

## Software Engineering & Project Management Lab Experiment No: - 03

**Aim:** To Perform various GIT operations on local and Remote repositories.

### **Theory:**

The command mkdir git creates a new directory (folder) named "git" in the current working directory. This command is used to make a new directory in a Unix-like operating system. The command cd git is used to change the current working directory to the directory named "git." After executing this command, any subsequent commands or file operations will occur within the "git" directory. "cd" stands for "change directory."

The git config --global user.name and git config --global user.email commands are used to set your global Git username and email address, respectively. They are part of the configuration settings in Git and are associated with the commits you make.

If you want to check your configuration settings, you can use the git config --list command to list all the settings Git can find at that point git commit -am "commit message" stages and commits all changes in tracked files with a commit message in a single command.

The command nano index.html opens the Nano text editor for the file named "index.html." Nano is a simple command-line text editor that allows you to view and edit files directly in the terminal.

The command touch teststatus creates an empty file named "teststatus" in the current directory. The touch command is commonly used to update the timestamps of a file or create an empty file if it doesn't exist.

git checkout -- teststatus: Discards changes to the file "teststatus" in the working directory. This reverts the file to the state it has in the last commit.

The git add command is used to stage changes in the working directory for the next commit in Git. It prepares modifications, additions, or deletions to be included in the upcoming commit.

The git log command is used to display the commit history of a Git repository. It shows a chronological list of commits, including commit hashes, author information, timestamps, and commit messages.

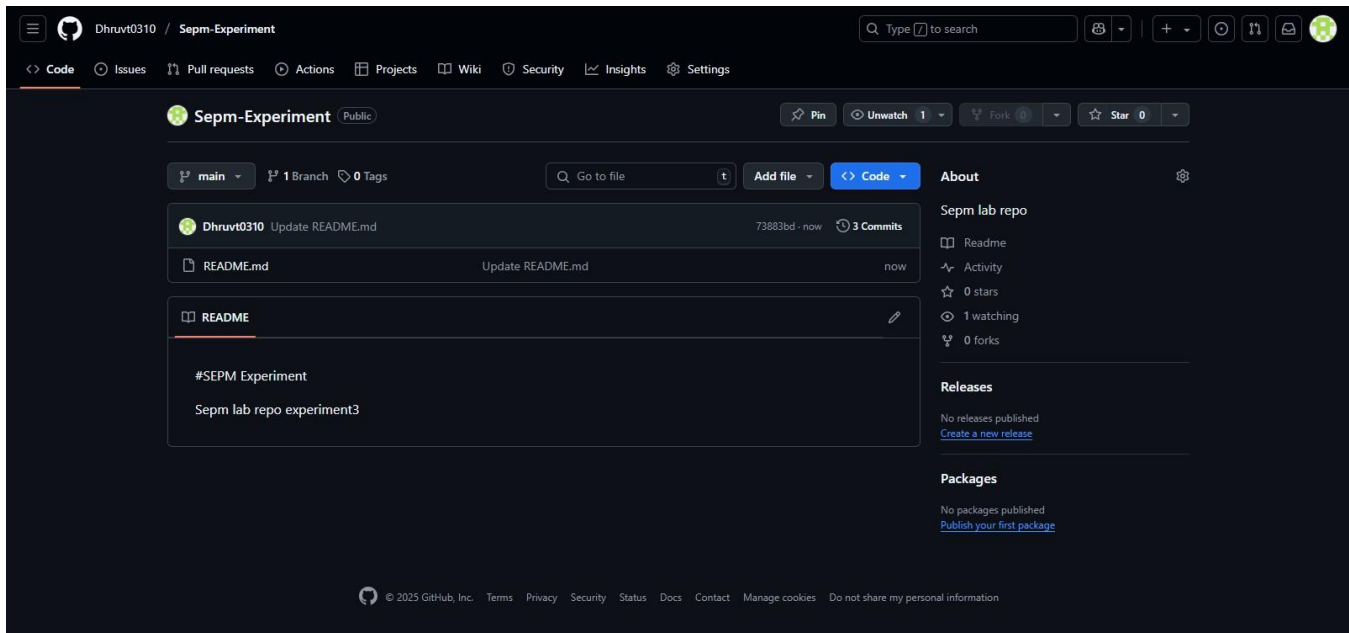
The command git log --oneline displays a simplified and concise one-line representation of the commit history in a Git repository, showing only the commit SHA-1 hash and the commit message.

The git clone command is used to create a copy of a Git repository. When you run this command, it duplicates the entire repository, including its files, commit history, and branches, and downloads it to your local machine. This is often the initial step when you want to work with a project hosted on a remote Git repository.

The git pull command is used to fetch and integrate changes from a remote repository into the current branch of your local repository. It combines two actions: it fetches the changes from the remote repository, and then it automatically merges those changes into your local branch. This is a convenient way to update your local repository with the latest changes from the remote repository. The git push command is used to upload or push the local changes in your Git repository to a remote repository. It updates the remote repository with the latest changes made in your local branch, making them accessible to others who share the same remote repository.

The git fetch command is used to retrieve changes from a remote repository. It fetches any new branches or changes made in the remote repository since your last interaction. However, it does not automatically merge these changes into your local branches. After using git fetch, you can inspect the changes and decide whether to integrate them using git merge or git rebase.

## Output:



```
MINGW64~/Users/151/git-SEPM-project/SEPM_EXP
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 271 bytes | 271.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/DX-STAR/SEPM_EXP.git
 * [new branch]      main -> main

151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (feature-branch)
$ git push origin feature-branch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-branch' on GitHub by visiting:
remote:   https://github.com/DX-STAR/SEPM_EXP/pull/new/feature-branch
remote:
To https://github.com/DX-STAR/SEPM_EXP.git
 * [new branch]      feature-branch -> feature-branch

151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (feature-branch)
$ git pull origin main
From https://github.com/DX-STAR/SEPM_EXP
 * branch            main       -> FETCH_HEAD
Already up to date.

151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (feature-branch)
$ git fetch
151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (feature-branch)
$ ^[[200-git merge origin/master
bash: $'\E[200-git': command not found

151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (feature-branch)
$ git merge origin/master
merge: origin/master - not something we can merge

151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (feature-branch)
$ |
```

```
MINGW64~/Users/151/git-SEPM-project/SEPM_EXP

151@203-12 MINGW64 ~/git-SEPM-project (main)
$ git clone https://github.com/DX-STAR/SEPM_EXP.git
Cloning into 'SEPM_EXP'...
warning: You appear to have cloned an empty repository.

151@203-12 MINGW64 ~/git-SEPM-project (main)
$ cd SEPM_EXP

151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (main)
$ echo "# SEPM Experiment" > README.md

151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (main)
$ touch README.md

151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (main)
$ nano README.md

151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (main)
$ git add .
git commit -m "Added README file"
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
[main (root-commit) 62af3ee] Added README file
 1 file changed, 3 insertions(+)
 create mode 100644 README.md

151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (main)
$ 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 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 271 bytes | 271.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/DX-STAR/SEPM_EXP.git
 * [new branch]      main -> main

151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

151@203-12 MINGW64 ~/git-SEPM-project/SEPM_EXP (main)
$ git checkout -b feature-branch
```

```
commit d2790bd543f84f6ea9842fa6c95b731a19931e81 (HEAD -> main)
Author: TL2_SEPM_EXP <work.devanggupta@gmail.com>
Date: Wed Mar 5 11:03:00 2025 +0530

    First Commit

15L@203-12 MINGW64 ~/git-SEPM-project (main)
$ git remote add origin https://github.com/your-username/your-repo.git
git remote show origin
remote: Repository not found.
fatal: repository 'https://github.com/your-username/your-repo.git/' not found

15L@203-12 MINGW64 ~/git-SEPM-project (main)
$ git remote add origin https://github.com/your-username/your-repo.git
git remote show origin
error: remote origin already exists.
fatal: User cancelled dialog.
error: unable to read askpass response from 'C:/Program Files/git/mingw64/bin/git-askpass.exe'
Username for 'https://github.com': DX-STAR
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for
information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/your-username/your-repo.git/'

15L@203-12 MINGW64 ~/git-SEPM-project (main)
$ git push -u origin master
error: src refspec master does not match any
error: failed to push some refs to 'https://github.com/your-username/your-repo.git'

15L@203-12 MINGW64 ~/git-SEPM-project (main)
$ AC

15L@203-12 MINGW64 ~/git-SEPM-project (main)
$ git status
On branch main
nothing to commit, working tree clean

15L@203-12 MINGW64 ~/git-SEPM-project (main)
$ git push -u origin main
info: please complete authentication in your browser...
remote: Repository not found.
fatal: repository 'https://github.com/your-username/your-repo.git/' not found

15L@203-12 MINGW64 ~/git-SEPM-project (main)
$ git clone https://github.com/DX-STAR/SEPM_EXP.git
Cloning into 'SEPM_EXP'...
warning: You appear to have cloned an empty repository.
```

```
15L@203-12 MINGW64 ~ (main)
$ git --version
git version 2.47.1.windows.2

15L@203-12 MINGW64 ~ (main)
$ git config --global user.name "TL2_SEPM_EXP"
git config --global user.email "work.devanggupta@gmail.com"

15L@203-12 MINGW64 ~ (main)
$ git config --global --list
color.ui=true
user.email=work.devanggupta@gmail.com
user.name=TL2_SEPM_EXP

15L@203-12 MINGW64 ~ (main)
$ mkdir git-SEPM-project
cd git-SEPM-project
git init
Initialized empty Git repository in C:/Users/15L/git-SEPM-project/.git/

15L@203-12 MINGW64 ~/git-SEPM-project (main)
$ touch index.html

15L@203-12 MINGW64 ~/git-SEPM-project (main)
$ git add .

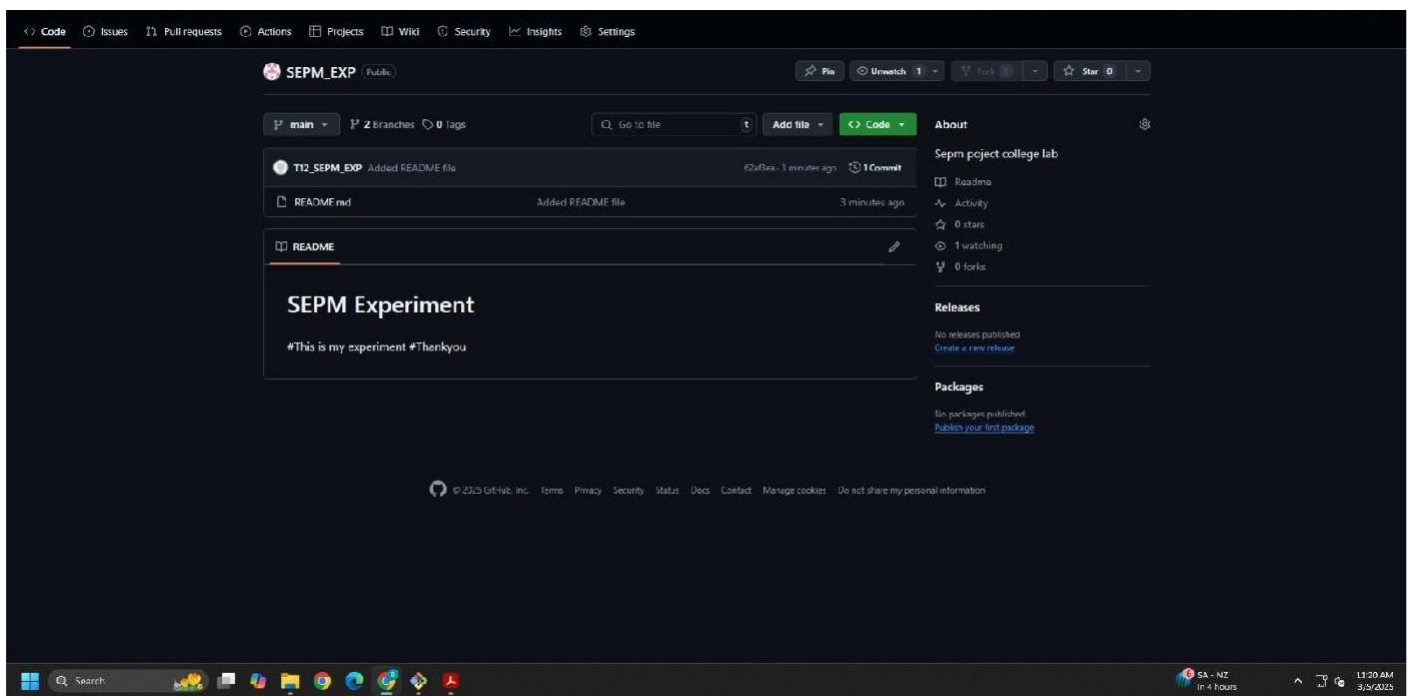
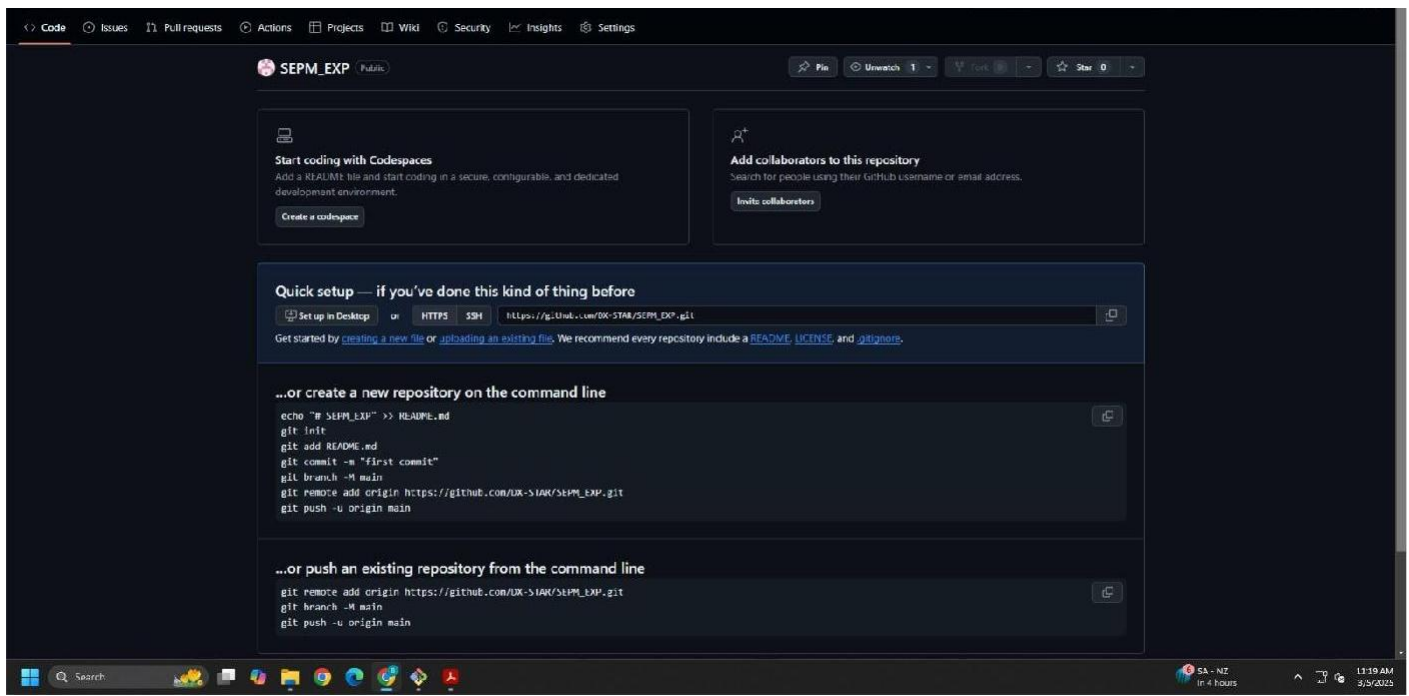
15L@203-12 MINGW64 ~/git-SEPM-project (main)
$ git commit -m "First Commit"
[main (root-commit) d2790bd] First Commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.html

15L@203-12 MINGW64 ~/git-SEPM-project (main)
$ git log --oneline
d2790bd (HEAD -> main) First Commit

15L@203-12 MINGW64 ~/git-SEPM-project (main)
$ git log
commit d2790bd543f84f6ea9842fa6c95b731a19931e81 (HEAD -> main)
Author: TL2_SEPM_EXP <work.devanggupta@gmail.com>
Date: Wed Mar 5 11:03:00 2025 +0530

    First Commit

15L@203-12 MINGW64 ~/git-SEPM-project (main)
$ git remote add origin https://github.com/your-username/your-repo.git
```



**Conclusion:** Thus, we have successfully Implemented GIT operations on local Remote repositories.

**LO Mapping:** LO2 is mapped.