**Hands-On 1 :-**

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git --version

git version 2.43.0.windows.1

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ 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.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt

core.autocrlf=true

core.fscache=true

core.symlinks=false

pull.rebase=false

credential.helper=manager

credential.https://dev.azure.com.usehttppath=true

init.defaultbranch=main

user.email=vsbangalore15@gmail.com

user.name=Vishu-15

core.repositoryformatversion=0

core.filemode=false

core.bare=false

core.logallrefupdates=true

core.symlinks=false

core.ignorecase=true

remote.origin.url=https://gitlab.com/gitdemo1001732/GitDemo.git

remote.origin.fetch=+refs/heads/\*:refs/remotes/origin/\*

branch.main.remote=origin

branch.main.merge=refs/heads/main

User@LAPTOP-GFV793Q2 MINGW64 ~

$ git config --global -e

User@LAPTOP-GFV793Q2 MINGW64 ~

$ mkdir GitDemo

cd GitDemo

git init

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

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ echo "Welcome to Git" > welcome.txt

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ ls

welcome.txt

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ cat welcome.txt

Welcome to Git

User@LAPTOP-GFV793Q2 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)

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git push -u origin main

Enumerating objects: 3, done.

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

Writing objects: 100% (3/3), 236 bytes | 236.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

To https://gitlab.com/gitdemo1001732/GitDemo.git

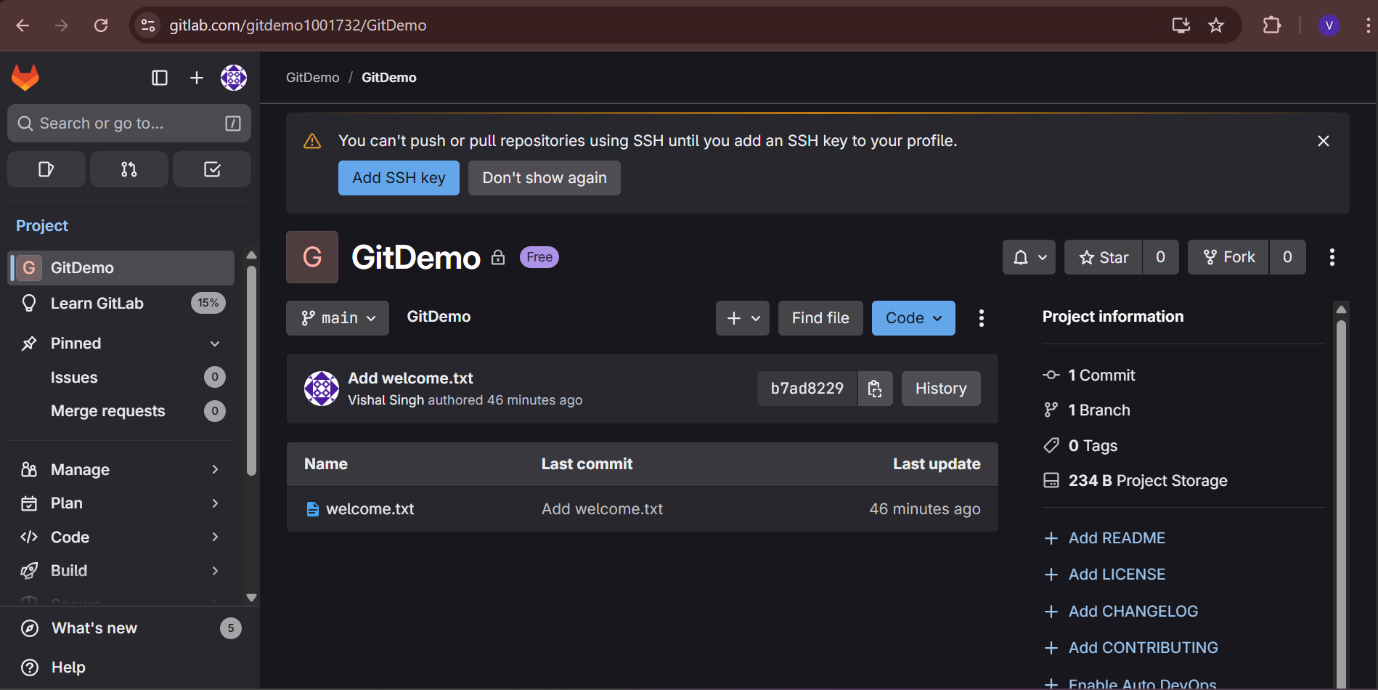
\* [new branch] main -> main

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

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git ls-remote --heads origin

b7ad8229c52bfb4fdf182958dbf3b55f0a788b41 refs/heads/main



**Hands-On 2 :-**

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ touch debug.log

mkdir log

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ ls

debug.log log/ welcome.txt

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ touch .gitignore

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ notepad++ .gitignore

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git status

On branch main

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

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)

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git add .gitignore

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

[main d60d734] Add .gitignore to ignore .log files and log folder

1 file changed, 5 insertions(+)

create mode 100644 .gitignore

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ touch test.log

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

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git push

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 12 threads

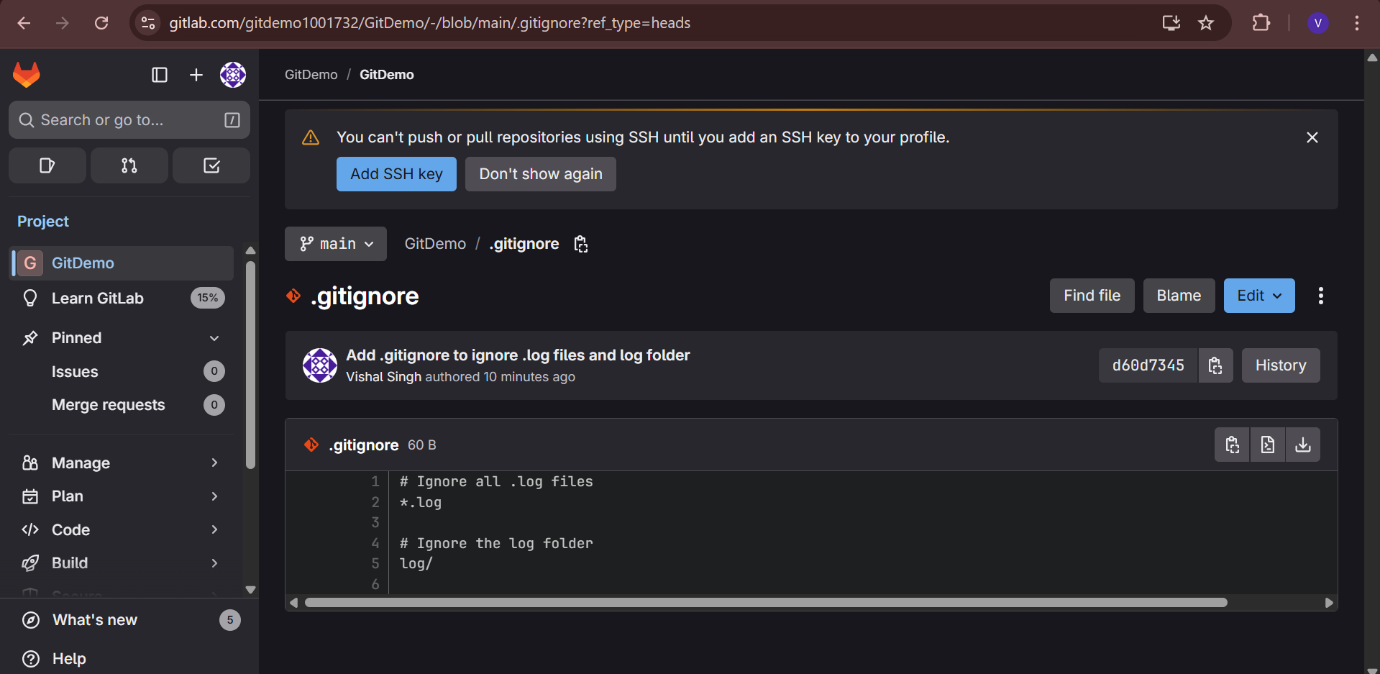
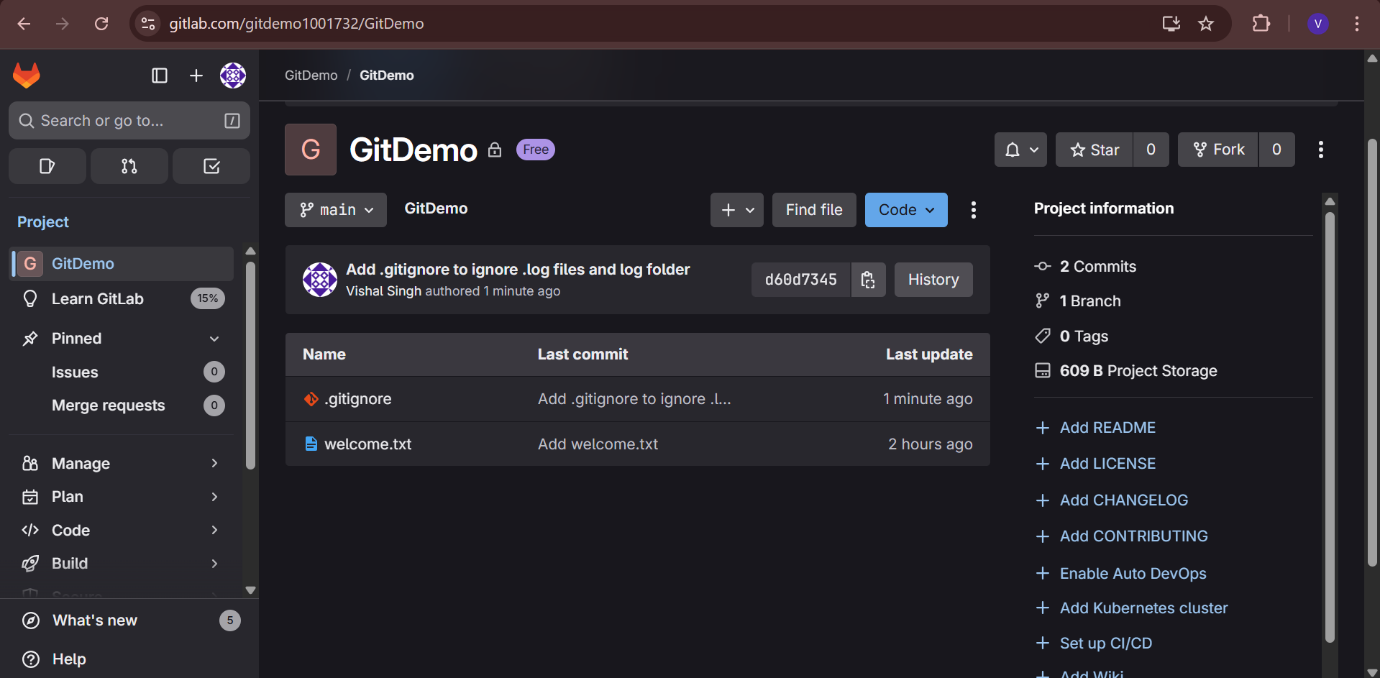
Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 353 bytes | 353.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

To https://gitlab.com/gitdemo1001732/GitDemo.git

b7ad822..d60d734 main -> main



**Hands-On 3 :-**

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git branch GitNewBranch

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git branch -a

GitNewBranch

\* main

remotes/origin/main

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git checkout GitNewBranch

Switched to branch 'GitNewBranch'

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (GitNewBranch)

$ echo "This is a file for GitNewBranch" > branchfile.txt

echo "Another file for testing" > branchnotes.txt

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (GitNewBranch)

$ git add .

git commit -m "Add branch-specific files in GitNewBranch"

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

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

[GitNewBranch ed36a0c] Add branch-specific files in GitNewBranch

2 files changed, 2 insertions(+)

create mode 100644 branchfile.txt

create mode 100644 branchnotes.txt

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (GitNewBranch)

$ git status

On branch GitNewBranch

nothing to commit, working tree clean

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (GitNewBranch)

$ git checkout main

Switched to branch 'main'

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

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git diff GitNewBranch

**diff --git a/branchfile.txt b/branchfile.txt**

**deleted file mode 100644**

**index 3eb56d2..0000000**

**--- a/branchfile.txt**

**+++ /dev/null**

@@ -1 +0,0 @@

-This is a file for GitNewBranch

**diff --git a/branchnotes.txt b/branchnotes.txt**

**deleted file mode 100644**

**index c3d7ac9..0000000**

**--- a/branchnotes.txt**

**+++ /dev/null**

@@ -1 +0,0 @@

-Another file for testing

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git mergetool

This message is displayed because 'merge.tool' is not configured.

See 'git mergetool --tool-help' or 'git help config' for more details.

'git mergetool' will now attempt to use one of the following tools:

opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecmerge p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff

No files need merging

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git merge GitNewBranch

Updating d60d734..ed36a0c

Fast-forward

branchfile.txt | 1 +

branchnotes.txt | 1 +

2 files changed, 2 insertions(+)

create mode 100644 branchfile.txt

create mode 100644 branchnotes.txt

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

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

\* ed36a0c (HEAD -> main, GitNewBranch) Add branch-specific files in GitNewBranch

\* d60d734 (origin/main) Add .gitignore to ignore .log files and log folder

\* b7ad822 Add welcome.txt

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git branch -d GitNewBranch

Deleted branch GitNewBranch (was ed36a0c).

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

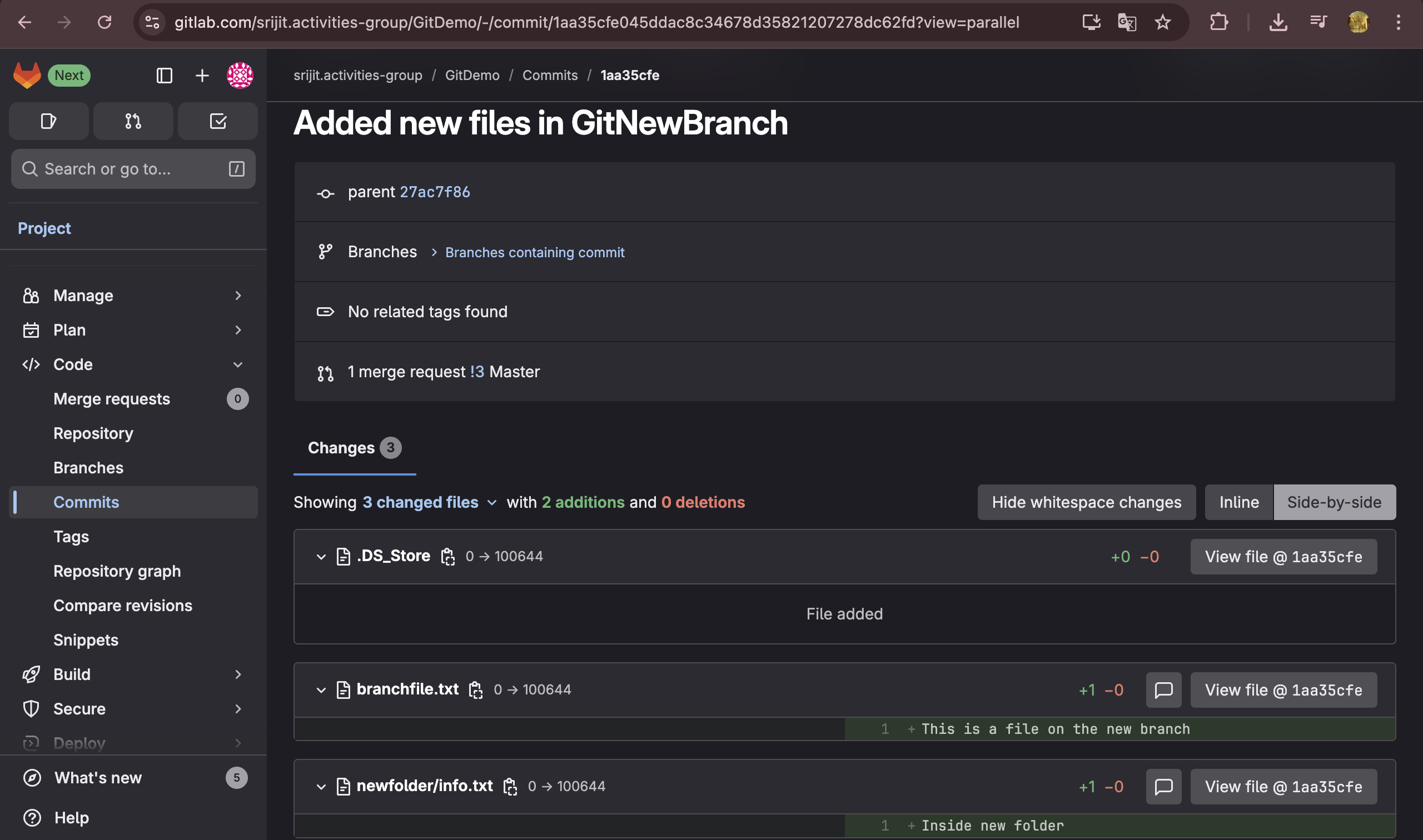
$ git status

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



**Hands-On 4 :-**

User@LAPTOP-GFV793Q2 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)

User@LAPTOP-GFV793Q2 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

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git checkout -b GitWork

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

Switched to a new branch 'GitWork'

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (GitWork)

$ echo "<message>Hello again from GitWork branch - updated</message>" > hello.xml

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)

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (GitWork)

$ git add hello.xml

git commit -m "Add and update hello.xml in GitWork"

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

[GitWork 94670eb] Add and update hello.xml in GitWork

1 file changed, 1 insertion(+)

create mode 100644 hello.xml

User@LAPTOP-GFV793Q2 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)

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ echo "<message>Completely different content from master branch</message>" > hello.xml

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git add hello.xml

git commit -m "Add hello.xml in master with different content"

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

[main af437a9] Add hello.xml in master with different content

1 file changed, 1 insertion(+)

create mode 100644 hello.xml

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

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

\* af437a9 (HEAD -> main) Add hello.xml in master with different content

| \* 94670eb (GitWork) Add and update hello.xml in GitWork

|/

\* ed36a0c Add branch-specific files in GitNewBranch

\* d60d734 (origin/main) Add .gitignore to ignore .log files and log folder

\* b7ad822 Add welcome.txt

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git diff GitWork

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

**index 4af9463..d69ab0a 100644**

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

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

@@ -1 +1 @@

-<message>Hello again from GitWork branch - updated</message>

+<message>Completely different content from master branch</message>

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git difftool GitWork

This message is displayed because 'diff.tool' is not configured.

See 'git difftool --tool-help' or 'git help config' for more details.

'git difftool' will now attempt to use one of the following tools:

opendiff kdiff3 tkdiff xxdiff meld kompare gvimdiff diffuse diffmerge ecmerge p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff

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

Launch 'vimdiff' [Y/n]? y

2 files to edit

User@LAPTOP-GFV793Q2 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.

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git add hello.xml

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git commit

On branch main

Your branch is ahead of 'origin/main' by 4 commits.

(use "git push" to publish your local commits)

Untracked files:

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

.hello.xml.swp

.hello\_LOCAL\_1536.xml.swp

.hello\_REMOTE\_1536.xml.swp

hello\_BACKUP\_1536.xml

hello\_BASE\_1536.xml

hello\_LOCAL\_1536.xml

hello\_REMOTE\_1536.xml

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

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

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

git status

On branch main

Your branch is ahead of 'origin/main' by 4 commits.

(use "git push" to publish your local commits)

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: .gitignore

Untracked files:

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

.hello.xml.swp

.hello\_LOCAL\_1536.xml.swp

.hello\_REMOTE\_1536.xml.swp

hello\_BACKUP\_1536.xml

hello\_BASE\_1536.xml

hello\_LOCAL\_1536.xml

hello\_REMOTE\_1536.xml

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

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git add .gitignore

git commit -m "Ignore backup files"

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

[main 85c119d] Ignore backup files

1 file changed, 1 insertion(+)

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git branch

GitWork

\* main

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git branch -d GitWork

Deleted branch GitWork (was 94670eb).

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

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

\* 85c119d (HEAD -> main) Ignore backup files

\* 10d9cdf Merge branch 'GitWork'

|\

| \* 94670eb Add and update hello.xml in GitWork

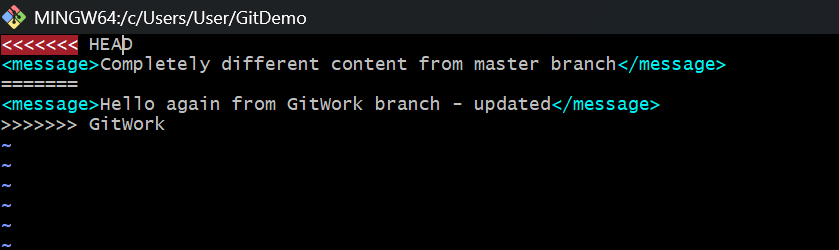
\* | af437a9 Add hello.xml in master with different content

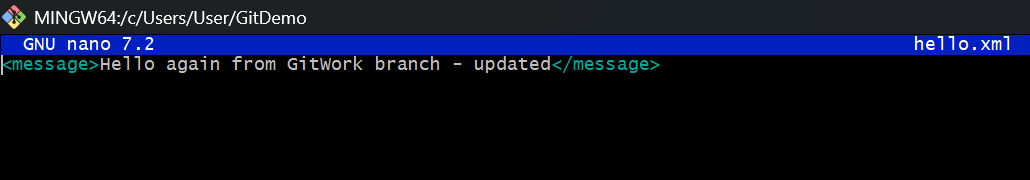
|/

\* ed36a0c Add branch-specific files in GitNewBranch

\* d60d734 (origin/main) Add .gitignore to ignore .log files and log folder

\* b7ad822 Add welcome.txt





**Hands-On 5 :-**

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git status

On branch main

Your branch is ahead of 'origin/main' by 6 commits.

(use "git push" to publish your local commits)

nothing to commit, working tree clean

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git branch temp

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git checkout temp

Switched to branch 'temp'

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (temp)

$ git checkout main

git merge temp

Switched to branch 'main'

Your branch is ahead of 'origin/main' by 6 commits.

(use "git push" to publish your local commits)

Already up to date.

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git push origin main

Enumerating objects: 19, done.

Counting objects: 100% (19/19), done.

Delta compression using up to 12 threads

Compressing objects: 100% (15/15), done.

Writing objects: 100% (17/17), 1.65 KiB | 843.00 KiB/s, done.

Total 17 (delta 6), reused 0 (delta 0), pack-reused 0

To https://gitlab.com/gitdemo1001732/GitDemo.git

d60d734..5464863 main -> main

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git branch -d temp

Deleted branch temp (was 5464863).

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git push origin --delete temp

error: unable to delete 'temp': remote ref does not exist

error: failed to push some refs to 'https://gitlab.com/gitdemo1001732/GitDemo.git'

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ ^C

User@LAPTOP-GFV793Q2 MINGW64 ~/GitDemo (main)

$ git pull origin main

From https://gitlab.com/gitdemo1001732/GitDemo

\* branch main -> FETCH\_HEAD

Already up to date.

