Subject Name: **Source Code Management**

Subject Code: **CS181**

Cluster: **Beta**

Department: **CSE**

CHITKARA UNIVERSITY

Submitted By:
Aadithya
2110990002
G01-B

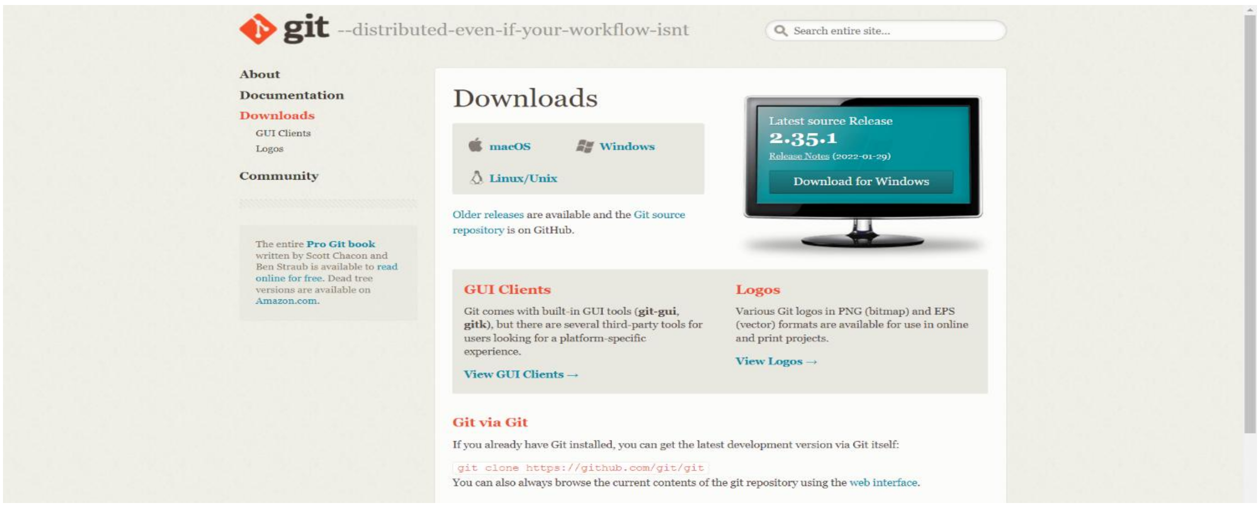
Submitted To:
Dr. Monit Kapoor

List of Programs

Experiment No. 01

Aim: - Setting up of Git client

1. Visit the provided link for the installation of git in your system.
https://git-scm.com/downloads



git --distributed-even-if-your-workflow-isnt

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

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.

Downloads

macOS Windows
Linux/Unix

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

Latest source Release
2.35.1
Release Notes (2022-01-29)
Download for Windows

GUI Clients
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.
View GUI Clients →

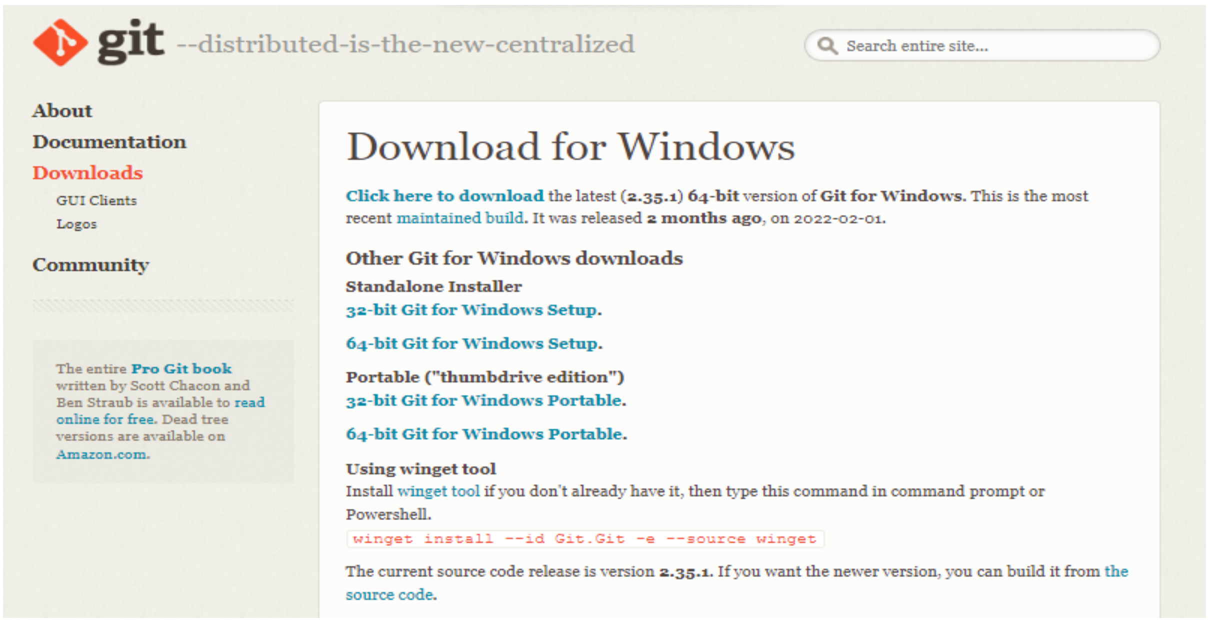
Logos
Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.
View Logos →

Git via Git
If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

You can also always browse the current contents of the git repository using the web interface.

2. After opening this site, you have to select your operating system by clicking on it. Here I will show you the steps for the Windows operating system.



git --distributed-is-the-new-centralized

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

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.

Download for Windows

Click here to download the latest (**2.35.1**) **64-bit** version of **Git for Windows.** This is the most recent maintained build. It was released **2 months ago**, on 2022-02-01.

Other Git for Windows downloads
Standalone Installer
32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

Portable ("thumbdrive edition")
32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

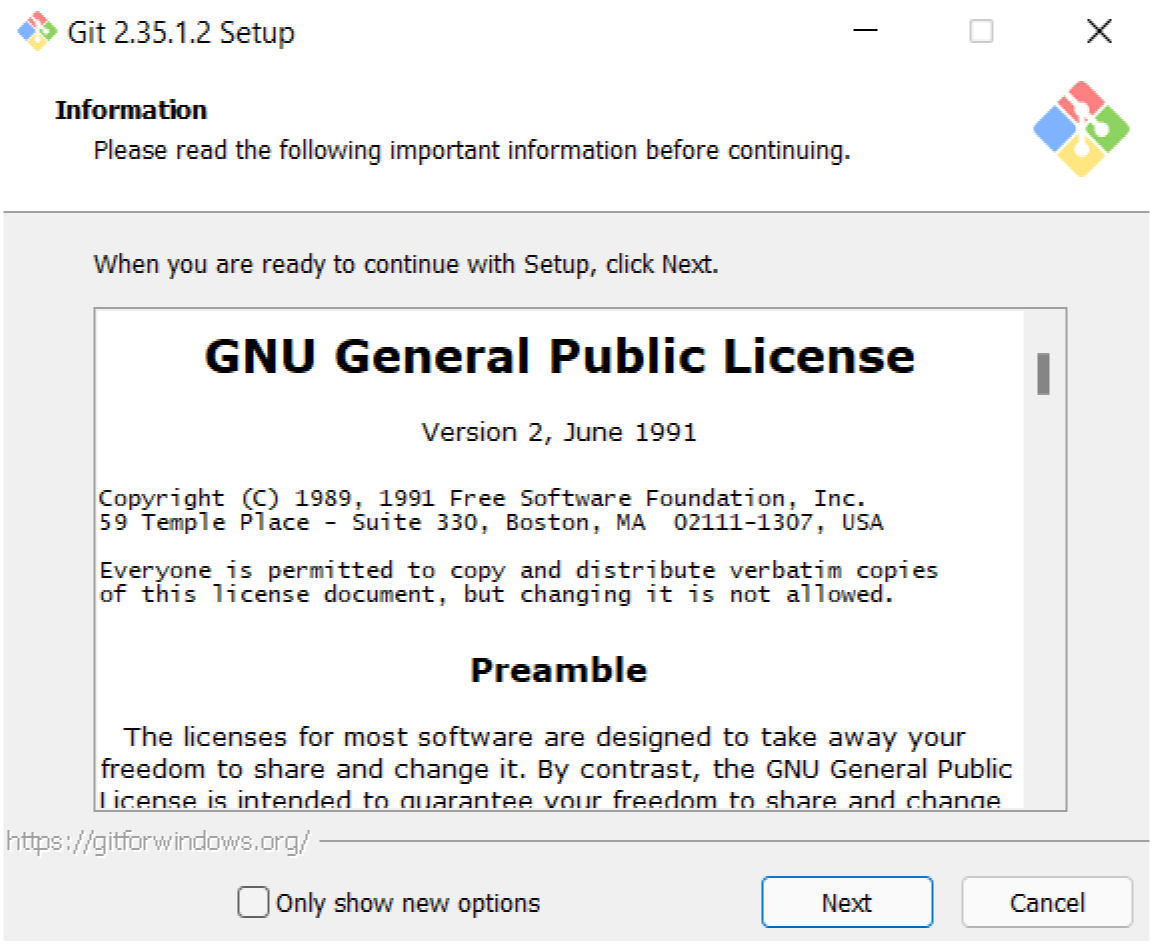
Using winget tool
Install winget tool if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

The current source code release is version **2.35.1**. If you want the newer version, you can build it from the source code.

3. Now select the processor of the system you have. (Most of the system are now of 64-bits) After selecting the processor your download will start.

4. Now you have to open this folder.
5. After opening you will give a notification "Do you want to allow this app to make changes in your PC"
6. Click YES


Git 2.35.1.2 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

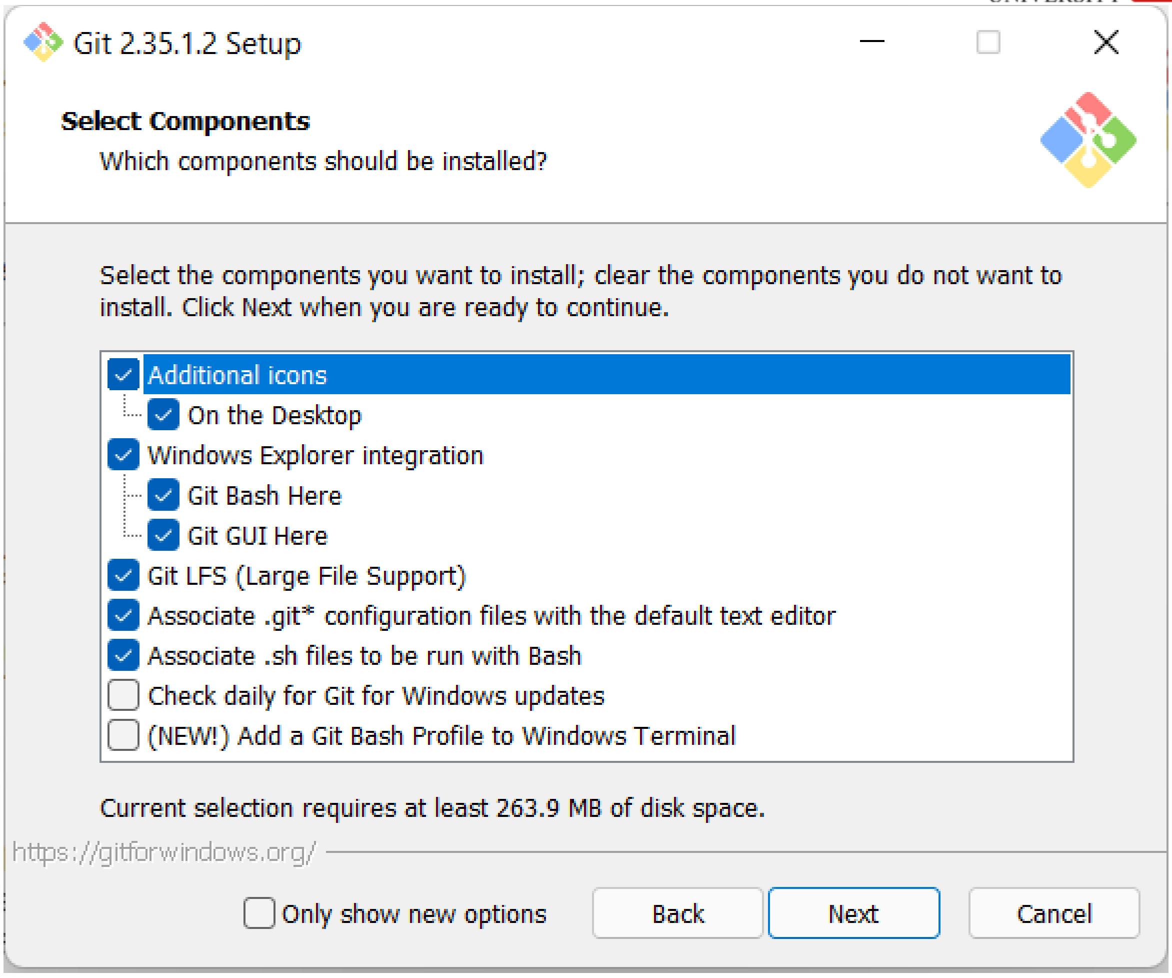
```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

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/

7. Click on Next



Git 2.35.1.2 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.

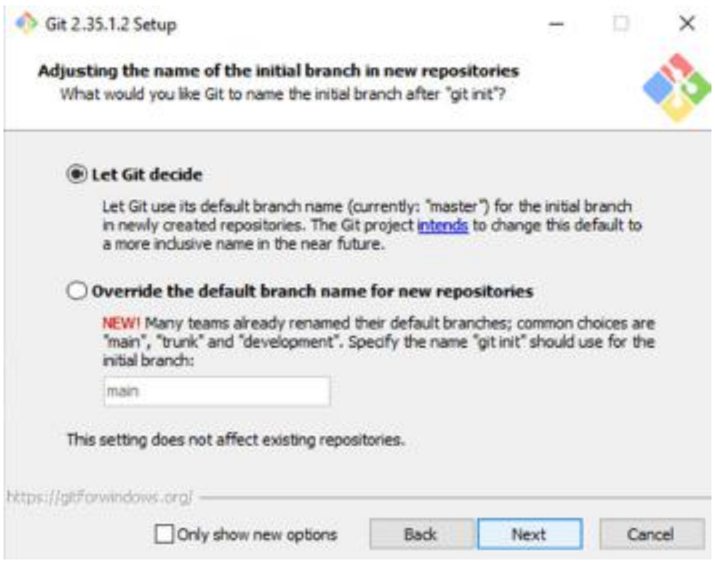
- ✓ Additional icons
  - ✓ On the Desktop
- ✓ Windows Explorer integration
  - ✓ Git Bash Here
  - ✓ Git GUI Here
- ✓ Git LFS (Large File Support)
- ✓ Associate .git* configuration files with the default text editor
- ✓ Associate .sh files to be run with Bash
- ☐ Check daily for Git for Windows updates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

Current selection requires at least 263.9 MB of disk space.

https://gitforwindows.org/

☐ Only show new options

8. Continue clicking on next few times more



Git 2.35.1.2 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 are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

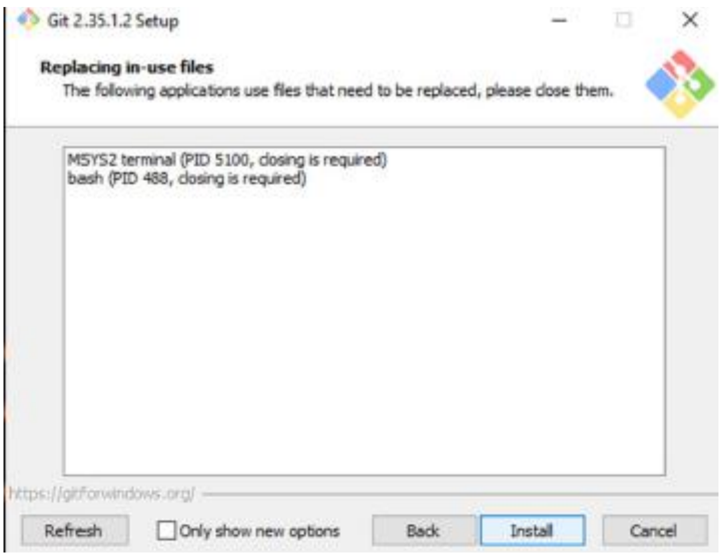
main

This setting does not affect existing repositories.

https://gitforwindows.org/

☐ Only show new options

9. Now select the Install option.


Git 2.35.1.2 Setup

Replacing in-use files

The following applications use files that need to be replaced, please close them.

```
MSYS2 terminal (PID 5100, closing is required)
bash (PID 488, closing is required)
```

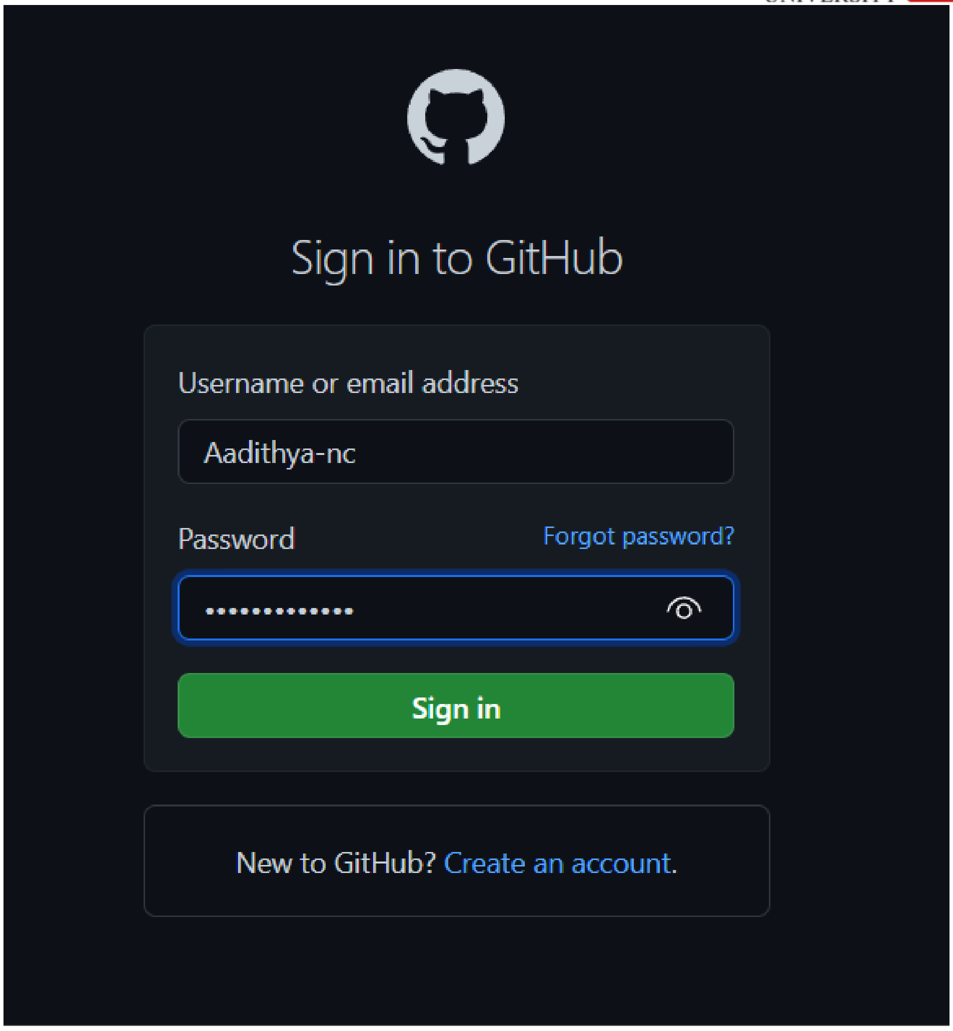
https://gitforwindows.org/

10.Click on Finish after the installation is finished.
11.The installation of the git is finished and now we have to setup git client and GitHub account.

EXPERIMENT NO:2 .

Aim: - Setting up GitHub Account

1. Open your web browser search GitHub login.
2. Click on Create an account if you are a new user or if you have already an account, please login.


Sign in to GitHub

Username or email address

Aadithya-nc

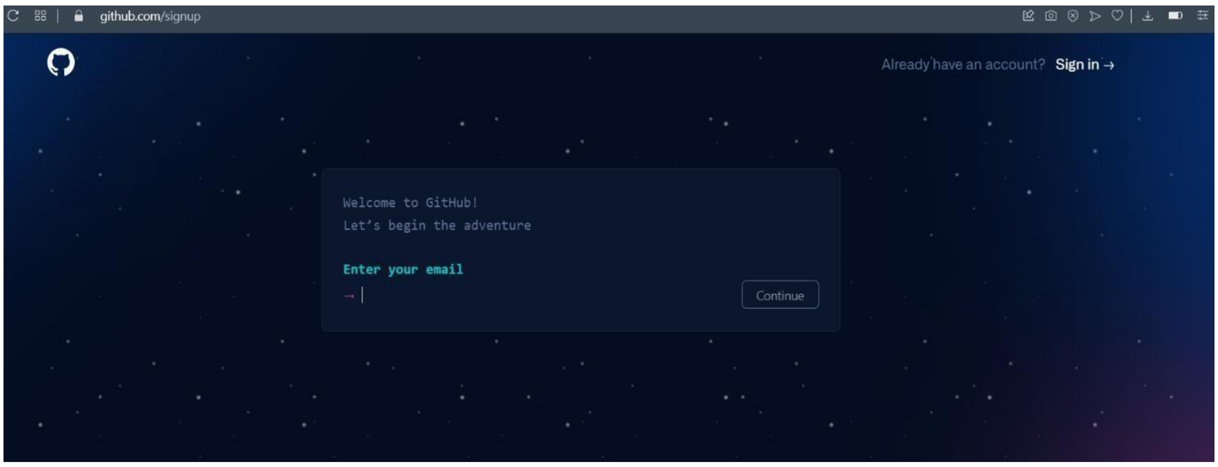
Password

Forgot password?

Sign in

New to GitHub? Create an account.

3. After Clicking on create a new account you will be redirected to a new page where you have to enter your email id which you want to use for your account. Now enter your password you want to create for your GitHub account. After that you will be asked to enter your username.


github.com/signup

Already have an account? Sign in →

Welcome to GitHub!
Let's begin the adventure

Enter your email
→

Continue

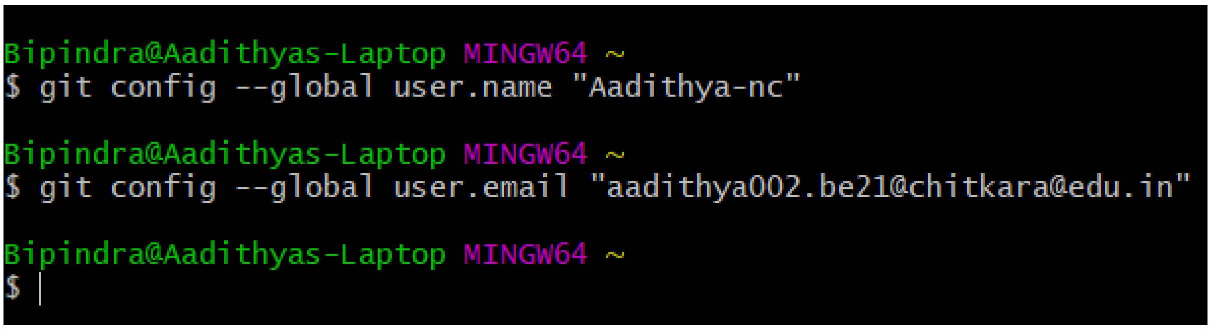
4. Now Click on Create Account.
5. Verify it from your email and you are all set to go.

EXPERIMENT NO:3

Aim: - Program to Generate logs

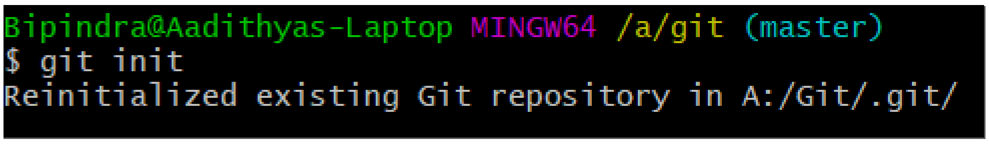
• When we use GIT for the first time, we have to give the user's name and email so that if I am going to change in project, it will be visible to all.
For this, we use command →

"Git config --global user.name "Name"
"Git config --global user. Email "email"



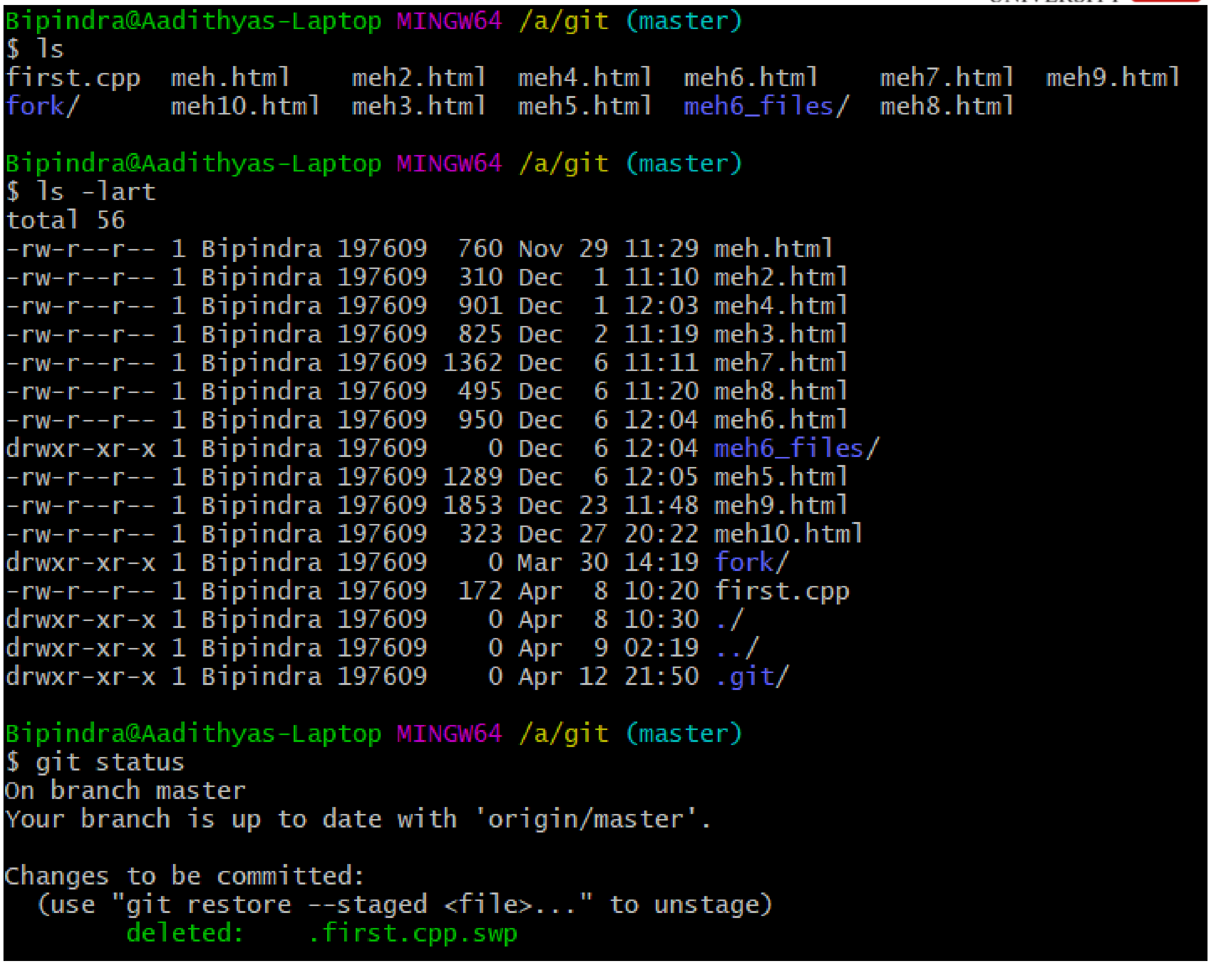
```
Bipindra@Aadithyas-Laptop MINGW64 ~
$ git config --global user.name "Aadithya-nc"

Bipindra@Aadithyas-Laptop MINGW64 ~
$ git config --global user.email "aadithya002.be21@chitkara@edu.in"

Bipindra@Aadithyas-Laptop MINGW64 ~
$ |
```

First of all, create a local repository using Git. For this, you have to make a folder in your device, right click and select "Git Bash Here". This opens the Git terminal. To create a new local repository, use the command "git unit" and it creates a folder. git.

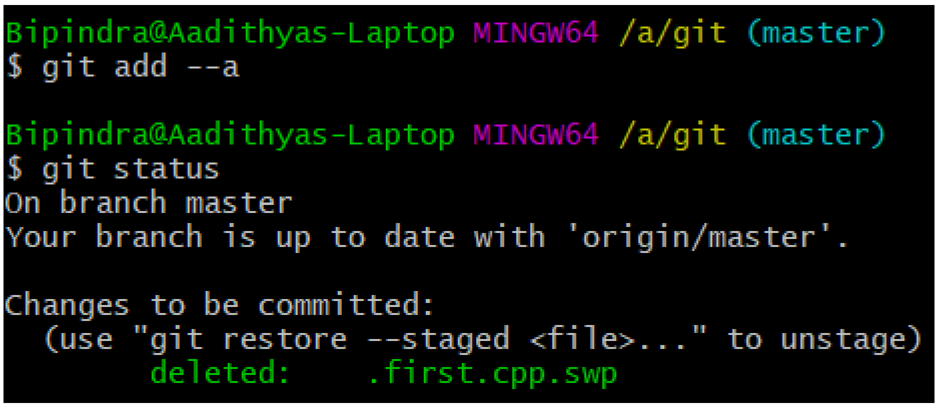


```
Bipindra@Aadithyas-Laptop MINGW64 /a/git (master)
$ git init
Reinitialized existing Git repository in A:/Git/.git/
```

Some Important Commands:
• **ls** → It gives the file names in the folder.
• **ls -lard** → Gives the hidden files also.
• **git status** → Displays the state of the working directory and the staged snapshot.
• **touch filename** → This command creates a new file in the repository.
• **Clear** → It clears the terminal.
• **rm** -rf. git → It removes the repository.
• **git log** → displays all of the commits in a repository's history
• **git diff** → It compares my working tree to staging area.
Now, we have to create some files in the repository. Suppose we created index.html. Now type git status:

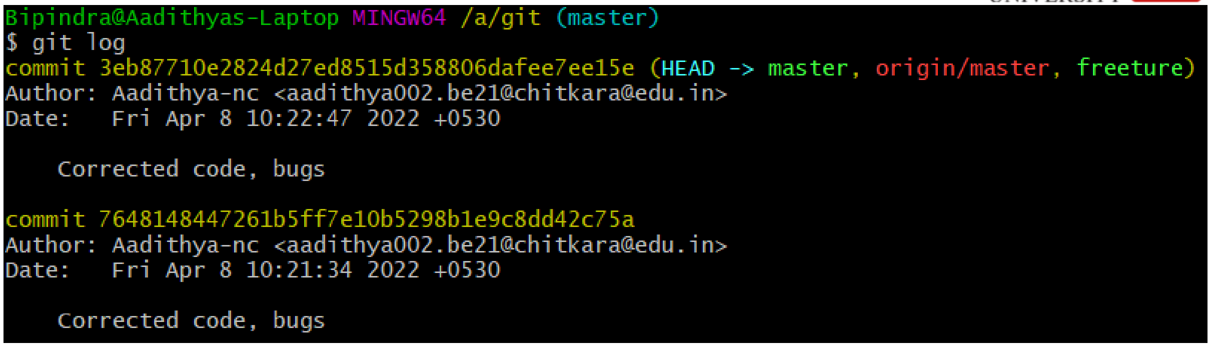

```
Bipindra@Aadithyas-Laptop MINGW64 /a/git (master)
$ ls
first.cpp   meh.html     meh2.html   meh4.html   meh6.html     meh7.html   meh9.html
fork/       meh10.html   meh3.html   meh5.html   meh6_files/   meh8.html

Bipindra@Aadithyas-Laptop MINGW64 /a/git (master)
$ ls -lart
total 56
-rw-r--r-- 1 Bipindra 197609  760 Nov 29 11:29 meh.html
-rw-r--r-- 1 Bipindra 197609  310 Dec  1 11:10 meh2.html
-rw-r--r-- 1 Bipindra 197609  901 Dec  1 12:03 meh4.html
-rw-r--r-- 1 Bipindra 197609  825 Dec  2 11:19 meh3.html
-rw-r--r-- 1 Bipindra 197609 1362 Dec  6 11:11 meh7.html
-rw-r--r-- 1 Bipindra 197609  495 Dec  6 11:20 meh8.html
-rw-r--r-- 1 Bipindra 197609  950 Dec  6 12:04 meh6.html
drwxr-xr-x 1 Bipindra 197609    0 Dec  6 12:04 meh6_files/
-rw-r--r-- 1 Bipindra 197609 1289 Dec  6 12:05 meh5.html
-rw-r--r-- 1 Bipindra 197609 1853 Dec 23 11:48 meh9.html
-rw-r--r-- 1 Bipindra 197609  323 Dec 27 20:22 meh10.html
drwxr-xr-x 1 Bipindra 197609    0 Mar 30 14:19 fork/
-rw-r--r-- 1 Bipindra 197609  172 Apr  8 10:20 first.cpp
drwxr-xr-x 1 Bipindra 197609    0 Apr  8 10:30 ./
drwxr-xr-x 1 Bipindra 197609    0 Apr  9 02:19 ../
drwxr-xr-x 1 Bipindra 197609    0 Apr 12 21:50 .git/

Bipindra@Aadithyas-Laptop MINGW64 /a/git (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    .first.cpp.swp
```

You can see that index.html is in red colour that means it is an untracked file. Now firstly add the file in staging area and then commit the file.
For this, use command →
git add --a [For add all the files in staging area.]


```
Bipindra@Aadithyas-Laptop MINGW64 /a/git (master)
$ git add --a

Bipindra@Aadithyas-Laptop MINGW64 /a/git (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    .first.cpp.swp
```

git commit -m "write any message" [For commit the file]

• **git log:** The git log command displays a record of the commits in a Git repository. By default, the git log command displays a commit hash, the commit message, and other commit metadata.

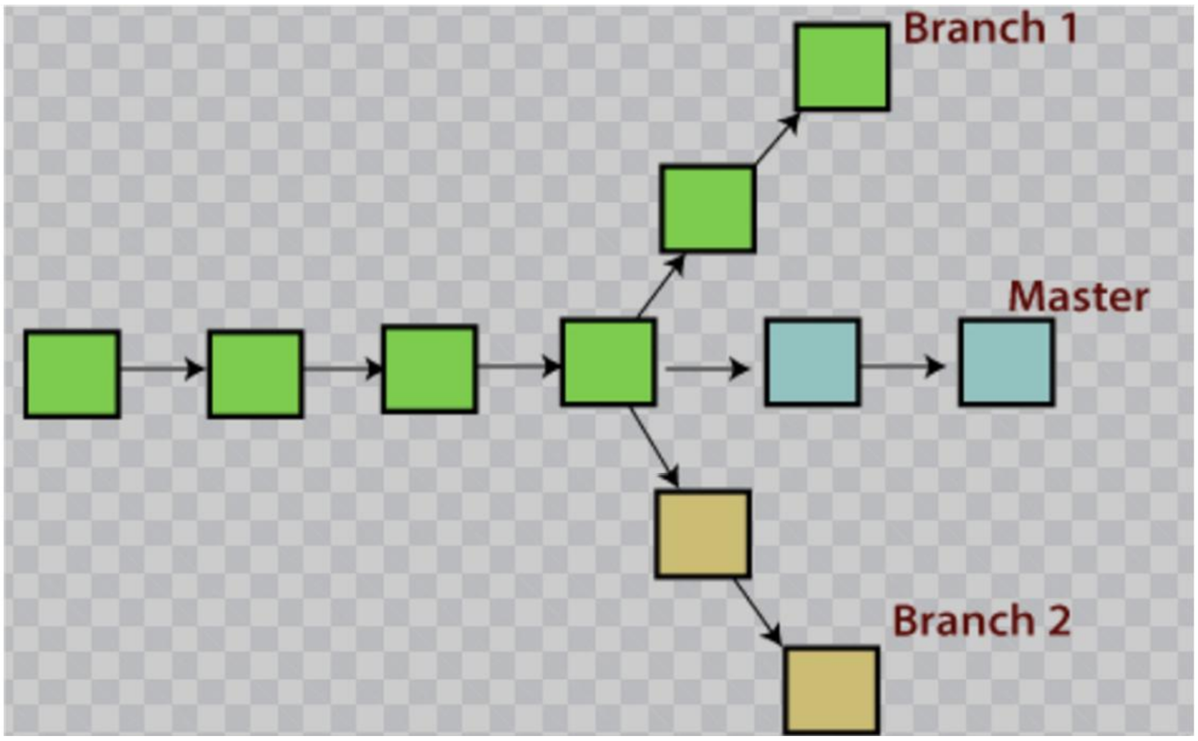


```
Bipindra@Aadithyas-Laptop MINGW64 /a/git (master)
$ git log
commit 3eb87710e2824d27ed8515d358806dafee7ee15e (HEAD -> master, origin/master, freeture)
Author: Aadithya-nc <aadithya002.be21@chitkara@edu.in>
Date:   Fri Apr 8 10:22:47 2022 +0530

    Corrected code, bugs

commit 7648148447261b5ff7e10b5298b1e9c8dd42c75a
Author: Aadithya-nc <aadithya002.be21@chitkara@edu.in>
Date:   Fri Apr 8 10:21:34 2022 +0530

    Corrected code, bugs
```

EXPERIMENT NO:4

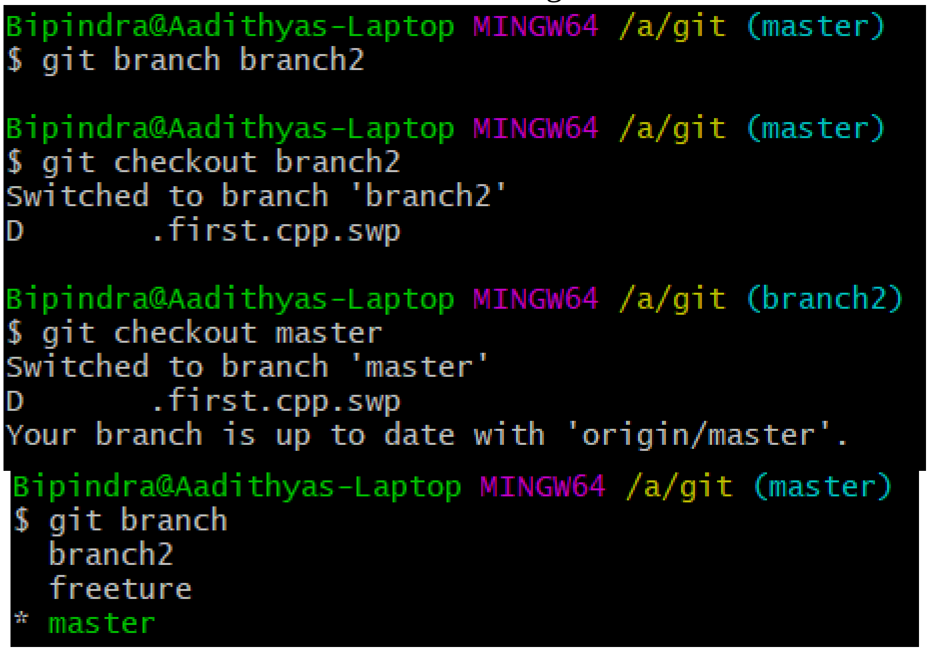
Aim: - Create and visualize branches

A branch in Git is an independent line of work (a pointer to a specific commit). It allows users to create a branch from the original code (master branch) and isolate their work. Branches allow you to work on different parts of a project without impacting the main branch.



Let us see the command of it:
Firstly, add a new branch, let us suppose the branch name is activity1.
For this use command →
git branch name [adding new branch]
git branch [use to see the branch's names]

git checkout branch name [use to switch to the given branch]



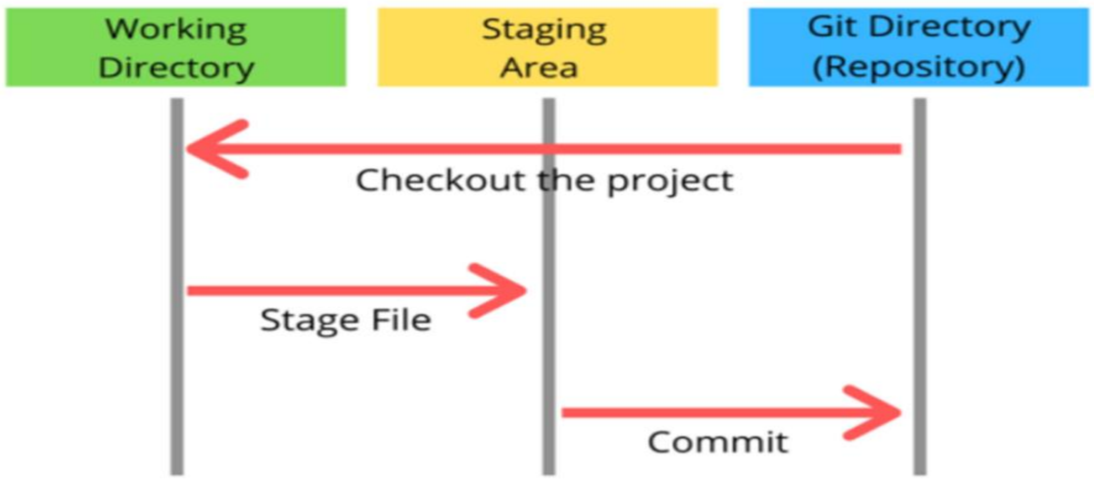
```
Bipindra@Aadithyas-Laptop MINGW64 /a/git (master)
$ git branch branch2

Bipindra@Aadithyas-Laptop MINGW64 /a/git (master)
$ git checkout branch2
Switched to branch 'branch2'
D       .first.cpp.swp

Bipindra@Aadithyas-Laptop MINGW64 /a/git (branch2)
$ git checkout master
Switched to branch 'master'
D       .first.cpp.swp
Your branch is up to date with 'origin/master'.
Bipindra@Aadithyas-Laptop MINGW64 /a/git (master)
$ git branch
  branch2
  freeture
* master
```

As, we can see we can switch the branches by using "git checkout branch name" command. After, we are done with the branch that we have made to do master branch better we can merge it with the master branch.
This can be done by using command: -
git merge branch name [that we have to add]

EXPERIMENT NO:5

Aim: - Git lifecycle description

. Now let's understand the three-stage architecture of Git

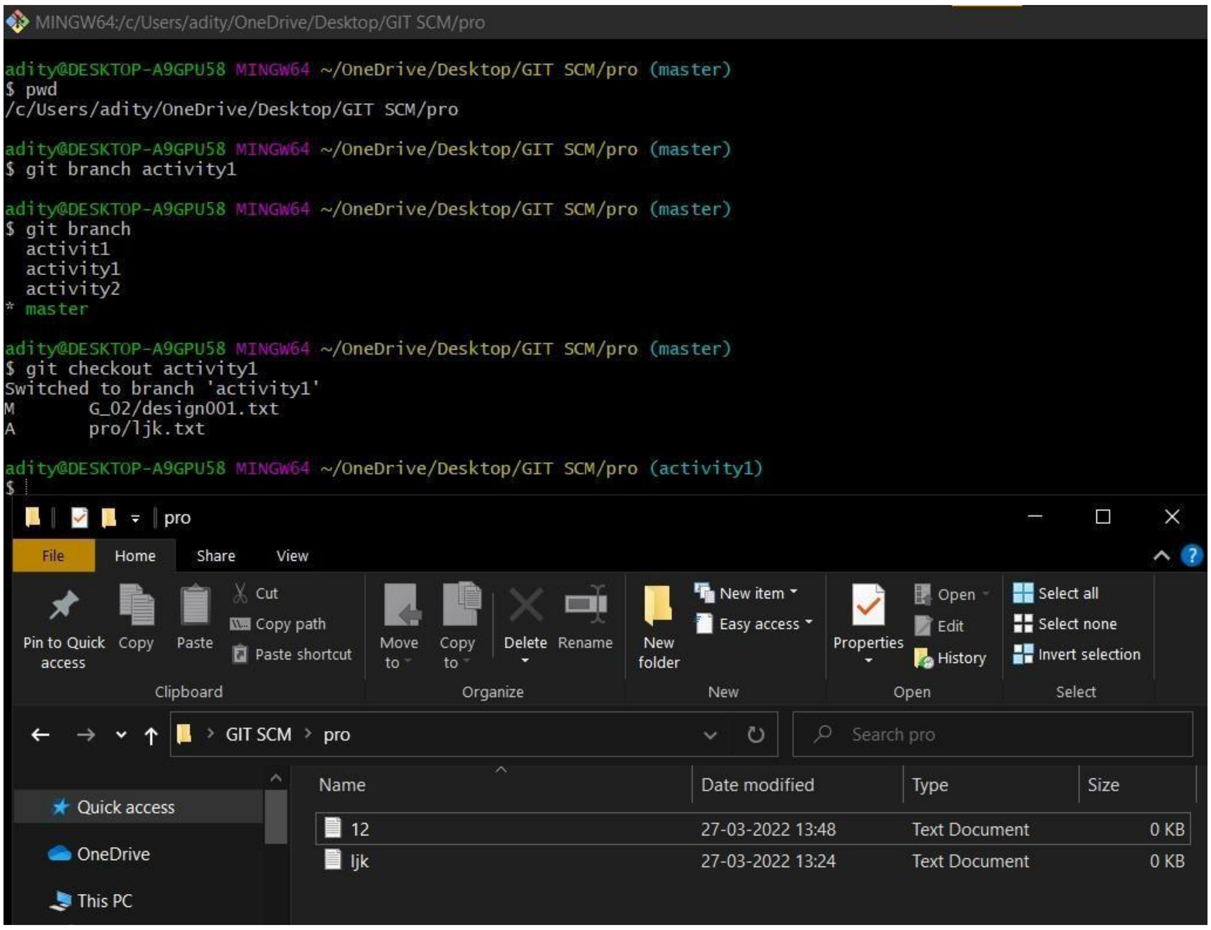


Git – Three Stage Architecture

• **Working Directory:** This is the directory that we've initialized, and here all the changes are made to commit on GitHub.
• **Staging Area**: This is where we first put out code or files of the working

repository. The command that we use to stage code is, "git add --a", "git add Filenames" or "git add -A". In simple terms, staging means telling Git what files we want to commit (new untracked files, modified files, or deleted files).

• **Git directory(repository):** This is where all the commits are stored whenever we make a commit. We can revert to an older version of or project using the **"Git checkout" command** from this directory.



```
MINGW64:/c/Users/adity/OneDrive/Desktop/GIT SCM/pro

adity@DESKTOP-A9GPU58 MINGW64 ~/OneDrive/Desktop/GIT SCM/pro (master)
$ pwd
/c/Users/adity/OneDrive/Desktop/GIT SCM/pro

adity@DESKTOP-A9GPU58 MINGW64 ~/OneDrive/Desktop/GIT SCM/pro (master)
$ git branch activity1

adity@DESKTOP-A9GPU58 MINGW64 ~/OneDrive/Desktop/GIT SCM/pro (master)
$ git branch
  activit1
  activity1
  activity2
* master

adity@DESKTOP-A9GPU58 MINGW64 ~/OneDrive/Desktop/GIT SCM/pro (master)
$ git checkout activity1
Switched to branch 'activity1'
M       G_02/design001.txt
A       pro/ljk.txt

adity@DESKTOP-A9GPU58 MINGW64 ~/OneDrive/Desktop/GIT SCM/pro (activity1)
$
```

Name	Date modified	Type	Size
12	27-03-2022 13:48	Text Document	0 KB
ljk	27-03-2022 13:24	Text Document	0 KB