

**Student Record System**  
**COAL Lab - Project Proposal**

**Group Members:**

***Rayyan Aamir (24K-0687) (Group Leader)***

***M. Usaid Khan (24K-0832)***

***Syed M. Furqan (24K-0766)***

## **Introduction:**

### ***Background:***

Student data management is a fundamental task in educational institutions. While high-level languages offer built-in support for such systems, implementing a student record system in Assembly Language provides valuable insights into low-level data handling, memory operations, and CPU-based execution flow. This project demonstrates how structured data (name, roll number, GPA) can be effectively managed using the x86 MASM Assembler with the Irvine32 library in Visual Studio 2022.

### ***Problem Statement:***

There is often a gap in understanding how basic record systems function at the machine level. Students rely on modern programming abstractions but lack hands-on exposure to how memory is allocated, searched, and sorted at the lowest level of programming. This project addresses this issue by building a student record management system entirely in Assembly Language, enabling learners to directly interact with CPU instructions for search, sort, and display operations.

### ***Objectives:***

- To implement a student record system that accepts Name, Roll Number, and GPA as inputs.
- To store these records in memory using arrays.
- To allow searching of records by roll number.
- To implement sorting of records by GPA (ascending) and by Name (alphabetical).
- To display results in a structured, tabular format.
- To strengthen the understanding of low-level algorithms and data representation.

## **Scope of the Project:**

### ***Inclusions:***

- Input of multiple student records.
- Storage of records in arrays (name → BYTE, roll number → WORD/DWORD, GPA → REAL4).
- Search by roll number.
- Sorting by GPA or name.
- Display of results in a formatted console output.

### ***Exclusions:***

- No permanent file storage (records exist only during program execution).
- No GUI (limited to text-based console).
- No advanced database or network features.

## **Project Description:**

### ***Overview:***

The Student Record Management System is a console-based Assembly Language program that manages multiple student records in memory. It allows data entry, searching, sorting, and displaying of records. This project emphasizes understanding of Assembly-level data operations, providing students with practical exposure to managing structured information without high-level abstractions.

### ***Technical Requirements:***

- Software:
  - ✓ Visual Studio 2022 (VS2022) with MASM assembler.
  - ✓ Irvine32 Library for console I/O operations.

➤ **Hardware:**

- ✓ A standard desktop or laptop with Windows OS, 8GB RAM minimum, and x86 processor compatibility.

***Project Phases:***

1. Phase 1 – Data Entry: Implement input module to accept student details and store them in memory.
2. Phase 2 – Search Module: Implement linear search by roll number.
3. Phase 3 – Sorting Module: Implement sorting (Bubble Sort/Selection Sort) by GPA and by Name.
4. Phase 4 – Display & Integration: Format records into tabular output and integrate all modules.
5. Phase 5 – Testing & Debugging: Validate input handling, search, and sorting results.

**Methodology:**

***Approach:***

The project will be developed in iterative phases, starting with data entry, followed by searching and sorting, and finally integration. Testing will be carried out after each phase. Development will follow a modular approach, ensuring that each part of the program (input, search, sort, display) can be tested independently before integration.

***Team Responsibilities:***

- **Syed M. Furqan:** Input & Storage
- ✓ Handles record input and validation.
  - ✓ Implements memory allocation for name, roll number, and GPA arrays.

- **Rayyan Aamir:** Search & Sort Algorithms
  - ✓ Develops roll number–based linear search.
  - ✓ Implements sorting functionality by GPA and Name.
- **Usaid Khan:** Integration & Output
  - ✓ Combines all modules into a single executable program.
  - ✓ Designs console output formatting for readability.
  - ✓ Performs debugging and ensures proper functionality.

## **Expected Outcomes:**

### ***Deliverables:***

- Software Deliverables:
  - ✓ Assembly Language program (.asm file) implementing the student record system.
  - ✓ Source code tested in Visual Studio 2022 with Irvine32 support.
  - ✓ Sample test cases for input, search, and sorting.
- Report Deliverables:
  - ✓ Project Proposal Document.
  - ✓ Final Project Report detailing system design, implementation, and algorithms.
  - ✓ Presentation Slides summarizing methodology and results.

### ***Relevance:***

- Demonstrates low-level data representation and memory management in Assembly.
- Provides practical exposure to sorting and searching algorithms at the CPU instruction level.
- Strengthens team collaboration and project planning skills.

- Connects directly to ICT concepts of data organization, processing, and low-level programming.

## **Resources Needed:**

### ***Software:***

- Visual Studio 2022 (VS2022).
- Irvine32 Library.
- Microsoft Office (for report and presentation preparation).

### ***Other Resources:***

- Online tutorials and reference material on MASM Assembly.
- Support from course instructor for debugging and validation.
- Access to lab computers with MASM and Irvine32 configured.