|  |  |  |
| --- | --- | --- |
|  | **K. J. Somaiya Institute of Engineering and Information Technology, Sion, Mumbai**  *Accredited ‘A’ Grade by NAAC with 3.21 CGPA*  *3 Programs Accredited by National Board of Accreditation*  *Permanently Affiliated to University of Mumbai,*  *Best College Award by University of Mumbai (Urban Region), ISTE (MH), and CSI (Mumbai)*  *UGC Recognized Institute under Section 2(f) and 12(B) of the UGC Act, 1956* |  |

# Experiment No. 3

## Aim:

* Version control using Git.

## Theory:

### Version Control:

Version control, otherwise called source control, is the act of following and overseeing changes to software code. Version control systems [VCS] are software apparatuses that help software groups oversee changes to source code over the long haul. As advancement conditions have sped up, VCS help coding/ development teams work quicker and become more astute.

It is like a backup saving program for your project. By tracking and logging the changes you make to your file or file sets over time, a version-control system gives you the power to review or even restore earlier versions. Version Control takes snapshots of every revision to your project. You can then access these versions to compare or restore them as needed.

### 1. Git -

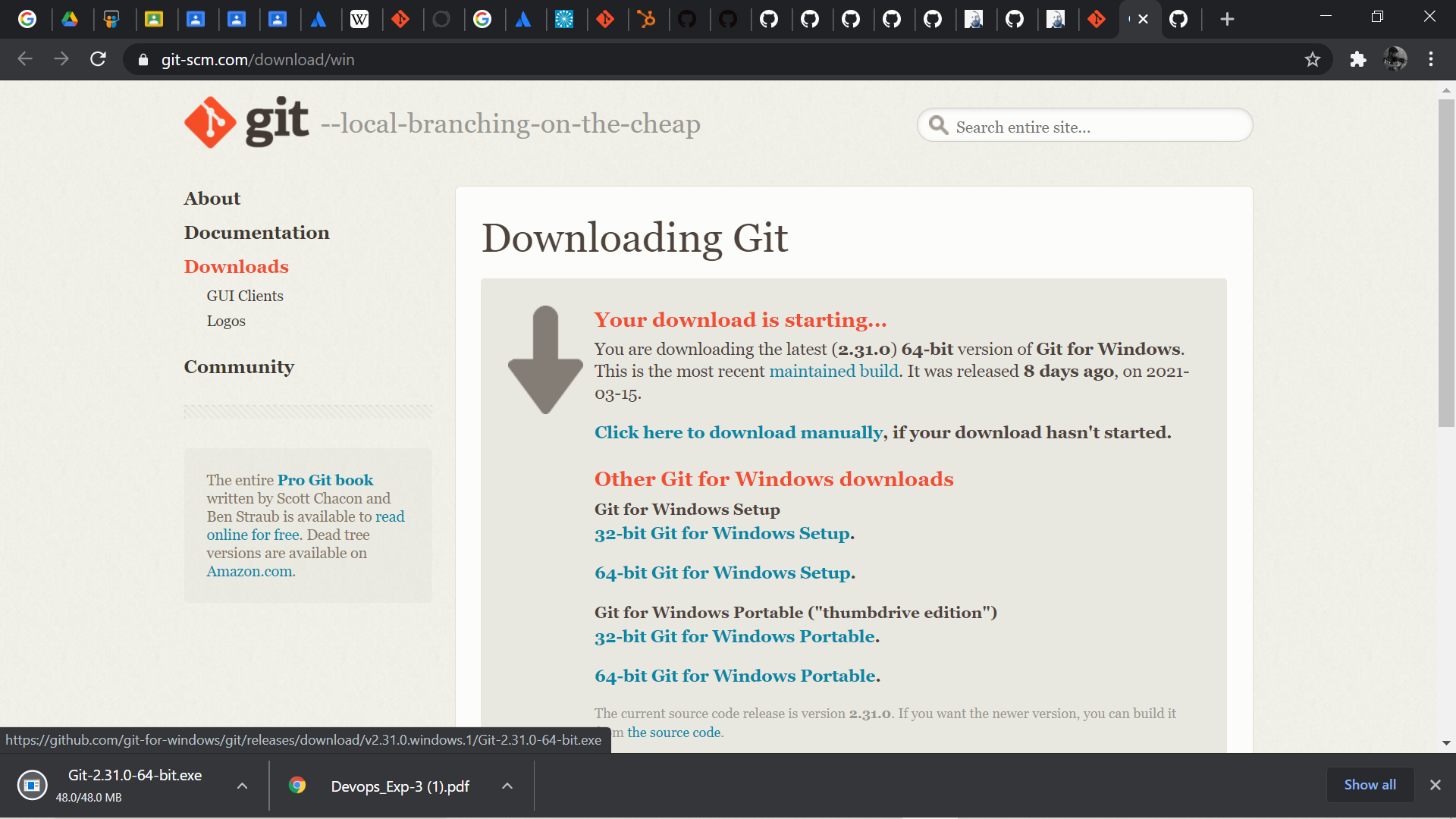
Git is a free and open-source distributed Version Control System or Protocol. It is a framework that permits you to record changes to documents over the long haul and permits you to see changes and explicit renditions of those documents later on.

Initially Git was a part of The Linux Kernel Project [1991-2002]. In 2002, the Linux kernel project began using a proprietary DVCS called BitKeeper

In 2005, the relationship between the community that developed the Linux kernel and the commercial company that developed BitKeeper broke down, and the tool’s free-of-charge status was revoked. This prompted the Linux development community (and in particular Linus Torvalds, the creator of Linux) to develop their own tool based on some of the lessons they learned while using BitKeeper. Some of the goals of the new system were as follows:

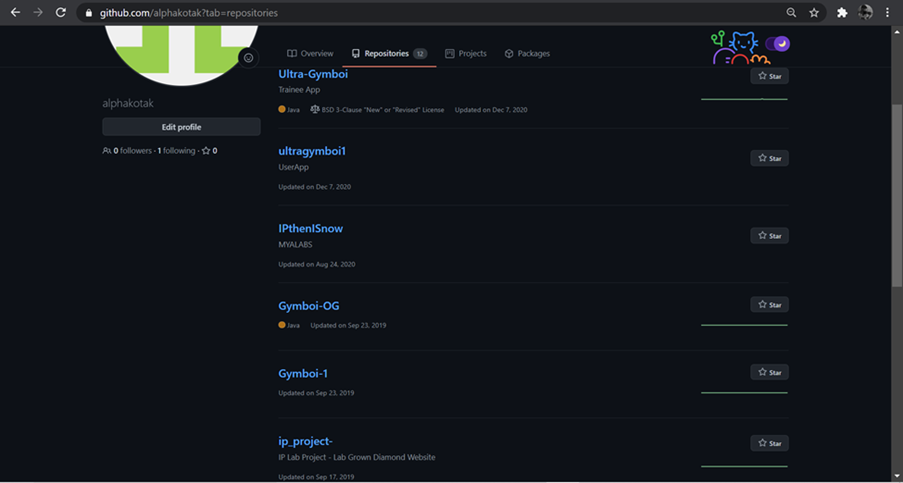
* Speed
* Simple design
* Strong support for non-linear development (thousands of parallel branches)
* Fully distributed
* Able to handle large projects like the Linux kernel efficiently (speed and data size)

Since its birth in 2005, Git has evolved and matured to be easy to use and yet retain these initial qualities. Git is largely helpful in projects which involve a group of developers working on a single project. Git helps in managing the code changes and errors for projects with a large codebase. It makes it easy to stage changes and revert back to a specific version easily. It has is now being used to automate the process of deployment for an application.



### 2. GitHub

At a high level, GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code. It is is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.



#### To create a new repository

1. In the upper right corner, next to your avatar or identicon, click and then select New repository.
2. Name your repository hello-world.
3. Write a short description.
4. Select Initialize this repository with a README.

#### Make and commit changes

1. Click the Readme.md file
2. Click the pencil icon in the upper right corner of the file view to edit.
3. In the editor, write a bit about yourself.
4. Write a commit message that describes your changes.
5. Click **Commit changes** button.

