

# Student Management System

## Report card

### Category | Evaluation

**Code Structure:** Clear modular design with functions like `addStudent()`, `displayStudents()`, `searchStudent()`, `deleteStudent()`, and `updateStudent()`. Very organized and readable.

**Functionality:** Basic student operations like adding, searching, updating, deleting, and displaying students work effectively.

**Input Handling:** Good handling overall, but slight improvement possible by using `fgets` instead of `scanf` to correctly handle spaces in inputs like names.

**Error Handling:** Basic error messages are provided (e.g., "Student not found"), but input validation (like checking valid year/semester ranges) can be added for robustness.

**Memory Management:** No dynamic memory (`malloc/free`) is used, so memory management concerns are minimal.

**File Handling:** No file operations (save/load data) included. Adding persistence would be a good enhancement.

**Coding Style:** Very readable variable names, consistent indentation, and separation of functionalities.

**Efficiency:** Efficient for a small number of students.

**Expandability:** Good base structure; can be expanded by adding sorting, file operations, etc.

**Grade:** A (Good Effort!)

**Remarks:** Code is functional, well-structured, and easy to understand. Minor improvements can make it even better.

### Strengths

- Good modular design.
- Covers all basic student management operations.
- Readable code with consistent formatting.

### Areas to Improve:

- Use `fgets` for better input handling.
- Validate fields like year, semester, age, and marks.
- Implement file handling to save/load students.
- Add boundary checks for fields like marks.

### Bonus Suggestions (Future Improvements):

- Save and load student data to/from a file.
- Sort students based on marks, name, or roll number.
- Display top performers.- Prevent duplicate roll numbers.- Add batch search or advanced filters.