**WEEK 8 GIT**

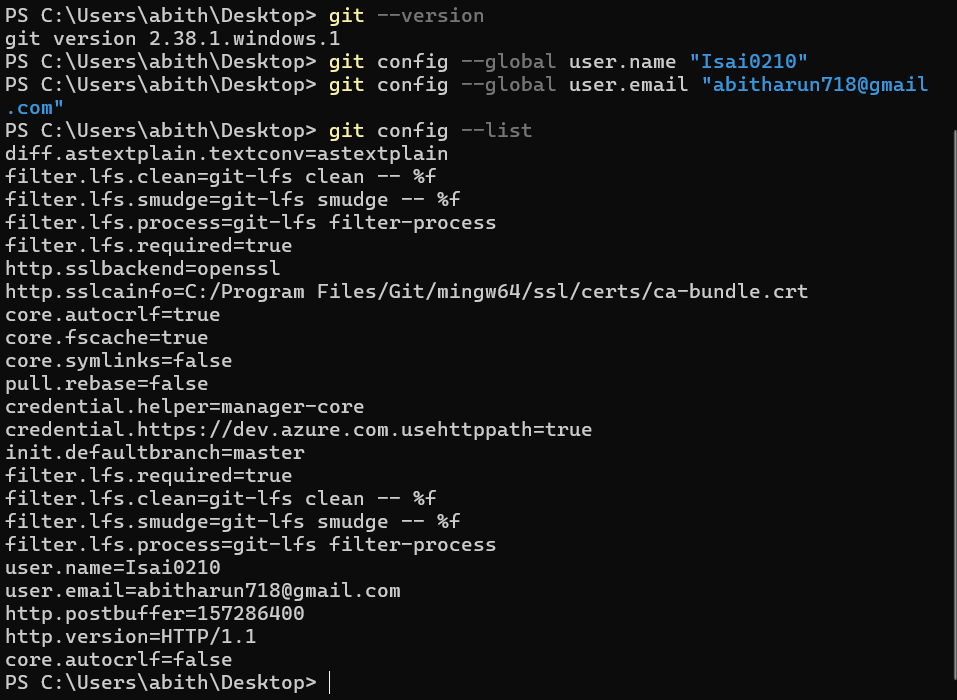
**EXERCISE 1:**

**1. Check if Git is installed:**

**git –version**

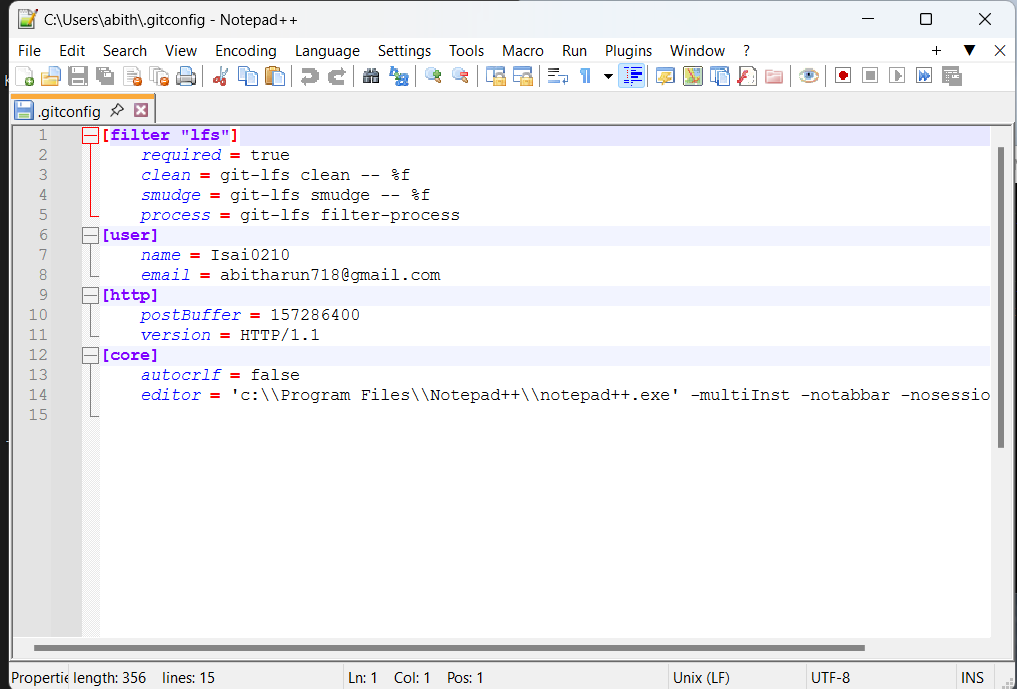
****

**2. Set up Git global user config:**

****

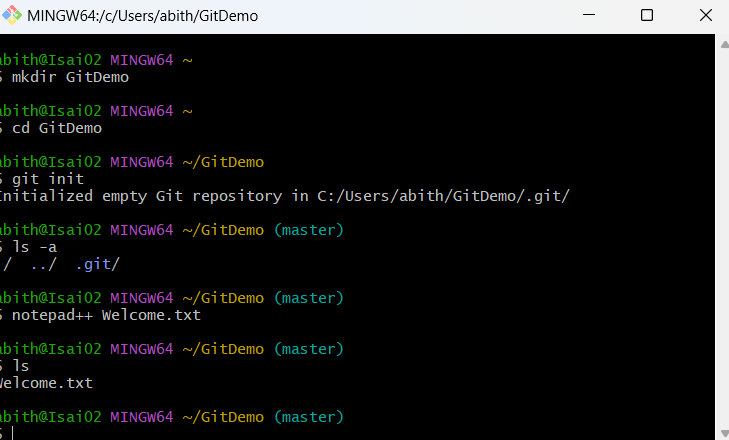
**3. Confirm configuration:**

* **git config –list**

****

**1. Create project folder and initialize Git:**

* **mkdir GitDemo**
* **cd GitDemo**
* **git init**

****

**2. Verify Git repo is initialized:**

* **ls -a**

**3. Create and edit a file:**

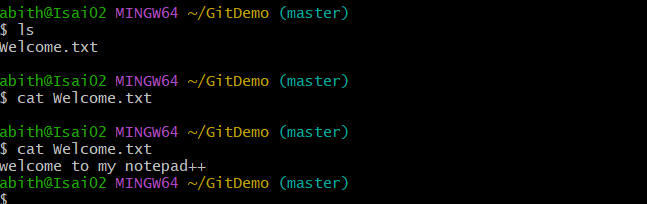
* **notepad++ welcome.txt**

**4. Check the file exists:**

* **ls**

**5. View the content:**

* **cat welcome.txt**

****

**6. Check Git status:**

* **git status**

**7. Stage the file:**

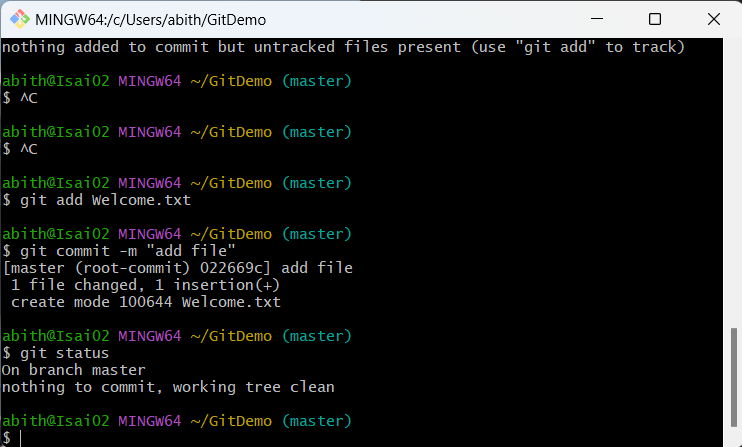
* **git add welcome.txt**

**8. Commit with multiline message (opens Notepad++):**

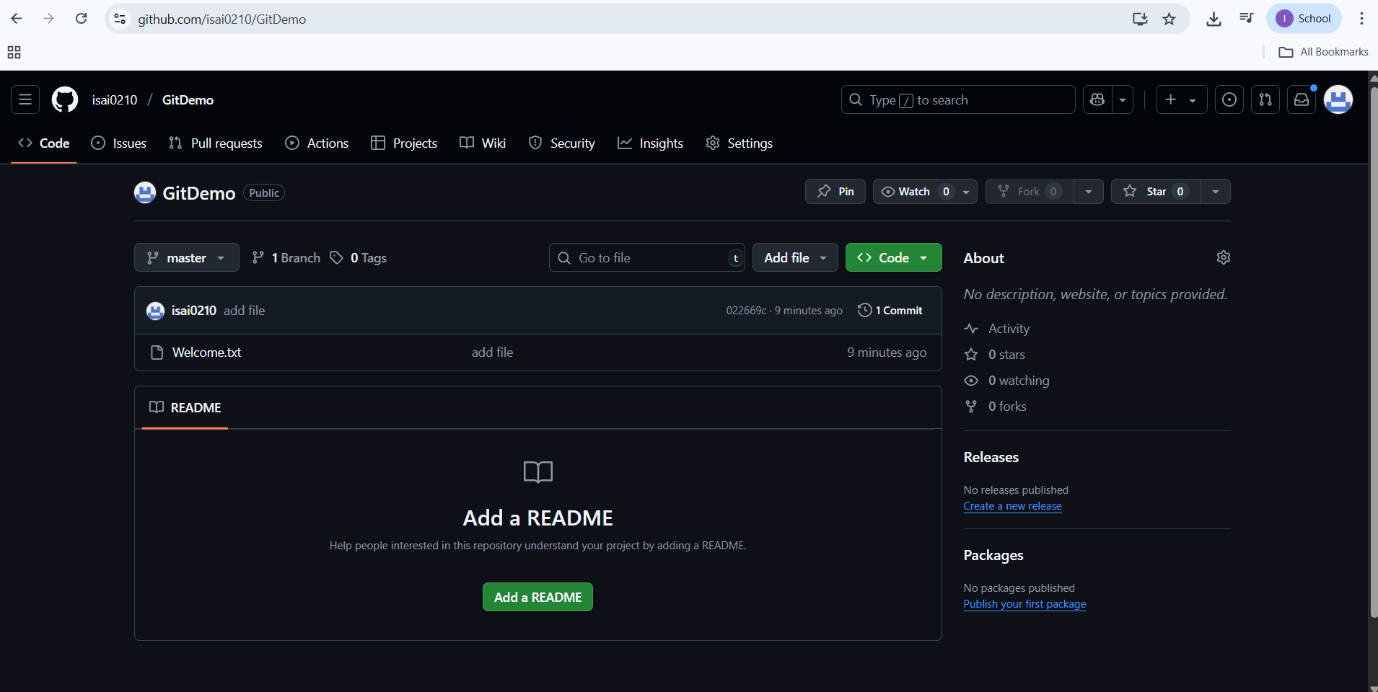
* **git commit**

**9. Confirm file is committed:**

* **git status**

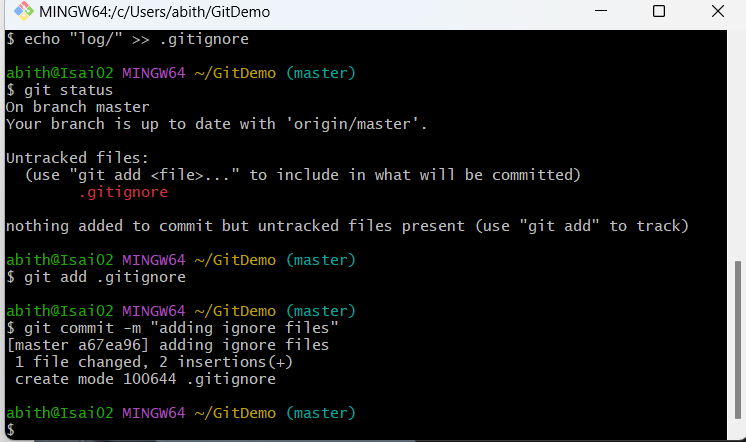
****

**Connect to Remote Repository:**

****

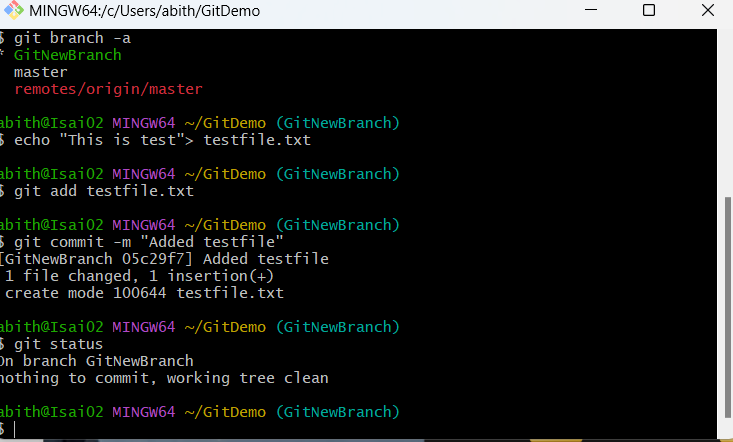
**EXERCISE 2:**

**cd ~/GitDemo**

****

**Create test files and folder to ignore**

* **echo "error log" > error.log**

****

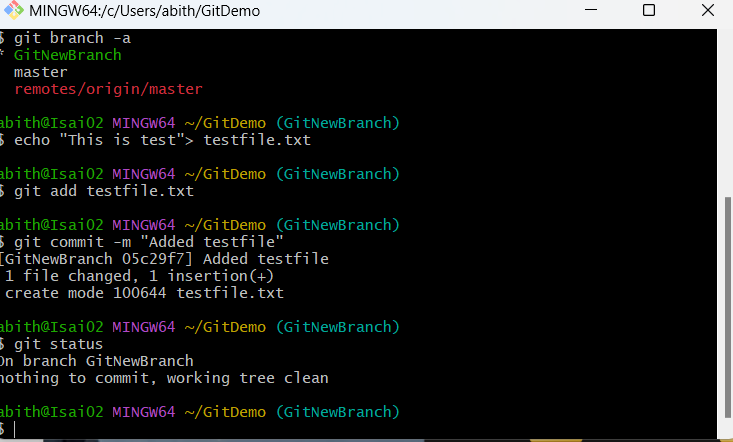
* **mkdir log**
* **echo "log content" > log/system.log**

**Create a .gitignore file**

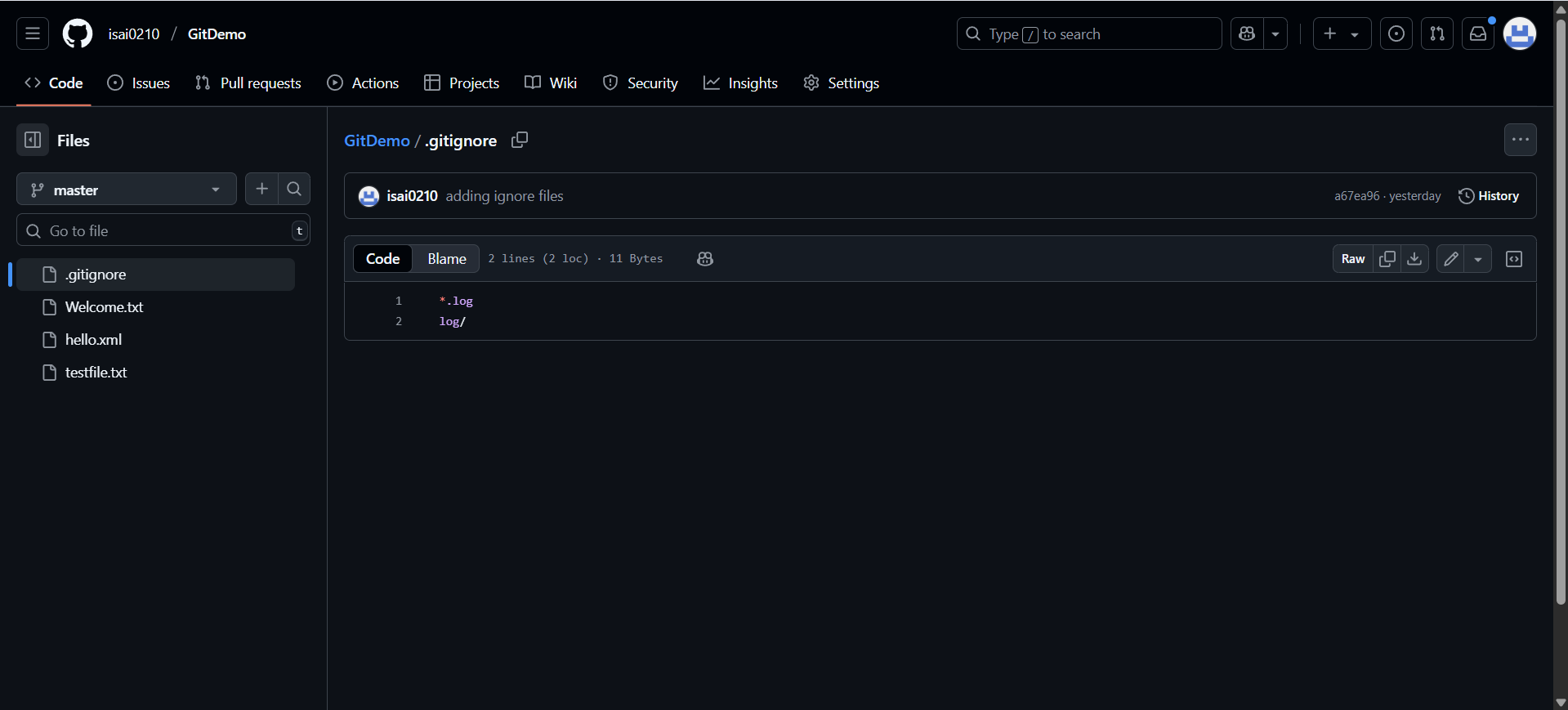
* **touch .gitignore**
* **echo "\*.log" >> .gitignore**
* **echo "log/" >> .gitignore**
* **\*.log**
* **log/**

**Check Git status**

* **git status**

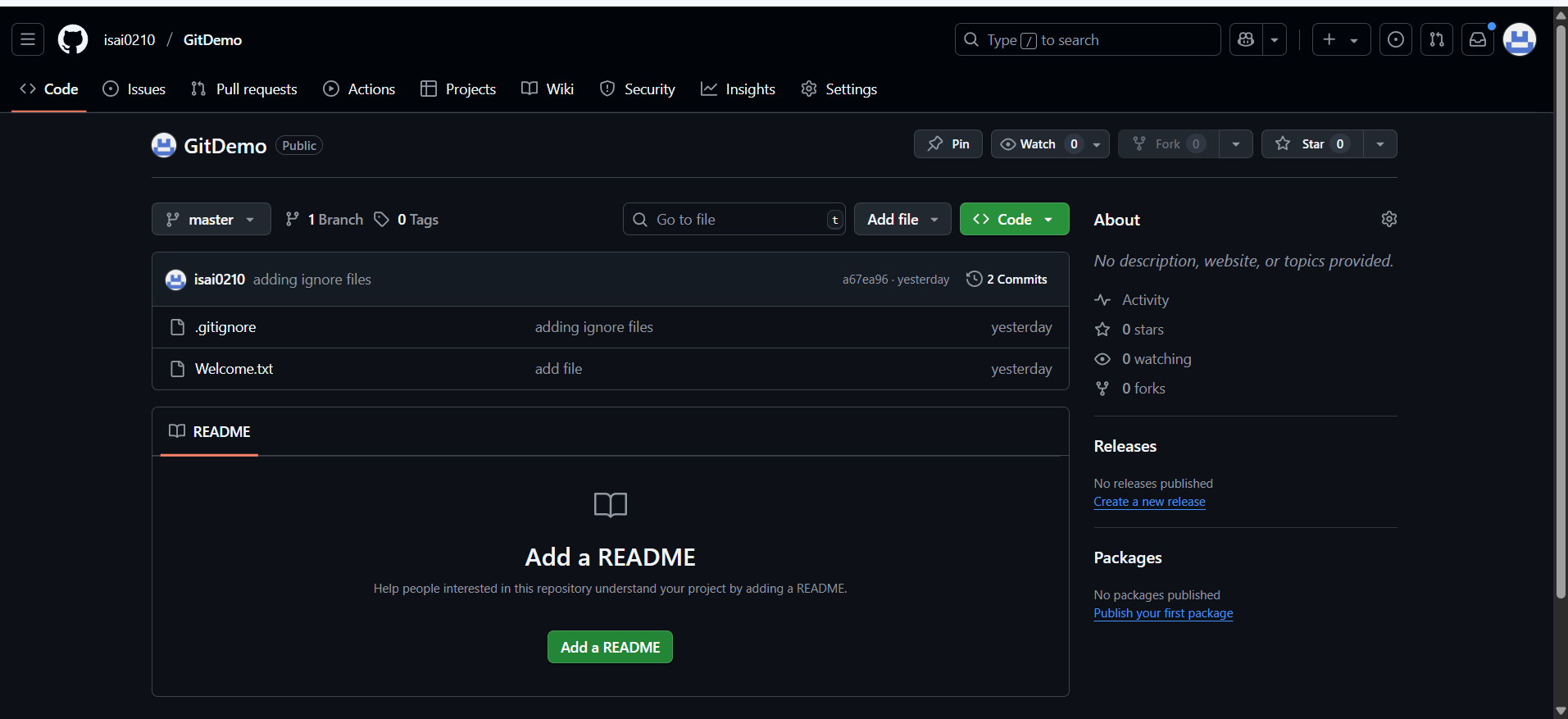
****

**Add and commit only .gitignore**

****

**git add .gitignore**

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

****

**EXERCISE 3:**

**Create a new branch called GitNewBranch**

* **git checkout -b GitNewBranch**

****

**List all branches**

* **git branch**

**Add a new file in this branch**

* **echo "This is a new feature file" > feature.txt**

**Stage and commit the file**

* **git add feature.txt**
* **git commit -m "Add feature.txt in GitNewBranch"**

**Check branch status**

* **git status**

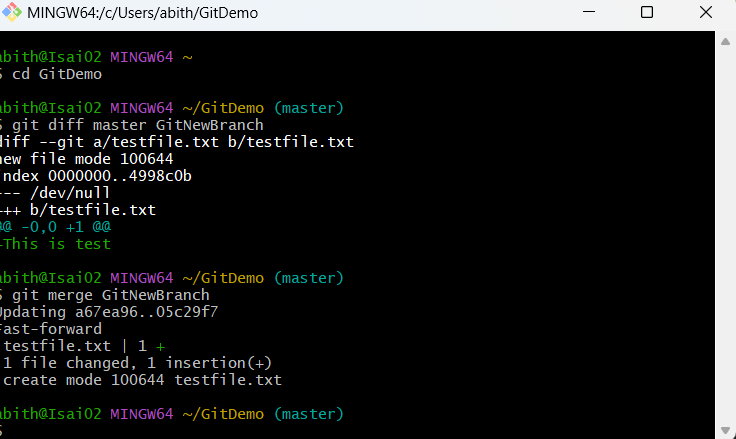
**MERGING**

**Switch back to the master branch**

* **git checkout master**

**Check differences between master and GitNewBranch**

* **git diff master..GitNewBranch**

****

**Merge GitNewBranch into master**

* **git merge GitNewBranch**

**Delete the merged branch**

* **git branch -d GitNewBranch**
* **git status**

****

**EXERCISE 4:**

**Create a new branch GitWork and switch to it**

* **git checkout -b GitWork**

**Add a file hello.xml and update its content**

* **echo "<message>Hello from GitWork branch</message>" > hello.xml**

**Stage and commit the file**

* **git add hello.xml**
* **git commit -m "Added hello.xml from GitWork branch"**

**Switch back to master**

* **git checkout master**

****

**Add a different version of hello.xml in master**

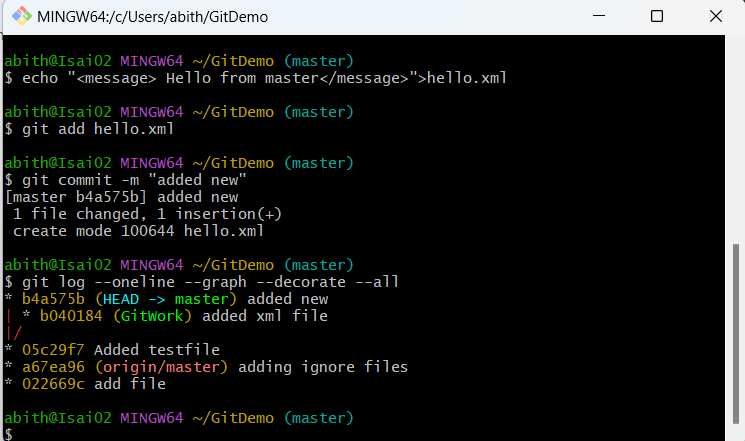
* **echo "<message>Hello from master branch</message>" > hello.xml**

**Commit changes in master**

* **git add hello.xml**
* **git commit -m "Added hello.xml from master branch"**

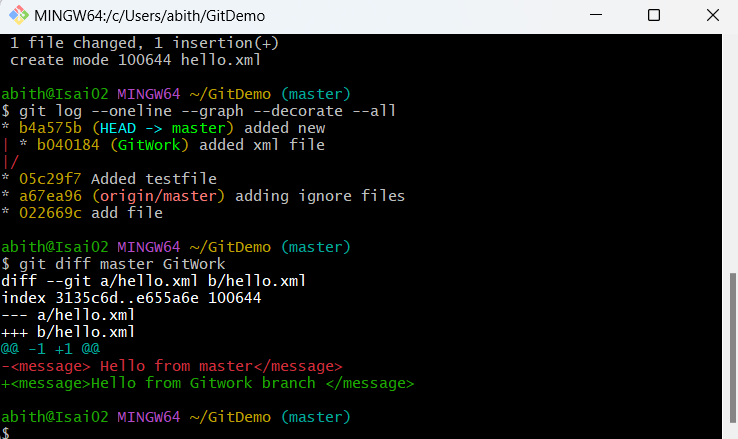
**View logs in graph format**

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

****

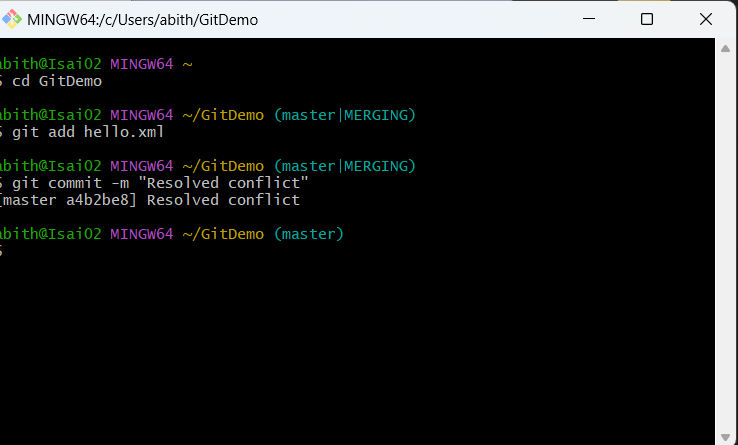
**Check differences using CLI tool**

* **git diff master GitWork**

****

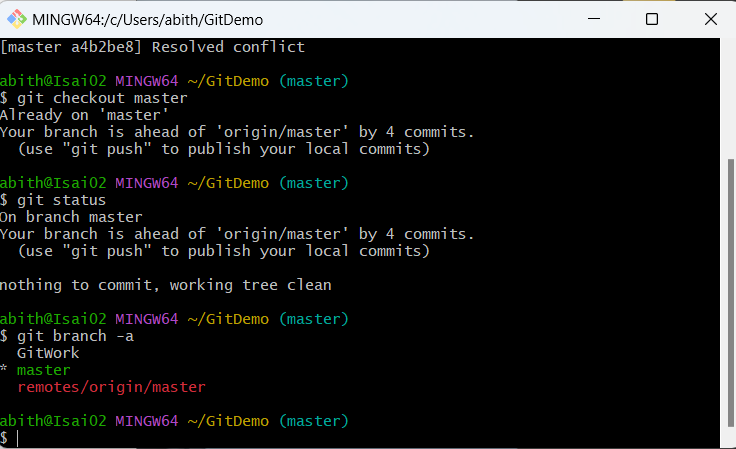
**Merge GitWork into master**

* **git merge GitWork**

****

**Commit after resolving conflict**

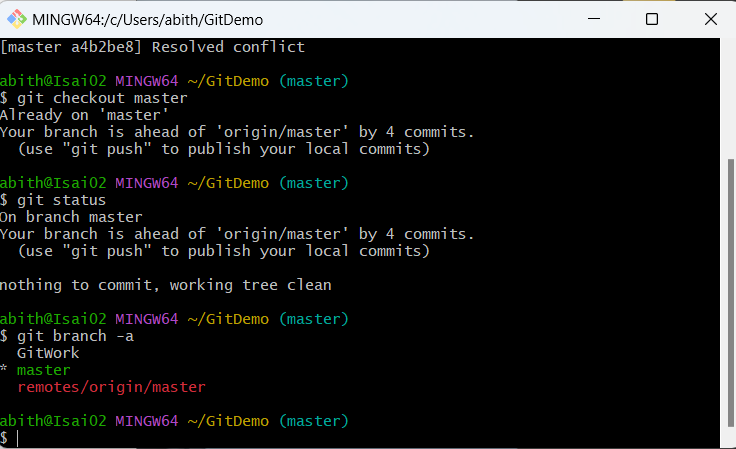
* **git add hello.xml**
* **git commit -m "Resolved merge conflict in hello.xml"**

****

**EXERCISE 5:**

**Verify if master is in clean state**

* **git checkout master**
* **git status**

****

**List all available branches**

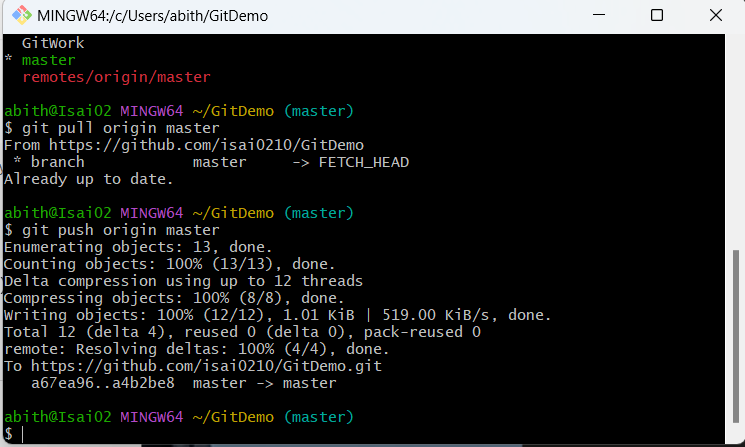
* **git branch -a**

**Pull latest changes from remote**

* **git pull origin master**

**Push your local commits to remote**

* **git push origin master**

****

