Ek folder hai , usme bahut sari folders and files hai , to inn sabke collection ko hi apan **REPOSITORY** bolte hai . ism eek .git folder hota hai , jisse apan sab cheeze track kar skte hai

Diagram

Description automatically generated

1)Jab aap start karte hai, tab ek empty git file mil jaati hai , jisme koi kuch hai hi nahi .

Initially sari files jo hai, wo **untracked** hogi

2)Usko add karna padega, tab jakr wo **unmodified** wli category me aayega , git add --a wale command se add karna padega ( ab yaha se tracking shuru hoti hai files ki )

3) iske baad jo bhi change karenge files me, wo **modified** wli category me chali jaegi(ab yaha se status me dene lagega)

4) Uske baad usko **staging area** me dalna hoga

5)Ab agar staged area se aakar apan ne file ko commit kar diya hai , to fir wo folder unmodified wle stage me chale jaega

**pwd** --> present working directory , jisme aap kaam kar rhe ho

**ls** --> content ko list karta hai

**cd** --> change directory

**git status** --> gives the status of the directory or the repository

**git init** --> initialises a file into repository which can be tracked (git repo ban jaega isse)

**git add –a(git add .)** --> adds all the files into staging area

**git commit** --> commits all the files from staging area

**git commit -m "your msg here"** --> commits with a message

**git log** --> gives the log of commits made (enter q to exit this dialogue box)

all the files which are committed into the git, can be traced and will observe anything which gets changed in the files

**git add filename.ext** --> adds only this particular file into staging area while other will not

**rm -rf .git** --> ye wla khatarnaak hai , kyuki isse puri git delete ho jaati hai

**touch filename 🡪** Isse ek file bana dega , apni working repository me , agar hume gitignore wali file banani hai , to apan humesha touch ka use karenge

**touch .gitignore🡪**  Isse ek .gitignore wali file ban jaegi, jissme apan jiske bhi naam dalenge ,wo ignore hone lag jaegi , like jese mene error.log wali file banayi , wo ab ignore hone lag jaegi

**gitignore 🡪**isme agar apan ne “\*.log” karke daal diya, to jitney bhi .log wali files hogi, usko ignore karne lag jaega, and kisi bhi file ko track nhi karega

* Ab kisi directory ko hi apan ko ignore karna ho, to usko bhi apan git ignore wali file me daal do, to apne aap hi track karna band kar dega
* Blank Folder ko git by default ignore karta hai, isiliye apan ko specify changes karna hoga, tabhi jakr wo apni folder ko track karega
* /dir/ --> agar ese kara hai apan ne, to particulary bahar wale dir ko ignore karega, warna agar sirf dir/ --> kara hai, to sare dir naam ke folder ko ignore karega

**git diff 🡪**  Ye working directory or staging area ke beech me difference batata hai , apni language me bole to git status karne ke baad same file red and green dono me ho , to unn dono ke bech ka difference batata hai

**git diff --staged 🡪**  ye apan ne pehle ke commit or staging area ko compare karta hai pichle commit ko and abhi ke staging area se difference nikalta hai

**git commit -a -m 🡪** Isse sari tracked file ko staged karke directly commit ho jaegi  **“direct commit”** Koi nayi file jo untracked hogi , wo nahi commit hogi

**git rm filename 🡪** Isse jo bhi filename daloge , wo sidhe delete ho jaegi ( remove) and staged bhi bhi ho jaegi

**git mv first\_name renamed 🡪** Isse jo bhi file ka name hoga , wo change ho jaega ,and wo staged bhi bhi ho jaega , (mv – move ) (to ek file ko move karke dusri file me transfer kar dega)

**git rm –cached filename🡪** Isse jo bhi filename daloge, usko track karna band kar dega , for example agar koi file apan ne gitignore me dali , but wo already track ho rhi thi to usko ignore karne k liye ye command chalana hoga, jisse ye track hona band ho jaye

**git clone link name🡪** isse git ka clone bana dega, jo bhi apan name denge, usse save hogi

**git log -p 🡪** sare changes dikha dega, with difference in them ki kya kya kara apan ne uske sath

**git log -p -4🡪** pichle 4 changes dikha dega, with difference in them ki kya kya kara apan ne uske sath

**git log –stat 🡪** short me bataega , ki kya kya hua uss commit me (short summary de dega)

**git log --pretty=oneline🡪** Sare commits ko ek hi line me bata dega(q daba kar exit)

**git log --pretty=short🡪** Short me sare commits dikha dega(q to exit)

**git log --pretty=full🡪** Full info dikhaega commit ki (q to exit)

**git log –since=2.days🡪**Pichle 2 din k commits dikha dega,(similarly for weeks ,months and years)

**git log --pretty=format:”dekhlo website par jakr”🡪** Different ways to view log, iska apan website par jakr dekh sakte kese implement karna hai , git scm wali website par jakr dekhlo

**git commit –amend** 🡪 Isse pichle commit me apna change add ho jaega pehle I daba kar edit kar paenge, fir escape fir :wq ,then usse exit ho jaoge

**To Unstage any file 🡪 git restore --staged filename**

**To Unmodify any file 🡪 git checkout -- filename** (agar yaad aata hai ki glt kra tha apna ne kuch but isse sare changes khoo baithoge ,agar commit nhi kara ho to to beware while using this PICHLE COMMIT SE MATCH KARA DEGA

**git checkout -f🡪** Jitne bhi modified files hai , wo sari apni purani commit se match ho jaegi

**shift + insert 🡪** paste hota hai git bash me

**git remote add origin url🡪** it will add a url named to origin(from github)

**How to add a git repository in github link 🡪**

<https://www.youtube.com/watch?v=Thx5yerhlhs&list=PLu0W_9lII9agwhy658ZPA0MTStKUJTWPi&index=14&ab_channel=CodeWithHarry>

**Alias 🡪** bade bade code ko chote me convert kar dega

Code 🡪 **git config --global alias.changed\_name original name**

Example :- git config --global alias.st status 🡪 to ab isse status ko apan sirf st se bhi call kar skte hai

**git config --global alias.unstage 'restore --staged --' :-**

isse restore –staged – ka name change karke unstage rakh diya, to jab bhi apan ko wo use karna hoga, to sidhe apan unstage likhkar kar sakte hai use

like unstage this.txt

**$ git config --global alias.last 'log -p -1'**

Isme apan ne log -p -1 wale ka name change karke last rakh diya, to ye apan ko last commit bata dega ki kya kara hai apan ne

Branches :-

**Git checkout -b develop🡪**  Isse ek nayi branch ban jaegi , jiska naam develop hoga(name apne hisaab se bhi rakh sakte hai )

**Git branch 🡪** Show the number of branches present

**git checkout branch\_name 🡪** isse branch change ho jaegi , jispr bhi shift karna hoga

**git merge branch\_name🡪**  Isse jo bhi branch hai wo merge kar dega master branch se