**Git-HOL Task 3**

**Aim :** Construct a branch, do some changes in the branch, and merge it with master.

**Procedure :**

**Explain branching and merging**

**Branching**

Branching allows you to create separate lines of development within the same repository. Each branch is an independent copy of the project’s codebase, enabling you to work on features, fixes, or experiments without affecting the main codebase.

**Merging**

Merging integrates changes from one branch into another, combining their histories. It’s how you bring work from a feature branch back into the main branch or another branch.

**Explain about creating a branch request in GitLab**

We can do it as below :

1. Creating a Branch

2. Creating a Merge Request

3. Review and merge

4. Clean up

**Explain about creating a merge request in GitLab**

We can do it as below :

Create and push a branch using below commands :

git checkout -b feature/add-login

git add .

git commit -m "Add login feature"

git push -u origin feature/add-login

Commands I have used during my assignment are :

mkdir GitLabDemo

cd GitLabDemo

git init

echo "Helloo all" > hello.txt

git add hello.txt

git commit -m "Initial commit"

git branch GitNewBranch

git checkout GitNewBranch

echo "This is a new file in GitNewBranch" > branchfile.txt

git add branchfile.txt

git commit -m "Added branchfile.txt in GitnewBranch"

git checkout main

git diff main..GitNewBranch

git merge GitNewBranch

git log --oneline --graph –decorate

git branch -d GitNewBranch

git status

**Procedure :**

**Branching:**

1. Create a new branch **“GitNewBranch”.**
2. List all the local and remote branches available in the current trunk. Observe the “\*” mark which denote the current pointing branch.
3. Switch to the newly created branch. Add some files to it with some contents.
4. Commit the changes to the branch.
5. Check the status with **“git status”** command.

**Merging:**

1. Switch to the master
2. List out all the differences between trunk and branch. These provide the differences in command line interface.
3. List out all the visual differences between master and branch using **P4Merge tool**.
4. Merge the source branch to the trunk.
5. Observe the logging after merging using **“git log –oneline –graph –decorate”**
6. Delete the branch after merging with the trunk and observe the git status.

The outputs are as follows :



