Objective:

In this task, I have learned how to configure Git on my local machine, integrate Notepad++ as the default editor, and create and manage a Git repository using essential Git commands such as git init, git add, git commit, git status, git push, and git pull.

Step 1: Setup Git Configuration on My Machine

First, I opened Git Bash to check whether Git is installed on my system by running the command:

git --version

This returned the installed Git version, which confirmed Git was properly set up.

Then, I configured my user name and email ID using the following commands:

git config --global user.name "My Name"

git config --global user.email "myemail@example.com"

To verify that these configurations were saved successfully, I ran:

git config --list

This displayed the configured user name and email.

Step 2: Setting Notepad++ as Default Git Editor

I checked if Notepad++ could be opened from Git Bash by typing:

notepad++

It showed an error, which meant Notepad++ was not added to the system path.

So, I added the Notepad++ path to the Environment Variables:

Opened Control Panel > System > Advanced System Settings

Went to Environment Variables

Edited the Path variable and added:

C:\Program Files\Notepad++

After saving and restarting Git Bash, I confirmed that Notepad++ opens by running:

notepad++

I then set Notepad++ as the default Git editor using:

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

Finally, I verified the editor configuration with:

git config --global -e

This command opened the config file in Notepad++, confirming successful integration.

Step 3: Creating and Managing a Git Repository

I created a new folder and initialized it as a Git repository:

mkdir GitDemo

cd GitDemo

git init

I confirmed Git was initialized by running:

ls -a

This showed the .git folder.

I created a new file called welcome.txt:

echo "Welcome to Git Hands-on Lab" > welcome.txt

I listed the files using:

ls

To read the content inside welcome.txt, I ran:

cat welcome.txt

Then, I checked the status of the file:

git status

It showed that the file was untracked.

I added the file to staging:

git add welcome.txt

To commit the file with a message, I used:

git commit

This opened Notepad++, where I typed the following commit message:

Initial Commit

Added welcome.txt with a sample message

I verified that the file was committed by checking:

git status

It showed no changes, meaning the working directory was clean.

Step 4: Pushing to Remote Repository on GitLab

I logged into GitLab using my personal account and created a new repository named GitDemo.

Then I linked my local repository with the remote repository:

git remote add origin https://gitlab.com/your-username/GitDemo.git

To make sure everything was up to date, I pulled from the remote (optional):

git pull origin master

Finally, I pushed my local commits to the remote repository:

git push origin master

This completed the Git push process successfully.

Conclusion:

In this hands-on task, I successfully:

Installed and configured Git on my local system

Integrated Notepad++ as my default Git editor

Created a Git repository and tracked a file

Pushed the repository to GitLab

This exercise helped me understand the Git workflow and how to manage source code efficiently using Git.