CHITKARA UNIVERSITY

SOURCE CODE MANAGEMENT (CS181)

Task 1.1

Submitted by –
Dhruv Jain
2110990443
G8-B

Submitted to –
Dr. Monit Kapoor

INDEX

Sr. No	Topic	Page No.
1.	Introduction	2
2.	Exp – 1 Setting up of Git Client	3-6
3.	Exp – 2 Setting up GitHub Account	7-8
4.	Exp – 3 Generate Log	9-10
5.	Exp – 4 Create and Visualize branches	11-12
6.	Exp – 5 Git life cycle description	13-14

Introduction

What is GIT?

Git is a source code management technology used by DevOps. Git is a piece of software that allows you to track changes in any group of files. It is a free and open-source version control system that may be used to efficiently manage small to big projects.

What is GITHUB?

GitHub is a version management and collaboration tool for programming. It allows you and others to collaborate on projects from any location.

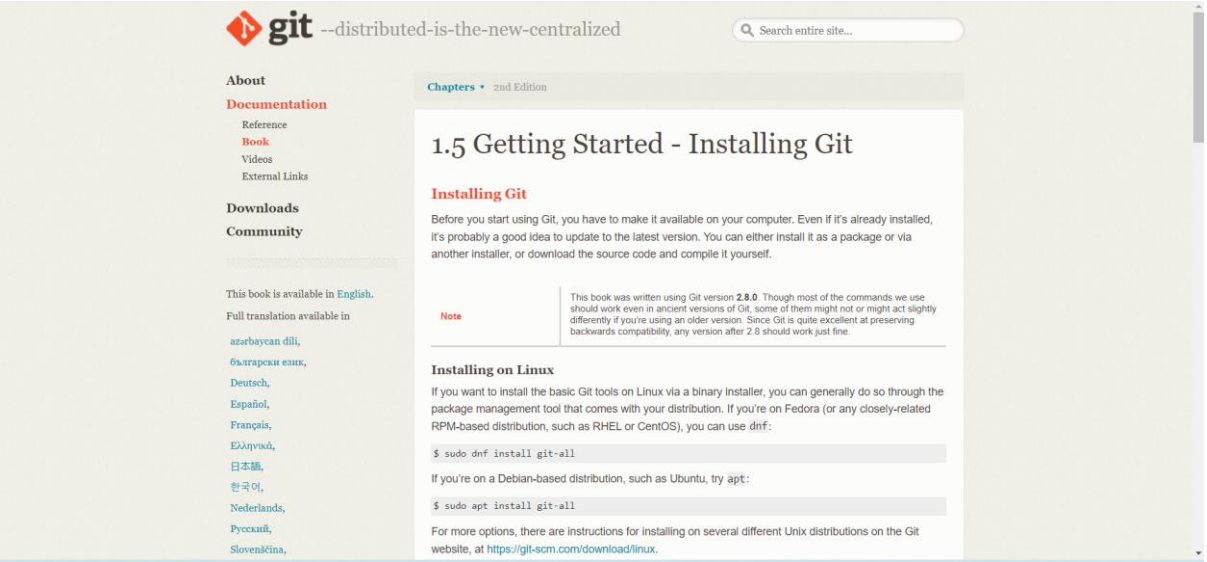
What is Repository?

A repository stores all of your project's files, as well as the revision history for each one. Within the repository, you may discuss and monitor your project's progress.

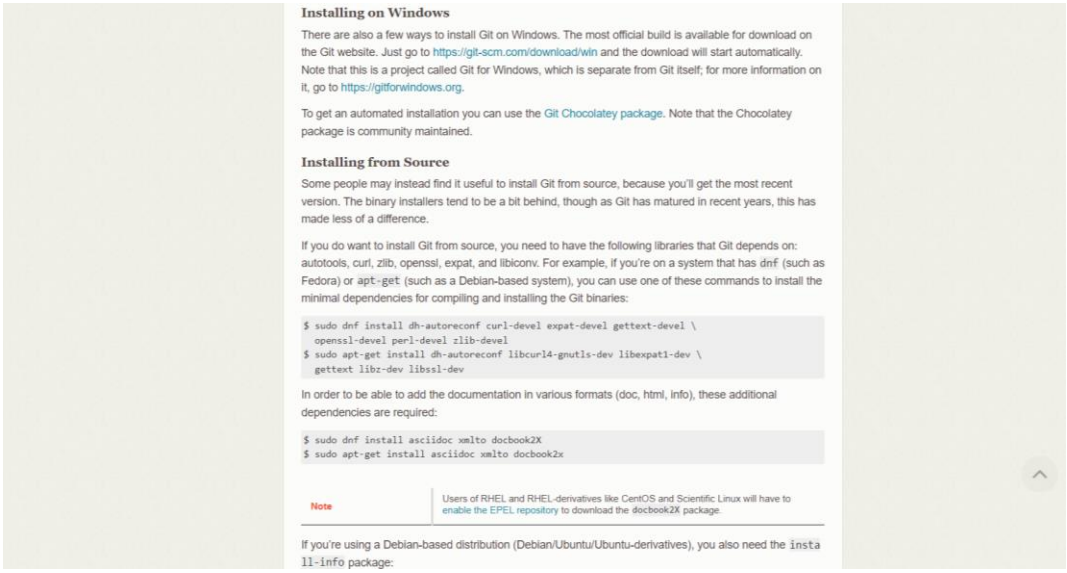
Experiment No. 01

Aim: Setting up of Git Client

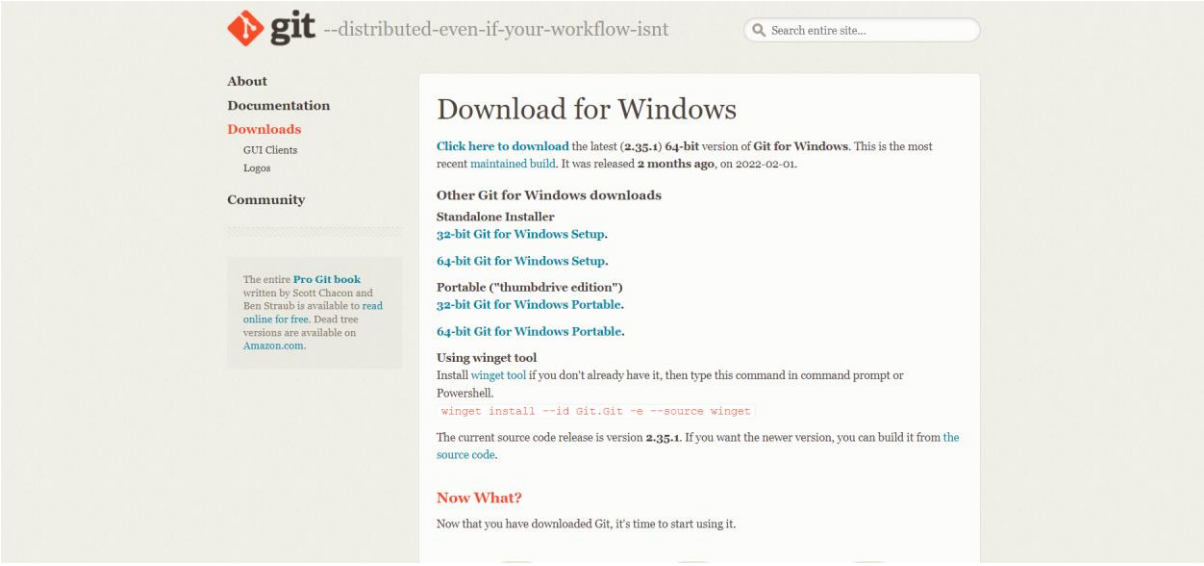
✧ For git installation on your system, go to https://git-scm.com/book/en/v2/Getting-Started-Installing-Git.



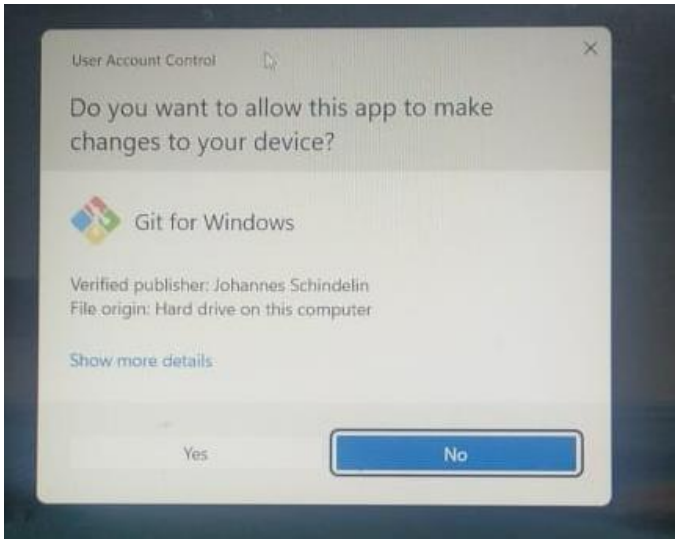
✧ Choose the operating system by clicking on it. I'll choose the Windows operating system.



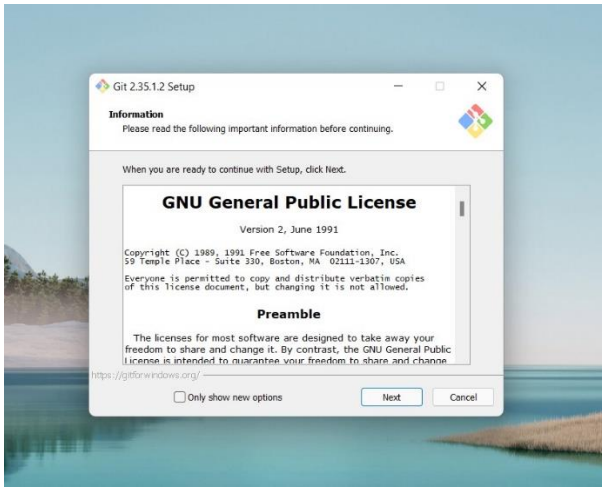
✧ Select the CPU for your system now. (I choose to 64-bit Git for Windows Setup.)



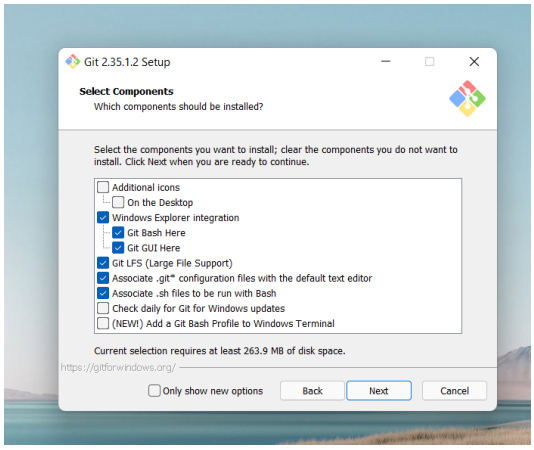
✧ Then, open the Git in Download folder. We will be asked if you want to enable this program to make modifications to your PC once you launch it. Select YES.

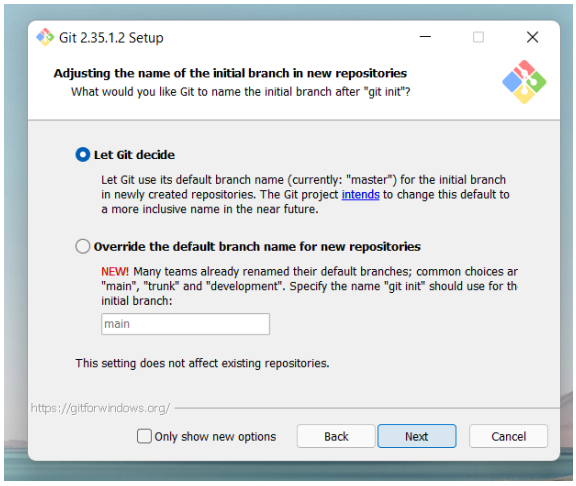


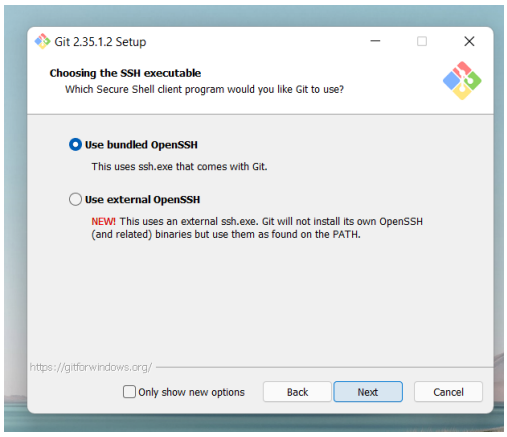
✧ The below window will appear. Click on Next.

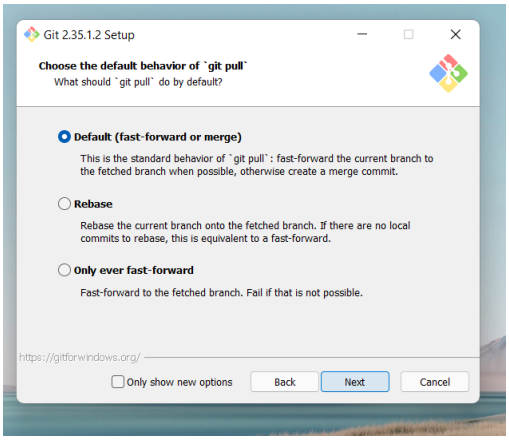


✧ Continue clicking on next few times more.

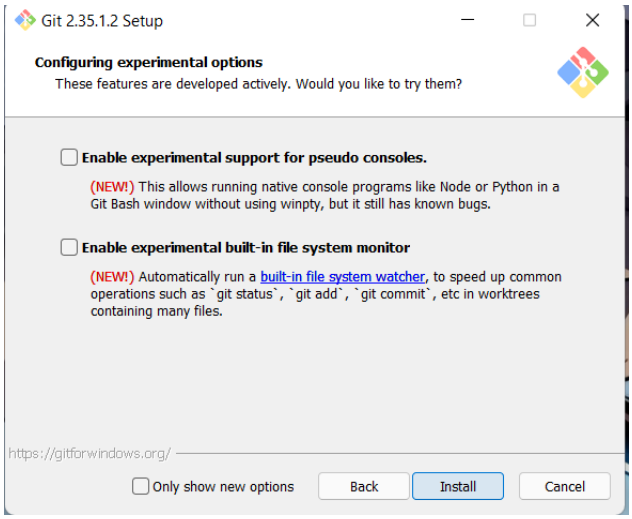


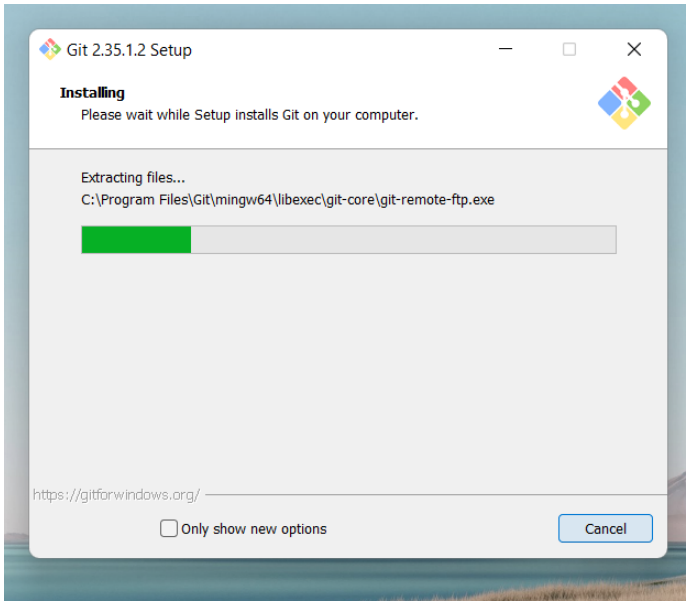




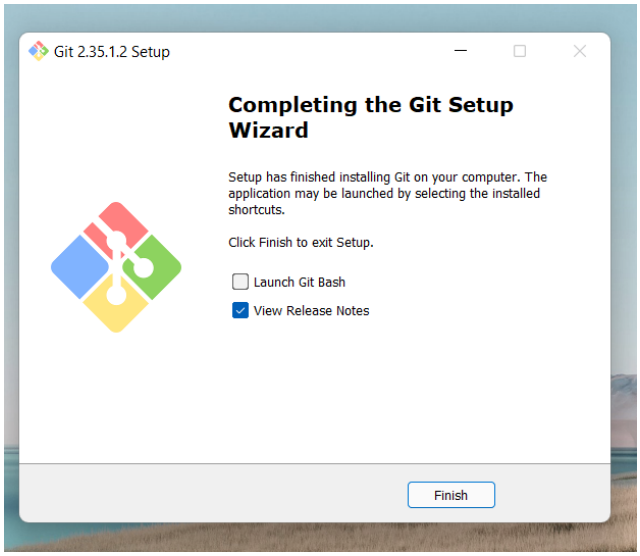


✧ Now select the Install option.





✧ After completion of installing, click on Finish after the installation is finished.



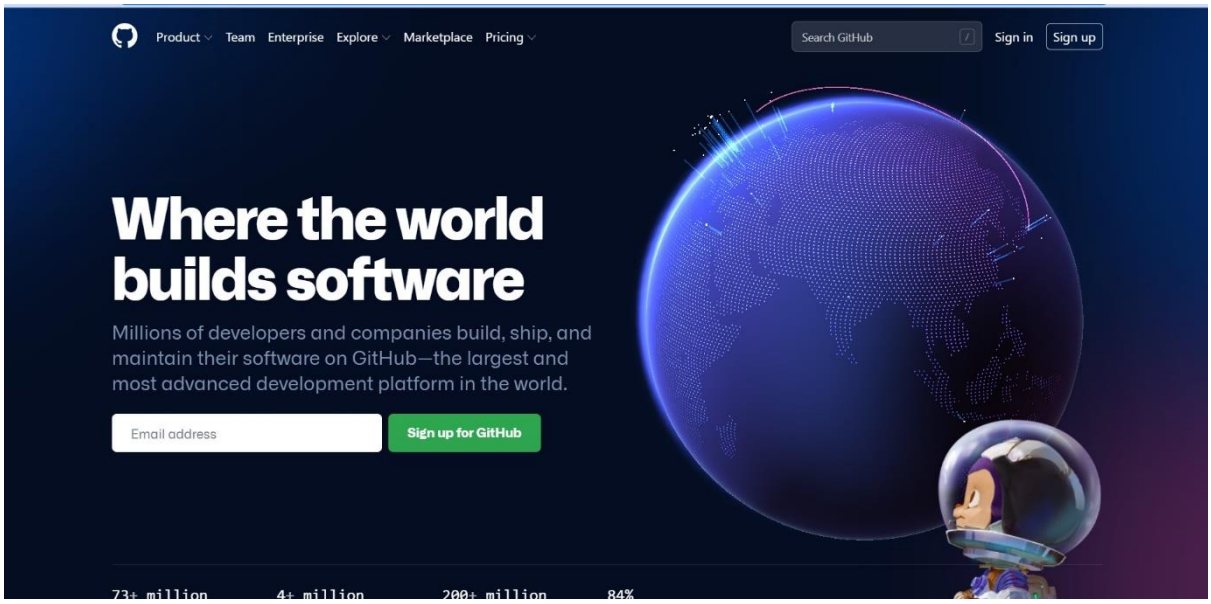
The installation of the git is finished and now we have to setup git client and GitHub account.

Experiment No. 02

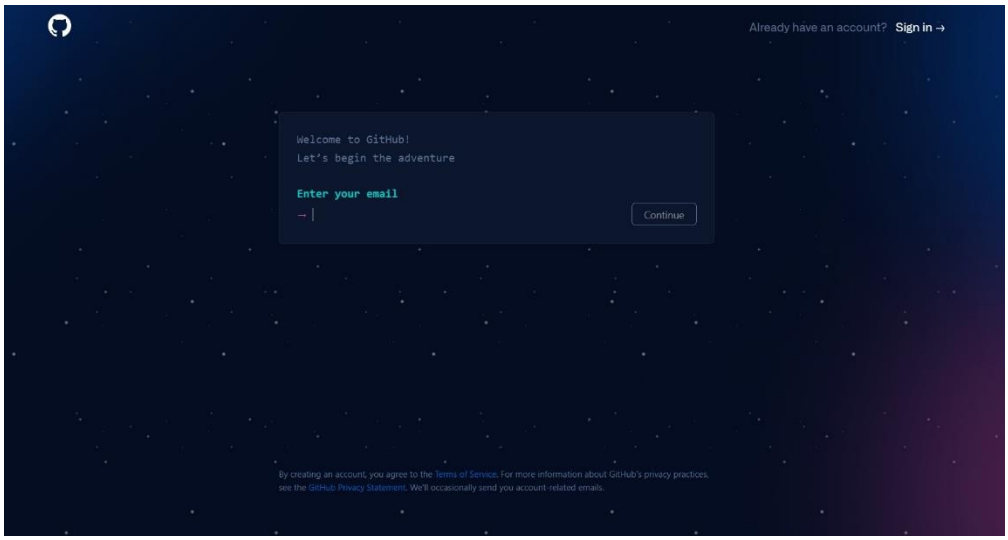
Aim: Setting up GitHub Account

✧ Open your web browser search http://github.com .

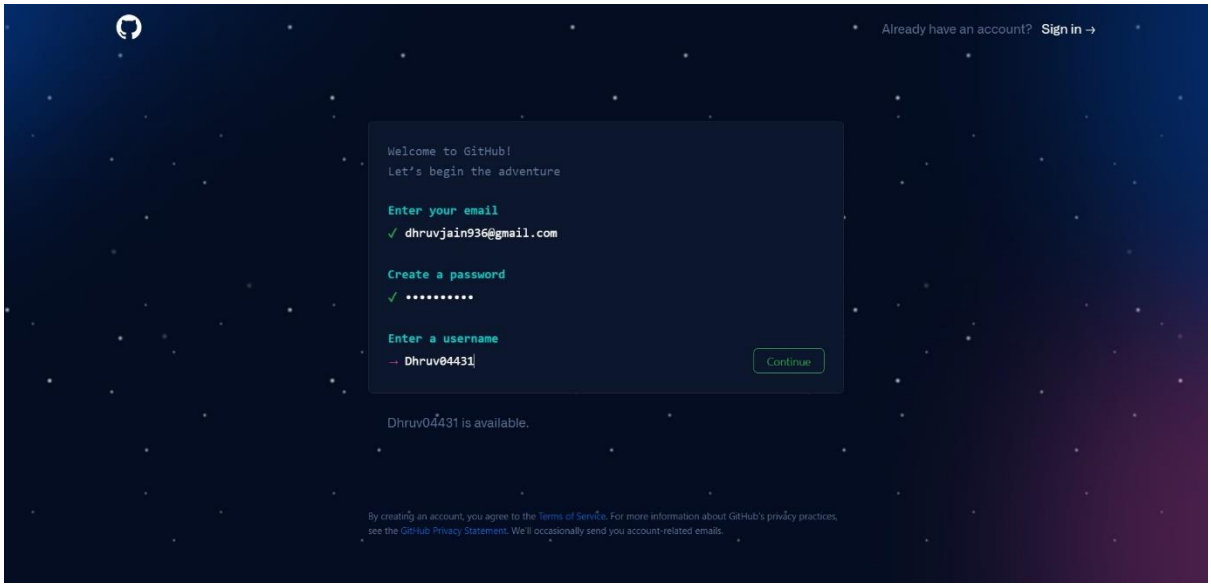
✧ Click on Sign Up on the upper-right conner of desktop window.



✧ After clicking on "Sign Up," a new page will appear where we must enter your email address for your account.

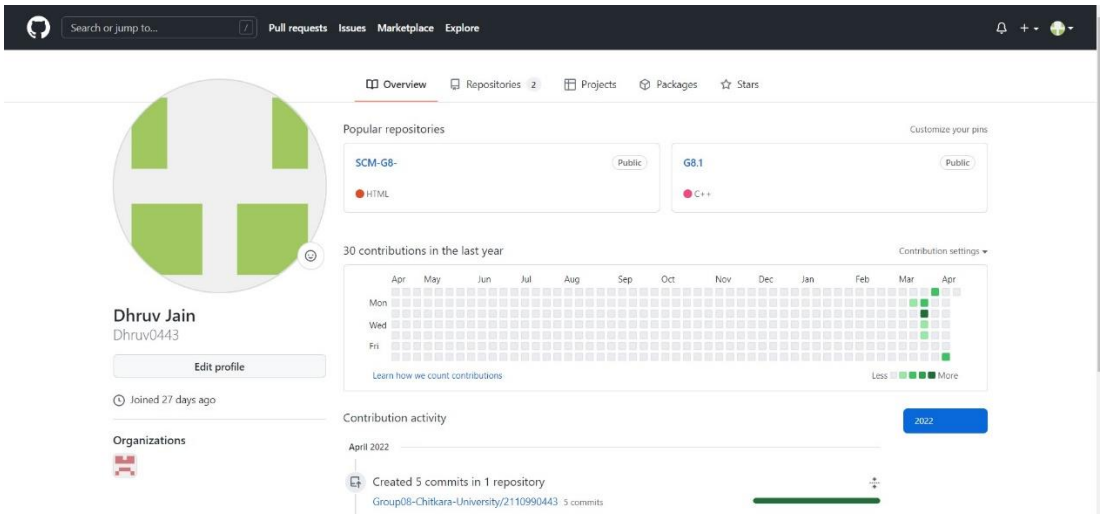


✧ Now type in the password we want to use for your GitHub account. Then you'll be prompted to enter your username.



✧ Now Click on Create Account.

✧ Verify the email and our Github account is ready.



Experiment No. 03

Aim: Program to Generate logs

✧ 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 init**" and it creates a folder **.git**.

```
DELL@DESKTOP-VRD4NDB MINGW64 /d/Dhruv (master)
$ git init
Initialized empty Git repository in D:/Dhruv/.git/

DELL@DESKTOP-VRD4NDB MINGW64 /d/Dhruv (master)
$
```

✧ When we use GIT for the first time, we have to give the user 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*"

Some Important Commands:

git add -A **[** For add all the files in staging area. **]**

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

```
DELL@DESKTOP-VRD4NDB MINGW64 /d/Dhruv (master)
$ vi abc.cpp

DELL@DESKTOP-VRD4NDB MINGW64 /d/Dhruv (master)
$ git add abc.cpp
warning: LF will be replaced by CRLF in Dhruv/abc.cpp.
The file will have its original line endings in your working directory

DELL@DESKTOP-VRD4NDB MINGW64 /d/Dhruv (master)
$ git commit -m"Changing Output"
[master ea7cef1] Changing Output
 1 file changed, 1 insertion(+), 1 deletion(-)
```

✧ **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.

```
DELL@DESKTOP-VRD4NDB MINGW64 /d (master)
$ git log
commit 67697cd9caecf8ce2f7ec5e1d797d1924d8400d8 (HEAD -> master, file/master)
Author: Dhruv0443 <dhruv0443.be21@chitkara.edu.in>
Date:   Sat Apr 9 22:50:38 2022 +0530

    REMOVING Conditions from loops

commit 748eee1f51f067252168888389fd9a8703eae4f9 (htmlfile/master)
Author: Dhruv0443 <dhruv0443.be21@chitkara.edu.in>
Date:   Sat Apr 9 22:45:27 2022 +0530

    Changing background colour

commit d64708be2be5a7f88961fc99ca0200410268d348
Author: Dhruv0443 <dhruv0443.be21@chitkara.edu.in>
Date:   Sat Apr 9 22:41:49 2022 +0530

    FLex Making in html
```

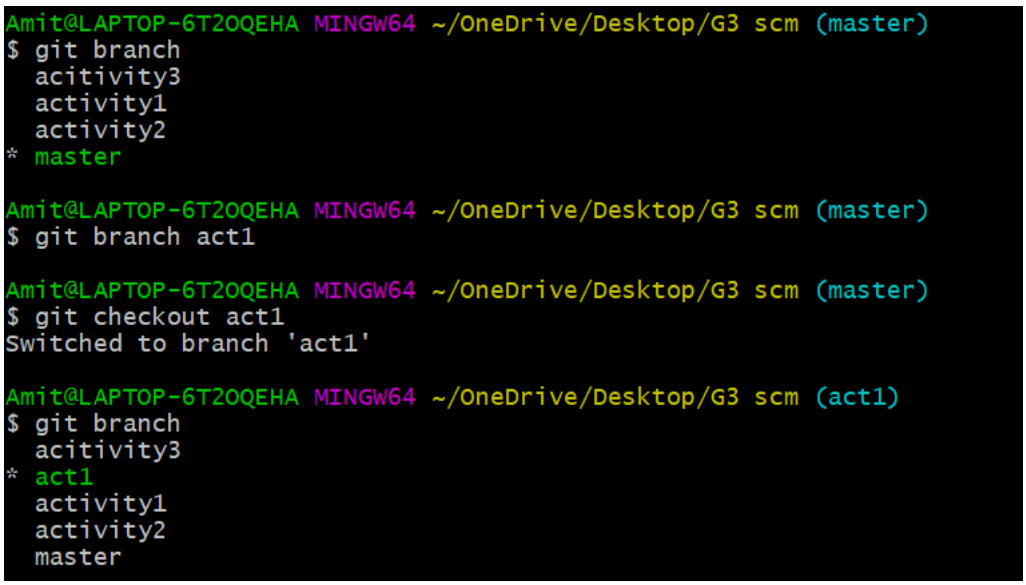
Experiment No. 04

Aim: Create and visualize branches

✧ **Branching:** 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**]**



In this you can see that firstly 'git branch' shows only one branch in green colour but when we add a new branch using 'git branch act1', it shows 2 branches but the green colour and star is on master. So, we have to switch to act1 by using 'git checkout act1'. If we use 'git branch', now you can see that the green colour and star is on act1. It means you are in activity1 branch and all the data of master branch is also on act1 branch. Use "ls" to see the files.

Now add a new file in activity1 branch, do some changes in file and commit the file.

```
Amit@LAPTOP-6T2OQEHA MINGW64 ~/OneDrive/Desktop/G3 scm (master)
$ touch contact.html

Amit@LAPTOP-6T2OQEHA MINGW64 ~/OneDrive/Desktop/G3 scm (activity1)
$ git add -A

Amit@LAPTOP-6T2OQEHA MINGW64 ~/OneDrive/Desktop/G3 scm (activity1)
$ git commit -m "committing contact.html"
[activity1 af7b1d9] committing contact.html
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 contact.html

Amit@LAPTOP-6T2OQEHA MINGW64 ~/OneDrive/Desktop/G3 scm (activity1)
$ ls
'Activity 1 file.txt'   contact.html  'wallpaperflare.com_wallpaper (1).jpg'
 comments.txt           index.txt     'wallpaperflare.com_wallpaper (2).jpg'

Amit@LAPTOP-6T2OQEHA MINGW64 ~/OneDrive/Desktop/G3 scm (activity1)
$
```

If we switched to master branch, ‘contact.html’ file is not there. But he file is in activity1 branch.

```
Amit@LAPTOP-6T2OQEHA MINGW64 ~/OneDrive/Desktop/G3 scm (activity1)
$ git checkout master
Switched to branch 'master'

Amit@LAPTOP-6T2OQEHA MINGW64 ~/OneDrive/Desktop/G3 scm (master)
$ ls
 OOPScalcu.py                filewrite.py    randomGame.py        'wallpaperflare.com_wallpaper (1).jpg'
 Playsound.py                findinfile.py   sample.txt           'wallpaperflare.com_wallpaper (2).jpg'
 comments.txt                index.txt       song.mp3             'wallpaperflare.com_wallpaper (4).jpg'
'factorial TrailingZero.py'  os.py           temp1.txt            'wallpaperflare.com_wallpaper (5).jpg'
 file.py                     pratice.py      try_except_final.py  'wallpaperflare.com_wallpaper (7)g.jpg'

Amit@LAPTOP-6T2OQEHA MINGW64 ~/OneDrive/Desktop/G3 scm (master)
$
```

- To add these files in master branch, we have to do merging. For this firstly switch to master branch and then use command →

  git merge branchname [use to merge branch**]**

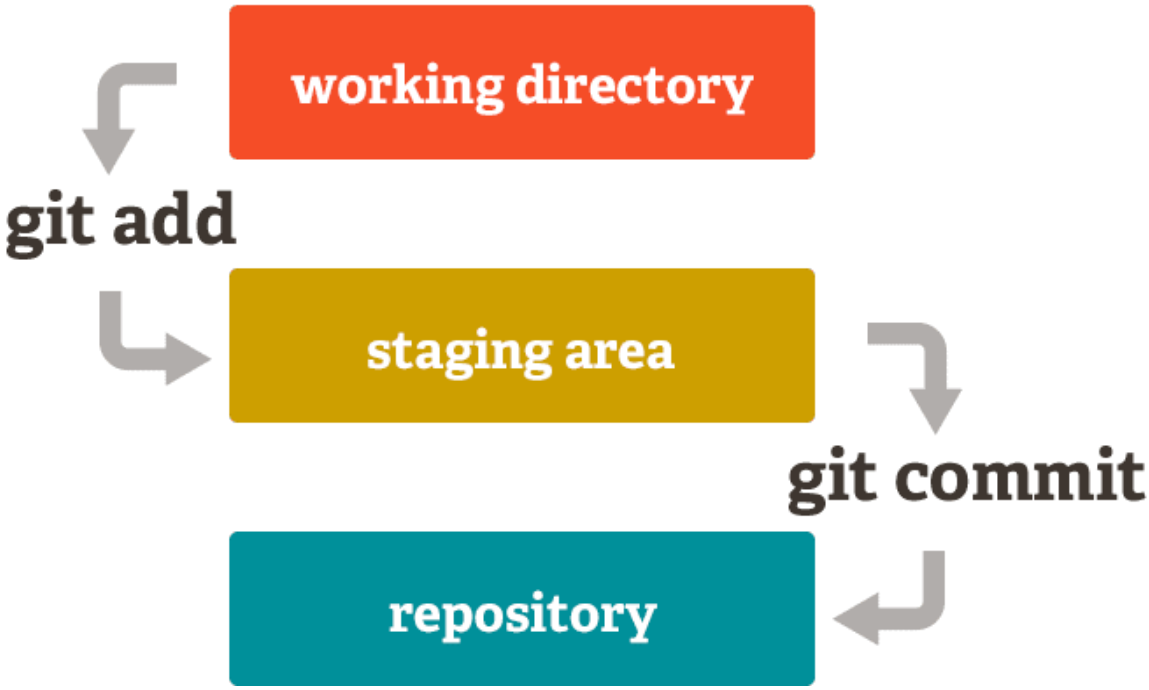
```
Amit@LAPTOP-6T2OQEHA MINGW64 ~/OneDrive/Desktop/G3 scm (master)
$ git log
commit 87ba6dc52876a0cda3d9397ec902c91e42dd79e0 (HEAD -> master)
Merge: 898faf3 af7b1d9
Author: XxFiEnDxX <amitkumargdgps@gmail.com>
Date:   Tue Mar 29 01:09:42 2022 +0530

    Merge branch 'activity1'
```

Aim: Git lifecycle description

Stages in GIT Life Cycle:

When a directory is made a git repository, there are mainly 3 states which make the essence of Git Version Control System. The three states are –

- Working Directory
- Staging Area
- Git Directory



- **Working Directory:** Whenever we want to initialize our local project directory to make it a git repository, we use the ***git init*** command. After this command, git becomes aware of the files in the project although it doesn't track the files yet. The files are further tracked in the staging area.

```
git init
```

- **Staging Area:** Now, to track the different versions of our files we use the command ***git add***. We can term a staging area as a place where

different versions of our files are stored. ***git add*** command copies the version of your file from your working directory to the staging area.

```
// to specify which file to add to the staging area
git add <filename>
// to add all files of the working directory to the staging area
git add .
```

- **Git directory(repository):** Now since we have all the files that are to be tracked and are ready in the staging area, we are ready to commit our files using the ***git commit*** command. Commit helps us in keeping the track of the metadata of the files in our staging area. We specify every commit with a message which tells what the commit is about.

```
git commit -m <Commit Message>
```