**DEPARTMENT OF INFORMATION TECHNOLOGY**

**COURSE CODE: DATE:**

**COURSE NAME: CLASS:**

**EXPERIMENT NO.1 and 2**

**CO/LO:**

**AIM / OBJECTIVE:**

To understand Version Control System / Source Code Management, install git and create a GitHub

account.

To Perform various GIT operations on local and Remote repositories using GIT Cheat-Sheet.

* Student should be able to add, commit and check status of the project.
* Student should be able to push and pull the files from the GitHub Repository
* Student should be able to access the files updated by other team member

**DESCRIPTION OF EXPERIMENT:**

Version control system.

A version control system, or VCS, tracks the history of changes as people and teams collaborate on projects together. As developers make changes to the project, any earlier version of the project can be recovered at any time.

Developers can review project history to find out:

* Which changes were made?
* Who made the changes?
* When were the changes made?
* Why were changes needed?

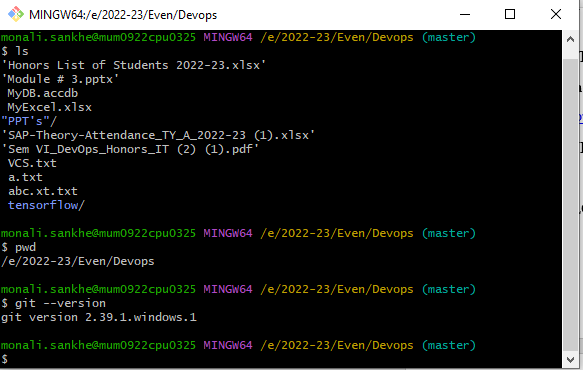
**INPUT DATA / DATASET:**

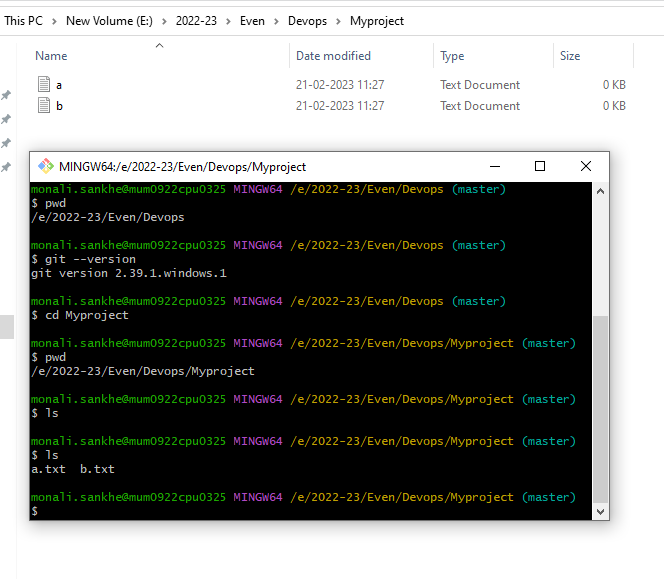
**Download and Install git from the link given below:**

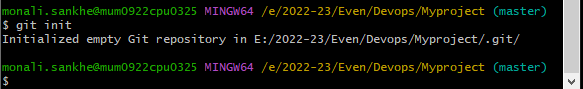
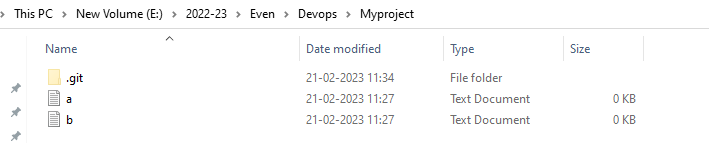
[Git for Windows](https://gitforwindows.org/)

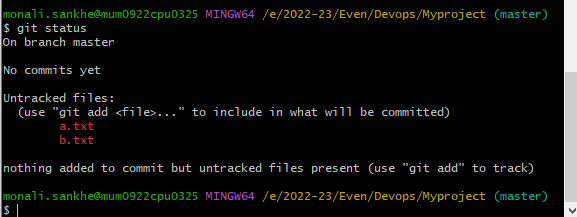
**PROCEDURE / ALGORITHM:**

1. Open GitBash in working directory
2. ls: displays list of files
3. pwd: print working directory
4. cd changing directory
5. git –version : version of git

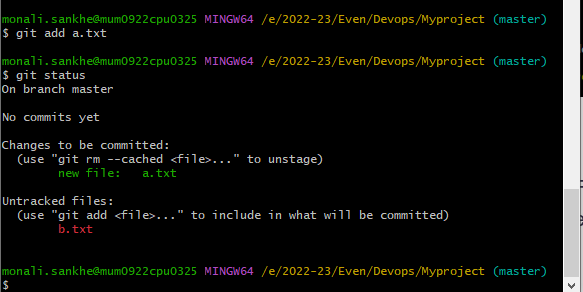




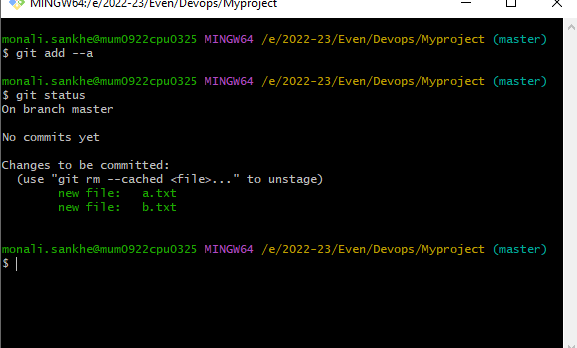
* **git init** will create a new local GIT repository. The following Git command will create a repository in the current directory:
* 
* 
* **git status** displays the list of changed files together with the files that are yet to be staged or committed.

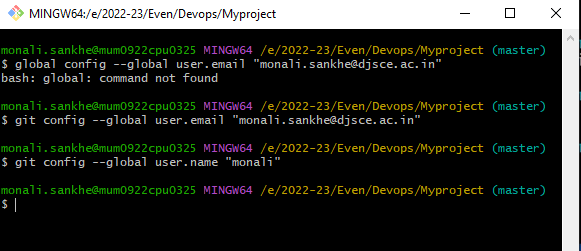


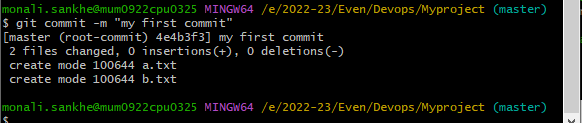
* **git add** is used to add files to the staging area. For example, the basic Git following command will index the a.txt file:

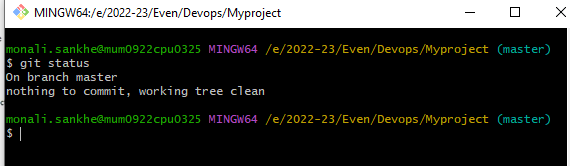


**git add** - -a will add all the files in git repository

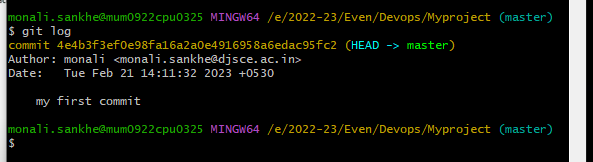


* **git config** can be used to set user-specific configuration values like email, username, file format, and so on. To illustrate, the command for setting up an email will look like this:
* git config --global user.email [youremail@example.com](mailto:youremail@example.com)
* git config --global user.name “yourname”
* 
* **git commit** will create a snapshot of the changes and save it to the git directory.
* git commit –m “Message to go with the commit here”



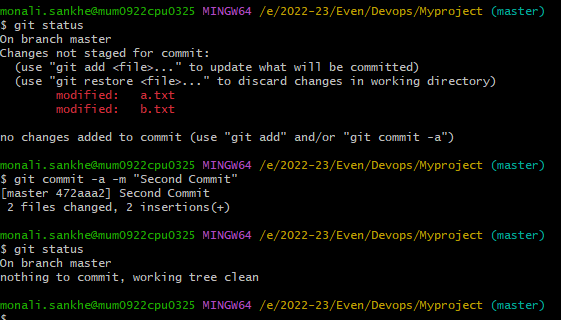


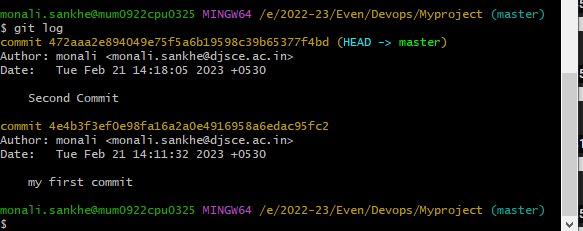
* **git log** is used to see the repository’s history by listing certain commit’s details. Running the command will get you an output that looks like this:



**After doing changes in file**



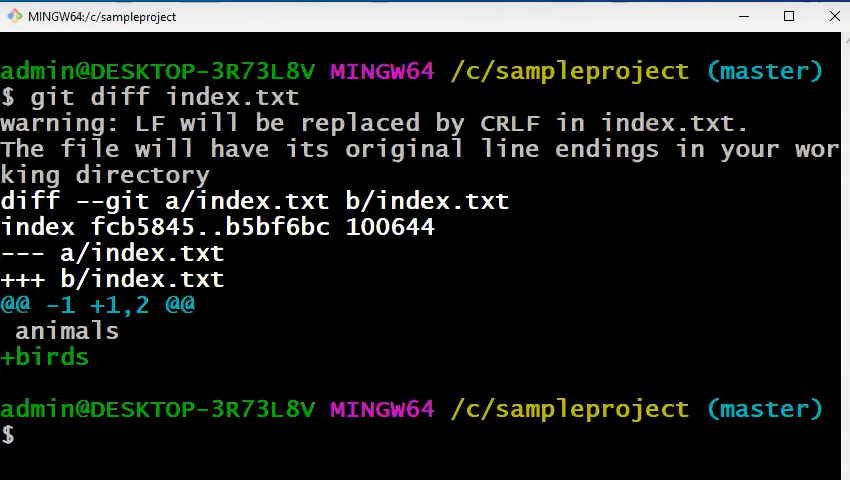


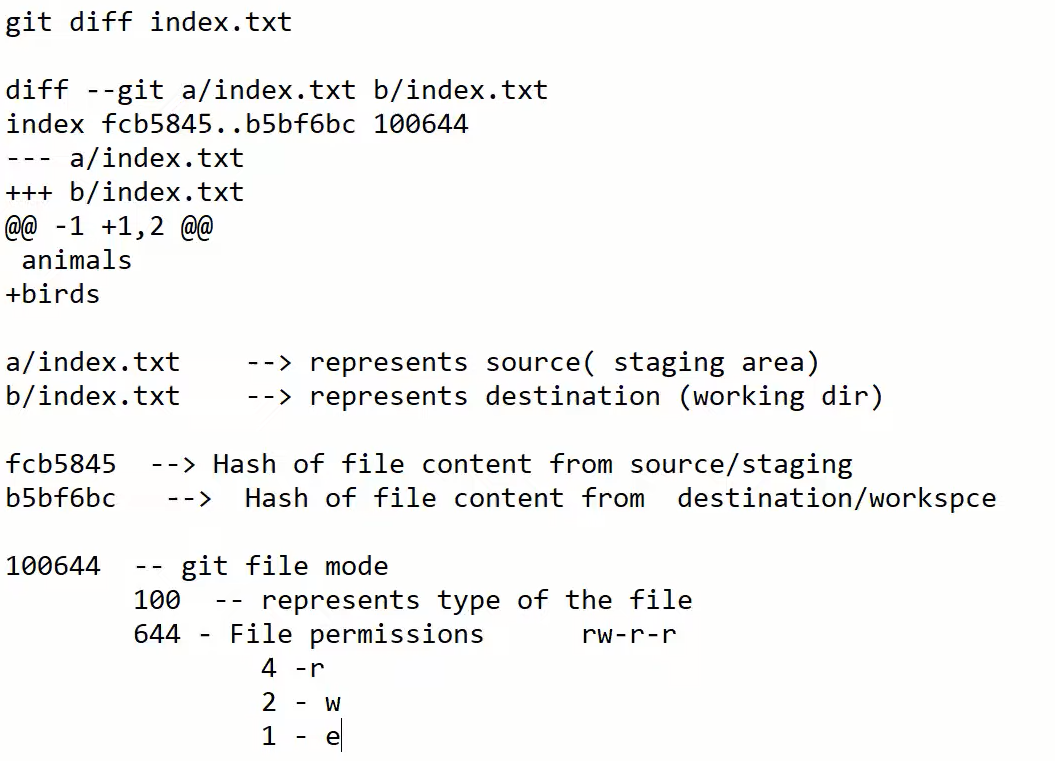


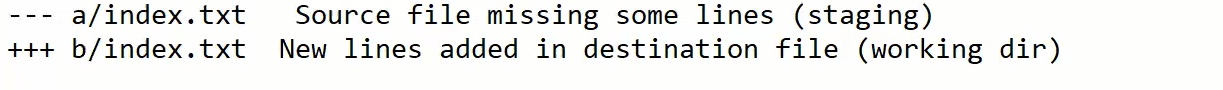
git diff:

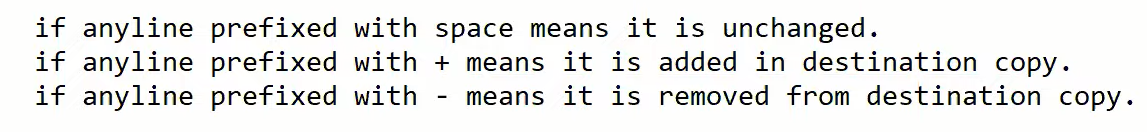
1. To see the the difference in the file content between working directory and stagging area









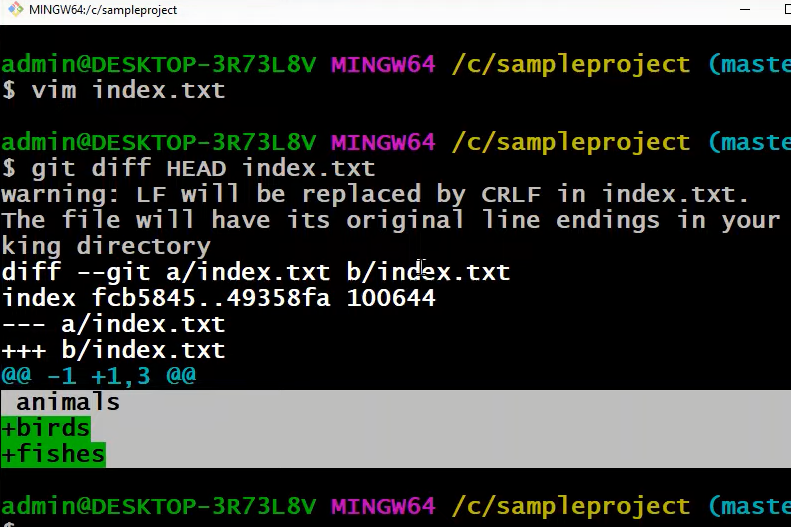


1. **Th see the difference between file content in working directory and last commit**

**\***last commit refred using HEAD

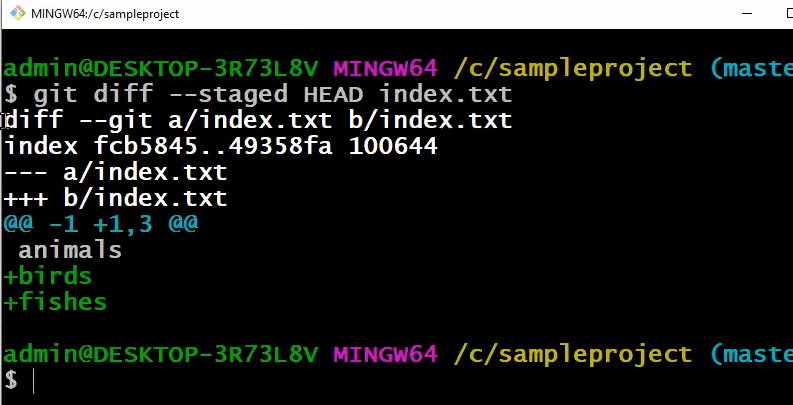
git diff HEAD index.txt



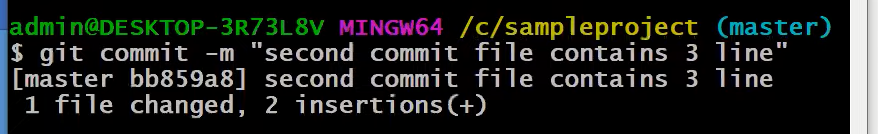


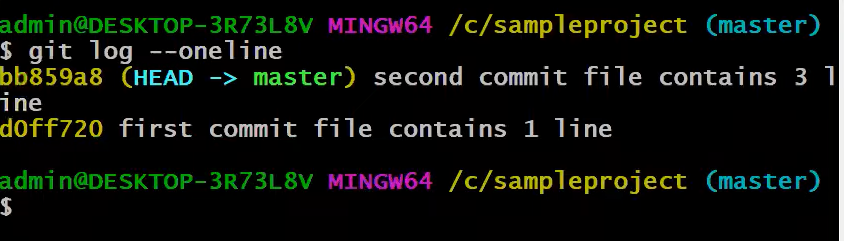
1. To see the difference file content in staging area and last commit

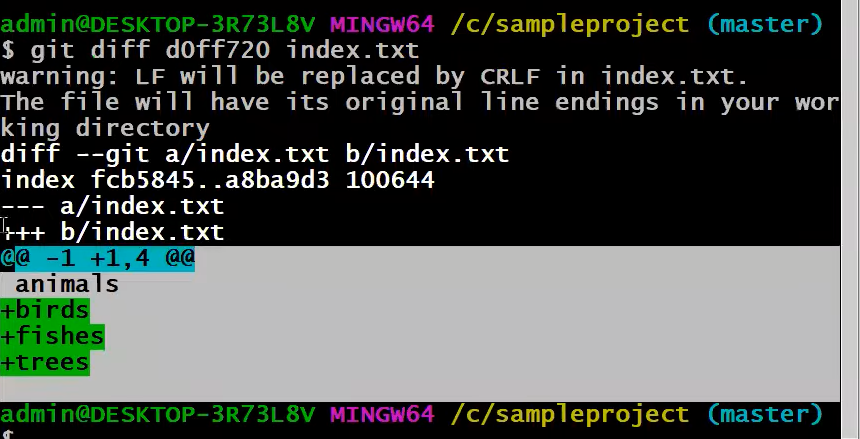
git diff –staged HEAD index.txt



1. To see the difference in file content between specific commit and working directory copy

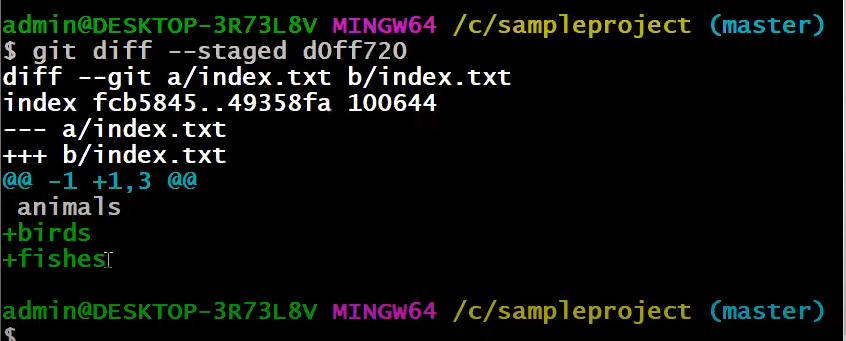


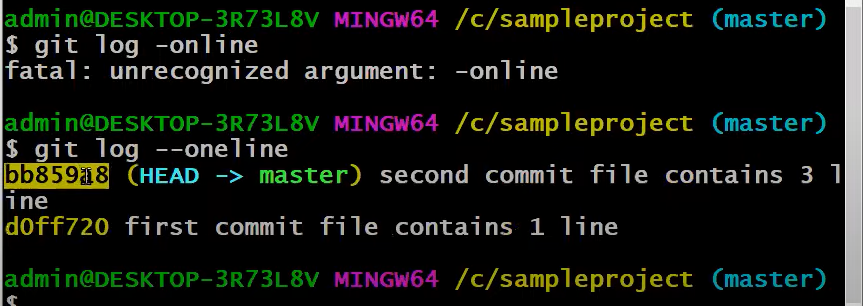




1. To see the difference in file content between specific commit and staging area copy

git diff - -staged d0ff720 index.txt





1. To see the difference in file content between 2 specified commits



1. To see the differences in content between to branches

Consider two branches:

-master

-test

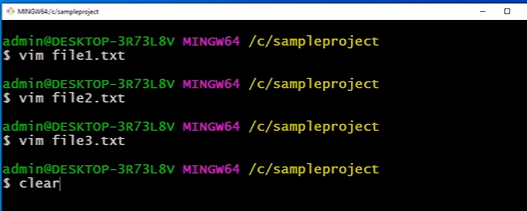
Git diff master test---> it shows differences in master and test branch

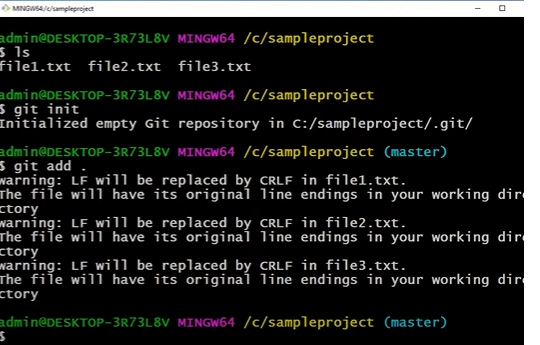
1. To see the differences between local and remote repositories

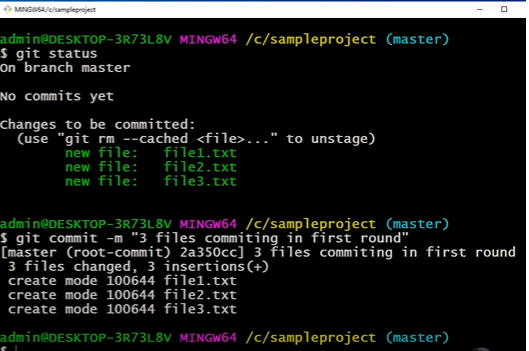
git diff master origin/master

**git rm**

Create three files using vim command

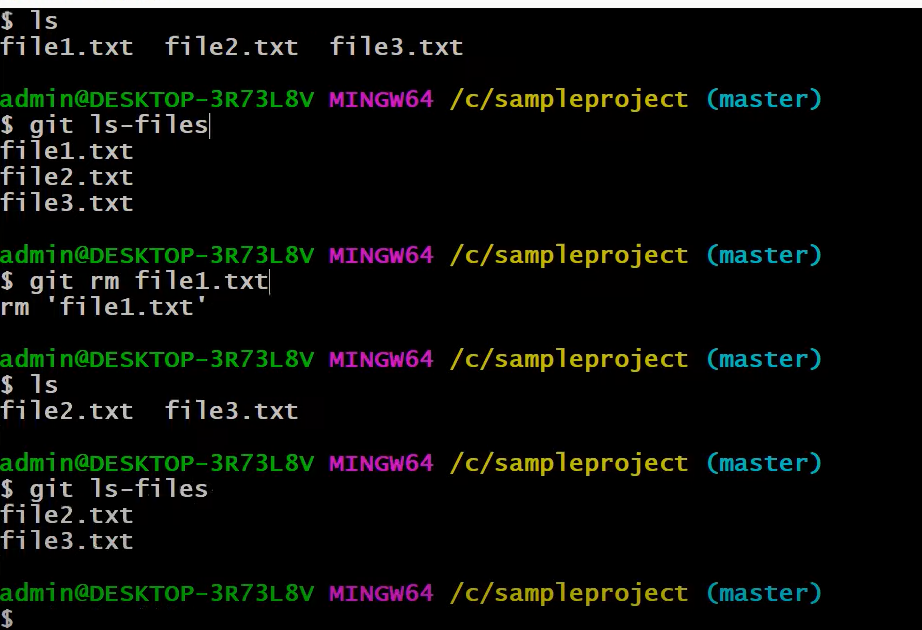






List the files working directory and staging area

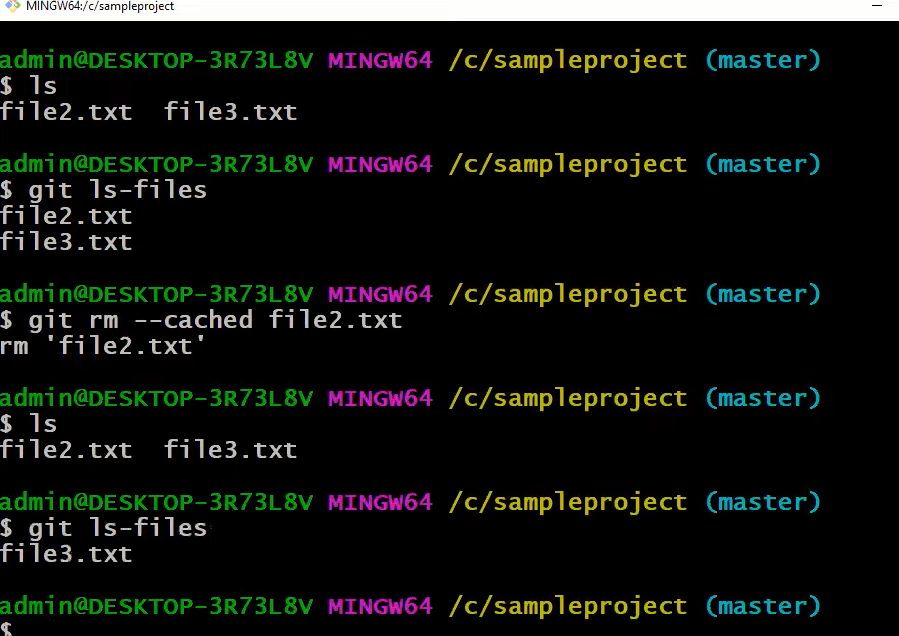
1. Remove the files from working directory and staging area



git rm –r .

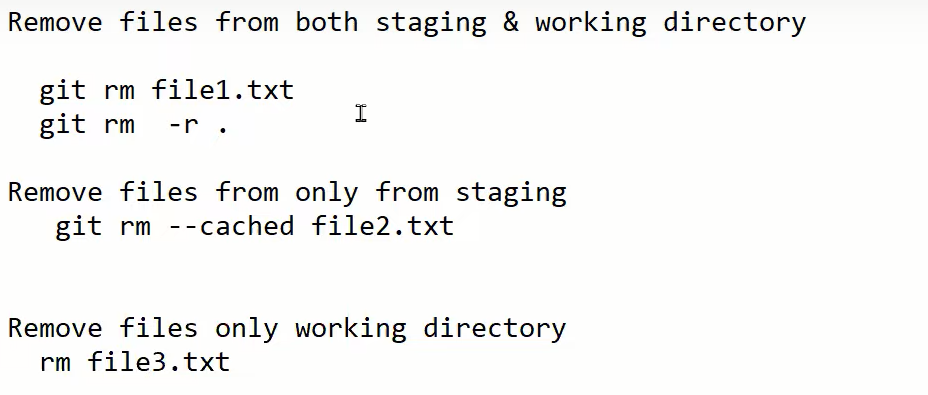
Above command Remove all the files from the working directory and staging area

1. Remove files only from staging area



1. Remove files only from working directory





**TECHNOLOGY STACK USED:**

**Git and GitHub**

**SOURCE CODE (OPTIONAL):**

**OBSERVATIONS / DISCUSSION OF RESULT:**

**CONCLUSION:**

**REFERENCES:**

**​Website References:​**