# **Name- Sayandeep Dey (SupersetID:** **6363427)**

# **WEEK – 8 (Handson- Exercises)**

**GIT:**

**LAB-1: GIT-HOL:-**

### **Code:**

git --version

git config --global user.name "Sayandeep Dey"

git config --global user.email "sayandeepdey1511@gmail.com"

mkdir -p ~/GitDemo && cd ~/GitDemo

git init

echo "Welcome to GitDemo — created on $(date)" > welcome.txt

git add welcome.txt

git commit -m "Add welcome.txt"

git remote add origin https://gitlab.com/git7373932/GitDemo.git

git branch -M main

git push -u origin main

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.

**LAB-2:GIT-HOL:-**

**Code:**

Create a .gitignore file with patterns for .log files and log/ folder

printf "%s\n" "\*.log" "log/" > .gitignore

Check status (shows .gitignore as untracked)

git status

Stage and commit .gitignore

git add .gitignore

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

Check ignored files

git status --ignored

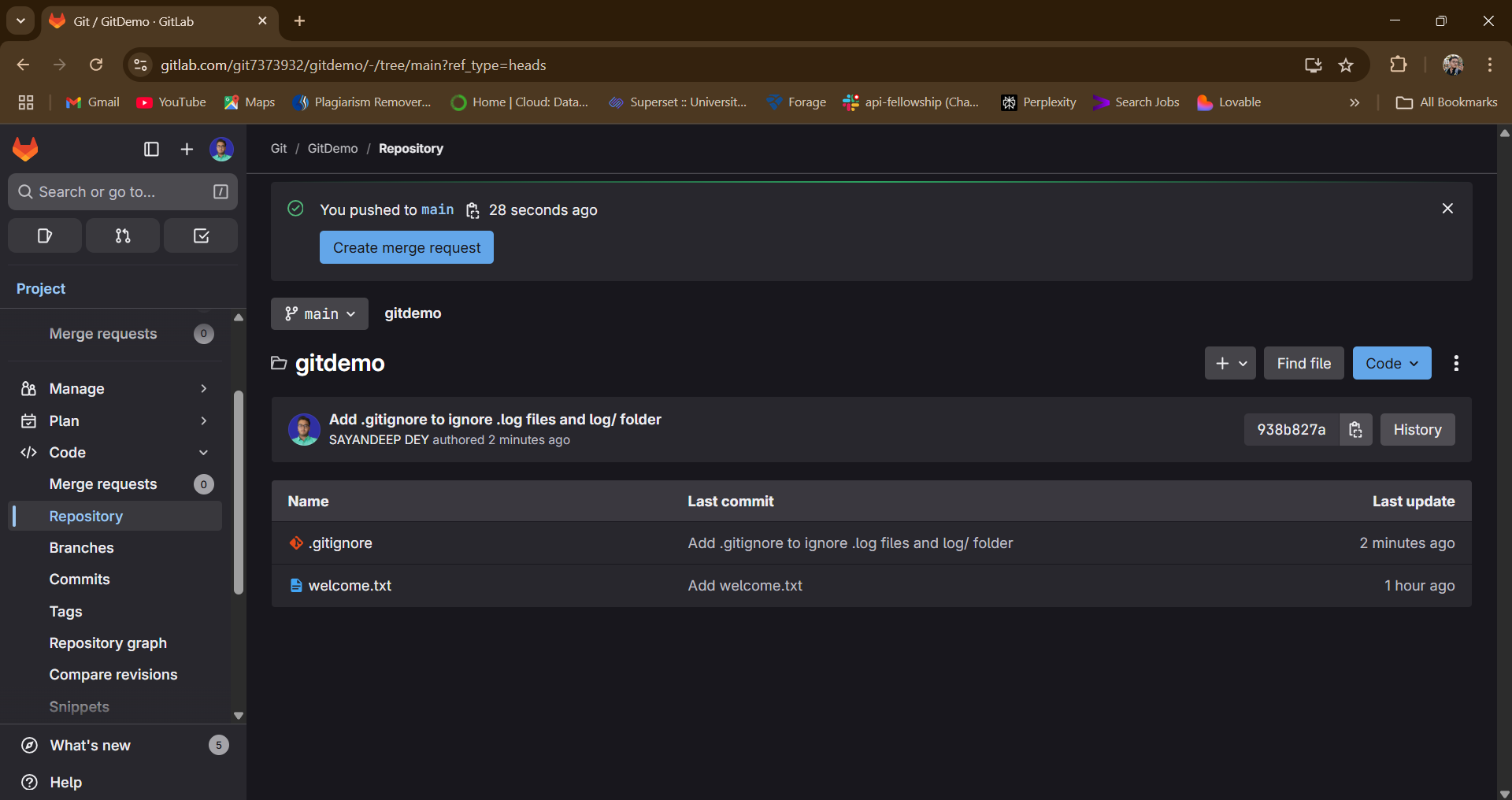
See which patterns are ignoring which files

git check-ignore -v debug.log log/server.log

Push to remote (if local branch name differs, specify both)

git push origin master:main

**Output:**

****

**LAB-3:** **GIT-HOL:-**

**Code:**

# create & switch branch

git checkout -b GitNewBranch

# work, stage, commit

echo "Work" > feature.txt

git add feature.txt

git commit -m "Add feature file"

# push branch to remote

git push -u origin GitNewBranch

# view diffs

git diff TRUNK..GitNewBranch

git difftool TRUNK..GitNewBranch

# merge locally

git checkout TRUNK

git pull origin TRUNK

git merge --no-ff GitNewBranch

# resolve conflicts (if any)

git mergetool

git add <files>

git commit

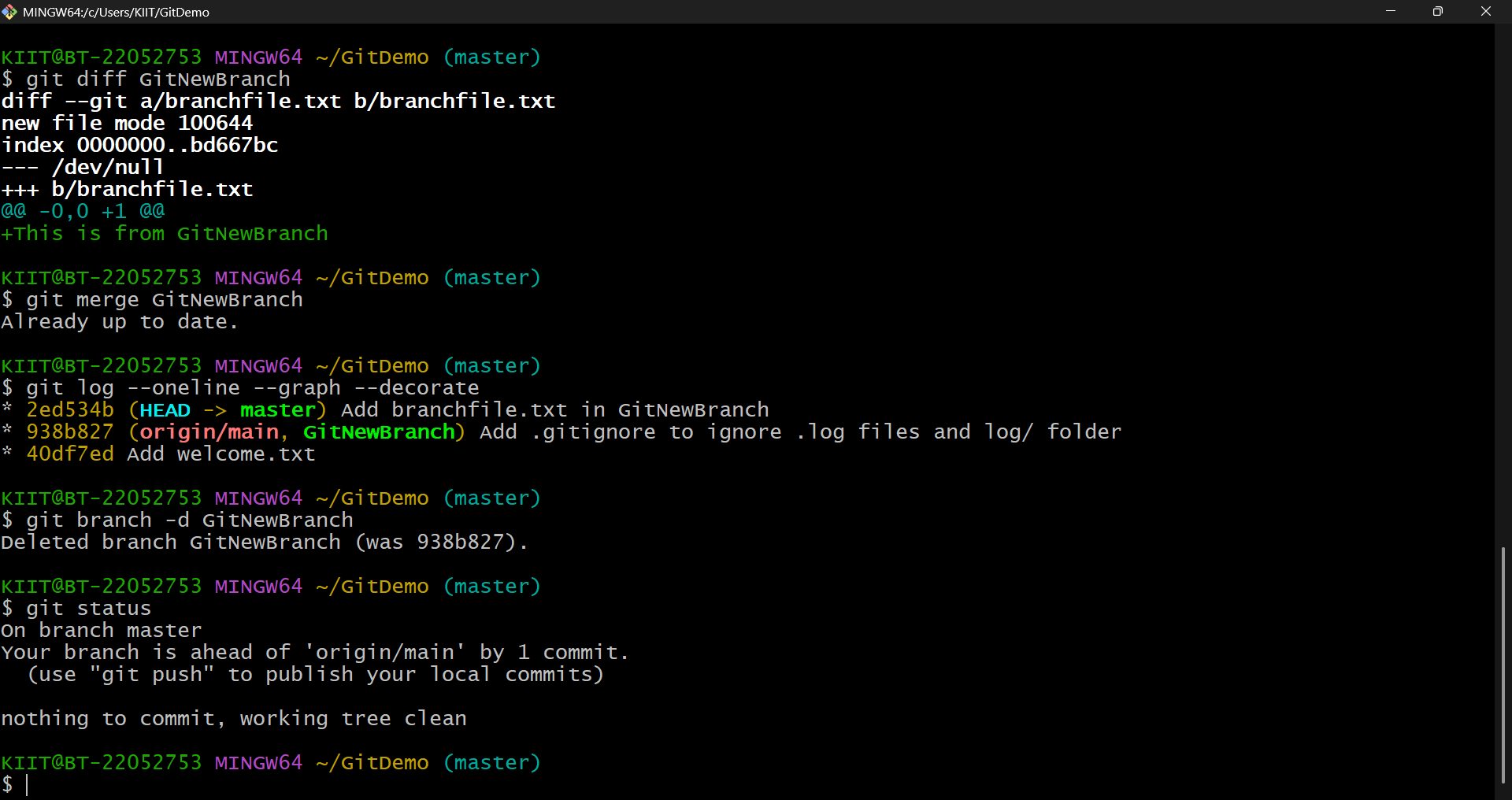
# view log & cleanup

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

git branch -d GitNewBranch

git push origin --delete GitNewBranch

**Output:**

****

**LAB-4 :GIT-HOL:-**

**Code:**

Verify master is clean

git status

Create and switch to branch

git checkout -b GitWork

Create hello.xml in branch

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

git status

Commit branch changes

git add hello.xml

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

Switch to master

git checkout master

Create hello.xml with DIFFERENT content in master

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

Commit master changes

git add hello.xml

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

View log (all branches)

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

Merge branch into master (will cause conflict)

git merge GitWork

Edit hello.xml to manually resolve conflict

(Remove conflict markers <<<<< ===== >>>>> and keep final desired content)

Mark conflict as resolved and commit

git add hello.xml

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

Add backup file to .gitignore (if any)

echo "\*.bak" >> .gitignore

git add .gitignore

git commit -m "Ignore backup files"

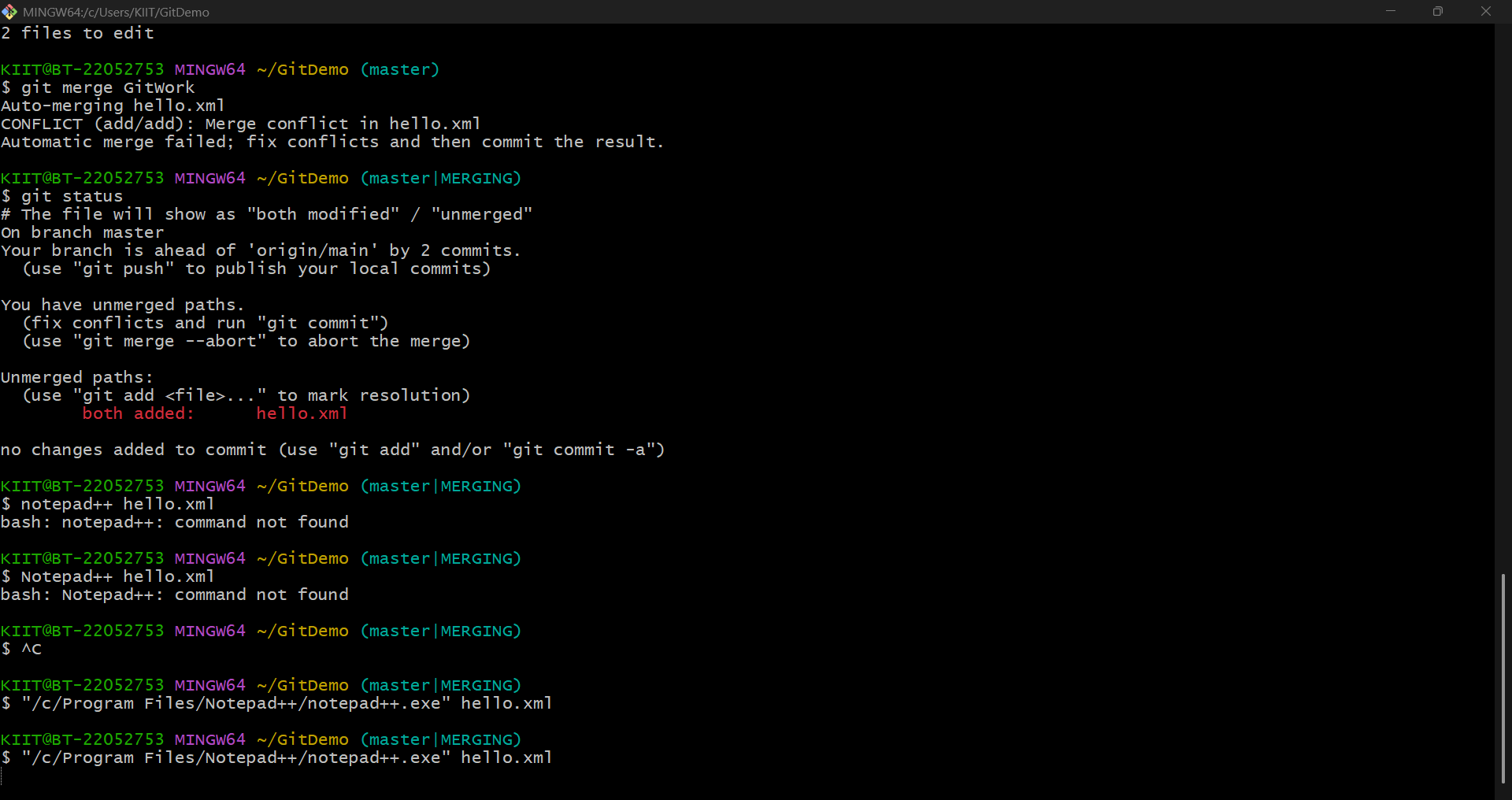
Delete branch after merge

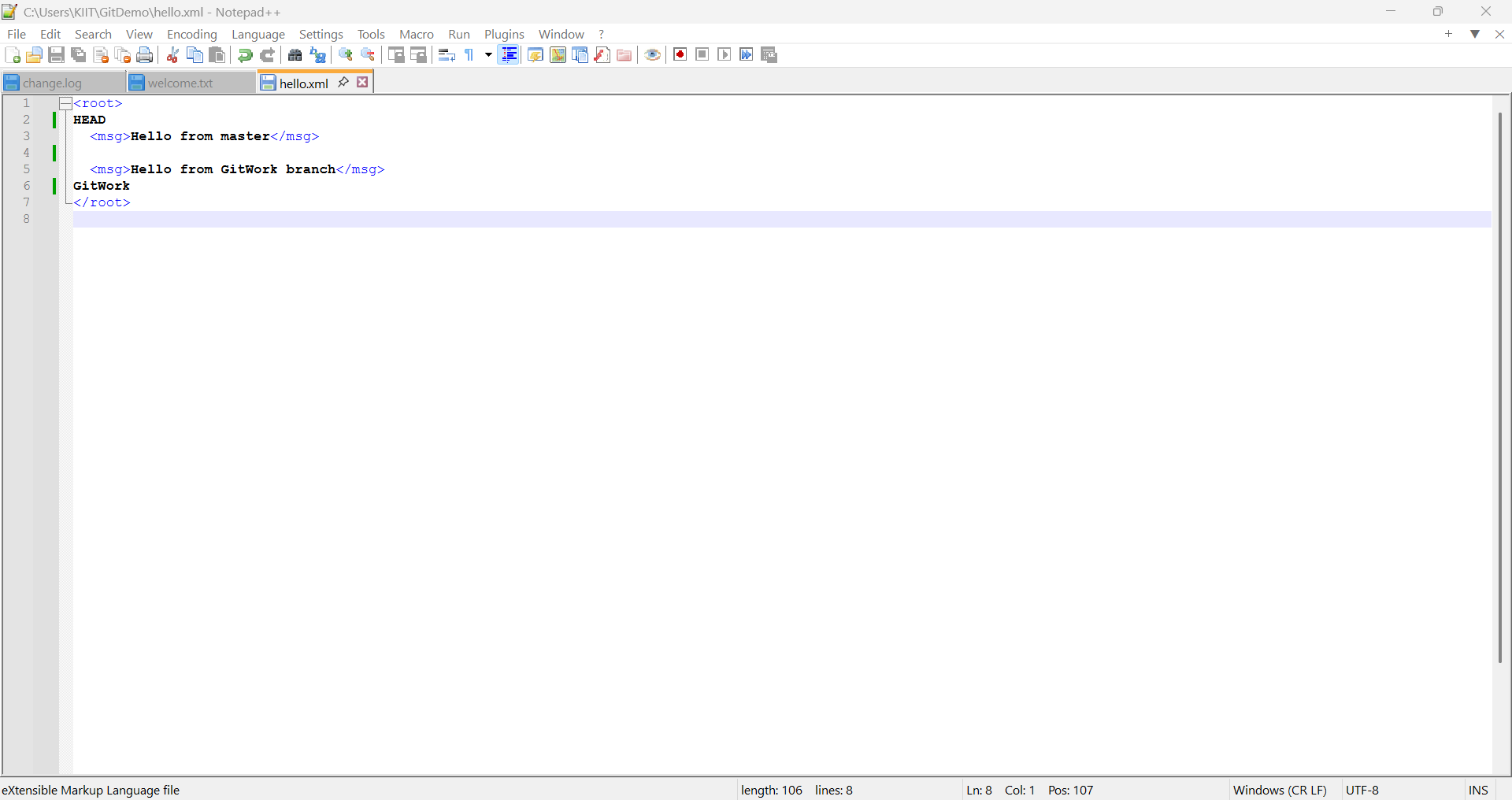
git branch -d GitWork

View final log

git log --oneline --graph –decorate

**Output:**

****

****

**LAB-5 :GIT-HOL:-**

**Code:**

cd ~/GitDemo

git checkout master

git status –porcelain

git add -A

git commit -m "WIP: save changes before push"

git stash push -m "WIP before push"

git remote -v

git remote show origin

git fetch origin

git pull origin master --allow-unrelated-histories

git branch --set-upstream-to=origin/master master

git push -u origin master

git fetch origin

git ls-remote --heads origin

git ls-tree -r origin/master --name-only | sed -n '1,200p'

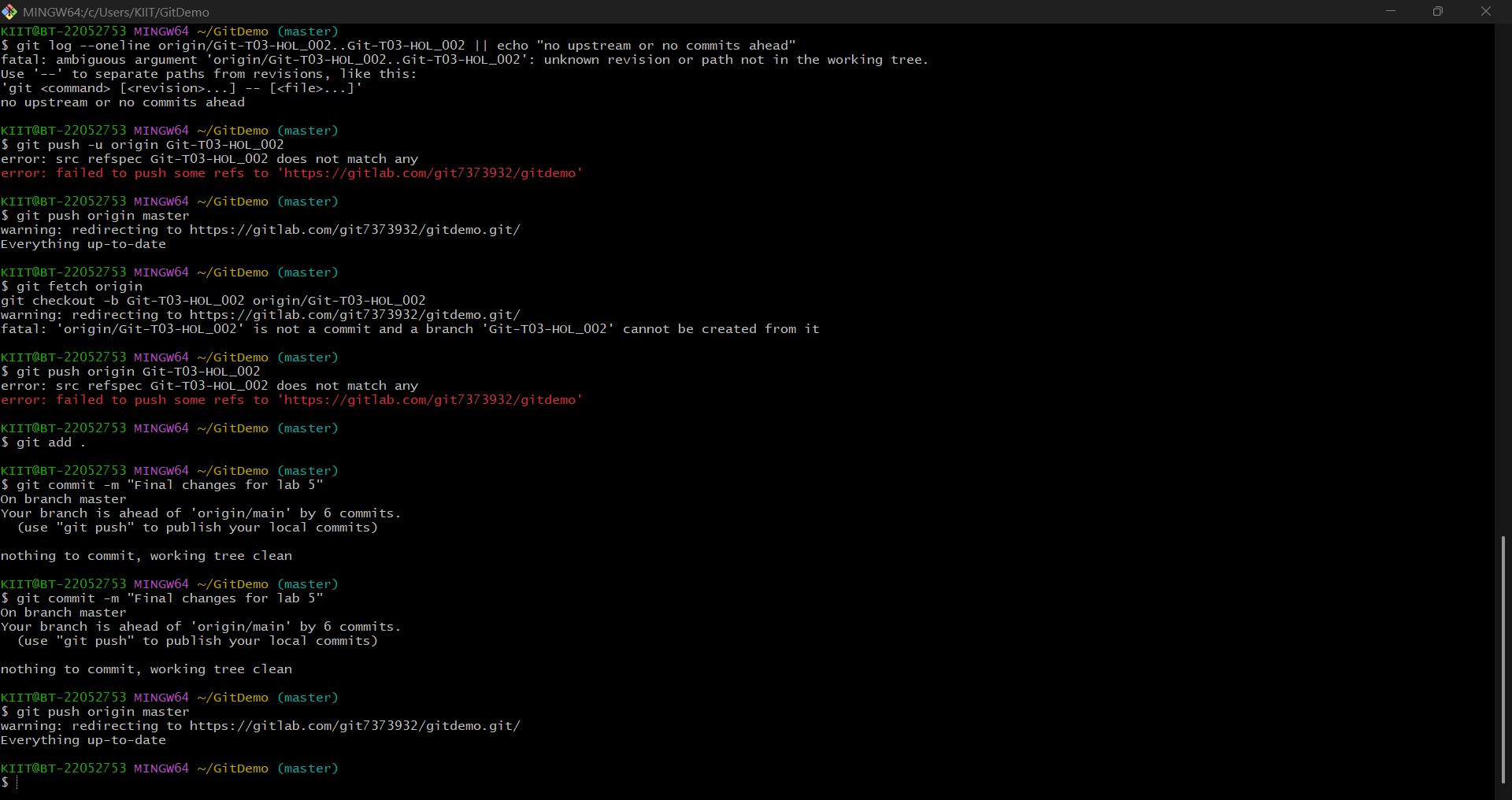
# check specifically for .gitignore

git ls-tree -r origin/master --name-only | grep -E '^\.gitignore$' || echo ".gitignore not found on remote"

git status

git log --oneline --graph --decorate -n 30

**Output:**

****