Subject Name: **Source Code Management**

Subject Code: **CS181**

Cluster: **Beta**

Department: **DCSE**

Submitted By:
Ananya Thakur
2110990180
G3

Submitted To:
Dr. Deepak Thakur

Aim: Setting up of Git Client

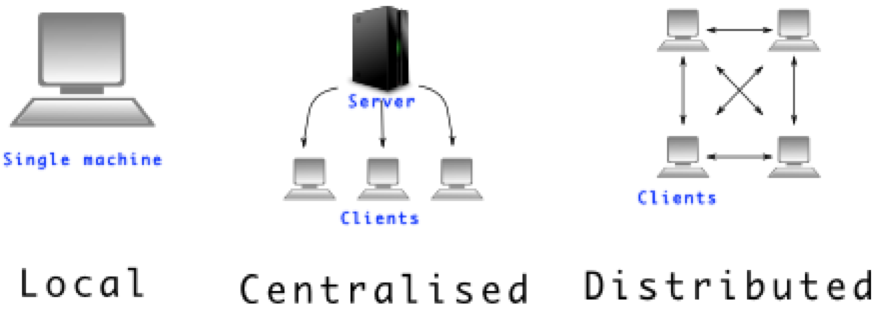
Theory:

What is Git?

Git is a software used for tracking changes in any set of files, usually used for coordinating work among members of a team.

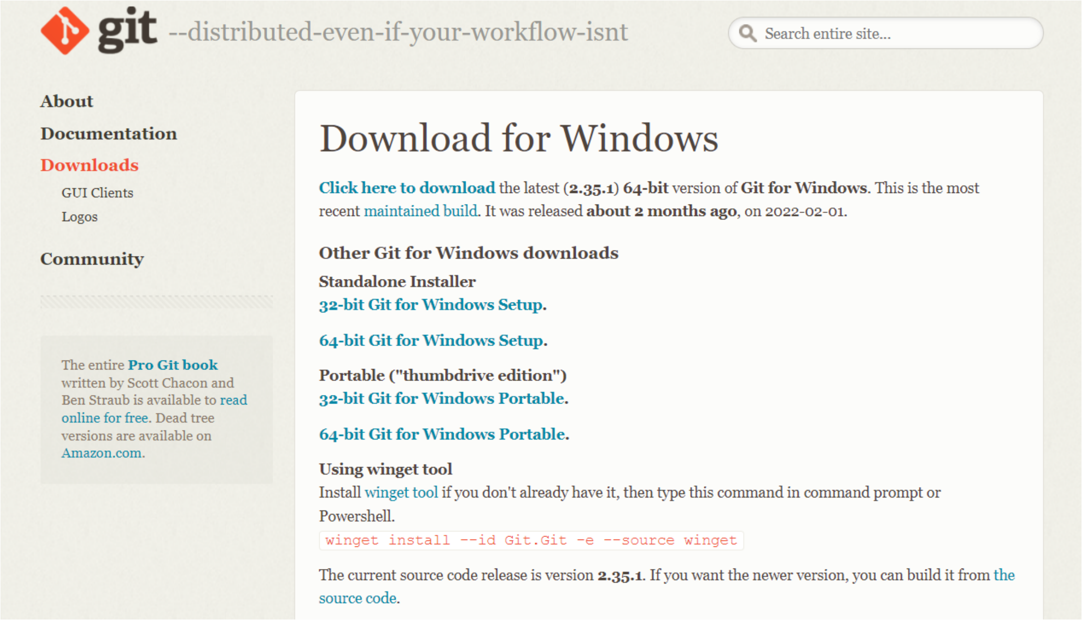
History of VCS:

- **Local VCS:** No internet is needed because it uses a database to keep track of files.
- **Centralized VCS:** Centralized version control systems are based on the idea that there is a single "central" copy of your project somewhere (probably on a server), and programmers will "commit" their changes to this central copy. "Committing" a change simply means recording the change in the central system.
- **Distributed VCS:** A type of version control where the complete codebase including its full version history is mirrored on every developer's computer.



How to install GIT on Windows?

There are also a few ways to install Git on Windows. The most official build is available for download on the Git website. Just go to https://git-scm.com/download/win and the download will start automatically. Note that this is a project called Git for Windows, which is separate from Git itself; for more information on it, go to https://gitforwindows.org.



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 **about 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.

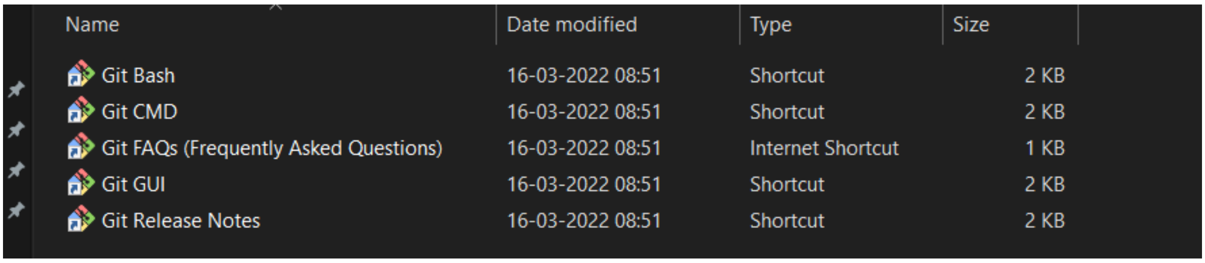
64-bit Git for Windows Portable.

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.



Name	Date modified	Type	Size
Git Bash	16-03-2022 08:51	Shortcut	2 KB
Git CMD	16-03-2022 08:51	Shortcut	2 KB
Git FAQs (Frequently Asked Questions)	16-03-2022 08:51	Internet Shortcut	1 KB
Git GUI	16-03-2022 08:51	Shortcut	2 KB
Git Release Notes	16-03-2022 08:51	Shortcut	2 KB

Check version of git by using git –version command.

Experiment No. 02

Aim: Setting up GitHub Account

Theory:

What is GitHub?

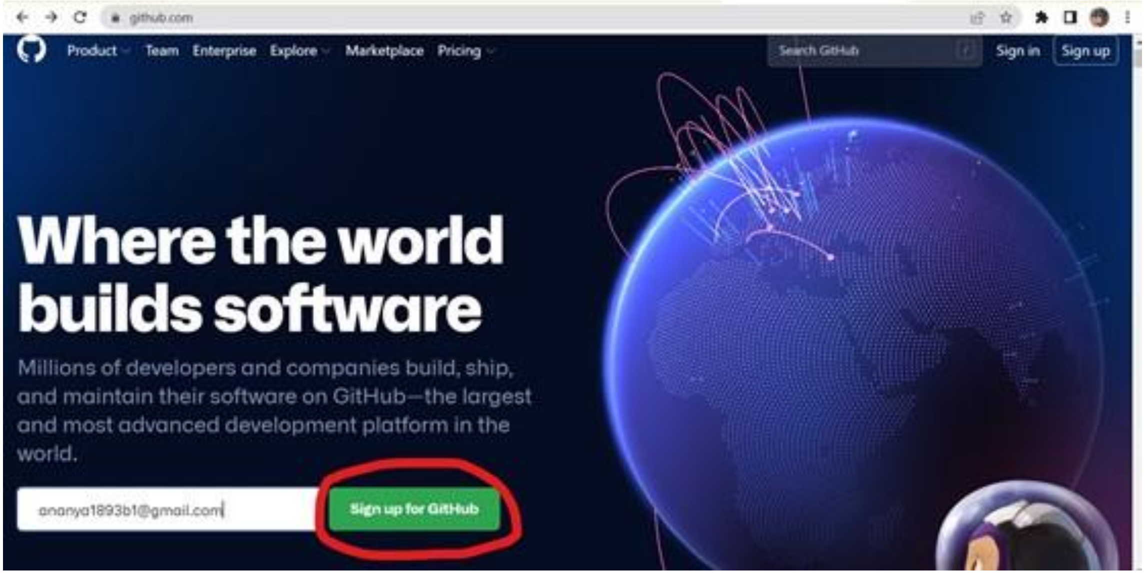
GitHub is a code hosting platform for version control and collaboration. In other words, it manages repositories.

Advantages:

- It makes it easy to contribute to Open-Source projects.
- Track changes in your code across versions.

Procedure:

Search for GitHub in any search engine or https://github.com/signup



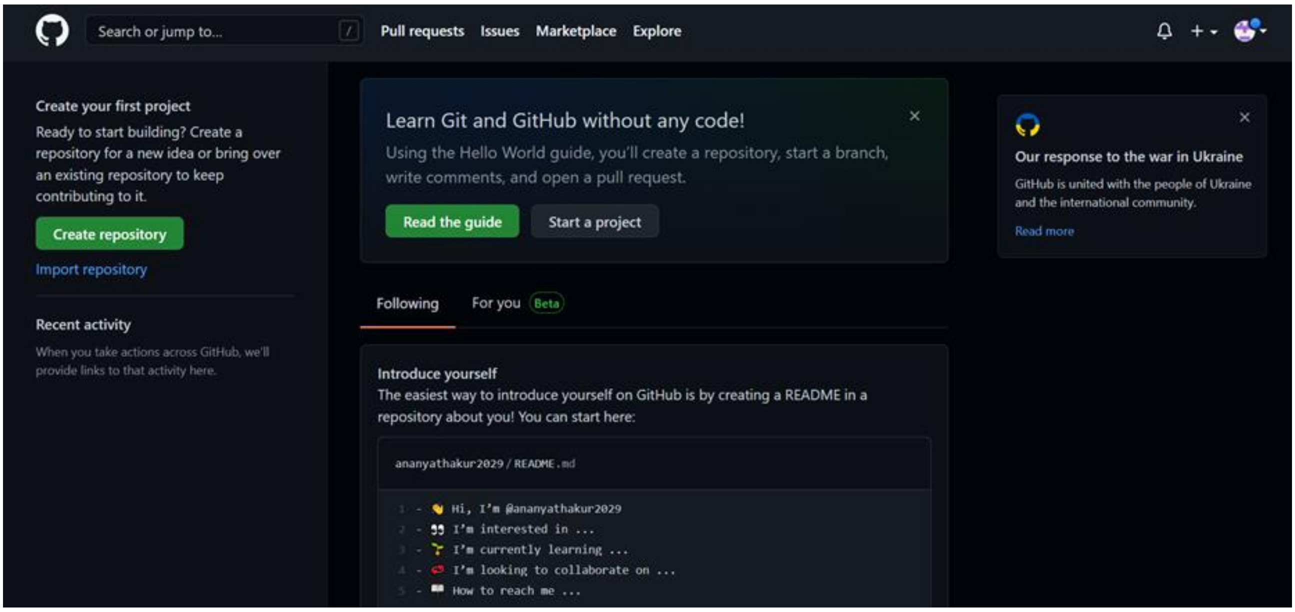
Where the world builds software

Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.

ananya1893b1@gmail.com

If you're a new user add your email and click on **Sign up for GitHub**. Otherwise click on **Sign In** at the top right corner

Signing into GitHub:



Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Our response to the war in Ukraine

GitHub is united with the people of Ukraine and the international community.

Introduce yourself

The easiest way to introduce yourself on GitHub is by creating a README in a repository about you! You can start here:

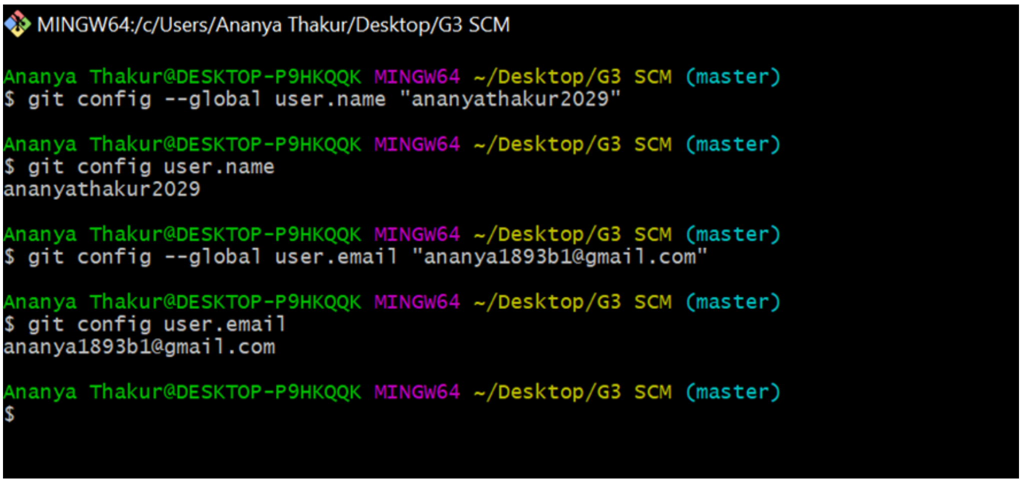
```
ananyathakur2029 / README.md

- 👋 Hi, I'm @ananyathakur2029
- 👀 I'm interested in ...
- 🌱 I'm currently learning ...
- 💞 I'm looking to collaborate on ...
- 📫 How to reach me ...
```

Linking GitHub account with Git Bash:

Username:
git config --global user.name “username in github”

Email:
git config --global user.email “your email in github”

Check Username & Email:
git config user.name
git config user.email



```
MINGW64:/c/Users/Ananya Thakur/Desktop/G3 SCM

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (master)
$ git config --global user.name "ananyathakur2029"

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (master)
$ git config user.name
ananyathakur2029

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (master)
$ git config --global user.email "ananya1893b1@gmail.com"

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (master)
$ git config user.email
ananya1893b1@gmail.com

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (master)
$
```

Aim: Program to Generate log

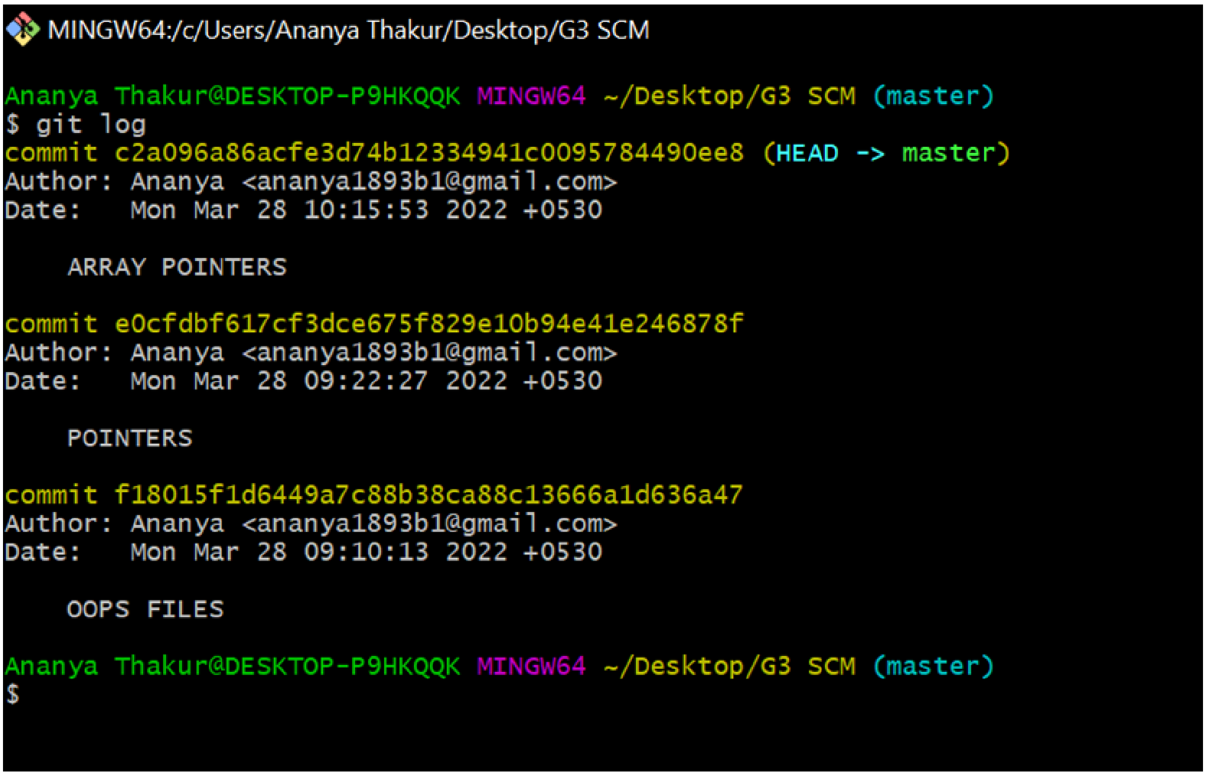
Theory:

Git Logs:

Logs are nothing but the history which we can see in Git by using the code Git log. It contains all the past commits, insertions and deletions which can be seen anytime.

Why do we need logs?

Logs help us to check the changes made in code or files and by whom. It also contains the details of insertions and deletions and also the time it was changed at.



```
Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (master)
$ git log
commit c2a096a86acfe3d74b12334941c0095784490ee8 (HEAD -> master)
Author: Ananya <ananya1893b1@gmail.com>
Date:   Mon Mar 28 10:15:53 2022 +0530

    ARRAY POINTERS

commit e0cfdbf617cf3dce675f829e10b94e41e246878f
Author: Ananya <ananya1893b1@gmail.com>
Date:   Mon Mar 28 09:22:27 2022 +0530

    POINTERS

commit f18015f1d6449a7c88b38ca88c13666a1d636a47
Author: Ananya <ananya1893b1@gmail.com>
Date:   Mon Mar 28 09:10:13 2022 +0530

    OOPS FILES

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (master)
$
```

- **Use command** git log **to access logs.**

Aim: Create and visualize branches

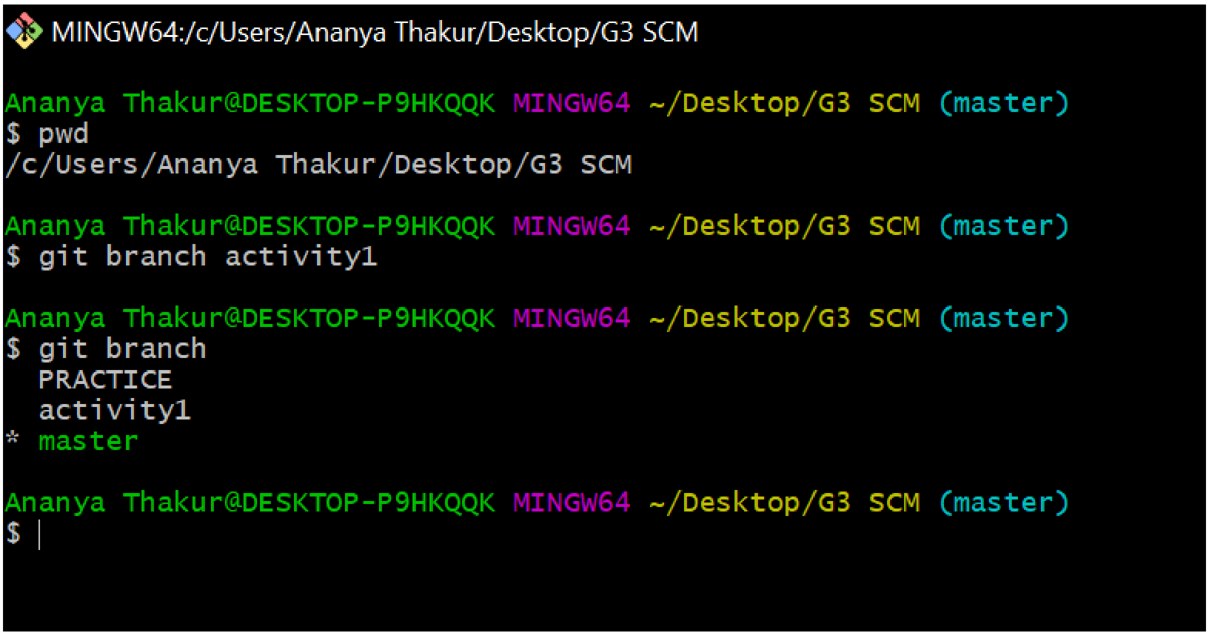
Theory:

How to create branches?

The main branch in git is called the master branch. But we can make branches out of this main master branch. All the files present in master can be shown in branch but the files which are created in branch are not shown in master branch. We can also merge both the parent (master) and child (other branches).

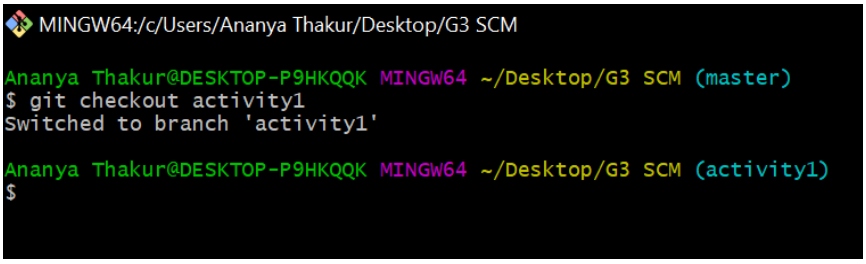
1. **For creating a new branch:** git branch “name of the branch”

2. **To check how many branches we have:** git branch



```
MINGW64:/c/Users/Ananya Thakur/Desktop/G3 SCM

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (master)
$ pwd
/c/Users/Ananya Thakur/Desktop/G3 SCM

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (master)
$ git branch activity1

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (master)
$ git branch
  PRACTICE
  activity1
* master

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (master)
$ |
```

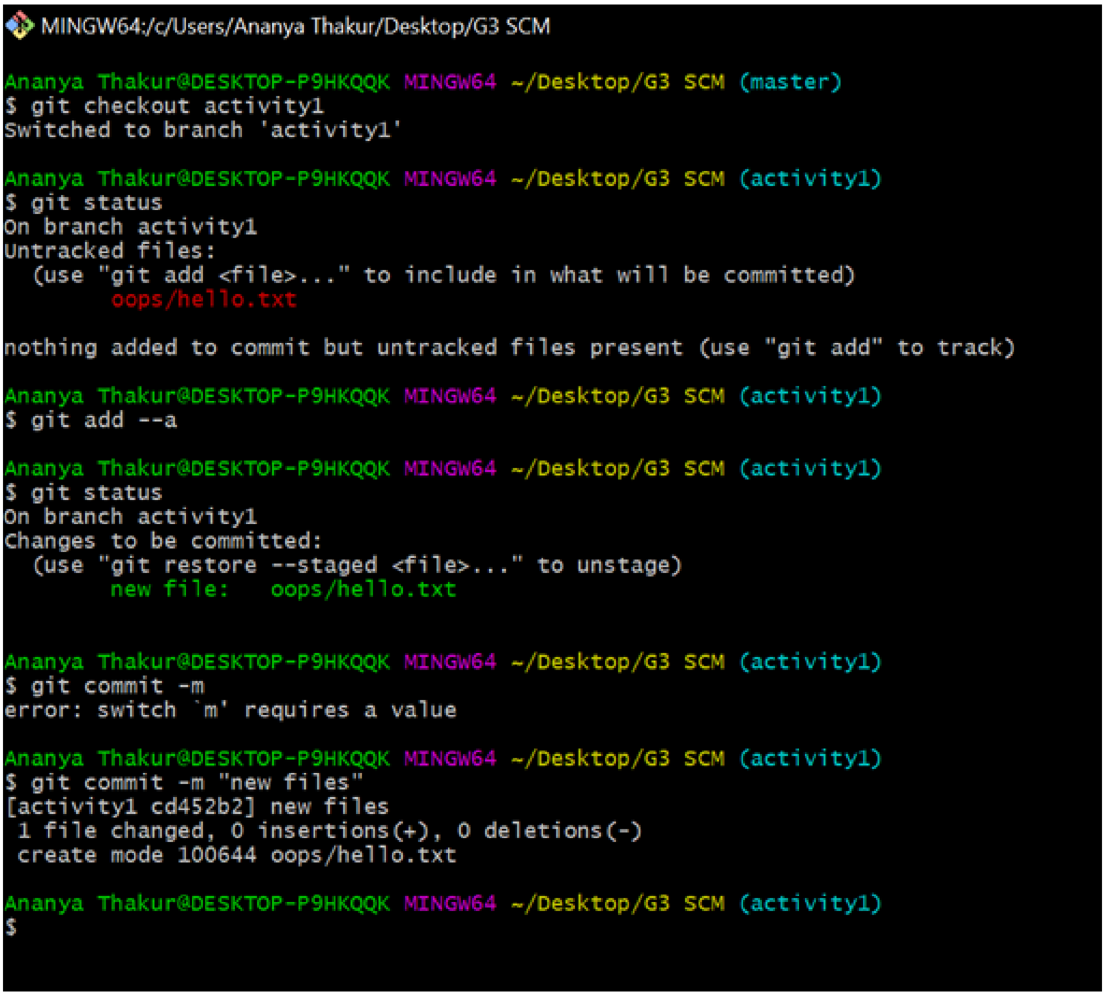
3. **To change the present working branch:** git checkout “name of the branch”



```
MINGW64:/c/Users/Ananya Thakur/Desktop/G3 SCM

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (master)
$ git checkout activity1
Switched to branch 'activity1'

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (activity1)
$
```

Visualizing branches:

To visualize, we have to create a new file in the new branch “activity1” instead of the master branch. After this we have to do three step architecture i.e working directory, staging area and git repository.



```
MINGW64:/c/Users/Ananya Thakur/Desktop/G3 SCM

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (master)
$ git checkout activity1
Switched to branch 'activity1'

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (activity1)
$ git status
On branch activity1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        oops/hello.txt

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

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (activity1)
$ git add --a

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (activity1)
$ git status
On branch activity1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   oops/hello.txt


Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (activity1)
$ git commit -m
error: switch `m' requires a value

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (activity1)
$ git commit -m "new files"
[activity1 cd452b2] new files
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 oops/hello.txt

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/G3 SCM (activity1)
$
```

After this I have done the 3 Step architecture which is tracking the file, send it to stagging
area and finally we can rollback to any previously saved version of this file.

After this we will change the branch from activity1 to master, but when we switch back to master branch the file we created i.e “hello” will not be there. Hence the new file will not be shown in the master branch. In this way we can create and change different branches. We can also merge the branches by using the git merge command.

In this way we can create and change different branches. We can also merge the branches by using git merge command.

Aim: Git lifecycle description

Theory:

Stages in GIT Life Cycle:

Files in a Git project have various stages like Creation, Modification, Refactoring, and Deletion and so on. Irrespective of whether this project is tracked by Git or not, these phases are still prevalent. However, when a project is under Git version control system, they are present in three major Git states in addition to these basic ones. Here are the three Git states:

- Working directory
- Staging area
- Git directory

Working Directory:

Consider a project residing in your local system. This project may or may not be tracked by Git. In either case, this project directory is called your Working directory.

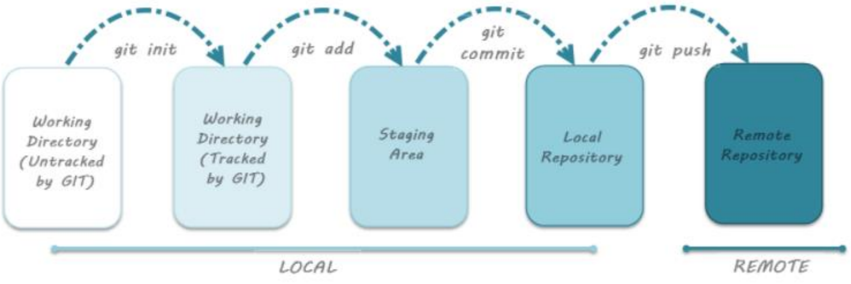
Staging Area:

Staging area is the playground where you group, add and organize the files to be committed to Git for tracking their versions.

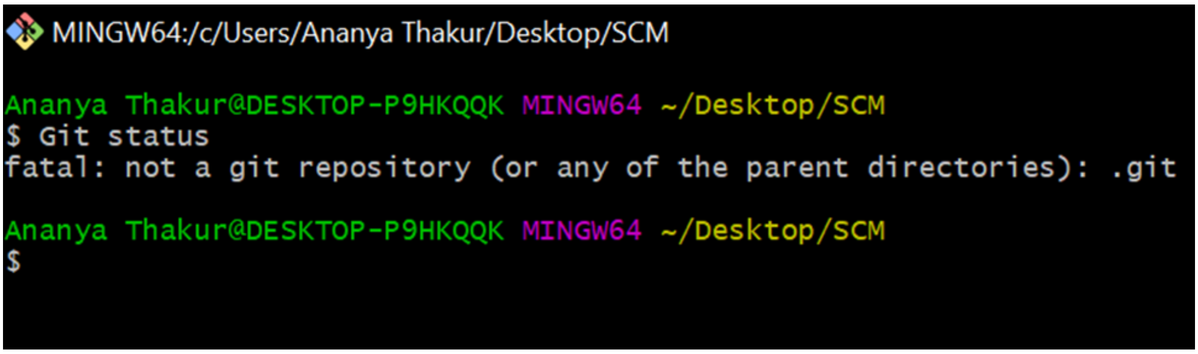
Git Directory:

Now that the files to be committed are grouped and ready in the staging area, we can commit these files. So, we commit this group of files along with a commit message explaining what is the commit about. Apart from commit message, this step also records the author and time of the commit. Now, a snapshot of the files in the commit is recorded by Git. The information related to this commit is stored in the Git directory.

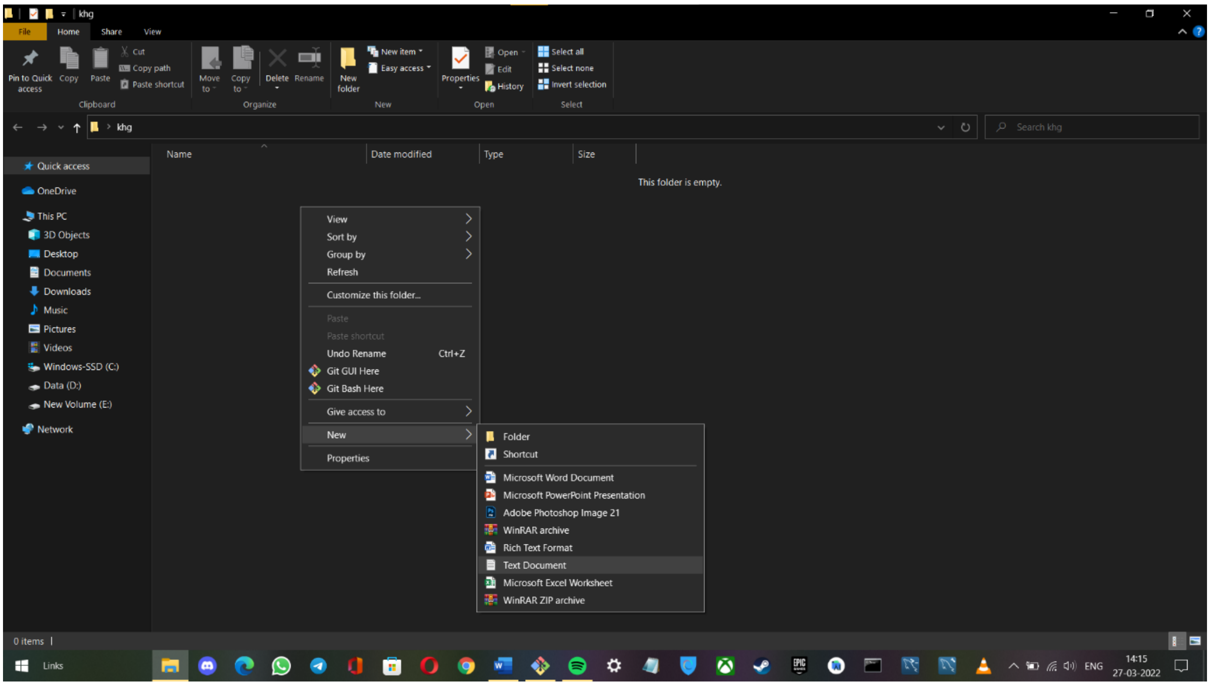
Remote Repository: means mirror or clone of the local Git repository in GitHub. And pushing means uploading the commits from local Git repository to remote repository hosted in GitHub.

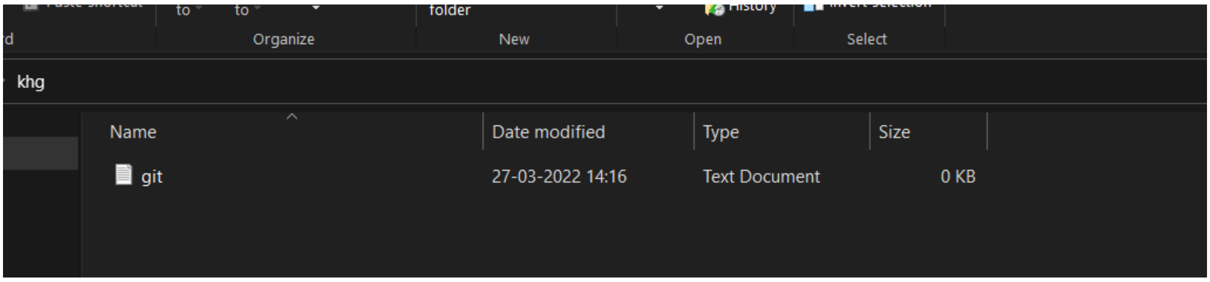


Snapshots:

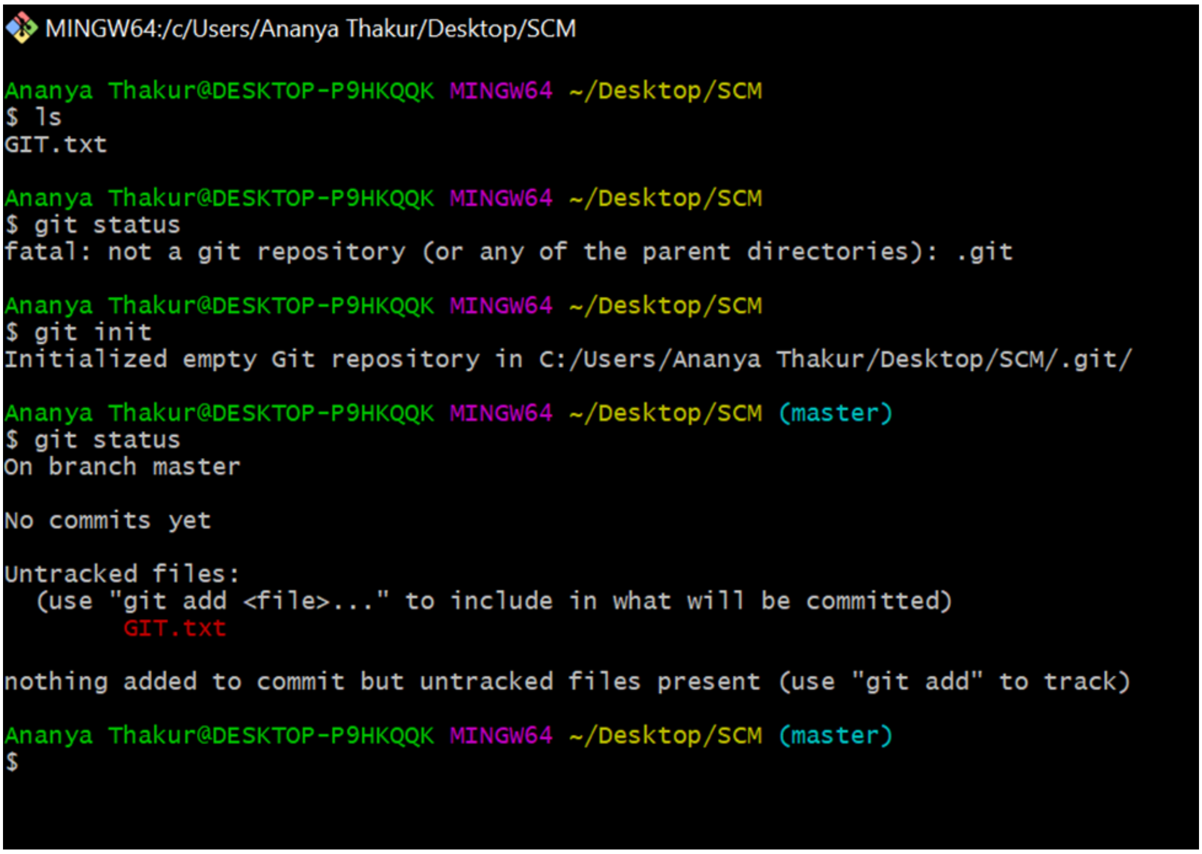


```
MINGW64:/c/Users/Ananya Thakur/Desktop/SCM

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/SCM
$ Git status
fatal: not a git repository (or any of the parent directories): .git

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/SCM
$
```

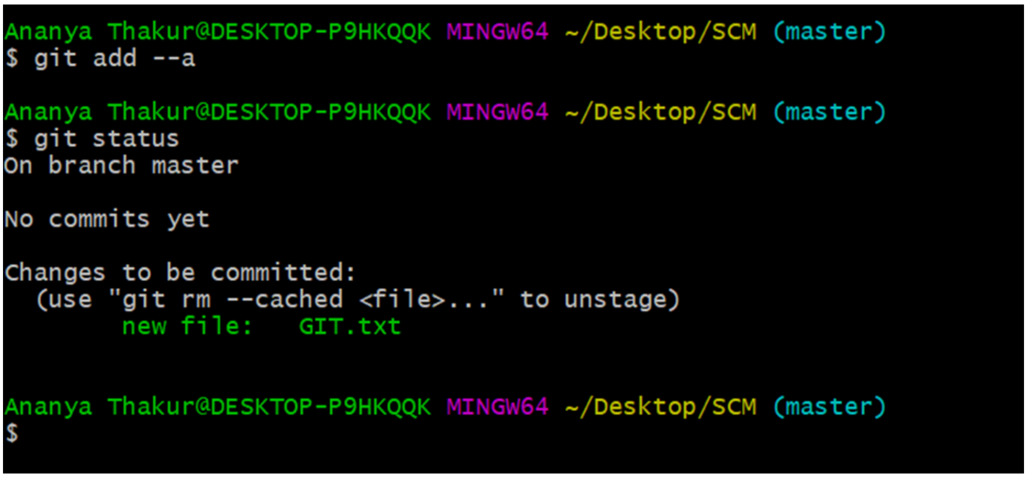




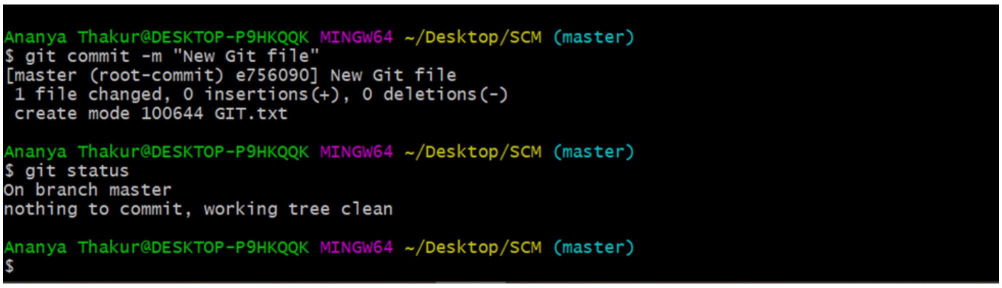
```
Name                      Date modified       Type             Size
git                       27-03-2022 14:16    Text Document    0 KB
```


MINGW64:/c/Users/Ananya Thakur/Desktop/SCM

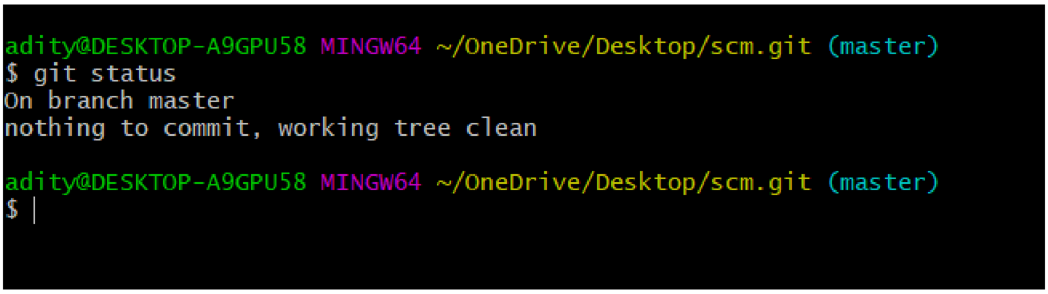
```
Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/SCM
$ ls
GIT.txt

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/SCM
$ git status
fatal: not a git repository (or any of the parent directories): .git

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/SCM
$ git init
Initialized empty Git repository in C:/Users/Ananya Thakur/Desktop/SCM/.git/

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/SCM (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        GIT.txt

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

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/SCM (master)
$
```



```
Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/SCM (master)
$ git add --a

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/SCM (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   GIT.txt


Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/SCM (master)
$
```



```
Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/SCM (master)
$ git commit -m "New Git file"
[master (root-commit) e756090] New Git file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 GIT.txt

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/SCM (master)
$ git status
On branch master
nothing to commit, working tree clean

Ananya Thakur@DESKTOP-P9HKQQK MINGW64 ~/Desktop/SCM (master)
$
```



```
adity@DESKTOP-A9GPU58 MINGW64 ~/OneDrive/Desktop/scm.git (master)
$ git status
On branch master
nothing to commit, working tree clean

adity@DESKTOP-A9GPU58 MINGW64 ~/OneDrive/Desktop/scm.git (master)
$ |
```