# Simple Student Management Project ( C# project 1 )

**Develop a simple Student Management System that allows a user (admin) to manage student records.**
The program should support the following operations:

1. **Add a new student record** (Name, Age, Marks)
2. **View all students with formatted output and subject-wise marks.**
3. **Find a student by name (case-insensitive search)**
4. **Calculate the class average (rounded to 2 decimals).**
5. **Find the top-performing student**
6. **Sort students by marks (highest to lowest)**
7. **Delete a student record (handle shifting logic).**
8. **Exit the system**

---

**Required Logic and Implementation Details**

1. **Menu System (Using switch-case & loop)**

- The program should display a menu allowing the admin to select an option.
- The menu should **keep repeating** until the user selects "Exit."

2. **Storing Student Data (Using Arrays)**

- Use **parallel arrays**:
    - string[] names → Student Names
    - int[] ages → Student Ages
    - double[] marks → Student Marks
    - DateTime[] enrollmentDate → Student Enrollment Date
- Keep track of the **total number of students** using a variable.

3. **Adding a New Student (Input & Array Insertion)**

- Ask for student details (name, age, marks).
    - age (validated > 21 ), marks (validated 0-100), and enrollment date (DateTime.Now).
- Store them in the **next available index** of the arrays.
- Ensure the user **cannot add more than MAX_STUDENTS**.

4. **Viewing All Students**

- Loop through the arrays and display **all stored students**.

5. **Searching for a Student by Name**

- Ask the user for a **name**.
- Loop through the names array to **find a match**.
    - Convert both **input and stored names** to lowercase before comparing.
- If found, print the student details. Otherwise, display "Not found".

6. **Calculating the Class Average**

- Loop through marks[] and **sum all values**.
- Divide by studentCount and **round the result** using Math.Round().

7. **Sorting Students by Marks (Descending Order)**

- Sort marks[] in **descending order**, swapping names[] and ages[] accordingly.

8. **Deleting a Student**

- Ask for a **name** to delete.
- Find the **index of the student**.
- Shift all elements **to the left** to remove the record.

---