################ important git commands ############

git config --global user.name --> set git username

git config --global user.email --> setting the email address associated

git remote add origin <url> --> adding the remote repo url

git init --> initilizes the git repository

git clone <repod_address> --> to clone repo into your local

git pull origin master --> to fetch and incorporate the remote changes into local

git pull –all --> Pulls all branches info

git add <files> --> add the untracked files to staging area

git commit -m "Descriptive Message" --> to commit the from staging area to local repo

git push -u origin master --> Pushing local changes to remote repo

git status --> to know the status of the repo

git reset file --> unstage a file

git clean -i --> removes untracked files from working directory

git diff --> diff of what is changed but not staged

git diff --staged --> diff of what is staged but not yet commited

git log --> to get commitid history of the repo

git show --> shows various types of objects

git branch --> gives currne working branch

git branch -m <newname> --> Rename branch name

git checkout branch_name --> chekcout to target branc_name

git remote -v --> To see remote repository

git branch -a --> to list out the remote branches

git remote -r --> To list all remote branches

git branch -d branch_name --> Deletes the branches locally

git branch -D branch_name --> Deltes the branch remotely

git checkout -b branch_name --> Creates and checks into newly created branch

git merge branch_name --> Merges the branch_name code into current branch

git log --follow --> logs of pariticular file

GIT MERGE:

->git checkout feature_branch
->git merge master (here master is the branch which we want to merge changes to feature bracnh)

git fetch --> Downloads all the history into current directory

differences between fetch and pull --> https://stackoverflow.com/questions/292357/what-is-the-difference-between-git-pull-and-git-fetch

git stash --> Adding staged files into stash directory (Ref: https://www.javatpoint.com/git-stash)

git stash save "stash message" --> Adding staged files into stash with stash message

git statsh list --> To list the stash things

git stash show --> Track the stash changes

git stash apply --> undo the stash changes & saves a copy in stash directory

git stash pop --> undo the stash changes

git stash drop --> Permnantly remove the stash changes one by one or we can mention stash id

git stash clear --> Clears all the stashs at one time

git cherry-pick <commit-hash> --> Cherry picking in Git means to choose a commit from one branch and apply it onto another.

git reset --soft <commit-hash> --> Changes present at stagingarea

git reset --hard <commit-hash> --> Changes will be gone

git reset <commit-hash> --> Changes present at working area

git revert <commit-hash> --> Reverts to previous commit-hash

git log --oneline --decorate --graph --all

git tag "tag_name" <branch_name> --> creating a tag

git tag tagname -m "write the message" --> it is used to add the message to the particular tag

git push origin "tag_name" --> Pushing a tag

git tag --d tag1.1 --> It deletes the tag

git push –tags --> Push all the tags at one time

git remote --> to check the remote

git push --set-upstream origin branchname --> to set the upstream branch

GIT REBASE:
Git rebase is an alterntive to merging
--> git checkout feature_branch
--> git rebase master
--> git rebase --abort (we can abort rebase and clear the conflicts)

Generally rebase rewrites projet hisotry. the main diff b/w merge and rabse, rebase eliminates unnecessary merge commits.

git log --oneline --decorate --graph --all

Git 3-tier architecture workflow