**V4. (Tracking the files)**

Note🡺 **Commit 🡺 change in the file “or” Change**

1. **git status**
2. **git init 🡺 Initialize a new git Repositary**
3. **git add --a 🡺 to put all files in staging Area // 🡺git add “filename.ext” 🡺will put the file into commit to stage**
4. **git commit –m “initial commit or xyz” 🡺 to commit all files**
5. **git log🡺 to know all the commits we have done till now (Press Q to Exit the PAger)**

**V.5 (Cloning a Remote Git Repository from Github)**

**Notes🡺 PWD(Present Working Directory) ,LS(List of Contents),cd(Current Directory)**

**🡺rm – rf .git (will delete the .git file Hence Git repository won’t be git repository any longer)**

* git clone “URL”🡺 Clone the Git URL
* 🡺git clone “URL” NewName 🡺 Put the new name of the git repositiory

**V.7 (.gitignore: Ignoring Files in Git)**

**Note 🡺1.** Touch errore.log(filename)🡺 will generate an error log file

**2 🡺 touch.gitignore🡺 will start ignoring to error file.**

* **\*.log(Will ignore the All Ignore Files ”Will have to Write this command into .gitignore File”)**
* **(/dir) will ignore the Whole Directory (This command also has to written into .gitignore file)**

**🡺**Put the file or folder name into .gitignore folder to ignoring that file or folder (along with the path) successfully.

**V.8 (Git Diff)**

**1🡺 git Diff🡺 git diff will compare working Directory from Staging Area.**

**2🡺 git diff –staged 🡺 will compare the last commit to staging Area.**

**V.9🡺Skipping The Staging Area**

**🡺 git commit –a –m “Direct Commit” 🡺 This Command Will Skip the Staging Area**

**V10🡺Moving and Renaming Files In Git**

* **Git rm “Filename.ext”🡺 will remove the Filename.ext from the git**
* **Git mv “PreviousName.ext” “NewName.ext”🡺 will rename the git File.**
* **Git rm –cached “FileName.ext”🡺 will help to ignore any file (Won’t Track the Given File)**

**V11🡺Git Log: Viewing & Changing Commits In Git**

**🡺 rm-rf .git🡺 (Current Git Repository won’t be git repository any longer by using this command)**

**🡺git log --stat 🡺 (This command will tell the summary of changes in short)**

* **Git log –p 🡺 This command will show the Commit and the changes occurred in that commit as compared the previous one.**
* **Git log –pretty=oneline🡺 (This Command will show all commits in one line)**
* **Git log –pretty=short🡺 (This command will show all commits in short)**
* **Git log –pretty=full🡺(this command will show all commits with more information)**
* **Git log –since=2.days🡺(this command will tell the last 2 days commit) “Days can be replaced by Months/weeks/years”**

**V12🡺 Unstaging & Unmodifying Files in Git🡺**

* **Git restore –staged FileName.ext🡺 This Command use to Unstage a Staged File**
* **Git checkout – FileName.ext🡺Will Take the File into Unmodified State(will Match the File as per Last Commit)**

**V13🡺 Working With Remote Repositories🡺**

* **Git Push –u Origin Master🡺 This command used for Pushing any file on github.**

**V14🡺Setting Alias in Git🡺**

**🡺git config –global alias.st status🡺will replace git status to git st**

**🡺git config –global alias.last ‘log –p -1’🡺 instead of Writing Full Command you can write git last**

**V15🡺Creating & Switching Branches In Git🡺**

**🡺git checkout –b develop🡺 will create a New Branch Given Name as develop**

**🡺git checkout master🡺 to switch master branch instead of writing master you can write any branch name where you want to go.**

**🡺git branch🡺 this command will give all the branches**

**V16🡺Branching and Merging a Production Grade Project 🡺**

**🡺git merge branchname🡺This Command will merge the branchname into master branch**

**V17🡺Resolving Merge Conflicts🡺**

**🡺git branch –v🡺 This Command will show Last Commits of Every Branch You have**

**🡺git branch –merged🡺 this command will show the branches that you have merged**

**🡺git branch –no-merged🡺 will show the branches that you haven’t merged**

**🡺git branch –d branchName🡺 will Delete the branch if the branch is Merged with other Branch otherwise Will give Warning or Error**

**V18🡺 Git Branching Workflow in Production**

**V19🡺 Pushing Git branches To Remote Repositories 🡺**

**🡺git push origin Branchname🡺 this command will push the current state in the given branchname (Caution 🡺 When you will use this command from the git bash inside the git bash use the same branch)**

**🡺git push –d origin BranchName🡺 This command will delete the Remote Branch..**