**Git and GitHub**

**What is Git?**

Git is a **version control system (VCS)** that helps track changes in your code. It allows multiple people to work on a project simultaneously and keeps a history of all changes.

**What is GitHub?**

GitHub is a **cloud-based hosting service** for Git repositories. It allows developers to collaborate, review code, and manage projects efficiently.

**Git Basics: Commands**

**Initializing a Repository: -** This initializes an empty Git repository.

git init

**Cloning an Existing Repository: - This** copies a repository from GitHub to your local machine.

git clone <repo\_url>

**Checking Repository Status: -** Shows the current state of your repository (new, modified, staged files).

git status

**Adding Changes to Staging Area: -**

git add <filename> # Add a specific file

git add . # Add all changes

**Committing Changes: -**

git commit -m "Your commit message"

**Saves your changes with a message describing them: -** Viewing Commit History

git log

Shows the list of commits.

**4. Working with Branches**

Branches allow working on different features without affecting the main codebase.

**Creating a New Branch**

git branch new-feature

**Switching to Another Branch**

git checkout new-feature

**or (in newer Git versions)**

**git switch new-feature**

**Merging Branches**

**git checkout main # Switch to the main branch**

**git merge new-feature # Merge the branch into main**

**5. Pushing Code to GitHub**

1. Link your local repository to GitHub:

**git remote add origin <repo\_url>**

1. Push your code:

**git push -u origin main**

**6. Pulling and Fetching Changes**

Pulling Updates from GitHub

**git pull origin main**

Fetches and merges changes from the remote repository.

Fetching Changes Without Merging

**git fetch origin**

Gets updates from GitHub but doesn’t merge them automatically.

**7. Resolving Merge Conflicts**

When multiple people edit the same file, conflicts can occur.

1. Git will show the conflict in the file.
2. Manually edit the file to resolve the conflict.
3. Add the resolved file and commit it:

git add <conflicted\_file>

git commit -m "Resolved merge conflict"

**8. Undoing Changes**

Undo Last Commit (But Keep Changes)

git reset --soft HEAD~1

Undo Last Commit (And Remove Changes)

**git reset --hard HEAD~1**