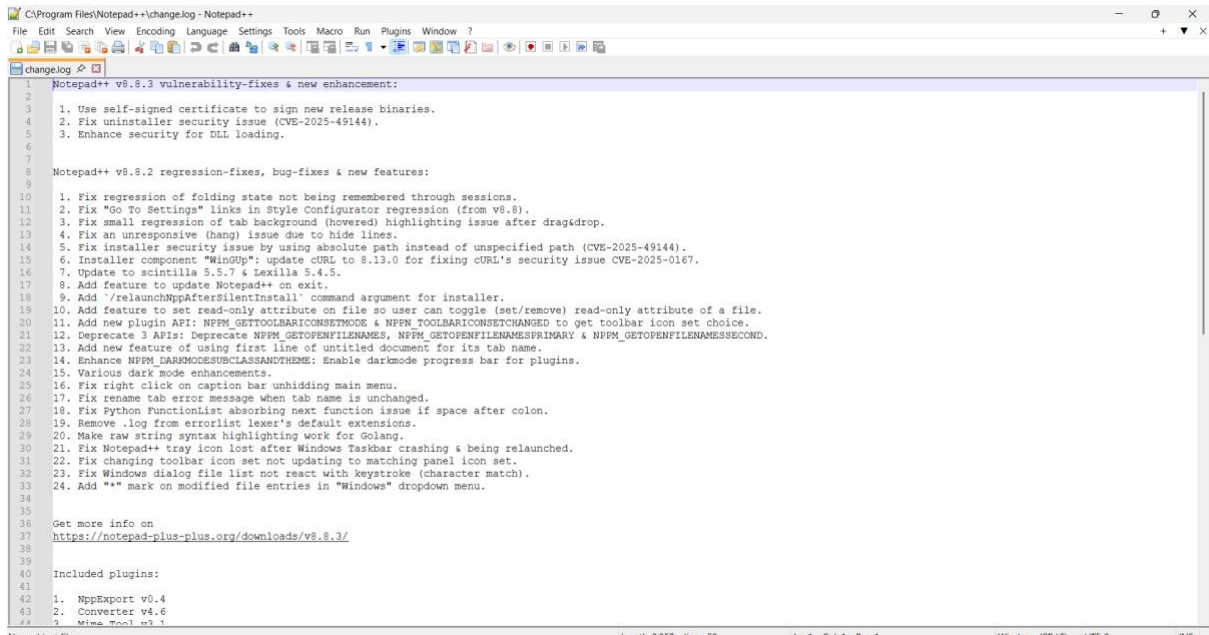


# WEEK 8 – GIT

1. In this hands-on lab, you will learn how to

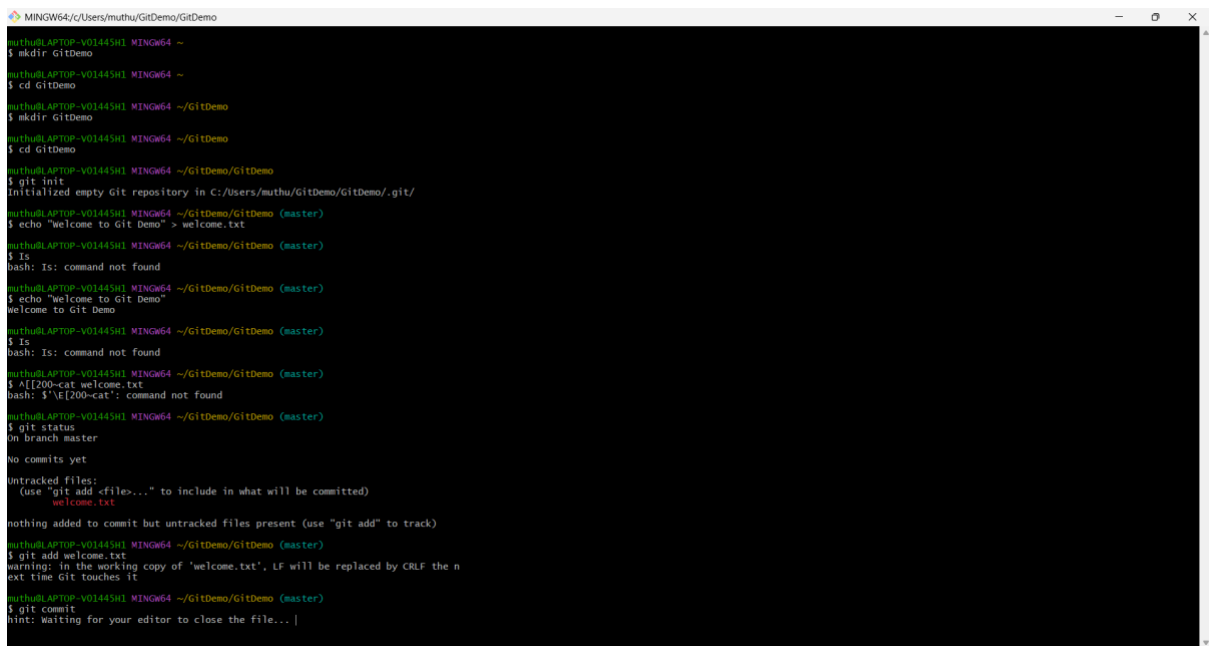
- Setup your machine with Git Configuration
- Integrate notepad++.exe to Git and make it a default editor
- Add a file to source code repository

## OUTPUT:



The screenshot shows the Notepad++ application window with the file 'C:\Program Files\Notepad++\changelog - Notepad++.txt' open. The text in the editor is as follows:

```
1 Notepad++ v8.8.3 vulnerability-fixes & new enhancement:
2
3 1. Use self-signed certificate to sign new release binaries.
4 2. Fix uninstaller security issue (CVE-2025-49144).
5 3. Enhance security for DLL loading.
6
7
8 Notepad++ v8.8.2 regression-fixes, bug-fixes & new features:
9
10 1. Fix regression of folding state not being remembered through sessions.
11 2. Fix "Go To Settings" links in Style Configurator regression (from v8.8).
12 3. Fix small regression of tab background (hovered) highlighting issue after drag&drop.
13 4. Fix an unresponsive (hang) issue due to hide lines.
14 5. Fix installer security issue by using absolute path instead of unspecified path (CVE-2025-49144).
15 6. Installer component "WinUp": update cURL to 8.13.0 for fixing cURL's security issue CVE-2025-0167.
16 7. Update to scintilla 5.5.7 & Lexilla 5.4.5.
17 8. Add feature to update Notepad++ on exit.
18 9. Add --relaunchNppAfterSilentInstall command argument for installer.
19 10. Add feature to set read-only attribute on file so user can toggle (set/remove) read-only attribute of a file.
20 11. Add new plugin API: NPPM_GETTOOLBARICONSETMODE & NPPM_TOOLBARICONSETCHANGED to get toolbar icon set choice.
21 12. Deprecate 3 APIs: Deprecate NPPM_GETOPENFILENAME, NPPM_GETOPENFILENAMEPRIMARY & NPPM_GETOPENFILENAMESECOND.
22 13. Add new feature of using first line of untitled document for its tab name.
23 14. Enhance NPPM_DARKMODESUBCLASSANTHEME: Enable darkmode progress bar for plugins.
24 15. Various dark mode enhancements.
25 16. Fix right click on caption bar unhiding main menu.
26 17. Fix rename tab error message when tab name is unchanged.
27 18. Fix Python FunctionList absorbing next function issue if space after colon.
28 19. Remove .log from errorlist lexer's default extensions.
29 20. Make raw string syntax highlighting work for Golang.
30 21. Fix Notepad++ tray icon lost after Windows Taskbar crashing & being relaunched.
31 22. Fix changing toolbar icon set not updating to matching panel icon set.
32 23. Fix Windows dialog file list not react with keystroke (character match).
33 24. Add "*" mark on modified file entries in "Windows" dropdown menu.
34
35
36 Get more info on
37 https://notepad-plus-plus.org/downloads/v8.8.3/
38
39
40 Included plugins:
41
42 1. NppExport v0.4
43 2. Converter v4.6
44 3. MimeTool v2.1
```



The screenshot shows a Windows command prompt window with the following commands and output:

```
authhuLAPTOP-V01445H1 MINGW64 --
$ mkdir GitDemo
authhuLAPTOP-V01445H1 MINGW64 --
$ cd GitDemo
authhuLAPTOP-V01445H1 MINGW64 --/GitDemo
$ mkdir GitDemo
authhuLAPTOP-V01445H1 MINGW64 --/GitDemo
$ cd GitDemo
authhuLAPTOP-V01445H1 MINGW64 --/GitDemo/GitDemo
$ git init
Initialized empty Git repository in C:/Users/muthu/GitDemo/GitDemo/.git/
authhuLAPTOP-V01445H1 MINGW64 --/GitDemo/GitDemo (master)
$ echo "welcome to Git Demo" > welcome.txt
authhuLAPTOP-V01445H1 MINGW64 --/GitDemo/GitDemo (master)
$ ls
bash: ls: command not found
authhuLAPTOP-V01445H1 MINGW64 --/GitDemo/GitDemo (master)
$ echo "welcome to Git Demo"
welcome to Git Demo
authhuLAPTOP-V01445H1 MINGW64 --/GitDemo/GitDemo (master)
$ ls
bash: ls: command not found
authhuLAPTOP-V01445H1 MINGW64 --/GitDemo/GitDemo (master)
$ cat welcome.txt
bash: $ cat: command not found
authhuLAPTOP-V01445H1 MINGW64 --/GitDemo/GitDemo (master)
$ git status
On branch master
no commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    welcome.txt
nothing added to commit but untracked files present (use "git add" to track)
authhuLAPTOP-V01445H1 MINGW64 --/GitDemo/GitDemo (master)
$ git add welcome.txt
warning: in the working copy of 'welcome.txt', LF will be replaced by CRLF the n
ext time Git touches it
authhuLAPTOP-V01445H1 MINGW64 --/GitDemo/GitDemo (master)
$ git commit
hint: waiting for your editor to close the file... |
```

## 2. Objectives

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

In this hands-on lab, you will learn how to:

- Implement git ignore command to ignore unwanted files and folders

### Prerequisites

The following are the pre-requisites to complete this hands-on lab:

- Setting up Git environment
- Integrate notepad++ as a default editor
- A Git repository in the local system and a remote repository in GitLab
- Estimated time to complete this lab: **20 minutes**.
- 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.
- 

Open **Git Bash** and run:

```
cd GitDemo
```

If not created yet:

```
mkdir GitDemo
```

```
cd GitDemo
```

```
git init
```

---

2. Create unwanted files/folders

```
echo "This is a log file" > error.log
```

```
mkdir log
```

```
echo "Log file in folder" > log/app.log
```

This creates:

- A file: error.log
- A folder: log/ containing app.log

---

3. Check Git status before ignoring

git status

Untracked files:

error.log

log/

---

✓ 4. Create a .gitignore file

notepad++ .gitignore

Add the following lines:

\*.log

log/

Explanation:

- \*.log → Ignore all .log files
- log/ → Ignore the entire log folder

Save and close the file.

---

5. Check Git status again

git status

---

6. Add and commit .gitignore only

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

---

7. Push to GitLab (if connected)

git push origin master

Use this only if your Git project is linked to a remote GitLab repo.

```
MINGW64/c/Users/muthu/GitDemo
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo/GitDemo (master)
$ cd ..
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo
$ AC
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo
$ echo "this is a log file" > error.log
mkdir log
echo "Inside log folder" > log/app.log
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo
$ git status
fatal: not a git repository (or any of the parent directories): .git
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo
$ echo "this is a log file"
this is a log file
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo
$ mkdir log
mkdir: cannot create directory 'log': File exists
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo
$ echo "Inside log folder"
Inside log folder
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo
$ git status
fatal: not a git repository (or any of the parent directories): .git
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo
$ git init
Initialized empty Git repository in C:/Users/muthu/GitDemo/.git/
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ echo "this is log file"
this is log file
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ mkdir log
mkdir: cannot create directory 'log': File exists
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ echo "this is a log file" > error.log
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ mkdir log
mkdir: cannot create directory 'log': File exists
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ AC
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ echo "Log inside folder" > log/app.log
authurLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ notepad++ .gitignore
```

3. Please follow the instruction to complete the hands-on. Each instruction expects a command for the Git Bash.

## Branching:

1. Create a new branch **"GitNewBranch"**.
2. List all the local and remote branches available in the current trunk. Observe the "\*" mark which denote the current pointing branch.
3. Switch to the newly created branch. Add some files to it with some contents.
4. Commit the changes to the branch.
5. Check the status with **"git status"** command.

## Merging:

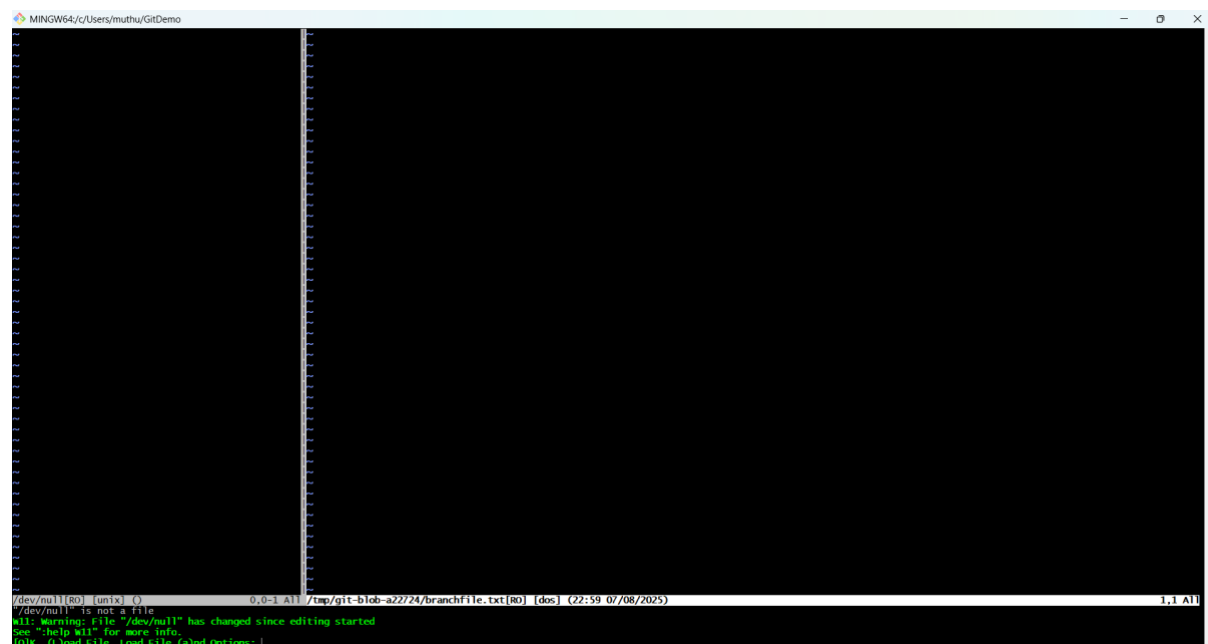
1. Switch to the master
2. List out all the differences between trunk and branch. These provide the differences in command line interface.
3. List out all the visual differences between master and branch using **P4Merge tool**.
4. Merge the source branch to the trunk.
5. Observe the logging after merging using **"git log --oneline --graph --decorate"**
6. Delete the branch after merging with the trunk and observe the git status.

output for branching:

```
MINGW64/c/Users/muthu/GitDemo
$ AC
muthu@LAPTOP-V0144561 MINGW64 ~/GitDemo (master)
$ echo "Initial File" > init.txt
muthu@LAPTOP-V0144561 MINGW64 ~/GitDemo (master)
$ git add init.txt
warning: in the working copy of 'init.txt', LF will be replaced by CRLF the next time git touches it
muthu@LAPTOP-V0144561 MINGW64 ~/GitDemo (master)
$ AC
muthu@LAPTOP-V0144561 MINGW64 ~/GitDemo (master)
$ git commit -m "Initial commit"
[master (root-commit) e3e37a3] Initial commit
1 file changed, 1 insertion(+)
create mode 100644 init.txt
muthu@LAPTOP-V0144561 MINGW64 ~/GitDemo (master)
$ git branch gitNewBranch
muthu@LAPTOP-V0144561 MINGW64 ~/GitDemo (master)
$ git branch
gitNewBranch
* master
muthu@LAPTOP-V0144561 MINGW64 ~/GitDemo (master)
$ git checkout gitNewBranch
Switched to branch 'gitNewBranch'
muthu@LAPTOP-V0144561 MINGW64 ~/GitDemo (gitNewBranch)
$ echo "This file is in gitNewBranch" > branchFile.txt
muthu@LAPTOP-V0144561 MINGW64 ~/GitDemo (gitNewBranch)
$ git add branchFile.txt
warning: in the working copy of 'branchFile.txt', LF will be replaced by CRLF the next time git touches it
muthu@LAPTOP-V0144561 MINGW64 ~/GitDemo (gitNewBranch)
$ git commit -m "Added branchFile.txt in gitNewBranch"
[gitNewBranch 50235fd] Added branchFile.txt in gitNewBranch
1 file changed, 1 insertion(+)
create mode 100644 branchFile.txt
muthu@LAPTOP-V0144561 MINGW64 ~/GitDemo (gitNewBranch)
$ git status
On branch gitNewBranch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    gitDemo/
    error.log
    log/

nothing added to commit but untracked files present (use "git add" to track)
muthu@LAPTOP-V0144561 MINGW64 ~/GitDemo (gitNewBranch)
$
```

For merging:



Estimated time to complete this lab: **10 minutes.**

Please follow the instructions to complete the hands-on. Each instruction expects a command for the Git Bash.

1. Verify if master is in clean state.
2. List out all the available branches.

3. Pull the remote git repository to the master
4. Push the changes, which are pending from “Git-T03-HOL\_002” to the remote repository.
5. Observe if the changes are reflected in the remote repository.

Output:

```

MINGW64~/c/Users/muthu/GitDemo
authuBLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ AC

authuBLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ git add
error: 'GitDemo/' does not have a commit checked out
fatal: adding files failed

authuBLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ rm -rf GitDemo/.git

authuBLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ git add GitDemo/
warning: in the working copy of 'GitDemo/welcome.txt', LF will be replaced by CRLF the next time Git touches it

authuBLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ git commit -m "added innerGit folder as a regular files"
[master 47b3877] added innerGit folder as a regular files
1 file changed, 1 insertion(+)
create mode 100644 GitDemo/welcome.txt

authuBLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 371 bytes | 371.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
to https://github.com/MUTHUGANESH03/GitDemo.git
e3e37a3..47b3877 master -> master

authuBLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ git add
warning: in the working copy of 'error.log', LF will be replaced by CRLF the next time Git touches it.
warning: in the working copy of 'log/app.log', LF will be replaced by CRLF the next time Git touches it

authuBLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ git commit -m "Completed Git-T03-HOL_002 hands-on"
[master 36c7685] Completed Git-T03-HOL_002 hands-on
2 files changed, 2 insertions(+)
create mode 100644 error.log
create mode 100644 log/app.log

authuBLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (5/5), 456 bytes | 456.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
to https://github.com/MUTHUGANESH03/GitDemo.git
47b3877..36c7685 master -> master

authuBLAPTOP-V01445H1 MINGW64 ~/GitDemo (master)
$

```

In github:

