**Git Hands-On Lab Documentation**

**Objective:**  
- Understand how merge conflicts occur in Git

- Learn how to resolve conflicts using Git and a 3-way merge tool

- Use P4Merge for visual conflict resolution

- Update .gitignore to exclude backup files

**Prerequisites**

Before starting, ensure the following are ready:

* Git environment installed and configured
* P4Merge tool installed and integrated with Git
* Hands-on ID: Git-T03-HOL\_001
* A local Git repository initialized and connected to a remote

**What is a Merge Conflict?**

A merge conflict occurs when Git cannot automatically reconcile differences between two branches. This typically happens when the same file is modified in both branches in overlapping lines.

**Lab Objectives**

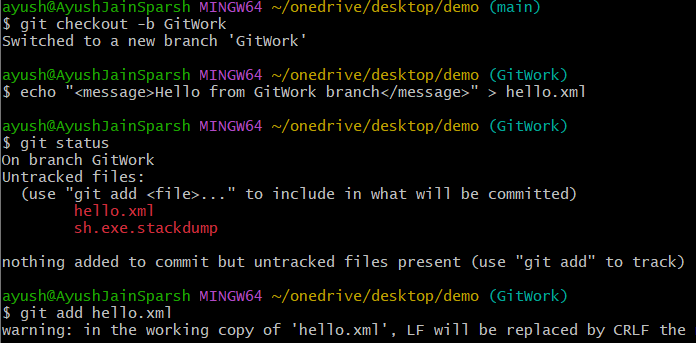
By the end of this lab, you will be able to:

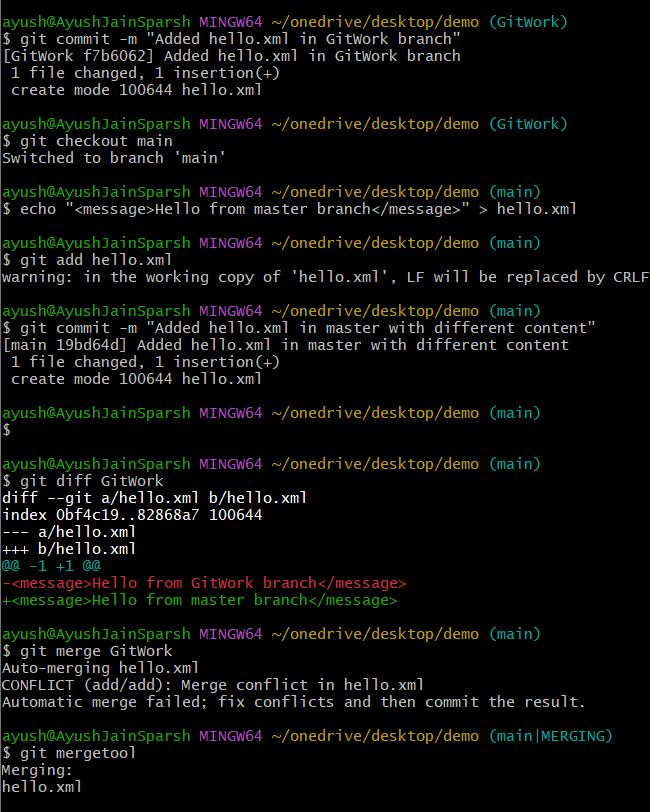
* Initialise .gitignore file to stop unwanted files like .log to get store in gitlab

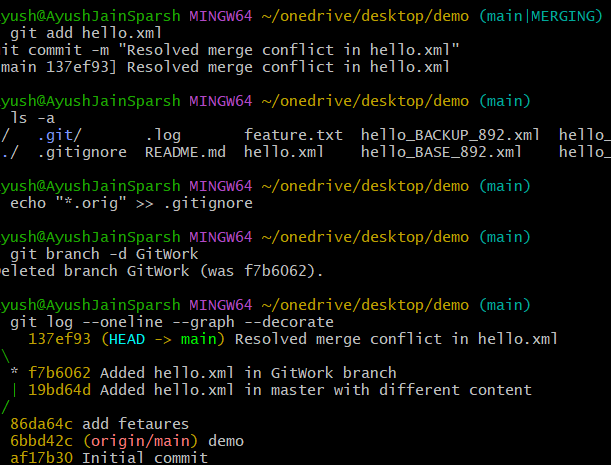
**Steps :**

1. Verify main is clean
2. Create a new branch. Add and modify hello.xml in branch
3. Commit the ChangesCheck Git Status
4. Switch Back to main and add conflicting hello.xml in main
5. Commit changes and check difference
6. Merge the branches
7. Resolve conflict using 3 way merge tool
8. Add backup file to .gitignore
9. Commit and list all branches and delete final log

**Output Snapshots :**

****

****

****