



**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:M Anumitha Department:AML



**INTRODUCTION:**

Version control is essential for managing changes to your codebase, especially for projects like static websites. Git, a distributed version control system, allows you to track changes, collaborate with others, and maintain a history of your work. Setting up a local repository is the first step toward managing your project effectively.

In this POC, I am going to demonstrate the process of initializing a Git repository locally. The goal is to showcase how to set up version control for a project from scratch and efficiently manage the codebase during the development phase.

**OVERVIEW:**

Git is a powerful distributed version control system that allows developers to track changes, collaborate efficiently, and maintain a history of project development. A local repository is a private, offline workspace on your machine where all project files and their version histories are stored. It serves as the foundation for version control, enabling developers to create, edit, and organize code before sharing it with others via remote repositories.

**OBJECTIVE:**

1.Enable Version Control:

To maintain a history of changes and modifications during the POC development phase, ensuring a clear and organized workflow.

2. Facilitate Experimentation:

To provide a safe space for testing and iterating on ideas while minimizing the risk of losing progress.

3. Enhance Collaboration:

To enable multiple contributors to work on the POC simultaneously, merging their efforts into a cohesive repository.

4. Validate Feasibility:

To showcase a functional prototype with traceable development steps, helping stakeholders evaluate the practicality and potential of the concept.

**Importance of Setting Up a Local Repository**

1. Version Control: Tracks changes and maintains a history of your code, allowing easy rollback and comparison.

2. Safe Experimentation: Provides a secure environment to test new ideas without affecting the main project.

3. Offline Development: Enables you to work without an internet connection and sync later with remote repositories.

4. Organized Workflow: Helps structure development with clear commits and systematic progress tracking.

5. Collaboration: Prepares for integration with remote repositories like GitHub, enabling teamwork and code sharing.

6. Backup and Recovery: Acts as a local backup, protecting your work from accidental loss or system failures.

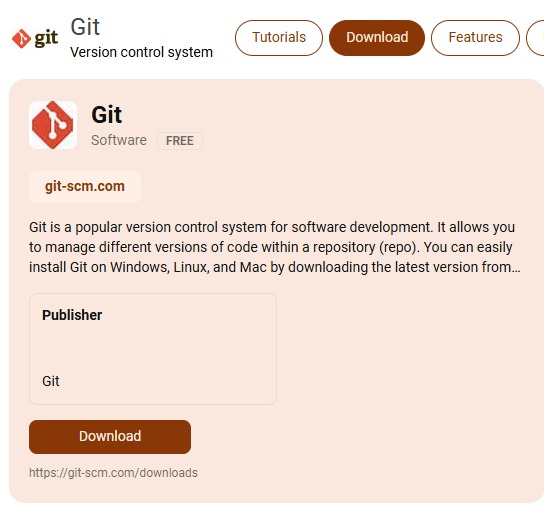
**STEP BY STEP OVERVIEW:**

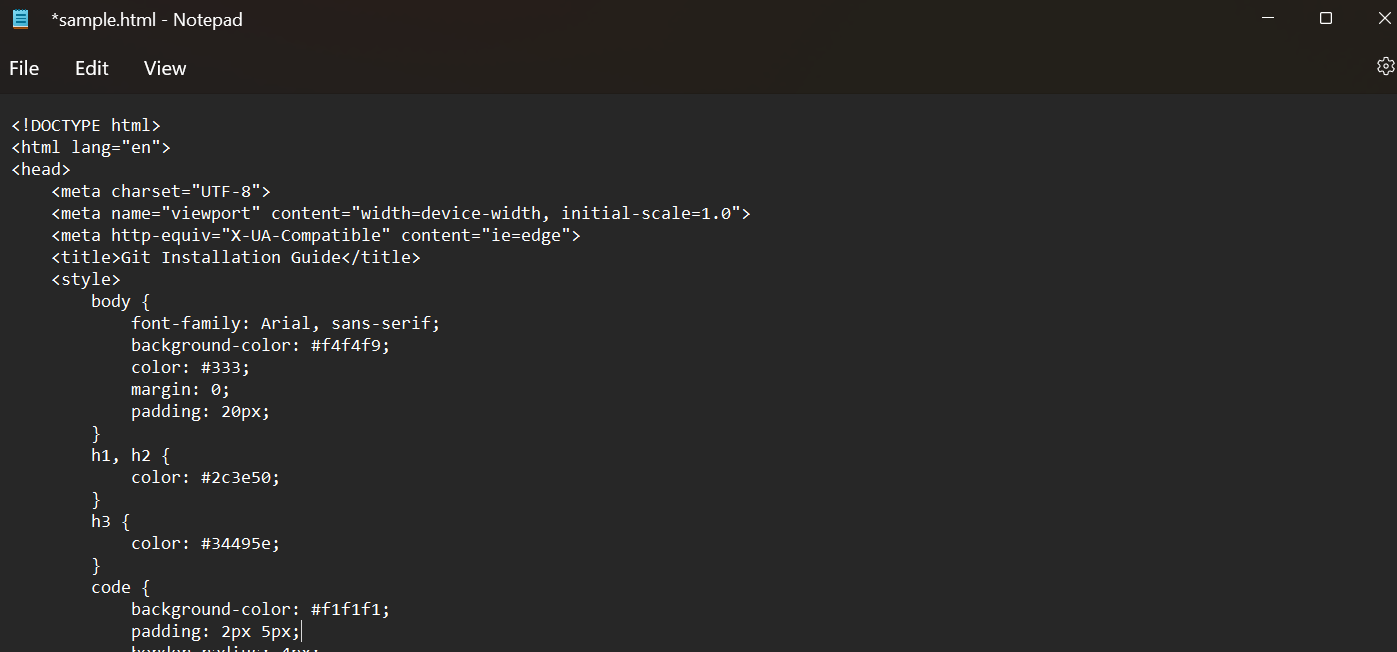
**1.download /install git:**



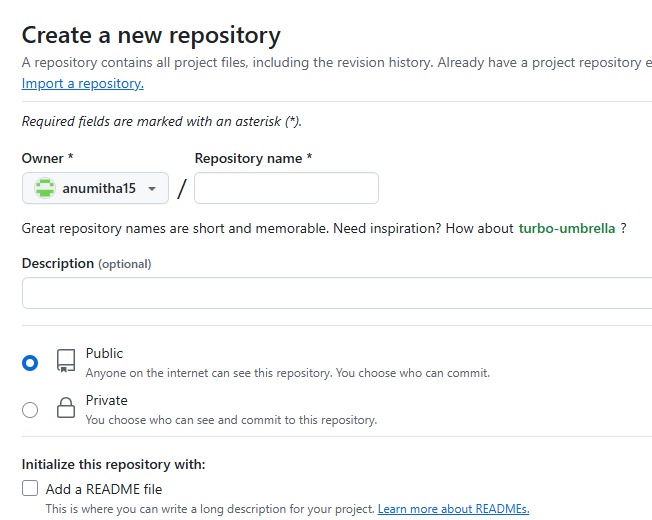
**Installation Methods:**

There are two primary ways to install Git on your system:**Using Command Line (Recommended for experienced users)Using the Official Website (For a graphical installation experience)**



**2.create a sample html file in a folder: **

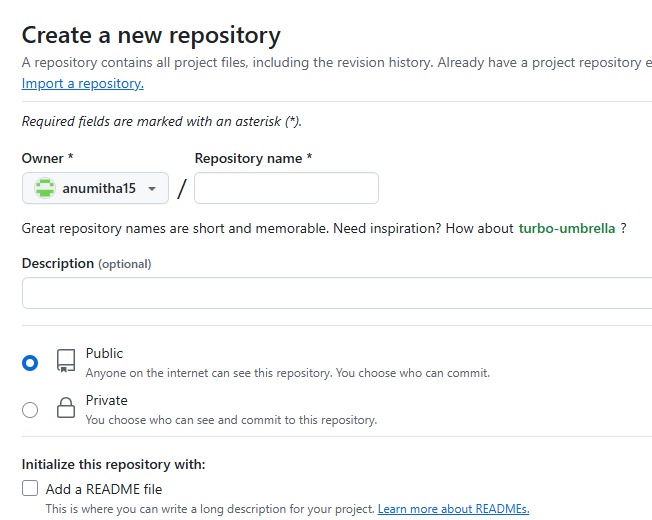
**3.create a new repository in your github account:**

****

**4.open the windows powershell**

**5.create a new repository:**

Make a new local repository to create a path for your folder to access the file in ur repository.

****

**6.Initialize the git repository:**

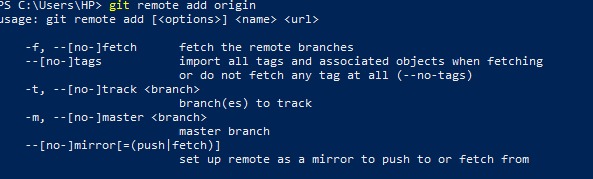
****

**7.Add the html file:**

****

Add the html file which u have created in ur file directory to the repository using this command.

**8.To link the local repository with remote origin:**



This remote origin will be the URL that Git uses to push and pull changes between your local repository and the remote repository.

**Overview of git commands:**

****

**Expected outcome:**

1. Initializing a Git Repository

* Creates a new local Git repository with a .git folder for version control.

2. Checking the Status

* Displays the state of the repository, showing staged, unstaged, and untracked files.

3.Staging Files

* Moves files to the staging area, preparing them for the next commit.

4. Committing Changes

* Saves changes to the repository with a commit ID, author, and message.

5. Viewing Commit History

* Lists past commits with IDs, authors, dates, and messages.