

AUTOMATE STATIC WEBSITE DEPLOYMENT LOCALLY

AIM:

Create a script that updates your server whenever changes are pushed.

INTRODUCTION:

GitHub Pages is a static site hosting service designed to publish your project 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.

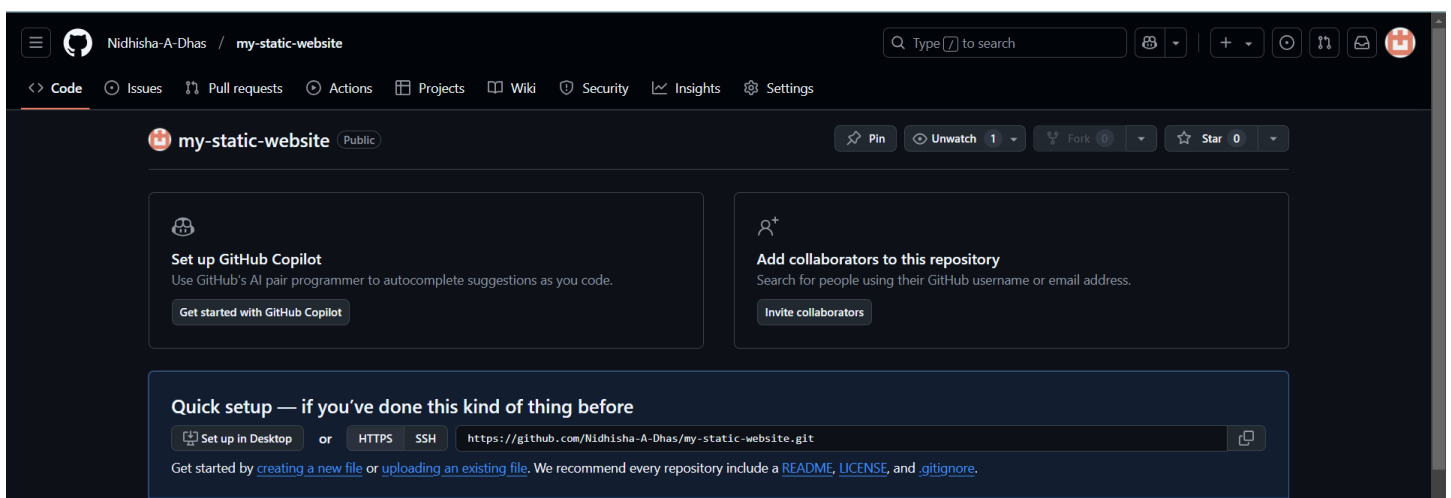
OVERVIEW:

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.

STEP BY STEP OVERVIEW:

Step 1: CREATE NEW REPOSITORY

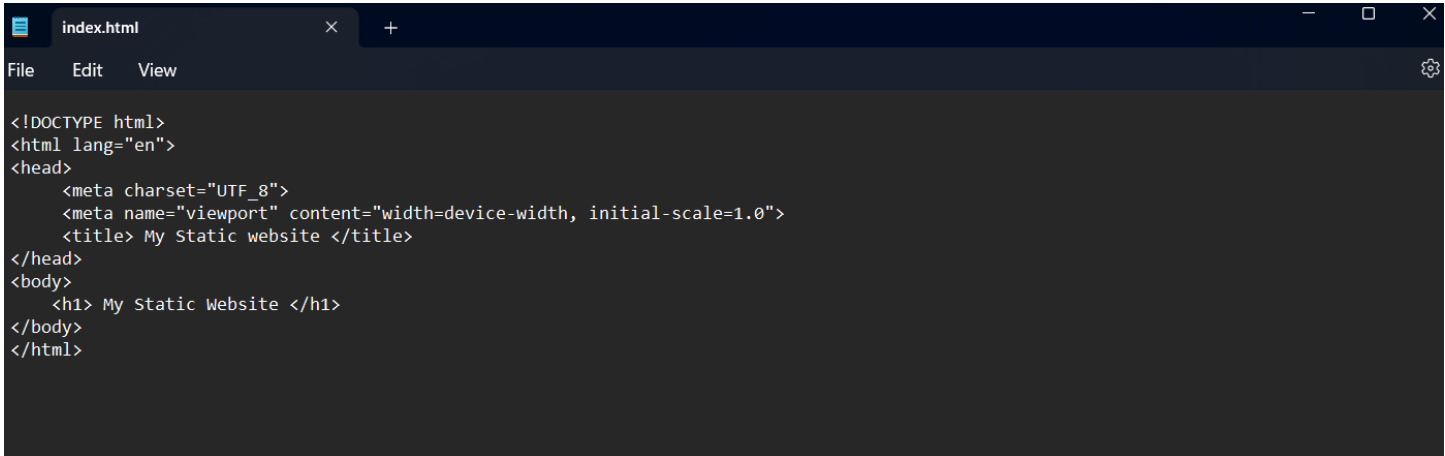
- Login into your Github account.
- Once logged-in, create a new repository. Specify the repository name eg: my-static-website.



Step 2: CREATE WEBSITE FILES AND FOLDER

- Create a folder named 'my-static-website'.
- Store all the website files in this folder.

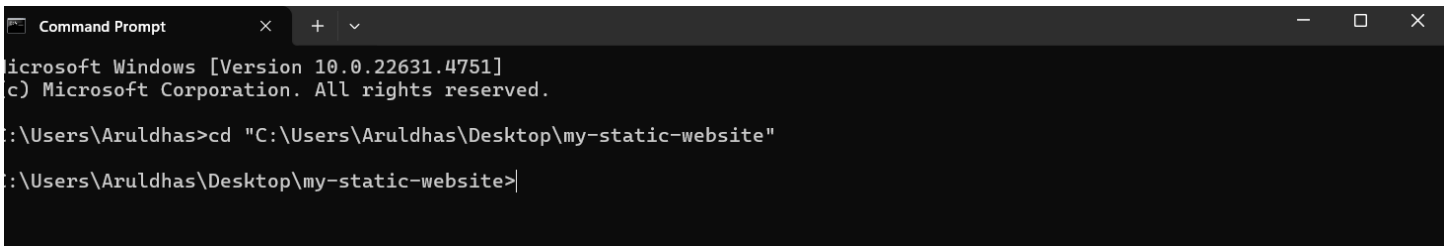
- Create a main file for your website called 'index.html'.

A screenshot of a code editor window titled 'index.html'. The editor shows the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF_8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title> My Static website </title>
</head>
<body>
  <h1> My Static Website </h1>
</body>
</html>
```

Step 3: INITIALISING GIT REPOSITORY

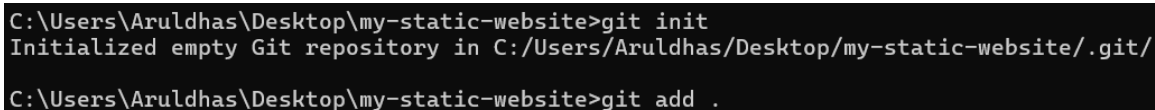
- Open the command prompt and navigate to the folder where the 'index.html' file is stored
- Use the command 'cd' to navigate it.

A screenshot of a Windows Command Prompt window. The text shows the user navigating to a specific directory:

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

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

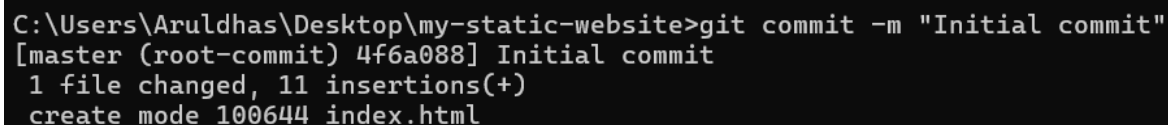
- Initialize the Git repository by using the command 'git init'.
- Add your website files to the repository by using 'git add .'.

A screenshot of a Windows Command Prompt window showing the execution of Git commands:

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

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

- Save the changes in git by using the commit message.

A screenshot of a Windows Command Prompt window showing the execution of a Git commit command:

```
C:\Users\Arulldhas\Desktop\my-static-website>git commit -m "Initial commit"
[master (root-commit) 4f6a088] Initial commit
1 file changed, 11 insertions(+)
create mode 100644 index.html
```

Step 4: LINK LOCAL REPOSITORY & PUSH FILES

- Go to your Github repository, copy the URL and link your local repository to the Github repository.

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

- Push files to the Github repository.

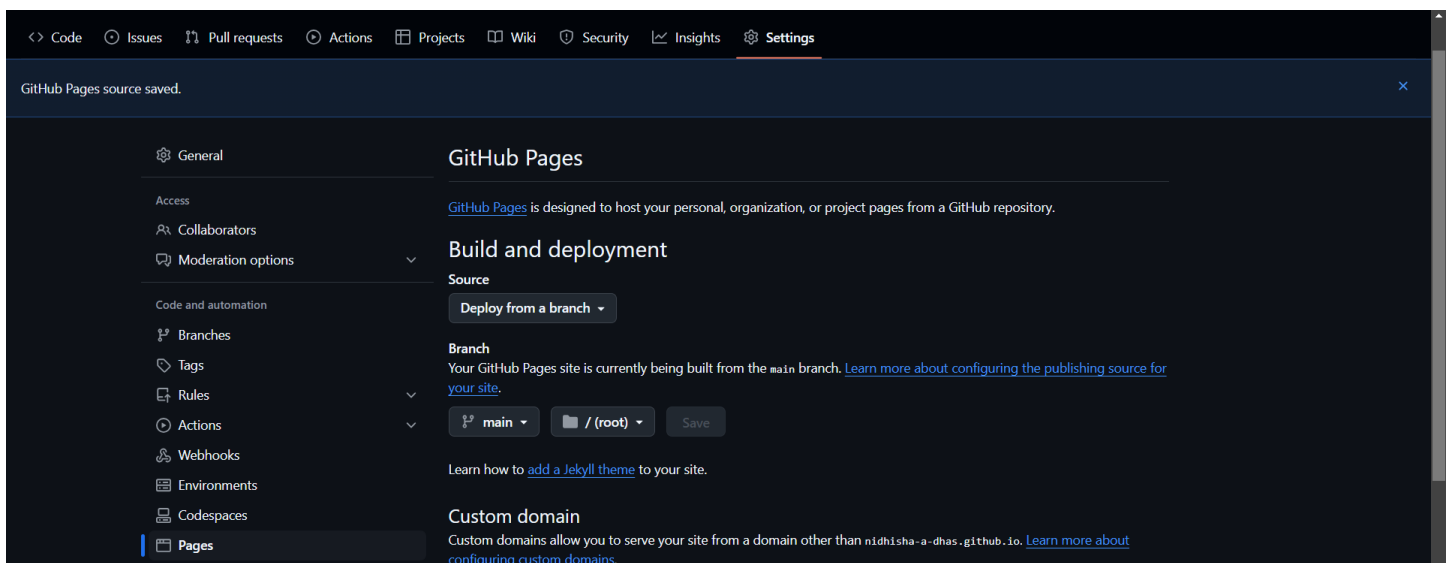
```
C:\Users\Aruldas\Desktop\my-static-website>git branch -M main

C:\Users\Aruldas\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), 385 bytes | 385.00 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\Aruldas\Desktop\my-static-website>|
```

Step 5: ENABLE GITHUB PAGES

- Go to your repository on Github.
- Click on settings; scroll down to pages section. Under source, select branch (main), folder (/(root))



Step 6: ACCESS YOUR WEBSITE

- You can access your website at ‘https://<your-username>.github.io/<your-repository>’.
- Wait for few minutes for Github pages to deploy.

My Static Website

CONCLUSION:

Thus, from this POC, we have learnt

- To successfully create and configure a Github repository.
- Initialize a Git repository in your local project folder and link it to Github.
- Upload static website files.
- Enable Github pages to host the static-website.
- Access the static website live on the web via a Github pages URL.