**Assignment 2 – Advanced Web Programming**

This assignment requires you to create a new NodeJS application using ExpressJS, MongoDB / Mongoose, and EJS Templating. Your mark counts for 25% of your final grade.

**Submission Requirements via Blackboard:**

**Due Part 1: Wed, Nov 16 @ 11:59 pm**

* Express Generator site set up
* MLab Database set up with Node connection created
* Link to your *Private* GitHub repository with a *minimum* of 5 commits (4 marks)

**Due Part 2: Wed, Nov 30 @ 11:59 pm**

* Link to your site home page, which must be hosted online with a cloud provider such as Azure, Heroku, or Digital Ocean (4 marks)
* Link to your *Private* GitHub repository with regular commits (4 marks)

This is an independent assignment. All work must be your own. Sharing code with another project is strictly prohibited and will constitute Academic Misconduct. Failure to submit an independent assignment will result in a grade of zero.

For this Assignment you have 2 options. You can pick either:

# Option A: Business Directory. For sample application, visit:

<http://barriechamber.com/Directory/Internet>

You are to build a similar online directory. Choose any type of business you like (i.e. Restaurants / Theatres / Pubs / Dentists). Build a model in your application with 4-10 logical properties of the type of Business you choose.

**Option B: An Alternate CRUD Application of your choice.**

In this option, you will need to propose what you would like to build (Workout Tracker, Grocery List Manager, etc.). The application must still contain full Create-Read-Update-Delete functionality. You must have Instructor approval ***via email*** for Option B.

YOU MUST WRITE THE CODE YOURSELF, NOT JUST TAKE MY GITHUB CODE AND MODIFY IT.

In this assignment you will use NodeJS to:

* Create a new web application
* Implement the ExpressJS web framework
* Use MongoDB and Mongoose to store and connect to your data
* Use EJS templating for your UI
* Add route files to handle application logic
* Customize the look and feel of your application with CSS (Bootstrap is fine but the application must look different than our in-class application example)
* Implement Authentication so only your Home page, Register page, Login page, and Read-Only page that displays your data are public. All other pages that perform CRUD operations must be Private.

**Application Requirements:**

1. Create a new ExpressJS application
2. Implement a site design using your own CSS, or a framework like Bootstrap or Foundation. Give the application the look and feel of a professional online directory. It should not look identical to our in-class application.
3. Set up your database on [www.mlab.com](http://www.mlab.com) and make sure the database credentials are stored in a config file (NOT in app.js)
4. Build a home page that serves as a splash page
5. Build a shared header and footer
6. For **Option A** (Business Directory), build a public page that displays all the businesses in your directory in a Read-Only format (No add / edit / delete). Businesses must include at least 5 fields. For **Option B** (Student-Chosen Application) build a page that displays a Read-Only version of your data (No add / edit / delete).
7. Build a user registration page.
8. Build a login page. Allow users to also log in with either a Github or Facebook account (choose 1, you don’t have to include both).
9. Build private pages that allow authenticated users to view, add and edit businesses (Option A) or your own data (option B)
10. Enable Delete functionality, including a Delete Confirmation.
11. Implement at least 1 additional feature of your choosing that show some independent learning. The feature you choose should be listed in your README.md file on GitHub. Options for this include but are not limited to:
    1. Authentication with an additional provider besides passport-local, Facebook, or GitHub
    2. Create an API that returns a list of all businesses (Option A) or your own data (Option B) in JSON format
    3. Create an API that adds a new business (Option A) / your own document (Option B) to the database
    4. File Uploads that allow users to upload a logo for a business (Option A) or an Image (Option B)
    5. Add a Keyword Search to the public data view page
    6. Any other feature of your choice
12. Comment your code.

**Evaluation Method**

Your work will be evaluated based on how your application performs on the following items:

# Evaluation Criteria (max 48 marks)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **0-2** | **3-4** | **5-6** | **7-8** | **Marks** |
| **Node / ExpressJS Setup** | - Little or no application | - Partly complete app | - Mostly complete app | - Fully functional ExpressJS app hosted on live server | **4** |
| **Routing** | - No Routing | - Partly complete Routing | - Mostly complete & accurate Routing | - Full CRUD | **16** |
| Authentication | - Auth missing | - Some authentication complete | - Most Auth complete | - All pages except Home page, register, login and Read-only list secured | **8** |
| Student Chosen Bonus Feature 1 | - None | - Some attempt to implement feature | - Mostly implemented | - Fully implemented | **4** |
| **Home Page** | None | Attempted | Mostly complete | Complete | **1** |
| **Css** | - None | - Minimal | - Partial | - Complete | **2** |
| **Code Commenting** | - No comments | - Some attempt to explain code | - Most sections clearly explained | - All sections clearly explained | **1** |
| **Version Control (Private)** |  |  |  |  | **8** |
| **Cloud Deployment** |  |  |  |  | **4** |