**Git Commands:**

**Set Username and Email:**

git config user.name “Vivek Makwana”

git config user.email “[vivekmakwana848350@gmail.com](mailto:vivekmakwana848350@gmail.com)”

**View User and Email:**

git config user.name

git config user.email

**Initialized the empty folder/directory:**

git init

change the directory:

cd d:

make a directory:

mkdir test

make a file:

touch index.html – will create a file only

nano index.html – will create the file and opened it in the editor

**Check files status:**

git status

**Add and commit file:**

2 ways to add file:

(1) individual file

git add index.html – with file name

(2) All files

git add . (last is period)

commit a file:

git commit -m “index files commited”

**Track the file changes:**

git log

git log -- oneline (use for single line message)

**Checkout commit: - only for the previous code preview**

git checkout logID (here the logid will be used at the end for particular commit)

git checkout master

**Revert commit: - to remove a particular feature**

git revert logID (here the logid will be used at the end for particular commit)

another editable text will appear for the revert to exit:

shift + :

type wq

**Reset commit: -- this will not make change anything in the present working directory**

git reset logID

**This will make a change in the commits (permanent deletes changes)**

git reset logID --hard

**Make new branch:**

git branch feature-1 (at last we use branch name)

**Making a new branch and go directly into it:**

git checkout -b feature-1 (at last use the branch name)

**Check available branch in repo:**

git branch -a

**To go to particular branch:**

git checkout feature-1 (at last we use branch name)

**Note:** In branch you can make changes in the project by not modifying the master branch, So the changes made to other branch will not be appear to the master branch.

**Delete a branch (if not merged):**

git branch -D feature-1 (at last use the branch name)

**Merging 2 branch into master branch:**

Note: If you want to merge below branches in the main, you have to be in the main branch and then write below commands this will merge both branch in the main branch.

git merge feature-a

git merge feature-b

**GitHub**

**2 ways to create a repo:**

**(1) if the project/code is locally available we upload it to GitHub with following steps:**

On GitHub

create a new repo, fill required fields on GitHub

copy the repo link and in the git bash type

git push link master

(here paste the link and mention in which branch you want to add)

by a default GitHub has a **main** branch.

**Note:** now if we change something in the code and then want to upload that code we can simply use the above command after commit the file

git push link master

but if you don't want to use the whole link than we can simply make an alias name for the link.

git remote add alias\_name link (here alias name can be anything)

than with the same command using alias we can push that code in the repo

git push alias\_name branch\_name (alias - origin, branch - master)

**(2) Clonnig of remote repo from GitHub:**

if we don't have any local projects/code we simply create a repo in GitHub

On GitHub

create a new repo, fill required fields on GitHub

by deafault main will be deafault branch

copy the repo link and type in git bash

come to parent directory

cd ..

this will clone the whole repo in the local machine with the directory

git clone link

go to that directory

cd portfolio

now make some changes in the portfolio like add files etc.

touch index.html

add the file to staging area & commit the changes

git add .

git commit -m "added index file"

upload that on the GitHub,

now here the repo has already has alias name so we don't require to specify any alias\_name for the repo below command give the alias name for the repo for push and fetch

git remote -v

now upload the the file on GitHub

git push alias\_name branch\_name

**Ex:** git push origin main

check the changes on the GitHub now

**Collabration on GitHub:**

using pull we can up-to-date our local repo from the remote repo

git pull alias\_name branch\_name

this will fetch all the code form the remote repo and merges to our local main/master branch

git pull origin main

now all the remote code has installed in the local repo

make a seprate branch & it will redirect to you to that branch

git checkout -b index.html

now make some changes in the index file

add the file and commit it

git add .

git commit -m "added index file"

now first merge the branch into the local master branch and than push that master branch into remote repo

git push origin index.html

it will push that file to the remote repo into the index.html branch

now click on the compare& pull request

here we can comapre the currernt state of the master branch and and chages we have mde to the file

we can make a pull request and merge the branch into the main/master branch

now open a pull request

other developr can review it, can see commits that have been made and check the file code as well can add comment also

if all are good we can merge the pull request

after merging that it will merge into the main/master branch

than we can delete that branch beacuse the changes madde to the branch has been merged to the main branch

now u can see the master branch index.html file that has been changed.

go to the main branch now

git checkout main

for adding any new feature you should always copy the remote repo into local repo

git pull origin main

and make a seprate branch for that feature for not letting any issue to the main branch

git checkout -b branch\_name

**Ex:** git checkout -b img-add

now made any required changes to the file and add it, commit it and push it to the GitHub

now create a pull request, review the changes made to the file

if all good merge the pull request to main branch and than delete the sub branch of that feature

now all changes have been made in the file has been merged to the master/main branch repo

**Forking**:

Used to contribute in the open source projects

it will copy whole repo in out account so we can make a change without harming the original code

on GitHub in any repo just click on the fork and it will make a copy of that repo in out account

now clone that repo in the local machine or also can pull

git clone link

git pull link

with alias\_name

git remote add origin link

go to that repo directory

cd dir\_name

check status

git status

add files after changes in the file

git add .

commit the files

git commit -m "commited"

push the file to the remote repo in our account

git push origin master/main

than create a pull request on the main account of that repo

they will review the code and than decide to merge that change in the repo if all good