**Cognizant Digital Nurture 4.0**

***WEEK - 7 Module 10 - GIT***

**HANDS ON – 1 (MANDATORY)**

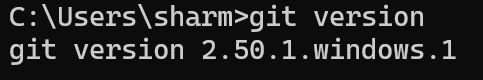
**Objectives:**

Familiar with Git commands like git init, git status, git add, git commit, git push, and git pull.

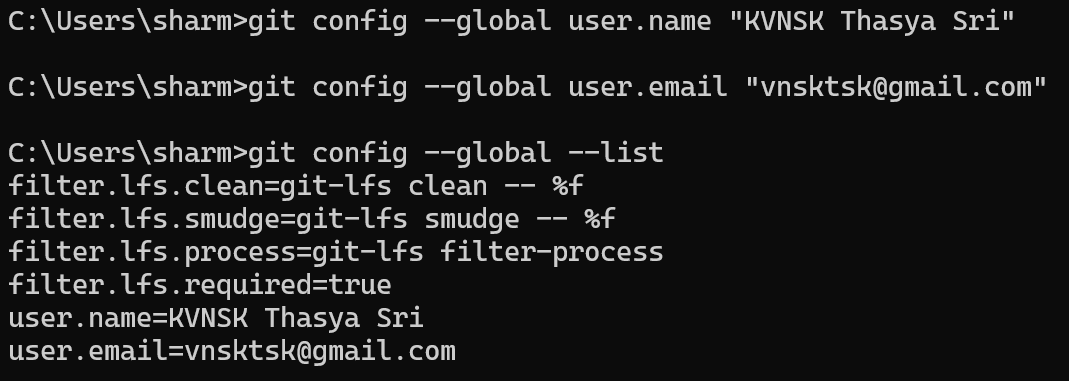
In this hands-on lab, you will learn how to

* Setup your machine with Git Configuration
* Integrate notepad++.exe to Git and make it a default editor
* Add a file to source code repository

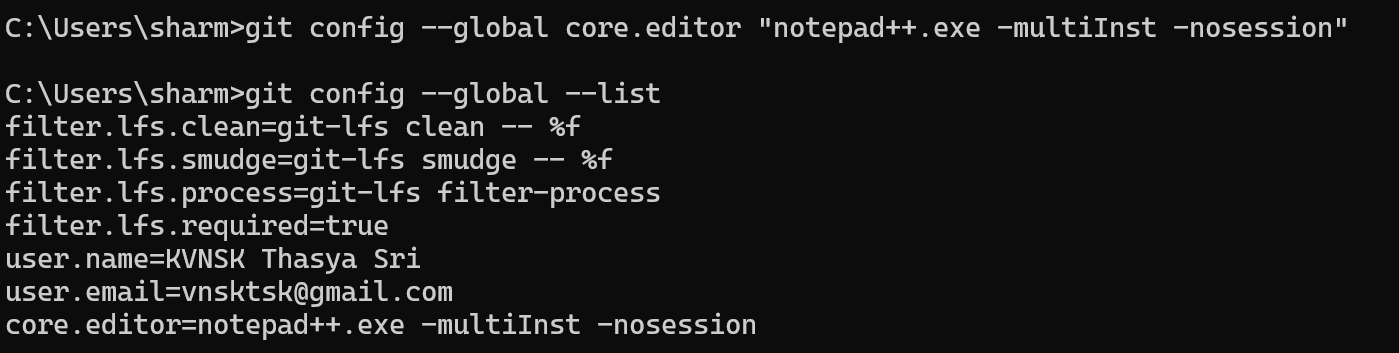
1. Verify Git Installation



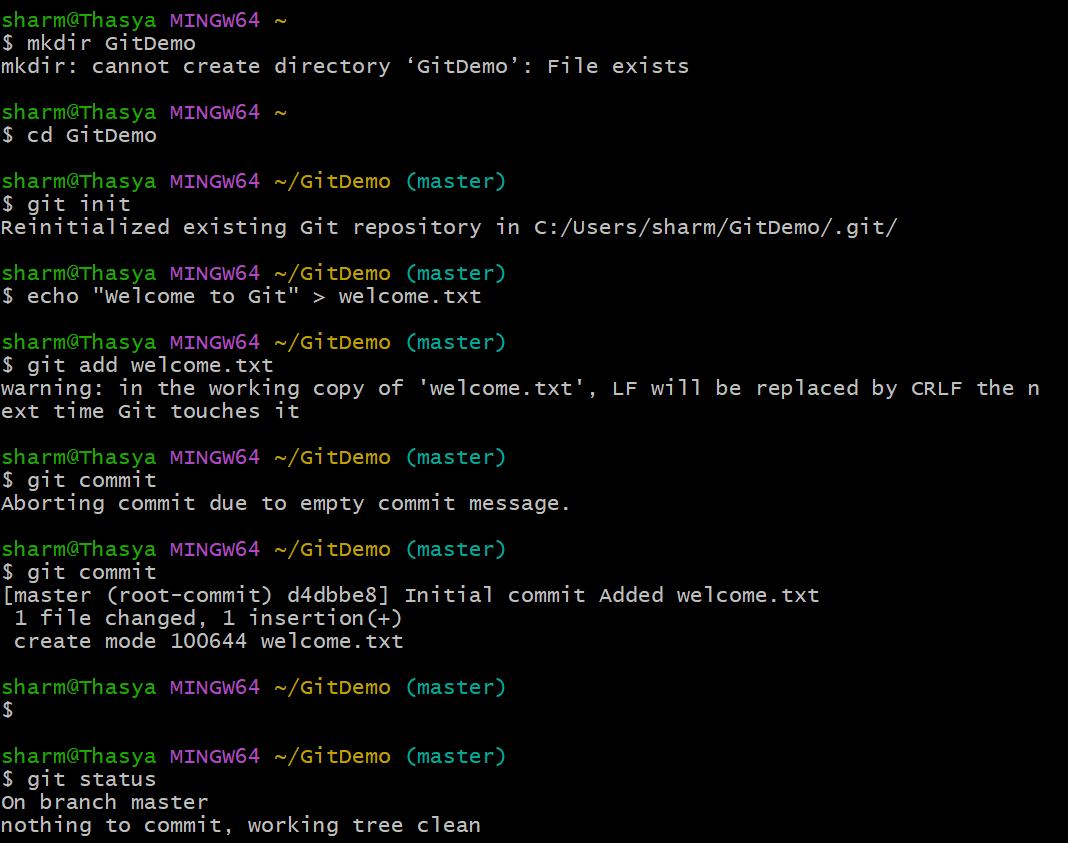
1. Configure Git with Your User Info



1. Set Notepad++ as Git’s Default Editor



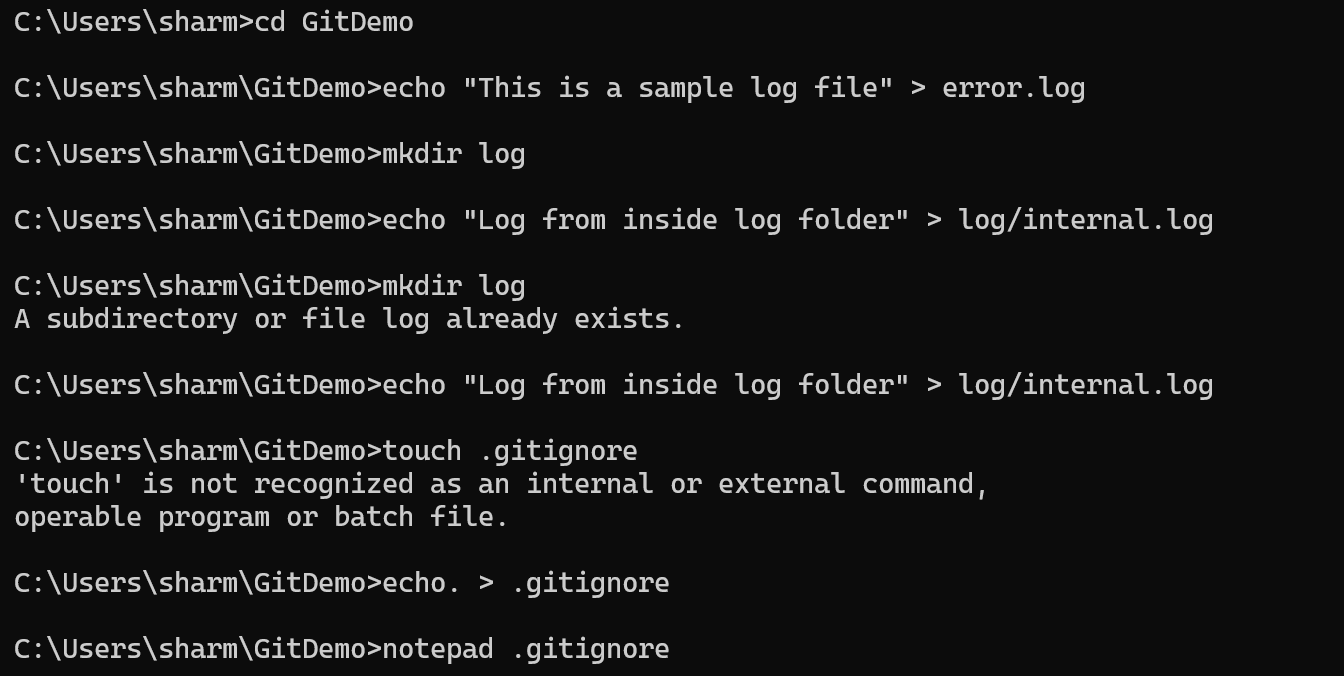
1. Create a New Git Project Locally



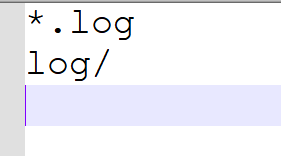
**HANDS-ON 2 (MANDATORY):**

**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.**

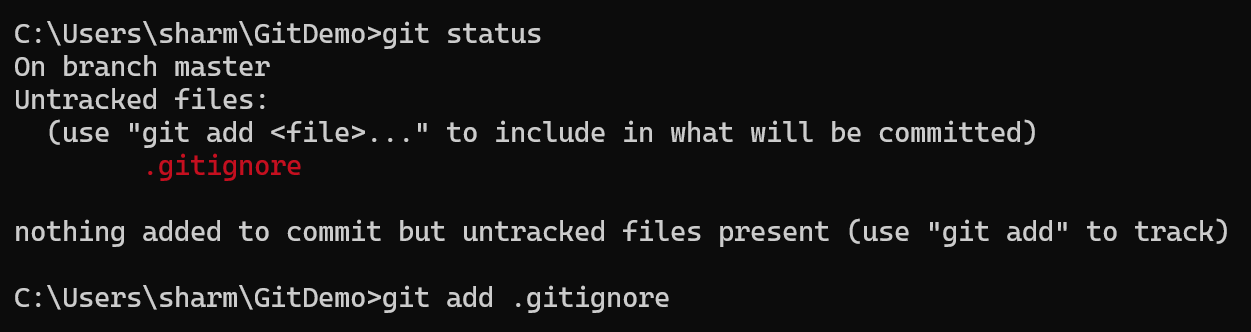
1. **Create a .log file**
2. **Create a log/ folder and a file inside it**



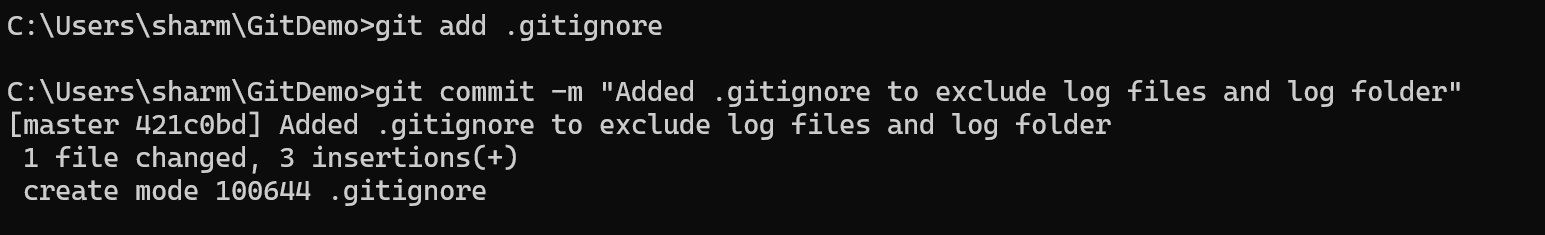
1. **Create or Update .gitignore**



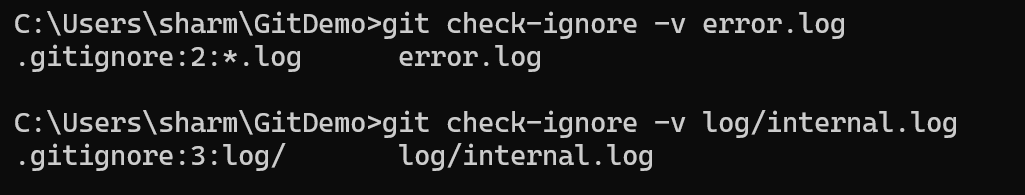
1. **Check Git Status**

****

1. **Commit the .gitignore File**

****

1. **Verify Git Repositories**



**HANDS-ON 3 (MANDATORY):**

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

**Branching:**

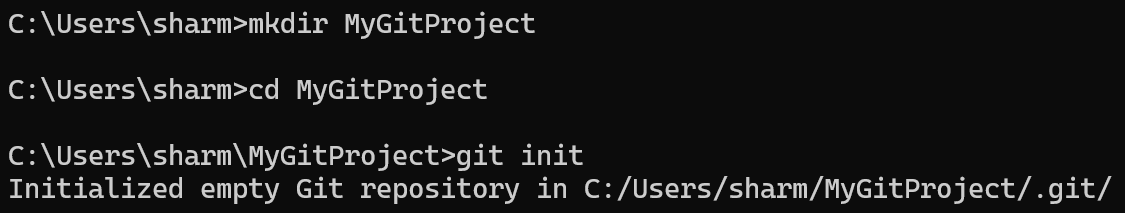
1. Create a new branch **“GitNewBranch”.**
2. List all the local and remote branches available in the current trunk. Observe the “\*” mark which denote the current pointing branch.
3. Switch to the newly created branch. Add some files to it with some contents.
4. Commit the changes to the branch.
5. Check the status with **“git status”** command.

**Merging:**

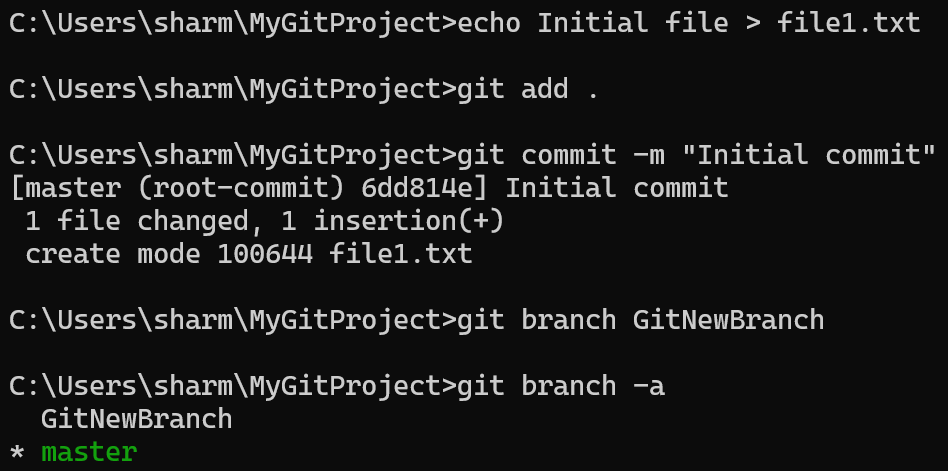
1. Switch to the master
2. List out all the differences between trunk and branch. These provide the differences in command line interface.
3. List out all the visual differences between master and branch using **P4Merge tool**.
4. Merge the source branch to the trunk.
5. Observe the logging after merging using **“git log –oneline –graph –decorate”**
6. Delete the branch after merging with the trunk and observe the git status.

Steps:

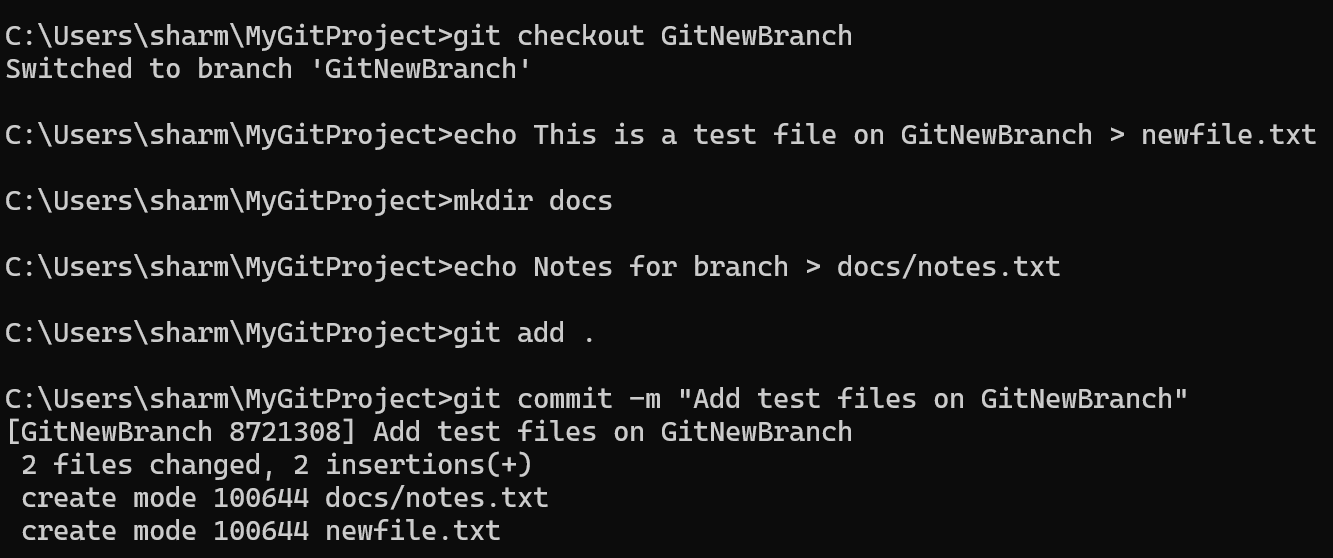
1. Create the new branch (only create):



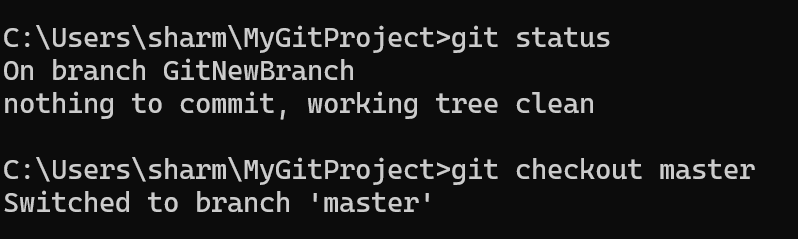
1. List all local and remote branches (see the \* for current):



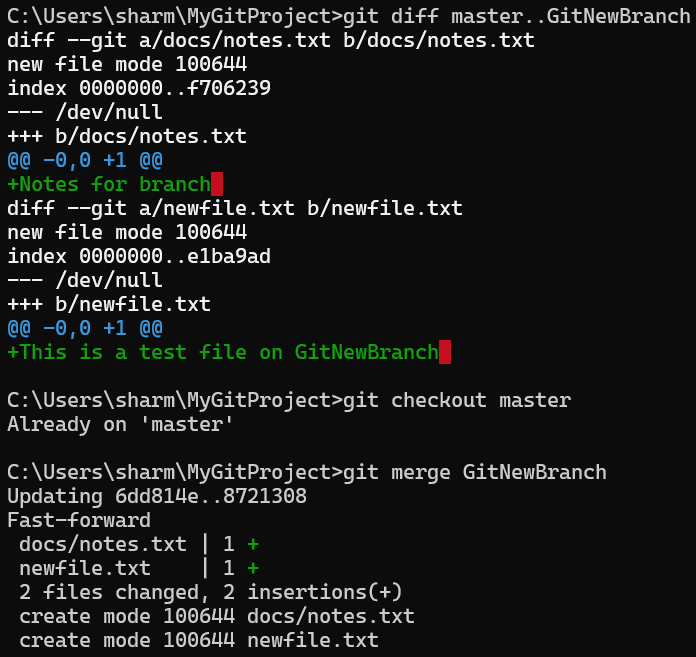
1. Switch to the new branch:
2. Add some files with content



1. Stage and commit the changes:



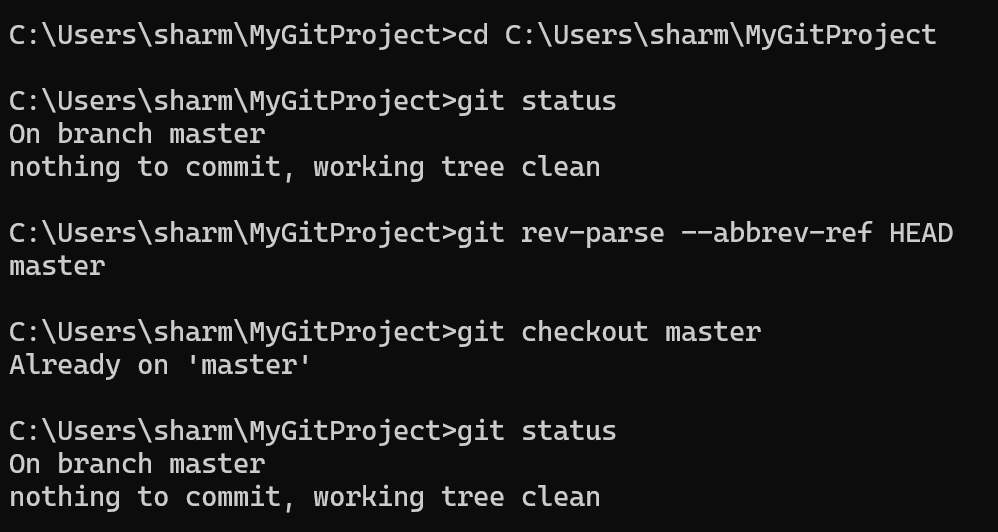
1. Check status:



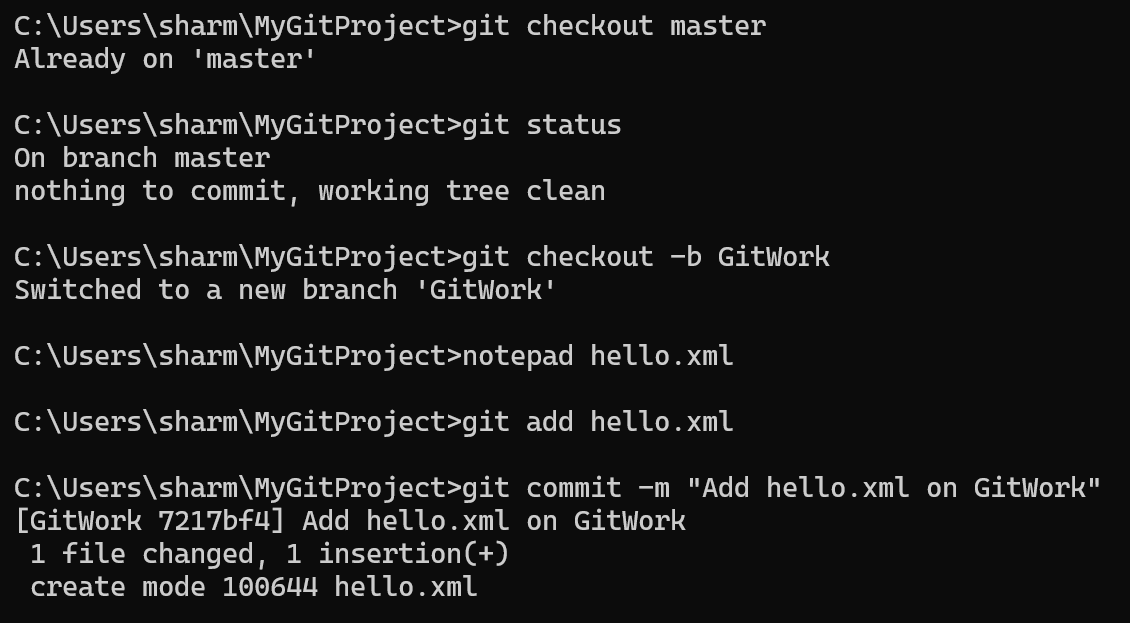
**HANDS-OFF 4 (MANDATORY):**

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

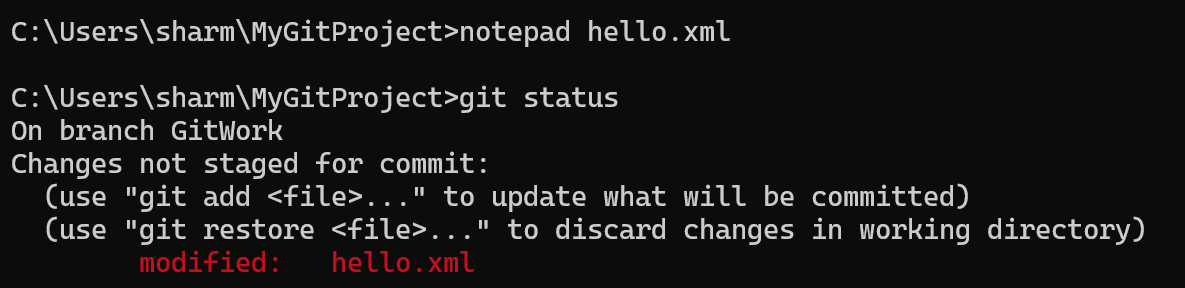
1. Verify if master is in clean state.



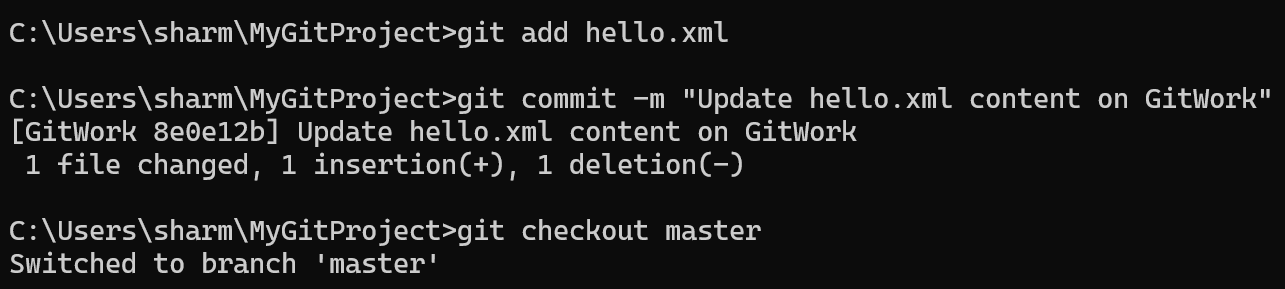
2. Create a branch “GitWork”. Add a file “hello.xml”.



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



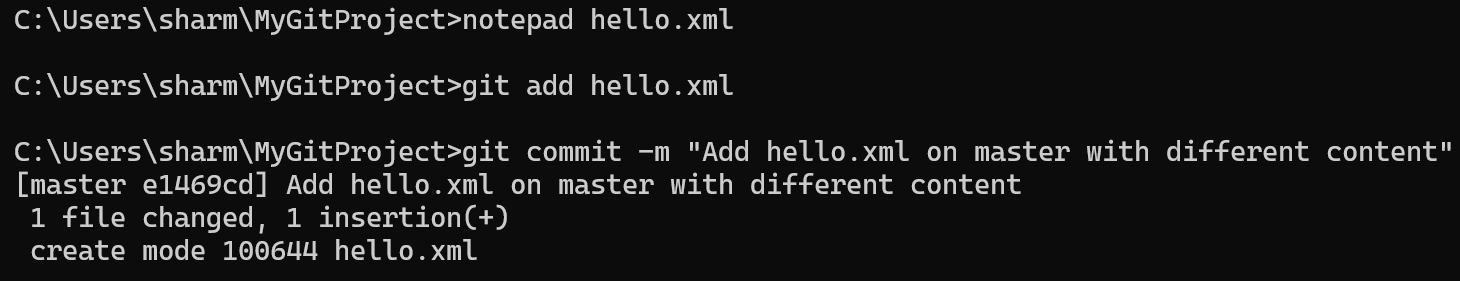
4. Commit the changes to reflect in the branch



5. Switch to master.

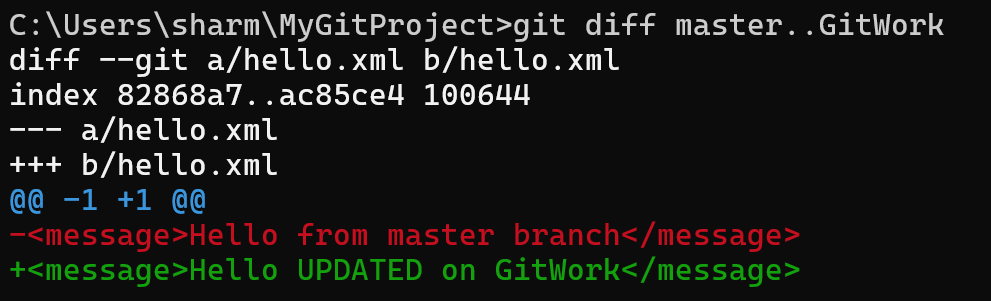
6. Add a file “hello.xml” to the master and add some different content than previous.

7. Commit the changes to the master

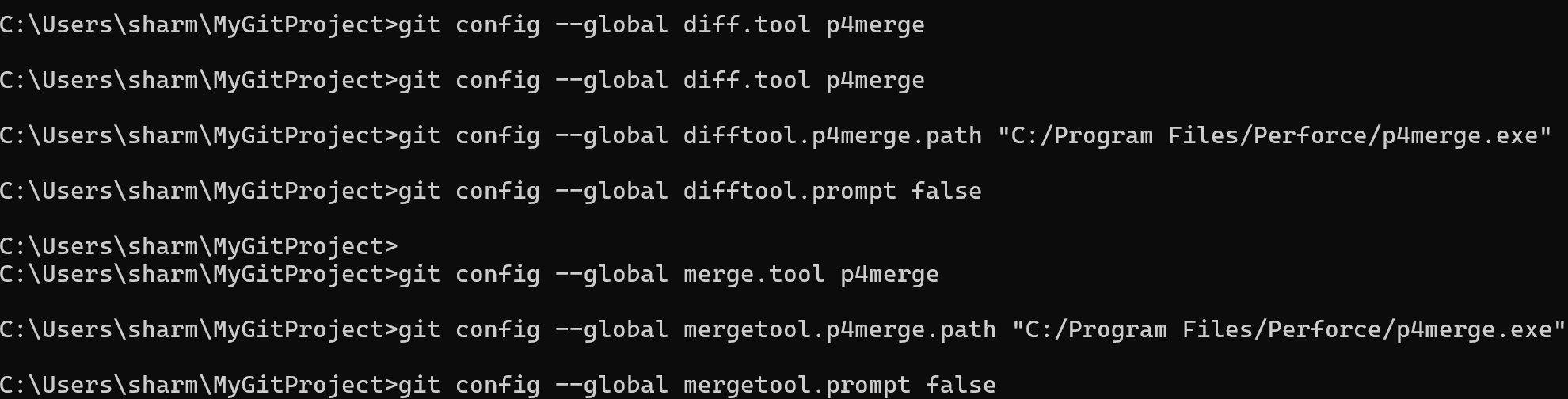


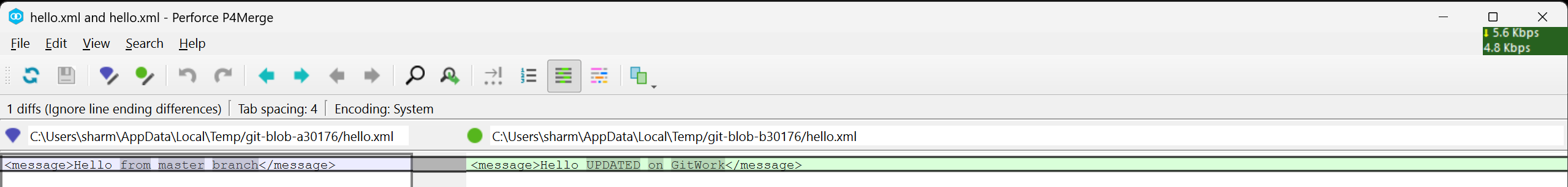
8. Observe the log by executing “git log –oneline –graph –decorate –all”

9. Check the differences with Git diff tool

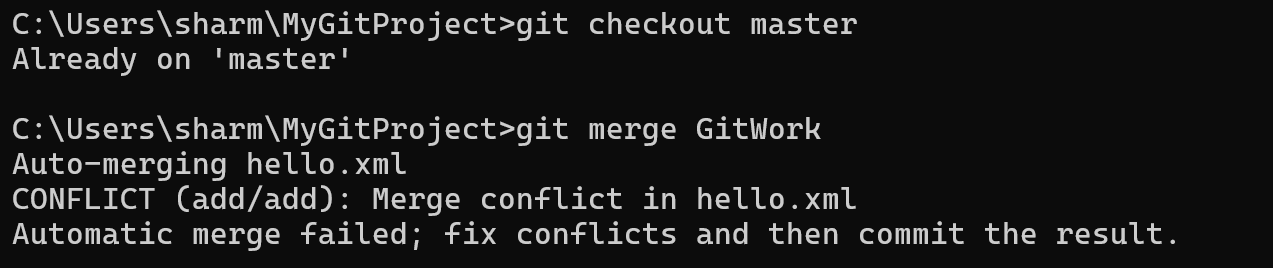


10. For better visualization, use P4Merge tool to list out all the differences between master and branch

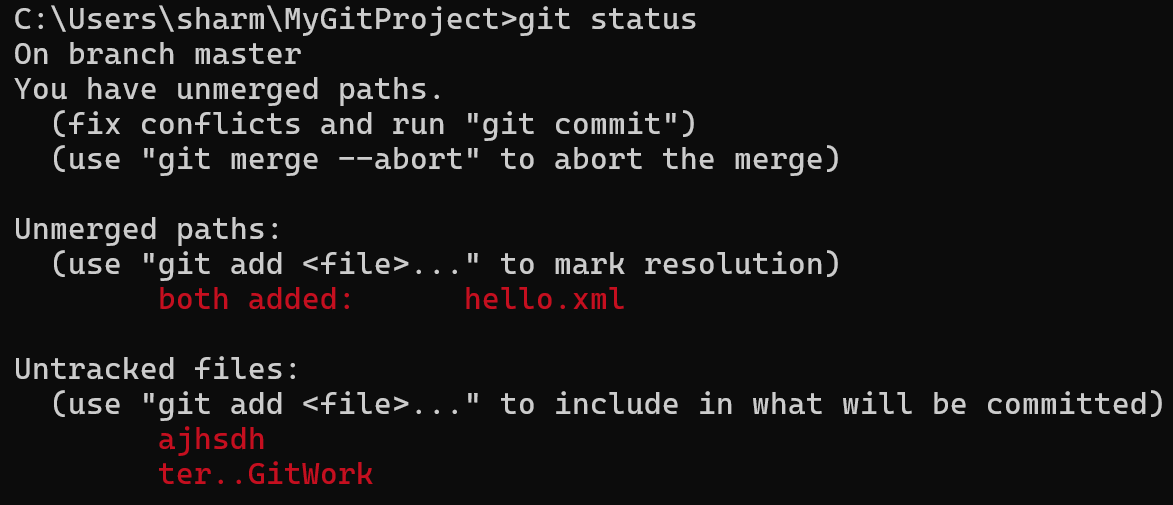


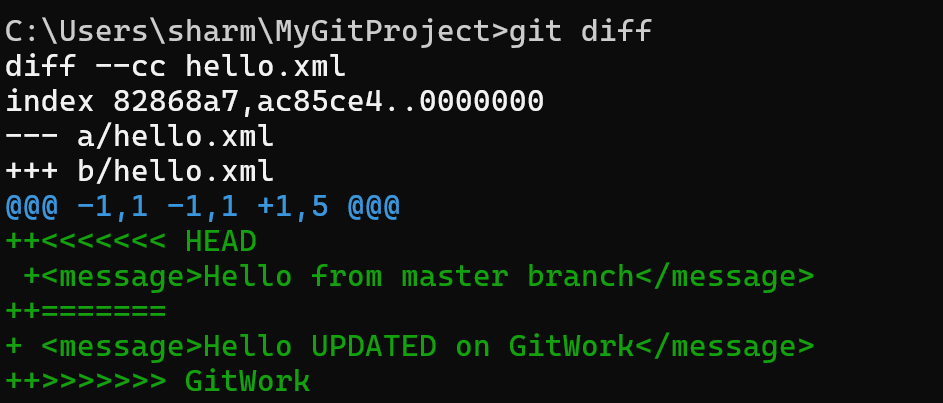


11. Merge the bran to the master



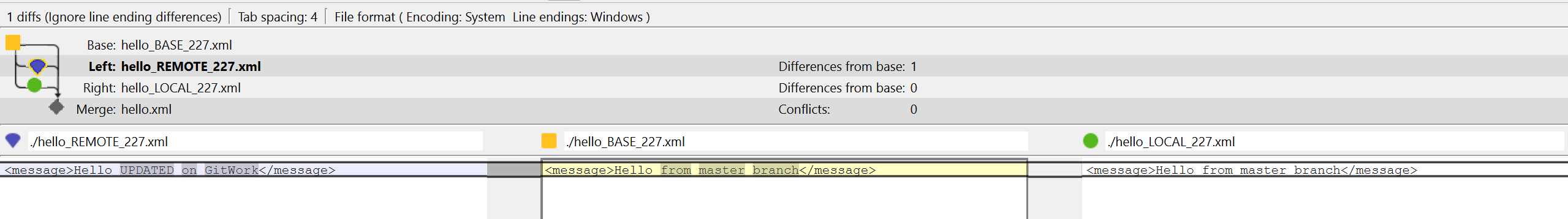
12. Observe the git mark up.



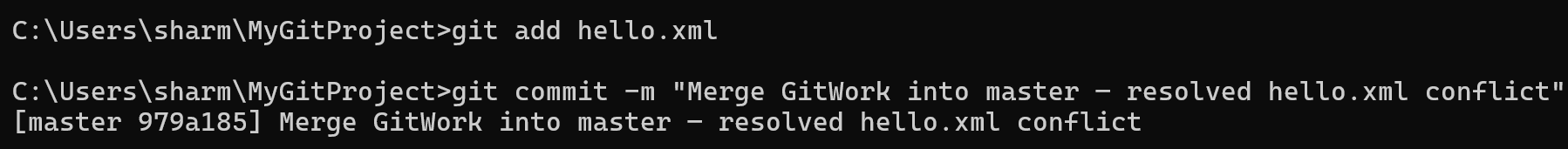


13. Use 3-way merge tool to resolve the conflict

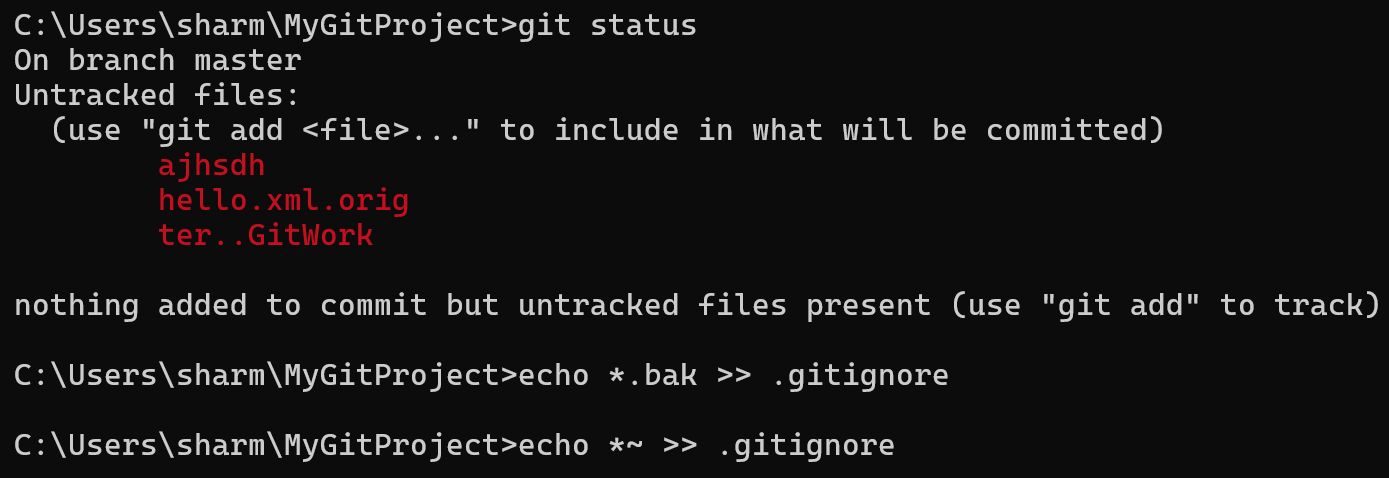




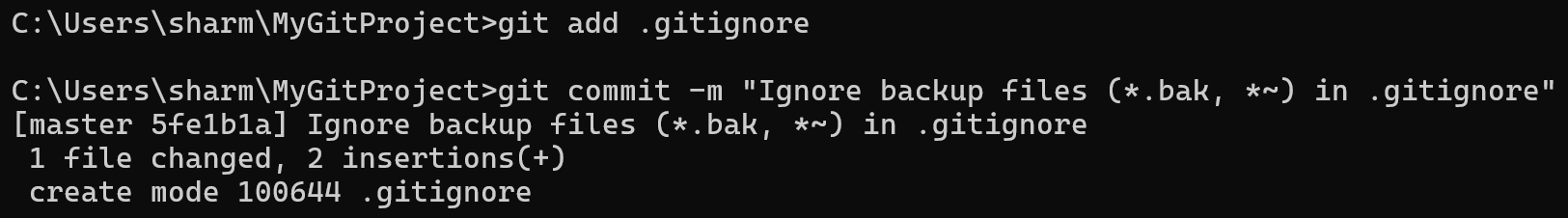
14. Commit the changes to the master, once done with conflict



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



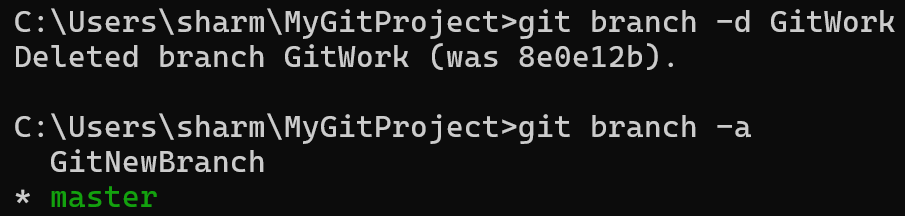
16. Commit the changes to the .gitignore



17. List out all the available branches



18. Delete the branch, which merge to master.



**HAND-ON 5:**

1. Verify if master is in clean state.
2. List out all the available branches.
3. Pull 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.

