**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

Set Up a Local Git Repository: Initialize a Git repository locally and version control your static website

Name: Samiya Afreen J Department: CSE



**Introduction**

Version control is a fundamental practice in software development that allows you to manage changes to your code over time. It provides a systematic way to track updates, collaborate with others, and revert to previous versions if needed. Git is one of the most widely used version control systems, known for its efficiency, flexibility, and distributed nature.

In this POC, we’ll initialize a local Git repository to version control your static website. By doing so, you’ll be able to track changes to your project files, experiment with new features in a controlled way, and easily share your project with others if needed. Setting up a Git repository is a critical step towards maintaining a structured and reliable workflow, especially for developers and teams working on collaborative projects.

**Overview**

Here’s what we will cover in this setup:

**1. Installing Git**: Ensure Git is installed on your system and properly configured.

**2. Creating a Local Repository**: Initialize a Git repository in the root folder of your static website

**3. Staging and Committing Files**: Add your project files to the staging area and commit them to the repository to save a snapshot of your work.

**4. Reviewing the** evolves. **Repository State**: Use Git commands to check the status of your repository and verify that everything is tracked properly.

**Objectives**

By the end of this POC, you will:

**1. Understand the Basics of Version Control**: Gain insight into the importance of Git for managing and tracking changes in your projects.

**2. Set Up a Git Repository**: Learn how to initialize a Git repository to version control your static website locally.

**3. Track Changes Effectively**: Understand how to stage and commit files to ensure every change is logged.

4. **Organize Your Project**: Maintain a clean and structured workflow for your static website, with the ability to roll back changes when needed.

5. **Prepare for Collaboration**: Lay the groundwork to share your repository and collaborate with others using Git when required

**Importance of Setting Up a Local Git Repository**

**Track Changes**: Git records all modifications, ensuring a clear history of your project.

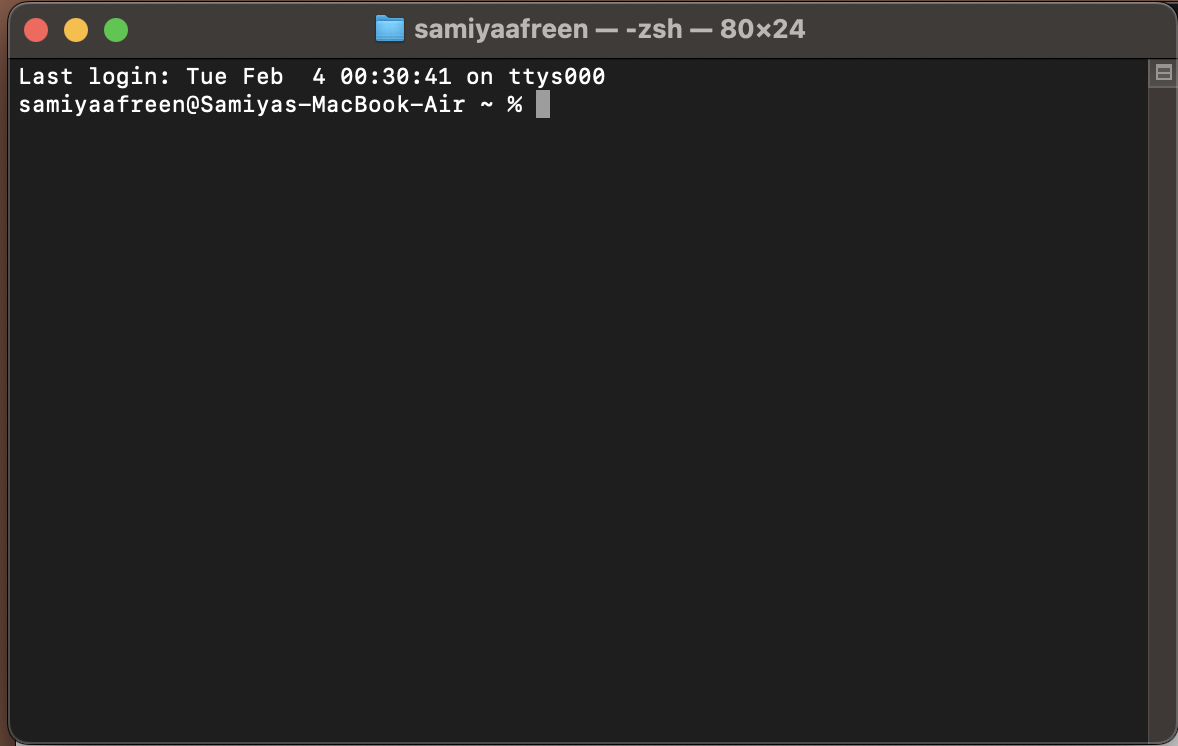
**Rollback**: Easily revert to previous versions to recover from mistakes.

**Collaboration**: Prepares your project for team work, enabling smooth integration of changes.

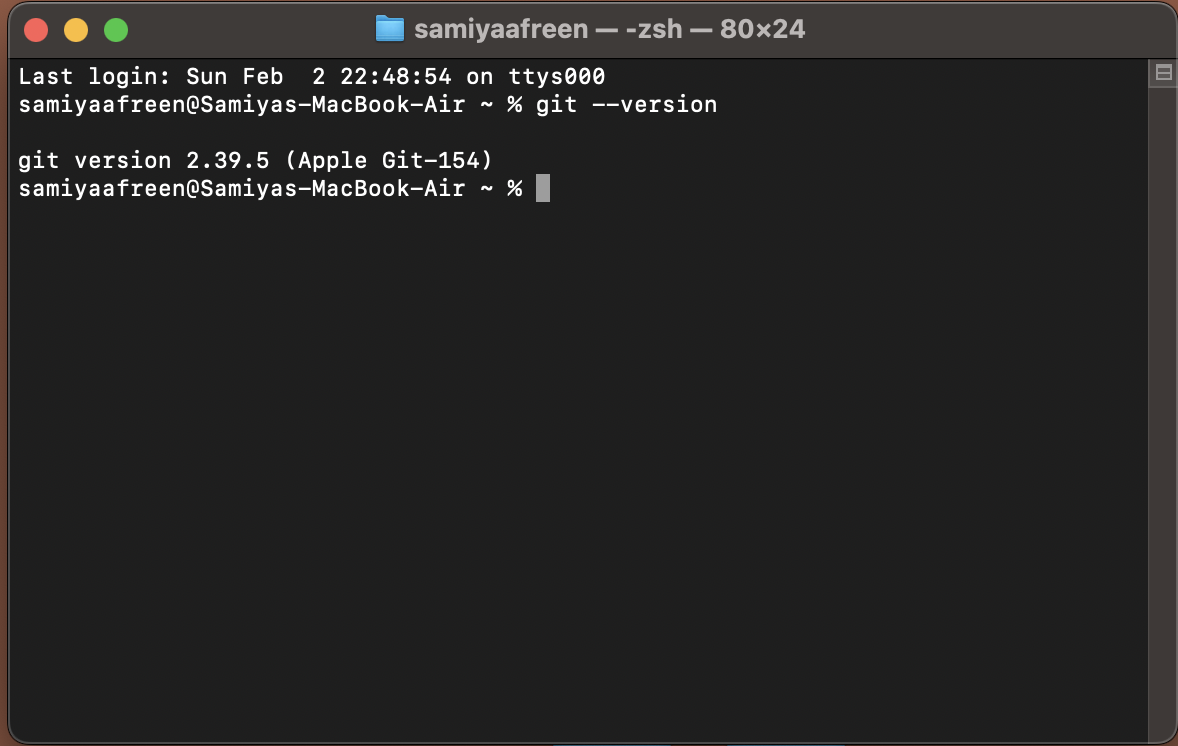
**Step-by-Step Overview**

Step 1:Install Git on Mac

Open Terminal on your Mac.

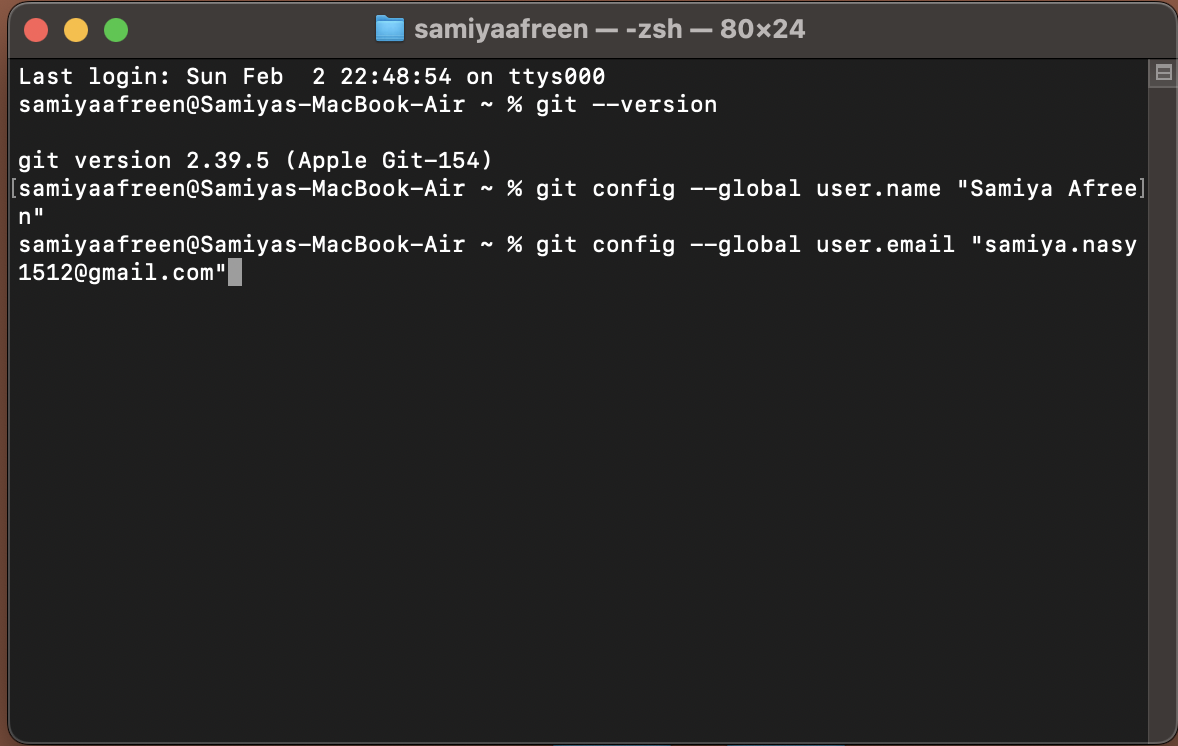


Type the following command to check if Git is already installed:

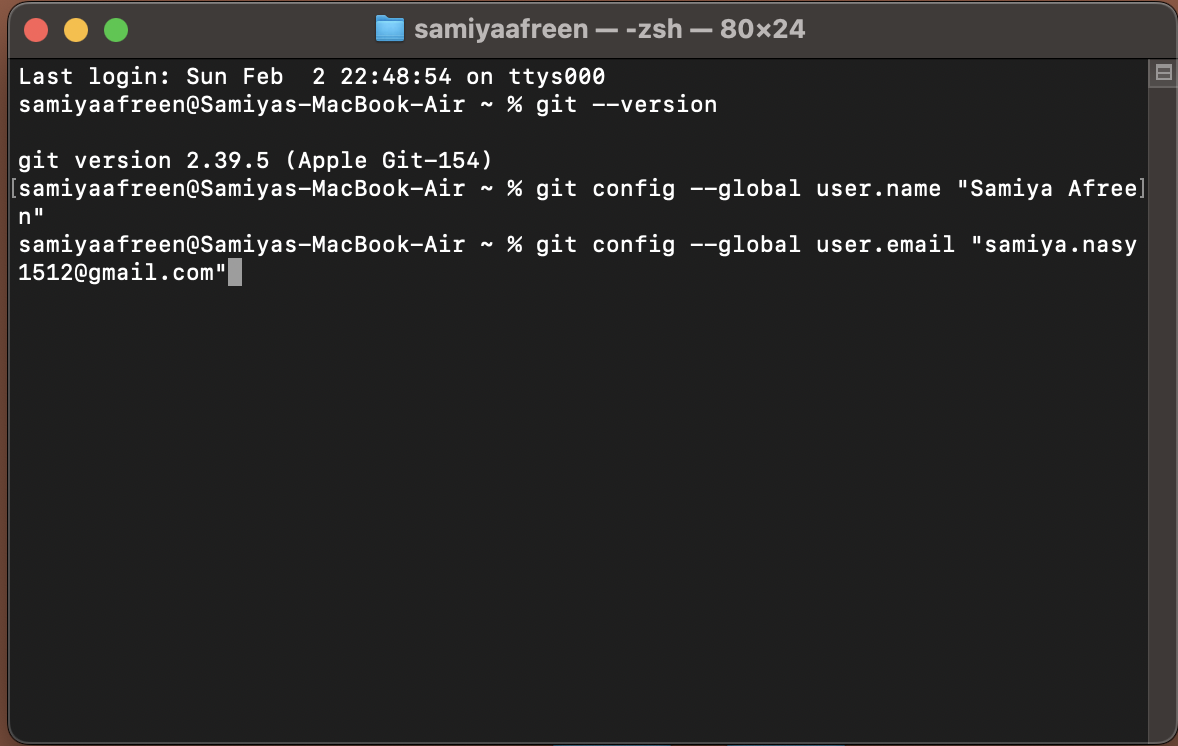


Step 2: Configure Git (One-time Setup)

Set your name:

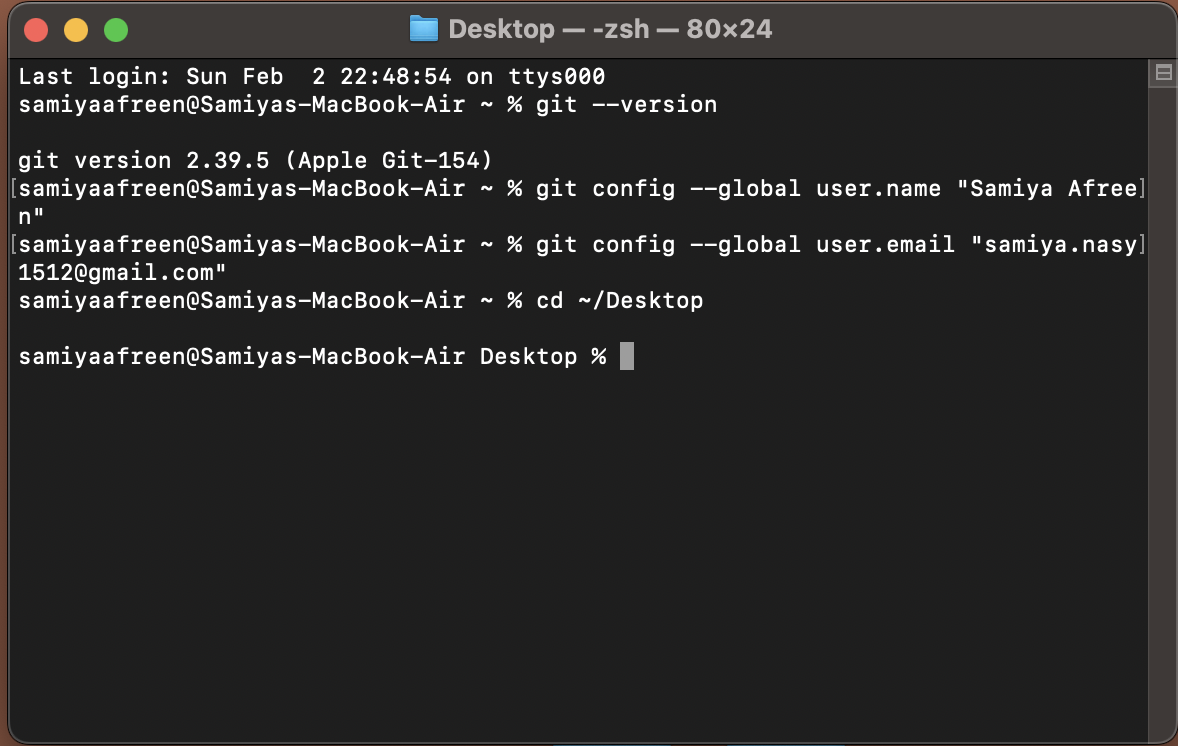


Set your email:

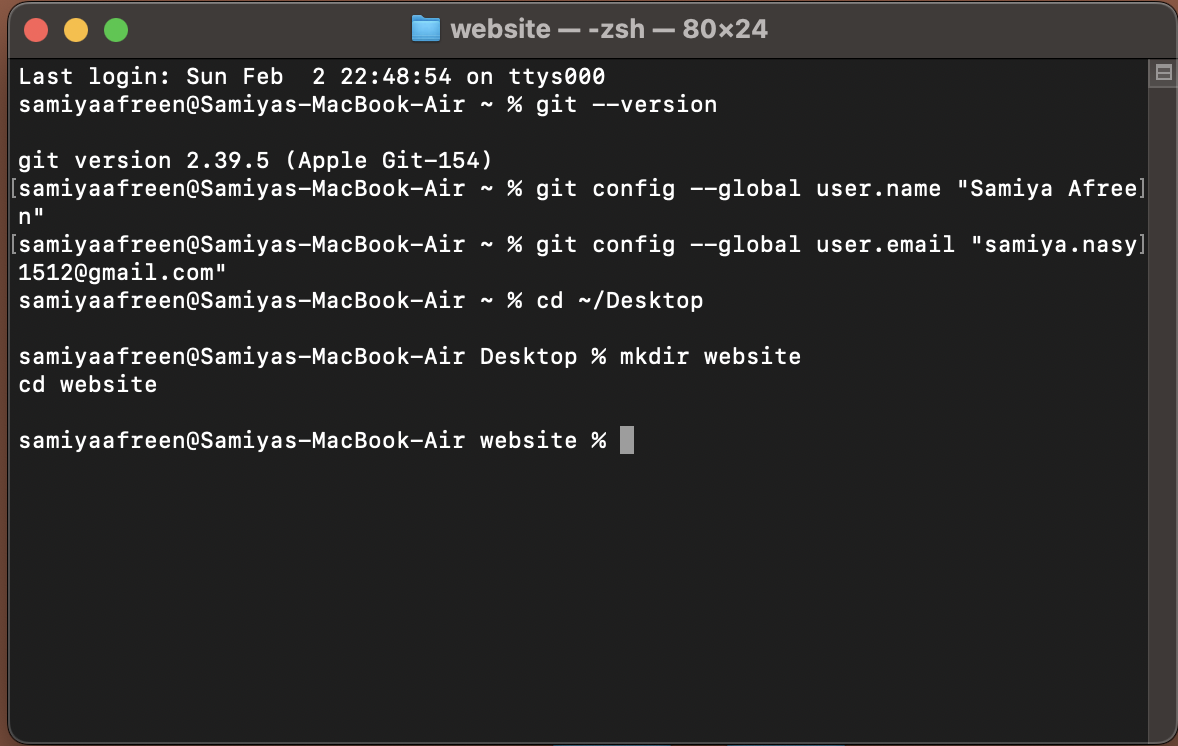


Step 3: Create a Project Folder

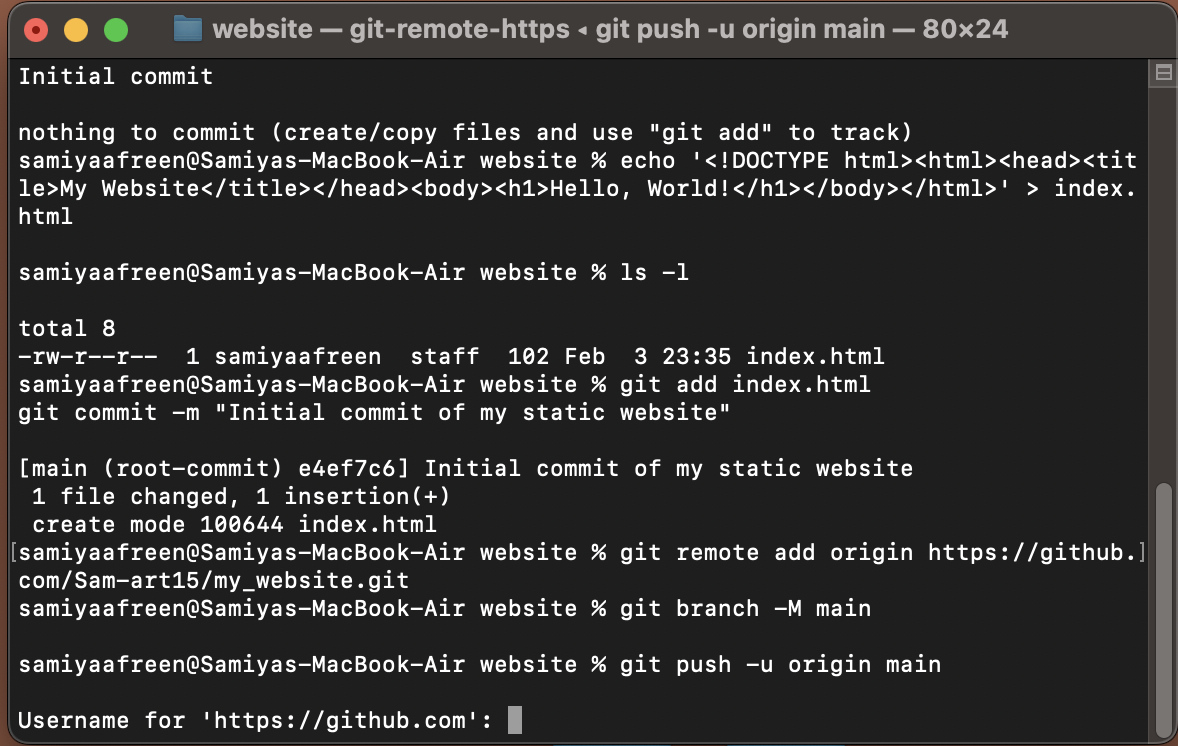
Open Terminal and navigate to your Desktop:



Create a new folder for your static website:



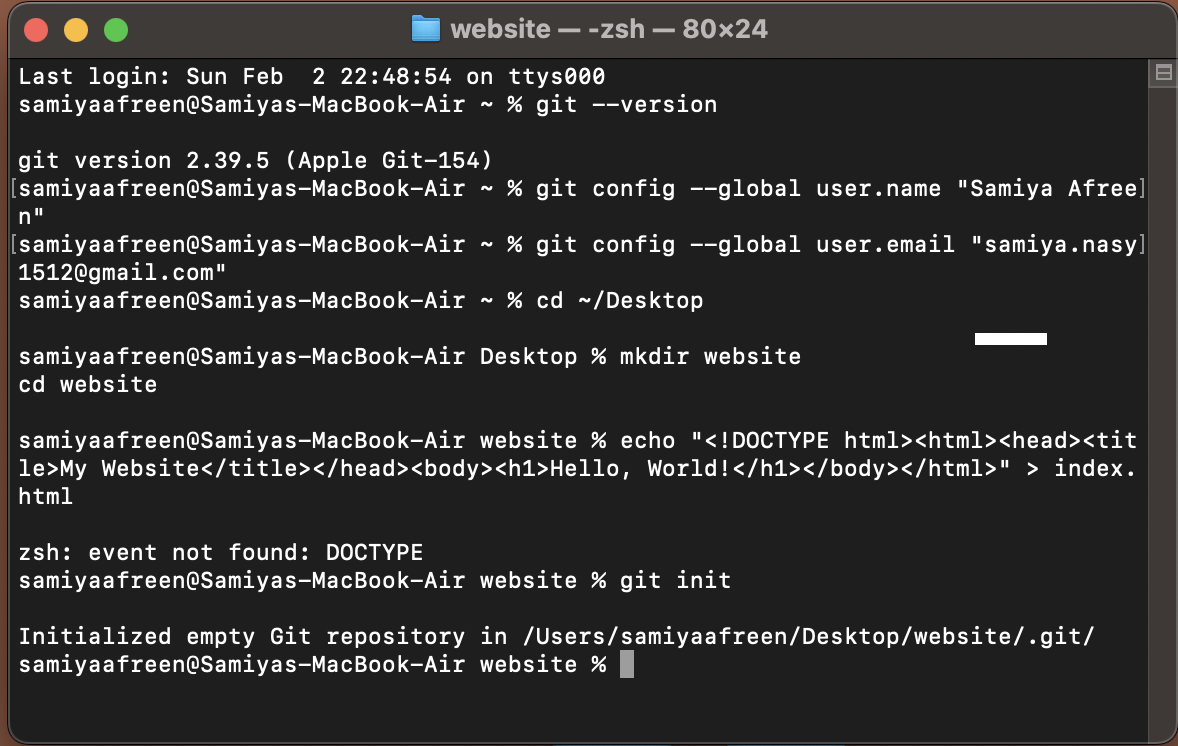
Create a simple HTML file:



Step 4: Initialize Git

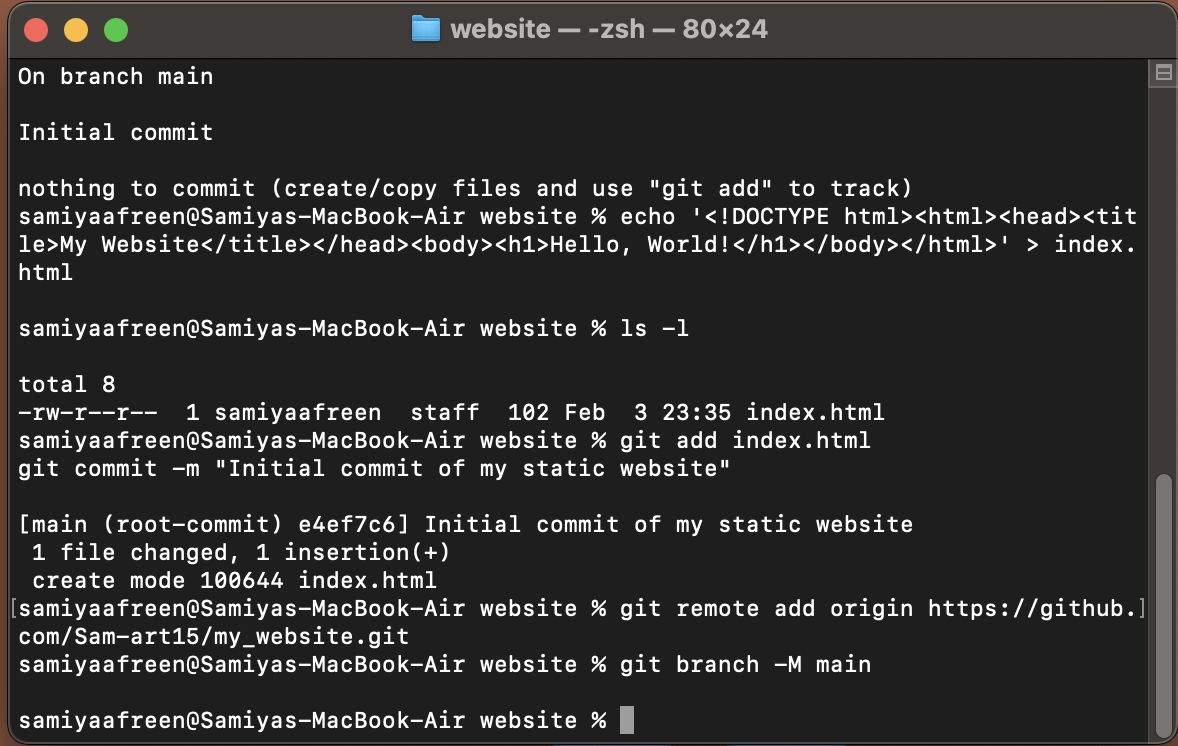
Inside the website folder, initialize Git:

This creates a .git folder to track changes.

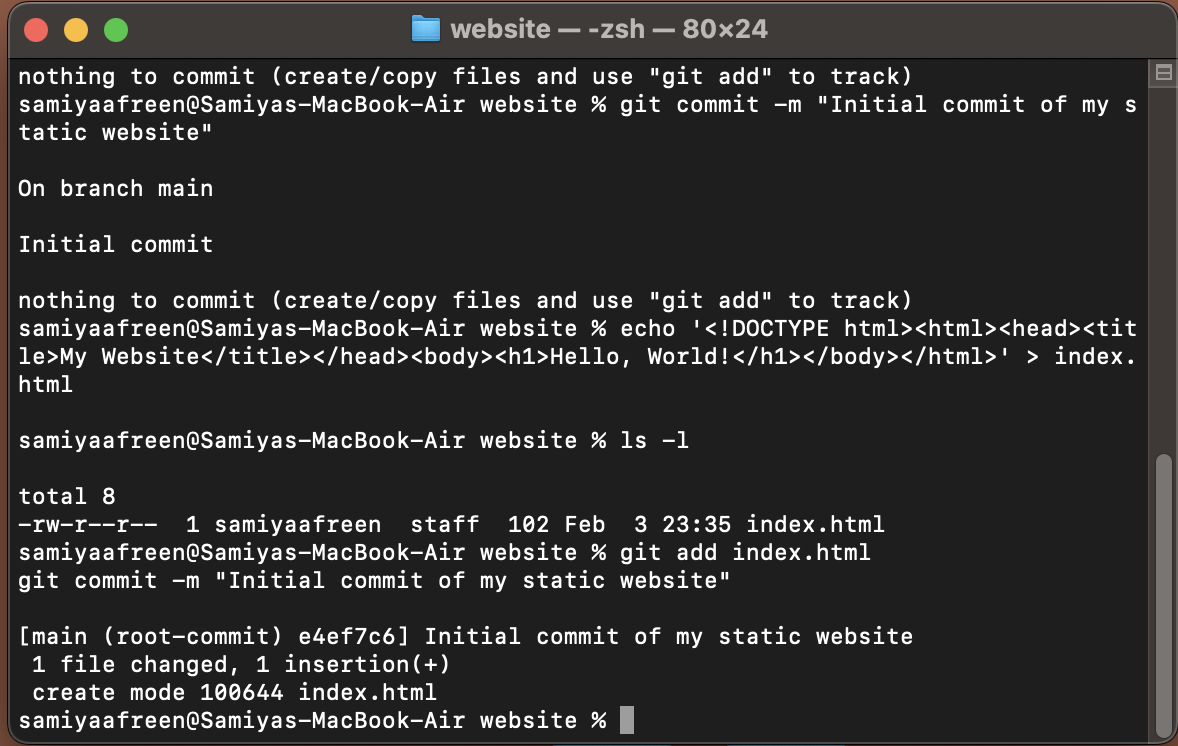


Step 5: Stage and Commit Files

Add all files to Git tracking:

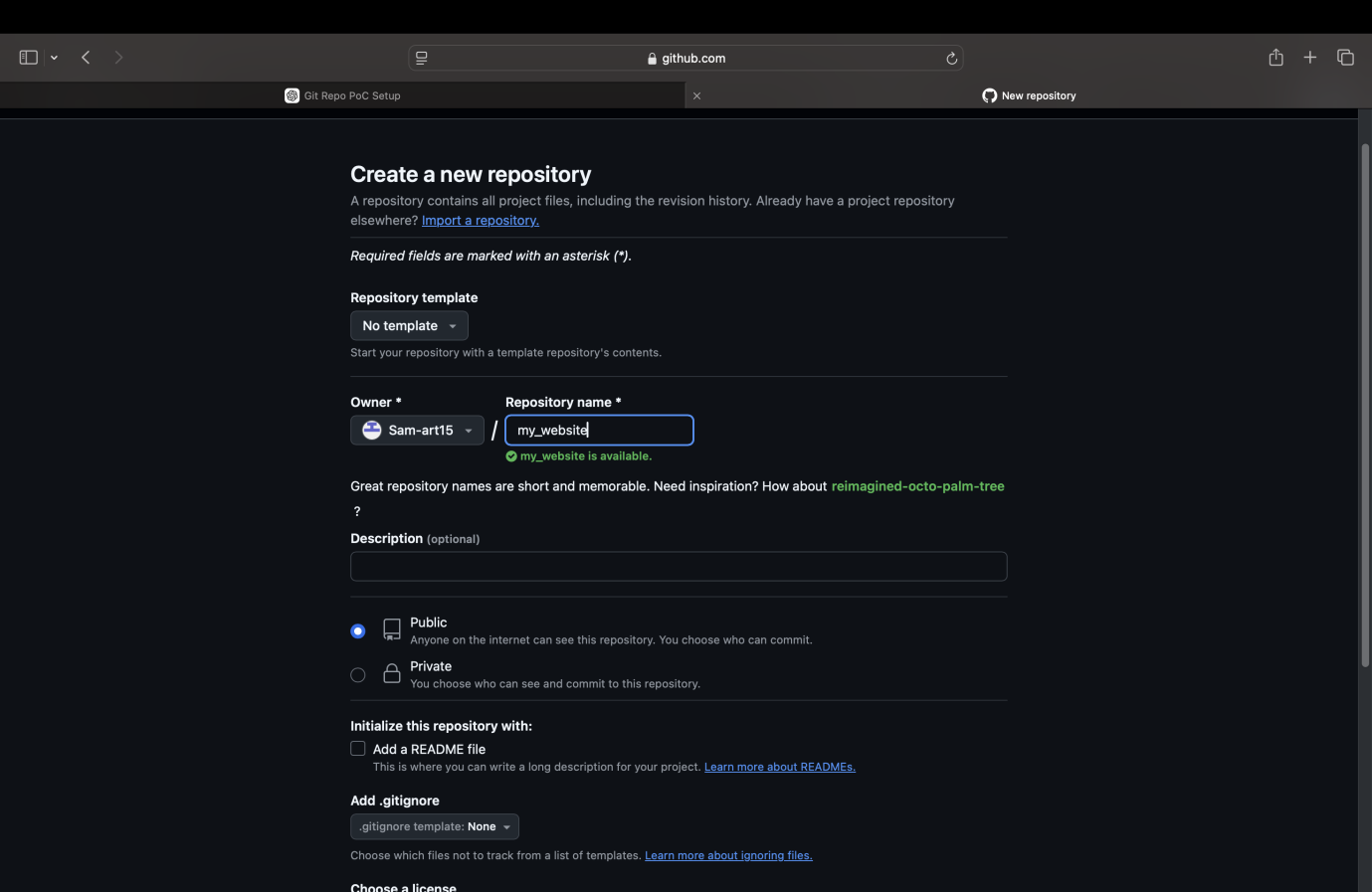


Commit the files with a message:



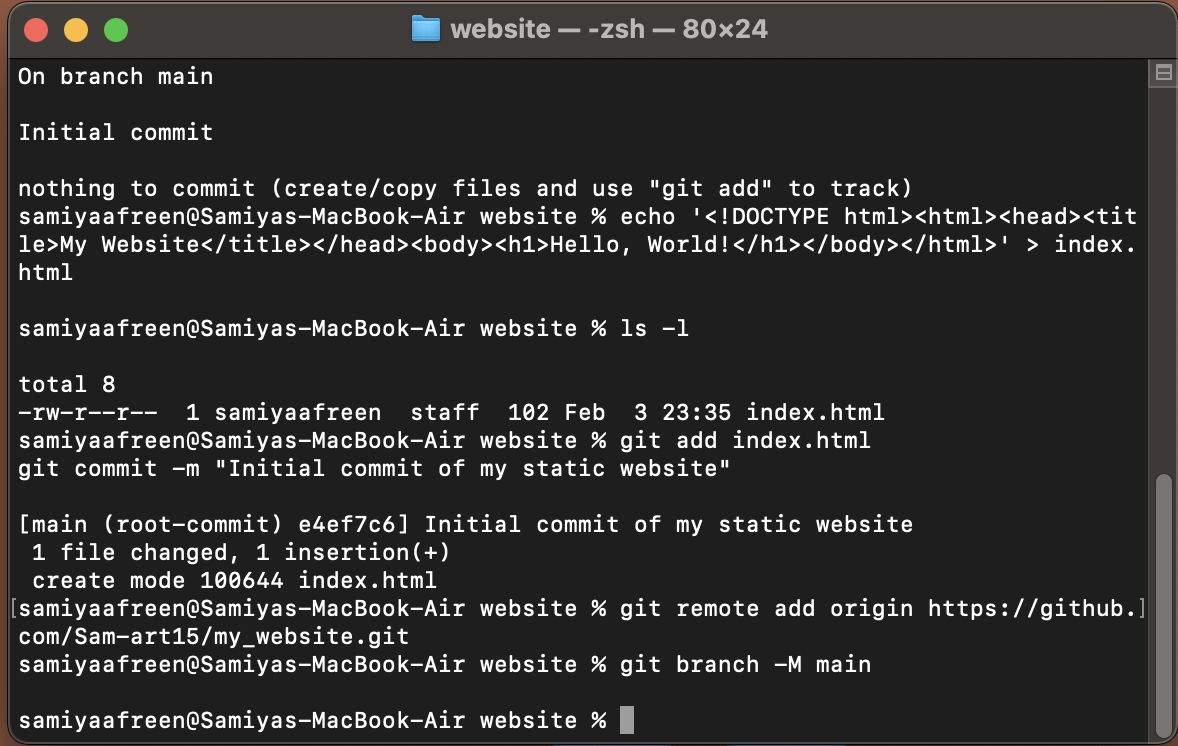
Step 6: Create a GitHub Repository

Go to [GitHub](https://github.com/" \t "/Users/samiyaafreen/Library/Containers/com.kingsoft.wpsoffice.mac.global/Data/Library/Application Support/Kingsoft/WPS Cloud Files/userdata/default/filecache/611623199/x/_new) and log in and click New Repository, name it, and create it.

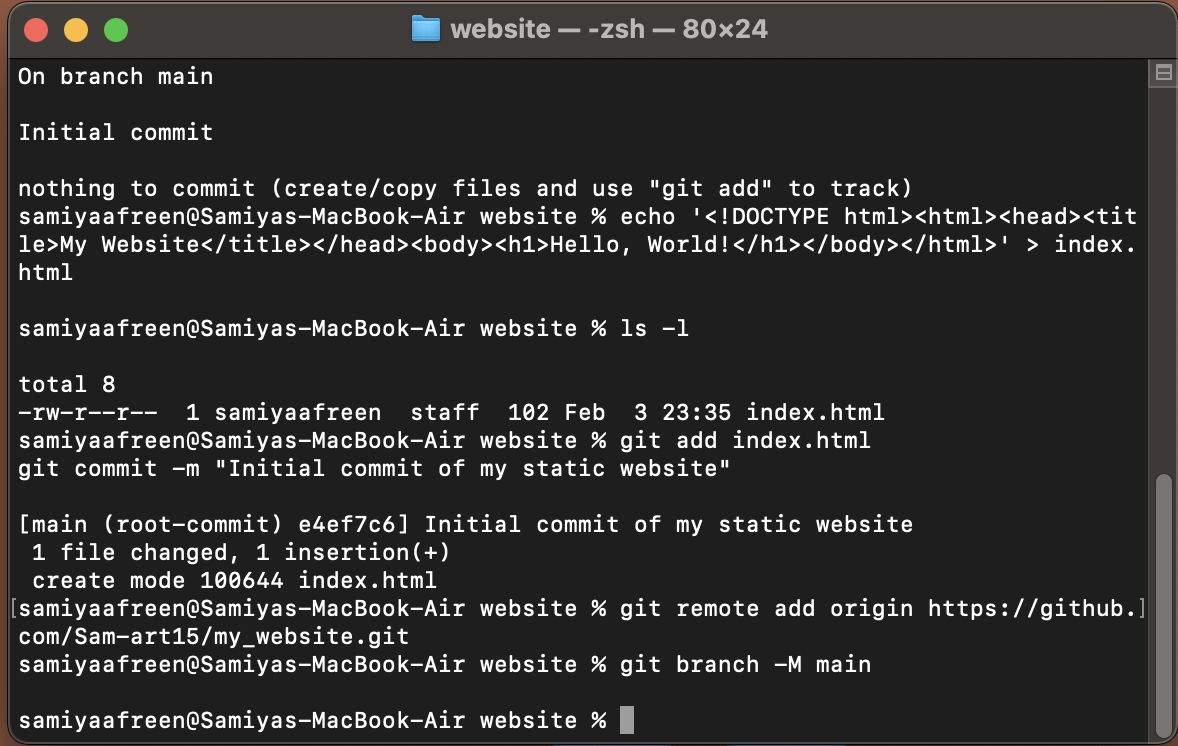


Step 7: Link Local Repository to GitHub

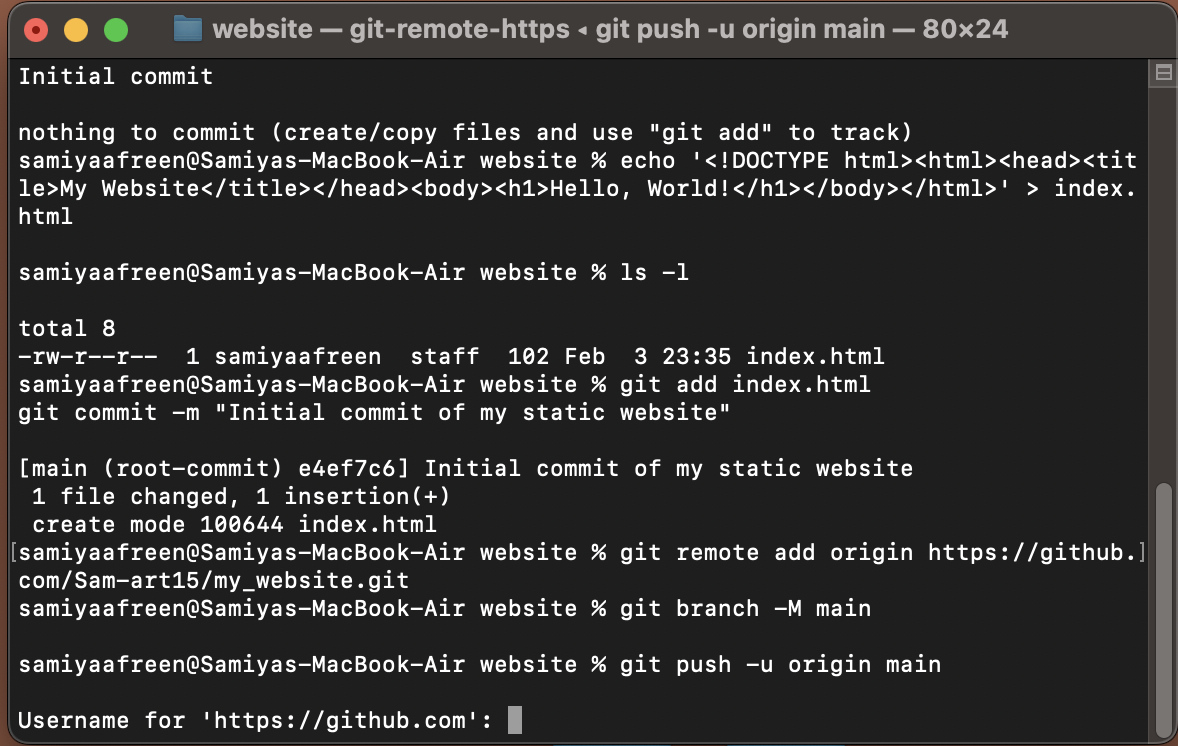
Add the remote repository URL:



Rename the main branch:

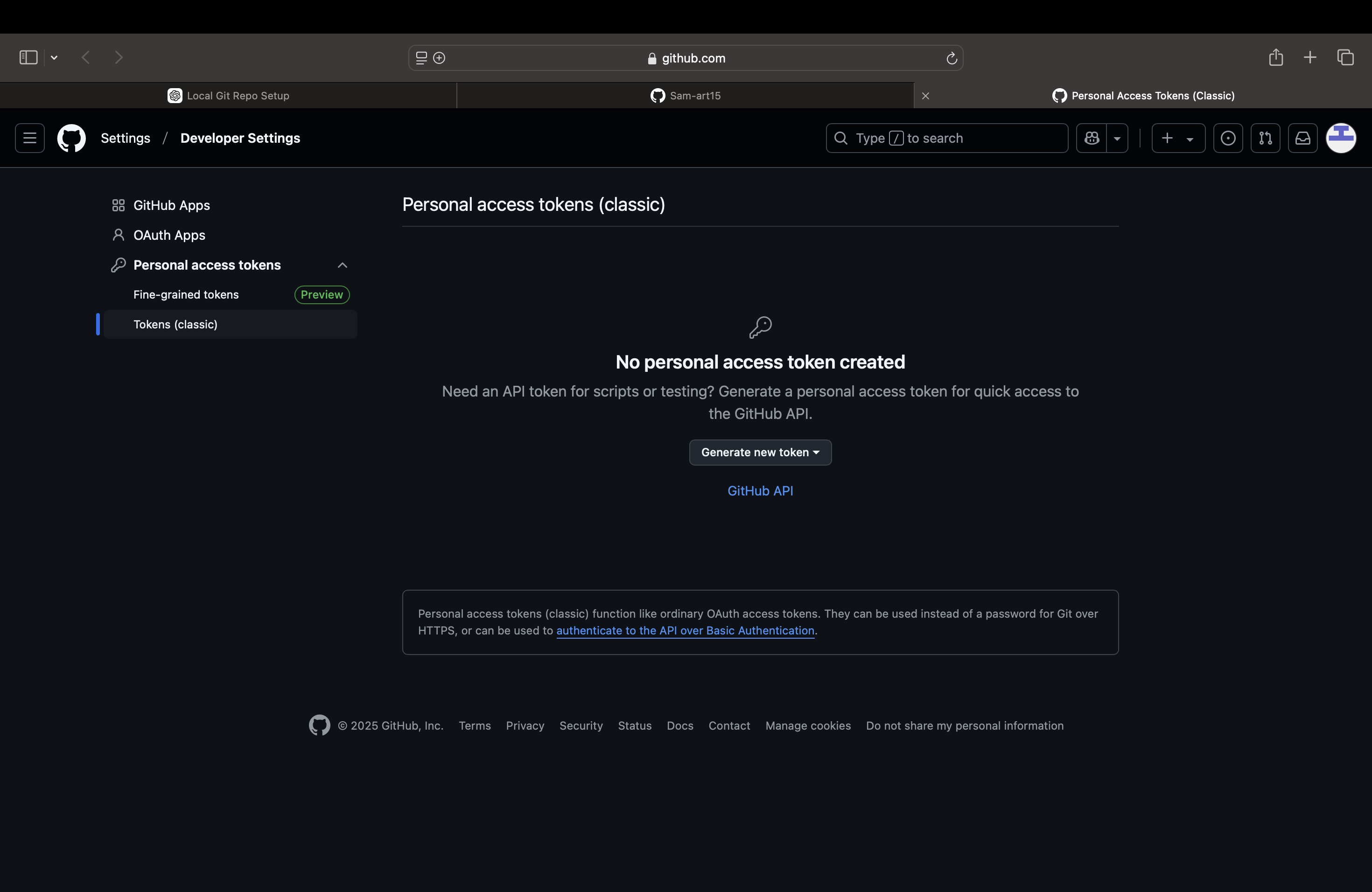


Push the local repository to GitHub:

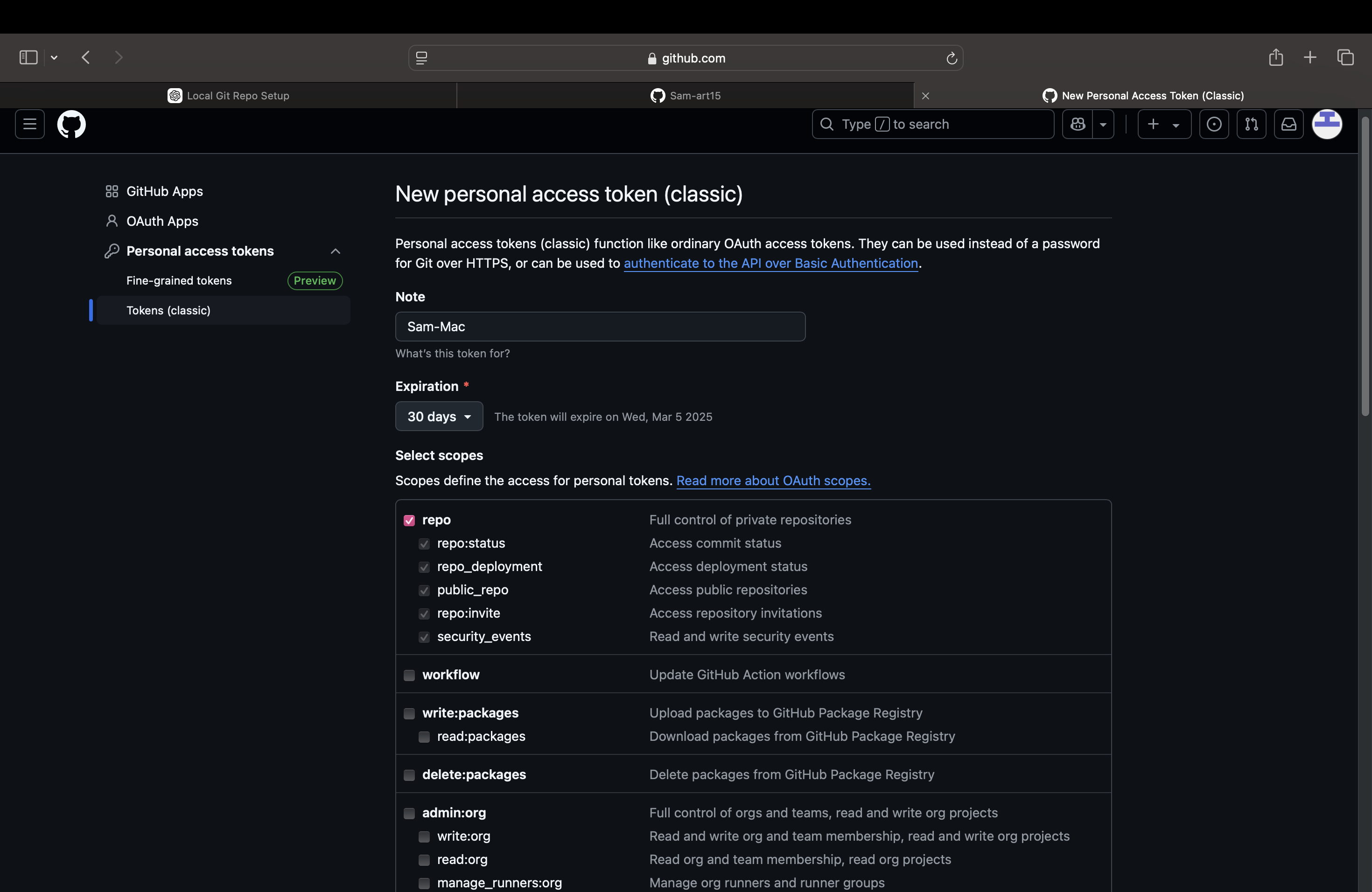


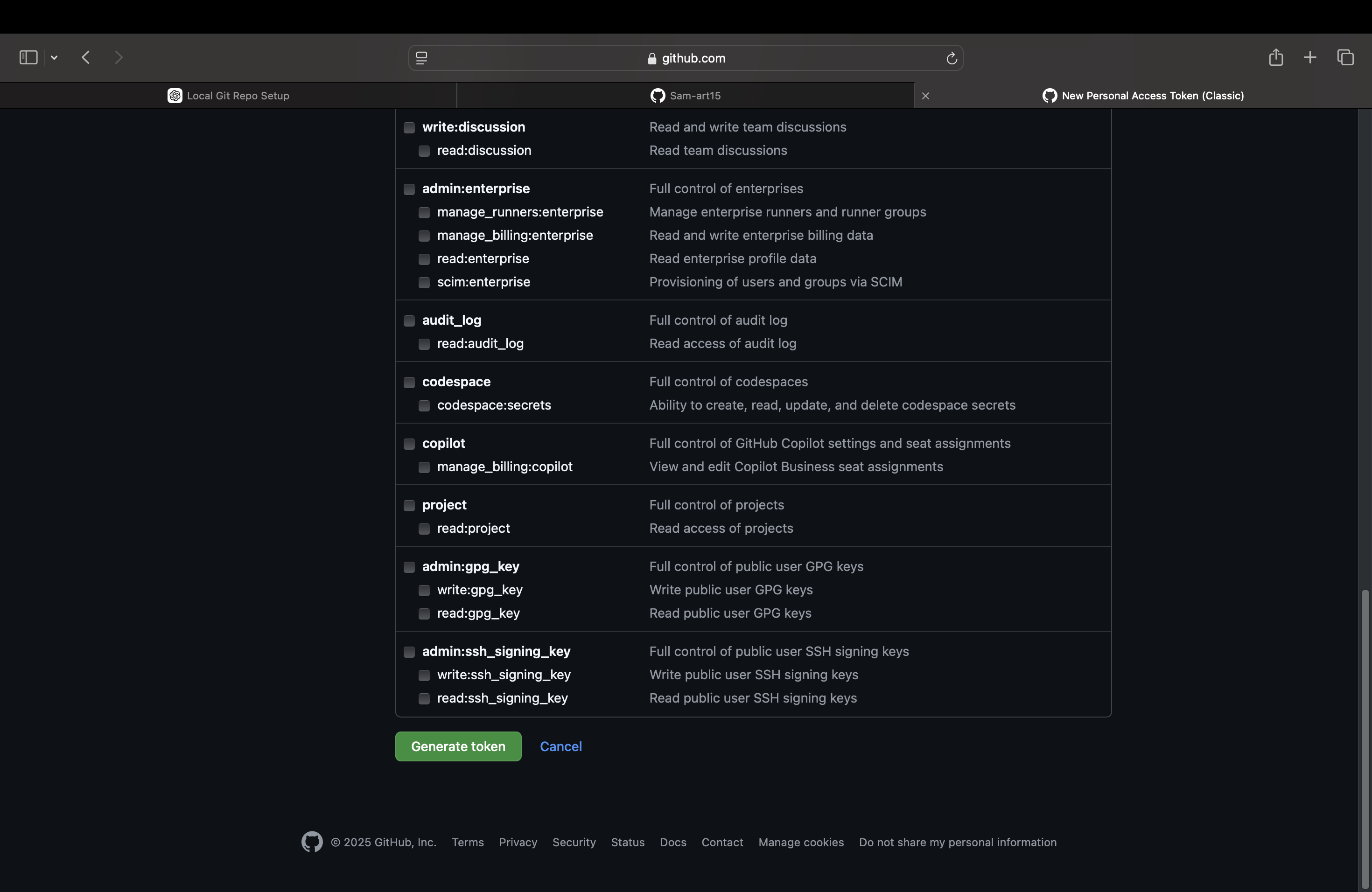
Step 8: Generate a GitHub Personal Access Token

Go to [GitHub's Personal Access Tokens page](https://github.com/settings/tokens" \t "/Users/samiyaafreen/Library/Containers/com.kingsoft.wpsoffice.mac.global/Data/Library/Application Support/Kingsoft/WPS Cloud Files/userdata/default/filecache/611623199/x/_new) and click on Generate new token.



Give the token a name and select the necessary scopes (usually repo is sufficient for most tasks).Click Generate token.Copy the generated token. Make sure to save it, as you won’t be able to see it again.

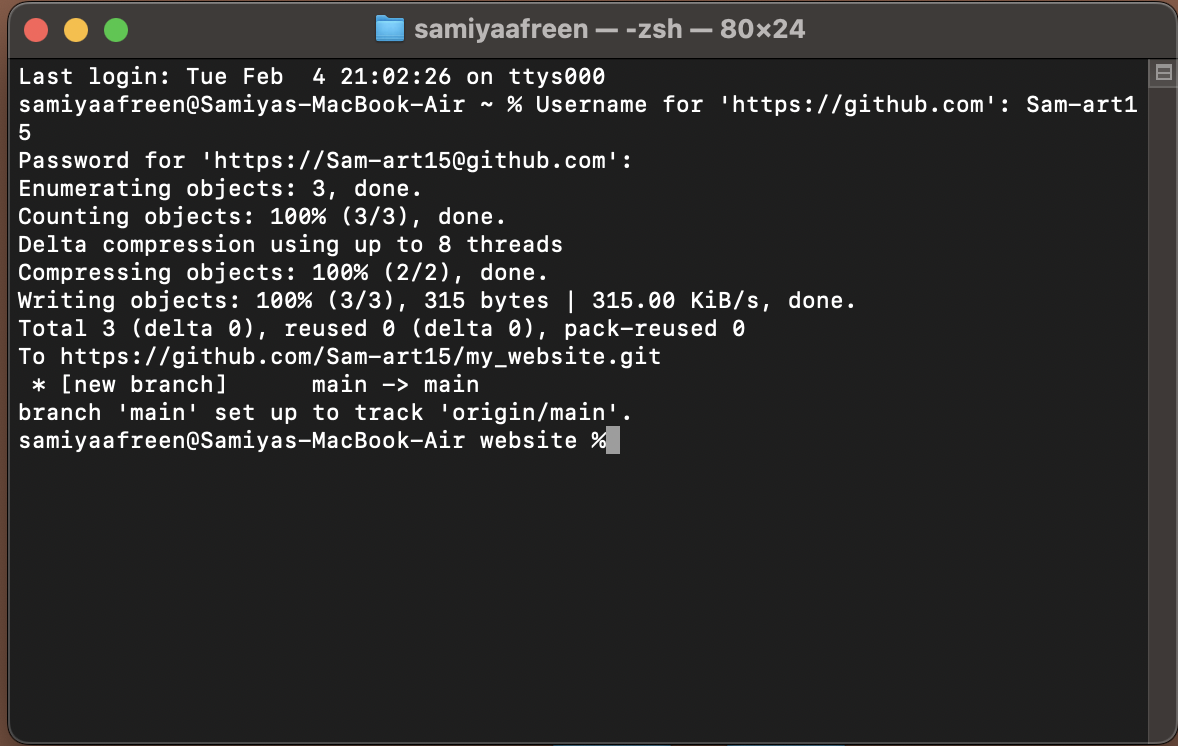




Use the Personal Access Token:

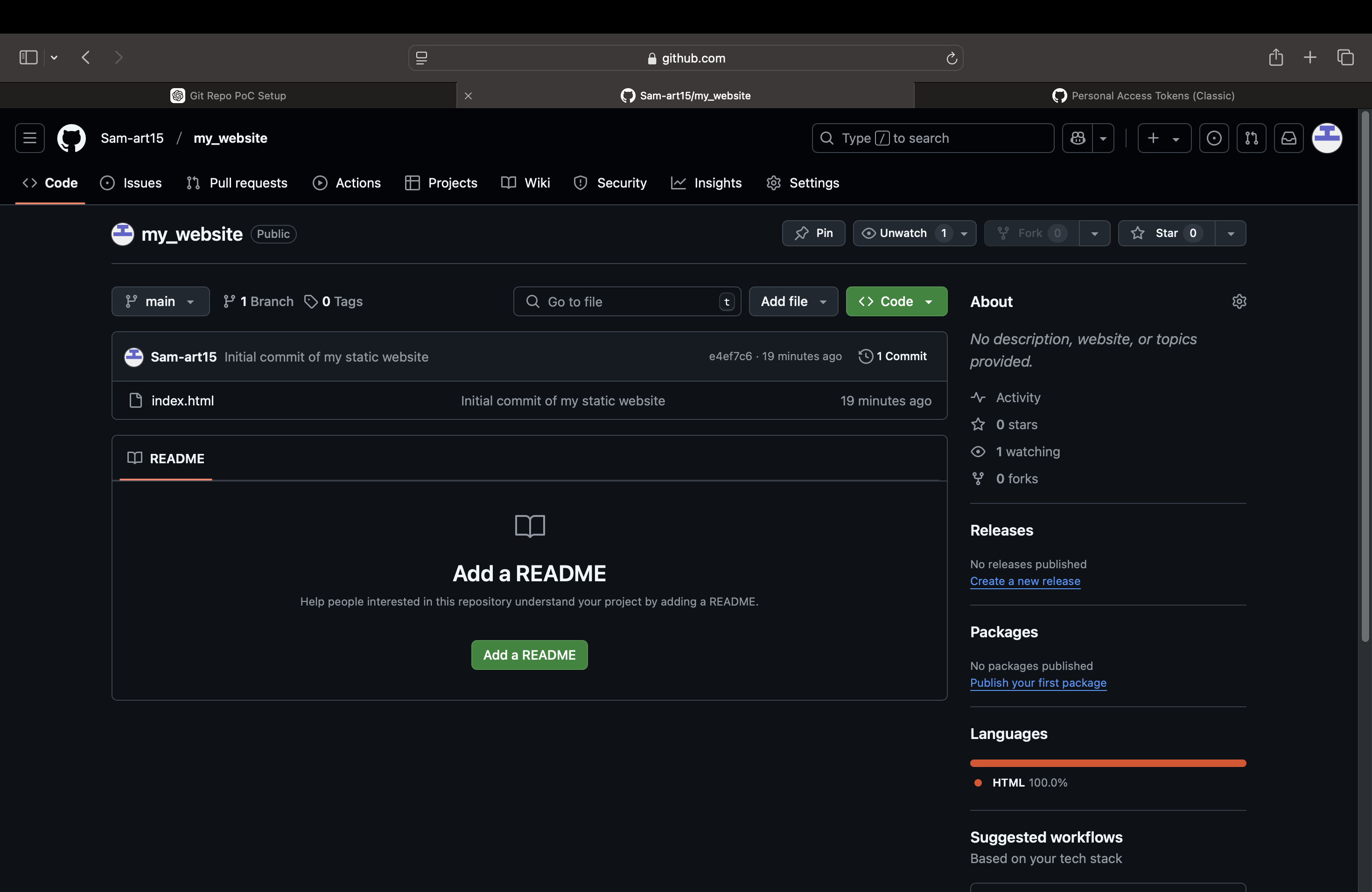
When prompted for your GitHub username, enter it. Instead of your password, paste the personal access token you just generated.

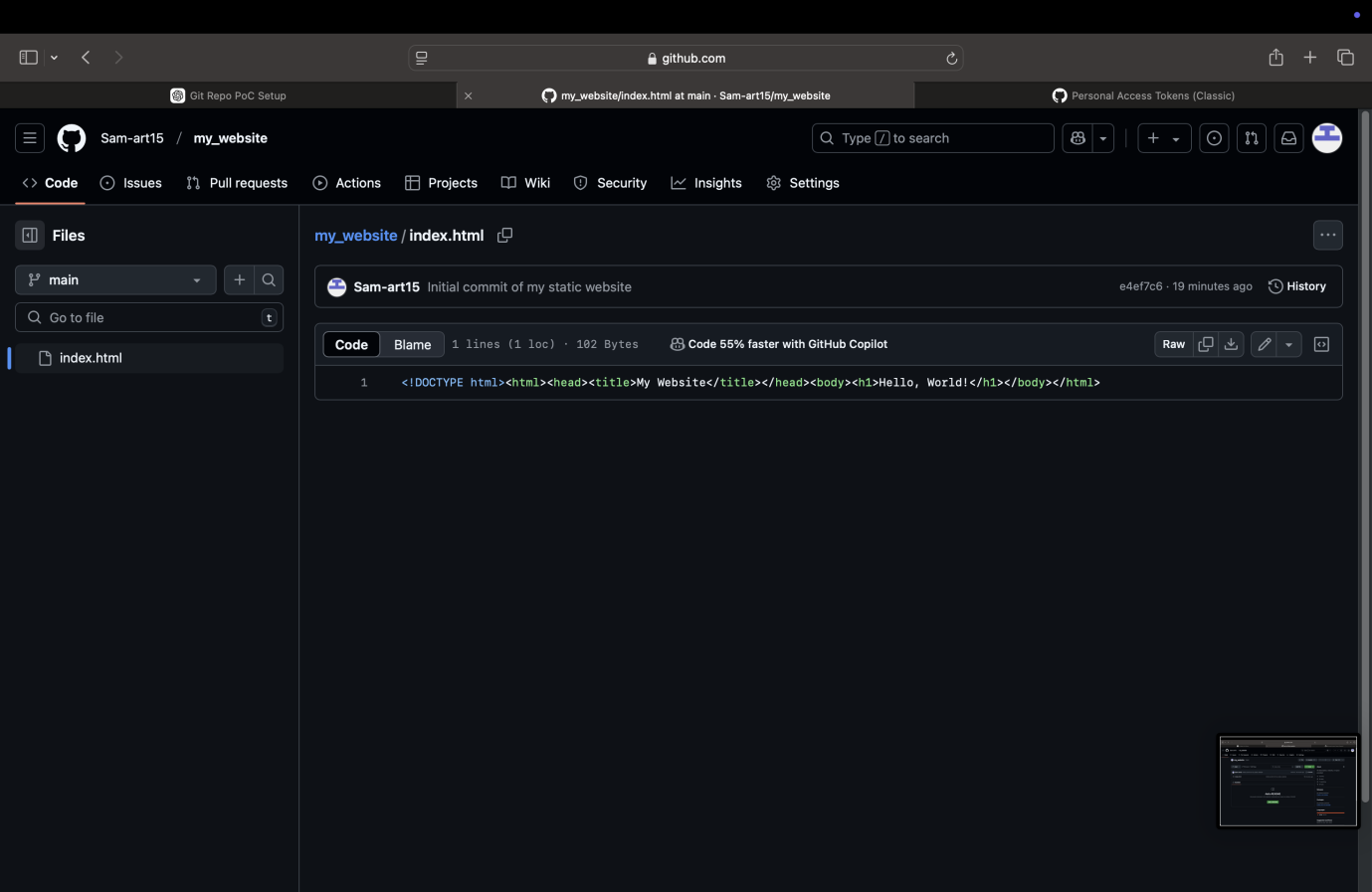
After entering the token, Git should authenticate successfully and push your changes to the remote repository.



Step 9: Verify the Push

To confirm the push was successful, you can check your GitHub repository page in a browser. You should see the index.html file in your repository.





**Expected Outcome**

By completing this PoC of setting up a local Git repository, you will:

1. Successfully initialize a Git repository in your local static website folder.

2. Track changes made to your website files (HTML, CSS, etc.) using Git version control.

3. Understand the basic Git commands (git init, git add, git commit) for version control.

4. Commit your changes locally with a descriptive commit message.

5. Gain hands-on experience with Git and how it helps manage and track website file changes.