**2. Git-HOL**

## **Objectives**

* **Explain git ignore**

The .gitignore file is a **special configuration file** in a Git repository that tells Git which files or folders it should **ignore** and not track.

* This is useful for **temporary**, **sensitive**, or **environment-specific** files that should not be pushed to the remote repository.
* Examples: log files, build output files, temporary files, IDE configuration files, OS-generated files.

Git will:

* Skip these files during git add (unless you force add with -f).
* Not show them in git status as untracked.
* Avoid committing them accidentally.
* **Explain how to ignore unwanted files using git ignore**
* **Create or open** a .gitignore file in the root directory of your Git repository.
* **Add patterns** for files or folders you want Git to ignore.
* To ignore all .log files:

\*.log

* To ignore a folder named log:

log/

* **Save** the .gitignore file and stage it:

git add .gitignore

git commit -m "Add .gitignore to ignore logs"

* **Verify** with:

git status

* The ignored files/folders will no longer appear as untracked.

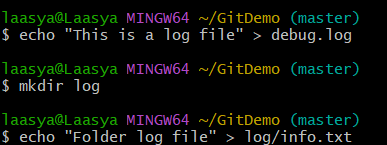
**Creating a “.log” file and a log folder in the working directory of Git. Update the .gitignore file in such a way that on committing, these files (.log extensions and log folders) are ignored. Verifying if the git status reflects the same about working directory, local repository and git repository.**

1. **Go into your project**

cd GitDemo

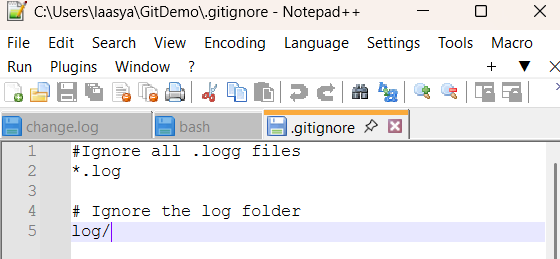


1. **Create a .log file and a log folder**

****

1. **Create a .gitignore file**

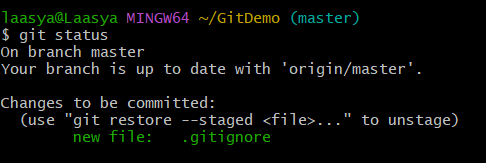
****

****

1. **Stage .gitignor**

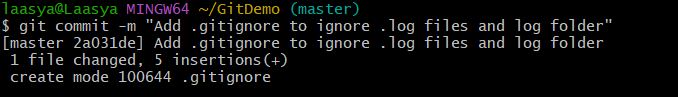
****

1. **Check git status**

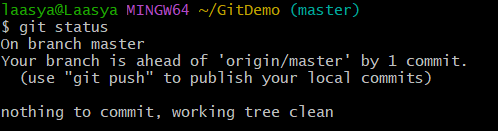


1. **Commit**

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



1. **Verify**

git status ****