**Objectives**

**Explain git ignore**

.gitignore is a plain text file used in Git repositories to specify which files or folders should be ignored by Git.  
Ignored files are not tracked, staged, or committed.  
It helps keep the repository clean by excluding temporary, build, log, or sensitive files from version control.

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

1. Create a file named .gitignore in the root of your Git repository.

2. Add patterns for files/folders you want Git to ignore.

Example:

\*.log # Ignore all .log files

log/ # Ignore any folder named log

3. Save the file.

4. If those files were already tracked, remove them from tracking with:

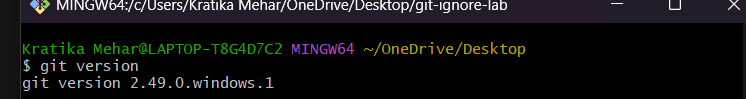
git rm --cached <file/folder>

5. Commit the .gitignore file so it applies to the repository.

**Steps**

**Step 1 – Verify Git Installation**

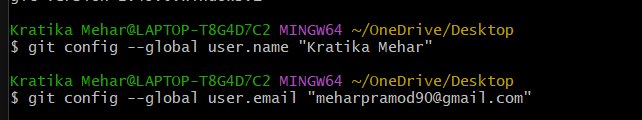
git –version



**Step 2 – Configure Git Username and Email**

git config --global user.name "Kratika Mehar"

git config --global user.email "meharpramod90@gmail.com"

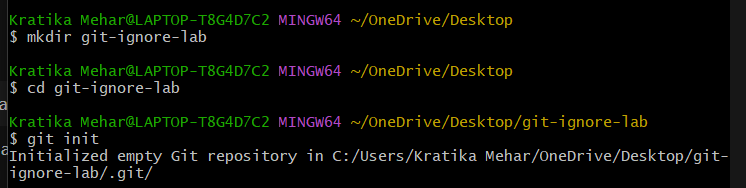
****

**Step 3 – Create a New Git Repository**

mkdir git-ignore-lab

cd git-ignore-lab

git init

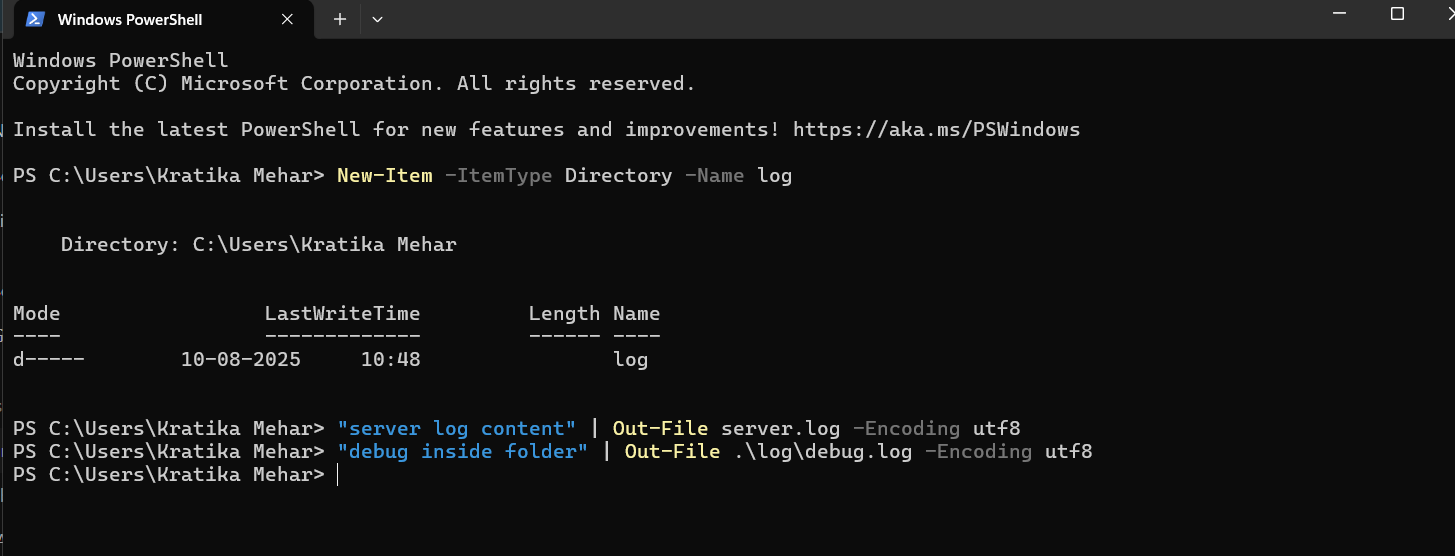
****

**Step 4 – Create .log File and log Folder**

New-Item -ItemType Directory -Name log

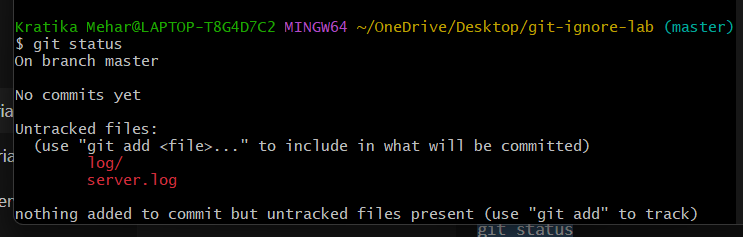
"server log content" | Out-File server.log -Encoding utf8

"debug inside folder" | Out-File .\log\debug.log -Encoding utf8



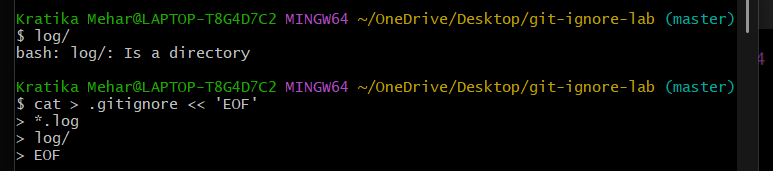
**Step 5 – Check Current Git Status**

git status



**Step 6 – Create .gitignore File**

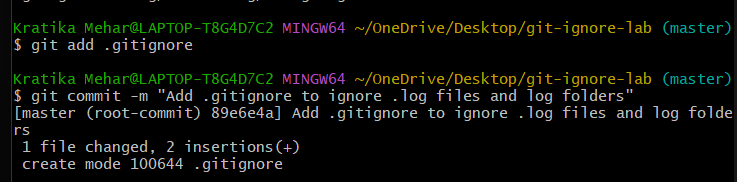
notepad .gitignore

****

**Step 7 – Stage and Commit .gitignore**

git add .gitignore

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



**Step 8 – Verify Ignored Files**

git status

git check-ignore -v server.log log/debug.log

