**PROJECT FINAL REPORT**

# The University of Azad Jammu and Kashmir, Muzaffarabad.

## OBJECT ORIENTED PROGRAMMING

| | |
|---|---|
| **Students Name:** | Feeza Kulsoom<br>Hasnain Malik<br>Syed Abdulrehman Naseem |
| **Roll No:** | 2024-SE-03<br>2024-SE-37<br>2024-SE-18 |
| **Instructor:** | Engr. Awais Rathore |
| **Project Title:** | **Smart Campus Management System** |
| **Report:** | Final Report |

# SMART CAMPUS MANAGEMENT SYSTEM

### Final Project Report

Submitted to: Engr. Awais Rathore
Course: Object Oriented Programming
Department of Software Engineering
The University of Azad Jammu and Kashmir, Muzaffarabad

Submitted by:
Feeza Kulsoom (2024-SE-03)
Hasnain Malik (2024-SE-37)
Syed Abdulrehman Naseem (2024-SE-18)

Date: September 27, 2025

## Abstract / Executive Summary

The Smart Campus Management System is a completed C++ console-based application that streamlines core academic and administrative processes. Unlike traditional manual methods, the implemented system provides secure, role-based access to students, faculty, and administrators. It enables efficient management of courses, attendance, fees, and digital identification. This final report details the design, implementation, testing, and evaluation of the system, highlighting its use of Object-Oriented Programming (OOP) concepts, modular architecture, and file handling for persistent data storage.

## Project Objectives

- Develop an integrated digital platform for campus management

- Provide role-based access and authentication

- Automate attendance, financial, and enrollment operations

- Maintain secure, persistent data storage

- Apply OOP principles to solve real-world problems

**PROJECT FINAL REPORT**

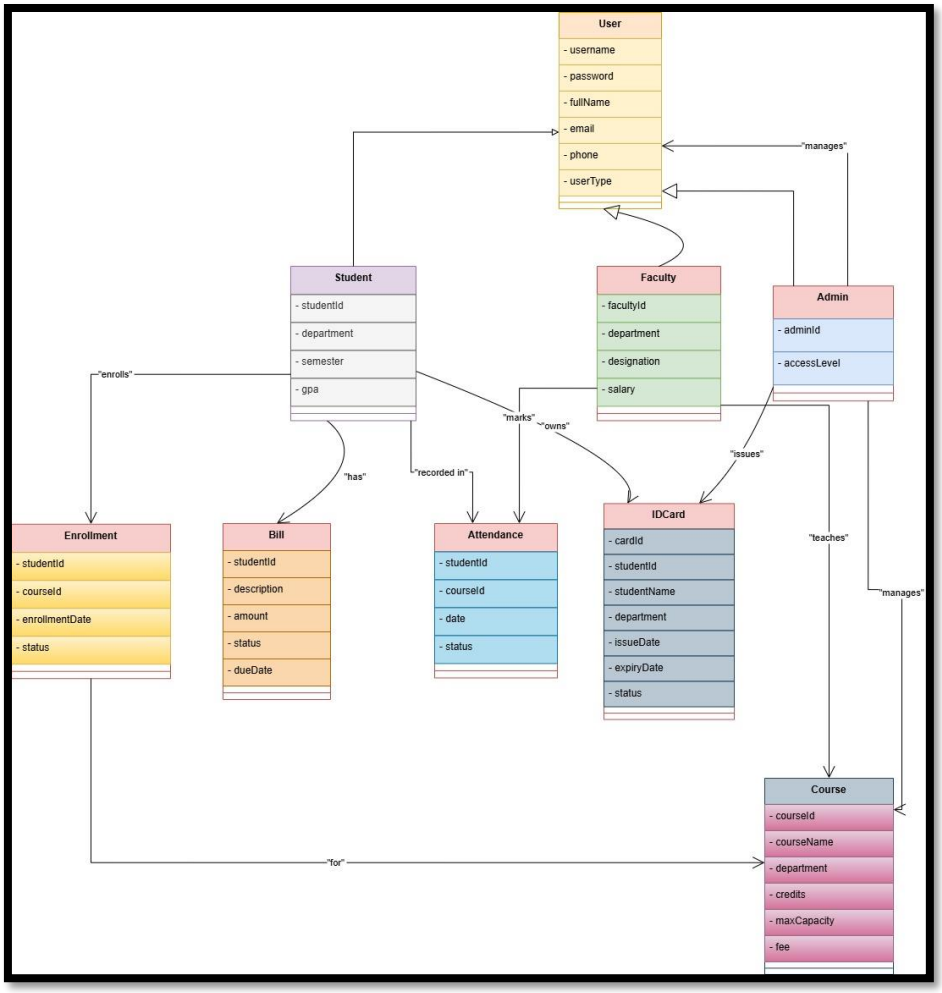
## Implemented Features

The following features were successfully implemented in the system:
• Role-based login system for students, faculty, and administrators
• Course management: add, enroll, search, and view courses
• Attendance marking and reporting by faculty
• Fee calculation, payment, and reporting for students
• Digital ID card generation with validity tracking
• File handling for persistent storage of all records

## System Design & UML Diagram

The Smart Campus Management System was modeled using UML diagram. This diagram represent the object relationships, user interactions, and overall workflow.

**PROJECT FINAL REPORT**




## Implementation

The system was implemented in C++ using Dev-C++ IDE. Object-oriented programming concepts such as encapsulation, inheritance, and polymorphism were applied to design classes for users, courses, billing, and attendance. File handling was used for persistent storage of users, courses, enrollments, and reports. The console interface was enhanced with ASCII formatting and simple colors for usability.

## Testing & Results

The application was tested through the following scenarios:
• Student login, enrollment in courses, and fee payment
• Faculty login, marking attendance, and generating reports
• Administrator login, managing users and courses

All test cases executed successfully, and the system consistently produced correct outputs. Console outputs confirmed accurate data storage, retrieval, and reporting.

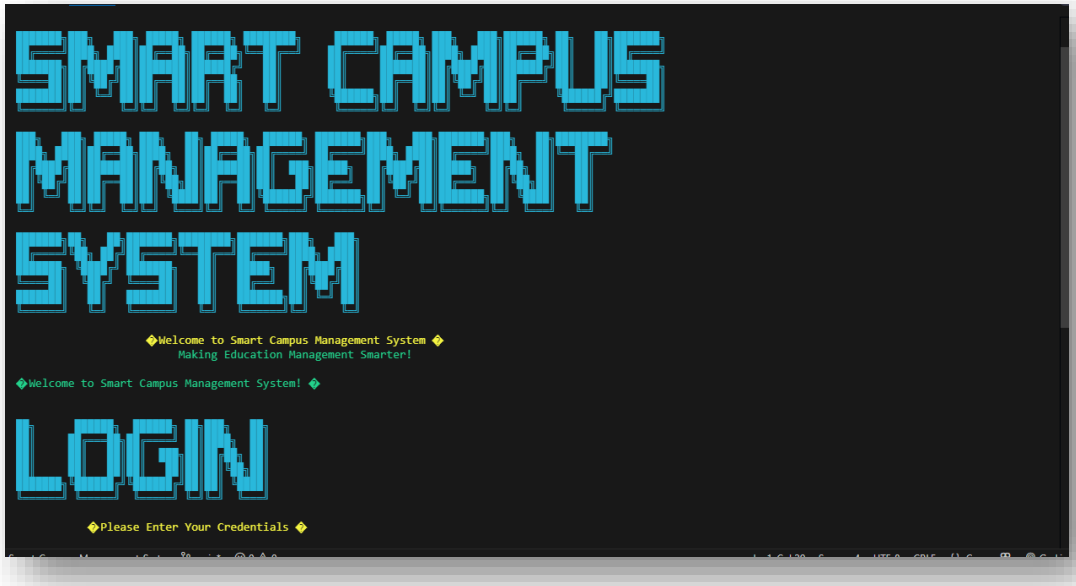



**Figure 1: Login Portal**

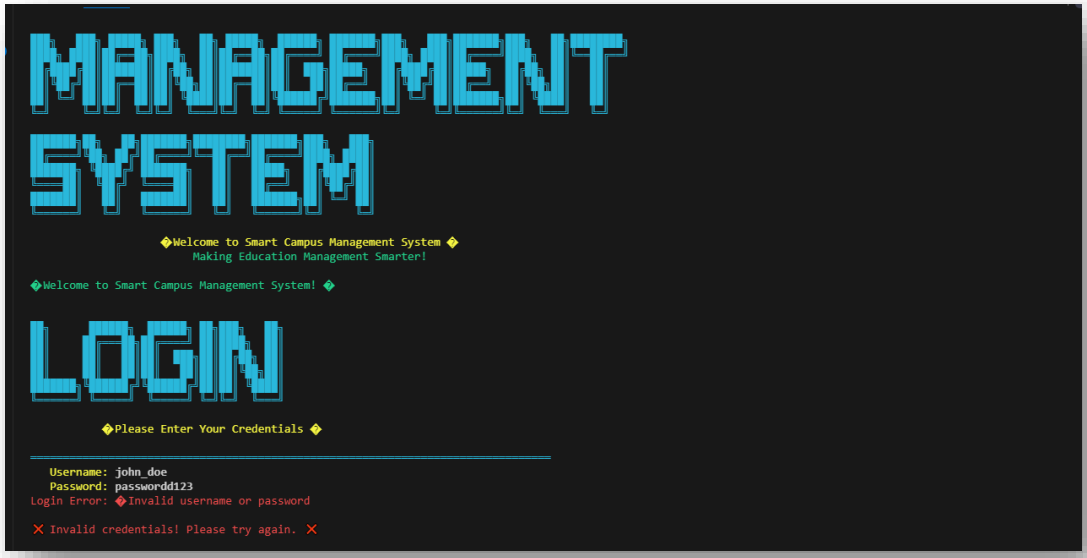



**Figure 2: Handling Invalidity**

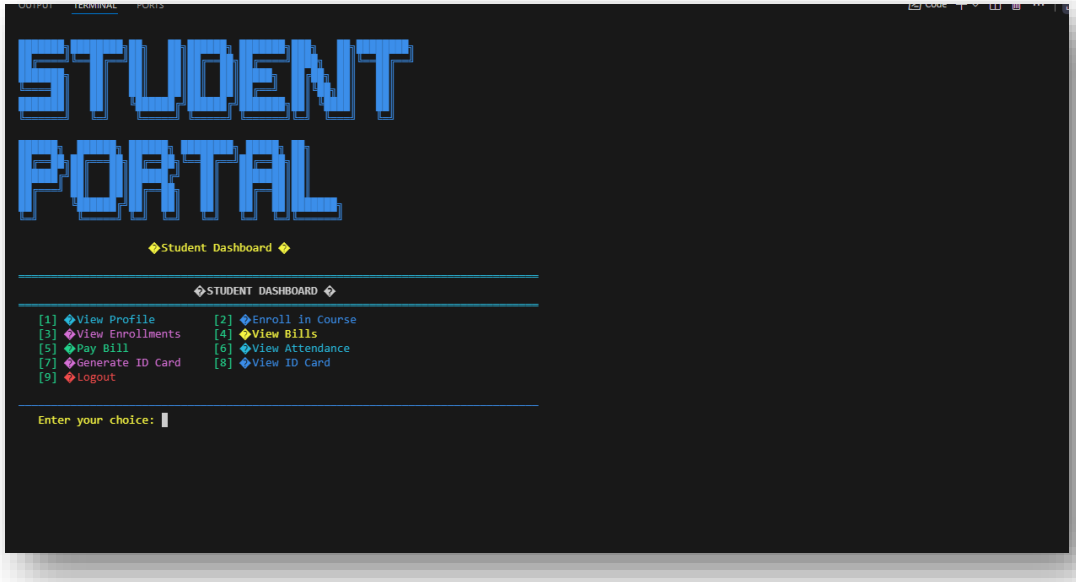



**Figure 3: Student Dashboard**

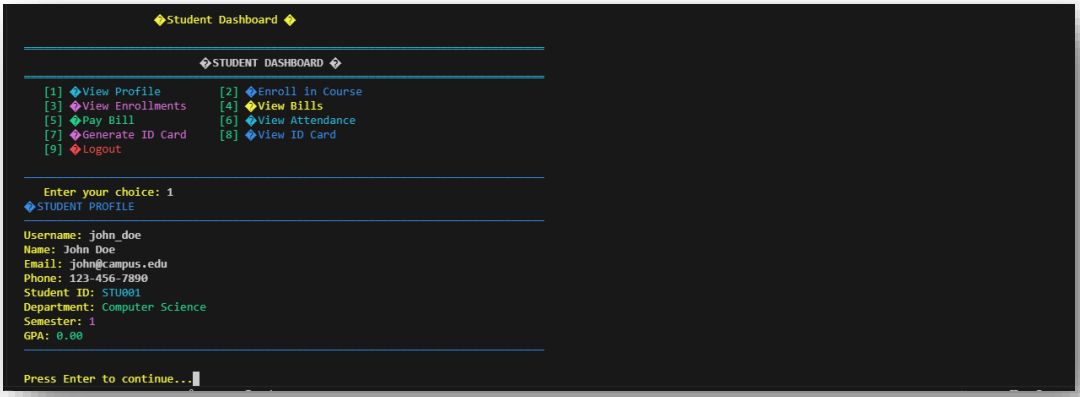



**Figure 4:Student Profile Generation**

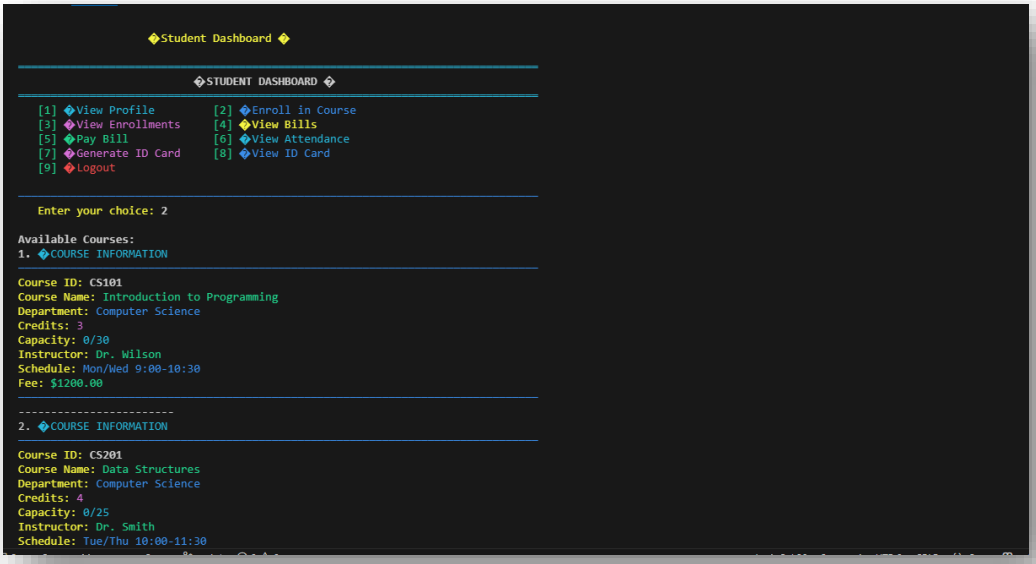



**Figure 5: Showing Available Courses**

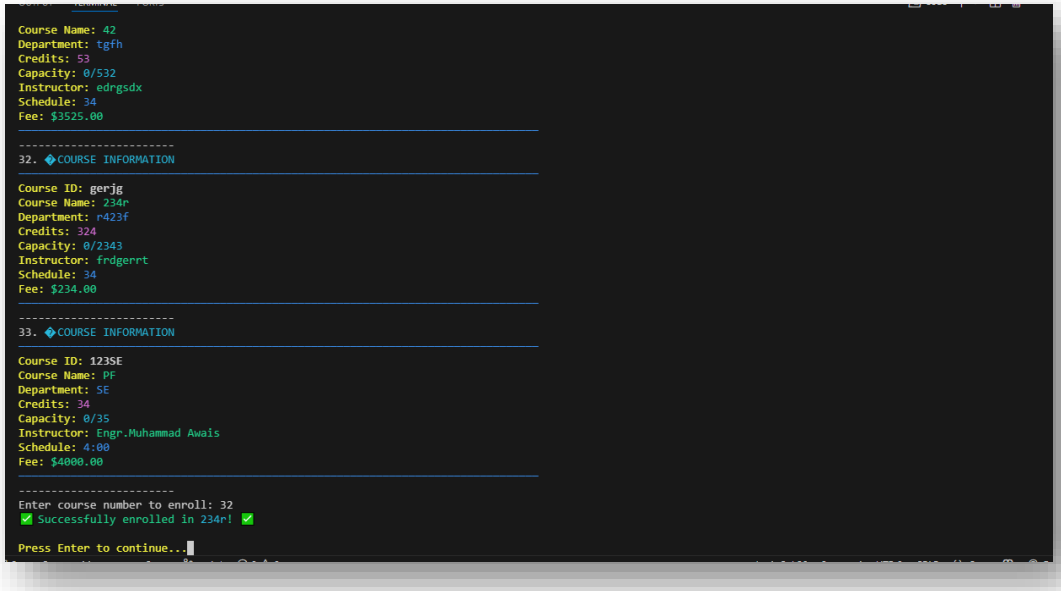



**Figure 6:Enroll In Course**

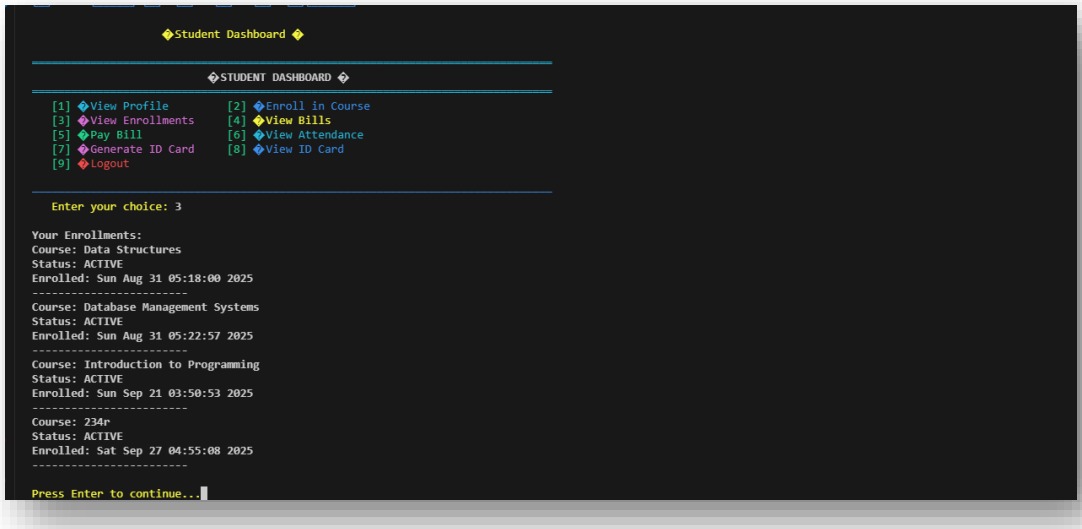



**Figure 7: Showing Student Enrollment**

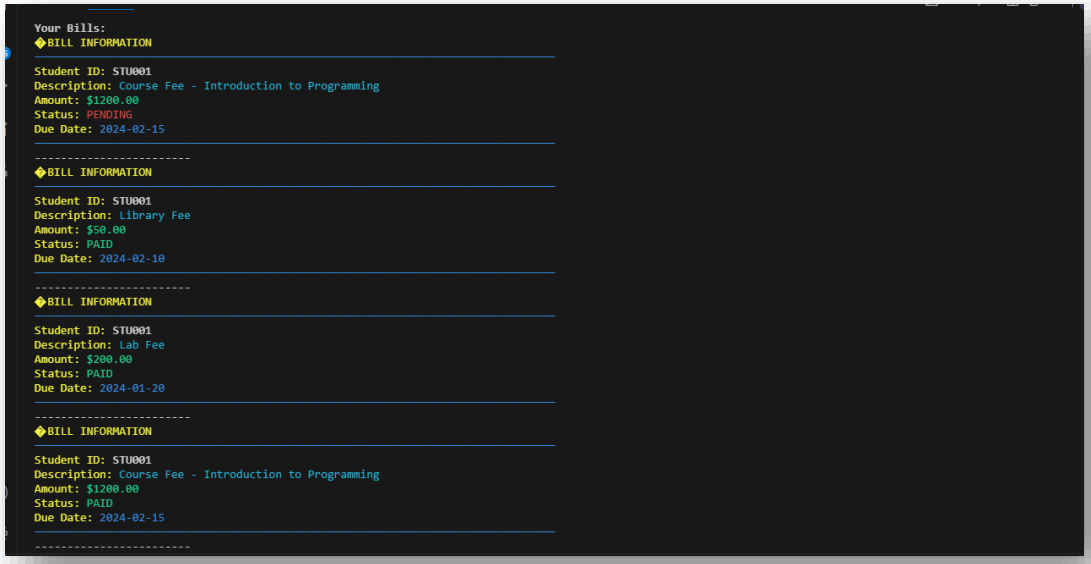



**Figure 8: View Bills**

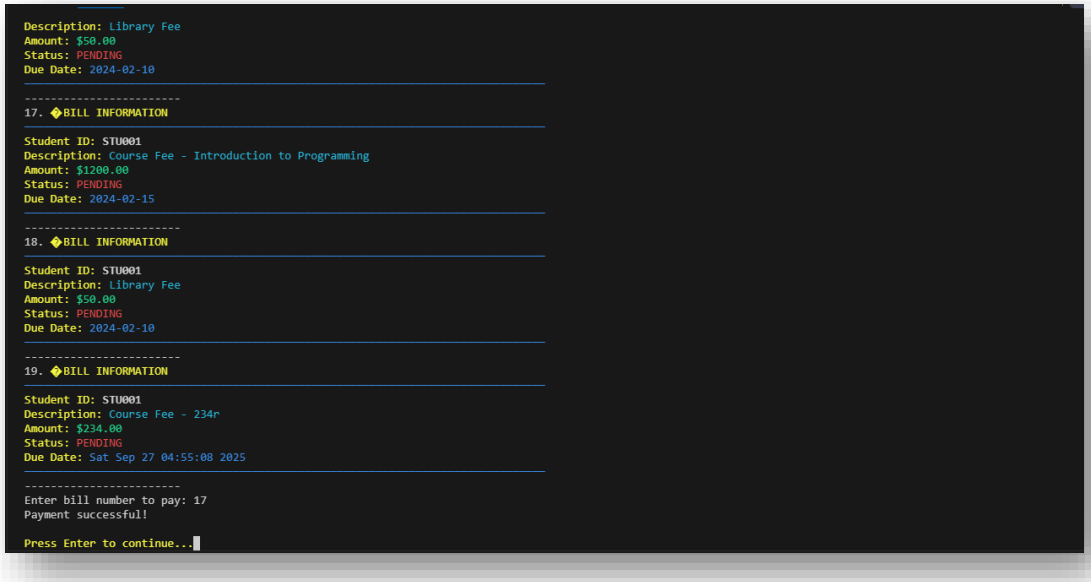



**Figure 9: Pay Bills**

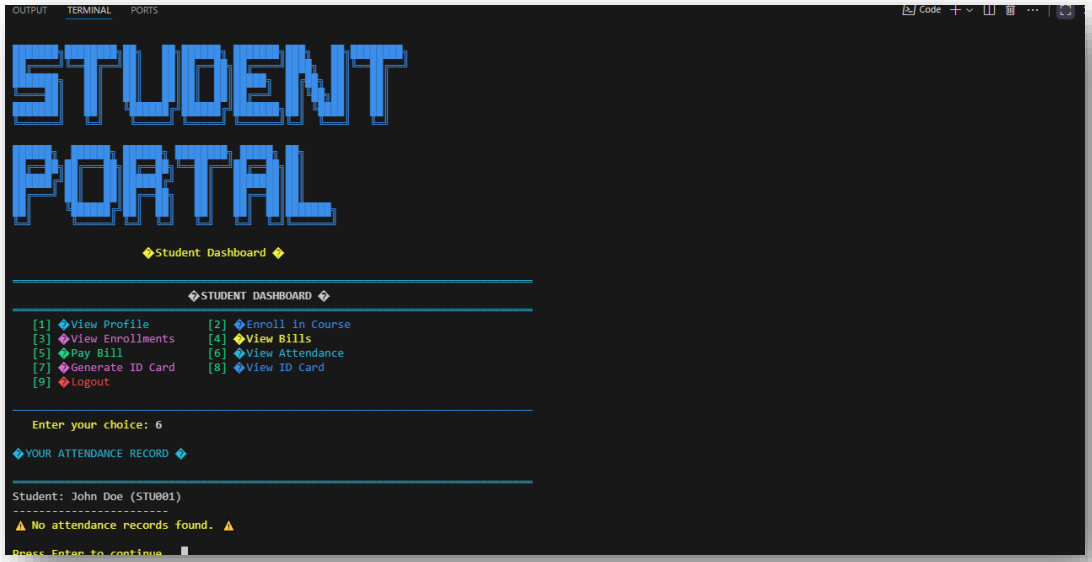



**Figure 10:View Attendance**

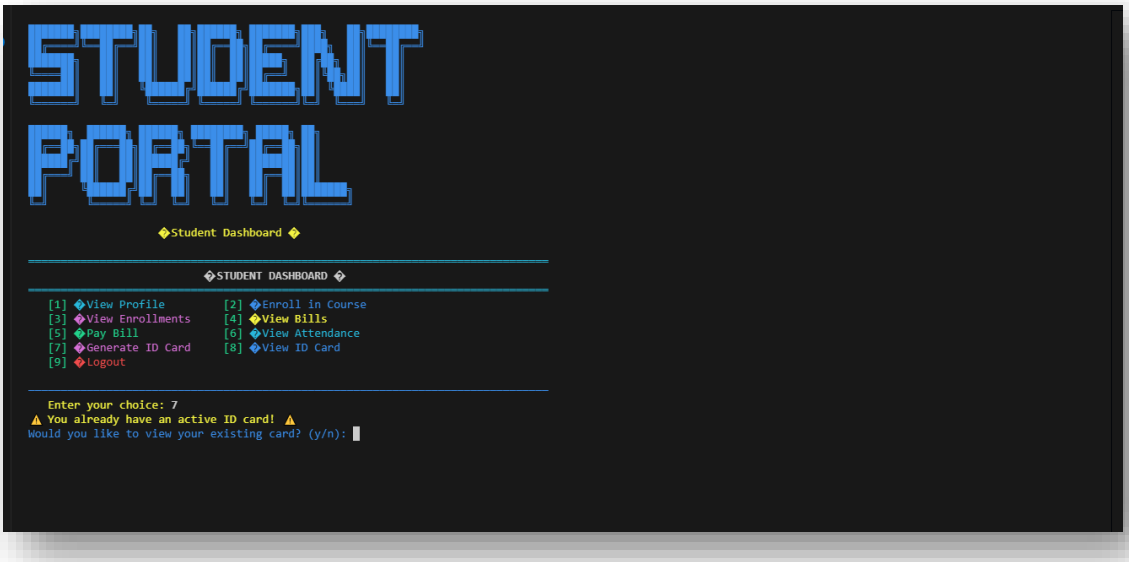


**Figure 11: Generate ID Card**

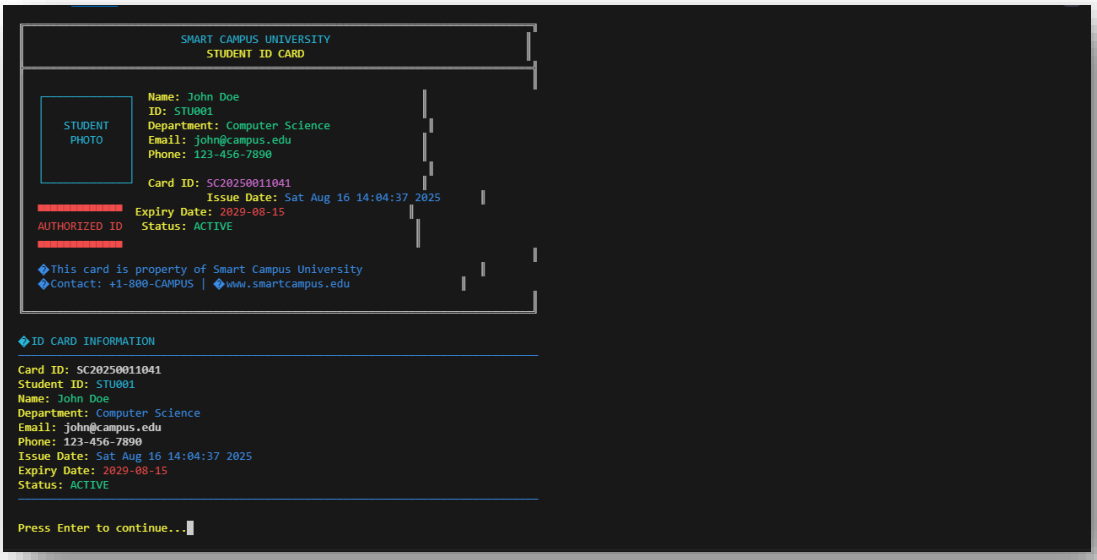



**Figure 12: View ID Card**

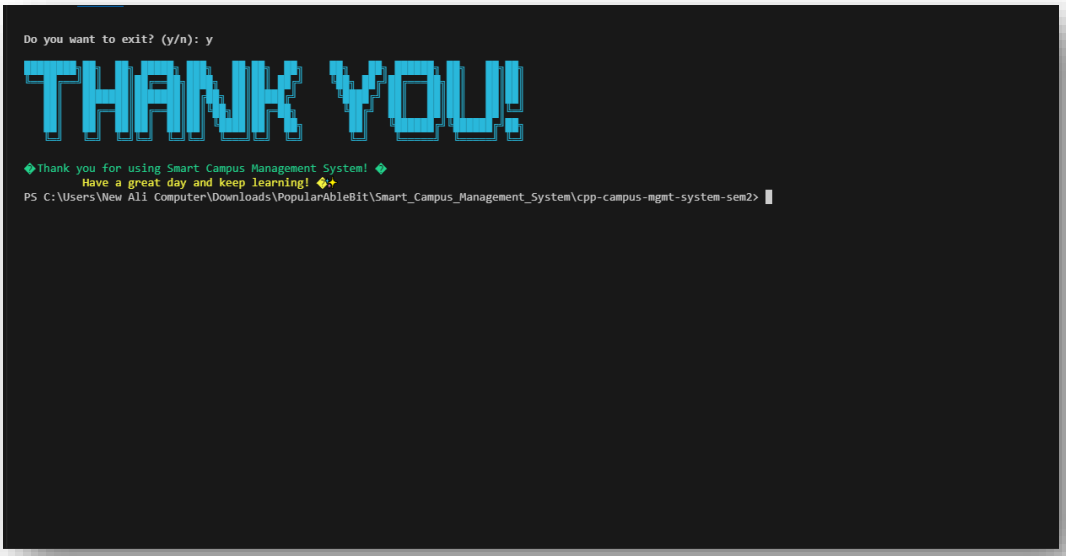



**Figure 13: Student Portal Close**


**Figure 14: Handling Invalid Teacher Credentials**

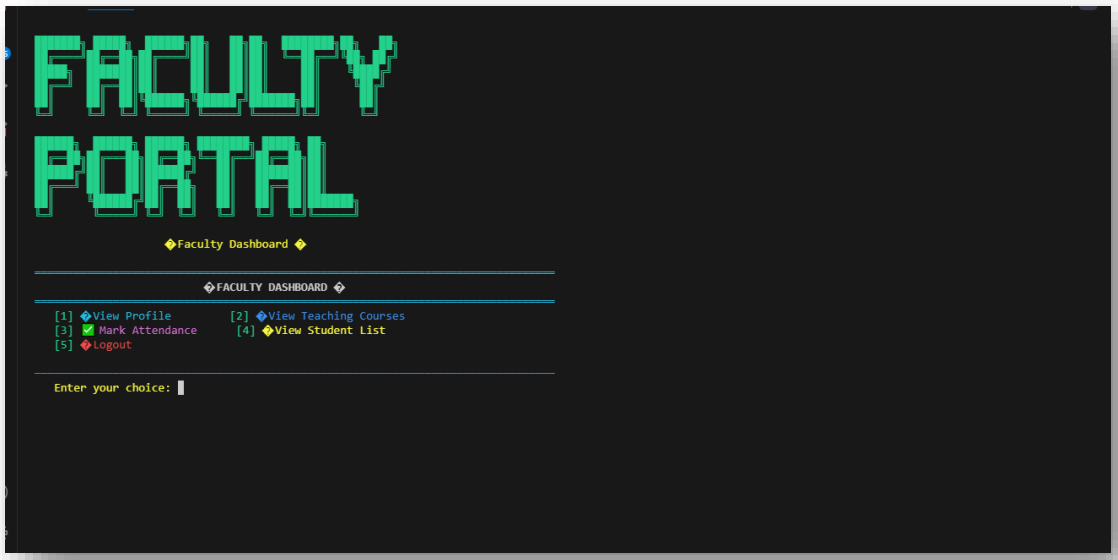



**Figure 15: Faculty Portal open on Correct Credentials**

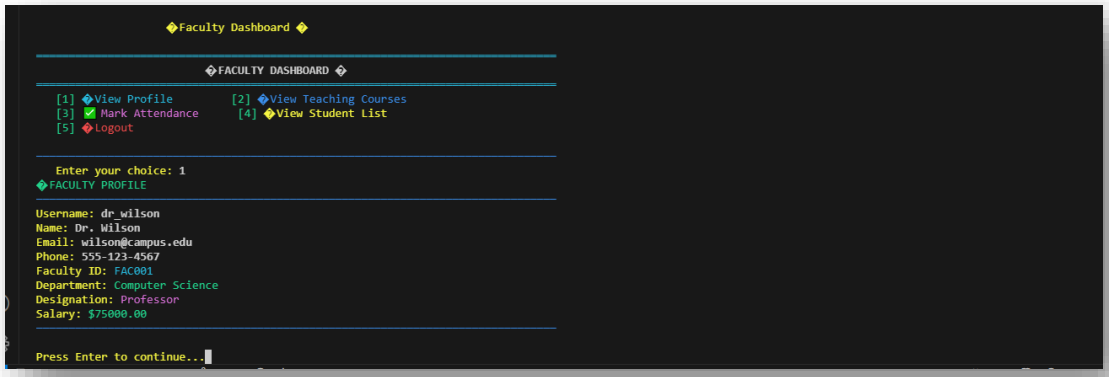



**Figure 16:View Faculty Profile**

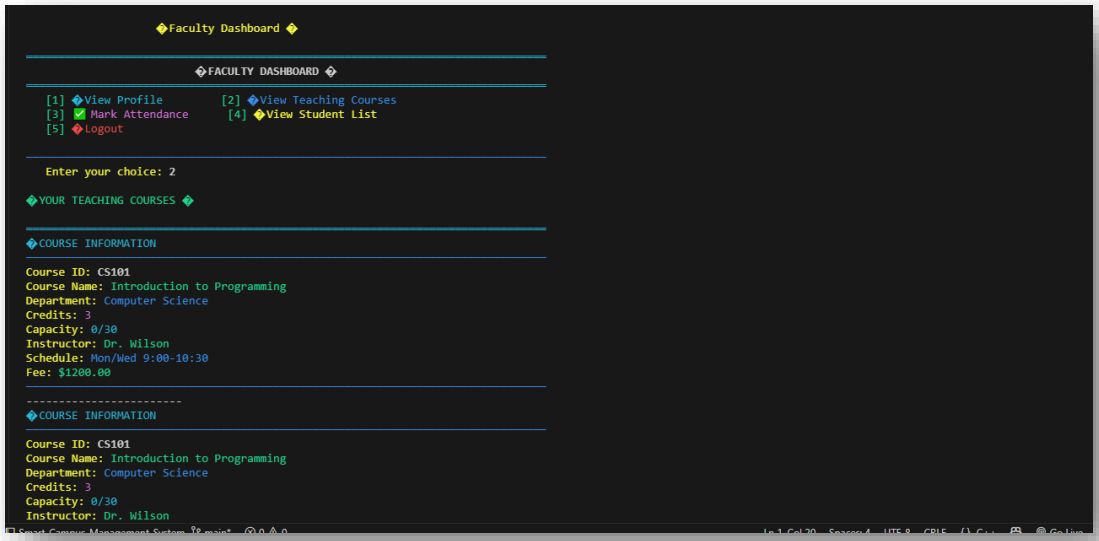



**Figure 17: View Teaching Courses**

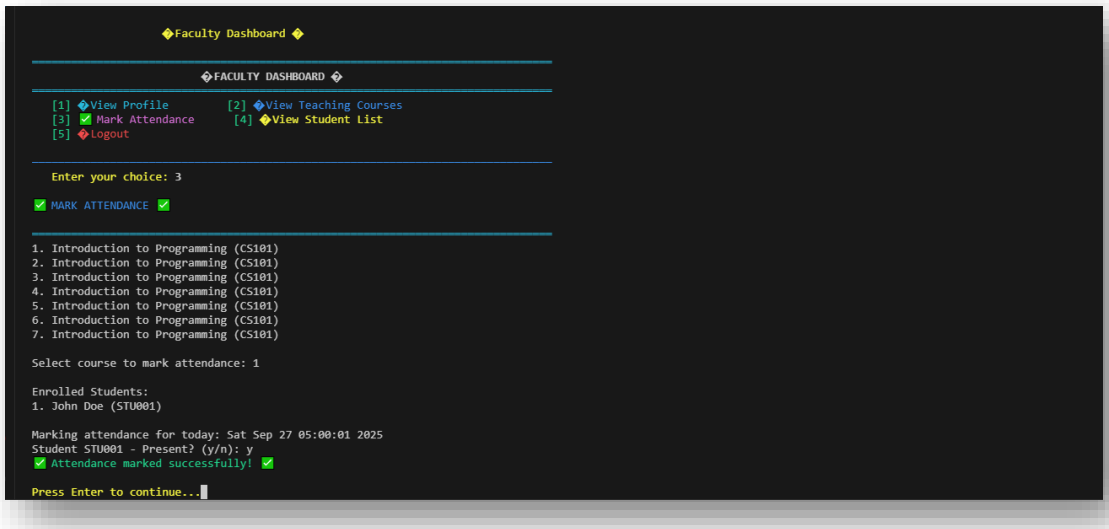



**Figure 18: Mark Attendance**

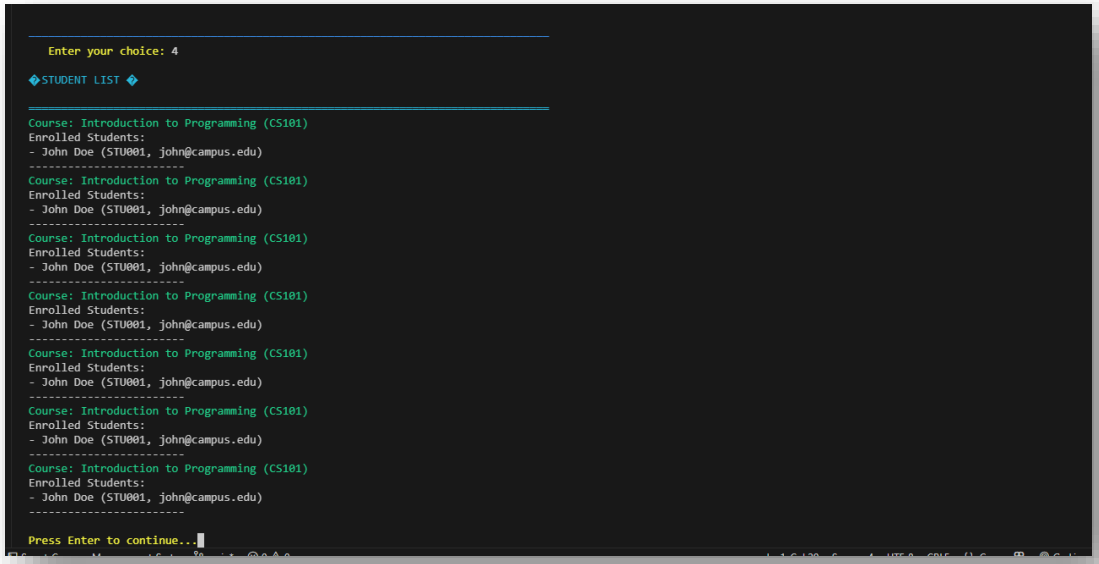



**Figure 19: View Student List**

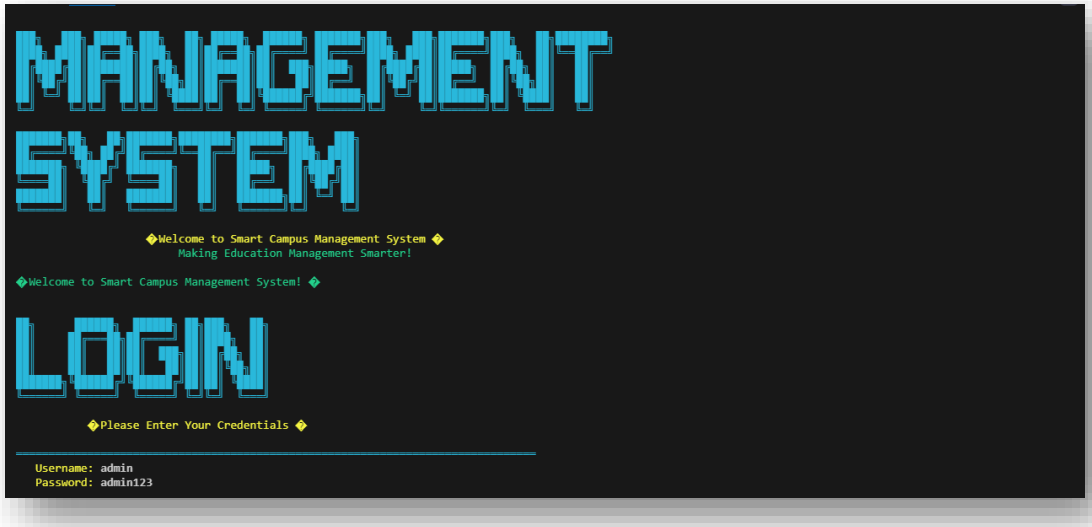



**Figure 20: Opening Admin Portal**

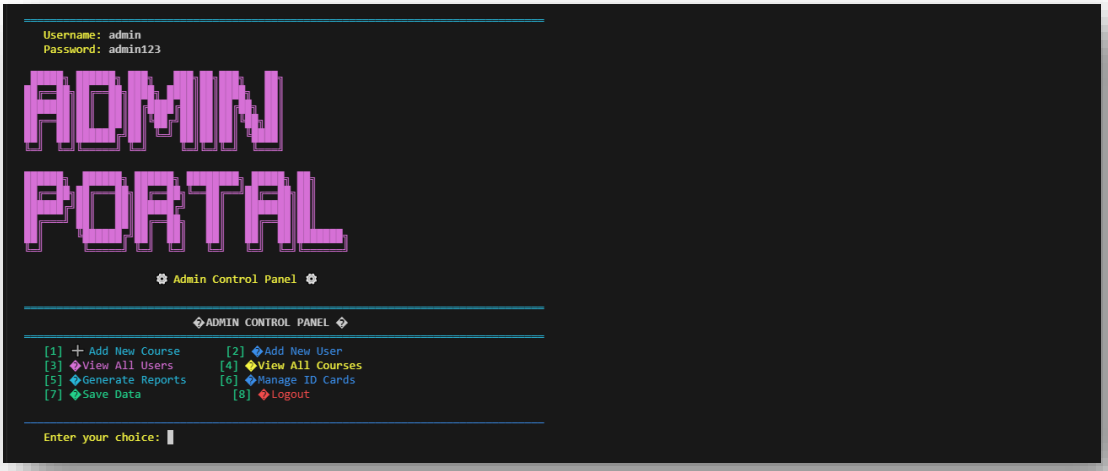



**Figure 21: Admin Portal Opened**

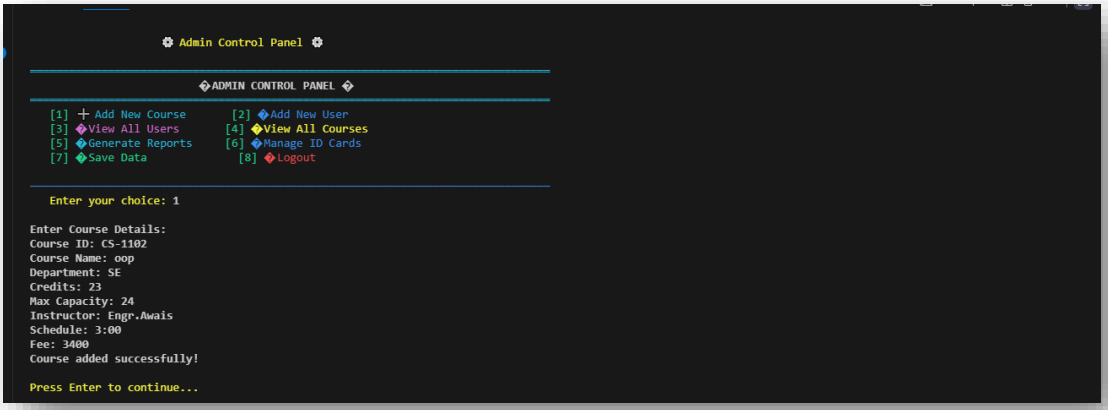



**Figure 21: Admin Adding New Course**

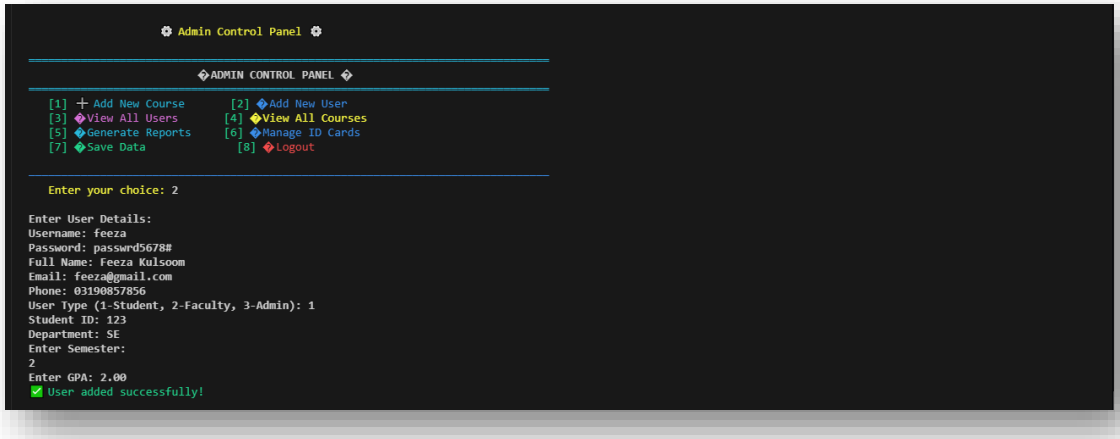



**Figure 22: Admin Add New User**

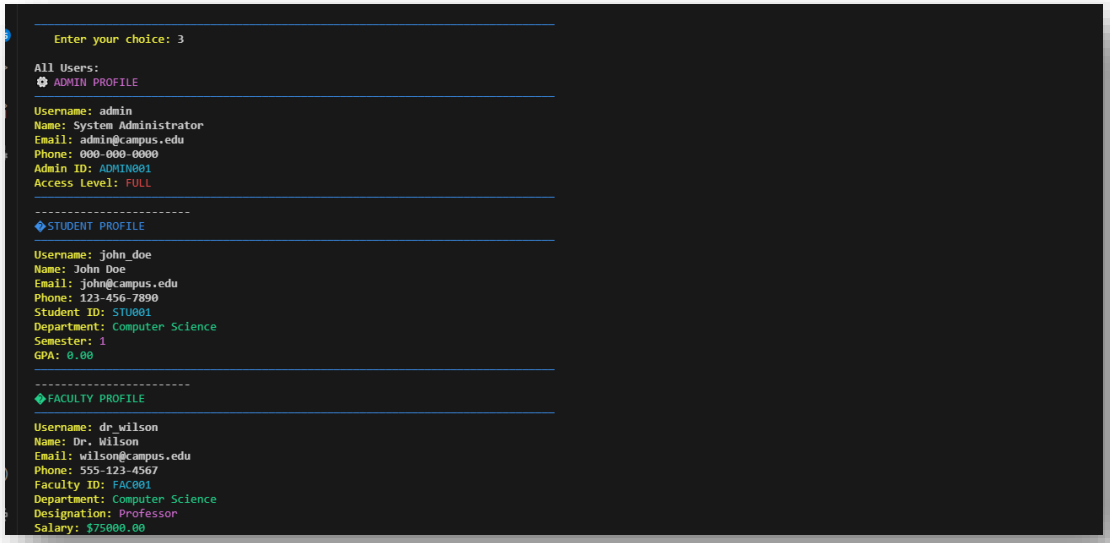



**Figure 23: Admin View All Users**

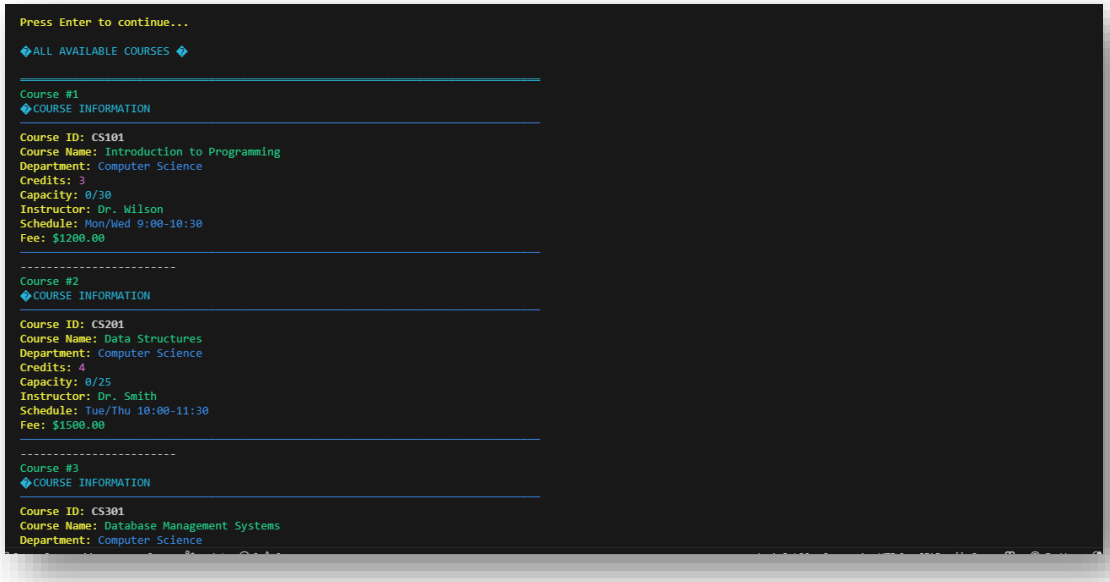



**Figure 24: Admin View All Available Courses**

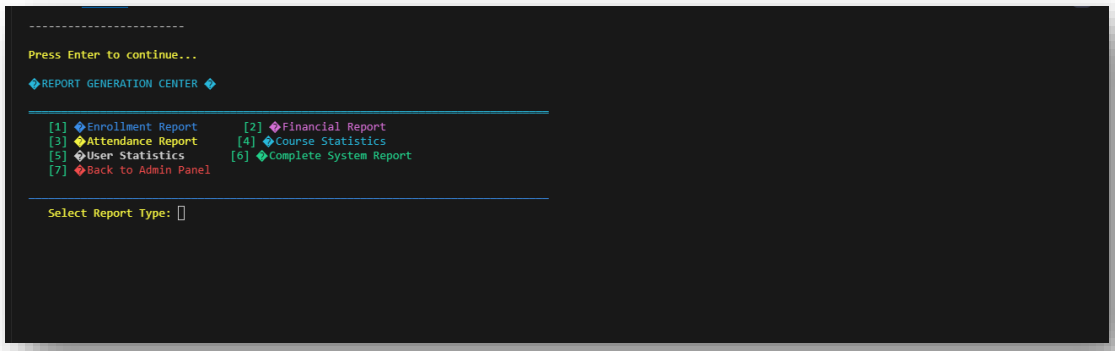



**Figure 25: Admin Generating Any Type Of Report**

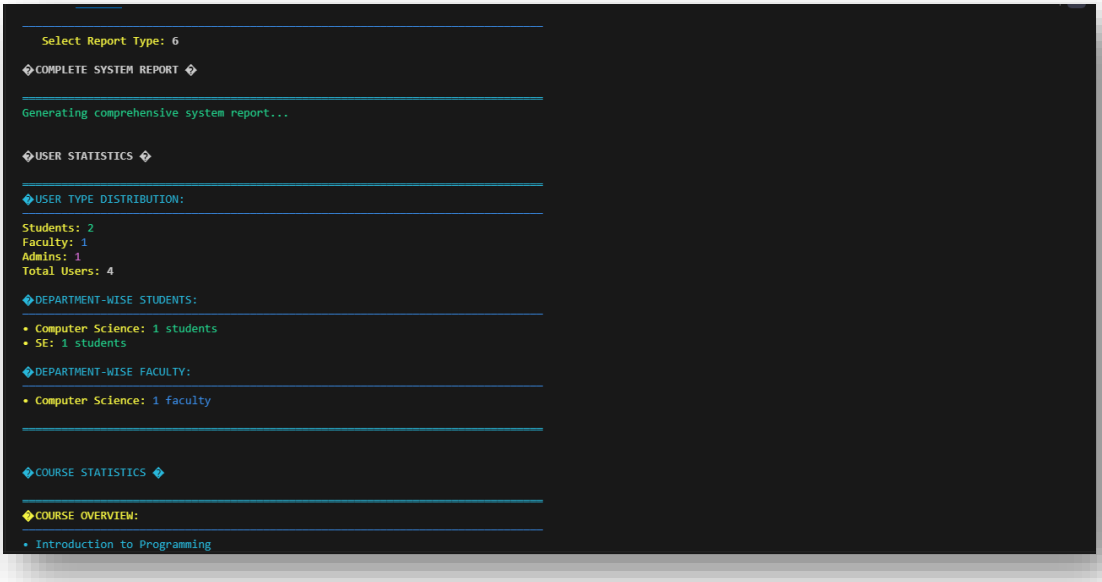



**Figure 26: Generating System & User Statistical Report**

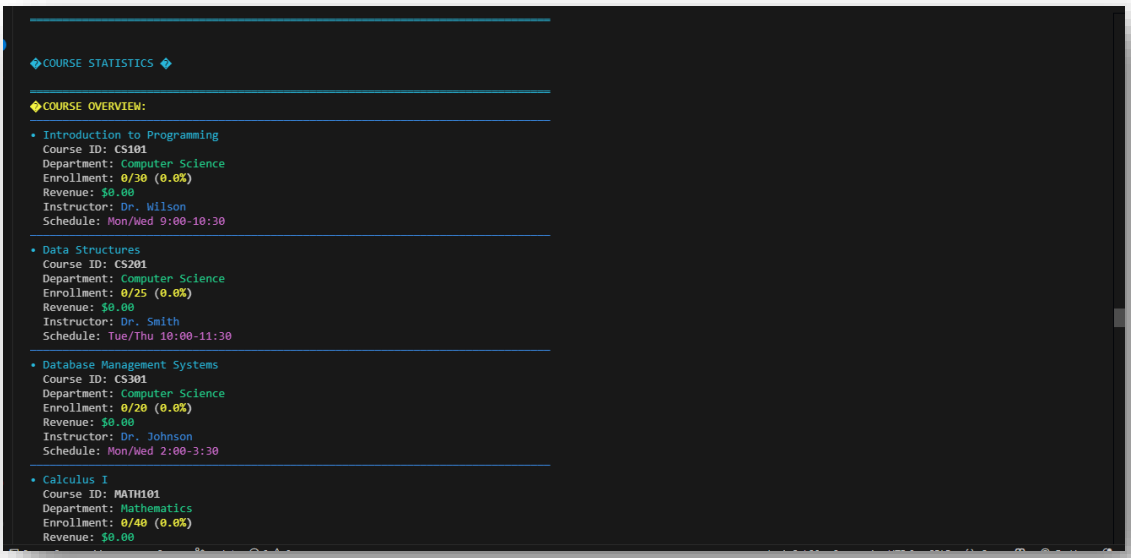



**Figure 27: Generating Course Statistic Report**

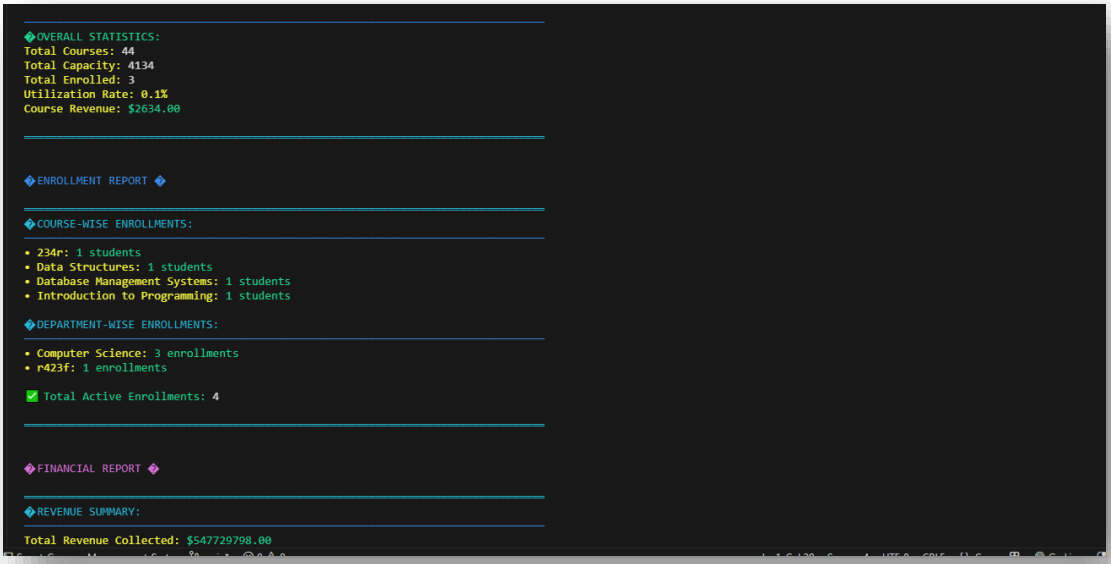



**Figure 28: Generating Financial and Enrollment Report**

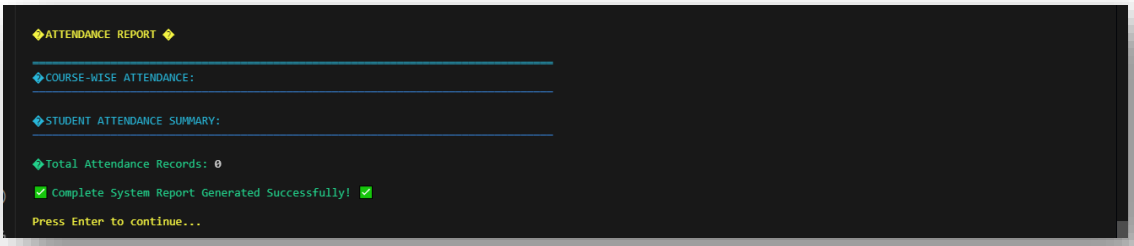



**Figure 29: Generating Attendance Report**

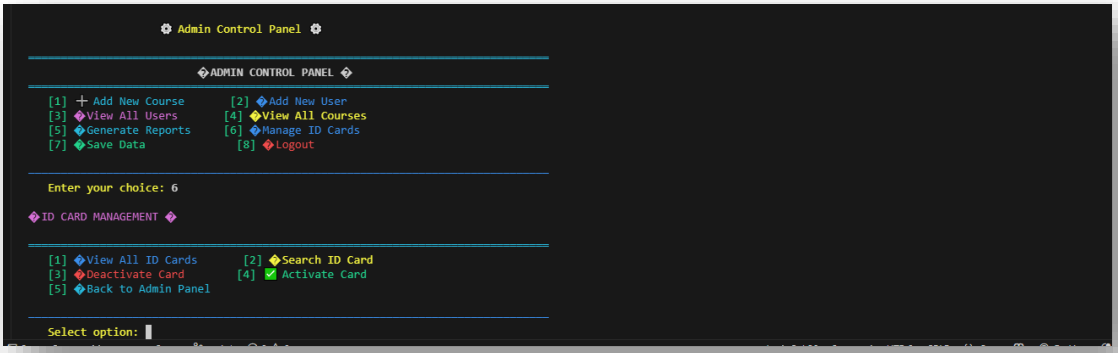



**Figure 30: ID Card Management By Admin (Can Select any option)**

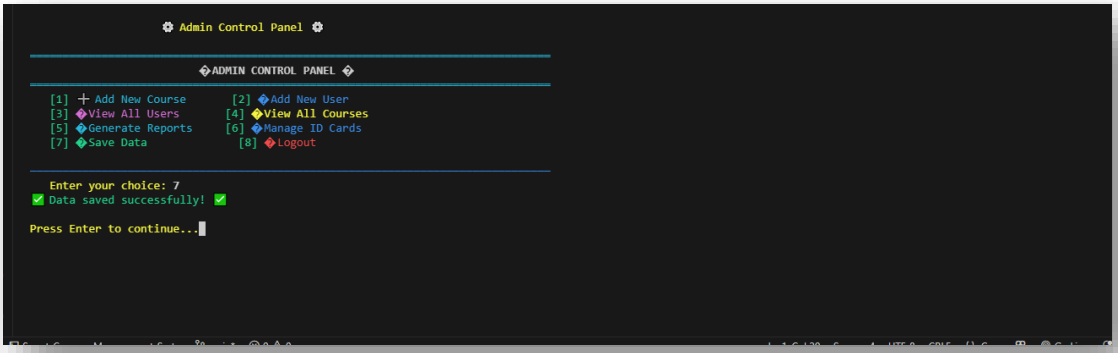



**Figure 31: Admin Saving All Data At The End**

## Discussion

The development process demonstrated the importance of OOP concepts in structuring software. Encapsulation ensured modularity, inheritance allowed reusability, and polymorphism enabled flexibility. The major challenge was ensuring data consistency across multiple files, which was addressed by careful file handling design. The system effectively balances functionality with simplicity.

## Future Work

While the current version of the system fulfills its objectives, several future improvements are possible:

- Transition from file handling to a relational database system
- Development of a graphical user interface (GUI)
- Web or mobile-based versions for remote access
- Enhanced analytics and dashboards for administrators

## Conclusion

The Smart Campus Management System successfully achieved its objectives by automating critical campus operations and demonstrating practical applications of OOP principles. The implementation provided hands-on experience in designing, coding, and testing a real-world project. The project not only improved technical skills but also showcased how OOP and file handling can be applied to solve real institutional challenges.