

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: SAIRAM S

Department: IT

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.

Step-by-Step Overview

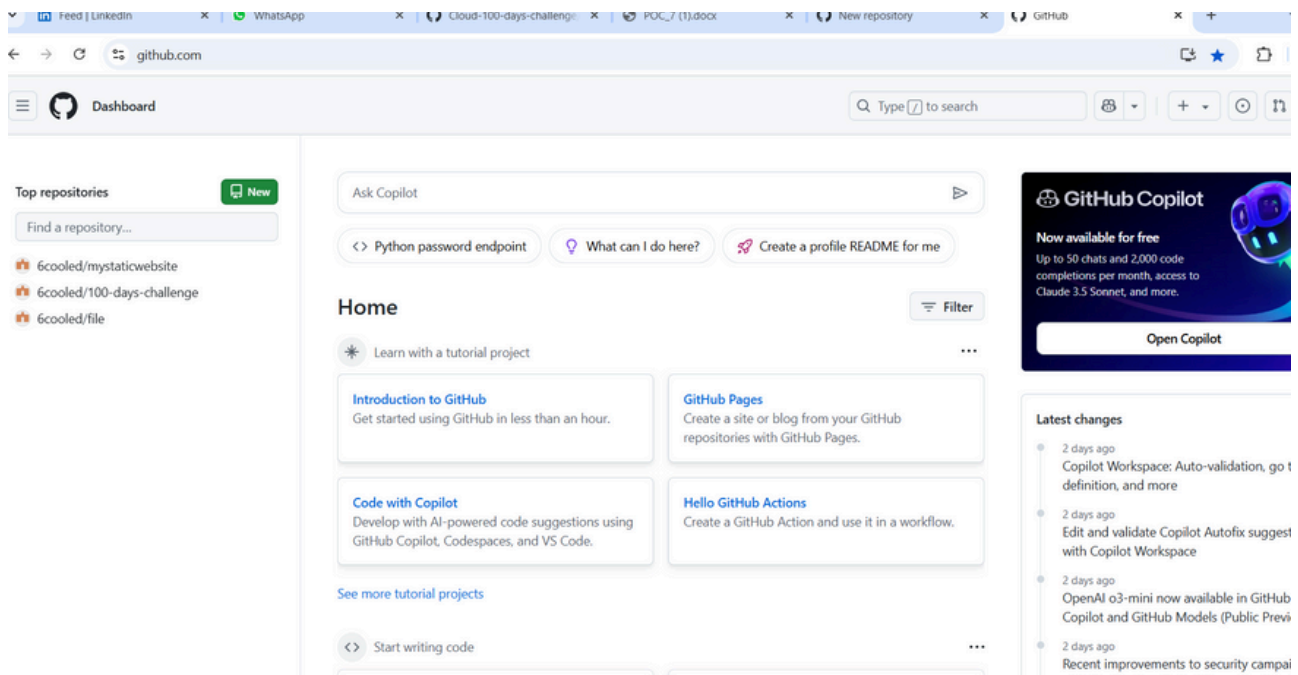
Step 1:

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:

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Welcome to Hackathon</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      background-color: #1e1e2e;
      color: white;
      margin: 0;
      padding: 0;
      display: flex;
      flex-direction: column;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    h1 {
      font-size: 3em;
      margin: 20px 0;
    }
    .btn {
      background-color: #ff9800;
      color: white;
      padding: 15px 30px;
      font-size: 1.2em;
      border: none;
      border-radius: 5px;
      cursor: pointer;
      transition: background 0.3s;
    }
    .btn:hover {
      background-color: #e68900;
    }
  </style>
</head>
<body>
  <h1>Welcome to Hackathon</h1>
  <button class="btn">Get Started</button>
</body>
</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\sairam>cd C:\Users\sairam\Desktop\poc 7
C:\Users\sairam\Desktop\poc 7>
```

Step 4:

Initialize a Git repository by running:

```
C:\Users\sairam\Desktop\poc 7>git init  
Initialized empty Git repository in C:/Users/sairam/Desktop/poc 7/.git/  
C:\Users\sairam\Desktop\poc 7>
```

Step 5:

Add your website files to the repository:

```
C:\Users\sairam\Desktop\poc 7>git add .  
  
C:\Users\sairam\Desktop\poc 7>
```

Step 6:

Save the changes in Git with a commit message:

```
C:\Users\sairam\Desktop\poc 7>git commit -m "Initial commit"  
[master (root-commit) 025e05e] Initial commit  
1 file changed, 50 insertions(+)  
create mode 100644 index.html  
  
C:\Users\sairam\Desktop\poc 7>
```

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:

```
C:\Users\sairam\Desktop\poc 7>git remote add origin https://github.com/6cooled/mystaticwebsite.git
C:\Users\sairam\Desktop\poc 7>
```

Step 8:

Push your files to GitHub:

```
C:\Users\sairam\Desktop\poc 7>git branch -M main
C:\Users\sairam\Desktop\poc 7>
C:\Users\sairam\Desktop\poc 7>git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 789 bytes | 789.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/6cooled/mystaticwebsite.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Step 9:

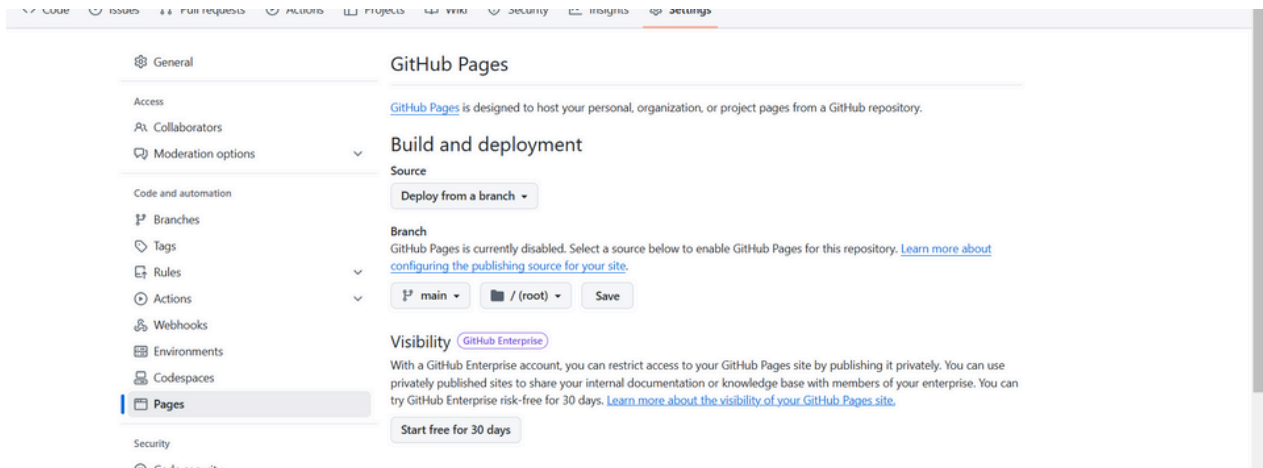
Enable GitHub Pages

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.

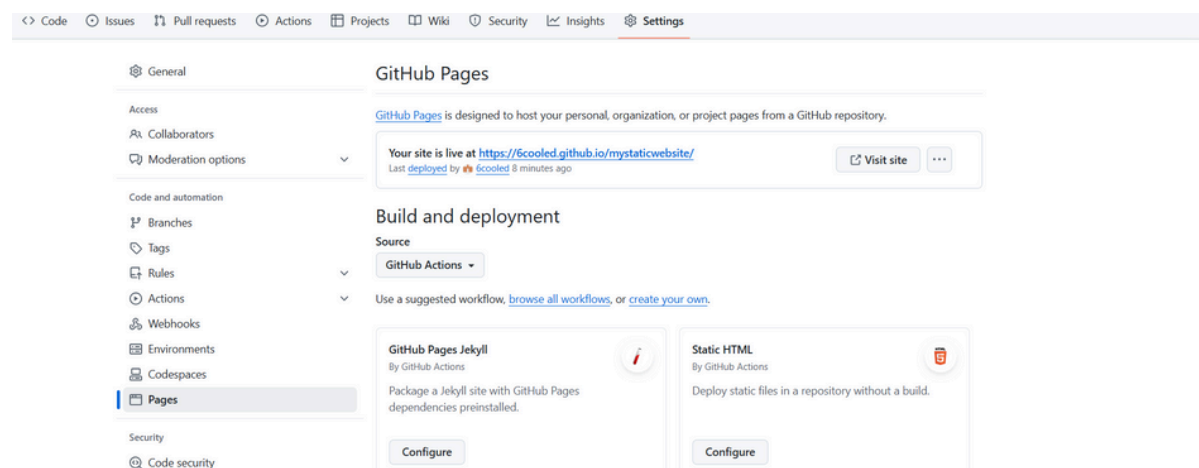


Step 10: Access Your Website

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.