# 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: Aksha S Department: CSE

# Introduction

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

# Overview

# This project showcases the deployment of a static website using GitHub Pages. Beginning with the setup of a GitHub repository, we will walk through the essential steps to host a fully functional static site. This includes initializing a Git repository, uploading files to GitHub, and configuring GitHub Pages for seamless deployment.

#### Key Features of GitHub Pages:

Free hosting for public repositories.

Support for static files (HTML, CSS, JavaScript). Easy integration with version control through Git.

# Objectives

1. Learn the fundamentals of GitHub Pages and its deployment process.
2. Understand the importance of static website hosting and its use cases.
3. Gain hands-on experience in using Git and GitHub for project versioning and hosting.
4. Successfully publish a static website and make it publicly accessible.

# 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.

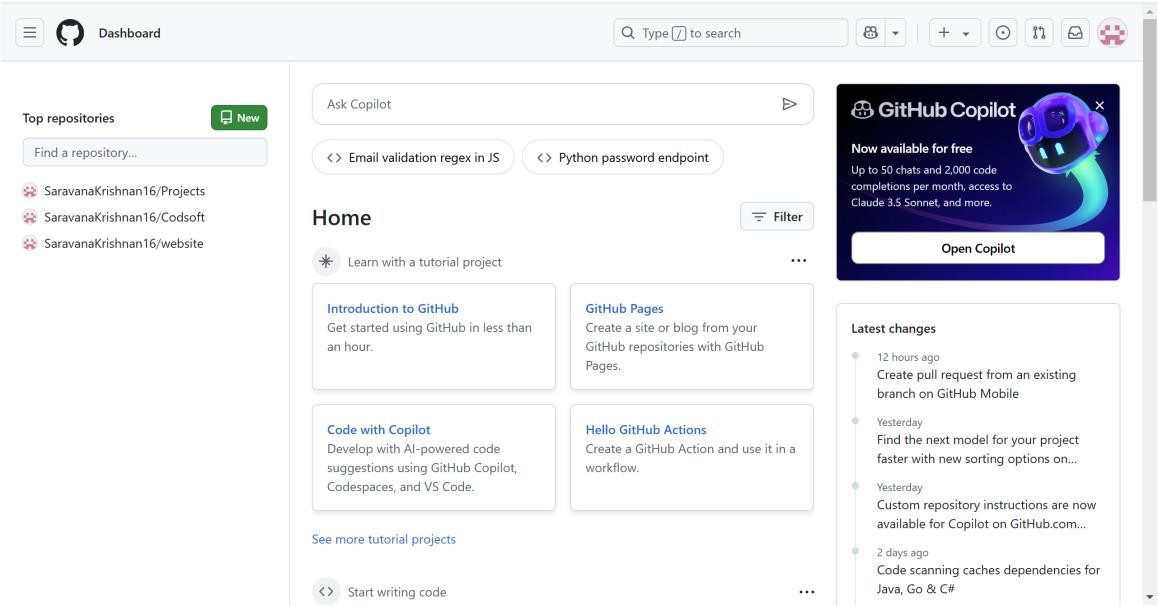
# Step-by-Step Overview

## Step 1:

#### Create a New Repository:

Once you're logged in, click the green **"New"** button on the top- right 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"**.



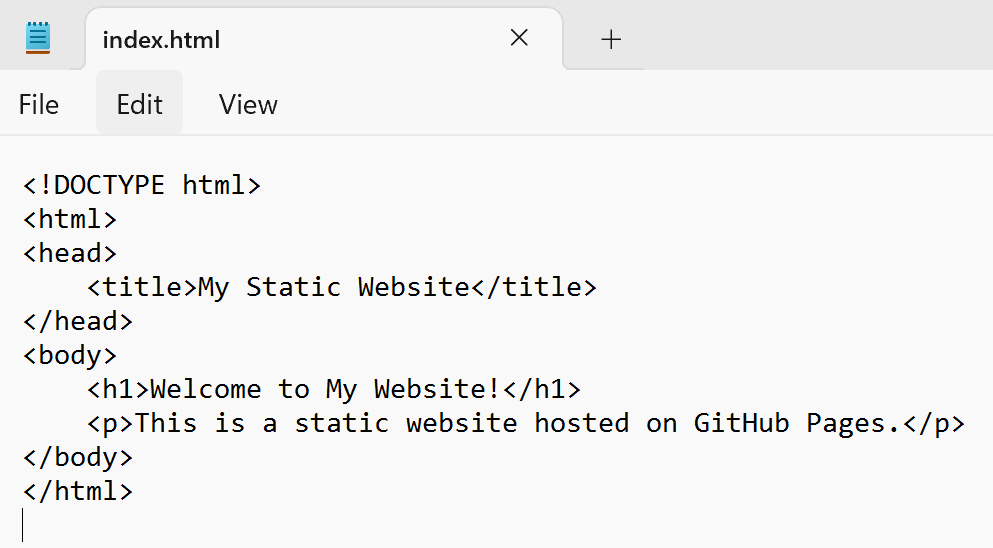
## Step 2:

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.

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.



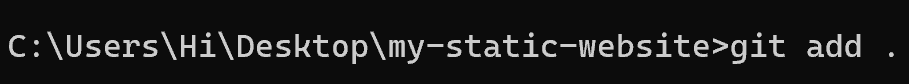
## Step 4:

Initialize a Git repository by running:



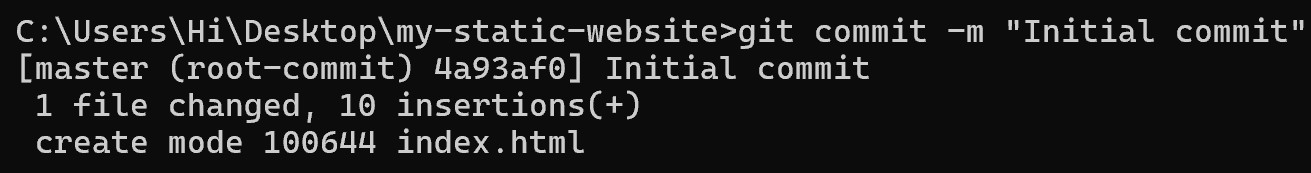
## Step 5:

Add your website files to the repository:



## Step 6:

Save the changes in Git with a commit message:



## Step 7:

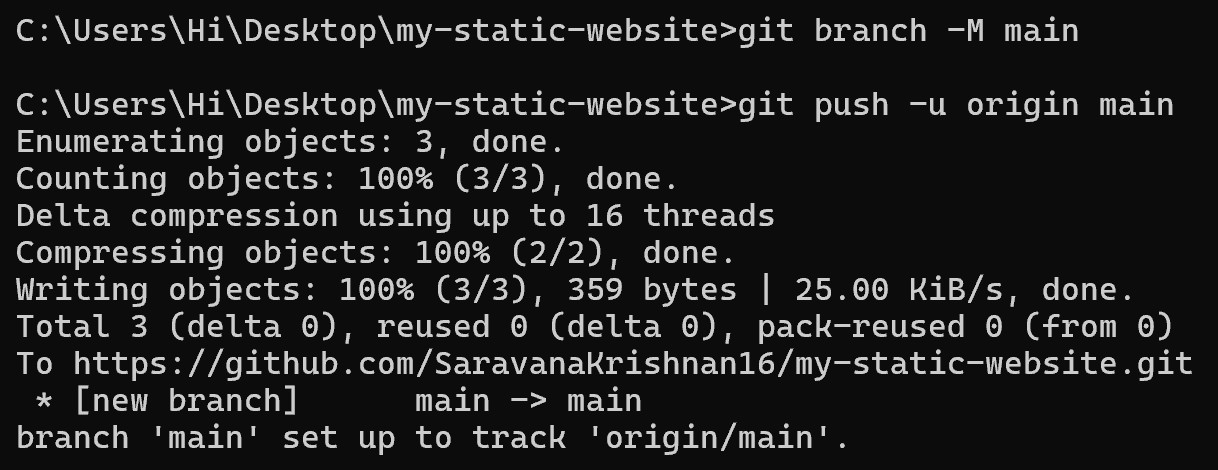
Go to your GitHub repository (the one you created earlier). Copy the **repository URL**:

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 Proof of Concept (PoC) for deploying a static website using GitHub Pages, you will:

1. Set up and configure a GitHub repository for your project.
2. Initialize a local Git repository and connect it to GitHub.
3. Upload static website files (HTML, CSS, JavaScript) to GitHub.
4. Enable GitHub Pages in the repository settings to publish your site.
5. Access your live website via a GitHub Pages URL.
6. Gain hands-on experience with essential Git commands such as git init, git add, git commit, git remote add, and git push.
7. Learn how to host a static website for free using GitHub Pages.