1. **Git-HOL**

**CODE:**

Step 1: Install and Configure Git

* Check Git is Installed

git --version

* Set your Name and Email

git config --global user.name "Nisha"

git config --global user.email "[nishasrini1433@gmail.com](mailto:nishasrini1433@gmail.com) "

* Verify Git Configuration

git config –list

Step 2: Set Notepad++ as Default Editor

* Add Notepad++ to System Path
* Try Running Notepad++ from Git Bash

notepad++

* Set Notepad++ as Git’s Default Editor

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

Step 3: Create Local Git Project

* Create a Folder for Your Project

mkdir GitDemo

cd GitDemo

* Initialize Git



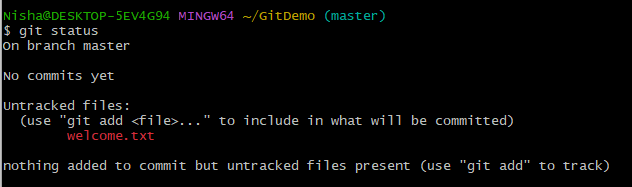
* Create a New File

echo "Welcome to Git hands-on lab!" > welcome.txt

* View the File

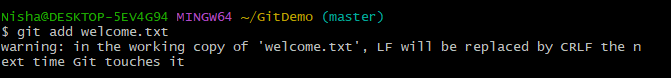


* **Check Git status**

****

Step 4: Stage and Commit the File

* Add the File to Staging Area



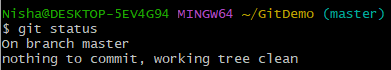
* Commit the File (Opens Notepad++)

git commit

Notepad++ opens → type commit message like:

Initial commit: Added welcome.txt file

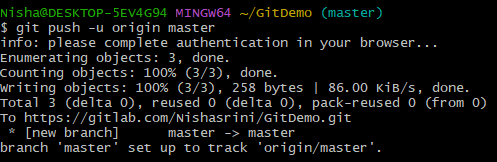
* Check Status Again



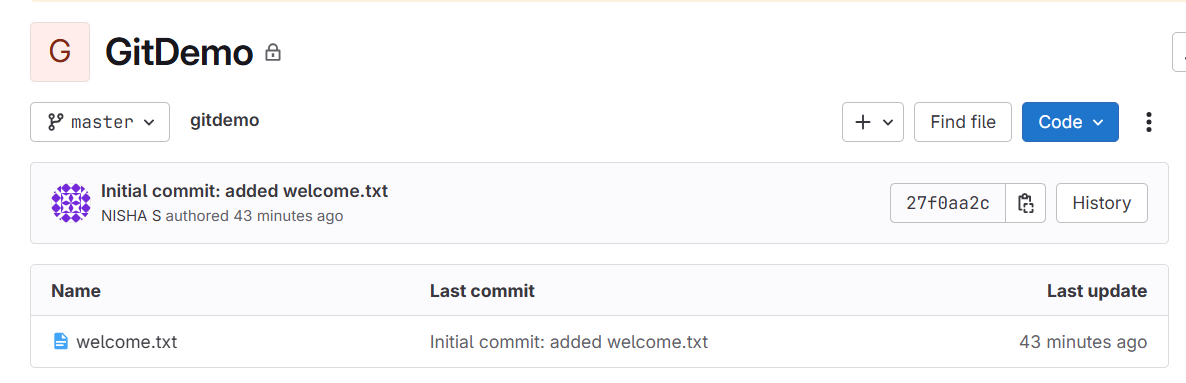
Step 5: Connect to GitLab and Push Code

* Login to [GitLab.com](https://gitlab.com)
* Create a New Project
* Connect Local Project to Remote



* Push Your Code to GitLab

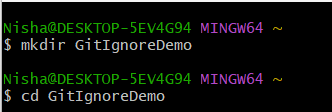
**OUTPUT:**



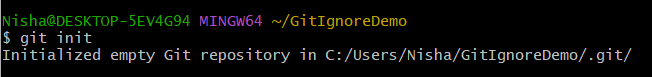
1. **Git-HOL**

**CODE WITH OUTPUT:**

* Create a Folder for Your Project:



* Initialize Git:



* Creates an empty .gitignore file:



* Open .gitignore file in Notepad++ to edit:



Add this in Notepad++:

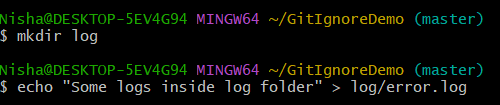
\*.log

log/

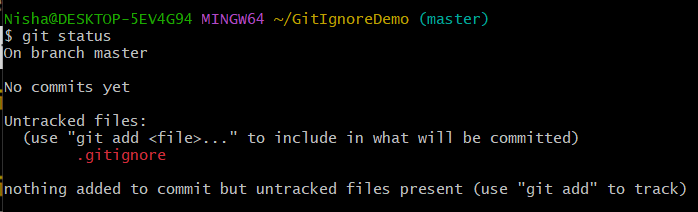
* Creates a sample log file:



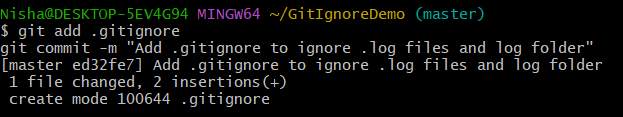
* Create a log folder and log file inside the log folder:



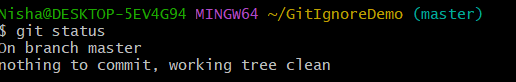
* Current status of files:



* Add and commit .gitignore file:



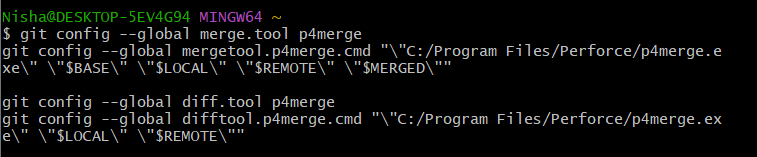
* Again check status:



1. **Git-HOL**

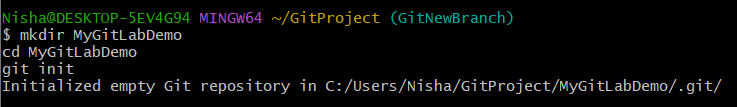
**CODE WITH OUTPUT:**

* Prerequisites:



Branching:

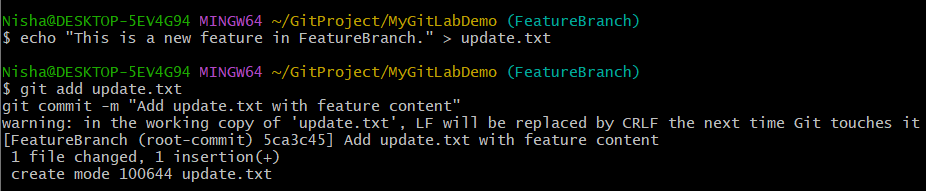
* Create a project and initialize repository:



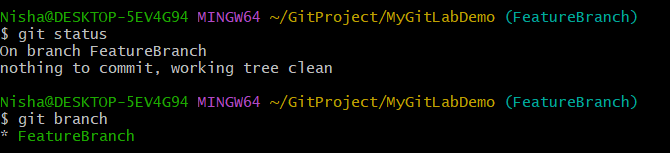
* Create a branch:



* Add and Commit a file:

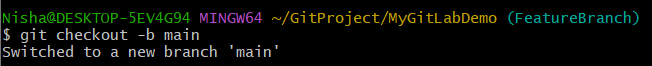


* Verify branch and status:

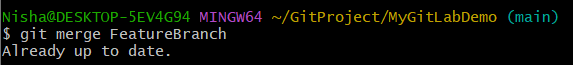


Merging:

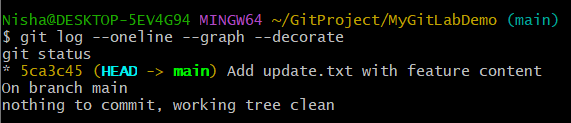
* Switch to base branch:



* Merge feature branch and main:



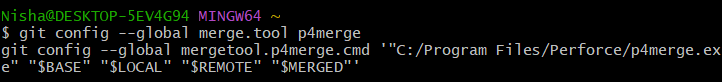
* Check log graphically:



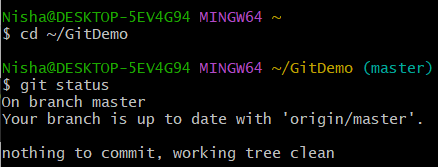
1. **Git-HOL**

**CODE WITH OUTPUT:**

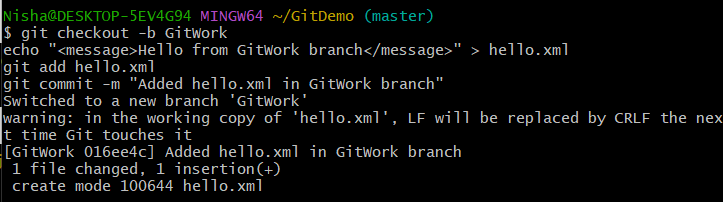
* Make sure P4Merge is configured



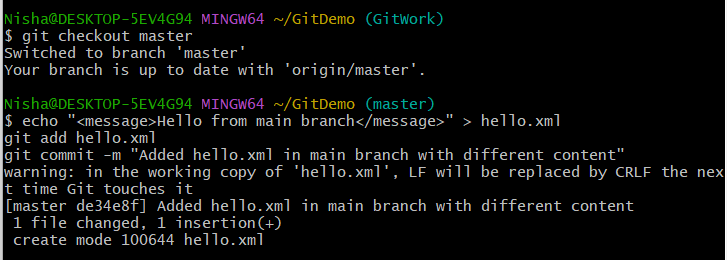
* Start in clean main branch



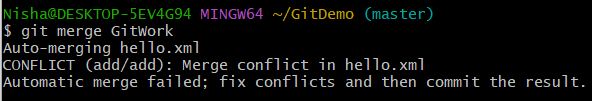
* Create GitWork branch and add file



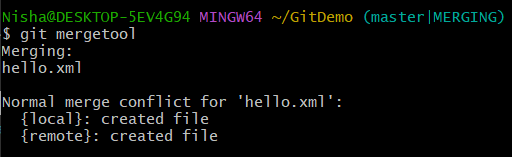
* Switch to main branch and add different file



* Merge GitWork into main (conflict will occur)



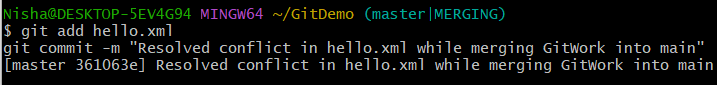
* Resolve conflict using P4Merge



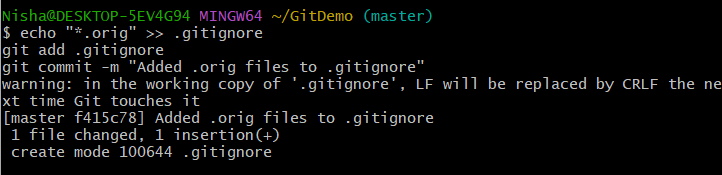
Add it in P4Merge:

<message>Hello from main branch and updated GitWork content</message>

* Commit the merge



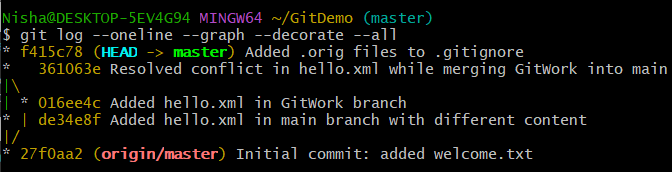
* Add .gitignore for backup files



* Delete merged branch



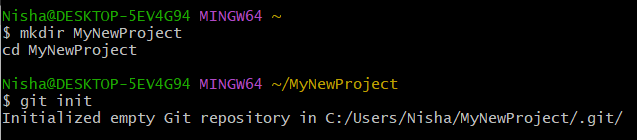
* View final history



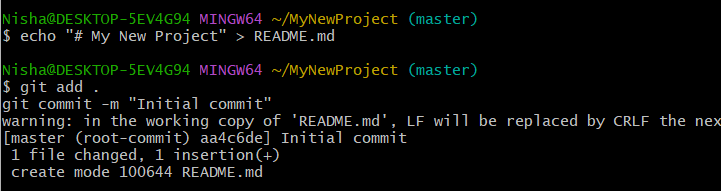
1. **Git-HOL**

**CODE :**

* Create project locally and initialize Git



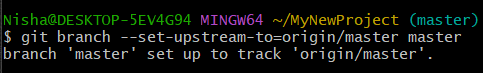
* Create files and make the first commit



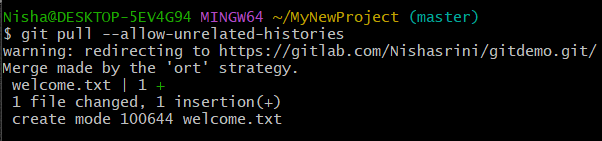
* Link to GitLab remote



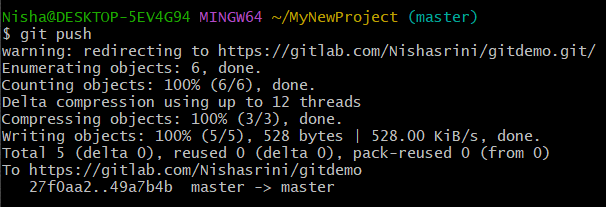
* Set upstream branch



* Pull remote content and merge histories



* Push to remote



**OUTPUT:**

