**FINAL ASSESSMENT EXERCISE**

**OVERVIEW**

Your repository: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

You will clone a repository that contains a starter project (all the files you need for this part of the assessment). Every file has been created and linked appropriately in the index.html. **You** must create the module, the two components, the routing, and write the HTML for the login.html and profile.html partials. You are also responsible for styling the application to match the images below, following the CSS requirements listed in the “VISUAL” section..

The service for this application is **already** completed. **The only thing you need to do in the login-service.js is include your module and factory method to create LoginService.** The only thing you need from login-service.js is the name of the service and the two methods it includes.

When completed, git add/git commit/git push

REQUIREMENTS

**FUNCTIONAL *[10 pts]***

* The ngApp and ngView directives are included in the index.html correctly. [1 pt]
* Upon opening the application, the ‘login’ route is displayed. [1 pt]
* In ‘login.html’, add ngClick or ngSubmit(whatever you are comfortable with) to execute a function in the loginComponent. [1 pt]
* The two inputs should have ngModel values that are different properties of the same object. One property is ‘username’ and the other is ‘password’. [1 pt]
* Both components have the service injected properly. [1 pt]
* Upon submitting the form, the loginComponent component should call the appropriate method from the service that will check the login credentials. [1 pt]
* The functionality in the service is already set-up to redirect the user to the profile page upon a successful login. Set up a ‘profile’ route, which will load the appropriate component and partial. [1 pt]
* In the profileComponent, call the method from the service that is designed to pass information back to a component. [1 pt]
* Declare a property on the vm variable named ‘userDetails’, initialize that property to the data retrieved from the service. [1 pt]
* Display the user’s email, name, location, and phone number from vm.userDetails in the ‘profile’ partial. [1 pt]

**VISUAL *[10 pts]***

* The body’s font needs to be ‘Open Sans’. [1 pt]
* Use the following colors for appropriate elements: ◼ #63CEC2, ◼ #F7F8F9,  
  ◼ rgba(67, 75, 89, 0.7), and ◼ #CCC. [1 pt]
* Use border-radius on the following elements: form container, button, inputs, profile container, and profile image. [1 pt]
* Buttons and anchors should have hover states to change the text color to #63CEC2. [1 pt]
* Image and profile information must be floated to the appropriate sides. [1 pt]
* The Edit Profile link must be at the bottom and centered inside the profile container. [1 pt]
* Form and profile container must be centered on page. [1 pt]
* Use margin on form elements to add spacing between them. [1 pt]
* Profile information should be displayed as listed elements without bullets. [1 pt]
* Transform the headers of the form and profile container to uppercase (using CSS). [1 pt]



