EXPERIMENT-3

Aim : To Perform various GIT operations on local and Remote repositories

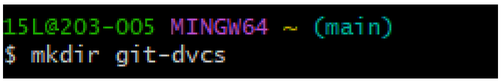
Theory:

The command mkdir git creates a new directory (folder) named "git" in the current working directory. This command is used to make a new directory in a Unix-like operating system.

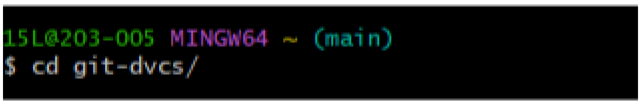
The command cd git is used to change the current working directory to the directory named "git." After executing this command, any subsequent commands or file operations will occur within the "git" directory. "cd" stands for "change directory."

The git config --global user.name and git config --global user.email commands are used to set your

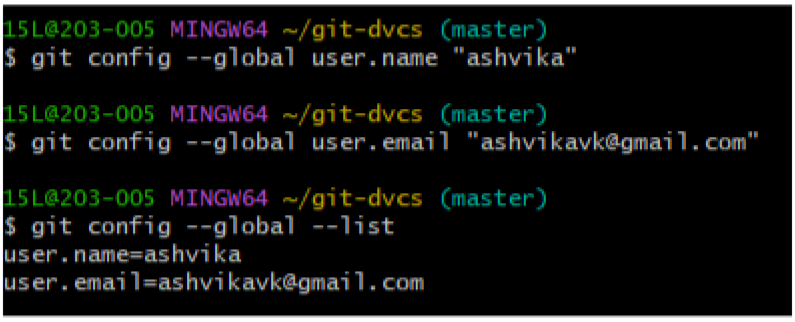
of the configuration settings in Git



```
15L@203-005 MINGW64 ~ (main)
$ mkdir git-dvcs
```



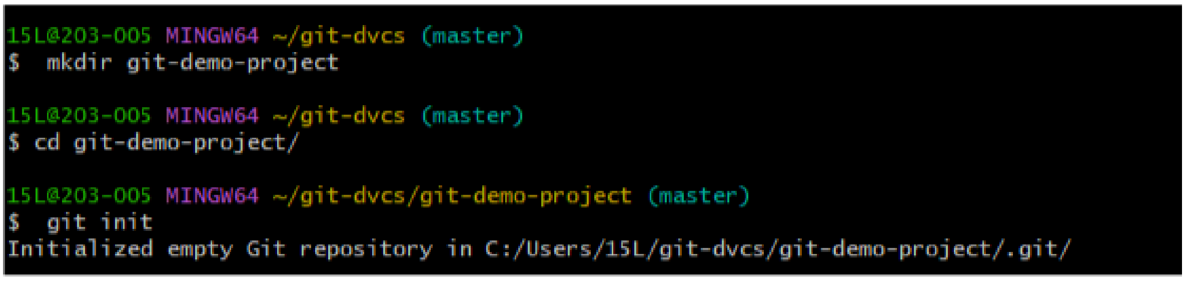
```
15L@203-005 MINGW64 ~ (main)
$ cd git-dvcs/
```



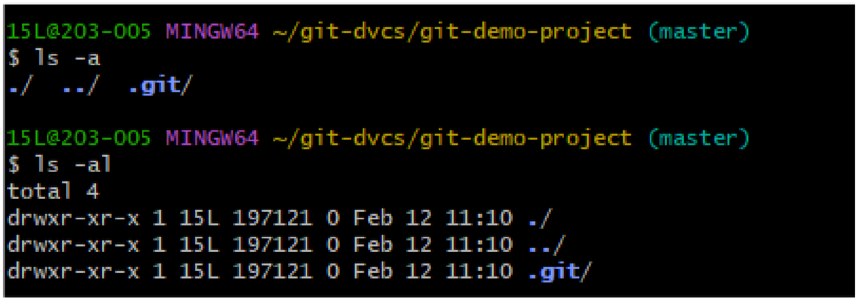
```
15L@203-005 MINGW64 ~/git-dvcs (master)
$ git config --global user.name "ashvika"

15L@203-005 MINGW64 ~/git-dvcs (master)
$ git config --global user.email "ashvikavk@gmail.com"

15L@203-005 MINGW64 ~/git-dvcs (master)
$ git config --global --list
user.name=ashvika
user.email=ashvikavk@gmail.com
```

If you want to check your configuration settings, you can use the git config --list command to list all the settings Git can find at that point

git commit -am "commit message"` stages and commits all changes in tracked files with a commit message in a single command.



```
15L@203-005 MINGW64 ~/git-dvcs (master)
$  mkdir git-demo-project

15L@203-005 MINGW64 ~/git-dvcs (master)
$ cd git-demo-project/

15L@203-005 MINGW64 ~/git-dvcs/git-demo-project (master)
$  git init
Initialized empty Git repository in C:/Users/15L/git-dvcs/git-demo-project/.git/
```



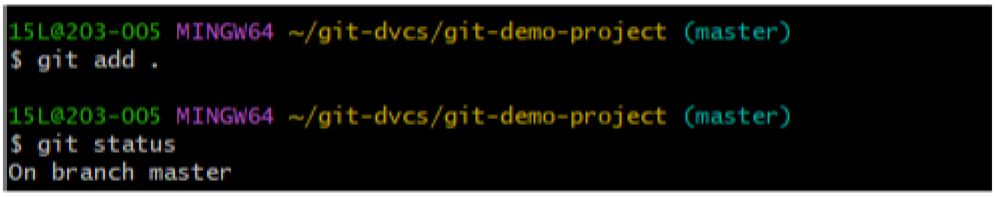
```
15L@203-005 MINGW64 ~/git-dvcs/git-demo-project (master)
$ ls -a
./  ../  .git/

15L@203-005 MINGW64 ~/git-dvcs/git-demo-project (master)
$ ls -al
total 4
drwxr-xr-x 1 15L 197121 0 Feb 12 11:10 ./
drwxr-xr-x 1 15L 197121 0 Feb 12 11:10 ../
drwxr-xr-x 1 15L 197121 0 Feb 12 11:10 .git/
```

The command nano index.html opens the Nano text editor for the file named "index.html." Nano is a simple command-line text editor that allows you to view and edit files directly in the terminal.

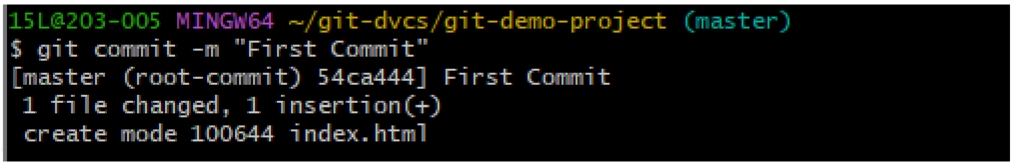
The command touch teststatus creates an empty file named "teststatus" in the current directory. The touch command is commonly used to update the timestamps of a file or create an empty file if it doesn't exist.

git checkout -- teststatus: Discards changes to the file "teststatus" in the working directory.This reverts the file to the state it has in the last commit.

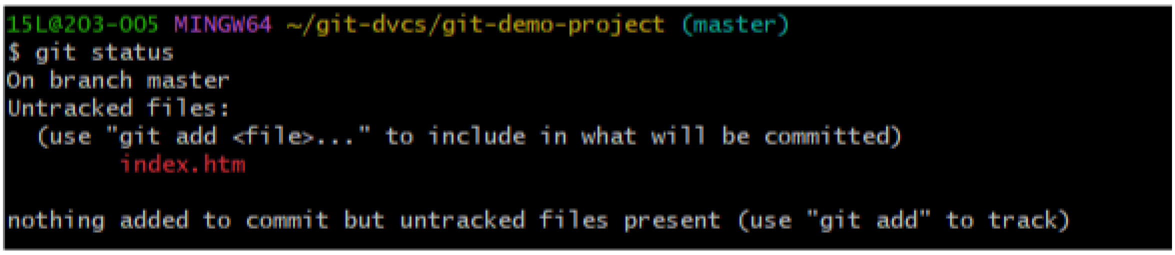
The git add command is used to stage changes in the working directory for the next commit in Git. It prepares modifications, additions, or deletions to be included in the upcoming commit.



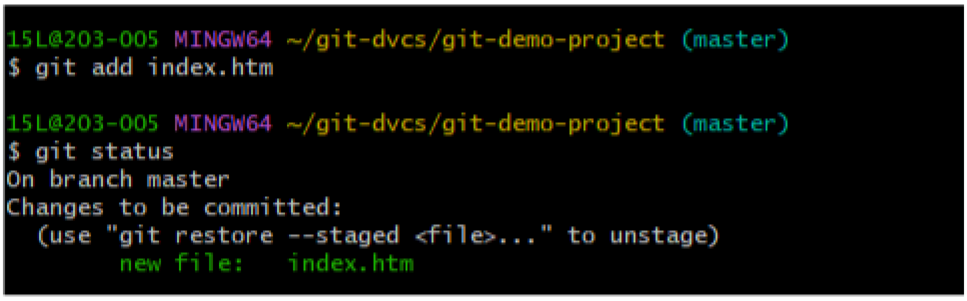
```
15L@203-005 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git add .

15L@203-005 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git status
On branch master
```



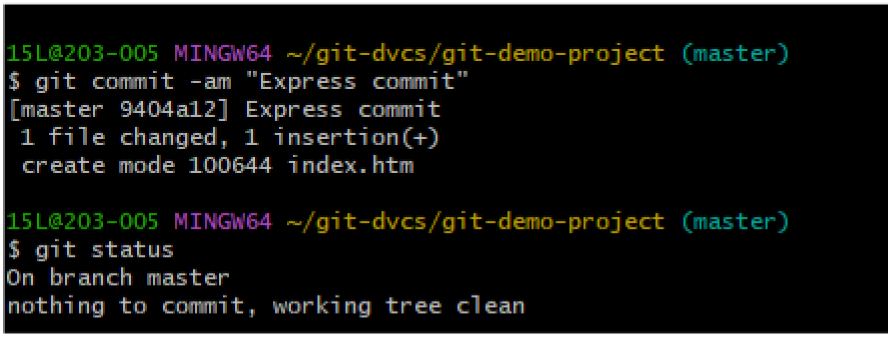
```
15L@203-005 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git commit -m "First Commit"
[master (root-commit) 54ca444] First Commit
 1 file changed, 1 insertion(+)
 create mode 100644 index.html
```



```
15L@203-005 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.htm

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



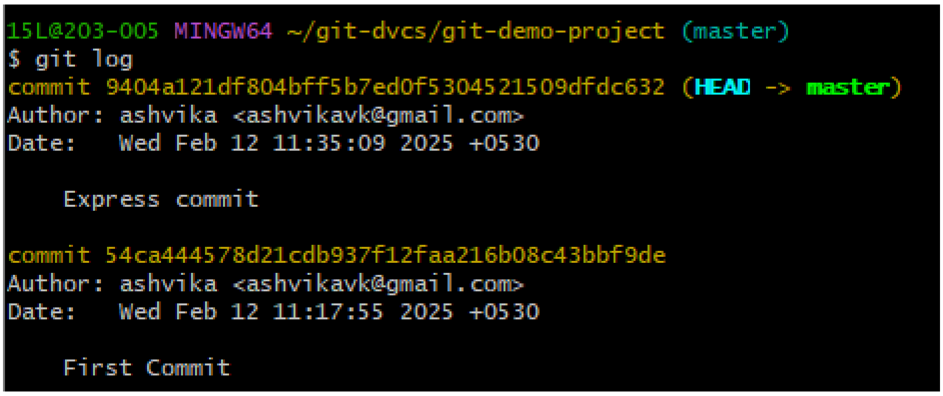
```
15L@203-005 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git add index.htm

15L@203-005 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   index.htm
```



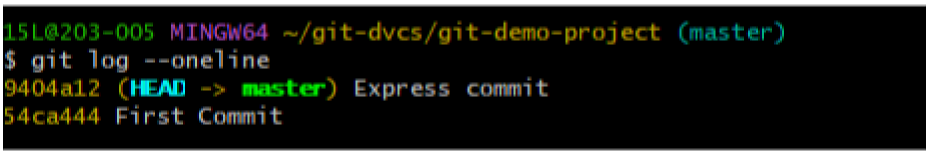
```
15L@203-005 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git commit -am "Express commit"
[master 9404a12] Express commit
 1 file changed, 1 insertion(+)
 create mode 100644 index.htm

15L@203-005 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git status
On branch master
nothing to commit, working tree clean
```

The git log command is used to display the commit history of a Git repository. It shows a chronological list of commits, including commit hashes, author information, timestamps, and commit messages.

The command git log --oneline displays a simplified and concise one-line representation of the commit history in a Git repository, showing only the commit SHA-1 hash and the commit message.



```
15L@203-005 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git log
commit 9404a121df804bff5b7ed0f5304521509dfdc632 (HEAD -> master)
Author: ashvika <ashvikavk@gmail.com>
Date:   Wed Feb 12 11:35:09 2025 +0530

    Express commit

commit 54ca444578d21cdb937f12faa216b08c43bbf9de
Author: ashvika <ashvikavk@gmail.com>
Date:   Wed Feb 12 11:17:55 2025 +0530

    First Commit
```



```
15L@203-005 MINGW64 ~/git-dvcs/git-demo-project (master)
$ git log --oneline
9404a12 (HEAD -> master) Express commit
54ca444 First Commit
```

The git clone command is used to create a copy of a Git repository. When you run this command, it duplicates the entire repository, including its files, commit history, and branches, and downloads it to your local machine. This is often the initial step when you want to work with a project hosted on a remote Git repository.

The git pull command is used to fetch and integrate changes from a remote repository into the current branch of your local repository. It combines two actions: it fetches the changes from the remote repository, and then it automatically merges those changes into your local branch. This is a convenient way to update your local repository with the latest changes from the remote repository.

The git push command is used to upload or push the local changes in your Git repository to a remote repository. It updates the remote repository with the latest changes made in your local branch, making them accessible to others who share the same remote repository.

The git fetch command is used to retrieve changes from a remote repository. It fetches

any new branches or changes made in the remote repository since your last interaction. However, it does not automatically merge these changes into your local branches. After using git fetch, you can inspect the changes and decide whether to integrate them using git merge or git rebase.

CONCLUSION : Successfully studied various Git operations on local and remote repositories