Github essay

Github is a great workflow space for all levels of programmers. It is very useful in many ways and convenient for teamwork projects or if you need a place to store and organize your programs Github is a great place for storage. They have a wide variety of IDE’s (Integrated Development Environments) so programmers can use any programming language they prefer. You can also work directly in Github as well. Every time you make a change, like adding a specific line of code or even making notes throughout the program to save or remind yourself about something for later, it will save your work at that moment.

The teamwork aspect of it gets even more convenient. You are able to share a project and give special permission to certain people that you can choose from, almost like forming your own team. While giving certain permissions and control to certain people, you can assign them jobs. For example, you put a big chunk of code on the project board, you can assign someone to test the code, someone else can make edits to the code, another person can scan the code and make notes to each line of code explaining what everything does. You can split the workload evenly and as you please. This makes working on projects way more efficient as ever.

Sometimes though, there could be something called a merge conflict when trying to update the project at the same time. For example, you and another person working on the project both try to update a certain line of code at the same time, github will prevent it from updating because both forms of the new project do not match each other which would cause a merge conflict. A way to resolve an issue like this is to communicate as a team and to decide how to fit the two different updated pieces together.

All of this is possible through the interface called a repository. A repository is the place where all your projects and work is stored. The fascinating part about the repository is that it updates in real time. So if someone in the group is working overtime on one part of the project, you'll be able to wake up the next morning and see where he left off automatically. It makes it convenient for people who are part of the team to work on their own time. Along with this there are other cool features that github implemented to ensure creativity and efficiency when working on your special projects. A neat implementation is called “branching”. Branching is a way of making a copy of your “project” or a program to test certain features out. For example, if you wanted to branch a specific program to make changes to it you can. You can alter it to your liking with no limitations and see how it will test out and change without affecting the original project. This is almost like a risk free implementation to the project so that you won't have to backtrack completely if there is a mistake or a bug somewhere along the way.

There are also features like push and pull requests. This can be useful in a way where if you have a team working on a big project, you can make specific changes that you were assigned and so that everyone is on the same page, you don't automatically put your update though the main project without anyone knowing. You can send a request to the team and everyone would be able to look it over and if everything seems to align then it will get pushed into the main project. This keeps everyone up to date on any new changes that have just happened. This also avoids merge conflicts and communication conflicts as well. This is called a push or a pull request. Another implementation called commit or commit messages plays a big role in team communication as well. This is a way of pushing out a update to the project with confirmation and a commit message is like leaving a note on a specific part of the project. For example, if someone is working after hours and there is no need to have a pull or push request. The rest of the team can see the logs of the project where github stores the “commits” that have been done to the project so that team members can keep track when certain changes have been done. This is another very useful feature provided by github.

Github is a very convenient, professional and in many cases a very good social platform for programmers to do what they love and do it efficiently. Many people can create special projects whether it's teamwork or solo. There are many opportunities that github provides. It can also be seen as a resume in the programming world. I will definitely be taking use of the github platform as time moves forward !