Use of a Web Based System for Material Delivery and Collection.

Bob Fithen

Mechanical Engineering Department
Arkansas Tech University

Abstract

The author has created a web-based system focused on educational material delivery and collection. This was accomplished by writing a server-based web interface using the Perl computer language. Using this server based interface, students are required to enter a username and password thereby limiting access to the system to the intended audience. The system's main focus is delivery of educational materials as well as serving as a collection point for student assignments. Educational materials are delivered in several forms including, written material, grading codes and slide shows with descriptive audio. The objective of written material is the augmentation of the class textbook as well as other information pertinent to the course. The overall objective of slide presentations is expanding the set of problems accessible to the students while offering audio explanations. Another focus of the web-based system is homework and project collection. The collected material takes several forms, most notably Microsoft Word documents, Matlab codes and scanned information. Homework and tests are pushed into the system through the web interface and either graded immediately or stored for grading at a later date. In this paper the author will give a complete description of how the system was constructed. This description will include each part of the complete system as well as experiences with each aspect of the system. The author will detail the capabilities of the system at present, as well as, how the system will change to meet new demands over time.

Introduction

This paper describes a web-based system built by the author that accepts, grades, and tracks student's progress through engineering based courses. These courses fall into two categories, computer language intensive and general engineering courses. At Arkansas Tech University the computer language of choice for engineering students is Matlab. This web-based system has evolved from the author's experiences with an intensive computer based course. In this course, students are given assignments consisting of problem statements, code requirements and presentation specifications. Upon completing these assignments the professor is left with the task of determining if the algorithm itself is correct. In addition the professor must determine if the student has properly understood

the engineering theories behind the algorithm. In the past, the professor would determine the student's comprehension by observing written description, plots, data outputs from the assignments, etc. In this "*one-pass assignment collection*" method the student must complete an algorithm using Matlab and use this algorithm to develop plots and write interpretations of his or her results. The correctness of student's plot and written descriptions are contingent on writing a correct algorithm. Students are left with the task of concentrating on the algorithm itself and the interpretation of the results. However, concentrating on the results produced from an incorrect algorithm will yield the entire assignment incorrect. For this reason, the author began a project that grades the algorithm prior to any interpretations written by the student. In essence, this approach can be thought of as a "*two-pass assignment collection*" method.

General Approach

The "*two-pass assignment collection*" method is very well suited to web development. The web system has evolved through several distinctive phases each semester into its present form[1,2]. Originally the system was configured to accept Matlab code from the student and automatically grade the students code based on a set of predetermined specifications. One major problem with this approach became the requirement of the professor to write a grading code in Matlab that returned diagnostics information through the web interface. As a result, this posed a major security risk since in this approach student codes would be run on the system giving them rights to execute certain system level commands. Consequently this approach was scraped in favor of a more secure approach. In this new "*secure*" approach the professor provides a Matlab grading code to the student through the web interface. In addition to downloading this grading code the student must download a data file, which contains randomly generated input and appropriate output variables. These grading codes will read in both data types and call the students code several times testing their code against the given data. Once the student receives full credit he or she will return to an upload location in the web and submits their code to be graded by the professors grading code at a later time. This effectively decouples the grading process from the web system while still maintaining the advantages of the "*two-pass assignment collection*" method. Once the two processes were decoupled the web based system began to evolve on its own and entered usage in other engineering courses.

System Description

The entire web system is based on the Perl[3,4] programming language. In order to enter the system the student is presented with the screen shown in figure 1. Three entry must be completed, course number, username and password. Notice the lock,  , at the bottom of this web page and all following web pages. This symbol indicates a secure connection between the web browser and the web server thereby minimizing the possibility of intrusion into the communication. Upon entering the correct User Name, Password, and course number a perl code is executed on the server when the submit button is hit. This perl code will attempt to set three per-session cookies available for retrieval in the directory https://mengr.atu.edu/cgi-bin/Plywood/. These three cookies are username,

engineering course number and an encrypted version of the password. The password is encrypted via the crypt command. These three cookies will allow the user to maneuver around within the web site without reentering their information. These cookies will expire upon closing the browser; hence no one can access the system by re-launching the browser and searching through the history. After entering the correct information the browser is redirected to the main location as shown in figure 2.

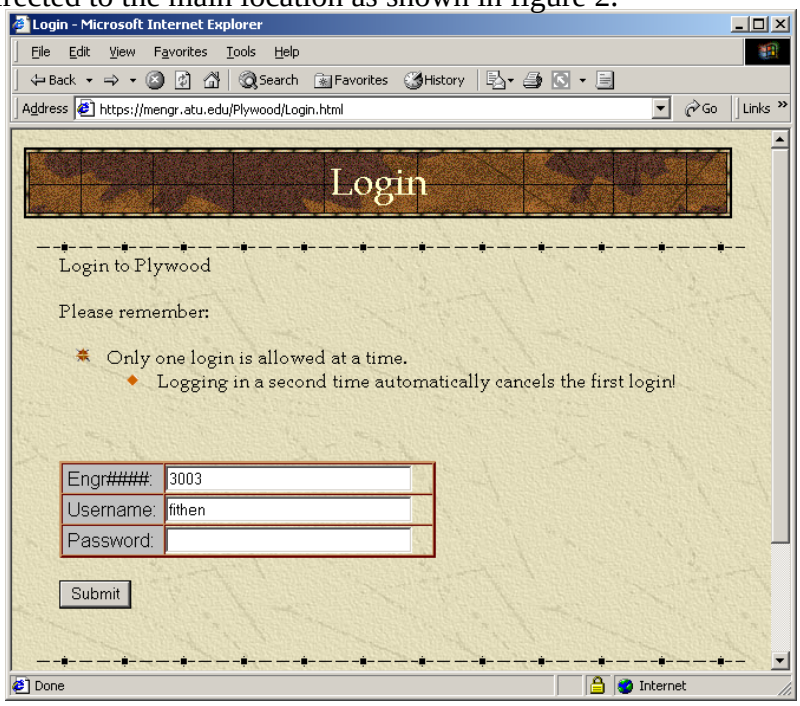

Figure 1. Login Screen.

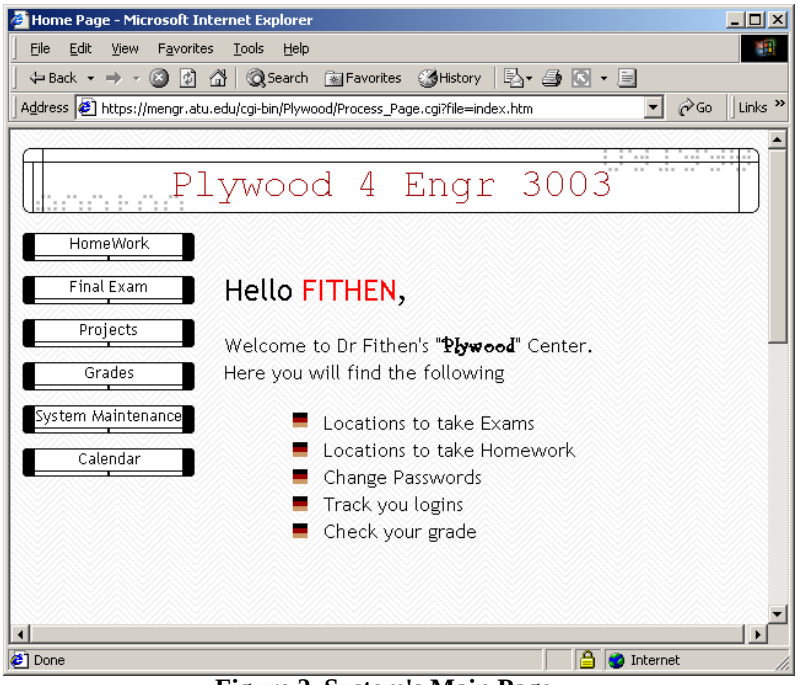

Figure 2. System's Main Page.

Within this main page there are several locations available for use/exploration. First on the list is a list of homework assignments broken down by chapters. Under the each chapter a page like that shown in figure 3 is given.

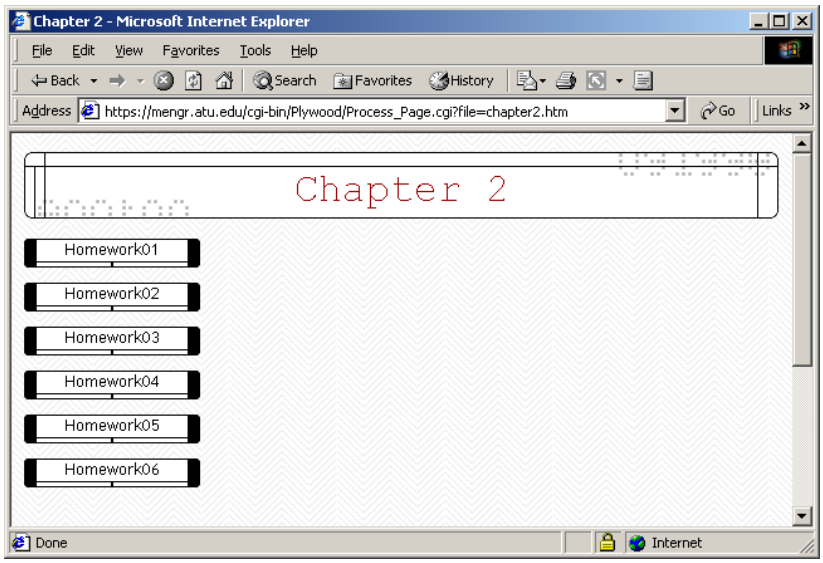

Figure 3. Individual Chapter Homeworks.

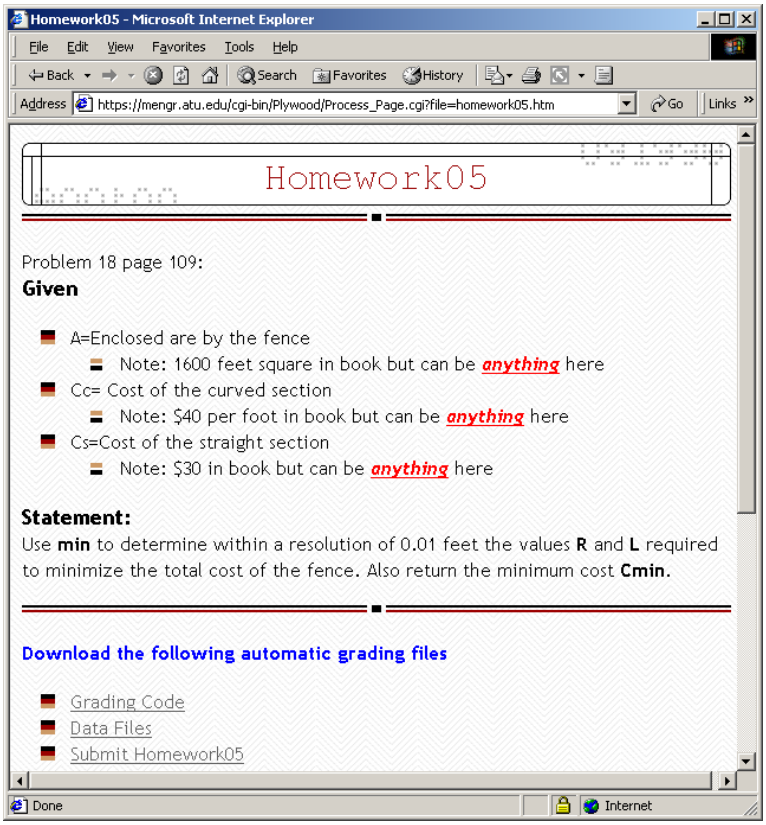

Figure 4. Homework 5.

A typical homework assignment is shown in figure 4. This homework is typical with respect to its layout. A problem statement in the form of givens is followed by information about the variables that are required. Near the bottom of the page there are three links, one for the grading code, one for the data file and one for a location to submit their homework. Since each student is identifiable by his or her own username, each submittal over the web is stored in that student's location on the web server. A typical grading code is shown in figure 6. In all grading codes the student's homework must be in the form of a subroutine. In this case the highlighted line shows the functional form of this subroutine call. One particular nice feature of Matlab is the format for passing variables. All variables suffixing the subroutine are input variables (A,Cc,Cs) and all variable prefixing the subroutines are output variables [Cmin,R,L,index]. For this reason the student will see the given/require text written on the web page matching the subroutine calls in the Matlab grading code.

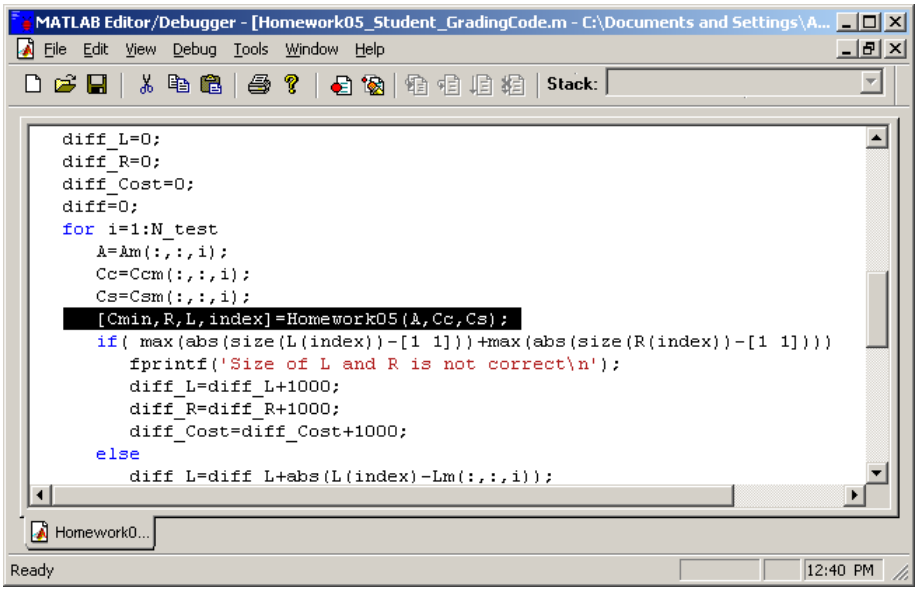

Figure 5. Typical Grading Code.

The final stage in the process is the submittal of homework by the students. This occurs through the bottom link shown in figure 4. Upon going through this link a web page like that shown in figure 6 is displayed. This page allows the student to browse their own file system and locate their assignment with the browse button to the right. Upon location of their assignment they can "Upload Selected File". Notice in red the information about the username, course number and the homework number. All this information is used to place the file in the appropriate location in the web server. Finally one of the most informative features of the web system, for the student's perspective, is online presentation of their own grade. Again the information passed through the cookies back to the system allows the server to identify each student and present only their own grades as shown in figure 7.

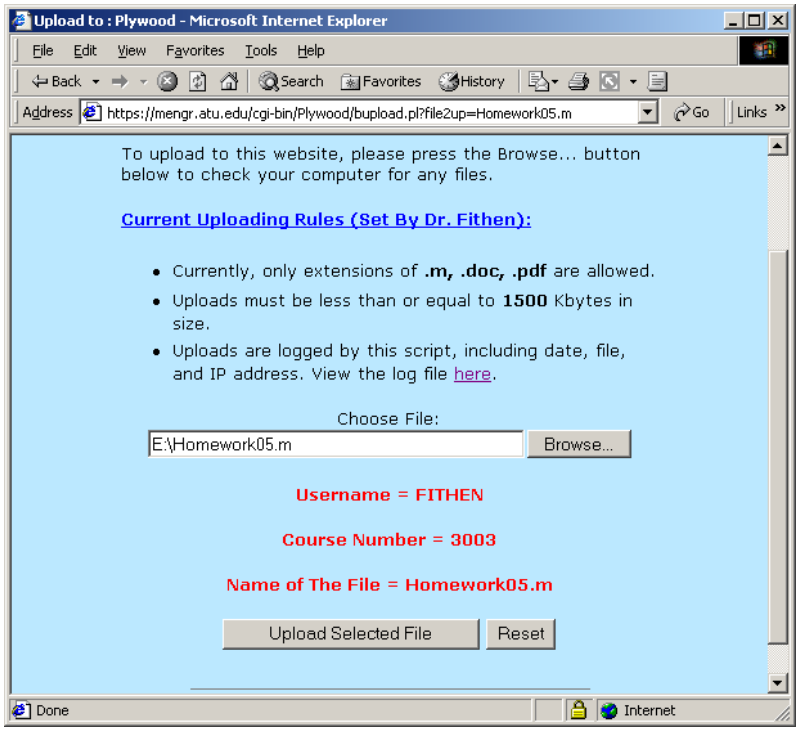

Figure 6. Upload Location in the Web System.

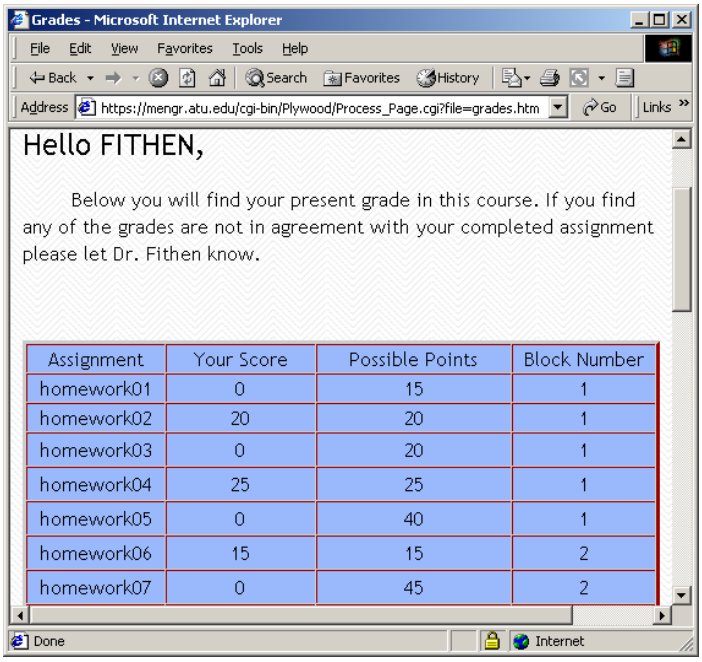

Figure 7. Grade Table.

Web Server Description

The web system is developed around a FreeBSD[5] system. However, the web system itself is an add on component using OpenSSL[6] and SSLmod[7] operating under the Apache[8] web server. All software packages are available freely on the web. This secure system was used primarily for protection of the student as well as protection of the professor's course material. Around this web server the entire system works within the context of the perl programming language and Microsoft's FrontPage. Once a professor develops a FrontPage web, the directory structure is stored in a location on the Linux machine not directly accessible by the web server. All files and directories contained within the FrontPage web must be readable by a Unix user named "web" since all web based cgi codes are run as user "web". Students enter the web system by visiting the web page shown in figure 1.

Students Data

One key to success of the system is its ability to reduce the bookkeeping required by the professor. To this end, each student is given his or her own location in the directory structure. This location will be class dependent and therefore the top-level directory will be, for example, *system_root*/ENGR3003. Users directories will be located in, for example, *system_root*/ENGR3003/*USERNAME.* In each students directory is a complete list of past submissions, grades, session id index, password, a specialized message tailored for each student, and a log file containing information about each login. In addition a directory in the structure contains the solution and a grading code for each assignment.

Summary and Conclusions

This paper described the evolution of a web-based system written by the author that accepts, grades, and tracks student's progress through engineering courses. For the courses that use Matlab as is central programming language, web pages are constructed with problem statements as well as links to downloadable Matlab grading codes, downloadable data files and a link to an upload page. Once these assignments are complete and upload by each student a grading code is executed on the web server by the professor which grades all the students assignment at once and records their grades in their own web based grade books. In order to protect the student's privacy as well as the privacy of the course content, the web pages are delivered through a secure connection. Through this secure web page, the author has written a server-based application interface using the Perl language. Through this interface the students are required to enter a username and password to enter the site, thereby limiting access to the site to the intended audience. Upon entering the site students can observe their grades, submit homework assignments, submit projects, and take exams.

Acknowledgments

The author wish to thank NSF for supporting this project under NSF project #9952284

References

1. Bob Fithen, *Construction of a Web System for a Computer Modeling Course*, ASEE Gulf-Southwest Section Annual Conference, March 28-30, 2001 in College Station, Texas.
2. Bob Fithen, *Building a Web-Based System for Automated Grading of Computer Modeling Assignments*, ASEE Annual Conference & Exposition, June 24-27, 2001 in Albuquerque, New Mexico.
3. Clinton Pierce, *Teach Yourself Perl in 24 Hours*, SAMS, 1999.
4. Larry Wall, Tom Christianson and Jon Orwant, *Programming Perl*, 3rd edition, Oreilly
5. FreeBSD Web Site: http://www.freebsd.org/
6. Open SSL Web Site: http://www.openssl.org
7. ModSSL Web Site: http://www.modssl.org
8. Apache Web Site: http://www.apache.org

BOB FITHEN
Dr. Fithen is an assistant professor at Arkansas Tech University. He received his B.S. in Mechanical Engineering from Louisiana Tech University, M.S. in Mechanical Engineering from Texas A&M University, and his PhD in Engineering Mechanics from Virginia Tech University. He spent four years working at General Dynamics, Fort Worth and a total of five years working in the research division of Wright Laboratories in Dayton, Ohio. Further information may be obtained at http://mengr.atu.edu/