St. JOSEPH'S
GROUP OF INSTITUTIONS
OMR, CHENNAI - 119

PLACEMENT EMPOWERMENT PROGRAM

CLOUD COMPUTING AND DEVOPS CENTRE

TASK 7 - Deploy your static website using Github Pages : Host your local Git repository's static website directly using Github pag

NAME - MAHASHREE U

DEPT - ADS

Introduction

GitHub Pages is a static site hosting service designed to publish your projects directly from a GitHub repository. It allows developers to showcase their work, create personal websites, or host documentation in an efficient, free, and straightforward way.

Overview

This project demonstrates how to deploy a static website using GitHub Pages. Starting with the basics of setting up a GitHub repository, we'll explore each step required to host a functional static website. This includes initializing a Git repository, pushing files to GitHub, and configuring GitHub Pages for deployment.

Importance of Hosting with GitHub Pages

1. Cost-effective: Free for public repositories, making it accessible for students and developers.

2. Version Control: Seamlessly integrates with GitHub, enabling easy updates and collaboration.

3. Visibility: A great way to showcase personal portfolios, projects, or documentation.

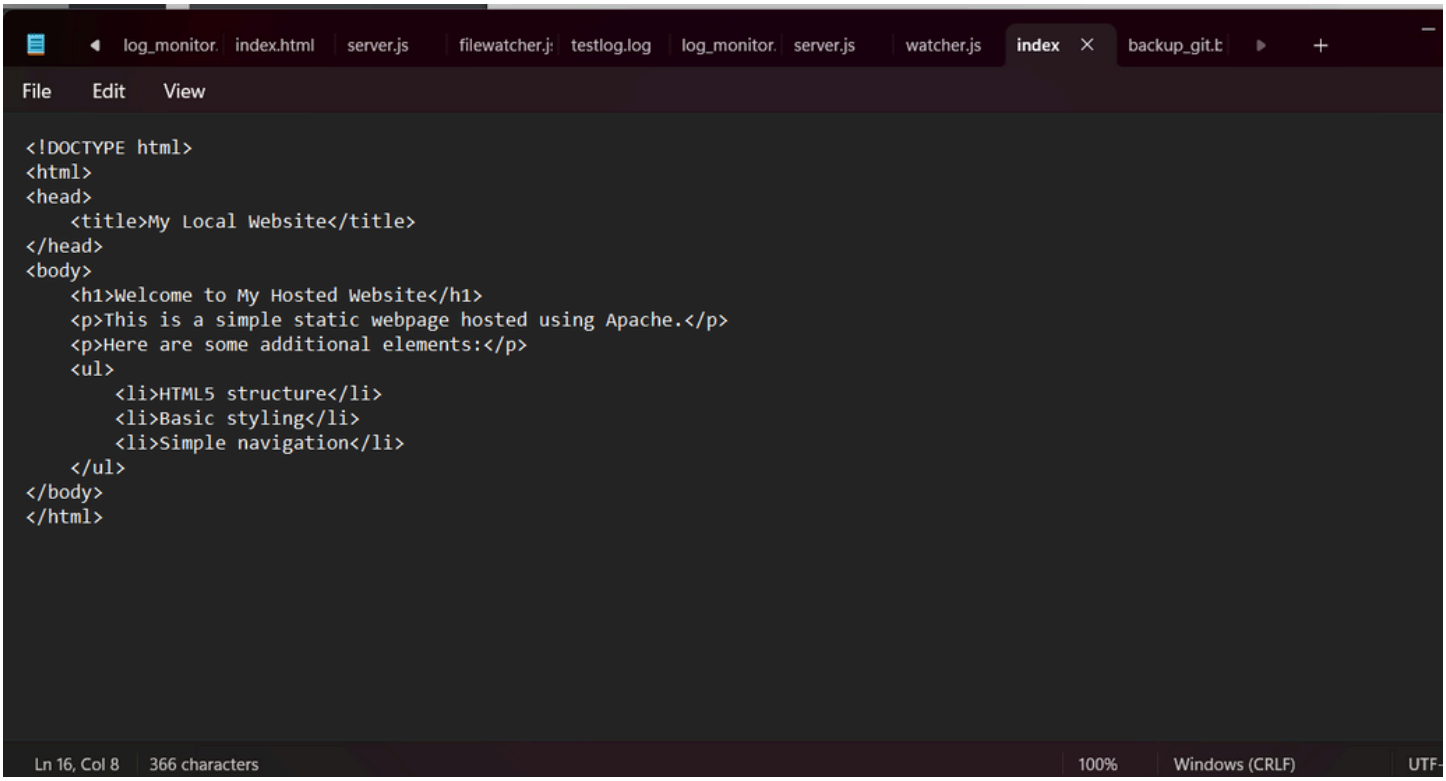
4. Ease of Use: Minimal setup required compared to other hosting platforms.

5. Custom Domains: Option to configure custom domains, enhancing the professional appeal of your website.

Step-by-Step Overview

Step 1: Create a New Repository: Once you're logged in, click the green "New" button on the topright of your GitHub homepage to create a new repository. Give your repository a name, for example, my-static-website. Leave the other settings as default, and click "Create repository"

Step 2: Create a folder (e.g., my-static-website) where you'll keep all your website files. Inside that folder, create the main file for your website, called index.html.



Step 3: Open Command Prompt and navigate to the folder where your index.html file is saved. Use the cd command to navigate.

```
C:\Users\Hi>cd C:\Users\Hi\Desktop\my-static-website
```

Step 4: Initialize a Git repository by running:

```
C:\Users\Hi\Desktop\my-static-website>git init
Initialized empty Git repository in C:/Users/Hi/Desktop/my-static-website/.git/
```

Step 5: Add your website files to the repository

```
C:\Users\Hi\Desktop\my-static-website>git add .
```

```
C:\Users\Hi\Desktop\my-static-website>git commit -m "Initial commit
[master (root-commit) 4a93af0] Initial commit
 1 file changed, 10 insertions(+)
 create mode 100644 index.html
```

Step 6: Push your files to GitHub

```
C:\Users\Hi\Desktop\my-static-website>git branch -M main

C:\Users\Hi\Desktop\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), 359 bytes | 25.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/SaravanaKrishnan16/my-static-website.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Step 7:

Enable GitHub Pages

1. Go to your repository on GitHub.

2. Click on the Settings tab (it's near the top, next to Code, Issues, etc.).

3. Scroll down to the Pages section (on the left menu, under "Code and automation").

4. Under Source, select: o Branch: main o Folder: / (root)

5. Click Save