**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: Oviya G Department: IT



**Introduction**

Version control is a crucial aspect of software development that helps in managing and tracking changes efficiently. Git, a widely used version control system, enables developers to maintain a history of modifications, collaborate seamlessly, and revert to previous versions when necessary. By setting up a local Git repository for your static website, you can systematically track edits, ensure a structured workflow, and safeguard your project from accidental changes or data loss. This setup allows you to experiment with new features, maintain a clean project history, and prepare for future collaboration by keeping your website organized and version-controlled.

**Overview**

In this guide, we will walk through the process of setting up a local Git repository to manage the version control of your static website. This will help you track changes, maintain a structured workflow, and ensure the integrity of your project. The key steps include:

**Installing Git** – Verify that Git is installed and properly configured on your system.

**Initializing a Local Repository** – Create a Git repository in the root folder of your static website.

**Staging and Committing Files** – Add files to the staging area and commit them to the repository to save changes.

**Checking Repository Status** – Use Git commands to track the state of your repository and ensure everything is properly versioned.

**Objectives**

By the end of this process, you will:

**Understand the Basics of Version Control** – Learn how Git helps in managing and tracking changes in projects.

**Set Up a Git Repository** – Initialize a local repository to efficiently version control your static website.

**Track Changes Effectively** – Use Git commands to stage, commit, and log every modification for better organization.

**Maintain a Structured Workflow** – Ensure a clear project history, making it easier to roll back changes if needed.

**Prepare for Collaboration** – Establish a foundation for future teamwork by enabling seamless integration of changes.

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

**Track Changes** – Git keeps a record of all modifications, providing a detailed history of your project.

**Rollback Capabilities** – Easily revert to previous versions in case of errors or unwanted changes.

**Collaboration Readiness** – A Git repository lays the groundwork for smooth teamwork and version-controlled contributions.

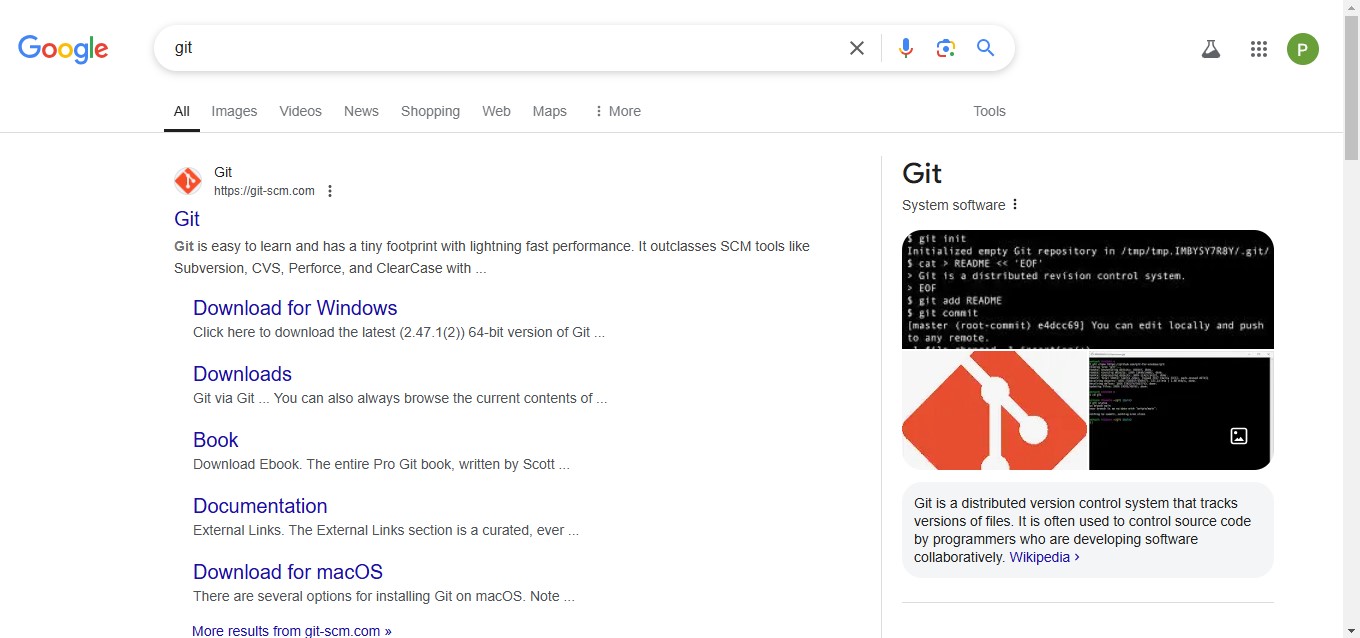
**Project Organization** – Maintain a structured and clean workflow, reducing the risk of accidental data loss or overwrites.

**Experiment Safely** – Try out new features without disrupting the main project, ensuring stability and flexibility in development.

**Step-by-Step Overview**

Step 1:

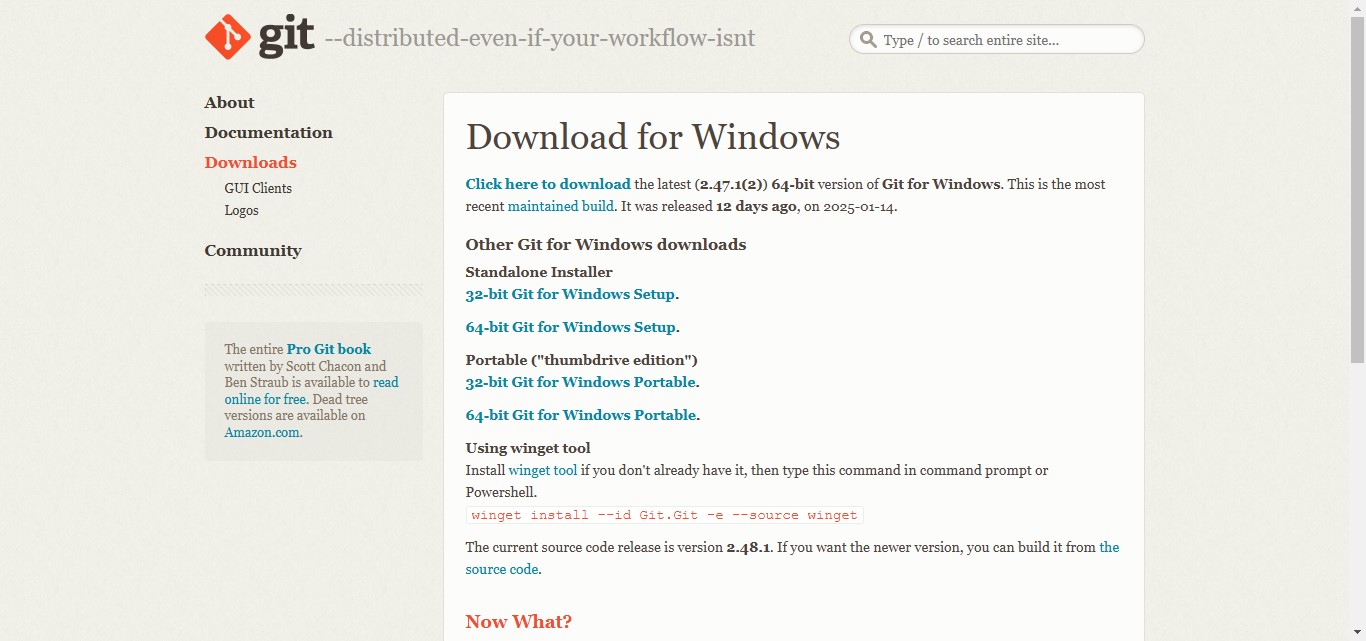
Search for "Git" in Chrome, download it, and click the "Downloads" option on the website.





Step 2

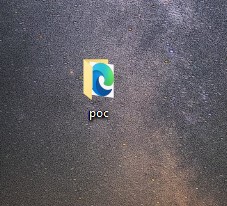
Click the **Windows** option on the download page and follow the installation wizard.

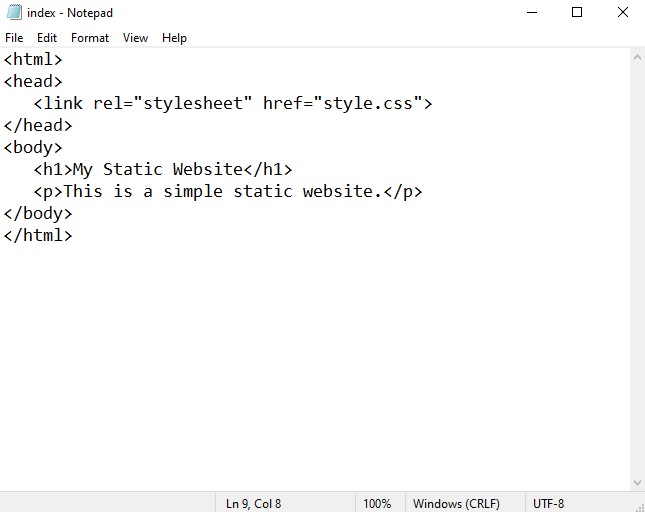


Step 3

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

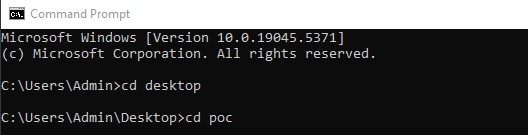
Inside that folder, create a simple HTML file named index.html. You can write some basic HTML





Step 5

Open the Command prompt and set the path to the folder named poc 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 poc 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

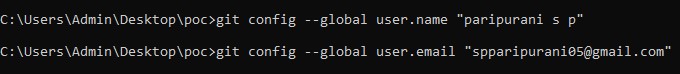
**git add .**

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.



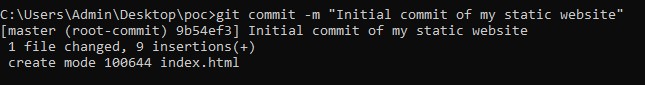
Step 9

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.



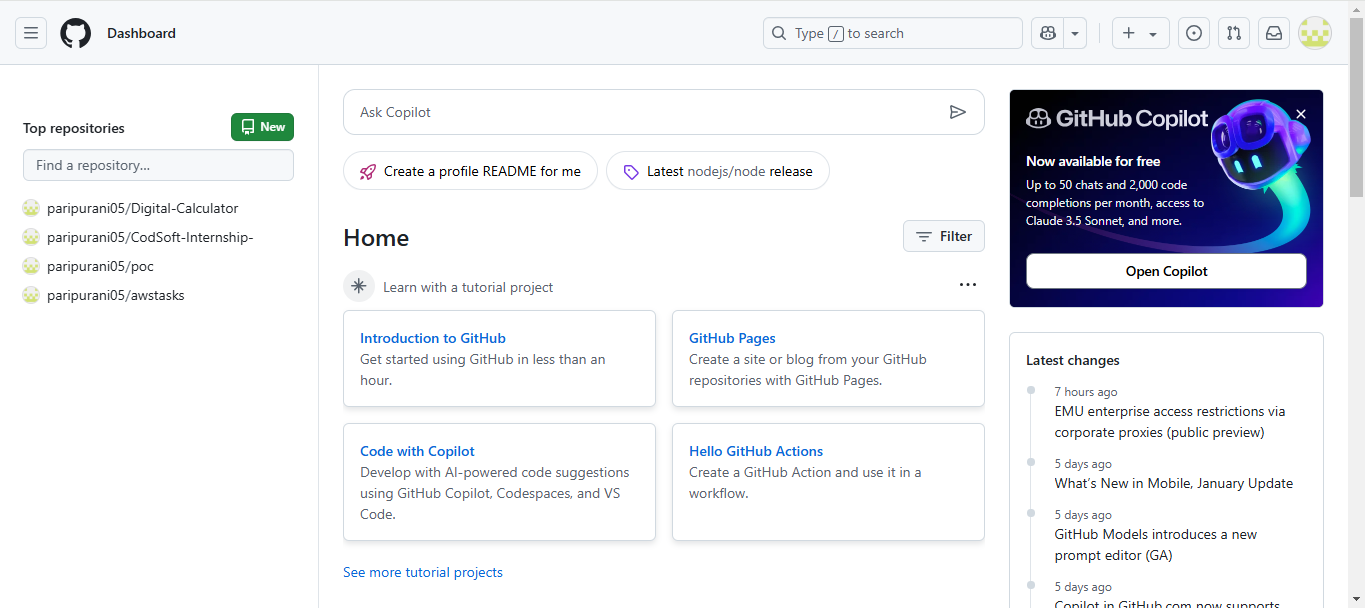
Step 10

**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**:

Go back to your Command Line and type the following:

**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:

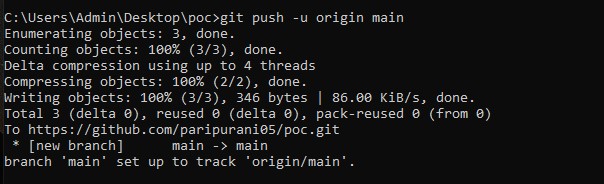
**-M**: This flag forces the renaming, even if a branch named main already exists. It will overwrite the existing main branch.

**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



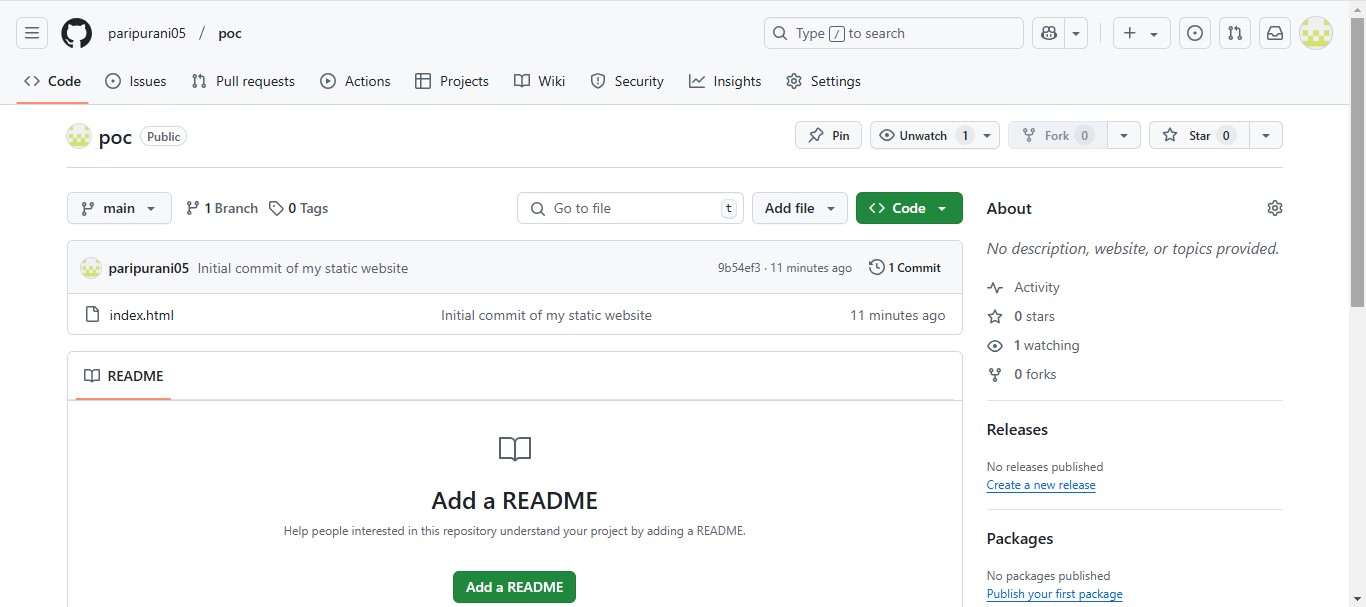
Step 14

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-poc).

You should see your website files there!



**Expected Outcome**

By the end of this process, you will achieve the following outcomes:

A Fully Initialized Local Git Repository – Your static website will have a local Git repository set up in its root directory, allowing you to track and manage changes over time.

Efficient Version Control – You will be able to effectively stage, commit, and track the progress of changes made to your website’s files. Each modification will be logged, providing a clear version history of your project.

Clear and Organized Project Structure – With Git, your project will maintain a structured and organized workflow. This will help avoid clutter and ensure the stability of your website as changes are tracked and managed efficiently.

Ability to Revert Changes – In case of errors or undesired changes, you will be able to roll back to previous versions of your website with ease, offering peace of mind during development.

Preparedness for Collaboration – With the foundation of a local Git repository, your website will be ready for future collaboration. If needed, you will be able to share your repository, integrate changes from team members, and maintain a smooth workflow for collaborative development.

Increased Confidence in Code Management – By using Git for version control, you will be able to experiment with new features or fixes without fear of losing work, knowing you can always revert to a stable version.