

Git and GitHub

What is Git?

Git is a tool (Software), it helps you track changes in your code @ project

- * It remembers every change you make to files in a project
- * If something break, you can go back to an older safe version.
- * If many people work together, Git helps avoid mixing up each other's work.
- * You can use the Git without Internet

without Git

- * You keep many copies:- project-final, project-final2, final-latest-really.
- * You forget what changed and when.

with Git

- * One clean folder.
- * Clear history:- who changed what, when, and why
- * You can try new ideas safely and keep the main work clean.

Example:-

College project without Git

- * Ram is making the frontend (UI) → takes 20 days
- * Seetha is making the Backend (Database + Logic) takes 30 days.

But if they don't use Git:-

Ram finishes his part in 20 days.

Then Seetha starts her part and takes 30 days.

total time = $20 + 30 = 50$ days.

Same project with Git

Both Ram and Seetha work at the same time.
Git allows them to:

- * Work on different parts of the same project.
- * Save their changes separately.
- * Later, Git can merge (combine) the changes into one project.

total time = 30 days

What is GitHub?

* GitHub is a website (like a social media for code).

* It stores your Git projects online (on the cloud).

* With GitHub you can:

- * Share your project with others.
- * Work together with your team.
- * Show your project to the world.
- * Back up your code safely (if your laptop crashes, your project is still on GitHub).

Git

Git is a distributed version control system.

- * Version Control System means Git records all the changes made to a project and lets us go back to any version when needed.
- * Distributed means instead of only one central server, every developer has a full copy of the repository, including its complete history, on their local machine.
- * This allows developers to work offline collaborate in parallel, and reduce the risk of data loss since the history exists on multiple systems.

Commands in windows

- cd → means change directory
- mkdir → means make make directory (create a new folder)
- git init → Start Git in that folder (make it a Git repo).
- Start . (or) explorer . → open that folder in windows Explorer.
- ls → list files and directories (current folder)
 - . → Alias of the current directory
 - .. → Alias of the parent directory
- cd . → Stay in the same place
- cd .. → go back to parent folder
- help → help on specific command
- Ex: - touch --help.

In mac @ Linux Commands.

nano → edit file.

clear → clear terminal.

Tab → Auto Complete command.

echo → print to terminal.

man → help on specific command.

pwd → prints path for current directory.

touch → create new file. } also work for window

> → write to the file.

>> → Append to the file.

cat → List content of the file.

ls -la → List of all folders.