Lab Assignment #2 – Developing an Express application with data access capabilities

Due Date: See e-centennial.

Purpose: The purpose of this homework is to:

- Become familiar with **MongoDB and NoSQL databases**
- Become familiar with **Mongoose**
- Develop a **Node** Web Application using **Express** and **MongoDB**

References: Read the textbook, lecture slides, and class examples. This material provides the necessary information that you need to complete the exercises.

Be sure to read the following general instructions carefully:

- This assignment must be completed individually by all the students.
- You **MUST name** your Visual Studio 2017 project as Yourfullname_COMP308Lab2.
- You **MUST** demonstrate your solution in a scheduled lab session, and submit the project using the assignment link on Dropbox.

Exercise 1

This exercise **extends Lab Assignment 1 exercise by adding database access**.

Create an Express Web Application which allows the customers to provide feedback for the services offered by your company. Your MongoDB database should have a collection ***customers*** to store **customer information** and customer **feedback**. You will have to provide a **login page** (ejs page) to allow the customers to login, a **sign up page** (ejs page), a **feedback page** (ejs) to allow the customers to enter the feedback, a **view customer feedback** page (ejs), and the **Thank You page** (ejs).

The **sign up page** should allow the customer to sign up by providing *first name*, *last name*, *email*, *password*, *and two other fields* that will be different for each student. For example, you may create fields for *favorite subject*, *number of languages*, *major*, *favorite sport*, *favorite team*, *favorite actor*, *favorite food*, *strongest technical skill*, etc.

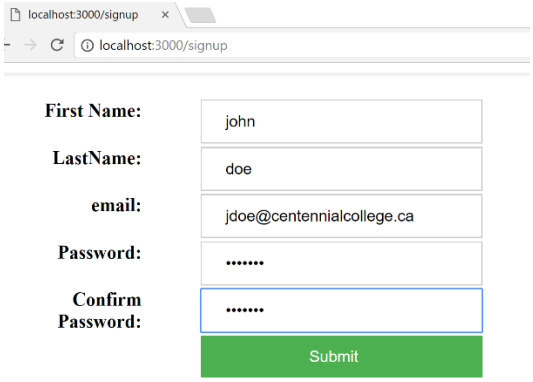

Figure 1. Sign Up page (ejs)

The login form should accept the user name (user's *email* address) and *password*.

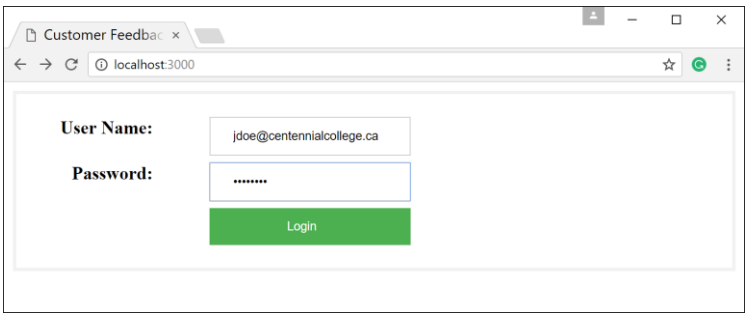

Figure 2. Login page (ejs)

Then it will use **MongoDB** and **Mongoose** methods to **find the user in the data store**. After finding the user, the application should display the feedback page and automatically **populate the** *first name, last name, email and two additional fields* as described above. Your application should **store customer feedback when the user clicks on submit button**. Then, a **Thank You form should display the customer name and comments** and thank the customer for providing the feedback.

The **view customer feedback** page should allow an admin person to view the feedback for each customer given the email address (user name).

Implement a **horizontal folder structure** for your application similar to Lab 1. Apply **MVC principles**. Design nice and friendly web pages.

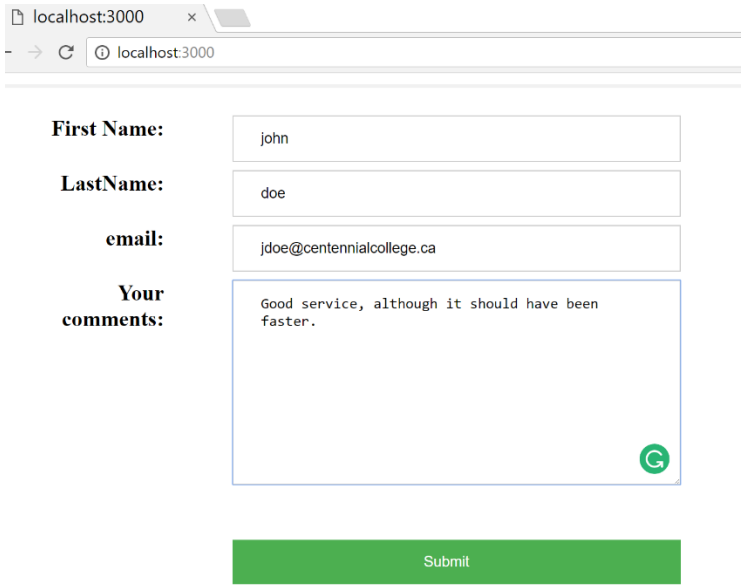

Figure 3. Feedback form

(10 marks)

Evaluation:

Functionality:	
UI views (index.ejs, signup.ejs, feedback.ejs, viewcustomerfeedback, thankyou.ejs)	10%
Correct routing code	15%
Correct controllers code	20%

Correct models code and MongoDB database	25%
Correct implementation of MVC architecture	10%
Correct server.js, express.js, mongoose.js and config.js files	10%
Friendliness	10%
Total	**100%**