**Digital Nurture 4.0 Week 8**

**GIT - HOL**

**Mandatory Hands On**

**File 1:GIT-HOL**

Step 1 – Git Configuration

1. Verify Git installation

git --version

Expected Output:

nginx

git version 2.xx.x.windows.x

1. Set your Git username and email

git config --global user.name "Saranya"

git config --global user.email "ganeshsaranya2018@gmail.com"

1. Verify configuration

git config --list

Expected Output:

ini

user.name=Saranya

user.email=ganeshsaranya2018@gmail.com

Step 2 – Integrate Notepad++ as Default Editor

1. Check if Notepad++ runs

notepad++

1. Optional alias (shortcut)

alias np='notepad++'

1. Set Notepad++ as Git’s default editor

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

1. Verify editor setting

git config --global -e

Step 3 – Create Project & Commit File

1. Create new directory and initialize Git

mkdir GitDemo

cd GitDemo

git init

Output:

Initialized empty Git repository in C:/path/to/GitDemo/.git/

1. Check .git folder exists

ls -a

1. Create a file

echo "Welcome to Git Demo" > welcome.txt

1. Check file

ls

cat welcome.txt

1. Check status

git status

1. Stage the file

git add welcome.txt

1. Commit the file (opens in Notepad++)

git commit

*(Or use a one-line commit without editor)*:

git commit -m "Initial commit: Added welcome.txt"

1. Verify

git status

Step 4 – Push to Remote GitLab

1. Create an empty repository on GitLab (same name: GitDemo).
2. Link local repo to remote

git remote add origin [Saranya\_81-group / GitDemo · GitLab](https://gitlab.com/saranya_81-group/gitdemo)

Pull remote changes (only if repo already has commits)

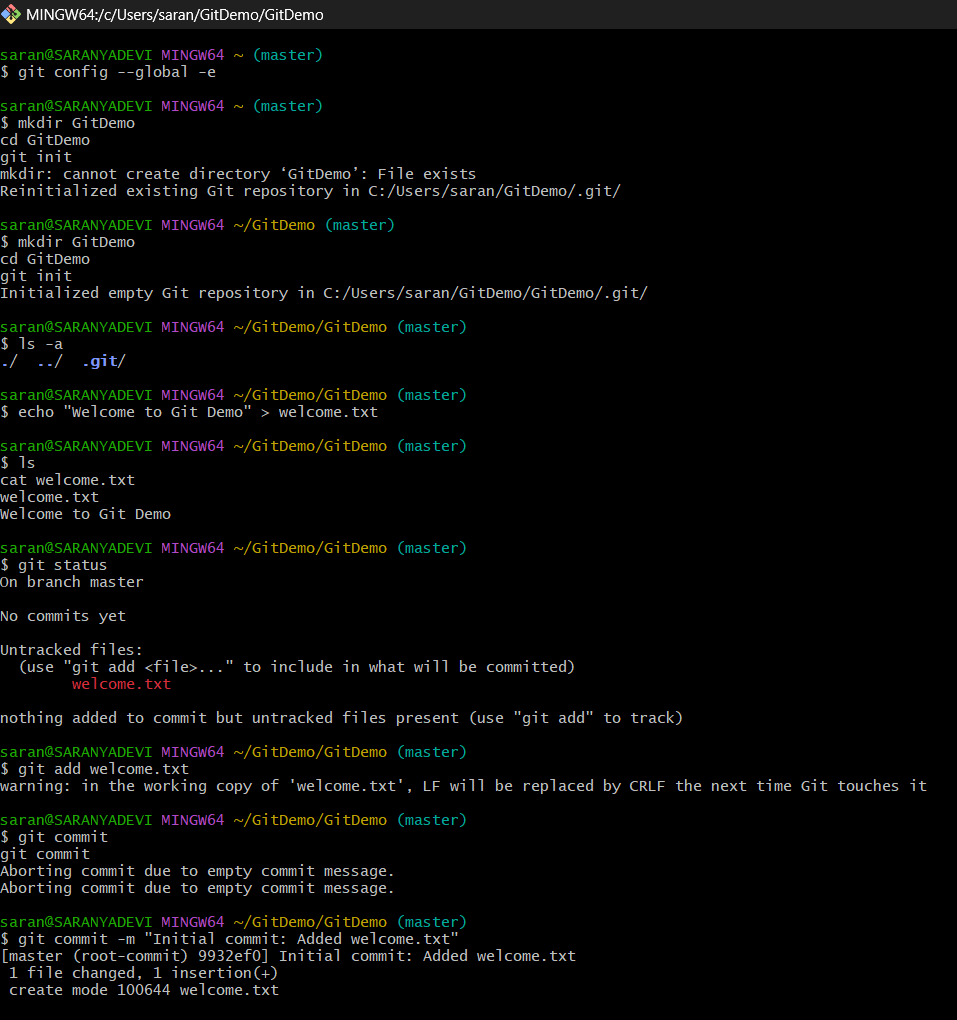
git pull origin master --allow-unrelated-histories

1. Push local commits to remote

git push -u origin master

**OUTPUT:**

****

****

**File 2:GIT-HOL**

Objectives

* Learn what .gitignore is.
* Learn how to ignore unwanted files or folders in Git.

**How to Ignore Files Using**

**.gitignore**

You **list patterns** in .gitignore for the files you want Git to ignore.

\*.log

log/

STEPS:

**Step 1 – Create a .log file and a log folder**

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

mkdir log

echo "Log details here" > log/debug.txt

Now your folder looks like:

lua

GitDemo/

error.log

log/debug.txt

**Step 2 – Create/Edit .gitignore**

notepad++ .gitignore

Add these lines:

\*.log

log/

Save and close.

**Step 3 – Check Git Status**

git status

Expected:  
error.log and log/debug.txt will **NOT** appear in "Untracked files" because Git is ignoring them.

**Step 4 – Commit Changes**

1. Add .gitignore:

git add .gitignore

1. Commit:

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

1. Push:

git push

**OUTPUT:**