**U18ITE0228T – PRINCIPLE OF DEVOPS**

**Experiment No. 5**

**GitHub CI/CD (Continuous Integration and Deployment) Pipeline**

**AIM:**

* This guide aims to help you build a robust **CI/CD pipeline** using **GitHub Actions**.
* By the end of this guide, you will be able to:
  + **Automate** testing and deployment of code changes.
  + **Integrate** CI/CD into your **GitHub repository**.
  + **Understand** the key components of a **GitHub Actions workflow**.

**EXPERIMENTAL STEPS:**

1. **Create an index.html file.**

**A screenshot of a computer program

Description automatically generated**

**2, Initialize a GitHub Repository**

* **Create a new repository on GitHub and push your project code.**

**A screenshot of a computer

Description automatically generated**

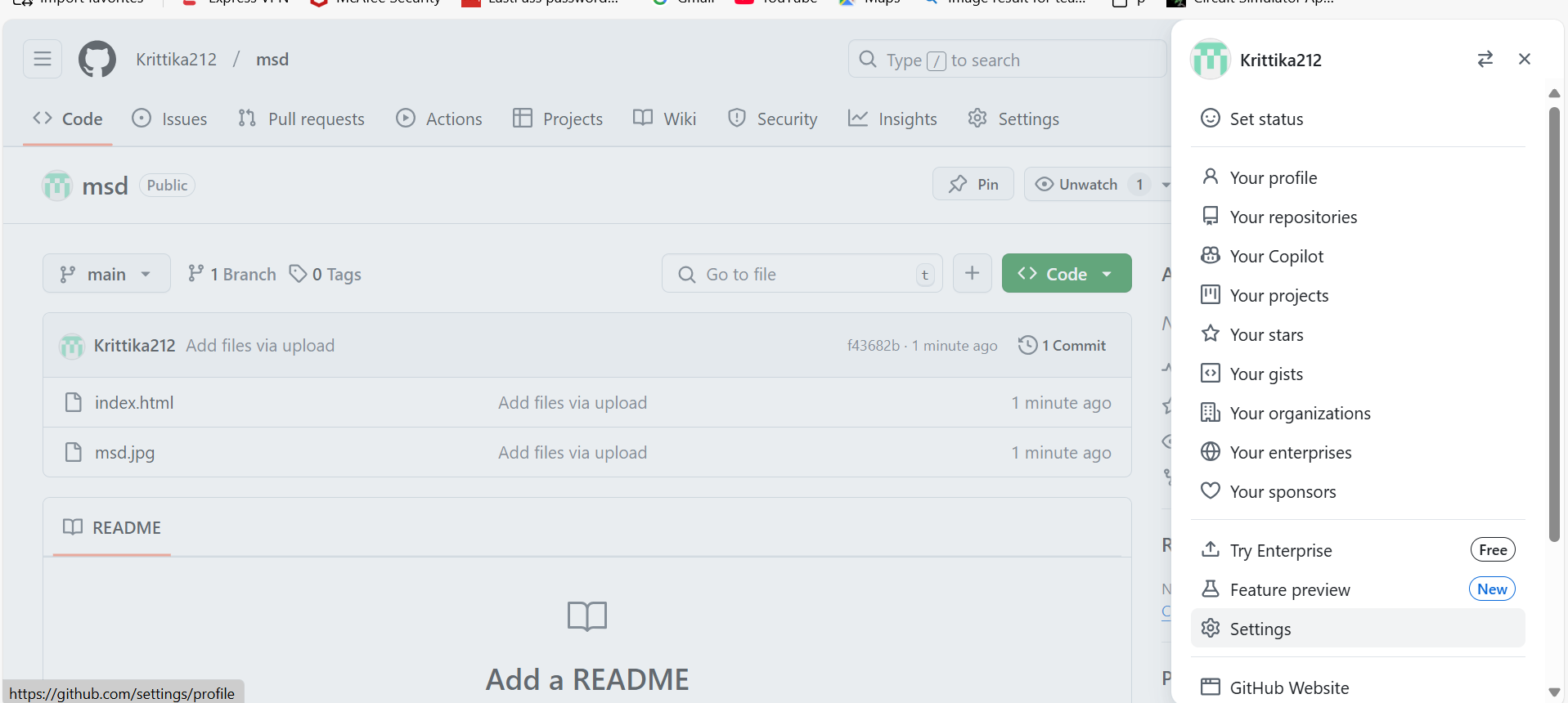
**3.Add the index.html file in the repository.**

**A screenshot of a computer

Description automatically generated**

**4.Generate a GitHub Access Token**

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

****

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**5.Add the Token to GitHub Secrets**

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

**A screenshot of a computer

Description automatically generated**

**6.Copy the token.**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**7.Navigate to the Settings of the Repository.**

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

**A screenshot of a computer

Description automatically generated**

**9.Click on the “New repository secret”.**

**A screenshot of a computer

Description automatically generated**

**10.Name the secret key.**

**A screenshot of a computer

Description automatically generated**

**A close-up of a person

Description automatically generated**

**11.Create GitHub Actions Workflow Directory**

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

**Create the directory as follows(after cloning the git repo in the pc)**

**.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

deploy:

needs: build

runs-on: ubuntu-latest

permissions:

contents: write # Required for pushing to gh-pages

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 # Explicitly deploy to gh-pages

publish\_dir: ./html

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

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

**A screenshot of a computer

Description automatically generated**

**13.Click on the Actions.**

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

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Conclusion**

By following these steps, you have successfully built a CI/CD pipeline using GitHub Actions. This pipeline automates the process of testing and deploying your code, ensuring faster and more reliable software delivery. GitHub Actions provides a flexible and powerful platform for CI/CD, enabling you to integrate with various tools and services.