NORAIZ RANA

Task #6

Topic

Introduction to Git and Github(version control)

Date:

7 July 2025

SUBMITTED TO:

APPVERSE TECHNOLOGIES

Understanding Git and Version Control

What is Version Control?

Version control is a system that allows developers to manage and track changes in their code over time. It enables collaboration among multiple developers and helps revert back to previous versions if needed.

There are two main types:

- **Local Version Control:** Tracks changes on a local machine.
- **Distributed Version Control:** Tracks changes on both local and remote repositories. (Git uses this)

Introduction to Git

Git is a free, open-source distributed version control system created by Linus Torvalds in 2005. It helps manage project history efficiently and supports multiple developers working on the same project.

Key Git Features:

- Tracks file changes and project history.
- Allows branching and merging.
- Works offline (local repo).
- Enables rollback to previous versions.

Common Git Commands:

Command	Purpose
`git init`	Initializes a Git repository
`git add .`	Adds all changes to staging
`git commit -m "msg"`	Saves changes with a message
`git status`	Shows current status of repo

Command	Purpose
`git log`	Shows commit history
`git branch`	Creates or lists branches
`git merge`	Merges branches
`git clone <url>`	Copies a repo from GitHub
`git push`	Uploads code to GitHub
`git pull`	Downloads latest code from GitHub

Introduction to GitHub

What is GitHub?

GitHub is a cloud-based platform that uses Git for version control. It allows developers to store code online, collaborate with others, track issues, and manage open-source or private projects.

Git vs GitHub:

Git	GitHub
Tool for version control	Platform for hosting Git repositories
Installed locally	Cloud-based web service
Works offline	Requires internet

Key Features of GitHub:

- **Repositories:** Storage spaces for code.
- **Branches & Pull Requests:** Support collaboration and code reviews.
- **Issues & Discussions:** Track bugs and features.
- **Actions:** Automate workflows like testing and deployment.

- **README & Markdown Support:** Write clean documentation.

GitHub Workflow Summary:

1. Create a new GitHub repo.
2. Clone it to your machine: `git clone <repo-url>`
3. Make changes in your code.
4. Stage the changes: `git add .`
5. Commit changes: `git commit -m "your message"`
6. Push to GitHub: `git push origin main`
7. Collaborators can pull, review, and merge changes via pull requests.

What I Learned:

- Git is essential for managing code history and collaboration.
- GitHub makes it easier to share code, work in teams, and contribute to open-source.
- I practiced Git commands and created my own repositories on GitHub.
- I now understand the difference between local and remote version control and the power of branches.