GIT/GITHUB NOTES

COMMON COMMANDS

ls - list files

cd (directory name)- change directory

cd ../ - go to previous directory

mkdir (foldername)- make a folder

rm (foldername/filename) - remove

code . - open folder in vs code

pwd - print current working directory

ls -a - hidden files

MEANING OF LOCAL AND REMOTE FILES

remote file – github file

local file – file on laptop

CONFIGURATION

git config --global user.name 'Vanshika Ganjoo'

git config --global user.email 'vanshiganjoo@gmail.com'

FLOW OF EVENTS FROM GITHUB TO LAPTOP

github -> laptop -- github repo ->clone-> changes

clone - cloning a repository on local machine from github

copy https link from github

status - displays state of code

4 TYPES OF FILES

- untracked-new files that git doesnt track yet because they were never committed
- modified - changed
- unmodified - unchanged
- staged - modified and ready to be committed

if you make changes in a new file it becomes untracked and in an old file it becomes modified

first you need to add the changes - then it reaches staged stage

then you need to commit the changes - then it reaches unchanged stage

add - adds new or changed files in your working directory to the git staging area

git add <-file name-> - to add one file

git add . - to add all files

commit - record of change

git commit -m "change name"

Your branch is ahead of 'origin/main' by 1 commit. means local system is 1 commit ahead of github

push- upload local repository(laptop) content to remote repository(github)

git push origin main

git: The command-line tool for interacting with the Git version control system.

push: The command used to upload local repository content to a remote repository.

origin: The name of the remote repository you want to push to. By default, the name origin is used for the remote repository cloned from or the primary remote repository.

main: The name of the branch you want to push your changes to. The main branch is typically the default branch in many Git repositories (though some projects might still use master or another name).

Putting it all together, git push origin main means you are pushing your changes from your local main branch to the main branch of the remote repository named origin

FLOW OF EVENTS FROM LAPTOP TO GITHUB

create a new folder

enter the folder

git init

init - used to create a new git repo

git remote add origin <-link->

remote is the github repo which is named as origin

git remote -v (to verify remote)

git branch (to check branch), by default - master

git branch -M main - to rename master to main

git push -u origin main

-u is creating shortcut for future - only type git push

branch is created - separate for each feature

main and feature branch code is made same through merge

BRANCH COMMANDS --

git branch - to check branch you are in

git branch -M main - to rename branch to main

git checkout <-branch name-> to move from one branch to another

git checkout -b <-new branch name-> - to create new branch

git branch -d <-branch name->

you cannot delete the branch that you are in. You will have to checkout to a new branch first

the changes you make in the feature branch will remain only in that branch and won't be reflected in main branch until you merge them

MERGING CODE --

2 ways

1st -

git diff <-branch name-> - to compare commits, branches, files etc.

git merge <-branch name-> - to merge 2 branches

2nd -

Pull Request -request to merge(pr review, review done by sr developer to see if your branch's code can be merged with the main code) - it lets you tell others about changes you've pushed to a branch in a repository on github

the changes are not reflected in vs code so use pull command to do so

PULL COMMAND --

git pull origin main

used to fetch and download content from a remote repo and immediately update the local repo to match that content

merge conflicts - 2 branches have change in same line

an event that takes place when git is unable to automatically resolve differences in code between two commits

git merge <-branch name with which you want to merge->

can change using vs code –

accept current change - change of current branch accept incoming change change of main branch

or

do manually - remove all the weird lines

and delete whatever

then save the changes. add and commit in the branch

move to other branch

then merge the two files

push the files to github

UNDOING CHANGES --

Case1: staged changes (added but not committed)

git reset <-file name->

git reset (all files)

reaches a stage like that of just before adding the change

Case 2: committed changes(for one commit)- if we want to move one commit back

git reset HEAD~1

undoes the last change and moves the latest change to the second last change

git log - to check the commits made

Case 3:committed changes(for many commits)

git reset <-commit hash->

git reset --hard <-commit hash->

creating a fork --

a fork is a new repository that shares code and visibility settings with the original "upstream" repository

fork is a rough copy

fork and then make changes, commit, create pull request for merging