**GIT**

Git is a distributed version control system that allows developers to track changes in code, collaborate with others, and manage different versions of the projects.

**Version Control:**Git tracks changes in project files, creating a snapshot of the code at different points in time. Developers can commit changes to their local repositories, creating a version history.

**Branching:** Git allows for the creation of branches, which are separate lines of development. Developers can work on different features or fixes in parallel without affecting the main codebase. **Merging:** Changes made in different branches can be merged back into the main branch

**GITHUB**

GitHub, is a web-based platform that provides hosting for Git repositories. It adds a user-friendly interface to Git, making it easier to collaborate on projects and manage code repositories.

**Remote Repositories:** GitHub hosts Git repositories remotely, making it easy for developers to share and collaborate on projects. Developers can clone repositories from GitHub to their local machines and push changes back to the remote repository. **Pull Requests:** Developers can propose changes to a repository by creating a pull request. Pull requests allow for code review, discussion, and collaboration before changes are merged into the main branch. **Issues and Projects:** GitHub provides tools for issue tracking and project management. Issues can be used to report bugs, request features, and discuss improvements. Projects help organize and manage tasks, providing a visual overview of the work being done. **Collaboration Features**: GitHub includes features such as wikis, actions, and discussions to enhance collaboration.

**GIT COMMANDS**

**git config --global user.name "Your Name":** Sets your name for all repositories.

**git config --global user.email "you@example.com":** Sets your email for all repositories**.**

**git init** :Initializes a new Git repository, creating a hidden subfolder within the project that houses the internal data structure required for version control.

**git add <file>**: Adds changes in the working directory to the staging area.

**git commit -m "Commit message"**: Commits the staged changes to the local repository.

**git status**: Shows the status of changes as untracked, modified, or staged.

**git log**: Displays the commit history.

**git diff**: Shows the differences between the working directory and the staging area.

**git branch <branch\_name>**: Creates a new branch.

**git clone <repository\_url**>: Creates a copy of a remote repository on your local machine.

**git checkout <branch\_name>**: Switches to the specified branch.

**git merge <branch\_name>**: Merges the specified branch into the currently checked-out branch.

**git remote -v**: Lists all remote repositories associated with the current repository.

**git remote add <name> <repository\_url>**: Adds a new remote repository.

**git push <remote\_name> <branch\_name>**: Pushes changes to a remote repository.

**git pull <remote\_name> <branch\_name>**: Fetches and merges changes from a remote repository.