**Digital Nurture 4.0**

Week 8 – Git

# Mandatory Exercises:

**File Name: 1.Git-HOL**

**Steps:**

* Install the Git and Open the Git bash .
* Run this command in command prompt or Git Bash

git -version

* Configure Git username, email, and editor

git config --global user.name "Hari"

git config --global user.email "solohariharan@gmail.com"

git config --global core.editor "'/c/Program Files/Notepad++/notepad++.exe' -multiInst -nosession -noPlugin"

* Create project folder and initialize repository

mkdir mydemoproject

cd mydemoproject

git init

* Create and stage file

echo "Welcome to Git " > welcome.txt

git add welcome.txt

* Commit file

git commit -m "Add welcome.txt"

* Link to remote GitHub repo

git remote add origin [https://github.com/<username>/mydemoproject.git](https://github.com/%3cusername%3e/mydemoproject.git)

* Push to remote

git push -u origin master

* Modify file, stage, and commit changes

notepad++ welcome.txt

* It opens the notepad++ then

git add welcome.txt

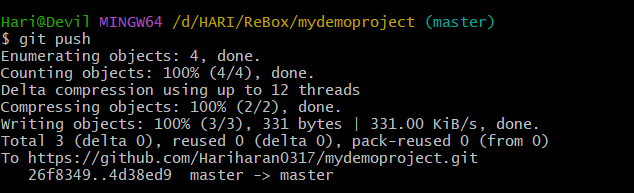
git commit -m "Update welcome message"

* Push updated changes

git push

* Pull changes from remote

git pull origin master

GIT BASH TERMINAL:

**File Name : 2.Git-HOL**

**Steps:**

* Go to your project folder

cd /d/HARI/ReBox/mydemoproject

* Create .log file and log folder

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

mkdir log

echo "Inside log folder" > log/sample.txt

* Create .gitignore file

notepad++ .gitignore

* Add these lines in the notepad++ file

\*.log

log/

* Save and close the notepad++
* Stage .gitignore file

git add .gitignore

* Commit .gitignore

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

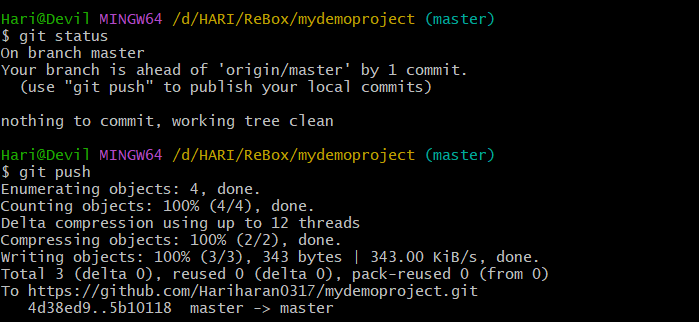
* Check status to verify ignored files

git status

* Push changes to remote

git push

GIT BASH TERMINAL:



**File name: 3.Git-HOL**

**Steps:**

* Create a new branch

git branch GitNewBranch

* List all branches (local & remote)

git branch -a

* Switch to the new branch

git checkout GitNewBranch

* Add a file in the branch

echo "Content in new branch" > branchfile.txt

git add branchfile.txt

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

git status

* Switch back to master

git checkout master

* View differences in CLI

git diff master GitNewBranch

* View differences in GUI (P4Merge)

git difftool master GitNewBranch

* Merge branch into master

git merge GitNewBranch

* View merge history

git log --oneline --graph –decorate

* Delete the merged branch

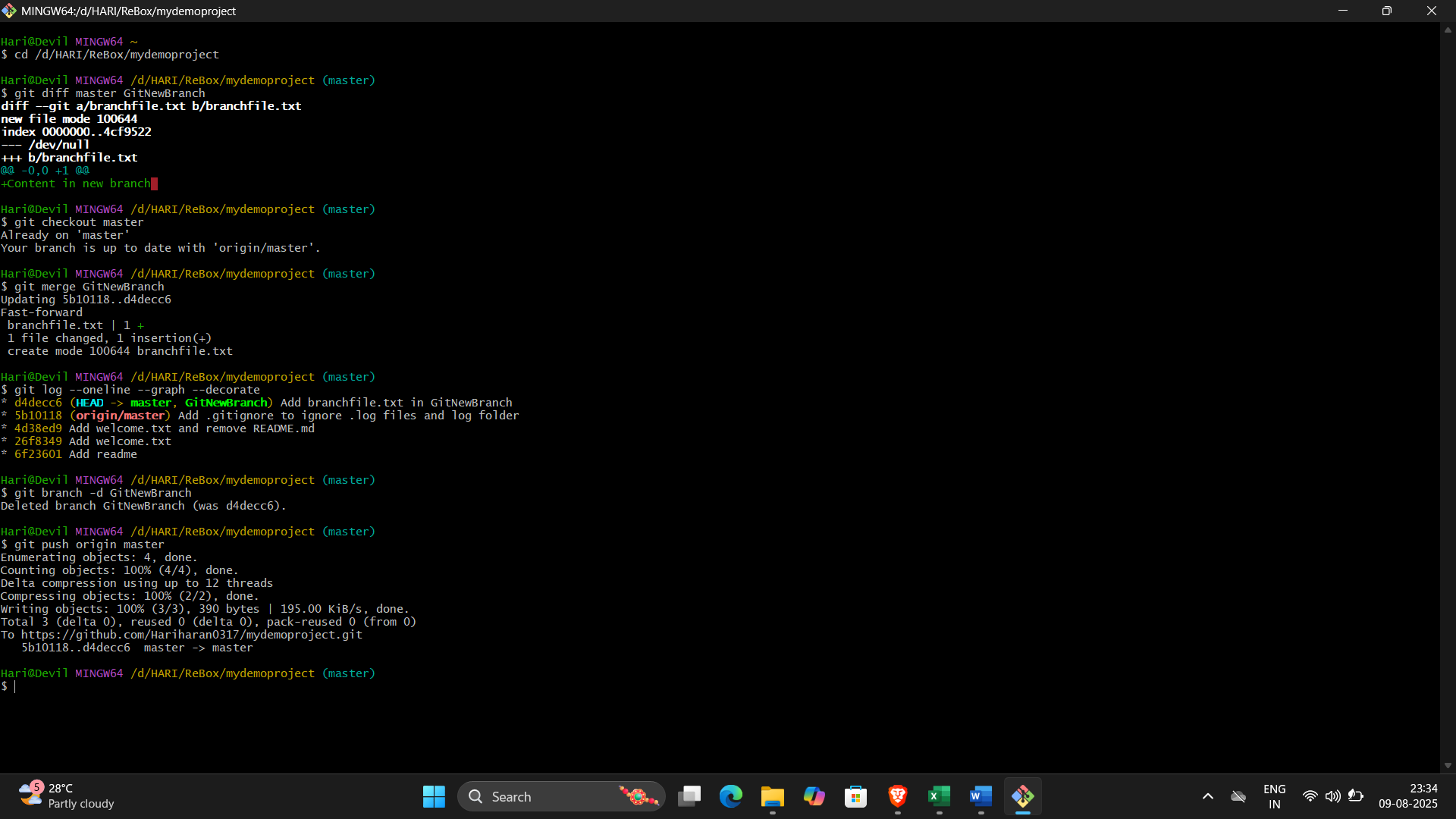
git branch -d GitNewBranch

git status

* Push merged changes to remote

git push origin master

GIT BASH TERMINAL:



**File name: 4.Git-HOL**

**Steps:**

* Ensure master is clean

git checkout master

git status

* Create a new branch

git branch GitWork

git checkout GitWork

* Add a file in branch

echo "<msg>Hello from branch</msg>" > hello.xml

git add hello.xml

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

* Modify the file in branch

echo "<msg>Updated branch content</msg>" > hello.xml

git status

git commit -am "Update hello.xml in GitWork"

* Switch to master

git checkout master

* Add the same file in master with different content

echo "<msg>Hello from master</msg>" > hello.xml

git add hello.xml

git commit -m "Add hello.xml in master"

* View commit history

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

* Compare differences (CLI)

git diff master GitWork

* Compare differences (P4Merge)

git difftool master GitWork

* Merge branch into master (conflict will occur)

git merge GitWork

* **Resolve conflict using 3-way merge tool (P4Merge)**

git mergetool

* Commit after resolving conflict

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

* Add backup files to .gitignore

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

git add .gitignore

git commit -m "Ignore merge backup files"

* List all branches

git branch -a

* Delete merged branch

git branch -d GitWork

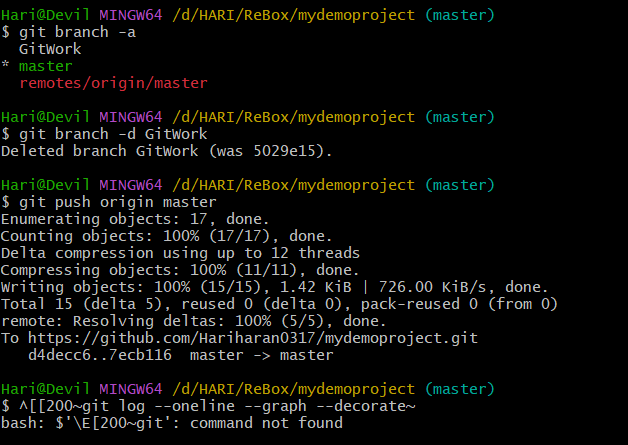
* Push changes to remote

git push origin master

* View final commit graph

git log --oneline --graph –decorate

GIT BASH TERMINAL:



**File name: 5.Git-HOL**

**Steps:**

* Ensure master is clean

git checkout master

git status

* List all branches

git branch -a

* Pull latest changes from remote

git pull origin master

* Push pending changes from previous lab (Git-T03-HOL\_002)

git push origin master

GIT BASH TERMINAL:

