

Session 4

4. Git & GitHub

4.1 How Git Works?

Imagine you and your friend are both working on the same project on two different computers.

- You change one file, your friend changes another file.
- Now the big question: **"Whose changes should we keep? Or can we keep both?"**

Without Git → total mess. Files get overwritten, work gets lost.

With Git → every change is tracked safely.

You can always:

- Save your work (commit)
- Send it to GitHub (push)
- Get your friend's changes (clone or pull)
- And even go back in time if something breaks

The commands you'll actually use:

```
git add .           # Tell Git "I want to add these changes"
git commit -m "message" # Save the changes as a checkpoint
git push           # Upload your changes to GitHub
git clone <url>      # Download someone else's project from GitHub
```

👉 That's literally 90% of Git you'll use in real life.

This way, students don't need to think about *staging area* or *repository database* at the start.

Just → **"Git helps multiple computers and people share changes without losing work."**

4.2 What is Git?

Git is a tool that tracks your code changes. It's like **undo/redo** for your entire codebase. Instead of saving versions manually, Git does it for you.

Note: Git is local (on your machine). GitHub is a platform where you upload Git-tracked projects.

4.3 Why Git?

- Saves every version of your project
- Lets teams collaborate on code
- Keeps backups
- Makes you look professional

4.4 Installing Git

Download Git from: git-scm.com

Once installed, you can use Git from your terminal:

```
git --version
```

Output:

```
git version 2.XX.0
```

4.5 Basic Git Commands You'll Actually Use

```
git add .          # Stage all changes
git commit -m "message" # Save snapshot
git push           # Upload snapshot to GitHub
git clone <url>     # Download a repo from GitHub
```

4.6 Cloning a Repository (with Token Authentication)

If you want to clone a private repo, you need a **GitHub token**.

Steps:

1. Go to **GitHub → Settings → Developer Settings → Personal access tokens → Tokens (classic)**
2. Click **Generate new token (classic)**
 - Select `repo` access
 - Generate token
3. Copy the token (you won't see it again!)
4. Use it while cloning:

```
git clone https://<TOKEN>@github.com/username/repo.git
```

👉 Replace `<TOKEN>` with your generated token.

4.7 Common Git Helpers

```
git status  # See what's changed
git log     # View commit history
git diff    # View changes before committing
```

SSSSO Vidyakshina 2025

