PRN: **121A3008**

Experiment No: 2

AIM: Installation of Git and Working with local and Remote Repository.

THEORY: GitHub is a global company that provides hosting for software development version control using Git. It is a Subsidiary of Microsoft, which acquired the company in 2018 for $7.5 billion. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

Commands: 1) Git

config command

This command configures the user. The Git config command is the first and necessary command used on the Git command line. This command sets the author name and email address to be used with your commits. Git config is also used in other scenarios. **Syntax**

$ git config --global user.name "ImDwivedi1"

$ git config --global user.email "Himanshudubey481@gmail.com"

2) Git Init command

This command is used to create a local repository.

3) Git add command

This command is used to add one or more files to staging (Index) area. $ git add filename

PRN: **121A3008**

To add more than one file
$ git add*

4) Git commit command
This command changes the head. It records or snapshots the file permanently in the version history with a message.

$ git commit -m " Commit Message"

5) Git status command
The status command is used to display the state of the working directory and the staging area. It allows you to see which changes have been staged, which haven't, and which files aren?t being tracked by Git. It does not show you any information about the committed project history. For this, you need to use the git log. It also lists the files that you've changed and those you still need to add or commit.

$ git status

6) Git push Command

It is used to upload local repository content to a remote repository. Pushing is an act of transfer commits from your local repository to a remote repo. It's the complement to git fetch, but whereas fetching imports commits to local branches on comparatively pushing exports commits to remote branches. Remote branches are configured by using the git remote command. Pushing is capable of overwriting changes, and caution should be taken when pushing.

This command sends the changes made on the master branch, to your remote repository.

$ git push [variable name] master

7) Git pull command
Pull command is used to receive data from GitHub. It fetches and merges changes on the remote server to your working directory.

$ git pull URL

8) Git log Command
This command is used to check the commit history.

$ git log

PRN: **121A3008**

9) Git remote Command

Git Remote command is used to connect your local repository to the remote server. This command allows you to create, view, and delete connections to other repositories. These connections are more like bookmarks rather than direct links into other repositories. This command doesn't provide realtime access to repositories.

10) Git concatenate command

This command is simply use to display one or multiple files, $ cat filename

This command is used to add contents to file

$ cat >>filename

This command is used to append lines to file

$ cat > filename

11) git touch command

creates a new file and adds it to the current index of a git repository[1]. It is similar to using the touch command followed by the git add command. For example, you can use git touch hello.txt to create and add a file named hello.txt to your repository. You can also specify multiple files to create and add at once, such as

$ git touch file1 file2 file3.

Setting up local repository:

```
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (master)
$ git init
Initialized empty Git repository in C:/Users/Mangesh/Desktop/All_PROJECTS/DOLlab
/.git/

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (master)
$ git status
On branch master

No commits yet

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

Adding files to git and committing:

PRN: **121A3008**

```
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (master)
$ nano f1.txt

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (master)
$ nano f2.txt
```

Checking the status of untracked files

```
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        f1.txt
        f2.txt

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

Adding files to staging area

```
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   f1.txt
        new file:   f2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Exp1/
```

Committing changes and Detecting changed files:

PRN: **121A3008**

```
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (master)
$ git commit -m 'f1.txt commited made' f1.txt
warning: in the working copy of 'f1.txt', LF will be replaced by CRLF the next time Git touches it
[master (root-commit) cfb960f] f1.txt commited made
 1 file changed, 1 insertion(+)
 create mode 100644 f1.txt

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   f2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Exp1/
```

Adding remote Repository

```
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git remote add origin https://github.com/AryaAngane11/DevOpsLab.git

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 218 bytes | 218.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/AryaAngane11/DevOpsLab.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

```
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git pull https://github.com/AryaAngane11/DevOpsLab.git
From https://github.com/AryaAngane11/DevOpsLab
 * branch            HEAD       -> FETCH_HEAD
Already up to date.
```

```
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (master)
$ git log
commit cfb960fca8ff64c90116035d095d1faa24729dd8 (HEAD -> master)
Author: AryaAngane <aryaangan@gmail.com>
Date:   Sun Jul 30 14:11:09 2023 +0530

    f1.txt commited made
```

Conclusion: Hence we have successfully installed Git and worked with local and Remote Repository.