

**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: **Bhavana Vijayan**

DEPARTMENT: **IT**



**Introduction**

Git is a widely used version control system that allows developers to track changes in their codebase, collaborate efficiently, and maintain different versions of their projects. By using Git, you can ensure the **stability and consistency** of your static website while keeping a history of modifications.

**Overview**

This POC walks you through the process of **initializing a Git repository for your static website and version-controlling it**. You will learn how to:

* Set up Git in your project directory
* Stage and commit files
* Connect to a remote repository (optional)
* Push changes to GitHub or another remote platform

**Objective**

**The main objective is to:**

* Initialize a Git repository for your static website.
* Track and manage changes efficiently.
* Maintain version history for future references.
* Enable collaboration and backup using a remote repository.

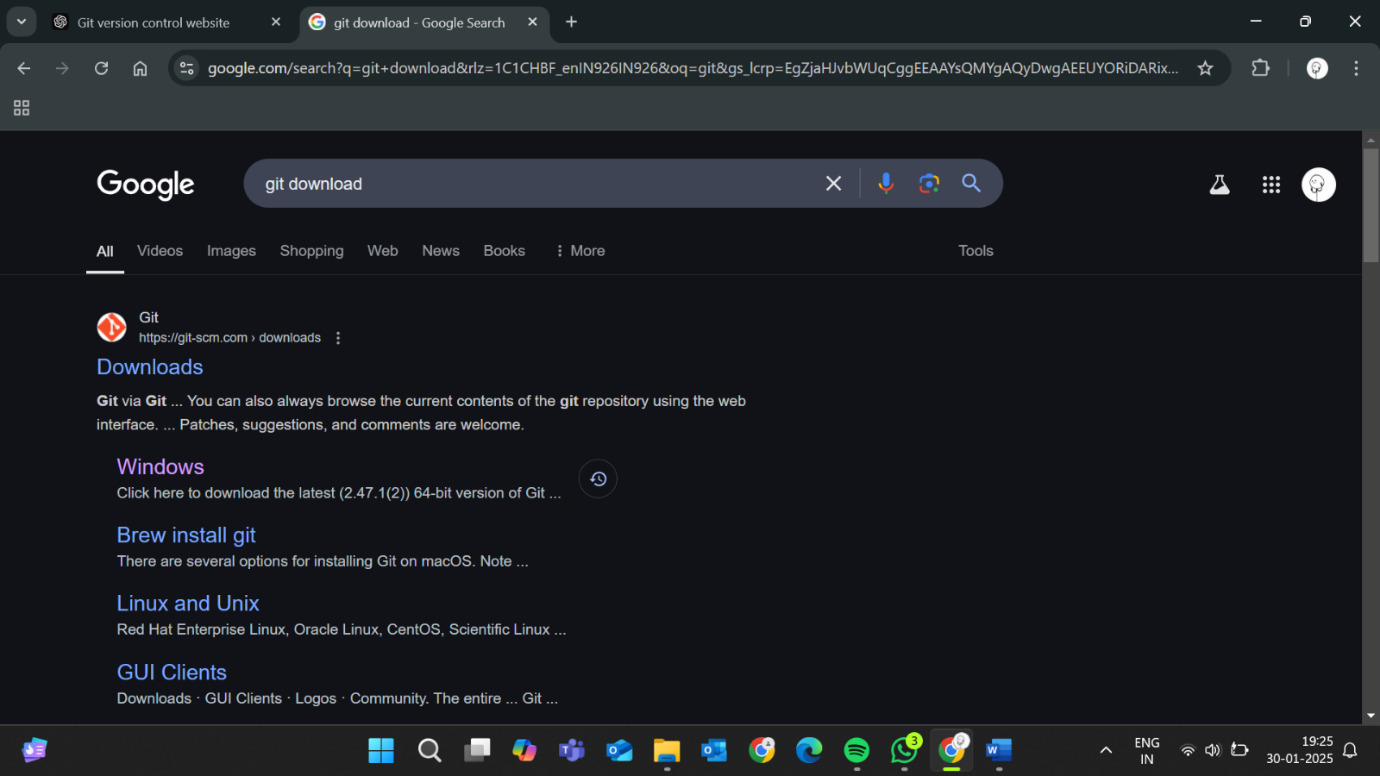
**Important Concepts**

* **Repository**: A storage location where your project's files and history are tracked.
* **Commit**: A snapshot of changes made to the project.
* **Branch**: A parallel version of the project for feature development.
* **Remote Repository**: A cloud-hosted repository (e.g., GitHub, GitLab).
* **Push/Pull**: Sending and receiving updates to/from the remote repository.

**Step-by-Step Overview**

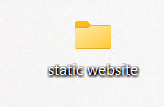
**Step1:**

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



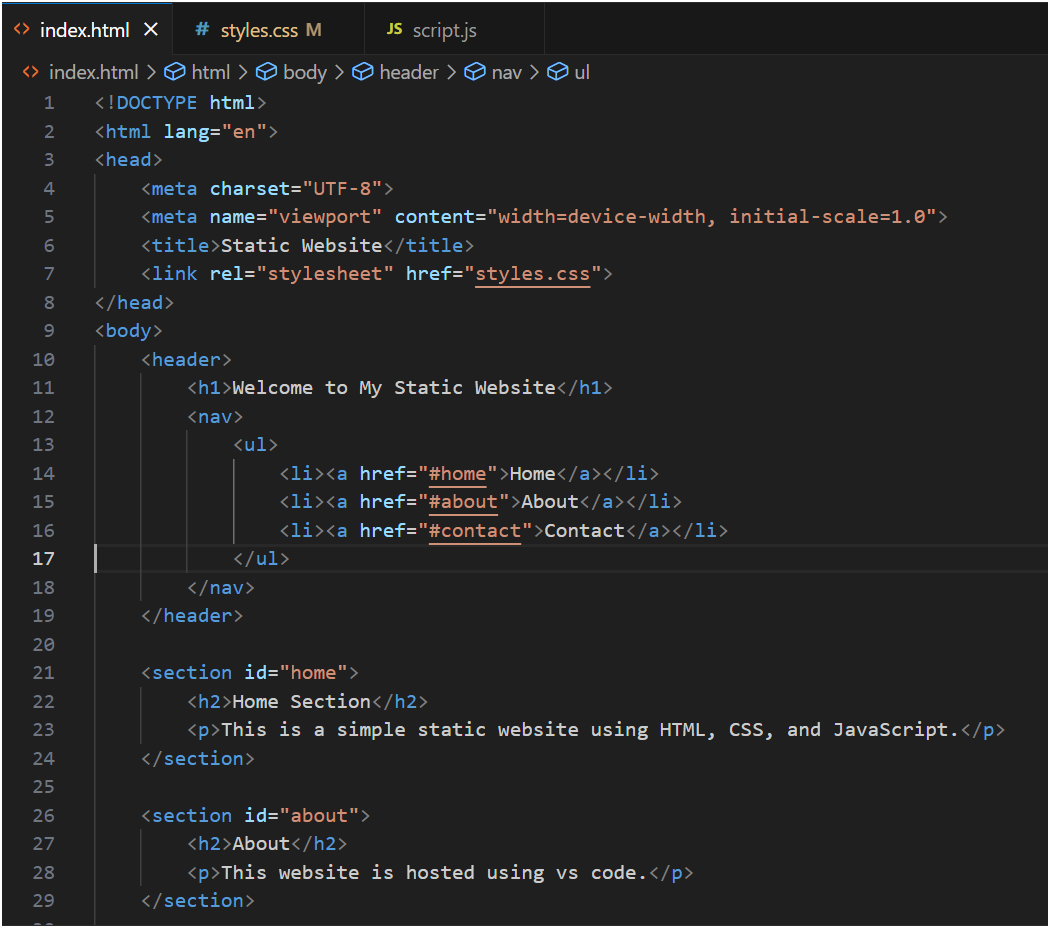
**Step 2 :**

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



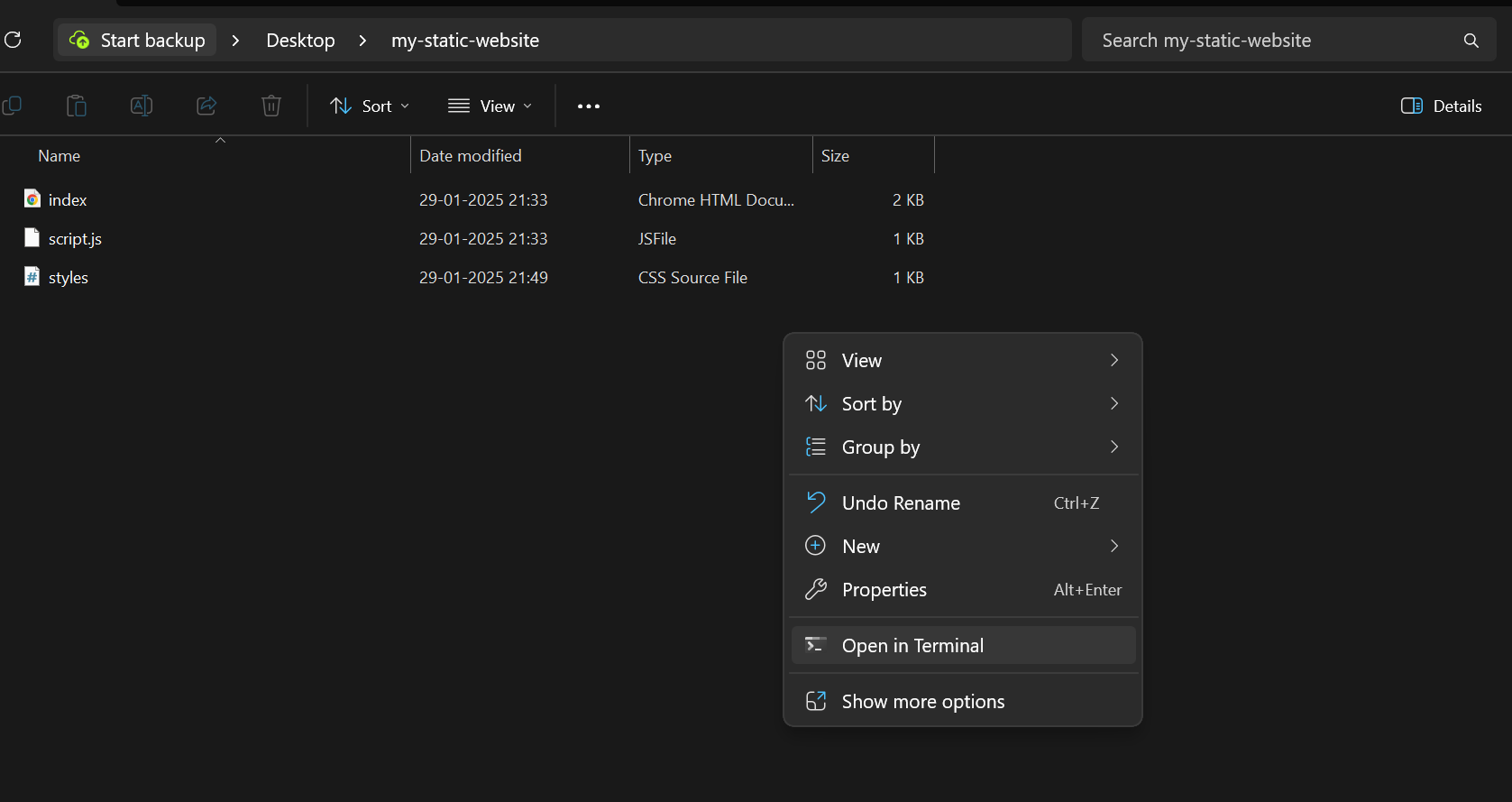
**Step 3:**

* Inside that folder, create a simple HTML file named index.html. You can write some basic HTML
* You can also write basic css code in file to style your static website.



**Step 4:**

Open the terminal in the folder where the html file is located.

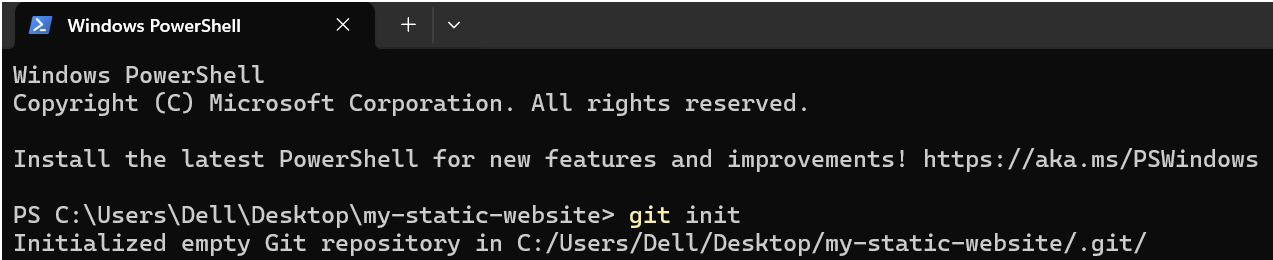


**Step 5:**

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.

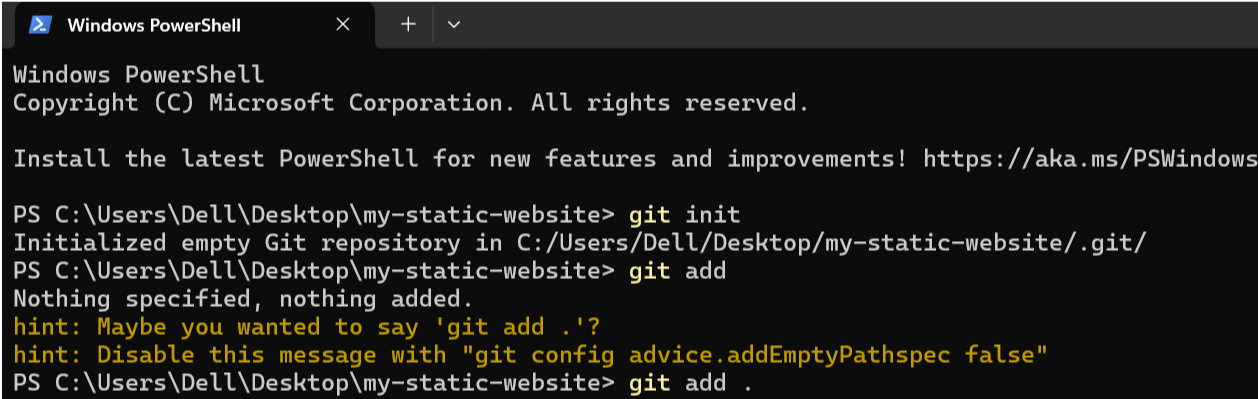


**Step 6:**

* 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.



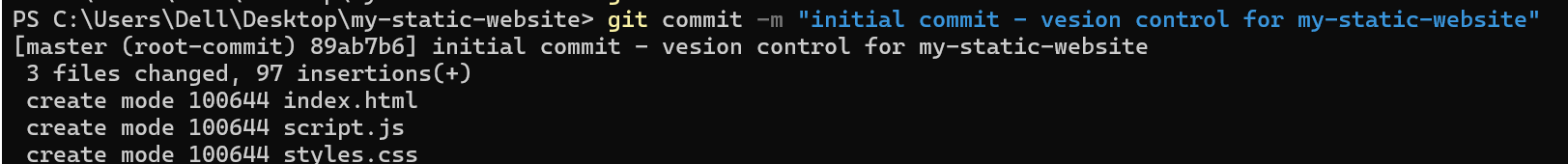
**Step 7:**

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 "First 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.

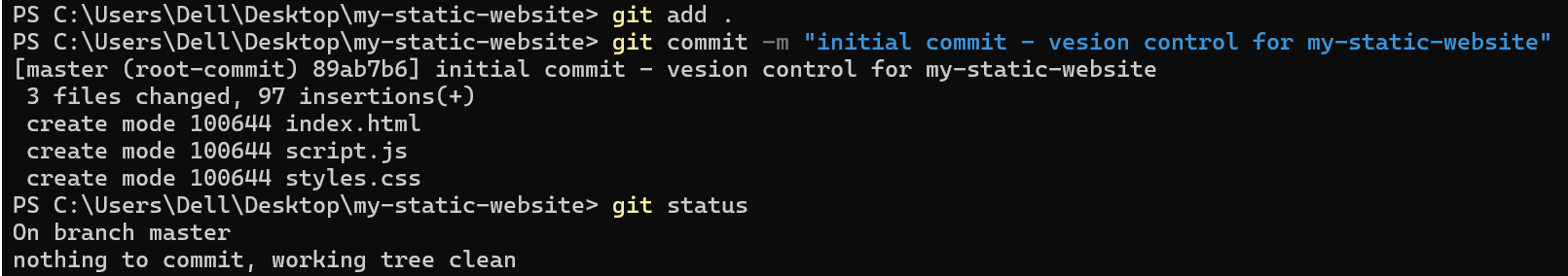


**Step 8:**

Verify Repository Status

* Check the status of your repository using:

**git status**



**Conclusion**

By following these steps, you have successfully initialized a Git repository and version-controlled your static website. This ensures better tracking of changes, collaboration, and backup of your project.