**WEEK-8**

**GIT-HOL**

**Excercise-1:** **Install Git Bash client in your machine**

git config --global user.name Vineela

git config --global user.email vineelamankidi@gmail.com

git config --global –list

git config --global core.editor "notepad"

mkdir GitDemo

cd GitDemo

git init

echo Welcome to Git Demo > welcome.txt

git add welcome.txt

git commit

In Notepad, type:

sql

CopyEdit

First commit

git status

git log

**Local Repo to GitLab**

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

git pull origin master

git push -u origin master

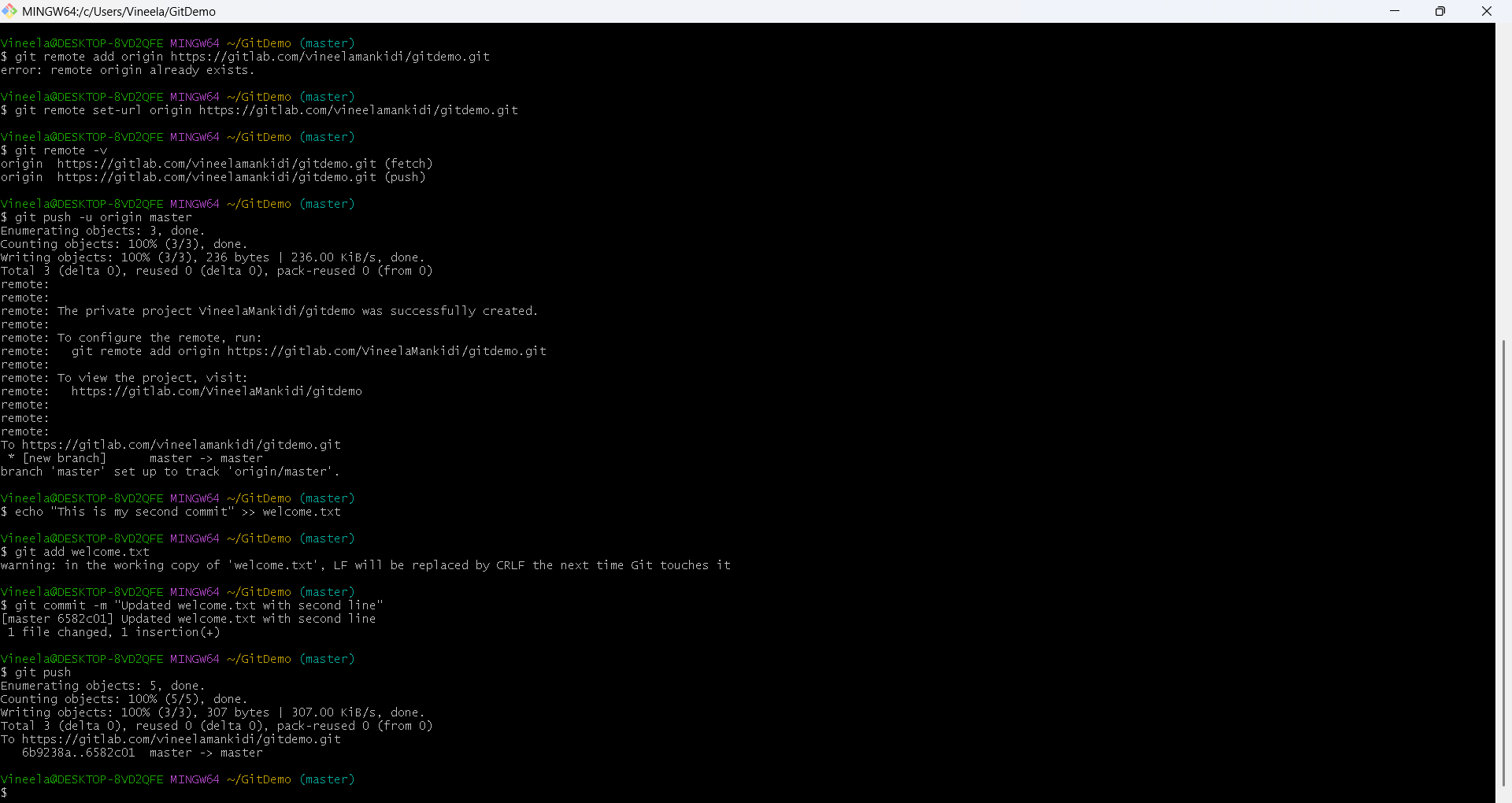
git remote -v

echo "This is my second commit" >> welcome.txt

git add welcome.txt

git commit -m "Updated welcome.txt with second line"

git push





**Exercise-2: Implement git ignore command to ignore unwanted files and folders**

Create a .log file:

echo "This is a sample log file" > debug.log

mkdir log

echo "Log file inside log folder" > log/error.log

touch .gitignore

notepad .gitignore

Add the following lines:

\*.log

log/

git add .gitignore

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

git status

git push

**OUTPUT:**



**Exercise-3: Construct a branch, do some changes in the branch, and merge it with master**

git branch GitNewBranch

git branch -a

git checkout GitNewBranch

echo "This is a file in GitNewBranch" > branchfile.txt

git add branchfile.txt

git commit -m "Added branchfile.txt in GitNewBranch"

git status

Merging:

git checkout master

git diff GitNewBranch

git merge GitNewBranch

git log --oneline --graph –decorate

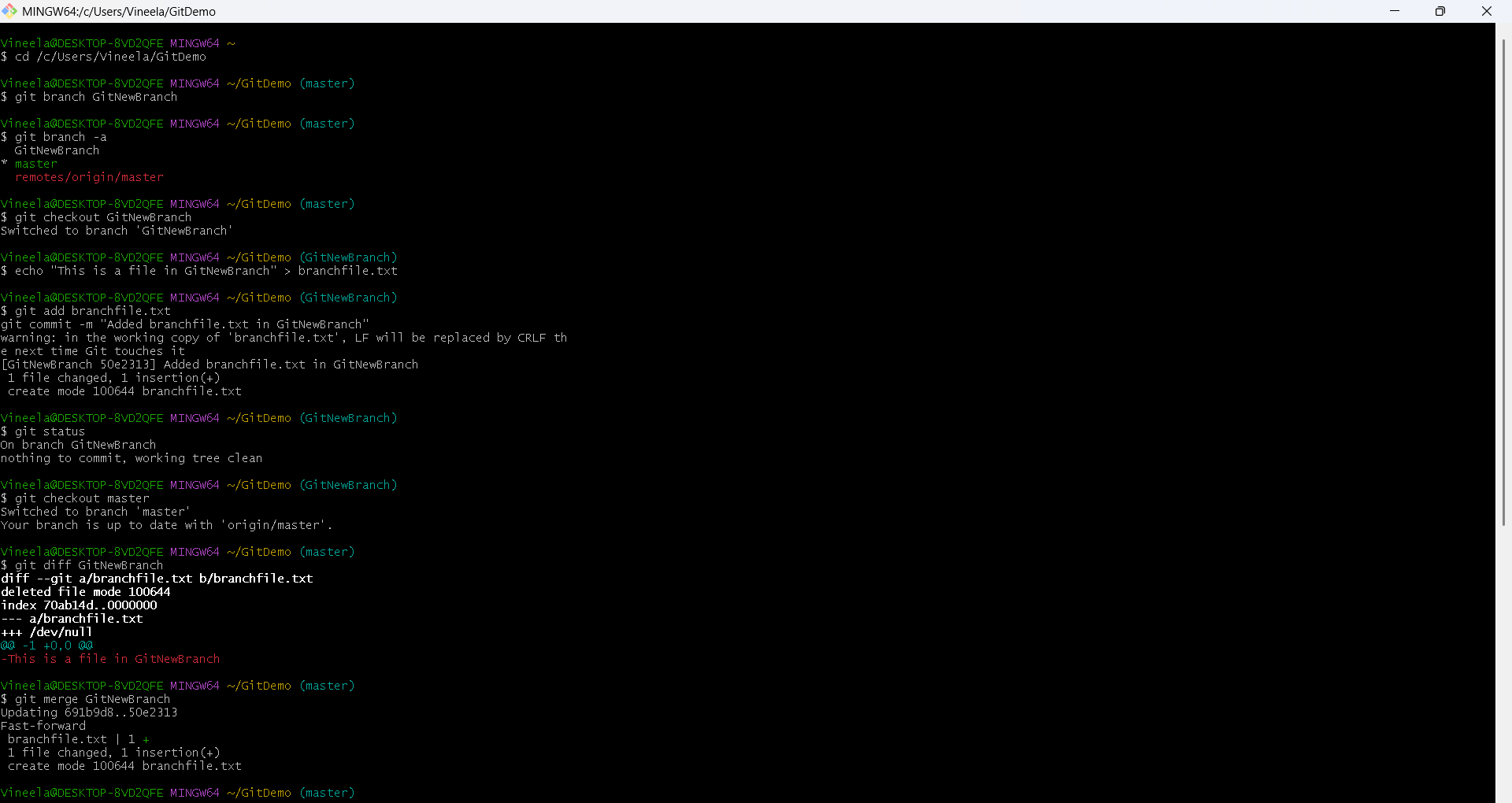
git branch -d GitNewBranch

git status

git push

**OUTPUT:**





**Exercise-4: Implement conflict resolution when multiple users are updating the trunk (or master) in such a way that it results into a conflict with the branch’s modification.**

cd /c/Users/Vineela/GitDemo

git checkout master

git status

git branch GitWork

git checkout GitWork

echo "<greeting>Hello from branch</greeting>" > hello.xml

git status

git add hello.xml

git commit -m "Added hello.xml in GitWork branch"

git checkout master

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

git add hello.xml

git commit -m "Added hello.xml in master branch"

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

git diff GitWork

git merge GitWork

notepad hello.xml

<greeting>Hello from both master and branch</greeting>

git add hello.xml

git commit -m "Resolved merge conflict in hello.xml"

echo "\*.\*~" >> .gitignore

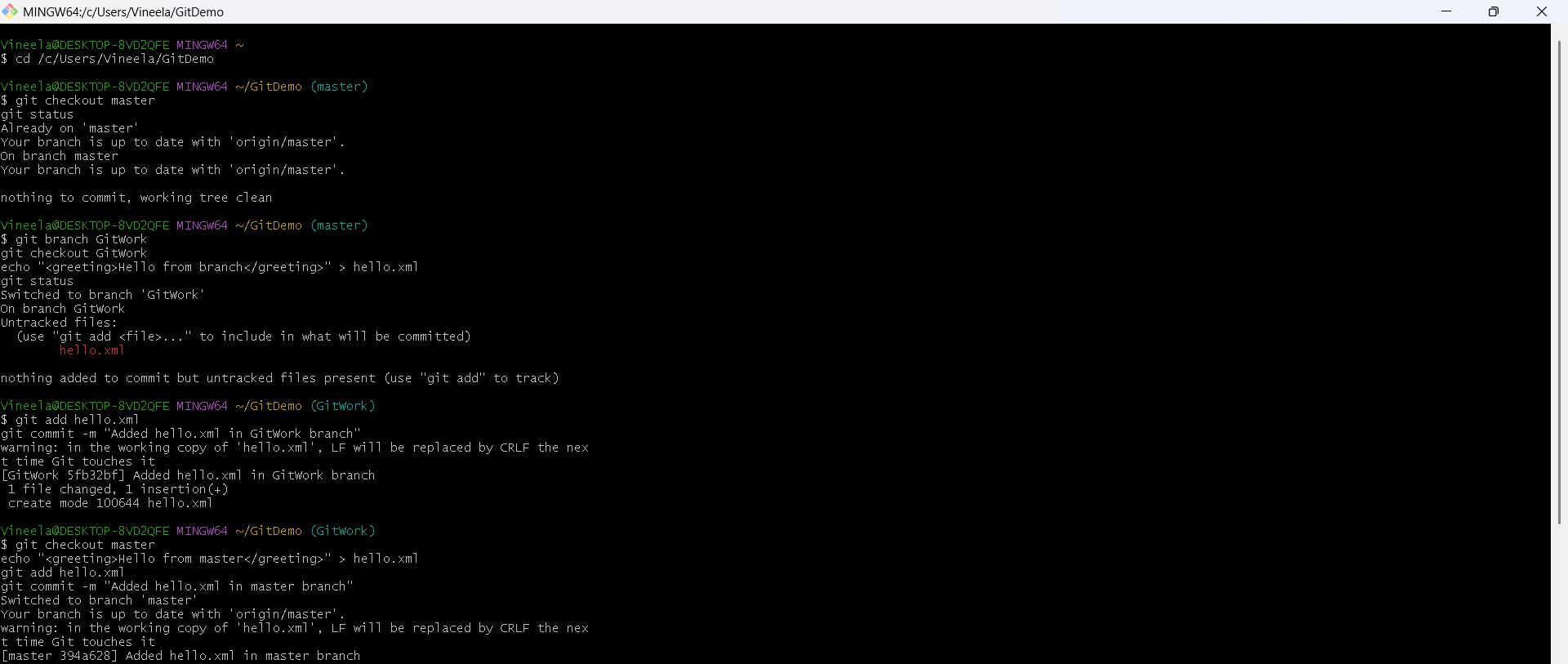
git add .gitignore

git commit -m "Added backup files pattern to .gitignore"

git branch -d GitWork

git push

**OUTPUT:**





**Exercise-5: Execute steps involving clean up and push back to remote Git.**

git checkout master

git status

git branch -a

git pull origin master

git push origin master

**OUTPUT:**

****