CLC – Networking Site

Gary Sundquist

Grand Canyon University: CST-256

4/22/2018

Description of project

Week 1:

Creation of the Networking Site. Users can now login and register to the site. Error handling is present if there is a username already in the database when someone tries to register. There is also error handling for unfilled out form options and also an invalid username and or password on login.

Q: Address the use of the Laravel framework and its advantages over “plain” PHP:

With my current understanding of the Laravel framework I have found that it’s much easier to create objects of tables by simply creating a model. This saves quite a bit of work compared to having to create a bunch of getters, setters, and another class to handle CRUD. CRUD is already handled by a parent class to the model. In the code you simply have to specify the DB class and the method you are using along with the model you’re using.

Week 2:

Created a profile and admin panel page. The profile page allows the logged in user to manage their user profile by changing their name, password, email, address, background, and skill information. The admin panel page allows an admin user to suspend or delete existing users.

Week 3:

Added additional fields to the profile page allowing the user to input education, job experience, and skills. Also edited the user deletion function to confirm with the admin prior to deleting the user that they really do want to delete the user. Added the ability to add job postings and delete them.

Week 4:

Added affinitygroup to the user profile page where the user can select an existing affinity group to join. Also added affinitygroup to the admin management page so that new affinity groups can be created, updated, or deleted. Added a job posting page where you can look up jobs by typing in a name or description and then the table will be updated with records similar to what your search parameters are. The description in the table is parsed to only show the first 4 characters of the description followed by … Once the job page button is clicked the user will be sent to the page of the job posting where they can see the entire job description.

Week 6:

Added additional functionality to the Job Search page allowing you to customize the order in which your search results are displayed. They can be ordered by ascending or descending and can be ordered by date, salary, or type (location). You can also add jobs to your cart on the job search page. Added a cart page to view the jobs that have been added to your cart. Added a apply page where you can apply for jobs.

Week 7:

Added the ability to delete jobs in the job cart. Fixed bugs on the adminManagement page where you had the ability to submit requests to create jobs or affinity groups with missing data that would cause the application to break. Added input fields for job postings so users can include a url, location, and salary.

Files for this project

https://github.com/Ninjaroz/CST256

List of Modules

Controllers:

* userController
* managementController
* jobPostingController

Models:

* User
* Roles
* JobPosting
* AffinityGroup

Views:

Layouts:

* Sidebar
* Header
* footer

Management:

* adminManagement

userViews:

* failedLogin
* login
* logout
* registerUser
* registerUserFailed
* userProfile
* userRegistered
* verifyUser

jobViews:

* jobSearch
* testJob
* apply
* jobCart

home

Routes:

* web

User Guide

Week1:

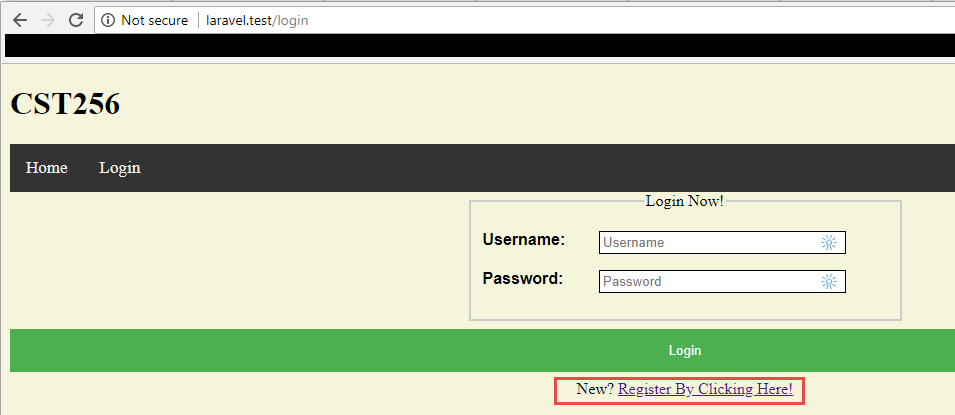
In order to login you will need to access the login page by going to: <http://laravel.test/login>

Once there you can use a user that has already been created for the site:

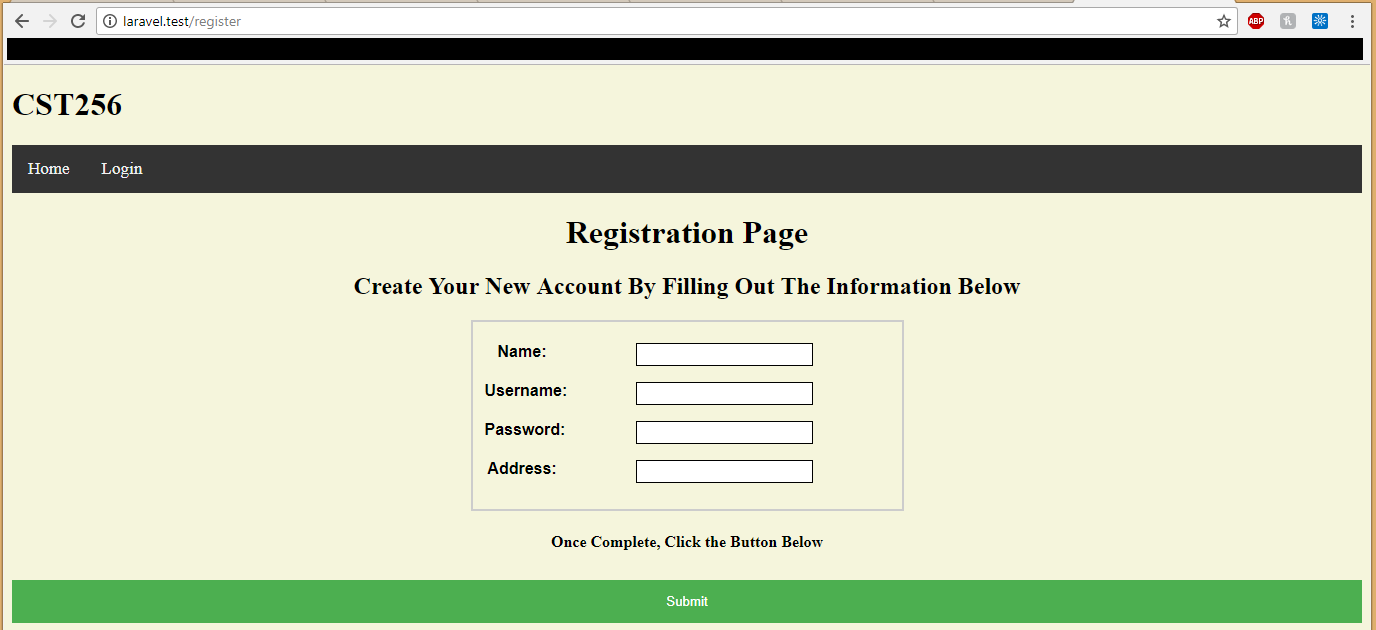
Username: test77

Password: test77

You can also register for the site by clicking on the link below the login button “New? Register by Clicking Here!”



You can register for the site by filling out the form below and clicking the submit button:



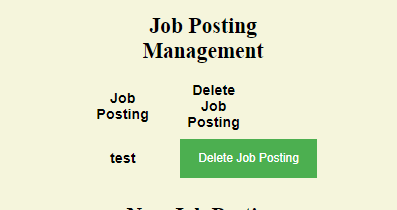
Week 2:

Once you successfully login using a user you will be able to see the profile link and access your profile. On the profile page you can update any/ all of the fields for your profile. Once you click the update button an error prompt will appear for a textfield or the page will reload with the updated information.

If your user has the admin role they will have access to the admin panel page. On the admin panel page it will automatically load all of the users in the database. You can delete or suspend a user by clicking the “Delete User” or “Suspend User” button next to the users name.

Week 3:

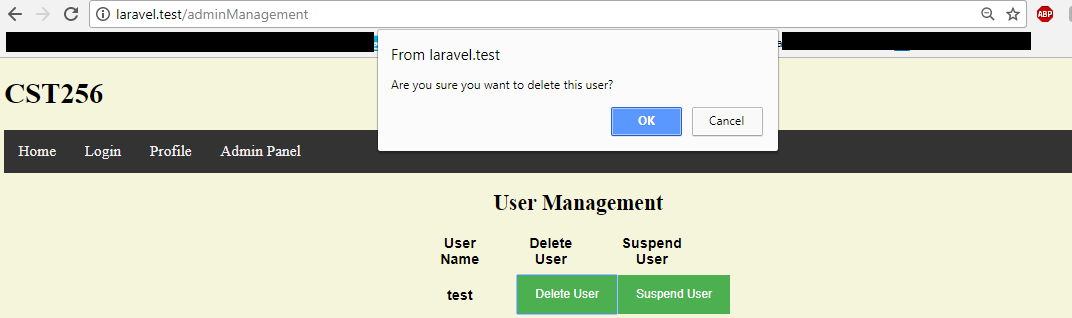
Once logged in as an administrator you can visit the admin panel and add and delete job postings. To delete a job posting you would just need to click the “Delete Job Posting” button next to the name of the job posting in the Job Posting Management table:



You can create a job posting by adding a name and or description and clicking the “Create Job Posting” button:



Added functionality to protect the admin user from accidently deleting a user has been implemented:

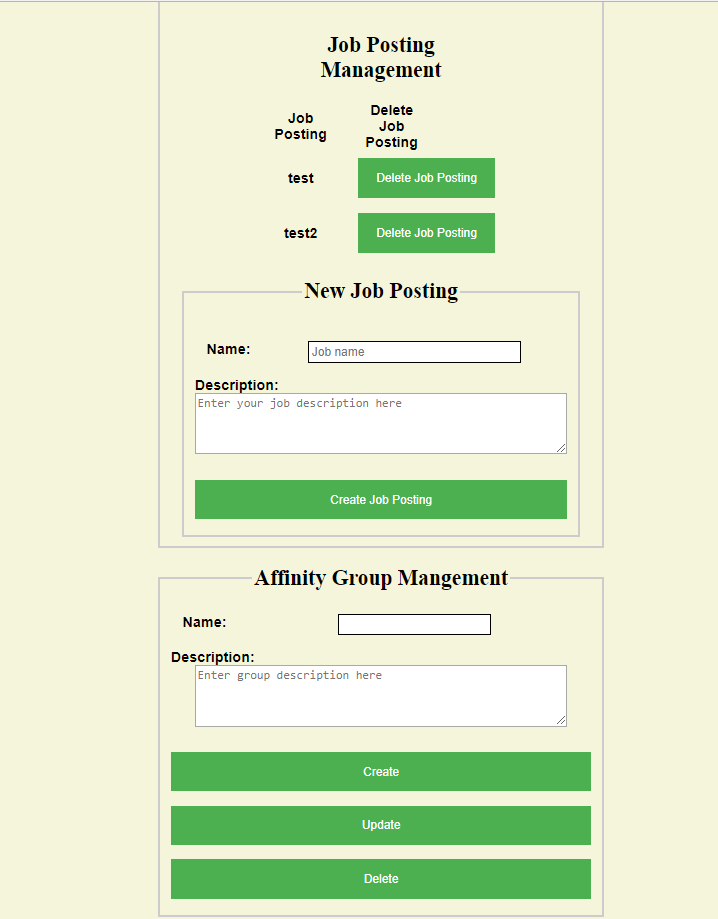


Week 4:

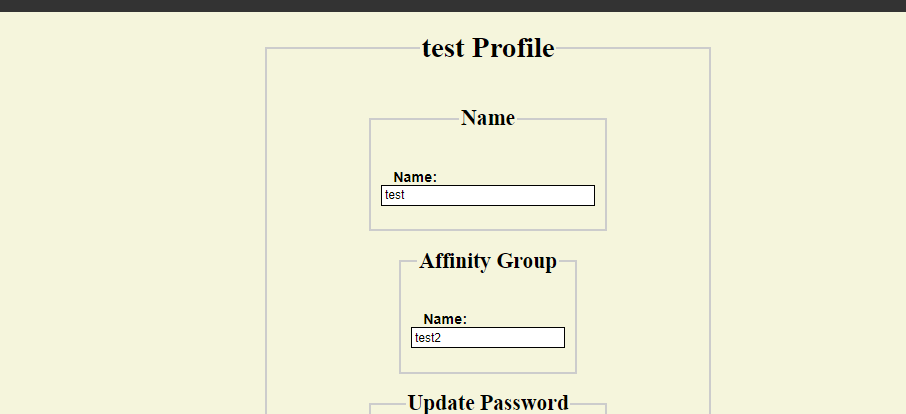
Job posting page:



Admin management affinity group:

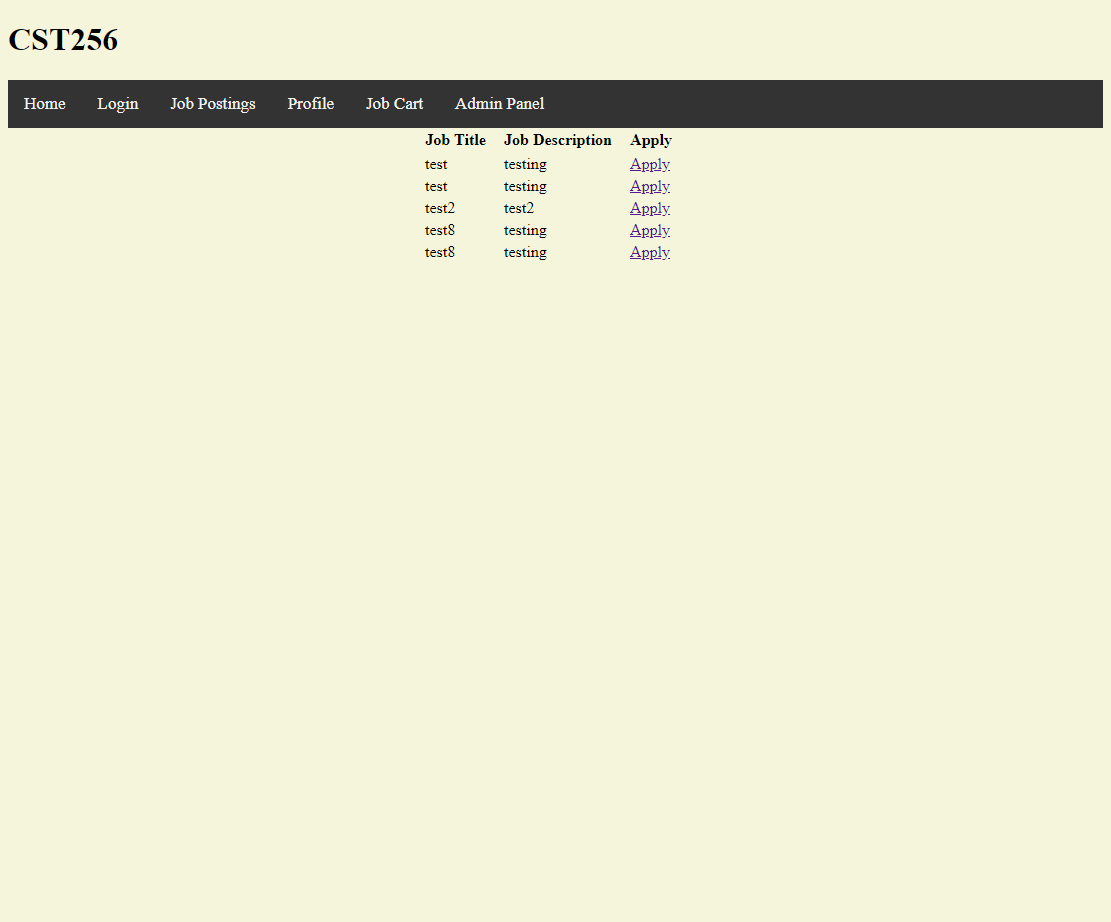


User profile affinity group:

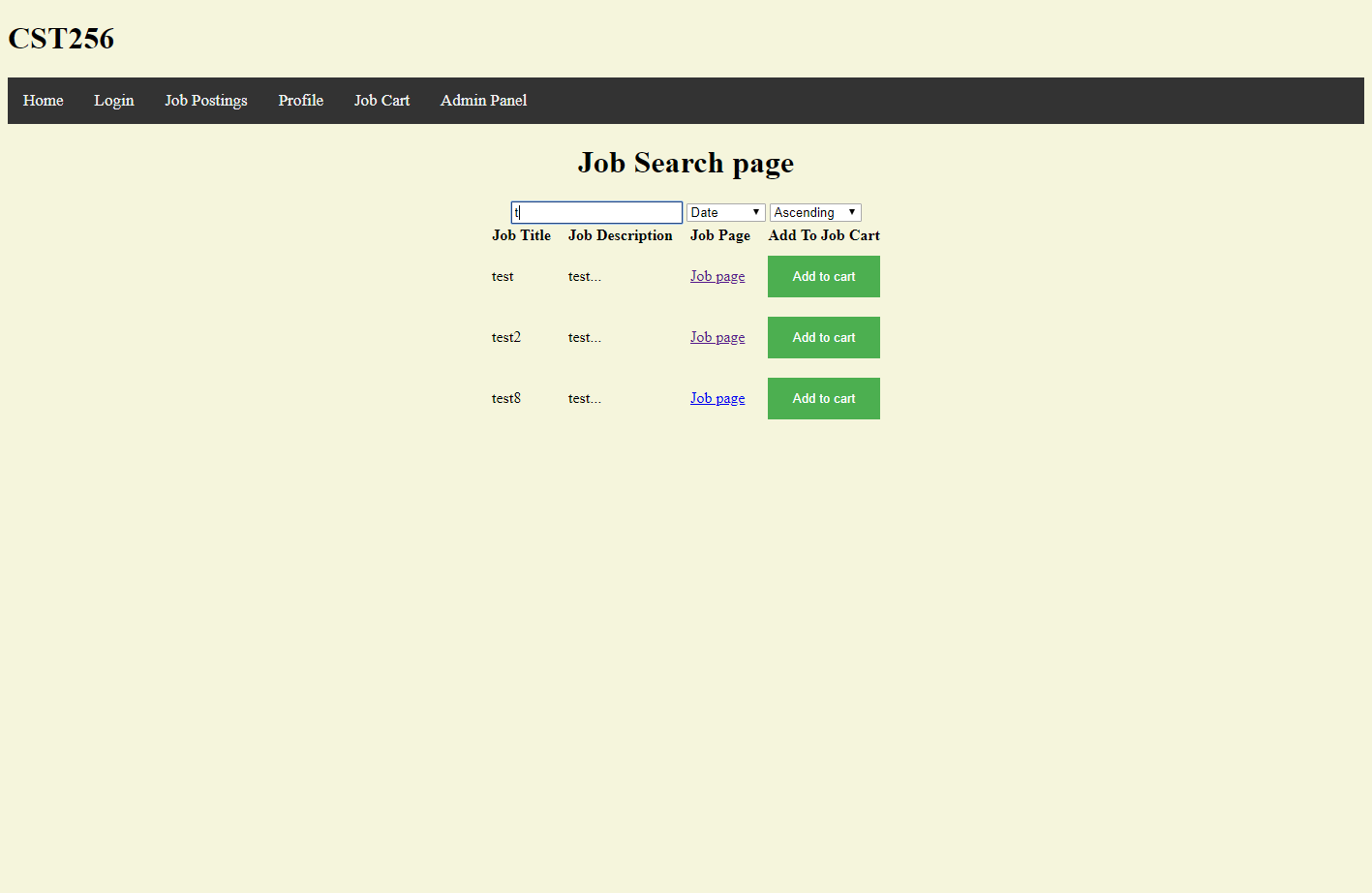


Week 6:

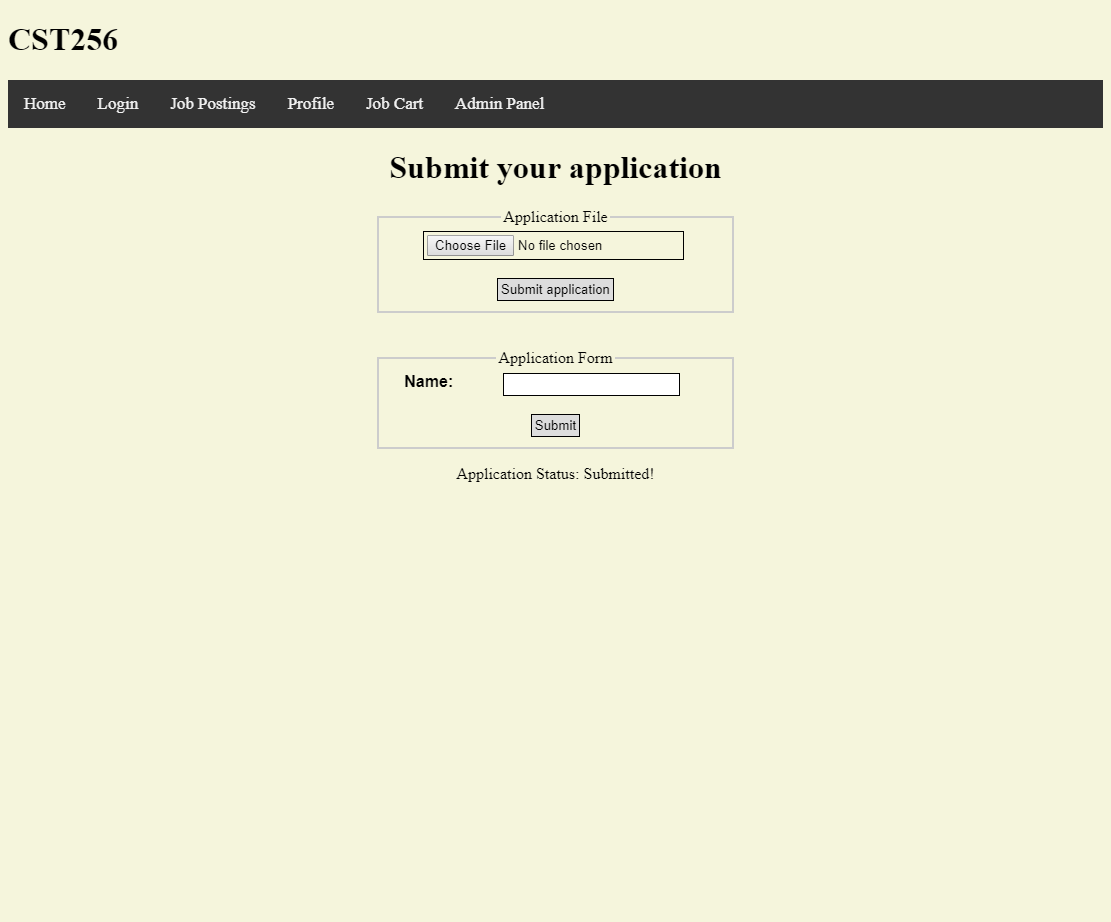
You can view the cart page as long as you are logged in and click “Job Cart” if no items are in your cart currently it will display that there are no items in your cart.



Job cart page you can now change the order in which results are displayed based on date, salary, or type (location). You can also add jobs to your cart on this page:

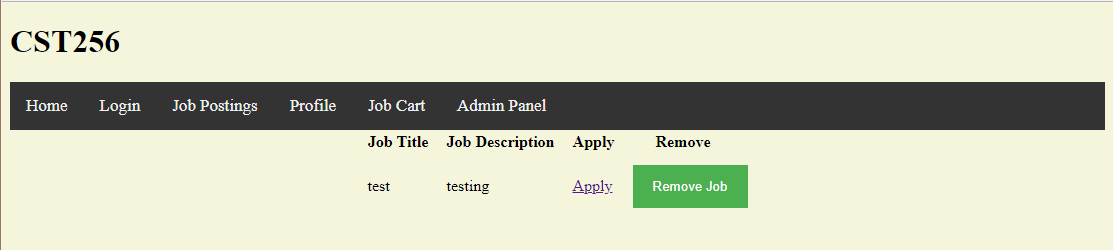


Apply page can be accessed by clicking the apply button on the cart page. Once here you can submit a file application or form application. When submitted the status will change to “Submitted!”



Week 7:

Added ability to remove job postings in cart:



Added url, location, and salary to admin management page to be loaded when a job is created. The only required field is a job name.



Improvements

Week 1:

I would like to add a different background color to the current page in the navigation bar. I would also like to change the way error handling is handled and instead of having a different login page come up if an invalid username or password is entered, change the existing login page legend to state that there was an invalid username or password entered.

Week 2:

Display an appropriate error message on the update profile page if the user specifies an incorrect user password. Currently the page is just reloaded with no changes made to the profile. Also proper error handling for the current user signed in if you try to delete your own user. You can currently suspend your user but cannot delete it. I do not think it makes sense for you to delete your own user, may just remove your user from coming up in the list entirely.

Week 3:

As of right now besides the changes I have previously included, I would like to clean up my web pages to make them look a little more appealing. As of right now they are really dull. The changes I made this week I feel are solid and do not need any alterations.

Week 4:

I would like to display the affinity group description whenever a different affinity group is selected from the drop down box. This way the user knows what the previous data was prior to updating. I would also like for the page to refresh to show the newly created affinity group and or updated group. Would also like to add error tracking for the user profile page to where if an affinity group the user tries to add does not exist it will display an error that the group does not exist. Right now if the user tries to add an affinity group that does not exist it simply will not update their affinity group.

Week 6:

Currently there is no way to tell what job you are applying for. The apply page is just a generic page in which you can apply for a job. There is also nothing preventing you from submitting nothing on the apply page. The only requirements were for there to be an apply page which was created. Also on the job cart page there is currently nothing from stopping the user from adding the same job to their cart multiple times. The admin panel currently does not allow you to put in a salary, location, or date. I added functionality to input the current date when a job is added to test the date and just performed update statements to test salary and location.