Iteration 3: JavaScript

1. **Form Validations and Other Events (60 points):**

    - **Inline Event Handler Approach (15 points):** Demonstrate the ability to attach event handlers directly within HTML elements.

On line 66 of preferences.html, there is an example of an inline event handler. The event handler is specified as follows: <button type="submit" style="width:100px; height:30px; float:right" onclick="alert('Your Preferences Have Been Saved!')">Save</button>.

- **Listener Approach (15 points):** Illustrate proficiency in using the `addEventListener` method to attach event handlers to elements.

There are multiple examples of ‘addEventListener’ is the listener-open.js, lazy-images.js, and profile-features.js JavaScript files.

- **Event Types (20 points):** Implement various event types, such as `click`, `onsubmit`, or `keydown`, to show different user interactions. Showcasing how to extract information like which element triggered the event or which key was pressed.

Event types such as ‘click’ and ‘scroll’ were used in my code. I extracted information from the parameter e passed to the event handling function that I wrote in profile-features.js.

- **Validating Forms (10 points):** Implement client-side validation for form inputs. This could include checks for required fields, correct data formats, or songs name matching. Demonstrate the ability to handle form submissions using JavaScript.

I validated my search bar form in the file search_valid.js. This disallowed the default submit action if the format was not two words.

2. **Manipulate Style (30 points)**

a. **Modifying a DOM Element (15 points):** Showcase the ability to select and modify DOM elements using JavaScript. This could include changing text content, attributes, or even removing/adding elements.

The listener-open.js JavaScript file is used to modify the DOM after its loaded in order to add the options of the drop-down menu into the div containing the listener menu opener.

b. **Properties (5 points):** Demonstrate understanding of object properties, possibly creating custom objects, and accessing/modifying their properties.

   Example:

   i. Creating a Custom Object:

   ```
   let radioHost = {
           name: "Alex",
           showName: "Morning Melodies",
           yearsExperience: 5

   };
   ```

   ii. Accessing Object Properties:

   ```
   console.log(radioHost.name);
   ```

   iii. Modifying Object Properties:

   ```
   radioHost.yearsExperience = 6;
   ```

   iv. Adding New Properties:

   ```
   radioHost.timeSlot = "8am - 10am";
   ```

   After these operations, the `radioHost` object would look like:

   ```
   {
    name: "Alex",
    showName: "Morning Melodies",
   ```

```
 yearsExperience: 6,
 timeSlot: "8am - 10am"
}
```

While I had no use for objects this time, I understood how to create and manipulate them using dot syntax and bracket notation. Ex: const object = { key1: value1, key2: value2}, object.key1 is the same as object[“key1”] among other properties.

c. Window Object (10 points): Explore the global `window` or ‘document’ object in JavaScript, showcasing methods or properties like `alert`, `setTimeout` or ‘DOMContentLoaded’.

Document is used to access elements and to set actions, or event-listeners, that modify the DOM after DOM content is loaded. I explored window initially in the lazy-images.js to check if the images were displayed on the screen and needed to be loaded. However, I ended up using the dimensions of the containing <div> instead.

3. Demonstrates Use of Various JavaScript Fundamentals (10 points)

- **Variables (2 points):** Using variables, showcasing understanding of data types and variable scope.

I use variables throughout the code especially in the listener-open.js program. I understand the different scopes such as block, global, and local scopes. The use of keywords let, var, and const and the lack of those meaning global is understood as well.

- **Comparison Operators (2 points):** Implement logic using comparison operators like `==`, `!=`, `>`.

I had to use logical comparison to check the current target values in profile.features.js. Specifically, I made heavy use of the “===” comparison operator.

- **Logical Operators (2 points):** Use logical operators such as `&&`, `||`, or `!`.

Similarly, I had to use logical operators to check if images were in frame in lazy-images.js. Specifically, the operator I used was conditional “and” or “&&.”

- **Conditionals (2 points):** Implement `if`, `else if`, and `else` statements.

Again, I used the “if” and “else if” statements thoroughly in profile-features.js.

- **Loops (1 point):** Use loops like `for`, `while`, or `forEach`.

I made use of forEach loops extensively in my code.

- **Functions (1 point):** Create and call functions, showcasing the ability to encapsulate code into reusable blocks.

In search_valid.js I make use of the function is valid to check the text of the search bar and change its color according to the regular expression.