

## SEPM Lab- Experiment 2

**Aim:** To understand Version Control System/ Source Code Management, Install Git and create a GitHub Account

### Theory:

#### Version Control System:

Version Control is a method to record changes to a file or set of files over time so that you can recall specific versions later. In the case of software, you will use your software source code as the files being version controlled, though, in reality, you can do this with nearly any type of file on a computer.

The systems used to track changes and different code versions are called Version Control Systems (VCS). Just like with anything else, each VCS has its unique features and comes with its own set of advantages and disadvantages. There are some basics, though, that make a VCS what it is.

The version control systems keep a record of the changes happening on a project over a long duration of time. This can include creation, deletion and even modifications made. They also have provisions for merging and branching. Separate branches have to be made in order to ensure that the “Main” Branch remains error-free. If there’s any error or issue during the project, we can always switch back to the last working version.

#### Types of Version Control Systems:

- Local VCS
- Centralised VCS
- Distributed VCS

#### Importance of Version Control in DevOps:

1. **Avoiding Dependency issues in modern applications:** Dependency issues are common in applications across multiple languages and have been since early programming days. Version Control plays a huge role here. Access to the different code iterations can reveal where new changes expose dependencies that clash with other parts of the code.
2. **Providing better performance:** VCS can view and understand how changes to one part of the code cause problems across the application. It enables coding practices like Continuous Integration and, as a result, Continuous Delivery/Deployment.
3. **Improving Application Reliability:** VCS can handle smaller changes more frequently. It can also track those changes so that everyone is looking at the same thing and so that troubleshooting is easier in the case of failures.

**Source Code Management:** Source code management (SCM) is used to track modifications to a source code repository. SCM tracks a running history of changes to a code base and helps resolve conflicts when merging updates from multiple contributors. SCM is also synonymous with Version control.

As software projects grow in lines of code and contributor head count, the costs of communication overhead and management complexity also grow. SCM is a critical tool to alleviate the organizational strain of growing development costs.

**Git:** Git is a distributed version control system (DVCS) for tracking changes to files. But what does that mean? Git is an open-source VCS, which is not file-based, unlike other systems. Rather, it stores information as snapshots.

Being a VCS, helps coders revert to their previous code when they hit a roadblock in the newer version, without affecting the source code. On the other hand, what makes it different from other VCS is the way it sees data, which is more like a series of snapshots.

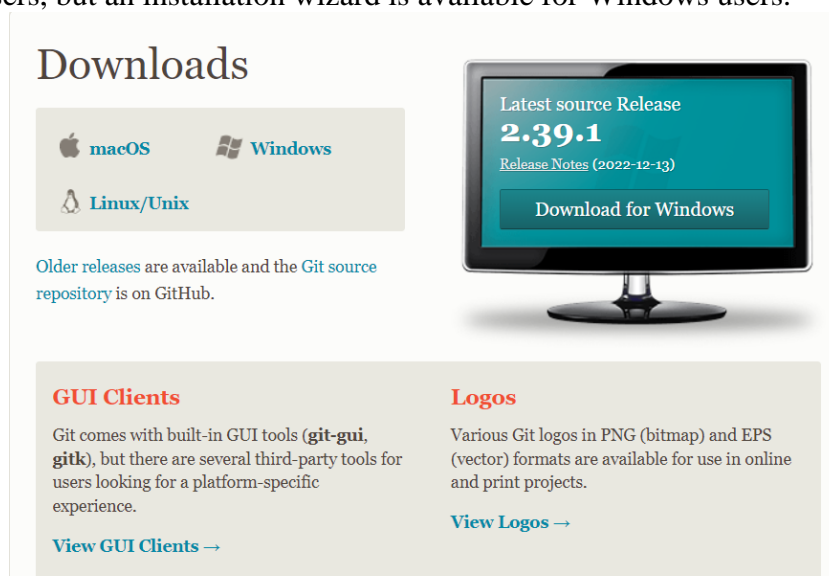
### **Git Terminologies:**

1. **Repository:** It is a directory that stores all the files, folders, and content needed for your project. It's the object database of the project, storing everything from the files themselves to the versions of those files, commits, deletions, etc.
2. **Branch:** A version of the repository that diverges from the main working project. Branches can be a new version of a repository, experimental changes, or personal forks of a repository for users to alter and test changes.
3. **Fork:** Creating a copy of a repository.
4. **Remote:** Updates a remote branch with the commits made to the current branch. You are literally "pushing" your changes onto the remote.
5. **HEAD:** HEAD is a reference variable used to denote the most current commit of the repository in which you are working. When you add a new commit, HEAD will then become that new commit.
6. **Master:** It is the primary branch of all repositories. All committed and accepted changes should be on the master branch.
7. **Merge:** Taking the changes from one branch and adding them into another (traditionally master) branch. These commits are usually first requested via pull request before being merged by a project maintainer.
8. **Origin:** The conventional name for the primary version of a repository. Git also uses origin as a system alias for pushing and fetching data to and from the primary branch. For example, `git push origin master`, when run on a remote, will push the changes to the master branch of the primary repository database.

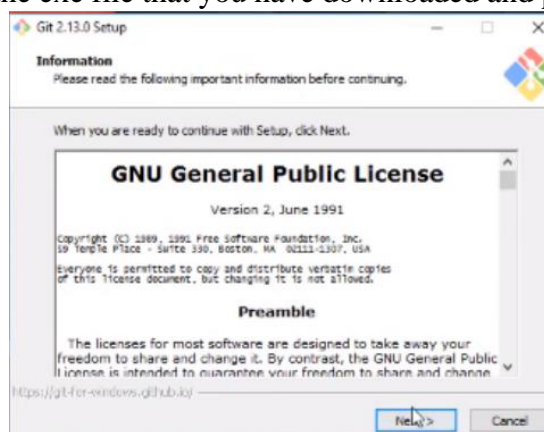
9. **Pull Requests:** If someone has changed code on a separate branch of a project and wants it to be reviewed to add to the master branch, that someone can put in a pull request. Pull requests ask the repo maintainers to review the commits made, and then, if acceptable, merge the changes upstream. A pull happens when adding the changes to the master branch.
10. **Push:** Updates a remote branch with the commits made to the current branch. You are literally “pushing” your changes onto the remote.
11. **Stash:** While working with Git, you may need to make multiple changes to files, but you may not want all changes to go in one commit. To put temporary changes on hold, you can “stash” your changes, essentially clearing them from the staging area until the changes are called again. You can only stash one set of changes at a time. To stash your staging area use `git stash [files]`; to retrieve the stashed files, run `git stash pop`. You can also clear the stashed files with `git stash drop`.

## INSTALLING GIT

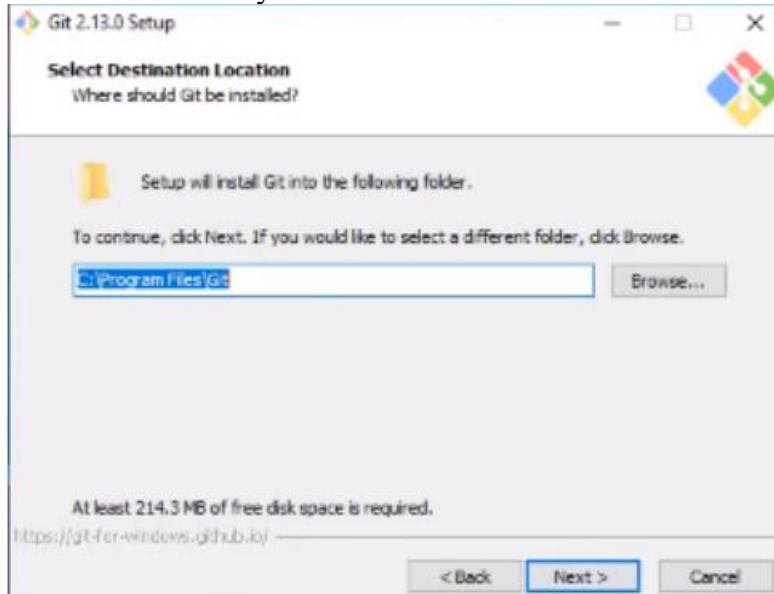
1. Open up your browser and go to <https://www.git-scm.com> and choose the download option for your operating system. It's a command-line operation for Linux and macOS users, but an installation wizard is available for Windows users.



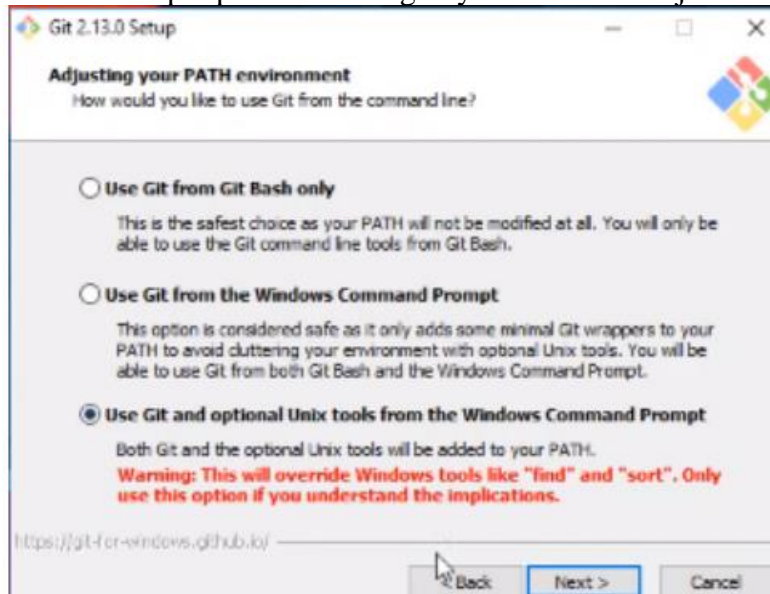
2. For windows: Open the exe file that you have downloaded and proceed further



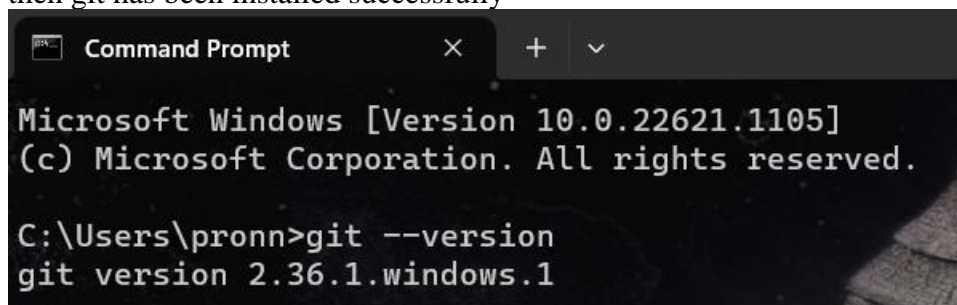
3. Choose the directory to install Git



4. Choose the apt option according to your need and adjust the Path environment

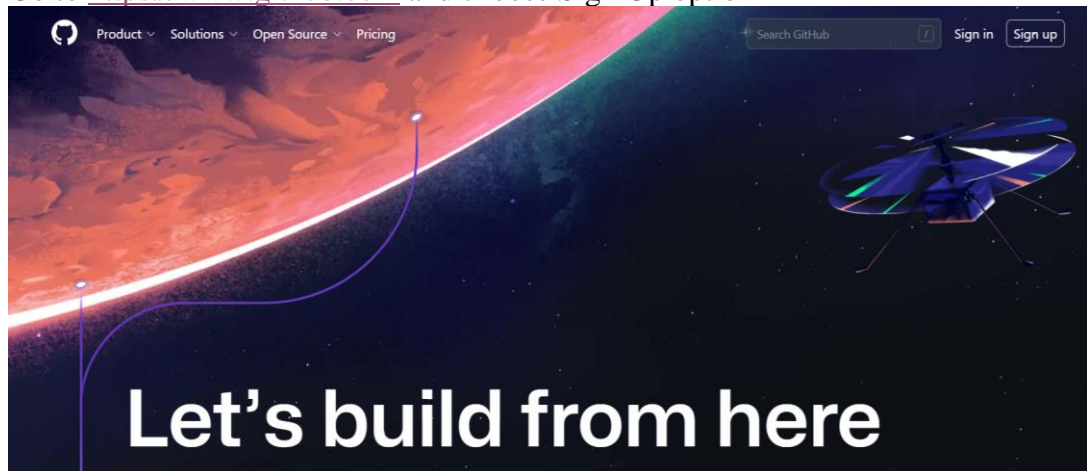


5. Proceed further, after the installation gets over, check for the git version. If it displays, then git has been installed successfully

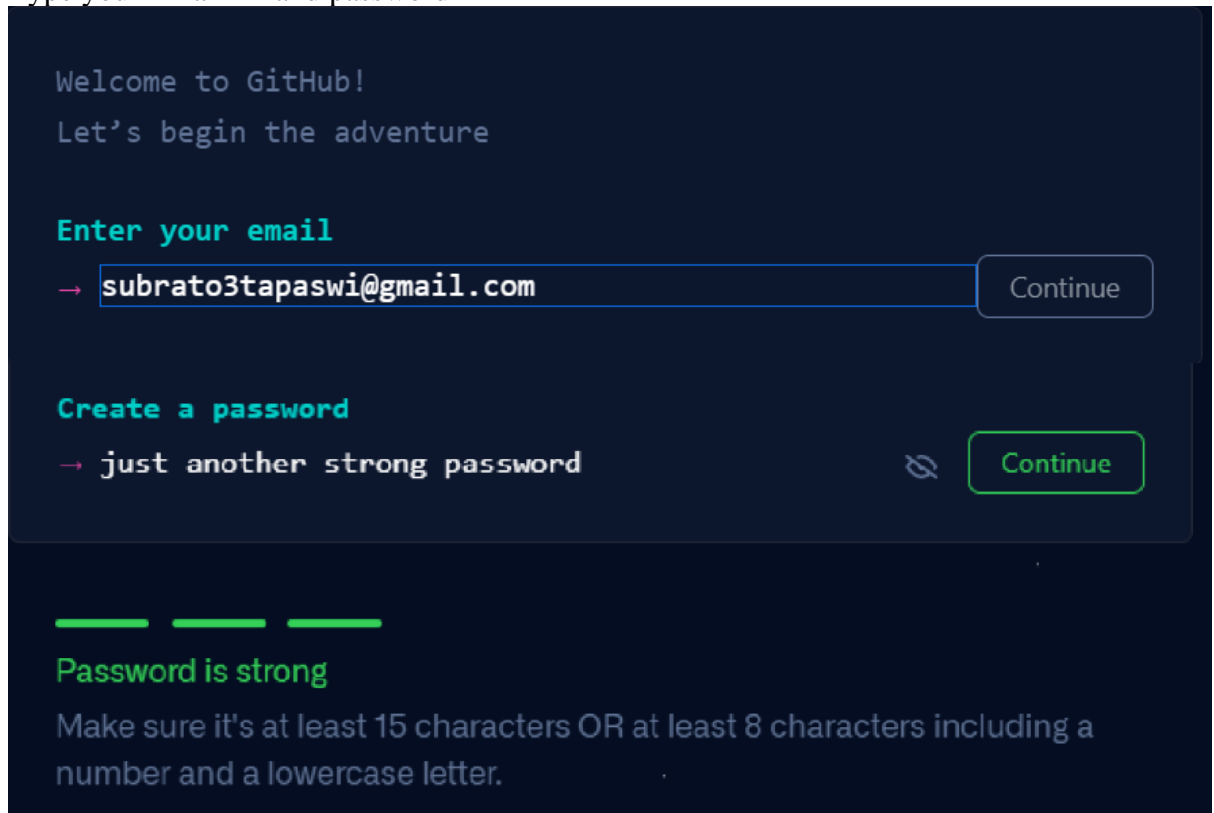


## CREATING A GITHUB ACCOUNT

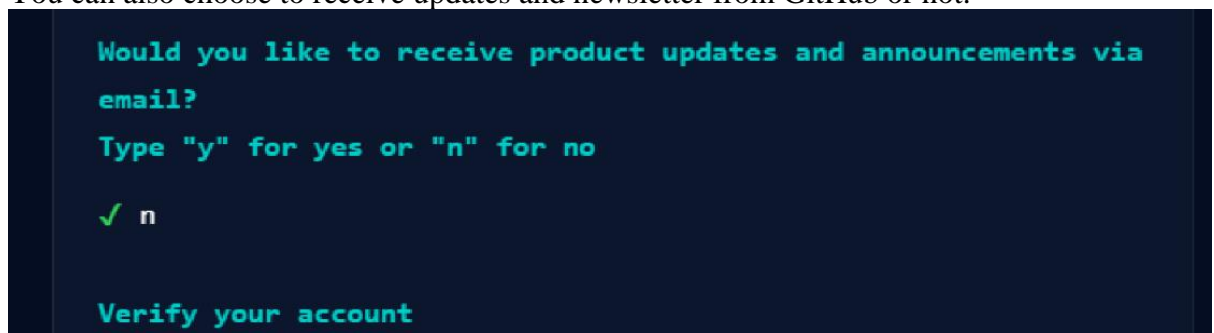
1. Go to <https://www.github.com> and choose Sign Up option



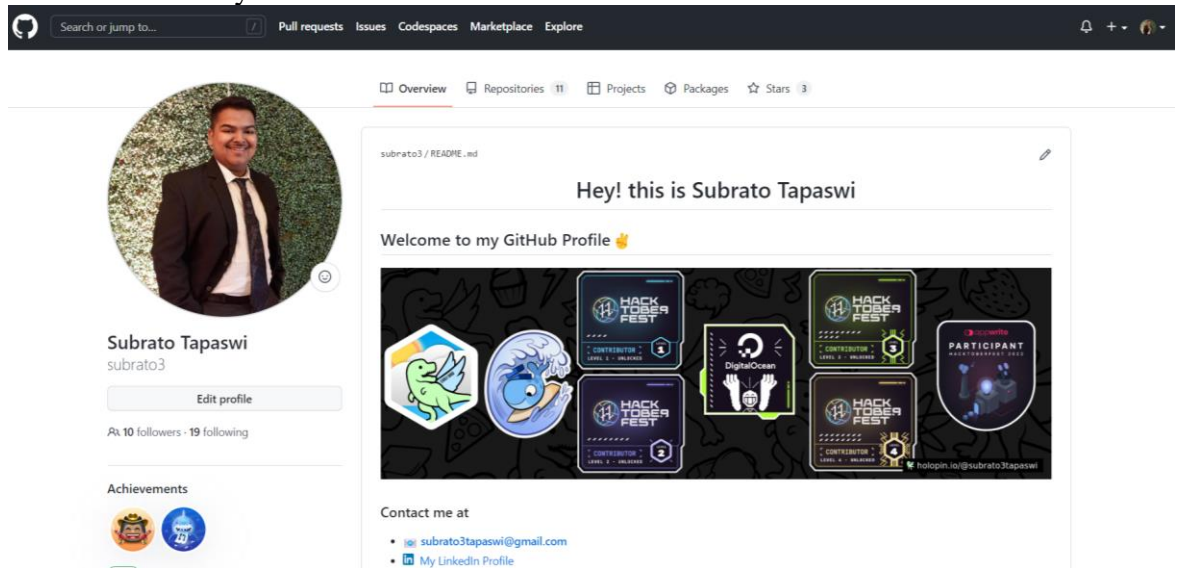
2. Type your Email ID and password

The image shows the GitHub sign-up form. It has a dark blue background with white text. The form starts with "Welcome to GitHub! Let's begin the adventure". The first section is "Enter your email" with a text input field containing "subrato3tapaswi@gmail.com" and a "Continue" button. The second section is "Create a password" with a text input field containing "just another strong password" and a "Continue" button. Below this, there's a green checkmark and the text "Password is strong". At the bottom, there's a note: "Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter."

3. Choose a username, this will also act as a link to your profile as people will be able to find you at **github.com/<username>**
4. You can also choose to receive updates and newsletter from GitHub or not.

The image shows the GitHub newsletter opt-in form. It has a dark blue background with white text. The form asks "Would you like to receive product updates and announcements via email?" and "Type 'y' for yes or 'n' for no". Below this, there's a green checkmark and the text "n". At the bottom, there's a button labeled "Verify your account".

5. Lastly, verify your email address by entering the code you received from GitHub. You can optionally fill up the details to personalise your GitHub. Thereafter, you are re-directed to your Dashboard.



**Conclusion:** We have studied the importance of Version Control System and Source Code Management. In addition, we have also installed Git CLI and created a GitHub account. We also studied about different terminologies of GitHub.