CSCI 406 – Advanced Web Development
Lab 6 – Secure API and User Authentication

For this lab you will start with your Blogger application from Lab 5 as the basis and modify it to include new REST API endpoints for user registration and login, add updates several of the REST API endpoints to make them secure, as well as include updates to the AngularJS client application to allow for user registration, login and dynamic navigation based on login state. Lastly, you will update the application for and secured blog addition, updating and deletion functionality. When tested and working properly you will publish this version of your application to your previously created GitHub repository as a new branch named: "Lab 6".

PUBLIC SERVICE ANNOUNCEMENT: Like the last lab, this lab will take some time. Expect to spend at least 2 to 4 hours updating the REST API to add the register and login endpoints and then make it secured through JWT (JSON Web Tokens). In addition, expect it to take another 6 to 10 hours to update the AngularJS front-end in order to add the registration and login pages, the dynamic navigation and add all the security updates that will pass secure tokens through to the REST endpoints. You should base your updates on chapter 11 of the Manning book.

DUE DATE: *This lab is due to the instructor by Monday, November 2nd at 11:59pm.*

Instructions:

1. Using your Amazon Lightsail MEAN instance, make a copy of your previous Lab 5 application either using Linux copy commands or GIT commands to pull down the previous application to a new (different than previous) directory.

2. Your application will be similar to the application you wrote in lab 5, it will use the REST API and the AngularJS front-end from lab 5; both of which will be updated.

3. Your web application must adhere to the following specification:

Specification:

i. **Port 80:** Your application must load from a browser via port 80, which is the standard HTTP port for a server.

ii. **Add *register* and *login* endpoints to the REST API:** In order to secure the REST API endpoints, you must first add registration and login endpoints to the REST API that are used to add a user to the MongoDB database as well as login and securely receive a JSON Web Token used for signing calls to secured endpoints.

iii. **Secure the REST API endpoints using JSON Web Tokens:** Your program should be updated so that the following REST API endpoints described are secured so that they cannot be executed and instead will return a "401 Unauthorized" response if proper security tokens are not included when called:

- i. Blog add endpoint (e.g. POST of /api/blogs)
- ii. Blog update endpoint (e.g. PUT of /api/blogs/:id)
- iii. Blog deletion endpoint (e.g. DELETE of /api/blogs/:id)

The work required to add the *register* and *login* endpoints and then to secure the endpoints is described in the Lecture 9 class notes as well as in Chapter 11 of the Manning book.

Use Postman to test the register and login endpoints and at least make sure that the previous endpoints you have secured do not work and return the "401 Unauthorized" response when executed without a valid security token before continuing to the next steps.

iv. **Add Registration and Login Pages to the Front-End**: Your front-end must include registration and login pages like the following:
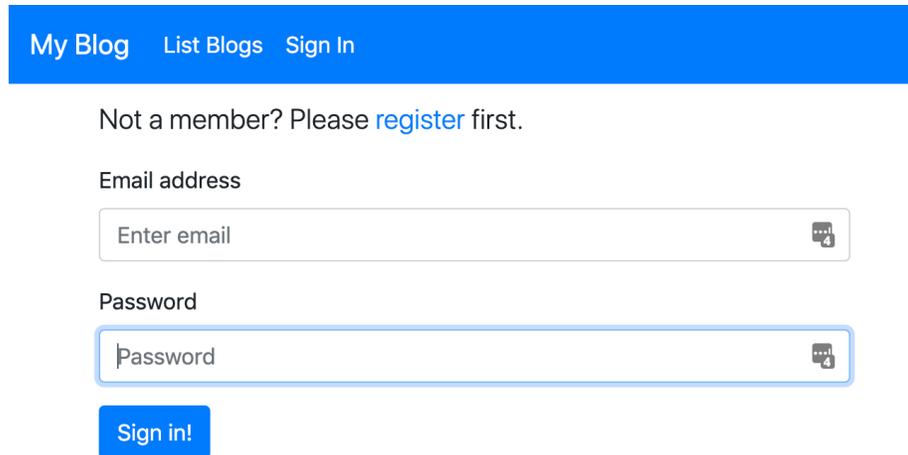

Figure 1: Sample registration page

Important: When the user attempts to register an email address that is already registered, the page should present an informative error message indicating they should try again with a different email address.

Figure 2: *Sample login page*

Important: If the user tries to login with an email address that is not registered or with an incorrect password, the page should present an informative error message for each of those cases.

v. **Update the Navigation Menu to be Dynamic:** The navigation menu should be updated to be dynamic as described by the feature specification below:

i. **No user is logged in:** If no user is logged in then remove the "Add Blog" menu item and add a menu item for "Sign In".

ii. **A user is logged in:** When a user is logged in include the "Add Blog" menu item and change the menu item for "Sign In" to instead allow for logging off and have a text label of: "Logout (<email_address>)", like shown below:

Figure 3: *Sample Logout menu item*

Important: It is highly recommended that you use an *Angular directive* for the dynamic navigation menu. When the user clicks the "Logout" menu item they should be logged out and the menu item should change back to "Sign In" *without*

having to refresh the application page (meaning, without hitting reload in the browser, for example).

    vi. **Update the Listing Page:** Update the listing page so that the *edit* and *delete* links for each of the blogs are only listed when someone is logged in (and not included if no one is logged in).

4. Your application must be setup to run even with your MEAN instance is not connected.

5. Using your GitHub account and the repository created and used prior and save this lab as a new branch ("Lab 6").

6. Once you have the application working and it is available via port 80 even when disconnected from your MEAN instance and you have your app uploaded to your GitHub repo, please send the full URL of your app AND your GitHub repo to the instructor via email with subject of "Lab6". Important:

    mailto:thomas.rogers@millersville.edu?subject=Lab6