Experiment -1.1

Student Name: Sneha Mehrotra	**UID: 22BDO100048**
Branch: CSE(DEVOPS)	**Section/Group:22BCD-1/A**
Semester: 4TH	**Date of Performance: 17/01/2023**
Subject Name:GIT AND GITHUB	**Subject Code: 22CSH-293**

1. Aim/Overview of the practical:. Install Git and creating repository

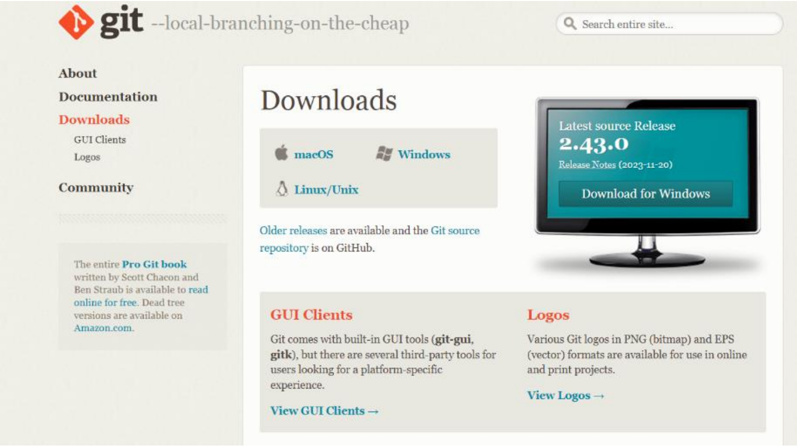
2. Software used: Git Bash and Github.

3. Hardware Used: Computer system.

4. Steps for experiment:

For Installation of Git :

1. Download Git for windows from chrome.



Latest source Release 2.43.0 Release Notes (2023-11-20)

Older releases are available and the Git source repository is on GitHub.

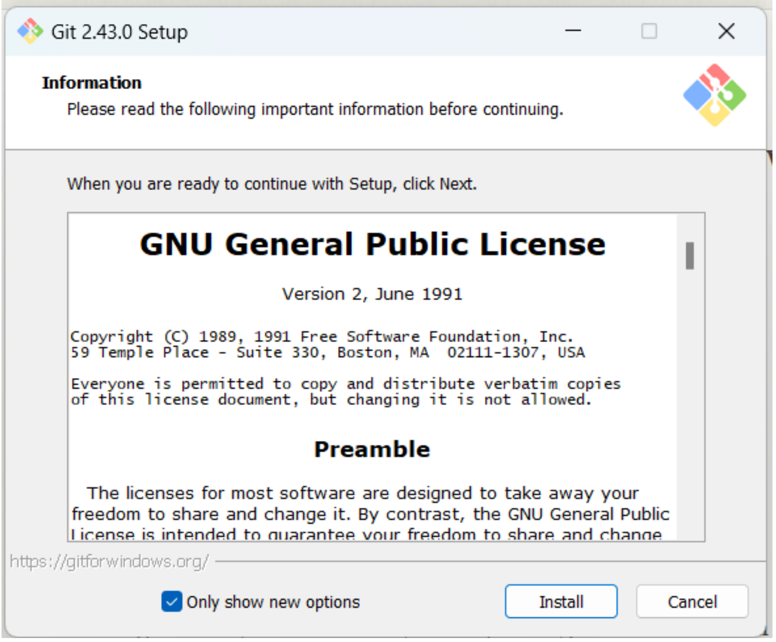
The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Git comes with built-in GUI tools (git-gui, gitk), but there are several third-party tools for users looking for a platform-specific experience.

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

2. Click on windows and go for 64 bit installer.

3. Click on install



Git 2.43.0 Setup

Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

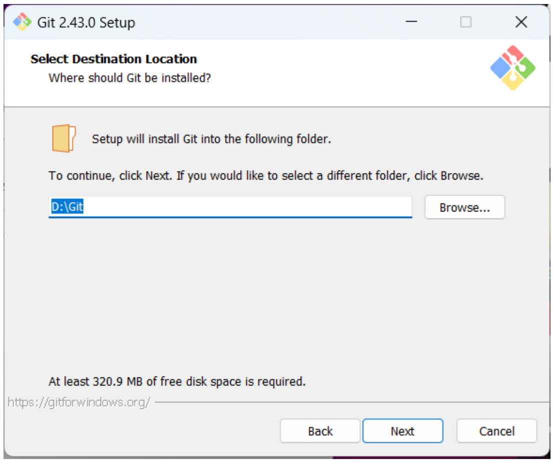
Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

https://gitforwindows.org/

☑ Only show new options

4. Select the location for installation for the git.



Git 2.43.0 Setup

Select Destination Location

Where should Git be installed?

Setup will install Git into the following folder.

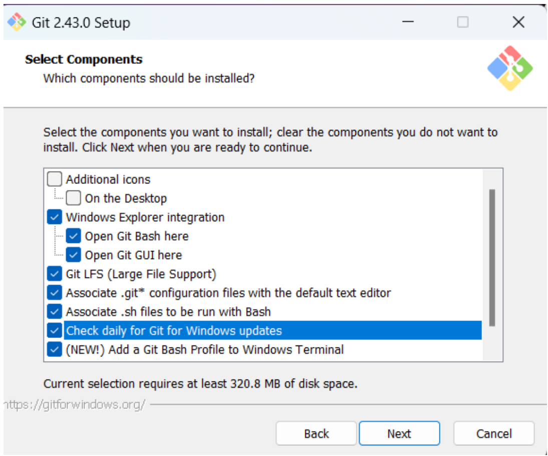
To continue, click Next. If you would like to select a different folder, click Browse.

D:\Git

At least 320.9 MB of free disk space is required.

https://gitforwindows.org/

6. Click on next for all default settings.



Select Components

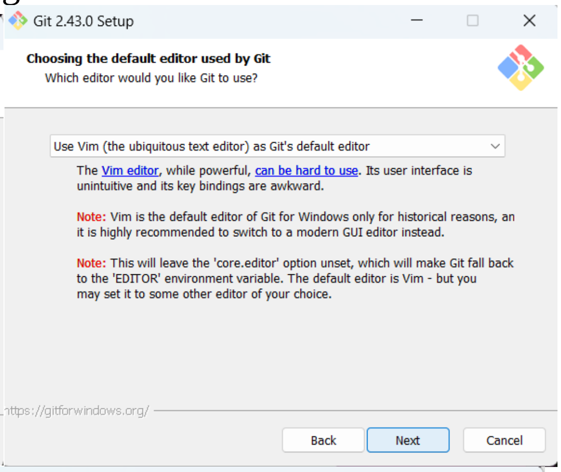
Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- □ Additional icons
  - □ On the Desktop
- ☑ Windows Explorer integration
  - ☑ Open Git Bash here
  - ☑ Open Git GUI here
- ☑ Git LFS (Large File Support)
- ☑ Associate .git* configuration files with the default text editor
- ☑ Associate .sh files to be run with Bash
- ☑ Check daily for Git for Windows updates
- ☑ (NEW!) Add a Git Bash Profile to Windows Terminal

Current selection requires at least 320.8 MB of disk space.

https://gitforwindows.org/



Choosing the default editor used by Git

Which editor would you like Git to use?

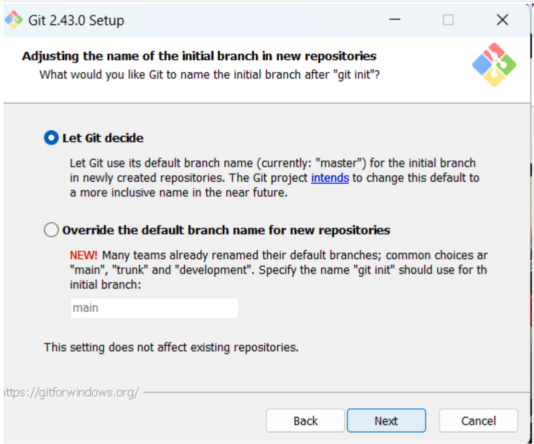
Use Vim (the ubiquitous text editor) as Git's default editor

The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.

Note: Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.

Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.

https://gitforwindows.org/



Adjusting the name of the initial branch in new repositories

What would you like Git to name the initial branch after "git init"?

● **Let Git decide**

Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project intends to change this default to a more inclusive name in the near future.

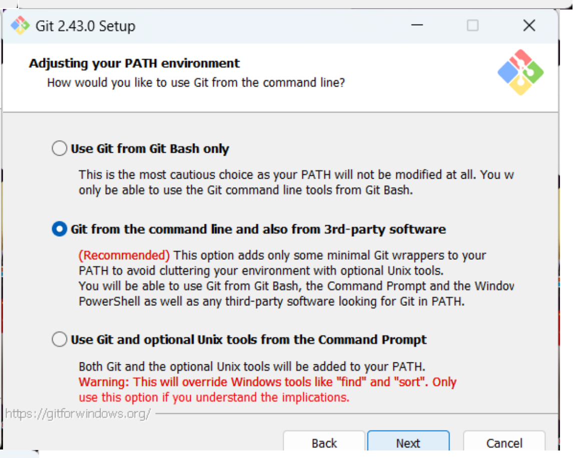
○ **Override the default branch name for new repositories**

NEW! Many teams already renamed their default branches; common choices ar "main", "trunk" and "development". Specify the name "git init" should use for th initial branch:

main

This setting does not affect existing repositories.

https://gitforwindows.org/



Adjusting your PATH environment

How would you like to use Git from the command line?

○ **Use Git from Git Bash only**

This is the most cautious choice as your PATH will not be modified at all. You w only be able to use the Git command line tools from Git Bash.

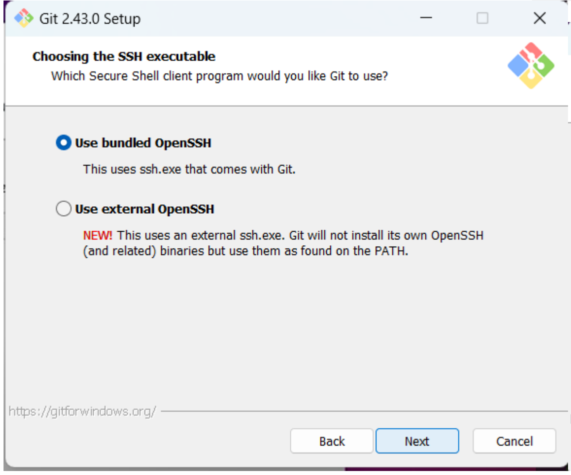
● **Git from the command line and also from 3rd-party software**

(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools.
You will be able to use Git from Git Bash, the Command Prompt and the Windov PowerShell as well as any third-party software looking for Git in PATH.

○ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

https://gitforwindows.org/



Choosing the SSH executable

Which Secure Shell client program would you like Git to use?

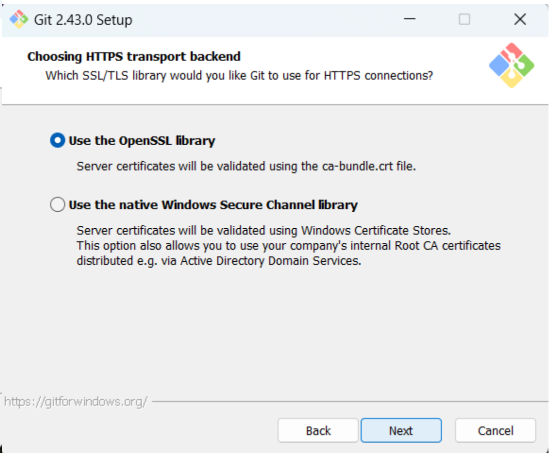
● **Use bundled OpenSSH**

This uses ssh.exe that comes with Git.

○ **Use external OpenSSH**

NEW! This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.

https://gitforwindows.org/



Choosing HTTPS transport backend

Which SSL/TLS library would you like Git to use for HTTPS connections?

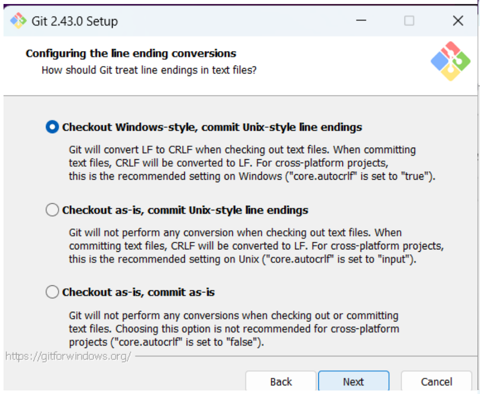
● **Use the OpenSSL library**

Server certificates will be validated using the ca-bundle.crt file.

○ **Use the native Windows Secure Channel library**

Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

https://gitforwindows.org/

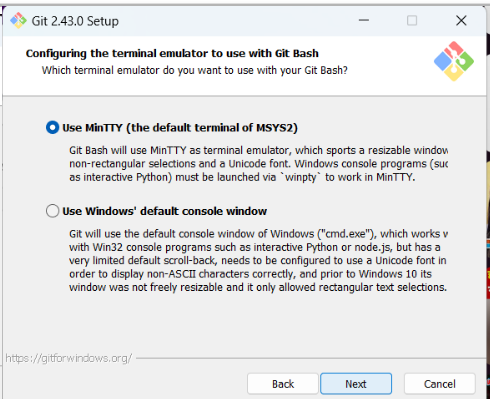


Configuring the line ending conversions
How should Git treat line endings in text files?

● **Checkout Windows-style, commit Unix-style line endings**
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

○ **Checkout as-is, commit Unix-style line endings**
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

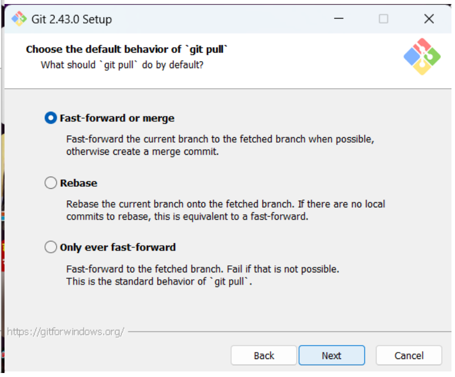
○ **Checkout as-is, commit as-is**
Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").



Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?

● **Use MinTTY (the default terminal of MSYS2)**
Git Bash will use MinTTY as terminal emulator, which sports a resizable window non-rectangular selections and a Unicode font. Windows console programs (suc with Win32 console programs such as interactive Python) must be launched via `winpty` to work in MinTTY.

○ **Use Windows' default console window**
Git will use the default console window of Windows ("cmd.exe"), which works v with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

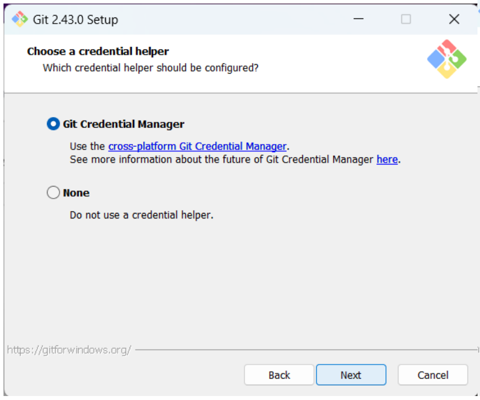


Choose the default behavior of `git pull`
What should `git pull` do by default?

● **Fast-forward or merge**
Fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.

○ **Rebase**
Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.

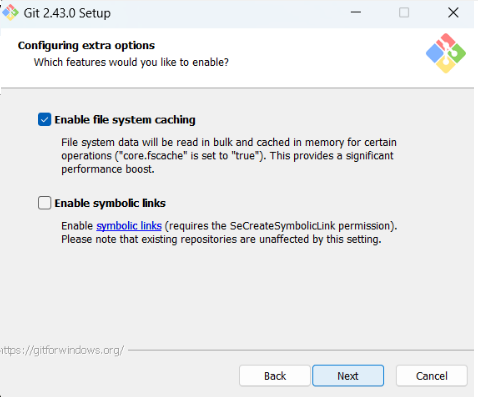
○ **Only ever fast-forward**
Fast-forward to the fetched branch. Fail if that is not possible. This is the standard behavior of `git pull`.



Choose a credential helper
Which credential helper should be configured?

● **Git Credential Manager**
Use the cross-platform Git Credential Manager.
See more information about the future of Git Credential Manager here.

○ **None**
Do not use a credential helper.

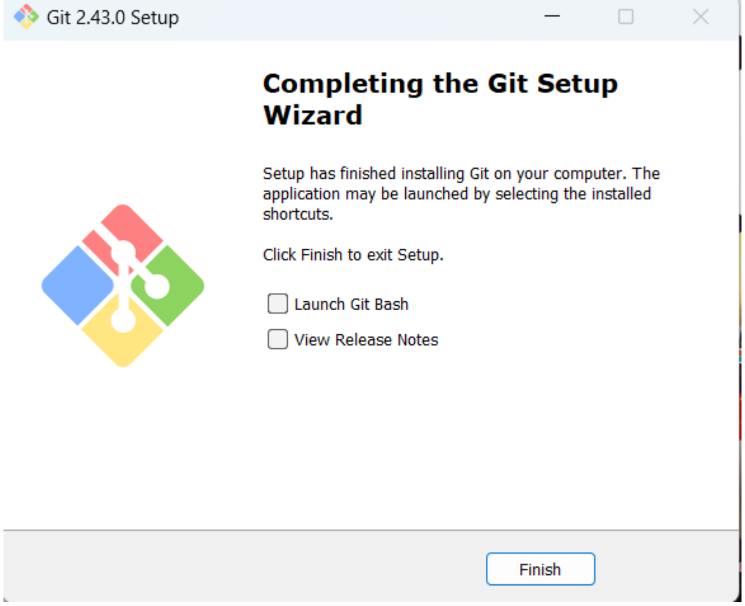


Configuring extra options
Which features would you like to enable?

☑ **Enable file system caching**
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

☐ **Enable symbolic links**
Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

7. Click on Finish after successful completing the installation.



Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☐ Launch Git Bash

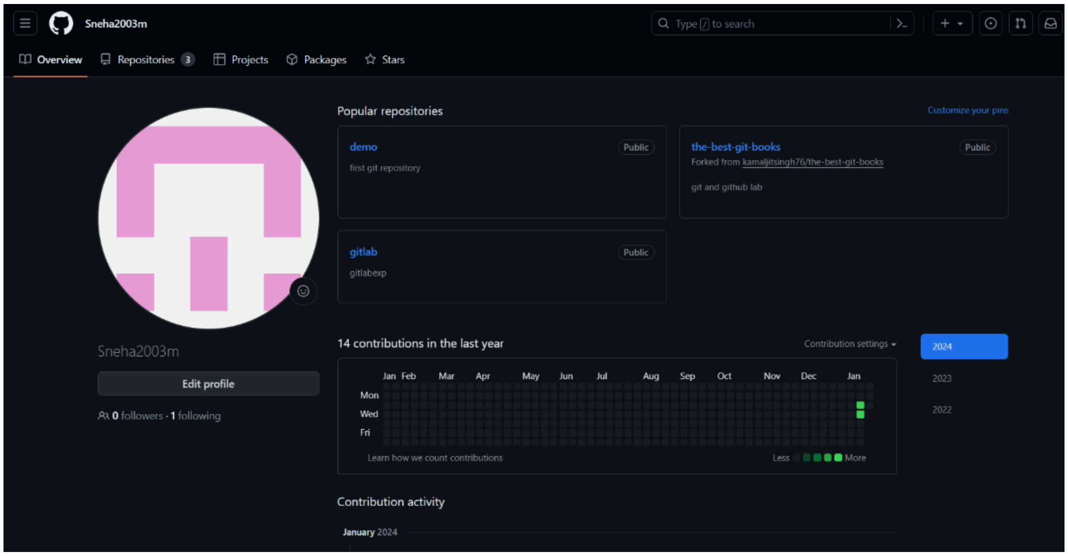
☐ View Release Notes

8. To verify git is installed in system or not follow these steps.
Go to CMD-> type *"git"*
It will give version and description regarding Git

Creation Of Account on Github

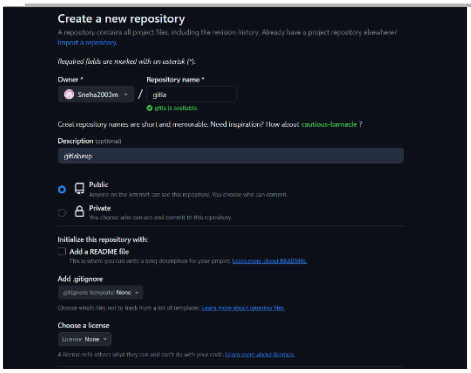
Follow these steps to create acoount on Github

1. Type Github on google.
2. Go for signup.
3. Login with email.
4. Give password.
5. Accept condition and fill necessary information like username DOB and more.
6. After completing formation of github account page will look like



Creating repository on Github

Goto github account -> Repositories -> New -> Name the Repository ->write Description(optional) -> Public(to access openly) ->Add a README file(Details of Project) -> add .gitignore - > License(none)-> click "Create Repository"



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner *: Sneha2003m / Repository name *: gitla

gitla is available.

Great repository names are short and memorable. Need inspiration? How about cautious-barnacle ?

Description (optional): gitlabexp

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:

□ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore
.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license
License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

Configuration of Git And Github:

To configure Git and Github we have to write some commands in Git Bash which we enlist our username and email in git.

Steps: For username

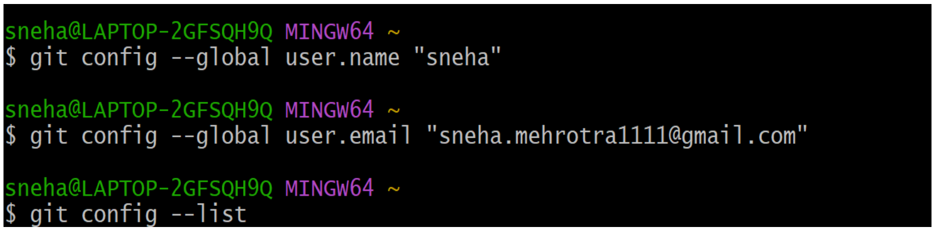
Type code:- *git config --global user.name* “Sneha ”

For email

Type code:- *git config --global user.email* “Sneha.mehrotra1111@gmail.com”

To verify

Type code:-*git config –list*



```
sneha@LAPTOP-2GFSQH9Q MINGW64 ~
$ git config --global user.name "sneha"

sneha@LAPTOP-2GFSQH9Q MINGW64 ~
$ git config --global user.email "sneha.mehrotra1111@gmail.com"

sneha@LAPTOP-2GFSQH9Q MINGW64 ~
$ git config --list
```

For cloning the repository:

Use Command

git clone https://github.com/Sneha2003m/Gitlab.git

Note: link is of the HTTPS

Press enter.

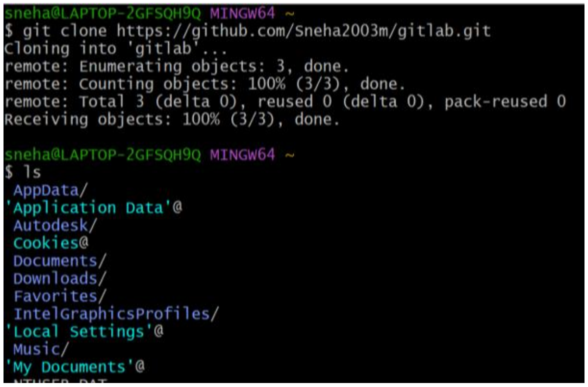
Some useful command

cd- Change the directory

touch- it is used to create file.

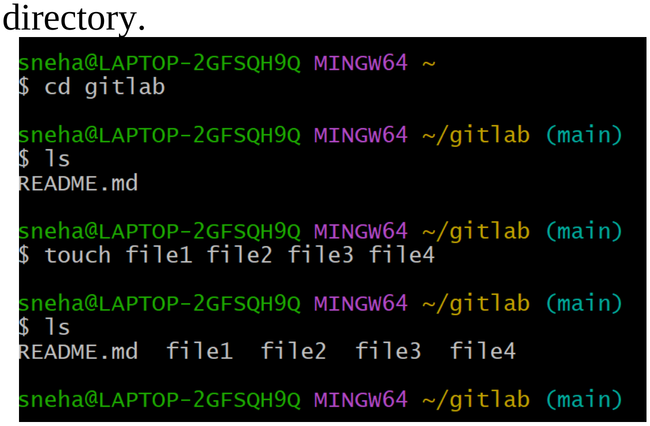
Ls -used to print no of files in repository.

pwd- used to print the current working directory.



```
sneha@LAPTOP-2GFSQH9Q MINGW64 ~
$ git clone https://github.com/Sneha2003m/gitlab.git
Cloning into 'gitlab'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

sneha@LAPTOP-2GFSQH9Q MINGW64 ~
$ ls
 AppData/
'Application Data'@
 Autodesk/
 Cookies@
 Documents/
 Downloads/
 Favorites/
 IntelGraphicsProfiles/
'Local Settings'@
 Music/
'My Documents'@
```



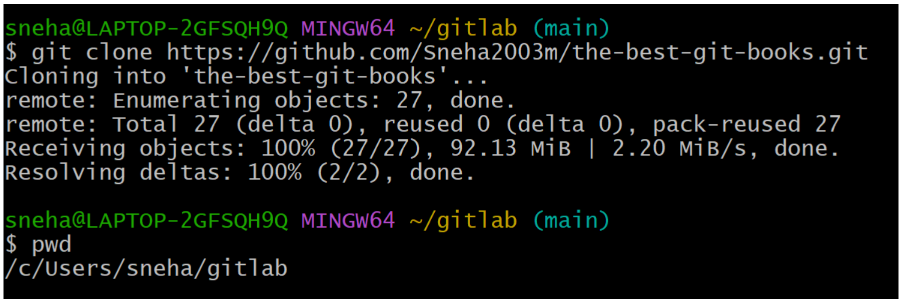
```
sneha@LAPTOP-2GFSQH9Q MINGW64 ~
$ cd gitlab

sneha@LAPTOP-2GFSQH9Q MINGW64 ~/gitlab (main)
$ ls
README.md

sneha@LAPTOP-2GFSQH9Q MINGW64 ~/gitlab (main)
$ touch file1 file2 file3 file4

sneha@LAPTOP-2GFSQH9Q MINGW64 ~/gitlab (main)
$ ls
README.md  file1  file2  file3  file4

sneha@LAPTOP-2GFSQH9Q MINGW64 ~/gitlab (main)
```



```
sneha@LAPTOP-2GFSQH9Q MINGW64 ~/gitlab (main)
$ git clone https://github.com/Sneha2003m/the-best-git-books.git
Cloning into 'the-best-git-books'...
remote: Enumerating objects: 27, done.
remote: Total 27 (delta 0), reused 0 (delta 0), pack-reused 27
Receiving objects: 100% (27/27), 92.13 MiB | 2.20 MiB/s, done.
Resolving deltas: 100% (2/2), done.

sneha@LAPTOP-2GFSQH9Q MINGW64 ~/gitlab (main)
$ pwd
/c/Users/sneha/gitlab
```

5. Result/Output/Writing Summary:

In this experiment we installed git, configured it with our GitHub account and write some commands such as clone to pull remote repository to our local machine, cd, touch, ls, pwd.

Learning outcomes (What I have learnt):

1. Learnt how to install git.

2. Learnt how to configure git with GitHub account.

3. Learnt about some basic commands such as cd and cat.

4. Learnt using git clone command.

5. Also learnt how to add and commit updates to the GitHub account.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			