Version Control & Git & GitHub

Introduction: Why Git? Why Version Control?

Let’s start with a real-life situation.

You’re working on a personal coding project — maybe a website, a mobile app, or even your graduation project. On **Monday**, everything is working fine. On **Tuesday**, you decide to try something new, and suddenly, the code breaks. You panic. You try to undo your changes manually, but it’s messy. You wish you had a **“go back” button** like in a video game.

Now imagine you’re **not working alone**. You’re working in a team. One person is updating the login system. Another is working on the homepage. You're both editing the same files. You send updated versions on WhatsApp, Google Drive, and sometimes by email. You now have files like:

- `project_final.js`
- `project_FINAL2.js`
- `project_USE_THIS_ONE.js`

Everyone is confused. Work is duplicated. Bugs appear out of nowhere. Deadlines are missed.

This is what happens **without version control**. It’s stressful, slow, and risky.

Now picture this:

- Every time you make a change to your code, it gets saved as a **checkpoint**.
- You can move back to any checkpoint whenever you want.
- You can write a message like “added login feature” or “fixed bug in profile page”.
- You can see exactly what changed, when, and who did it.
- You can try a new idea in a separate space, and only merge it when it’s ready.

That’s what **Git** does. It gives you full control over your project’s history.
And when you want to share your code or collaborate with others? That’s where **GitHub** comes in.

Git is the tool.
GitHub is the platform where teams connect, store code, review changes, and work together — all safely and efficiently.

Why Should You Care?

Whether you're building solo projects or working on a team, Git helps you:

- Keep your project safe
- Never lose progress
- Try new things without fear
- Go back in time if something breaks
- Work with others without stepping on each other’s toes

GitHub helps you:

- Store your project online
- Show off your work to employers
- Collaborate with developers all over the world

That’s why learning Git and GitHub is one of the most important skills for any developer

---

What is Version Control?

Version control is like a smart timeline for your project. It lets you:

- Save your progress step by step
- Go back to any previous version
- See who changed what and when
- Work with other people on the same codebase

Git is the most popular version control tool in the world.

What is Git?

Git is a tool that runs on your computer. It’s the most widely used version control system in the world — built for speed, flexibility, and teamwork.Git allows you to:

- Track every change in your code
- Save “checkpoints” (called **commits**)
- Work on different features using **branches**
- Merge your changes when you’re ready

And the best part? **It works offline**. You don’t need the internet to use Git on your machine.

What is GitHub?

GitHub is a cloud platform that stores your Git projects online. It allows you to:

- Back up your work to the cloud
- Share your code with others
- Collaborate with teammates in real-time
- Track issues, changes, and feature requests
- Contribute to open source projects

You can think of it like this:

- **Git** is the engine — it runs locally on your machine.
- **GitHub** is the garage — it stores your project safely and lets others access it.

https://git-scm.com/downloads

Part 1 – Git Basics

Before You Start

To start using Git, you need to:

1. **Install Git** on your machine (Windows, Mac, Linux)
2. **Configure Git** with your name and email — this info will be used in your commit history

Now you're ready to create your first Git project, also known as a **repository**.

Local vs Remote Repository

- **Local Repository**: A project folder on your computer that Git tracks.
- **Remote Repository**: The same project stored online, usually on GitHub.

You work on the local repo, and when you're ready, you push your changes to the remote one.

Part 2 – Git Core Commands

Here are the key Git commands every developer should know:

Command	Description
`git init`	Starts a new Git project in your folder
`git add`	Tells Git which files to track or stage for a commit
`git commit`	Saves a snapshot of your code with a message
`git status`	Shows which files have changed or are staged
`git log`	Shows your project's commit history
`git clone`	Downloads a remote GitHub repo to your local machine
`git push`	Uploads your local changes to GitHub
`git pull`	Downloads and applies changes from GitHub

Part 3 – GitHub Basics

Step 1: Create a GitHub Account

Visit github.com, sign up, and set up your profile.

Step 2: Create a New Repository

Click **"New Repository"**, give it a name, and choose whether to make it public or private.

Step 3: Connect Your Project to GitHub

Once your GitHub repo is created, you can **connect it to your local project** using:

- `git remote add origin <repo-url>`

Summary

- Git is your **local version control tool** — it tracks every change, lets you undo mistakes, and keeps your code organized.
- GitHub is your **online platform** to store, share, and collaborate on projects.
- Using both together gives you the best of both worlds — **control + teamwork**.
- The skills you learn here are essential in real-world development, jobs, and open-source contributions.