**Name – Jay Deshmukh**

**Roll No. - 126**

**Batch – T23**

EXPERIMENT 3

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

# Theory :

**Directory and File Commands**

## mkdir git

Creates a new directory (folder) named "git" in the current working directory. This command is used to create a new directory in Unix-like operating systems.

## cd git

Changes the current working directory to the directory named "git." After executing this command, all subsequent commands or file operations will occur within the "git" directory.

**Note**: "cd" stands for "change directory."

## 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.

## 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.

# Git Configuration

## git config --global user.name "Your Name"

Sets your global Git username, which will be associated with your commits.

## git config --global user.email "[youremail@example.com](mailto:youremail@example.com)"

Sets your global Git email address, which will be associated with your commits.

## git config --list

Displays the current configuration settings for Git, including user details and other preferences.

# Staging and Committing Changes

## git add <file>

Stages changes in the working directory for the next commit in Git. It prepares modifications, additions, or deletions to be included in the upcoming commit.

## git commit -am "commit message"

Stages and commits all changes in tracked files with a commit message in a single command. This is a shorthand for git add <file> followed by git commit.

## git log

Displays the commit history of the Git repository, showing a chronological list of commits, including commit hashes, author information, timestamps, and commit messages.

## git log --oneline

Displays a simplified, one-line representation of the commit history, showing only the commit SHA-1 hash and the commit message.

# Git Remote Operations

## git clone <repository>

Creates a copy of a Git repository. This command duplicates the entire repository (files, commit history, branches) and downloads it to your local machine. It is often the initial step when working with a project hosted on a remote Git repository.

## git pull

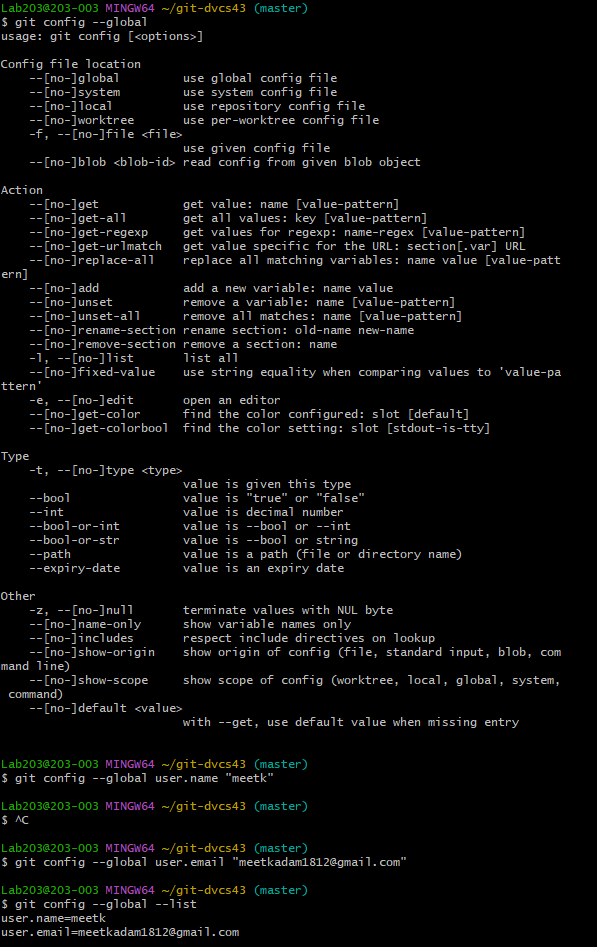
Fetches and integrates changes from a remote repository into the current branch of your local repository. It combines two actions: it fetches changes from the remote repository and merges them into your local branch.

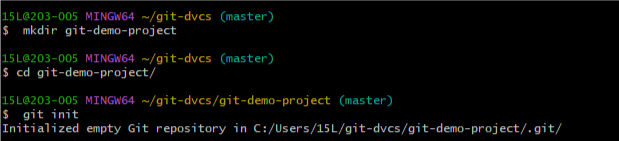
## git push

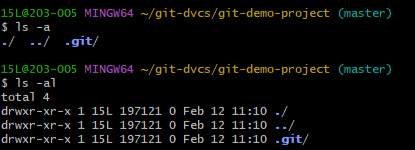
Uploads or pushes local changes from your Git repository to a remote repository. It updates the remote repository with the latest changes made in your local branch.

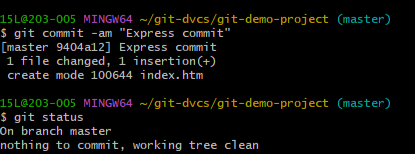
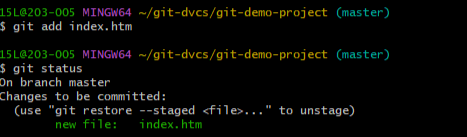
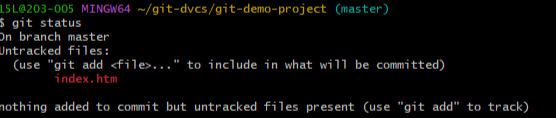
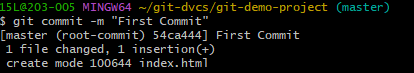
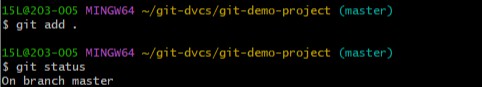
## git fetch

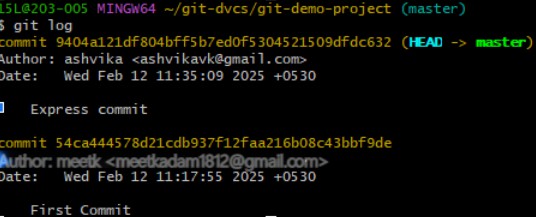
Retrieves changes from a remote repository, including new branches or changes made since your last interaction. However, it does not automatically merge these changes into your local branches. After using git fetch, you can inspect the changes and decide whether to integrate them using git merge or git rebase.

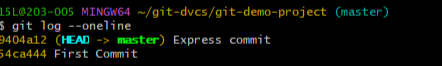












**Conclusion :** Successfully performed various GIT operations on local and Remote repositories.