**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: Kirthika S Department: CSE



**Introduction**

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.

**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.

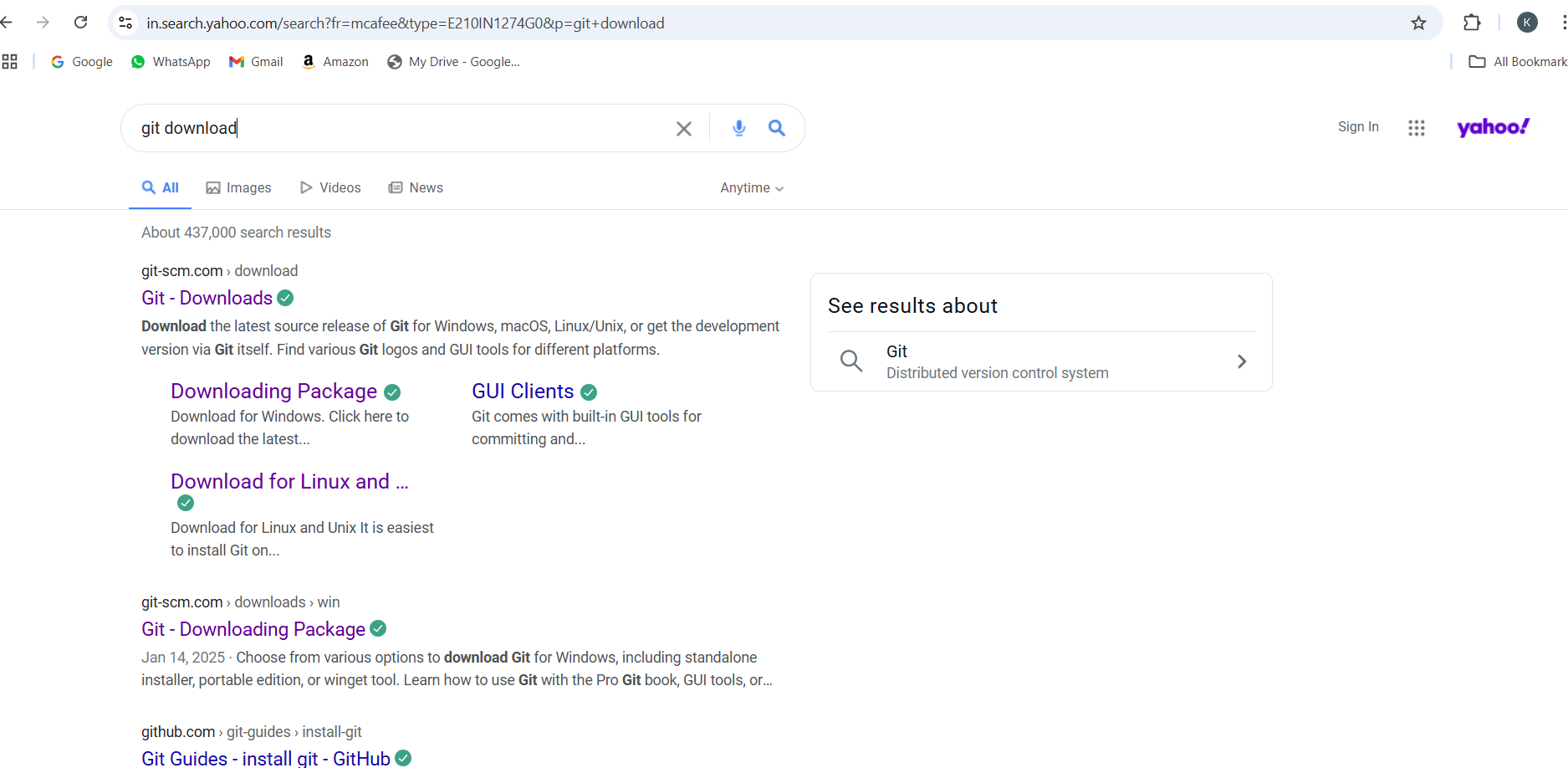
**Rollback**: Easily revert to previous versions to recover from mistakes.

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

**Step-by-Step Overview**

Step 1:

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



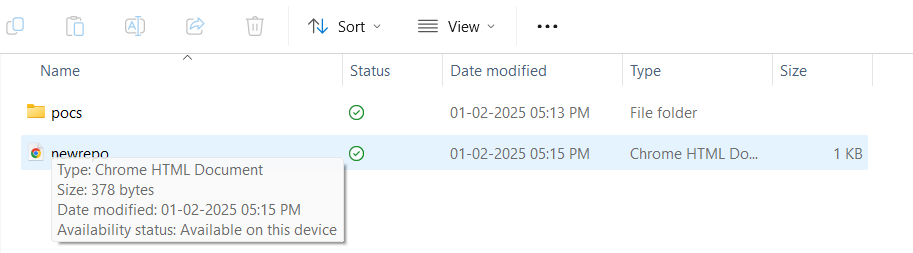


Step 2:

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

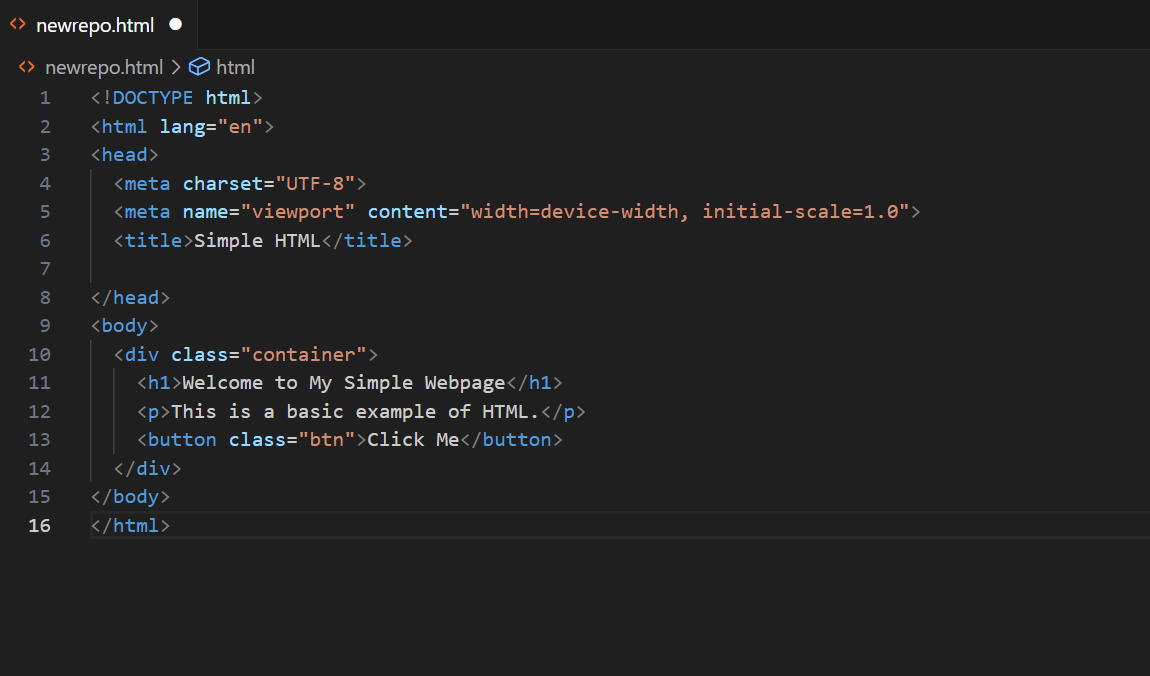
Step 3:

First, open your terminal or command prompt and create a new folder for your project. Let's name it mynew



Step 4:

Open VS code, create a simple HTML file named **newrepo.html**. You can write some basic HTML



Step 5:

Output for the HTML code



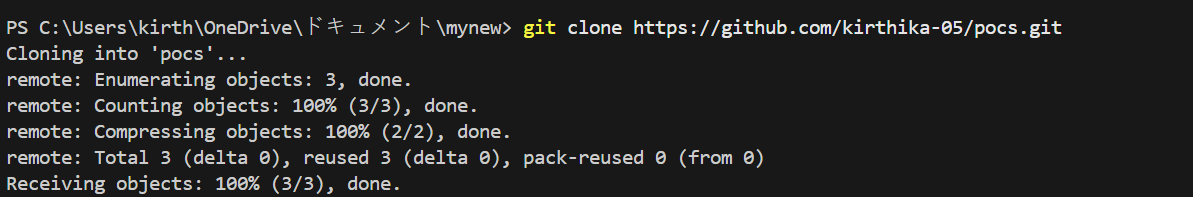
Step 6:

Open the **VS code Terminal** and set the path to the folder named that we created.

**Git clone**

**B**asic syntax for git clone is **git clone <repository\_url>**

 Find the repository URL on GitHub (or another Git hosting service).  
 Run the command:

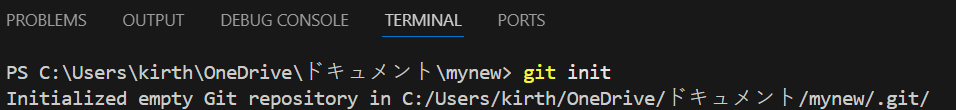


Step 7:

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 8:

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 .**

Basic syntax is **git add newrepo.html** This command adds all the files to Git’s tracking system.

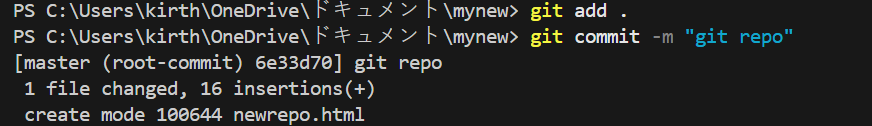
**git commit**

Basic syntax is **git commit -m "Your commit message"**

A **commit** is a snapshot of the project's current state. When you commit, Git records changes to tracked files and saves them in the repository history.

 **git commit**: Saves changes in the local repository.

**-m "message**": Adds a message describing what the commit does.



Step 9:

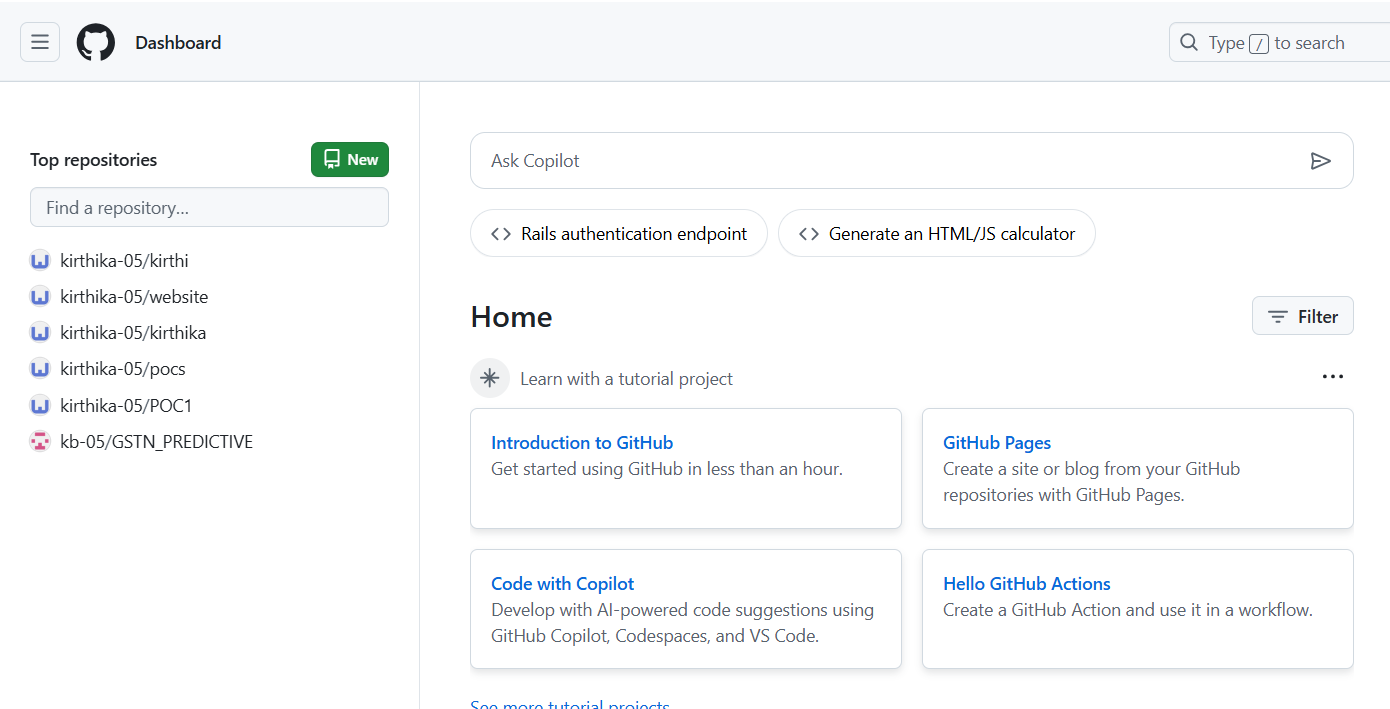
Set Up Your Name and Email Globally Git doesn’t know who is making the commit because you haven’t configured your name and email yet. Git uses this information to track who made the changes.



Step 9:

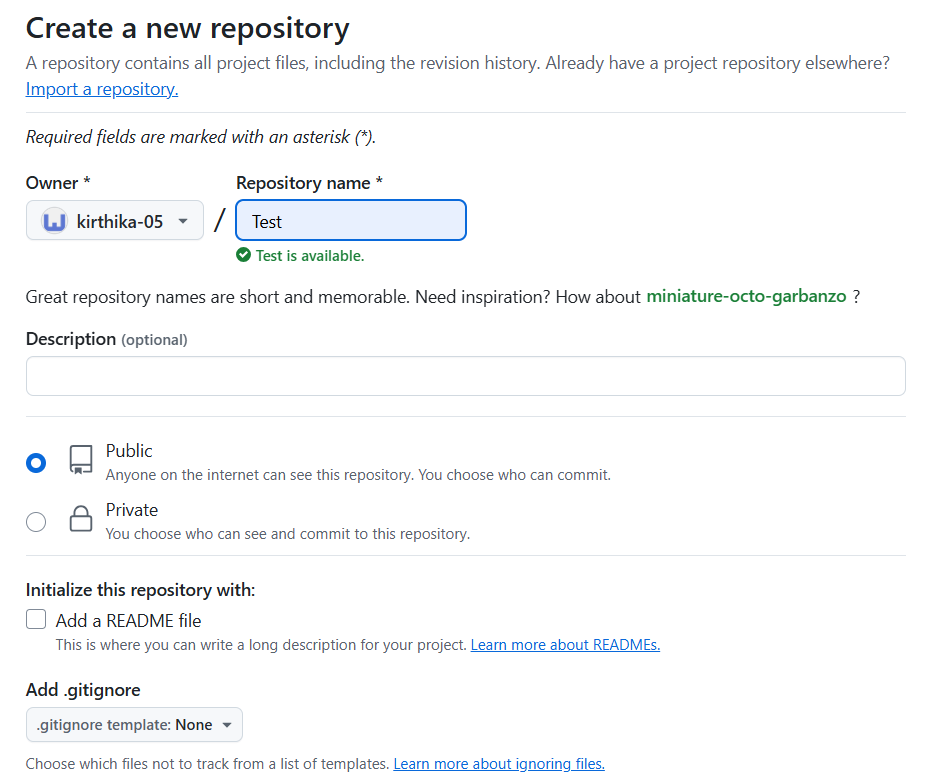
**Create a New Repository**:

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



Step 10:

Give your repository a name, for example, test and click **"Create repository"**.



Step 11

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

Go back to your Command Line and type the following:

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

Replace your username with your GitHub username and with the name of your GitHub repository



Step 12

**git branch -V**

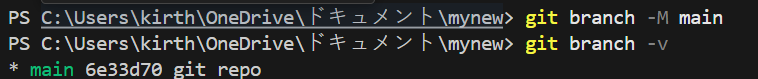


**git branch -M**

The 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:

**git push -u origin main**

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



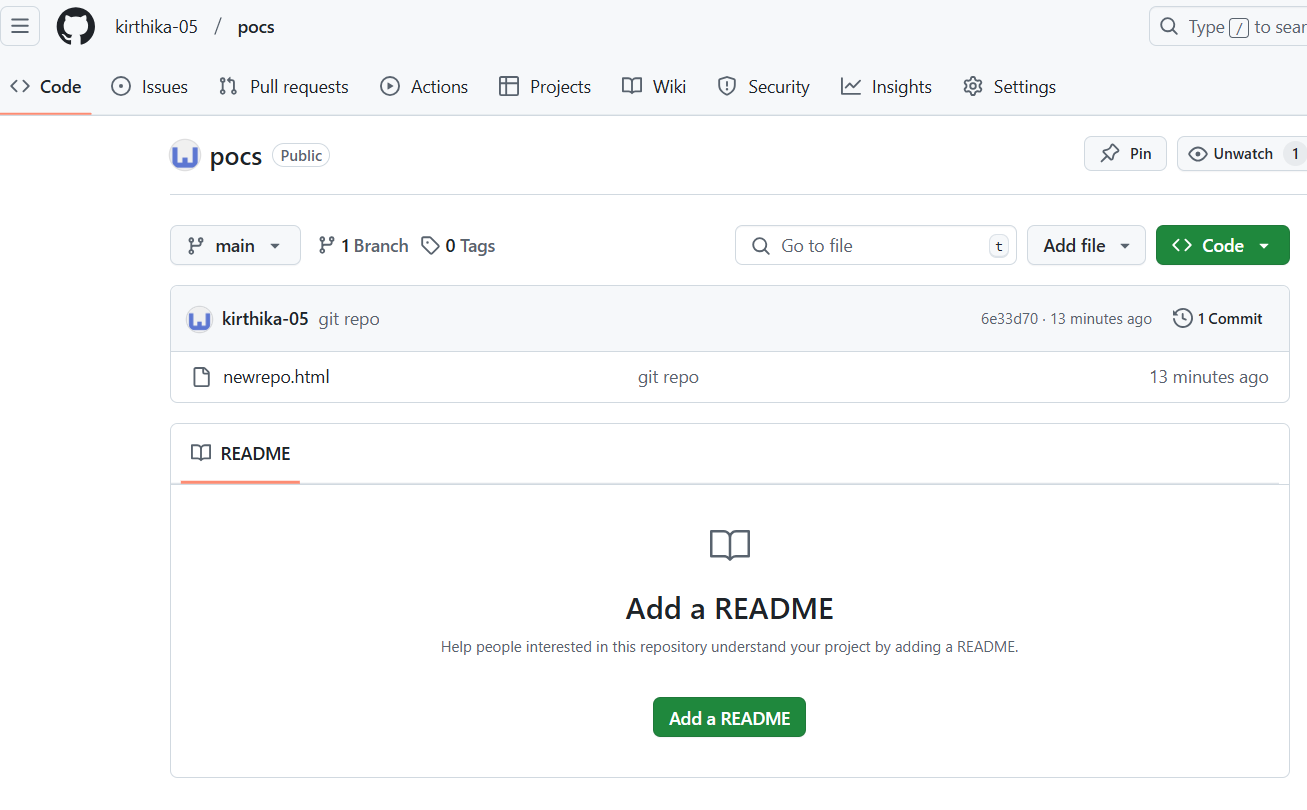
Step 14

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!



**Outcome**

By completing this PoC, you will have:

✅ A local Git repository initialized inside your project folder.

✅ The ability to track and manage changes using Git commands.  
✅ A log of all commits for easy reference.