

TERM PROJECT

**ONLINE LEARNING MANAGEMENT SYSTEM**

SUBMITTED BY

**PURVA RAJWADE**

**POONAM THAPA**

PROJECT SUBMITTED TO

**Dr. JIANGPING CHEN**

SCHOOL AND COURSE

**UNIVERSITY OF NORTH TEXAS**

**5717- DYNAMIC WWW**

**CONTROL STRUCTURES**

EMAIL

[purvarajwade@yahoo.com](mailto:purvarajwade@yahoo.com)

[poonthapa@gmail.com](mailto:poonthapa@gmail.com)

**Team Participation:**

|  |  |
| --- | --- |
| Task | Work done by |
| 1.1.1 Objectives & Scope | Purva |
| 1.1.2 User Requirements | Purva |
| 1.2.1 Business rules | Purva |
| Entities & attributes | Purva |
| 1.3 Data Dictionary | Purva and Poonam |
| 1.4 ER diagram | Purva |
| 1.5 SQL queries | Purva |
| 1.6 Website Description | Purva and Poonam |
| 1.7 Usability considerations | Poonam |
| Future enhancements | Purva |
| PPT preparation | Purva and Poonam |
| Project  Deliverable | 12/09/2015 |

**1.1.1 Objectives of the Project:-**

The general objective of this project is to provide various courses to the students as well as others who are interested in learning online. The use of the internet has been increased and so the courses are available to be learned online. By providing this option, it would be easy for the users to learn by sitting at the comfortable place and learning based on the time they have.

The courses will be added, deleted, etc. by the admin. This project consist of the front end and back end sides. The front end side of the website is developed for the general public who are interested to know about us and to find out the courses provided by the management system. The back end is for the users who have created their account and admin utilities.

**Scope of the Project:-**

For this project, to store all the data various tables and attributes are required. As the main idea of this project is to provide courses for the students to enroll and learn, it will also allow the instructors to teach courses. Thus the scope of this project is limited to the following tables: COURSES, USERDETAILS, STUDENTXREF and INSTRUCTORXREF.

**1.1.2 User requirements:-**

Following would be the basic user requirements-

1. Users should be able to view the Home page, About Us and login/Register page of the website.
2. Users should be able to search for the courses in which they are interested.
3. Users will be asked to enter the user type as: Student, Instructor while creating the account.
4. Users who have been marked as Student will be allowed to enroll for the available courses and will be displayed the courses in which they are already enrolled.
5. Users marked as Instructor will be able to teach the course which is not taught by anyone else. They will be able to see the courses which are teaching.
6. Admin type of user will be able to add or delete courses as well as delete instructor from the courses.

**\*Note – User can be general public without creating an account, student, instructor and admin.**

**1.2.1 Business Rules:-**

Below are some basic business rules-

1. One student can enroll for many courses.
2. Every single course can be taken by many students.
3. One instructor can teach many courses.
4. One course will be taught by only one instructor.
5. User can be student or instructor based on the details entered while creating account.

**Entities and its attributes:**

In this proposal, there are basically 4 entities. The description for each is given below-

1. **USERDETAILS Entity –**.When a student or instructor wants to enroll/teach for any course, first of all they will have to create an account on the website by using login page. All the details entered while creating an account will be stored in this table. This table will consist of the attributes as User id, user lastname, user firstname, user type (it can be student or instructor) and password. The primary key of this table will be user id. Once the user has created an account, they will be able to know about the courses.
2. **COURSE Entity –** The course table will store all the details related to the courses which are offered. This table will have following attributes Course id, Course name, course department, course description, course start date and course end date. Course id will be the primary key for this table. Based on the above attributes, students can decide which course they are interested. The instructors will also be able to know about the courses which are already taught by some other instructor and the courses available for them to teach.
3. **STUDENTXREF Entity-** This table will act as a reference table between COURSE and USERDETAILS table. The attributes of this table are student id, course id and user id. The primary key of this table will be student id.
4. **INSTRUCTORXREF entity –** This table is also a reference table between COURSE and USERDETAILS table. If the user type is instructor then the details for that user id will be stored in this table. The attributes of this table are instructor id, user id and course id. Instructor id will be the primary key for this table. Here the course id will be unique as 1 course cannot be taught by multiple instructors.

**1.3 Data Dictionary:-**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TABLE NAME** | **ATTRIBUTE NAME** | **CONTENTS** | **TYPE** | **FORMAT** | **REQUIRED** | **PK Or FK** | **FK REFERENCED TABKE** |
| **USERDETAILS** | USER\_ID | User ID | INT(11) | 999 | Y | PK |  |
| USER\_EMAIL | User email address | VCHAR(30) | xxxxxxxx | Y |  |  |
| USER\_PASSWORD | User password | VCHAR(30) | xxxxxxxx | Y |  |  |
| USER\_NAME | User name | VCHAR(30) | xxxxxxxx | Y |  |  |
| USER\_DOB | User date of birth | DATE | mm-dd-yyyy |  |  |  |
| USER\_ACCTYPE | User account type: Student/instructor | VCHAR(30) | xxxxxxxx | Y |  |  |
| **COURSE** | COURSE\_ID | Course id | INT(11) | 999 | Y | PK |  |
| COURSE\_NAME | Course name | VCHAR(30) | xxxxxxxx | Y |  |  |
| COURSE\_DEPT | Course Department | VCHAR(50) | xxxxxxxx |  |  |  |
| COURSE\_DESCRIPTION | Course description | VCHAR(500) | xxxxxxxx |  |  |  |
| COURSE\_STARTDATE | Course start date | DATE | mm-dd-yyyy | Y |  |  |
| COURSE\_ENDDATE | Course end date | DATE | mm-dd-yyyy | Y |  |  |
| **STUDENTXREF** | S\_ID | Student id | INT(11) | 999 | Y | PK |  |
| USER\_ID | User id | INT(11) | 999 | Y | FK | USERDETAILS |
| COURSE\_ID | Course id | INT(11) | 999 | Y | FK | COURSE |
| **INSTRUCTORXREF** | I\_ID | Instructor id | INT(11) | 999 | Y | PK |  |
| USER\_ID | User id | INT(11) | 999 | Y | FK | USERDETAILS |
| COURSE\_ID | Course id | INT(11) | 999 | Y | FK | COURSE |

**1.4 Entity-Relationship Diagram:-**

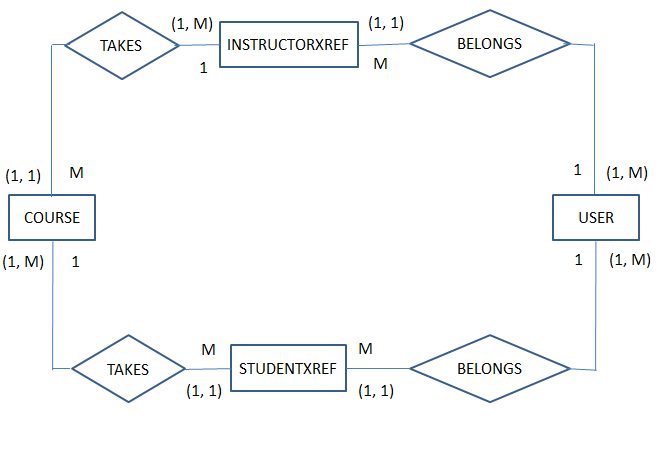


Fig 1: Entity relationship diagram for Online Learning Management System.

**USERDETAILS** (**USER\_ID,** USER\_EMAIL, USER\_PASSWORD, USER\_NAME, USER\_DOB, USER\_ACCOUNTTYPE)

**COURSE** (**COURSE\_ID,** COURSE\_NAME, COURSE\_DEPT, COURSE\_DESCRIPTION, COURSE\_STARTDATE, COURSE\_ENDDATE)

**STUDENTXREF** (**S\_ID,** USER\_ID, COURSE\_ID)

**INSTRUCTORXREF** (**I\_ID,** USER\_ID, COURSE\_ID)

**1.5 SQL queries to create tables:-**

CREATE TABLE USERDETAILS(

USER\_ID INT(11) UNSIGNED NOT NULL AUTO\_INCREMENT UNIQUE PRIMARY KEY,

USER\_EMAIL VARCHAR(30) NOT NULL,

USER\_PASSWORD VARCHAR(30) NOT NULL,

USER\_NAME VARCHAR(30) NOT NULL,

USER\_DOB DATE,

USER\_ACCOUNTTYPE VARCHAR(30) NOT NULL);

CREATE TABLE COURSE(

COURSE\_ID INT(11) UNSIGNED NOT NULL AUTO\_INCREMENT UNIQUE PRIMARY KEY,

COURSE\_NAME VARCHAR(30) NOT NULL UNIQUE,

COURSE\_DEPT VARCHAR(50) NOT NULL,

COURSE\_DESCRIPTION VARCHAR(500) NOT NULL,

COURSE\_STARTDATE DATE NOT NULL,

COURSE\_ENDDATE DATE NOT NULL

);

CREATE TABLE STUDENTXREF(

S\_ID INT(11) UNSIGNED NOT NULL UNIQUE AUTO\_INCREMENT PRIMARY KEY,

USER\_ID INT(11),

COURSE\_ID INT(11),

CONSTRAINT `USER\_ID\_FK` FOREIGN KEY ( `USER\_ID` ) REFERENCES `fa15grp1`.`USER` (`USER\_ID`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `COURSE\_ID\_FK` FOREIGN KEY ( `COURSE\_ID` ) REFERENCES `fa15db6`.`COURSE` (`COURSE\_ID`) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE INSTRUCTORXREF(

I\_ID INT(11) UNSIGNED NOT NULL UNIQUE AUTO\_INCREMENT PRIMARY KEY,

USER\_ID INT(11),

COURSE\_ID INT(11) UNIQUE,

CONSTRAINT `IUSER\_ID\_FK` FOREIGN KEY ( `USER\_ID` ) REFERENCES `fa15db6`.`USER` (`USER\_ID`) ON DELETE CASCADE ON UPDATE CASCADE,

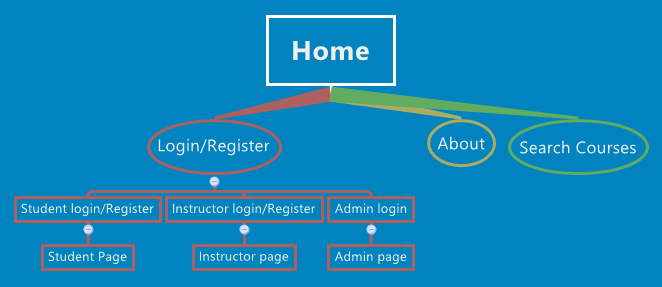
CONSTRAINT `ICOURSE\_ID\_FK` FOREIGN KEY ( `COURSE\_ID` ) REFERENCES `fa15db6`.`COURSE` (`COURSE\_ID`) ON DELETE CASCADE ON UPDATE CASCADE

);

**1.6 Website Description:-**

Online learning management system is a system which provides students to learn about the various courses online. Various pages have been created for the users like Home Page, About Us page, Login/Register Page and Search courses page.

Sitemap for the online Learning management system is as:



The project consist of Front end and back end. The front end pages are described below:

**1.6.1 Home Page:**

This page is the main page of the project. This page gives the brief description about the management system and also posts the latest courses which are going to be added in future. This page will be linked to all other pages. Whenever the user is interested in learning any course, then they need to create an account by clicking on the Login/Register button provided on the navigation bar.

**1.6.2 Login/Register Page:**

The login page lets user enter their username and password to login to their account. There are

three different links for students, instructors and admin in the login/register page. Students can

click on the enroll/register for courses link in order to login into the account. Students who are

already registered can put their username and password to login to the account. In case of new

users, they can register for a new account by clicking Join us button. In order to sign up for an

account a user needs to enter an email address, preferred username and password and date of

birth. The user has to make a selection of the account type either teacher or student from a drop

down menu. Then they can simply login into their account

The login page lets user enter their username and password to login to their account. There are three different links for students, instructors and admin in the login/register page. Students can click on enroll/register for courses link in order to login into the account. Students who are already registered can put their username and password to login to the account. In case of new users, they can register for a new account by clicking Join us button. In order to sign up for an account a user needs to enter an email address, preferred username and password and date of birth. The user has to make a selection of the account type either teacher or student from a drop down menu. Then they can simply login into their account.

**1.6.3 About Us Page:**

This page describes the goal of developing this project. This will make clear to the users what the purpose of developing this website is and how they can use it. Contact information is also provided for the users so that if they have any doubts, they can send an email with the query.

**1.6.4 Search course Page:**

When an user is interested in learning any course, before creating an account he can check whether the course in which he is interested is provided by the management system or not. This search option will help them to know about the courses and then they can decide whether to enroll in those courses or not. While using search option, users are not required to create an account.

The back end part will come in picture when the user has created an account. User can be student or instructor. While creating an account, user will be asked to select User type from drop-down option. The pages at the backend side are described below:

**1.6.5 Student Page:**

This page (student.php) will be displayed to the user whose user type is Student. It will display the courses which are available for a particular student to enroll. He can select multiple courses for enrolling. Checkboxes are provided in front of all courses and then finally enroll button is provided. Once the student selects any course and clicks on Enroll button, he will be shown a thank you message for enrolling in those courses. Student.php will call enroll.php when an enroll button is clicked by the student. Back button is provided on enroll.php which will re-direct student to student.php. The student will also be displayed the courses in which he has already enrolled on student.php.

**1.6.6 Instructor Page:**

When a user selects user type as instructor, he will be directed to instructor.php page. On this page he will be displayed with the courses which he is already teaching. The instructor will be able to teach courses which are not being taught by any other instructor. Checkbox option will be provided to the instructor to select courses and then he can click on teach button. This button will direct the instructor to teach.php page where he will be displayed a thank you message.

**1.6.7 Admin Page:**

Admin page is designed for the administrator to add and delete courses in the system. In order to login as an admin the user needs to select the admin-Add/Delete courses link from the Login/Register page. Two options appears in the admin page 1.Add a course and 2.Remove a course. Add a course lets the admin add new courses in the system. The details the admin needs to provide in order to add a course are course name, course department, course description, course start date, and course end date. The admin can simply press the Add Class button. Success message appears in the page after the submission of the details. In order to remove a class the admin can make a selection from the drop down menu and press Delete Class button. The success message appears afterwards.

**Functions provided by the website are:**

1. Following are the functions provided by online learning management system:
2. Browse homepage, about us page.
3. Search courses without creating an account.
4. Interested users can create account.
5. Students can enroll in courses.
6. Instructor can teach courses which are not being taught by any other instructor.
7. Admin can add/delete courses.

**1.7 Usability considerations:**

The online management learning system website has a logical navigation. Users can easily navigate around the webpage. The home page gives the users information about the whole system. All the contents are written in simple language. It has a user friendly interface and functions well with no broken links. All the links and sub links are clearly represented. The procedures to login to the account, register for courses, and add or remove the courses from the system are very clear. The system lets the users know if their attempts of adding courses, removing courses were successful. When the users put wrong credentials while logging in, it lets users know their attempt has failed because of the wrong username or password. So, that user can put correct information and login to the account. The website has all the required functions based on user types. That makes the website informative and user centered.

**1.8 URL of the website:**

The website for the online learning management system at the front end can be accessed at:

<http://coolt-courses.lis.unt.edu/slis5717/Purva/trial2/HomePage.html>

To access the website as an admin, below are the details:

Username: admin

Password: admin

**FUTURE ENHANCEMENTS:**

1. Based on the entries in database, graphs can be created to provide information about the number of students in each course.
2. Class entity can be added to provide more information about courses like class room, class timings, etc.
3. Change the template to make it more attractive and user friendly.
4. Passwords can be stored in an encrypted way to avoid giving away personal information.