Task_05

Name : Manyu kumar

Reg : 2141011082

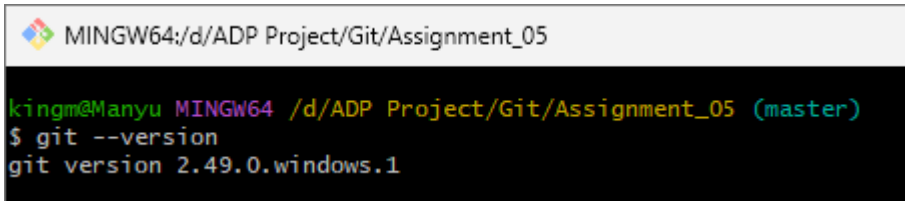
Git Important commands

$ - this sign means you are root user and you have all the permissions

(Master) or (main) indicate that you are currently on the root branch of the repository.

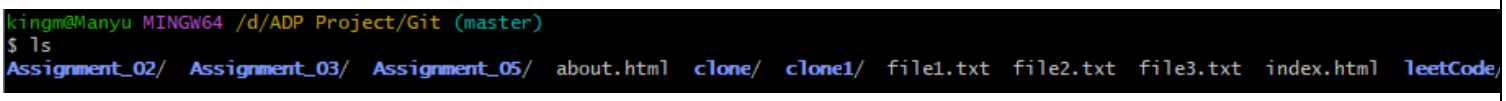
git --version

- It shows the which version is installed into your system



ls

- The ls command displays a list of files and directories in the current working directory.
-



git config --global user.name "Manyu Kumar"
git config --global user.email "manyu084@gmail.com"

- These settings ensure that all commits made from your system are associated with your name and email across all repositories.

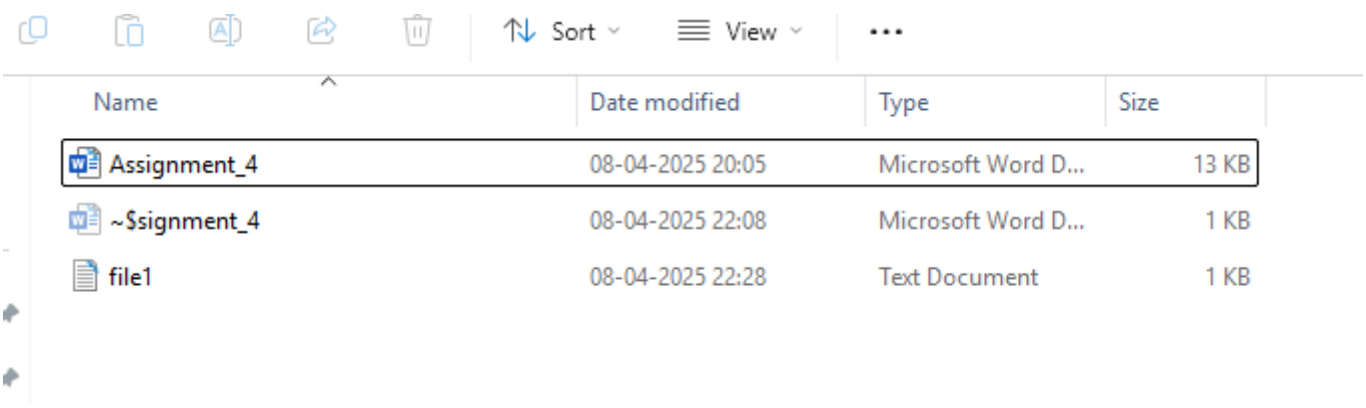
git config --global user.email
git config --global user.name

- The following commands display the global Git configuration settings for your email and username:

echo hello >> file.txt

- The command echo hello >> file.txt appends the text hello to the end of the file named file.txt.
- If file.txt does not already exist, it will be created automatically.

```
kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_05 (master)
$ echo "Hello from Manyu" >> file1.txt

kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_05 (master)
$
```



git add file1.txt

- The command git add file1.txt stages the file file1.txt for the next commit.
- This means any changes made to file1.txt will be included when you run git commit.

```
kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_05 (master)
$ git add file1.txt
warning: in the working copy of 'Assignment_05/file1.txt', LF will be replaced by CRLF the next time Git touches it

kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_05 (master)
$
```

Note: You can safely ignore any warnings that appear after adding files to the staging area using git add.

git status

- The command git status displays the current state of the working directory and the staging area.
- It shows which changes have been staged, which haven't, and which files aren't being tracked by Git.

Example is attached in next page :

```
kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_05 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file1.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   ../file1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ../Assignment_02/Assignment_02.docx
        ../Assignment_03/
        Assignment_4.docx
        ~$signment_4.docx
        ../about.html
        ../clone/
        ../clone1/
        ../leetCode/


kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_05 (master)
```

git commit -m “file updated”

- The command git commit -m "file updated" creates a new commit with the message *"file updated"*.
- The -m flag allows you to include a brief commit message directly in the command, summarizing the changes made.

```
kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_05 (master)
$ git commit -m "file1.txt updated"
[master 720e2c9] file1.txt updated
 1 file changed, 1 insertion(+)
 create mode 100644 Assignment_05/file1.txt
```

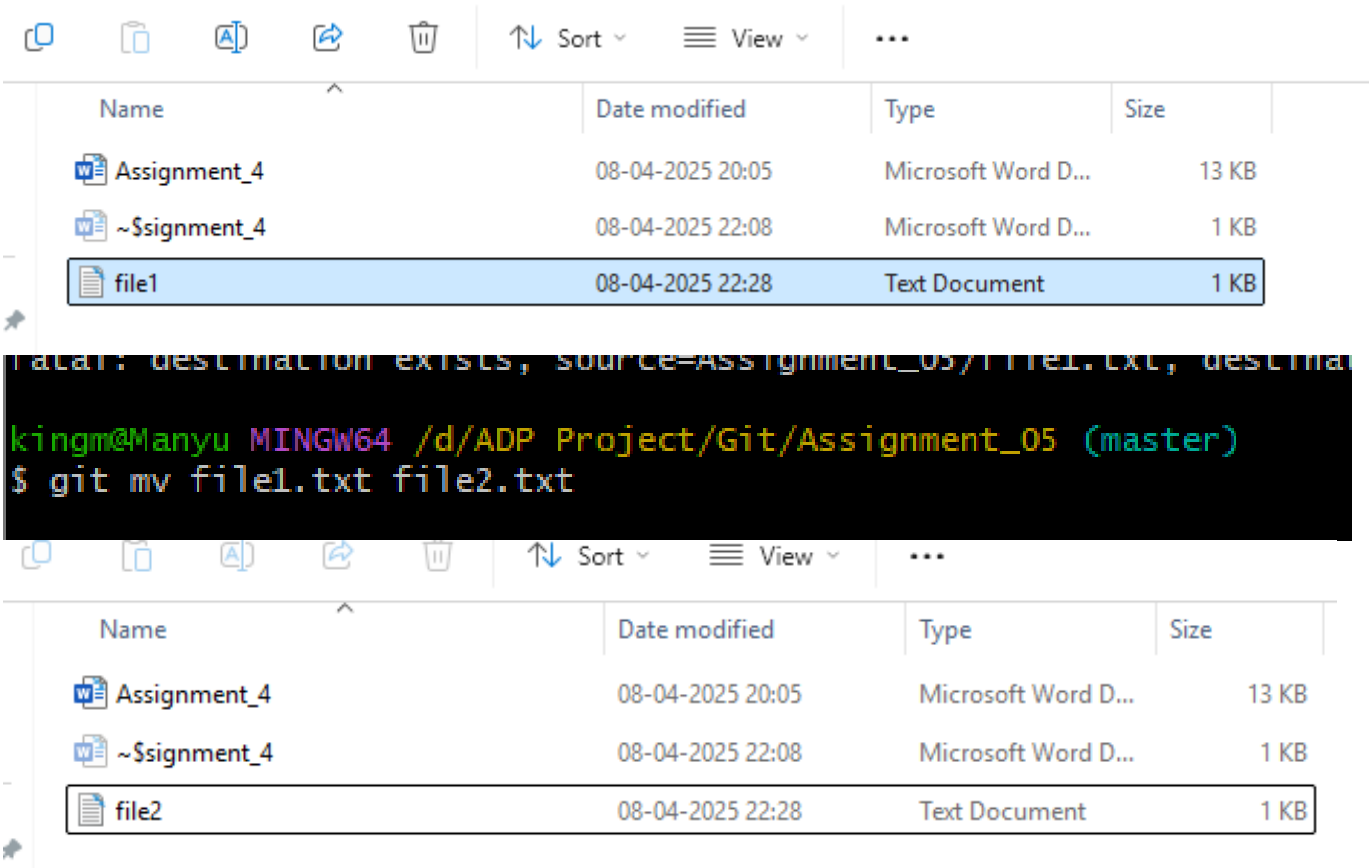
git checkout -f

- The command git checkout -f forces Git to discard all local changes in the working directory and switch to the currently checked-out branch.
- It restores all files to their last committed state, effectively overriding any uncommitted modifications.
- Note: This command will permanently delete any uncommitted changes. Use with caution.

git mv file1.txt file2.txt

- The command git mv file1.txt file2.txt renames or moves the file file1.txt to file2.txt and stages the change for the next commit.
- This is equivalent to running mv file1.txt file2.txt followed by git add file2.txt and git rm file1.txt, but in a single, streamlined command.

Example is attached in next page :



git ls-files

- The command git ls-files lists all the files that are currently being tracked by Git in the working directory.
- This includes files that have been staged, committed, or are part of the repository's version history—excluding untracked or ignored files by default.

git log

- The command git log displays the commit history of the current branch in reverse chronological order (most recent commit first).
- It shows details such as the commit hash, author, date, and commit message, allowing you to track changes over time.

```
kingm@Manyu MINGW64 /d/ADP Project/Git (master)
$ git log
commit 720e2c99f55d738f9a2912ec61f2f7be8fed2adb (HEAD -> master)
Author: Manyu Kumar <manyu084@gmail.com>
Date:   Tue Apr 8 23:42:10 2025 +0530

    file1.txt updated

commit bf7f1a19b333c6dd08707897adcabaa192ef8907
Merge: 1e350d1 99228e0
Author: 2141011082_Manyu Kumar <kingmanyu951@gmail.com>
Date:   Wed Apr 2 19:08:00 2025 +0530

    Trying resolve merge conflit

commit 1e350d14c8c852303d6a685966a5bac1769a2e00
Author: 2141011082_Manyu Kumar <kingmanyu951@gmail.com>
Date:   Wed Apr 2 19:03:58 2025 +0530

    file1.txt is add from master brach

commit 99228e0f76988555153f88e2f4d6cf6581e7005c (Assignment_02)
```

git branch {double tab}

- This will list all available branch names in your repository, helping you quickly view or switch between them without typing the full names manually.

```
    file updated

kingm@Manyu MINGW64 /d/ADP Project/Git (master)
$ git branch
Assignment_02   HEAD            ORIG_HEAD       checkout        master          wipro1
```

git branch “name of branch”

- The command git branch "name of branch" creates a new branch with the specified name.

```
kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_05 (master)
$ git branch Assignment_05

kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_05 (master)
$ git branch
Assignment_02   Assignment_05   HEAD            ORIG_HEAD       checkout        master          wipro1

kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_05 (master)
```

- We can see new branch Assignment_05 is created

git checkout {name of branch}

- The command git checkout {name of branch} switches your working directory to the specified branch.
-

```
kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_05 (master)
$ git checkout Assignment_05
D       Assignment_05/file1.txt
A       Assignment_05/file2.txt
Switched to branch 'Assignment_05'

kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_05 (Assignment_05)
$
```

- **Note:** We may observe that the branch name master has been replaced by Assignment_05.
- This indicates that you are currently working on the Assignment_05 branch instead of the default master branch.

git merge “name of current branch”

- The command git merge "name of current branch" merges the specified branch into your current branch.
- Important: We typically merge another branch into your current branch—not the current branch into itself.

```
Switched to branch 'Assignment_05'

kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_05 (Assignment_05)
$ git merge Assignment_05
Already up to date.

kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_05 (Assignment_05)
$
```

git branch -d “name of branch that we wanted to delete”

- The command git branch -d "branch-name" deletes the specified local branch, but only if it has been fully merged with your current branch or another base branch (like main or master).

```
kingm@Manyu MINGW64 /d/ADP Project/Git (Assignment_05)
$ git checkout master
D       Assignment_05/file1.txt
A       Assignment_05/file2.txt
Switched to branch 'master'

kingm@Manyu MINGW64 /d/ADP Project/Git (master)
$ git branch -d Assignment_05
Deleted branch Assignment_05 (was 720e2c9).

kingm@Manyu MINGW64 /d/ADP Project/Git (master)
$
```

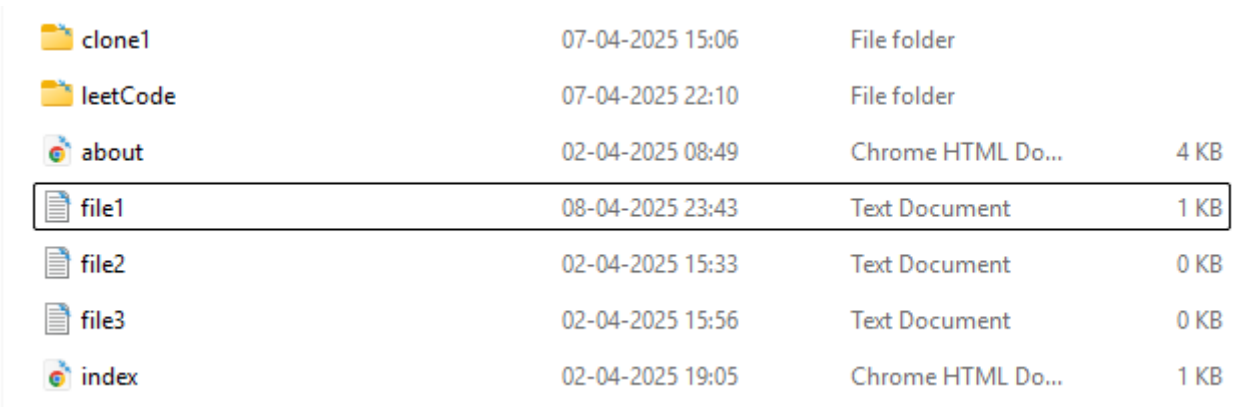
git clone “link of repository “

- The command git clone "repository-link" is used to create a local copy of a remote Git repository.
- It downloads all files, branches, and commit history from the specified repository.

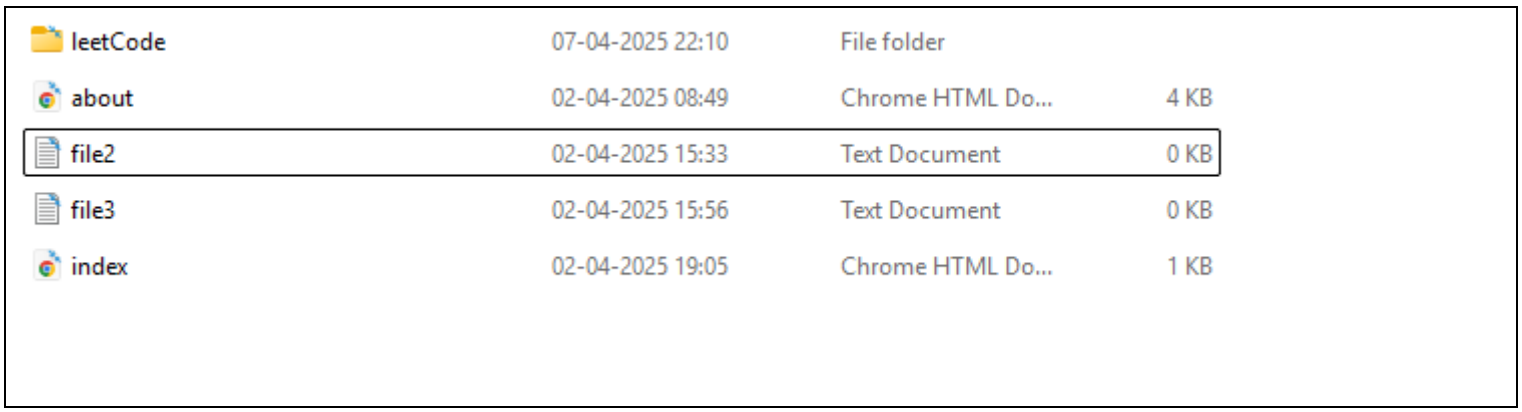
```
kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_03/leetCode (master)
$ git remote add upstream https://github.com/manyu951/leetCode.git

kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_03/leetCode (master)
$ git remote -v
origin  https://github.com/manyu951/leetCode.git (fetch)
origin  https://github.com/manyu951/leetCode.git (push)
upstream        https://github.com/manyu951/leetCode.git (fetch)
upstream        https://github.com/manyu951/leetCode.git (push)

kingm@Manyu MINGW64 /d/ADP Project/Git/Assignment_03/leetCode (master)
$
```

git rm file1.txt

- **The command git rm file1.txt removes the file file1.txt from both the working directory and the staging area.**




```
kingm@Manyu MINGW64 /d/ADP Project/Git (master)
$ git rm file1.txt
rm 'file1.txt'
```


leetCode 07-04-2025 22:10 File folder
about 02-04-2025 08:49 Chrome HTML Do... 4 KB
file2 02-04-2025 15:33 Text Document 0 KB
file3 02-04-2025 15:56 Text Document 0 KB
index 02-04-2025 19:05 Chrome HTML Do... 1 KB