**WEEK 8**

**Superset ID: 6396692**

**1)**

**Commands**

git –version

****

git config --global user.name "Harshini"

git config --global user.email [harsh123@gmail.com](mailto:harsh123@gmail.com)

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

git config --global –edit

mkdir GitDemo

cd GitDemo

git init

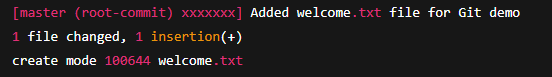
Output:

****

echo "Welcome to Git demo lab!" > welcome.txt

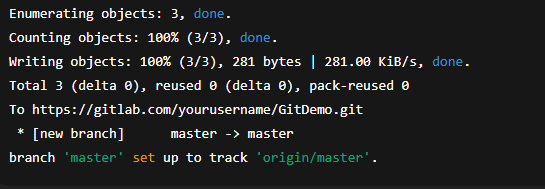
git add welcome.txt

git commit



git remote add origin <https://gitlab.com/harshini2127/GitDemo.git>

git push -u origin master



**2)**

**Commands**

**Setup:**

mkdir GitIgnoreDemo

cd GitIgnoreDemo

git init

**Step 2: Create Files and Folders to Ignore**

echo "Temporary log info" > error.log

mkdir log

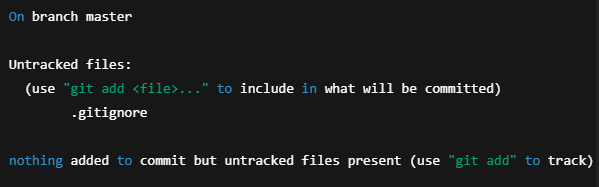
echo "Log folder content" > log/info.txt

**Step 3: Create .gitignore File**

notepad++ .gitignore

**Step 4: Check Git Status**

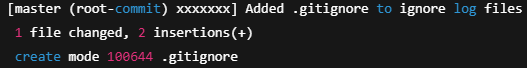
git status



**Step 5: Add and Commit .gitignore File**

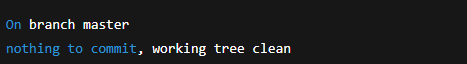
git add .gitignore

git commit -m "Added .gitignore to ignore log files and log directory"



**Step 6: Recheck Git Status**

git status



**3)**

**Step 1: Create a Git working directory**

mkdir GitIgnoreLab

cd GitIgnoreLab

git init

****

**Step 2: Create .log files and a log/ folder**

echo "This is a test log file." > test.log

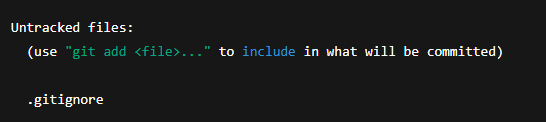
mkdir log

echo "Inside log folder" > log/log1.txt

**Step 3: Create .gitignore file**

touch .gitignore

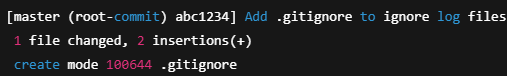
**Step 4: Check status**

****

**Step 5: Add and commit**

git add .gitignore

git commit -m "Add .gitignore to ignore log files and folders"



**Step 6: Verify ignored files**

git status





4)

**1. Verify if master is in clean state**

git status

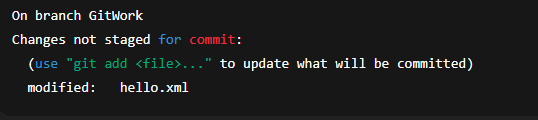


**2. Create a branch “GitWork”. Add a file “hello.xml”**

git checkout -b GitWork

echo "<message>Hello from GitWork</message>" > hello.xml

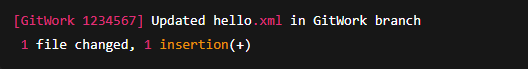
**3. Update the content of “hello.xml” and observe the status**

****

**4. Commit the changes to reflect in the branch**

git add hello.xml

git commit -m "Updated hello.xml in GitWork branch"

****

**5. Switch to master**

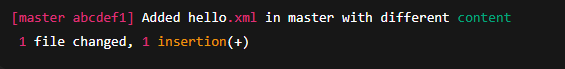
git checkout master

**6. Add a file “hello.xml” to the master and add different content**

echo "<message>Hello from master</message>" > hello.xml

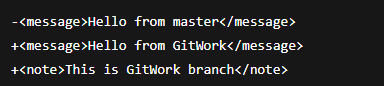
git add hello.xml

git commit -m "Added hello.xml in master with different content"



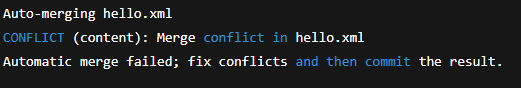
**7. Check the differences with Git diff tool**

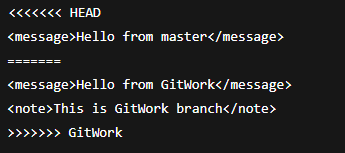
git diff GitWork hello.xml



**8. Merge the branch to master**

git merge GitWork





**9. Commit the changes to the master, once done with conflict**

git add hello.xml

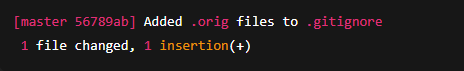
git commit -m "Resolved conflict between master and GitWork"



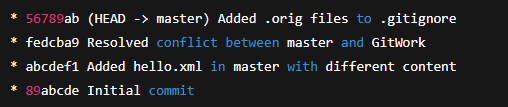
**10.Commit the changes to the .gitignore**

git add .gitignore

git commit -m "Added .orig files to .gitignore"



git log --oneline --graph –decorate

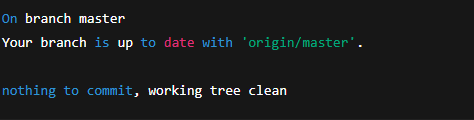


**5)**

**Commands**

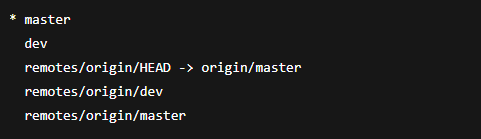
1. **Verify if master is in clean state**

git status



**2. List out all the available branches**

git branch -a



**3. Pull the remote Git repository to the master**

git pull origin master



**4.Push the changes (from “Git-T03-HOL\_002”) to the remote repository**

git add .

git commit -m "Push changes from Git-T03-HOL\_002"

git push origin master

