WEEK8\_GIT\_HANDSON

**Question : 01**

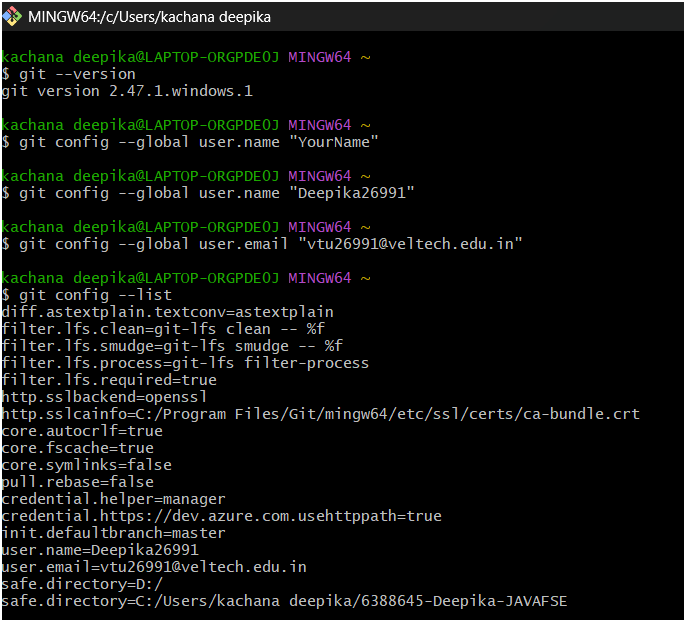
Use basic Git commands (init, status, add, commit, push, pull)

* Configure Git on your system
* Connect your local repository to a remote GitLab repository

**Git Configuration**

**Check if Git is installed:**

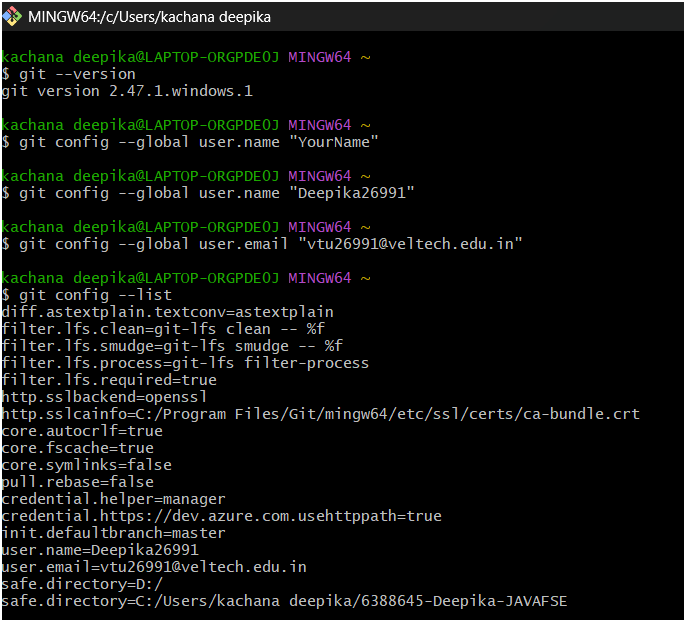
git --version



**Configure your user name and email:**

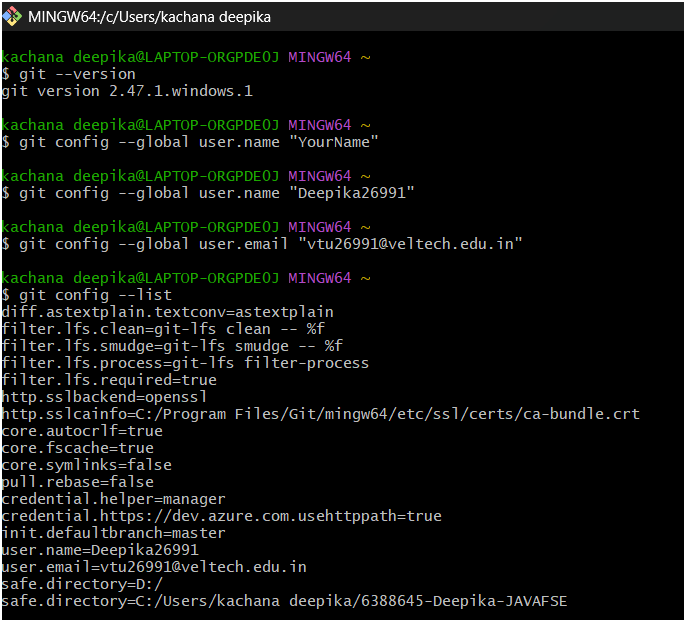
git config --global user.name "Deepika26991"

git config --global user.email [vtu26991@veltech.edu.in](mailto:vtu26991@veltech.edu.in)



**Verify configuration:**

git config --global –list



**Create and Track a File in Git**

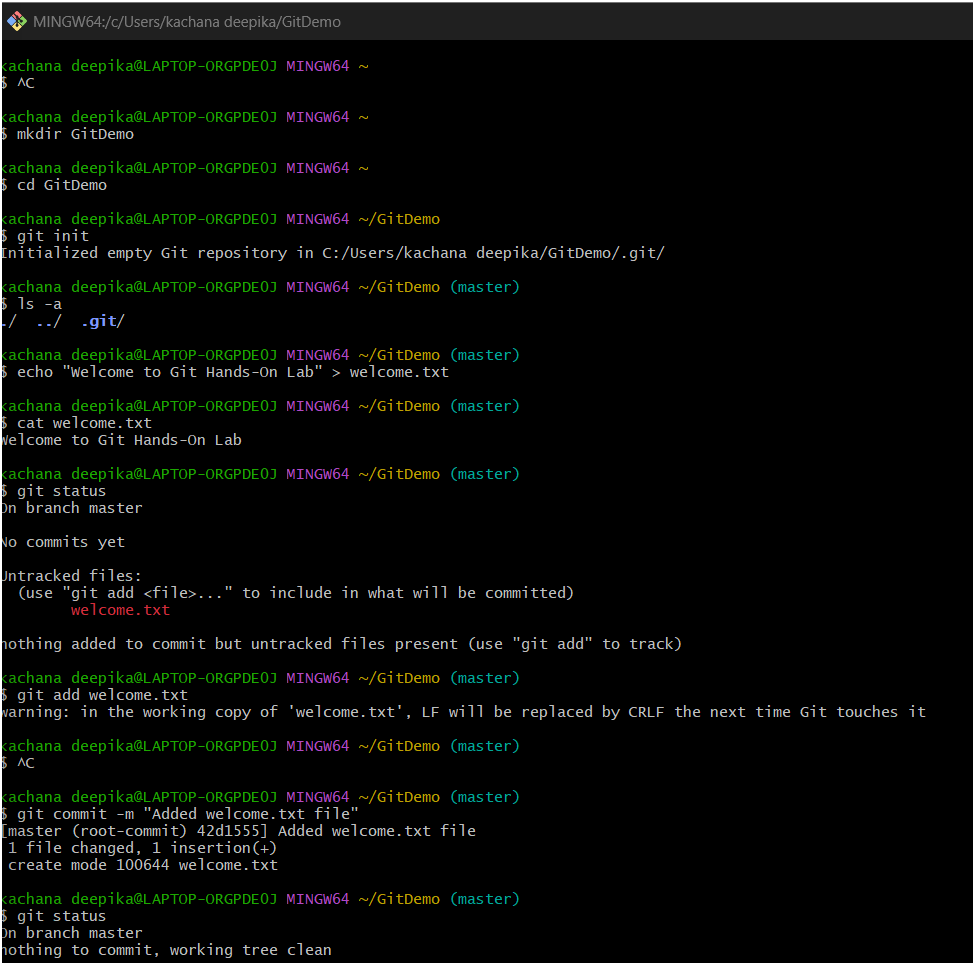
* **Create a project folder:**

mkdir GitDemo

cd GitDemo

* **Initialize Git repo:**

git init



* **Confirm Git is tracking this folder:**

ls -a

.git folder should appear.

* **Create a new file:**

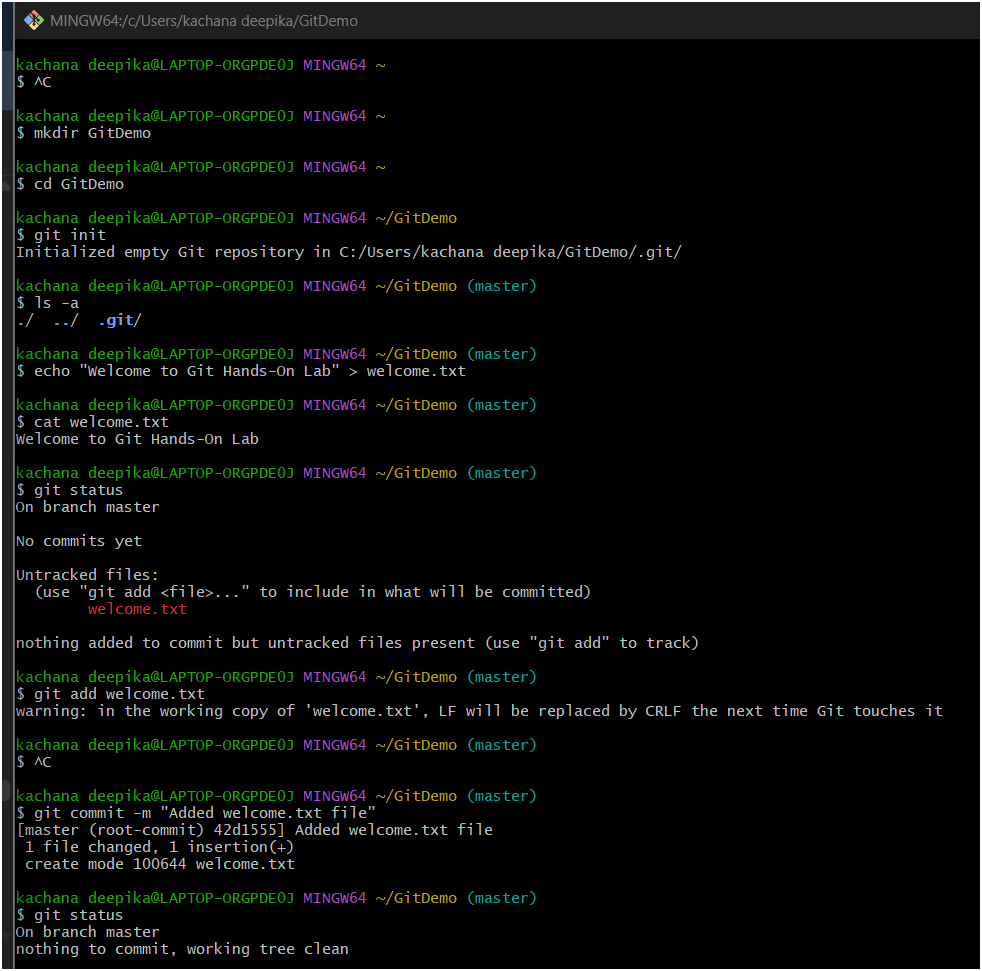
echo "Welcome to Git Hands-on Lab" > welcome.txt

* **Confirm the file exists:**

ls

* **View file content:**

cat welcome.txt



* **Check file status:**

git status

* **Stage the file:**

git add welcome.txt

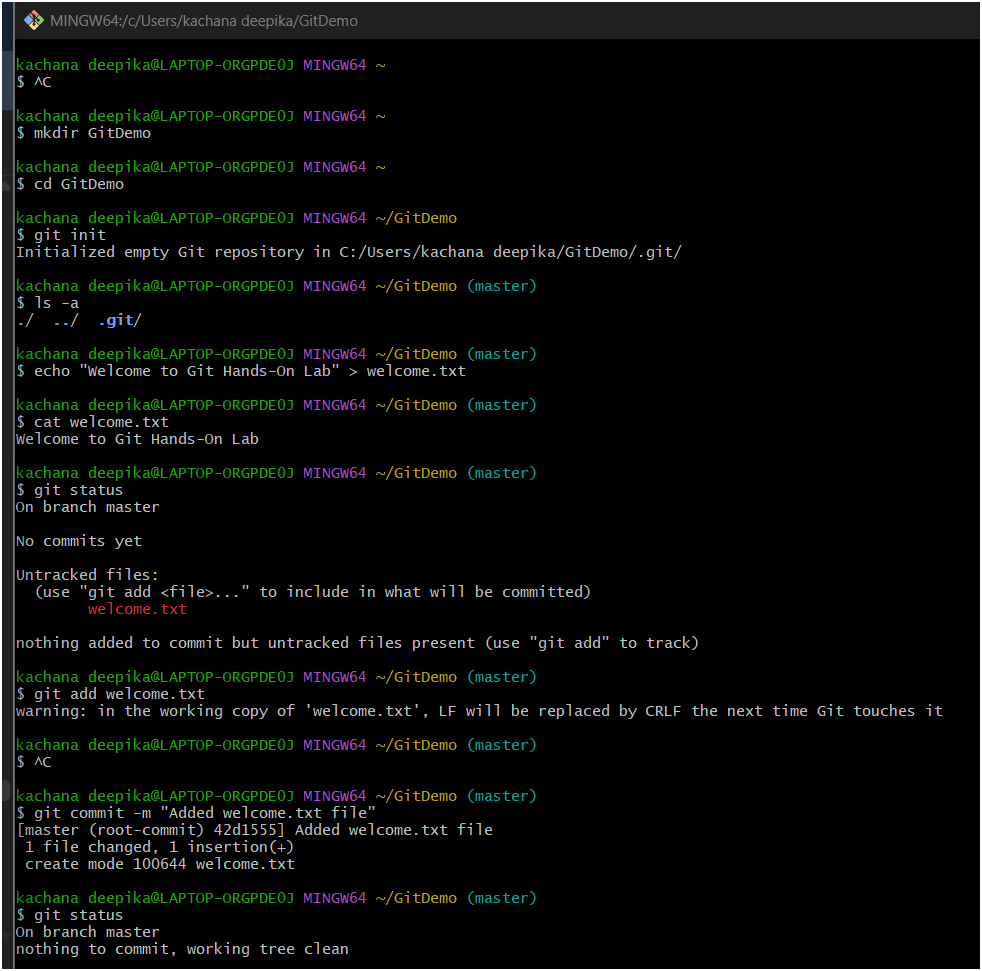
* **Commit the file:**

git commit -m "Added welcome.txt file"

* **Verify status:**

git status

Should show "nothing to commit" if successful.



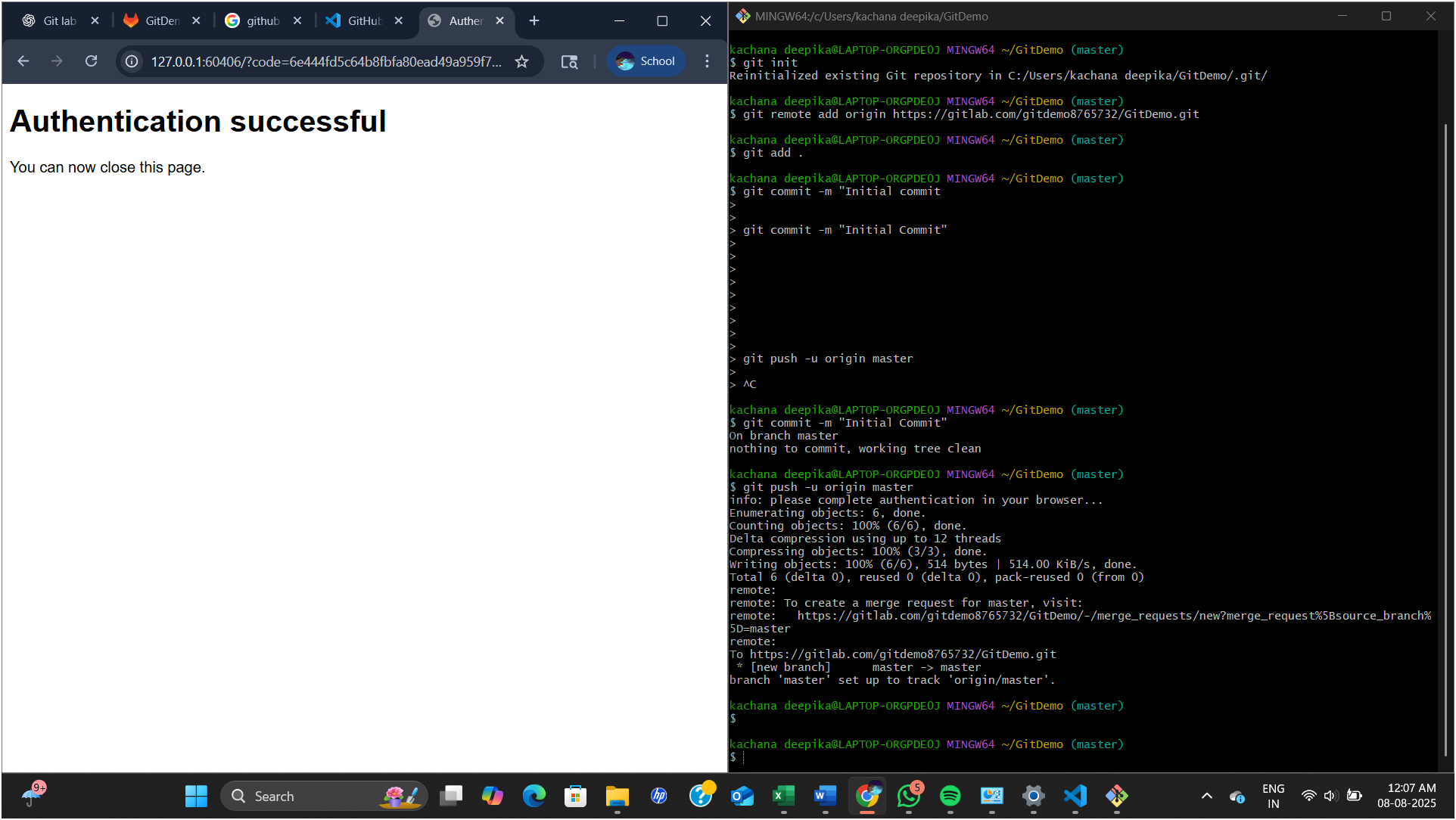
**Connect to Remote GitLab Repository**

* **Create a new project on GitLab called GitDemo**

No README file, keep it empty.

* **Add the remote repository**

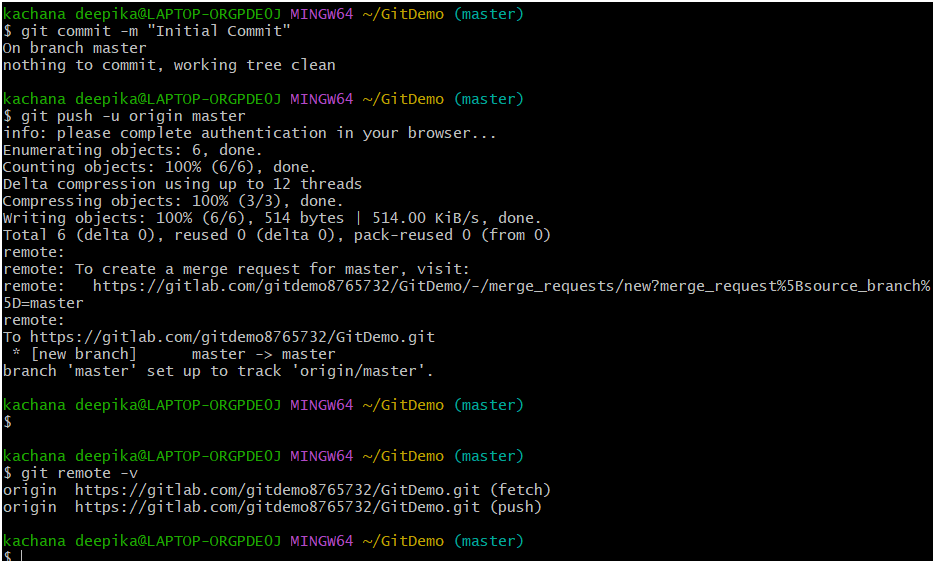
git remote add origin https://gitlab.com/gitdemo8765732/GitDemo.git

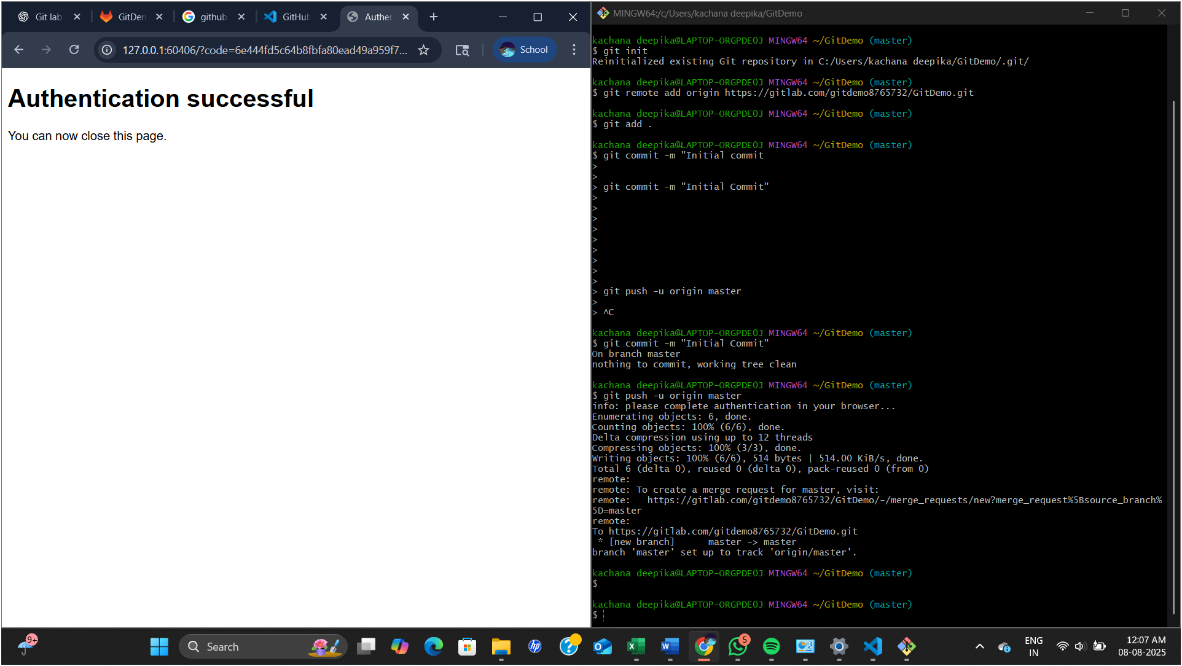


* **Push local commits to remote:**

git push -u origin master

* Log in to GitLab
* Open your **GitDemo** project
* You should see welcome.txt and your latest commit





**Question :02**

• Explain git ignore

• Explain how to ignore unwanted files using git ignore In this hands-on lab, you will learn how to: • Implement git ignore command to ignore unwanted files and folders

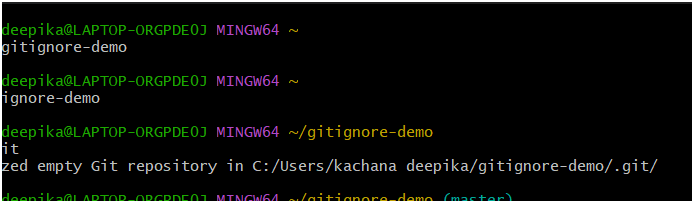
**Step-by-Step Instructions**

**Step 1: Create a Git Repository (if not already created)**

mkdir gitignore-demo

cd gitignore-demo

git init

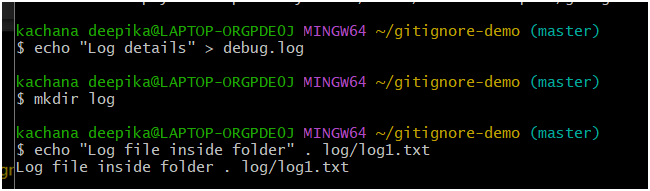


**Step 2: Create Unwanted Files/Folders**

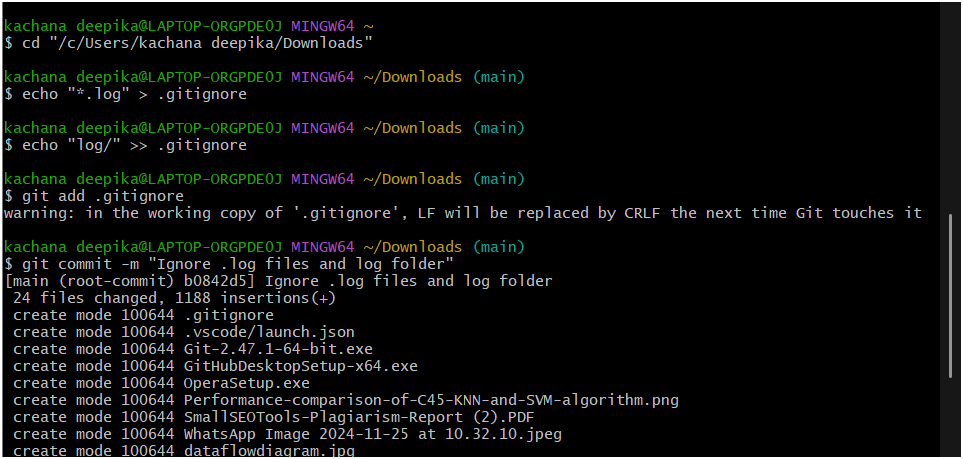
echo "This is a sample log" > debug.log

mkdir log

echo "Log folder file" > log/log1.txt



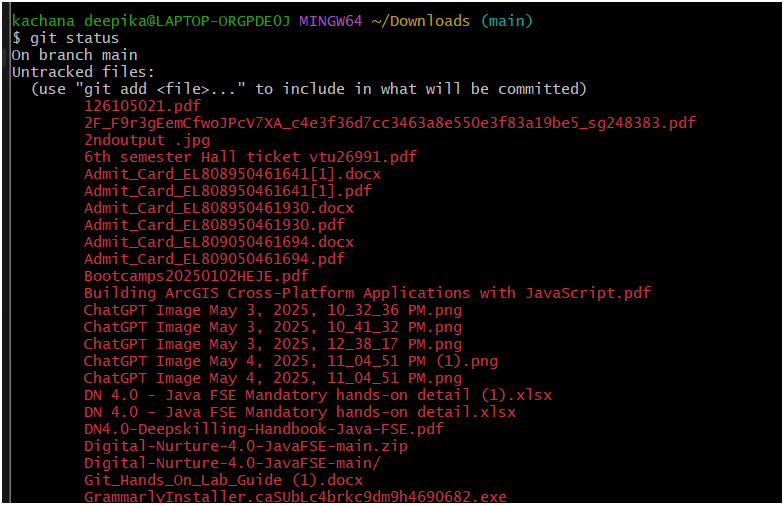
**Step 3: Create .gitignore File**



**Step 4: Check git status**

git status

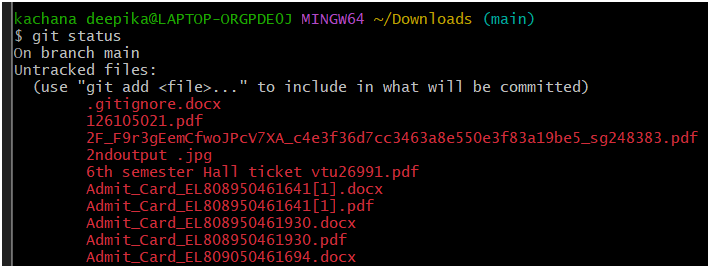
You should see that .gitignore is untracked, but debug.log and log/ are not shown — meaning Git is **ignoring them correctly**.



**Step 5: Add & Commit .gitignore**

git add .gitignore

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



**Example : 03**

**Objectives**

* Explain branching and merging
* Explain about creating a branch request in GitLab
* Explain about creating a merge request in GitLab

In this hands-on lab, you will learn how to:

* Construct a branch, do some changes in the branch, and merge it with master (or trunk)

**Step by Execution :**

**Create a new branch**

git branch GitNewBranch

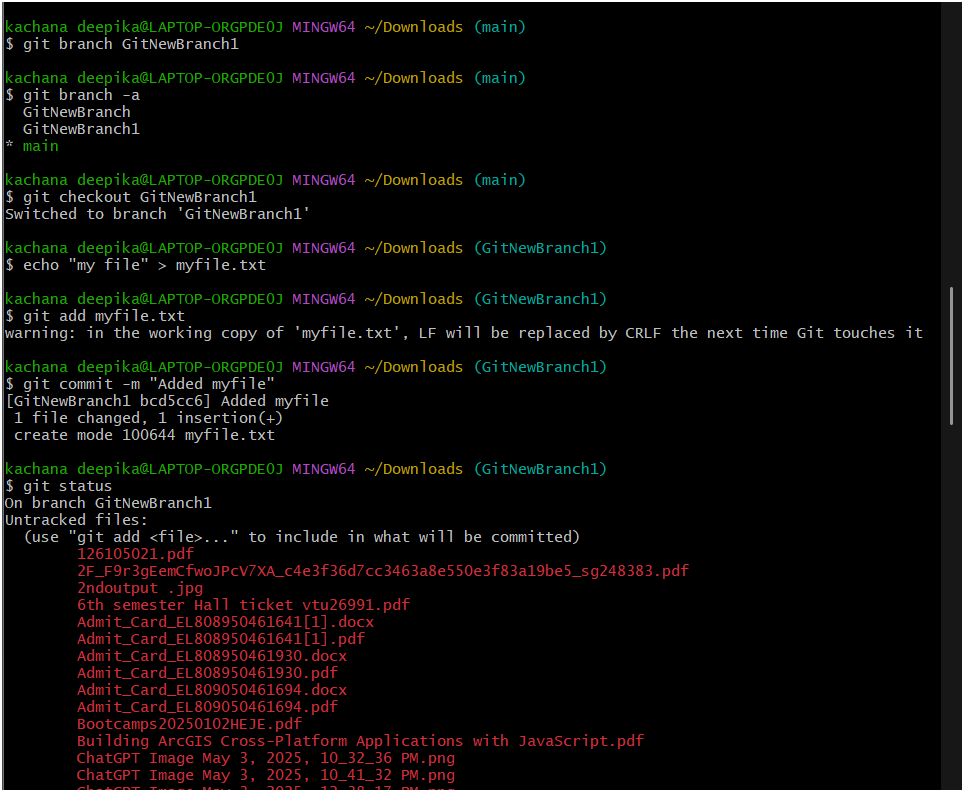
**List all branches** (local and remote)

git branch -a

The \* shows your current branch

**Switch to your new branch**

git checkout GitNewBranch



**Add a file and some content**

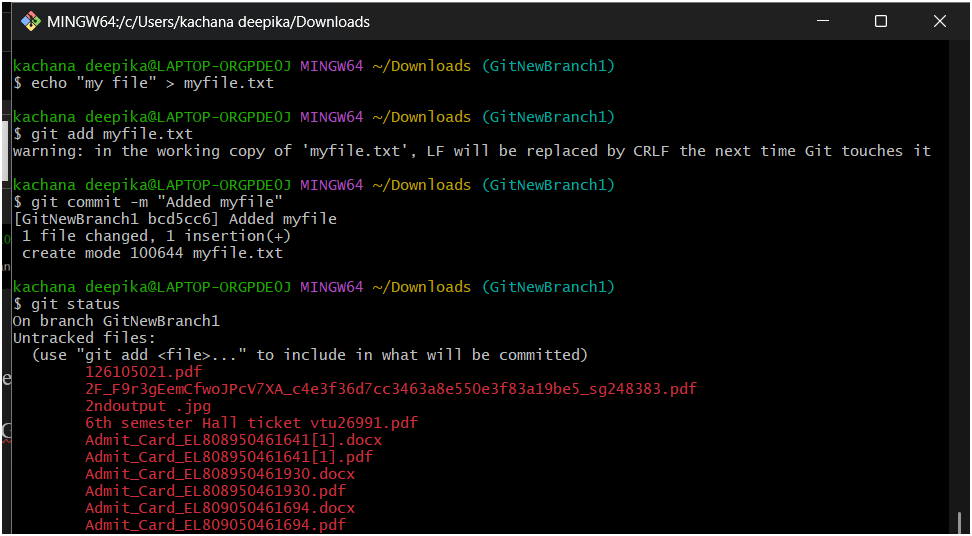
echo "This is my new branch content" > myfile.txt

git add myfile.txt

git commit -m "Add myfile.txt in GitNewBranch"

**Check the status**

git status



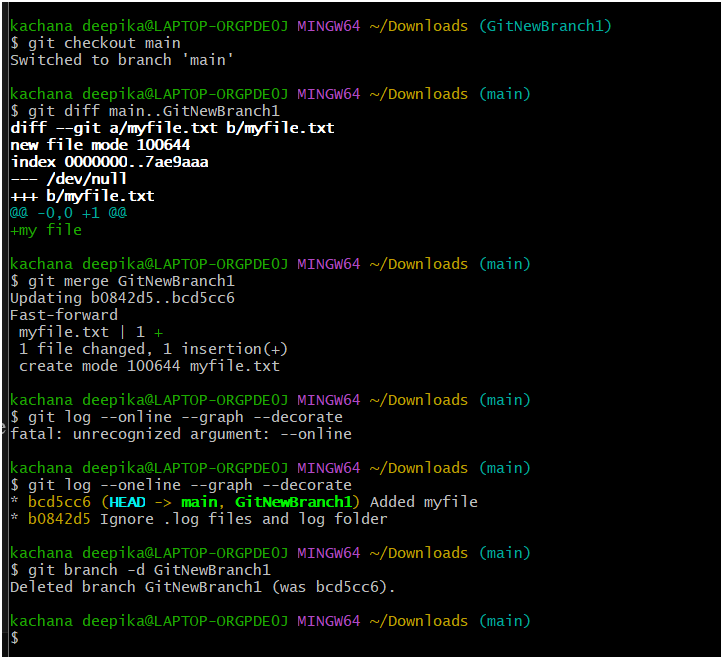
**Switch to your new branch**

git checkout GitNewBranch1

**Merging**

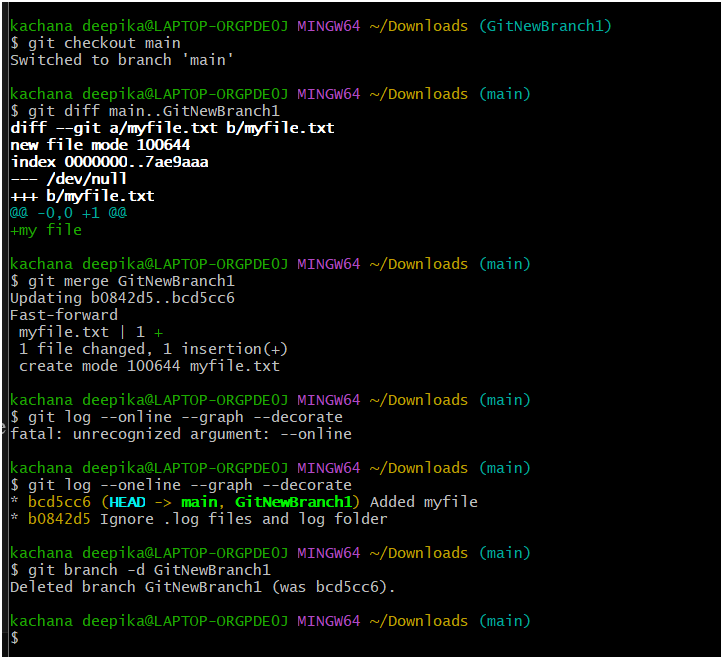
**Switch back to master (or main)**

git checkout main



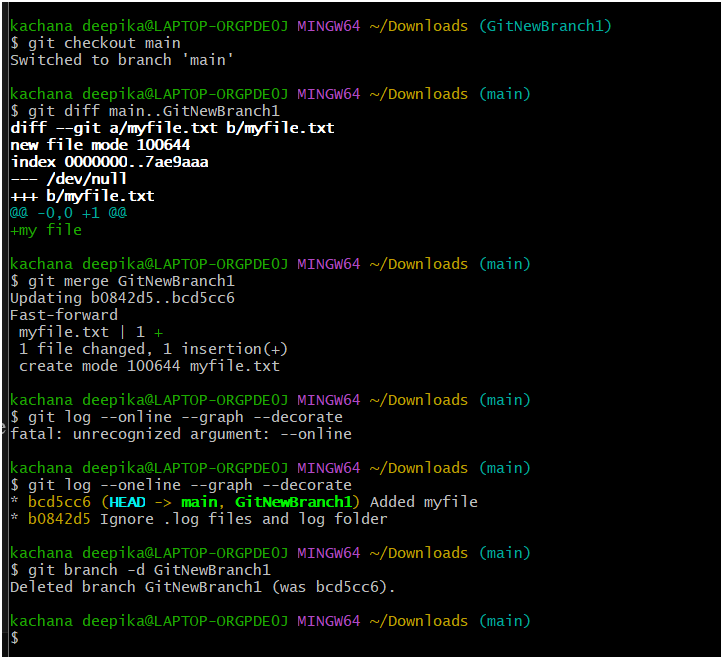
**List differences (text output)**

git diff main..GitNewBranch1



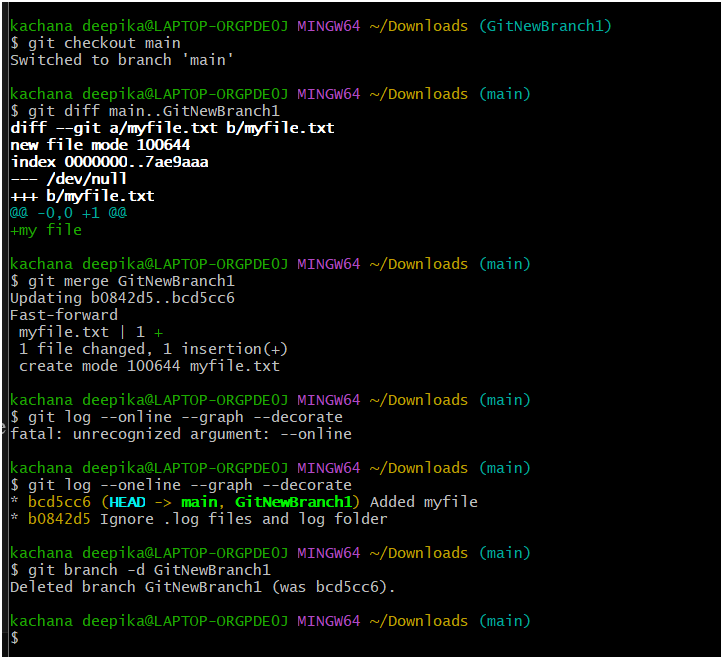
**Merge the branch into main**

git merge GitNewBranch1



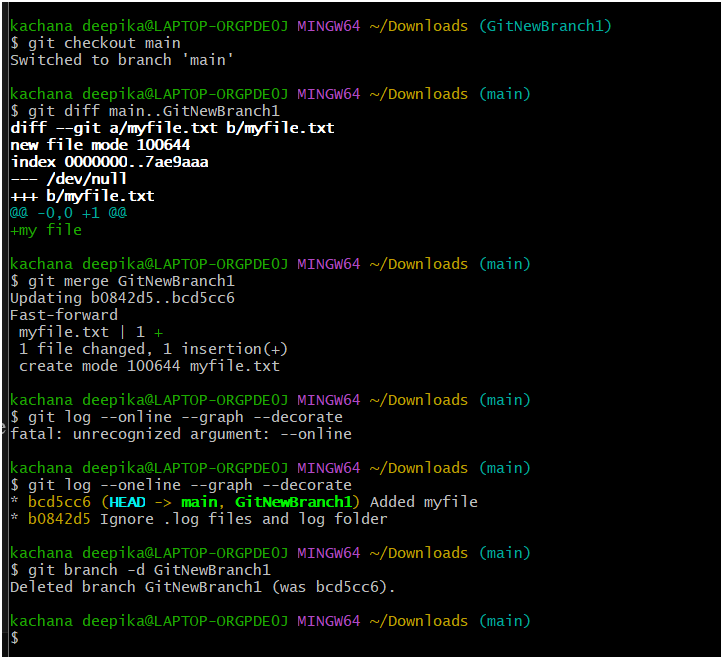
**See commit history visually**

git log --oneline --graph –decorate



**Delete the merged branch**

git branch -d GitNewBranch



**Check status again**

git status

