**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: Mathimalar.P Department : ADS



**Introduction and Overview**

GitHub Pages is a free hosting service that allows developers to deploy static websites directly from their GitHub repositories. It provides an easy way to publish personal, project, or organizational pages without requiring external hosting services. By leveraging GitHub Pages, developers can efficiently manage and update their websites using Git version control.

**Objective**

.

The objective of this process is to host a static website using GitHub Pages by pushing the local Git repository to GitHub. This includes:

* Setting up a GitHub repository for the website
* Configuring GitHub Pages for deployment
* Ensuring the website is accessible through the provided GitHub Pages URL

.

Importance

Hosting a static website using GitHub Pages is significant for several reasons:

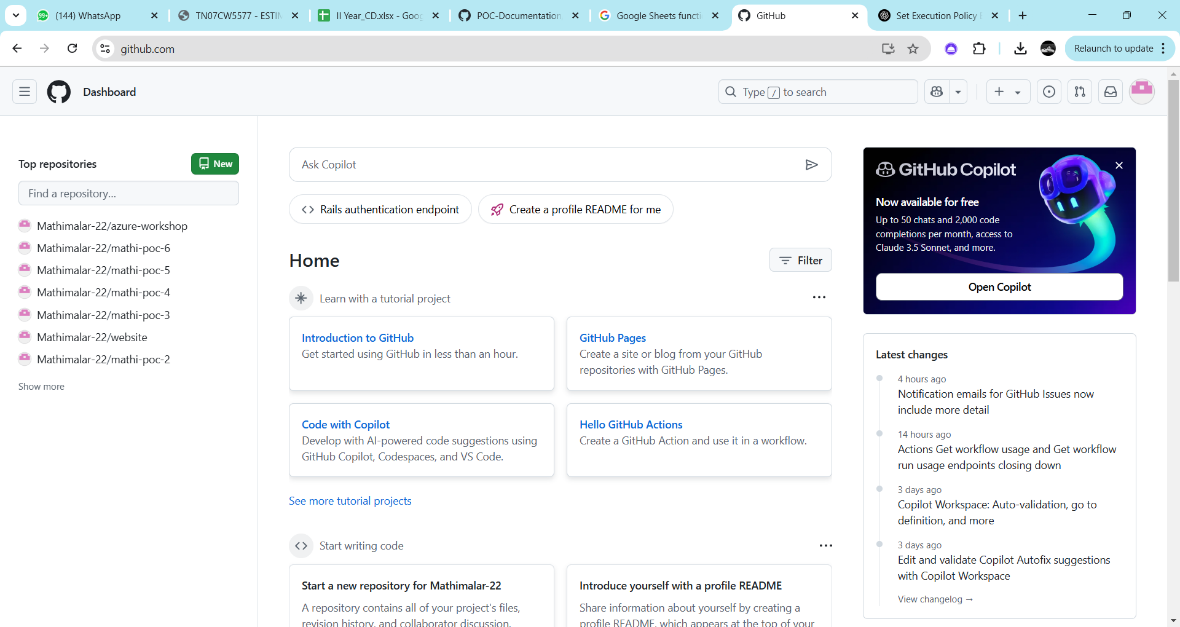
* **Free and Reliable Hosting** – GitHub Pages provides a cost-free and stable hosting solution for personal and project-based websites.
* **Easy Deployment** – Websites can be deployed with minimal effort by pushing changes to a GitHub repository.
* **Version Control** – Since it integrates with Git, all changes are tracked, making collaboration and rollback to previous versions seamless.
* **Custom Domain Support** – Users can link their custom domains to GitHub Pages for a professional web presence.
* **Ideal for Documentation and Portfolios** – Many developers and organizations use GitHub Pages to host documentation, resumes, or project showcases.

This makes GitHub Pages an essential tool for web developers, students, and open-source contributors.

**Step-by-Step Overview**

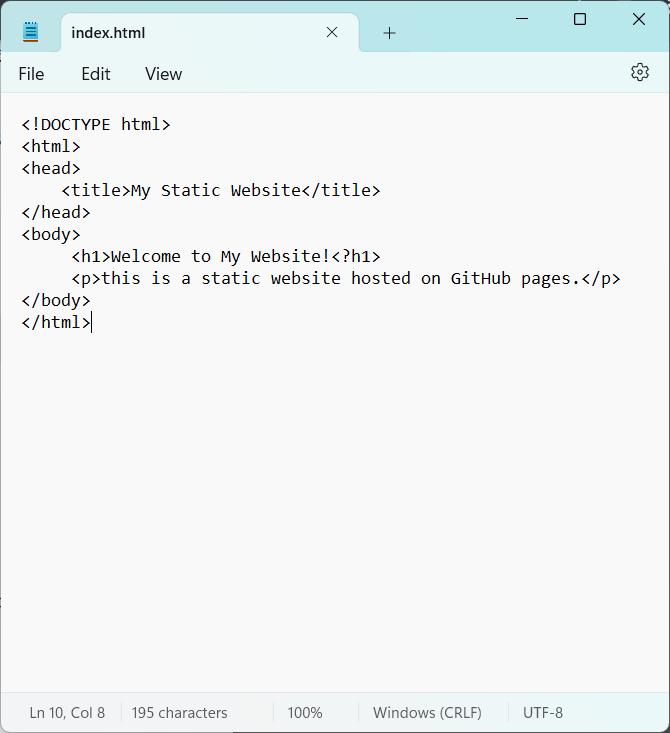
**Step 1**

Create a New Repository: Once you're logged in, click the green "New" button on the topright 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".



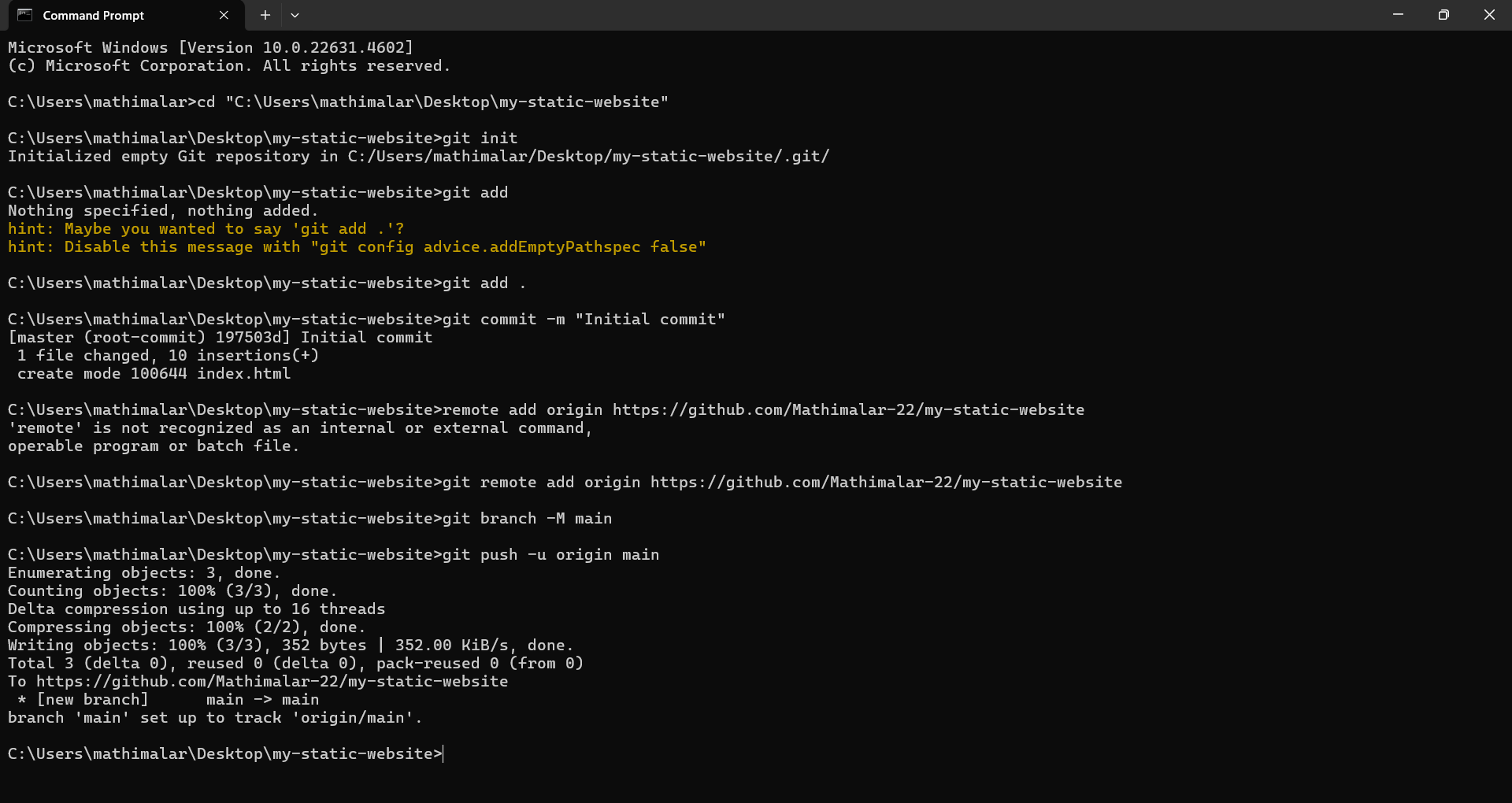
**Step 2**

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:



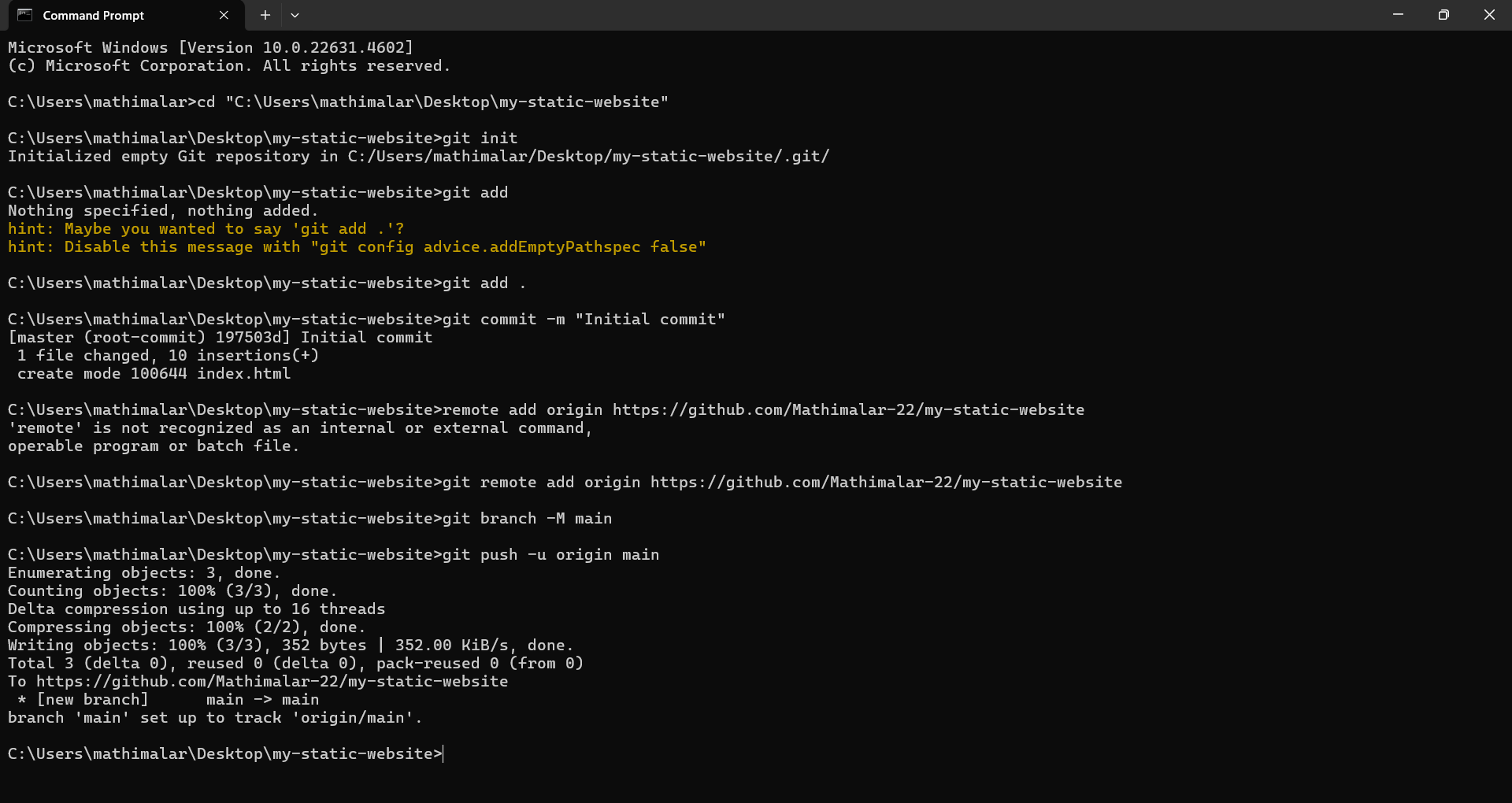
**Step 3**

Open Command Prompt and navigate to the folder where your index.html file is saved. Use the cd command to navigate.

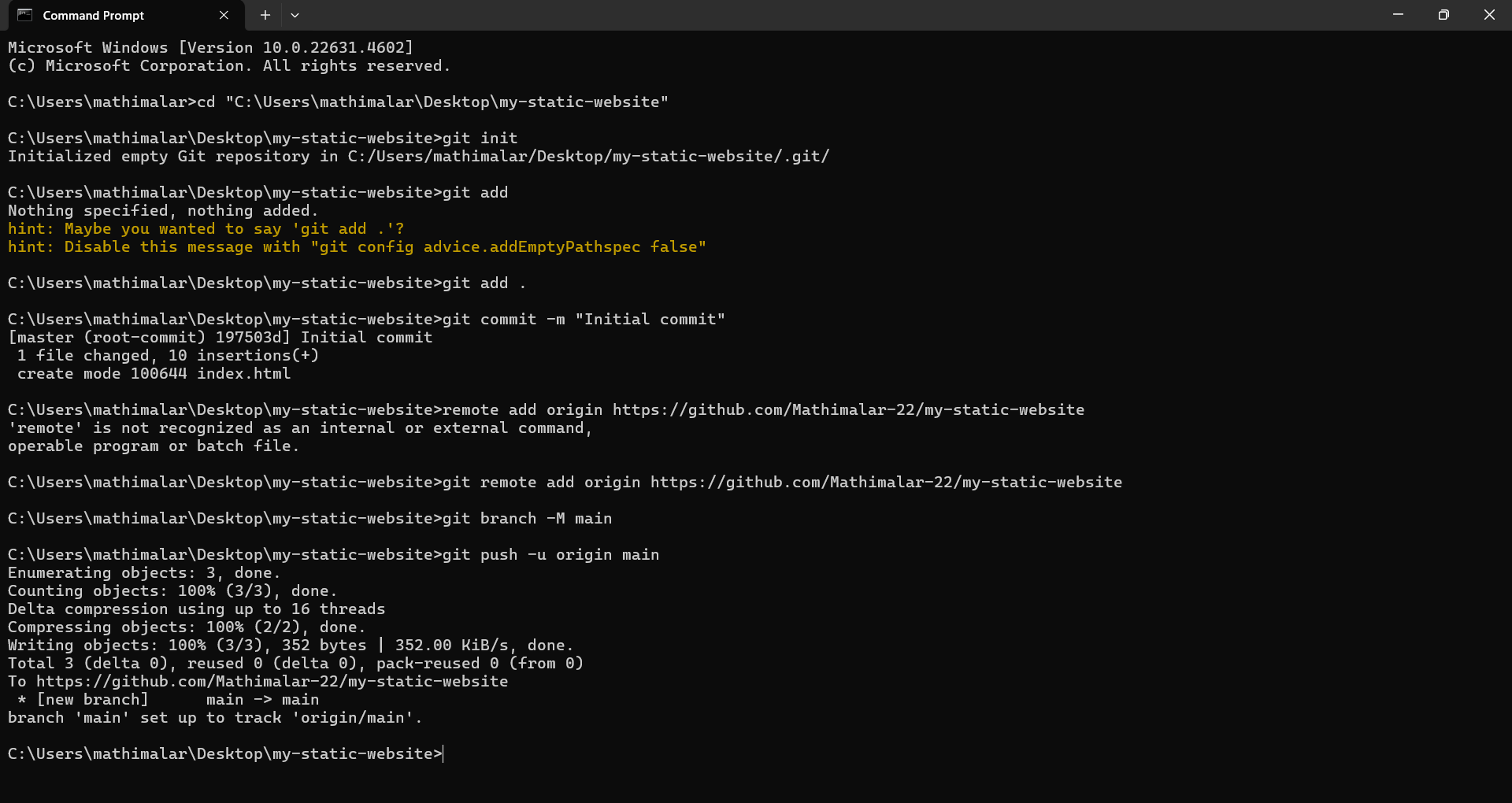


**Step 4:**

Initialize a Git repository by running:

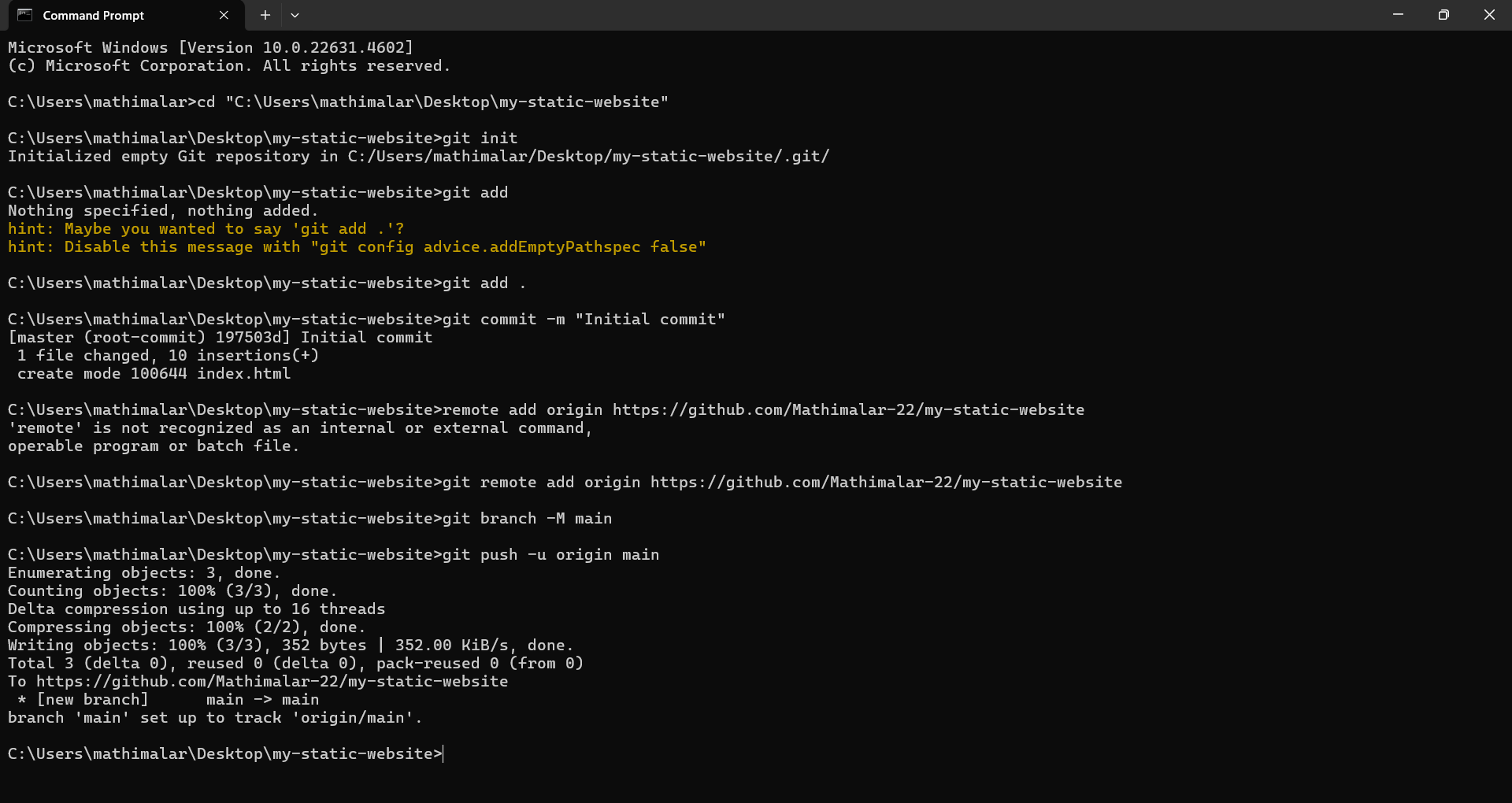


**Step 5:** Add your website files to the repository:



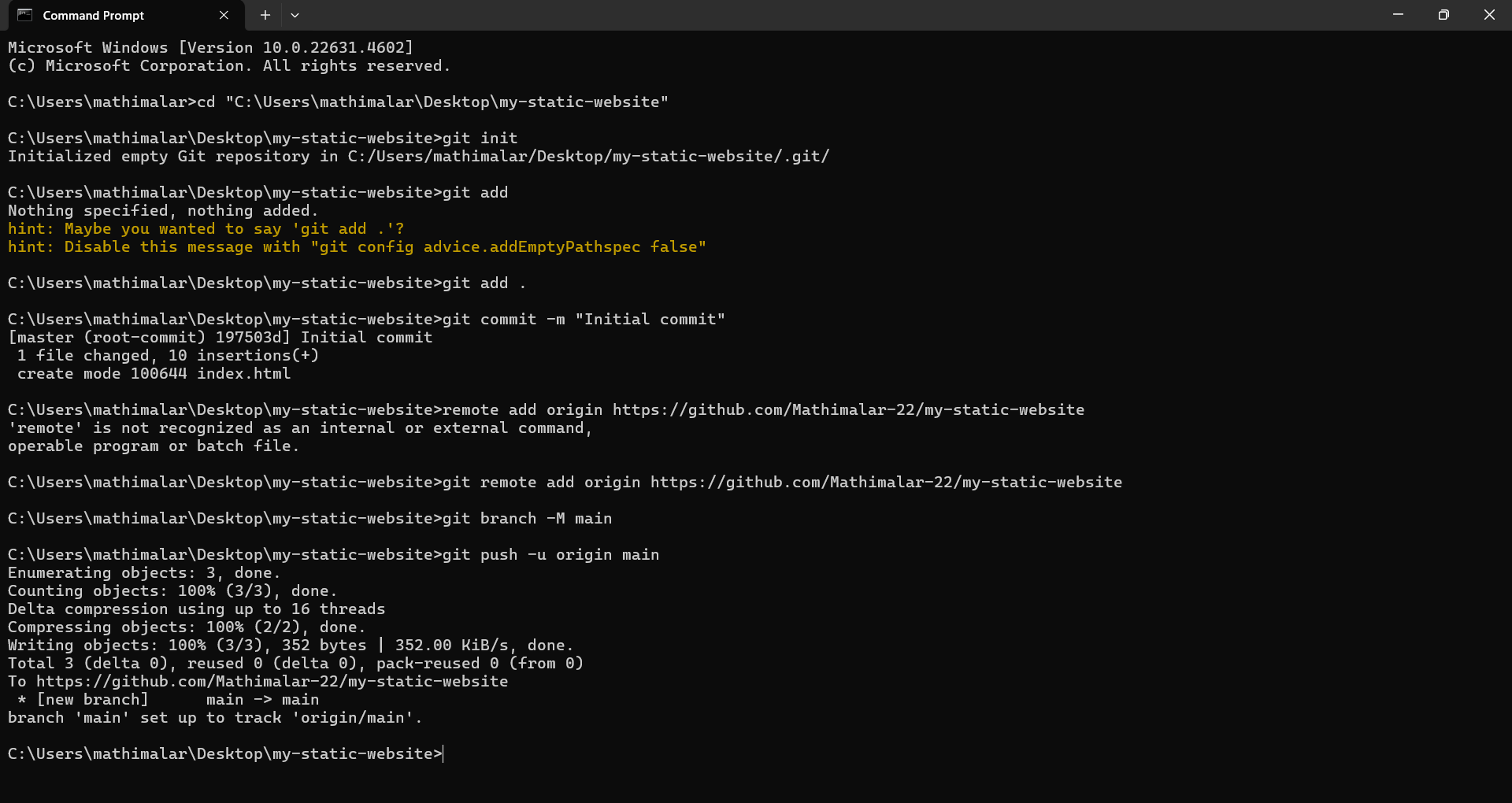
**Step 6**

Save the changes in Git with a commit message:



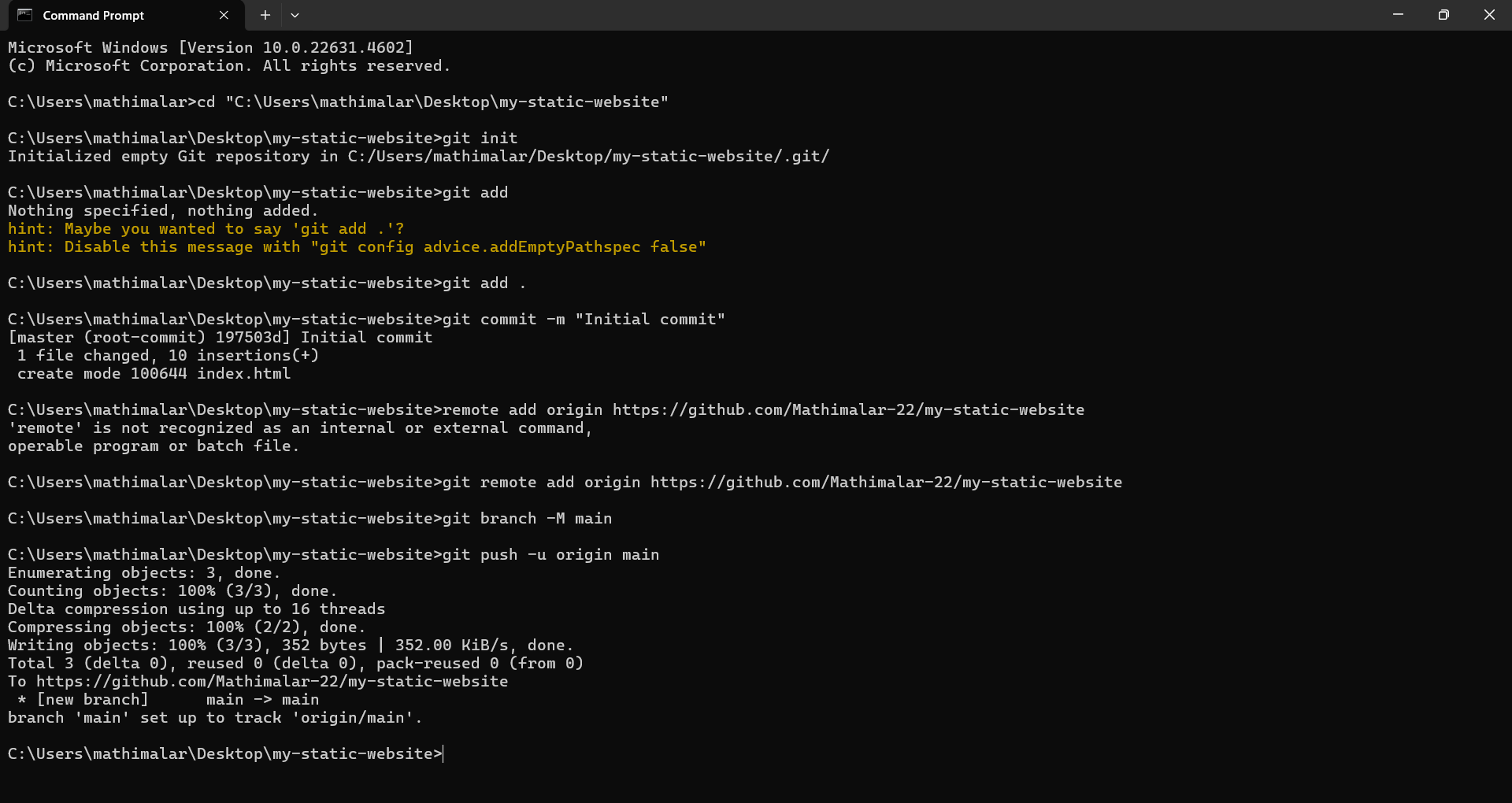
**Step 7:**

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:

****

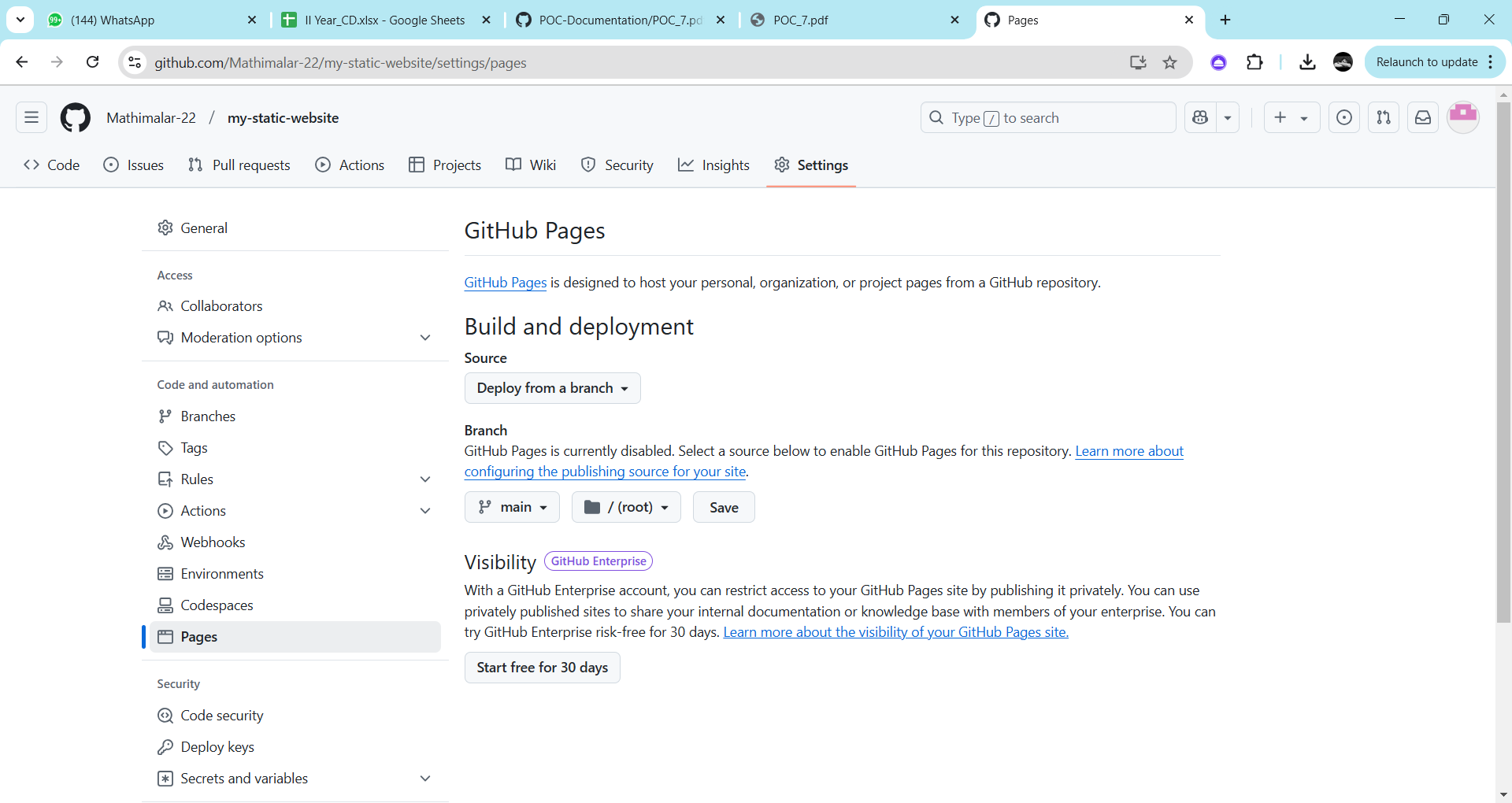
**Step 9:** Enable GitHub Pages

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: o Branch: main o Folder: / (root)

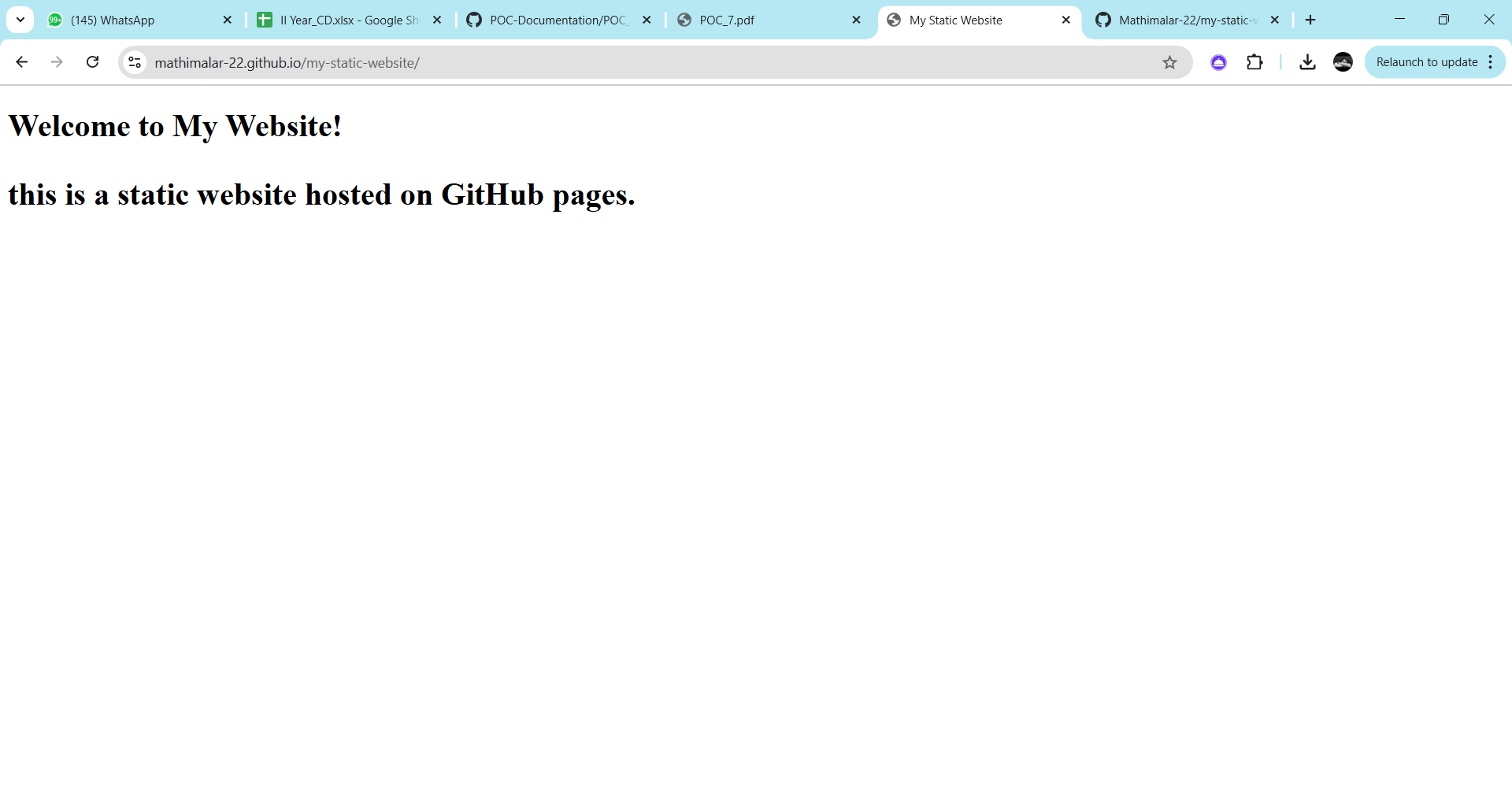
5. Click Save.



**Step 10:**

Access Your Website Wait a few minutes for GitHub Pages to deploy your site.

Visit your website



**Expected Outcome**

By following this approach, the website will be successfully hosted on GitHub Pages. Users will be able to access it via a live URL, and any future updates can be deployed by simply pushing changes to the GitHub repository. This method ensures a streamlined, version-controlled, and cost-free hosting solution for static websites.