**ReactJS HOL-4**

**Questions & Answers –**

1. **Explain how to resolve the conflict during merge.**

**Ans -** A merge conflict occurs when Git cannot automatically combine changes from different branches because the same part of a file was modified in both branches.

**Steps to Resolve a Merge Conflict**

1. **Identify the Conflict**
   * After running git merge <branch>, Git will stop and show which files have conflicts.
   * Use git status to see the list of files with conflicts.
2. **Open the Conflicted File**
   * In the file, Git will mark conflicts like this:

Ex: <<<<<<< HEAD

Your changes on the current branch

=======

Changes from the branch being merged

>>>>>>> branch-name

1. **Edit the File to Resolve the Conflict**

* Decide whether to:
  + Keep your changes
  + Keep the other branch’s changes
  + Combine both changes
* Remove the conflict markers (<<<<<<<, =======, >>>>>>>) after editing.

1. **Mark the Conflict as Resolved**

git add <file>

1. **Complete the Merge**

git commit

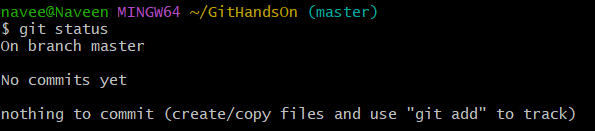
* Git will use a default merge commit message unless you change it.

1. **Verify the Result**

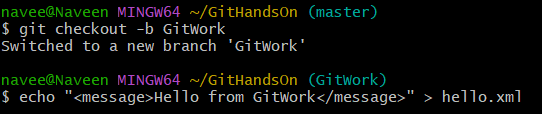
* Test your code to ensure everything works as expected.

**Hands-On –**

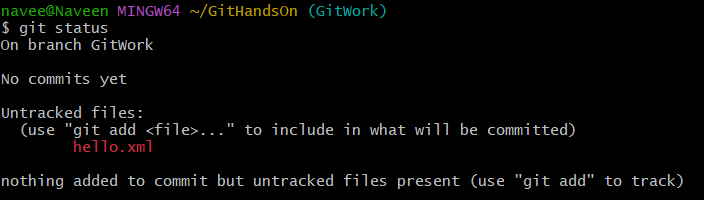
1. **Verify if master is in clean state.**

****

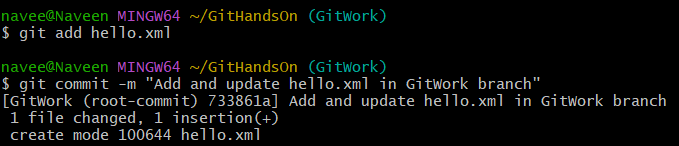
1. **Create a branch “GitWork”. Add a file “hello.xml”.**

****

1. **Update the content of “hello.xml” and observe the status**

****

1. **Commit the changes to reflect in the branch**

****

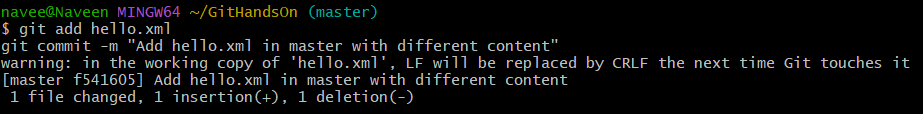
1. **Switch to master.**

****

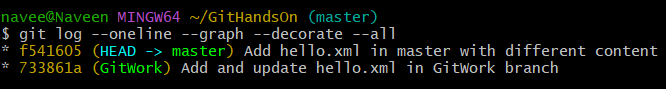
1. **Add a file “hello.xml” to the master and add some different content than previous.**

****

1. **Commit the changes to the master**

****

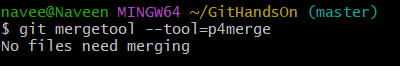
1. **Observe the log by executing “git log –oneline –graph –decorate –all”**

****

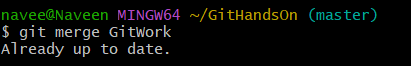
1. **Check the differences with Git diff tool**

****

1. **For better visualization, use P4Merge tool to list out all the differences between master and branch**

****

1. **Merge the bran to the master**

****

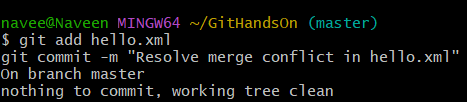
1. **Observe the git mark up.**

****

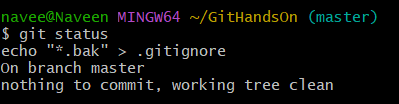
1. **Use 3-way merge tool to resolve the conflict**

****

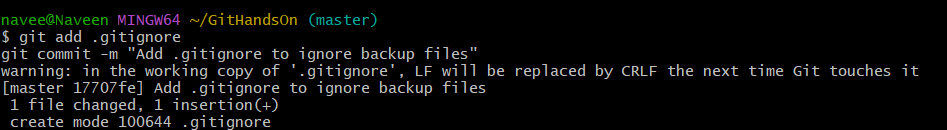
1. **Commit the changes to the master, once done with conflict**

****

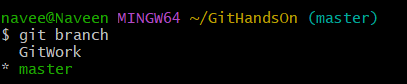
1. **Observe the git status and add backup file to the .gitignore file.**

****

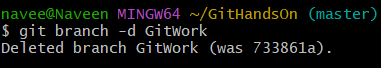
1. **Commit the changes to the .gitignore**

****

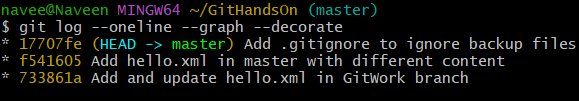
1. **List out all the available branches**

****

1. **Delete the branch, which merge to master.**

****

1. **Observe the log by executing “git log –oneline –graph –decorate”**

****