Team 8

Project Moodle

Abhinav Pratap Singh (180010001), Rishit Saiya (180010027), Vibhuti Raman (180010040)

Objectives:

- It helps in the smooth hosting of courses. Faculties can post assignments and students can evaluate the same and the same can be graded as well.
- The Moodle application will have three modes of login: student login, administrator login, and staff login. The administrator has the most authority.
- Whenever a new event or notification is posted students are automatically notified of it by an Email/SMS.
- It also informs the staff and gives them a list of students with high marks and those below standard attendance on a single button click.
- It also includes a student grievances area where students may post college inconsistencies and other problems. These posts will only be visible to the administrator.
- Only staff, administrator have the permission to post events. Students can view those notifications and react accordingly.
- The Moodle should have provision for adding courses, faculties, and students.
- Discussion forum for discussing on topics pertaining to course should also be present.

Since the implementation for the notification posts by Email/SMS required a paid API, we rather posted it on the dashboard itself.

The implementation is inspired from seeing different Moodle cloud facilities, the main open source Moodle project and our college Moodle.

We have concentrated on the key aspects of the moodle.

PROGRAMMING LANGUAGES/TOOLS

WEB DESIGNING:

- HTML5 - Basic Design and other features like buttons, textbooks, etc.
- Bootstrap CSS – For further styling and more enhancing and appealing look of the web pages used in software.

SERVER SIDE SCRIPTING LANGUAGE:

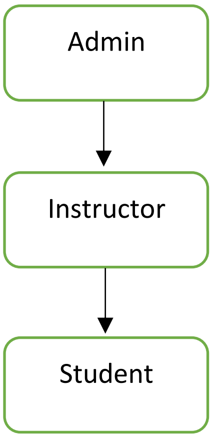
- PHP – The scripting side language we have used, to interact with the database.

DATABASE:

- MySQL – The default features of the database such as engine, etc. have been unaltered.

HIERARCHY

The admin is the **apex** user of the software system. It has nearly all the data of the courses as in course's instructor, student undergoing the courses, etc.



ACCESS TO MOODLE

Admin is added to database initially.

The formalities of an Instructor/Student:

- It is mandatory to register as an initial requirement to get yourself added in the database by administrator.
- Later to access our own account in the moodle, you have to use your registration credentials to sign in.

Note: The database is dynamically being changed as the new user of the moodle is registering himself/herself. That implies, its credentials and designations is being saved in the database as he/she registers.

DASHBOARD

The dashboard of the student and instructors' are different. However as per the hierarchy, the instructor must possess more user credibility.

Instructor has the additional features to add the assignments, projects details as per dates, etc. and also to view the files uploaded by the students, download them, grade them, etc.

And can see students with high marks and low attendance.

Student will have a complete access in its own Dashboard in terms of uploading projects, assignments, putting on queries, reading instructor's announcements, etc.

The Dashboard of the student will have options to view all the courses pages, view the notifications sent by the instructor, etc.

Admin, as mentioned above, will have all the data regarding the instructor, student course details.

IMPLENTATION FILES AND ROLES

We haven't bifurcated the frontend and backend files as we have implemented HTML5 and Bootstrap CSS in PHP files itself. Here PHP files also possess the backend code.

- addFaculty.php, administrator_addfaculty.php: These php files add the faculty after registration to the database.
- addStudent.php, administrator_addstudent.php: These php files add the student after registration to the database.
- home.php: This file is student's dashboard page.
- login.php: This file is for login portal.
- logout.php: This file is for logout. Here we have included session out, so that after logging out, user would not like to get redirected back to the last page where it left off.
- notification.php: This file is for adding notification from teacher to database.
- studentprofile.php: This file is to show the personal details of the student. It's like your own profile.
- teacher.php: This file is for the instructor's dashboard.
- teacherlab.php: This file is for showing the details of the files which have been uploaded by the student. This is mainly for the purpose of instructor.
- validation.php: This file is used to check whether the user logging in is a student or a faculty of college.
- administratorhome.php: This file is for the dashboard of administrator. Since it has different functionalities from the students and instructors', it had to be created separately.
- course_i.php: These are the dashboard pages of the respective course_i. Here i = 1, 2, 3.
- fileopen.php: This file is for opening the submitted student's assignments/projects files, tar, and zip.
- highmarks_lowatt.php: This file redirects us to the details of students having high marks (>=80%) and low attendance (<=80%). This is represented in the form of a table.
- lab.php, upload.php: These files have been used for a page whereby the students will upload their assignments & projects.

NETWORK/SERVERS TO RUN SOFTWARE

The functionality of the software is tried and tested on **localhost**. So it is compatible on localhost network provided all the files and proper database functioning on local system.

We used MySQL incorporated with XAMPP for DBMS. The server used is Apache server. Other features of MySQL databases pertaining are set default as taught in lab sessions.

DBMS

The database we used in XAMPP is named as ‘sessionpractical’ on our local machines.

The tables in ‘sessionpractical’ are ‘signin’ & ‘files’.

signin table column description:

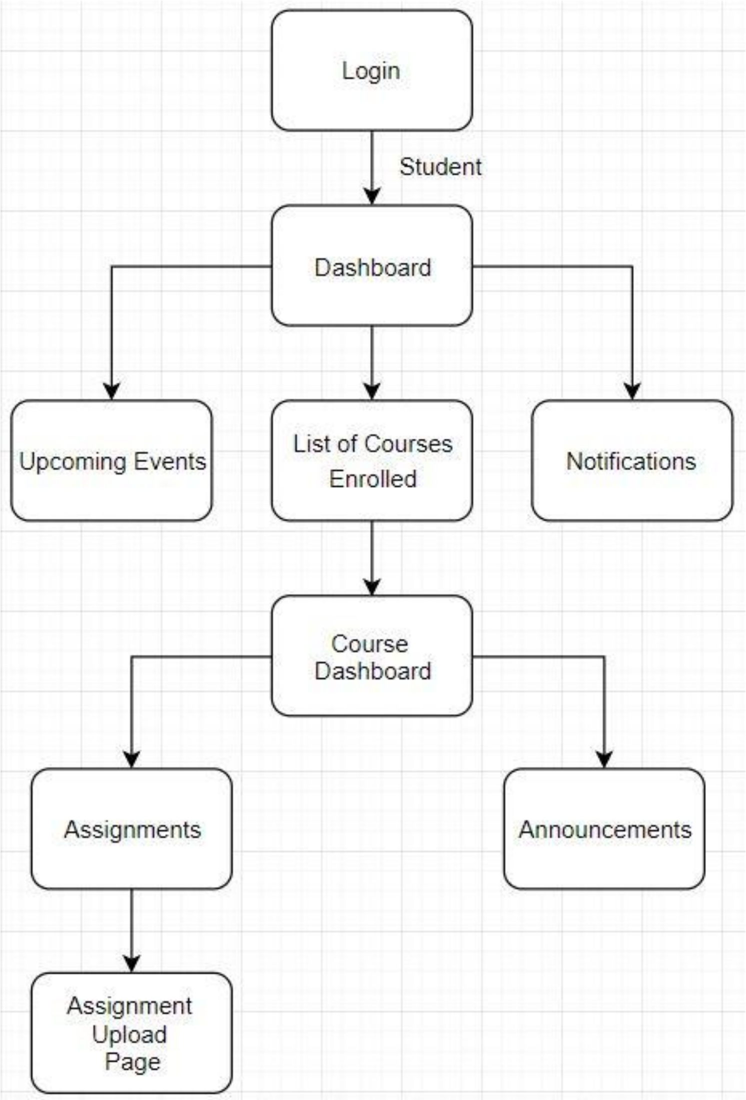
- id : Storing the serial number
- name : Storing the username of the user
- password : Storing the password of the user
- designation : Storing the designation of the user
- course_1 : Storing the course name
- marks_course1 : Storing the marks in Course1
- attendence_course1 : Storing the attendance record of the student in course1
- course_2 : Storing the course name
- marks_course2 : Storing the marks in Course2
- attendence_course2 : Storing the attendance record of the student in course2
- naam : Storing the name of the user
- image : Profile picture of the user
- notification1 : For storing the notification posted by the teacher
- announcements1 : For storing the announcements posted by the teacher
- discussions : For storing the feedback given by the student

files table column description:

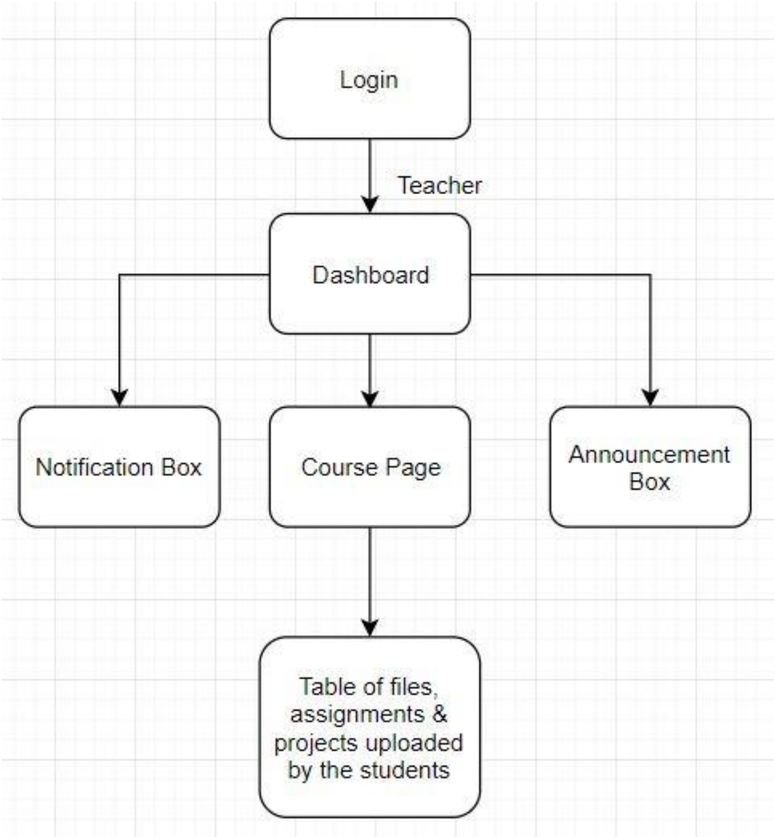
- id : Storing the serial number
- filename : Storing the name of file
- time : Stores the time of submission
- name : Stores the username of the user who submitted the file

MODEL-VIEW-CONTROLLER

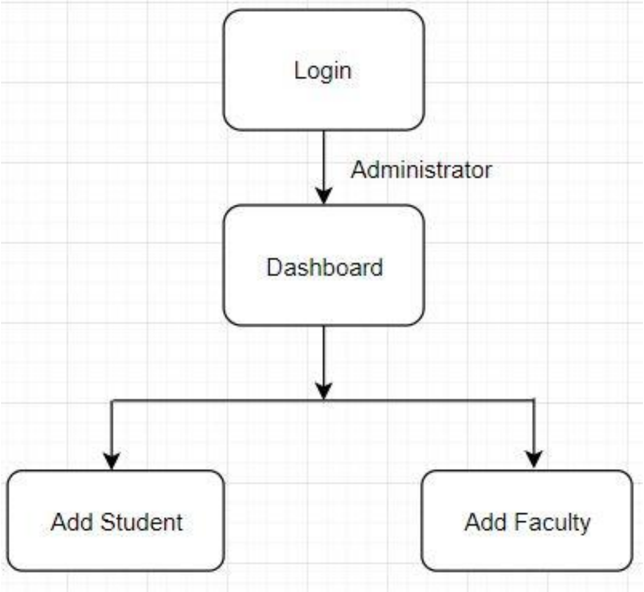
- For the **students**, the following MVC describes the flow between the files and pages respectively.



- For the **instructors**, the following MVC describes the flow between the files and pages respectively.



- For the **administrator**, the following MVC describes the flow between the files and pages respectively.


Login
Administrator
Dashboard
Add Student
Add Faculty