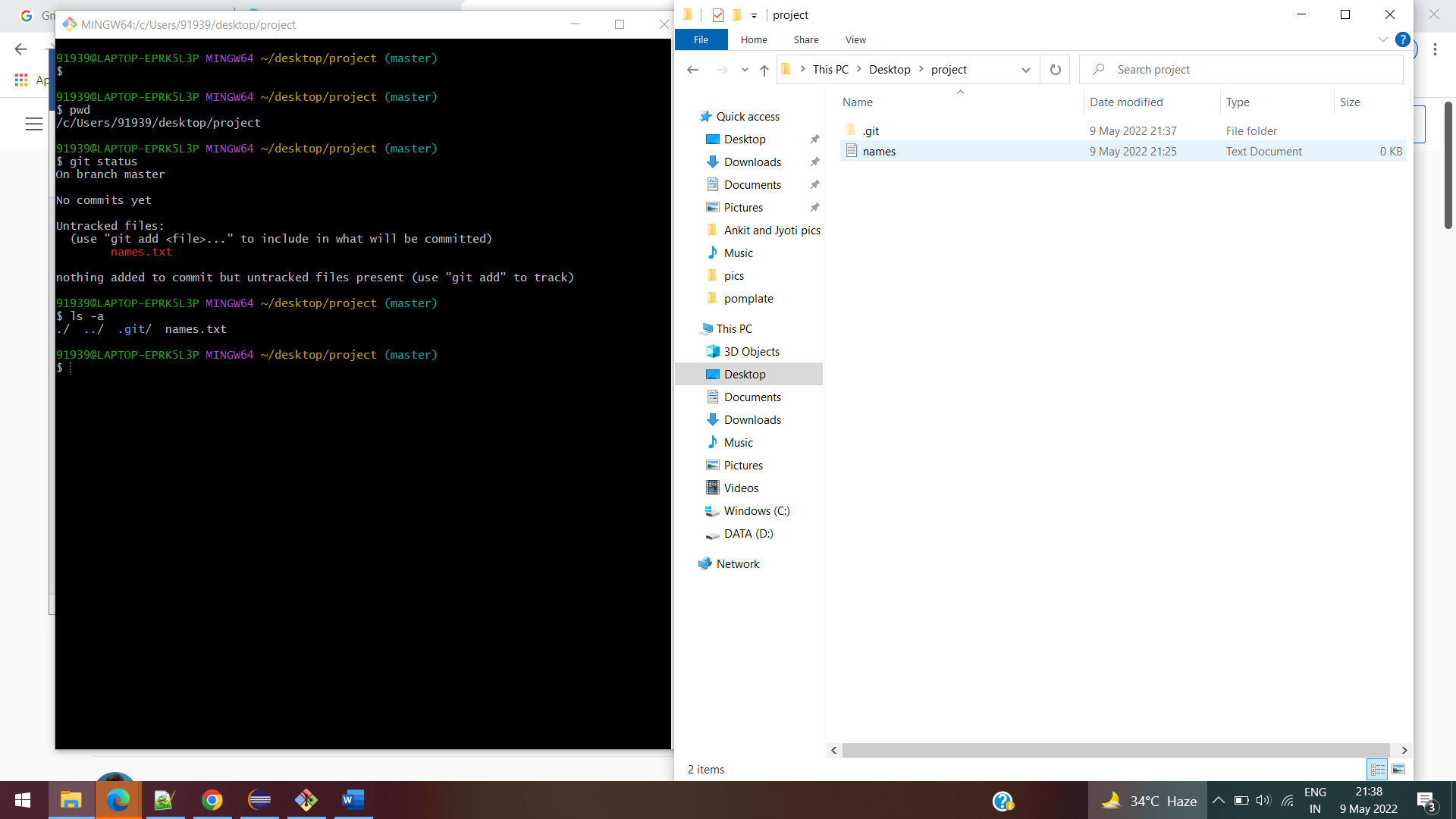
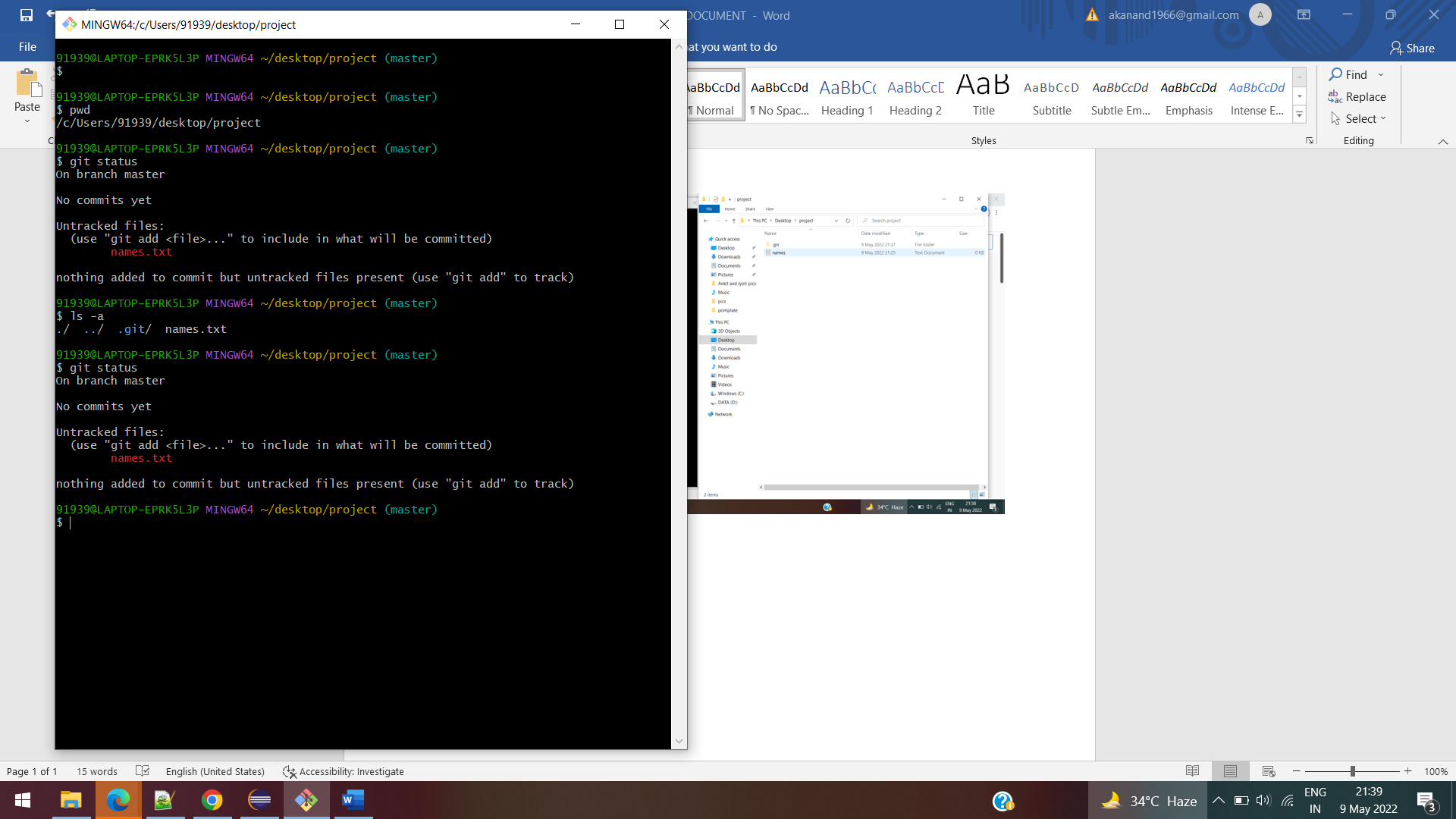
Create a file inside the folder

Project> touch names.txt



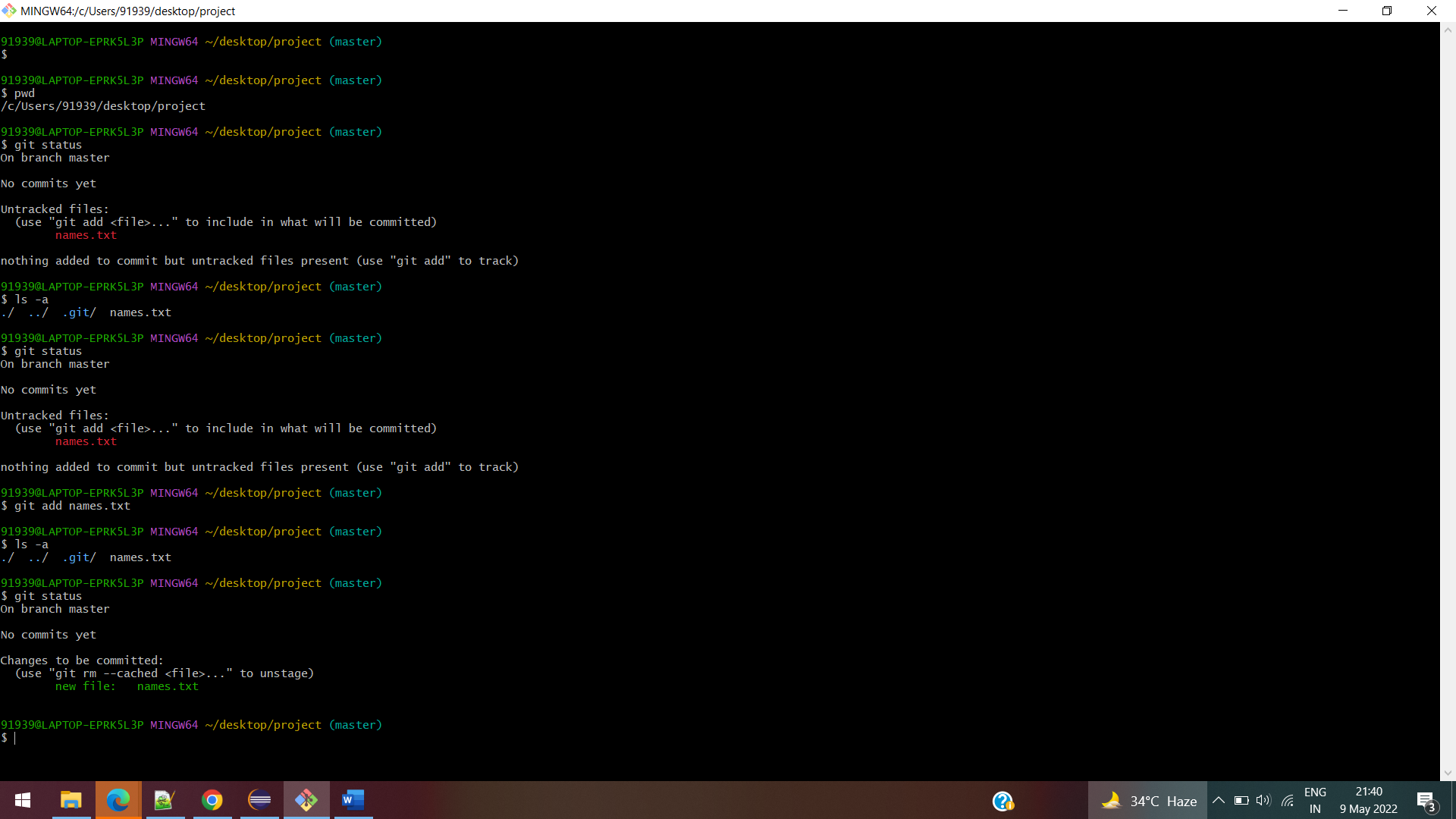
Check the status of the file

$ git status



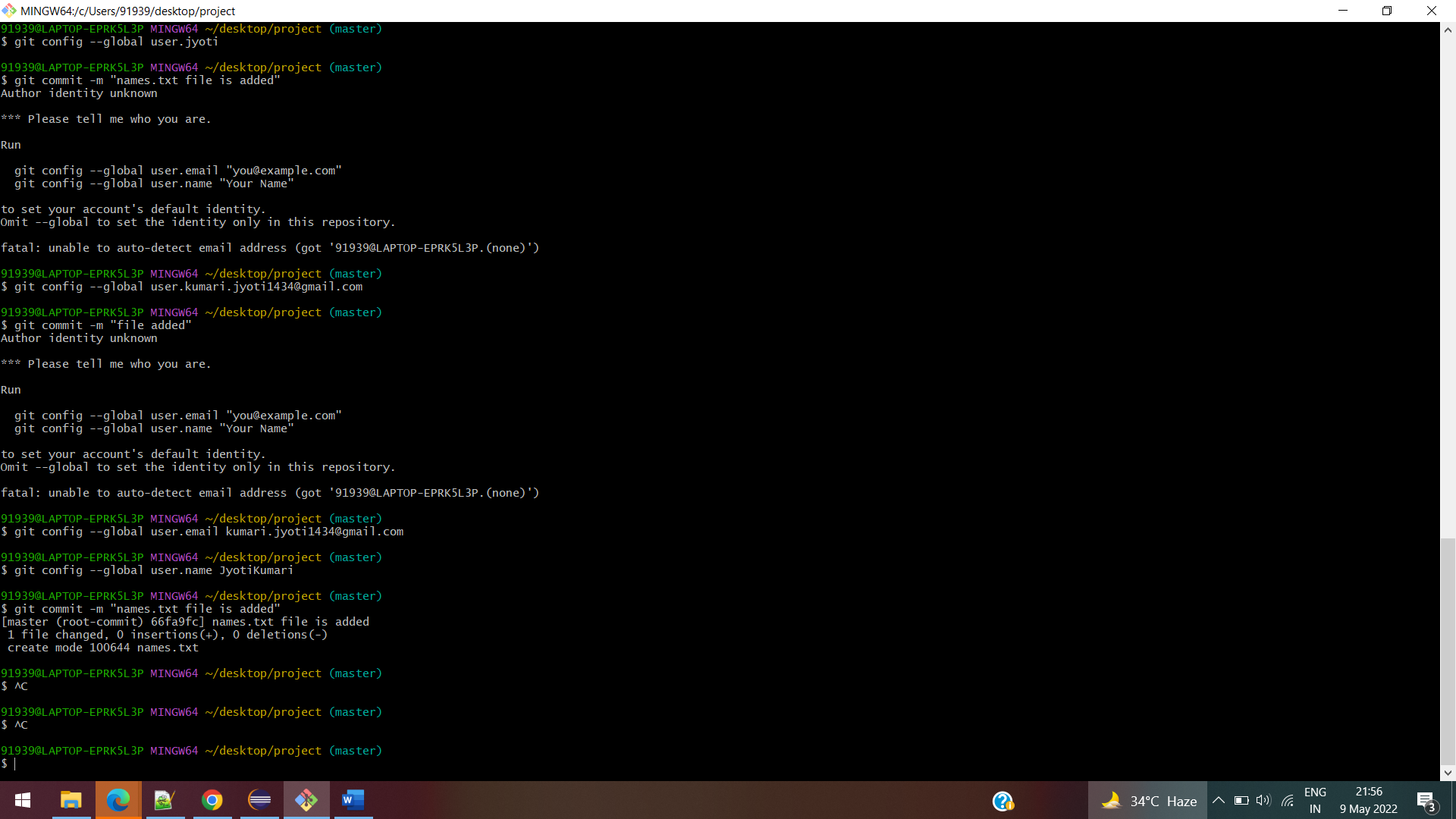
git add names.txt

Status of the file changed and it changed to green color



To config the user name and email address

$ git config --global user.email [kumari.jyoti1434@gmail.com](mailto:kumari.jyoti1434@gmail.com)

$ git config --global user.name JyotiKumari

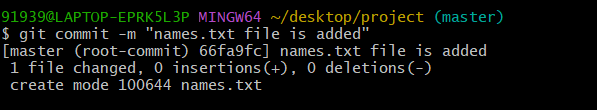
$ git commit -m "names.txt file is added"

[master (root-commit) 66fa9fc] names.txt file is added

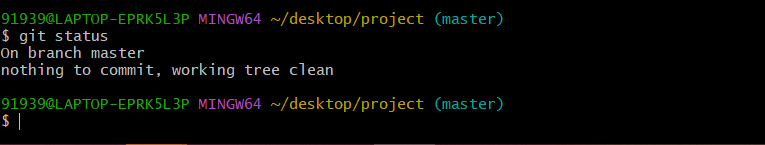
1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 names.txt

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)



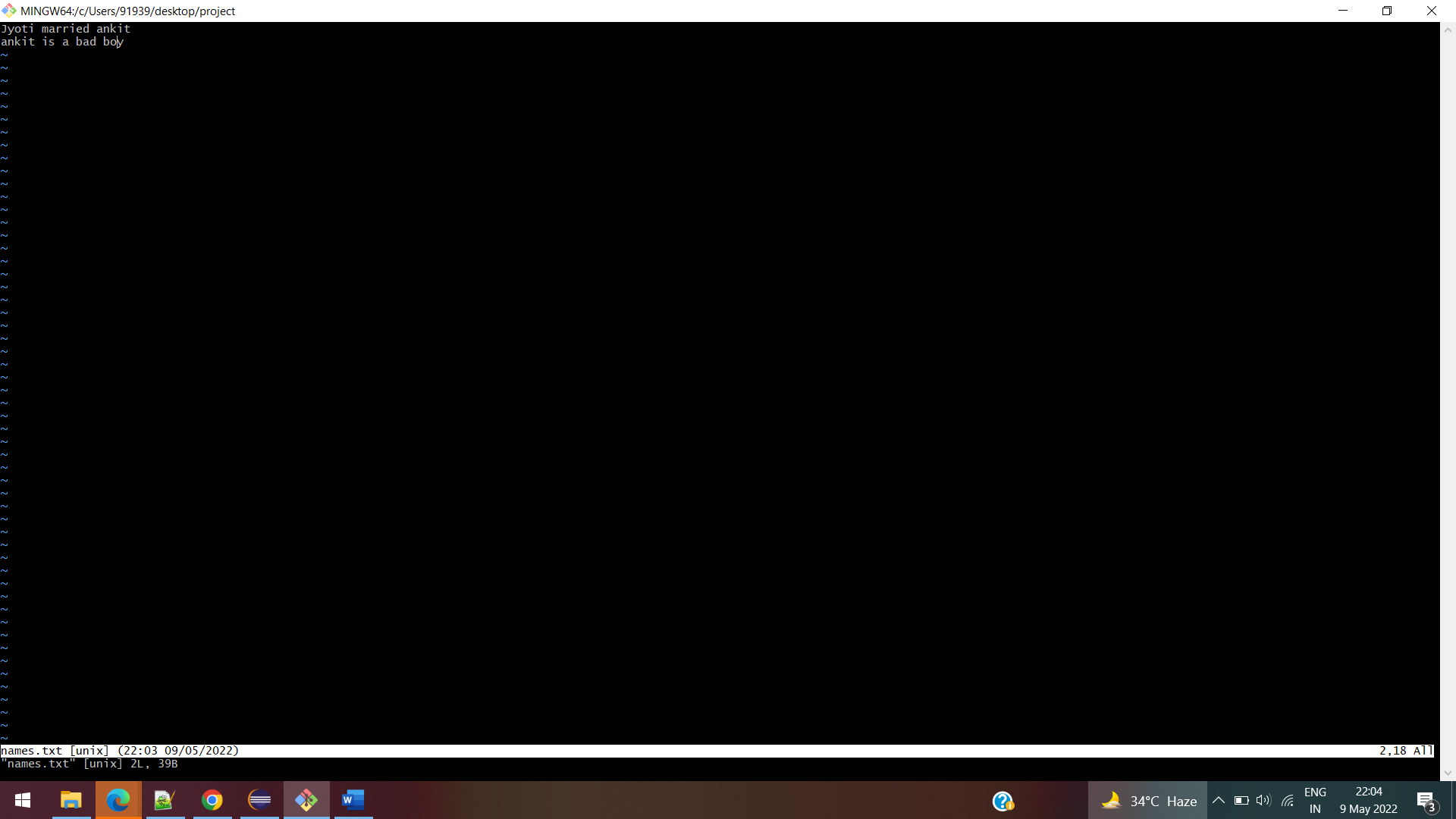
$ git status



To edit the file

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ vi names.txt

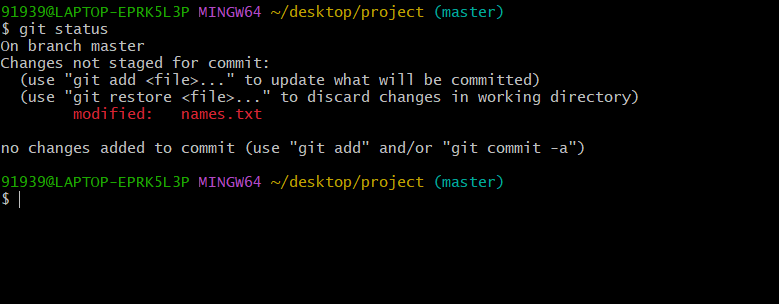


Press I button to insert the data into files

To display the content of the files.

$ cat names.txt

Files has been modified

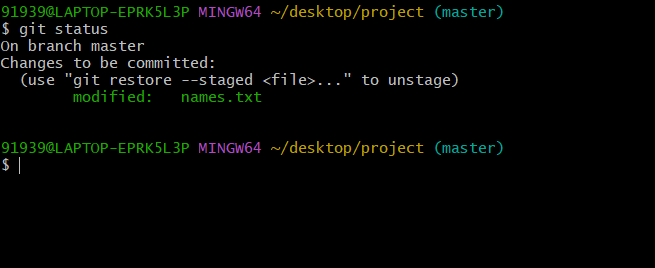


91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git add .

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git status

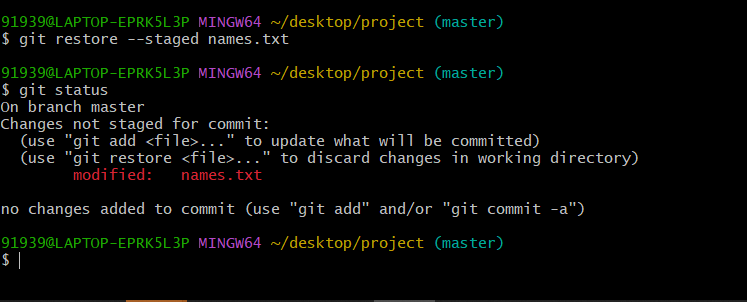


If we want to delete the files which is already been modified

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git restore --staged names.txt

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git status

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

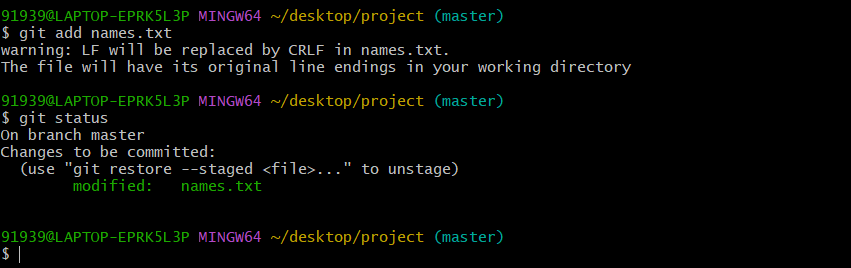
$ git add names.txt

warning: LF will be replaced by CRLF in names.txt.

The file will have its original line endings in your working directory

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

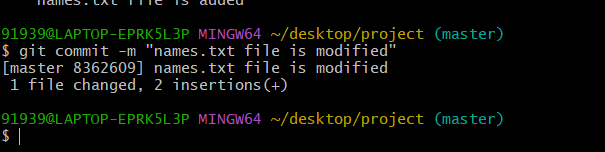
$ git status



To see the history of the files has been modified

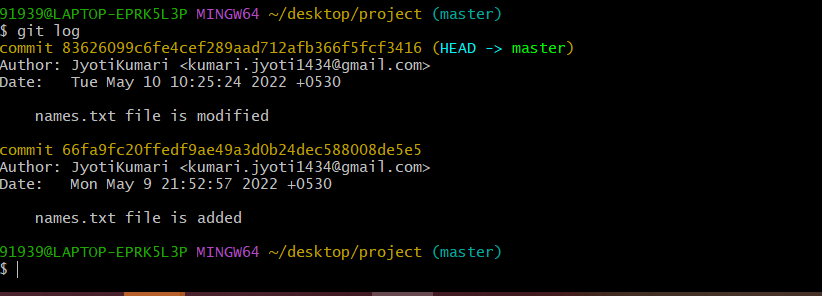
91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git commit -m "names.txt file is modified"



91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

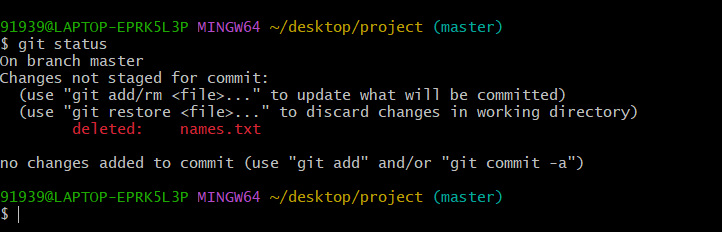
$ git log



Delete the file

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ rm -rf names.txt



Add the file

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git add .

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git status

On branch master

Changes to be committed:

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

deleted: names.txt

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git commit -m "names.txt file is deleted"

[master 4b1ebb7] names.txt file is deleted

1 file changed, 2 deletions(-)

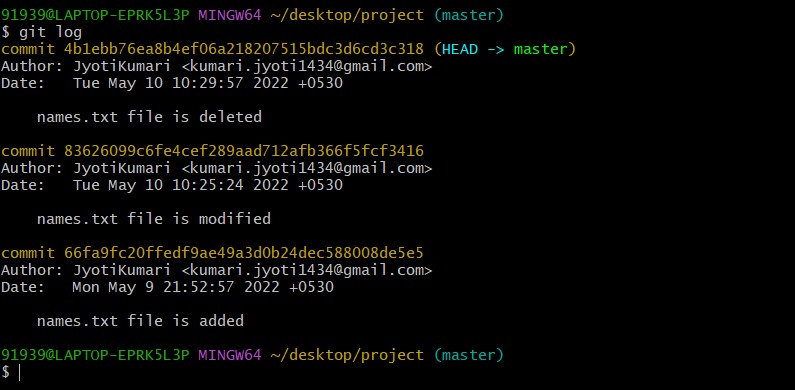
delete mode 100644 names.txt

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git log



If we want to remove the modified and delete history from log and we want the original file which it was before we modified it. likely the first version of the file.

We can use the reset command to remove those history from the file.

We need to hash id( commit) to remove the history.

Suppose we have 3 commit ids

1….delete commit

2---modified commit

3---Add commit

And we want to delete modified and delete history we have to use the below commit id to remove the above commits

We have to use 3 commit id, above this commit everything will get removed

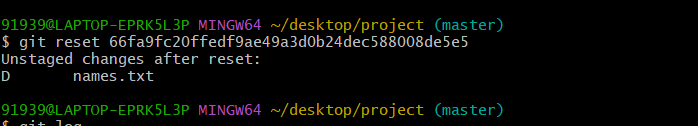
91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git reset 66fa9fc20ffedf9ae49a3d0b24dec588008de5e5

Unstaged changes after reset:

D names.txt

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)



Now the check the history file will be removed

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git log

commit 66fa9fc20ffedf9ae49a3d0b24dec588008de5e5 (HEAD -> master)

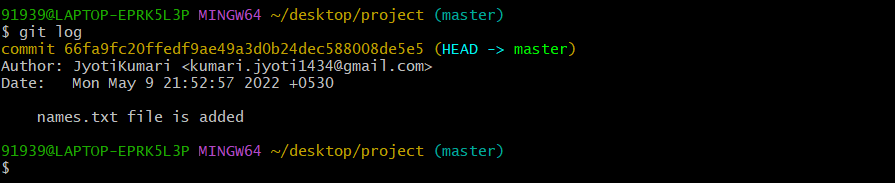
Author: JyotiKumari <kumari.jyoti1434@gmail.com>

Date: Mon May 9 21:52:57 2022 +0530

names.txt file is added

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$



91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git status

On branch master

Changes not staged for commit:

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

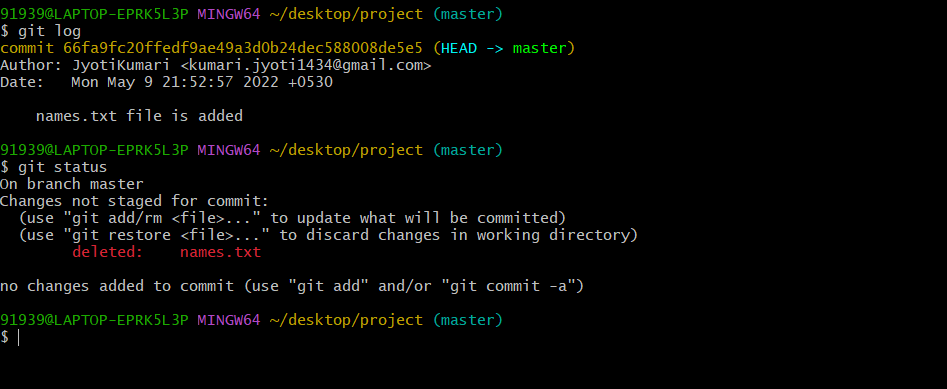
(use "git restore <file>..." to discard changes in working directory)

deleted: names.txt

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

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$



If we want the deleted files back we have the command called restore

If we want some files our saved work without any history we can keep those file in stash and later we can use that file .

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git add .

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git status

On branch master

Changes to be committed:

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

deleted: names.txt

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

Create a different file

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ touch surname.txt

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git status

On branch master

Changes to be committed:

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

deleted: names.txt

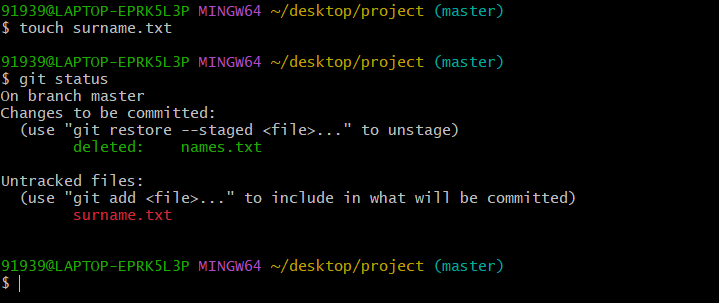
Untracked files:

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

surname.txt

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$



Add file

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git add .

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git status

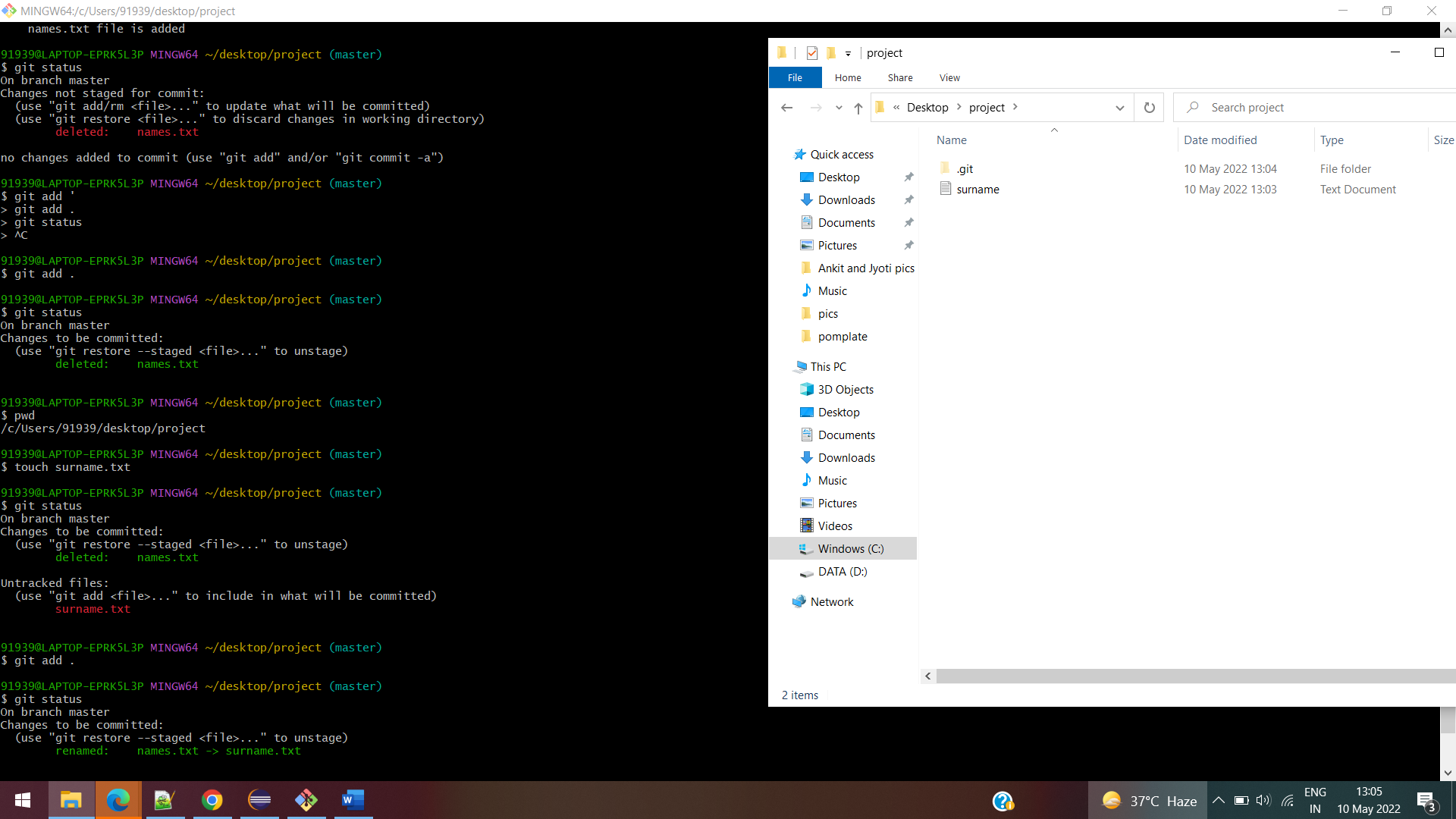
On branch master

Changes to be committed:

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

renamed: names.txt -> surname.txt

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)



91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ vi surname.txt

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ cat surname.txt

jyoti kumari

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

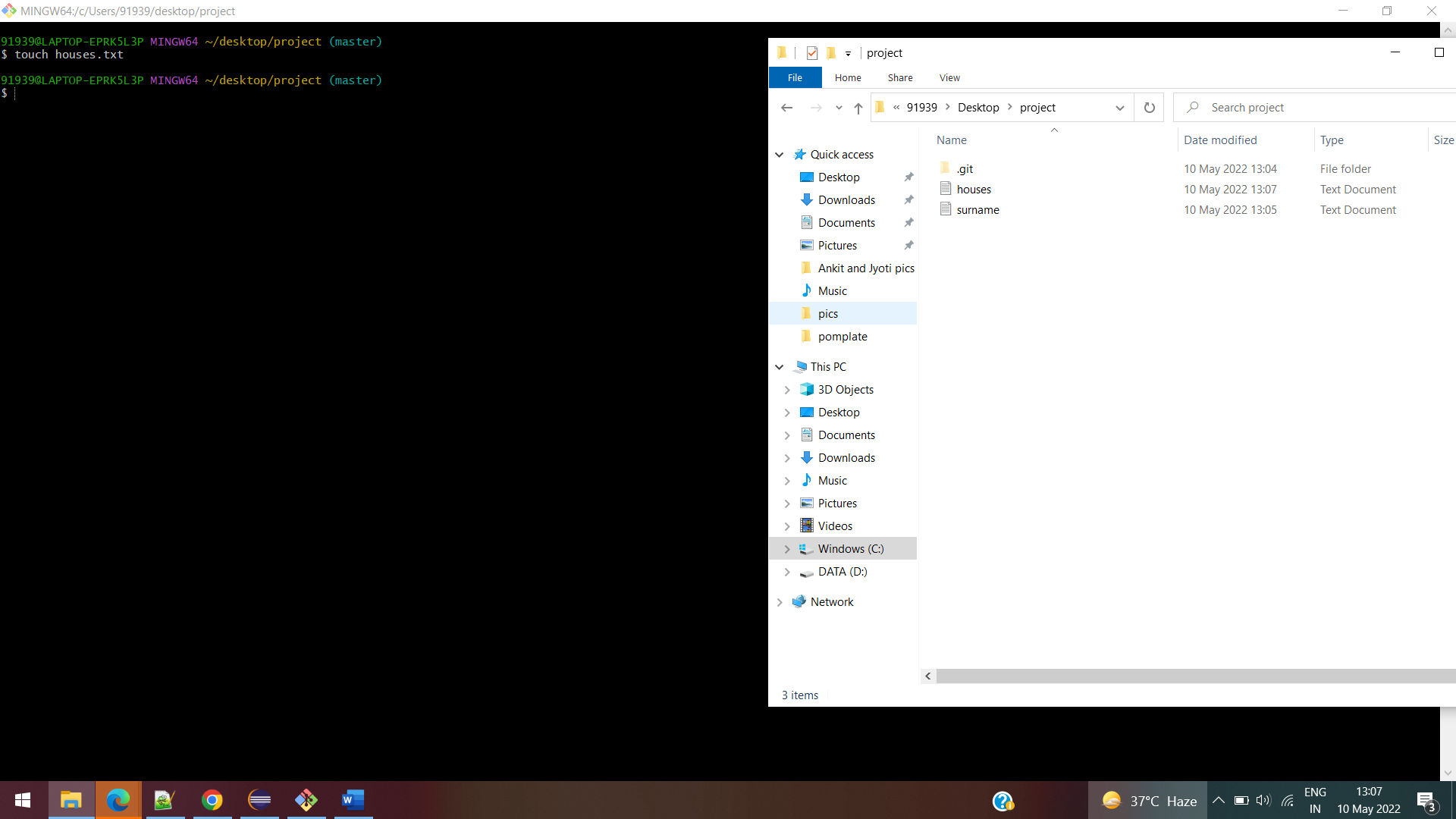
Create another file

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ touch houses.txt

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$



91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git status

On branch master

Changes to be committed:

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

renamed: names.txt -> surname.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: surname.txt

Untracked files:

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

houses.txt

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git add .

warning: LF will be replaced by CRLF in surname.txt.

The file will have its original line endings in your working directory

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git status

On branch master

Changes to be committed:

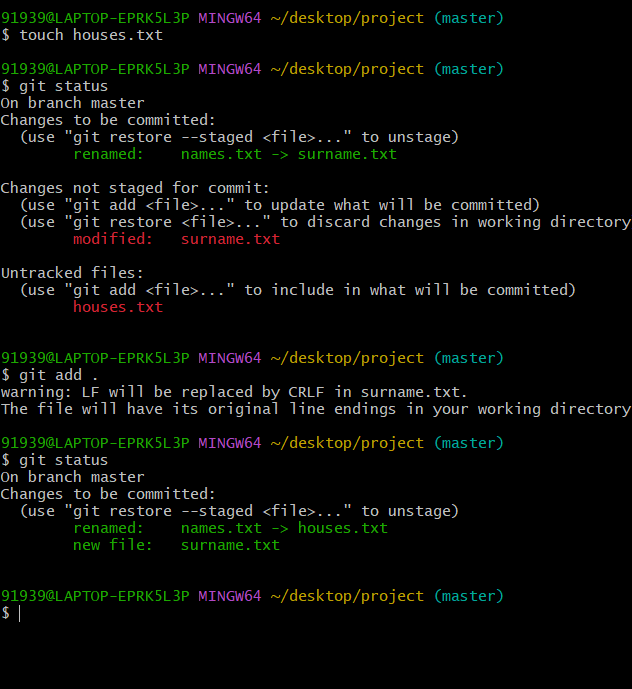
(use "git restore --staged <file>..." to unstage)

renamed: names.txt -> houses.txt

new file: surname.txt

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$



Now we want to put the above files in stash without commit and whenever we need we can bring that back from stash

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git stash

Saved working directory and index state WIP on master: 66fa9fc names.txt file is added

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

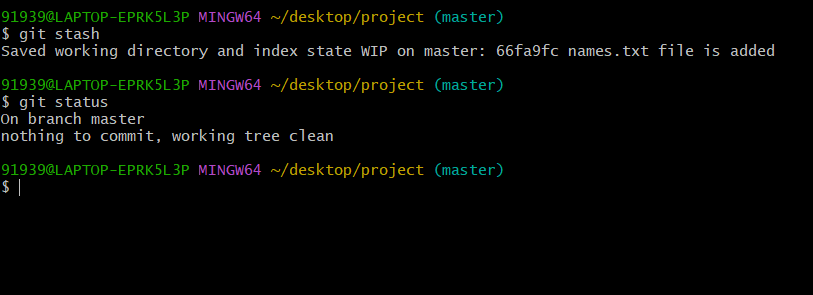
$ git status

On branch master

nothing to commit, working tree clean

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$



91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git log

commit 66fa9fc20ffedf9ae49a3d0b24dec588008de5e5 (HEAD -> master)

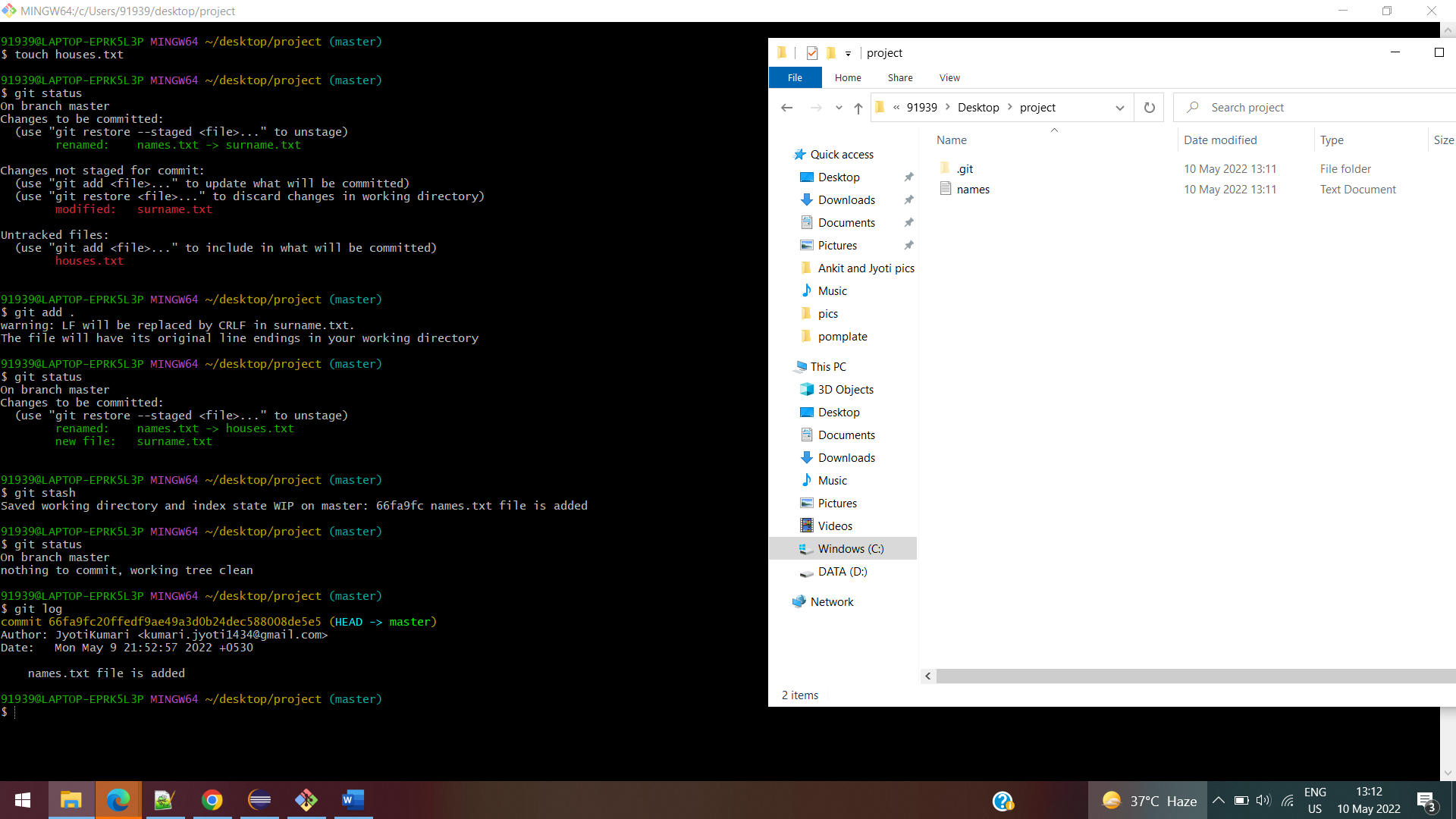
Author: JyotiKumari <kumari.jyoti1434@gmail.com>

Date: Mon May 9 21:52:57 2022 +0530

names.txt file is added

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$



To get the files back from stash

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git stash pop

Removing names.txt

On branch master

Changes to be committed:

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

new file: houses.txt

new file: surname.txt

Changes not staged for commit:

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

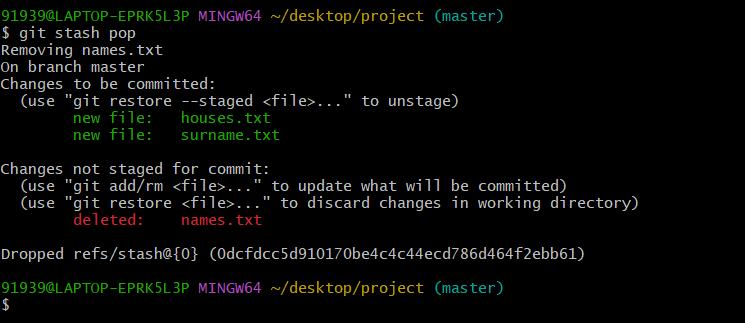
(use "git restore <file>..." to discard changes in working directory)

deleted: names.txt

Dropped refs/stash@{0} (0dcfdcc5d910170be4c4c44ecd786d464f2ebb61)

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$



And if we want to remove the files from stash we don’t want that anymore we have to use the below command and process, files which were not committed

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git status

On branch master

Changes to be committed:

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

new file: houses.txt

new file: surname.txt

Changes not staged for commit:

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

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

deleted: names.txt

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git add .

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git stash

Saved working directory and index state WIP on master: 66fa9fc names.txt file is added

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git stash clear

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git status

On branch master

nothing to commit, working tree clean

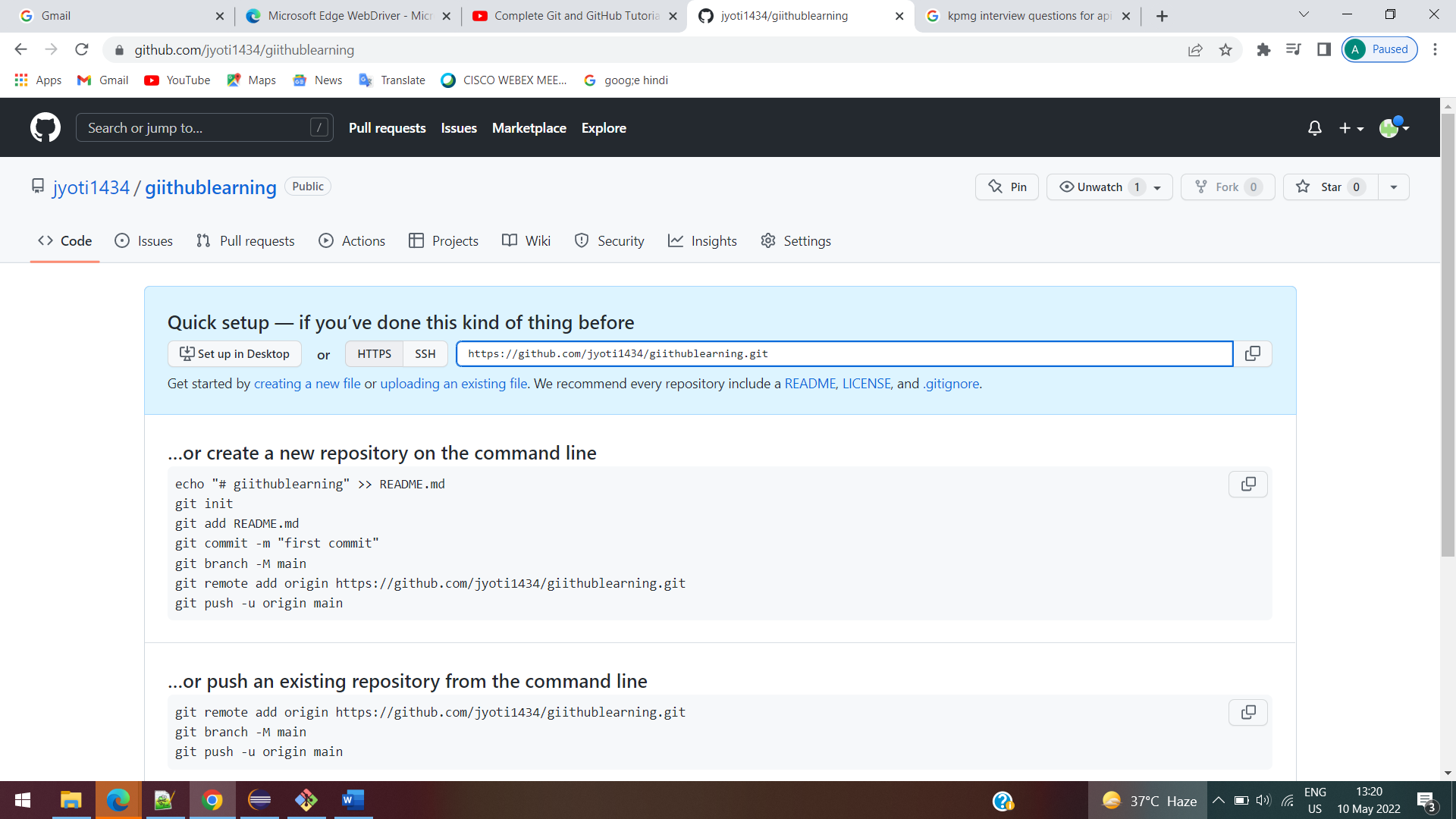
91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$

GIT HUB

Open github.com and click on profile and click on new to create the repository

Give the repository name and create it

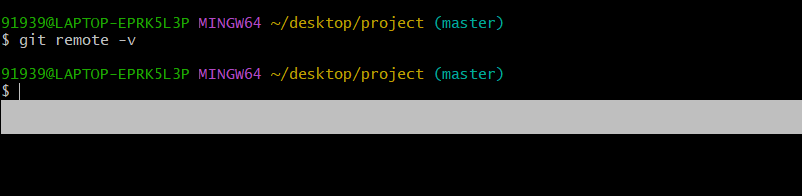


To associate the url (repository) to the project

Now the no url is associated

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git remote -v



Connecting remote repository to local repository

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

$ git remote add origin https://github.com/jyoti1434/giithublearning.git

To view url associated or not

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

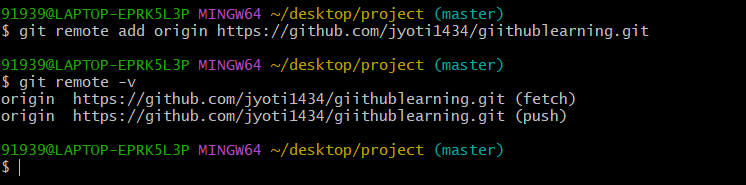
$ git remote -v

origin https://github.com/jyoti1434/giithublearning.git (fetch)

origin https://github.com/jyoti1434/giithublearning.git (push)

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (master)

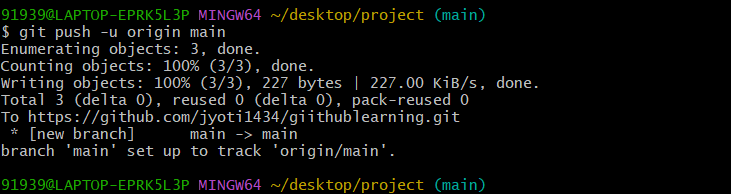
$



To push the file in repository

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (main)

$ git push -u origin main



We don’t have to keep the working files which the code is not completed yet we can create the other branch and keep our files in that branch

$ git log

commit 3856571b564d1099361cbc168589bb4deb266eb9 (HEAD -> main)

Author: JyotiKumari <kumari.jyoti1434@gmail.com>

Date: Tue May 10 15:37:08 2022 +0530

txt added

commit 39c8077be3e4863b5fcae39c487455ec7ba638d9

Author: JyotiKumari <kumari.jyoti1434@gmail.com>

Date: Tue May 10 15:36:02 2022 +0530

hotel.txt is added

commit 66fa9fc20ffedf9ae49a3d0b24dec588008de5e5 (origin/main)

Author: JyotiKumari <kumari.jyoti1434@gmail.com>

Date: Mon May 9 21:52:57 2022 +0530

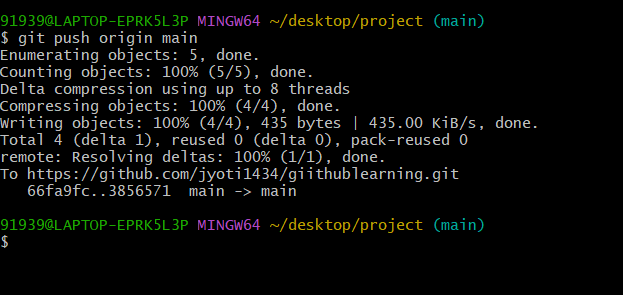
names.txt file is added

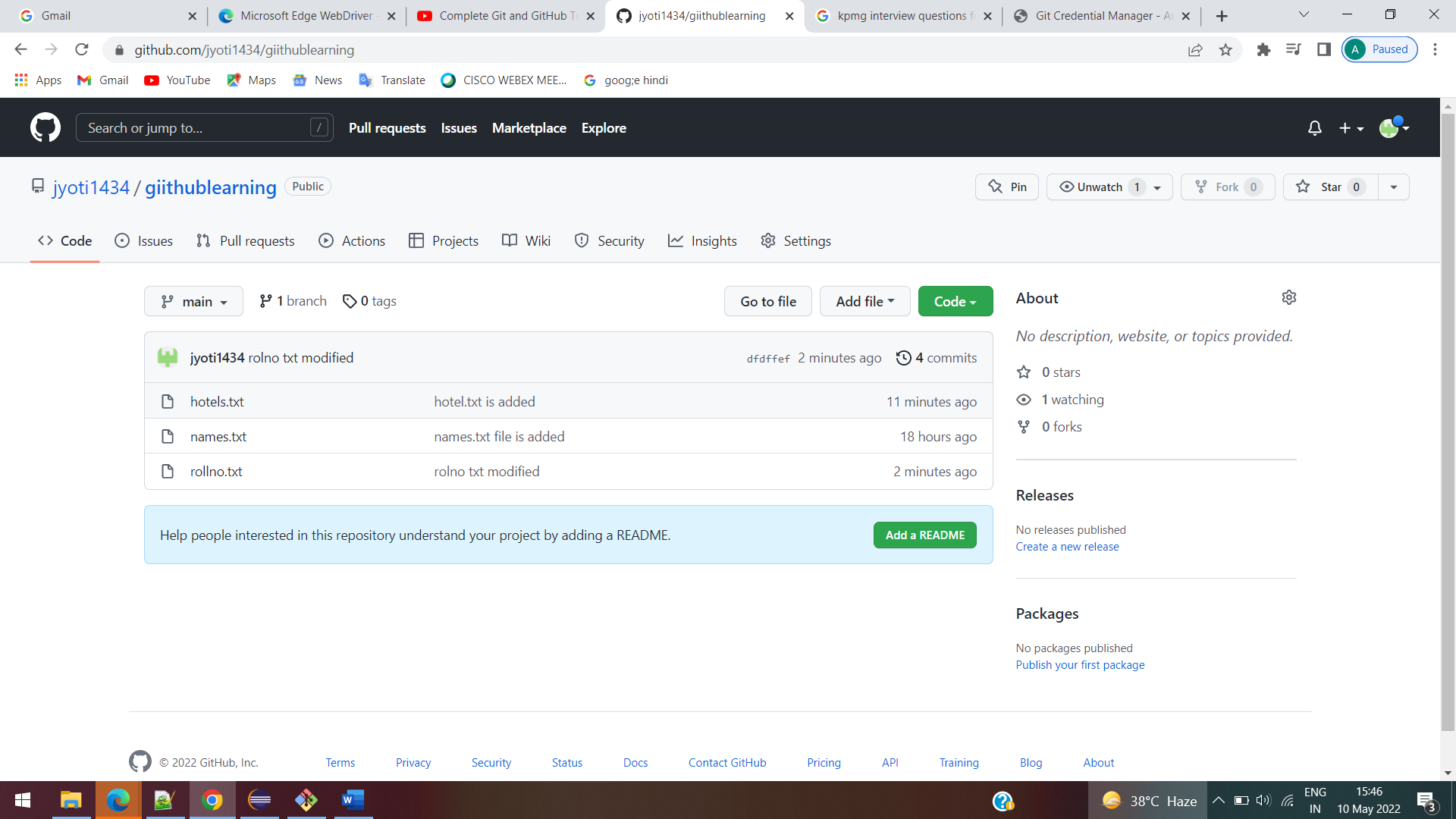
91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (main)

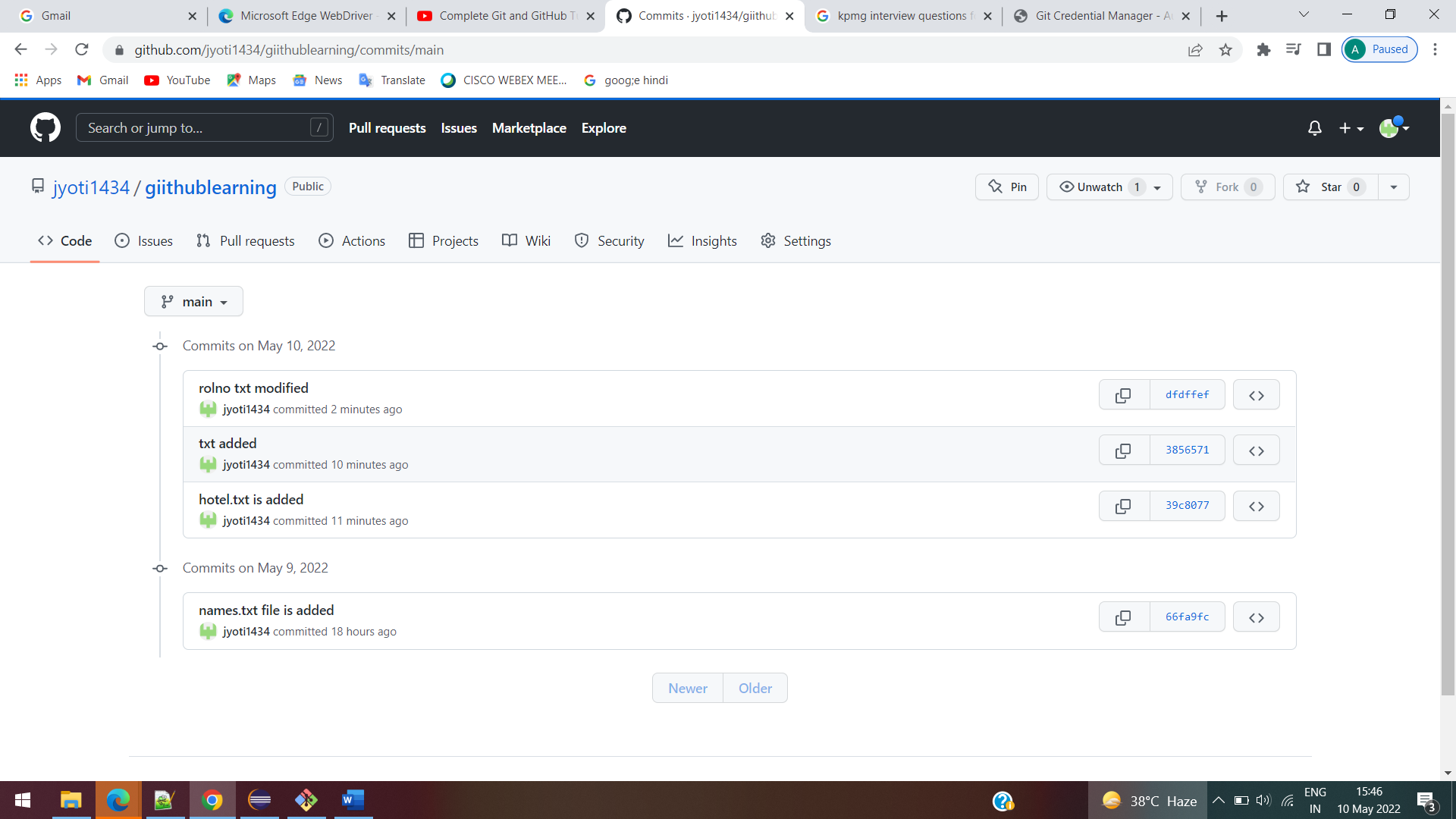
**HEAD- points the branch we are adding the features or code**

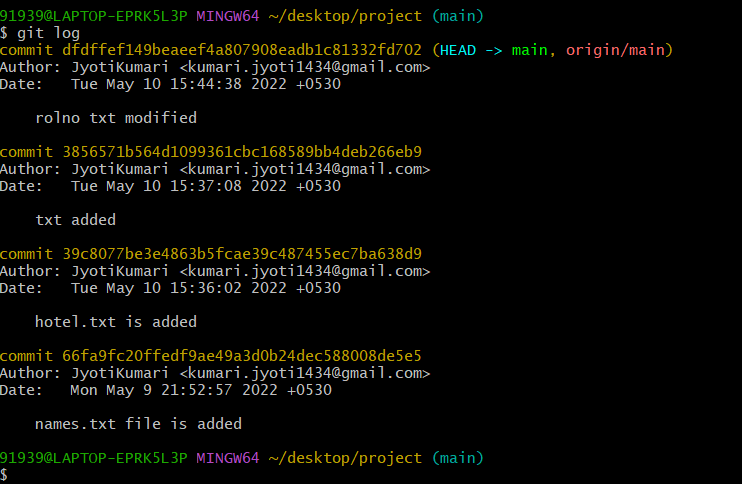
Now new commit we want to push to the min branch

$ git push origin main





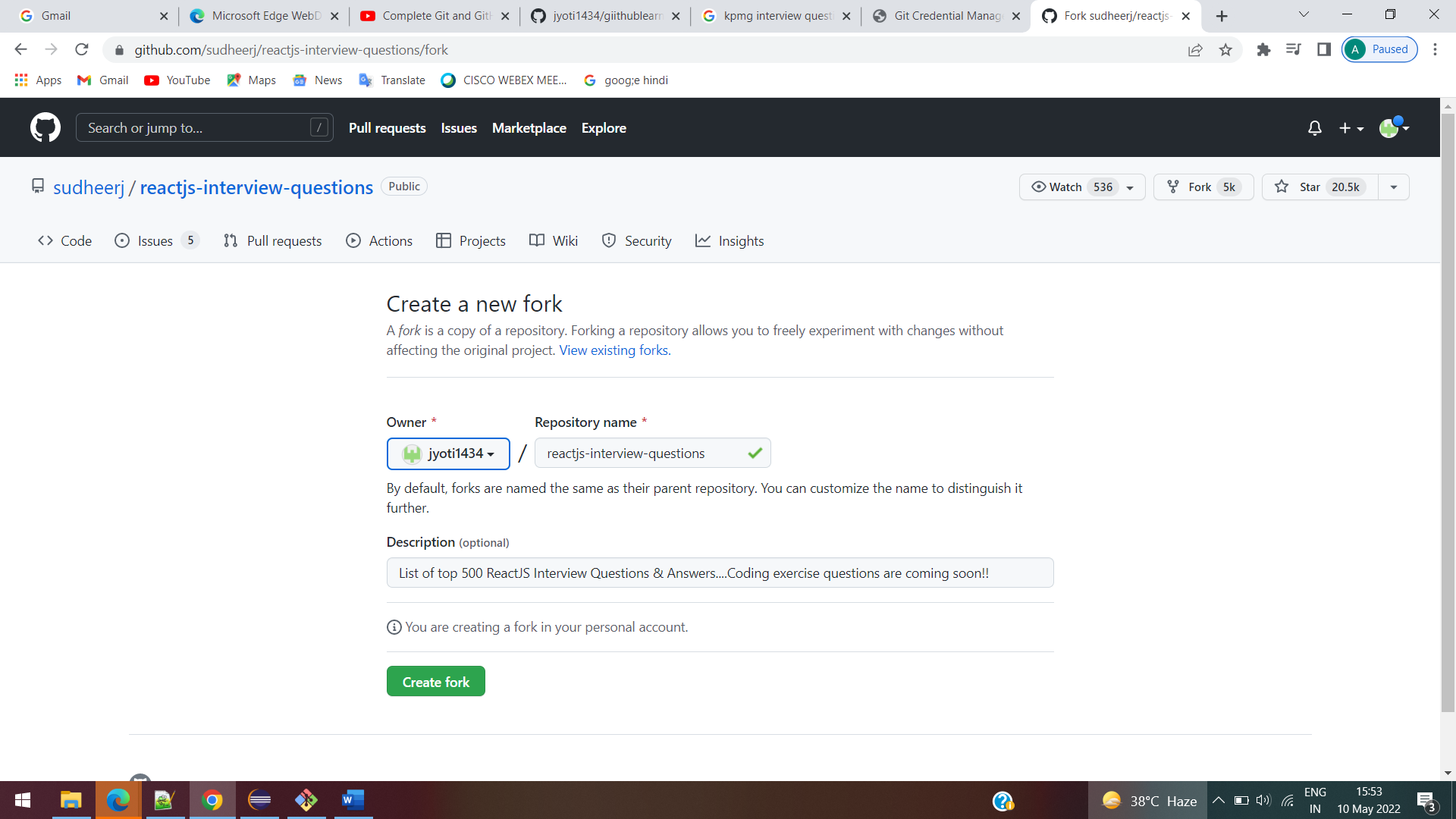


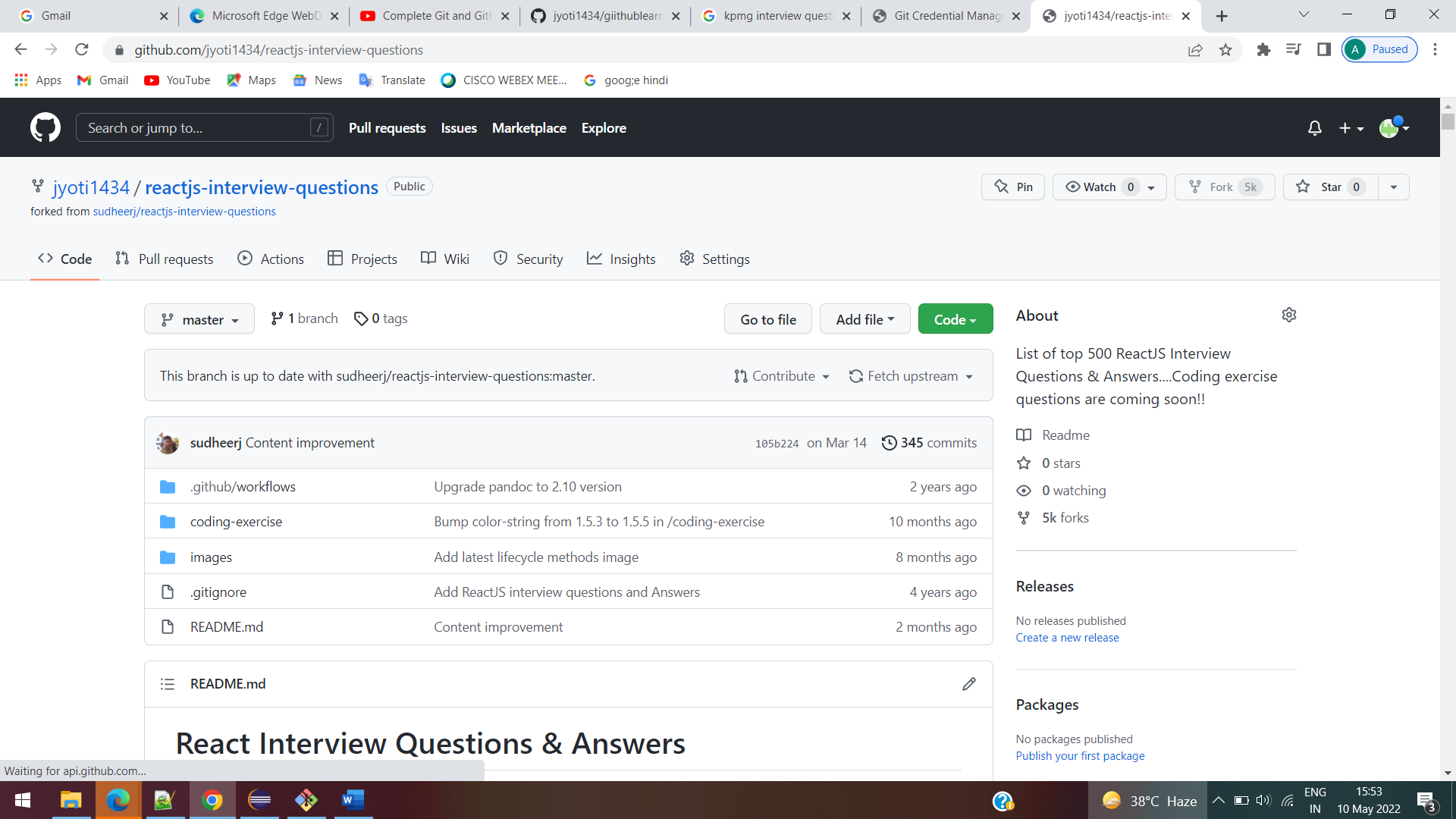


**To work on the existing projects which we don’t have access**

**Create a copy of that project in your own account**

**Click on fork option**

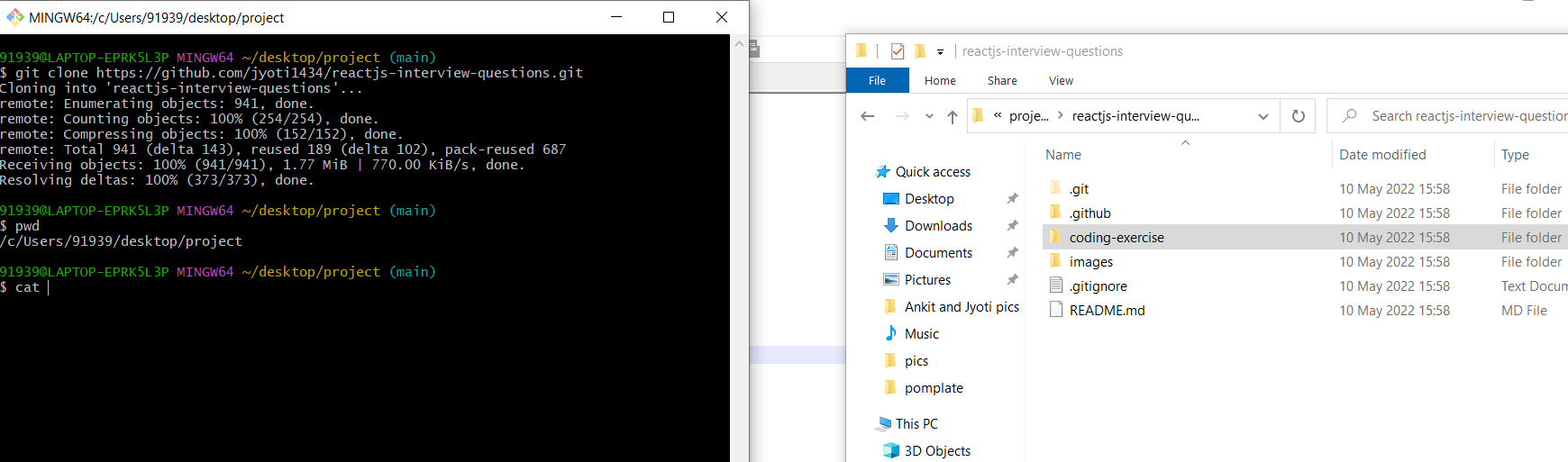




Clone the code to the system

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project (main)

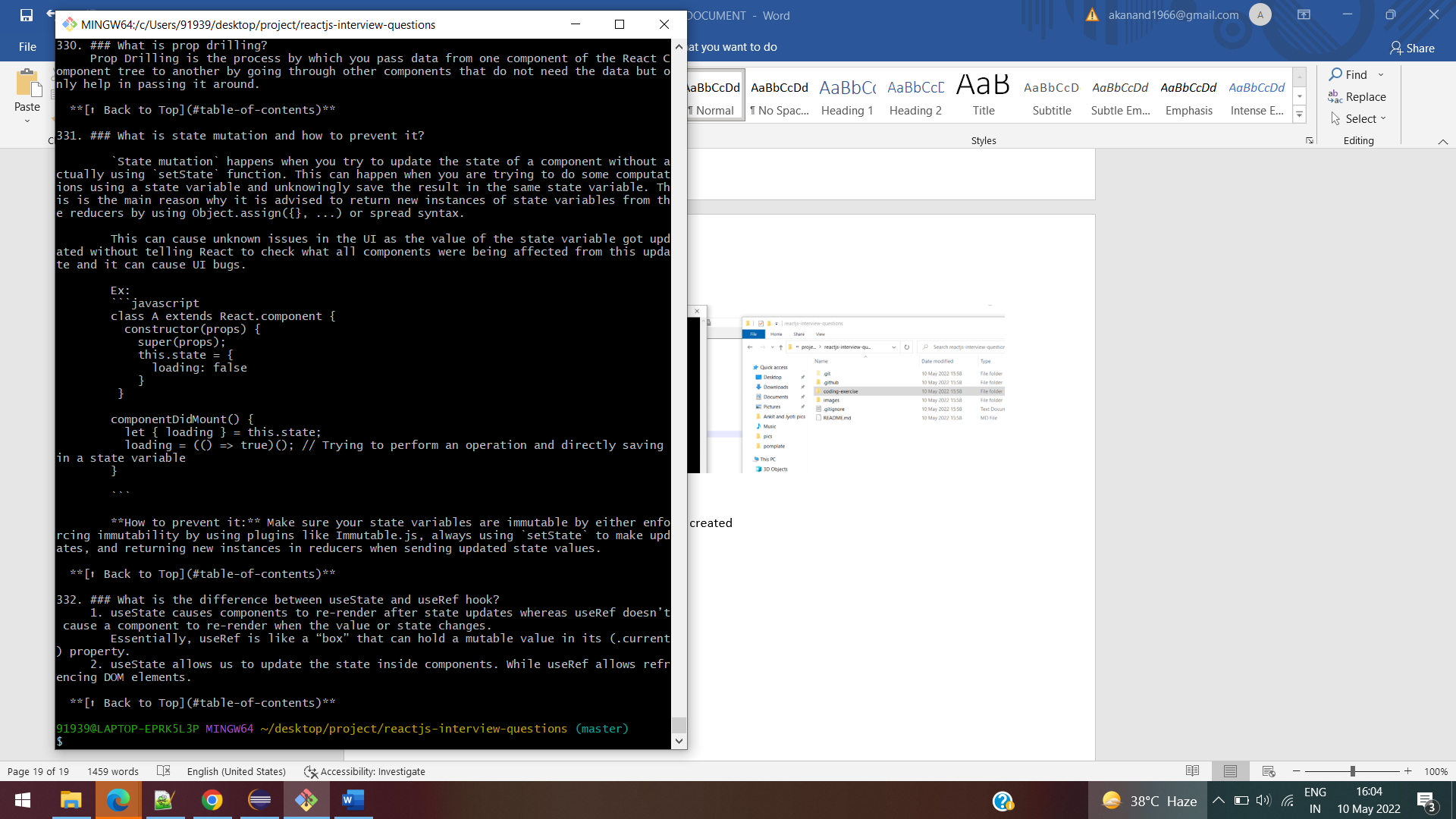
$ git clone https://github.com/jyoti1434/reactjs-interview-questions.git



Change the directory and move to the folder created

Cd folder name

Now see the details of the readme file



Add the url of the original branch from where we forked or copied to our branch

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project/reactjs-interview-questions (master)

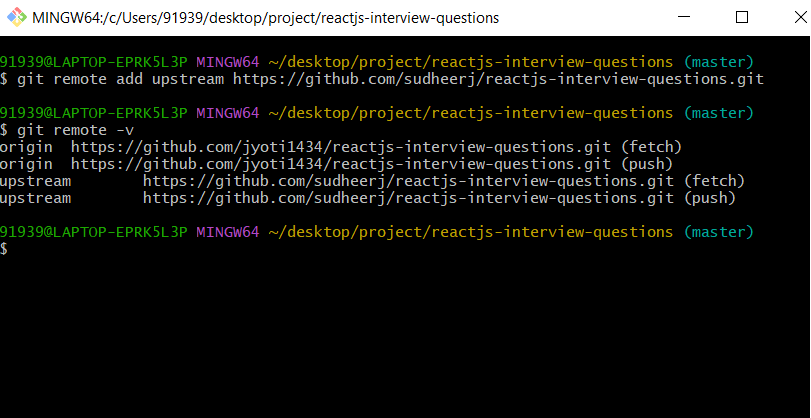
$ git remote add upstream https://github.com/sudheerj/reactjs-interview-questions.git

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project/reactjs-interview-questions (master)

$

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project/reactjs-interview-questions (master)

$ git remote -v



**Create a branch, we don’t want to add it to the main branch**

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project/reactjs-interview-questions (master)

$ git branch jyoti

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project/reactjs-interview-questions (master)

$

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project/reactjs-interview-questions (master)

$ git checkout jyoti

Switched to branch 'jyoti'

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project/reactjs-interview-questions (jyoti)

$

**Add file to the branch**

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project/reactjs-interview-questions (jyoti)

$ git add .

warning: LF will be replaced by CRLF in README.TXT.

The file will have its original line endings in your working directory

91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project/reactjs-interview-questions (jyoti)

$ git commit -m "jyoti added a readme file"

[jyoti 3a94917] jyoti added a readme file

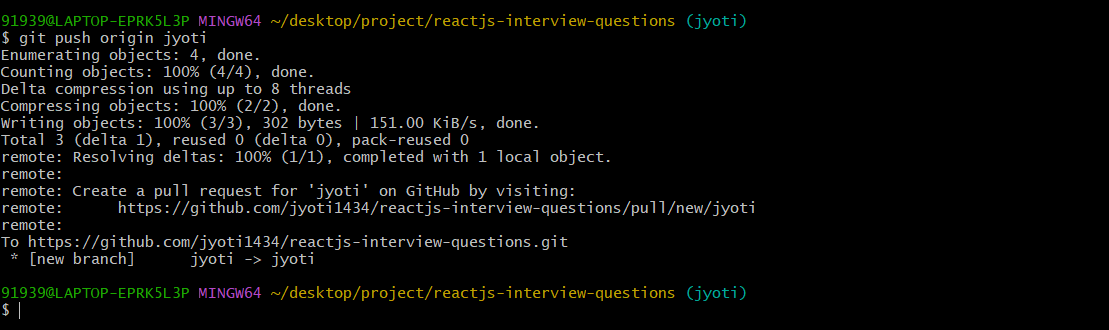
1 file changed, 1 insertion(+)

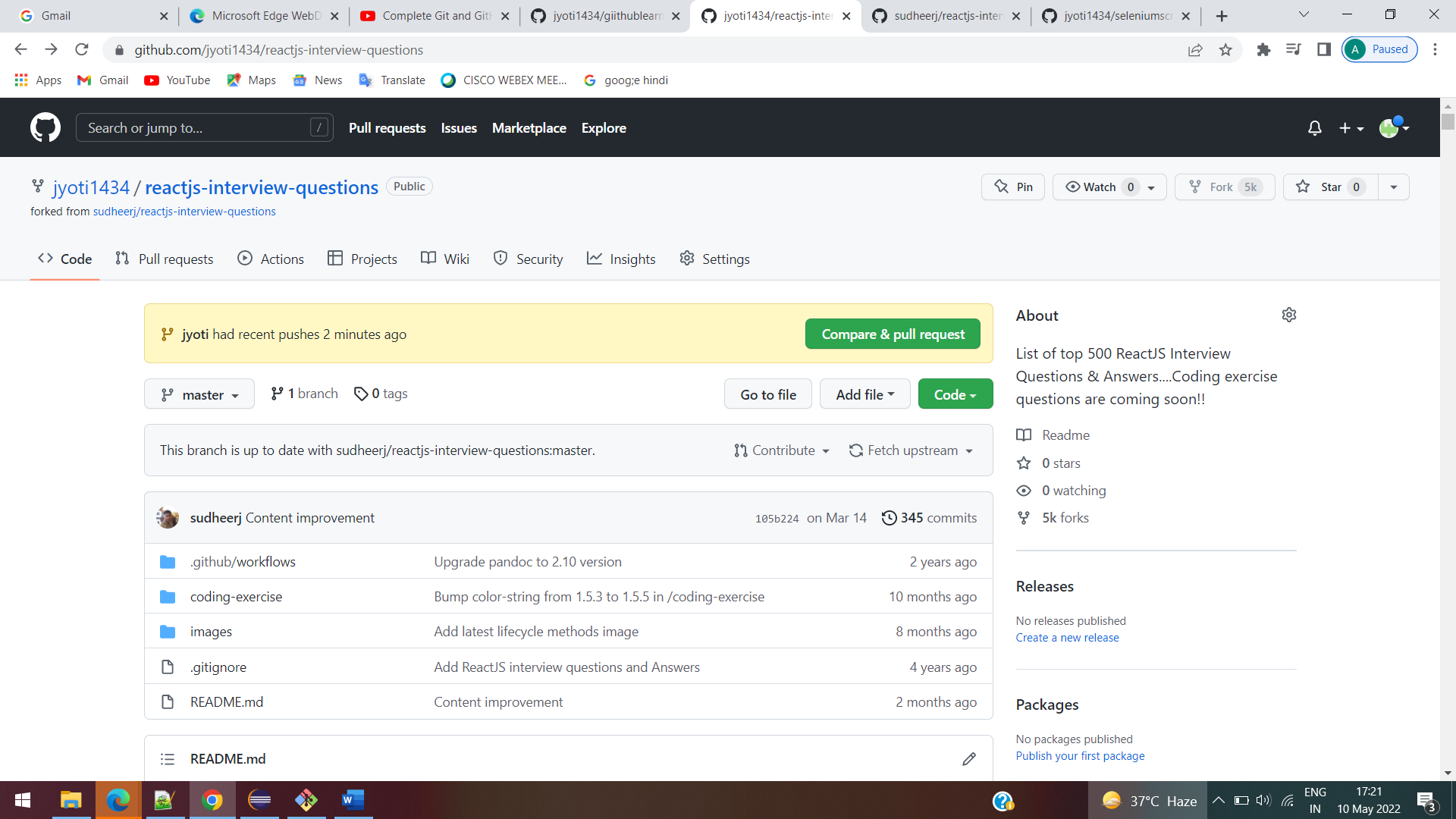
create mode 100644 README.TXT

push the files to your own branch

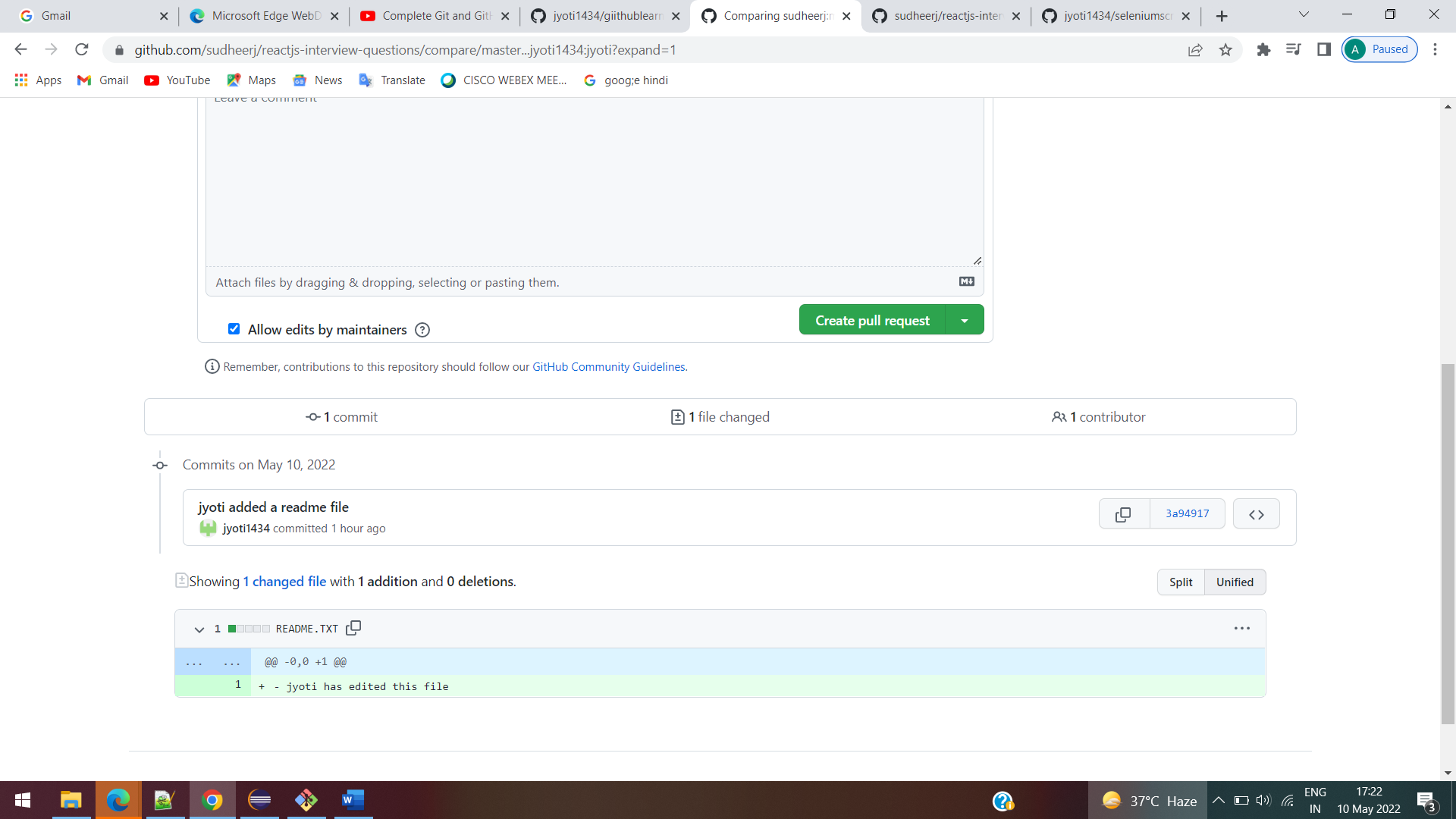
91939@LAPTOP-EPRK5L3P MINGW64 ~/desktop/project/reactjs-interview-questions (jyoti)

$ git push origin jyoti





Create a pull request



One pull request will be associated to one branch

If we commit multiple features and push to the repository then it will get added to the same pull request which will be difficult to review

So in easy way we need to create separate branch for separate feature

**REMOVING A COMING FROM THE PULL REQUST by force pushing to it**

If we have two commits in a single pull request we can delete the committed files

$git reset commit

$git status

$git add .

$git stash---names.txt will be removed

**Force push**

$ git push origin jyoti -f

MERGING A PULL REQUEST