Git and GitHub Assignment

Install GIT & make sure it is added into PATH.

Use GIT as local VCS. Steps to follow:

1. Create a directory ‘project_dir’ & cd to ‘project_dir’.
2. Initialize git version database. (git init)
3. Create a new file index.html.
4. Check the git status. You should find index.html as untracked file.
5. Stage the index.html file.
6. Commit index.html
7. Make few changes in index.html & create a new file info.txt file.
8. Check git status. You should find index.html & info.txt as untracked files.
9. Configure GIT to ignore all txt files.

10. Again check the git status. You should find only index.html as untracked file.

11. State & commit index.html

12. Log all your comments so far.

13. Make some changes in index.html.

14. Revert the change made in the previous step using git command.

15. Again change index.html.

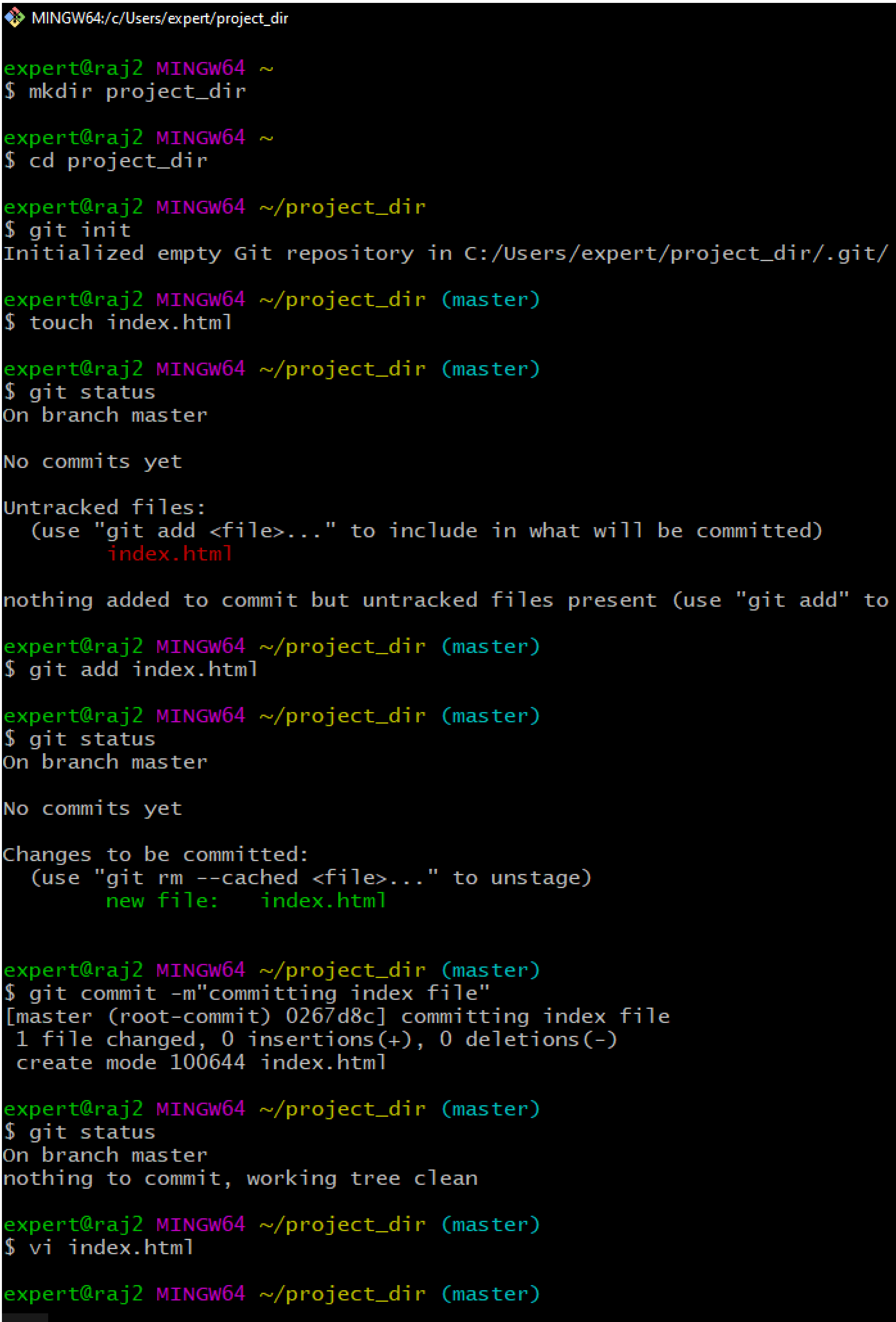
16. Stage index.html

17. Revert back the last stage.

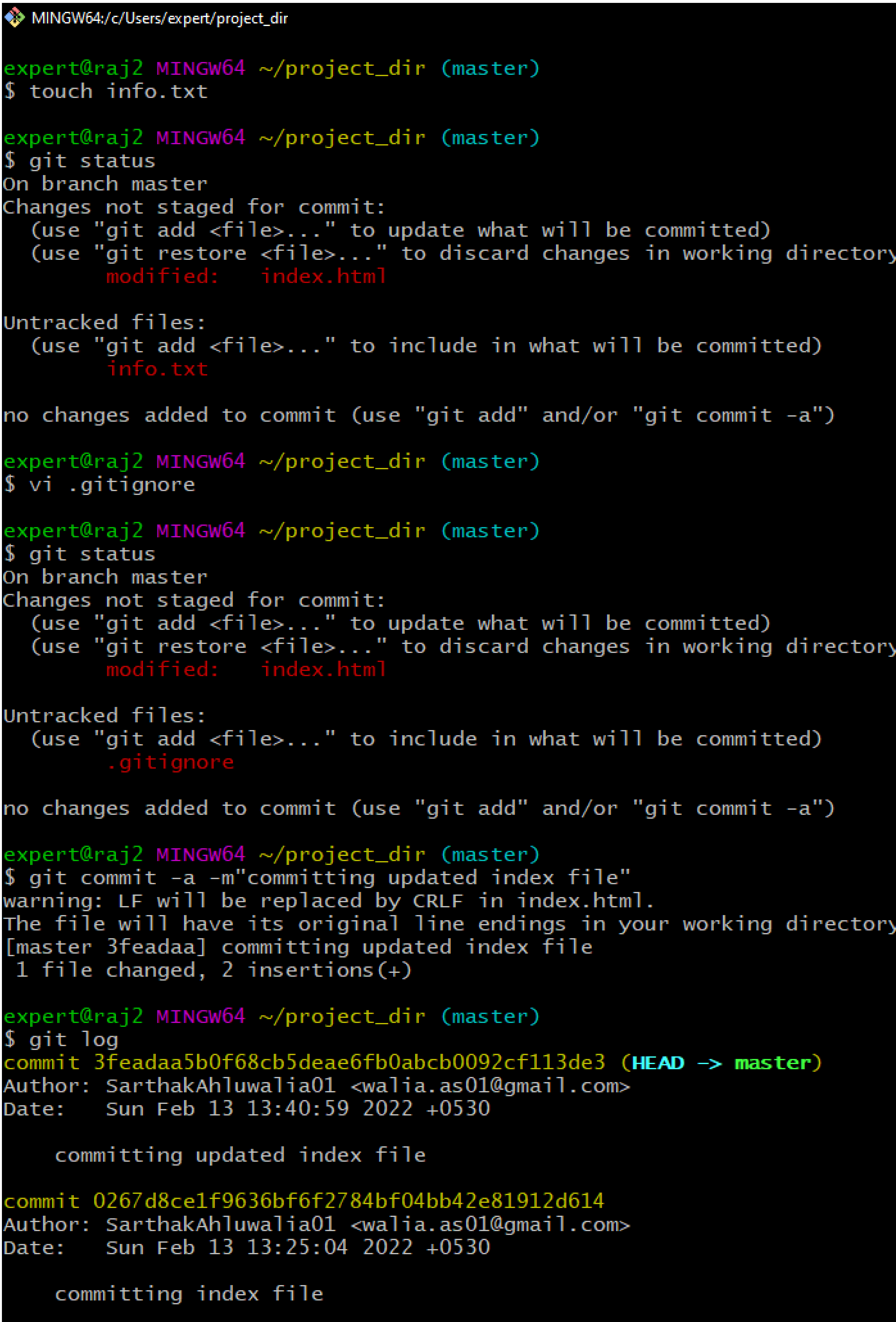
18. Rename ‘add’ command to ‘my-add’.

19. Using my_add command Stage index.html again & commit the changes.

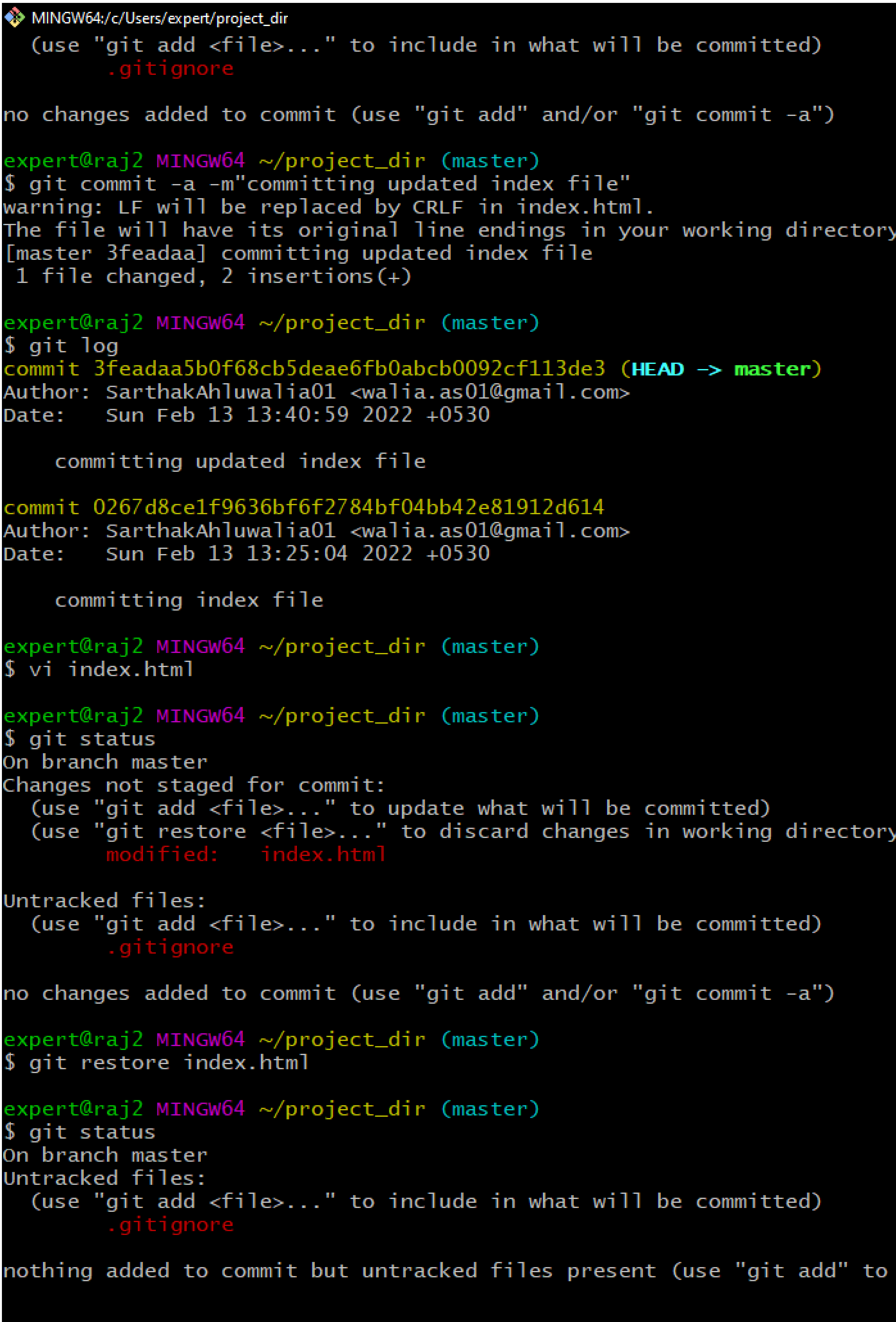
20. Revert the last commit.



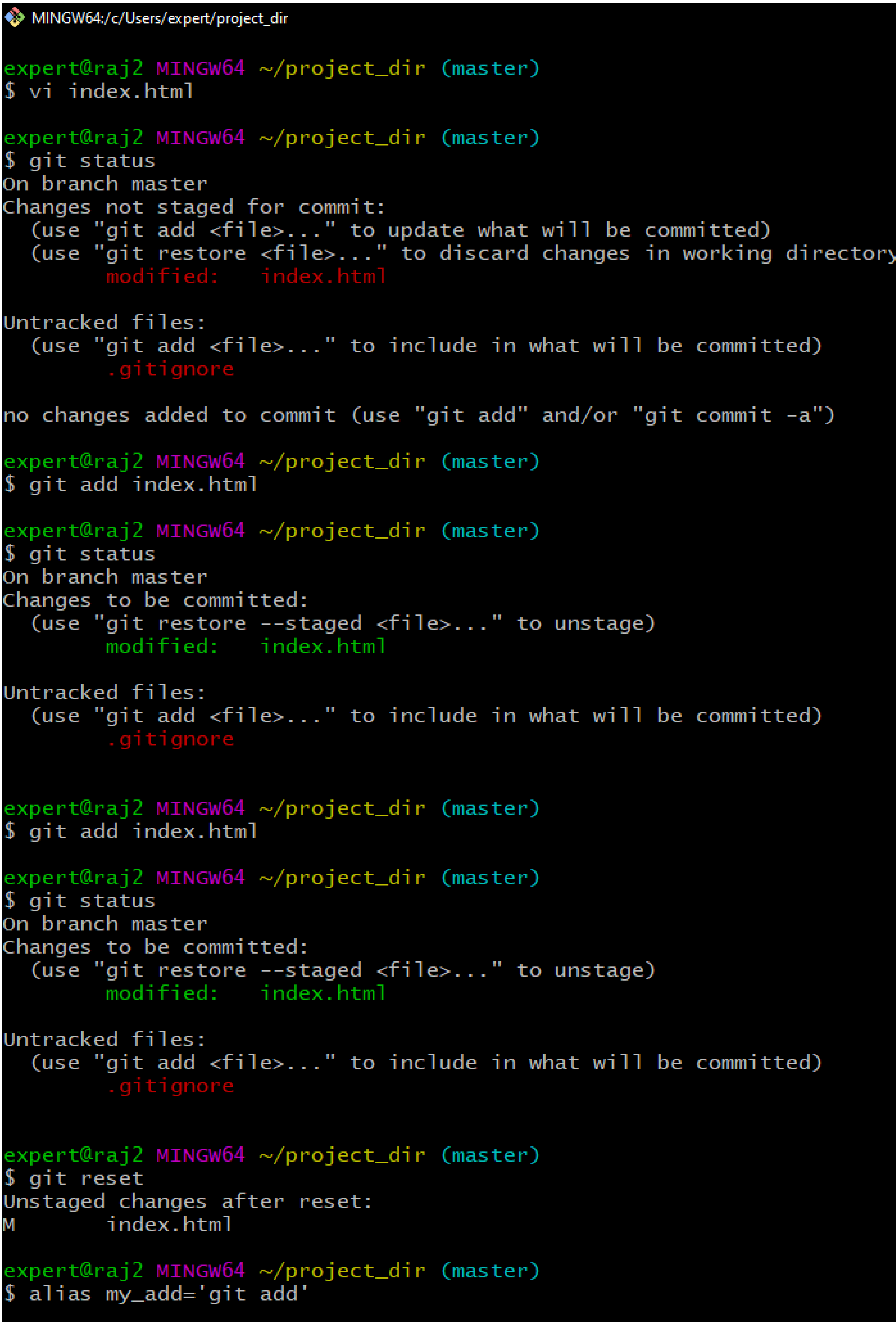
```
expert@raj2 MINGW64 ~
$ mkdir project_dir

expert@raj2 MINGW64 ~
$ cd project_dir

expert@raj2 MINGW64 ~/project_dir
$ git init
Initialized empty Git repository in C:/Users/expert/project_dir/.git/

expert@raj2 MINGW64 ~/project_dir (master)
$ touch index.html

expert@raj2 MINGW64 ~/project_dir (master)
$ git status
On branch master

No commits yet

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

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

expert@raj2 MINGW64 ~/project_dir (master)
$ git add index.html

expert@raj2 MINGW64 ~/project_dir (master)
$ git status
On branch master

No commits yet

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


expert@raj2 MINGW64 ~/project_dir (master)
$ git commit -m"committing index file"
[master (root-commit) 0267d8c] committing index file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html

expert@raj2 MINGW64 ~/project_dir (master)
$ git status
On branch master
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/project_dir (master)
$ vi index.html

expert@raj2 MINGW64 ~/project_dir (master)
```



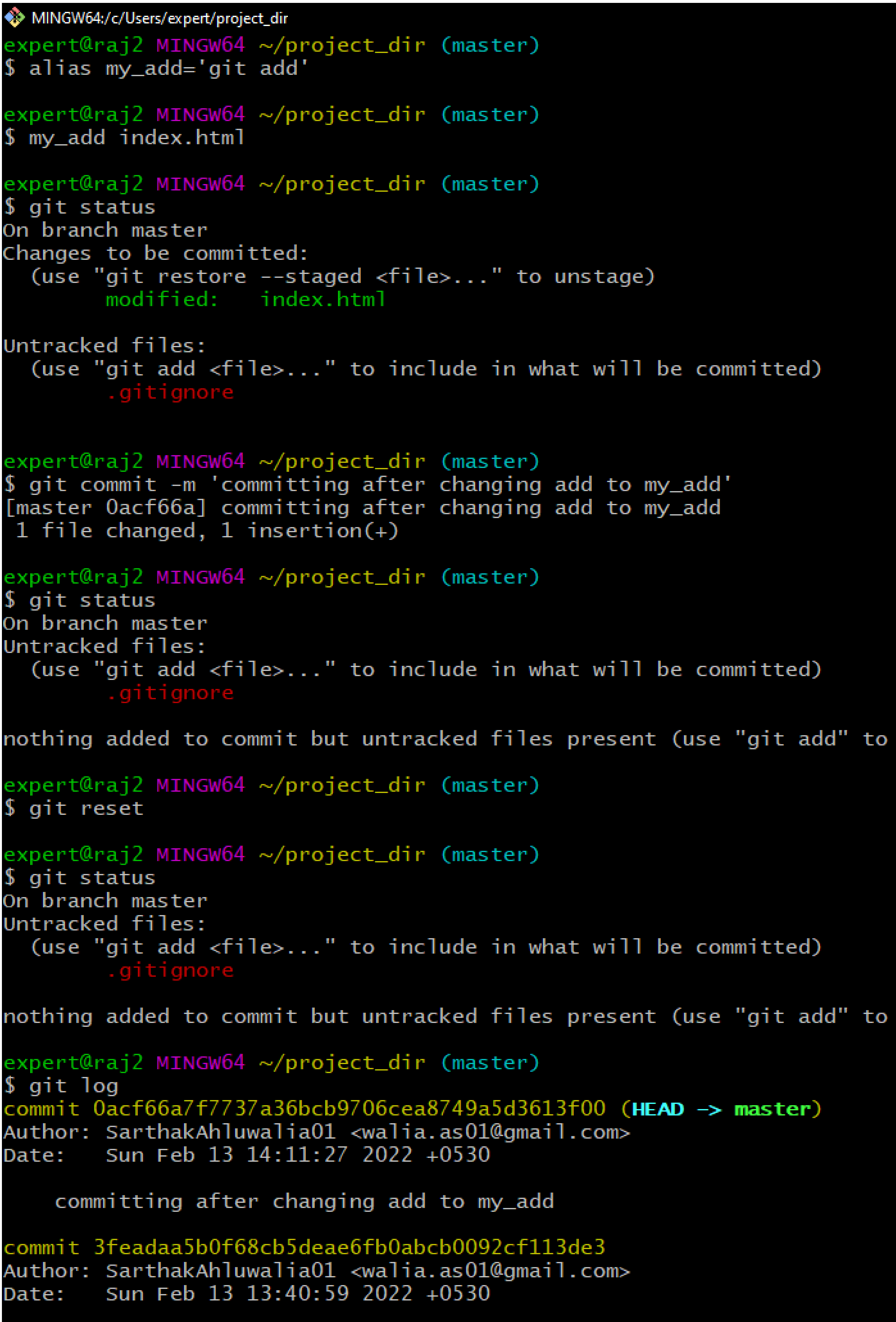
```
expert@raj2 MINGW64 ~/project_dir (master)
$ touch info.txt

expert@raj2 MINGW64 ~/project_dir (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:   index.html

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

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

expert@raj2 MINGW64 ~/project_dir (master)
$ vi .gitignore

expert@raj2 MINGW64 ~/project_dir (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:   index.html

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

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

expert@raj2 MINGW64 ~/project_dir (master)
$ git commit -a -m"committing updated index file"
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory
[master 3feadaa] committing updated index file
 1 file changed, 2 insertions(+)

expert@raj2 MINGW64 ~/project_dir (master)
$ git log
commit 3feadaa5b0f68cb5deae6fb0abcb0092cf113de3 (HEAD -> master)
Author: SarthakAhluwalia01 <walia.as01@gmail.com>
Date:   Sun Feb 13 13:40:59 2022 +0530

    committing updated index file

commit 0267d8ce1f9636bf6f2784bf04bb42e81912d614
Author: SarthakAhluwalia01 <walia.as01@gmail.com>
Date:   Sun Feb 13 13:25:04 2022 +0530

    committing index file
```



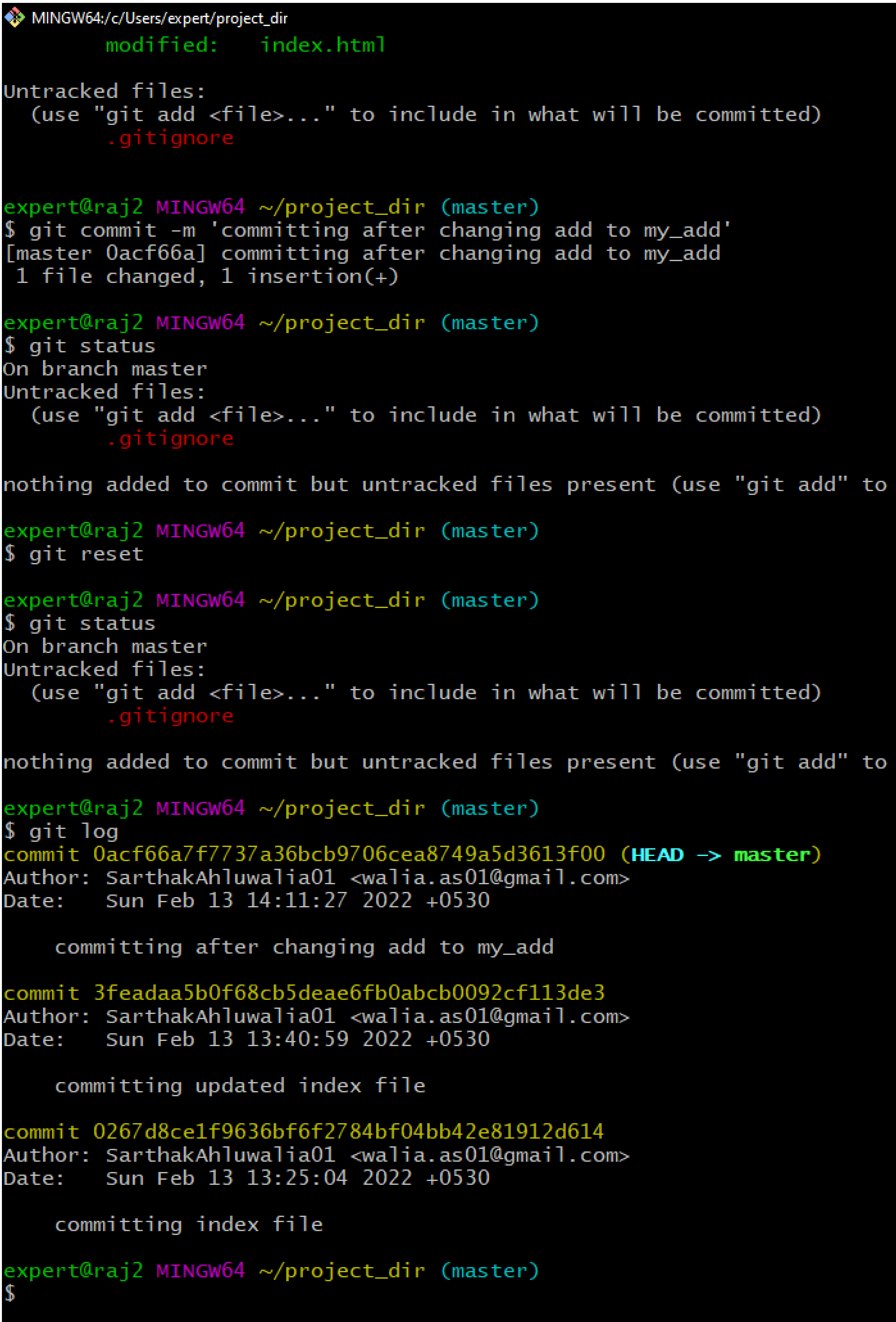
```
  (use "git add <file>..." to include in what will be committed)
        .gitignore

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

expert@raj2 MINGW64 ~/project_dir (master)
$ git commit -a -m"committing updated index file"
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory
[master 3feadaa] committing updated index file
 1 file changed, 2 insertions(+)

expert@raj2 MINGW64 ~/project_dir (master)
$ git log
commit 3feadaa5b0f68cb5deae6fb0abcb0092cf113de3 (HEAD -> master)
Author: SarthakAhluwalia01 <walia.as01@gmail.com>
Date:   Sun Feb 13 13:40:59 2022 +0530

    committing updated index file

commit 0267d8ce1f9636bf6f2784bf04bb42e81912d614
Author: SarthakAhluwalia01 <walia.as01@gmail.com>
Date:   Sun Feb 13 13:25:04 2022 +0530

    committing index file

expert@raj2 MINGW64 ~/project_dir (master)
$ vi index.html

expert@raj2 MINGW64 ~/project_dir (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:   index.html

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

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

expert@raj2 MINGW64 ~/project_dir (master)
$ git restore index.html

expert@raj2 MINGW64 ~/project_dir (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

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



```
MINGW64:/c/Users/expert/project_dir

expert@raj2 MINGW64 ~/project_dir (master)
$ vi index.html

expert@raj2 MINGW64 ~/project_dir (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:   index.html

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

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

expert@raj2 MINGW64 ~/project_dir (master)
$ git add index.html

expert@raj2 MINGW64 ~/project_dir (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

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


expert@raj2 MINGW64 ~/project_dir (master)
$ git add index.html

expert@raj2 MINGW64 ~/project_dir (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

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


expert@raj2 MINGW64 ~/project_dir (master)
$ git reset
Unstaged changes after reset:
M       index.html

expert@raj2 MINGW64 ~/project_dir (master)
$ alias my_add='git add'
```

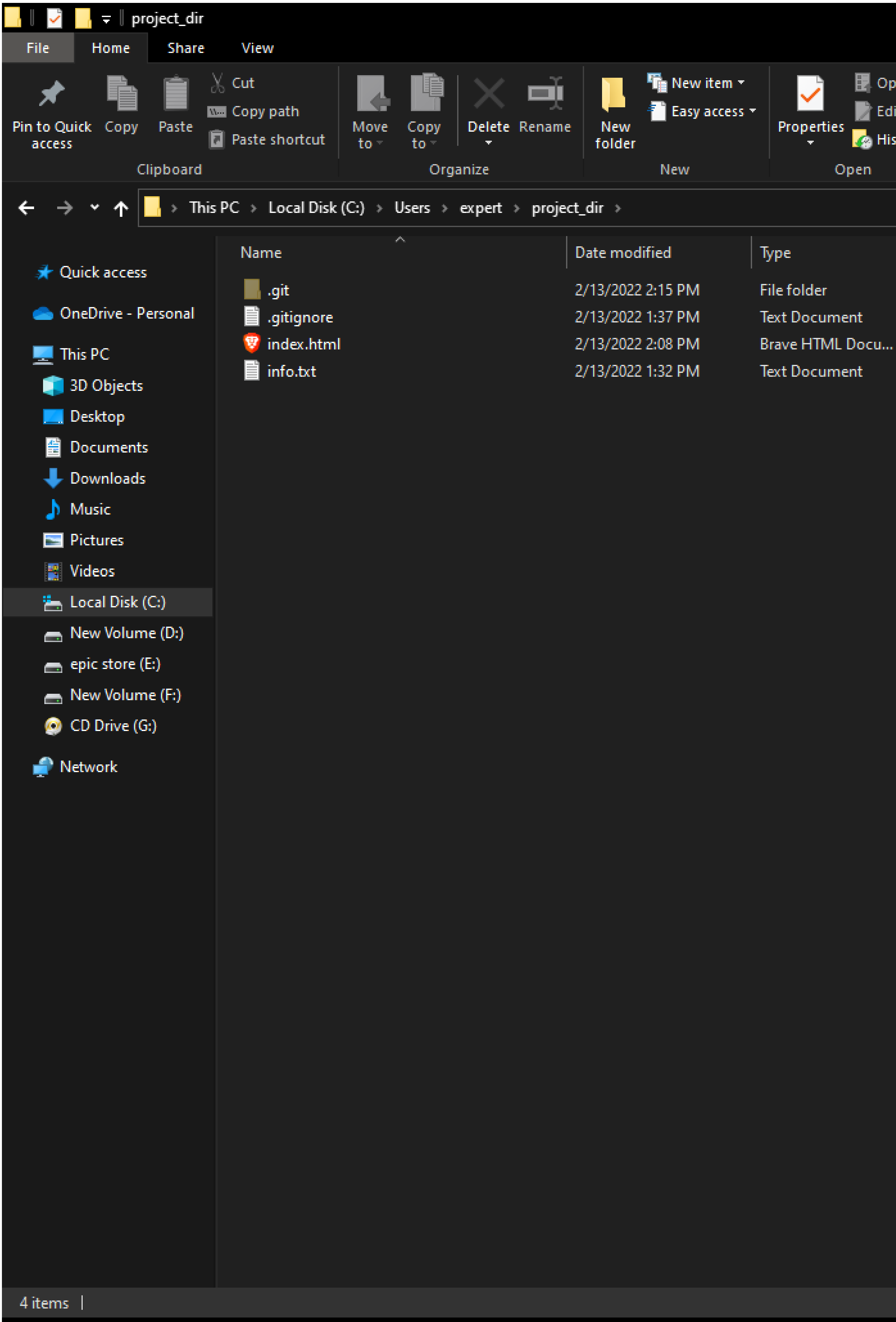


```
expert@raj2 MINGW64 ~/project_dir (master)
$ alias my_add='git add'

expert@raj2 MINGW64 ~/project_dir (master)
$ my_add index.html

expert@raj2 MINGW64 ~/project_dir (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

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


expert@raj2 MINGW64 ~/project_dir (master)
$ git commit -m 'committing after changing add to my_add'
[master 0acf66a] committing after changing add to my_add
 1 file changed, 1 insertion(+)

expert@raj2 MINGW64 ~/project_dir (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

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

expert@raj2 MINGW64 ~/project_dir (master)
$ git reset

expert@raj2 MINGW64 ~/project_dir (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

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

expert@raj2 MINGW64 ~/project_dir (master)
$ git log
commit 0acf66a7f7737a36bcb9706cea8749a5d3613f00 (HEAD -> master)
Author: SarthakAhluwalia01 <walia.as01@gmail.com>
Date:   Sun Feb 13 14:11:27 2022 +0530

    committing after changing add to my_add

commit 3feadaa5b0f68cb5deae6fb0abcb0092cf113de3
Author: SarthakAhluwalia01 <walia.as01@gmail.com>
Date:   Sun Feb 13 13:40:59 2022 +0530
```



MINGW64:/c/Users/expert/project_dir

```
        modified:   index.html

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


expert@raj2 MINGW64 ~/project_dir (master)
$ git commit -m 'committing after changing add to my_add'
[master 0acf66a] committing after changing add to my_add
 1 file changed, 1 insertion(+)

expert@raj2 MINGW64 ~/project_dir (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

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

expert@raj2 MINGW64 ~/project_dir (master)
$ git reset

expert@raj2 MINGW64 ~/project_dir (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

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

expert@raj2 MINGW64 ~/project_dir (master)
$ git log
commit 0acf66a7f7737a36bcb9706cea8749a5d3613f00 (HEAD -> master)
Author: SarthakAhluwalia01 <walia.as01@gmail.com>
Date:   Sun Feb 13 14:11:27 2022 +0530

    committing after changing add to my_add

commit 3feadaa5b0f68cb5deae6fb0abcb0092cf113de3
Author: SarthakAhluwalia01 <walia.as01@gmail.com>
Date:   Sun Feb 13 13:40:59 2022 +0530

    committing updated index file

commit 0267d8ce1f9636bf6f2784bf04bb42e81912d614
Author: SarthakAhluwalia01 <walia.as01@gmail.com>
Date:   Sun Feb 13 13:25:04 2022 +0530

    committing index file

expert@raj2 MINGW64 ~/project_dir (master)
$
```



Name	Date modified	Type
.git	2/13/2022 2:15 PM	File folder
.gitignore	2/13/2022 1:37 PM	Text Document
index.html	2/13/2022 2:08 PM	Brave HTML Docu...
info.txt	2/13/2022 1:32 PM	Text Document

Git Branching

Objective: Commit HTML, CSS & JavaScript assignments into GIT.

SECTION-1 (HTML assignments) - Steps to follow:

21. First take a backup of your assignments & projects. This is required because due to incorrect GIT operation you may lose your files.
22. Create an empty directory 'Assignments' & cd to 'Assignments'.
23. Create a file README.txt inside 'Assignments' & write few lines about the contents of 'Assignments' folder.
24. Commit README.txt file.
25. Now create a new branch 'html-assignments'.
26. Switch to 'html-assignments' branch.
27. Copy all HTML assignments inside 'Assignments' folder.

28. Commit HTML assignments into ‘html-assignments’ branch.

29. Make minor changes into few files belonging to ‘html-assignments’ branch.

30. Commit those changed files.

31. Switch to master branch.

32. Make minor changes into README.txt file & commit those changes into master.

33. Again switch to ‘html-assignments’ branch.

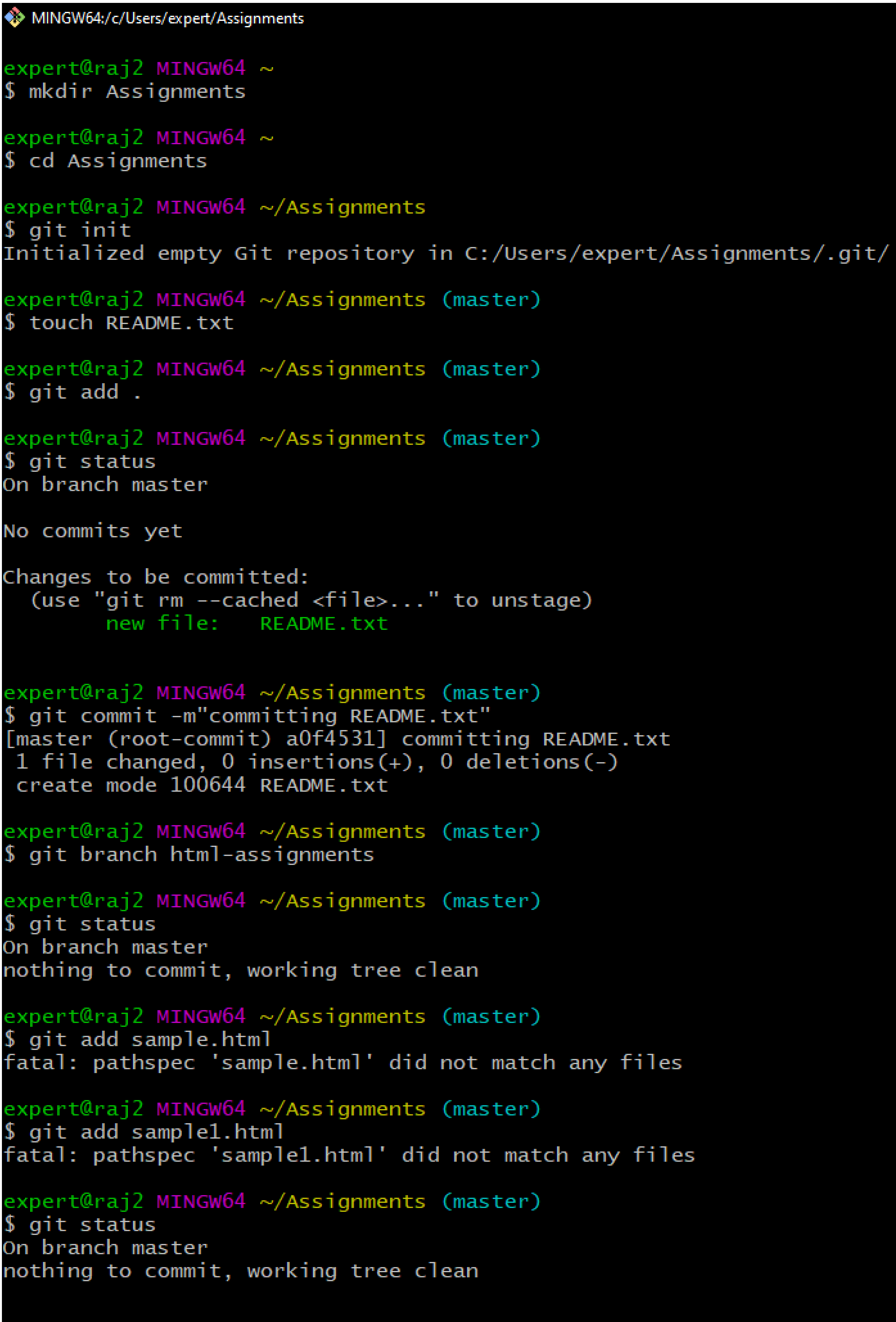
34. Make minor changes into few files belonging to ‘html-assignments’ branch.

35. Commit those changes.

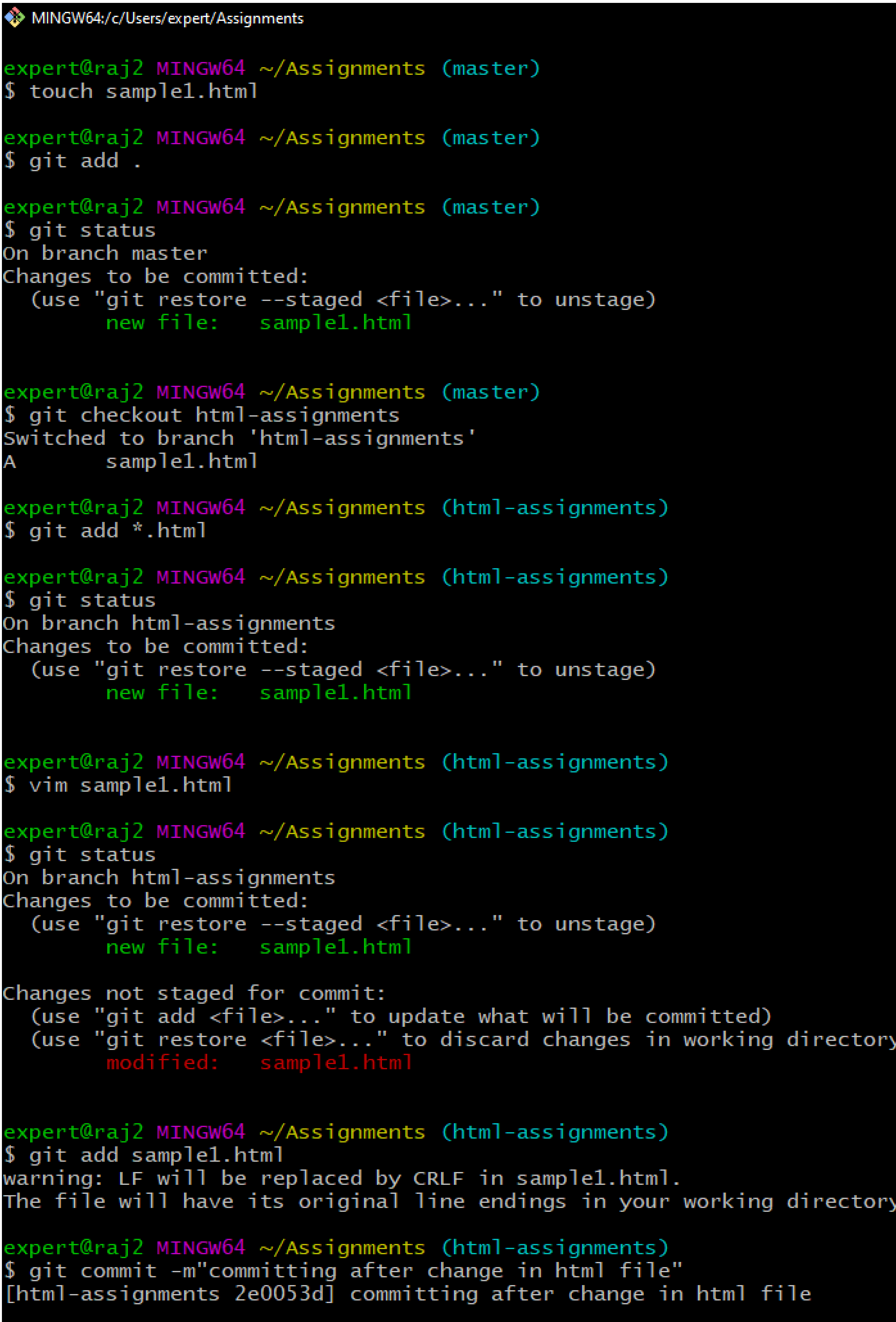
36. Switch to master.

37. Merge ‘html-assignments’ branch into master. Confirm all html assignments are shown in master.

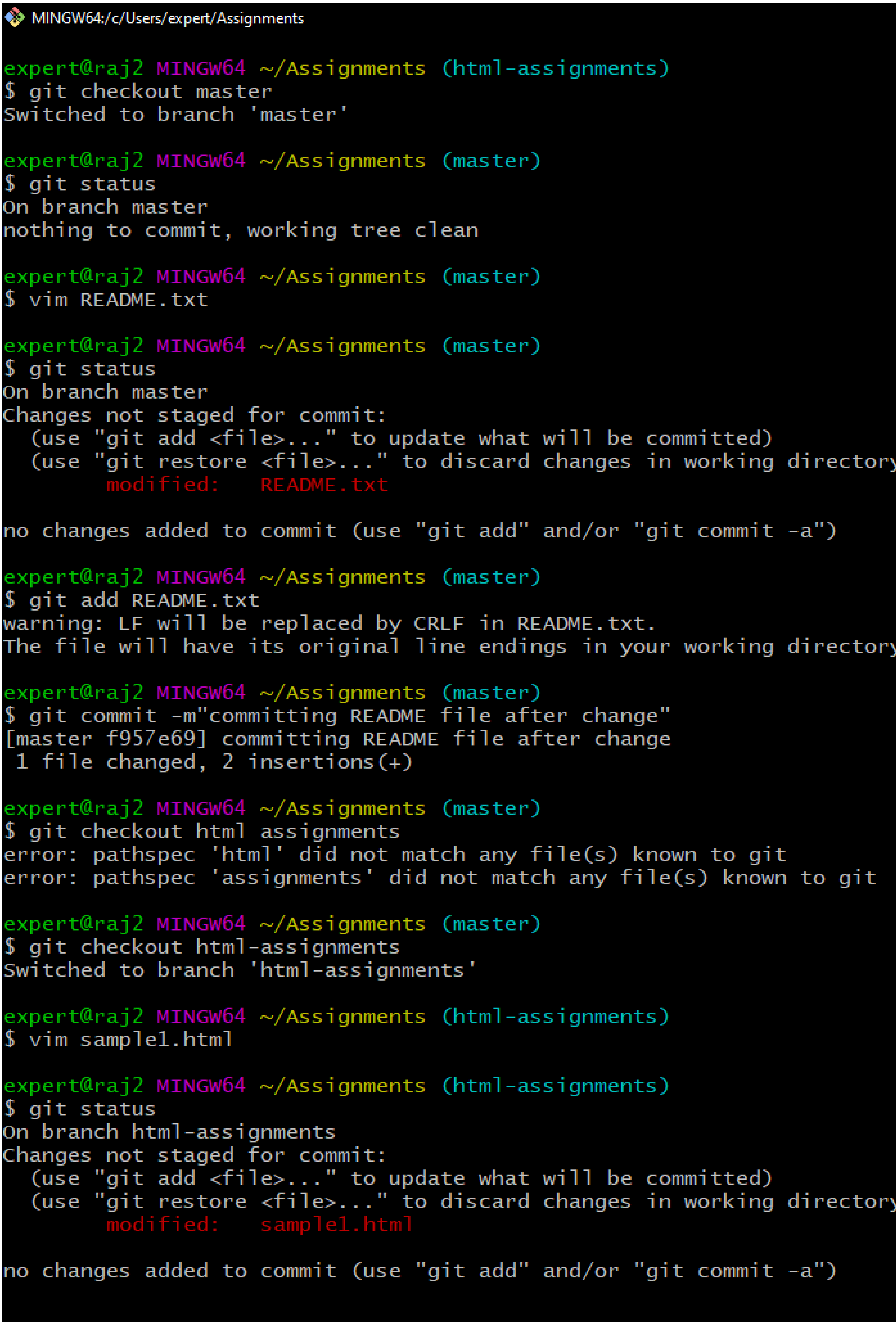
38. Finally delete the ‘html-assignments’ branch.



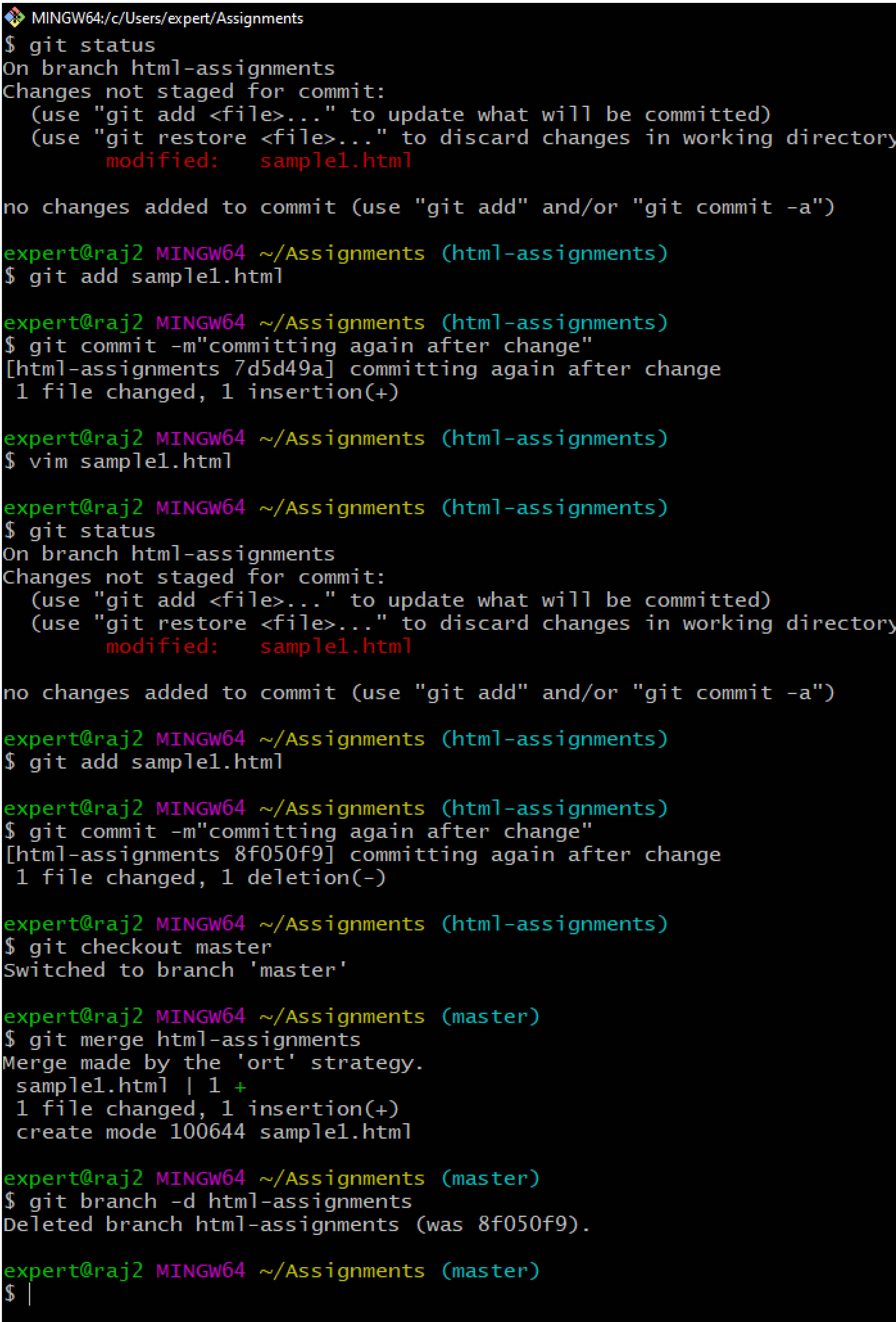
```
MINGW64:/c/Users/expert/Assignments

expert@raj2 MINGW64 ~
$ mkdir Assignments

expert@raj2 MINGW64 ~
$ cd Assignments

expert@raj2 MINGW64 ~/Assignments
$ git init
Initialized empty Git repository in C:/Users/expert/Assignments/.git/

expert@raj2 MINGW64 ~/Assignments (master)
$ touch README.txt

expert@raj2 MINGW64 ~/Assignments (master)
$ git add .

expert@raj2 MINGW64 ~/Assignments (master)
$ git status
On branch master

No commits yet

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


expert@raj2 MINGW64 ~/Assignments (master)
$ git commit -m"committing README.txt"
[master (root-commit) a0f4531] committing README.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.txt

expert@raj2 MINGW64 ~/Assignments (master)
$ git branch html-assignments

expert@raj2 MINGW64 ~/Assignments (master)
$ git status
On branch master
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/Assignments (master)
$ git add sample.html
fatal: pathspec 'sample.html' did not match any files

expert@raj2 MINGW64 ~/Assignments (master)
$ git add sample1.html
fatal: pathspec 'sample1.html' did not match any files

expert@raj2 MINGW64 ~/Assignments (master)
$ git status
On branch master
nothing to commit, working tree clean
```



```
expert@raj2 MINGW64 ~/Assignments (master)
$ touch sample1.html

expert@raj2 MINGW64 ~/Assignments (master)
$ git add .

expert@raj2 MINGW64 ~/Assignments (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   sample1.html


expert@raj2 MINGW64 ~/Assignments (master)
$ git checkout html-assignments
Switched to branch 'html-assignments'
A       sample1.html

expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ git add *.html

expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ git status
On branch html-assignments
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   sample1.html


expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ vim sample1.html

expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ git status
On branch html-assignments
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   sample1.html

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:   sample1.html


expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ git add sample1.html
warning: LF will be replaced by CRLF in sample1.html.
The file will have its original line endings in your working directory

expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ git commit -m"committing after change in html file"
[html-assignments 2e0053d] committing after change in html file
```



```
MINGW64:/c/Users/expert/Assignments

expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ git checkout master
Switched to branch 'master'

expert@raj2 MINGW64 ~/Assignments (master)
$ git status
On branch master
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/Assignments (master)
$ vim README.txt

expert@raj2 MINGW64 ~/Assignments (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:   README.txt

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

expert@raj2 MINGW64 ~/Assignments (master)
$ git add README.txt
warning: LF will be replaced by CRLF in README.txt.
The file will have its original line endings in your working directory

expert@raj2 MINGW64 ~/Assignments (master)
$ git commit -m"committing README file after change"
[master f957e69] committing README file after change
 1 file changed, 2 insertions(+)

expert@raj2 MINGW64 ~/Assignments (master)
$ git checkout html assignments
error: pathspec 'html' did not match any file(s) known to git
error: pathspec 'assignments' did not match any file(s) known to git

expert@raj2 MINGW64 ~/Assignments (master)
$ git checkout html-assignments
Switched to branch 'html-assignments'

expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ vim sample1.html

expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ git status
On branch html-assignments
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:   sample1.html

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



MINGW64:/c/Users/expert/Assignments

```
$ git status
On branch html-assignments
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:   sample1.html

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

expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ git add sample1.html

expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ git commit -m"committing again after change"
[html-assignments 7d5d49a] committing again after change
 1 file changed, 1 insertion(+)

expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ vim sample1.html

expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ git status
On branch html-assignments
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:   sample1.html

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

expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ git add sample1.html

expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ git commit -m"committing again after change"
[html-assignments 8f050f9] committing again after change
 1 file changed, 1 deletion(-)

expert@raj2 MINGW64 ~/Assignments (html-assignments)
$ git checkout master
Switched to branch 'master'

expert@raj2 MINGW64 ~/Assignments (master)
$ git merge html-assignments
Merge made by the 'ort' strategy.
 sample1.html | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 sample1.html

expert@raj2 MINGW64 ~/Assignments (master)
$ git branch -d html-assignments
Deleted branch html-assignments (was 8f050f9).

expert@raj2 MINGW64 ~/Assignments (master)
$ |
```

SECTION-2 - (CSS assignments) Steps to follow:

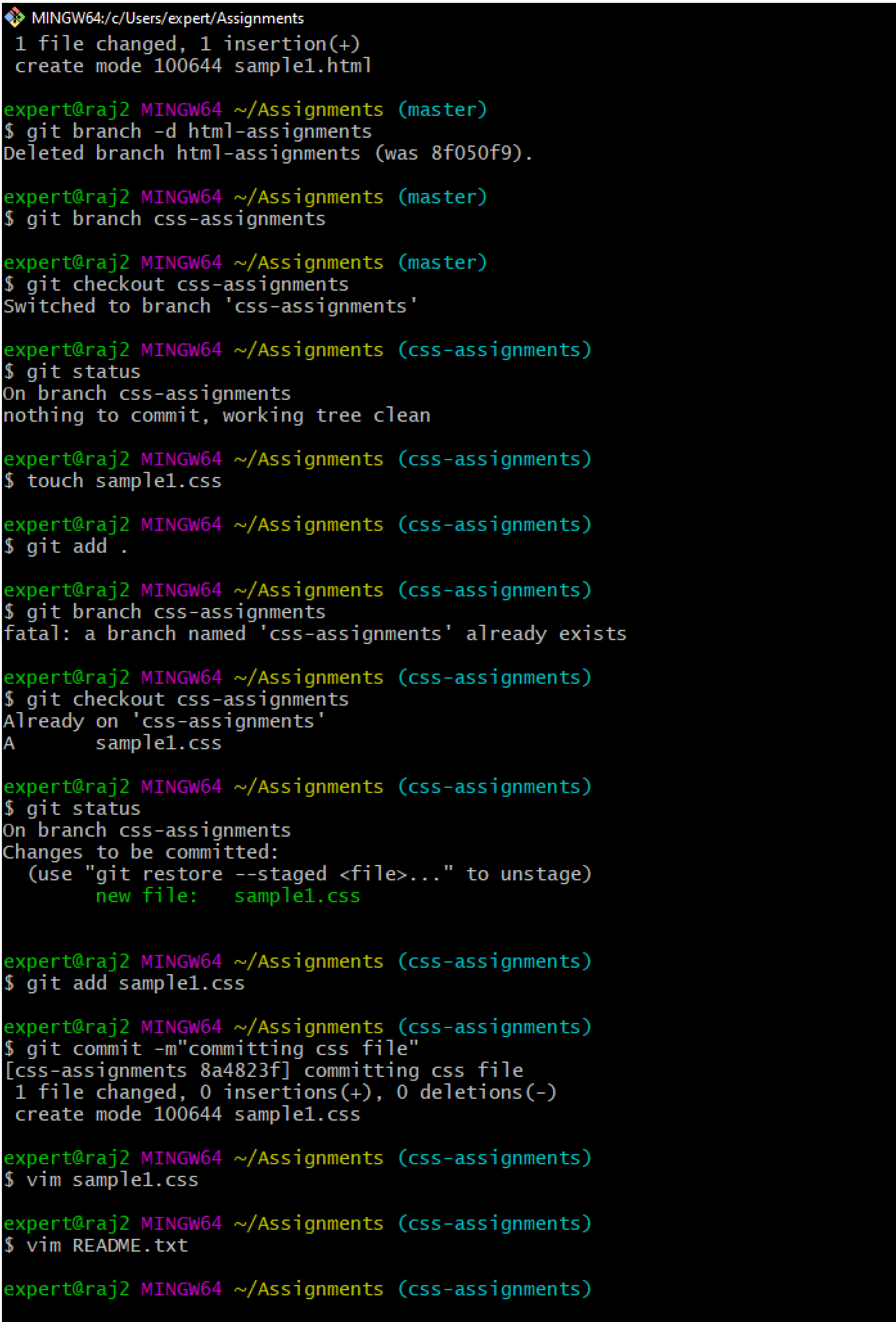
1. Create a new branch ‘css-assignments’.
2. Switch to ‘css-assignments’ branch.
3. Copy all CSS assignments inside ‘Assignments’ folder.
4. Commit CSS assignments into ‘css-assignments’ branch.
5. Make minor changes into README.txt file on line 1 belonging to ‘css-assignments’ branch.
6. Commit those changed files.
7. Switch to master branch.
8. Make minor changes into README.txt file on line 3 & commit those changes into master.
9. Again switch to ‘css-assignments’ branch.
10. Make minor changes into few files belonging to ‘css-assignments’ branch.

11. Commit those changes.

12. Switch to master.

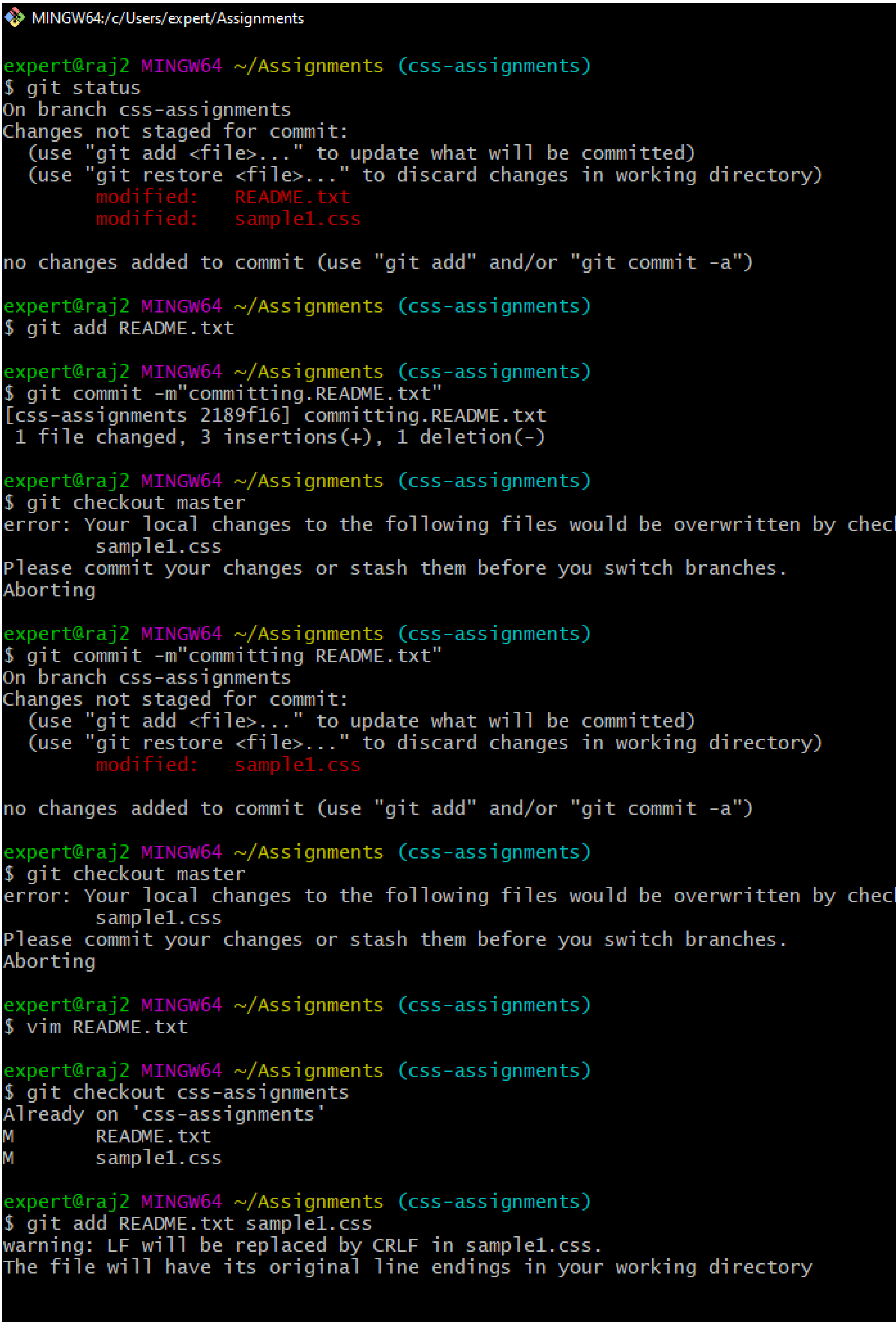
13. Merge ‘css-assignments’ branch into master. Confirm all css assignments are shown in master.

14. Finally delete the ‘css-assignments’ branch.

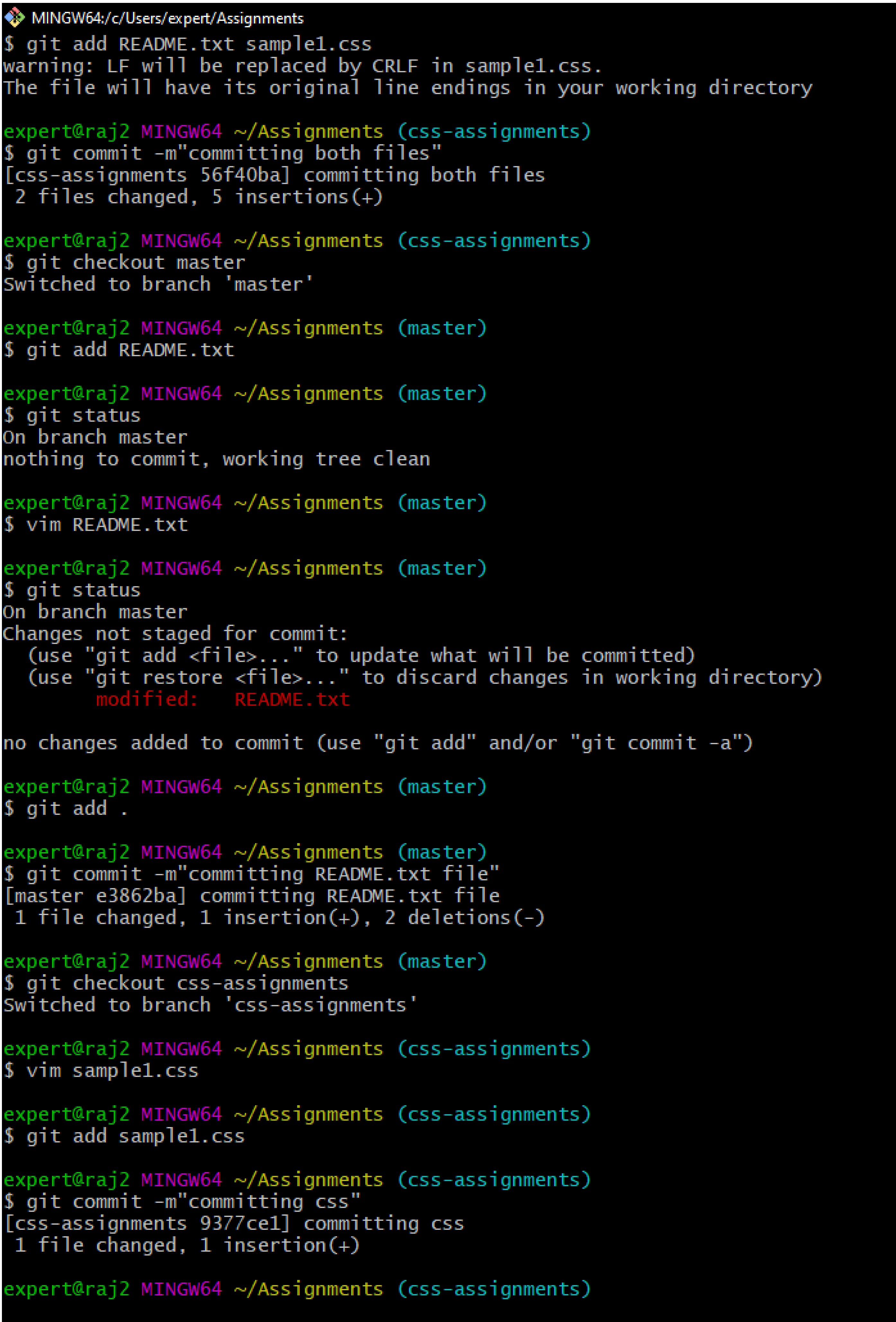


MINGW64:/c/Users/expert/Assignments

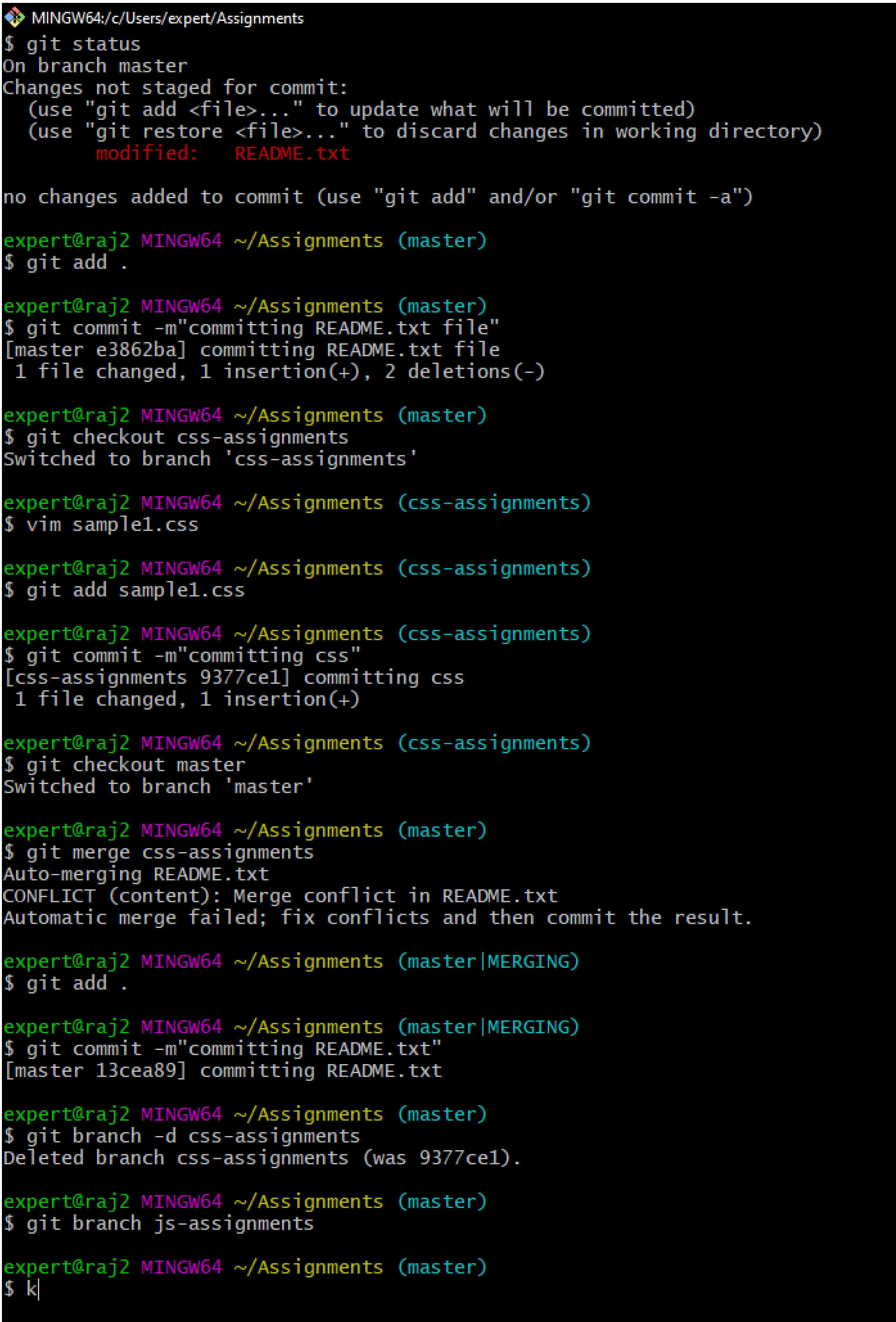
```
 1 file changed, 1 insertion(+)
 create mode 100644 sample1.html

expert@raj2 MINGW64 ~/Assignments (master)
$ git branch -d html-assignments
Deleted branch html-assignments (was 8f050f9).

expert@raj2 MINGW64 ~/Assignments (master)
$ git branch css-assignments

expert@raj2 MINGW64 ~/Assignments (master)
$ git checkout css-assignments
Switched to branch 'css-assignments'

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git status
On branch css-assignments
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ touch sample1.css

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git add .

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git branch css-assignments
fatal: a branch named 'css-assignments' already exists

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git checkout css-assignments
Already on 'css-assignments'
A       sample1.css

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git status
On branch css-assignments
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   sample1.css


expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git add sample1.css

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git commit -m"committing css file"
[css-assignments 8a4823f] committing css file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 sample1.css

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ vim sample1.css

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ vim README.txt

expert@raj2 MINGW64 ~/Assignments (css-assignments)
```



```
MINGW64:/c/Users/expert/Assignments

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git status
On branch css-assignments
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:   README.txt
        modified:   sample1.css

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

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git add README.txt

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git commit -m"committing.README.txt"
[css-assignments 2189f16] committing.README.txt
 1 file changed, 3 insertions(+), 1 deletion(-)

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git checkout master
error: Your local changes to the following files would be overwritten by chec
        sample1.css
Please commit your changes or stash them before you switch branches.
Aborting

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git commit -m"committing README.txt"
On branch css-assignments
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:   sample1.css

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

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git checkout master
error: Your local changes to the following files would be overwritten by chec
        sample1.css
Please commit your changes or stash them before you switch branches.
Aborting

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ vim README.txt

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git checkout css-assignments
Already on 'css-assignments'
M       README.txt
M       sample1.css

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git add README.txt sample1.css
warning: LF will be replaced by CRLF in sample1.css.
The file will have its original line endings in your working directory
```



```
MINGW64:/c/Users/expert/Assignments
$ git add README.txt sample1.css
warning: LF will be replaced by CRLF in sample1.css.
The file will have its original line endings in your working directory

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git commit -m"committing both files"
[css-assignments 56f40ba] committing both files
 2 files changed, 5 insertions(+)

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git checkout master
Switched to branch 'master'

expert@raj2 MINGW64 ~/Assignments (master)
$ git add README.txt

expert@raj2 MINGW64 ~/Assignments (master)
$ git status
On branch master
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/Assignments (master)
$ vim README.txt

expert@raj2 MINGW64 ~/Assignments (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:   README.txt

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

expert@raj2 MINGW64 ~/Assignments (master)
$ git add .

expert@raj2 MINGW64 ~/Assignments (master)
$ git commit -m"committing README.txt file"
[master e3862ba] committing README.txt file
 1 file changed, 1 insertion(+), 2 deletions(-)

expert@raj2 MINGW64 ~/Assignments (master)
$ git checkout css-assignments
Switched to branch 'css-assignments'

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ vim sample1.css

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git add sample1.css

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git commit -m"committing css"
[css-assignments 9377ce1] committing css
 1 file changed, 1 insertion(+)

expert@raj2 MINGW64 ~/Assignments (css-assignments)
```



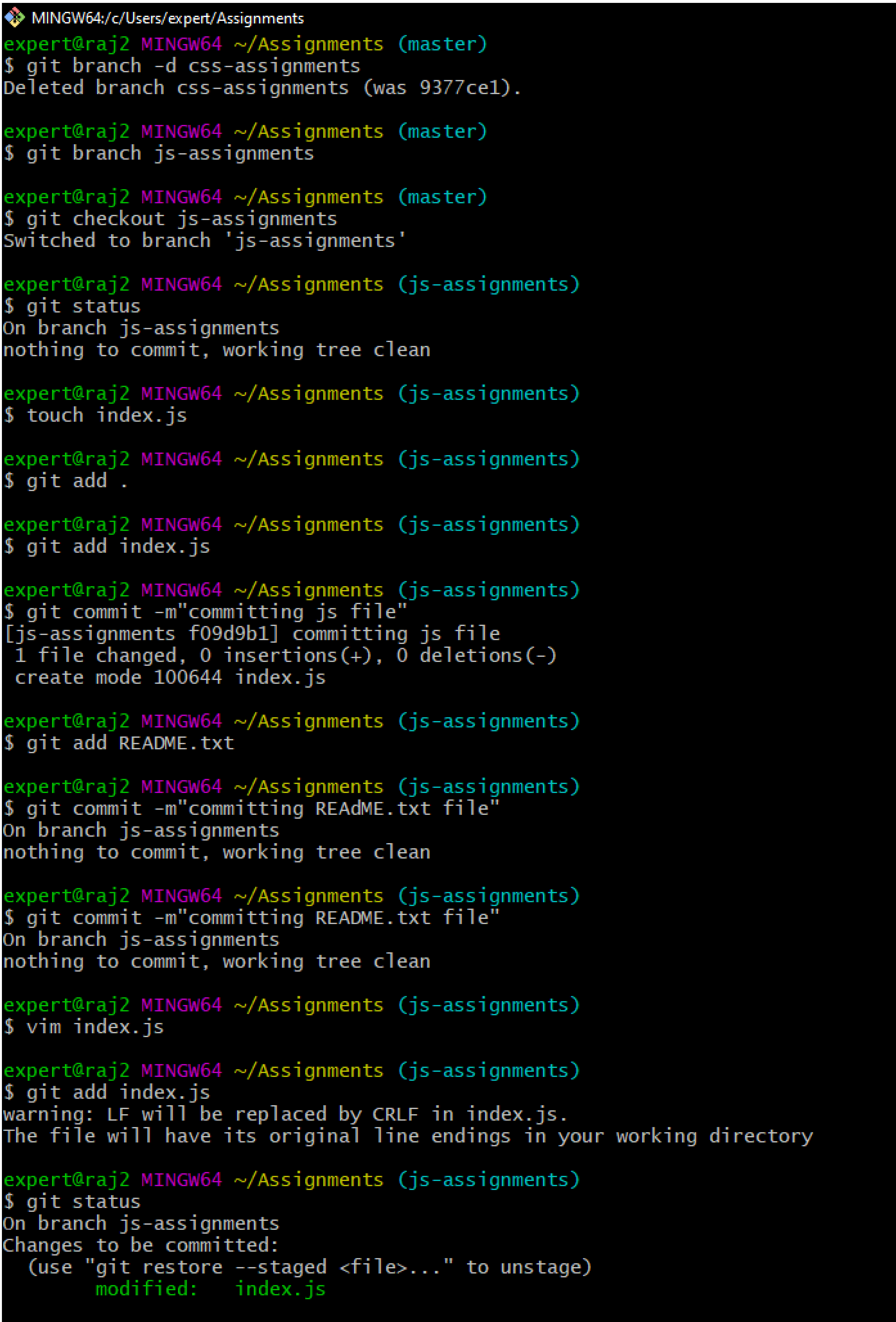
```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.txt

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

expert@raj2 MINGW64 ~/Assignments (master)
$ git add .

expert@raj2 MINGW64 ~/Assignments (master)
$ git commit -m"committing README.txt file"
[master e3862ba] committing README.txt file
 1 file changed, 1 insertion(+), 2 deletions(-)

expert@raj2 MINGW64 ~/Assignments (master)
$ git checkout css-assignments
Switched to branch 'css-assignments'

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ vim sample1.css

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git add sample1.css

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git commit -m"committing css"
[css-assignments 9377ce1] committing css
 1 file changed, 1 insertion(+)

expert@raj2 MINGW64 ~/Assignments (css-assignments)
$ git checkout master
Switched to branch 'master'

expert@raj2 MINGW64 ~/Assignments (master)
$ git merge css-assignments
Auto-merging README.txt
CONFLICT (content): Merge conflict in README.txt
Automatic merge failed; fix conflicts and then commit the result.

expert@raj2 MINGW64 ~/Assignments (master|MERGING)
$ git add .

expert@raj2 MINGW64 ~/Assignments (master|MERGING)
$ git commit -m"committing README.txt"
[master 13cea89] committing README.txt

expert@raj2 MINGW64 ~/Assignments (master)
$ git branch -d css-assignments
Deleted branch css-assignments (was 9377ce1).

expert@raj2 MINGW64 ~/Assignments (master)
$ git branch js-assignments

expert@raj2 MINGW64 ~/Assignments (master)
$ k
```

SECTION-3 - (JavaScript assignments) Steps to follow:

1. Create a new branch ‘js-assignments’.
2. Switch to ‘js-assignments’ branch.
3. Copy all JavaScript assignments inside ‘Assignments’ folder.
4. Commit JavaScript assignments into ‘js-assignments’ branch.
5. Make minor changes into README.txt file on line 1 belonging to ‘js-assignments’ branch.
6. Commit those changed files.
7. Switch to master branch.
8. Make minor changes into README.txt file on line 1 & commit those changes into master.
9. Again switch to ‘js-assignments’ branch.
10. Make minor changes into few files belonging to ‘js-assignments’ branch.
11. Commit those changes.
12. Switch to master.

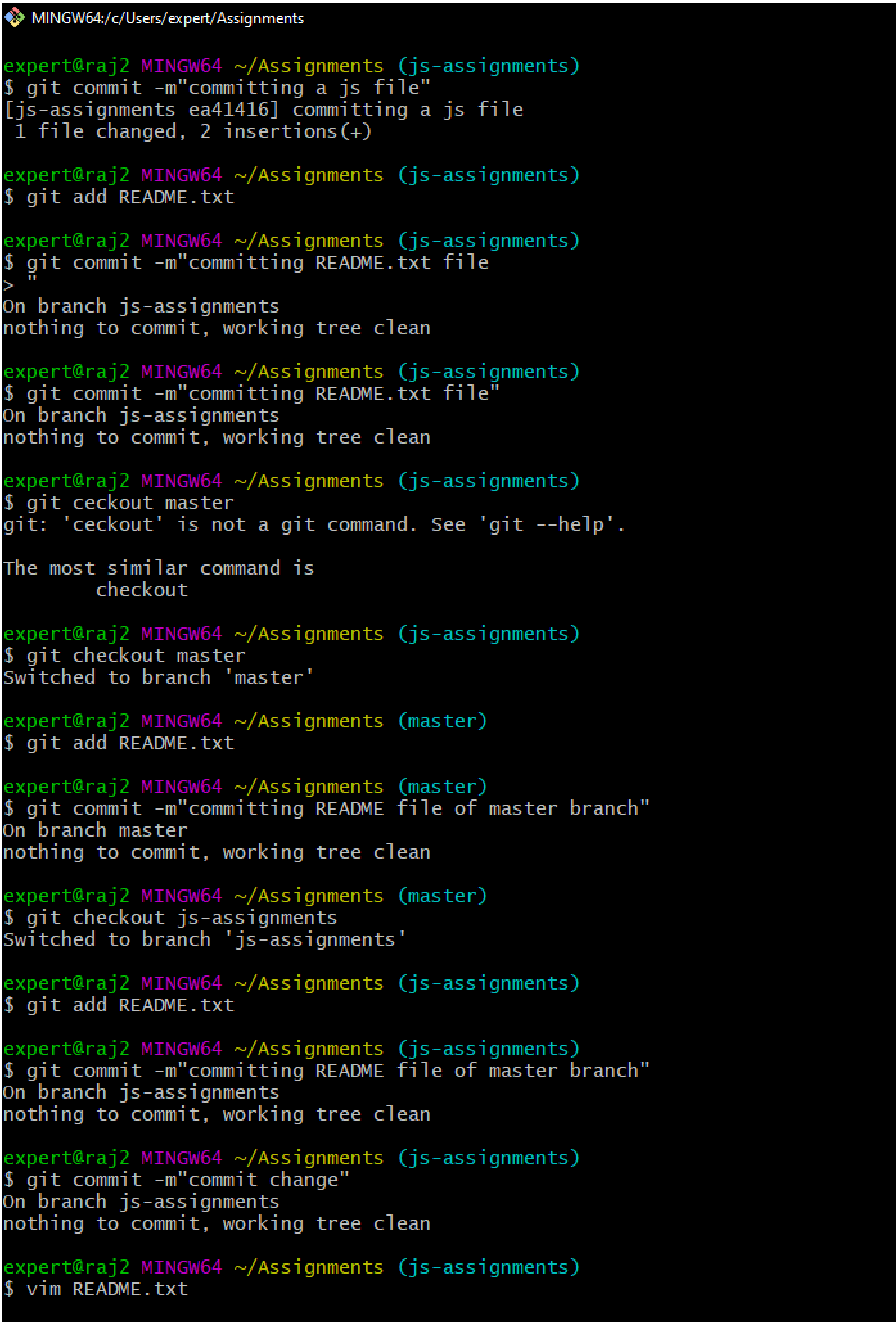
13. Merge ‘js-assignments’ branch into master. Confirm all JavaScript assignments are shown in master.

14. Finally delete the ‘js-assignments’ branch.

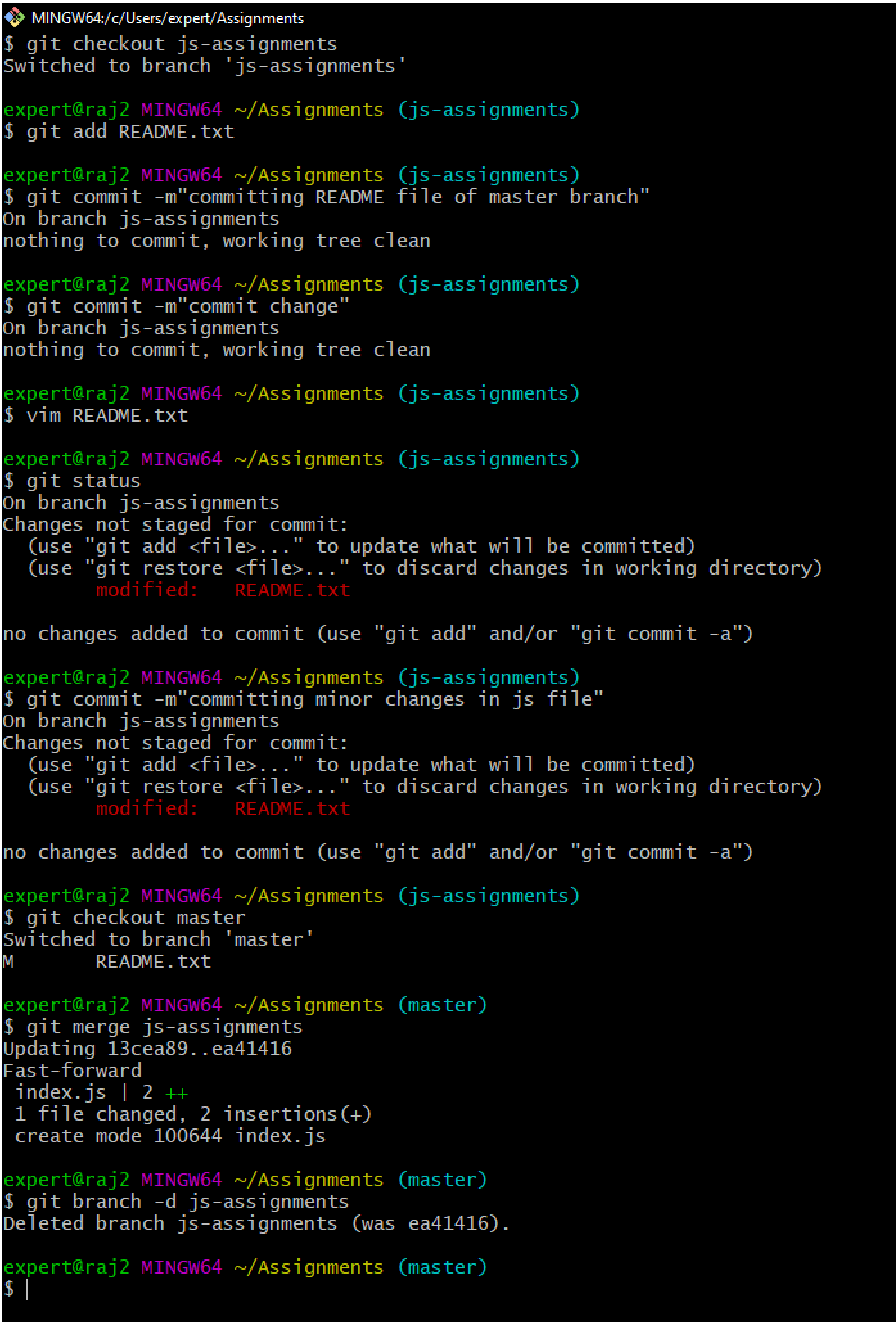


MINGW64:/c/Users/expert/Assignments

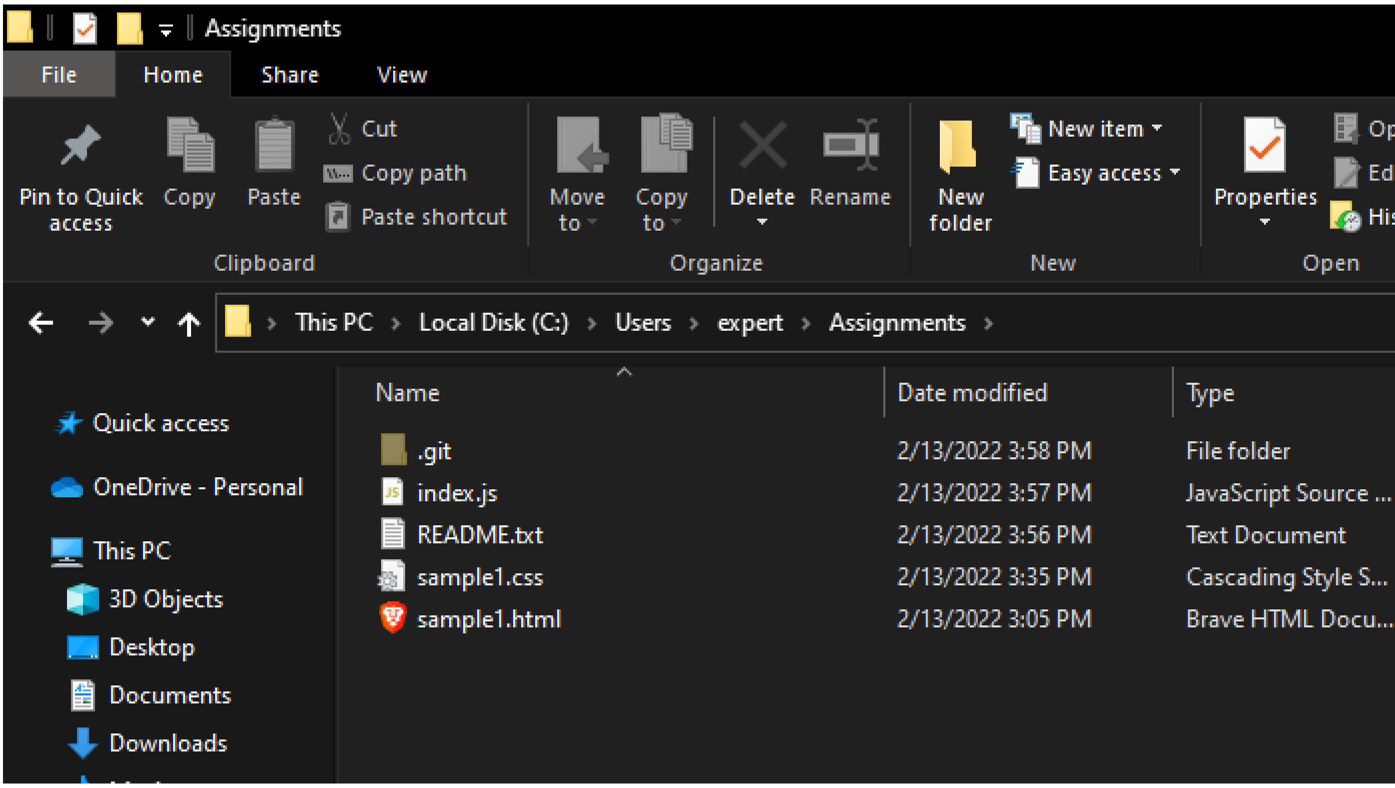
```
expert@raj2 MINGW64 ~/Assignments (master)
$ git branch -d css-assignments
Deleted branch css-assignments (was 9377ce1).

expert@raj2 MINGW64 ~/Assignments (master)
$ git branch js-assignments

expert@raj2 MINGW64 ~/Assignments (master)
$ git checkout js-assignments
Switched to branch 'js-assignments'

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git status
On branch js-assignments
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ touch index.js

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git add .

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git add index.js

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git commit -m"committing js file"
[js-assignments f09d9b1] committing js file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.js

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git add README.txt

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git commit -m"committing README.txt file"
On branch js-assignments
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git commit -m"committing README.txt file"
On branch js-assignments
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ vim index.js

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git add index.js
warning: LF will be replaced by CRLF in index.js.
The file will have its original line endings in your working directory

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git status
On branch js-assignments
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.js
```



```
expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git commit -m"committing a js file"
[js-assignments ea41416] committing a js file
 1 file changed, 2 insertions(+)

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git add README.txt

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git commit -m"committing README.txt file
> "
On branch js-assignments
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git commit -m"committing README.txt file"
On branch js-assignments
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git ceckout master
git: 'ceckout' is not a git command. See 'git --help'.

The most similar command is
        checkout

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git checkout master
Switched to branch 'master'

expert@raj2 MINGW64 ~/Assignments (master)
$ git add README.txt

expert@raj2 MINGW64 ~/Assignments (master)
$ git commit -m"committing README file of master branch"
On branch master
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/Assignments (master)
$ git checkout js-assignments
Switched to branch 'js-assignments'

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git add README.txt

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git commit -m"committing README file of master branch"
On branch js-assignments
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git commit -m"commit change"
On branch js-assignments
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ vim README.txt
```



```
MINGW64:/c/Users/expert/Assignments
$ git checkout js-assignments
Switched to branch 'js-assignments'

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git add README.txt

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git commit -m"committing README file of master branch"
On branch js-assignments
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git commit -m"commit change"
On branch js-assignments
nothing to commit, working tree clean

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ vim README.txt

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git status
On branch js-assignments
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:   README.txt

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

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git commit -m"committing minor changes in js file"
On branch js-assignments
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:   README.txt

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

expert@raj2 MINGW64 ~/Assignments (js-assignments)
$ git checkout master
Switched to branch 'master'
M       README.txt

expert@raj2 MINGW64 ~/Assignments (master)
$ git merge js-assignments
Updating 13cea89..ea41416
Fast-forward
 index.js | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 index.js

expert@raj2 MINGW64 ~/Assignments (master)
$ git branch -d js-assignments
Deleted branch js-assignments (was ea41416).

expert@raj2 MINGW64 ~/Assignments (master)
$ |
```



This PC › Local Disk (C:) › Users › expert › Assignments ›

Name	Date modified	Type
.git	2/13/2022 3:58 PM	File folder
index.js	2/13/2022 3:57 PM	JavaScript Source ...
README.txt	2/13/2022 3:56 PM	Text Document
sample1.css	2/13/2022 3:35 PM	Cascading Style S...
sample1.html	2/13/2022 3:05 PM	Brave HTML Docu...