Week-2 Learning Summary: Git & GitHub

Pre-requisites

Before starting the session, I installed and set up:

1. **Git Bash** – A terminal to run Git commands
2. **GitHub Account** – Signed up using my personal email

---

1. Introduction to Version Control Systems

Version Control Systems (VCS) help track and manage code changes.

Types of VCS:

- **Centralized VCS** – A single central server (e.g., SVN)
- **Distributed VCS** – Every developer has a full copy (e.g., Git)

Git Features:

- Fast and distributed
- Tracks history and changes
- Supports branches, merging, and collaboration

---

2. Basic Git Workflow

Git Structure:

- **Working Directory** – Your local project folder
- **Staging Area (Index)** – Prepares changes before commit
- **Local Repository** – Your versioned code on your machine
- **Remote Repository** – Stored on GitHub or GitLab

Important Terms:

- **Origin** – Name for the remote repository
- **Master/Main** – Default primary branch
- **HEAD** – Points to the current working commit

---

3. Git Commands vs GUI (GitHub Desktop)

A. Local Repository Commands

Task	Command
Initialize Git	git init
Check status	git status
Add files to stage	git add filename or git add .
Commit changes	git commit -m "commit message"
View commit log	git log
Remove staged file	git reset filename
Amend last commit	git commit --amend

B. Remote Repository (GitHub)

Task	Command
Clone repo	git clone <repo-url>
Add remote	git remote add origin <url>
Push changes	git push origin main
Pull updates	git pull origin main
Remove remote	git remote remove origin

---

4. Branching in Git

A. Branching

Task	Command
Create branch	git branch feature-xyz
Switch branch	git checkout feature-xyz
Create & switch	git checkout -b feature-xyz
Push branch	git push origin feature-xyz
Delete local branch	git branch -d branch-name
Delete remote branch	git push origin --delete branch-name
Compare branches	git diff main feature-xyz
Update local with remote	git pull

B. Merging & Conflict Resolution

Task	Command
Merge branches	git merge branch-name
View conflicts	Occurs during merge – manual resolution
Rebase branch	git rebase branch-name

C. Tagging in Git

Task	Command
Create tag	git tag v1.0
Push tag	git push origin v1.0
Delete tag (local)	git tag -d v1.0
Delete tag (remote)	git push origin --delete tag v1.0

D. Patches

Task	Command
Generate patch	git format-patch1
Apply patch	git applypatchfile.patch

---

5. Pull Requests

Pull Requests are used to propose changes and request review before merging.

Pull Request Steps:

1. Push your branch to GitHub
2. Click “New Pull Request”
3. Select base and compare branches
4. Submit PR and request reviewers