SINDH MADRESSTUL ISLAM UNIVERSITY
Department Of Software Engineering
(BSSE Fall 2025)

2025

FINAL PROJECT

ATMS
(ATTENDANCE MANAGEMENT SYSTEM)

Prepared for:
Miss. Iqra

Prepared By:
Okasha Nadeem
BSE-25F-086
Muhammad Tabish
BSE-25F-068
Muhammad Yahya Alvi
BSE-25F-061
Syed Muhammad Sarim
BSE-25-062

DATE OF SUBMISSION
29/12/2025

Contents

1. Introduction:

This project is an **Attendance Management System** developed using the C programming language. It is a console-based application designed to modernize the traditional method of tracking student attendance. The system allows users to register students, mark their daily attendance (Present/Absent), and view attendance records efficiently. By utilizing a menu-driven interface, the project demonstrates core programming fundamentals such as data structures, control flow, and user input handling, while providing a practical solution for classroom management.

2. Objective:

To design and develop a user-friendly, console-based Attendance Management System that automates the process of recording and retrieving student attendance data using standard C programming structures.

3. Problem Statement:

What: The current manual system of taking attendance using pen and paper is time-consuming, prone to human error, and difficult to manage as the number of students increases. Records can be easily lost or damaged, and calculating attendance history for individual students is a tedious task.

Why: Digitizing this process is essential to ensure accuracy and efficiency. A computerized system eliminates physical paperwork and allows for instant retrieval of records. This project addresses the issues of manual logging by providing a structured, digital method to store student names, IDs, and their attendance status, ensuring that data is organized and easily accessible for instructors.

4. Methodology:

How: The system is built using the **C programming language**, employing a **procedural programming approach**.

- **Data Storage:** We will use **Arrays of Structures** (struct) to store student details (Name, ID) and their corresponding attendance records in the system's RAM.
- **Interface:** A do-while loop will power a menu-driven interface, allowing the user to navigate between adding students, marking attendance, and viewing reports until they choose to exit.
- **Logic:** The system utilizes standard Input/Output functions (printf, scanf) for user interaction and string manipulation functions to handle student data.

5. Project Scope:

In Scope:

- **Registration:** Ability to add new students with unique IDs and names.
- **Attendance Marking:** Marking "Present" or "Absent" for a specific date.
- **Reporting:** Viewing a list of all registered students and detailed attendance logs for individual students.
- **Platform:** The system is designed for Windows-based console environments.

Out of Scope:

- **Persistent Storage:** This version does not use file handling (e.g., .txt or .csv files) or databases; data is volatile and exists only while the program is running.
- **GUI:** The project does not include a Graphical User Interface; it is strictly command-line based.
- **Validation:** Advanced validation (e.g., preventing duplicate IDs or validating date formats) is not included in this iteration.

6. Feasibility Study:

Risks Involved:

- **Data Volatility:** Since the system uses RAM for storage, a sudden power failure or program exit will result in the loss of all entered data. This risk is accepted as the project focuses on logic building rather than commercial deployment.

Resource Requirement:

- **Hardware:** A standard personal computer or laptop.
- **Software:** A C Compiler (GCC, Dev-C++, or MSVC) and a Windows Operating System (for specific console commands like system("cls")).

7. Solution Application Areas:

8. **Target Domain:** The primary target domain is **Education**. This system is applicable for small-scale educational setups, such as coaching centers, individual classrooms, or tuition academies, where a simple and quick method for digital attendance tracking is required without the complexity of enterprise software.
9. **Benefits:** Instructors can save time usually spent on manual roll calls and have an immediate view of student regularity.

10. Tools/Technology:

- Programming **Language:** C
- Compiler**:** GCC / Dev-C++ / MSVC
- IDE**:** Dev-C++ or Visual Studio Code
- Operating **System:** Windows (Recommended for system("cls") compatibility)

11. Expertise of the Team Members:

The team members are currently pursuing a degree in Software Engineering and have completed the **Programming Fundamentals** course.

- **Skills:** The team is proficient in C syntax, control structures (loops, conditions), functions, and data structures (arrays, structs).
- **Interest:** All members share a strong interest in software development and logic building, ensuring equal contribution to the project's success.

10. Milestones:

- Project **Proposal:** defining objectives and scope.
- Algorithm **Design:** Planning the data structures (Student and Attendance
    - structs) and flowcharts.
- Implementation **(Phase 1):** Coding the add Student and view all Students
    - functions.
- Implementation **(Phase 2):** Coding the mark Attendance logic and ensuring
- data linkage.
- Testing**:** Running the program on Dev-C++ to check for bugs and compilation
    - errors.
- Final **Documentation:** Preparing the final report and presentation.

11. References

- Kernighan, Brian W., and Dennis M. Ritchie. *"The C Programming Language"*. Prentice Hall, 1988.
- "*C Tutorial - Learn C Programming"*. W3Schools. [Accessed Dec 2025]. https://www.w3schools.com/c/
- Prata, Stephen. *"C Primer Plus"*. Addison-Wesley Professional, 2013.