

# 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: MONISH KJ**

**REG NO:312323243103**

**DEPT: ADS**

# 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

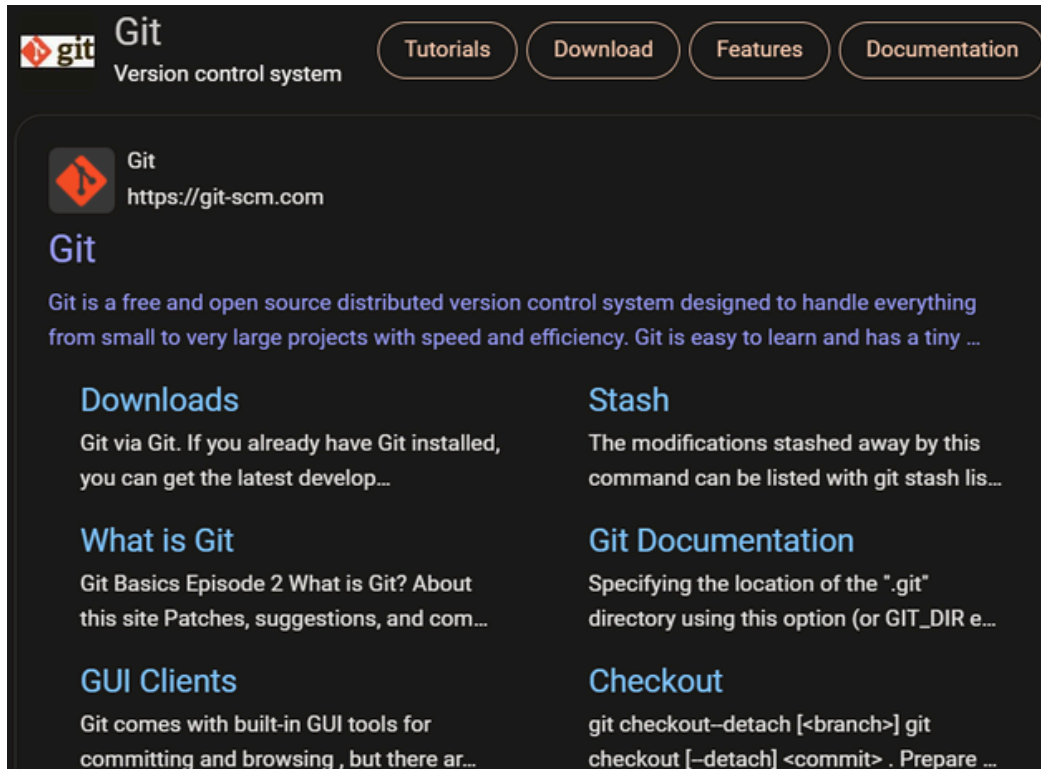
**Track Changes:** Git records all modifications, ensuring a clear history of your project.

**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:

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



The screenshot shows the Git website homepage. At the top, there's a navigation bar with the Git logo and the text "Git Version control system". To the right of the logo are four buttons: "Tutorials", "Download", "Features", and "Documentation". Below the navigation bar, there's a section with the Git logo and the URL "https://git-scm.com". The main heading is "Git". Below it, a paragraph describes Git as a free and open source distributed version control system. There are six columns of links: "Downloads", "Stash", "What is Git", "Git Documentation", "GUI Clients", and "Checkout". Each column has a brief description of the link's content.

**Git**  
Version control system

[Tutorials](#) [Download](#) [Features](#) [Documentation](#)

**Git**  
<https://git-scm.com>

**Git**

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny ...

**Downloads**  
Git via Git. If you already have Git installed, you can get the latest develop...

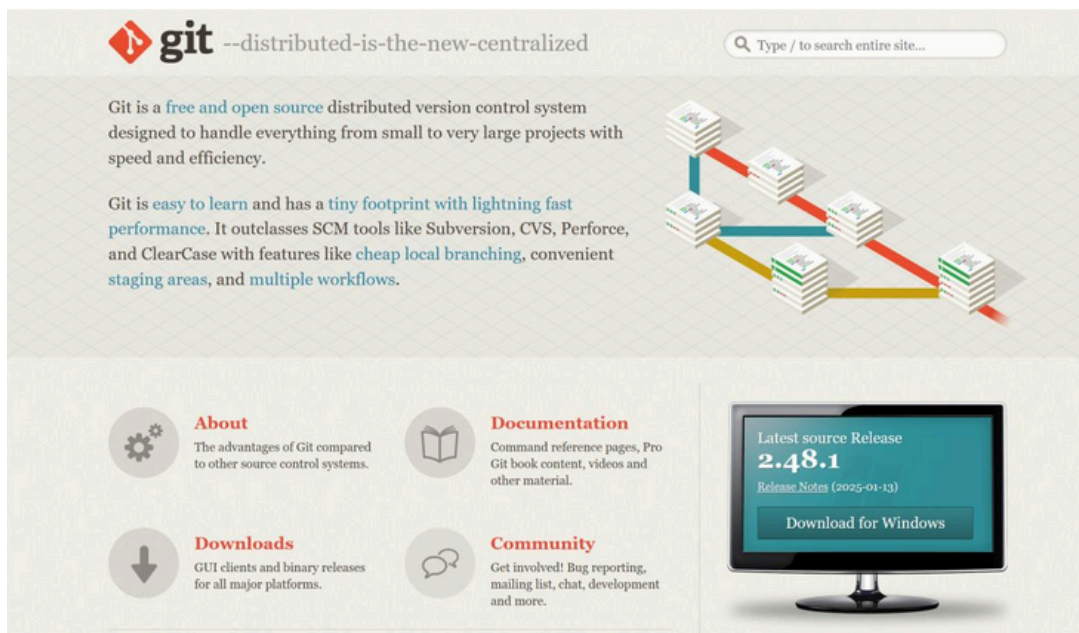
**Stash**  
The modifications stashed away by this command can be listed with git stash lis...

**What is Git**  
Git Basics Episode 2 What is Git? About this site Patches, suggestions, and com...

**Git Documentation**  
Specifying the location of the ".git" directory using this option (or GIT\_DIR e...

**GUI Clients**  
Git comes with built-in GUI tools for committing and browsing , but there ar...

**Checkout**  
git checkout--detach [<branch>] git checkout [--detach] <commit> . Prepare ...



The screenshot shows the Git website homepage. At the top, there's a navigation bar with the Git logo and the text "git --distributed-is-the-new-centralized". To the right of the logo is a search bar with the text "Type / to search entire site...". Below the navigation bar, there's a section with the Git logo and the text "Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency." There's also a paragraph about Git being easy to learn and having a tiny footprint with lightning fast performance. To the right of the text is a diagram showing a branching model with multiple stacks of code blocks connected by lines. Below the text and diagram, there are four columns of links: "About", "Documentation", "Downloads", and "Community". Each column has a brief description of the link's content. To the right of these columns is a monitor displaying the latest source release "2.48.1" and a button to "Download for Windows".

**git** --distributed-is-the-new-centralized

Type / to search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

**About**  
The advantages of Git compared to other source control systems.

**Documentation**  
Command reference pages, Pro Git book content, videos and other material.

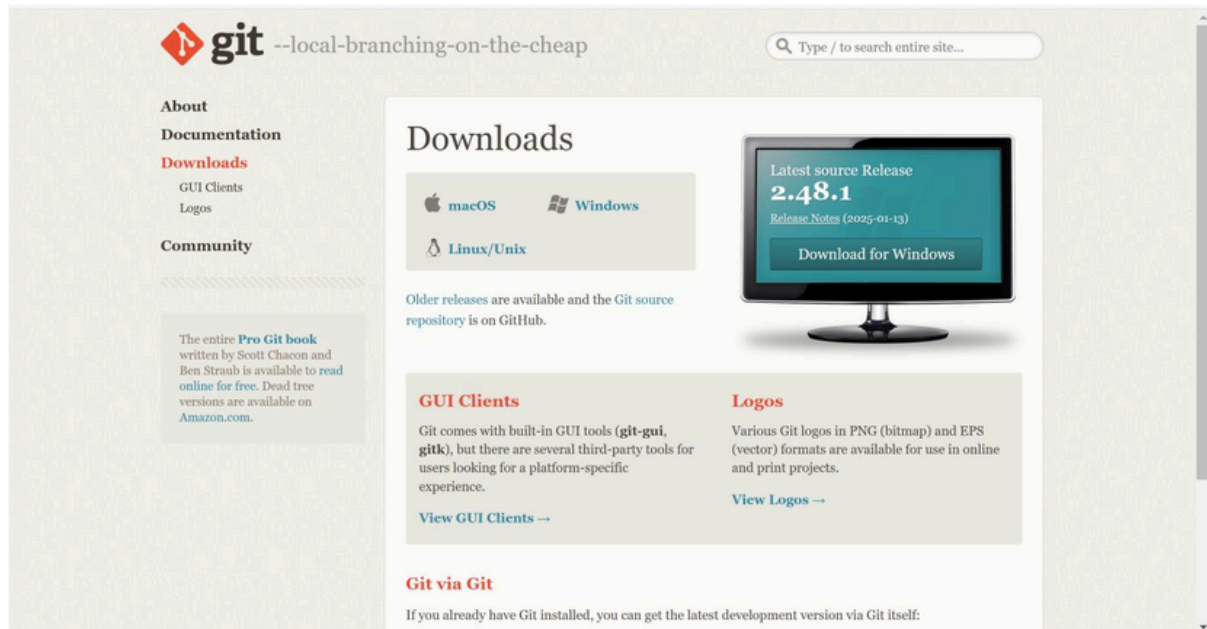
**Downloads**  
GUI clients and binary releases for all major platforms.

**Community**  
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release  
**2.48.1**  
[Release Notes \(2025-01-13\)](#)  
[Download for Windows](#)

## Step 2

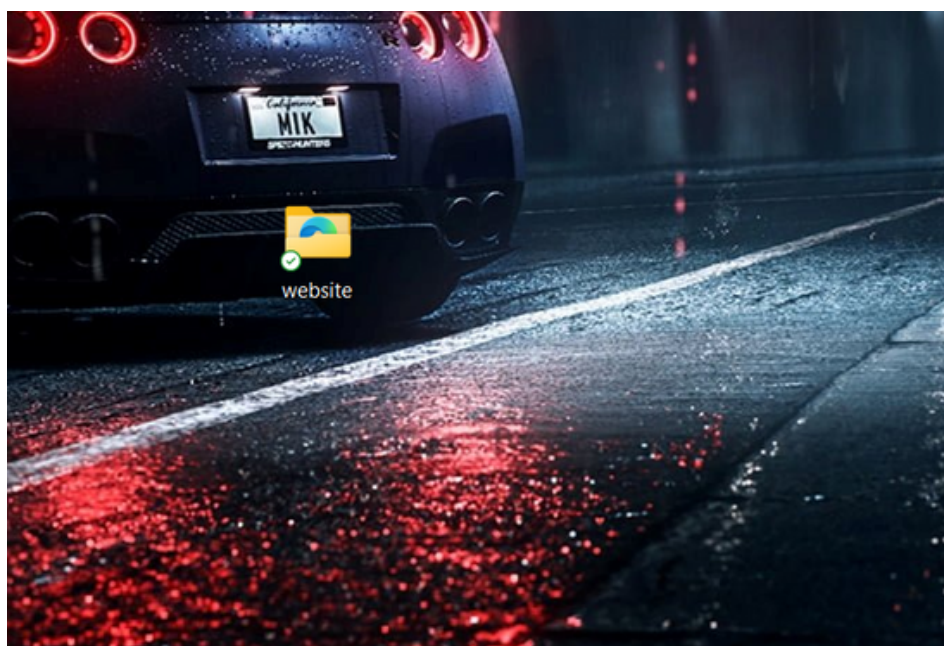
Select the **Windows** option on the download page and follow the installation wizard.

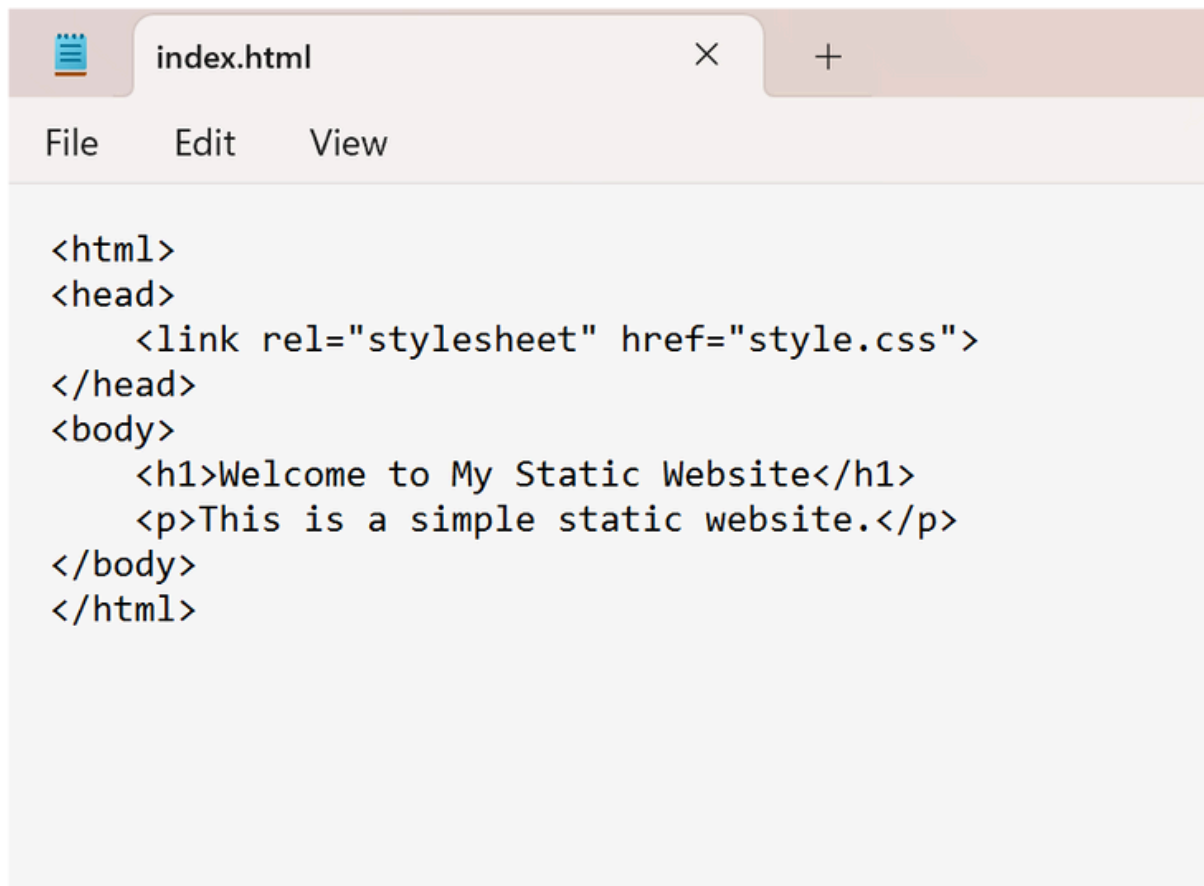


## Step 3

In Desktop Create a folder named website <sup>for static</sup> website  
Inside that folder, create a simple HTML file named index.html.

You  
can write some basic HTML



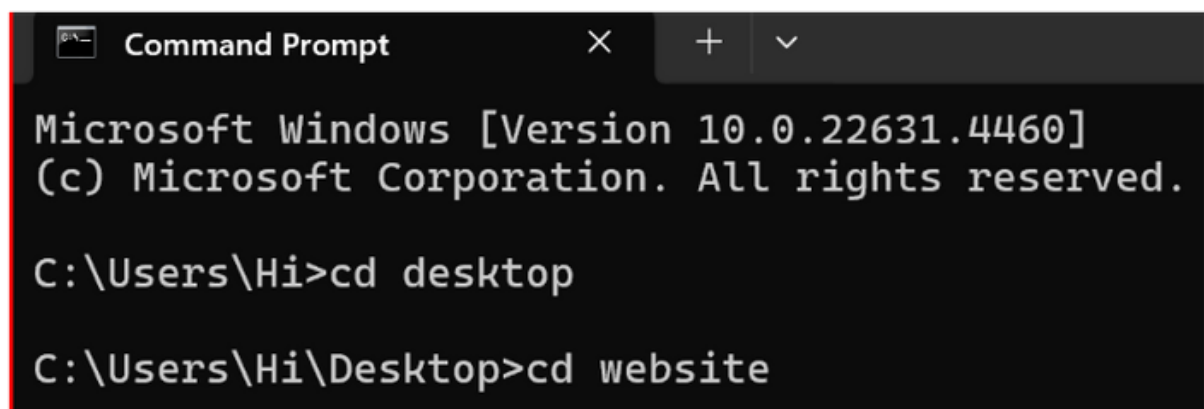


A screenshot of a web browser window. The tab is titled 'index.html'. The address bar is empty. The page content is a simple HTML document with a white background and black text. The text reads: 'Welcome to My Static Website' followed by 'This is a simple static website.' The browser's menu bar shows 'File', 'Edit', and 'View'.

```
<html>
<head>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Welcome to My Static Website</h1>
  <p>This is a simple static website.</p>
</body>
</html>
```

## Step 5

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



A screenshot of a Windows Command Prompt window. The title bar says 'Command Prompt'. The window shows the following text: 'Microsoft Windows [Version 10.0.22631.4460] (c) Microsoft Corporation. All rights reserved. C:\Users\Hi>cd desktop C:\Users\Hi\Desktop>cd website'. The prompt is at the 'website' directory.

```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Hi>cd desktop
C:\Users\Hi\Desktop>cd website
```

## 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.

```
C:\Users\Hi\Desktop\website>git init  
Initialized empty Git repository in C:/Users/Hi/Desktop/website/.git/
```

## 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.

```
C:\Users\Hi\Desktop\website>git add .
```

## 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.

```
C:\Users\Hi\Desktop\website>git config --global user.name "Saravana Krishnan J"  
C:\Users\Hi\Desktop\website>git config --global user.email "saravanakrishnan16@gmail.com"
```



## 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.

```
C:\Users\Hi\Desktop\website>git commit -m "Initial commit of my static website"
[master (root-commit) 3afbd24] Initial commit of my static website
1 file changed, 9 insertions(+)
create mode 100644 index.html
```

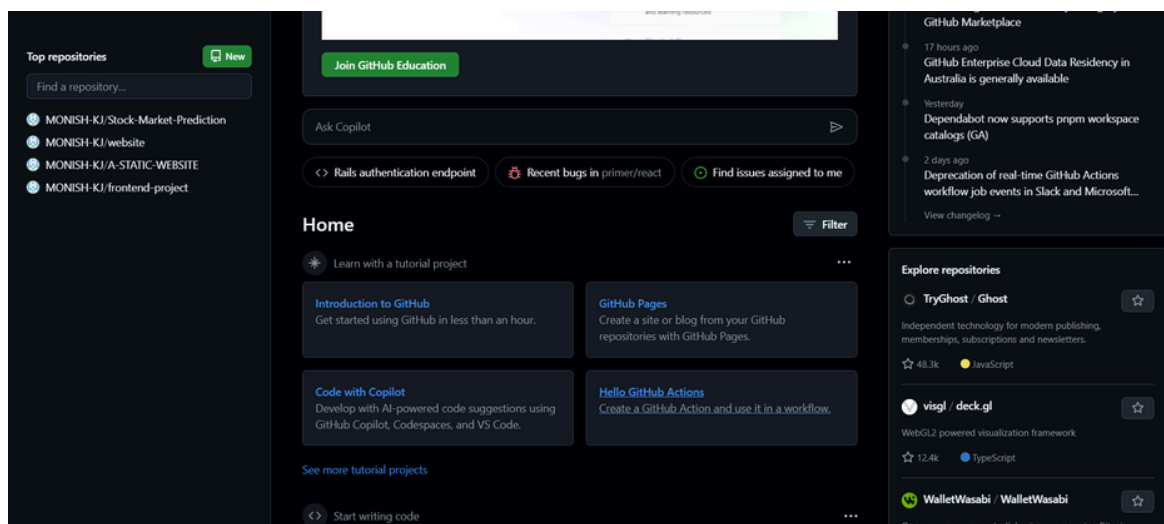
## Step 10

### 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.

```
C:\Users\Hi\Desktop\website>git remote add origin https://github.com/SaravanaKrishnan16/website.git
```

## 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.

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

## 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

```
C:\Users\Hi\Desktop\website>git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 359 bytes | 359.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/SaravanaKrishnan16/website.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

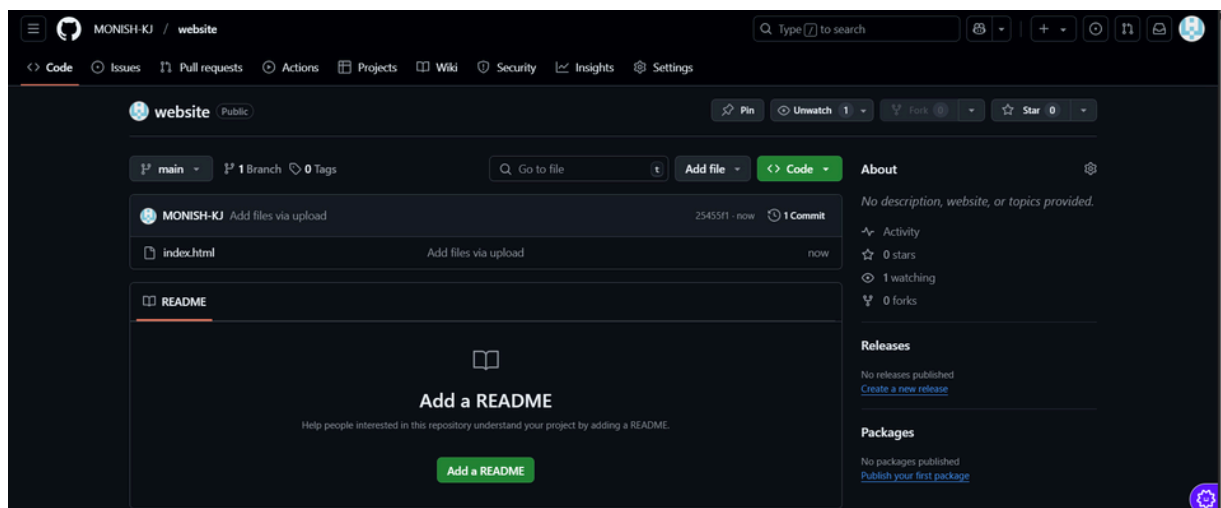
## Step 14

### 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!



# Outcomes

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.