Assignment 4

CURD App

Due Date: November 6, 2015 11.59PM

In this assignment, you will develop a simple profile system for users using NodeJS, MongoDB and Express, where users should be able to sign up and update their profiles. You may use Bootstrap or similar framework to help you style your application.

Overview

Implement a simple web application with both server side and client side components that:

- Allows users to signup and create a profile
- Allows users to login/log out
- Allows users to update a profile picture and changes details on their profile (after log in)
- Allows users to view a list of all users and their profiles
- Allows administrators to delete and edit user profiles
- Allows administrators to to track and view some users behaviours, such as which page the users are viewing most, users IP address, Users viewing device(Desktop/mobile, OS, Screen size etc.), user locations, and so on. (This list is minimal, you are encouraged to add more).

This is an individual assignment, each student must complete and submit their own solution.

Specification

User Accounts

There will be three different types of user accounts with different permission levels:

- Regular users
  - Can update their own account information
  - Can view a list of all users available in the system and view their profiles
- Administrative users
  - Has all the functionalities of a regular user
  - Can update and delete all regular users available in the system
  - Can view user behaviour.
- Super Administrative users
  - The first user in the database is automatically assigned Super Administrator privileges
  - Has all the functionalities of an administrative user
  - Can assign/unassign administrative privileges to another user

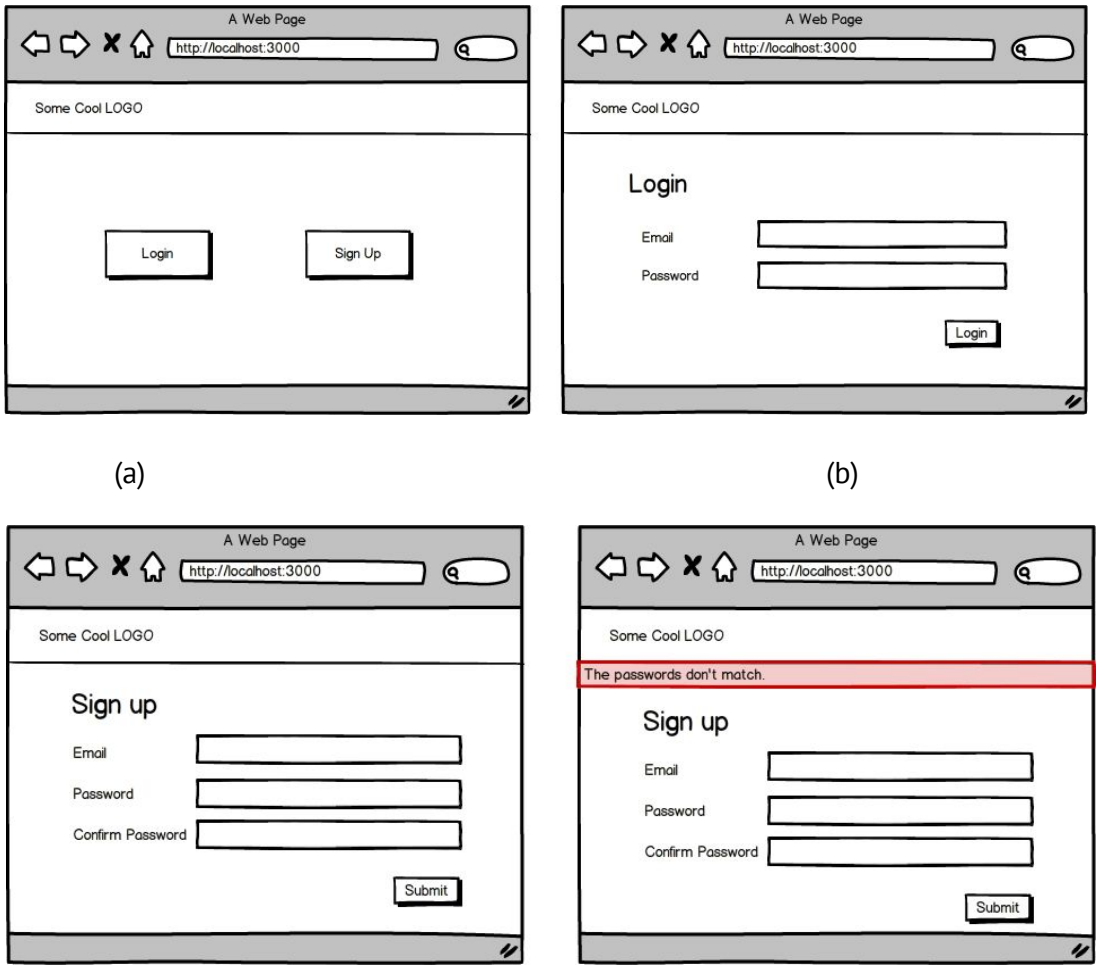
User profiles will include a minimum of the following fields (you can added more if you want)

- email address (this is the uniquely identified field)
- password (hidden to other users)
- description (less than 500 characters)
- profile image (cannot be empty, assign some default photo. you can also use gravatar)
- display name (does not have to be unique)

Views

Your web application should follow the following user flow, however, the content and "look and feel" of the pages is up to you.

The home page should include a Login and SignUp button. Clicking the Login button will check the email and password and return an error message if the credential don't match.

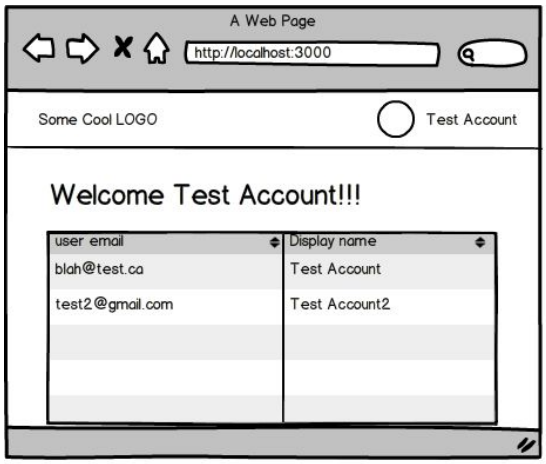


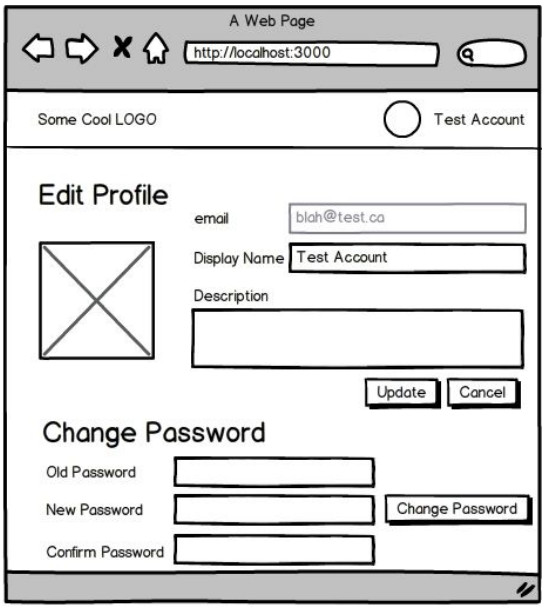
(a) (b)

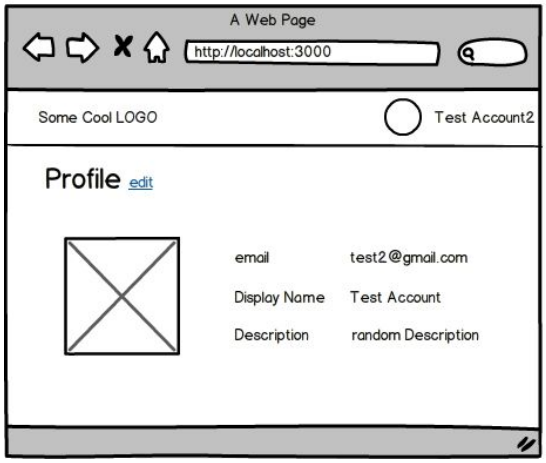
(c) Figure: 1 (d)

The signup page will look something like Figure 1 (c), with the application doing form validation on both the front end or back end to ensure that the fields are correct.

Once signup or login is complete, a normal user gets redirected to the following page (get creative with your design!):



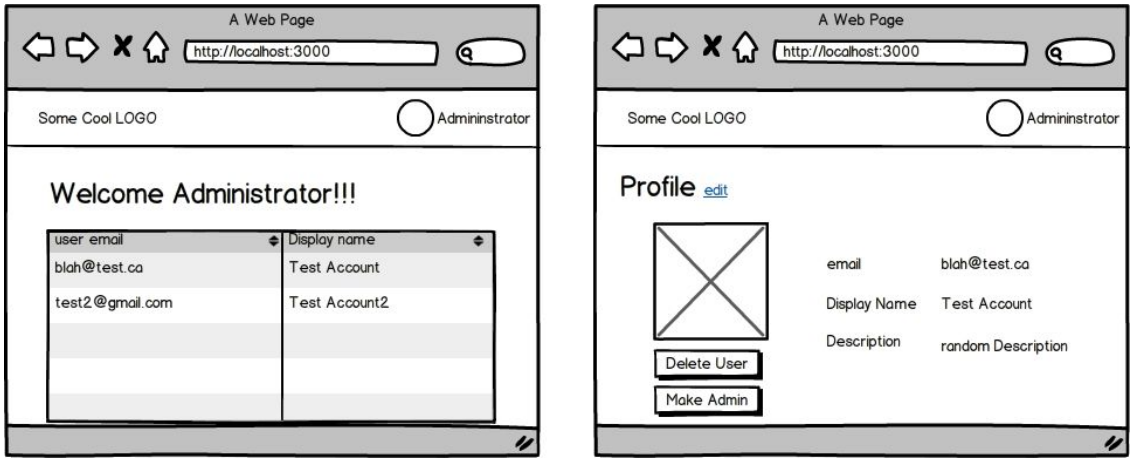




- The display name field may be empty -- in which case you will display the email address in lieu of their display name
- clicking on their profile image or their username will bring the user to the page where they can edit their display name and description. Note that although the email address is displayed -- they will not be able to change it.
    - again, form validation should be done so that the users know of mistakes they made:
        - incorrect current password
        - mismatched new passwords
- clicking the “logo” in the upper left hand corner of the page, users should taken back to the welcome page (if they are logged in) or the home page (if they are not logged in).
- logged in users should be able to click one of the rows in the list of users to view their profile in more detail.

An administrative account would sees a slightly different view.

- In addition to seeing their user profiles, there will be several options available to them to edit the user's profile, delete users or make them an administrator (only super administrator)
    - you should also be able to revoke someone's administrator abilities (only super administrator)
    - once they click the edit button, they will be taken back to a similar page as when the user edits their profile



Requirements

1. You should develop your application using HTML5, CSS3, Node, Express, and MongoDB.
2. You should be able to access the application on localhost:3000
3. You can use any third-party libraries or images (Use Creative Commons and attribute).
4. Your code should work on Google Chrome 43+.

Deliverables

Your assignment should be submitted as a zip file on MarkUs:

- `a4.zip`: All project files -- do not include the node modules folder in your zip file
    - TA will run:

```
sudo npm install
sudo node server.js
```

Rubric

- Functionality stated above - 70%
- Uses appropriate REST api structure - 10%
- Code formatting, commenting, and cleanliness - 10%
- Creativity and Visual Appeal - 10% (do as you like, no instruction will be provided)