Placement Empowerment Program

Cloud Computing and DevOps Centre

Deploy your static website using Github Pages:

Host your local Git repository's static website directly using Github pages

Name: CHANDRU S

Department: INFORMATION TECHNOLOGY

St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY

AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

Introduction

GitHub Pages is a static site hosting service that enables developers to publish projects directly from a GitHub repository. It provides a free and efficient way to showcase work, create personal websites, and host documentation with ease.

Overview

This project showcases the deployment of a static website using GitHub Pages. Beginning with setting up a GitHub repository, it walks through the essential steps to host a functional static site, including initializing a Git repository, pushing files to GitHub, and configuring GitHub Pages for deployment.

Key Features of GitHub Pages:

- **Free Hosting** – Available for public repositories at no cost.
- **Static File Support** – Hosts HTML, CSS, and JavaScript files effortlessly.
- **Seamless Git Integration** – Easily manage and update your site with Git version control.

Objectives

1. Explore the fundamentals of GitHub Pages and its deployment process.
2. Understand the significance of static website hosting and its applications.
3. Develop hands-on experience with Git and GitHub for version control and hosting.
4. Successfully deploy and publish a static website for public access.

Importance of Hosting with GitHub Pages

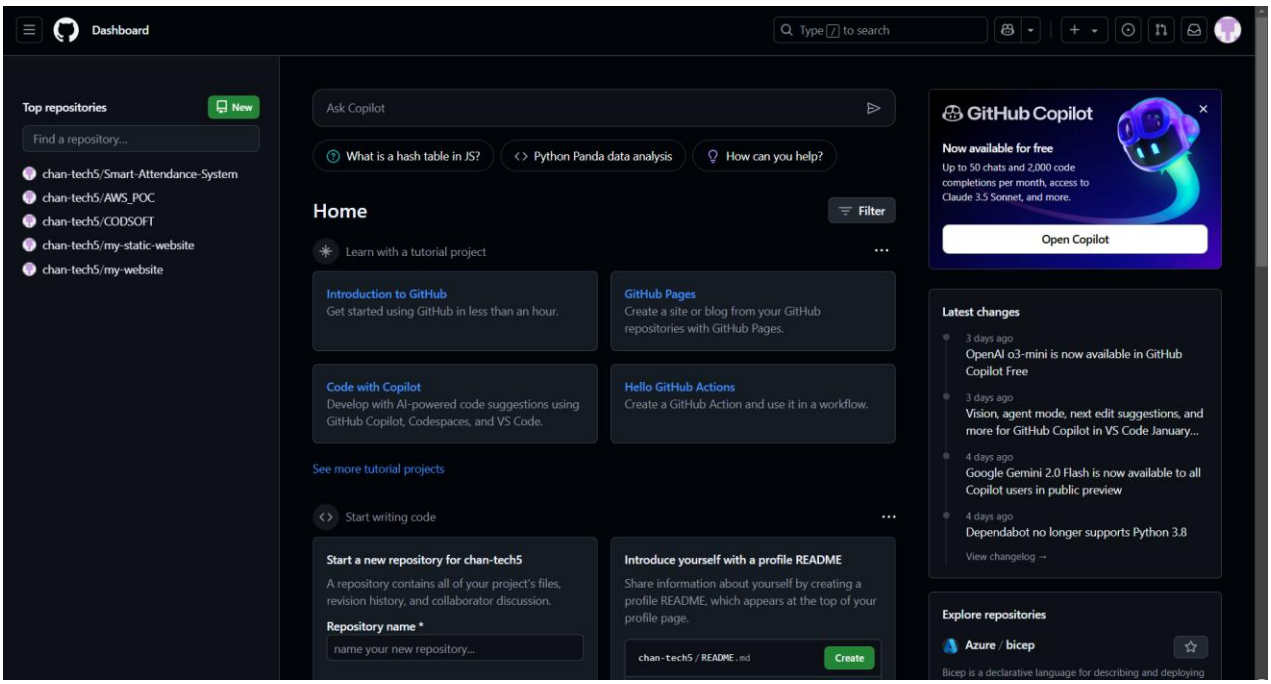
- **Cost-Effective –** Free for public repositories, making it an excellent choice for students and developers.
- **Built-in Version Control** – Seamlessly integrates with GitHub, allowing easy updates and collaborative development.
- **Increased Visibility –** Ideal for showcasing portfolios, projects, and documentation.
- **User-Friendly –** Requires minimal setup compared to other hosting solutions.
- **Custom Domains –** Supports custom domain configuration for a more professional online presence.

Step-by-Step Overview

Step 1:

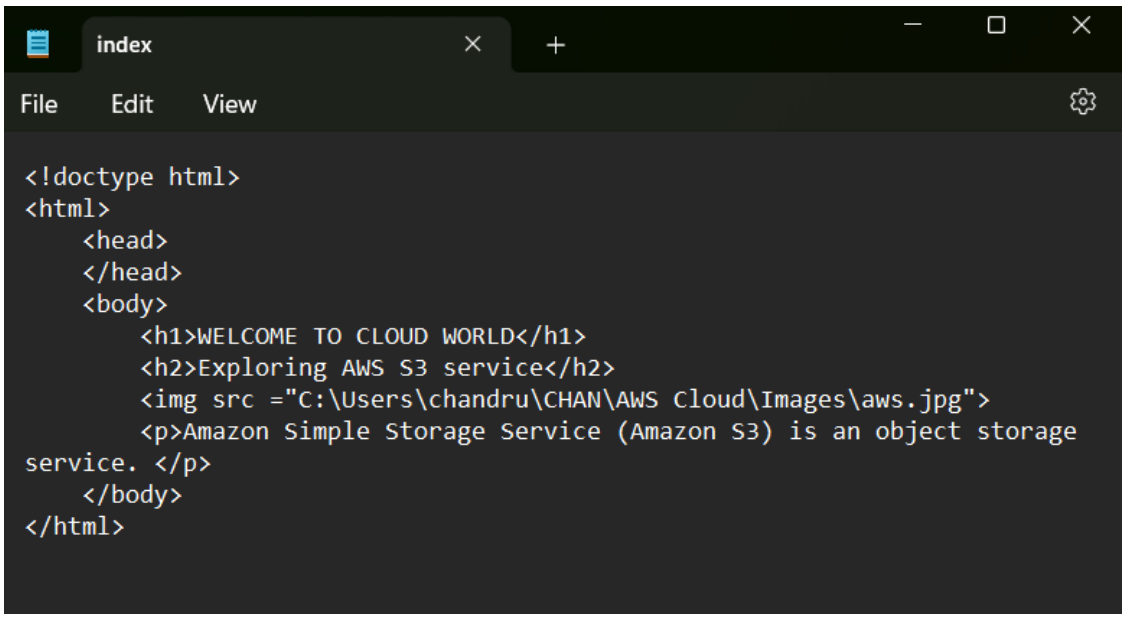
Create a New Repository:

- Log in to GitHub and click the green **"New"** button on the top-right of your homepage.
- Enter a repository name (e.g., my-static-website).
- Keep the default settings and click **"Create repository"**.



Step 2:

- Create a local folder (e.g., my-static-website) to store your website files.
- Inside the folder, create an index.html file with basic HTML content.



Step 3:

- Open Command Prompt or Terminal.
- Use the cd command to navigate to your project folder.

```
C:\Users\chandru>cd C:\Users\chandru\Downloads\poc7\my-static-website
```

Step 4:

Run the following command to initialize Git in your project folder:
git init

```
C:\Users\chandru\Downloads\poc7\my-static-website>git init
Initialized empty Git repository in C:/Users/chandru/Downloads/poc7/my-static-website/.git/
```

Step 5:

Stage your files for commit:

git add .

```
C:\Users\chandru\Downloads\poc7\my-static-website>git add .
```

Step 6:

Save the changes with a commit message:
git commit -m "Initial commit"

```
C:\Users\chandru\Downloads\poc7\my-static-website>git commit -m "Initial commit"
[master (root-commit) 06e3d5e] Initial commit
 1 file changed, 11 insertions(+)
 create mode 100644 index.html
```

Step 7:

- Go to your GitHub repository and copy the repository URL.
- In your terminal, link your local project to the GitHub repository:
  git remote add origin <your-repository-url>

```
C:\Users\chandru\Downloads\poc7\my-static-website>git remote add origin https://github.com/chan-tech5/my-static-website.git
```

Step 8:

Upload your project files to GitHub:

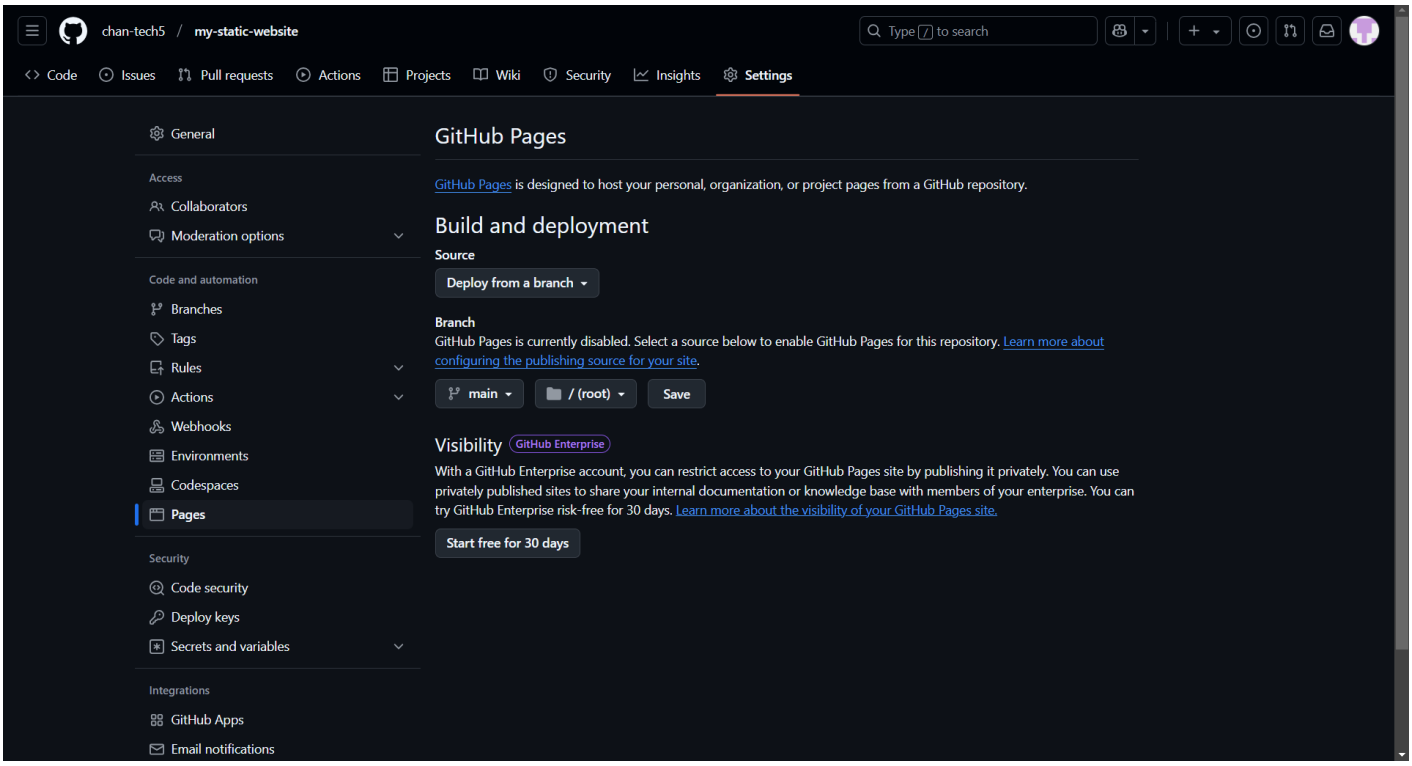
```
C:\Users\chandru\Downloads\poc7\my-static-website>git branch -M main

C:\Users\chandru\Downloads\poc7\my-static-website>git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 429 bytes | 429.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/chan-tech5/my-static-website.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Step 9:

Enable GitHub Pages

- Go to your repository on GitHub.
- Click on the **Settings** tab.
- Scroll down to the **Pages** section (under "Code and automation").
- Under **Source**, select:
  - **Branch:** main
  - **Folder:** / (root)
- Click **Save**.

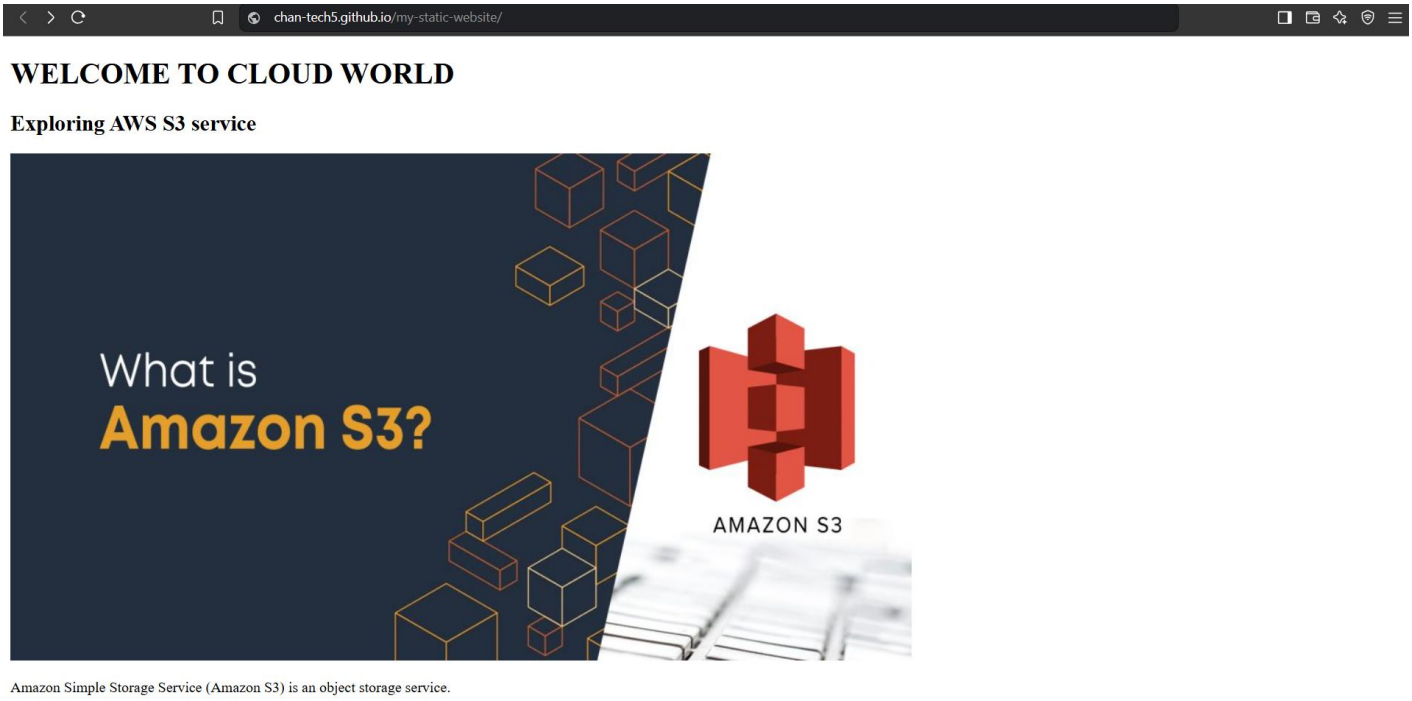


Step 10:

Access Your Website

- Wait a few minutes for deployment.
- Visit your website at:

  https://<your-username>.github.io/<your-repository>



Outcome

By completing this Proof of Concept (PoC), you will:

1. Successfully create and configure a GitHub repository.
2. Initialize a local Git repository and link it to GitHub.
3. Upload static website files (HTML, CSS, JavaScript).
4. Enable GitHub Pages for hosting.
5. Access your live website via a GitHub Pages URL.
6. Gain hands-on experience with essential Git commands (git init, git add, git commit, git remote add, git push).
7. Understand the process of hosting a static site for free with GitHub Pages.