CS/VT 4990: Advanced Web Development with Django

Assignment: A Better Job Board

Objectives

The objective of this assignment is to build a better job board. Building on your knowledge with multi-user sites and roles & permissions (Kaizen, Twitterclone), this assignment introduces the use of email and messages for your assignment.

Dixie’s CIT program maintains a job board site: [http://cit.dixie.edu/citjobs/](http://cit.dixie.edu/citjobs/). The job board is old and in need of repair. There is no immediate feedback if a job is posted, so people unknowingly post the same job over and over. There are no options for formatting text – the description for the entire job posting gets smooshed into one paragraph. I’m sure you can build a better job board site. Perhaps the your job board site will have the privilege of replacing the current, outdated system, and your legacy will live on for years to come in CIT lore.

Requirements

- At a minimum, a job post has:
  - An employer profile
  - A title
  - A job description
  - A job type (Part Time, Full Time, Internship)
  - Wage (optional)
  - Publish Date
  - Expiration Date
  - Active / Inactive
- At a minimum, an employer profile has:
  - A user
  - A company name
  - A company description
  - A company website
  - A preferred contact email
  - A preferred contact name
- Views
  - A page that lists the latest 20 or so job postings from all employers that’s visible to the public
  - A job detail page that’s visible to the public
  - Login, Log Out, Create Account, Forgot Password pages
  - Create and edit an employer profile page
  - As an employer, I want to create a new job post
  - As an employer, I want to see a list of all of my jobs and their status (active, inactive, etc.)
  - As an employer, I want to Edit an existing job, including deactivating or activating a job
  - Sign up for new job post announcement emails
  - Unsubscribe to job post announcement emails
- Tests
  - Jobs after the expiration date shouldn’t be publicly visible on the site, even when trying to visit the job post’s ID in the URL.
  - You shouldn’t be able to publicly view inactive jobs either
  - You can’t edit other people’s jobs

Extra Credit Ideas

- An easy contact form on the job post detail page so that job applicants can contact the job poster easily
- A moderator (through the admin?) approves each job posting to make sure it’s made in good taste

Hints

- Django’s email documentation: [https://docs.djangoproject.com/en/1.5/topics/email/](https://docs.djangoproject.com/en/1.5/topics/email/)
- Django’s messages framework: [https://docs.djangoproject.com/en/1.5/ref/contrib/messages/](https://docs.djangoproject.com/en/1.5/ref/contrib/messages/)
- Settings.py for the messages framework context processor:

```
import django.conf.global_settings as DEFAULT_SETTINGS
```
To override the URL that new registrations come to:

In your project's urls.py:

```python
defaul_settings.TEMPLATE_CONTEXT_PROCESSORS + ( 'django.contrib.messages.context_processors.messages',)

from jobboard.views import MyRegistrationBackend

urlpatterns = patterns(',
    ... url(r'^accounts/register/', MyRegistrationBackend.as_view(), name="registration_register"),
    ...)
```

And in your project's view's file (e.g. jobboard.views):

```python
from registration.backends.simple.views import RegistrationView
class MyRegistrationBackend(RegistrationView):
    def get_success_url(self, request, user):
        return reverse('jobboard:index')
```

How to render a message body like you would template:

```python
template = loader.get_template('jobboard/email.txt')
context = RequestContext(request, {
    'name': subscriber.name,
    'post': post,
})
body = template.render(context)```