\*\*Git/Github\*\*

What is Git/GitHub & Why do we need it?

**Git**:

So Git is a version control system. Let’s assume I have made an AI desktop assistant. Now I was working on version 1 which has 3 files, f1, f2, and data.csv.

After a few days I have changed the code and made version 3 which also has 3 files, but this time f1, f2 file doesn't exist, instead it has f3, f4 and data.csv. So now f1, f2 files are not around.

If in the future we get into a situation where we need those files there will be no option, but if we use a version control system like Git then we can go back and see those files and work with it.

**Pros of Version Control System:**

1. Easy File Recovery: You can easily recover files from the version control system.
2. Issue: If more than 1 person is working on a project and the project stops working after someone’s commit then you can easily check who’s commit it was and rollback a previously working state.

**History of Version Control Systems:**

**Local Version Control System:**So the first version control system made by programmers was a Local Version Control System, which used Database to keep track of files. But the major drawback was all the changes were saved in the computer. So if the computer gets corrupted then all of the data will be gone.

**Pros:**

* You can track files.
* You can Rollback.

**Cons:**

* If the local Computer/ Laptop gets corrupted then all of the data will be lost.

**Centralised Version Control System:**So after Local Version control System the programmers made Centralised Version Control System. In the Centralised Version Control System the changes were saved in a server, so there is no chance of losing all the data and the changes due to data corruption. The programmer needs to pull the file from the server, make his changes and then push that modified file to the server. Now there is a chance that if the server gets damaged then the files can be lost forever.

**Pros:**

* You can track files.
* You can Rollback to a specific version of the file.
* Changes were saved in a server instead of a local computer.

**Cons:**

* If somehow the server gets damaged then the files and changes will be gone.

**Distributed Version Control System:**

This is almost like the Centralised Version Control System. The only difference is when the programmer pulls the file he gets the complete changes of that project and each programmer has a complete backup of the server. So if the server gets damaged the programmers can fix it easily without losing any files.

**How was Git Started:**In 2002 the programmers who were working for development of Linux started using Bit keeper VCS as their version control system. But in 2005 Bitkeeper removed their free of charge status from them and asked to pay the Linux Development team a huge amount of money. After this**Linus Torvalds**, the creator of **Linux** started making his own version control system and then he made Git, a free Centralised Version Control System.  And everyone started using Git.

**Features of Git:**

* It stores the snapshot not the differences so the only file which is changed will be stored and disk space will be saved.
* Almost every operation is local. You can work on your local machine and then push those changes to the centralised repository.
* Git has integrity. Git generates SHA-1 Checksum for each and every change, so only those who have access can modify files.
* Git generally adds data. So, as you change the versions it stores them.

**GitHub:** GitHub is a hosting website which hosts Git repositories. Git is free to use but GitHub charges for managing those files. There are also alternative websites which host git repositories like Bitbucket, Gitlab etc. Among these GitHub is most popular.

What is the bash & terminal?

The terminal is the GUI window that you see on the screen. ...

It takes commands and shows output. The shell is the software that interprets and executes the various commands that we type in the terminal.

Bash is a particular shell.

step 1

**Create a floder & initilize the it by using user name & email ID.**

git config --global user.name "amolNagrale"/Amolnagrale

git config --global user.email "amolmnagrale1@gmail.com"/amolmnagrale1@gmail.com

if we are see which mail ID has initialize

git config --global user.name

gii config –global user.email

step 2

git clone - To get any existing project all code direct from server.

git init - to create the new repository

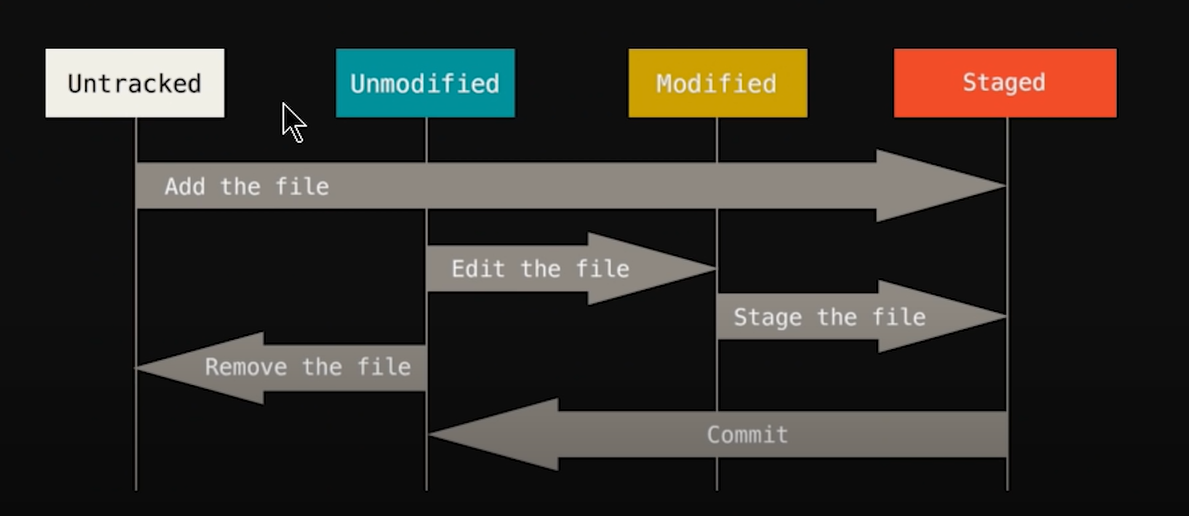
step 3

**ls – to show the list of file in directory.**

Goto inside the folder & create the new files.

Add files by using git add ./ git add floder name.

Commit – save the file went to unmodified state



Touch f3.txt – to add file in folder

Touch f4.txt – to add file in folder

Git add .

Git commit -m “file added”

Git status

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Do the some changes on existing files

Git add .

Git commit -m “Modified files”

If any file contents has deleted by mistake the how to restored it.

Git status

git checkout f3.txt (file name)

If someone do the un-necessory changes on your multiple file (1000 files), below commant has support to match the previews file & recover it.

git checkout -f

Impotant note – 1. Please try to use git status commant for the current status.

2. commit the file once your confidant the files.

**\*\* Clear – this command used to clear the terminal**

Git log- It is the history to track & managing the changed.git

Git log- to check all history of logs.

Git log -p -2(as per your requirement you have checkout your commit history..eg I have to check the history of last 2 commits)

Press q to exit git logs…..

\*Git diff – to compare the workng directory with staging area.

If we have do some changes in our file the run diff command then we have to show the comparision between them.

Git checkout -f/ git add -A :: to recovred the original files.

If we want to remove the file from dirctory.

Create the file f5 using touch f5.txt

Git add .

Git commit -m “add f5 file”

Ls - to show the all available file in present directory.

Git rm f5.txt ---- to remove the file from directory.

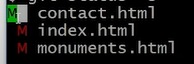
Git commit -m “remove f5 file”

Git status

How to findout the short git status

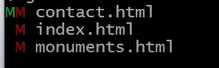
Git status – s ----- before that do the some changes in directory folders.

Git add contact.html



Green high lighted – staged area, Red highter show the currect tree.

If we are do the some changes in again contact.xtml



MM - file show both stages are in modified file (stage area current working tree.

Git checkout -f

Branches If we want to some changes master file, in thast condition, will go with branch

Main branch means master

Git branch feature1

Git branch ---- show the master branch

Git checkout feature1 – then you are in branch.

Git checkout master

Git merge feature1 – we have to merge the master & branch.

Git branch ---- to show all branches

**Create the new branch –**

Git checkout -b feature2 – to create new branch

Github – hosting site.

Remote repository – we have push local repository in remote.

First we have add the remote repository with local repository, using below command.

git remote add origin https://github.com/AmolNagrale/git\_01.git

git remote ---- showing origin remote

git remote -v --- to show the path fetch & push

git push master origin ----- for the first time we need to get permission from github server for push/fetch the repo, hence So we want to add the ssh key.

**SSH keys** enable the automation that makes modern cloud services and other computer-dependent services possible and cost-effective. They offer convenience and improved security when properly managed. Functionally **SSH keys** resemble passwords. They grant access and control who can access what.

$ ssh-keygen -t rsa -b 4096 -C "amolmnagrale1@gmail.com"

cat /c/users/user/.ssh/id\_rsa.pub --- copy address paste using **cat** command.

Then copy the command get push into origin master.

Copy key and past on SSH key in github

Fork -

Amol@DESKTOP-QSAAOEP MINGW64 ~/Desktop/repo\_02 (master)

$ git status

On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

Amol@DESKTOP-QSAAOEP MINGW64 ~/Desktop/repo\_02 (master)

$ ssh-keygen -t rsa -b 4096 -C "amolmnagrale1@gmail.com"

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/Amol/.ssh/id\_rsa): mykey

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in mykey

Your public key has been saved in mykey.pub

The key fingerprint is:

SHA256:wGMCvi9dBZVkYH5/7bKQM6PApDoM09rn+a1TSvY3gAU amolmnagrale1@gmail.com

The key's randomart image is:

+---[RSA 4096]----+

| . +++. |

| . . +Eo. |

| . . \*.o |

| . o =.. . |

| .. ooS . . . |

|o .o =+ o o . |

| \*. +oo+ .\* . . |

|. +o..oo..o= o |

| .+o.ooo. .. |

+----[SHA256]-----+

Amol@DESKTOP-QSAAOEP MINGW64 ~/Desktop/repo\_02 (master)

$ cat mykey.pub

ssh-rsa  amolmnagrale1@gmail.com