Assignment 5

This assignment assesses your understanding of:

- Using a database with an Express web application
- Using a <form> to submit data to an endpoint
- Server side processing of <form> data

In this assignment, you will create a web application that models a travel journal. A sample user interface is provided in the course webpage. You may modify the colours, fonts, images, but the overall layout must be similar.

Submission Instructions

When you are ready to submit, follow these instructions to create your submission:

1. At the top of your server.js file, add the following declaration

```
/*********************************************************************************
*  WEB322 – Assignment 05
*
*  I declare that this assignment is my own work and completed based on my
*  current understanding of the course concepts.
*
*  The assignment was completed in accordance with:
*  a. The Seneca's Academic Integrity Policy
*  https://www.senecacollege.ca/about/policies/academic-integrity-policy.html
*
*  b. The academic integrity policies noted in the assessment description
*
*  I did NOT use generative AI tools (ChatGPT, Copilot, etc) to produce the code
*  for this assessment.
*
*  Name: _____________________ Student ID: _____________
*
********************************************************************************/
```

2. Rename your project folder: `A5_FIRSTNAME`. Replace `FIRSTNAME` with your preferred name. Example: `A5_TOMMY`

3. Create a zip file of the folder. Name the zip file: `A5_FIRSTNAME.zip`. Replace `FIRSTNAME` with your preferred name. Example: `A5_TOMMY.zip`

4. Submit your zip file to the dropbox by the specified due date.

Important Notes

- **NO LATE SUBMISSIONS** for assignments. Late assignment submissions will not be accepted and will receive a grade of zero (0).

- **Submitted assignments must run locally,** ie: start up errors causing the assignment/app to fail on startup will result in a grade of zero (0) for the assignment.

Academic Integrity

Please familiarize yourself with the college's Academic Integrity Policy

This is an individual assessment.

Permitted activities:

- Using course materials or other internet sources to clarify course concepts.
- Using the internet to lookup HTML, CSS, or Javascript syntax

Not permitted:

- Reposting any part of the assessment to online forums or homework help websites
- Contract plagiarism: Purchasing a solution, or completing a solution for compensation
- Sharing or receiving source code, references, or assistance from others

Usage of Artificial Intelligence (AI) tools:

- AI usage to generate solutions is NOT permitted. Examples: ChatGPT, Copilot,the AI tools built into VSCode.
- This assignment should be created based on your current knowledge of the course concepts, NOT based on what an AI knows.

Problem Description

Using the techniques described in class, implement a web application that represents a travel journal about **memorable locations** in a destination of your choice.

The app must provide these features:

- Add a memorable location
- Delete a memorable location
- Show all memorable locations

You may use the assignment starter code located in the assignment folder on the course webpage.

1. Choose a Travel Destination

Select a destination of your choice. Users will add memorable locations that are located in that destination.

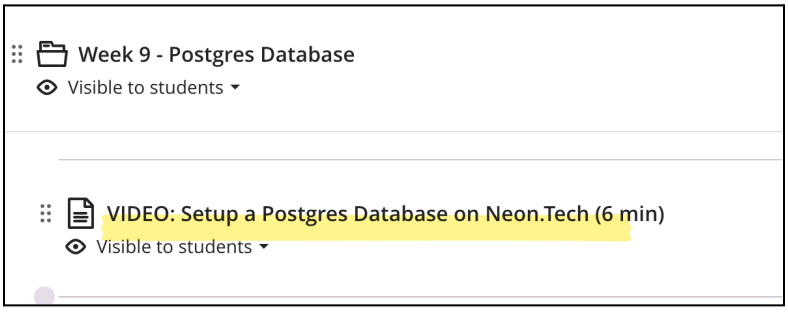
2. Database

The web application must save the user's location to a Postgres database.

a. Database setup and configuration

- App data must be stored in a Postgres database hosted on neon.tech.
- The application must connect to the Postgres database *before* starting the web server.
- All database configuration information must be stored in a .env file. Specifically: PGDATABASE, PGHOST, PGUSERNAME, PGPASSWORD

NOTE: If you are **unsure** how to do this, follow instructions on the course webpage



b. Database modelling

- The table for storing the user's memorable locations must be implemented using a Sequelize *model*, as follows:

Column Name	Sequelize DataType
name	Sequelize.TEXT
address	Sequelize.TEXT
category	Sequelize.TEXT
comments	Sequelize.TEXT
image	Sequelize.TEXT

- You do not have to explicitly specify a primary key column. As discussed in the lecture notes, Sequelize will automatically create a primary key column called "id".
- Ensure the Sequelize model ignores the created_at and updated_at fields.

3. Implement Server Routes (Endpoints)

Your web application must respond to the following routes (endpoints)

Method	Route Name	Description

GET	/	**Displays a home page containing:**
		• A `hero with overlay` section.
		  ○ Show **destination name**, **photo**, and a **brief description** about the destination.
		  ○ DaisyUI link: https://v4.daisyui.com/components/hero/#hero-with-overlay-image
		• A `link` to **Add** an location
		• A `table` showing all the locations in the database.
		  ○ For each location, show the image, name, category, address, comments, and a delete link.
		  ○ DaisyUI link:https://v4.daisyui.com/components/table

Routes for adding new location to the app		

GET	/memories/add	**Displays a page for the user to add a memorable location.**
		The page must display:
		• A `hero` section with the page name and instructions on what the user should do. Daisy UI link: https://v4.daisyui.com/components/hero/#centered-hero
		• A `<form>` for the user to add a location details.
		  ○ You are responsible for selecting the most appropriate form field elements.
		  ○ All form fields are required.
		  ○ For category: Provide 3 categories that the user can choose from.
		  ○ For the image: Assume the user will enter a *url* to the image.

POST	/memories/add	**Inserts a location into the database.** The endpoint should: 1. Receive the location details from a <form>, 2. Attempt to insert the location to the database 3. Redirect the user to the / endpoint You are not required to handle the scenario where the database operation fails.
Routes for deleting existing location		
GET	/memories/delete/:id	**Deletes the location with the specified id** and **redirects the user to the / endpoint** You are not required to handle the scenario where the database operation fails.

4. User Interface

A sample user interface is provided in the course webpage. Your user interface does not have to be exactly the same, but it must contain similar UI elements to what is shown in the sample.

- The app's user interface must be implemented using EJS templates.
- Pages should be reasonably pretty and styled with Tailwind/DaisyUI. You can use the HTML code from the sample user interface as a starting point for relevant Tailwind/DaisyUI CSS classes.
- A fully responsive design is **not** required.

5. Deployment

Deploy the application to Vercel. See the course webpage for instructions.

END OF ASSESSMENT