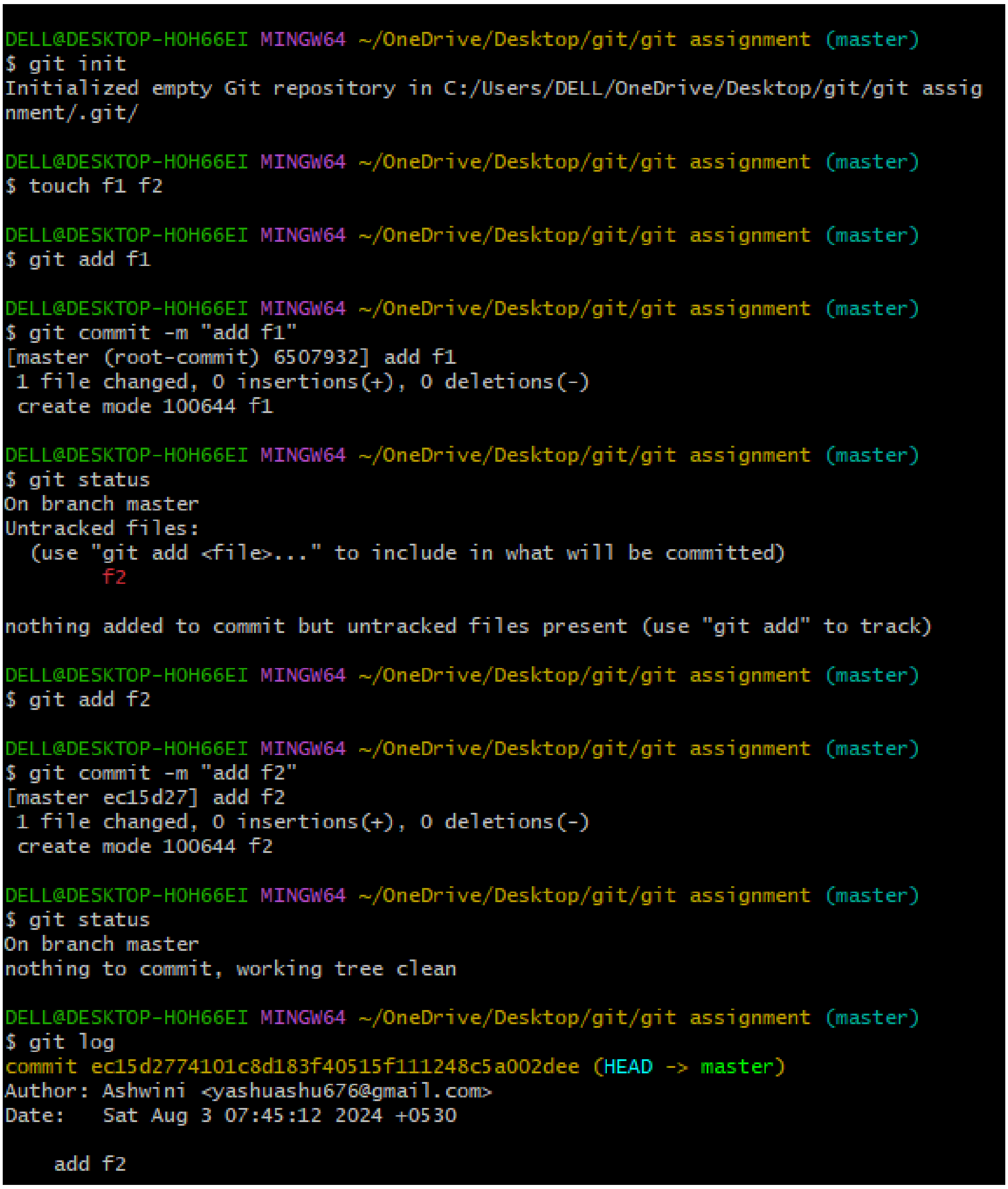
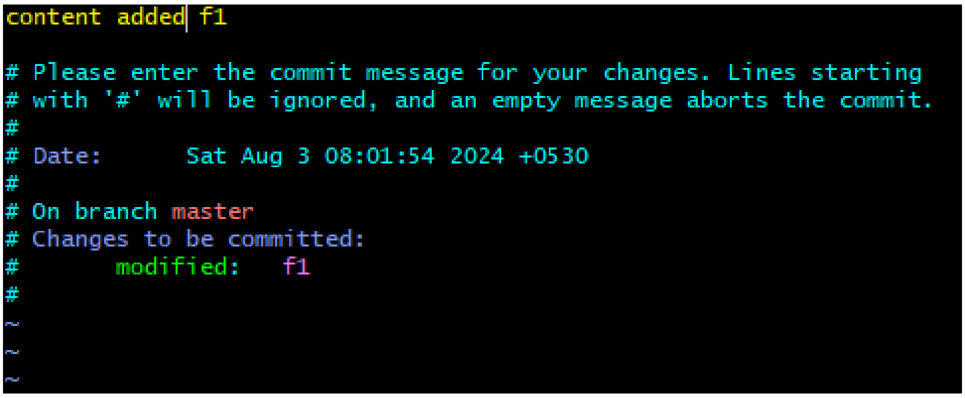
GIT ASSIGNMENT

1. **git init :-** To initialize/open a git repository and a new subfolder .git is created in local directory.
2. **git add :-** To add a file to the staging area.
3. **git commit -m "message" :-** This command commits the staged changes to the local repository
4. **git status :-** It will show whether files are in workspace or staging are on in git repo

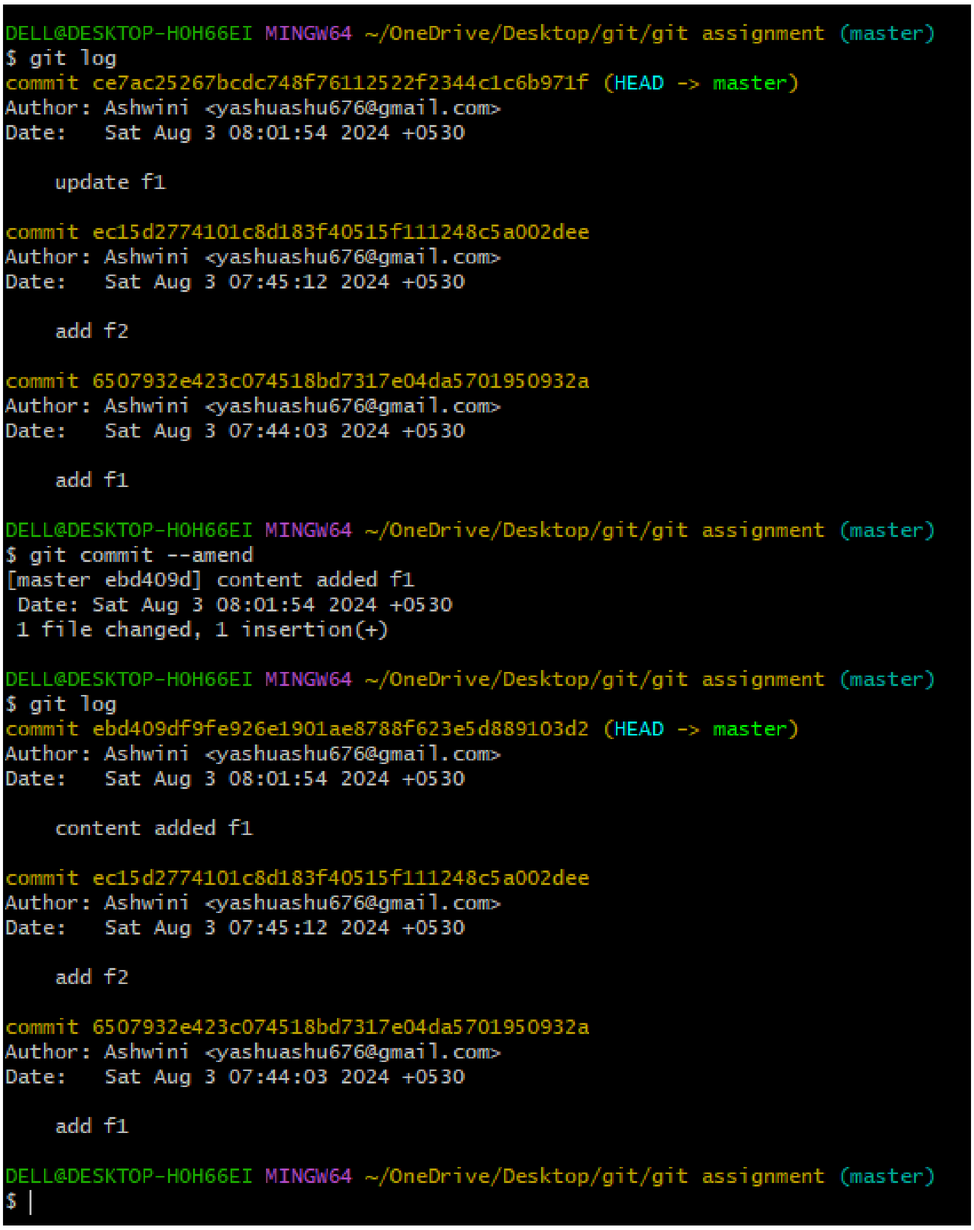


```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git init
Initialized empty Git repository in C:/Users/DELL/OneDrive/Desktop/git/git assig
nment/.git/

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ touch f1 f2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git add f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git commit -m "add f1"
[master (root-commit) 6507932] add f1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        f2

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

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git add f2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git commit -m "add f2"
[master ec15d27] add f2
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git status
On branch master
nothing to commit, working tree clean

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git log
commit ec15d2774101c8d183f40515f111248c5a002dee (HEAD -> master)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 07:45:12 2024 +0530

    add f2
```

5. **git commit --amend :-** It will help you edit the commit message already commited. The old commit is replaced by new commit. It means old commit will not be there

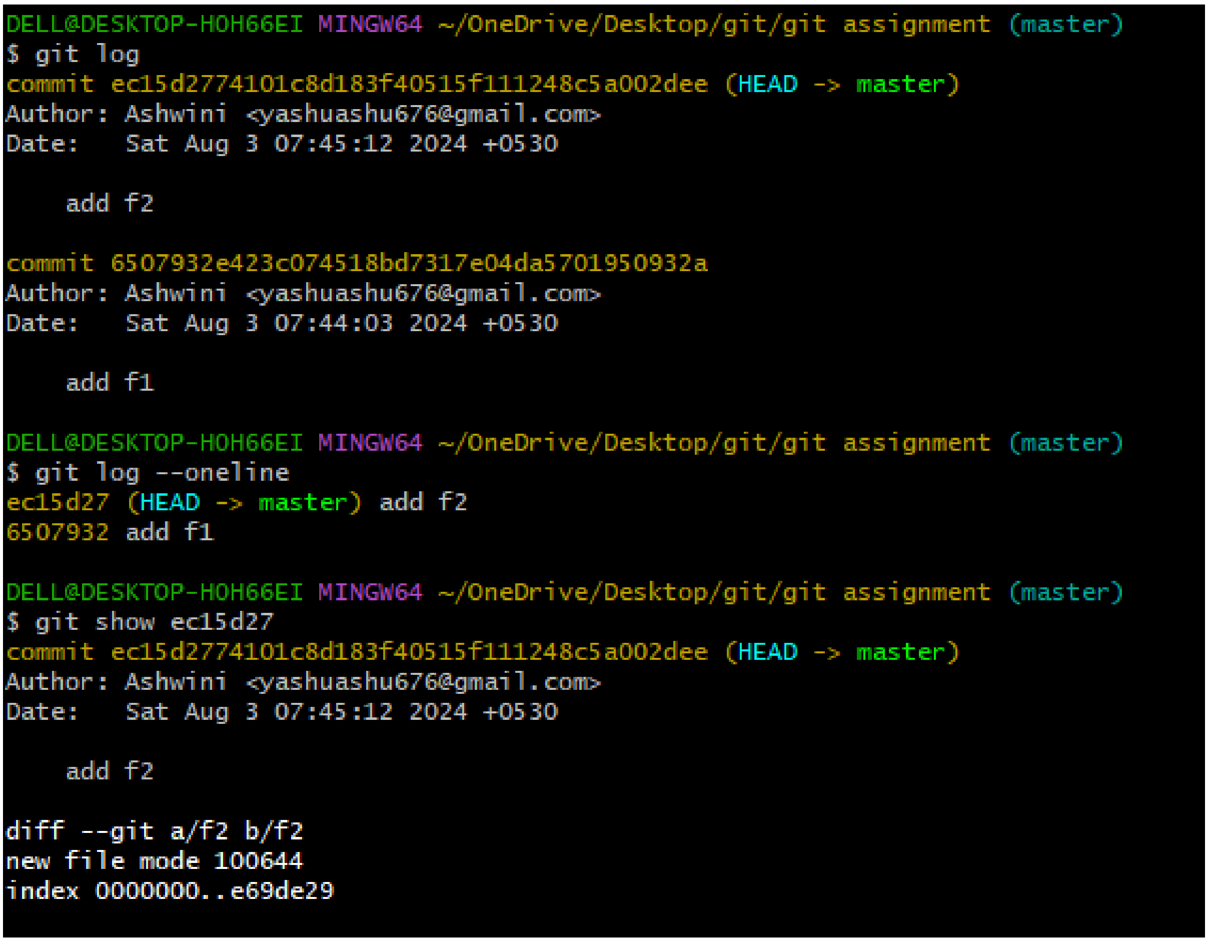


```
content added f1

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Sat Aug 3 08:01:54 2024 +0530
#
# On branch master
# Changes to be committed:
#       modified:   f1
#
~
~
~
```



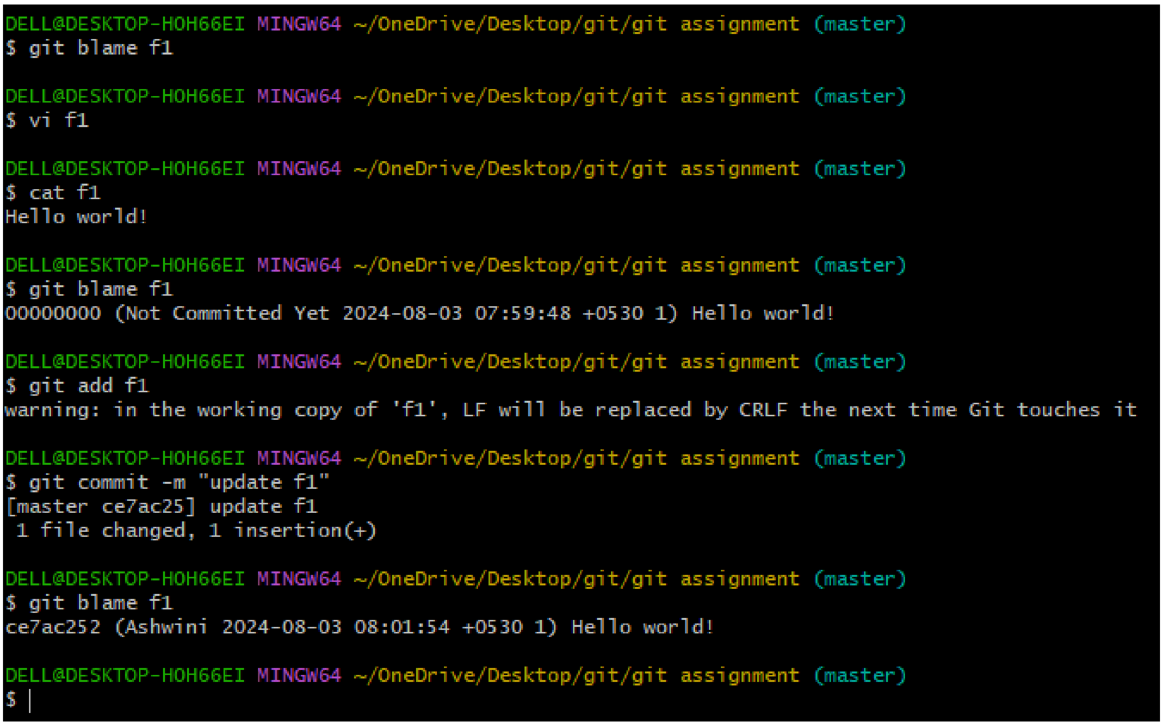
```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git log
commit ce7ac25267bcdc748f76112522f2344c1c6b971f (HEAD -> master)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 08:01:54 2024 +0530

    update f1

commit ec15d2774101c8d183f40515f111248c5a002dee
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 07:45:12 2024 +0530

    add f2

commit 6507932e423c074518bd7317e04da5701950932a
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 07:44:03 2024 +0530

    add f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git commit --amend
[master ebd409d] content added f1
 Date: Sat Aug 3 08:01:54 2024 +0530
 1 file changed, 1 insertion(+)

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git log
commit ebd409df9fe926e1901ae8788f623e5d889103d2 (HEAD -> master)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 08:01:54 2024 +0530

    content added f1

commit ec15d2774101c8d183f40515f111248c5a002dee
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 07:45:12 2024 +0530

    add f2

commit 6507932e423c074518bd7317e04da5701950932a
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 07:44:03 2024 +0530

    add f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ 
```

6. **git log :-** To list/check the committed file or commit history
7. **git log --oneline :-** To list all the commit in column format (single column)
8. **git show :-** To display detailed information about a specific commit, including the commit message, author, date, and the actual changes made to the files.
9. **git show <commit_ID> :-** To know the modifications in the last commit(prints the line)



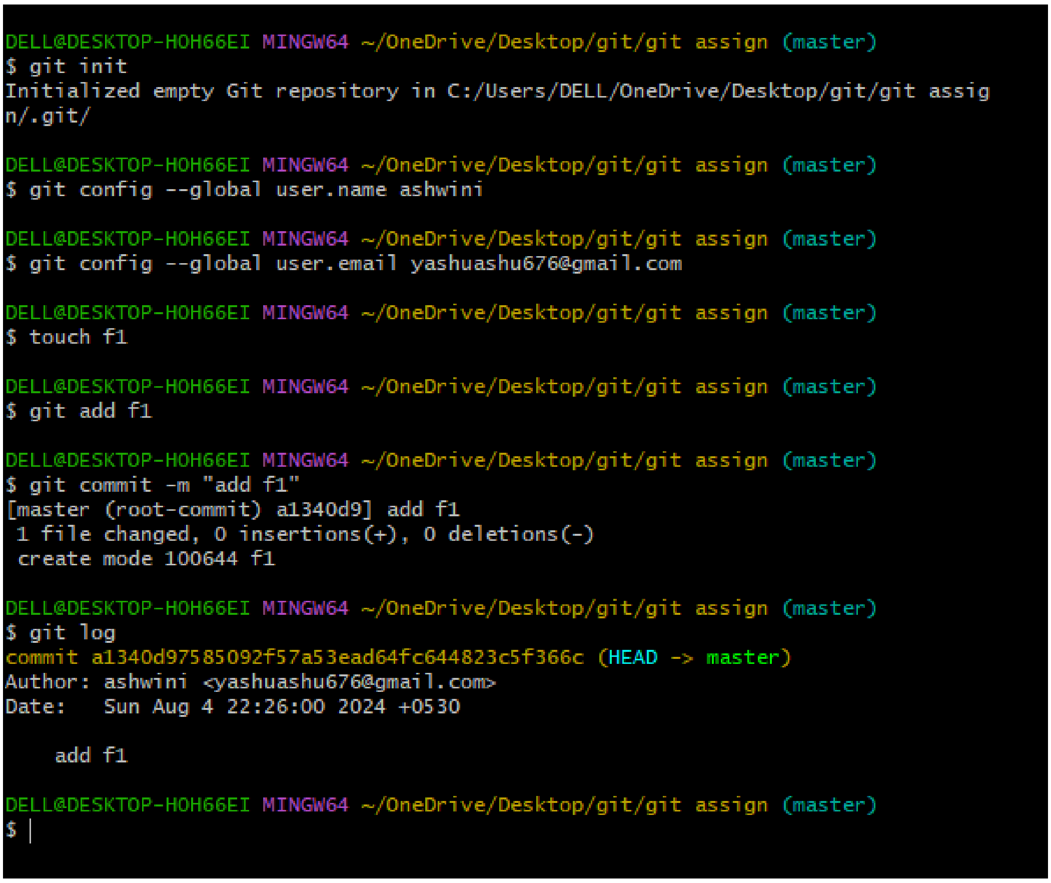
```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git log
commit ec15d2774101c8d183f40515f111248c5a002dee (HEAD -> master)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 07:45:12 2024 +0530

    add f2

commit 6507932e423c074518bd7317e04da5701950932a
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 07:44:03 2024 +0530

    add f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git log --oneline
ec15d27 (HEAD -> master) add f2
6507932 add f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git show ec15d27
commit ec15d2774101c8d183f40515f111248c5a002dee (HEAD -> master)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 07:45:12 2024 +0530

    add f2

diff --git a/f2 b/f2
new file mode 100644
index 0000000..e69de29
```

10. **git blame <filename> :-** To show the last modification for each line in a file, it will show who has modified the code(eachline of the code)



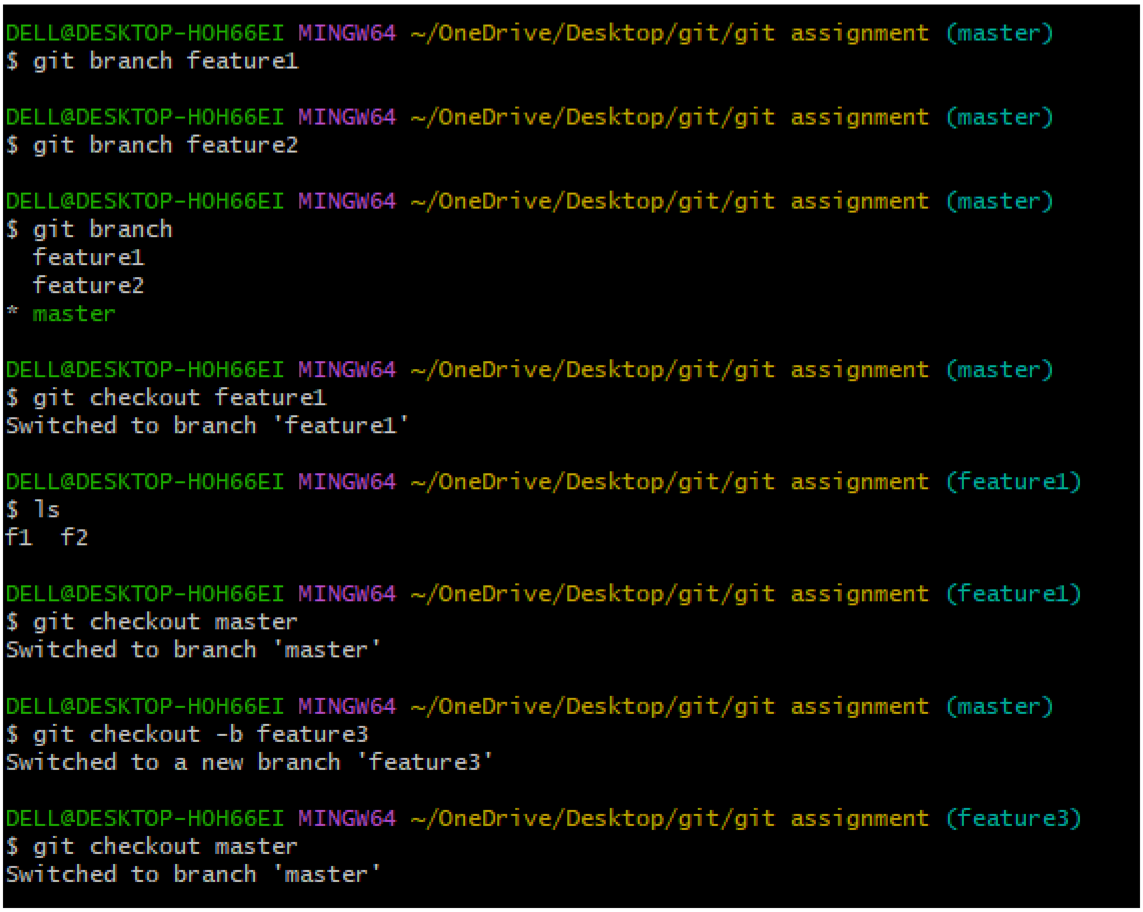
```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git blame f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ vi f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ cat f1
Hello world!

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git blame f1
00000000 (Not Committed Yet 2024-08-03 07:59:48 +0530 1) Hello world!

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git add f1
warning: in the working copy of 'f1', LF will be replaced by CRLF the next time Git touches it

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git commit -m "update f1"
[master ce7ac25] update f1
 1 file changed, 1 insertion(+)

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git blame f1
ce7ac252 (Ashwini 2024-08-03 08:01:54 +0530 1) Hello world!

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ |
```

11. **git config --global user.name <username>** :- To update the user name.
12. **git config --global user.email <email>** :- To update the email in the terminal.

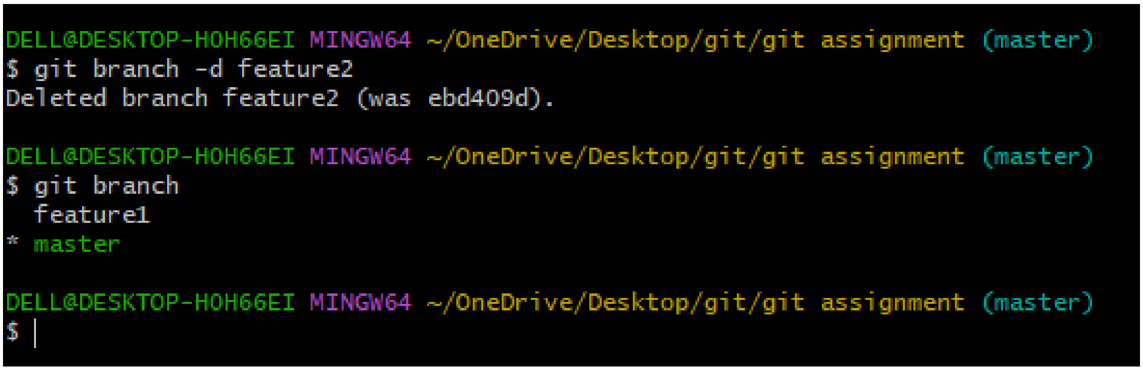


```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git init
Initialized empty Git repository in C:/Users/DELL/OneDrive/Desktop/git/git assig
n/.git/

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git config --global user.name ashwini

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git config --global user.email yashuashu676@gmail.com

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ touch f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git add f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git commit -m "add f1"
[master (root-commit) a1340d9] add f1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log
commit a1340d97585092f57a53ead64fc644823c5f366c (HEAD -> master)
Author: ashwini <yashuashu676@gmail.com>
Date:   Sun Aug 4 22:26:00 2024 +0530

    add f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ |
```

13. **git branch <name> :-** To create a branch.
14. **git branch :-** To list the branchs.
15. **git checkout <branch name> :-** To switch to other branch.
16. **git checkout -b <branch name> :-** To create new branch and switch to new branch.
17. **git branch -d <branch name> :-** To delete a branch.

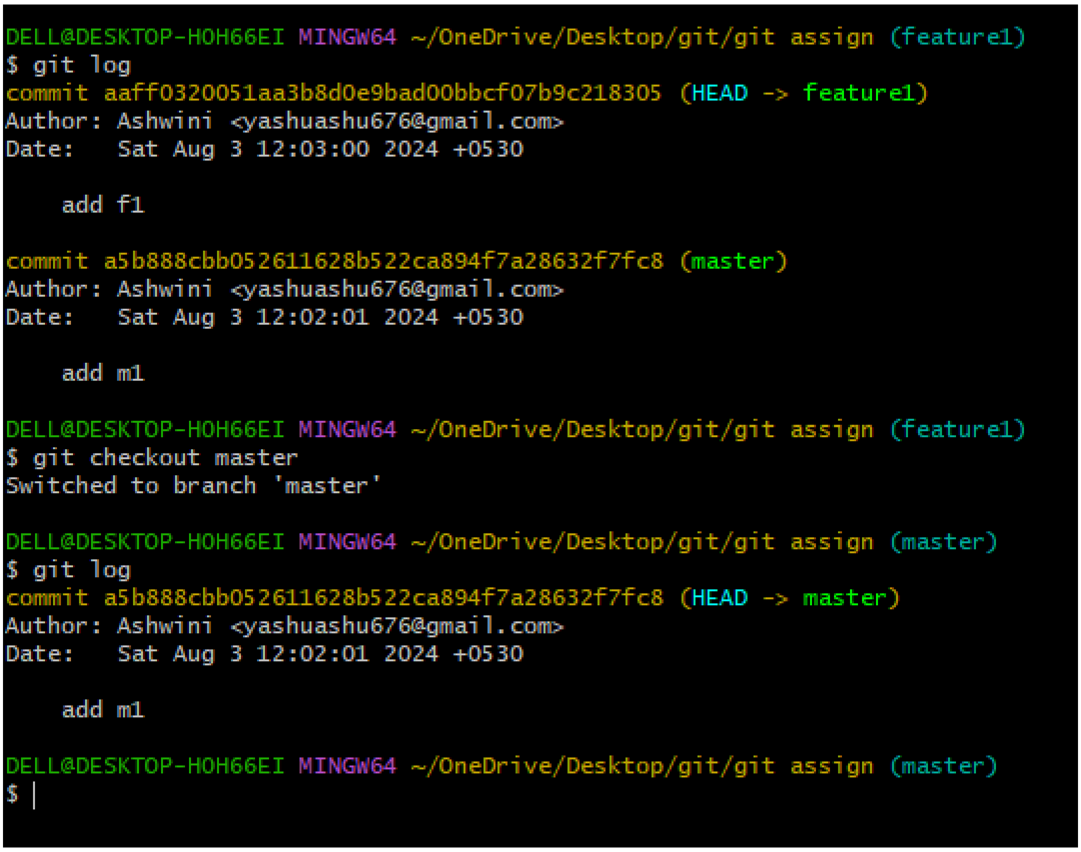


```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git branch feature1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git branch feature2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git branch
  feature1
  feature2
* master

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git checkout feature1
Switched to branch 'feature1'

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (feature1)
$ ls
f1  f2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (feature1)
$ git checkout master
Switched to branch 'master'

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git checkout -b feature3
Switched to a new branch 'feature3'

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (feature3)
$ git checkout master
Switched to branch 'master'
```

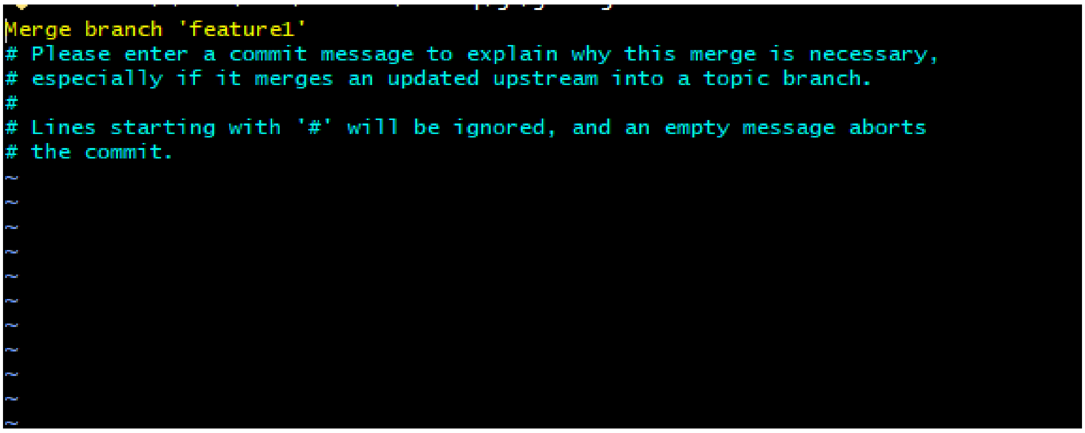


```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git branch -d feature2
Deleted branch feature2 (was ebd409d).

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git branch
  feature1
* master

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ |
```

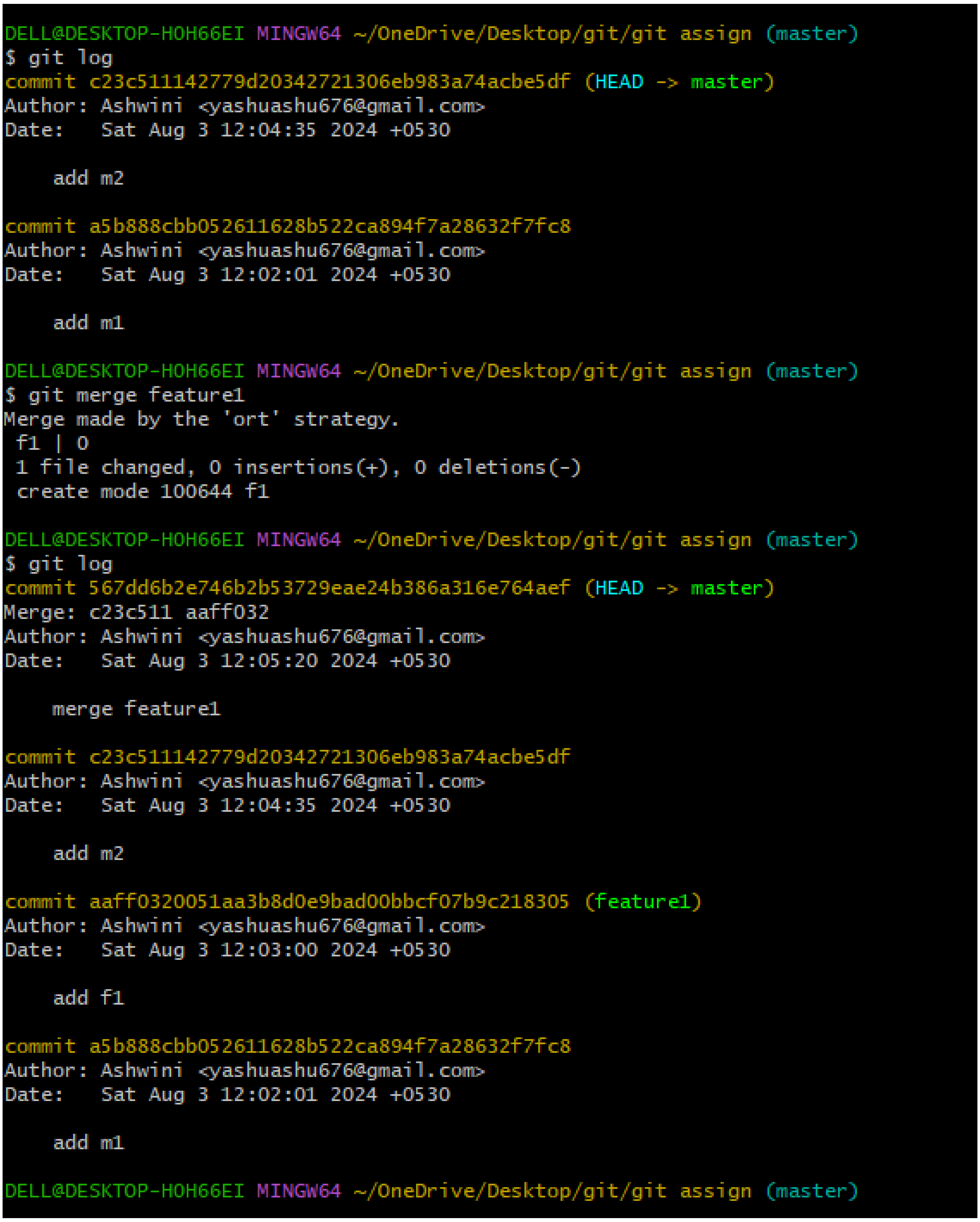
18. git merge <branch name> :- This command will merge current working directory with the specified branch, creates merge commit



```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git log
commit aaff0320051aa3b8d0e9bad00bbcf07b9c218305 (HEAD -> feature1)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:03:00 2024 +0530

    add f1

commit a5b888cbb052611628b522ca894f7a28632f7fc8 (master)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:02:01 2024 +0530

    add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git checkout master
Switched to branch 'master'

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log
commit a5b888cbb052611628b522ca894f7a28632f7fc8 (HEAD -> master)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:02:01 2024 +0530

    add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ |
```

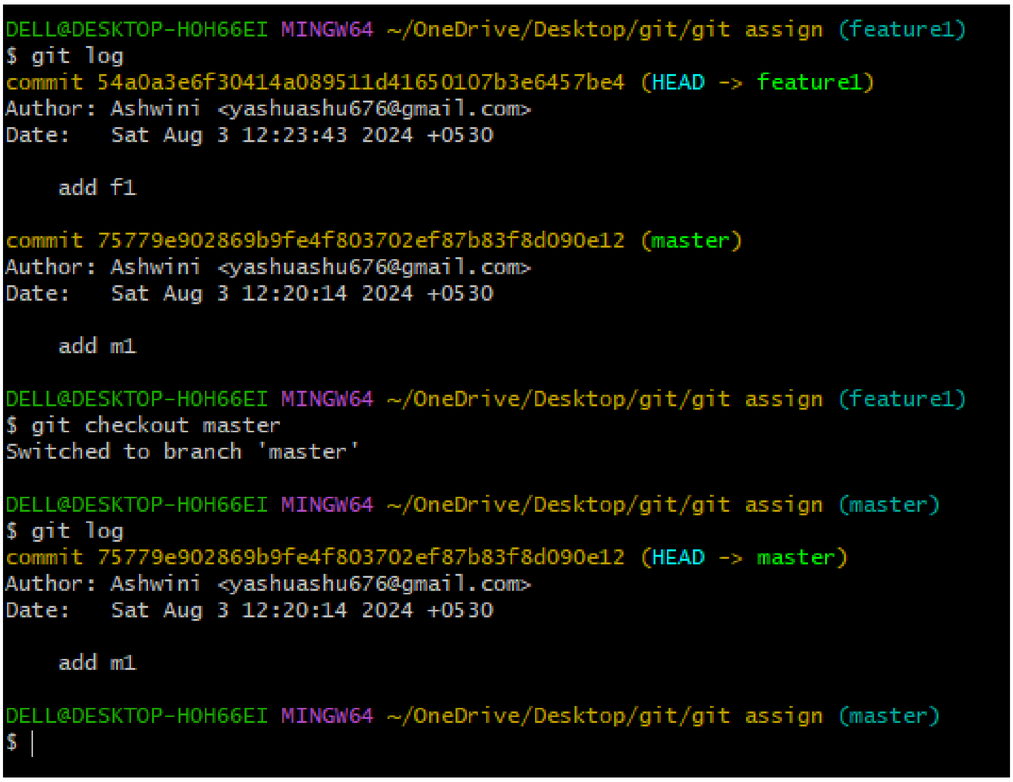


```
Merge branch 'feature1'
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
~
~
~
~
~
~
~
~
~
~
~
```

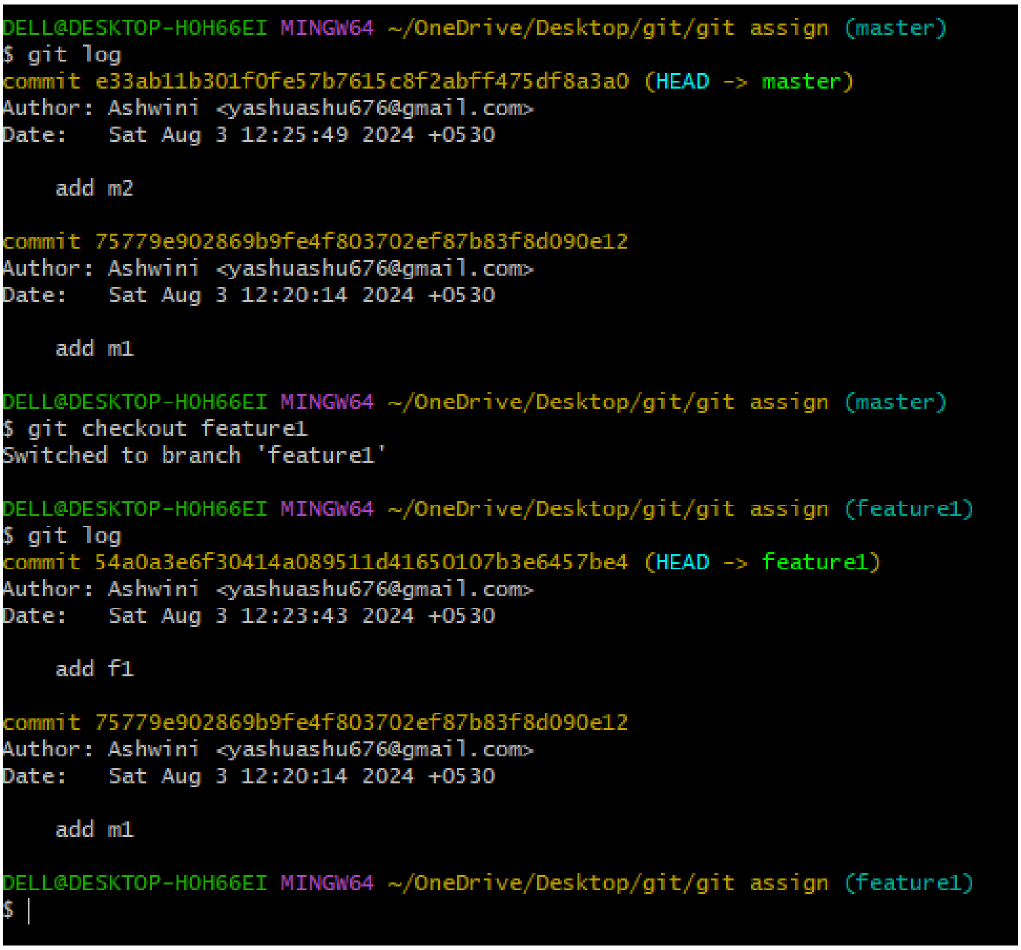


```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log
commit c23c511142779d20342721306eb983a74acbe5df (HEAD -> master)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:04:35 2024 +0530

    add m2

commit a5b888cbb052611628b522ca894f7a28632f7fc8
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:02:01 2024 +0530

    add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git merge feature1
Merge made by the 'ort' strategy.
 f1 | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log
commit 567dd6b2e746b2b53729eae24b386a316e764aef (HEAD -> master)
Merge: c23c511 aaff032
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:05:20 2024 +0530

    merge feature1

commit c23c511142779d20342721306eb983a74acbe5df
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:04:35 2024 +0530

    add m2

commit aaff0320051aa3b8d0e9bad00bbcf07b9c218305 (feature1)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:03:00 2024 +0530

    add f1

commit a5b888cbb052611628b522ca894f7a28632f7fc8
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:02:01 2024 +0530

    add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
```

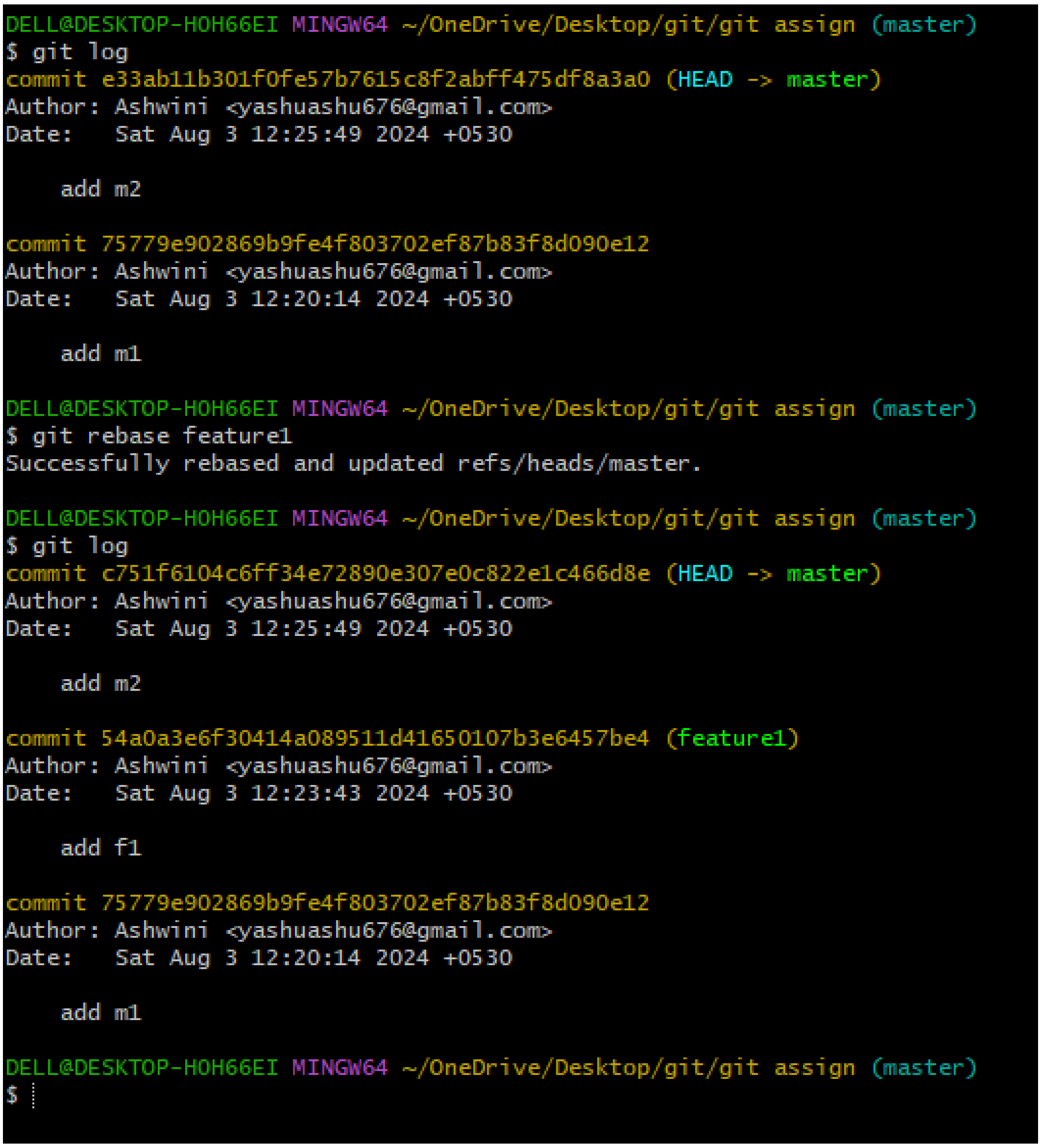
19. git rebase <branch name> :- It is nothing but a merge. one branch will get added to the tip of the branch. (rewrites the commit history).



```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git log
commit 54a0a3e6f30414a089511d41650107b3e6457be4 (HEAD -> feature1)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:23:43 2024 +0530

    add f1

commit 75779e902869b9fe4f803702ef87b83f8d090e12 (master)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:20:14 2024 +0530

    add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git checkout master
Switched to branch 'master'

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log
commit 75779e902869b9fe4f803702ef87b83f8d090e12 (HEAD -> master)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:20:14 2024 +0530

    add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ |
```



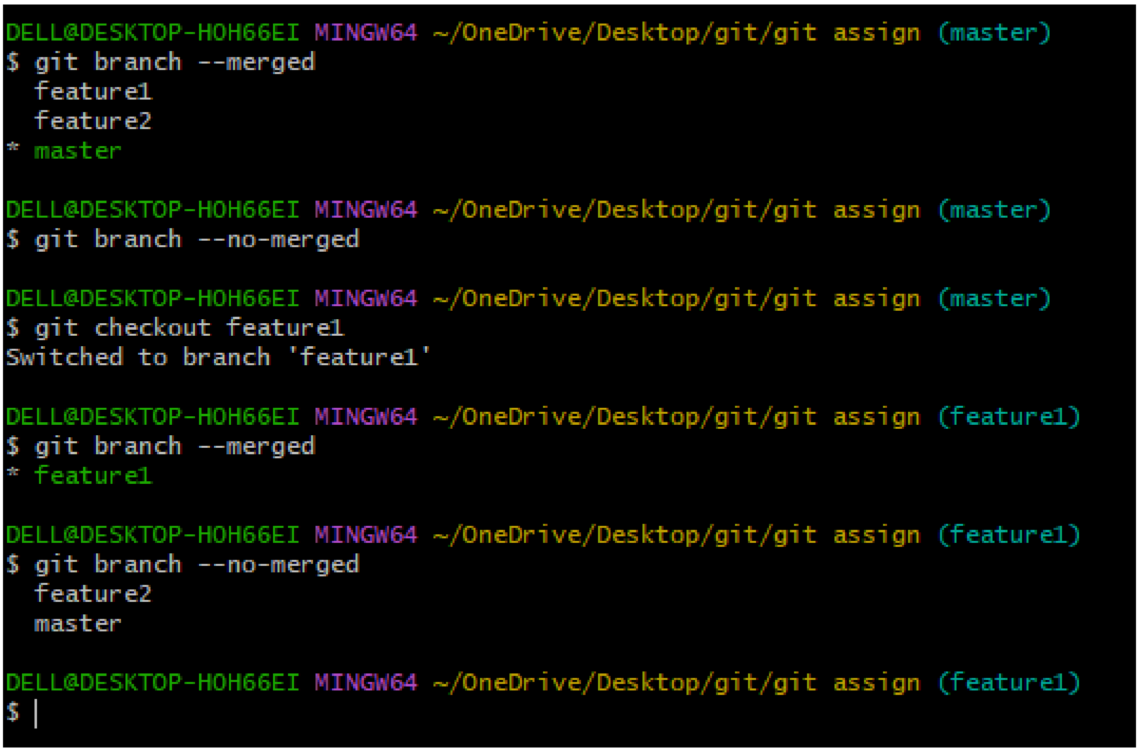
```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log
commit e33ab11b301f0fe57b7615c8f2abff475df8a3a0 (HEAD -> master)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:25:49 2024 +0530

    add m2

commit 75779e902869b9fe4f803702ef87b83f8d090e12
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:20:14 2024 +0530

    add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git checkout feature1
Switched to branch 'feature1'

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git log
commit 54a0a3e6f30414a089511d41650107b3e6457be4 (HEAD -> feature1)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:23:43 2024 +0530

    add f1

commit 75779e902869b9fe4f803702ef87b83f8d090e12
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:20:14 2024 +0530

    add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ |
```



```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log
commit e33ab11b301f0fe57b7615c8f2abff475df8a3a0 (HEAD -> master)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:25:49 2024 +0530

    add m2

commit 75779e902869b9fe4f803702ef87b83f8d090e12
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:20:14 2024 +0530

    add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git rebase feature1
Successfully rebased and updated refs/heads/master.

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log
commit c751f6104c6ff34e72890e307e0c822e1c466d8e (HEAD -> master)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:25:49 2024 +0530

    add m2

commit 54a0a3e6f30414a089511d41650107b3e6457be4 (feature1)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:23:43 2024 +0530

    add f1

commit 75779e902869b9fe4f803702ef87b83f8d090e12
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 12:20:14 2024 +0530

    add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$
```

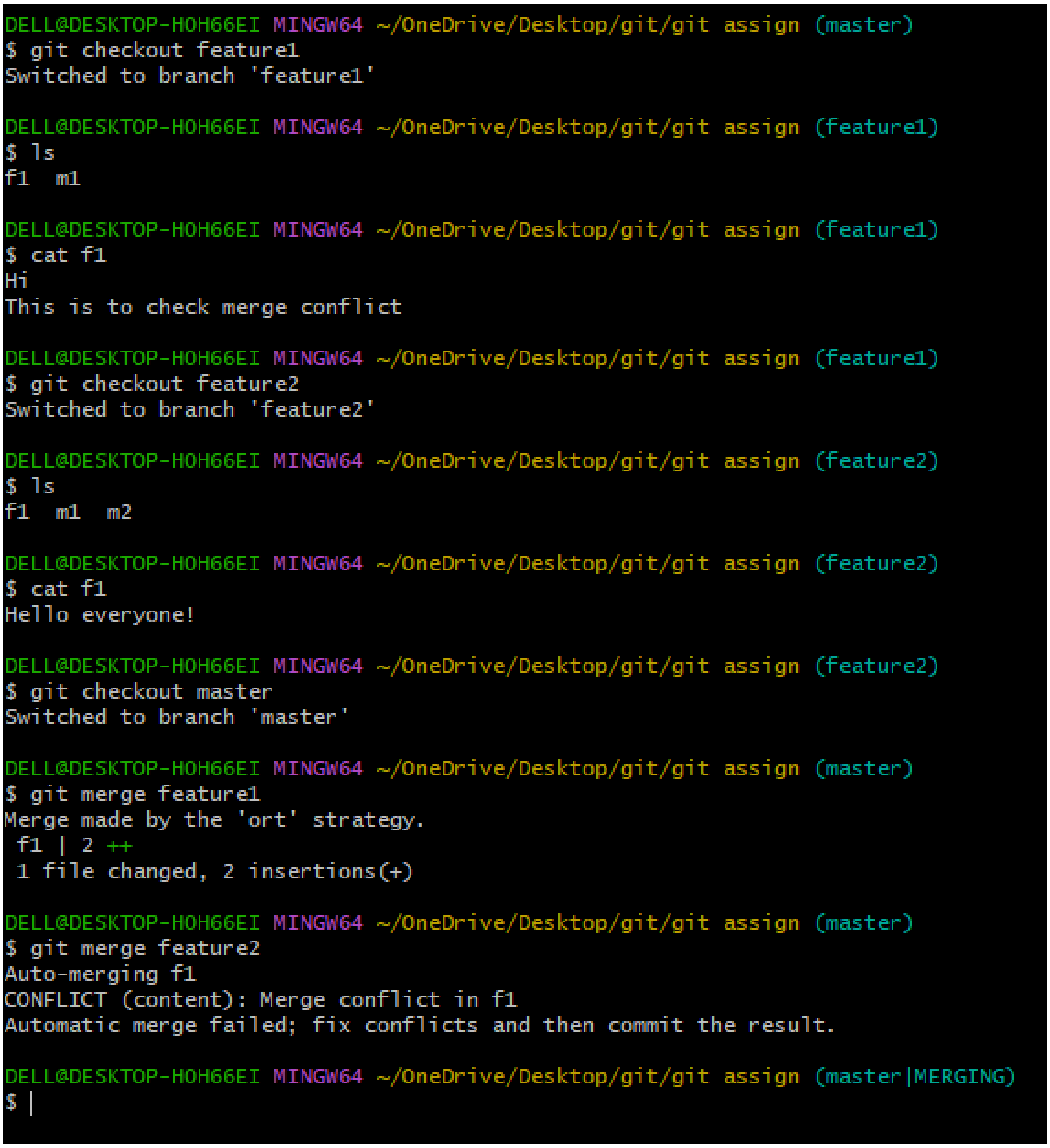
20. **git branch -merged :-** To check the merged branches.

21. **git branch --no-merged:-** To check non merged branches

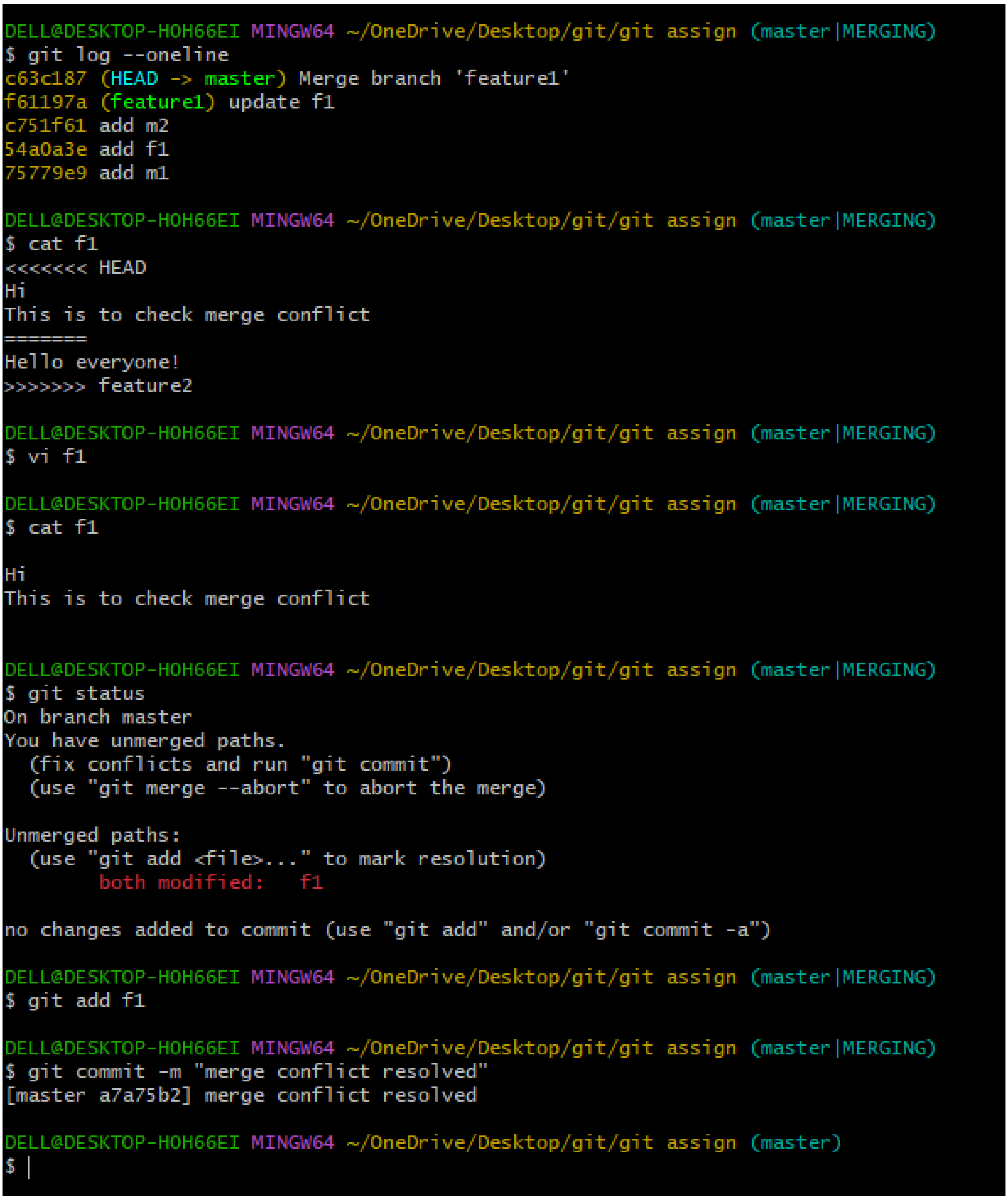


```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git branch --merged
  feature1
  feature2
* master

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git branch --no-merged

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git checkout feature1
Switched to branch 'feature1'

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git branch --merged
* feature1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git branch --no-merged
  feature2
  master

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ |
```

22. **git conflict:-** A piece of code work parralelly in 2 branches, when they merge conflict arise



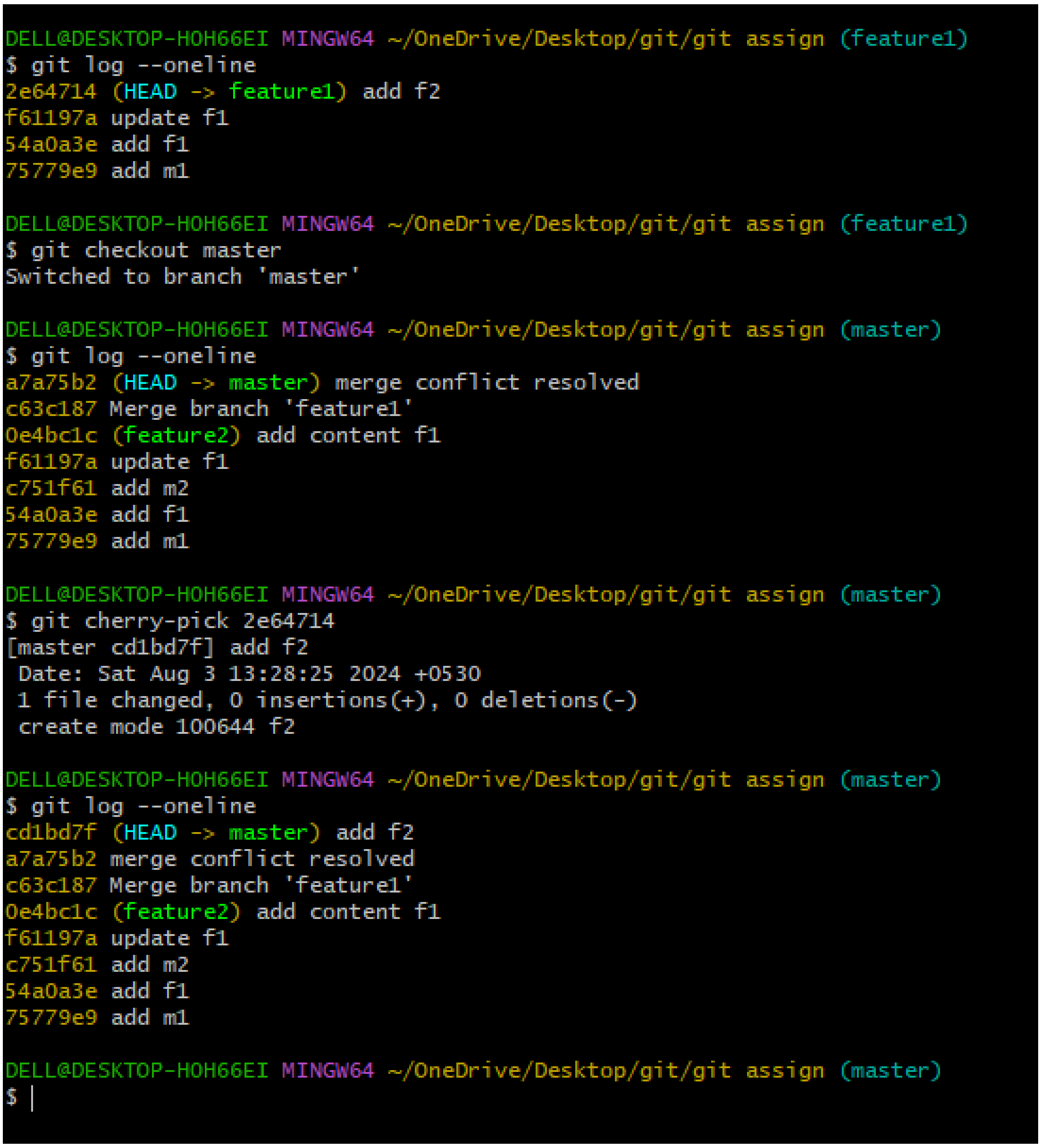
```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git checkout feature1
Switched to branch 'feature1'

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ ls
f1  m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ cat f1
Hi
This is to check merge conflict

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git checkout feature2
Switched to branch 'feature2'

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature2)
$ ls
f1  m1  m2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature2)
$ cat f1
Hello everyone!

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature2)
$ git checkout master
Switched to branch 'master'

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git merge feature1
Merge made by the 'ort' strategy.
 f1 | 2 ++
 1 file changed, 2 insertions(+)

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git merge feature2
Auto-merging f1
CONFLICT (content): Merge conflict in f1
Automatic merge failed; fix conflicts and then commit the result.

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master|MERGING)
$ |
```



```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master|MERGING)
$ git log --oneline
c63c187 (HEAD -> master) Merge branch 'feature1'
f61197a (feature1) update f1
c751f61 add m2
54a0a3e add f1
75779e9 add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master|MERGING)
$ cat f1
<<<<<<< HEAD
Hi
This is to check merge conflict
=======
Hello everyone!
>>>>>>> feature2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master|MERGING)
$ vi f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master|MERGING)
$ cat f1

Hi
This is to check merge conflict


DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   f1

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

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master|MERGING)
$ git add f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master|MERGING)
$ git commit -m "merge conflict resolved"
[master a7a75b2] merge conflict resolved

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ |
```

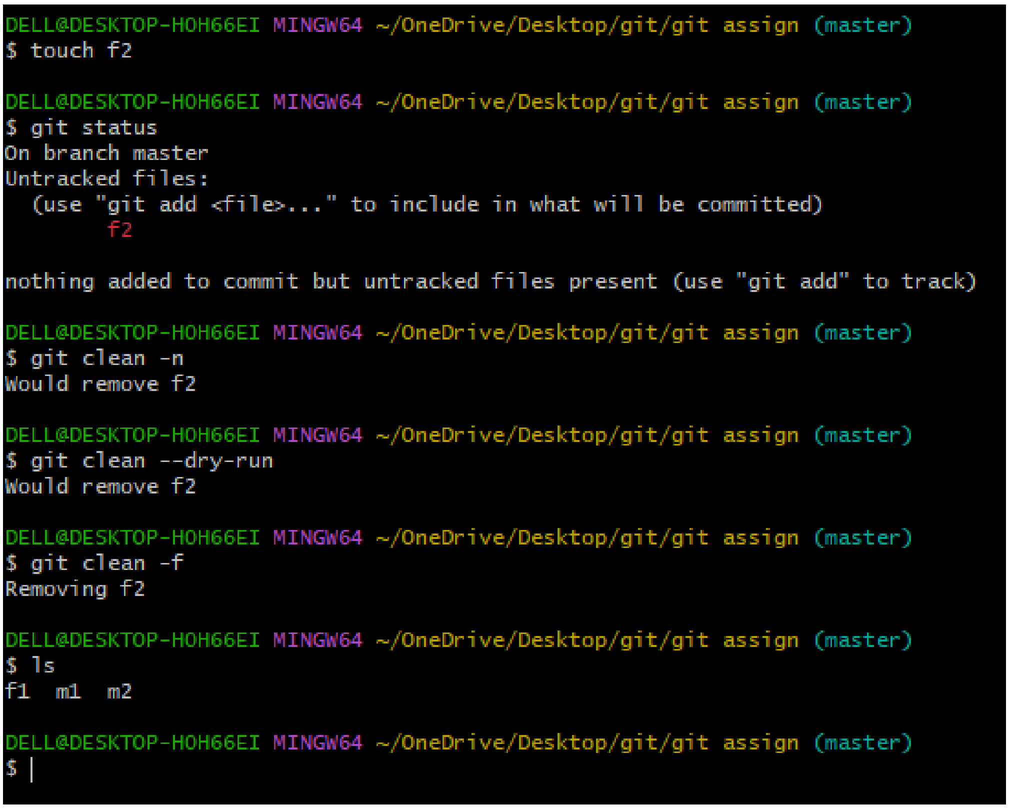
Resolved merge conflict

23. git cherry-pick <commit ID>:- to merge a particular file/code from a branch to master/branch with new commit.



```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git log --oneline
2e64714 (HEAD -> feature1) add f2
f61197a update f1
54a0a3e add f1
75779e9 add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git checkout master
Switched to branch 'master'

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log --oneline
a7a75b2 (HEAD -> master) merge conflict resolved
c63c187 Merge branch 'feature1'
0e4bc1c (feature2) add content f1
f61197a update f1
c751f61 add m2
54a0a3e add f1
75779e9 add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git cherry-pick 2e64714
[master cd1bd7f] add f2
 Date: Sat Aug 3 13:28:25 2024 +0530
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 f2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log --oneline
cd1bd7f (HEAD -> master) add f2
a7a75b2 merge conflict resolved
c63c187 Merge branch 'feature1'
0e4bc1c (feature2) add content f1
f61197a update f1
c751f61 add m2
54a0a3e add f1
75779e9 add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ |
```

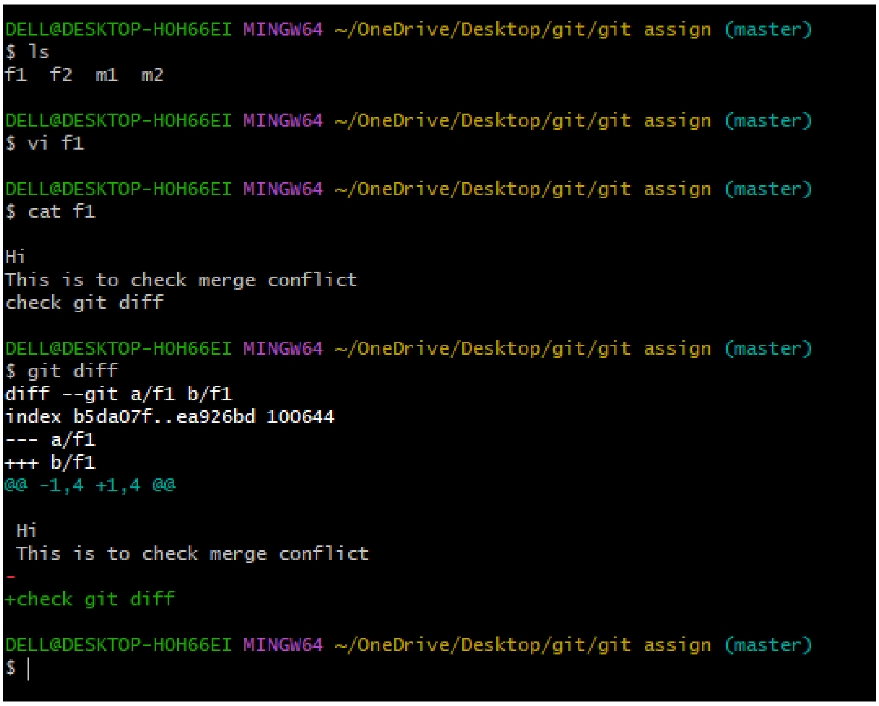
24. Git clean -f :- To delete all untracked files.



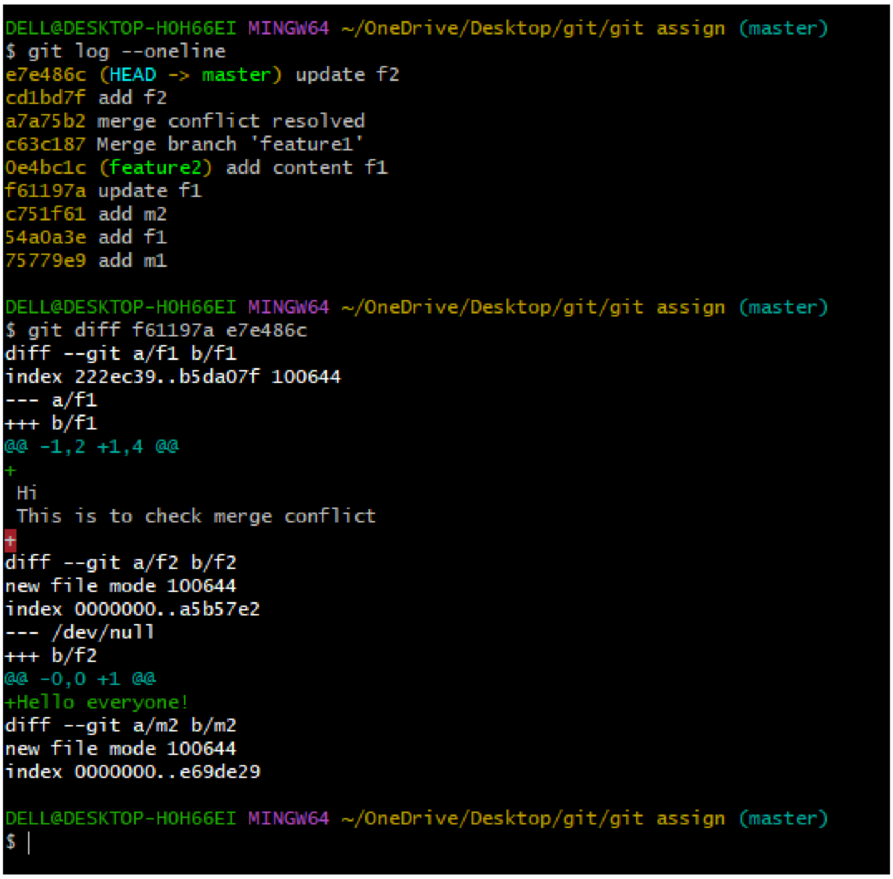
```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ touch f2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        f2

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

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git clean -n
Would remove f2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git clean --dry-run
Would remove f2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git clean -f
Removing f2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ ls
f1  m1  m2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ |
```

25. Git diff :- The git diff command shows the differences between the files in two commits or between your current repository and a previous commit.

26. git diff <commitid> <commitid> or git diff CID..CID :- put starting 7 char to check the difference b/w commit id's

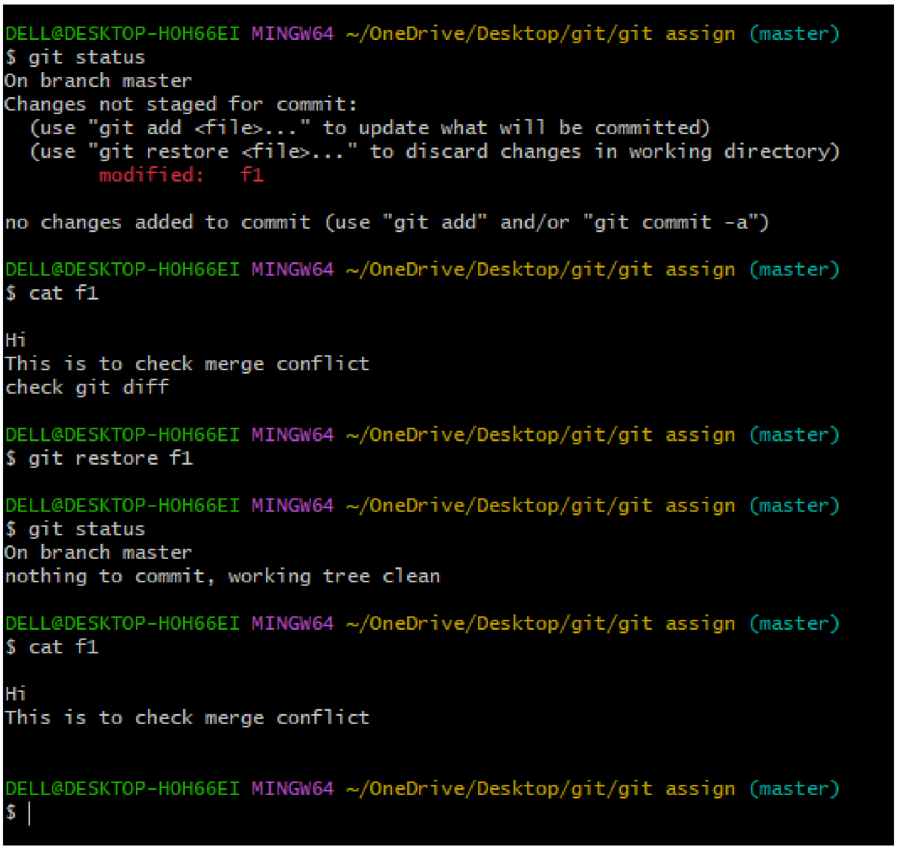


```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ ls
f1  f2  m1  m2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ vi f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ cat f1

Hi
This is to check merge conflict
check git diff

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git diff
diff --git a/f1 b/f1
index b5da07f..ea926bd 100644
--- a/f1
+++ b/f1
@@ -1,4 +1,4 @@

 Hi
 This is to check merge conflict
-
+check git diff

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ |
```



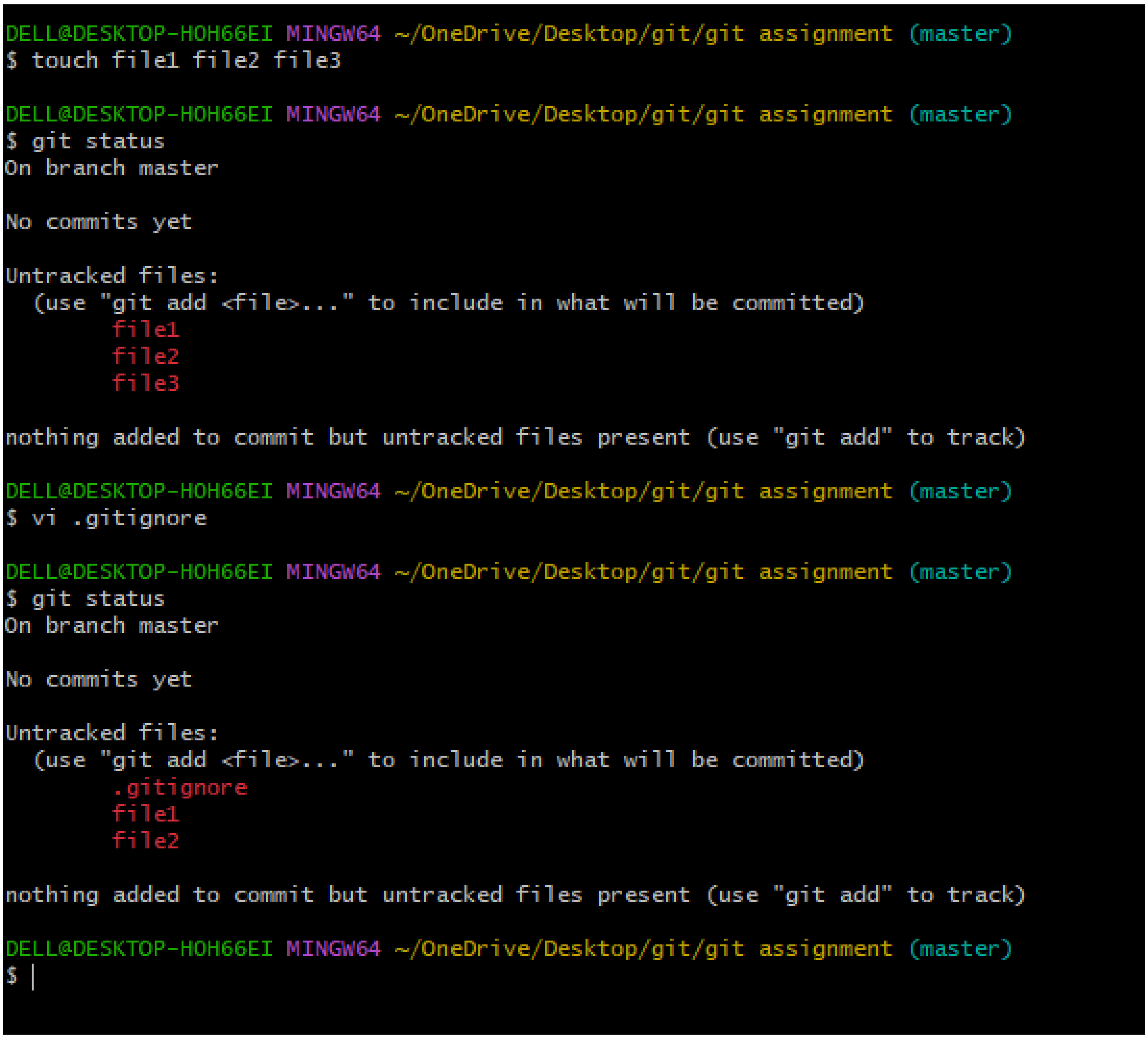
```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log --oneline
e7e486c (HEAD -> master) update f2
cd1bd7f add f2
a7a75b2 merge conflict resolved
c63c187 Merge branch 'feature1'
0e4bc1c (feature2) add content f1
f61197a update f1
c751f61 add m2
54a0a3e add f1
75779e9 add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git diff f61197a e7e486c
diff --git a/f1 b/f1
index 222ec39..b5da07f 100644
--- a/f1
+++ b/f1
@@ -1,2 +1,4 @@
+
 Hi
 This is to check merge conflict
+
diff --git a/f2 b/f2
new file mode 100644
index 0000000..a5b57e2
--- /dev/null
+++ b/f2
@@ -0,0 +1 @@
+Hello everyone!
diff --git a/m2 b/m2
new file mode 100644
index 0000000..e69de29

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ |
```

27. **git restore :-** To undo the the latest changes done to the file.

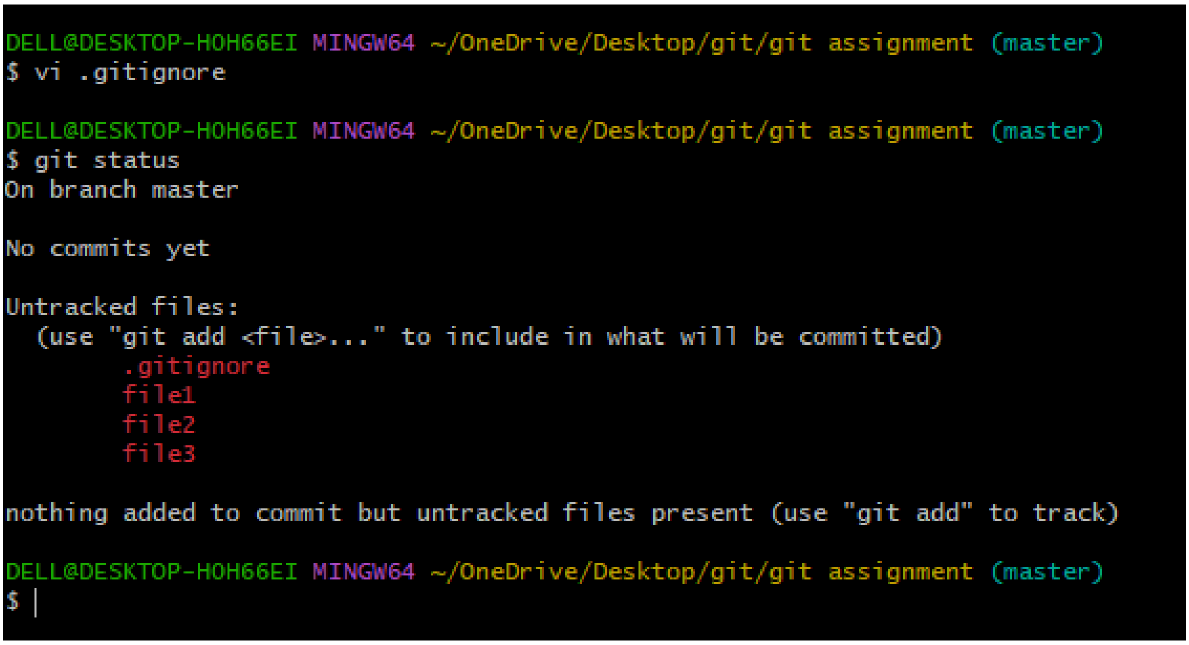


```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   f1

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

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ cat f1

Hi
This is to check merge conflict
check git diff

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git restore f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git status
On branch master
nothing to commit, working tree clean

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ cat f1

Hi
This is to check merge conflict


DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ |
```

28. **.gitignore** :- It specifies which files and dir should be ignored or excluded for staging

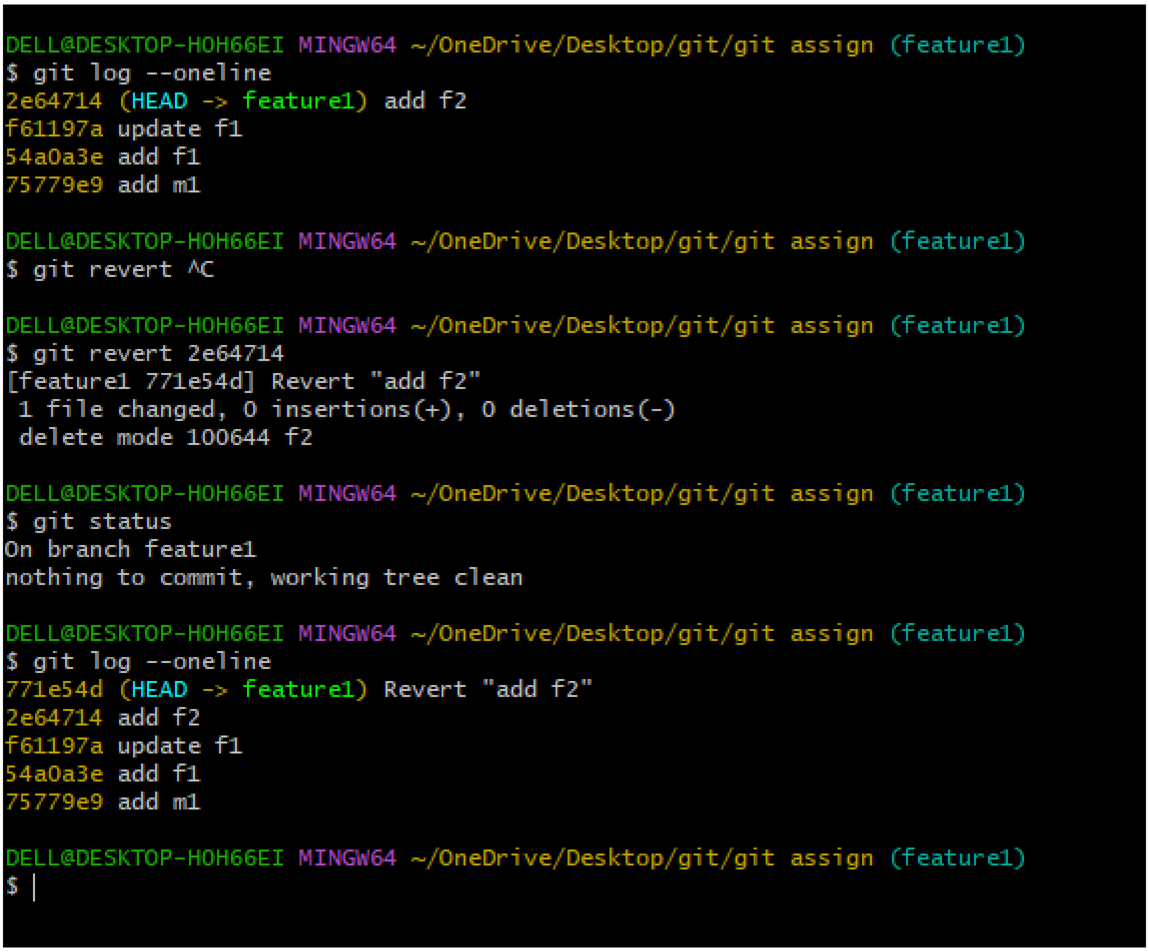


```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ touch file1 file2 file3

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file1
        file2
        file3

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

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ vi .gitignore

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        file1
        file2

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

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ |
```



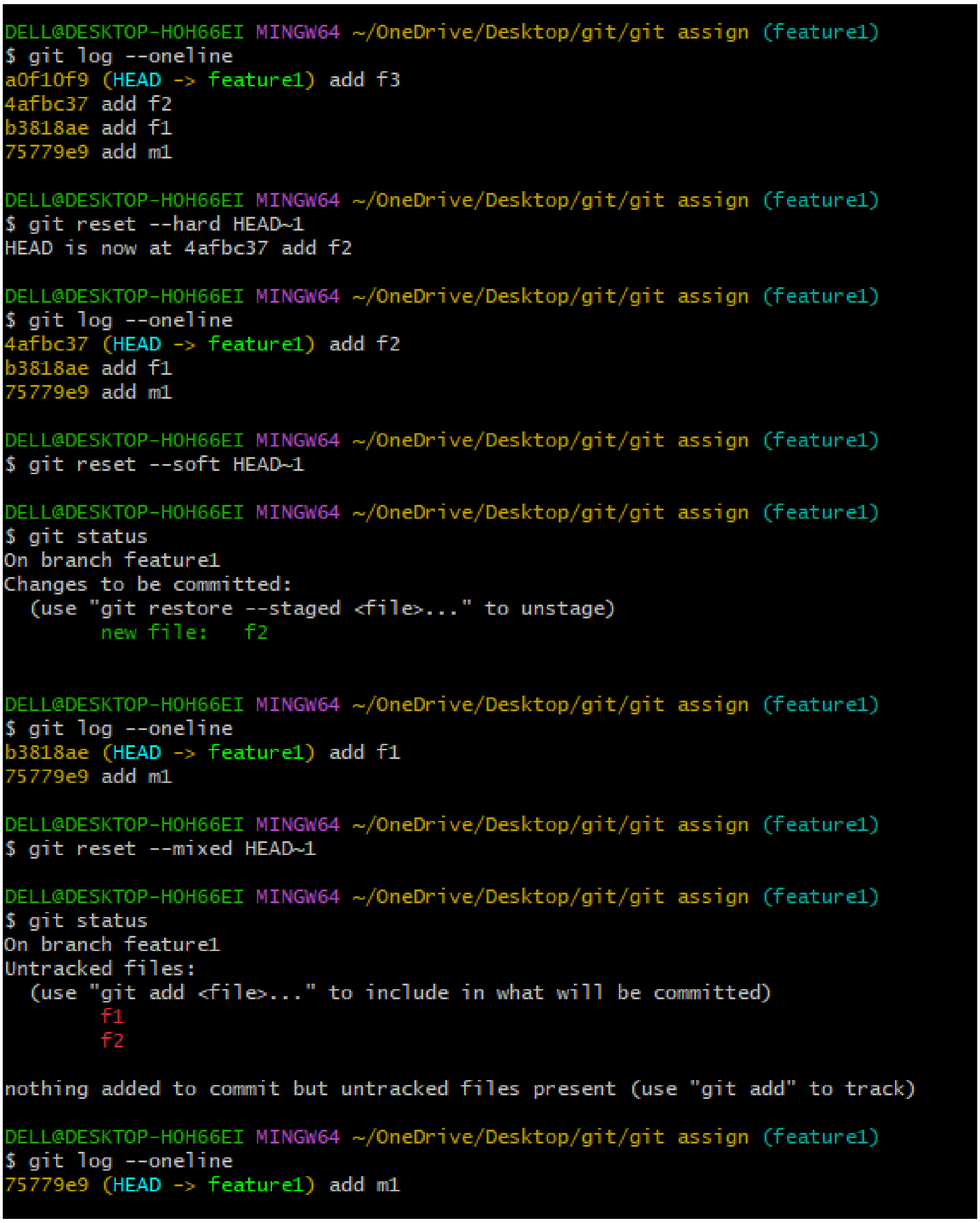
```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ vi .gitignore

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        file1
        file2
        file3

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

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assignment (master)
$ |
```

29. **git revert <commit ID>:-** undo the commited changes (history will be tracked)



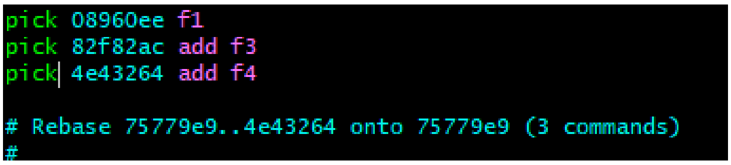
```
DELL@DESKTOP-H0H66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git log --oneline
2e64714 (HEAD -> feature1) add f2
f61197a update f1
54a0a3e add f1
75779e9 add m1

DELL@DESKTOP-H0H66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git revert ^C

DELL@DESKTOP-H0H66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git revert 2e64714
[feature1 771e54d] Revert "add f2"
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 f2

DELL@DESKTOP-H0H66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git status
On branch feature1
nothing to commit, working tree clean

DELL@DESKTOP-H0H66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git log --oneline
771e54d (HEAD -> feature1) Revert "add f2"
2e64714 add f2
f61197a update f1
54a0a3e add f1
75779e9 add m1

DELL@DESKTOP-H0H66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ |
```

30. **git reset :-** Undo the commited changes (history will not be tracked)
31. **git reset --soft HEAD~1**:- To remove the file from commit history only (takes back file to staging area).
32. **git reset --mixed HEAD~2:-** To remove the file from commit history only (takes back file to working dir)
33. **git reset --hard HEAD~3:-** To delete the file from commit and git permannently.

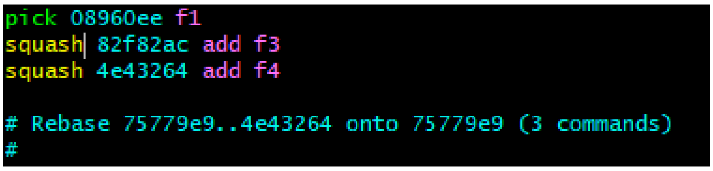


```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git log --oneline
a0f10f9 (HEAD -> feature1) add f3
4afbc37 add f2
b3818ae add f1
75779e9 add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git reset --hard HEAD~1
HEAD is now at 4afbc37 add f2

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git log --oneline
4afbc37 (HEAD -> feature1) add f2
b3818ae add f1
75779e9 add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git reset --soft HEAD~1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git status
On branch feature1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   f2


DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git log --oneline
b3818ae (HEAD -> feature1) add f1
75779e9 add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git reset --mixed HEAD~1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git status
On branch feature1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        f1
        f2

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

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git log --oneline
75779e9 (HEAD -> feature1) add m1
```

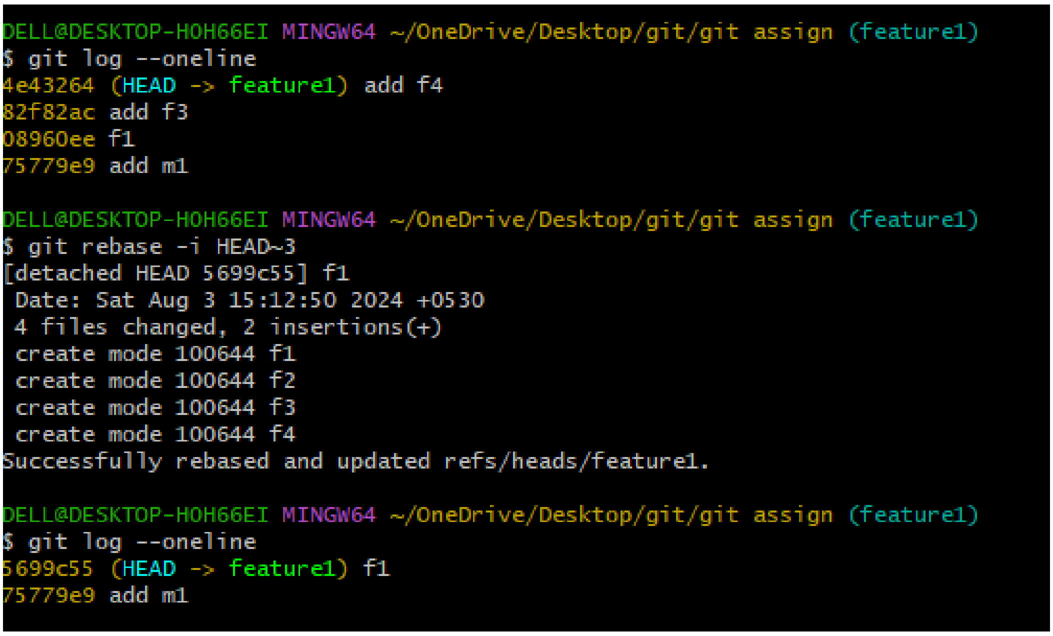
34. **git squash (git rebase -i HEAD ~ 2 or 3 (no of commit)** :- to merge multiple commits into a single commit, used to cleanup a commit history before merging



```
pick 08960ee f1
pick 82f82ac add f3
pick 4e43264 add f4

# Rebase 75779e9..4e43264 onto 75779e9 (3 commands)
#
```

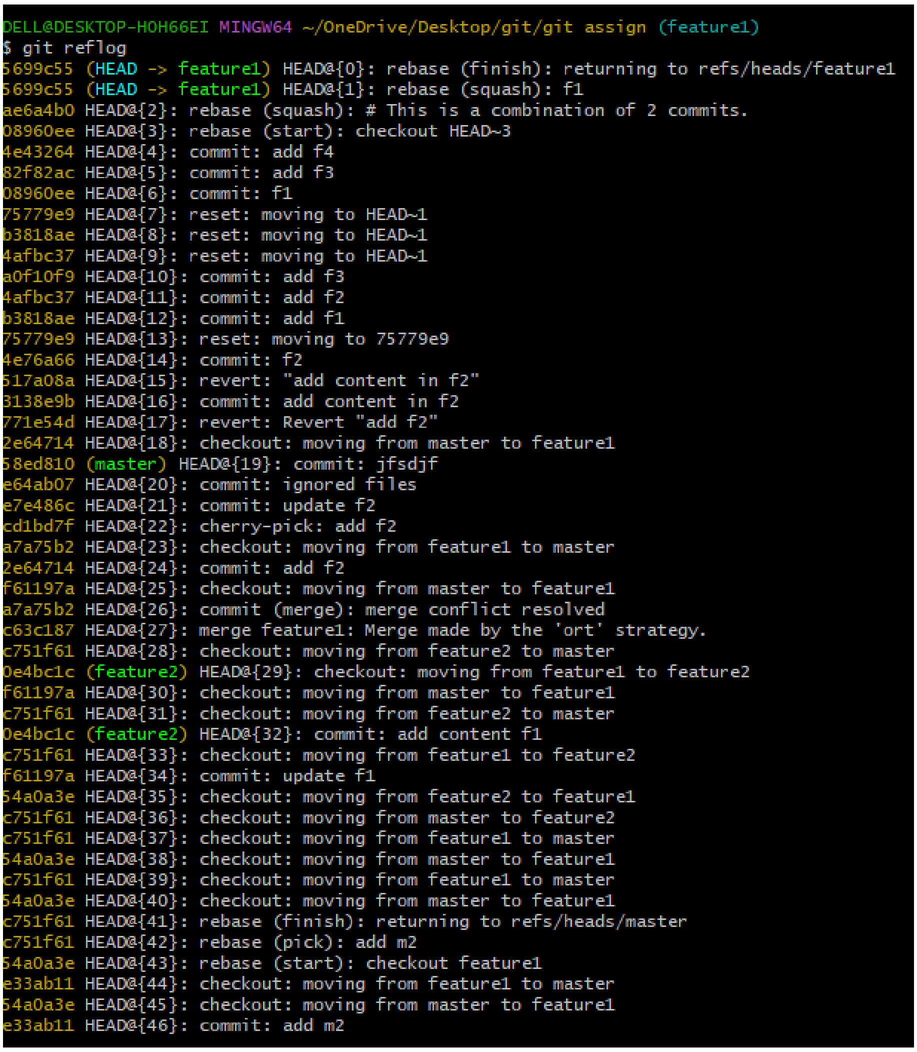


```
pick 08960ee f1
squash 82f82ac add f3
squash 4e43264 add f4

# Rebase 75779e9..4e43264 onto 75779e9 (3 commands)
#
```



```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git log --oneline
4e43264 (HEAD -> feature1) add f4
82f82ac add f3
08960ee f1
75779e9 add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git rebase -i HEAD~3
[detached HEAD 5699c55] f1
 Date: Sat Aug 3 15:12:50 2024 +0530
 4 files changed, 2 insertions(+)
 create mode 100644 f1
 create mode 100644 f2
 create mode 100644 f3
 create mode 100644 f4
Successfully rebased and updated refs/heads/feature1.

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git log --oneline
5699c55 (HEAD -> feature1) f1
75779e9 add m1
```

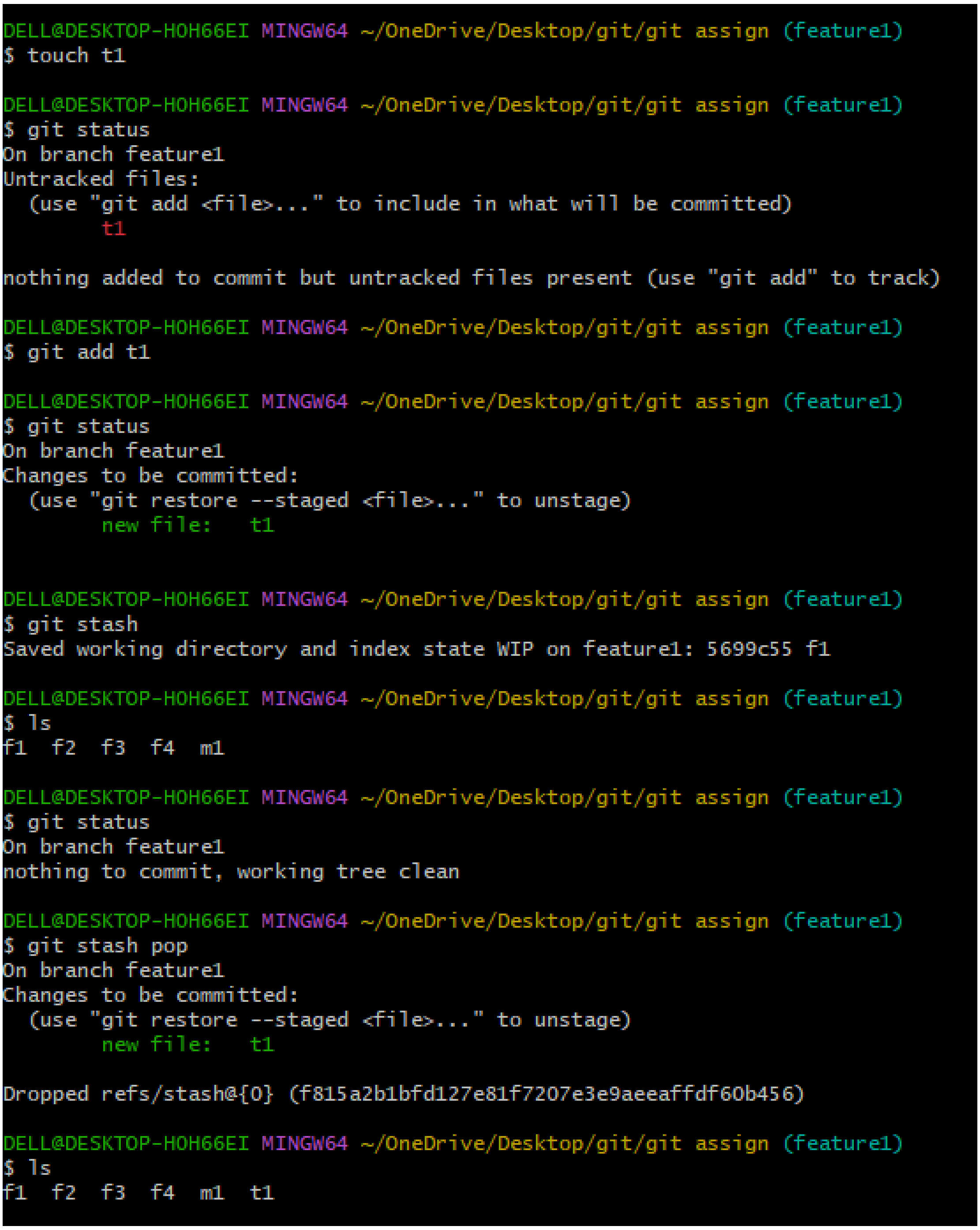
35. **git reflog:-** To view the history of all references and changes in the repository, including those that are not visible in the standard commit log.



```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git reflog
5699c55 (HEAD -> feature1) HEAD@{0}: rebase (finish): returning to refs/heads/feature1
5699c55 (HEAD -> feature1) HEAD@{1}: rebase (squash): f1
ae6a4b0 HEAD@{2}: rebase (squash): # This is a combination of 2 commits.
08960ee HEAD@{3}: rebase (start): checkout HEAD~3
4e43264 HEAD@{4}: commit: add f4
82f82ac HEAD@{5}: commit: add f3
08960ee HEAD@{6}: commit: f1
75779e9 HEAD@{7}: reset: moving to HEAD~1
b3818ae HEAD@{8}: reset: moving to HEAD~1
4afbc37 HEAD@{9}: reset: moving to HEAD~1
a0f10f9 HEAD@{10}: commit: add f3
4afbc37 HEAD@{11}: commit: add f2
b3818ae HEAD@{12}: commit: add f1
75779e9 HEAD@{13}: reset: moving to 75779e9
4e76a66 HEAD@{14}: commit: f2
517a08a HEAD@{15}: revert: "add content in f2"
3138e9b HEAD@{16}: commit: add content in f2
771e54d HEAD@{17}: revert: Revert "add f2"
2e64714 HEAD@{18}: checkout: moving from master to feature1
58ed810 (master) HEAD@{19}: commit: jfsdjf
e64ab07 HEAD@{20}: commit: ignored files
e7e486c HEAD@{21}: commit: update f2
cd1bd7f HEAD@{22}: cherry-pick: add f2
a7a75b2 HEAD@{23}: checkout: moving from feature1 to master
2e64714 HEAD@{24}: commit: add f2
f61197a HEAD@{25}: checkout: moving from master to feature1
a7a75b2 HEAD@{26}: commit (merge): merge conflict resolved
c63c187 HEAD@{27}: merge feature1: Merge made by the 'ort' strategy.
c751f61 HEAD@{28}: checkout: moving from feature2 to master
0e4bc1c (feature2) HEAD@{29}: checkout: moving from feature1 to feature2
f61197a HEAD@{30}: checkout: moving from master to feature1
c751f61 HEAD@{31}: checkout: moving from feature2 to master
0e4bc1c (feature2) HEAD@{32}: commit: add content f1
c751f61 HEAD@{33}: checkout: moving from feature1 to feature2
f61197a HEAD@{34}: commit: update f1
54a0a3e HEAD@{35}: checkout: moving from feature2 to feature1
c751f61 HEAD@{36}: checkout: moving from master to feature2
c751f61 HEAD@{37}: checkout: moving from feature1 to master
54a0a3e HEAD@{38}: checkout: moving from master to feature1
c751f61 HEAD@{39}: checkout: moving from feature1 to master
54a0a3e HEAD@{40}: checkout: moving from master to feature1
c751f61 HEAD@{41}: rebase (finish): returning to refs/heads/master
c751f61 HEAD@{42}: rebase (pick): add m2
54a0a3e HEAD@{43}: rebase (start): checkout feature1
e33ab11 HEAD@{44}: checkout: moving from feature1 to master
54a0a3e HEAD@{45}: checkout: moving from master to feature1
e33ab11 HEAD@{46}: commit: add m2
```

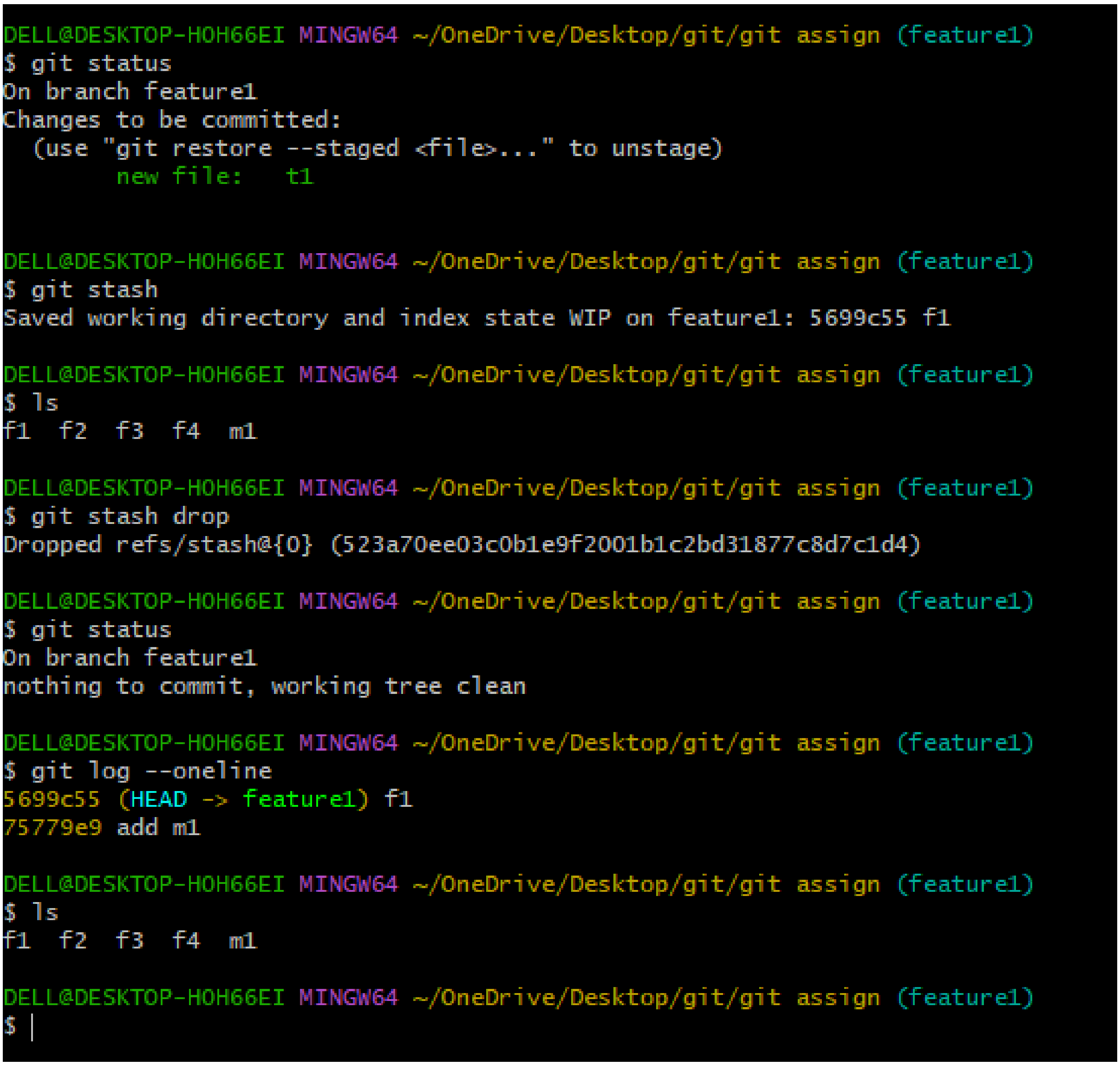
36. **git stash:-** To hide the file from master/branch (after staging the file).

37. git stash pop:- To get back files from temporary area



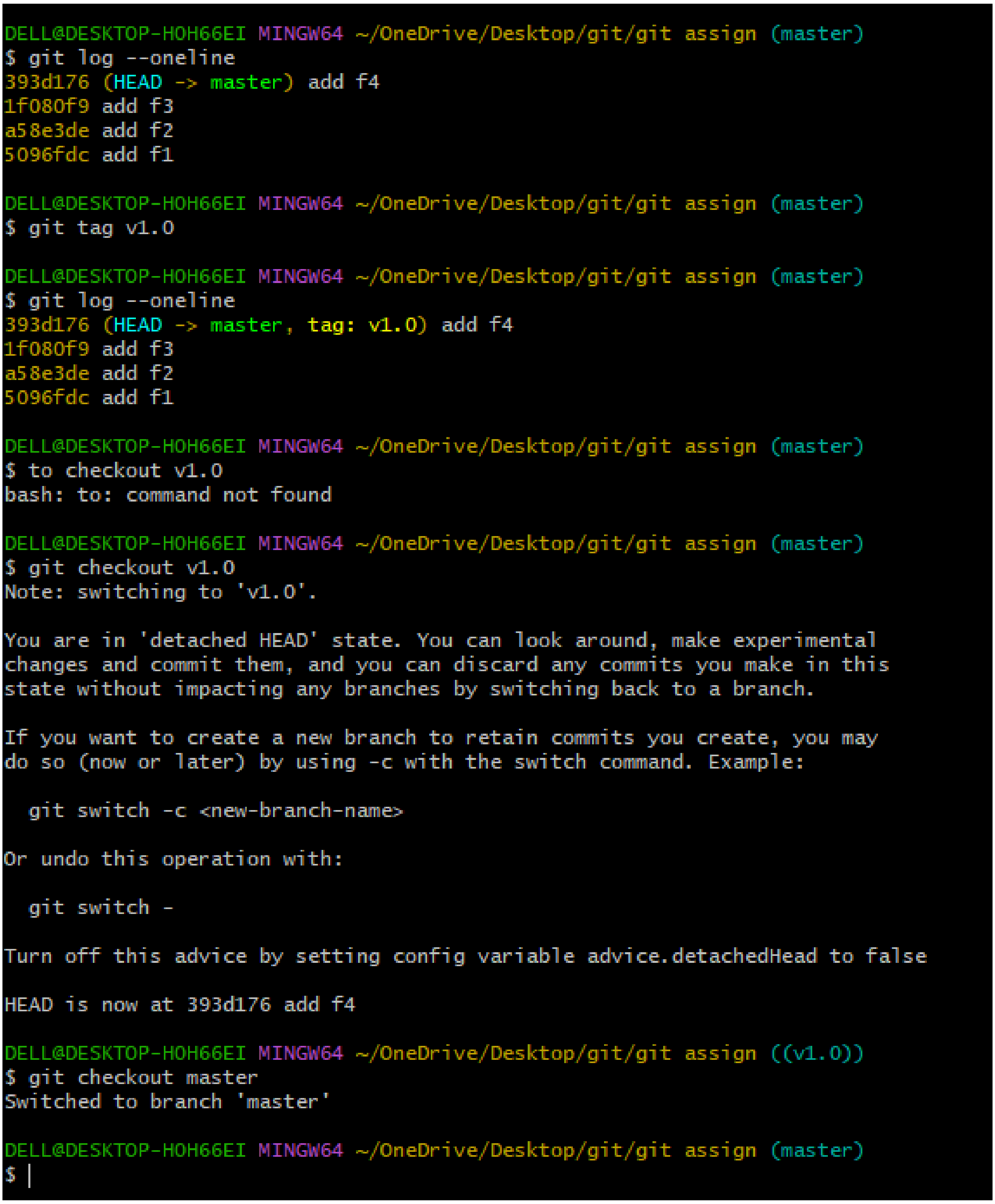
```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ touch t1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git status
On branch feature1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        t1

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

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git add t1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git status
On branch feature1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   t1


DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git stash
Saved working directory and index state WIP on feature1: 5699c55 f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ ls
f1  f2  f3  f4  m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git status
On branch feature1
nothing to commit, working tree clean

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git stash pop
On branch feature1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   t1

Dropped refs/stash@{0} (f815a2b1bfd127e81f7207e3e9aeeaffdf60b456)

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ ls
f1  f2  f3  f4  m1  t1
```

38. git stash drop:- When you are done with the stashed item or want to remove it from the list. It will remove the last added stash item by default

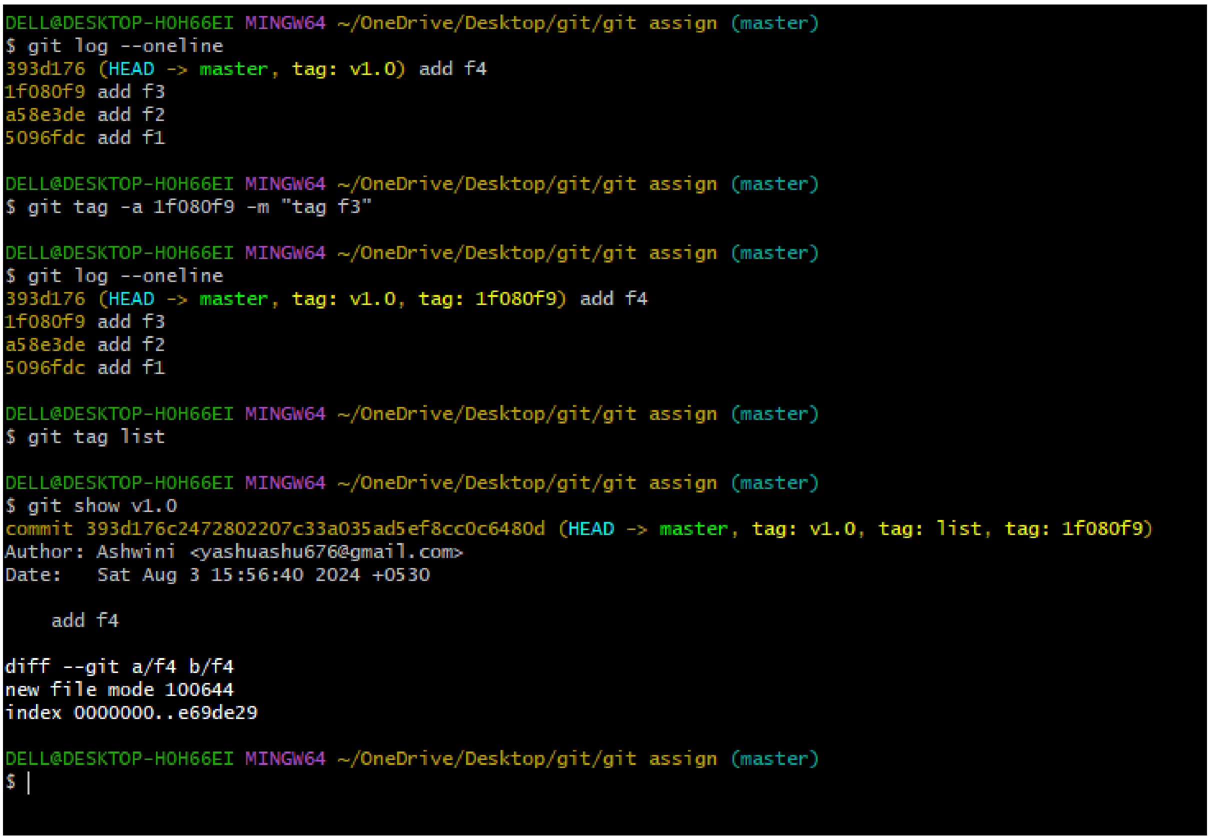


```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git status
On branch feature1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   t1


DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git stash
Saved working directory and index state WIP on feature1: 5699c55 f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ ls
f1  f2  f3  f4  m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git stash drop
Dropped refs/stash@{0} (523a70ee03c0b1e9f2001b1c2bd31877c8d7c1d4)

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git status
On branch feature1
nothing to commit, working tree clean

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ git log --oneline
5699c55 (HEAD -> feature1) f1
75779e9 add m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ ls
f1  f2  f3  f4  m1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (feature1)
$ |
```

39. **git tag tagname or git tag -a <commit id> -m "tagging f2" :-** To create a tag name
40. **git checkout tagname :-** Switch to tag
41. **git tag -d tagname :-** To delete the tag

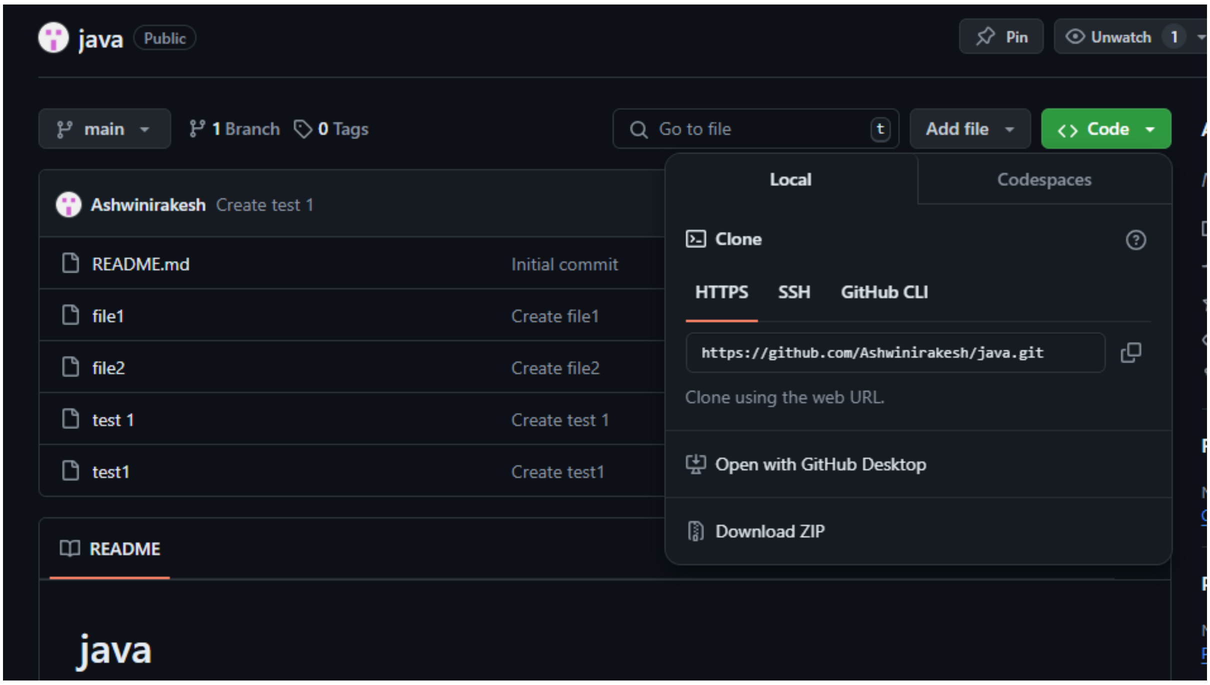


```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log --oneline
393d176 (HEAD -> master) add f4
1f080f9 add f3
a58e3de add f2
5096fdc add f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git tag v1.0

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log --oneline
393d176 (HEAD -> master, tag: v1.0) add f4
1f080f9 add f3
a58e3de add f2
5096fdc add f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ to checkout v1.0
bash: to: command not found

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git checkout v1.0
Note: switching to 'v1.0'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 393d176 add f4

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign ((v1.0))
$ git checkout master
Switched to branch 'master'

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ |
```



```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log --oneline
393d176 (HEAD -> master, tag: v1.0) add f4
1f080f9 add f3
a58e3de add f2
5096fdc add f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git tag -a 1f080f9 -m "tag f3"

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git log --oneline
393d176 (HEAD -> master, tag: v1.0, tag: 1f080f9) add f4
1f080f9 add f3
a58e3de add f2
5096fdc add f1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git tag list

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git show v1.0
commit 393d176c2472802207c33a035ad5ef8cc0c6480d (HEAD -> master, tag: v1.0, tag: list, tag: 1f080f9)
Author: Ashwini <yashuashu676@gmail.com>
Date:   Sat Aug 3 15:56:40 2024 +0530

    add f4

diff --git a/f4 b/f4
new file mode 100644
index 0000000..e69de29

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ |
```

42. **Git clone <URL>:-** synch central repo to local repo only 1st time



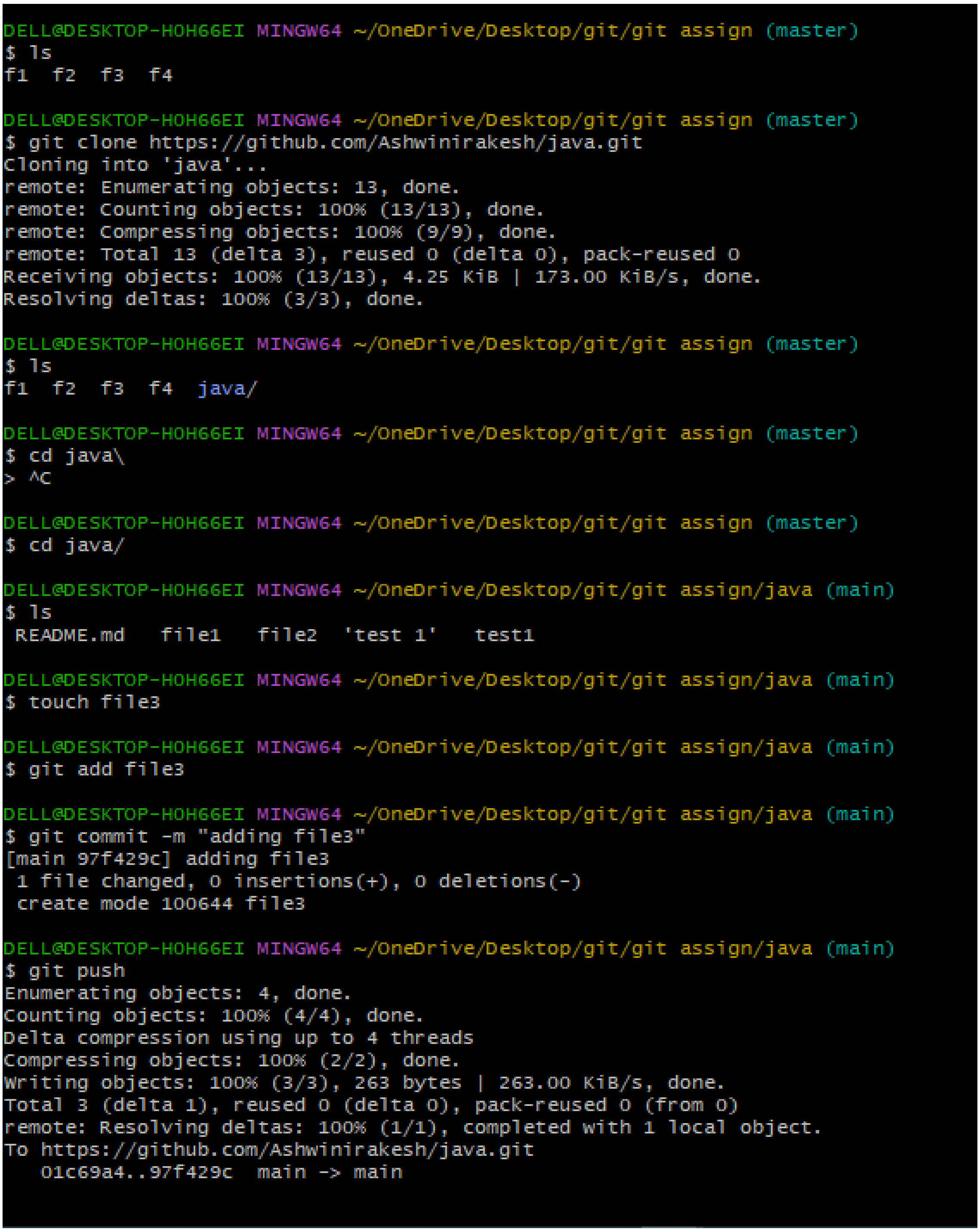
Ashwinirakesh Create test 1

README.md	Initial commit
file1	Create file1
file2	Create file2
test 1	Create test 1
test1	Create test1

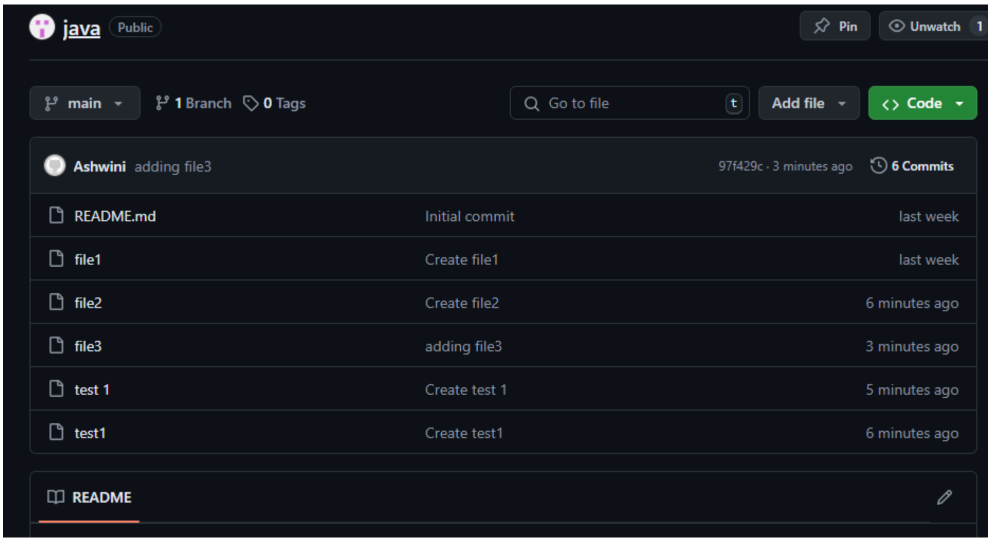
https://github.com/Ashwinirakesh/java.git

Clone using the web URL.

Central or remote repository



```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ ls
f1  f2  f3  f4

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ git clone https://github.com/Ashwinirakesh/java.git
Cloning into 'java'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 13 (delta 3), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (13/13), 4.25 KiB | 173.00 KiB/s, done.
Resolving deltas: 100% (3/3), done.

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ ls
f1  f2  f3  f4  java/

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ cd java\
> ^C

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign (master)
$ cd java/

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ ls
 README.md   file1   file2  'test 1'   test1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ touch file3

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ git add file3

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ git commit -m "adding file3"
[main 97f429c] adding file3
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file3

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 263 bytes | 263.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Ashwinirakesh/java.git
   01c69a4..97f429c  main -> main
```



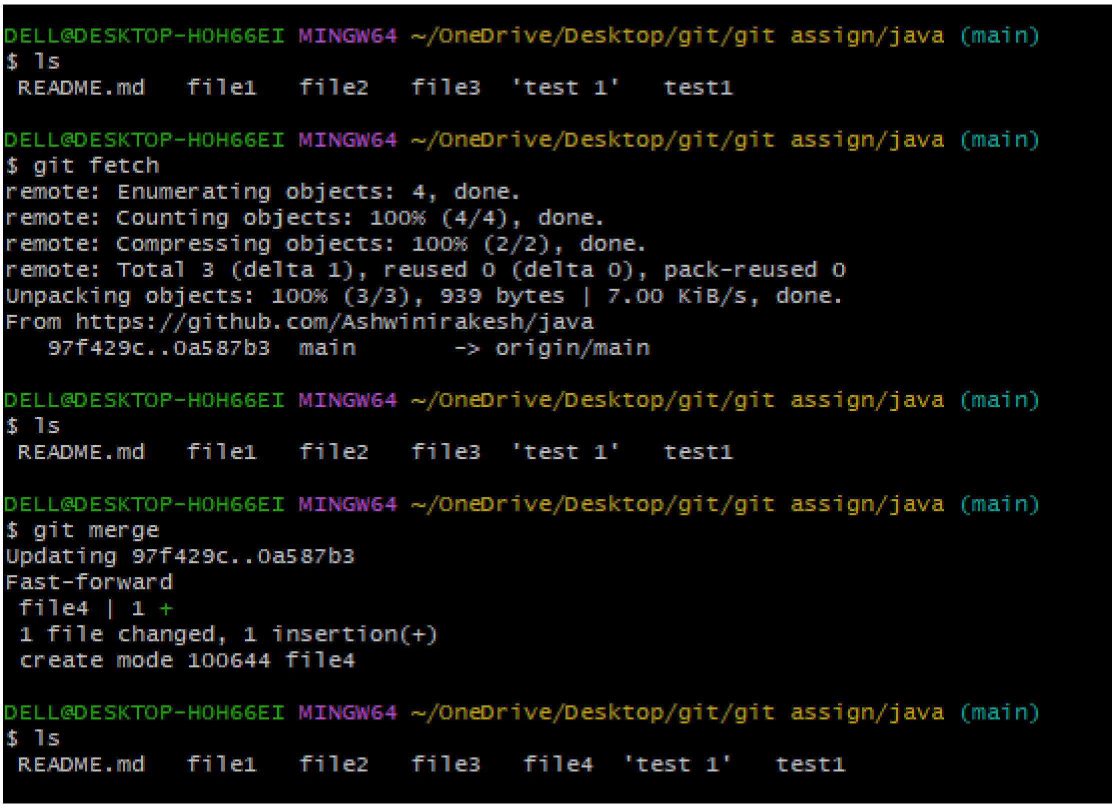
Ashwini adding file3 97f429c · 3 minutes ago

README.md	Initial commit	last week
file1	Create file1	last week
file2	Create file2	6 minutes ago
file3	adding file3	3 minutes ago
test 1	Create test 1	5 minutes ago
test1	Create test1	6 minutes ago

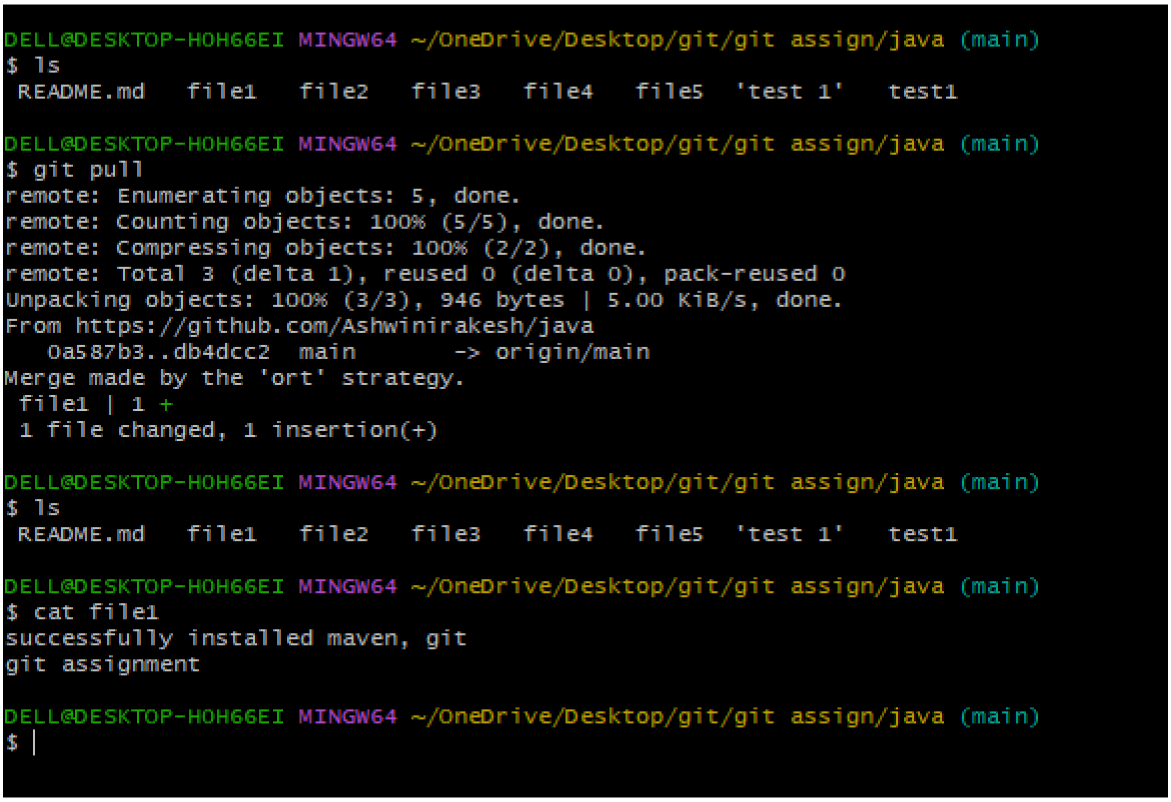
After git push

43. **Git fetch:-** It is same as pull but the comiits are not merged, it is in a temporary branch fetch head.

Git pull = git fetch +git merge



```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ ls
 README.md   file1   file2   file3  'test 1'   test1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ git fetch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 939 bytes | 7.00 KiB/s, done.
From https://github.com/Ashwinirakesh/java
   97f429c..0a587b3  main       -> origin/main

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ ls
 README.md   file1   file2   file3  'test 1'   test1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ git merge
Updating 97f429c..0a587b3
Fast-forward
 file4 | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 file4

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ ls
 README.md   file1   file2   file3   file4  'test 1'   test1
```



```
DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ ls
 README.md   file1   file2   file3   file4   file5  'test 1'   test1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 946 bytes | 5.00 KiB/s, done.
From https://github.com/Ashwinirakesh/java
   0a587b3..db4dcc2  main       -> origin/main
Merge made by the 'ort' strategy.
 file1 | 1 +
 1 file changed, 1 insertion(+)

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ ls
 README.md   file1   file2   file3   file4   file5  'test 1'   test1

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ cat file1
successfully installed maven, git
git assignment

DELL@DESKTOP-HOH66EI MINGW64 ~/OneDrive/Desktop/git/git assign/java (main)
$ |
```