

Git and GitHub

ASSIGNMENT : 2(a)

- **Create version control account on GitHub and using Git Commands to create repository and push your code to GitHub.**

❖ What is Git?

- **Git** is a distributed version control system designed to handle projects of any size with speed and efficiency.
 - It tracks changes in source code, allowing multiple developers to collaborate.
 - **Purpose : Git** focuses on **local version control** and managing changes in code.
 - Key Features:
 - Version control
 - Branching and merging
 - Offline work capability
-

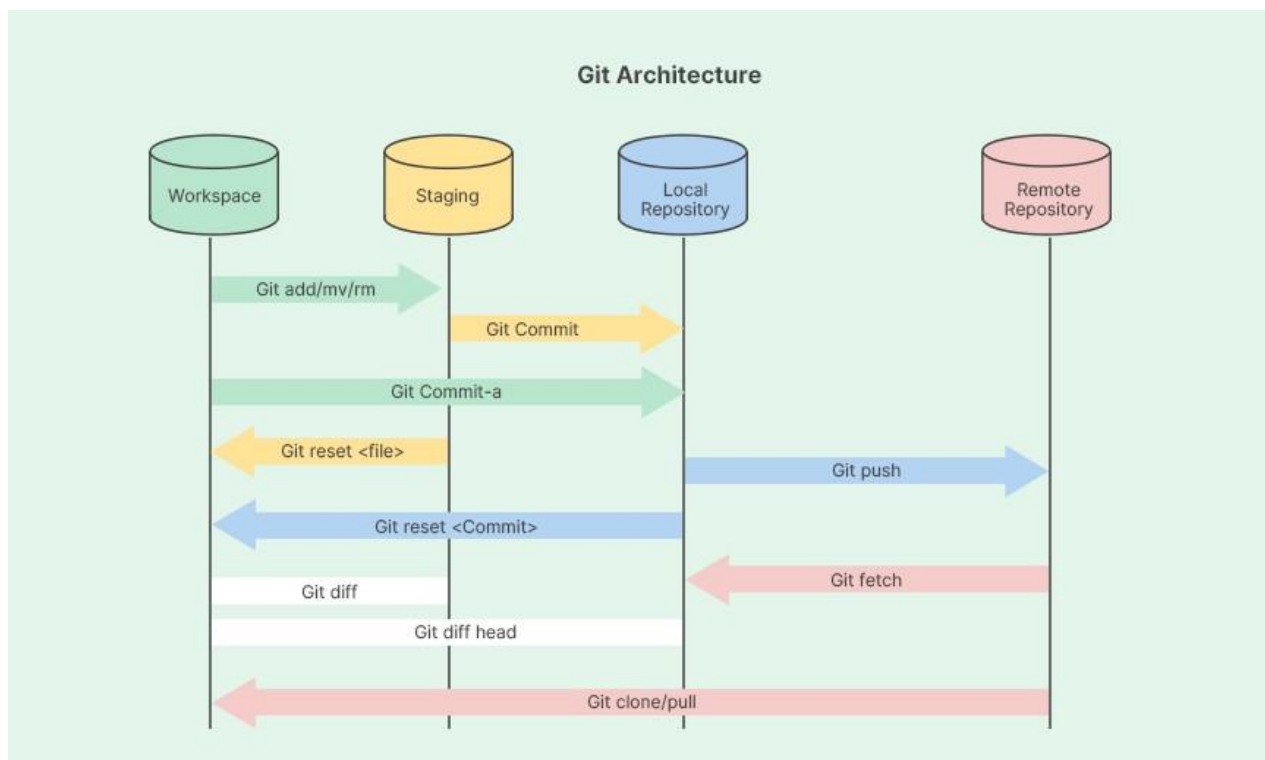
❖ What is GitHub?

- **GitHub** is a cloud-based platform for hosting Git repositories.
- It provides tools for collaboration, code review, and project management.
- **Purpose : GitHub** focuses on **remote hosting** and providing collaboration and project management tools.
- Features:
 - Remote repository hosting
 - Pull requests for collaboration
 - Issues for tracking bugs or tasks
 - Actions for automation and CI/CD

❖ Difference Between Git and GitHub

<u>Aspect</u>	<u>Git</u>	<u>GitHub</u>
Definition	A version control system.	A platform to host and manage Git repositories.
Usage	Works locally on your machine.	Requires an internet connection for hosting.
Functionality	Tracks code changes and allows collaboration.	Provides a GUI, collaboration tools, and hosting.
Dependency	Does not need GitHub to work.	Requires Git for version control.

❖ Git Architecture



Components of Git architecture

1. Working Directory

- The working directory is the place where you work on your project files.
 - It contains the actual files and directories of the project.
 - Any changes made here need to be staged and committed to be tracked by Git.
-

2. Staging Area (Index)

- The staging area is an intermediate space where changes are prepared before committing.
 - Files are added to the staging area using the `git add` command.
 - It acts as a preview of what will be included in the next commit.
-

3. Local Repository

- The local repository is the `.git` folder located inside your project directory.
 - It contains the entire history of the project, including commits, branches, and configurations.
 - Changes are saved to the local repository using the `git commit` command.
-

4. Remote Repository

- The remote repository is a version of your repository hosted on a remote server (e.g., GitHub, GitLab, Bitbucket).
 - It allows multiple developers to collaborate by sharing and synchronizing code.
-

5. Branches

- A branch is an independent line of development within a repository.
- The default branch is usually called `main` or `master`.
- Feature branches allow developers to work on separate features without affecting the main branch.

❖ Git Commands Executed on Git Bash :

```
MINGW64:/d/Sumit_Git-1

Slazer Electronics@SumitsDell MINGW64 /d/Sumit_Git-1 (main)
$ git init
Reinitialized existing Git repository in D:/Sumit_Git-1/.git/

Slazer Electronics@SumitsDell MINGW64 /d/Sumit_Git-1 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   First_1.txt

no changes added to commit (use "git add" and/or "git commit -a")

Slazer Electronics@SumitsDell MINGW64 /d/Sumit_Git-1 (main)
$ git add *

Slazer Electronics@SumitsDell MINGW64 /d/Sumit_Git-1 (main)
$ git commit -m "First Commit"
[main 8a79578] First Commit
1 file changed, 1 insertion(+)
```

```
MINGW64:/d/Sumit_Git-1

$ git add *

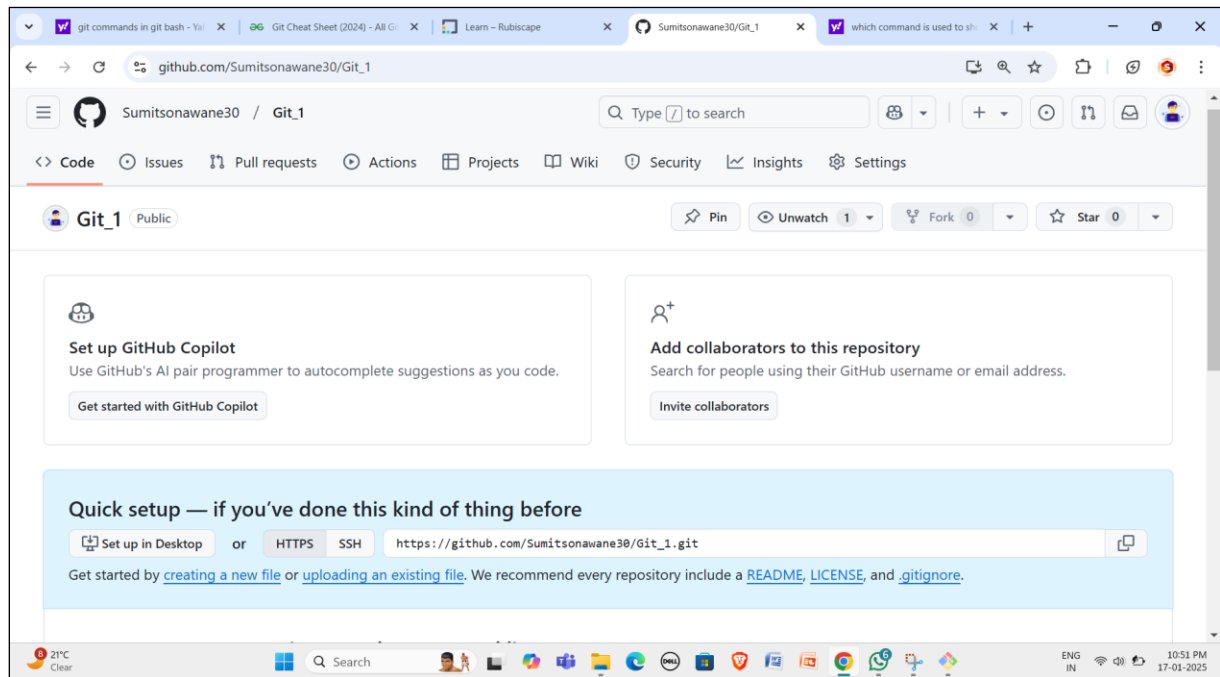
Slazer Electronics@SumitsDell MINGW64 /d/Sumit_Git-1 (main)
$ git commit -m "First Commit"
[main 8a79578] First Commit
1 file changed, 1 insertion(+)

Slazer Electronics@SumitsDell MINGW64 /d/Sumit_Git-1 (main)
$ git remote add origin https://github.com/Sumitsonawane30/Git_1.git
error: remote origin already exists.

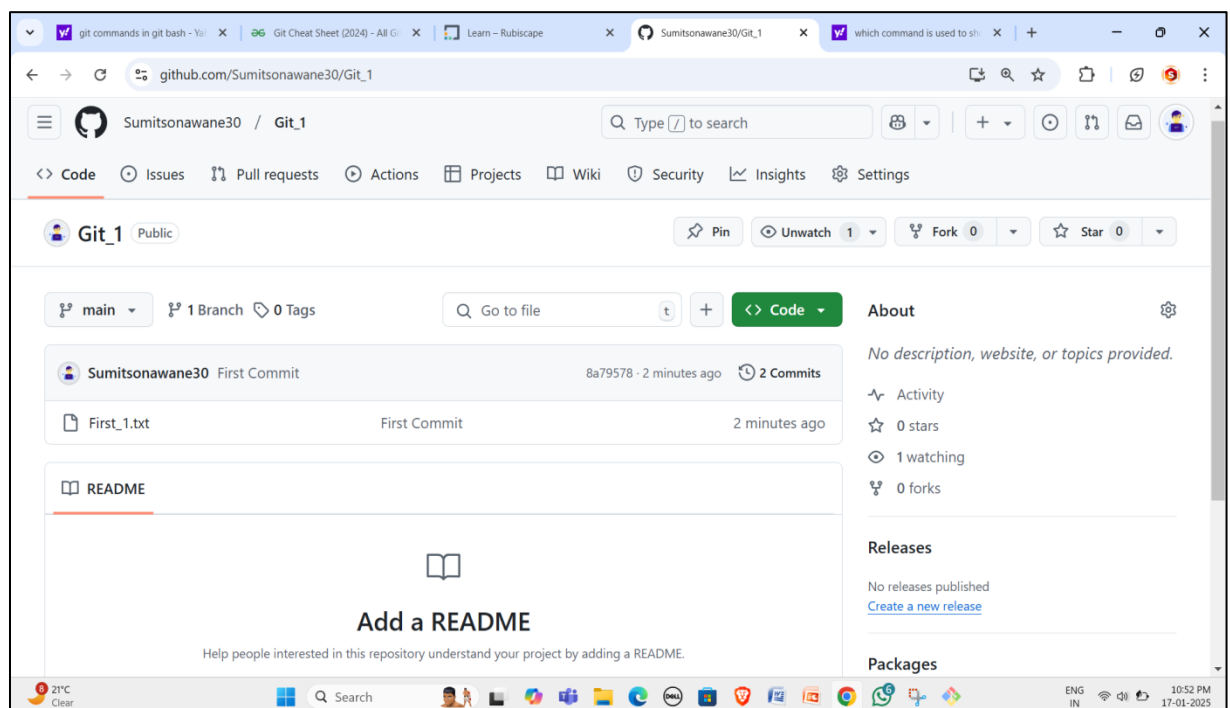
Slazer Electronics@SumitsDell MINGW64 /d/Sumit_Git-1 (main)
$ git push -u origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 467 bytes | 233.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Sumitsonawane30/Git_1.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

Slazer Electronics@SumitsDell MINGW64 /d/Sumit_Git-1 (main)
$
```

• Before Push

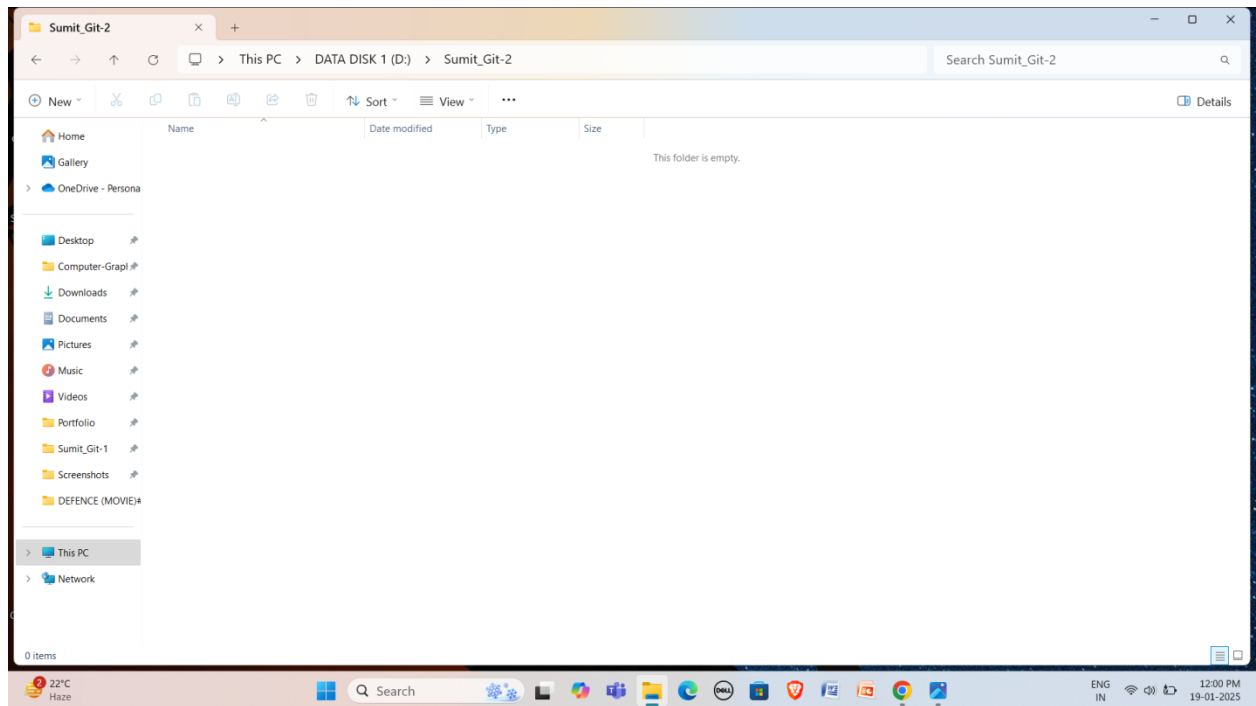


• After Push

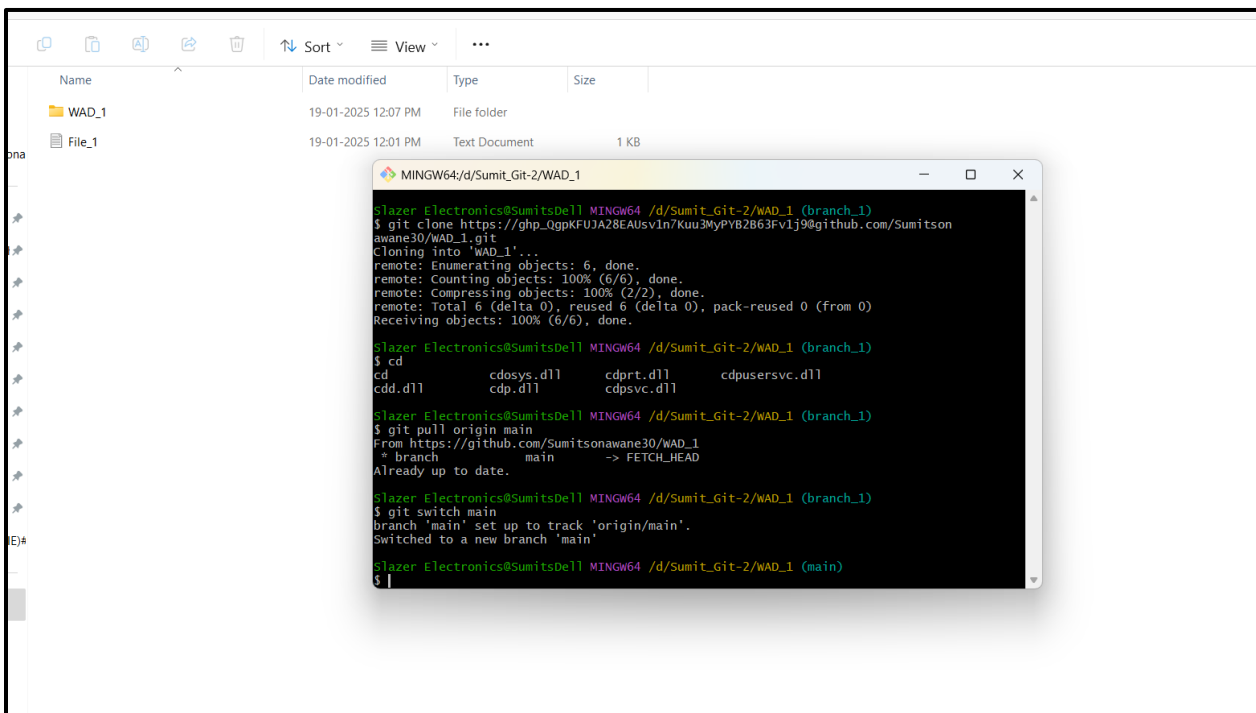


❖ Cloning the repository

- Before Cloning :



- After using Clone Command :



❖ Pull Command :

