**Step 1: Git Client Setup and Configuration**

**1.1 Verifying Git Installation**

To begin, I verified that Git was installed correctly on my system by executing the following command in Git Bash:

git --version

**Output:**

git version 2.41.0.windows.1

If Git is not installed, it can be downloaded from the official website: <https://git-scm.com>

**1.2 Configuring User Details in Git**

Git tracks all changes by associating them with a user's name and email ID. These need to be configured as follows:

git config --global user.name "Koushika"

git config --global user.email "koushika.kolipineni@gmail.com"

The --global flag ensures that these settings apply across all repositories on the system.

**1.3 Verifying Git Configuration**

To confirm that the user details have been set correctly, the following command was executed:

git config --global --list

**Output:**

user.name=Your Name

user.email=your.email@example.com

**Step 2: Integrating Notepad++ with Git**

**2.1 Checking Notepad++ Execution from Git Bash**

To use Notepad++ as Git's default editor, it must be accessible from the Git Bash shell. This was checked with:

notepad++

If Git Bash returns an error like 'notepad++' is not recognized, it means Notepad++ is not added to the system’s PATH variable.

**2.2 Adding Notepad++ to System PATH**

Steps to add Notepad++ to the PATH environment variable:

1. Open Control Panel → System and Security → System
2. Click on Advanced System Settings
3. Click on Environment Variables
4. Under User Variables, select Path and click Edit
5. Click New and add:

C:\Program Files\Notepad++

1. Click OK and restart Git Bash

**2.3 Creating an Alias for Notepad++**

To make launching Notepad++ simpler, an alias was created by adding the following to the .bashrc file:

alias np='notepad++'

Then the shell was refreshed using:

source ~/.bashrc

Now, typing np in Git Bash opens Notepad++.

**2.4 Configuring Git to Use Notepad++ as Default Editor**

To make Notepad++ the default editor for Git:

git config --global core.editor "notepad++"

**2.5 Verifying Default Editor**

To confirm that Notepad++ is set as the default editor:

git config --global --edit

This command opens the Git configuration file in Notepad++, confirming the integration.

**Step 3: Creating and Managing a Local Repository**

**3.1 Initializing a Local Git Repository**

A new folder was created and initialized as a Git repository using:

mkdir GitDemo

cd GitDemo

git init

**Output:**

Initialized empty Git repository in C:/Users/Koushika-lab/GitDemo/.git/

**3.2 Verifying Repository Initialization**

The presence of the .git directory was confirmed with:

ls -a

**3.3 Creating a Text File**

To create a file and add content:

echo "Welcome to Git" > welcome.txt

**3.4 Verifying File Creation and Content**

To list and view the contents of the file:

ls

cat welcome.txt

**3.5 Checking Git Status**

git status

Output showed welcome.txt as an untracked file.

**3.6 Adding File to Staging Area**

git add welcome.txt

This staged the file for commit.

**3.7 Committing Changes with Notepad++ Editor**

git commit

This command opened Notepad++ to enter a multi-line commit message:

Initial commit

- Added welcome.txt

- Initialized Git repository

After saving and closing Notepad++, the commit was registered.

**3.8 Verifying Commit**

git status

The working directory was confirmed as clean.

**Step 4: Creating and Connecting to Remote Repository (GitLab)**

**4.1 Creating GitLab Repository**

Logged in to <https://gitlab.com> and created a new project named GitDemo. The HTTPS clone URL was copied, e.g.:

https://gitlab.com/yourusername/GitDemo.git

**4.2 Adding Remote Origin**

Inside the local GitDemo folder:

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

**4.3 Pulling from Remote Repository**

If remote repository has initial commits (like README):

git pull origin master --allow-unrelated-histories

**4.4 Pushing Local Repository to GitLab**

git push -u origin master

This uploaded the local files and commit history to the GitLab repository.

**4.5 Verifying Upload**

Logged into GitLab and verified that welcome.txt appeared in the repository.

**OUTPUT:**  
  
