St. JOSEPH'S GROUP OF INSTITUTIONS
OMR, CHENNAI - 119

aws

# 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:** ROHITH S

**Department:** INFORMATION TECHNOLOGY


St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY

AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

## Introduction

Version control is an essential practice in software development, enabling you to manage and track changes to your code over time. It facilitates collaboration, provides a structured way to update projects, and allows you to revert to previous versions when necessary. Git, a widely used version control system, is known for its efficiency, flexibility, and distributed nature.

In this Proof of Concept (POC), we will initialize a local Git repository to version control your static website. This setup will help you monitor changes, experiment with new features in a controlled environment, and seamlessly share your project with others. Establishing a Git repository is a crucial step in maintaining an organized and reliable workflow, particularly for developers and teams working on collaborative projects.

## Overview

In this setup, we will cover the following steps:

1. **Installing Git** – Verify that 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 project files to the staging area and commit them to save snapshots of your work.
4. **Reviewing Repository State** – Use Git commands to check the status of your repository and ensure all changes are tracked correctly.

## Objective

By the end of this Proof of Concept (POC), you will:

1. **Understand Version Control Basics** – Learn the significance of Git in managing and tracking changes in your projects.
2. **Set Up a Git Repository** – Initialize a local Git repository to version control your static website.
3. **Track Changes Efficiently** – Master staging and committing files to log every modification accurately.
4. **Maintain Project Organization** – Establish a structured workflow that allows for easy rollbacks and version management.
5. **Prepare for Collaboration** – Gain the foundation needed to share your repository and collaborate seamlessly using Git.

## Importance

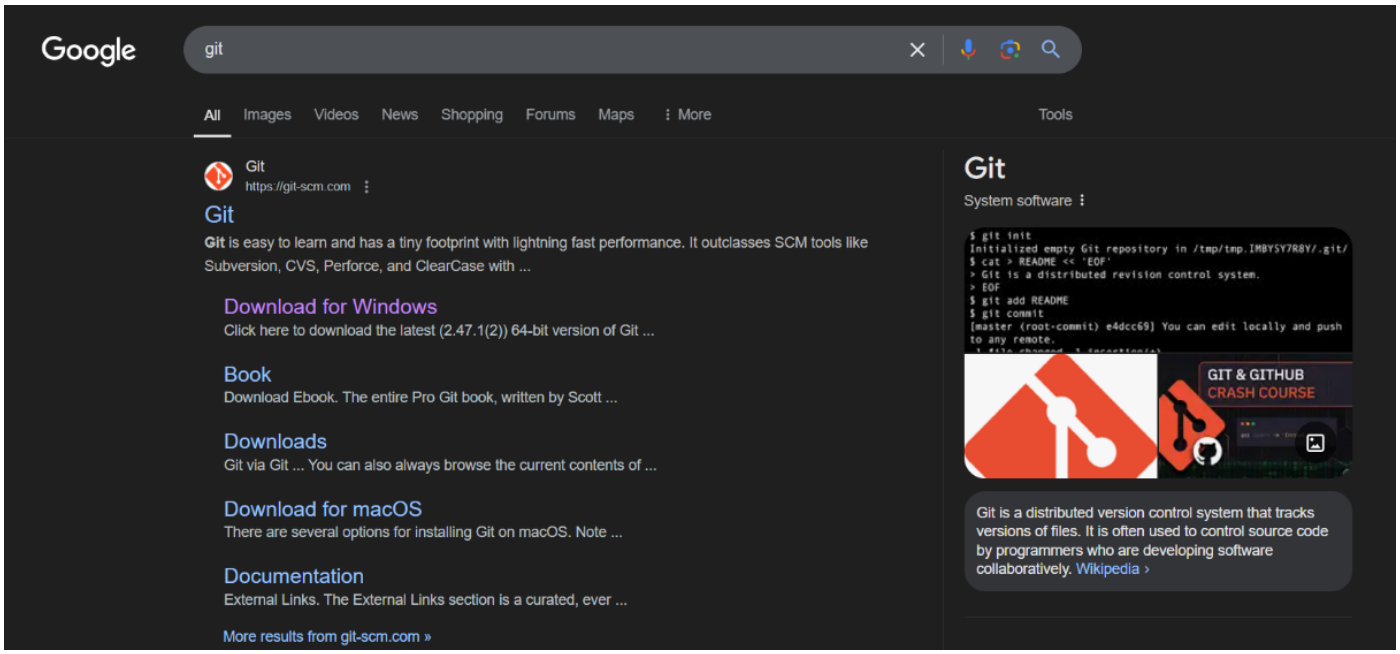
- **Track Changes** – Git maintains a detailed history of all modifications, providing clear version tracking for your project.
- **Rollback** – Effortlessly revert to previous versions to correct mistakes or restore earlier states.
- **Collaboration** – Streamline teamwork by enabling seamless integration of changes and shared development efforts.

---

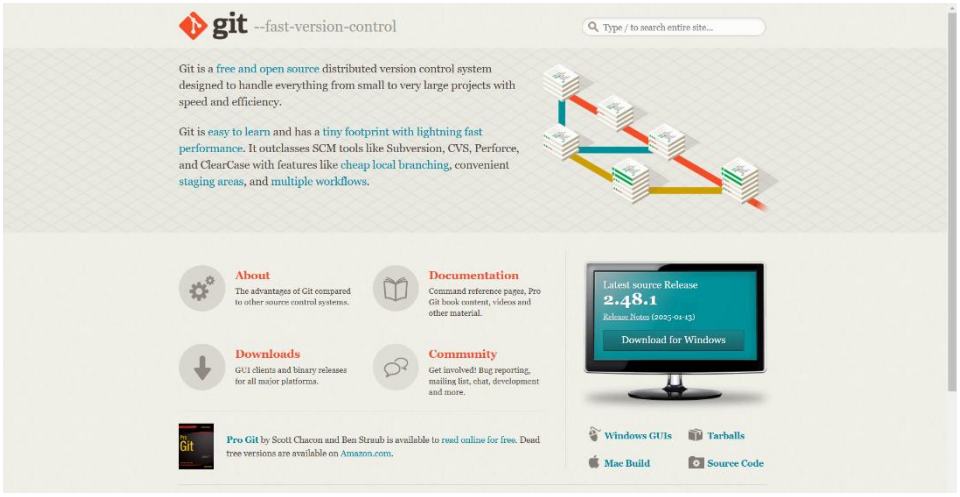
## Step-by-Step Overview

### Step 1:

- Open Chrome and search for **"Git"**.

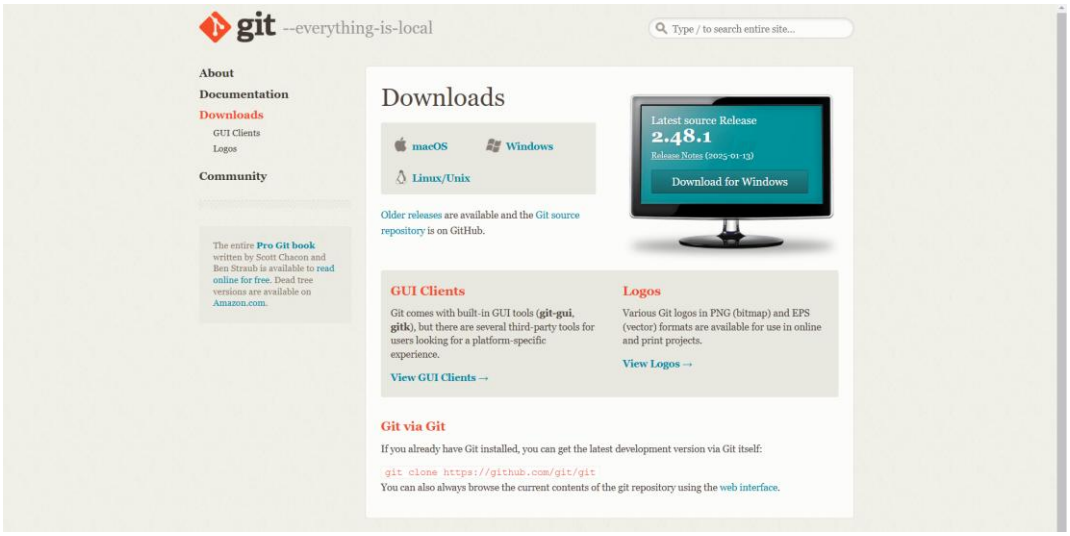


- Visit the official Git website and click on the **"Downloads"** option.



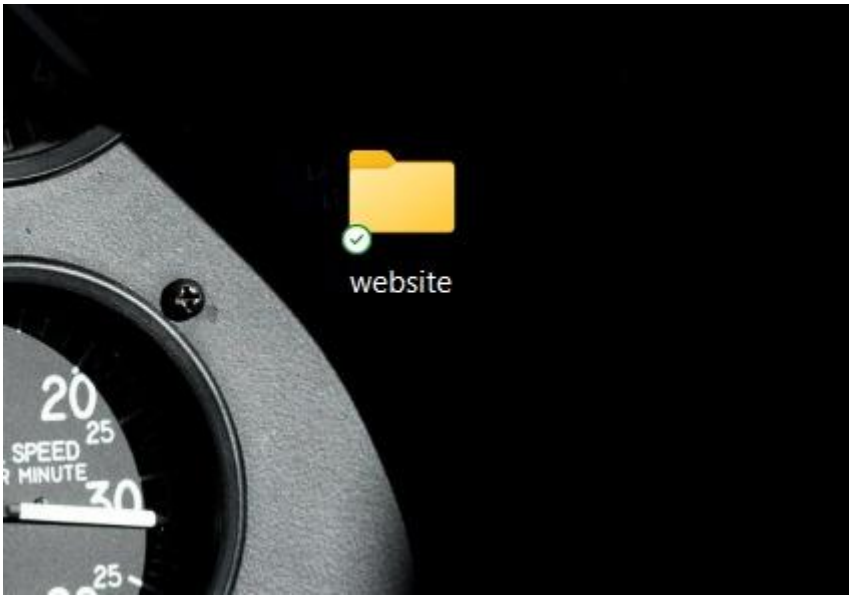

## Step 2:

Select the **Windows** version and follow the installation wizard to complete the setup.

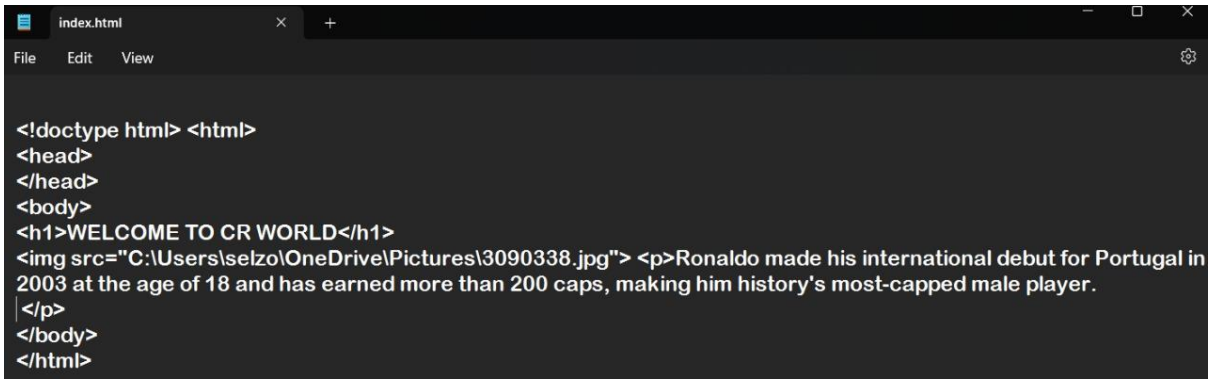



## Step 3:

- On your **Desktop**, create a new folder named **website**.

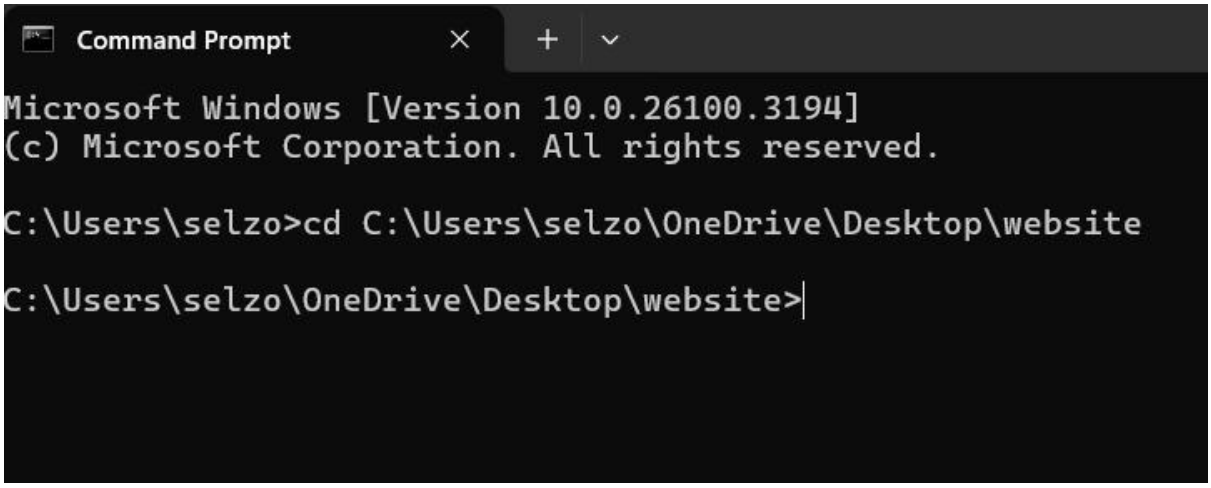



- Inside this folder, create a simple HTML file called **index.html** and write some basic HTML content.




## Step 4:

- Open the **Command Prompt**.
- Change the directory to the newly created **website** folder using the command:




## Step 5:

- Initialize Git in your project folder by running:
  **git init**
- This creates a hidden .git folder, signaling that Git will start tracking your files.


## Step 6:

- To track all the files in your project, use:
  **git add .**
- This stages all files for version control.


## Step 7:

Set your name and email globally so Git can identify your commits:

**git config --global user.name "Your Name"**

**git config --global user.email "your.email@example.com"**

```
C:\Users\selzo\OneDrive\Desktop\website>git config --global user.name "rohith s"

C:\Users\selzo\OneDrive\Desktop\website>git config --global user.email "selzor123456@gmail.com"
```

## Step 8:

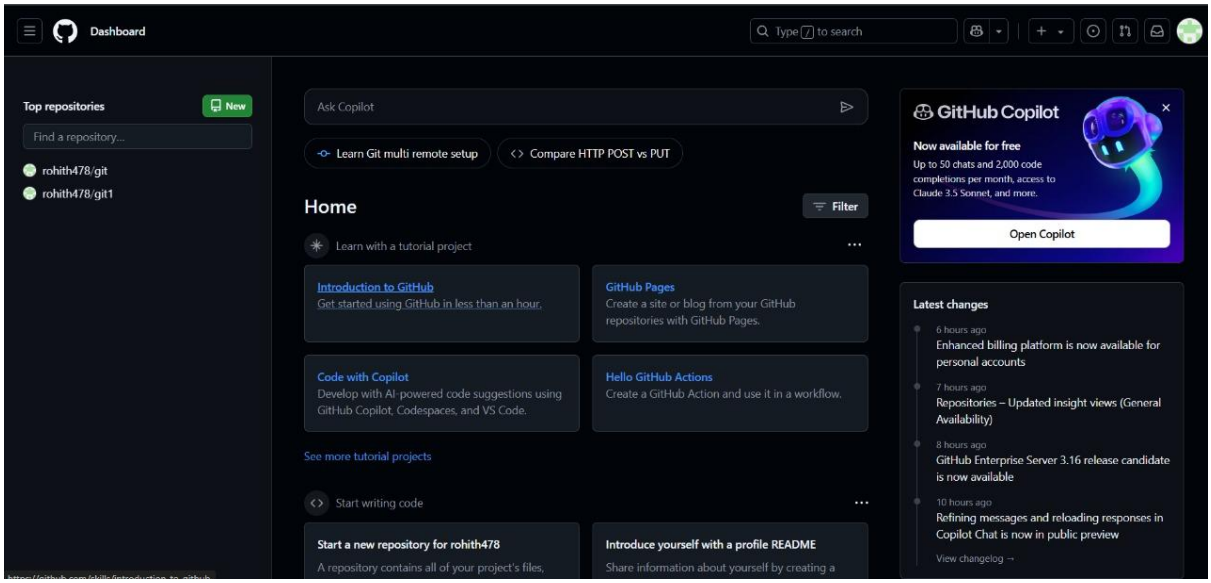
- Now, we need to save these changes in Git. When you "commit" changes, Git takes a snapshot of your files.
- Save your changes in Git with a meaningful commit message:
  **git commit -m "Initial commit of my static website"**
- The **-m** flag allows you to add a message about your changes. In this case, we're saying this is the "initial commit", meaning the first time we're saving our work.

```
C:\Users\selzo\OneDrive\Desktop\website>git commit -m "Initial commit of my static website"
[master (root-commit) 899f5fe] Initial commit of my static website
 1 file changed, 10 insertions(+)
 create mode 100644 index.html
```

## Step 9:

### Create a New GitHub Repository

1. Log in to GitHub.
2. Click the "**New**" button at the top-right corner.
3. Give your repository a name (e.g., **my-website**), leave the settings as default, and click "**Create repository**".



## Step 10:

- Add the GitHub repository as a remote origin by running:
  **git remote add origin https://github.com/yourusername/my-website.git**

*(Replace yourusername with your actual GitHub username and my-website with your repository name.)*

```
C:\Users\selzo\OneDrive\Desktop\website>git remote add origin https://github.com/rohith478/my-website.git
```

## Step 11:

- The command is used to rename the current branch to main:
  **git branch -M main**
- Here **-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.

```
C:\Users\selzo\OneDrive\Desktop\website>git branch -M main
```

## Step 12:

- The command is used to upload your project to GitHub by running:
  **git push -u origin main**
- Then, set it as the upstream branch.

```
C:\Users\selzo\OneDrive\Desktop\website>git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 468 bytes | 93.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/rohith478/my-website.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

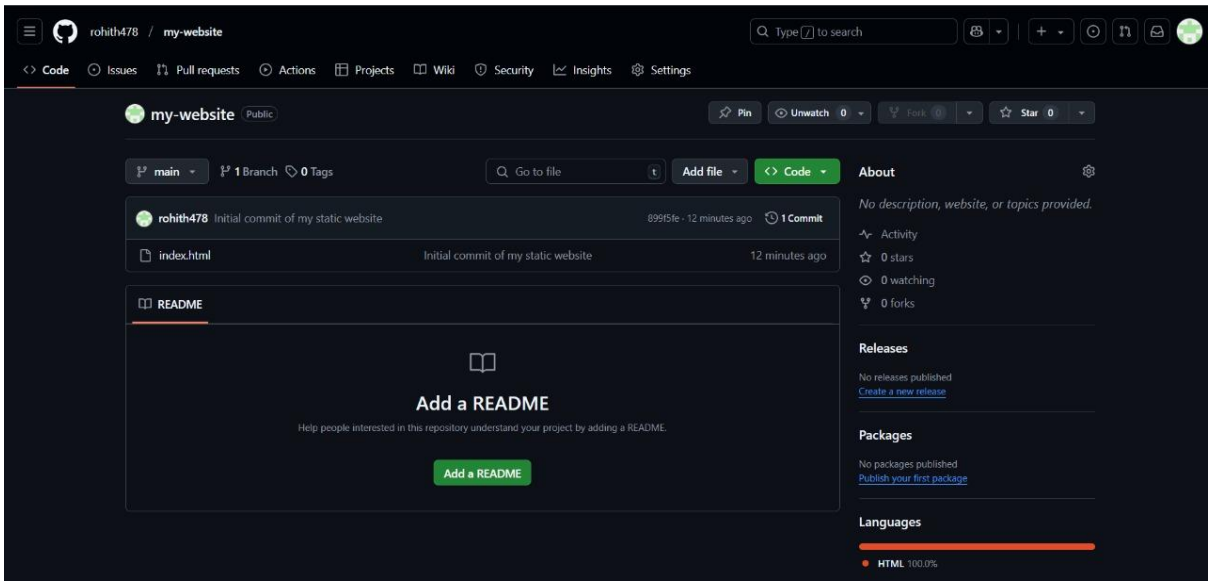
## Step 13:

### Verify Your Files on GitHub

1. Open your web browser and navigate to your GitHub repository:

   **https://github.com/yourusername/my-website**

2. You should see your website files successfully uploaded!




# Expected Outcome

By completing this Proof of Concept (PoC), you will:

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

2. Track and manage changes to your **website files** (HTML, CSS, etc.) using Git.
3. Learn and apply fundamental **Git commands** (git init, git add, git commit) for version control.
4. Commit changes locally with **clear and descriptive messages** to maintain a structured history.
5. Gain **practical experience** in using Git for tracking and managing website file modifications.