WEEK 8 ASSIGNMENT

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:

COMMANDS:

1.git config --global user.name "ARUN KUMAR"

2.git config --global user.email "[your.email@example.com](mailto:your.email@example.com)

echo "# GitDemo" >> README.md

3.git init

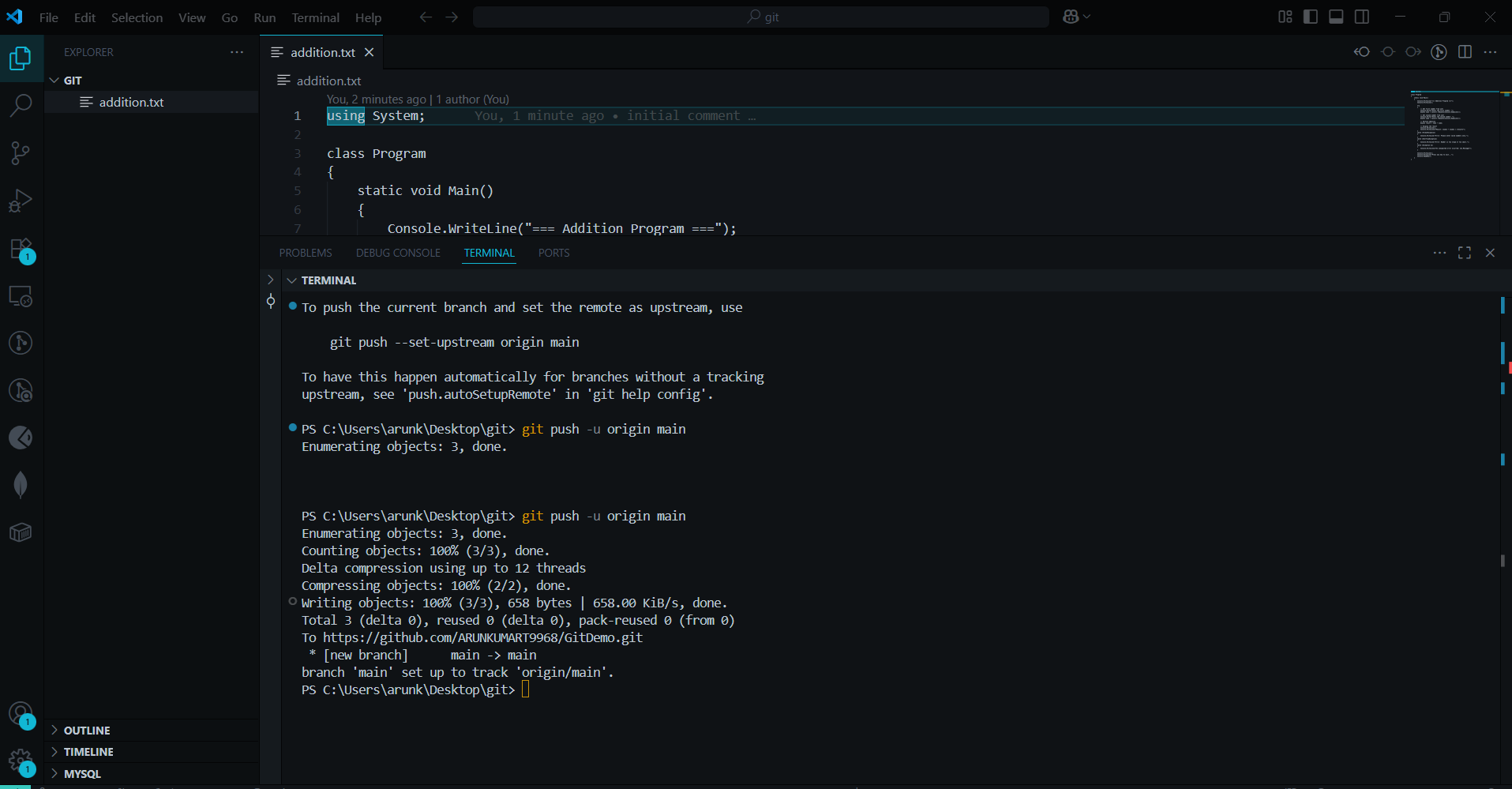
4.git add README.md

5.git commit -m "first commit"

6.git branch -M main

7.git remote add origin https://github.com/ARUNKUMART9968/GitDemo.git

8.git push -u origin main

  
  
  
  
  
  
  
  
  
2.

Explain git ignore

Explain how to ignore unwanted files using git ignore

OUTPUT:

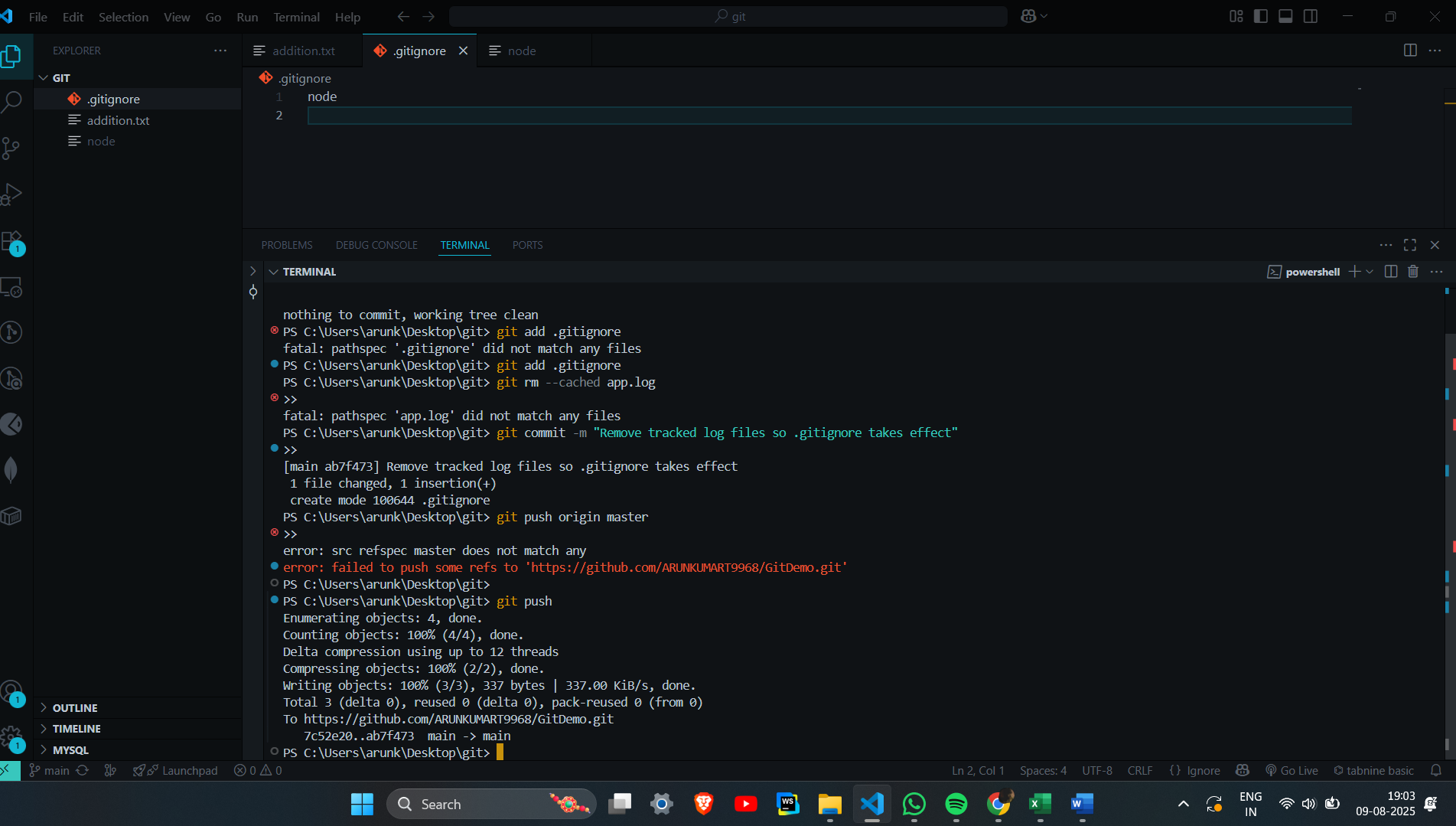
git status

git add .gitignore

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

git commit -m "Remove tracked log files so .gitignore takes effect"

git push



3. Explain branching and merging

Explain about creating a branch request in GitLab

Explain about creating a merge request in GitLab|

OUTPUT:

COMMANDS:

git checkout -b GitNewBranch

git checkout master

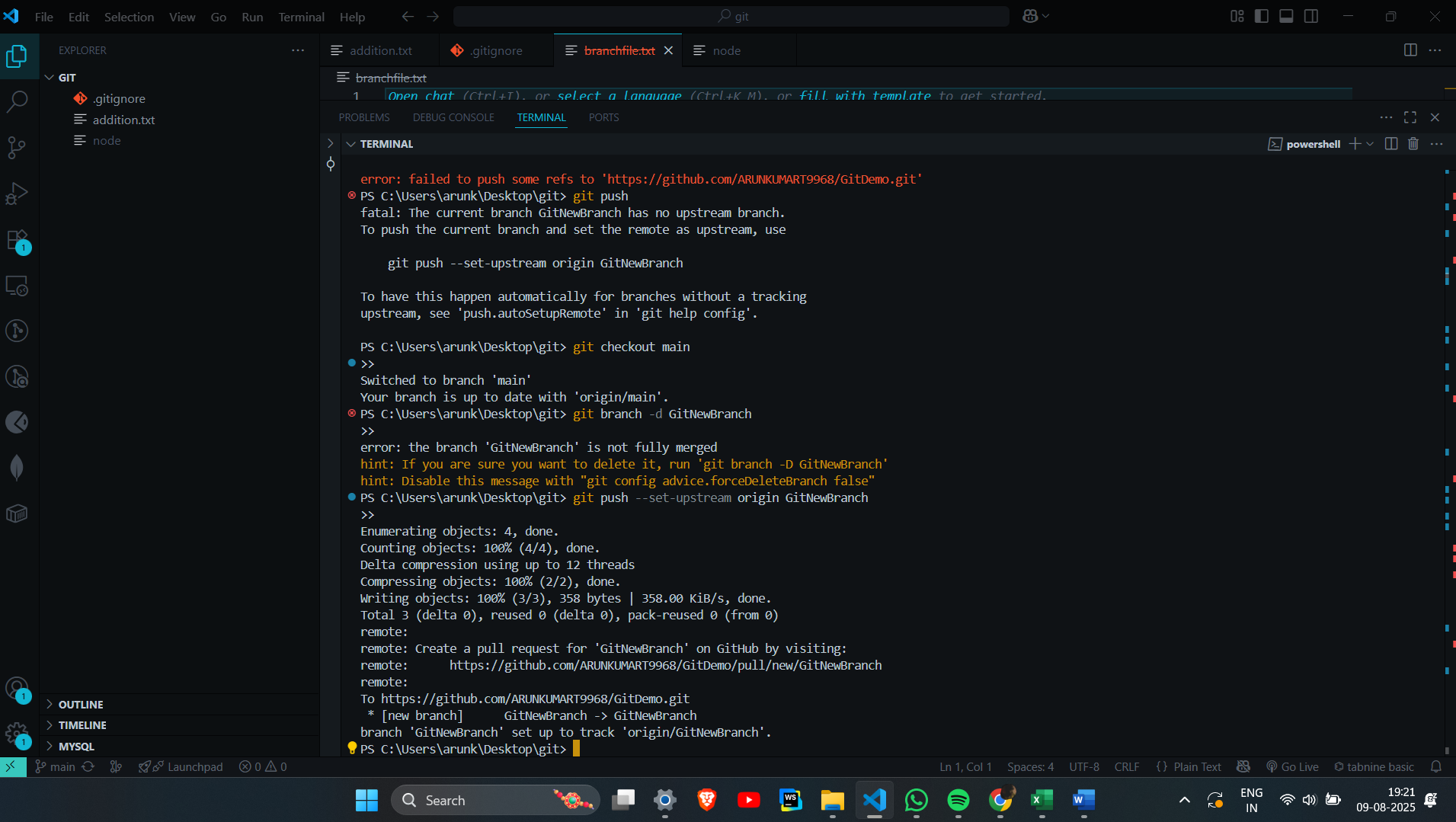
git fetch origin

git diff master GitNewBranch

git checkout main

git branch -d GitNewBranch

git push origin --delete GitNewBranch

  
  
  
4.

Explain how to resolve the conflict during merge.  
  
OUTPUT:

git checkout master

git pull origin master

git checkout -b GitWork

@"

<!-- content from GitWork -->

<hello>Branch content A</hello>

"@ | Out-File hello.xml -Encoding utf8

git add hello.xml

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

git checkout master

@"

<!-- content from master -->

<hello>Master content B</hello>

"@ | Out-File hello.xml -Encoding utf8

git add hello.xml

git commit -m "Add hello.xml on master with different content"

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

git merge GitWork

git add hello.xml

git commit -m "Merge GitWork into master — resolved conflict"

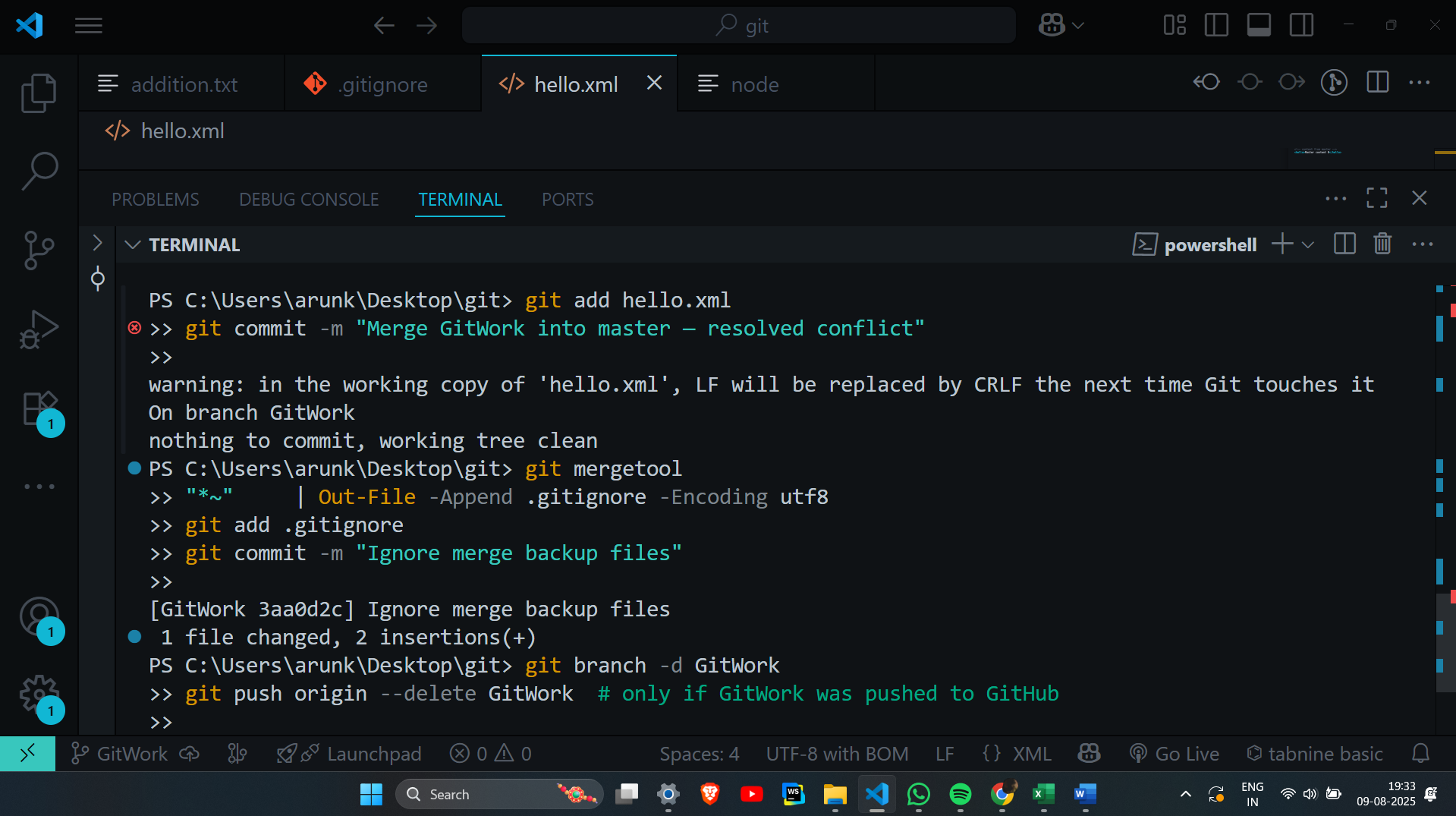
git mergetool

"\*.orig" | Out-File -Append .gitignore -Encoding utf8

"\*~" | Out-File -Append .gitignore -Encoding utf8

git add .gitignore

git commit -m "Ignore merge backup files"



5.Explain how to resolve the conflict during merge.

# 1. Navigate to your repo

cd "C:\path\to\GitDemo"

# 2. Verify you're on master and working tree is clean

git checkout master

git status # should show "nothing to commit, working tree clean"

# 3. List local and remote branches

git branch -a

# 4. Fetch and pull the latest from remote

git fetch --all

git pull origin master

# 5. If you have local commits not yet pushed, push them

git push origin master

# If upstream not set:

# git push --set-upstream origin master

# 6. (Optional) Push all branches to remote

git push --all origin

# 7. Prune stale remote-tracking branches (cleanup)

git fetch --prune

# 8. Verify remote and local are in sync

git log --oneline origin/master..master

# If this shows nothing, master is in sync with origin/master

