

SOURCE REPOSITORY ASSIGNMENT-2

NAME : S UVETHA V

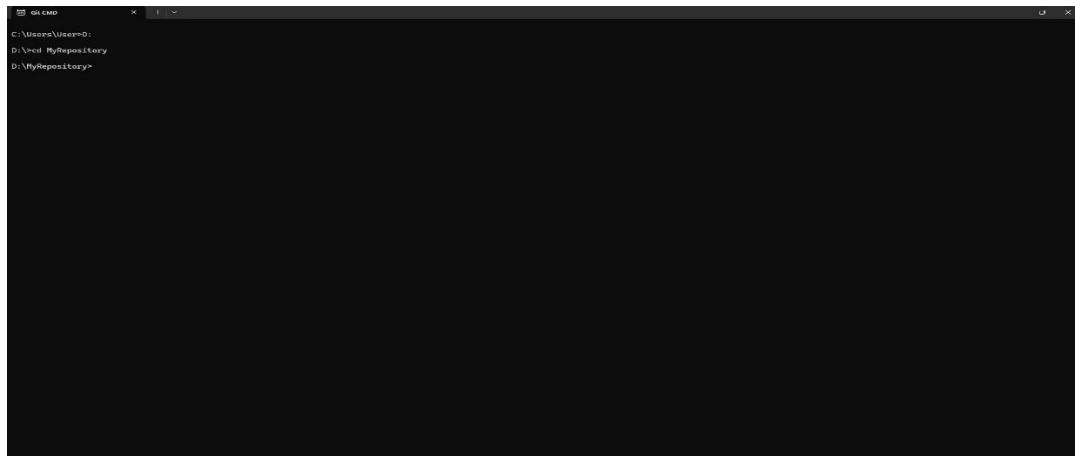
BRANCH : COMPUTER SCIENCE AND ENGINEERING

REG NO : 73772114218

EMAIL : suvethavelusamy697@gmail.com

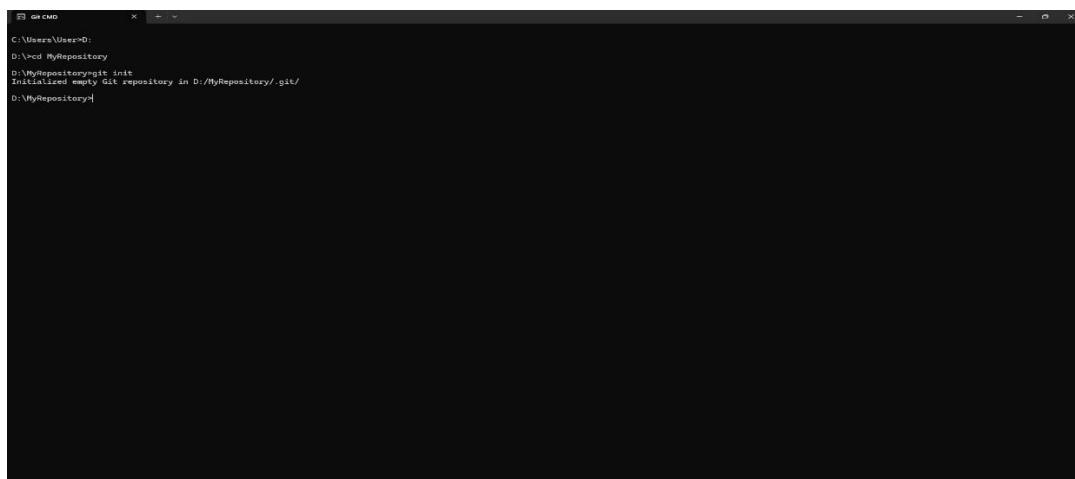
Exercise 1:

1. Create a new directory and change into it.



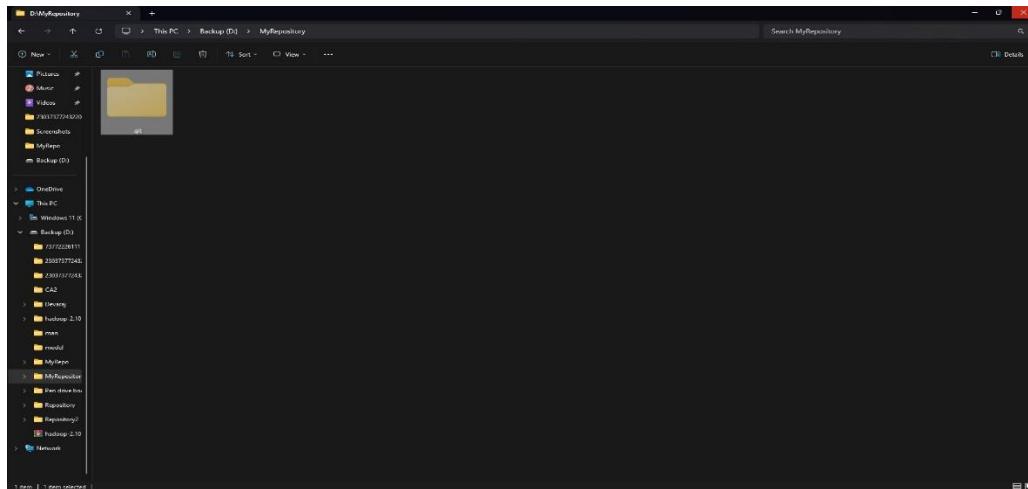
```
C:\Users\User>D:  
D:\>cd MyRepository  
D:\MyRepository>
```

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

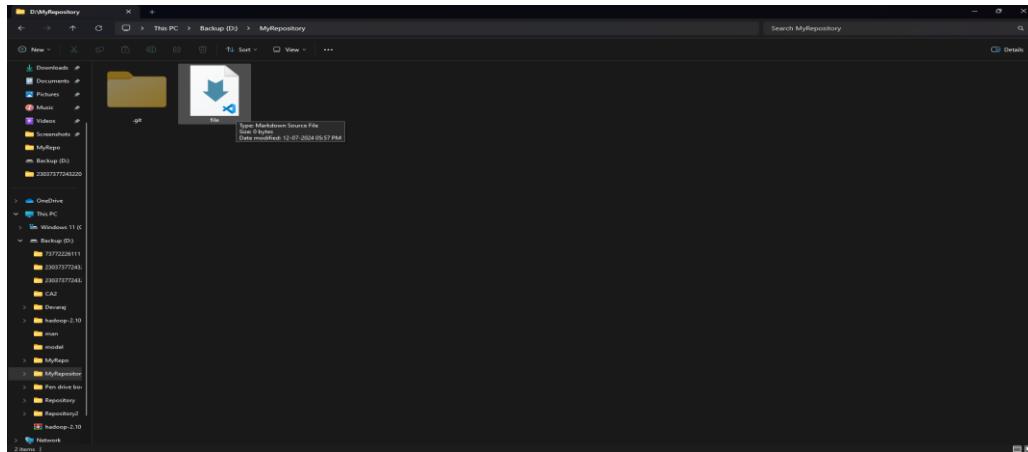


```
C:\Users\User>D:  
D:\>cd MyRepository  
D:\MyRepository>git init  
Initialized empty Git repository in D:/MyRepository/.git/  
D:\MyRepository>
```

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.

```
Git CMD x + v

C:\Users\User>D:
D:\>cd MyRepository
D:\MyRepository>git init
Initialized empty Git repository in D:/MyRepository/.git/
D:\MyRepository>git status
On branch master

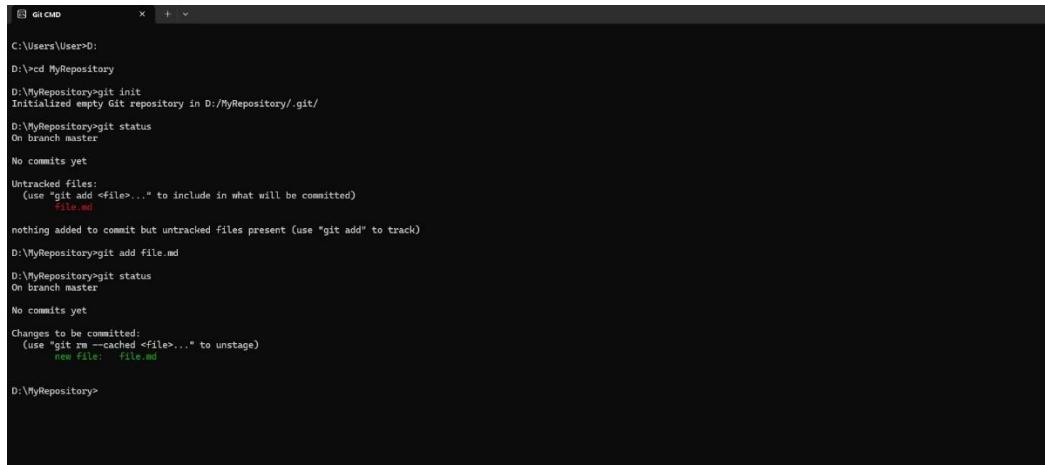
No commits yet

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

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

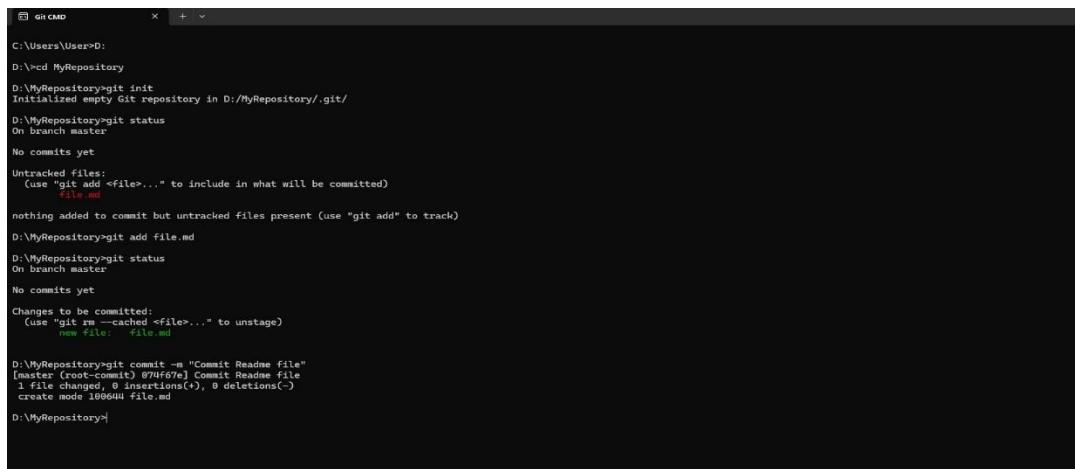
D:\MyRepository>
```

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



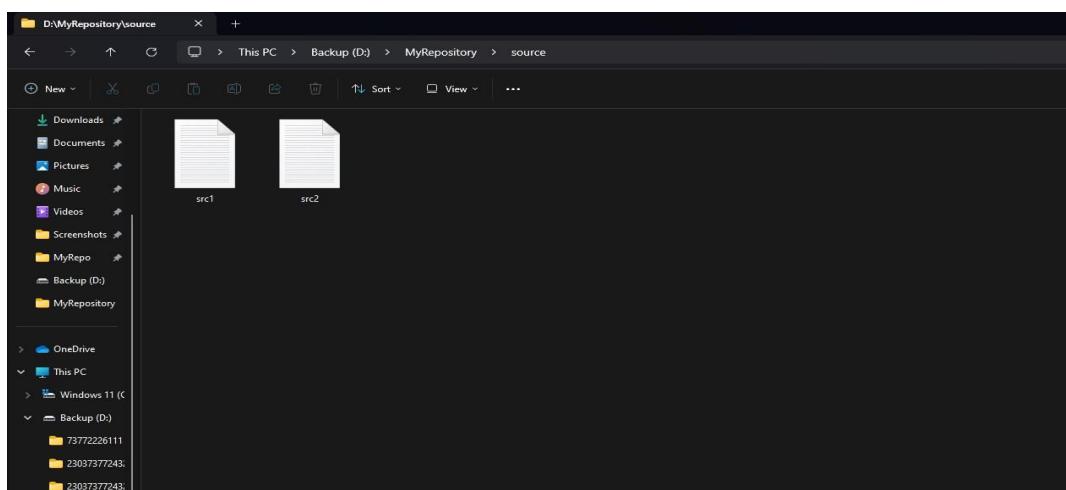
```
git CMD x + v
C:\Users\User>D:
D:>cd MyRepository
D:\MyRepository>git init
Initialized empty Git repository in D:/MyRepository/.git/
D:\MyRepository>git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file.md
nothing added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

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



```
git CMD x + v
C:\Users\User>D:
D:>cd MyRepository
D:\MyRepository>git init
Initialized empty Git repository in D:/MyRepository/.git/
D:\MyRepository>git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file.md
nothing added to commit but untracked files present (use "git add" to track)
D:\MyRepository>git add file.md
D:\MyRepository>git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   file.md
D:\MyRepository>git commit -m "Commit Readme file"
[master (root-commit) 074f67e] Commit Readme file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file.md
D:\MyRepository>
```

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.

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

D:\MyRepository>git commit -m "Commit Readme file"
[master (root-commit) 07af67e] Commit Readme file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file.md
D:\MyRepository>git add source
D:\MyRepository>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   source/src1.txt
    new file:   source/src2.txt

D:\MyRepository>git commit -m "Commit source files"
[master a13ee67] Commit source files
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 source/src1.txt
 create mode 100644 source/src2.txt
D:\MyRepository>
```

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

```
gitCMD X + ▾
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file.md
nothing added to commit but untracked files present (use "git add" to track)
D:\MyRepository>git add file.md
D:\MyRepository>git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   file.md

D:\MyRepository>git commit -m "Commit Readme file"
[master (root-commit) 07af67e] Commit Readme file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file.md
D:\MyRepository>git add source
D:\MyRepository>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   source/src1.txt
    new file:   source/src2.txt

D:\MyRepository>git commit -m "Commit source files"
[master a13ee67] Commit source files
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 source/src1.txt
 create mode 100644 source/src2.txt
D:\MyRepository>git diff
diff --git a/source/src1.txt b/source/src1.txt
index e69de29..936cfd4 100644
--- a/source/src1.txt
+++ b/source/src1.txt
@@ -0,0 +1 @@
+suwetha
\ No newline at end of file
D:\MyRepository>
```

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?

```
gitCMD      x + ~
index e69de29..936cf1d 100644
--- /source/src1.txt
+++ b/source/src1.txt
@@ -0,0 +1,6 @@
+suveetha
\ No newline at end of file
D:\MyRepository>git add src1.txt
fatal: pathspec 'src1.txt' did not match any files
D:\MyRepository>git add .
D:\MyRepository>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)

    (use "git restore --<file>..." to discard changes in working directory)
           modified: source/src1.txt

D:\MyRepository>git commit -m "Change Directory"
[master cfd7c47] Change Directory
 1 file changed, 1 insertion(+)

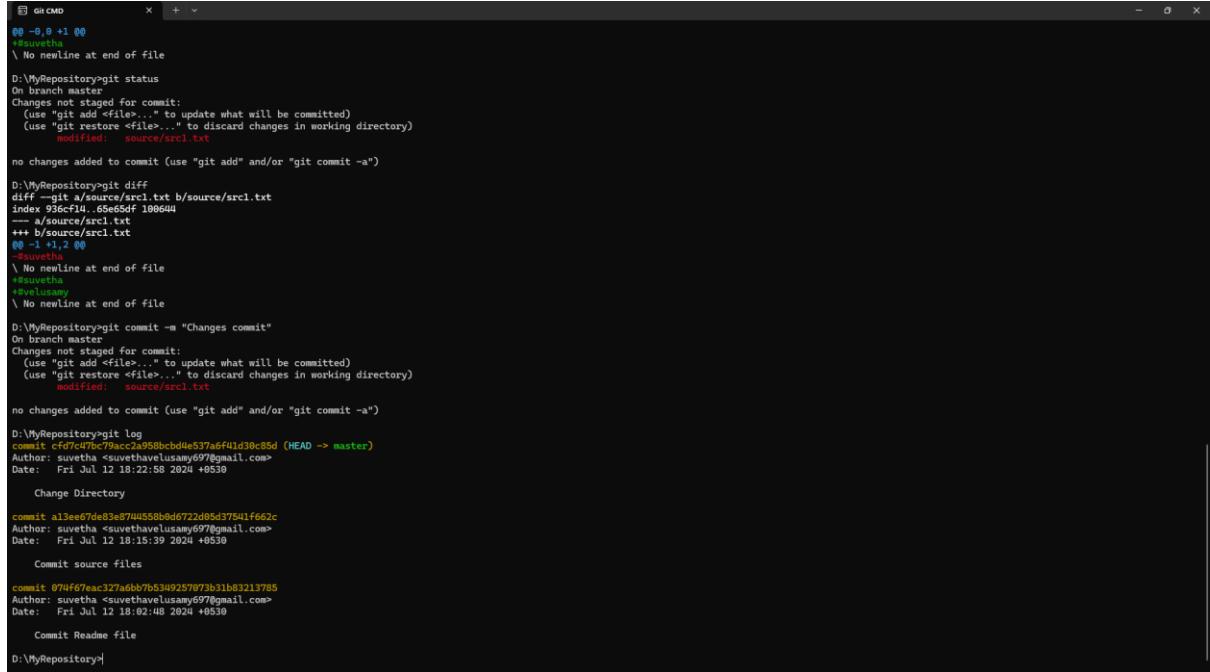
D:\MyRepository>git log
commit a13ee67de83e7f0458b8d6722d05d37541f662c (HEAD -> master)
Author: suveetha <suvethavelusamy69@gmail.com>
Date:  Fri Jul 12 18:22:58 2024 +0530
  Change Directory
commit a13ee67de83e7f0458b8d6722d05d37541f662c
Author: suveetha <suvethavelusamy69@gmail.com>
Date:  Fri Jul 12 18:22:58 2024 +0530
  Commit source file
commit 07ff6f7eac327adab7b53d0257973b31b83213785
Author: suveetha <suvethavelusamy69@gmail.com>
Date:  Fri Jul 12 18:02:48 2024 +0530
  Commit Readme file
Commit source file
D:\MyRepository>git diff a13ee67de83e7f0458b8d6722d05d37541f662c
diff --git a/source/src1.txt b/source/src1.txt
index e69de29..936cf1d 100644
--- /source/src1.txt
+++ b/source/src1.txt
@@ -0,0 +1,6 @@
+suveetha
\ No newline at end of file
D:\MyRepository>
```

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.

```
gitCMD      x + ~
commit a13ee67de83e7f0458b8d6722d05d37541f662c
Author: suveetha <suvethavelusamy69@gmail.com>
Date:  Fri Jul 12 18:15:39 2024 +0530
  Commit source file
commit 07ff6f7eac327adab7b53d0257973b31b83213785
Author: suveetha <suvethavelusamy69@gmail.com>
Date:  Fri Jul 12 18:02:48 2024 +0530
  Commit Readme file
Commit source file
D:\MyRepository>git diff a13ee67de83e7f0458b8d6722d05d37541f662c
diff --git a/source/src1.txt b/source/src1.txt
index e69de29..936cf1d 100644
--- /source/src1.txt
+++ b/source/src1.txt
@@ -0,0 +1,6 @@
+suveetha
\ No newline at end of file
D:\MyRepository>git status
On branch master
Changes to be committed:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
           modified: source/src1.txt
no changes added to commit (use "git add" and/or "git commit -a")

D:\MyRepository>git diff
diff --git a/source/src1.txt b/source/src1.txt
index 936cf14..65e65df 100644
--- /source/src1.txt
+++ b/source/src1.txt
@@ -1,2 +1,2 @@
-suveetha
+suveetha
+suvethavelusamy
\ No newline at end of file
D:\MyRepository>git commit -m "Changes commit"
On branch master
Changes to be committed:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
           modified: source/src1.txt
no changes added to commit (use "git add" and/or "git commit -a")
D:\MyRepository>
```

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



```
git CMD x + ~
@@ -0,0 +1 @@
+suveetha
\ No newline at end of file

D:\MyRepository>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: source/src1.txt

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

D:\MyRepository>git diff
diff --git a/source/src1.txt b/source/src1.txt
index 936cf14..65e65df 100644
--- a/source/src1.txt
+++ b/source/src1.txt
@@ -1,1 +1 @@
+suveetha
\ No newline at end of file
+suveetha
+suveetha
\ No newline at end of file

D:\MyRepository>git commit -m "Changes commit"
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: source/src1.txt

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

D:\MyRepository>git log
commit cf7c747c9ac2a958bcbd4e537a6f41d30c85d (HEAD -> master)
Author: suveetha <suveethavelusamy697@gmail.com>
Date:   Fri Jul 12 18:22:58 2024 +0530

  Change Directory

commit a13ee67de3e874a558b0d6722d05d37541f662c
Author: suveetha <suveethavelusamy697@gmail.com>
Date:   Fri Jul 12 18:15:39 2024 +0530

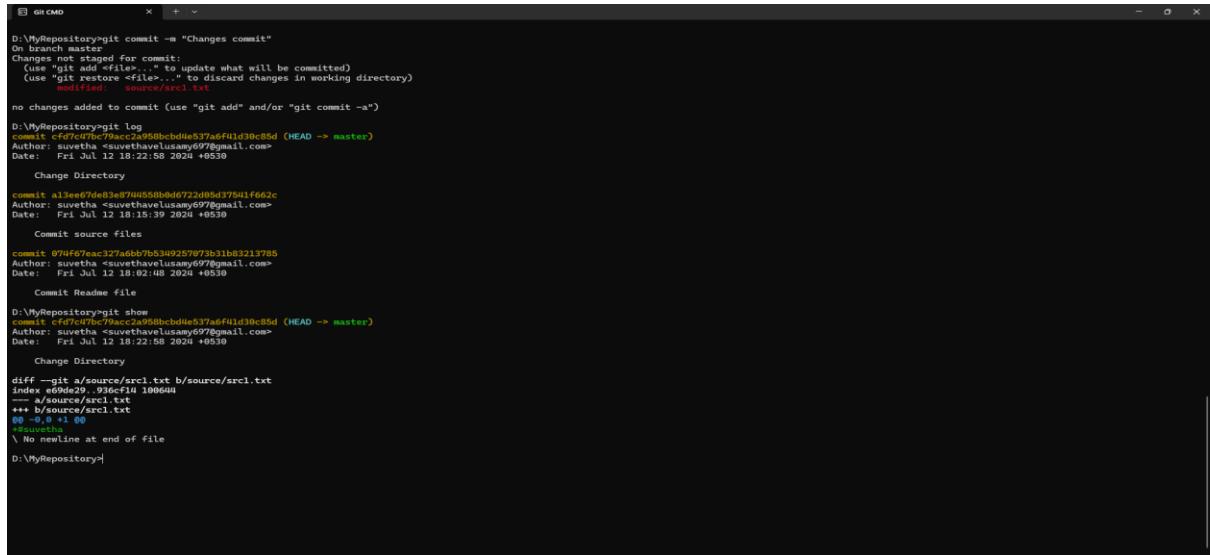
  Commit source files

commit 07af67eac327a6bb7b5349257073b31b83213785
Author: suveetha <suveethavelusamy697@gmail.com>
Date:   Fri Jul 12 18:02:48 2024 +0530

  Commit Readme file

D:\MyRepository|
```

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 + ~
D:\MyRepository>git commit -m "Changes commit"
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: source/src1.txt

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

D:\MyRepository>git log
commit cf7c747c9ac2a958bcbd4e537a6f41d30c85d (HEAD -> master)
Author: suveetha <suveethavelusamy697@gmail.com>
Date:   Fri Jul 12 18:22:58 2024 +0530

  Change Directory

commit a13ee67de3e874a558b0d6722d05d37541f662c
Author: suveetha <suveethavelusamy697@gmail.com>
Date:   Fri Jul 12 18:15:39 2024 +0530

  Commit source files

commit 07af67eac327a6bb7b5349257073b31b83213785
Author: suveetha <suveethavelusamy697@gmail.com>
Date:   Fri Jul 12 18:02:48 2024 +0530

  Commit Readme file

D:\MyRepository>git show
commit cf7c747c9ac2a958bcbd4e537a6f41d30c85d (HEAD -> master)
Author: suveetha <suveethavelusamy697@gmail.com>
Date:   Fri Jul 12 18:22:58 2024 +0530

  Change Directory

diff -git a/source/src1.txt b/source/src1.txt
index 936cf14..65e65df 100644
--- a/source/src1.txt
+++ b/source/src1.txt
@@ -1,1 +1 @@
+suveetha
\ No newline at end of file
D:\MyRepository|
```

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

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

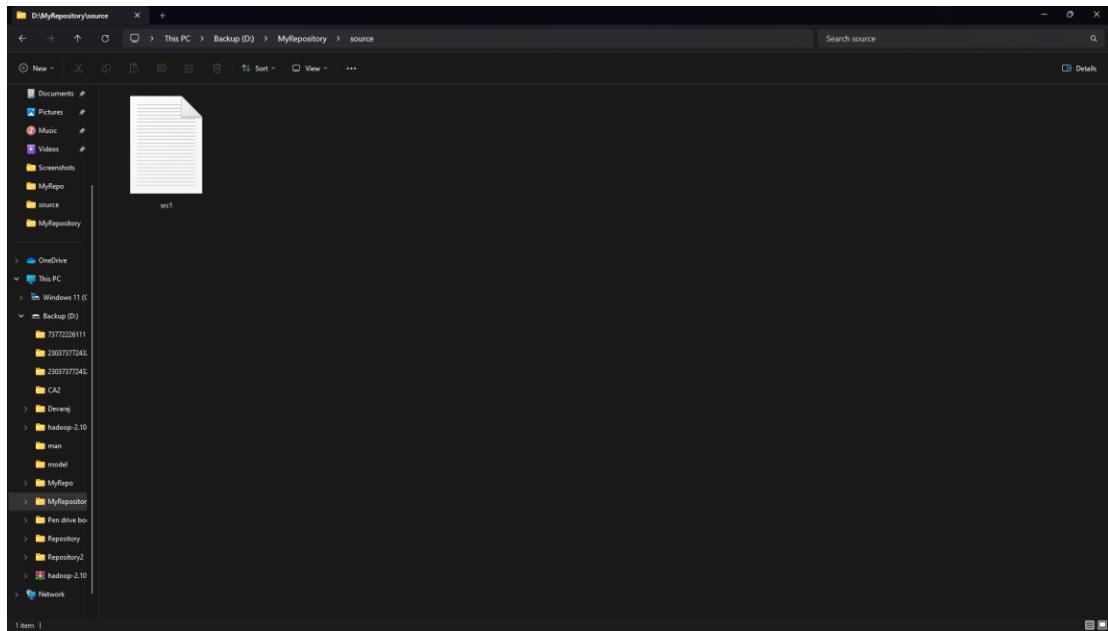
```
git CMD * + ~
Date: Fri Jul 12 18:22:58 2024 +0530
Change Directory
commit a13ee67de83e8744550b9d6722d05d37541f662c
Author: suvetha <suvehavevelusamy69@gmail.com>
Date: Fri Jul 12 18:15:39 2024 +0530
Commit source files
commit 07af67eadc327a6bb7b5349257073b31b83213785
Author: suvetha <suvehavevelusamy69@gmail.com>
Date: Fri Jul 12 18:02:48 2024 +0530
Commit Readme file
D:\MyRepository>git show
commit cf07c47c9ac2a958bc9d4e537a6f41d30c85d (HEAD -> master)
Author: suvetha <suvehavevelusamy69@gmail.com>
Date: Fri Jul 12 18:22:58 2024 +0530
Change Directory
diff --git a/source/src1.txt b/source/src1.txt
index e69de29..936cf14 100644
--- a/source/src1.txt
+++ b/source/src1.txt
@@ -0,0 +1 @@
+suvetha
\ No newline at end of file
D:\MyRepository>git commit -m "Commit First"
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: source/src1.txt
no changes added to commit (use "git add" and/or "git commit -a")
D:\MyRepository>git add .
D:\MyRepository>git commit -m "Commit First"
[master 46d956d] Commit First
1 file changed, 2 insertions(+), 1 deletion(-)
D:\MyRepository>git add .
D:\MyRepository>git commit -m "commit the last changes"
[master d16a672] commit the last changes
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file.txt
D:\MyRepository>
```

Stretch Task:

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

```
git CMD * + ~
D:\MyRepository>git show
commit cf07c47c9ac2a958bc9d4e537a6f41d30c85d (HEAD -> master)
Author: suvetha <suvehavevelusamy69@gmail.com>
Date: Fri Jul 12 18:22:58 2024 +0530
Change Directory
diff --git a/source/src1.txt b/source/src1.txt
index e69de29..936cf14 100644
--- a/source/src1.txt
+++ b/source/src1.txt
@@ -0,0 +1 @@
+suvetha
\ No newline at end of file
D:\MyRepository>git commit -m "Commit First"
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: source/src1.txt
no changes added to commit (use "git add" and/or "git commit -a")
D:\MyRepository>git add .
D:\MyRepository>git commit -m "Commit First"
[master 46d956d] Commit First
1 file changed, 2 insertions(+), 1 deletion(-)
D:\MyRepository>git add .
D:\MyRepository>git commit -m "commit the last changes"
[master d16a672] commit the last changes
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file.txt
D:\MyRepository>git rm file.txt
rm 'file.txt'
D:\MyRepository>git status
On branch master
Changes to be committed:
(use "git restore --staged <file>..." to unstage)
 deleted: file.txt
D:\MyRepository>git commit -m "Delete the commit"
[master f8624f1] Delete the commit
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 file.txt
D:\MyRepository>
```

- 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.**



- 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\User>D:
D:\>cd MyRepository
D:\MyRepository>cd source
D:\MyRepository\source>del src2.txt
D:\MyRepository\source>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:   src2.txt

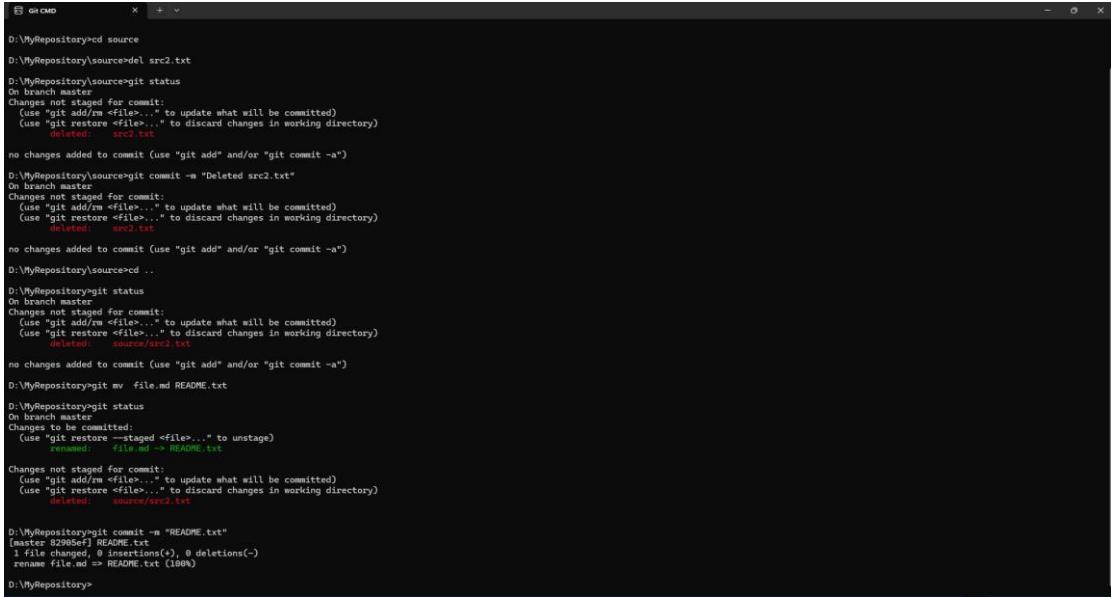
no changes added to commit (use "git add" and/or "git commit -a")

D:\MyRepository\source>git commit -m "Deleted src2.txt"
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:   src2.txt

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

D:\MyRepository\source>
```

- 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.**



```

D:\MyRepository>cd source
D:\MyRepository\source>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:   src2.txt

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

D:\MyRepository\source>git commit -m "Deleted src2.txt"
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:   src2.txt

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

D:\MyRepository\source>cd ..
D:\MyRepository>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:   source/src2.txt

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

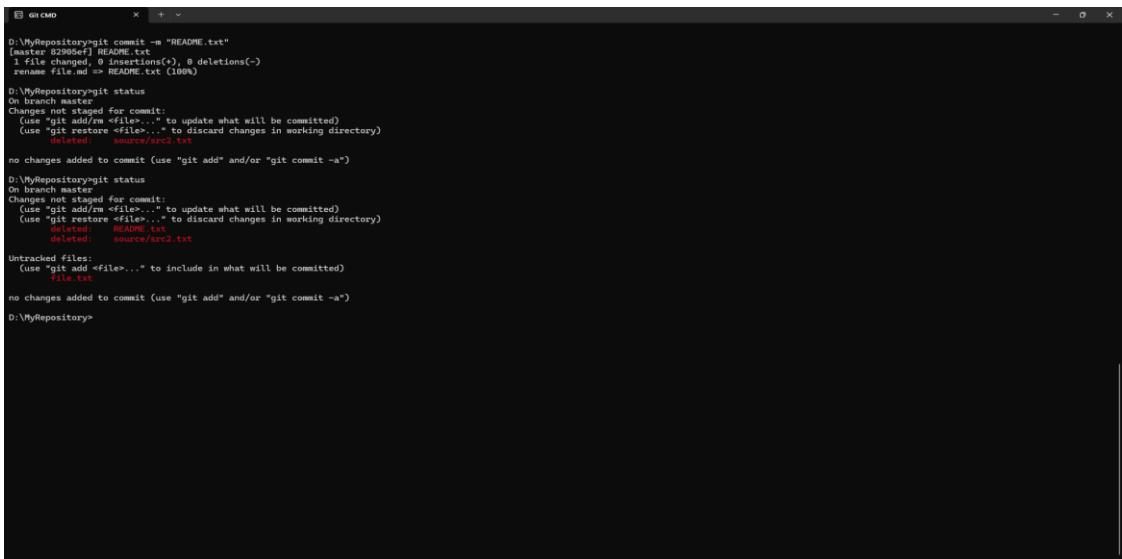
D:\MyRepository>git mv file.md README.txt
D:\MyRepository>git status
On branch master
Changes to be committed:
  (use "git add/rm <file>..." to update what will be committed)
    renamed:   file.md --> README.txt

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:   source/src2.txt

D:\MyRepository>git commit -m "README.txt"
[master b2905ef] README.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename file.md => README.txt (100%)
D:\MyRepository>

```

- 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:\MyRepository>git commit -m "README.txt"
[master b2905ef] README.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename file.md => README.txt (100%)
D:\MyRepository>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:   source/src2.txt

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

D:\MyRepository>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:   source/src2.txt

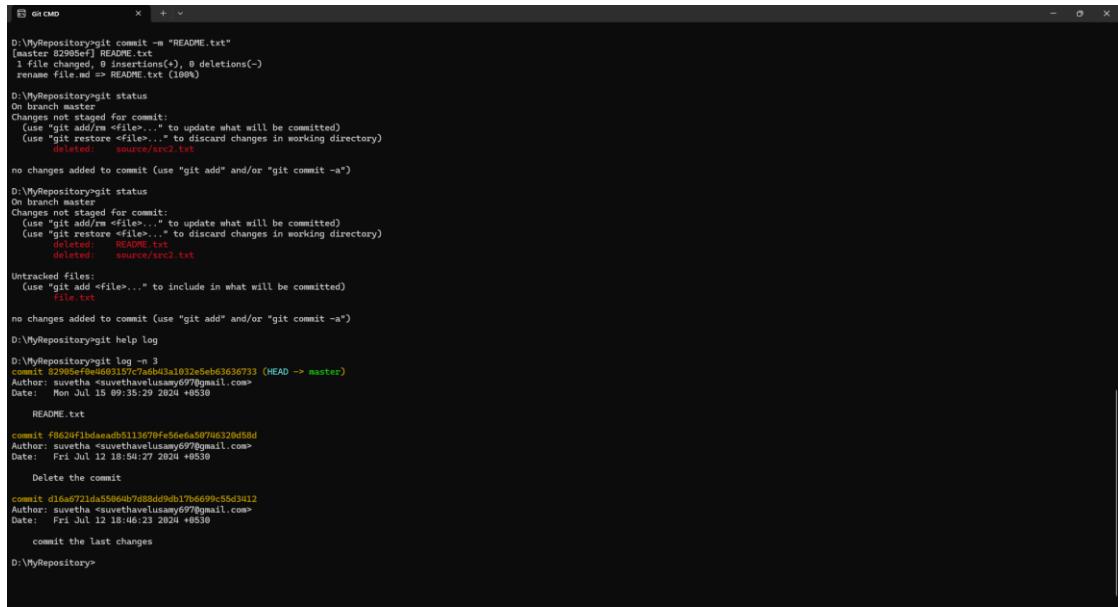
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file.txt

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

D:\MyRepository>

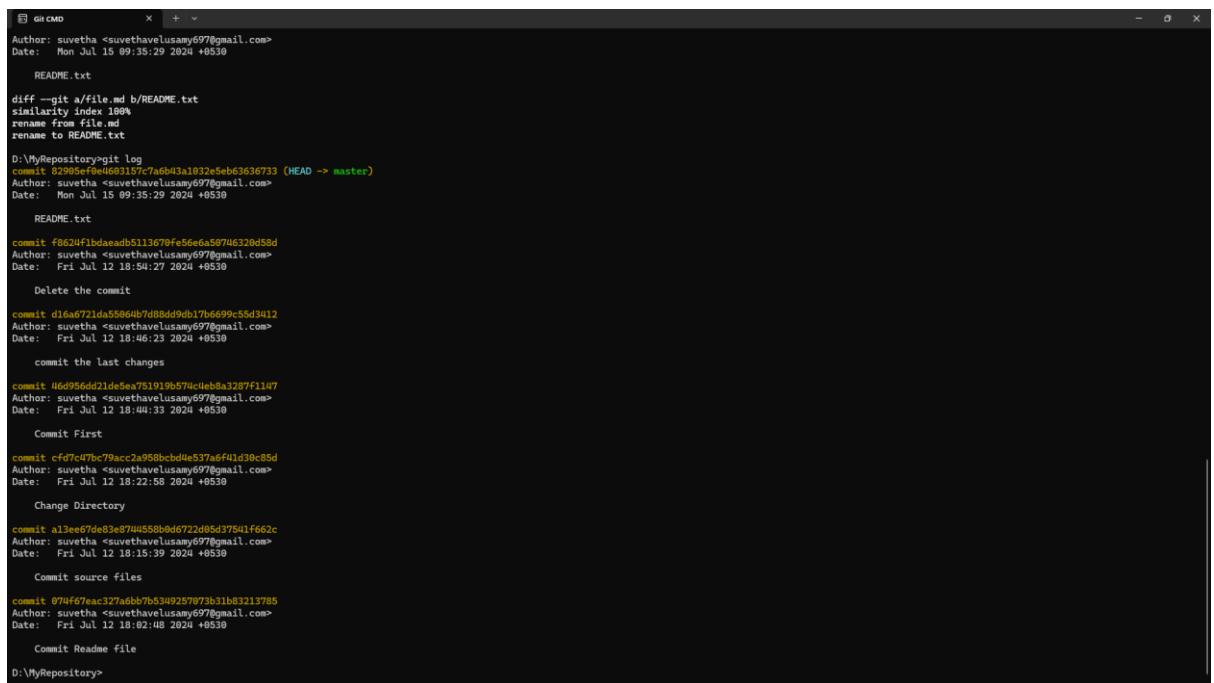
```

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



```
D:\MyRepository>git commit -m "README.txt"
[master 629505f] README.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename file.ad => README.txt (100%)
D:\MyRepository>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:  source/src2.txt
no changes added to commit (use "git add" and/or "git commit -a")
D:\MyRepository>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:  source/src2.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:\MyRepository>git help log
D:\MyRepository>git log --oneline
commit f8624fbdaeadb5113670fe56e6a50746320d58d
Author: suvetha <suvethavelusamy697@gmail.com>
Date:  Mon Jul 15 09:35:29 2024 +0530
  README.txt
commit f16a0721da55964b7d88dd9db17b6699c55d3412
Author: suvetha <suvethavelusamy697@gmail.com>
Date:  Fri Jul 12 18:46:23 2024 +0530
  commit the last changes
D:\MyRepository>
```

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?



```
D:\MyRepository>git log --stat
Author: suvetha <suvethavelusamy697@gmail.com>
Date:  Mon Jul 15 09:35:29 2024 +0530
  README.txt
diff --git a/file.ad b/README.txt
similarity index 100%
rename from file.ad
rename to README.txt
D:\MyRepository>git log
commit 629505fde4608157c7a6b43a1032e5eb63636733 (HEAD -> master)
Author: suvetha <suvethavelusamy697@gmail.com>
Date:  Mon Jul 15 09:35:29 2024 +0530
  README.txt
commit f8624fbdaeadb5113670fe56e6a50746320d58d
Author: suvetha <suvethavelusamy697@gmail.com>
Date:  Fri Jul 12 18:54:27 2024 +0530
  Delete the commit
commit d16a0721da55964b7d88dd9db17b6699c55d3412
Author: suvetha <suvethavelusamy697@gmail.com>
Date:  Fri Jul 12 18:46:23 2024 +0530
  commit the last changes
commit f16a0721da55964b7d88dd9db17b6699c55d3412
Author: suvetha <suvethavelusamy697@gmail.com>
Date:  Fri Jul 12 18:46:23 2024 +0530
  Commit First
commit cfd7c474fc79aacc2a958bc0d4e537a6f4fd30c85d
Author: suvetha <suvethavelusamy697@gmail.com>
Date:  Fri Jul 12 18:22:58 2024 +0530
  Change Directory
commit a13ee670e83e7440558bb0d6722d05d37501f662c
Author: suvetha <suvethavelusamy697@gmail.com>
Date:  Fri Jul 12 18:15:39 2024 +0530
  Commit source files
commit 970f67eac327adbb7b53409257073b31b43213785
Author: suvetha <suvethavelusamy697@gmail.com>
Date:  Fri Jul 12 18:02:48 2024 +0530
  Commit Readme file
D:\MyRepository>
```

- 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 log
commit d92e24f1bdaedb5113670fe56e6a50746320d58d
Author: suvetha <suvethavelusamy697@gmail.com>
Date:   Fri Jul 12 18:38:29 2024 +0530

    README.txt

commit 0f624f1bdaedb5113670fe56e6a50746320d58d
Author: suvetha <suvethavelusamy697@gmail.com>
Date:   Fri Jul 12 18:38:27 2024 +0530

Delete the commit

commit 0f624f1bdaedb5113670fe56e6a50746320d58d
Author: suvetha <suvethavelusamy697@gmail.com>
Date:   Fri Jul 12 18:46:23 2024 +0530

    commit the last changes

commit 46d956dd21de5e751919b7c1eb0a3287f1147
Author: suvetha <suvethavelusamy697@gmail.com>
Date:   Fri Jul 12 18:46:33 2024 +0530

Commit First

commit cfdf7c7b79acc2a95864b7d88dd9db17bd699c55d3412
Author: suvetha <suvethavelusamy697@gmail.com>
Date:   Fri Jul 12 18:46:23 2024 +0530

    Commit Directory

commit a1e07d4ed74974115588db07248d7781f662c
Author: suvetha <suvethavelusamy697@gmail.com>
Date:   Fri Jul 12 18:15:39 2024 +0530

    Commit source files

commit 07ff57ee227a0d9b830095777b51b83213785
Author: suvetha <suvethavelusamy697@gmail.com>
Date:   Fri Jul 12 18:02:48 2024 +0530

    Commit Readme file

D:\MyRepository>git diff 82905ef0e4603157c7a6b43a1032e5eb63636733..f8624f1bdaedb5113670fe56e6a50746320d58d
diff --git a/README.txt b/README.txt
similarity index 100%
rename from README.txt
rename to README.txt
D:\MyRepository>

```

Exercise 2:

Main Task:

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

```

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:    source/src2.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:\MyRepository>

```

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

```
git commit d16a6721da55064b7d8dd49eb17b6699-55d3412
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:46:23 2024 +0530

commit the last changes

git commit 46d956dd21de5e751919b570cdeb9a3287f1107
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:44:33 2024 +0530

Commit First

git commit fcf7c7b2c79acc2a958bcbd4e537a6f01d30c85d
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:22:58 2024 +0530

Change Directory

git commit a13ee07de38e8744558b0d0722d85d37541f662c
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:18:39 2024 +0530

Commit source files

git commit 8746f7eac327a6bb7b534925797b31b83213785
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:02:48 2024 +0530

Commit Readme file

D:\MyRepository>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: source/src2.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:\MyRepository>git branch MyBranch
D:\MyRepository>
```

3. Use the checkout command to switch to it.

```
git commit d16a6721da55064b7d8dd49eb17b6699-55d3412
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:46:23 2024 +0530

commit the last changes

git commit 46d956dd21de5e751919b570cdeb9a3287f1107
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:44:33 2024 +0530

Commit First

git commit fcf7c7b2c79acc2a958bcbd4e537a6f01d30c85d
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:22:58 2024 +0530

Change Directory

git commit a13ee07de38e8744558b0d0722d85d37541f662c
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:18:39 2024 +0530

Commit source files

git commit 8746f7eac327a6bb7b534925797b31b83213785
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:02:48 2024 +0530

Commit Readme file

D:\MyRepository>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: source/src2.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:\MyRepository>git branch MyBranch
D:\MyRepository>git checkout MyBranch
D: README.txt
D: source/src2.txt
Switched to branch 'MyBranch'

D:\MyRepository>
```

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

```
git CMD -> 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:\MyRepository>git branch MyBranch
D:\MyRepository>git checkout MyBranch
D:\MyRepository> README.txt
D:\MyRepository> source/src2.txt
Switched to branch 'MyBranch'

D:\MyRepository>git add Myrepo.txt
fatal: pathspec 'Myrepo.txt' did not match any files

D:\MyRepository>cd source

D:\MyRepository\source>git add Myrepo
fatal: pathspec 'Myrepo' did not match any files

D:\MyRepository\source>git commit -m "commit Myrepo"
On branch MyBranch
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:  src2.txt

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

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

D:\MyRepository\source>git add Myrepo.txt

D:\MyRepository\source>git commit -m "commit Myrepo"
[MyBranch 3b7eca7] commit Myrepo
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 106664 source/Myrepo.txt

D:\MyRepository\source>git mv file.txt README.txt
fatal: bad source, source=source/file.txt, destination=source/README.txt

D:\MyRepository\source>git mv Myrepo.txt README.txt

D:\MyRepository\source>git commit -m "commit README.txt"
[MyBranch 3b7eca7] commit README.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename source/Myrepo.txt => README.txt {100%}

D:\MyRepository\source>
```

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:\Users\User\AppData\Local\Temp\git-clone-1\MyRepository>git mv Myrepo.txt README.txt
D:\MyRepository\source>git commit -m "commit README.txt"
[MyBranch 3b7eca7] commit README.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename source/Myrepo.txt => README.txt {100%}

D:\MyRepository\source>git log
commit 3b7eca7f0e6c4ff40d5300152d0f01bb0fec2586 (HEAD --> MyBranch)
Author: suvetha <suvethavelusamy69@gmail.com>
Date:   Mon Jul 15 18:24:58 2024 +0530

    commit README.txt

commit e59045f9824e1dd8bc039e4812ed116224c277ce
Author: suvetha <suvethavelusamy69@gmail.com>
Date:   Mon Jul 15 18:22:52 2024 +0530

        commit Myrepo

commit 82905ef04603157c7a7dbf03a1073e5ab63636733 (master)
Author: suvetha <suvethavelusamy69@gmail.com>
Date:   Mon Jul 15 09:35:29 2024 +0530

        README.txt

commit f8624f1bdbeadb5113670fe56e6a50746320d58d
Author: suvetha <suvethavelusamy69@gmail.com>
Date:   Fri Jul 12 18:54:27 2024 +0530

        Delete the commit

commit d16a6721da55966b7d80dd9dh17b6699c55d3412
Author: suvetha <suvethavelusamy69@gmail.com>
Date:   Fri Jul 12 18:46:23 2024 +0530

        commit the last changes

commit 44d996dd21de5ea7519196574c4ebba3237f1107
Author: suvetha <suvethavelusamy69@gmail.com>
Date:   Fri Jul 12 18:40:33 2024 +0530

        Commit First

commit cf4fc7c7b7c79acc2a9580cb04e537a6f41d30c85d
Author: suvetha <suvethavelusamy69@gmail.com>
Date:   Fri Jul 12 18:22:58 2024 +0530

        Change Directory

commit a13ee07de83e08744e55080dd6722d05d37541f662c
Author: suvetha <suvethavelusamy69@gmail.com>
Date:   Fri Jul 12 18:15:39 2024 +0530
```

- 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 x + v
Commit source files
commit 074f67eac327a6bb7b53b9257073b31b83213785
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:02:48 2024 +0530

    Commit Readme file

D:\MyRepository\source>git checkout master
D README.txt
D source/src2.txt
Switched to branch 'master'

D:\MyRepository\source>git log
commit 02985fe6ed031157c7a0b1a1032e5eb33636733 (HEAD -> master)
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Mon Jul 15 09:35:29 2024 +0530

    README.txt

commit f8630f1b1aaed65113670fa54e6a597ff6320d58d
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:54:27 2024 +0530

    Delete the commit

commit d16a9721a5586fb7d80dd9db17b6699c55d3412
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:46:23 2024 +0530

    commit the last changes

commit 86d956d71d1d5e751919b570c4eb8a2287f11v7
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:44:33 2024 +0530

    Commit First

commit cf7c47bc79ucc2a588cbcd4e537afef41d30c85d
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:22:58 2024 +0530

    Change Directory

commit a13ed07de83e874ff558b0d6722d95d375a1f662c
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:15:39 2024 +0530

    Commit source files

commit 074f67eac327a6bb7b53b9257073b31b83213785
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:02:48 2024 +0530

```

- 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.**

```

git CMD x + v
D source/src2.txt
Switched to branch 'master'

D:\MyRepository>gitk --all
commit 02985fe6ed031157c7a0b1a1032e5eb33636733 (HEAD -> master)
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Mon Jul 15 09:35:29 2024 +0530

    Commit Readme file

README.txt
commit f8630f1b1aaed65113670fa54e6a597ff6320d58d
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:54:27 2024 +0530

    Delete the commit

commit d16a9721a5586fb7d80dd9db17b6699c55d3412
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:46:23 2024 +0530

    commit the last changes

commit 86d956d71d1d5e751919b570c4eb8a2287f11v7
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:44:33 2024 +0530

    Commit First

commit cf7c47bc79ucc2a588cbcd4e537afef41d30c85d
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:22:58 2024 +0530

    Change Directory

commit a13ed07de83e874ff558b0d6722d95d375a1f662c
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:15:39 2024 +0530

    Commit source files

commit 074f67eac327a6bb7b53b9257073b31b83213785
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:02:48 2024 +0530

    Commit Readme file

commit 02985fe6ed031157c7a0b1a1032e5eb33636733 (HEAD -> MyBranch)
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Mon Jul 15 09:35:29 2024 +0530

    Commit Readme file

README.txt
commit f8630f1b1aaed65113670fa54e6a597ff6320d58d
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:54:27 2024 +0530

    Delete the commit

commit d16a9721a5586fb7d80dd9db17b6699c55d3412
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:46:23 2024 +0530

    commit the last changes

commit 86d956d71d1d5e751919b570c4eb8a2287f11v7
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:44:33 2024 +0530

    Commit First

commit cf7c47bc79ucc2a588cbcd4e537afef41d30c85d
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:22:58 2024 +0530

    Change Directory

commit a13ed07de83e874ff558b0d6722d95d375a1f662c
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:15:39 2024 +0530

    Commit source files

commit 074f67eac327a6bb7b53b9257073b31b83213785
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Fri Jul 12 18:02:48 2024 +0530

    Commit Readme file

D:\MyRepository>git checkout MyBranch
D README.txt
D source/src2.txt
Switched to branch 'MyBranch'

D:\MyRepository>source>gitk &
D:\MyRepository>

```

- 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.**

```

git CMD      X + v
commit README.txt

commit e590a5f9824e1dd8b3794d812ed116224c277ce
Author: suvetra <suethevelusamy697@gmail.com>
Date: Mon Jul 15 09:35:29 2024 +0530

    commit Myrepo

commit 82995fe0e660387c7a0b41a032e5eb63636733
Author: suvetra <suethevelusamy697@gmail.com>
Date: Mon Jul 15 09:35:29 2024 +0530

    README.txt

commit f8624f1bd4eadb5113670fe56e6a50746320d58d
Author: suvetra <suethevelusamy697@gmail.com>
Date: Fri Jul 12 18:54:27 2024 +0530

        Delete the commit

commit d16a6771da5864b7d88ddbd17b6699c55d3412
Author: suvetra <suethevelusamy697@gmail.com>
Date: Fri Jul 12 18:54:27 2024 +0530

        commit the last changes

commit 46d956d21de5ea751919b574c4eb8a3287f1147
Author: suvetra <suethevelusamy697@gmail.com>
Date: Fri Jul 12 18:44:33 2024 +0530

        Commit First

commit cfdf7c7bc79acc2a958bcd4e537a6f41d30c85d
Author: suvetra <suethevelusamy697@gmail.com>
Date: Fri Jul 12 18:22:58 2024 +0530

        Change Directory

commit a13ee67de83e874558b0d6722d05d37541f662c
Author: suvetra <suethevelusamy697@gmail.com>
Date: Fri Jul 12 18:15:39 2024 +0530

        Commit source files

commit 07f467eac327adab7b53d9257073b31b83213785
Author: suvetra <suethevelusamy697@gmail.com>
Date: Fri Jul 12 18:02:48 2024 +0530

        Commit Readme file

D:\MyRepository\source>gitk
D:\MyRepository\source>
```

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

```

git CMD      X + v
commit the last changes

commit 46d956d21de5ea751919b574c4eb8a3287f1147
Author: suvetra <suethevelusamy697@gmail.com>
Date: Fri Jul 12 18:44:33 2024 +0530

    Commit First

commit cfdf7c7bc79acc2a958bcd4e537a6f41d30c85d
Author: suvetra <suethevelusamy697@gmail.com>
Date: Fri Jul 12 18:22:58 2024 +0530

    Change Directory

commit a13ee67de83e874558b0d6722d05d37541f662c
Author: suvetra <suethevelusamy697@gmail.com>
Date: Fri Jul 12 18:15:39 2024 +0530

    Commit source files

commit 07f467eac327adab7b53d9257073b31b83213785
Author: suvetra <suethevelusamy697@gmail.com>
Date: Fri Jul 12 18:02:48 2024 +0530

    Commit Readme file

D:\MyRepository\source>git checkout MyBranch
D:\MyRepository\source> README.txt
D:\MyRepository\source> README.txt
Switched to branch 'MyBranch'

D:\MyRepository\source>cd source
The system cannot find the path specified.

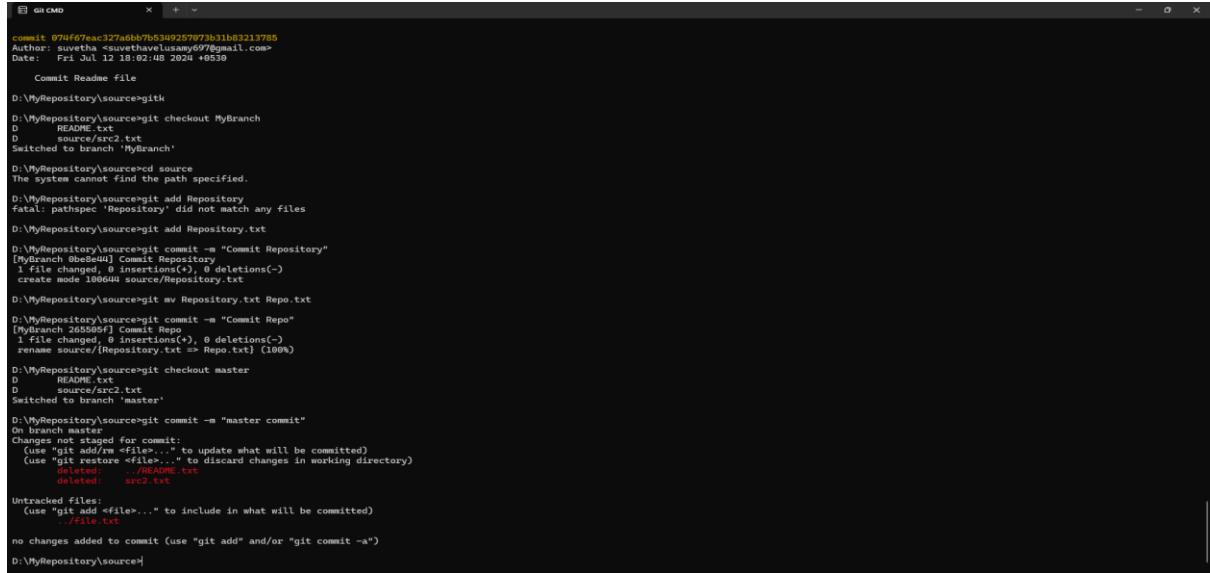
D:\MyRepository\source>git add Repository
fatal: pathspec 'Repository' did not match any files

D:\MyRepository\source>git add Repository.txt

D:\MyRepository\source>git commit -m "Commit Repository"
[MyBranch 0be86ef] Commit Repository
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 source/Repository.txt

D:\MyRepository\source>git mv Repository.txt Repo.txt
D:\MyRepository\source>git commit -m "Commit Repo"
[MyBranch 26550ff] Commit Repo
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename source/[Repository.txt => Repo.txt] (100%)
D:\MyRepository\source>
```

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.



```
gitcmd x + ~
D:\MyRepository\source>git
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:02:48 2024 +0530

Commit Readme file
D:\MyRepository\source>git checkout MyBranch
D README.txt
D source/src2.txt
Switched to branch 'MyBranch'

D:\MyRepository\source>cd source
The system cannot find the path specified.

D:\MyRepository\source>git add Repository
fatal: pathspec 'Repository' did not match any files

D:\MyRepository\source>git add Repository.txt
D:\MyRepository\source>git commit -m "Commit Repository"
[MyBranch 0b8e844] Commit Repository
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 source/Repository.txt

D:\MyRepository\source>git mv Repository.txt Repo.txt
D:\MyRepository\source>git commit -m "Commit Repo"
[MyBranch 265586f] Commit Repo
1 file changed, 0 insertions(+), 0 deletions(-)
rename source/Repository.txt => Repo.txt (100%)

D:\MyRepository\source>git checkout master
D README.txt
D source/src2.txt
Switched to branch 'master'

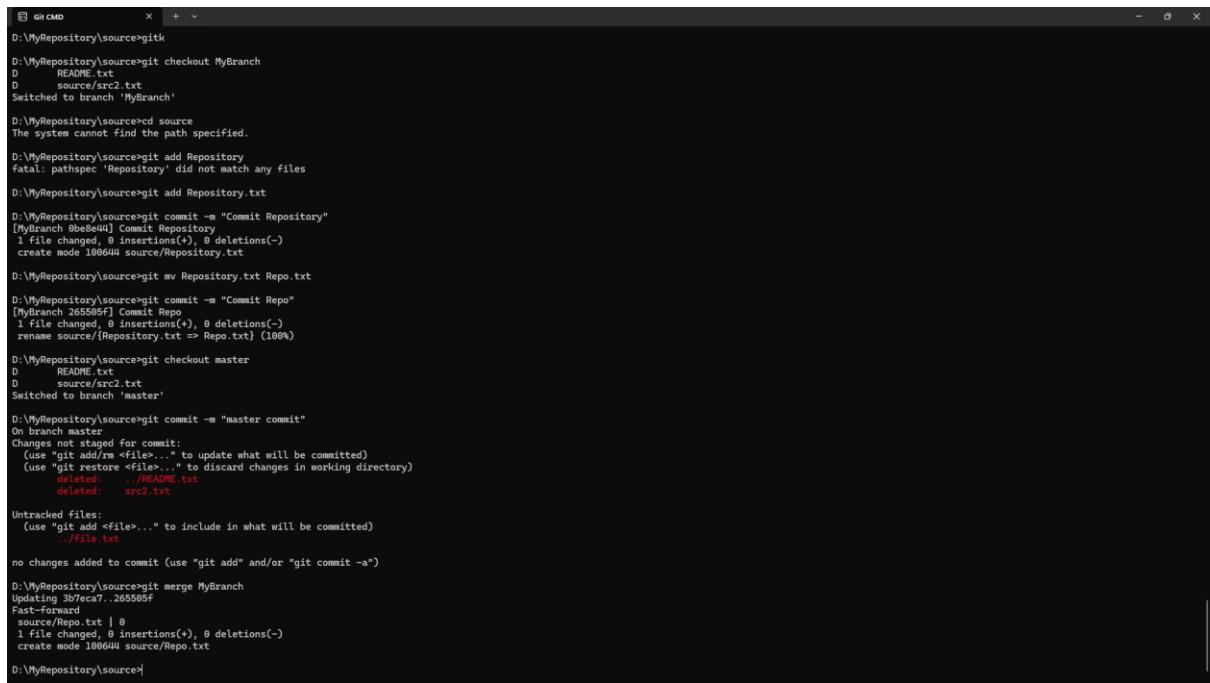
D:\MyRepository\source>git commit -m "master commit"
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted: README.txt
    deleted: src2.txt

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:\MyRepository\source>
```

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.)



```
gitcmd x + ~
D:\MyRepository\source>git
D:\MyRepository\source>git checkout MyBranch
D README.txt
D source/src2.txt
Switched to branch 'MyBranch'

D:\MyRepository\source>cd source
The system cannot find the path specified.

D:\MyRepository\source>git add Repository
fatal: pathspec 'Repository' did not match any files

D:\MyRepository\source>git add Repository.txt
D:\MyRepository\source>git commit -m "Commit Repository"
[MyBranch 0b8e844] Commit Repository
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 source/Repository.txt

D:\MyRepository\source>git commit -m "Commit Repo"
[MyBranch 265586f] Commit Repo
1 file changed, 0 insertions(+), 0 deletions(-)
rename source/Repository.txt => Repo.txt (100%)

D:\MyRepository\source>git checkout master
D README.txt
D source/src2.txt
Switched to branch 'master'

D:\MyRepository\source>git commit -m "master 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: src2.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:\MyRepository\source>git merge MyBranch
Updating 3b7eca7..265586f
Fast-forward
 source/Repo.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 source/Repo.txt

D:\MyRepository\source>
```

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.

The terminal window shows the following git log output:

```

commit 0be8e44f8b4e9575126ffddaca0d63886561d9c1
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Mon Jul 16 11:12:18 2024 +0530

Commit Repository

commit 3b7eca7ff0e6cff4ff0bd53001528d0ff4fb0fec2586
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Mon Jul 16 10:24:35 2024 +0530

    commit README.txt

commit e59445f9824e1dd8bc039e4812ed11624c277ce
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Mon Jul 15 10:22:18 2024 +0530

        commit Myrepo

commit 62905fe0e1603157c7a6b43a1032e5eb63636733
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Mon Jul 15 09:35:29 2024 +0530

    README.txt

commit f0824f1bdaeadb5113670fe56e6a50746320d58d
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:54:27 2024 +0530

        Delete the commit

commit d16a6721da55064b7d88dd9db17b6699c55d3412
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:46:23 2024 +0530

        commit the last changes

commit 46d956d212de5ea751919b570c4ebba3287f1147
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:44:33 2024 +0530

        Commit First

commit cf7c747bc79acc2a958bcd4e537a6f41d30c85d
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:22:58 2024 +0530

        Change Directory

D:\MyRepository\source>git
D:\MyRepository\source>
```

The gitk window shows the commit history as a Directed Acyclic Graph (DAG). It displays several commits, including a merge commit that joins the branches. The commits are color-coded and show their relationships through arrows indicating merges and rebases.

Stretch Task:

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

The terminal window shows the following sequence of git commands and their outputs:

```

commit d16a6721da55064b7d88dd9db17b6699c55d3412
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:46:23 2024 +0530

    commit the last changes

commit 46d956d212de5ea751919b570c4ebba3287f1147
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:44:33 2024 +0530

    Commit First

commit cf7c747bc79acc2a958bcd4e537a6f41d30c85d
Author: suvetha <suvethavelusamy69@gmail.com>
Date: Fri Jul 12 18:22:58 2024 +0530

    Change Directory

D:\MyRepository\source>git
D:\MyRepository\source>git checkout MyBranch
D      README.txt
D      source/src2.txt
Switched to branch "MyBranch"

D:\MyRepository\source>git add New
fatal: pathspec 'New' did not match any files
D:\MyRepository\source>git add New.txt

D:\MyRepository\source>git commit -m "New Commit"
[MyBranch 0169fb] New Commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 source/New.txt

D:\MyRepository\source>git rv New.txt New2.txt
git: 'rv' is not a git command. See 'git --help'.

The most similar commands are
  mv
  rm

D:\MyRepository\source>git rm New.txt New2.txt
fatal: pathspec 'New2.txt' did not match any files
D:\MyRepository\source>git mv New.txt New2.txt

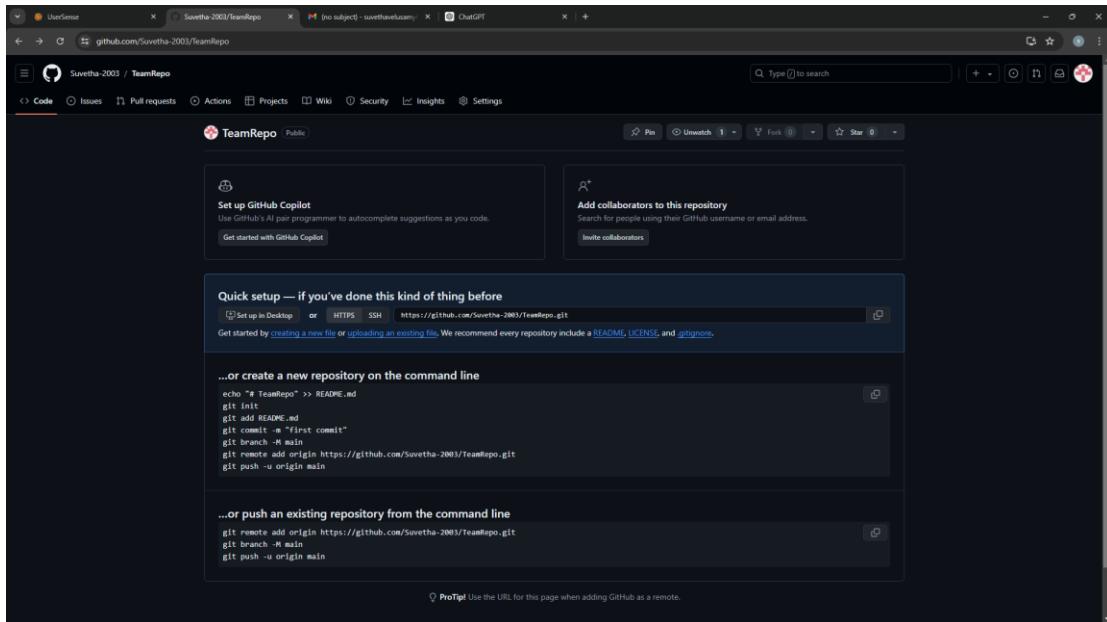
D:\MyRepository\source>git commit -m "New2 commit"
[MyBranch 6fc652c] New2 commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename source/[New.txt => New2.txt] (100%)

D:\MyRepository\source>
```

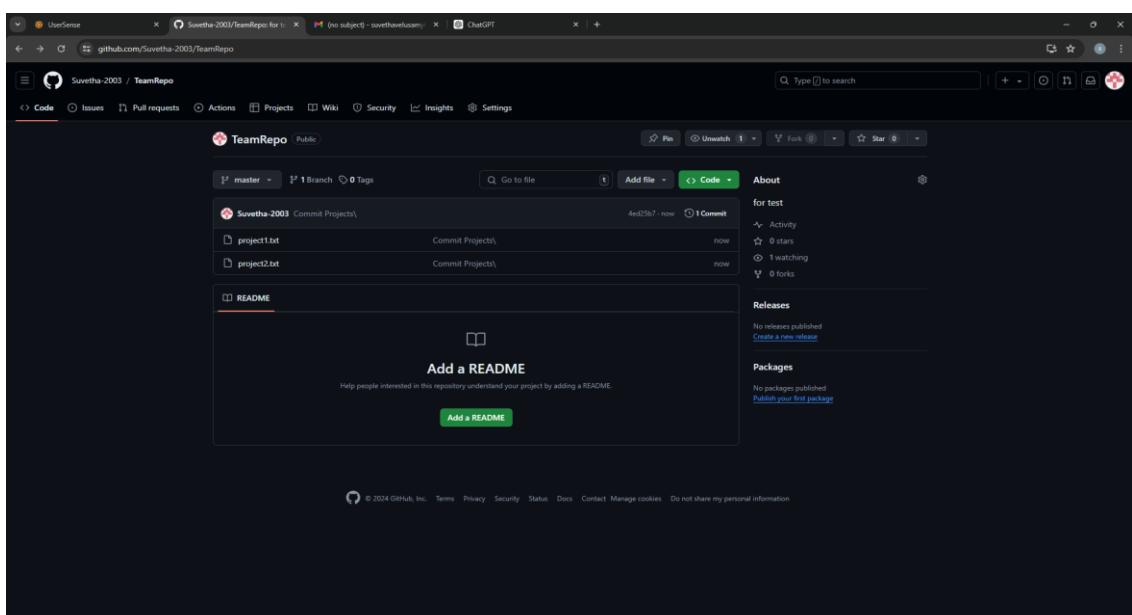
Exercise 3:

Main Task:

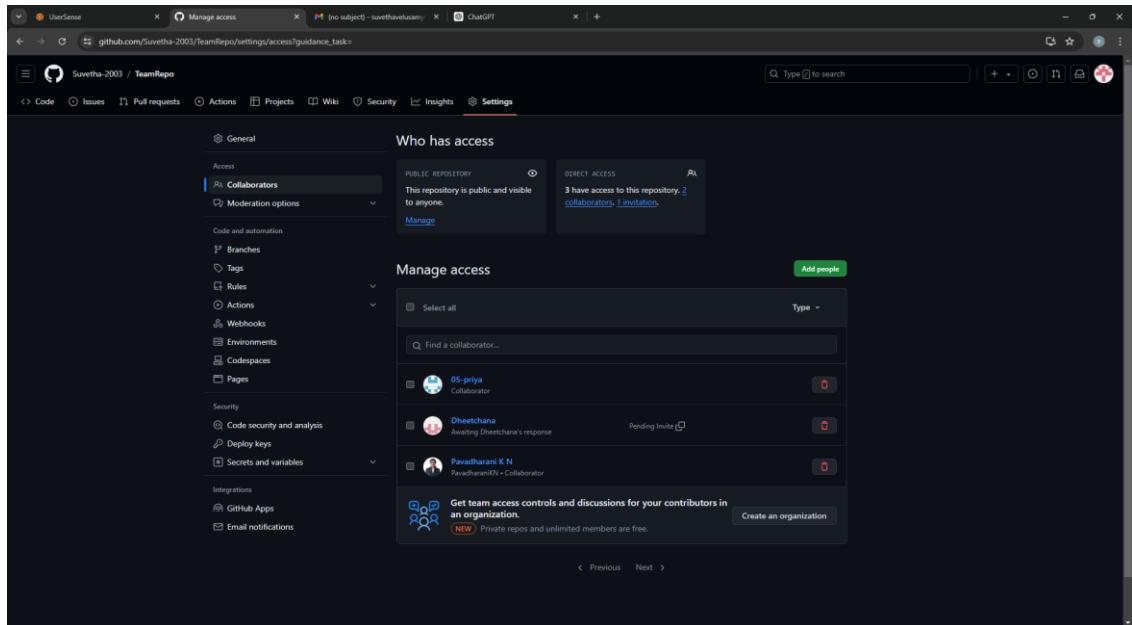
1. First, one person in the group should create a public repository using their GitHub account.



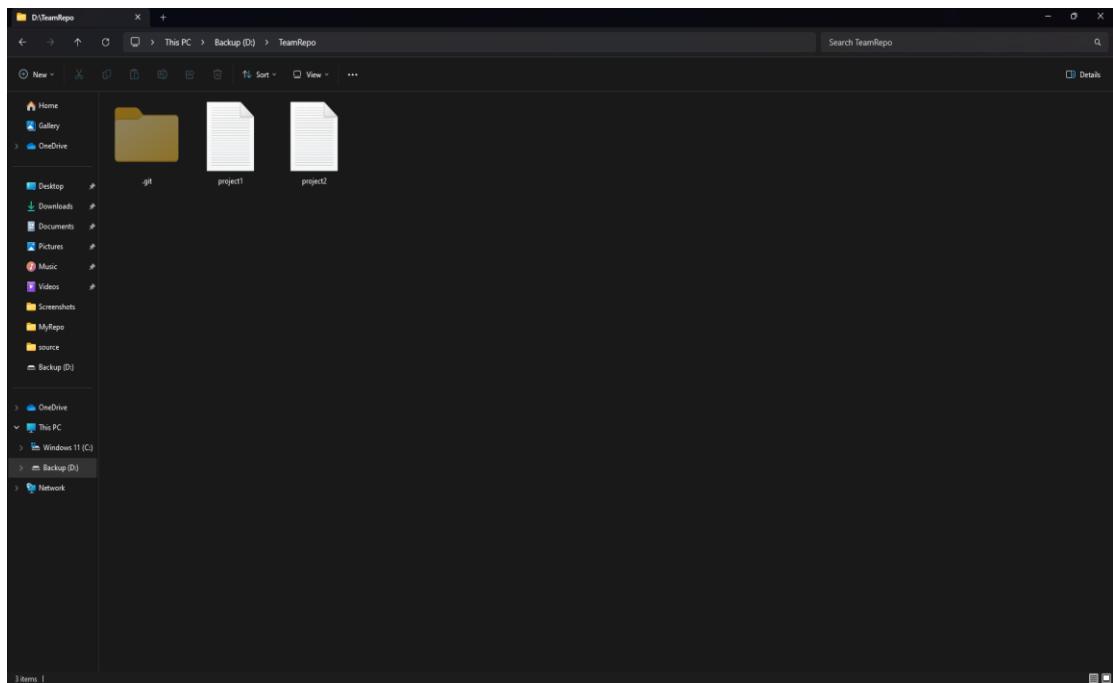
2. This same person should then follow the instructions from GitHub to add a remote, and then push their repository. Do not forget the –u flag, as suggested by GitHub!



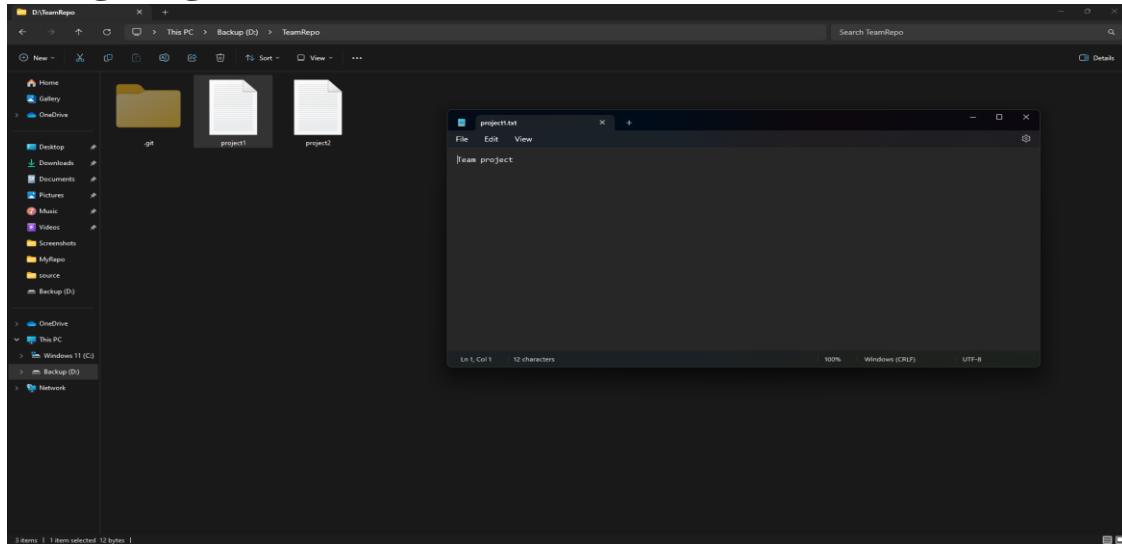
3. All of the other members of the group should then be added as collaborators, so they can commit to the repository also.



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



- 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).**



- 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 branch
* master

D:\TeamRepo>git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 232 bytes | 232.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Suvetha-2003/TeamRepo.git
 * [new branch] master -> master
branch 'master' set up to track 'origin/master'.

D:\TeamRepo>cd ..

D:\>git clone "https://github.com/Suvetha-2003/TeamRepo.git"
Cloning into 'TeamRepo'.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Writing objects: 100% (3/3), 92.00 KiB/s, done.
From https://github.com/Suvetha-2003/TeamRepo
 * branch    master      -> FETCH_HEAD
 * branch    7eca4d4 master      -> origin/master
Fast-forward
 project1.txt | 1 +
 1 file changed, 1 insertion(+)

D:\TeamRepo>git log
commit 7eca4d4c5b0f39881c6d809a71ce735f7cb55079 (HEAD --> master, origin/master, origin/HEAD)
Author: Pavadharani Narugunray <pavadharaninarugunray74@gmail.com>
Date:   Tue Jul 16 09:45:39 2024 +0530

    Commit changes

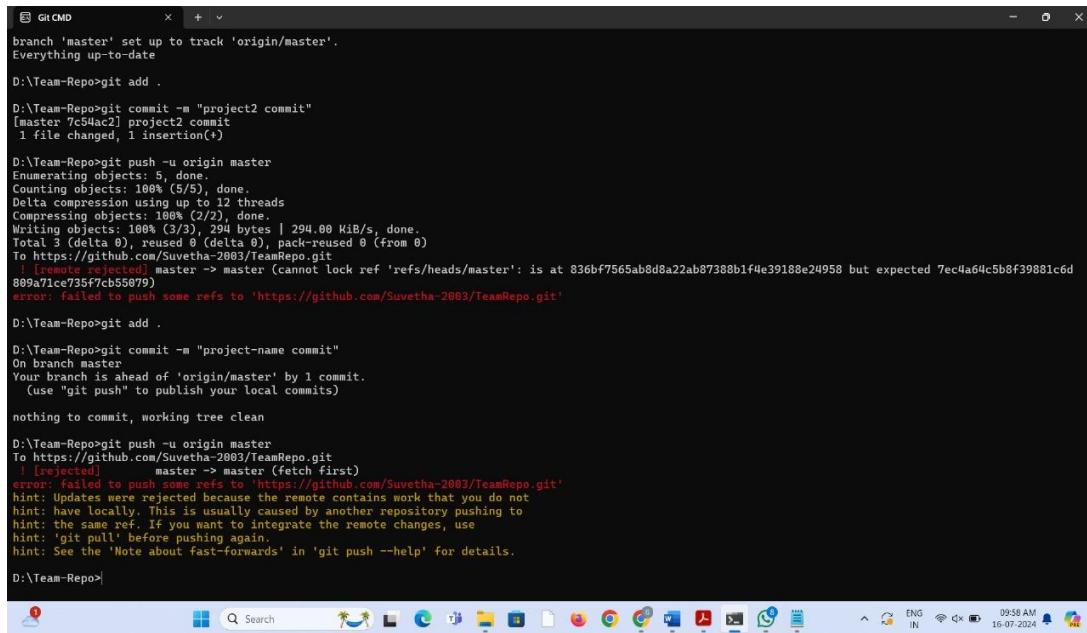
commit 4ed5b75e5dc7f797075456b12f2e532d011d6b063774
Author: suvetha <suvethavelusamy697@gmail.com>
Date:   Tue Jul 16 09:45:39 2024 +0530

    Commit Projects

D:\TeamRepo>
```

The consistent SHA-1 hash across all copies of a repository is crucial for maintaining data integrity, ensuring consistency, tracking changes accurately, facilitating efficient collaboration, and providing security against tampering.

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?



```
Git CMD
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 (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 836bf7565ab8d8a22ab87388b1f4e39188e24958 but expected 7ec4a64c5b8f39881c6d809a71ce735f7cb5879)
error: failed to push some refs to 'https://github.com/Suvetha-2003/TeamRepo.git'

D:\Team-Repo>git add .

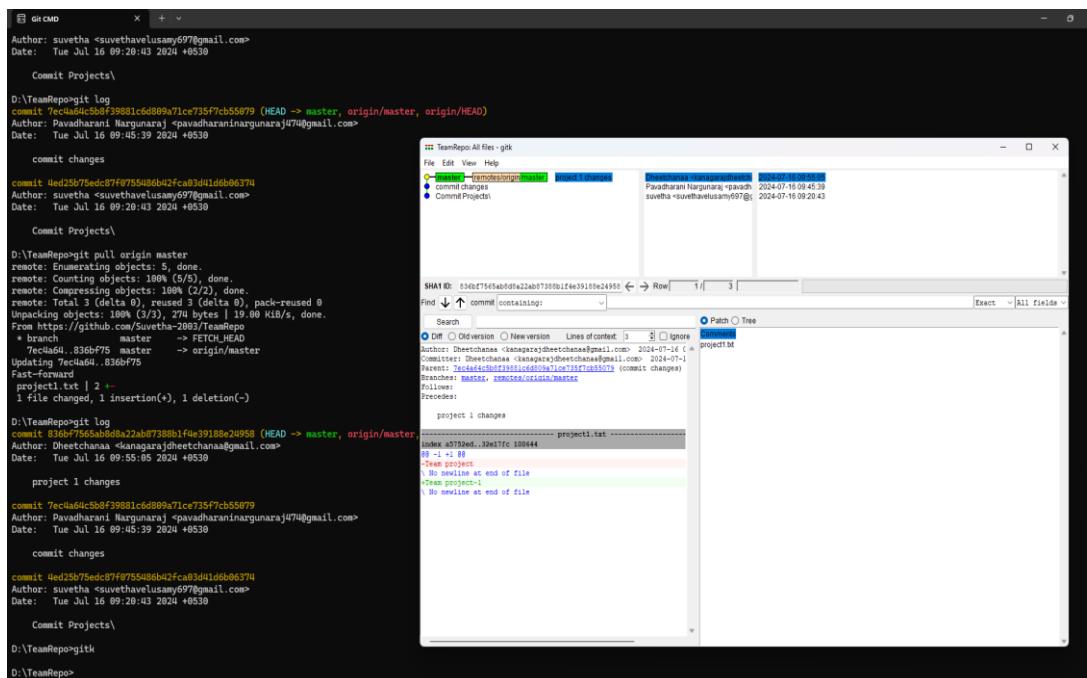
D:\Team-Repo>git commit -m "project-name commit"
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

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.



```
Git CMD
Author: suvetha <suvethavealusamy597@gmail.com>
Date: Tue Jul 16 09:20:43 2024 +0530

Commit Projects<

commit 7ec4a64c5b8f39881c6d809a71ce735f7cb5879 (HEAD => master, origin/master, origin/HEAD)
Author: Pavadharani Nargunraju <pavadharaninargunraju74@gmail.com>
Date: Tue Jul 16 09:45:39 2024 +0530

  Commit changes

commit 8ed15b75edc87ff07554b8d42fca83d01d0b86374
Author: suvetha <suvethavealusamy597@gmail.com>
Date: Tue Jul 16 09:20:43 2024 +0530

  Commit Projects<

Commit Projects<

D:\TeamRepo>git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Upgrading objects: 100% (3/3), 294 bytes | 19.00 KiB/s, done.
From https://github.com/Suvetha-2003/TeamRepo
 * branch      master    -> FETCH_HEAD
 7ec4a64..836bf75* master    -> origin/master
Updating 7ec4a64..836bf75
Fast-Forward
  project1.txt | 2 ++
  1 file changed, 1 insertion(+), 1 deletion(-)

D:\TeamRepo>git log
commit 836bf7565ab8d8a22ab87388b1f4e39188e24958 (HEAD => master, origin/master)
Author: suvetha <suvethavealusamy597@gmail.com>
Date: Tue Jul 16 09:55:05 2024 +0530

  project 1 changes

commit 7ec4a64c5b8f39881c6d809a71ce735f7cb5879
Author: Pavadharani Nargunraju <pavadharaninargunraju74@gmail.com>
Date: Tue Jul 16 09:45:39 2024 +0530

  Commit changes

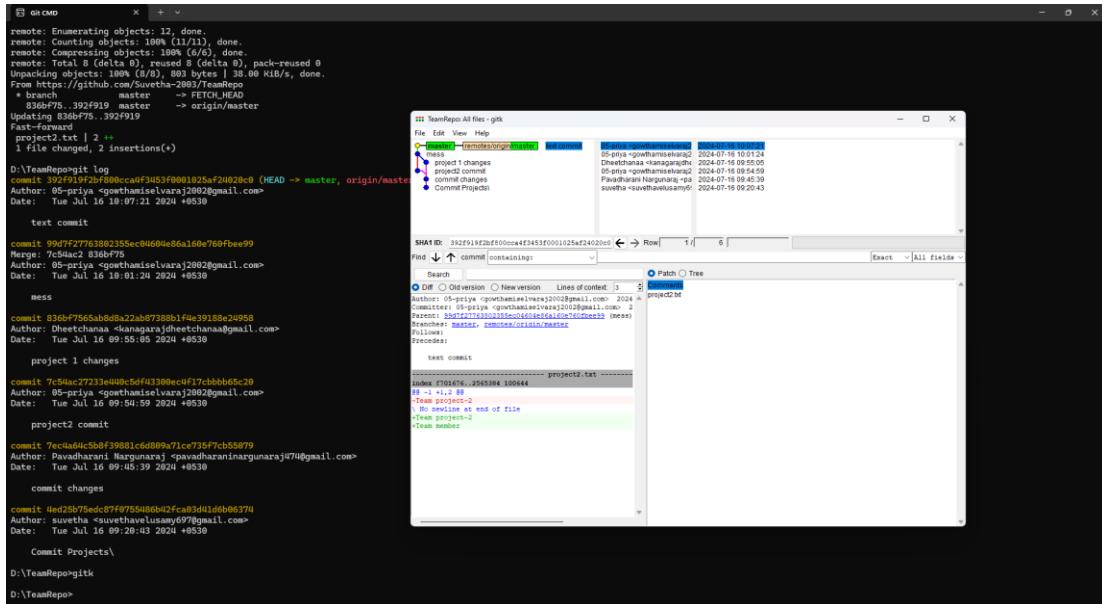
commit 8ed15b75edc87ff07554b8d42fca83d01d0b86374
Author: suvetha <suvethavealusamy597@gmail.com>
Date: Tue Jul 16 09:20:43 2024 +0530

  Commit Projects<

Commit Projects<

D:\TeamRepo>gitk
```

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



```
git cmd x + ~
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 8 (delta 0), reused 8 (delta 0), pack-reused 0
Unpacking objects: 100% (8/8), 893 bytes | 38.00 KiB/s, done.
From https://github.com/priyaj2024/TeamRepo
 * branch      master      -> FETCH_HEAD
   836bf75..392f919 master -> origin/master
Updating 836bf75..392f919
Fast-forward
  project2.txt | 2 ++
  1 file changed, 2 insertions(+)

D:\TeamRepo>git log
commit 392f9192bf800cca4f3453f0001025af24020c0 [HEAD -> master, origin/master]
Author: 05-priya <sowthaiselvaraj2024@gmail.com>
Date: Tue Jul 16 10:07:21 2024 +0530

    text commit

commit 99cf427f780021b5ec04600e86a160e760fbef99
Merge: 836bf75 392f919
Author: 05-priya <sowthaiselvaraj2024@gmail.com>
Date: Tue Jul 16 10:01:24 2024 +0530

    BESS

commit 836bf7565ab8d8a22ab87388b1fae39188e24958
Author: Dheetchanaa <kanganarajdhetchanaa@gmail.com>
Date: Tue Jul 16 09:55:05 2024 +0530

    project 1 changes

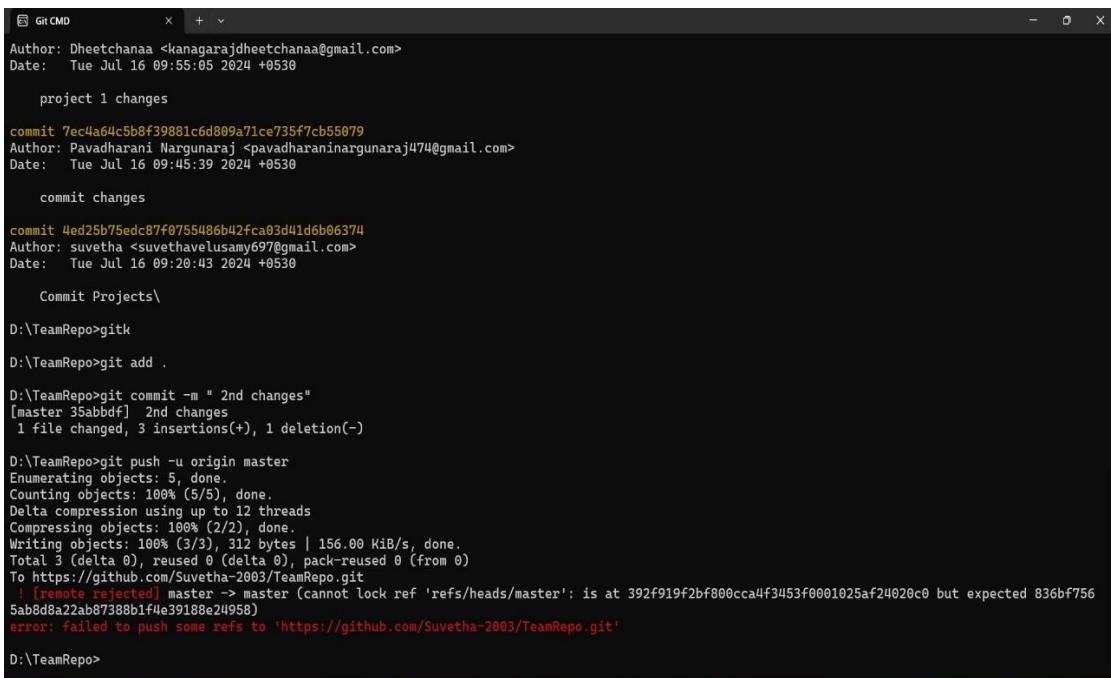
commit 7ec4a64c5b8f39881c6d809a71ce735f7cb55079
Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Date: Tue Jul 16 09:45:39 2024 +0530

    commit changes

commit 4ed25b75edc87f0755486b42fcfa03d41d6b06374
Author: suvetha <suvethavelusamy697@gmail.com>
Date: Tue Jul 16 09:20:43 2024 +0530

    Commit Projects\

D:\TeamRepo>gitk
D:\TeamRepo>
```



```
git cmd x + ~
Author: Dheetchanaa <kanganarajdhetchanaa@gmail.com>
Date: Tue Jul 16 09:55:05 2024 +0530

project 1 changes

commit 7ec4a64c5b8f39881c6d809a71ce735f7cb55079
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>git add .

D:\TeamRepo>git commit -m " 2nd changes"
[master 35abbd] 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 392f9192bf800cca4f3453f0001025af24020c0 but expected 836bf756
5ab8d8a22ab87388b1f4e39188e24958)
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.

```
GH CMD      x + v
Date: Tue Jul 16 09:45:39 2024 +0530
commit changes
commit 4ed25b75edc87f97554065b2fc63d1ld6b06374
Author: suvetha <suvethavelusamy97@gmail.com>
Date: Tue Jul 16 09:28:43 2024 +0530
  Commit Projects
D:\TeamRepo>git pull origin master
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 16 (delta 0), reused 16 (delta 0), pack-reused 0
Unpacking objects: 100% (16/16), 1.64 KB | 36.00 KiB/s, done.
From https://github.com/SuVetha-2003/TeamRepo
 * branch    master      -> FETCH_HEAD
  a4b0925..fe00ba7 master      -> origin/master
Up-to-date.
Fast-forward
 project1.txt | 1 -
 1 file changed, 1 deletion(-)
D:\TeamRepo>git add .
D:\TeamRepo>git commit -m "commit file"
[master db08d23] 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 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.

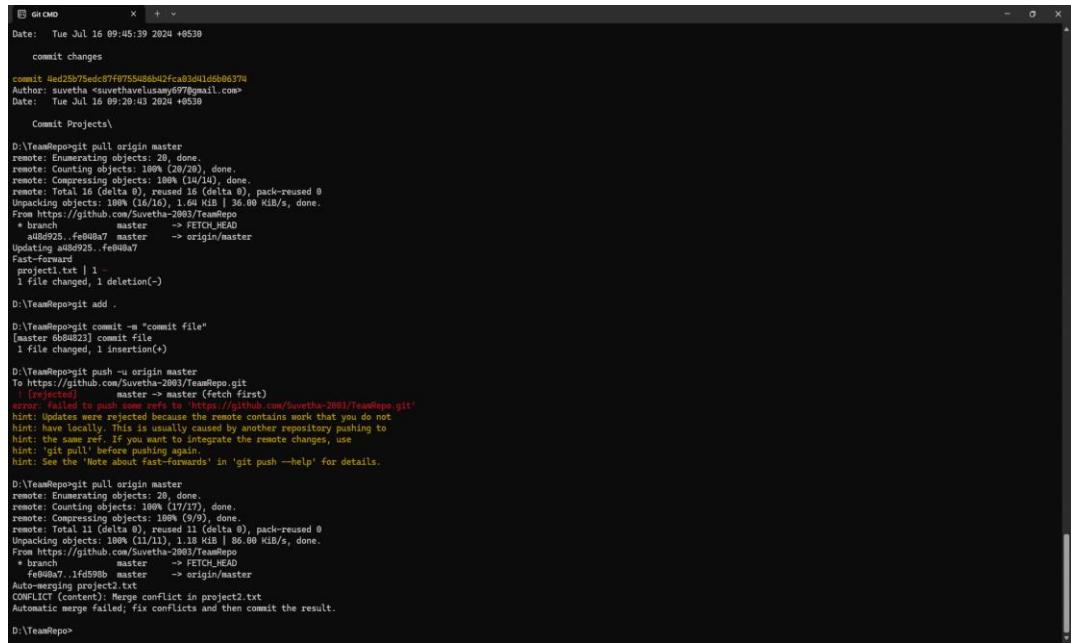
D:\TeamRepo>git pull origin master
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 11 (delta 0), reused 11 (delta 0), pack-reused 0
Unpacking objects: 100% (11/11), 1.13 KB | 36.00 KiB/s, done.
From https://github.com/SuVetha-2003/TeamRepo
 * branch    master      -> FETCH_HEAD
  + fe00ba7..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>
```

```
GH 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/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 KB | 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
  fe00ba7..1fd598b master -> master
branch 'master' set up to track 'origin/master'.
D:\TeamRepo>
```

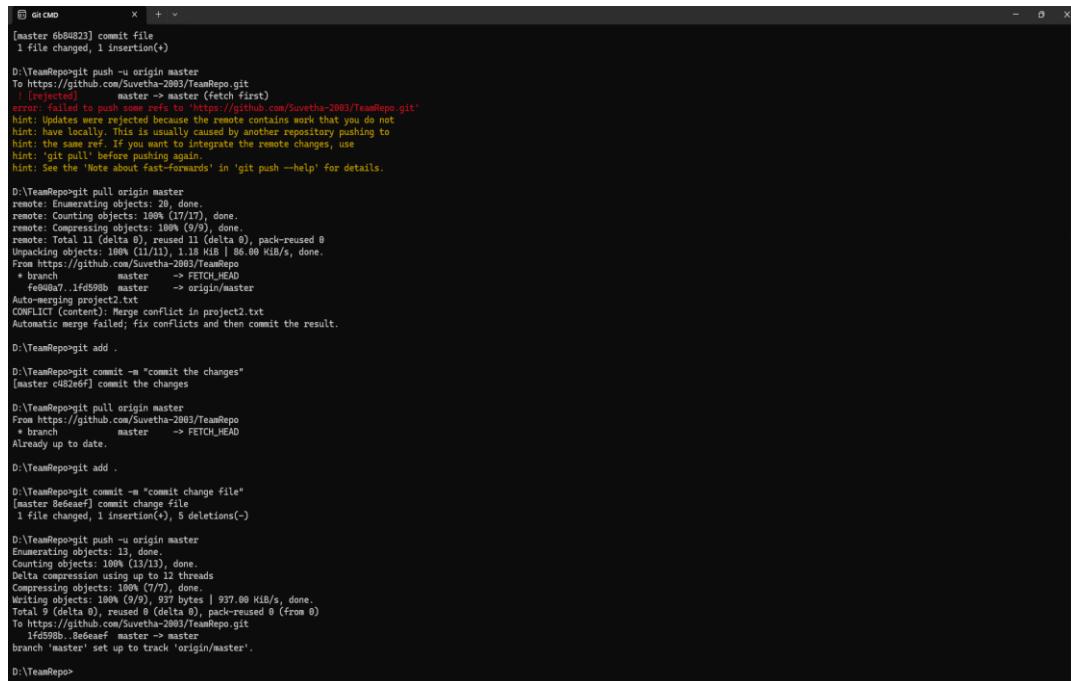
2. When the runner-up does a pull, they should get a merge conflict.



```
gitcmd x + 
Date: Tue Jul 16 09:45:39 2024 +0530
commit changes
commit 4edc25b7dec27f75d95b2fcfa32d16db06574
Author: suvetha <suvethavellusamy597@gmail.com>
Date: Tue Jul 16 09:29:43 2024 +0530
  Commit Projects
D:\TeamRepo>git pull origin master
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 11 (delta 0), reused 0 (delta 0), pack-reused 0
From https://github.com/Suventha-2003/TeamRepo
 * branch            master      -> FETCH_HEAD
  ab8d925..fe80a07  master      -> origin/master
Updating ab8d925..fe80a07
Fast-forward
  Auto-merging 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/Suventha-2003/TeamRepo.git
 ! [rejected]   master -> master (fetch first)
error: failed to push some refs to 'https://github.com/Suventha-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:\TeamRepo>git pull origin master
remote: Enumerating objects: 28, 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
From https://github.com/Suventha-2003/TeamRepo
 * branch            master      -> FETCH_HEAD
  fe80a07..1fd99b  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.

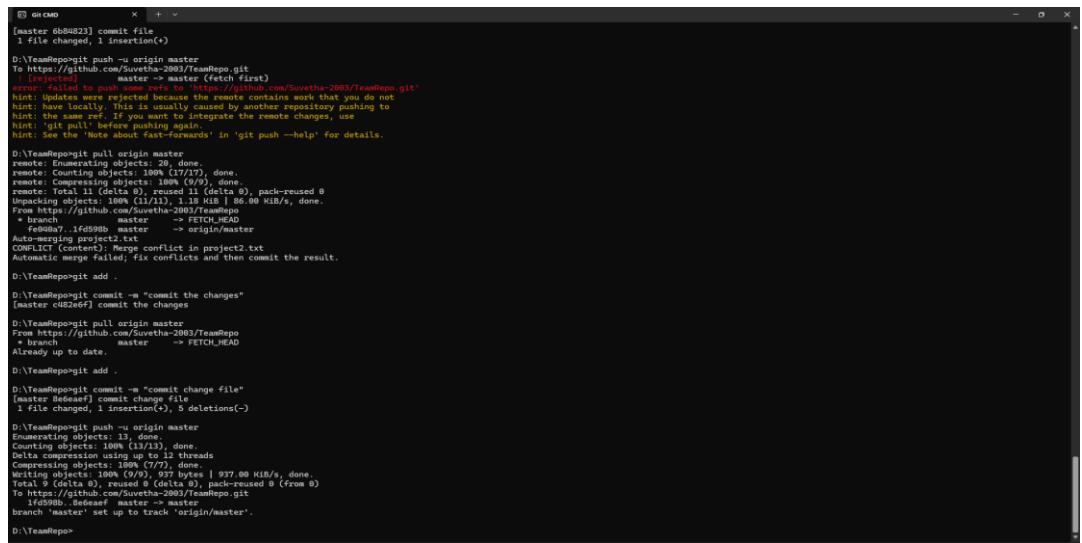


```
gitcmd x + 
[master 6b8a823] commit file
 1 file changed, 1 insertion(+)
D:\TeamRepo>git push -u origin master
To https://github.com/Suventha-2003/TeamRepo.git
 ! [rejected]   master -> master (fetch first)
error: failed to push some refs to 'https://github.com/Suventha-2003/TeamRepo.git'
hint: Updates 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:\TeamRepo>git pull origin master
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (20/20), 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), 1.18 kB | 86.00 kB/s, done.
From https://github.com/Suventha-2003/TeamRepo
 * branch            master      -> FETCH_HEAD
  fe80a07..1fd99b  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>git add .
D:\TeamRepo>git commit -m "commit the changes"
[master c462e6f] commit the changes
 1 file changed, 1 insertion(+)
D:\TeamRepo>git pull origin master
From https://github.com/Suventha-2003/TeamRepo
 * branch            master      -> FETCH_HEAD
Already up to date.

D:\TeamRepo>git .
D:\TeamRepo>git commit -m "commit change file"
[master 8e6eaef] commit change file
 1 file changed, 1 insertion(+), 5 deletions(-)
D:\TeamRepo>git push -u origin master
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 937 bytes | 937.00 kB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Suventha-2003/TeamRepo.git
 8e6eaef..c462e6f  master      -> master
branch 'master' set up to track 'origin/master'.
D:\TeamRepo>
```

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] x + v
[master db84b23] commit file
 1 file changed, 1 insertion(+)

D:\TeamRepo>git push -u origin master
To https://github.com/Suveetha-2003/TeamRepo.git
 [rejected]
error: failed to push some refs to 'https://github.com/Suveetha-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.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

D:\TeamRepo>git pull origin master
remote: Enumerating objects: 109K (17/17), done.
remote: Counting objects: 109K (17/17), done.
remote: Total 109K (delta 0), reused 11K (delta 0), pack-reused 0
Unpacking objects: 109K (11/11), 1.00 MiB | 86.00 KiB/s, done.
 * branch      master    -> FETCH_HEAD
fe@fe0a7...1fd99b8  master -> origin/master
Automatic merge failed; fix conflicts and then commit the result.

D:\TeamRepo>git add .
D:\TeamRepo>git commit -m "commit the changes"
[master c62e6ef] commit the changes
 1 file changed, 1 insertion(+), 5 deletions(-)

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

D:\TeamRepo>git add .
D:\TeamRepo>git commit -m "commit change file"
[master b6eaeaf] commit change file
 1 file changed, 1 insertion(+), 5 deletions(-)

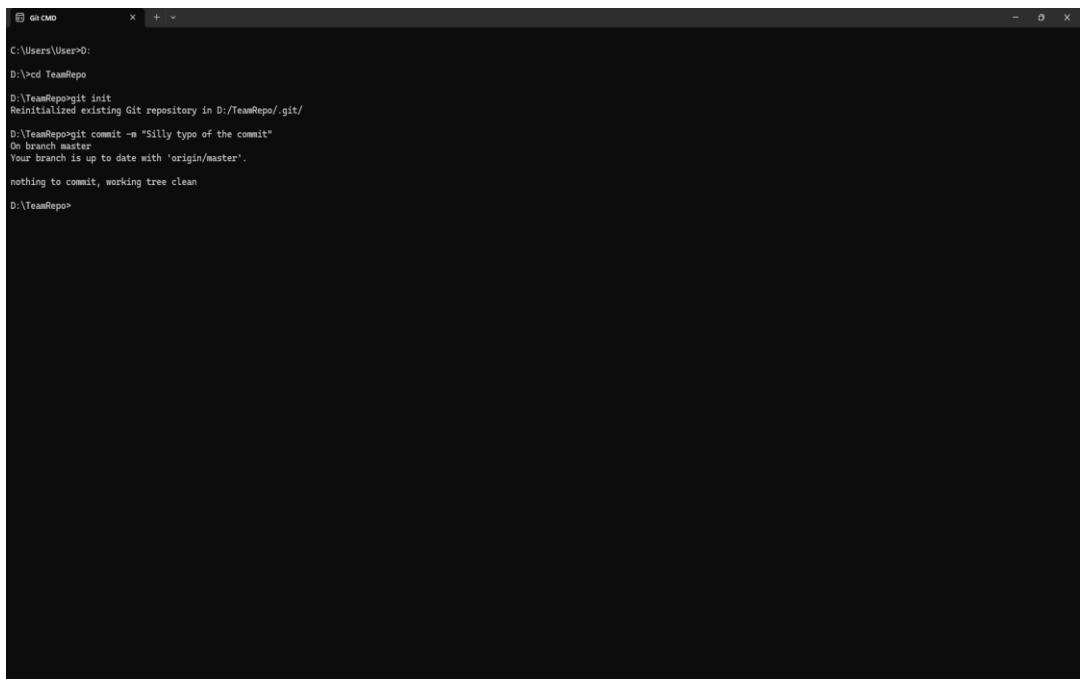
D:\TeamRepo>git push -u origin master
Counting objects: 109K (13/13), done.
Delta compression using up to 12 threads.
Compressing objects: 100% (97/97) | 937.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Suveetha-2003/TeamRepo.git
 fe@fe0a7...b6eaeaf master
branch 'master' set up to track 'origin/master'.
D:\TeamRepo>

```

Exercise 4:

Main Task:

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



```

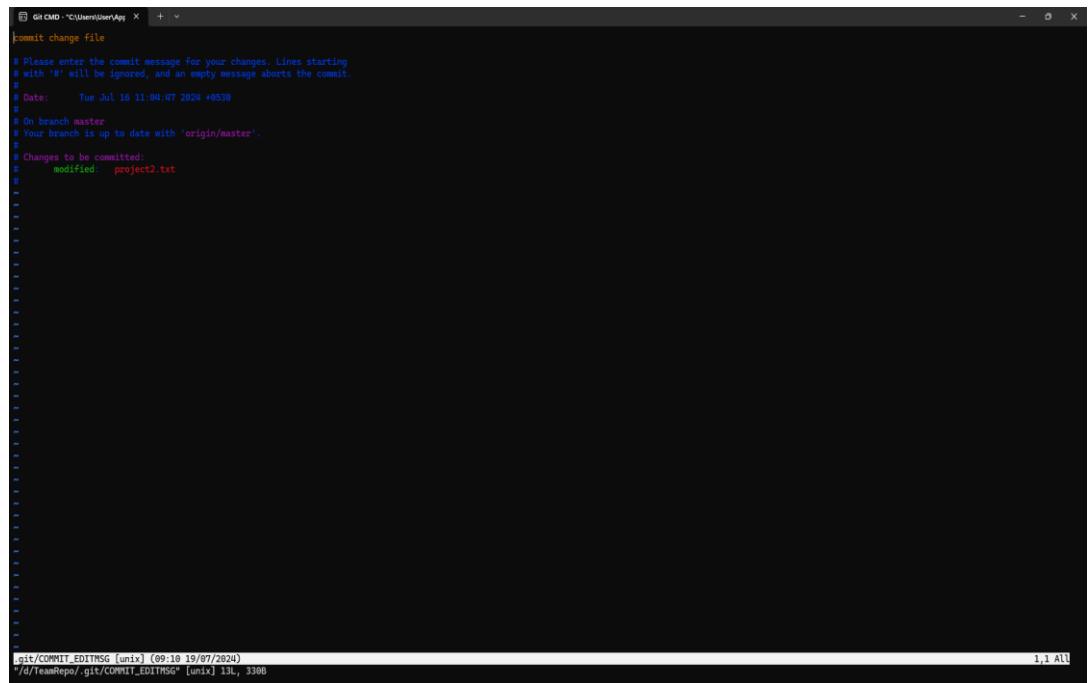
[git CMD] x + v
C:\Users\User>D:
D:\>cd TeamRepo

D:\TeamRepo>git init
Reinitialized existing Git repository in D:/TeamRepo/.git/
D:\TeamRepo>git commit -m "Silly typo of the commit"
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
D:\TeamRepo>

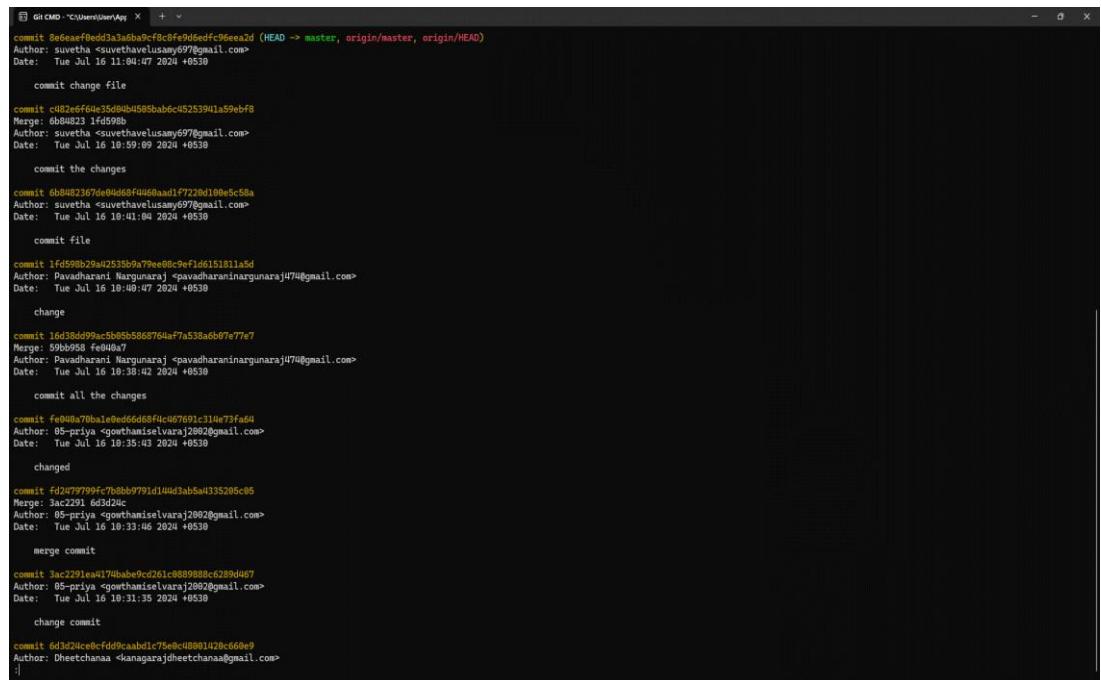
```

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



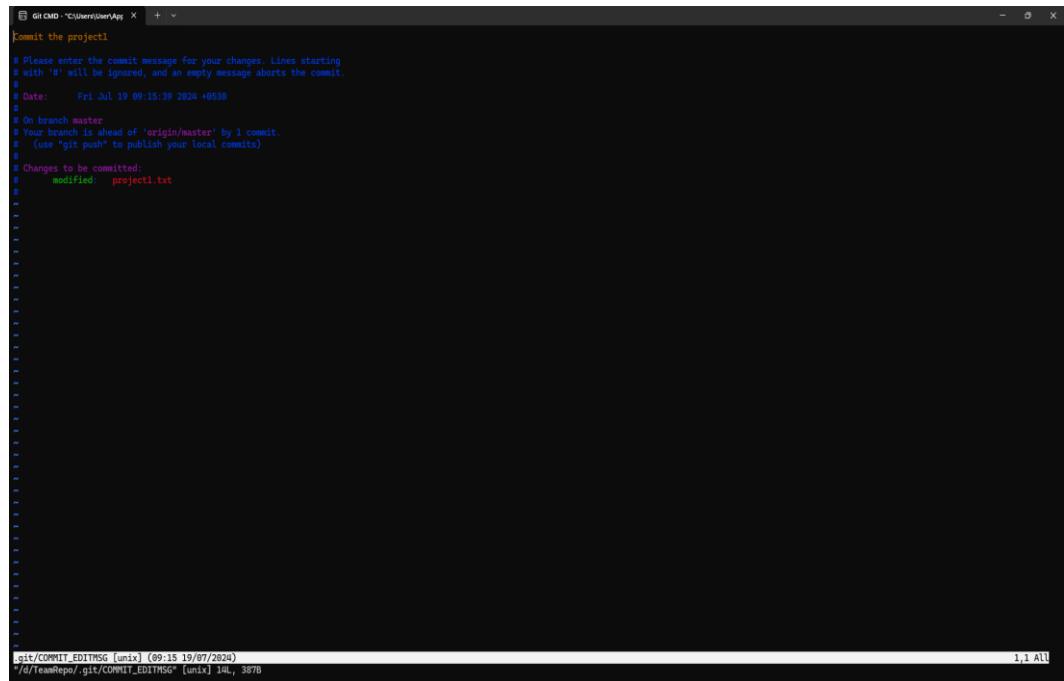
A screenshot of a Windows Command Prompt window titled "GIT CMD - C:\Users\user\App". The command entered is "git commit -amend". The output shows the current commit message, which includes a note about ignoring lines starting with '#'. The message itself is mostly blank, with a few lines of code visible at the bottom. The status bar at the bottom right shows "1,1 All" and the file path "/d/teamRepo/.git/COMMIT_EDITMSG [unix] (09:18 19/07/2024) 13L, 338B".

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



A screenshot of a Windows Command Prompt window titled "GIT CMD - C:\Users\user\App". The command entered is "git log". The output displays a detailed commit history. The first few commits are identical to the one shown in the previous screenshot, but they are followed by several more commits from different authors (suvetra, pavadarani, 05-pritya, Dheetchanaa) with various commit messages and dates. The log ends with a final commit from Dheetchanaa.

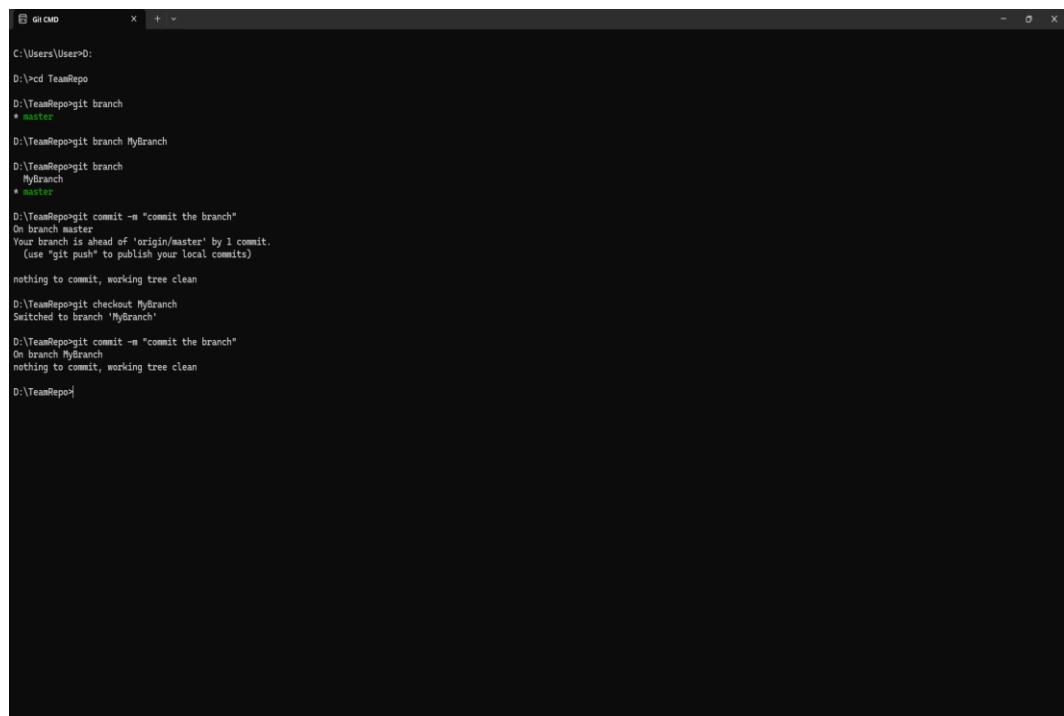
4. Now make a commit where you make a typo in one of the files. Once again, use --amend to magic away your problems.



```
git CMD - "C:\Users\User\AppData\Local\Temp\git-COMMIT_EDITMSG" [unix] (09:15 19/07/2024)
/D:/teamRepo/.git/COMMIT_EDITMSG [unix] 14L, 387B
```

The screenshot shows a terminal window titled 'GIT CMD'. The command entered is 'git commit -m "Commit the project"'. The terminal then displays the commit message editor. The message starts with '# Please enter the commit message for your changes. Lines starting # with '#' will be ignored, and an empty message aborts the commit'. It includes details like the date ('Date: Fri Jul 19 09:15:39 2024 +0530'), branch ('On branch master'), and the file being committed ('Changes to be committed: modified: project1.txt'). The message body is empty, indicated by numerous blank lines. At the bottom right of the editor, it says '1,1 All'.

5. Create a branch. Make a commit.

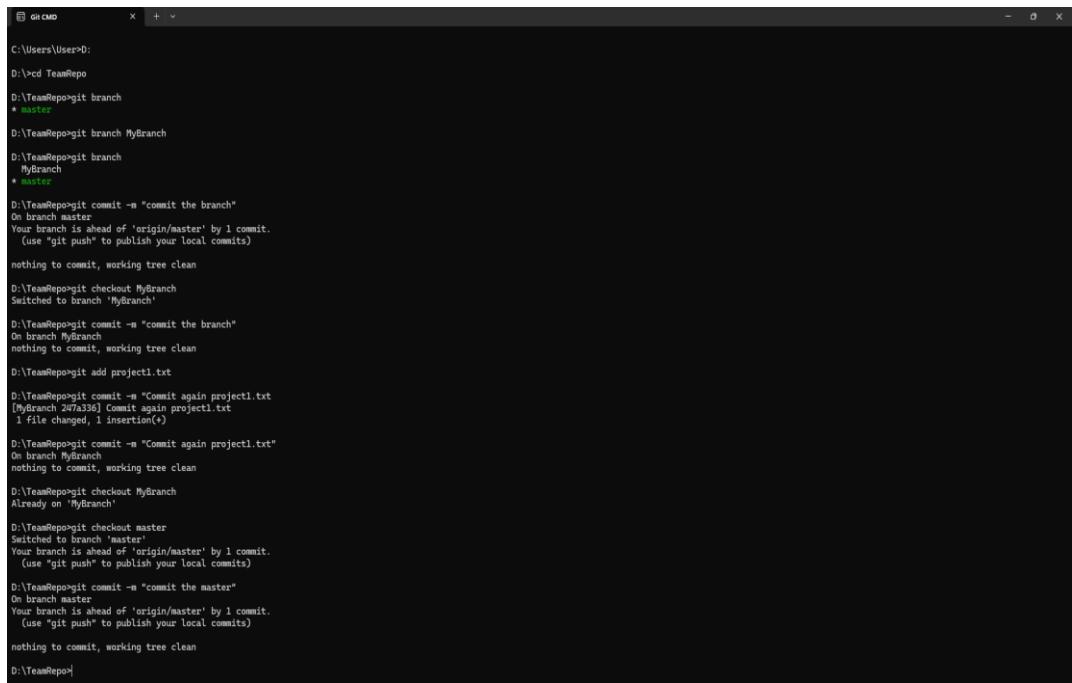


```
C:\Users\User>D:
D:\>cd TeamRepo
D:\TeamRepo>git branch
* master
D:\TeamRepo>git branch MyBranch
D:\TeamRepo>git branch
  MyBranch
* master
D:\TeamRepo>git commit -m "commit the branch"
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
D:\TeamRepo>git checkout MyBranch
Switched to branch 'MyBranch'
D:\TeamRepo>git commit -m "commit the branch"
On branch MyBranch
nothing to commit, working tree clean
D:\TeamRepo>
```

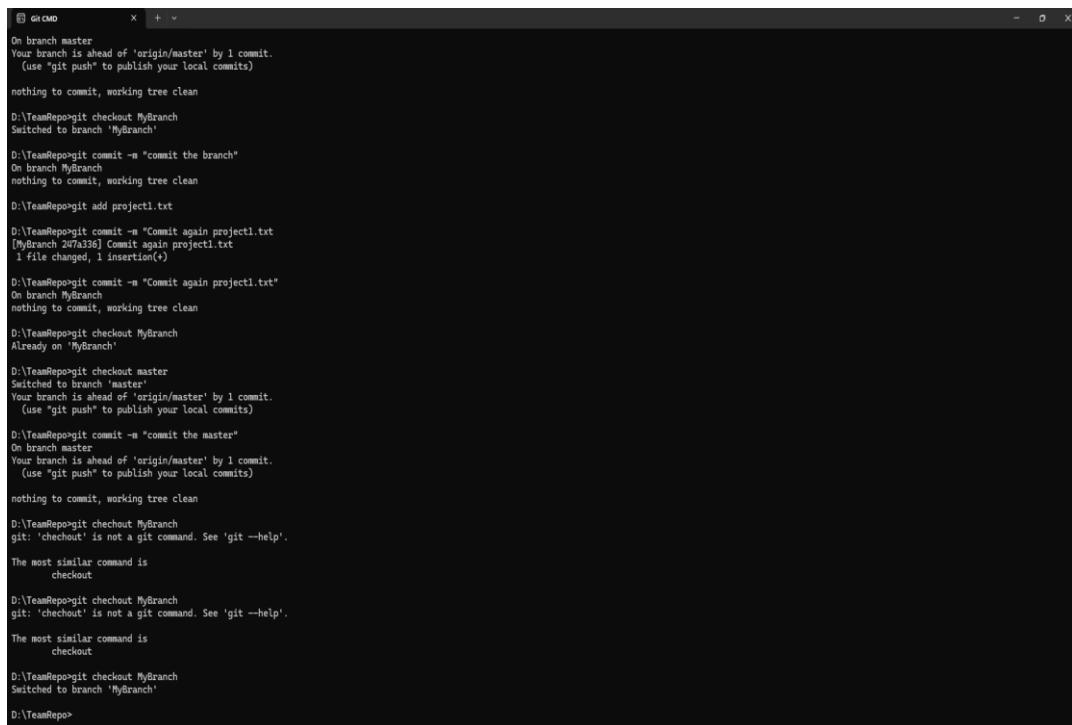
The screenshot shows a terminal window titled 'GIT CMD'. The user navigates to the 'TeamRepo' directory and lists existing branches ('git branch'). They then create a new branch named 'MyBranch' ('git branch MyBranch'). After switching to the new branch ('git checkout MyBranch'), they make a commit ('git commit -m "commit the branch"'). The commit message is empty. Finally, they switch back to the 'master' branch ('git checkout master'). The terminal shows the commit message 'commit the branch' was made on the 'MyBranch' branch.

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



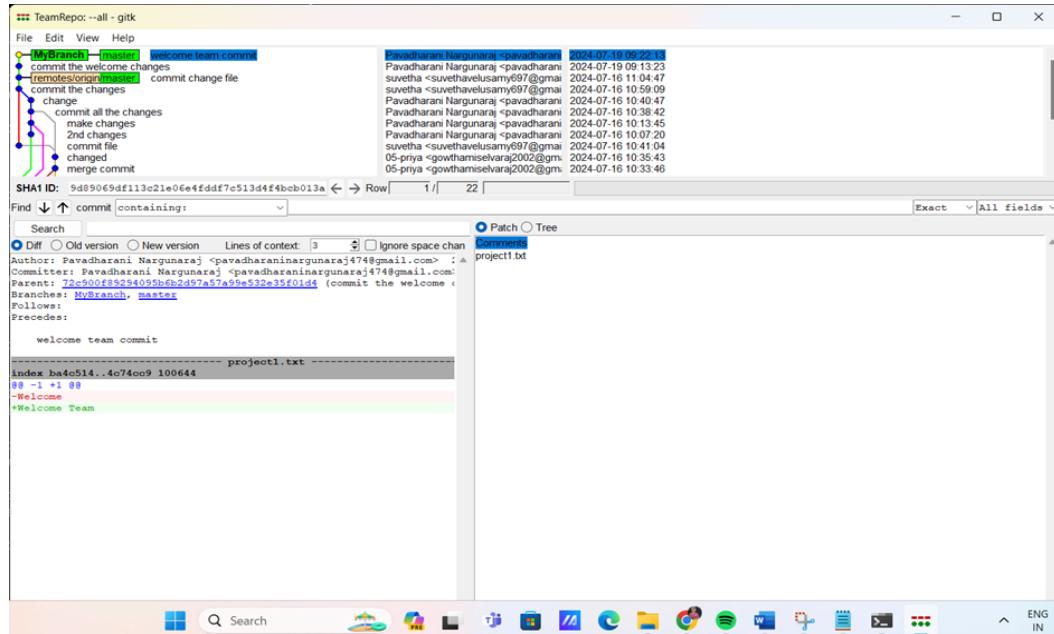
```
GH CMD      X + v
C:\Users\User>D:
D:\>cd TeamRepo
D:\TeamRepo>git branch
* master
D:\TeamRepo>git branch MyBranch
D:\TeamRepo>git branch
* MyBranch
* master
D:\TeamRepo>git commit -m "commit the branch"
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
nothing to commit, working tree clean
D:\TeamRepo>git checkout MyBranch
Switched to branch 'MyBranch'
D:\TeamRepo>git commit -m "commit the branch"
On branch MyBranch
nothing to commit, working tree clean
D:\TeamRepo>git add project1.txt
D:\TeamRepo>git commit -m "Commit again project1.txt"
[MyBranch 2d7a336] Commit again project1.txt
1 file changed, 1 insertion(+)
D:\TeamRepo>git commit -m "Commit again project1.txt"
On branch MyBranch
nothing to commit, working tree clean
D:\TeamRepo>git checkout MyBranch
Already on 'MyBranch'
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 commit -m "commit the master"
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>
```

7. Now switch back to your branch.



```
GH CMD      X + v
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 commit -m "commit the branch"
On branch MyBranch
nothing to commit, working tree clean
D:\TeamRepo>git add project1.txt
D:\TeamRepo>git commit -m "Commit again project1.txt"
[MyBranch 2d7a336] Commit again project1.txt
1 file changed, 1 insertion(+)
D:\TeamRepo>git commit -m "Commit again project1.txt"
On branch MyBranch
nothing to commit, working tree clean
D:\TeamRepo>git checkout MyBranch
Already on 'MyBranch'
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 commit -m "commit the master"
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
git: 'checkout' is not a git command. See 'git --help'.
The most similar command is
    checkout
D:\TeamRepo>git checkout MyBranch
git: 'checkout' is not a git command. See 'git --help'.
The most similar command is
    checkout
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.



gitk --all - gitk

File Edit View Help

MyBranch -> master welcome team commit

commit the welcome changes
commit change file
commit the changes
change
commit the changes
make changes
2nd changes
commit file
changed
merge commit

Pavadharani Nargunaraj <pavadharani.su@velusamy697@gmail.com> 2024-07-16 11:04:47
suveetha <suveethavelusamy697@gmail.com> 2024-07-16 10:59:09
Pavadharani Nargunaraj <pavadharani.su@velusamy697@gmail.com> 2024-07-16 10:40:47
Pavadharani Nargunaraj <pavadharani.su@velusamy697@gmail.com> 2024-07-16 10:38:42
Pavadharani Nargunaraj <pavadharani.su@velusamy697@gmail.com> 2024-07-16 10:36:45
Pavadharani Nargunaraj <pavadharani.suveetha <suveethavelusamy697@gmail.com> 2024-07-16 10:07:20
05-prya <gowthamiselvaraj2002@gmail.com> 2024-07-16 10:35:43
05-prya <gowthamiselvaraj2002@gmail.com> 2024-07-16 10:33:46

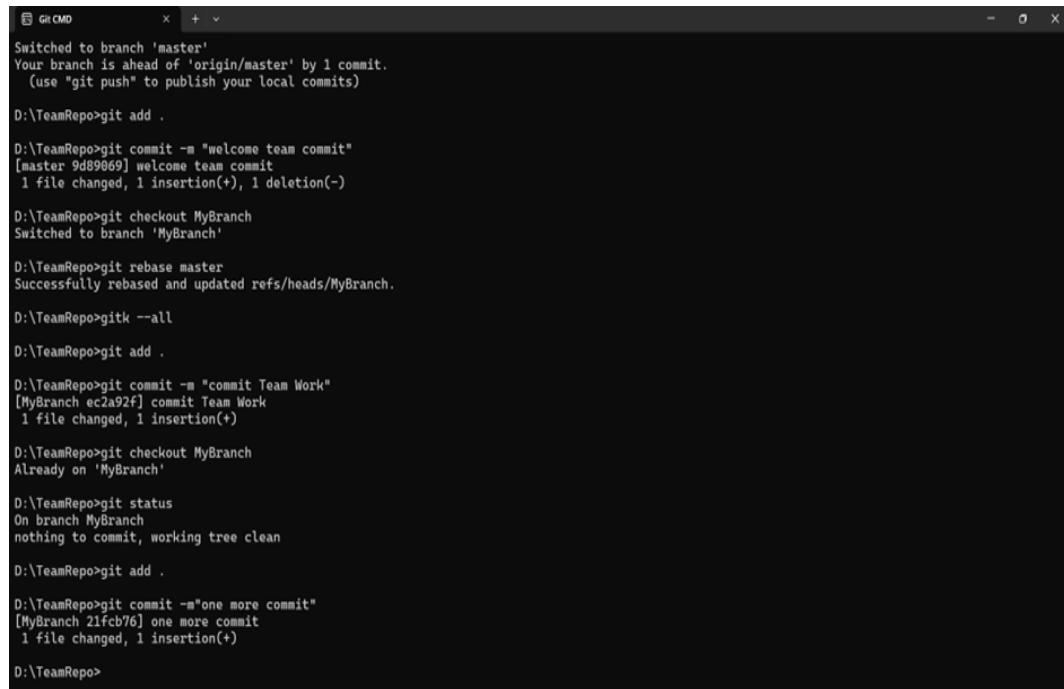
SHA1ID: 9d89069df113c21e06e4fddfc513d4f4bcb01a ← → Row| 1 / 22 | Exact All fields

Find ↑ comm → containing: Search Patch Tree Comments project1.txt

Author: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Committer: Pavadharani Nargunaraj <pavadharaninargunaraj474@gmail.com>
Parent: 72c900f5929405b6b2d57a57a59e532e35f01d4 (commit the welcome changes)
Branches: MyBranch, master
Follows:
Precedes:
welcome team commit

index b4c514..4e74cc09 100644
@@ -1 +1 @@
+Welcome
+Welcome Team

9. Make one more commit in your branch.



```
git CMD x + v
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>git --all

D:\TeamRepo>git add .

D:\TeamRepo>git commit -m "commit Team Work"
[MyBranch ec2a2f] commit Team Work
1 file changed, 1 insertion(+)

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

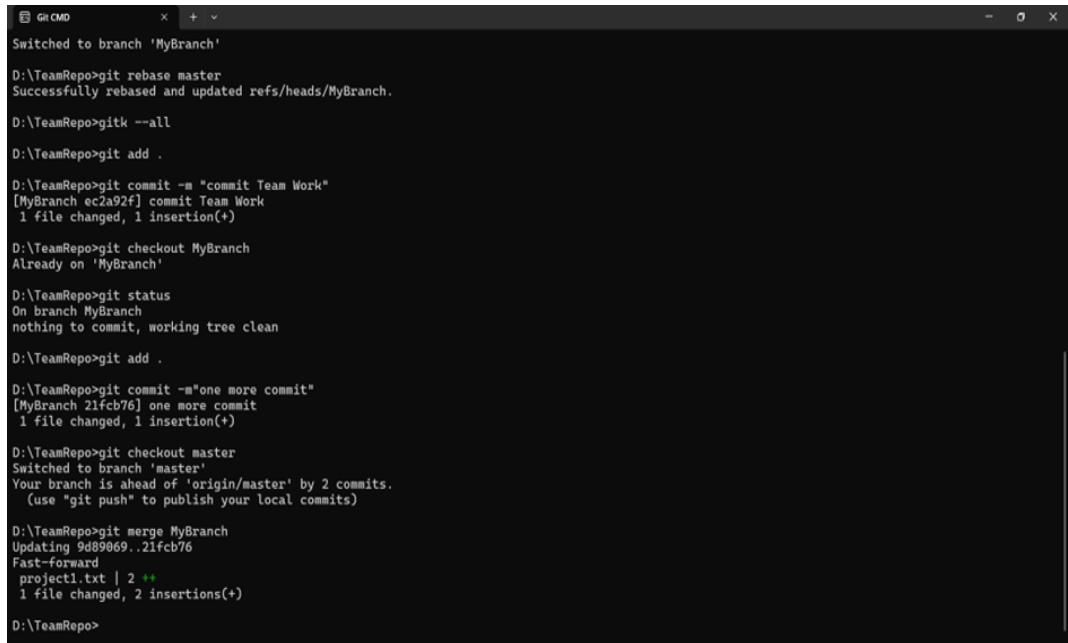
D:\TeamRepo>git status
On branch MyBranch
nothing to commit, working tree clean

D:\TeamRepo>git add .

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

D:\TeamRepo>
```

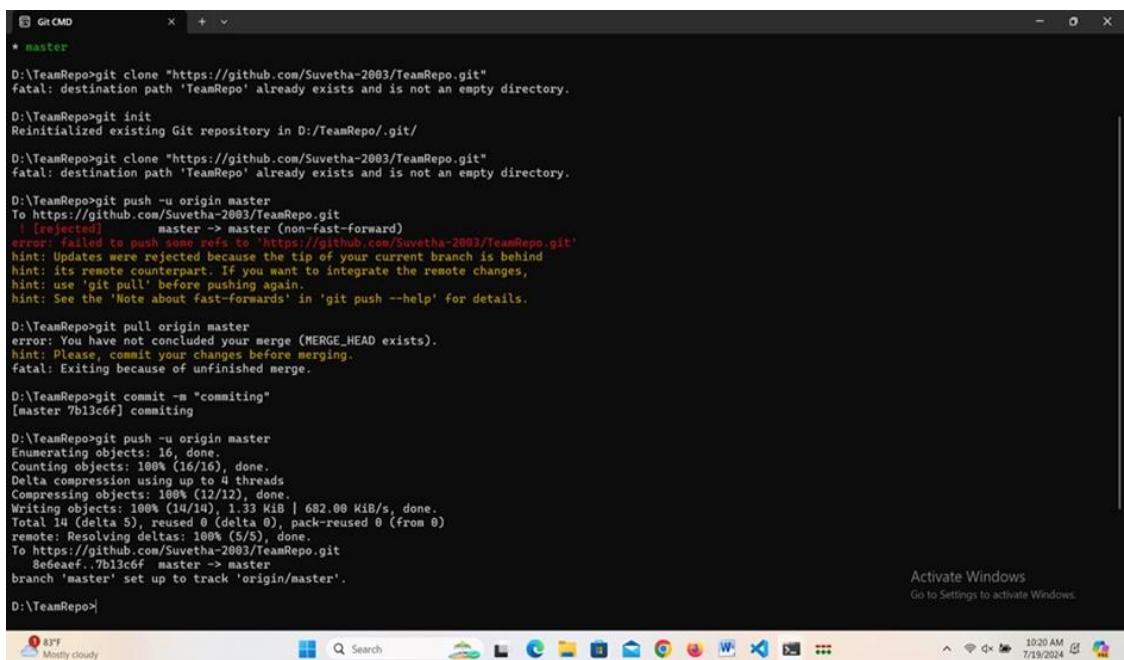
10. Return to master. Merge your branch. Notice how, thanks to the rebase, this is a fastforward merge.



```
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>
```

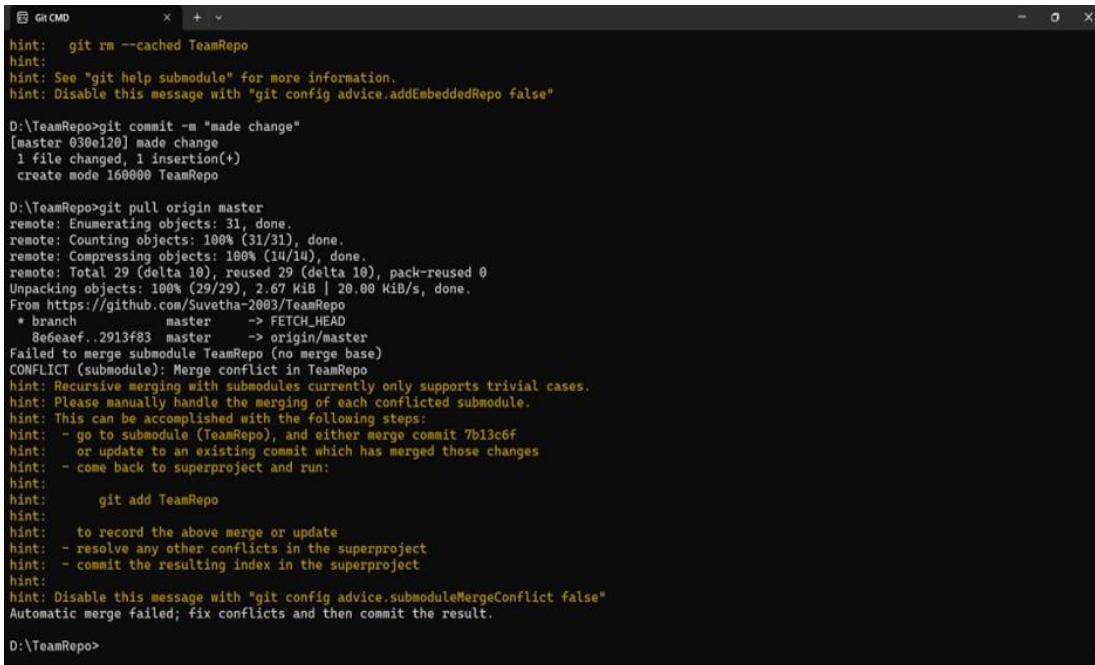
Stretch Task:

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



```
* master
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
  86eaeef..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.

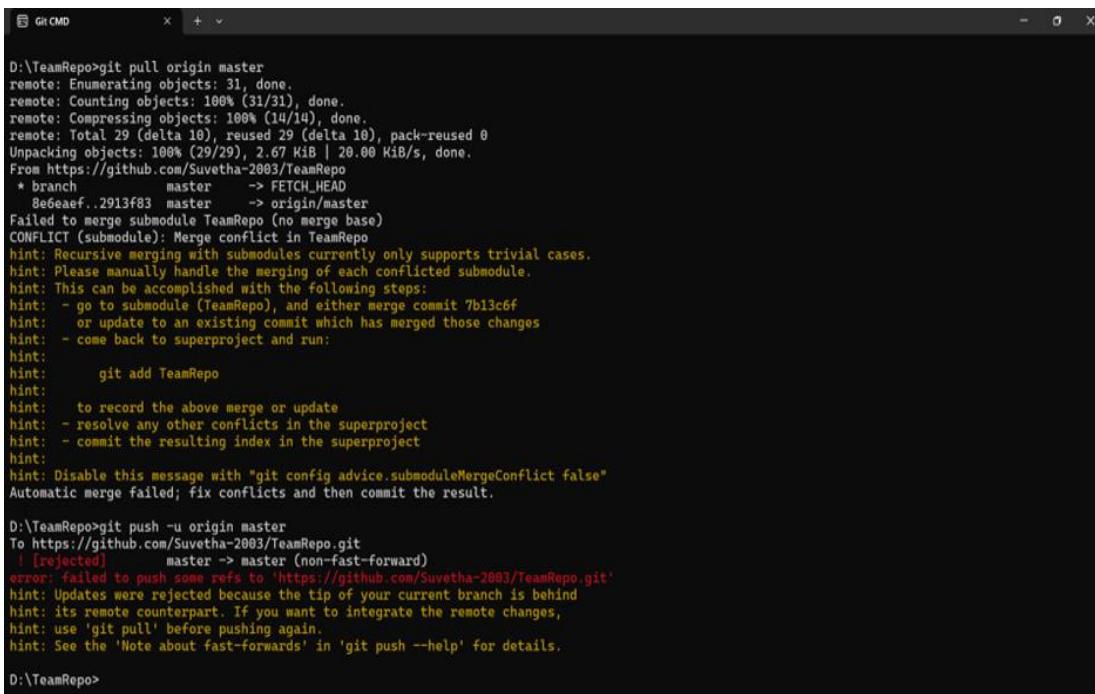


```
hint: git rm --cached TeamRepo
hint:
hint: See "git help submodule" for more information.
hint: Disable this message with "git config advice.addEmbeddedRepo false"
D:\TeamRepo>git commit -m "made change"
[master 030e120] made change
 1 file changed, 1 insertion(+)
 create mode 160000 TeamRepo

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:
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.

```
Git CMD x + v
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
4f1a27d 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:

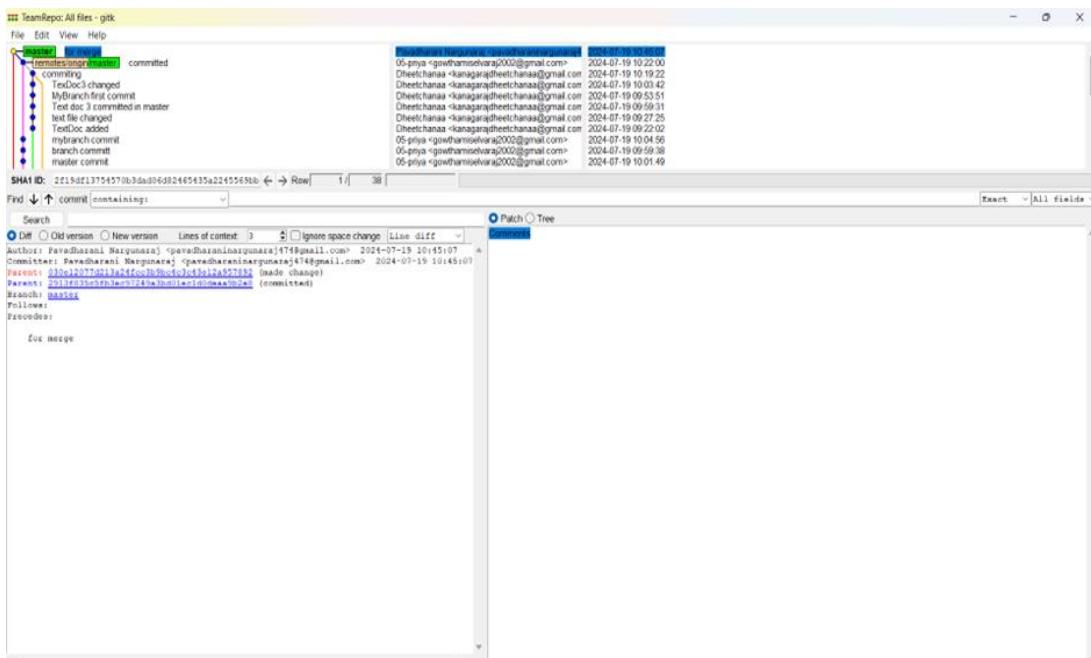
git branch <new-branch-name> b08ee2f

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.



- 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?**

This occurs because the commit history is linear, reflecting the order of integration rather than the temporal order of creation. Git tracks the order of commits based on when they are merged into the repository, not when they are individually made.