3. Git-HOL

**Objectives**

* Explain Git branching and merging.
* Demonstrate how to create and switch branches.
* Understand how to perform merge operations.
* Explain how to create a merge request in GitLab.

**🔷 Prerequisites**

* Git installed and configured.
* P4Merge installed (for visual diffs, optional).
* Local Git repository initialized.
* GitLab or GitHub account (optional for merge requests).

**🔷 Tools Used**

* Git Bash
* Notepad++ (for editing)
* P4Merge (not used in this session, but recommended)

**🔹 1. Initialize Git Repository**

mkdir git-branch-lab

cd git-branch-lab

git init

**🔹 2. Create and Commit an Initial File**

bash

Copy code

echo "Initial file" > main.txt

git add .

git commit -m "Initial commit on main"

**🔹 3. Create and Switch to New Branch**

bash

Copy code

git branch GitNewBranch

git checkout GitNewBranch

**🔹 4. Add a File in the Branch and Commit It**

bash

Copy code

echo "This is a new branch file" > branchfile.txt

git add .

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

**🔹 5. Switch Back to Master and Merge the Branch**

bash

Copy code

git checkout master

git merge GitNewBranch

✅ Merge successful using fast-forward.

**🔹 6. View Git Log as Graph**

bash

Copy code

git log --oneline --graph --decorate

Shows the visual commit history including the merged branch.

**🔹 7. Delete the Branch After Merge**

bash

Copy code

git branch -d GitNewBranch



