**1.GIT\_HOL**

PS C:\Users\HP> git --version

git version 2.50.1.windows.1

PS C:\Users\HP> git config --global user.name "SreeshmaG"

PS C:\Users\HP> git config --global user.email "sreeshgopinath33@gmail.com"

PS C:\Users\HP> Notepad++

PS C:\Users\HP> git config --global core.editor "notepad++ -multiInst -nosession"

PS C:\Users\HP> git config --global -e

hint: Waiting for your editor to close the file... notepad++ -multiInst -nosession: line 1: notepad++: command not found

error: there was a problem with the editor 'notepad++ -multiInst -nosession'

PS C:\Users\HP> cd GitDemo

PS C:\Users\HP\GitDemo> git init

Reinitialized existing Git repository in C:/Users/HP/GitDemo/.git/

PS C:\Users\HP\GitDemo> ls -a

PS C:\Users\HP\GitDemo> echo "Hello, Git World!" > welcome.txt

PS C:\Users\HP\GitDemo> ls

Directory: C:\Users\HP\GitDemo

Mode LastWriteTime Length Name

---- ------------- ------ ----

-a---- 10/08/2025 9:51 PM 40 welcome.txt

PS C:\Users\HP\GitDemo> cat welcome.txt

Hello, Git World!

PS C:\Users\HP\GitDemo> git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: welcome.txt

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: welcome.txt

PS C:\Users\HP\GitDemo> git add welcome.txt

PS C:\Users\HP\GitDemo> git commit -m "Added welcome.txt with greeting"

[master (root-commit) 776cf15] Added welcome.txt with greeting

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 welcome.txt

PS C:\Users\HP\GitDemo> git status

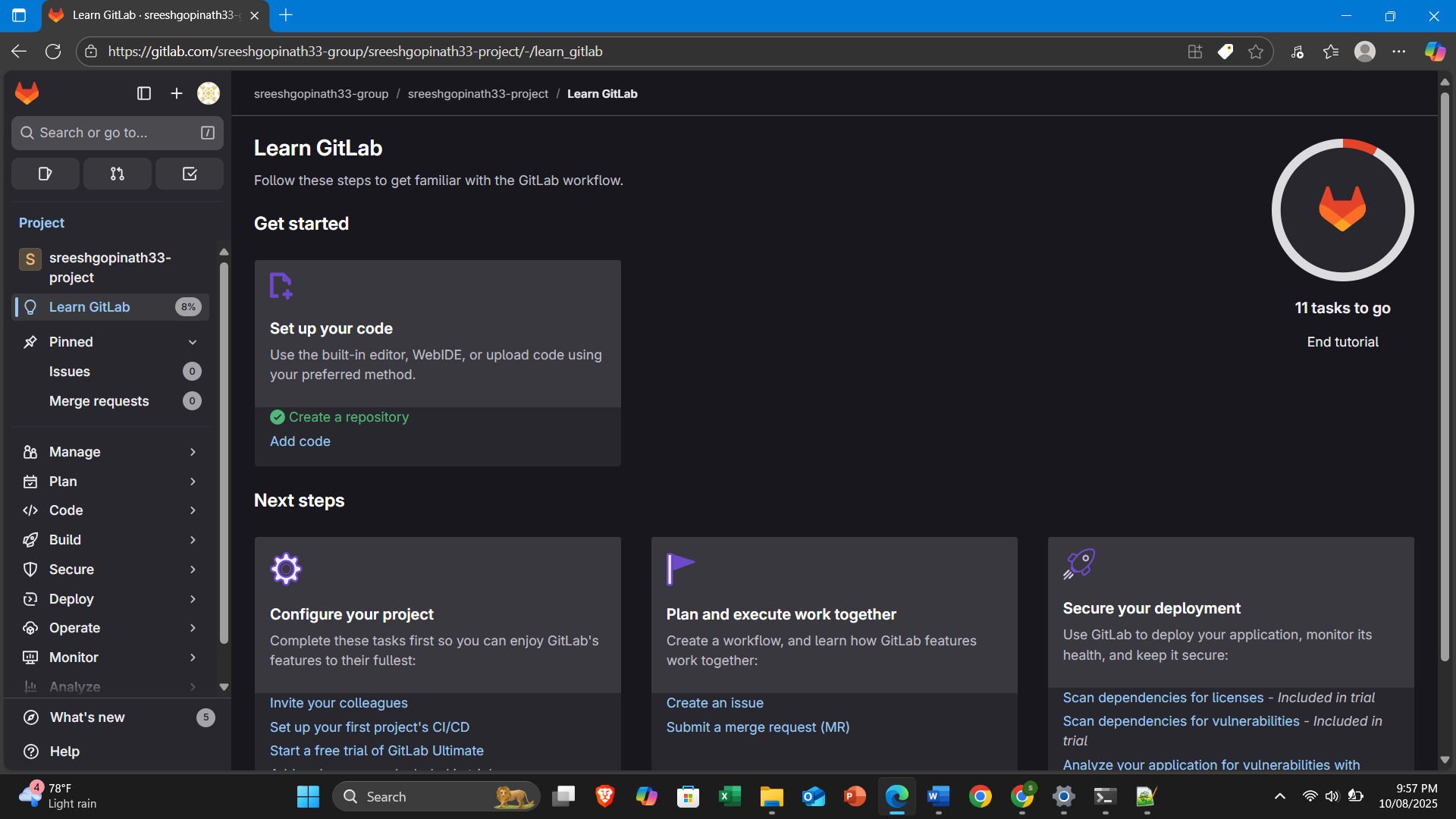
On branch master

nothing to commit, working tree clean

PS C:\Users\HP\GitDemo> git remote add origin https://gitlab.com/SreeshmaG/GitDemo.git

PS C:\Users\HP\GitDemo> git pull origin master --allow-unrelated-histories

info: please complete authentication in your browser...



**2.GIT\_HOL**

1. Understanding .gitignore

* What is it?  
  .gitignore is a special text file where you list file patterns that Git should ignore — meaning Git won’t track or commit them.
* Why use it?  
  To avoid committing:
  + Temporary files (.log, .tmp, .swp)
  + OS-specific files (Thumbs.db, .DS\_Store)
  + Build artifacts (/dist, /build)

PS C:\Users\HP\GitDemo> echo "This is a log file" > debug.log

PS C:\Users\HP\GitDemo> mkdir log

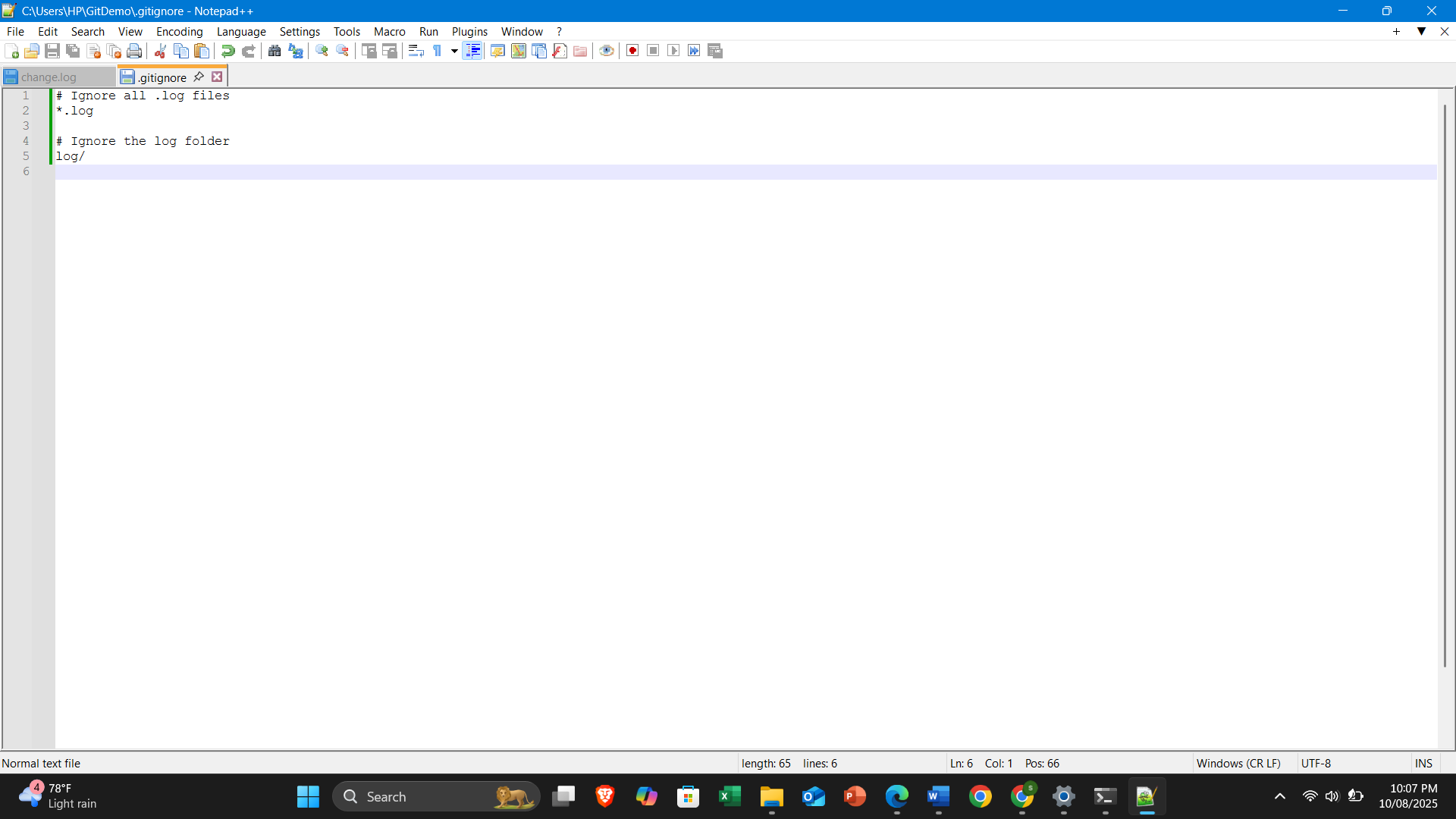
Directory: C:\Users\HP\GitDemo

Mode LastWriteTime Length Name

---- ------------- ------ ----

d----- 10/08/2025 10:05 PM log

PS C:\Users\HP\GitDemo> notepad++ .gitignore



PS C:\Users\HP\GitDemo> notepad++ .gitignore

PS C:\Users\HP\GitDemo> git status

On branch master

Untracked files:

(use "git add <file>..." to include in what will be committed)

.gitignore

nothing added to commit but untracked files present (use "git add" to track)

PS C:\Users\HP\GitDemo> git add .gitignore

PS C:\Users\HP\GitDemo> git commit -m "Added .gitignore to ignore .log files and log folder"

[master 65643e9] Added .gitignore to ignore .log files and log folder

1 file changed, 5 insertions(+)

create mode 100644 .gitignore

PS C:\Users\HP\GitDemo> git status

On branch master

nothing to commit, working tree clean

PS C:\Users\HP\GitDemo> git push origin master

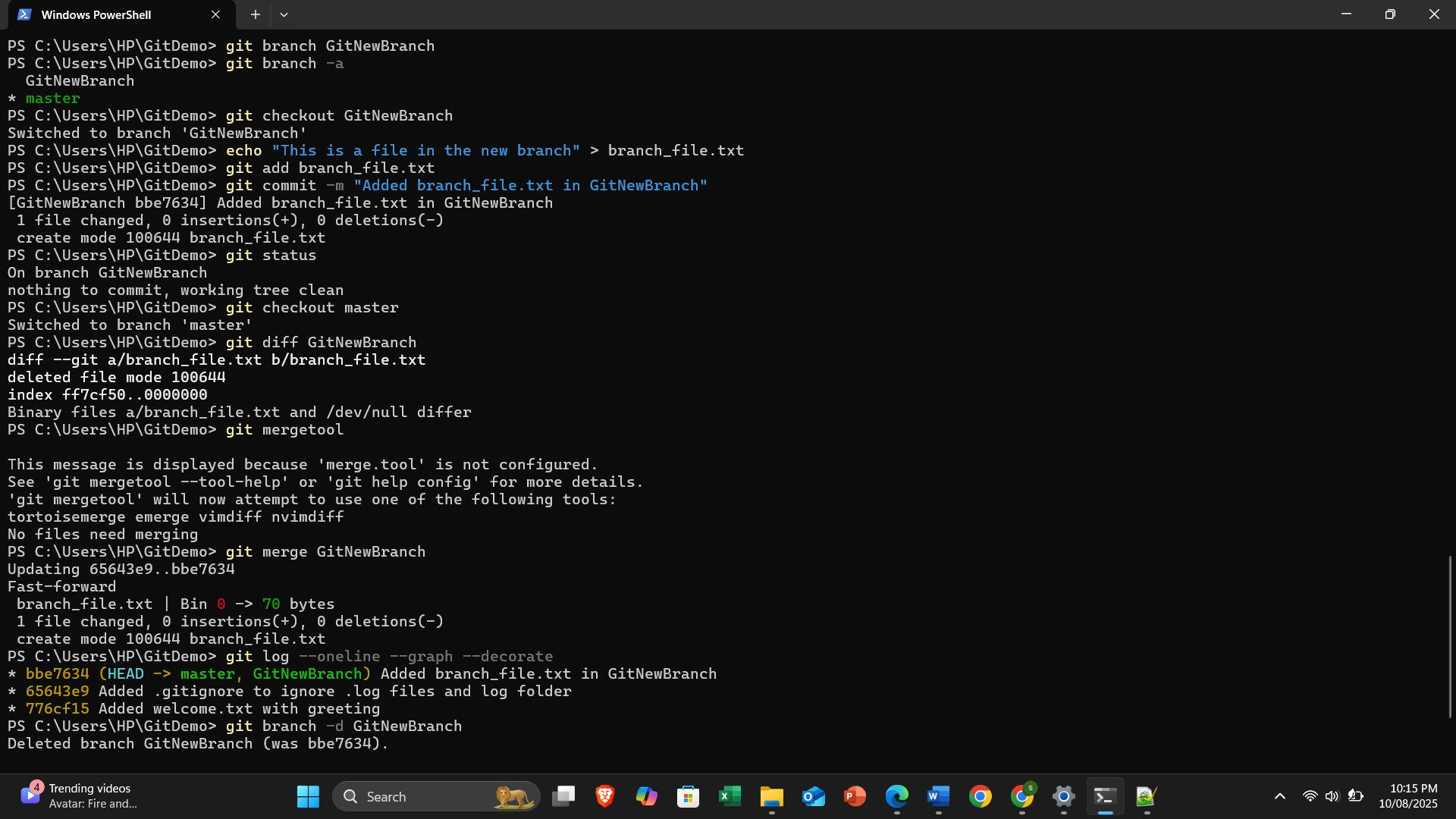
**3.GIT\_HOL**

**Branching**

* In Git, a branch is like a separate workspace for your code changes.
* master (or main) is usually the default branch.
* Creating a branch lets you work without affecting the main code until you merge.

**Merging**

* Merging combines changes from one branch into another.
* You can merge via:
  + **Git command line**
  + **Merge request** in GitLab (recommended for collaboration



**4.GIT\_HOL**

