Date:
Page No:

GIT & GITHUB

Clone Topics Covered:-

- clone
- add
- commit
- push
- pull

① clone → the repo will be copied/initialed in our resp dir

② add → whenever there is any change in the dir
- new file created
- changes in existing file

git add . → all the changes occurred will be tracked

git add <file_name> → only that file

Note:

If there is any change in existing file, then it will be M - modified

If there is any new file created, then it will be U - untracked

when add cmd is used U and M will be tracked and added!

③ commit
↳ this is to just give a title for the changes we have made with a title and a description

git commit -m " " -m " "
↓ ↓
title description

④ push
↳ the prev cmds were all for changes within the local pc
↓
for changes to be shown in the git repo we have to push

git push origin main

→ also can use '-u' or '--set-upstream'
↓
this is to not mention the branch name again
↓
after that we can use 'git push'

Note:
When you create a folder in your pc, add files ~~create~~ ~~many~~ and use it
AND then you want to upload everything the total folder into a git repo you will have to use ~~git~~ "remote" cmd
step 1 ⇒ create a repository in git and initialize using git init
↓
step 2 ⇒ git remote add origin ___ git repo link

git remote -v → to check connected remote repos

step 3 ⇒ git push origin master

Date:
Page No:

→ Git Branching :-

- git branch ~~to~~ ⇒ to list all the active branches of the repo
- git checkout -b ______ -11-readme

↓ switch b/w branches

↘ to create a new branch

→ any name could include the issue number,

Note:

git diff <branch-name>.

→ will let you see all the changes made in the new branch

- git merge <branch-name>

→ Both the branches will be merged

→ before this what we can do is (general practice)

↓

1) first get into the newly created branch → git checkout < >
2) then push the changes ⇒ git push origin <——>

(or)

git push -u origin <——>

3) When you push this, you will get a link for a PR

go to github and create a pull request ↓ pull request

4) After this the changes will be made in the github repo, merged and not in your local env so I have to PULL

go to your main

git pull origin main

5) After this you can safely delete the prev branch by

git branch -d <branch-name>

Note:

If you have not created any new file and only have modified files → you can add and commit at the same time using → git commit -am "

Date:
Page No:

⇒ Merge conflicts:

↳ must be manually done

changed only in main / changed only in branch

↓

Merge both

↓

Manually correct it in Code editor

↓

Add & Commit changes

⇒ Undoing:

Undo

- Add: (added by mistake)

  git add <file name> } to undo

  git reset <file name>
  git reset } undo all changes

- Undo a Commit:

  git add .
  git commit -m " "

  } to undo

  git reset HEAD~1

  ↓ points to the latest commit

  → One commit further

Note:

git log

→ here you can see all the commits made in your current branch

↓

Also there will be unique hashes for each commits use them to reset if required

↓

git reset <hash> → Unstages it

git reset --hard <hash> → Unstages and completely removes it

⇒ Forking: (complete copy of the repo)

→ when you are working on others repos you ~~have~~ do not always have access to make changes

↓

So make your own copy by forking it.

↓

Then you can do/make changes

↓

Then create a PR