

# FINAL ASSESSMENT EXERCISE

## OVERVIEW

You will be given a .zip folder to download that contains a starter project (all the files you need for this part of the assessment). Inside the .zip folder, every file has been created and linked appropriately in the `index.html`. **You** must create the module, the controllers, 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 the module.** The only thing you need from `login-service.js` is the name of the service and the two methods it includes.

When completed, please send an email to [frontend@grandcircus.co](mailto:frontend@grandcircus.co) with a link to the GitHub repository that contains this project.

## REQUIREMENTS

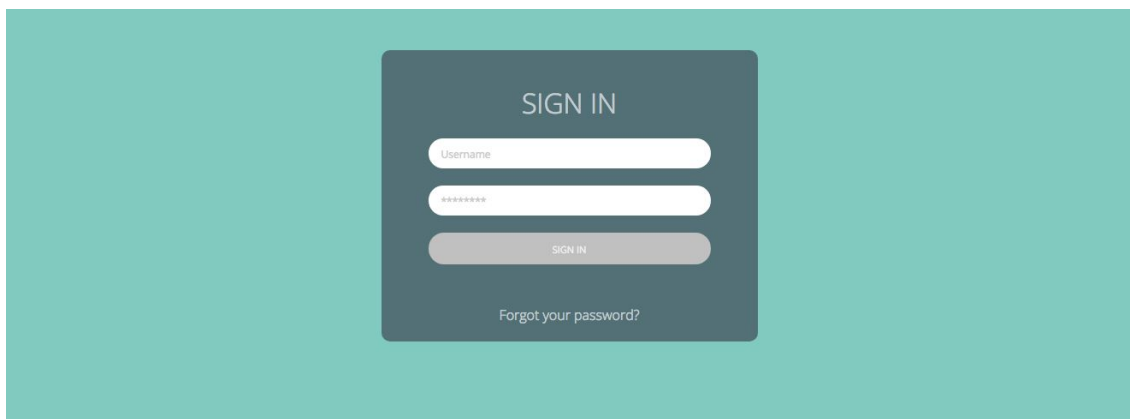
### FUNCTIONAL [10 pts]

- The `ng-app` and `ng-view` 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 `ng-click` or `ng-submit` to execute a function in the `login-controller.js`. [1 pt]
- The two inputs should have `ng-model` values that are different properties of the same object. One property is 'username' and the other is 'password'. [1 pt]
- Both controllers have the service injected properly. [1 pt]
- Upon submitting the form, your controller 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 controller and partial. [1 pt]
- In the profile controller, call the method from the service that is designed to pass information back to a controller. [1 pt]
- Declare a property on the scope 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 `$scope.userDetails` in the 'profile' partial. [1 pt]

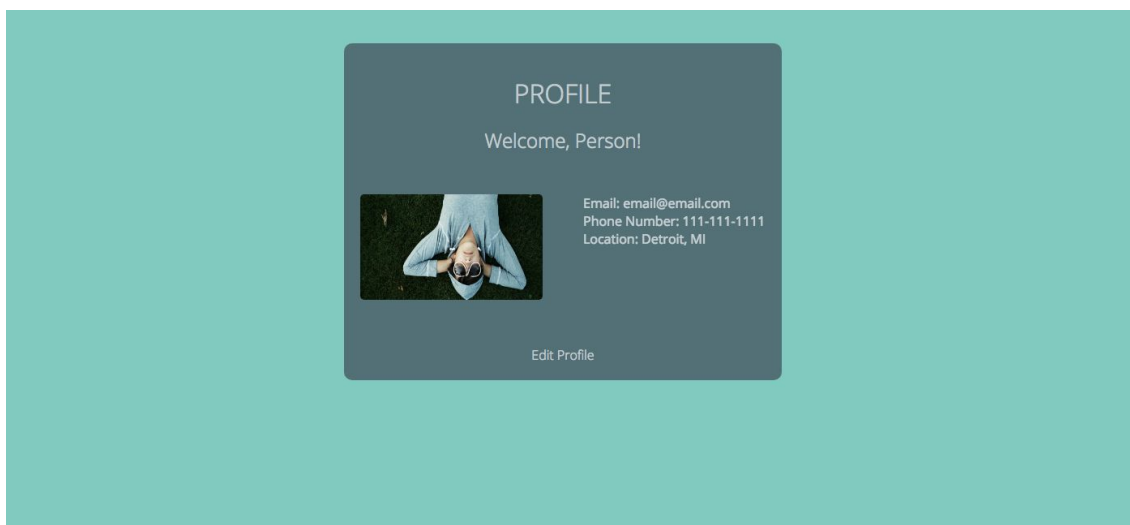


## VISUAL [10 pts]

- The body's font needs to be 'Open Sans'. [1 pt]
- Use the following colors for appropriate elements: #63CEC2, rgba(67, 75, 89, 0.7), #F7F8F9, 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]



A mockup of a 'SIGN IN' form. The form is a dark teal rounded rectangle centered on a light teal background. It contains the title 'SIGN IN' at the top, followed by two white input fields with rounded ends. The first field is labeled 'Username' and the second is masked with '\*\*\*\*\*'. Below the inputs is a light gray 'SIGN IN' button with rounded corners. At the bottom of the form is a link that says 'Forgot your password?'.



A mockup of a 'PROFILE' page. The page is a dark teal rounded rectangle centered on a light teal background. It features the title 'PROFILE' and a greeting 'Welcome, Person!'. On the left is a square profile picture of a person. To the right of the image, the following text is displayed: 'Email: email@email.com', 'Phone Number: 111-111-1111', and 'Location: Detroit, MI'. At the bottom center is a link that says 'Edit Profile'.

