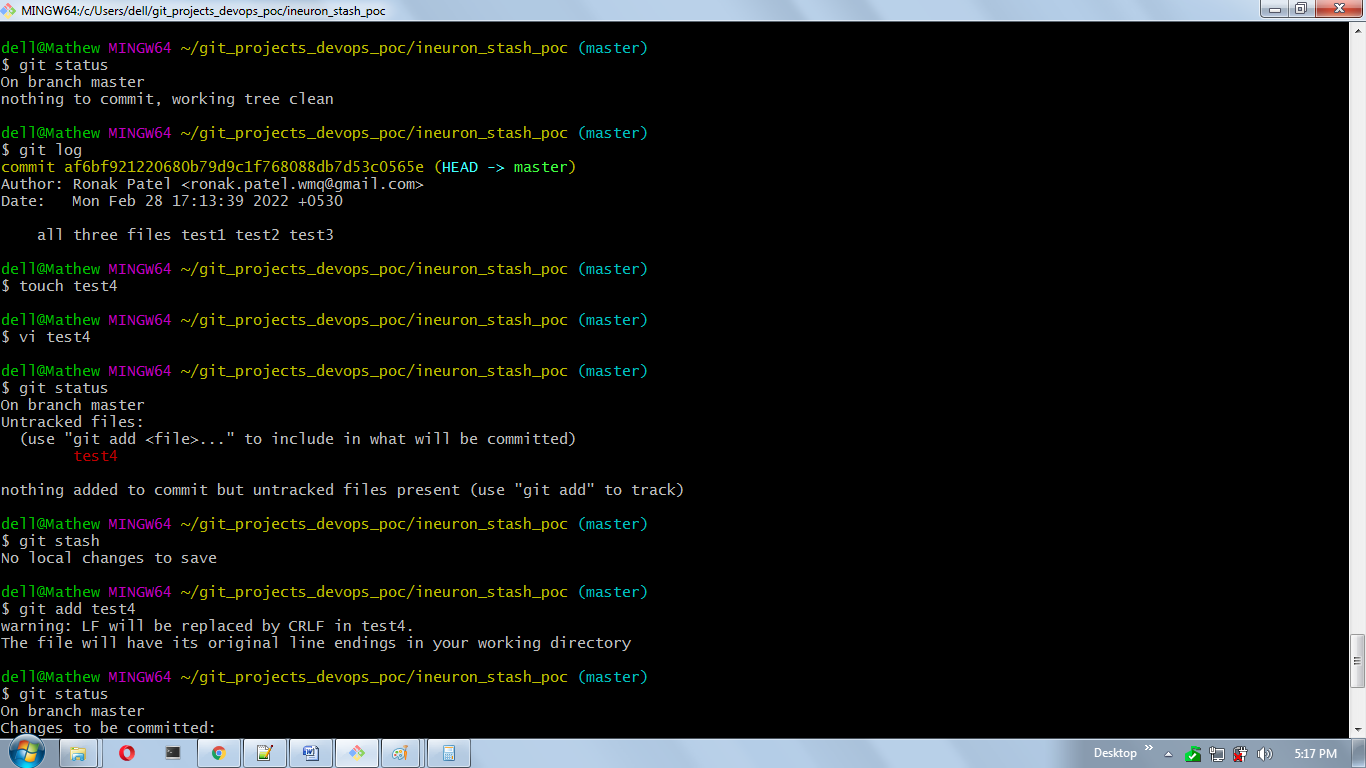
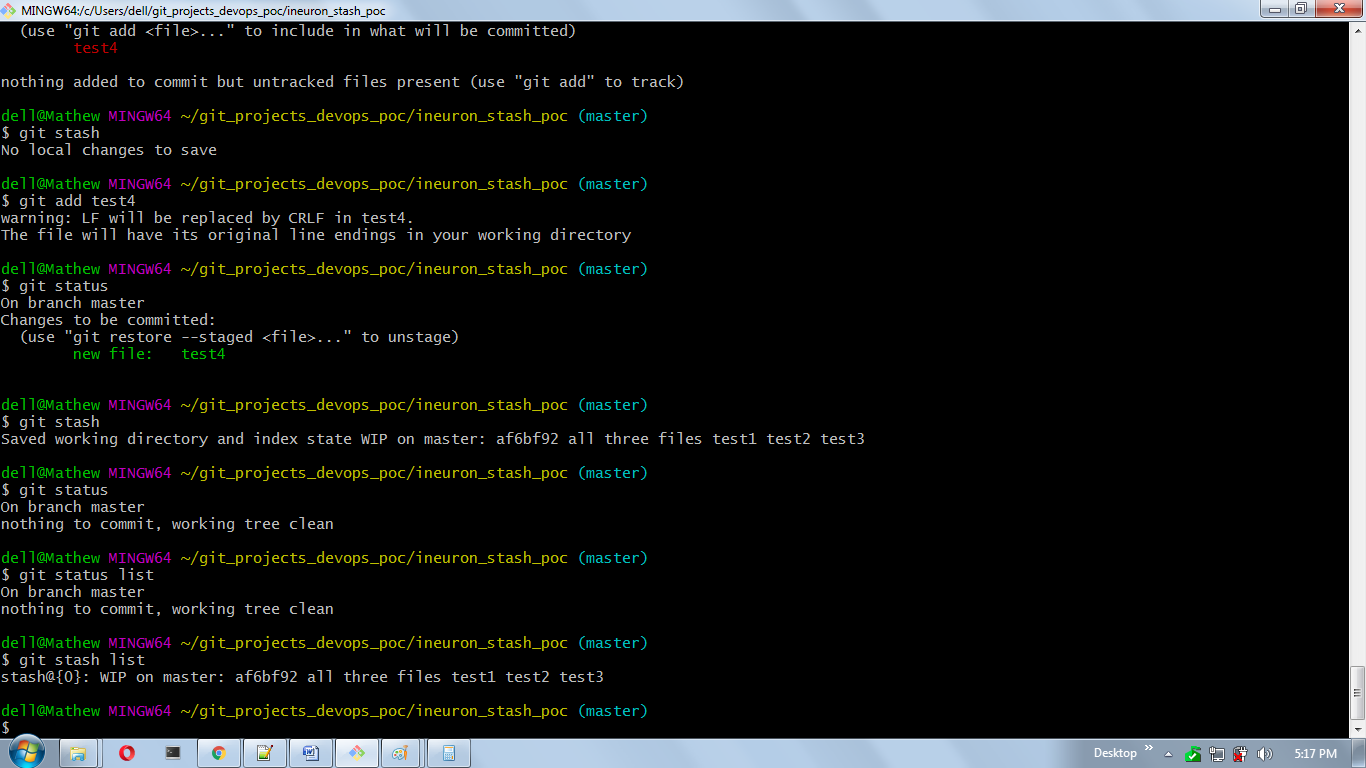
**GIT Assignment 1**

1. What is a git stash list?

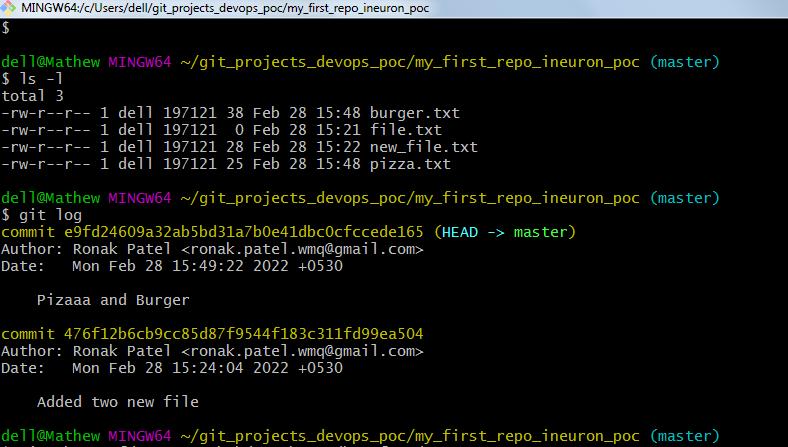
* Let us first understand git stash, it will allow us to save changes in work area without commit for temporary purpose and after then we can come back and retrieve and work and commit them later.
* Let's us take example you are working in two different area production and development, so currently you are in development are where you are making some changes locally but the source code is not ready to do commit and suddenly some production issue comes and you are ask to leave all your work and work on production issue on priority, so to save you work locally in development you need **NOT** to commit but can stash you work locally and then go to production area work/resolve the issue and then come back to development area and retrieve you work in development and once all change done can commit.
* **so this can happen multiple time so we save that locally multiple times. so we might have saved/stash them multiple times/items so to list all those stashed work we can use command** **git stash list**
* Let's see below POC as usual :)



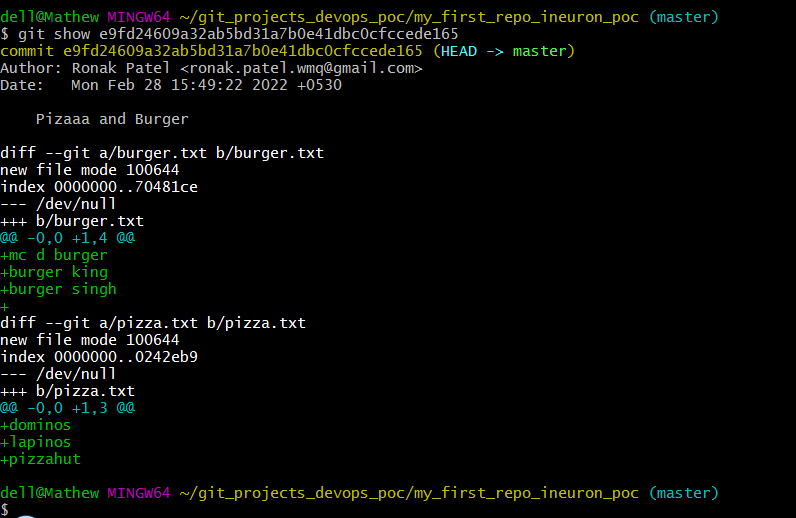


1. How do you get a list of all the files that have been updated in a given commit?

* we can use **git show commit\_id**, but let's go with some poc :)
* let's understand Commit first. Commit means that you want to save a snapshot of that particular state of you project so at later point you can review that changes with commit id.
* To see which are the commit done and its information use command **git log**

****

* Let us take an below example there are 4 files as in below poc done and i have edited those file with some content and added them to commit.(from working area->staging->commit)
* so now i can use **git log** to see all commit id and info
* now if we want to find the files and there info based on that commit id then use below command **git show commit\_id**
* Below is the example. Hope so this question is asking below only :)

****

1. What is a Git merge conflict?

* merge conflicts can occur in the process of integrating commits from a different source branch
* when a merge cannot be fully performed automatically then a merge conflict takes place.
* git will try to combine as much as it can but will leave special markers(eg. >>> and <<<)
* in other words conflict can arise when two different users are accessing same place or same lines of a code in source code. when try to merge git will give an error

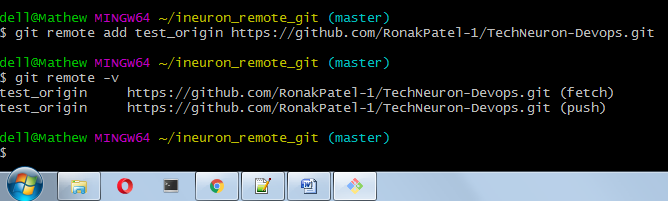
1. How do you distinguish between git fetch and git pull? How do you differentiate between Git Merge and Git Rebase?

* **git fetch** means you are doing a operation from remote to your local
* **git fetch**  only downloads new data from a remote repository but it doesn't integrate any of this new data into your working files.
* **git fetch** is safer as it will not screw your code or repository
* **git pull** means that you are downloading new data from a remote repository and also integrating that into your working files.
* with **git pull** there are more chances of getting merge conflict issue.
* To avoid that ensure you are doing a **git pull** should not have any uncommitted changes in your local repos
* **Git pull** is a **git fetch** request plus additional action **git merge**
* **Git Merge** means adding the commits together with another branch(or eg. Master branch)
* **Git Merge** and **Git Rebase** both are used to integrate changes from one branch into another.
* **rebasing** is a process of taking one branch and adding to tip of another, where tip is the last commit in that branch
* the main difference between **git merge** and **git rebase** is that while merge preserves history and rebase rewrites it.

1. What command uploads any GitHub repository to your computer using the git command?

Adding remotes

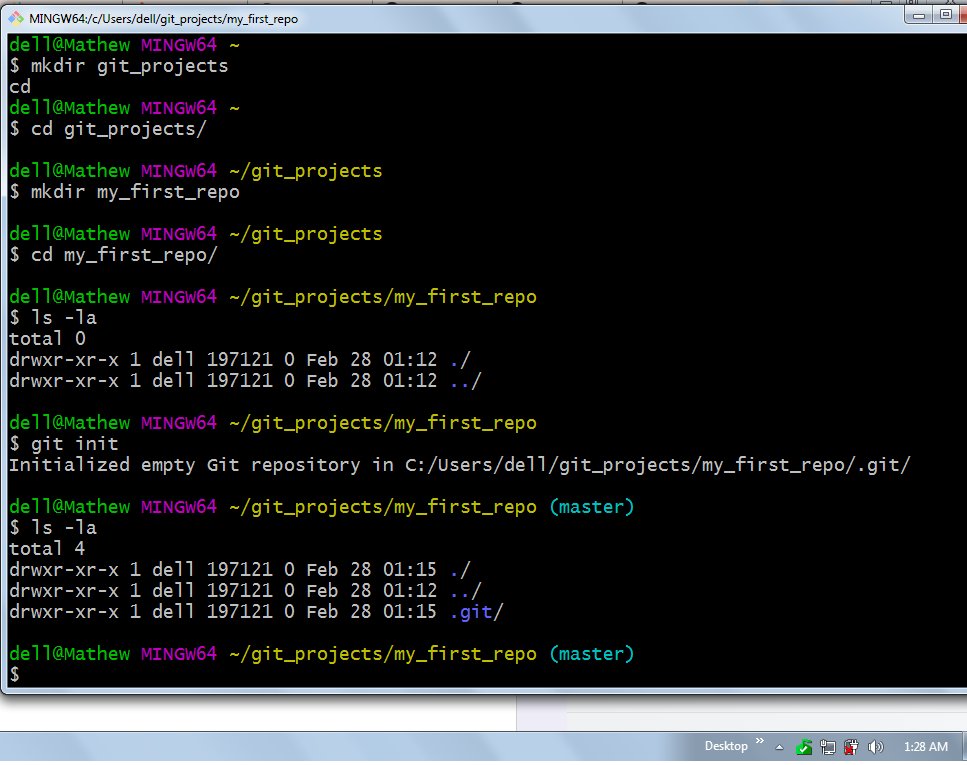
**git remote add origin\_name https\_link** or **git remote add origin\_name sshink**

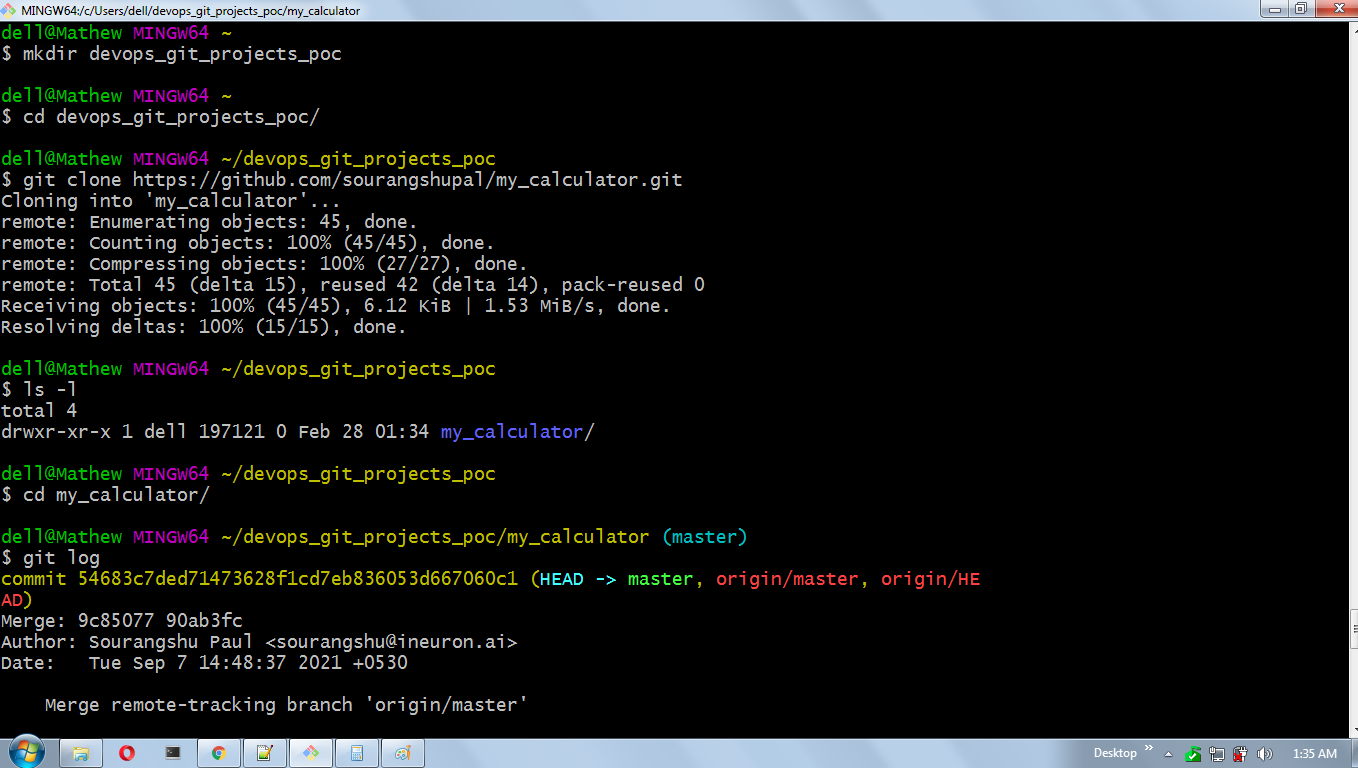
****

1. How do you write a commit message using the command? How do you push code in GitHub?

**git commit -m "type your message"**

1. How do you make a Git repository?

* In simple terms Repository is the centralized storage location for a software package/source code etc.
* In GIT there are **Local Repository** and **Remote Repository**
* there are two ways to create repository
* initializing a Git repository in **Local Machine** and **Cloning a Remote repository**
* Let see first how to create in Local Machine
* open git bash and open the directory, or create a directory and cd to that directory
* now if you check **ls -la** there won't be any files since we are creating for first time
* next we need to initialization use command : **git init**
* this means you are telling that make this a git local repository and from here onwards please keep a track of all changes made
* now again run command **ls -la** and this time you will see .git folder created
* this .git tells you this is my project/working directory
* also notice that after initialization it is show as master
* now lets check status by using command: **git status**
* now lets create a file say file.txt use command vi file.txt and enter some text, save and quit(escape :wq!)
* now lets check status again git status
* this time you will see No commits yet and untracked file
* this is how we can create a local repos
* Below is the screen shot from git bash which i did some poc :)
* 
* Lets see second way - **cloning a Remote repository**
* for eg. say a new team member has joined and he want to clone all existing project/source code then he has to do mirroring from centralised system for first time
* go to directory then use command **git clone url\_from\_centralized\_server**
* this command will clone or mirror the repository from above url
* after that you can also check **git log**  to check what all changes were done by whom/when. Below is screen shot from poc done :)



1. What is Git, and how does it work?

* Git is distributed version control system that allows developers and operations team to collaborate and keep track of changes made on a project/source code/files.
* It keeps track of all version or changes that have you have made on sometime and keep tracks of it.
* suppose you want to go back to say before 5 days what changes were done then with the help of GIT you can go back and check
* this is very helpful for eg. you are working in a very large team and you want to see which team member has done what changes in the source code then you can track that.
* Git is used to tracking changes in the source code with version control.
* it is a version control that record changes to a file/source code over time so that you can recall specific version later
* for eg. if you had screwed up the source code then you can easily recover.
* Git is DVCS(distributed version control system) in which it keeps a copy of all files from server to user's locally..or you can see mirroring
* A Git repository is a key value object store where all (commits, files, tags and filesystem tree nodes) are indexed by their SHA-1 hash value.
* Git works by taking snapshots of files
* there are 3 states
* **Working directory/modified** - place where you are making changes to file but it is not saved to keep a track of it
* **Staged Area** - it is pre-committing state, Stage the files and add snapshots to your staging area
* **commit** - data is safely stored in DB, Perform a commit that stores the snapshots permanently to your Git directory.