Table of Contents

[What is Git? 2](#_Toc169725051)

[Key features of Git include: 2](#_Toc169725052)

[What is GitHub? 2](#_Toc169725053)

[Key features of GitHub include: 2](#_Toc169725054)

[Getting Started with Git and GitHub 3](#_Toc169725055)

[Alternative (Terminal): 3](#_Toc169725056)

[Benefits for Students 4](#_Toc169725057)

Understanding Git and GitHub: A Student's Perspective

Git and GitHub are indispensable tools for modern software development. They enable developers to track changes in their code, collaborate efficiently, and manage their projects with ease. As a student venturing into the world of coding, understanding these tools is crucial for both personal and professional growth.

# What is Git?

Git is a distributed version control system developed by Linus Torvalds in 2005. It allows multiple developers to work on a project simultaneously without overwriting each other's changes. Git keeps a detailed history of all changes made to the code, enabling developers to revert to previous versions if necessary. This makes it easier to experiment with new features or fix bugs without the fear of losing previous work.

## Key features of Git include:

* Branching and Merging: Developers can create branches to work on new features independently and merge them back into the main codebase once they are ready.
* Distributed Development: Each developer has a full copy of the repository, including its history, on their local machine. This ensures that work can continue even if the central server is down.
* Tracking Changes: Git records who made changes and when, making it easy to track progress and understand the context of changes.

# What is GitHub?

GitHub is a web-based platform that uses Git for version control. It provides a collaborative environment where developers can host their repositories, share code, and collaborate on projects. GitHub enhances Git's functionality with features like issue tracking, project management, and a user-friendly interface.

## Key features of GitHub include:

* Repositories: These are storage spaces for projects. Developers can create repositories to host their code, documentation, and related resources.
* Pull Requests: This feature allows developers to propose changes to a codebase. Other team members can review the changes, discuss potential issues, and merge the changes if approved.
* Forking: Developers can create their own copy of someone else's repository to experiment with changes without affecting the original project.
* GitHub Pages: This feature allows users to host static websites directly from their repositories, making it easy to share project documentation or personal portfolios.

# Getting Started with Git and GitHub

To begin using Git and GitHub, follow these steps:

* Install Git: Download and install Git from [https://git-scm.com/downloads.](https://git-scm.com/downloads.%20) Verify the installation by running git --version in your terminal.
* Create a GitHub Account: Sign up for a free account on GitHub.com.
* Configure Git: Set up your Git username and email on browser.
* Create a Repository: On GitHub, click the "+" icon and select "New repository." Name your repository and initialize it with a README file.
* Clone the Repository: Clone your repository to your local machine using Github Desktop (<https://desktop.github.com/>) as it is easy to follow through the process rather then the terminal process.
* Make Changes and Commit: Add your code files to the repository, then stage and commit your changes.
* Push to GitHub: Push your local commits to the GitHub repository.

## Alternative (Terminal):

* Install Git: Download and install Git from [https://git-scm.com/downloads.](https://git-scm.com/downloads.%20) Verify the installation by running git --version in your terminal.
* Create a GitHub Account: Sign up for a free account on GitHub.com.
* Configure Git: Set up your Git username and email on browser.

git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

* Create a Repository Locally:In the terminal, navigate to the directory where you want to create your project, or create a new directory:

mkdir my-project

cd my-project

git init

* Add a README File:Create a README file to provide an initial file to commit:

echo "# My Project" > README.md

* Add Files to the Repository: Add your README file to the staging area:

git add README.md

git commit -m "Initial commit with README"

* Create a New Repository on GitHub: Create a new repository on GitHub using the GitHub CLI. First, install GitHub CLI from cli.github.com.

gh auth login

* Create a new repository on GitHub:

gh repo create my-repository --public --source=. --remote=origin

* Replace my-repository with your desired repository name.
* Push Changes to GitHub: Push your local commits to the GitHub repository:

git push -u origin main

# Benefits for Students

Learning Git and GitHub is highly beneficial for students. These tools are industry standards, and proficiency in them can significantly enhance employability. They also foster good coding practices, such as version control and collaborative development, which are essential skills for any software developer. Additionally, GitHub provides a platform to showcase projects and contributions, helping students build a professional portfolio.