**2. Git - HOL**

**1. Explain git ignore**

* **Purpose**: .gitignore is a special file used in Git to tell it which files or folders should not be tracked or committed to the repository.
* **Keeps repo clean**: It helps avoid uploading temporary, sensitive, or unnecessary files (like .log, .class, or node\_modules/) to the remote repository.
* **Customizable**: You can add specific filenames, file types (like \*.log), or folder names (like temp/) to the .gitignore file based on your project needs.

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

* **Create/Edit .gitignore**: Inside your Git project folder, open or create a file named .gitignore.
* **Add rules**: Write the filenames, file extensions (e.g., \*.log), or folder paths (e.g., build/) you want Git to ignore—one per line.
* **Check status**: When you run git status, those ignored files won't appear in the changes list if they are already being ignored.

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.

**Create a New Folder for Your Git Project**

**Using Windows File Explorer (easy way)**

1. Open **File Explorer** (the folder icon on your taskbar).
2. Go to a place where you want your project folder (like Documents or Desktop).
3. Right-click on empty space → **New** → **Folder**.
4. Name the folder, for example: GitDemo.

Now you have a folder named GitDemo where your Git project will live.

**Step 2: Open Git Bash in That Folder**

1. Right-click the folder GitDemo.
2. Click **Git Bash Here** (if you have Git installed, this option shows up).
   * This opens the terminal (command line) directly in your new folder.

**Step 3: Initialize Git Repository**

In Git Bash, type this and press **Enter**:

git init

**Step 4: Create a .log file and log folder inside your project folder using Git Bash**

In your Git Bash window (you should already be inside your GitDemo folder), run these commands one by one:

# Create an empty file named example.log

touch example.log

# Create a folder named log

mkdir log

**Step 5: Create or edit the .gitignore file to ignore .log files and the log folder**

Run this command to open the .gitignore file with a simple text editor called nano (easy for beginners):

nano .gitignore

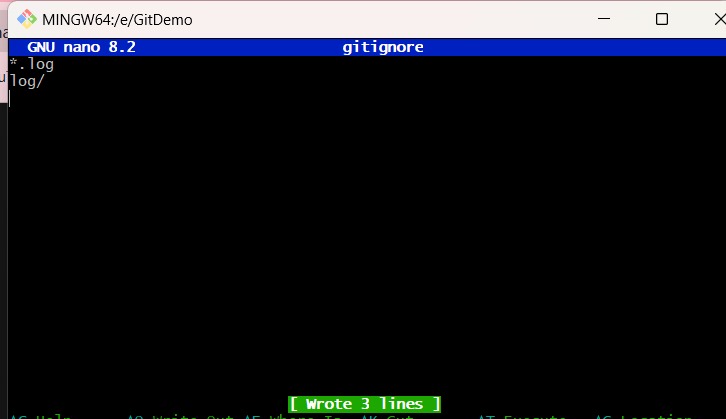
When nano opens:

* Type these two lines:

.log

log/

* To save and exit:
  + Press **Ctrl + O** (it will ask for file name, just press Enter)
  + Then press **Ctrl + X** to exit nano

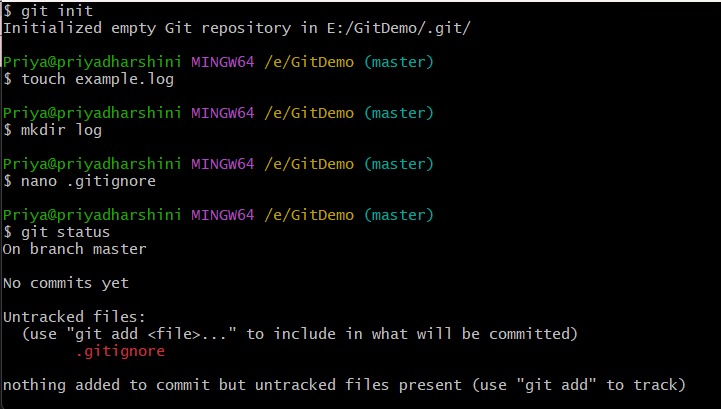


**Step 6: Check if Git is ignoring the .log file and log folder**

git status

What you want to see:

* Your .log file (example.log) and log folder **should NOT** appear in the list of untracked files.
* If you see **nothing related to those**, it means your .gitignore is working perfectly!



**Step 7: Add .gitignore and commit your changes**

Run these commands:

git add .gitignore

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

* git add .gitignore tells Git to start tracking the .gitignore file.
* git commit -m "message" saves this change with a message so you have a record.

**Step 8: Check status again**

git status

Now it should say:

On branch master

nothing to commit, working tree clean

This means:

* Your .gitignore is tracked.
* There are no untracked files visible (your .log and log are ignored correctly).
* Your working directory is clean.

