Task\_05

Name : Manyu kumar

Reg : 2141011082

**Git Important commands**

|  |
| --- |
| **$** - this sign means you are root user and you have all the permissions |
| (Master) or (main) indicate that you are currently on the root branch of the repository. |
| **git --version**   * It shows the which version is installed into your system | |
| **ls**   * The ls command displays a list of files and directories in the current working directory. | |
| **git config --global user.name "Manyu Kumar"**  **git config --global user.email "manyu084@gmail.com"**   * These settings ensure that all commits made from your system are associated with your name and email across all repositories.   **git config --global user.email**  **git config --global user.name**   * The following commands display the global Git configuration settings for your email and username: | |
| **echo hello >> file.txt**   * The command echo hello >> file.txt appends the text hello to the end of the file named file.txt. * If file.txt does not already exist, it will be created automatically. | |
| **git add file1.txt**   * The command git add file1.txt stages the file file1.txt for the next commit. * This means any changes made to file1.txt will be included when you run git commit.     Note: You can safely ignore any warnings that appear after adding files to the staging area using git add. | |
| **git status**   * The command git status displays the current state of the working directory and the staging area. * It shows which changes have been staged, which haven’t, and which files aren’t being tracked by Git.   **Example is attached in next page :** | |
| **git commit -m “file updated”**   * The command git commit -m "file updated" creates a new commit with the message *"file updated"*. * The -m flag allows you to include a brief commit message directly in the command, summarizing the changes made. | |
| **git checkout -f**   * The command git checkout -f forces Git to discard all local changes in the working directory and switch to the currently checked-out branch. * It restores all files to their last committed state, effectively overriding any uncommitted modifications. * Note: This command will permanently delete any uncommitted changes. Use with caution. | |
| **git mv file1.txt file2.txt**   * The command git mv file1.txt file2.txt renames or moves the file file1.txt to file2.txt and stages the change for the next commit. * This is equivalent to running mv file1.txt file2.txt followed by git add file2.txt and git rm file1.txt, but in a single, streamlined command.   **Example is attached in next page :** | |
| **git ls-files**   * The command git ls-files lists all the files that are currently being tracked by Git in the working directory. * This includes files that have been staged, committed, or are part of the repository’s version history—excluding untracked or ignored files by default. | |
| **git log**   * The command git log displays the commit history of the current branch in reverse chronological order (most recent commit first). * It shows details such as the commit hash, author, date, and commit message, allowing you to track changes over time. | |
| **git branch {double tab}**   * This will list all available branch names in your repository, helping you quickly view or switch between them without typing the full names manually. | |
| **git branch “name of branch”**   * The command git branch "name of branch" creates a new branch with the specified name.      * We can see new branch Assignment\_05 is created | |
| **git checkout {name of branch}**   * The command git checkout {name of branch} switches your working directory to the specified branch.      * **Note:** We may observe that the branch name master has been replaced by Assignment\_05. * This indicates that you are currently working on the Assignment\_05 branch instead of the default master branch. | |
| **git merge “name of current branch”**   * The command git merge "name of current branch" merges the specified branch into your current branch. * Important: We typically merge another branch into your current branch—not the current branch into itself. | |

|  |
| --- |
| **git branch -d “name of branch that we wanted to delete”**   * The command git branch -d "branch-name" deletes the specified local branch, but only if it has been fully merged with your current branch or another base branch (like main or master). |
| **git clone “link of repository “**   * The command git clone "repository-link" is used to create a local copy of a remote Git repository. * It downloads all files, branches, and commit history from the specified repository. |
| **git rm file1.txt**   * **The command git rm file1.txt removes the file file1.txt from both the working directory and the staging area.** |