# Git Hands-On Lab — Exercise 2 (.gitignore)

## Objective

The objective of this exercise is to learn how to use a .gitignore file in Git to ignore unwanted files and folders, such as .log files and a log folder, so that they are not tracked or committed to the repository.

## Prerequisites

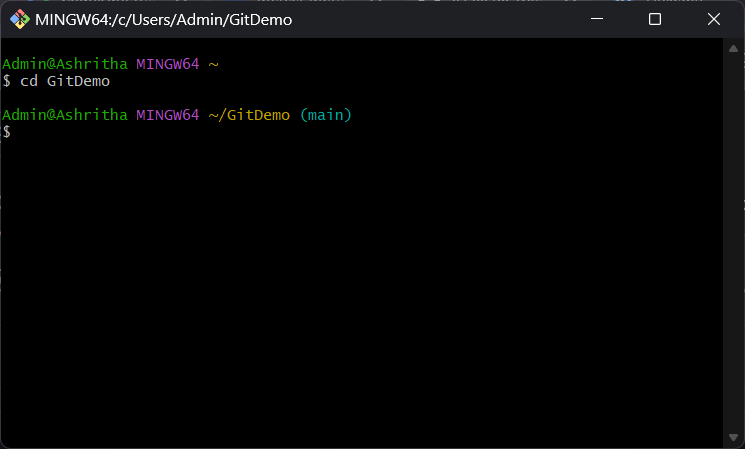
1. Git installed and configured.

2. Notepad++ integrated as default editor (optional).

3. Existing Git repository linked to a remote repository (GitHub).

### Step 1 — Navigate to Git Repository

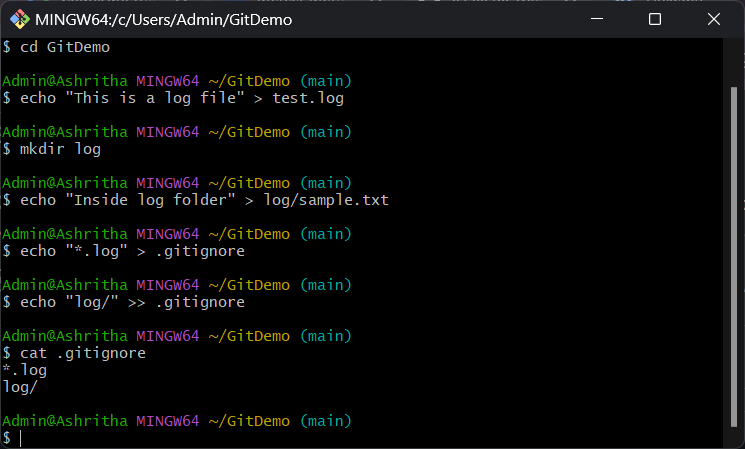
Command used: cd GitDemo



### Step 2 — Create .log File and log Folder

Commands used:  
echo "This is a log file" > test.log  
mkdir log  
echo "Inside log folder" > log/sample.txt

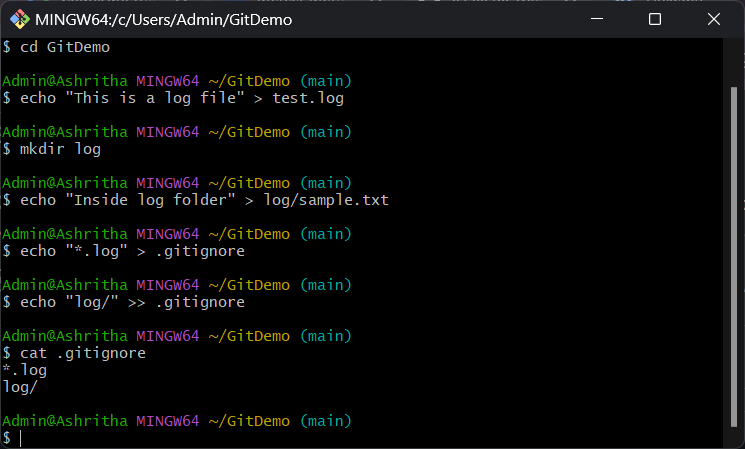
This step creates a sample log file and a folder named log containing another sample file.



### Step 3 — Create and Verify .gitignore File

Commands used:  
echo "\*.log" > .gitignore  
echo "log/" >> .gitignore  
cat .gitignore

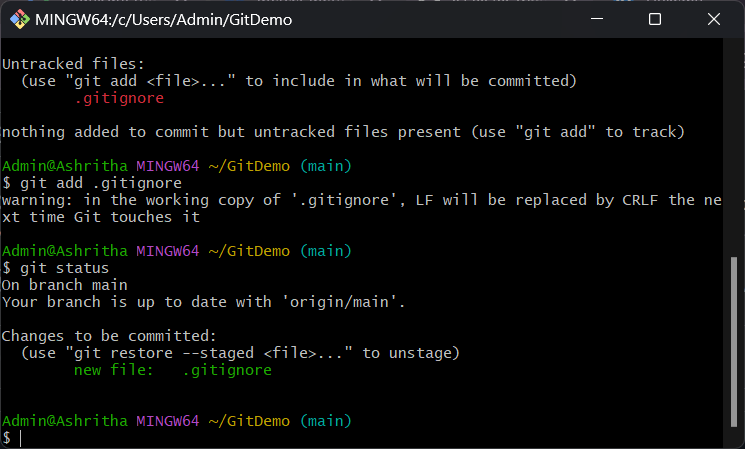
This creates a .gitignore file with patterns to ignore all .log files and the entire log folder.



### Step 4 — Stage .gitignore File

Command used: git add .gitignore  
git status

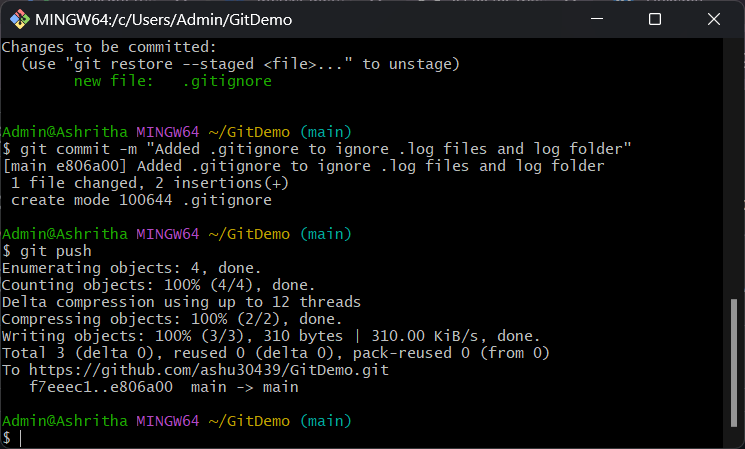
The git add command stages the .gitignore file for commit. git status confirms that the file is staged and will be included in the next commit.



### Step 5 — Commit and Push .gitignore File

Commands used:  
git commit -m "Added .gitignore to ignore .log files and log folder"  
git push

The git commit command records the change in the local repository, and git push sends the change to the remote repository on GitHub.



## Conclusion

By using the .gitignore file, we successfully configured Git to ignore specific unwanted files (.log) and folders (log/). This ensures these files are not tracked or pushed to the remote repository.

