**EX-2 Git\_Hol**

**1. Explain .gitignore**

* .gitignore is a special file in a Git repository that tells Git which files or folders it should ignore (not track).
* This helps avoid committing unnecessary files like temporary files, build outputs, or system files that don’t need to be in version control.

# Ignore all .log files

\*.log

# Ignore the build folder

/build

# Ignore system files

.DS\_Store

**2. How to ignore unwanted files using .gitignore**

1. Create a .gitignore file in the root directory of your repository.
2. Add file/folder patterns you want to ignore. Examples:

\*.tmp # Ignore all .tmp files

/node\_modules # Ignore node\_modules folder

secret.txt # Ignore specific file

3. **Save the file**.

4.If files are already tracked, remove them from Git’s index before ignoring:

git rm --cached filename

5. Commit the .gitignore changes:

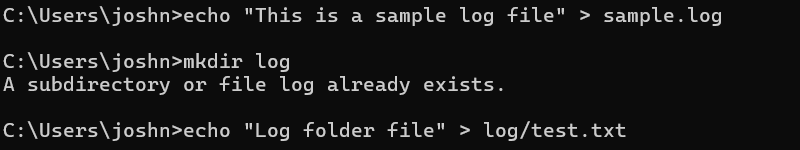
git add .gitignore

git commit -m "Add .gitignore"

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.

**1.Create files and folders to ignore**

****

**2.Create or update .gitignore file**

notepad++ .gitignore

# Ignore all .log files

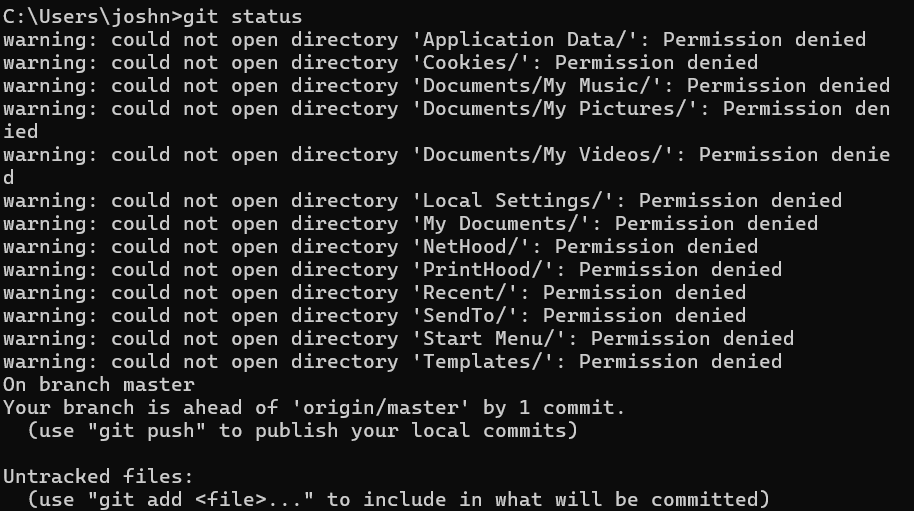
\*.log

# Ignore log folder

log/

Save and close the file.

1. **Check git status before committing**

****

4. **Add and commit changes**

Add .gitignore only (since other files are ignored):

git add .gitignore

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

git push origin main