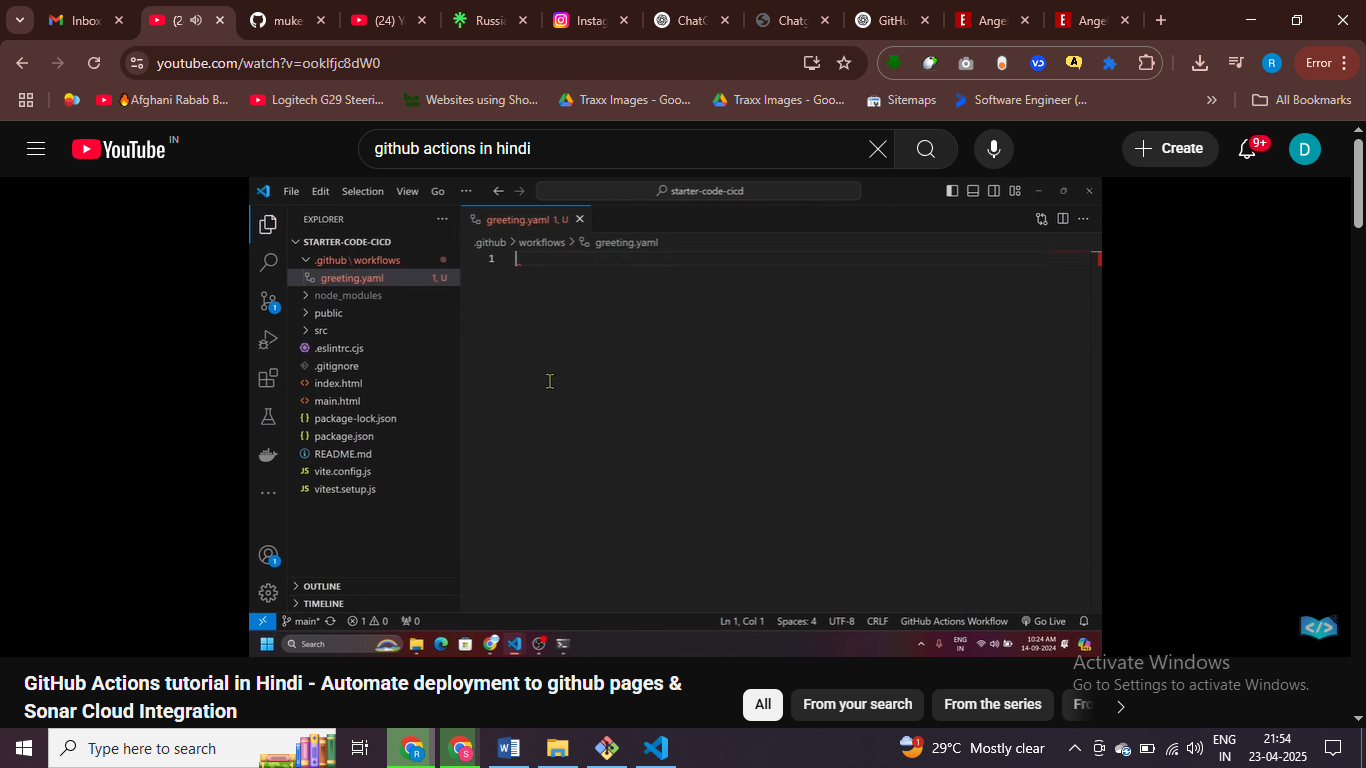


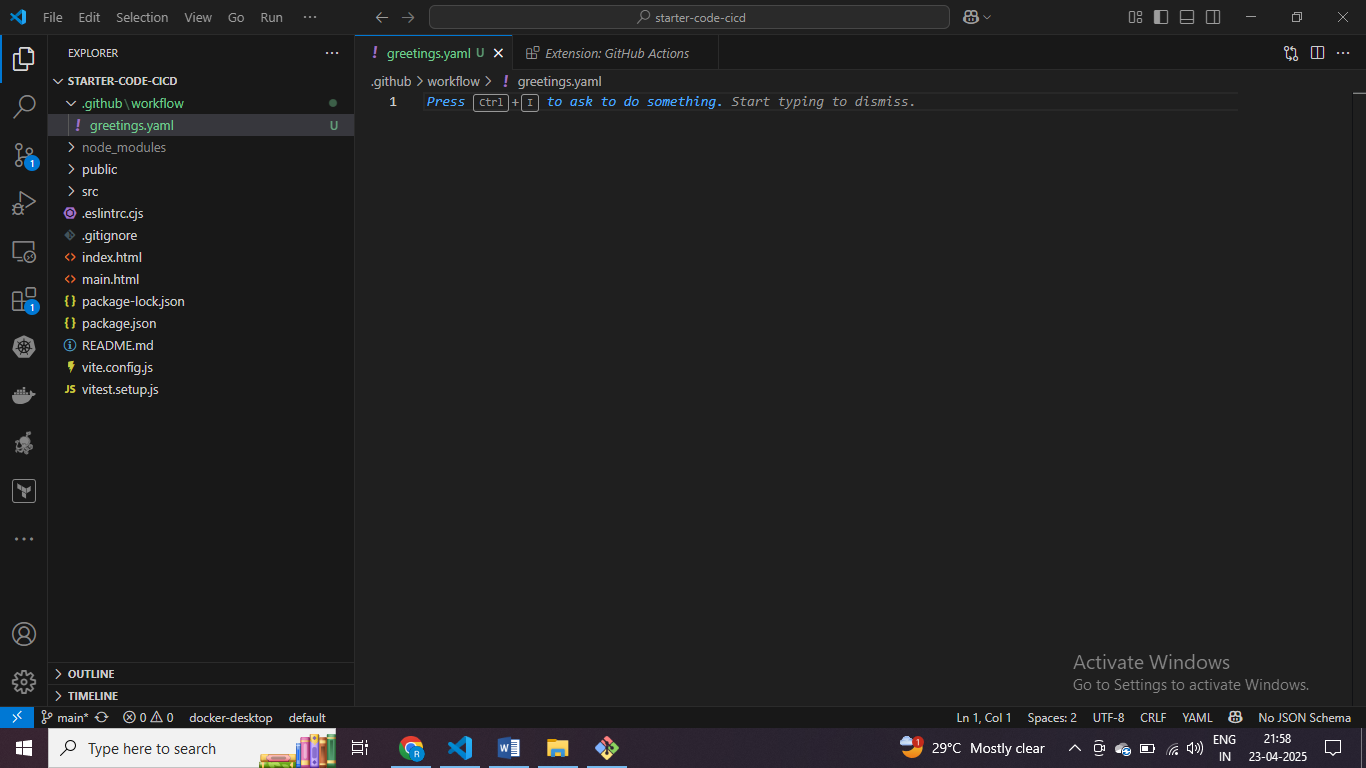
# starter project we have

Npm install

Npm run dev



Created a workflow project in which we write syntax , like folders and all after that write files



This is folder sturcutre that we have .github/workflow and inside that we have greetings.yaml

name: workflow

on: push

name of workflow and this workflow will be trigger based upon the push , code push karenge toh ye workflowew trgier hoga

toh ye event hai hamara yha paa

multiple event dalna hai toh aarray main dal do ye sab

workflow kuch task perform karega unko ham jobs ka andar likehgne

name: workflow

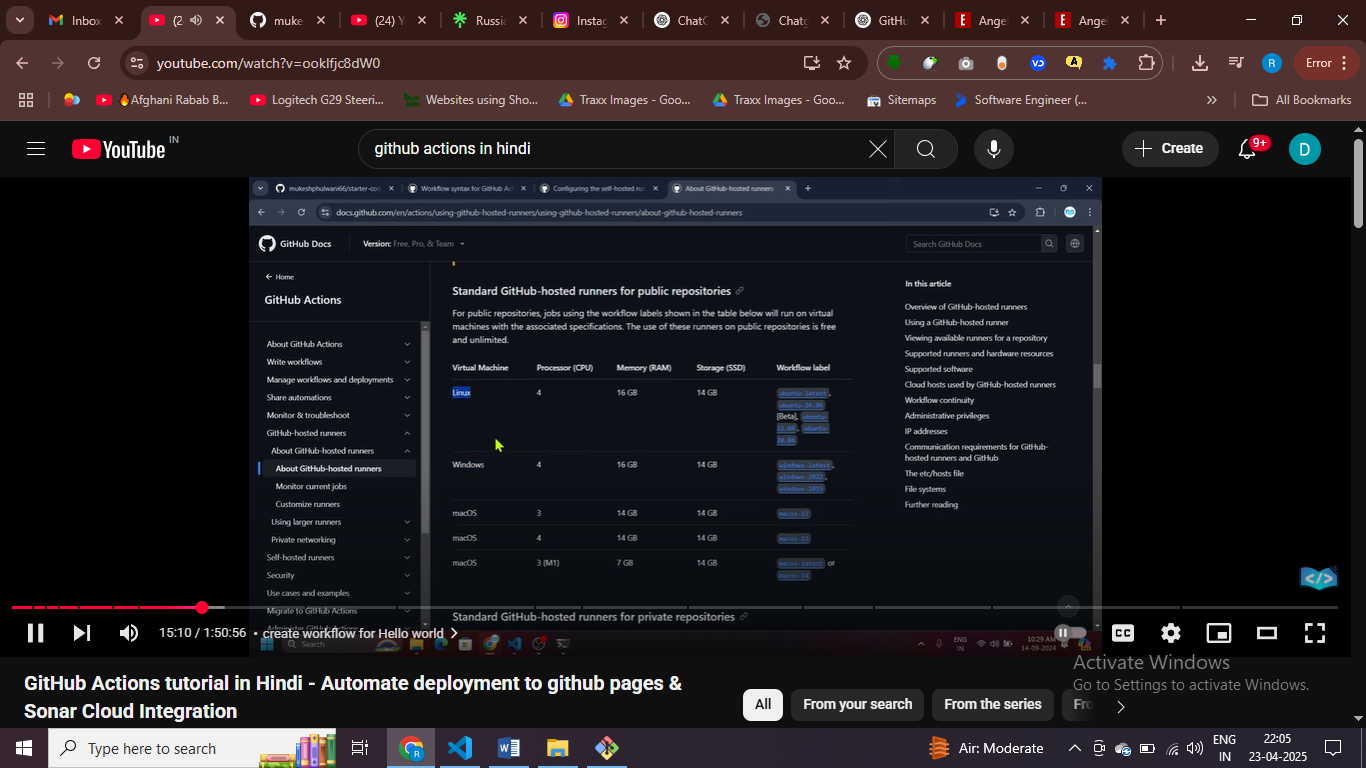
on: push

jobs:

    greet:

job name Is greet – so greet print hello world

toh jo bhi code hai who remote machine pa chalge



Code remote machine pa chagle who hai remote machine hamari ,

Here it linux or Ubuntu based lineux machine pa chaglega hamra code

name: workflow

on: push

jobs:

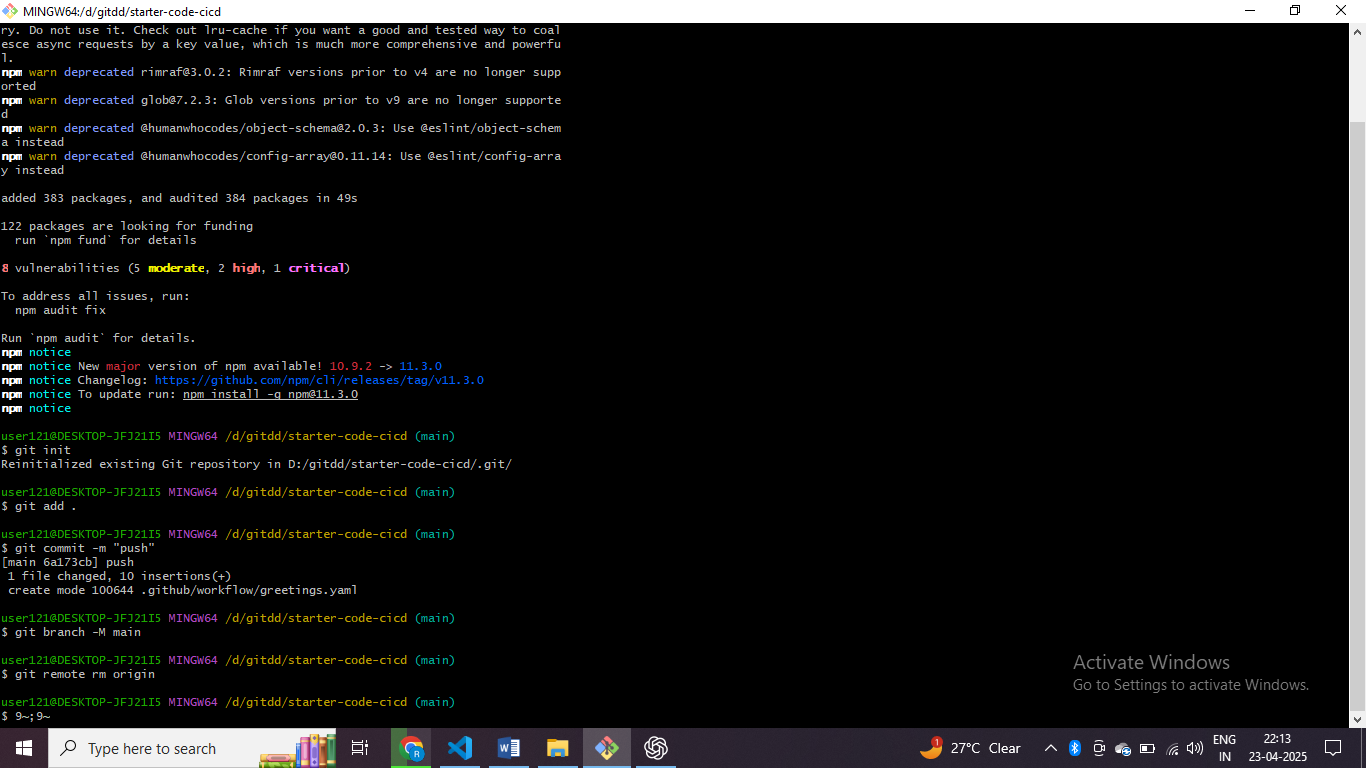
  greet:

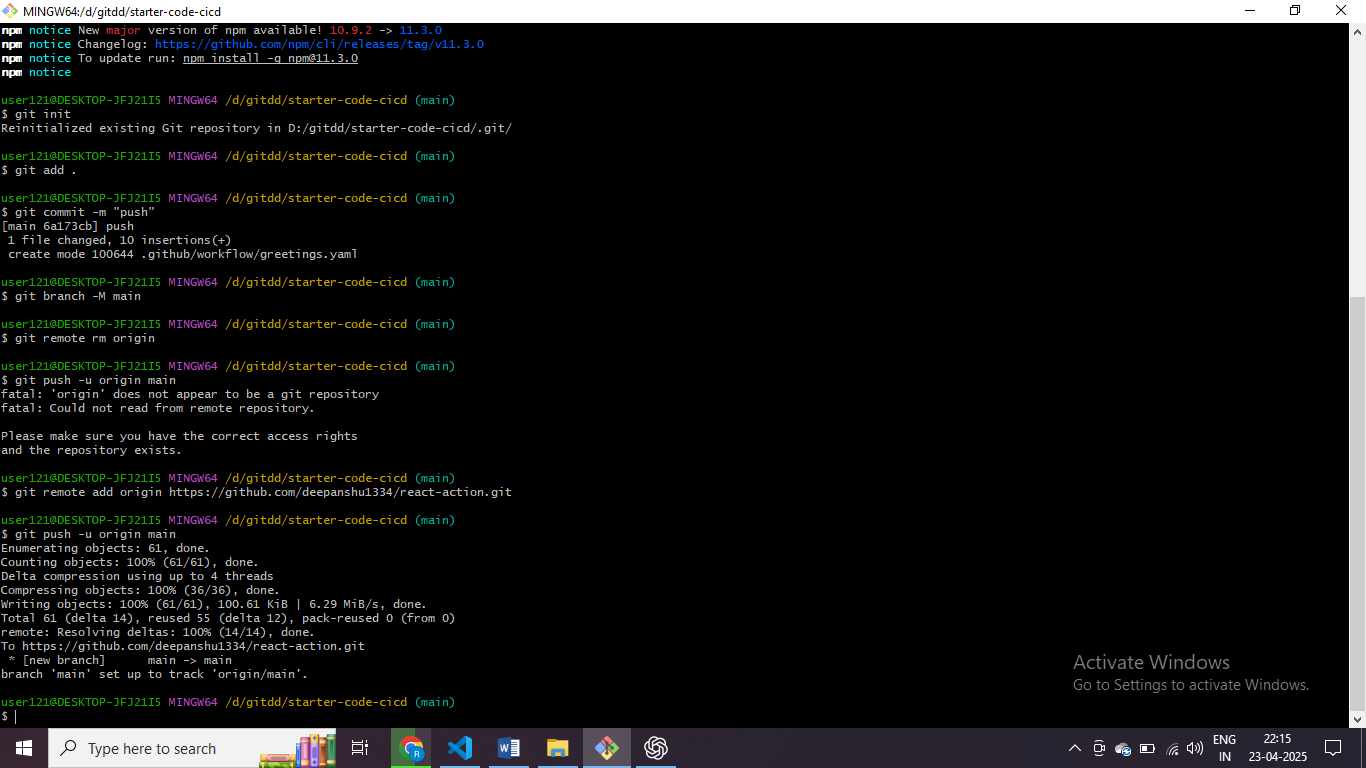
    runs-on: ubuntu-latest

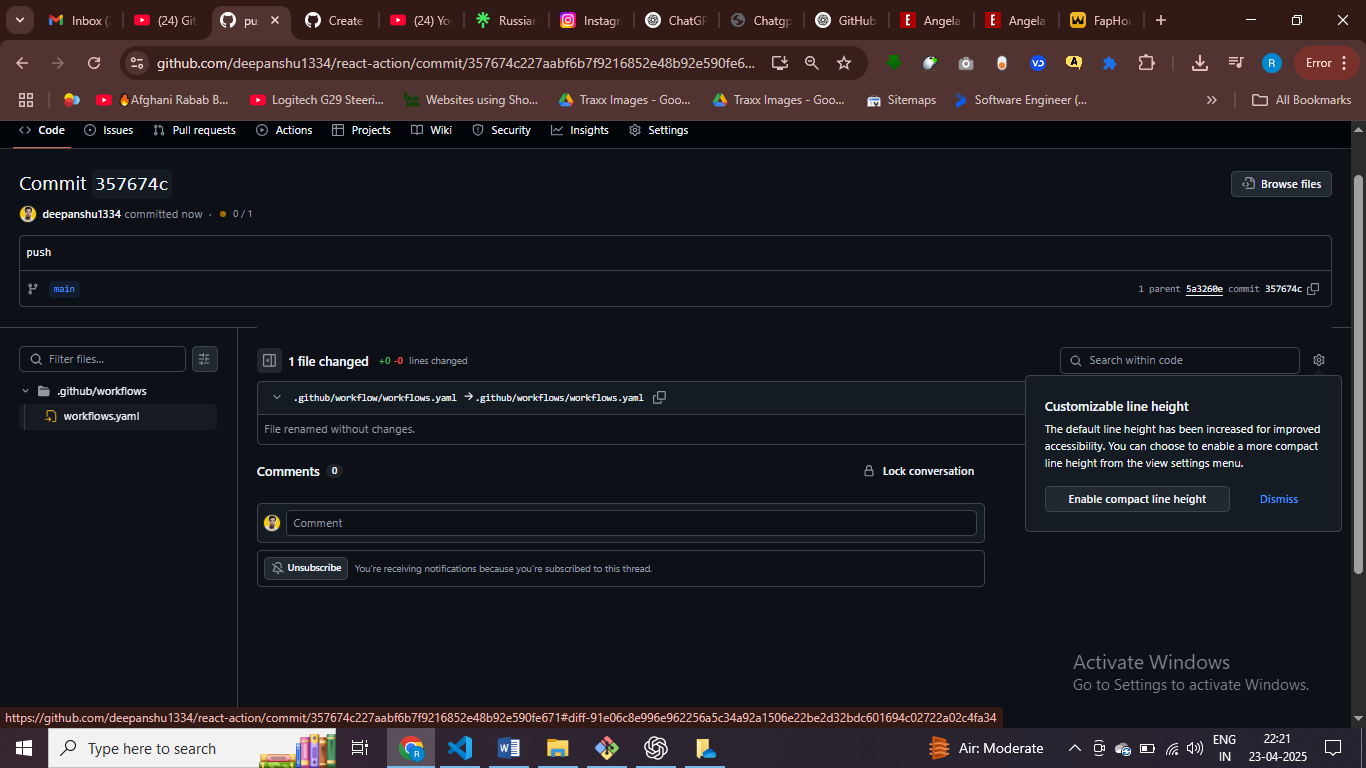
    steps:

      - name: greet-user

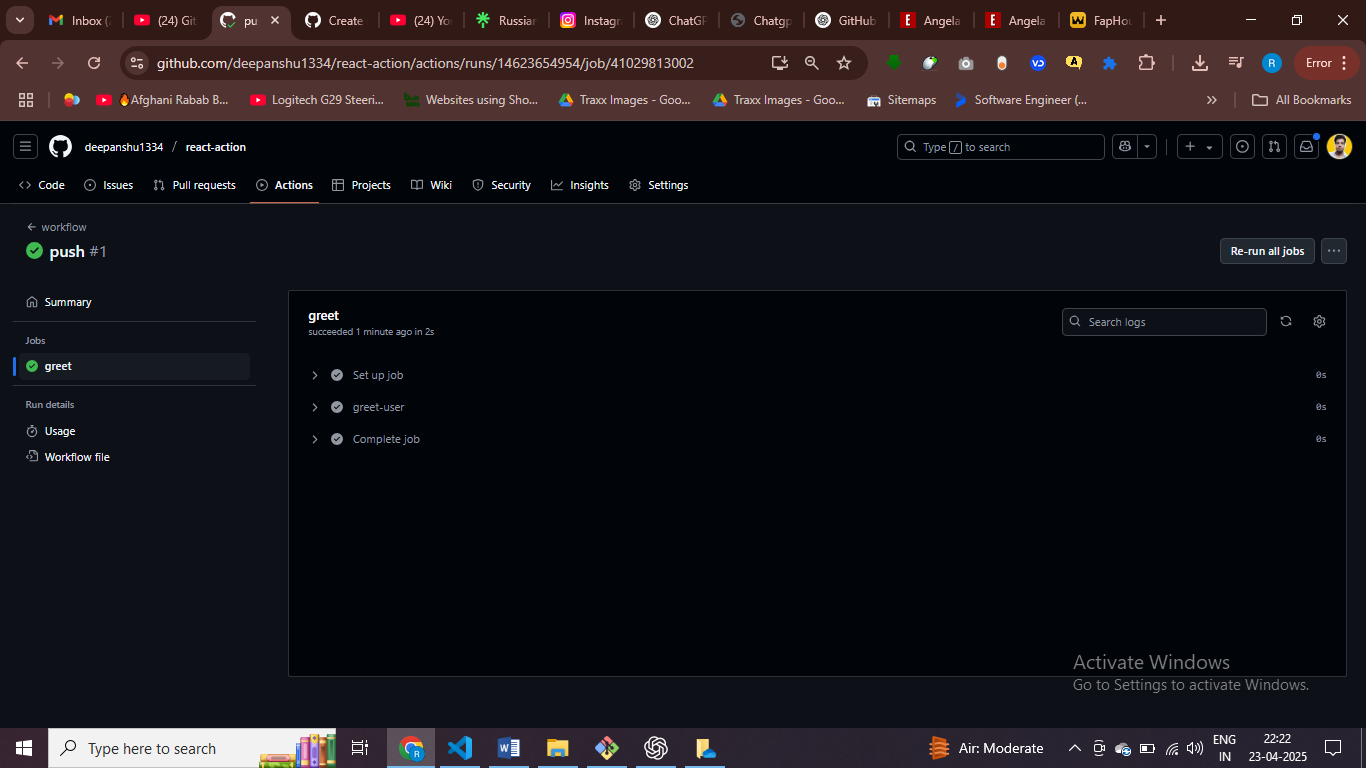
        run: echo "Hello-world"







Workflow has been trigger succefully

to print hello world we write a workflow here

name: workflow

on: push

jobs:

  greet:

    runs-on: ubuntu-latest

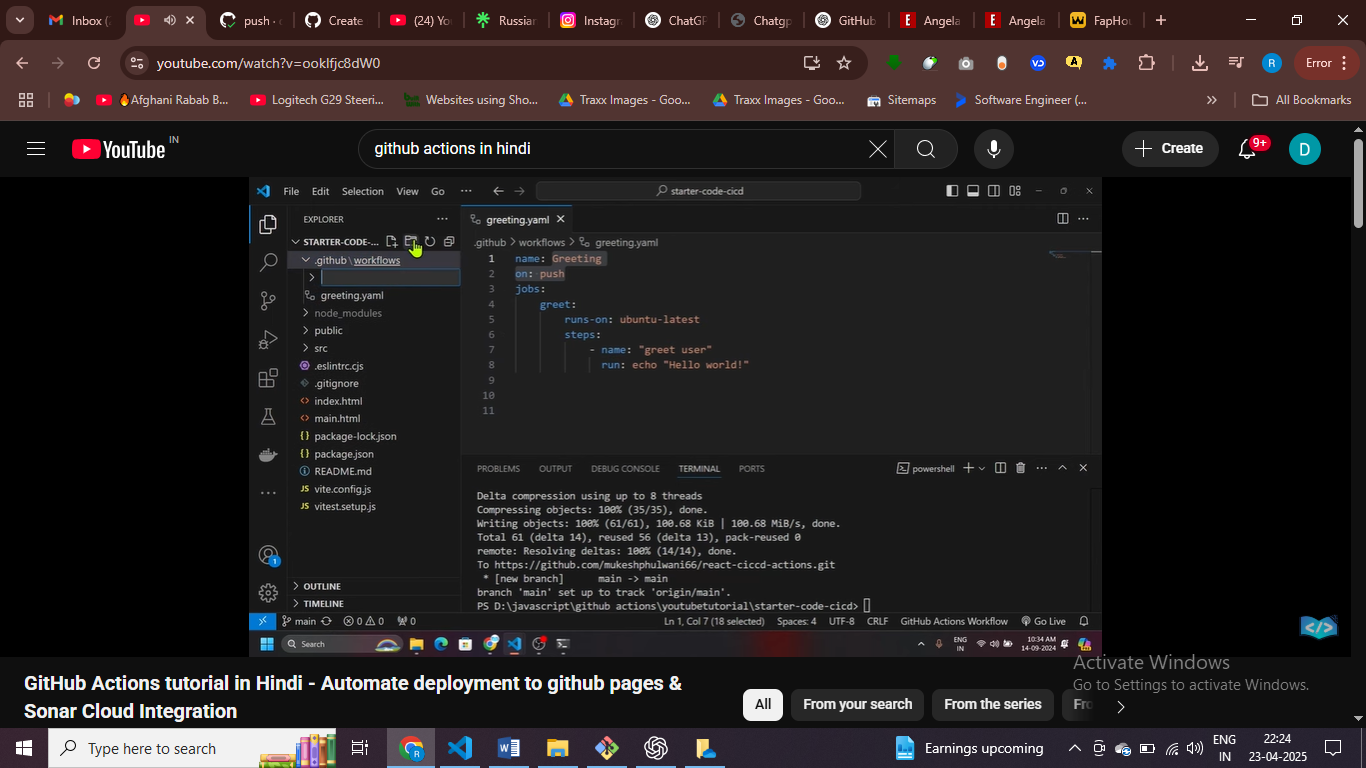
    steps:

      - name: greet-user

        run: echo "Hello-world"

name of workflow is workflow

we have attach the workflow with rppositry

jaise hi code push hua workflow apne app trigger ho jayega

name: Deploy

on: [push, workflow\_dispatch]

jobs:

  greet:

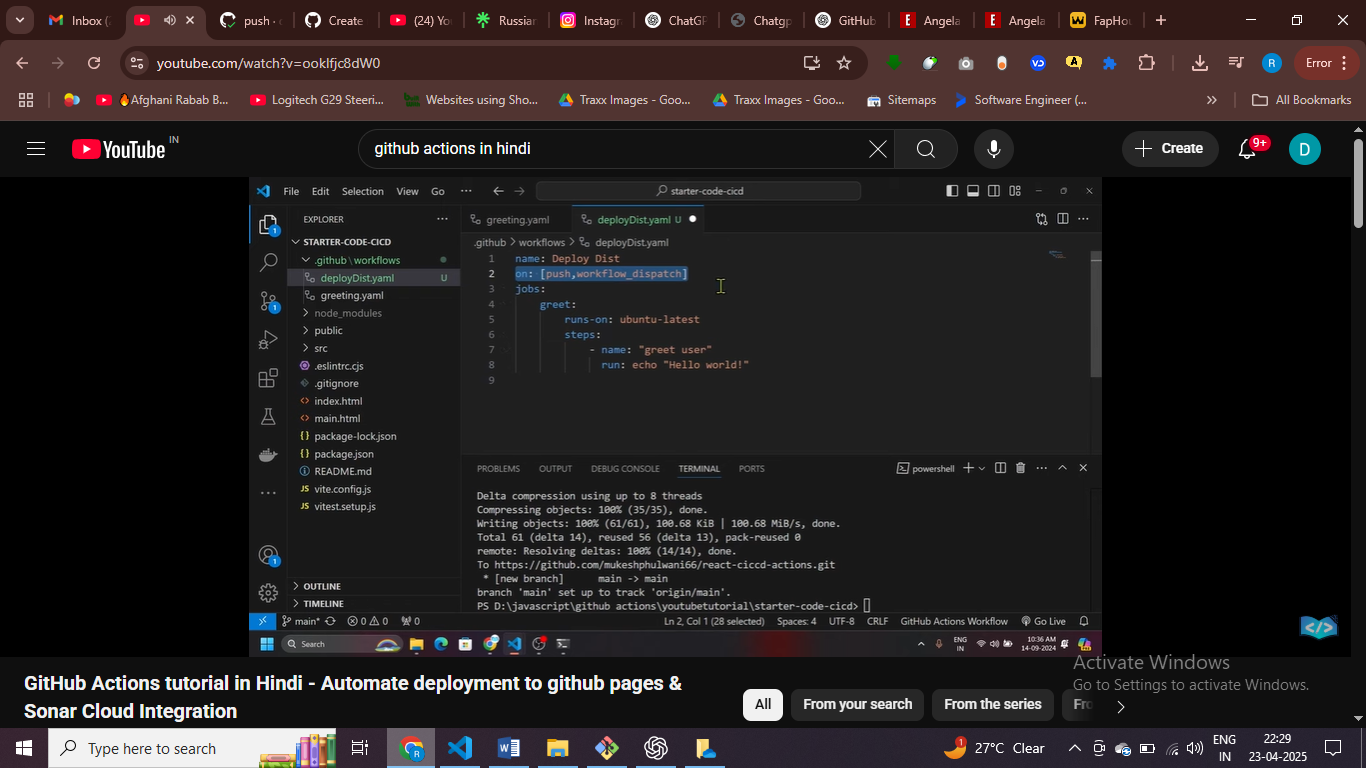
    runs-on: ubuntu-latest

    steps:

      - name: greet-user

        run: echo "Hello-world"

# buttom click then workflow trigger – using dipay dispatch event

#push pa bhi trigger karo aur manually bhi trigger karo

name: Deploy

on: [push, workflow\_dispatch]

jobs:

  test:

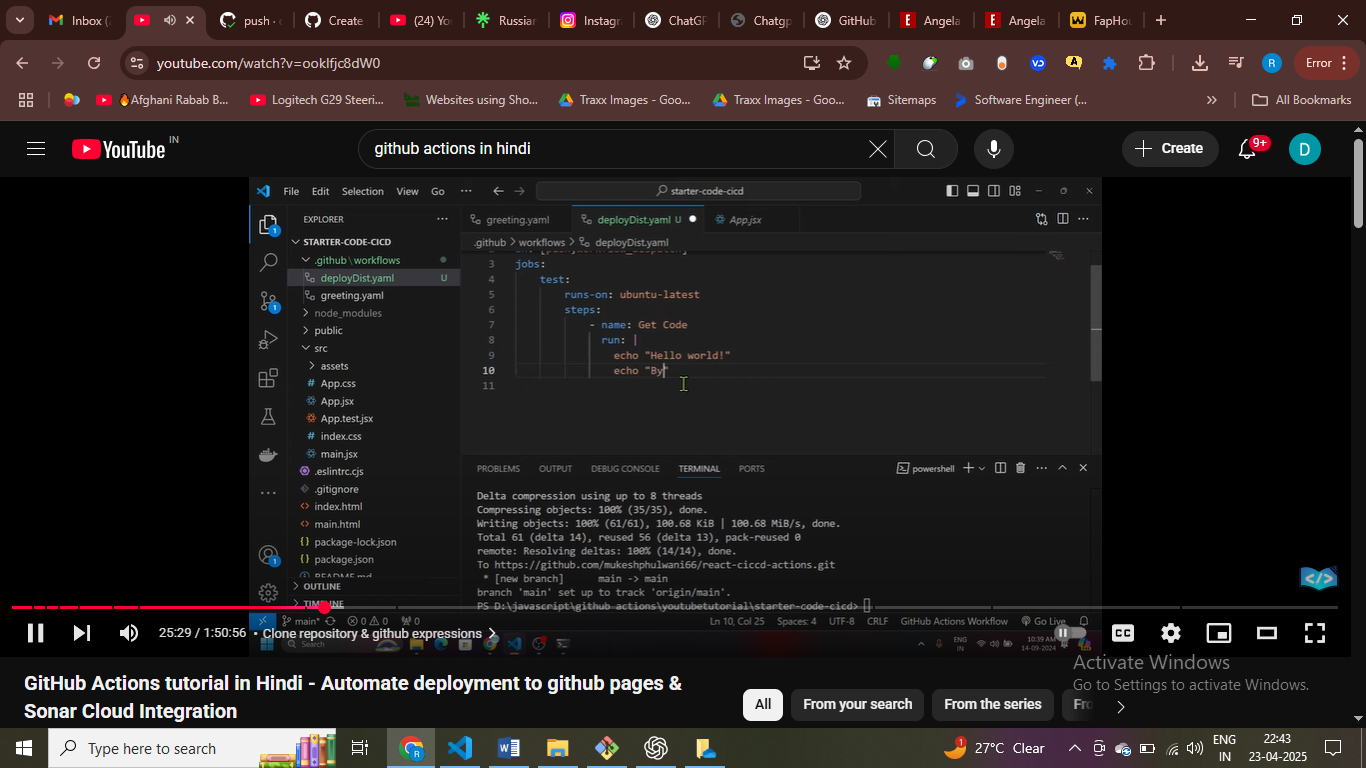
    runs-on: ubuntu-latest

    steps:

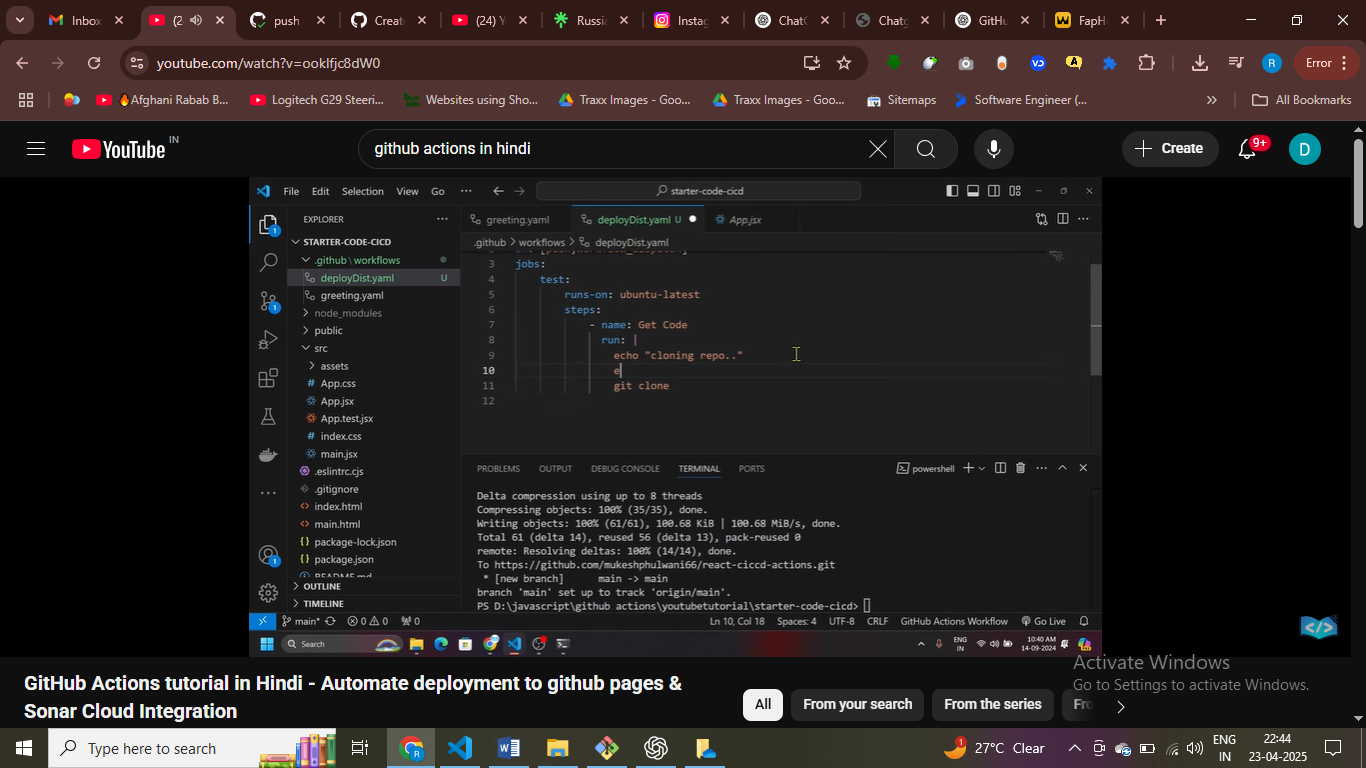
      - name: Get-code

        run: echo "Hello-world"

define task and task name



Use pipe for multiple cmd

we can manually clone a repo but we don’t want to hardode the variables

name: Deploy

on: [push, workflow\_dispatch]

jobs:

  test:

    runs-on: ubuntu-latest

    steps:

      - name: Get-code

        run: |

          echo "${{ github }}"

          echo "cloning a repo"

here I am using a context {{github}] valirble

name: Deploy

on: [push, workflow\_dispatch]

jobs:

  test:

    runs-on: ubuntu-latest

    steps:

      - name: Checkout code

        uses: actions/checkout@v3

      - name: Print GitHub context and clone repo

        run: |

          echo "${{ toJson(github) }}"

          echo "Cloning the repo"

          git clone https://github.com/${{ github.repository }}.git

cloning a repo without hardocign using github varibales

seeing the json object and geeting all the information

 echo "${{ toJson(github) }}"

syntax of experiesssion – this is an eperession syntax

successfully accessint the repo

github markepalace

use action to clone the repo – use checkout action to clone the repo

name: Deploy

on: [push, workflow\_dispatch]

jobs:

  test:

    runs-on: ubuntu-latest

    steps:

      - name: Checkout code

        uses: actions/checkout@v4  # You can use @v4 directly here instead of v3

    #   - name: Print GitHub context and clone repo

    #     run: |

    #       echo "${{ toJson(github) }}"

    #       echo "Cloning the repo"

    #       git clone https://github.com/${{ github.repository }}.git

Scucceuly clone the rpo -using checkout actions to get clone the repo

Now perform test

name: Deploy

on: [push, workflow\_dispatch]

jobs:

  test:

    runs-on: ubuntu-latest

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

isnaling node we are using uses beacse we are using third party apps

name: Deploy

on: [push, workflow\_dispatch]

jobs:

  test:

    runs-on: ubuntu-latest

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

      - name: Print Node.js version

        run: node -v

      - name: Install dependencies

        run: npm ci

checking node verison in my machine and npm ci install specific version

npm ci package lock file ko use karta hai , exact verison ko install use npm ci – pcakge lock file ko use karta hai

npjm intall – use latest version ko install kar sakte hai –

afte r installing depency we can run the test cases

name: Deploy

on: [push, workflow\_dispatch]

jobs:

  test:

    runs-on: ubuntu-latest

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

      - name: Print Node.js version

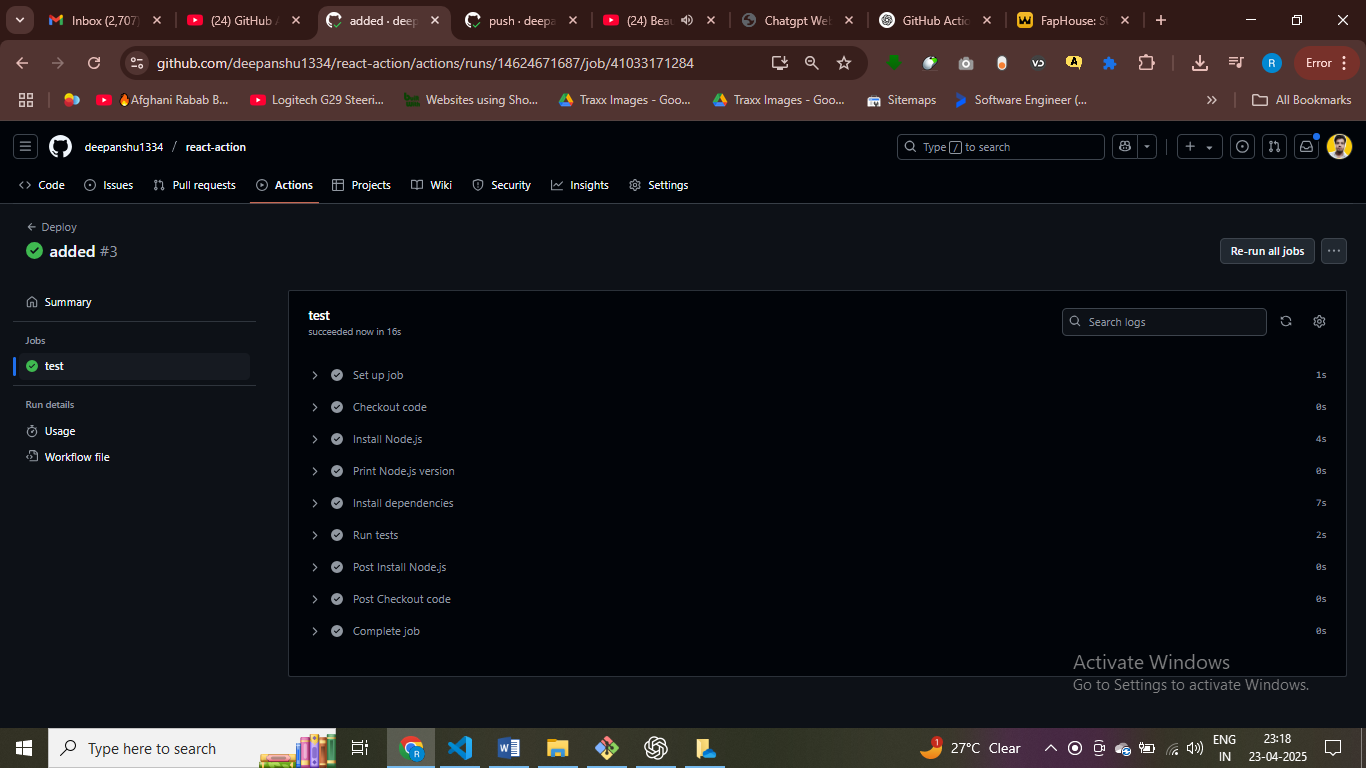
        run: node -v

      - name: Install dependencies

        run: npm ci

      - name: Run tests

        run: npm run test

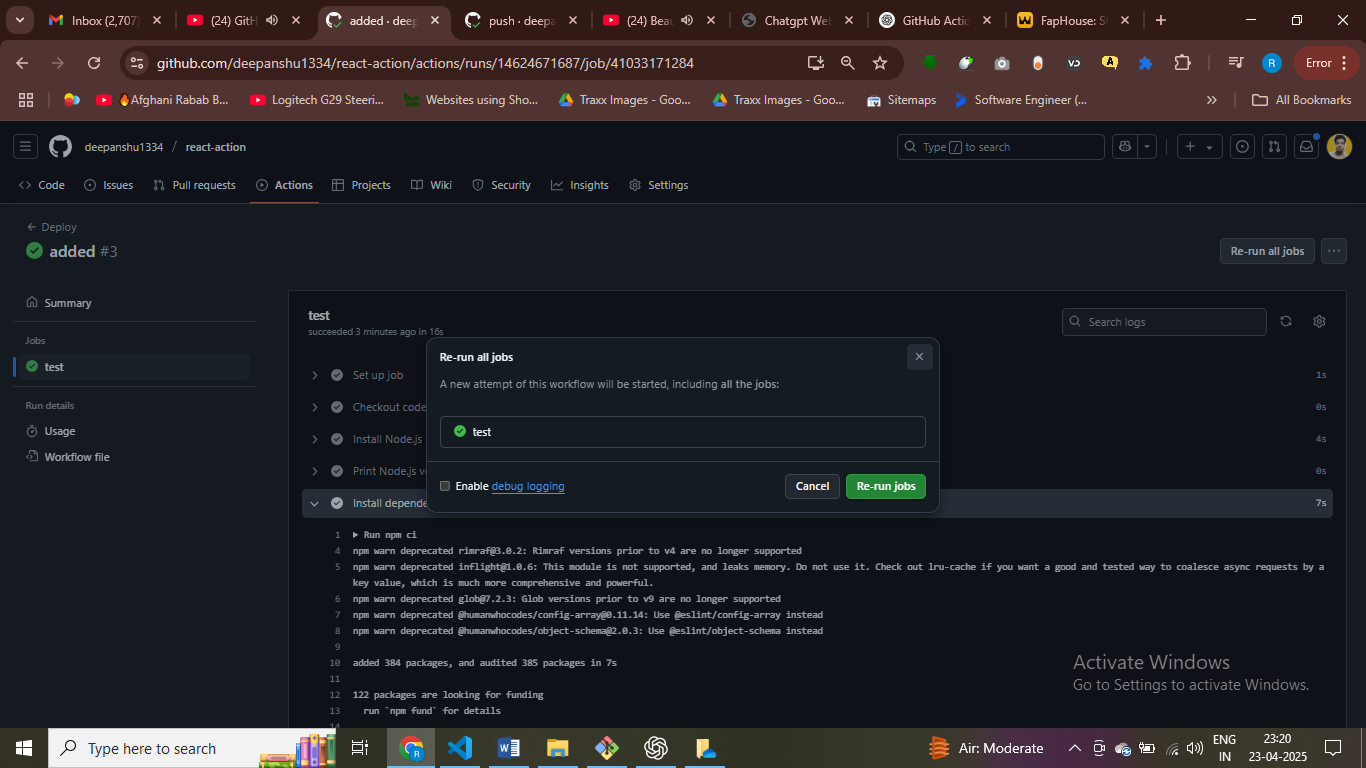


# so we are installing everything

Intall depency and run the test cases

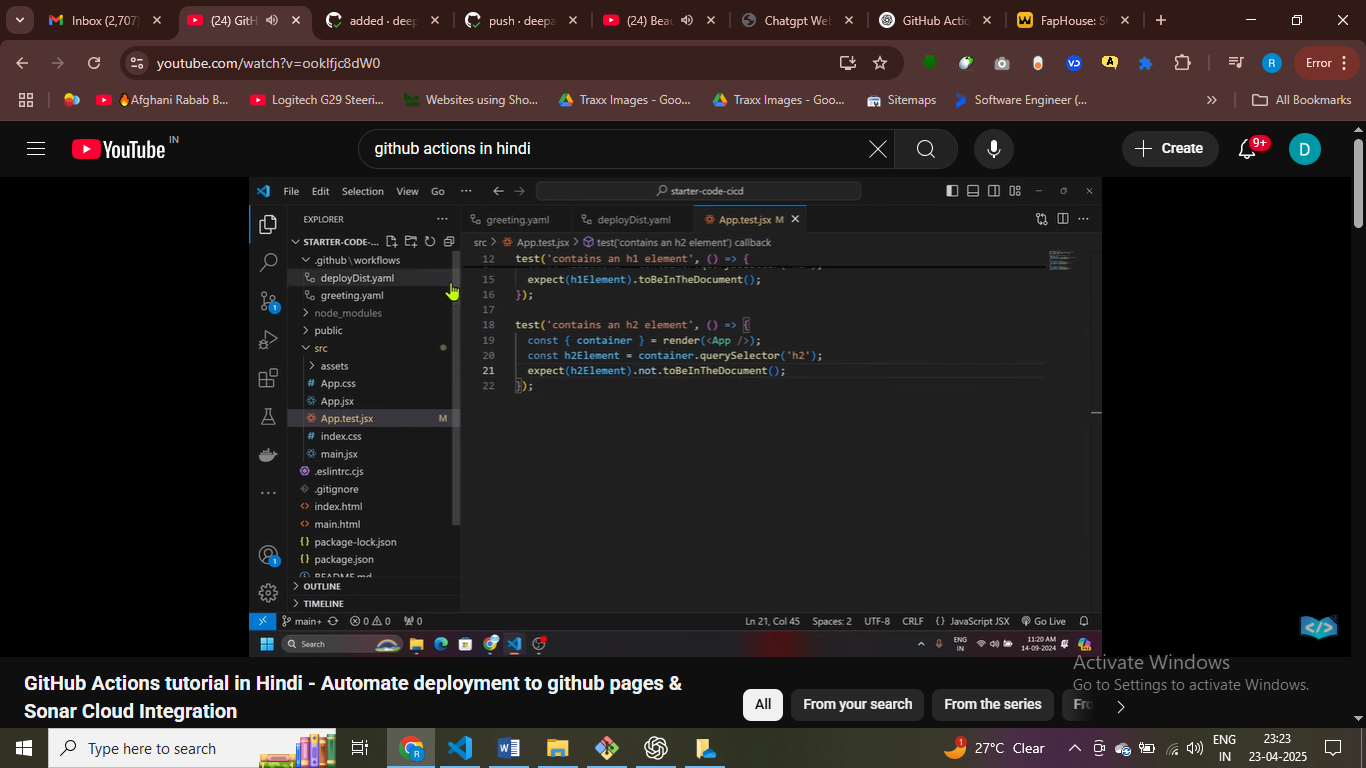
Code push kara workflow trigger ho jayega

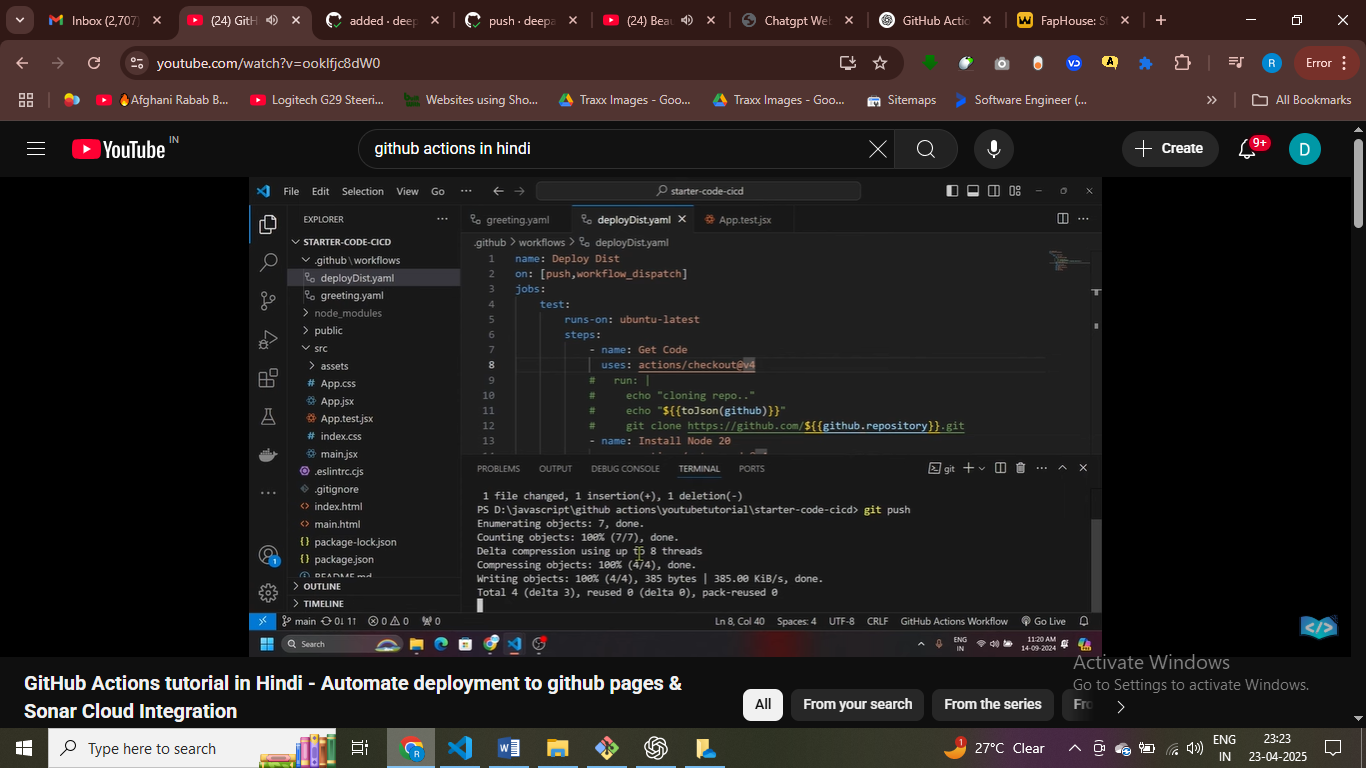
Jaise hi code push workrlfow trigger and install depency

here we have button that we give while workflow-dispatch

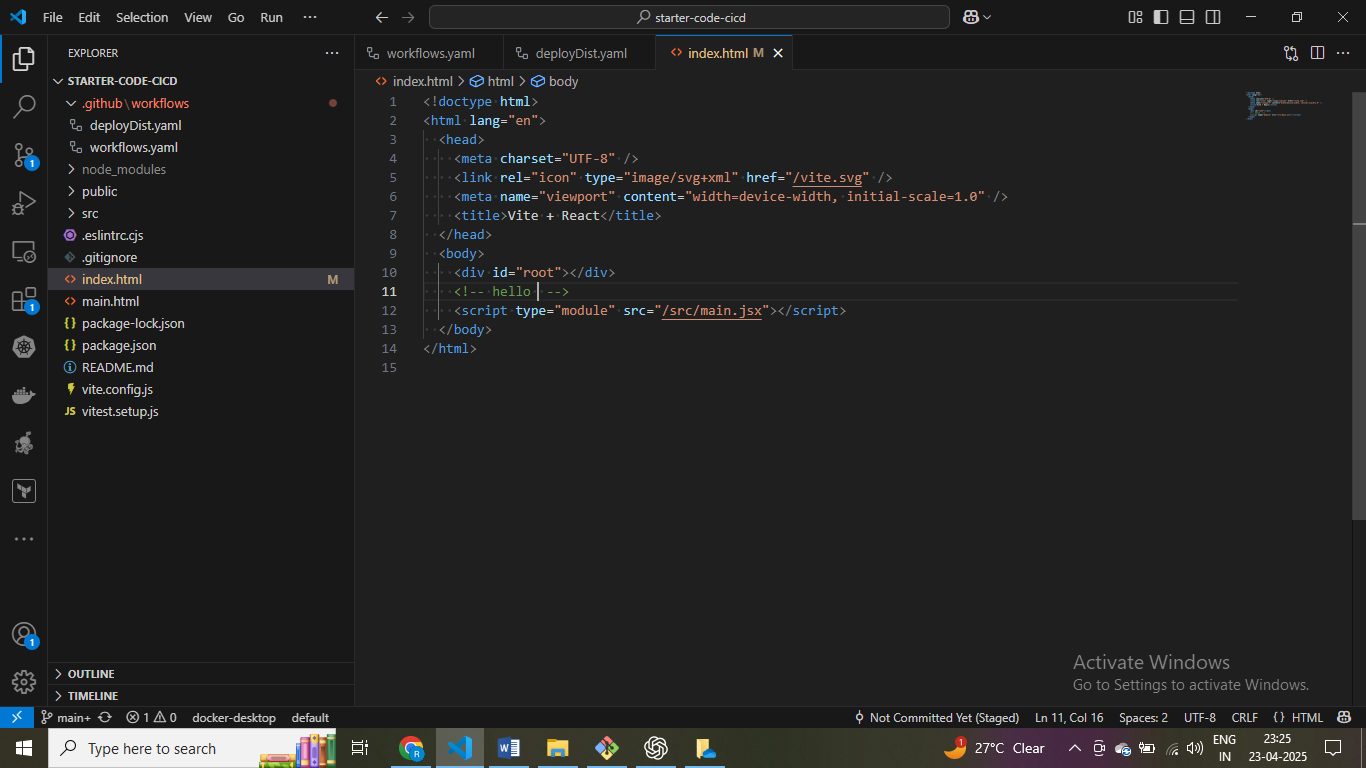
Kisi bhi branch main push hoga tph ye code trigger ho jayega

* Use filter to run only on specific branch



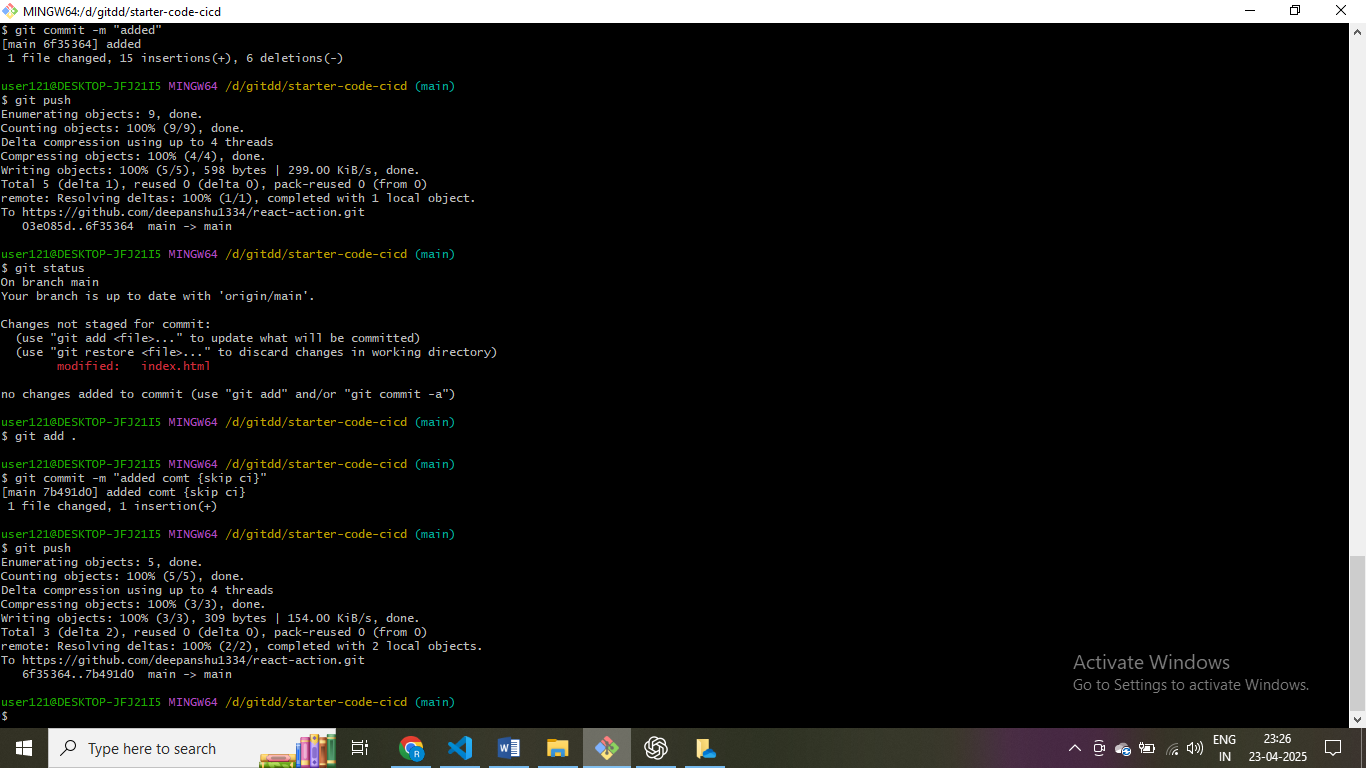


# automatiicaly workflow trigger hone chahiye hamra



$ I have made slight changes in the code

We don’t want to push this cmt

skip ci toh push apne aap trigger nahi hoga hamar

Deploy ka liye code chaiye hame

Toh ham second job likhenge

name: Deploy

on: [push, workflow\_dispatch]

jobs:

  test:

    runs-on: ubuntu-latest

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

      - name: Print Node.js version

        run: node -v

      - name: Install dependencies

        run: npm ci

      - name: Run tests

        run: npm run test

  deploy:

    runs-on: ubuntu-latest

    needs: test  # this ensures deploy only runs if test passes

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

      - name: Print Node.js version

        run: node -v

      - name: Install dependencies

        run: npm ci

      - name: Run tests

        run: npm run test

      - name: Build Project

        run: npm run build

name: Deploy

on: [push, workflow\_dispatch]

jobs:

  test:

    runs-on: ubuntu-latest

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

      - name: Print Node.js version

        run: node -v

      - name: Install dependencies

        run: npm ci

      - name: Run tests

        run: npm run test

  deploy:

    runs-on: ubuntu-latest

    needs: test  # this ensures deploy only runs if test passes

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

      - name: Print Node.js version

        run: node -v

      - name: Install dependencies

        run: npm ci

      - name: Run tests

        run: npm run test

      - name: Build Project

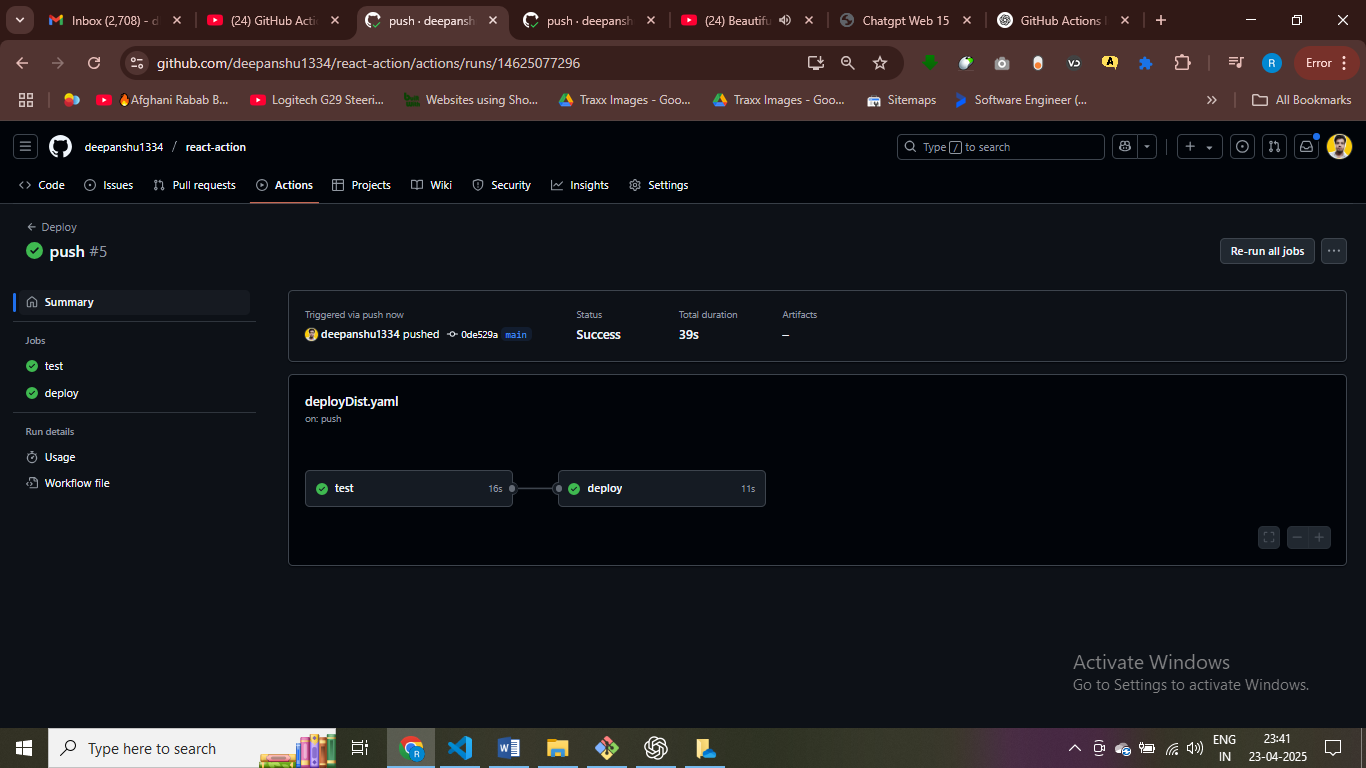
        run: npm run build

      - name: Deploy

        run: echo "Deploy scucesffule"

define jobs for deplouyement running on linux machine and again need code for deplouing so again cloning the job

after push

both tet and deploy are executeing at same time

name: Deploy

on: [push, workflow\_dispatch]

jobs:

  test:

    runs-on: ubuntu-latest

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

      - name: Print Node.js version

        run: node -v

      - name: Install dependencies

        run: npm ci

      - name: Run tests

        run: npm run test

  deploy:

    runs-on: ubuntu-latest

    needs: test  # this ensures deploy only runs if test passes

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

      - name: Print Node.js version

        run: node -v

      - name: Install dependencies

        run: npm ci

      - name: Run tests

        run: npm run test

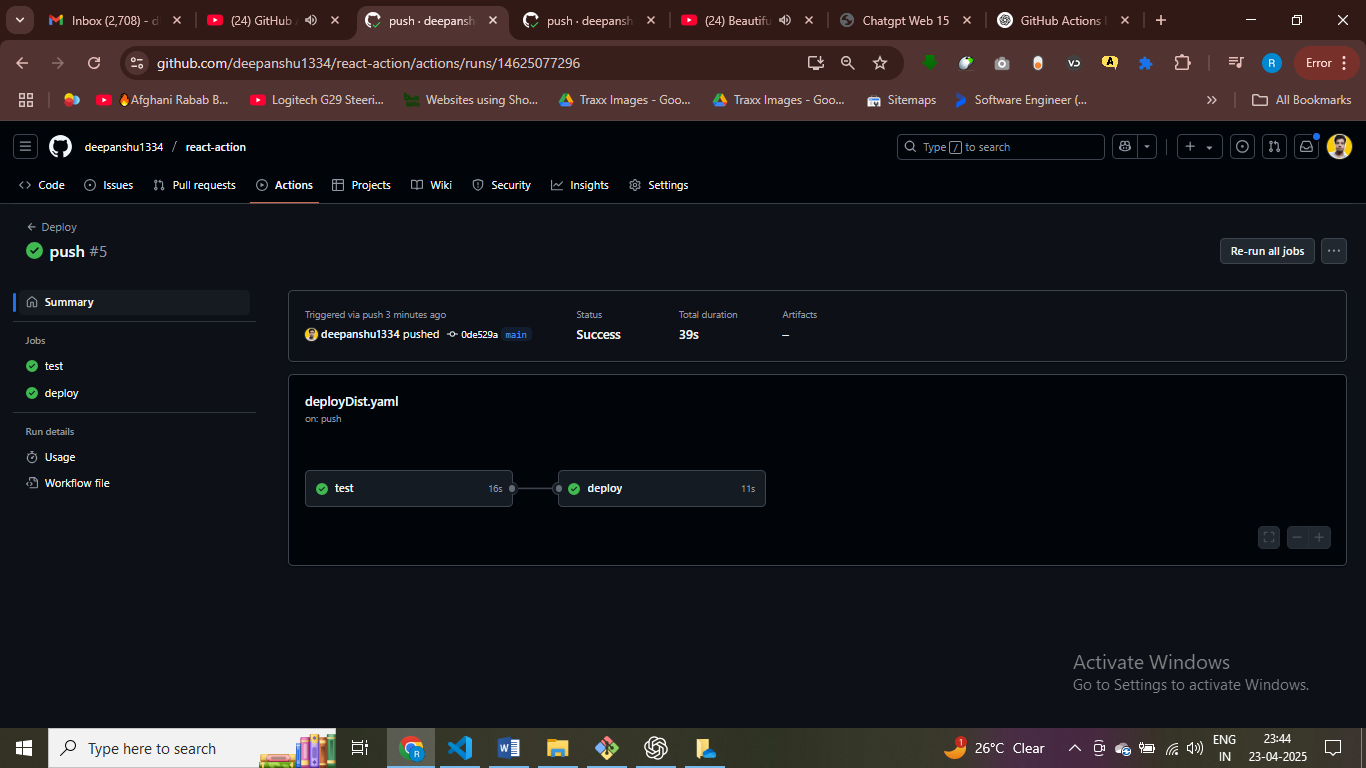
      - name: Build Project

        run: npm run build

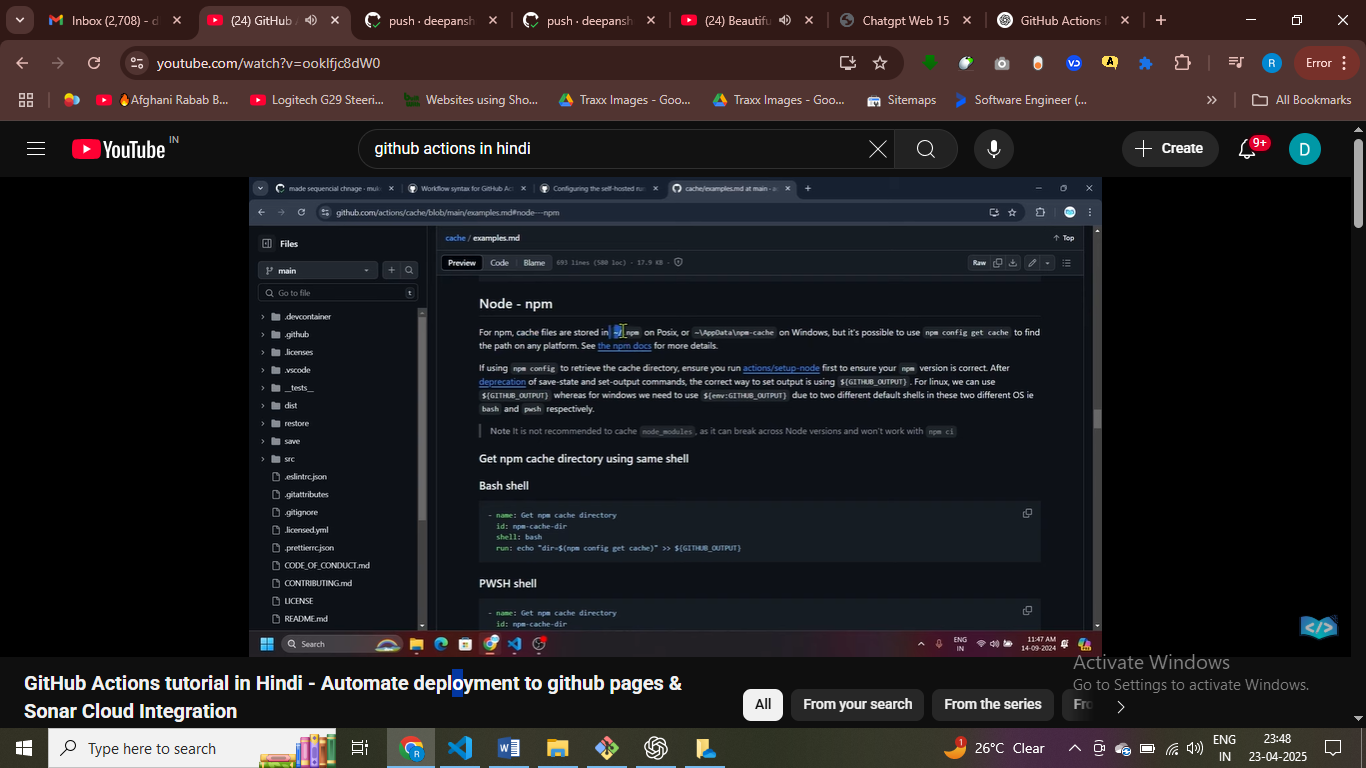
      - name: Deploy

        run: echo "Deploy scucesffule"

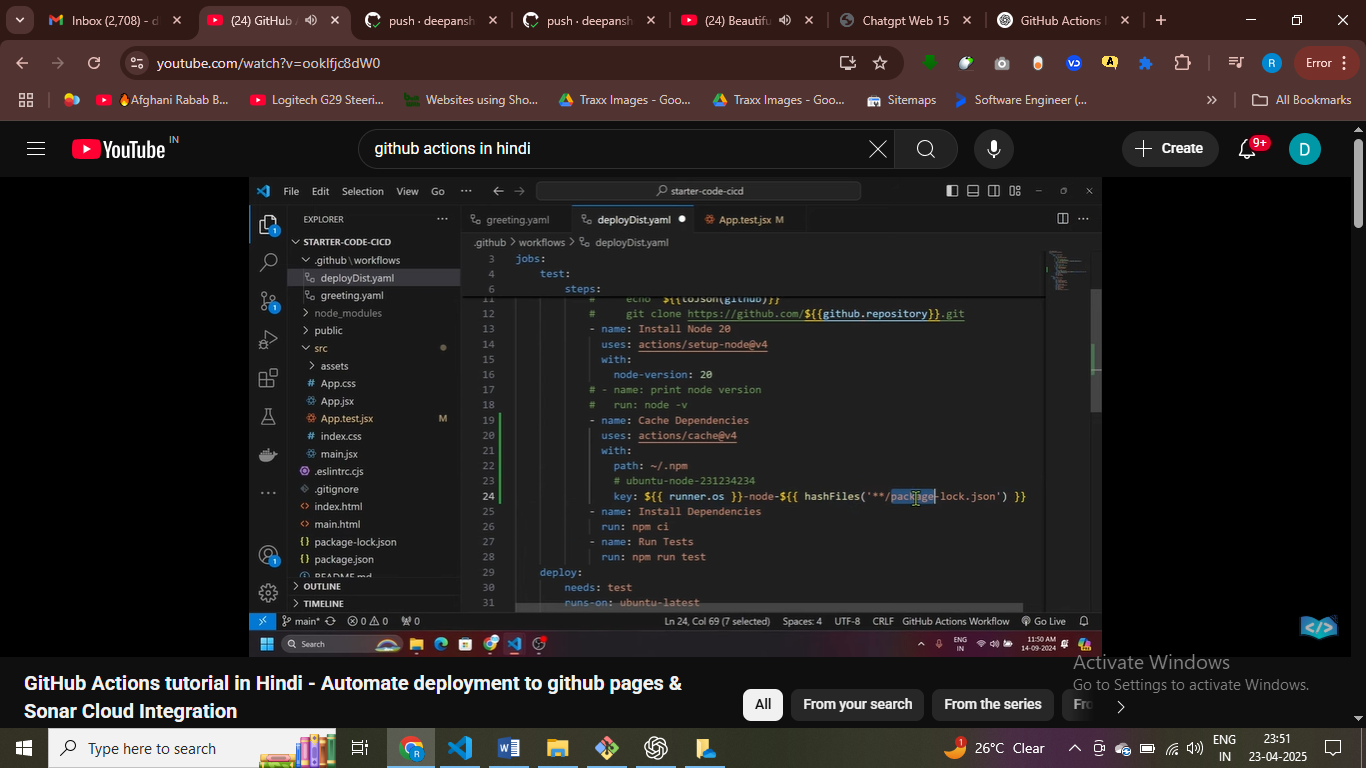
need:test – it ensure first test complted then deploy started



As both the test and deploy use npm install so we can use cache



Action ka use kar rha hai toh define uses



# we are creting a key that will install depency

name: Deploy

on: [push, workflow\_dispatch]

jobs:

  test:

    runs-on: ubuntu-latest

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

      - name: Cache Dependencies

        uses: actions/cache@v4

        with:

          path: ~/.npm

          key: ${{ runner.os }}-node-${{ hashFiles('\*\*/package-lock.json') }}

      - name: Install dependencies

        run: npm ci

      - name: Run tests

        run: npm run test

      - name: ✅ Test job completed successfully

        run: echo "✅ Test phase completed successfully"

  deploy:

    runs-on: ubuntu-latest

    needs: test

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

      - name: Print Node.js version

        run: node -v

      - name: Install dependencies

        run: npm ci

      - name: Run tests

        run: npm run test

      - name: Build Project

        run: npm run build

      - name: ✅ Deployment Successful

        run: echo "🚀 Deployment completed successfully"

cahce version so that we don’t need ot install again the depency rather we can directly use the one that we created earlier

main branch main push kare tabhi hamara workflow trigger hone chaiye

on specify when it should trigger on push or pull

name: Deploy

on:

  push:

    branches:

      - main

      - 'feature/\*\*'

jobs:

  test:

    runs-on: ubuntu-latest

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

      - name: Cache Dependencies

        uses: actions/cache@v4

        with:

          path: ~/.npm

          key: ${{ runner.os }}-node-${{ hashFiles('\*\*/package-lock.json') }}

      - name: Install dependencies

        run: npm ci

      - name: Run tests

        run: npm run test

      - name: ✅ Test job completed successfully

        run: echo "✅ Test phase completed successfully"

  deploy:

    runs-on: ubuntu-latest

    needs: test

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

      - name: Print Node.js version

        run: node -v

      - name: Install dependencies

        run: npm ci

      - name: Run tests

        run: npm run test

      - name: Build Project

        run: npm run build

      - name: ✅ Deployment Successful

        run: echo "🚀 Deployment completed successfully"

trigger the workflow when we push on github

name: Deploy

on:

  push:

    branches:

      - main

      - 'feature/\*\*'

    paths-ignore:

          - '.github/workflows/\*'

          - README.md

jobs:

  test:

    runs-on: ubuntu-latest

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

      - name: Cache Dependencies

        uses: actions/cache@v4

        with:

          path: ~/.npm

          key: ${{ runner.os }}-node-${{ hashFiles('\*\*/package-lock.json') }}

      - name: Install dependencies

        run: npm ci

      - name: Run tests

        run: npm run test

      - name: ✅ Test job completed successfully

        run: echo "✅ Test phase completed successfully"

  deploy:

    runs-on: ubuntu-latest

    needs: test

    steps:

      - name: Checkout code

        uses: actions/checkout@v4

      - name: Install Node.js

        uses: actions/setup-node@v4

        with:

          node-version: 20

      - name: Print Node.js version

        run: node -v

      - name: Install dependencies

        run: npm ci

      - name: Run tests

        run: npm run test

      - name: Build Project

        run: npm run build

      - name: ✅ Deployment Successful

        run: echo "🚀 Deployment completed successfully"

= if we change on radme then don’t push ,

Don’t trigger the workflow

name: Deploy Dist

# on: [push,workflow\_dispatch]

on:

    push:

        branches:

            - main

            - 'feature/\*\*'

        paths-ignore:

            - '.github/workflows/\*'

            - README.md

    workflow\_dispatch:

jobs:

    test:

        runs-on: ubuntu-latest

        steps:

            - name: Get Code

              uses: actions/checkout@v4

            #   run: |

            #     echo "cloning repo.."

            #     echo "${{toJson(github)}}"

            #     git clone https://github.com/${{github.repository}}.git

            - name: Install Node 20

              uses: actions/setup-node@v4

              with:

                node-version: 20

            # - name: print node version

            #   run: node -v

            - name: Cache Dependencies

              uses: actions/cache@v4

              with:

                path: ~/.npm

                key: ${{ runner.os }}-node-${{ hashFiles('\*\*/package-lock.json') }}

            - name: Install Dependencies

              run: npm ci

            - name: Run Tests

              run: npm run test

    build:

        needs: test

        runs-on: ubuntu-latest

        steps:

          - name: Get Code

            uses: actions/checkout@v4

          - name: Install Node 20

            uses: actions/setup-node@v4

            with:

              node-version: 20

          - name: Cache Dependencies

            uses: actions/cache@v4

            with:

              path: ~/.npm

              key: ${{ runner.os }}-node-${{ hashFiles('\*\*/package-lock.json') }}

          - name: Install Dependencies

            run: npm ci

          - name: Build Project

            run: npm run build

          - name: Upload Dist Artifact

            uses: actions/upload-artifact@v4

            with:

                name: dist-files

                path: dist

    deploy:

        needs: build

        runs-on: ubuntu-latest

        steps:

            - name: Download Dist Artifact

              uses: actions/download-artifact@v4

              with:

                name: dist-files

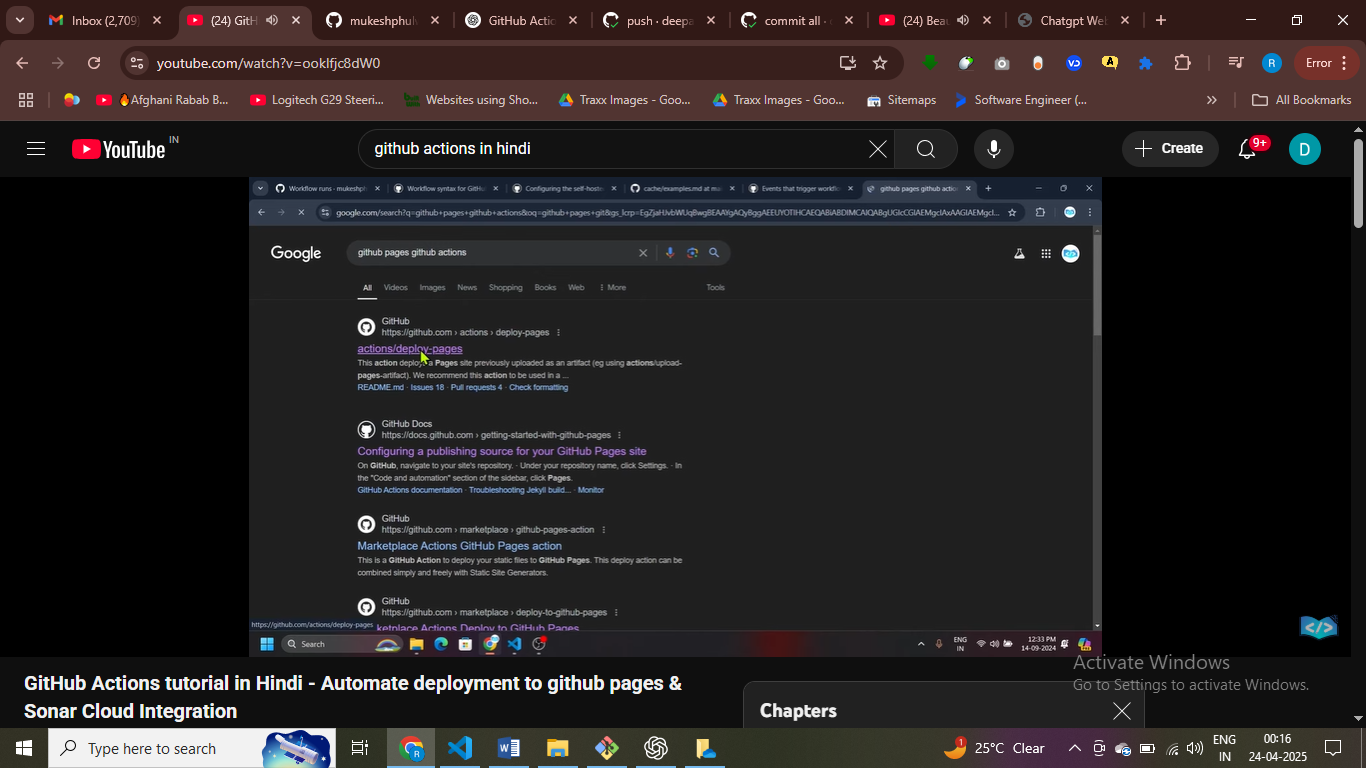
                path: dist

            - name: Deploy

              run: echo "Deploy successfull"

here we are building the project and deploying our appn

artifact folder generate hpga aur usko deploy kar raha hai



https://www.youtube.com/watch?v=ookIfjc8dW0

