ASSIGNMENT-01 **2021**

INFO 16206
WEB & SCRIPTING LANGUAGES

Max Marks	:	100
Due Date	:	Available on SLATE
Type	:	Individual Work
Weight	:	10% of Total
Objectives	:	Demonstrating understanding of HTML Forms and client-side validation. Persisting JSON data using Node.js Using AJAX for partial page updates.

Requirements

In this exercise you are required to create a web application to maintain a collection of records of your choice. You could maintain information about sports, products, movies, superheroes, or any other entity you want to work with.

Use HTML, CSS and JavaScript to create an HTML form to accept the data for creating a record. Make sure to use a reasonable set of attributes to represent the entity whose record is to be maintained. Try to add a variety of relevant controls to capture user input. Make sure to enforce appropriate validation checks on the user input and notify user if the input does not satisfy the constraints. Validate the constraints before user can submit the form. [25 Points]

When the POST request is received on the server end, add the record to a JSON file. Create the file if it does not already exist. Old data contained in the file should not be lost. Ensure duplicate entries or invalid data is not added to the file. Notify the user about the status of operation. [25 Points]

Create a page to search for records based on a key value and a non-key value. The search page should allow the user to enter the search criteria and display the results on the same page. Perform a partial page reload whenever user searches for data based on either a key or a non-key field. [25 Points]

Create a page to document all challenges, research, and learnings. Provide consistent means to navigate between the pages. Make sure to maintain a git repository. **Submissions with missing documentation and/or git repo will be given a grade of Zero.** [25 Points]

Bonus:

10 Points: Allow editing the record returned by key-based search.

Notes:

1. The **quality of your code** is extremely important. You are required to place the referenced resources (e.g. images) in separate dedicated folder, and use appropriate indentation and comments. Up to 30% of the mark for a submission can be deducted due to poor quality: *incorrect folder structure (10%)*, *indentation* and *comments (10%), poor presentation (10%)*.

2. **All assignments shall be submitted by the deadline.** Late submissions will be penalized with 10% per day for up to 3 calendar days after which the assignment cannot be submitted anymore. **An email must be sent** should you choose to submit a late assignment. **Assignments are not accepted after the deadline.** See Academic Procedures for Evaluations **http://academic.fast.sheridanc.on.ca/r/ac_academic_procedures_for_evaluations.pdf**

3. This assignment shall be **completed individually**. Remember that completing the assignment by yourself will also ensure your success on the midterm and final exam. See https://www.sheridancollege.ca/student-life/student-services/library-services/academic-integrity

4. Submission is done through SLATE assignments UI as a single Zip file**. Make sure you DO NOT email your submission.**

	Level 3	Level 2	Level 1
HTML Form and Client-side Validation (25)	A relevant set of attributes was chosen to represent the state of the entity. The HTML form used a variety of input controls relevant to the nature of the attributes. Meaningful validation checks were identified an enforced on all attributes through JavaScript. The code had a clean implementation, with no global variables, modularized logic (divided into independent functions with appropriate parameters and return values)	Most of the attributes were used to represent the state of the entity were relevant any important attribute was not missing. Most of the input controls used in the HTML form were relevant to the nature of the attributes. Meaningful validation checks were identified an enforced on most of the attributes through JavaScript. The code used some global variables. Some logic was modularized.	Some relevant attributes were used to represent the state of the entity. Or most of the important attributes were missing. The chosen entity might be a copy of the class exercise. Some of the input controls used in the HTML form were relevant to the nature of the attributes. Or most of the input was done using text boxes only. Meaningful validation checks were identified an enforced on a few of the attributes through JavaScript. The code mostly used global variables. The logic was not modularized, or the modularization was not correct.
	25-19	18-12	11-0
Save Record (25)	The submission could successfully add the posted form data to a JSON file. The user was notified about the completed operation. Duplicate and invalid data was rejected, and the client was notified about the error. The code had a clean implementation, with no global variables, modularized logic (divided into independent functions with appropriate parameters and return values)	The code to add the posted form data to a JSON file had some errors. The user was notified about the completed operation. The code to detect and reject duplicate and/or invalid data might have some errors. Notification to the client might not be sent. The code used some global variables. Some logic was modularized.	The code to add the posted form data to a JSON file had many errors or was logically incorrect/incomplete. Notification to the user might not be sent. The code to detect and reject duplicate and/or invalid data might have many errors or might be incorrect/incomplete. Notification to the client might not be sent. The code mostly used global variables. The logic was not modularized, or the modularization was not correct.
	25-19	18-12	11-0
Search Record (25)	The submission could search records stored in the JSON file on basis of a key and a non-key attribute. The search results were used to update a part of the page (And did not require a full page reload).	The code to search records stored in the JSON file on basis of a key and/or a non-key attribute, had some errors. The search results used to update a part of the page might have some errors but did not require a full page reload.	The code to search records stored in the JSON file on basis of a key and a non-key attribute, had many errors or was logically incorrect/incomplete. The search results used to update a part of the page might have many errors or might be incorrect/incomplete or might require a full page reload.

	The search results were presented in a well formatted way on the client end. The code had a clean implementation, with no global variables, modularized logic (divided into independent functions with appropriate parameters and return values)	The formatting of the search results might need improvement. The code used some global variables. Some logic was modularized.	The search results were presented with inconsistent/little/no formatting. The code mostly used global variables. The logic was not modularized, or the modularization was not correct.
	25-19	18-12	11-0
Research, Learnings, Navigation (25)	All pages were navigable using a consistent navigation scheme. Submission shows that a good amount of research was done on the topic and technology. Technical challenges faced were documented and their resolution approach was clearly explained. All research (related to topics beyond class teachings) and learnings were clearly documented as required with references to specific resources (Reference links pointed to a specific page instead of the home page of the website e.g., https://developer.mozilla.org/ is not acceptable as a valid reference) All important milestones were committed using git.	Most pages were navigable. Navigation scheme might not be consistent. Submission shows that some research done was on the topic and/or technology. Documentation for technical challenges faced was either too general or their resolution approach was not explained clearly. Most research (related to topics beyond class teachings) and learnings were clearly documented as required with references to specific resources (Reference links pointed to a specific page instead of the home page of the website) Most milestones were committed using git.	Most pages were disconnected. There was no planned navigation approach. Submission shows that limited research done on the topic and/or technology. Documentation for technical challenges faced was either too general or missing. The resolution approach was not explained or was incorrect. Either no research done on topics beyond class teachings) or learnings were not documented. Most references were either missing or too general (e.g., pointing the home page of a website) The git repo was not submitted, or most milestones were missing from the git repository.
	25-19	18-12	11-0