*Git Guide - Basics*

1. *Git Setup: open Command Prompt (CMD)*

***1.1) Check git version:***

-> git --version

***1.2) Settings:***

***a) Name***

-> git config --global user.name *“Muqeeb Ansari”*

***b) Email***

-> git config --global user.name[*mailmuqeeb@gmail.com*](mailto:mailmuqeeb@gmail.com)

***c) Default Editor (Eg: VS Code)***

-> git config --global core.editor “code --wait”

-> git config --global -e

--this will open default editor file to edit all our settings.

***d) Line Ending***

-> git config --global core.autocrlf true

--above line will be “true” for windows, and “input” for mac or linux.

***Note: setting can be done on 3 different LEVEL’s as below***

1. --system***: Setting apply to All Users of the current computer.***
2. --global***: Setting apply to All Repositories of current user.***
3. --local***: Setting apply to The Current Repository***
4. *Initializing a Repository:*

***2.1) Go to the required directory:***

-> git init

-> ls -a …(this will list all files of directory including hidden files)

-> rm -rf .git …(this will undo git init OR remove .git)

1. *Staging Files:*

***3.1) Staging area Commands:***

-> git status

-> git add *<file\_name>* ...(adding multiple files )

-> git add \*.txt ...(adding all files with ,txt extensions)

-> git add . ...(adding all files in directory)

-> git rm --cached *<file\_name>* ...(this will remove files from staging area)

1. *Committing Changes:*

***4.1) Committing Commands:***

-> git commit -m “this is initial commit message.”

***Note:***

***a) in above command ‘-m’ flag is used for message.***

***b) suppose we commit without -m and message then it will open default editor (Eg: VS Code) where we can type our message.***

***c) message should be meaningful and understandable to other coders.***

***4.2) Listing all the commits:***

-> git log ***…this command will list all the commits***

-> git log -p -1 ***…this command will list Last 1 commit (press ‘q’ key to exit editors mode)***

-> git log -p -2 ***…this command will list last 2 commits (press ‘q’ key to exit editors mode)***

1. *Pushing Code to Remote Server:*

***5.1) First add remote Repository:***

-> git remote add ***<remoteRepo\_name> <URL\_remoteRepo>***

----Eg: -> git remote add ***meanremote*** [***https://github.com/mmahub/myMEANstack.git***](https://github.com/mmahub/myMEANstack.git)

***5.1) Push current directory to remote server:***

-> git push -u ***<remote\_repo\_name> <current\_committed\_directory>***

--Eg: -> git push -u ***meanremote master***

1. *Cloning Codebase from Remote Server:*

***6.1) Go to the desired location using CMD where you want to clone the remote code & perform below commands:***

-> git clone *<remoteRepo\_URL>*

1. *Getting Last Committed File Stage:*

***7.1) You can go back to last committed stage of file (or get to the last checkpoint ):***

-> git checkout *<file\_name>*

***7.2) You can go back to last committed stage of ALL file (or this will get all files on their last committed stage or checkpoint ):***

-> git checkout -f

1. *Git Branching:*

***8.1)*** ***It shows all the available branch in the git:***

-> git branch

***8.2)*** ***Command to create new branch:***

-> git branch *<new\_branch\_name>*

***8.3)*** ***Command to Switch to the branch:***

-> git checkout <*branch\_name>*

1. *Git Merging:*

*NOTE: 1) We use git merge command to merge the two branches.*

*2) Suppose developer have developed a new feature in branch B1.*

*3) Manager have approved the new feature and instructed developers to add this feature in main application (master branch).*

*4) Then developer have to merge branch – B1 in master branch to include new feature in main application.*

*5) Developer will first checkout in master branch using git checkout command. Then will use the git merge command to add new feature in application. Commands mentioned below -*

-> git checkout master

-> git merge *<branch\_name\_that\_have\_new \_feature>*

1. *Git Credentials Remove:*

***10.1)*** ***Follow below instructions to remove Github login credentials or Github account Logout from local machine:***

Go to: Control Panel → User Accounts → Manage your credentials → Windows Credentials → under Generic Credentials there are some credentials related to GitHub. Click on them and click "Remove".