Resume Application

The goal of this application is for you to explore/demonstrate understanding of web development using some of our core technologies. You must use Angular.js on the client-side, a Bootstrap template for the UI, Node.js/Express for the server-side, Bookshelf.js as the ORM, PostgreSQL as the database, and Bcrypt to encrypt passwords in PostgreSQL. You must also create a git repository on git.uc.edu and push all of your code there frequently.

Technologies

- git and Github
- Node.js
- Express
- PostgreSQL
- Bookshelf.js
- Angular.js (1.x) (NOTE: Ask Yahya or Dr. Said which version of Angular to use)
- Angular (2.x+) (use most recent stable version)
- Bcrypt

Architecture

The application must be architected using an API model. What is meant by this is that once a HTML page is rendered, all future communication between the client/server should be done using HTTP requests sending/receieving JSON.

Roles

- Applicant
- Manager

Features

- Allow applicants to submit form with the following information
  - First Name (text)
  - Last Name (text)
  - Email (text)
  - Position for which they are applying
    - NOTE: This should be a dropdown of available positions from the database
  - Resume (file)

- Allow Managers to log into the system
  - This must authenticate against the database
- Allow Managers to sort/search/filter/review submitted applications
  - Recommend using Angular UI Grid
- Allow Managers to delete application submissions
  - This should be a soft delete in the database (a soft delete does not remove the data from the database, it sets a flag on the record so it does not show up on the front end)

Resources

- AngularJS Tutorial (1.x)
- Angular Tutorial (2.x+)
- Bootstrap Templates
- AngularJS Routing (1.x)
- Angular Routing (2.x+)
- Express Application Generator
- Express API Reference
- Bookshelf.js: A Node.js ORM
- Bookshelf.js Example
- AngularJS HTTP Requests (1.x)
- Angular HTTP Requests (2.x+)
- Express File Uploads with Multer
- Angular UI Grid Documentation
- Use ExpressJS to Deliver Files as Attachment
- Hashing Passwords with Node.js and Bcrypt
- User Authentication using Bcrypt
- AngularJS Access Control by Route