1. Version Control: keeps track of every modification to the code And If a mistake is made, developers can turn back the clock.

Git: modern version control system/ open source project.=================🡺

gitk (for UI)

1. **Config**: git config - - global user.name [name] // git config - - global user.email [email] // git config - - global color.ui auto (true)// git config format.pretty oneline // git add -I (use interactive adding)
2. git command help (command instructions) // git remote (list remote repos)
3. git init [Project No.] d) git clone [url]

e) git **commit** -m [message] // git commit . (Anything that has been staged with git add will be commited)

f) **Synchronize changes**: git fetch [bookmark] (drop changes, download all history from repository link)// git pull (download bookmark history & incorporate changes).

git checkout -b [branch-name] (OR -d)// git add file1.md OR git add \* // git pull // git clone [url]

git **merge** [bookmark]/[branch] (combine link’s branch into current local branch) //

git **push** –set-upstream origin [branch-name] (Upload all local branch commit to git) // git push origin master

git **remote add** origin <https://github.com/YOUR-Username/YOUR-Repo.git> (to connect repo to remote server)

g) **Make Changes**: git add [file\_name] // git add \* // git status // git diff // git diff - - staged // git diff origin file// git diff [first\_branch] [second\_branch] // git reset [file\_name] [unstage file & preserve its contents] // git mv [file-original] [file-renamed] (rename filename)

h) **Branch Changes**: git checkout [branch-name] (switch to branch) // git branch (list all branches) // git branches [branch-name] (create new branch) // git merge branch (combine branch to current branch) // git branch -d [branch-name] (delete branch)

i) **Refactor filename**: git rm [file] (delete file from working directory &stage deletion) // git rm –cached [file] (remove from git but preserve locally)

j) **Suppress tracking** : in .gitignore file (ignore files list matching patterns) // git ls-files - - other - - ignored - - exclude-standard (list all ignored files)

k) **Save Fragments locally**: git stash (temporarily store all modified tracked files) // git stash pop (restore back stashed files) // git stash list (list all stashed changesets) // git stash drop (discards recently stashed changesets)

L) **Review history**: git log (current branch history) // git log –follow [file\_name] // git show [commit] (content changes of specified commit) // git blame [file] (shown file annotated with line modifications)

m) **Redo Commits**: git reset [commit] (undo all commit after [commit])// git reset - -hard [commit] (discard all history & change )

git checkout [file\_Name] (restore file from current branch). // git clean (clean unknown files from working tree)

n) **Tag**: git tag name [revision] (create tag for a given revision) // git tag -l [pattern] (list tags matching pattern)

File Status: M : modified // C: copy-edit // R: rename-edit // A: added // D:deleted // U:unmerged