Melvin Vazquez

3/7/24

**GitHub Essay**

GitHub is a website where developers can go to create, store, and manage code that they publish on the website. Developers can have teams that work in a repository that can be public or private and they would all be able to work on the code. It is a website that developers of all levels use to publish code and those that want to can visit other people’s public repositories and have the options to fetch, commit, push, or merge from their repositories. All of these options do different things and depending on what the developer is trying to do, they can choose one of those options to achieve their goal.

The first option they can choose is to fetch from a remote repository, which basically downloads the commits, files, and refs from a remote repository. This is something that is usually done when you want to see what everyone else is doing. One good thing about this command is that the content must be checked by a git checkout command, so it makes this a safe way to review commits before deciding whether to integrate into your local repository. On the other hand, we have another option called Push that is used to upload local repository content to a remote repository. This is a counterpart to fetch because this is transferring commits from your local repository to a remote repository, which brings about an issue in terms of the push potentially overwriting changes.

A similar action to fetch is the git pull that is used to download content from a remote repository and use that to update the local repository so that it matches the content of that download. This action is a combination of git fetch and git merge and follows the order in which git pull will enter a merge workflow once content is downloaded and a new merge commit will be created. To continue from that, git merge is another option that developers can use when they need to combine multiple sequences of commits into one. Usually, it is used in combining branches, which are pointers to changes you have made, and trying to find the common base commit between them. However, git merge won’t always work because of a conflict found during the merge or at the start of a merge. These are called merge conflicts, and these are caused when two developers are editing the same content, and they aren’t working in separate branches. These conflicts are only shown when a developer is trying to merge and git marks the file as conflicted, which then becomes the developer’s responsibility to resolve the merge conflict.

Another option that people can use is commit, which is similar to how a person saves in a word document, or in a video game. Every time you finish editing or adding to your code, you will want to commit to your branch so that it records the changes and gives it unique IDs. These IDs help to figure out what changes were made, when they were made, and who created the changes in the file. Lastly, the place that these commits will go to is to a repository that the person has created. A repository is like storage for the developer because it is a place to store your code, your files, and any changes that were made since the creation of that repository. A repository can also have multiple owners, if you are working in a team of developers, or you can have full ownership of the repository that is for all your private or public projects. You also have the freedom to give people access by giving them permission, and you can also use repositories to collaborate with others by collecting feedback on issues in your code, or use the pull request to propose a change to a repository. Plus, the GitHub discussion is a great way to share information with others or communicate about anything else relevant to you.