Heaven's Light is Our Guide

Rajshahi University of Engineering and Technology

Course Code

ECE 3118

Course Title

Software Engineering & Information System Design Sessional

Lab Report 3: Study of different git commands.

Submitted to	**Submitted by**
Oishi Jyoti	Md. Tajim An Noor
Assistant Professor	Roll: 2010025
Dept of ECE, Ruet	

Study of different git commands.

Introduction

Git is a widely-used version control system that allows developers to track changes in code, collaborate, and manage projects efficiently. It records each modification in a repository, making it easy to revisit or revert to previous states. GitHub, a platform that hosts Git repositories, provides additional tools for code sharing, collaboration, and project management. Together, Git and GitHub enable streamlined teamwork, transparency in project history, and an organized approach to software development.

Basic Git Commands

Here is a list of essential Git commands used for creating, managing, and collaborating on Git repositories:

`git init` — Initializes a new Git repository in the current directory.

`git clone <url`
Clones an existing repository from a remote source (e.g., GitHub) to your local machine.

`git status`
Displays the current status of the repository, showing staged, unstaged, and untracked files.

`git add <file>`
Adds specified files to the staging area in preparation for a commit. Use `git add .` to stage all changes.

`git commit -m "message"`
Commits the staged changes with a descriptive message.

`git push`
Uploads local commits to a remote repository (e.g., GitHub).

`git pull`
Retrieves the latest changes from a remote repository and merges them into the local branch.

`git branch`
Lists all branches in the repository. Add `git branch <branch_name>` to create a new branch.

`git checkout <branch_name>`
Switches to the specified branch. Use `git checkout -b <new_branch>` to create and switch to a new branch simultaneously.

`git merge <branch_name>`
Merges the specified branch into the current branch.

`git log`
Displays the commit history, including commit IDs, messages, and authors.

`git diff`
Shows changes made to tracked files that have not been staged. Use `git diff <file>` to view changes for a specific file.

`git stash`
Temporarily saves changes that are not ready to be committed, allowing a clean working directory.

`git remote add origin <url>`
Links a local repository to a remote repository for easy collaboration.

`git rm <file>`
Removes a file from the repository and stages the deletion.