A REPORT

ON
Student record management system

By

M.JOWSHNAVI (AP24110010927)

Prepared in the partial fulfillment of the

Project Based Learning of Course CSE 201 -coding skills-1

SRM UNIVERSITY, AP

DECEMBER 2025

TABLE OF CONTENTS

1. Introduction

The “Student Record Management System” is a C-language based project designed to help manage basic details of students, staff, and admin roles in a college environment. The main aim of this project is to store and manage information such as attendance, timetable, fees, notices, and personal details in an organized way.

Instead of writing or tracking everything manually, this system allows the user to interact through a menu-driven interface where each role (Admin, Student, Staff, User) has its own functions.

This project helped me understand file handling in C, login authentication, conditional operations, and creating a modular program with multiple features.

2. Objectives of the Project

The main objectives of this system are:

To maintain records of students in a structured format

To provide login security using username and password

To allow different roles (Admin, Staff, Student, User) with different levels of access

To manage important student information like attendance, timetable, fees, and notices

To understand the use of C programming concepts like structures, functions, and files

Features of the System

1. Login System

Every user must enter a username and password.

If the details match the file passwords.txt, the user is allowed to continue.

2. Admin Features

The admin has full control:

Add new student

Update student details

View student list

Enter notices

Update timetable

Check fees and attendance of any student

3. Student Features

The student can:

View own attendance

View timetable

Check fees status

View notices

4. Staff Features

The staff member can:

Mark attendance

Enter marks (if needed)

Update notices (optional)

View student details

5. General User

General user can only:

View notices

View timetable

Technologies / Concepts Used

C Programming Language

File Handling for storing:

login credentials

attendance

timetable

notices

student details

Structures for storing grouped data

Functions for different role operations

Conditional statements and loops

Basic error checking

System Design

Modules:

1. Login Module: This module verifies the username and password of Admin, Student, Staff, and User to allow secure access.

2. Admin Module: The Admin module provides controls to manage users, update records, view reports, and oversee the entire system.

3. Student Module: This module allows students to view their attendance, timetable, fees details, and notice board updates**.**

4. Staff Module: Staff can mark attendance, update student information, and manage academic-related data using this module.

5. User Module: A general-purpose module that allows basic users to view public information like notices and announcements.

6. Attendance Module: This module stores, updates, and displays attendance records for each student.

7. Timetable Module: The timetable module shows the scheduled classes and allows updates by admin or staff.

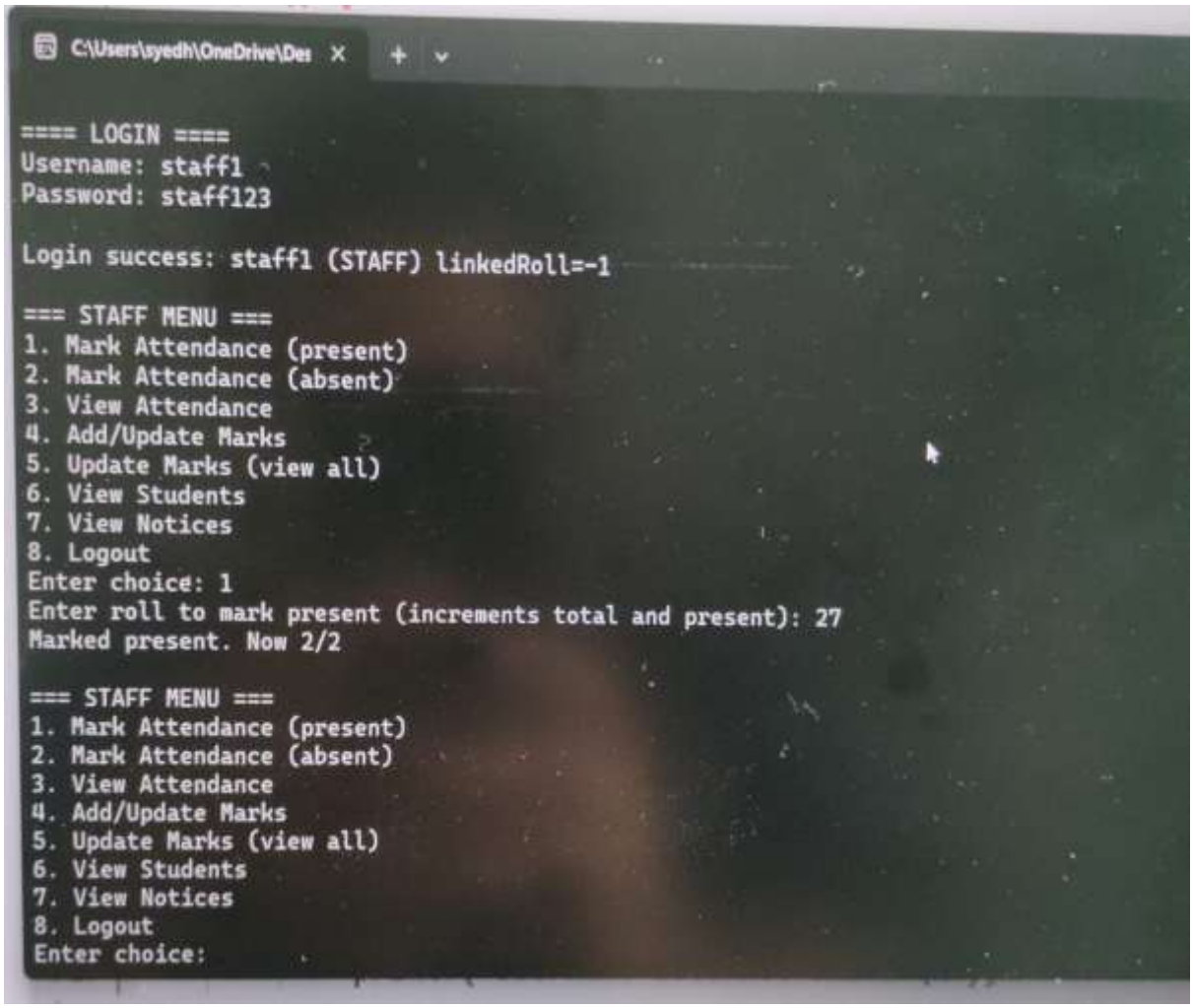
8. Notice Board Module: This module displays important announcements, events, and updates for all users.

9. Fees Module: The fees module manages fee details, payment status, and allows students to view their fee records.

Code Snippet:

```
int main() {
    printf("Student Record Management System (Multi-role)\n");
    if (!fileExists("users.txt")) {
        FILE *fp = fopen("users.txt","w");
        if (fp) {
            fprintf(fp, "admin admin123 ADMIN\n");
            fclose(fp);
            printf("Created default users.txt with admin/admin123\n");
        }
    }
    while (1) {
        int res = login();
        if (res == 1) {
            printf("\nLogin success: %s (%s) linkedRoll=%d\n", currentUsername, currentRole, curre
            if (strcmp(currentRole,"ADMIN")==0) adminMenu();
            else if (strcmp(currentRole,"STAFF")==0) staffMenu();
            else if (strcmp(currentRole,"STUDENT")==0) studentMenu();
            else userMenu();
            printf("Logged out.\n");
        } else if (res == -1) {
            printf("Wrong password!\n");
        } else if (res == -2) {
            printf("Username does not exist!\n");
```

Results/output



5.Conclusion

This project helped me understand how real-world management systems work internally.

By implementing login access, file handling, and individual user roles, I learned how to build a structured program in C.
Even though it is simple, it covers all basic functionalities of a student management system and can be extended in the future.

6.References

1. College classroom notes and lecture materials on File Handling, Structures, Functions, and Modular Programming in C.

2. GeeksforGeeks – C Programming Language Tutorials, available at: https://www.geeksforgeeks.org/c-programming-language/

3. TutorialsPoint – C Programming, https://www.tutorialspoint.com/cprogramming/