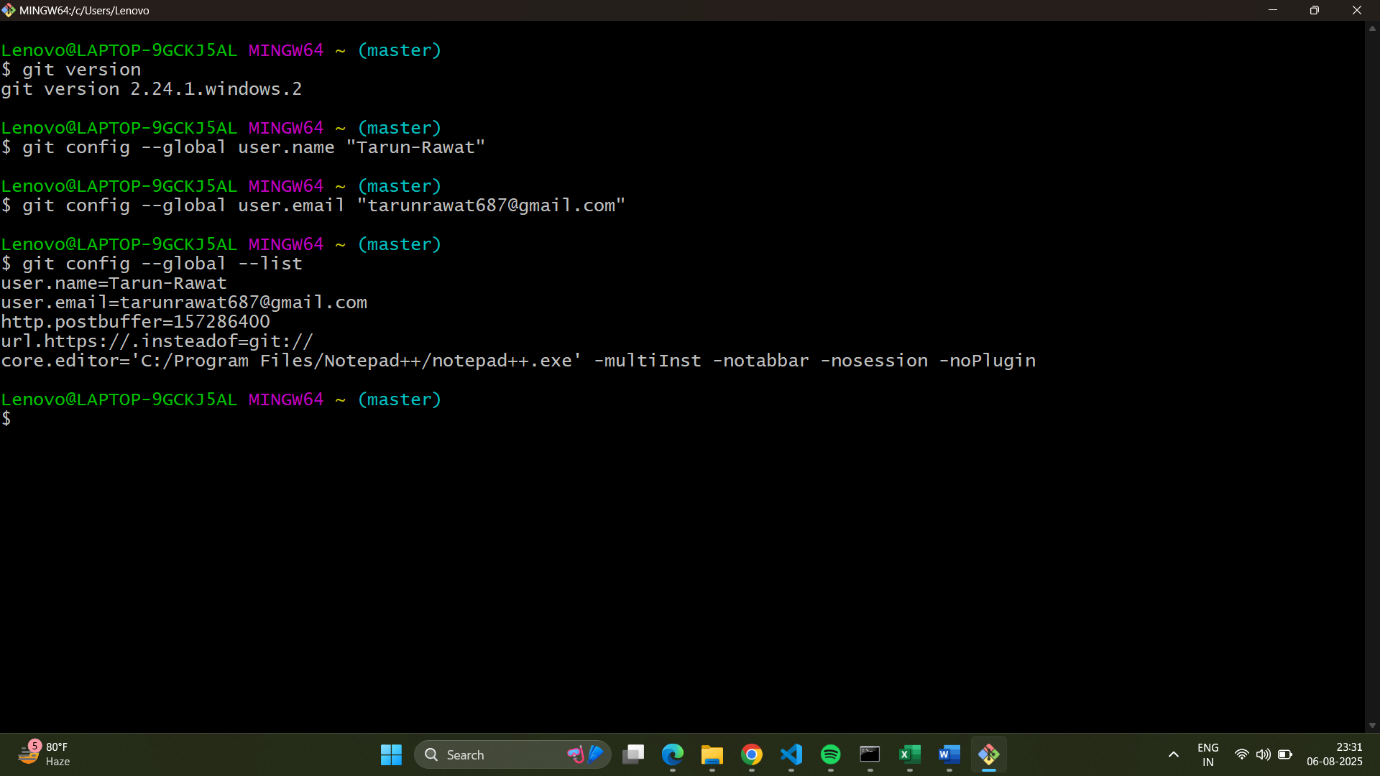
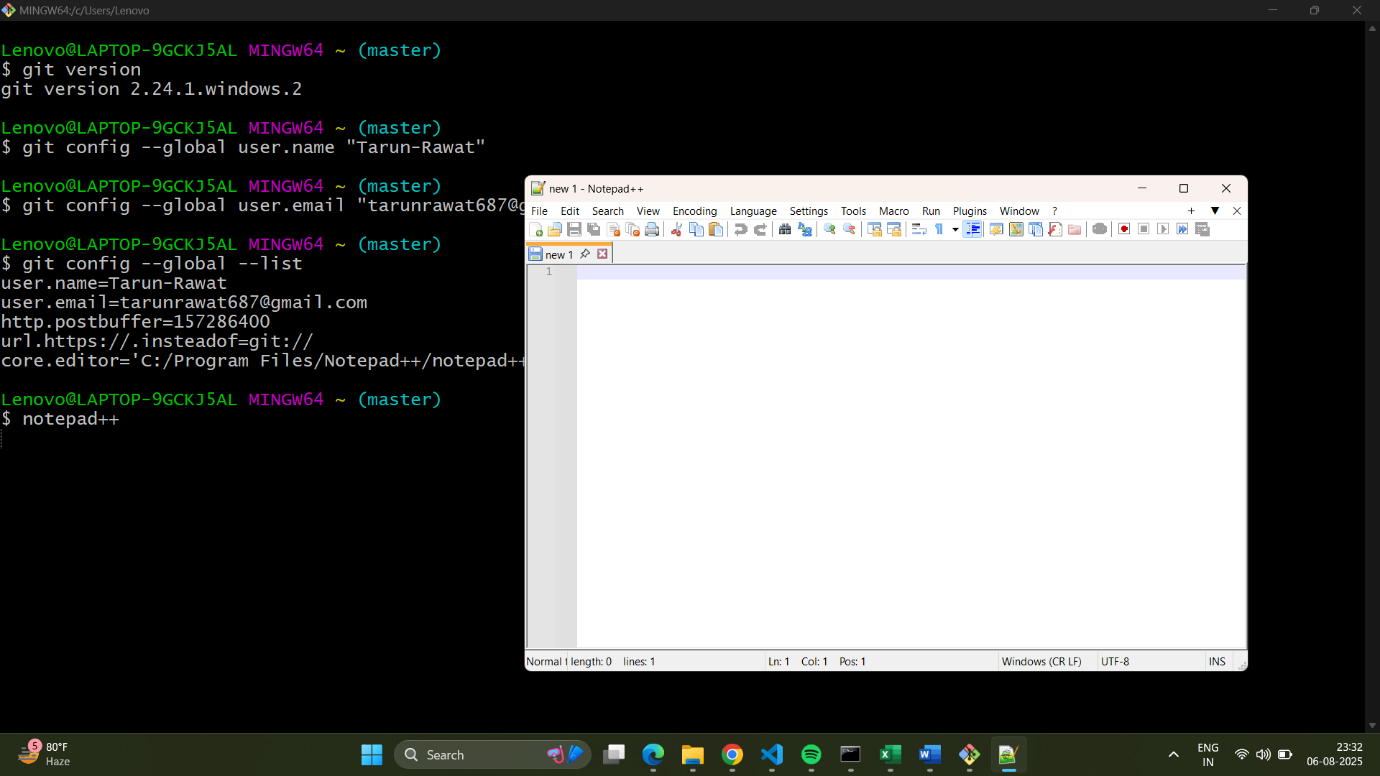
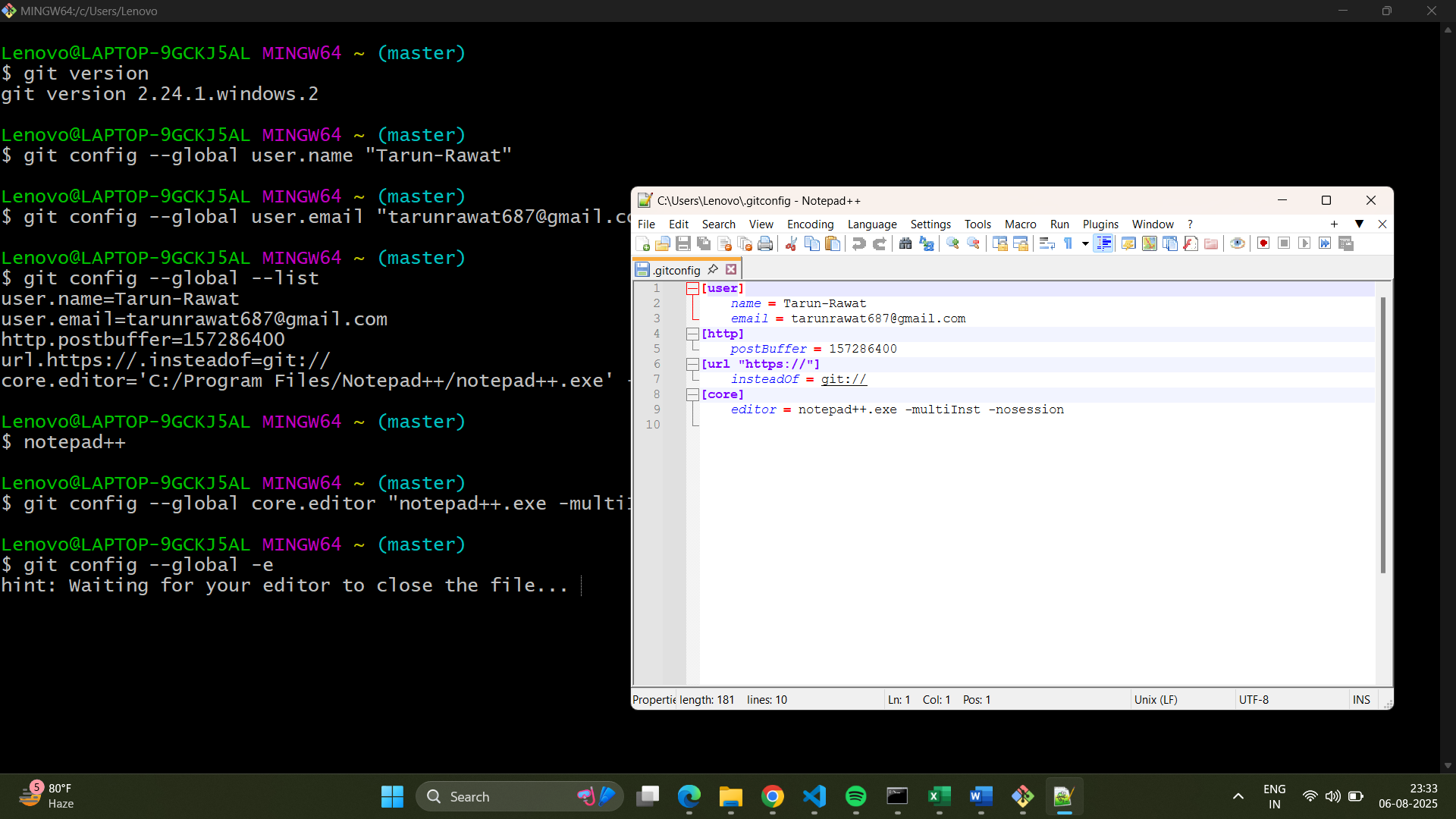
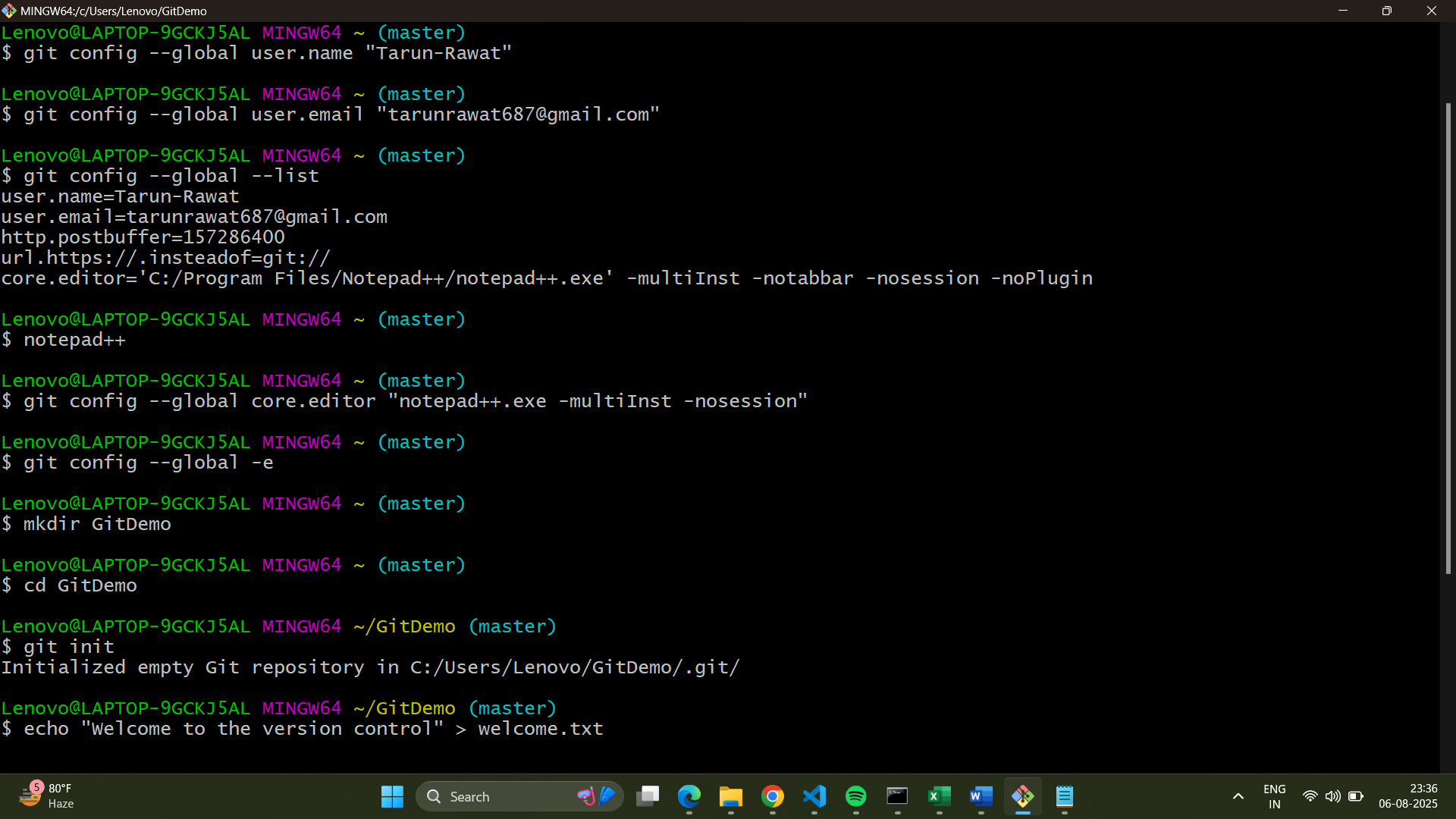
**Hands on 1**

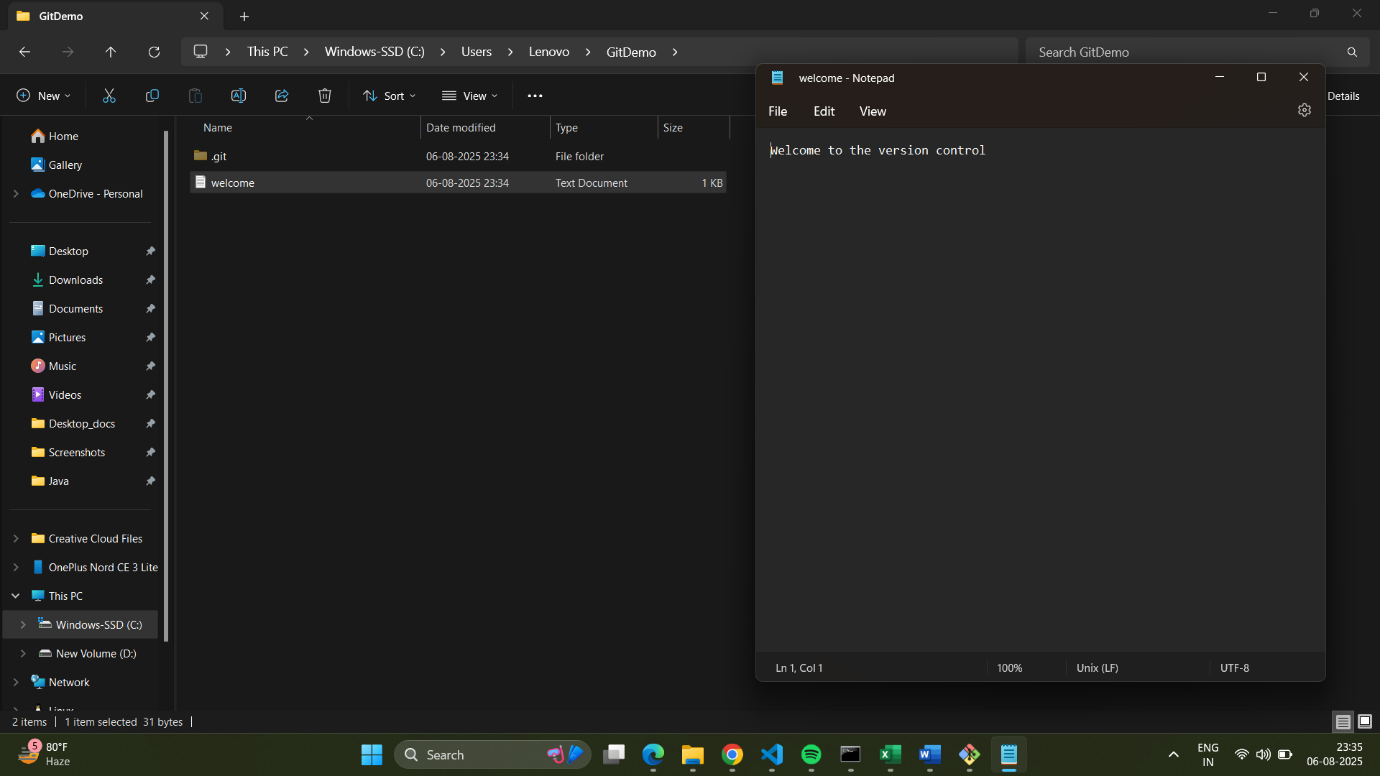
**Output:**

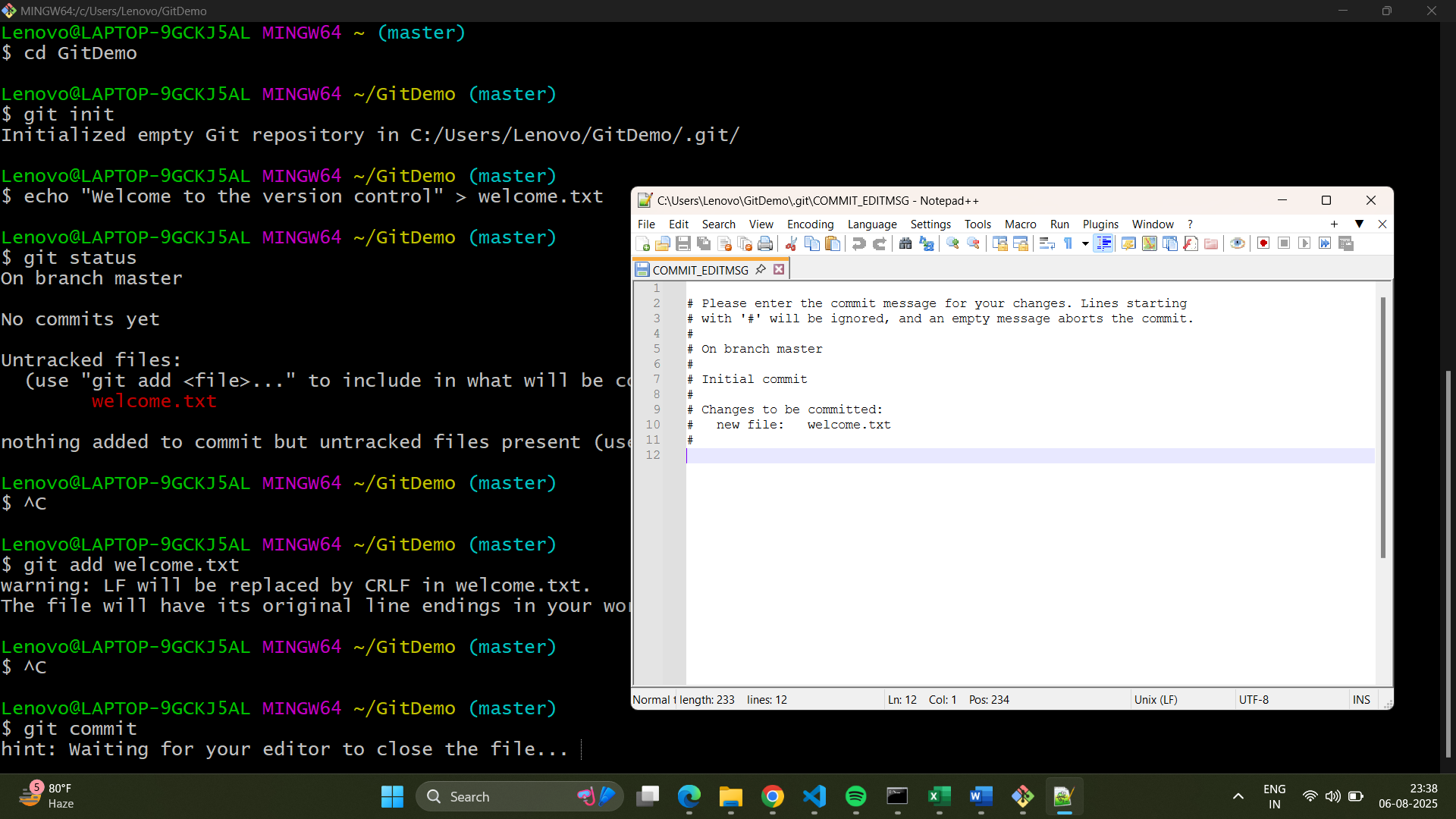


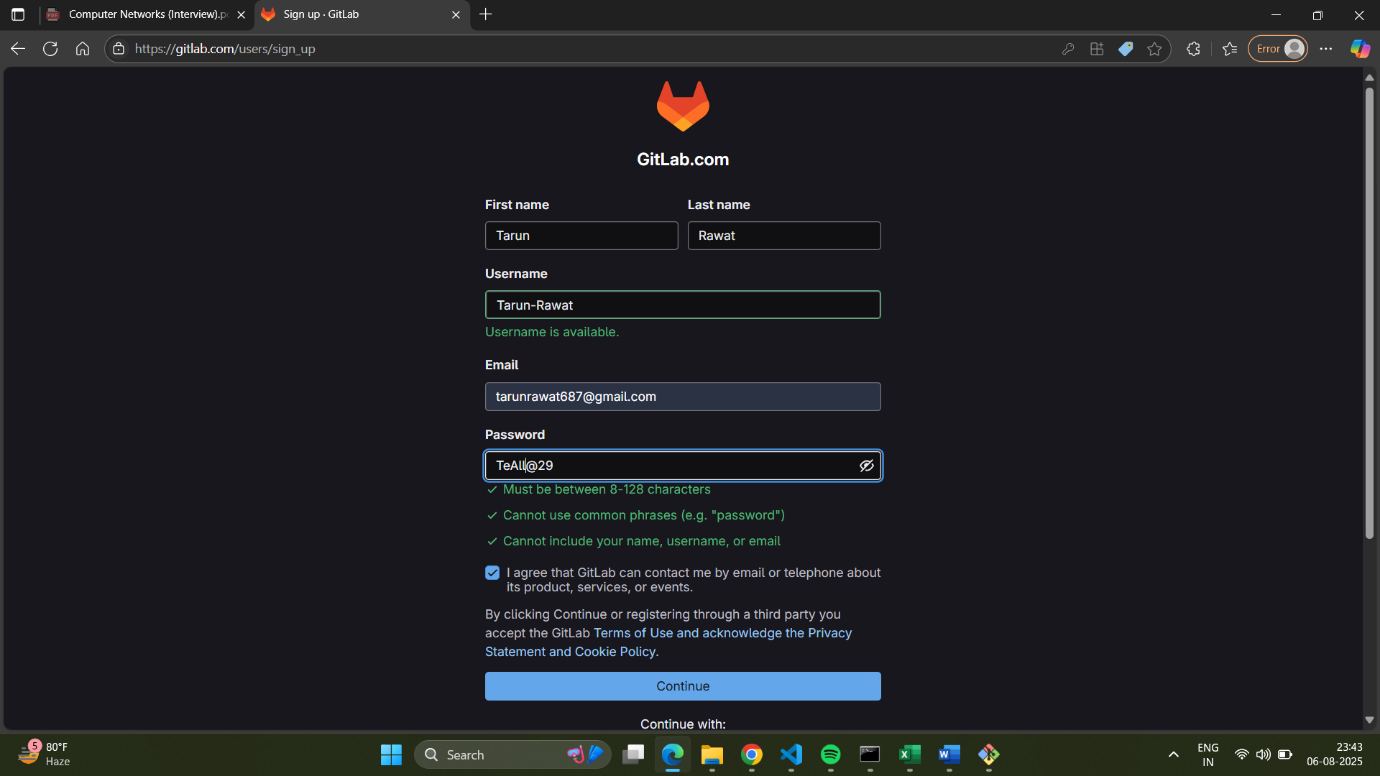


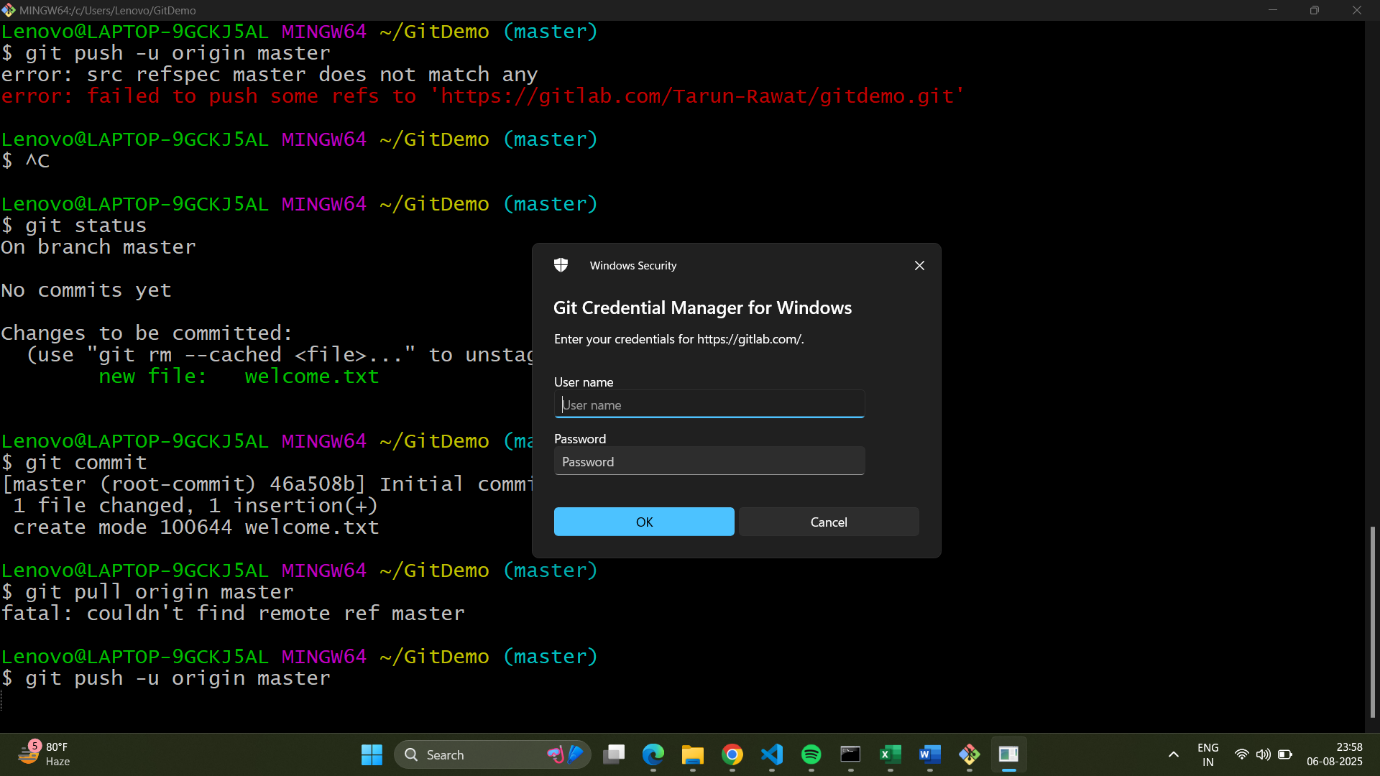


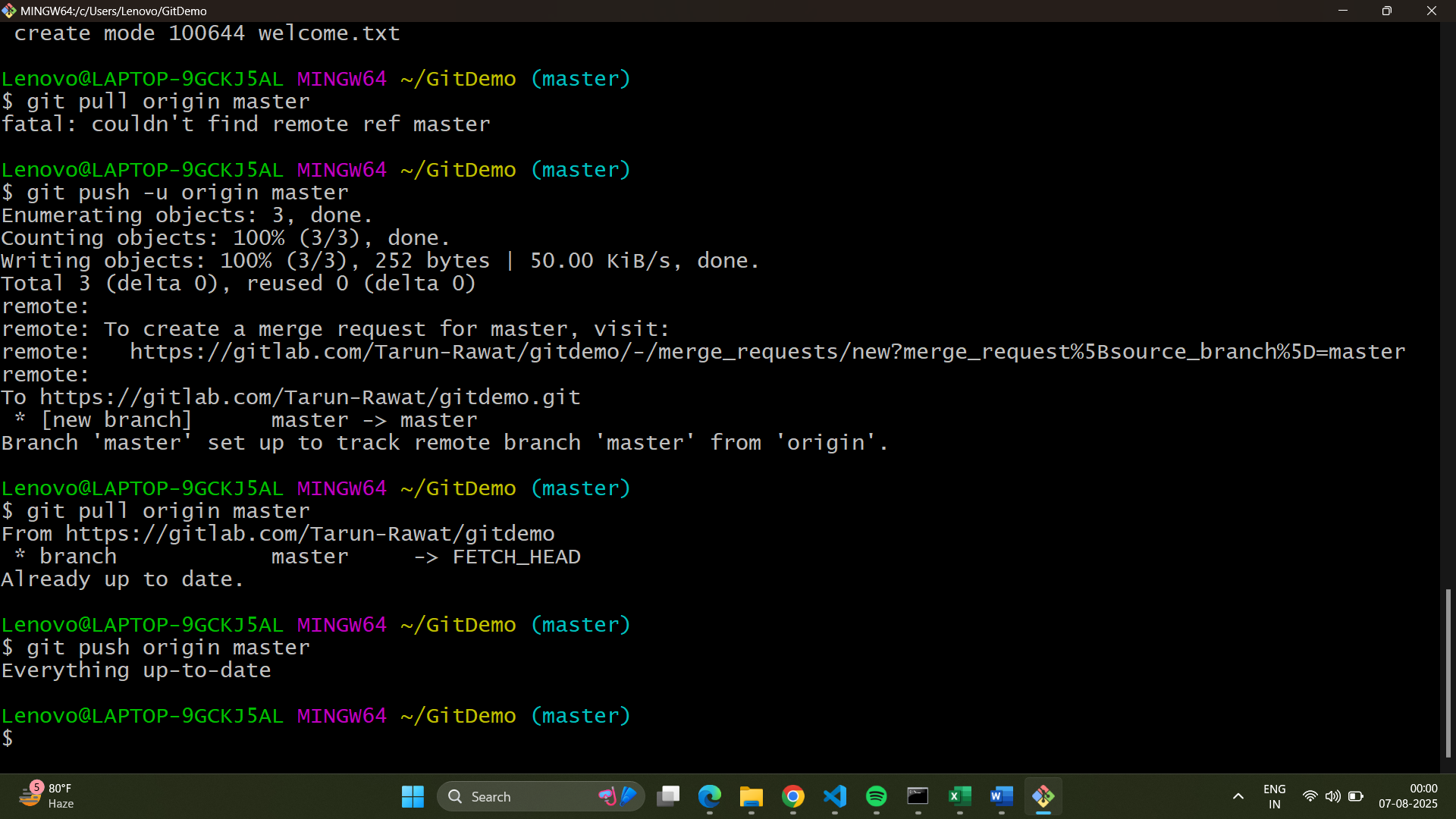
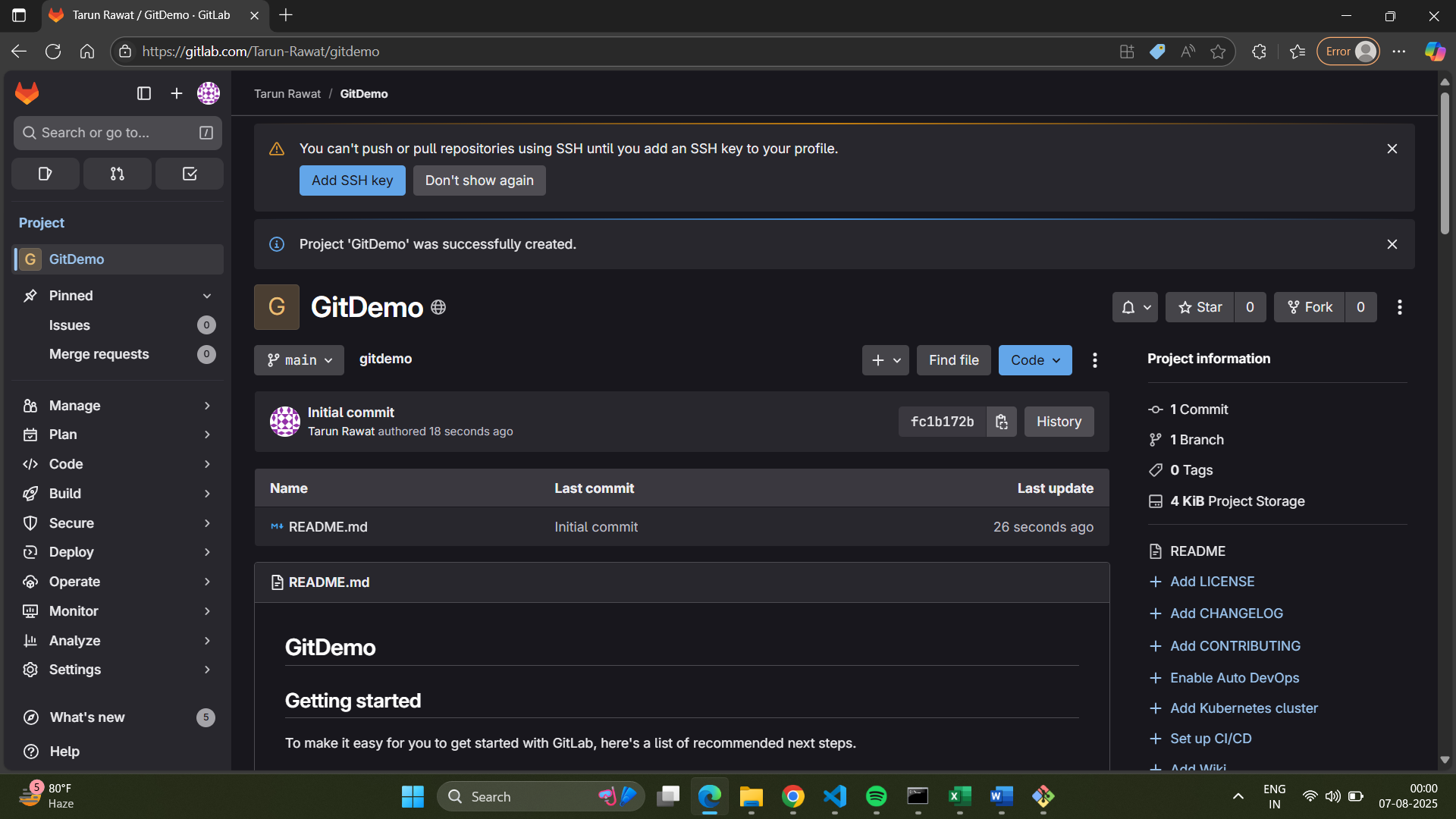










**Hands on 2**

**Commands:**1. Create the files and folders to be ignored

$ cd GitDemo

$ touch error.log

$ mkdir log

$ touch log/debug.log

2. Check the repository status before creating .gitignore

$ git status

On branch master

Your branch is up to date with 'origin/master'.

Untracked files:

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

error.log

log/

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

3. Create and edit the .gitignore file.

$ notepad++ .gitignore

On the Notepad++

# Ignore all files with a .log extension

\*.log

# Ignore the log folder and all its contents

log/

4. Check the repository status after creating .gitignore

$ git status

On branch master

Your branch is up to date with 'origin/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)

5. Add and commit the .gitignore file

$ git add .gitignore

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

[master 56fd68a] Add .gitignore to ignore log files and folders

1 file changed, 5 insertions(+)

create mode 100644 .gitignore

6. Verify the final status and push the changes

$ git status

On branch master

Your branch is ahead of 'origin/master' by 1 commit.

(use "git push" to publish your local commits)

nothing to commit, working tree clean

$ git push origin master

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 8 threads

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 387 bytes | 387.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0)

remote:

remote: To create a merge request for master, visit:

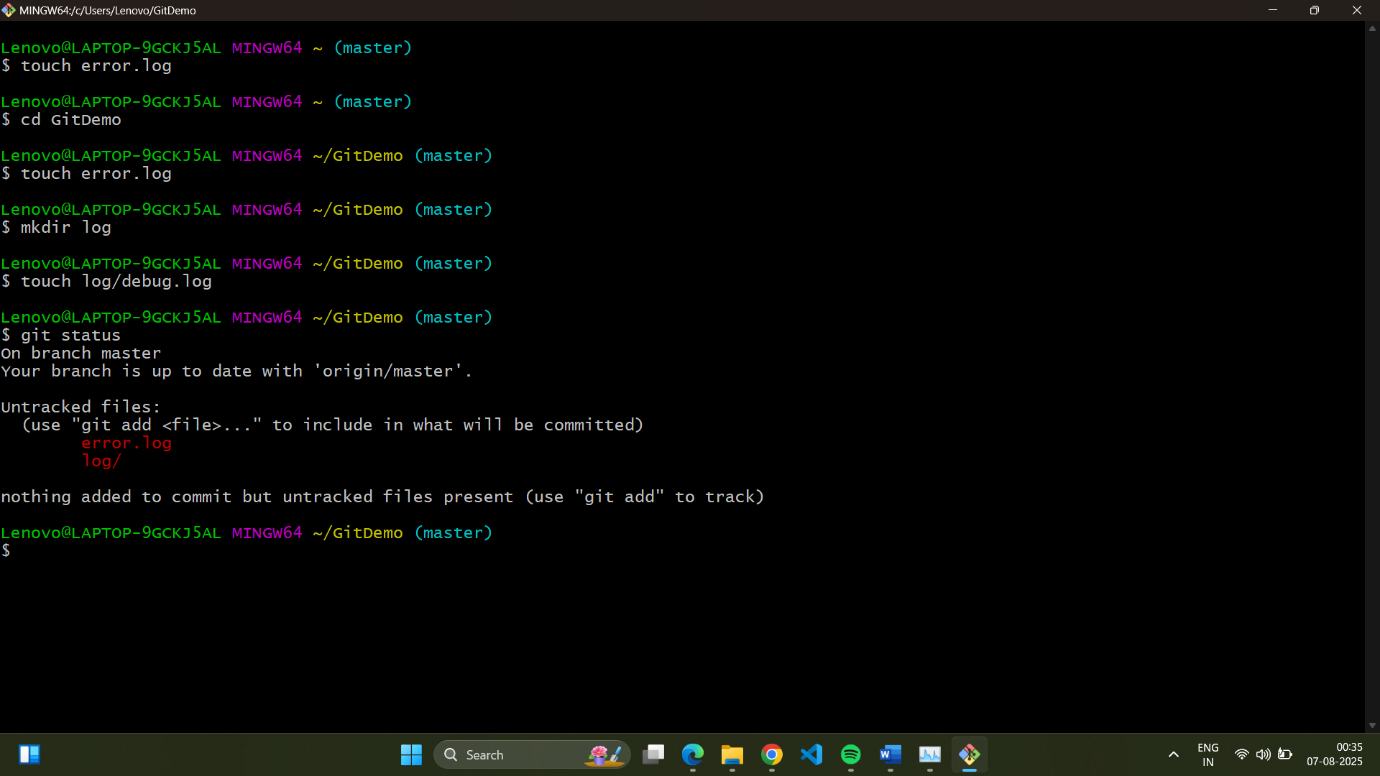
remote: https://gitlab.com/Tarun-Rawat/gitdemo/-/merge\_requests/new?merge\_request%5Bsource\_branch%5D=master

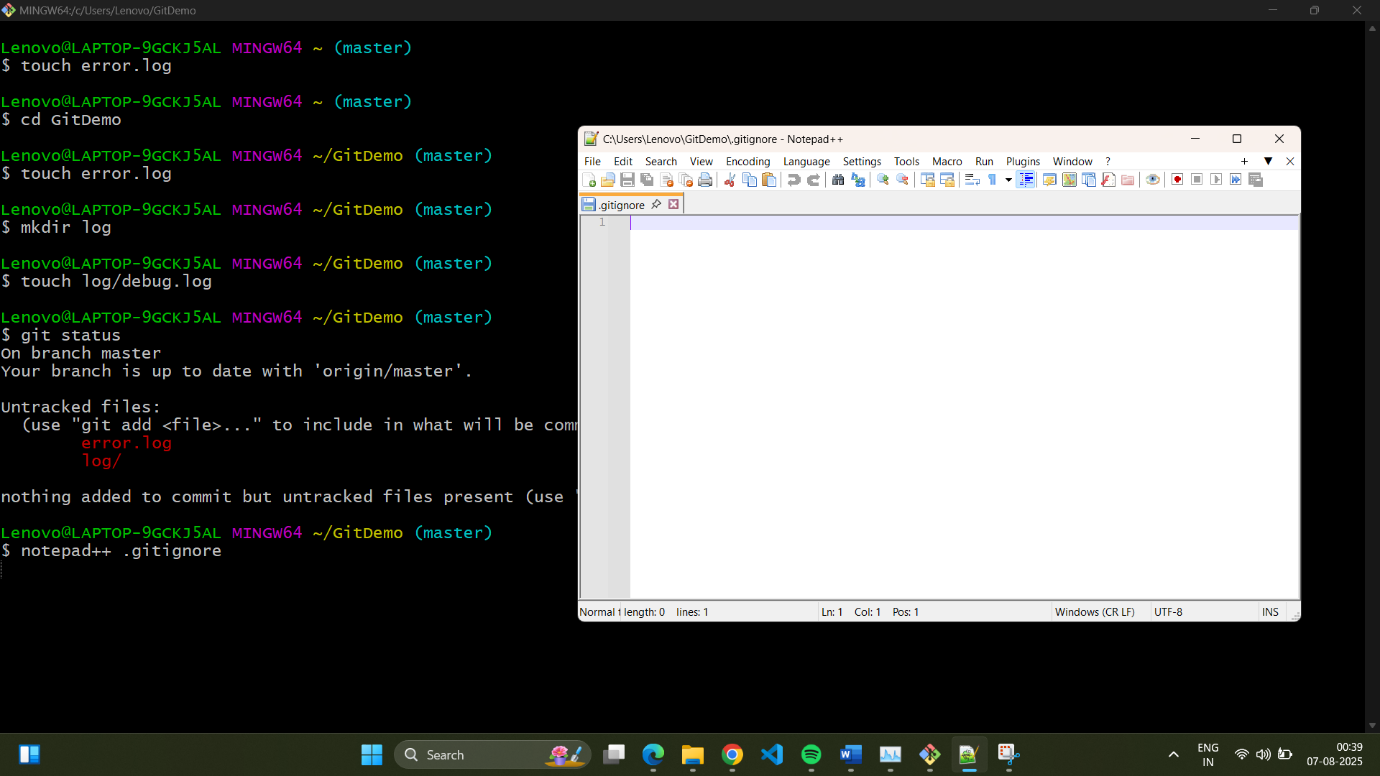
remote:

To https://gitlab.com/Tarun-Rawat/gitdemo.git

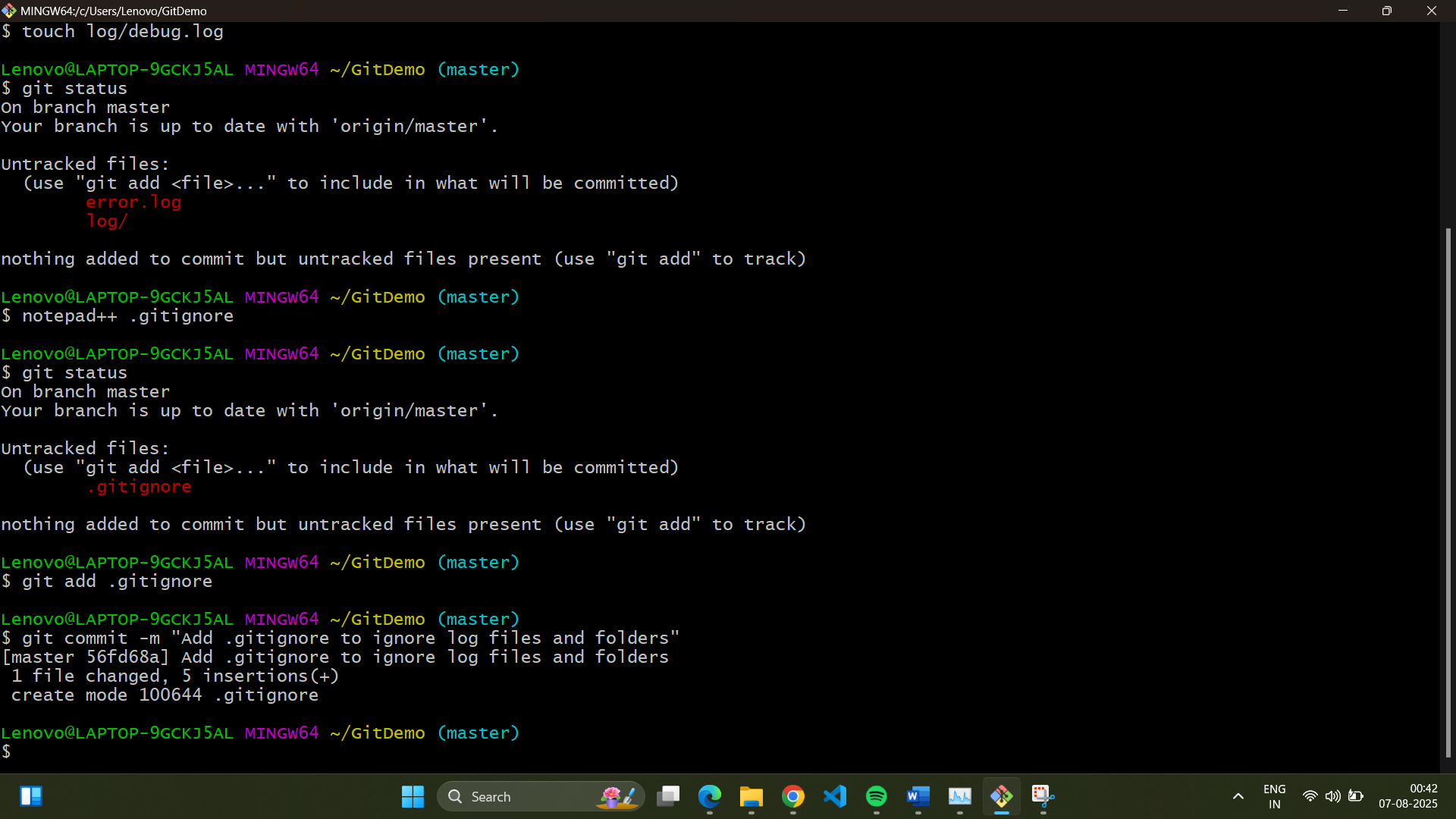
46a508b..56fd68a master -> master

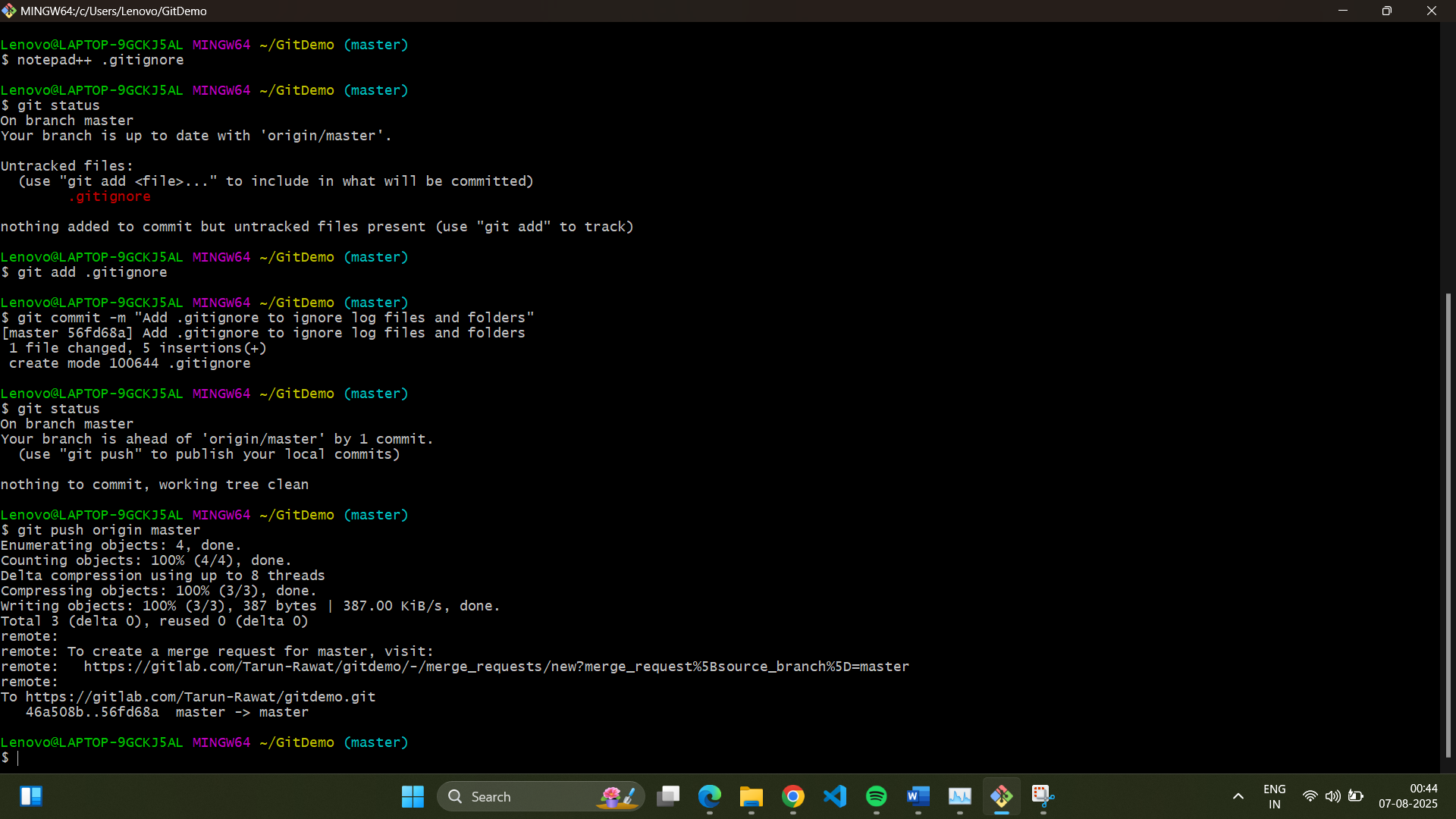
**Output:**

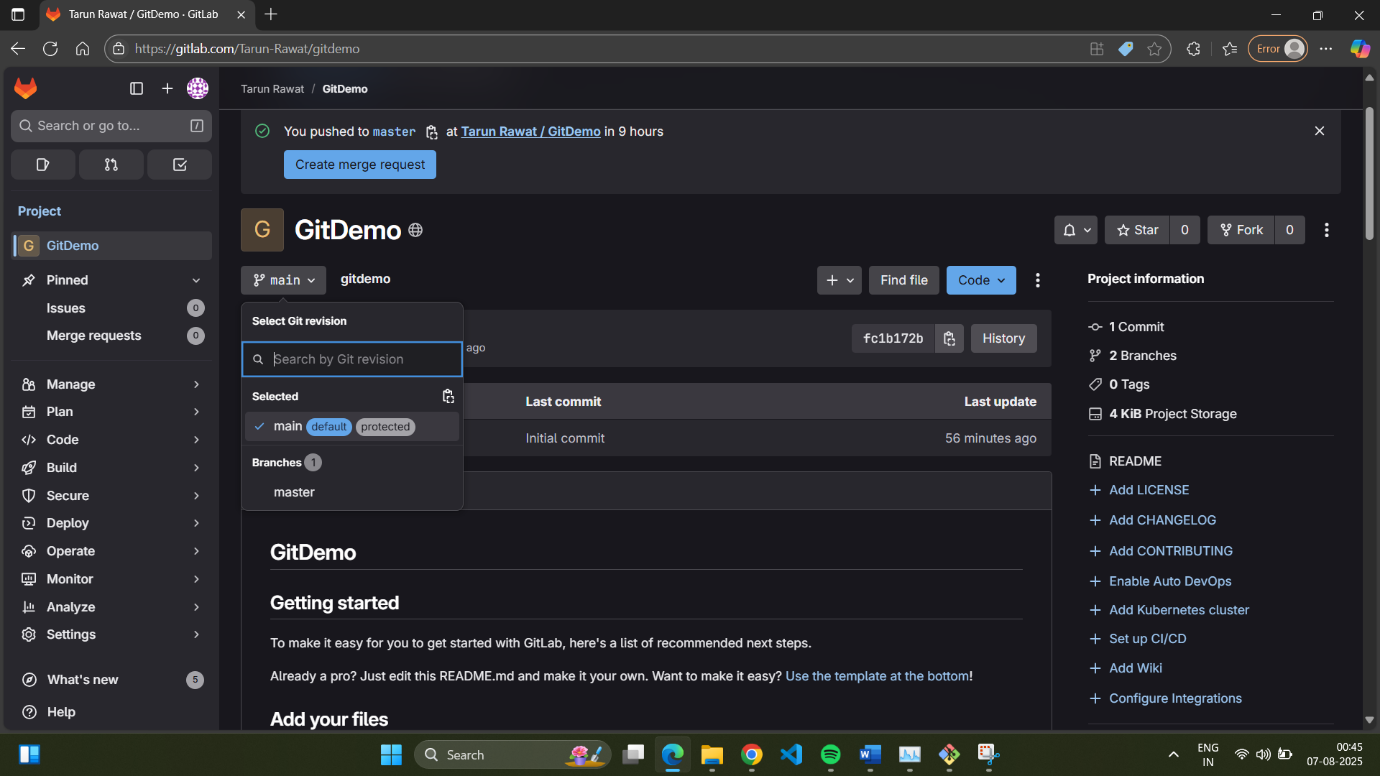


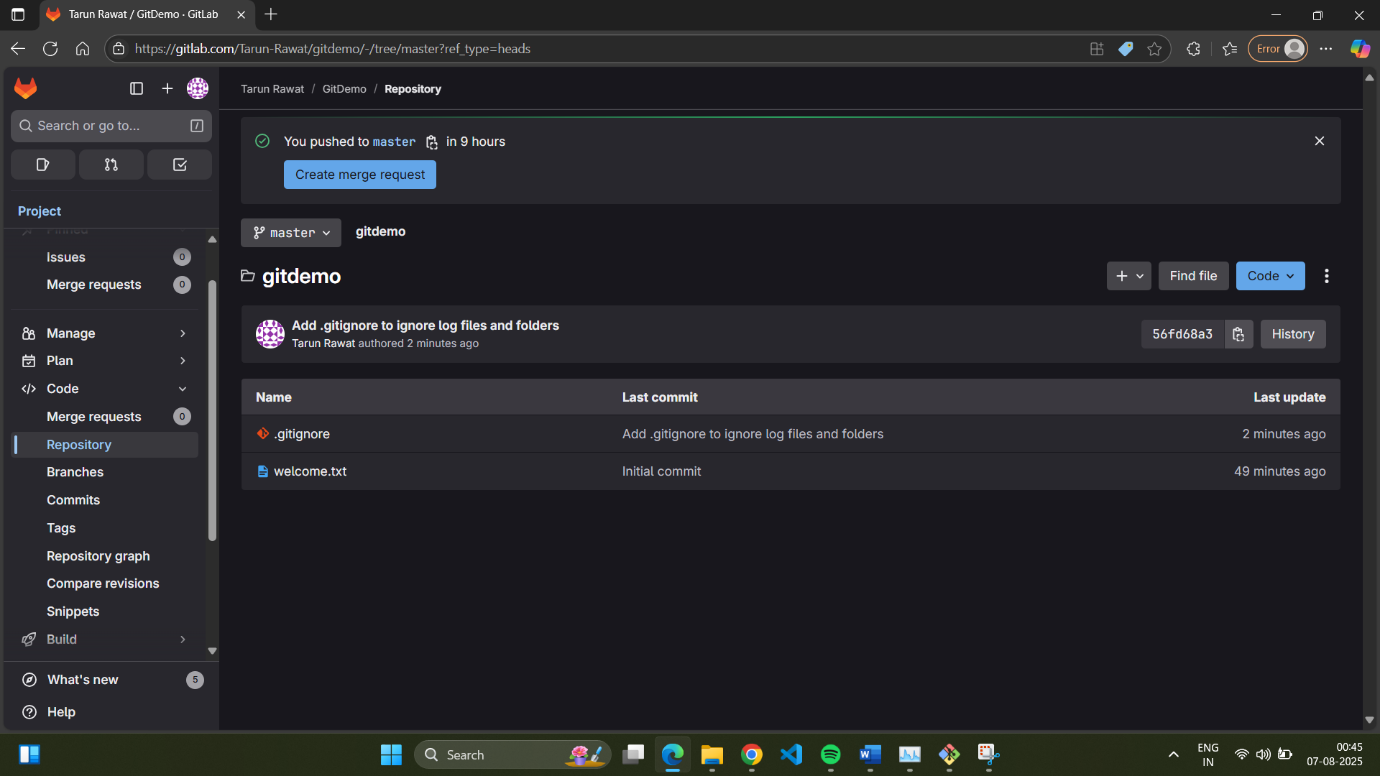












**Hands on 3**

**Commands:**

**Prerequisites**

$ git config --global diff.tool p4merge

$ git config --global difftool.p4merge.cmd "C:/Program\ Files/Perforce/p4merge.exe \"\$LOCAL\" \"\$REMOTE\""

$ git config --global mergetool.p4merge.cmd "C:/Program\ Files/Perforce/p4merge.exe \"\$BASE\" \"\$LOCAL\" \"\$REMOTE\" \"\$MERGED\""

$ git config --global mergetool.p4merge.trustExitCode false

$ git config --global mergetool.keepBackup false

**1. Branching**

1. Create a new branch

$ git branch GitNewBranch

2. Create a new branch

$ git branch -a

GitNewBranch

\* master

remotes/origin/master

3. Switch to the new branch and add a file

$ git checkout GitNewBranch

Switched to branch 'GitNewBranch'

$ echo "This is a new file created in GitNewBranch" > newfile.txt

4. Commit the changes to the new branch

$ git add newfile.txt

warning: LF will be replaced by CRLF in newfile.txt.

The file will have its original line endings in your working directory

$ git commit -m "Add newfile.txt to GitNewBranch"

[GitNewBranch 2122d54] Add newfile.txt to GitNewBranch

1 file changed, 1 insertion(+)

create mode 100644 newfile.txt

5. Check the repository status

$ git status

On branch GitNewBranch

nothing to commit, working tree clean

**Merging**

1. Switch to the master branch

$ git checkout master

Switched to branch 'master'

Your branch is up to date with 'origin/master'.

2. List differences between branches

$ git diff master GitNewBranch

**diff --git a/newfile.txt b/newfile.txt**

**new file mode 100644**

**index 0000000..e55e644**

**--- /dev/null**

**+++ b/newfile.txt**

@@ -0,0 +1 @@

+This is a new file created in GitNewBranch

3. List visual differences using P4Merge

$ git difftool master GitNewBranch

4. Merge the source branch to the trunk

$ git merge GitNewBranch

Updating 56fd68a..2122d54

Fast-forward

newfile.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 newfile.txt

5. Observe the logging after merging

$ git log --oneline --graph --decorate

\* 2122d54 (HEAD -> master, GitNewBranch) Add newfile.txt to GitNewBranch

\* 56fd68a (origin/master) Add .gitignore to ignore log files and folders

\* 46a508b Initial commit

6. Delete the branch after merging

$ git branch -d GitNewBranch

Deleted branch GitNewBranch (was 2122d54).

$ git status

On branch master

Your branch is ahead of 'origin/master' by 1 commit.

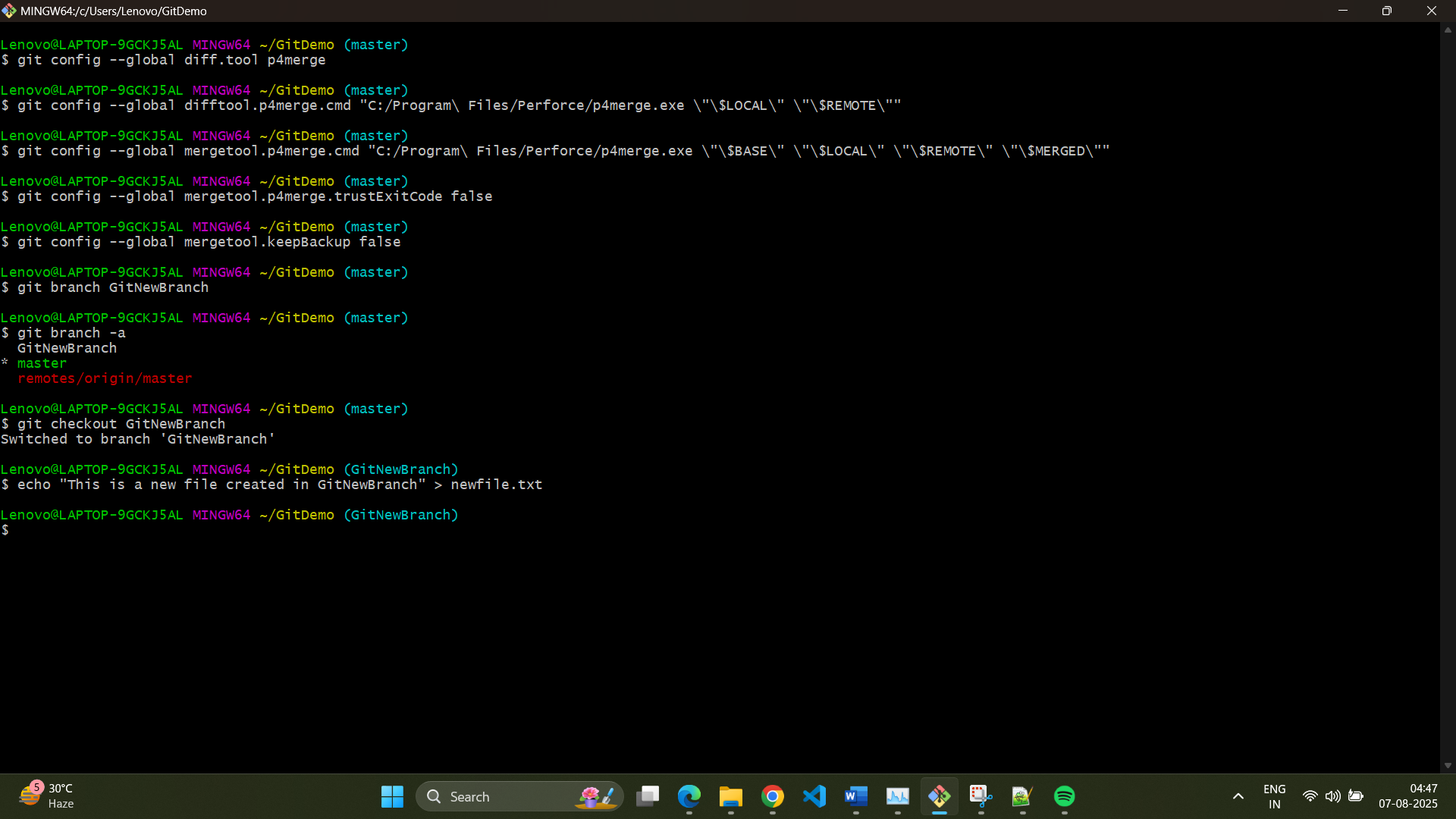
(use "git push" to publish your local commits)

nothing to commit, working tree clean

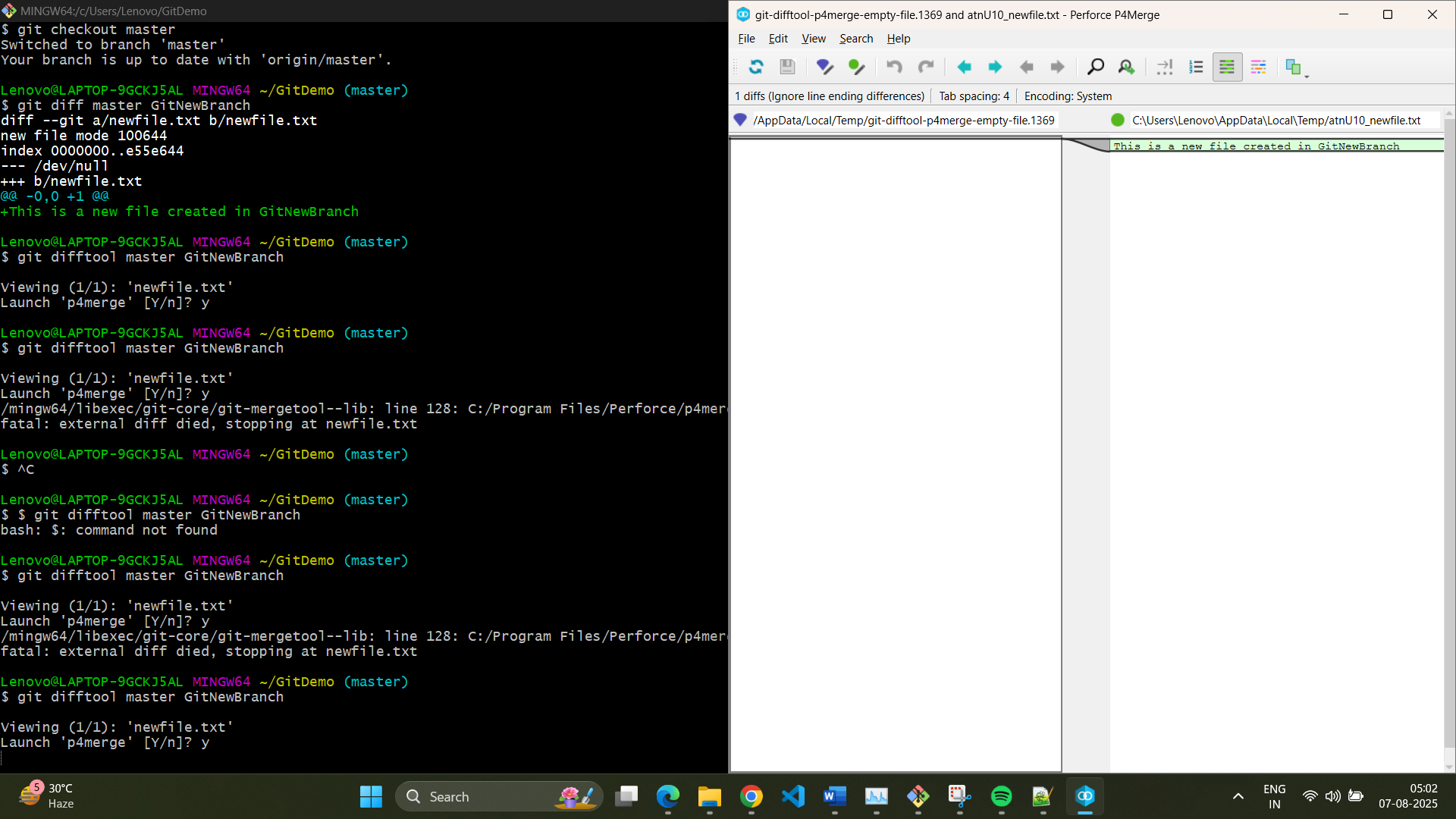
$ git branch

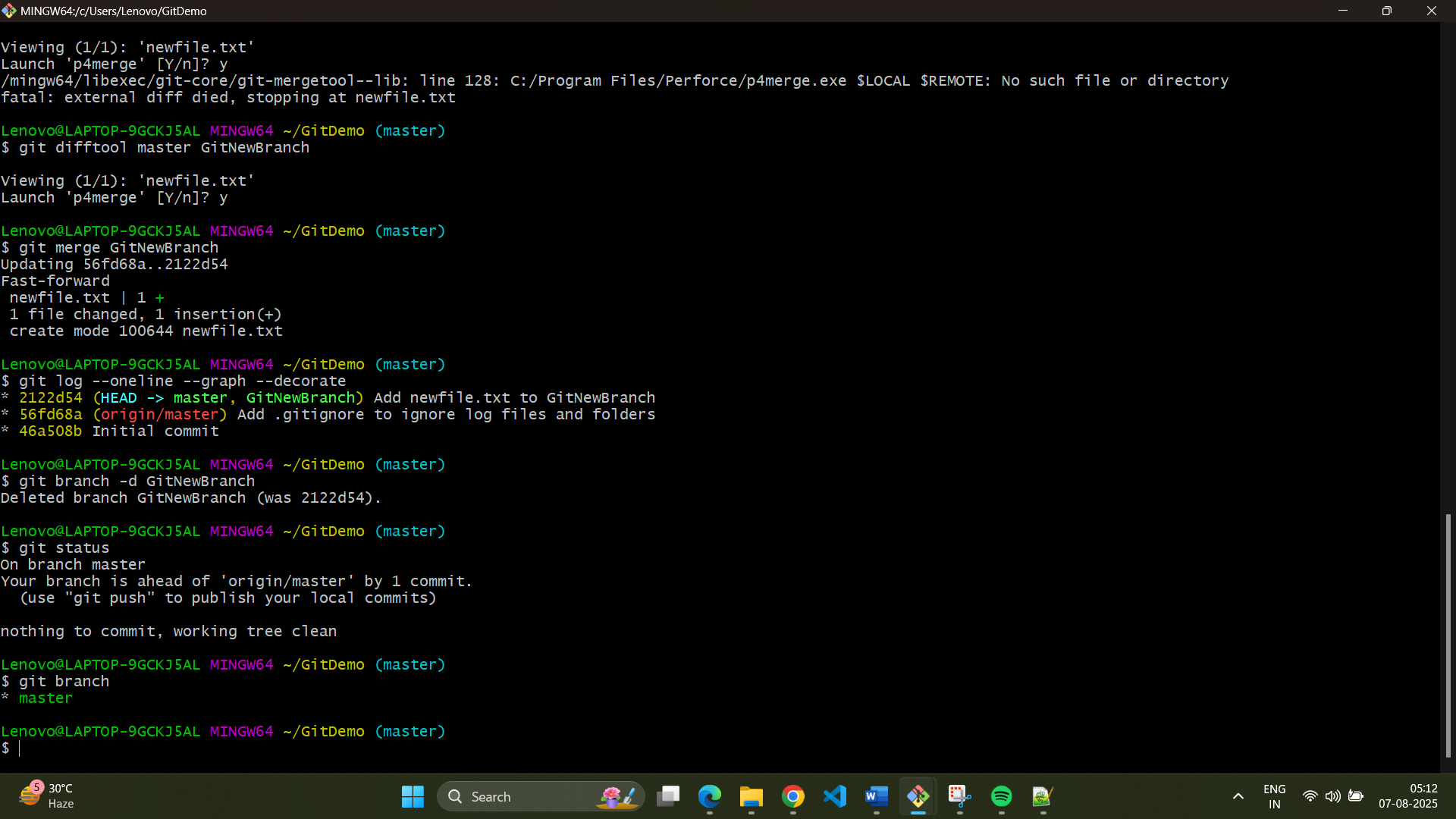
\* master

**Output:**









**Hands on 4**

**Commands:**1. Verify if master is in a clean state

$ git status

On branch master

Your branch is ahead of 'origin/master' by 1 commit.

(use "git push" to publish your local commits)

nothing to commit, working tree clean

2. Create a branch "GitWork" and add a file

$ git checkout -b GitWork

Switched to a new branch 'GitWork'

$ echo "<message>Hello from GitWork branch</message>" > hello.xml

3. Update the content and commit

$ git add hello.xml

warning: LF will be replaced by CRLF in hello.xml.

The file will have its original line endings in your working directory

$ git commit -m "Add hello.xml on GitWork branch"

[GitWork 59794fe] Add hello.xml on GitWork branch

1 file changed, 1 insertion(+)

create mode 100644 hello.xml

4. Switch to master and create the conflict

$ git checkout master

Switched to branch 'master'

Your branch is ahead of 'origin/master' by 1 commit.

(use "git push" to publish your local commits)

$ echo "<message>Hello from master branch</message>" > hello.xml

5. Commit the changes to master

$ git add hello.xml

warning: LF will be replaced by CRLF in hello.xml.

The file will have its original line endings in your working directory

$ git commit -m "Add different hello.xml on master branch"

[master 948e037] Add different hello.xml on master branch

1 file changed, 1 insertion(+)

create mode 100644 hello.xml

6. Observe the log

$ git log --oneline --graph --decorate --all

\* 948e037 (HEAD -> master) Add different hello.xml on master branch

| \* 59794fe (GitWork) Add hello.xml on GitWork branch

|/

\* 2122d54 Add newfile.txt to GitNewBranch

\* 56fd68a (origin/master) Add .gitignore to ignore log files and folders

\* 46a508b Initial commit

7. Check the differences with Git diff tool

$ git diff master GitWork

**diff --git a/hello.xml b/hello.xml**

**index 82868a7..0bf4c19 100644**

**--- a/hello.xml**

**+++ b/hello.xml**

@@ -1 +1 @@

-<message>Hello from master branch</message>

+<message>Hello from GitWork branch</message>

8. Use P4Merge tool to list out all the differences

$ git mergetool

Merging:

hello.xml

Normal merge conflict for 'hello.xml':

{local}: created file

{remote}: created file

error: invalid path './hello\_BASE\_1595.xml'

9. Commit the changes to master, once done with conflict

$ git add hello.xml

$ git commit -m "Resolved merge conflict and merged GitWork"

[master 5a980f6] Resolved merge conflict and merged GitWork

10. Observe the git status and add backup file to the .gitignore

$ git status

On branch master

Your branch is ahead of 'origin/master' by 4 commits.

(use "git push" to publish your local commits)

Untracked files:

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

hello.xml.orig

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

$ echo "\*.orig" >> .gitignore

11. Delete the branch after merging

$ git add .gitignore

warning: LF will be replaced by CRLF in .gitignore.

The file will have its original line endings in your working directory

$ git commit -m "Add .orig files to gitignore"

[master 2066aa1] Add .orig files to gitignore

1 file changed, 1 insertion(+), 1 deletion(-)

12. List out all the available branches

$ git branch -a

GitWork

\* master

remotes/origin/master

13. Delete the branch

$ git branch -d GitWork

Deleted branch GitWork (was 32edfe0).

14. Observe the log

$ git log --oneline --graph --decorate

\* 2066aa1 (HEAD -> master) Add .orig files to gitignore

\* 5a980f6 Resolved merge conflict and merged GitWork

|\

| \* 32edfe0 Add hello.xml on GitWork branch

\* | 9bf4b38 Add hello.xml on master branch

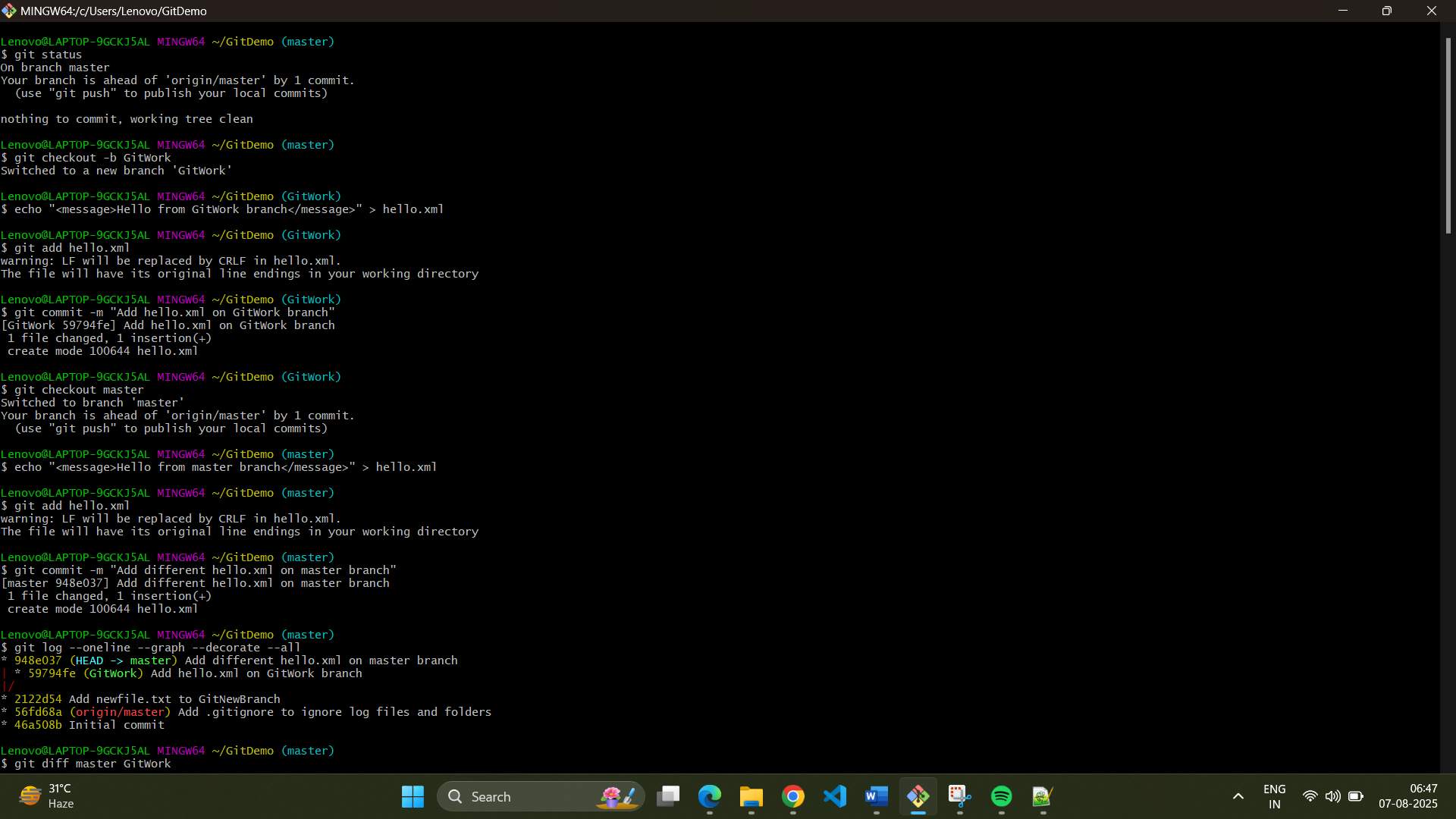
|/

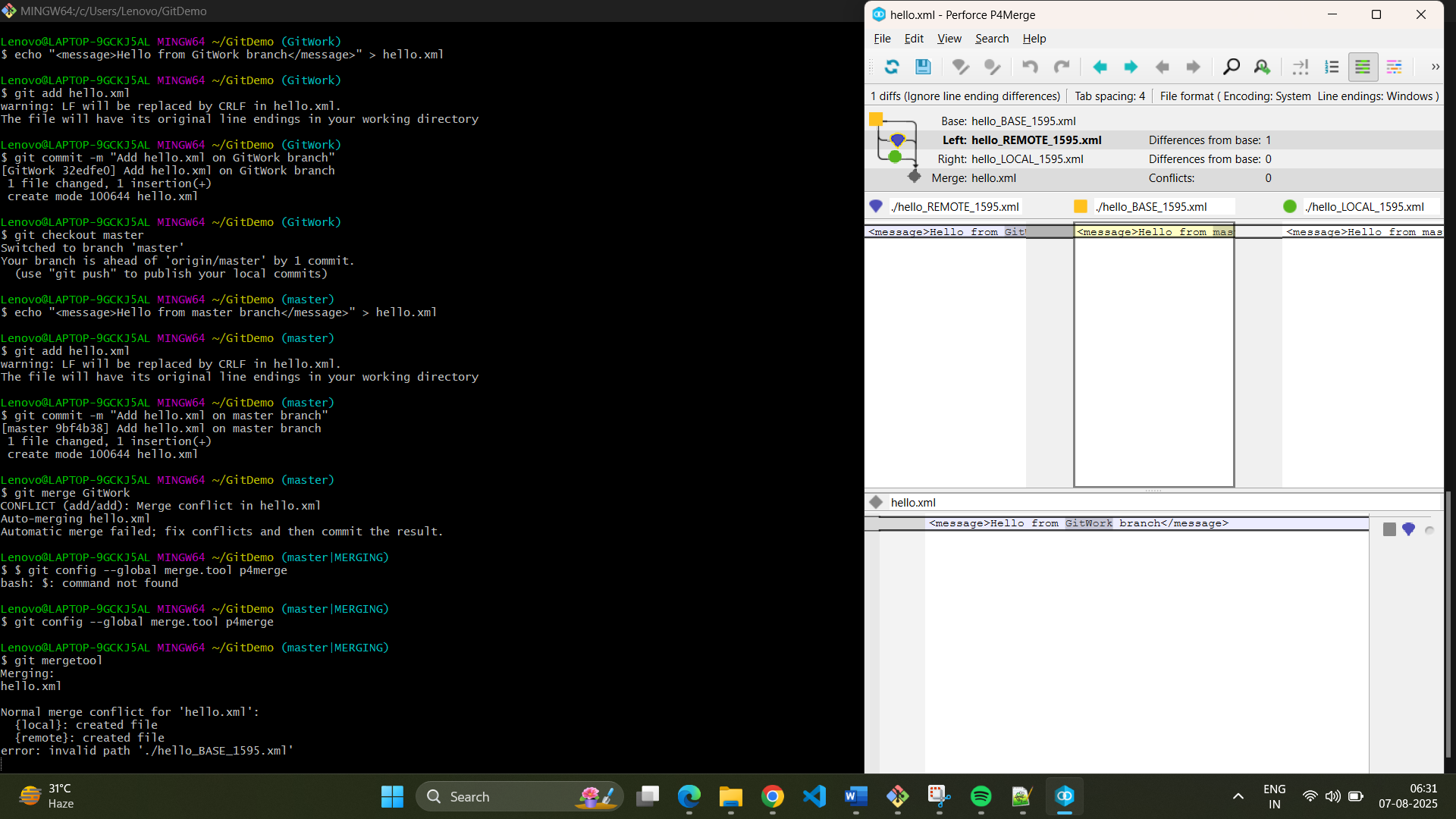
\* 2122d54 Add newfile.txt to GitNewBranch

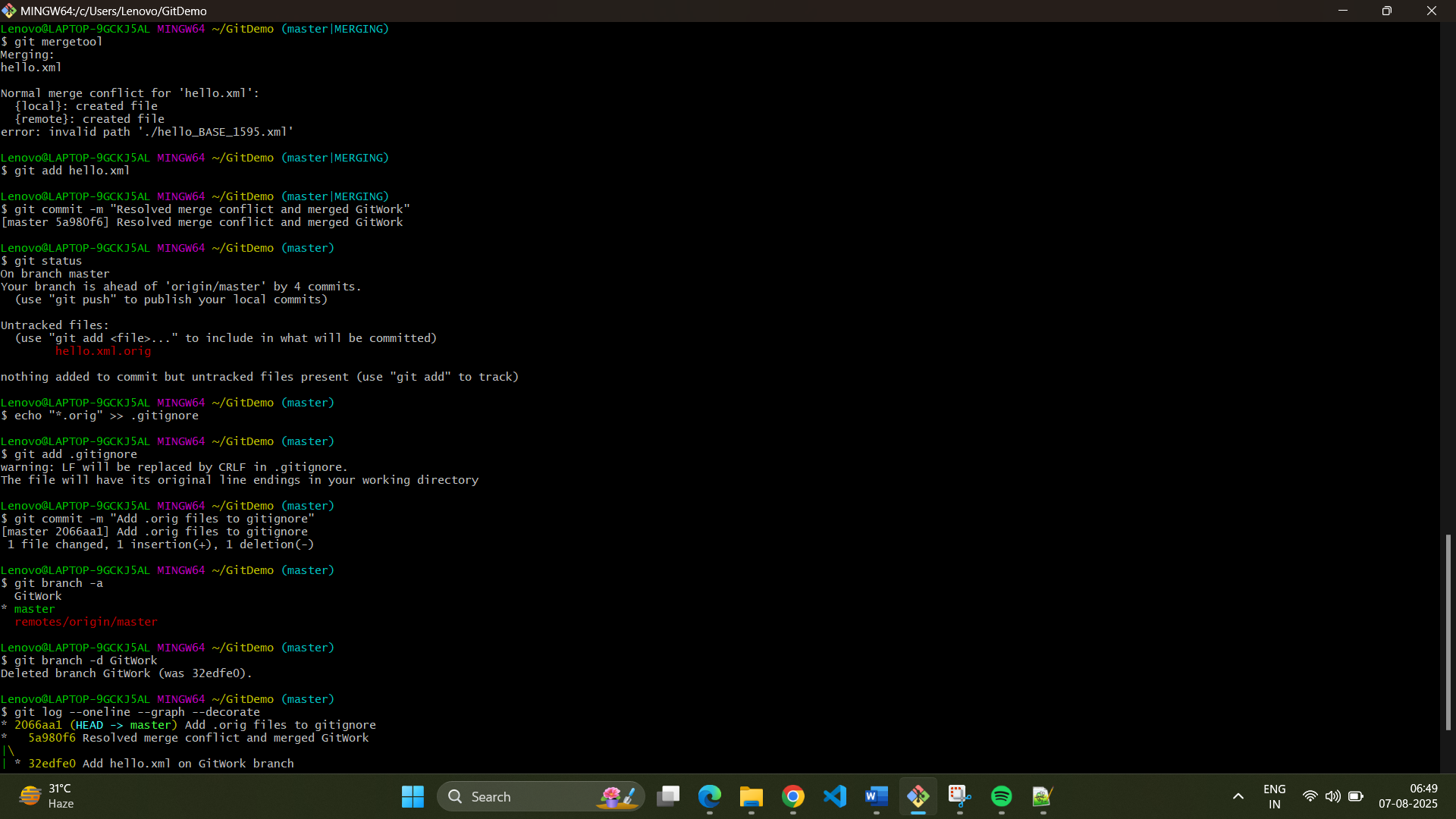
\* 56fd68a (origin/master) Add .gitignore to ignore log files and folders

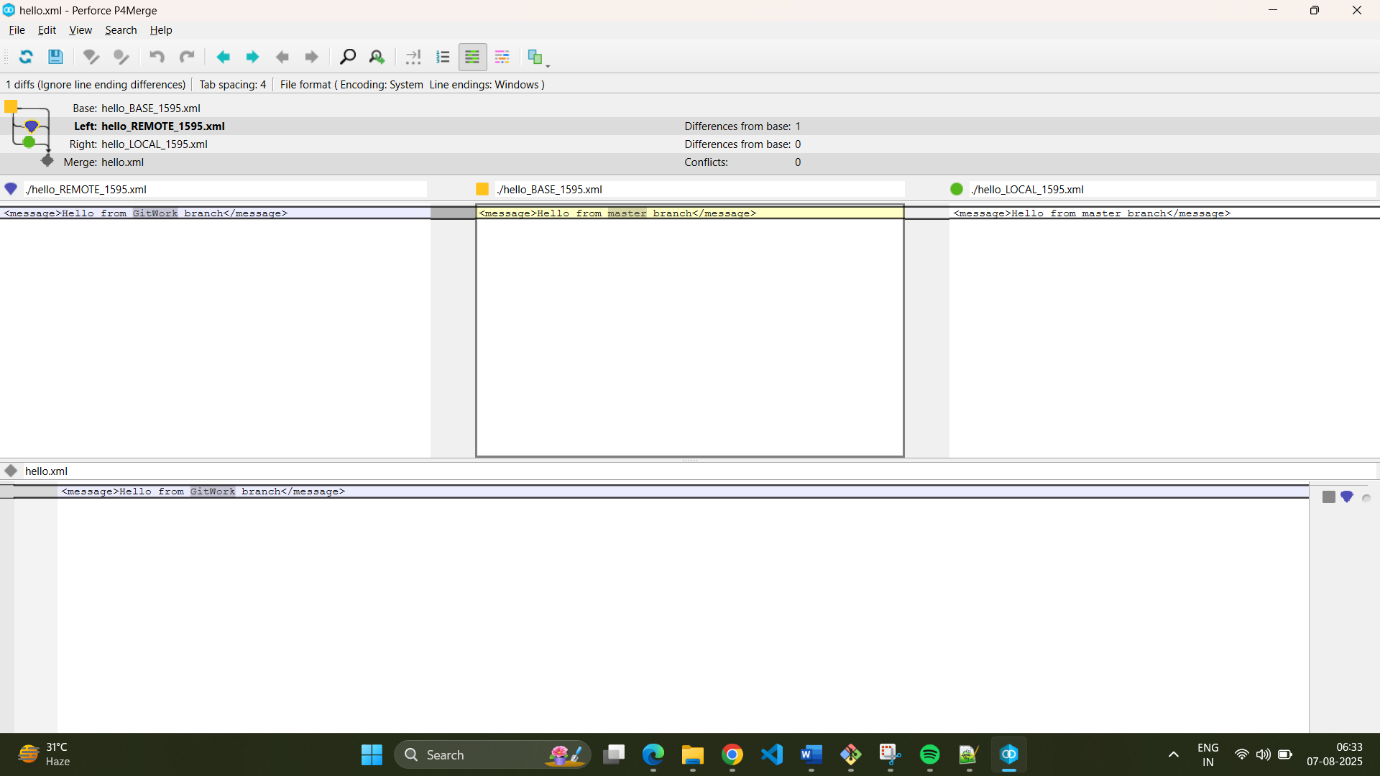
\* 46a508b Initial commit

**Output:**









**Hands on 5**

**Commands:**

1. Verify if master is in a clean state

$ git status

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

2. List out all the available branches

$ git branch -a

\* master

remotes/origin/master

3. Pull the remote git repository to the master

$ git pull origin master

From https://gitlab.com/Tarun-Rawat/gitdemo

\* branch master -> FETCH\_HEAD

Already up to date.

4. Push the changes to the remote repository

$ git push origin master

Everything up-to-date

Lenovo@LAPTOP-9GCKJ5AL MINGW64 ~/GitDemo (master)

5. Ignore the backup file and commit the change

$ git status

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

$ git log --oneline --graph --decorate --all

\* 5d535b7 (HEAD -> master, origin/master) Add .orig files to gitignore

\* 2066aa1 Add .orig files to gitignore

\* 5a980f6 Resolved merge conflict and merged GitWork

|\

| \* 32edfe0 Add hello.xml on GitWork branch

\* | 9bf4b38 Add hello.xml on master branch

|/

\* 2122d54 Add newfile.txt to GitNewBranch

\* 56fd68a Add .gitignore to ignore log files and folders

\* 46a508b Initial commit

**Output:**

