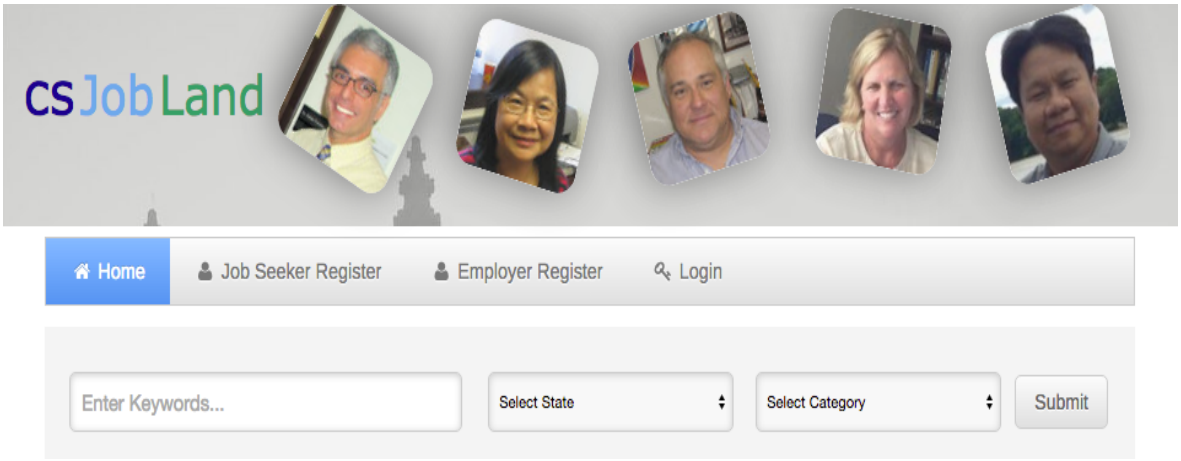


CSJobLand

Web Site Development

Fall 2015
IAD Final Project

Lixia Zhao
Edith Castro Granja
Shanshan Jiang
Xueyue Zhang
Zackline Mendy

Table Of Content

Introduction **3**

Implementation **4**

1. HTML Page Structures 4

1.1 Header and Footer

1.2 Navigators

1.3 Job List and Search Page

1.4 Job Details Page

1.5 Registration and Login System

1.6 Job Seeker Profile

1.7 Job Seeker Edit Profile

1.8 Employer Job Listing

1.9 Employer Edit A Job

1.10 Employer Add A Job

1.11 Employer Delete A Job

1.12 Employer Browse Job Seeker

1.13 Employer Search For Job Seeker's Profile

1.14 Logout

2. Data Attributes and ER Diagrams ER Diagrams 21

Future Work **26**

Conclusion **27**

Reference **27**

Introduction

The general purpose of our project is to build a job board website especially for SJU computer science students. SJU students can browse the latest jobs posted in the website. They can also search jobs by job categories, state and keyword, or combines any of the three searching fields.

There are two nice functions for this website:

1) If a user registers and logs in as a job seeker, he/she can add his/her profile by filling in his/her experience in a form. An html profile page generates based on the form filled by the user. The job seeker can also update his/her profile. Therefore, his/her profile can be browsed and searched by a user who logins in as an employer;
2) If a user registers and logs in as an employer, he/she can add jobs by filling in form. The employer can also edit or delete his/her job list. As an employer, he/she can also browse and search job seekers' profiles.

Now, most medium or large size companies have an online job board. This project is a good practice to apply our skills learnt in the

classroom setting to the real life. It also could potentially benefit SJU students majoring in the computer science.

Implementation

The website is built by using PHP, html, JavaScript/jQuery, Ajax, CSS and MySQL database structure. It also implement HTMLkickstart. For example: the icon classes are from kickstart package.

```
<li class="current"><a href="index.php"><i class="icon-home"></i> Home</a></li>
<!--<li><a href="jobs.php"><i class="icon-desktop"></i> Browse Jobs</a></li>-->
<li><a href="SeekerReg.php"><i class="icon-user"></i> Job Seeker Register</a></li>
<li><a href="employerRegistr.php"><i class="icon-user"></i> Employer Register</a></li>
<li><a href="login.php"><i class="icon-key"></i> Login</a></li>
```

1. HTML Page Structures

All of the web pages of the website include the header and footer. Many of them also include navigation menus. In order to avoid the duplication of workload and to manage the codes easily, all of the html pages except the login page (login.html) were saved as PHP files. Therefore, they can be dynamically included in the other PHP files. For example:

index.php

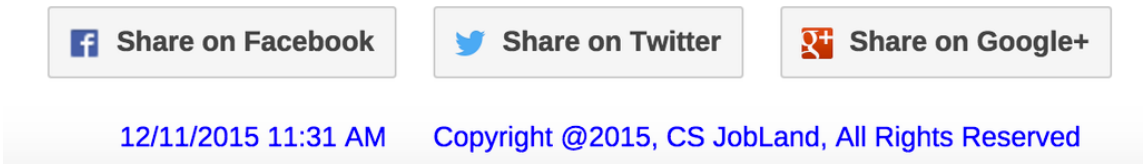
```
1
2   <?php include ("header.php"); ?>
3   <?php include("navigator.php");?>
4   <?php include("jobsearchbox.php");?>
5   <?php require ("conn.php");?>
6
7
8       <div id = "myDiv" class="col_12 column">
9           <?php include("latestjobs.php");?>
10      </div>
11
12      <div class="clearfix"></div>
13      <?php include("footer.php");?>
14  </body>
15  </html>
```

In this way, the codes are managed in a modularized way.

1.1 Header and Footer

Every page of the website has header and footer.

Header shows the title of the website: csJobLand and the pictures of faculty in the SJU CS department. Using CSS animation, such as transfer and rotate, these pictures can rotated when the mouse hovers over them.



Footer includes Facebook, twitter, Google+ icons and the current time. These icons can be accessed to share the website. The current time is generated by JavaScript functions.

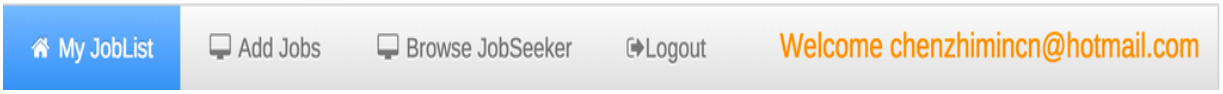
1.2 Navigators

Every page has a navigator. There are three sets of navigators. The home page, registration pages and login page have the same navigator. The menu highlights with blue when in the current page.

If a user login in as a job seeker, the user will see a different navigator:

The navigator includes my profile, edit profile, browse jobs, logout and welcome message that shows the name of the user.

If a user login in as an employer, the user will see his/own navigator:

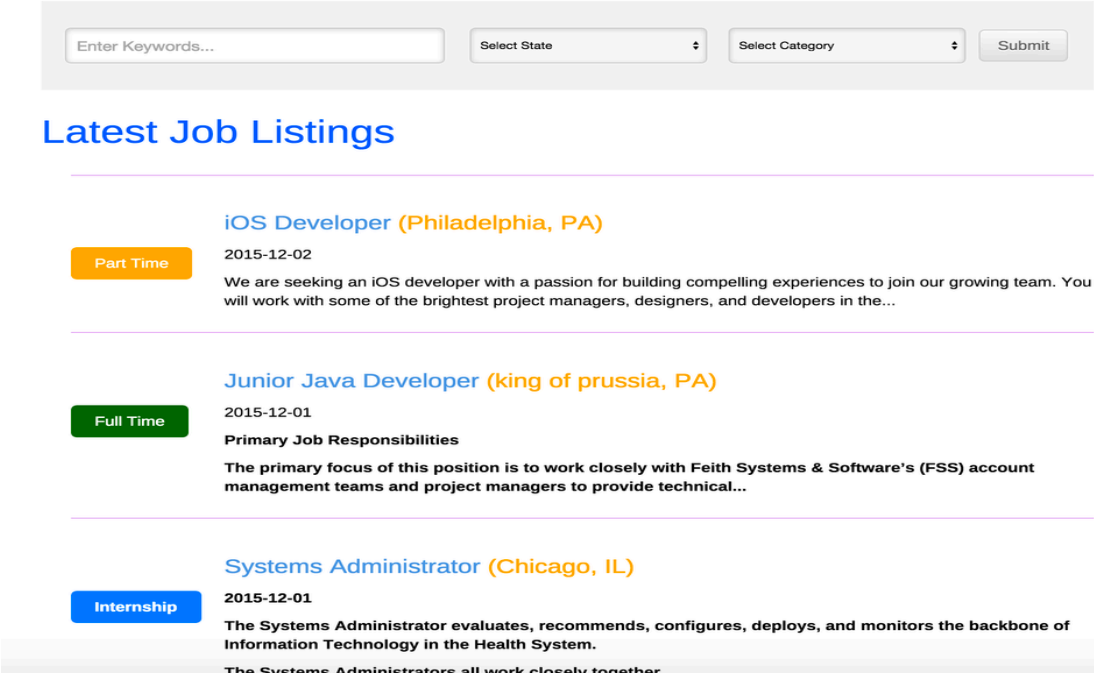


The navigator includes my job list, add jobs, browse jobseeker, logout and welcome message that shows the email of the user (The employer table does not have name attributes).

The navigators function as the restriction of user's privilege granted based on their type (employer or job seeker).

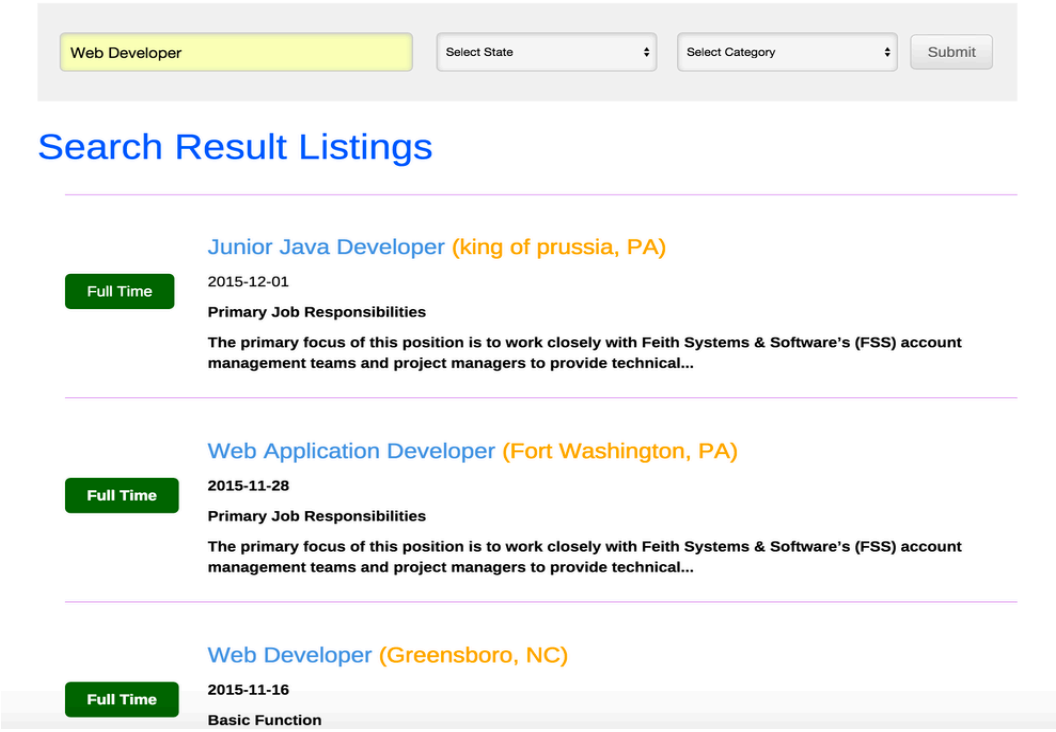
1.3 Job List and Search Page

Users can browse the latest job list and search job even they do not login to the website.

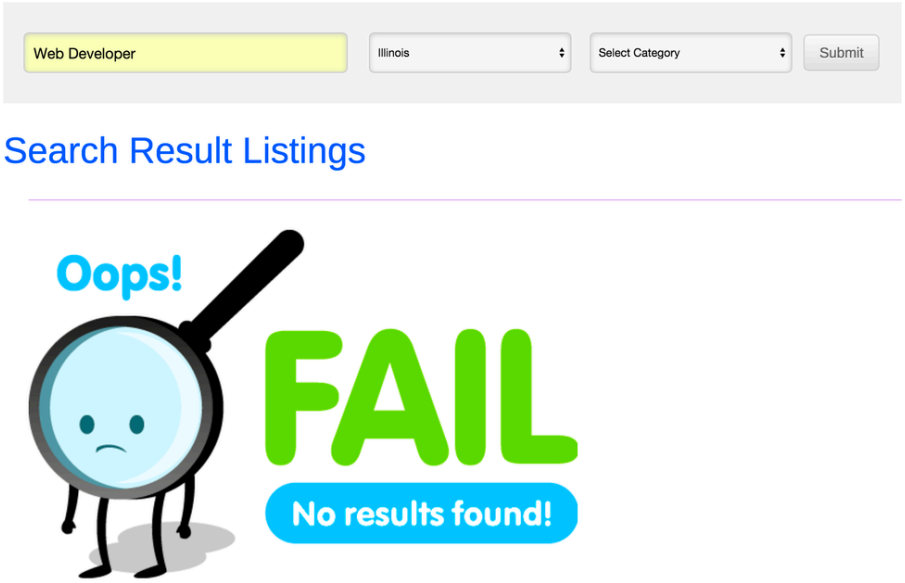


The "Latest Job Listings" lists the jobs ordered by the date added. The list shows the type of jobs (full time, part time and internship) in different colors, job title, location, post date and abbreviation of the job description.

The search can combine keywords (title and description) or state or category or one of the three. The layout the result page is similar to the "Latest Job Listing":

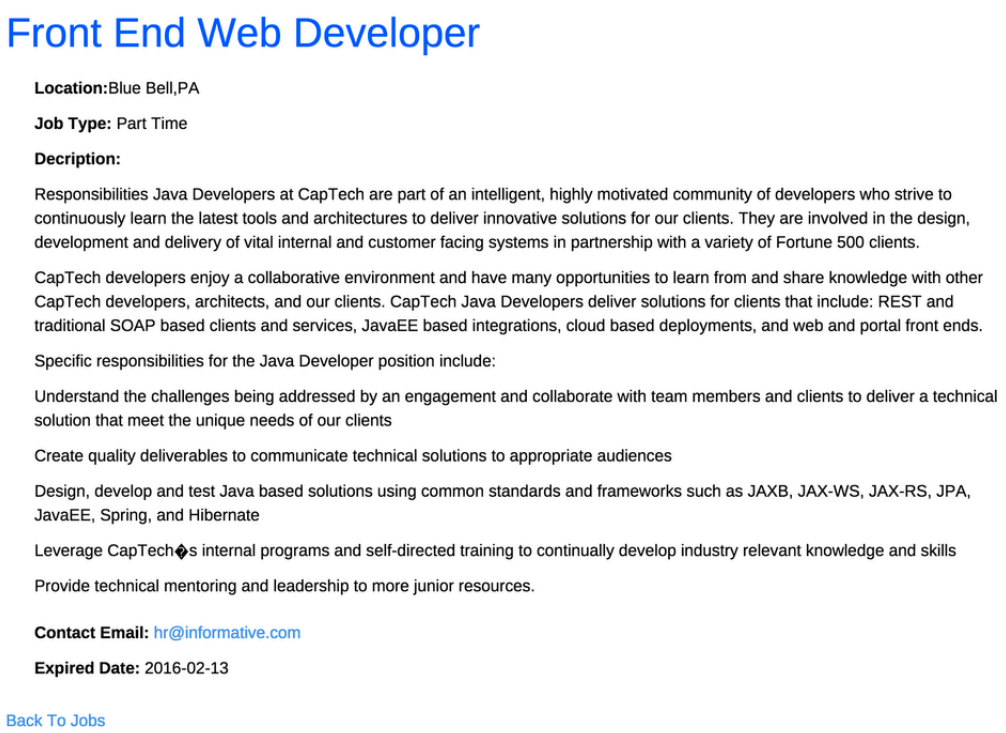


If no result was found, an image will show:



1.4 Job Details Page

Click the job title, the details of a job will show up:



Front End Web Developer

Location: Blue Bell,PA

Job Type: Part Time

Decription:

Responsibilities Java Developers at CapTech are part of an intelligent, highly motivated community of developers who strive to continuously learn the latest tools and architectures to deliver innovative solutions for our clients. They are involved in the design, development and delivery of vital internal and customer facing systems in partnership with a variety of Fortune 500 clients.

CapTech developers enjoy a collaborative environment and have many opportunities to learn from and share knowledge with other CapTech developers, architects, and our clients. CapTech Java Developers deliver solutions for clients that include: REST and traditional SOAP based clients and services, JavaEE based integrations, cloud based deployments, and web and portal front ends.

Specific responsibilities for the Java Developer position include:

Understand the challenges being addressed by an engagement and collaborate with team members and clients to deliver a technical solution that meet the unique needs of our clients

Create quality deliverables to communicate technical solutions to appropriate audiences

Design, develop and test Java based solutions using common standards and frameworks such as JAXB, JAX-WS, JAX-RS, JPA, JavaEE, Spring, and Hibernate

Leverage CapTech's internal programs and self-directed training to continually develop industry relevant knowledge and skills

Provide technical mentoring and leadership to more junior resources.

Contact Email: hr@informative.com

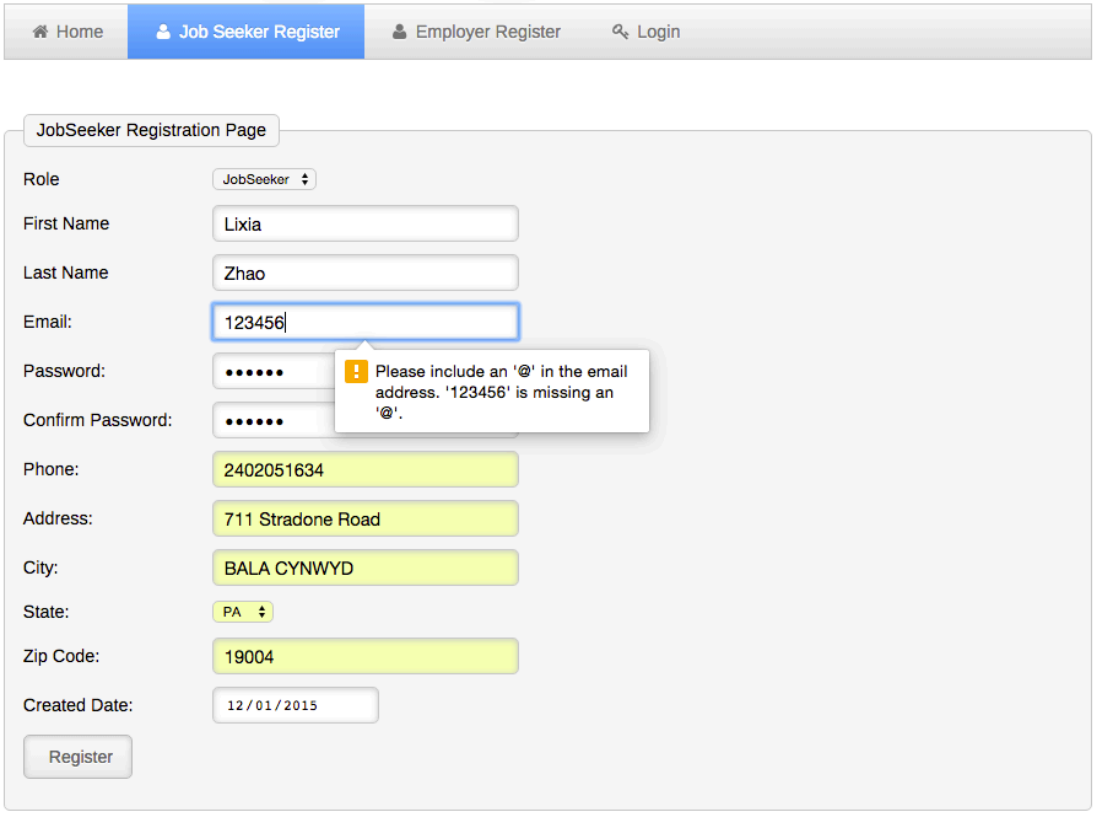
Expired Date: 2016-02-13

Back To Jobs

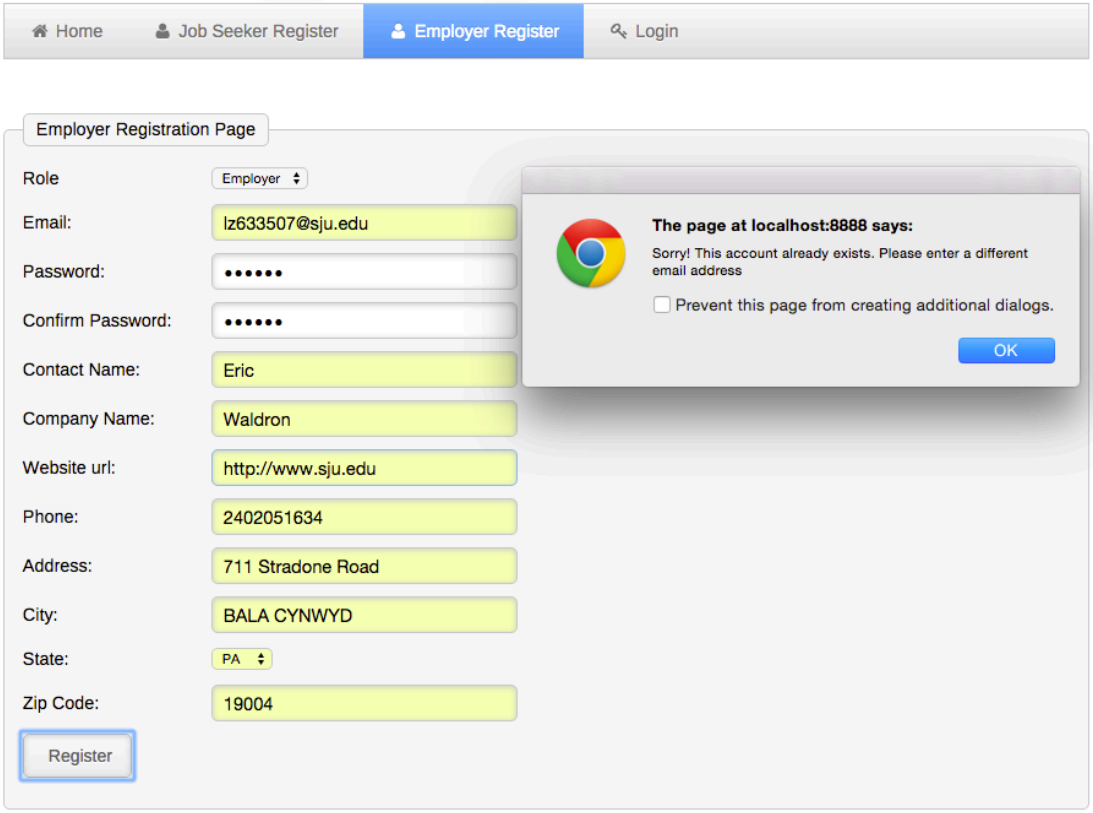
The details of a job include job title, location, job type, description, contact email and expired date.

1.5 Registration and Login System

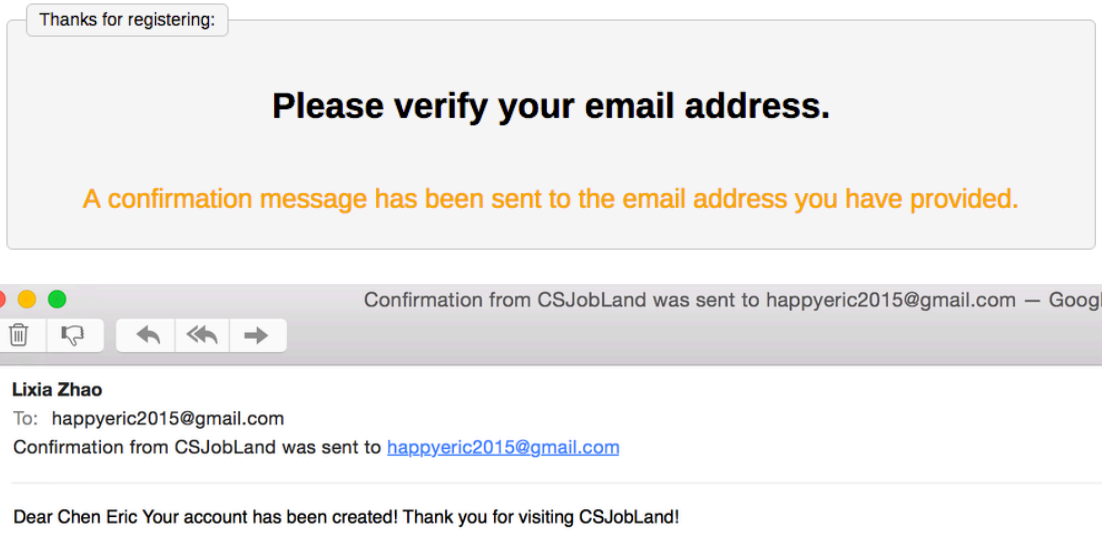
The website has two separate registration systems. One is for the job seeker, and one is for the employer.



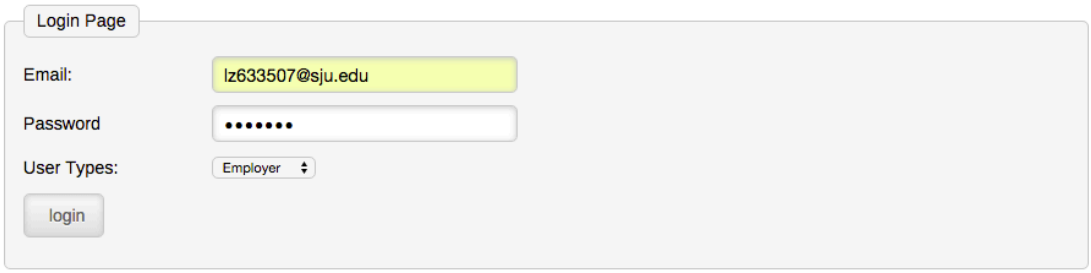
JavaScript error checking checks the formats of input fields, such as email, password, phone and address etc.



After successfully registered as an employer or job seeker, a confirmation email will send to the user's email address.

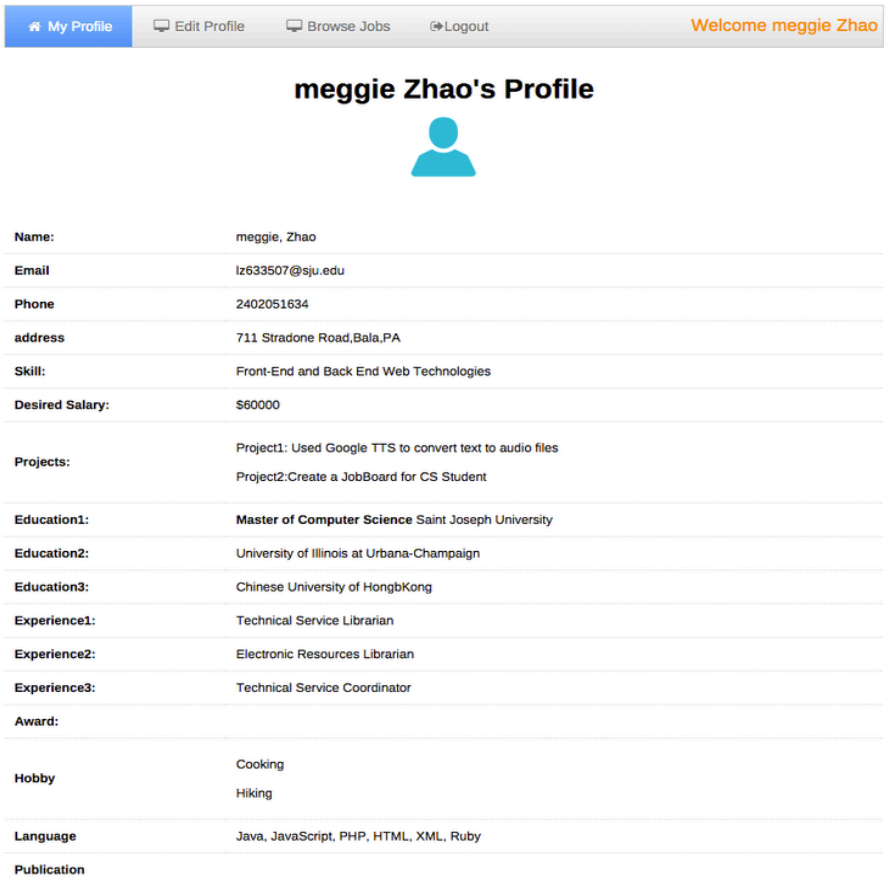


If a user login successfully as an employer or a job seeker, the session track starts until the user logout. User's email address is used as the value of the session tracking.

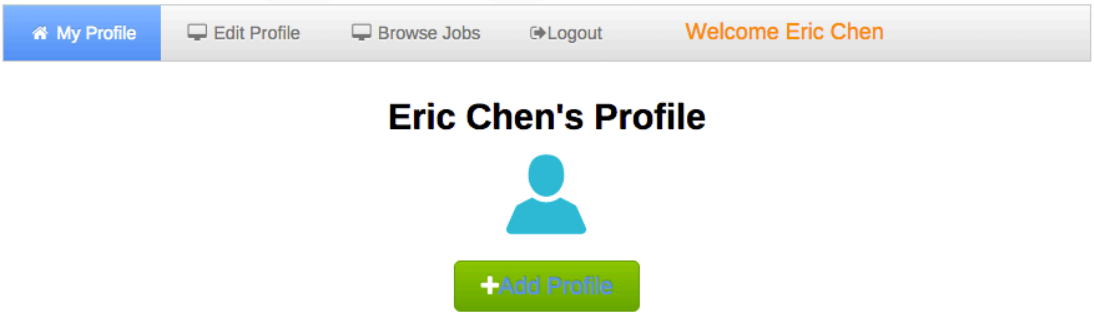


1.6 Job Seeker Profile

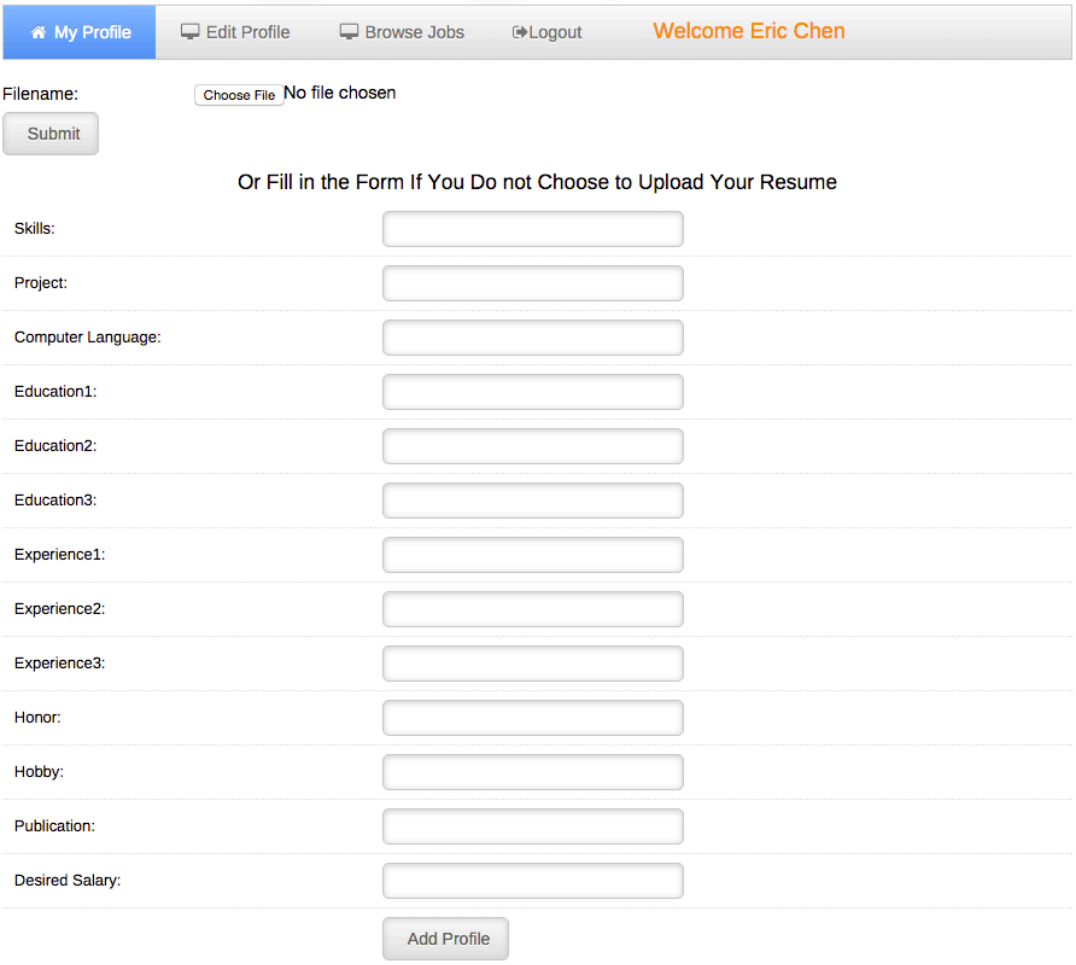
If a job seeker login to the system, the seeker will see a different navigator menu from the homepage, his/her welcome message and the seeker's profile if he/she already created one.



The job seeker can add profile if he/she does not have one.

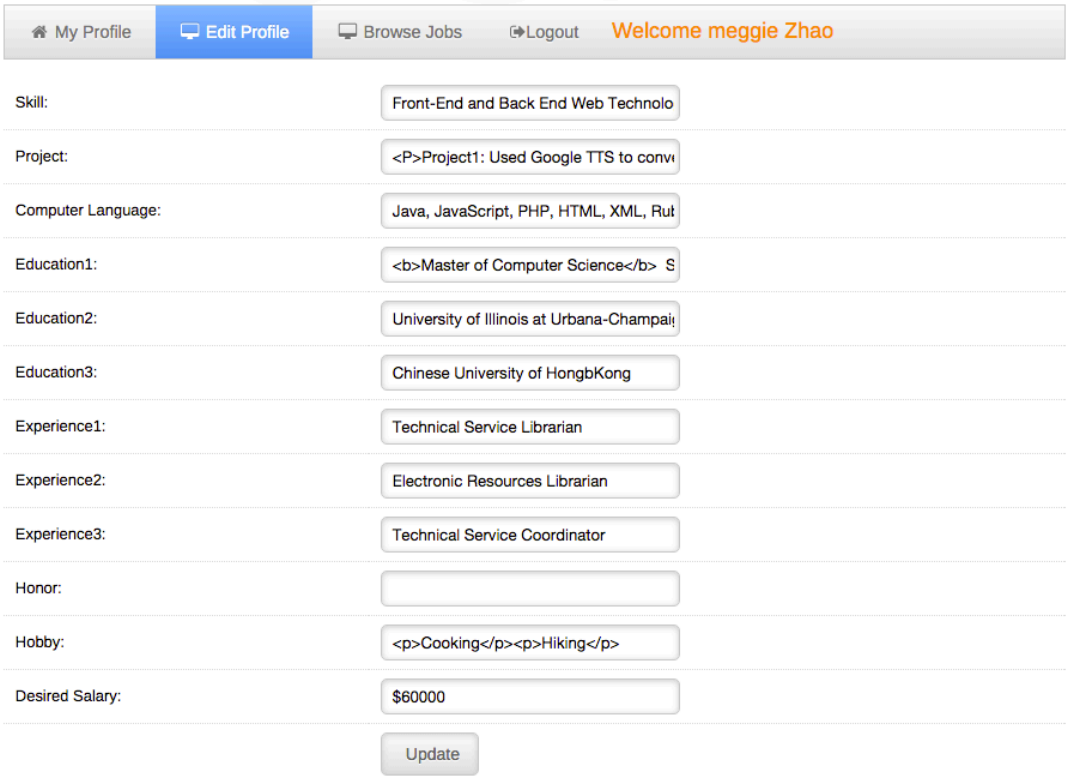


Fill in the form to create a profile:



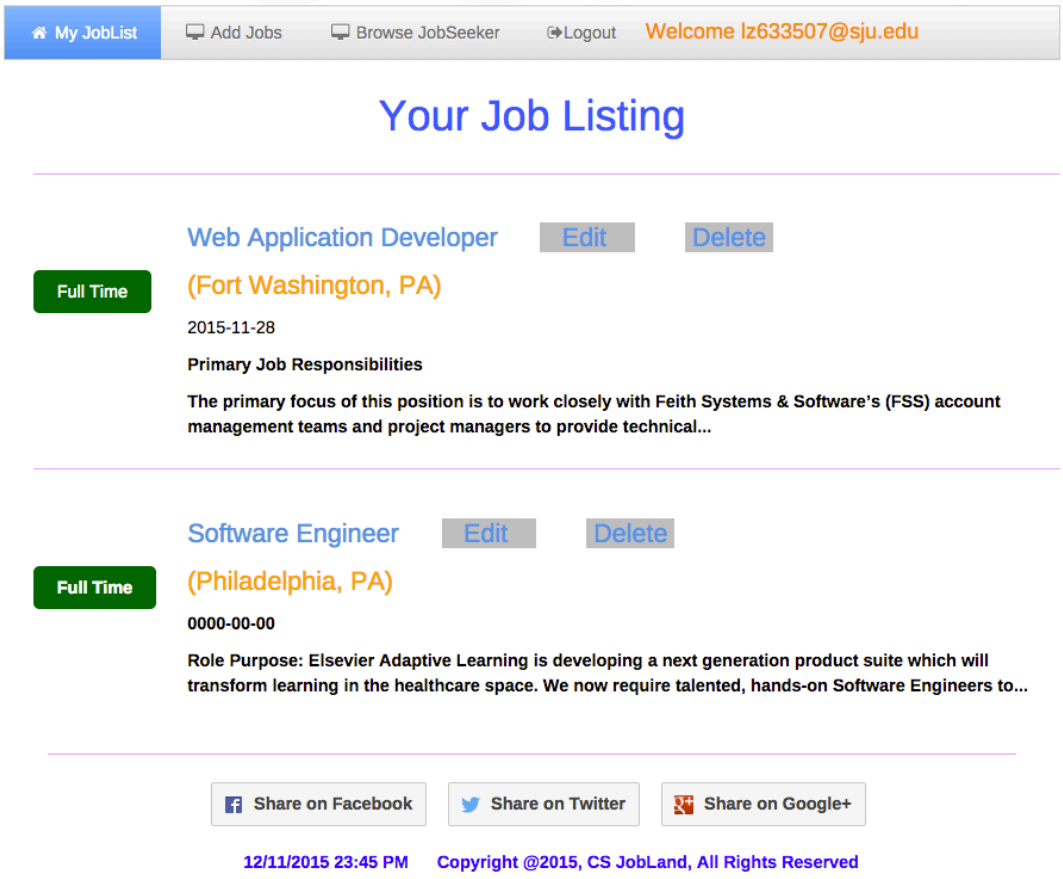
1.7 Job Seeker Edit Profile

Job seeker can also update his/her existing profile:



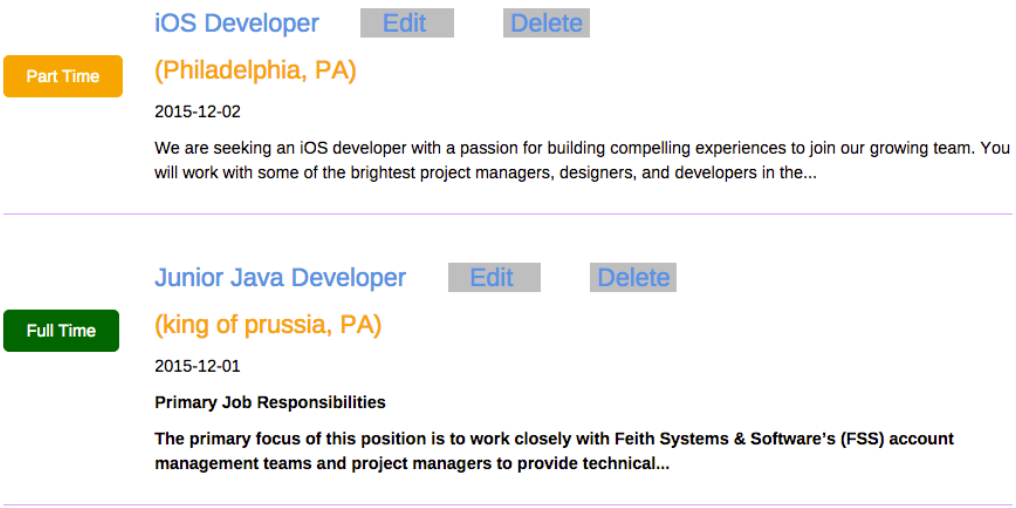
1.8 Employer' s Job Listing

If an employer already created jobs, the employer will see the list of jobs he/she created after he/she login.

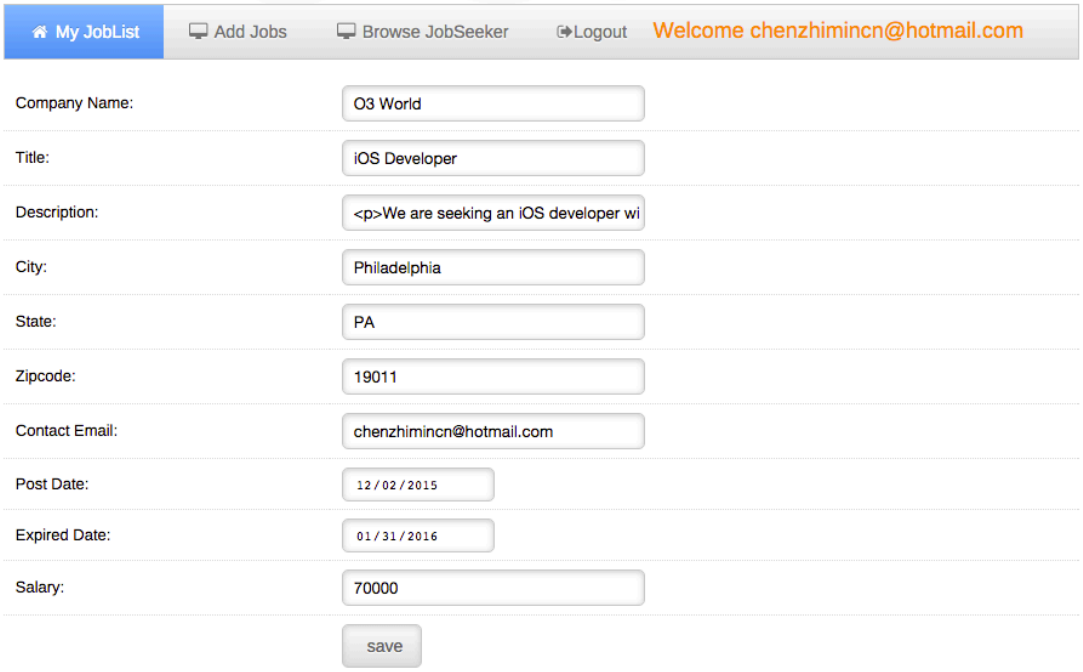


If the employer has not posted a job yet, his/her list will be empty. The employer can add a job by clicking "Add Job" button.

1.9 Employer Edit A Job

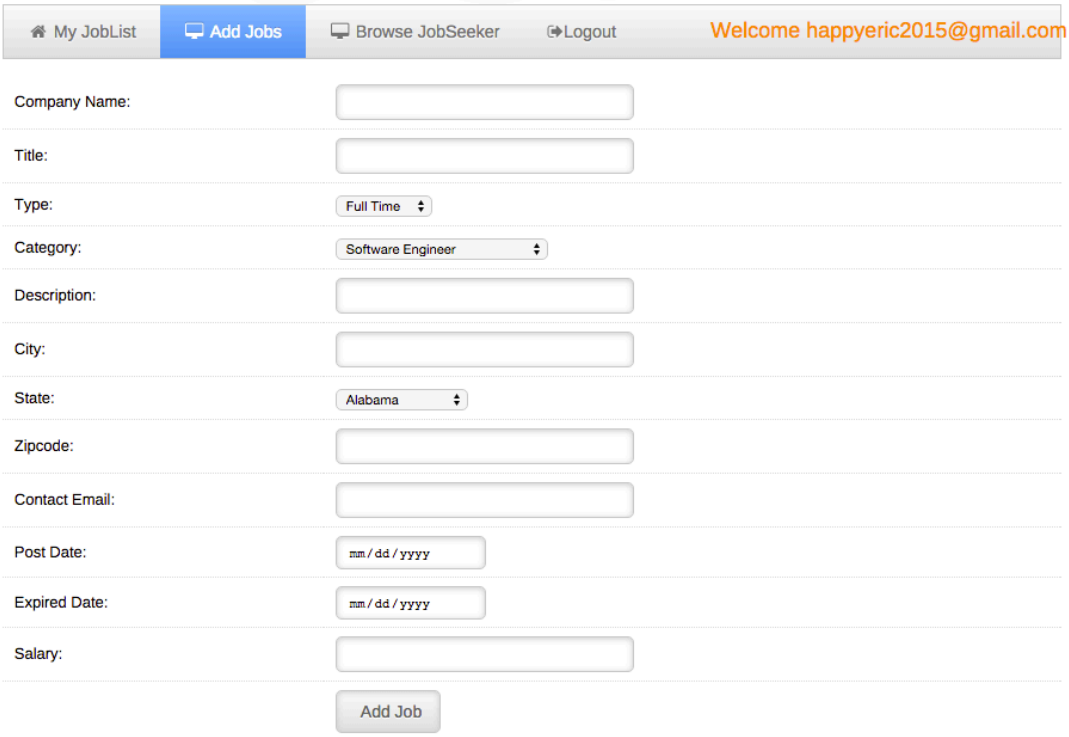


The jobs posted by an employer can be deleted or edited. The editing form shows up after clicking the "edit" button.



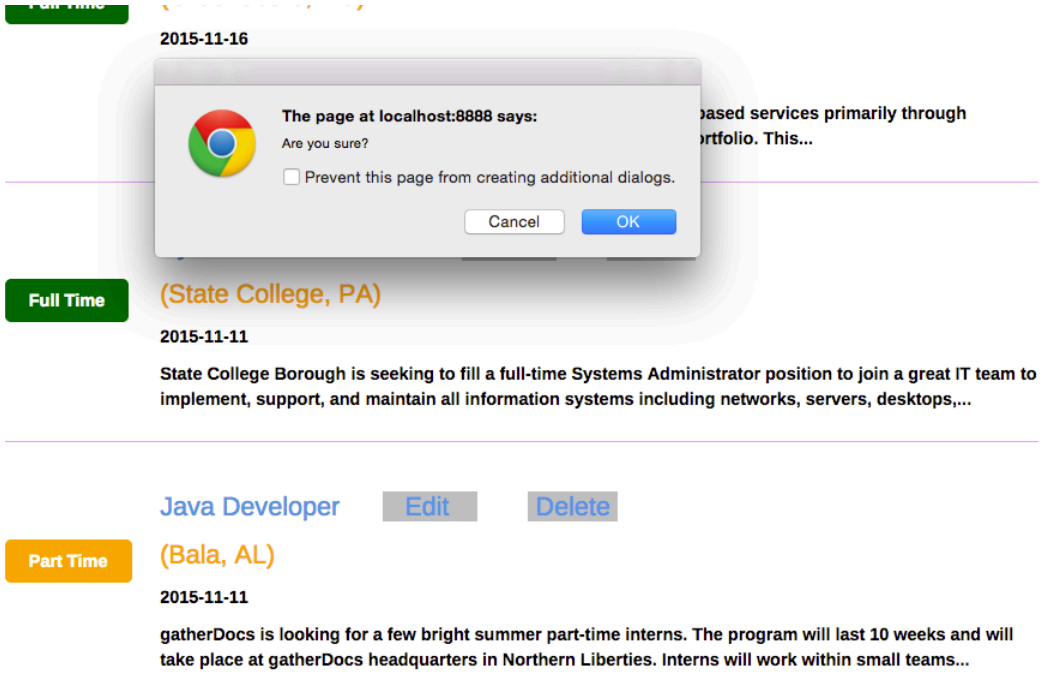
After update the form and click "save" button, the page redirects to "My JobList" which lists the employer's jobs.

1.10 Employer's Add A Job



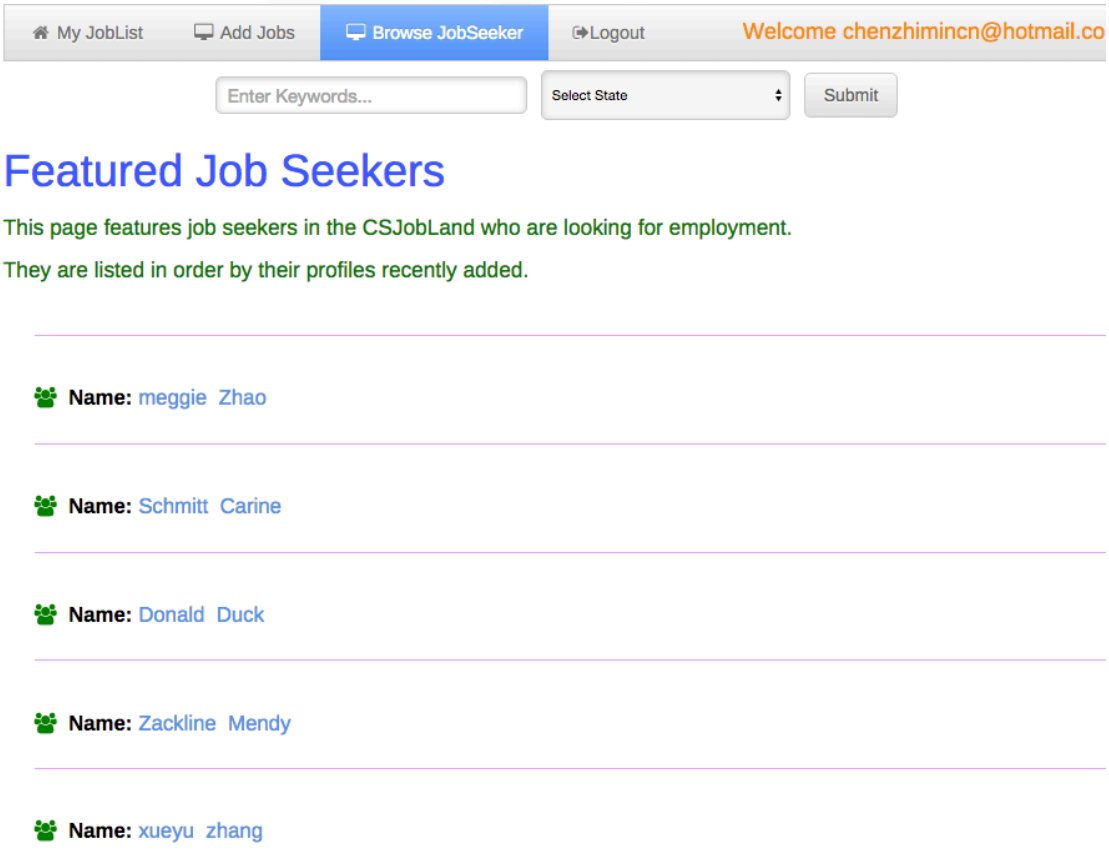
1.11 Employer Delete A Job

Employer can also delete a job by clicking delete button. A pop up message will remind if the employer really want to delete it.



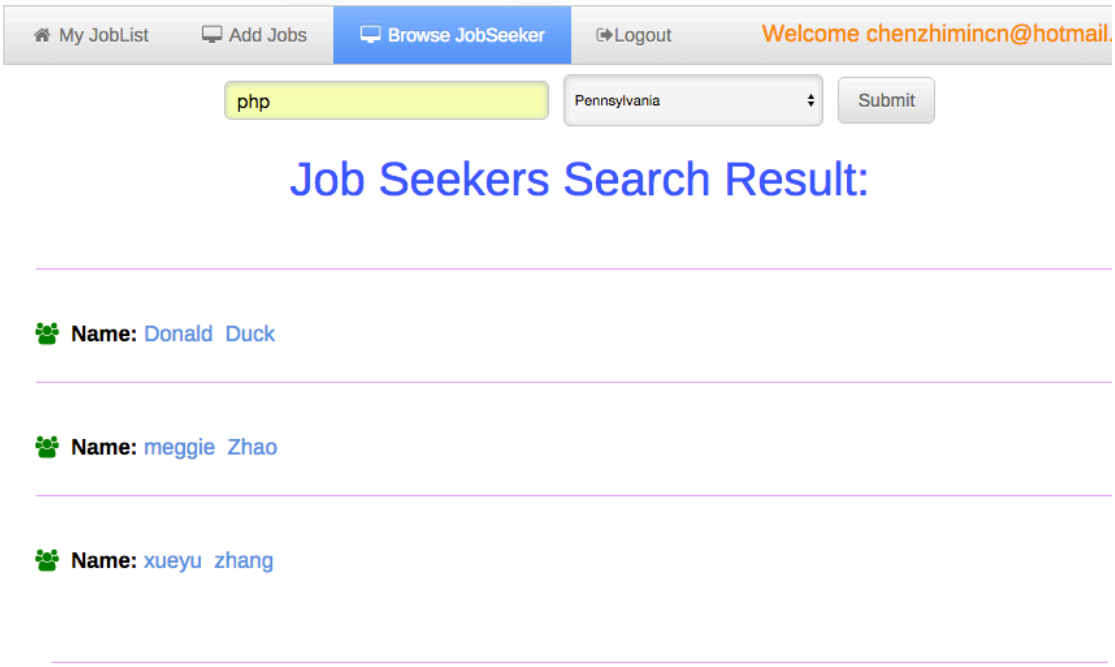
1.12 Employer Browse Job Seeker

Employer can also browse the featured job seekers. The job seekers profiles are listed in order by their profiles recently added.

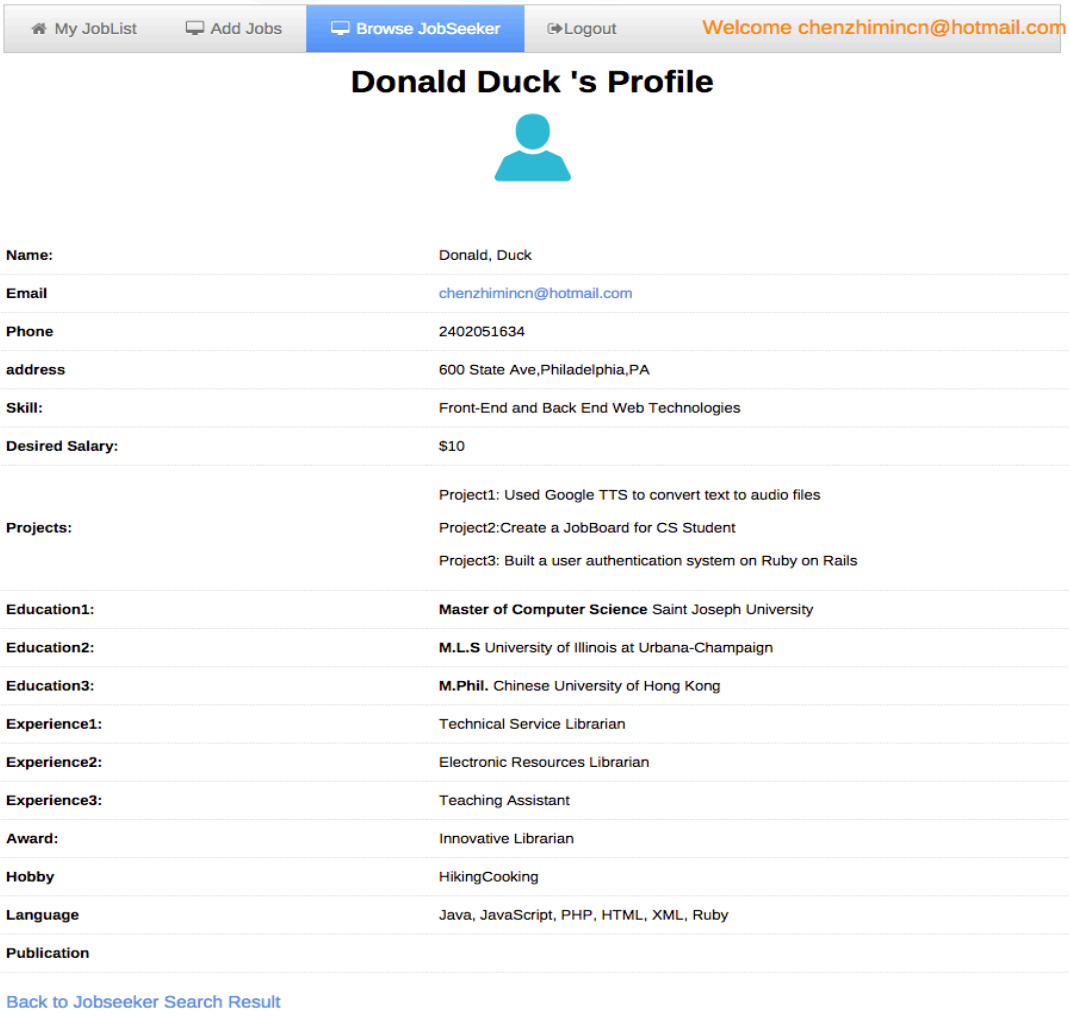


1.13 Employer Search For Job Seeker's Profile

Employers can also search job seekers' profiles by combining keywords which retrieve words from "skills", "education", "experience", "hobby", "project" and "language" from the profile table with the location.



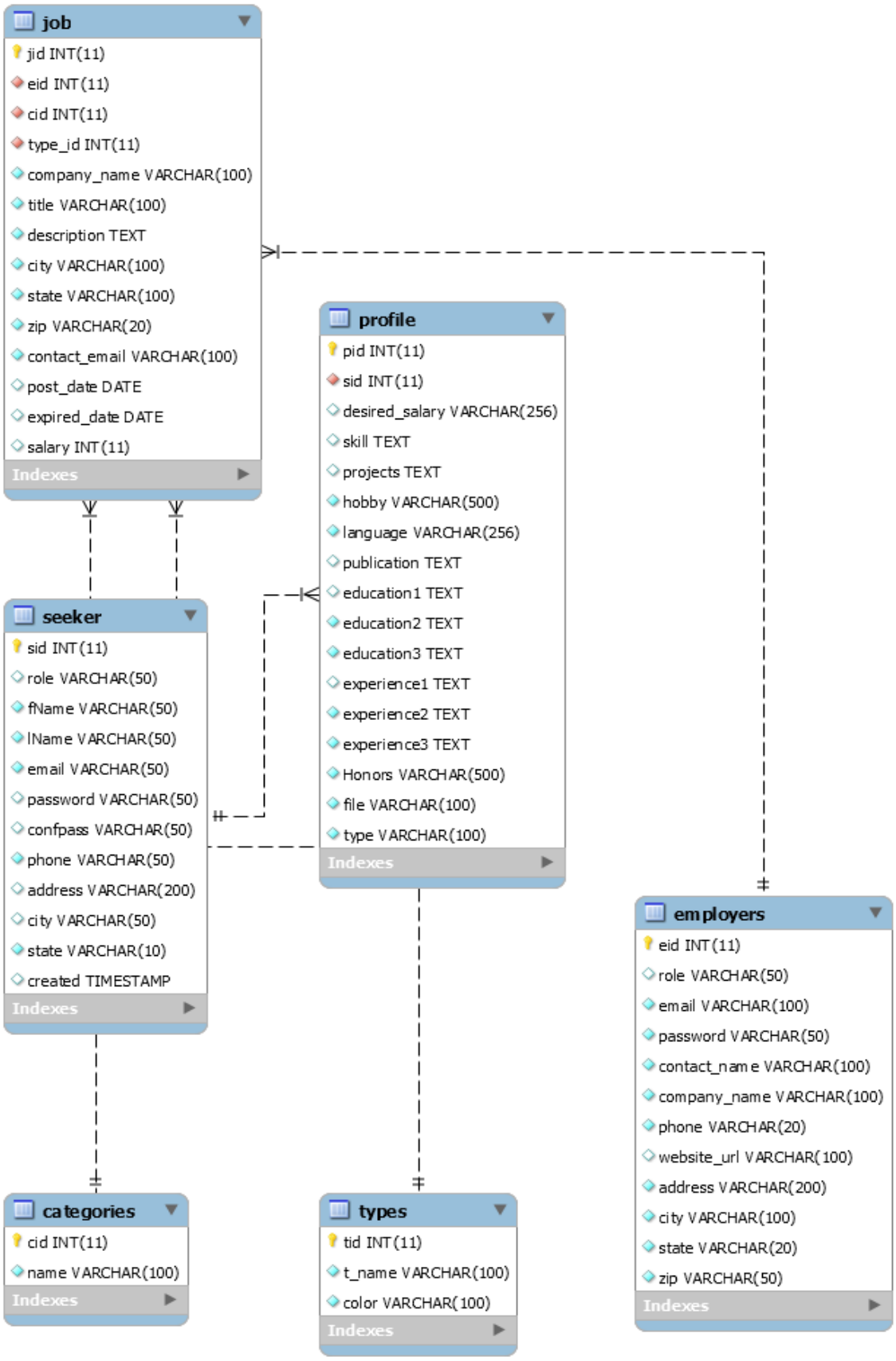
Detailed page of job seeker's profile:



1.14 Logout

Session tracking will be destroyed after user logout. The page will be redirected to the homepage.

2. Data Attributes and ER Diagrams



Categories

- Categories ID
- Name

For deciding what information needed to be stored for categories, we include the categories id and name as attributes. The categories id was added to be the primary key to allow for searching.

```
-- Table structure for table `categories`
CREATE TABLE `categories` (
`cid` int(11) NOT NULL AUTO_INCREMENT,
 `name` varchar(100) NOT NULL,
PRIMARY KEY (`cid`)
);
```

Employers

- Employer ID
- Role
- Email
- Password
- Contact_name
- Company_name
- Phone
- Websie_url
- Address
- City
- State
- Zip

For deciding what information to be stored for the employer, we decided on having the minimum information necessary for fulfilling an employer form. The email address and password were included as a way to contact the employer and identify them when they log in. We also need employer id for primary key to connect to other tables. We add contact_name, company_name, phone, website_url, address, city, state and zip for the more information about the employer.

```
-- Table structure for table `employers`
CREATE TABLE `employers` (
 `eid` int(11) NOT NULL AUTO_INCREMENT,
 `role` varchar(50) DEFAULT 'employer',
 `email` varchar(100) NOT NULL,
 `password` varchar(50) NOT NULL,
 `contact_name` varchar(100) NOT NULL,
 `company_name` varchar(100) NOT NULL,
```

```
 `phone` varchar(20) NOT NULL,
 `website_url` varchar(100) DEFAULT NULL,
 `address` varchar(200) NOT NULL,
 `city` varchar(100) NOT NULL,
 `state` varchar(20) NOT NULL,
 `zip` varchar(50) NOT NULL,
  PRIMARY KEY (`eid`)
);
```

Types

- Type ID
- Type Name
- Color

For the types information we add type id, type name and color. The type id was added to be the primary key to allow for searching.

```
-- Table structure for table `types`
CREATE TABLE `types` (
`tid` int(11) NOT NULL AUTO_INCREMENT,
 `t_name` varchar(100) NOT NULL,
 `color` varchar(100) NOT NULL,
  PRIMARY KEY (`tid`)
) ;
```

Job

- Job ID
- Employer ID
- Categories ID
- Type ID
- Company Name
- Title
- Description
- City
- State
- Zip
- Contact Email
- Post Date
- Expired Date
- Salary

For job table we included the job id so that the job is associated with a specific job. The employer id, categories id and type id are also added in case we use them as

foreign keys to connect with other tables. The job should also contain the company name, title, description, city state, zip, contact email and salary for the more details about the job. The job also has a status to let the others know it's posted date and expired date.

```
-- Table structure for table `job`
CREATE TABLE `job` (
 `jid` int(11) NOT NULL AUTO_INCREMENT,
 `eid` int(11) NOT NULL,
 `cid` int(11) NOT NULL,
 `type_id` int(11) NOT NULL,
 `company_name` varchar(100) NOT NULL,
 `title` varchar(100) NOT NULL,
 `description` text NOT NULL,
 `city` varchar(100) NOT NULL,
 `state` varchar(100) NOT NULL,
 `zip` varchar(20) NOT NULL,
 `contact_email` varchar(100) NOT NULL,
 `post_date` date DEFAULT NULL,
 `expired_date` date DEFAULT NULL,
 `salary` int(11) DEFAULT NULL,
 PRIMARY KEY (`jid`),
 FOREIGN KEY (eid) REFERENCES employers (eid),
FOREIGN KEY (cid) REFERENCES categories (cid),
FOREIGN KEY (type_id) REFERENCES types (tid)
);
```

Seeker

- Seeker ID
- Role
- First Name
- Last Name
- Email
- Password
- Confirm Password
- Phone
- Address
- City
- State
- Created

The seeker table includes the seeker id of the job seeker as the primary key. The seeker table should contain the role, first name, last name, phone, address, city,

state and created time for the details of the seeker. We also need password and confirm password to make sure the security of our job seekers.

```
-- Table structure for table `seeker`
CREATE TABLE `seeker` (
 `sid` int(11) NOT NULL AUTO_INCREMENT,
  `role` varchar(50) DEFAULT 'jobseeker',
  `fName` varchar(50) NOT NULL,
  `lName` varchar(50) NOT NULL,
  `email` varchar(50) NOT NULL,
  `password` varchar(50) DEFAULT NULL,
  `confpass` varchar(50) DEFAULT NULL,
  `phone` varchar(50) NOT NULL,
  `address` varchar(200) DEFAULT NULL,
  `city` varchar(50) DEFAULT NULL,
  `state` varchar(10) NOT NULL,
  `created` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`sid`)
) ;
```

Profile

- Profile ID
- Seeker ID
- Desired salary
- Skill
- Projects
- Hobby
- Language
- Publication
- Education1-3
- Experience1-3
- Honors
- File
- Type

The profile entity contains the profile id of the profile as the primary key. The seeker id is used as foreign key to connect other tables. In the profile we also need desired salary, skill, projects hobby, language, publication, education, experience, honors, type and file of the seeker.

```
-- Table structure for table `profile`
CREATE TABLE `profile` (
 `pid` int(11) NOT NULL AUTO_INCREMENT,
  `sid` int(11) NOT NULL,
  `desired_salary` varchar(256) DEFAULT NULL,
```

```
 `skill` text,
 `projects` text,
 `hobby` varchar(500) NOT NULL,
 `language` varchar(256) NOT NULL,
 `publication` text,
 `education1` text,
 `education2` text NOT NULL,
 `education3` text NOT NULL,
 `experience1` text,
 `experience2` text NOT NULL,
 `experience3` text NOT NULL,
 `Honors` varchar(500) NOT NULL,
 `file` varchar(100) NOT NULL,
 `type` varchar(100) NOT NULL,
 PRIMARY KEY (`pid`),
FOREIGN KEY (sid) REFERENCES seeker (sid)
) ;
```

Future Work

This website includes many pages. The Integration and management of the codes are not easy. Adopting MVC model to rewrite the project is one of the plans for the future work to improve our project. We were intended to use the popular PHP framework Laravel 5 as the framework for our project. Because of the learning curve for all of our team members and the short of tutorials for the Laravel 5, it is difficult to master this framework in such a short time. We plan this application would be eventually rewritten with Laravel5.

We are also working on to support more functions, such as supporting PDF, word file upload for the job seeker's profile. This function was added and is supported after we have done the group presentation.

The other function we would like to work on is to add the subscription function. So job seekers can subscribe particular jobs they are interested in.

Twitter-like following function is another function we want to add to our project.

Conclusion

It is a great practice for us to apply the skills we have learnt in the classroom to a real time project. It is also wonderful journey that we experienced our ideals become reality via web technologies. During this journey, we deeply felt that a blueprint can be complex during its implementation. During this process, we have to reflect frequently what is the main focus of the project, what are the top priorities of the project. The reflection helped us concentrate on our main focus while explore other possibilities.

This is also a process for us to explore the variety of web technologies. Working on this project helps us recognize our potentials and limitations in terms of the web development. The Acknowledgement of our limitations and the appreciation of our potentials motivate us continue exploring the state-of-art technologies.

Working as a team is a valuable experience for us. Teamwork teaches us take responsibility to complete the task and trust each other.

Reference

"20 Best Job Board Themes and Plugins for WordPress To Create Awesome Job Sites 2015" online available at **https://colorlib.com/wp/best-wordpress-job-board-themes/**

"My Favorite Recipes Web Site Development"
http://cs.sju.edu/~bforoura/courses/f15/620projects/Sample_Final_Report.pdf

"HTML KickStart" http://www.99lime.com/
"Font Awesome, the Iconic Font and CSS Toolkit"
https://fortawesome.github.io/Font-Awesome/