**Experiment No. 3**

**Aim:** Sketch out and analyze architecture of Moodle cloud portal and Moodle cloud site and create different entities dynamically.

**Theory: Moodle Architecture**

Moodle is an open source **Learning Management System** (**LMS**). It can be used to deliver online learning in a variety of settings.

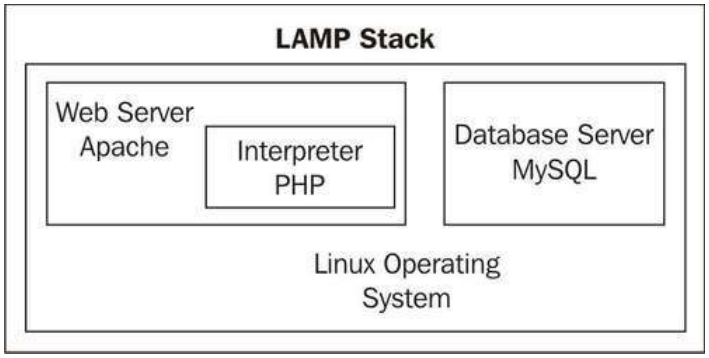
This Practical introduces some of the important concepts of Moodle architecture; how Moodle is structured and how Moodle works. This Practical introduces the following concepts:

- Components of the system (operating system, web server, PHP interpreter, database, and browser)
- Directory and system structure (Moodle code, database, and file storage)
- Installation (how to install, what happens during installation, and how to upgrade)
- Program execution (the major calling structure, included libraries, execution paths, and separation of function/display/data)
- Configuration (from the interface, from the config file, and from the database)
- Application Programming Interface (a brief description of what the major libraries do)
- Other common libraries (PEAR, ADOdb, YUI, and XMLDB)
- Access control for users, courses, and other security objects

**Understanding the stack**

Moodle is an example of a "LAMP" application. LAMP originally stood for Linux, Apache, MySQL, and Perl. Over time, the various components of the acronym have shifted. For example, PHP has become the predominate language for "LAMP" applications. In truth, any of the components can be exchanged for another. However, the title has stuck to refer to applications written in web scripting languages, using an SQL database to store information. With the increasing popularity of running open source web applications on both Windows and Mac OS X, two new terms have been coined, respectively: WAMP and MAMP.

See the following figure, which illustrates components of the system in Moodle:



Moodle is written in PHP, and the current version as of this writing (Version 1.9) requires PHP Version 4.3.0 or higher. Version 2.0 of Moodle, which is currently in development, will require PHP Version 5.2.8 or higher.

### Database

Moodle's database layer is written using the PHP ADOdb library, which was created to provide a standardized method of accessing various database systems, using a consistent programming interface.

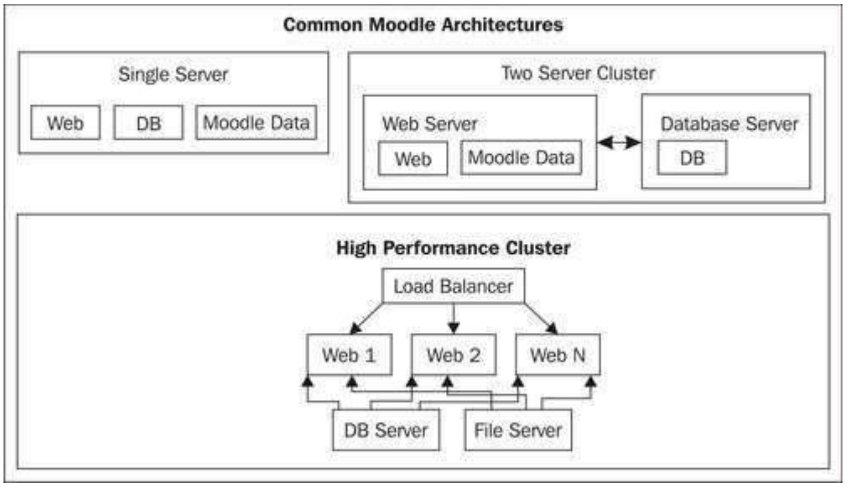
### Operating system

Moodle can run on any operating system that supports the required version of PHP and the database. Moodle is generally installed on one of the three major operating systems: Windows, Mac OS X,or Linux (or Unix/Unix-like operating systems).

### Web server

Moodle will typically work with any web server that supports running the appropriate versions of PHP.

Moodle as a web application has support and development constraints that are different from the norm. This is due to the way in which Moodle is used.



Moodle, as is common with the standard PHP application model, scales at particular points. The first scaling point is the database, which can easily be moved to a separate physical server. After that, we can bring in additional frontend web servers by using a load balancer. When using multiple web servers we will also need shared storage for our Moodle data. Session data can be stored in either Moodle data or in the database. The database server is the point where Moodle's scalability is most limited.

### An overview of Moodle core

Moodle core provides all the infrastructure necessary to build a Learning Management System. It implements the key concepts that all the different plugins will need to work with. These include:

**Courses and activities:** A Moodle course is a sequence of activities and resources grouped into sections. Courses themselves are organized into a hierarchical set of categories within a Moodle site.

**Users:** In moodle, users are anyone who uses the moodle system. In order to participate in course users need to be enrolled into course with a given role, such as:

- Students
- Teachers

**Course enrolment**:

- Enrolment gives user the possibility to participate in course as a student or teacher.

**User functionality in moodle:**

- User roles in moodle: Roles assigned to users give them a set of capabilities in given context. For example: Teacher, Student and Forum moderator are examples of roles.
- User's capabilities in moodle: A capability is a description of some particular Moodle feature. Capabilities are associated with roles. For example, mod/forum: reply post is a capability.
- Context: A context is a "space" in the Moodle, such as courses, activity modules, blocks etc.
- Permissions: Permission is some value that is assigned for a capability for a particular role. For example, allow or prevent.

**The most important plugin types**

Please see the page Plugin types for the full list. A selection of the most common ones is highlighted here.

**Activities and resources**

Activities and resources are the most basic individual components that make up a course and are the main tools for teaching and learning. Some examples of resources are: pages, links and IMS content packages. Examples of activities include: forums, wikis, quizzes, and assignments.

Activities are by far the largest type of plugin in terms of amount of code. A forum or wiki system could be a software project in its own right. Both activities and resources are installed in the mod folder.

**Blocks**

Blocks are small bits of interface functionality that can be added to (normally the sides of) pages. Many blocks provide additional views of data stored and modified elsewhere. Blocks live in the blocks folder.

Administrators can also manually change any of the standard user interface strings if the terms used in the installed language pack are not appropriate.

**Course formats**

Control how the structure of the course, a sequence of activities grouped into sections, is presented to the users. Course formats live in the course/format folder.

**Authentication plugins**

Control how users log in. Moodle can manage usernames and passwords itself, or use those stored in LDAP or another database. Alternatively, Moodle can use a number of single-sign-on schemes. Authentication plugins live in the auth folder.

### Enrolment plugins

Control which users are enrolled in which courses. Again this can be by synchronizing with another system, perhaps a student information system, or it can be tracked internally by Moodle. Enrolment plugins live in the enroll folder. (Moodle was created by an Australian, so enroll is the correct spelling

### Repository plugins

Ways for users to get content (files) into Moodle, either by uploading from their hard drive, or by getting the file from another location on the Internet, perhaps Drop Box, Google Docs, or Flickr. Repository plugins live in the repository folder.

### Steps of Performance:

Goto:

--https://moodle.com/moodlecloud/

**--Click on Tab Mentioning:** Try all of the features of Moodle Cloud free for 45 days.

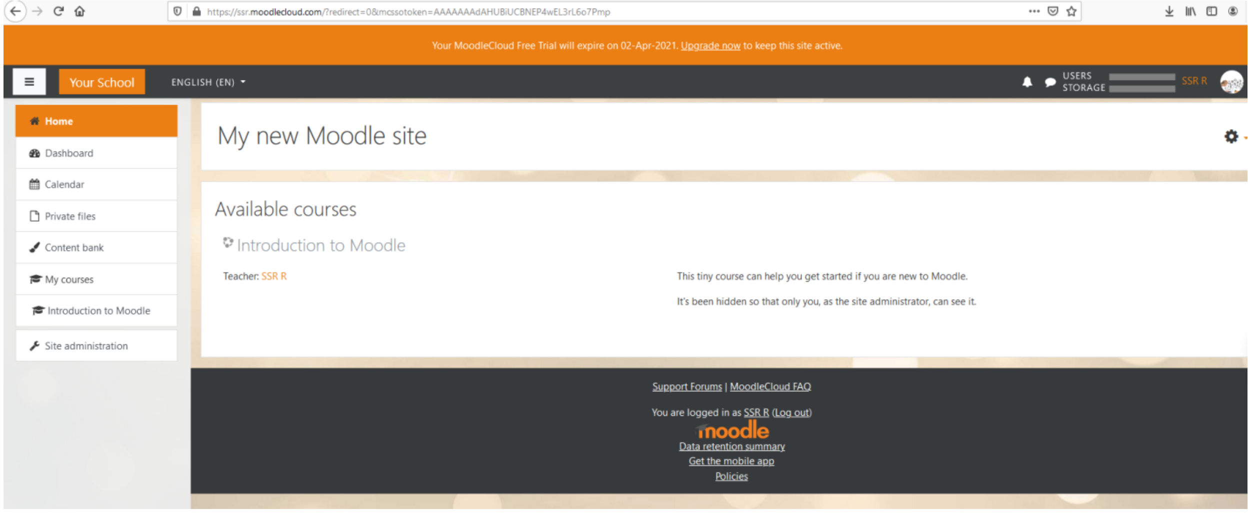
--Create a New Account

--Select (Tick) for all the options and Select Next

--Setup your Moodle Cloud site (5 Steps)

--Fill all the information given

--Moodle Cloud Site is ready and hosted



https://ssr.moodlecloud.com/?redirect=0&mcssotoken=AAAAAAAdAHUBiUCBNEP4wEL3rL6o7Pmp

Your MoodleCloud Free Trial will expire on 02-Apr-2021. Upgrade now to keep this site active.

My new Moodle site

Available courses

Introduction to Moodle

Teacher: SSR R

This tiny course can help you get started if you are new to Moodle.

It's been hidden so that only you, as the site administrator, can see it.

Support Forums | MoodleCloud FAQ

You are logged in as SSR R (Log out)

--Creating Courses on Moodle Site:

--Go to Site Administration

--Select and click on Courses

**--Courses**

- Manage courses and categories
- Course custom fields
- Add a category
- Add a new course
- Restore course
- Course default settings
- Course request
- Upload courses

--Select Manage courses and categories

--Click on Create New Category

--Select Parent category as Top

--Name Category name as Studying English

--Select the Option as Create Category

--Studying English Category is created

**--Select create a new Course**

--Enter Course full name as English Grammar

**--Enter all the details such as:**

--Course short name

--Course category

--Course visibility

--Course start date

--Course end date

--Course ID number

--Description

--Course summary

**Other Options Available are:**

--Course format

--Appearance

--Files and uploads

--Completion tracking

--Groups

--Role renaming

--Tags

--Select the Option Save and Return

--English Grammar Course is successfully created and added in the Category Studying English.

**Conclusion:** Hence, I have studied the architecture of Moodle and created an account on cloud portal and created different entities such as Category and Courses.