Setup

* Git**✅**
* Git config --global user.name “Susmitha789257”**✅**
* Git config --global user.email [“atigaddasusmitha789@gmail.com”](mailto:\“atigaddasusmitha789@gmail.com\”)**✅**
* Git config --global --add safe.directory E:/

Inside New folder

* Git clone <https://github.com/Susmitha789257/timesheet.git>**✅**
* **We cant modify github file not available in local pull it after delete in local push after deleted in git.**
* **Incase deleted in local not pushed but I need again that test file but here pull again not works we need use restore.**
* Git restore test.txt**✅**
* Git log (to come out “q" / github - commits)
* git log --oneline**✅**
* Git blame test.txt**✅**
* Git pull**✅**
* Cd foldername**✅**
* New file creation test2**✅**
* Git status**✅**
* Git add test2.txt**✅**
* Git status**✅**
* Git commit -m “test2.txt added”**✅**
* Git push origin main**✅**

Delete test3.txt

* Git status**✅**
* Git add test3.txt**✅**
* Git commit -m “deleted”**✅**
* Git push origin main**✅**

Git to local

* Git pull**✅**

Local to git

* Cd ..**✅**
* Cd newfoldername**✅**
* Git init**✅**
* Git add .**✅**
* Git status**✅**
* Git commit -m “created index.html and style.css”**✅**
* Git remote add origin https://github.com/Susmitha789257/JobPortal.git**✅**
* Git branch(\*master)**✅**
* Git push origin master**✅**
* Git branch -m main (modify)**✅**
* Git branch**✅**
* Git push origin main**✅**

Branch

* Git branch -a
* Git pull
* Git checkout branchname
* Git checkout main
* Git diff branch1
* Git merge branchname
* Git push origin main

Pull request

* Clear (in vscode)
* Git branch newbranch (local creation branch)
* Q for typeing if stuck
* Git checkout newbranch
* Git push origin newbranch
* Inside git create pull request

Project

* Git pull origin main

Chat-gpt

* mkdir repo
* Git init
* Git add .
* git commit -m "Initial commit"
* https://github.com/YOUR\_USERNAME/my-repo.git

**Learn with Mithran**

* Cloud and devops important to apply many positions.
* Source control management
* Plubic cloud -aws,azure,gloogle…
* Distributed version control system
* Git --version
* Git status -s
* Git reset abc.txt (to untrack)
* Git reset
* Git add abc.txt myfile.txt
* Git commit -m “this is my first commit” (-m message)
* Git log
* Git config --global user.name “Susmitha789257”
* Git config --global user.email [“atigaddasusmitha789@gmail.com”](mailto:\“atigaddasusmitha789@gmail.com\”)
* Git log --oneline
* Git remote add origin <https://github.com/Susmitha789257/abc.git>
* Git push -u origin master (set upstream)
* Git branch
* Git branch develop
* Git checkout develop
* Git commit -a -m “hi”
* **GIt VISUALIZING (visualizing-git) --> <https://git-school.github.io/visualizing-git/>**
* Git checkout -b dev --> branch create jump branch --> create & switch
* git switch -c dev --> Equivalent to checkout -b
* Git rebase X --> cut copy paste -->from common point to x end point.
* git cherry-pick commit\_id
* Git commit -am “message”
* Git merge dev
* Git config --global core.editor vim
* Git config --global core.compression 2
* Git config --global diff.tool vimdiff
* Git mergetool (document goggle)
* Git fetch
* Git pull (fetch + merge)
* Git clone
* Git reset HEAD~2
* **Git reset --hard or --soft or --mixed(TASK)**
* Git revert HEAD~4 (current branch copy)
* Git checkout -b Y commit\_ID
* Git branch -d branch\_name
* Git reflog
* **git checkout commit\_ID**
* Git tag v1.0.0
* Git stash --> add to bin
* Git stash list
* Git stash pop --> recover from bin

Advanced:-

* .git folder
* Git init --bare devproject.git
* Git clone diarictory location using ip address.
* Touch .gitignore(hidden) --> 3.txt,5.txt
* Git commit --amend --reset-author
* Git commit --global --edit
* Git diff first\_commit second\_commit
* Git branching model ???
* Git merge --sqaush ???
* **Completed.**