**GIT HOL**

**Git Handson 4**

**Merge Conflict Resolution**

Handle merge conflicts when multiple users modify the same file in different branches and how to resolve them using Git and visual tools.

**Verify master branch is clean**

git checkout master

git status

**Create a branch and add hello.xml**

git branch GitWork

git checkout GitWork

echo "<greeting>Hello from GitWork branch</greeting>" > hello.xml

git add hello.xml

git commit -m "Add hello.xml in GitWork branch"

**Update hello.xml**

echo "<greeting>Updated from GitWork branch</greeting>" > hello.xml

git status

git add hello.xml

git commit -m "Update hello.xml in GitWork branch"

**Switch back to master**

git checkout master

**Add different hello.xml in master and commit**

echo "<greeting>Hello from master branch</greeting>" > hello.xml

git add hello.xml

git commit -m "Add hello.xml in master branch"

**View commit log graph**

git log --oneline --graph --decorate --all

**Check differences with Git diff tool**

git diff GitWork master

**Use P4Merge for visual diff**

git merge GitWork

**Merge GitWork into master**

git merge GitWork

**Resolve conflict**

Open hello.xml, find conflict markers (<<<<<<<, =======, >>>>>>>)

Edit file to desired final content

Save the file

**Mark conflict resolved and commit**

git add hello.xml

git commit -m "Resolve merge conflict in hello.xml"

**Check git status**

git status

**Add backup files to .gitignore**

echo "\*.bak" >> .gitignore

git add .gitignore

git commit -m "Ignore backup files"

**List all branches**

git branch

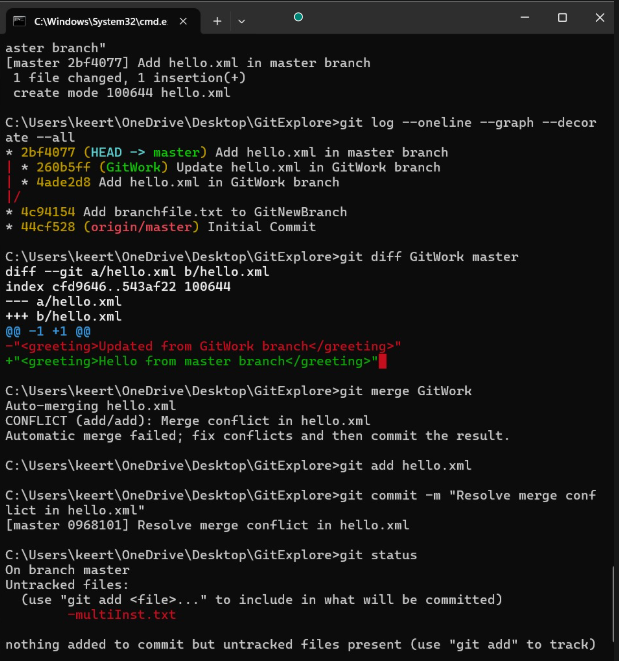
**Delete the merged branch**

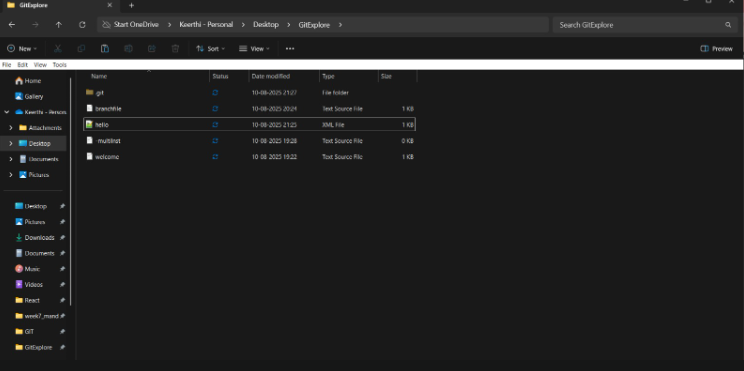
git branch -d GitWork

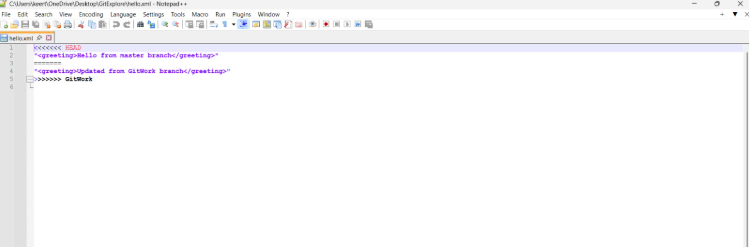
**Observe final commit log**

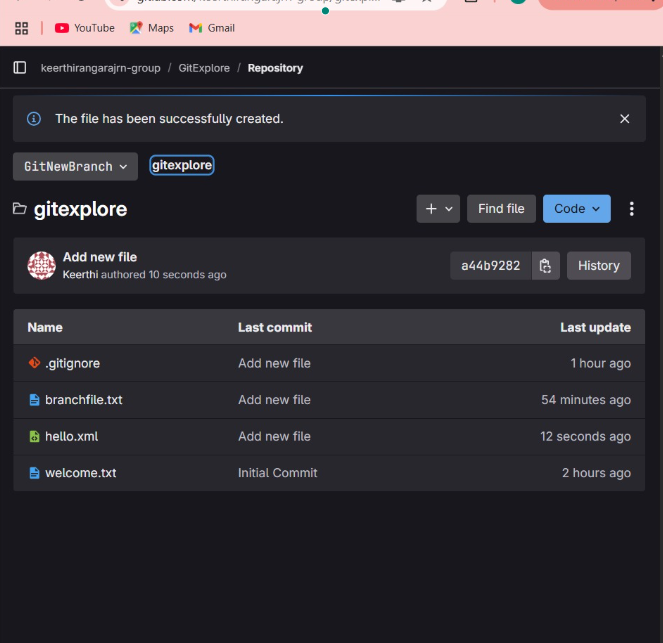
git log --oneline --graph –decorate

**Output:**

****

****

****

****