## **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

Notes\*:

|  |
| --- |
| Please follow the below steps for creating a free account in GitHub.  Do not use cognizant credentials to login to GitHub. |

Estimated time to complete this lab: **20 minutes.**

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 :**

**Step 1: Create .log file and log folder**

In your **GitDemo** working directory, run:

echo "Sample log entry" > debug.log

mkdir log

echo "Error: Something went wrong" > log/error.txt

**Step 2: Create/Edit .gitignore**

If .gitignore does not exist, create it:

touch .gitignore

Open .gitignore in your editor and add:

# Ignore all .log files

\*.log

# Ignore log folder

log/

**Step 3: Verify Git status**

Run:

git status

You should **not** see debug.log or the log/ folder listed as untracked files.

If they still appear, it means they were already tracked before adding .gitignore. In that case, remove them from Git tracking (but keep them locally):

git rm --cached debug.log

git rm -r --cached log/

**Step 4: Commit .gitignore changes**

git add .gitignore

git commit -m "Updated .gitignore to ignore .log files and log folder"

git status