



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: Shana R S

Department : CSE

Introduction:

This task shows you how to set up a local Git repository to track and manage changes for your static website. It makes it easy to collaborate, backup your work, and revert changes when needed.

Why Is This Task Important?

- **Track Changes:** You can see what changes were made and when.
- **Collaboration:** Multiple people can work on the project together.
- **Backup:** You have a safe copy of your project, so you can recover it if something goes wrong.
- **Organization:** It keeps your work structured and organized.

Steps Followed

- **Create a Folder:**
Set up a folder on your computer to store your website files.
- **Create an HTML File:**
Inside the folder, create an HTML file (e.g., index.html) that contains your website's content.
- **Initialize Git:**
Turn the folder into a Git repository to start tracking changes.
- **Stage Your File:**
Add your HTML file to the Git staging area so that it's ready to be saved.
- **Commit Your Changes:**
Save your changes with a commit message to record the current state of your project.
- **Check Git Status:**
Verify which files are tracked and ensure there are no uncommitted changes.
- **Connect to GitHub:**
Link your local repository to a remote repository on GitHub to enable online backup and collaboration.
- **Rename Branch to Main:**
Make sure your main working branch is correctly named “main” to follow modern conventions.
- **Push to GitHub:**
Upload your commits to the remote repository so that your project is stored online.

Where Can This Be Applied?

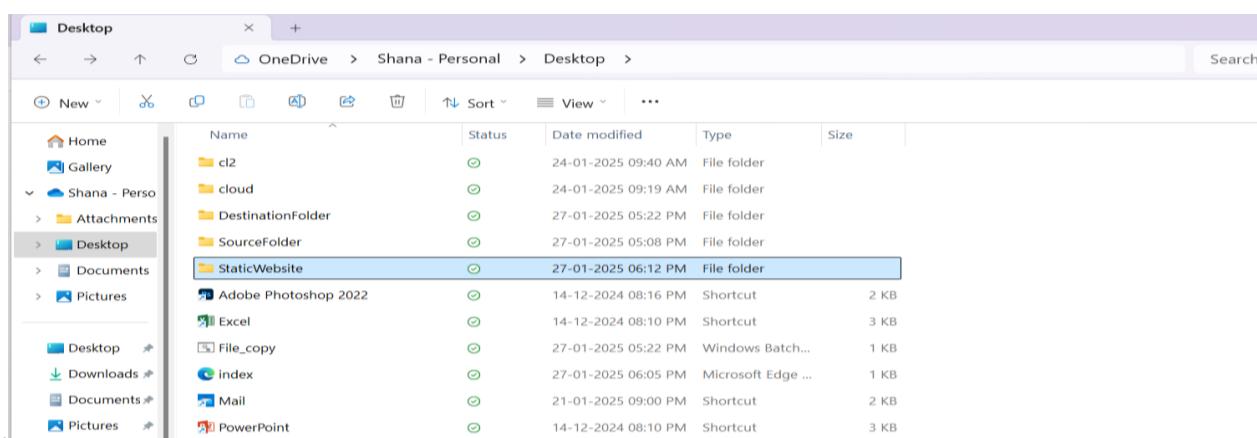
- **Web Development:** Manage and track changes to websites.
- **Software Projects:** Keep code organized and collaborate with team members.
- **Personal Projects:** Experiment with ideas without the fear of losing work.
- **Education:** Help students learn coding and collaboration through version control.

Step-by-Step Overview

Step 1: Create a Static Website Folder

Instruction:

- Create a folder on your local machine
(e.g., C:\Users\YourName\Desktop\staticwebsite).

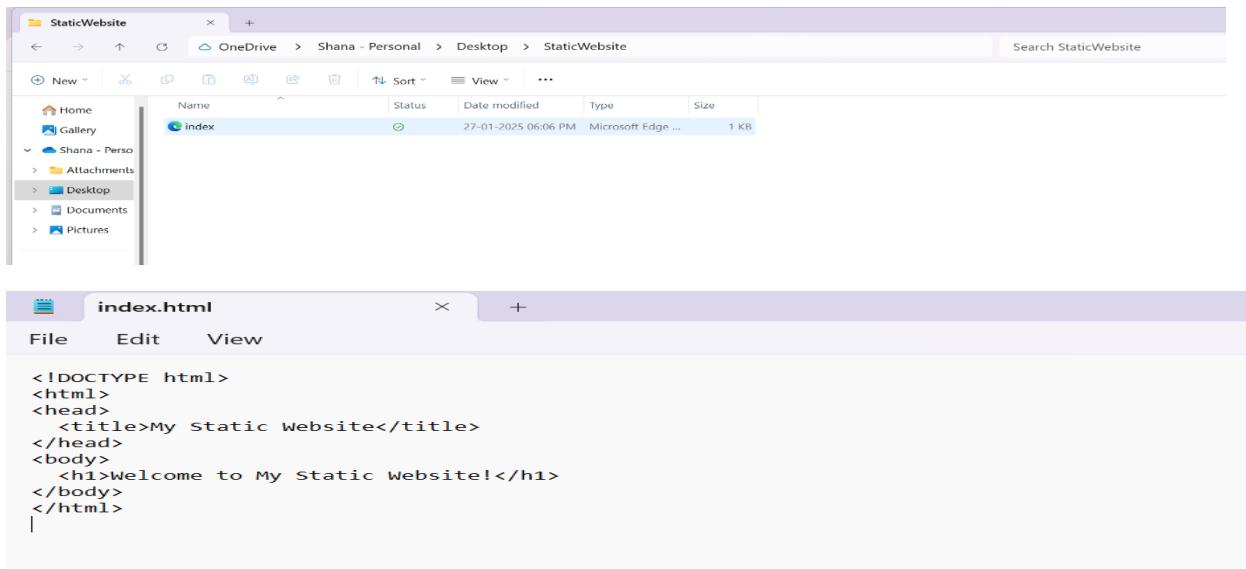


This folder houses all website files and serves as the repository for version control.

Step 2 : Create an HTML File Inside the Folder

Instruction:

- Inside the staticwebsite folder, create an index.html file with the following content:



This HTML file serves as the content of your website, and Git will track any changes made to it.

Step 3 : Initialize a Git Repository

Instruction:

- Open Command Prompt or Git Bash, navigate to the **staticwebsite** folder, and run:

`git init`

```
Microsoft Windows [Version 10.0.22621.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SmileE>d C:\Users\SmileE\OneDrive\Desktop\StaticWebsite
C:\Users\SmileE\OneDrive\Desktop\StaticWebsite>git init
Initialized empty Git repository in C:/Users/SmileE/OneDrive/Desktop/StaticWebsite/.git/
```

This command creates the necessary Git repository structure (the hidden .git folder) for version control.

Step 4 : Stage Your Website Files for Commit

Instruction:

- Stage the HTML file by running:

`git add .`

```
Microsoft Windows [Version 10.0.22621.4317]
(c) Microsoft Corporation. All rights reserved.

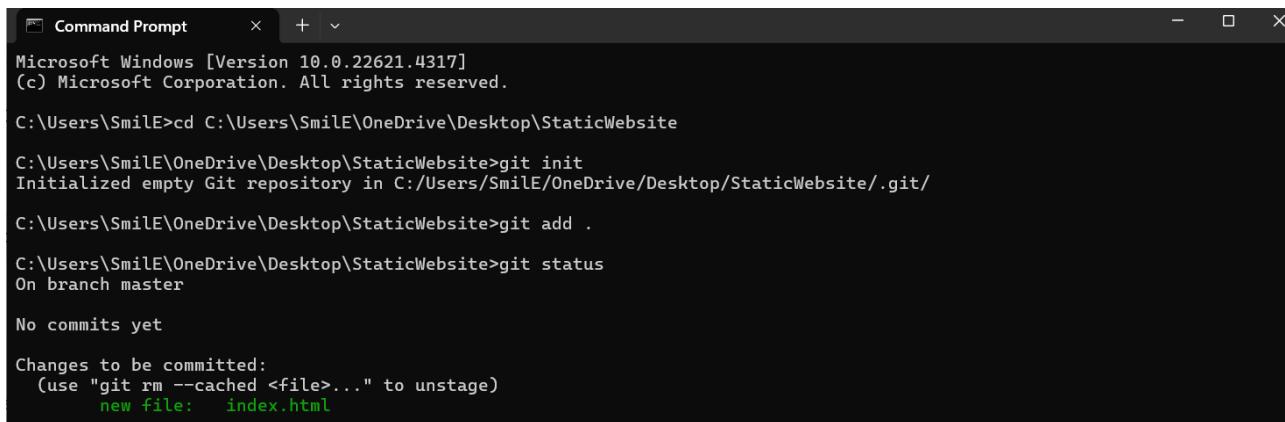
C:\Users\SmileE>d C:\Users\SmileE\OneDrive\Desktop\StaticWebsite
C:\Users\SmileE\OneDrive\Desktop\StaticWebsite>git init
Initialized empty Git repository in C:/Users/SmileE/OneDrive/Desktop/StaticWebsite/.git/
C:\Users\SmileE\OneDrive\Desktop\StaticWebsite>git add .
```

Staging prepares your file for commit, letting Git know which changes should be included in the next snapshot.

Step 5 : Check Git Status

Instruction:

- Verify which files are tracked and check for uncommitted changes:
`git status`



```
Microsoft Windows [Version 10.0.22621.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Smile>cd C:\Users\Smile\OneDrive\Desktop\StaticWebsite
C:\Users\Smile\OneDrive\Desktop\StaticWebsite>git init
Initialized empty Git repository in C:/Users/Smile/OneDrive/Desktop/StaticWebsite/.git/
C:\Users\Smile\OneDrive\Desktop\StaticWebsite>git add .
C:\Users\Smile\OneDrive\Desktop\StaticWebsite>git status
On branch master

No commits yet

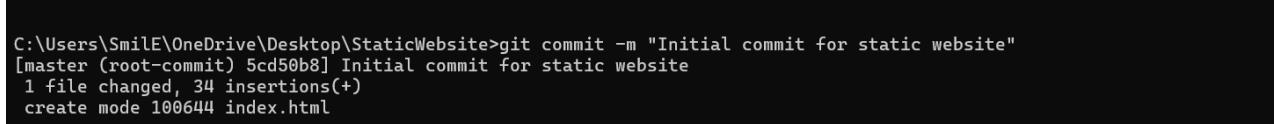
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   index.html
```

This command helps ensure that all intended files are being tracked and that there are no outstanding changes that need to be committed.

Step 6 : Commit the Changes

Instruction:

- Commit the staged file with a descriptive message:
`git commit -m "Initial commit of static website"`



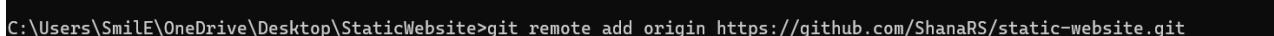
```
C:\Users\Smile\OneDrive\Desktop\StaticWebsite>git commit -m "Initial commit for static website"
[master (root-commit) 5cd50b8] Initial commit for static website
 1 file changed, 34 insertions(+)
 create mode 100644 index.html
```

Committing saves a snapshot of your project and records your changes with a message, which is crucial for tracking the project's history.

Step 7 : Connect Your Local Repository to GitHub

Instruction:

- Link your local repository to the remote GitHub repository:
`git remote add origin https://github.com/your-username/staticwebsite.git`



```
C:\Users\Smile\OneDrive\Desktop\StaticWebsite>git remote add origin https://github.com/ShanaRS/static-website.git
```

It creates a connection to your remote repository, allowing for easy pushing and pulling of changes.

Step 8 : Rename the Current Branch to Main

Instruction:

- Ensure the primary branch is correctly named:

```
git branch -M main
```

```
C:\Users\Smile\OneDrive\Desktop\StaticWebsite>git commit -m "Initial commit for static website"
[master (root-commit) 5cd50b8] Initial commit for static website
 1 file changed, 34 insertions(+)
 create mode 100644 index.html

C:\Users\Smile\OneDrive\Desktop\StaticWebsite>git remote add origin https://github.com/ShanaRS/static-website.git
origin  https://github.com/ShanaRS/static-website.git (fetch)
origin  https://github.com/ShanaRS/static-website.git (push)

C:\Users\Smile\OneDrive\Desktop\StaticWebsite>git branch -M main
```

Aligns your repository with modern conventions and ensures consistency for future interactions with GitHub.

Step 9 : Push Your Local Repository to GitHub

Instruction:

- Upload your commits to the remote repository:

```
git push -u origin main
```

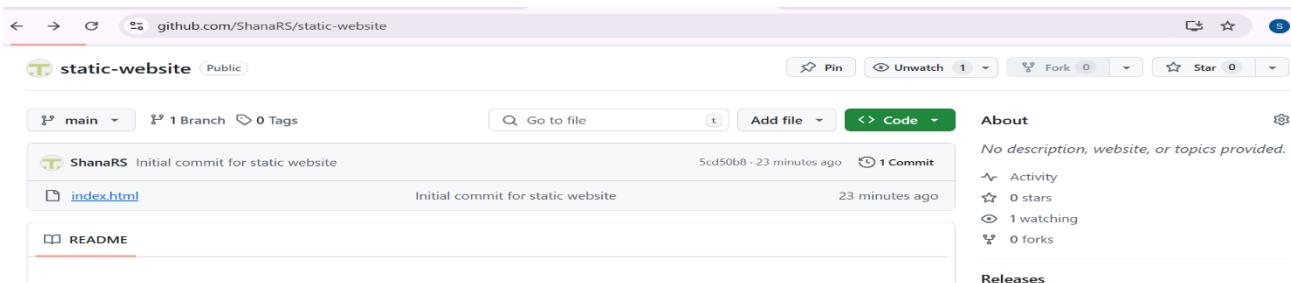
```
C:\Users\Smile\OneDrive\Desktop\StaticWebsite>git branch -M main
C:\Users\Smile\OneDrive\Desktop\StaticWebsite>git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 643 bytes | 643.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/ShanaRS/static-website.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

This step synchronizes your local work with GitHub, setting up tracking for future operations and providing an online backup of your project.

Step10: Check on GitHub

Instruction:

- Go to your GitHub repository page.
- Ensure your code (e.g., `index.html`) appears there.



Outcome

By completing this POC, you will:

- **Version-controlled your static website** using Git.
- **Connected your local repository to GitHub** for online backup.
- **Successfully pushed your files** to GitHub.
- **Learned key Git commands** for tracking and managing your project.