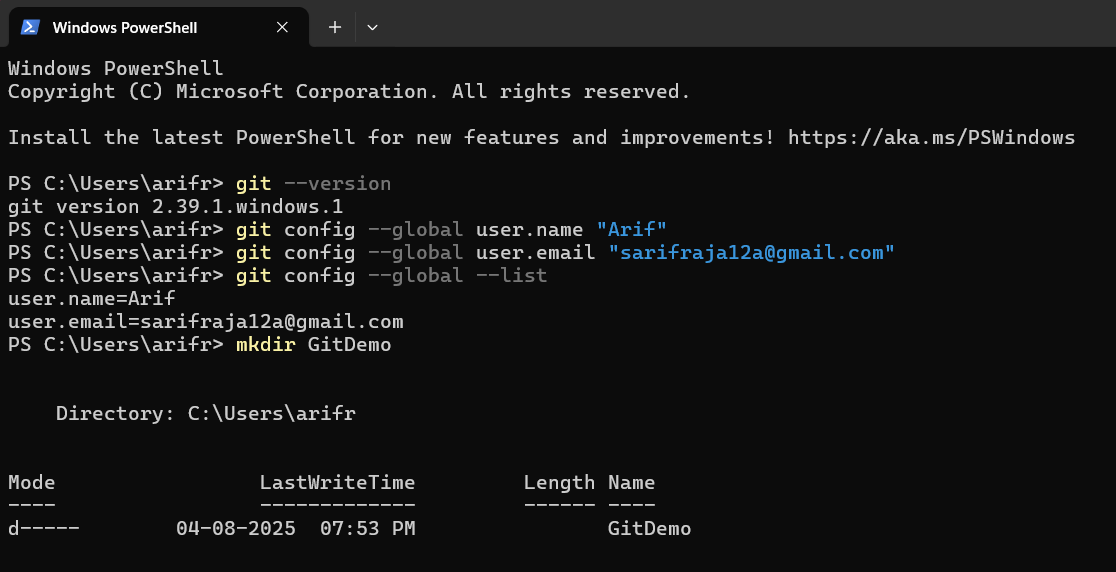
**Week 8**

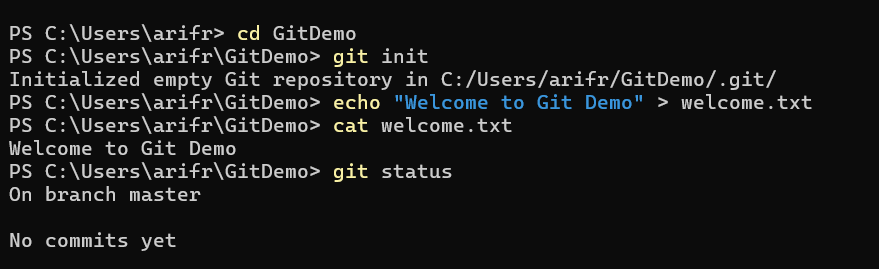
**Git HOL**

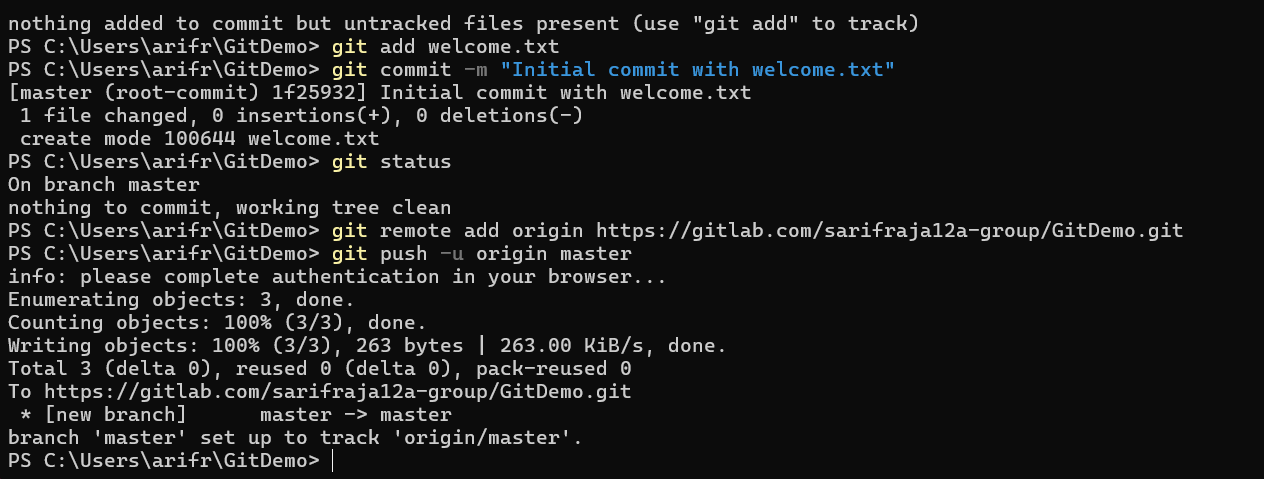
1. **Familiar with Git commands like git init, git status, git add, git commit, git push, and git pull. In this hands-on lab, you will learn how to setup your machine with Git Configuration. Integrate notepad++.exe to Git and make it a default editor. Add a file to source code repository.**

**Solution:**

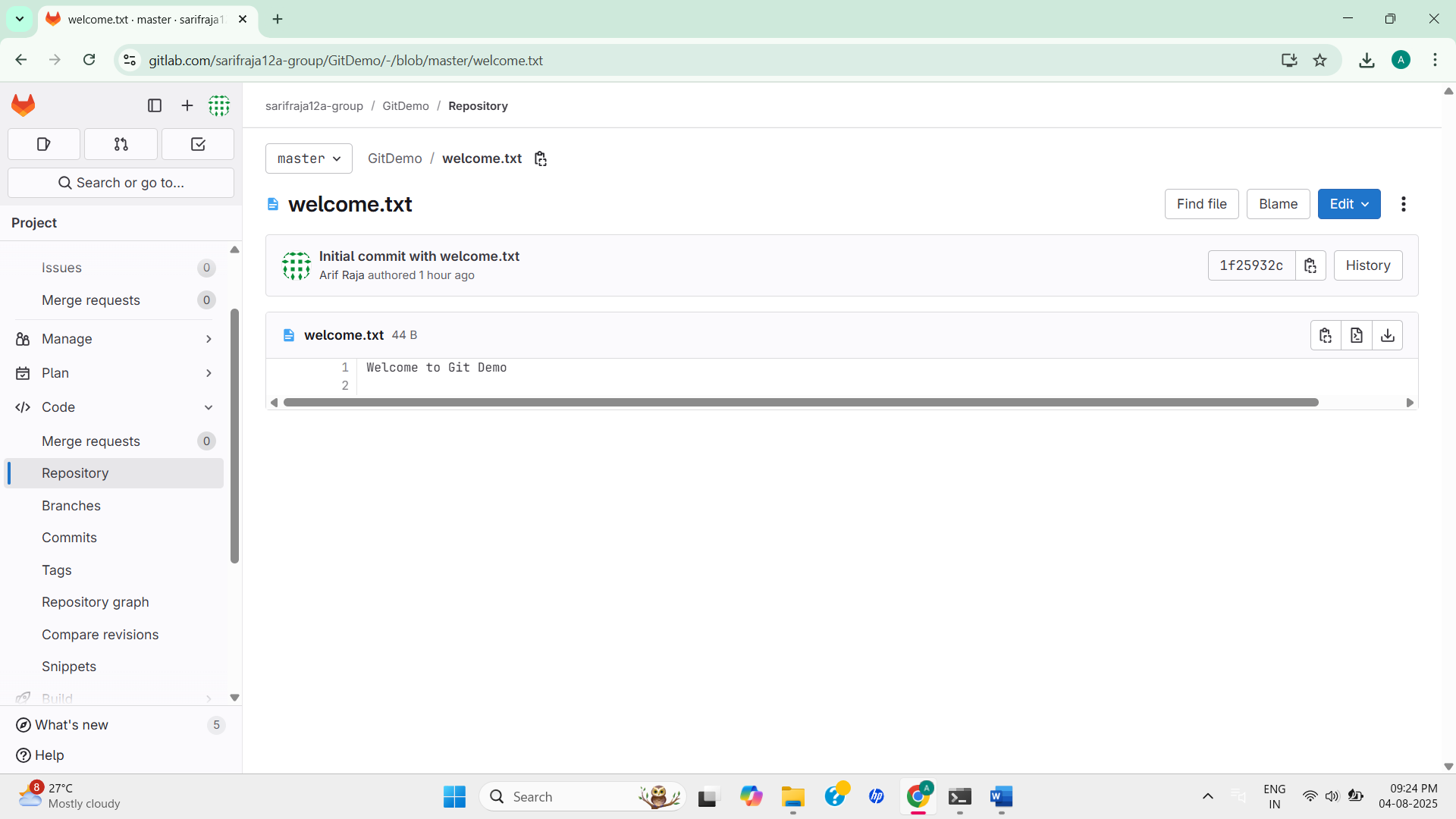
**Terminal:**

****

****

****

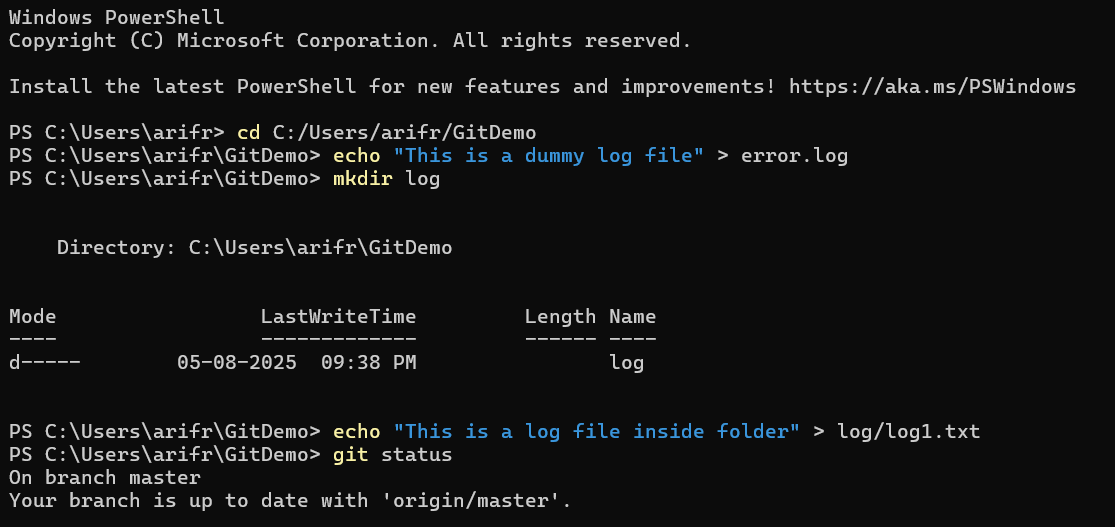
**Gitlab:**

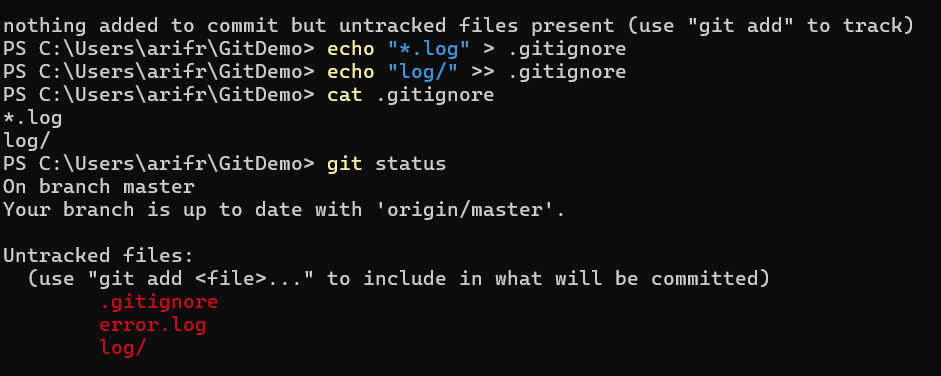


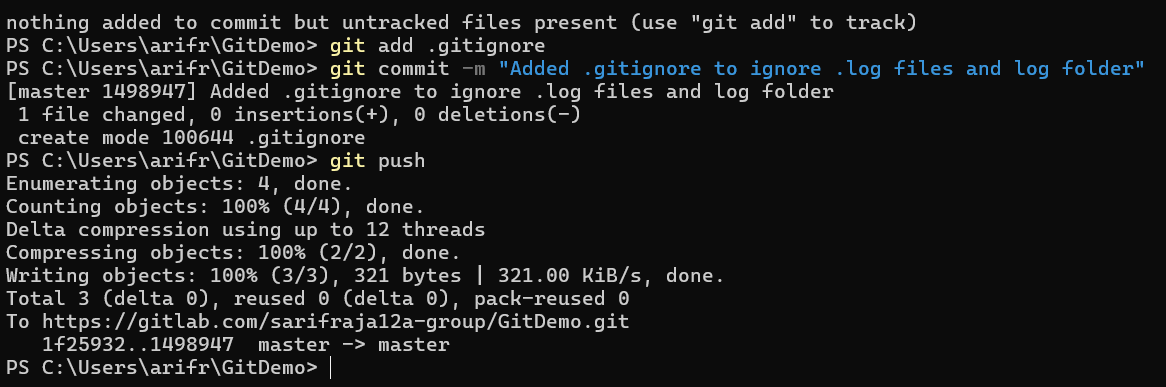
1. **Create a “.log” file and a log folder in the working directory of Git. Update the .gitignore file in such a way that on committing, these files (.log extensions and log folders) are ignored. Verify if the git status reflects the same about working directory, local repository and git repository.**

**Solution:**

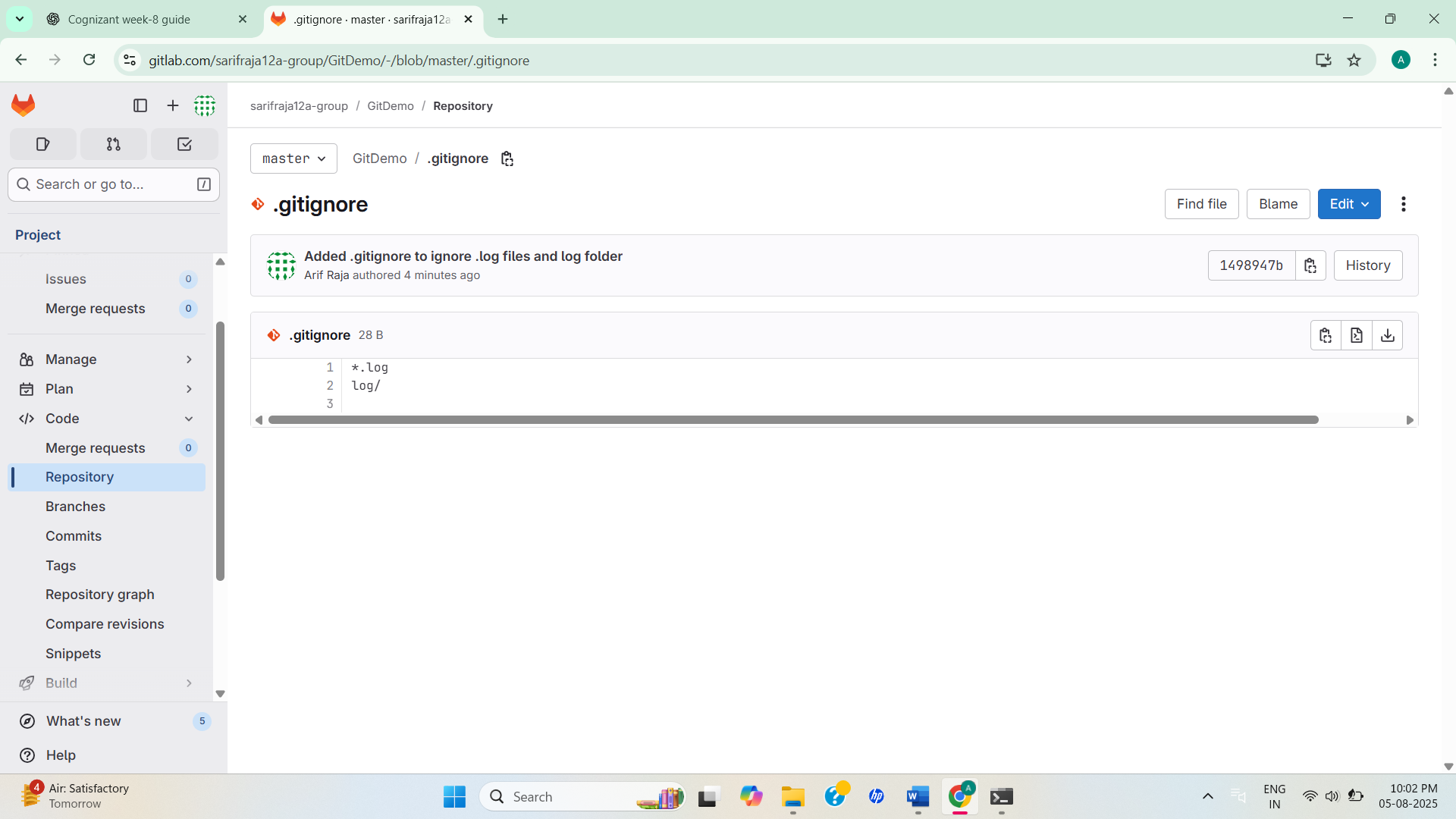
**Terminal:**

****

****

****

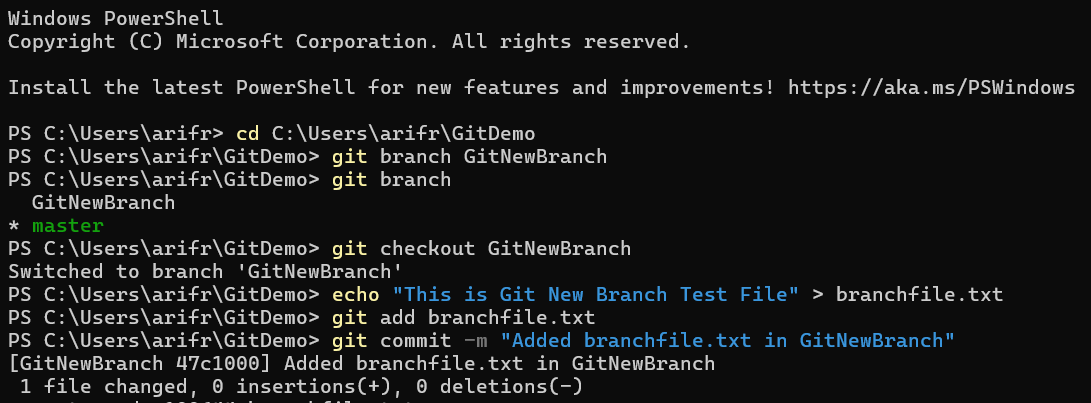
**Gitlab:**

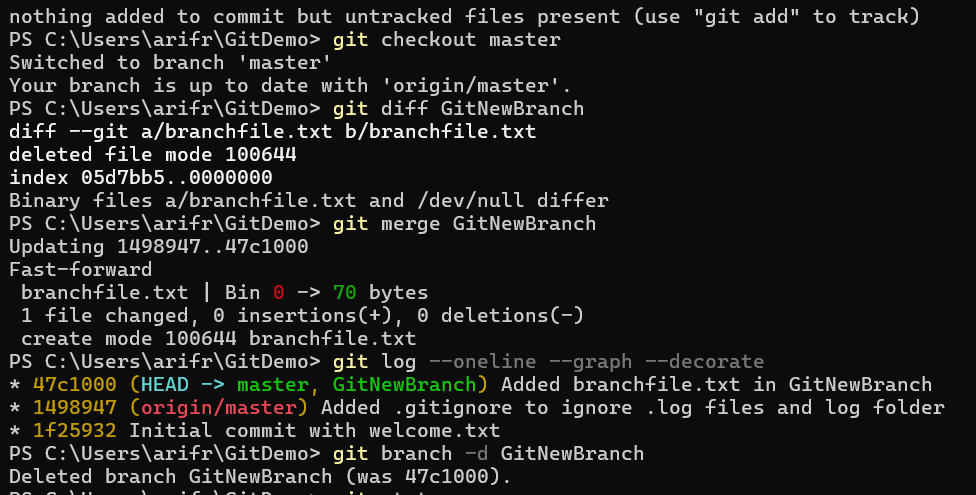


1. **Construct a branch, do some changes in the branch, and merge it with master (or trunk)**

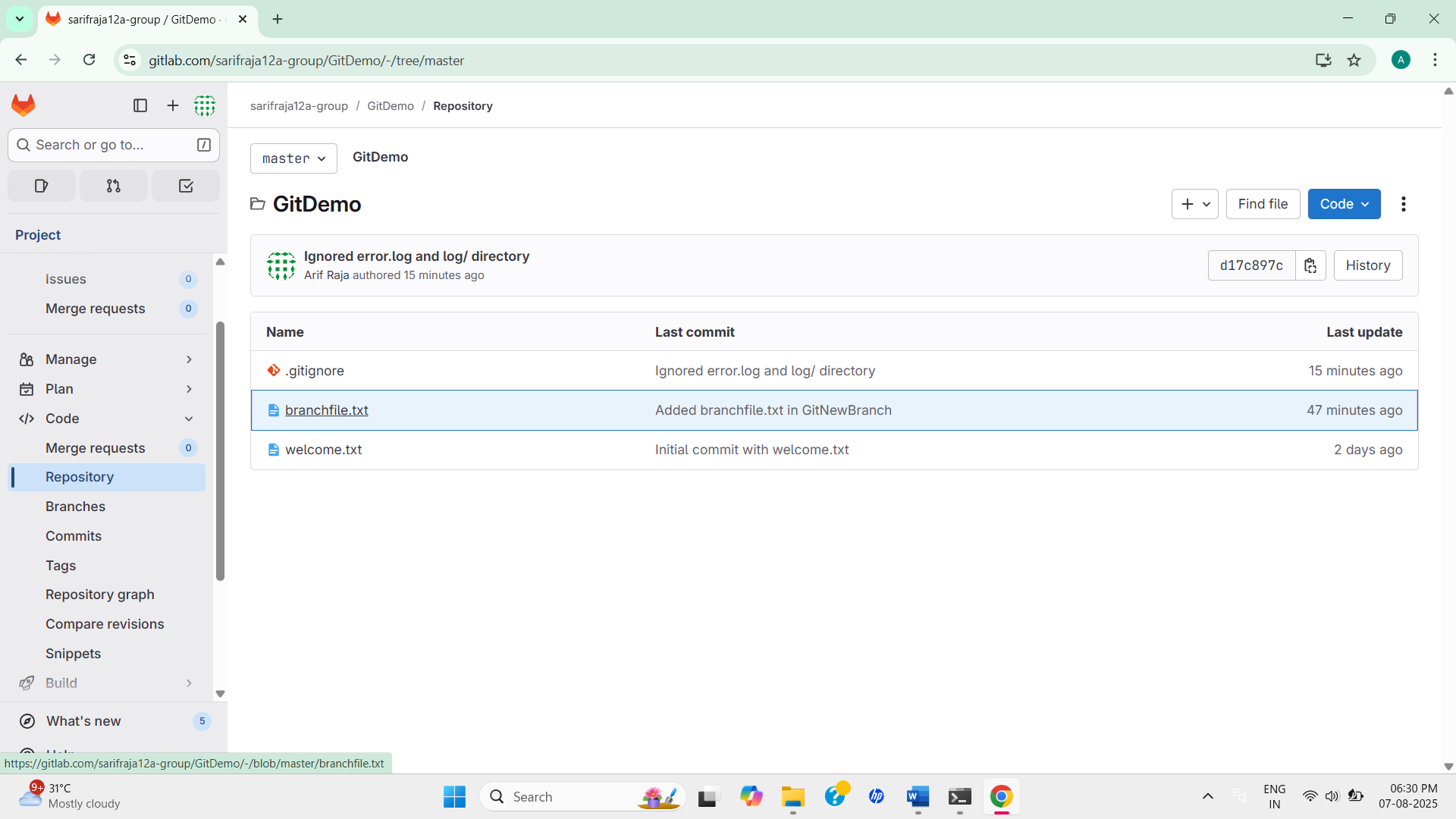
**Solution:**

**Terminal:**

****

****

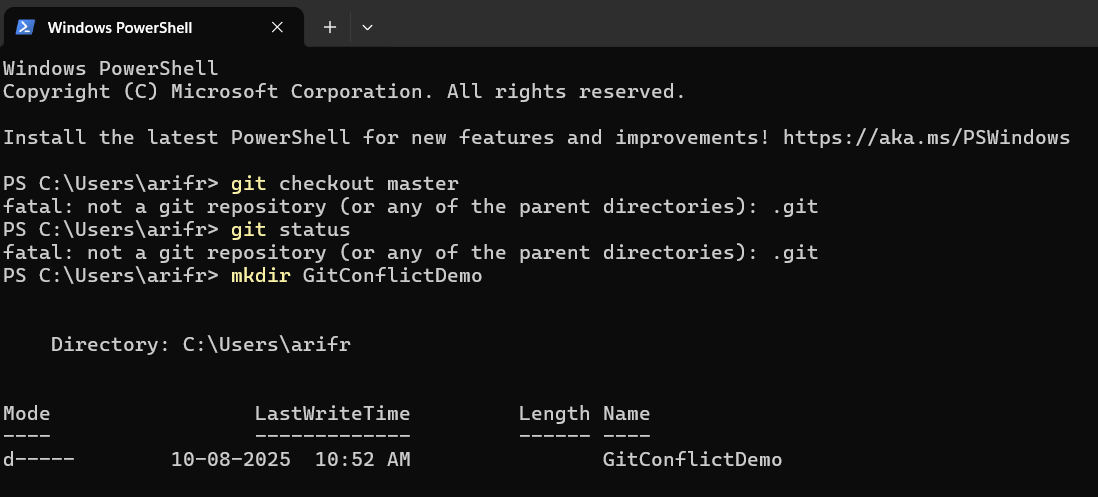
**Gitlab:**

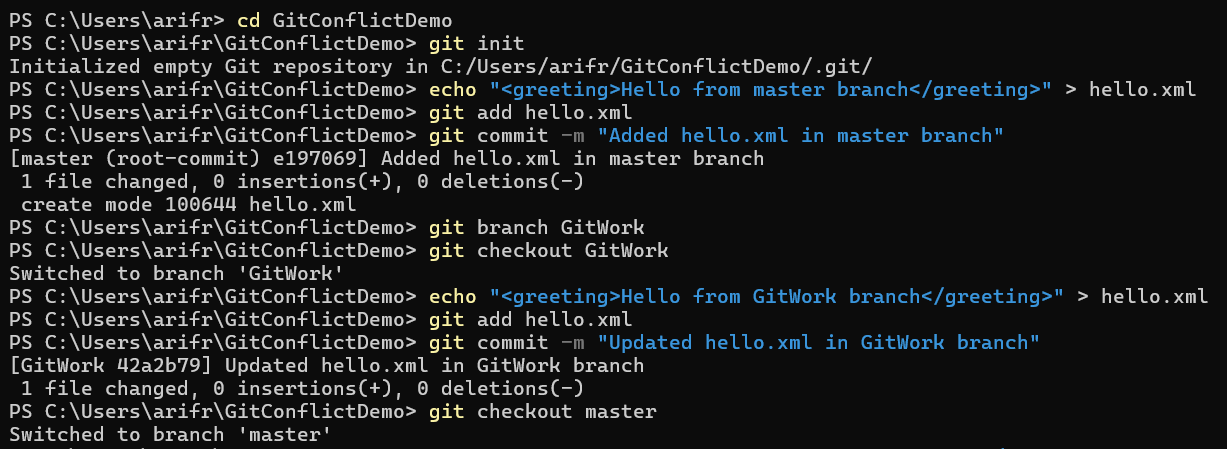


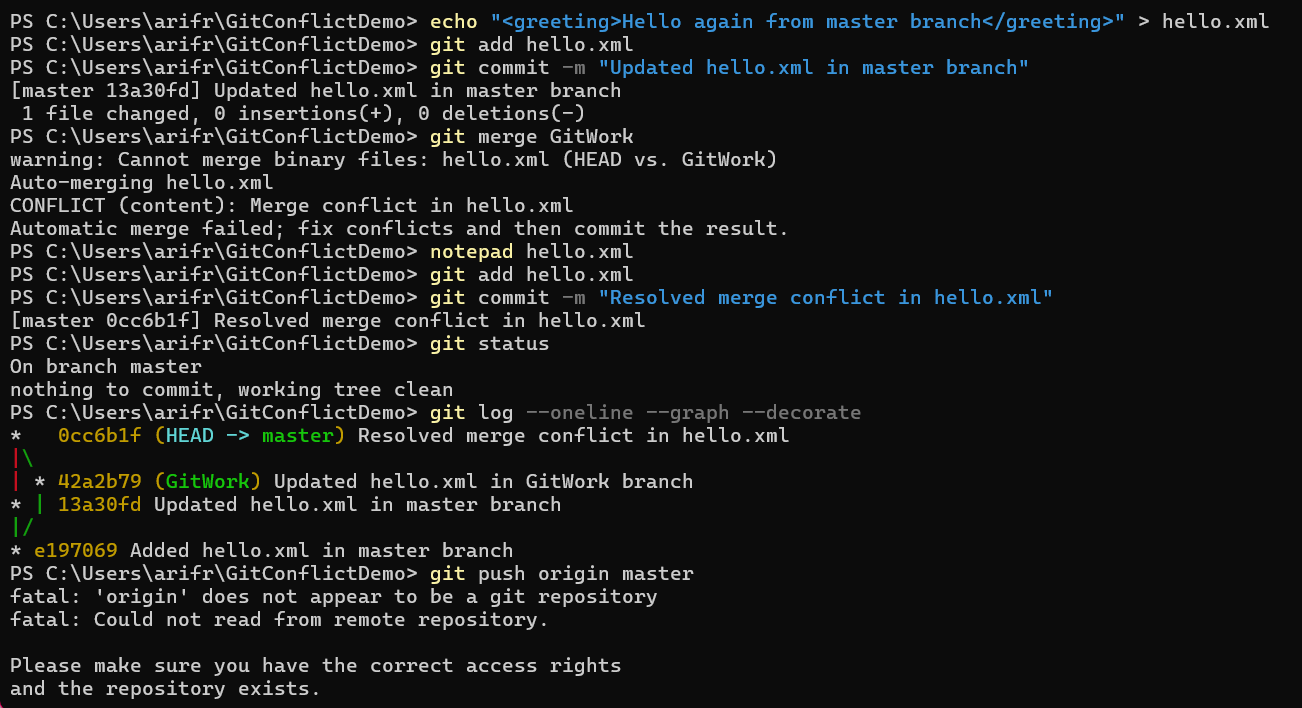
1. **Explain how to resolve the conflict during merge. Implement conflict resolution when multiple users are updating the trunk (or master) in such a way that it results into a conflict with the branch’s modification.**

**Solution:**

**Terminal:**

****

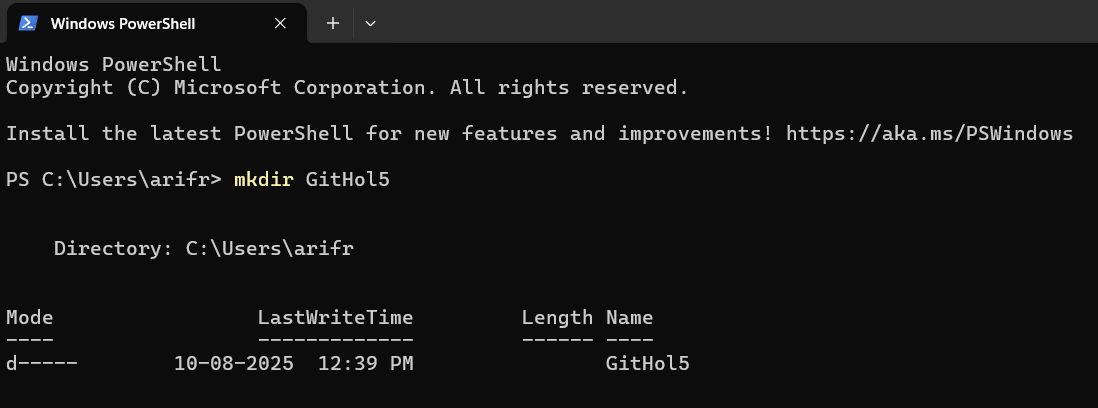
****

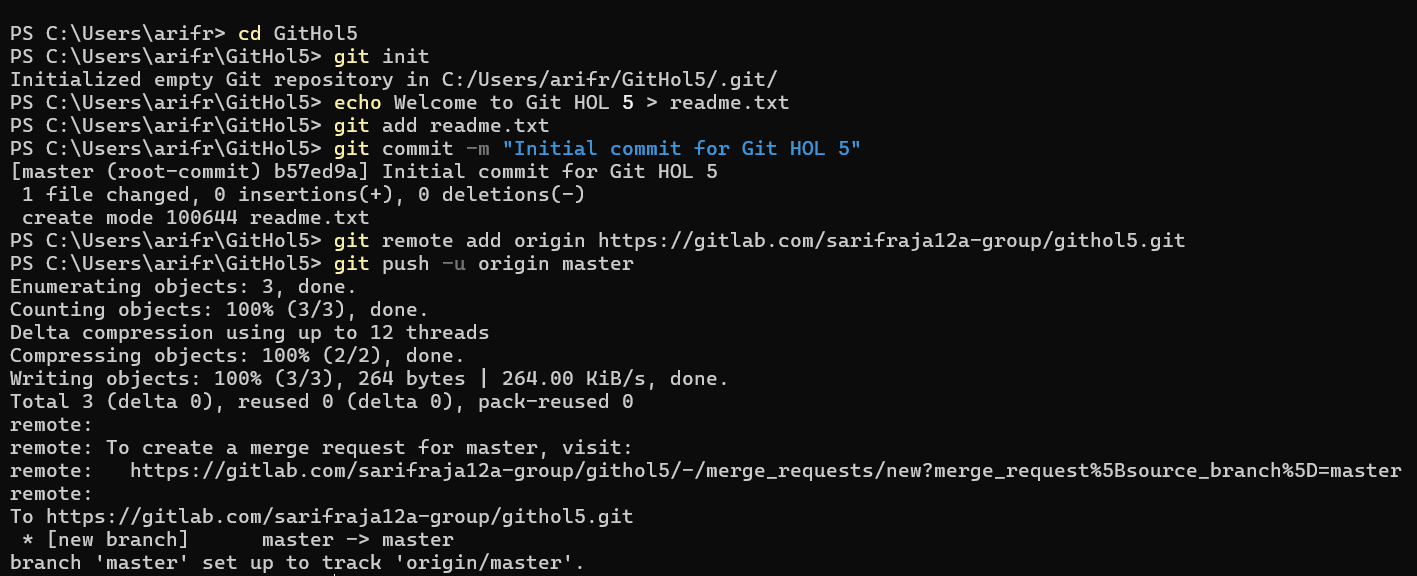
****

1. **Explain how to clean up and push back to remote Git. Execute steps involving clean up and push back to remote Git.**

**Solution:**

**Terminal:**

****

****

**Gitlab:**

