**Git Hands-On Lab Documentation**

**Objective:**  
 Understand Git branching and merging workflows

* Learn how to create a branch and merge it into the master (trunk)
* Explore GitLab's branch and merge request features
* Use P4Merge for visual diffing during merges

**Prerequisites**

Ensure the following are ready before starting:

* Git environment installed and configured
* P4Merge tool installed and set as Git’s diff/merge tool
* A local Git repository initialized
* A remote GitLab repository connected

**What is Branching in Git?**

Branching allows developers to diverge from the main codebase (usually master or main) to work on features or fixes independently.

Benefits:

- Isolated development

- Parallel workflows

- Safe experimentation

**What is Merging in Git?**

Merging integrates changes from one branch into another. Typically, feature branches are merged into master after testing.

**Lab Objectives**

By the end of this lab, you will be able to:

* Initialise .gitignore file to stop unwanted files like .log to get store in gitlab

**Steps :**

1. Create a New Branch
2. Switch to the New Branch Add Files and Make Changes
3. Commit the ChangesCheck Git Status
4. Switch Back to Master & List CLI Differences
5. View Visual Differences with p4merge
6. Merge the branches