**WEEK-8 :- GIT**

**HANDS-ON – 1 :- GitDemo**

Check Git Version

git –version

OUTPUT:

git version 2.45.1.windows.1

Configure Git

git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

git config –list

OUTPUT:

user.name=Your Name

user.email=your.email@example.com

Initialize Repository

mkdir GitDemo

cd GitDemo

git init

OUTPUT:

Initialized empty Git repository in C:/path/to/GitDemo/.git/

Create a File

echo "Welcome to Git" > welcome.txt

cat welcome.txt

OUTPUT:

Welcome to Git

Check Status

git status

OUTPUT:

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

welcome.txt

nothing added to commit but untracked files present (use "git add" to track)

Add File to Staging

git add welcome.txt

git status

OUTPUT:

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: welcome.txt

Push to Remote

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

git push -u origin master

OUTPUT:

Enumerating objects: 3, done.

Counting objects: 100% (3/3), done.

Writing objects: 100% (3/3), 256 bytes | 256.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To https://github.com/yourusername/GitDemo.git

\* [new branch] master -> master

branch 'master' set up to track 'origin/master'.

**HANDS-ON – 2 :- Git Ignore Demo**

Create Project Folder and Initialize Git

mkdir GitIgnoreDemo

cd GitIgnoreDemo

git init

OUTPUT:

Initialized empty Git repository in C:/path/to/GitIgnoreDemo/.git/

Create Files and Folders to Ignore

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

mkdir logs

echo "Log file inside logs folder" > logs/test.log

Create .gitignore File

echo "\*.log" > .gitignore

echo "logs/" >> .gitignore

cat .gitignore

OUTPUT:

\*.log

logs/

Check Git Status

git status

OUTPUT:

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

.gitignore

nothing added to commit but untracked files present (use "git add" to track)

Add and Commit .gitignore

git add .gitignore

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

OUTPUT:

[master (root-commit) 1a2b3c4] Added .gitignore to ignore .log files and logs folder

1 file changed, 2 insertions(+)

create mode 100644 .gitignore

**HANDS-ON – 3 :- Branching and Merging in Git**

Create and View Branches

git branch GitNewBranch

git branch

OUTPUT:

\* master

GitNewBranch

Switch to New Branch

git checkout GitNewBranch

OUTPUT:

Switched to branch 'GitNewBranch'

Add a File in New Branch

echo "This is a new branch file" > branchfile.txt

git add branchfile.txt

git commit -m "Added branchfile.txt in GitNewBranch"

OUTPUT:

[GitNewBranch 1a2b3c4] Added branchfile.txt in GitNewBranch

1 file changed, 1 insertion(+)

create mode 100644 branchfile.txt

Check Status in Branch

git status

OUTPUT:

On branch GitNewBranch

nothing to commit, working tree clean

Switch to Master and Compare

git checkout master

git diff master GitNewBranch

OUTPUT:

diff --git a/branchfile.txt b/branchfile.txt

new file mode 100644

index 0000000..1a2b3c4

--- /dev/null

+++ b/branchfile.txt

@@

+This is a new branch file

Merge Branch into Master

git merge GitNewBranch

OUTPUT:

Updating 5d6e7f8..1a2b3c4

Fast-forward

branchfile.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 branchfile.txt

Delete Merged Branch

git branch -d GitNewBranch

OUTPUT:

Deleted branch GitNewBranch (was 1a2b3c4).

**HANDS-ON – 4 :- Resolving Merge Conflicts in Git**

Create Branch and Add File

git checkout -b GitWork

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

git add hello.xml

git commit -m "Added hello.xml in GitWork"

OUTPUT:

[GitWork 1a2b3c4] Added hello.xml in GitWork

1 file changed, 1 insertion(+)

create mode 100644 hello.xml

Switch to Master and Add Same File

git checkout master

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

git add hello.xml

git commit -m "Added hello.xml in master"

OUTPUT:

[master 5d6e7f8] Added hello.xml in master

1 file changed, 1 insertion(+)

create mode 100644 hello.xml

Merge Branch into Master (Conflict)

git merge GitWork

OUTPUT:

Auto-merging hello.xml

CONFLICT (add/add): Merge conflict in hello.xml

Automatic merge failed; fix conflicts and then commit the result.

Resolve Conflict

git add hello.xml

git commit -m "Resolved merge conflict in hello.xml"

OUTPUT:

[master abc1234] Resolved merge conflict in hello.xml

Delete Branch After Merge

git branch -d GitWork

OUTPUT:

Deleted branch GitWork (was 1a2b3c4).

**HANDS-ON – 5 :- Cleaning Up and Pushing to Remote Git**

Check Status

git status

OUTPUT:

On branch master

nothing to commit, working tree clean

List Branches

git branch -a

OUTPUT:

master

remotes/origin/master

Pull from Remote

git pull origin master

OUTPUT:

Already up to date.

Push Pending Changes to Remote

git push origin master

OUTPUT:

Enumerating objects: 3, done.

Counting objects: 100% (3/3), done.

Writing objects: 100% (3/3), 256 bytes | 256.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To https://github.com/yourusername/repo.git

abc1234..def5678 master -> master