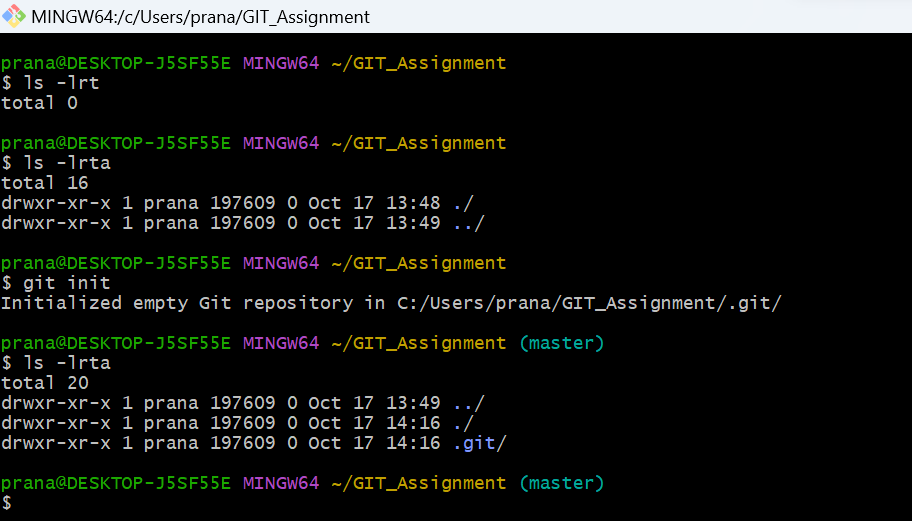
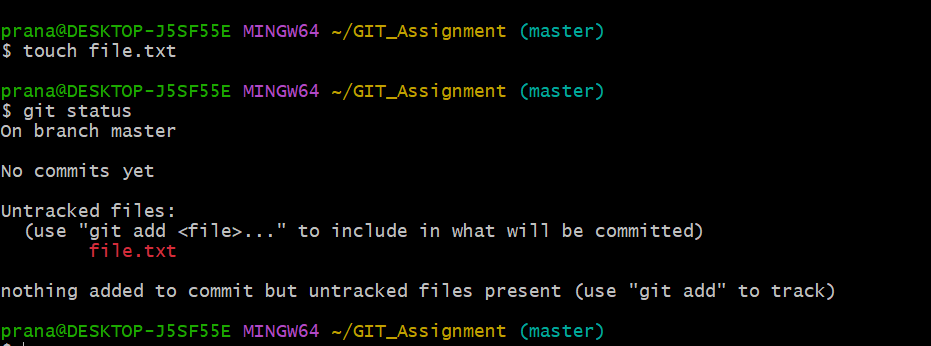
# **GIT Commands**

1. Initialize GIT in the local working repository . After initializing the git in local working directory .git extension is created which we can share the files in the directory with other users who wish to collaborate on the project. We use following command – **git init**



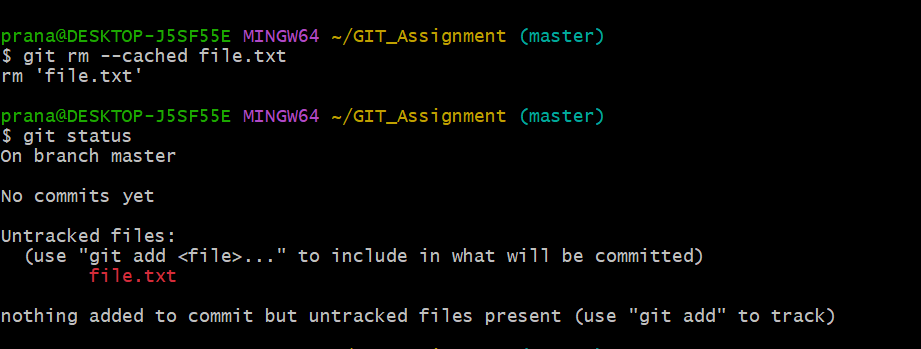
1. Create files in repository and check git status to check file changes in working directory which needs to be moved to staging area and tracking commits. We use following command – **git status**



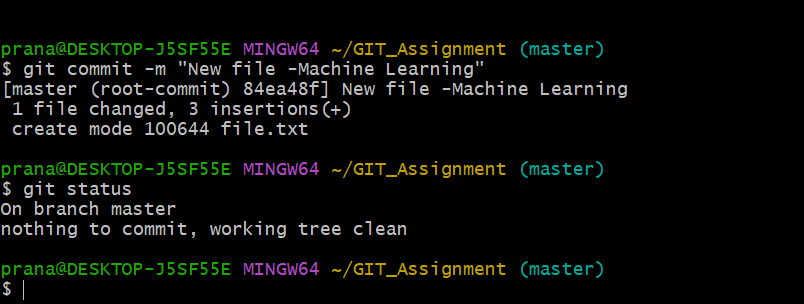
1. We want to add untracked changes in the local repository to staging area which is a pre-commit stage . We can add using following command – **git add <file> or git add .** for all files in the working repository.



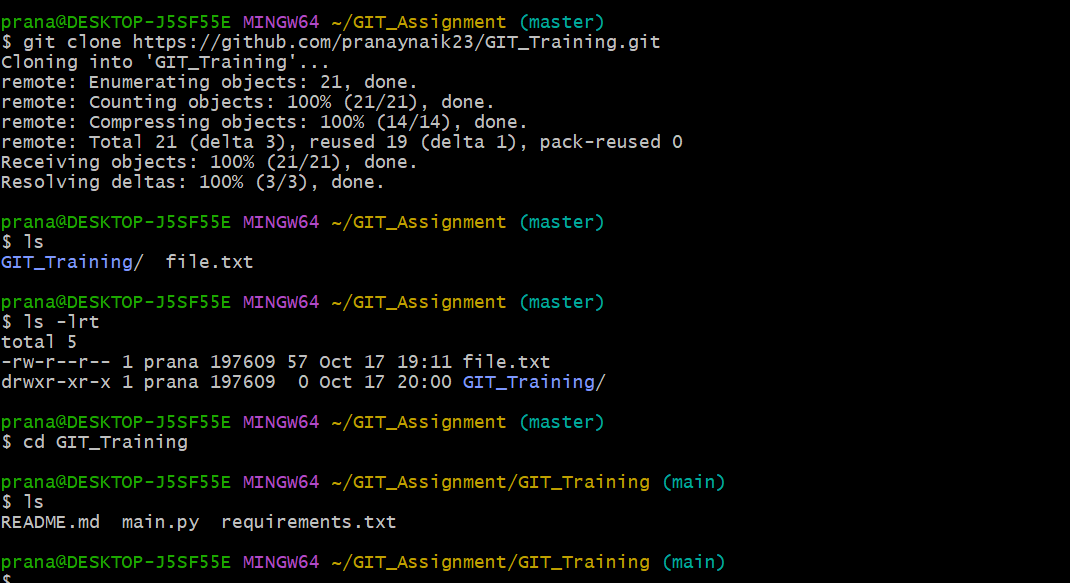
1. Now ,if we want to revert the changes in the file before commit , we can unstage the file from staging area to working repository using following command – **git rm --cached <file>**

****

1. After adding files again in staging area, we can commit the file , after the file is committed we can check the git status and we will observe no files in staging area . We can execute following command for commit – **git commit -m “comment about the file**”

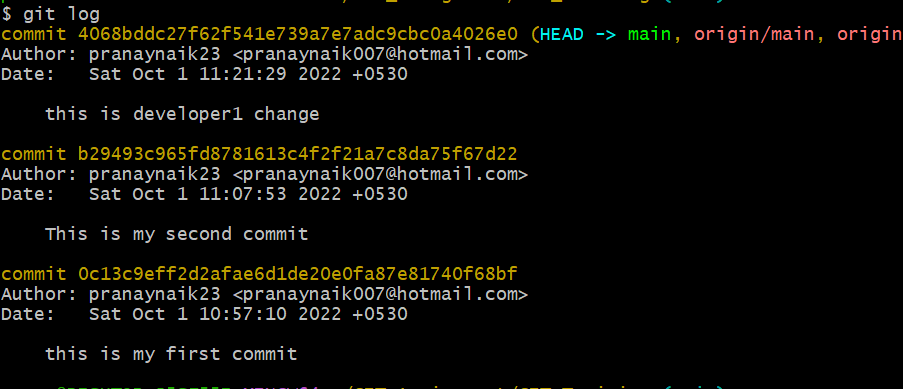
****

1. We can clone the remote repository like GitHub, bitbucket on to our local working repository using git . We can use following command for cloning the following repository in github<https://github.com/pranaynaik23/GIT\_Training.git> and work on the files in our local repository – **git clone <github url>**

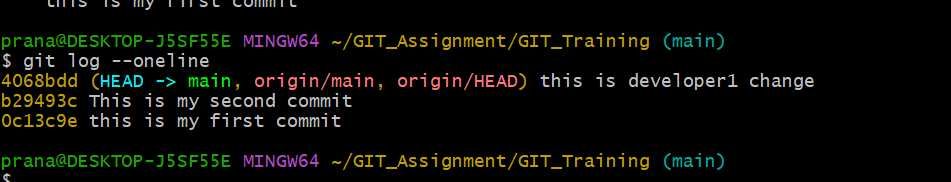
****

1. Check log file for any commits related to files in the repository. We use following command-

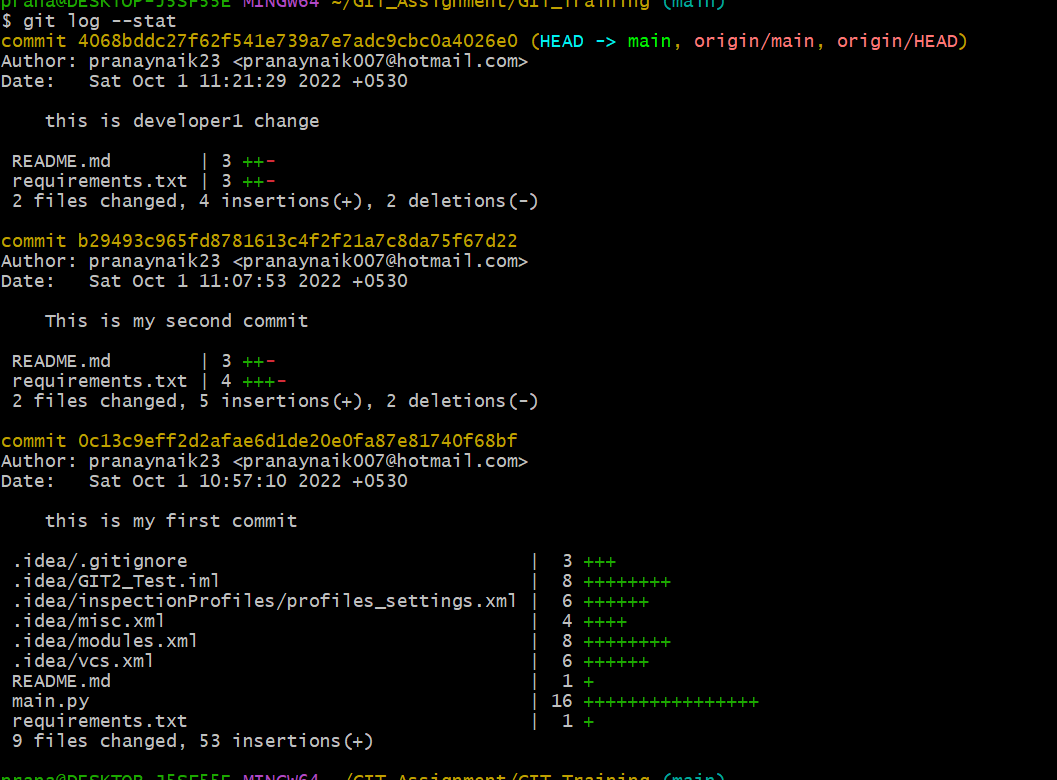
**git log**

****

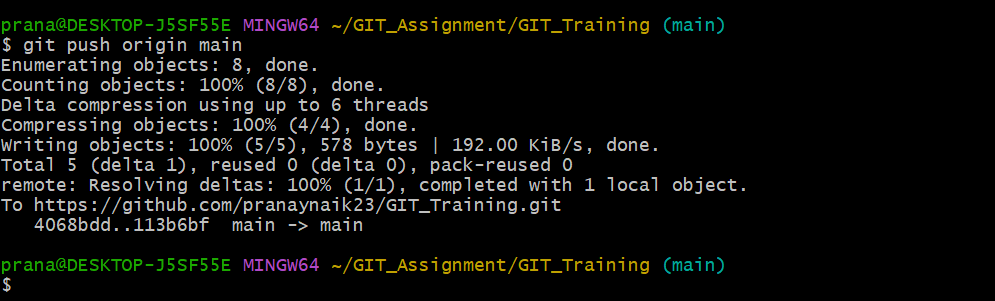
1. We can use **git log –oneline** to display shorter version of log file

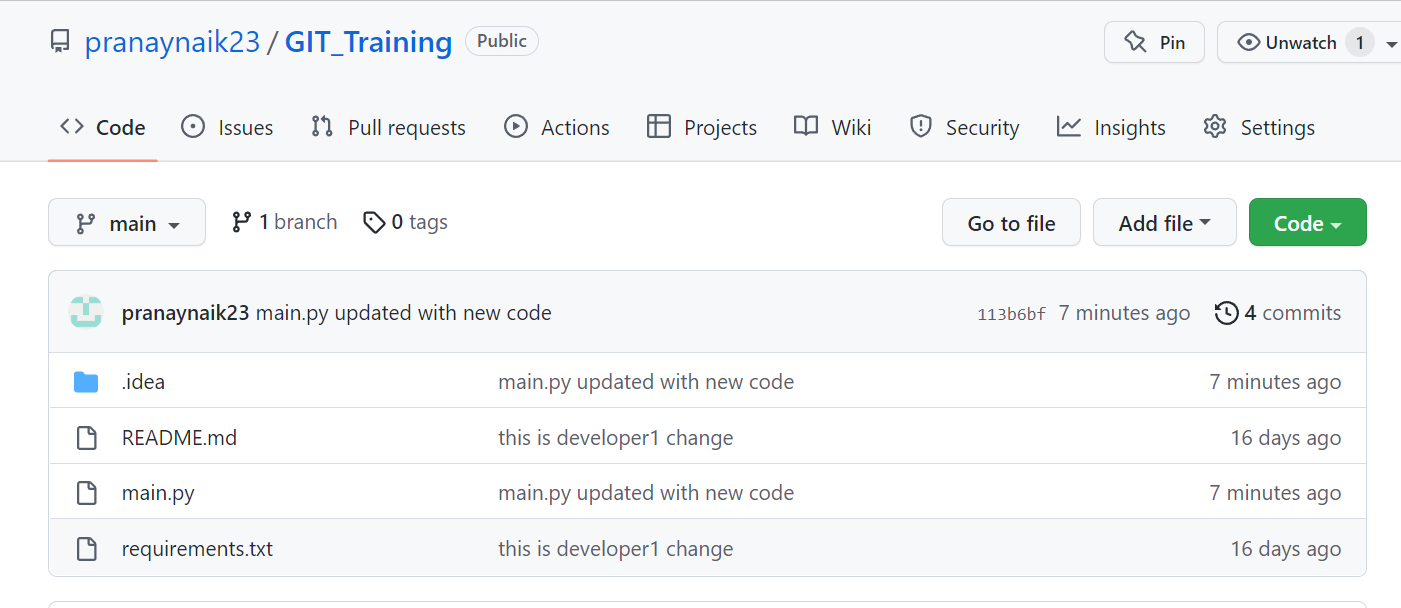
****

1. We can use **git log –stat** to detailed version of each file change with how many line insertions and deletions

****

1. We can push commited changes in our local working repository to remote github repository using following command – **git push origin <branch>**

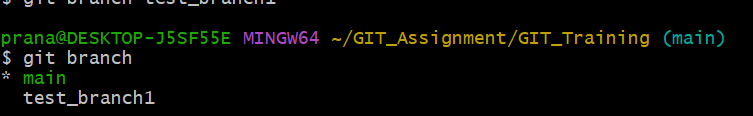
****

****

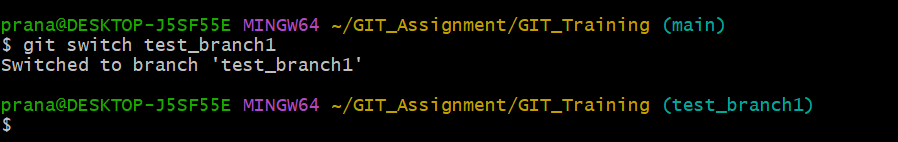
1. We can create branches for our main branch to add features to our source code and without affecting the source code repository. All changes in the branch will be merged with main branch in remote repository if there are no conflicts . We can create branch in our repository using following command – **git branch <branch\_name>**

****

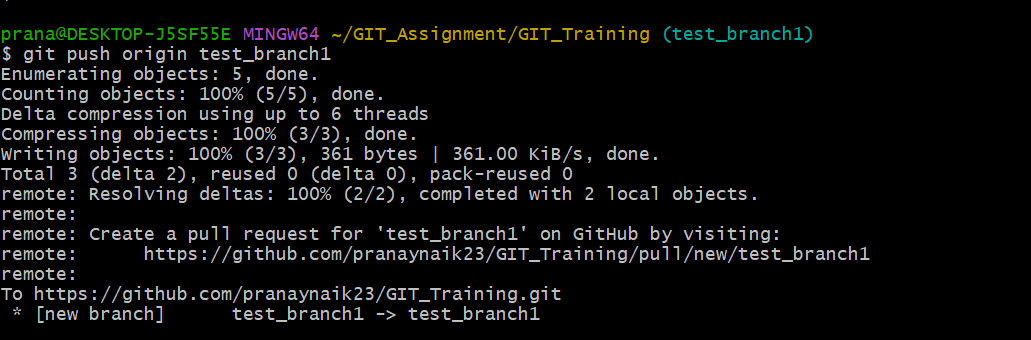
1. To list all branches ,use command – **git branch**

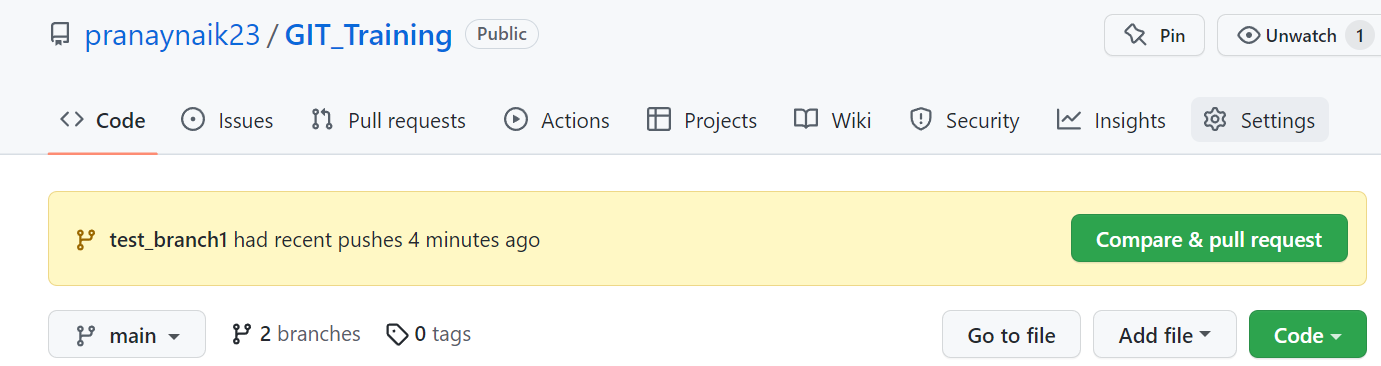
****

1. To switchover to git branch and start working on branch , use command – **git switch <branch\_name>**

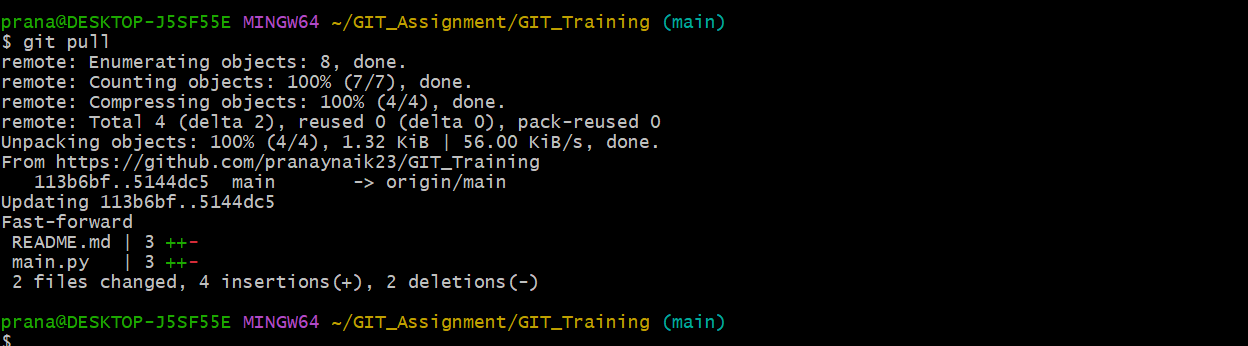
****

1. After creating a test branch ,we can push the commited branch changes to remote repository. Remote repository will get a pull request to accept or reject the changes to the main branch.

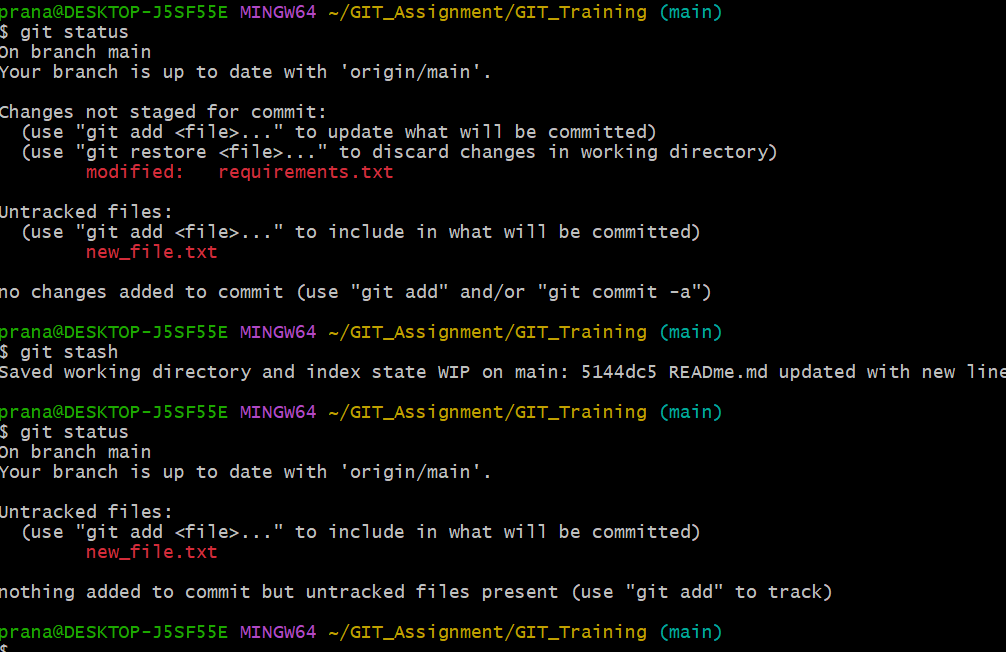
****

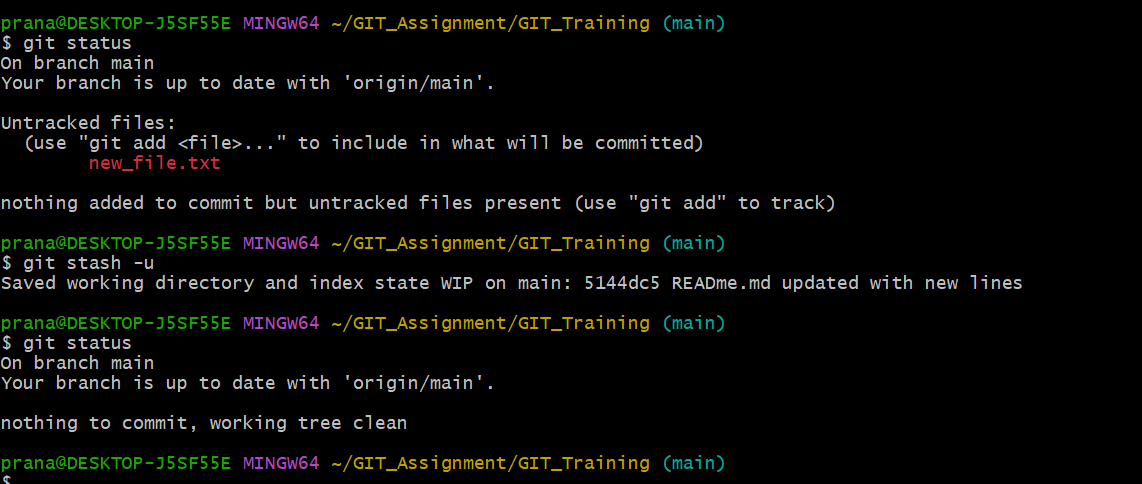
****

1. The changes in remote repository will be updated in the local repository after pulling the changes from remote repo. We use commands – **git pull** or 2-way authentication with **git fetch** and **git merge**

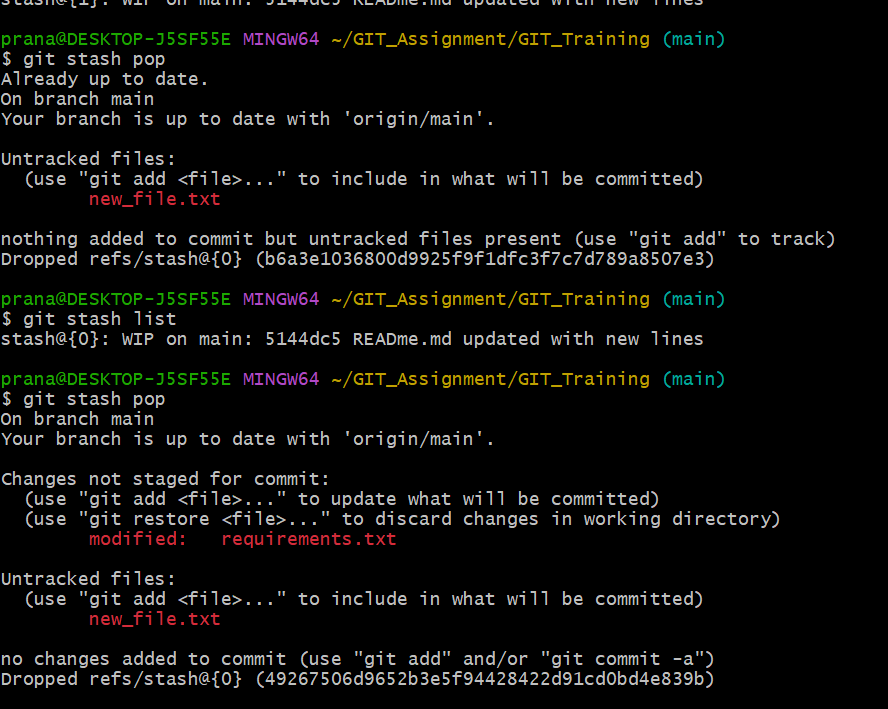


1. If we want to make some changes but needs to be applied later and work on some other commits we can use following command – **git stash .** Untracked files in working directory which are not yet in staging area can be stashed using **git stash -u**

****

****

1. To apply the stashed changes later , we can use following command : **git stash list** to check the list of stashes and then apply command **git stash pop .**This will stash the latest stash .

****