Basics of Git

Introduction to Git

Git is a version control system (VCS) that helps developers track and manage changes to their codebase. It allows multiple people to collaborate on the same project without overwriting each other’s work.

Why Use Git?

Track Changes: Keeps a history of modifications.

Collaboration: Multiple people can work on a project simultaneously.

Branching and Merging: Create separate branches for different features and merge them later.

Basic Git Concepts

Repository (Repo): A directory where Git stores the project’s files and history.  
You can create a repo locally or clone an existing one.

Commit: A snapshot of your changes. Every commit has a unique ID (hash).

Branch: A branch is a separate line of development. The default branch is usually main or master.

Merge: Combines changes from different branches.

Staging Area: A place where you prepare your changes before committing.

Basic Git Workflow

Initialize a Repository

git init - Creates a new Git repository in the current directory.

Clone an Existing Repository

git clone <repository\_url> - Clones a remote repository to your local machine.

Check Status

git status - Shows the current status of your working directory.

Add Files to Staging Area

git add <filename> # Add a specific file

git add . # Add all files in the directory

Commit Changes

git commit -m "Your commit message"

Saves the changes to the local repository with a message describing the changes.

View Commit History

git log - Shows the commit history.

Create a New Branch

git branch <branch\_name>

Switch to Another Branch

git checkout <branch\_name>

Merge Branches

git merge <branch\_name>

Push Changes to Remote Repository

git push origin <branch\_name>

Pull Changes from Remote Repository

git pull

Common Commands

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

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

git remote add origin <repository\_url>

git diff – Show changes between commits

git reset <file> – Unstage file

Undoing Changes

Unstage a File

git reset <file>

Undo Last Commit (Soft Reset)

git reset --soft HEAD~1

Discard All Local Changes

git checkout -- .