

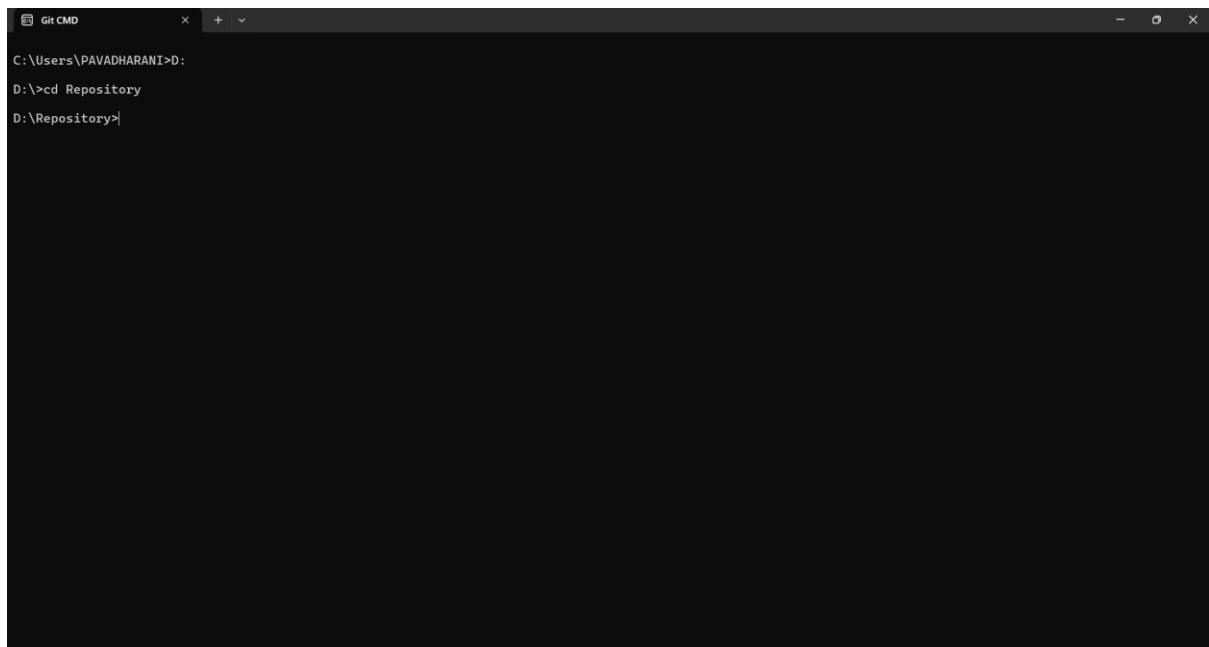
SOURCE REPOSITORY ASSIGNMENT-2

NAME : PAVADHARANI K N
BRANCH : COMPUTER SCIENCE AND ENGINEERINNG
REG NO : 73772114176
EMAIL : pavadharaninargunaraj474@gmail.com

EXERCISE-1

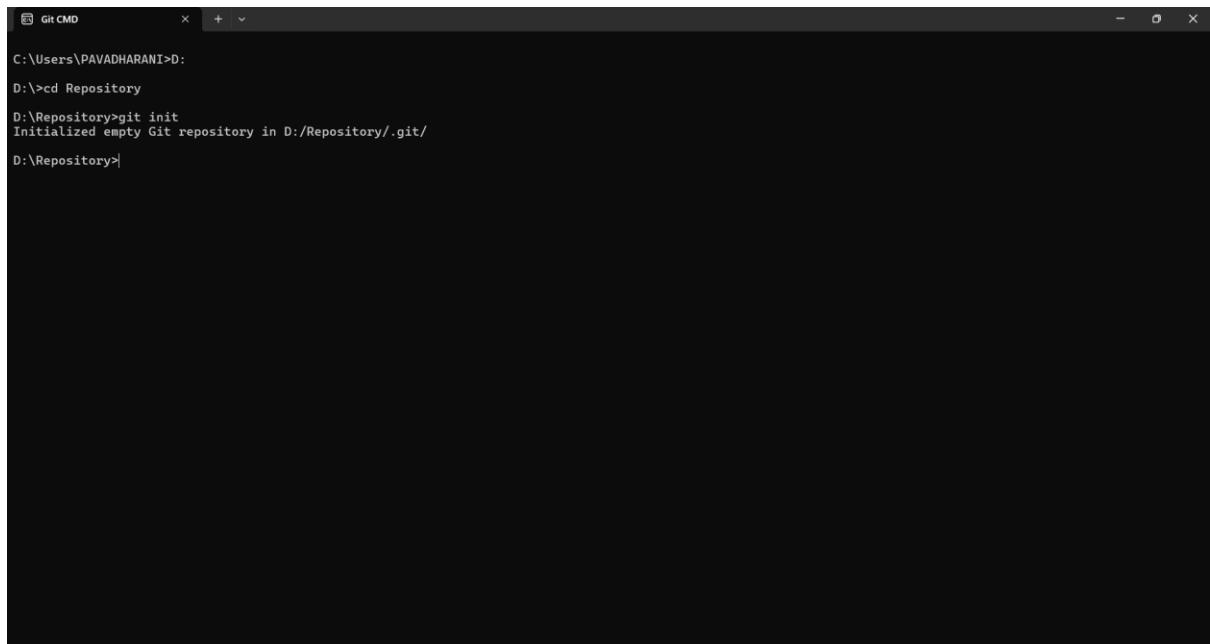
Main Task

- 1.Create a new directory and change into it.



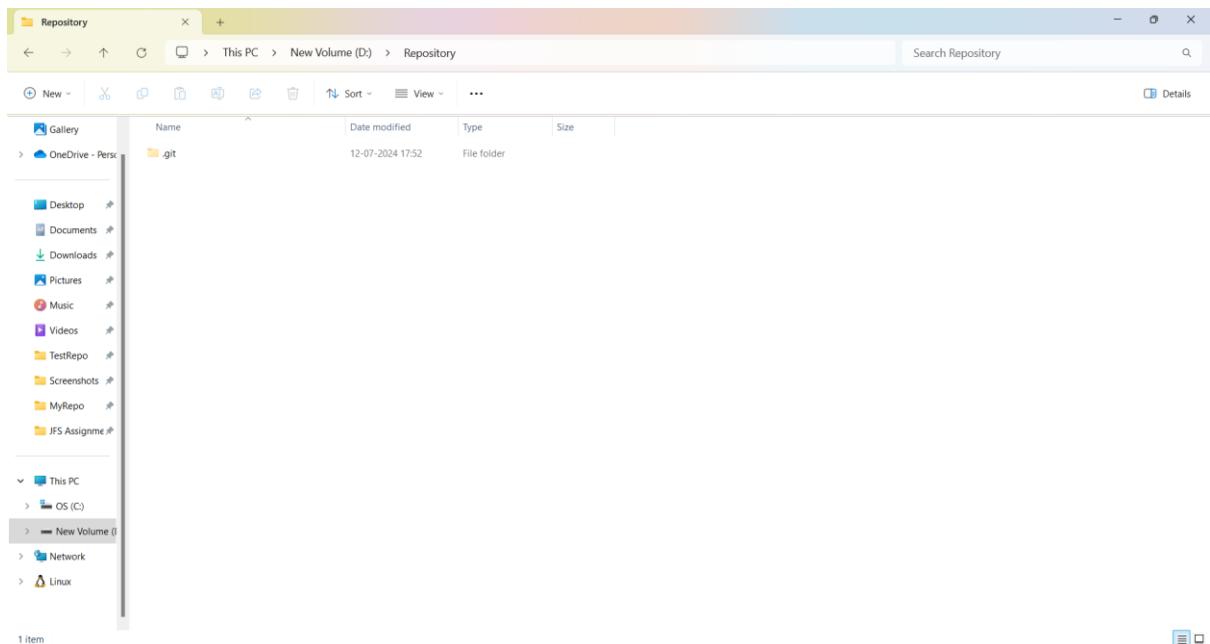
```
Git CMD
C:\Users\PAVADHARANI>D:
D:\>cd Repository
D:\Repository>
```

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

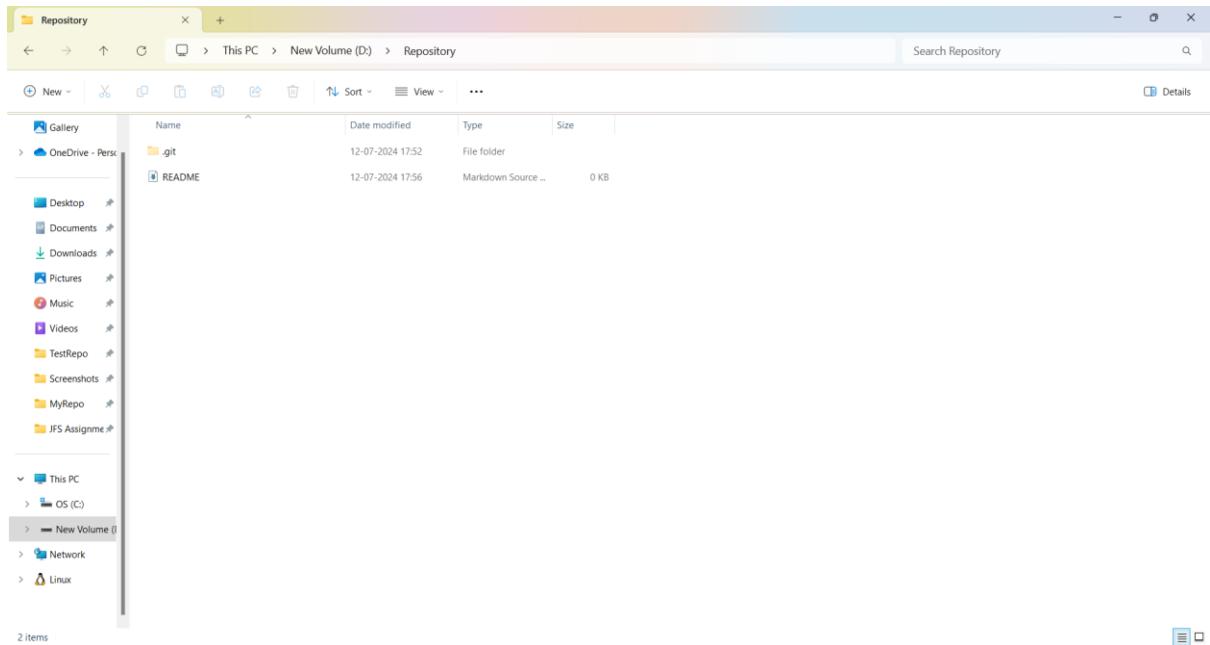


```
C:\Users\PAVADHARANI>D:  
D:\>cd Repository  
D:\Repository>git init  
Initialized empty Git repository in D:/Repository/.git/  
D:\Repository>
```

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



4.Create a README file.

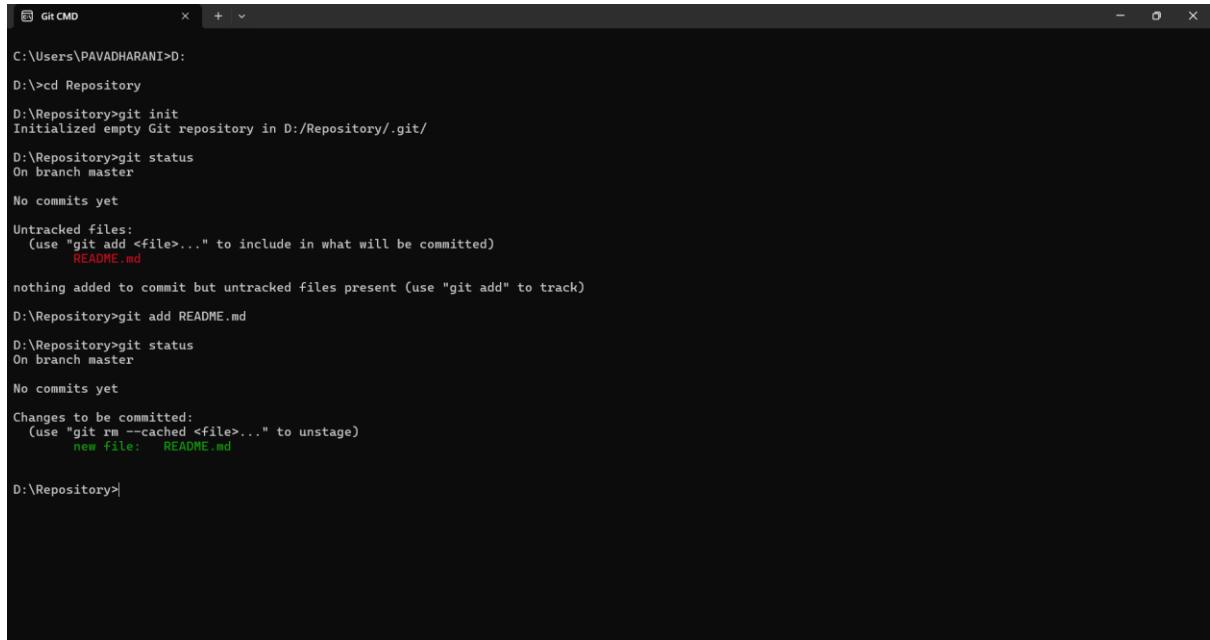


5. Look at the output of the **status** command; the **README** you created should appear as an untracked file.

```
C:\Users\PAVADHARANI>D:  
D:\>cd Repository  
D:\Repository>git init  
Initialized empty Git repository in D:/Repository/.git/  
D:\Repository>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:\Repository>
```

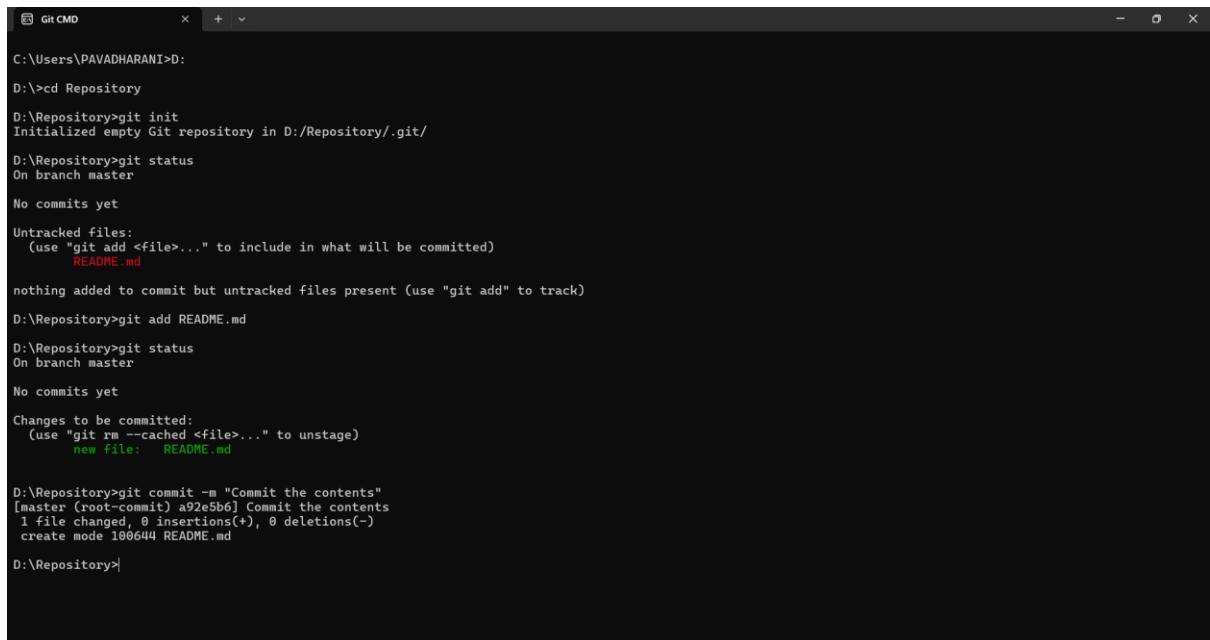
A screenshot of a Git Command Prompt window titled "Git CMD". The command history shows the user navigating to the "Repository" directory on drive D, initializing a new Git repository, and then checking the status. The output indicates that there are no commits yet, but an untracked file named "README.md" is present. The prompt ends with "D:\Repository>".

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



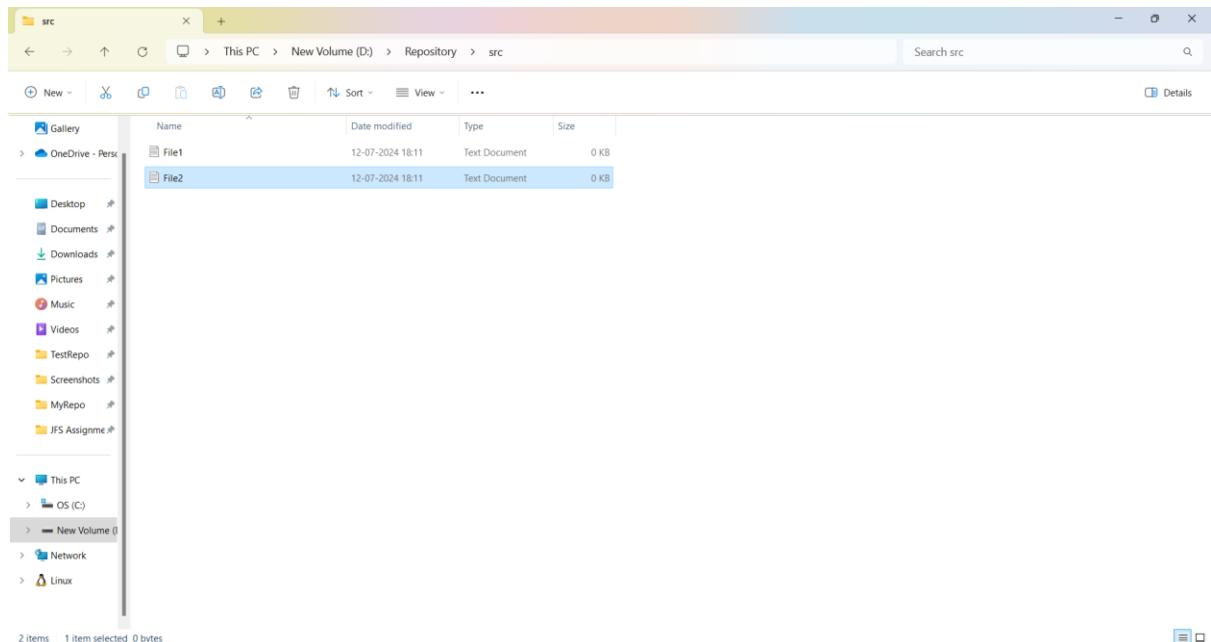
```
C:\Users\PAVADHARANI>D:  
D:\>cd Repository  
D:\Repository>git init  
Initialized empty Git repository in D:/Repository/.git/  
D:\Repository>git status  
On branch master  
No commits yet  
Untracked files:  
(use "git add <file>..." to include in what will be committed)  
 README.md  
nothing added to commit but untracked files present (use "git add" to track)  
D:\Repository>git add README.md  
D:\Repository>git status  
On branch master  
No commits yet  
Changes to be committed:  
(use "git rm --cached <file>..." to unstage)  
 new file: README.md  
D:\Repository>
```

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



```
C:\Users\PAVADHARANI>D:  
D:\>cd Repository  
D:\Repository>git init  
Initialized empty Git repository in D:/Repository/.git/  
D:\Repository>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:\Repository>git add README.md  
D:\Repository>git status  
On branch master  
No commits yet  
Changes to be committed:  
(use "git rm --cached <file>..." to unstage)  
 new file: README.md  
D:\Repository>git commit -m "Commit the contents"  
[master (root-commit) a92e5b6] Commit the contents  
 1 file changed, 0 insertions(+), 0 deletions(-)  
 create mode 100644 README.md  
D:\Repository>
```

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.

```
Git CMD
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:\Repository>git add README.md

D:\Repository>git status
On branch master

No commits yet

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

D:\Repository>git commit -m "Commit the contents"
[master (root-commit) a92e5b6] Commit the contents
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md

D:\Repository>git add src

D:\Repository>git status
On branch master
Changes to be committed:
(use "git restore --staged <file>..." to unstage)
 new file: src/File1.txt
 new file: src/File2.txt

D:\Repository>git commit -m "Commit the src director"
[master 732af82] Commit the src director
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/File1.txt
 create mode 100644 src/File2.txt

D:\Repository>
```

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

```
Git CMD
D:\Repository>git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file: README.md

D:\Repository>git commit -m "Commit the contents"
[master a92e5b6] Commit the contents
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md

D:\Repository>git add src
D:\Repository>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file: src/File1.txt
    new file: src/File2.txt

D:\Repository>git commit -m "Commit the src director"
[master 732aef2] Commit the src director
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/File1.txt
 create mode 100644 src/File2.txt

D:\Repository>git diff
diff --git a/src/File1.txt b/src/File1.txt
index e69de29..7a595e0 100644
--- a/src/File1.txt
+++ b/src/File1.txt
@@ -0,0 +1 @@
+It is a Git Exercise -1
\ No newline at end of file
D:\Repository>
```

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.)

```
Git CMD
D:\Repository>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified: src/File1.txt

D:\Repository>git commit -m "Commit the changed file"
[master ea2abdc] Commit the changed file
 1 file changed, 1 insertion(+)

D:\Repository>git log
commit ea2abdcf0a74c4187e30a5ec15b39cdf6808b6f5 (HEAD -> master)
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:22:31 2024 +0530

  Commit the changed file

commit 732af825414cbf0afe59cd88652c5ceddd4d5ed8
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:13:23 2024 +0530

  Commit the src director

commit a92e5b634c9a12b9eb7f3c634580ed534b9af8
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:02:16 2024 +0530

  Commit the contents

D:\Repository>git diff 732af825414cbf0afe59cd88652c5ceddd4d5ed8 ea2abdcf0a74c4187e30a5ec15b39cdf6808b6f5
diff --git a/src/File1.txt b/src/File1.txt
index e69de29..7a595e0 100644
--- a/src/File1.txt
+++ b/src/File1.txt
@@ -0,0 +1 @@
+It is a Git Exercise -1
\ No newline at end of file
D:\Repository>
```

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.

```
Git CMD x + v
index e69de29..7a595e0 100644
--- a/src/File1.txt
+++ b/src/File1.txt
@@ -0,0 +1 @@
+It is a Git Exercise -1
\ No newline at end of file

D:\Repository>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:   src/File1.txt

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

D:\Repository>git diff
diff --git a/src/File1.txt b/src/File1.txt
index 7a595e0..ea73a49 100644
--- a/src/File1.txt
+++ b/src/File1.txt
@@ -1,5 @@
-It is a Git Exercise -1
\ No newline at end of file
+It is a Git Exercise -1
+
+Done By Pavadharani K M
+
+From Computer science and engineering
\ No newline at end of file

D:\Repository>git commit -m"Commit the second Changes"
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:   src/File1.txt

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

D:\Repository>
```

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

```
Git CMD x + v
--- a/src/File1.txt
+++ b/src/File1.txt
@@ -1,5 @@
-It is a Git Exercise -1
\ No newline at end of file
+It is a Git Exercise -1
+
+Done By Pavadharani K M
+
+From Computer science and engineering
\ No newline at end of file

D:\Repository>git commit -m"Commit the second Changes"
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:   src/File1.txt

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

D:\Repository>git log
commit ea2abdcf0a74c4187e30a5ec15b39cdf6808b6f5 (HEAD -> master)
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:22:31 2024 +0530

  Commit the changed file

commit 732af825414cbf0afe59cd88652c5ceddd4d5ed
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:13:23 2024 +0530

  Commit the src director

commit a92e5b634c9a12b9eb7f3c634500ed534b9a2f82
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:02:16 2024 +0530

  Commit the contents

D:\Repository>
```

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?

```
git CMD x + v
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working directory)
 modified: src/File1.txt

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

D:\Repository>git log
commit ea2abdcf0a74c4187e30a5ec15b39cdf6808b6f5 (HEAD -> master)
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:22:31 2024 +0530

    Commit the changed file

commit 732af825414cbf0afe59cd88652c5cebdd4d5ed
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:13:23 2024 +0530

    Commit the src director

commit a92e5b634c9a12b9eb7f3c634500ed534b9a2f82
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:02:16 2024 +0530

    Commit the contents

D:\Repository>git show
commit ea2abdcf0a74c4187e30a5ec15b39cdf6808b6f5 (HEAD -> master)
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:22:31 2024 +0530

    Commit the changed file

diff --git a/src/File1.txt b/src/File1.txt
index e69de29..7a595e0 100644
--- a/src/File1.txt
+++ b/src/File1.txt
@@ -0,0 +1 @@
+It is a Git Exercise -1
\ No newline at end of file

D:\Repository>
```

It is not necessary to use all the character in commit identifier. The first 7 or 8 characters are sufficient to uniquely identify the commit.

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

```
git CMD x + v
D:\Repository>git show
commit ea2abdcf0a74c4187e30a5ec15b39cdf6808b6f5 (HEAD -> master)
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:22:31 2024 +0530

    Commit the changed file

diff --git a/src/File1.txt b/src/File1.txt
index e69de29..7a595e0 100644
--- a/src/File1.txt
+++ b/src/File1.txt
@@ -0,0 +1 @@
+It is a Git Exercise -1
\ No newline at end of file

D:\Repository>git commit -m"commit the adding changes"
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: src/File1.txt

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

D:\Repository>git add .

D:\Repository>git commit -m"commit the last changes"
[master 3c20855] commit the last changes
 1 file changed, 5 insertions(+), 1 deletion(-)

D:\Repository>git add .

D:\Repository>git commit -m"commit the last changes"
[master 6faa5b9] commit the last changes
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/File3.txt

D:\Repository>
```

Stretch Task

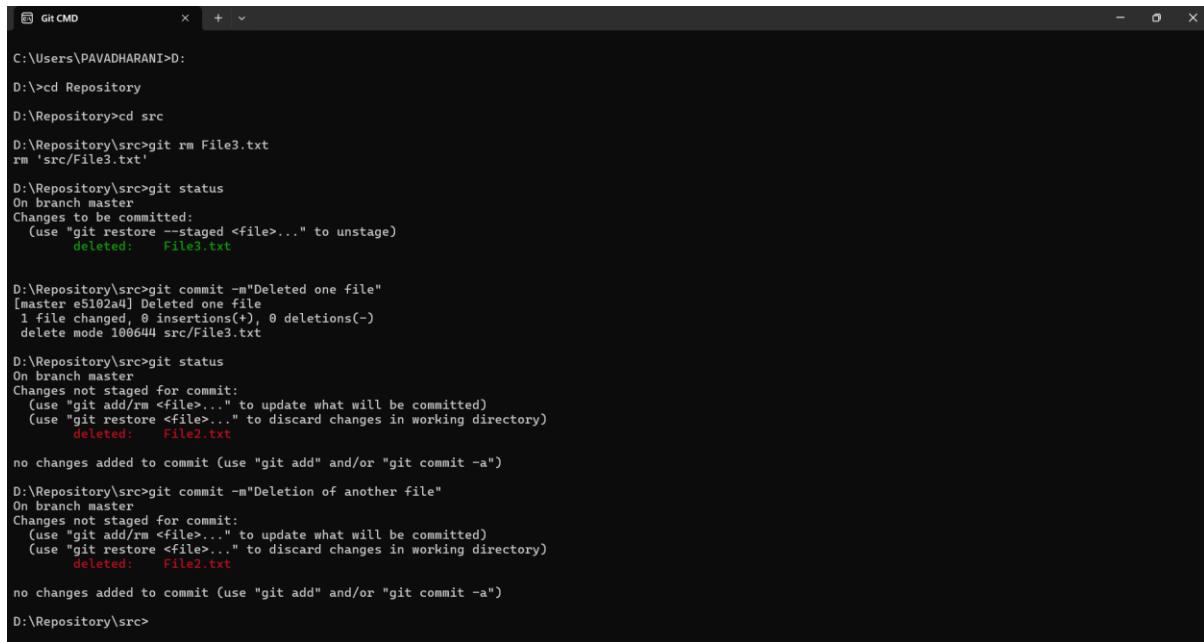
1. Use the Git **rm** command to remove a file. Look at the **status** afterwards. Now commit the deletion.

```
C:\Users\PAVADHARANI>D:  
D:\>cd Repository  
D:\Repository>cd src  
D:\Repository\src>git rm File3.txt  
rm 'src/File3.txt'  
D:\Repository\src>git status  
On branch master  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    deleted:   File3.txt  
D:\Repository\src>git commit -m"Deleted one file"  
[master e5102ad] Deleted one file  
 1 file changed, 0 insertions(+), 0 deletions(-)  
 delete mode 100644 src/File3.txt  
D:\Repository\src>
```

2. Delete another file, but this time do not use Git to do it; e.g. if you are on Linux, just use the normal (non-Git) **rm** command; on Windows use **del**.

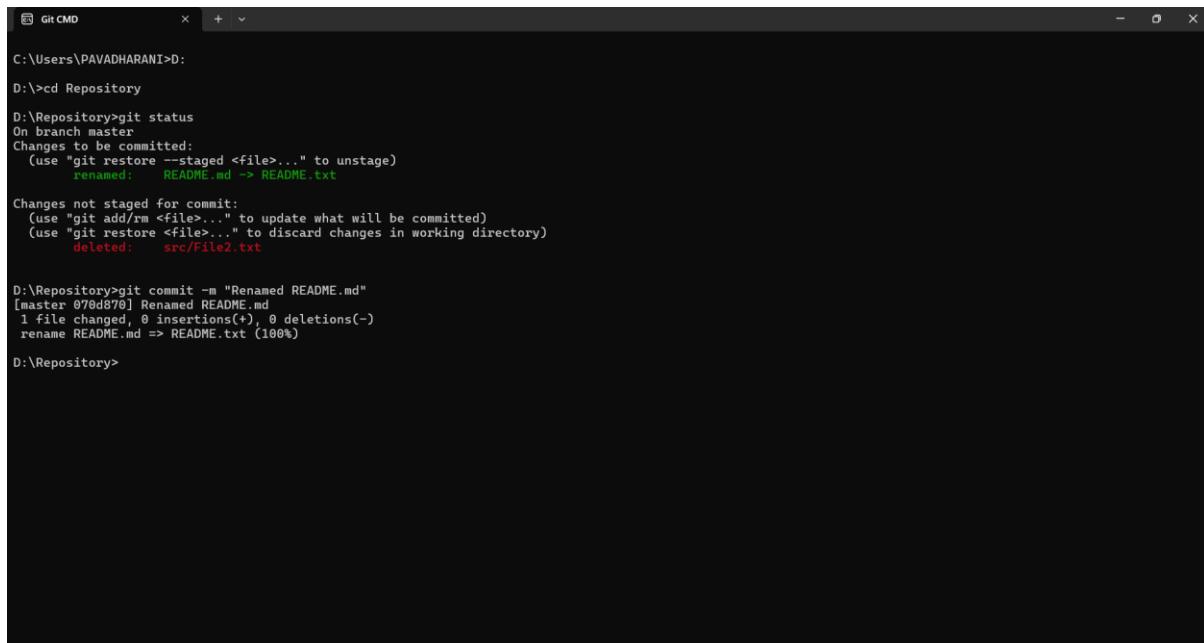
```
Microsoft Windows [Version 10.0.22631.3447]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users\PAVADHARANI>cd Repository  
The system cannot find the path specified.  
C:\Users\PAVADHARANI>cd src  
The system cannot find the path specified.  
C:\Users\PAVADHARANI>D:  
D:\>cd Repository  
D:\Repository>cd src  
D:\Repository\src>del File2.txt  
D:\Repository\src>
```

3. Look at the **status**. Compare it to the status output you had after using the Git built-in **rm** command. Is anything different? After this, commit the deletion.



```
C:\Users\PAVADHARANI>D:  
D:>cd Repository  
D:\Repository>cd src  
D:\Repository\src>git rm File3.txt  
rm 'src/File3.txt'  
D:\Repository\src>git status  
On branch master  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    deleted:   File3.txt  
  
D:\Repository\src>git commit -m"Deleted one file"  
[master e5102a4] Deleted one file  
 1 file changed, 0 insertions(+), 0 deletions(-)  
 delete mode 100644 src/File3.txt  
  
D:\Repository\src>git status  
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:   File2.txt  
  
no changes added to commit (use "git add" and/or "git commit -a")  
D:\Repository\src>git commit -m"Deletion of another file"  
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:   File2.txt  
  
no changes added to commit (use "git add" and/or "git commit -a")  
D:\Repository\src>
```

4. Use the Git **mv** command to move or rename a file; for example, rename **README** to **README.txt**. Look at the status. Commit the change.



```
C:\Users\PAVADHARANI>D:  
D:>cd Repository  
D:\Repository>git status  
On branch master  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    renamed:   README.md => README.txt  
  
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/File2.txt  
  
D:\Repository>git commit -m "Renamed README.md"  
[master 078d870] Renamed README.md  
 1 file changed, 0 insertions(+), 0 deletions(-)  
 rename README.md => README.txt (100%)  
D:\Repository>
```

5. Now do another rename, but this time using the operating system's command to do so. How does the status look? Will you get the right outcome if you were to **commit** at this point? (Answer: almost certainly not, so don't. ☐) Work out how to get the **status** to show that it will not lose the file, and then commit. Did Git at any point work out that you had done a rename?

```
D:\Repository>git status
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: README.txt
      deleted: src/File2.txt

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

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

D:\Repository>git commit -m"Renamed"
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: README.txt
    deleted: src/File2.txt

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

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

D:\Repository>
```

6. Use **git help log** to find out how to get Git to display just the most recent 3 commits. Try it.

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

D:\Repository>git commit -m"Renamed"
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: README.txt
    deleted: src/File2.txt

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

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

D:\Repository>git help log
D:\Repository>git log -n 3
commit 070d876687b27b6adc36381960b5dccf8b0d200d (HEAD -> master)
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Mon Jul 15 09:32:50 2024 +0530

  Renamed README.md

commit e5102a45f1a929d977ed47715bba39dcade25ec4
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Mon Jul 15 09:14:35 2024 +0530

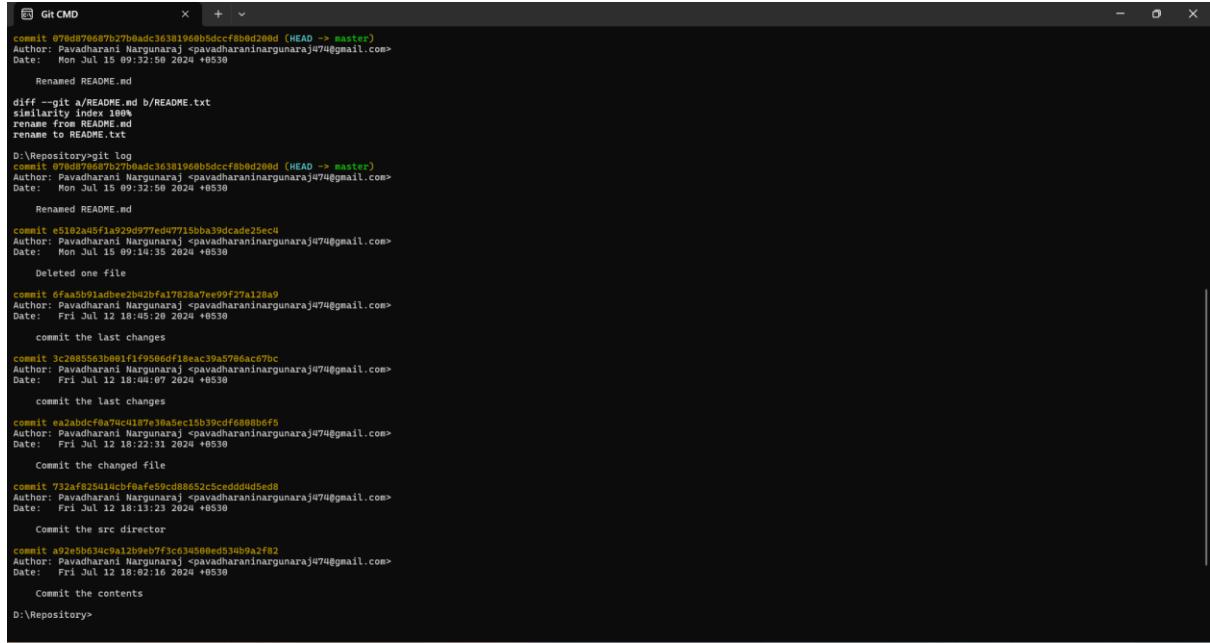
  Deleted one file

commit 6faa5b91adbbee2b42bfa17828a7ee99f27a128a9
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:45:20 2024 +0530

  commit the last changes

D:\Repository>
```

7. If you don't remember, look back in the slides to see what the **--stat** option did on the **diff** command. Find out if this also works with the **show** command. How about the **log** command?



```
Git CMD
commit 070d87067b27b0adc36381960b5dccf8b0d200d (HEAD -> master)
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Mon Jul 15 09:32:58 2024 +0530

Renamed README.md

diff --git a/README.md b/README.txt
similarity index 100%
rename from README.md
rename to README.txt

D:\Repository>git log
commit 070d87067b27b0adc36381960b5dccf8b0d200d (HEAD -> master)
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Mon Jul 15 09:32:58 2024 +0530

Renamed README.md

commit e5102a45f1a929d977ed47715bba39dcade25ec4
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Mon Jul 15 09:14:35 2024 +0530

Deleted one file

commit 6faab91adbee2b02fa17828a7ee99f27a128a9
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:45:28 2024 +0530

    commit the last changes

commit 3c2085563b001f1f9506df18eac39a5706ac67bc
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:44:07 2024 +0530

    commit the last changes

commit ea2abdcf0a74c4187e30aec15b39cdf6808b6f5
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:22:31 2024 +0530

    Commit the changed file

commit 732af825404cbfaef5cd88652c5ceddd4d5ed8
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:15:23 2024 +0530

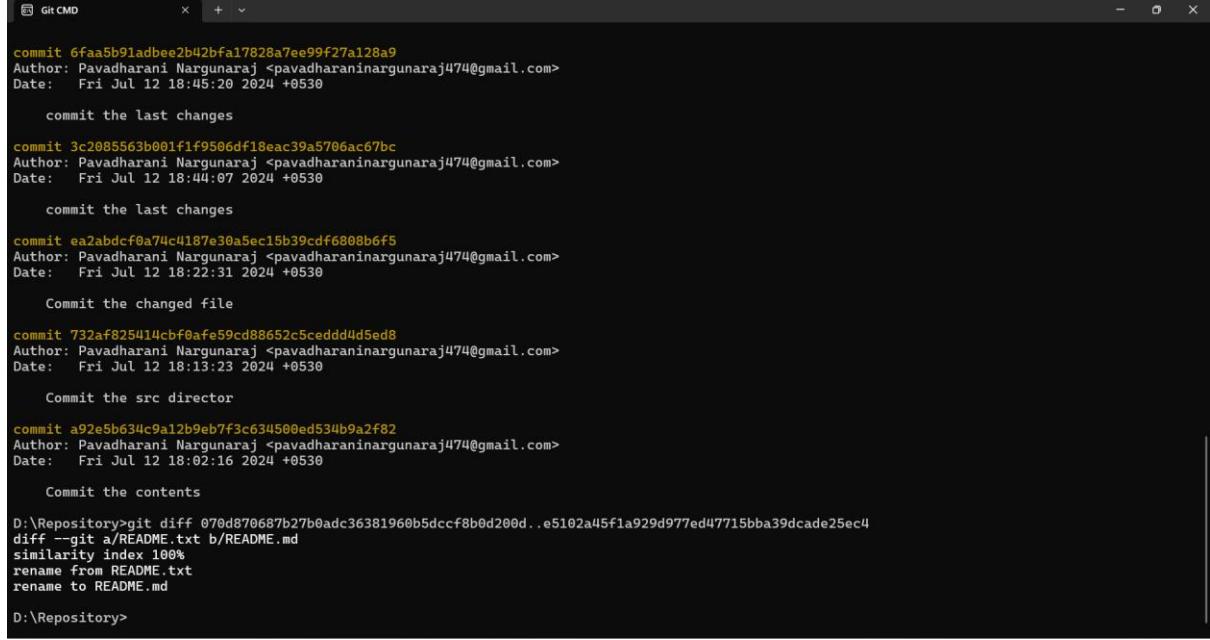
    Commit the src director

commit a92eb634c9a12b9eb7f3c634500ed534b9a2f82
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:02:16 2024 +0530

    Commit the contents

D:\Repository>
```

8. Imagine you want to see a diff that summarizes all that happened between two commit identifiers. Use the **diff** command, specifying two commit identifiers joined by two dots (that is, something like **abc123..def456**). Check the output is what you expect.



```
Git CMD
commit 6faa5b91adbee2b42bfa17828a7ee99f27a128a9
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:45:20 2024 +0530

    commit the last changes

commit 3c2085563b001f1f9506df18eac39a5706ac67bc
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:44:07 2024 +0530

    commit the last changes

commit ea2abdcf0a74c4187e30aec15b39cdf6808b6f5
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:22:31 2024 +0530

    Commit the changed file

commit 732af825414cbf0afe59cd88652c5ceddd4d5ed8
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:13:23 2024 +0530

    Commit the src director

commit a92e5b634c9a12b9eb7f3c634500ed534b9a2f82
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:02:16 2024 +0530

    Commit the contents

D:\Repository>git diff 070d87067b27b0adc36381960b5dccf8b0d200d..e5102a45f1a929d977ed47715bba39dcade25ec4
diff --git a/README.txt b/README.md
similarity index 100%
rename from README.txt
rename to README.md

D:\Repository>
```

Exercise 2

Main Task

1. Run the **status** command. Notice how it tells you what branch you are in.

```
Git CMD
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:22:31 2024 +0530

Commit the changed file

commit 732af825414cbf0afe59cd88652c5ceddd4d5ed8
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:13:23 2024 +0530

Commit the src director

commit a92e5b634c9a12b9eb7f3c634500ed534b9a2f82
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:02:16 2024 +0530

Commit the contents

D:\Repository>git diff 070d870687b27b0adc36381960b5dccf8b0d200d..e5102a45f1a929d977ed47715bba39dcade25ec4
diff --git a/README.txt b/README.md
similarity index 100%
rename from README.txt
rename to README.md

D:\Repository>git status
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:   README.txt
    deleted:   src/File2.txt

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

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

D:\Repository>
```

2. Use the **branch** command to create a new branch.

```
Git CMD
Commit the changed file

commit 732af825414cbf0afe59cd88652c5ceddd4d5ed8
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:13:23 2024 +0530

Commit the src director

commit a92e5b634c9a12b9eb7f3c634500ed534b9a2f82
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date:   Fri Jul 12 18:02:16 2024 +0530

Commit the contents

D:\Repository>git diff 070d870687b27b0adc36381960b5dccf8b0d200d..e5102a45f1a929d977ed47715bba39dcade25ec4
diff --git a/README.txt b/README.md
similarity index 100%
rename from README.txt
rename to README.md

D:\Repository>git status
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:   README.txt
    deleted:   src/File2.txt

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

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

D:\Repository>git branch MyBranch
D:\Repository>
```

3. Use the **checkout** command to switch to it.

```
Git CMD x + ~
Date: Fri Jul 12 18:13:23 2024 +0530
Commit the src director
commit a92e5b634c9a12b9eb7f3c634500ed534b9a2f82
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:02:16 2024 +0530
Commit the contents
D:\Repository>git diff 070d870687b27b0adc36381960b5dccf8b0d200d..e5102a45f1a929d977ed47715bba39dcade25ec4
diff --git a/README.txt b/README.md
similarity index 100%
rename from README.txt
rename to README.md
D:\Repository>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)
      deleted: README.txt
      deleted: src/File2.txt

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

no changes added to commit (use "git add" and/or "git commit -a")
D:\Repository>git branch MyBranch
D:\Repository>git checkout MyBranch
D       README.txt
D       src/File2.txt
Switched to branch 'MyBranch'
D:\Repository>
```

4. Make a couple of commits in the branch – perhaps adding a new file and/or editing existing ones.

```
Git CMD x + ~
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    File.txt

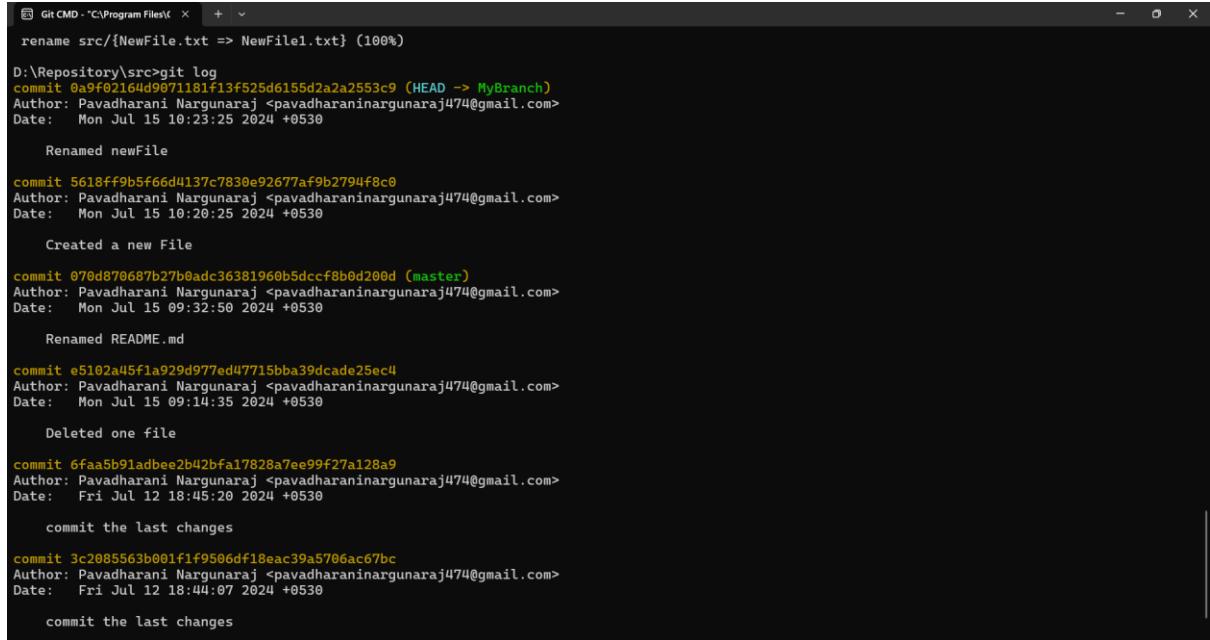
no changes added to commit (use "git add" and/or "git commit -a")
D:\Repository>git branch MyBranch
D:\Repository>git checkout MyBranch
D       README.txt
D       src/File2.txt
Switched to branch 'MyBranch'
D:\Repository>d src
D:\Repository\src>git add NewFile
fatal: pathspec 'NewFile' did not match any files
D:\Repository\src>git add NewFile.txt
'git' is not recognized as an internal or external command,
operable program or batch file.
D:\Repository\src>git add NewFile.txt

D:\Repository\src>git commit -m"Created a new File"
[MyBranch 5618ff9] Created a new File
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 src/NewFile.txt

D:\Repository\src>git mv NewFile.txt NewFile1.txt
D:\Repository\src>git commit -m"Renamed newFile"
[MyBranch 0a9f021] Renamed newFile
  1 file changed, 0 insertions(+), 0 deletions(-)
  rename src/{NewFile.txt => NewFile1.txt} (100%)

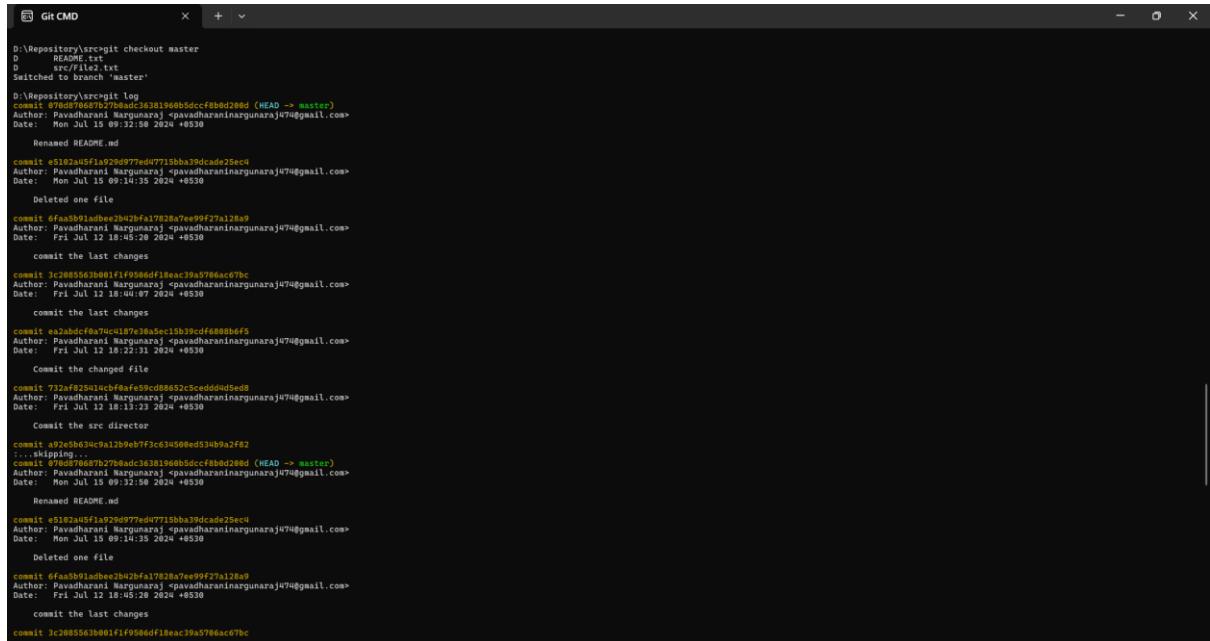
D:\Repository\src>
```

5. Use the **log** command to see the latest commits. The two you just made should be at the top of the list.



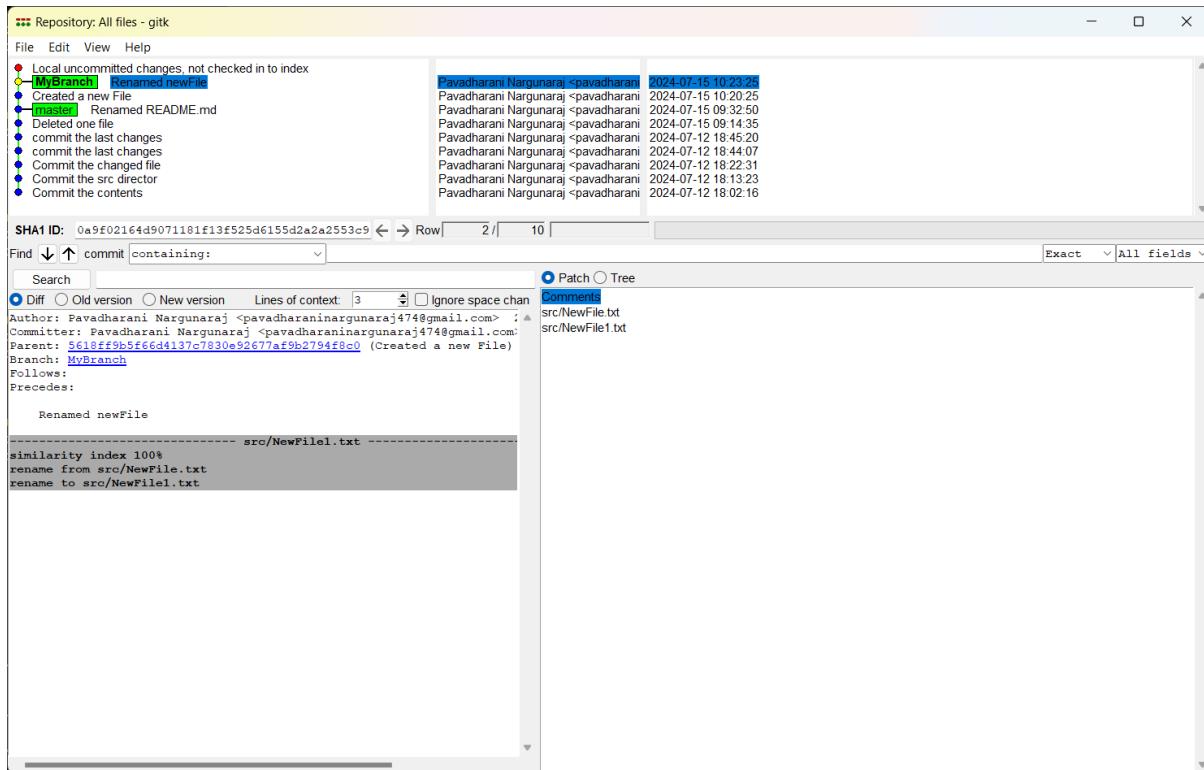
```
Git CMD - 'C:\Program Files\...' + 
rename src/{NewFile.txt => NewFile1.txt} (100%)  
D:\Repository\src>git log  
commit 0a9f02164d9071181f13f525d6155d2a2a2a2553c9 (HEAD -> MyBranch)  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Mon Jul 15 10:23:25 2024 +0530  
  
Renamed newFile  
  
commit 5618ff9b5f66d4137c7830e92677af9b2794f8c0  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Mon Jul 15 10:20:25 2024 +0530  
  
Created a new File  
  
commit 078d8786687b27b0adc36381960b5dccf8b0d200d (master)  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Mon Jul 15 09:32:50 2024 +0530  
  
Renamed README.md  
  
commit e5102a45f1a929d977ed47715bba39dcade25ec4  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Mon Jul 15 09:14:35 2024 +0530  
  
Deleted one file  
  
commit 6faa5b91adbee2b42bfa17828a7ee99f27a128a9  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Fri Jul 12 18:45:20 2024 +0530  
  
commit the last changes  
  
commit 3c2085563b001f1f9506df18ac39a5706ac67bc  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Fri Jul 12 18:44:07 2024 +0530  
  
commit the last changes
```

6. Use the **checkout** command to switch back to the master branch. Run **log** again. Notice your commits don't show up now. Check the files also – they should have their original contents.



```
Git CMD - 'C:\Program Files\...' + 
D:\Repository\src>git checkout master  
D:\Repository\src>git log  
commit 078d8786687b27b0adc36381960b5dccf8b0d200d (HEAD -> master)  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Mon Jul 15 09:32:50 2024 +0530  
  
Renamed README.md  
  
commit e5102a45f1a929d977ed47715bba39dcade25ec4  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Mon Jul 15 09:14:35 2024 +0530  
  
Deleted one file  
  
commit 6faa5b91adbee2b42bfa17828a7ee99f27a128a9  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Fri Jul 12 18:45:20 2024 +0530  
  
commit the last changes  
  
commit 3c2085563b001f1f9506df18ac39a5706ac67bc  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Fri Jul 12 18:44:07 2024 +0530  
  
commit the last changes  
  
commit e2abdef674c187e38a5e15b3cdfe6800bf65  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Fri Jul 12 18:22:31 2024 +0530  
  
commit the changed file  
  
commit 732af92514cbfaef9cd8652c5e6dd4d5ed8  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Fri Jul 12 18:13:23 2024 +0530  
  
commit the src director  
  
commit a929b8b34c9a129e9ef7fc3c345000d34b9a2f82  
... skipping.  
commit 0f0089897b27846d30381960b5dccf8b0d200d (HEAD -> master)  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Mon Jul 15 09:32:50 2024 +0530  
  
Renamed README.md  
  
commit e5102a45f1a929d977ed47715bba39dcade25ec4  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Mon Jul 15 09:14:35 2024 +0530  
  
Deleted one file  
  
commit 6faa5b91adbee2b42bfa17828a7ee99f27a128a9  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Fri Jul 12 18:45:20 2024 +0530  
  
commit the last changes  
commit 3c2085563b001f1f9506df18ac39a5706ac67bc
```

7. Use the **checkout** command to switch back to your branch. Use **gitk** to take a look at the commit graph; notice it's linear.



The screenshot shows the gitk graphical interface for a repository named "All files - gitk". The left pane displays a list of commits for the "MyBranch" branch:

- Local uncommitted changes, not checked in to index
- Created a new File
- Renamed README.md
- Deleted one file
- commit the last changes
- commit the last changes
- Commit the changed file
- Commit the src director
- Commit the contents

The right pane shows a linear commit graph with the following details:

- Pavadharani Nargunaraj <pavadharani>** 2024-07-15 10:23:25
Pavadharani Nargunaraj <pavadharani> 2024-07-15 10:20:25
Pavadharani Nargunaraj <pavadharani> 2024-07-15 09:32:50
Pavadharani Nargunaraj <pavadharani> 2024-07-15 09:14:35
Pavadharani Nargunaraj <pavadharani> 2024-07-12 18:45:20
Pavadharani Nargunaraj <pavadharani> 2024-07-12 18:44:07
Pavadharani Nargunaraj <pavadharani> 2024-07-12 18:22:31
Pavadharani Nargunaraj <pavadharani> 2024-07-12 18:13:23
Pavadharani Nargunaraj <pavadharani> 2024-07-12 18:02:16

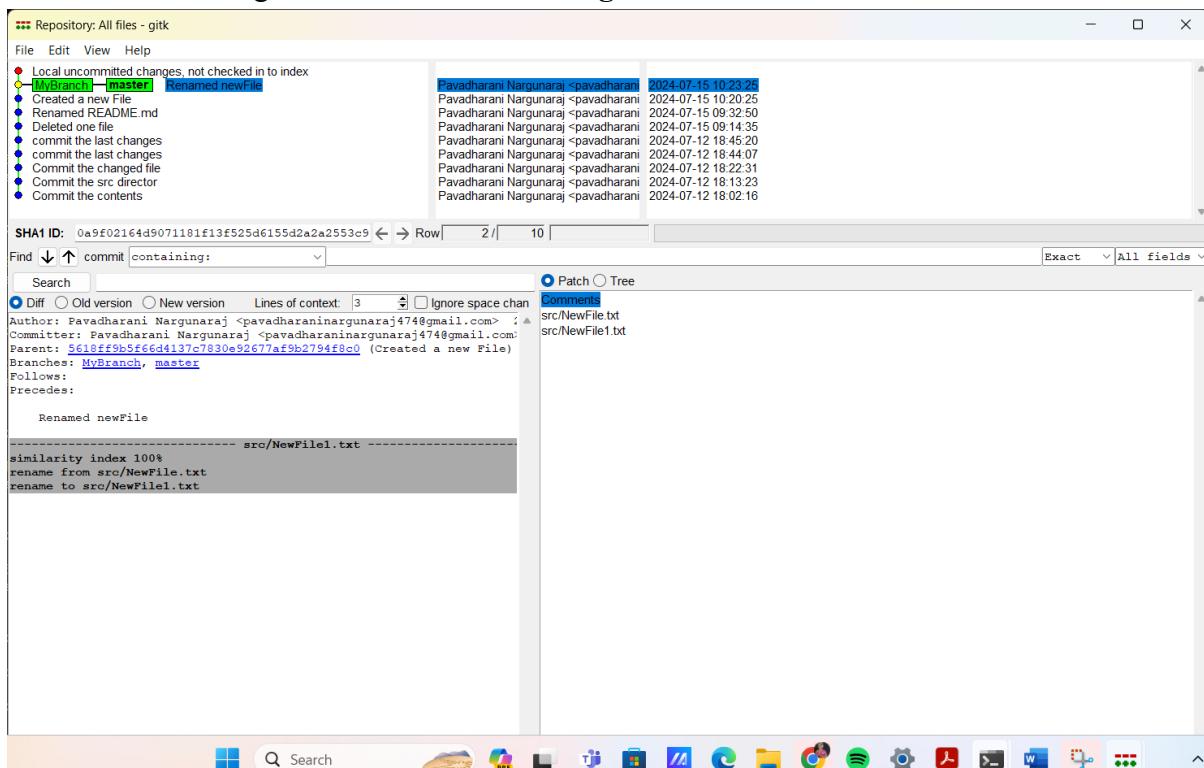
SHA1 ID: 0a9f02164d9071181f13f525d6155d2a2a2a2553c9

Find commit containing: Exact All fields

Search Diff Old version New version Lines of context: 3 Ignore space chan

Comments:
src/NewFile.txt
src/NewFile1.txt

8. Now **checkout** the master branch again. Use the **merge** command to merge your branch in to it. Look for information about it having been a fast-forward merge. Look at **git log**, and see that there is no merge commit. Take a look in **gitk** and see how the DAG is linear.



The screenshot shows the gitk graphical interface for a repository named "All files - gitk". The left pane displays a list of commits for the "master" branch:

- Local uncommitted changes, not checked in to index
- Created a new File
- Renamed README.md
- Deleted one file
- commit the last changes
- commit the last changes
- Commit the changed file
- Commit the src director
- Commit the contents

The right pane shows a linear commit graph with the following details:

- Pavadharani Nargunaraj <pavadharani>** 2024-07-15 10:23:25
Pavadharani Nargunaraj <pavadharani> 2024-07-15 10:20:25
Pavadharani Nargunaraj <pavadharani> 2024-07-15 09:32:50
Pavadharani Nargunaraj <pavadharani> 2024-07-15 09:14:35
Pavadharani Nargunaraj <pavadharani> 2024-07-12 18:45:20
Pavadharani Nargunaraj <pavadharani> 2024-07-12 18:44:07
Pavadharani Nargunaraj <pavadharani> 2024-07-12 18:22:31
Pavadharani Nargunaraj <pavadharani> 2024-07-12 18:13:23
Pavadharani Nargunaraj <pavadharani> 2024-07-12 18:02:16

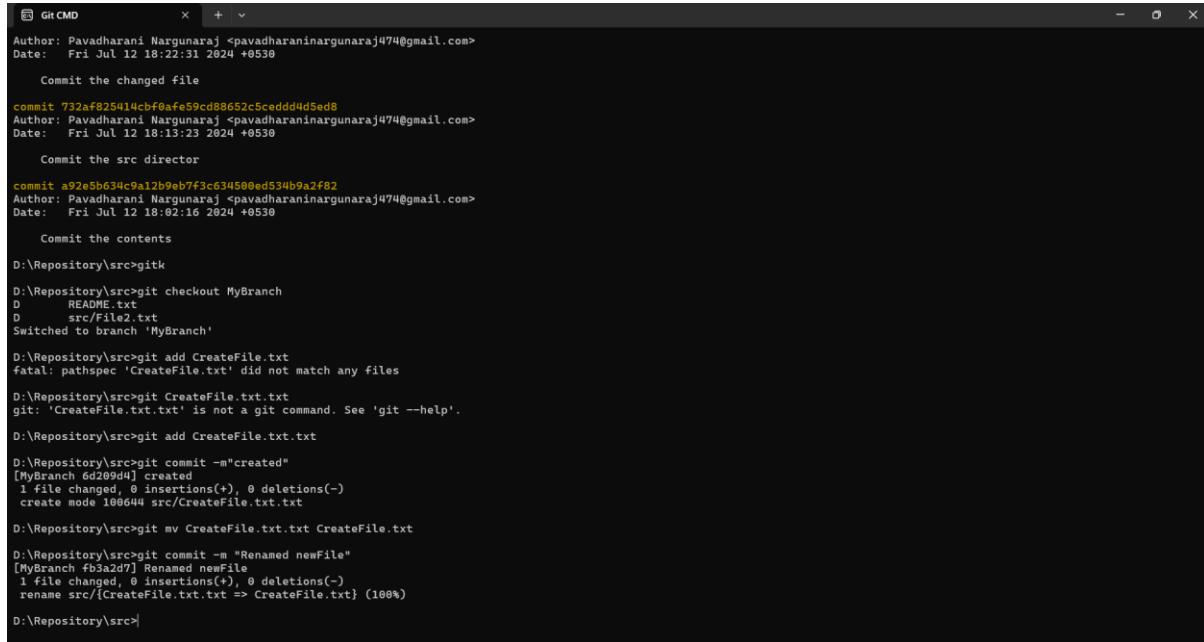
SHA1 ID: 0a9f02164d9071181f13f525d6155d2a2a2a2553c9

Find commit containing: Exact All fields

Search Diff Old version New version Lines of context: 3 Ignore space chan

Comments:
src/NewFile.txt
src/NewFile1.txt

9. Switch back to your branch. Make a couple more commits.



```
Git CMD
Author: Pavadharnani Nargunaraaj <pavadharaninargunaraaj474@gmail.com>
Date:   Fri Jul 12 18:22:31 2024 +0530

Commit the changed file
commit 732af825414cbffafe59cd88652cc5cedd4d5ed8
Author: Pavadharnani Nargunaraaj <pavadharaninargunaraaj474@gmail.com>
Date:   Fri Jul 12 18:15:23 2024 +0530

Commit the src director
commit a92e5b34c9a12b9eb7f3c634580ed534b9a2f82
Author: Pavadharnani Nargunaraaj <pavadharaninargunaraaj474@gmail.com>
Date:   Fri Jul 12 18:02:16 2024 +0530

Commit the contents
D:\Repository\src>gitk

D:\Repository\src>git checkout MyBranch
D      README.txt
D      src/File2.txt
Switched to branch 'MyBranch'

D:\Repository\src>git add CreateFile.txt
fatal: pathspec 'Createfile.txt' did not match any files

D:\Repository\src>git CreateFile.txt.txt
git: 'CreateFile.txt.txt' is not a git command. See 'git --help'.

D:\Repository\src>git add CreateFile.txt.txt

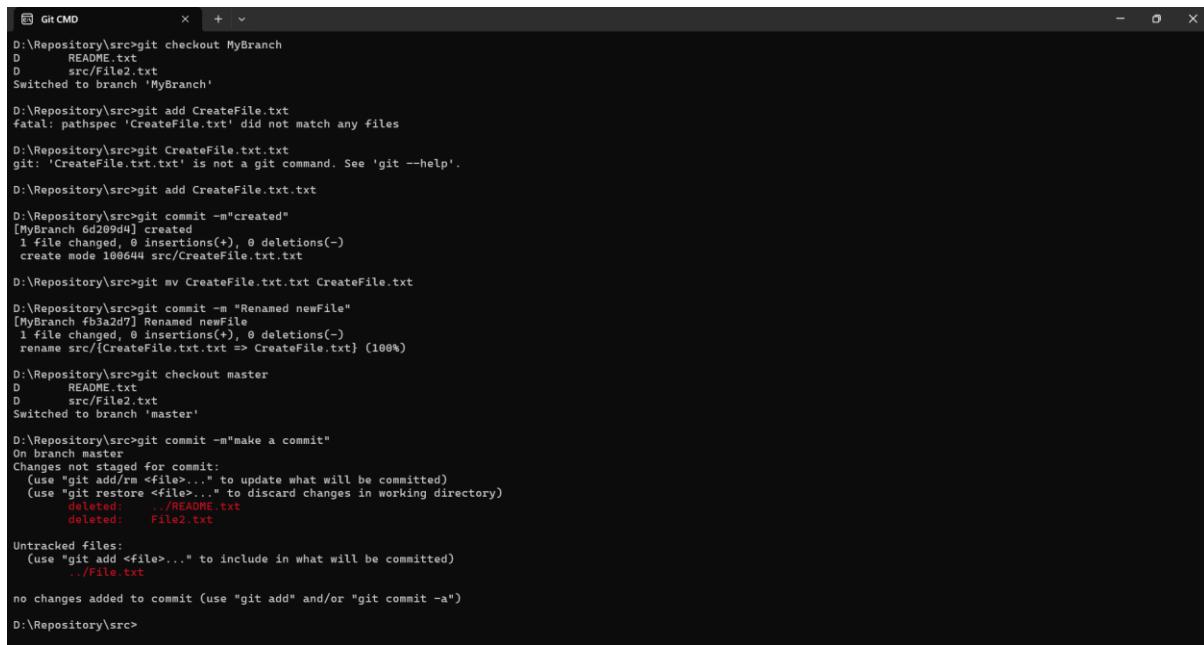
D:\Repository\src>git commit -m"created"
[MyBranch 6d209d4] created
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/CreateFile.txt.txt

D:\Repository\src>git mv Createfile.txt.txt CreateFile.txt

D:\Repository\src>git commit -m "Renamed newFile"
[MyBranch fb3a2d7] Renamed newFile
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename src/{Createfile.txt.txt => CreateFile.txt} (100%)

D:\Repository\src>
```

10. Switch back to master. Make a commit there, which should edit a different file from the ones you touched in your branch – to be sure there is no conflict.



```
Git CMD
D:\Repository\src>git checkout MyBranch
D      README.txt
D      src/File2.txt
Switched to branch 'MyBranch'

D:\Repository\src>git add CreateFile.txt
fatal: pathspec 'Createfile.txt' did not match any files

D:\Repository\src>git CreateFile.txt.txt
git: 'CreateFile.txt.txt' is not a git command. See 'git --help'.

D:\Repository\src>git add CreateFile.txt.txt

D:\Repository\src>git commit -m"created"
[MyBranch 6d209d4] created
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/CreateFile.txt.txt

D:\Repository\src>git commit -m "Renamed newFile"
[MyBranch fb3a2d7] Renamed newFile
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename src/{Createfile.txt.txt => CreateFile.txt} (100%)

D:\Repository\src>git checkout master
D      README.txt
D      src/File2.txt
Switched to branch 'master'

D:\Repository\src>git commit -m"make a commit"
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:   ./README.txt
      deleted:   File2.txt

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

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

D:\Repository\src>
```

Yes, there is no Conflict.

11. Now **merge** your branch again. (Aside: you don't need to do anything to inform Git that you only want to merge things added since your previous merge. Due to the way Git works, that kind of issue simply does not come up, unlike in early versions of Subversion.)

```

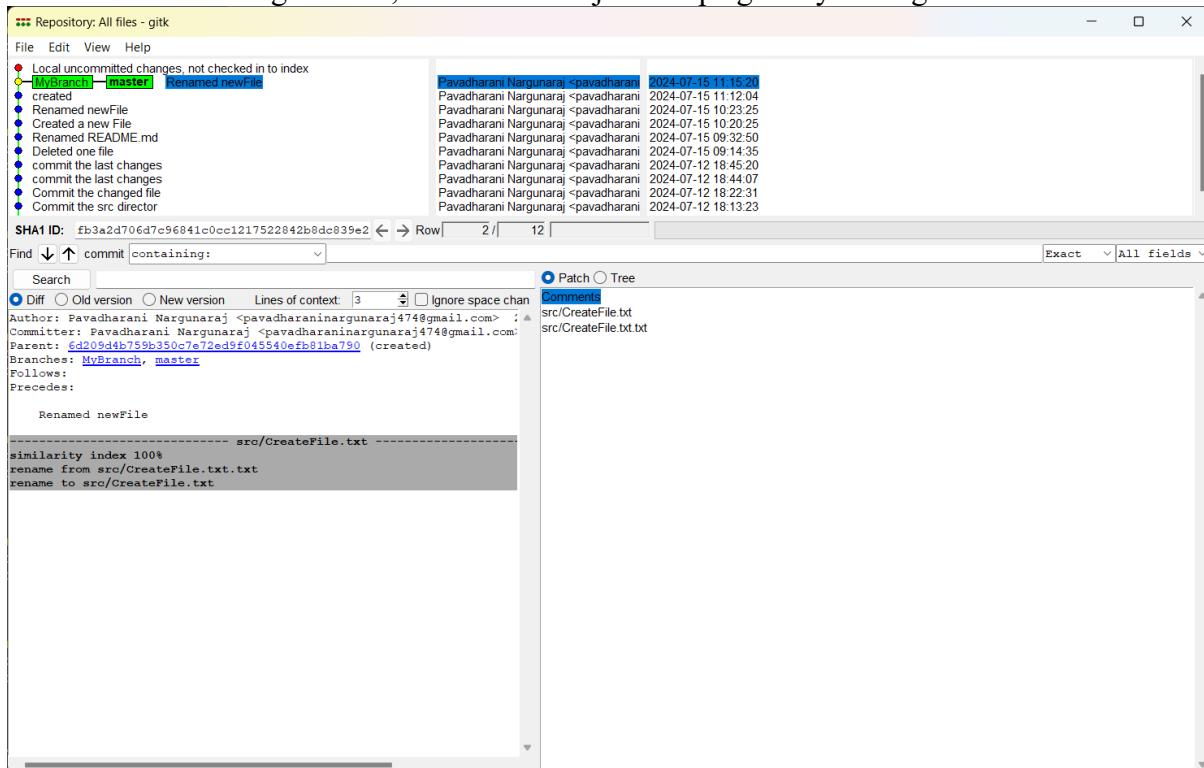
Git CMD
D:\Repository\src>git CreateFile.txt.txt
git: 'CreateFile.txt.txt' is not a git command. See 'git --help'.
D:\Repository\src>git add CreateFile.txt.txt
D:\Repository\src>git commit -m"created"
[MyBranch 6d289d4] created
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/CreateFile.txt.txt
D:\Repository\src>git mv CreateFile.txt.txt CreateFile.txt
D:\Repository\src>git commit -m "Renamed newFile"
[MyBranch fb3a2d7] Renamed newFile
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename src/{CreateFile.txt.txt => CreateFile.txt} (100%)
D:\Repository\src>git checkout master
D   README.txt
D   src/File2.txt
Switched to branch 'master'
D:\Repository\src>git commit -m"make a commit"
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:  ./README.txt
    deleted:  File2.txt

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

no changes added to commit (use "git add" and/or "git commit -a")
D:\Repository\src>git merge MyBranch
Updating 6a9f021..fb3a2d7
Fast-forward
 src/CreateFile.txt | 8
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/CreateFile.txt
D:\Repository\src>

```

12. Look at **git log**. Notice that there is a merge commit. Also look in **gitk**. Notice the DAG now shows how things forked, and then were joined up again by a merge commit.



Stretch Task

- Once again, **checkout** your branch. Make a couple of commits.

```
Git CMD x + v
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:44:07 2024 +0530
    commit the last changes
commit ea2abdcf0a74c4187e30a5ec15b39cdf6808b6f5
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:22:31 2024 +0530
    Commit the changed file
commit 732af825d14cbf0afe59cd88652c5ce0dd4d5ed8
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:13:23 2024 +0530
    Commit the src director
commit a92e5bd34c9a12b9eb7f3c634580ed534b9a2f82
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Fri Jul 12 18:02:16 2024 +0530
    Commit the contents
D:\Repository\src>gitk
D:\Repository\src>git checkout MyBranch
D   README.txt
D   src\File2.txt
Switched to branch 'MyBranch'
D:\Repository\src>git add NewFile2.txt
D:\Repository\src>git commit -m"commit the new file"
[MyBranch 68db386] commit the new file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/NewFile2.txt
D:\Repository\src>git mv NewFile2.txt File2.txt
D:\Repository\src>git commit -m"Renamed"
[MyBranch 2cec695] Renamed
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 src/NewFile2.txt
D:\Repository\src>
D:\Repository\src>
```

- Return to your master branch. Make a commit there that changes the exact same line, or lines, as commits in your branch did.

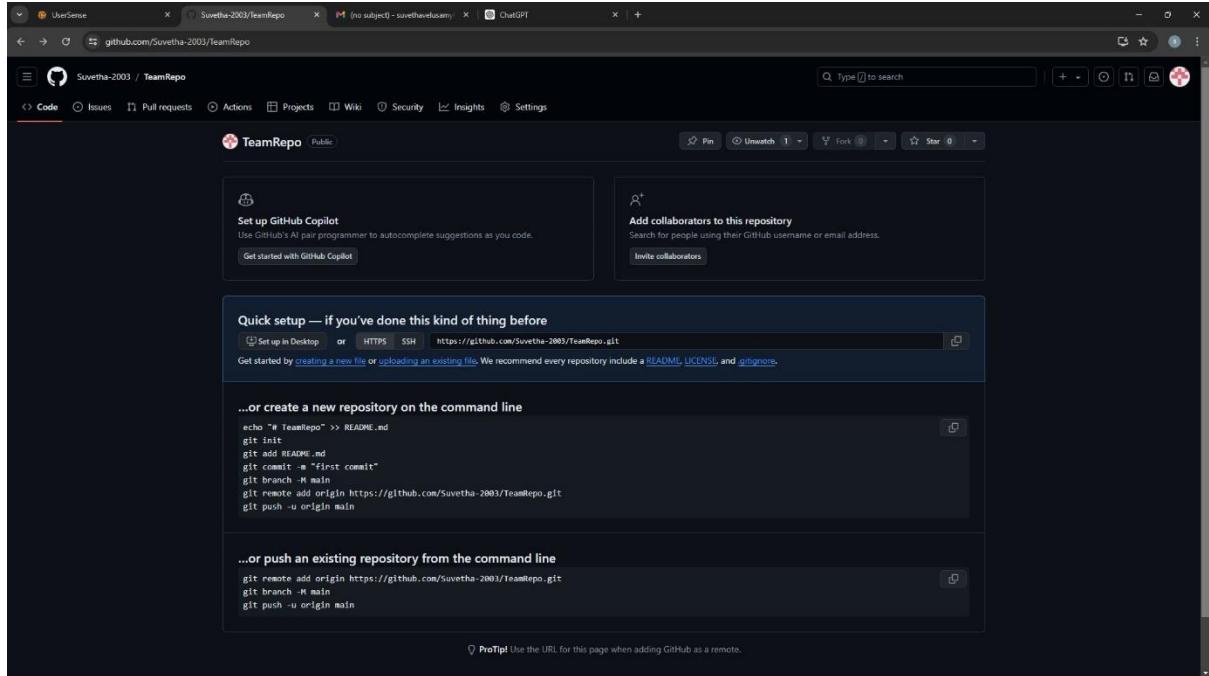
```
Git CMD x + v
Switched to branch 'MyBranch'
D:\Repository\src>git add NewFile2.txt
D:\Repository\src>git commit -m"commit the new file"
[MyBranch 68db386] commit the new file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/NewFile2.txt
D:\Repository\src>git mv NewFile2.txt File2.txt
D:\Repository\src>git commit -m"Renamed"
[MyBranch 2cec695] Renamed
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 src/Newfile2.txt
D:\Repository\src>git checkout master
D   README.txt
Switched to branch 'master'
D:\Repository\src>git add File2.txt
D:\Repository\src>git commit -m"commit the same file in master"
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:  ./README.txt
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  ./file.txt
no changes added to commit (use "git add" and/or "git commit -a")
D:\Repository\src>git add NewFile2.txt
fatal: pathspec 'NewFile2.txt' did not match any files
D:\Repository\src>git mv File2.txt NewOne.txt
D:\Repository\src>git commit -m"check the changes"
[master 9e8a8a8] check the changes
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename src/{File2.txt => NewOne.txt} (100%)
D:\Repository\src>
```

Exercise 3

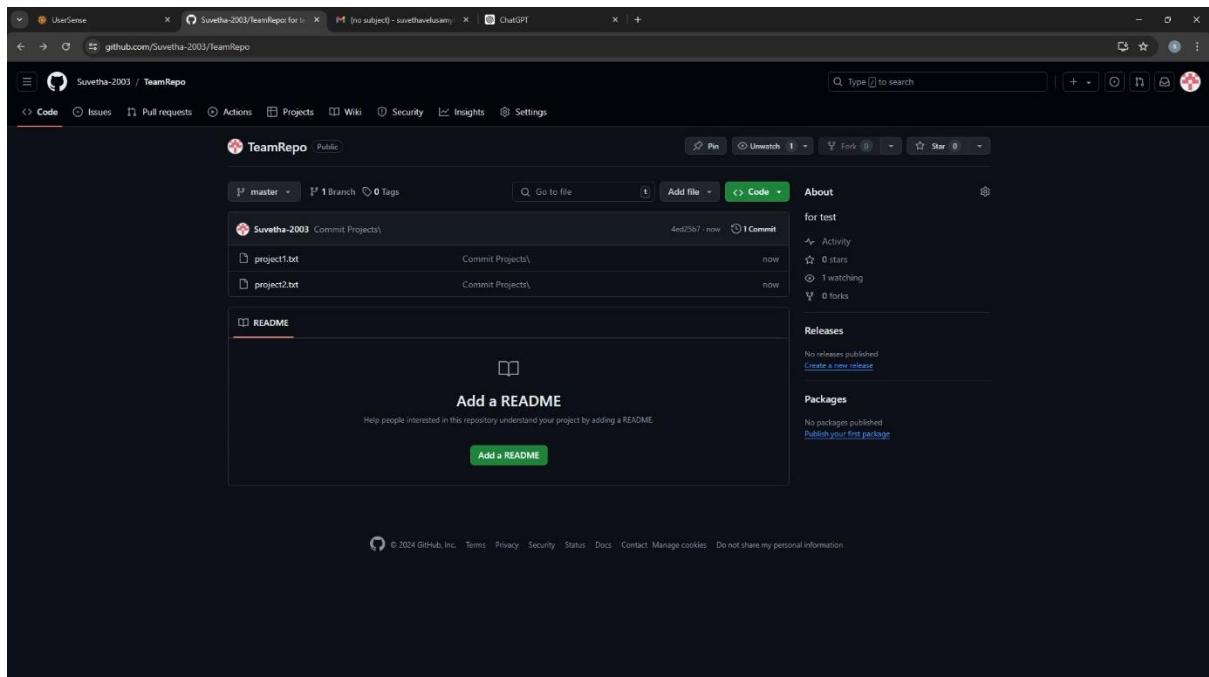
For this task, you will work in a small group. Between 2 and 4 people is about right.

Main Task

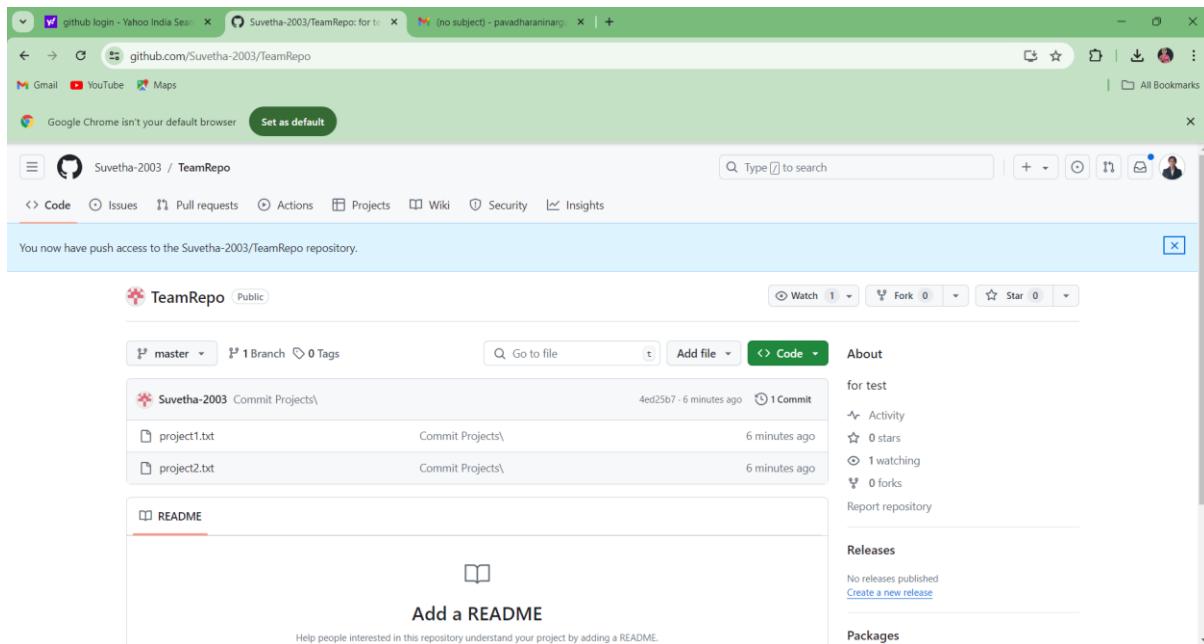
1. First, one person in the group should create a public repository using their GitHub account.



2. This same person should then follow the instructions from GitHub to add a **remote**, and then **push** their repository. Do not forget the **-u** flag, as suggested by GitHub!

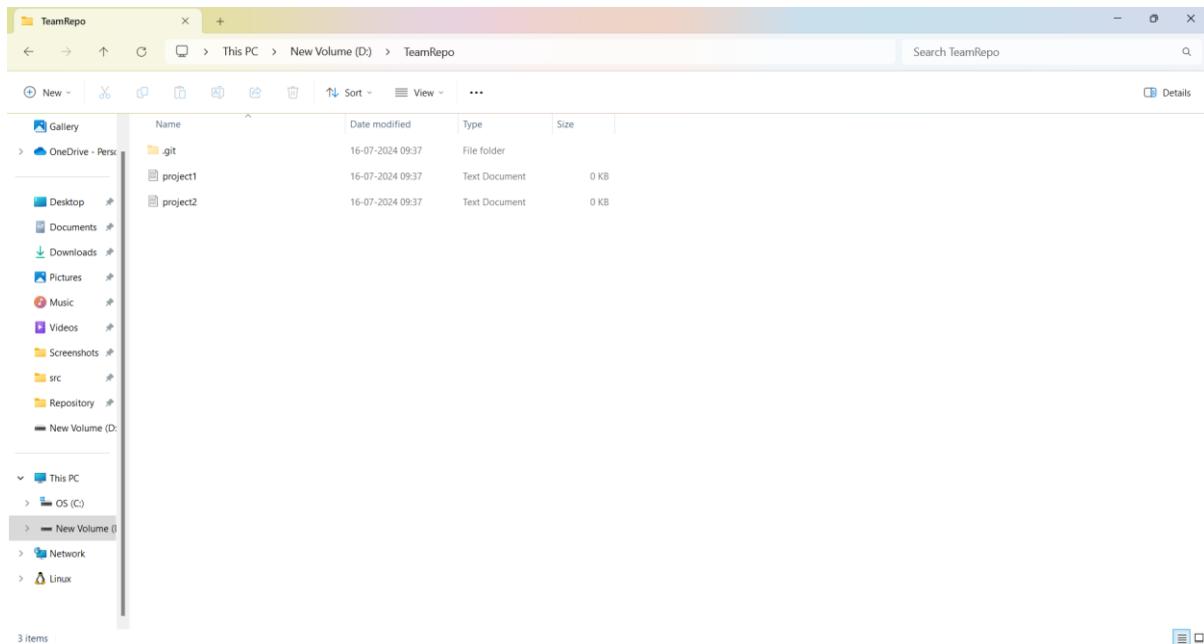


3. All of the other members of the group should then be added as collaborators, so they can commit to the repository also.



The screenshot shows a web browser window with three tabs open. The active tab is 'Suvetha-2003/TeamRepo' on GitHub. The page displays a public repository named 'TeamRepo'. It shows two commits from 'Suvetha-2003' with the message 'Commit Projects'. Below the commits is a 'README' section with a placeholder 'Add a README'. On the right side of the repository page, there is an 'About' section with basic repository statistics: 1 commit, 0 stars, 1 watching, and 0 forks. There are also sections for 'Releases' and 'Packages'.

4. Next, everyone else in the group should **clone** the repository from GitHub. Verify that the context of the repository is what is expected.



The screenshot shows a Windows File Explorer window displaying the contents of a cloned GitHub repository. The repository is located at 'D:\TeamRepo'. The folder structure includes a '.git' folder and two text files named 'project1.txt' and 'project2.txt', both of which were committed 6 minutes ago. The file explorer interface shows standard navigation tools like back, forward, and search.

5. One of the group members who just cloned should now make a local **commit**, then **push** it. Everyone should verify that when they **pull**, that commit is added to their local repository (use **git log** to check for it).

```
D:\TeamRepo>git add .  
D:\TeamRepo>git commit -m "commit all the file"  
On branch master  
Your branch is up to date with 'origin/master'.  
nothing to commit, working tree clean  
D:\TeamRepo>git add .  
D:\TeamRepo>git commit -m "commit changes"  
[master 7ec4a64] commit changes  
1 file changed, 1 insertion(+)  
D:\TeamRepo>git remote add origin "https://github.com/Suvetha-2003/TeamRepo.git"  
error: remote origin already exists.  
D:\TeamRepo>git push -u origin master  
Enumerating objects: 5, done.  
Counting objects: 100% (5/5), done.  
Delta compression using up to 12 threads  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (3/3), 298 bytes | 149.00 KiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
To https://github.com/Suvetha-2003/TeamRepo.git  
 4ed25b7..7ec4a64 master -> master  
branch 'master' set up to track 'origin/master'.  
D:\TeamRepo>
```

6. Look at each other's **git log** output. Notice how the SHA-1 is the same for a given commit across every copy of the repository. Why is this important?

```
D:\TeamRepo>git log  
commit 7ec4a64c5b8f39881c6d809a71ce735f7cb55079 (HEAD -> master, origin/master, origin/HEAD)  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Tue Jul 16 09:45:39 2024 +0530  
commit changes  
commit 4ed25b75edc87f0755486b42fcfa03d41d6b06374  
Author: suvetha <suvethavelusamy697@gmail.com>  
Date: Tue Jul 16 09:20:43 2024 +0530  
Commit Projects\  
D:\TeamRepo>
```

This helps to track the History . Because all the team members should know what are the changes happened and to know who made it.

7. Two members of the group should now make a **commit** locally, and race to **push** it. To keep things simple, be sure to edit different files. What happens to the runner-up?

A screenshot of a Windows desktop environment. At the top is a black terminal window titled 'GIT CMD' with white text. The text inside the window shows a sequence of git commands and their output. It starts with 'branch 'master' set up to track 'origin/master''. Then, a commit is made to 'project2'. When attempting to push this commit to 'origin/master', the command fails with an error message: 'error: failed to push some refs to 'https://github.com/Suvetha-2003/TeamRepo.git''. Below this, another commit is made to 'project-name'. Finally, a push attempt is made to 'origin master', which also fails with the same error message. The taskbar at the bottom of the screen shows various pinned icons and the date/time '16-07-2024 09:58 AM'.

```
branch 'master' set up to track 'origin/master'.
Everything up-to-date

D:\Team-Repo>git add .

D:\Team-Repo>git commit -m "project2 commit"
[master 7c54ac2] project2 commit
 1 file changed, 1 insertion(+)

D:\Team-Repo>git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 294 bytes | 294.00 KiB/s, done.
Total 3 (delta 0), reused 0 (from 0)
To https://github.com/Suvetha-2003/TeamRepo.git
 ! [rejected]      master --> master (cannot lock ref 'refs/heads/master': is at 836bf7565ab8d8a22ab87388b1f4e39188e24958 but expected 7ec4a64c5b8f39881c6d809a71ce735f7cb55079)
error: failed to push some refs to 'https://github.com/Suvetha-2003/TeamRepo.git'

D:\Team-Repo>git add .

D:\Team-Repo>git commit -m "project-name commit"
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

D:\Team-Repo>git push -u origin master
To https://github.com/Suvetha-2003/TeamRepo.git
 ! [rejected]      master --> master (fetch first)
error: failed to push some refs to 'https://github.com/Suvetha-2003/TeamRepo.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

D:\Team-Repo>
```

8. The runner-up should now **pull**. As a group, look at the output of the command. Additionally, look at the **git log**, and notice that there is a merge commit. You may also wish to view the DAG in **gitk**.

A screenshot of the 'gitk' graphical interface. The main window displays a commit history for the 'master' branch. The first commit, 'project 1 changes', was made by 'Dheetchanaa <kanagarajdhheetchanaa@gmail.com>' on 2024-07-16 09:55:05. It has two parents: 'commit changes' (SHA1: 7ec4a64c5b8f39881c6d809a71ce735f7cb55079) and 'Commit Projects' (SHA1: 836bf7565ab8d8a22ab87388b1f4e39188e24958). The second commit, 'project 1 changes', was made by 'Suvetha <suvethavelusamy697@gmail.com>' on 2024-07-16 09:20:43. A 'Find' bar at the top allows searching for commits containing specific text. Below the commit list, a detailed view of the first commit's content is shown in a 'Patch' or 'Tree' format. The patch shows the addition of a file named 'project1.txt' with the following content:

```
index a5752ed..32e17fc 100644
@@ -1 +1 @@
-Team project
` No newline at end of file
+Team project-1
` No newline at end of file
```

9. Repeat the last two steps a couple of times, to practice.

```
GRCMD x + 
Author: Dheetchanaa <kanagarajdheetchanaa@gmail.com>
Date: Tue Jul 16 09:55:05 2024 +0530

project 1 changes

commit 7ec4a64c5b8f39881c6d809a71ce735f7cb55079
Author: Pavadharani Nargunaraj <pavadharaninargunaraaj474@gmail.com>
Date: Tue Jul 16 09:45:39 2024 +0530

    commit changes

commit 4ed25b75edc87f0755486b42fca03d41d6b06374
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Tue Jul 16 09:20:43 2024 +0530

    Commit Projects\

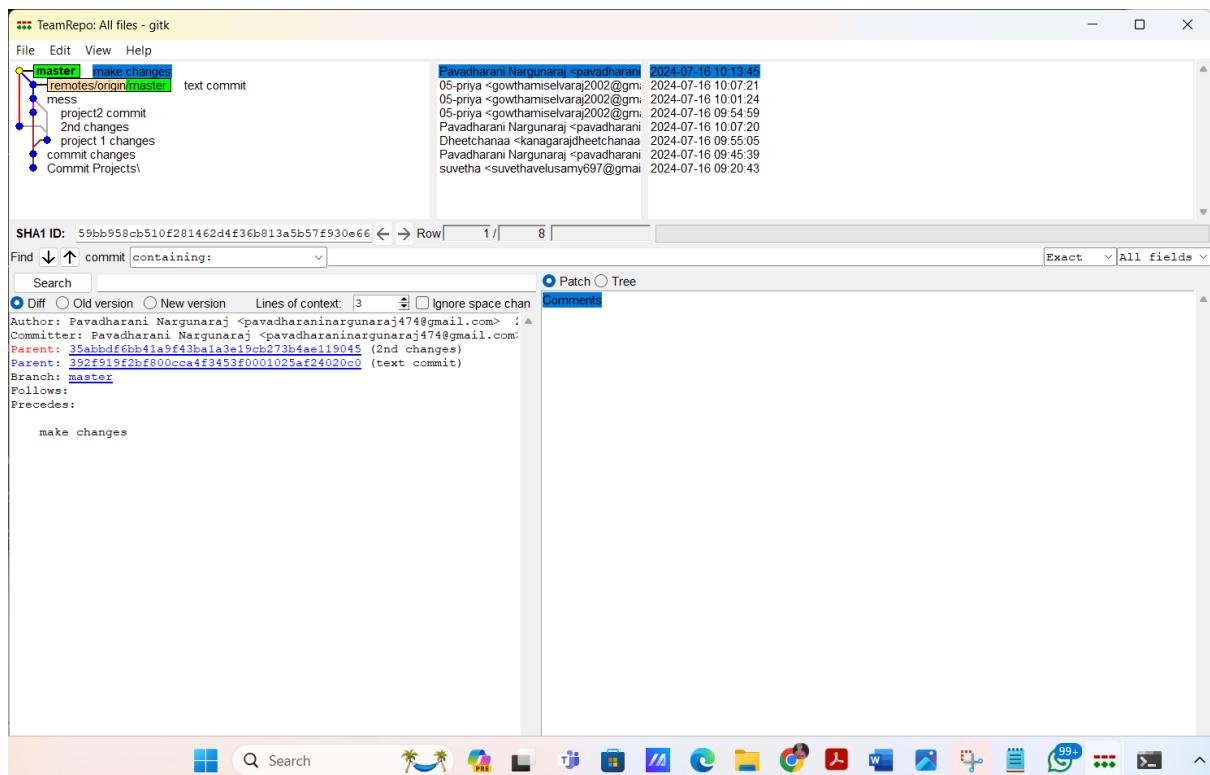
D:\TeamRepo>gitk

D:\TeamRepo>git add .

D:\TeamRepo>git commit -m " 2nd changes"
[master 35abddf] 2nd changes
 1 file changed, 3 insertions(+), 1 deletion(-)

D:\TeamRepo>git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 312 bytes | 156.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Suvetha-2003/TeamRepo.git
! [remote rejected] master -> master (cannot lock ref 'refs/heads/master': is at 392f919f2bf800ccaf3453f0001025af24020c0 but expected 836bf756
5ab8a22ab87388b1f4e39188e24958)
error: failed to push some refs to 'https://github.com/Suvetha-2003/TeamRepo.git'

D:\TeamRepo>
```



Stretch Task

1. Now create a situation where two group members both edit the same line in the same file and **commit** it locally. Race to **push**.

```
git CMD x + v
Automatic merge failed; fix conflicts and then commit the result.

D:\TeamRepo>git add .

D:\TeamRepo>git commit -m "commit all the changes"
[master 16d38dd] commit all the changes

D:\TeamRepo>git pull origin master
From https://github.com/Suventha-2003/TeamRepo
 * branch            master      -> FETCH_HEAD
Already up to date.

D:\TeamRepo>git add .

D:\TeamRepo>git commit -m "change"
[master 1fd598b] change
 1 file changed, 1 insertion(+)

D:\TeamRepo>git push -u origin master
Enumerating objects: 20, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (11/11), 1.28 KiB | 245.00 KiB/s, done.
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Suventha-2003/TeamRepo.git
 fe040a7..1fd598b master -> master
branch 'master' set up to track 'origin/master'.

D:\TeamRepo>=
```

```
git CMD x + v
Date: Tue Jul 16 09:45:39 2024 +0530
commit changes
commit 4ed25b75edc87f0755486b42fc0a03d41d6b06374
Author: suventha <suvenhavelsamay69@gmail.com>
Date: Tue Jul 16 09:26:43 2024 +0530
  Commit Projects\

D:\TeamRepo>git pull origin master
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 16 (delta 0), reused 16 (delta 0), pack-reused 0
Unpacking objects: 100% (16/16), 1.64 KiB | 36.00 KiB/s, done.
From https://github.com/Suventha-2003/TeamRepo
 * branch            master      -> FETCH_HEAD
   a08d925..fe040a7  master      -> origin/master
Up-to-date with origin/master.
Fast-forward
  project1.txt | 1 -
  1 file changed, 1 deletion(-)

D:\TeamRepo>git add .

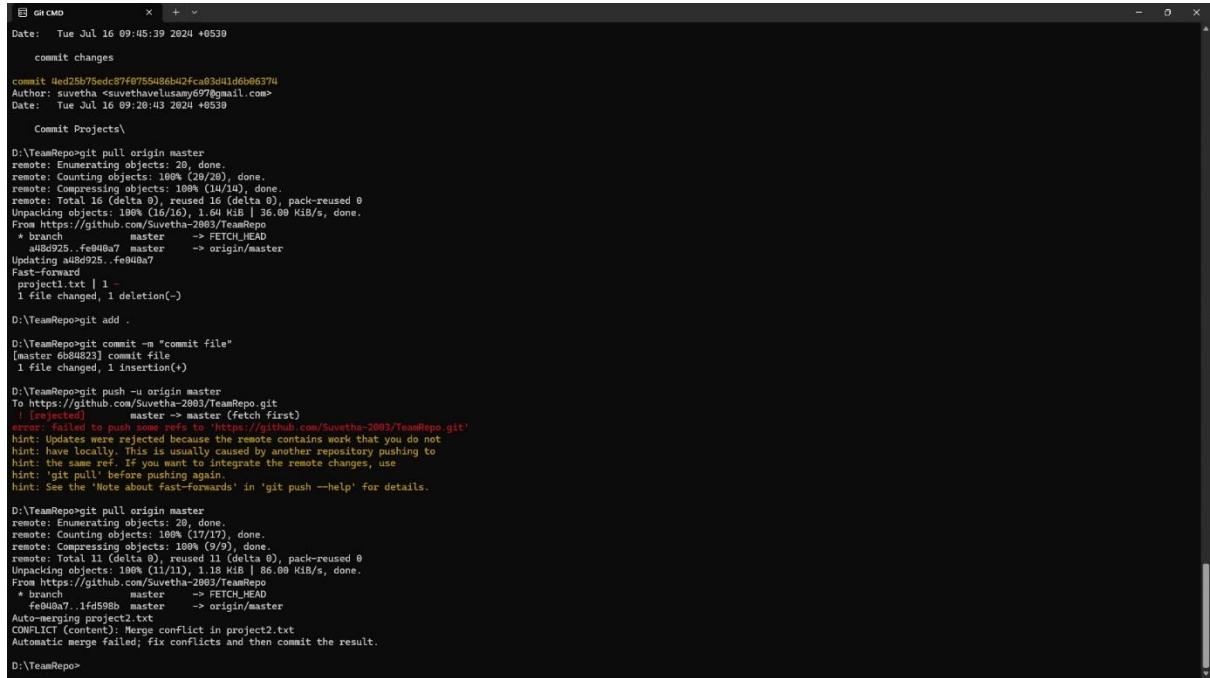
D:\TeamRepo>git commit -m "commit file"
[master 6b8a023] commit file
 1 file changed, 1 insertion(+)

D:\TeamRepo>git push -u origin master
To https://github.com/Suventha-2003/TeamRepo.git
 ! [rejected]        master -> master (fetch first)
error: failed to push local master to remote master
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

D:\TeamRepo>git pull origin master
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 11 (delta 0), reused 11 (delta 0), pack-reused 0
Unpacking objects: 100% (11/11), 1.17 KiB | 86.00 KiB/s, done.
From https://github.com/Suventha-2003/TeamRepo
 * branch            master      -> FETCH_HEAD
   fe040a7..1fd598b  master      -> origin/master
Auto-merging project2.txt
CONFLICT (content): Merge conflict in project2.txt
Automatic merge failed; fix conflicts and then commit the result.

D:\TeamRepo>
```

2. When the runner-up does a **pull**, they should get a merge conflict.



```
git cmd
Date: Tue Jul 16 09:45:39 2024 +0530
commit changes
commit 4ed25b75edc87f0755486b42#ca03d41d6b06374
Author: suvetha <suvethavellisamy69@gmail.com>
Date: Tue Jul 16 09:26:45 2024 +0530
Commit Projects\

D:\TeamRepo>git pull origin master
remote: Enumerating objects: 20, done.
remote: Counting objects: 20 (delta 0), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 16 (delta 0), reused 16 (delta 0), pack-reused 0
Unpacking objects: 100% (16/16), 1.64 KiB | 36.00 KiB/s, done.
From https://github.com/Suvetha-2003/TeamRepo
 * branch      master    -> FETCH_HEAD
   a8d925..fe040a7  master    -> origin/master
Updating a8d925..fe040a7
Fast-forward
 project1.txt | 1 -
 1 file changed, 1 deletion(-)

D:\TeamRepo>git add .

D:\TeamRepo>git commit -m "commit file"
[master 6b8a823] commit file
 1 file changed, 1 insertion(+)

D:\TeamRepo>git push -u origin master
To https://github.com/Suvetha-2003/TeamRepo.git
 ! [rejected]        master --> master (fetch first)
error: failed to push local master to remote
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

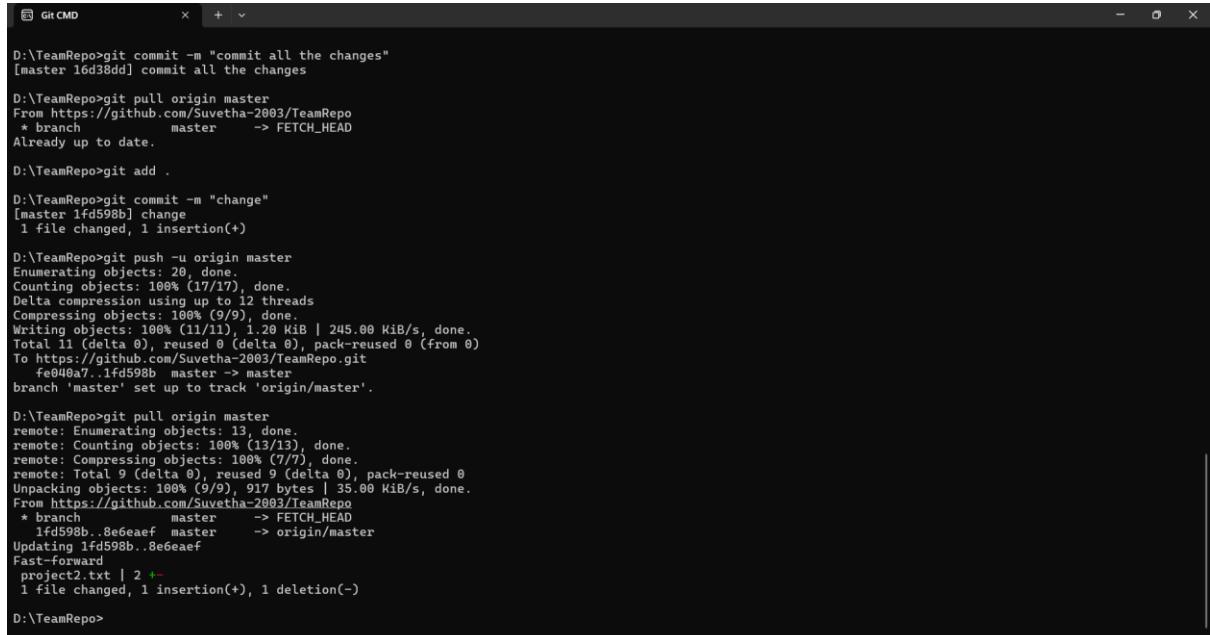
D:\TeamRepo>git pull origin master
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 11 (delta 0), reused 11 (delta 0), pack-reused 0
Unpacking objects: 100% (11/11), 1.20 KiB | 86.00 KiB/s, done.
From https://github.com/Suvetha-2003/TeamRepo
 * branch      master    -> FETCH_HEAD
   fe040a7..1fd598b  master    -> origin/master
Auto-merging project2.txt
CONFLICT (content): Merge conflict in project2.txt
Automatic merge failed; fix conflicts and then commit the result.

D:\TeamRepo>
```

3. Look as a group at the file in conflict, and resolve it.

4. Use the **add** command to stage the fix, and then use **commit** to make the merge commit.

Notice how this procedure is exactly the one you got used to when resolving conflicts in branches.



```
git cmd
D:\TeamRepo>git commit -m "commit all the changes"
[master 16d38dd] commit all the changes

D:\TeamRepo>git pull origin master
From https://github.com/Suvetha-2003/TeamRepo
 * branch      master    -> FETCH_HEAD
Already up to date.

D:\TeamRepo>git add .

D:\TeamRepo>git commit -m "change"
[master 1fd598b] change
 1 file changed, 1 insertion(+)

D:\TeamRepo>git push -u origin master
Enumerating objects: 20, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (11/11), 1.20 KiB | 245.00 KiB/s, done.
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Suvetha-2003/TeamRepo.git
  fe040a7..1fd598b  master -> master
branch 'master' set up to track 'origin/master'.

D:\TeamRepo>git pull origin master
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 0), reused 9 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), 917 bytes | 35.00 KiB/s, done.
From https://github.com/Suvetha-2003/TeamRepo
 * branch      master    -> FETCH_HEAD
   1fd598b..8e6eaef  master    -> origin/master
Updating 1fd598b..8e6eaef
Fast-forward
 project2.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\TeamRepo>
```

Exercise 4

Main Task

1. Make a commit, and make a silly typo in the commit message.

```
C:\Users\PAVADHARANI>D:  
D:>>cd TeamRepo  
D:\TeamRepo>git add .  
D:\TeamRepo>git commit -m "refresh commit"  
On branch master  
Your branch is up to date with 'origin/master'.  
nothing to commit, working tree clean  
D:\TeamRepo>
```

2. Use the **--amend** flag to enable you to fix the commit message.

3. Look at the **log** and notice how the mistake is magically gone.

```
C:\Users\PAVADHARANI>D:  
D:\>cd TeamRepo  
D:\TeamRepo>git log  
commit 8beeaef0edd3a3a6ba9cf8c8fe9d6edfc96eea2d (HEAD -> master, origin/master, origin/HEAD)  
Author: suvetha <suvethavelusamy697@gmail.com>  
Date: Tue Jul 16 11:04:47 2024 +0530  
  
    commit change file  
  
commit c482e6f64e35d04b4505bab6c45253941a59ebf8  
Merge: 6b84823 1fd598b  
Author: suvetha <suvethavelusamy697@gmail.com>  
Date: Tue Jul 16 10:59:09 2024 +0530  
  
    commit the changes  
  
commit 6b8482367de04d68f4460aad1f7220d100e5c58a  
Author: suvetha <suvethavelusamy697@gmail.com>  
Date: Tue Jul 16 10:41:04 2024 +0530  
  
    commit file  
  
commit 1fd598b29a42535b9a79ee08c9ef1d6151811a5d  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Tue Jul 16 10:40:47 2024 +0530  
  
    change  
  
commit 16d38dd99ac5b05b5868764af7a538a6b07e77e7  
Merge: 89bb958 fe040a7  
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>  
Date: Tue Jul 16 10:38:42 2024 +0530  
...skipping...  
commit 8beeaef0edd3a3a6ba9cf8c8fe9d6edfc96eea2d (HEAD -> master, origin/master, origin/HEAD)  
Author: suvetha <suvethavelusamy697@gmail.com>  
Date: Tue Jul 16 11:04:47 2024 +0530  
  
    commit change file
```

4. Now make a commit where you make a typo in one of the files. Once again, use **--amend** to magic away your problems.

5. Create a branch. Make a commit.

```
C:\Users\PAVADHARANI>D:  
D:>cd TeamRepo  
D:\TeamRepo>git branch MyBranch  
D:\TeamRepo>git branch  
* MyBranch  
* master  
D:\TeamRepo>git commit -m "branch commit"  
On branch master  
Your branch is ahead of 'origin/master' by 1 commit.  
(use "git push" to publish your local commits)  
nothing to commit, working tree clean  
D:\TeamRepo>git checkout MyBranch  
Switched to branch 'MyBranch'  
D:\TeamRepo>git branch  
* MyBranch  
* master  
D:\TeamRepo>git commit -m "once again"  
On branch MyBranch  
nothing to commit, working tree clean  
D:\TeamRepo>
```

6. Now switch back to your master branch. Make a (non-conflicting) commit there also.

```
C:\Users\PAVADHARANI>D:  
D:>cd TeamRepo  
D:\TeamRepo>git branch MyBranch  
D:\TeamRepo>git branch  
* MyBranch  
* master  
D:\TeamRepo>git commit -m "branch commit"  
On branch master  
Your branch is ahead of 'origin/master' by 1 commit.  
(use "git push" to publish your local commits)  
nothing to commit, working tree clean  
D:\TeamRepo>git checkout MyBranch  
Switched to branch 'MyBranch'  
D:\TeamRepo>git branch  
* MyBranch  
* master  
D:\TeamRepo>git commit -m "once again"  
On branch MyBranch  
nothing to commit, working tree clean  
D:\TeamRepo>git checkout master  
Switched to branch 'master'  
Your branch is ahead of 'origin/master' by 1 commit.  
(use "git push" to publish your local commits)  
D:\TeamRepo>git add .  
D:\TeamRepo>git commit -m "welcome team commit"  
[master 9d89669] welcome team commit  
1 file changed, 1 insertion(+), 1 deletion(-)  
D:\TeamRepo>
```

7. Now switch back to your branch.

```
Git CMD
D:\>cd TeamRepo
D:\TeamRepo>git branch MyBranch
D:\TeamRepo>git branch
* MyBranch
  master

D:\TeamRepo>git commit -m "branch commit"
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

D:\TeamRepo>git checkout MyBranch
Switched to branch 'MyBranch'

D:\TeamRepo>git branch
* MyBranch
  master

D:\TeamRepo>git commit -m "once again"
On branch MyBranch
nothing to commit, working tree clean

D:\TeamRepo>git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

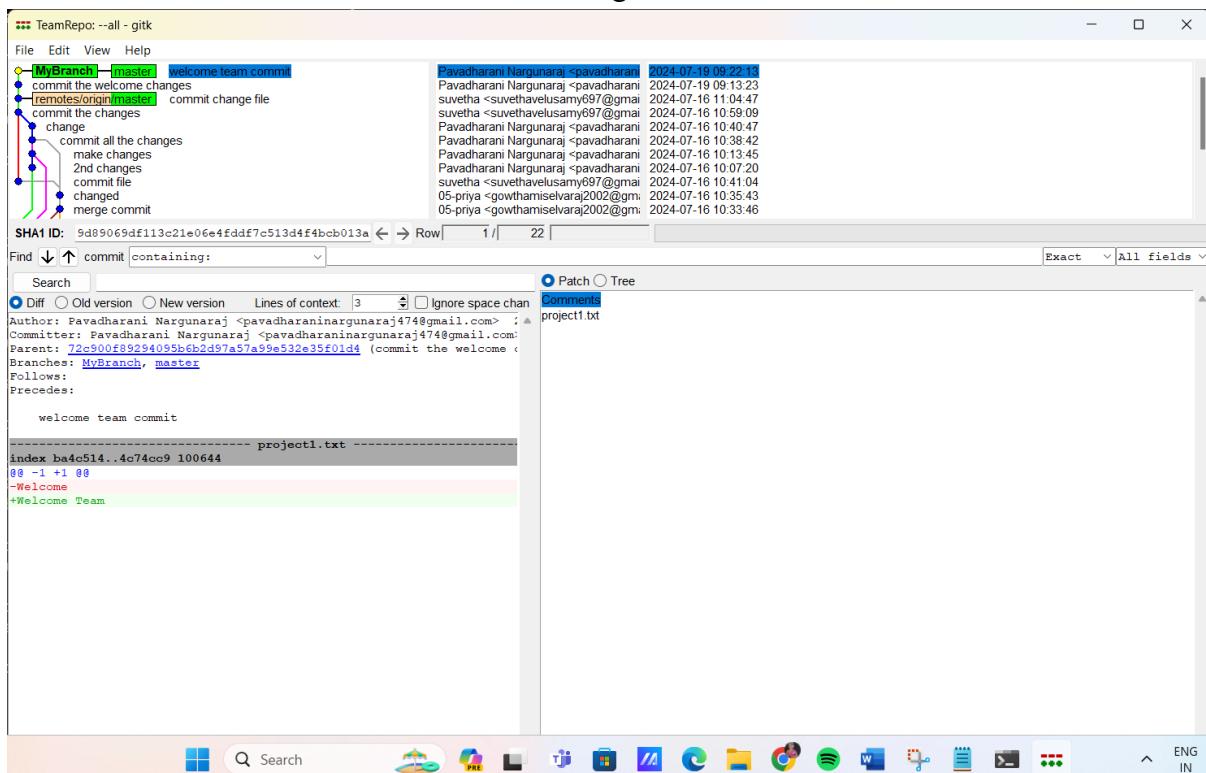
D:\TeamRepo>git add .

D:\TeamRepo>git commit -m "welcome team commit"
[master 9d89069] welcome team commit
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\TeamRepo>git checkout MyBranch
Switched to branch 'MyBranch'

D:\TeamRepo>
```

8. Use the **rebase** command in your branch. Look at the DAG in **gitk**, and note that you have the commit from the master branch, but no merge commit.



9. Make one more commit in your branch.

```
Git CMD + - X
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

D:\TeamRepo>git add .

D:\TeamRepo>git commit -m "welcome team commit"
[master 9d89069] welcome team commit
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\TeamRepo>git checkout MyBranch
Switched to branch 'MyBranch'

D:\TeamRepo>git rebase master
Successfully rebased and updated refs/heads/MyBranch.

D:\TeamRepo>gitk --all

D:\TeamRepo>git add .

D:\TeamRepo>git commit -m "commit Team Work"
[MyBranch ec2a92f] commit Team Work
 1 file changed, 1 insertion(+)

D:\TeamRepo>git checkout MyBranch
Already on 'MyBranch'

D:\TeamRepo>git status
On branch MyBranch
nothing to commit, working tree clean

D:\TeamRepo>git add .

D:\TeamRepo>git commit -m"one more commit"
[MyBranch 21fc76] one more commit
 1 file changed, 1 insertion(+)

D:\TeamRepo>
```

10. Return to master. Merge your branch. Notice how, thanks to the rebase, this is a fast-forward merge.

```
Git CMD + - X
Switched to branch 'MyBranch'

D:\TeamRepo>git rebase master
Successfully rebased and updated refs/heads/MyBranch.

D:\TeamRepo>gitk --all

D:\TeamRepo>git add .

D:\TeamRepo>git commit -m "commit Team Work"
[MyBranch ec2a92f] commit Team Work
 1 file changed, 1 insertion(+)

D:\TeamRepo>git checkout MyBranch
Already on 'MyBranch'

D:\TeamRepo>git status
On branch MyBranch
nothing to commit, working tree clean

D:\TeamRepo>git add .

D:\TeamRepo>git commit -m"one more commit"
[MyBranch 21fc76] one more commit
 1 file changed, 1 insertion(+)

D:\TeamRepo>git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

D:\TeamRepo>git merge MyBranch
Updating 9d89069..21fc76
Fast-forward
  project1.txt | 2 ++
  1 file changed, 2 insertions(+)

D:\TeamRepo>
```

Stretch Task

1. Find somebody from your team from the previous exercise. Have them **push** a commit to the central repository.

A screenshot of a Windows desktop. In the center is a terminal window titled "Git CMD" showing a sequence of git commands and their outputs. The commands include cloning a repository, initializing a local repository, pushing changes, pulling changes, committing, and pushing again. A "hint" message about a merge conflict is visible. The taskbar at the bottom shows various icons for applications like File Explorer, Edge, and Mail. The system tray indicates it's 10:20 AM on 7/19/2024, with a battery level of 83% and a mostly cloudy weather icon.

```
D:\TeamRepo>git clone "https://github.com/Suvetha-2003/TeamRepo.git"
fatal: destination path 'TeamRepo' already exists and is not an empty directory.

D:\TeamRepo>git init
Reinitialized existing Git repository in D:/TeamRepo/.git/

D:\TeamRepo>git clone "https://github.com/Suvetha-2003/TeamRepo.git"
fatal: destination path 'TeamRepo' already exists and is not an empty directory.

D:\TeamRepo>git push -u origin master
To https://github.com/Suvetha-2003/TeamRepo.git
 ! [rejected]      master --> master (non-fast-forward)
error: failed to push some refs to 'https://github.com/Suvetha-2003/TeamRepo.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. If you want to integrate the remote changes,
hint: use 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

D:\TeamRepo>git pull origin master
error: You have not concluded your merge (MERGE_HEAD exists).
hint: Please, commit your changes before merging.
fatal: Exiting because of unfinished merge.

D:\TeamRepo>git commit -m "committing"
[master 7b13c6f] committing

D:\TeamRepo>git push -u origin master
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 4 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (14/14), 1.33 KiB | 682.00 KiB/s, done.
Total 14 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), done.
To https://github.com/Suvetha-2003/TeamRepo.git
 8e6eaef..7b13c6f master --> master
branch 'master' set up to track 'origin/master'.

D:\TeamRepo>
```

2. Make a commit locally yourself also. Note that you should not have pulled their commit at this point.

A screenshot of a Windows desktop. In the center is a terminal window titled "Git CMD" showing a sequence of git commands and their outputs. The commands include committing changes, pulling from the remote repository, and attempting to merge. A "CONFLICT (submodule)" message is displayed, indicating a merge conflict between the local submodule and the remote submodule. The taskbar at the bottom shows various icons for applications like File Explorer, Edge, and Mail. The system tray indicates it's 10:20 AM on 7/19/2024, with a battery level of 83% and a mostly cloudy weather icon.

```
hint: git rm --cached TeamRepo
hint:
hint: See "git help submodule" for more information.
hint: Disable this message with "git config advice.addEmbeddedRepo false"

D:\TeamRepo>git commit -m "made change"
[master 030e120] made change
 1 file changed, 1 insertion(+)
 create mode 160000 TeamRepo

D:\TeamRepo>git pull origin master
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 29 (delta 10), reused 29 (delta 10), pack-reused 0
Unpacking objects: 100% (29/29), 2.67 KiB | 20.00 KiB/s, done.
From https://github.com/Suvetha-2003/TeamRepo
 * branch      master      -> FETCH_HEAD
   8e6eaef..2913f83  master      -> origin/master
Failed to merge submodule TeamRepo (no merge base)
CONFLICT (submodule): Merge conflict in TeamRepo
hint: Recursive merging with submodules currently only supports trivial cases.
hint: Please manually handle the merging of each conflicted submodule.
hint: This can be accomplished with the following steps:
hint: - go to submodule (TeamRepo), and either merge commit 7b13c6f
hint: or update to an existing commit which has merged those changes
hint: - come back to superproject and run:
hint:     git add TeamRepo
hint:
hint:     to record the above merge or update
hint: - resolve any other conflicts in the superproject
hint: - commit the resulting index in the superproject
hint:
hint: Disable this message with "git config advice.submoduleMergeConflict false"
Automatic merge failed; fix conflicts and then commit the result.

D:\TeamRepo>
```

3. Try to **push**, and watch it fail.

```
D:\TeamRepo>git pull origin master
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 29 (delta 10), reused 29 (delta 10), pack-reused 0
Unpacking objects: 100% (29/29), 2.67 KiB | 20.00 KiB/s, done.
From https://github.com/Suvetha-2003/TeamRepo
 * branch            master      -> FETCH_HEAD
   8e6eaef..2913f83  master      -> origin/master
Failed to merge submodule TeamRepo (no merge base)
CONFLICT (submodule): Merge conflict in TeamRepo
hint: Recursive merging with submodules currently only supports trivial cases.
hint: Please manually handle the merging of each conflicted submodule.
hint: This can be accomplished with the following steps:
hint: - go to submodule (TeamRepo), and either merge commit 7b13c6f
hint: or update to an existing commit which has merged those changes
hint: - come back to superproject and run:
hint:
hint:     git add TeamRepo
hint:
hint:     to record the above merge or update
hint: - resolve any other conflicts in the superproject
hint: - commit the resulting index in the superproject
hint:
hint: Disable this message with "git config advice.submoduleMergeConflict false"
Automatic merge failed; fix conflicts and then commit the result.

D:\TeamRepo>git push -u origin master
To https://github.com/Suvetha-2003/TeamRepo.git
! [rejected]        master -> master (non-fast-forward)
error: failed to push some refs to 'https://github.com/Suvetha-2003/TeamRepo.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. If you want to integrate the remote changes,
hint: use 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

D:\TeamRepo>
```

4. Now, **pull** but using the **--rebase** flag.

```
No commands remaining.
You are currently rebasing branch 'master' on '2913f83'.
  (all conflicts fixed: run "git rebase --continue")

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   TeamRepo

D:\TeamRepo>git checkout master
Warning: you are leaving 4 commits behind, not connected to
any of your branches:

  b08ee2f one more commit
  f219eb3 commit Team Work
  4f1fa27d welcome team commit
  75ff1f1 commit the welcome changes

If you want to keep them by creating a new branch, this may be a good time
to do so with:

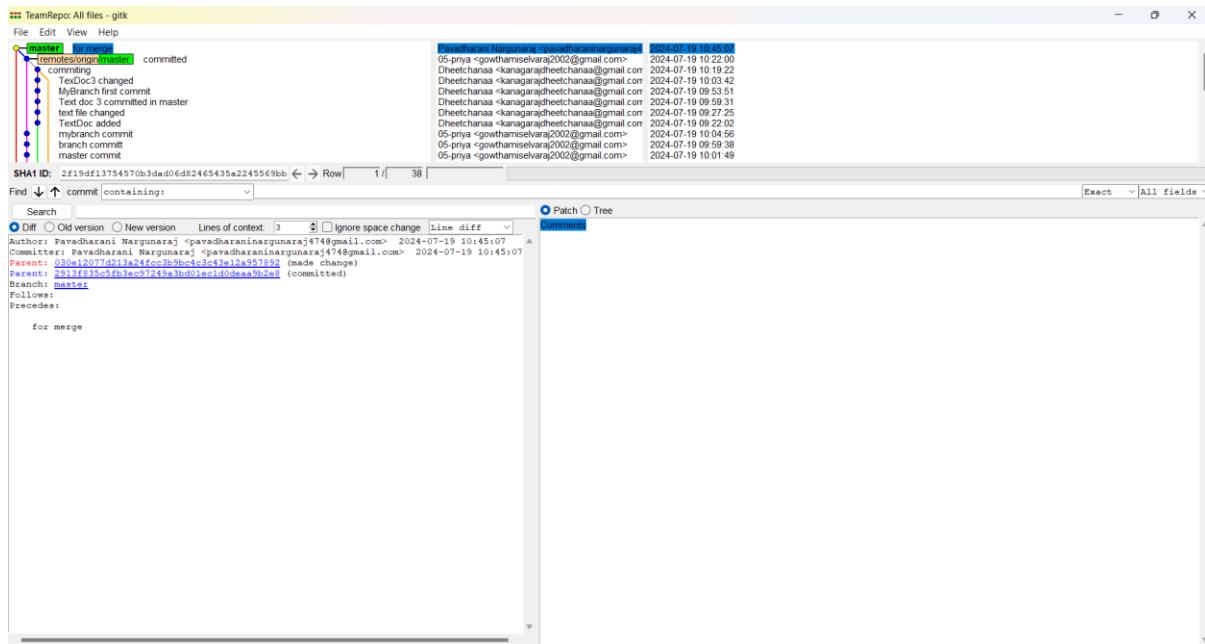
  git branch <new-branch-name> b08ee2f

Switched to branch 'master'
Your branch is ahead of 'origin/master' by 6 commits.
  (use "git push" to publish your local commits)

D:\TeamRepo>git pull --rebase origin master
From https://github.com/Suvetha-2003/TeamRepo
 * branch            master      -> FETCH_HEAD
fatal: It seems that there is already a rebase-merge directory, and
I wonder if you are in the middle of another rebase. If that is the
case, please try
  git rebase (--continue | --abort | --skip)
If that is not the case, please
  rm -fr ".git/rebase-merge"
and run me again. I am stopping in case you still have something
valuable there.

D:\TeamRepo>
```

5. Use **git log** and **gitk** to verify that there is no merge commit, and the DAG is linear.



The screenshot shows the gitk application window titled "TeamRepo: All files - gitk". The main pane displays a linear history of commits from a local branch named "remotes/origin/master". The commits are listed in chronological order from top to bottom:

- committing (author: Dheetchanaa <kanagarajdheetchanaa@gmail.com>, date: 2024-07-19 10:19:22)
- TexDoc3 changed (author: Dheetchanaa <kanagarajdheetchanaa@gmail.com>, date: 2024-07-19 10:03:42)
- Text doc 3 committed (author: Dheetchanaa <kanagarajdheetchanaa@gmail.com>, date: 2024-07-19 09:59:51)
- Text file changed (author: Dheetchanaa <kanagarajdheetchanaa@gmail.com>, date: 2024-07-19 09:59:31)
- TexDoc added (author: Dheetchanaa <kanagarajdheetchanaa@gmail.com>, date: 2024-07-19 09:27:25)
- branch commit (author: Dheetchanaa <kanagarajdheetchanaa@gmail.com>, date: 2024-07-19 09:22:02)
- branch commit (author: Dheetchanaa <kanagarajdheetchanaa@gmail.com>, date: 2024-07-19 09:19:59)
- master commit (author: Dheetchanaa <kanagarajdheetchanaa@gmail.com>, date: 2024-07-19 09:19:49)

The commit "committing" is highlighted with a blue selection bar. The bottom right corner of the gitk window has a small icon of a person with a question mark.

6. Notice that your commit is the latest one, even though temporally the other member of your team made their commit afterwards. Why is this?

When you pull with the '--rebase' flag, Git fetches the latest changes from the remote repository and then re-applies your local commits on top of these changes. This can make it seem like your commits were made after the remote commits, even if they were made before, because the rebase process effectively rewrites the commit history.