**“Open Source Software Laboratory”**

**Code: 4IT475**

Submitted by

**Mr. Pratham Yadav (2020BTEIT00006)**

**Email:** pratham.yadav@walchandsangli.ac.in

**Mobile:** 7058492414

****

DEPARTMENT OF INFORMATION TECHNOLOGY

**WALCHAND COLLEGE OF ENGINEERING, SANGLI**

**(An Autonomous Institute)**

**2023-2024**

1. **Use of Version Control System.**

**(**[**Mercurial**](https://www.mercurial-scm.org/) **(hg),** [**Bazaar**](http://bazaar-vcs.org/)**, Monotone, etc: Any One).**

**(Submission by Individual and Group [I and G])**

***Objective****: To use the online and offline Version Control System in Open Source/for their project work.*

***Outcome****: lifelong learning (PO: b, c, k, l)*

Students have to experiment any two **Version Control System** and use the tool for their project/FOSS project/mini project/ etc.

Sample code developments example of **Version Control System** on both Windows and Linux clients/server.

Make the Official Repositories of any one **Version Control System** on docker store (https://hub.docker.com/) and experiment.

In Journal, They have to write Basic Information about **Version Control System**, commands, their working, diagrams, differences, pros and cons, developments history, etc .

Reference:-

1. https://try.github.io/levels/1/challenges/1
2. https://github.com/princeton-8/princeton-8.github.io
3. http://wiki.openhatch.org/Open\_Source\_Comes\_to\_Campus/Practicing\_Git/Students
4. GIT Official Documentation:- [http://git-scm.com/documentatio](http://git-scm.com/documentation)n
5. SVN Official Documentation:- <http://svnbook.red-bean.com/en/1.7/index.html>
6. Perforce Helix is a commercial, proprietary revision control system developed by Perforce Software
7. <https://www.smashingmagazine.com/2008/09/the-top-7-open-source-version-control-systems/>
8. http://wiki.bazaar.canonical.com/WindowsDownloads

## **4. Use of Version Control System.**

**Title-** Use of Version Control System.

**Objective-** To use/experiment with the online and offline version control system for foss project work

**Outcome-** lifelong learning

**About Git-**

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is used to track changes in the open-source code, enabling multiple developers to work together on non-linear development

**Creator-** Linus Torvalds created Git in 2005 for the development of the Linux kernel

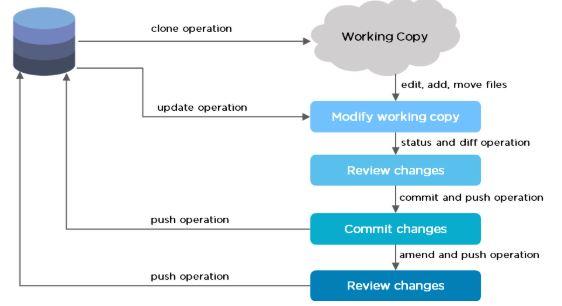
**Installation-**

Open the terminal and write the command **sudo apt-get install git** for Linux-based machines and for Windows download the file from Git’s official website and run it

**Features of Git-**

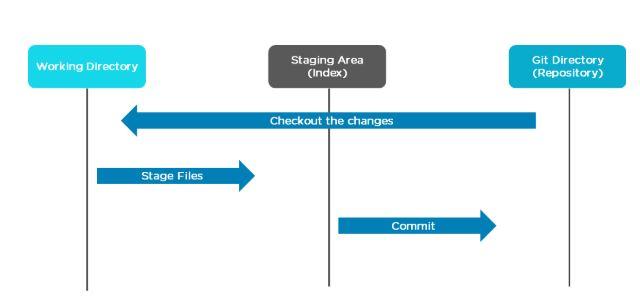
* Tracks history
* Free and open-source
* Supports non-linear development
* Creates backups
* Scalable
* Supports collaboration
* Branching is easier
* Distributed development

**Git workflow-**

****

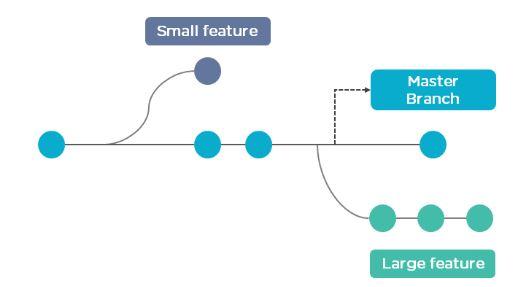
The Git workflow is divided into three states:

* Working directory - Modify files in your working directory
* Staging area (Index) - Stage the files and add snapshots of them to your staging area
* Git directory (Repository) - Perform a commit that stores the snapshots permanently in your Git directory. Check out any existing version, make changes, stage them, and commit.



**Branch in Git-**

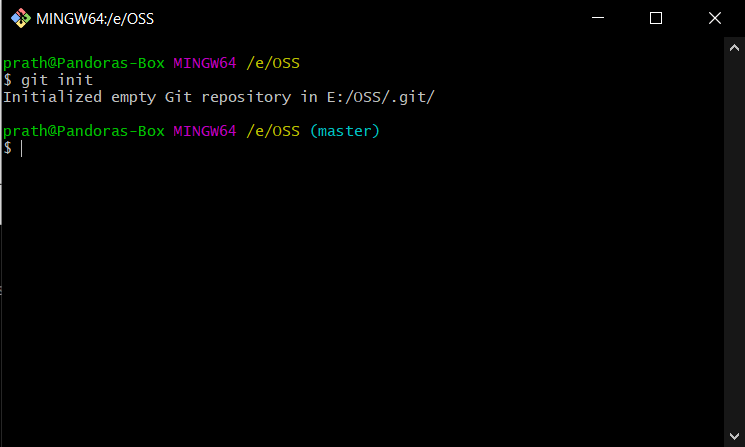
A branch in Git is used to keep your changes until they are ready. You can do your work on a branch while the main branch (master) remains stable. After you are done with your work, you can merge it with the main office.

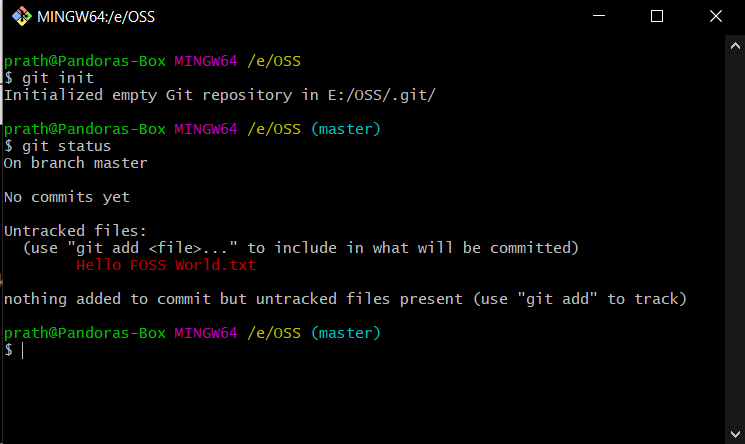


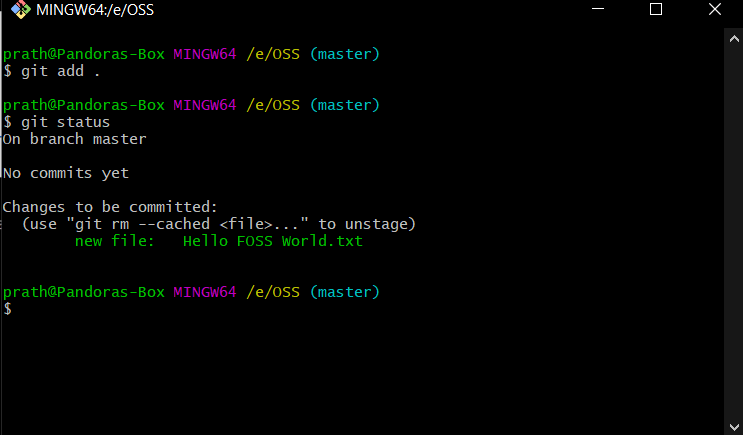
The above diagram shows there is a master branch. There are two separate branches called “small feature” and “large feature.” Once you are finished working with the two separate branches, you can merge them and create a master branch.

**Some commands in Git-**

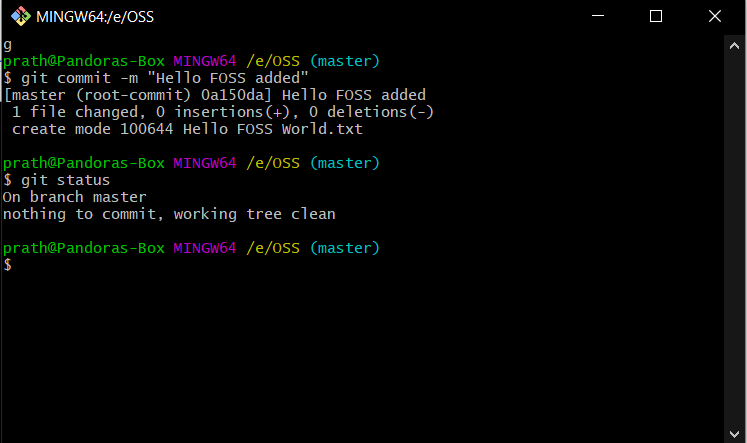
Create Repositories- creates a new git repository

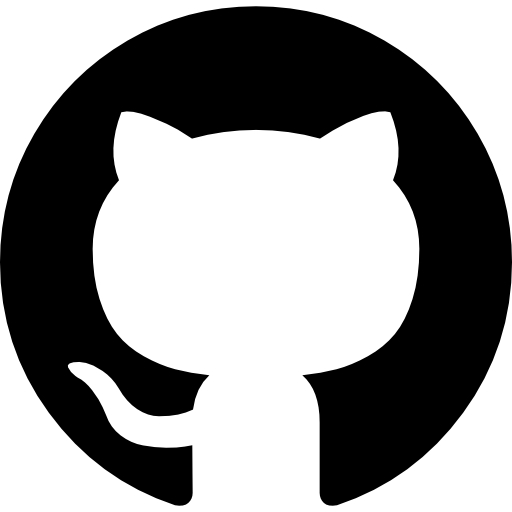


Check git status- Gives status of the working directory and staged area

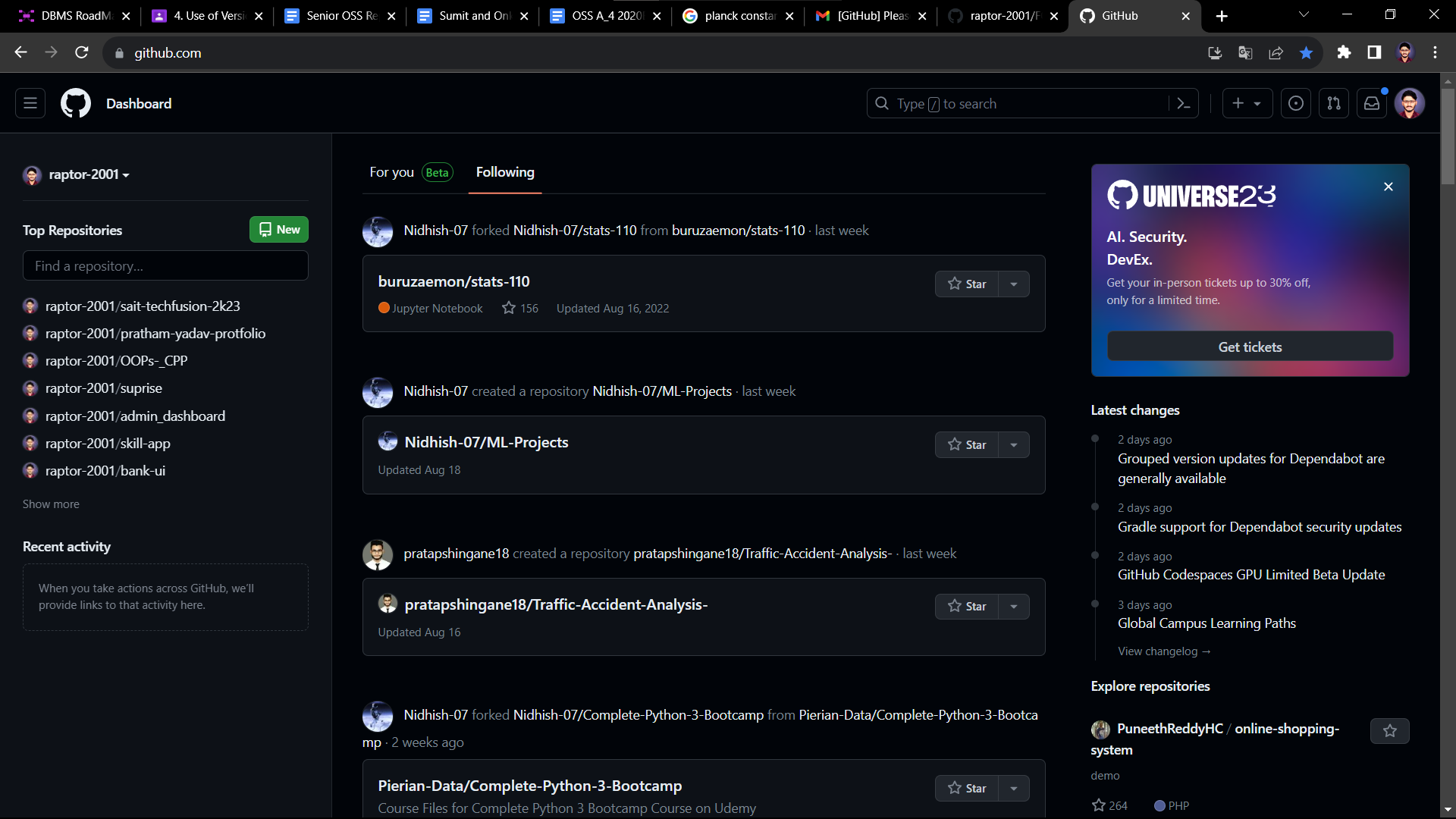
Git add command- Adds untracked files to the staged area

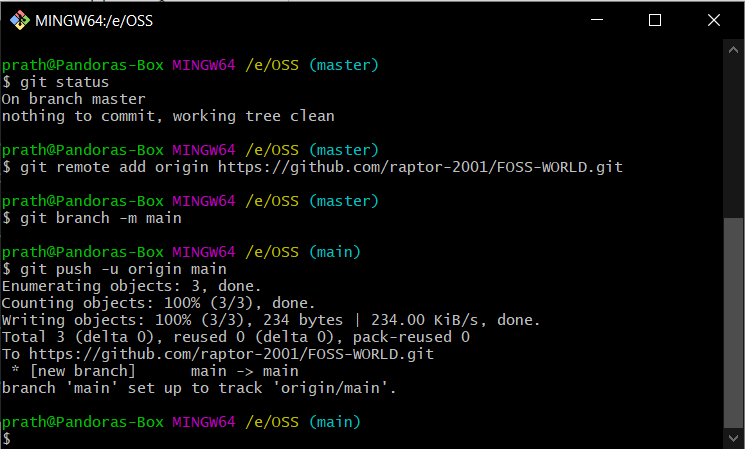
Git status- Commits the changes to the version control

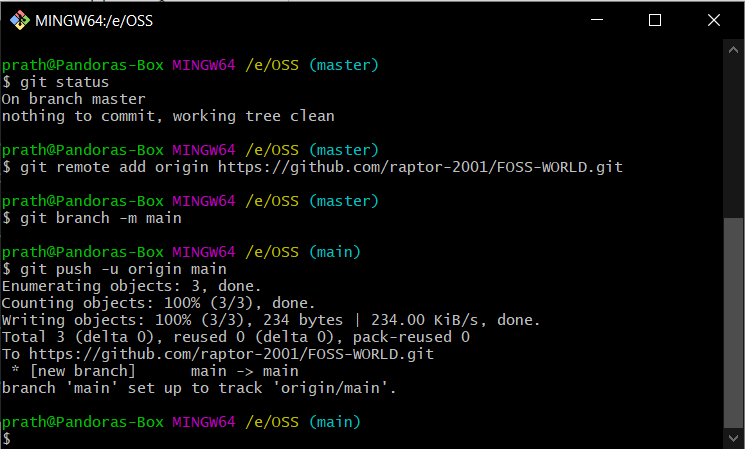


**Git-hub -** At a high level, GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code.

Home screen-

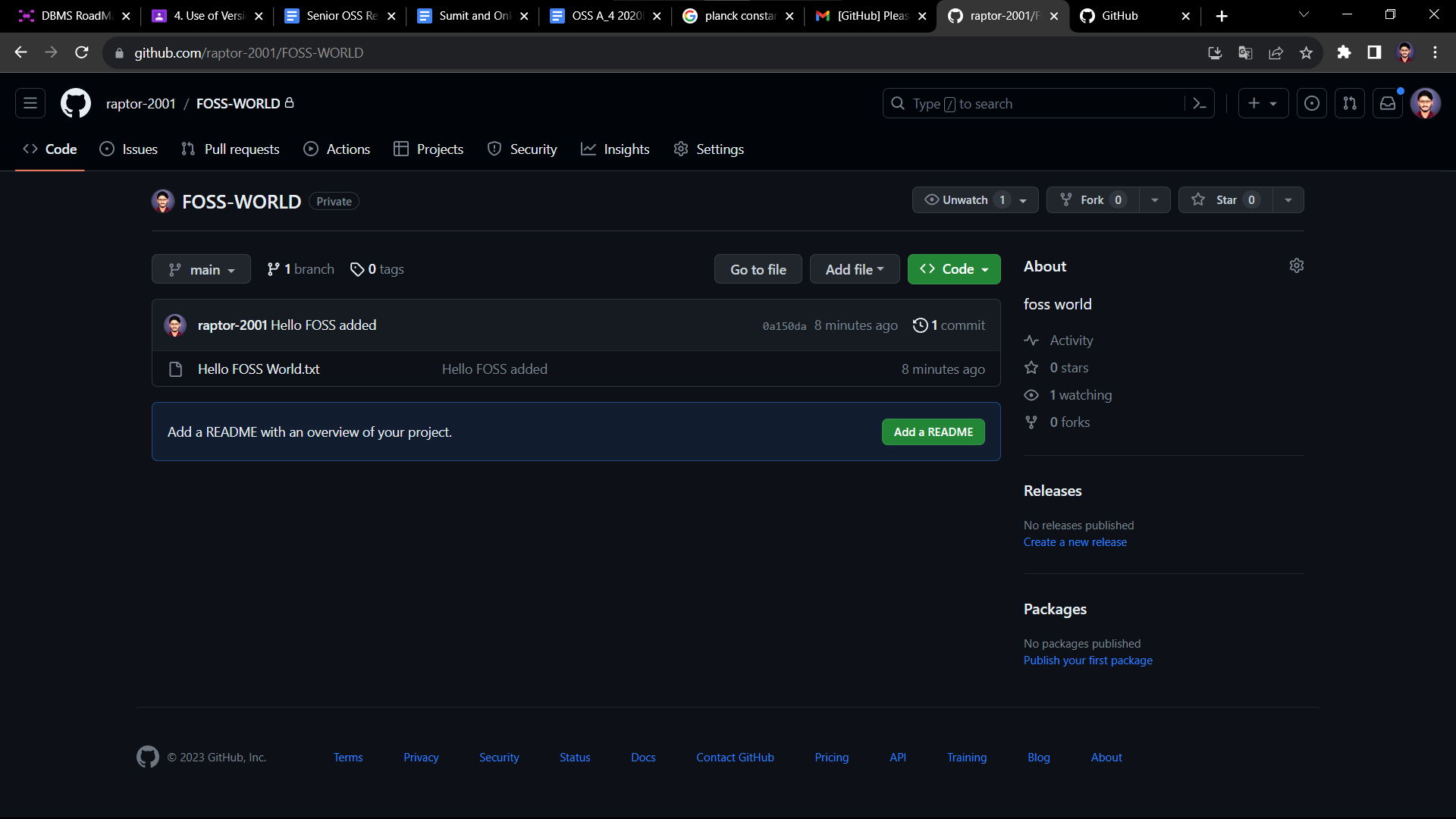


Connecting with remote repository-



Pushing to remote repository-

Check if the changes has been updated to git-hub repository-



## 

## 

## 

## **Conclusion :**

1.We learnt different VCS tools. 2.Compared Git with other VCS.

3.Overall use of GIT is topp-notch and productive.

## **Reference**

[ 1 ] <https://en.wikipedia.org/wiki/Git>

[ 2 ] <http://git-scm.com/documentation>

[ 3 ] <https://www.git-tower.com/blog/git-cheat-sheet/>