NAME: HARIHARAN T

REG NO: 73772114140

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING

COLLEGE: K S RANGASAMY COLLEGE OF TECHNOLOGY

ASSIGNMENT(GIT)

EXERCISE 1

Main Task

1. Create a new directory and change into it.

```
C:\Users\User>d:

D:\>cd Localrepo

D:\Localrepo>git init
Initialized empty Git repository in D:/Localrepo/.git/
```
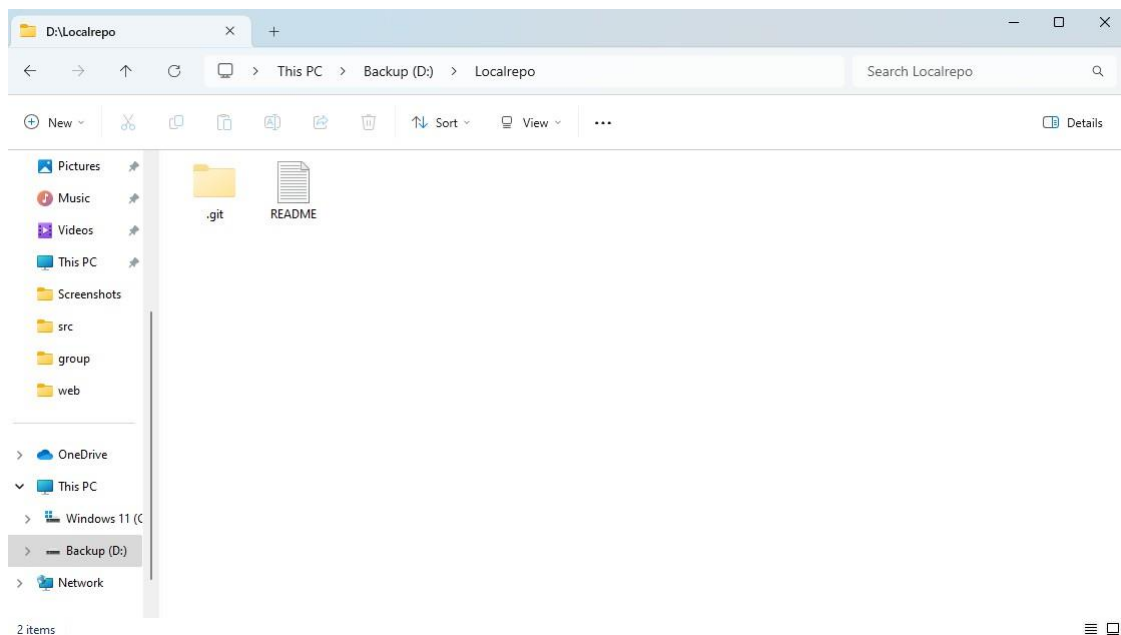
2. Use the init command to create a Git repository in that directory.

```
C:\Users\User>d:

D:\>cd Localrepo

D:\Localrepo>git init
Initialized empty Git repository in D:/Localrepo/.git/
```

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

```
D:\>cd src

D:\src>git init
Initialized empty Git repository in D:/src/.git/
```

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:\Localrepo>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)
```

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


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

D:\Localrepo>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md
```
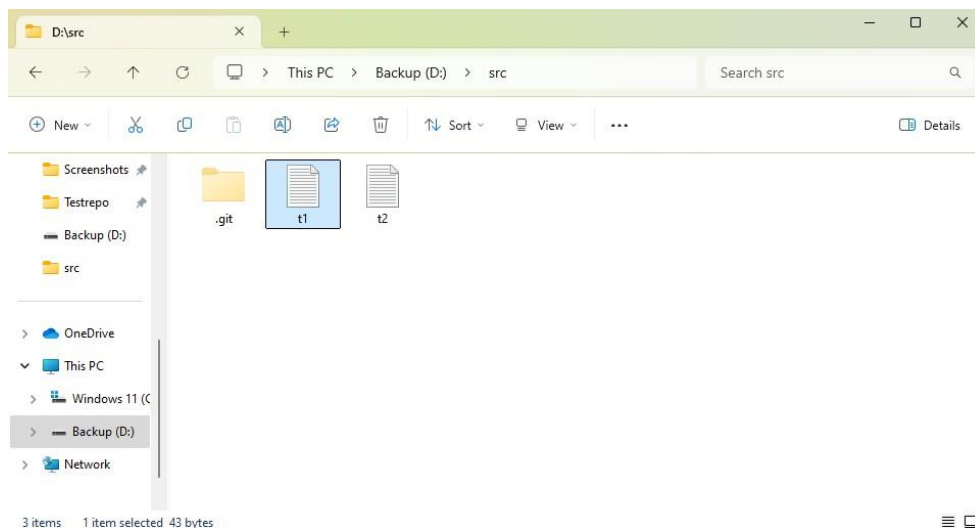
7. Now use the commit command to commit the contents of the staging area.

```
D:\Localrepo>git commit -m "README COMMITED"
[master (root-commit) 7f00e3a] README COMMITED
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md

D:\Localrepo>git status
On branch master
nothing to commit, working tree clean
```

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

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:\src>git add .

D:\src>git commit -m "Files commited"
[master (root-commit) 50f4d58] Files commited
 4 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f1.c
 create mode 100644 j1.java
 create mode 100644 t1.txt
 create mode 100644 t2.txt
```

```
D:\src>git commit -m "Files commited"
[master (root-commit) 50f4d58] Files commited
 4 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f1.c
 create mode 100644 j1.java
 create mode 100644 t1.txt
 create mode 100644 t2.txt

D:\src>git status
On branch master
nothing to commit, working tree clean
```

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

```
D:\src>git status
On branch master
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:   t2.txt

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

```
D:\src>git status
On branch master
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:   t2.txt

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

D:\src>git diff
diff --git a/t2.txt b/t2.txt
index 374db27..32c4b26 100644
--- a/t2.txt
+++ b/t2.txt
@@ -1 +1,2 @@
-hi how are you
\ No newline at end of file
+hi how are you
+i am kirthivashan
\ No newline at end of file

D:\src>git add .

D:\src>git commit -m "updated2"
[master b0f2f40] updated2
 1 file changed, 2 insertions(+), 1 deletion(-)
```

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:\src>git diff
diff --git a/t2.txt b/t2.txt
index e69de29..374db27 100644
--- a/t2.txt
+++ b/t2.txt
@@ -0,0 +1 @@
+hi how are you
\ No newline at end of file
```

```
D:\src>git add t2.txt

D:\src>git commit -m "updated"
[master 58b9ce8] updated
 1 file changed, 1 insertion(+)

D:\src>git status
On branch master
nothing to commit, working tree clean
```

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.

```
D:\src>git add t2.txt

D:\src>git commit -m "updated"
[master 58b9ce8] updated
 1 file changed, 1 insertion(+)

D:\src>git status
On branch master
nothing to commit, working tree clean
```

```
D:\src>git diff
diff --git a/t2.txt b/t2.txt
index e69de29..374db27 100644
--- a/t2.txt
+++ b/t2.txt
@@ -0,0 +1 @@
+hi how are you
\ No newline at end of file
```

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

```
D:\src>git log
commit b0f2f408377232cdd3dd4d2262c023eb1e6935ff (HEAD -> master)
Author: kirthivashan <vkvashan24@gmail.com>
Date:   Mon Jul 15 09:39:19 2024 +0530

    updated2

commit 58b9ce8692cfada1c789dd791a28bbfc8b918ed3
Author: kirthivashan <vkvashan24@gmail.com>
Date:   Mon Jul 15 09:35:25 2024 +0530

    updated

commit 98e1f27be696bb0ace4d89eba8eb0b6215eae875
Author: kirthivashan <vkvashan24@gmail.com>
Date:   Mon Jul 15 09:28:36 2024 +0530

    changed and updated

commit 50f4d58dca1ccbaa0537fe89437860ae2c43aba2
Author: kirthivashan <vkvashan24@gmail.com>
Date:   Mon Jul 15 09:24:13 2024 +0530

    Files commited
```

```
D:\src>git log --stat -M
commit b9d68d17b1a067e766f7a7c6c9cd5fe40d312590 (HEAD -> master)
Author: kirthivashan <vkvashan24@gmail.com>
Date:   Mon Jul 15 09:46:12 2024 +0530

    deleted

 f1.c    | 1 -
 j1.java | 0
 2 files changed, 1 deletion(-)

commit 01082425dd87d5b6d20d72f63ff108379a51c500
Author: kirthivashan <vkvashan24@gmail.com>
Date:   Mon Jul 15 09:43:32 2024 +0530

    c commited

 f1.c | 1 +
 1 file changed, 1 insertion(+)

commit b0f2f408377232cdd3dd4d2262c023eb1e6935ff
Author: kirthivashan <vkvashan24@gmail.com>
Date:   Mon Jul 15 09:39:19 2024 +0530

    updated2

 t2.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)

commit 58b9ce8692cfada1c789dd791a28bbfc8b918ed3
Author: kirthivashan <vkvashan24@gmail.com>
Date:   Mon Jul 15 09:35:25 2024 +0530

    updated

 t2.txt | 1 +
 1 file changed, 1 insertion(+)

commit 98e1f27be696bb0ace4d89eba8eb0b6215eae875
Author: kirthivashan <vkvashan24@gmail.com>
Date:   Mon Jul 15 09:28:36 2024 +0530

    changed and updated
```

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:\src>git show b0f2f408
commit b0f2f408377232cdd3dd4d2262c023eb1e6935ff
Author: kirthivashan <vkvashan24@gmail.com>
Date:   Mon Jul 15 09:39:19 2024 +0530

    updated2

diff --git a/t2.txt b/t2.txt
index 374db27..32c4b26 100644
--- a/t2.txt
+++ b/t2.txt
@@ -1 +1,2 @@
-hi how are you
\ No newline at end of file
+hi how are you
+i am kirthivashan
\ No newline at end of file

D:\src>git show b0f2
commit b0f2f408377232cdd3dd4d2262c023eb1e6935ff
Author: kirthivashan <vkvashan24@gmail.com>
Date:   Mon Jul 15 09:39:19 2024 +0530

    updated2

diff --git a/t2.txt b/t2.txt
index 374db27..32c4b26 100644
--- a/t2.txt
+++ b/t2.txt
@@ -1 +1,2 @@
-hi how are you
\ No newline at end of file
```

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

```
D:\Localrepo>git mv README.md README.txt

D:\Localrepo>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    README.md -> README.txt


D:\Localrepo>git commit -m "renamed"
[master b6618de] renamed
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename README.md => README.txt (100%)
```