Assignment 2 – JavaScript Frameworks
CRUD Application

Description

This assignment requires you to create a new Node application using Express, MongoDB/Mongoose, and HBS views. Your site must be hosted live on a cloud service such as Render (Recommended), Azure, Heroku, AWS, or Digital Ocean.

Assignment 2 counts for 25% of your final grade and is split into two parts with different submission requirements.

- **For 2A**, you will propose an idea for an application that implements CRUD functionality. Some examples include: Workout Tracker, Assignment Tracker, etc.
- **For 2B**, you will write the code for this application to implement CRUD, authentication, and one additional feature.

Deliverables

Assignment 2A

- Completed Project Plan Template (download from platform).

Assignment 2B

- Link to your live site, which must be hosted online with a cloud provider.
- Link to your GitHub repository.
  - Code must be committed/pushed using either GitHub Desktop or Git Bash (if you are familiar with this tool).
  - Commit/Push your code to the **ASSIGNMENT2** folder in your class repository.
  - Manual Uploads via GitHub Web UI will not be accepted.
  - Zip files will not be accepted.

Important Notes

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., ChatGPT, 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 README file.
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.

Application Requirements

Part 1

1. Answer the questions in the Project Plan document

Part 2

2. Project Template and Home Page
    a. Create a new Express application using the Express Generator tool.
    b. Make sure to select the HBS templating engine when creating the project template.
    c. Implement a site design using your own CSS, or a Bootstrap template.
    d. Give the application the look and feel of a professional online directory. It should not look identical to our in-class application.
    e. Build a home page that serves as a splash page.
    f. Build a shared header and footer.
3. Authentication
    a. 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).
    b. Build a user registration page.
    c. Build a login page. Allow users to also log in with a GitHub account.
4. CRUD
    a. Set up your database on www.mongodb.com and make sure the database credentials are stored in a config file (NOT in app.js)
    b. Build private pages that allow authenticated users to view, add and edit data.
    c. Enable Delete functionality, including a Delete Confirmation.
5. Additional Feature
    a. 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:
        i. Authentication with an additional provider besides passport-local or GitHub.
        ii. File Uploads.
        iii. Add a keyword search to the public data view page.
        iv. Another feature of your choice.
6. Commenting
    a. Add a brief description of your application in the README file.
    b. Include a link to your live site in the README file.
    c. Add a brief description of the Additional Feature you implemented in the README file.
7. Version Control
    a. Make a minimum of 4 commits with descriptive messages to your repository.
    b. Commit/Push your code to the ASSIGNMENT2 folder in your class repository.
8. Cloud Deployment
    a. Deploy your code to a Cloud Hosting environment.

Evaluation Criteria

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

2A: 8 marks
2B: 36 marks

Criteria					**Marks**
Project Plan	None	Partly Complete	Mostly complete	Complete	**8**
Project Template and Home Page	None	Attempted	Mostly complete	Complete	**2**
CRUD	No CRUD	Partly complete CRUD	Mostly complete & accurate CRUD	Full CRUD, DB correctly configured	**14**
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**
Additional Feature	None	Some attempt to implement feature	Mostly implemented	Fully implemented	**4**
CSS	None	Minimal	Partial	Complete	**2**
Code Commenting	No comments or readme file			Additional feature clearly explained, info provided in README file	**2**
Version Control				Min 4 descriptive commits	**2**
Cloud Deployment				Live link provided and working	**2**