**GITHUB NOTES**

**REPOSITORY:**

Repositories is like a folder. We can store the files there and can set a name for that repository.

We can create a repository with simple steps.

1. Repository Name
2. Description
3. Public or Private
4. Add a ReadMe file if you wish. (Optional)

**COMMIT:**

Commit means saving the current state. After editing some codings or corrections in the file, we will have to give the commit message and description.

I have given some examples below.,

Audio-Feature-Implemented

Video Feature Updated

CHAT AI bot integrated

Etc..

**BRANCHES:**

Branch means that it is possible to create a separate backup like a main branch. After creating a new branch, we can edit or update the coding’s or files there. It won’t affect the main branch. Mainly it is used for create a new feature, bug fixing, etc..

So that main branch won’t affect. After completing the bug fixing, new feature implementation in the new branch, you can merge the new branch to the main branch by using the pull request.

You will see that option `pull request` after editing the code in the new branch.

**FORK:**

Fork is an option to get the exact copy of the project (main branch or another branch) from an account to your account.

For example An organization has a project and it has a main branch. Now you need to work on that project. So you can fork that branch (main branch). When you fork, it will be copied to your profile.

Then you can start to do the work in that project which is in your profile (your account).

After done your work, you can send a pull request to the main branch of the project.

If it is helpful and no conflicts, they will merge the code with the main branch.

For above purpose, Fork is used.

**PULL REQUEST:**

Before we understand the PULL request, we have to remember the FORK. For working purpose, we have forked a branch (set of codes) from the main branch. (organization profile or any other profile).

Now we have done that coding work in our profile which was forked. Now we are going to send a request to the main branch to merge our changes in the coding’s (from where we forked the project).

This process is called a PULL request. It is simply giving the request to the main branch to merge our codes in their project (from where we forked the project).

**MERGE:**

Before knowing the MERGE, we need to remember the PULL request. We have edited the coding’s in our profile and given the PULL request to merge our code. We have forked the project from the original author. (An organization profile or another user profile).

Now they have to merge the codes in their main branch (original profile). After we sending the pull request, they will check whether any conflicts found or not in our code, then they will merge the codes in their main branch. This is called MERGE or Merging.

**MERGE CONFLICT:**

Generally, many developers will work on the same project. So each developer will create a separate module. After completing those modules, everyone will give the pull request. So now the person who manages the main branch, will merge those developer codes one by one into the main branch.

That time there is an opportunity to produce an error `Merge Conflict`.

It means the developers (one or more) has written the codes for the same line of the file.

Coding Line No: 9 => Developer A and Developer B both has edited the coding’s in that line. So this time only, merge conflict error will appear when trying to Merge the coding’s into the main branch.

So now, we will have to check the coding. After review the coding’s we have to take a decision there which one we should keep and which one needs to be removed.

After doing the corrections in the coding’s, there will be no error like merge conflict. After that we can start to merge the codes into the main branch.

This whole process is called Merge Conflict Issue fixing.