PROJECT REPORT

STUDENT RECORD MANAGEMENT SYSTEM (SRMS)

SRM
UNIVERSITY AP
Andhra Pradesh

DONE BY

NAME: Y siddarda

REGISTRATION NUMBER: AP24110011084

CSE 2ND YEAR THIRD SEMESTER

SECTION R

SUBMITTED TO

RAKESH RAMA RAJU

CODING SKILLS – I (CSE 201)

CERTIFICATE

This is to certify that the project entitled “Student report management system” has been successfully Completed by Y SIDDARDA (AP24110011084) as a required academic component of the course CODING SKILLS CSE 201 during Semester 3 of Academic Year 2025-26 in the Department of Computer Science and Engineering at SRM University, AP. This work was executed under the guidance of the undersigned and is deemed to have successfully met all course requirements as of 6th December 2025

(Signature of the Course Instructor)

Dr. Rakesh Rama Raju P

Campus Corporate Connect

ABSTRACT

This project presents a Student Record Management System developed in the C programming language to offer a reliable, secure and easy-to-use solution for managing student information. The system stores key details such as roll number, name and marks in text files, ensuring that all data is safely preserved even after the program closes. It provides the essential operations required for effective record management, including adding new entries, viewing stored data, searching for specific students, updating information and deleting records when necessary.

A role-based login system forms the foundation of its security and usability. Users can log in as Admin, Staff or Guest, and each role is granted only the level of access needed for its responsibilities. Admin users can perform all operations, staff users can view and search records and guest users are limited to viewing information. This structured approach not only protects the system from unauthorized modifications but also makes the interface simple and user-friendly for every type of user.

Overall, this project demonstrates the practical importance of core C programming concepts while showing how a straightforward application can effectively represent real-world data management. It stands as a highly valuable learning tool that combines simplicity, security and smooth usability to create a dependable student information management system.

INTRODUCTION

The **Student Record Management System** is a C-based application developed to efficiently manage and maintain student information using basic file-handling techniques. The system stores essential details such as roll number, name, and marks in external text files, ensuring that the data remains preserved even after the program is terminated. To ensure secure and organized access, the application incorporates a role-based login mechanism that restricts operations based on user privileges.

The system supports three distinct user roles: **Admin, Staff, and Guest**. Admin users possess complete access and can add, update, delete, search, and view student records. Staff members are granted permissions to search and view existing records, whereas Guest users are limited to viewing the data. This layered permission structure promotes data integrity and prevents unauthorized modifications.

In addition to managing student data, the project demonstrates the practical application of key concepts in the C programming language, including structures, file operations, string handling, and conditional logic. Overall, the Student Management System serves as a foundational example of how real-world data management applications can be designed and implemented using fundamental programming principles.

Why C Language?

✓ Low-level file handling capability
✓ Fast execution
✓ Large compiler support
✓ Good for learning fundamentals of storage and memory

This project is specially designed to:

- Teach the concept of file operations
- Demonstrate structure usage for storing student information
- Introduce authentication mechanisms
- Implement role-based access control

OBJECTIVES

The main objectives of this project are:

- To create an easy and efficient system for managing student information using C programming.
- To add a secure login system with different roles for Admin, Staff and Guest users.
- To give each user type the right level of access so the system is used properly.
- To store student details in text files so the data can be saved for a long time and accessed easily.
- To allow Admin users to add, view, search, update and delete student records.
- To allow Staff users to only view and search student information without making any changes.
- To give Guest users read-only access to keep the data safe.
- To show the practical use of file handling, structures and conditional statements in C.
- To protect the accuracy of the data by preventing unauthorized changes.
- To provide a simple, user-friendly, menu-based interface for smooth and easy navigation.

PROBLEM STATEMENT

Traditional student record management systems rely on paper-based documentation, which results in:

❌ Difficulty in maintaining large amounts of data
❌ Time-consuming search process
❌ Data insecurity and risk of damage
❌ Manual errors in editing or deleting records

Therefore, there is a need for an automated software application that:

✓ Stores data securely
✓ Reduces manual workload
✓ Provides fast search and update mechanisms
✓ Ensures secure access to student information

The proposed system addresses these limitations by developing a Student Management System using C language.

PROJECT LAYOUT / SYSTEM ARCHITECTURE

System Workflow:

```
 +---------------+     +------------------+     +---------------------+
| User/Admin    | ---> | Login System     | ---> | Role-based Menu     |
+---------------+     +------------------+     +---------------------+
                              |
                              v
                 +-------------------------------+
                 | Student File Operations       |
                 | Add / Search / Delete / Update|
                 +-------------------------------+
                              |
                              v
                  +----------------------------+
                  | Output / Data Stored       |
                  +----------------------------+
```

Data Storage Format:

```
RollNo Name Marks
101   Ram  89.50
102   Ravi 78.00
```

MODULES

The system consists of the following major modules:

1. Login Module

- Takes username and password.
- Checks credentials from credentials.txt.
- Identifies role: Admin, Staff, or Guest.

2. Role-Based Menu Module

- Shows menu according to user role.
- Controls access to operations for Admin, Staff, and Guest.

3. Student Addition Module (Admin Only)

- Allows adding new student details.
- Saves roll number, name, and marks in students.txt.

4. Display Students Module

- Reads all records from students.txt.
- Displays roll number, name, and marks.

5. Search Student Module

- Searches for a student by roll number.
- Shows matching record.

6. Update Student Module

- Updates student name or marks.
- Uses a temporary file to save updated data.

7. Delete Student Module (Admin Only)

- Deletes a student record using roll number.
- Rewrites file without the deleted entry.

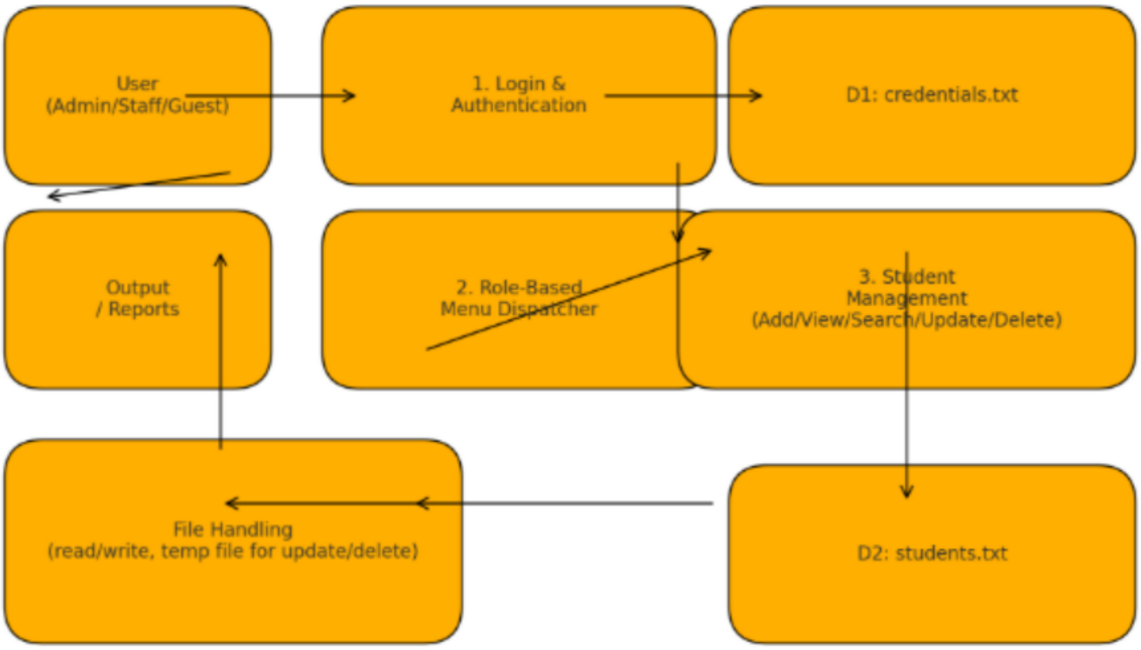
8. File Handling Module

- Handles reading and writing of credentials.txt and students.txt.

FEATURES IMPLEMENTED

Feature	Description
Authentication System	Only valid users can access the system
Role-Based Access Control	Admin > User in permissions
File Handling	Records stored using text file
Error Handling	Checks file exists or not
Non-destructive Update/Delete	Temporary file to rewrite updated content
Simple UI	Command-line based menu system

DATA FLOW DIAGRAM



Explanation

1. Login & Authentication:
   The system checks the entered username and password with *credentials.txt* and identifies the user role.

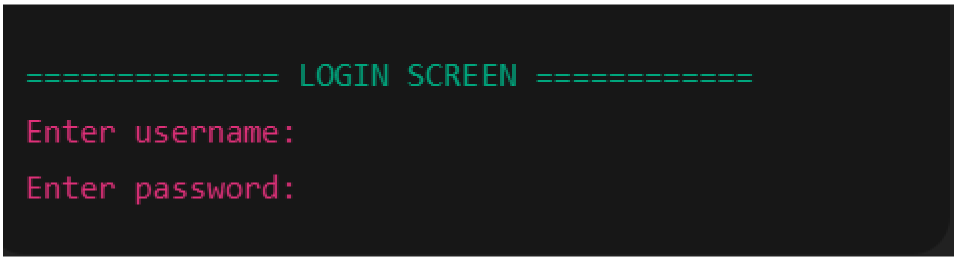
2. Role-Based Menu:
   Based on the role (Admin, Staff, Guest), the appropriate menu is shown and the chosen option is passed to the next process.

3. Student Management:
   Handles adding, viewing, searching, updating and deleting student records. Updates and deletions use a temporary file for safe rewriting.

4. File Handling:
   All student data is stored in ***students.txt***, ensuring permanent storage and reliable read/write operations.

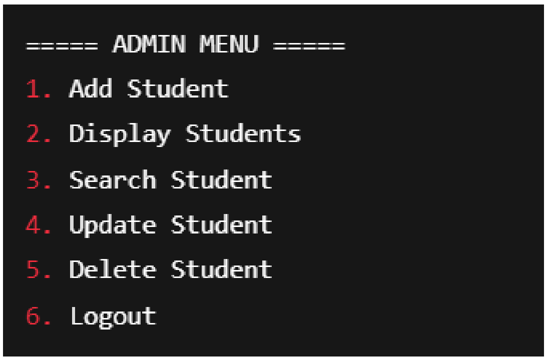
CODE & OUTPUTS

Login Screen



```
============== LOGIN SCREEN ============
Enter username:
Enter password:
```

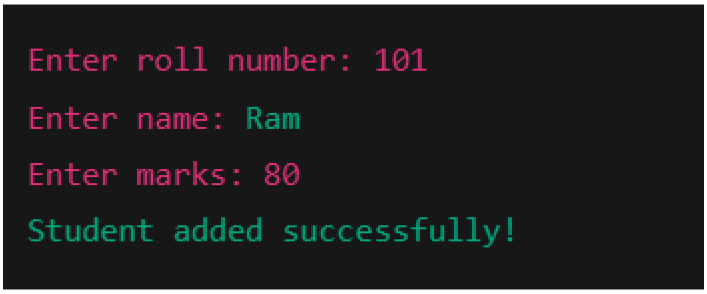
Admin Menu



```
===== ADMIN MENU =====
1. Add Student
2. Display Students
3. Search Student
4. Update Student
5. Delete Student
6. Logout
```

Student Added Successfully

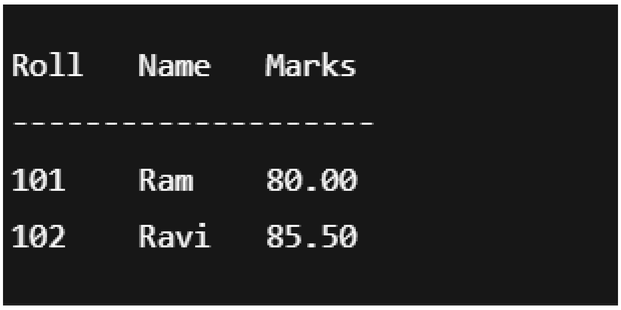
After login, the Admin Menu appears, allowing the admin to add, view, search, update, or delete student records.



```
Enter roll number: 101
Enter name: Ram
Enter marks: 80
Student added successfully!
```

Display Records

This output shows the list of all stored student records. It displays roll numbers, names, and marks in a structured table format.

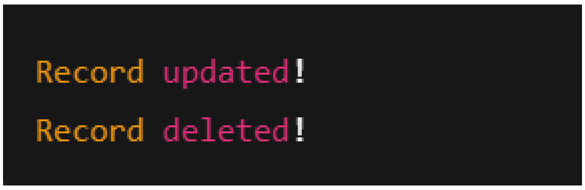


```
Roll   Name   Marks
--------------------
101    Ram    80.00
102    Ravi   85.50
```

Update/Delete Confirmation

The update screen enables modification of an existing student's details. The admin can update the student's name and marks after entering the roll number.

This output shows the result of deleting a student record. The system removes the matching roll number from the database and confirms the action.



```
Record updated!
Record deleted!
```

CONCLUSION

This project successfully demonstrates how to build a Student Management System using C programming language and text file storage. It provides secured login authentication and role-based access control. Admin can manage student data such as adding, updating, deleting and viewing records, whereas user can only view and search information.

The system solves the problem of manual record management by:
✓ Improving data accuracy
✓ Reducing paperwork
✓ Increasing retrieval speed
✓ Ensuring secure access

It is a helpful desktop-based application for understanding file handling in C and can be deployed in small-scale educational institutes.

e Student Record Management System is a simple program that helps store and manage student information in an easy and organised way. Users can add, view, search, update, and delete student records depending on their role. Admin, Staff, and Guest users have different levels of access, which helps keep the records safe and prevents unwanted changes. The system is menu-driven and easy to understand, making it simple for anyone to use.

This project shows how basic C programming concepts like structures, functions, loops, conditions, and simple file storage can be used to build a small working application. It performs all the main tasks needed for handling student records in a clear and reliable way. Overall, the Student Record Management System is a good example of how useful applications can be created using simple and fundamental C programming techniques.

Future Enhancements

This project can be enhanced by integrating:

✓ Graphical User Interface (GUI) using GTK or Visual Studio
✓ Password encryption for improved security
✓ Multiple admin/user roles with registration options
✓ Sorting and filtering in display screen
✓ Backup and restore database features
✓ Storage in binary file for faster access
✓ Migration to SQL/Cloud databases for large-scale usage
✓ Mobile or Web Application version

These improvements will make the system more professional, secure, and scalable.

References

E. Balagurusamy — Programming in ANSI C
Byron S. Gottfried — Schaum's Outline of Programming with C
The C Programming Language – Kernighan & Ritchie (K&R)
TutorialsPoint — File Handling in C
Javatpoint — C Structure and File Handling
Online resources and lecture notes
greeks for greeks - Search