Please follow the instructions to complete the hands-on. Each instruction expect a command for the Git Bash.

* 1. Verify if master is in clean state.
  2. Create a branch **“GitWork”.** Add a file “hello.xml”.
  3. Update the content of “hello.xml” and observe the status
  4. Commit the changes to reflect in the branch
  5. Switch to master.
  6. Add a file **“hello.xml”** to the master and add some different content than previous.
  7. Commit the changes to the master
  8. Observe the log by executing **“git log –oneline –graph –decorate –all”**
  9. Check the differences with Git diff tool
  10. For better visualization, use P4Merge tool to list out all the differences between master and branch
  11. Merge the bran to the master
  12. Observe the git mark up.
  13. Use 3-way merge tool to resolve the conflict
  14. Commit the changes to the master, once done with conflict
  15. Observe the git status and add backup file to the .gitignore file.
  16. Commit the changes to the .gitignore
  17. List out all the available branches
  18. Delete the branch, which merge to master.
  19. Observe the log by executing **“git log –oneline –graph –decorate”**

**Repository link:**  
<https://github.com/rishi-demo/GitDemo>

## **Step-by-Step Procedure**

### **1. Verify master branch status**

bash

Copy code

git checkout master

git status

### **2. Create branch GitWork and add hello.xml**

bash

Copy code

git checkout -b GitWork

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

git add hello.xml

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

### **3. Update hello.xml in GitWork**

bash

Copy code

echo "<message>Updated content in GitWork branch</message>" > hello.xml

git add hello.xml

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

### **4. Switch to master and add different hello.xml**

bash

Copy code

git checkout master

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

git add hello.xml

git commit -m "Add hello.xml in master with different content"

### **5. View commit history graphically**

bash

Copy code

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

### **6. Check differences between branches**

bash

Copy code

git diff master GitWork

### **7. Merge GitWork into master**

bash

Copy code

git merge GitWork

Encountered merge conflict in hello.xml.

### **8. Resolve merge conflict manually**

* Open hello.xml and edit conflict markers <<<<<<<, =======, >>>>>>>.
* Save the resolved content.
* Stage and commit the resolution:

bash

Copy code

git add hello.xml

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

### **9. Add backup file patterns to .gitignore and commit**

bash

Copy code

echo "\*.bak\n\*~" >> .gitignore

git add .gitignore

git commit -m "Add backup file patterns to .gitignore"

### **10. Delete merged branch**

bash

Copy code

git branch -d GitWork

### **11. Final commit log**

bash

Copy code

git log --oneline --graph –decorate

**OUTPUT:**

\* 5d52a4e (HEAD -> master) Add backup files patterns to .gitignore

\* bf3b7c4 Resolve merge conflict in hello.xml

|\

| \* 45d37dd Add hello.xml in master branch with different content

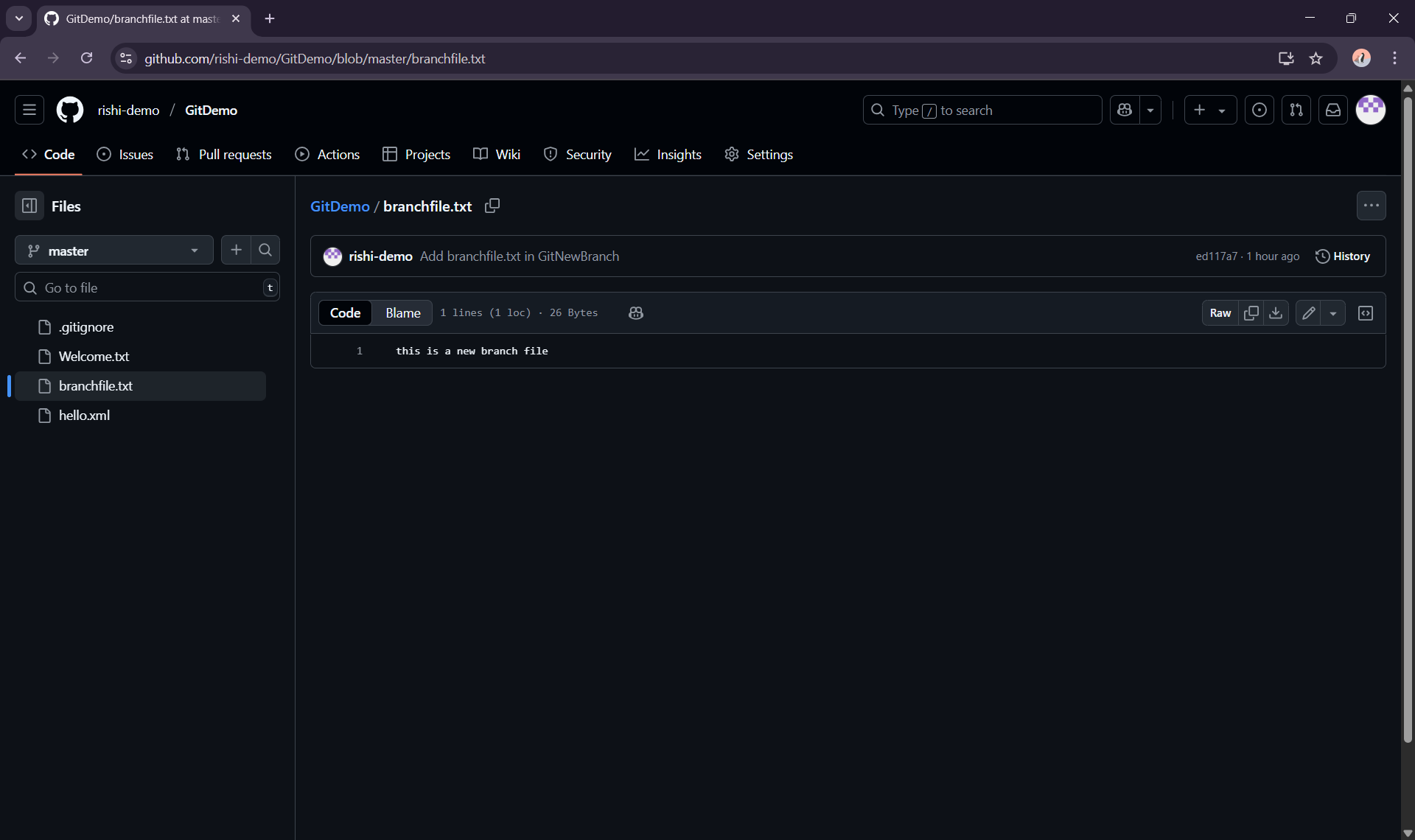
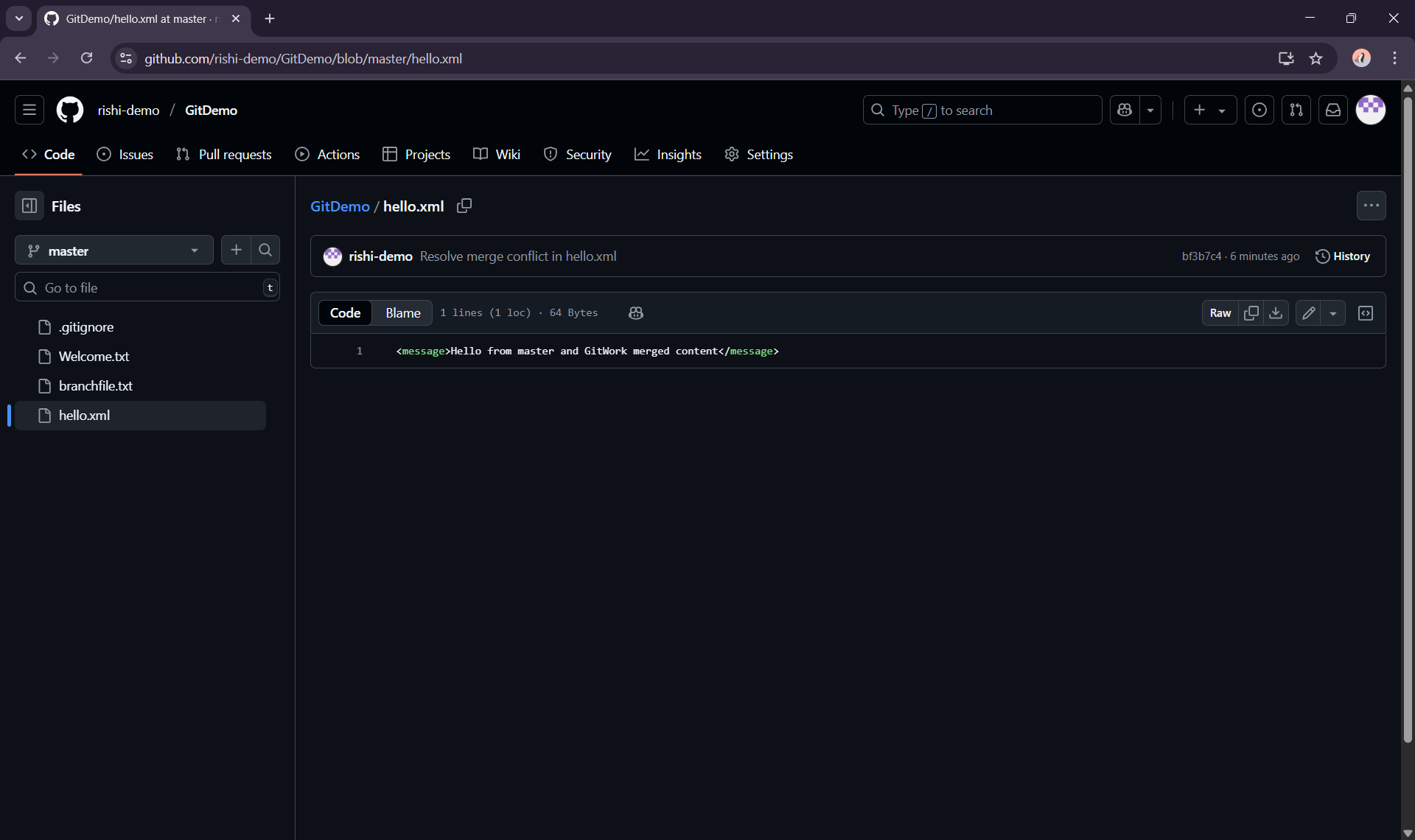
\* | 65abc39 Add hello.xml in master wuth different content

|/

\* ed117a7 (origin/master) Add branchfile.txt in GitNewBranch

\* 5c3de3f Ignore .log files and log folder

\* 7011ccc Add Welcome.txt with welcome message

**OUTPUT**: