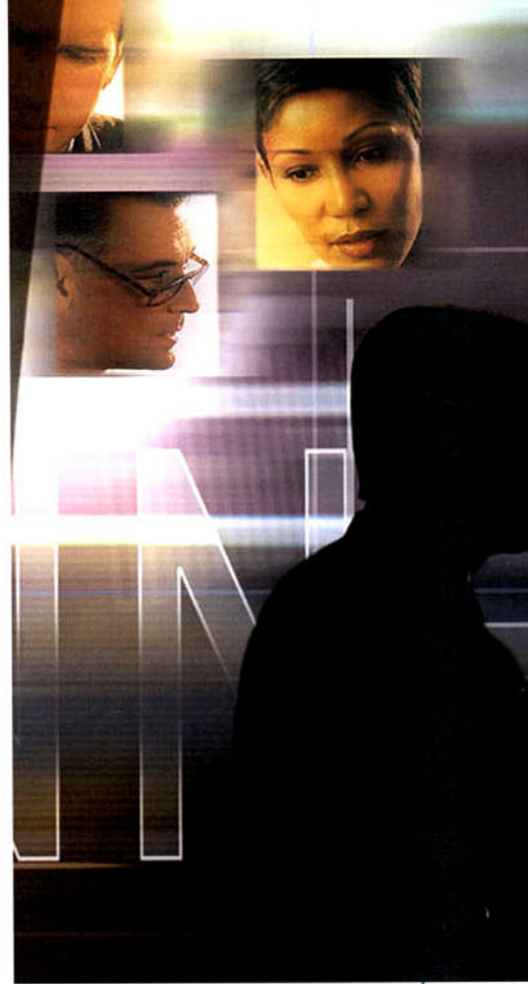


HOW time flies! July 2001 saw the birth of the Jenkins Law Library intranet. The staff was thrilled—an actual Web-based intranet. Prior to 2001, the “intranet” was a shared folder on the network. Now the staff could download important HR forms, search for policies and procedures, and submit requests for information via the Web. On top of all of that, it was organized by departments. What more could they ask for? As it turned out, a lot!

By 2005, our intranet was exploding with information and documents, so much so that people had stopped searching for files and just asked us where the files were or—worse—stopped using the intranet altogether. It was time for a change.



BY NICOLE C. ENGARD & RAYANA M. PARK

DESIGNING FOR A BRIGHTER OUTLOOK

There is no other way to explain version 1.0 of our intranet except to say it was “blah.” The colors were bland (gold and brown), and the design was flat. The great thing about designing an intranet

The great thing about designing an intranet is that it doesn't have to match the company's brand ...

is that it doesn't have to match the company's brand if you don't want it to. We decided to have a little fun with the design this time around, and what better way to encourage our staff to use the intranet than to give it a bright image?

The design was spruced up without making it too graphically intense. After all, the intranet was still an information resource, and the revamp in design was to aid, not distract, in finding information. Compared to our old site, the new intranet consisted of a wide range of warm colors (mostly blue, green, and mauve) and 3-D images for depth and an inviting look.

Task-related tabs were used to move away from organizing data based on departments and toward fostering collaboration among all areas of the library. These tabs are the main navigation system for the intranet. Buttons and icons were created to enable the

staff to easily identify their purposes. Symbols remained constant throughout the site, making the correlation between illustration and function obvious. The staff's feedback and suggestions were encouraged when we weren't sure what images would be easily understood. This further promoted the idea that the intranet was our own, that our goal for it to be easy-to-use and used by all was a common one.

The bright colors make it a pleasant place to work. I like having the tabs across the top listing the main library activities. The bread crumbs, under the tabs, and the quick links in the right column make it easy for me to get to the sites I use during the day.

—Kathy Coon, Deputy Director

As we sought to modularize the intranet, we decided to use cascading



Intranet 2.0: Fostering Collaboration

style sheets (CSS) so that we could control the look of the entire site from a singular location. Not only did this mean that the look of the intranet would be consistent and cohesive throughout, but it also meant that a design upgrade would be much easier to execute because the styles would be in one place.

Since our new intranet was less flat than its predecessor, we wanted to offer our staff a printable version of every page. For this we created another CSS file. Printable versions were easily implemented by putting `<div>` tags around the parts of the page that we did

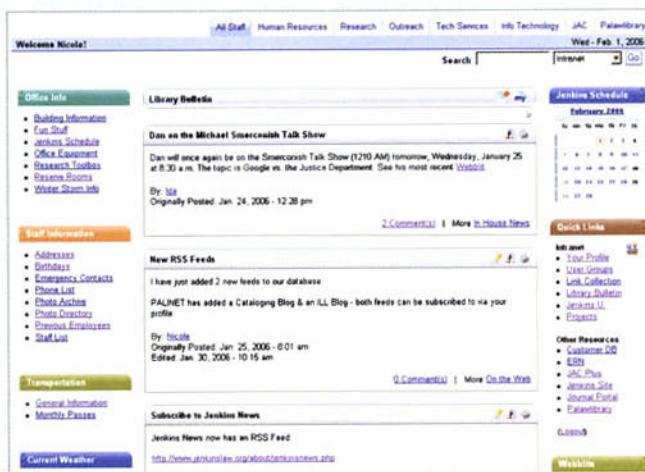
not want to print. We then added a style for each section of the page that we did not want to print, giving it a display value of none (`display: none;`). When the user clicked on the icon to view the printable version, the variable (`css type=printable`) was passed through the URL to define which CSS file to use. This is an efficient way to print a page without printing irrelevant information and without duplicating the code for every page. We then added a simple *if* statement to the header of each page to choose the correct CSS file.

DECIDING TO GROW AT HOME

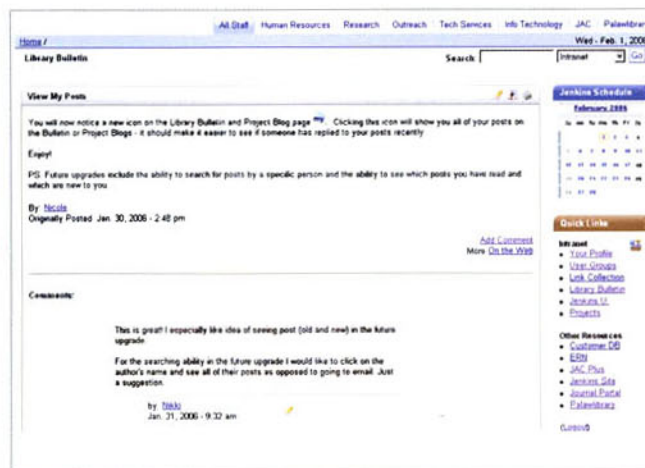
While the design was important, we saw an opportunity for a complete redevelopment. After researching what other libraries were doing with their intranets, we decided to use read/write Web or Web 2.0 technology. In May 2005, we offered an introduction to the read/write Web for our staff. We defined terms such as blog, wiki, and portal, then pointed staff members to Wikipedia [www.wikipedia.org] and encouraged them to edit articles that interested them so that they could get used to wiki technology and syntax.



The home page of our old intranet was a static HTML page that, while sufficient for its time, became an unwieldy source of information and outdated links. The colors were boring, and the site was too text-intensive.



Our newly designed intranet shows a refreshing new look with dynamic features. The home page contains the Library Bulletin blog, a mini-version of our shared calendar, RSS feeds according to the user's preference, and a section for current weather updates.



Here's an in-depth view of a library bulletin post, which is editable by either the Web team or its original author; comments to the post are listed below.

Once we had a direction, we needed to decide whether to use a prepackaged site or develop something in-house. We wanted more than just a wiki; we wanted blogs (one for news and interdepartment communication and several for ongoing projects), a Web-based help desk, and a shared calendar. Most importantly, we wanted to be able to easily link to our homegrown modules. At first we looked at free and low-cost portal/content management packages, but nothing lived up to our expectations. In the end we decided to build our own site using PHP and MySQL.

We didn't think this would be too big an undertaking considering that we already had written pieces of code for the library Web site [www.jenkinslaw.org]. The library's blog program and our Research Links module (Engard, Nicole, "Following the Yellow Brick Road to Simplified Link Management," *Computers in Libraries*, Vol. 25, No. 4, pp. 10–15) could both be used on the new intranet. That left four major components to work on: user authentication, the wiki, the shared calendar, and the Web-based help desk.

DEVELOPING FROM SCRATCH

Having never written a login script before, we decided to read through some online tutorials to see how it was done. We ended up using a tutorial we found on evolt.org [http://evolt.org/node/60265] that walked us through the process of creating a login script with a "remember me" feature. Since this script resided on our intranet, we thought the staff would appreciate not having to log in every time they opened their browsers. Using this tutorial we were able to develop a login script that stored the user's information both in a session and in a cookie (if they chose to do so).

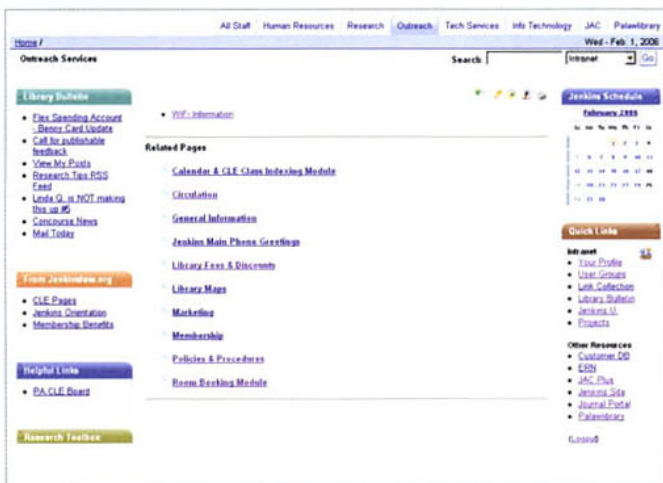
The biggest question with the wiki was how to track changes to the pages in order to be able to revert back to older versions. This was easier than we had originally thought. We actually needed only one table with two primary keys.

Thus, each time a page was edited, the pageID would remain the same but the version would be incremented by .01. Most of the popular wikis use the title and version as the primary keys. That model wouldn't work on our intranet because there were several pages with identical titles. "Policies," for example, can be found under almost every tab, so we needed to use a different identifier (the pageID).

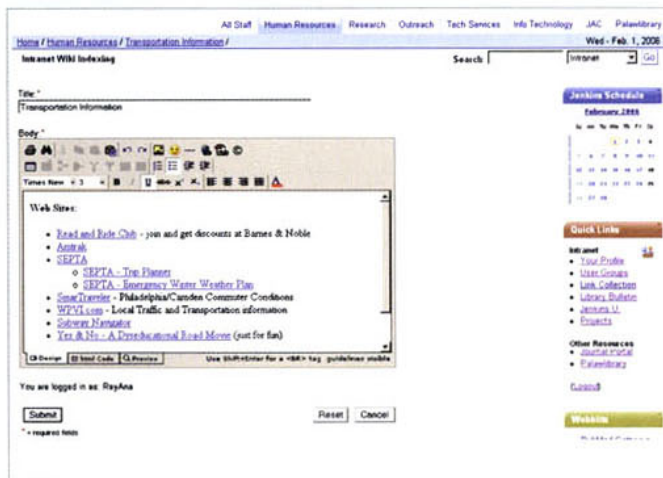
The wiki has helped me work much more efficiently. Before, when I wanted to make a change to one of the pages or reorganize something, I'd have to write a detailed explanation and submit it to the Web Team. This explanation took longer to write than for me to make the changes myself.

—Nancy Garner, Head of Information & Research Services

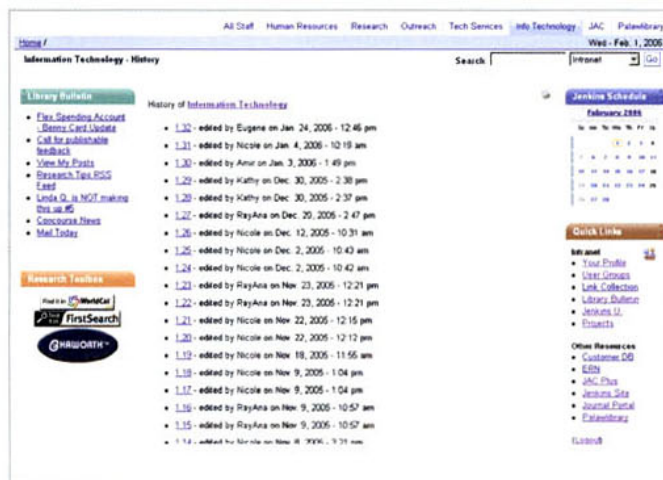
Next, we wanted to build in a logical navigation system so that pages were easy to find in version 2.0, something that was sorely missing in the prior version. Breadcrumbs (paths) had been successful on our library Web site, and we decided to stick with that method. However, this meant adding



Every wiki page can be edited or have related pages added under it. This is how hierarchical relationships between pages are formed.



WysiwygPro allows our entire staff (even those without HTML knowledge) to edit pages.



Every wiki page has a history so that we can revert back to older versions, if necessary.

structure to the wiki, which is somewhat unusual. Wikis usually depend on links between pages for navigation; we wanted to create a parent/child relationship between our pages to easily build paths. This way the page could be assigned more than one parent if needed.

PUTTING THE PIECES TOGETHER

With the calendar, we had to decide whether to use the script we had already written to keep track of events on the library Web site or to go with something a bit more robust. We chose ExtCalendar 2 [http://extcal.sourceforge.net], an open source calendar application. Although still in beta, this program was more powerful than anything we would have written on our own and gave us the freedom to edit the code and integrate it with our other modules.

Lastly, we had to find a Web-based help desk that fit our needs as programmers and our IT department's requirements. Most importantly, it had to be easy for the staff to use. We wanted something that was written in PHP so that we would be able to easily integrate it into the site. Our IT department wanted something that would track their answers and allow them to be searchable. We downloaded five demos, reviewed them, and purchased h2desk [www.heathco.com/h2desk]. It was written in PHP, included the ability to search tickets, and offered a knowledgebase for our users to look up frequently asked questions and tutorials.

I love the helpdesk feature of the new Intranet! Everyone in the reference department can check on the status of a particular problem, so our communication is much better.

—Jenny Hohenstein, Reference/Electronic Services Librarian

While it was easy to change the default CSS files for each of these applications so that the color scheme fit with ours, getting our framework around the applications became more of a problem. Since we were keeping track of login information in a session variable that was different from the calendar and help desk, we found that we couldn't put our template around either without making major alterations to our code. So, in order to seamlessly integrate the ExtCalendar 2 and h2desk, we decided to use iframes. Frames have become more versatile since the dawn of the HTML era, so don't be afraid to use them to incorporate an outside application into your own site. We used iframes primarily for aesthetics, in order to support continuity throughout our intranet.

CUSTOMIZING FOR THE PERFECT FIT

The blog we had written for the Jenkins Web site did not allow patrons to comment on posts, but we wanted to enable that for the intranet. In addition, the staff needed the ability to create new blogs for individual projects and to get permission to edit their own blog posts, if needed.



The Library Bulletin [blog] allows us to get the most recent library news and also to add our comments and questions. Instant feedback!

—Alice McCreary, Reference Librarian

We also wanted to upgrade the text area to a WYSIWYG editor so that all of our staff (even those without HTML knowledge) could contribute. This editor would need to work not only for our blogs but also for our wiki. After reviewing at least 10 different free and low-cost editors listed in the htmlArea directory [www.htmlarea.com], we chose to purchase WysiwygPro [www.wysiwygpro.com], a PHP class that allows for multiple editors on one page, easy button customization, and the ability to upload images and files.

I am glad we are a blogging community now. I feel it keeps us better informed about our library's happenings and we get a better perspective of our colleagues' jobs.

—Nikki Butler, Acquisition Coordinator

Once we had an editor chosen, we wanted to incorporate it into all of the other modules that had been written over the past 2 years. In addition to editing the text areas in these modules, we also wanted to address permission issues. In the past we used htaccess files to limit the number of users who were allowed to enter data. Now that we were customizing everything, we wanted to make it easier for the staff and eliminate the need to remember multiple passwords. We decided to create groups and then assign rights to each group.

We added a couple of lines of code at the top of each application to see which group the user belonged to and to see if they had rights to go any further. This user information was stored in the user's session. If they were members of the approved group, they entered the page without having to

enter a password. If they weren't, they were automatically redirected to a page informing them that they did not have the appropriate permissions:

```
if (!strstr($_SESSION['group'], " 1,") {
    header('Location: /norights.php');
}
```

INCREASING PRODUCTIVITY

In addition to addressing communication issues within the library we wanted to increase productivity wherever possible. The project blogs, which would hopefully make it unnecessary to hunt for old e-mails, was a step in the right direction, but there was more that we could do.

WIKI MySQL		
pgs_content		
FIELD	TYPE	LINKS TO
pageID	int(11)	
title	varchar(255)	
body	longtext	
user_id	int(11)	users -> user_id
created	datetime	
version	decimal(10,2)	
editable	int(2)	
pgs_subcategories		
FIELD	TYPE	LINKS TO
key	int(11)	
pageID	int(11)	pgs_content -> pageID
parentID	int(11)	pgs_content -> pageID

Table 1

BLOG MySQL		
blog_comments		
FIELD	TYPE	LINKS TO
postID	int(11)	blog_posts -> postID
commentID	int(11)	blog_posts -> postID
blog_posts		
FIELD	TYPE	LINKS TO
postID	int(11)	
typeID	int(11)	blog_types -> typeID
projID	int(11)	blog_projects -> projID
user_id	int(11)	users -> user_id
title	varchar(255)	
body	longtext	
date	datetime	
edited	datetime	
blog_projects		
FIELD	TYPE	
projID	int(11)	
title	varchar(255)	
description	text	
begin_date	datetime	
end_date	datetime	
archive	int(11)	
blog_types		
FIELD	TYPE	
typeID	int(11)	
type	varchar(255)	

Table 2

The project module is the coolest thing. In our busy library we can see what is going on, who is working on what, suggest ideas, and comment in one collaboration spot.

—Katrina Piechnik, Head of Technical Services

Our HR department kept several Microsoft Word documents with staff information: phone lists, birthday lists,

USER MySQL		
groups		
FIELD	TYPE	
groupID	int(11)	
name	varchar(100)	
groups_members		
FIELD	TYPE	LINKS TO
groupID	int(11)	groups -> groupID
user_id	int(11)	users -> user_id
users		
FIELD	TYPE	LINKS TO
user_id	int(11)	
username	varchar(25)	
password	varchar(32)	
firstname	varchar(50)	
lastname	varchar(50)	
email	varchar(100)	
birthday	date	
typeID	int(11)	users_types -> typeID
title	varchar(100)	
rss1	varchar(255)	
rss2	varchar(255)	
rss3	varchar(255)	
users_addresses		
FIELD	TYPE	LINKS TO
addressID	int(11)	
user_id	int(11)	users -> user_id
homephone	varchar(20)	
cellphone	varchar(20)	
address1	varchar(255)	
address2	varchar(255)	
city	varchar(100)	
state	varchar(35)	
zip	varchar(10)	

Table 3

emergency contacts, and home addresses. Whenever a member of the staff moved or got a new phone, they had to tell the HR department to update these documents. Now that we were storing user information in a database this was pointless. We created a profile page for each user that could be edited and kept up-to-date by the individual. Reports were then generated from this information.

Prior to this version of our intranet, we had a shared calendar that was accessible through our e-mail client. This meant that all events added to the schedule on the Jenkins Law Library Web site also needed to be added to the employee shared calendar. Once again, this was redundant, so we decided to edit our public events module and linked it right into the new Web-based calendar. This way events entered in one place appeared in both places.

I love the calendar because it is easy to locate and editing is a snap. It is much easier to use than the one that was part of our e-mail system.

—Regina Smith, Library Director

We also included a box with today's weather, a mini-calendar, and a search engine box. The weather box uses The Weather Channel's [www.weather.com] free weather update service, the mini-calendar links to the shared calendar page and the search engine form searches not only the entire intranet but also the library's Web site, our catalog, Google, Webster's dictionary, and more.

USER MySQL (continued)		
users_emergency		
FIELD	TYPE	LINKS TO
user_id	int(11)	users -> user_id
name1	varchar(200)	
home-	phone1 varchar(20)	
workphone1	varchar(20)	
cellphone1	varchar(20)	
relation1	varchar(50)	
name2	varchar(200)	
homephone2	varchar(20)	
workphone2	varchar(20)	
cellphone2	varchar(20)	
relation2	varchar(50)	
users_phones		
FIELD	TYPE	LINKS TO
user_id	int(11)	users -> user_id
extension	varchar(25)	

Table 3 continued



We wanted to make it so that our staff could use the intranet as their gateway to the World Wide Web, but, at the same time, we didn't want them to have to leave the intranet unless it was necessary. Having the weather easily accessible right from the home page was one step in this direction, but what about the people who like to read their favorite blogs or news sites in the morning? We found an RSS parser [http://lastrss.Webdot.cz] and incorporated it into the intranet. We added more than 50 RSS URLs to a MySQL database table (since most of our staff was too unfamiliar with RSS to find feeds without assistance) and let each user choose up to three feeds to display right on their home pages. We also created subject-specific news pages for computer virus information and library news and a page that lets users read the most recent "Unshelved" [http://www.overduemedia.com] comic strip.

The RSS feeds right on the homepage are great. I can check out headlines for certain sites without leaving the homepage.
—Joan Mazuchowski, Collection Development Librarian

Most obviously, the wiki made intranet changes much easier for everyone involved. The staff no longer had to e-mail the Web Team with their changes; they could make edits themselves with ease. We also made it so that if the staff had trouble formatting their pages with the WYSIWYG editor, they could easily report a problem with the pages to the Web Team. We could then go in and make any fixes that were necessary. This report icon could be found on every page on the intranet so that it was easy to contact us with the click of a button:

```
//Get the URL for the current page
function get_current_url() {
    $currentpage = "http://intranet" .
$_SERVER['PHP_SELF'];
    if((isset($_SERVER['QUERY_STRING'])) &&
(empty($_SERVER['QUERY_STRING']))) {
        $currentpage .= "?";
    }
    $_SERVER['QUERY_STRING'];
}
return $currentpage;
}
//print the report icon
print "<a href=\"/" . get_current_url() . "\">";
print "<img src=\"/" . images/icon-report.gif\" alt=
\"Report to Web Team\" border=\"0\">";
print "</a>";
```

INTRODUCING ... INTRANET 2.0

On Jan. 2, 2006, while the library was closed, we came in to roll out and test the new intranet. The next morning when staff returned to work, they had a brand new home page set in their browsers.

We have gotten amazing feedback from the staff; everyone seems to enjoy their new freedom and control. Now that everyone is used to the interface, we're seeing new pages added all of the time and project-related communication has never been easier to follow.

In the month since the intranet has gone live, we have spent most of our time tweaking the shared calendar, adding features to the blogs, helping with wiki page formatting, and, most importantly, providing training. We offered a month of sessions, each on a different area of the intranet. We wanted to make the staff as comfortable as possible with the change and that meant walking them through all of the new features.

Thanks for the workshops and listening to our feedback and suggestions.
—Kathy Coon, Deputy Director

We have gotten amazing feedback from the staff; everyone seems to enjoy their new freedom and control. Now that everyone is used to the interface, we're seeing new pages added all of the time and project-related communication has never been easier to follow.

Thanks to the project planning section our disaster planning doesn't look like a disaster.
—Ken Snyder, System Administrator

Deciding to develop everything ourselves was one of the best decisions we could have made. Since almost everything was written in-house, we know the code inside out and can fix bugs and make changes relatively quickly. Our new intranet makes the staff happy and has given us a great sense of accomplishment. What more could we ask for?

Nicole C. Engard [nengard@jenkinslaw.org] is Web manager, Jenkins Law Library, Philadelphia. *Rayana M. Park* [rpark@jenkinslaw.org] is Web assistant, Jenkins Law Library.
Comments? E-mail letters to the editor to marydee@xmission.com.