U18ITE0228T – PRINCIPLE OF DEVOPS

Experiment No. 5

GitHub CI/CD (Continuous Integration and Deployment) Pipeline

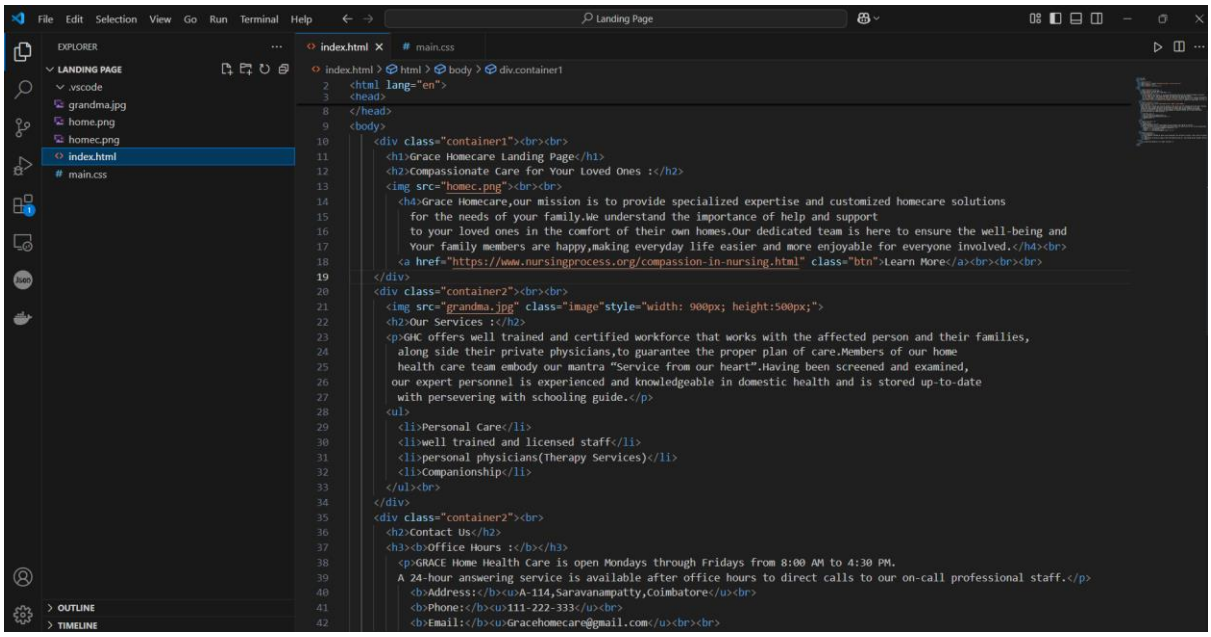
AIM:

The aim of this guide is to help you set up a simple CI/CD pipeline using GitHub Actions. By the end, you will be able to:

- Automate testing and deploying code changes.
- Add CI/CD to your GitHub project.
- Understand how a GitHub Actions workflow works.

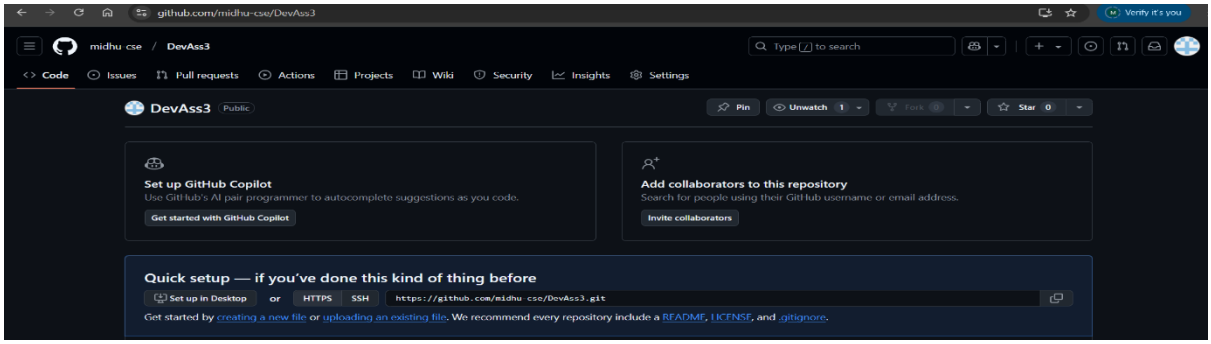
STEPS:

1. **Create an index.html and main.css file.**

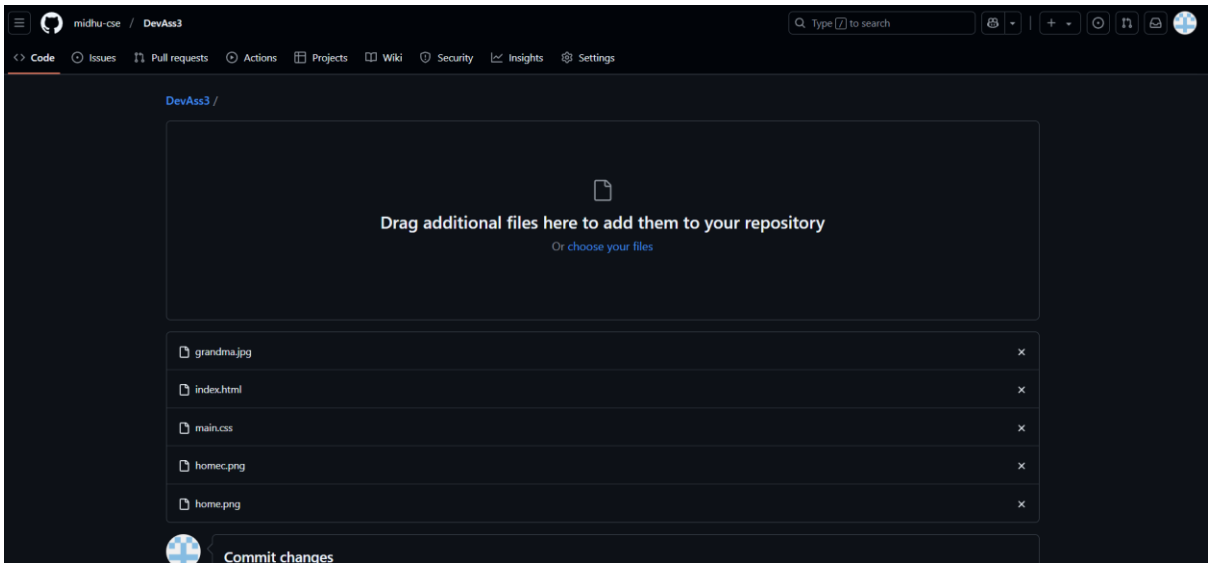


2, Initialize a GitHub Repository

- Create a new repository on GitHub (repository name is DevAss3)and push your project code(index.html,main.css)

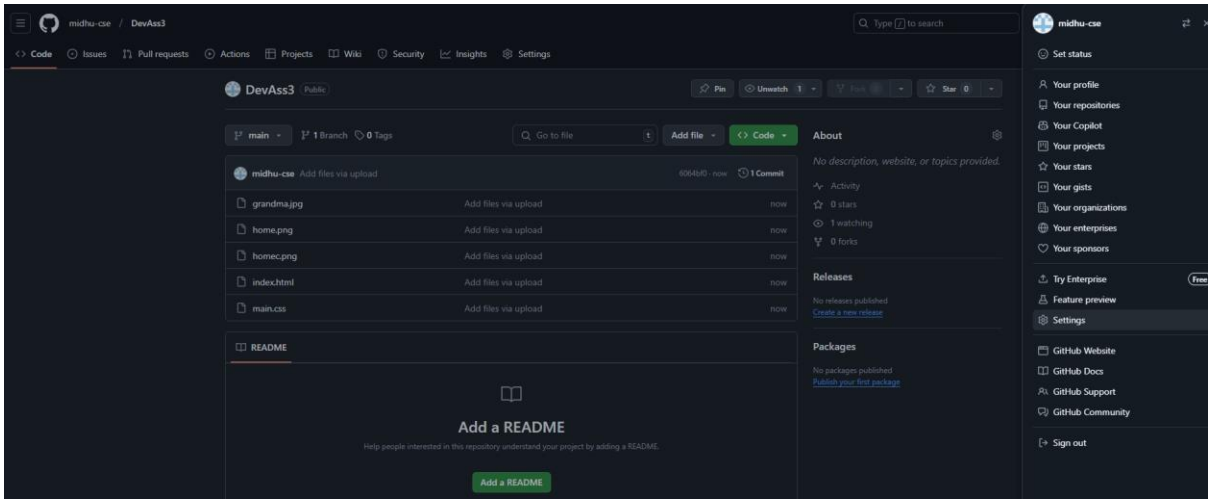


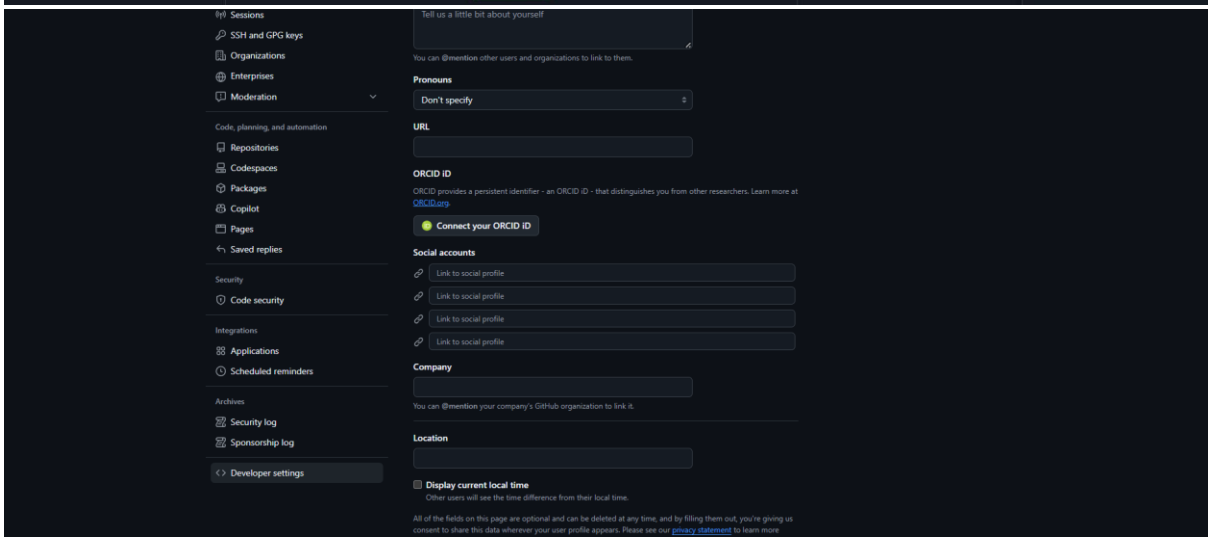
3.Add the index.html,main.css file in the repository.

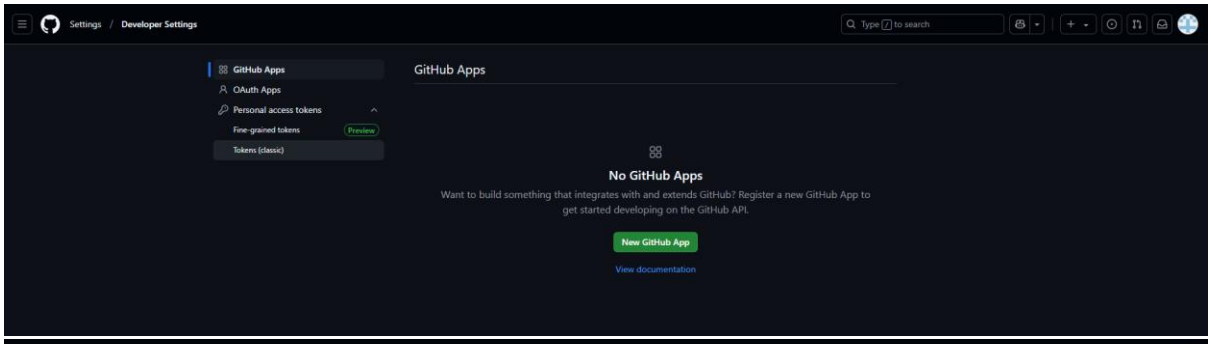


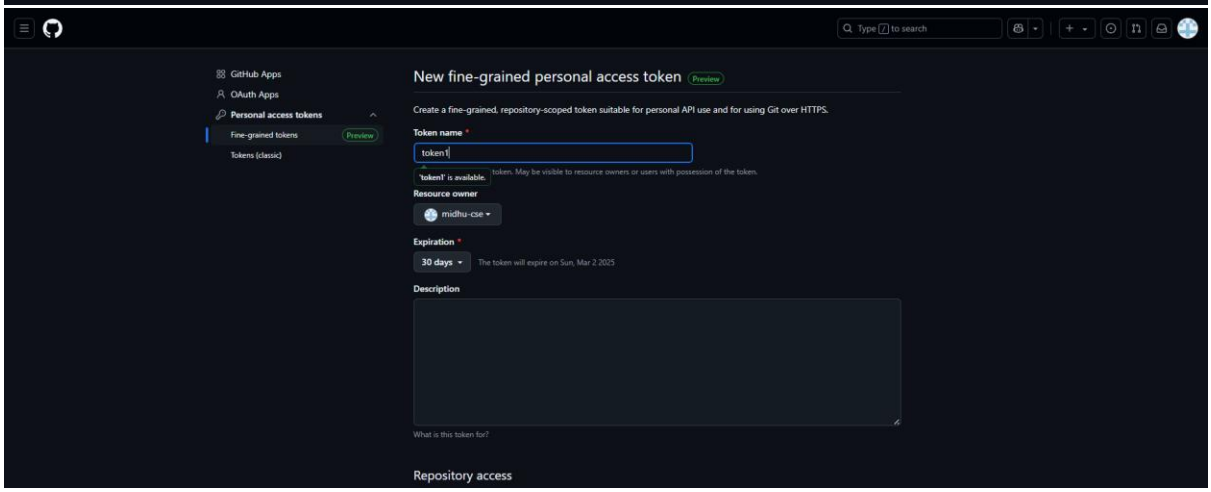
4.Generate a GitHub Access Token

- Navigate to Settings > Developer settings > Personal access tokens and generate a token with repo permissions.





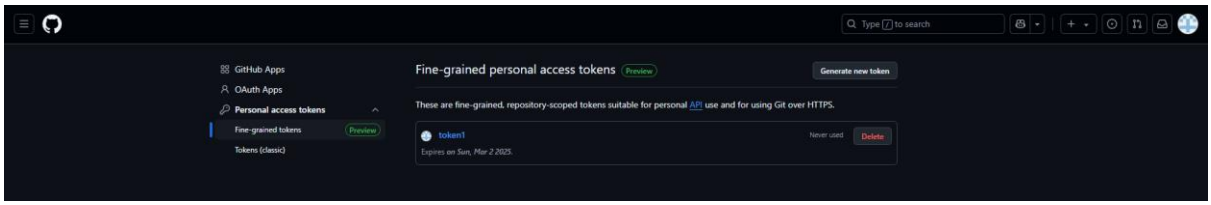




5.Add the Token to GitHub Secrets

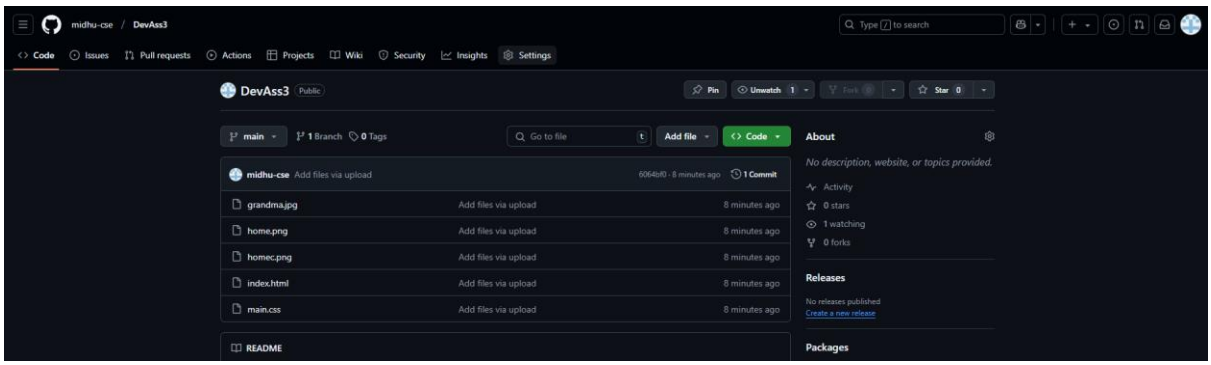
- Go to Settings > Secrets and variables > Actions in your repository.
- Click New repository secret, name it (KEY), and paste the token.

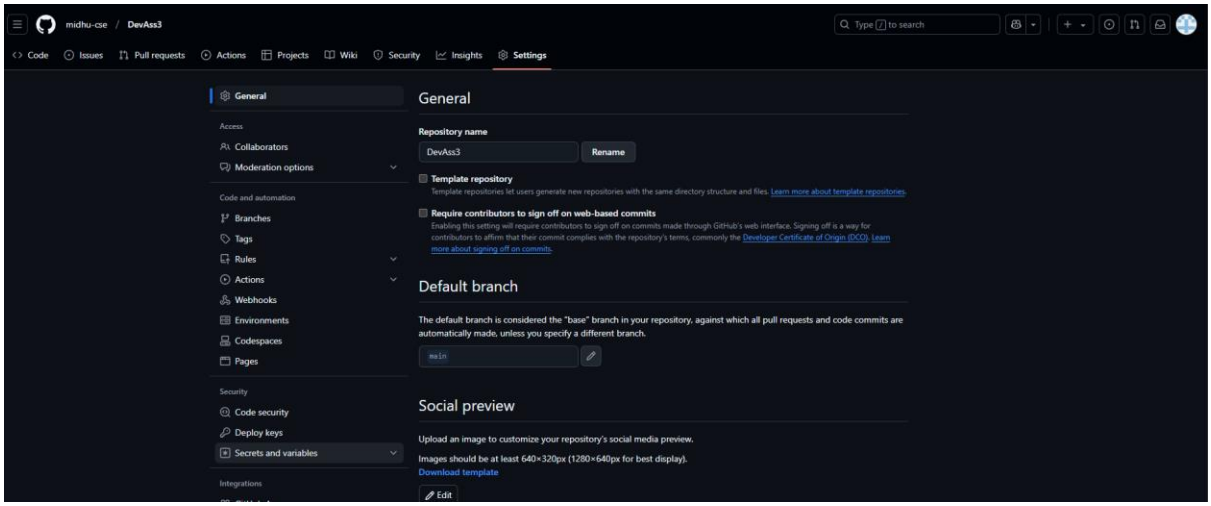
6.Copy the token.



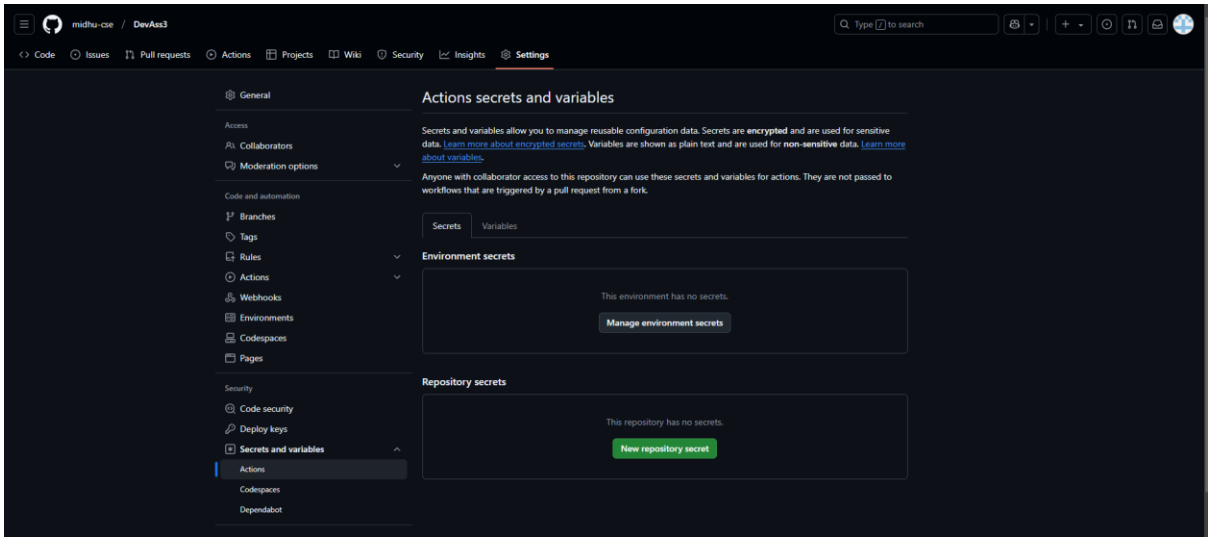
7.Navigate to the Settings of the Repository.

8.Under the Security, Click on the Secrets and Variables.

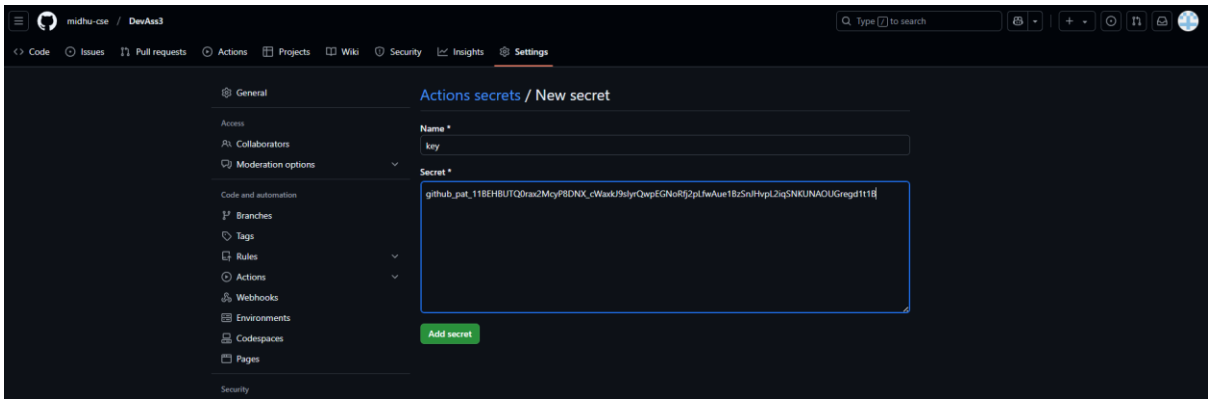


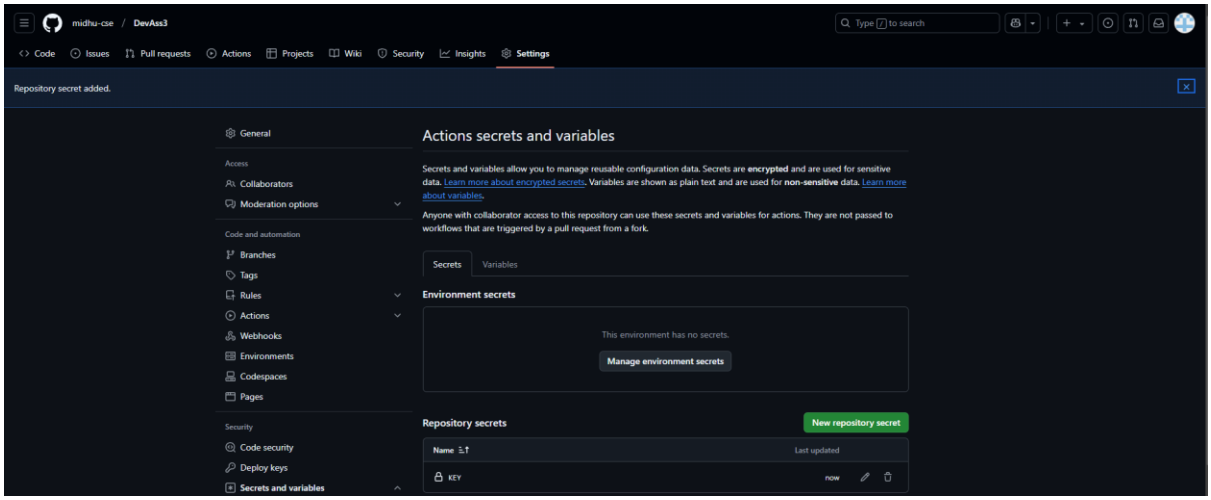


9.Click on the "New repository secret".



10.Name the secret key.





11.Create GitHub Actions Workflow Directory

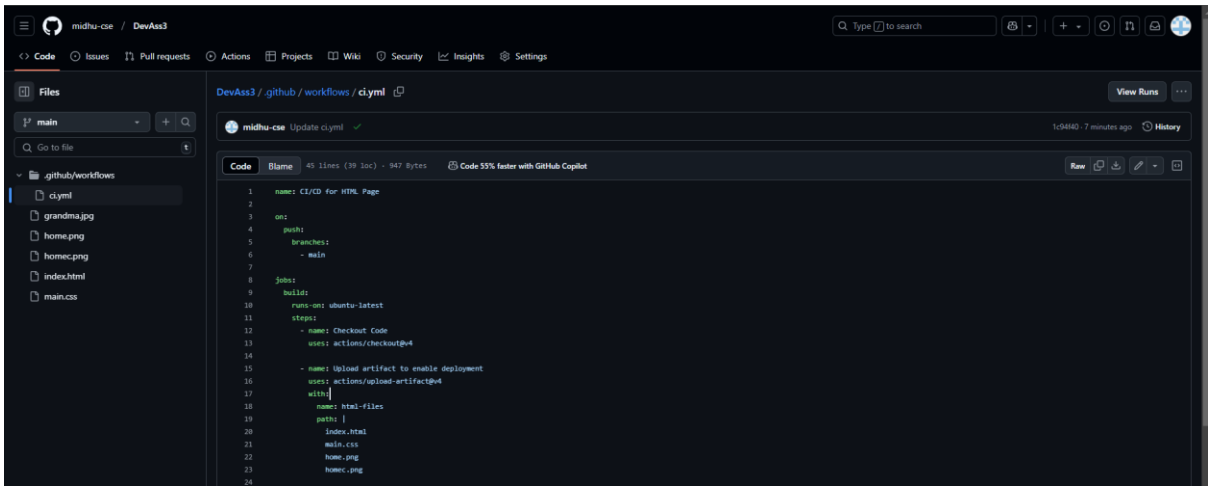
- In the project root, create a .github/workflows directory.

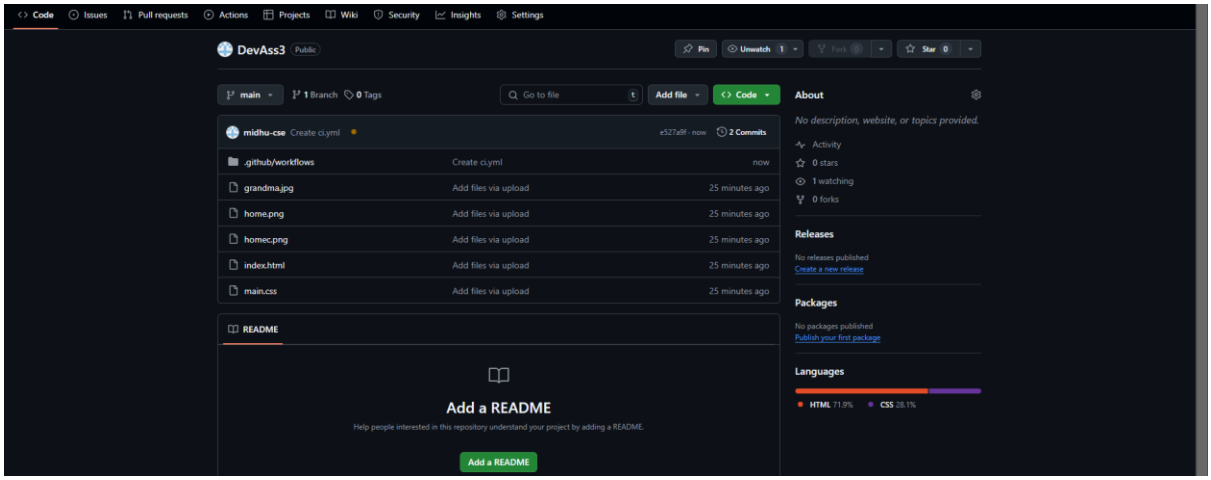
Create the directory as follows:

.github/workflows/ci.yml:

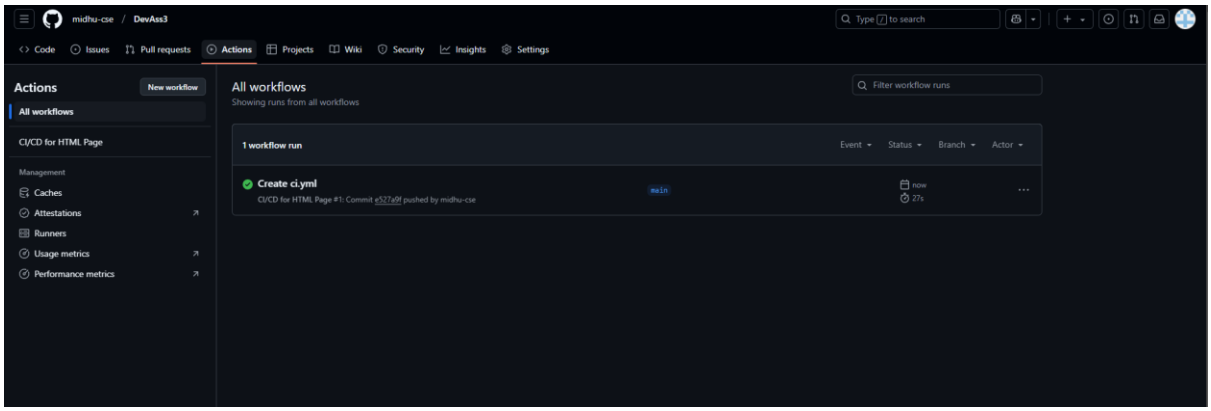
```
name: CI/CD for HTML Page

on:
 push:
  branches:
   - main
jobs:
 build:
  runs-on: ubuntu-latest
  steps:
   - name: Checkout Code
     uses: actions/checkout@v4


   - name: Upload artifact to enable deployment
     uses: actions/upload-artifact@v4
     with:
      name: html-files
      path: |
       index.html
```

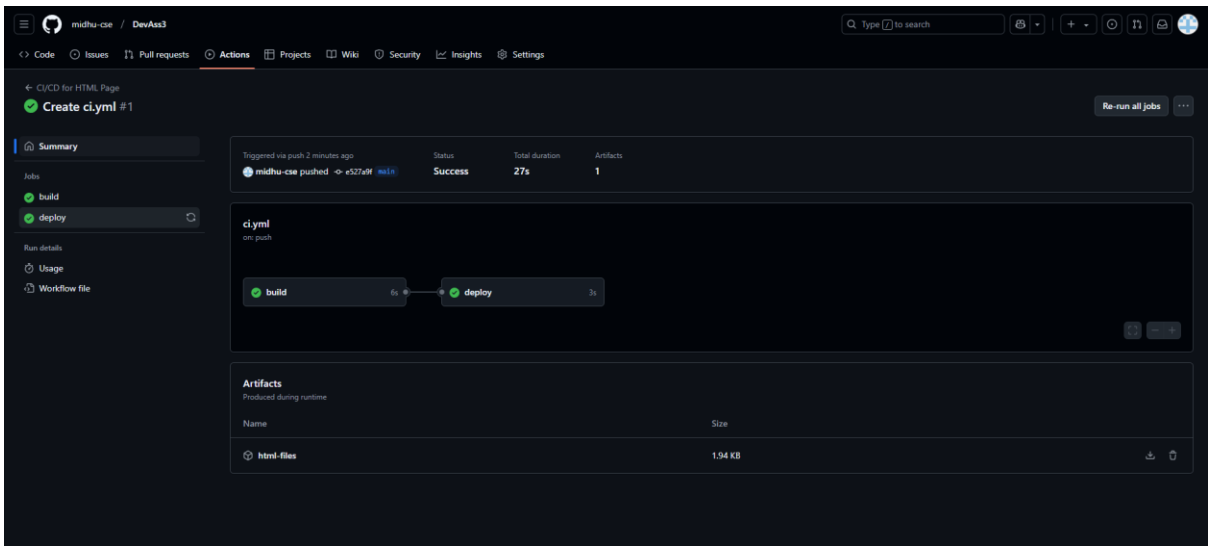
```
        main.css
        home.png
        homec.png
deploy:
  needs: build
  runs-on: ubuntu-latest
  permissions:
    contents: write
  steps:
    - name: Checkout Repository
      uses: actions/checkout@v4
    - name: Download artifact
      uses: actions/download-artifact@v4
      with:
        name: html-files
        path: ./html
    - name: Deploy to GitHub Pages
      uses: peaceiris/actions-gh-pages@v4
      with:
        github_token: ${{ secrets.GITHUB_TOKEN }}
        publish_branch: gh-pages
        publish_dir: ./html
```





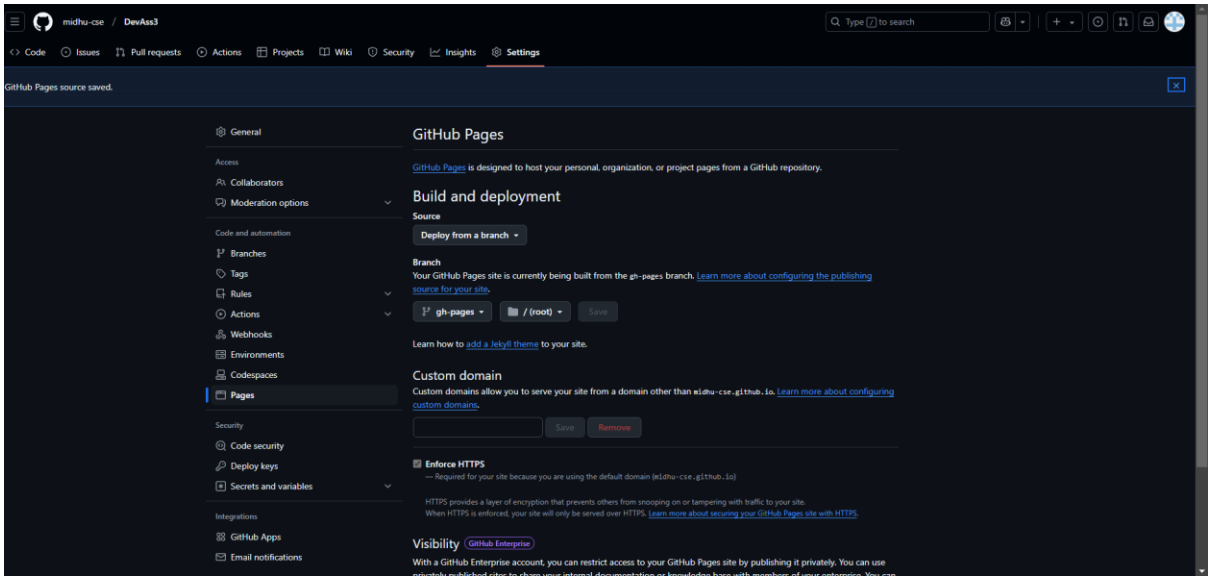
12.Go to the Actions tab in your repository to track the CI/CD process.

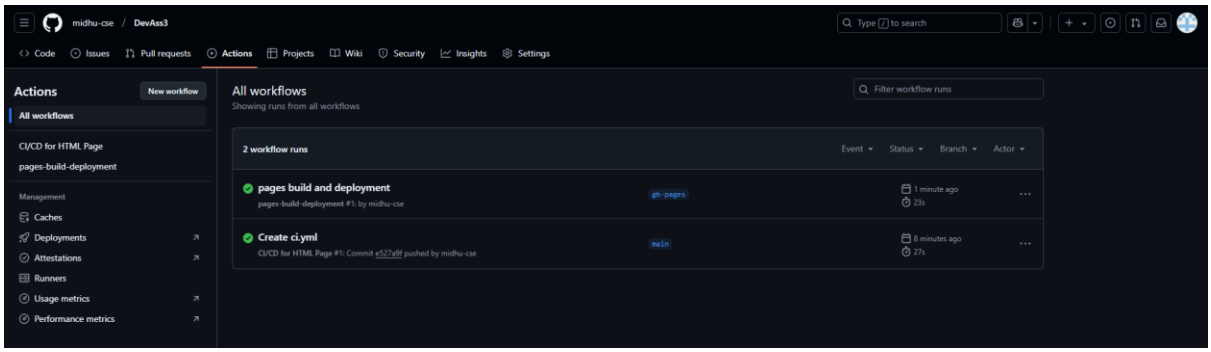


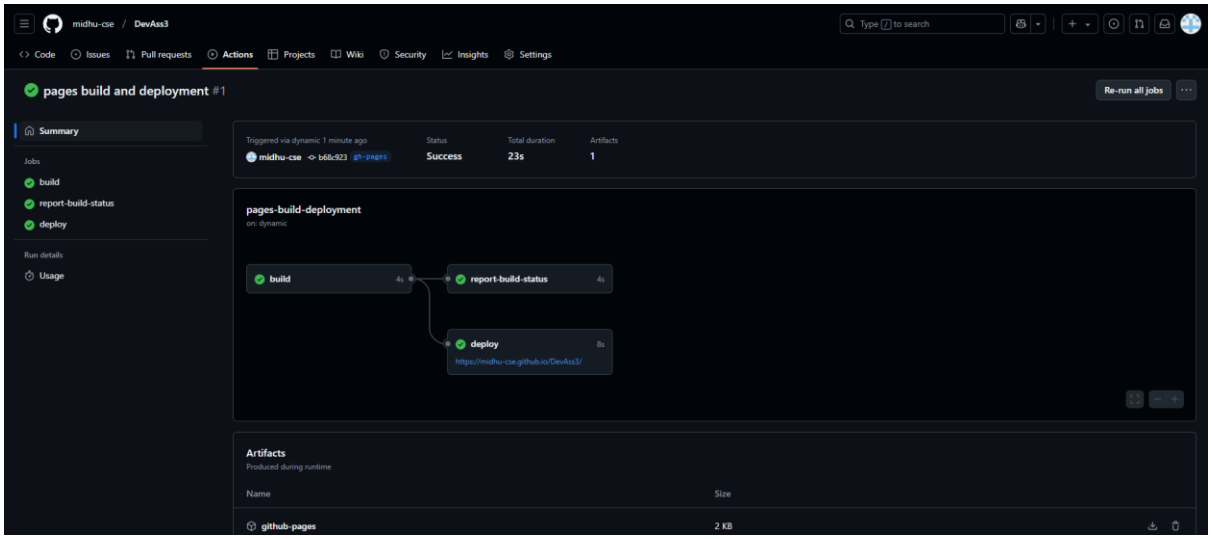


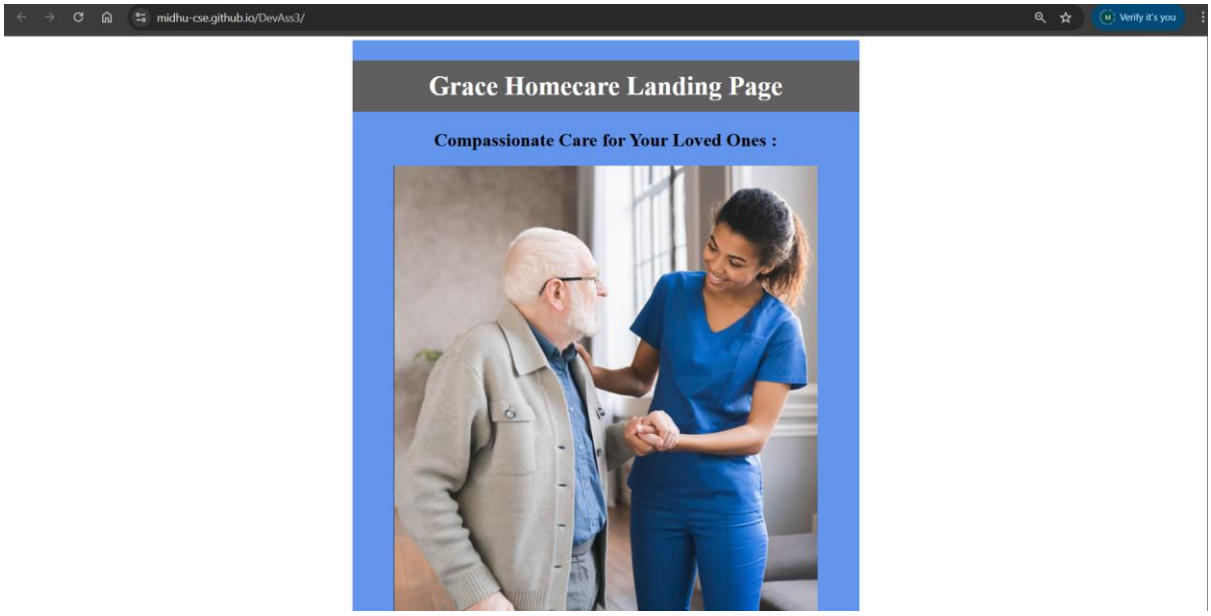
13.Click on the Actions.

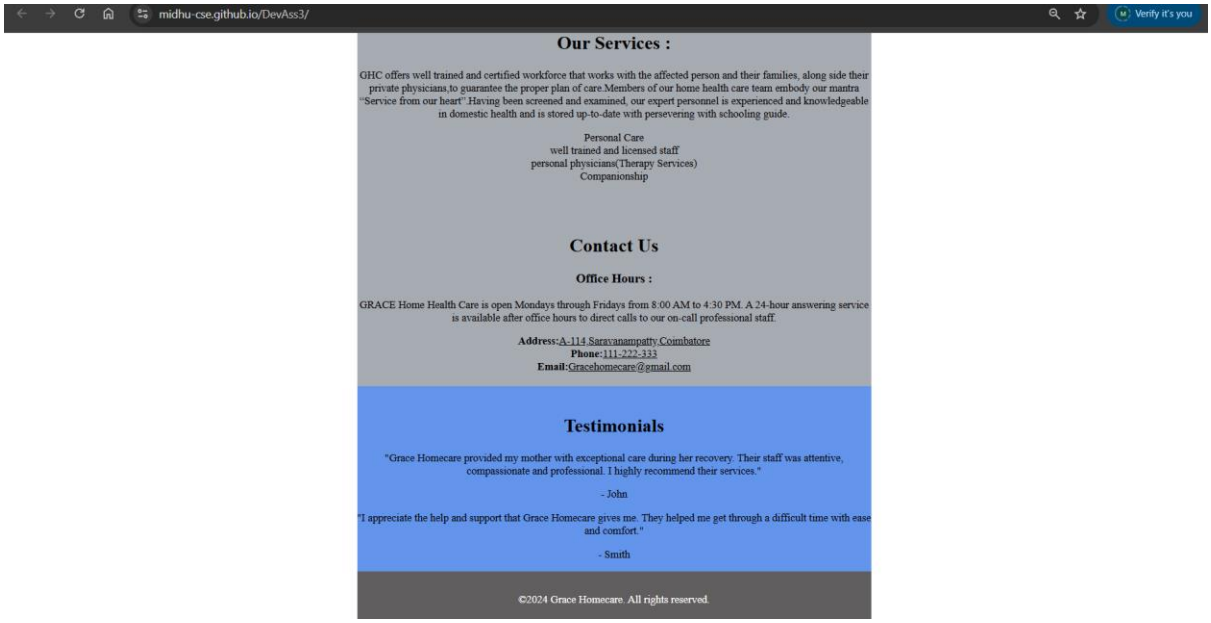
14.When you click on the link generated in “deploy”, the content of index.html file is shown.


midhu-cse / DevAss3
Code Issues Pull requests Actions Projects Wiki Security Insights Settings
GitHub Pages source saved.
General
Access
Collaborators
Moderation options
Code and automation
Branches
Tags
Rules
Actions
Webhooks
Environments
Codespaces
Pages
Security
Code security
Deploy keys
Secrets and variables
Integrations
GitHub Apps
Email notifications
GitHub Pages
GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.
Build and deployment
Source
Deploy from a branch
Branch
Your GitHub Pages site is currently being built from the gh-pages branch. Learn more about configuring the publishing source for your site.
gh-pages
/ (root)
Save
Learn how to add a Jekyll theme to your site.
Custom domain
Custom domains allow you to serve your site from a domain other than midhu-cse.github.io. Learn more about configuring custom domains.
Save
Remove
Enforce HTTPS
— Required for your site because you are using the default domain (midhu-cse.github.io)
HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site.
When HTTPS is enforced, your site will only be served over HTTPS. Learn more about securing your GitHub Pages site with HTTPS.
Visibility
GitHub Enterprise
With a GitHub Enterprise account, you can restrict access to your GitHub Pages site by publishing it privately. You can use


midhu-cse / DevAss3
Code Issues Pull requests Actions Projects Wiki Security Insights Settings
Actions
New workflow
All workflows
CI/CD for HTML Page
pages-build-deployment
Management
Caches
Deployments
Attestations
Runners
Usage metrics
Performance metrics
All workflows
Showing runs from all workflows
Filter workflow runs
2 workflow runs
Event
Status
Branch
Actor
pages build and deployment
pages-build-deployment #1: by midhu-cse
gh-pages
1 minute ago
23s
Create ci.yml
CI/CD for HTML Page #1: Commit e527a9f pushed by midhu-cse
main
8 minutes ago
27s


midhu-cse / DevAss3
Code Issues Pull requests Actions Projects Wiki Security Insights Settings
pages build and deployment #1
Re-run all jobs
Summary
Jobs
build
report-build-status
deploy
Run details
Usage
Triggered via dynamic 1 minute ago
midhu-cse b68c923 gh-pages
Status
Success
Total duration
23s
Artifacts
1
pages-build-deployment
on: dynamic
build
4s
report-build-status
4s
deploy
8s
https://midhu-cse.github.io/DevAss3/
Artifacts
Produced during runtime
Name
Size
github-pages
2 KB





Conclusion:

By following these steps, I have created a CI/CD pipeline using GitHub Actions. This pipeline automates testing and deployment, making software delivery faster and more reliable. GitHub Actions offers a flexible platform to integrate CI/CD with various tools and services.