## **GIT HANDSON 3**

## **Objectives**

1. **Explain Branching and Merging**

**Branching** is the process of creating a separate line of development in your Git repository. Think of it as taking a snapshot of your code so you can make changes without affecting the main version (commonly called main or master). This allows you to safely develop new features, fix bugs, or experiment.

**Merging** is how you combine changes from one branch back into another (usually back into main). Once your work in a branch is finished and tested, you use a merge to update the main codebase with those new changes.

1. **Explain About Creating a Branch Request in GitLab**

The process means **creating a new branch** so you can work separately from the main codebase:

1. **Go to your GitLab project**.
2. Navigate to the project’s **Repository** > **Branches**.
3. Click **New branch**.
4. Enter a name for your new branch (like feature-login or bugfix-issue42).
5. Choose which branch to base it on (usually main).
6. Click **Create branch**.

Now you have your own branch, and any commits you make will stay here until you’re ready to merge.

1. **Explain About Creating a Merge Request in GitLab**

A **merge request** (MR) in GitLab is how you ask to merge your branch’s changes into another branch (typically main). It’s also where team members can review your code, suggest changes, and approve the update. Here’s how to create one:

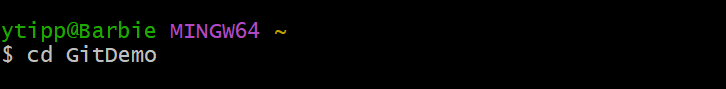
1. **Push your branch** to GitLab if you’ve been working locally.
2. In your project, go to **Merge Requests**.
3. Click **New merge request**.
4. Select your source branch (with your new changes) and the target branch (often main).
5. Fill in the title and description—explain what your changes do.
6. Assign reviewers if required.
7. Click **Submit merge request**.

Your teammates can now review and discuss your changes before approving and merging them.

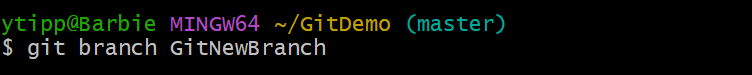
**BRANCHING**

**Steps**

**Step – 1:** Navigate to your repository



**Step – 2:** Create a new branch



**Step – 3:** List all the local branches so you can see GitNewBranch and your current branch (with \* next to it)

**A black background with yellow and blue text

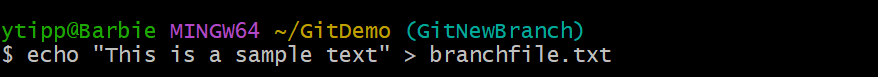
AI-generated content may be incorrect.**

**Step – 4:** Switch to the new branch

A screen shot of a computer code

AI-generated content may be incorrect.

**Step – 5:** Create a new file in this branch



**Step – 6:** Add the new file to staging and commit it

A screen shot of a computer

AI-generated content may be incorrect.

**Step – 7:** Check the branch status to confirm there’s nothing left to commit

A black screen with white text

AI-generated content may be incorrect.

**MERGING**

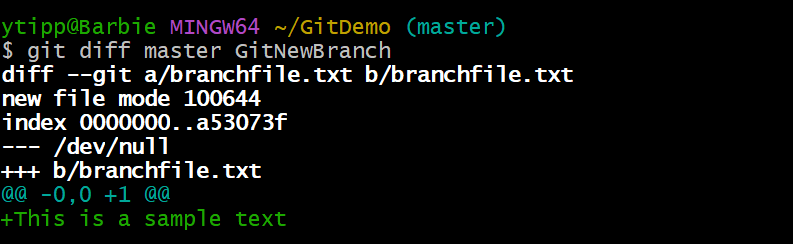
**Steps**

**Step – 1:** Switch back to the **master** branch

A black background with white text

AI-generated content may be incorrect.

**Step – 2:**  List the differences between **master** and **GitNewBranch** in the command line



**Step – 3:**  Merge **GitNewBranch** into **master**

A screen shot of a computer code

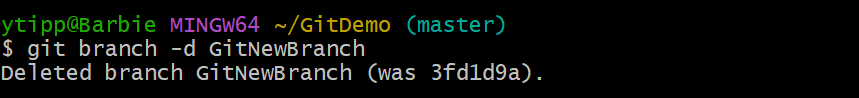
AI-generated content may be incorrect.

**Step – 4**: View the commit history in a simple visual format

A screen shot of a computer

AI-generated content may be incorrect.

**Step – 5**: Delete the branch **GitNewBranch** now that it’s merged



**Step – 6**: Push your changes to GitHub

A screen shot of a computer

AI-generated content may be incorrect.

**Gitlab Project**

A screenshot of a computer

AI-generated content may be incorrect.