**GIT HANDS\_ON-Week-8**

**GIT-HANDS\_ON-1:**

git –version

git version 2.45.1.windows.1

git config --global user.name "abc”

git config --global user.email “[abc@example.com](mailto:abc@example.com)”

git config –list

user.name=abc

user.email=abc@example.com

notepad++

alias np='notepad++'

notepad++ ~/.bash\_profile

git config --global core.editor "notepad++ -multiInst -notabbar -nosession -noPlugin"

git config --global -e

[core]

editor = notepad++ -multiInst -notabbar -nosession -noPlugin

mkdir GitDemo

cd GitDemo

git init

ls -a

. .. .git

echo "Welcome to Git Demo" > welcome.txt

ls

welcome.txt

cat welcome.txt

git status

git add welcome.txt

git commit

Added welcome.txt file

This file contains a welcome message for the Git demo.

git status

git remote add origin <https://gitlab.com/yourusername/GitDemo.git>

git pull origin master

git push origin master

**GIT HANDS\_ON-2:**

GitDemo

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

mkdir log

echo "Folder log file" > log/error.log

.gitignore:

notepad++ .gitignore

\*.log

log/

git status

**OUTPUT:**

On branch master

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

Untracked files:

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

.gitignore

git add .gitignore

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

git push origin master

**GIT-HANDS\_ON-3:**

**GitNewBranch:**

git branch GitNewBranch

git branch -a

\* master

GitNewBranch

remotes/origin/master

git checkout GitNewBranch

echo "This is a new file in GitNewBranch" > branchfile.txt

git add branchfile.txt

git commit -m "Added branchfile.txt in GitNewBranch"

git status

git checkout master

git diff master GitNewBranch

git difftool master GitNewBranch

git merge GitNewBranch

git log --oneline --graph –decorate

**OUTPUT:**

\* 8b91a2d (HEAD -> master) Merge branch 'GitNewBranch'

|\

| \* 4c3d8fa (GitNewBranch) Added branchfile.txt in GitNewBranch

\* 91b1f23 Initial commit

git branch -d GitNewBranch

git status

**OUTPUT:**

On branch master

nothing to commit, working tree clean

**GIT-HANDS\_ON-4:**

git checkout master

git status

**OUTPUT:**

On branch master

nothing to commit, working tree clean

git branch GitWork

git checkout GitWork

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

**hello.xml:**

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

git status

git add hello.xml

git commit -m "Added hello.xml in GitWork branch"

git checkout master

**hello.xml:**

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

git add hello.xml

git commit -m "Added hello.xml in master branch"

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

git diff master GitWork

git difftool master GitWork

git merge GitWork

hello.xml:

<<<<<<< HEAD

<message>Hello from master branch</message>

=======

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

>>>>>>> GitWork

git mergetool

<message>Hello from master and GitWork branch</message>

git add hello.xml

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

git status

echo "\*.orig" >> .gitignore

git add .gitignore

git commit -m "Added .gitignore to exclude backup files"

git branch

git branch -d GitWork

git log --oneline --graph –decorate

**OUTPUT:**

\* 82d2a7a (HEAD -> master) Resolved merge conflict in hello.xml

|\

| \* 5fcd02a (GitWork) Added hello.xml in GitWork branch

\* | 7c1a0ef Added hello.xml in master branch

|/

\* 1f72c3b Initial commit

**GIT-HANDS\_ON-5:**

git checkout master

git status

On branch master

nothing to commit, working tree clean

git branch -a

\* master

GitWork

remotes/origin/master

git pull origin master

git push origin master

git checkout Git-T03-HOL\_002

git push origin Git-T03-HOL\_002

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