- Git

1) Version Control system is a tool that helps to track to changes/history in code

2) Git helps in collabration (2 or more people works on same project, It helps for managing the code)

3) Git is a Version Control system.

4) It is Popular, Freee and open Source, Fast and scalable

* When we want to ignore any file from github, the

1) create “.gitignore” file

2) write the file name in “.gitignore” file that we want to ignore form github.

- Commands for Git

1) git config --global user.name “[firstname lastname]” =====> set a name that is identifiable for credit when review version history.

2) git config --global user.email “[valid-email]” =====> set an email address that will be associated with each history marker

3) git config --list =====> Shows what we have changes

- Github

1) Website that allows developers to store and manage their code using Git.

- commit ===> save the changes in the code.

- modified ===> changes in code, but not in github repo's code is known as modify

- Untracked ===> new files that git doesn't tract (When we create new file and doesn't upload commit on repo is a type of Untrack.)

- Unmodify ===> unchange file

- staged ===> When we add (using "git add ." or "git add [filename]") the the file but not commit then the file is in Staged status.

Q) What is README.md and how to change it?

--> 1) README.md ===> README.md is a file with extension of .md

- in this file, we can write notes about that project.

2) steps to changes README.md file

i) click on README.md file

ii) edit the notes

iii) click on "Commit changes" button which is top-right-button.

iv) write a short not on "Commit message" section

v) click on "Commit Changes" button.

- Imp basic git Commands

1) git clone "repo URL" =====> When we want to open any repo which is in any githup account in VSCode the we use this Command

2) cd =====> change directory

cd .. =====> back from currect directory/folder

clear =====> clear the terminal

ls =====> shows the list of files in repo

ls -a =====> shows all/hidden files in repo

mkdir [folder/directory-name] ========> for create new directory/folder

3) git status =====> shows status.

- imp Add & Commit git Commands

1) Add =====> (git add [file-name] ) =======> add specific file.

=====> (git add .) =======> add all files.

- adds new or changed files in your working directory

- add a file as it looks now to your next commit (stage)

- If we want to commit the file the the first stape is add the file using command (git add [file-name] ) and then we can commit using (git commit -m "some-message")

2) commit =====> (git commit -m "some-message")

- It saves the changes in git repo.

- If we want to commit the file the the first stape is add the file using command (git add [file-name] ) and then we can commit using (git commit -m "some-message")

3) git log ====🡺 (shows the list of all the commit)

- imp Push Command

1) push =====> git push origin main

- Upload local (personal PC) repo content to remote (githup) repo.

- Personal pc's folder to github repo managment.

1) init =====> (git init)

- used to create git repo.

2) git add .

3) git commit -m "[message]"

4) git status

5) create repo manually

6) git remote add origin ["repo-URL"] ====> The git remote add origin ["repo-URL"] command is used to link a local(personal pc) Git repository to a remote(githup) repository.

7) git branch ====> (to check branch)

8) git branch -M main ====> (to rename branch)

9) git push -u origin main

- steps of new file add in same githup repo

1) create new file

2) git add .

3) git commit -m "[message]"

4) git push

- flow of githup repo to pc

1) githup repo

2) clone

3) changes in code

4) add

5) commit

Branch in Github

Shiam (Navbar) Ruchi (Footer)

232 Lines code

132 Lines code

32 Lines code

Main 1 copy

232 Lines Code

132 Lines Code

Copy copy copy

Main 2 Main 3

या ठिकाणी main1 ला 32 lines cha code आहे. तर xyz company मध्ये Shubham नावाचा developer ला Navbar design करायला सांगितले. त्याने ते design केल्यावर total 132 lines cha code झाला. त्यानंतर Ruchi नावाच्या developer ला footer कहा code design करायला सांगितलं. तिने तो code design केल्यानंतर तो code 232 lines कहा झाला.

त्या नंतर Shivam an ruchi यांची code file चा code main file मध्ये merge होतो. आणि त्यांची file remove होते.

मग main file अशी बनते

Main (This time)

232 Lines Code

132 Lines Code

32 Lines Code

* Basic commands of git branch

1. git branch ====🡺 to check branch
2. git branch -M main ====🡺 to rename branch

====🡺 here main is branch name/ we can change the branch name also.

1. git branch [branch-name] / git checkout -b [branch-name] ====🡺 to create new branch
2. git switch [branch-name] / git checkout [branch-name] ====🡺 for switching main branch to other branch

#Note ====🡺 जर आपल्याला branch switch करायची असेल तर सर्वात अगोदर आपल्याला main branch ला स्विच व्हाव लागेल. आणि मग आपण दुसऱ्या कोणत्याही branch ला switch होऊ शकतो.

1. git branch -d [branch-name] ====🡺 delete the specific branch.

#Note ====🡺 जर आपल्याला xyz branch delete करायची असेल तर सर्वात अगोदर आपल्याला main branch ला स्विच व्हाव लागेल. आणि मग आपण दुसऱ्या कोणत्याही branch ला आपण delete करू शकतो.

1. git diff [branch-name] ====🡺 आता सध्याला ज्या बरंच मध्ये आपण आहोत तो बरंच आणि दिलेल्या command मधल्या branch name मधील branch कहा difference सांगतो.
2. git pull origin main ====🡺 githup repo मधून आपल्या personal pc मध्ये code ला download karun आपल्या VSCode मध्ये आणते.
3. git merge [branch-name] ====🡺 merge 2 branches .

Merge Conflict In Branching

branch1 branch2

33 lines code

line 33 code is var a = 12;

32 Lines code

33 lines code

32 Lines code

main1 line 33 code is

var c = 15;

main2

या ठिकाणी main मध्ये 32 lines चा code आहे. तर आता आपण त्याचे branch तयार केले. तर आता branch2 मध्ये आणि main2 मध्ये line no 33 ला वेगवेगळे कोड लिहिला आहे. तर आता git confuse होत आहे की कोणता code main मध्ये merge करावा.

यावर आता git conflict/exception देईल. 1) yellow line accept करा. 2) green line accept करा. 3) yellow आणि green दोन्ही line accept करा.

* How to handle Merge conflict

1. git commit [branch-name] ====🡺 2 branch marge hotel पण त्यात 3 option येतील a) accept current change b) Accept incoming change

c) accept both changes

- जो code पाहिजे त्या option वर click करायचे

* Other imp command in github

1. Undoing changes =🡺 Undoing changes म्हणजे आशे changes की जे add तर झाले आहेत परंतु कम्मिट नाही झाले.
2. git reset [file-name] / git reset ====🡺 जेव्हा आपण file add करतो पण commit karat नाही आणि काही करणास्त आपल्याला परत file back करायची असेल त्या वेळेस ही command वापरता येते.

- git reset [file-name] ====🡺 जेव्हा specific file रेसेट करायची असेल तेव्हा ही कमांड वापरता येते.

- git reset ====🡺 जेव्हा एक वेळेस अनेक files reset करायच्या असतात त्या वेळेस ही कमांड use करता येते.

B) commited changes (for one commit) ====🡺 जेव्हा आपण चुकीच्या code ला चुकीने commit करतो आणि आपल्याला परत back यायचे असते तेव्हा आपण ही command use करू शकतो

1) git reset HEAD~1 ====🡺 जेव्हा आपण एकाच फाइल मध्ये 4 5 वेळेस commit करतो परंतु काही कारणास्तव आपल्याला एक commit back यावे लागू शकते त्या वेळेस आपण ही command use करू शकतो.

2) get reset [commit-hash-code] ====🡺 जेव्हा आपल्याला मागच्या कोणत्याही commit ला जायचे असेल त्या वेळी आपण त्या commit चा hash-code copy करून command मध्ये paste करतो. म्हणजे आपण त्या commit च्या code ला back जातो.

- hash-code माहीत करून घेण्यासाठी (git log) command use करा.

1. Fork (setting) =====🡺 जेव्हा आपल्याला दुसऱ्या व्यक्तीच्या किवा दुसऱ्या company च्या githup च्या code आपल्या git वर आणायचा असेल किवा त्यावर काम करायचे असेल त्यावेळेस आपण fork (github ची एक setting) करत असतो. Fork केल्यावर तो repo आपल्या git मध्ये open होतो.

* Steps for fork

1. Search the repo, that we want to work on github
2. Open that repo
3. Click on “Fork” button.
4. Put repo name
5. Click on “Create Fork” button.
6. Successfully open that repo in out github.