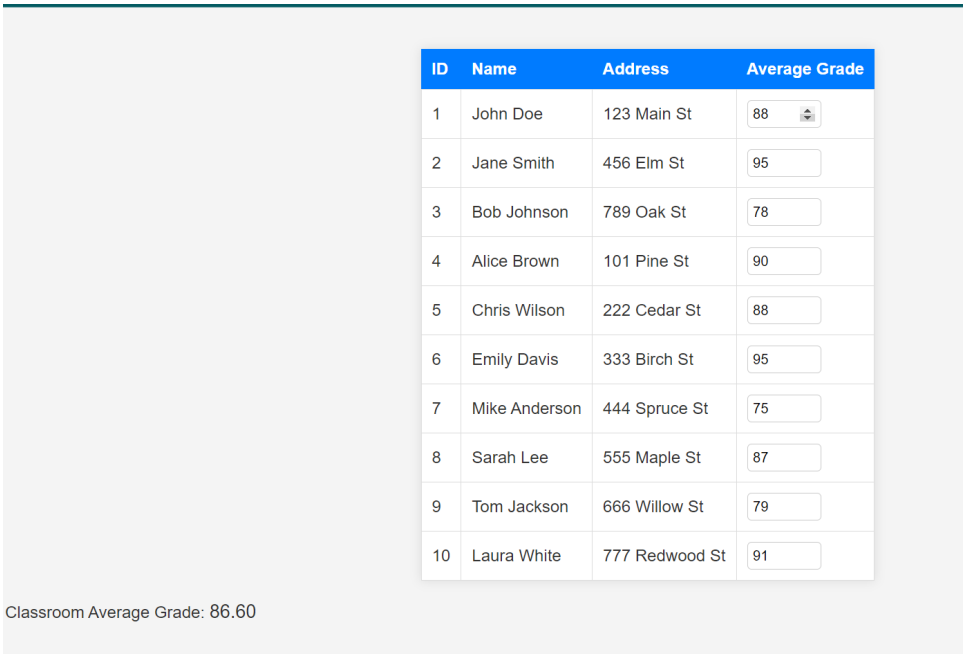
Classroom Average

Created by	W Wren Purcell
Created time	@November 16, 2023 11:31 AM
Tags	
written by	@Tea Melendez



ID	Name	Address	Average Grade
1	John Doe	123 Main St	88
2	Jane Smith	456 Elm St	95
3	Bob Johnson	789 Oak St	78
4	Alice Brown	101 Pine St	90
5	Chris Wilson	222 Cedar St	88
6	Emily Davis	333 Birch St	95
7	Mike Anderson	444 Spruce St	75
8	Sarah Lee	555 Maple St	87
9	Tom Jackson	666 Willow St	79
10	Laura White	777 Redwood St	91

Classroom Average Grade: 86.60

Objective:

In this assignment, you will build a basic web application for managing student grades. You will use JavaScript to create, display, and update student data within an HTML table. Additionally, you will calculate and display the classroom's average grade.

Skills to Learn:

- JavaScript programming
- DOM manipulation
- Modular development

- Event handling
- Importing and exporting modules

Instructions:

Step 1: Project Setup

- Create a new web development project using a code editor or online IDE.
- Set up the project structure with the following files inside a new folder you can name classroom-average:
  - **index.html** to display the student data and classroom average.
    - you can create a table in the html to use like this:

```
<table id="student-table">
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Name</th>
                    <th>Address</th>
                    <th>Average Grade</th>
                </tr>
            </thead>
            <tbody></tbody>
        </table>
```

  - **styles.css** for styling the HTML elements.
  - **studentData.js** to store an array of student information.
  - **studentTable.js** for creating and updating the student table.
  - **calculateClassroomAverage.js** for calculating the classroom average grade.
  - **main.js** for the main application logic.
- Link the necessary CSS and JavaScript files to your **index.html** file.
- Refer to **Importing and Exporting JS modules.pdf**

Step 2: Data Setup

In the **studentData.js** file, create an **array** of student **objects**. Each student should have the following properties:

- id: A unique student identifier.
- name: The student's name.
- address: The student's address.
- averageGrade: The student's initial average grade.
- you may use this sample data:

```
{
    id: 1,
    name: "Alice",
    address: "123 Main Street",
    averageGrade: 85,
  },
  {
    id: 2,
    name: "Bob",
    address: "456 Elm Street",
    averageGrade: 78,
  },
  {
    id: 3,
    name: "Charlie",
    address: "789 Oak Street",
    averageGrade: 92,
  },
  {
    id: 4,
    name: "David",
    address: "321 Pine Street",
    averageGrade: 70,
  },
```

export the array

Step 3: Build the Student Table

In the **studentTable.js file,**

- Import the student data from **studentData.js**
- create a function named `renderStudentTable` that takes two parameters:

    - `table` (a reference to the HTML table where the data will be displayed).
    - `data` (the student data from **studentData.js**).

    Use a for loop to iterate through the data array and dynamically create table rows to display each student's information.

- Use DOM manipulation to insert the student data into the table cells, including their ID, name, address,
    - for the average, create an input field for that users can edit.
    - you can research more DOM methods to use for this
- Implement an `onchange` event handler that calls a function named `updateGrade` when a student's average grade is modified. This function will update the student's grade and recalculate the classroom average.

Step 4: Update Student Grades

- Create a function named `updateGrade` in the `studentTable.js` file, which takes an input element as a parameter.
- Use this function for updating a student's grade.
- Extract the student ID and new grade from the input element.
    - research methods to do this
- Update the student's average grade in the `studentData` array.
- Recalculate the classroom average grade by calling the `calculateClassroomAverage` function from the `calculateClassroomAverage.js` module.
- Update the classroom average display in the HTML table.

Step 5: Calculate Classroom Average

In the **calculateClassroomAverage.js** file, create a function named `calculateClassroomAverage` that takes the `studentData` as a parameter. Use a higher order function to iterate through the student data and calculate the average of all student grades. Return the classroom average grade.

Step 6: Main Application Logic

In the main.js file, import the necessary functions and modules:

- `renderStudentTable` and `updateGrade` from `studentTable.js`.
  `calculateClassroomAverage` from **calculateClassroomAverage.js**
  `studentData` from **studentData.js**.
- Add an event listener for the `DOMContentLoaded` event to ensure the HTML elements are ready. Initialize the table with student data using the `renderStudentTable` function. Calculate and display the initial classroom average using the `calculateClassroomAverage` function. Set up an event listener to detect changes in student grades and trigger the `updateGrade` function.

Step 7: Styling

Apply CSS styles to make the student table and classroom average visually appealing. You can use the styles.css file for this purpose.

Step 8: Testing and Submission

- Test your web application to ensure it functions correctly.
- Verify that you can add and modify student grades.
- Check that the classroom average is updated accordingly.
- Create a README.md file that provides instructions on how to run your project.
- Host your project on GitHub.
- Submit the repository link.

Grading Criteria:

Your project will be evaluated based on the following criteria:

- Proper project structure and organization.
- Correct implementation of functions and modules.
- Functionality of the student table and classroom average.
- User interface and styling.
- Documentation and instructions provided in the README.md.