EXPERIMENT NO 5.2

## Aim

To set up a Continuous Integration and Continuous Deployment (CI/CD) pipeline using GitHub Actions, which will automate building, testing, and deploying an application whenever code changes are pushed to the repository.

## Theory

### What is CI/CD?

* Continuous Integration (CI): Developers frequently merge their code changes into a central repository where automated builds and tests are run.
* Continuous Deployment (CD): Every change that passes the automated tests is automatically deployed to a staging or production environment.

This ensures:

* Faster development cycles.
* Fewer integration problems.
* Early bug detection.
* Consistent deployment process.

### What is GitHub Actions?

* A CI/CD tool built into GitHub.
* Allows you to create workflows that run automatically based on triggers (push, pull request, schedule, etc.).
* Workflows are defined using YAML files inside .github/workflows/.
* Provides runners (Linux, Windows, macOS) to execute jobs.

## Code (Workflow YAML Example)

Here’s a simple Node.js app CI/CD pipeline using GitHub Actions.

Create a file:  
 .github/workflows/ci-cd.yml

name: CI/CD Pipeline

# Trigger workflow on push or pull request to main branch

on:

push:

branches: [ "main" ]

pull\_request:

branches: [ "main" ]

jobs:

build-and-test:

runs-on: ubuntu-latest

steps:

- name: Checkout repository

uses: actions/checkout@v3

- name: Set up Node.js

uses: actions/setup-node@v3

with:

node-version: "18"

- name: Install dependencies

run: npm install

- name: Run tests

run: npm test

deploy:

runs-on: ubuntu-latest

needs: build-and-test

if: github.ref == 'refs/heads/main'

steps:

- name: Checkout repository

uses: actions/checkout@v3

- name: Deploy to Server (Example: GitHub Pages)

uses: peaceiris/actions-gh-pages@v3

with:

github\_token: ${{ secrets.GITHUB\_TOKEN }}

publish\_dir: ./dist

This workflow does:

1. Runs when code is pushed or pull request is made to main.
2. Build & Test Job: Checks out code → installs dependencies → runs tests.
3. Deploy Job: If tests pass and branch is main, it deploys (here example is GitHub Pages).

## Learning Outcomes

By the end of this practical, you will be able to:

1. Understand the concepts of Continuous Integration and Continuous Deployment.
2. Configure GitHub Actions to automate build, test, and deployment pipelines.
3. Use YAML to define workflows in .github/workflows/.
4. Set up triggers (push, pull request) to automatically start workflows.
5. Deploy applications (static sites, Node.js apps, Docker containers, etc.) using GitHub Actions.
6. Improve software quality, reliability, and delivery speed through automation.