## **2. Git-HOL**

**Task 2: Using .gitignore to Ignore Unwanted Files**

**Objectives:**

* Understand .gitignore and its use.
* Ignore .log files and the log folder in Git.

**Step-by-Step Instructions with Commands**

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

* Inside your local repo (gitdemo), create:
  + A file named example.log
  + A folder named log
  + (Optional) Inside log, create a sample file, e.g., sample.txt

**Step 2: Create or edit .gitignore**

* In your gitdemo folder, create a file named .gitignore if it doesn’t exist.
* Open .gitignore in Notepad++.
* Add the following lines:

\*.log

log/

* Save and close .gitignore.

**Step 3: Rename files if needed**

* Ensure .gitignore and example.log files do **not** have .txt extensions.
* Rename them if necessary to remove .txt.

**Step 4: Check Git status**

git status

* .gitignore should appear as untracked.
* example.log and log/ should **NOT** appear (ignored).

**Step 5: Stage and commit .gitignore**

git add .gitignore

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

**Step 6: Push commit to remote**

git push origin main

**Step 7: Verify**

* On GitLab, check that .gitignore is updated.
* Confirm that .log files and log folder are not tracked.

**Notes:**

* File extensions matter: .gitignore must not have .txt.
* .gitignore rules help keep unwanted files out of your repo.