**Mahavir Education Trust's**

Shah & Anchor Kutchhi Engineering College,

**Chembur, Mumbai 400 088**

UG Program in Information Technology

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Experiment No. 02** | | | | | |
| **Date of Performance:** |  | | | | |
| **Date of Submission:** |  | | | | |
| **Program formation/ Execution/**  **ethical practices (07)** | **Documentation (02)** | **Timely Submission (03)** | **Viva Answer (03)** | **Experiment Marks (15)** | **Teacher Signature with date** |
|  |  |  |  |  |  |

**NAME: KRISHA SAYLA**

**ROLL.NO: 70**

**CLASS: TE-6**

**Experiment No. 2**

**Aim:** To Understand Version Control System / Source Code Management, install git and create GitHub account

**Lab Outcome:**  To obtain complete knowledge of the “version control system” to effectively track

changes augmented with Git and GitHub

**Theory:**

**Definition of Git:**

* **Git** is an **open-source distributed version control system**. It is designed to handle minor to major projects with high speed and efficiency. It is developed to co-ordinate the work among the developers. The version control allows us to track and work together with our team members at the same workspace.
* Git is foundation of many services like **GitHub** and **GitLab**, but we can use Git without using any other Git services. Git can be used **privately** and **publicly**.
* Git was created by **Linus Torvalds** in **2005** to develop Linux Kernel. It is also used as an important distributed version-control tool for **the DevOps**.
* Git is easy to learn, and has fast performance. It is superior to other SCM tools like Subversion, CVS, Perforce, and ClearCase.

**Features of Git:**



1. **Open Source**:

* Git is an open-source tool. It is released under the GPL (General Public License) license.

1. **Scalable**:

* Git is scalable, which means when the number of users increases, the Git can easily handle such situations.

1. **Distributed**:

* One of Git's great features is that it is distributed. Distributed means that instead of switching the project to another machine, we can create a "clone" of the entire repository.
* Also, instead of just having one central repository that you send changes to, every user has their own repository that contains the entire commit history of the project.
* We do not need to connect to the remote repository; the change is just stored on our local repository. If necessary, we can push these changes to a remote repository.

1. **Security**:

* Git is secure. It uses the SHA1 (Secure Hash Function) to name and identify objects within its repository.
* Files and commits are checked and retrieved by its checksum at the time of checkout. It stores its history in such a way that the ID of particular commits depends upon the complete development history leading up to that commit.
* Once it is published, one cannot make changes to its old version.

1. **Speed**:

* Git is very fast, so it can complete all the tasks in a while. Most of the git operations are done on the local repository, so it provides a huge speed.
* Also, a centralized version control system continually communicates with a server somewhere.  
  Performance tests conducted by Mozilla showed that it was extremely fast compared to other VCSs.
* Fetching version history from a locally stored repository is much faster than fetching it from the remote server. The core part of Git is written in C, which ignores runtime overheads associated with other high-level languages.
* Git was developed to work on the Linux kernel; therefore, it is capable enough to handle large repositories effectively. From the beginning, speed and performance have been Git's primary goals.

**GitHub Definition:**

GitHub is a Git repository hosting service. GitHub also facilitates with many of its features, such as access control and collaboration. It provides a Web-based graphical interface.

**Features of GitHub:**

* Collaboration
* Integrated issue and bug tracking
* Graphical representation of branches
* Git repositories hosting
* Project management
* Team management
* Code hosting
* Track and assign tasks
* Conversations

**Difference of Git & GitHub**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Git** | **GitHub** |
| 1. | Git is a software. | GitHub is a service. |
| 2. | Git is a command-line tool | GitHub is a graphical user interface |
| 3. | Git is installed locally on the system | GitHub is hosted on the web |
| 4. | Git is maintained by linux. | GitHub is maintained by microsoft. |
| 5. | Git is focused on version control and code sharing. | GitHub is focused on centralized source code hosting. |
| 6. | Git is a version control system to manage source code history. | GitHub is a hosting service for Git repositories. |
| 7. | Git was first released in 2005. | GitHub was launched in 2008. |
| 8. | Git has no user management feature. | GitHub has built-in user management feature. |

**Version Control System:**

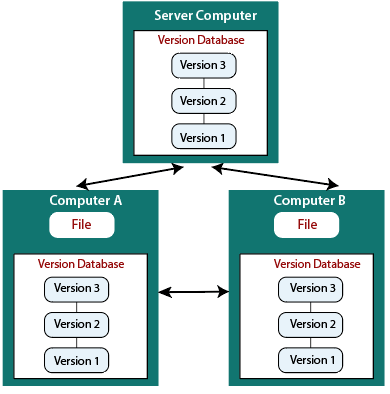
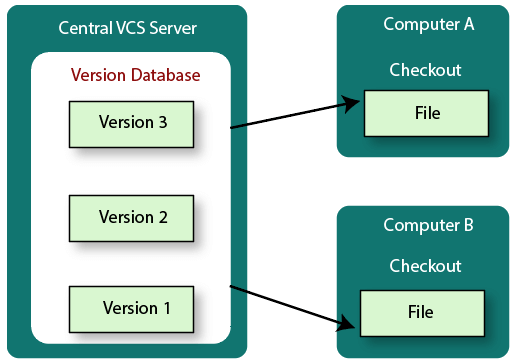
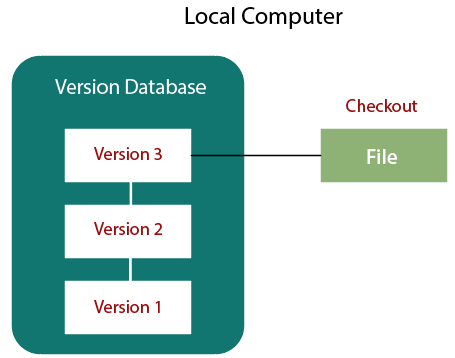
* A version control system is a software that tracks changes to a file or set of files over time so that you can recall specific versions later. It also allows you to work together with other programmers.
* The version control system is a collection of software tools that help a team to manage changes in a source code. It uses a special kind of database to keep track of every modification to the code.
* Developers can compare earlier versions of the code with an older version to fix the mistakes.

**Benefits:**

* Complete change history of the file
* Simultaneously working
* Branching and merging
* Traceability

**Types of Version Control System**

**1) Localized 2) Centralized 3) Distributed**



**1) Localized version Control System**

* The localized version control method is a common approach because of its simplicity. But this approach leads to a higher chance of error. In this approach, you may forget which directory you're in and accidentally write to the wrong file or copy over files you don't want to.
* To deal with this issue, programmers developed local VCSs that had a simple database. Such databases kept all the changes to files under revision control. A local version control system keeps local copies of the files.
* The major drawback of Local VCS is that it has a single point of failure.

**2) Centralized version control systems**

* The developers needed to collaborate with other developers on other systems. The localized version control system failed in this case. To deal with this problem, Centralized Version Control Systems were developed.
* These systems have a single server that contains the versioned files, and some clients to check out files from a central place.
* Centralized version control systems have many benefits, especially over local VCSs.

1. Everyone on the system has information about the work what others are doing on the project.
2. Administrators have control over other developers.
3. It is easier to deal with a centralized version control system than a localized version control system.
4. A local version control system facilitates with a server software component which stores and manages the different versions of the files.

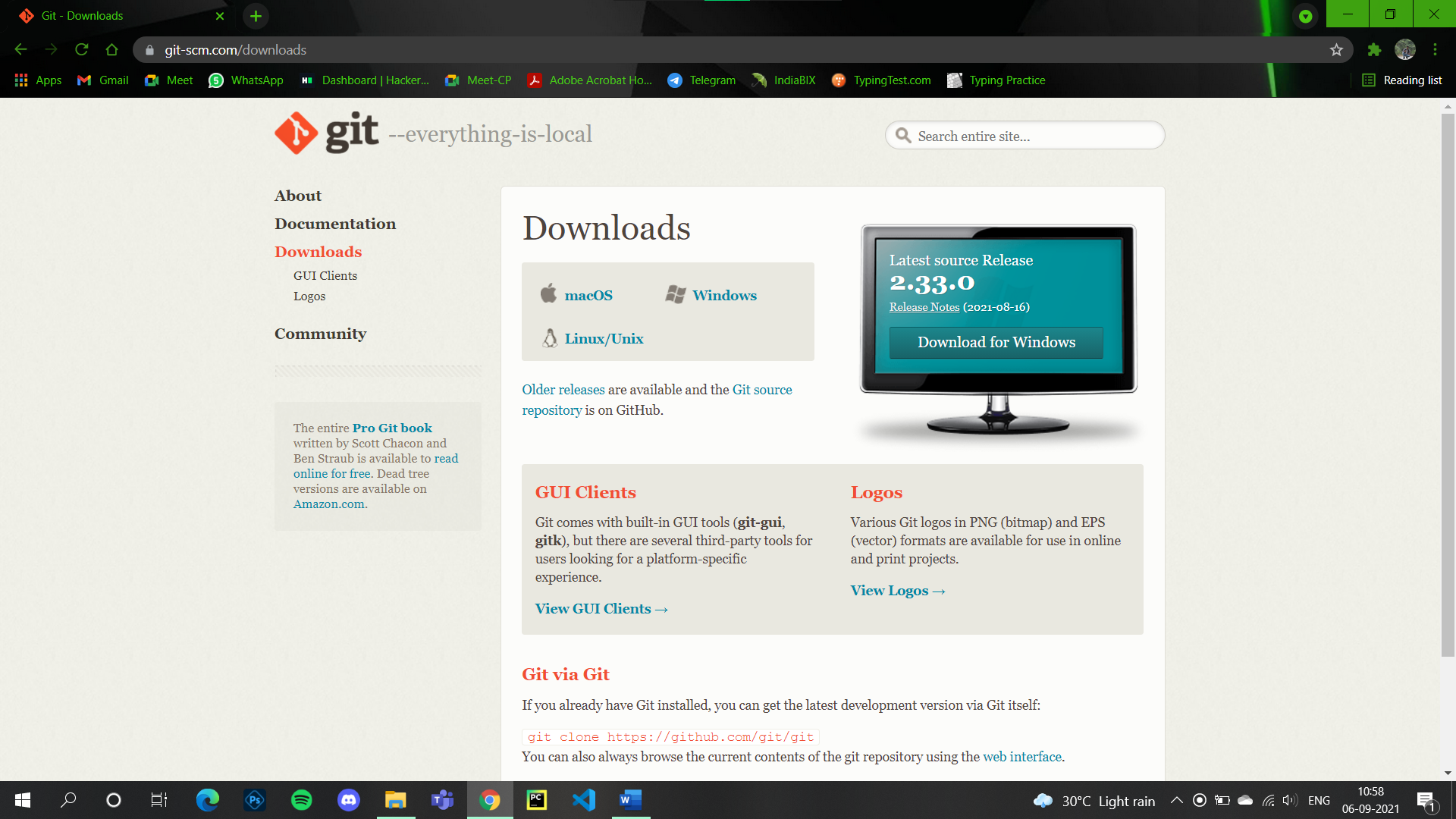
**3) Distributed version control systems**

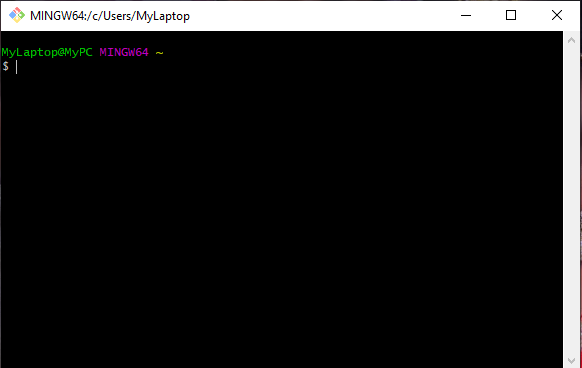
* Centralized Version Control System uses a central server to store all the database and team collaboration. But due to single point failure, which means the failure of the central server, developers do not prefer it. Next, the Distributed Version Control System is developed.
* In a Distributed Version Control System (such as Git, Mercurial, Bazaar or Darcs), the user has a local copy of a repository. So, the clients don't just check out the latest snapshot of the files even they can fully mirror the repository. The local repository contains all the files and metadata present in the main repository.
* DVCS allows automatic management branching and merging. It speeds up of most operations except pushing and pulling. DVCS enhances the ability to work offline and does not rely on a single location for backups. If any server stops and other systems were collaborating via it, then any of the client repositories could be restored by that server. Every checkout is a full backup of all the data.
* These systems do not necessarily depend on a central server to store all the versions of a project file.

**Output:**

**Installation of Git:**

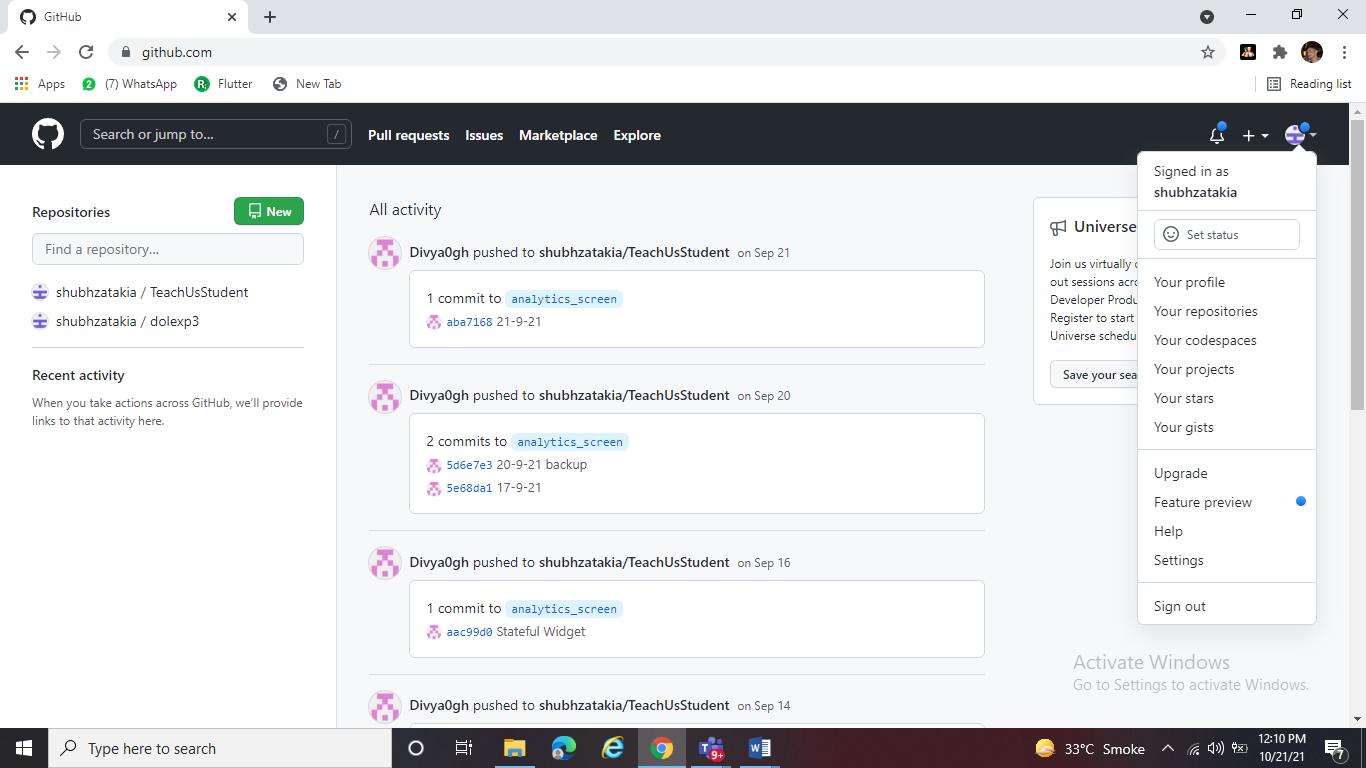
* Go to the website and download the ‘git’ file according to your system configuration.

****



**Creation of GitHub account:**

* Go to the website of GitHub and register with your email address for account creation.



Conclusion:

Version Control System was understood in this experiment. Git was installed and Git bash was exercised. GitHub account was created and a repository was made. Difference between Git and GitHub was made clear.