Objectives

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

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

- Implement git ignore command to ignore unwanted files and folders

Prerequisites

The following are the pre-requisites to complete this hands-on lab:

- Setting up Git environment
- Integrate notepad++ as a default editor
- A Git repository in the local system and a remote repository in GitLab

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.

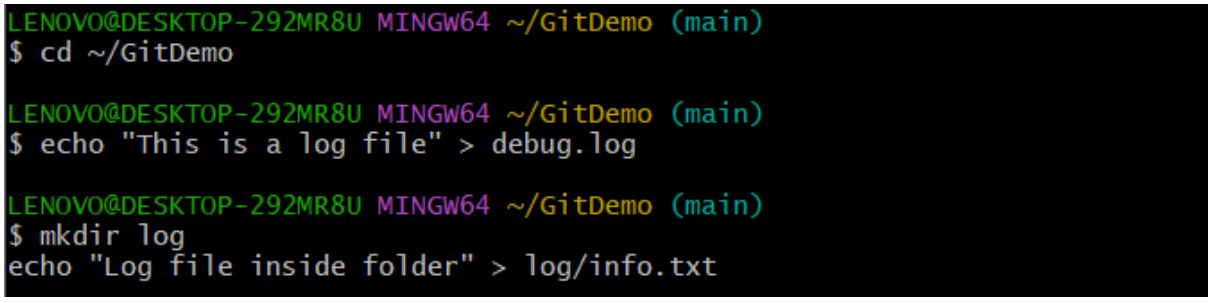
Verify if the git status reflects the same about working directory, local repository and git repository.

Step By Step Procedure:

1 — Go to your working Git repo
cd ~/GitDemo

2 — Create unwanted files & folders
Create a .log file
echo "This is a log file" > debug.log

Create a folder named log with a file inside
mkdir log
echo "Log file inside folder" > log/info.txt



3 — Create/Edit .gitignore file

The .gitignore file tells Git which patterns to skip.

Open .gitignore in your editor:

notepad .gitignore

Add these lines:

```
# Ignore all .log files
*.log

# Ignore the log folder
log/
```

Save & close Notepad.

4 — Check status before adding .gitignore

git status

- You should see .gitignore as **untracked**.
- The .log file and log/ folder will **not** appear because they are now ignored.

```
LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (main)
$ notepad .gitignore

LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

nothing added to commit but untracked files present (use "git add" to track)
```

5 — Stage & commit .gitignore

```
git add .gitignore
git commit -m "Add .gitignore to ignore .log files and log folder"
```

6 — Verify ignore is working

- Create another .log file and see if Git tracks it:

```
echo "Another log file" > error.log
git status
```

You should see **no mention** of error.log in git status.

```
LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (main)
$ git add .gitignore
git commit -m "Add .gitignore to ignore .log files and log folder"
[main 46bf5cd] Add .gitignore to ignore .log files and log folder
 1 file changed, 5 insertions(+)
 create mode 100644 .gitignore

LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (main)
$ echo "Another log file" > error.log
git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

7 — Push to remote

git push

```
LENOVO@DESKTOP-292MR8U MINGW64 ~/GitDemo (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 352 bytes | 352.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/pooja-skcet/GitDemo.git
   e2e88a9..46bf5cd  main -> main
```