First open cmd in any folder

Like here I open cmd in my desktop

Text

Description automatically generated

Then make a project folder by mkdir project

And after that cd project

Text

Description automatically generated

So now we have our project now we want to maintain the history of our project using git

Now first initialize git by git init command

Text

Description automatically generated

Now inside my project folder I have opened gitbash

And if you type ls -a then it will show you a .git folder

Text

Description automatically generated

Now we have git folder over here now we will be able to maintain the history of our project

By history here I mean that you will be able to see when and what changes we did with our project

And also who made those changes

One command “git status” - The git status command **displays the state of the working directory and the staging area**. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git

Text

Description automatically generated

**Now we will add the files into the local repository in order to be able to start tracking the files and changes we make.**This process of adding a file to a local repository is called committing.

It is a two step process in order for us to commit the code to the local repository. We will need to first add the file into a staging area and then commit. This extra step in the process gives you control over what files you actually want to commit to the local repository as well as allow you to change your mind if you accidentally choose the wrong file initially.

To add only a single file to the staging area, you can use git add program.py

To add multiple files, use this: git add program1.py program2.py

If you have many files or are too lazy to type out your file names, you may use git add . to add all files that are in the folder.

Text

Description automatically generated

Now as you can see in above image we have a untracked file program.cpp

And now type git add . and its status will be changed to green

Text

Description automatically generated

Now we will commit our files by command – git commit -m “program.cpp file added”

Text

Description automatically generated

Now type command – vi program.cpp to make some changes in our file

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

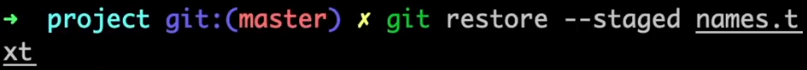
Now again type git status and it will show that our program.cpp has been modified

Text

Description automatically generated

And now to remove this file from staging

Use command – git – - restore staged program.cpp



Text

Description automatically generated

Now to see entire history of the project use command – git log

Text, chat or text message

Description automatically generated

And now to remove our file use command – rm -rf program.cpp

And check git status

A screenshot of a computer

Description automatically generated with medium confidence

M itni der se commit kiye ja rha tha bina add kiye 🤣🤣🤣🤣

Text

Description automatically generated

Imagine you deleted this file by mistake and if you want to remove a particular commit you cannot just remove one particular commit as each commit has a hash id and each commit build on top of each other

You can unstage those commits

Text

Description automatically generated

Now with git reset command I have removed all the commits above that particular commit

Now type command git log

Text

Description automatically generated

As you can see I have removed that one particular commit that was above that commit which we typed with git reset

Text

Description automatically generated

Now as I typed git status you can see that our that particular commit which we has unstaged

Git stash

git stash **temporarily shelves** (or stashes) changes you've made to your working copy so you can work on something else, and then come back and re-apply them later on.

Text

Description automatically generated

Now type command – git stash pop

Text

Description automatically generated

Now git stash clear

Text

Description automatically generated

So now those changes which were in our unstaged area have gone now

Now create a repository on github.com

Now copy url of that repository

Now type command – git remote add origin URL

And now to push our changes to our repository use command – git push origin master

Here master is our branch

Also one more thing I you want to make a new file you command – touch file2

Whenever you are creating a new feature or resolving a bug always create a separate branch

To create a new branch use command – git branch branchname

Now type command – git checkout branchname

Now our head will point towards our new branch and all the commits will happen to that branch

Now to make that branch part of your main branch use command – git merge branchname

Now to contribute to someone else’s project first go to that project on github.com

Click on fork select your account

Text

Description automatically generated

Now clone this by copying URL

Graphical user interface, application, Teams

Description automatically generated

Now type command – git clone URL

It will download that project In your local environment

Text

Description automatically generated

Now we will add upstream

Upstream is url from where you have forked it

Text

Description automatically generated

Now to make pull request

First do some changes

Create new branch

Then git add and git commit and after that type command git add remote branchname

After that click on pull request

Background pattern

Description automatically generated

Remember one thing one branch = one pull request

Make sure for every new feature/bug you make a new branch