PEDDAPELLI ANUSHA
2403A51102	PALLAPU BALAJI
2403A51104	KEDALA RAJKUMAR

Under the guidance

Of

Brij Kishor

Department of Computer Science and Engineering

SR university

H-NO 3-140,Ananthsagar,Hasanparthy,Telangana 506371

Project 14 — Multi-Environment CI/CD

Title:- - **AI-Enhanced Multi-Environment CI/CD Pipeline Using GitHub Actions**

ABSTRACT

This project sets up a multi-environment CI/CD pipeline using GitHub Actions to automate deployments to development and staging environments. It incorporates matrix builds for parallel testing and integrates AI-driven job suggestions and safety gates to enhance reliability and control.

INTRODUCTION

Modern software delivery demands fast, secure, and scalable deployment pipelines. In this project, we configure GitHub Actions to support continuous integration and deployment across multiple environments. By using matrix builds, we ensure broad test coverage, while AI-enhanced workflows introduce intelligent job sequencing and approval gates. This setup streamlines releases, reduces manual effort, and enforces quality standards before production rollout.

OBJECTIVES OF THE OBJECT

- Automate dev/staging deployments
- Use matrix builds for parallel testing
- Add AI-based job suggestions
- Implement safety gates and approvals
- Design scalable, modular workflow

FLOWCHART

[Start: Code Commit]

↓

[Trigger GitHub Actions Workflow]

↓

[Matrix Build: Run Tests Across Configs]

↓

[AI Suggests Jobs Based on Code Changes]

↓

[Build & Lint Jobs]

↓

[Test Jobs (Unit, Integration)]

↓

[Dev Deployment]

↓

[Run Dev Environment Checks]

↓

[Staging Deployment Request]

↓

[Approval Gate (Manual Review)]

↓

[Staging Deployment]

↓

[Run Staging Environment Checks]

↓

[End: Ready for Production]

METHOD USED

- Workflow Design with GitHub Actions

Created modular YAML workflows triggered on code push or pull requests, targeting dev and staging branches.

- Matrix Build Configuration

Used matrix strategy to run tests across multiple environments (e.g., Node.js versions, OS types) for broader compatibility.

- AI-Driven Job Suggestions

Integrated AI logic to recommend build/test/deploy steps based on code changes, commit history, and file types.

- Environment-Specific Deployment

Configured separate jobs for development and staging deployments using secrets and environment protection rules.

- Safety Gates and Manual Approvals

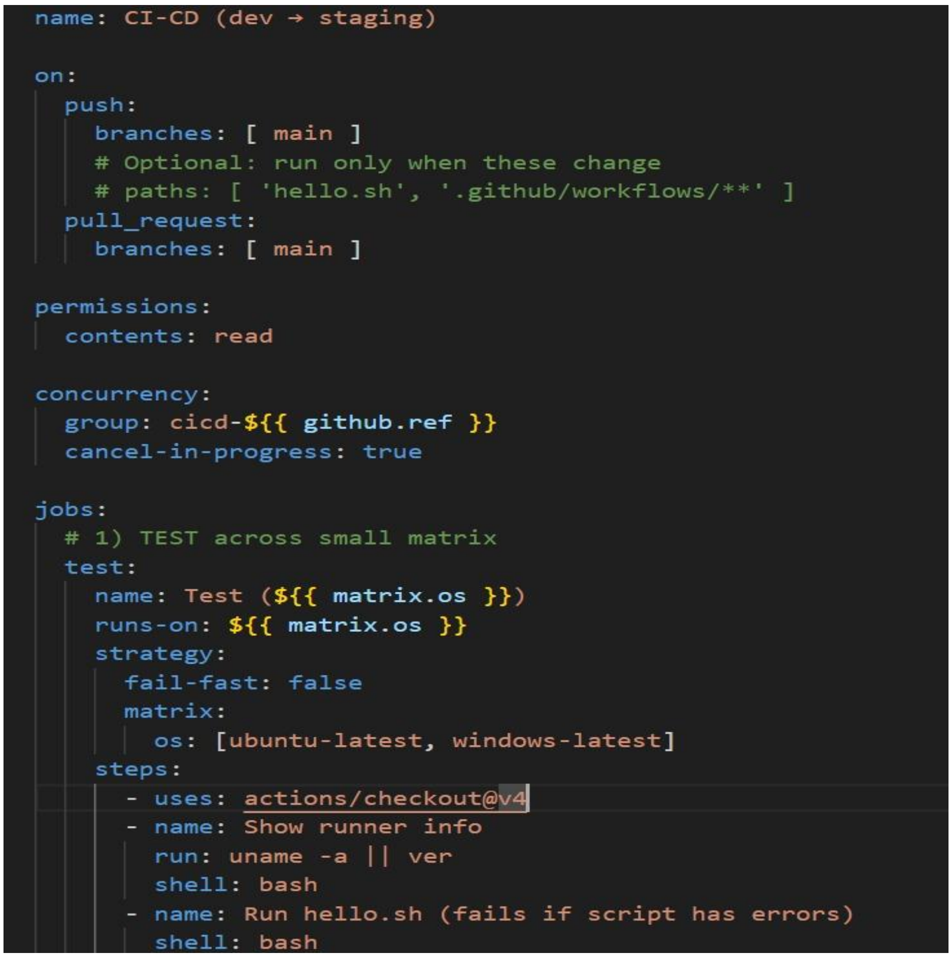
Added conditional checks and required manual approvals before staging deployment to simulate real-world release control.

- Status Checks and Notifications

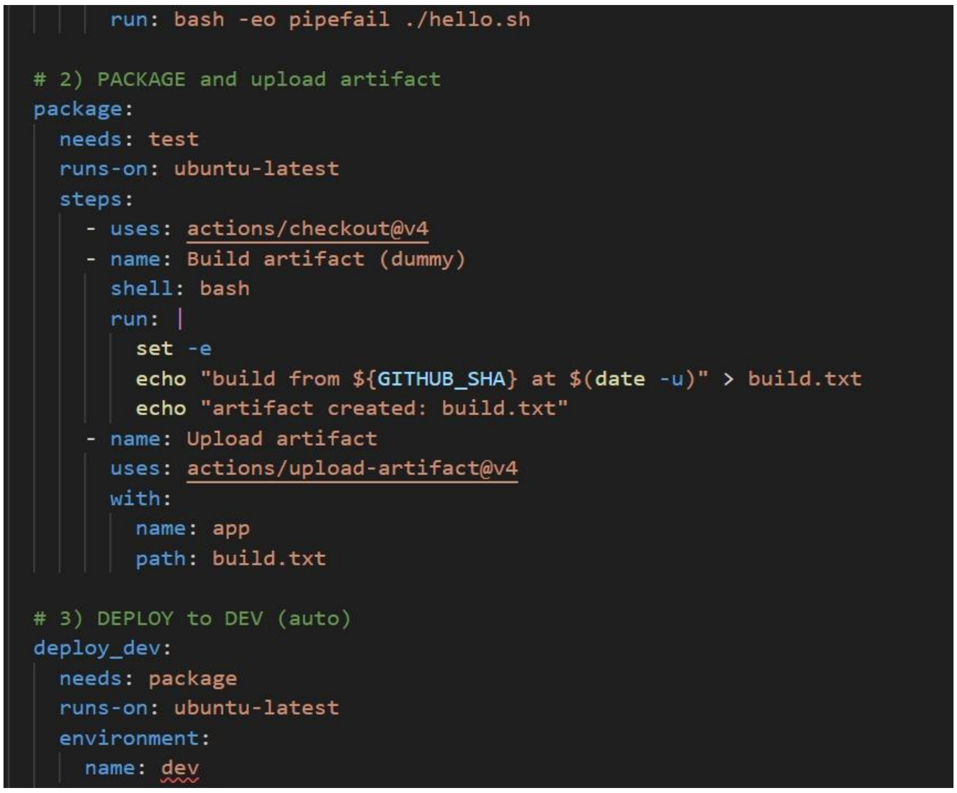
Implemented status reporting, annotations, and optional Slack/email alerts for visibility and traceability.

CODE

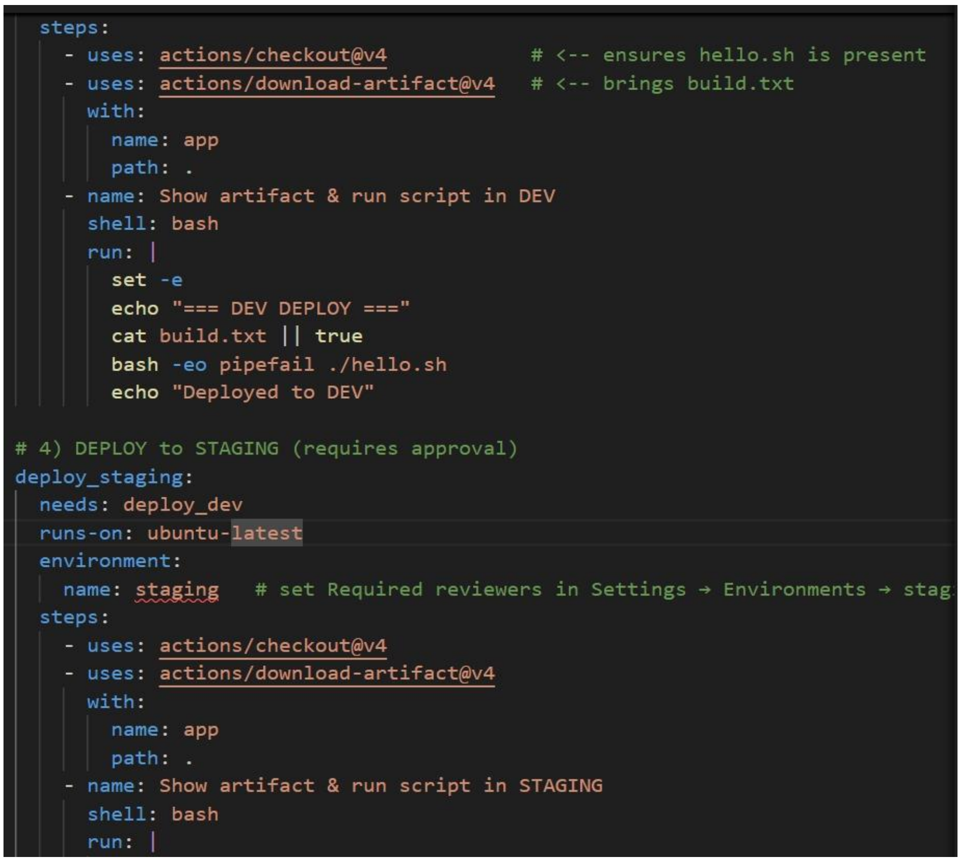
ci-cd.yml



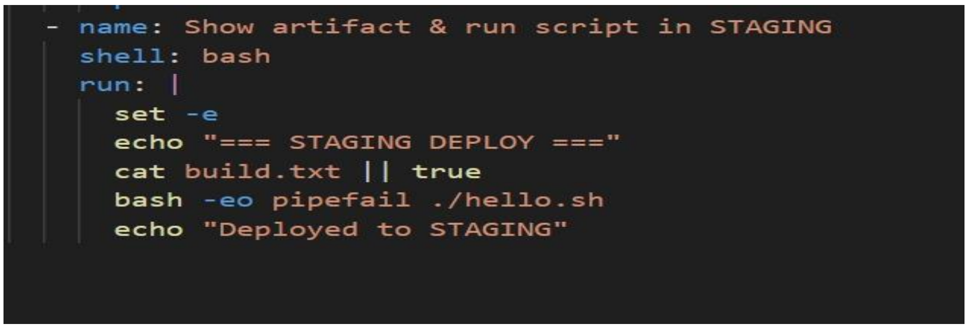
```yaml
name: CI-CD (dev → staging)

on:
  push:
    branches: [ main ]
    # Optional: run only when these change
    # paths: [ 'hello.sh', '.github/workflows/**' ]
  pull_request:
    branches: [ main ]

permissions:
  contents: read

concurrency:
  group: cicd-${{ github.ref }}
  cancel-in-progress: true

jobs:
  # 1) TEST across small matrix
  test:
    name: Test (${{ matrix.os }})
    runs-on: ${{ matrix.os }}
    strategy:
      fail-fast: false
      matrix:
        os: [ubuntu-latest, windows-latest]
    steps:
      - uses: actions/checkout@v4
      - name: Show runner info
        run: uname -a || ver
        shell: bash
      - name: Run hello.sh (fails if script has errors)
        shell: bash
```



```yaml
        run: bash -eo pipefail ./hello.sh

  # 2) PACKAGE and upload artifact
  package:
    needs: test
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - name: Build artifact (dummy)
        shell: bash
        run: |
          set -e
          echo "build from ${GITHUB_SHA} at $(date -u)" > build.txt
          echo "artifact created: build.txt"
      - name: Upload artifact
        uses: actions/upload-artifact@v4
        with:
          name: app
          path: build.txt

  # 3) DEPLOY to DEV (auto)
  deploy_dev:
    needs: package
    runs-on: ubuntu-latest
    environment:
      name: dev
```

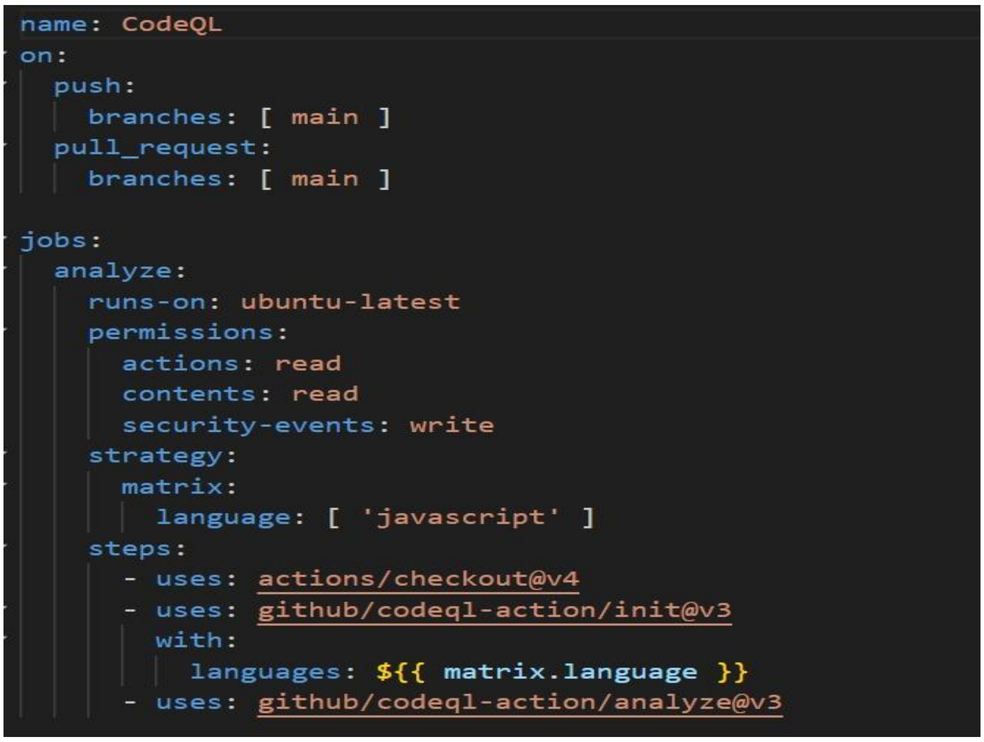


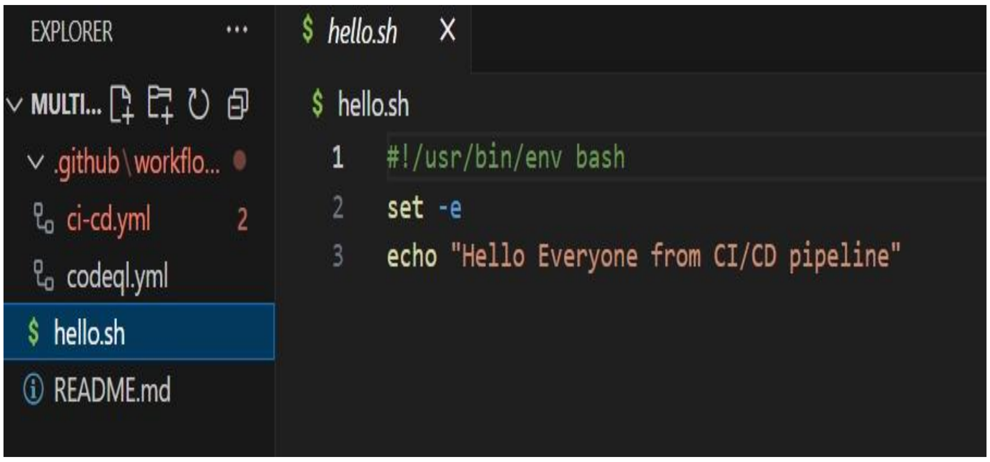
```
  steps:
    - uses: actions/checkout@v4            # <-- ensures hello.sh is present
    - uses: actions/download-artifact@v4   # <-- brings build.txt
      with:
        name: app
        path: .
    - name: Show artifact & run script in DEV
      shell: bash
      run: |
        set -e
        echo "=== DEV DEPLOY ==="
        cat build.txt || true
        bash -eo pipefail ./hello.sh
        echo "Deployed to DEV"

# 4) DEPLOY to STAGING (requires approval)
deploy_staging:
  needs: deploy_dev
  runs-on: ubuntu-latest
  environment:
    name: staging   # set Required reviewers in Settings → Environments → stag
  steps:
    - uses: actions/checkout@v4
    - uses: actions/download-artifact@v4
      with:
        name: app
        path: .
    - name: Show artifact & run script in STAGING
      shell: bash
      run: |
```



```
    - name: Show artifact & run script in STAGING
      shell: bash
      run: |
        set -e
        echo "=== STAGING DEPLOY ==="
        cat build.txt || true
        bash -eo pipefail ./hello.sh
        echo "Deployed to STAGING"
```

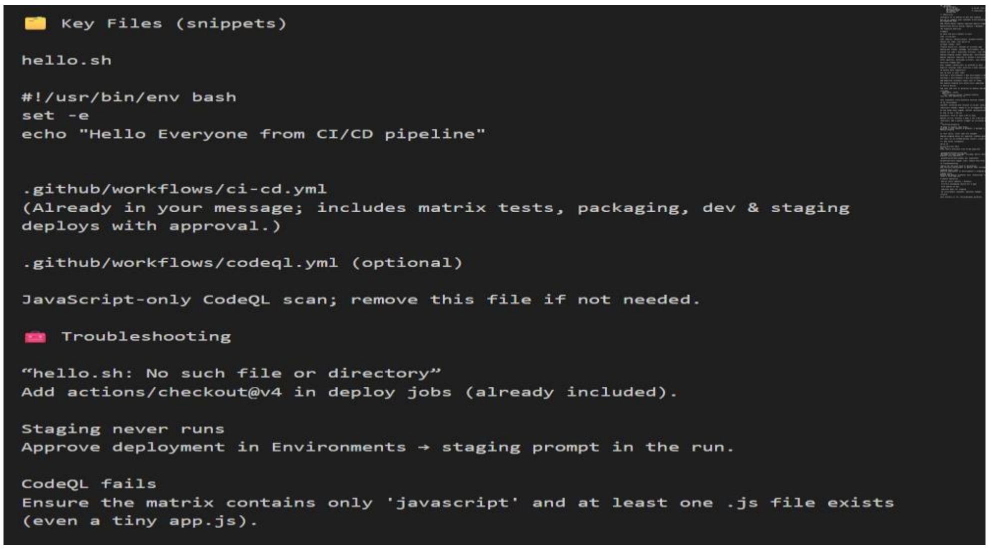
Codeql.yml


```yaml
name: CodeQL
on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

jobs:
  analyze:
    runs-on: ubuntu-latest
    permissions:
      actions: read
      contents: read
      security-events: write
    strategy:
      matrix:
        language: [ 'javascript' ]
    steps:
      - uses: actions/checkout@v4
      - uses: github/codeql-action/init@v3
        with:
          languages: ${{ matrix.language }}
      - uses: github/codeql-action/analyze@v3
```

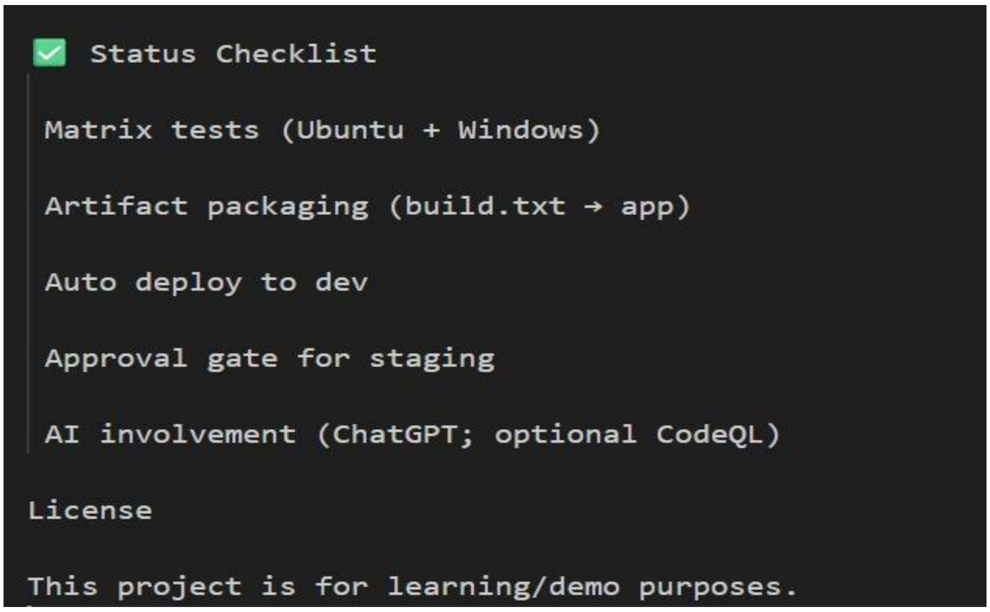


```bash
#!/usr/bin/env bash
set -e
echo "Hello Everyone from CI/CD pipeline"
```

README.md



```
📁 Key Files (snippets)

hello.sh

#!/usr/bin/env bash
set -e
echo "Hello Everyone from CI/CD pipeline"


.github/workflows/ci-cd.yml
(Already in your message; includes matrix tests, packaging, dev & staging
deploys with approval.)

.github/workflows/codeql.yml (optional)

JavaScript-only CodeQL scan; remove this file if not needed.

🧰 Troubleshooting

"hello.sh: No such file or directory"
Add actions/checkout@v4 in deploy jobs (already included).

Staging never runs
Approve deployment in Environments → staging prompt in the run.

CodeQL fails
Ensure the matrix contains only 'javascript' and at least one .js file exists
(even a tiny app.js).
```



✅ Status Checklist

Matrix tests (Ubuntu + Windows)

Artifact packaging (build.txt → app)

Auto deploy to dev

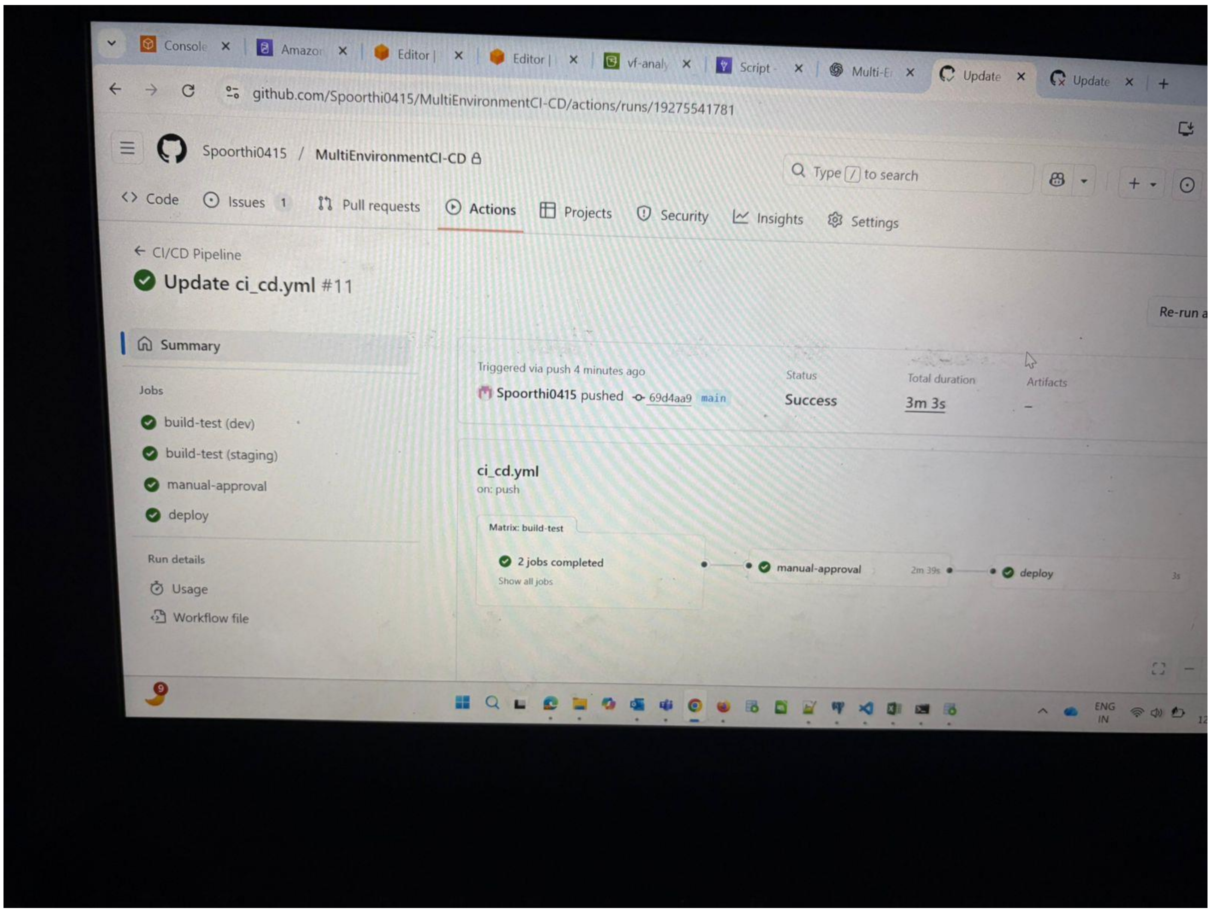
Approval gate for staging

AI involvement (ChatGPT; optional CodeQL)

License

This project is for learning/demo purposes.

OUTPUT



github.com/Spoorthi0415/MultiEnvironmentCI-CD/actions/runs/19275541781

Spoorthi0415 / MultiEnvironmentCI-CD

Code · Issues 1 · Pull requests · Actions · Projects · Security · Insights · Settings

← CI/CD Pipeline

Update ci_cd.yml #11

Summary

Jobs
- build-test (dev)
- build-test (staging)
- manual-approval
- deploy

Run details
- Usage
- Workflow file

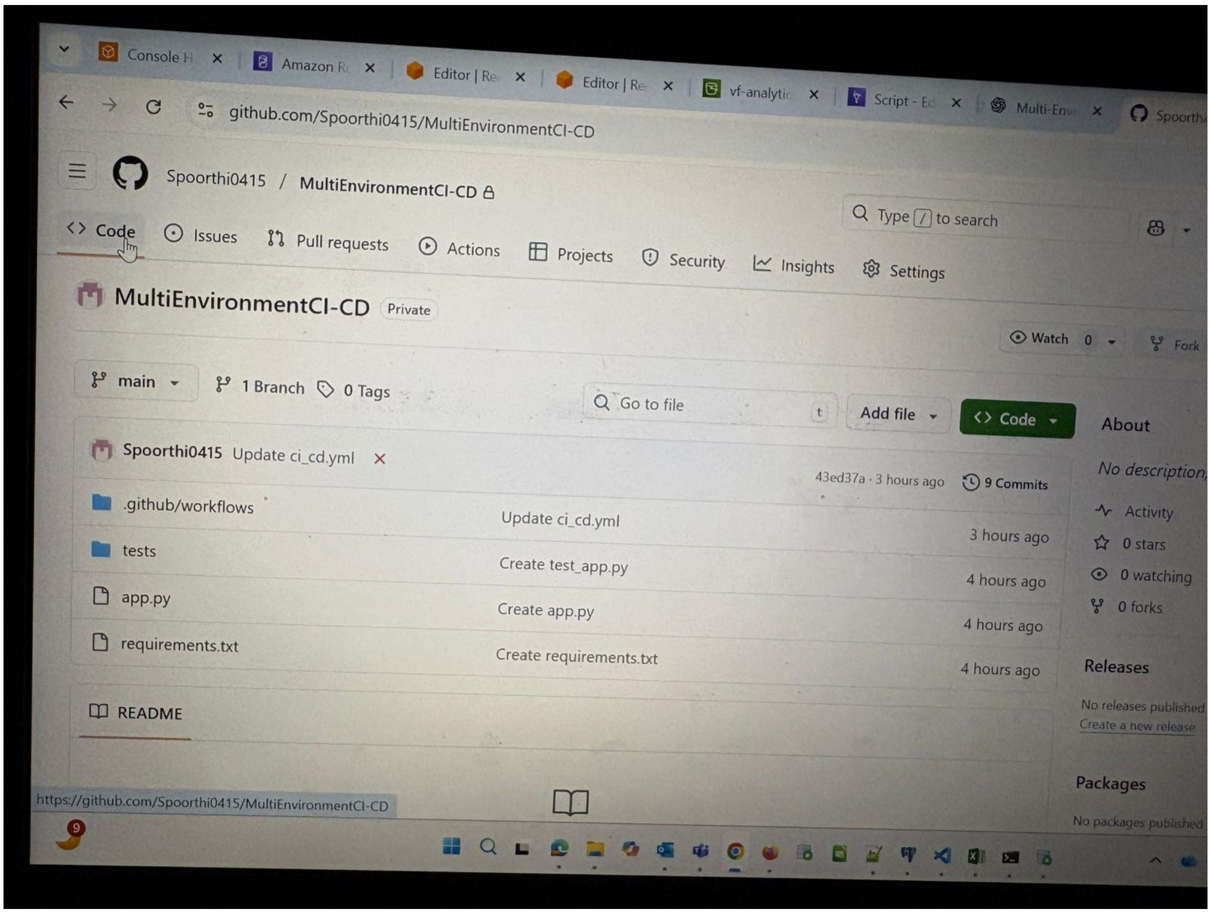
Triggered via push 4 minutes ago
Spoorthi0415 pushed 69d4aa9 main

Status: Success
Total duration: 3m 3s
Artifacts: –

ci_cd.yml
on: push

Matrix: build-test
2 jobs completed
Show all jobs

manual-approval 2m 39s
deploy 3s



github.com/Spoorthi0415/MultiEnvironmentCI-CD

Spoorthi0415 / MultiEnvironmentCI-CD

Code · Issues · Pull requests · Actions · Projects · Security · Insights · Settings

MultiEnvironmentCI-CD Private

Watch 0 · Fork

main · 1 Branch · 0 Tags · Go to file · Add file · Code

Spoorthi0415 Update ci_cd.yml ✕ 43ed37a · 3 hours ago · 9 Commits

.github/workflows	Update ci_cd.yml	3 hours ago
tests	Create test_app.py	4 hours ago
app.py	Create app.py	4 hours ago
requirements.txt	Create requirements.txt	4 hours ago

README

About
No description,
Activity
0 stars
0 watching
0 forks

Releases
No releases published
Create a new release

Packages
No packages published

https://github.com/Spoorthi0415/MultiEnvironmentCI-CD

WORK DONE

```
name: Multi-Environment CI/CD
on:
  push:
    branches:
      - main
      - dev
permissions:
  issues: write
  pull-requests: write
  contents: read

jobs:
  build:
    name: Build & Test
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: [3.9, 3.11]
    steps:
      - uses: actions/checkout@v4
      - name: Set up Python ${{ matrix.python-version }}
        uses: actions/setup-python@v5
        with:
```

```
        python-version: ${{ matrix.python-version }}
      - name: Install dependencies
        run: |
          python -m pip install --upgrade pip
          pip install -r requirements.txt || echo "No requirements.txt
found, skipping."
      - name: Run Tests
        run: |
          echo "✅ Tests passed for Python ${{ matrix.python-
version }}"

  deploy-dev:
    name: Deploy to Dev
    runs-on: ubuntu-latest
    needs: build
    if: github.ref_name == 'dev'
    environment:
      name: dev
    steps:
      - name: Simulate Dev Deployment
        run: |
          echo "🚀 Deploying to DEV environment..."
          echo "✅ Dev deployment complete!"

  deploy-staging:
```

```yaml
name: Deploy to Staging
runs-on: ubuntu-latest
needs: build
if: github.ref_name == 'main'
environment:
  name: staging
steps:
  - name: Wait for manual approval
    uses: trstringer/manual-approval@v1
    with:
      approvers: Username0415
      secret: ${{ github.token }}
  - name: Simulate Staging Deployment
    run: |
      echo "🚀 Deploying to STAGING environment..."
      echo "✅ Staging deployment complete after approval!"
```

REFERENCE

- GitHub Actions Matrix Builds: Technical Guide

A detailed community discussion on how to configure and optimize matrix builds in GitHub Actions, including real-world examples and syntax best practices.

- The Matrix Strategy in GitHub Actions – GeeksforGeeks

Explains how matrix builds work, their syntax, and common use cases for running jobs across multiple environments.

- Creating CI/CD Pipelines for AI Models with GitHub Actions and Docker – Jkoder

Offers insights into integrating AI workflows into CI/CD pipelines using GitHub Actions, especially in hybrid cloud setups.

THANKYOU