**1. Introduction to Git and Version Control**

* You led a project involving complex functionalities and expected many code revisions.
* The lab introduced **Git**, a distributed version control system.
* Key skills included:
  + Connecting to a **Linux VM**
  + Installing and configuring **Git**
  + Creating a **local repository**
  + Performing **basic Git operations**: add, edit, commit

**2. Key Git Terms and Definitions**

* **Branch**: An independent line of development.
* **Commit ID**: Unique identifier for a commit.
* **Fast-forward merge**: A simple merge with no divergence.
* **Head**: Points to the latest commit on the current branch.
* **Master**: Default branch name for new repositories (now often called main).
* **Merge conflict**: Happens when conflicting changes are made to the same file.
* **Rollback**: Reverting to a previous commit.
* **Three-way merge**: Merge based on a common ancestor between two branches.

**3. Exploring an Existing Repository**

* Checked out an existing Git repository.
* Made and committed changes.
* Practiced **rolling back commits** to fix issues and produce correct results.

**4. Introduction to GitHub and Remote Workflows**

* Created and cloned **GitHub repositories**.
* Explained the concept of a **remote repository**.
* Commands learned:
  + modify, stage, commit for local changes
  + fetch to retrieve changes from remotes
* **Protocols**:
  + **SSH** and **API keys** for secure communication
* Conflict management:
  + Learned the **pull-merge-push** workflow
  + Pushing branches for **collaborative testing**
  + Using rebase for a cleaner commit history

**5. Collaboration and Project Management with GitHub**

* Overview of **pull requests (PRs)**:
  + How to create, review, and merge PRs
  + Typical PR workflow on GitHub
* **Squashing commits** to clean up history
* **Fork and pull request** model explained
* **Code reviews**:
  + Workflow and etiquette
  + Reviewing code on GitHub
* **Project management**:
  + Best practices for collaboration
  + **Issue tracking**
  + Introduction to **Continuous Integration (CI)**

**Learning Objectives Recap**

* Explain and use branches effectively
* Understand and perform rebasing
* Work with remote repositories
* Resolve conflicts using Git workflows
* Push and manage code for collaboration
* Participate in pull request and code review processes
* Manage projects using GitHub tools and CI practices