

Automated Job Notification/Posting

Kevin Foote
CS-395

15.April.2003

Contents

1	Introduction	4
2	Project Overview	4
2.1	Data	4
3	Project Requirements	5
3.1	Functional Specs	5
3.2	System Specifications	5
4	System Design	5
4.1	Architecture	6
4.2	Security	6
5	Development Process	7
5.1	Testing	7
6	Results	7
6.1	Future Development	7
7	Summary Conclusion	7

Abstract

Job posting on the CS/MATH web site has become stagnant. It is not uncommon for job postings to remain on the site for well after the closing deadline. This project aimed to create an easily maintainable web-application that is capable of rotating job postings and is simple for an individual to both add and remove postings. This paper describes the program that was created to automatically rotate displayed job postings based on the date the posting was submitted.

1 Introduction

This software project was developed for the department of Computer Science at UAA. Often the CS department will receive notices of job opening from local businesses. Naturally, the department wants to make this information available to its students. Currently, these job postings are placed within a folder attached to the computer lab door. This is a very archaic method of making information available to computer science students. This project intends to clean up the lab door and display job notifications on the Internet.

The need to get this information published on the department web-site is evident. Along with posting active information on a web-site, it is customary to change content regularly. Many times, HTML updates get overlooked during the department's business day. These tasks are not "crucial" to daily operation but, rather, keep others informed and updated. Therefore, when a job is posted, it is helpful for the posting to only remain viewable for a given length of time.

Since I am employed by the CS department as a lab-technician, I already have scheduled hours to work. Not all of these hours are filled with maintenance issues, which leaves me time to work on projects such as this one. As a lab-tech, it is helpful to create useful software and scripts for the department.

2 Project Overview

The goal of this project was to develop a system that automatically rotates the web content of the jobs' HTML page. Automation of web content is a large part of keeping information updated and pertinent to current viewers. This software will assist the CS department in maintaining current information pertaining to local jobs within this field for students.

A simple, easily maintainable product was the design goal throughout this project. The rotating (transient) nature of the positions responsible for both using this software and maintaining it emphasize this. The resulting software was designed for use by any administrative assistant or student worker. This product also needed to be portable. Currently, the CS department web-site is hosted on a UNIX based system; in the future, it may not be.

2.1 Data

The data composition of a job posting is multiple text fields with headings that describe something about this position. These fields are simple text elements with the one unique data member being a date associated with the posting's creation. Since this data is naturally in text format, it seemed best to keep the data in its natural form rather than storing it in another format.

3 Project Requirements

The requirements for this project were loosely drafted using a suggestion by Kenrick Mock during Fall 2002 CS470. These requirements were initially adopted and then modified to suit a more basic web automation approach. During the course of developing this project, I arrived at the following specifications.

3.1 Functional Specs

Functionality for this software project was straightforward.

1. Postings should be viewable for 3 months
2. Postings should be kept on file for 9 months
3. All HTML forms should have reset buttons
4. Instructions or how-to should be included on secure page
5. Only authenticated user should have the ability to add/remove postings

3.2 System Specifications

The web-based nature of this project dictated that some derivative of web scripting be used. The scripting language that I chose to use was PHP. This language was available on the server and was simple to learn. The project was designed to run on an Apache web server using the mod_php module. Another reason PHP was chosen as the scripting language was its availability on a variety of systems.

4 System Design

The application design of this project was kept very simple. I broke the project into two parts based on accessibility. The job postings needed to be viewable by everyone, while the addition and removal of job postings needed to be restricted (fig 1). This was a natural division within the project scope, which led to easy data storage and manipulation.

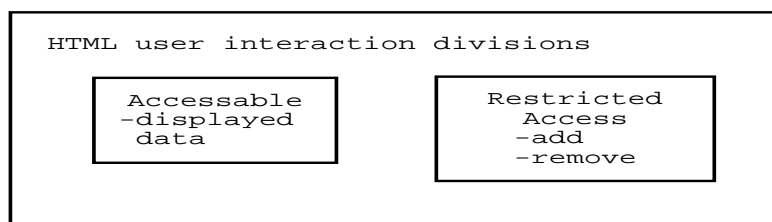


figure 1: (Access Division)

4.1 Architecture

The overall system was broken down into more specific parts, as seen in figure 2.

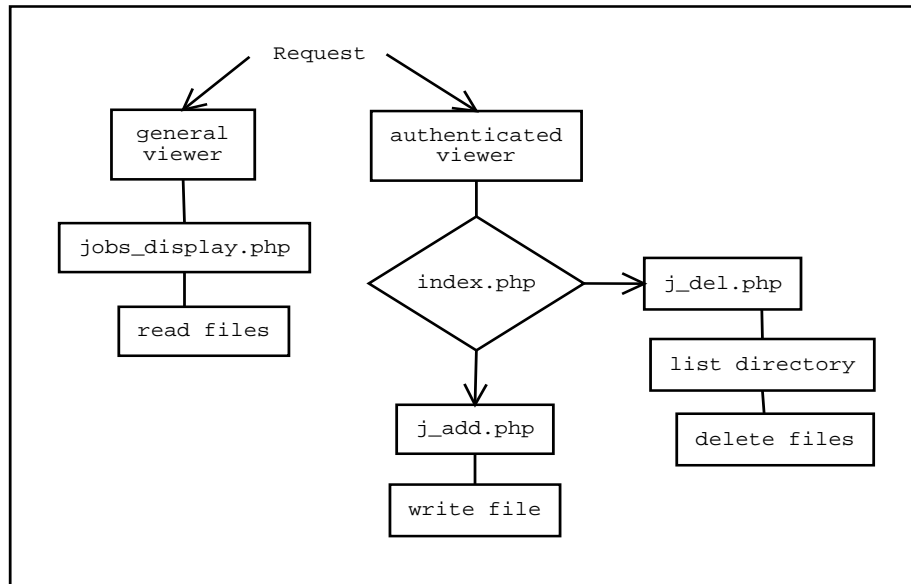


figure 2: (Architecture)

Upon an http request, the viewer is taken to either the general notification viewing area or the restricted administrative area. A viewing request will invoke a read access to the files contained within the jobs directory. The date of each file is checked and, if the date is within the calculation range, the file is displayed. A restricted request will require authentication. After access is granted to the restricted area, a viewer may choose to create additional notifications or remove a notification from the system.

Both options within the restricted area use web forms to obtain user input on what he or she wishes to accomplish. These web forms allow the necessary entry of data for the manipulation of current system data.

4.2 Security

The specifications of this system required there to be a method of restriction to access the functionality of creating and removing job postings. Since this system runs on the Apache web server, it made sense to make use of Apache's security settings. Apache implements per-directory security by using a combination of .htaccess and .htpasswd files. By placing the PHP scripts that modify the job data files into the jobs_adm directory, (which is password protected by

htaccess), a relatively secure system has been created.

The downside of skirting the security issue by use of htaccess is that the password is sent in plain text format. I felt this was an acceptable solution in respect to the traffic and use of this site.

5 Development Process

A combination prototyping/mock-up design approach was implemented throughout this project. When writing web-based applications, it is often beneficial to view what will be displayed to the end viewer. The defining factor in this project was to create something that would put the information in students' eyes.

5.1 Testing

A modified Black Box testing method was used. I would submit data to the system in various ways in an attempt to test whether or not these files were dealt with properly. In one instance, I created a number of test files (notifications) that were placed in the jobs directory. The resulting display produced improper calculations and an arithmetic date bug was found.

6 Results

The final result that emerged from this project is a working automatic job and internship rotation system. The back end web-based processing of each data request successfully delivers the current job information to the viewer, while the back end web forms deliver simple and direct functionality to an individual who maintains this information.

6.1 Future Development

I plan to implement further code modifications in this project. Currently, I would like to enhance the date rotation capability by providing the maintainer with the option to display all notifications for a specific length of time. I plan to implement this modification during the upcoming summer months.

7 Summary Conclusion

The automated job posting project was developed using PHP and web forms for the UAA CS department with the goal of creating a system that automatically rotates job notifications. The project was completed on time and incorporates the needed functionality of the system specifications.

Overall, this project provided a great hands-on learning experience for developing a useful web-based product. I was able to deepen my understanding of the need for project organization and keeping to a time frame. The added lessons of learning a new scripting language and better understanding web data transmissions all contributed to the overall educational experience.

References

- [1] Sklar, David & Trachtenberg, Adam (2002), PHP Cookbook. O'Reilly & Associates, Inc., Sebastopol, CA.