Section-0

1. Create a directory ‘project\_dir’ & cd to ‘project\_dir ’.

**$ mkdir project\_dir**

**$ cd project\_dir**

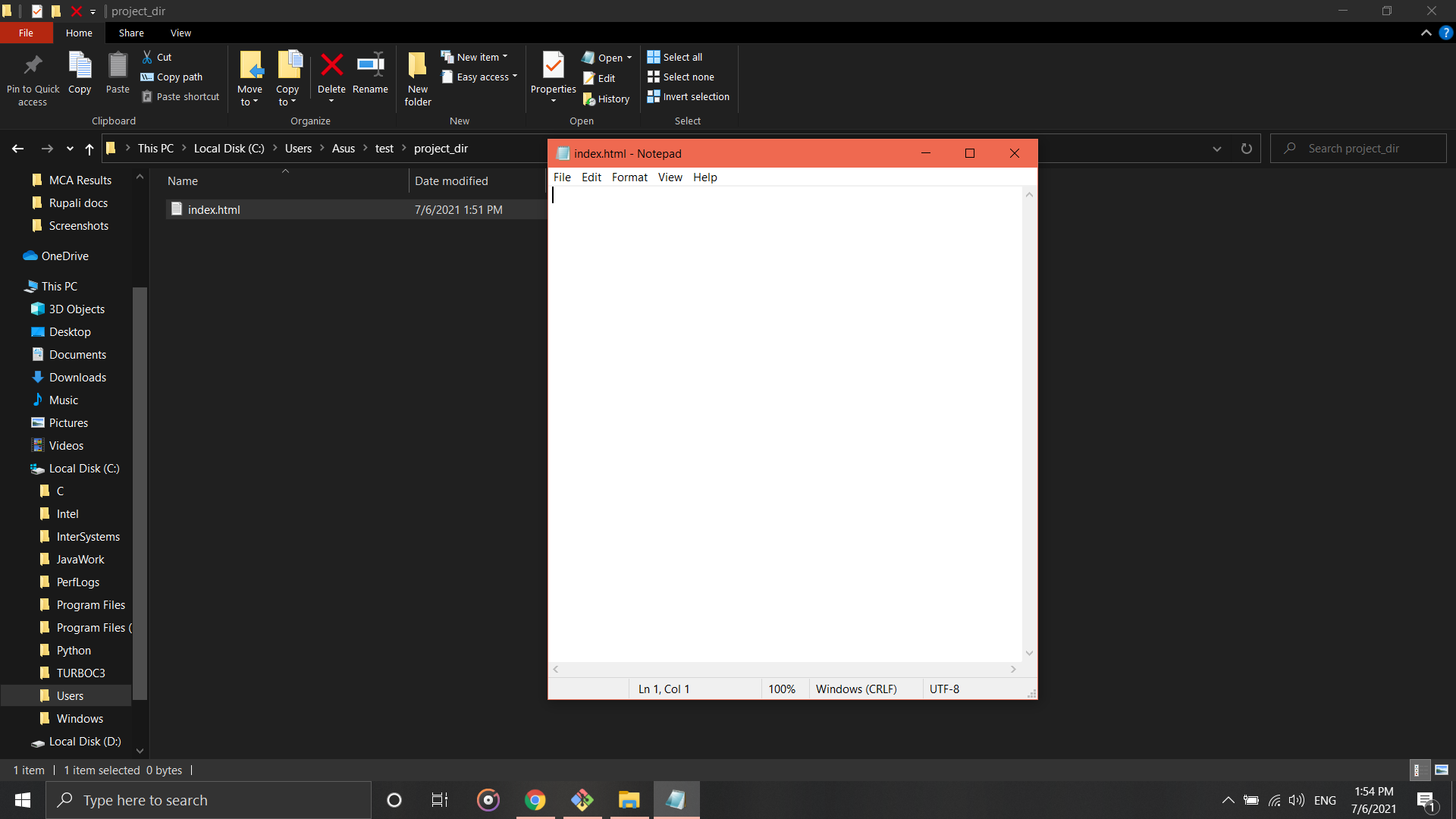
1. Initialize git version database.

**$ git init**

Initialized empty Git repository in C:/Users/Asus/test/project\_dir/.git/

1. Create a new file index.html

**$ touch index.html**



1. Check the git status. You should find index.html as untracked file.

**$ git status**

On branch master

No commits yet

Untracked files:

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

index.html.txt

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

1. Stage the index.html file.

**$ git add index.html.txt**

1. Commit index.html

**$ git commit -m "index file committed"**

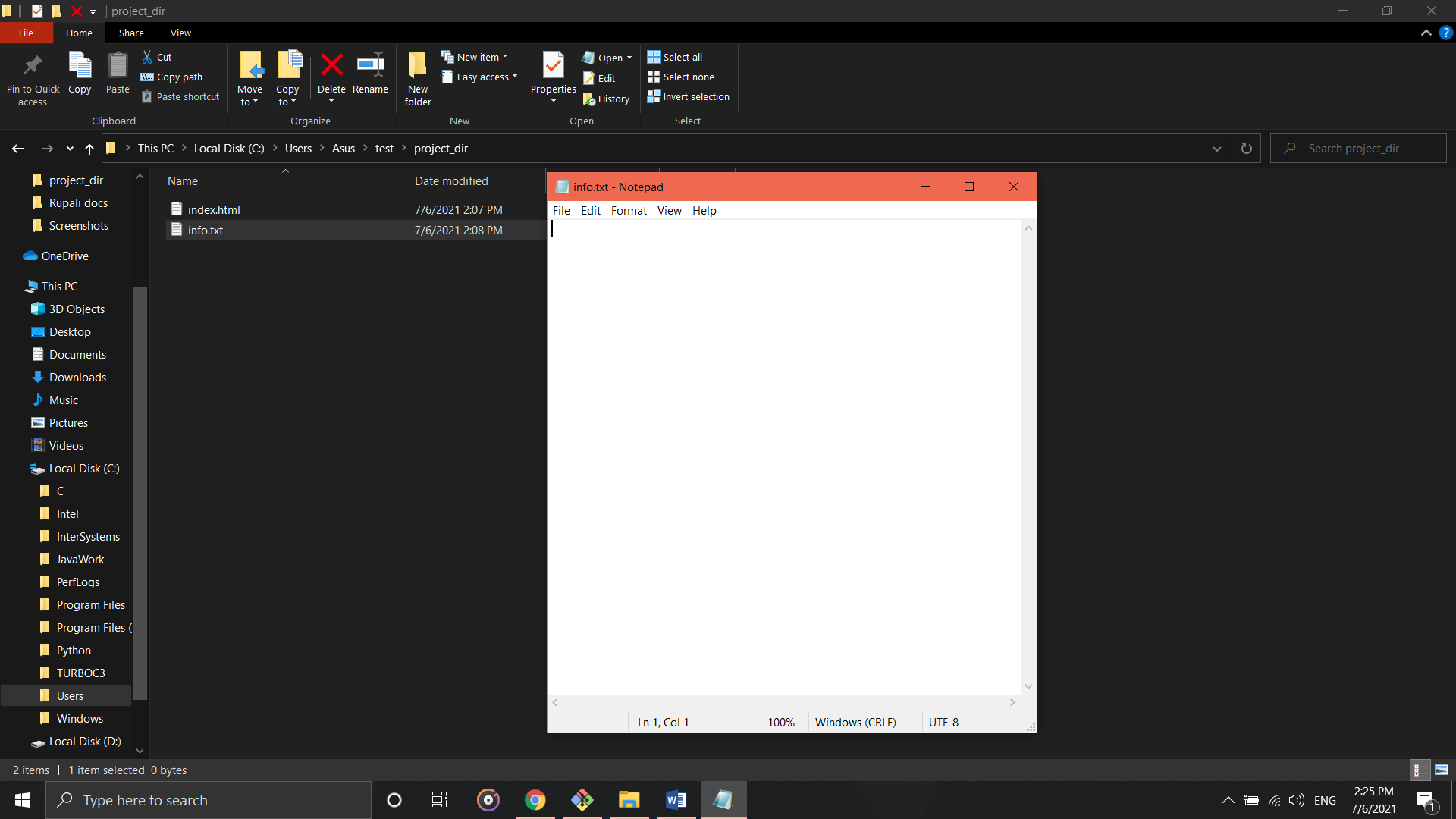
[master (root-commit) 4d0a520] index file committed

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 index.html.txt

1. Make few changes in index.html & create a new file info.txt file.

**$ touch info.txt**



1. Check git status. You should find index.html & info.html as untracked files.

**$ 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.txt

Untracked files:

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

info.txt.txt

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

1. Configure GIT to ignore all txt files.

$ touch .gitignore

$ touch file.txt

$ git add \*.txt

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

**$ git status**

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: file.txt

new file: info.txt.txt

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.txt

Untracked files:

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

.gitignore

1. State and commit index.html

**$ git commit -m "index file committed"**

[master 00de51d] index file committed

2 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 file.txt

create mode 100644 info.txt.txt

1. Log all your commands so far.

**$ git log**

commit 00de51def1e1f61fc94f399844099bbff5f625a5 (HEAD -> master)

Author: MuskaanSingh09 <singhmuskaan@gmail.com>

Date: Tue Jul 6 16:58:51 2021 +0530

index file committed

commit 4d0a520b3396d580e98408cfda8d9ba93428f395

Author: MuskaanSingh09 <singhmuskaan@gmail.com>

Date: Tue Jul 6 14:05:45 2021 +0530

index file committed

1. Make some changes in index.html.

**Made changes in the file**

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

$ git checkout

M index.html.txt

1. Again change index.html.

**Made changes in the file**

1. Stage index.html

**$ git commit -m "changes done in index.html file"**

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 track)

1. Revert back the last stage.

**$ git checkout –**

M index.html.txt

1. Rename ‘add’ command to ‘my-add’.

**$ git config --global alias.my-add add**

1. Using my-add command stage index.html again & commit the changes.

**$ git commit -m "index.html file is committed after renaming add to my-add"**

[master d684245] index.html file is committed after renaming add to my-add

1 file changed, 3 insertions(+)

1. Revert the last commit.

**$ git checkout --**

**GIT Branching**

Section-1 (HTML Assignments)

1. First take a backup of your assignments & projects. This is required because due to incorrect GIT operation you may lose your files.

* Done with the backup

1. Create an empty directory ‘Assignments’ & cd to ‘Assignments.
2. Create a file README.txt inside ‘Assignments’ & write few lines about the contents of ‘Assignments’ folder.
3. Commit README.txt file
4. Now create a new branch ‘html assignment’.
5. Switch to ‘html assignment’ branch.
6. Copy all html assignment inside “Assignment” folder.
7. Commit HTML assignments into ‘html assignment’ branch
8. Make minor changes into few files belonging to ‘html assignment’ branch.
9. Commit those changed files.
10. Switch to master branch.
11. Make minor changes into README.txt file & commit those changes into master.
12. Again switch to ‘html assignment’ branch.
13. Make minor changes into few files belonging to ‘html assignment’ branch.
14. Commit those changes.
15. Switch to master.
16. Merge ‘html assignment’ branch into master
17. Finally delete the ‘html assignment’ branch.

**Lines of Code**

Asus@DESKTOP-514VHJC MINGW64 ~/test (master)

$ mkdir Assignments

Asus@DESKTOP-514VHJC MINGW64 ~/test (master)

$ cd Assignments

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ touch README.txt

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git init

Initialized empty Git repository in C:/Users/Asus/test/Assignments/.git/

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git config --global user.name 'MuskaanSingh09'

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git config --global user.email 'singhmuskaan09@gmail.com'

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git add README.txt

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git commit -m 'Committed Readme.txt file'

[master (root-commit) e5f2127] Committed Readme.txt file

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 README.txt

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git branch html-assignments

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git checkout html-assignments

Switched to branch 'html-assignments'

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ touch H1.html

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ touch H2.html

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git add H1.html H2.html

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git commit -m 'H1 and H2 html assignments committed in html-assignments branch'

[html-assignments 62545ad] H1 and H2 html assignments committed in html-assignments branch

2 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 H1.html

create mode 100644 H2.html

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git checkout --

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git add H1.html

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git commit -m 'H1 html file changed commit'

On branch html-assignments

nothing to commit, working tree clean

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git checkout master

Switched to branch 'master'

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git add README.txt

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git commit -m 'Modified & committed README.txt'

On branch master

nothing to commit, working tree clean

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git checkout html-assignments

Switched to branch 'html-assignments'

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git add H1.html

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git commit -m 'H1.html file changed and committed'

On branch html-assignments

nothing to commit, working tree clean

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git add H2.html

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git commit -m 'H2.html file changed and committed'

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: H1.html

modified: H2.html

modified: README.txt

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

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git add H1.html

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git commit -m 'H1.html file changed and committed'

[html-assignments f249ba7] H1.html file changed and committed

1 file changed, 9 insertions(+)

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git add H2.html

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git commit -m 'H2.html file changed and committed'

[html-assignments 2c3b595] H2.html file changed and committed

1 file changed, 9 insertions(+)

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (html-assignments)

$ git checkout master

Switched to branch 'master'

M README.txt

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git merge html-assignments

Updating e5f2127..2c3b595

Fast-forward

H1.html | 9 +++++++++

H2.html | 9 +++++++++

2 files changed, 18 insertions(+)

create mode 100644 H1.html

create mode 100644 H2.html

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git branch -d html-assignments

Deleted branch html-assignments (was 2c3b595).

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git status

On branch master

Changes not staged for commit:

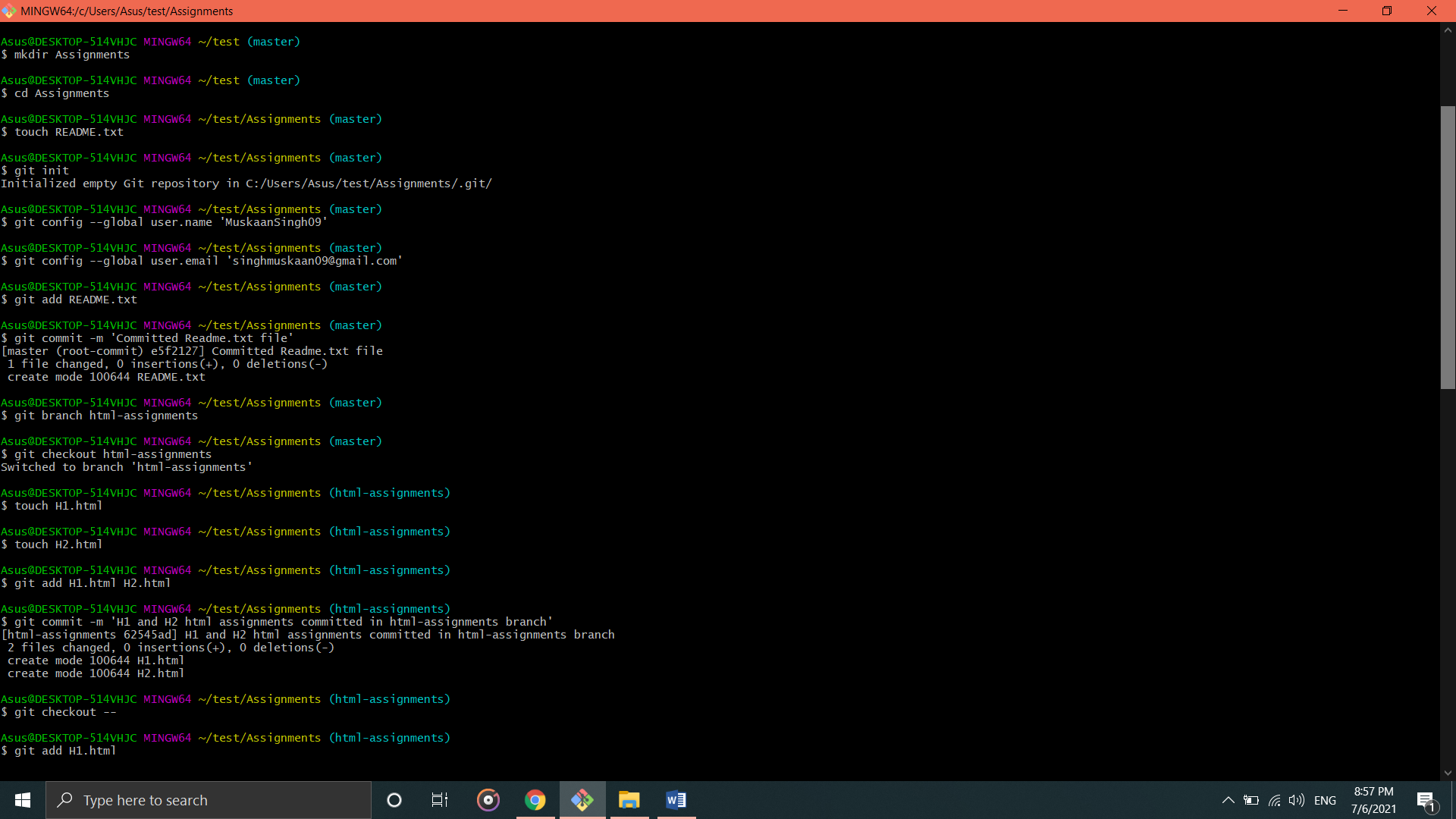
(use "git add <file>..." to update what will be committed)

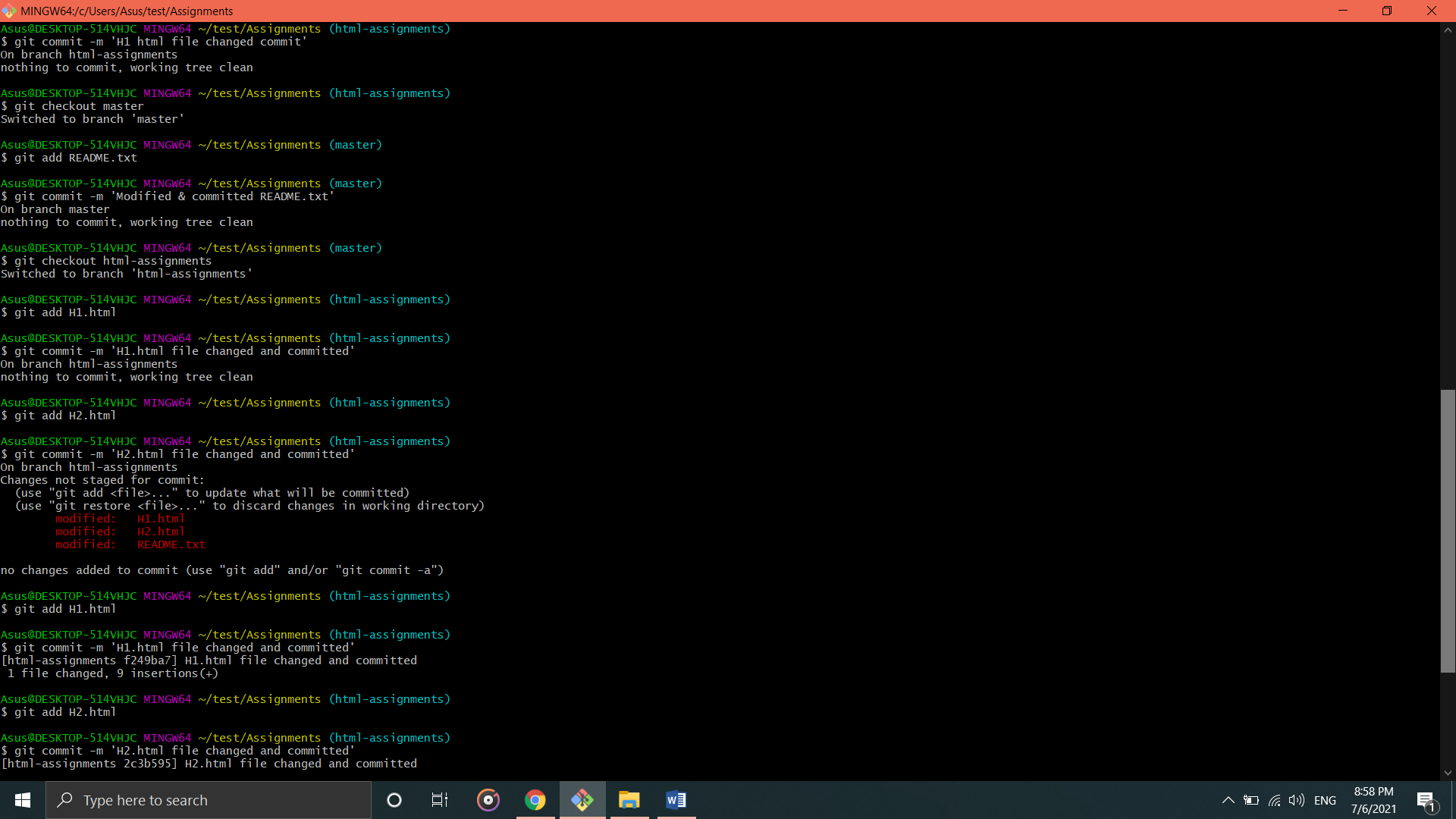
(use "git restore <file>..." to discard changes in working directory)

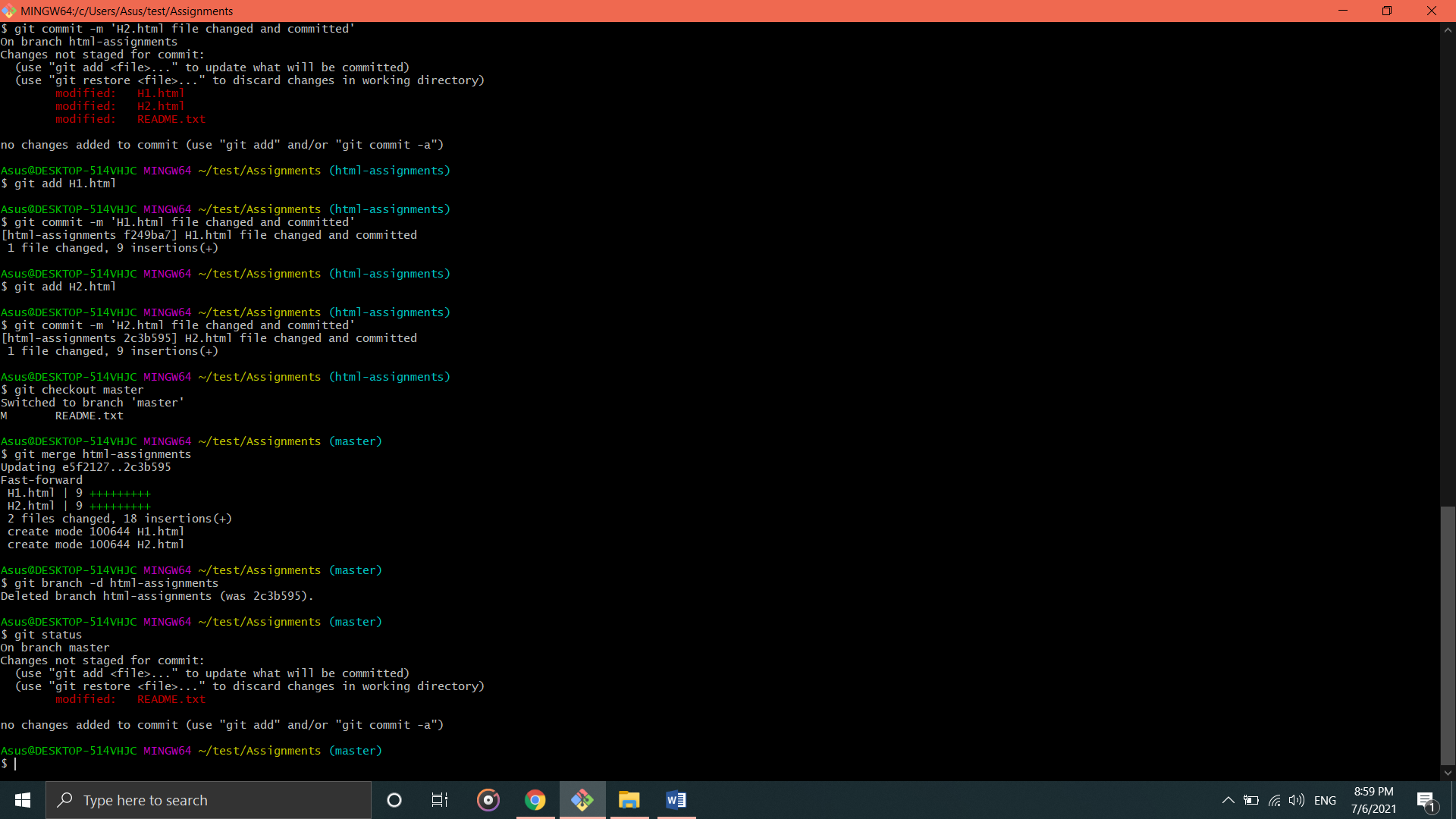
modified: README.txt

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

**Screenshots of task performed**







Section-2 (CSS Assignments)

1. Create a new branch ‘css-assignments’.
2. Switch to ‘css-assignments’ branch.
3. Copy all CSS assignment 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’.

**Lines of code**

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git branch css-assignments

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git checkout css-assignments

Switched to branch 'css-assignments'

M README.txt

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (css-assignments)

$ touch index.css

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (css-assignments)

$ touch style.css

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (css-assignments)

$ git add index.css style.css

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (css-assignments)

$ git commit -m "css assignments committed"

[css-assignments 08deecb] css assignments committed

2 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 index.css

create mode 100644 style.css

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (css-assignments)

$ git add README.txt

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (css-assignments)

$ git commit -m "README.txt files committed after modification"

[css-assignments b42a0d8] README.txt files committed after modification

1 file changed, 9 insertions(+)

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (css-assignments)

$ git checkout master

Switched to branch 'master'

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git add README.txt

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git commit -m "README.txt files committed second time after modification"

[master 3375aa2] README.txt files committed second time after modification

1 file changed, 9 insertions(+)

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git checkout css-assignments

Switched to branch 'css-assignments'

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (css-assignments)

$ git add index.css

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (css-assignments)

$ git commit -m "index.css file committed after changes"

[css-assignments 01df474] index.css file committed after changes

1 file changed, 21 insertions(+)

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (css-assignments)

$ git add style.css

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (css-assignments)

$ git commit -m "style.css file committed after changes"

[css-assignments d557f97] style.css file committed after changes

1 file changed, 9 insertions(+)

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (css-assignments)

$ git checkout master

Switched to branch 'master'

Asus@DESKTOP-514VHJC MINGW64 ~/test/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.

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master|MERGING)

$ git add README.txt

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master|MERGING)

$ git commit -m "README.txt files committed after modification"

[master 6cab9bc] README.txt files committed after modification

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git merge css-assignments

Already up to date.

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git branch -d css.assignments

error: branch 'css.assignments' not found.

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

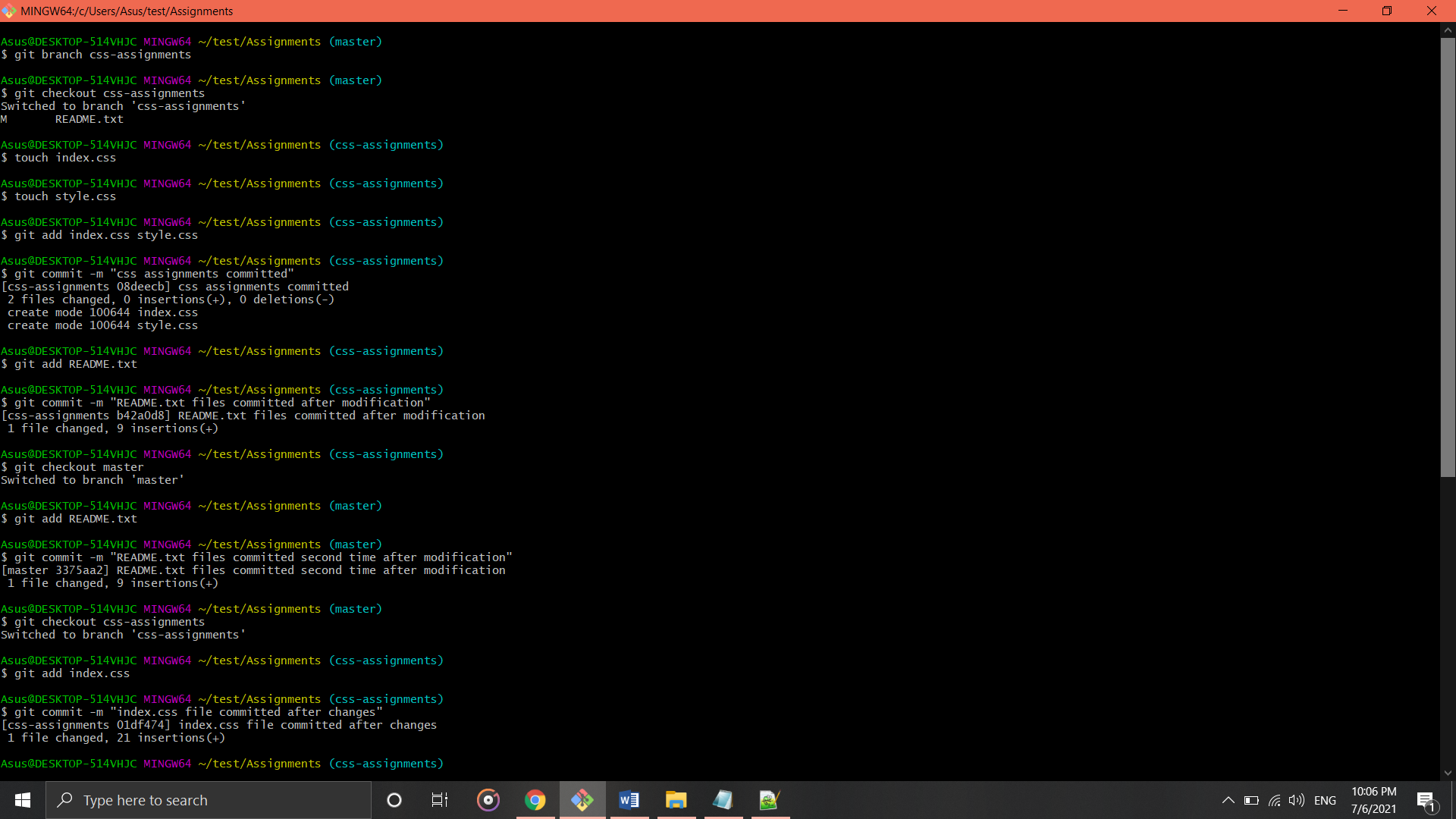
$ git branch -d css-assignments

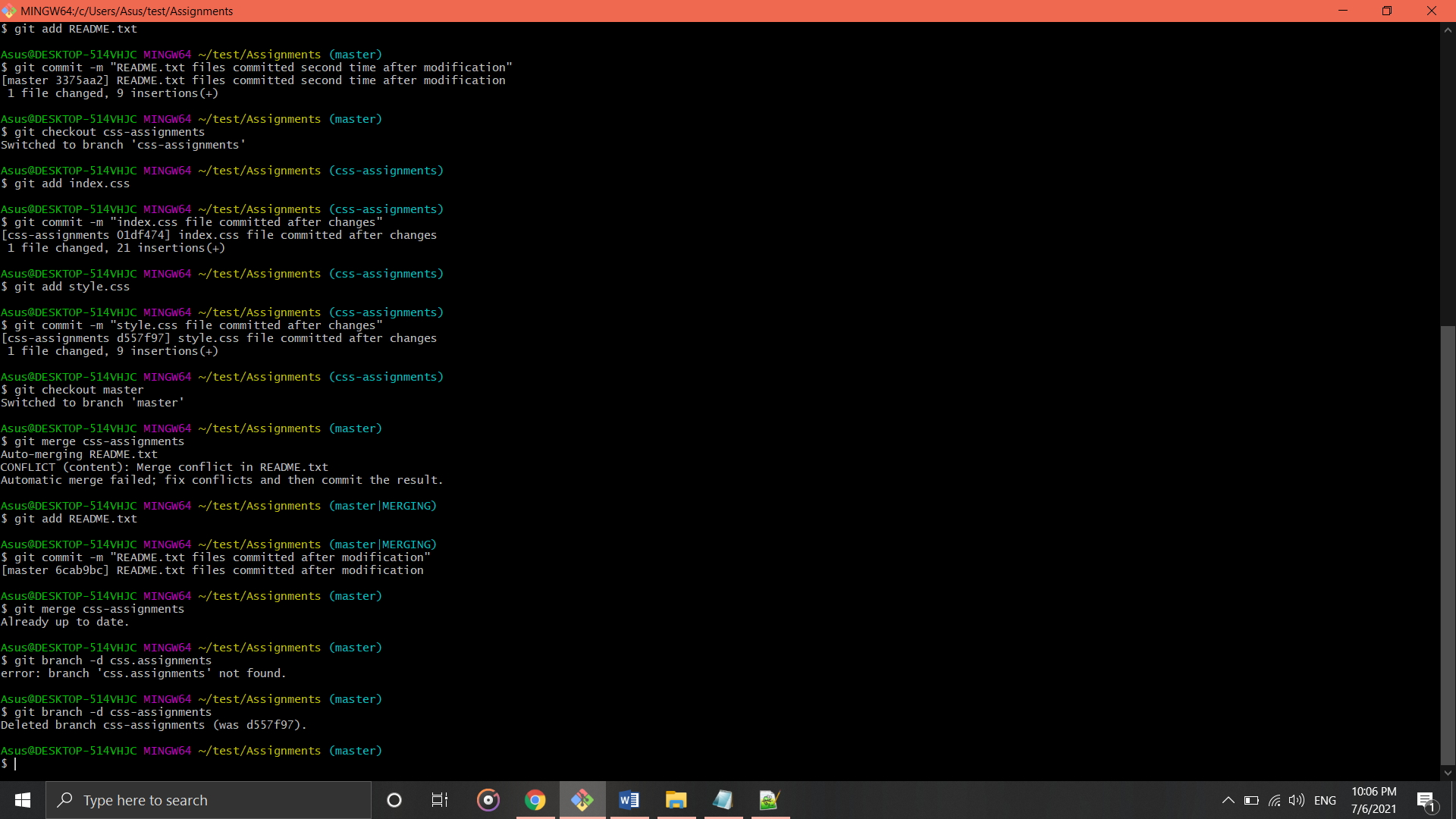
Deleted branch css-assignments (was d557f97).

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$

**Screenshots**

****



Section-3 (JavaScript assignments)

1. Create a new branch ‘js-assignments’.
2. Switch to ‘js-assignments’ branch.
3. Copy all JS assignment inside ‘Assignments’ folder.
4. Commit JS 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 3 & 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 JS assignments are shown in master.
14. Finally delete the ‘js-assignments’.

Codes

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git branch js-assignments

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git checkout js-assignments

Switched to branch 'js-assignments'

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (js-assignments)

$ touch J1.js

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (js-assignments)

$ touch J2.js

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (js-assignments)

$ git add J1.js J2.js

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (js-assignments)

$ git commit -m "JS assignments committed"

[js-assignments 79282ae] JS assignments committed

2 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 J1.js

create mode 100644 J2.js

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (js-assignments)

$ git add README.txt

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (js-assignments)

$ git commit -m "README.txt files get committed for the first time after changes"

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")

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (js-assignments)

$ git checkout master

Switched to branch 'master'

M README.txt

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git add README.txt

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git commit -m "README.txt files get committed for the second time after changes"

[master 413e373] README.txt files get committed for the second time after changes

1 file changed, 3 insertions(+), 5 deletions(-)

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git checkout js-assignments

Switched to branch 'js-assignments'

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (js-assignments)

$ git add J1.js

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (js-assignments)

$ git commit -m "J1.js files get committed for the first time after changes"

[js-assignments 4677f8d] J1.js files get committed for the first time after changes

1 file changed, 1 insertion(+)

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (js-assignments)

$ git add J2.js

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (js-assignments)

$ git commit -m "J2.js files get committed for the first time after changes"

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: J2.js

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

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (js-assignments)

$ git add J2.js

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (js-assignments)

$ git commit -m "J2.js files get committed for the first time after changes"

[js-assignments a0bfa93] J2.js files get committed for the first time after changes

1 file changed, 1 insertion(+)

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (js-assignments)

$ git checkout master

Switched to branch 'master'

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git add README.txt

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git commit -m "committed after changes"

On branch master

nothing to commit, working tree clean

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git merge js-assignments

Merge made by the 'recursive' strategy.

J1.js | 1 +

J2.js | 1 +

2 files changed, 2 insertions(+)

create mode 100644 J1.js

create mode 100644 J2.js

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$ git branch -d js.assignments

error: branch 'js.assignments' not found.

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

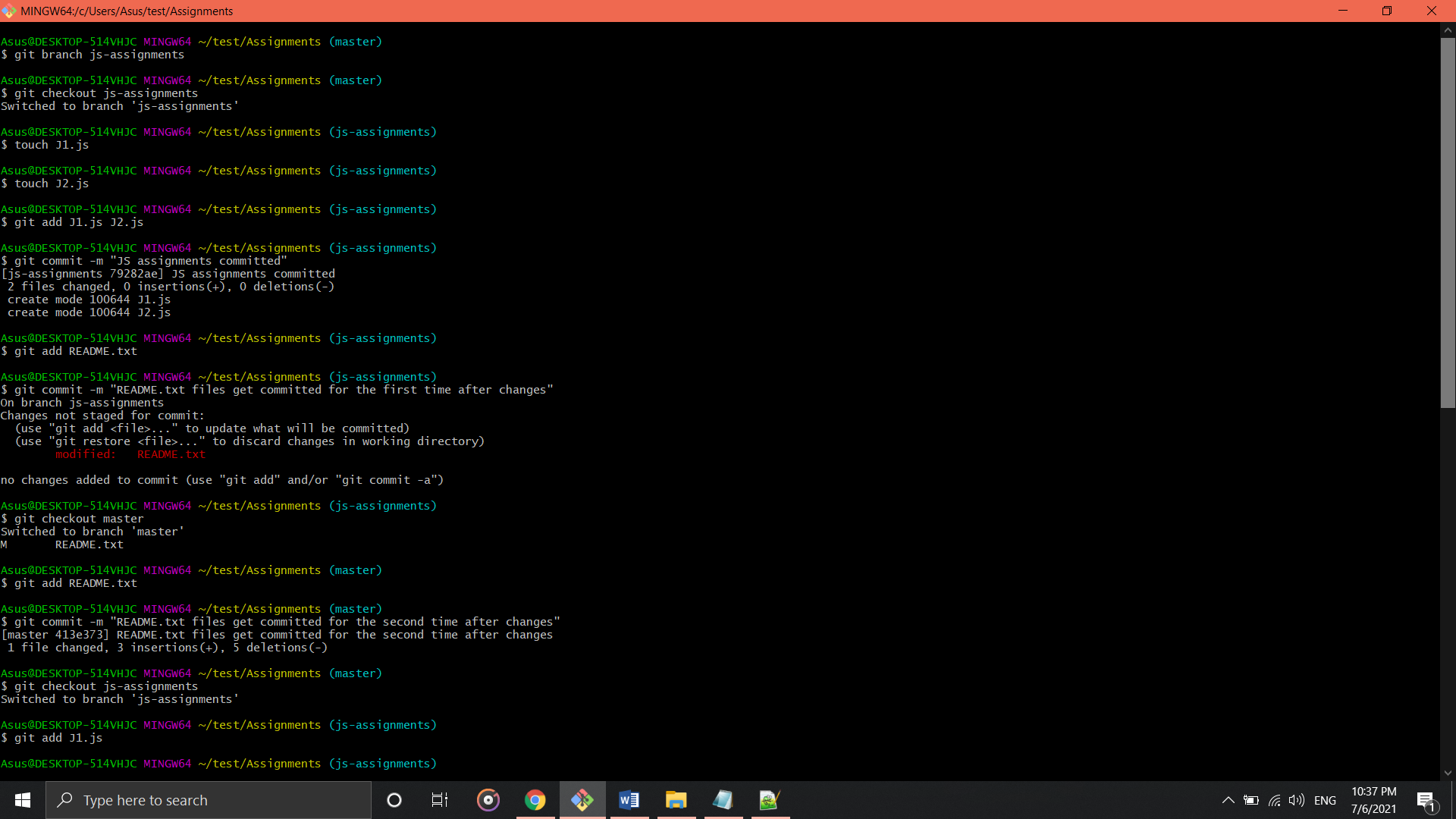
$ git branch -d js-assignments

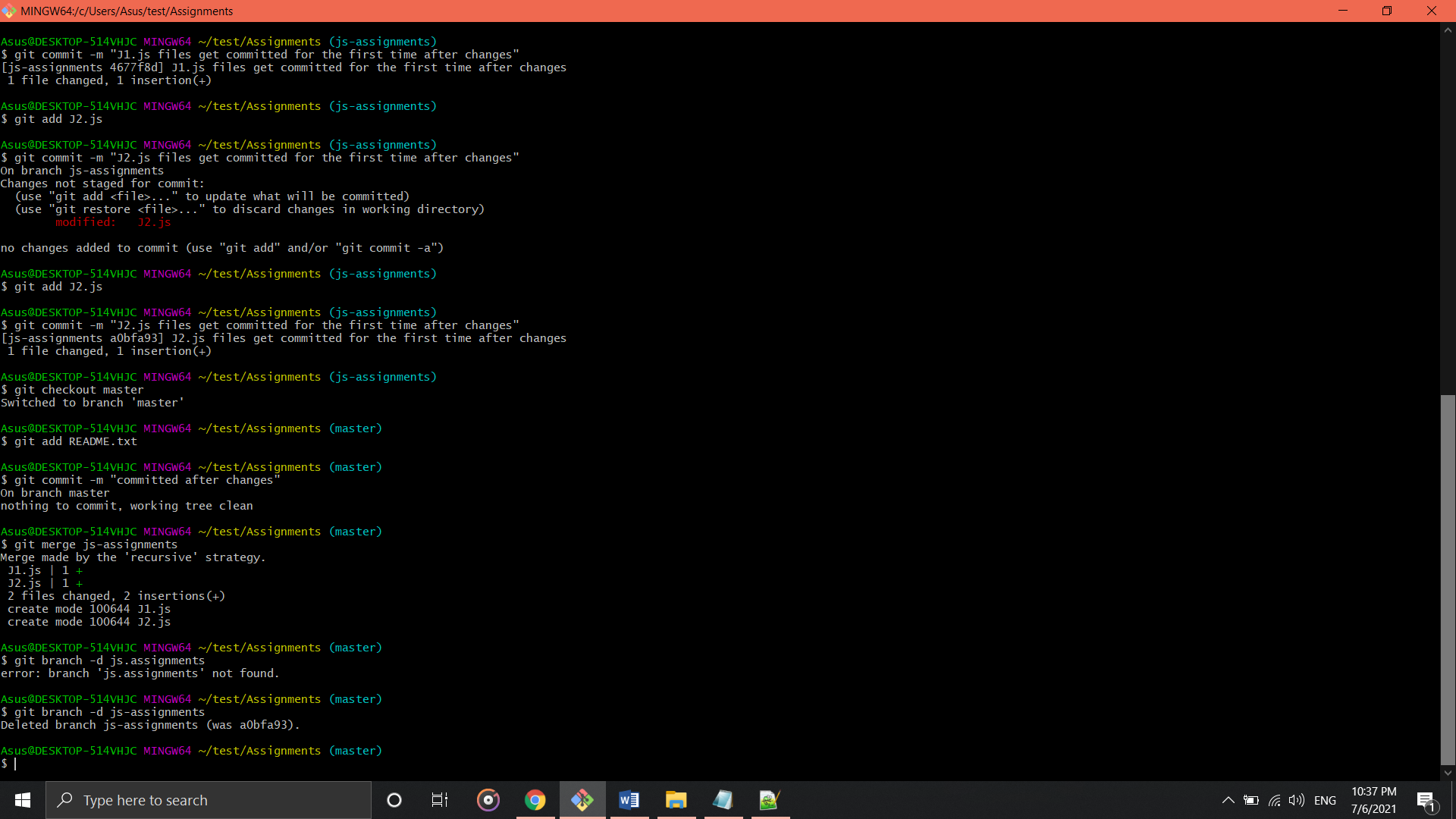
Deleted branch js-assignments (was a0bfa93).

Asus@DESKTOP-514VHJC MINGW64 ~/test/Assignments (master)

$

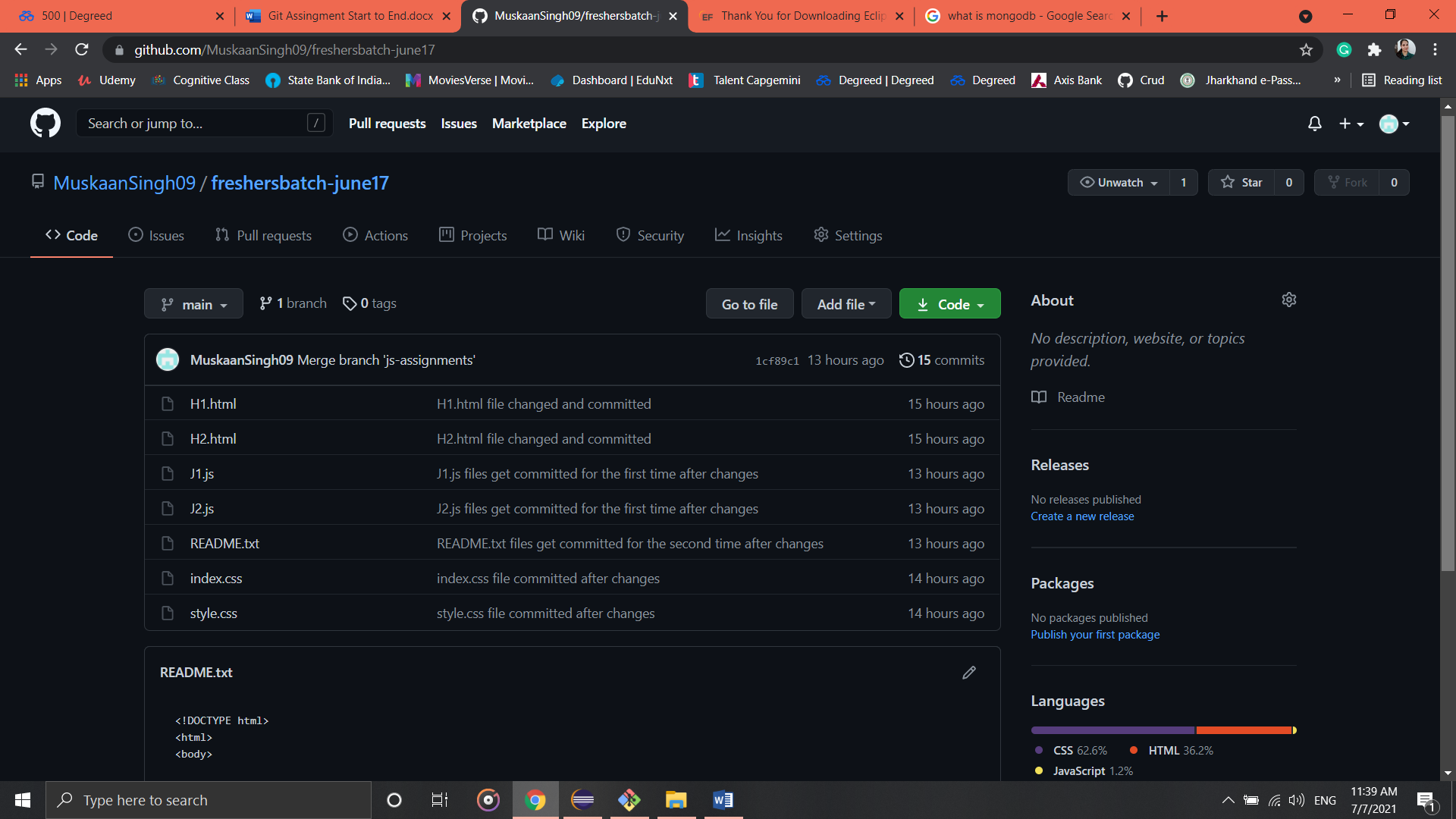
Screenshots





**GIT Remoting**

**Section-3 (Pushing assignments to remote repository)**



**Section-4**

**Pushing source code to remote repository using Eclipse GIT plugin**

**Step 1: Create github account add a repository copy the link of the repo.**

**Step 2: Install Eclipse, open it, open perspective by clicking on perspective icon in the right corner, type GIT, select it and open it.**

**Step 3: click on clone a GIT repository, paste the url of github repo, give username and password of github account, click on next then finish.**

**Step 4: Create new project** (go to file<new<Project<JavaProject<Project

Name<Next<Finish).

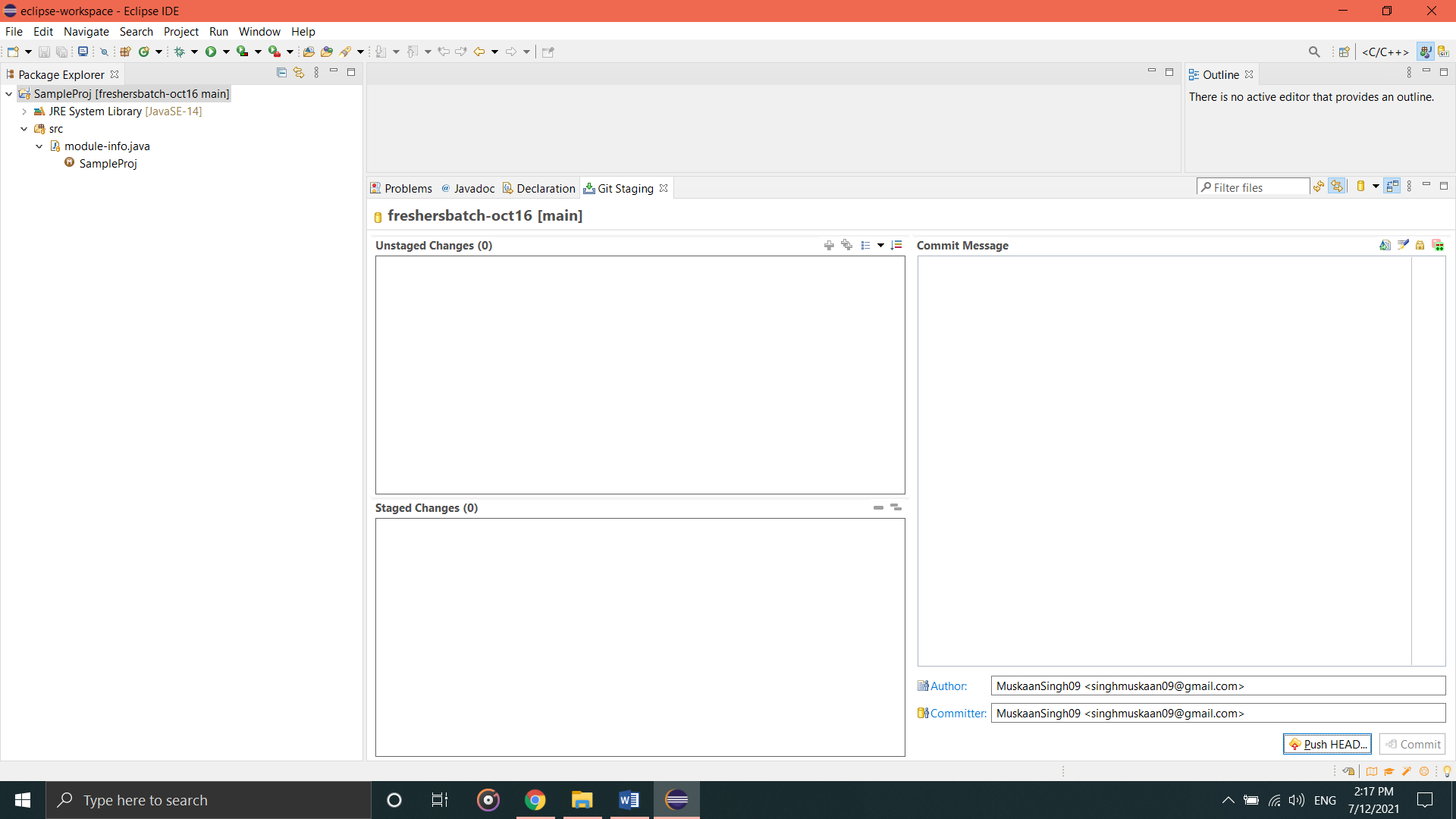
**Step 5: Open perspective, now the project will be shown here, right click on it, then goto team<share project<select the git repo URL<Finish.**

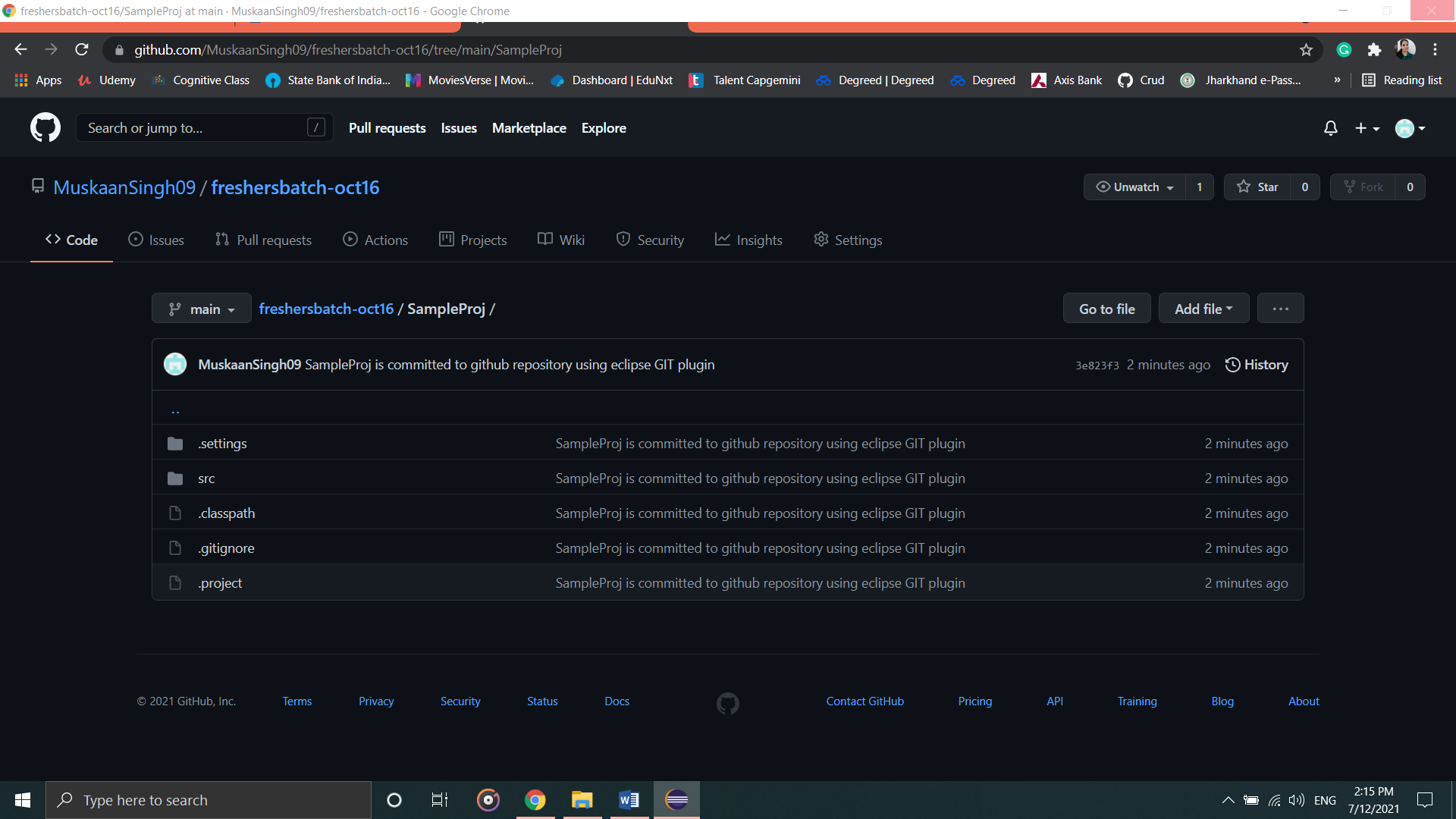
**Step 6: You will be able to see the project icon having git repo name with it, just right click on it<team<commit.**

**Step 6: Drag and drop the files in unstaged changes which you want to push in github account.**

**Step 7: Give a commit message, Click on commit and push, give you username and password of github account.**

**Step 8: go to github account and refresh it, You can see you eclipse project.**

****

****