Digital Nurture 4.0 – Week 8

GIT

Filename : 1. Git-HOL

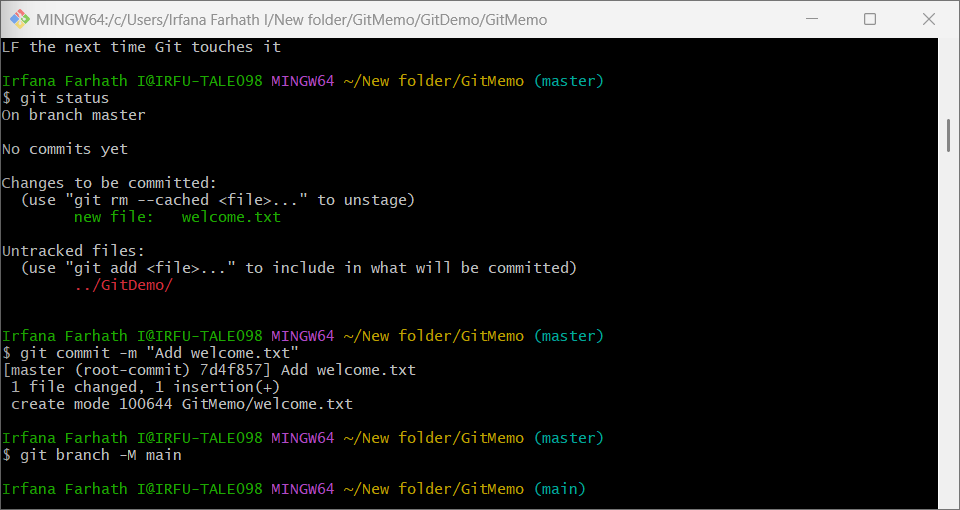
**Objective:**

To Built a Git bash and logging with details and creating directory and repositories,add files and Gitlab with various operation.

**Steps:**

1. Install Git.
2. Check Git version:  
    git --version
3. Set up Git user config:  
    git config --global user.name "irfana"  
    git config --global user.email "irfana@gmail.com"
4. Create new directory & initialize repo:  
    mkdir GitDemo  
    cd GitDemo  
    git init
5. Create and add file:  
    echo "Welcome to Git" > welcome.txt  
    git status  
    git add welcome.txt
6. Commit with message:  
    git commit
7. Set up remote GitLab repository:  
    git remote add origin <https://gitlab.com/irfananoorjahan333-group/irfananoorjahan333-project.git>
8. Pull from and push to remote:  
    git pull origin master  
    git push origin master

**Output:**



Filename : 2. Git-HOL

**Objective:**

To Create a GitMemo and add log file with folders And creating the ignore files ,check status add gitognore files.

**Steps:**

1. Inside the GitDemo project directory:  
 cd GitDemo

2.Create .log file and log folder:  
 touch example.log

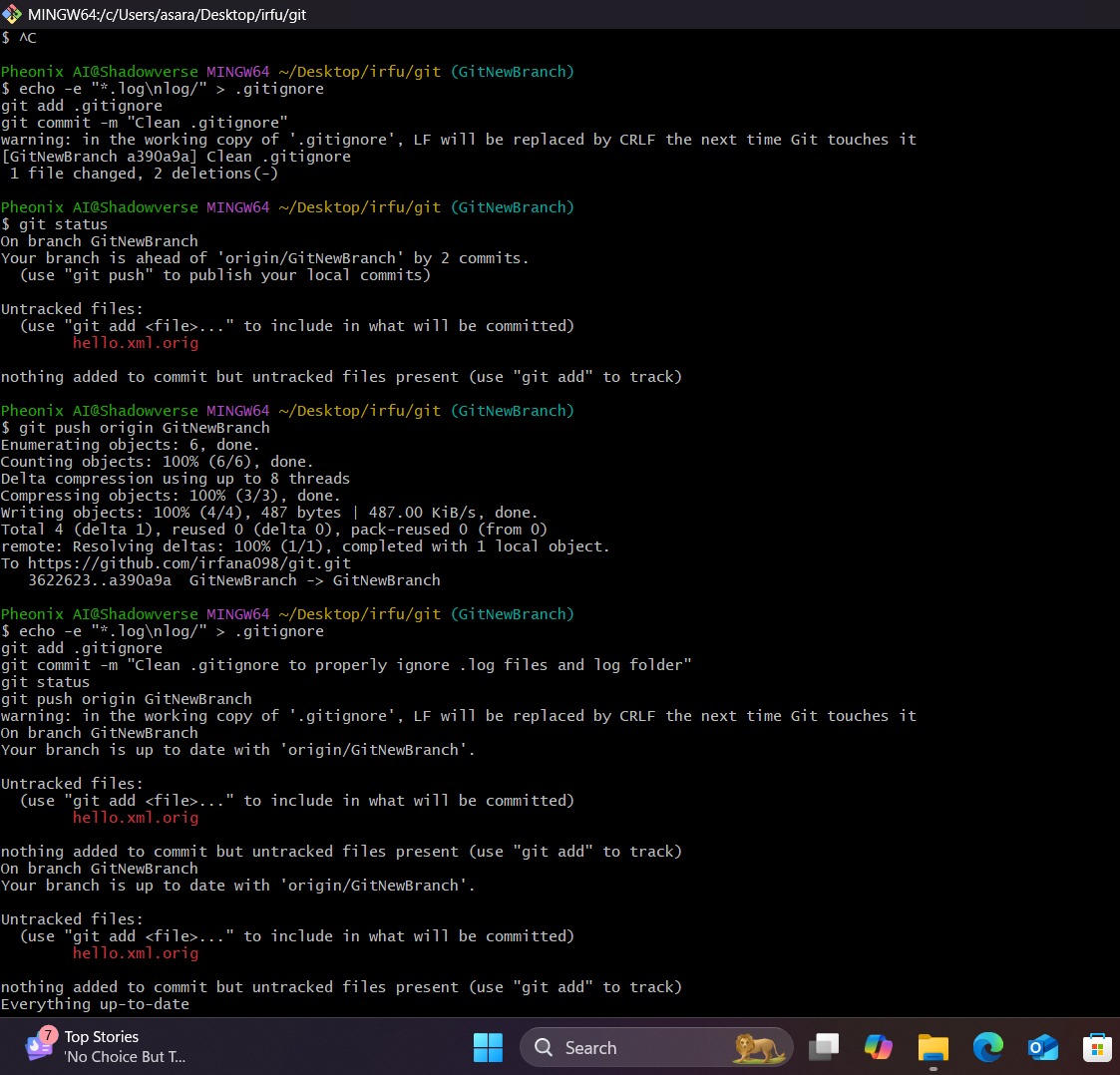
mkdir log

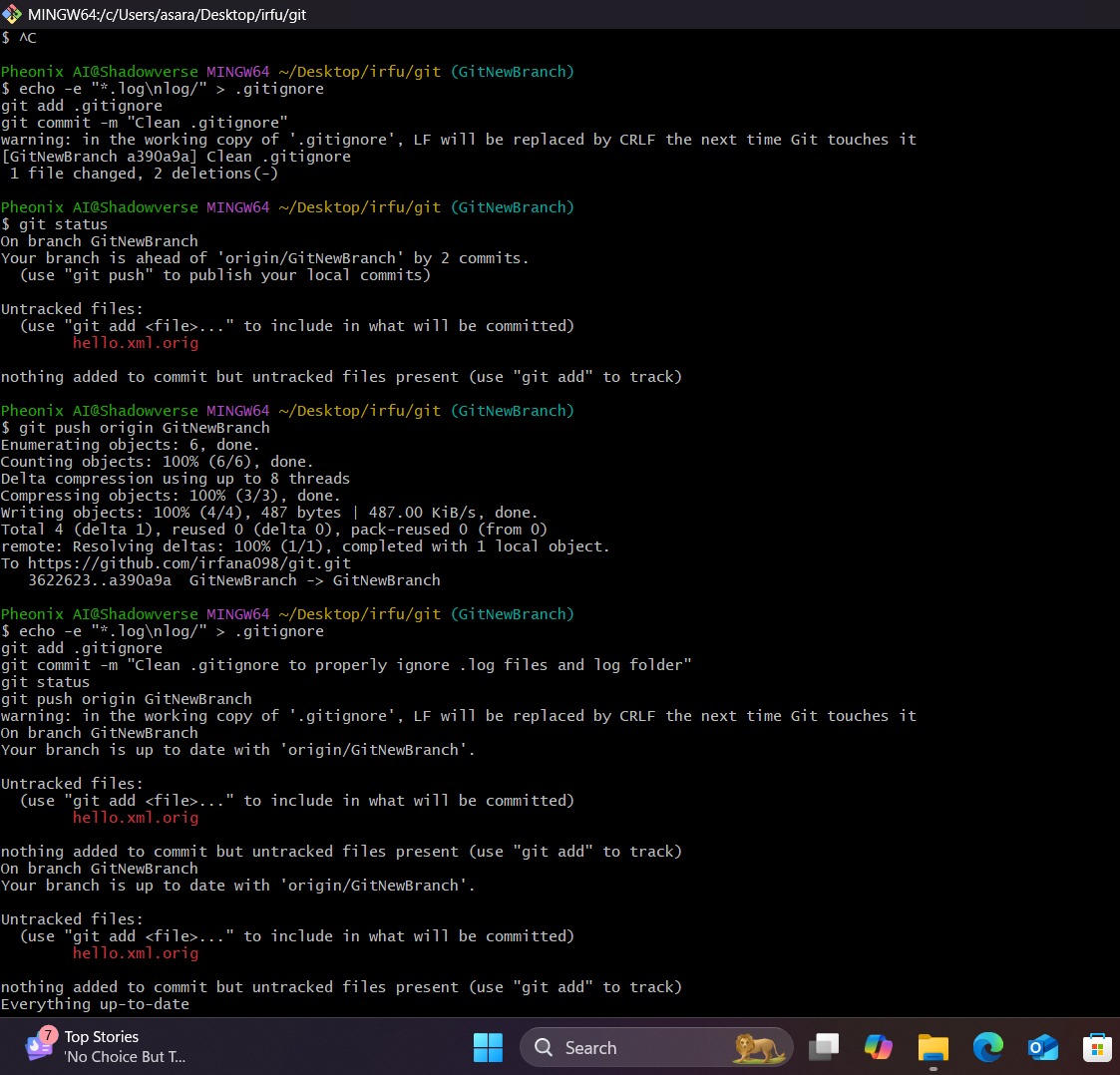
3.Create .gitignore file:  
 echo "\*.log" > .gitignore  
 echo "log/" >> .gitignore

4.Check Git status:  
 git status

5.Add .gitignore and commit:  
 git add .gitignore  
 git commit -m "Ignore .log files and log folder"

**Output:**





Filename : 3. Git-HOL

**Objective:**

To Create a branches and switches , Understand the concept and purpose of branching and merging in Git.Learn how to create branches, commit changes, and merge branches back to the master (or trunk).Gain familiarity with creating branch requests and merge requests in GitLab for collaboration.

**Steps:**

1.Create branch and switch:  
 git checkout -b GitNewBranch

2.Add file and commit:  
 echo "This is a new branch file" > newbranch.txt  
 git add newbranch.txt  
 git commit -m "Added newbranch.txt in GitNewBranch"

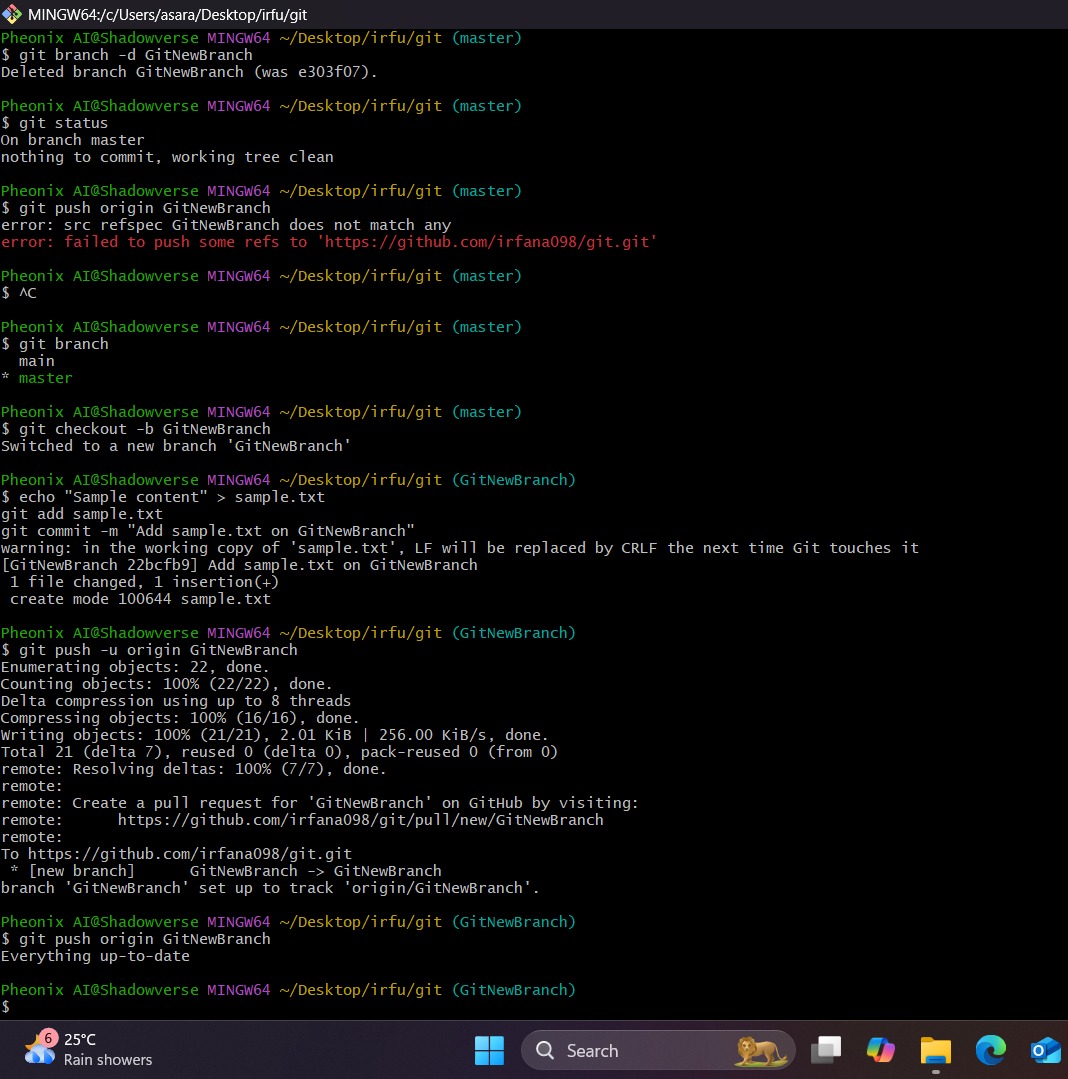
3.Check status:  
 git status

4.Switch to master and merge:  
 git checkout master  
 git diff GitNewBranch  
 git merge GitNewBranch

5.View log as graph:  
 git log --oneline --graph --decorate

6.Delete branch:  
 git branch -d GitNewBranch

**Output:**



Filename : 4. Git-HOL

**Objective:**

To demonstrate the process of cleaning up the local Git repository, synchronizing it with the remote repository, and pushing local changes back to the remote. To ensure understanding of Git commands involved in branch management, remote tracking, and authentication during push and pull operations.

**Steps:**

1.Verify master is clean:  
 git status

2.Create GitWork branch and switch:  
 git checkout -b GitWork

3.Create and commit hello.xml:  
 echo "Branch version of XML" > hello.xml  
 git add hello.xml  
 git commit -m "Added hello.xml in GitWork"

4.Switch to master and create different hello.xml:  
 git checkout master  
 echo "Master version of XML" > hello.xml  
 git add hello.xml  
 git commit -m "Added hello.xml in master"

5.View history:  
 git log --oneline --graph --decorate --all

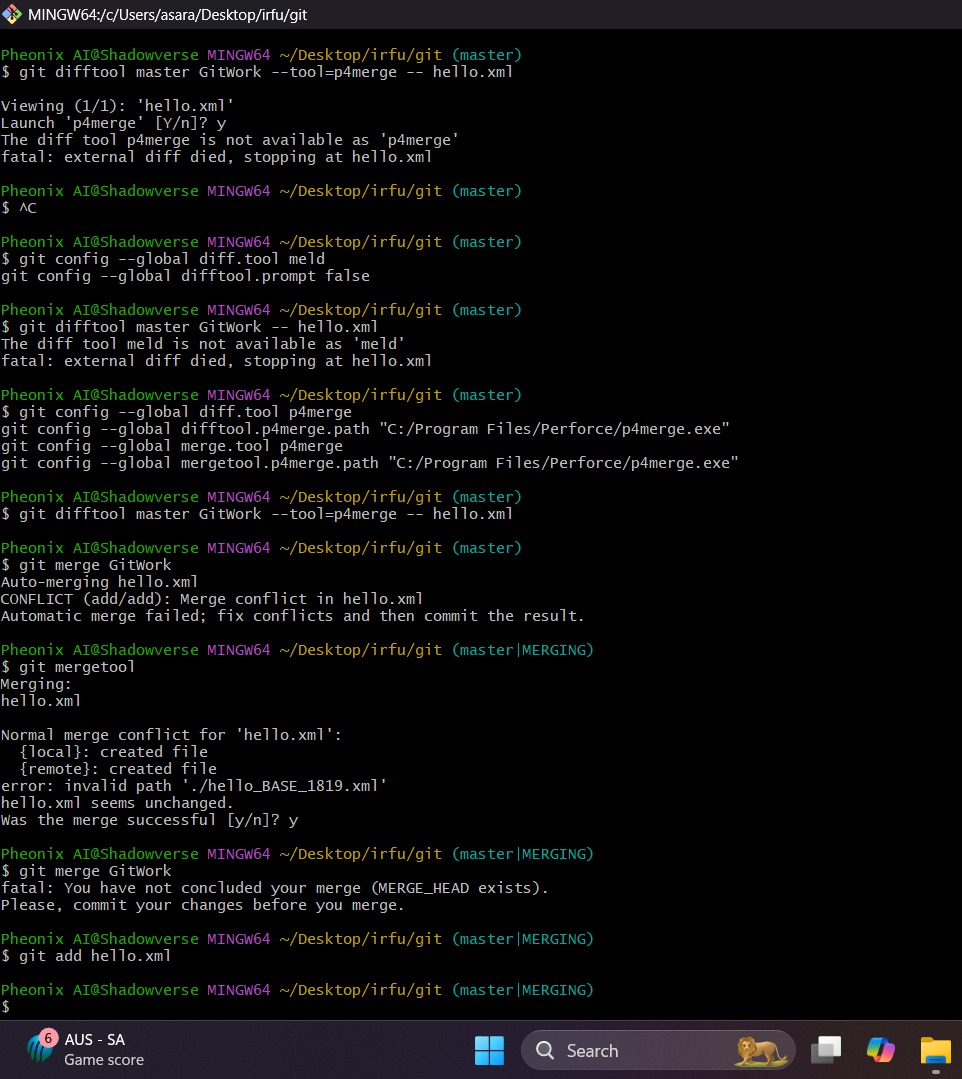
6.Check diff and attempt merge:  
 git diff GitWork  
 git merge GitWork

7.Resolve conflict manually:  
 nano hello.xml  
 git add hello.xml  
 git commit -m "Resolved conflict in hello.xml"

8.Add backup files to .gitignore:  
 echo "\*~" >> .gitignore  
 git add .gitignore  
 git commit -m "Ignore backup files"

1. Delete GitWork branch:  
    git branch -d GitWork  
    git log --oneline --graph --decorate

**Output:**



Filename : 5. Git-HOL

**Objective:**

To understand and implement conflict resolution when multiple users modify the same file differently on master and a feature branch. To gain hands-on experience with Git commands for branch management, merging, conflict detection, and resolution using command line and merge tools.

**Steps:**

1.Verify clean state:  
 git status

2.List branches:  
 git branch -a

3.Pull latest from remote:  
 git pull origin master

4.Push local changes:  
 git push origin master

5.Check changes reflected on GitLab

**Output:**

