**SOURCE CODE MANAGEMENT**

NAME:-ABHIJAY SHARMA

GROUP:-G01

ROLL NO:-2110990039





**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | EXPERIMENT NAME |  |
| 1. | **INTRODUCTION** |  |
| 2. | **SETTING UP OF GIT CLIENT** |  |
| 3. | **SETTING UP GITHUB ACCOUNT** |  |
| 4. | **USING GIT COMMANDS AND GENERATING LOGS** |  |
| 5. | **GIT BRANCHING** |  |
| 6. | **GIT LIFECYCLE DESCRIPTION** |  |

****

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.

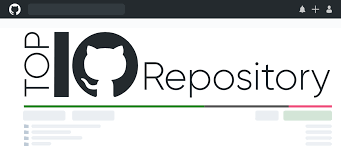
Git is a version control system that allows numerous developers to collaborate on non-linear development projects.

Git is an example of a distributed version control system (DVCS) (hence Distributed Version Control System).

**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.

GitHub is **an online software development platform used for storing, tracking, and collaborating on software projects**. It enables developers to upload their own code files and to collaborate with fellow developers on open-source projects.



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. The .git/ subdirectory within a project is a Git repository. This repository keeps track of any changes made to files in your project over time, creating a history. That is, if you delete the.git/ subdirectory, you are also deleting the history of your project.

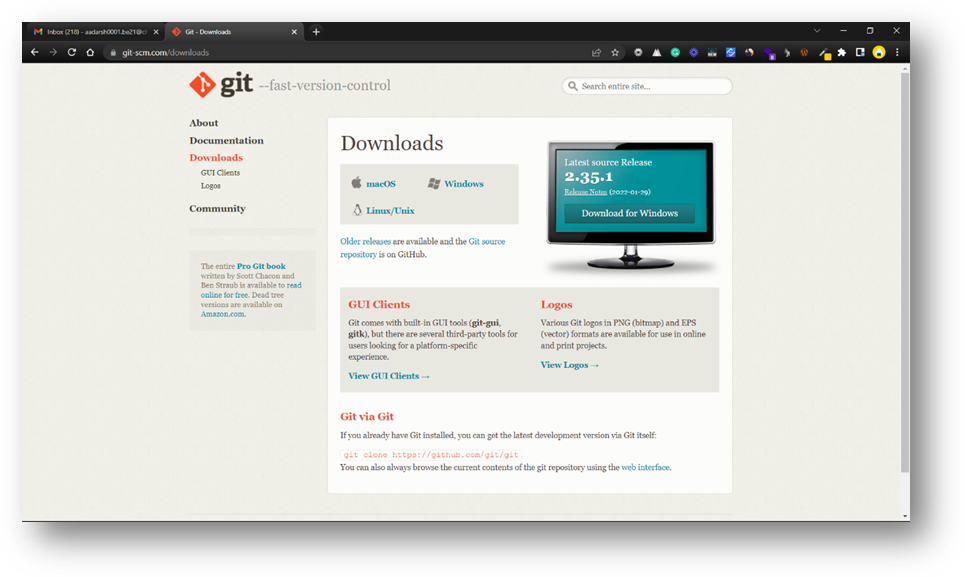


Version Control Systems are the software tools for tracking/managing all the changes made to the source code during the project development. It keeps a record of every single change made to the code. It also allows us to turn back to the previous version of the code if any mistake is made in the current version. Without a VCS in place, it would not be possible to monitor the development of the project.

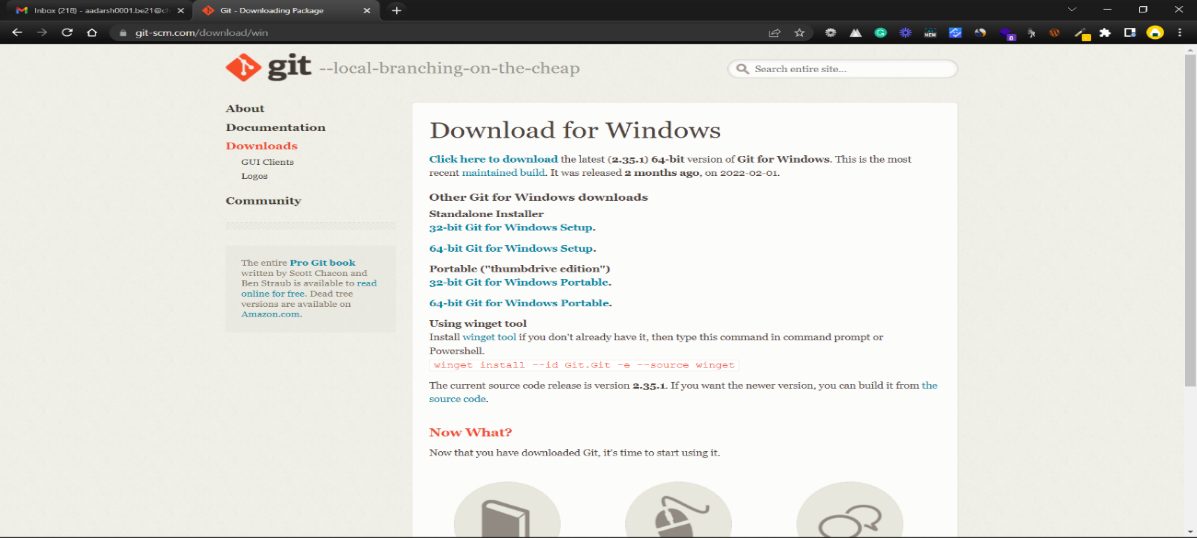
SETTING UP OF GIT CLIENT

For git installation on your system, go to the linked URL.

<https://git-scm.com/downloads>



You must first access this webpage and then choose your operating system by clicking on it. I'll walk you through the processes for the Windows operating system in this article.



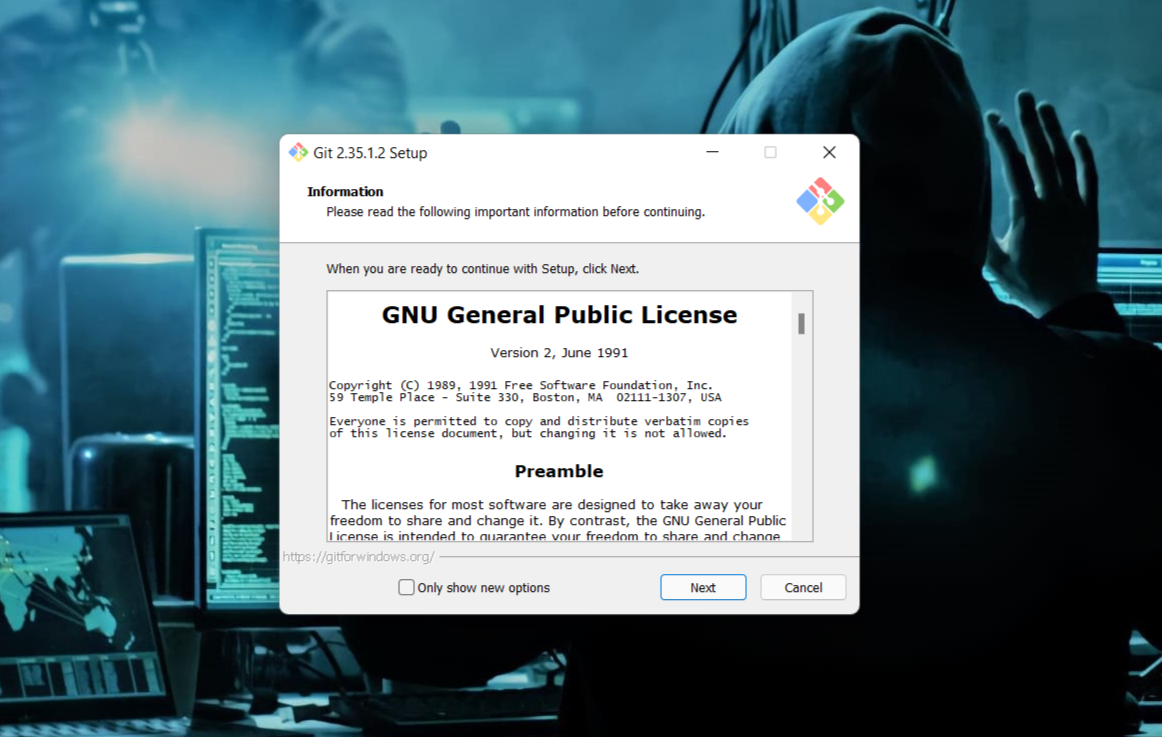
Select the CPU for your system now. (Most of the system now runs on 64-bit processors.) Your download will begin when you pick a processor.



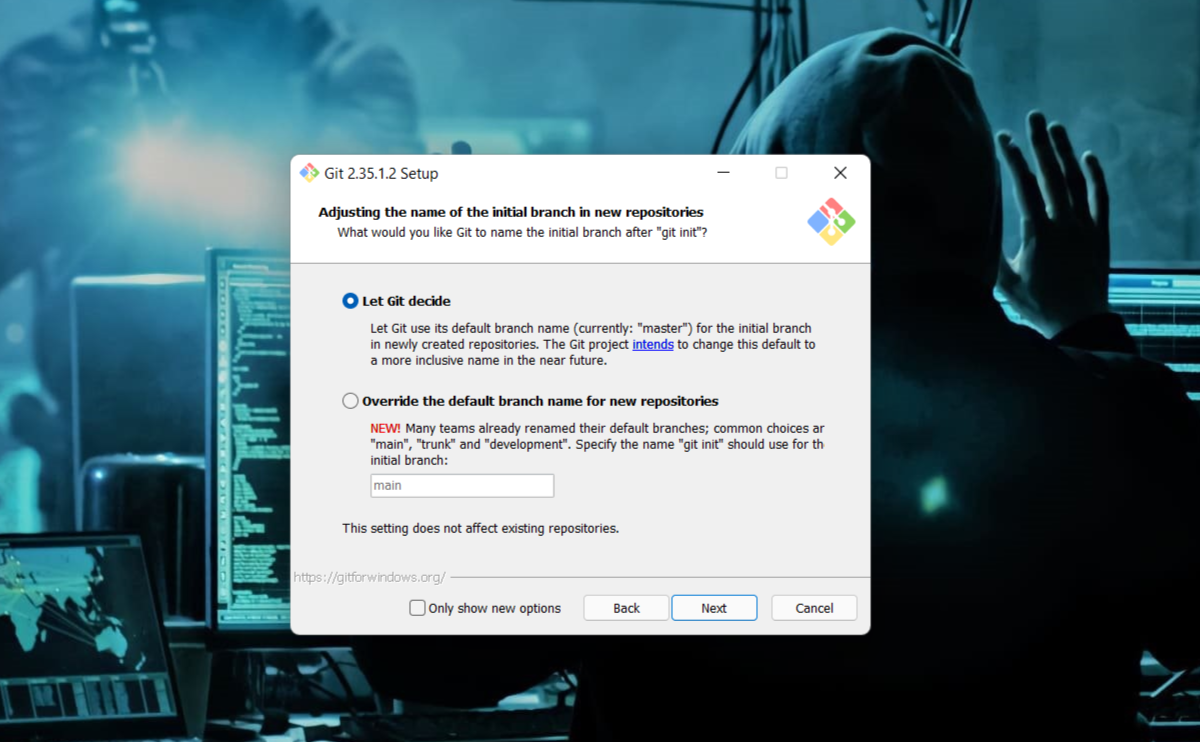
You must now open the Git folder.

You will be asked if you want to enable this program to make modifications to your PC once you launch it.

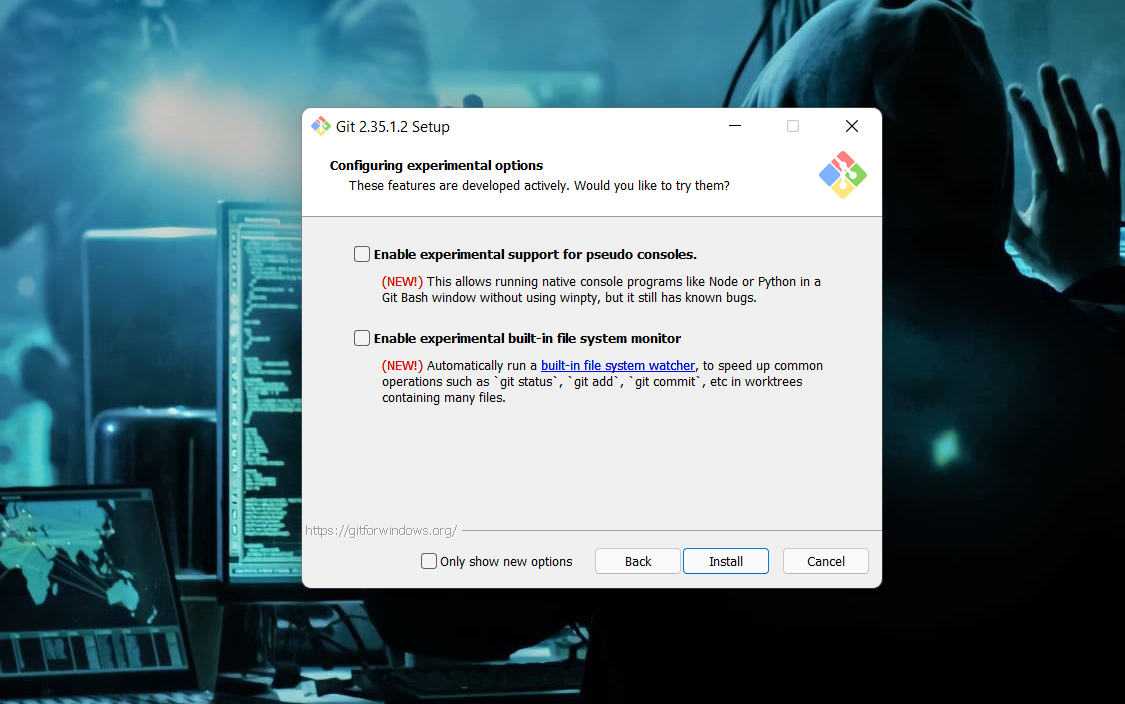
YES should be selected.



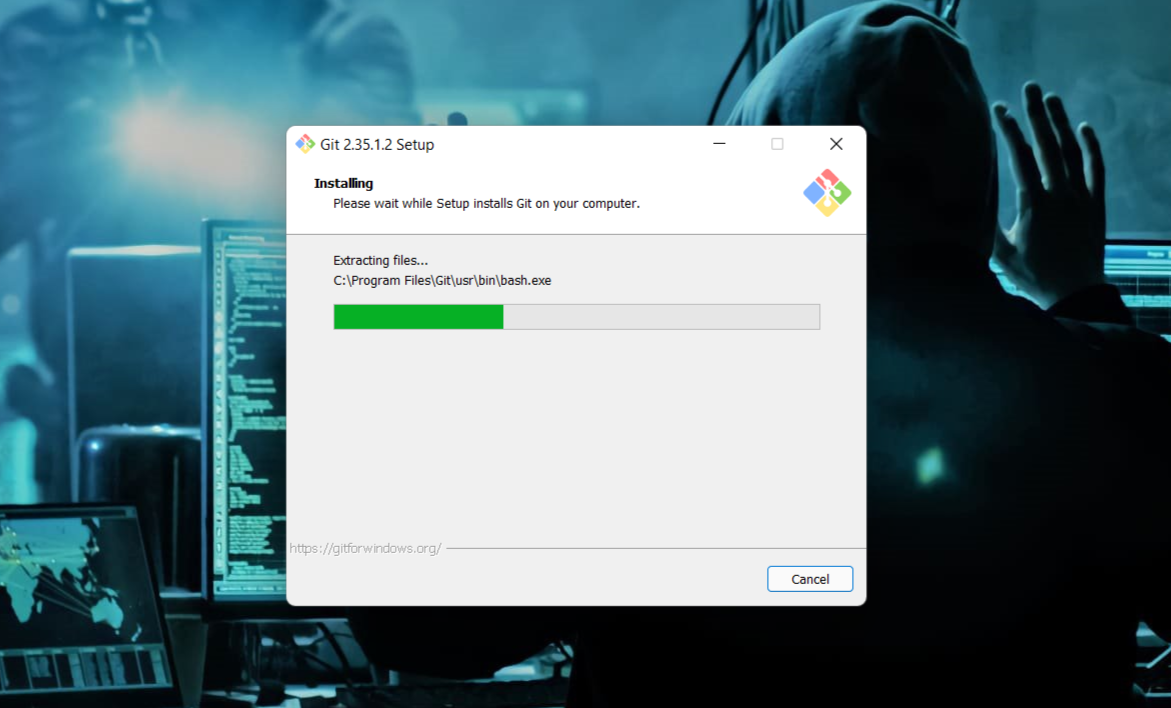
Click on Next



Continue clicking on next few times more



Now select the Install option.



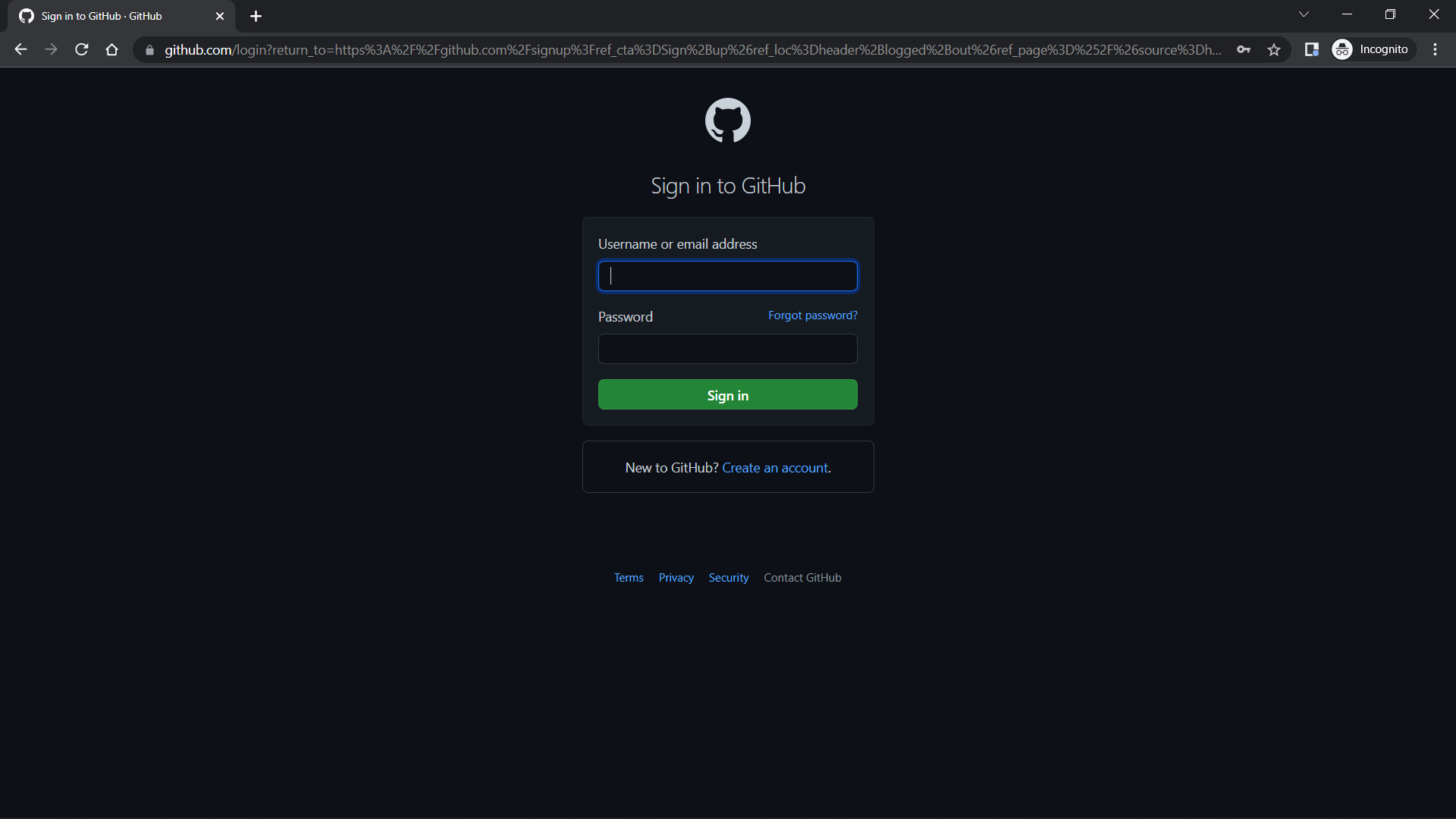
Click on Finish after the installation is finished.

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

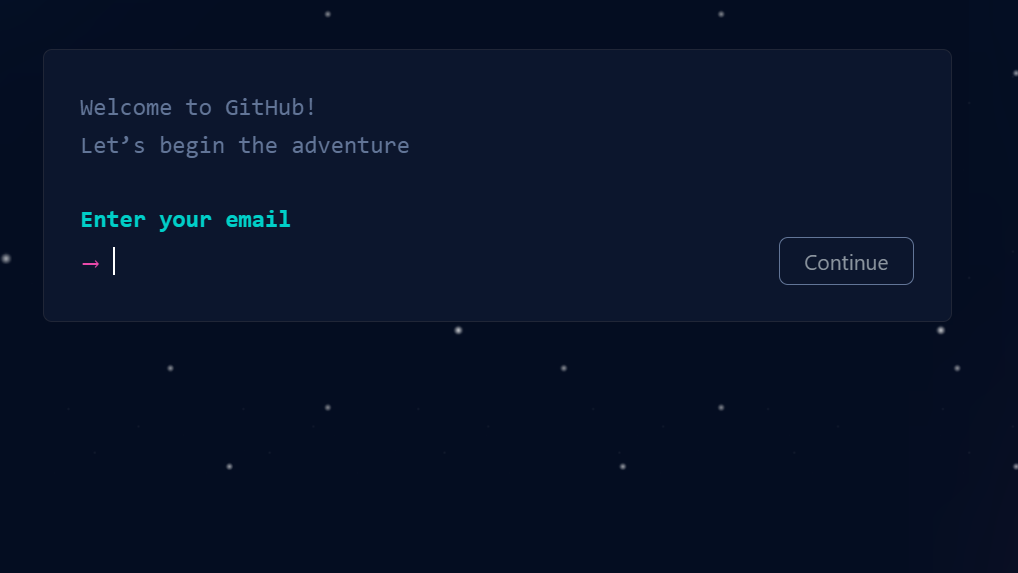
SETTING UP GITHUB ACCOUNT

Open your web browser search GitHub login.

Click on Create an account if you are a new user or if you have already an account, please login.



After clicking on "Create a New Account," you will be sent to a new page where you must enter your email address for your account. Now type in the password you'd want to use for your GitHub account. Then you'll be prompted to enter your username.



Now Click on Create Account.

Verify it from your email and you are all set to go.

USING GIT COMMANDS

### SETTING USER-NAME AND EMAIL



Mkdir: It is used for creating a directory using Git Bash.

cd command: Both Bash and Windows console host have a cd command. cd is an acronym for 'Change Directory'. cd is invoked with an appended directory name. Executing cd will change the terminal sessions current working directory to the passed directory argument.

Pwd: The Bash command pwd is used to print the 'present working directory'. This is the folder or path that the current Bash session resides in.

ls: The Bash command ls is used to 'list' contents of the current working directory.

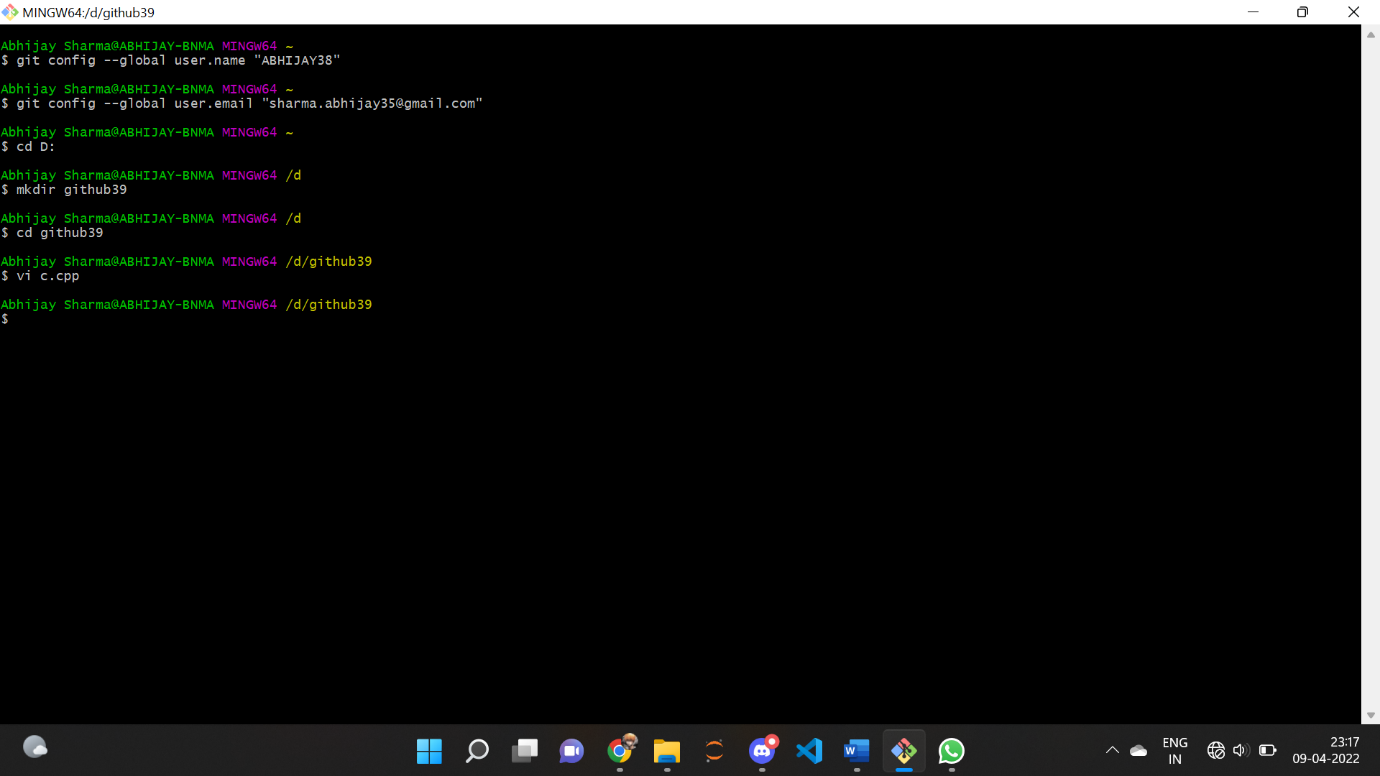
Git status: The main tool you use to determine which files are in which state is the git status command. the command tells you which branch you’re on and informs you that it has not diverged from the same branch on the server.

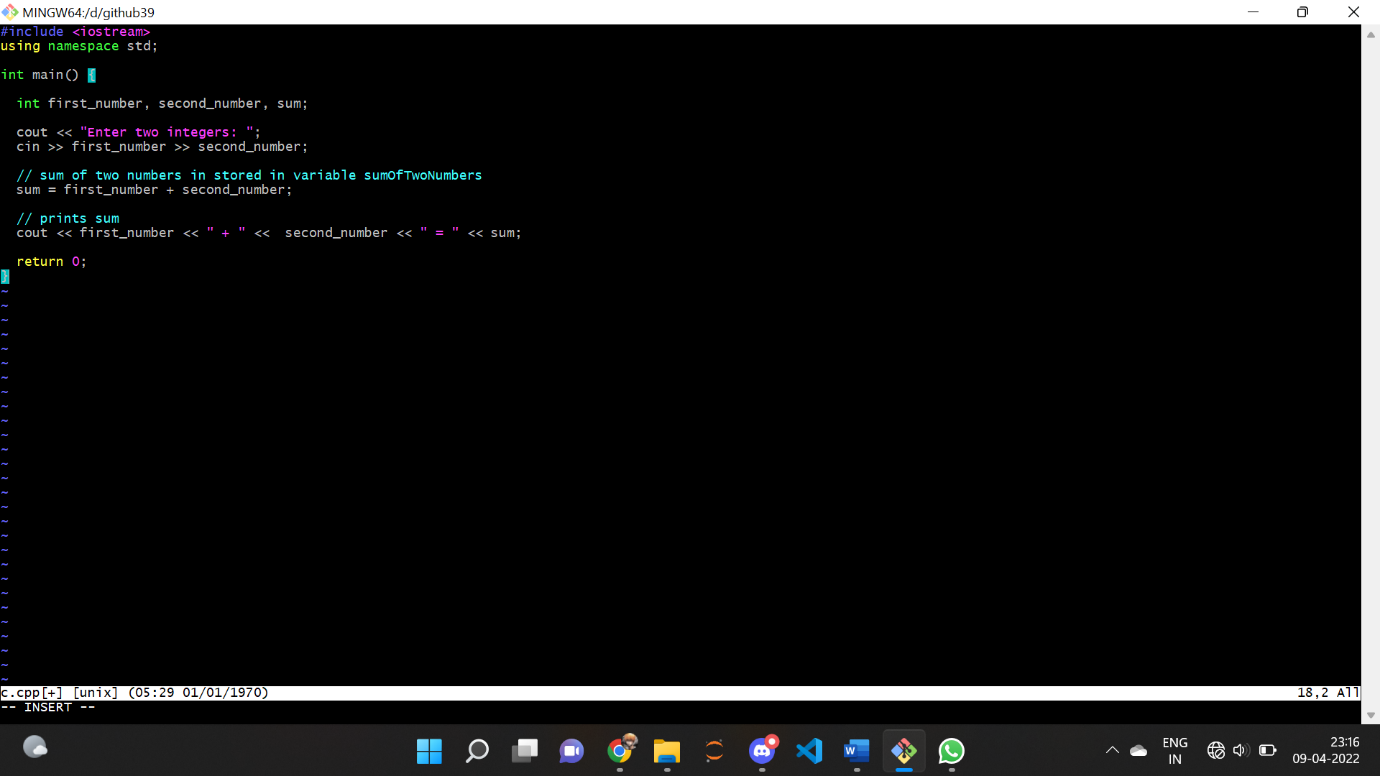
Git add: In order to begin tracking a new file, you use the command git add. git add is a multipurpose command — you use it to begin tracking new files, to stage files, and to do other things like marking merge-conflicted files as resolved. It may be helpful to think of it more as “add precisely this content to the next commit” rather than “add this file to the project”.

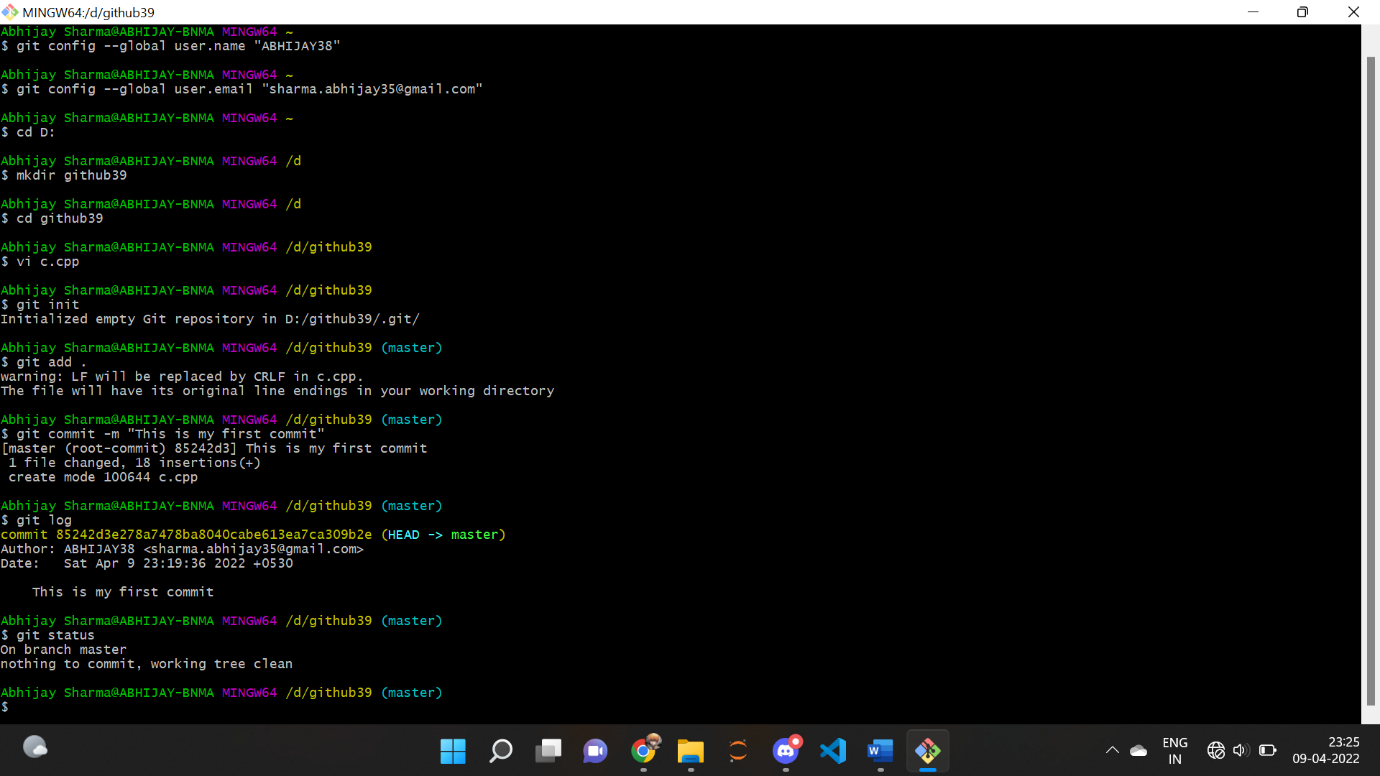
Git commit: The git commit command captures a snapshot of the project's currently staged changes. Committed snapshots can be thought of as “safe” versions of a project—Git will never change them unless you explicitly ask it to. Prior to the execution of git commit, The git add command is used to promote or 'stage' changes to the project that will be stored in a commit. These two commands git commit and git add are two of the most frequently used.

Git push: When you have your project at a point that you want to share, you have to push it upstream. The command for this is simple: git push . If you want to push your master branch to your origin server (again, cloning generally sets up both of those names for you automatically), then you can run this to push any commits you’ve done back up to the server

git push origin master

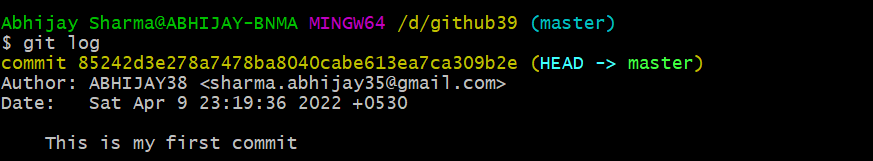






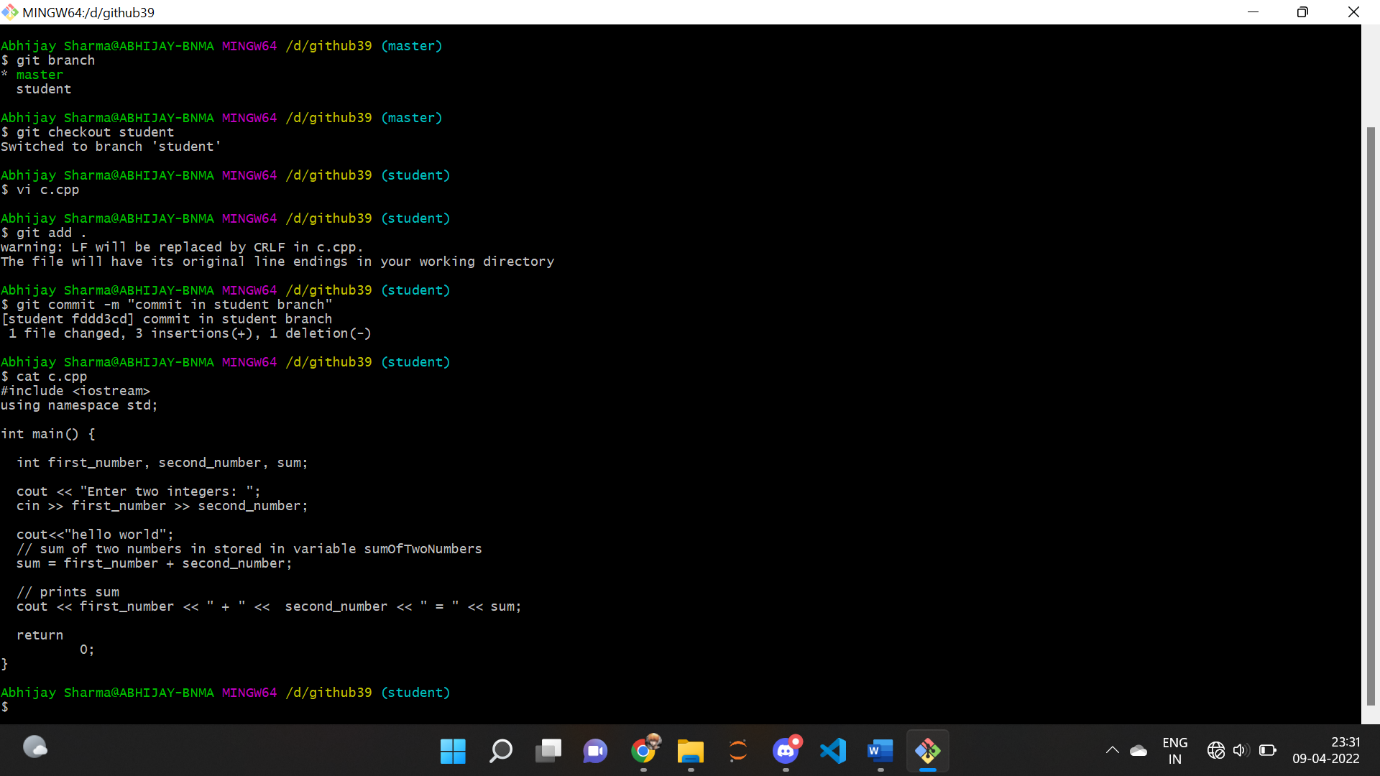
GENERATE LOGS

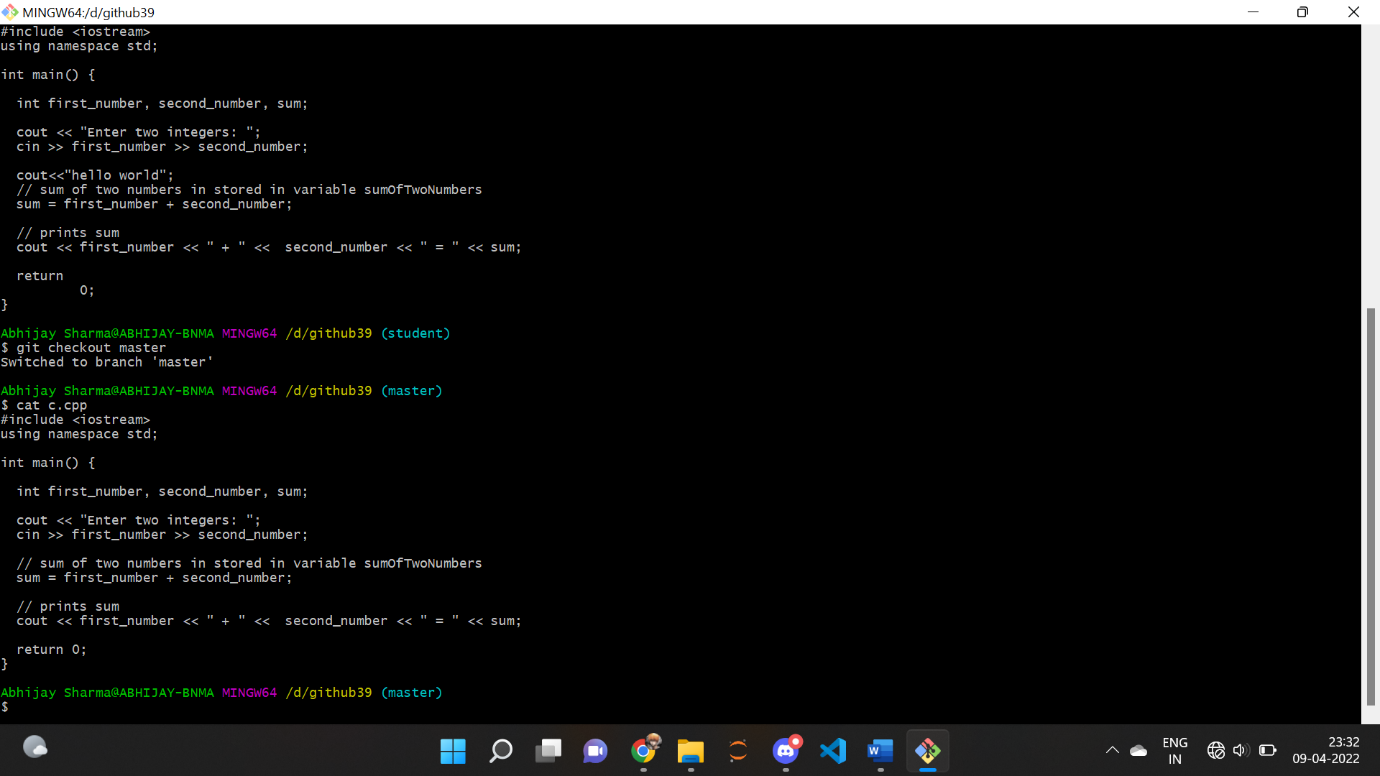
Logs are nothing but the history which we can see in Git by using the code Git log. It contains all the past commits, insertions and deletions which can be seen anytime



GIT 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.



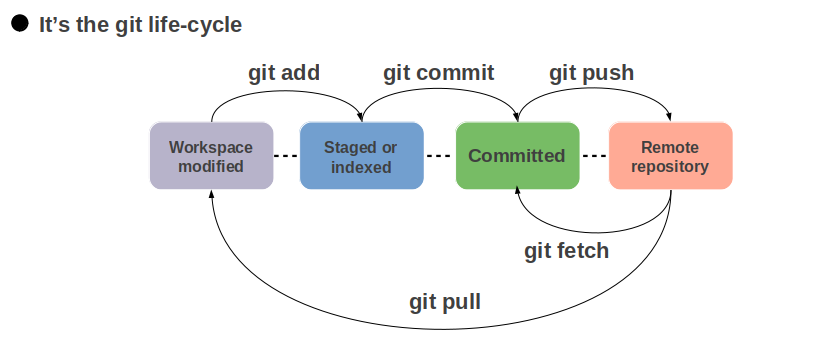


GIT LIFECYCLE DESCRIPTION

* **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 File-Name” 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

**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. 

THANK YOU