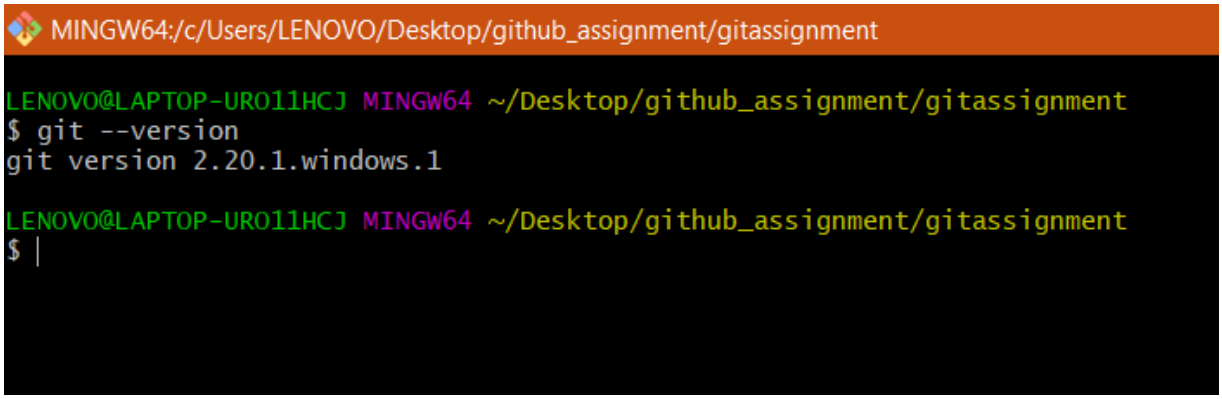
Introduction to Version Control

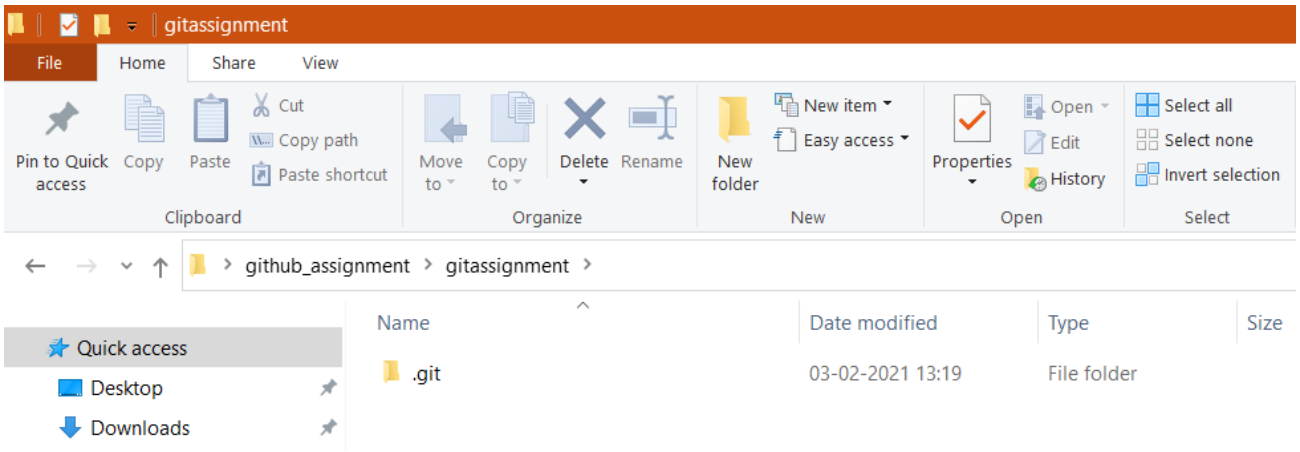
Assignment

1. Git Setup



2. Initialize a Git Repository

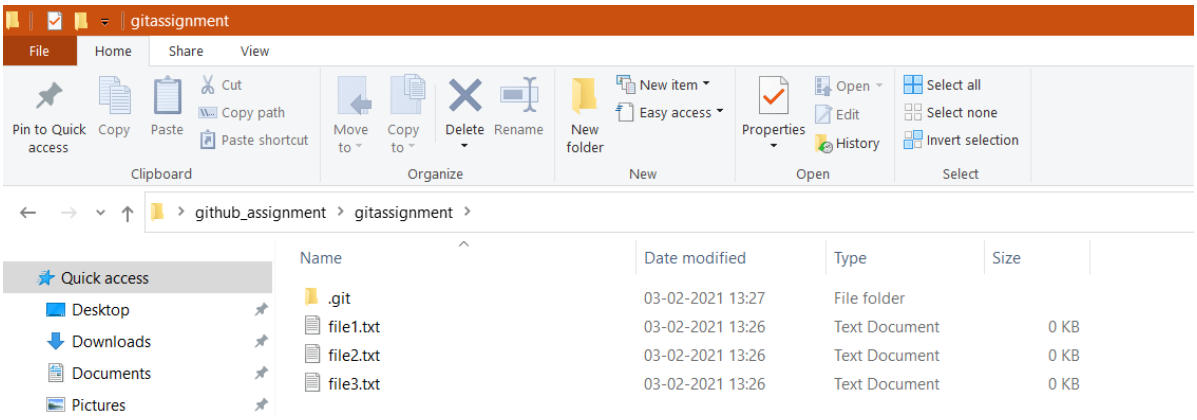
- ⇨ **Open git bash inside the project folder (gitassignment is the name of my project folder)**
- ⇨ **git init**



3. Add files to the repository

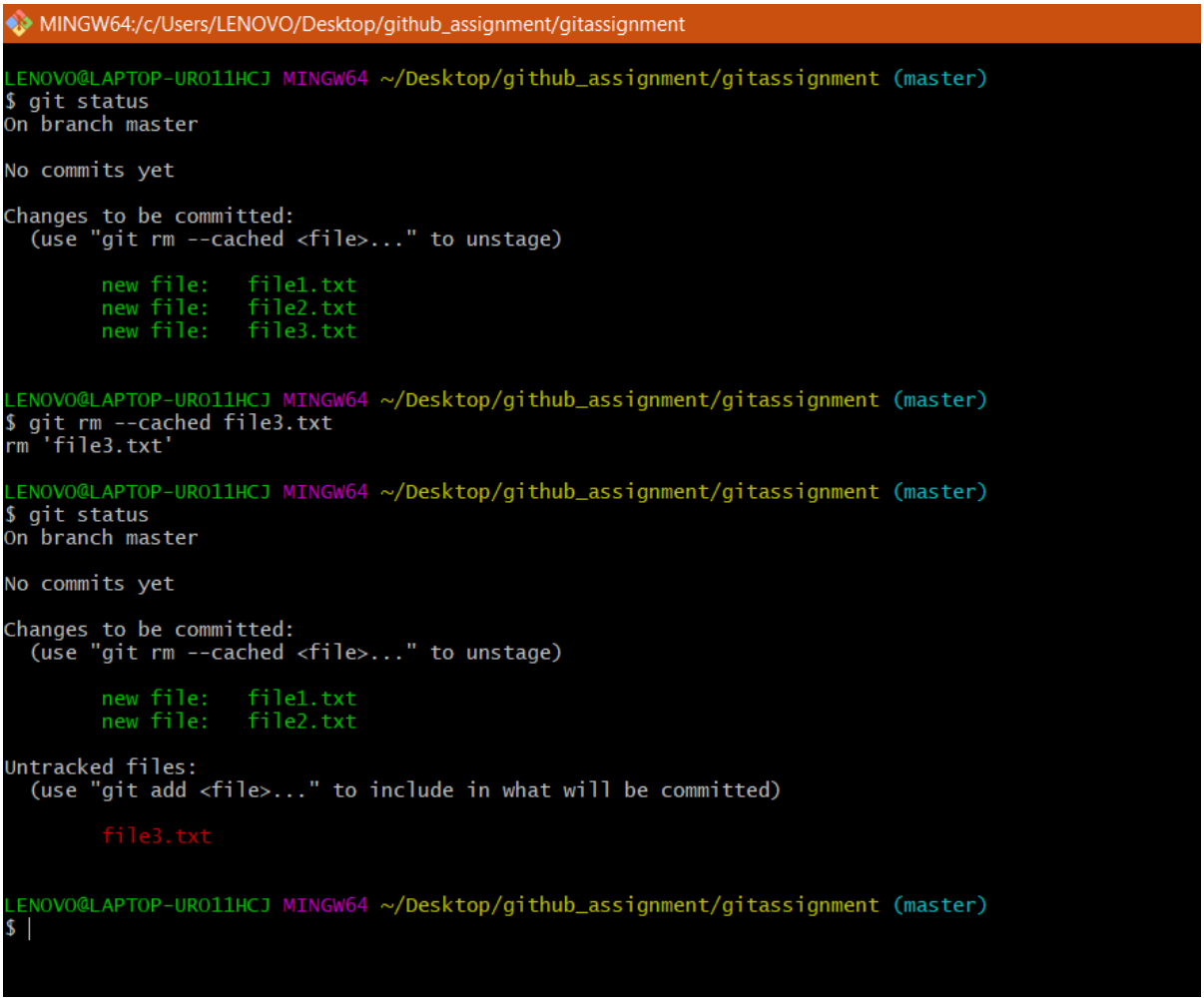
⇨ **Using touch filename create files**
⇨ **Git add .**

```
MINGW64:/c/Users/LENOVO/Desktop/github_assignment/gitassignment

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ touch file1.txt

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ touch file2.txt

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ touch file3.txt

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git status
On branch master

No commits yet

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

        file1.txt
        file2.txt
        file3.txt

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

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git add .

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git status
On branch master

No commits yet

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

        new file:   file1.txt
        new file:   file2.txt
        new file:   file3.txt
```



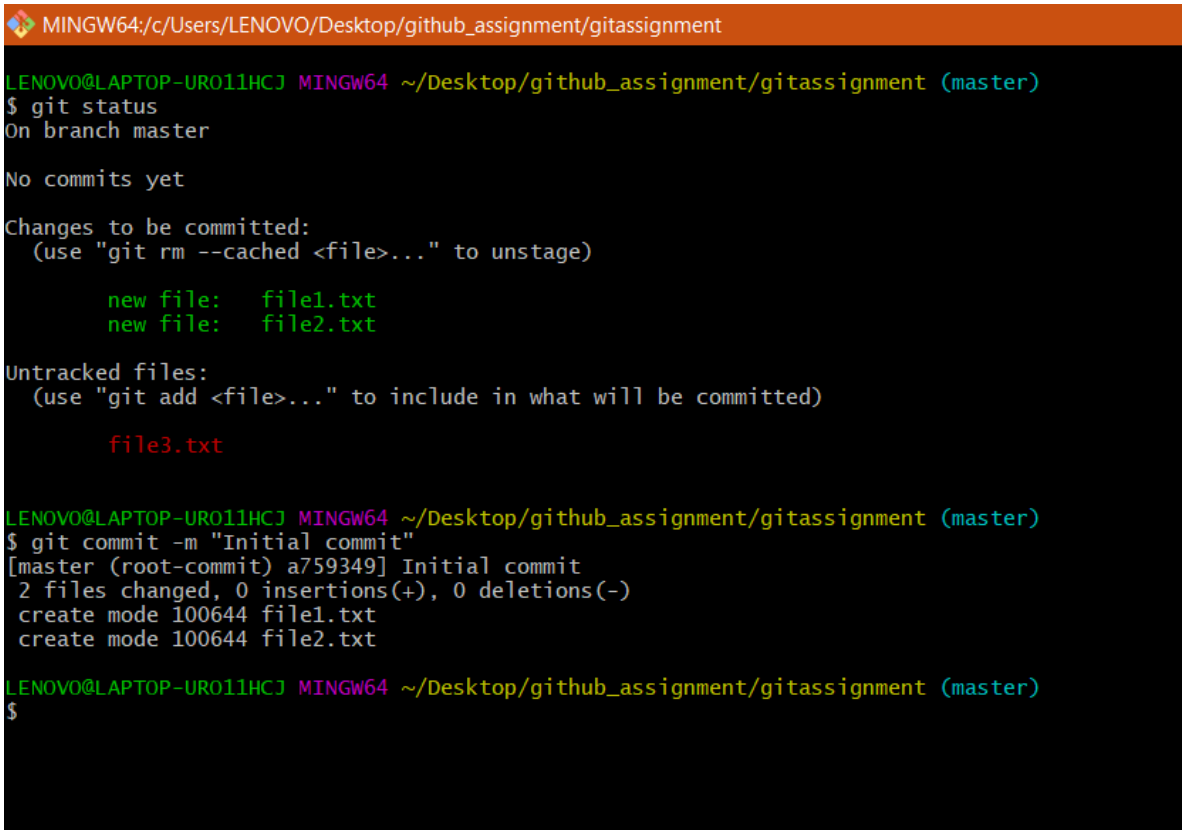
4. Unstage 1 file

⇨ Git rm --cached filename



5. **Commit the file**

   ⇨ **Git commit –m “message”**

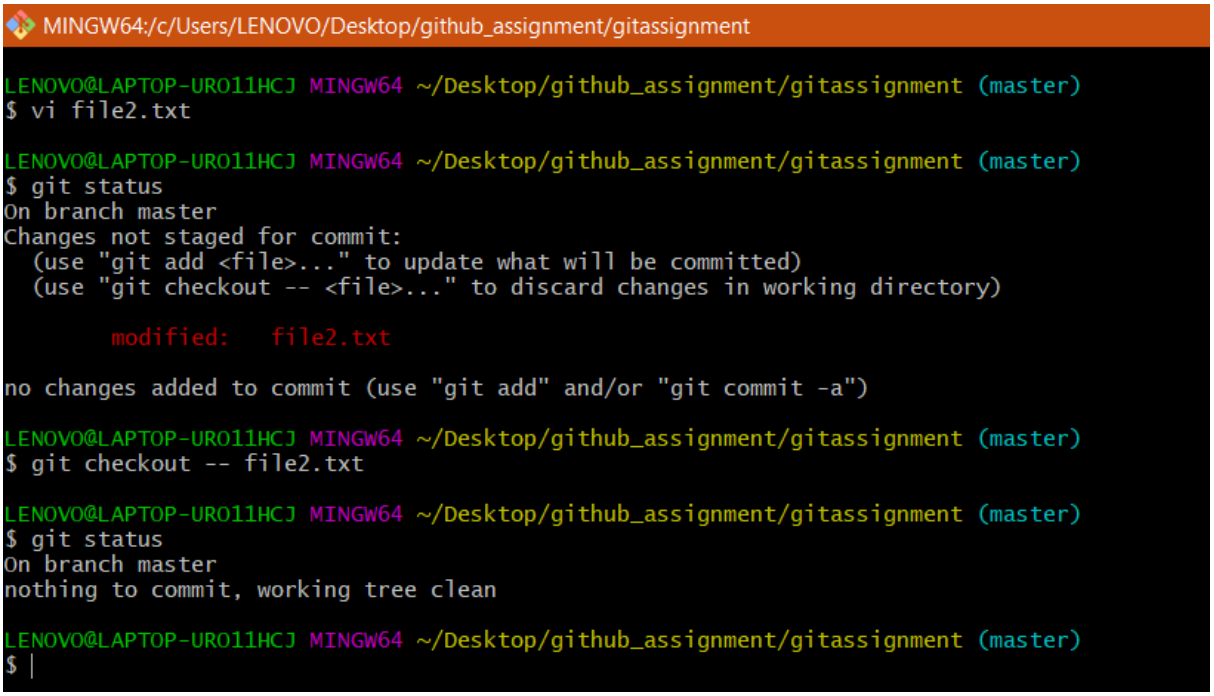


6. **Add a remote**

   ⇨ **Create a repository on github and the copy the https url**
   ⇨ **In gitbash write**
   ⇨ **Git remote add origin <url>**
   ⇨ **https://github.com/damineesaini/MyRepository (link of my git repository)**

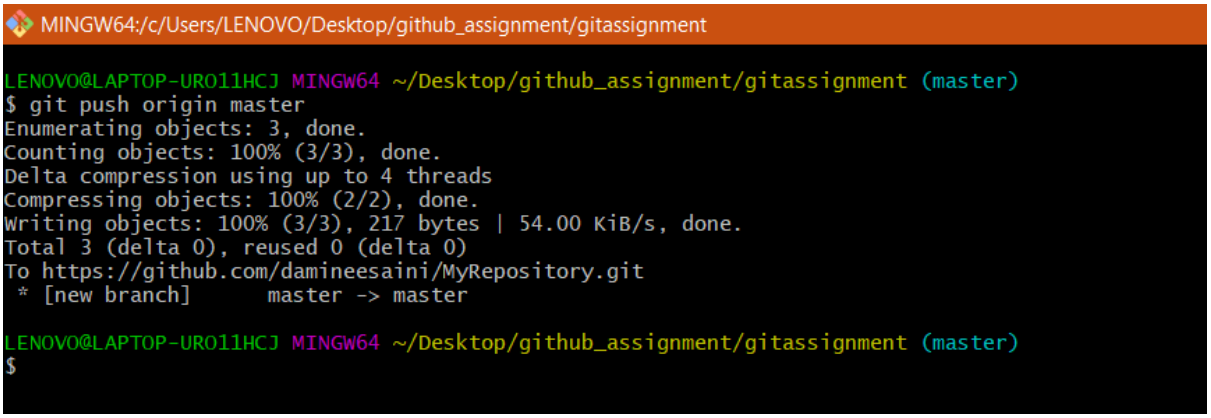
7. Undo changes to a particular file

⇨ Git checkout -- <filename>



8. Push changes to Github

⇨ Git push origin master



9. **Clone the repository**

⇨ **Git clone <url of repository from github>**

⇨ **Cloned the repository (MyRepository) in gitclone folder**

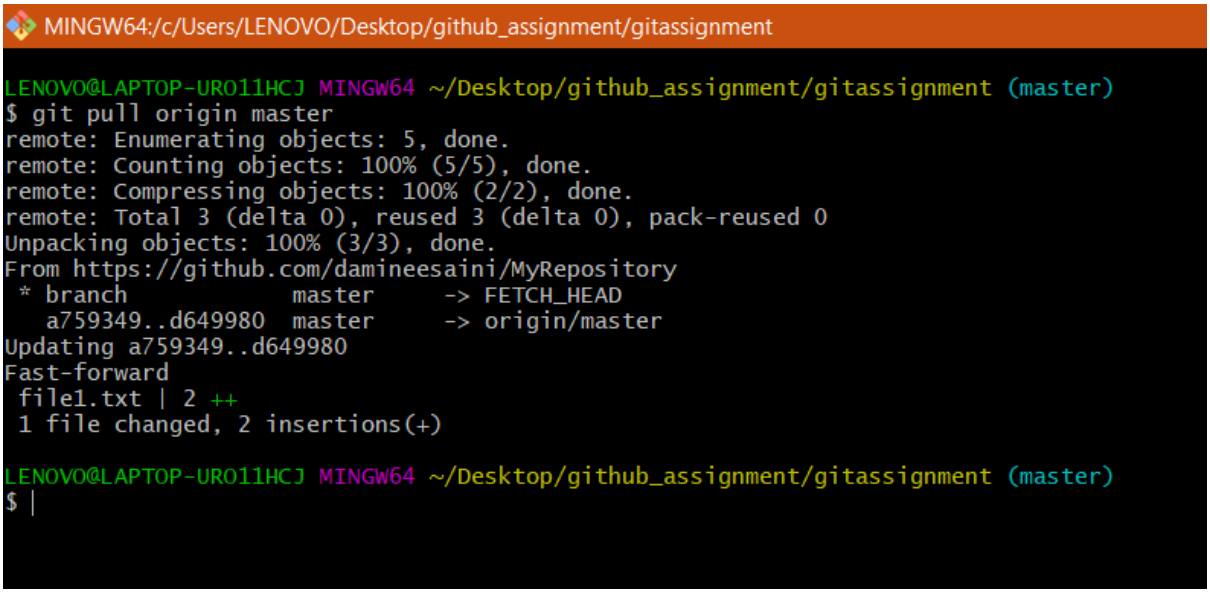
```
MINGW64:/c/Users/LENOVO/Desktop/github_assignment/gitclone

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitclone
$ git clone https://github.com/damineesaini/MyRepository.git
Cloning into 'MyRepository'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitclone
$
```

10. **Add changes to one of the copies and pull the changes in the other.**

⇨ **Gitclone is the folder where I cloned my github repository (MyRepository). I did some changes in file1 here and pushed them**

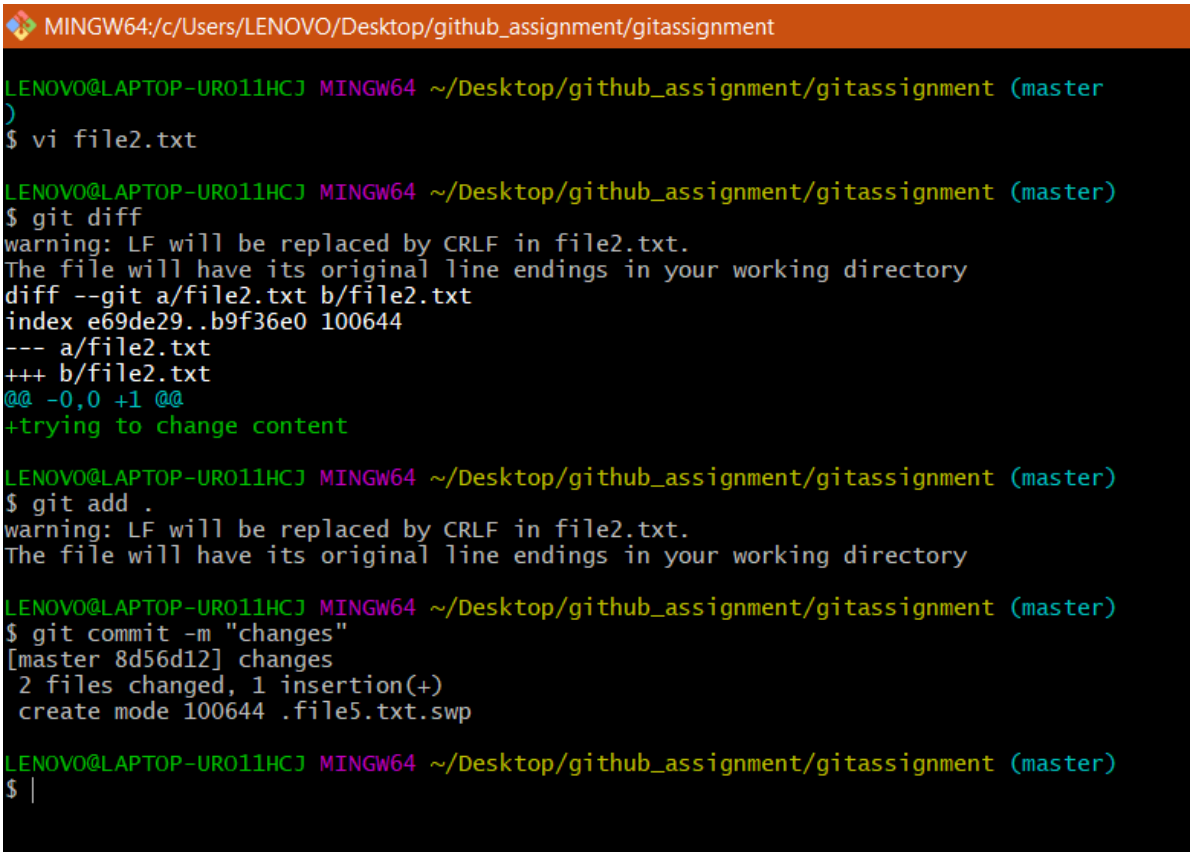
```
MINGW64:/c/Users/LENOVO/Desktop/github_assignment/gitclone/MyRepository
c
LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitclone
$ cd MyRepository/

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitclone/MyRepository (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   file1.txt

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

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitclone/MyRepository (master)
$ git add .

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitclone/MyRepository (master)
$ git commit -m "made some changes"
[master d649980] made some changes
 1 file changed, 2 insertions(+)

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitclone/MyRepository (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 292 bytes | 41.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/damineesaini/MyRepository.git
   a759349..d649980  master -> master

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitclone/MyRepository (master)
$
```

⇨ **Gitassignment is my main project folder which I created initially.**
⇨ **Pulled changes in it using => git pull origin master**



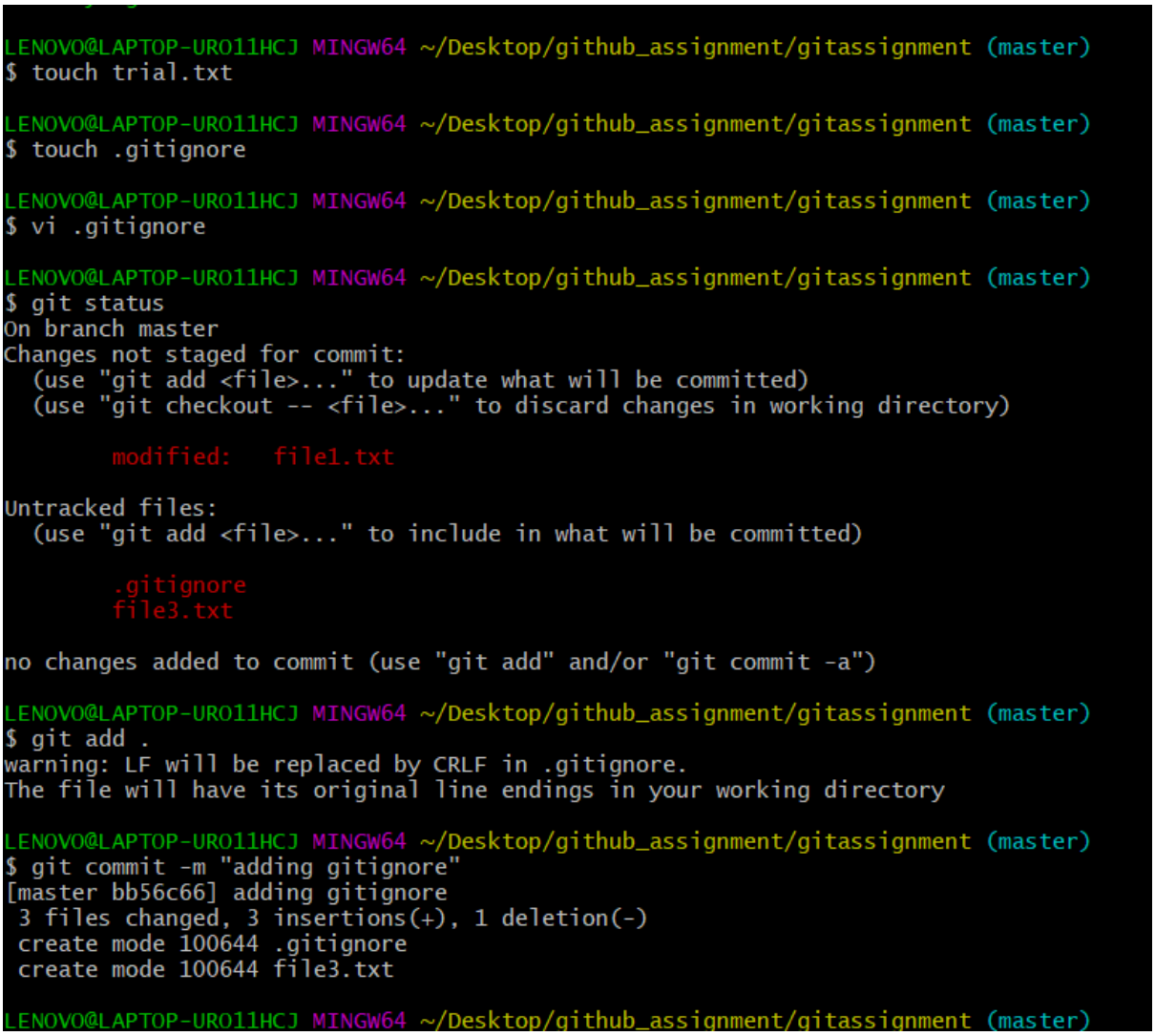
11. Check differences between a file and its staged version

⇨ **Git diff**



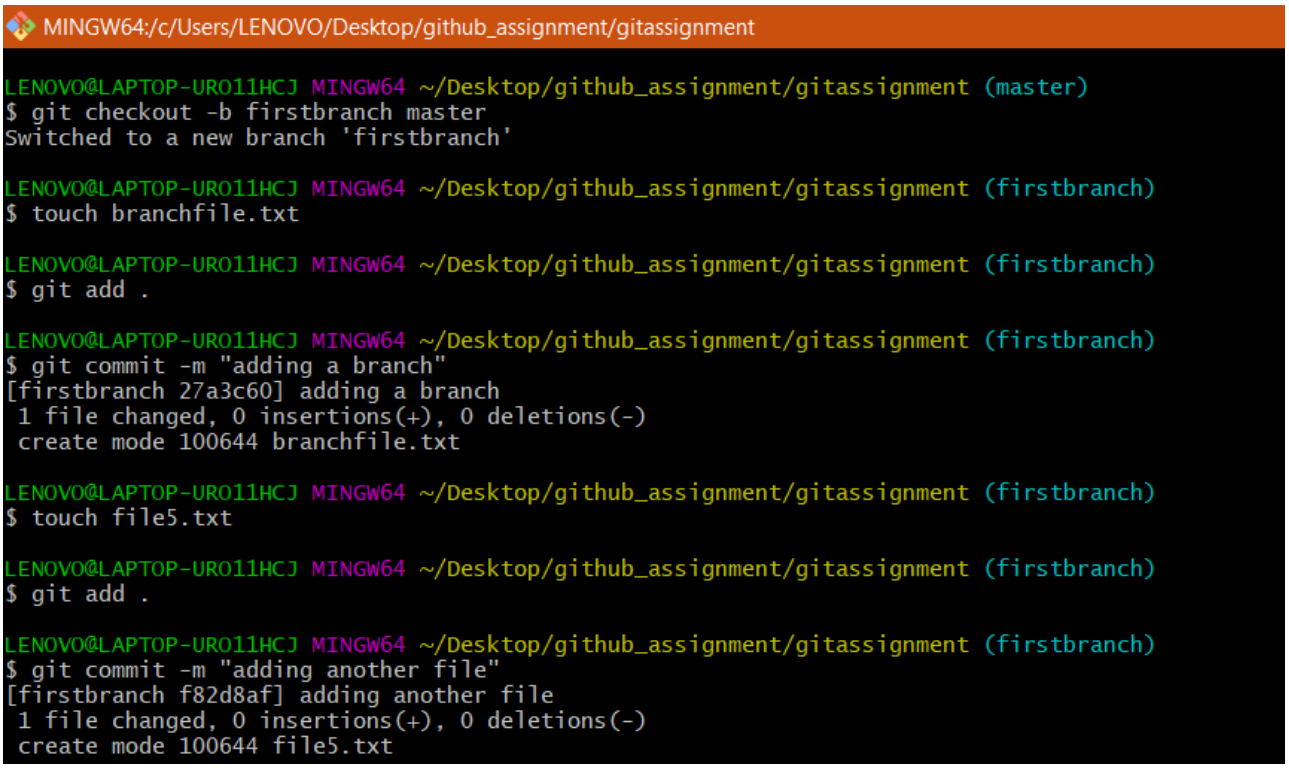
12. Ignore a few files to be checked in

- ⇨ **Created .gitignore and mentioned the filename in it that you want to be ignored.**
- ⇨ **Then commit**



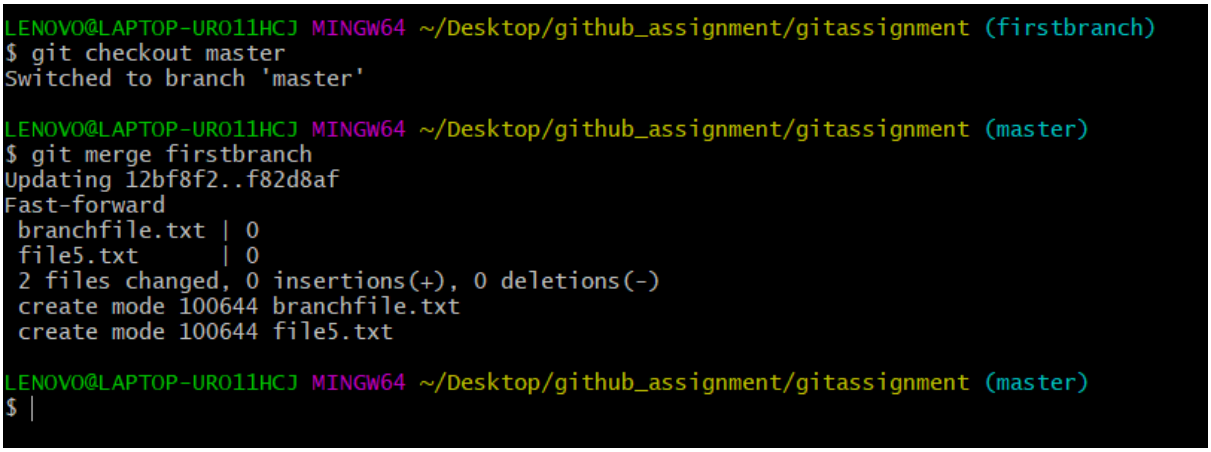
13. Create a new branch.

- ⇨ **Git checkout –b <branchname> master**
- ⇨ **And also added some files to the new branch**



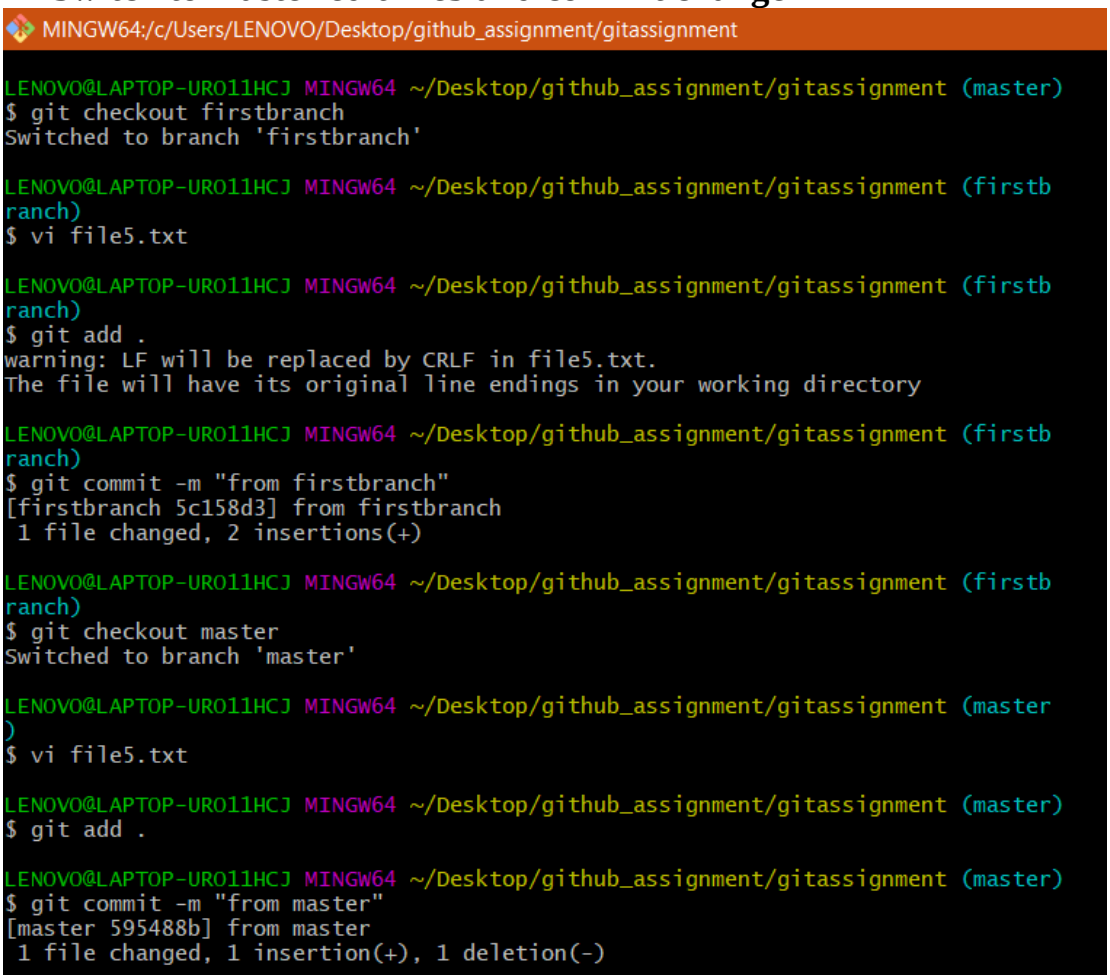
14. Diverge them with commits

- ⇨ **Git checkout master**
- ⇨ **Git merge firstbranch**



15. **Edit the same file at the same line on both branches and commit**

- ⇨ **Switched to newbrand and edited file5 and commit change**
- ⇨ **Switch to master edit file5 and commit change**



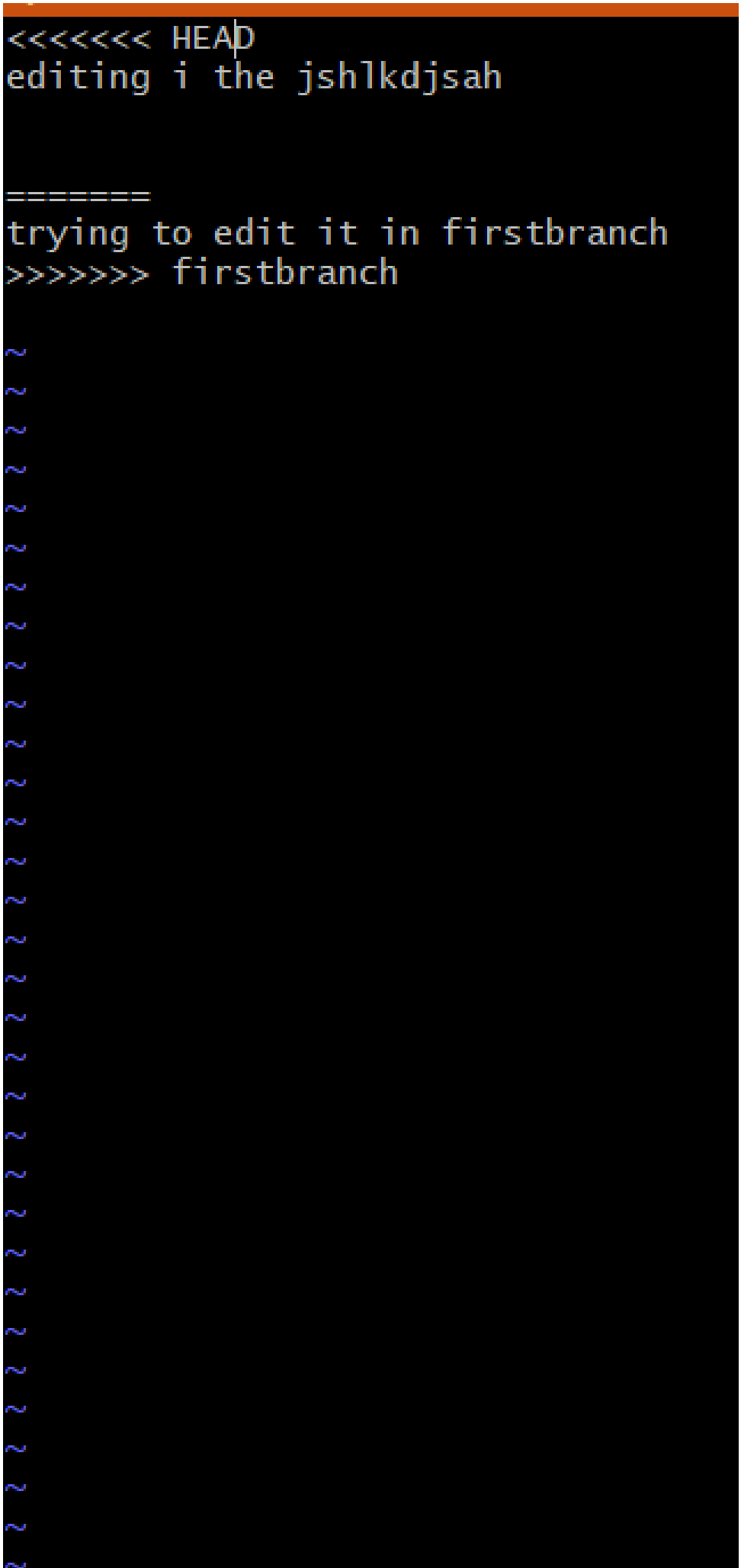
16. Try merging and resolve merge conflicts

- ⇨ **Try to merge new branch created with master**
- ⇨ **Conflict is shown and merge is failed**
- ⇨ **The edit the file5 choose to keep the correct content and then add and commit**

```
LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git merge firstbranch
Auto-merging file5.txt
CONFLICT (content): Merge conflict in file5.txt
Automatic merge failed; fix conflicts and then commit the result.
```

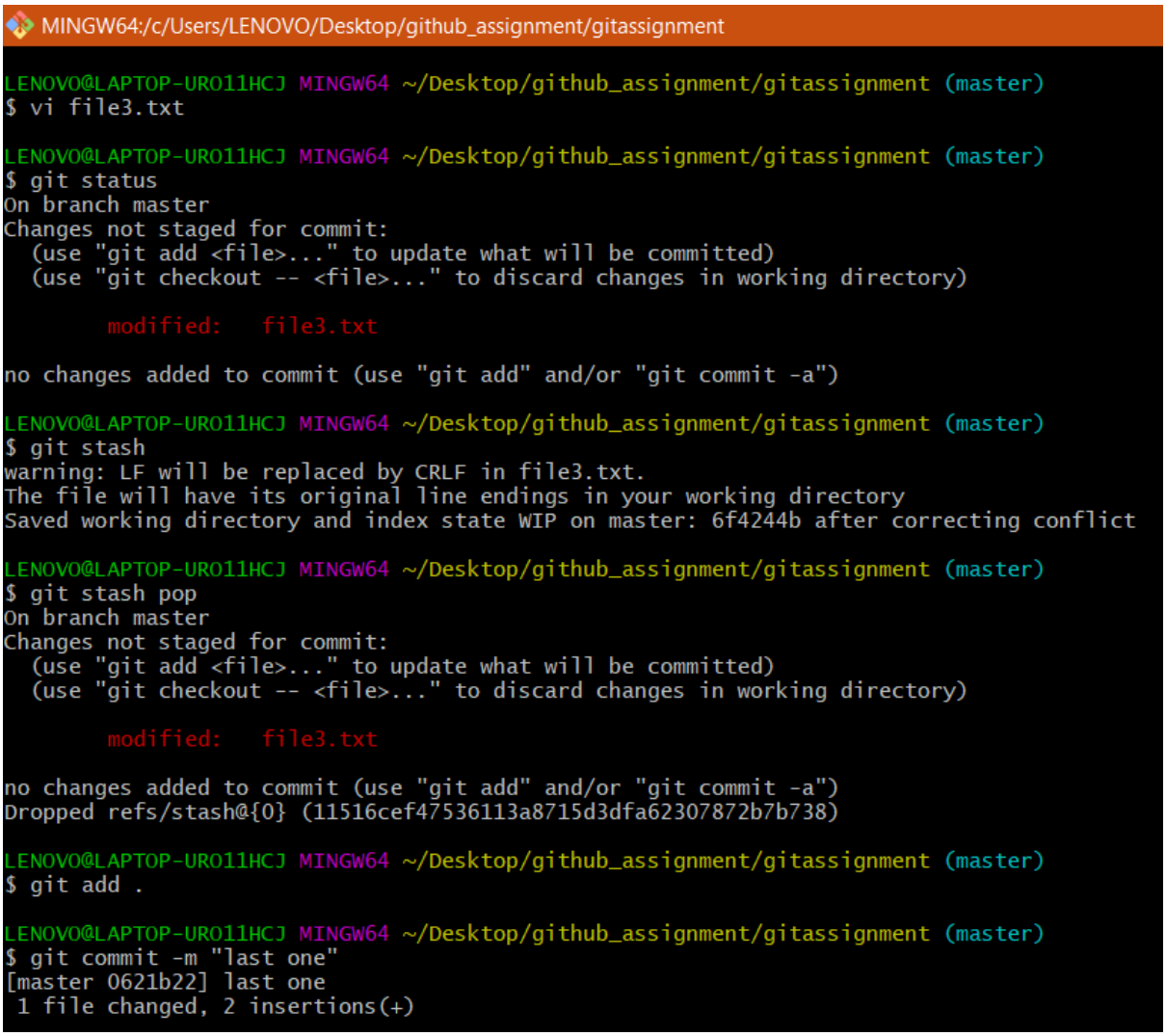
```
LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master|MERGING)
$ vi file5.txt

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master|MERGING)
$ git merge firstbranch
error: Merging is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master|MERGING)
$ git add .

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master|MERGING)
$ git commit -m "after correcting conflict"
[master 6f4244b] after correcting conflict

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$ git merge firstbranch
Already up to date.

LENOVO@LAPTOP-URO11HCJ MINGW64 ~/Desktop/github_assignment/gitassignment (master)
$
```

Below is the file5.txt with merge conflict


<<<<<<< HEAD
editing i the jshlkdjsah
=======
trying to edit it in firstbranch
>>>>>>> firstbranch

17. Stash the changes and pop them

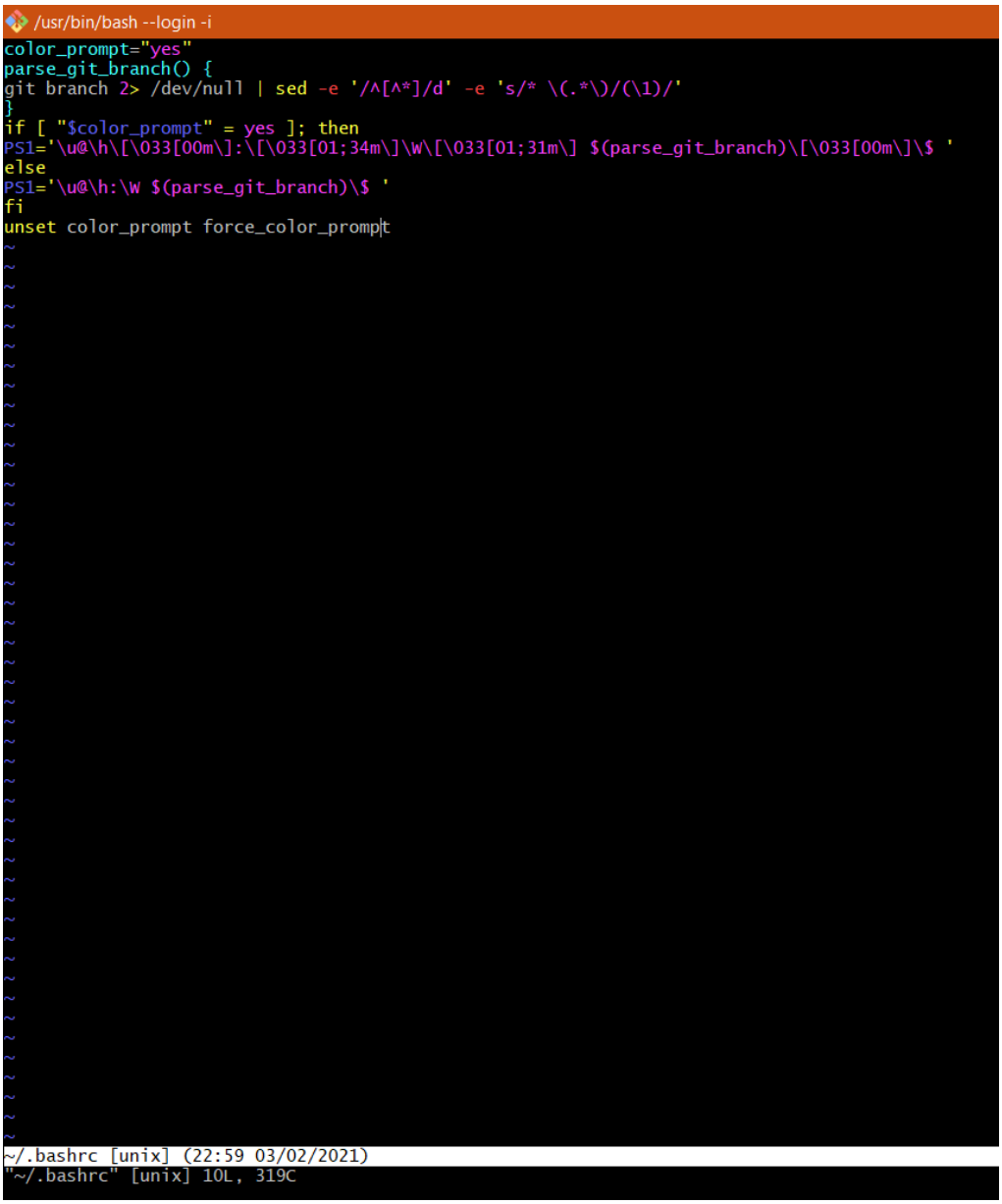
- ⇨ **Git stash**
- ⇨ **Git pop**

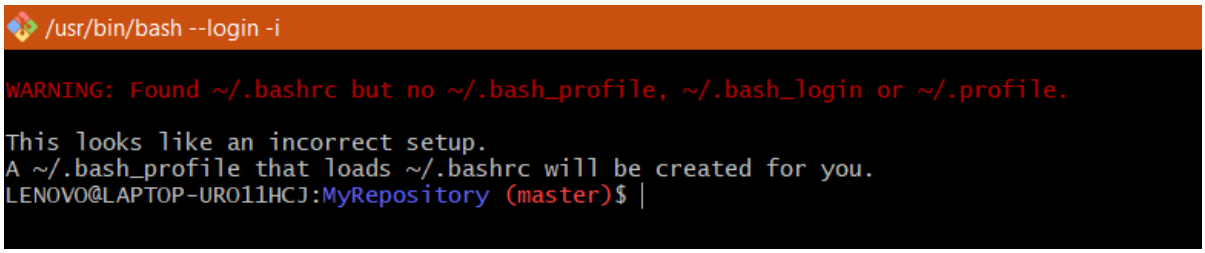


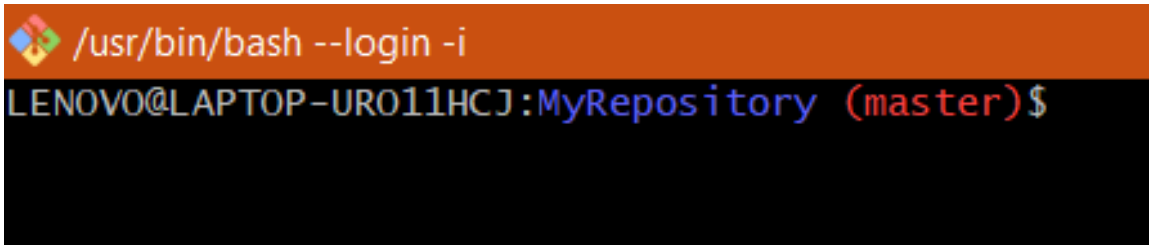
18. **Add the following code to your .bashrc file : color_prompt="yes"**

```
parse_git_branch() {
git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*\)/(\1)/'
}
if [ "$color_prompt" = yes ]; then
PS1='\u@\h\[\033[00m\]:\[\033[01;34m\]\W\[\033[01;31m\]
$(parse_git_branch)\[\033[00m\]\$ '
else
PS1='\u@\h:\W $(parse_git_branch)\$ '
fi
unset color_prompt force_color_prompt
```

- ⇨ **In windows gitbash bashrc file was created initially.**
- ⇨ **And that's why it showed warning after reopening the gitbash and then bashrc_profile was reated automatically**




/usr/bin/bash --login -i
WARNING: Found ~/.bashrc but no ~/.bash_profile, ~/.bash_login or ~/.profile.
This looks like an incorrect setup.
A ~/.bash_profile that loads ~/.bashrc will be created for you.
LENOVO@LAPTOP-URO11HCJ:MyRepository (master)$


/usr/bin/bash --login -i
LENOVO@LAPTOP-URO11HCJ:MyRepository (master)$

Bashrc file in Ubuntu (linux)