**Cognizant Digital Nurture 4.0**

**Name: Siri Chandana Chittipolu**

**Email:** [**sirichittipolu11@gmail.com**](mailto:sirichittipolu11@gmail.com)

**Superset ID: 6386277**

**Mandatory Hands-On Exercises**

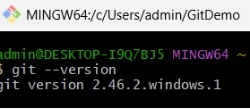
**GIT Hands-On Exercises**

1. **GIT-HOL:**

**Solution:**

**Objective:**  
Familiar with Git commands like git init, git status, git add, git commit, git push, and git pull.

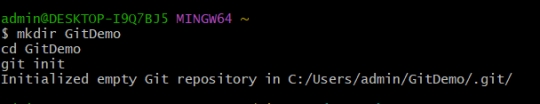
**Step 1 – Verify Git Installation**

Command:  
 git --version  
  
 Output:  
   
**Step 2 – Configure Git**

Command:  
 git config --global user.name "studentnewbie"  
 git config --global user.email "sirichittipolu11@gmail.com"  
 git config --list  
  
**Step 3 – Set Notepad as Default Git Editor**

Command:  
 git config --global core.editor "notepad"  
 git config --global -e

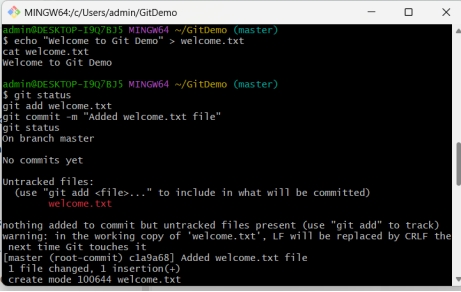
**Step 4 – Create Local Repository**

Command:  
 mkdir GitDemo  
 cd GitDemo  
 git init  
  
Output:  
 

**Step 5 – Create and View File**

Command:  
 echo "Welcome to Git Demo" > welcome.txt  
 cat welcome.txt

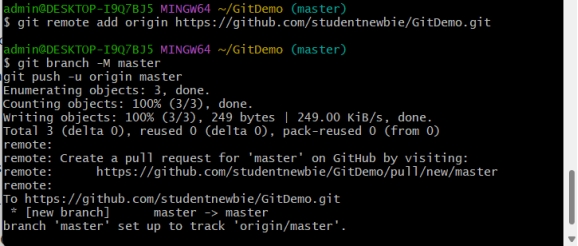
**Step 6 – Stage and Commit File**

Command:  
 git status  
 git add welcome.txt  
 git commit -m "Added welcome.txt file"  
 git status  
  
Output:  


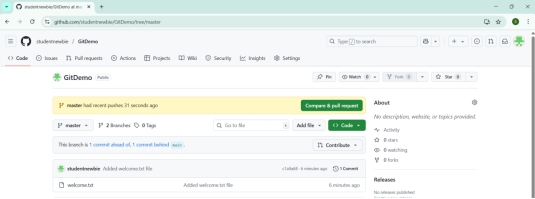
**Step 7 – Link Local Repo to GitHub**

Command:  
 git remote add origin https://github.com/studentnewbie/GitDemo.git

**Step 8 – Push to GitHub**

Command:  
 git branch -M master  
 git push -u origin master  
  
Output:  
 

**Step 9 – Verify on GitHub**

Action:  
 Open https://github.com/studentnewbie/GitDemo in your browser.  
Output:  
 

1. **GIT-HOL:**

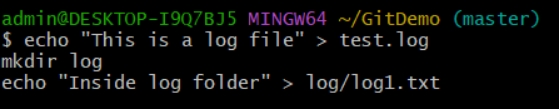
**Solution:**

**Step 1 – Open Existing Git Repository**

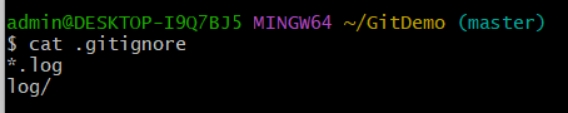
Command:  
 cd ~/GitDemo

**Step 2 – Create Files to Ignore**

Command:  
 echo "This is a log file" > test.log  
 mkdir log  
 echo "Inside log folder" > log/log1.txt  
Output:

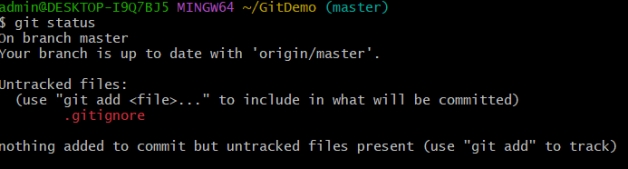


**Step 3 – Create/Edit .gitignore**

Command:  
 echo "\*.log" > .gitignore  
 echo "log/" >> .gitignore  
 cat .gitignore  
Output:  


**Step 4 – Check Git Status**

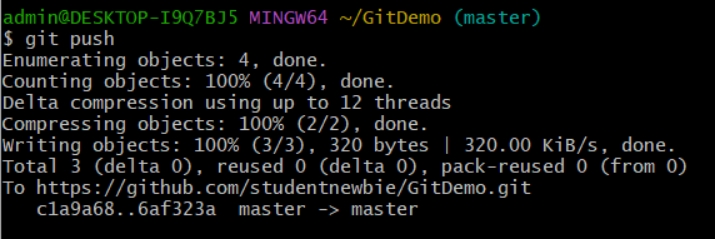
Command:  
 git status  
Output:



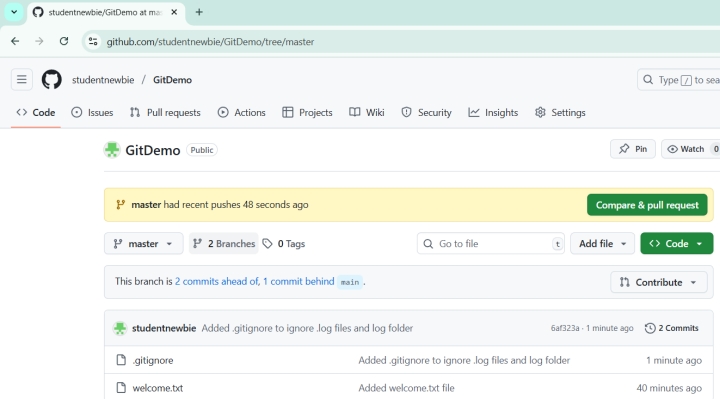
**Step 5 – Stage & Commit .gitignore**

Command:  
 git add .gitignore  
 git commit -m "Added .gitignore to ignore .log files and log folder"  
**Step 6 – Push to GitHub**

Command:  
 git push  
Output:



**Step 7 – Verify on GitHub**

Action:  
 Open the GitHub repository page.  
Output:  
 

1. **GIT-HOL:**

**Solution:**

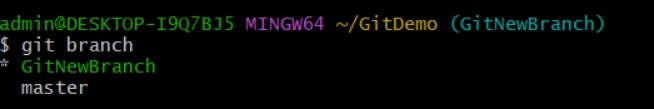
**Step 1 – Open Existing Git Repository**

**Command:**  
cd ~/GitDemo

**Step 2 – Create New Branch**

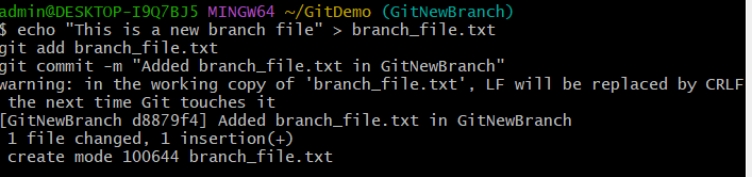
**Command:**  
git checkout -b GitNewBranch  
git branch

**Output:**



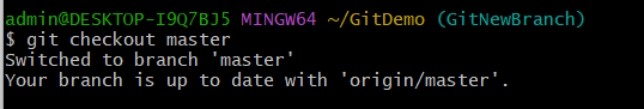
**Step 3 – Add Files to Branch**

**Command:**  
echo "This is a new branch file" > branch\_file.txt  
git add branch\_file.txt  
git commit -m "Added branch\_file.txt in GitNewBranch"

**Output:**  


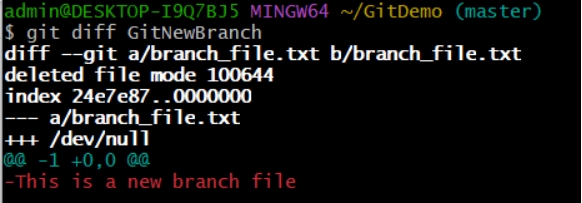
**Step 4 – Switch to Master**

**Command:**  
git checkout master

**Output:**  


**Step 5 – See Differences**

**Command:**  
git diff GitNewBranch

**Output:**  


**Step 6 – Merge Branch into Master**

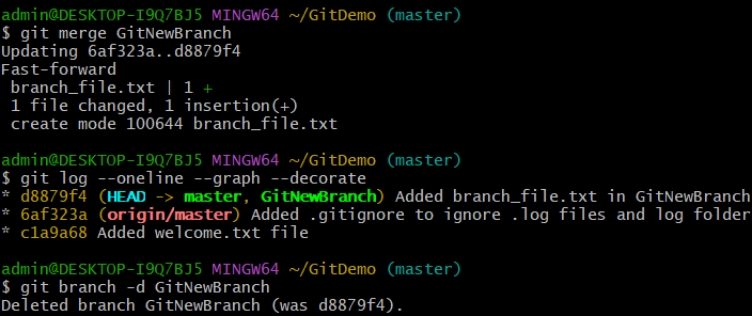
**Command:**  
git merge GitNewBranch

**Step 7 – View Log**

**Command:**  
git log --oneline --graph --decorate

**Step 8 – Delete Branch**

**Command:**  
git branch  
git branch -d GitNewBranch  
git branch

**Output:**  


1. **GIT-HOL:**

**Solution:**

**Step 1 – Verify Master is in Clean State**

**Command:**  
git status

**Output:**  
Working tree clean.

**Step 2 – Create a Branch and Add File**

**Command:**  
git checkout -b GitWork  
echo "Hello from GitWork branch" > hello.xml  
git add hello.xml  
git commit -m "Added hello.xml in GitWork"

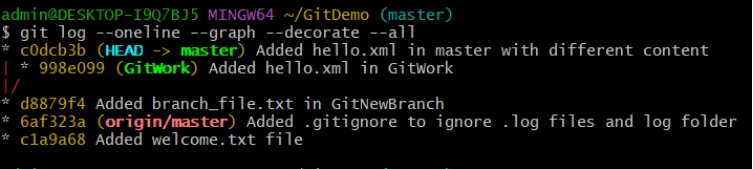
**Step 3 – Switch to Master and Add Same File with Different Content**

**Command:**  
git checkout master  
echo "Hello from master branch" > hello.xml  
git add hello.xml  
git commit -m "Added hello.xml in master with different content"

**Step 4 – View Commit Log**

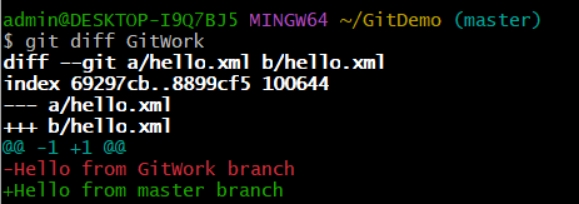
**Command:**  
git log --oneline --graph --decorate --all

**Output :**



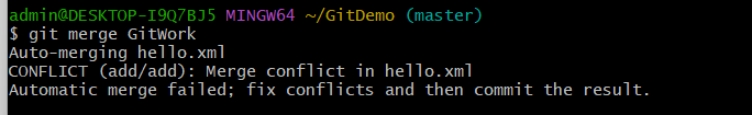
**Step 5 – Check Differences**

**Command:**  
git diff GitWork

**Output:**  


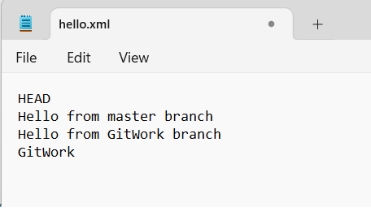
**Step 6 – Merge Branch into Master (Cause Conflict)**

**Command:**  
git merge GitWork

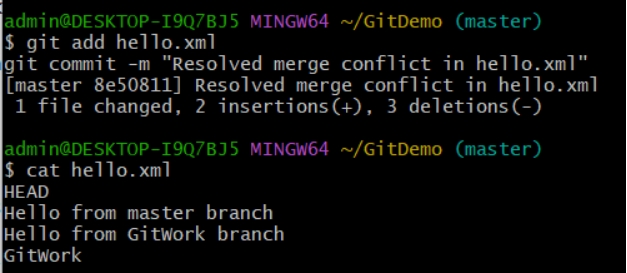
**Output:**  


**Step 7 – Resolve Conflict**

**Action:**  
Open hello.xml in an editor, manually keep the correct content, remove conflict markers (<<<<<<<, =======, >>>>>>>).

  
**command:**  
git add hello.xml  
git commit -m "Resolved merge conflict in hello.xml"

cat hello.xml



**Step 8 – Add Backup File to .gitignore (If Any)**

**Command:**  
echo "\*~" >> .gitignore  
git add .gitignore  
git commit -m "Added backup file pattern to .gitignore"

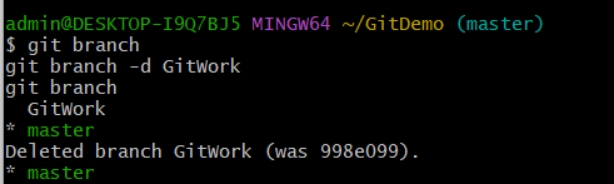
**Step 9 – View Final Status**

**Command:**  
git status

**Output:**  
Working tree clean.

**Step 10 – Delete Merged Branch**

**Command:**  
git branch  
git branch -d GitWork  
git branch

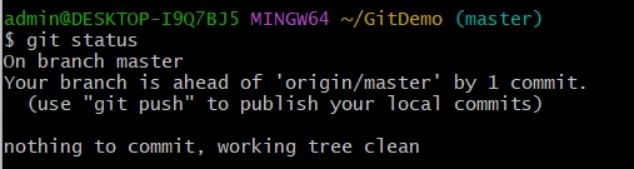
**Output:**  


**5. GIT-HOL:**

**Solution:**

**Step 1 – Verify Master is in Clean State**

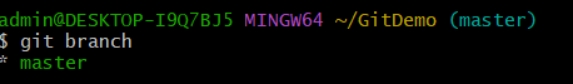
**Command:**  
git status

**Output:**  


**Step 2 – List All Available Branches**

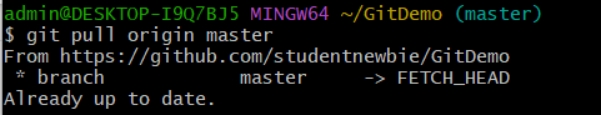
**Command:**  
git branch

**Output :**



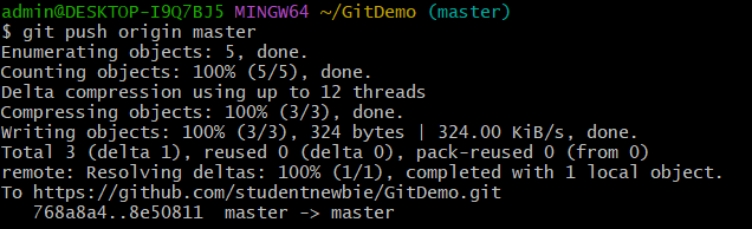
**Step 3 – Pull Latest from Remote**

**Command:**  
git pull origin master

**Output:**  


**Step 4 – Push Pending Changes**

**Command:**  
git push origin master

**Output:**  


**Step 5 – Verify on GitHub**

