❖ GIT CLONE Vs GIT FORK:

- Any public Git repository can be forked or cloned.

GIT CLONE:

- Git clone creates a linked copy that will continue to synchronize with the target repository.
- When you clone a repository, you are creating a local copy on your computer that you can sync with the remote on GitHub.

GIT FORK:

- Forking creates a new repository under your account on the hosting service, allowing you to work independently of the original project.
- A fork is a copy of a repository that allows you to make your own changes without impacting the original project.

Fork→clone→make changes→create pull request.

❖ GIT FETCH & PULL:

GIT FETCH:

- Download objects and refs from another repository.
- Git fetch downloads only latest changes into the local repository. It downloads fresh changes that other developers have pushed to the remote repository
- since the last fetch and allows you to review and merge manually at a later time using git merge. Because it doesn't change your working directory or the staging area, it is entirely safe, and you can run it as often as you want.

GIT PULL:

- It downloads latest changes into the local repository and it also automatically merges change in your working directory.
- One important thing to keep in mind is that it will merge only into the current working branch.

  git pull = git fetch + git merge

❖ MERGING VS. REBASING

- Rebasing and merging are both designed to integrate changes from one branch into another branch in Git, but they accomplish this in different ways.

GIT MERGE:

- It combines changes from one branch into another, creating a unified history of commits.
- It allows developers to merge Git branches while the logs of commits on branches remain intact.

GIT REBASE:

- It is a command that lets users integrate changes from one branch to another. Once the action is complete, the logs are modified.
- Rebasing in Git integrates a change from the base of the feature branch to the master branch's endpoint. It's useful for streamlining complex histories.