Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Information Technology

COURSE CODE: DJ19ITL602 **DATE:**

COURSE NAME: Software Engineering Laboratory **CLASS: T.Y.BTech**

EXPERIMENT NO. 7

CO/LO Analyze real world problem using software engineering principles.

AIM / OBJECTIVE: To Perform Version Control on any project using any Version control tool (GIT).

THEORY:

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny foot print with lightning-fast performance.

Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development, but it can be used to keep track of changes in any set of files.

It outclasses SCM tools like Subversion, CVS, Perforce and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Some of the basic operations in Git are:

1.Initialize

2.Add

3.Commit

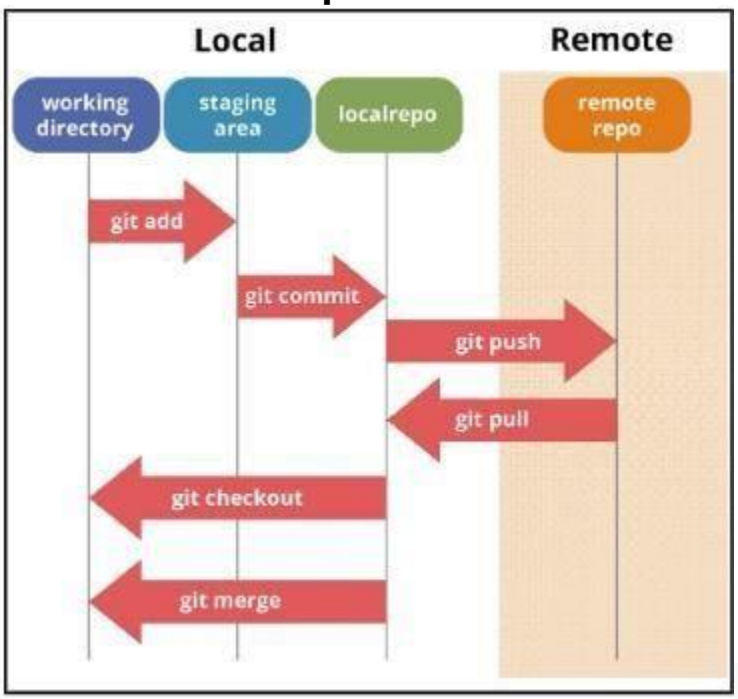
4.Pull

5.Push

- Some advanced Git operations are:
    1. Branching
    2. Merging
- The following diagram depict the all supported operations in GIT

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Information Technology



COMMANDS:

1. $git –version

```
ganes@DESKTOP-4EO7LOM MINGW64 ~
$ git --version
git version 2.33.1.windows.1
```

2. $git config

   The git config command is a convenience function that is used to set Git configuration values on a global or local project level.

   $git config –global

   Global level configuration is user-specific, meaning it is applied to an operating system user. Global configuration values are stored in a file that is located in a user's home directory.

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Information Technology

```
ganes@DESKTOP-4EO7LOM MINGW64 ~
$ git config --global
usage: git config [<options>]

Config file location
    --global              use global config file
    --system              use system config file
    --local               use repository config file
    --worktree            use per-worktree config file
    -f, --file <file>     use given config file
    --blob <blob-id>      read config from given blob object

Action
    --get                 get value: name [value-pattern]
    --get-all             get all values: key [value-pattern]
    --get-regexp          get values for regexp: name-regex [value-pattern]
    --get-urlmatch        get value specific for the URL: section[.var] URL
    --replace-all         replace all matching variables: name value [value-pattern]
    --add                 add a new variable: name value
    --unset               remove a variable: name [value-pattern]
    --unset-all           remove all matches: name [value-pattern]
    --rename-section      rename section: old-name new-name
    --remove-section      remove a section: name
    -l, --list            list all
    --fixed-value         use string equality when comparing values to 'value-pattern'
    -e, --edit            open an editor
    --get-color           find the color configured: slot [default]
    --get-colorbool       find the color setting: slot [stdout-is-tty]

Type
    -t, --type <>         value is given this type
    --bool                value is "true" or "false"
    --int                 value is decimal number
    --bool-or-int         value is --bool or --int
    --bool-or-str         value is --bool or string
    --path                value is a path (file or directory name)
    --expiry-date         value is an expiry date

Other
    -z, --null            terminate values with NUL byte
    --name-only           show variable names only
    --includes            respect include directives on lookup
    --show-origin         show origin of config (file, standard input, blob, command line)
    --show-scope          show scope of config (worktree, local, global, system, command)
    --default <value>     with --get, use default value when missing entry
```

1. git config --global --list

```
ganes@DESKTOP-4EO7LOM MINGW64 ~
$ git config --global --list
user.email=jaishruti0002@gmail.com
user.name=jaishrutimahadevan
```

2. $git config --global user.name “<user_name>”

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Information Technology

```
ganes@DESKTOP-4EO7LOM MINGW64 ~
$ git config --global user.name "jaishrutimahadevan"

ganes@DESKTOP-4EO7LOM MINGW64 ~
$ git config --global user.email "jaishruti0002@gmail.com"
```

$git config –global user.email "<user_email>"

3. mkdir git-demo-project

```
ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop (master)
$ mkdir git-demo-project

ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop (master)
$ cd git-demo-project/
```

cd git-demo-project

4. git init git status

```
ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop/git-demo-project (master)
$ git init
Initialized empty Git repository in C:/Users/ganes/OneDrive/Desktop/git-demo-project/.git/

ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop/git-demo-project (master)
$ git status
On branch master

No commits yet

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

1. mkdir git-demo-project

```
ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop (master)
$ mkdir git-demo-project

ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop (master)
$ cd git-demo-project/
```

cd git-demo-project

1. git init git status

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Information Technology

```
ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop/git-demo-project (master)
$ git init
Initialized empty Git repository in C:/Users/ganes/OneDrive/Desktop/git-demo-project/.git/

ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop/git-demo-project (master)
$ git status
On branch master

No commits yet

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

7. git add new_git.txt

   git commit -m “First Commit”

```
ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop/git-demo-project (master)
$ nano new_git.txt

ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop/git-demo-project (master)
$ git add new_git.txt
warning: LF will be replaced by CRLF in new_git.txt.
The file will have its original line endings in your working directory

ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop/git-demo-project (master)
$ git commit -m "First Commit"
[master (root-commit) 012db69] First Commit
 1 file changed, 1 insertion(+)
 create mode 100644 new_git.txt
```

8.

```
ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop/git-demo-project (master)
$ git log
commit 012db69f98ff80eacce83b88b28d1df35ec774fe (HEAD -> master)
Author: jaishrutimahadevan <jaishruti0002@gmail.com>
Date:   Tue Jan 25 10:52:21 2022 +0530

    First Commit
```

   git log

9. git clone https://github.com/siddii/angular-timer.git

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Information Technology

```
ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop/git-demo-project (master)
$ git clone https://github.com/siddii/angular-timer.git
Cloning into 'angular-timer'...
remote: Enumerating objects: 9991, done.
remote: Total 9991 (delta 0), reused 0 (delta 0), pack-reused 9991
Receiving objects: 100% (9991/9991), 38.29 MiB | 4.20 MiB/s, done.
Resolving deltas: 100% (2509/2509), done.
```

10. git push origin master

```
ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop/git-repo-project (master)
$ git remote add origin https://github.com/jaishrutimahadevan/git_test.git

ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop/git-repo-project (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 238 bytes | 238.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/jaishrutimahadevan/git_test.git
 * [new branch]      master -> master
```

11.

```
ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop/git-repo-project (master)
$ git checkout -b feature
Switched to a new branch 'feature'
```

git checkout -b feature

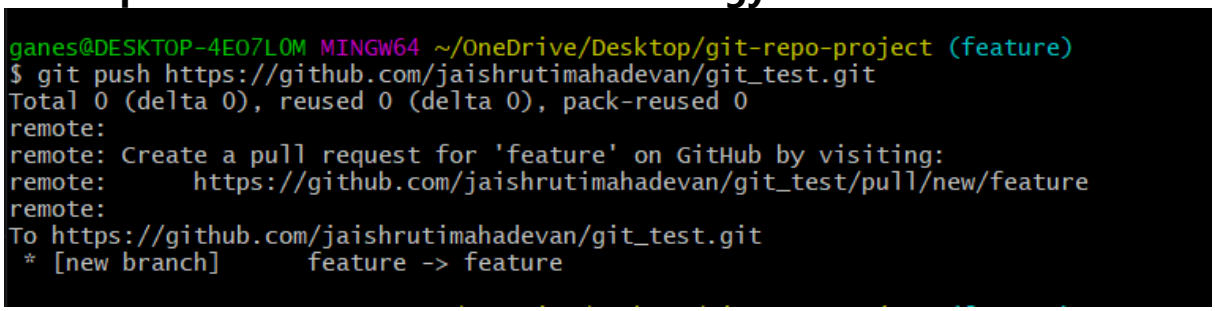
12.

```
ganes@DESKTOP-4EO7LOM MINGW64 ~/OneDrive/Desktop/git-repo-project (feature)
$ git commit -m "New Commit"
On branch feature
nothing to commit, working tree clean
```

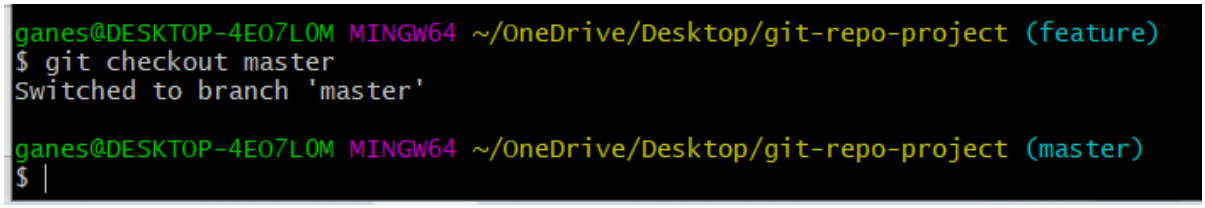
git commit -m “New Commit”

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Information Technology

13. 

git push https://github.com/jaishrutimahadevan/git_test.git

14. 

git checkout master

OUTPUT:

a. Screenshots of every step

QUESTION:

1. What are the advantages of using GIT?
2. What are the benefits of using Version Control System?

REFERENCE

www.geeksforgeeks.com

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Information Technology

Steps to Perform Version Control using Git:

1. Initialize a Git Repository

cd your-project-folder
git init

This creates a .git/ folder in your project, which tracks all version history.

2. Add Files to Git

git add .
or
git add index.html app.js

This stages all files or specific files to be tracked.

3. Commit Your Changes

git commit -m "Initial commit: added base project files"

Commits are snapshots of your project.

4. Connect to a Remote Repository (e.g., GitHub)

git remote add origin https://github.com/your-username/your-repo.git

Replaces the URL with your actual GitHub repo URL.

5. Push to GitHub

git push -u origin master
or (if main branch)
git push -u origin main

6. Track Changes Over Time

git add changed_file.py
git commit -m "Updated the visualization logic"
git push

7. Create and Switch to a New Branch

git checkout -b feature-new-graph

Later merge into main:
git checkout main
git merge feature-new-graph

8. View Git History

git log

Bonus: Useful Git Commands Summary

Task	Command
Clone a repo	git clone <repo_url>
View status	git status
View diff before commit	git diff
Discard changes	git checkout -- filename
Pull latest changes	git pull
Revert to a previous commit	git revert <commit_hash>