1. **Demonstrate to use of Version Control System (Git offline: on local machine with multiple user).**

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. **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 trackingtool. How bug tracking improves the quality of code** (on answer sheet).

1. **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

1. **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 or adding feature to it.

Upload the contributed code.

1. **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 schoology/Moodle (on answer sheet)

1. **Create of RPM packages.**

(Multiple modules/code packaging of java).

Take utility source code (MPI/OpenMP/Latex/doxygen, 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 Debian package manager (on answer sheet)

1. **Create of RPM packages.**

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

Take utility source code (MPI/OpenMP/Latex/doxygen, 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 Debian package manager (on answer sheet).

1. **Create of Debian packages.**

(Multiple modules/code packaging of python).

Take utility source code (MPI/OpenMP/Latex/doxygen, 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 of Debian packages.**

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

Take utility source code 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. **Demonstrate the use/features 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. **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 trackingtool. How bug tracking improves the quality of code** (on answer sheet).

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

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

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

1. **Configure and demonstrate the use of FTP and Telnet.**

Show the imp steps and file name of configurations**.** (on answer sheet)

1. **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. **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 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 trackingtool. How bug tracking improves the quality of code** (on answer sheet)

1. Demonstrate use of bug tracking tool and create the docker image of that tool. Push that image.

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

Push that image.

1. Write a Docker File to pull the Ubuntu with open jdk and write any java application.
2. Demonstrate use of bug tracking tool and create the docker image of that tool. Push that image.

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

Push that image.

18.**Demonstrate the use/features of Project Management toolfor 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.

create the docker image of that tool. Push that image.

19. Create two applications in two different docker containers. Push those applications and run to show the communications between two dockers.

20.**Demonstrate the use/features of Project Management tool 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.

create the docker image of that tool. Push that image.

21. 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.

22.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. **With the help of Docker-compose deploy the ‘Wordpress’ and ‘Mysql’ container and access the front end of ‘Wordpress’**
2. **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. **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 a web application with simple web page containing login details and create a docker image of the application.(Use Apache Web server)**
3. **Run the Docker container from recently created image and run the container at port number 80 in host system.**
4. **Write a python program to perform arithmetic operations and create Docker image accordingly.**
5. **Run the Docker container with created image .**
6. **Create a simple web application using LAMP Stack on docker container.**
7. **Create a web application with simple web page containing login details and create a docker image of the application.(Use Ngnix Web server)**
8. **Run the Docker container from recently created image and run the container at port number 80 in host system.**
9. **Create a simple Hello-world python flask application and create the docker image of that Flask application.**
10. **Run the docker container from recently created image and run that docker container to 5000 port of host system.**
11. **Pull the LAMP Stack container from docker hub and host a web application of your own.**
12. **Create a Docker image of simple web application from using HTTP web server at port 5000 in host.**
13. **Create a docker image of simple login form using Flask on port 7000.**
14. **Create a docker image of simple login form using django on port 6000.**
15. **Create a container with ngnix web server and create one more container with mysql.**
16. **Create a simple web form to insert the records in mysql data base.**
17. **Mount any directory of host system to the container.**
18. **Write a Docker File to pull the Ubuntu with open jdk and write any java application.**
19. **Run a LAMP Stack Container at port 8080 and host media wiki site on native machine.**
20. **Write a C program to create singly linked list and containerize it.**
21. **Create a LAMP Stack container and host a web application** of your own.