# ASSIGNMENT – 1

# Git and GitHub

## Task 1

- Demonstrate minimum 15 basic Git command with explanation and screenshot.

**Example**

git status

```
F:\Production\NER_NLP>git status
On branch master
Your branch is up to date with 'origin/master'.

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

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

Ans.

1. **git init** ➔ initialize an existing directory as a Git repository.

```
91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment
$ git init
Initialized empty Git repository in C:/Users/91800/Desktop/Assignment/.git/

91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (master)
$
```

2. **git status** ➔ It gives us all the necessary information about the current branch. Information like:
    - Whether the current branch is upto date.
    - Whether there is anything to commit, push or pull.
    - Whether there are files staged, unstaged or untracked.
    - Whether there are files created, modified or deleted.

```
91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (master)
$ git status
On branch master

No commits yet

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

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

3. **git add [file_name]** ➔ This command adds one file to staging area.

```
91800@DESKTOP-RSASM2O MINGW64 ~/desktop/Assignment (master)
$ git add app.py

91800@DESKTOP-RSASM2O MINGW64 ~/desktop/Assignment (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   app.py

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

**git add .** ➔ This command adds one or more files to staging area.

```
91800@DESKTOP-RSASM2O MINGW64 ~/desktop/Assignment (master)
$ git add .

91800@DESKTOP-RSASM2O MINGW64 ~/desktop/Assignment (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md
        new file:   app.py
        new file:   hello.py


91800@DESKTOP-RSASM2O MINGW64 ~/desktop/Assignment (master)
$
```

4. **git clone [url]** ➔ This command is used to obtain a repository from an existing URL.

```
91800@DESKTOP-RSASM2O MINGW64 ~/desktop
$ git clone https://github.com/mohitmahi004/Ineuron-projects.git
Cloning into 'Ineuron-projects'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

91800@DESKTOP-RSASM2O MINGW64 ~/desktop
$
```

5. **git commit -m "[ Type in the commit message ]"** ➔ This command records the file permanently in the version history.

```
91800@DESKTOP-RSASM2O MINGW64 ~/desktop/Assignment (master)
$ git commit -m " This is My first commit "
[master (root-commit) 828bf94]  This is My first commit
 3 files changed, 3 insertions(+)
 create mode 100644 README.md
 create mode 100644 app.py
 create mode 100644 hello.py
```

6. **git reset** ➔ This command unstage the files from staging area while keeping the file changes.

```
91800@DESKTOP-RSASM2O MINGW64 ~/desktop/Assignment (master)
$ git reset

91800@DESKTOP-RSASM2O MINGW64 ~/desktop/Assignment (master)
$ git status
On branch master
nothing to commit, working tree clean

91800@DESKTOP-RSASM2O MINGW64 ~/desktop/Assignment (master)
$ ls
README.md  app.py  hello.py
```

7. **git --version** ➔ This command shows which version of git you are using.

```
91800@DESKTOP-RSASM2O MINGW64 ~/desktop/Assignment (master)
$ git --version
git version 2.37.3.windows.1
```

8. **git branch** ➔ This command shows all the branches.

```
91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (main)
$ git branch
* main
```

9. **git branch -M main**➔ This command is used to change it from master branch to main branch as GitHub by default uses main branch.

```
91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (main)
$ git branch -M  main

91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (main)
$ git branch
* main
```

10. **git branch [branch_name]** ➔ This command is use to creates a new branch.

```
91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (main)
$ git branch dev1

91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (main)
$ git branch
  dev1
* main
```

11. **git branch -d [branch_name]** ➔ This command is use to deletes a branch.

```
91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (main)
$ git branch
  dev1
* main

91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (main)
$ git branch -d dev1
Deleted branch dev1 (was 828bf94).
```

12. **git checkout [branch_name]** ➔ This command is use to switch between branches.

```
91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (main)
$ git branch
  dev1
* main

91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (main)
$ git checkout dev1
Switched to branch 'dev1'

91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (dev1)
$ git branch
* dev1
  main
```

13. **git remote add [alias_name] [url]** ➔ This command is use to connect your local repository to the remote.

```
91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (dev1)
$ git remote add origin https://github.com/mohitmahi004/ineuronAssignment1.git

91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (dev1)
$ git remote
origin
```

14. **git push -u [alias_name] [your_branch_name]** ➔ This command is use to push your changes to GitHub repository.

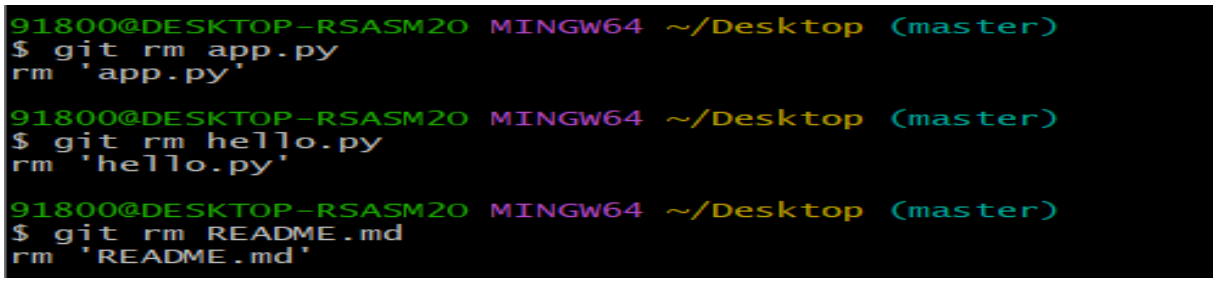
```
91800@DESKTOP-RSASM2O MINGW64 ~/Desktop/Assignment (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 379 bytes | 189.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/mohitmahi004/ineuronAssignment1.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

15. **git pull [remote_url]** ➔ This command is use to download the content but not the metadata and immediately updates your local repository with latest content.

```
91800@DESKTOP-RSASM2O MINGW64 ~/Desktop
$ git init
Initialized empty Git repository in C:/Users/91800/Desktop/.git/

91800@DESKTOP-RSASM2O MINGW64 ~/Desktop (master)
$ git pull https://github.com/mohitmahi004/ineuronAssignment1.git
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 359 bytes | 10.00 KiB/s, done.
From https://github.com/mohitmahi004/ineuronAssignment1
 * branch            HEAD       -> FETCH_HEAD
```

16. **git rm [your_file_name]** ➔ This command is use to delete files from your codebase.




**Note:- To add git credentials use following commands:**

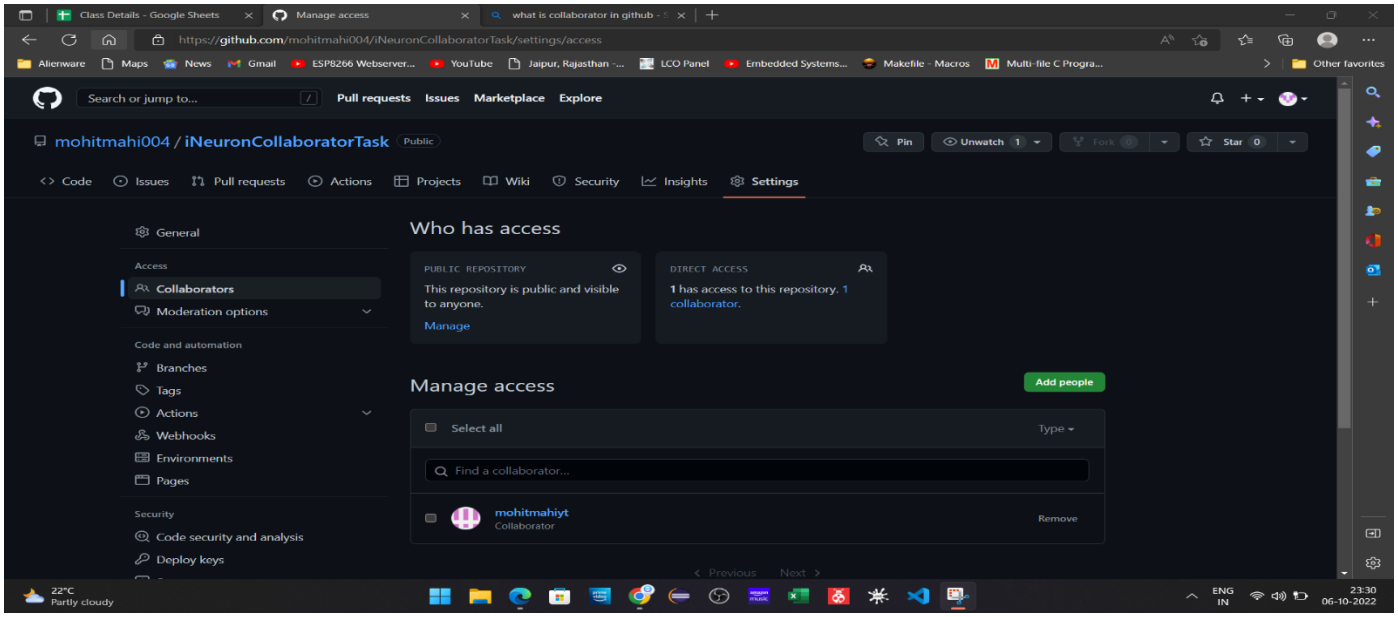
1. **git config --global user.name "FirstName LastName"**
2. **git config --global user.email "User_emailid"**

## Task 2

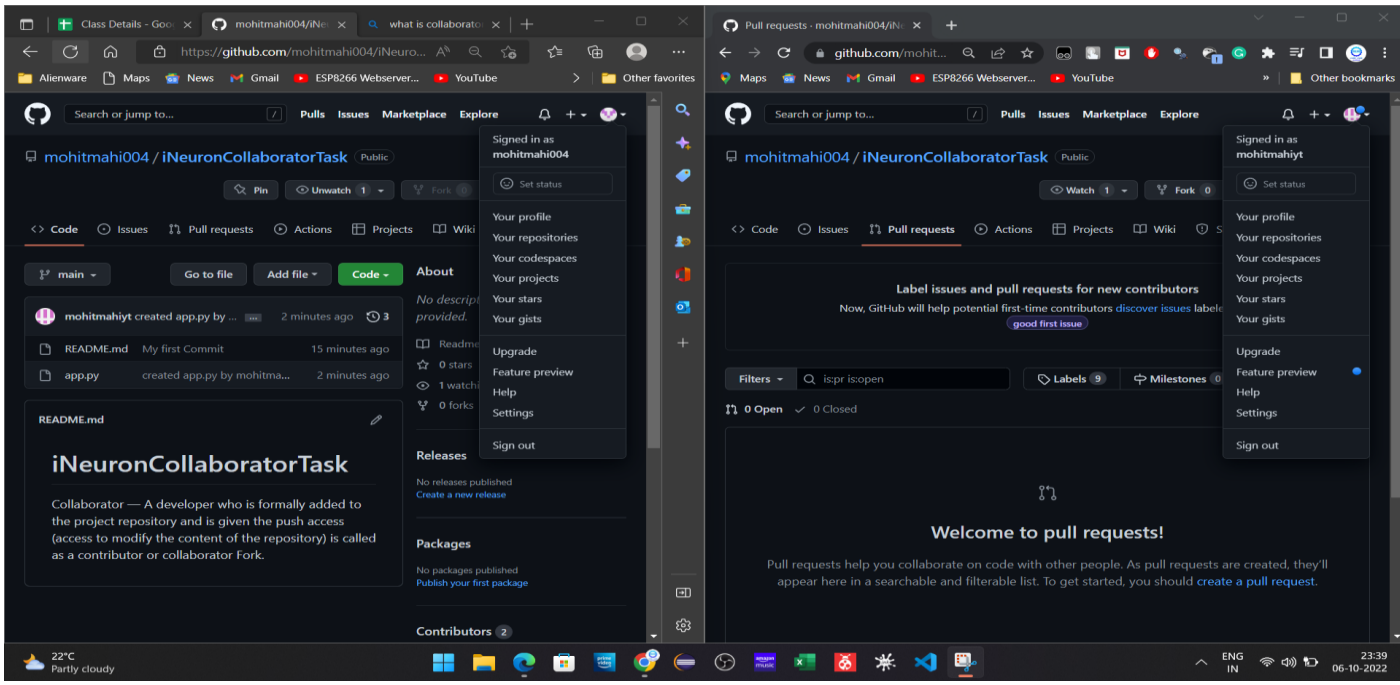
Consider that you want to start an open-source project in your organization. Perform all the standard operation to create a repository with minimal permission for all the users. It should contain.

1. Proper open source structure
2. Proper Readme
3. Add 2 collaborator
4. Host GitHub Pages using settings (Designed to host your personal, organization, or project pages from a GitHub repository)

**Ans.** **Link to the project-** mohitmahi004/iNeuronCollaboratorTask (github.com)




**The above screenshot shows collaborator access.**




**The above screenshot shows the collaborator with project owner.**