# 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: Shakthi Lakshmanan V Department: CSE



### ****Introduction****

GitHub Pages is a free service that lets you host static websites directly from a GitHub repository. It's an easy way to showcase your projects, build a personal website, or host documentation without needing a separate hosting service.

### ****Overview****

This guide will walk you through deploying a static website using GitHub Pages. You'll learn how to:

* Set up a GitHub repository.
* Push your website files (HTML, CSS, JavaScript) to GitHub.
* Enable GitHub Pages for hosting your site.

#### ****Key Features of GitHub Pages:****

* Free hosting for public repositories.
* Supports static files (HTML, CSS, JavaScript).
* Simple integration with Git version control.

### ****Objectives****

1. Understand how GitHub Pages works and its benefits.
2. Learn how to deploy a static website step by step.
3. Gain practical experience with Git and GitHub.
4. Successfully publish your website and make it live.

### ****Why Use GitHub Pages for Hosting?****

1. **Free & Cost-Effective** – Ideal for students and developers with public projects.
2. **Version Control** – Easily track changes and collaborate with others using GitHub.
3. **Increased Visibility** – Showcase portfolios, projects, or documentation online.
4. **Easy to Use** – No complex setup, works seamlessly with GitHub.
5. **Custom Domains** – Option to add a personalized domain for a professional touch

# Step-by-Step Overview

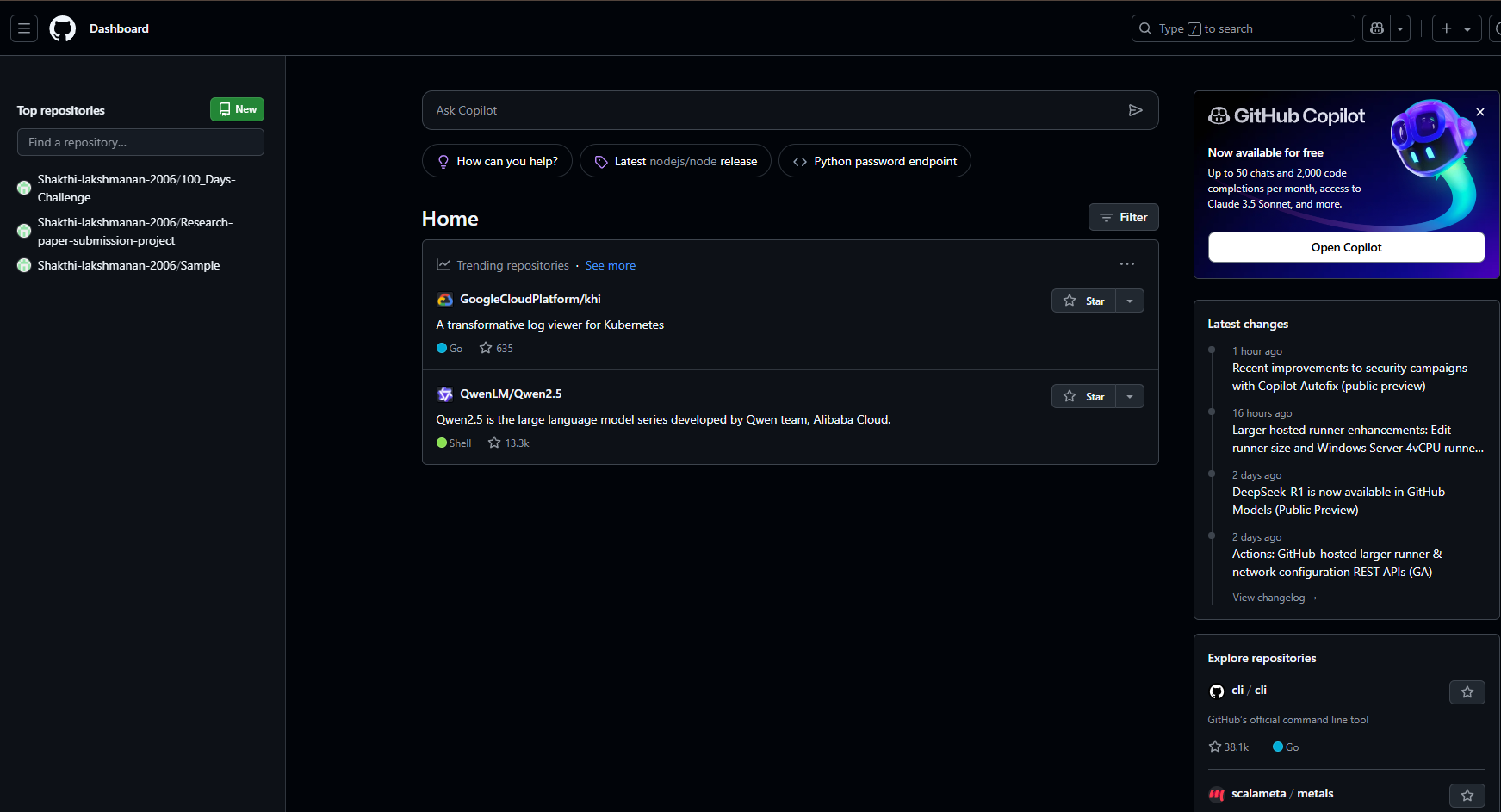
## Step 1:

#### Create a New Repository:

Once you're logged in, click the green **"New"** button on the top- right of your GitHub homepage to create a new repository.

Give your repository a name, for example, Sample.

Leave the other settings as default, and click **"Create repository"**.



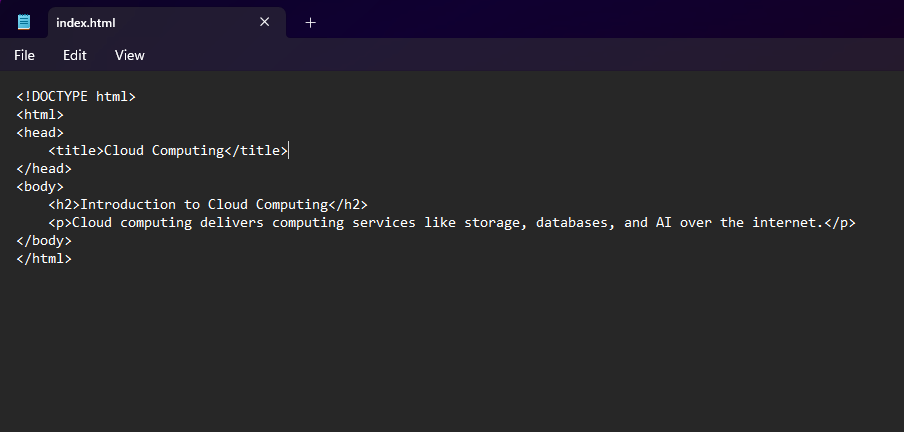
## Step 2:

Create a folder (e.g., demo) where you’ll keep all your website files.

Inside that folder, create the main file for your website, called

#### index.html.

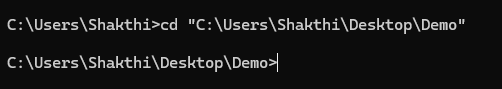
Here’s a simple example of what to put in your index.html:



Step 3:

Open **Command Prompt** and navigate to the folder where your index.html file is saved.

Use the cd command to navigate.



## Step 4:

Initialize a Git repository by running:



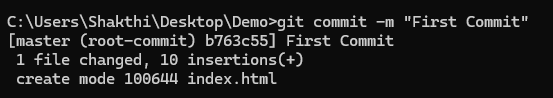
## Step 5:

Add your website files to the repository:



## Step 6:

Save the changes in Git with a commit message:



## Step 7:

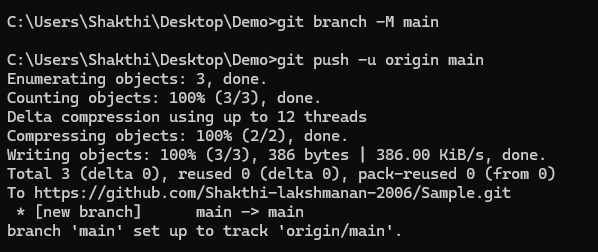
Go to your GitHub repository (the one you created earlier). Copy the **repository URL**:

In your Command Prompt, link your local repository to the GitHub repository:



## Step 8:

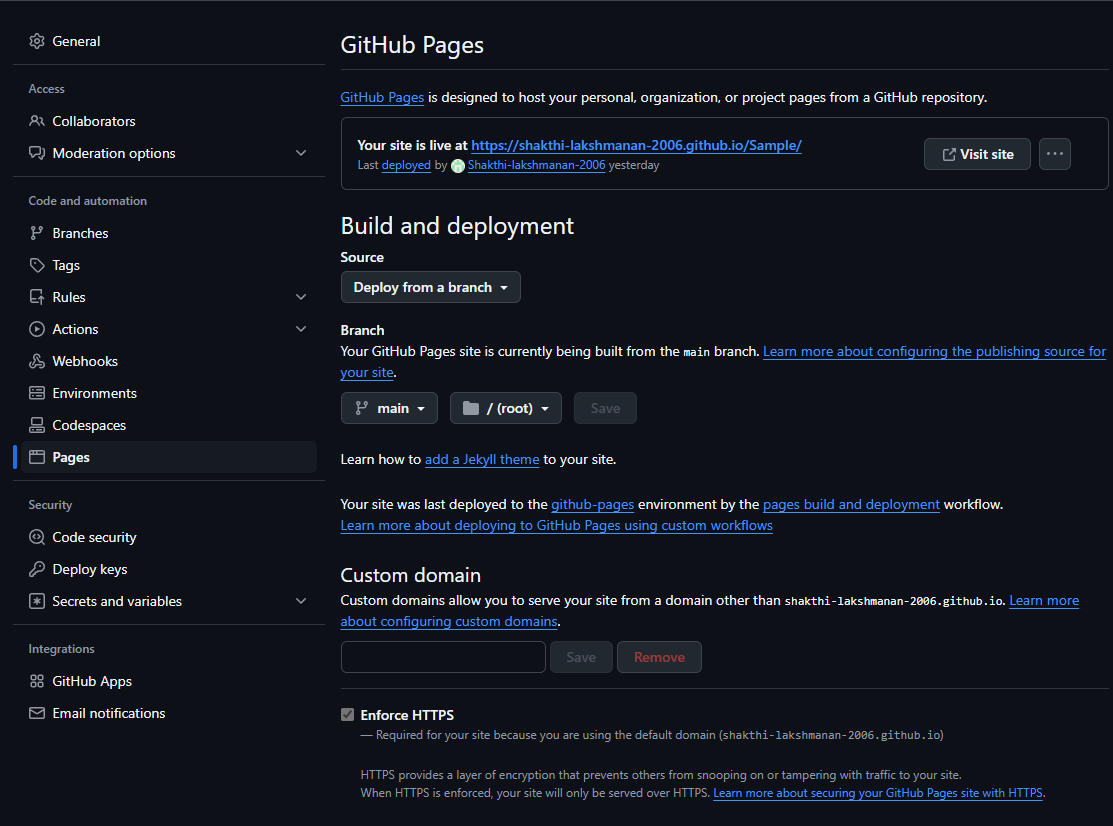
Push your files to GitHub:



## Step 9:

### 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:
   * **Branch**: main
   * **Folder**: / (root)
5. Click **Save**.

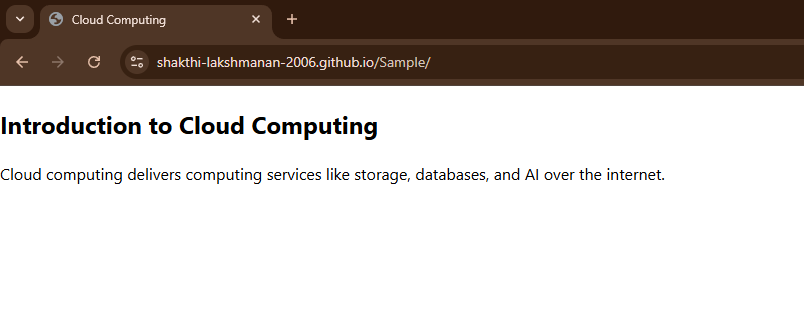


## Step 10:

### Access Your Website

Wait a few minutes for GitHub Pages to deploy your site. Visit your website at:

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



## **Outcome**

By completing this PoC of deploying a static website using GitHub Pages, you will:

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