

# WEEK 8 HANDS ON-GIT

## 1.

Step 1: Setup your machine with Git Configuration

To create a new repository, signup with GitLab and register your credentials  
Login to GitLab and create a "GitDemo" project

1. To check if Git client is installed properly: Open Git bash shell and execute

If output shows Git with its version information that indicates, that Git Client installs properly.

To configure user level configuration of user ID and email ID execute.

To check if the configuration is properly set, execute the following command.

1. Open Git bash shell and create a new project "**GitDemo**" by executing the command

1. Git bash initializes the "**GitDemo**" repository. To verify, execute the command

It will display all the hidden files in the Git "working directory".

2. To create a file "**welcome.txt**" and add content to the file, execute the command

**welcome.txt** is added to the local repository.

3. Signup with GitLab and create a remote repository "**GitDemo**"

4. To pull the remote repository, execute

`git pull origin master`

5. To push the local to remote repository, execute

`git push origin master`

**OUTPUT:**

```
MINGW64/c/Users/bhats/GitDemo
bhats@LAPTOP-614GE080 MINGW64 ~
$ git config --global user.name
sinchana30@gmail.com

bhats@LAPTOP-614GE080 MINGW64 ~
$ git config --global user.email bhatsinchana30@gmail.com

bhats@LAPTOP-614GE080 MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcert=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symbols=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.ushttppath=true
init.defaultbranch=main
user.name=sinchana30@gmail.com
user.email=bhatsinchana30@gmail.com

bhats@LAPTOP-614GE080 MINGW64 ~
$ mkdir GitDemo

bhats@LAPTOP-614GE080 MINGW64 ~
$ cd GitDemo

bhats@LAPTOP-614GE080 MINGW64 ~/GitDemo
$ git init
Initialized empty Git repository in C:/Users/bhats/GitDemo/.git/

bhats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ echo "welcome to Git!">welcome.txt

bhats@LAPTOP-614GE080 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

bhats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git commit -m "Add welcome.txt"
[main (root-commit) 515bcd1] Add welcome.txt
1 file changed, 1 insertion(+),
create mode 100644 welcome.txt

bhats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$
```

```
MINGW64/c/Users/bhats/GitDemo
bhats@LAPTOP-614GE080 MINGW64 ~
$ git config --global user.name
sinchana30@gmail.com

bhats@LAPTOP-614GE080 MINGW64 ~
$ git config --global user.email bhatsinchana30@gmail.com

bhats@LAPTOP-614GE080 MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcert=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symbols=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.ushttppath=true
init.defaultbranch=main
user.name=sinchana30@gmail.com
user.email=bhatsinchana30@gmail.com

bhats@LAPTOP-614GE080 MINGW64 ~
$ mkdir GitDemo

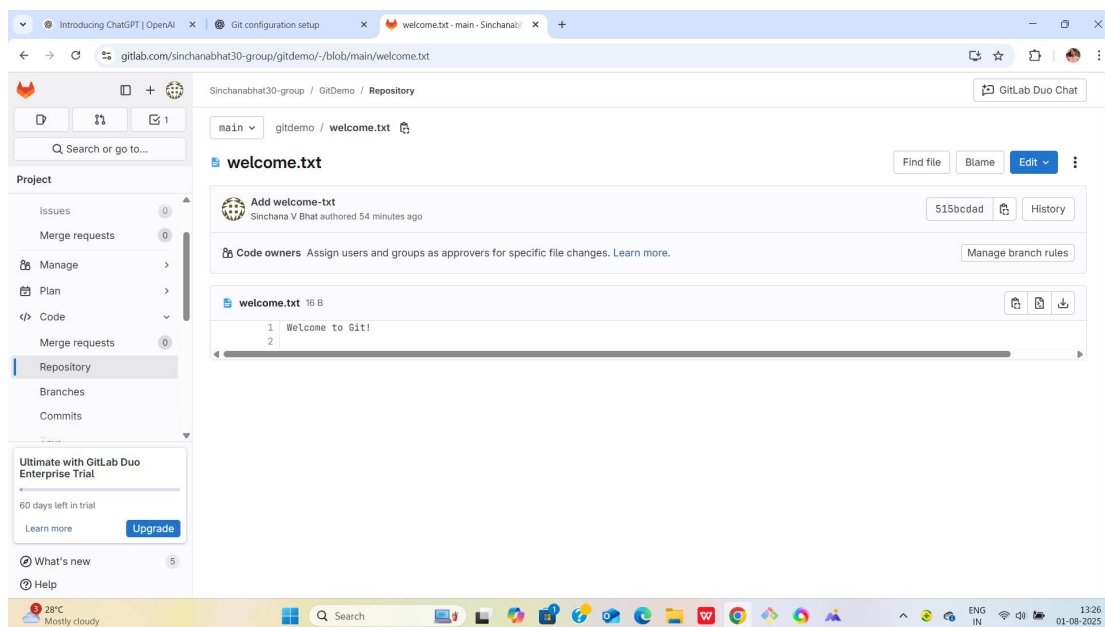
bhats@LAPTOP-614GE080 MINGW64 ~
$ cd GitDemo

bhats@LAPTOP-614GE080 MINGW64 ~/GitDemo
$ git init
Initialized empty Git repository in C:/Users/bhats/GitDemo/.git/
```

```
MINGW64/c/Users/bhats/GitDemo

warning: redirecting to https://gitlab.com/sinchanabhat30-group/gitdemo.git/
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), 2.74 KiB | 255.00 KiB/s, done.
From https://gitlab.com/sinchanabhat30-group/gitdemo
* 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

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git push -u origin main
warning: redirecting to https://gitlab.com/sinchanabhat30-group/gitdemo.git/
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 537 bytes | 537.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://gitlab.com/sinchanabhat30-group/gitdemo
 f3d9253..134e91d main -> main
branch 'main' set up to track 'origin/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

## OUTPUTS:

```
MINGW64/C/Users/shats/GitDemo
shats@LAPTOP-614GE080 MINGW64 ~
$ cd ~/GitDemo
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ echo "This is a log file">debug.log
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ mkdir log
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ echo "Inside log folder">log/info.txt
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ touch .gitignore
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ notepad++ .gitignore
bash: notepad++: command not found
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ notepad .gitignore
git add .gitignore
git commit -m "ignore .log files and log folders"
git status
rm .gitignore
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git add .gitignore
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git commit -m "ignore .log files and log folders"
[main d/930fa] ignore .log files and log folders
1 file changed, 2 insertions(+)
create mode 100644 .gitignore
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use 'git push' to publish your local commits)

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

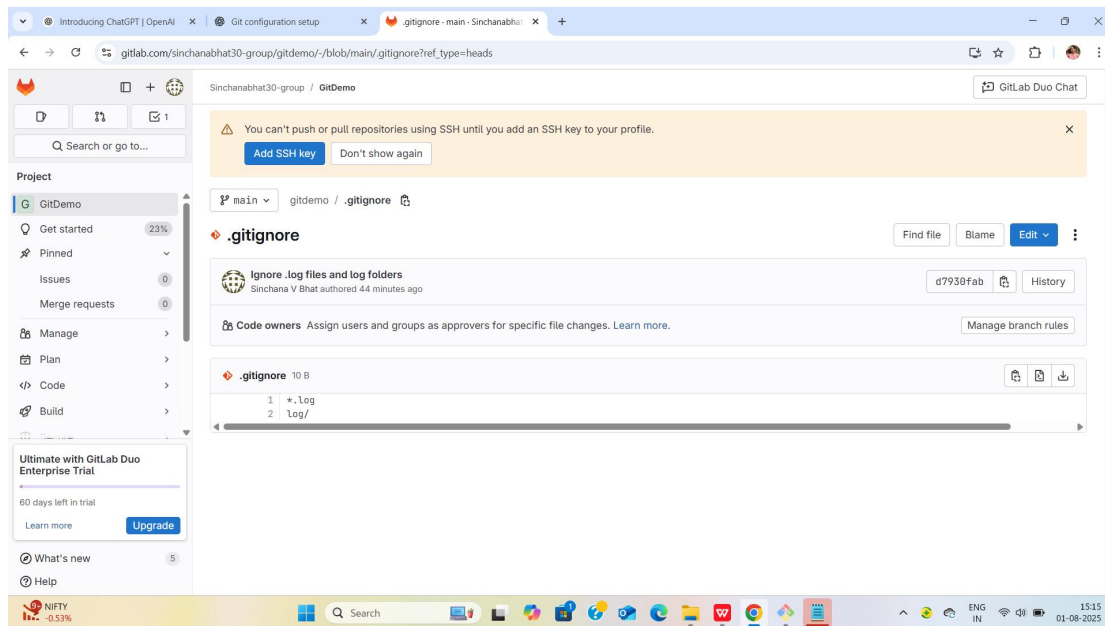
nothing added to commit but untracked files present (use "git add" to track)
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ rm .gitignore
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ notepad .gitignore
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git add .gitignore
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
```

```
MINGW64/C/Users/shats/GitDemo
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ rm .gitignore
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ notepad .gitignore
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git add .gitignore
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git commit -m "correct .gitignore to ignore log files and log folder"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use 'git push' to publish your local commits)

nothing to commit, working tree clean
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use 'git push' to publish your local commits)

nothing to commit, working tree clean
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ AC
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git push
warning: redirecting to https://gitlab.com/sinchanabhat30-group/gitdemo.git/
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 336 bytes | 168.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://gitlab.com/sinchanabhat30-group/gitdemo
134e91d..d7930fa main -> main
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ AC
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ echo "test log">test.log
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ AC
shats@LAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$
```



3.

### Branching:

1. Create a new branch “**GitNewBranch**”.
2. List all the local and remote branches available in the current trunk. Observe the “\*” mark which denote the current pointing branch.
3. Switch to the newly created branch. Add some files to it with some contents.
4. Commit the changes to the branch.
5. Check the status with “**git status**” command.

### OUTPUT:

```

MINGW64/C:/Users/bhats/GitDemo
$ cd ~/GitDemo

$ git branch
$ git branch -a
* GitNewBranch
main
Remotes/origin/main

$ git checkout GitNewBranch
Switched to branch 'GitNewBranch'

$ echo "This is from GitNewBranch">branchfile.txt
$ git add branchfile.txt
warning: in the working copy of 'branchfile.txt', LF will be replaced by CRLF the next time Git touches it
$ git commit -m "Add branchfile.txt in GitNewBranch"
[GitNewBranch 6cebcbf] Add branchfile.txt in GitNewBranch
1 file changed, 1 insertion(+)
create mode 100644 branchfile.txt

$ git status
On branch GitNewBranch
nothing to commit, working tree clean

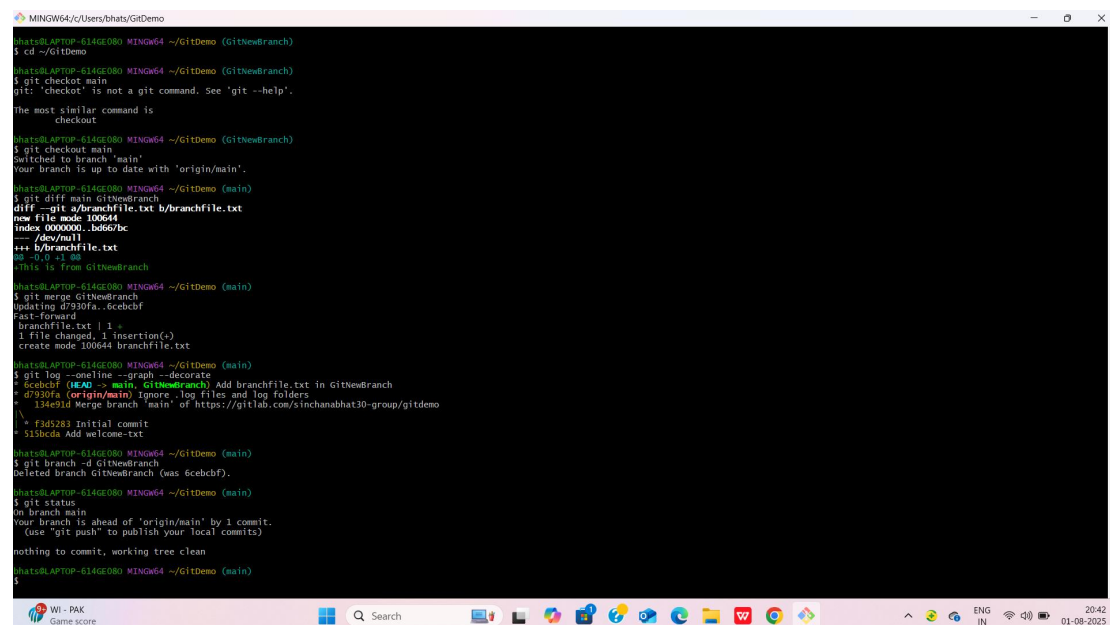
$

```

## Merging:

1. Switch to the master
2. List out all the differences between trunk and branch. These provide the differences in command line interface.
3. List out all the visual differences between master and branch using **P4Merge tool**.
4. Merge the source branch to the trunk.
5. Observe the logging after merging using “**git log --oneline --graph --decorate**”
6. Delete the branch after merging with the trunk and observe the git status.

## OUTPUT:



```
shats@LAPTOP-614GED80 MINGW64 ~/GitDemo (GitNewBranch)
$ cd ~/GitDemo

shats@LAPTOP-614GED80 MINGW64 ~/GitDemo (GitNewBranch)
$ git checkout main
git: 'checkout' is not a git command. See 'git --help'.
The most similar command is
  checkout

shats@LAPTOP-614GED80 MINGW64 ~/GitDemo (GitNewBranch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

shats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git diff main GitNewBranch
diff --git a/branchfile.txt b/branchfile.txt
new file mode 100644
index 000000..bd66bc
--- /dev/null
+++ b/branchfile.txt
@@ -0,0 +1 @@
+This is from GitNewBranch

shats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git merge GitNewBranch
Updating d7930fa..6cebcbf
Fast-forward
 branchfile.txt | 1
 1 file changed, 1 insertion(+)
 create mode 100644 branchfile.txt

shats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git log --oneline --graph --decorate
 6cebcbf (HEAD -> main, GitNewBranch) Add branchfile.txt in GitNewBranch
 d7930fa (origin/main) Ignore .log files and log folders
 134e51d Merge branch 'main' of https://gitlab.com/sinchanabhat30-group/gitdemo
 * f3d5283 Initial commit
 * 233bdc6 Add welcome.txt

shats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git branch -d GitNewBranch
Deleted branch GitNewBranch (was 6cebcbf).

shats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use 'git push' to publish your local commits)

nothing to commit, working tree clean
shats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$
```

4.

1. Verify if master is in clean state.
2. Create a branch “**GitWork**”. Add a file “hello.xml”.
3. Update the content of “hello.xml” and observe the status
4. Commit the changes to reflect in the branch
5. Switch to master.
6. Add a file “**hello.xml**” to the master and add some different content than previous.
7. Commit the changes to the master
8. Observe the log by executing “**git log --oneline --graph --decorate --all**”
9. Check the differences with Git diff tool
10. For better visualization, use P4Merge tool to list out all the differences between master and branch
11. Merge the bran to the master
12. Observe the git mark up.
13. Use 3-way merge tool to resolve the conflict

14. Commit the changes to the master, once done with conflict
15. Observe the git status and add backup file to the .gitignore file.
16. Commit the changes to the .gitignore
17. List out all the available branches
18. Delete the branch, which merge to master.
19. Observe the log by executing “git log –online –graph –decorate”

## OUTPUTS:

```
MINGW64/c/Users/bhats/GitDemo
$ cd ~/GitDemo

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git checkout main
Already on 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git checkout -b Gitwork
Switched to a new branch 'Gitwork'

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Gitwork)
$ echo "message:hello from Gitwork/messages">hello.xml

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Gitwork)
$ git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touches it

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Gitwork)
$ git commit -m "Add hello.xml in Gitwork branch"
[Gitwork 0fc207b] Add hello.xml in Gitwork branch
1 file changed, 1 insertion(+)
create mode 100644 hello.xml

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Gitwork)
$ echo "cupdate:updated content in Gitwork/updates">>hello.xml

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Gitwork)
$ git status
On branch Gitwork
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.xml

no changes added to commit (use "git add" and/or "git commit -a")

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Gitwork)
$ git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touches it

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Gitwork)
$ git commit -m "Update hello.xml in Gitwork"
[Gitwork bf6e674] Update hello.xml in Gitwork
1 file changed, 1 insertion(+)

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Gitwork)
```

```
MINGW64/c/Users/bhats/GitDemo

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Gitwork)
$ git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touches it

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Gitwork)
$ git commit -m "Update hello.xml in Gitwork"
[Gitwork bf6e674] Update hello.xml in Gitwork
1 file changed, 1 insertion(+)

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Gitwork)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ echo "message:hello from new branch/messages">hello.xml

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touches it

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git commit -m "Add hello.xml in main with different content"
[main ce36b30] Add hello.xml in main with different content
1 file changed, 1 insertion(+)
create mode 100644 hello.xml

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git log --online --graph --decorate --all
fatal: unrecognized argument: --online

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git log --online --graph --decorate --all
ce36b30 (HEAD -> main) Add hello.xml in main with different content
* bf6e674 (Gitwork) Update hello.xml in Gitwork
* 0fc207b Add hello.xml in Gitwork branch
/
* f6cebd9 Add branchFile.txt in Gitnewbranch
* 8f330fa (origin/main) Ignore .log files and log folders
* 134e91d Merge branch 'main' of https://github.com/sinchanabhat30-group/gitdemo
/
* f3d5283 Initial commit
* 915b6da Add welcome.txt

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git diff main Gitwork
diff --git a/hello.xml b/hello.xml
index 497b831..d8e1077 100644
--- a/hello.xml
+++ b/hello.xml
@@ -1,2 @@
- message:hello from new branch/messages
+ message:hello from Gitwork/messages
+ cupdate:updated content in Gitwork/updates

bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
```

```
MINGW64/C/Users/bhats/GitDemo
shutsBLAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git checkout main
Already on 'main'
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)

shutsBLAPTOP-614GE080 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.

shutsBLAPTOP-614GE080 MINGW64 ~/GitDemo (main|MERGING)
$ notepad hello.xml
$ git add hello.xml

shutsBLAPTOP-614GE080 MINGW64 ~/GitDemo (main|MERGING)
$ git commit -m "Merge Gitwork into main, resolved conflict"
[main 47cf1c0] Merge Gitwork into main, resolved conflict

shutsBLAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 5 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

shutsBLAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git branch -a
* main
remotes/origin/main

shutsBLAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git branch -d Gitwork
Deleted branch Gitwork (was bf6e674).

shutsBLAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$ git log --oneline --graph --decorate
* 47cf1c0 (HEAD -> main) Merge Gitwork into main, resolved conflict
* bf6e674 Update hello.xml in Gitwork
* 0fc207b Add hello.xml in Gitwork branch
* ce50b50 Add hello.xml in main with different content
* 6cebc3f Add branchfile.txt in GitNewBranch
* 07310fa (origin/main) ignore .log files and log folders
* 134e91d Merge branch 'main' of https://gitlab.com/sinchanabhat30-group/gitdemo
* f3d5283 Initial commit
* 515bcd4 Add welcome-txt

shutsBLAPTOP-614GE080 MINGW64 ~/GitDemo (main)
$
```

## 5.

Please follow the instructions to complete the hands-on. Each instruction expects a command for the Git Bash.

Verify if master is in clean state.

List out all the available branches.

Pull the remote git repository to the master

Push the changes, which are pending from **"Git-T03-HOL\_002"** to the remote repository.

- Hands-on ID: **"Git-T03-HOL\_002"**

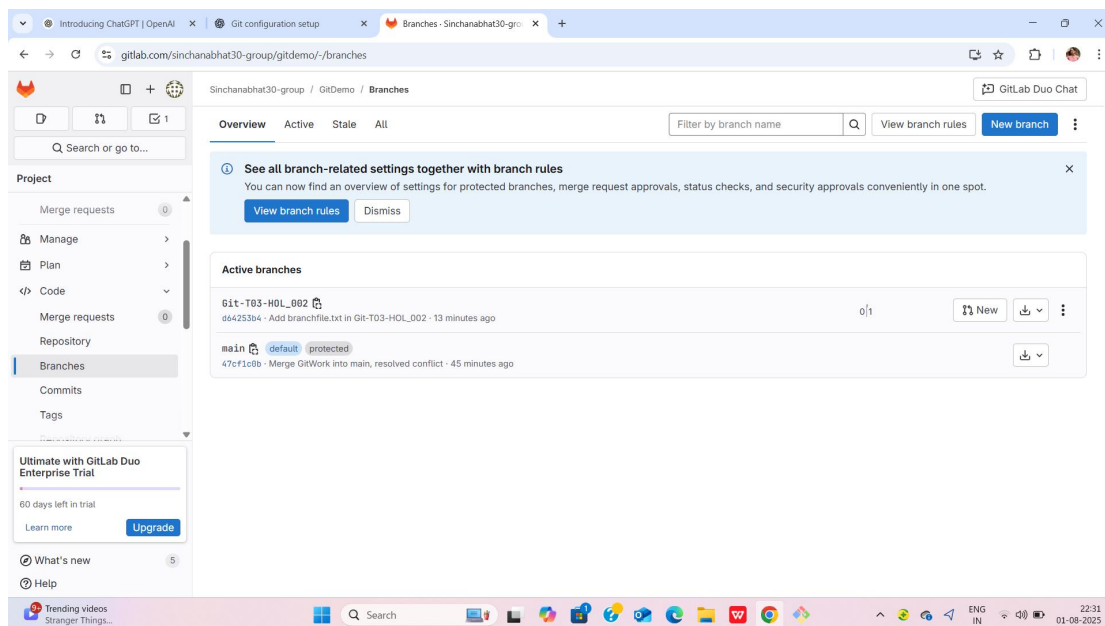
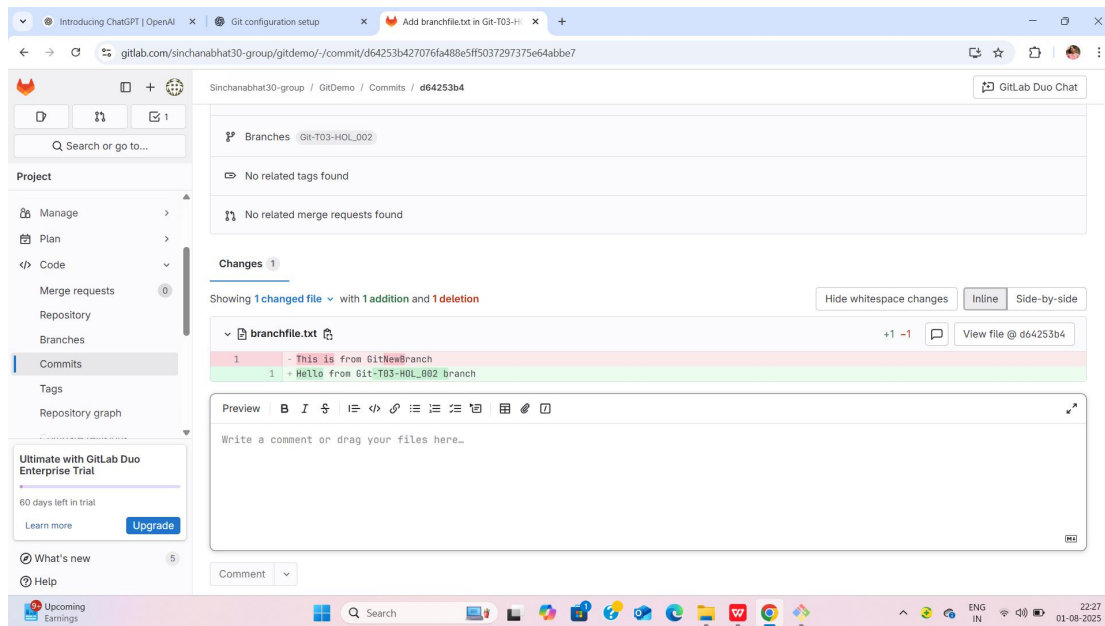
Observe if the changes are reflected in the remote repository.

## OUTPUTS:



```
MINGW64/C/Users/bhats/GitDemo
bhats@LAPTOP-614GED80 MINGW64 ~
$ cd ~/GitDemo
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git checkout main
bash: git: command not found
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git checkout main
Already on 'main'
Your branch is ahead of 'origin/main' by 5 commits.
(use "git push" to publish your local commits)
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 5 commits.
(use "git push" to publish your local commits)
nothing to commit, working tree clean
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git branch -a
* main
remotes/origin/main
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git pull origin main
warning: redirecting to https://gitlab.com/sinchanabhat30-group/gitdemo.git/
From https://gitlab.com/sinchanabhat30-group/gitdemo
  branch      main       -> FETCH_HEAD
Already up to date.
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git pull origin master
warning: redirecting to https://gitlab.com/sinchanabhat30-group/gitdemo.git/
fatal: couldn't find remote ref master
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git checkout Git-T03-HOL_002
error: pathspec 'Git-T03-HOL_002' did not match any file(s) known to git
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ AC
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ AC
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git checkout -b Git-T03-HOL_002
Switched to a new branch 'Git-T03-HOL_002'
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ echo 'Hello from Git-T03-HOL_002 branch'->branchfile.txt
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git add branchfile.txt
```

```
error: src refspec Git-T03-HOL_002 does not match any
error: failed to push some refs to 'https://gitlab.com/sinchanabhat30-group/gitdemo'
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git push origin Git-T03-HOL_002
error: src refspec Git-T03-HOL_002 does not match any
error: failed to push some refs to 'https://gitlab.com/sinchanabhat30-group/gitdemo'
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ AC
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git push origin Git-T03-HOL_002
error: src refspec Git-T03-HOL_002 does not match any
error: failed to push some refs to 'https://gitlab.com/sinchanabhat30-group/gitdemo'
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git push origin Git-T03-HOL_002
warning: redirecting to https://gitlab.com/sinchanabhat30-group/gitdemo.git/
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 12 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (18/18), 1.69 KiB | 346.00 KiB/s, done.
Total 18 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: To create a merge request for Git-T03-HOL_002, visit:
remote:   https://gitlab.com/sinchanabhat30-group/gitdemo/-/merge_requests/new?merge_requests%5Bsource_branch%5D=Git-T03-HOL_002
To https://gitlab.com/sinchanabhat30-group/gitdemo
 * [new branch]    Git-T03-HOL_002 -> Git-T03-HOL_002
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (Git-T03-HOL_002)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 5 commits.
(use "git push" to publish your local commits)
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git pull origin main
warning: redirecting to https://gitlab.com/sinchanabhat30-group/gitdemo.git/
From https://gitlab.com/sinchanabhat30-group/gitdemo
  branch      main       -> FETCH_HEAD
Already up to date.
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git merge Git-T03-HOL_002
merge: Git-T03-HOL_002 - not something we can merge
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$ git push origin main
warning: redirecting to https://gitlab.com/sinchanabhat30-group/gitdemo.git/
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://gitlab.com/sinchanabhat30-group/gitdemo
 d7930fa..47cf1c0 main -> main
bhats@LAPTOP-614GED80 MINGW64 ~/GitDemo (main)
$
```



## Description about this Hands On 5:

After pushing the branch to GitLab, verify the changes by checking that the branch name appears under **Repository** → **Branches** in GitLab.

Open the branch and confirm that the committed files are present.

Check the file contents to ensure they match your local changes.

View the branch's commit history to confirm your latest commit is recorded.

This ensures your push was successful and the remote repository is up to date.



