COMP 4513 Assignment #2: Mean Stack

Version 1.6
Due Friday April 15

Overview

This multi-part group assignment will allow you to experiment with the MEAN web development stack. Like the first assignment, I will likely add or clarify some content over the next week or so. Groups can contain three or four people. Please don't ask to be a group of five, eight, eleven, or any other number larger than four.

Submitting

You will not be submitting your source code. Instead, I will mark your code from a GitHub repository. I will mark the functionality by visiting some type of live server. Thus, you will submit your assignment by emailing me a short message consisting of the group member names, the GitHub repository URL, and the web address where I will be able to find your working assignment.

Grading

The grade for this assignment (both parts) will be broken down as follows:

Visual Design and Styling	15%
Programming Design	20%
Functionality (follows requirements)	65%

Data Files

I have provided you with a JSON data file (employees.json and books.json) that you will import into Mongodb. The employees json file contains 100 employee objects. The structure of each employee is shown on the next page.

```
{
  "employee":{
    "id":2,
    "guid":"c29cb8f4-d8d5-40e9-8bc8-b054daa2e863",
    "firstname":"Elizabeth",
    "lastname":"Morgan",
    "username":"emorgan1",
    "password":"lLMVck6NjIlu",
    "salt":"velit",
    "todo":[
      {
        "id":1,
        "status":"completed",
        "priority":"low",
        "date":"3/23/2015",
        "description":"Maecenas leo odio, condimentum id, luctus nec"
      },
      ...
    ],
    "messages":[
      {
        "id":1,
        "contact":{
          "firstname":"Anthony",
          "lastname":"Ford",
          "university":{
            "id":227757,
            "name":"Rice University",
            "address":"6100 S Main",
            "city":"Houston",
            "state":"Texas",
            "zip":"77005-1827",
            "website":"www.rice.edu",
            "latitude":29.716485,
            "longitude":-95.403625
          }
        },
        "date":"8/25/2015",
        "category":"Query",
        "content":"Nam dui."
      },
      ...
    ],
    "books":[
      {
        "id":567,
        "isbn10":"0321906365",
        "isbn13":9780321906366,
        "title":"Writing and Reading Across the Curriculum, Brief Edition",
        "category":"English"
      },
      ...
    ]
  }
},
...
```

Overview

You will be creating what is commonly called a Single-Page Application (SPA) that will use Angular to construct the JavaScript front-end. The data will come from node.js REST web services you will write. The hypothetical user is an employee of a textbook publisher. The user will be able to log on and off, view the books she supports/sells, view and change her To Do list, and view existing Messages.

Requirements

1. Last assignment you made use of a "lite" implementation of Google Material. In this assignment, you can use any front-end CSS library (or indeed no library). You must experiment with something new and will receive a proportional reward.

   You might try Angular Material (https://material.angularjs.org), which is the reference implementation of Google's Material Design Specification. Maybe you want to use Foundation, which is the main competitor to Bootstrap.

   Or perhaps you can try using no CSS framework, and instead experiment with something in the CSS3 world. Maybe use a CSS preprocessor such as Less or SASS (I would recommend this for anyone keen to buff their web dev resume). Maybe try using the new FlexBox layout model in CSS3.

   Be sure to add an About page to your application where you tell me the nature of your experimentation (plus the usual info such as GIT url, project members, references to any 3rd party resources).

2. To begin, you are going to want to import the JSON data into a mongo database.

3. You are then going to need create node.js and Express REST web services for retrieving data from the mongo db. Your service will need retrieve functionality for the employee information, CRUD functionality for the employee's To Do data, retrieve functionality for the employee's Book information, and retrieve functionality for the employee's Messages.

   I would recommend either writing some type of JavaScript testers for these services to verify that they work. Alternately, you could use something like Postman to test your services.

4. The rest of the assignment will make use of Angular. Your assignment will be a single page application, and you are free to construct the user interface however you'd like. The remaining items describe in more detail this SPA.

5. It will being with a login form. Display appropriate messages if user is unsuccessful.

6. If the user id and password are correct, then some type of dashboard page will be displayed. This page will display the To Do list for the user, the Message List for the user, and their Book List.

   After the user logins in, then the user shouldn't have to login in again for some time (maybe 20 min). This will require making use of some type of session cookie. If we had more time, I would have you implement proper password hashing and some type of authentication ticket system. But in this assignment you just need to add in some type of (insecure) cookie-based authentication system and be aware that from a security standpoint it is very limited.

   To make it easier to test this out, also add a log out functionality / link

7. For the To Do list, style the status and priority so that they are similar to Bootstrap Badges or Angular Material Chips. Provide some clear way to create new entries, and update and delete existing entries. The To Do list should be sorted by date.

8. For the Message List, display all the contact name, university name, date, category, and message. The university name should be a link that opens a dialog which displays the rest of the details of the university as well as a map. The messages should be sorted by date.

9. For the Book List, display the isbn10, title, and category. Provide a mechanism for the user to change the sort order. You should be able to find a third party mechanism to help you with this.