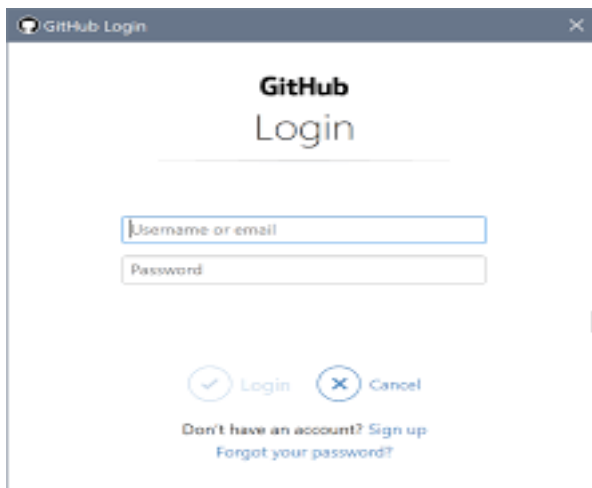




Git And GitHub Session No.04

Summary 26-02-2022

- **GitHub** - GitHub is a CVCS(centralized version control system) tool it is a centralized website for developers and programmers to collaboratively work on a single project.
- To connect our GitHub with your local git Repository 📁
 - **#git config --global user.name "<your GitHub user name>"**
 - **#git config --global user.email "<your "email id" Using which you have created the GitHub account>"**
 - **#git config --global -l** -> It will show us your config details
 - Now try to pull any repository, Git bash will open a popup, where you can write your "GitHub account" password, And that's it, You are authenticated.



```
Vimal Daga@DESKTOP-3E1AGGT MINGW64 ~/gittraining2022/firstproj (master)
$ git config --global user.name "Vimal Daga"

Vimal Daga@DESKTOP-3E1AGGT MINGW64 ~/gittraining2022/firstproj (master)
$ git config --global user.email "vdaga@lwindia.com"

Vimal Daga@DESKTOP-3E1AGGT MINGW64 ~/gittraining2022/firstproj (master)
$ git config --global -l
user.email=vdaga@lwindia.com
user.name=Vimal Daga
core.longpaths=true
merge.tool=p4merge
mergetool.p4merge.path=C:/Program Files/Perforce/p4merge.exe
difftool.prompt=false
diff.tool=p4merge
difftool.p4merge.path=C:/Program Files/Perforce/p4merge.exe
mergetool.prompt=false

Vimal Daga@DESKTOP-3E1AGGT MINGW64 ~/gittraining2022/firstproj (master)
$ |
```

- **.gitignore** -> Create a file named **“.gitignore”** Here we can mention all the files and folders that we don't want “git” to track in my repository.

```
AzureAD+SudhanshuPandey@DESKTOP-0NU1RKK MINGW64 /c/my_projects
$ cat .gitignore
a.mp4
b.mp4
c.gana
```

- **#rm file1.txt** -> To remove a file from the working area.
- **Reset** - We use the git reset command if we have to go back to some particular commit version.
 - Hard reset
 - Soft reset
 - Mixed reset
- Now here in the below repository, we can see we have 4 commits, and currently, Git is pointing to the latest commit(i.e HEAD commit)

```
MINGW64:/c/my_projects/gitsummary2

AzureAD+SudhanshuPandey@DESKTOP-0NU1RKK MINGW64 /c/my_projects/gitsummary2 (master)
$ git log --oneline
f1a21da (HEAD -> master) 4
6c543bc 3
d81a667 2
79080ba 1
```

#git reset 6c543bc file1.txt - Now the Staging area will be pointing to this “6c53bc” commit (behind the scene it is using mixed reset)

```
AzureAD+SudhanshuPandey@DESKTOP-0NU1RKK MINGW64 /c/my_projects/gitsummary2 (master)
$ git reset 6c543bc file1.txt
Unstaged changes after reset:
M   file1.txt
```

But If we try to read our “file1.txt” " it still shows the data we have in the last commit

```
AzureAD+SudhanshuPandey@DESKTOP-0NU1RKK MINGW64 /c/my_projects/gitsummary2 (master)
$ cat file1.txt
1
2
3
4
```

* Now, run **#git checkout file1.txt** -> And now if we try to read our file, It will show us whatever data we had in this commit(“6c53bc”)

```
AzureAD+SudhanshuPandey@DESKTOP-0NU1RKK MINGW64 /c/my_projects/gitsummary2 (master)
$ git checkout file1.txt
Updated 1 path from the index

AzureAD+SudhanshuPandey@DESKTOP-0NU1RKK MINGW64 /c/my_projects/gitsummary2 (master)
$ cat file1.txt
1
2
3
```

- Hence, with the “reset” command, we can get back to any particular version of our code.
- **#git log –help** -> It will open the help document of the “git log” Command.

- **#git config --global alias.myhistory "log --oneline"** -> It will give an alias name “myhistory” to the “git log –online” command,

```
AzureAD+SudhanshuPandey@DESKTOP-0NU1RKK MINGW64 /c/my_projects/gitsummary2 (master)
$ git config --global alias.myhistory "log --oneline"

AzureAD+SudhanshuPandey@DESKTOP-0NU1RKK MINGW64 /c/my_projects/gitsummary2 (master)
$ git myhistory
f1a21da (HEAD -> master) 4
6c543bc 3
d81a667 2
79080ba 1
```

- Hence now instead of running “#git log –online” we can directly run “#git myhistory”, we will get the same output.
- **#git config –global -l** -> Here we can see our alias, which we have just created.

```
AzureAD+SudhanshuPandey@DESKTOP-0NU1RKK MINGW64 /c/my_projects/gitsummary2 (master)
$ git config --global -l
user.name=sudhanshunilever
gui.recentrepo=C:/UnileverProjects/AA-Models-Project
alias.myhistory=log --oneline
```

- **#git commit. -m “my commit”** -> It will commit all the files present in the staging area in one go.
- To will delete the “file1.txt” file from all the areas i.e(commit area, working area, and staging area) –
#git rm -r file1.txt
#git add -u
#git add -A
#git commit -m “removed marker” .