Iteration 3 Rubric: SWE 432

Form Validations and Other Events: 40 pts

- Inline Event Handler Approach (10 points): Demonstrate the ability to attach event handlers directly within HTML elements.
    - Example:

    ```
    <button onclick="alert('Button was clicked!')">Click Me</button>
    ```

- Listener Approach (10 points): Illustrate proficiency in using the `addEventListener` method to attach event handlers to elements.
    - Example:

    ```
    document.querySelector("button").addEventListener("click", function() {
        alert('Button was clicked!');
    });
    ```

- Event Types (10 points):
    - Implement various event types: 8 pts
        - Example: `click`, `onsubmit`, or `keydown`
    - Event triggered successfully or action completed successfully: 2 pts

- Validating Forms (10 points):
    - Implement client-side validation for form inputs: 8 pts
        - Example: checks for required fields, correct data formats, or songs name matching.
    - Form Validations visible and successfully shown: 2 pts

Manipulate Style: 30 pts

- Modifying a DOM Element (10 points): Showcase the ability to select and modify DOM elements using JavaScript.
    - Include changing text content, attributes, or even removing/adding elements.
- Properties (10 points): Demonstrate understanding of object properties
    - Create custom objects and accessing/modifying their properties
- Window Object (10 points):
    - Explore the global `window` or 'document' object in JavaScript
    - Use methods or properties like `alert`, `setTimeout` or 'DOMContentLoaded'.

Example of Manipulate Style:

1. Creating a Custom Object:

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

2. Accessing Object Properties:

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

3. Modifying Object Properties:

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

4. 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"
}
```

Implement advanced concepts: 20 pts

- Uses prototypes/classes (10 points): Showcase the ability of reusability and object-orientated programming
- Use modules: (10 points): Split the code into different files or modules for better organization and reusability.

Example:

1. Class

```
class User
{
    constructor(name, email)
    {
        this.name = name;
        this.email = email;
    }
    login()
    {
        console.log(` ${this.name} has logged in`);
    }
}
```

2. Modules:

```
import { User } from './user.js';
```

Demonstrate use of various JavaScript fundamentals: 10 pts

- Variables (2 points): Using variables, showcasing understanding of data types and variable scope.
- Comparison Operators (2 points): Implement logic using comparison operators
  - Example: `==`, `!=`, `>`
- Logical Operators (2 points): Use logical operators such as `&&`, `||`, or `!`.
- Conditionals (2 points): Implement `if`, `else if`, and `else` statements.
- Loops (2 point): Use loops like `for`, `while`, or `forEach`.

Capabilities address role: 10 pts

- Includes functionality according to user profile: 6 pts
- Elements of JavaScript present accurately based on profile: 2 pts
- Complexity and understanding of profile displayed accurately: 2 pts

Must Follow:

- Submission Format Followed
- Submit their source code files via a GitHub repository
- Demonstration Video Included
  - Be sure to walk through rubric and show implementation for credit
- Self-Evaluation Documentation