Understanding Git

What it is?

&

Why we need it?

Created by

Simran Kanojia

What is Git?

Git is a distributed version control (DVCS) that helps developers manage and track changes in their projects.

It’s especially useful when working with a team on the same code and also in case of switching back to some version that worked well earlier... (that's called version control or versioning)

Why do we need Git?

- Version control
- Collaboration
- Work Offline
- Experiment safely
- Mistake Recovery

1 Version Control

Git keeps track of every change in the project you are working on. If something goes wrong, you can go back to a previous version.

Example:

You're designing a website:

- In Version 1, you created a basic homepage layout.
- In Version 2, you added an image slider.
- In Version 3, you added a contact form, but it causes the slider to break. Now using Git, you can quickly go back to Version 2, where the slider was working perfectly and fix the issue, and then move forward.

2 Collaboration

Git allows multiple people to work on the same project without overwriting each other's work.

***Example*:**

- You and a teammate are editing the same file. So what git will do is.. it will merge your changes and highlight any conflicts so you can fix them easily.

3 Work Offline

With Git, each developer has a full copy of the project. You don’t need to be online to make changes.

Example:

- You can code while traveling, even without an internet connection (on your local machine). Later, you can upload your changes when you’re online.

4 Experiments Safely

Git lets you create branches to test new ideas without affecting the main project. Once your idea works, you can merge it into the main branch.

Example:

- You're building a new login feature for some application. Instead of changing the main project, you create a branch, try your idea, and merge it back when it's ready.

5 Mistakes Recovery

Git helps fix mistakes without losing work.

Example:

- You accidentally delete a file. Git can help you restore it from the previous version, that's where version controlling works wonders, right?

Git is like the ultimate "Save and Load" feature in a video game:

1 Save: Imagine you're at a tough level in the game and you hit Save to record your progress (just like saving the current state of your code in Git).

2 Load: Something goes wrong, and you lose the game (or mess up your code). No problem! You can return to a previous save point.

3 Collaborate: Think of multiplayer mode, where everyone plays together, but each player can save their progress separately and then merge it all back into the main storyline when ready.