Milestone - 7 **Frontend Development** **Codrel**

Array Methods, DOM Manipulation, and Form Validation

In this assignment, we will explore JavaScript array methods, DOM manipulation, and AJAX to strengthen our understanding of these concepts. Let's dive into each topic and complete the tasks step by step:

Part 1: JavaScript Array Methods

Task 1: Array Manipulation

Write a program that performs the following operations on an array:

1. Create an array of numbers.
2. Use the array methods to add, remove, or modify elements in the array.
3. Display the updated array in the console after each operation.

Task 2: Array Filtering and Transformation

Write a program that performs the following operations on an array of objects:

1. Create an array of objects with different properties.
2. Use array methods like filter, map, or reduce to filter and transform the array based on specific conditions.
3. Display the updated array in the console after each operation.

Part 2: DOM Manipulation

Task 3: DOM Element Creation and Manipulation

Write a program that creates, modifies, and appends DOM elements dynamically using JavaScript. Perform the following tasks:

1. Create an HTML page with a container element.
2. Use JavaScript to create new elements, modify their attributes and content, and append them to the container element.
3. Style the elements using CSS classes and apply event listeners for interactivity.

Task 4: Form Validation

Enhance the previous program to include form validation using JavaScript. Perform the following tasks:

1. Create an HTML form with input fields and a submit button.

2. Use JavaScript to validate the form inputs based on specific conditions (e.g., required fields, email format).

3. Display error messages for invalid inputs and prevent form submission if any fields are invalid.

For each task, create separate JavaScript files or functions to encapsulate the functionality. Use HTML, CSS, and JavaScript to build interactive and visually appealing applications.

Remember to break down the tasks into smaller steps and test your code frequently. Feel free to explore additional array methods and DOM manipulation techniques.

If you have any questions or need assistance, don't hesitate to ask. Good luck, and enjoy working on these assignments!

Milestone - 7 **Backend Development** **Codrel**

CRUD Book Management App

The assignment involves creating a simple CRUD (Create, Read, Update, Delete) application for managing a collection of books. Here are the requirements:

1. Set up a new Express.js project with EJS as the templating engine.

2. Create a MongoDB database and configure the connection in your project.

3. Create a Book model/schema with the following fields: title, author, genre, and description.

4. Implement an HTTP GET endpoint to display all books in a list format.

5. Implement an HTTP GET endpoint to display details of a specific book by its ID.

6. Implement an HTTP GET endpoint to render a form for adding a new book.

7. Implement an HTTP POST endpoint to add a new book to the database.

8. Implement an HTTP GET endpoint to render a form for editing an existing book.

9. Implement an HTTP PATCH endpoint to update the details of a book.

10. Implement an HTTP DELETE endpoint to remove a book from the database.

11. Use EJS templates to render the views for displaying, adding, and editing books.

12. Implement appropriate error handling and validation for the form inputs.

13. Style the application using CSS or a front-end framework of your choice.

You are free to structure your code and project files as per your preference. Make sure to follow best practices, modularize your code, and use appropriate middleware for handling requests and data validation.

To complete the assignment, you'll need to install the necessary dependencies, set up Express routes, integrate with MongoDB using a suitable library (e.g., Mongoose), create EJS templates for rendering views, and handle form submissions.

Feel free to add extra features or enhancements to the application if you'd like to challenge yourself further.

Remember to test your application thoroughly and ensure that it meets the requirements listed above.

You will need to complete till section 40 to understand and finish this project efficiently.

(You will be tested based on how good you are with creating the api and the structure for the database)

Milestone - 7 **Full Stack Development** **Codrel**

Working with npm Packages

In this assignment, we will be exploring various npm packages to understand their functionalities and how they can enhance our JavaScript programs. Let's dive into each package and complete the tasks step by step:

1. Chance:

Task: Write a program that generates a random name, email address, and phone number using the Chance package. Display the generated data in the console.

2. Axios:

Task: Create a program that makes an HTTP GET request to an API endpoint of your choice using Axios. Retrieve data from the response and display it in the console.

3. Moment.js:

Task: Write a program that uses Moment.js to format the current date and time in a specific format. Display the formatted date and time in the console.

4. Lodash:

Task: Implement a program that uses Lodash to sort an array of numbers in ascending order. Display the sorted array in the console.

5. Request:

Task: Create a program that makes an HTTP POST request to an API endpoint of your choice using the Request package. Send JSON data in the request body and handle the response.

6. Chalk:

Task: Build a program that uses Chalk to add different colors and styles to console messages. Print messages with different colors and styles to the console.

7. Inquirer:

Task: Write a program that uses Inquirer to prompt the user for their name, email address, and age. Display the collected information in the console.

8. Validator.js:

Task: Implement a program that validates a user's email address using the Validator.js package. Prompt the user to enter an email address and display a message indicating whether it is valid or not.

For each task, create a separate JavaScript file and follow these steps:

1. Initialize a new Node.js project using npm.

2. Install the required npm packages for the respective task.

3. Implement the functionality as described in the task.

4. Run the program and observe the output in the console.

Remember to refer to the package documentation, examples, and online resources to understand the usage of each package. Feel free to explore additional functionalities and customize the programs according to your curiosity and creativity.

I encourage you to complete these tasks in order, as each one builds on the concepts of the previous package. Don't hesitate to reach out if you have any questions or need assistance. Good luck, and have fun exploring the world of npm packages!