**WEEK-08**

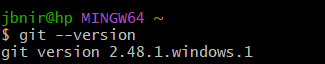
**MODULE-11:VERSION CONTROL - GIT**

**MANDATORY HANDSON**

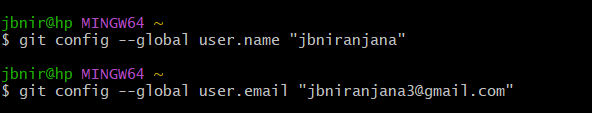
**1.GIT -HOL**

**Step 1: Setup your machine with Git Configuration**

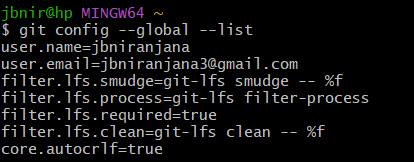
1.To check if Git client is installed properly: Open Git bash shell and execute



2.Configuring user level configuration of user ID and email ID execute



3.Checking if configuration set correctly.



**Step 2: Integrate notepad++.exe to Git and make it a default editor**

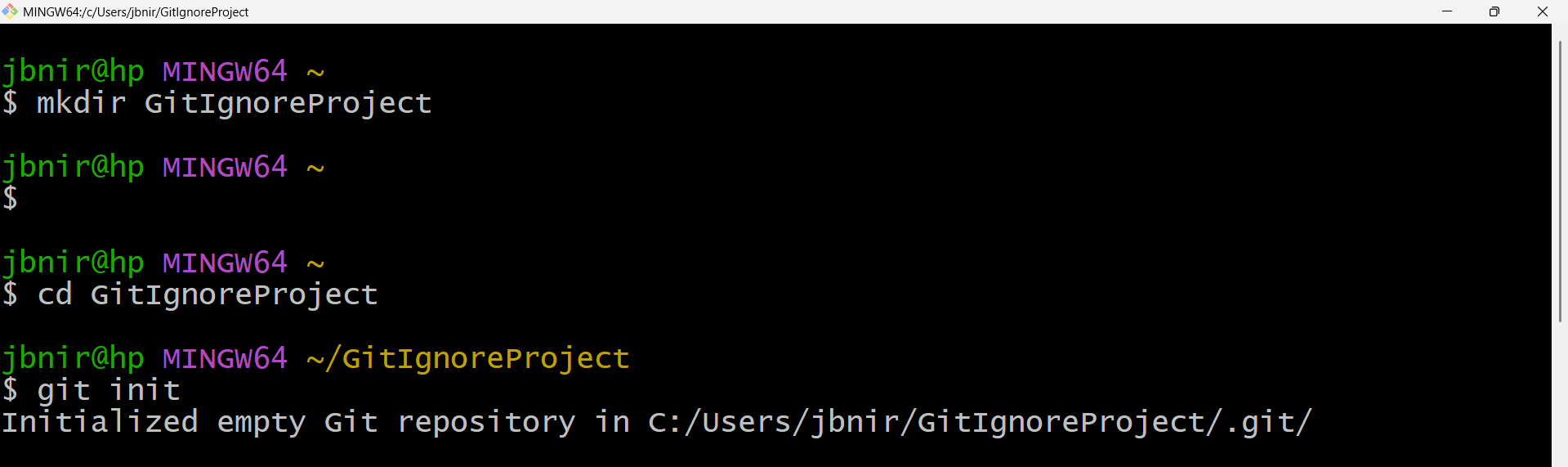
Integrating notepad++ with Git bash

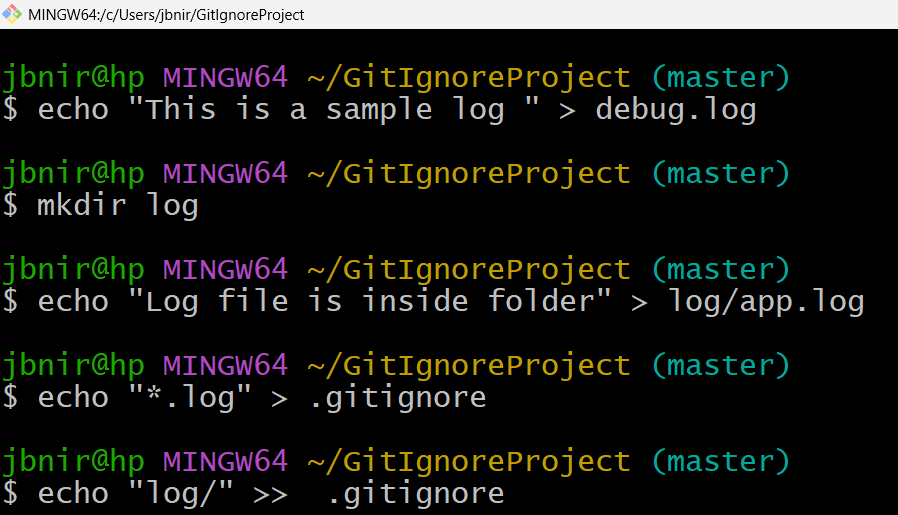


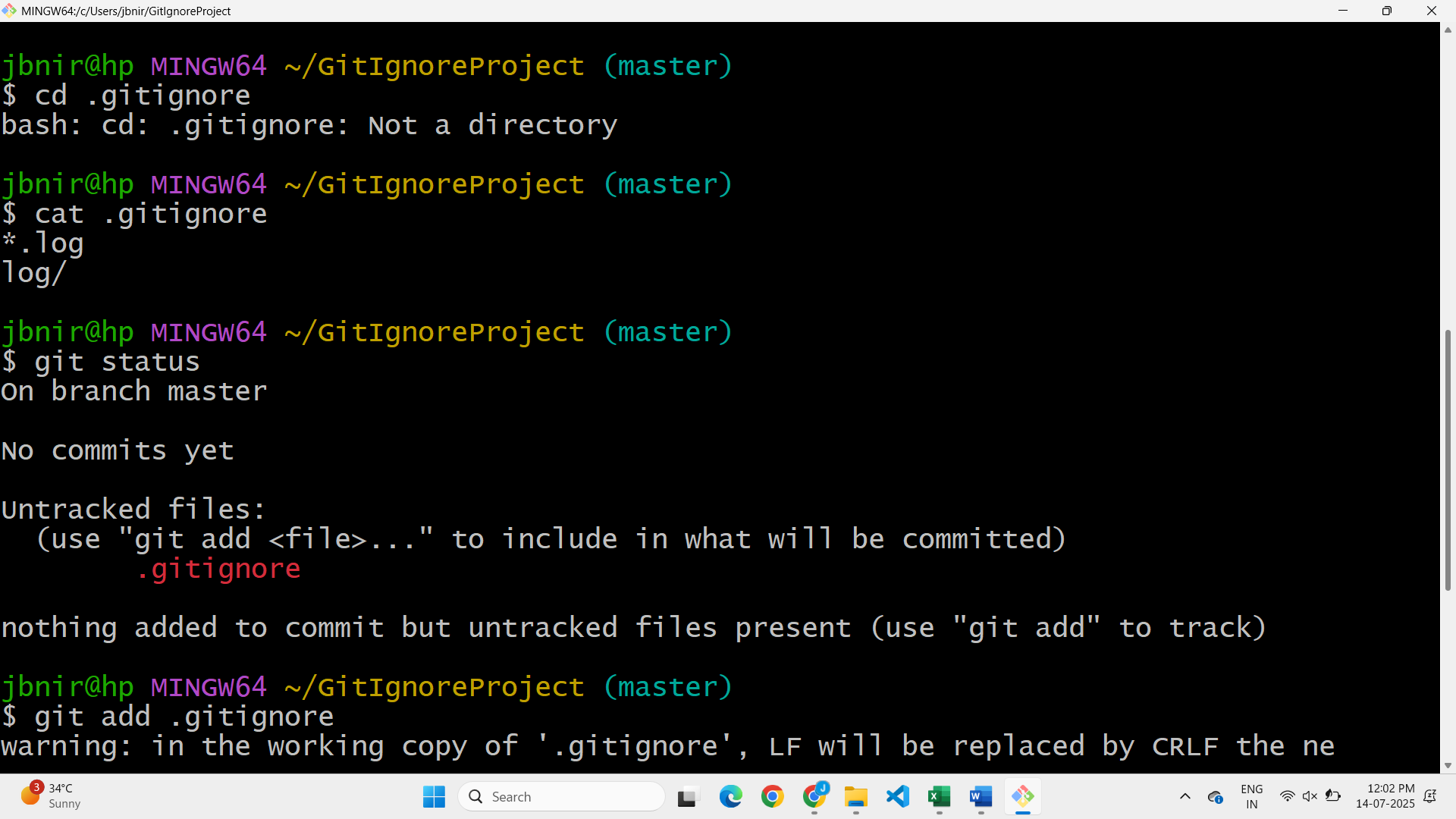
**2.GIT -HOL**

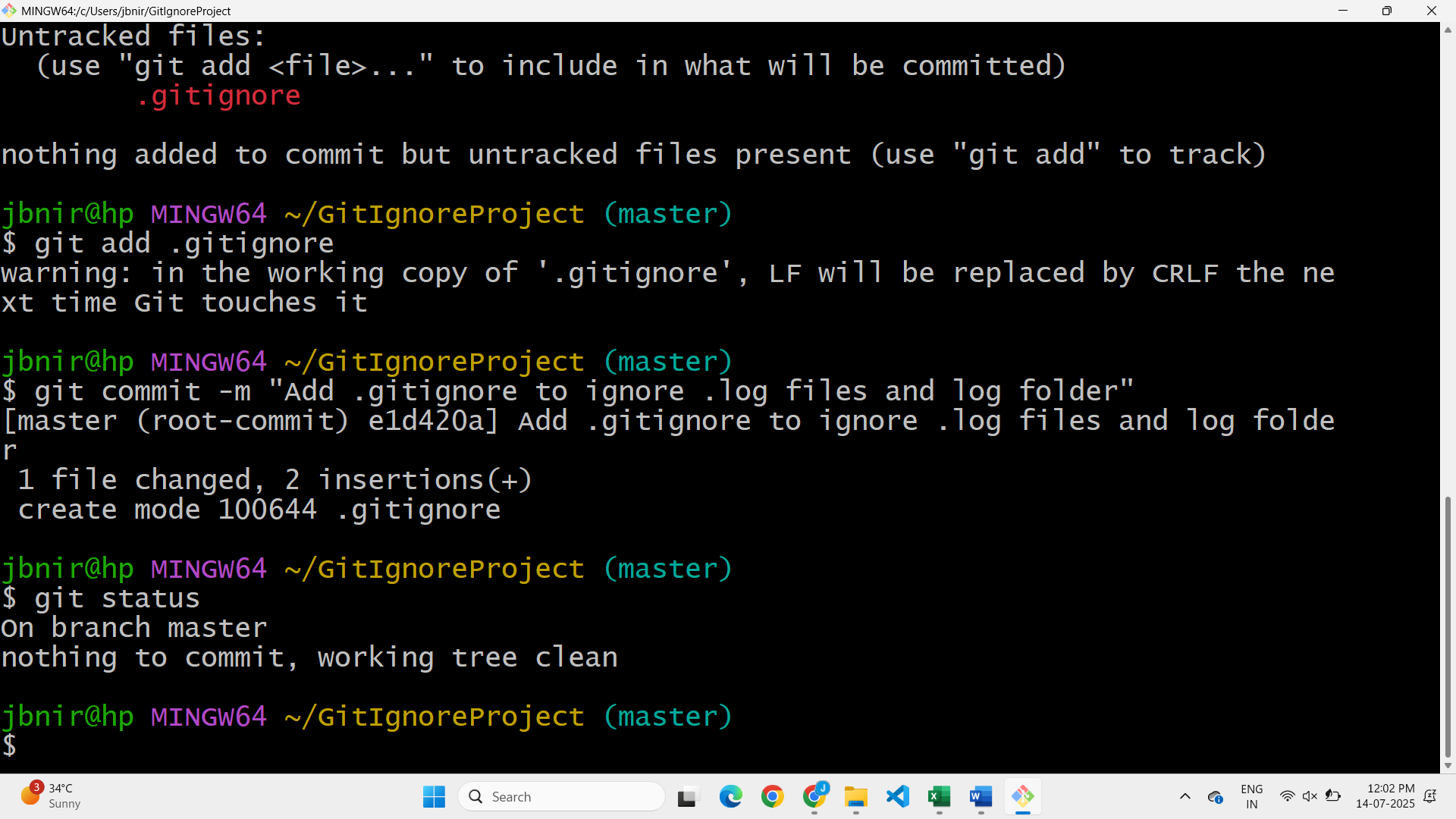
**Create a “.log” file and a log folder in the working directory of Git. Update the .gitignore file in such a way that on committing, these files (.log extensions and log folders) are ignored.**

**Verify if the git status reflects the same about working directory, local repository and git repository**





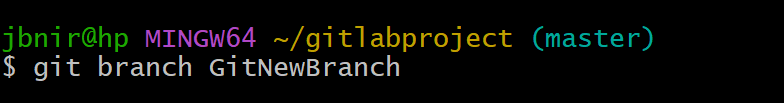




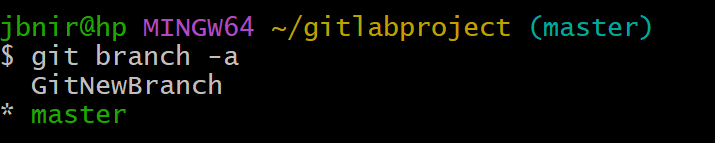
**3.GIT -HOL**

**BRANCHING**

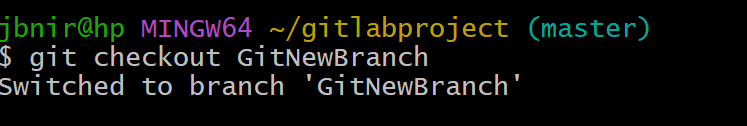
1. Creating a new branch “GitNewBranch”



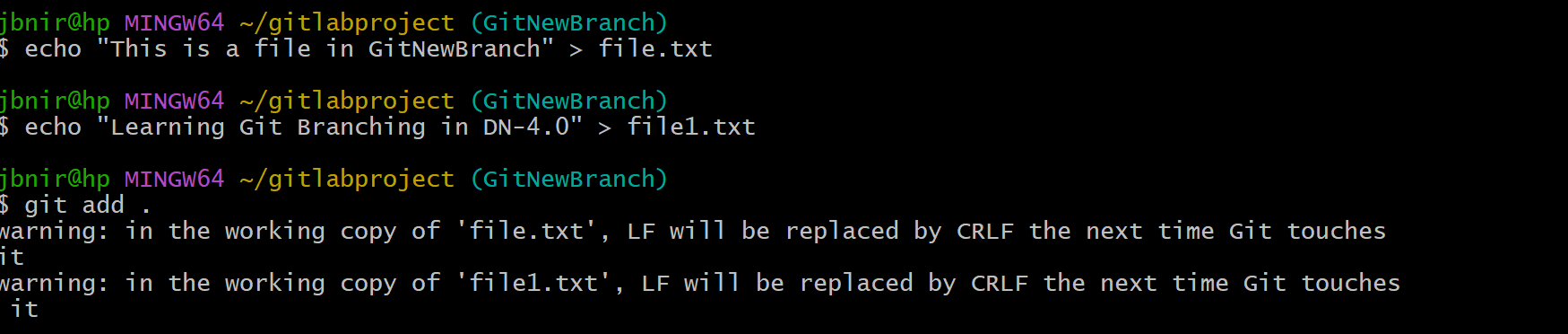
2. Listing all local and remote branches



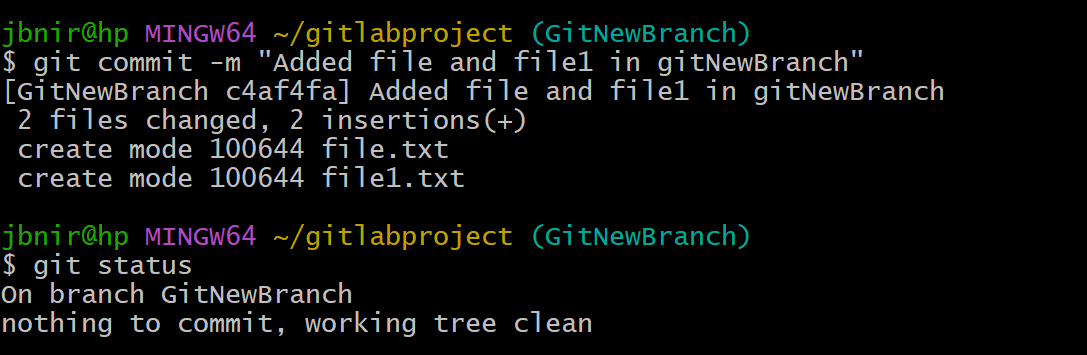
3. Switching to the newly created branch



4. Adding some files with content

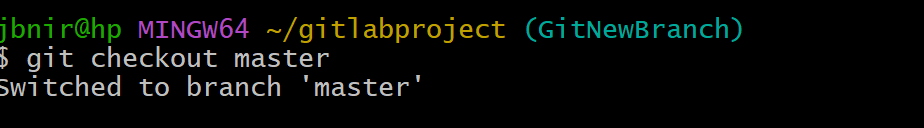


5. Commiting the changes

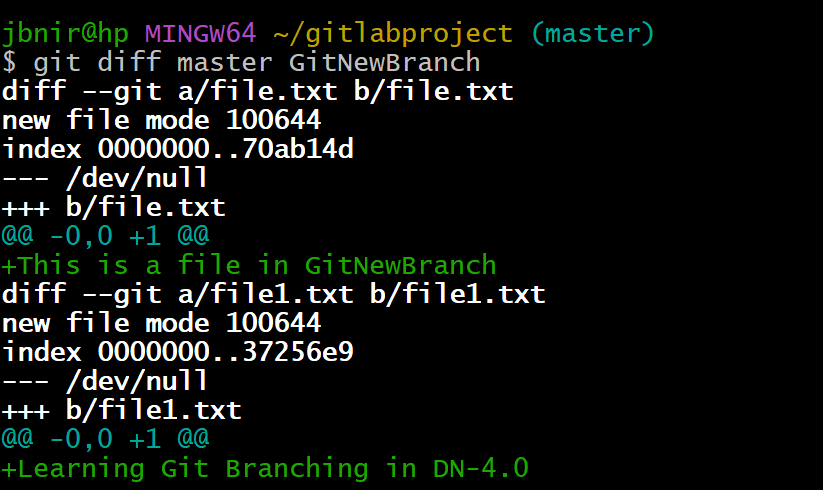


**MERGING**

1. Switching back to master

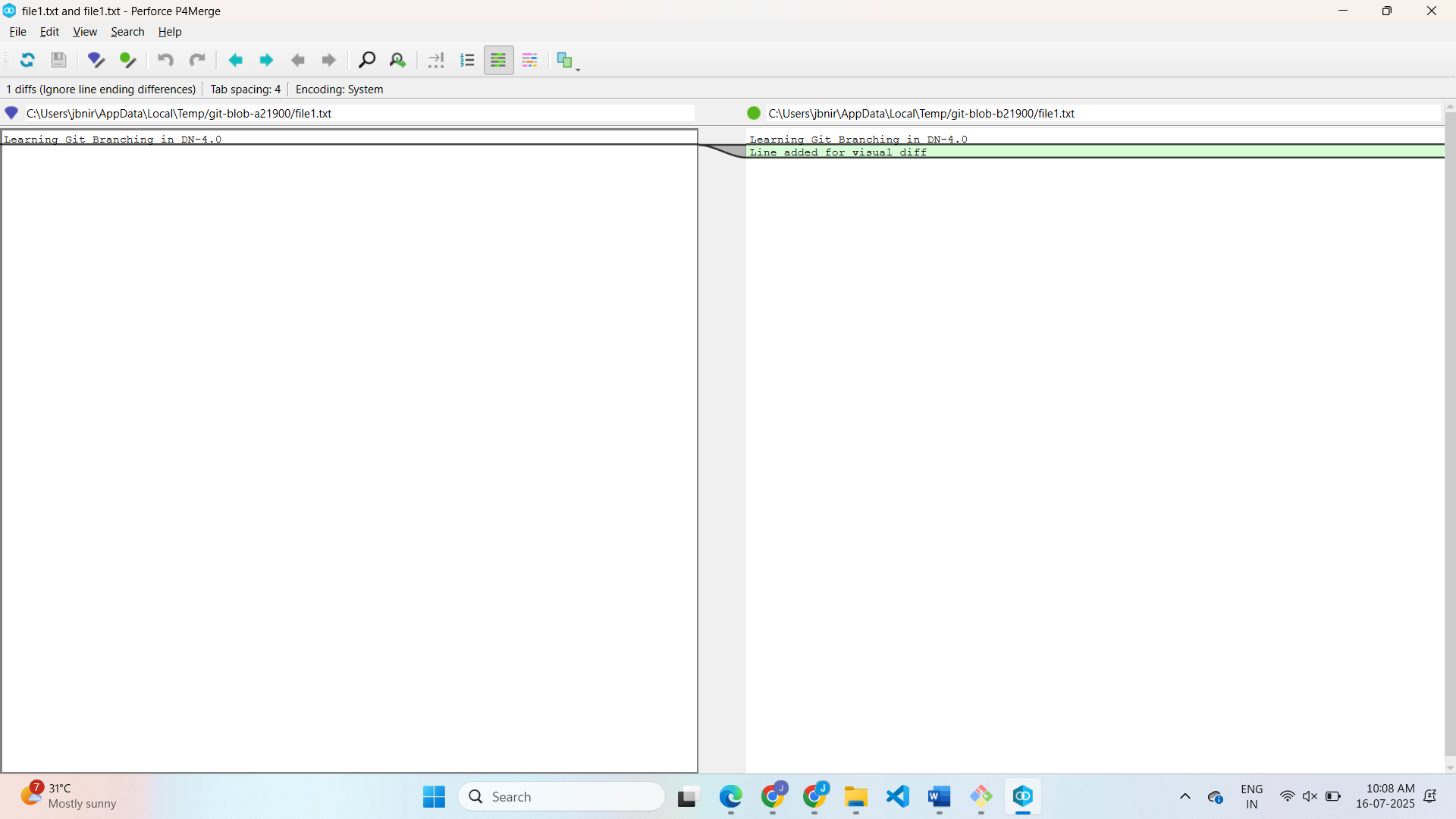


2. Listing all differences between master and GitNewBranch

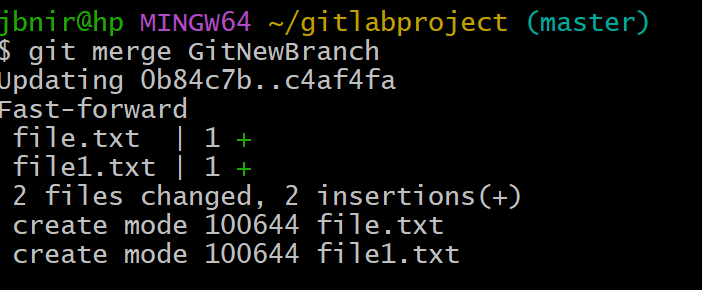


3. Show visual differences after merging with P4Merge tool

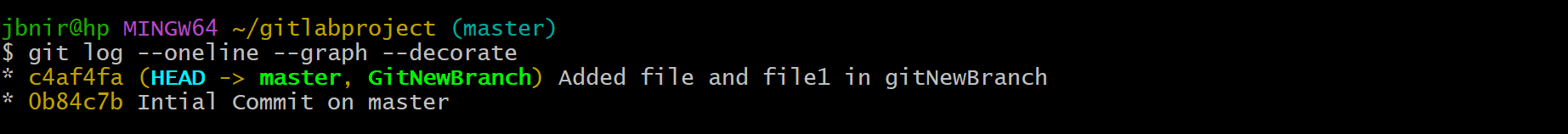




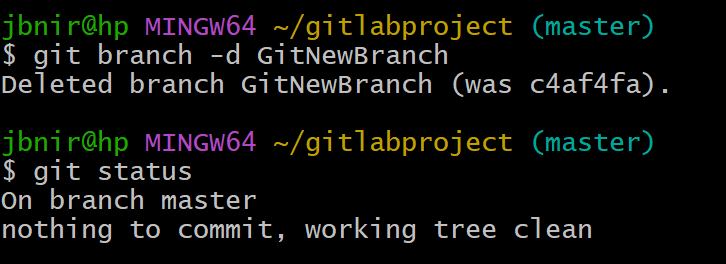
4. Merging GitNewBranch into master



5. Viewing the merged logs in pretty format

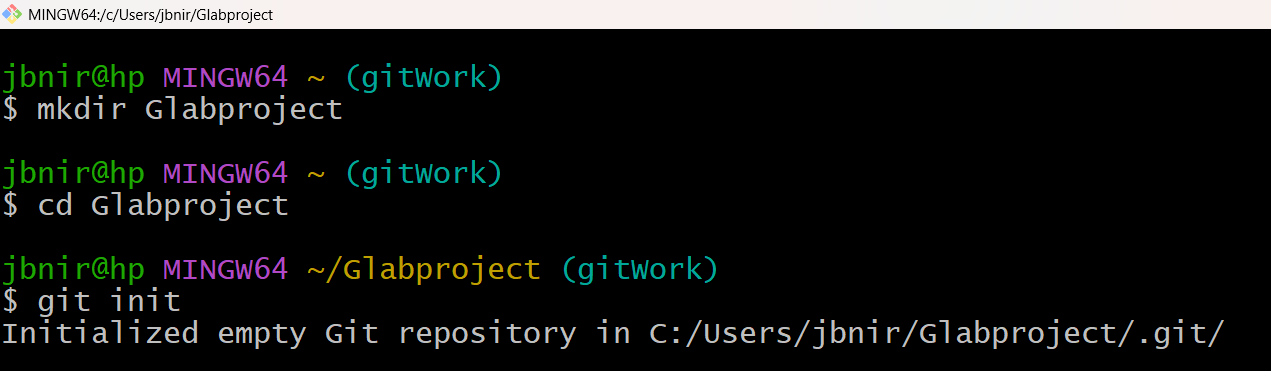


6. Deleting the branch and check status

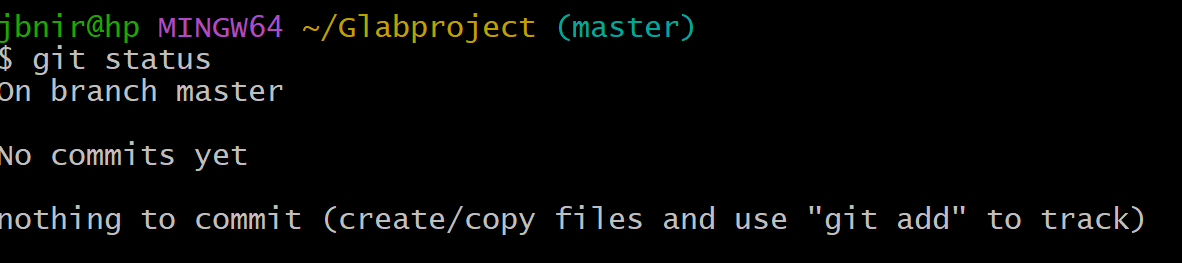


**4.GIT -HOL**

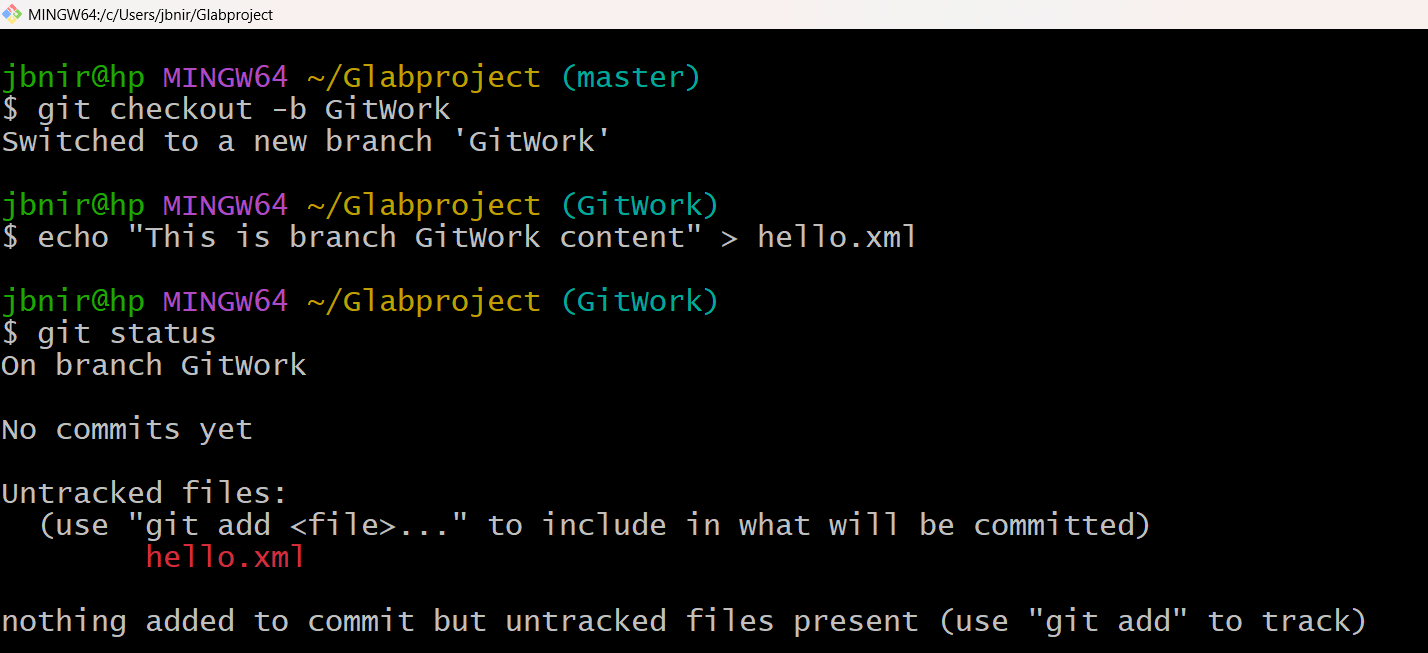
Initial Setup



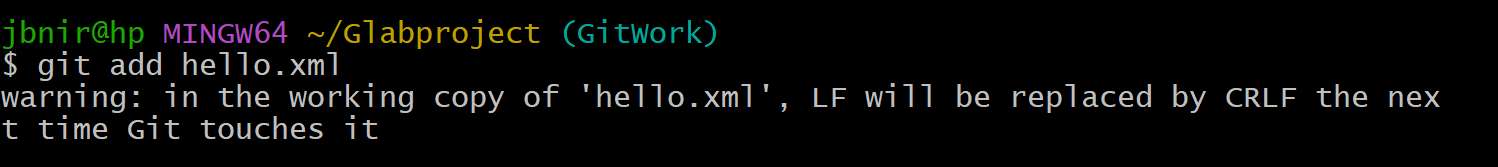
1. Verify if master is in clean state



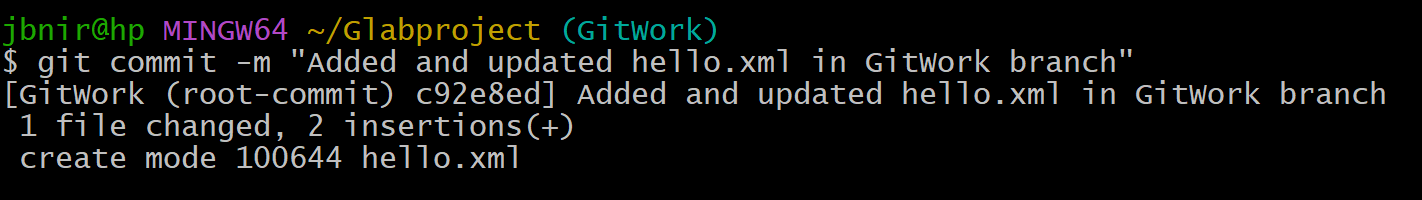
2. Create a branch “GitWork”. Add a file “hello.xml”



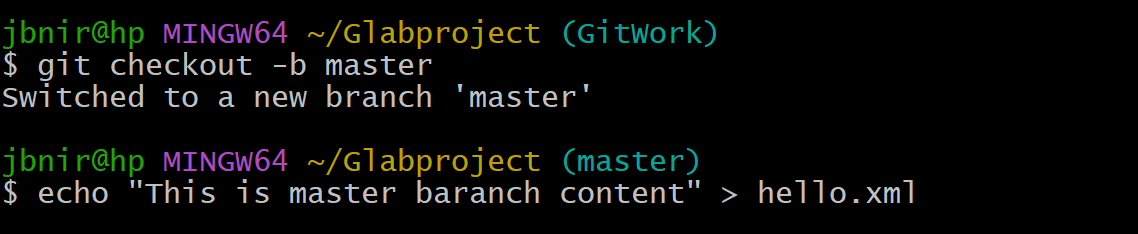
3. Update the content of “hello.xml” and observe the status



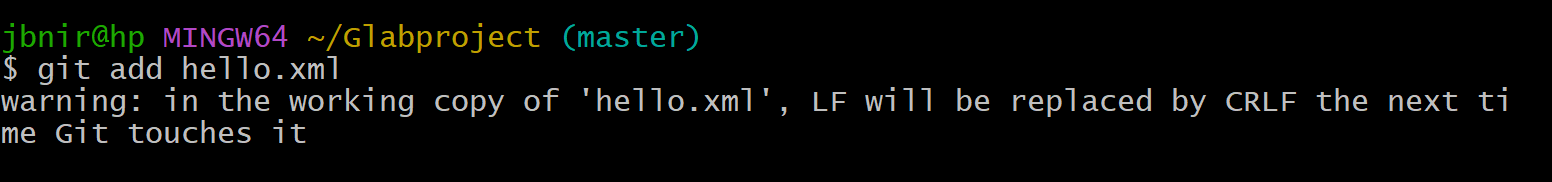
4. Commit the changes to reflect in the branch



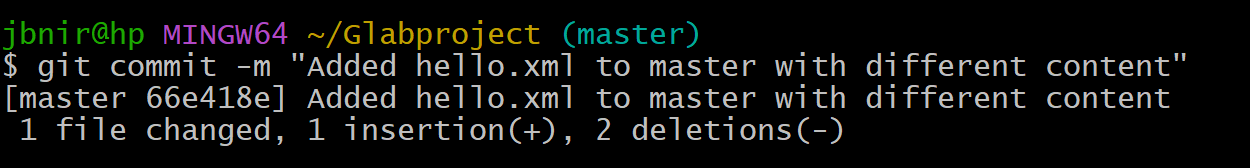
5. Switch to master



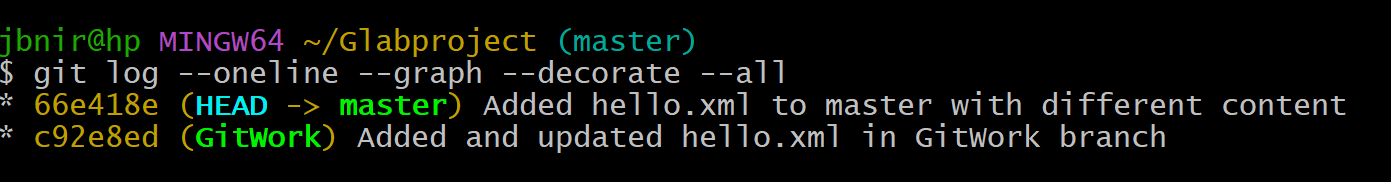
6. Add a file “hello.xml” to the master and add some different content



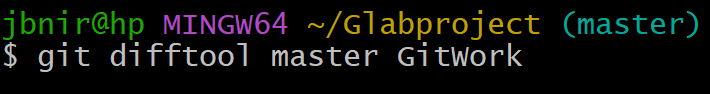
7. Commit the changes to the master



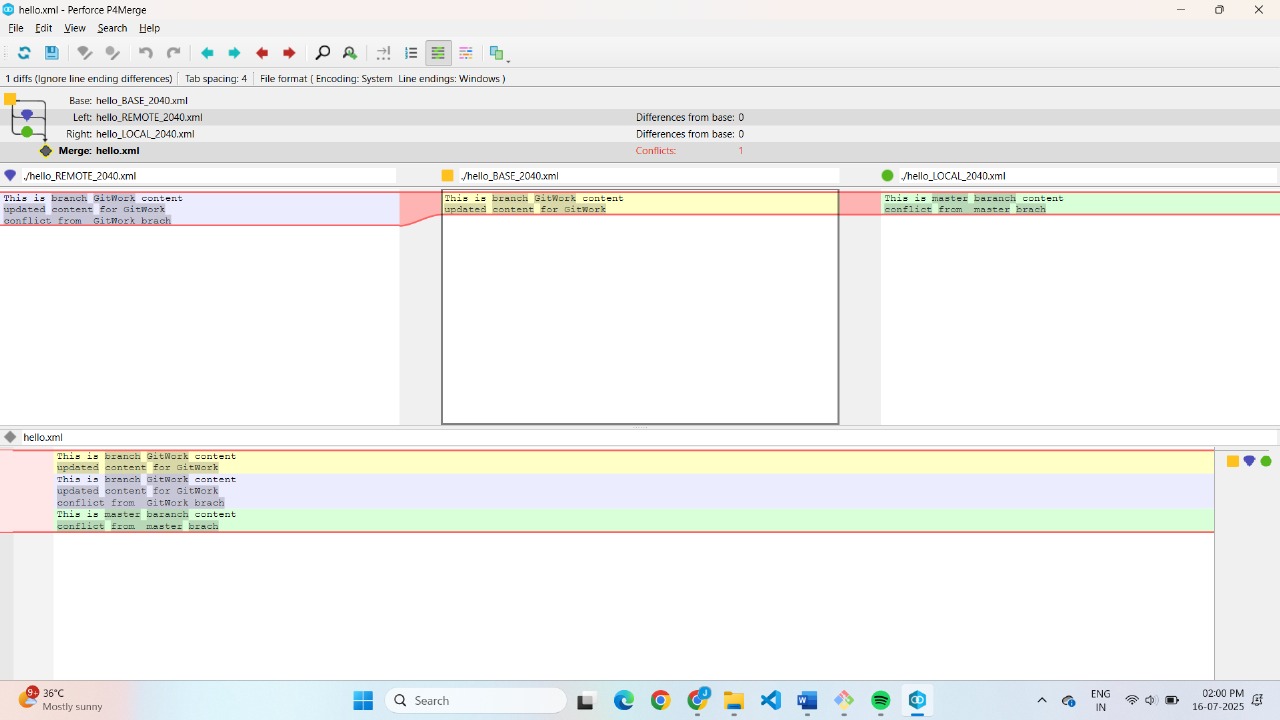
8. Observe the log



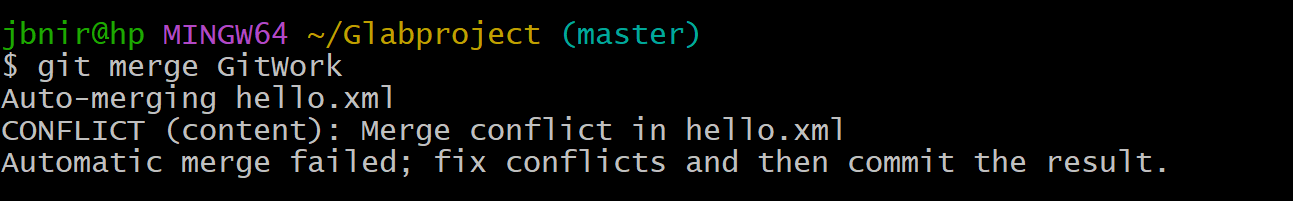
9. Check the differences with Git diff tool



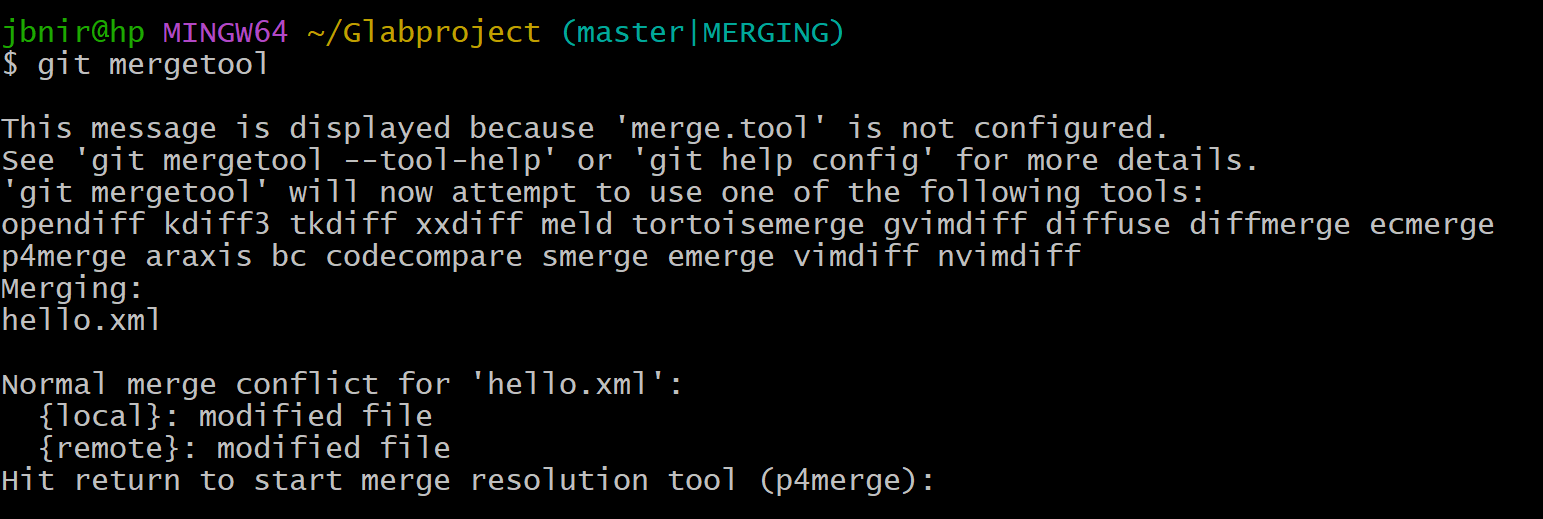
10. Use P4Merge tool to list out all the differences



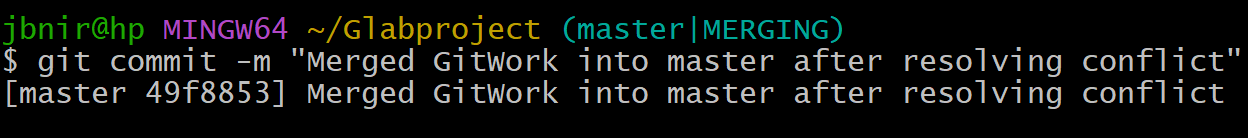
11. Merge the branch to the master



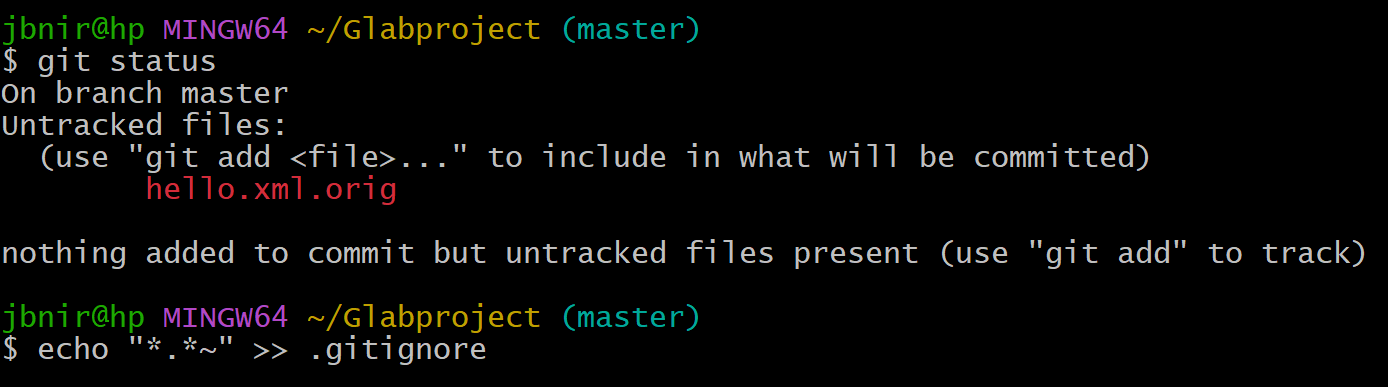
13. Use 3-way merge tool to resolve the conflict



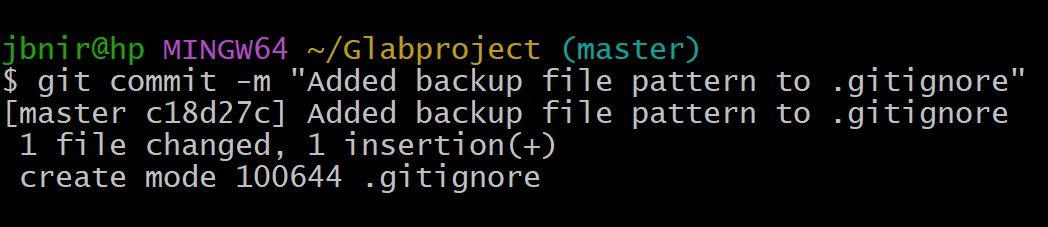
14. Commit the changes to the master, once done with conflict



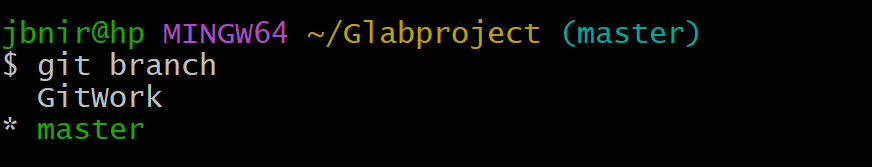
15. Observe git status and add backup file to the .gitignore



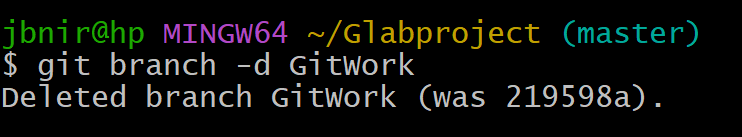
16. Commit the changes to the .gitignore



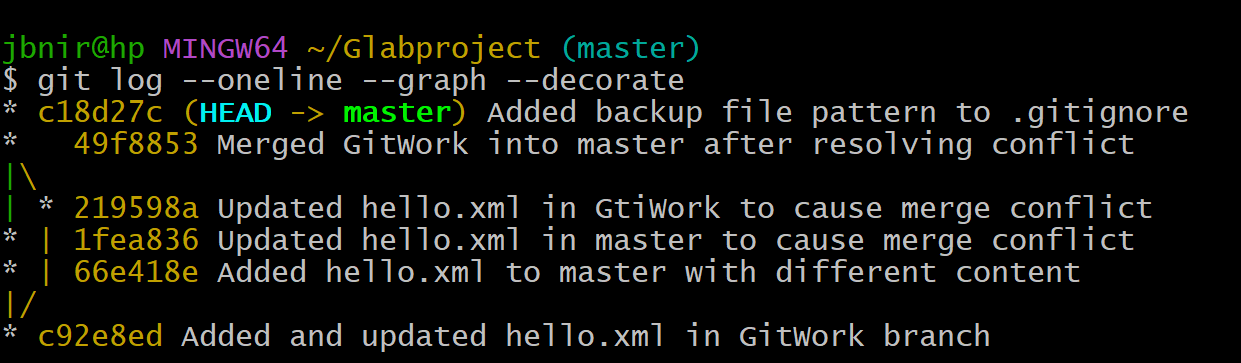
17. List out all the available branches



18. Delete the branch, which is merged to master



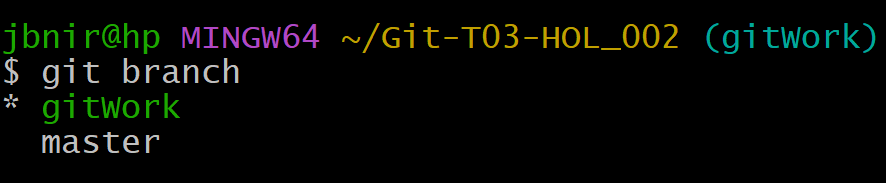
19. Observe the log again



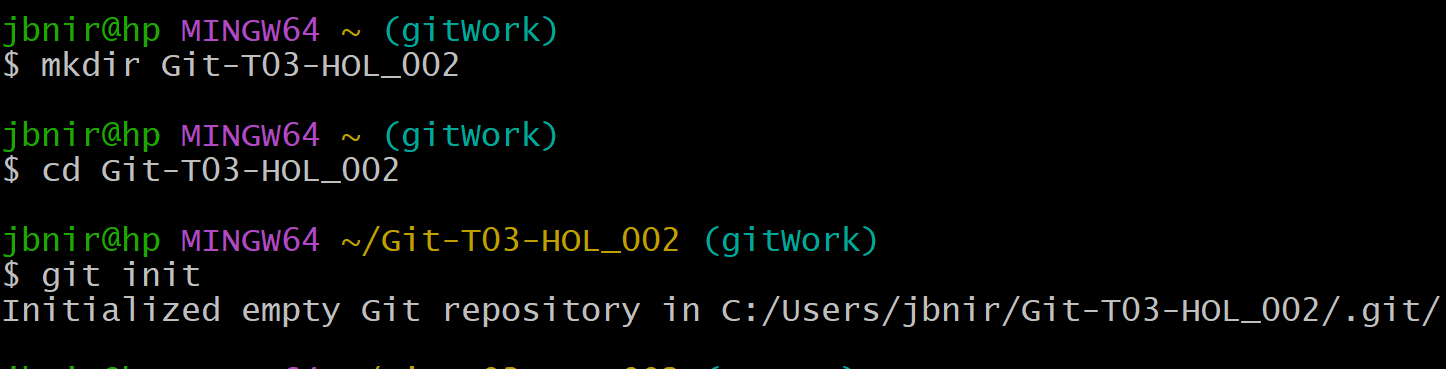
**5.GIT -HOL**

1.Create a new repository on GitHub

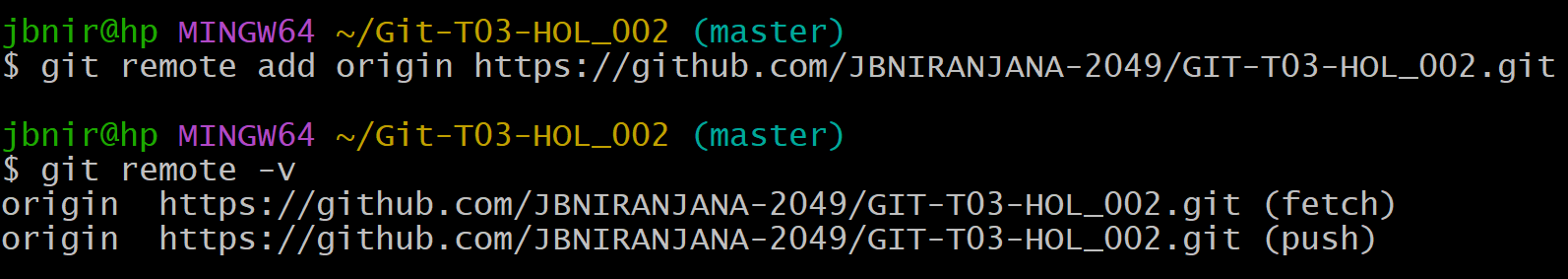
2. List out all the available branches.



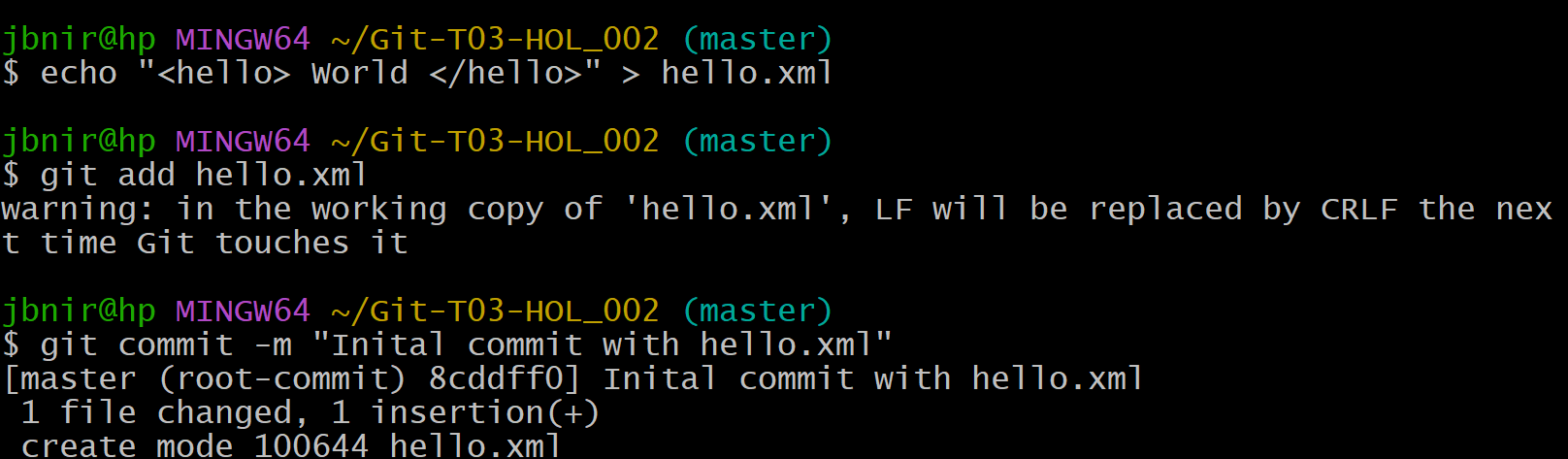
2: Set up Git locally in Git Bash



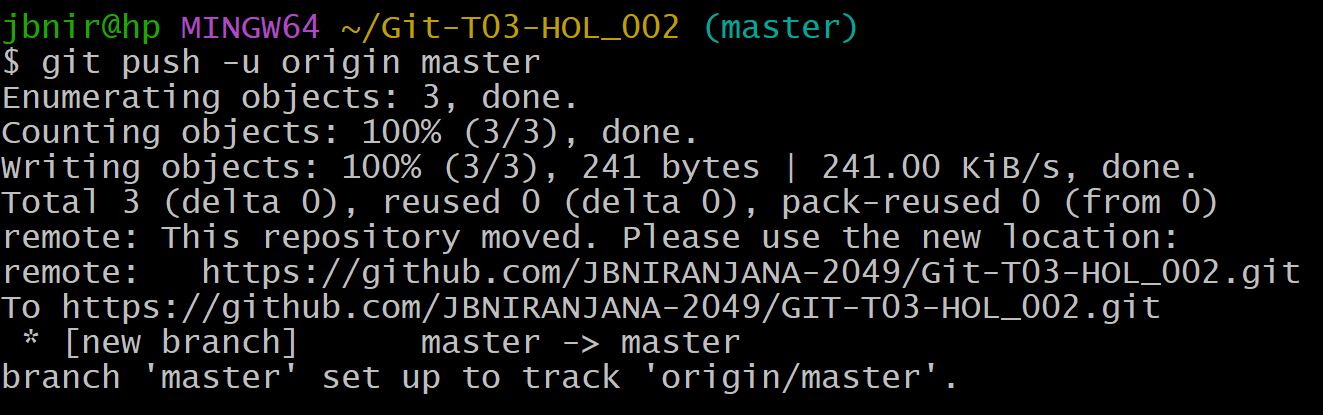
3: Link local folder to remote GitHub repo



4: Create files and commit changes



5: Push to GitHub



6.Check on GitHub

