Experiment -1.1

Student Name: ANAND TIWARI **UID:** 22BDO10022
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

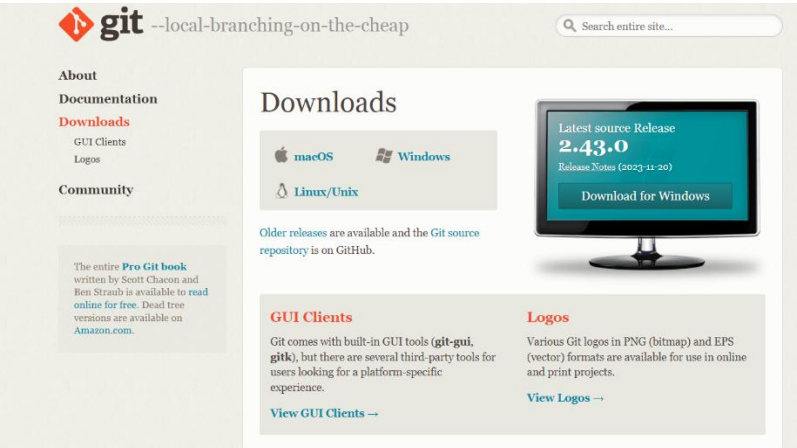
2. Software used: Git Bash and Github.

3. Hardware Used: Computer system.

4. Steps for experiment:

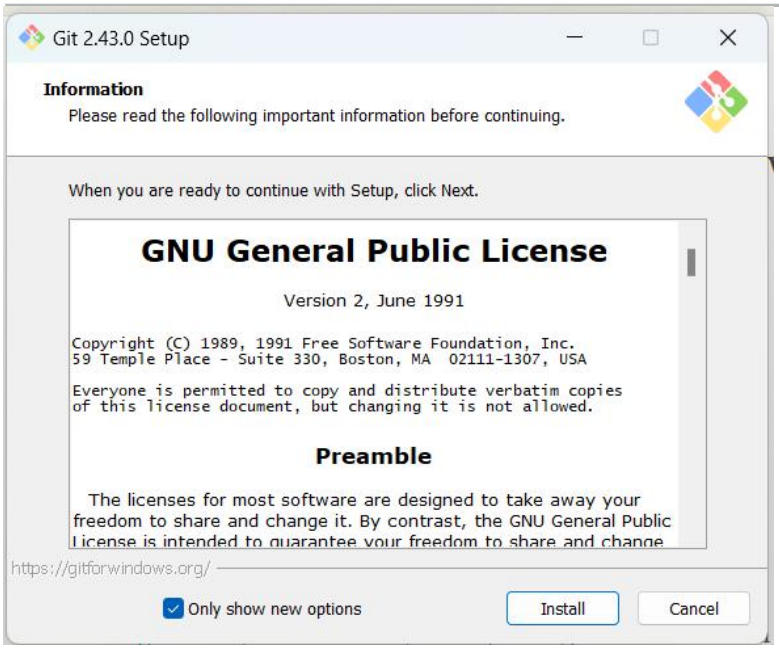
For Installation of Git :

1. Download Git for Windows using Chrome.

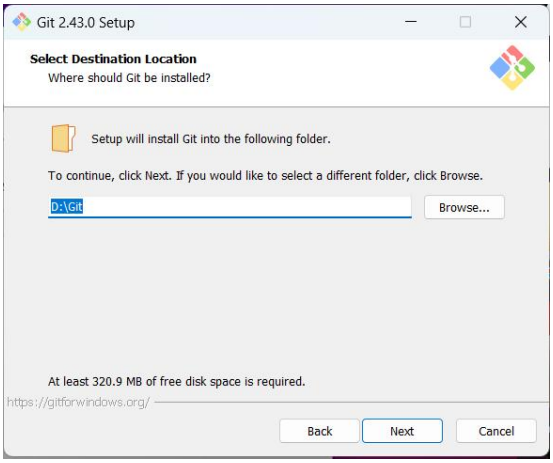


2. Select the Windows option and opt for the 64-bit installer.

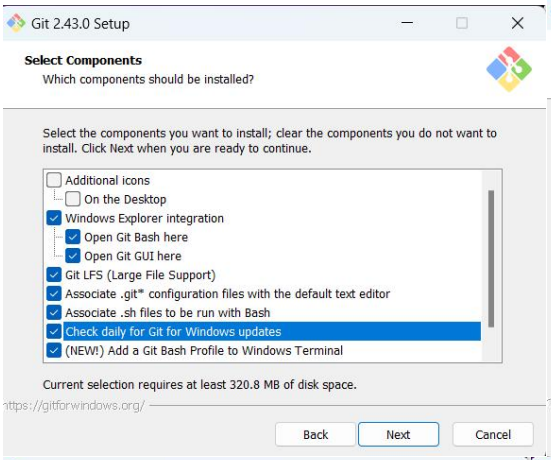
3. Click on install

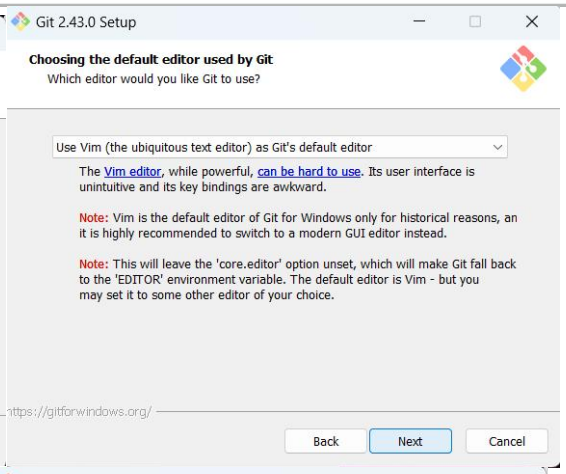


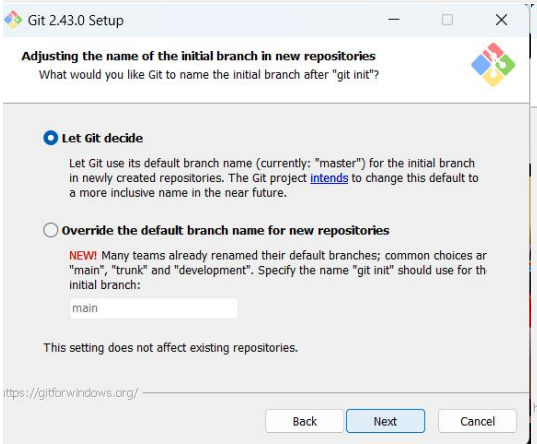
4. Select the location for installation for the git.

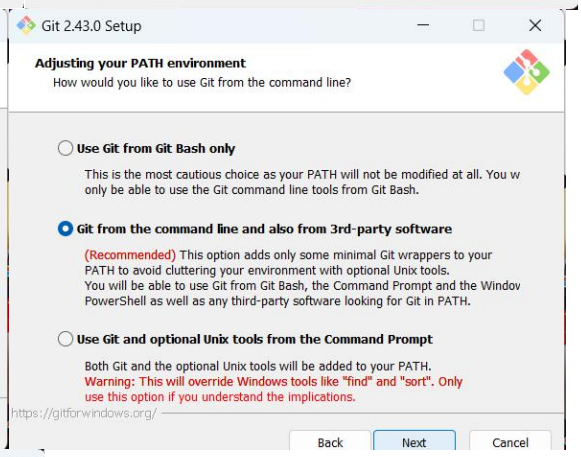


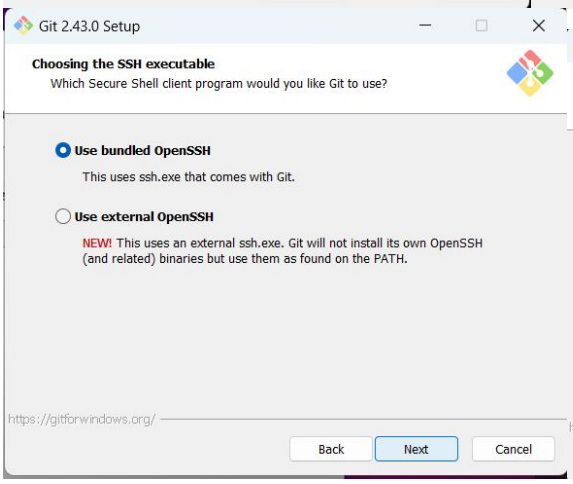
6. Click on next for all default settings.

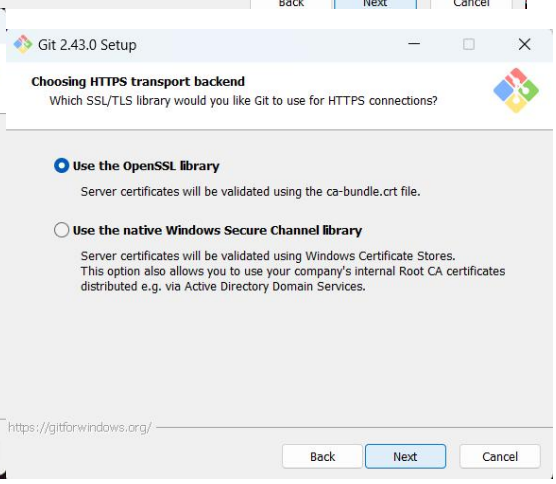

Git 2.43.0 Setup
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Additional icons
On the Desktop
Windows Explorer integration
Open Git Bash here
Open Git GUI 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/
Back
Next
Cancel

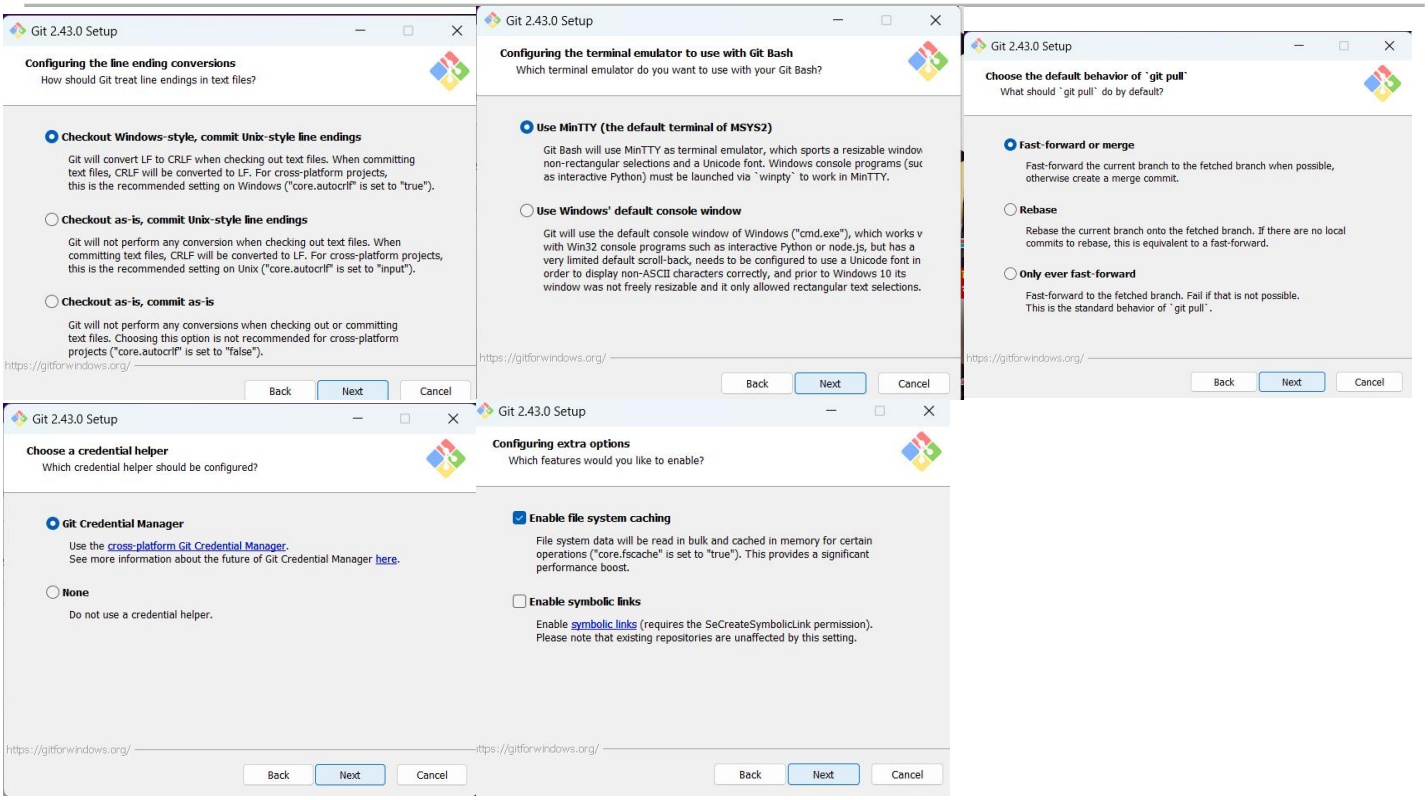

Git 2.43.0 Setup
Choosing the default editor used by Git
Which editor would you like Git to use?
Use Vim (the ubiquitous text editor) as Git's default editor
The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.
Note: Vim is the default editor of Git for Windows only for historical reasons, an 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/
Back
Next
Cancel


Git 2.43.0 Setup
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/
Back
Next
Cancel

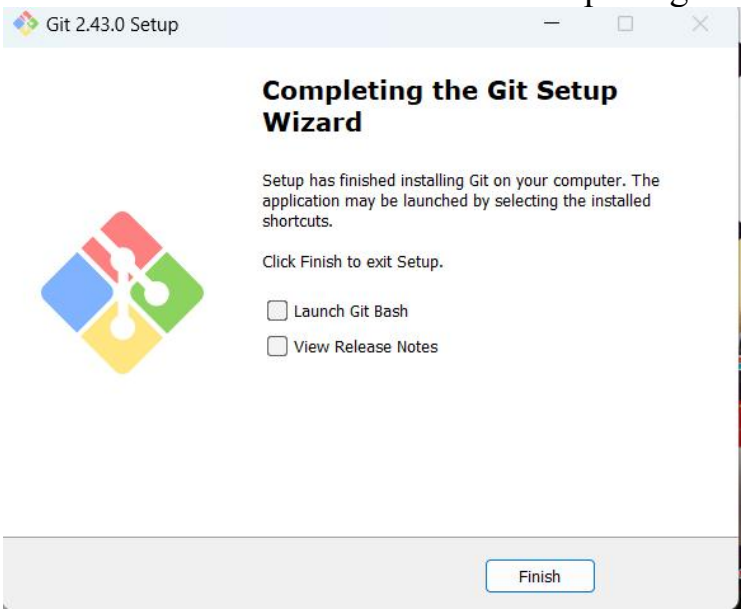

Git 2.43.0 Setup
Adjusting your PATH environment
How would you like to use Git from the command line?
Use Git from Git Bash only
This is the most cautious choice as your PATH will not be modified at all. You w only be able to use the Git command line tools from Git Bash.
Git from the command line and 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/
Back
Next
Cancel


Git 2.43.0 Setup
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/
Back
Next
Cancel


Git 2.43.0 Setup
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/
Back
Next
Cancel



7. Click on Finish after successful completing the installation.



8. To verify git is installed in system or not follow these steps.

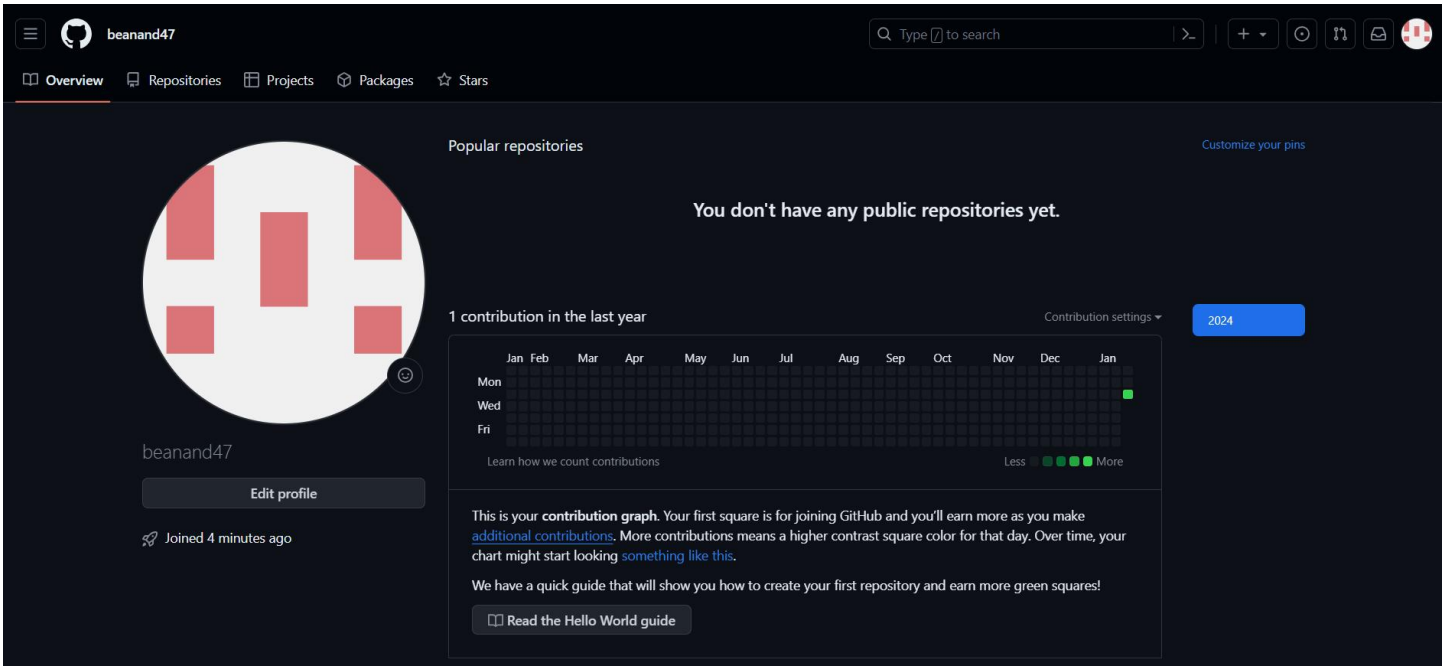
Go to CMD-> type *"git"*

It will give version and description regarding Git

Creation Of Account on Github

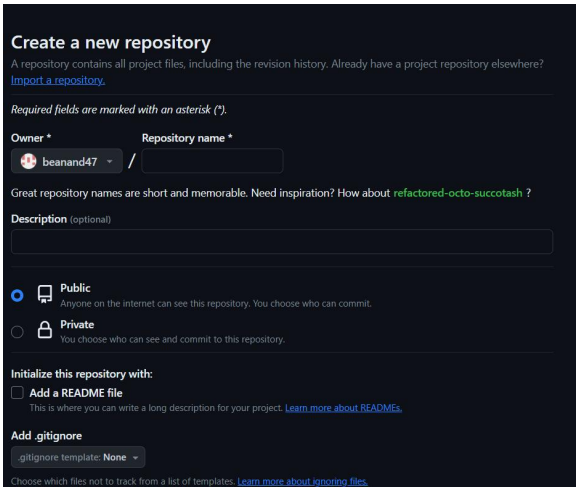
Follow these steps to create acoount on Github

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”



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

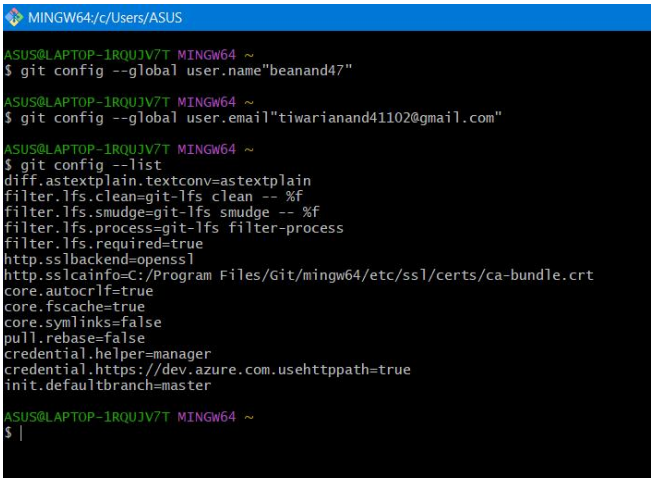
Type code:- *git config --global user.name "beanand47"*

For email

Type code:- *git config --global user.email "tiwarianand41102@gmail.com"*

To verify

Type code:-*git config --list*



For cloning the repository:

Use Command

git clone https://github.com/beanand47/simplework.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.

```
ASUS@LAPTOP-1RQUJV7T MINGW64 ~
$ git clone https://github.com/beanand47/simplework.git
Cloning into 'simplework'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), 12.68 KiB | 249.00 KiB/s, done.

ASUS@LAPTOP-1RQUJV7T MINGW64 ~
$ ls
'3D Objects'/          Links/                                                                        OFFICE/            Videos/                                  pipwin/
 AppData/             'Local Settings'@                                                              OneDrive/         'VirtualBox VMs'/                         pywhatkit_dbs.txt
'Application Data'@    MJKJRegInfo_WLHF6G5GDO5A5AUT6EURO6RMAKGWUK6T                                  Pictures/          ansel/                                   simplework/
 Contacts/             Music/                                                                        PrintHood@         battery-report.html                      test.c
 Cookies@             'My Documents'@                                                                PycharmProjects/   d4ac4633ebd6440fa397b84f1bc94a3c.7z      test.exe*
 Desktop/              NTUSER.DAT                                                                    Recent@            inittk.ini                               useruid.ini
 Documents/            NTUSER.DAT{c242fac9-3c33-11eb-81ad-f02f74478c05}.TM.blf                      'Saved Games'/      inst.ini                                 vmlogs/
 Downloads/            NTUSER.DAT{c242fac9-3c33-11eb-81ad-f02f74478c05}.TMContainer00000000000000000001.regtrans-ms   Searches/   ntuser.dat.LOG1
 Favorites/            NTUSER.DAT{c242fac9-3c33-11eb-81ad-f02f74478c05}.TMContainer00000000000000000002.regtrans-ms   SendTo@     ntuser.dat.LOG2
 IdeaProjects/         NetHood@                                                                     'Start Menu'@       ntuser.ini
 IntelGraphicsProfiles/   Nox_share/                                                                 Templates@         nuuid.ini
```


```
ASUS@LAPTOP-1RQUJV7T MINGW64 ~
$ ls
'3D Objects'/          Links/                                                                        OFFICE/            Videos/                                  pipwin/
 AppData/             'Local Settings'@                                                              OneDrive/         'VirtualBox VMs'/                         pywhatkit_dbs.txt
'Application Data'@    MJKJRegInfo_WLHF6G5GDO5A5AUT6EURO6RMAKGWUK6T                                  Pictures/          ansel/                                   simplework/
 Contacts/             Music/                                                                        PrintHood@         battery-report.html                      test.c
 Cookies@             'My Documents'@                                                                PycharmProjects/   d4ac4633ebd6440fa397b84f1bc94a3c.7z      test.exe*
 Desktop/              NTUSER.DAT                                                                    Recent@            inittk.ini                               useruid.ini
 Documents/            NTUSER.DAT{c242fac9-3c33-11eb-81ad-f02f74478c05}.TM.blf                      'Saved Games'/      inst.ini                                 vmlogs/
 Downloads/            NTUSER.DAT{c242fac9-3c33-11eb-81ad-f02f74478c05}.TMContainer00000000000000000001.regtrans-ms   Searches/   ntuser.dat.LOG1
 Favorites/            NTUSER.DAT{c242fac9-3c33-11eb-81ad-f02f74478c05}.TMContainer00000000000000000002.regtrans-ms   SendTo@     ntuser.dat.LOG2
 IdeaProjects/         NetHood@                                                                     'Start Menu'@       ntuser.ini
 IntelGraphicsProfiles/   Nox_share/                                                                 Templates@         nuuid.ini
```


```
ASUS@LAPTOP-1RQUJV7T MINGW64 ~
$ cd simplework

ASUS@LAPTOP-1RQUJV7T MINGW64 ~/simplework (main)
$ ls
LICENSE

ASUS@LAPTOP-1RQUJV7T MINGW64 ~/simplework (main)
$ pwd
/c/Users/ASUS/simplework
```

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.			