**3. Git-HOL**

**Objective:**Q1: What is branching and merging in Git?  
Branching allows developers to create separate lines of development within a project. It lets you work on new features or changes without affecting the main code (usually called master or main).Merging is the process of combining changes from one branch (like a feature branch) back into another branch (like master), integrating the work into the main project.

Q2: What is a branch request in GitLab?  
A branch request in GitLab typically refers to the action of creating a new branch from an existing one. This is usually done before making changes or working on new features, allowing teams to organize code changes independently before merging them back.

Q3: What is a merge request in GitLab?  
A merge request in GitLab is a request to merge code from one branch into another (usually into master). It allows team members to review, discuss, and approve changes before integrating them into the main project. This ensures better collaboration, version control, and code quality.

**Branching Steps**

1. Navigate to Git project folder:

cd ~/GitDemo

2. Create and switch to a new branch:

git checkout -b GitNewBranch

3. Confirm active branch:

git branch

4. Create file.txt with content:

echo "This is a test file created in GitNewBranch" > new\_branch\_file.txt

5. Add and commit the new file:

git add new\_branch\_file.txt

git commit -m "Added new\_branch\_file.txt in GitNewBranch"

6. Check git status:

git status

**Merging Steps**

1. Switch back to master branch:

git checkout master

2. View command-line differences:

git diff master GitNewBranch

3. View visual differences using P4Merge:

git difftool master GitNewBranch

4. Merge GitNewBranch into master:

git merge GitNewBranch

5. View commit graph after merge:

git log --oneline --graph --decorate

6. Delete the merged branch:

git branch -d GitNewBranch

7. Final status check:

git status

Output:  


  
  
