Name : Sohail Ali Khwazada
Roll No. : 49
Batch : T13

Experiment 3 : GIT Operations

Aim :

To Perform various GIT operations on local and Remote repositories.

Theory :

Git is a powerful distributed version control system, and it offers a wide range of operations to manage code effectively. These operations can be performed on both **local repositories** (on your computer) and **remote repositories** (hosted on platforms like GitHub, GitLab, Bitbucket).

---

1. Git Repositories: Local vs. Remote

- **Local Repository:** This is the version of the repository on your computer. It contains the working directory, staging area, and the `.git` folder where Git stores all metadata and history.
- **Remote Repository:** This is a version of the repository hosted on a remote server (like GitHub). It allows multiple developers to collaborate on the same project.

---

2. Git Operations on Local Repositories

2.1. Creating a Local Repository

Initialize a New Repository:

```
git init
```

- This command creates a new `.git` directory in your current folder, initializing it as a Git repository.

Clone an Existing Repository:

```
git clone <repository_url>
```

- This command copies an existing remote repository to your local machine, including its history and branches.

2.2. Tracking Changes

Check the Status of Files:

```
git status
```

- This command shows the current status of the repository, including untracked files, changes to be committed, and changes not staged.

Track New Files (Stage Changes):

```
git add <file>
```

Stage a specific file:

```
git add index.html
```

Stage all changes:

```
git add .
```

Unstage Files:

```
git reset <file>
```

- This command removes a file from the staging area without deleting the changes in the working directory.

2.3. Committing Changes

Commit Staged Changes:

```
git commit -m "Your commit message"
```

- A commit represents a snapshot of your repository at a particular point in time.

Commit with Detailed Message:

```
git commit
```

- This opens the default text editor to write a detailed commit message.

Amend the Last Commit:

```
git commit --amend
```

- This allows you to modify the last commit, either to change the commit message or include new changes.

2.4. Branching and Merging

Create a New Branch:

```
git branch <branch_name>
```

Switch to a Branch:

```
git checkout <branch_name>
```

Create and Switch to a New Branch (Single Command):

```
git checkout -b <branch_name>
```

List All Branches:

```
git branch
```

Merge a Branch into the Current Branch:

```
git merge <branch_name>
```

Delete a Branch (Locally):

```
git branch -d <branch_name>
```

2.5. Undoing Changes

Undo Changes in Working Directory:

```
git checkout -- <file>
```

- This reverts the file to its last committed state.

Unstage Changes (Keep Changes in Working Directory):

```
git reset HEAD <file>
```

Remove the Last Commit (Preserve Changes):

```
git reset --soft HEAD~1
```

Completely Remove the Last Commit (Delete Changes):

```
git reset --hard HEAD~1
```

2.6. Viewing History

View Commit History

```
git log
```

To see a compact version:

```
git log --oneline
```

To view the history of a specific file:

```
git log -- <file>
```

Show Changes in a Commit:

```
git show <commit_hash>
```

---

3. Git Operations on Remote Repositories

3.1. Setting Up Remote Repositories

Add a Remote Repository:

```
git remote add origin <repository_url>
```

- This adds a remote named `origin` pointing to the given URL.

View Remote Repositories:

```
git remote -v
```

Remove a Remote Repository:

```
git remote remove origin
```

3.2. Pushing Changes to Remote

Push Changes to the Remote Repository:

```
git push origin <branch_name>
```

Example:

```
git push origin main
```

Push All Branches:

```
git push --all origin
```

Force Push (Overwrite Remote History):

```
git push --force
```

- Be careful with this command as it can overwrite history on the remote repository.

3.3. Pulling Changes from Remote

Pull Changes from a Remote Repository:

```
git pull origin <branch_name>
```

- This fetches the changes from the remote repository and merges them into your current branch.

Fetch Changes Without Merging:

```
git fetch origin
```

- This downloads updates from the remote repository without automatically merging them.

3.4. Working with Branches Remotely

Push a New Branch to Remote:

```
git push -u origin <branch_name>
```

Delete a Remote Branch:

```
git push origin --delete <branch_name>
```

Rename a Remote Branch:

```
git push origin :old-branch-name new-branch-name
```

3.5. Handling Merge Conflicts

When multiple people edit the same part of a file, Git may encounter conflicts when merging changes.

Identify Conflicts:

```
git status
```

- **Resolve Conflicts:**
    - Open the conflicted file(s).

Look for conflict markers:

```
<<<<<<< HEAD
(Your changes)
=======
(Incoming changes)
>>>>>>> branch-name
```

    - Edit the file to resolve conflicts, then save.

Mark as Resolved:

```
git add <file>
```

Complete the Merge:

```
git commit
```

---

4. Advanced Git Operations

4.1. Rebasing

Rebase a Branch:

```
git rebase <branch_name>
```

Interactive Rebase (Edit Commits):

```
git rebase -i HEAD~n
```

- This allows you to squash, reorder, or edit commits.

4.2. Stashing Changes

Stash Uncommitted Changes:

```
git stash
```

Apply Stashed Changes:

```
git stash apply
```

List All Stashes:

```
git stash list
```

4.3. Tagging Releases

Create a Tag:

```
git tag v1.0.0
```

Push Tags to Remote:

```
git push origin v1.0.0
```

List Tags:

```
git tag
```

---

5. Summary of Key Git Commands

Operation	Command
Initialize Repository	`git init`
Clone Repository	`git clone <url>`
Check Status	`git status`

Add Files	`git add <file>`
Commit Changes	`git commit -m "message"`
Create Branch	`git branch <branch>`
Switch Branch	`git checkout <branch>`
Merge Branches	`git merge <branch>`
Pull Changes	`git pull origin <branch>`
Push Changes	`git push origin <branch>`
View Commit History	`git log`
Stash Changes	`git stash`
Create Tag	`git tag <tag>`

Screenshots:

```
Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~
$ git config --global user.name
Sohail Ali Khwazada

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~
$ git config --global user.email
sohailali.khwazada2011@gmail.com

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~
$ cat ~/.gitconfig
[user]
        name = Sohail Ali Khwazada
        email = sohailali.khwazada2011@gmail.com
[filter "lfs"]
        clean = git-lfs clean -- %f
        smudge = git-lfs smudge -- %f
        process = git-lfs filter-process
        required = true
```

```
Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop
$ cd SEPM_T13_49/

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49
$ mkdir sepm_git_demo

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49
$ cd sepm_git_demo/

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/sepm_git_demo
$ git init
Initialized empty Git repository in C:/Users/Sohail Ali/Desktop/SEPM_T13_49/sepm_git_demo/.git/

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/sepm_git_demo (main)
$ ls -a
./  ../  .git/

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/sepm_git_demo (main)
$ ls -al
total 4
drwxr-xr-x 1 Sohail Ali 197121 0 Apr  4 19:03 ./
drwxr-xr-x 1 Sohail Ali 197121 0 Apr  4 19:03 ../
drwxr-xr-x 1 Sohail Ali 197121 0 Apr  4 19:03 .git/
```

```
Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/sepm_git_demo (main)
$ git status
On branch main

No commits yet

nothing to commit (create/copy files and use "git add" to track)

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/sepm_git_demo (main)
$ vi README.md
Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/sepm_git_demo (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md

nothing added to commit but untracked files present (use "git add" to track)

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/sepm_git_demo (main)
$ git add .
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
```

```
Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/sepm_git_demo (main)
$ git commit -m "initialize repository with a readme"
[main (root-commit) 7a75afd] initialize repository with a readme
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/sepm_git_demo (main)
$ git log
commit 7a75afd4b3c98debb9baa542adf0f5cfac1fb0ba (HEAD -> main)
Author: Sohail Ali Khwazada <sohailali.khwazada2011@gmail.com>
Date:   Fri Apr 4 19:12:32 2025 +0530

    initialize repository with a readme

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/sepm_git_demo (main)
$ git log --stat
commit 7a75afd4b3c98debb9baa542adf0f5cfac1fb0ba (HEAD -> main)
Author: Sohail Ali Khwazada <sohailali.khwazada2011@gmail.com>
Date:   Fri Apr 4 19:12:32 2025 +0530

    initialize repository with a readme

 README.md | 1 +
 1 file changed, 1 insertion(+)

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/sepm_git_demo (main)
$ git log --oneline^C

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/sepm_git_demo (main)
$ git log --oneline
7a75afd (HEAD -> main) initialize repository with a readme

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/sepm_git_demo (main)
$
```

```
Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/sepm_git_demo (main)
$ cd ..

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49
$ git clone "https://github.com/Sohail-Ali-Khwazada/AOA_LAB.git"
Cloning into 'AOA_LAB'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 24 (delta 2), reused 19 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (24/24), 10.73 KiB | 2.68 MiB/s, done.
Resolving deltas: 100% (2/2), done.

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49
$ ls
AOA_LAB/  T13_49_SEPM_Exp02.pdf  sepm_git_demo/

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49
$ cd AOA_LAB/

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/AOA_LAB (master)
$ ls
Dijkstra.cpp            Rabinkarp.cpp      kmpalgo.cpp    prims2.cpp        zerooneKnapsack.cpp
DisjointsetSnippet.cpp  Sorting1.cpp       kruskals.cpp   quickSort.cpp
FloydWarshall.cpp       binarySearch.cpp   mergeSort.cpp  sumofSubsets.cpp
Nqueens.cpp             jobSequencing.cpp  prims.cpp      tsp.cpp

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/AOA_LAB (master)
$ git remote -v
origin  https://github.com/Sohail-Ali-Khwazada/AOA_LAB.git (fetch)
origin  https://github.com/Sohail-Ali-Khwazada/AOA_LAB.git (push)

Sohail Ali@LAPTOP-PN9KM680 MINGW64 ~/desktop/SEPM_T13_49/AOA_LAB (master)
$ git pull
Already up to date.
```

Conclusion :

Thus, we have successfully studied and performed various GIT operations on local and Remote repositories.