Experiment 01 : WINDOWS 20/2/25

01] Explore git/github commands [⇒ Step1 :]

ex: 1) git config
2) git init
3) git add.

1) to check if git installed
> git --version.

2] Download git version control
→ web browser
→ download git
for windows 64 bit.
version
latest stable version.

3] Also download IDE
ex: VS code
→ download for windows.

---

git status →

Record → git/github intro & it's Advantages :
All git commands in order ⇒ explanation + Screenshots (min 4 commands).

→ Commands :

S-1) VS code → terminal, create a directory.
> mkdir gitfile. ↓ open a folder.
> cd / change directory into that folder

S2) Set up user name / user email
> git config --global user.name "Monish."
> git config --global user.email "monish@gmail.com"

S3) Create a file in VS code & folder
add some file (ex text, a code, etc) & (cd)
→ terminal → git init // to initialize repo (local).

s4) now after changes in that folder,
to engage →
> git add <filename>

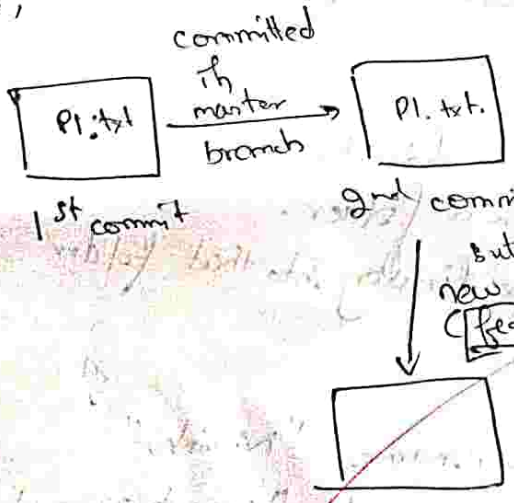
s5) then to commit,
> git commit -m "<any comments>"

s7) (change again (some changes) in code again & save
then check (> git status)
→ since we have modified a txt file,
it shows modified file → gitfile/txtfile.
// so need to add first then commit.
when added, turns to green → means saved, yet to commit.

s8) git commit -m "second commit"
→ 1 file changed 1 insertion (+) ✓

s9) Git status
→ on branch, nothing to commit.
// committed successfully.

now,



s10] git branch <branchname> → to create branch
s11] git checkout <-//-> → to navigate to that branch
s12] //to verify.
→ git status → on branch branch1
nothing to commit.

S13] create changes, add & commit in new branch (branch 1).

S14] to navigate to master branch. (default),
> git checkout master
// we can observe there is no updated txt file of feature branch (not changed in master branch).
∴ to merge (master) & feature (if no bugs).

S15] > git ~~merge~~ fetch
S16) > git merge <feature branch> branch 1

} only works in master branch ∴ navigate to it first.

// integration done
code updated from feature branches to master branch.

S17] new branch2
→ git branch branch2 (df).
→ git checkout -b branch2 → directly navigates & creates

do some changes here, save, add, commit.

S18) to show all commits on that file
> git log.
:q // to exit log.

S19] to delete a branch (in master)
> git branch -d branch2
<branchname>

} do on master branch after (git merge branch 2) ✓ to avoid bugs

> git checkout branch2
✗ no file(s).

S20] to go to previous version of commit,
copy the hashcode of that commit in git log,
or git log --oneline
copy hash code

→ go to master,
> git checkout <hash-code of that commit.

// same with github → new repo → (cd), git init.

→ save add, commit in vs code

→ git push origin main (updates in git hub).

→ git pull // then any changes if required again

→ git push origin main

to connect local - remote → git remote add origin URL of repo

after changes in local repo

git push origin _____

branch name to push to remote repo

20/2/25

→ to fetch updated from remote

git fetch origin _____ // to

b-name

→ then to update the code

git pull origin _____

b-name

b-name

→ usually main

default in github

→ or master → default in vs code (local git branch)