**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: Priyadharshini S Department: CSE



**Introduction**

Version control is an essential practice in software development that helps manage code changes over time. It offers a structured approach to tracking modifications, collaborating with team members, and rolling back to earlier versions when necessary. Git, a widely popular version control system, is renowned for its speed, adaptability, and decentralized nature.

In this proof of concept (POC), we’ll set up a local Git repository to manage the version control of your static website. This setup will allow you to monitor changes in your project files, test new features in an organized manner, and share your work with others effortlessly. Initializing a Git repository is a vital step in establishing a systematic and dependable workflow, particularly for developers and teams engaged in collaborative projects.

**Overview**

Here’s what this setup will include:

1. **Installing Git**: Verify that Git is installed on your system and configured correctly.
2. **Initializing a Local Repository**: Set up a Git repository in the root directory of your static website.
3. **Staging and Committing Files**: Add project files to the staging area and commit them to create a saved snapshot of your progress.
4. **Reviewing Repository Changes**: Use Git commands to inspect the repository’s status and ensure all files are properly tracked.

**Objectives**

**By completing this POC, you will:**

1. **Learn the Fundamentals of Version Control: Understand the role of Git in managing and tracking project changes effectively.**
2. **Initialize a Git Repository: Acquire the skills to set up a local Git repository for version controlling your static website.**
3. **Track Project Updates: Master the process of staging and committing files to log every change systematically.**
4. **Streamline Your Workflow: Organize your static website project with a structured approach that allows for easy rollback of changes when necessary.**
5. **Get Ready for Team Collaboration: Build a solid foundation to share your repository and collaborate with others seamlessly using Git.**

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

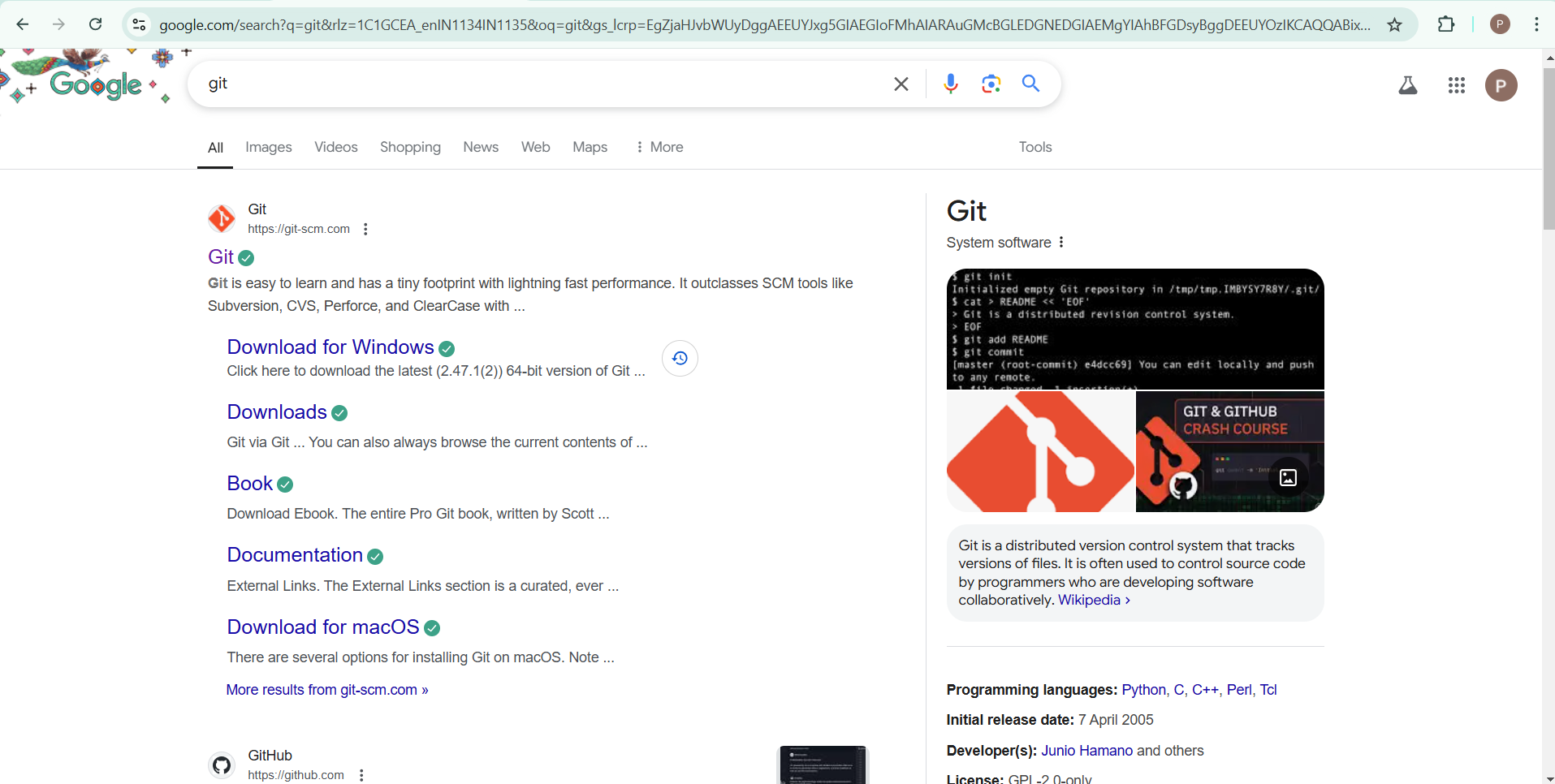
**Track Changes**: Git keeps a detailed record of every modification, providing a comprehensive history of your project.

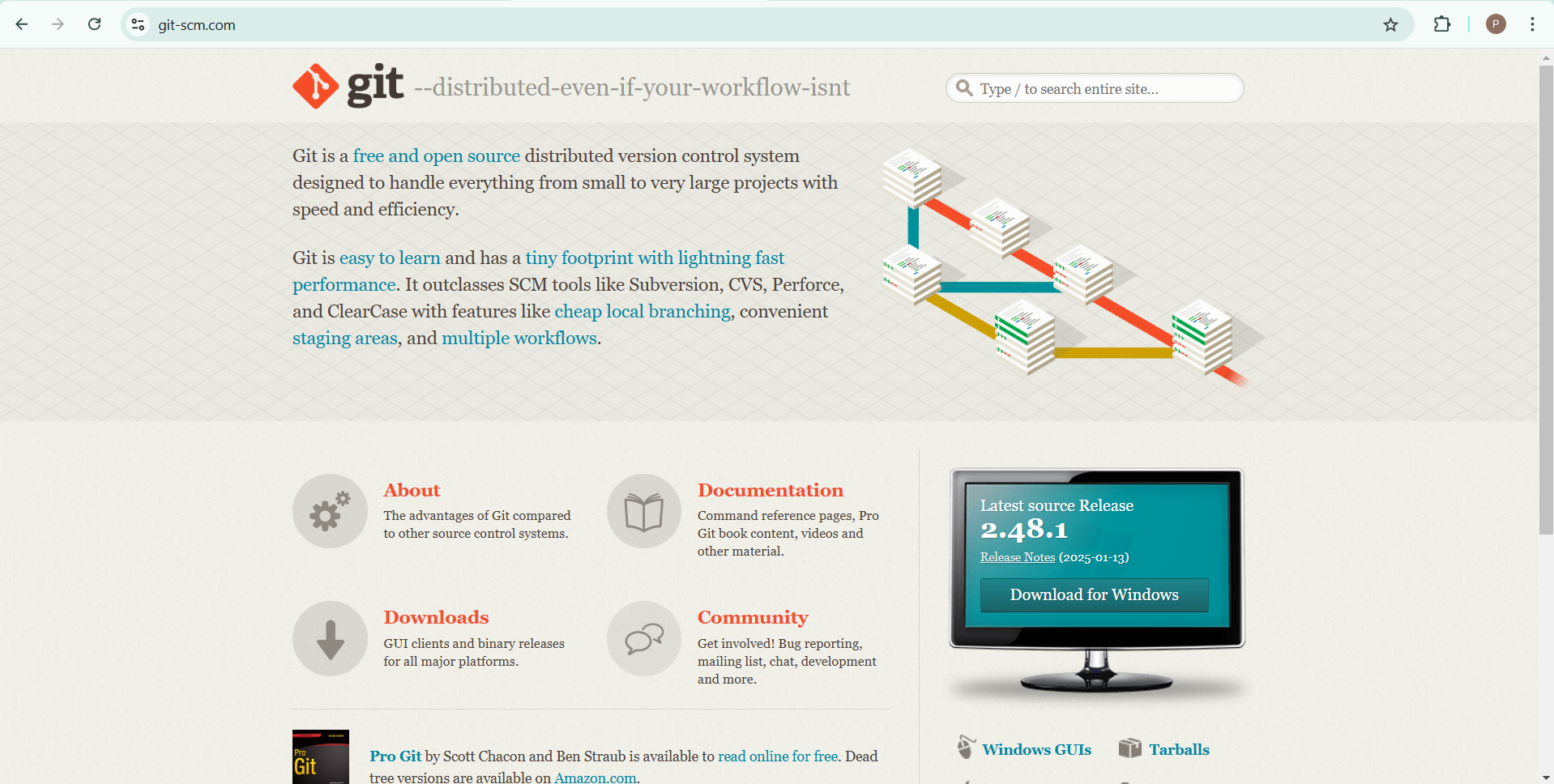
**Rollback**: Effortlessly restore previous versions to fix errors or undo unwanted changes.

**Collaboration**: Get your project ready for teamwork, allowing seamless integration of updates from multiple contributors

**Step-by-Step Overview**

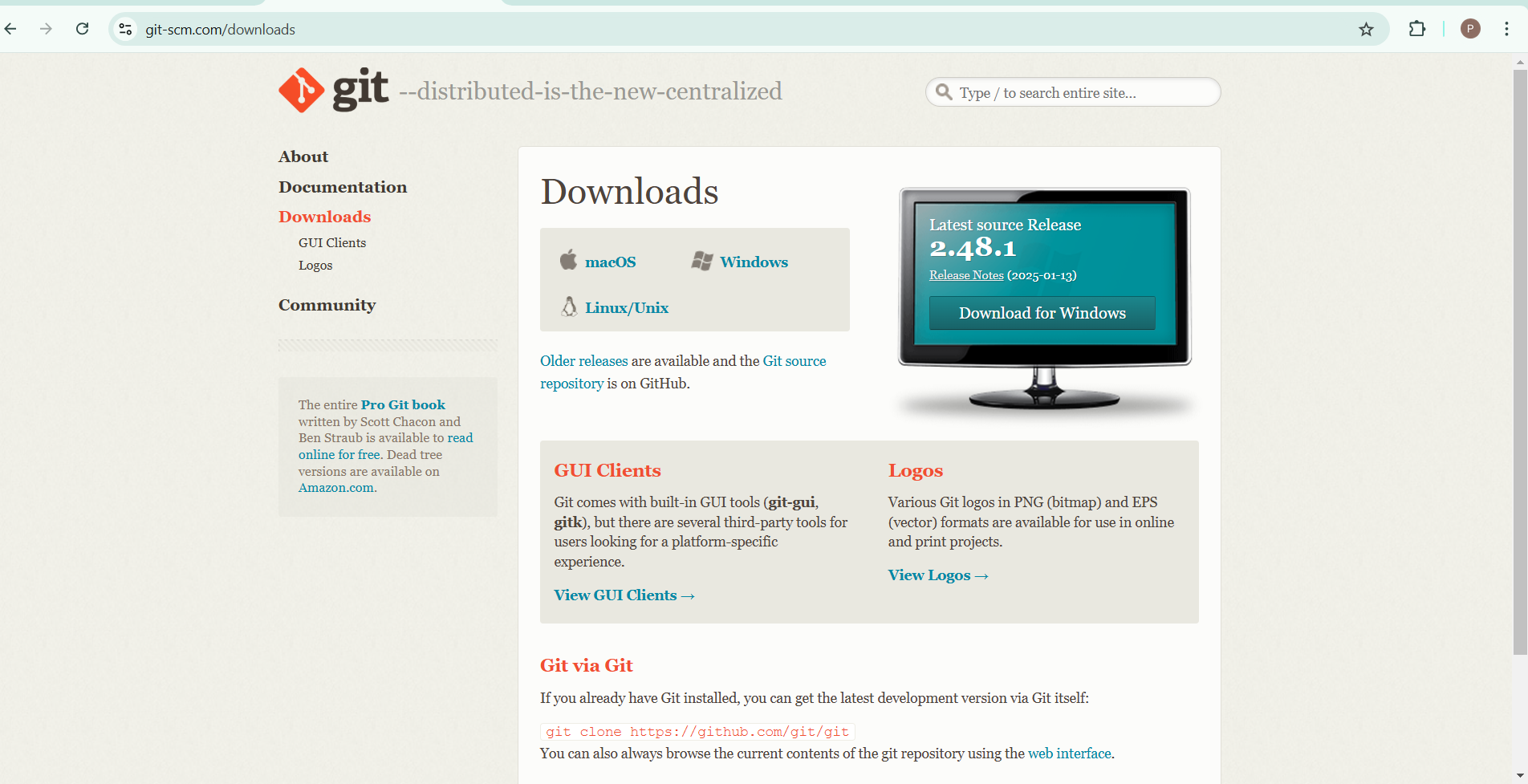
Step 1:

Search for "Git" in Chrome, download it, and click the "Downloads" option on the website. 



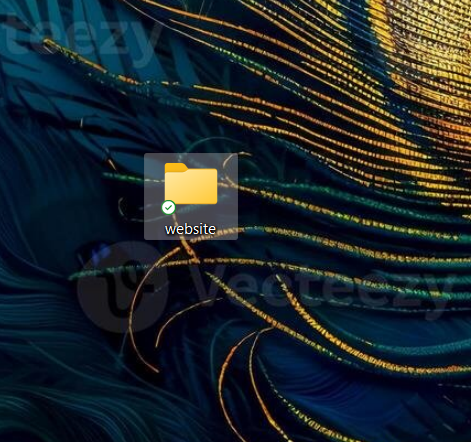
Step 2

Click the **Windows** option on the download page and follow the installation wizard.

Step 3

In your Desktop Create a folder named website for your static website

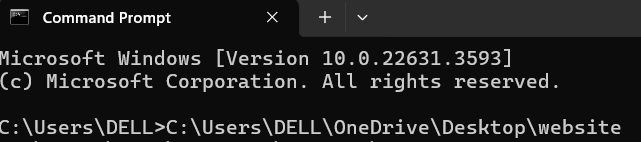
Inside that folder, create a simple HTML file named index.html. You can write some basic HTML





Step 5

Open the Command prompt and set the path to the folder named website we created



Step 6

Now, initialize Git by typing this command:

**git init**

This command will create a .git folder inside your project folder, which tells Git to start tracking your files.



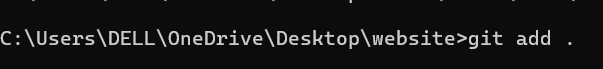
Step 7

Next, we need to tell Git to start tracking your website files.

To tell Git which files to track, use the git add command. If you want to track all the files in your folder, type

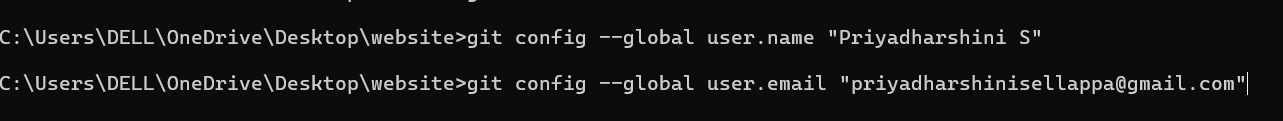
**git add .**

This command adds all the files to Git’s tracking system.



Step 8

Set Up Your Name and Email Globally Git doesn’t know who is making the commit because you haven’t configured your name and email yet. Git uses this information to track who made the changes.



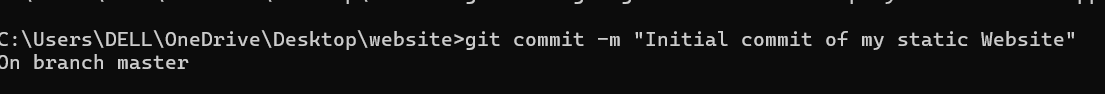
Step 9

Now, we need to save these changes in Git. When you "commit" changes, Git takes a snapshot of your files.

Type the following command to commit your changes:

**git commit -m "Initial commit of my static website"**

The -m flag allows you to add a message about your changes. In this case, we’re saying this is the "initial commit," meaning the first time we’re saving our work.



Step 10

**Create a New Repository**:

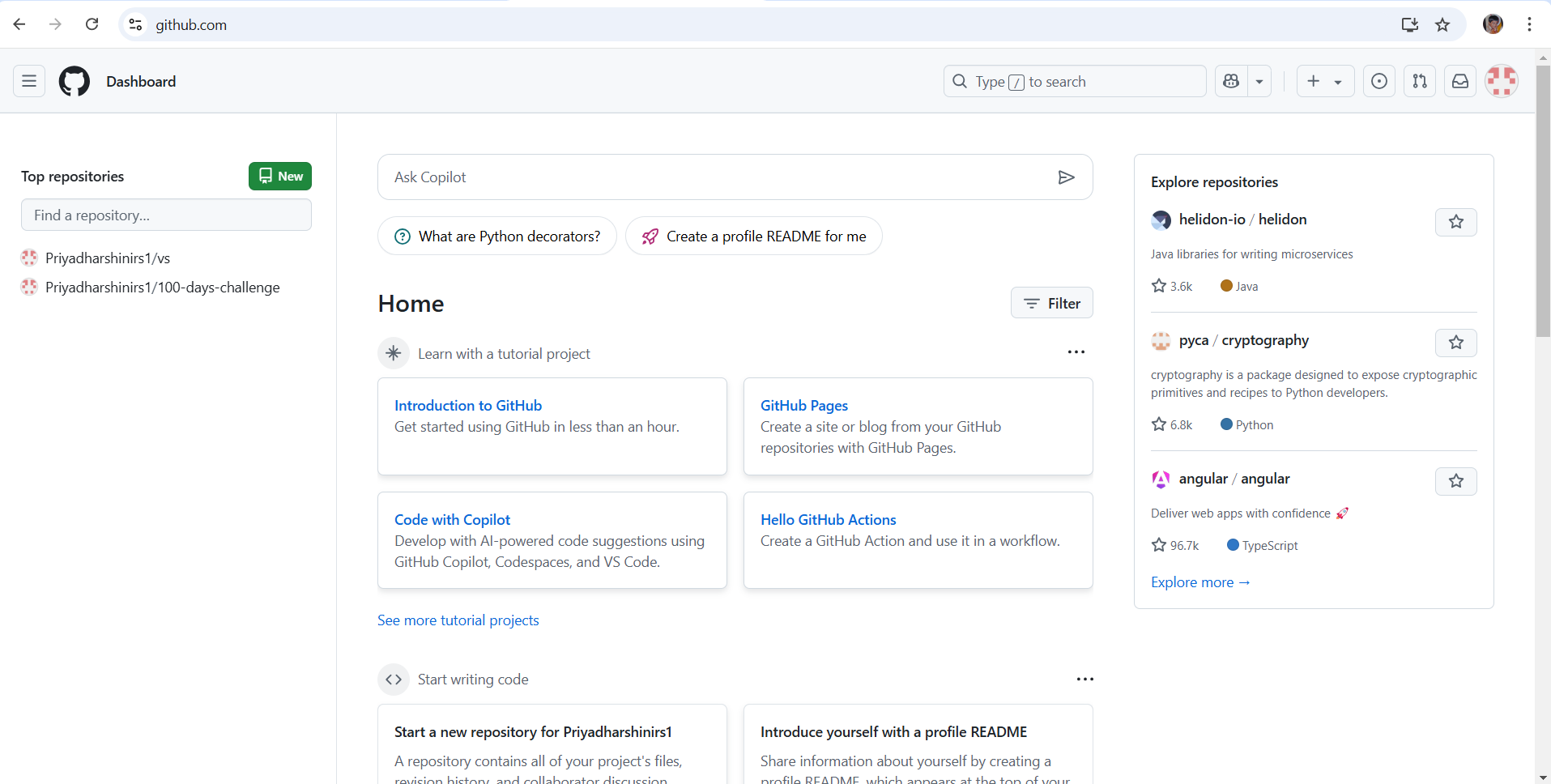
Once you're logged in, click the green **"New"** button on the top- right of your GitHub homepage to create a new repository.

Give your repository a name, for example, my-website.

Leave the other settings as default, and click **"Create repository"**.

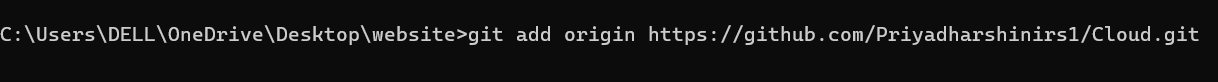
Step 11

**Add the Remote Repository URL to Your Local Repository**:



**git remote add origin https://github.com/yourusername/my-website.git**

Replace yourusername with your GitHub username and my-website with the name of your GitHub repository.

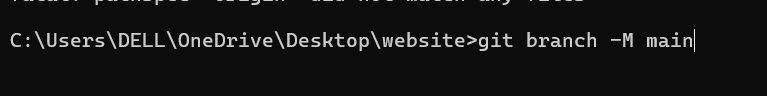


Step 12

The **git branch -M** main command is used to **rename the current branch** to main. Here's what it does:

**-M**: This flag forces the renaming, even if a branch named main already exists. It will overwrite the existing main branch.

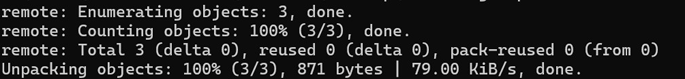
**main**: This is the new name for the current branch.



Step 13

The command git **push -u origin main** is used to **push your local main branch to the remote repository (origin)** and set it as the upstream branch





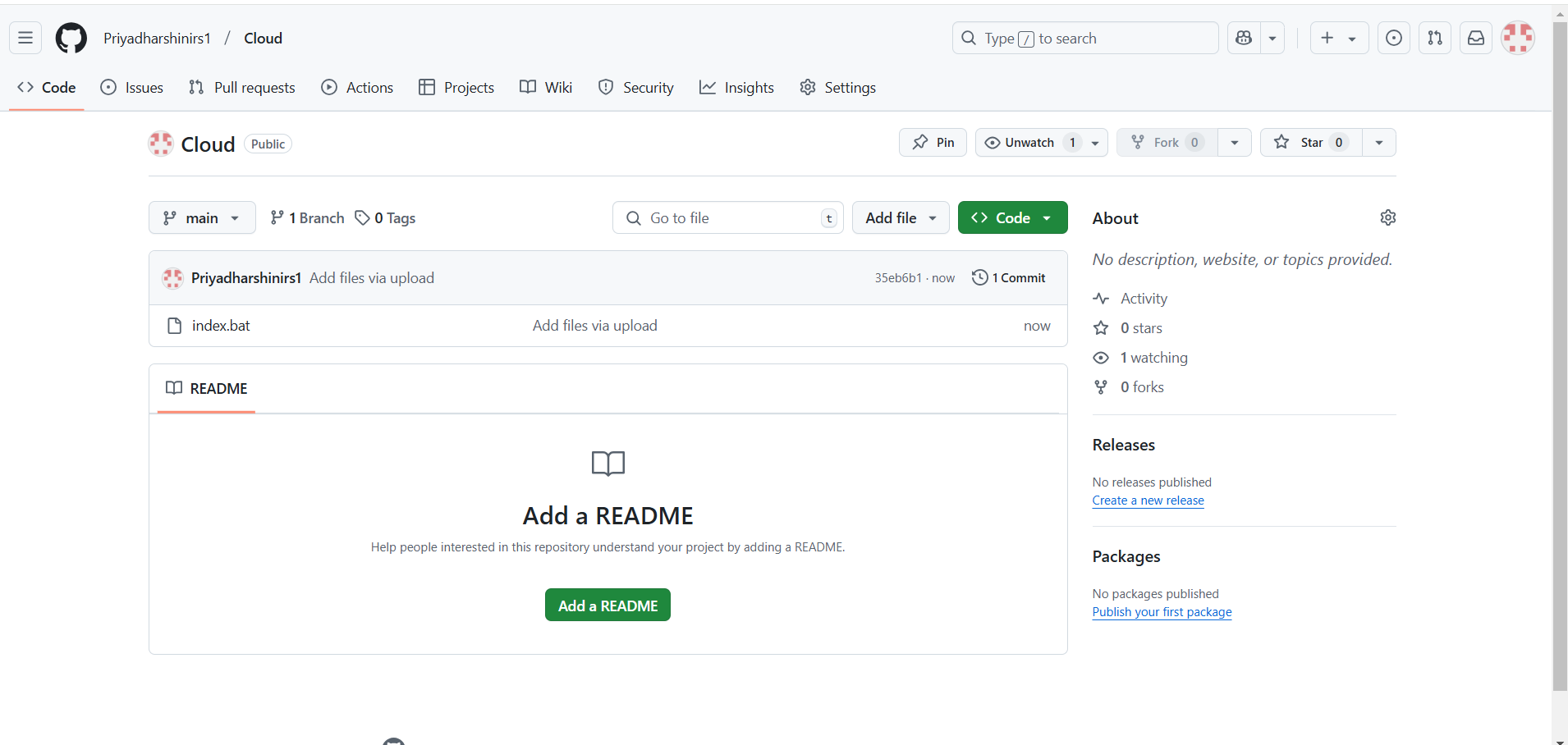
Step 14

Verify Your Files on GitHub

Go to your GitHub Repository:

Open your web browser and navigate to your GitHub repository (e.g., https://github.com/yourusername/my-website).

You should see your website files there!



**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.