ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING TRAINING TR-102 REPORT DAY 4 26 JUNE 2025

Overview:

The fourth day of training focused on Git and GitHub, two essential tools for version control and collaborative software development. We learned how to track changes in projects, manage versions, and collaborate with others using repositories. The session emphasized the importance of Git in managing AI and programming projects efficiently.Through practical exercises, we created repositories, committed changes, and synchronized local projects with GitHub — a platform that allows developers to share and manage code online.

Learning Objectives:

- Understand the concepts of Git and GitHub.
- Learn the purpose of version control systems.
- Execute essential Git commands for version management.
- Create and manage repositories locally and remotely.
- Understand how GitHub enables team collaboration in projects

Introduction to Git and Github

Git is a distributed version control system that tracks changes in files and helps multiple developers work together on the same project efficiently. It allows users to revert to earlier versions, merge changes, and maintain a detailed history of modifications.GitHub is a cloud-based platform built on top of Git. It provides an online space to host repositories, collaborate with team members, and contribute to open-source projects. Together, Git and GitHub are vital for developers, especially in large AI and software projects.

Why Git and GitHub are Important:

1. **Version Control:**
   Git keeps track of every modification, allowing users to go back to a previous version anytime.

2. **Collaboration:**
   Multiple users can work on the same project simultaneously without overwriting each other's changes.

3. **Backup and Accessibility:**
   GitHub stores code in the cloud, so it can be accessed from anywhere.

4. **Open Source Contributions:**
   Developers can contribute to global projects, build portfolios, and share knowledge.

5. **Integration with AI and Development Tools:**
   GitHub integrates with tools like Jupyter Notebook, VS Code, and CI/CD pipelines, making it essential for machine learning development.

Basic Git Workflow:

A typical Git workflow includes the following steps:

1. **Create or clone a repository**

2. **Make changes to files**

3. **Stage changes for commit**

4. **Commit changes with a message**

5. **Push the changes to GitHub**

Common Git Commands:

1. Setup Commands:

- `git config --global user.name "Your Name"` – Sets your username.
- `git config --global user.email "your_email@example.com"` – Sets your email address.

2. Repository Initialization:

- `git init` – Initializes a new local Git repository.
- `git clone <repository_url>` – Clones an existing repository from GitHub to your local machine.

3. Basic File Operations:

- `git status` – Displays the status of your working directory.
- `git add <filename>` – Stages a specific file.
- `git add .` – Stages all modified files.
- `git commit -m "Commit message"` – Commits staged files with a descriptive message.

4. Branching and Merging:

- `git branch` – Lists all branches
- `git branch <branch_name>` – Creates a new branch.
- `git checkout <branch_name>` – Switches to another branch.
- `git merge <branch_name>` – Merges a branch into the current branch.

5. Remote Repository Commands:

- `git remote add origin <repository_url>` – Connects local repo to remote GitHub repo.
- `git push -u origin main` – Pushes changes to the remote repository.
- `git pull` – Fetches and merges changes from the remote repo.
- `git fetch` – Downloads changes without merging them.

6. Viewing History and Logs:

- `git log` – Displays commit history.
- `git diff` – Shows differences between file versions.
- `git show <commit_id>` – Displays detailed information about a specific commit.

Conclusion:

Day 4 gave us hands-on experience with Git and GitHub, essential tools for any software developer or AI engineer. We learned to manage versions, track changes, and collaborate effectively using repositories.This session highlighted that Git and GitHub are not just tools but an integral part of professional software development, research projects, and open-source communities.