GIT LAB ASSIGNMENT-DAY 2

NAME: NANDHINI K

REG NO: 731621104033

MAIL ID: subamohan993@gmail.com

EXERCISE – 01

1. Create a new directory and change into it.

2. Use the init command to create a Git repository in that directory.

3. Observe that there is now a .git directory.

4. Create a README file.

5. Look at the output of the status command; the README you created should appear as an untracked file.

```
D:\nandhini>git status
On branch master

No commits yet

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

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

D:\nandhini>git add README.md
```

6. Use the add command to add the new file to the staging area. Again, look at the output of the status command.

```
D:\nandhini>git add README.md

D:\nandhini>git status
On branch master

No commits yet

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

7. Now use the commit command to commit the contents of the staging area.

```
D:\nandhini>git commit -m "mycommit1"
[master (root-commit) 8eb8539] mycommit1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
```

8. Create a src directory and add a couple of files to it.

```
D:\nandhini>mkdir src

D:\nandhini>git add src
```

9. Use the add command, but name the directory, not the individual files. Use the status command. See how both files have been staged. Commit them.

```
D:\nandhini>git add src

D:\nandhini>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   src/file1
        new file:   src/file2


D:\nandhini>git commit -m "mycommit2"
[master 0ef046b] mycommit2
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/file1
 create mode 100644 src/file2
```

10. Make a change to one of the files. Use the diff command to view the details of the change.

file2	12-07-2024 15:45	File	0 KB
file3	12-07-2024 15:58	Text Document	0 KB
file111	12-07-2024 15:45	File	0 KB

11. Next, add the changed file, and notice how it moves to the staging area in the status output. Also observe that the diff command you did before using add now gives no output. Why not? What do you have to do to see a diff of the things in the staging area? (Hint: review the slides if you can't remember.)

```
D:\nandhini>git add file11
fatal: pathspec 'file11' did not match any files

D:\nandhini>git add src/file11

D:\nandhini>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   src/file11

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

12. Now – without committing – make another change to the same file you changed in step 10. Look at the status output, and the diff output. Notice how you can have both staged and unstaged changes, even when you're talking about a single file. Observe the difference when you use the add command to stage the latest round of changes. Finally, commit them. You should now have started to get a feel for the staging area.

file2	12-07-2024 15:45	File	0 KB	
file3	12-07-2024 15:58	Text Document	0 KB	
file111	12-07-2024 15:45	File	0 KB	

```
D:\nandhini>git diff
diff --git a/src/file1 b/src/file1
deleted file mode 100644
index e69de29..0000000
diff --git a/src/file11 b/src/file11
deleted file mode 100644
index e69de29..0000000

D:\nandhini>git add src/file111

D:\nandhini>git commit -m "mycommit3"
[master 073b6e3] mycommit3
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/file11
 create mode 100644 src/file111
```

13. Use the log command in order to see all of the commits you made so far.

```
D:\nandhini>git log
commit 073b6e32573d0315299e1162c557250f4b864f05 (HEAD -> master)
Author: Priyadharshini Narayanan <priyanarayanan2603@gmail.com>
Date:   Fri Jul 12 15:55:12 2024 +0530

    mycommit3

commit 0ef046bad34300fd1942e3fbe9df41df467b2a68
Author: Priyadharshini Narayanan <priyanarayanan2603@gmail.com>
Date:   Fri Jul 12 15:46:38 2024 +0530

    mycommit2

commit 8eb85397b7bc861fd4767a9cfd58330954a7427c
Author: Priyadharshini Narayanan <priyanarayanan2603@gmail.com>
Date:   Fri Jul 12 15:44:55 2024 +0530

    mycommit1
```

14. Use the show command to look at an individual commit. How many characters of the commit identifier can you get away with typing at a minimum?

```
D:\nandhini>git show
commit 073b6e32573d0315299e1162c557250f4b864f05 (HEAD -> master)
Author: Priyadharshini Narayanan <priyanarayanan2603@gmail.com>
Date:   Fri Jul 12 15:55:12 2024 +0530

    mycommit3

diff --git a/src/file11 b/src/file11
new file mode 100644
index 0000000..e69de29
diff --git a/src/file111 b/src/file111
new file mode 100644
index 0000000..e69de29
```

15. Make a couple more commits, at least one of which should add an extra file.

```
D:\nandhini>git commit -m "mycommit4"
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    src/file1
        deleted:    src/file11

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        src/file3.txt

no changes added to commit (use "git add" and/or "git commit -a")

D:\nandhini>git add src/file3
fatal: pathspec 'src/file3' did not match any files

D:\nandhini>git add src/file3.txt

D:\nandhini>git commit -m "mycommit5"
[master 93eb6dd] mycommit5
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/file3.txt
```