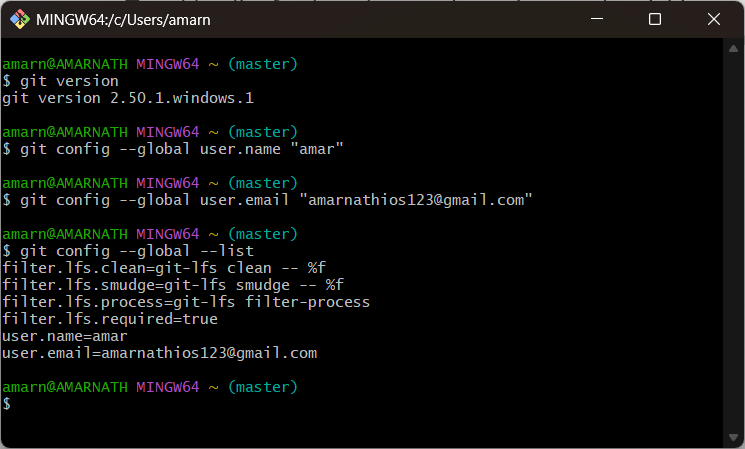
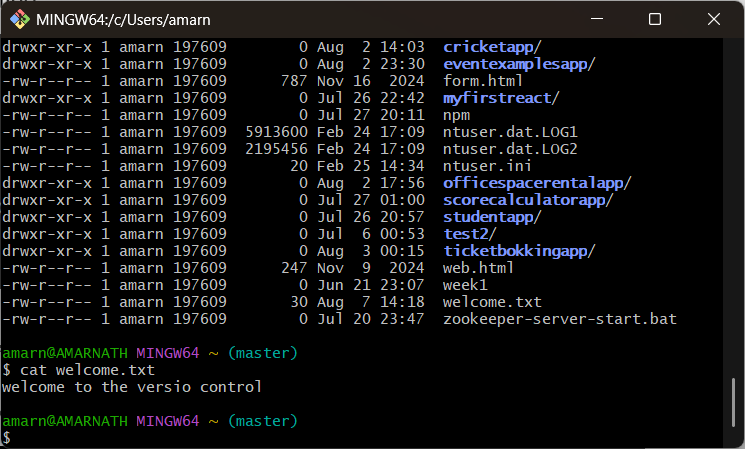
**Step 1:** Setting up your machine with Git Configuration.

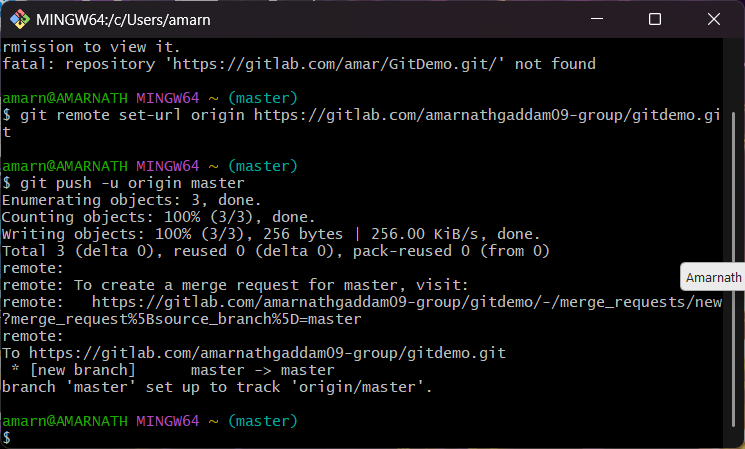
****

****

**Step 3:** Adding a file to source code repository



**welcome.txt is added to the local repository.**

****

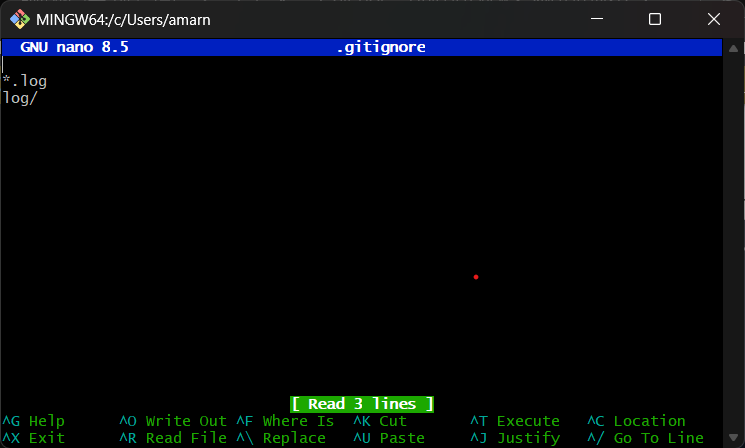
**2: Implementing git ignore command to ignore unwanted files and folders.**

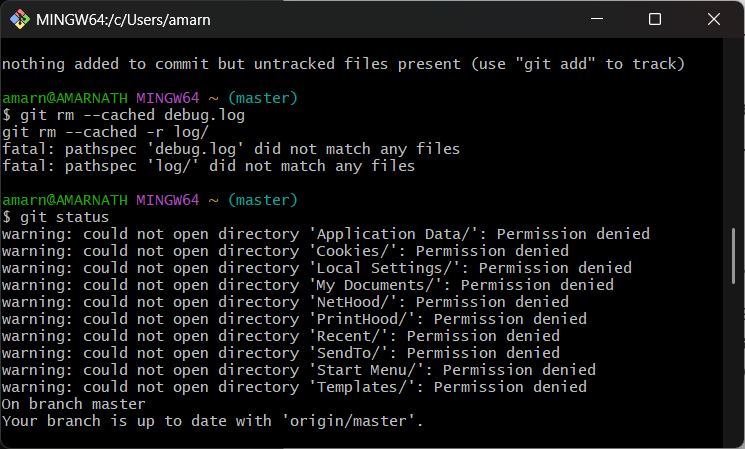
**Step1:After creating the log files :**

touch debug.log

mkdir log

**creating gitnore file in working directory:**

****

**To get clean working folder with ignored files:  
3:** **Setting up Git environment with P4Merge tool for Windows.**

**Branching:**

**1.creating a new branch.**

After installing and setting up the p4.

**Code:**

git branch GitNewBranch

**2.Switch to the new branch**:

**Code:**

git checkout GitNewBranch

**3. Add new files with content**:

**Code:**

echo "This is new content" > file1.txt

git add file1.txt

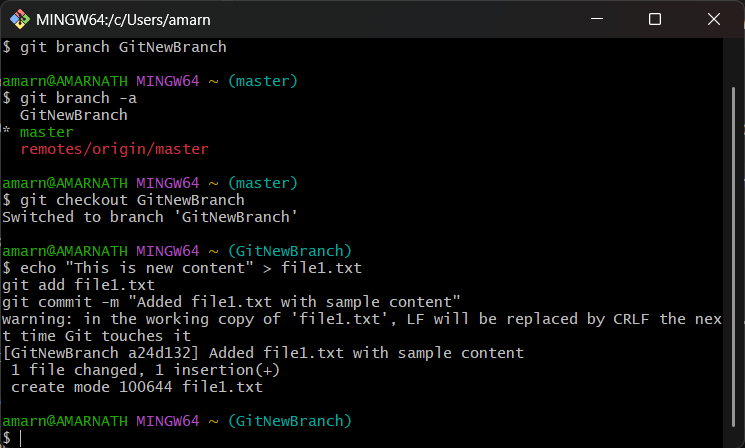
git commit -m "Added file1.txt with sample content"

**4. Check git status:**

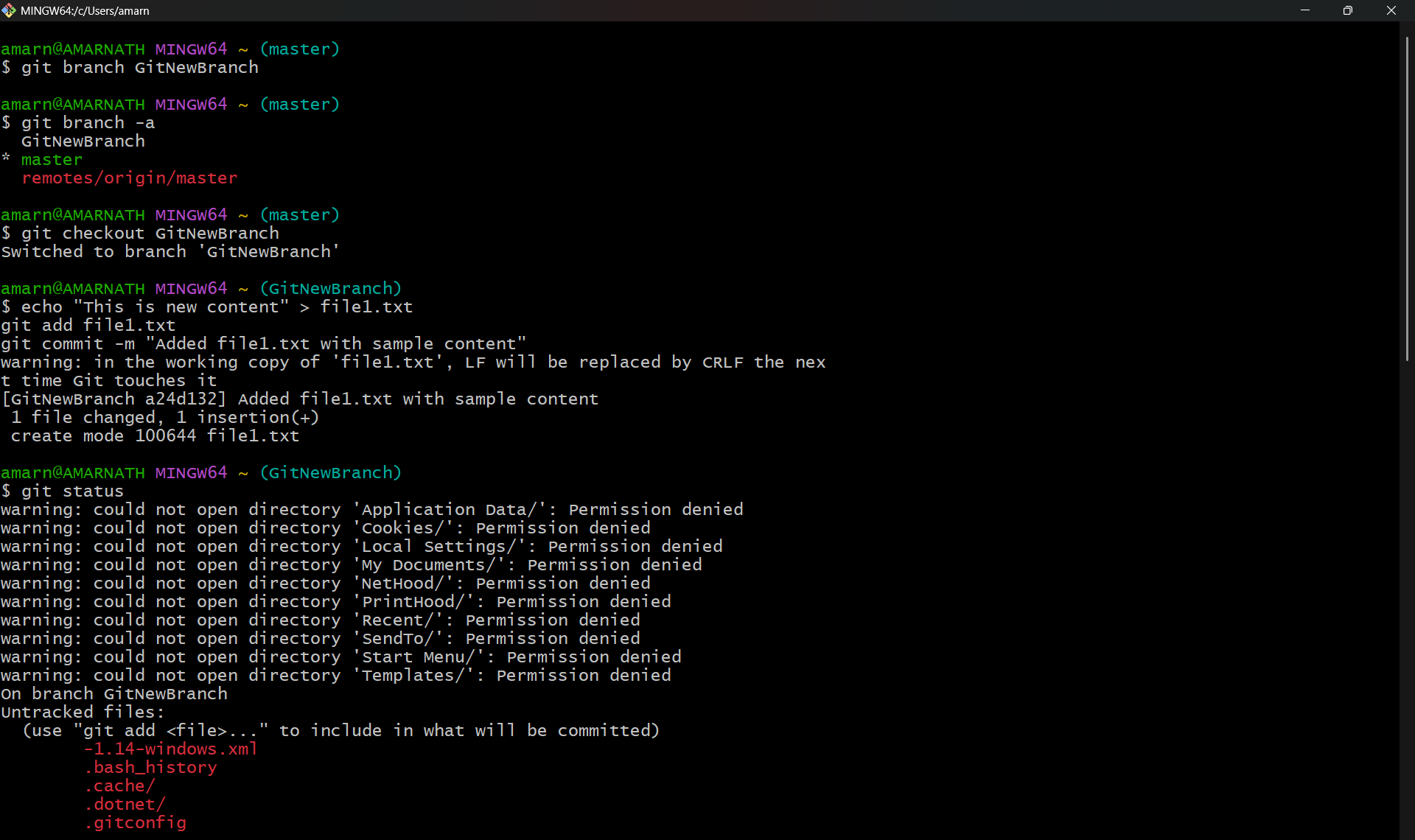
**Code:**

git status.

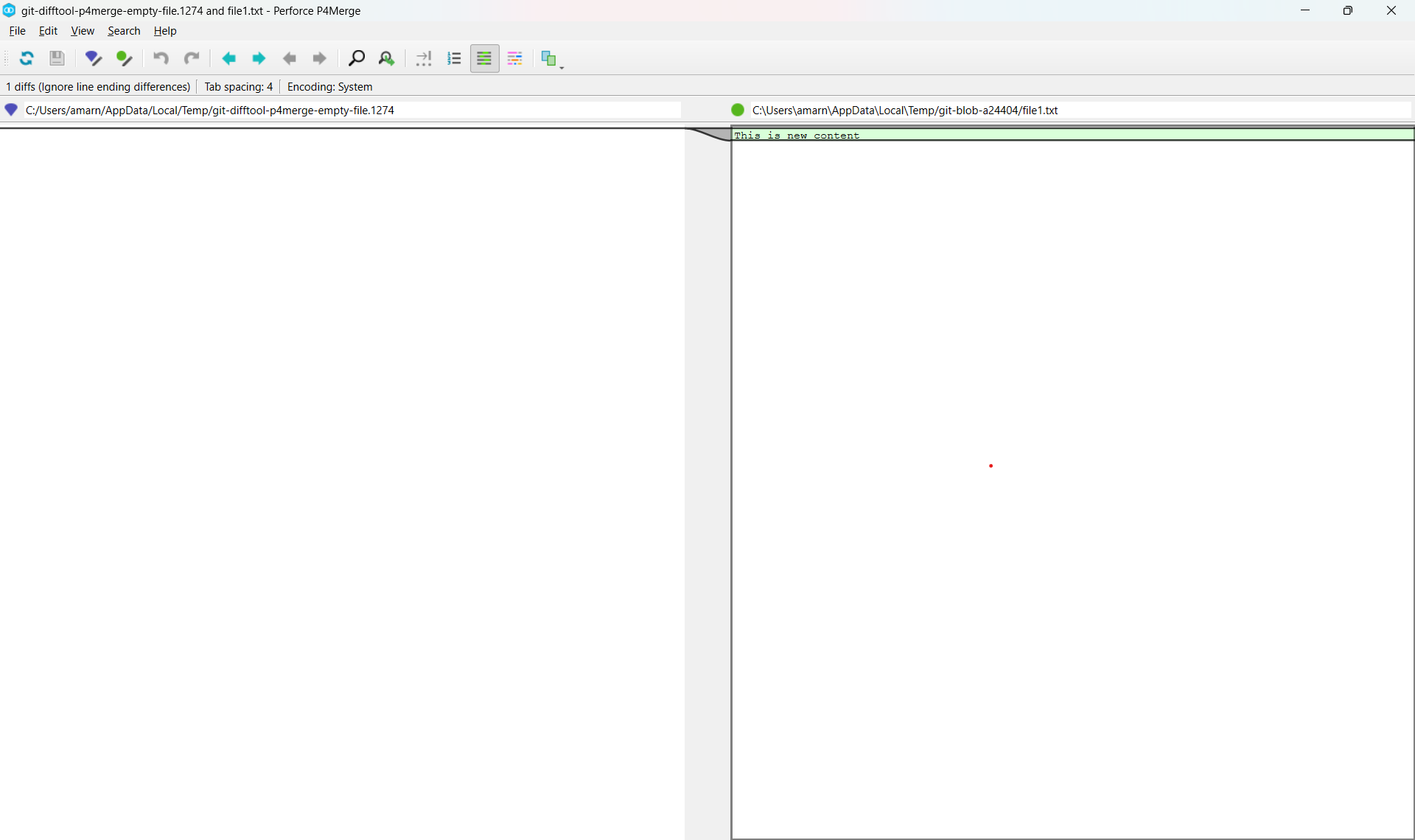
this will check the status



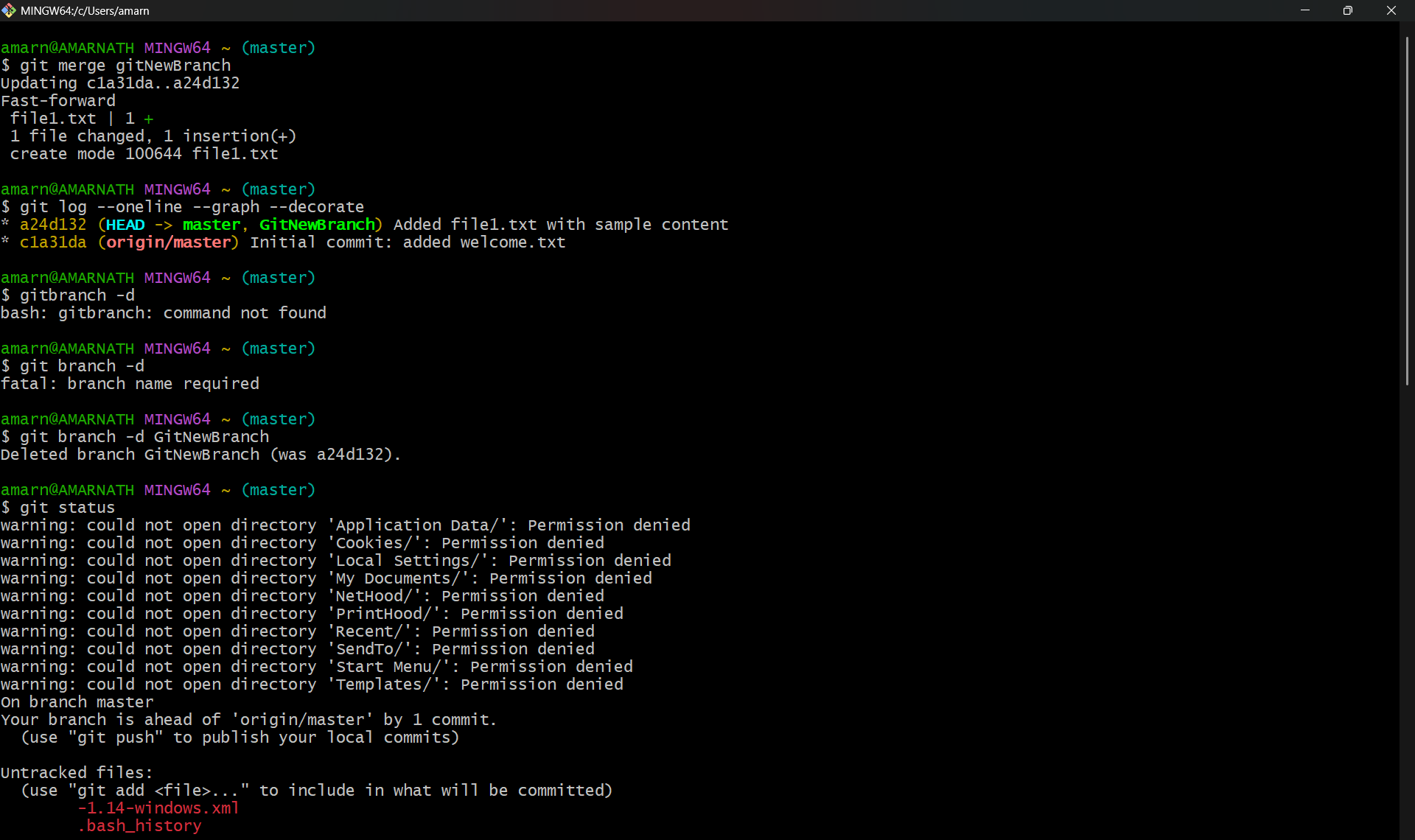
**Merging:**

1. **Switch to the master.**

**P4 merge:**

****

**Merge status:**

****

**4.** **Implementing conflict resolution when multiple users are updating the trunk (or master) in such a way that it results into a conflict with the branch’s modification.**

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

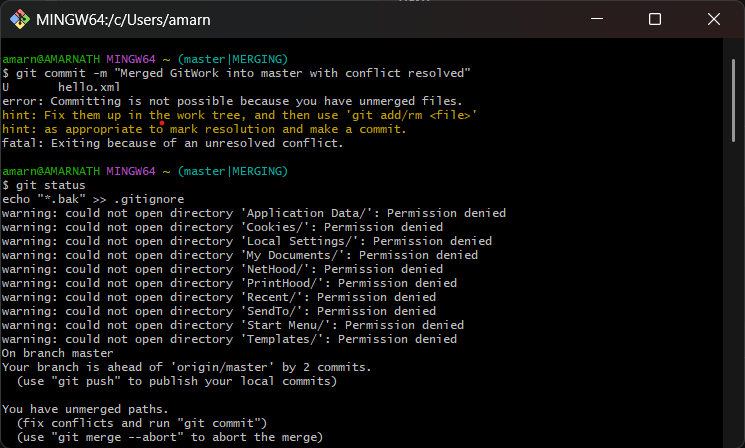
2.Switch to master.

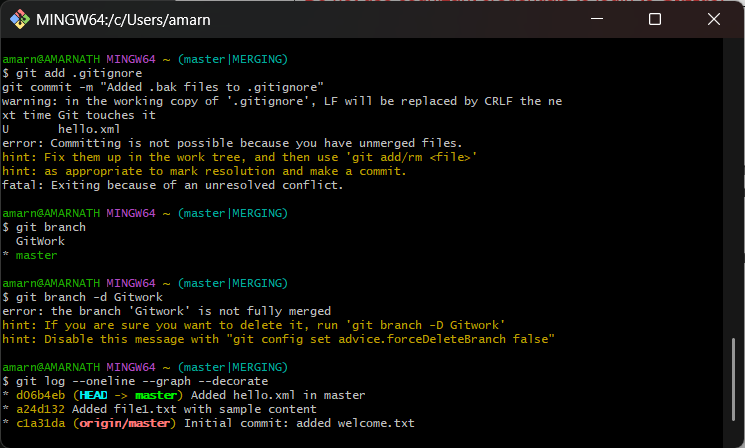
3.Add a file **“hello.xml”** to the master and add some different content than previous

4.Use P4Merge tool to visualize differences

5.Merge the branch to the master

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

****



7. Commit the changes after resolving conflict:

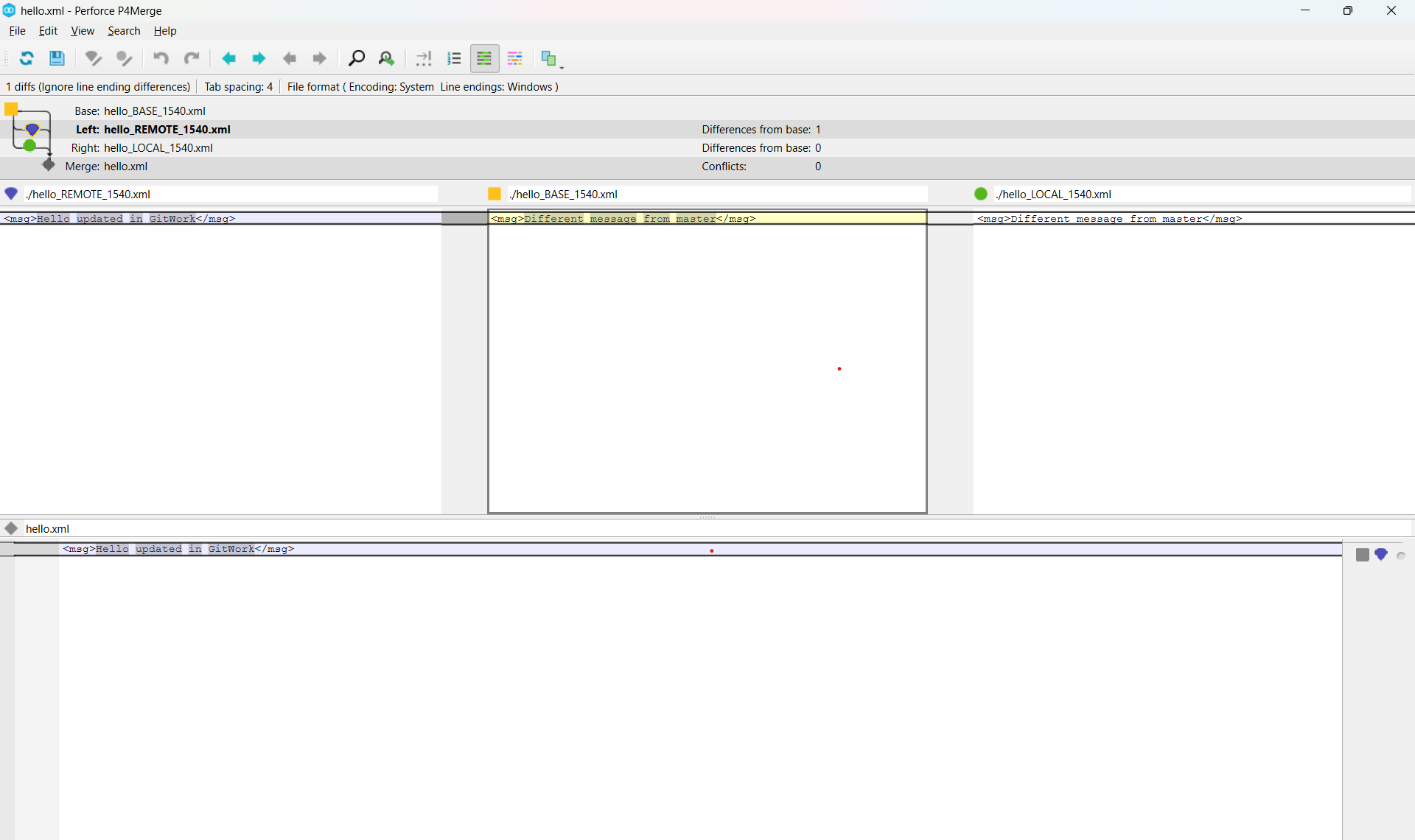
Use 3 way merge tool

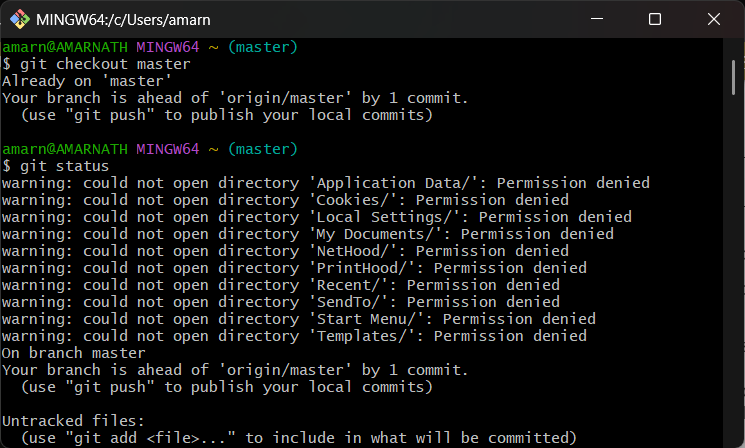
8. Observe git status and add backup file to .gitignore:

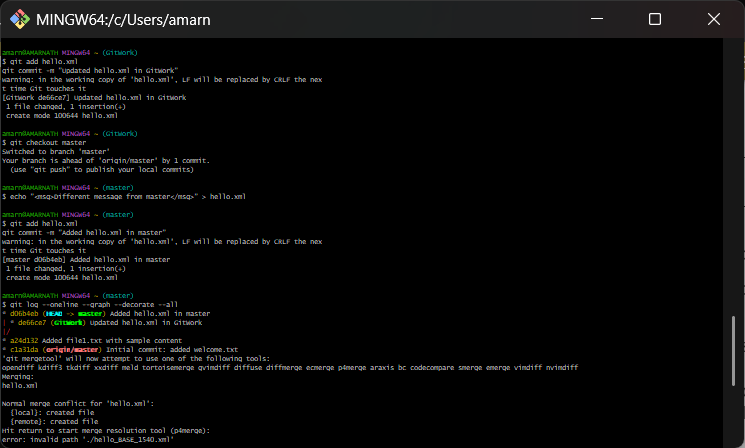
9.after commint the changes List out all available branches:

10. Delete the branch merged to master and observe the log again.

Output:

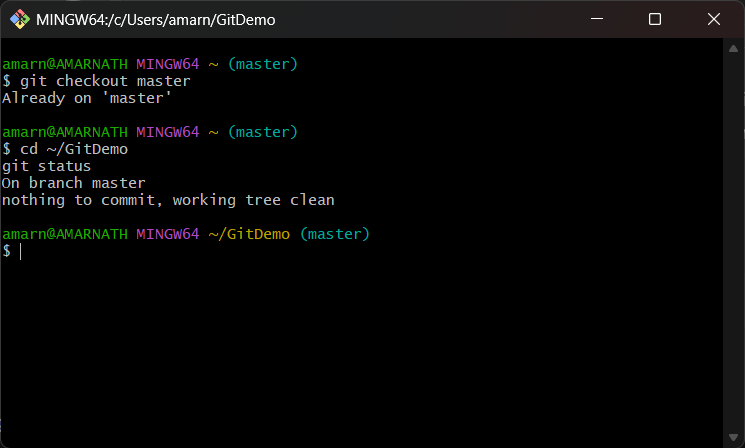




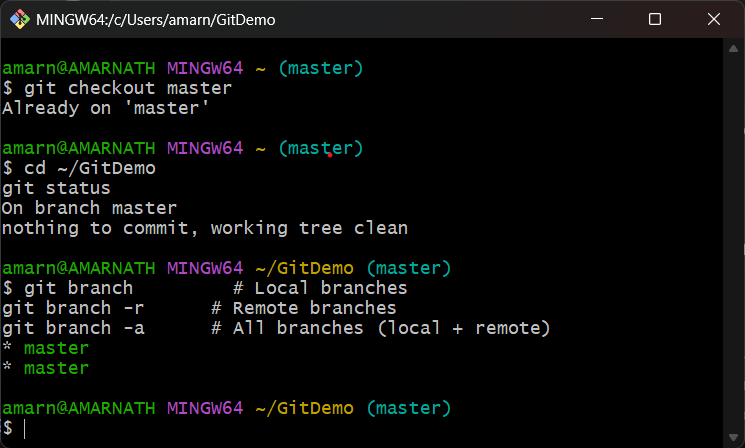
****

**5: executing clean up and push back to remote Git.**

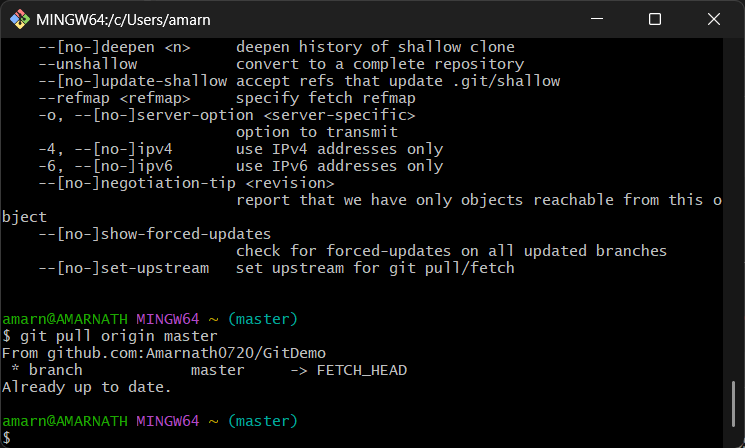
1. Verifying if master is in clean state.



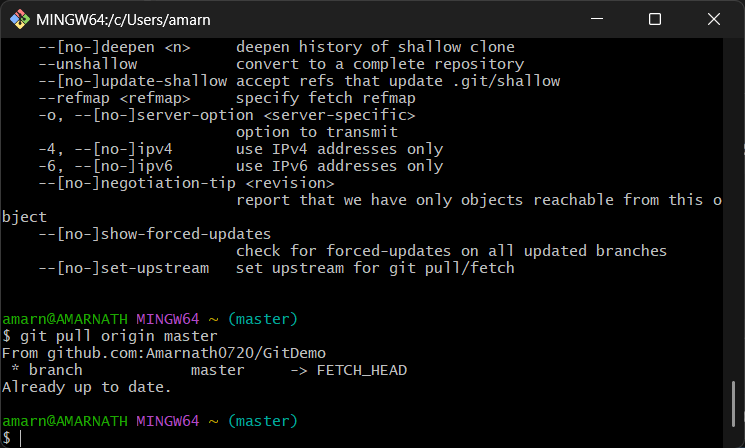
1. Listing all the available branches.



3.Pulling the remote git repository to the master



4.Push the changes, which are pending from Git-T03-HOL\_002 to the remote repository



5. Observe if the changes are reflected in the remote repository

