



St. JOSEPH'S
GROUP OF INSTITUTIONS
OMR, CHENNAI - 119



Placement Empowerment Program

Cloud Computing and DevOps Centre

Deploy your static website using GitHub pages

Name: Jeffersen Godfrey A M

Department: CSE



INTRODUCTION:

Web hosting is essential for making websites accessible to users worldwide. **GitHub Pages** is a free and convenient way to host static websites directly from a GitHub repository. It is widely used by developers to showcase projects, portfolios, and documentation. This project focuses on deploying a static website using **GitHub Pages**, ensuring an efficient and seamless web hosting experience.

OVERVIEW:

This documentation covers the step-by-step process of deploying a **static website** on GitHub Pages. It includes:

- Setting up a **GitHub repository**
- Uploading **HTML, CSS, and other files**
- Configuring **GitHub Pages settings**
- Accessing the **live website**

This project leverages **Git** and **GitHub** for version control, ensuring easy management and updates of the website.

OBJECTIVES:

The primary objectives of this project are:

- 1) **Understand** how to deploy a static website using GitHub Page.
- 2) **Learn** how to use Git for version control
- 3) **Host** a website without any paid services
- 4) **Share** the website with a public link
- 5) **Improve** web development and cloud deployment skills.

IMPORTANCE OF THIS TASK:

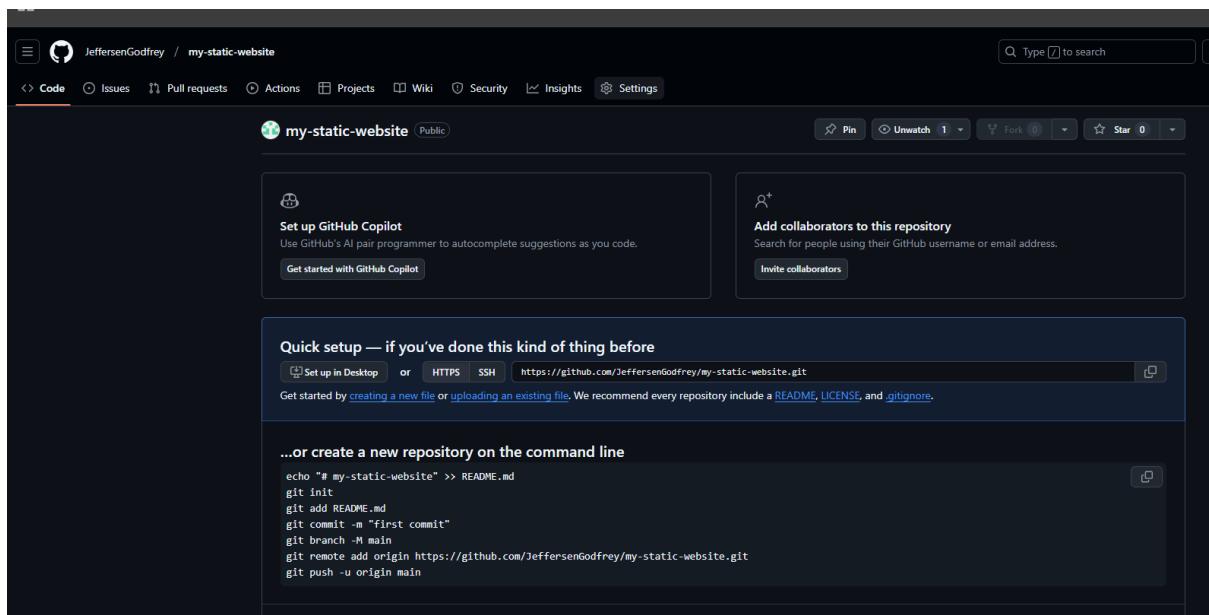
Deploying a static website on GitHub Pages is beneficial for:

- ◆ **Beginner Developers** – Helps in learning website hosting and version control
- ◆ **Project Showcasing** – Ideal for hosting portfolios, resumes, and personal projects
- ◆ **Cost Efficiency** – Free hosting with custom domain support
- ◆ **Collaboration** – Makes sharing and maintaining websites easier

STEP-BY-STEP OVERVIEW:

Step 1: Create a GitHub Repository

Go to GitHub and click on new repository and create a repository of your desired name.



Step 2: Create a project folder (GitHub Pages)



Step 3: Create a html file and paste into the folder.

```
File    Edit    View
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Football Lover</title>
    <body>WE ARE UNITED</body>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            background-
```

Step 4: Open command prompt and type cd and paste your folder path.

```
C:\Users\Jeffersen Godfrey>cd C:\Users\Jeffersen Godfrey\Desktop\GitHub Pages
```

Step 5: Initialize Git and enter the code **git init**

```
C:\Users\Jeffersen Godfrey\Desktop\GitHub Pages>git init  
Initialized empty Git repository in C:/Users/Jeffersen Godfrey/Desktop/GitHub Pages/.git/
```

Step 6: To enter branch type **git branch**

```
C:\Users\Jeffersen Godfrey\Desktop\GitHub Pages>git branch
```

Step 7: Following that enter the code **git branch -M main**

```
C:\Users\Jeffersen Godfrey\Desktop\GitHub Pages>git branch -M main
```

Step 8: Enter **git add .** to add the files

```
C:\Users\Jeffersen Godfrey\Desktop\GitHub Pages>git add .
```

Step 9: Then to commit give the code **git commit -m "Initial commit"**

```
C:\Users\Jeffersen Godfrey\Desktop\GitHub Pages>git commit -m "Initial commit"
[main (root-commit) 6cf8b4f] Initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 AWS 1.pdf
```

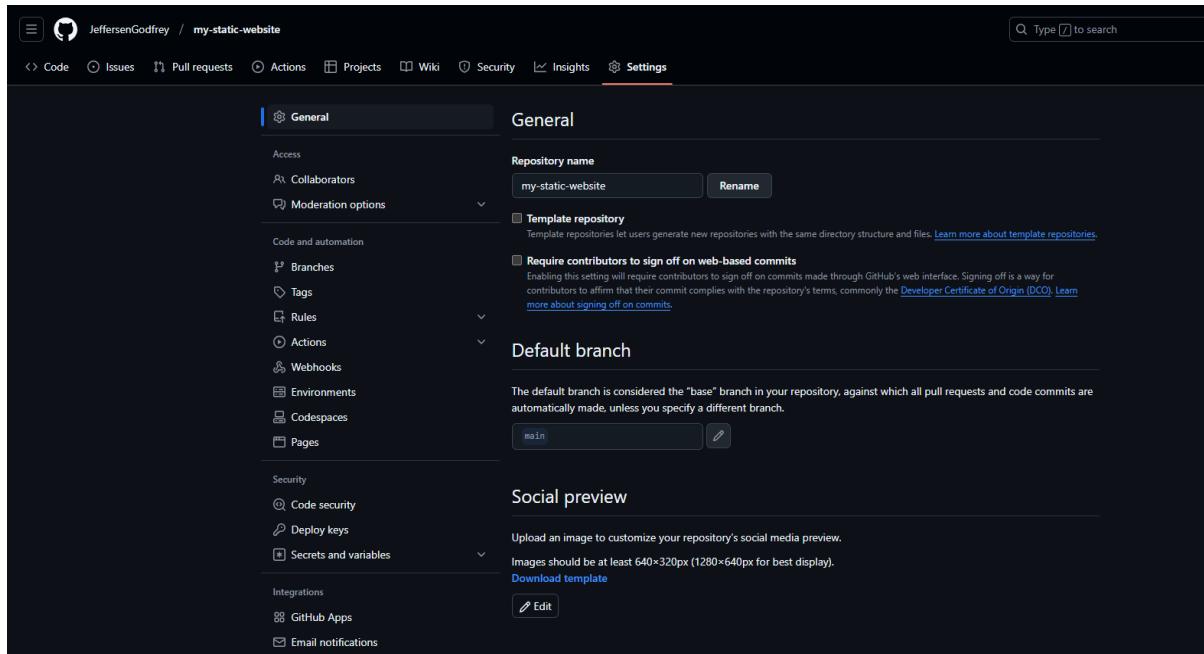
Step 10: To push the main branch to GitHub use **git push -u origin main**

```
C:\Users\Jeffersen Godfrey\Desktop\GitHub Pages>git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 196.50 KiB | 14.04 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/JeffersenGodfrey/my-static-website
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

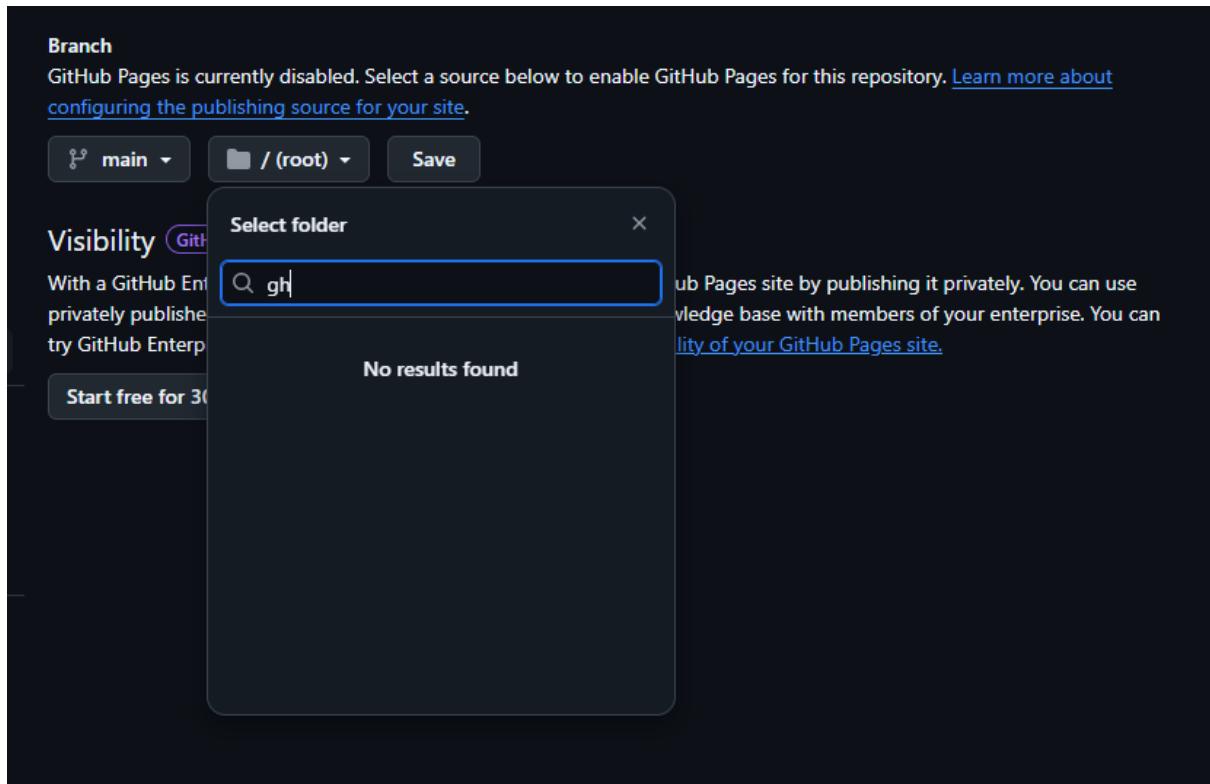
Step 11: To make sure your GitHub is connected correctly.

```
C:\Users\Jeffersen Godfrey\Desktop\GitHub Pages>git remote -v
origin  https://github.com/JeffersenGodfrey/my-static-website (fetch)
origin  https://github.com/JeffersenGodfrey/my-static-website (push)
```

Step 12: Go to your repository settings

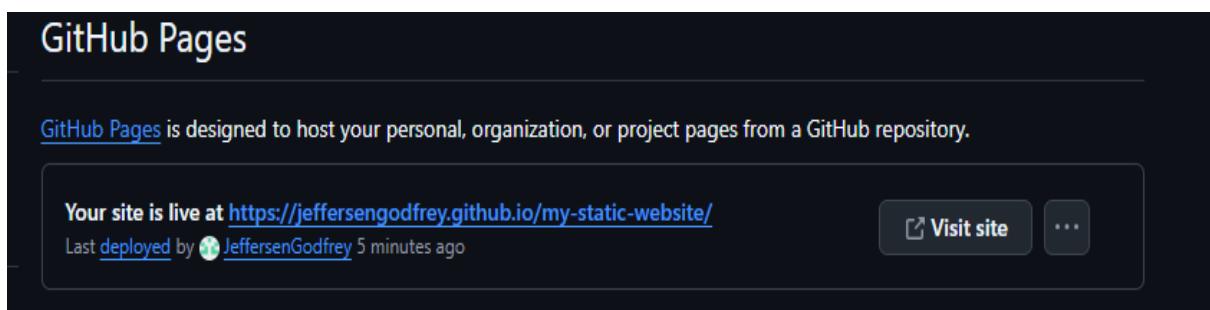


Step 13: In settings enable GitHub Pages and under branch select main



Step 14: After enabling GitHub Pages, GitHub will automatically deploy your website.

Wait for 1-2 minutes, then check the live link.



Step 15: Open the link on a browser to see the result

EXPECTED OUTCOME:

Successful Deployment: A fully functional **static website** will be hosted on **GitHub Pages** with a publicly accessible URL.

Version Control Management: The website files will be managed using **Git** and **GitHub**, enabling easy updates and collaboration.

Web Hosting Without Cost: The website will be live without any **server setup** or **hosting fees**, utilizing GitHub's free hosting service.

Improved Git & Web Development Skills: The process will enhance **Git commands knowledge**, **repository management**, and **basic web hosting concepts**.

Professional Online Presence: The project can serve as a **portfolio**, **documentation site**, or **personal webpage**, making it a valuable asset for future projects.

