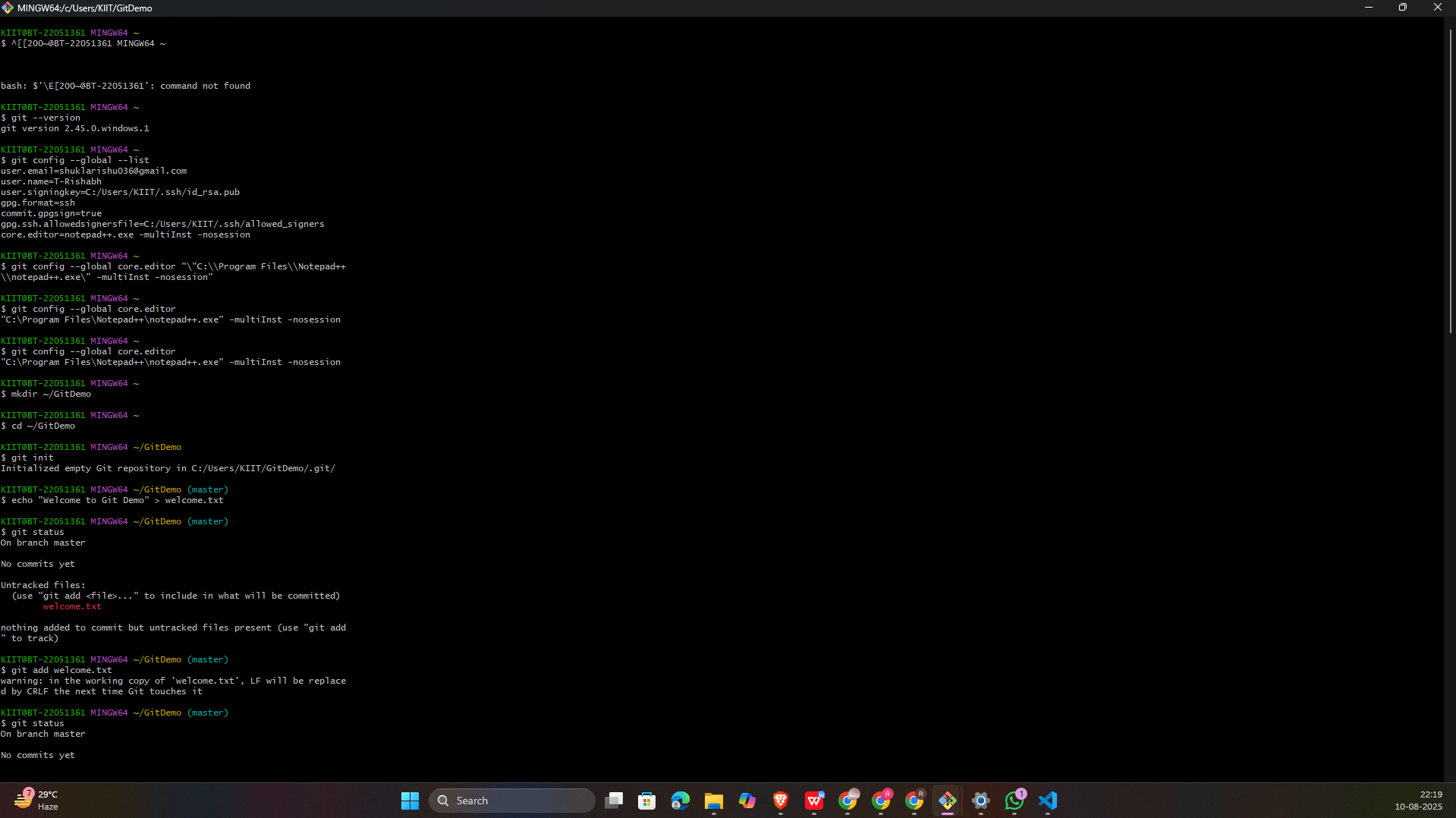
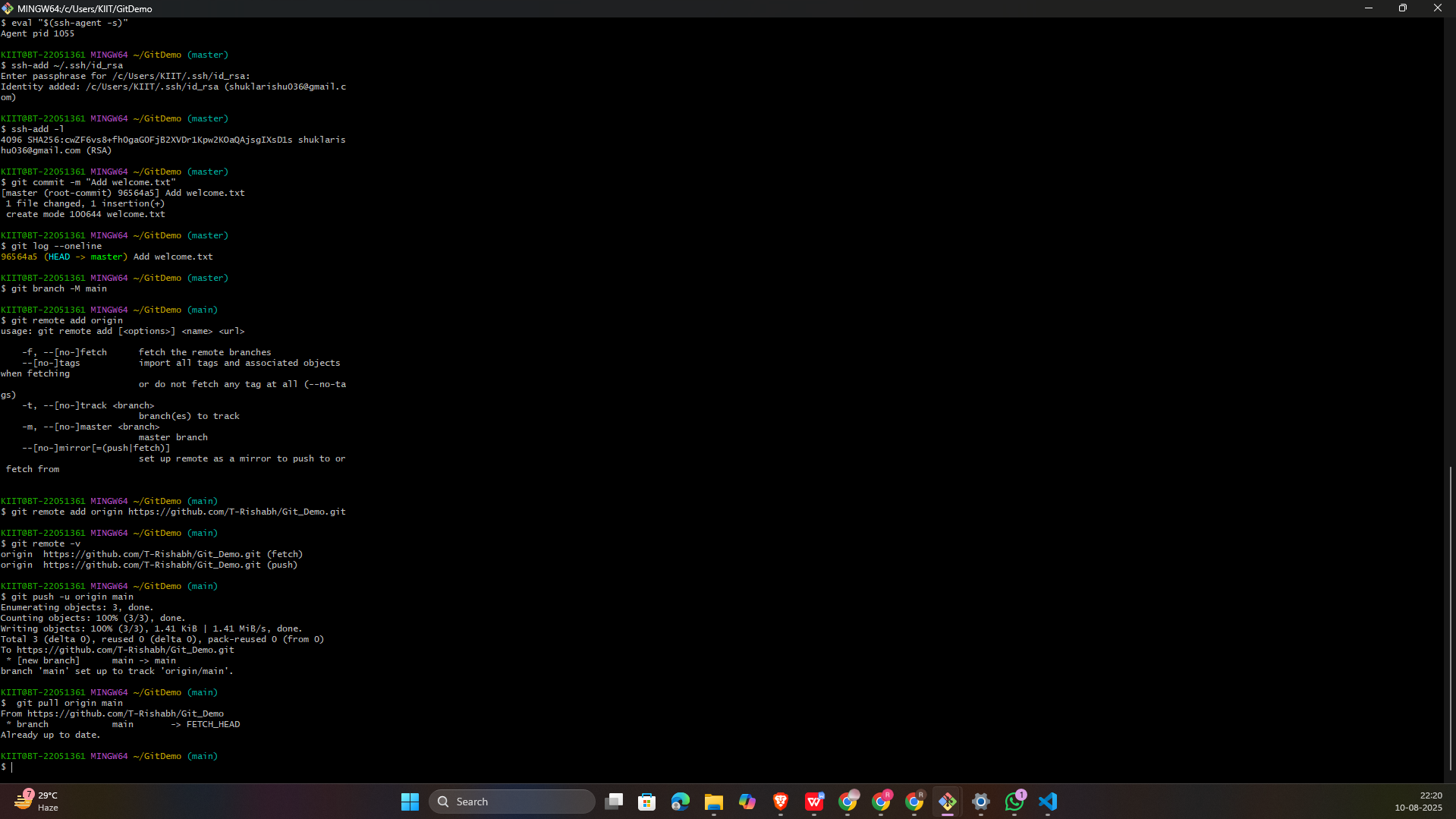
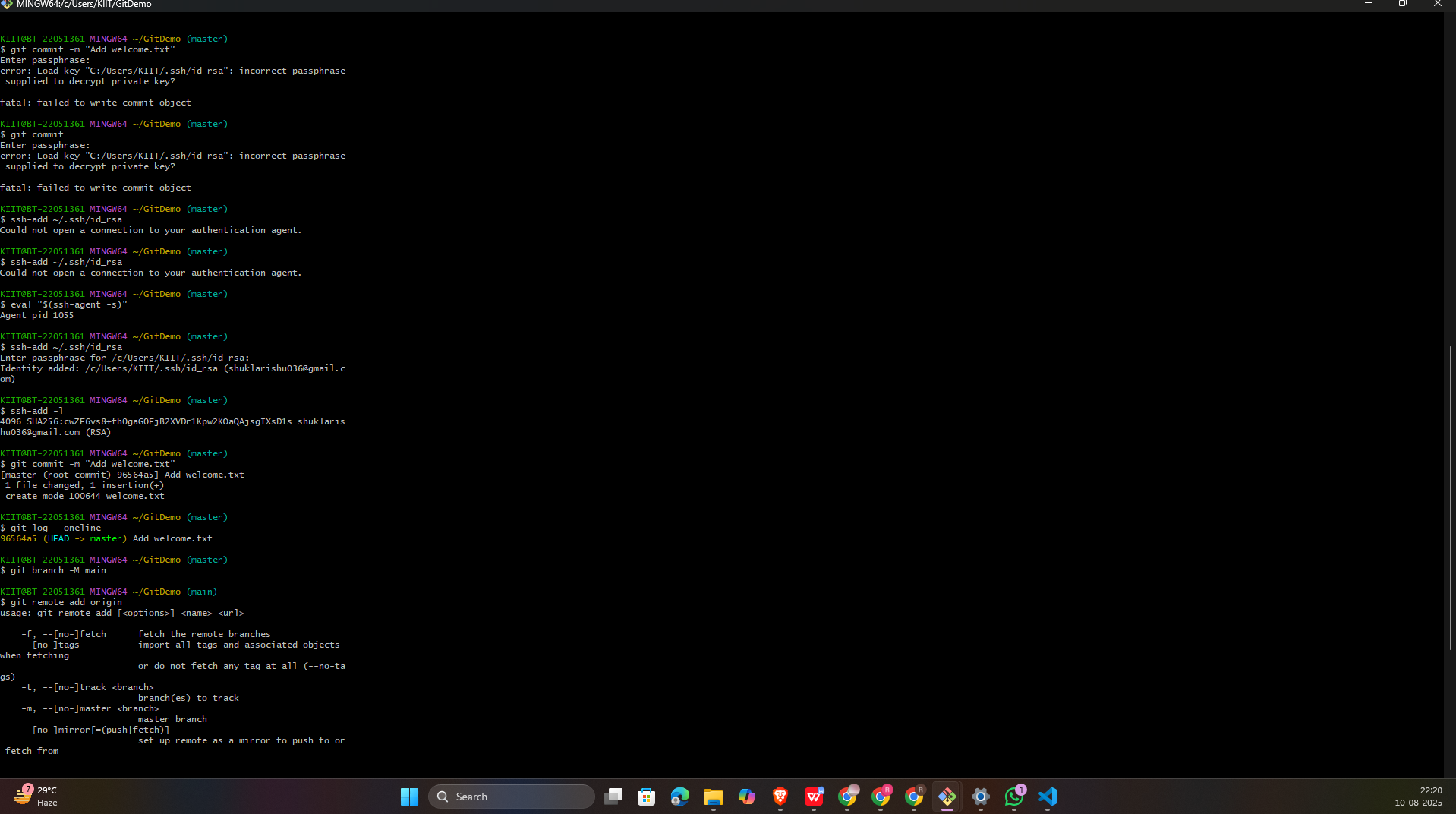
**Git-HOL-HandsOn**

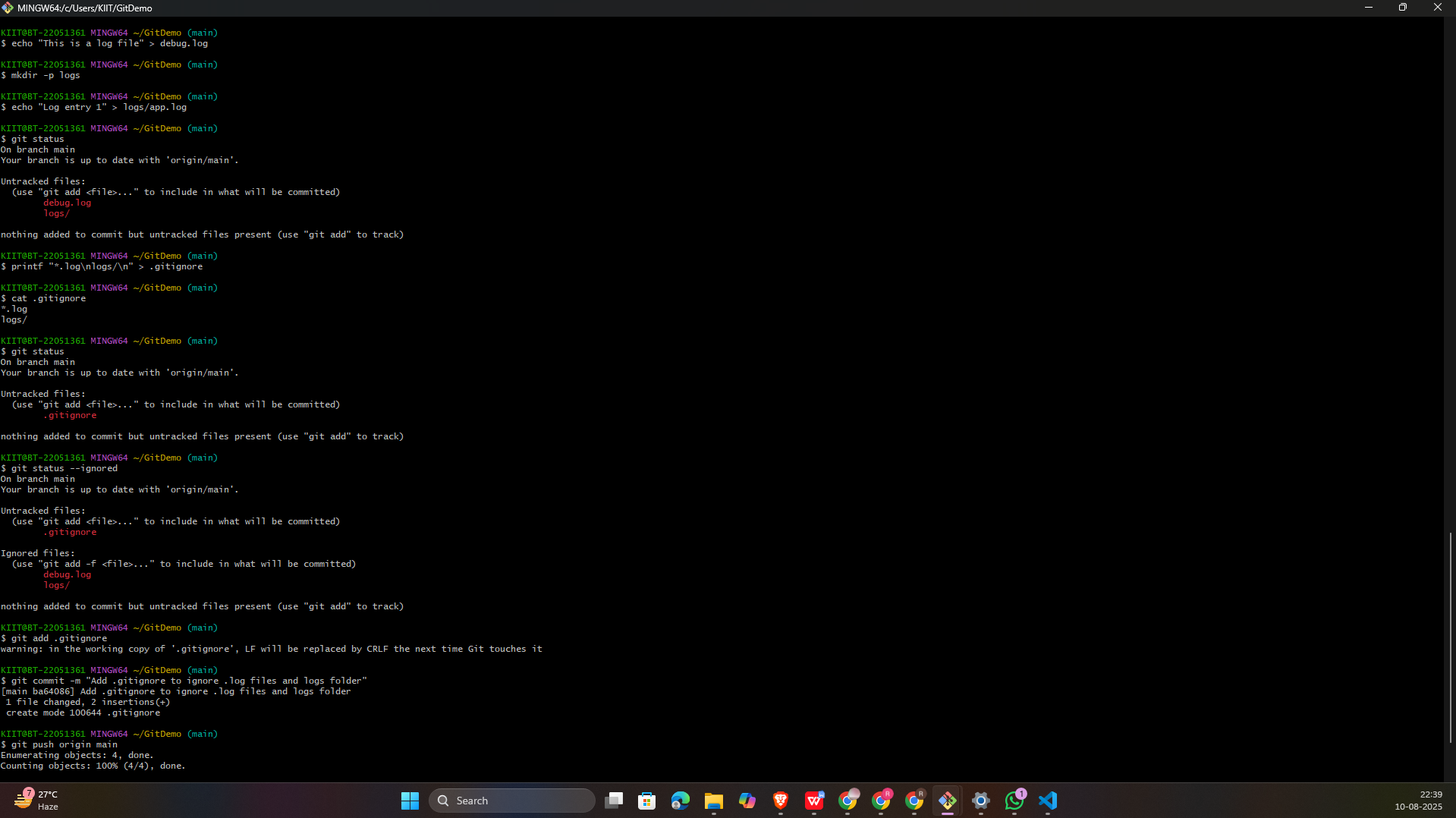
1.Git-HOL:

**Output:** 



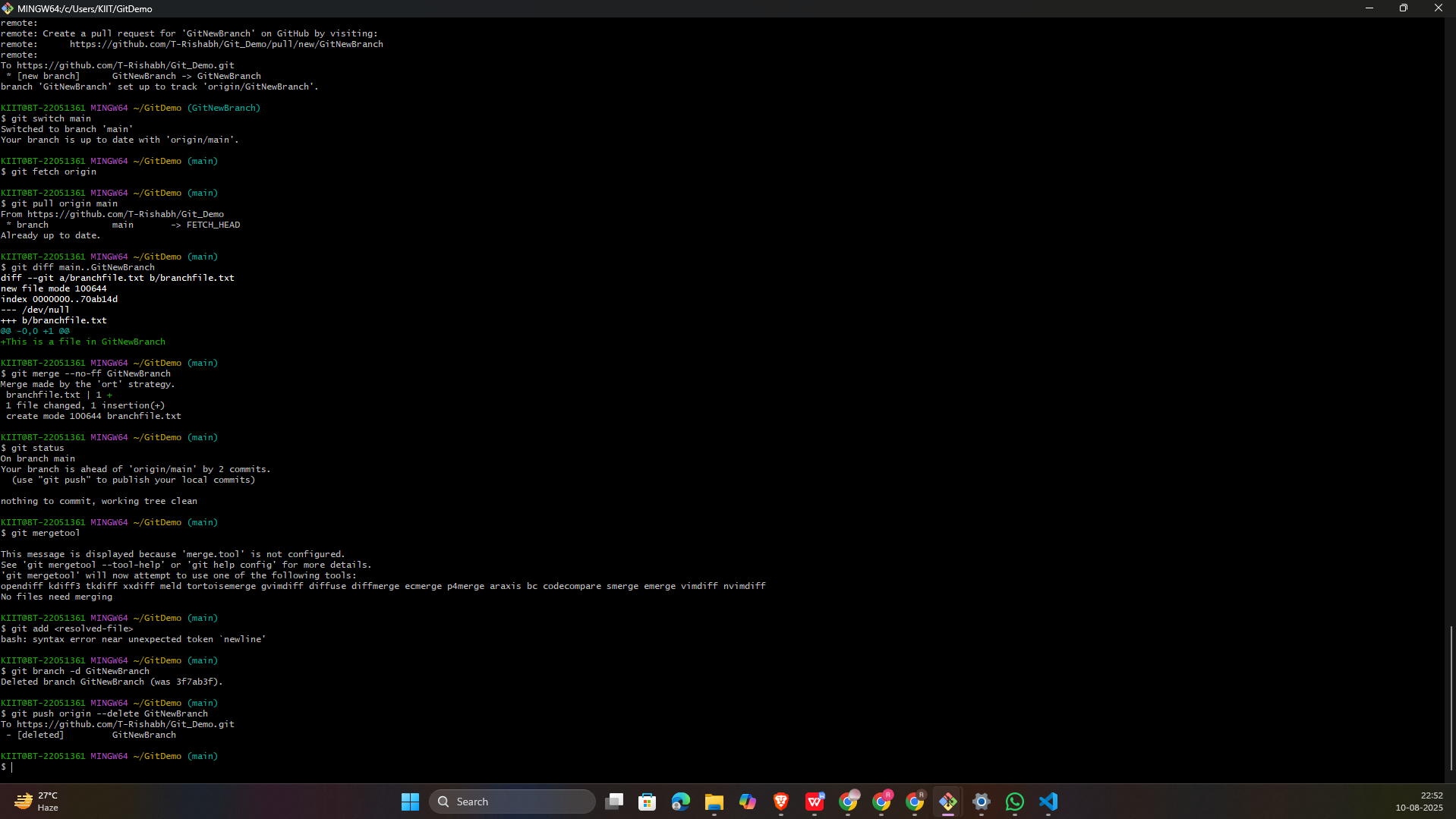
2.Git-HOL:

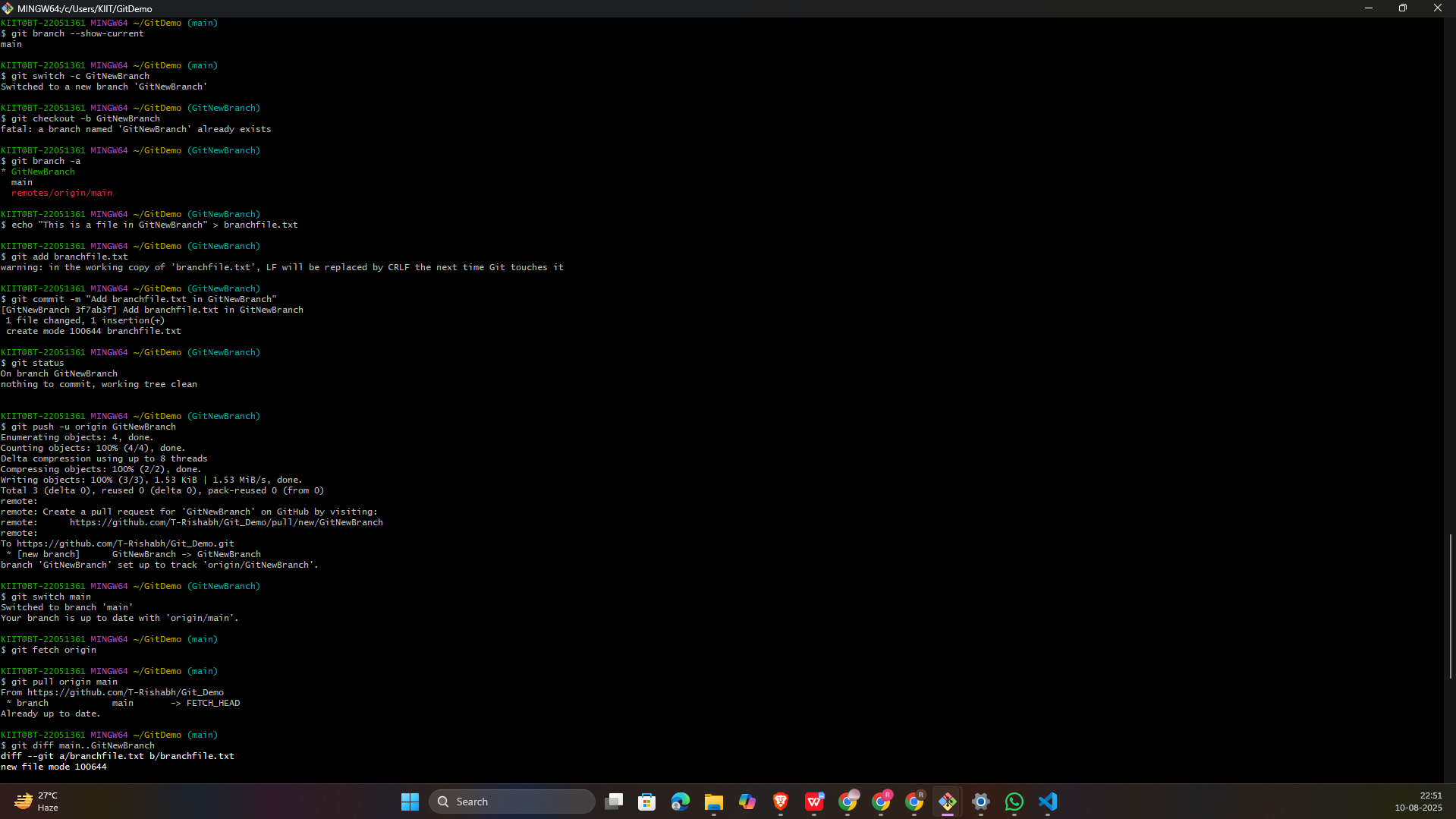
**Output:**



3.Git-HOL:

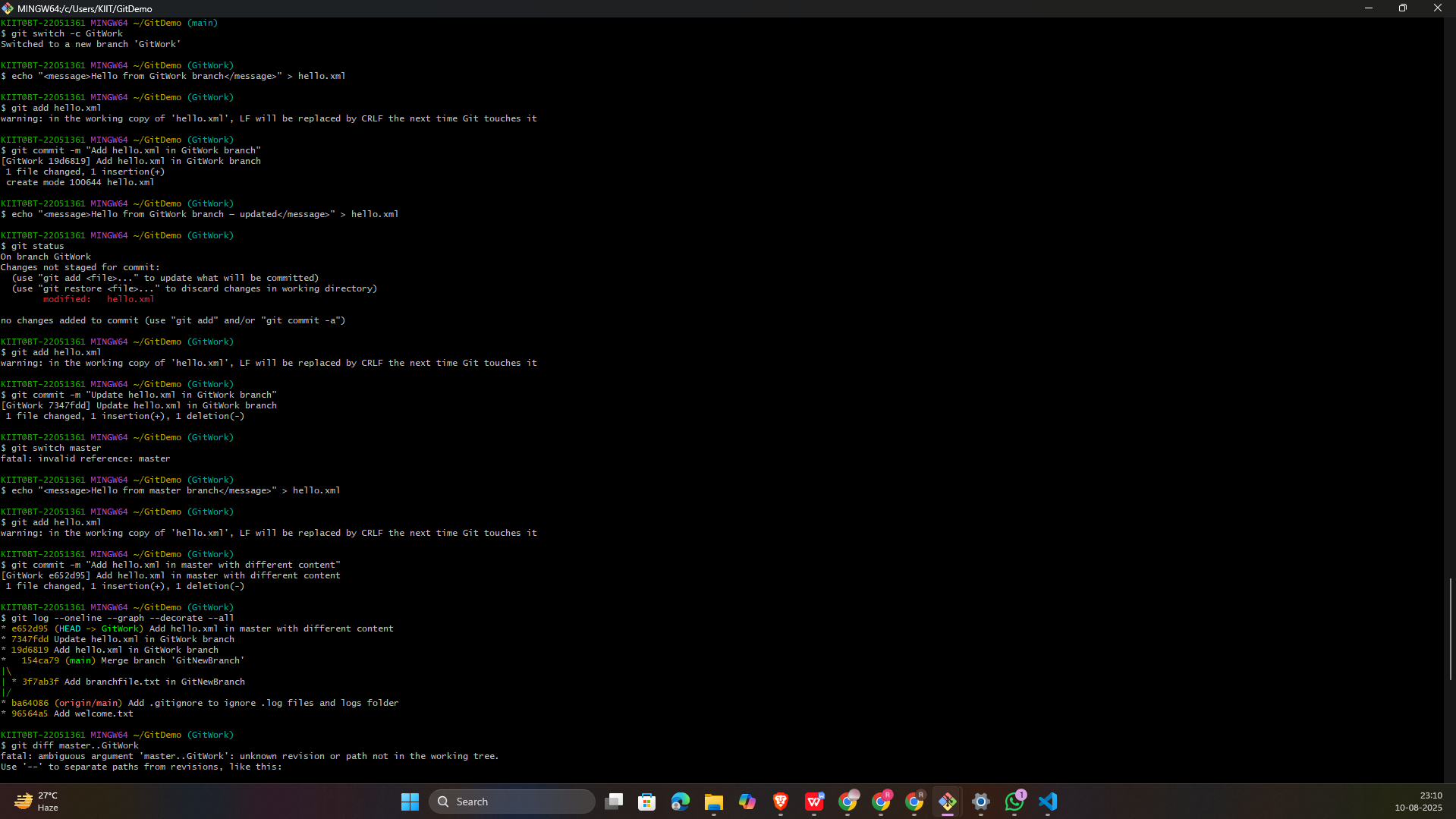
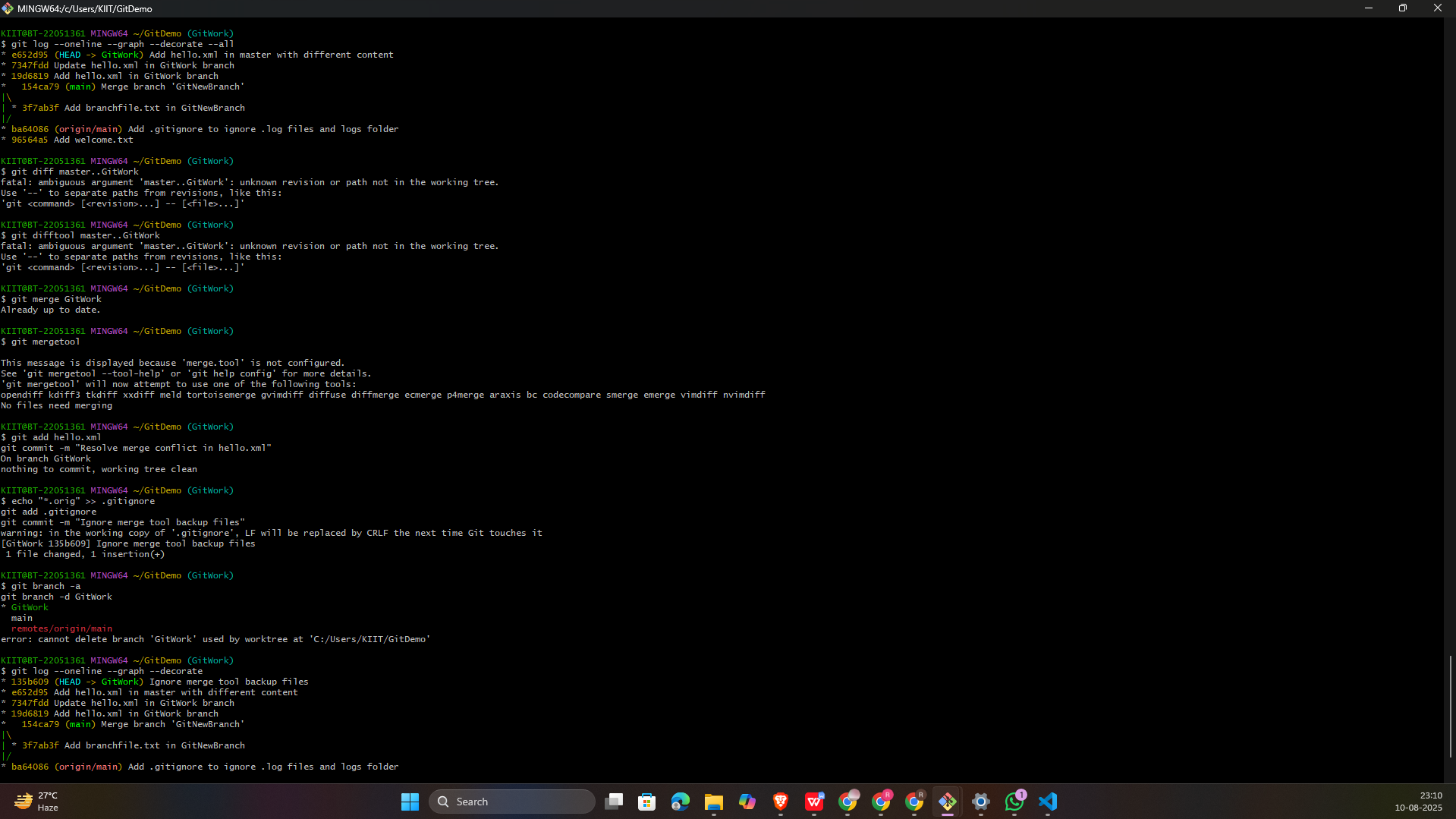
**Output:**





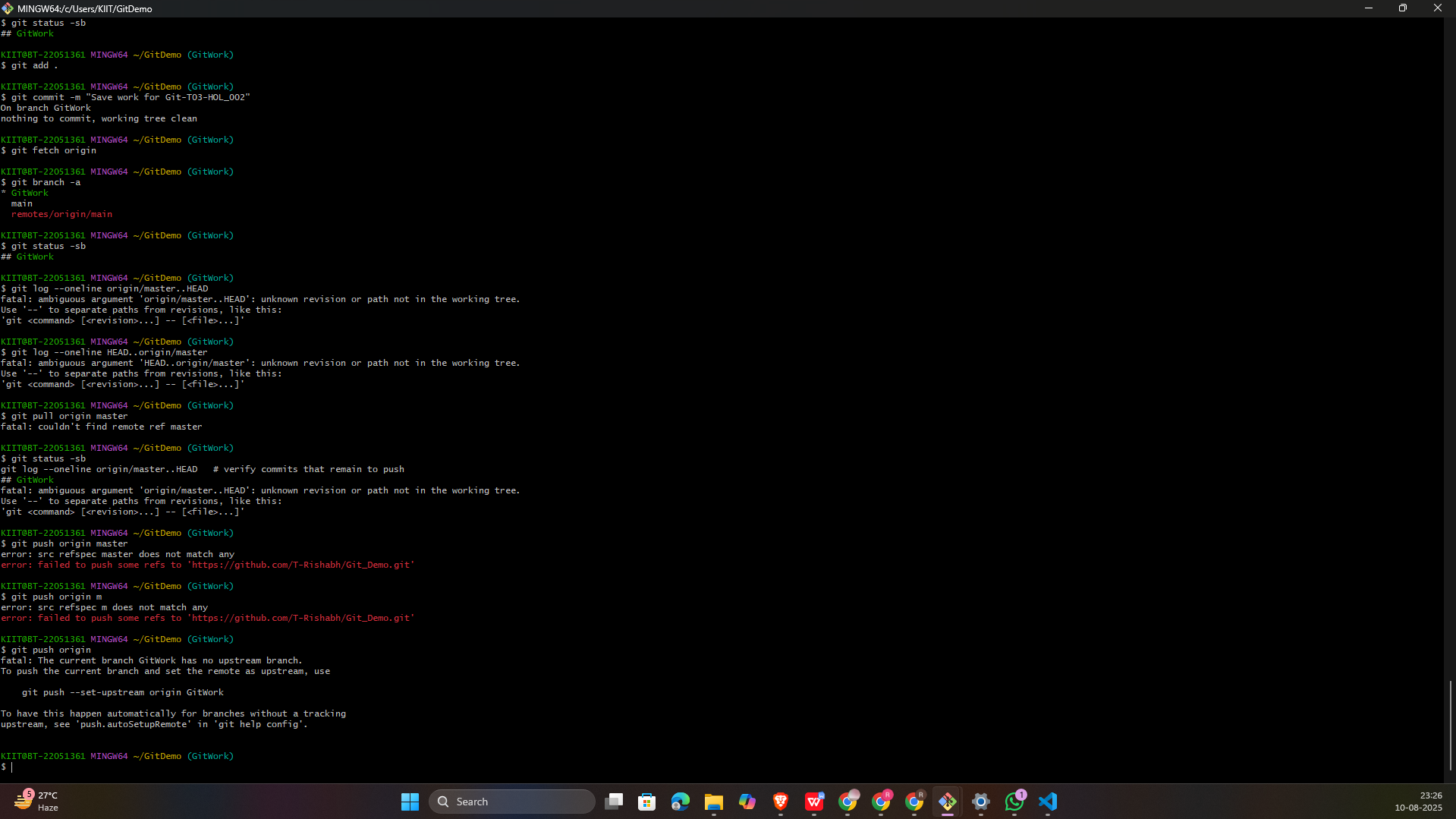
4.Git-HOL:

**Output:**



5.Git-HOL:

**Output:**



**Commands Used By me:-**

git --version

git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

git config --list

If Notepad++ not recognized: Add C:\Program Files\Notepad++ to PATH

alias np='notepad++'

git config --global core.editor "notepad++ -multiInst -nosession"

git config --global -e

mkdir GitDemo && cd GitDemo

git init

ls -a

echo "Welcome to Git Hands-On Lab" > welcome.txt

ls

cat welcome.txt

git status

git add welcome.txt

git commit

git status

git remote add origin https://gitlab.com/username/GitDemo.git

git pull origin master --allow-unrelated-histories

git push origin master

echo "Some logs" > error.log

mkdir logs && echo "Log content" > logs/app.log

echo "\*.log" > .gitignore && echo "logs/" >> .gitignore

git status

git add .gitignore

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

git status

git checkout -b GitWork

echo "Hello from branch" > hello.xml

git add hello.xml && git commit -m "Add hello.xml in branch"

git checkout master

echo "Hello from master" > hello.xml

git add hello.xml && git commit -m "Add hello.xml in master"

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

git diff GitWork

git merge GitWork

Edit hello.xml to resolve conflict

git add hello.xml && git commit -m "Resolve merge conflict"

echo "\*.\*~" >> .gitignore && git add .gitignore && git commit -m "Ignore backup

files"

git branch -d GitWork

git log --oneline --graph --decorate

git status

git branch -a

git pull origin master

git push origin master

Check GitLab/GitHub repo