**1. Familiar with Git commands like git init, git status, git add, git commit, git push, and git pull.**

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

* Setup your machine with Git Configuration
* Integrate notepad++.exe to Git and make it a default editor
* Add a file to source code repository

chigu@Kp MINGW64 ~

$ git version

git version 2.50.0.windows.1

chigu@Kp MINGW64 ~

$ git config --global user.name "ChigurupatiRishitha"

git config --global user.email "rishitha.chigurupati.email@cognizant.com"

chigu@Kp MINGW64 ~

$ git config --global --list

core.editor="C:\Users\chigu\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait

user.name=ChigurupatiRishitha

user.email=rishitha.chigurupati.email@cognizant.com

chigu@Kp MINGW64 ~

$ notepad++

chigu@Kp MINGW64 ~

$ git config --global core.editor "notepad++.exe -multiInst -nosession"

chigu@Kp MINGW64 ~

$ git config --global -e

chigu@Kp MINGW64 ~

$ git init GitDemo

Initialized empty Git repository in C:/Users/chigu/GitDemo/.git/

chigu@Kp MINGW64 ~

$ cd GitDemo

chigu@Kp MINGW64 ~/GitDemo (main)

$ echo "Welcome to the version control" >> welcome.txt

chigu@Kp MINGW64 ~/GitDemo (main)

$ ls -al

total 21

drwxr-xr-x 1 chigu 197609 0 Aug 7 00:38 **.**/

drwxr-xr-x 1 chigu 197609 0 Aug 7 00:38 **..**/

drwxr-xr-x 1 chigu 197609 0 Aug 7 00:38 **.git**/

-rw-r--r-- 1 chigu 197609 31 Aug 7 00:38 welcome.txt

chigu@Kp MINGW64 ~/GitDemo (main)

$ cat welcome.txt

Welcome to the version control

chigu@Kp MINGW64 ~/GitDemo (main)

$ git status

On branch main

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

welcome.txt

nothing added to commit but untracked files present (use "git add" to track)

chigu@Kp MINGW64 ~/GitDemo (main)

$ git add welcome.txt

warning: in the working copy of 'welcome.txt', LF will be replaced by CRLF the next time Git touches it

chigu@Kp MINGW64 ~/GitDemo (main)

$ git add welcome.txt

chigu@Kp MINGW64 ~/GitDemo (main)

$ git commit

[main (root-commit) 7bbf320] Initial commit - Added welcome.txt - Contains intro message

1 file changed, 1 insertion(+)

create mode 100644 welcome.txt

chigu@Kp MINGW64 ~/GitDemo (main)

$ git status

On branch main

nothing to commit, working tree clean

chigu@Kp MINGW64 ~/GitDemo (main)

$ git remote add origin https://gitlab.com/rishitha.chigurupati-group/GitDemo.git

\*\* Git platforms like GitHub, GitLab, Bitbucket, and others have shifted from **master** to **main** to use more inclusive and neutral language.

chigu@Kp MINGW64 ~/GitDemo (main)

$ git pull origin main --allow-unrelated-histories

From https://gitlab.com/rishitha.chigurupati-group/GitDemo

\* branch main -> FETCH\_HEAD

Merge made by the 'ort' strategy.

README.md | 93 +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

1 file changed, 93 insertions(+)

create mode 100644 README.md

chigu@Kp MINGW64 ~/GitDemo (main)

$ git push origin main

Enumerating objects: 6, done.

Counting objects: 100% (6/6), done.

Delta compression using up to 8 threads

Compressing objects: 100% (3/3), done.

Writing objects: 100% (5/5), 614 bytes | 153.00 KiB/s, done.

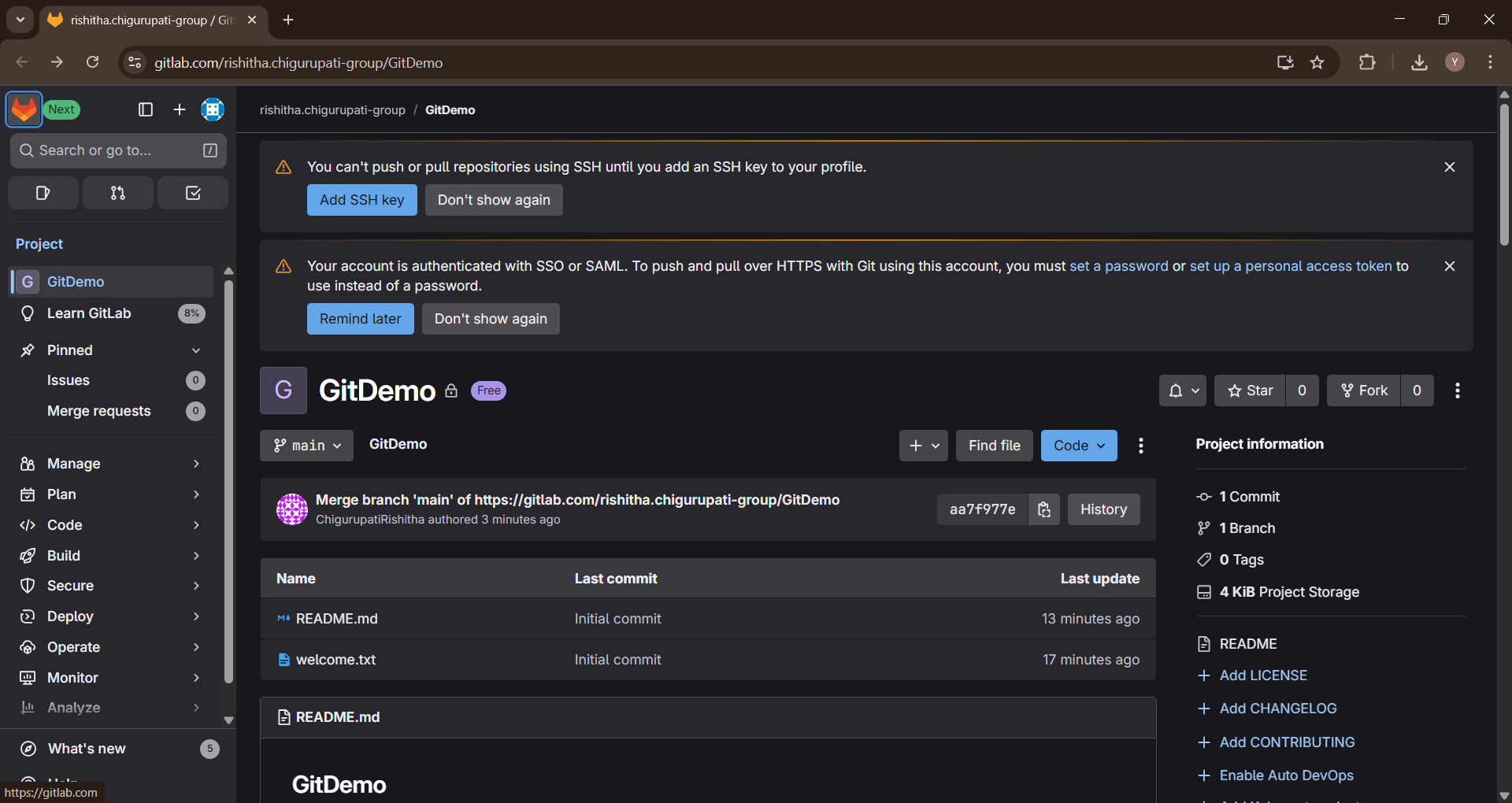
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

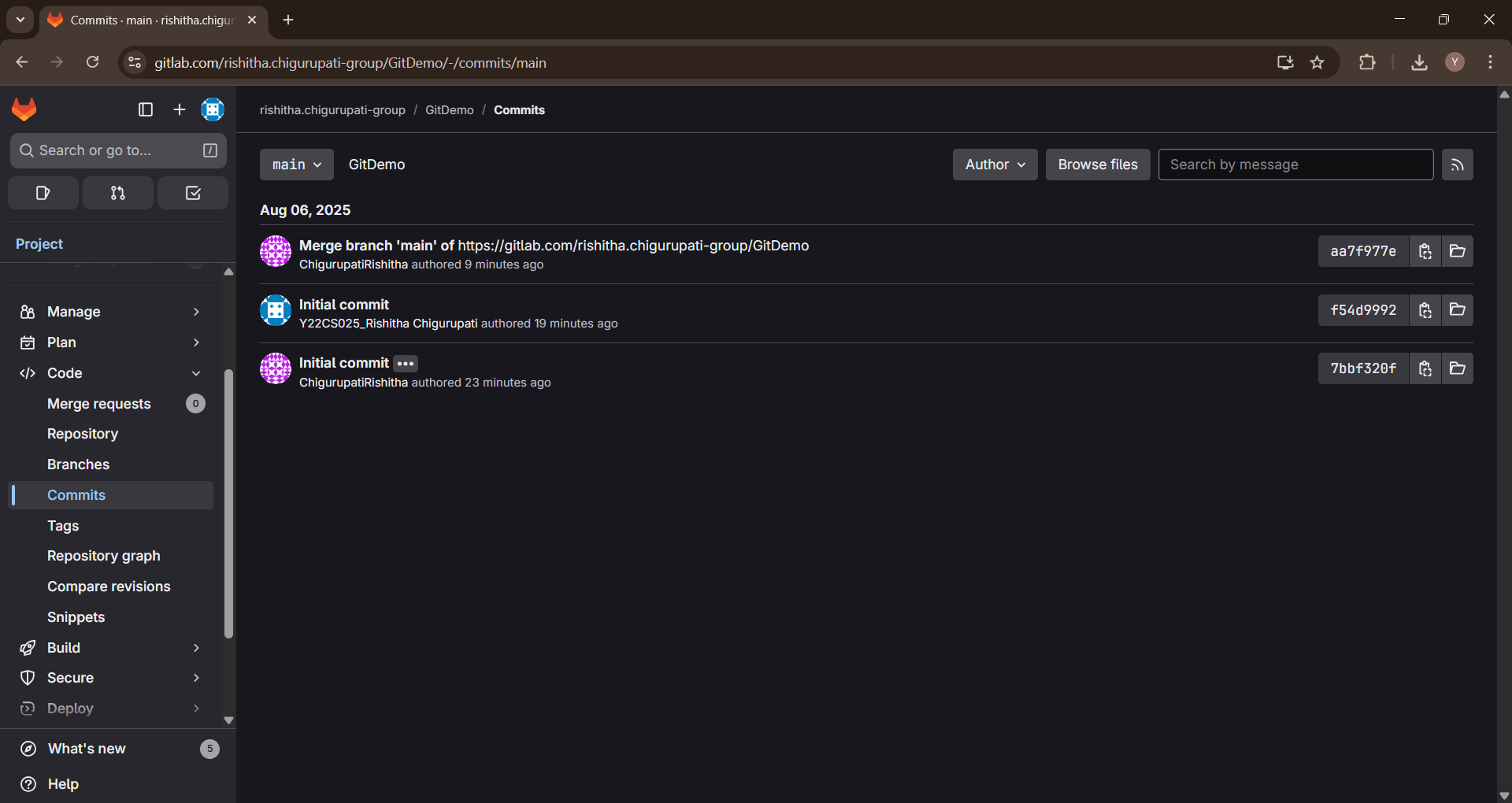
To https://gitlab.com/rishitha.chigurupati-group/GitDemo.git

f54d999..aa7f977 main -> main

chigu@Kp MINGW64 ~/GitDemo (main)

$





**2. Create a *“.log”* file and a *log folder* in the working directory of Git. Update the *.gitignore* file in such a way that on committing, these files (.log extensions and log folders) are ignored.**

Verify if the git status reflects the same about working directory, local repository and git repository.

chigu@Kp MINGW64 ~

$ git config --global user.name "ChigurupatiRishitha"

git config --global user.email "rishitha.chigurupati.email@cognizant.com"

chigu@Kp MINGW64 ~

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

chigu@Kp MINGW64 ~

$ mkdir gitignore-lab

cd gitignore-lab

git init

Initialized empty Git repository in C:/Users/chigu/gitignore-lab/.git/

chigu@Kp MINGW64 ~/gitignore-lab (main)

$ echo "This is a log file" > error.log

mkdir log

echo "Some logs inside folder" > log/debug.txt

chigu@Kp MINGW64 ~/gitignore-lab (main)

$ notepad++ .gitignore

chigu@Kp MINGW64 ~/gitignore-lab (main)

$ git status

On branch main

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

.gitignore

nothing added to commit but untracked files present (use "git add" to track)

chigu@Kp MINGW64 ~/gitignore-lab (main)

$ git add .gitignore

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

[main (root-commit) 4c02aec] Added .gitignore to ignore log files and log folder

1 file changed, 2 insertions(+)

create mode 100644 .gitignore

chigu@Kp MINGW64 ~/gitignore-lab (main)

$ git remote add origin https://gitlab.com/rishitha.chigurupati-group/gitignore-lab.git

chigu@Kp MINGW64 ~/gitignore-lab (main)

$ git pull origin main --allow-unrelated-histories

remote: Enumerating objects: 3, done.

remote: Counting objects: 100% (3/3), done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

Unpacking objects: 100% (3/3), 2.76 KiB | 108.00 KiB/s, done.

From https://gitlab.com/rishitha.chigurupati-group/gitignore-lab

\* branch main -> FETCH\_HEAD

\* [new branch] main -> origin/main

Merge made by the 'ort' strategy.

README.md | 93 +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

1 file changed, 93 insertions(+)

create mode 100644 README.md

chigu@Kp MINGW64 ~/gitignore-lab (main)

$ git push -u origin main

Enumerating objects: 6, done.

Counting objects: 100% (6/6), done.

Delta compression using up to 8 threads

Compressing objects: 100% (3/3), done.

Writing objects: 100% (5/5), 583 bytes | 97.00 KiB/s, done.

Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

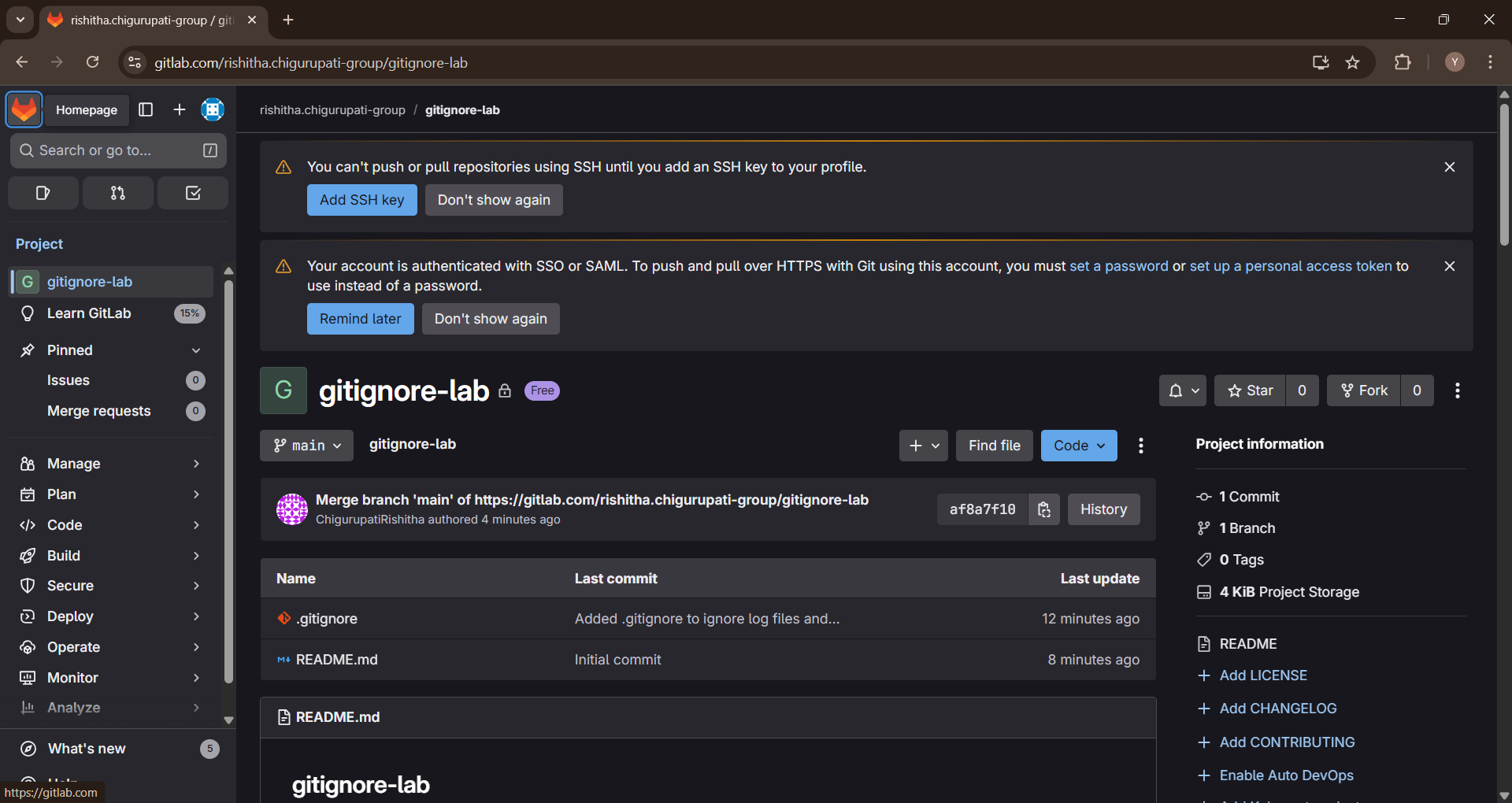
To https://gitlab.com/rishitha.chigurupati-group/gitignore-lab.git

298f44e..af8a7f1 main -> main

branch 'main' set up to track 'origin/main'.

chigu@Kp MINGW64 ~/gitignore-lab (main)

$



**3. Explain branching and merging**

**Explain about creating a branch request in GitLab**

**Explain about creating a merge request in GitLab**

chigu@Kp MINGW64 ~

$ #Clone the Repository

git clone https://gitlab.com/rishitha.chigurupati-group/git-branch-merge-lab.git

Cloning into 'git-branch-merge-lab'...

remote: Enumerating objects: 3, done.

remote: Counting objects: 100% (3/3), done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

Receiving objects: 100% (3/3), done.

chigu@Kp MINGW64 ~

$ cd git-branch-merge-lab

chigu@Kp MINGW64 ~/git-branch-merge-lab (main)

$ #Create a New Branch

git checkout -b GitNewBranch

Switched to a new branch 'GitNewBranch'

chigu@Kp MINGW64 ~/git-branch-merge-lab (GitNewBranch)

$ git branch # local branches

\* GitNewBranch

main

chigu@Kp MINGW64 ~/git-branch-merge-lab (GitNewBranch)

$ git branch -r # remote branches

origin/HEAD -> origin/main

origin/main

chigu@Kp MINGW64 ~/git-branch-merge-lab (GitNewBranch)

$ #Add a New File in the Branch

echo "My first Git branch content" > myfile.txt

chigu@Kp MINGW64 ~/git-branch-merge-lab (GitNewBranch)

$ git add myfile.txt

warning: in the working copy of 'myfile.txt', LF will be replaced by CRLF the next time Git touches it

chigu@Kp MINGW64 ~/git-branch-merge-lab (GitNewBranch)

$ git commit -m "Added myfile.txt to GitNewBranch"

[GitNewBranch 9e98b1f] Added myfile.txt to GitNewBranch

1 file changed, 1 insertion(+)

create mode 100644 myfile.txt

chigu@Kp MINGW64 ~/git-branch-merge-lab (GitNewBranch)

$ git status

On branch GitNewBranch

nothing to commit, working tree clean

chigu@Kp MINGW64 ~/git-branch-merge-lab (GitNewBranch)

$ git checkout main

Switched to branch 'main'

Your branch is up to date with 'origin/main'.

chigu@Kp MINGW64 ~/git-branch-merge-lab (main)

$ git config --global merge.tool p4merge

chigu@Kp MINGW64 ~/git-branch-merge-lab (main)

$ git config --global mergetool.p4merge.cmd "\"C:/Program Files/Perforce/p4merge.exe\" \"$BASE\" \"$LOCAL\" \"$REMOTE\" \"$MERGED\""

chigu@Kp MINGW64 ~/git-branch-merge-lab (main)

$ git config --global diff.tool p4merge

chigu@Kp MINGW64 ~/git-branch-merge-lab (main)

$ git config --global difftool.p4merge.cmd "\"C:/Program Files/Perforce/p4merge.exe\" \"$LOCAL\" \"$REMOTE\""

chigu@Kp MINGW64 ~/git-branch-merge-lab (main)

$ # Switch to GitNewBranch

git checkout GitNewBranch

Switched to branch 'GitNewBranch'

chigu@Kp MINGW64 ~/git-branch-merge-lab (GitNewBranch)

$ # Edit file to make it different

echo "Some new content in branch" >> myfile.txt

chigu@Kp MINGW64 ~/git-branch-merge-lab (GitNewBranch)

$ # Commit change

git add myfile.txt

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

warning: in the working copy of 'myfile.txt', LF will be replaced by CRLF the next time Git touches it

[GitNewBranch f258e43] Updated myfile.txt in GitNewBranch

1 file changed, 1 insertion(+)

chigu@Kp MINGW64 ~/git-branch-merge-lab (GitNewBranch)

$ # Switch back to main

git checkout main

Switched to branch 'main'

Your branch is up to date with 'origin/main'.

chigu@Kp MINGW64 ~/git-branch-merge-lab (main)

$ #Merge Feature Branch into Main

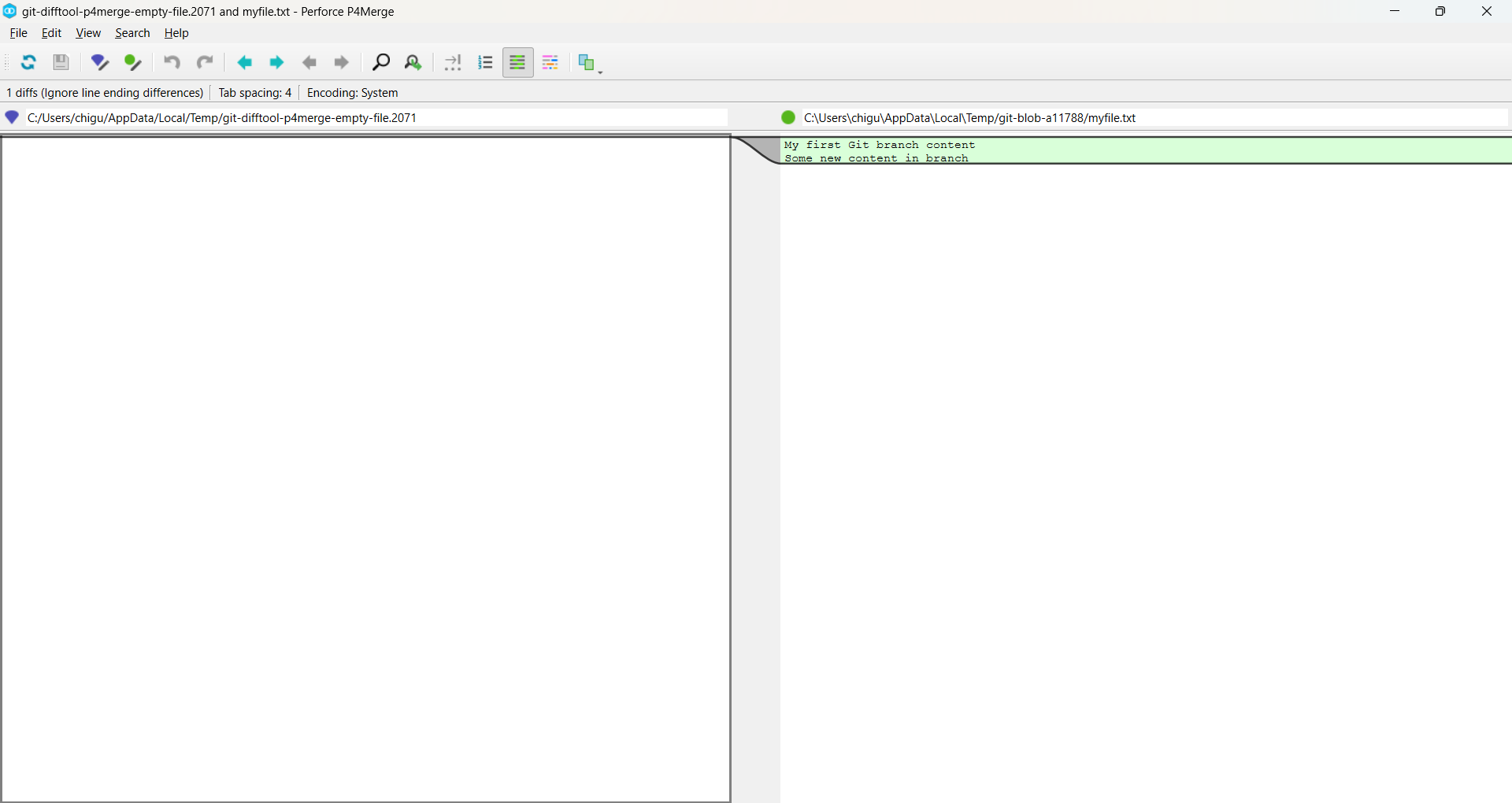
git checkout main

Already on 'main'

Your branch is up to date with 'origin/main'.

chigu@Kp MINGW64 ~/git-branch-merge-lab (main)

$ git difftool main GitNewBranch --tool=p4merge --no-prompt



chigu@Kp MINGW64 ~/git-branch-merge-lab (main)

$ git merge GitNewBranch

Updating e18c499..f258e43

Fast-forward

myfile.txt | 2 ++

1 file changed, 2 insertions(+)

create mode 100644 myfile.txt

chigu@Kp MINGW64 ~/git-branch-merge-lab (main)

$ #View Commit History

git log --oneline --graph --decorate

\* f258e43 (**HEAD** -> **main**, **GitNewBranch**) Updated myfile.txt in GitNewBranch

\* 9e98b1f Added myfile.txt to GitNewBranch

\* e18c499 (**origin/main**, **origin/HEAD**) Initial commit

chigu@Kp MINGW64 ~/git-branch-merge-lab (main)

$ #Delete Merged Branch

git branch -d GitNewBranch

Deleted branch GitNewBranch (was f258e43).

chigu@Kp MINGW64 ~/git-branch-merge-lab (main)

$ #Push Changes to Remote

git push origin main

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (5/5), done.

Writing objects: 100% (6/6), 673 bytes | 134.00 KiB/s, done.

Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To https://gitlab.com/rishitha.chigurupati-group/git-branch-merge-lab.git

e18c499..f258e43 main -> main

chigu@Kp MINGW64 ~/git-branch-merge-lab (main)

$ #Push Changes to Remote

git push origin main

Everything up-to-date

chigu@Kp MINGW64 ~/git-branch-merge-lab (main)

$ # Recreate branch from the commit where you had changes

git checkout -b GitNewBranch f258e43

# Push to remote

git push -u origin GitNewBranch

Switched to a new branch 'GitNewBranch'

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

remote:

remote: To create a merge request for GitNewBranch, visit:

remote: https://gitlab.com/rishitha.chigurupati-group/git-branch-merge-lab/-/merge\_requests/new?merge\_request%5Bsource\_branch%5D=GitNewBranch

remote:

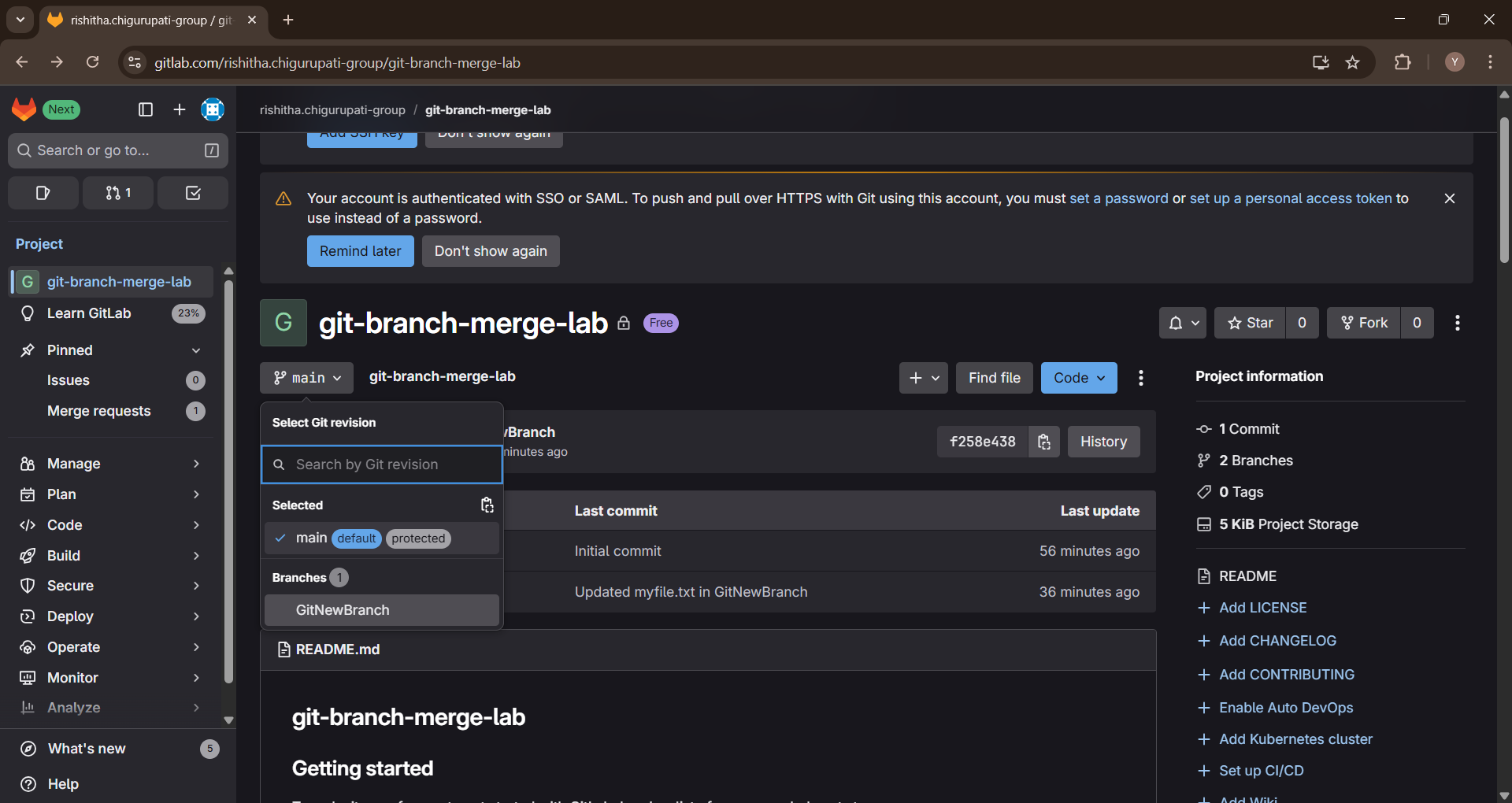
To https://gitlab.com/rishitha.chigurupati-group/git-branch-merge-lab.git

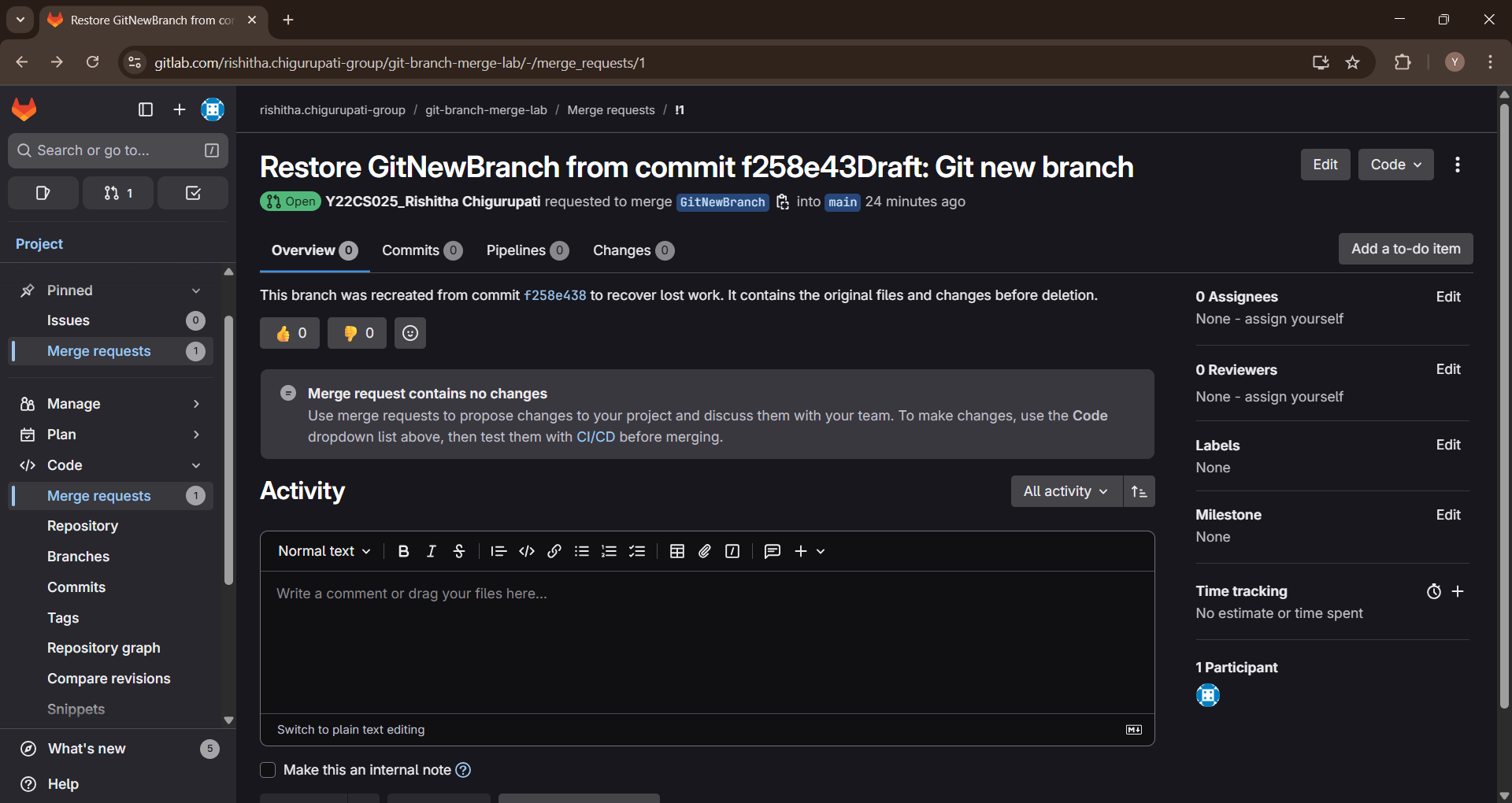
\* [new branch] GitNewBranch -> GitNewBranch

branch 'GitNewBranch' set up to track 'origin/GitNewBranch'.

chigu@Kp MINGW64 ~/git-branch-merge-lab (GitNewBranch)

$





**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.**

chigu@Kp MINGW64 ~

$ cd GitDemo

chigu@Kp MINGW64 ~/GitDemo (main)

$ git rev-parse --abbrev-ref HEAD

main

chigu@Kp MINGW64 ~/GitDemo (main)

$ git status

On branch main

nothing to commit, working tree clean

chigu@Kp MINGW64 ~/GitDemo (main)

$ git switch -c GitWork

# older git: git checkout -b GitWork

Switched to a new branch 'GitWork'

chigu@Kp MINGW64 ~/GitDemo (GitWork)

$ echo "<hello>branch content v1</hello>" > hello.xml

chigu@Kp MINGW64 ~/GitDemo (GitWork)

$ git status

On branch GitWork

Untracked files:

(use "git add <file>..." to include in what will be committed)

hello.xml

nothing added to commit but untracked files present (use "git add" to track)

chigu@Kp MINGW64 ~/GitDemo (GitWork)

$ git add hello.xml

git commit -m "Add hello.xml on GitWork (branch version)"

warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touches it

[GitWork 44d2ca8] Add hello.xml on GitWork (branch version)

1 file changed, 1 insertion(+)

create mode 100644 hello.xml

chigu@Kp MINGW64 ~/GitDemo (GitWork)

$ git switch main

# or git switch master

Switched to branch 'main'

chigu@Kp MINGW64 ~/GitDemo (main)

$ echo "<hello>master content vA</hello>" > hello.xml

chigu@Kp MINGW64 ~/GitDemo (main)

$ git add hello.xml

git commit -m "Add hello.xml on main (different content)"

warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touches it

[main 6d92450] Add hello.xml on main (different content)

1 file changed, 1 insertion(+)

create mode 100644 hello.xml

chigu@Kp MINGW64 ~/GitDemo (main)

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

\* 6d92450 (**HEAD** -> **main**) Add hello.xml on main (different content)

| \* 44d2ca8 (**GitWork**) Add hello.xml on GitWork (branch version)

|/

\* aa7f977 (**origin/main**) Merge branch 'main' of https://gitlab.com/rishitha.chigurupati-group/GitDemo

|\

| \* f54d999 Initial commit

\* 7bbf320 Initial commit - Added welcome.txt - Contains intro message

chigu@Kp MINGW64 ~/GitDemo (main)

$ git diff main..GitWork -- hello.xml

**diff --git a/hello.xml b/hello.xml**

**index ce15d55..c1a3e26 100644**

**--- a/hello.xml**

**+++ b/hello.xml**

@@ -1 +1 @@

-<hello>master content vA</hello>

+<hello>branch content v1</hello>

chigu@Kp MINGW64 ~/GitDemo (main)

$ git config --global merge.tool p4merge

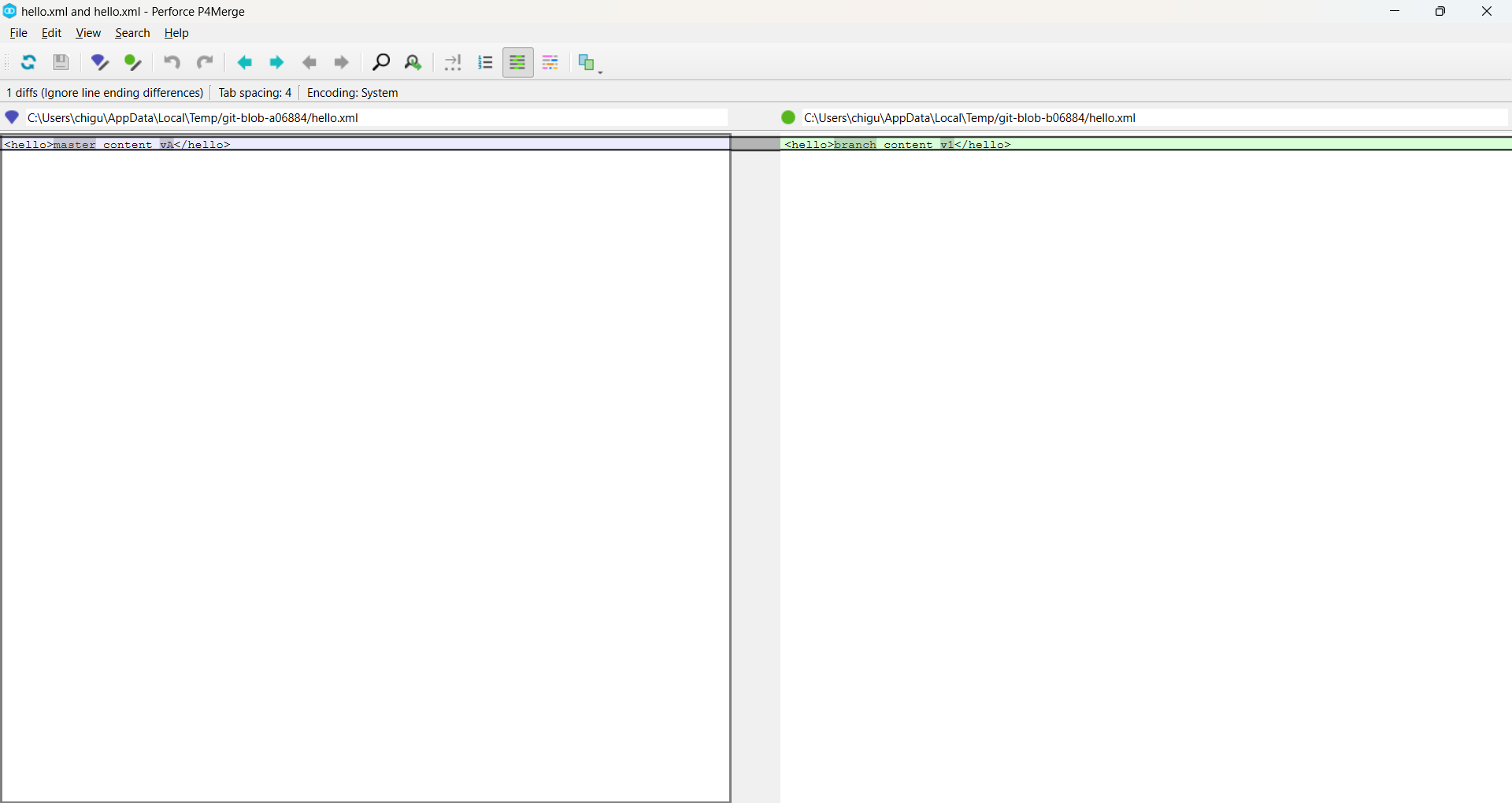
git config --global mergetool.p4merge.path "C:/Program Files/Perforce/p4merge.exe"

chigu@Kp MINGW64 ~/GitDemo (main)

$ git difftool main..GitWork -- hello.xml

Viewing (1/1): 'hello.xml'

Launch 'p4merge' [Y/n]? Y



chigu@Kp MINGW64 ~/GitDemo (main)

$ git switch main

Already on 'main'

chigu@Kp MINGW64 ~/GitDemo (main)

$ git merge GitWork

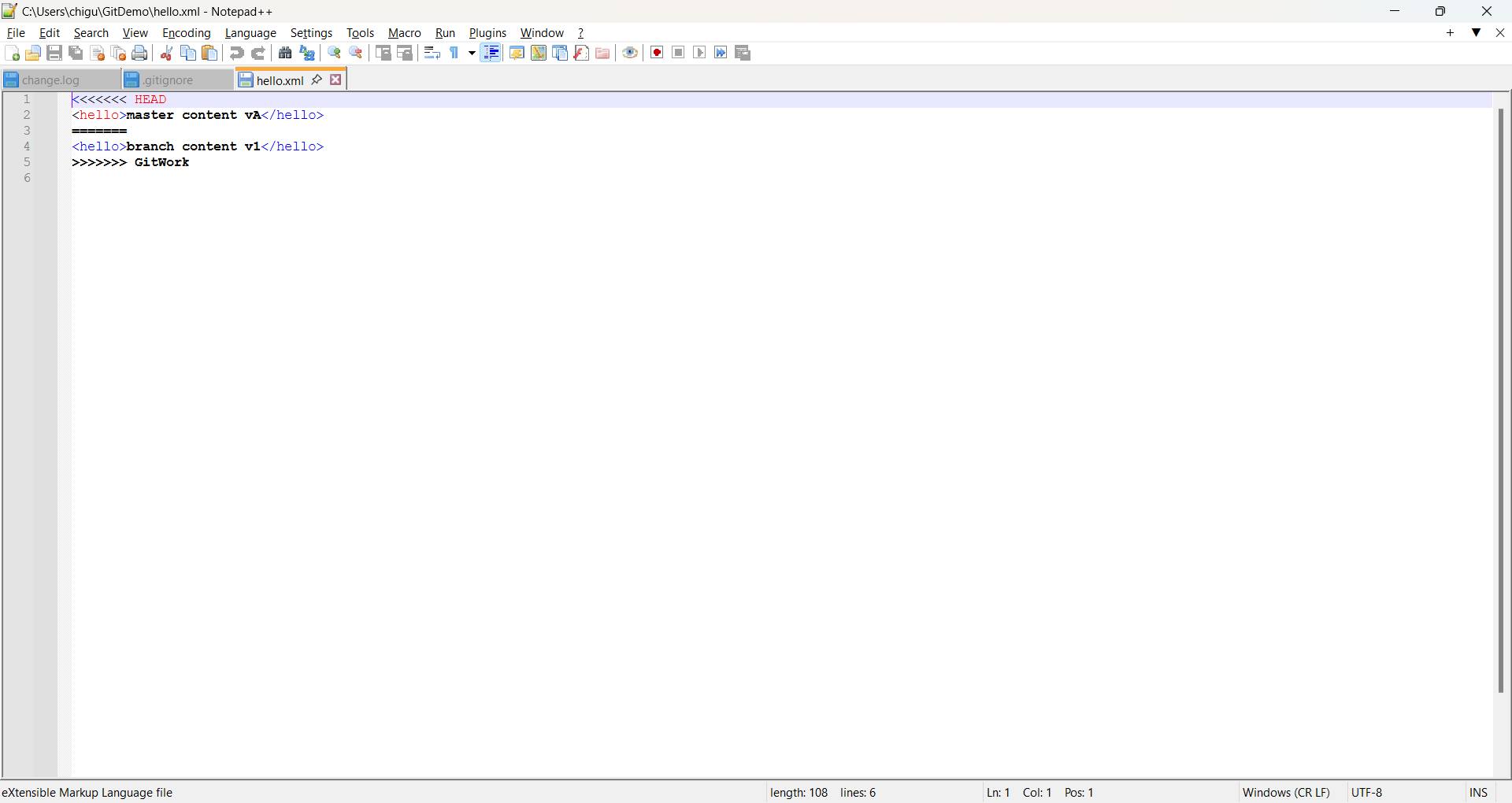
Auto-merging hello.xml

CONFLICT (add/add): Merge conflict in hello.xml

Automatic merge failed; fix conflicts and then commit the result.

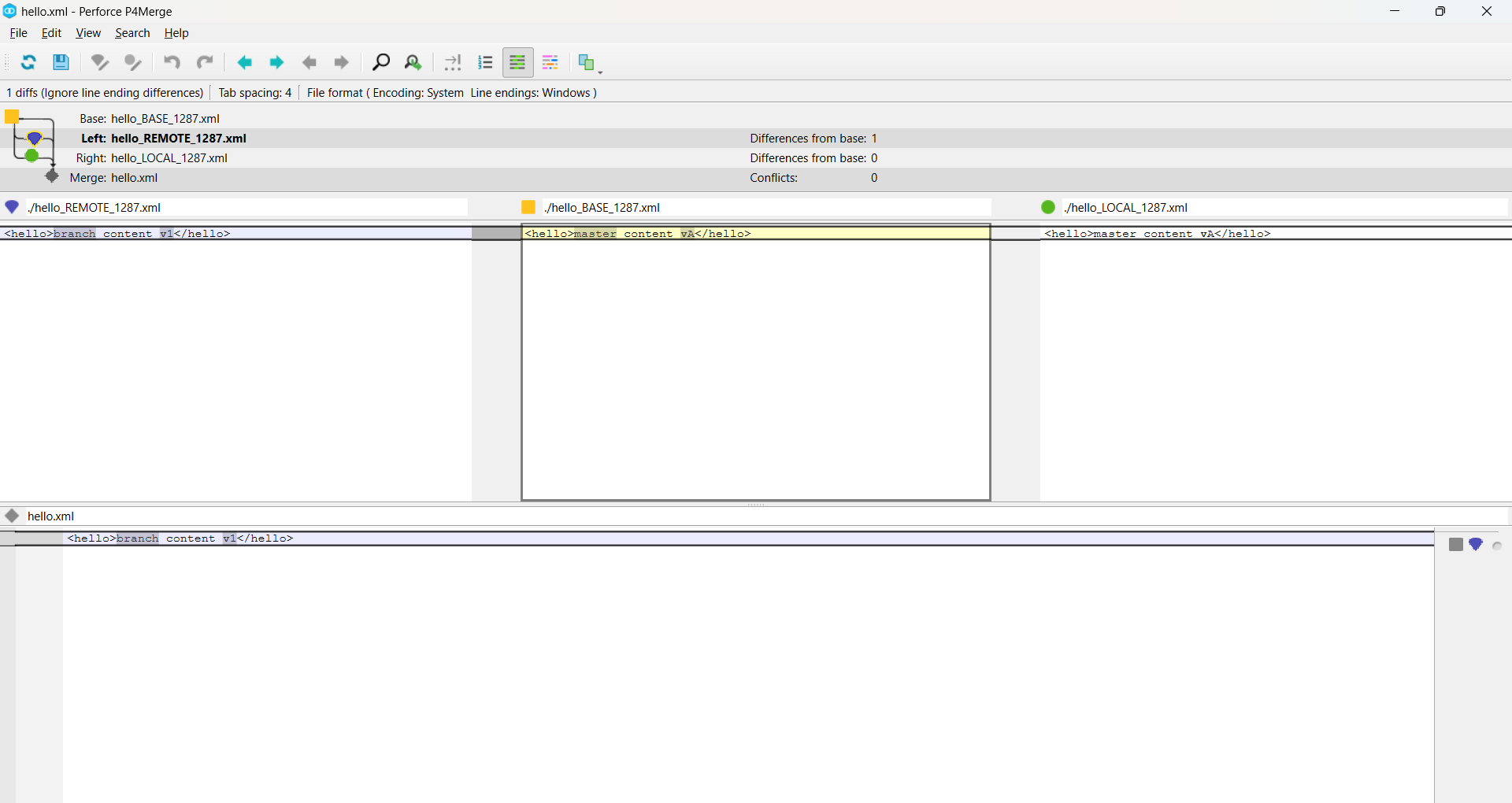
chigu@Kp MINGW64 ~/GitDemo (main|MERGING)

$ notepad++ hello.xml



chigu@Kp MINGW64 ~/GitDemo (main|MERGING)

$ git mergetool



Merging:

hello.xml

Normal merge conflict for 'hello.xml':

{local}: created file

{remote}: created file

error: invalid path './hello\_BASE\_1287.xml'

chigu@Kp MINGW64 ~/GitDemo (main|MERGING)

$ git status

On branch main

All conflicts fixed but you are still merging.

(use "git commit" to conclude merge)

Changes to be committed:

modified: hello.xml

Untracked files:

(use "git add <file>..." to include in what will be committed)

hello.xml.orig

chigu@Kp MINGW64 ~/GitDemo (main|MERGING)

$ git commit

[main 642f67b] Merge branch 'GitWork'

chigu@Kp MINGW64 ~/GitDemo (main)

$ echo "\*.orig" >> .gitignore

git add .gitignore

git commit -m "Ignore merge backup files"

warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it

[main 5e644d3] Ignore merge backup files

1 file changed, 1 insertion(+)

create mode 100644 .gitignore

chigu@Kp MINGW64 ~/GitDemo (main)

$ git push origin main

Enumerating objects: 11, done.

Counting objects: 100% (11/11), done.

Delta compression using up to 8 threads

Compressing objects: 100% (7/7), done.

Writing objects: 100% (10/10), 1.10 KiB | 140.00 KiB/s, done.

Total 10 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)

To https://gitlab.com/rishitha.chigurupati-group/GitDemo.git

aa7f977..5e644d3 main -> main

chigu@Kp MINGW64 ~/GitDemo (main)

$ git status

On branch main

nothing to commit, working tree clean

chigu@Kp MINGW64 ~/GitDemo (main)

$ git add hello.xml

chigu@Kp MINGW64 ~/GitDemo (main)

$ git commit

On branch main

nothing to commit, working tree clean

chigu@Kp MINGW64 ~/GitDemo (main)

$ git commit -m "Merge GitWork into main — resolved hello.xml conflict"

On branch main

nothing to commit, working tree clean

chigu@Kp MINGW64 ~/GitDemo (main)

$ ls -a

git status

**.**/ **..**/ **.git**/ .gitignore README.md hello.xml hello.xml.orig welcome.txt

On branch main

nothing to commit, working tree clean

chigu@Kp MINGW64 ~/GitDemo (main)

$ git add .gitignore

git commit -m "Add backup patterns to .gitignore"

On branch main

nothing to commit, working tree clean

chigu@Kp MINGW64 ~/GitDemo (main)

$ git branch # local branches

GitWork

\* main

chigu@Kp MINGW64 ~/GitDemo (main)

$ git branch -a # local + remote branches

GitWork

\* main

remotes/origin/main

chigu@Kp MINGW64 ~/GitDemo (main)

$ git branch -d GitWork

Deleted branch GitWork (was 44d2ca8).

chigu@Kp MINGW64 ~/GitDemo (main)

$ git branch -a

\* main

remotes/origin/main

chigu@Kp MINGW64 ~/GitDemo (main)

$ git log --oneline --graph --decorate

\* 5e644d3 (**HEAD** -> **main**, **origin/main**) Ignore merge backup files

\* 642f67b Merge branch 'GitWork'

|\

| \* 44d2ca8 Add hello.xml on GitWork (branch version)

\* | 6d92450 Add hello.xml on main (different content)

|/

\* aa7f977 Merge branch 'main' of https://gitlab.com/rishitha.chigurupati-group/GitDemo

|\

| \* f54d999 Initial commit

\* 7bbf320 Initial commit - Added welcome.txt - Contains intro message

chigu@Kp MINGW64 ~/GitDemo (main)

$ git push origin main

Everything up-to-date

chigu@Kp MINGW64 ~/GitDemo (main)

$

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

chigu@Kp MINGW64 ~

$ cd gitignore-lab

chigu@Kp MINGW64 ~/gitignore-lab (main)

$ git status

On branch main

Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

chigu@Kp MINGW64 ~/gitignore-lab (main)

$ git branch -a

\* main

remotes/origin/main

chigu@Kp MINGW64 ~/gitignore-lab (main)

$ git pull origin main

From https://gitlab.com/rishitha.chigurupati-group/gitignore-lab

\* branch main -> FETCH\_HEAD

Already up to date.

chigu@Kp MINGW64 ~/gitignore-lab (main)

$ git push origin main

Everything up-to-date

chigu@Kp MINGW64 ~/gitignore-lab (main)

$

