# Git Hands-On Lab: Ignoring Unwanted Files with .gitignore

## Objectives

• Explain git ignore  
• Explain how to ignore unwanted files using git ignore

## Execution Steps

1. Verified prerequisites:  
 - Git environment setup completed.  
 - Notepad++ integrated as default Git editor.  
 - Git repository created locally and linked to a remote GitLab repository.

2. Navigated to the working directory of the Git repository:  
 cd ~/GitDemo

3. Created a .log file and a log folder:  
 echo "This is a log file" > debug.log  
 mkdir log

4. Created a .gitignore file and added rules to ignore all .log files and log folders:  
 echo "\*.log  
log/" > .gitignore

5. Verified .gitignore contents:  
 cat .gitignore  
 Output:  
 \*.log  
 log/

6. Checked Git status to ensure debug.log and log/ folder are ignored:  
 git status  
 Result:  
 - `.gitignore` appears as an untracked file.  
 - `debug.log` and `log/` are not listed, confirming they are ignored.

7. Added and committed the .gitignore file:  
 git add .gitignore  
 git commit -m "Add .gitignore to ignore .log files and log directory"

8. Verified final status:  
 git status  
 Output: nothing to commit, working tree clean.  
 This confirms the ignored files and folders are excluded from Git tracking.



