DAY:3,4,5
DATE:13.06.2025

GIT:

🔷 What is Git?

Git is a **distributed version control system (DVCS)** used to track changes in source code during software development. It enables multiple developers to work on the same project efficiently without overwriting each other's work. Git is free, open-source, and was created by **Linus Torvalds** in **2005**.

🔷 Key Features of Git

1. **Distributed Architecture** – Every developer has a local copy of the repository.
2. **Data Integrity** – Uses SHA-1 hash for secure identification of files.
3. **Branching and Merging** – Lightweight and fast branches for experimentation.
4. **Staging Area** – Intermediate area to prepare commits.
5. **Speed and Performance** – Fast operations for commits, merges, diffs, etc.
6. **Collaboration** – Supports teamwork and remote repository syncing (e.g., with GitHub).

🔷 Why Use Git?

- Version control for projects
- Keeps track of every change
- Enables collaboration

- Supports backup and restore
- Helps in code reviews and testing

◈ Basic Git Workflow

1. Create or clone a repository.
2. Make changes to your code.
3. Stage the changes.
4. Commit the changes.
5. Push the changes to a remote repository.
6. Pull changes made by others.

repositories

◈ Common Git Commands

1. 🔧 Configuration

git config --global user.name "priyadharshini2926"

git config --global user.email "priyadharshini2926@gmail.com"

Set up your identity for all Git repositories.

---

2. 🗁 Repository Setup

git init

Initialize a new Git repository.

git clone <namerepo>

Clone a remote repository to your local system.

---

3. 📄 File Operations

git add <file>

git add .

Stage files for commit.

git commit -m "Commit message"

Commit the staged changes with a message.

---

4. 🔁 Working with Changes

git status

Check the status of changes in your project.

git diff

See the differences between files before committing.

---

5. 🔀 Branching and Merging

git branch

git branch <branch_name>

git checkout <branch_name>

git merge <branch_name>

Create, switch, and merge branches.

---

6. 📤 Pushing and Pulling

git push origin <branch_name>

Push changes to the remote repository.

git pull

Fetch and merge changes from the remote repository.

---

7. 📜 Logs and History

git log

View the commit history.

git show <commit_id>

Show details of a specific commit.

---

8. 🚫 Undo Changes

git reset <file>

git checkout -- <file>

Unstage or discard local changes.

What is GitHub?

GitHub is a **web-based platform** used for **hosting Git repositories**. It adds a graphical user interface (GUI) and collaboration features on top of Git. Developers use GitHub to **store**, **manage**, **track**, and **collaborate** on software projects.

GitHub was launched in **2008** and is now owned by **Microsoft** (acquired in 2018).

◈ Key Features of GitHub

1. **Repository Hosting**
   Stores your Git repositories online and keeps them secure.
2. **Collaboration**
   Multiple developers can work on the same project using forks, branches, and pull requests.
3. **Pull Requests**
   Used to propose changes and review code before merging.
4. **Issues and Bug Tracking**
   Built-in issue tracker to report bugs, suggest features, and manage tasks.
5. **Actions (CI/CD)**
   Automate your workflows like testing, building, and deployment.
6. **Code Review**
   Tools for reviewing code, suggesting changes, and discussing commits.
7. **README and Documentation**
   Projects often include README.md to explain purpose, setup, and usage.
8. **Project Management**
   Includes tools like Kanban boards, milestones, and labels.
9. **Wiki**
   Add structured project documentation using GitHub Wiki.
10. **Security**
    Dependabot alerts, secret scanning, and branch protection rules.

🔷 Why Use GitHub?

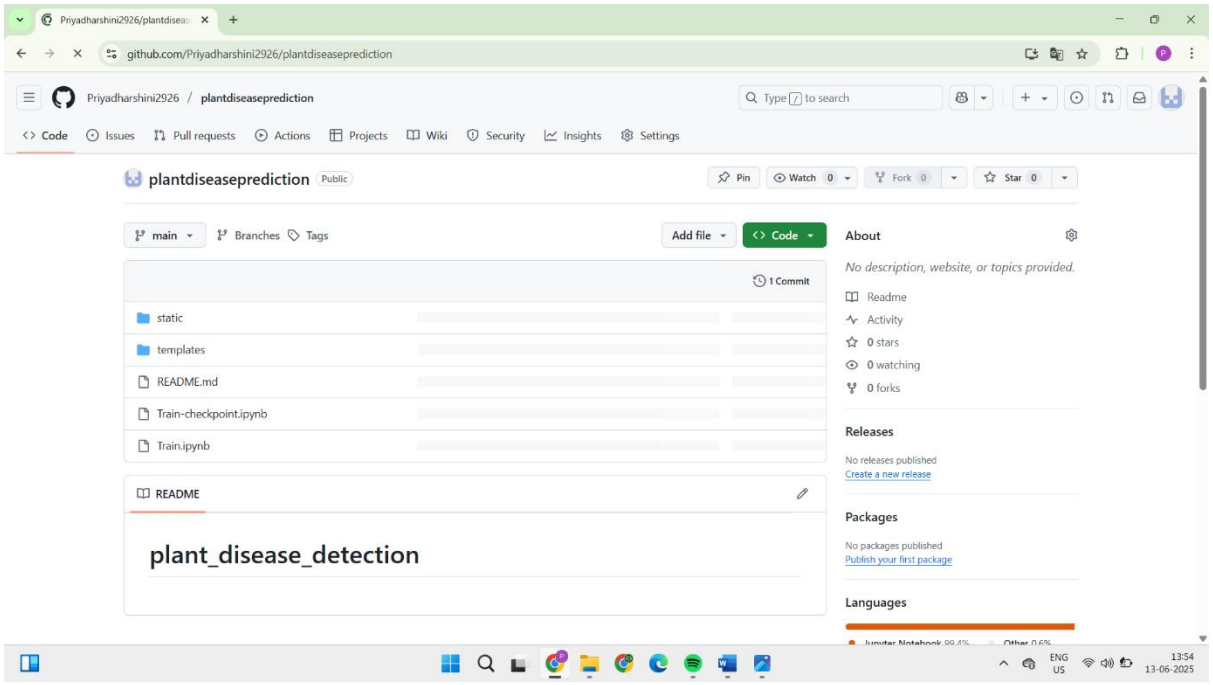
- Central location for open-source and private code
- Helps teams collaborate easily
- Offers tools for code reviews and tracking issues
- Integrates with tools like Slack, Trello, VS Code, and Jenkins
- Supports continuous integration and deployment
- Has a large open-source community

🔷 Important GitHub Terms

Term	Description
Repository (Repo)	A project folder hosted on GitHub containing code and files
Fork	A copy of a repository you can modify freely
Pull Request	A request to merge changes from one branch or fork to another
Commit	A saved change in the project
Branch	A version of the repo used for development
Issue	A way to track bugs, tasks, or enhancements

Term	Description
README	A markdown file that describes the project
Star	Bookmark or like a project
Watch	Subscribe to updates in a repository

WITH WORKING WITH GIT COMMANDS I HAVE PUSH MY PLANT DISEASE PROJECT IN GITHUB WITH HELP OF GIT COMMANDS:



THANK YOU