A Beginner’s Guide to Git and GitHub

1. What is Version Control?

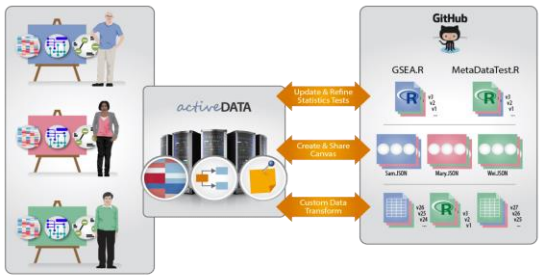
Version control, also known as **source control**, is the practice of tracking and managing changes to software code. It allows multiple contributors to work on a project without overwriting each other's changes. Version control also stores a complete history of modifications, so developers can easily roll back to previous versions if needed.

why version control is essential for collaboration and for gaining public trust.

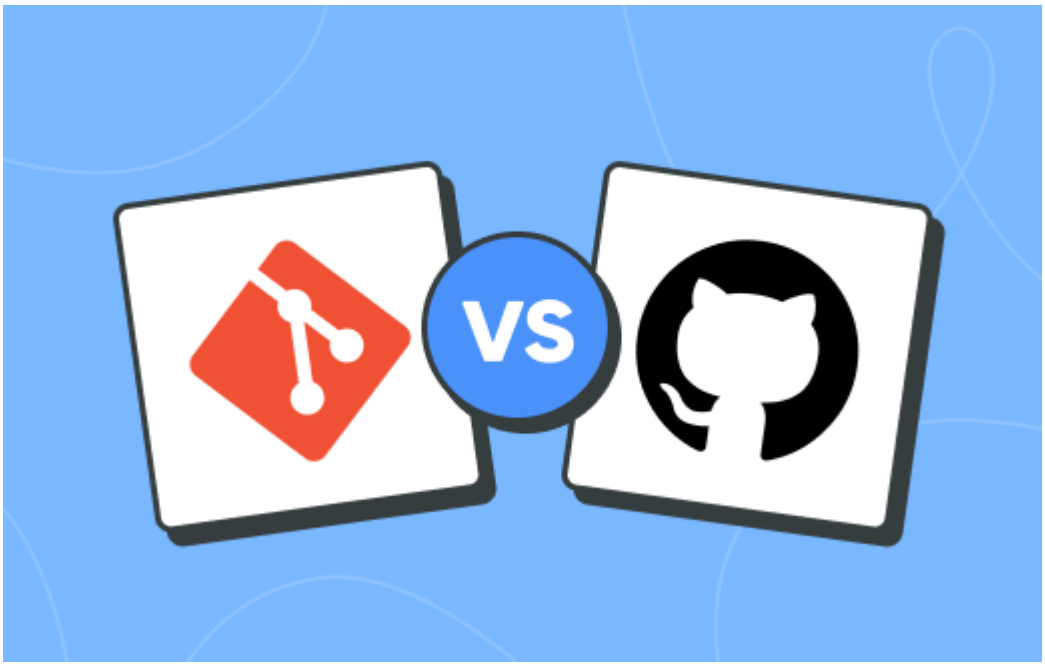
Version control enables multiple contributors to work simultaneously without interfering with one another by using separate branches that are later merged in a controlled manner. All changes are saved as commits, ensuring no work is lost and allowing teams to safely revert to previous versions when errors occur. By clearly recording who made each change, when it was made, and why, version control promotes accountability and reduces confusion. In addition, structured review processes allow teams to discuss, refine, and approve updates before integration, resulting in faster collaboration, fewer conflicts, and higher-quality outcomes.

Why Version Control Builds Public Trust

Version control systems enhance public trust by ensuring transparency, as the complete history of a project is visible and traceable. They support reproducibility by allowing others to access and use exact versions of data or code, which is crucial in research and public reporting. The immutability of recorded changes protects the integrity of work by preventing hidden alterations. Furthermore, open platforms such as GitHub enable independent inspection, testing, and improvement, strengthening confidence through verifiable and accountable practices.



3)Git vs GitHub



Git

Git is a distributed version control system used to track changes in source code during software development. It helps developers manage code efficiently while supporting fast and flexible collaboration.

- Designed for speed, data integrity, and reliable version tracking.
- Supports distributed, non-linear workflows for teams and individuals.

GitHub

GitHub is a web-based platform that hosts Git repositories and enables developers to store and manage code online. It extends Git's core version control capabilities with powerful collaboration features.

- Provides distributed version control and source code management.
- Adds features like pull requests, issue tracking, and team collaboration.

- Below is a table of differences between Git and GitHub:

Git	GitHub
Git is a software.	GitHub is a service.
Git is a command-line tool	GitHub is a graphical user interface
Git is installed locally on the system	GitHub is hosted on the web
Git is maintained by Linux.	GitHub is maintained by Microsoft.
Git is focused on version control and code sharing.	GitHub is focused on centralized source code hosting.
Git is a version control system to manage source code history.	GitHub is a hosting service for Git repositories.
Git was first released in 2005.	GitHub was launched in 2008.
Git has no user management feature.	GitHub has a built-in user management feature.
Git is open-source licensed.	GitHub includes a free-tier and pay-for-use tier.
Git has minimal external tool configuration.	GitHub has an active marketplace for tool integration.
Git provides a Desktop interface named Git GUI.	GitHub provides a Desktop interface named GitHub Desktop.

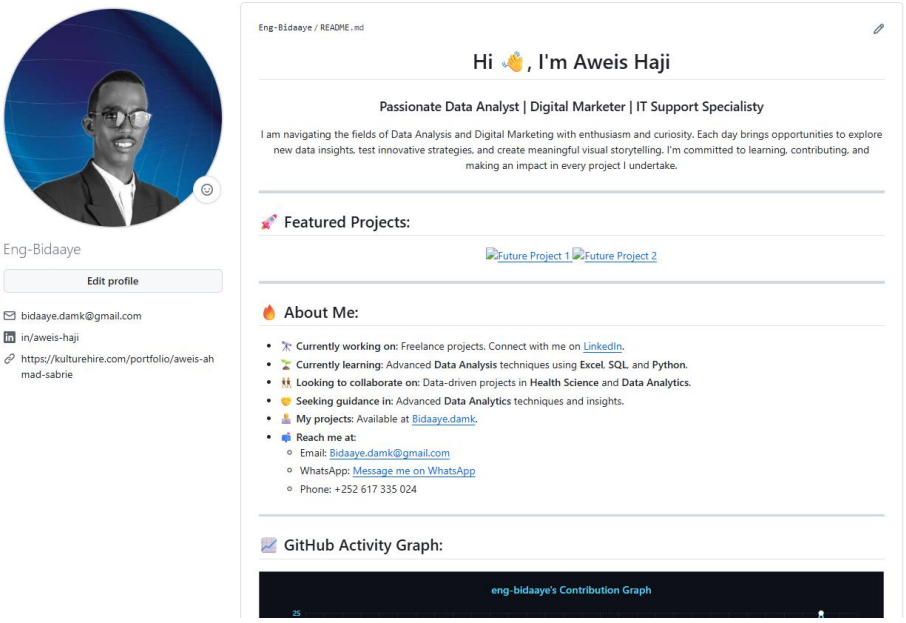
```
Eng-Bidaaye@DESKTOP-1QKR9RH MINGW64 ~
$ ^[[200~git init
bash: $'\E[200~git': command not found

Eng-Bidaaye@DESKTOP-1QKR9RH MINGW64 ~
$ git config --global user.name "Eng-Bidaaye"
git config --global user.email "bidaaye.damk@gmail.com"

Eng-Bidaaye@DESKTOP-1QKR9RH MINGW64 ~
$ mkdir my-first-project
cd my-first-project

Eng-Bidaaye@DESKTOP-1QKR9RH MINGW64 ~/my-first-project
$ git init
Initialized empty Git repository in C:/Users/Eng-Bidaaye/my-first-project/.git/

Eng-Bidaaye@DESKTOP-1QKR9RH MINGW64 ~/my-first-project (master)
$
```



The main differences between local and remote repositories are:

Local Repository

- **Location**: Stored on your local machine.
- **Access**: You can access and modify it without an internet connection.
- **Usage**: Used for development, testing, and making changes before pushing to a remote repository.
- **Commands**: You can use commands like git commit, git branch, and git checkout directly on your local repository.

Remote Repository

- **Location**: Hosted on a server (e.g., GitHub, GitLab, Bitbucket).
- **Access**: Requires an internet connection to access and modify.
- **Usage**: Used for collaboration, sharing code with others, and backing up your work.
- **Commands**: You use commands like git push to send changes from your local repository to the remote, and git pull to fetch updates from the remote repository.

In summary, the local repository is where you work on your project, while the remote repository is where you share and collaborate with others.