**Assignment 2 – JavaScript Frameworks**

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

**Submission Requirements via Blackboard:**

**Part 1 Due: March 14th EOD**

* Completed Project Plan Template (download from platform)

**Part 2 Due: April 4th EOD**

* Link to your site home page, which must be hosted online with a cloud provider such as Azure, Heroku, or Digital Ocean
* Link to your *Private* GitHub repo

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. All submissions are subject to a code review at the instructor’s discretion where you may be asked to explain any of the code in your project.

External code (e.g., from the internet or other sources) can be used for student submissions within the following parameters:

1. The code source (i.e., where you got the code and who wrote it) must be cited in your internal documentation.
2. It encompasses a maximum of 10% of your code (any more will be considered cheating).
3. You must understand any code you use and include documentation (comments) around the code that explains its function.
4. You must get written approval from me via email.

**Create a secure CRUD Application.**

In this option, you will need to propose what you would like to build (Workout Tracker, Assignment Tracker, etc.). The application must still contain full Create-Read-Update-Delete functionality.

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 HBS templating for your UI
* Add route / controller 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). If you use a 3rd party stylesheet, reference the source in both your header and your README.md file. If you do your own CSS, note this in the README.md file.
* 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:**

**Part 1**

1. Create a Project Plan

**Part 2**

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.mongodb.com](http://www.mongodb.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. Build a public page that displays a list of all the documents in 1 of the collections in your database in a Read-Only format (No add / edit / delete).
7. Build a user registration page.
8. Build a login page. Allow users to also log in with a GitHub account*.*
9. Build private pages that allow authenticated users to view, add and edit data.
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 or GitHub
    2. File Uploads
    3. Add a Keyword Search to the public data view page
    4. 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

Part 1: 8 marks

Part 2: 38 marks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **0-2** | **3-4** | **5-6** | **7-8** | **Marks** |
| **Project Plan** | None | Partly Complete | Mostly complete | Complete | **8** |
| **Home Page** | None | Attempted | Mostly complete | Complete | **2** |
| **Routing / Controllers** | - No CRUD | - Partly complete CRUD | - Mostly complete & accurate CRUD | - Full CRUD | **16** |
| Authentication | - Auth missing | - Some authentication complete | - Most Auth complete | - All pages except Home page, register, login and Read-only list secured, 1 social auth working | **8** |
| Student Chosen Feature | - None | - Some attempt to implement feature | - Mostly implemented | - Fully implemented | **4** |
| **CSS** | - None | - Minimal | - Partial | - Complete | **2** |
| **Code Commenting** | - No comments | - Some attempt to explain code | - Most sections clearly explained | - All sections clearly explained | **2** |
| **Version Control** |  |  |  | Private repo, eduardojaime invited, README file, min 4 descriptive commits over min 2 days | **2** |
| **Cloud Deployment** |  |  |  |  | **2** |