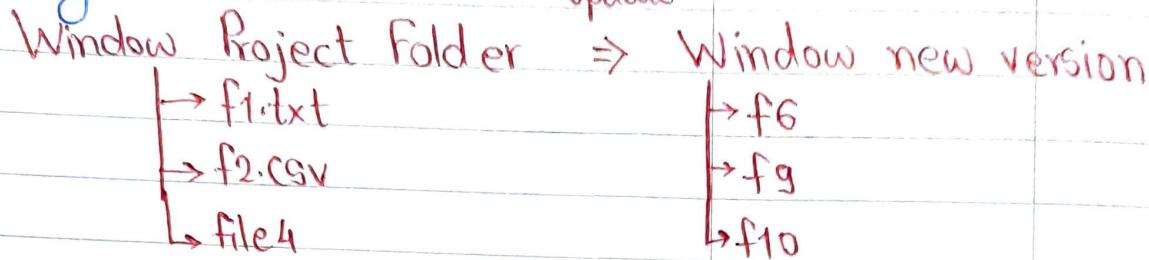


Git & Git Hub.

Why? \Rightarrow



Git is a Version Control System

For ex~~e~~ You have one folder of 20GB after someday you made small modification, Hence you create another folder with new update & version - Hence you copying same folder again & again & Resultant into High memory utilisation

Feature

easily file Recovery - to get deleted, last edited file.

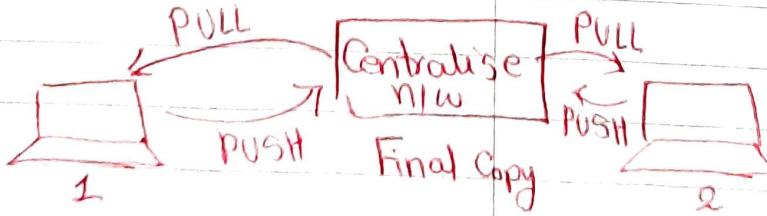
who introduce issue & when - Track record -

roll back to previously working state

History Local Version Control System \Rightarrow But if corruption of HDD ^{No Hacking / No Pulling} data loss

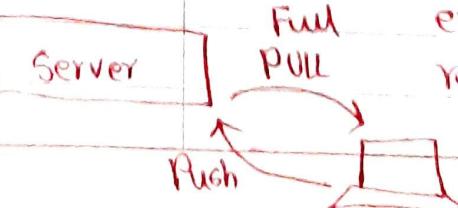
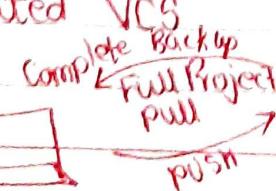
pros: but can track file. & roll back

Centralise VCS :-



Cons: Server damage then lost old changes.

Distributed VCS



even if Server damage recover from Sys.

NAME, ADDRESS & E-MAIL

PHONES

MOBILE

How was Git Created
 Linux Kernel → Git develop ✓ {Winner}

To made VCS Free for All - When developing Linux OS they come up with Git hub in year 2005.
 Git is Repository to maintain version of Project.

Git Hub :- Hosting Service - To provide access to Git documentary. (Git is Free).

BitBucket, Gitlab other Services.

Git Feature :-

1) Snapshot Capture not differences :-

- Take Snapshot of your editing - put in .git repository.
- .git Provide you all past History of your project.
- from .git you can fetch any Version or previous data.

2) Almost every Operation is local :-

- Can make any modification locally once done with modification then push to Center repository.

Local ⇒ W.O. internet

Remote ⇒ on internet

3) Git has integrity ⇒

- Unique checksum on every modification - SHA checksum changes on every modification - If you get different checksum then data may be Compromised.

4) Git generally only add data

Job DevOps
C / D Navi Mumbai
Engg. 25 {Fresher}

NAME, ADDRESS & E-MAIL

PHONES

MOBILE

Download Git.

✓ Install

Git Bash Terminal (Terminal) Pro

> git ----- get all detail.

X

Create Folder for Git Uses.

→ Create Username & email for registration of
git config --global user.name "Akshay"

git config --global user.email "akshay.dange15@gmail.com"

git config --list ----- give details.

git config --global core.editor

git config user.name

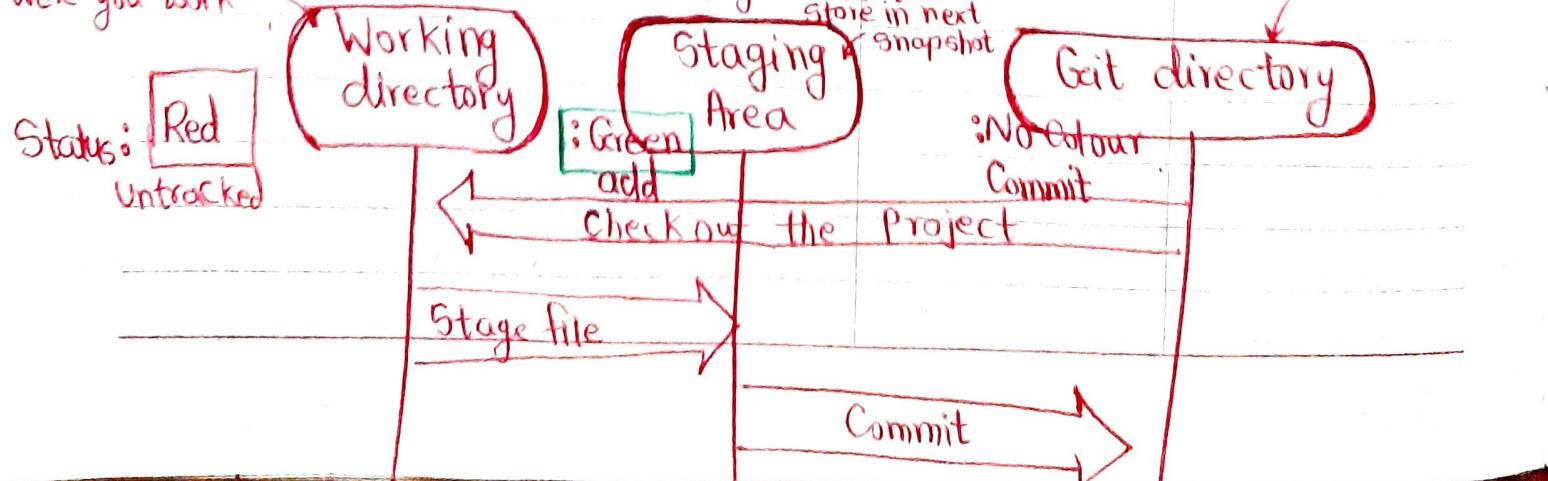
X

Snapshot ⇒ Store the running Task so in future
any mishappening happen then You can access Content.

In git we say "Commit"

Three Stage Architecture of Git

were you work



C/D

NAME, ADDRESS & E-MAIL

PHONES

MOBILE

Staging File \Rightarrow

You develop v1 & take Snapshot of it - put in git repository

Now you update files out of Three two files work as per your "request appl" & one not - So you want only that two file & lastly old file came into new Snapshot - So two correct file know snap, staging file & place into git repository with v2. with that older version file 3.

git status

 \rightarrow know status of your folder

git init

 \rightarrow Convert folder into git Repository.

git add --a

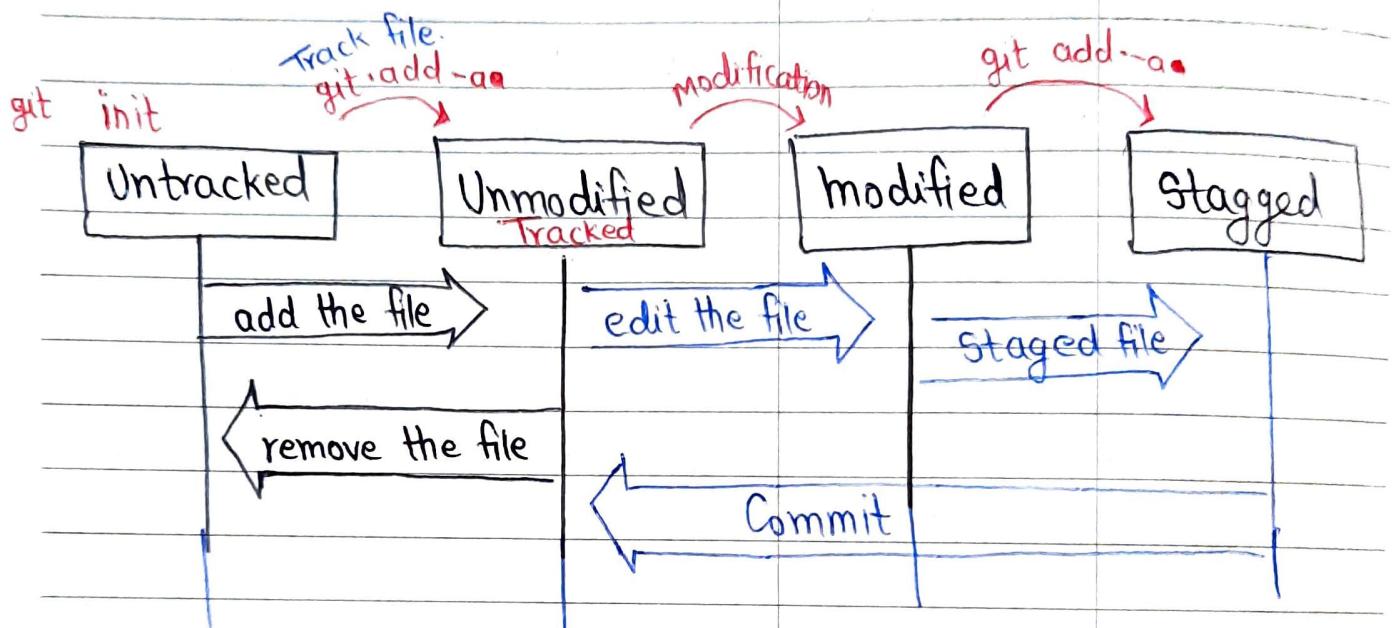
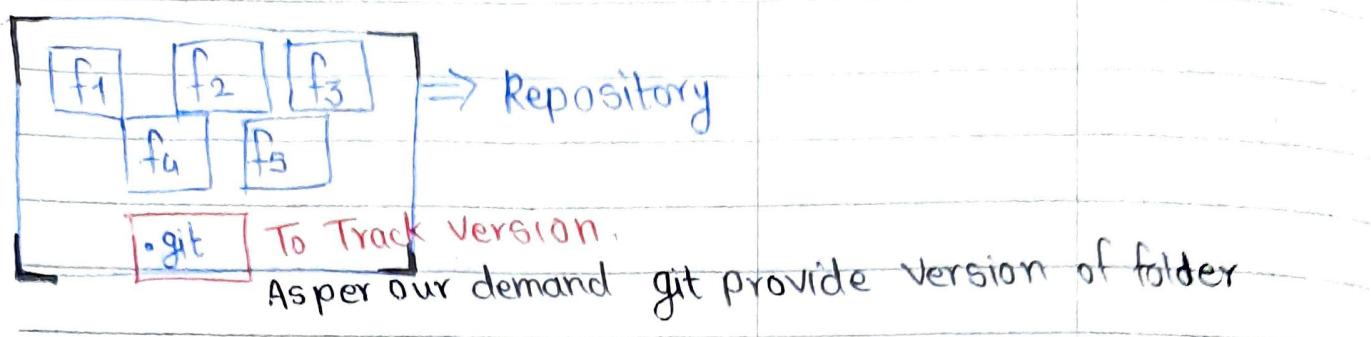
 \rightarrow add files into Staging State, --a denote allgit Commit -m "initial Commit" \rightarrow Create Snapshotgit Commit --amend \rightarrow Shell open to enter data "mess"git log \rightarrow get all detail. Commit Comment modify. \rightarrow :wq to save - now yourgit log -p -3 \rightarrow for log get detail log who make changes & all.git add "file name" \rightarrow individual file add into staging.git commit -m "mess" \rightarrow Staging file add into Snapshot.rm -rf .git \rightarrow remove all - danger caution - Xgit clone <URL from git hub> Name as per your demand
add Github to your git.git log --stat \rightarrow in short what happen with file.git log --pretty = ¹one line ²short / ³Full \rightarrow one line log. Comment.git log --since = 2 days \rightarrow last 2 day status.

C/D

NAME, ADDRESS & E-MAIL

PHONES

MOBILE



• `.gitignore` {Ignore any file, directory?}

When you are working in any folder or file & you don't want to git continuously track that file. then put that specific file (name) into `.gitignore`.

Hands ON

- 1) touch akshay.log. ---> file you don't want to track.
- 2) touch `.gitignore` ---> Create `.gitignore`
- 3) nano `.gitignore`
 > akshay.log ---> add file name into it.
- 4) ~~git add .~~ ---> Stage all
- 5) git commit -m "ignore added"

file
directory
also
ignore

Now onward if any change in `akshay.log` will not be shown at ~~modified~~ section while asking `git status`.

Note:- If you already commit any file and now you put that in `.ignore` file then it will still track the file to explicitly untrack any file from tracking ---> `git rm --cached fileName`

E/F

NAME, ADDRESS & E-MAIL

PHONES

MOBILE

Compare Working directory & Staging file

Some time we stag some file & afterward we made change - then file shown in both staging area & in modified area.

To check we what data is being modified use.

git diff --> show which file & Content change.
Compare staged vs modified - ignore + add

git diff --staged --> Give difference between last Commit & Staging area.

Skip Staging Stage

git commit -a -m "Comment" --> without adding file into staging area can Commit.

* But Untracked (new file) will not directly Commit only Tracked file Commit directly.

Remove files

git rm "file name" --> remove specific file.

git mv first.txt Second.txt --> rename file.

E/F

NAME, ADDRESS & E-MAIL

PHONES

MOBILE

To Unstage any file or folder to modified status.
`git restore --staged <file name>`

Recover modified file to Last Committed file version
`git checkout -- <file name>` --- now last Committed version restored.

To recover multiple file modified file to last Committed file version.

`git checkout -f` --- To restore to previous Commit

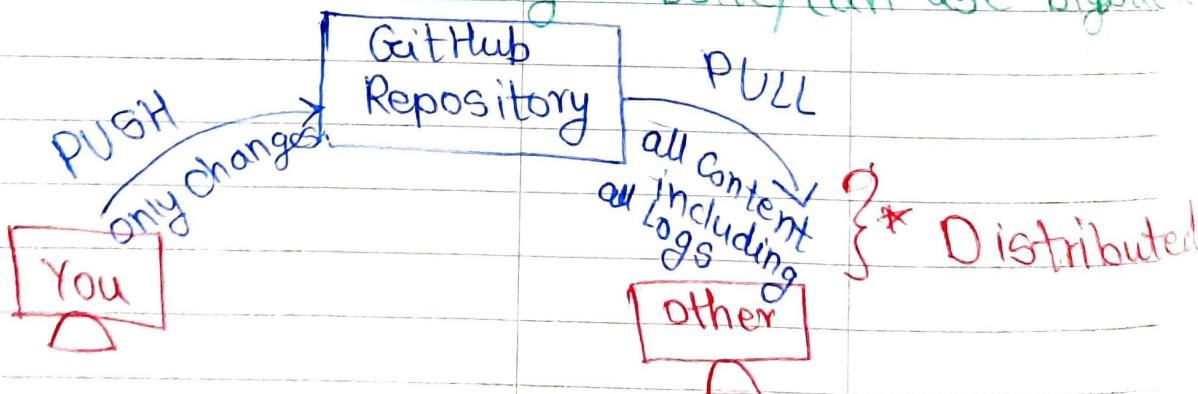
Host Your Code on Server (GitHub)

So publically people can push or pull Content.



PUSH Process

- 1) go to Git Hub → Create new repository - name it description - Just Hosting Website/ can use big bucket



- 2) You have your own repository

now create remote > go to git Hub & copy your URL
`$ git remote add origin git@github.com:CodewithAkshay/
name of remote`

`akshayTutorial.git`

`$ git remote --get your remote name.`

NAME, ADDRESS & E-MAIL

PHONES

MOBILE

You cannot directly push your Content to git Hub - need SSH access - To get that.

In google Search SSH key generation for git follow step

```
$ ssh-keygen -t rsa -b 4096 -c "<your mailid>"  
$ eval $(ssh-agent -s)  
$ ssh-add ~/.ssh/id_rsa  
$ tail ~/.ssh/id_rsa.pub ---- get SSH key Copy
```

on Go to Git Hub Setting SSH & GPG key

In SSH keys Add SSH key.

Now PUSH

\$ git push -u origin master

Now onward You Commit anything push to Server

\$ git push -u origin master

To Push Branch > go to branch > then Push.

git push origin <branch name>

Git Alias { Convert long Command into short keyword }

git config --global alias.st status --- git status & st
keyword general command

To Save time & Convience.

Note :- Be on Some branch to push on git hub

To delete branch on Git Hub ✓

✓ git push -d origin <branch Name>

* Before switching Branch always Commit First

G/H

NAME, ADDRESS & E-MAIL

PHONES

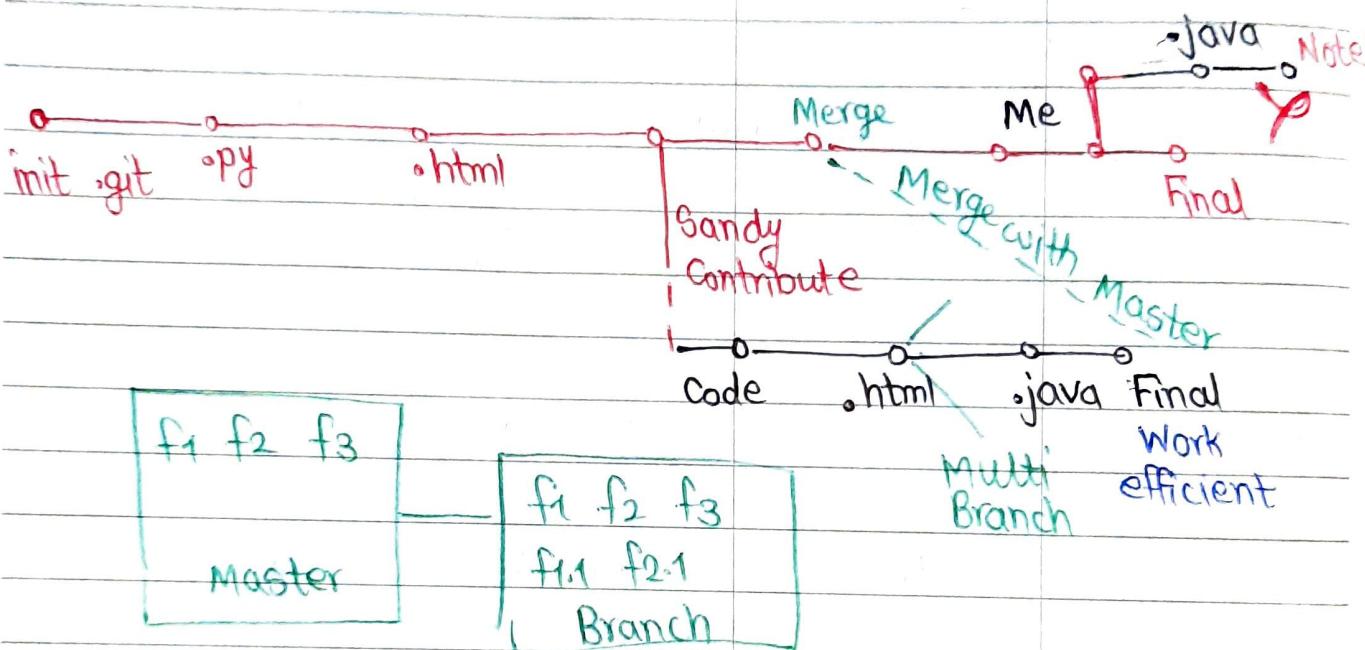
MOBILE

Branching

Master is Hero (Head) of Project.

Branch is a Subsidiary Task Performance - If in future task perform by Branch shows positive result then that Branch merge with Master Branch.

Akshay



If work Continue in master & Branch - If work done by branch efficient then added into Master otherwise reject & Master Continue his task.

Create Branch from master

git checkout -b develop
new branch

To Come back to master from Branch

git checkout master

Check all Branch

git branch

NAME, ADDRESS & E-MAIL

PHONES

MOBILE

Make Project

- 1) download Visual Studio
- 2) Create one folder for project
- 3) Open folder in VS → add index.html → go to bootstrap Documentation > Starter template > Copy.
- 4) Paste into VS.
- 5) Download Live Server extension in VS.
- 6) On bottom go live option > get Hello world on web.
- 7) Now add nav bar from git bootstrap & check on web
- 8) As you wanted Some other changes to made not sure → Create branch → git checkout -b test
- 9) Now in test add, modify Content it will reflect changes but not impact the main file in master.

Now result is Good Hence merge in master

- 1) Commit Changes in branch
- 2) enter into master
- 3) git merge <branch name>
Accept one out of two in VS.
- 4) git add.
git commit -m "merge paths"

G / H

NAME, ADDRESS & E-MAIL

PHONES

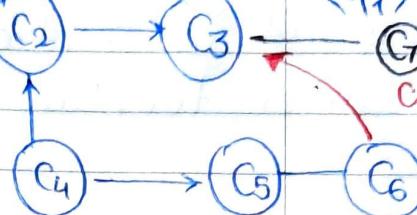
MOBILE

* * Merge Conflict

Main



Branch.



< Html >

< Head >

< T1 > Ganesh < /T1 >
G7 committed (merged)

Conflict whoes title to put

< Html >

< Head >

< T1 > Akshay < /T1 >

<<< Conflict resolution markers --> denote Conflict.

in example <<< Head >>> in main look like

>>> >>> in branch look like

Branch Management

When you are not Confident made one branch & Work on

git branch -v ---> get last message on each branch

git branch --merged ---> already merged branches

git branch --no-merged ---> show not merged branch

deleting Branch

git branch -d develop _{<Branch name>} ⇒ gives error if branch not merged

To avoid accident delete:

git branch -D _{<Branch name>} ⇒ without error mess.

Branch delete

NAME, ADDRESS & E-MAIL

PHONES

MOBILE

Branch Type

1) Long Running Branches

(2) Topic Branches

In long running Branches Have main branch & other Sub branches running until project get end.

Topic Branch

For a particular change in core we create this Topic Branch, in that we run codes & once effective result get we either merge in main branch or just delet it.
example:- Adding new extensions.