**How to Use Git & Github**

1. git is a version control system

2. github is a kind of host that hosts all the repository of git

**Why to use Git ?**

1. Easy file recovery
2. We get to know if somebody else has introduced some code or not
3. Rollback to previous working state

**Github commands**

First add username and id to git using

Git config --global user.email “piyush@example.com”

Git config –global user.name “piyush”

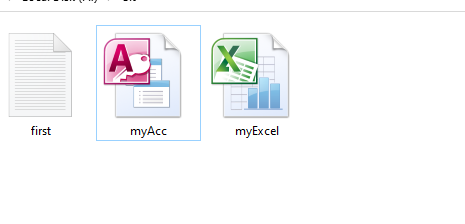
TO check wheather the username got added or not type

Git config –global user.name

Git config –global user.email

#4 Tracking our First Project

At first I make a folder of any name and make three files in It like this



Now we open git bash and write the command

**Git status**

this command gives us the status and if it gives fata error

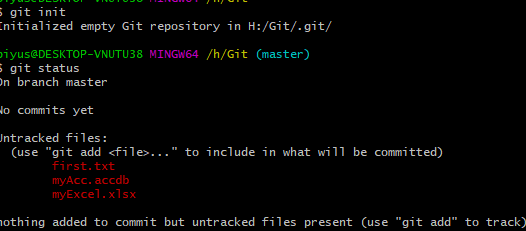
that means ki abhi hum git repository mein kaam nhi karrhe hai



To make it a git repository

We use the command **git init**

The again we use git status to check the status



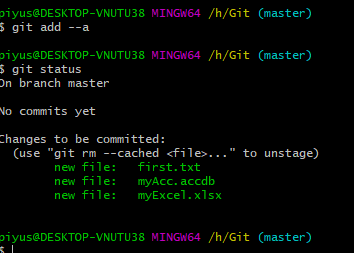
Here we got

1. Untracked files

Means the files we have in our folder Is not staged yet to commit

Now to add the files into staging area we use command

**Git add –a**

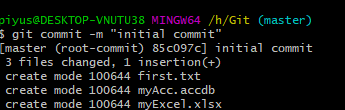


Now we can see all the three files are staged

Now to commit all these files we use command

**Git commit –m “this is a message”**

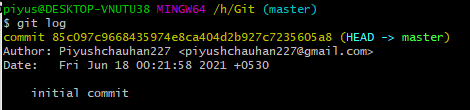
Here –m means we are writing the message in the same line if we don’t write –m here then the new window will open where we have to write the message



Here means all the files have been saved or commited succesfuly

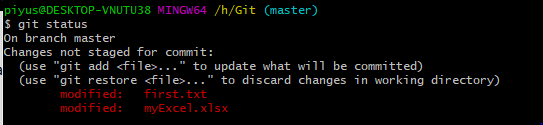
Now if we want to see all the ccommits we have done before the we write the code

**Git log**



It has all the info about the commits we have done

Now If I change two files but I want to commit only one file then



Here see two files are untracked now I want to commit only first.txt

The use the command

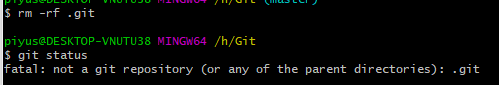
**Git add first.txt**

#5 cloning a git repository

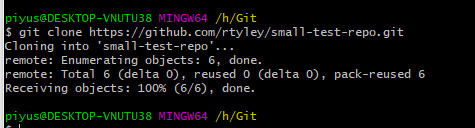
First look at one command is we want to delete a repository from .git

It’s a dangerous command

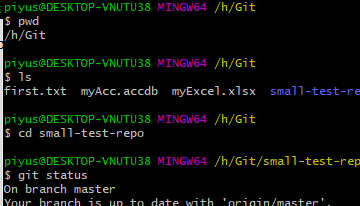
**Rm –rf .git**



Now for cloning



Some basic commands see



Pwd = present working directory

Cd= change directory

Ls = content listing

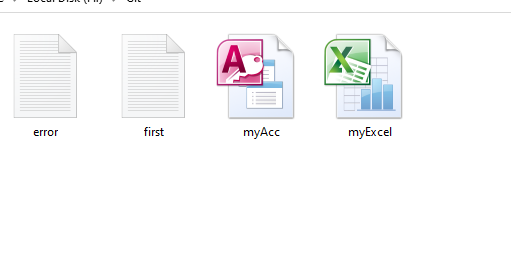
#7 Git ignore

First we make a new file by using command

**Touch filename**



It will create a new file in the folder



Now I want ki when I commit I don’t want ignrore file to be included

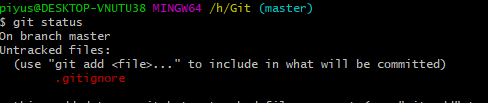
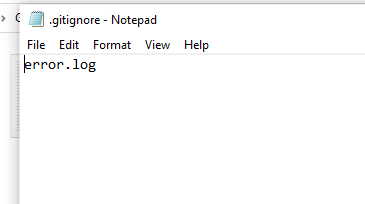
So for that I make a ignore file

Example using touch command

**Touch .gitignore**



Now whatever files we want to ignore we write it In gitignore like this



Now look heres only one file in untracked error .log got ignored

If we want to ignore one whole type of files like if I want ki saari .log vali file ignore hojae toi will write in gitignore



Now it will ignore all files that has .log extension

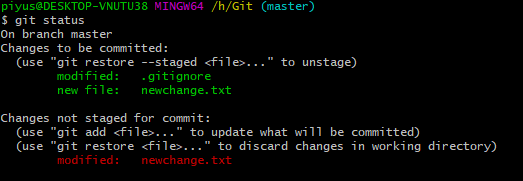
Showing Changes Between Commits/Staging Area & Working Directory

Now if I add new file and the see that file it is untracked

And is stage that file



Now newchange.txt is in staging area but what if before cmmiting I will do some changes to that file

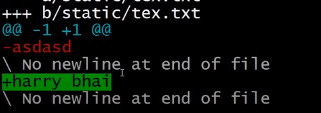


Now the same file is in first staged and second untracked

Now these one more command **git diff**

**Git diff compares staging area with working directory**

It shows us the difference between the staged file and untracked file



Like this now if I again stage that flie the older one will get replace

Skipping The Staging Area

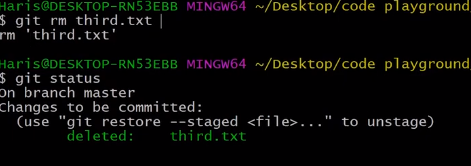
If we want to skip the staging step

Like we do with command **git add . or git add --a**

We can use command **git commit –a –m “this is direct commit”**

***NOTE***  this command will only skip the staging step for unmodified files if the files are untracked it will not work

Moving and Renaming Files In Git



It will delete that file and also stage that file but if we delete this file from folder then we have to manually stage the file again



For renaming

To remove or delete the whole git we use command

**Rm –rf .git**

Viewing & Changing Commits In Git

If we do **git log** then we it will show us all the commits happen

But if we want to see the difference on all these logs then use command

**Git log –p**

If we use **git log –p -2**

Then it will show us last two commits with differences

If we use **git log –stat**

It will shoe us only less info about the commits

If we use **git log –pretty=online**

It will show only one line change in commit

Unstaging & Unmodifying Files

If we somehow delete the content of the file but now we want it back to the previous commit use command

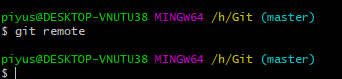
**Git checkout -- filename**

If we want to ignore all the changes in the working directory and want to go back to the previous commit

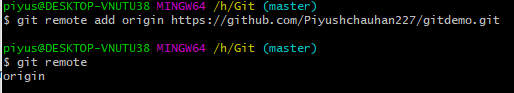
Use **git checkout -f**

Working with Remote Repositories

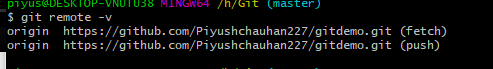
First we use **git remote**



To see wheather we have any remote repository or not If it will send us nothing that means we are not wotking with any remote repository



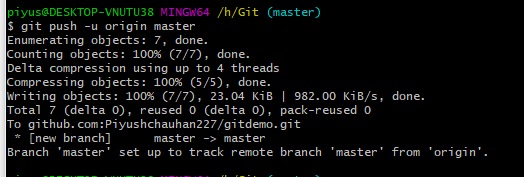
Now we use git remote add origin URL this code means working with remote repository whose url name is origin



Now we use the code git remote –v this code helps us to get the urls for push and pull

For pushin the code first make sure working tree is clean

Then use **git push -u master**



Creating & Switching Branches In Git

To make a new branch use command

**Git checkout –b branchName**



Now we can change anything in develop branch it will not efct the master branch until we commit

We can always go to master branch by using command

**Git checkout master**

Use command **git branch** to see al the branches name

For merging use **git merge branchName**

