

php | architect™

The Magazine For PHP Professionals

Drupal

a modular
framework
for content
management

php | tek 
\$build->deploy->scale
Orlando, April 26-28, 2006

INTRODUCTION TO SOAP

LOGGING WITH log4php

ADVANCED SQLITE DEVELOPMENT

ZEND STUDIO 5.0 REVIEWED

SPECIAL REPORT:
WHERE IN THE WORLD IS PHP?
INSIGHT ON THE USE OF PHP AND ITS ADOPTION IN THE WORLD



Drupal

an introduction

Organizations these days are demanding content management applications, from company homepages to large, community websites. Let Drupal help you build these sites quickly and efficiently by building on a common, modular framework.

by **TITUS BARIK**

In their every day development, most web consulting companies encounter many of the same components when designing pages for non-profit organizations and activist groups. These requirements include forums, a donation system, membership management, newsletters, and articles, just to name a few. Many of these organizations also have small to mid-size e-commerce needs, to effectively sell and distribute promotional materials, or to handle signups for paid events.

Developers often tackle this task by rolling their own systems in an attempt to make their sites as customizable as possible, while still providing core components to build on for each individual client. In theory, it sounds like a good idea. In practice, the limitations of their underlying frameworks ultimately result in nothing more than a mash of hacks that only loosely integrate the unique elements required for their clients.

In the worst case, web development shops end up coding the entire thing from scratch, each and every

PHP: 4.4

O/S: *Debian*

OTHER SOFTWARE: *Drupal 4.6.3*

LINKS:

Drupal Handbook:

<http://drupal.org/handbook>

Drupal Modules:

<http://drupal.org/project/Modules>

PHP Template:

<http://drupal.org/project/phptemplate>

Democratika Theme, by Chris Messina, part of the CivicSpace distribution of Drupal:

<http://drupal.org/handbook>

TO DISCUSS THIS ARTICLE VISIT:

<http://forum.phparch.com/272>

time. Somehow, rolling your own system never quite works out the way it needs to. Even if you have two well-written components, it is often the case that it's not the components themselves that are the problem, but the integration points that add burdensome complexity. Integrating disjoint components, say Forum A with Shopping Cart B, has until now, been a difficult problem to solve.

This article introduces Drupal, a modular, open source content management platform that attempts to address these difficult challenges. Drupal provides a flexible, extensible framework—an alternative to roll-your-own content management solutions. It can be refit for a variety of different content systems, including community portal sites, personal weblogs, and resource directories, simply by adding and removing sophisticated modules.

Drupal offers you a working, tested framework for building components, and handles the insignificant but tedious details of module management, user management, and integration, so you can focus on developing the core of your project. More importantly, Drupal has a very large user base, and modules to handle most common functionality are already available. When these modules don't exist, the hooks system allows the developer to create custom modules to interact with the Drupal core, while leveraging the existing Drupal building blocks.

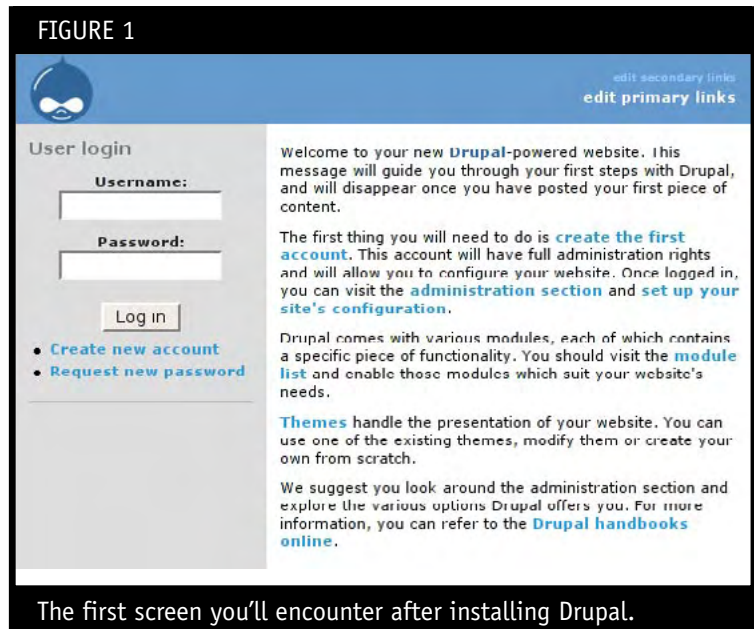
It's not at all surprising, then, that the focus of this article is not on complex development or PHP code, at least not directly. In fact, it's quite the opposite. In this article, we'll develop a mock political organization's web site, Democratica, without writing a single line of code. Along the way, I'll demonstrate the various

built-in and contributed modules that allow you to get a site up running quickly and effortlessly.

Installation

This section discusses the installation and configuration details of Drupal, including the pre-requisite software requirements. Our test installation consists of Drupal 4.6.3, PHP 4, and Apache 2, running on Linux with a MySQL 4.1 database server.

Though Drupal supports both MySQL and PostgreSQL, be advised that most available third-party modules are coded specifically for MySQL. This effectively forces you to use the former, regardless of



what the Drupal core modules support.

Drupal allows you to host multiple sites on a single Drupal instance. These sites can be found under the sites folder. For simplicity, I'll only utilize a single site setup in this article.

After creating and assigning the appropriate permissions to your database, load the tables:

```
mysql -u nobody -p drupal < database/database.mysql
```

Next, enter the sites/default folder and modify `settings.php`; namely, modify the `db_url`, and `base_url`.

Drupal works best on dedicated hosting systems. Since Drupal

memory requirements increase proportionally with the number of active modules, the default PHP memory limit of 8 MB provided by most shared hosting providers is typically not enough. To be safe, set this value to 32M or 64M in `php.ini`. If the memory requirements are not successfully met, Drupal will behave strangely, displaying broken administrative menus and erratic page rendering.

Next, add the following to your `crontab`:

```
0 * * * * wget -O - -q http://democratica/cron.php
```

This allows for scheduled events to trigger properly within Drupal. Such events include search indexing, mass mailing, and scheduled publishing.

That's all there is to the installation. Navigate to your Drupal site using your web browser, and you'll be presented with a base page, as shown in Figure 1.

Finding Your Way Around

It's time to create your first account. User #1 is provided with full administrative access. All subsequent users are given authorized user access. Additional groups can be generated through the Drupal system as well, providing fine granularity amongst different group types. In Democratica, for example, we may want special access for employees, and different access privileges for members and guests,

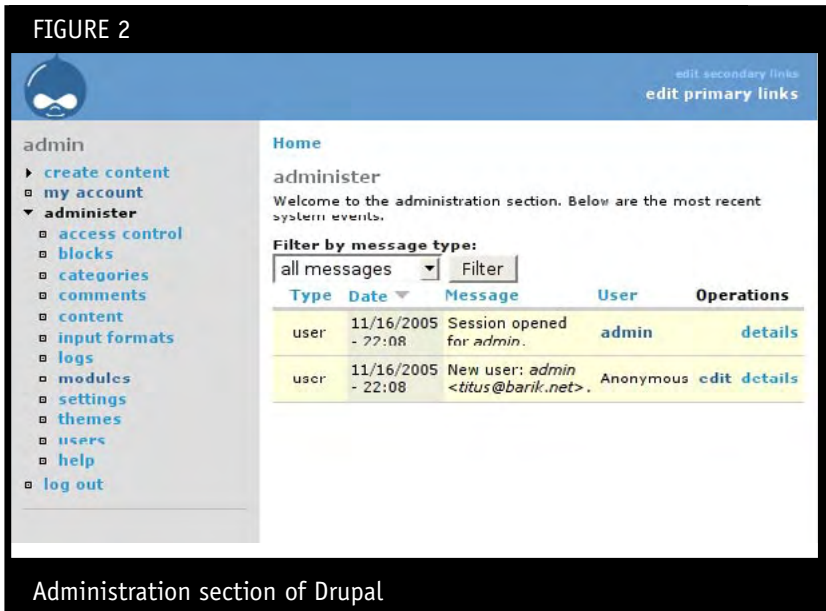
After logging in, select the administer menu. You'll be presented with Drupal's recent system events, a web-based logging system for monitoring the activity on your web site. A typical administration page is shown in Figure 2. For now, enter the settings menu, and change the

default name of the site to Democratica. You should also set your time zone. Feel free to modify any other settings, as well.

Perhaps the two most useful menus under *administer* are *blocks* and *modules*. The modules menu allows you to extend the core functionality of Drupal. Within this menu, you'll activate and install module components to tailor the functionality that your site requires. For example, you could enable search, forums, events, and so forth. The blocks menu enables and disables boxes that can be positioned on the left and right side bars of your web site. The available blocks vary, depending on which modules are currently active. Under modules, notice that the page and story modules are checked by default.

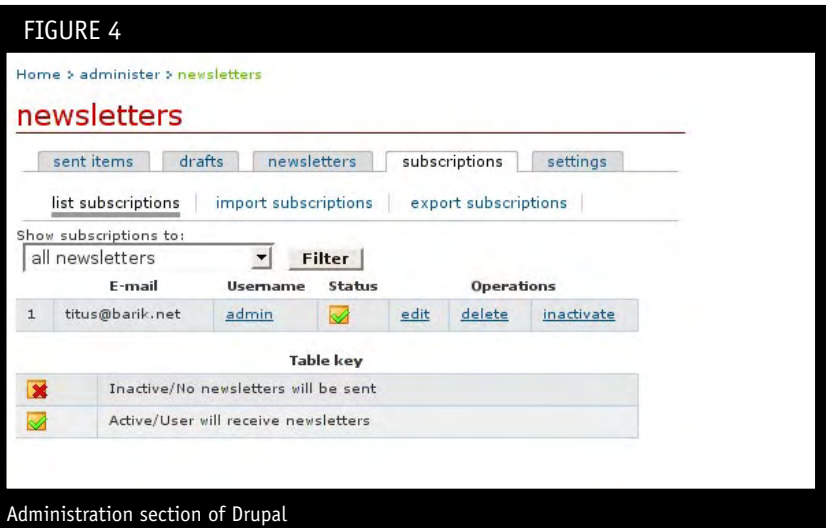
Now, jump up to the *create content* menu. You'll notice that there are two available content types: page and

FIGURE 2



Administration section of Drupal

FIGURE 4



Administration section of Drupal

FIGURE 3



The Drupal donation block.

story, the same as the modules we previously witnessed. Enabling content modules will add additional content types to this menu. After initial setup, most interaction with the Drupal site will occur through this menu.

Finally, let's examine the content menu, again under *administer*. This menu allows us to configure the properties of content nodes, which at first, can be a bit perplexing. Unlike a dedicated content system such as WordPress for blogging, or osCommerce for e-commerce, Drupal aims to be flexible, to suit any type of web platform. In order to provide such flexibility, the fundamental type of almost all content is that of a simple node. Custom behavior is then implemented by simulating the base node type, and subsequently adding or removing properties to that node type. The *content* menu allows us to set additional properties, modify nodes, and add custom nodes to facilitate our development process.

designers, for example, can independently develop multiple, pluggable layouts, while your logic developers can write custom modules and other core functionality.

Modules

The remainder of this article will focus on modules. The process of installing some modules is more involved than others, though none of the modules are exceptionally difficult to integrate. The typical installation procedure for a module is as follows:

First, load the database schema. This is usually one or more iterations of the following command:

```
mysql -u user -p database < schema.sql
```

Next, copy the module to the modules directory. Then activate the module in the *administer modules* menu.

Unlike a dedicated content system such as WordPress for blogging, or osCommerce for e-commerce, Drupal aims to be flexible, to suit any type of web platform.

Democratica Theme

In Drupal, themes are rendered through the use of template engines. Unfortunately, Drupal by default only provides `xtemplate`, and the Democratica theme that we're interested in requires `PHPTemplate`. Luckily, this takes very little effort. Simply download the `PHPTemplate` engine from the Drupal web site, and extract it to the `themes/engines` folder. Next, download and extract the Democratica theme, and place it in your themes folder. Login to your web site as `administrator` and access the themes menu. The Democratica theme is now automatically available. Set it to `default`, and save your configuration. Your new theme is now applied.

Theming is a powerful component of the Drupal framework that will greatly streamline development. Namely, it enables you to parallelize your backend business logic and your presentation code. Your graphic

Then, set configuration parameters for the module under the *administer settings* menu, and configure access control permissions on the module through the **access** control menu.

Finally, if the module has a presentation layer, enable the block in the *block* menu, specifying its location and ordering.

Some modules have dependencies on other modules. Many modules, for example, depend on *Flexinode*, a module that allows non-programmers to create new node types from within the content types menu, without having to custom load database node schemas for each new node type.

Before continuing, install and activate the *Flexinode* module. A *content type* menu will appear under *content*, and a *flexinode* menu item will appear under *settings*.

Donations

Our first module will be a simple one. Most political organizations have a need for donations, and our site is no different. The Donations module allows us to

through the simplenews module. Install it now. Like the donation module, it will appear as though nothing has changed. Click the new newsletters menu item under *administration*; this will create a default *Drupal*

Drupal provides an excellent, out of the box, event handling system.

accomplish just this goal, with seamless integration to the Donorge.org donation service. Under settings, modify the text as desired. Since this isn't a live site, the donation ID doesn't actually matter.

Nothing appears to have actually happened, but that's because we haven't told Drupal to place the content block on the web site. Let's do that now. Click on the blocks menu, enable *Donations*, and place it on the right sidebar. A third column on the Democratica theme will now be enabled, with your new donation module, as shown in Figure 3.

Without having to modify a single line of code, we've managed to add donation functionality to our Democratica site. Reusable components such as the *donations* module are what make developing under the Drupal framework so efficient.

Newsletter

After interviewing several non-profit organizations, the most demanded feature next to donations is that of newsletters. Newsletters, when used effectively, are a vital method of directly addressing your members, notifying them of important news and issues. Our political organization, in particular, wants to let our subscribers know about the latest campaign news.

Drupal provides such functionality

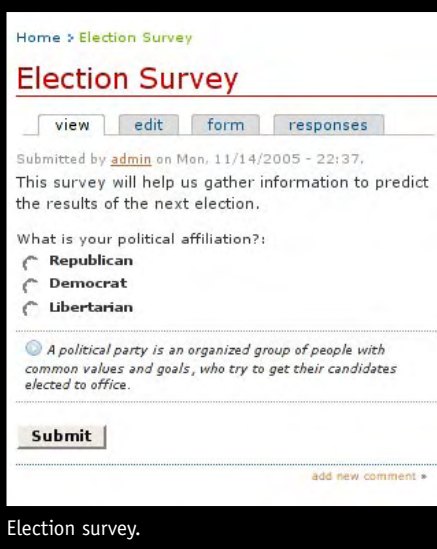
newsletters. Under the *newsletters* tab, click edit type for Drupal, and change the name to "Democratica Newsletter." The Newsletters interface allows you to create multiple newsletters, track the progress of newsletters, and track subscribers, as shown in Figure 4. Under blocks, activate the newly created newsletter, placing it on the right sidebar.

Next, create a newsletter from the *create content* menu. Creating a newsletter item is simple enough that it doesn't require any further explanation here. Assuming your PHP configuration and mail server are setup correctly, all subscribers will receive an e-mailed copy of the newsletters you've just posted.

Survey

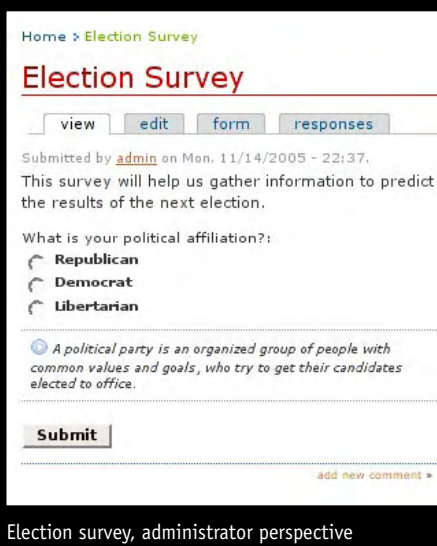
Our political organization wants to collect data that may help in predicting the outcome of the next election. To do this, they would like to create a survey, asking various questions on the web site. At first glance, such a task seems like a great deal of work. We'd have to design a backend database to store a variety of survey questions,

FIGURE 5



Election survey.

FIGURE 6



Election survey, administrator perspective

implement forms on the front-end, and provide facilities to allow the organization to collect and download the data in a user-friendly format. In addition, we would have to provide some sort of dynamic form creation system to allow users to create their own surveys.

Unsurprisingly, Drupal makes this task simple and reusable through the survey module. The *survey* module depends on the forms module, which provides an API for adding user-customizable form elements within modules. First, install the *form* module and activate it. Then, install and activate the *survey* module.

Under the *create content* menu, the *survey* item will appear, in addition to the standard story and page items. Let's now create a survey called "Election Survey." After entering the general information, you'll find yourself in a *survey* menu, from where you'll go to the form tab.

Add a radio field. We'd like to know which political party you belong to. Under selection options, enter "Republican; Democrat; Libertarian". The semi-colon is used to delimit the multiple options for the given field type. Add any additional desired fields. A non-administrator user will be presented with the survey, as shown in Figure 5. As an administrator, your window will appear more like that in Figure 6. When a user completes a survey, his or her results can be found under the *responses* tab.

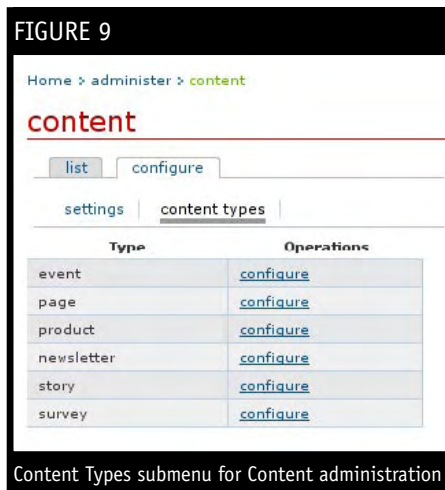
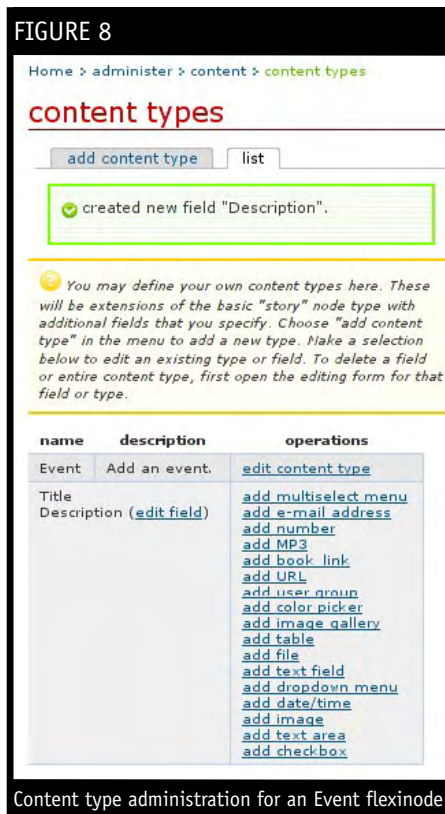
E-Commerce

Throughout the political campaign, our organization wants to sell tangible items through its fulfillment house. These items include bumper stickers, signs, buttons, and so on.

The e-commerce package includes several modules. Some of the modules are located in Figure 7.

E-commerce preferences can set under the *settings* menu. Among other options, items include things like whether customers must create an account to order, shipping methods, and payment methods.

For our purposes, activate all of the required



modules. In addition, activate the *cod* module and the *tangible* module, for tangible products. Under *create content*, a product item will now appear. Here, you can set the product title, description, price, and the type of product. For Democratica, the fulfillment house handles all inventory, and we thus disable inventory management.

Under the *block* menu, enable the *Shopping Cart* module.

Under the *create content* menu, create your product. Once completed, it will appear in the top-level products menu. From here, you can add it to your cart and checkout. The e-commerce module can and probably will need to be customized through PHP to suit your particular needs. However, the provided shopping cart already provides much of the required base functionality.

Events

Democratica can't be successful in rallying support if it can't successfully organize events. Drupal provides an excellent, out of the box, event handling system.

The *event* module relies on the *Flexinode* module, which was installed earlier in this article. A close cousin of the *event* module is the *location* module, which allows for additional *location* information.

The *location* module will add the final touch to our event system. Currently, the location module fully supports United States zip codes. The module provides routines for conducting postal code proximity searches, linking to Google maps, and other functions for collecting locations. Follow the directions for installing this module, carefully. This module is unique in that it requires you to load three mysql data files: a location schema, a zipcode schema, and US zip code data. Then, just as we've done for all other modules, activate it through the *module* menu.

After enabling both modules, access the *block* menu and enable *List of upcoming events*. An empty *upcoming events* listing will appear, depending on your sidebar selection.

Unlike most of our previous modules, an event is not a node type in itself. Specifically, event properties can be added to any node type. So, for example, you can create your own *Generic Event* node type using *Flexinodes*, or you can create a *Birthday* node type for birthdays, and so on, to describe different categories of events.

Let's create a *Generic Event* node now. Click on *content type* in the *content* menu under administration, and click the *add content type* tab. Call our content type name "event," and give it a short blurb in the description. You'll now be placed in the *list* tab, where you can add additional fields to your custom node. Title is already included, so let's add a text area for the event description. After following these steps, your screen should resemble Figure 8.

Though we've created a generic event node, we have yet to bind the event module properties to it. To do so, click on the content menu, then the *configure* tab. From here, click the content type subtype, where you'll find the configure option for the event. Yes, it's deeply buried under a lot of menus, so look at Figure 9 for assistance.

Under *configure*, scroll down to "show in event calendar." Set this to all views to enable *event times* for the generic event node that you've just created. Under the *blocks* menu, enable "List of upcoming events." In addition, select "Enable for location," a feature contributed by the *location* module. Set "Allow" for street, city, state, postal, and country.

You're on the final stretch. It's time to actually add an event. Under *create content*, you'll now find an *event* item. Fill in the details for your event. In our example, we're celebrating the *Democratca Launch Party*, shown in Figure 10. Notice also, that the event has been added to the upcoming events list in the sidebar.

RSVP

The Democratca organization is pleased with the events system. However, only special guests are invited to the Democratca Launch Party, based on, say, the number of charitable contributions, and our currents events system has no way to track this.

This is where the RSVP module comes in. RSVP lets users invite people to attend an event. It sends an invitation email to a list of people, and can the track which users have looked at the invitation and their responses. Invitees can view and reply without having user accounts.

After enabling the RSVP option, a *Create RSVP* link will become active on all events. From here, you can invite selected guests to your event. An invite preview is shown in Figure 11.

Other Modules

Our last module is as big one. *CiviCRM* is a web-based, open source, internationalized, constituent relationship management (CRM) application, designed specifically to meet the needs of advocacy, non-profit and non-governmental organizations. In short, it keeps track of people.

CiviCRM is large enough to be a product on its own. It is also the most complicated of modules to install, and its installation instructions would comprise an entire article by itself. The *CiviCRM* module is also a dependency for the other modules which require contact management, and it is also an optional requirement for the event finder module, which allows the user to search for events based on event type, geographic location, and proximity to major metropolitan area.

Thankfully, the *OpenNGO* web site provides excellent installation instructions for the *CiviCRM* module. Here, I'll omit the installation steps and show you the brief demo of the *CiviCRM* and *Event Finder* modules.

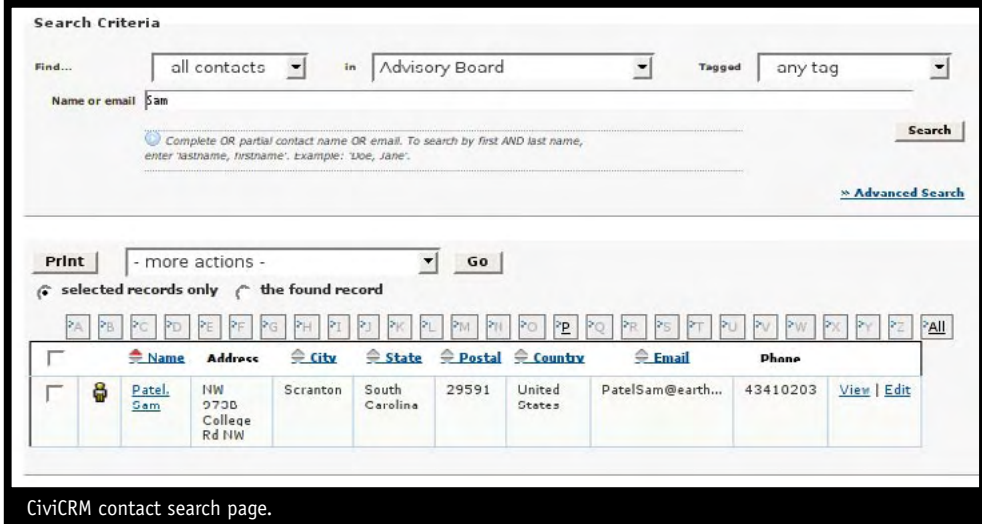
CiviCRM primarily enables the *Find Contacts* feature, shown in Figure 12. It also allows you to manage groups, relationship types between contacts, and



location information. For those who don't have in-house contact management tools, *CiviCRM* may be a good option.

Event Finder takes the functionality of the existing events module and adds the ability to actively search for local events by category, event type, and zip code. It also enables online registration for events, where one can specify the maximum number of registrants.

FIGURE 12



CiviCRM contact search page.

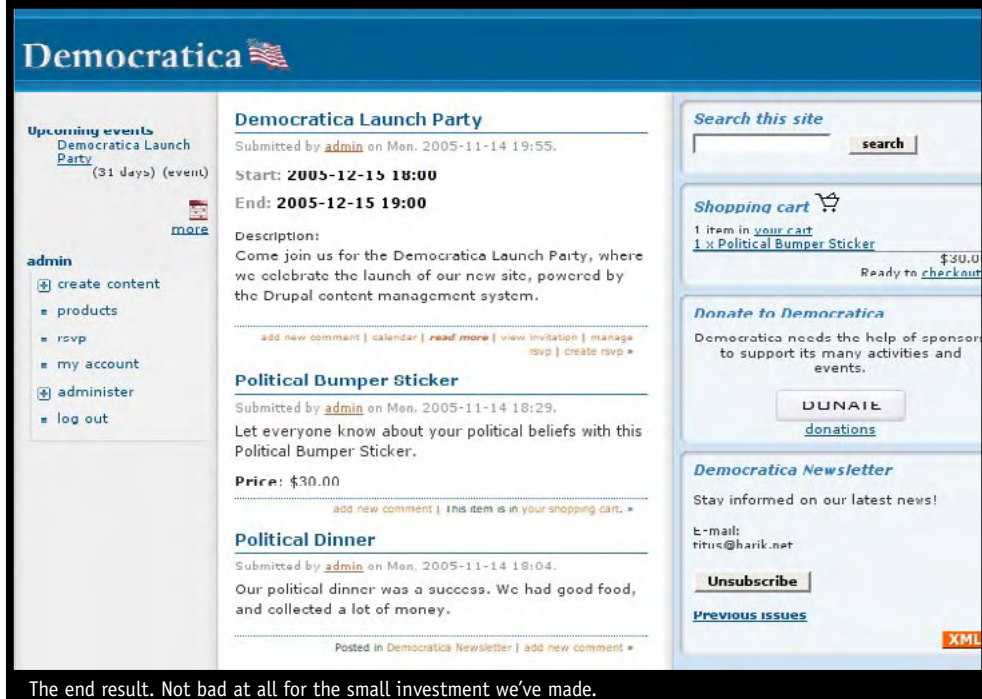
Conclusion

The final result is shown in Figure 13. In the course of about an hour or two, you've created a dynamic site that provides a shopping cart, donation system, newsletters, event management, and surveys. You've also taken advantage of the underlying Drupal framework, including its user management and permissions systems. Even if you're required to customize each and every one of these components for your specific project, the time saved in comparison to writing a custom solution from scratch is simply tremendous.

Drupal is a large and powerful framework, and you'll no doubt run into a few obstacles here and there, along the way. When you do, the Drupal Handbook is available online for your use. The Drupal web site also provides forums to address questions and issues that aren't covered by the Handbook.

Sites that have successfully implemented Drupal include Spread Firefox, Kernel Trap, Linux Journal, and The Onion. I hope that you'll be next. ■

FIGURE 13



The end result. Not bad at all for the small investment we've made.

TITUS BARIK is a content application developer with an interest in open source Enterprise solutions. He has deployed both open source and proprietary content management systems successfully in corporate and non-profit environments. His personal weblog is available at barik.net, and he welcomes your comments and suggestions.