

DEPLOY YOUR STATIC WEBSITE USING GITHUB PAGES

(Host your local Git repository's static website directly using GitHub Pages.)

INTRODUCTION:

GitHub Pages is a free hosting service provided by GitHub that allows you to host static websites directly from a repository. It is commonly used for personal websites, project documentation, and open-source projects.

If you have a 'local Git repository' containing a static website (HTML, CSS, and JavaScript files), you can publish it online without needing a separate hosting provider. By pushing your local repository to GitHub and enabling GitHub Pages, your website will be accessible via a github.io domain.

IMPORTANCE:

1. Free and Easy Hosting

- ❖ GitHub Pages provides a free hosting service for static websites.
- ❖ No need for additional servers or hosting providers.

2. Version Control with Git

- ❖ Since the website is hosted on a Git repository, you get version control capabilities.
- ❖ You can track changes, revert to previous versions, and collaborate easily.

3. Custom Domain Support

- ❖ You can use a custom domain instead of the default username.github.io URL.
- ❖ Supports HTTPS for security.

4. Automated Deployment

- ❖ Any changes pushed to the repository automatically update the live website.
- ❖ No need for manual deployment steps.

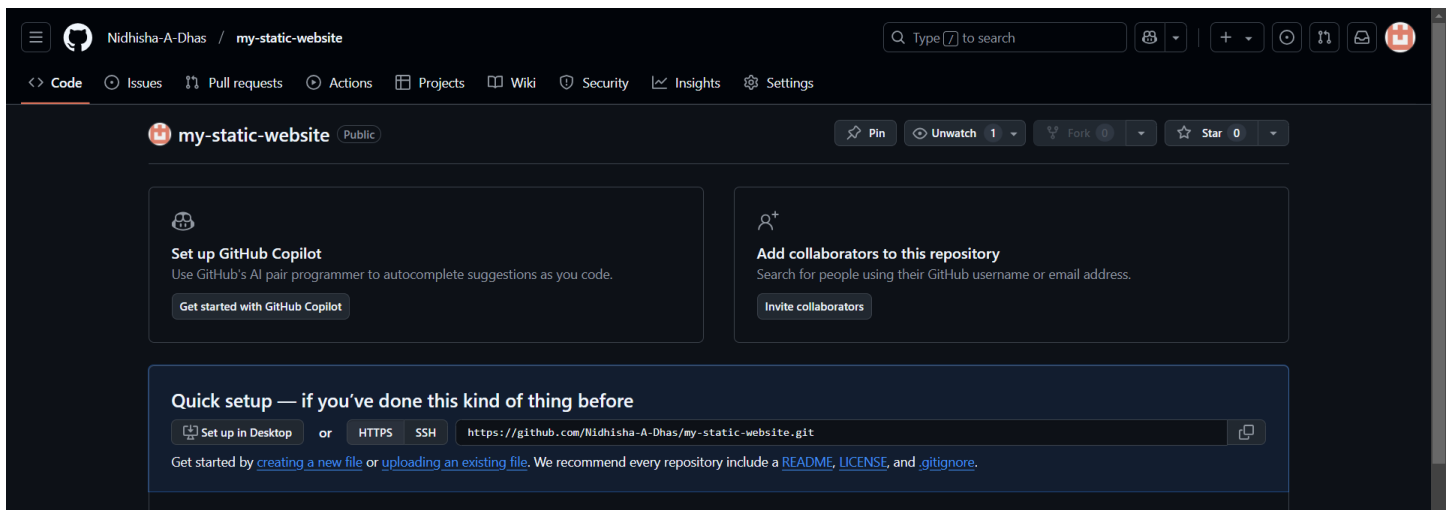
5. Ideal for Portfolios & Documentation

- ❖ Perfect for personal portfolios, project pages, and documentation.
- ❖ Widely used by open-source developers to showcase projects.

STEP BY STEP OVERVIEW:

Step 1: CREATE A NEW REPOSITORY

- Login into your GitHub account and create a new repository.
- Give the repository name and leave the other settings as default.



Step 2: CREATE A WEBSITE

- Create a folder to save the simple html file.
- Create a simple html file for your website and save it successfully.

```
index.html
File Edit View
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My Simple Website</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      margin: 50px;
    }
    h1 {
      color: blue;
    }
  </style>
</head>
<body>
  <h1>Welcome to My Website</h1>
  <p>This is a simple HTML page hosted using GitHub Pages.</p>
</body>
</html>
```

Step 3: DEPLOYMENT

- Open the command prompt and navigate to the folder where the 'index.html' file is stored.

```
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

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

- Now, initialize the git repository.

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

- Then, add the website file to the repository.

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

- Now, save the changes with the commit command.

```
C:\Users\Arulldhas\Desktop\my-static-website>git commit -m "Initial commit"
[master (root-commit) a7a57ed] Initial commit
1 file changed, 22 insertions(+)
create mode 100644 index.html
```

Step 4: PUSH FILES TO GITHUB:

- Go to the GitHub repository and copy the repository URL.
- In the command prompt, link your local repository to your GitHub repository.

```
C:\Users\Arulldhas\Desktop\my-static-website>git remote add origin https://github.com/Nidhisha-A-Dhas/my-static-website.git
```

- Now push the files as follows:

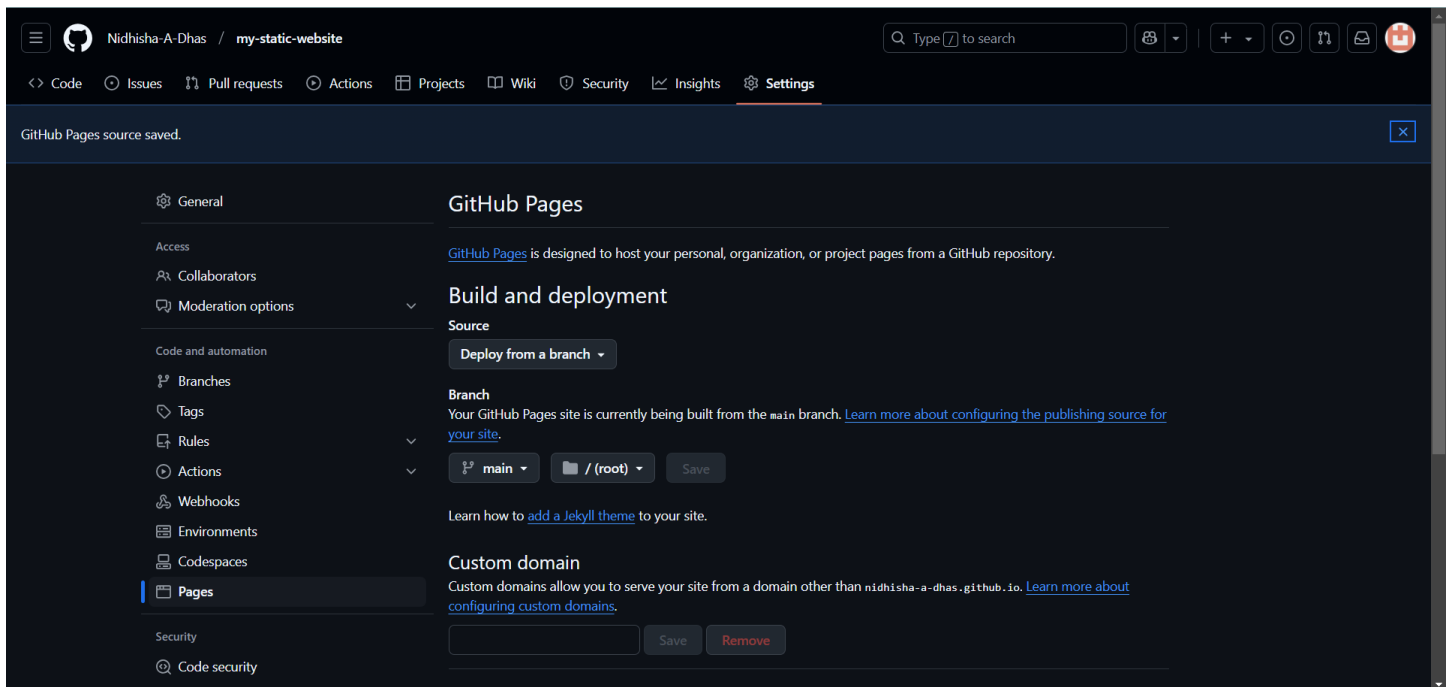
```
C:\Users\Arulldhas\Desktop\my-static-website>git branch -M main

C:\Users\Arulldhas\Desktop\my-static-website>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), 519 bytes | 259.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Nidhisha-A-Dhas/my-static-website.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

C:\Users\Arulldhas\Desktop\my-static-website>
```

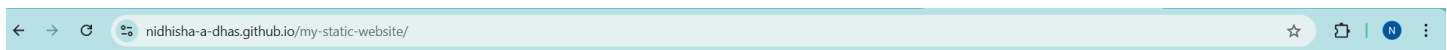
Step 5: ENABLE THE PAGES

- Go to your GitHub repository and click on settings and the navigate to 'pages'.
- Under 'Source' select BRANCH- main & FOLDER- /(root)
- Then, click 'save'.



Step 6: ACCESS YOUR WEBSITE

- Wait a few minutes for the GitHub pages to deploy your site.
- Visit your website at <https://<your-username>.github.io/<your-repository>>



CONCLUSION:

By completing this POC, you will:

- Successfully create and configure GitHub repository for your project.
- Initialize a Git repository in your local folder and link it with GitHub.
- Enable GitHub pages to host the static website.