**EXPERIMENT NO.3**

**BY- Gatik Veer**

**Roll N.o. - 2110990493**

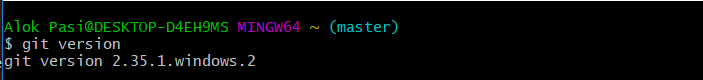
**Aim: Generating Logs:**

**GIT BASICS:-**

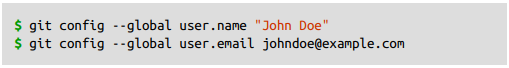
Firstly we need to form a connection between git bash and our github account before we can perform any other tasks on git.

To do this follow the following steps:

* Check your Git version using git version command



* Now let Git know where you are as this is important for version control systems,as each commit uses this info.You can specify Git configuration settings with the git config command.



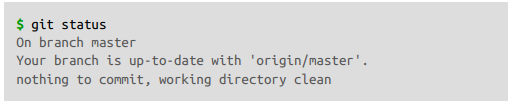
Now all your command will affect your github account.

**Getting started with commands:**

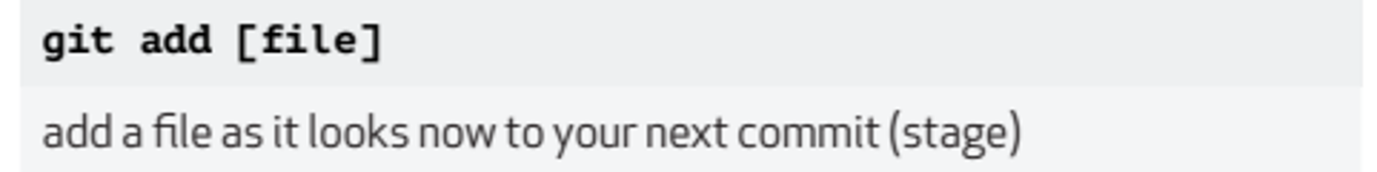
* ***git init* -** First of all create a local repository using Git.For this, you have to make a folder in your device, right click and select “Git Bash Here”. This opens the Git terminal. To create a new local repository, use the command ‘git init’ and it **creates a hidden folder .git that contains all of your necessary repository files – a Git repository skeleton.

****

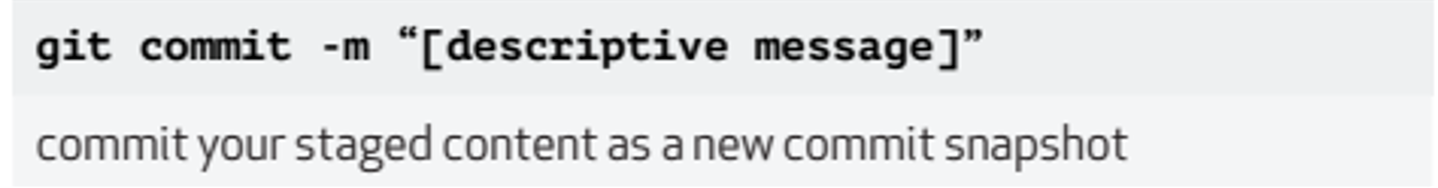
* ***pwd (Print Working Directory) -*** It prints the path of the working directory, starting from the root.
* ***cd (Change Directory) -*** It allows one to change their directory.
* ***mkdir –*** The mkdir stands for **'Make Directory'**. With the help of mkdir command, you can create a new directory wherever you want in your system.
* ***ls -*** The **ls** is the list command. It will show the full list or content of your directory.
* ***ls -ah -*** The hidden files start with . (dot) symbol and they are not visible in the regular directory. The (ls -a) command will enlist the whole list of the current directory including the hidden files.
* ***git status -*** The git status command is used to display the state of the repository and staging area. It allows us to see the tracked, untracked files and changes. This command will not show any commit records or information.

****

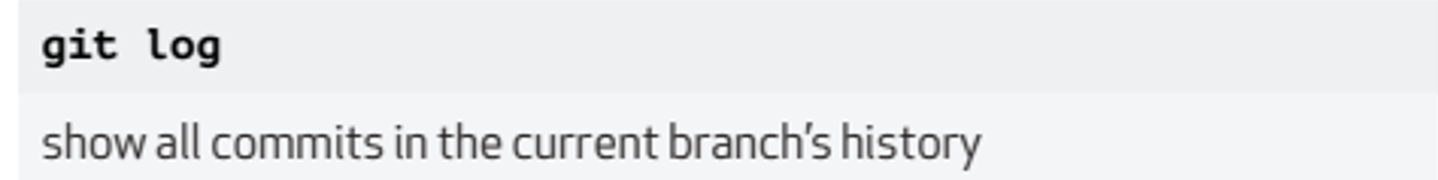
* ***git add -*** In review, git add is the first command in a chain of operations that directs Git to "save" a snapshot of the current project state, into the commit history. When used on its own, git add will promote pending changes from the working directory to the staging area.



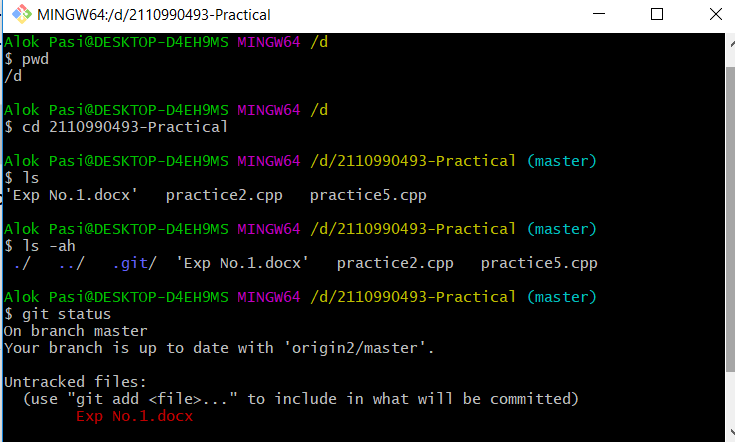
* ***git commit -*** A commit, or "revision", is an **individual change to a file (or set of files). It's like when you save a file, except with Git, every time you save it creates a unique ID (a.k.a. the "SHA" or "hash") that allows you to keep record of what changes were made when and by who.

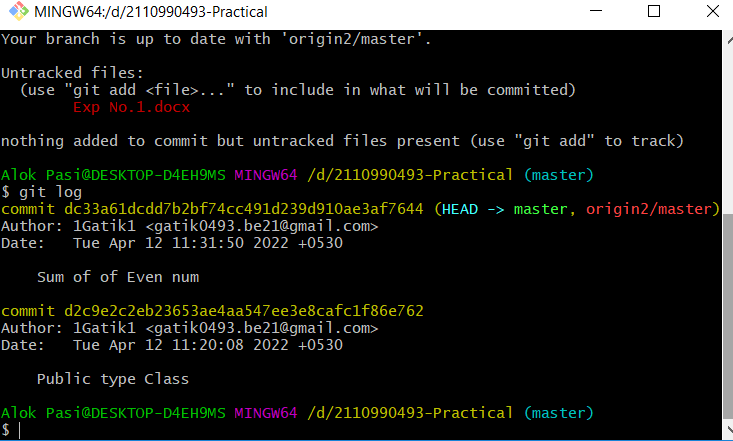


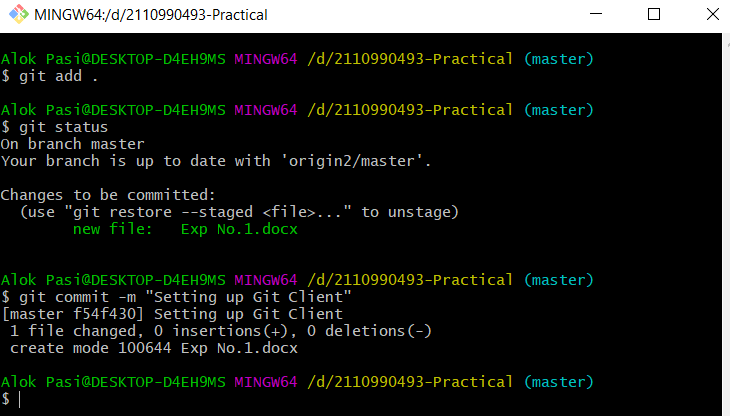
* ***git log -*** Git log command is one of the most usual commands of git. It is the most useful command for Git. Every time you need to check the history, you have to use the git log command. The basic git log command will display the most recent commits and the status of the head.

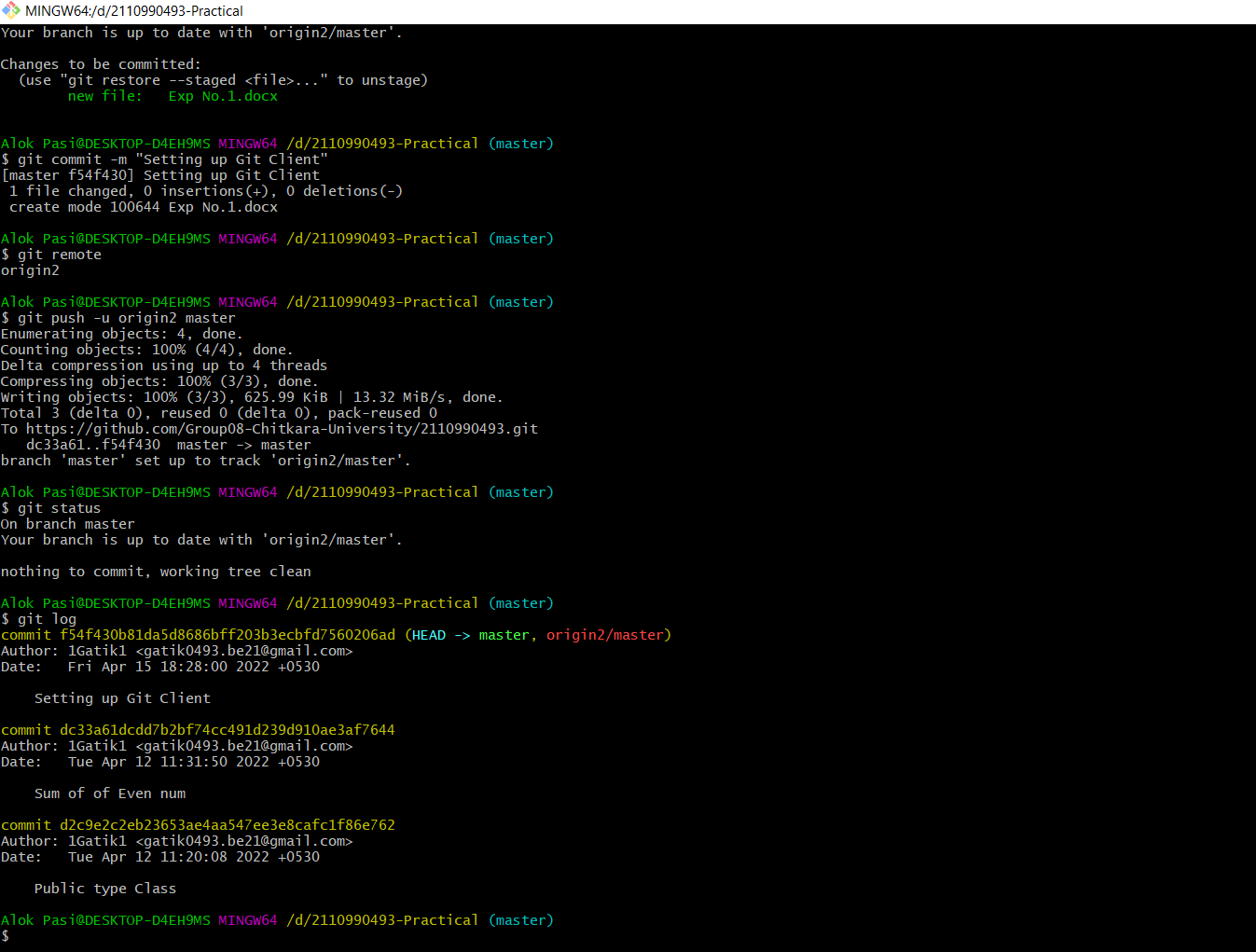
****

**Creating a new Log :**

******

****

******

****