

Institutional Information Management with Open Source Software

Prof. M.S. Gaur (msgaur[at]mnit.ac.in)
Anant Narayanan (anant[at]kix.in)
Rahul Murmuria (rahul.murmuria[at]gmail.com)
Raghavendra Narasimhan V. (raghavn86[at]gmail.com)

Department of Computer Engineering
Malaviya National Institute of Technology
JLN Marg, Jaipur, Rajasthan, India

Abstract

The administrative responsibilities of any university are usually extremely complex and rather time consuming. Universities would benefit greatly by automating several of their routine tasks. Although there are several generic content management systems available, we will see how a customized solution for every university can be built using simple and free open source technologies such as PHP and MySQL, and also why it is better than using the generic solutions already available. In this paper, we describe a modular content management system. We particularly focus on how the system can be used to deliver information to their respective destinations in a transparent manner besides some other applications like library, hostel and academic management. The solution is not merely an Intranet, but rather an integrated system of services to manage and generate content.

1. Introduction

Several universities, particularly those in India, still follow conventional techniques to tackle administration. Routine tasks such as grade compilation and hostel room allotment are still done by hand. In instances where there is some kind of automation, it is found that the efforts are single-handed, local and unorganized. In this context, we identify three main problems:

- Manual handling of data is known to be error prone
- No strict time bounds and delays are recurrent
- No proper channels of communication between departments
- Major variation in the academic and other activities structure

Universities will benefit greatly from automation that will eliminate the mentioned problems. As we will see, developing and deploying such a solution is neither difficult nor costly, and requires minimal resources, resources that almost every university is guaranteed to have.

1.1 Prerequisites

First prerequisite for deploying such a system is the presence of a campus wide network, which may also be termed as the intranet. This includes the hardware required to run such a network; terminals, switches, and server machines. We also assume that all faculty members, staff and students are computer literate. By computer literate, we mean the ability to operate a computer and perform basic tasks such as word processing and using the internet. We believe it is reasonable to expect everyone to have this basic knowledge, and in circumstances where this is not true, arranging training sessions to impart this knowledge will not be very difficult.

The second prerequisite is that everybody involved in the functioning of the university must have some means of accessing the intranet. This includes the faculty, staff and the students. At the very minimum, every department of the university must contain terminals connected to the intranet

that are freely accessible by all those concerned.

The third prerequisite is the presence of a voluntary support team, or a team of system administrators that will be responsible for developing, maintaining and supporting the system. The team may consist of faculty members, staff and students as may be appropriate for the university concerned.

As mentioned before, we believe it is reasonable to assume that most universities will already have these requirements in place, since all universities maintain a website and have some sort of a campus network. The solution we propose is flexible enough to work on the most sophisticated of networks as well as on bare bone infrastructure. This makes the solution suitable for implementation in all universities irrespective of their size.

1.2 What will it provide?

A content management system intends to provide services for the uniform and accurate collection of content from various sources, and then providing a methodology for publishing that content as appropriate. For a system in place at a university, the CMS is only a part. There are several other modules that provide a variety of other services that work along with the CMS to bring complete automation to all tasks at the Institute.

We will be discussing the various modules that form the complete automation system. All these modules put together form a system wherein routine tasks are automated to a large extent. It is important for the modules to work together because all the modules share a single data source; i.e. information about the faculty, students, staff and the institute itself. We propose a method and implementation for achieving the above mentioned goals in this tutorial.

Some common myths about such automated systems exist. For example, one may assume that once the automated system is in place, everything is taken care of. This is wrong for two reasons: one is that for the system to generate any useful output, sufficient amount of information must be provided as in input. Secondly, as is the case with any software, the system required maintenance; for which a team of dedicated individuals will be required.

2. Motivating Example and Scenario

We take our own university as a case study. We will consider several aspects of university administration and describe the current, mostly manual, procedures. We will take a look at the proposed solutions for automating these tasks in the next section.

2.1. Institute Website

Although the institute website in its current form serves many purposes, it also has many drawbacks. In most universities, including ours, the onus of updating the information reflected on the website is on the webmaster, or a team of select few with knowledge of web design and HTML. This scenario brings about a bottleneck in the flow of information.

In any university, the sources of information are numerous. It is unreasonable to assume that every source of information is capable of editing and updating the website. Even if they were capable, it would pose a security risk to allow all sources of information to edit and update the institute website.

Hence, all this information is sent to the team of webmasters, who are responsible for performing the requisite changes on the website. As mentioned before, this is a bottleneck and creates delays in delivering the information to those who require it. Any system that depends heavily on only a few technical people is bound to be flawed.

2.2. Academic Management

Assignment and project evaluation, timetable generation, compilation of mid term examination results and normalization of grades are some of the important tasks involved in academic administration.

Even today, most universities require students to use portable memory devices to carry their projects and reports. In many cases, even the computer based courses require hard copy to document the work done during the labs.

Currently, most faculty members manually compile each student's marks in electronic format and then tabulate them on paper. This tabulated result is then sent to a grade moderation committee, which oversees the grade allotment. After that, every department forwards the grade lists to the Department of Academic Affairs for generation of semester wise grade sheets. Ironically, these grade sheets are to be electronically printed so they are reentered into the computer.

We observe that the process is quite lengthy and multilevel with almost no automation at any stage despite the use of a computer at various stages. This manual handling may lead to several errors, in addition to huge delays. It would be desirable to make the entire process a little more transparent and automated in nature.

The problem with setting timetables is that the requirements of the faculty and some times the faculty itself changes for a particular course, and even if the department timetable is finally ironed out of clashes, it is difficult to synchronize the timetable for the whole institute. As a result there is huge shuffling of timeslots in the first month of every semester.

2.3. Library Management

Most libraries today already use management software to keep track of the books available and those issued. However, requesting for books is still a manual process. While the librarian always has the software for assistance, any faculty or student still depends on the librarian for simple activities like searching a book or checking the date when it would be returned. Further, the library is still a physically bound entity and no information of any kind can be obtained outside its walls. There is also no mode of instant communication between the librarian and the student. An electronic link will help the librarian inform the student when his requested book arrives, and also warn if the due date for return is surpassed.

In addition to book issue and renewal, book purchase requests and inventory management are also important tasks that, currently is the responsibility of the librarian. The problem most institute libraries face even today is identifying which books are in high demand and need. There is no formal way by which the faculty and students collectively request for purchase and provide the exact details like author and ISBN of the book required.

2.4. Student Affairs

The Department of Student Affairs performs several important and recurrent tasks. Registration of students at the beginning of every semester, collection of fees and issue of ID cards are just some examples. The current system in many universities is both cumbersome and time consuming, which proves to be a great inconvenience to both students and staff. Elimination of long queues is essential, which can be brought about only by implementing proper, secure and efficient automated system.

2.5. Hostel Allotment

Every year, several students admitted to university are allotted hostels. This process is also manually performed in most universities. It is always uncertain whether the student will get the

required room. It is also quite difficult to process preferences and choices of the students. Primary reason is that the wardens rarely have any idea of the design and structure of the hostel and the locations of the rooms in question.

Other maintenance tasks such as mess bill review and payment are also cumbersome processes. Mess bill payment may require high-end payment gateways and enhanced security. If the institute were to integrate the system with a bank where all students need to have compulsory accounts, even the payments could be automated.

2.6. Social Networking

Many universities provide email ids to their students but the state of social networking is primitive. In places where students communicate electronically, it is done using public services like Google Groups, Facebook, Writely, Bugzilla and Instant Messengers. This generates huge double sided traffic towards the institute's internet connection. Notices, even though computer generated, are still displayed on physical notice boards all across the institute, and in many cases, the documents displayed are outdated.

Matters of the institute are best kept within. Therefore, an in-house social network could be provided through forums, Newsletters, e-bulletin boards and Wiki pages where students could discuss both intellectual and community issues.

3. Proposed Solution

We now discuss the details of the various components of the proposed system. While the idea is universal, the following implementation is dependent on the environment and background that was available to us at the time of implementation and it could be possible to organize it better. We have therefore tried at best to give each component as much tolerance as possible for future upgrade.

3.1. The Content Management Framework

The content management system, as mentioned before, forms the core of the system. This is our primary weapon that helps provide dynamism to the entire system. Since all data is stored in the database, a user-friendly interface to that data is all that is necessary to change the corresponding information that might be shared between many modules.

Such an interface, popularly known as a WYSIWYG (What You See Is What You Get) interface would be easy enough for anyone with basic word processing skills to use. This forms the first and most important step in ensuring a smooth flow of information.

Every source of information can, irrespective of their knowledge of web or database design, update or create content for a specific module of the system, through the WYSIWYG editor. The editor is very easy to use, as might be evident from the illustration.

3.2. Your Online Desktop

Each user, based on their access levels will be able to use different modules in different ways. As an example, a faculty member will use the academic module to enter the marks for students, whereas a student would use it to view his/her academic record. Here we introduce the concept of your "Desktop".

The idea is to provide every user with the necessary links to allow him to perform his role on a single click. New email messages, warning messages, academic information, links to mess details, links to library, a page to edit personal information or manage a custom homepage and links to the forums are just one click away, once the user logs in from anywhere in the institute.

It would be appropriate to mention that dynamic appearance of links has been possible with the help of the access level and the roles defined in the LDAP Directory for every user. A PHP script runs through the directory to retrieve what links are to be provided to the user. Also, every requested page does a reverse check for the user's access level to confirm that he has access to the page in question.

3.3. Newsletters and e-bulletin

News and Notices are posted directly to the concerned group every day automatically. A digest can be created that would compile the summary and distribute the article. This system is already used in most communities and companies and has great application in a university too, especially in cases where the campuses are spread over multiple cities.

Permitted users can also run an editorial or a graffiti using this same system. This module is primarily of interest to students who can put their community building skills to test.

3.4. Academic Management

The various aspects to the academic system of our institute have already been discussed in section 2.2. Most of these processes have been automated in this module.

The primary milestone achieved by this system is the complete computer based compilation of test scores which are automatically converted to grades after the moderation function has been executed. The results show that the grades follow a perfect Bell Curve for the class and all without the involvement of any Grade Moderation Committee. The faculty concerned however has the super privileges for any last minute changes that might be required of him/her. All entries and changes are logged and hence we promise complete transparency of the system. The final grades are also delivered to the central database from which mark sheets can be generated easily. The grades are also automatically delivered to the students' inbox.

Next is timetable generation. This is a very tedious process which needs to take preferences from faculty (many of whom are often on guest duties) and also check the convenience of students. Hence an intelligent system has been devised using scripting, based on a dynamic algorithm that governs against any clashes. Time spaces have been implemented as building blocks and the annual calendar of the institute has been taken into consideration.

The annual report generation can also be included here. We can use other opensource tools here which already facilitate report generation. The profiles Database together with the Event Details of the year and other such relevant data is extracted from the MySQL central database and submitted to the report generation software using a simple script.

3.5. Library Management

As we have already mentioned, in addition to record of Issue and Renewal of books, we can also manage the procurement of books. A hierarchal system has been devised where the students and faculty submit their requests to the Head of the Departments via the CMS and then the Heads decide the final list and forward it to the library advisor. The library advisor receives these lists from each department and has many tools at his disposal to examine every request and also check his own stock of already ordered books for reference. Whoever is responsible for distribution of funds can now confer to the Library Advisor the maximum limit on every department and then the library advisor can finalize the lists of every department leaving the lesser priority books for next month. All of this is logically managed through the mySQL database.

The faculty and students can also use their respective interfaces to search for a book or request the librarian to reserve a particular book from sitting anywhere in the institute.

3.6. Student Registration

The registration process has been described under the section 2.4: Methodology while explaining the central and single user authentication system that governs the system. Once the registration is completed, the details are summarized for the ID cards and printed for distribution.

For hostel management, the Hostel manager has an interface where he can specify parameters such as which type of students need new rooms (all students, first years only, individuals and so on), change in number of hostels or in number of rooms in a given hostel and so on. The students have a separate interface where they can give their preferences such as roommates or preferred hostel and so on. This module also uses MySQL to mark each student with the preferences temporarily and allot rooms on the fly. Once the supervisor feels all entries have been made he can print out the final list which will be generated in a table format.

3.7. Institute Website

The institute website is generated entirely from a PHP scripts which extract relevant data from the database. New sections, pages and links are automatically added when the MySQL Database has a new entity. Upcoming events are automatically highlighted, and the database is archived periodically. All new data are entered from distributed sources using the CMS as discussed before. The website also provides an RSS feed for the notices and upcoming events which are of interest to the institute as a whole.

Apart from the main website, separate department websites are also generated and editorial content can be uploaded by any faculty. Job vacancy notices are also placed here. All research activities are retrieved from the database and summarized for external view. This same data is later forwarded to the Annual Report Generator.

A greater emphasis has been laid on speed and hence images are reduced to bare minimum. The style Sheet can be replaced within seconds because of the use of Smarty Style coding, which has simplified the HTML page, now devoid of data and containing only variables.

4. Implementation

Each of the areas in which automation would help a university has been outlined. The proposed solutions consider the following:

- Migration to the new system should be as smooth and effortless as possible
- The solution should be compatible with the systems currently in place
- Solution should be as user-friendly as possible with a gentle learning curve
- Solution should be cost-effective and easy to maintain

4.1 Technology

Each of the above aspects has been kept in mind during the preparation of the solution. The resulting solution is a tight integration of various LAMP (Linux-Apache-mysql-PHP) based web applications, mostly designed indigenously. The following are the tools that were deployed in addition to the custom scripted code base:

- Able managed network back bone
- OpenLDAP 2.3.3+: A directory service that stores data in a hierarchical structure. We use it to maintain all the user accounts along with other administrative details like department.

- PHP 5.1+: A popular scripting language, which forms the core of the dynamic pages on the system.
- MySQL 5.0+: A light Database Management System that was tested as most appropriate for the web scenario. It forms the core database backbone, and stores all information about any person, department or event related to the institute.
- Smarty 2.6+: A PHP based templating engine, used to completely separate content from design and introduce caching techniques to effectively scale the system
- Apache 2.0+: The robust and popular web server, used to host the entire set of web applications including the website.
- Dovecot 1.0+: A lightweight IMAP/POP3 server to handle the email facilities for the users.
- qMail 1.02+: A highly secure and configurable alternate to sendmail
- Squirrelmail: A webmail client implemented in PHP, that we appreciate for its simplicity and customizability.

The working of the final system as adopted is discussed later in the document.

4.2 Justification

The first and most significant reason that drove our choices of technology used is their being "OpenSource" and free.

4.2.1. Why OpenSource?

This greatly reduces the total cost of the proposed system by eliminating the software cost. In addition to the cost advantage, the resulting code would be easier to maintain and integrate with our own system, in comparison to other more complex technologies that could have been used.

By choosing these "open" tools, we can also avoid any issues with private licenses and Intellectual Property Rights while trying to modify the tools to suit our needs. What is even better is, in the event of us improving the software we can help the community by submitting the improvements to the moderators of the software in question.

4.2.2. Why Web based?

The solution is completely web based because of the ease at which it can be accessed from any kind of system and has least client side dependency. Technology wise, it is to be noted that most web applications today are as capable in features as a client application and it is today possible to give the power of an entire Operating System on the HTML Browser with the help of a powerful server and central accounts. Our system however attempts to achieve a distributed environment to reduce bandwidth wastage and promote delegation of control of flow of information.

4.2.3. Why not use existing CMS tools available?

The core of our system is a Content Management System (CMS). It is debatable if using existing CMS software that is widely available in the open source arena is easier. In fact, several IITs use such systems as PHPNuke or Drupal to manage their websites. We did look at this prospect at first but found that all the CMS available are rather generic in nature. We were interested in developing a CMS that was tailor-made to suit universities specifically. As a result, the system we built best suits a university environment. The role of the CMS in our system is also

greater than a simple website. It helps manage all types of information whether relevant to the website, the intranet, both or neither. The users get interactive interfaces for data collection of any type to be submitted to the central database, and retrieve on a need-basis, instead of just displaying everything as it is. The institute website interface as such has been kept light and speedy with only relevant data displayed.

4.2.4. Why openLDAP and mySQL together?

OpenLDAP is a Light weight directory that stores data in a hierarchical manor. As a result, searching for a data element is much faster than any DBMS. The users have been classified under their respective departments and batches and also bear the data that defines their access level. OpenLDAP hence is perfect for authentication.

OpenLDAP however, cannot be used in places where much change of data, insert, delete and update are expected to happen. Hence, for all other data storage purposes, mySQL has been used. mySQL has been designed specifically for deployment in the web environment and can handle one million users per second. Therefore, it was an obvious choice.

We maintain a central LDAP directory of all users, with multiple access levels. Upon receiving a request for accessing the intranet, the application looks up the user in the LDAP directory and allows him access to his data or the computer that is part of the college network, incase the institute needs to monitor every user's activity on college premises. The permissions available to the user to perform any task on the network depend on the access level defined for the user.

A general recommendation for deciding access criteria would be to create 3 classes each (User, Admin and SuperAdmin) at 2 levels (Global and Department). This leads to 6 different access control levels. The access levels are hierarchical, which means that a user with a higher access level, can perform all actions that all the levels below can. As an example, a Global Admin can perform all actions that a Department Admin, Department User and Global User can.

When the user logs into the web interface, cookies are set to define the user's access rights and to store the identity. This is done using session management and hence we handle the possibility of the user tampering with the cookies.

4.3. Methodology

The primary feature that led to the designing of the structure of our system is the central authentication system based on OpenLDAP and single user account for all the services we provide. This design led to the integration of all the applications of our system.

When a new person joins the institute, the online registration is undertaken where the user enters all the personal details that will eventually be stored in the MySQL's 'Profiles' database. A personal homepage for the user is automatically generated, to be visible externally. A new LDAP account is created and the user uses this account to login to any computer across the campus or carry out other functions and roles on the online system as assigned to him from time to time. The same userid and password is tied to the mail server also and the user's email account is activated. Thereafter the users will be notified of all new messages on the same interface.

The external website is supervised by a user who has one of his "Roles" as a Website administrator. His actions on the website superscribe all other users but he is not directly responsible for uploading and editing details about a course or an event for example. Each data item is uploaded by whoever concerned with the event or automatically from the database as configured by the Website administrator.

The faculty and staff can issue notices to the concern group of students through mailing lists and e-bulletin boards. They can also distribute assignments online through the submission system which will also be delivered via email to every student.

4.4. Scalability

The modular nature of this web application ensures that additional modules may be designed, or existing ones modified with ease and with no effect to the overall functioning of the system. We believe the system to be specific to managing a university, while still being generic enough to keep up with the variation between each university. We have tested the system with nearly 150 users undertaking the registration process simultaneously, but the technologies used promise efficiency and stability for a much higher load.

The scalability therefore depends not on the software but the hardware. The bottleneck here is network bandwidth. Hence the tools used are completely justified and the system is a recommend to any institute which still carries out its major functions manually.

5.Summary

We conclude that using a framework of loosely connected components as detailed above is the best method of managing information for an institute. It offers several advantages over adopting commercial or ready-made solutions, while still maintaining minimal required effort.