GIT AND GITHUB

GIT - is a **version control system(VCS)** - To kind of have/hold to the version changes that we did in the code and save the project for future use purposes we use the VCS called GIT.

- It includes a .git directory, which contains metadata and information about all branches, commits, remotes, and configurations.
- The repository can have multiple branches, including the default branch called "master" (or "main" in more recent Git versions). But the repository itself is not a branch.

Branch : are specific lines of development within the repository and the default branch created when you initialize a new repository is called the **Master branch**

GIT is a local repository / in your computer .

git init - means to **initialize an Empty git repository** - creates a repository in the project working directory folder .

git add <filename> - used to add any new files before commit one by one selectively.

git add . - used to add all the files once at a time before the commit.

git commit - is a **confirmation command** for the changes we make into the local repository / we can say that it saves the changes permanently to local Repository

git status - used to check the **current status** of the master branch .

git branch - to check which branch the current local repository is working on .

git branch -a To list out all the local branches .

git checkout - The git checkout command is used to switch branches but **git checkout -b <new branch>** means jump to branch while creating a new one **or** to jump for existing one we use **git checkout <branch_name>.**

git branch -d <branchname>- To delete the branch specifically.

git branch -M <branch name> - used to rename the branch we wish to do .

Note : **untracked files** are the files that needs to be added into the repository and basically they are the files present outside the repository in the working directory

Staging area : means that is **present in between local and working directory - used to hold particular number of files/selective files** from working directory and from Staging we can send to local repo **whereas from working directory this kind of task is not possible -** It commits all the files to local repo

what happens if we use only add and don't commit - Answer is that it goes to the staging area and waits for the commit.

What is fork ..?

Fork is a copy of a repository , forking a repository allows you to freely experiment on the project without affecting the actual project.

git clone - is a command used to create a local working copy of a remote repository . It downloads the remote repository to the computer - **git clone <remote_URL>.**

git pull - is a command used to fetch and merge the changes made in remote repository to the **local repository** - **git pull <remote URL> main**

git push -u origin master /git push origin main -is used to transfer the commits or pushing the content from the local repository to the remote repository ,that means local repository has been modified, and the modifications are to be shared with the remote team members. **Here -u is used to set upstream which means once we use this upstream then we can simply use git push from next time onwards without mentioning the remaining part(origin main).**

git remote add origin <remote repository URL> - is used to tell the git to establish connection to remote repo in advance and **whatever we push the changes from now on , it will go above remote repo URL.**

git remote -v : is used to check which remote repo we are thinking of as a origin from now on .

Note : Here origin is the default , convention name of the main remote repository from which we have cloned the project .

Two ways merge the code into the master branch(deployable branch they call it)

Way 1 : Through the git using the below commands.

git diff <branch name> - **used to compare the files in 2 branches before merge** to get an idea of what extra files or code we have and that's not there in another one.

git merge <branch name> - used to merge the 2 branches for example if we are in dev1 branch currently and we are looking to merge it into main branch then we use **git merge main**

Way 2 : **Create a PR (Pull request) in github** and those who have authorization, for example **senior developer or team lead will check the PR** by reviewing it and approve the merge request to the main branch/master branch.

After Installing git into your computer - we need to create a local repository / repo .

Repository - usually means a place where you save your project files / It's a project folder.

Github - **is an example of remote repository** hosting service /saving service like Bitbucket , GitLab , beanstalk , AWS code-commit etc ,which allows developers to store and manage the code .