SFWRTECH 4WP3 – Advanced Web Programming - Assignment #4

Weight

4% of total course grade

Due date

Sunday February 20th at 11:59pm

Primary learning objectives

- Use NodeJS to respond to requests.
- Use AJAX to send and request data from a server-side script

Requirements

Unzip the starter code on Avenue to Learn. The starter code will run by running:

```
npm install
node server.js
```

on the command-line.

The route:

http://localhost:3000/search

Will serve a placeholder search page.

The routes:

All: http://localhost:3000/all

Bathrooms: http://localhost:3000/bathroom_search?bathrooms=4

Bedrooms: http://localhost:3000/bed_search?bedrooms=4

Price: http://localhost:3000/price_search?min=850000&max=4000000

Community: http://localhost:3000/community_search?community=Stoney%20Creek

Will return JSON arrays of house data, using the query parameters contained in each URL.

We'll use this data to build the **Search** page for our **Hamilton Homes** website that functions using AJAX! Use your usual Bootsrap navigation bar for this page.

When the search page is requested with the URL http://localhost:3000/search and it first loads, at the top of the page there should be a form, and beneath it should be a table containing all of the homes. The data to display all of the homes should be fetched using AJAX and the **All** route above. You can use either jQuery or the fetch API to make the AJAX request. The form should include an "All" button that allows the user to populate the table with all of the housing data, again by using an AJAX request to the server.

The form should give the user a row of buttons that allow them to filter by community ("Mountain", "Ancaster", etc.).

The form should give the user a dropdown selection box with options in order to select a bedroom number: 1, 2, 3, 4, 5, 6, 7. Next to this dropdown should be a button that allow them to "Filter By Bedroom".

The form should give the user a dropdown selection box with options in order to select a bathroom number: 1, 2, 3, 4, 5, 6, 7. Next to this dropdown should be a button that allow them to "Filter By Bathroom".

The form should give the user two input textboxes with appropriate labels that allow the user to enter a min and max price. Next to these should be a button that allows them to "Filter By Price".

All of these filter functionalities should be handled by making an AJAX request to the assoicated route (e.g. localhost:3000/price_search in the case of filtering by price). The form input values provided should be used as the query parameters. The filtered data returned from the AJAX request should be used to re-construct the table such that it only contains the filtered elements.

Only one filter should be applied at a time, after each button click. In the case of the All button being clicked, none of the filters will be applied.

The filters cannot be cumulative. We'll introduce concepts like SQL that will help with "cumulative filtering", but for now we are not doing this. The filtering of the results MUST be done by the backend by making AJAX requests to the backend, no marks will be awarded for fetching all the data once and then performing filtering on the front-end.

The table and form should be styled with Bootstrap Version 5. Use jQuery to handle all events and perform all access and updates to the DOM, such pulling data from the form inputs and creating the updated of the table contents. None of the other “pages” for the Hamilton Homes website need to work, just the Search page.

Do not modify the server code, only write the front-end code.

Submission

Your solution should include your search.html page, the files that come with the starter code, and any other files like images etc, required for your solution to work.

Your solution will be tested by unzipping your code, running npm install in the resulting folder, running **node server.js** to start the server, and navigating to localhost:3000/search in a web browser. Your solution is required to work this way.

Zip your solution as assignment4.zip and upload it to the dropbox on Avenue.

Marking rubric

Component	**Description**	**Marks**
Form, Table styling	Form and table styled using Bootstrap	10
JQuery events, DOM access and manipulation	JQuery used for events, DOM access and manipulation to carry out the functionality required	30
AJAX	AJAX requests, handling responses, providing correct parameter values, etc.	30
Functionality	Does filtering, presentation of results in ta ble, etc, work as required	30
	Total:	/100