

Available online at <http://www.mecs-press.net/ijeme>

Design and Implementation of a General Web-based Course Teaching Management System

Yuying Jiang, Zhan Huang, Zhanhong Huang

Department of Computer Science Jinan University Guangzhou, China

Abstract

A general and comprehensive course teaching management system is developed, so that different teachers in different colleges may use this system to manage their courses and teaching work. The system's front-stage provides users with login, query and search capabilities, and students can query related learning resources, assignment submission, scores, and attendance. The backstage offers effective management from colleges to curriculum subjects and provides teachers with management of teaching resources, student attendance, assignments, and notice board. Teachers and students can use the message board for communication and question answering. The PHP is used as the core programming language to manipulate the MySQL database, and Html, CSS and JavaScript are used for designing the web pages.

Index Terms: Course teaching website; PHP programming; MySQL database

© 2012 Published by MECS Publisher. Selection and/or peer review under responsibility of the International Conference on E-Business System and Education Technology

1. Introduction

Course teaching Web sites built by school or teachers are to convey the teaching and learning activities. The purpose is to realize the reform of teaching methods, ways of presenting teaching content, learning styles and teacher-student interactions, and to give full play to the advantages of information technology, offering a rich and colorful learning and development environment for students [1, 2].

However, at present most of the course teaching websites are strongly thematic with monotonous subject, poor website content and poor website administration autonomy, and lack of interaction between teachers and students [3].

In view of the above phenomenon, this paper discusses design and development of a general Web-based course teaching management system, which can manage multiple subjects from different colleges, majors, and teachers, and provides flexible website administrations and teacher-student interactions.

Main features of the system are: 1) rich teaching materials to facilitate student learning; 2) flexible Web site administration for colleges, majors, teachers, and courses; 3) online assignment submission and evaluation, and attendance checking, etc.; 4) uploads and downloads of related materials; 5) message board or guestbook for teacher-student interaction and question answering.

* Corresponding author.

E-mail address: 313jyy@163.com, thz@jnu.edu.cn, zhhuang@jnu.edu.cn

The system is developed by PHP plus MySQL, where PHP is the core programming language and MySQL is the web database.

2. System Functional Structure

The system consists of two parts: the front-stage and the backstage. Functional structure of the system is shown in Fig.1:

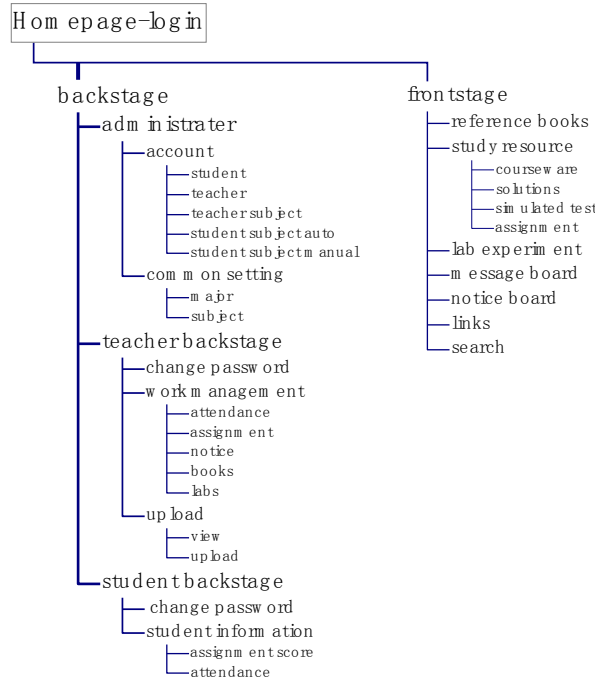


Figure 1. Functional structure of the system

The system's main functions are as follows:

With the front-stage: Users (administrators, teachers, students) can 1) authenticate through the login module to achieve different permissions for different users on the site; 2) view the latest announcements, most popular reference books and search the website information; 3) download leaning/teaching resources; 4) use the message board features to communicate.



Figure 2. Homepage of the system

With the backstage: 1) the administrator can add, delete, and modify information of colleges, majors, courses, students, teachers, teacher-course and student-course information. 2) Teachers and students can change password. 3) Teachers can add, delete, modify course attendance records and manage student attendance. 4) Teachers can add, delete, modify the course assignments and manage the students' assignment submissions and scores. 5)

Teachers can view, add, delete, and modify announcements and reference books. 6) Teachers can upload files, and view, delete, or modify the uploaded files. 7) Students can view the assignments and classroom attendance.

3. System Implementations

A. Database Design

Eighteen MySQL datatables are designed for storing the related website information, as in Fig.3:

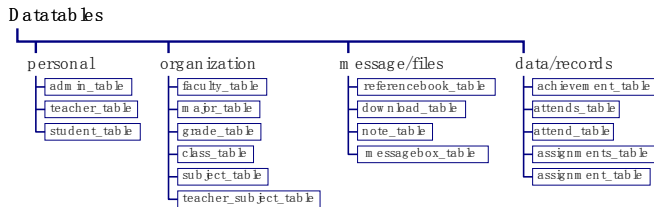


Figure 3. Datatables for the system

The achievement_table is for attendance and assignment management in the teacher backstage; admin_table is for administrator account and password; assignments_table is for assignment information set by teachers; assignment_table is for assignment information completed by the student; class_table is for class information; download_table is for data uploaded to/ downloaded from the database; attends_table is for the attendance information of the day; attend_table is for the student’s attendance of a course; faculty_table is for the college/department information; grade_table is for grade information; major_table is for major information; note_table is for the latest news of electronic courseware, simulated tests and assignments; messagebox_table is for messages in the message board; referencebook_table is for details of reference books; student_table is for details of students; subject_table is for details of subjects; teacher_subject_table is for correspondence between teachers and subjects; and teacher_table is for details of teachers .

Structures of typical data tables are shown in Fig.4.

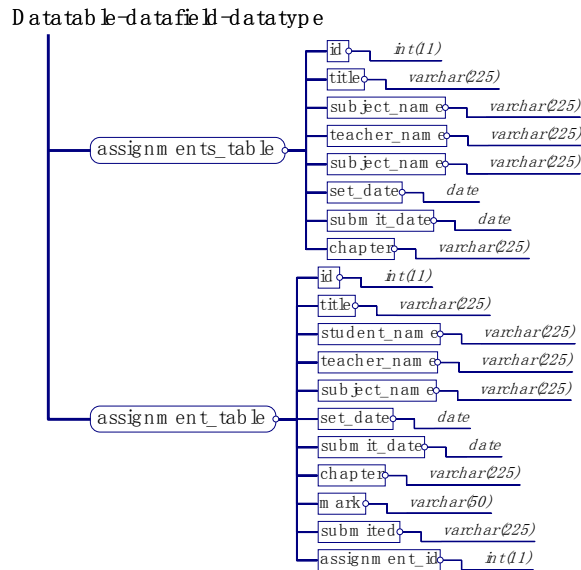


Figure 4. Sample datatable structures

B. Global Data Structure

PHP provides a large number of predefined variables to any script which it runs. PHP provides an additional set of predefined arrays containing variables from the web server, the environment, and user input. These arrays are automatically available in every scope. For this reason, they are often known as "superglobals"[4].

Some frequently used "superglobals" are described as follows:

1) `$_GET`

`$_GET` is an associative array that will get the variables passed into a script through the URL. For example if a script is called as `http://localhost/a_script.php?name=John` then `$_GET['name']` would equal John. You can also pass multiple variables through the URL. For example `http://localhost/a_script.php?name=Marry&age=34` would put the two keys 'name' and 'age' into the `$_GET` superglobal variable with values 'Marry' and 34. `$_GET` variables cannot exceed 100 characters.

2) `$_POST`

`$_POST` is an associative array that will get the variables passed into a script through HTTP POST. If a form's method is post the `$_POST` variable will get the variables in that form but will not show them in the address bar. This is useful for when a page is passing variables that it would not want the user to see such as a login password. `$_POST` variables have to limit to the number of characters they can contain.

3) `$_FILES`

`$_FILES` is an associative array that will get the files uploaded into a script through the HTTP POST method. If you are uploading files to a script with the `type="file"` input then your form element must have an attribute `enctype="multipart/form-data"`. Without this the upload form will not work. The `$_FILES['file']` array is a multidimensional array with the subkeys 'name', 'type', 'size', 'tmp_name' and 'error' where *name* is the name of the file, *type* is the application type of the file (e.g. `application/x-httpd-php`), *size* is the size of the file in bytes, *tmp_name* is the temp name of the file on the server and *error* is the error code of the file.

4) `$_SESSION`

A `$_SESSION` is an associative array which is used to store the session variables. Every time `session_start()` is called it will start a new session or continue the current session that is stored in the clients browser and initiate the `$_SESSION` variables defined. Every time `session_destroy()` is called all the keys of `$_SESSION` are unset and the browsers session is closed.

C. Implementations of Typical Modules

1) *The Login Module*

The login module (Fig.5) involves the following pages: login.php, check.php, logout.php and welcome.php, there is also a `$_SESSION` variable.

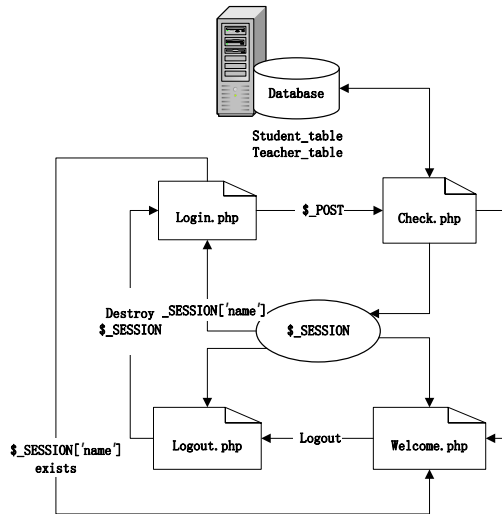


Figure 5. The login module

First of all, login.php uses the `$_POST` variable to transfer the user’s account, password, and permission to check.php, then check.php queries teacher_table or student_table according to the user’s permission.

If the user presents in the database, then check the user’s password. If the password is incorrect, return to the login.php. If the password is correct, then create `$_SESSION`, store the user’s information in `$_SESSION`, and jump to welcome.php. Welcome.php gets the login information through `$_SESSION` and displays it.

Finally, if the user needs to log out, then press “log out” to link to logout.php, which will use `session_unset()` and `session_destroy()`; These two PHP functions are used to destroy the `$_SESSION` variable to logout and return to the login.php.

2) The Notice Board Module

The notice board module (Fig.6) mainly related to the following page: index.php, note_list.php, shownote.php, note.php and error.php, there is also a `$_SESSION` variable.

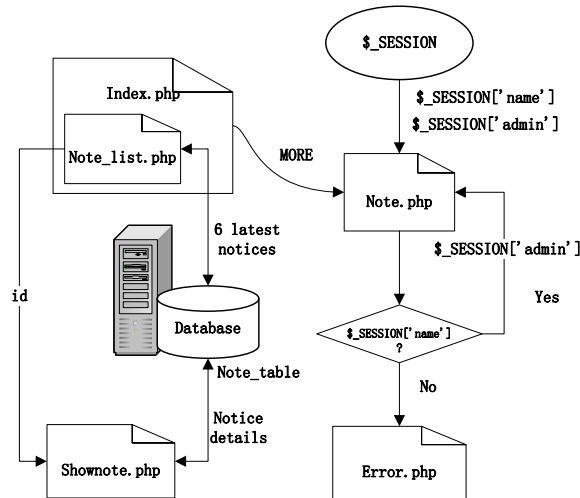


Figure 6. The notice board module

note_list.php embedded in index.php enables users to view the latest announcements. First of all, note_list.php will query the database note_table by using the SQL order instruction to sort the query results in descending order, and six of the latest announcements will be shown in note_list.php. Every announcement has an id parameter that marks its presence, and the id parameter is sent explicitly to the shownote.php by the get variable when the user click on a recent announcement and the shownote.php will display the corresponding announcement details by query the note_table according to the id value.

If the MORE-button next to the latest announcements is pressed, then the browser will jump to note.php. note.php gets the user name and user permissions from \$_SESSION['name'] and \$_SESSION['admin']. The PHP function, isset(\$_SESSION['name']) is used to determine whether a user is logged in, if not, the error.php will notify the user to sign in. \$_SESSION['admin'] is used to determine whether the user has write-permission, since students can only view the public notice and cannot write a notice. The value 1 of \$_SESSION['admin'] is for an administrator, 2 for a teacher and 3 for a student.

In note.php, three values of the parameter “action” are used: list, create, save for listing, creating and saving functionality.

3) The Assignment Management Module

The assignment management module (Fig.8) mainly involves the following pages: admin.php, header.php, assignments_list.php and assignment_list.php, there is also a \$_SESSION variable.

If the user selects “assignment management” in header.php which is embedded in admin.php, the browser will jump to assignments_list.php. This page uses five values of the parameter “action”: “list”, “create”, “edit”, “save” and “del”, to represent the functions of listing, creating, editing, saving, and deleting of assignments.

Initially the default value for the “action” in assignments_list.php is “list”, to display the assignment information. First, assignments_list.php gets the user name from \$_SESSION['name']; then queries the assignments_table by the user name, to get all the user's assignment information and displays them.

If the user selects “set assignment” (Fig.7), the parameter act=create will be passed to assignments_list.php and allows the user to set assignments.



Figure 7. The assignment management page

If the user presses the “edit” option, the parameter act = edit and the assignment id will be passed to assignments_list.php which will query the assignments_table according to the user name, \$_SESSION['name']. Then the system will use the id value to determine whether to allow the user to modify the assignments.

If the user presses the “delete” option, the parameter act = del and the assignment id will be passed to assignments_list.php and the SQL delete command will be executed to delete the assignment in the assignments_table.

The “save” function is startup after the user has finished the “create” or “modify” process. At this point, if it is at the “set assignments” stage, the parameters act = save, type = create and \$_POST variable (holding the update information) are passed to assignments_list.php, and SQL insert instruction will be executed to insert the new assignment record to the assignments_table; and if it is at the “modify assignment” stage, the parameters act = save, id, type = edit and \$_POST variable (holding the update information) are passed to

assignments_list.php, and SQL update instruction will be executed to update the corresponding assignment record in the assignments_table.

In addition, when the user clicks an assignment name in assignments_list.php, assignment_list.php will be in active and parameters $act = list$ and $assignmentid = \$fid$ (assignment id for the assignments_table) will be passed to it. Then assignment_list.php gets the user name from the $\$_SESSION['name']$ and queries assignment_table according to the $\$_SESSION['name']$ and $assignmentid$, allows the user to carry out assignment management operations and displays the results.

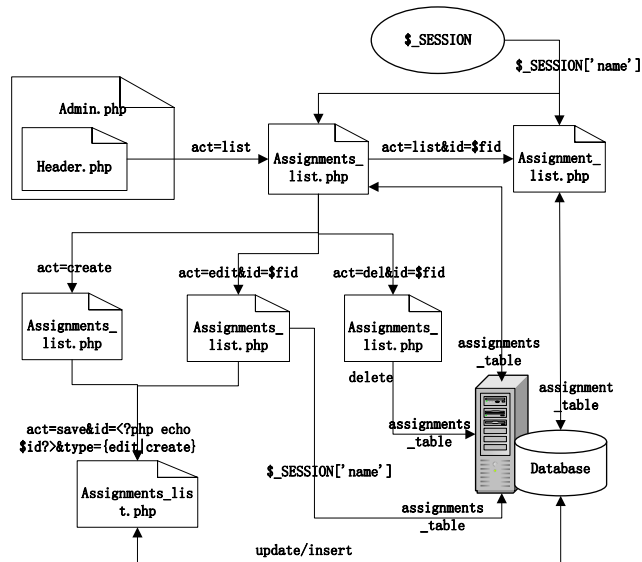


Figure 8. The assignment module

4. Conclusions

Our design goal is aiming at the system's adaptability. The system offers flexible management from college to curriculum subject and from teachers to students, so that different colleges or teachers in the University may use this system to manage their courses and teaching activities.

Now the system is running for management of a number of courses for different teachers, offering an effective teaching and learning environment for teachers and students.

References

- [1] Ronald D. Owston, "The World Wide Web: a technology to enhance teaching and learning?," Educational Researcher, vol.26, no.2, pp.27-33, 1997.
- [2] Sarah Horton, Web Teaching Guide: A Practical Approach to Creating Course Web Sites. Yale University Press, 2000.
- [3] Zhijun Wang, Xue Wang, and Xu Wang, "Research and implementation of Web-based e-learning course auto-generating platform," Edutainment 2008, LNCS 5093, pp. 70-76, 2008.
- [4] Luke Welling, Laura Thomson, PHP and MySQL Web Development (4th Edition). Addison-Wesley Professional, 2008.