DEVOPS PRACTICAL

Experiment: CI/CD Pipeline using GitHub Actions

AIM

To create a complete CI/CD pipeline using GitHub Actions that automates code quality checks, testing, security scanning, Docker image creation, and deployment to staging and production environments.

CODE

```
name: CI/CD Pipeline
on:
  push:
    branches: [ main, develop ]
  pull_request:
    branches: [ main, develop ]
  workflow_dispatch:

env:
  NODE_VERSION: '18.x'
  REGISTRY: ghcr.io
  IMAGE_NAME: ${{ github.repository }}

jobs:
  code-quality:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v4
        with:
          node-version: ${{ env.NODE_VERSION }}
          cache: 'npm'
      - run: npm ci
      - run: npm run lint
      - run: npm run format:check

  test:
    runs-on: ubuntu-latest
    needs: code-quality
    strategy:
      matrix:
        node-version: [16.x, 18.x, 20.x]
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v4
        with:
          node-version: ${{ matrix.node-version }}
          cache: 'npm'
      - run: npm ci
      - run: npm run test:unit
      - run: npm run test:integration
      - run: npm run test:coverage
```

DEVOPS PRACTICAL

```
      - uses: codecov/codecov-action@v3
        with:
          files: ./coverage/coverage-final.json

  security:
    runs-on: ubuntu-latest
    needs: code-quality
    steps:
      - uses: actions/checkout@v4
      - run: npm audit --audit-level=moderate
      - uses: snyk/actions/node@master
        continue-on-error: true
        env:
          SNYK_TOKEN: ${{ secrets.SNYK_TOKEN }}

  build:
    runs-on: ubuntu-latest
    needs: [test, security]
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v4
        with:
          node-version: ${{ env.NODE_VERSION }}
          cache: 'npm'
      - run: npm ci
      - run: npm run build
      - uses: actions/upload-artifact@v4
        with:
          name: build-artifacts
          path: dist/

  docker:
    runs-on: ubuntu-latest
    needs: build
    if: github.event_name == 'push' && github.ref == 'refs/heads/main'
    permissions:
      contents: read
      packages: write
    steps:
      - uses: actions/checkout@v4
      - uses: docker/setup-buildx-action@v3
      - uses: docker/login-action@v3
        with:
          registry: ${{ env.REGISTRY }}
          username: ${{ github.actor }}
          password: ${{ secrets.GITHUB_TOKEN }}
      - uses: docker/metadata-action@v5
        id: meta
        with:
          images: ${{ env.REGISTRY }}/${{ env.IMAGE_NAME }}
      - uses: docker/build-push-action@v5
        with:
          context: .
          push: true
          tags: ${{ steps.meta.outputs.tags }}
```

DEVOPS PRACTICAL

```
  deploy-staging:
    runs-on: ubuntu-latest
    needs: docker
    if: github.ref == 'refs/heads/develop'
    steps:
      - uses: actions/checkout@v4
      - run: echo "Deploying to staging environment..."

  deploy-production:
    runs-on: ubuntu-latest
    needs: docker
    if: github.ref == 'refs/heads/main'
    steps:
      - uses: actions/checkout@v4
      - uses: actions/download-artifact@v4
        with:
          name: build-artifacts
          path: dist/
      - run: echo "Deploying to production environment..."

  rollback:
    runs-on: ubuntu-latest
    if: github.event_name == 'workflow_dispatch'
    steps:
      - uses: actions/checkout@v4
      - run: echo "Rolling back to previous version..."
```

EXPECTED OUTPUT

The GitHub Actions workflow automates a complete CI/CD pipeline that performs:

1. Linting and code formatting checks.
2. Running tests across multiple Node.js versions.
3. Security scanning using npm audit and Snyk.
4. Building and pushing a Docker image to GitHub Container Registry.
5. Deploying to staging and production environments.
6. Sending notifications on success or rollback.