## **Objectives**

* **Explain git ignore**

**Purpose:** Tells Git which files or folders to ignore (not track in commits).

Commonly used for:

Temporary files (.log, .tmp)

Build output (/dist, /bin)

OS-specific files (Thumbs.db, .DS\_Store)

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

The .gitignore file tells Git which files and folders to skip when tracking changes.  
You list patterns inside it, like \*.log to ignore all log files or logs/ to ignore a folder.  
It helps keep temporary, build, or sensitive files out of your repository.  
.gitignore only affects new or untracked files, not files already committed.  
Commit the .gitignore file so everyone on the project uses the same ignore rules.

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.





