EXPERIMENT NO:2 **DATE:**

Git and GitHub Commands

Aim:

Explore the Git and GitHub Commands.

Description:

- **git help:** Take help from the Git help section for different commands and other errors.
- **git config:** To set the basic configurations on Git like your name and email.
- **git init:** To create a local git repository for us in our store folder. This will help to manage the git commands for that particular repository.
- **git status:** To see what's changed since the last commit. It shows all the files that have been added and modified and are ready to be committed and files that are untracked.
- **git commit:** To commit our changes (taking a snapshot) and provide a message to remember for future reference.
- **git log:** To check the history of commits for our reference.
- **git add:** To add a specific list of files to the staging area.
- **git push:** To push all the contents of our local repository that belong to the master branch to the server (Global repository).
- **git branch:** To see all the branches present and current branches that we are working on.

- **git fetch:** To fetch down any changes from the global repository to the current repository.

- **git version:** It is used to show the current version of Git.

- **git checkout:** The git checkout command to switch and existing branch (or) to create and switch to a new one.

- **git pull:** The contents of the remote repository are fetched and integrated into your local repository using this command.

- **git merge:** The command git merge joins your branch to the parent branch. Depending on your local process, the present parent branch can be either a development branch or a master branch.

- **git clone:** To make a local copy of the global repository in your system (git clone command downloads the repository and creates a remote named origin which can be checked by the command – git remote -v).

- **git remove rm:** To remove a remote from our local repository.

Conclusion:

The listed Git commands are essential for managing repositories, tracking changes, and collaborating effectively. They cover setup, version control, branch management, and syncing between local and remote repositories.