Jason Chandler

GitHub Essay

GitHub is a common professional tool programmers use to collaborate on code. It is an essential non-coding skill; every programmer should get familiar with it. It allows coworkers or academic group project members to easily share or modify their projects. Moreover, GitHub is home to many open-source projects, whose source code is public — anyone can use, view, or modify it (Hanna, 2021). Users can build atop and modify the code as they see fit, whether to create a better derivative or to boost their resume (Hanna, 2021). Some examples of open-source projects are the Linux operating system and the Apache HTTP server (Pratiyush, 2024). Another example is the basis for GitHub — the Git version-control system.

Linus Torvald, the same man who created Linux in the 1990s, created Git because he was frustrated by the version-control systems of the time (Baker, 2023). Most were centralized systems, like Subversion, which made it hard for multiple people to work on the same project remotely or concurrently. So in 2002, he created Git; in 2005, he and his Linux kernel developers started using it (Baker, 2023). The system became widely used by the open-source community in 2008 — when GitHub was created.

Ironically, although Linus developed Git with the intention of making coding easier and more collaborative, it was not popular upon creation — until GitHub launched in 2008. GitHub made Git extremely accessible, leading to its dominance in the code-sharing space. It hosts Git repositories, collections of files that make up a project. Those files can be anything, but Git repositories are designed specifically for code.

GitHub is incredibly convenient for modern programmers; it gives them a space to collaborate. These spaces, called repositories, are essentially virtual file systems with enhanced version control (Baker, 2023). Contributing to one can be very simple; the user does not have to interact with the terminal or command lines if they do not want to. Users can access, modify, or push code directly in the browser if they wish.

There are a few ways to contribute to an open repository and some keywords associated with doing so. Users can use the Git client and command lines to do so. However, it is much easier in the browser: It does not require users to clone the repository locally. Users navigate to the repository they wish to contribute to and click the Fork button; it looks like a fork in a road with circles at all ends. This Clones the repository into their account; users can directly upload new files or edit existing ones here. When finished, they must Commit the changes to save them to their account's version. Finally, the user must submit a Pull Request to the original repository to officialize the changes. The repository’s maintainers will review the code and approve or deny the request, likely leaving feedback.

Users do not need to contribute to the main Branch. They can create a new Branch and use that to modify the repository, allowing them to edit specific aspects of the code without disrupting anything else. It decreases the risk of destabilizing the main, functioning branch with new, possibly experimental code. A user can Merge branches by going to the Pull requests tab, selecting New, and selecting which branches to merge. Sometimes, Git will have trouble distinguishing which line of code to keep. It marks these as Merge Conflicts. The user must manually resolve them. They are usually the result of two branches modifying the same line of code, so the user must pick which one to keep.

For Project One, I first set the framework of my repository. I added a folder for each specific assignment. Then, I updated an important aspect of my repository: The README.md file. GitHub offers to automatically create it when users make a new repository, as it is crucial for describing the layout, download and commit instructions, and project goal. It is a markdown file, a lightweight markup language used to easily format plain text (Markdown Guide). It uses simple characters such as the pound sign (#) for headers and double asterisks (\*\*) for bold text (Markdown Guide, Headers/Bold).