



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: Harshana Perianayaki B

Department: IT



INTRODUCTION

Version control is an essential tool for managing and tracking changes in software development projects. Git, a widely-used distributed version control system, allows developers to collaborate, maintain code history, and manage changes efficiently. This guide will walk you through the process of setting up a local Git repository to version control a static website, ensuring your project is well-organized and ready for collaboration or deployment.

OVERVIEW

This process is broken down into three main steps:

1. **Setting Up the Local Repository:** Learn how to use Git commands such as `git init`, `git add`, and `git commit` to initialize a repository and track changes locally.
2. **Creating a Remote Repository:** Create a GitHub repository to store your project online and prepare for collaboration or deployment.
3. **Connecting Local and Remote Repositories:** Use `git remote add origin` and `git push` commands to link the local repository with the remote one, enabling seamless synchronization of changes.

By following these steps, you'll establish a version-controlled workflow for your static website, making your project easier to maintain and share.

OBJECTIVES

By the end of this guide, you will be able to:

1. Initialize a local Git repository to manage your static website files.
2. Add and commit files to the repository, enabling version control.
3. Create a remote repository on GitHub and connect it to your local repository.
4. Push local changes to the remote repository for centralized storage and collaboration.

Importance of Setting Up a Local Git Repository

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

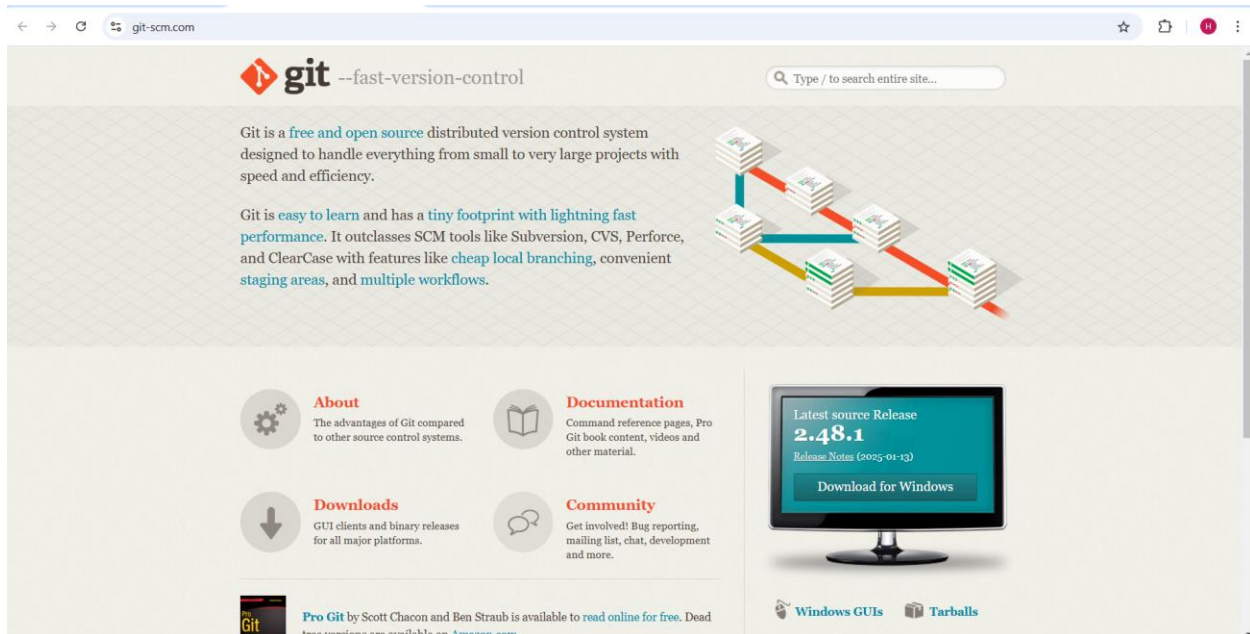
Rollback: Easily revert to previous versions to recover from mistakes.

Collaboration: Prepares your project for team work, enabling smooth integration of changes.

Step-by-Step Overview

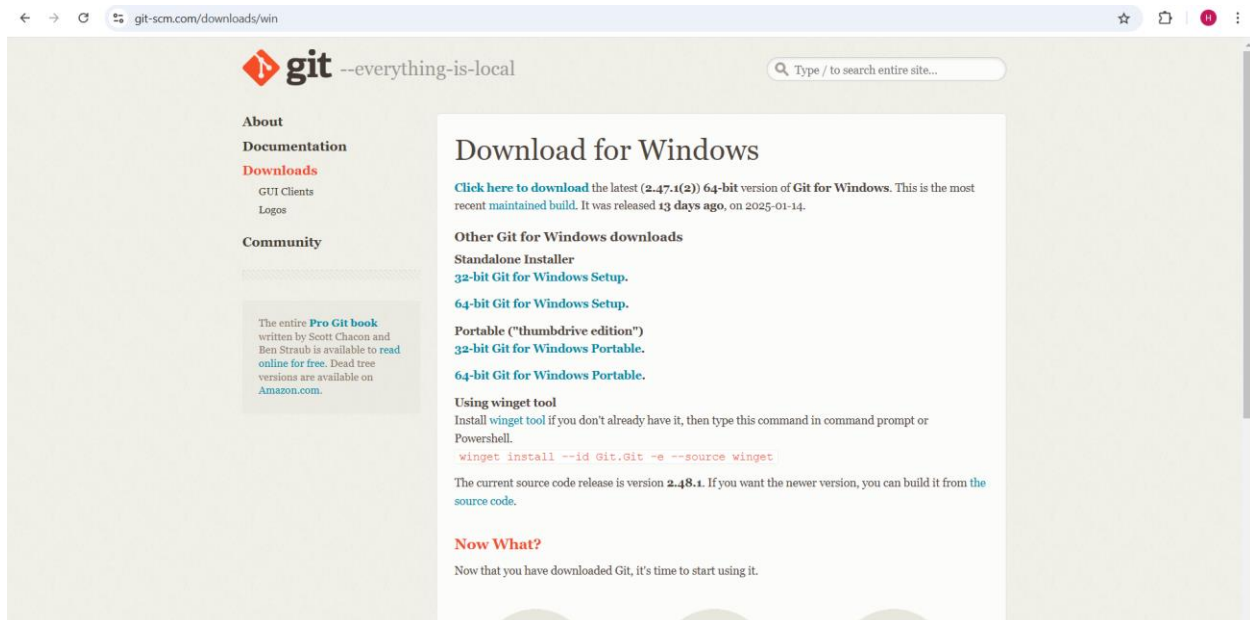
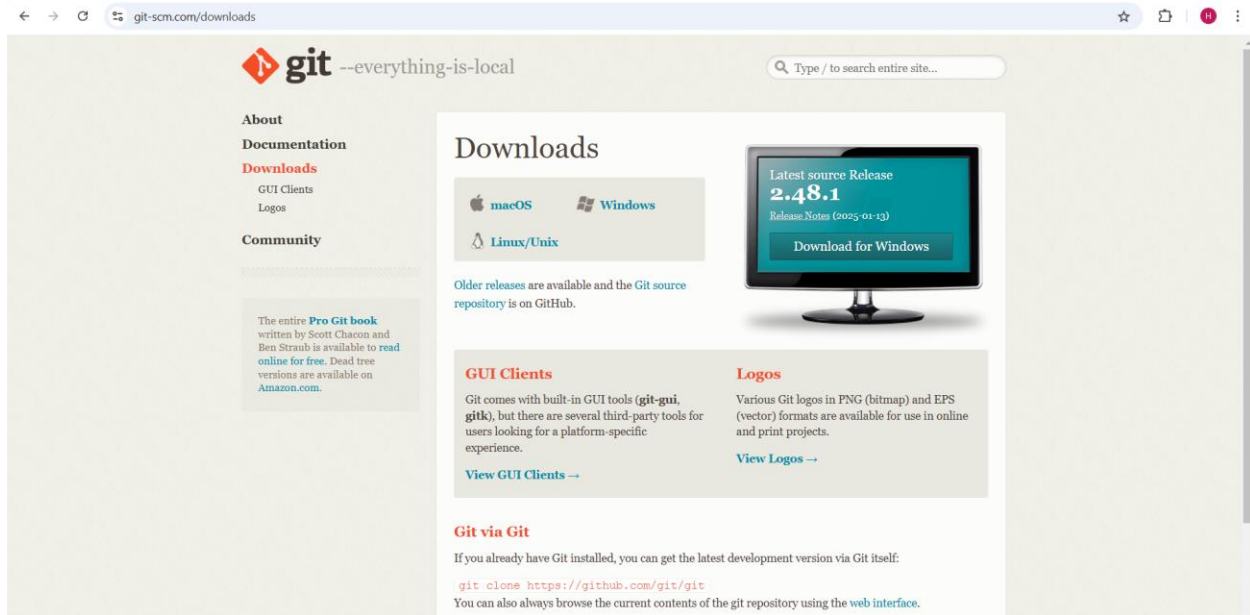
Step 1:

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



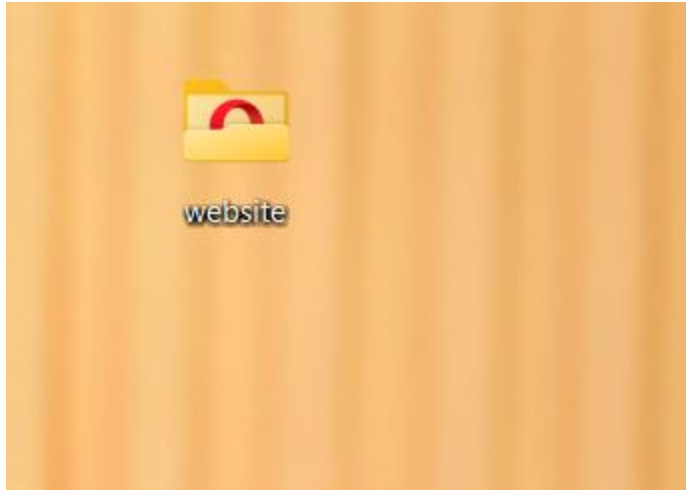
Step 2

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



Step 3

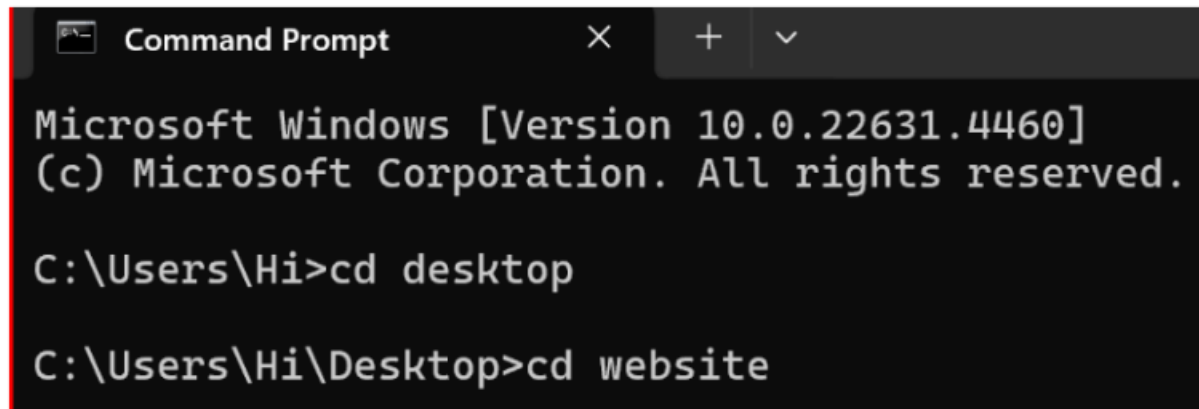
In your Desktop Create a folder named website for your static website
Inside that folder, create a simple HTML file named index.html. You
can write some basic HTML



```
index.html
File Edit View
a {
  color: #6200ea;
  text-decoration: none;
}
a:hover {
  text-decoration: underline;
}
</style>
</head>
<body>
  <header>
    <h1>Welcome to My Static Website</h1>
  </header>
  <main>
    <p>This is a simple static website created using HTML and CSS.</p>
    <p>Feel free to <a href="https://github.com" target="_blank">check out my GitHub</a> for more projects!</p>
  </main>
  <footer>
    <p>&copy; 2025 My Static Website. All rights reserved.</p>
  </footer>
</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 text inside shows the Windows version and copyright information, followed by two directory change commands: "cd desktop" and "cd website".

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

C:\Users\Hi>cd desktop

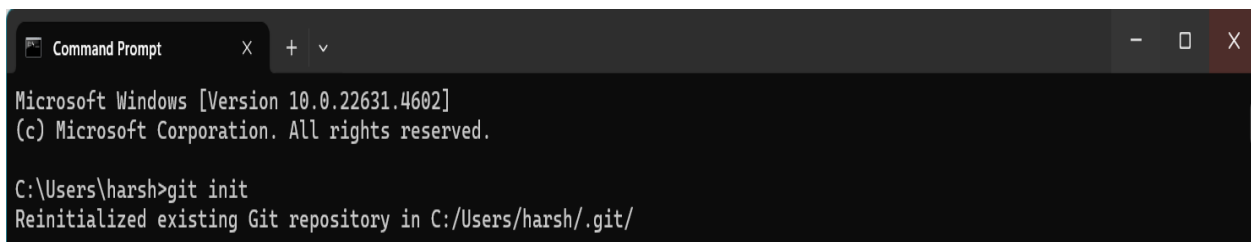
C:\Users\Hi\Desktop>cd website
```

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.

A screenshot of a Windows Command Prompt window. The title bar says "Command Prompt". The text inside shows the Windows version and copyright information, followed by the command "git init" and its output: "Reinitialized existing Git repository in C:/Users/harsh/.git/".

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\harsh>git init
Reinitialized existing Git repository in C:/Users/harsh/.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.


```
Command Prompt
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\harsh>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.

```
Command Prompt
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\harsh>git config --global user.name "Harshana Perianayaki B"
C:\Users\harsh>git config --global user.email "harshanab2006@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.

```
PS C:\Users\harsh\OneDrive\Desktop\class1 sample\class1> git commit -m "new "
[main (root-commit) b4fb111] new
1 file changed, 49 insertions(+)
create mode 100644 sample code.html
```

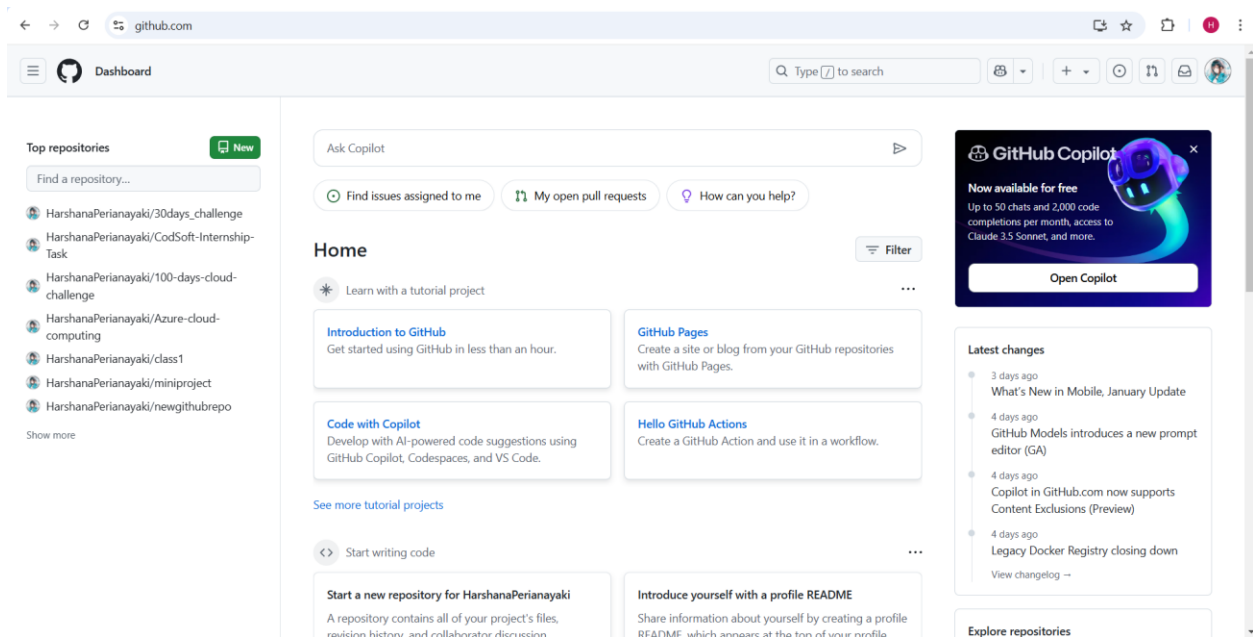
Step 10

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.

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.

```
PS C:\Users\harsh\OneDrive\Desktop\class1 sample\class1> git branch
* main
PS C:\Users\harsh\OneDrive\Desktop\class1 sample\class1> git checkout -b HarshanaPerianayaki
Switched to a new branch 'HarshanaPerianayaki'
PS C:\Users\harsh\OneDrive\Desktop\class1 sample\class1> git status
On branch HarshanaPerianayaki
nothing to commit, working tree clean
```

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

```
PS C:\Users\harsh\OneDrive\Desktop\class1 sample\class1> git push
fatal: unable to access 'https://github.com/HarshanaPerianayaki/class1.git/': Could not resolve host: github.com
PS C:\Users\harsh\OneDrive\Desktop\class1 sample\class1> git push origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
```

```
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 683 bytes | 170.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/HarshanaPerianayaki/class1.git
 * [new branch]      main -> main
```

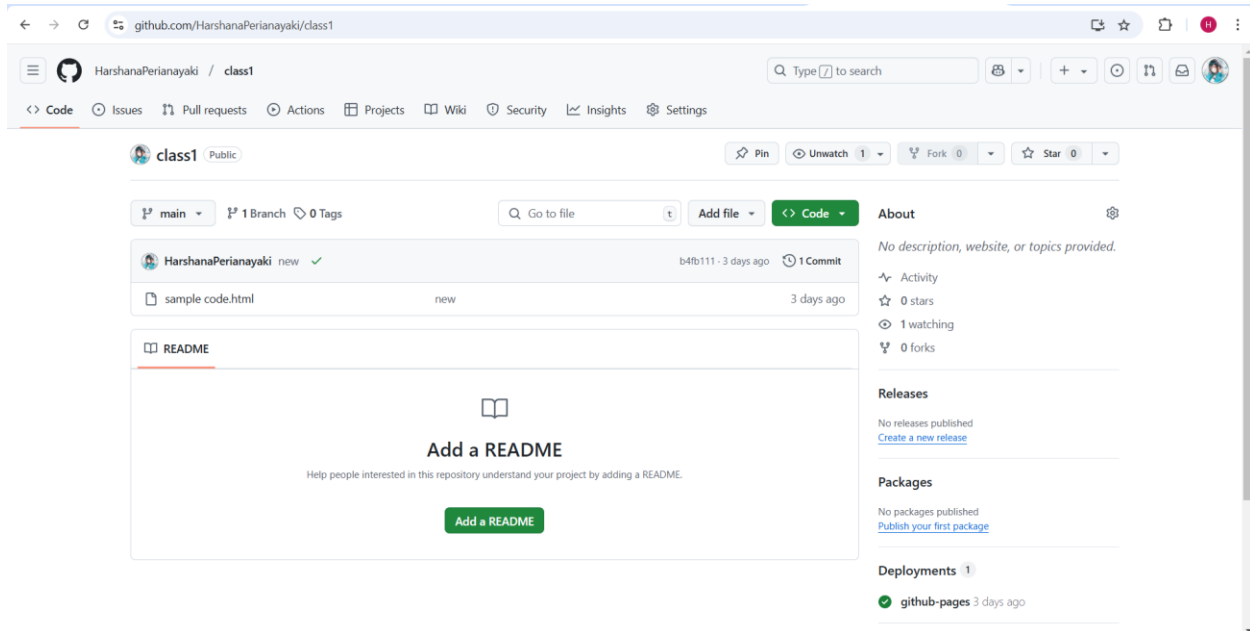
Step 14

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!



OUTCOME

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.

