**Problem Statement:**

The Student Management System (SMS) provides a streamlined and efficient way to manage student data. The key functionalities of the system include adding, updating, retrieving, and deleting student records. Below is a detailed explanation of each functionality:

Student Structure: id, name, age

**1. Add Student**

* **Description:** Allows the administrator to add a new student to the system.
* **Input:** The system requires the following details:
  + **ID:** A unique identifier for each student.
  + **Name:** The full name of the student.
  + **Age:** The student's age.
* **Process:** The system validates the input data to ensure that the ID is unique and that all required fields are provided. Once validated, the student is added to the list of students.
* **Output:** Confirmation of successful addition, or an error message if the ID is not unique or data is invalid.

**2. Retrieve Student by ID**

* **Description:** Allows the administrator to search for and retrieve a student's details using their unique ID.
* **Input:** The unique ID of the student.
* **Process:** The system searches for the student in the list using the provided ID. If found, it retrieves the student's details.
* **Output:** The student's details, including ID, name, and age. If the student is not found, an appropriate message is displayed.

**3. Update Student**

* **Description:** Enables the administrator to update the details of an existing student.
* **Input:** The unique ID of the student and the updated details (name, age, etc.).
* **Process:** The system first checks if the student exists using the provided ID. If the student exists, the system updates the relevant details with the new data provided.
* **Output:** Confirmation of successful update, or an error message if the student ID is not found.

**4. Delete Student**

* **Description:** Allows the administrator to remove a student's record from the system.
* **Input:** The unique ID of the student to be deleted.
* **Process:** The system checks if the student exists in the list. If found, the student's record is removed from the list.
* **Output:** Confirmation of successful deletion, or an error message if the student ID is not found.

**5. Get All Students**

* **Description:** Provides a way to retrieve a list of all students currently in the system.
* **Input:** No specific input is required.
* **Process:** The system retrieves all student records from the list.
* **Output:** A list of all students, including their IDs, names, and ages.

**Note: Please update only the “studentManager.js” file.**

**Test Case 1: Add a new student**

* **Expected Outcome**: The StudentManager should contain the newly added student when retrieving the list of all students.

**Test Case 2: Retrieve a student by ID**

* **Expected Outcome**: The returned student object should be the same as the one added (matching ID, name, and age).

**Test Case 3: Update a student**

* **Expected Outcome**: The student's name should be "Jane Doe", and their age should be 21 after the update.

**Test Case 4: Delete a student**

* **Expected Outcome**: The student with ID 1 should no longer exist in the StudentManager.

**Test Case 5: UI 4 Marks**

* **Expected Outcome:** 
  + The web pages should be responsive
  + The web pages should be neat and clean