**GIT**

**Suman Sourabh**

**6363866**

**HANDS-ON: 1 Git-HOL**

KIIT@DESKTOP-JE09TUR MINGW64 ~

$ git --version

git version 2.50.1.windows.1

KIIT@DESKTOP-JE09TUR MINGW64 ~

$ git config --global --list

core.editor="C:\Users\KIIT\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait

user.name=Suman Sourabh

user.email=sumansourabh852.90@gmail.com

KIIT@DESKTOP-JE09TUR MINGW64 ~

$ notepad++

KIIT@DESKTOP-JE09TUR MINGW64 ~

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

KIIT@DESKTOP-JE09TUR MINGW64 ~

$ git config --global --list

core.editor=notepad++

user.name=Suman Sourabh

user.email= sumansourabh852.90@gmail.com

KIIT@DESKTOP-JE09TUR MINGW64 ~

$ mkdir GitDemo

KIIT@DESKTOP-JE09TUR MINGW64 ~

$ cd GitDemo

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo

$ git init

Initialized empty Git repository in C:/Users/KIIT/GitDemo/.git/

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ ls -a

**.**/ **..**/ **.git**/

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ echo "Welcome to Git Demo" > welcome.txt

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ ls

welcome.txt

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ cat welcome.txt

Welcome to Git Demo

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git status

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)

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git add welcome.txt

warning: in the working copy of 'welcome.txt', LF will be replaced by CRLF the next time Git touches it

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git commit

[master (root-commit) dafd8df] Added welcome.txt file This file is the first commit in GitDemo

1 file changed, 1 insertion(+)

create mode 100644 welcome.txt

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git status

On branch master

nothing to commit, working tree clean

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git remote add origin https://gitlab.com/sumansourabh-group/GitDemo.git

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git push -u origin master

info: please complete authentication in your browser...

Enumerating objects: 3, done.

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

Writing objects: 100% (3/3), 278 bytes | 278.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To https://gitlab.com/sumansourabh-group/GitDemo.git

\* [new branch] master -> master

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

**HANDS-ON: 2 Git-HOL**

KIIT@ DESKTOP-JE09TUR MINGW64 ~

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

KIIT@DESKTOP-JE09TUR MINGW64 ~

$ git config --global core.editor

'C:/Program Files/Notepad++/notepad++.exe' -multiInst -nosession

KIIT@DESKTOP-JE09TUR MINGW64 ~

$ cd /c/Users/KIIT/GitDemo

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ pwd

/c/Users/KIIT/GitDemo

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ echo "this is a log file for testing" > sample.log

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ mkdir -p log

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ echo "debug info" > log/debug.log

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ notepad++ .gitignore

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git add .gitignore

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

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

[master 9120e71] Add .gitignore to ignore .log files and log directory

1 file changed, 2 insertions(+)

create mode 100644 .gitignore

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git status --short

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git ls-files --others --ignored --exclude-standard

log/debug.log

sample.log

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git check-ignore -v sample.log

.gitignore:1:\*.log sample.log

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git push

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 8 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 320 bytes | 320.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To <https://gitlab.com/sumansourabh-group/GitDemo.git>

dafd8df..9120e71 master -> master

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git status

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

**HANDS-ON: 3 Git-HOL**

* Explain branching and merging

Branching

Branching allows you to create a new line of development within your Git repository. It allows you to:

Add a new feature or a bug fixwithout changing the base codebase(usually the'main ' or'master ' branch).

Each branch is akin to referring to a snapshot of your modifications.

Example:

You start at'main ' and create a new branch:

'bash '

git checkout -b feature/login-page

Now all your commits will be kept on'feature/login-page '.

Merging

Integration merges the modifications of one branch to another.

Typically, once you have completed building your feature, youmerge it backinto the master branch.

Git tries toauto-resolvethe changes; otherwise, you are presented with amerge conflictto manually resolve.

Example:

bash

/bin/bash

git checkout main

git merge feature/login-page

* Explain about creating a branch request in GitLab

In GitLab, "branch request" is not a formal term, but it is most commonly used to describe:

Checking out a new branchfor your feature/bugfix from the GitLab UI or terminal.

Via GitLab UI:

1. Go to your repository on GitLab.

2. Click onRepository > Branches .

3. PressNew branch .

4. Enter thebranch nameand name it following a branch like'main '.

5. ClickCreate branch .

Via Git (terminal) :

bash

git checkout -b feature/branch-name

git push origin feature/branch-name

* Explain about creating a merge request in GitLab

A Merge Request or an MR in GitLab is utilized to:

Propose to apply changes from one branch (e.g.,'feature/login-page ') to another (e.g.,'main ').

It's used forcode review ,discussions , andapprovalbefore merging.

Step-by-step process to form a Merge Request in GitLab:

1. Stage your version in your feature branch:

'bash '

git push origin feature/branch-name

2. Go to your project on GitLab.

3. GitLab will prompt regularly:"Create merge request"for fresh branches.

4. Click it or go to.

Merge Requests > New Merge Request5. Choose: Source branch:'feature/branch-name ' Target branch:'main ' 6. Add atitle, description , includereviewers , and optionally addlabels or milestones . 7. ClickSubmit merge request .Once you're approved, you or any user canmergeit into the target branch from the GitLab UI

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git branch GitNewBranch

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git branch -a

GitNewBranch

\* master

remotes/origin/master

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git checkout GitNewBranch

Switched to branch 'GitNewBranch'

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (GitNewBranch)

$ echo "This is a file in GitNewBranch" > newfile.txt

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (GitNewBranch)

$ git add newfile.txt

warning: in the working copy of 'newfile.txt', LF will be replaced by CRLF the next time Git touches it

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (GitNewBranch)

$ git commit -m "Added newfile.txt in GitNewBranch"

[GitNewBranch c37d448] Added newfile.txt in GitNewBranch

1 file changed, 1 insertion(+)

create mode 100644 newfile.txt

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (GitNewBranch)

$ git status

On branch GitNewBranch

nothing to commit, working tree clean

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (GitNewBranch)

$ git checkout master

Switched to branch 'master'

Your branch is up to date with 'origin/master'.

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git diff GitNewBranch

**diff --git a/newfile.txt b/newfile.txt**

**deleted file mode 100644**

**index 70ab14d..0000000**

**--- a/newfile.txt**

**+++ /dev/null**

@@ -1 +0,0 @@

-This is a file in GitNewBranch

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git mergetool

This message is displayed because 'merge.tool' is not configured.

See 'git mergetool --tool-help' or 'git help config' for more details.

'git mergetool' will now attempt to use one of the following tools:

opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecmerge p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff

No files need merging

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git merge GitNewBranch

Updating 9120e71..c37d448

Fast-forward

newfile.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 newfile.txt

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git log --oneline --graph --decorate

\* c37d448 (**HEAD** -> **master**, **GitNewBranch**) Added newfile.txt in GitNewBranch

\* 9120e71 (**origin/master**) Add .gitignore to ignore .log files and log directory

\* dafd8df Added welcome.txt file This file is the first commit in GitDemo

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git branch -d GitNewBranch

Deleted branch GitNewBranch (was c37d448).

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ git status

On branch master

Your branch is ahead of 'origin/master' by 1 commit.

(use "git push" to publish your local commits)

nothing to commit, working tree clean

**HANDS-ON: 4 Git-HOL**

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ mkdir git-conflict-lab

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ cd git-conflict-lab

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (master)

$ git init

Initialized empty Git repository in C:/Users/KIIT/GitDemo/git-conflict-lab/.git/

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (master)

$ git status

On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (master)

$ git checkout -b GitWork

Switched to a new branch 'GitWork'

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (GitWork)

$ git status

On branch GitWork

No commits yet

Untracked files:

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

hello.xml

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

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (GitWork)

$ git add hello.xml

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (GitWork)

$ git commit -m "Add hello.xml in GitWork branch"

[GitWork (root-commit) 537b5c2] Add hello.xml in GitWork branch

1 file changed, 2 insertions(+)

create mode 100644 hello.xml

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ mkdir git-conflict-lab

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo (master)

$ cd git-conflict-lab

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (master)

$ git init

Initialized empty Git repository in C:/Users/KIIT/GitDemo/git-conflict-lab/.git/

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (master)

$ git add hello.xml

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (master)

$ git commit -m "Add hello.xml in master branch"

[master (root-commit) eb6315d] Add hello.xml in master branch

1 file changed, 2 insertions(+)

create mode 100644 hello.xml

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (master)

$ git checkout -b GitWork

Switched to a new branch 'GitWork'

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (GitWork)

$ git add hello.xml

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (GitWork)

$ git commit -m "Modify hello.xml in GitWork branch"

[GitWork 2952294] Modify hello.xml in GitWork branch

1 file changed, 1 insertion(+), 1 deletion(-)

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (GitWork)

$ git checkout master

Switched to branch 'master'

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (master)

$ git merge GitWork

Updating eb6315d..2952294

Fast-forward

hello.xml | 2 +-

1 file changed, 1 insertion(+), 1 deletion(-)

**HANDS-ON: 5 Git-HOL**

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (master)

$ git status

On branch master

nothing to commit, working tree clean

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (master)

$ git branch

GitWork

\* master

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (master)

$ git pull origin master --allow-unrelated-histories

From https://gitlab.com/sumansourabh-group/GitDemo

\* branch master -> FETCH\_HEAD

Merge made by the 'ort' strategy.

.gitignore | 2 ++

welcome.txt | 1 +

2 files changed, 3 insertions(+)

create mode 100644 .gitignore

create mode 100644 welcome.txt

KIIT@DESKTOP-JE09TUR MINGW64 ~/GitDemo/git-conflict-lab (master)

$ git push origin master

Enumerating objects: 9, done.

Counting objects: 100% (9/9), done.

Delta compression using up to 8 threads

Compressing objects: 100% (6/6), done.

Writing objects: 100% (8/8), 992 bytes | 992.00 KiB/s, done.

Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To <https://gitlab.com/sumansourabh-group/GitDemo.git>

9120e71..f96424d master -> master

