

UC Berkeley School of Information

Website Redesign & Development Request for Proposal

Overview

Founded in 1994, the School of Information (I School) is a graduate education and research community at the University of California, Berkeley. Using a hands-on, multidisciplinary approach, we bring together experts in the social sciences, computer science, law, business, design, linguistics, and philosophy to explore the organizational, social, and technical aspects of information in all of its many forms. Our two professional master's programs prepare students for leadership roles in data science and information management. Our PhD program is educating the next generation of top scholars focused on the study of information. I School faculty conduct leading-edge research at the intersection of technology and society in a wide variety of information-related fields.

The I School's current website (<http://www.ischool.berkeley.edu/>) was launched in 2009. Based on Drupal 6, it serves as both the public face of the school, informing external audiences about our academic programs and research, as well as an intranet, hosting internal content and automation supporting many of our business functions. We have three key motivations for redesigning the website at this time:

- We want a modern, elegant, responsive website design befitting an institution whose teaching and research areas include usability, user experience, and user interface design.
- We are currently re-aligning our brand to better integrate with the Berkeley brand (see <http://brand.berkeley.edu/>).
- We want to better serve our key constituencies, including faculty, staff, prospective students, current students, and alumni, as well as academic and industry partners.

The I School seeks a vendor to create a new visual design, implement this as a Drupal 7 site, and migrate the content from our existing Drupal 6 site into the new site.. We do not anticipate making large-scale changes to our existing information architecture and site content in this project. However, we would welcome suggestions regarding changes to the architecture and content that would result in a more appealing and effective visual design. I School personnel will be responsible for ongoing management of the website after the June 2015 launch.

Project Objectives

- Enhance and expand our brand through a modern, elegant, responsive, and accessible website that serves a wide range of constituencies
- Develop a website design that represents Berkeley's reputation as one of the top universities in the world while conveying the I School's unique identity as an innovative academic community in a rapidly growing field of study
- Increase student recruitment and engagement using dynamic and interactive website features

- Deliver relevant content and an optimal user experience across a broad range of devices
- Provide clear, intuitive navigation capabilities

Project Scope

The selected vendor will create a visual design and implement a Drupal 7 site with the following attributes:

1. Page design and site navigation that enhance the school's branding strategy, allow us to engage key audiences (see Appendix A), and serve the business requirements of the site's primary stakeholders (including the dean's office, external relations, student services, career services, and business services)
2. Custom themes (and subthemes) with page templates to accommodate a variety of content types (see Appendix B), based on Drupal coding standards
3. Pages based on mobile-friendly, responsive web design, delivering content optimized for a wide variety of devices
4. Rich multimedia offerings throughout the site, including photos, videos, audio clips, slideshows, graphics, etc.
5. Easy sharing of site content with social media services
6. Integration with our existing Google Analytics tracking program
7. Compliance with web (HTML5, CSS3 validation) and accessibility (WCAG 2.0, ADA Section 508) standards
8. Style sheet for print-optimized versions of all pages

In addition, the vendor will develop and implement a process for migrating the content from our existing Drupal 6 site to the new Drupal 7 site (see Appendix C for details on our current site). Finally, the vendor will also support detail-oriented, experienced project and stakeholder management that will help to keep the project running smoothly, on-time and in budget.

The scope of work does not include hosting, content development/editing, photography or video production, development of mobile applications, or ongoing maintenance of the website.

Appendix A: Hierarchy of Audiences

In cases where design decisions require us to optimize for particular audiences, we would like to prioritize the audiences as follows (from highest priority to lowest):

1. Prospective students
2. Prospective employers
3. Current students/faculty/staff
4. Alumni
5. Academic and industry partners
6. Other

Appendix B: Content Types

In addition to the standard Drupal content types, we have a number of custom content types. The following is a representative listing of these:

- Audio Recording

- Course
- Event
- Job Listing
- News Article
- Newsletter
- Person
- Publication
- Research Project
- Video Recording

Appendix C: Drupal 6 Site Details

Our existing Drupal 6 site includes an intranet that serves content to our internal audience of roughly 1,500 faculty, staff, students, alumni, and visitors. It also allows users to configure their I School computing accounts and automates some business functions. Users authenticate against our LDAP service via the ldap_integration module. There are roughly 12 LDAP groups that map to corresponding Drupal roles, allowing us to control which content and functions a given segment of our user population has access to. The site uses a custom theme as well as a large number of core, third-party, and custom modules as summarized below.

- 28,000+ nodes
- Primary theme:
 - 92 .tpl.php template files
 - 3,846 lines in main CSS file
- 87 active views
- 116 active blocks

Modules from Drupal core

- **Aggregator**
Aggregates syndicated content (RSS, RDF, and Atom feeds).
- **Block**
Controls the boxes that are displayed around the main content.
- **Comment**
Allows users to comment on and discuss published content.
- **Database logging**
Logs and records system events to the database.
- **Filter**
Handles the filtering of content in preparation for display.
- **Forum**
Enables threaded discussions about general topics.
- **Help**
Manages the display of online help.
- **Menu**
Allows administrators to customize the site navigation menu.
- **Node**
Allows content to be submitted to the site and displayed on pages.
- **Path**
Allows users to rename URLs.

- **PHP filter**
Allows embedded PHP code/snippets to be evaluated.
- **Search**
Enables site-wide keyword searching.
- **Statistics**
Logs access statistics for your site.
- **Syslog**
Logs and records system events to syslog.
- **System**
Handles general site configuration for administrators.
- **Taxonomy**
Enables the categorization of content.
- **Trigger**
Enables actions to be fired on certain system events, such as when new content is created.
- **Update status**
Checks the status of available updates for Drupal and your installed modules and themes.
- **Upload**
Allows users to upload and attach files to content.
- **User**
Manages the user registration and login system.

Contrib (third-party) modules

- **advanced_help**
 - **Advanced help** — *advanced_help*
Allow advanced help and documentation.
- **audio**
 - **Audio** — *audio*
Allows you to upload and playback audio files.
 - **Audio Attach** — *audio_attach*
Allows audio files to be attached to any node type.
 - **Audio Feeds** — *audio_feeds*
Provide XSPF, M3U, and PLS feeds for attached audio nodes.
 - **Audio getID3** — *audio_getid3*
Adds the ability to read artist info from and write to audio files. Requires that the getID3 library be installed.
- **autoload**
 - **Autoload** — *autoload*
Helper module to autoload classes under PHP 5.
- **backup_migrate**
 - **Backup and Migrate** — *backup_migrate*
Backup or migrate the Drupal Database quickly and without unnecessary data.
- **cacheexclude**
 - **Cache Exclude** — *cacheexclude*
Exclude certain pages from being cached.
- **captcha**
 - **CAPTCHA** — *captcha*
Base CAPTCHA module for adding challenges to arbitrary forms.

- cck
 - **Content** — *content*
Allows administrators to define new content types.
 - **Content Copy** — *content_copy*
Enables ability to import/export field definitions.
 - **Content Permissions** — *content_permissions*
Set field-level permissions for CCK fields.
 - **Fieldgroup** — *fieldgroup*
Create display groups for CCK fields.
 - **Node Reference** — *nodereference*
Defines a field type for referencing one node from another.
 - **Number** — *number*
Defines numeric field types.
 - **Option Widgets** — *optionwidgets*
Defines selection, check box and radio button widgets for text and numeric fields.
 - **Text** — *text*
Defines simple text field types.
 - **User Reference** — *userreference*
Defines a field type for referencing a user from a node.
- ckeditor
 - **CKEditor** — *ckeditor*
Enables CKEditor (WYSIWYG HTML editor) for use instead of plain text fields.
- cnr
 - **Corresponding Node References** — *corresponding_node_references*
Syncs the node reference between two node types which have a nodereference to each other.
- computed_field
 - **Computed Field** — *computed_field*
Allows the user to define computed values in custom content types.
- content_taxonomy
 - **Content Taxonomy** — *content_taxonomy*
Defines a field type for taxonomy terms
 - **Content Taxonomy Autocomplete** — *content_taxonomy_autocomplete*
Defines an autocomplete widget type for content_taxonomy
 - **Content Taxonomy Options** — *content_taxonomy_options*
Defines an option widget type for content_taxonomy for selects, radios/checkboxes
- ctools
 - **Chaos tools** — *ctools*
A library of helpful tools by Merlin of Chaos.
- date
 - **Date** — *date*
Defines CCK date/time fields and widgets.
 - **Date API** — *date_api*
A Date API that can be used by other modules.
 - **Date Popup** — *date_popup*
Enables jquery popup calendars and time entry widgets for selecting dates and times.

- **Date Timezone** — *date_timezone*
Needed when using Date API. Overrides site and user timezone handling to set timezone names instead of offsets.
- **emfield**
 - **Embedded Media Field** — *emfield*
Provides an engine for modules to integrate various 3rd party media content providers.
 - **Embedded Video Field** — *emvideo*
Defines a field type for displaying third party videos, such as YouTube and Google Video.
- **feeds**
 - **Feeds** — *feeds*
Aggregates RSS/Atom/RDF feeds, imports CSV files and more.
 - **Feeds Admin UI** — *feeds_ui*
Administrative UI for Feeds module.
 - **Feeds Import** — *feeds_import*
An example of a node importer and a user importer.
- **filefield**
 - **FileField** — *filefield*
Defines a file field type.
- **flickrapi**
 - **FlickrAPI** — *flickrapi*
Use Flickr APIs with Drupal using phpFlickr
- **getid3**
 - **getID3()** — *getid3*
getID3() extracts useful information from multimedia file formats.
- **google_analytics**
 - **Google Analytics** — *googleanalytics*
Adds Google Analytics javascript tracking code to all your site's pages.
- **imageapi**
 - **ImageAPI** — *imageapi*
ImageAPI supporting multiple toolkits.
 - **ImageAPI GD2** — *imageapi_gd*
Uses PHP's built-in GD2 image processing support.
 - **ImageAPI ImageMagick** — *imageapi_imagemagick*
Command Line ImageMagick support.
- **imagecache**
 - **ImageCache** — *imagecache*
Dynamic image manipulator and cache.
 - **ImageCache UI** — *imagecache_ui*
ImageCache User Interface.
- **imagefield**
 - **ImageField** — *imagefield*
Defines an image field type.
- **itunes**
 - **iTunes** — *itunes*
An iTunes specific RSS plugin for views feeds.
- **job_scheduler**

- **Job Scheduler** — *job_scheduler*
Scheduler API
- jquery_ui
 - **jQuery UI** — *jquery_ui*
Provides the jQuery UI plug-in to other Drupal modules.
- ldap_integration
 - **Authentication** — *ldapauth*
Implements LDAP authentication.
 - **Groups** — *ldapgroups*
Implements LDAP groups to Drupal roles mapping.
- lightbox2
 - **Lightbox2** — *lightbox2*
Enables Lightbox2 for Drupal
- listhandler
 - **Listhandler** — *listhandler*
Integrate forums with mailing lists.
- logintoboggan
 - **LoginToboggan** — *logintoboggan*
Improves Drupal's login system.
- mail_edit
 - **Mail Editor** — *mail_edit*
Allows editing of all mails.
- mailhandler
 - **Mailhandler** — *mailhandler*
Accepts submissions via e-mail.
- media_vimeo
 - **Media: Vimeo** — *media_vimeo*
Provides Vimeo support to the Embedded Media Field module.
- media_youtube
 - **Media: YouTube** — *media_youtube*
Provides YouTube support to the Embedded Media Field module.
- node_breadcrumb
 - **Node breadcrumb** — *node_breadcrumb*
Customize breadcrumb and menu location of nodes depending on their content type, associated terms and others conditions.
- oauth
 - **OAuth** — *oauth_common*
Provides OAuth functionality
- pathauto
 - **Pathauto** — *pathauto*
Provides a mechanism for modules to automatically generate aliases for the content they manage.
- recaptcha
 - **reCAPTCHA** — *recaptcha*
Uses the reCAPTCHA web service to improve the CAPTCHA system.
- securelogin
 - **Secure Login** — *securelogin*
Enables secure login
- securepages

- **Secure Pages** — *securepages*
Set which pages are always going to be used in secure mode (SSL) Warning: Do not enable this module without configuring your web server to handle SSL with this installation of Drupal
- stringoverrides
 - **String Overrides** — *stringoverrides*
Provides a quick and easy way of replacing text.
- subscriptions
 - **CCK Fields** — *subscriptions_cck*
Provides CCK field data as mail variables.
 - **Content Subscriptions** — *subscriptions_content*
Allows users to subscribe to nodes.
 - **Subscriptions** — *subscriptions*
The basic Subscriptions framework.
 - **Subscriptions Mail** — *subscriptions_mail*
Sends out mails with drupal_mail, customizable with the separate mail_edit module.
 - **Taxonomy Subscriptions** — *subscriptions_taxonomy*
Allows users to subscribe to taxonomy terms.
- taxonomy_access
 - **Taxonomy Access Control** — *taxonomy_access*
Access control for user roles based on taxonomy categories.
- themekey
 - **ThemeKey** — *themekey*
Map themes to Drupal paths or object properties.
- token
 - **Token** — *token*
Provides a shared API for replacement of textual placeholders with actual data.
- transliteration
 - **Transliteration** — *transliteration*
Converts non-latin text to US-ASCII and sanitizes file names.
- twitter
 - **Twitter** — *twitter*
Adds integration with the Twitter microblogging service.
 - **Twitter Post** — *twitter_post*
Enables posting to Twitter
- twitter_pull
 - **Twitter Pull** — *twitter_pull*
Pull Info from Twitter and Create Contextual Insets
- variablecheck
 - **Variable Check** — *variablecheck*
This module allows site administrators to check whether settings stored in the variable table have been saved in the correct format.
- views
 - **Views** — *views*
Create customized lists and queries from your database.
 - **Views exporter** — *views_export*
Allows exporting multiple views at once.

- **Views UI** — *views_ui*
Administrative interface to views. Without this module, you cannot create or edit your views.
- *views_bonus*
 - **Bonus: Views Export** — *views_bonus_export*
Plugin to export views a couple of formats including Comma Separated Values(CSV), Doc or XML
- *webform*
 - **Webform** — *webform*
Enables the creation of forms and questionnaires.
- *webform_validation*
 - **Webform Validation** — *webform_validation*
Add validation rules to webforms

Custom modules (developed in-house)

- **Account Request Form** — *arf* — (128 kB)
Account invitation and request workflow
Creates three new DB tables; provides info by XML remote procedure call
- **CalMail Integration** — *calmail_integration* — (37 kB)
Manage CalMail email accounts and redirects
Uses function library to interact with UC Berkeley email system. Creates email & updates accounts and returns information about those accounts.
- **CalNet Directory** — *calnet_directory* — (9 kB)
Display CalNet Directory Info
Pulls info from UC Berkeley's authentication & directory system. Also authenticates a user in the campus CAS and records their ID#.
- **I School Class Registration** — *ischool_reg* — (88 kB)
Workflow for student course registration
Uses our CCK data. Six new DB tables. Admin & end-user workflows.
- **I School Email Template** — *ischool_email_template* — (16 kB)
Alternate version of a page, with an email-friendly version
Creates alternate displays of a node, designed to be input into email program.
- **I School Miscellaneous** — *ischoolmisc* — (36 kB)
Miscellaneous tools for the School of Information
Pulls data from local disk & print quota records for display. A few small cron tasks. Custom Flickr image blocks, using Flickr library. A few custom Twitter feeds using twitter_pull module.
- **I School Newsletter** — *ischool_newsletter* — (16 kB)
Alternate versions of a page, with email-friendly HTML and text-only versions
Creates alternate displays of a node, designed to be input into email program.
- **I School Password Change** — *ischool_password_change* — (7 kB)
New page for resetting your password
Connects to our LDAP and Active Directory servers to make password updates.
- **I School Password Reset** — *ischool_pwd* — (11 kB)
Updates to the built-in password reset functionality
Alters existing password-reset form & emails. Adds 1 DB table.
- **I School Subscriptions UI** — *ischool_subscr_ui* — (9 kB)
Provides a new, customized UI to Subscriptions module.
- **I School Taxonomy Cleanup** — *ischool_taxonomy_cleanup* — (1 kB)
Fix some little stuff about the taxonomy module

- **I School Timezone Tools** — *ischool_timezone* — (22 kB)
Tools to help users set their timezones — an improved UI, and new forms and display tools.
- **I School Twitter** — *ischool_twitter* — (8 kB)
Form to post to a shared Twitter account; specific blocks using “Twitter Pull” module; specific blocks using “Twitter Photos” module.
- **I School Twitter Cards** — *ischool_twitter_cards* — (3 kB)
Display a page thumbnail when someone Tweets a link to the page. Adds info to the header of every page providing a thumbnail image & page summary for display in Twitter, Facebook, and other sharing sites. Uses imagecache & specific CCK fields.
- **I School Webform Email** — *ischool_webform_email* — (3 kB)
Email a regular digest of webform submissions
Adds 1 DB table
- **Insert Imagefield** — *insert_imagefield* — (7 kB)
An input filter to include images from a CCK Imagefield inline in a filtered text field. Also includes its own theme template file.
- **Mailman Integration** — *mailman_int* — (33 kB)
Integrate with Mailman email list software
Uses code library to interact with UC Berkeley email list system. Subscribe & unsubscribe users from a variety of lists, and return information about those lists. Info cached in Drupal DB. Includes 2 new DB tables
- **Obfuscate Email** — *obfuscate_email* — (9 kB)
Provides a function and an input filter to transform email addresses into a format difficult for spam-bots to harvest
- **Remote Management** — *remote_mgmt* — (14 kB)
Manage Drupal remotely via XML-RPC
Creates & updates nodes & users, based on external commands from our account-management system.
- **Twitter Photos** — *twitter_photos* — (19 kB)
Pull Photos from Twitter
A clone of the “Twitter Pull” module that returns photos posted to Twitter from specific hashtags, queries, or handles.