GIT and GITHUB

SCSS – Source code control System > git and github

1. Maintain history of the project
2. Github helps in making repositories
3. Git- version control system
4. Red- deleted green –added

.git – hidden foder

Commands

1. Git init
2. ls –a // show hidden file
3. touch names.txt
4. git status – what files are changed
5. git add . –all files in the staging area
6. git status –picture is clicked
7. git commit –m “Names is added”
8. git restore - -staged names.txt // outside the stage
9. git log //get the log of file
10. rm –rf names.txt
11. git reset 13u923u014912214
12. git stash // go to back stage
13. touch surname.txt
14. git stash pop //come back to the unstage area
15. git stash clear //the backstage people are gone

GITHUB

1. git remote add origin “urls”
2. git remote –v //show all urls attached to folder
3. git push origin master

branching- directed acyclic graph

Never commit on main branch directly we create a new branch

1. Git branch feature
2. Git checkout feature HEAD pointer is pointing to the feature branch
3. Git checkout main
4. Git commit
5. Git merge feature // my branch is part of main branch
6. Git clone “url”
7. Git remote add upstream “url” -- upstream url is the url from where I have forked it
8. Git branch kunal
9. Git checkout kunal
10. Git push upstream branch name //error
11. Git push origin kunal //access is their
12. No new commits possible in main branch. It will add commits

Imagine working on 10 features and every feature we create one pull request

Create a new pull request.

1 pull request = 1 branch

1. Git log
2. Git reset 213413213315
3. Git add .
4. Git stash
5. Git push origin kunal –f // force push it
6. Git checkout master
7. Git fetch - -all - -prune
8. Git reset - - hard upstream main
9. Git push origin main

Other method

1. Git pull upstream main
2. Git push origin main

Merge Conflicts and Squashing commits

Git branch temp

1. Git checkout temp

// before chaning branch we have to make sure main branch is uptodate

1. Git rebase –i copy // interactive environment

s //squash –whichever is listed as pick then above these s whichever pick we have merge it in these commit

1. ESC then :x –allow to create a new message
2. Commits merged :x //back

// git conflict- when A made change in line 3 and B also made change in line 3 git confused wants my help