1. **VCS: Demonstrate to use of Version Control System (Git offline: on local machine with multiple user and connect to online vcs/github/bitbucket).**

Multiuser usage with team leader role and coder role/Merge/fork (branching)/diff/versions/commit/pull/push on repository.

Compare it with svn. (on answer sheet)

1. **\*VCS: Demonstrate to use of Version Control System using git containers (Git offline: on local machine with multiple user and connect to online vcs/github/bitbucket).**

Multiuser usage with team leader role and coder role/Merge/fork (branching)/diff/versions/commit/pull/push on repository.

Compare it with svn. (on answer sheet)

1. **Bug tracking: Demonstrate the use/features of online Bug Tracking/Issue Tracking "BugZilla".**

Take source code of any open source you like and find minimum three bugs (mention their type on answer sheet) in terms removing error/bug or adding feature to it.

[Bug tracking](https://www.openproject.org/collaboration-software-features/#bug-tracking) on any suitable open source (code) from internet.

Compare it with other online **Bug tracking tool. How bug tracking improves the quality of code** (on answer sheet).

1. **VCS: Demonstrate to use of Version Control System (SVN).**

Multiuser usage with team leader role and coder role/Merge/fork (branching)/diff/versions/commit/ pull/push on repository. And connect it to any online repo.

1. **\*VCS: Demonstrate to use of Version Control System (SVN) using containers.**

Multiuser usage with team leader role and coder role/Merge/fork (branching)/diff/versions/commit/ pull/push on repository. And connect it to any online repo.

1. **VCS: Development of contribution to existing Open Source Software (on line upload of its git/svn repository with your valid login)**

(Language: java/pyth/perl/c/cpp/etc ).

Take source code of any open source you like and contribute in terms removing error/bug (Min Two) to it.

Upload the contributed code. Show the date time of upload, bug reported and resolved.

1. **VCS: Development of contribution to existing Open Source Software (on line upload of its git/svn repository with your valid login)**

(Language: java/pyth/perl/c/cpp/etc ).

Take source code of any open source you like and contribute in terms adding feature (Min Two) to it.

Upload the contributed code. Show the date time of upload, added report.

1. **CMS : Demonstrate the use/features of CMS software: Joomla.**

Crate users and show how Joomla manages web sites for a client. Also implement the working of core features of Joomla.

Compare it with other **CMS** like Druple (on answer sheet)

1. **\*CMS : Demonstrate the use/features of CMS software: Using Joomla container .**

Crate users and show how Joomla manages web sites for a client. Also implement the working of core features of Joomla.

Compare it with other **CMS** like Druple (on answer sheet)

1. **Create packages: Create of RPM packages.**

Take utility source code (C++/java/Python/, etc) of any open source code you like and contribute in terms removing error/bug or adding feature to it.

Demonstrate the package on suitable OS and upload at its repository.

Compare it with Debian package manager (on answer sheet)

1. **\*Create packages: Create of RPM packages.**

Take utility source code (multiple modules/code packaging of c/cpp/python/java, etc) of any open source you like.

Demonstrate the package on suitable OS and upload at its repository.

Compare it with Debian package manager (on answer sheet).

1. **\*Create packages: Create of Debian packages.**

Take utility source code (Multiple modules/code packaging of python,C/C++/Java etc) of any open source you like and contribute in terms removing error/bug or adding feature to it.

Demonstrate the package on suitable OS and upload at its repository.

Compare it with RPM package manager (on answer sheet)

1. **Create packages: Create of Debian packages.**

(multiple modules/code packaging of java/c/cpp).

Take utility source code of any open source in c/cpp/ajava/python/shell script you like.

Demonstrate the package on suitable OS and upload at its repository.

Compare it with RPM package manager (on answer sheet)

1. **Project Management tool: Demonstrate the use of Project Management tool: “SONAR” for managing projects.**

[Project planning and scheduling](https://www.openproject.org/collaboration-software-features/#project-planning)/ [Product roadmap and release planning](https://www.openproject.org/collaboration-software-features/#product-management)/ [Task management and team collaboration](https://www.openproject.org/collaboration-software-features/#task-management)/ [Agile and Scrum](https://www.openproject.org/collaboration-software-features/#agile-scrum)/[Time tracking, cost reporting and budgeting](https://www.openproject.org/collaboration-software-features/#time-tracking)/ [Bug tracking](https://www.openproject.org/collaboration-software-features/#bug-tracking) on any suitable open source (code) from internet.

Compare it with other **Project Management tool** (on answer sheet)

1. **\*Project Management tool: Demonstrate the features of Project Management tool: “SONAR” for managing projects.**

Demonstrate the code quality features of sonar for open source code (yours project code) c/cpp/java/python codes.

Compare it with other **Project Management tool** (on answer sheet)

1. **Project Management tool: Demonstrate the features of Project Management tool: “Jira” for managing projects.**

Demonstrate the code quality features of jira for open source code (yours project code) c/cpp/java/python codes.

Compare it with other **Project Management tool** (on answer sheet)

1. **Project Management tool: Demonstrate the features of Project Management tool: “Jira” for managing projects.**

Demonstrate the code quality features of jira for open source code (yours project code) c/cpp/java/python codes.

Compare it with other **Project Management tool** (on answer sheet)

1. **Bug Tracking: Demonstrate the use/features of Bug Tracking/management: "YouTrack".**

Take source code of any open source you like and find minimum three bugs (mention their type on answer sheet ) in terms removing error/bug or adding feature to it.

[Bug tracking](https://www.openproject.org/collaboration-software-features/#bug-tracking) on any suitable open source (code) from internet.

Compare it with other **Bug tracking tool. How bug tracking improves the quality of code** (on answer sheet).

1. **CMS: Demonstrate the use/features of CMS software: "Drupal".**

Crate users and show how Drupal manages contents of web sites for a client. Also configure the working of core features of Drupal.

Compare it with other **CMS** like Joomal/(on answer sheet)

1. **CMS: Demonstrate the use/features of CMS software: "Media Wiki” from bitnami or official website.**

Crate users and show how wiki manages contents for a client. Also configure the working of core features of wiki. Can we use wiki as alternative to writing journal

Compare it with other **CMS** (on answer sheet)

1. **Server: Configure and demonstrate the use of FTP**

Show the imp steps and file name of configurations**.** (on answer sheet). Creates users and put file for sharing among users. Use admin for creation of pub folders.

1. **Server: Configure and demonstrate the use of Telnet.**

Show the imp steps and file name of configurations**.** (on answer sheet). Creates users and demo user accounts from admin login.

1. **\*Server: Configure and demonstrate the use of NIS and NFS.**

Show the imp steps and file name of configurations**.** (on answer sheet). Creates users and demo use of accounts from remote desktop Linux logins.

1. **Project Management tool: Demonstrate the use/features of Project Management tool: “Open Atrium” for managing.**

[Project planning and scheduling](https://www.openproject.org/collaboration-software-features/#project-planning)/ [Product roadmap and release planning](https://www.openproject.org/collaboration-software-features/#product-management)/ [Task management and team collaboration](https://www.openproject.org/collaboration-software-features/#task-management)/ [Agile and Scrum](https://www.openproject.org/collaboration-software-features/#agile-scrum)/[Time tracking, cost reporting and budgeting](https://www.openproject.org/collaboration-software-features/#time-tracking)/ [Bug tracking](https://www.openproject.org/collaboration-software-features/#bug-tracking) on any suitable open source (code) from internet.

Compare it with other **Project Management tool** (on answer sheet)

1. **Bug tracking: Demonstrate the use/features of Bug Tracking/management: Mantis.**

Take source code of any open source you like and find minimum three bugs (mention their type on answer sheet) in terms removing bugs. Show users and bug management. Is the VCS can attached to mantis?

Compare it with other **Bug tracking tool. How bug tracking improves the quality of code** (on answer sheet)

1. **Bug tracking: Demonstrate the use/features of Bug Tracking/management: Remine.**

Take source code of any open source you like and find minimum three bugs (mention their type on answer sheet) in terms removing bugs. Show users and bug management. Is the VCS can attached to mantis?

Compare it with other **Bug tracking tool. How bug tracking improves the quality of code** (on answer sheet)

1. **\*Bug Tracking: Demonstrate use** of suitable bug tracking tool and create the Docker image of that tool. Push that image on Docker-hub account.

Run the docker container from recently created image and run that docker container.

Push that image.

1. **Bug Tracking: Demonstrate use** of suitable bug tracking tool and use the Docker image of bug tracking tool. Demo management of users and bugs.
2. **\*Code review: Demonstrate use** of suitable code review (from bitnami) tool and demo its used for how to write code review.
3. **\*CMS: Demonstrate use** of Mastodon, which is self-hosted social network from bitnami.
4. **Docker: Write** a Docker File to pull the Ubuntu with open jdk and write any java application. Give the difference between YML and dockerfile.
5. **\*Docker: Create** two applications in two different docker containers. Push those applications and run to show the communications between two Dockers. (Hint IPC)
6. **\*Docker: Pull** the LAMP Stack container from Docker hub and host a web application of your own. Push that image back to repository. Make use of database.
7. **\*Docker: Create** a web application with simple web page containing login details and create a docker image of the application.(Use Apache Web server)

Run the Docker container from recently created image and run the container at port number 80 in host system. Push that image to repository. Make use of database. Try to access it from other instance of docker.

1. **\*Docker :With the help of Docker-compose deploy the ‘Wordpress’ and ‘Mysql’ container and access the front end of ‘Wordpress’**
2. **\*Docker: A. Create a simple Hello-world python flask application and create the docker image of that Flask application.**

**B. Run the docker container from recently created image and run that docker container to 5000 port of host system.**

1. **\*Docker: Create the ‘nginx’ container from ‘nginx’ image. And create the load balancing so that if we go to tha address of ‘nginx ‘ it can redirect it to the above created applications (Flask and Wordpress).**
2. **Create packages: For Calculator code in Shell build a simple RPM package and Install that on your system.**
3. **Create packages: For Calculator code in Shell build a simple DEB package and Install that on your system.**
4. **VCS:Demonstrate version control system for your code using MERCURIAL (hg). Refer:** <http://mercurial.selenic.com/wiki/Tutorial> **and pull code from bitbucket and demo three function of vcs.**
5. **\*CMS: Install and Demonstrate Vanilla Forum CMS on your system. Refer:** <http://vanillaforums.org/get/vanilla-core>
6. **\*CMS: Install and Demonstrate Media Wiki CMS on your system. Refer:** <http://www.mediawiki.org/wiki/Installation>
7. **\*Docker : Create a web application with simple web page containing login details and create a docker image of the application.(Use Apache Web server)**
8. **\*Docker: Run the Docker container from recently created image and run the container at port number 80 in host system.**
9. **\*Docker: Write a python program to perform arithmetic operations and create Docker image accordingly.**
10. **\*Docker: Run the Docker container with created image for c/java/python .**
11. **\*Docker: Create a simple web application using LAMP Stack on docker container.**
12. **\*Docker :Create a web application with simple web page containing login details and create a docker image of the application.(Use Ngnix Web server)**
13. **\*Docker: Run the Docker container from recently created image and run the container at port number 80 in host system.**
14. **\*Docker: Create a simple Hello-world python flask application and create the docker image of that Flask application.**
15. **\*Docker: Run the Docker container from recently created image and run that docker container to 5000 port of host system.**
16. **\*Docker: Pull the LAMP Stack container from docker hub and host a web application of your own.**
17. **\*Docker: Create a Docker image of simple web application from using HTTP web server at port 5000 in host.**
18. **\*Docker: Create a Docker image of simple login form using Flask on port 7000.**
19. **\*Docker: Create a Docker image of simple login form using django on port 6000.**
20. **\*Docker: Create a container with ngnix web server and create one more container with mysql.**
21. **\*Docker: Create a simple web form to insert the records in mysql data base.**
22. **\*Docker: Mount any directory of host system to the container.**
23. **\*Docker: Write a Docker File to pull the Ubuntu with open jdk and write any java application.**
24. **\*Docker :Run a LAMP Stack Container at port 8080 and host media wiki site on native machine.**
25. **\*Docker: Write a C program to create singly linked list and containerize it.**
26. **\*Docker: Create a LAMP Stack container and host a web application** of your own.
27. **Create packages: For String Addition code in Python build a simple RPM package and Install that on your system.**
28. **Create packages: For String Addition code in Python build a simple DEB package and Install that on your system.**
29. CMS: Install and Demonstrate Joomla CMS on your system. Refer: <http://docs.joomla.org/J3.1:Installing_Joomla>!
30. \*Documentation Tools: Install and Demonstrate doxygen for report generation. Refer: <http://www.cs.cmu.edu/~410/doc/doxygen.html>
31. \*Kernel compilation Compile and demo any suitable Linux Kernel on your system. Download it from: <https://www.kernel.org/>

Make change in kernel code before compile, and show it during demo

1. **Open Source Project Development**.:Use the Calculator code in Python from [github.com/drebin8/calculator](https://github.com/drebin8/calculator). Study and enhance the open source code on github.

**Add feature or remove reported bug.** Writes steps to add the developer for it (team member) and how code developments took place in team.

1. **Open Source Project Development:**Calculator in C: <https://github.com/desg/Calculator/blob/master/Calc.c>

Study and enhance the open source code on github. **Add feature or remove/Find reported bug** .Writes steps to add the developer for it (team member) and how code developments took place in team.

1. **Open Source Project Development: Get the digital** Clock code in Python [from github.com](https://github.com/Sifodeas/Clock)

Study and enhance the open source code on github. Writes steps to add the developer for it (team member) and how code developments took place in team. **Add feature or remove reported bug**

1. **Open Source Project Development**. :

Print Time in Various Format in C: <https://github.com/averello/clock/blob/master/clock.c>

Study and enhance the open source code on github. Writes steps to add the developer for it (team member) and how code developments took place in team. **Add feature or remove/Find reported bug**

1. **Open Source Project Development**. :Matrix Permutation in Python: <https://github.com/alexaylwin/PythonProjects>

Study and enhance the open source code on github. Writes steps to add the developer for it (team member) and how code developments took place in team. **Add feature or remove/Find reported bug**

1. **Open Source Project Development:**TicToe Game in Python: <https://github.com/bthesorceror/tictactoe-console/blob/master/main.py>

Study and enhance the open source code on github. Writes steps to add the developer for it (team member) and how code developments took place in team. **Add feature or remove/Find reported bug**

1. **Open Source Project Development** :Reverse a String in C++: <https://github.com/bnc119/reverse_string>

Study and enhance the open source code on github. Writes steps to add the developer for it (team member) and how code developments took place in team. **Add feature or remove/Find reported bug**

1. **Open Source Project Development**. :Bubble Sort in Python: <https://github.com/linuz/Bubble-Sort>

Study and enhance the open source code on github. Writes steps to add the developer for it (team member) and how code developments took place in team. **Add feature or remove/Find reported bug**

1. **Open Source Project Development**. : Palindrome Checker in Python: <https://github.com/fridahell/palindrome_checker> Study and enhance the open source code on github. Writes steps to add the developer for it (team member) and how code developments took place in team. **Add feature or remove/Find reported bug**
2. **\*VCS: Install and demo use hg mercurial as a vcs (create two users show merge and branch) and connect it bit bucket**
3. **\*VCS: Install and demo GUI of git as a vcs (create two users show merge and branch) and connect it github.**
4. **Project management tools: Install and demo Open project (bitnami) as a project management tools (create two users show merge and branch) and connect it github.**
5. **\*Licensing of open source: For** TicToe Game in Python: <https://github.com/bthesorceror/tictactoe-console/blob/master/main.py>

Study and enhance the open source code on github. Writes steps to add the developer for it (team member) and how code developments took place in team. **Add feature or remove/Find bug.**

Choose the suitable OSS license modified TicToe Game.

1. **\*Licensing of open source: For** any new Game (assume OSS python libraries are used) developed in Python by you.

Choose the suitable OSS license Game, which will ethically correct and you can earn finance/money out of it.