Git Tutorial :

Git is version control system it means it- Track and Manages changes in code and documents.
E.g: website ,desktop applin{we continuously make changes in web and application code}
Lets have simple example : priviously we was having windows 7 suppose it is of 2 Gb and now we have extended version that is windows 11 (10 GB) so we dont want prious windows 7 to be there in our pc so that case will require 12 GB so git help to mange the chages so that it will require only 10 Gb space.

Suppose we have 8GB space and An AI chatbot initially it contain F1,F2,data.csv as project.zip of 2GB and lets say it takes 2GB now we made some changes F1->F3 now we have saved it with projectn.zip again we have saved it with projectnm.zip in that case it will take 6 GB space so to avoid and manage changes in code we use git.

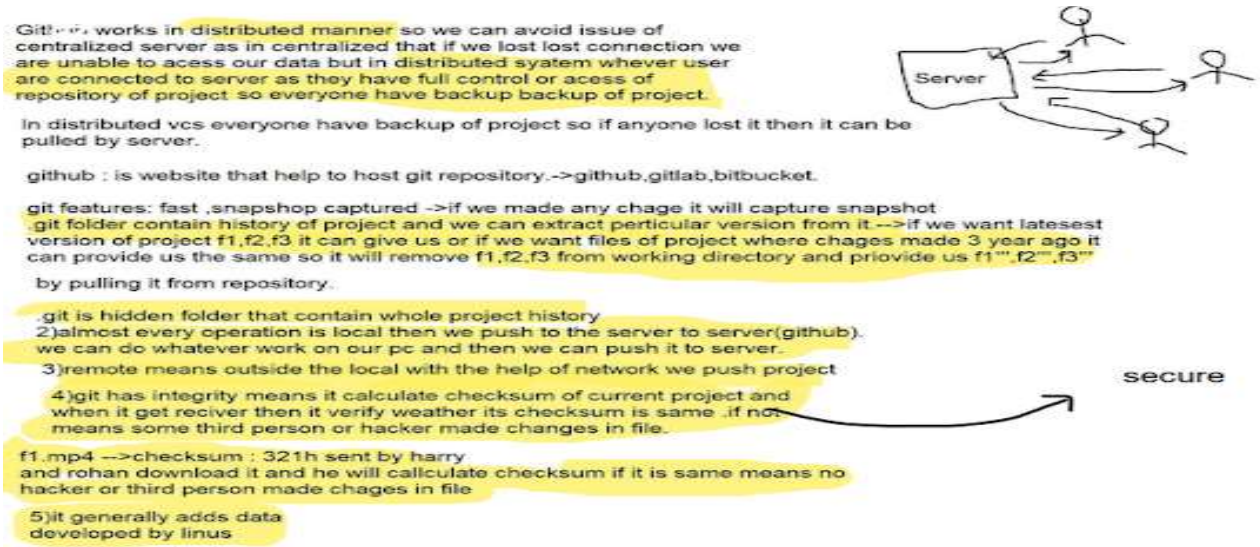
Advantages of Version Control System :

1)Easy File Recovery : suppose in project we have deleted two files and made new version and we want that files back then it is possible with the help of git.
2)Rollback to previous working stage : suppose someone made changes in working web commit and now it stop working so we can rollback to previous working stage.

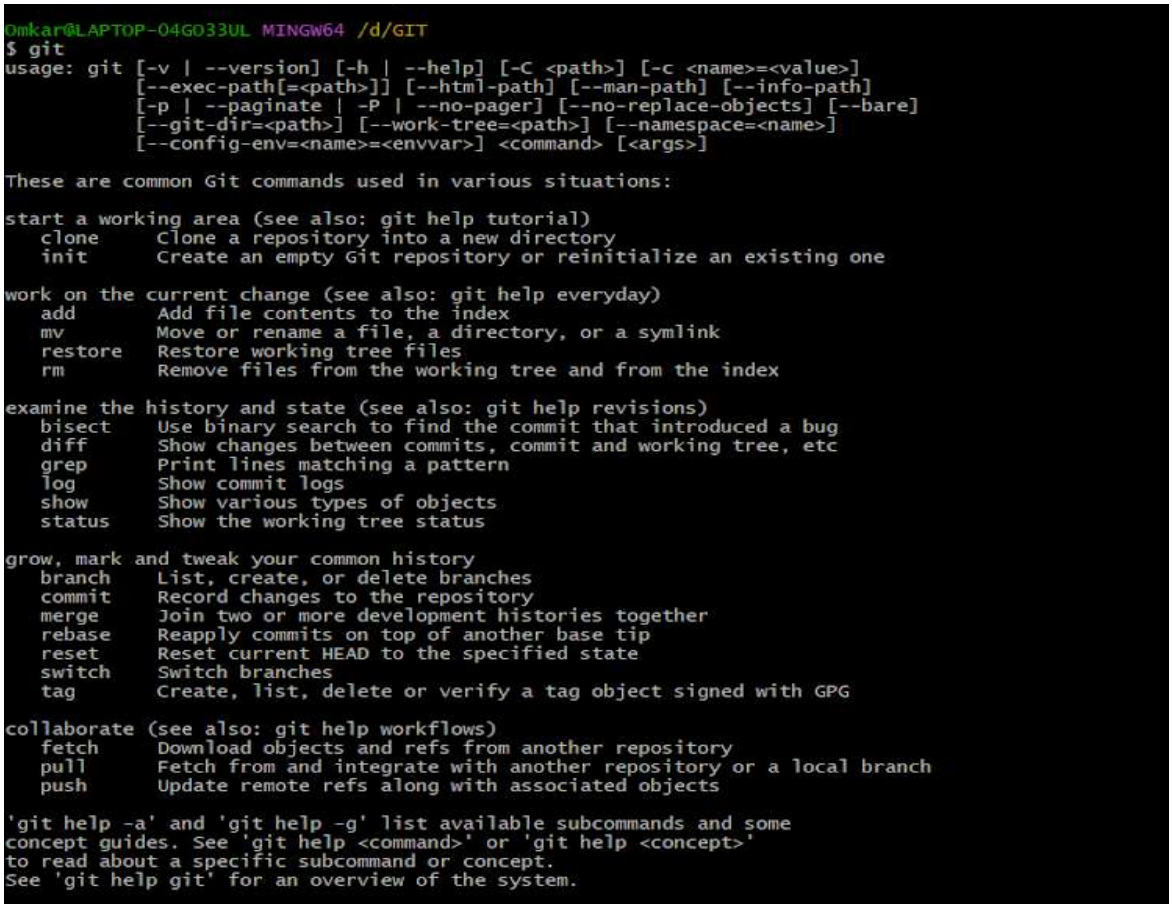
VCS History :

Local VCS : db was used to track pros : can track file and rollback cons: if u lose hd then everything lost.

Centralized VCS : To avoid pb of local vcs(hd) centralized vcs is made.developer decided to maintain server and started pull files and push to centralized server.



GIT Bash : its command line tool through which we can run git commands on specific directory .it is helpful when we switch to mac or linux and it talkies commands in linux form.
Cd command we can use to switch to the folder can use git commands like push ,pull commit and clone for that folder
Name and email so that user from global can get to know who has made changes and they can contact.



pwd :to get present working directory
/d/GIT
If omfolder is present inside GIT so we can write cd om
/d/GIT/om
If we write : /c
will go in c directory (bash always takes linux commands and its advantage is to go in any folder -this help to run commands for that specific directory we can make any number of commit,rollback for that directory)

```
Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT
$ git status
fatal: not a git repository (or any of the parent directories): .git

Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT
$ pwd
/d/GIT

Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT
$ cd om

Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT/om
$
```

```
Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT
$ git config --global user.name "Omkar"

Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT
$ git config --global user.email "omkarlokhande425@gmail.com"

Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=Omkar
user.email=omkarlokhande425@gmail.com

Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT
$ git config user.name
Omkar

Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT
$ git config user.email
omkarlokhande425@gmail.com

Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT
$
```

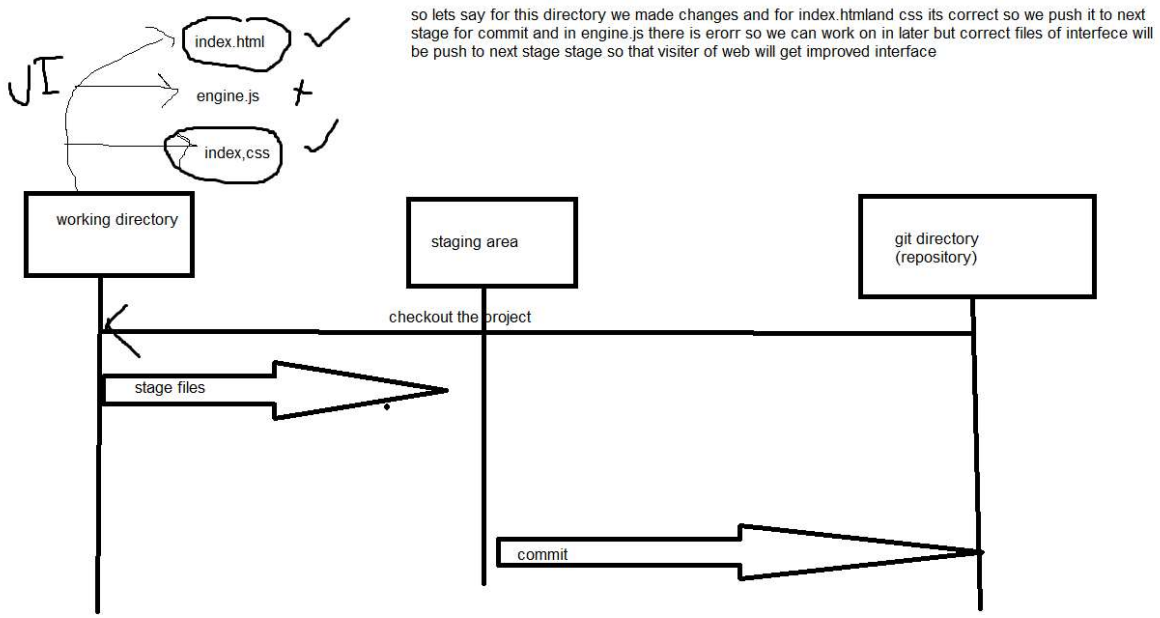
Git Workflow :

Suppose we have directory UI to improve where we have- index.html
-static folder /temp
-engine.js

Now we have made changes and created version v1 and again lets we are working on it and due to some error its not working properly so back to working version v1 and made commit with commit msg like working C1[version]

Suppose we want to improve UI for this we changed index file ,temp file but suppose we made changes in engine.js and it cause error but index,html changes are correct .so will wish

index.html to kept in new snapshot (version) and engine.js to be in old snapshot [we dont want to introduce in new commit] so for this three stage archi is introduced.



For linux user we can set editor using command:with the help of this editors we can open any file in editors and work on it.
git config –global code.editor emacs
git config –global code.editor vim

Master : main branch

Git Commands :

git status : used to check status of git repository .if .git is not present in folder then it show it is not git repository.

git init : we can made that folder git repository by using this command.[convert folder into git repository]

git add --a : all files are added to staging area in one time.

git commit -m “msg” :for commit the repository .means record changes .

git log :get list of commit done so far.

```
Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT
$ git status
fatal: not a git repository (or any of the parent directories): .git

Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT
$ git init
Initialized empty Git repository in D:/GIT/.git/

Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        db.accdb
        first.txt
        myxcel.xlsx

nothing added to commit but untracked files present (use "git add" to track)
```

```
Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT (master)
$ git commit -m "Initial Commit"
[master (root-commit) 5ef7601] Initial Commit
 3 files changed, 1 insertion(+)
 create mode 100644 db.accdb
 create mode 100644 first.txt
 create mode 100644 myxcel.xlsx

Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT (master)
$ git status
On branch master
nothing to commit, working tree clean

Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT (master)
$ git log
commit 5ef7601220b2e2ff71c3ee8e89a14fbda4e39057 (HEAD -> master)
Author: Omkar <omkarlokhande425@gmail.com>
Date:   Thu Jul 13 15:02:54 2023 +0530

    Initial Commit
```

Suppose we make change in first.txt file then it is tracked .

```
Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   first.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

We have db file we add some data to it it check log now it show both db and first.txt files are modified .

Now we want to commit only first.txt so will write command as -

```
Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT (master)
$ git add first.txt

Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   first.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   db.accdb
```

It shows that we allow to stage first.txt for commit.

```
Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT (master)
$ git commit -m "chnged first.txt and added better design"
[master 2455e48] chnged first.txt and added better design
 1 file changed, 2 insertions(+), 1 deletion(-)

Omkar@LAPTOP-04GO33UL MINGW64 /d/GIT (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   db.accdb

no changes added to commit (use "git add" and/or "git commit -a")
```

Db.accdb will go in next commmit

```
Omkar@LAPTOP-O4GO33UL MINGW64 /d/GIT (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   db.accdb

no changes added to commit (use "git add" and/or "git commit -a")

Omkar@LAPTOP-O4GO33UL MINGW64 /d/GIT (master)
$ git add db.accdb

Omkar@LAPTOP-O4GO33UL MINGW64 /d/GIT (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   db.accdb

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        calc/


Omkar@LAPTOP-O4GO33UL MINGW64 /d/GIT (master)
$ git commit -m "second commit"
[master ffbcc21] second commit
 1 file changed, 0 insertions(+), 0 deletions(-)

Omkar@LAPTOP-O4GO33UL MINGW64 /d/GIT (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        calc/

nothing added to commit but untracked files present (use "git add" to track)

Omkar@LAPTOP-O4GO33UL MINGW64 /d/GIT (master)
$ git log
commit ffbcc216c1c5252800c3d8d7826cfd6cc3894274 (HEAD -> master)
Author: Omkar <omkarlokhande425@gmail.com>
Date:   Thu Jul 13 15:34:20 2023 +0530

    second commit

commit 2455e48e8f12914365e76fe1325e7ce48f046df4
Author: Omkar <omkarlokhande425@gmail.com>
Date:   Thu Jul 13 15:25:47 2023 +0530

    chnged first.txt and added better design

commit 5ef7601220b2e2ff71c3ee8e89a14fbda4e39057
Author: Omkar <omkarlokhande425@gmail.com>
Date:   Thu Jul 13 15:02:54 2023 +0530

    Initial Commit
```