1. **GIT-HOL**

git –version

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

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

git config –list

notepad++

git config --global core.editor "'C:/Program Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noPlugin"

git config --global -e

mkdir GitDemo

cd GitDemo

git init

ls -a

echo "Welcome to Git!" > welcome.txt

cat welcome.txt

git add welcome.txt

git commit

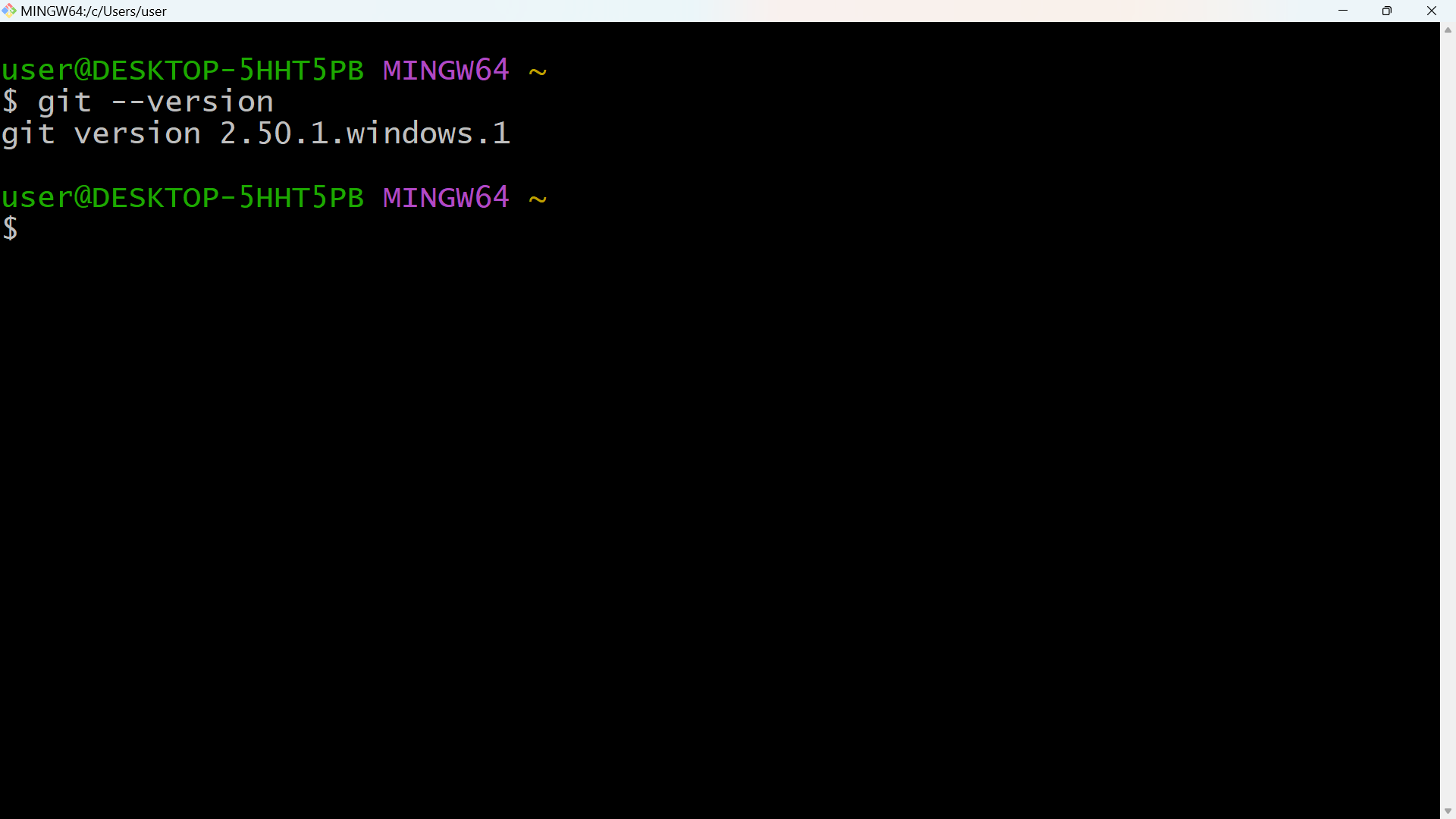
git remote add origin <https://gitlab.com/my-username/GitDemo.git>

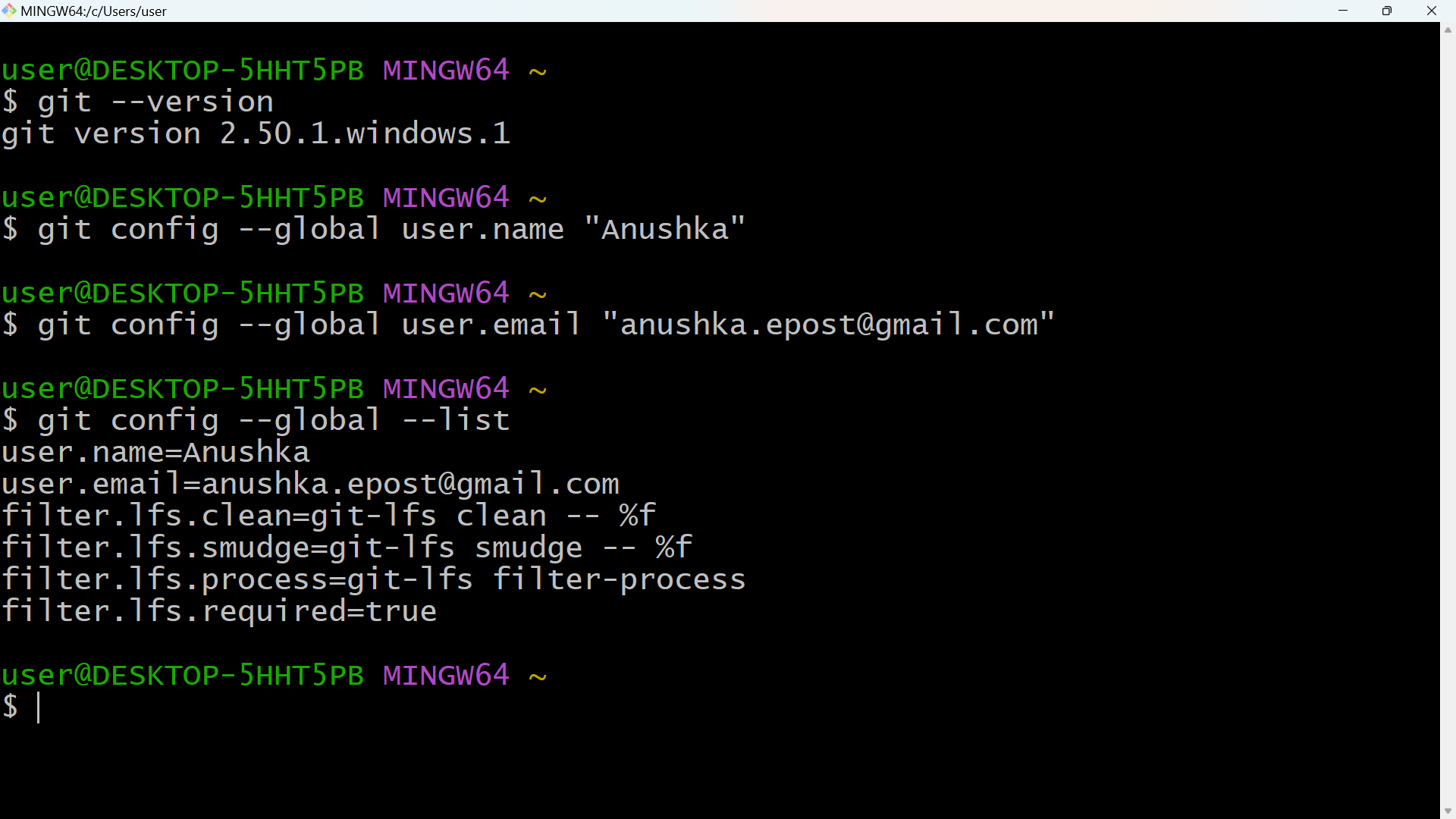
git remote -v

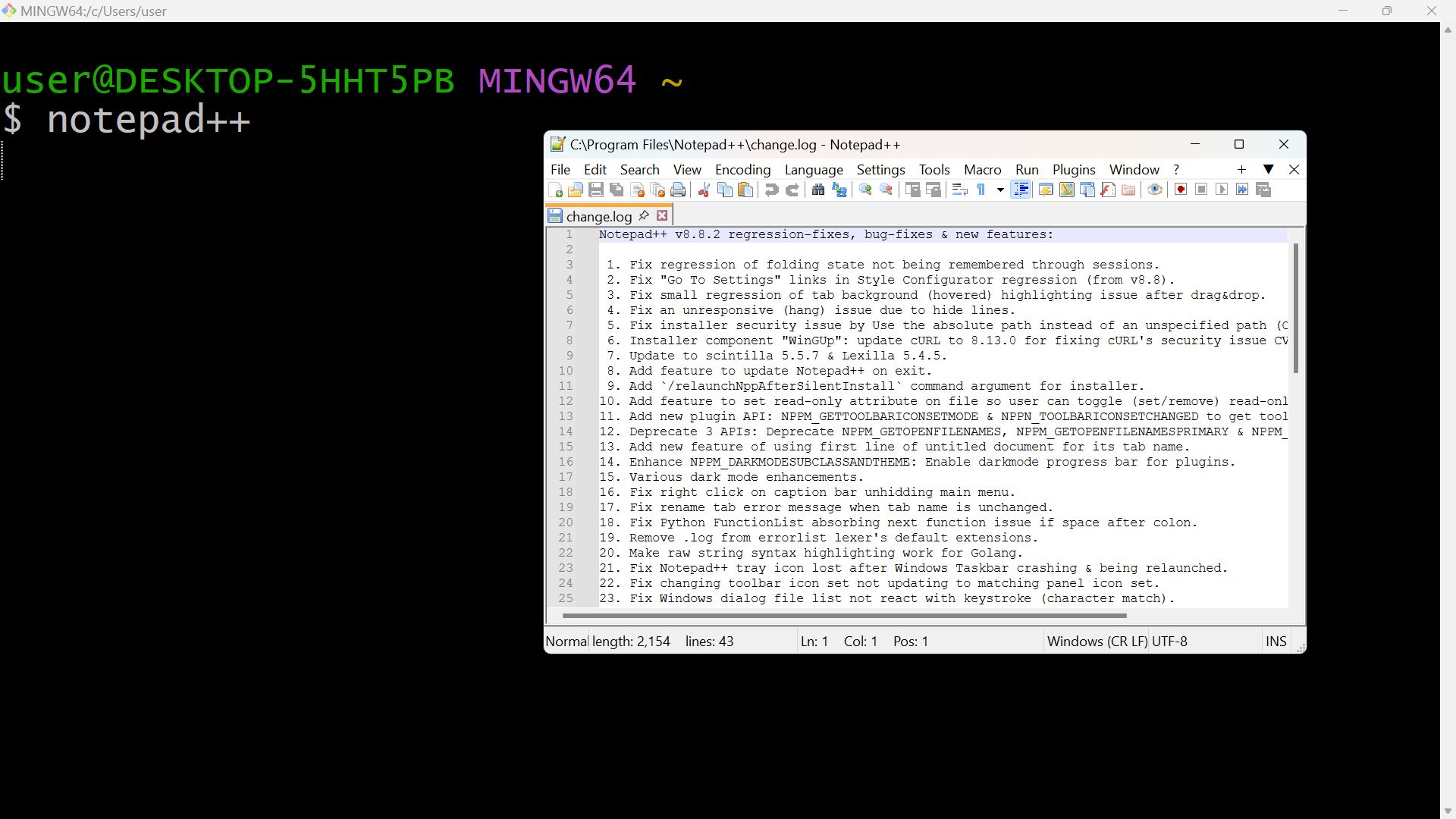
git pull origin master --allow-unrelated-histories

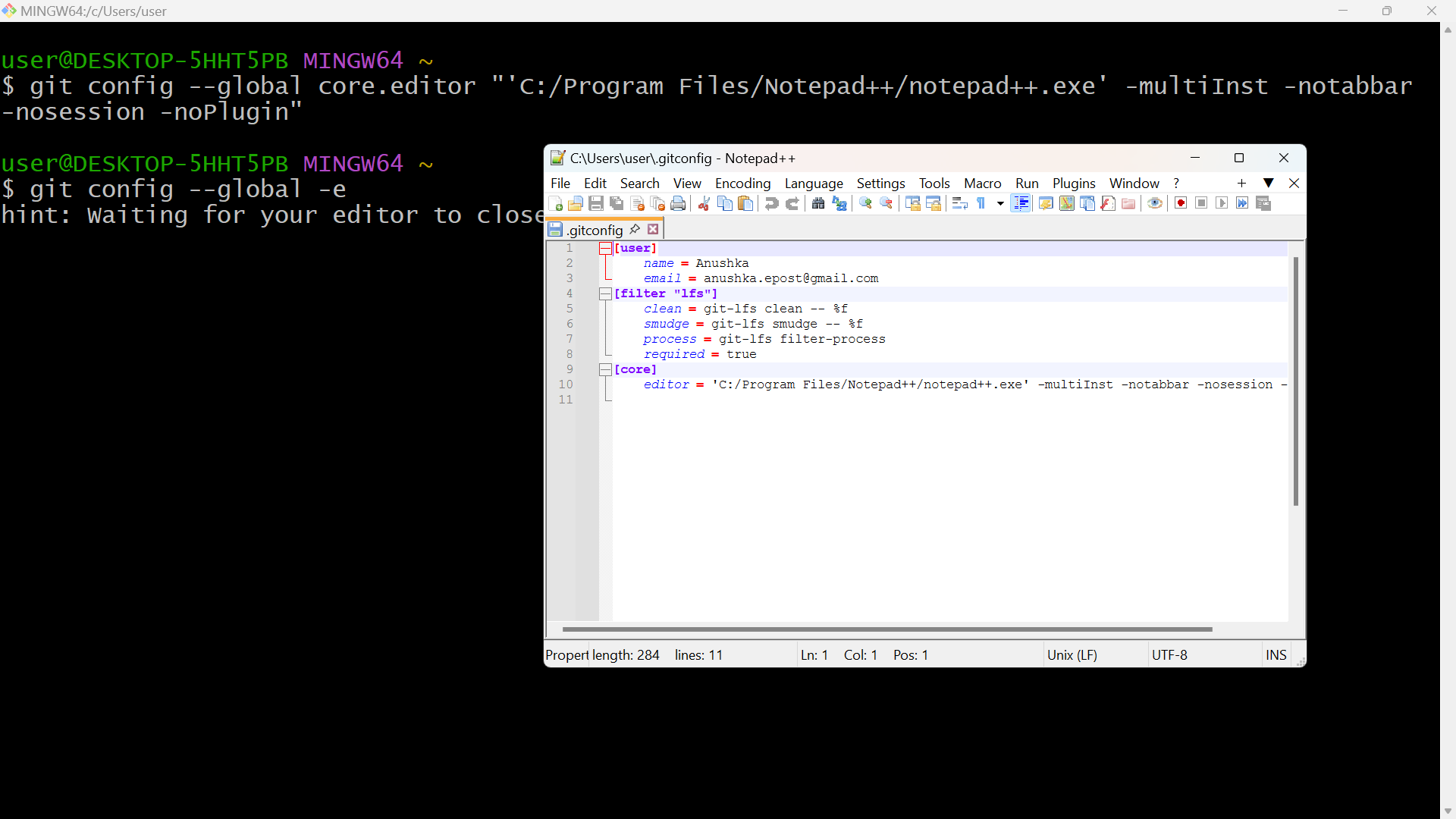
git push -u origin master

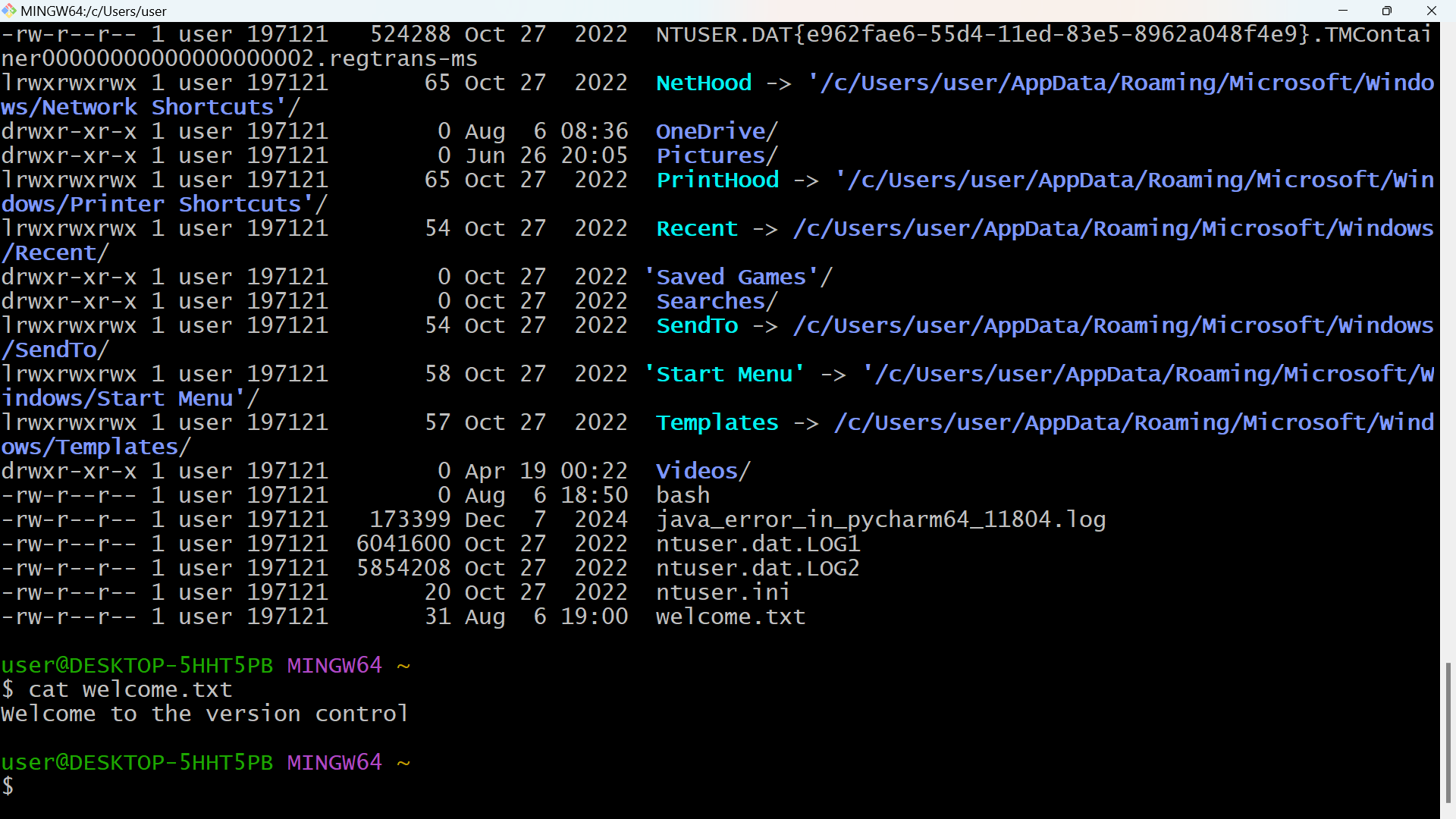
**SNAPSHOTS**



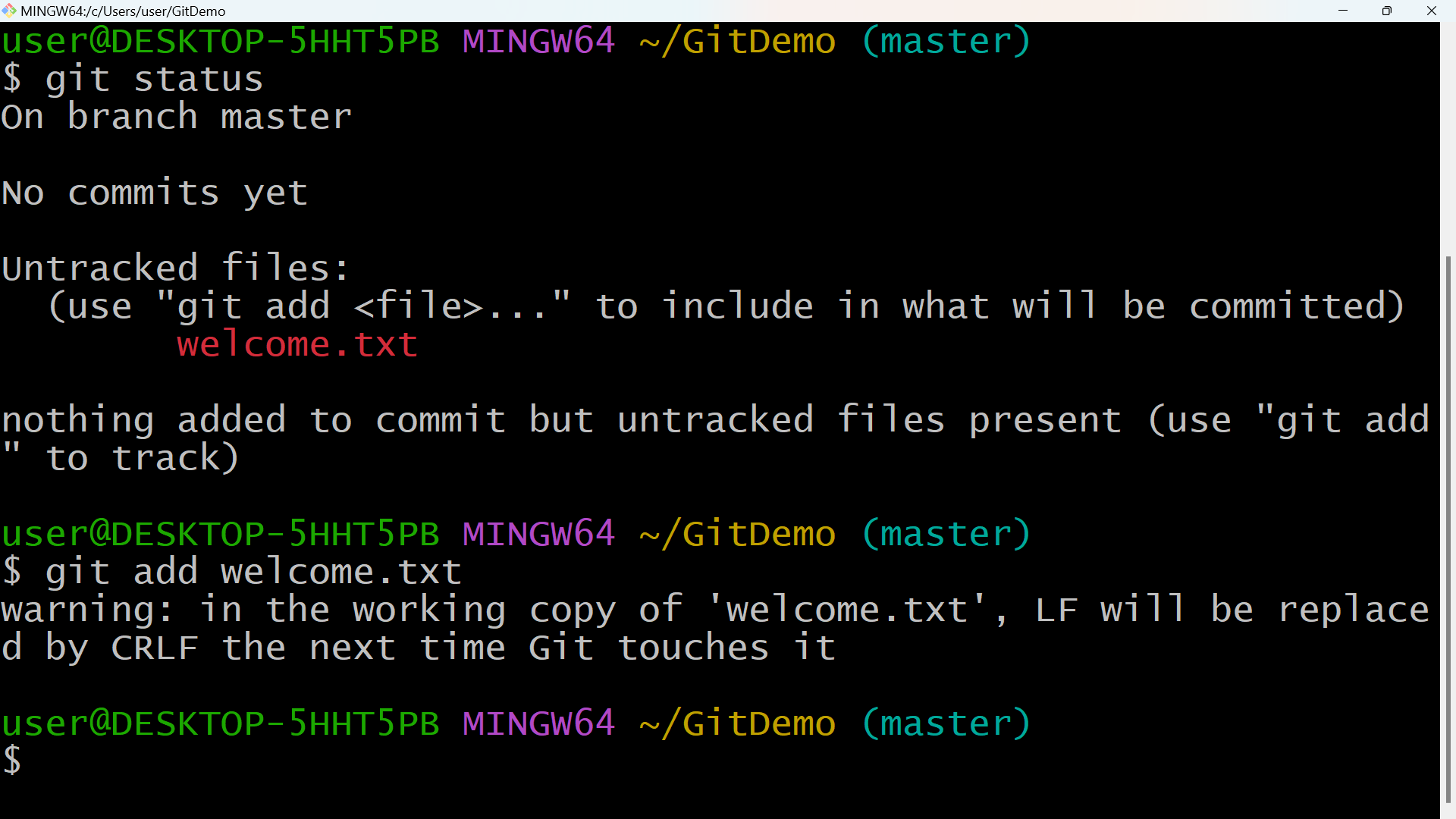


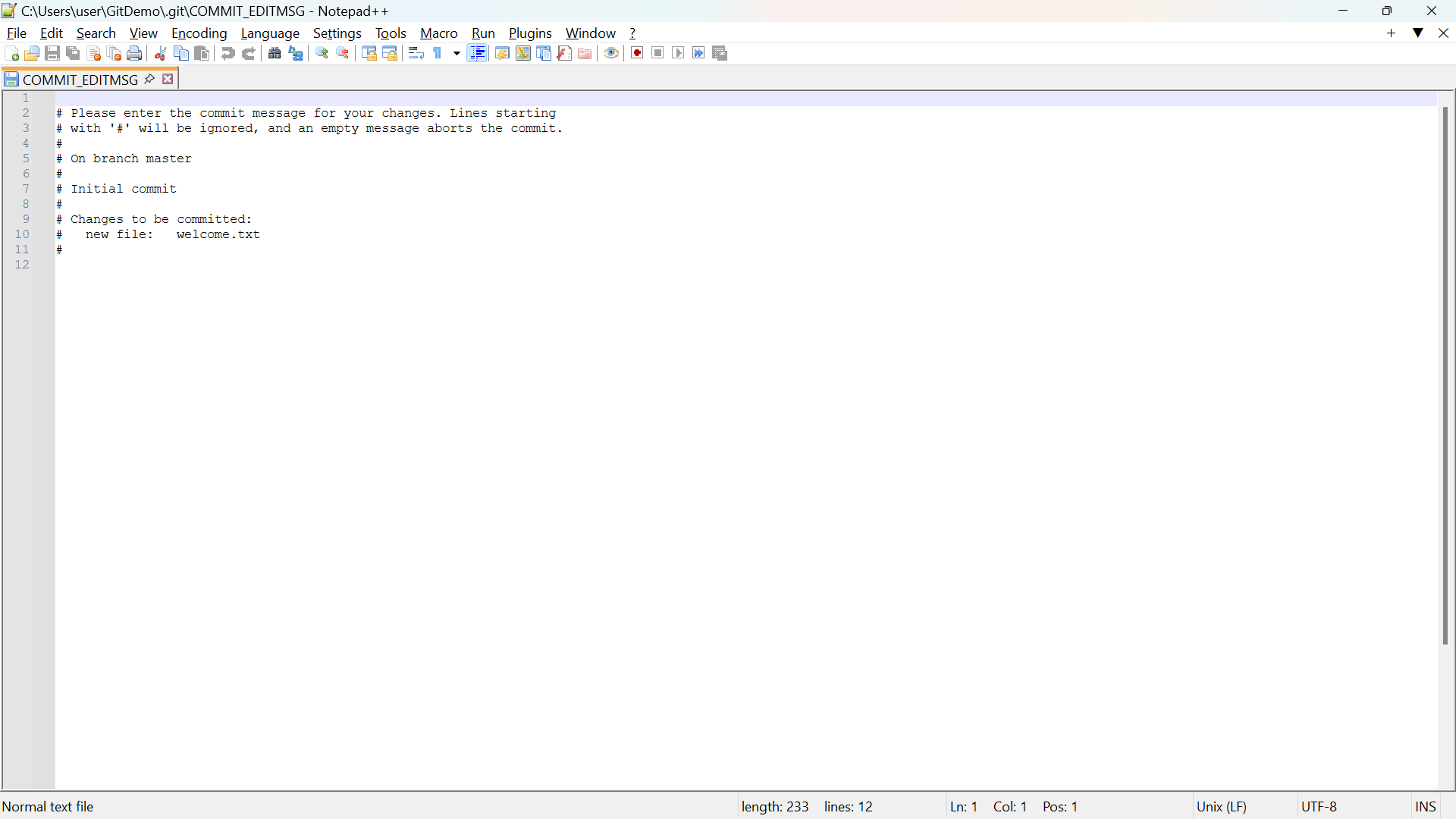


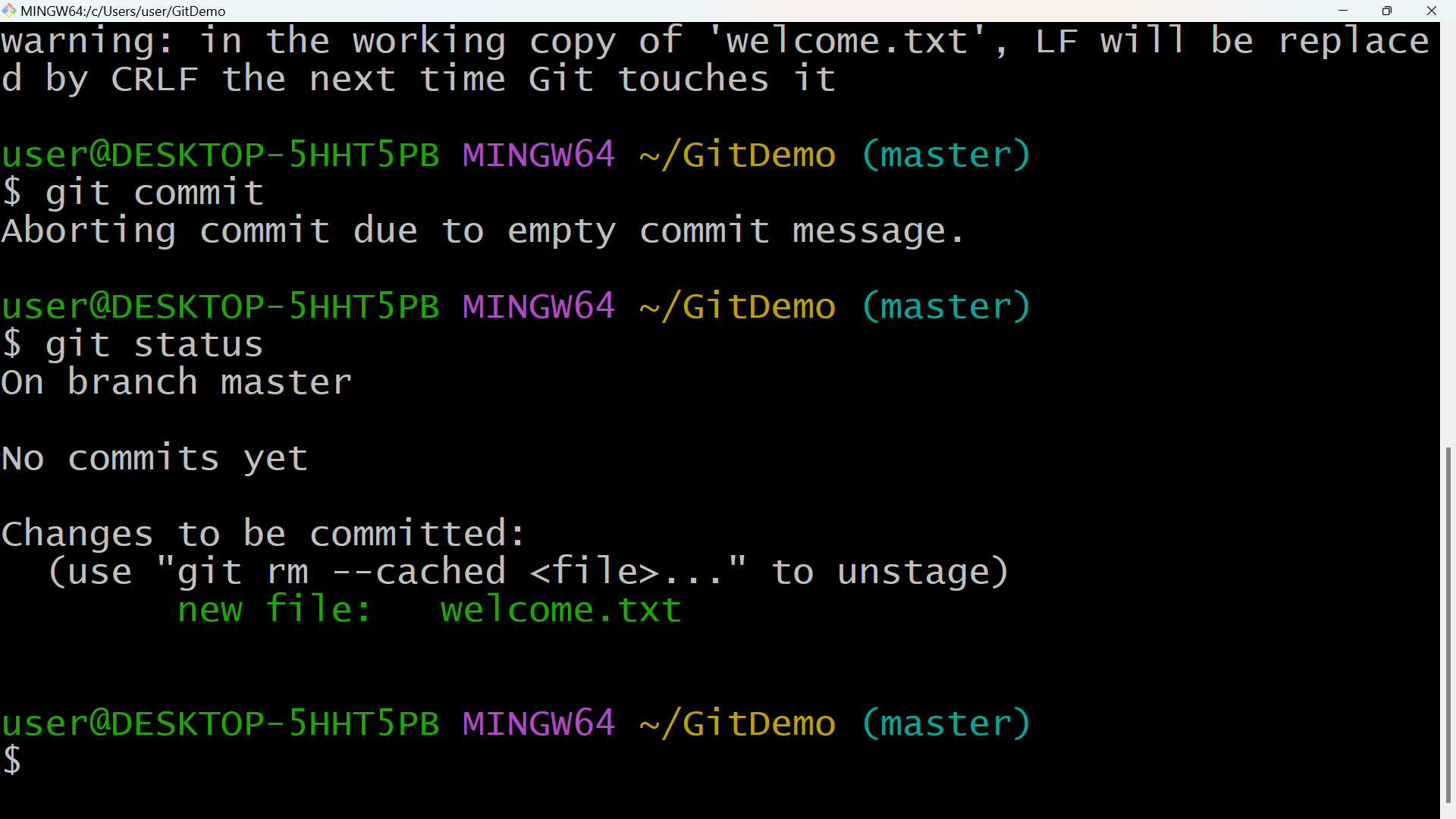


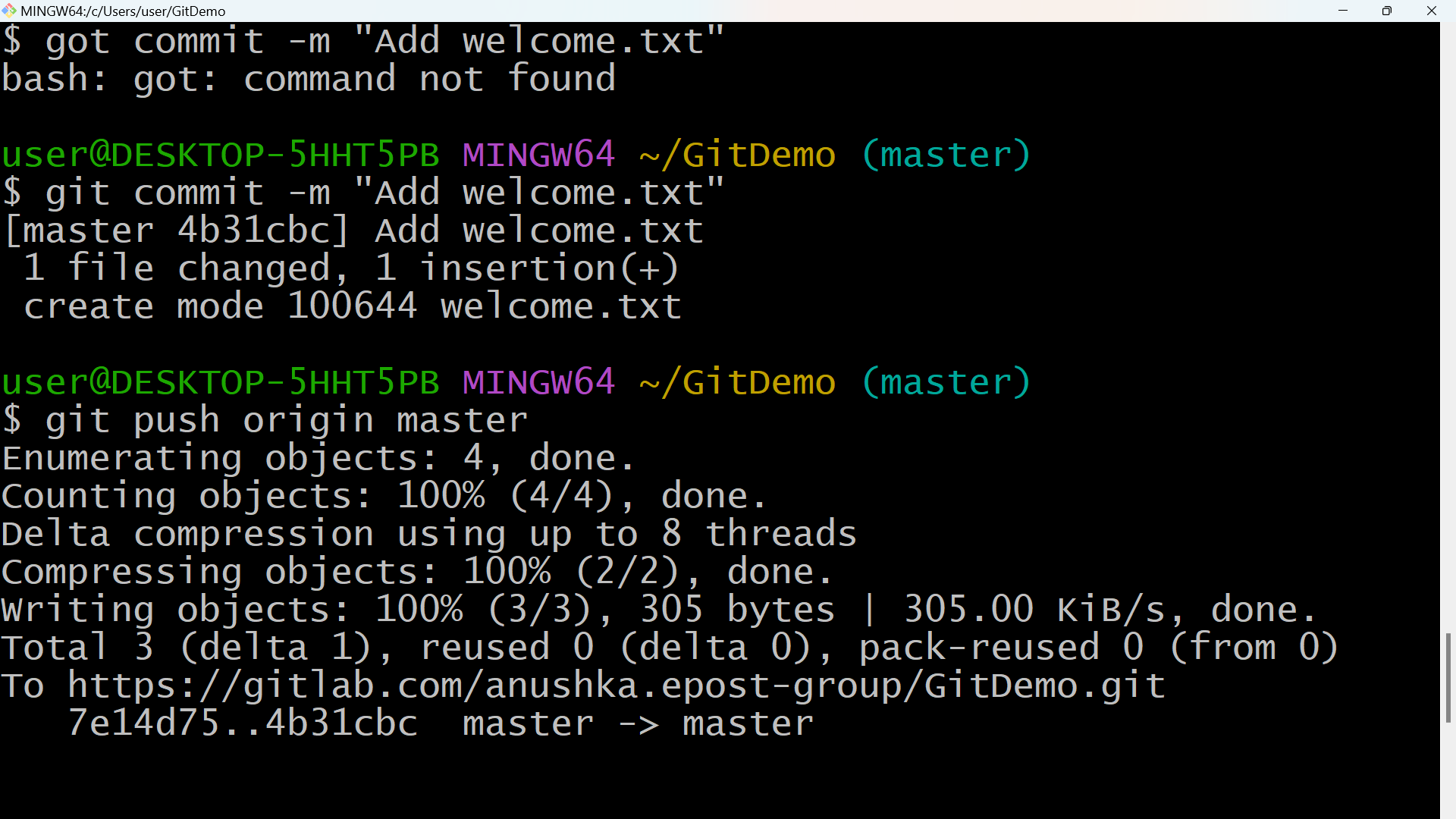












**2.GIT-HOL**

touch debug.log

mkdir log

touch log/app.log

touch .gitignore

notepad++ .gitignore

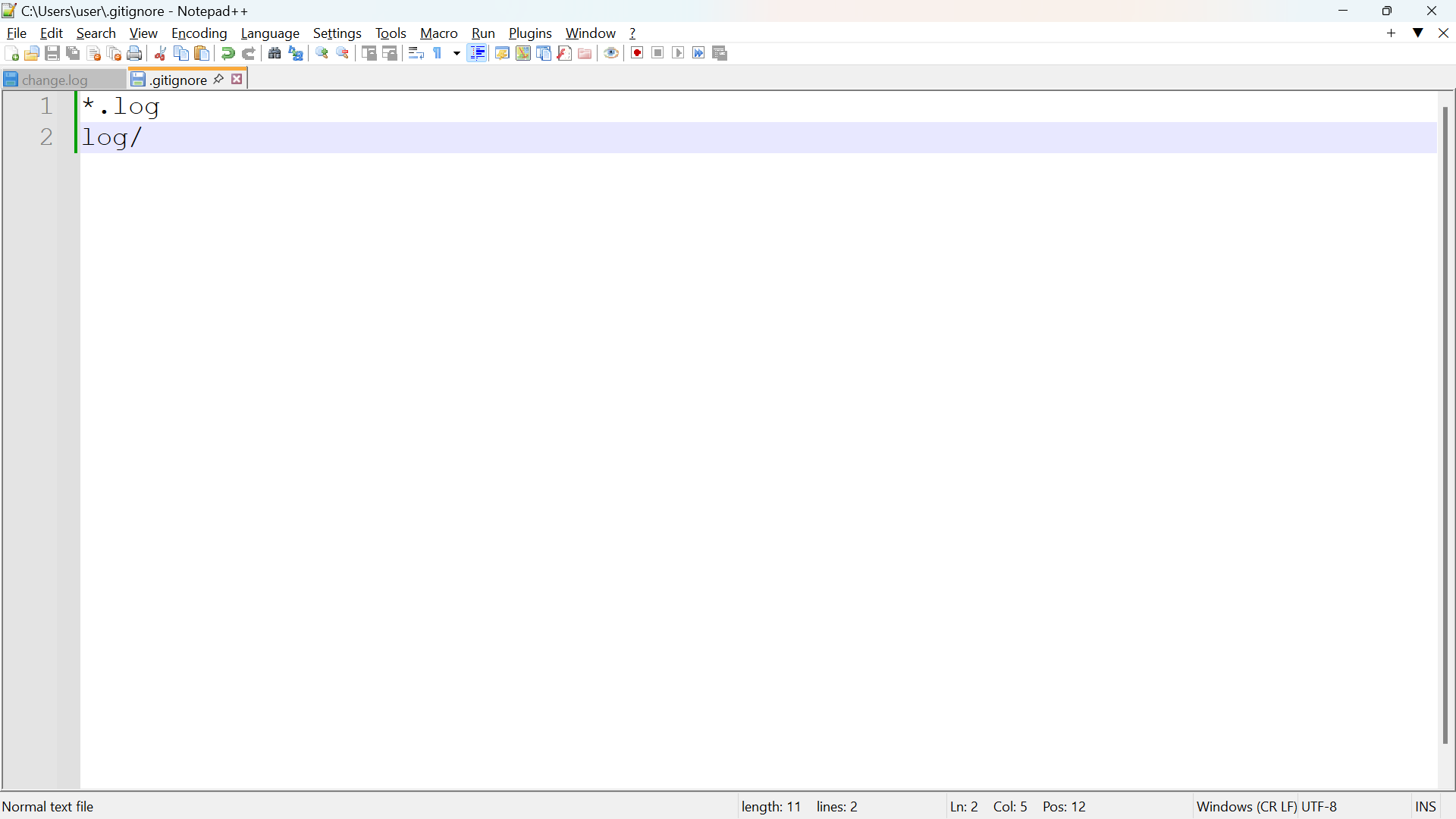
git status

git add .gitignore

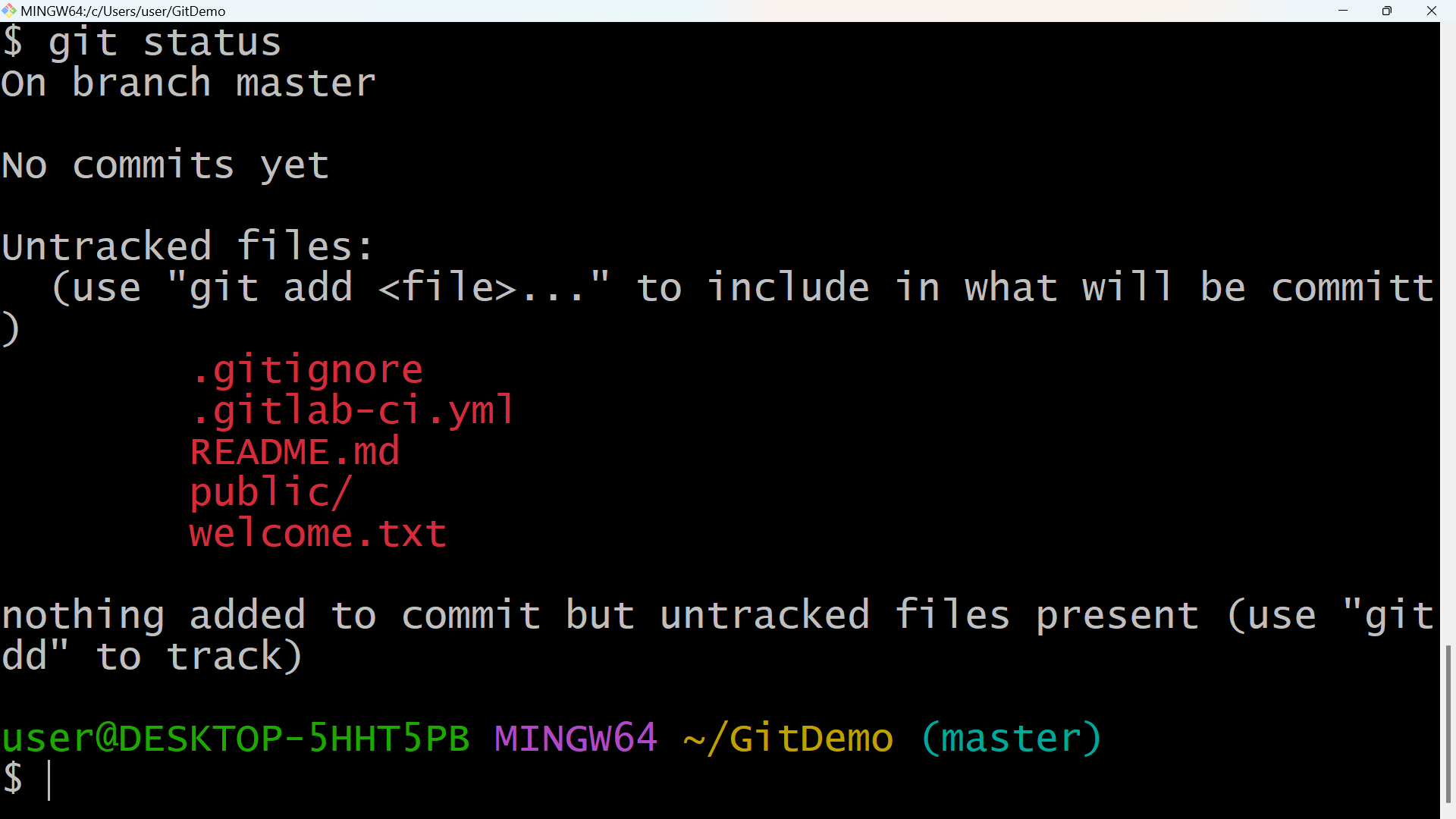
git commit -m "Add .gitignore to ignore .log files and log folder"

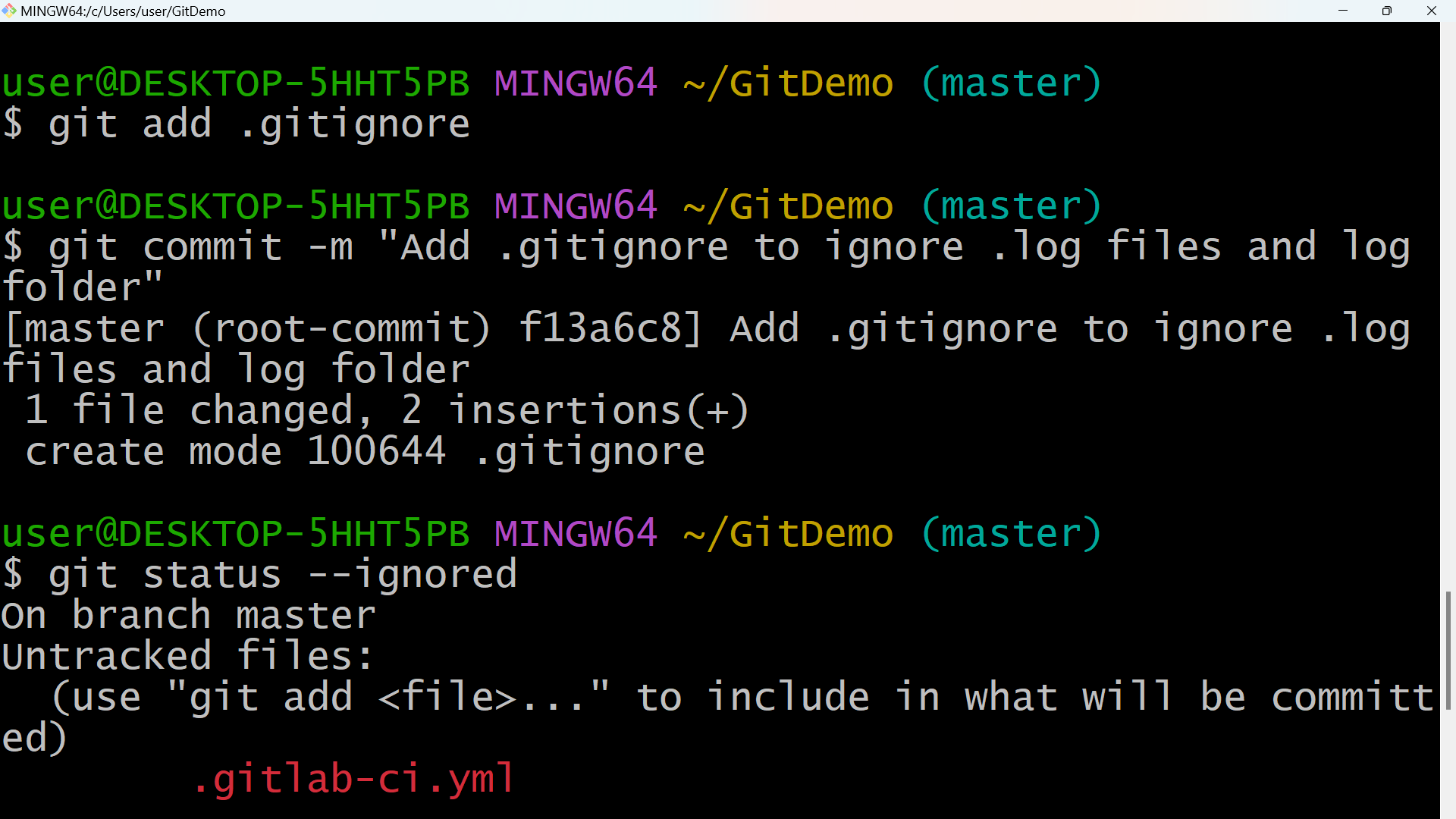
git status --ignored

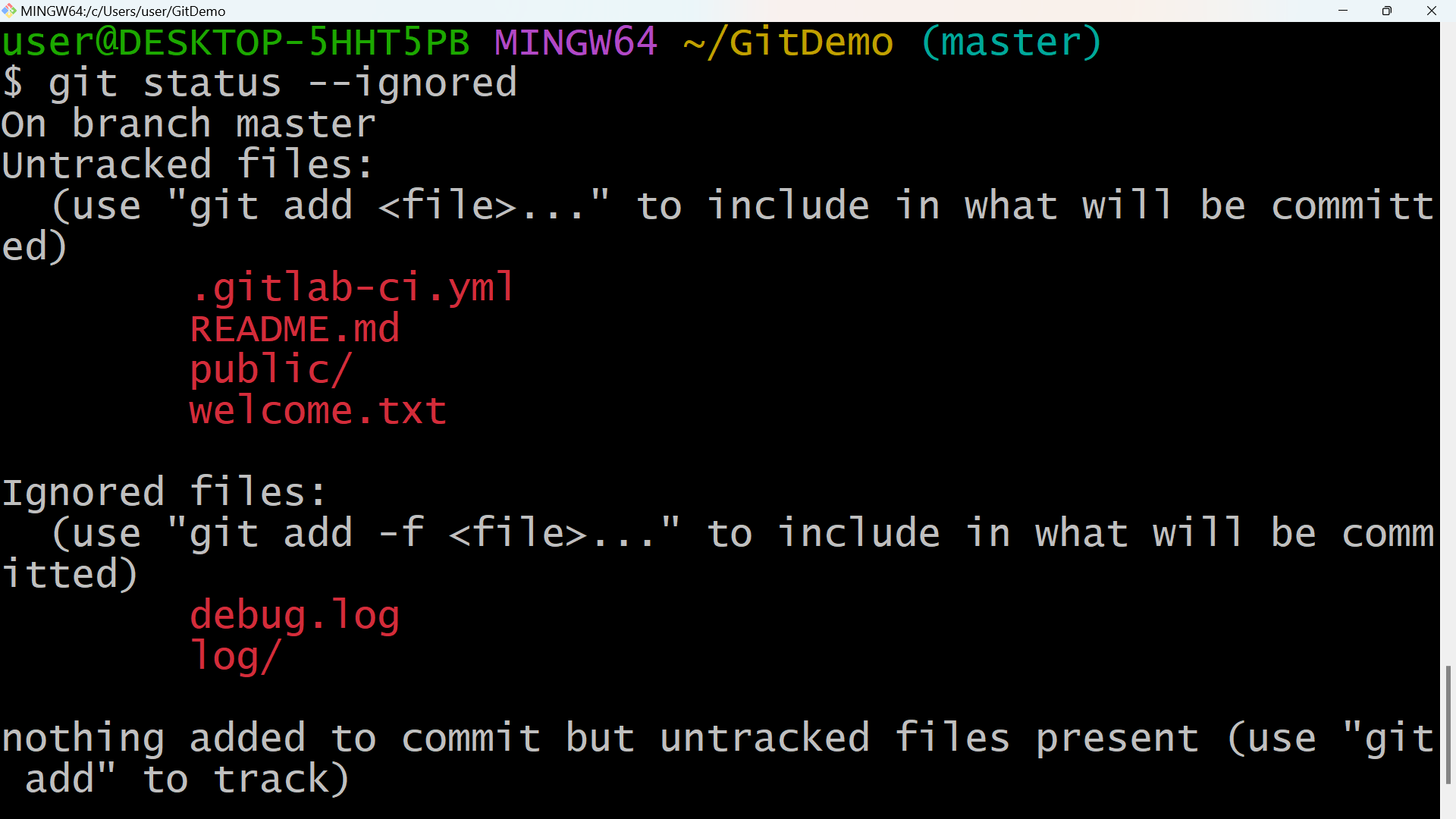
**SNAPSHOTS**











**3.GIT-HOL**

git branch GitNewBranch

git branch -a

git checkout GitNewBranch

echo "This is a feature branch file." > feature.txt

mkdir newFolder

echo "Inside new folder" > newFolder/info.txt

git add .

git commit -m "Add feature.txt and newFolder/info.txt in GitNewBranch"

git status

git checkout master

git diff master GitNewBranch

git difftool master GitNewBranch

git merge GitNewBranch

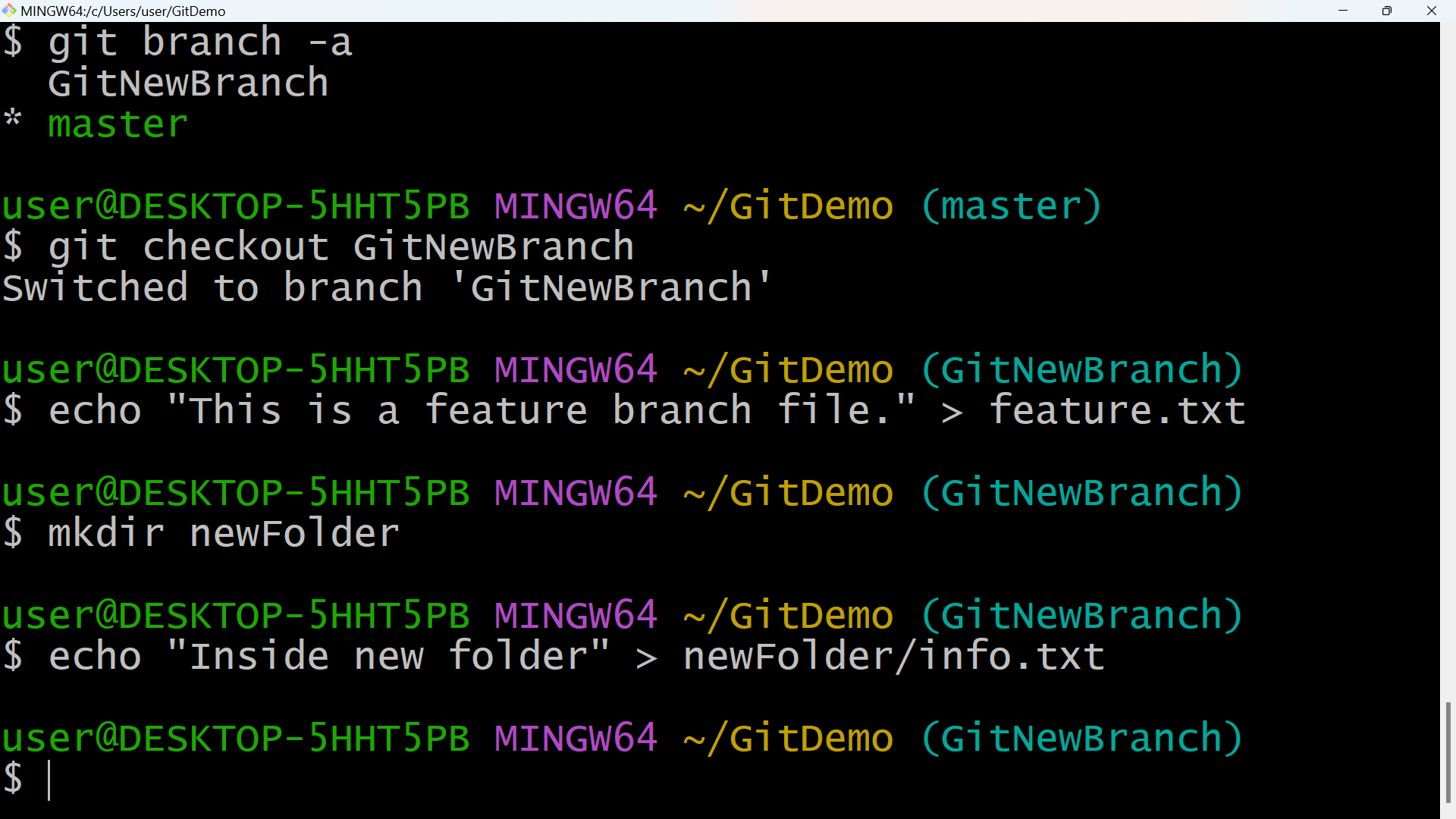
git log --oneline --graph –decorate

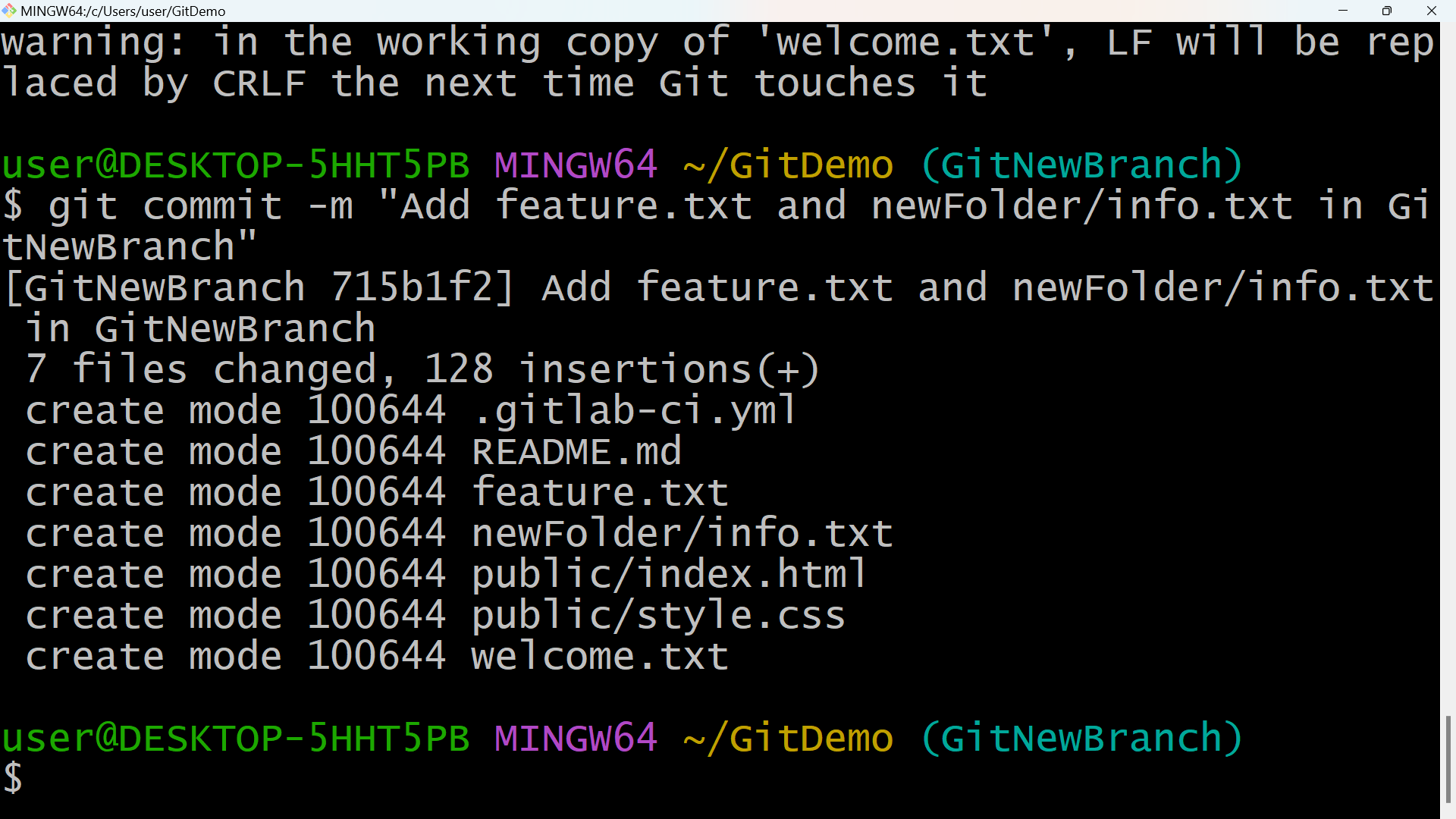
git branch -d GitNewBranch

git status

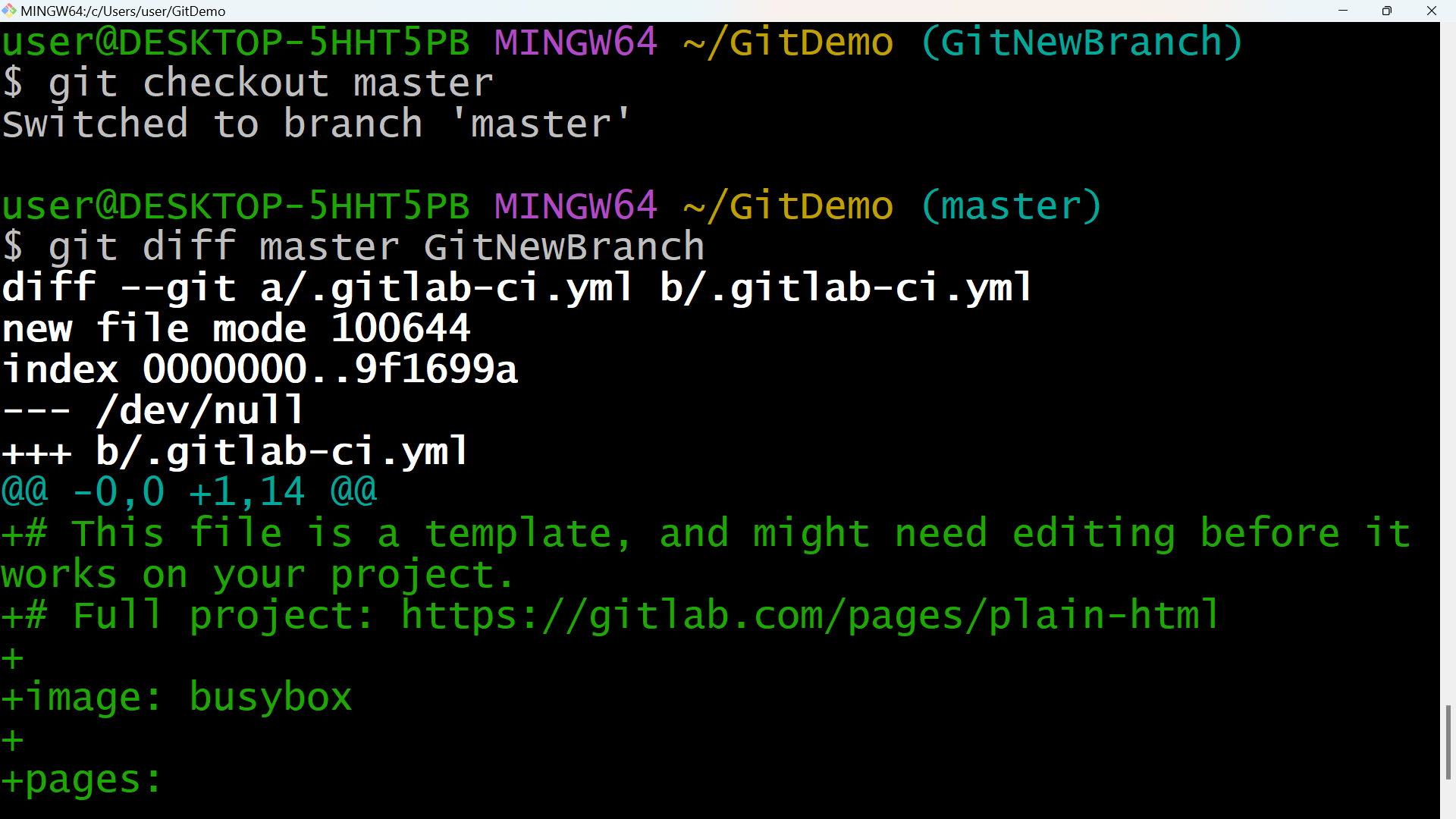
**SNAPSHOTS**

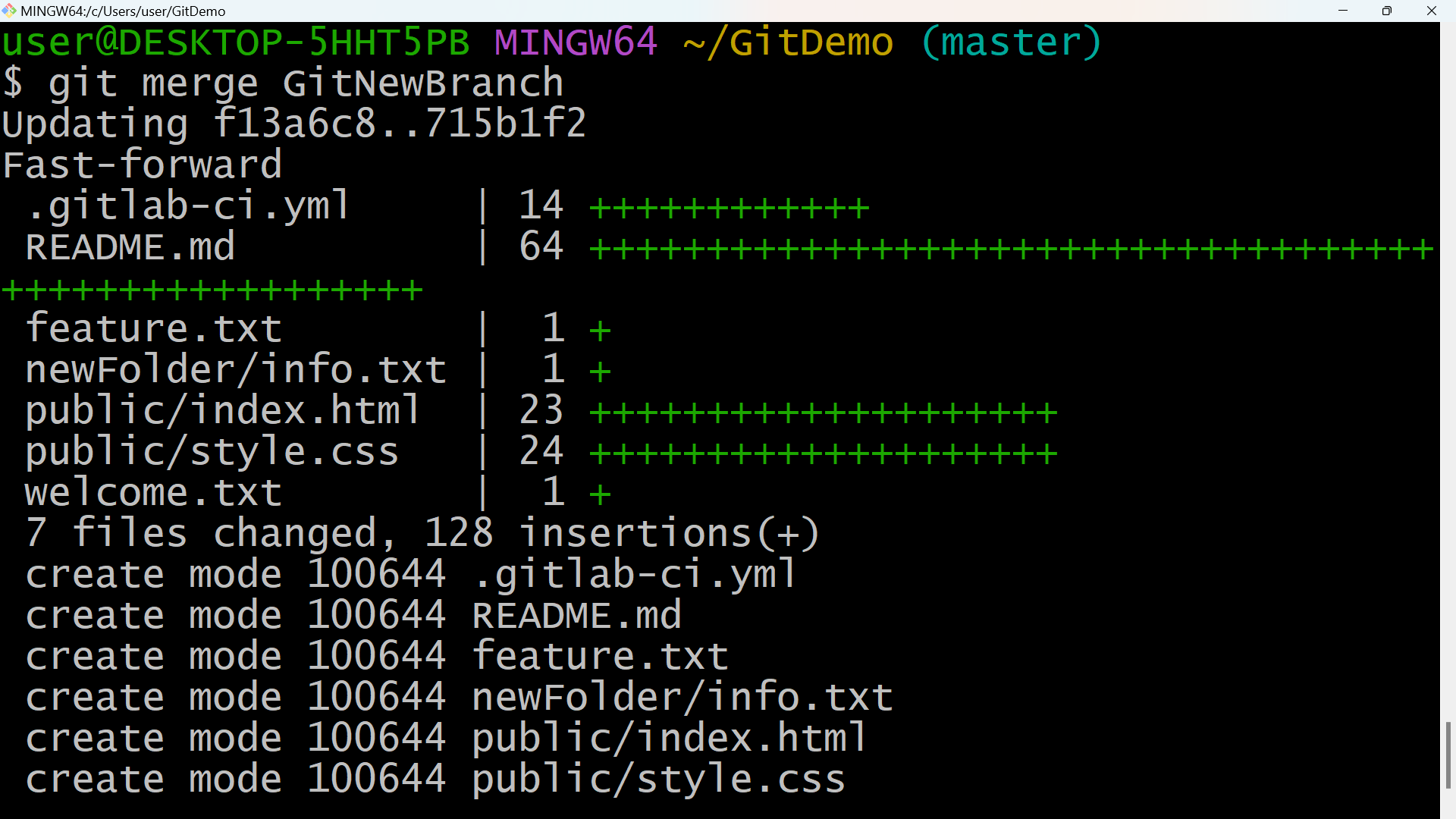


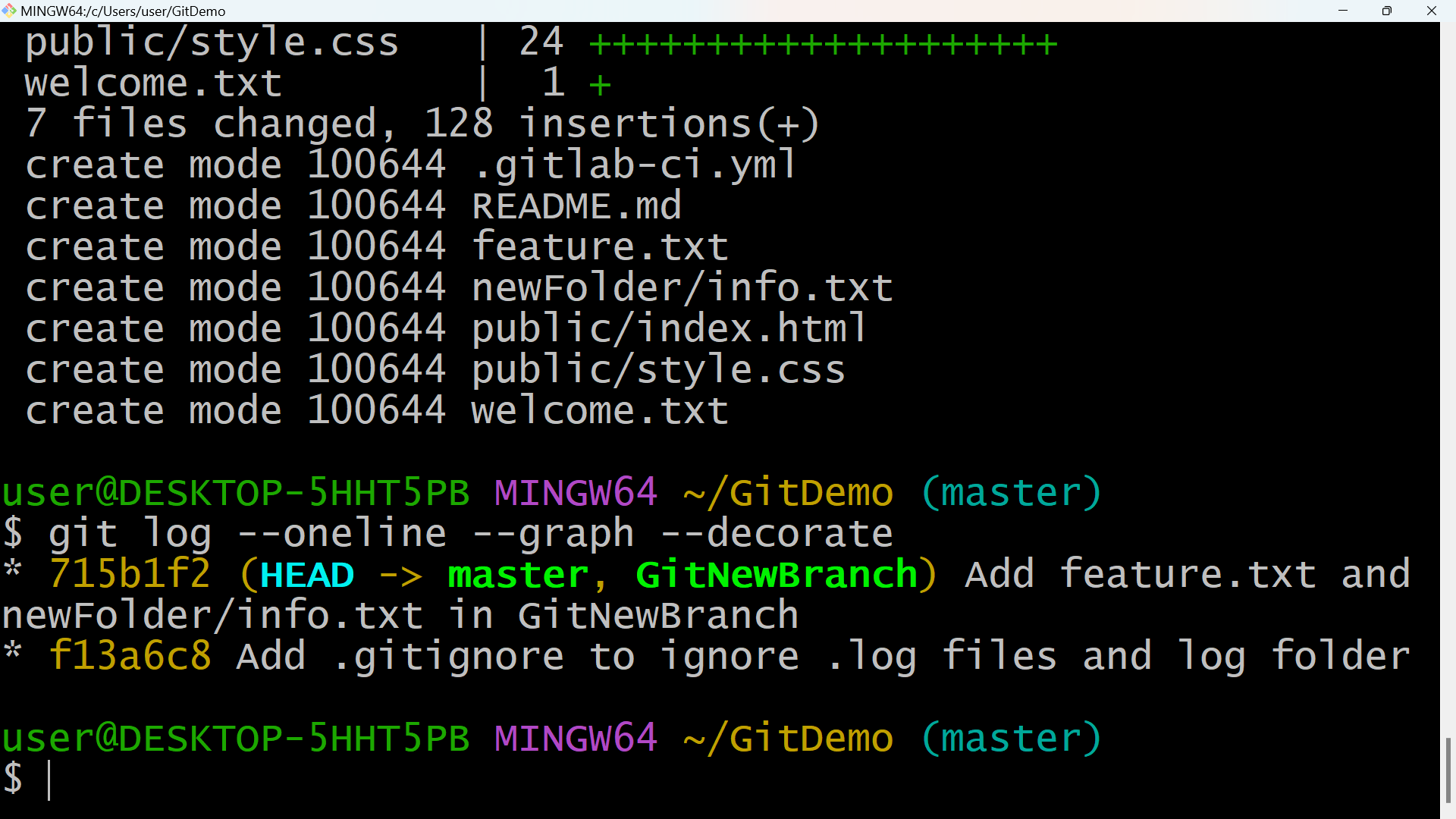


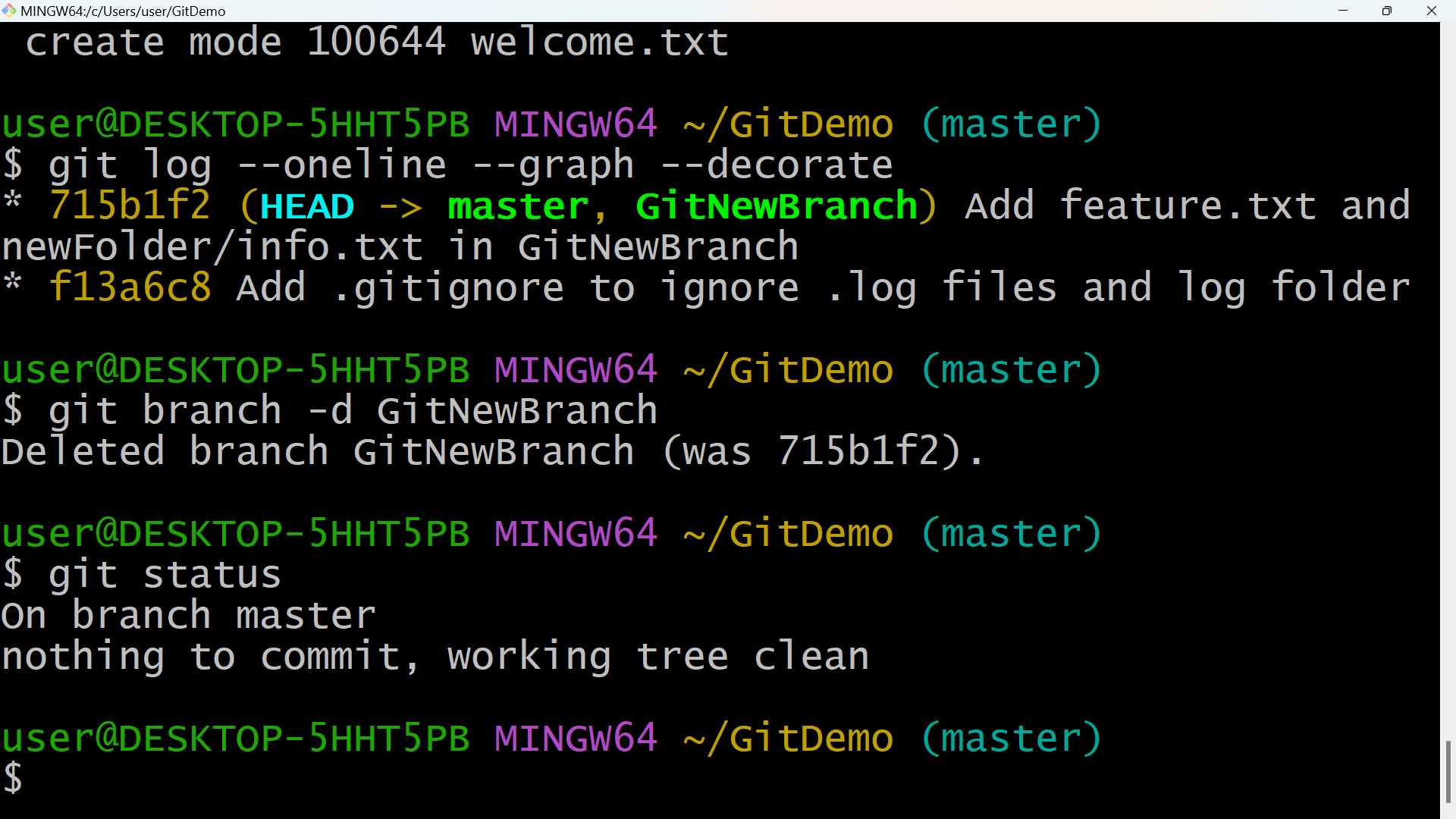












**4.GIT-HOL**

git checkout master

git status

git checkout -b GitWork

echo "<note><to>User</to></note>" > hello.xml

echo "<note><to>User</to><from>Git</from></note>" > hello.xml

git status

git add hello.xml

git commit -m "Add hello.xml with initial content in GitWork branch"

git checkout master

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

git add hello.xml

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

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

git diff master GitWork

git mergetool

git config --global merge.tool p4merge

git merge GitWork

cat hello.xml

git mergetool

git add hello.xml

git commit -m "Merge GitWork into master and resolve conflicts"

git status

echo "\*.orig" >> .gitignore

git add .gitignore

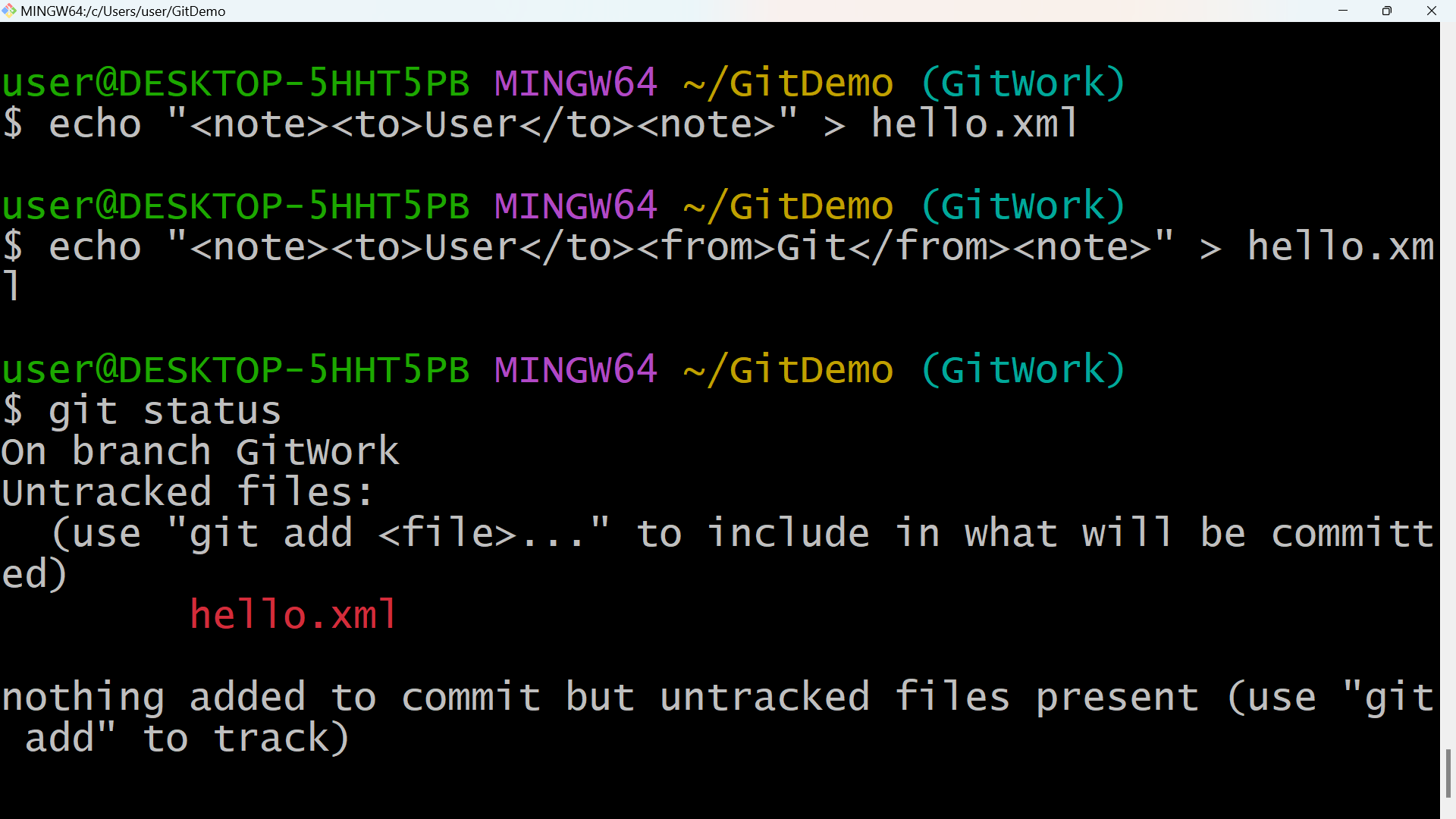
git commit -m "Ignore backup files"

git branch

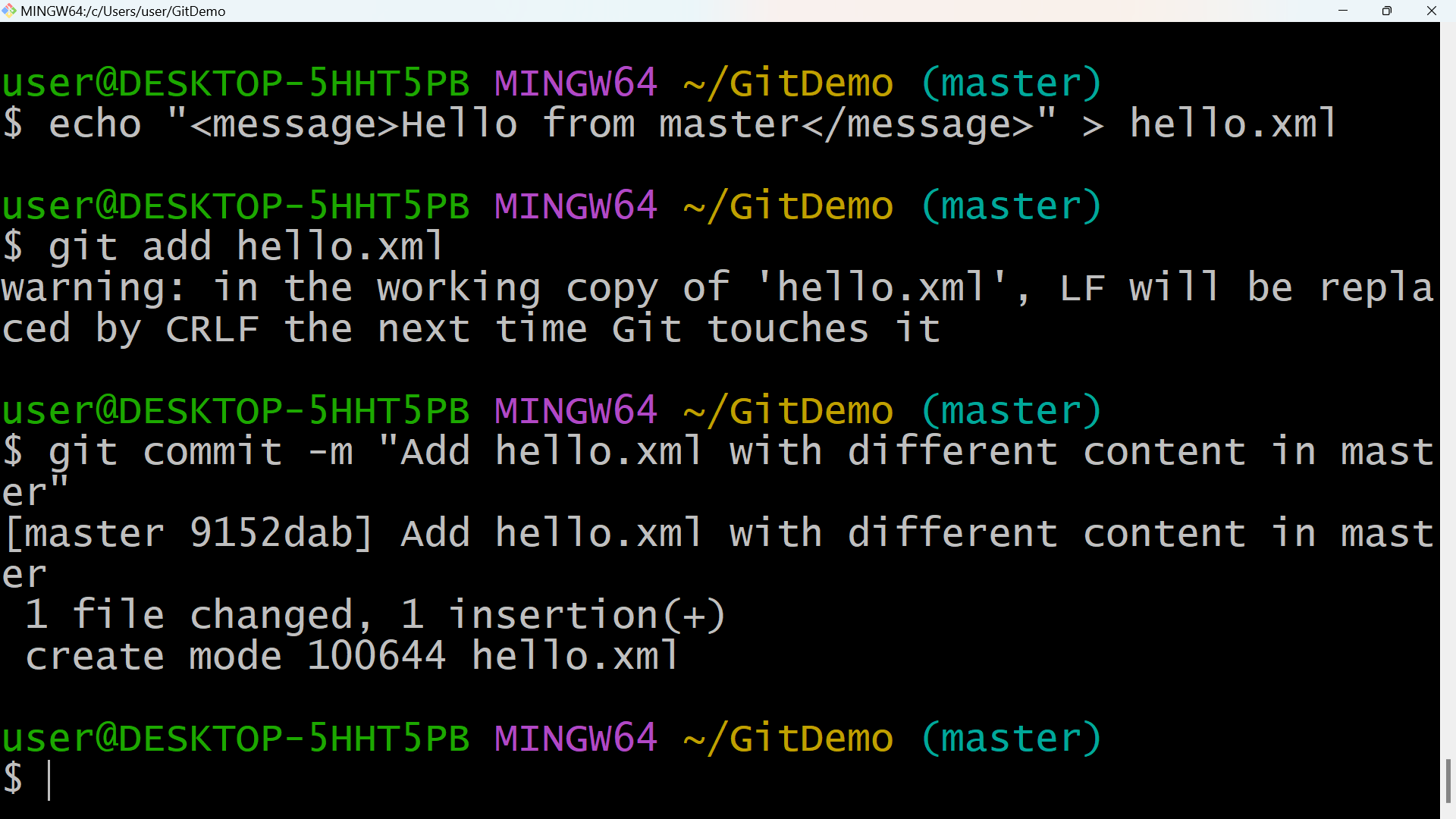
git branch -d GitWork

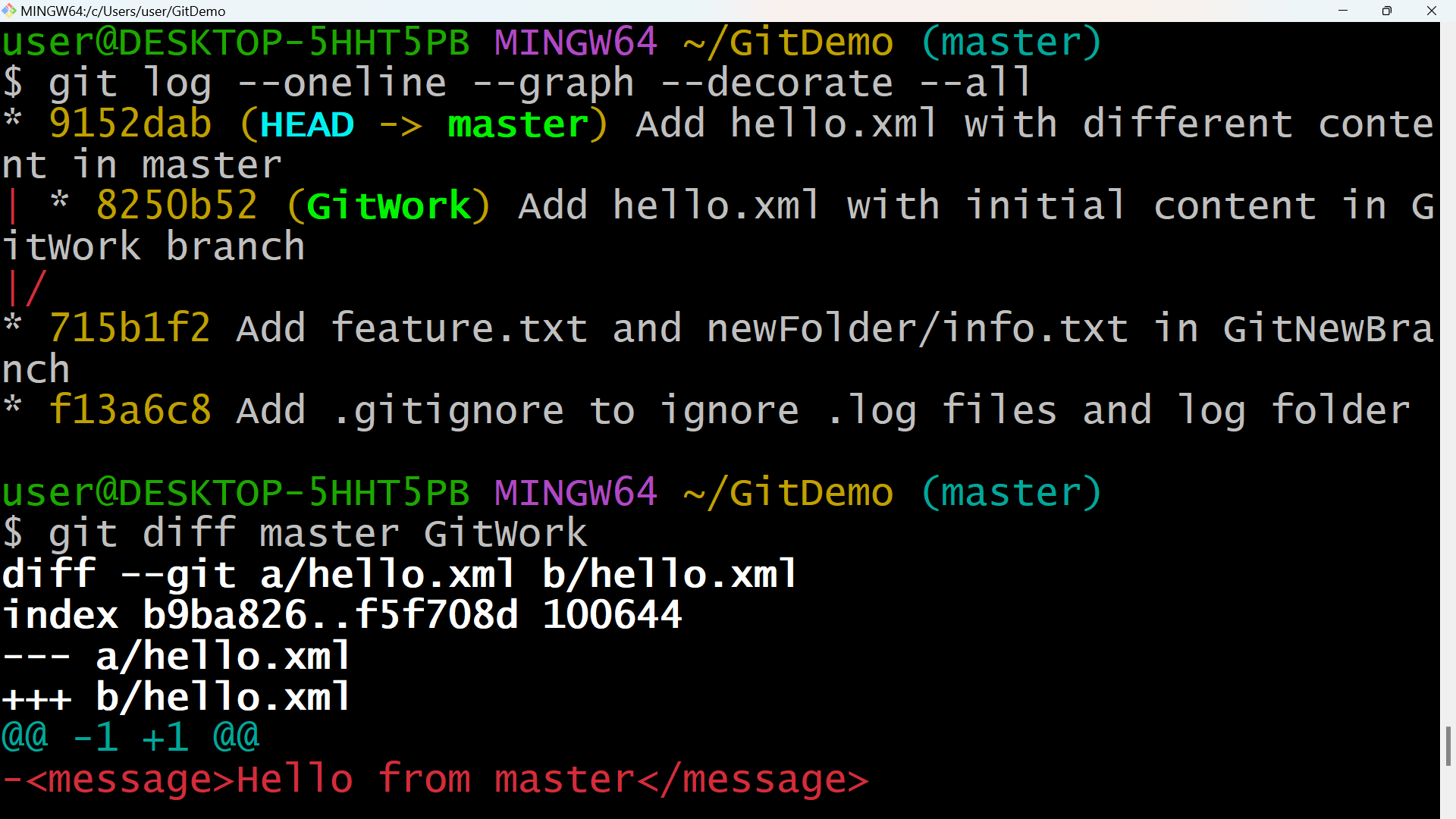
git log --oneline --graph --decorate

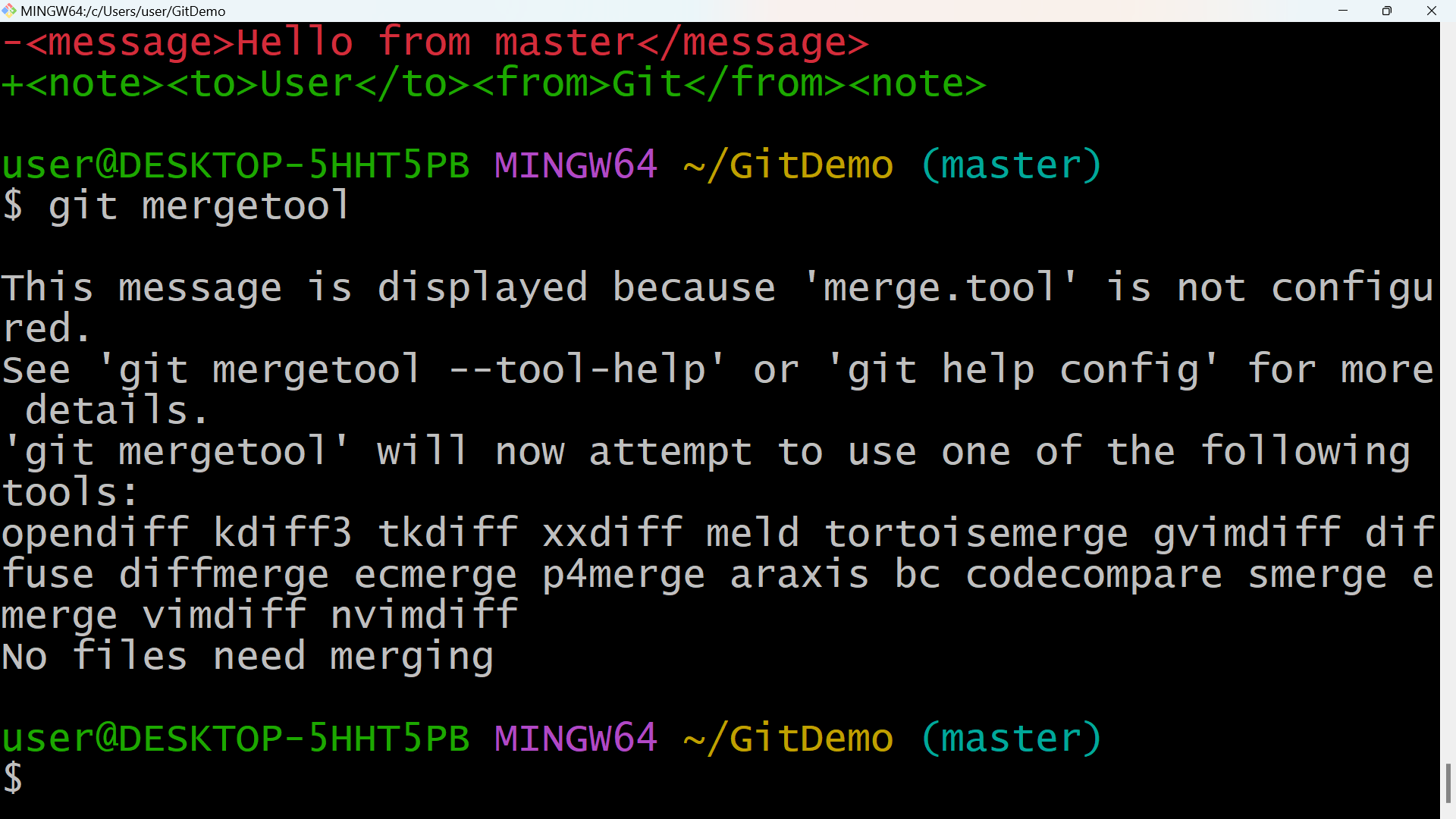
**SNAPSHOTS**



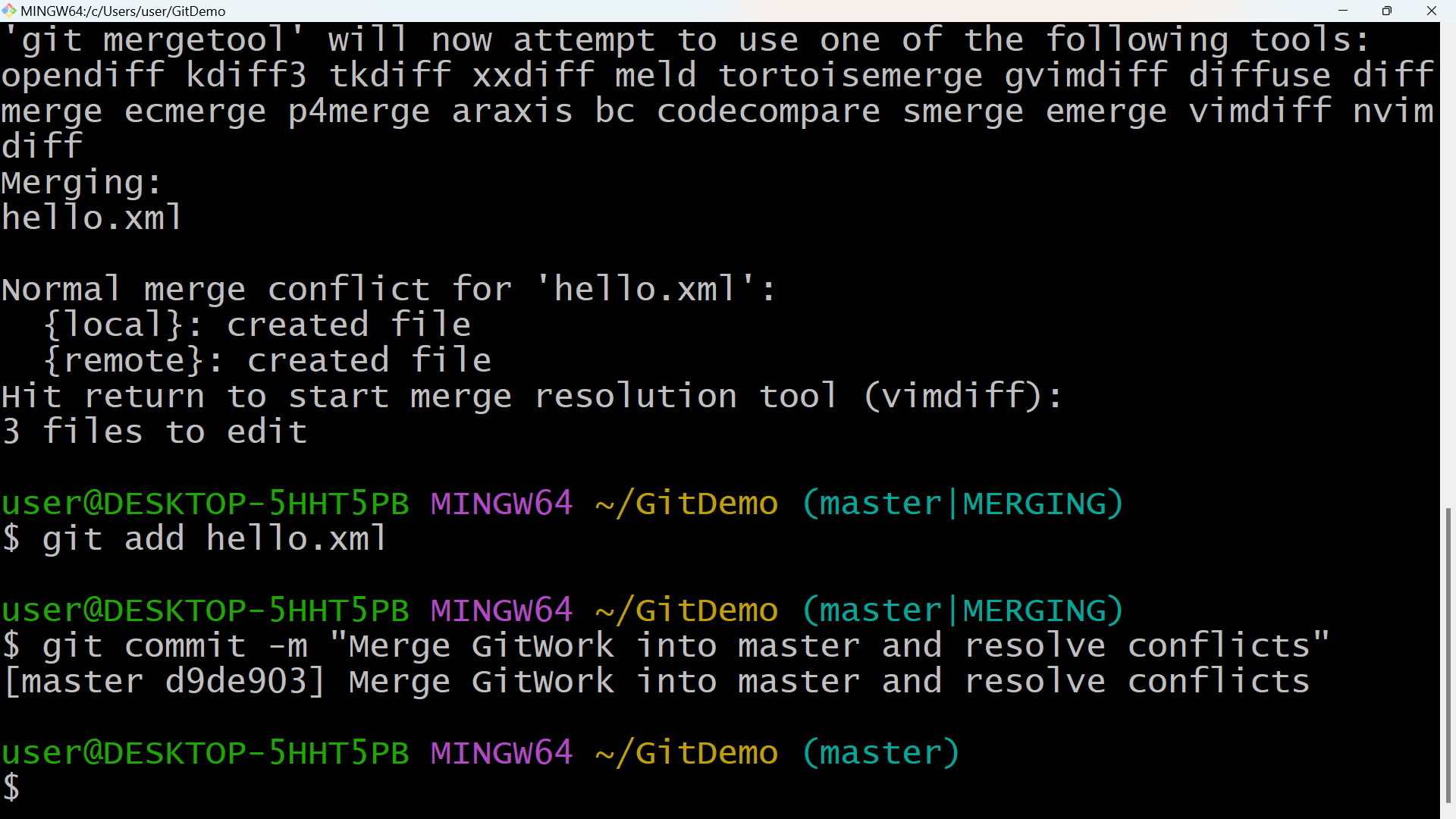




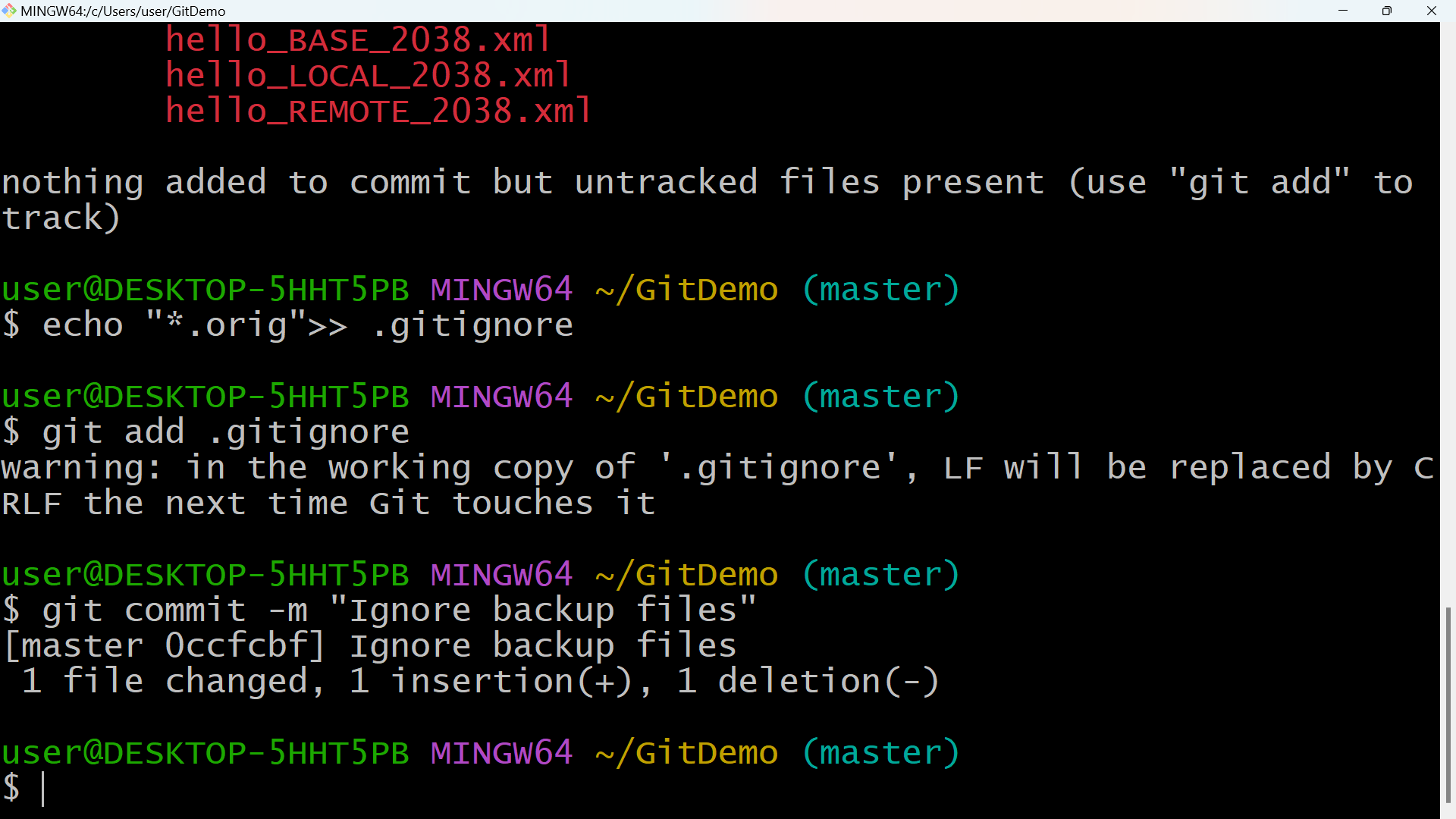


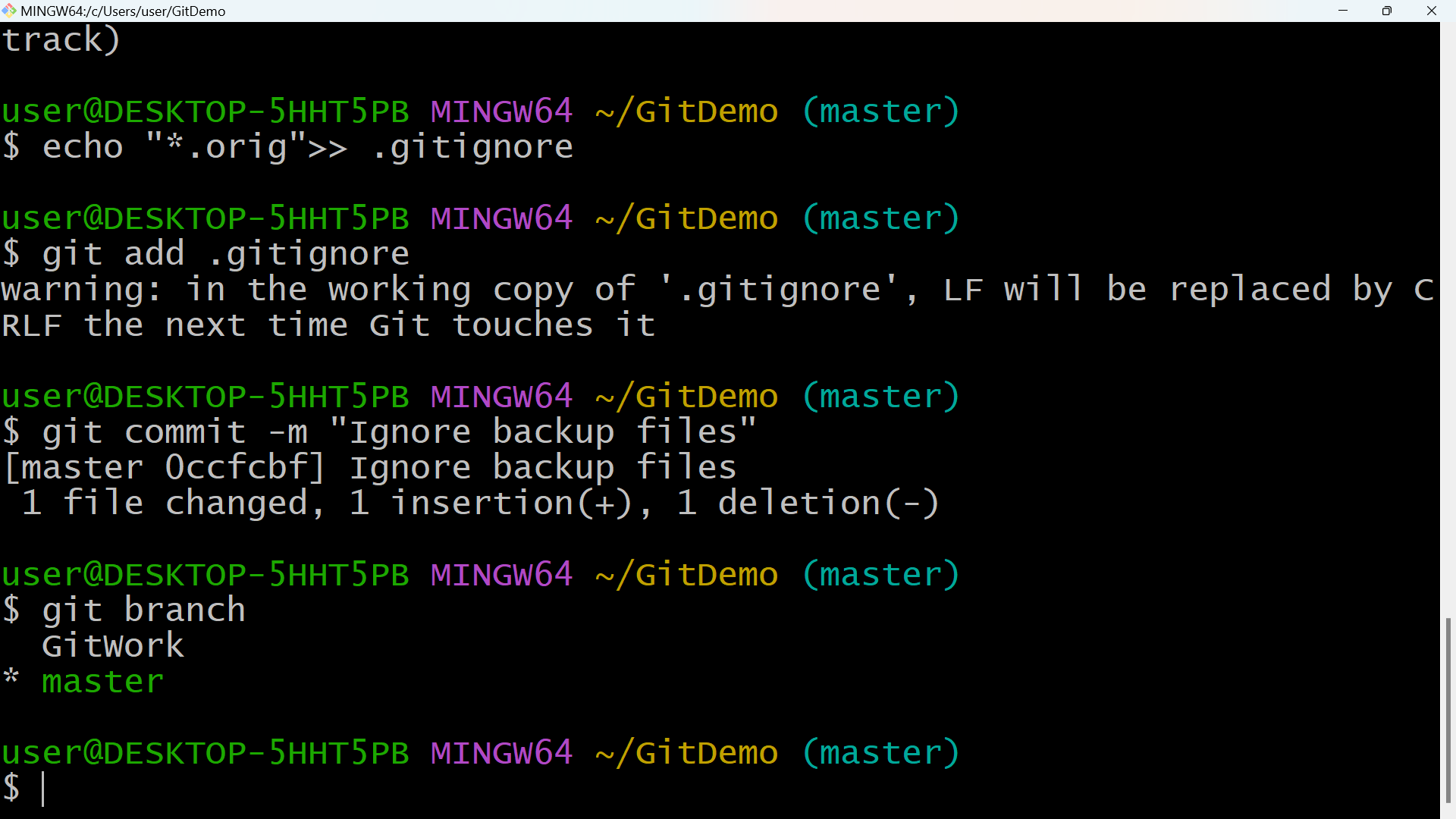


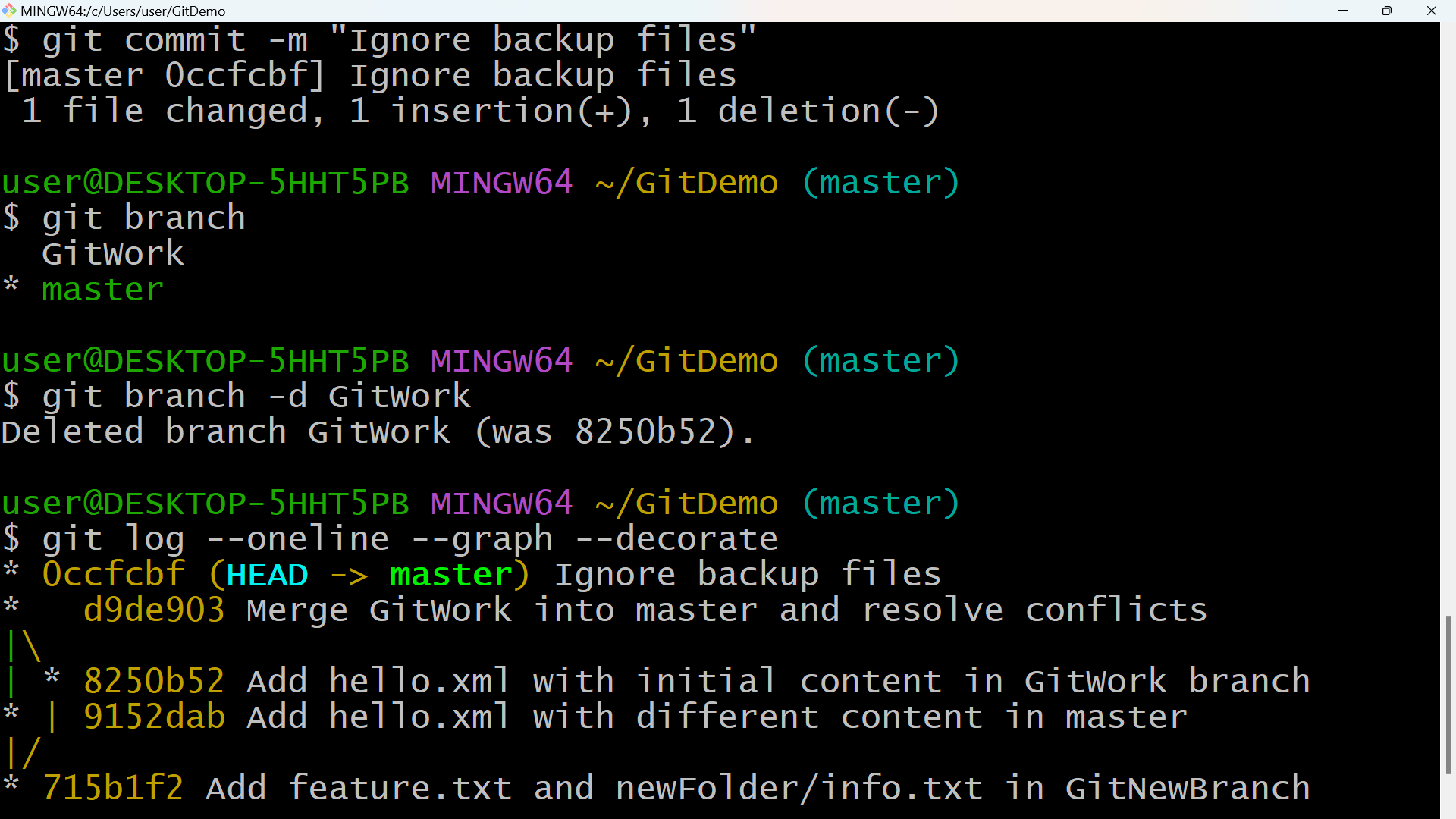












**5.GIT-HOL**

git status

git clean -f # this I used for cleaning the git

git branch

git branch -r

git branch -a

git checkout master

git pull origin master

git checkout Git-T03-HOL\_002

git push origin Git-T03-HOL\_002

**SNAPSHOTS**

