

**Source Code Lab**

**Name: Partha R**

**Branch: B.Tech**

**Section: CSE**

**Course: Source Code Lab**

**SEN: A86605223038**

**Submitted To: Dr. Monit Kapoor**

**Date of Submission: 6 June 2025**

**INDEX:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Topic** | **Date** | **Pg No** |
| **1** | **Git & GitHub Fundamentals-** Install Git & GitHub- Basic Linux Commands | **31-01-2025** |  |
| **2** | **Committing & Viewing History-** Understanding Commits- Viewing Git Logs | **14-02-2025** |  |
| **3** | **Branching & Version Comparison-** Creating & Managing Branches- Using git diff | **07-03-2025** |  |
| **4** | **Remote Repositories-** Pushing Local Repos to GitHub | **16-05-2025** |  |
| **5** | **Collaboration & Conflict Resolution-** Inducing & Resolving Conflicts- Opening & Closing Pull Requests | **21-05-2025** |  |

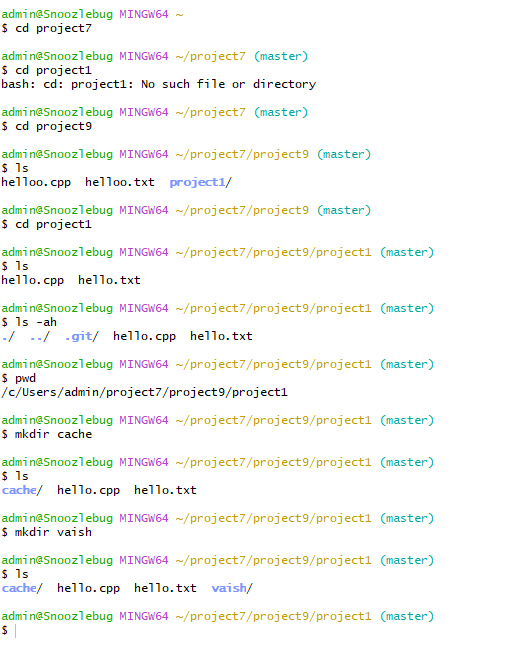


Amity School of   
Engineering & Technology

**1.Git & GitHub Fundamentals**

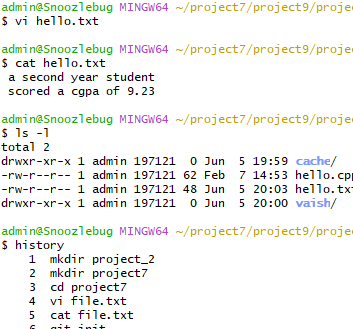
* **Basic Commands**
* ls – List directory contents
* ls -ah – List all files including hidden ones
* pwd – Print working directory
* cd .. – Move to the parent directory
* mkdir – Create a new directory

cd <folder> – Change into a directory



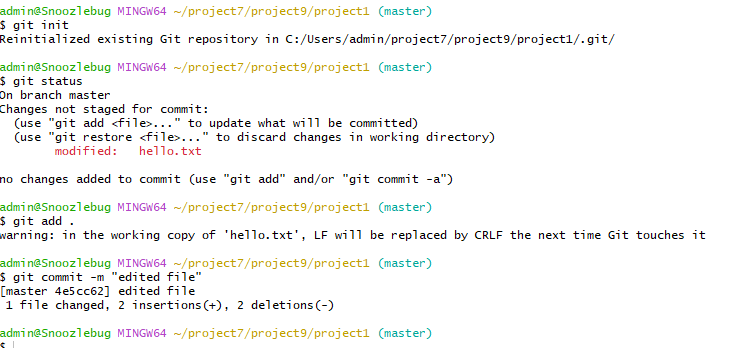
**2.File Operations**

* vi <filename> – Open file in vi editor
  + Press i to insert
  + Press Esc to exit insert mode
  + Use :wq to save and quit
* cat <filename> – Display contents of a file
* ls -l – Show file permissions and details
* history – Show command history
* clear – Clear the terminal screen



3. **Git Initialization**

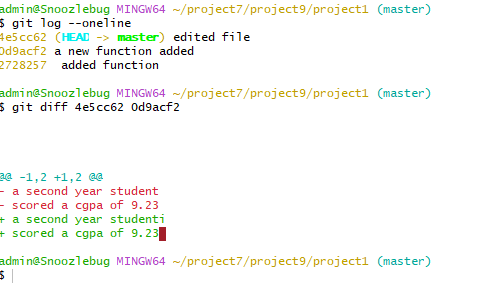
* git init – Initialize a Git repository
* git status – Show the working tree status
* git add <filename> – Stage a file for commit
* git commit -m "message" – Commit changes with a message
* Configure Git with your email and username



**4.Committing & Viewing History**

1. **git log --oneline – This command shows a condensed version of your commit history**

**2. git diff - shows the differences between various states of your repository**



**5.Branching & Version Comparison**

**1. Branching:**

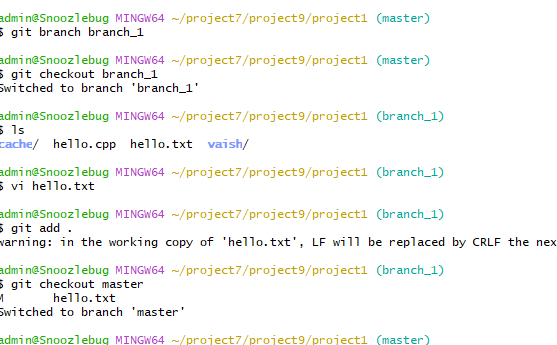
\* git branch [branchname]: Create a new branch

\* git checkout [branchname]: Switch to a different branch

\* git checkout master: Switch back to the master branch

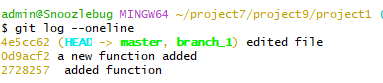
\* **Concept:** Any branch and the master branch can have completely different codebases after many commits in their respective branches.

\* **HEAD:** A special pointer that points to the last commit in the current branch



**6.Comparing Changes:**

\* git diff: Show changes between commits, commit and working tree, etc.



**7.Induce Conflicts, Open and Close Pull Requests**

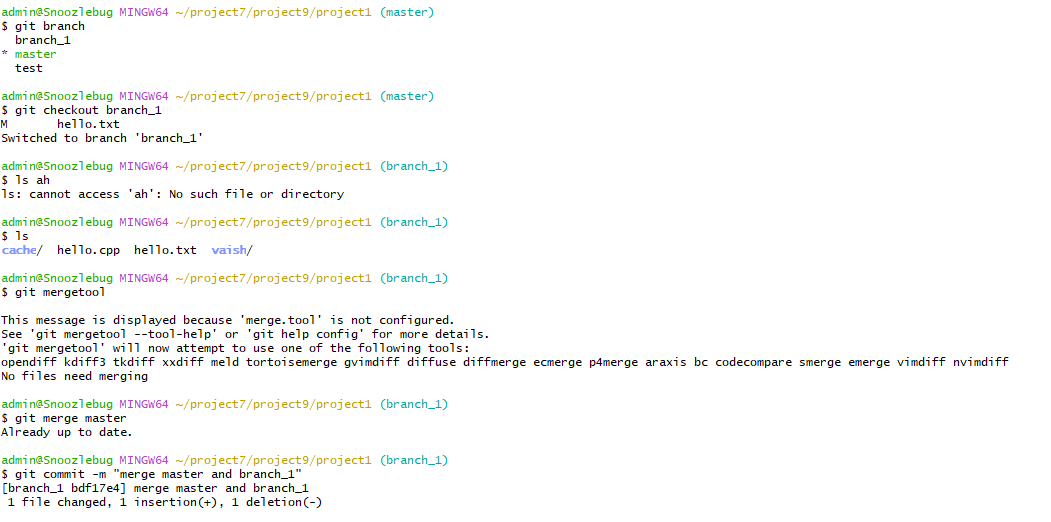
* 1. **Git Merging & Conflicts:**

\* git merge: Integrate changes from one branch into another.

\* Conflicts arise during merging when Git cannot automatically reconcile diverging changes.

\* Use a merge tool to resolve conflicts.

\* git mergetool: Command to launch the configured merge tool.



A screenshot of a computer

AI-generated content may be incorrect.

**8.Gitignore**: \*. gitignore file: Specifies intentionally untracked files that Git should ignore.

\* Files listed in .gitignore will not be tracked by Git and will not be mirrored on GitHub

**9.Pull Requests**: \* Opening a pull request

A screenshot of a computer

AI-generated content may be incorrect.

10.Closing a pull request:

A screenshot of a computer

AI-generated content may be incorrect.