LAB-1

Git and Github

Theory

**Git**

Git is a **distributed version control system (DVCS)** that helps developers track changes in their code, collaborate with others, and manage different versions of a project efficiently. It was created by **Linus Torvalds** in 2005 for Linux kernel development.

#### ****Git Workflow****

1. **Working Directory** – The files you are currently working on.
2. **Staging Area (Index)** – Files that are marked to be committed.
3. **Repository (Local Repo)** – The committed files stored locally.
4. **Remote Repository** – A shared repository (e.g., GitHub, GitLab, Bitbucket).

#### ****Why Use Git?****

* **Version Control**: Tracks changes in files over time.
* **Collaboration**: Multiple developers can work on the same project simultaneously.
* **Branching & Merging**: Work on new features without affecting the main project.
* **Backup & Recovery**: Keeps a history of changes, preventing data loss.
* **Speed & Efficiency**: Git is lightweight and fast compared to other version control systems like SVN.

**GitHub**

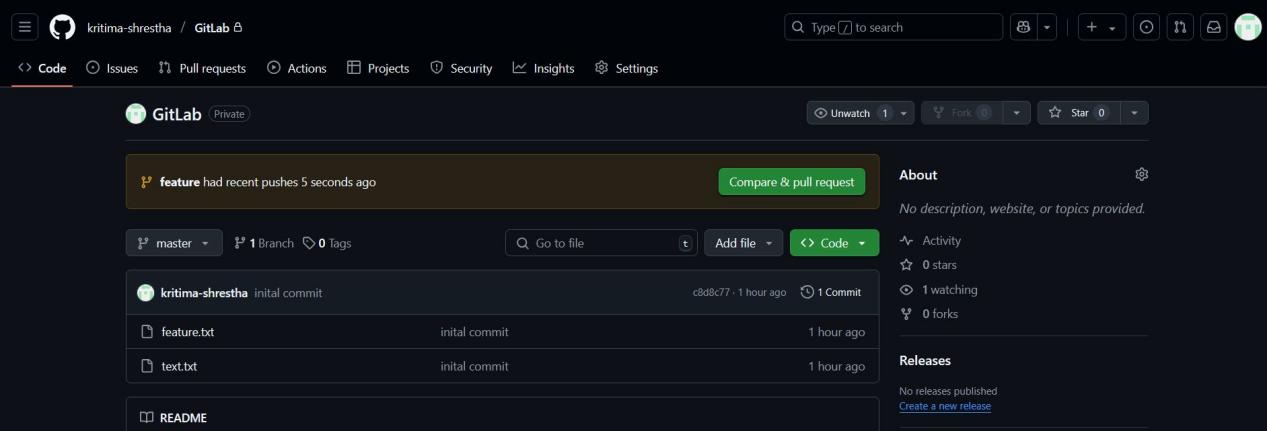
GitHub is a cloud-based platform for **version control** and **collaboration**, primarily used for managing software development projects. It is built around **Git**, a distributed version control system created by **Linus Torvalds**.

**Forking & Cloning**

* **Forking** creates a personal copy of another user’s repository.
* **Cloning** downloads a repository to a local computer for offline development.

**Common Git Commands**

1. Git init - Initialize a new Git repository
2. Git remote add origin <https://github.com/kritima-shrestha/GitLab.git> - link new repository of github with the local codes
3. Git add . - Add file(s) to the staging area
4. Git commit -m “message” - Commit changes with a message
5. Git status- Check the status of the working directory
6. Git log - View commit history
7. Git branch - List the branches exist
8. Git branch <branch\_name> - Create new branch
9. Git checkout <branch> - Switch to another branch
10. Git switch <branch> - Switch to another branch
11. Git merge <branch> - Merge a branch into the current branch
12. Git push -u origin <branch\_name> - Upload commits to a remote repository
13. Git clone <repo\_url> - Copy(clone) an existing repository
14. Git pull - Fetch and merge changes from a remote repository

Outputs

