INTERNET BASED LESSON AND TEST DELIVERY, AUTOMATIC GRADING AND RECORD KEEPING SYSTEM

Omer Farook, Chandra R. Sekhar, Jai P. Agrawal, Ashfaq Ahmed and Michael Holtz
Purdue University Calumet

ABSTRACT

The paper describes the “Internet Based Lesson and Test Delivery, Automatic Grading and Record Keeping System”. This system is conceived and designed as part of the Senior Design Project of Electrical Engineering Technology curriculum during a two-semester course offering. These two courses are the capstone courses in Electrical Engineering Technology curriculum offered in 7th and 8th semester. This project and similar projects provides the student a unique opportunity to design and integrate the knowledge and skills acquired through number of different courses.

The paper expounds in detail, the features from user’s perspectives and software design methodology. The system uses the HTML language as the GUI interface for users. All the data is passed using CGI forms and the Perl scripting language[1,2,3,5]. All necessary data is kept in one database with 5 tables within that database that holds information on the users, lessons, grades and students. For the interaction between the HTML forms and database, the SQL[6] programming language is used with an Open Database Connection (ODBC).

I. INTRODUCTION

This paper describes a project to write a program that could be used by teachers and students for mathematics classes. It takes the concept of the math lesson and puts it on the Internet for many to use. One computer or many computers connected to a server are able to use this software package. The teacher’s responsibility would be to provide the lessons for their class by using the provided problem builder with this program. The students can log onto a computer with their user name and view from their personal home page all previous lessons and scores. When the student is ready to take the lesson they click on the lesson button and they are ready to go. The student's lesson is automatically graded and submitted to the database. All grades are entered into the teacher's grade book for their use.

The software package uses the HTML programming language as the GUI interface for users. All data is passed using CGI forms and the Perl scripting language. All necessary data is kept in one database, with five tables within that database that holds information on the users, lessons, grades and students. For the interaction between the HTML forms and the database, the SQL programming language is used with an Open Database Connection (ODBC).

The MathComp (the name for our system) program involves both the student and the instructor together in the whole process. While teacher is involved with design of the lesson and the grades the student is involved with taking the given lesson and learning from it while getting experience with Internet applications.

II. SYSTEM DESCRIPTION

The project uses the Web to deliver math lessons to students. It can be installed on a single computer or on a server and enable access to many computers over a private network. This enables delivery of education on the anywhere- anytime basis.

The components of MathComp:

- The operating system runs on Microsoft Windows 2000 Professional Edition.

- The Web pages are served out using Internet Information Services(IIS) on Windows 2000. This service establishes the *InetPub* and *www.root directory* on the working drive of the computer/server. With the use of IIS the administrator can establish the security and view rights of all of the users within the system.

- Perl programming language is used to pass the data from one form to another form. Perl programs can be compiled on Windows, Unix or Linux operating systems. All of the HTML forms in this program are stored inside Perl scripts that are marked with their .pl extension. The HTML is encapsulated inside the Perl for data sharing among the forms. Using just HTML scripts limits you from passing anything beyond the current form. Perl enables the data storage through the use of use variables in memory. Later the data can be retrieved for reuse.

- Permanent storage and retrieval of data is accomplished through the use of a database. The Structured Query Language (SQL) allows us to perform this task. SQL can be implemented in Perl scripts and executed using an Open Database Connection (ODBC). For this project we have used MySQL for the database needs. MySQL is similar to Microsoft's SQL Server and Access except that there is no cost incurred.

- In order for the form to interact with the database, an ODBC connection had to be established. This is done through the Administrative Tools under the Control Panel. In this case, since MySQL is not supported by Microsoft in the ODBC drivers list we had to download one from www .mysql.com. Once the driver is installed a connection can be established successfully.

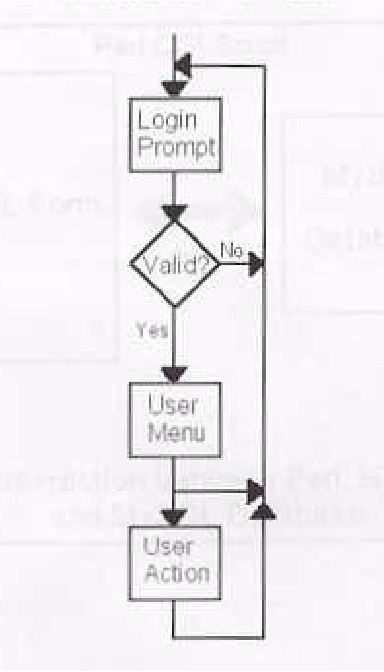
- We used a GUI interface with the MySQL database. In a normal situation, a MySQL database can only be accessed through the command prompt. Parsing a very large database through the command prompt is tedious because the lines wrap around each other and make viewing very hard. We have downloaded a GUI interface for MySQL databases from www.mysql.com and this put an end to this problem.

- We have used Javascript[4] for form error checking. This provides an audio and visual alert when a mistake is made while entering data into a form. JavaScript is also used in this project to add visual effects to buttons and icons.

2.1 PROGRAM OPERATION

- When the user starts up the program they are prompted with a login box. For this application a login box was necessary in order to find out, who, the users are and what can they access. We created following three accounts: administrator, instructor and student. When the user enters user name and password the form is submitted and the code queries the Users table to see if there is a matching user with the given information. If everything is correct, the user is granted access.

- The administrator is required to create instructors and students. The administrator account can also be used for database cleanup, deleting old user records to free up space in the database. The instructor is also an administrator of sort, but just for their group of students. They are responsible for entering in and maintaining lessons. The instructors are also responsible for adding students and maintaining student records throughout the year. When a student logs in they are taken to their own homepage where they can view past lessons and grades, this is also where they are given the option to take the next lesson.



.

Fig. 1 Flow Chart

-

.

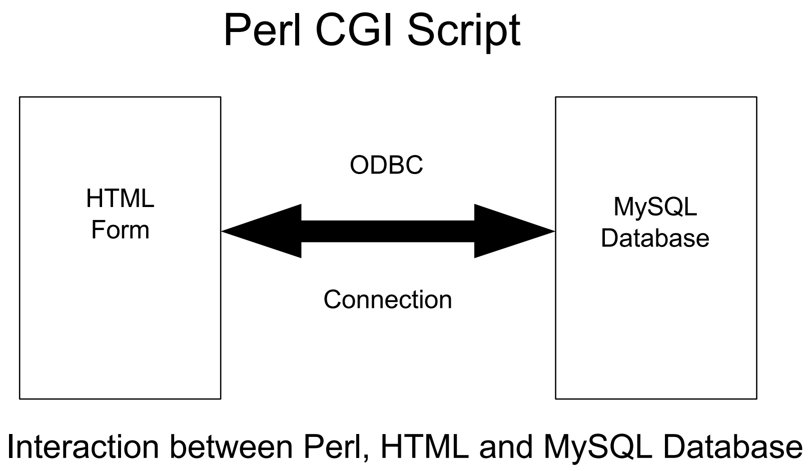


Fig. 2 System Operation Diagram

III. SAMPLE PROGRAM OPERATION

To better explain how this whole process works, we created two different examples of how a Perl program could be written. In the first example we wrote a very basic Perl program that asked for the users name and age. This program is run from the command prompt, so there is no graphics just text. Here is what it looks like:

Once the information has been entered, a subroutine is invoked to tell if the name entered is between A-M or N-Z and what age bracket you would fall into child, teenager, adult or senior citizen. The result looks like this:

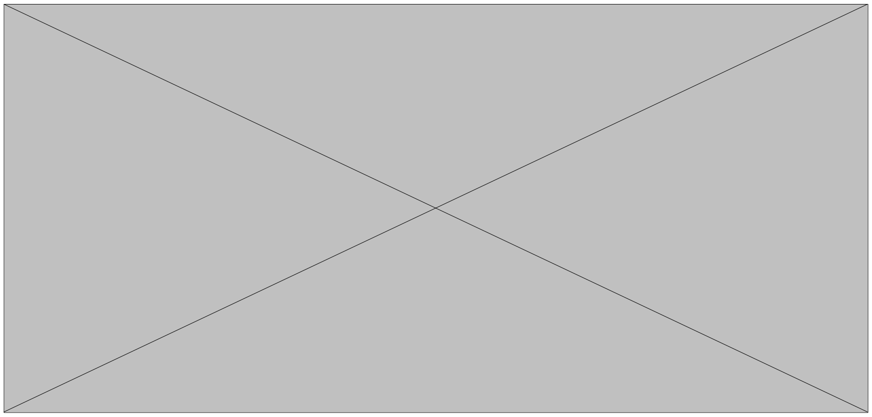


Fig. 3 Sample Command Line Interface

Depending on the choice made the program would either run again or by hitting the Enter key the Command Prompt would completely close down and end the program.

Another example of how this program could be written and this is how we used it for the project would be to use an HTML form to get the data from the user (Fig. 4). The advantages to using an HTML form would be that it could be executed from the Internet, graphics could be added to it to make it look more appealing to the user and finally it is a lot easier to use. Here is an example of the same kind of program that was used above except that it is done so using an HTML form:

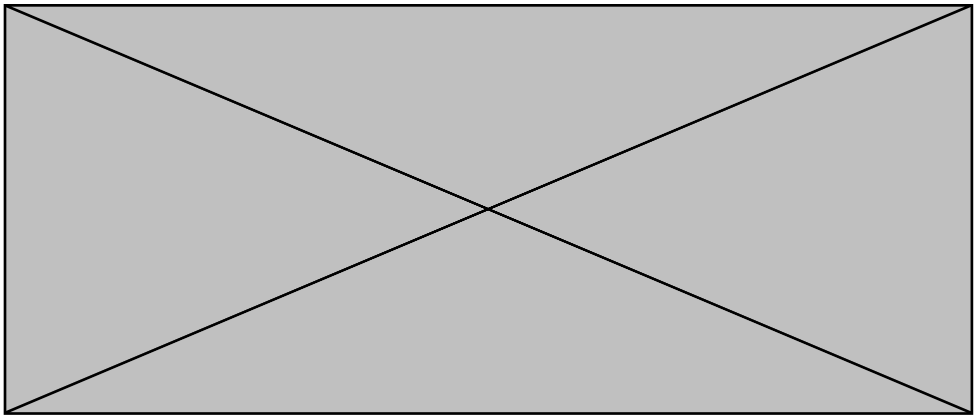


Fig. 4 HTML Form Interface

In Figure 4, the address bar depicts a Perl program that is being run and it is called create.pl. The last part of the address is telling the Perl script to run the create_user subroutine, which is shown because it prompts the user for the name. When the name is entered and the submit button is clicked the create.pl program will reload and a different subroutine is called that would insert the name into a table called USERS.

In order for the create.pl, HTML form to properly operate, it has to be executed from an Internet browser. If this program is compiled, and try to execute it from the command prompt the window would only pop up momentarily and none of the HtML form would be shown. A regular Perl program can be executed from anywhere on the computer, but when it has HTML tags it has to be operated from a browser.

IV. EXPERIMENTAL RESULTS

From the beginning we decided on how to proceed and what subsystem to choose in order to implement the project. The one part we were not sure about was what kind of database software to use. First we started planning this project we planned to use Microsoft's SQL Server 2000. We decided not to use it due to its cost. Microsoft Access was another choice which we had to give up since it had lot of problems while running a script that was inserting, deleting or updating any records from that database, it was getting locked up. In Access, if the page opened or another user was performing the same operation then an error would come up on the screen. Finally we found the solution to this in MySQL package. It was free to use and it was a very effective tool, the operating speed of the forms was a little quicker. We had to do some extra research on how to operate the

MySQL software, but it wasn't too hard to pick up on.

V. CONCLUSION

While writing the programs we wrote a lot of different examples early on and then later scrapped those ideas when we figured out that we didn't like them any more. By doing this we really helped ourselves find what we really wanted instead of forcing ourselves to a pre conceived charted plan. What really took us the longest to figure out was how we were going to layout the color and designs of the HTML forms. In the last 6 months we probably went through ten different styles that we wanted to use. To write programs to accomplish a particular task wasn't the hardest part. How we wanted to share and show with others, in other word interactivity with others was the difficult part in the end.

Some limitations that are associated with this project are that the security aspects on it needs some work done to it. Within the scope of this project, we could not have designed a fool proof design to make this application really secure. Another part of this project that we would have liked to make better is the type of lessons that could have been made. The ones that can be made in this application are using basic mathematics. With the time limitation imposed we couldn't get involved in making a way to create more complex mathematical problems.

VI. PEDAGOGICAL NOTE

This project was accomplished by a student for the Capstone course “Senior Design I & II”. These two courses are required in the Electrical Engineering Technology Curriculum. These two courses are taken in the 7th and 8th semester of bachelor degree program. In this project student was able to integrate different segment of competencies from his discipline into a viable commercial grade end product. The team concept was accomplished by student and faculty advisers pairing together this d provides an environment for Cooperative and Discovery Learning. The roles of faculty advisers were to facilitate and guide the student. At the same time being cognitive of the fact that faculty should shun from leading the student. These courses further provide the student a platform to engage in Interactive Learning. This process of self directed learning once triggered translates into a life long learning process which is both rewarding and intellectually stimulating.

References

[1] Activeperl user guide: http://tekweb.idi.ntnu.no/proginfo/activeperl/perltoc.html

[2] Perl by Example, Quigley, Ellie. Prentice Hall 1998.

[3] Programming Perl, Wall, Larry. O’Reily 1996.

[4] JavaScript: The Defintive Guide, Flanagan, David. O’Reily 2002.

[5] www.language.perl.com

[6] www.mysql.com

Biography

CHANDRA R. SEKHAR is a member of the faculty of the Electrical Engineering Technology at Purdue University Calumet. Professor Sekhar earned a Bachelor’s Degree in Chemistry from the University of Madras (India), a Diploma in Instrumentation from Madras Institute of Technology and Master’s Degree in Electrical Engineering from University of Pennsylvania. Professor Sekhar’s primary teaching and research focus is in the areas of Biomedical and Process Control Instrumentation and Clinical Engineering.

OMER FAROOK is a member of the faculty of the Electrical Engineering Technology Department at Purdue University Calumet. Professor Farook received the Diploma of Licentiate in Mechanical Engineering and BSME in 1970and 1972 respectively. He further received BSEE and MSEE in 1978 and 1983 respectively from Illinois Institute of Technology. Professor Farook’s current interests are in the areas of Embedded System Design, Hardware – Software Interfacing, Digital Communication, Networking, C^{++} and Java Languages.

JAI AGRAWAL is an Associate Professor with joint assignment in Electrical Engineering Technology and Electrical & Computer Engineering. He received his PH.D. in Electrical Engineering from University of Illinois, Chicago, in 1991, M.S. and B.S. also in Electrical Engineering from I.I.T. Kanpur, India in 1970 and 1968 respectively. Professor Agrawal has worked recently for two years in optical networking industry in the Silicon Valley in California. Professor Agrawal is the Founder Advisor to Agni Networks Inc., San Jose, California. His expertise includes optical networking at Physical and Data link layers, optical and WDM interface, SONET and Gigabit Ethernet and analog electronic systems. He is the author of a Textbook in Power Electronics, published by Prentice-Hall. His professional career is equally divided in academia and industry. He has authored several research papers in IEEE journals and conferences.

ASHFAQ AHMED is Professor of Electrical Engineering Technology Department at Purdue University Calumet. Professor Ahmed received his Bachelors of Science degree in Electrical Engineering from the University of Karachi in 1973 and Master of Applied Science degree in 1978 from University of Waterloo. He is the author of a Textbook in Power Electronics, published by Prentice-Hall. He is a registered professional engineer in the state of Indiana. He is senior member of IEEE. Professor Ahmed’s current interests are in the areas of Embedded System Design, C^{++} and Networking,

MICHAEL HOLTZ received his B.S. in Electrical Engineering Technology with an option in Computers, Telecommunications and Networking from Purdue University Calumet in 2003. He is currently working as an independent consultant in the area of robotics. His current interests reside in software design using C^{++} and in embedded systems. He has received numerous awards for the development of “Computer Assisted GOAL-Oriented Walking Robot” and various other semester honors.