**1. Abstract**

The Student Management System is a software project designed to streamline and enhance the efficiency of managing student information within an educational institution. This project aims to provide a robust solution for administrators and teachers to seamlessly handle various aspects of student data, including enrollment, academic records and personal details.

The system is designed with a modular approach, consisting of distinct modules to manage specific functionalities. The Student Data Module handles the creation and maintenance of student and teachers profiles, encompassing personal information and enrollment details. The Academic Records Module manages course details, students, teachers and fee payment records. The User Interface Module ensures a user-friendly experience for all students offering intuitive navigation and functionality. The system incorporates input validation mechanisms to ensure data integrity and system reliability.

This System involves the use of C programming language, with a focus on file handling for database management. This project seeks to provide a scalable and maintainable solution, allowing for future updates and enhancements. The Student Management System aims to optimize the process of managing student information, facilitating efficiency, accuracy, and ease of use within educational institutions.

In conclusion, the Student Management System is not only addresses fundamental data management needs but also enhances the overall educational ecosystem by incorporating advanced security and adaptability features. This holistic approach aims to optimize the efficiency, accuracy, and user experience within educational institutions, contributing to the continual improvement of academic processes and outcomes.

**2. Software Specification**

|  |  |
| --- | --- |
| **Operating system** | * Microsoft windows 7 or above * Ubuntu 16.04 LTS or above * MacOS 12 or above |
| **Processor** | * Intel i3 or higher |
| **RAM** | * 4GB or higher |
| **Storage** | * 128 GB or higher |
| **Development Environment** | * Dev C++, Turbo C or Visual Studio |
| **Programming Language** | * C OR C++ |

**3. Scope & Objectives**

The primary objective of the Student Management System is to store and maintain student and teacher’s data efficiently through console-based application. This system aims to achieve operational excellence by providing a centralized platform for managing various activity of school or colleges through this application.

In this system real-time data updation is done into the files. The main goal of the Student Management System is to improve the accuracy and efficiency in the student records management.

The system is developed for Administration Staff, Teacher and Student to maintain records. The Administration Staff is the main user of this system. They can access all the data related to student and teachers and they can manipulate it according to their needs. They can manage course details, teacher’s details, student details along with this an additional fee payment facility is also given in this application. In this System Teacher can add or update student marks which will be directly updated and can be viewed in student result section. The role authentication functionality is given in this system in which the teacher can add marks of students of their own department only. Teacher can’t access the data of student who belongs to another department.

The goal of this system is to increase administrative efficiency by simplifying the data maintaining process.

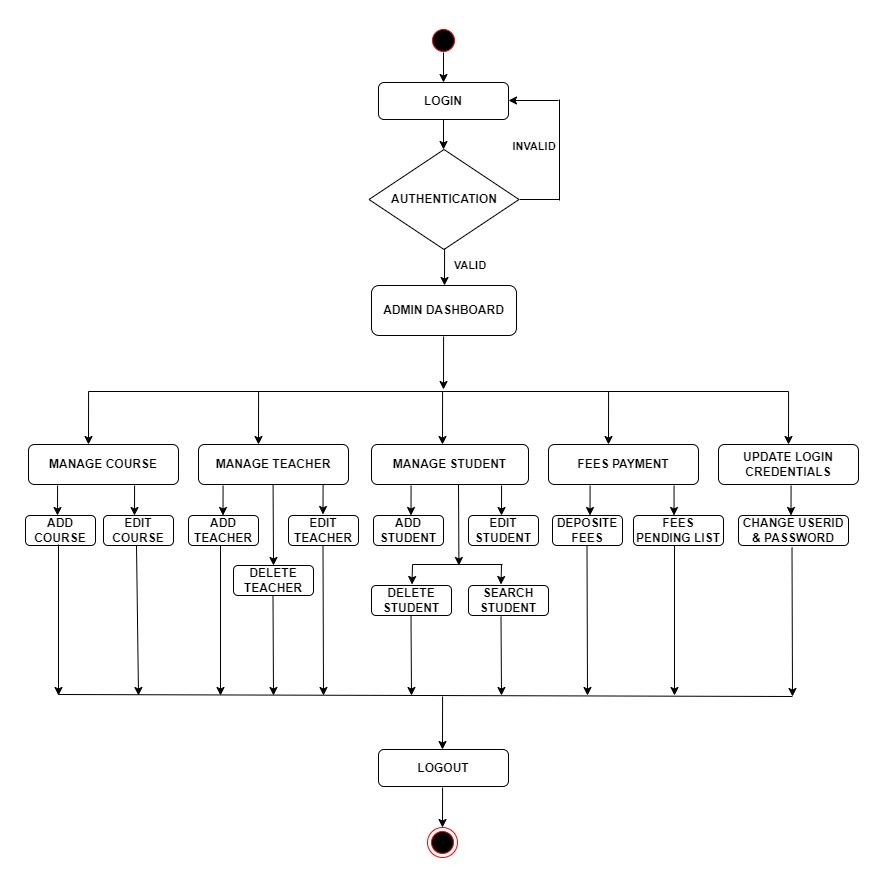
**4. Database Design**

**4.1 Use Case Diagram**

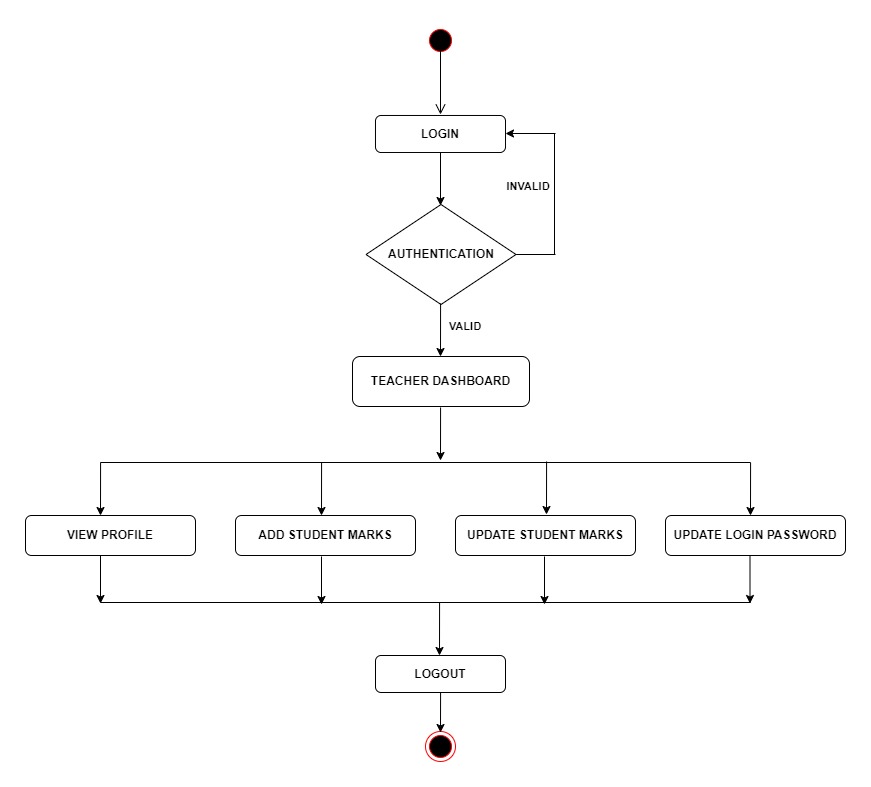
****

**[Fig. 4.1 Use Case Diagram]**

**4.2 Activity Diagram**

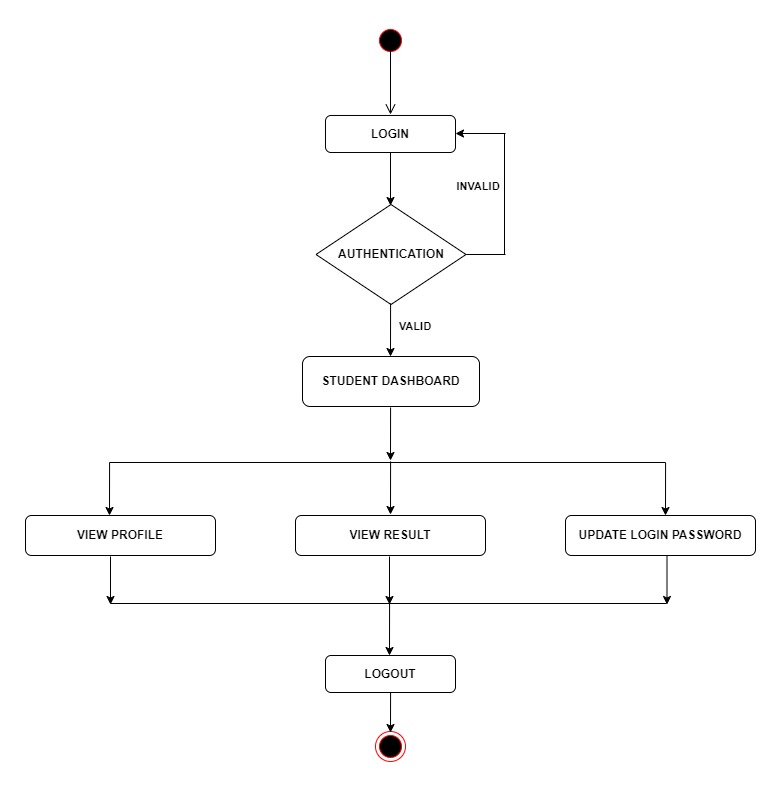
* **ADMIN **

**[Fig. 4.2.1 Admin Activity Diagram]**

* **TEACHER**

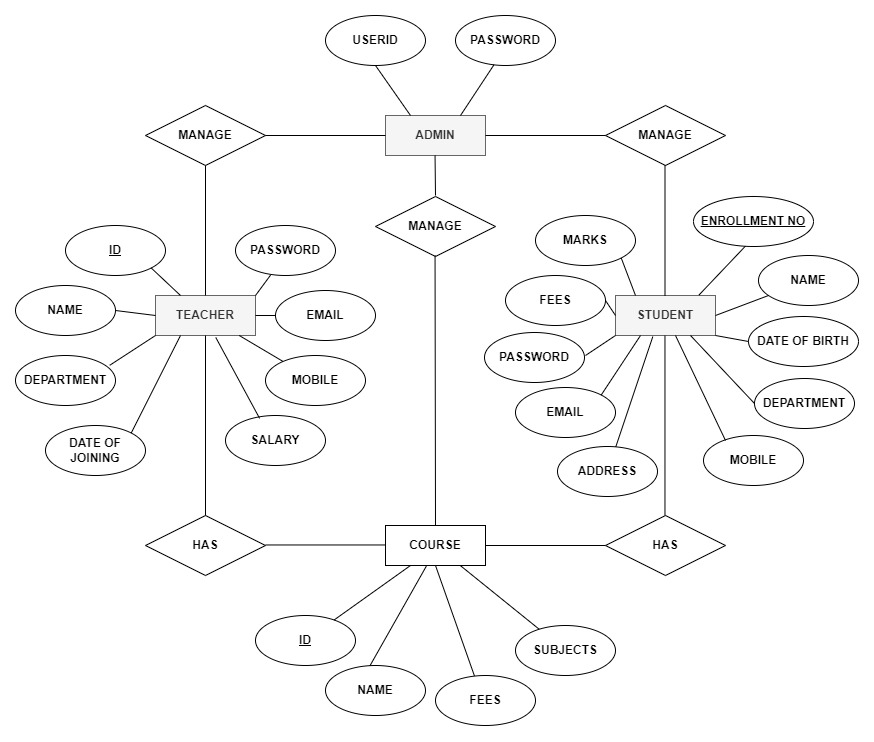
**[Fig. 4.2.2 Teacher Activity Diagram]**

* **STUDENT**

****

**[Fig. 4.2.3 Student Activity Diagram]**

**4.3 ER Diagram**

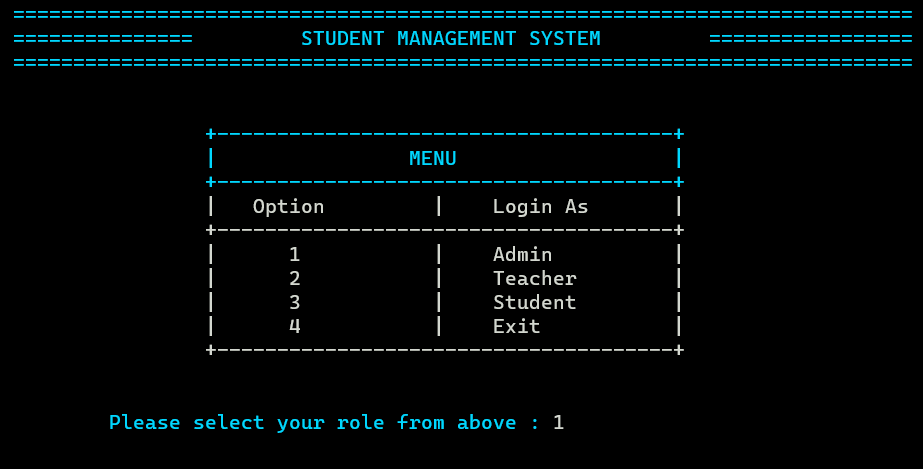
****

**[Fig. 4.3 ER Diagram]**

**5. System Design**

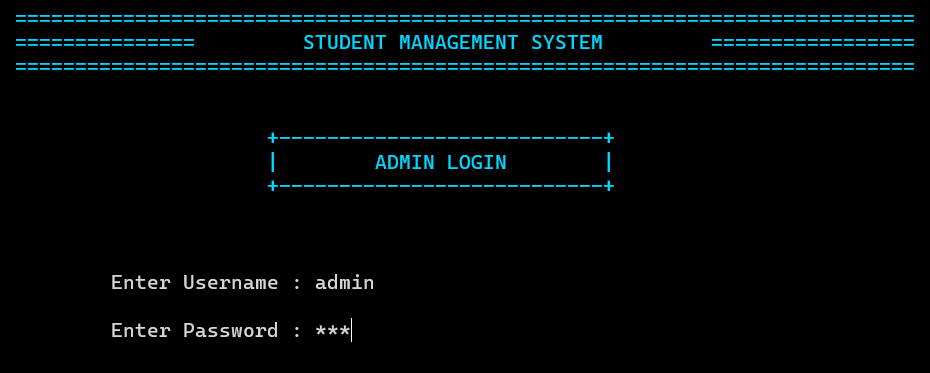
**5.1 Screenshot**

**Welcome Page**

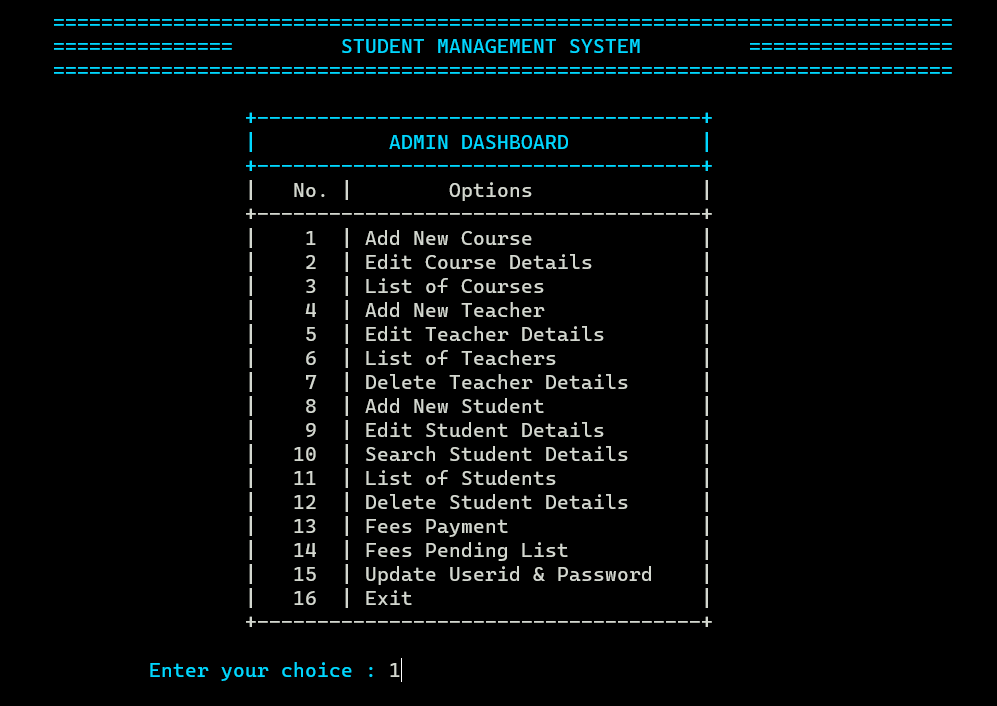


**[Fig. 5.1.1 Welcome Page]**

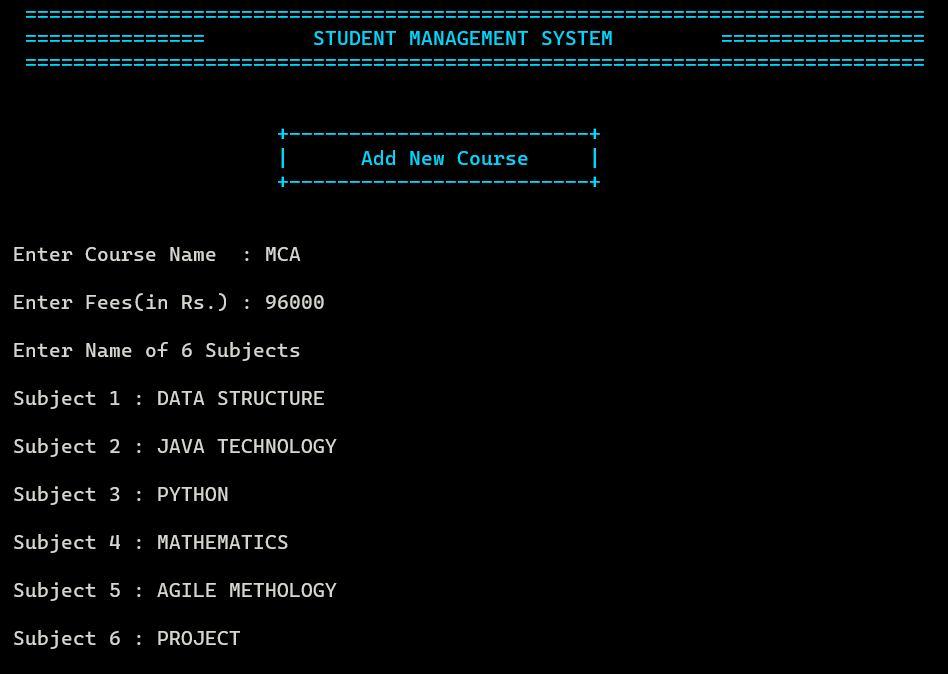
**Admin Login**



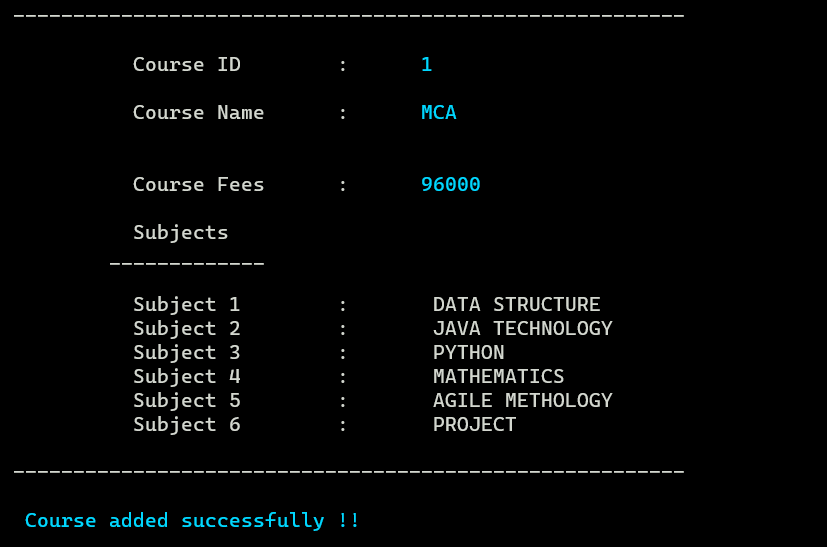
**[Fig. 5.1.2 Admin Login]**



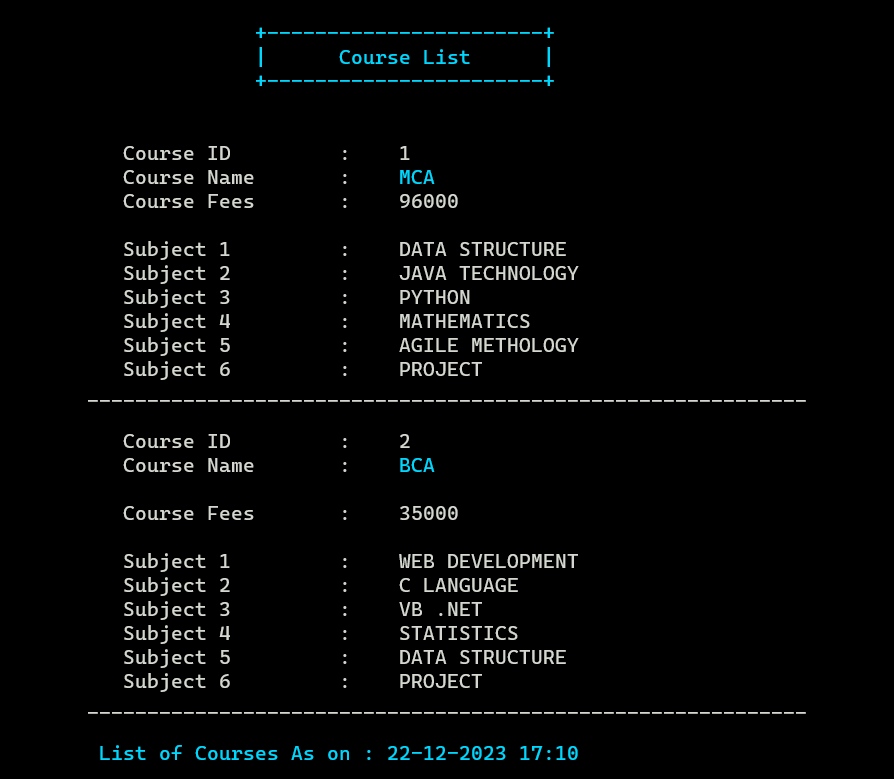
**[Fig. 5.1.3 Admin Dashboard]**



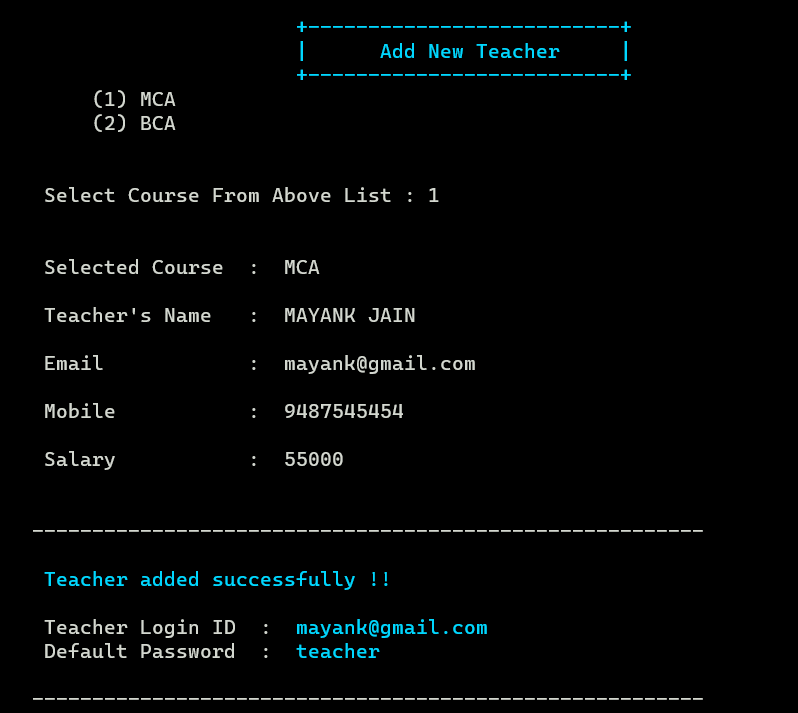
**[Fig. 5.1.4 Add New Course]**



**[Fig. 5.1.5 Course Added]**



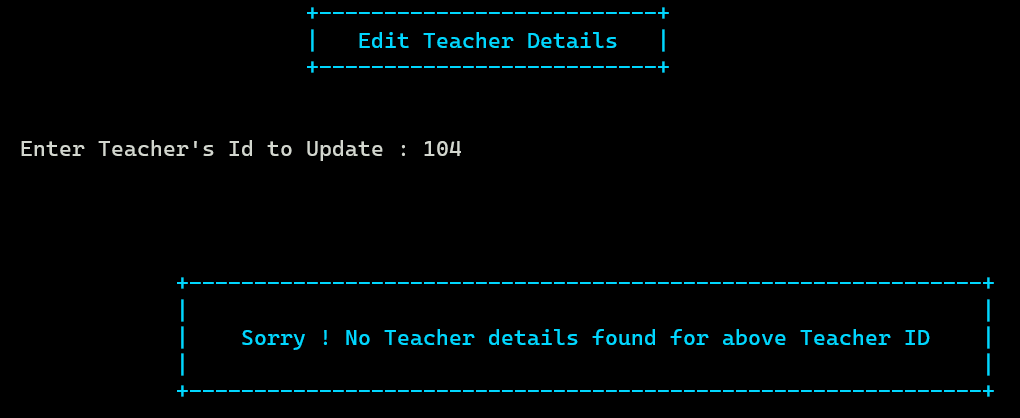
**[Fig. 5.1.6 Course List]**



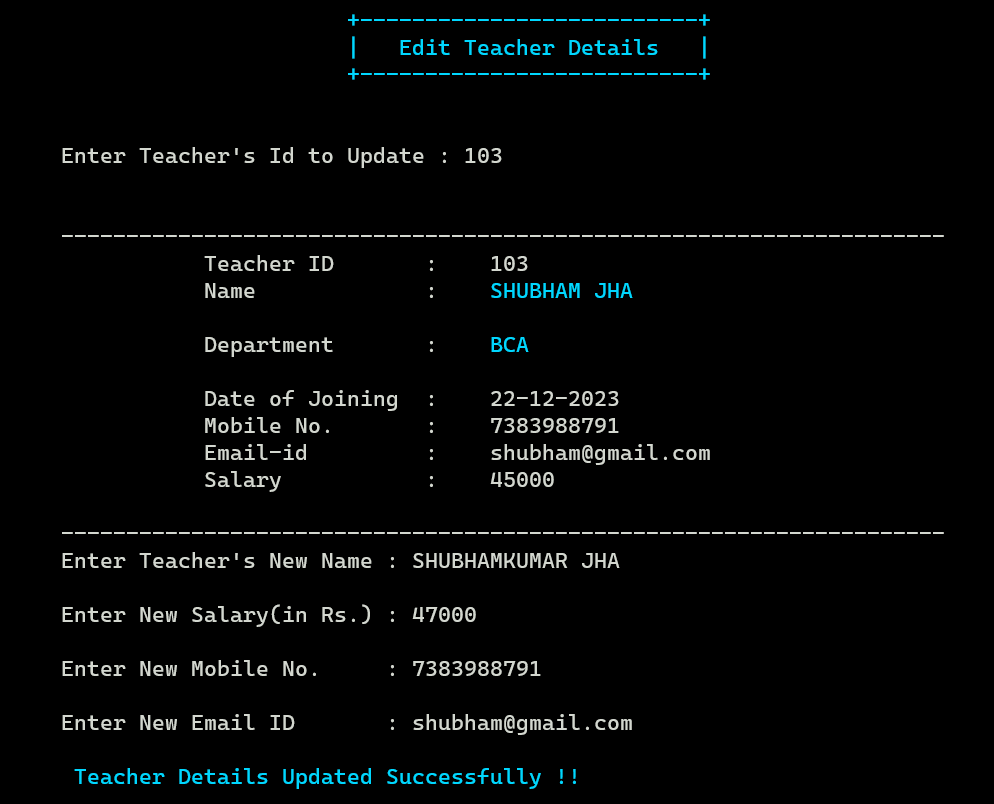
**[Fig. 5.1.7 Add New Teacher]**



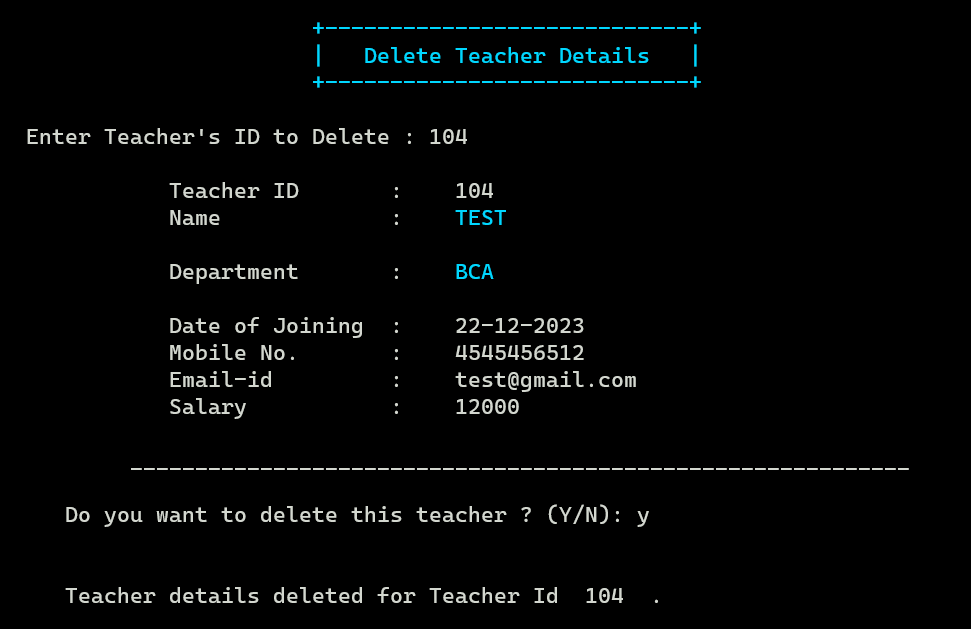
**[Fig. 5.1.8 Teacher’s List]**



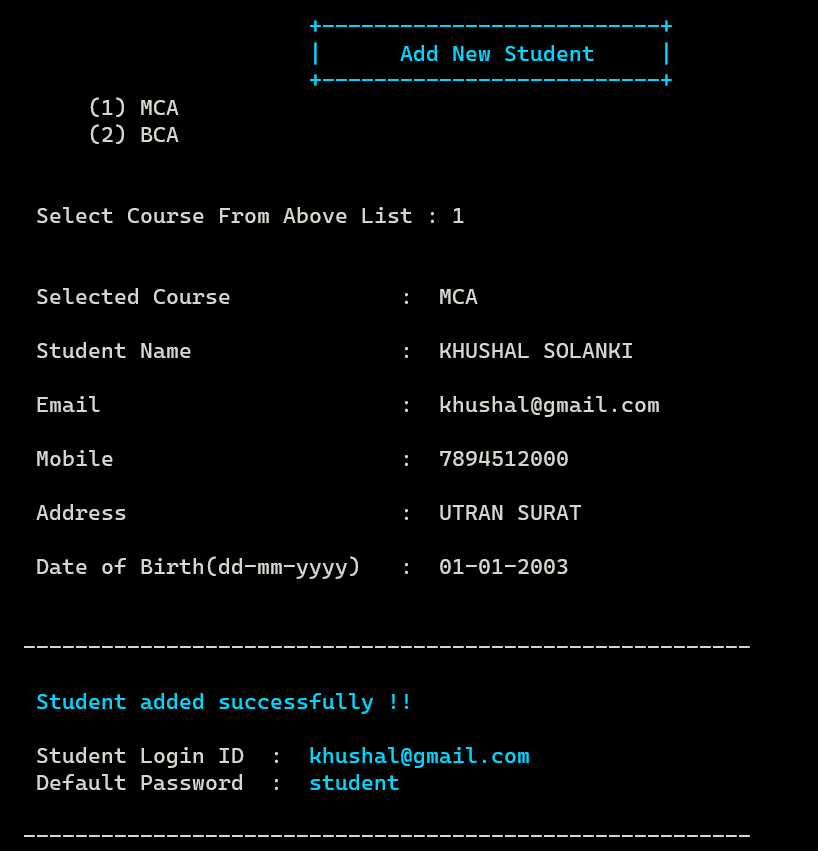
**[Fig. 5.1.9 Edit Teacher Details(Not Found)]**



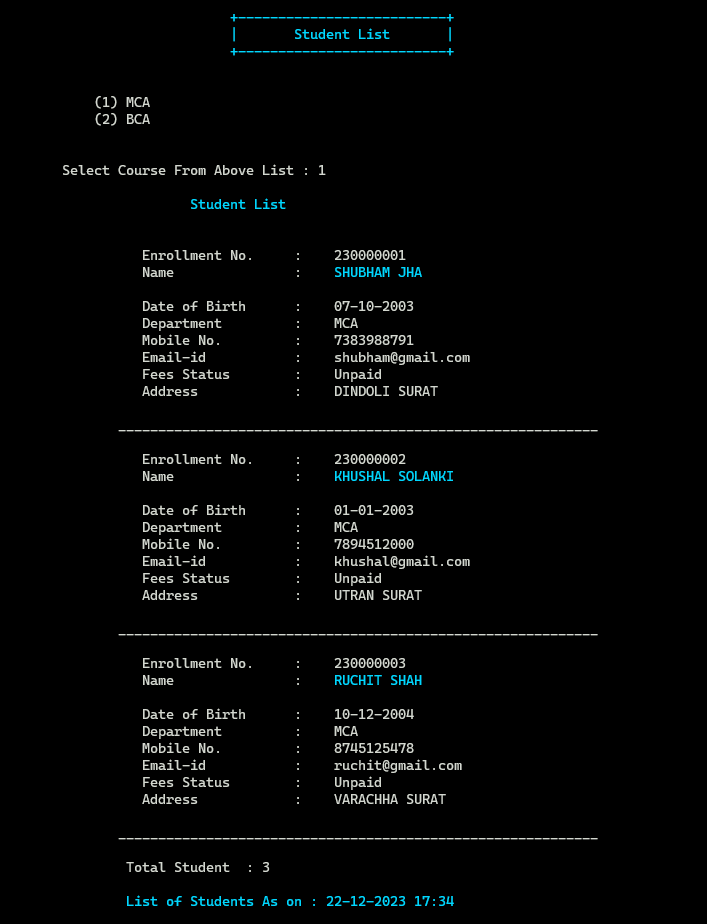
**[Fig. 5.1.10 Edit Teacher Details]**



**[Fig. 5.1.11 Delete Teacher Details]**



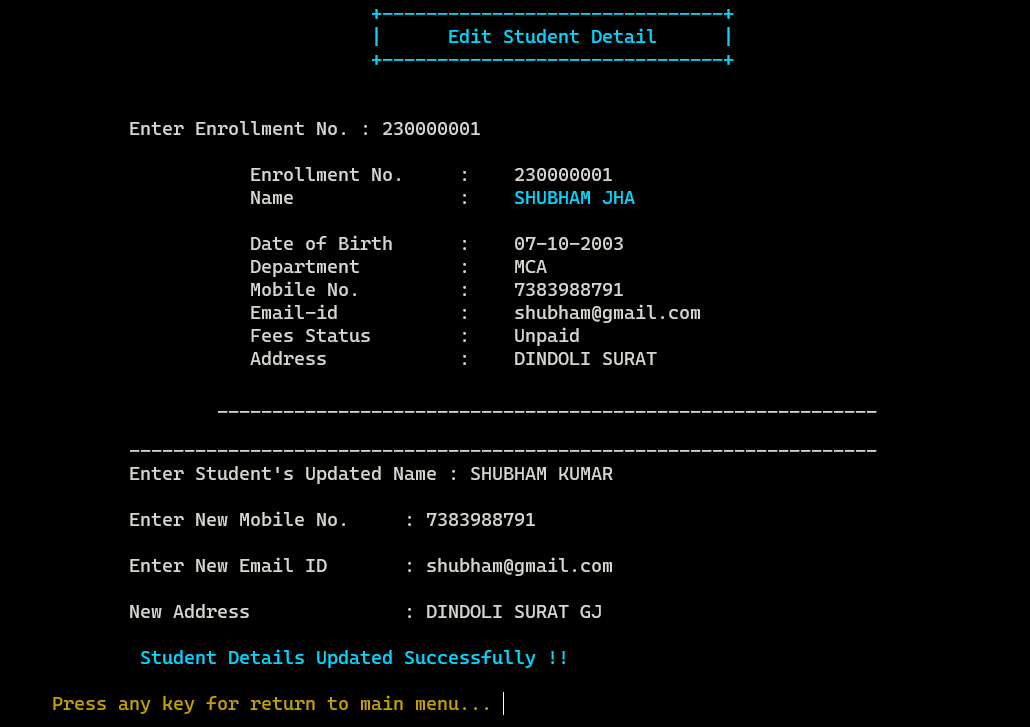
**[Fig. 5.1.12 Add New Student]**

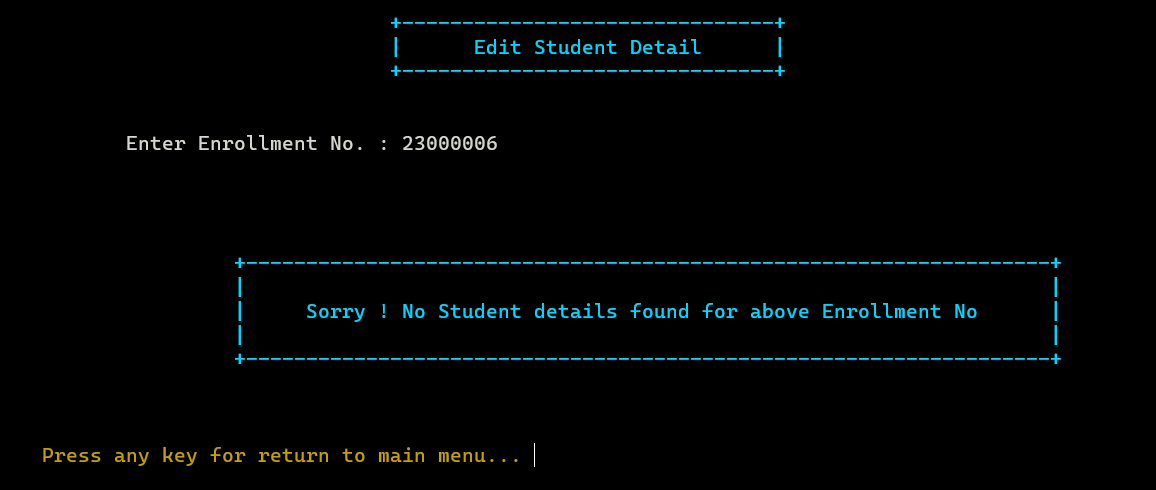


**[Fig. 5.1.13 Student List (Coursewise)]**

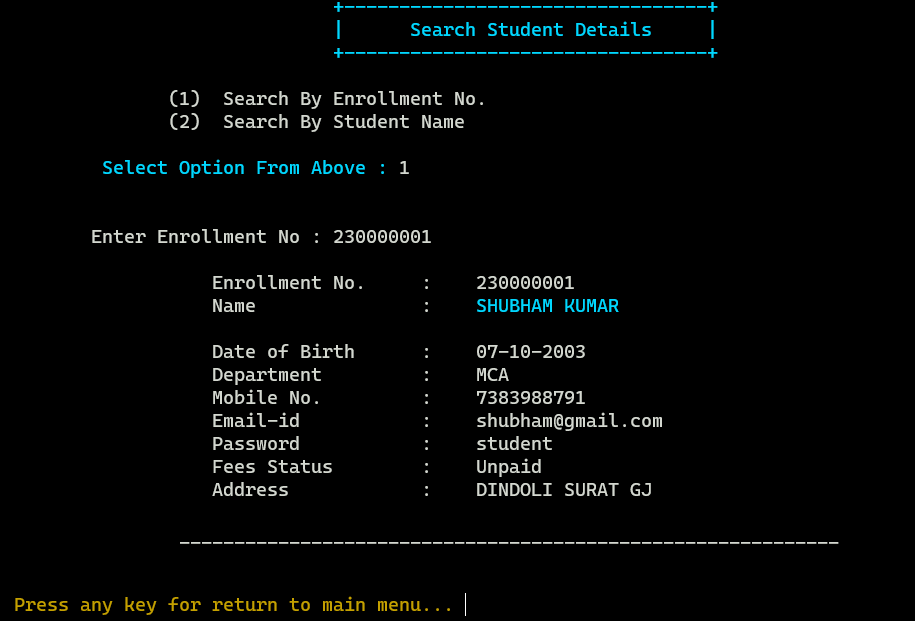


**[Fig. 5.1.14 Student List (Coursewise)]**

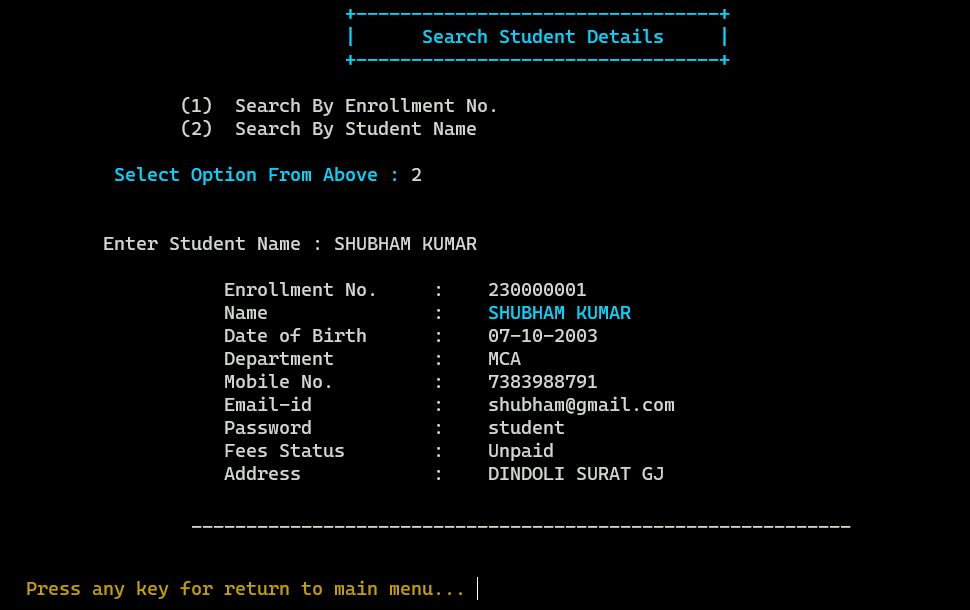




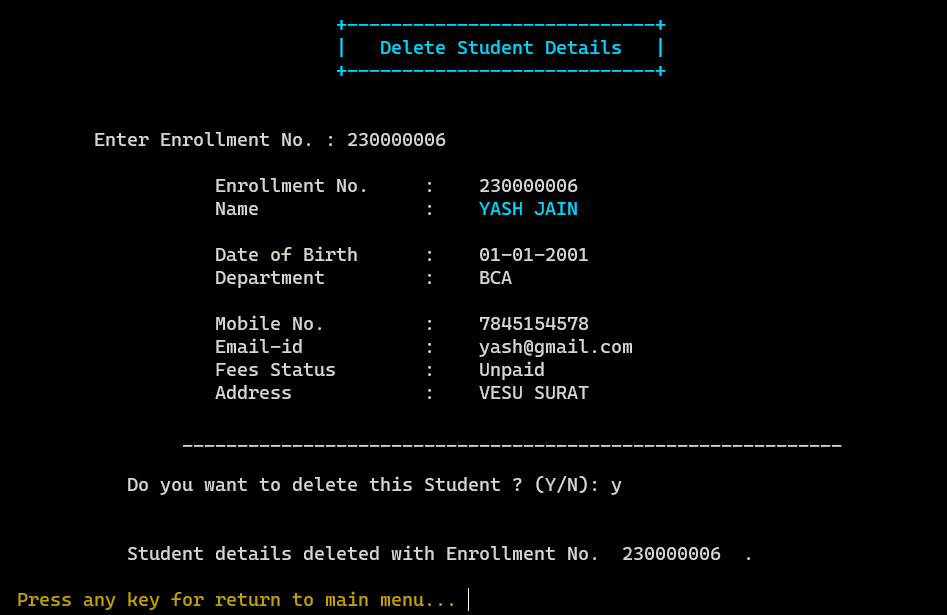
**[Fig. 5.1.15 Edit Student Details]**

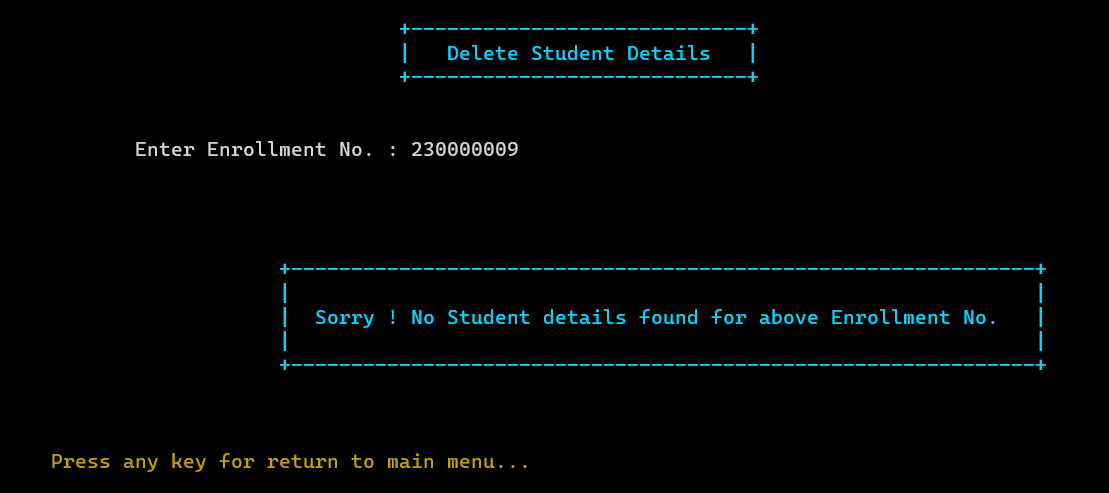


**[Fig. 5.1.16 Search Student (Using Enrollment)]**

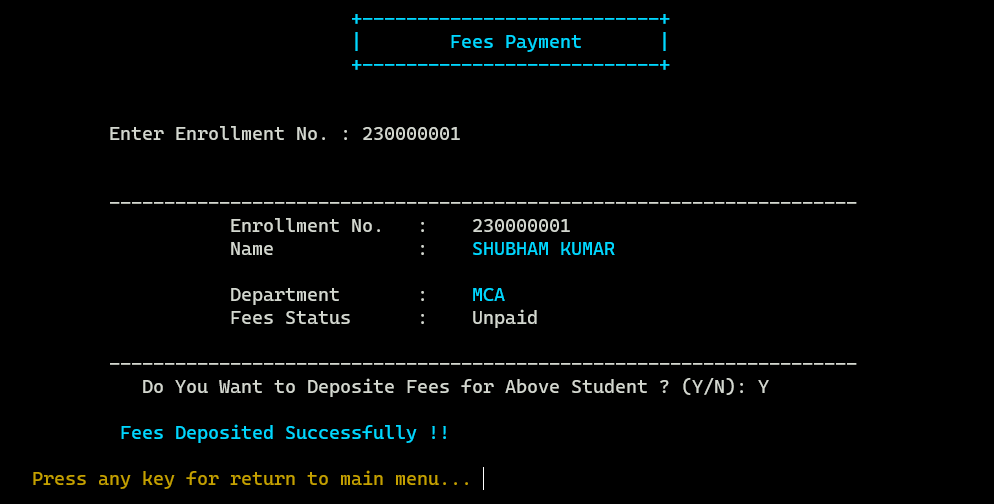


**[Fig. 5.1.17 Search Student (Using Name)]**

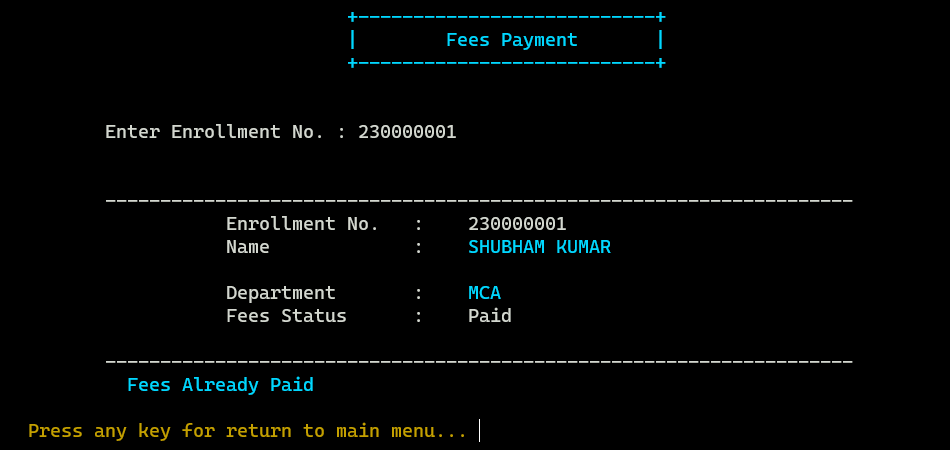




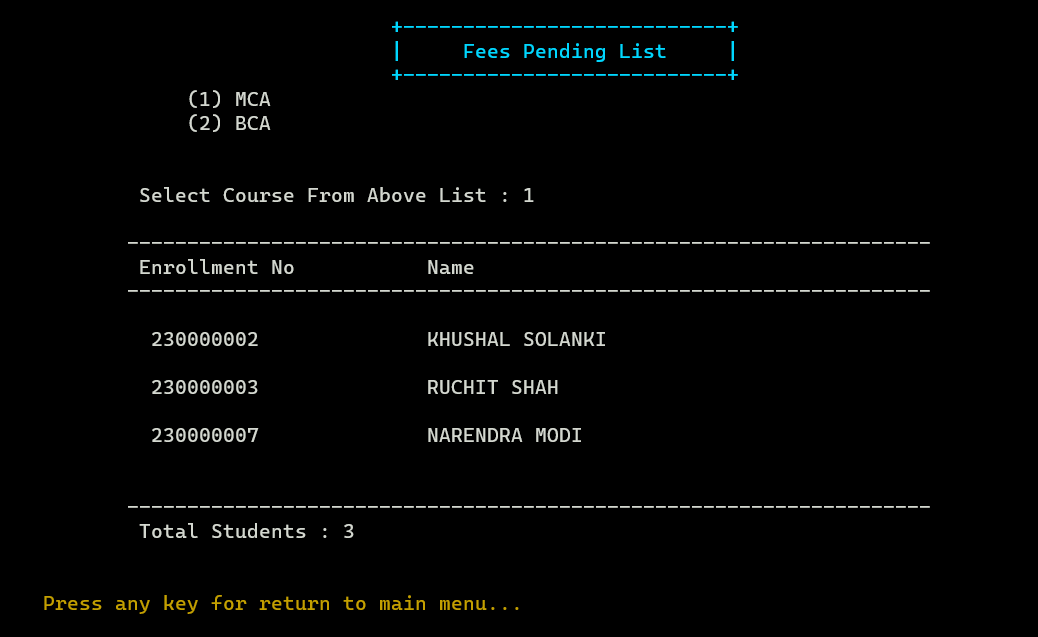
**[Fig. 5.1.18 Delete Student]**

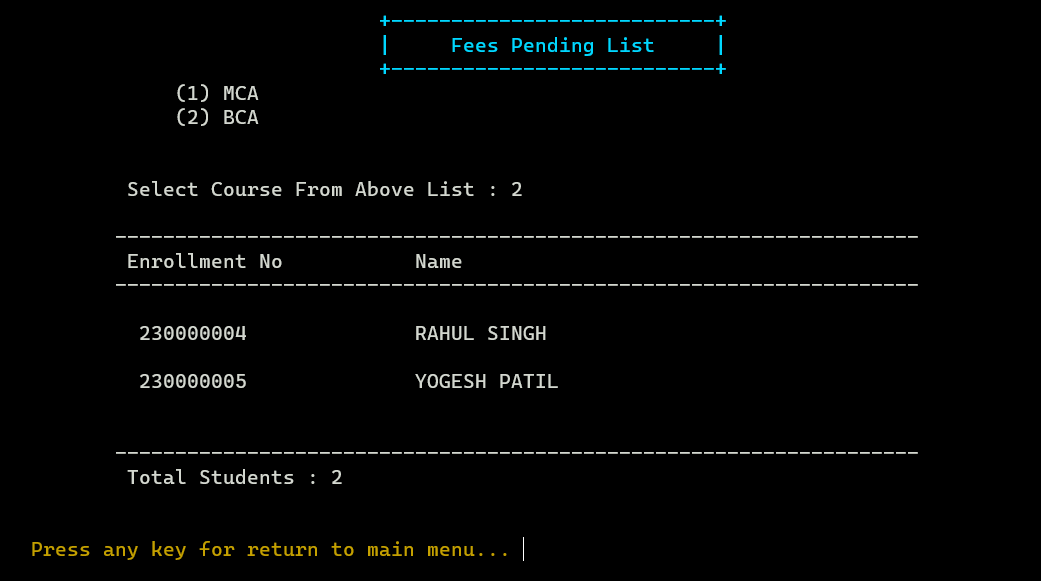


**[Fig. 5.1.19 Fees Payment]**

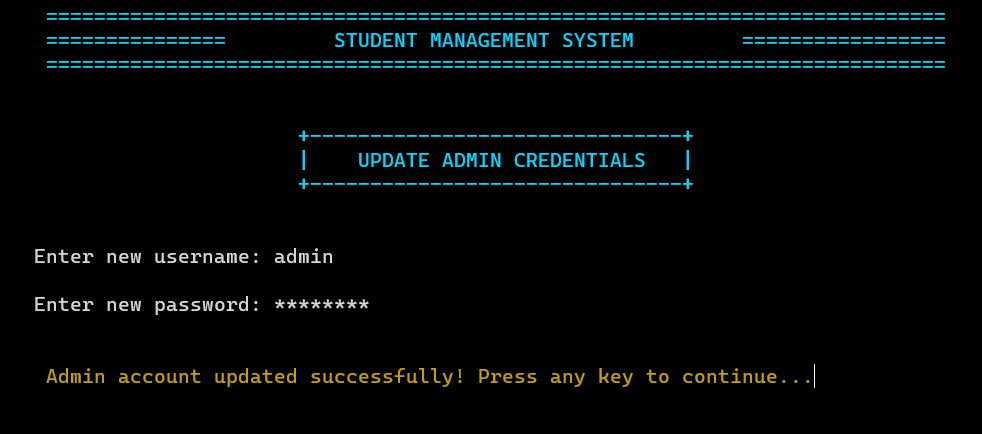


**[Fig. 5.1.20 Check Fees Status]**

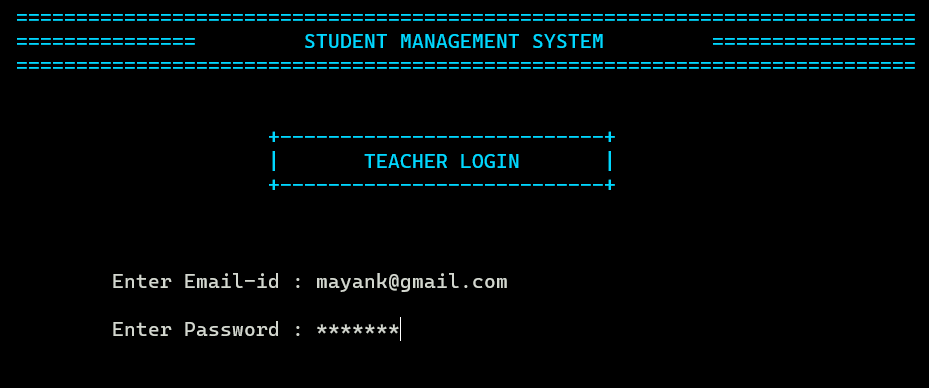




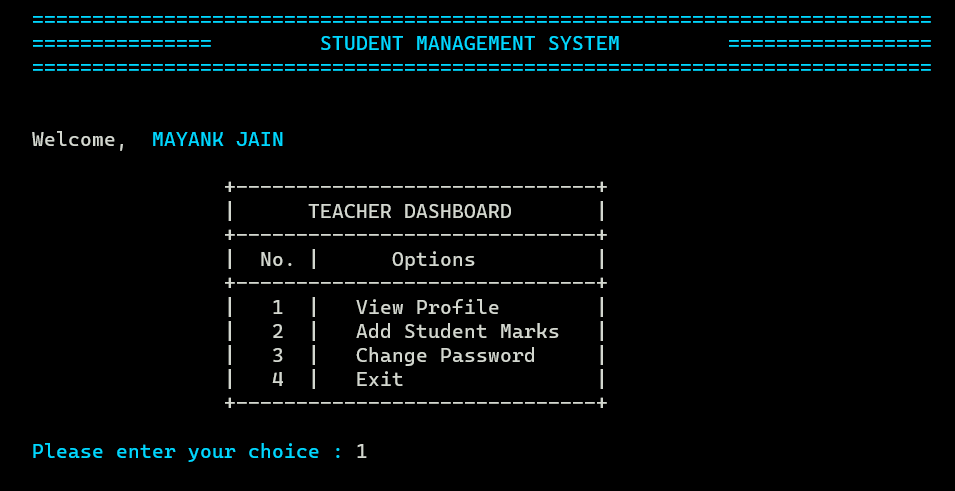
**[Fig. 5.1.21 Fees Pending List]**



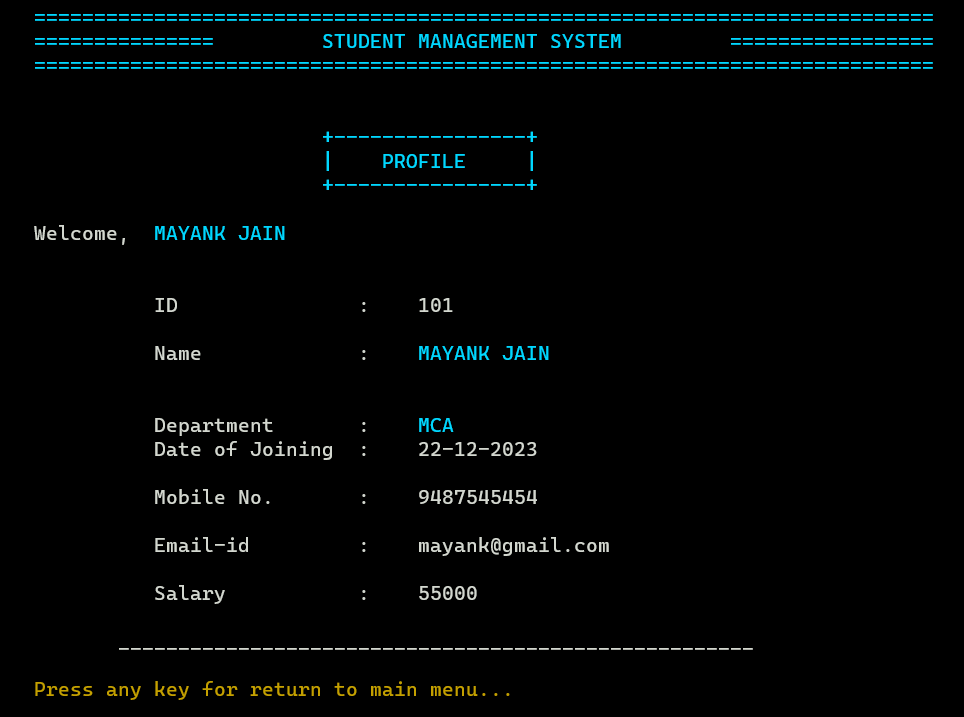
**[Fig. 5.1.22 Update Admin Credentials]**



**[Fig. 5.1.23 Teacher Login]**



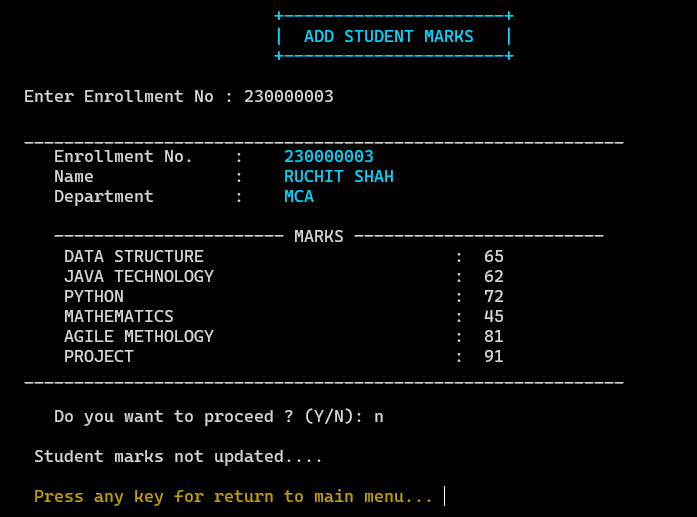
**[Fig. 5.1.24 Teacher Dashboard]**



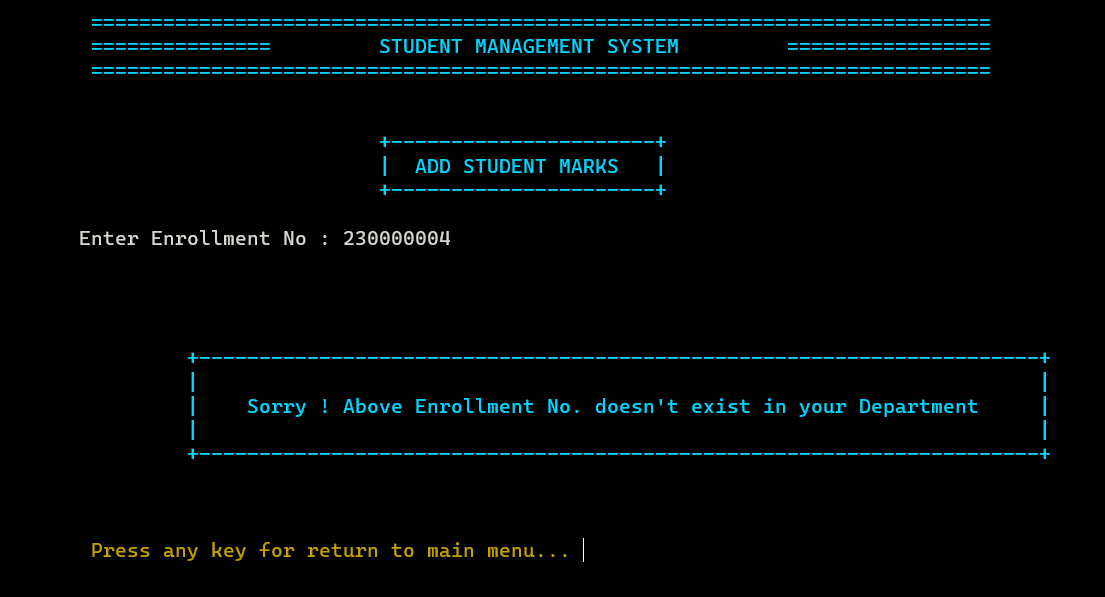
**[Fig. 5.1.25 Teacher Profile]**



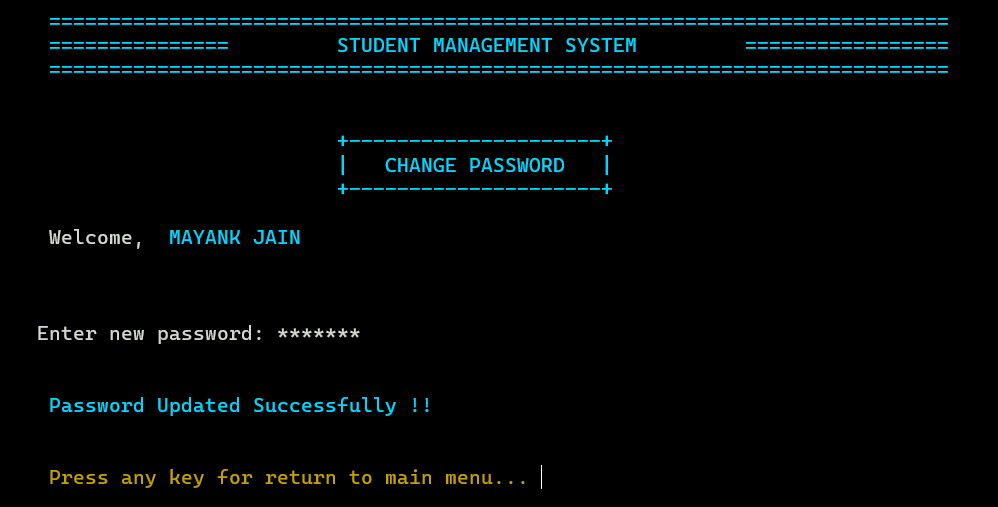
**[Fig. 5.1.26 Add Student Marks]**



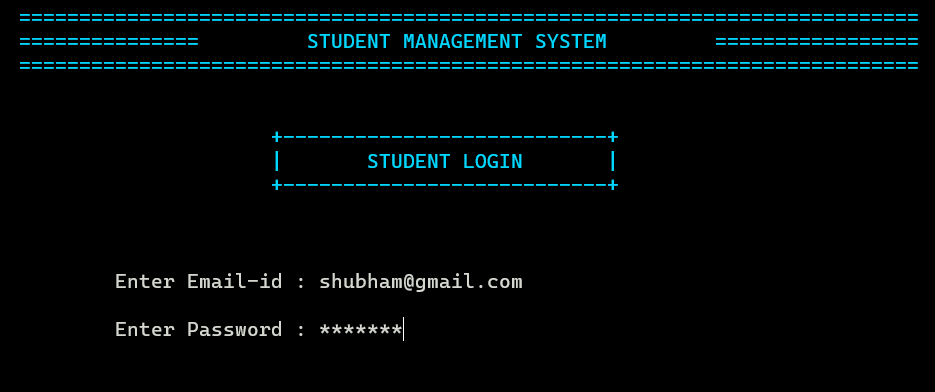
**[Fig. 5.1.27 Update Student Marks]**



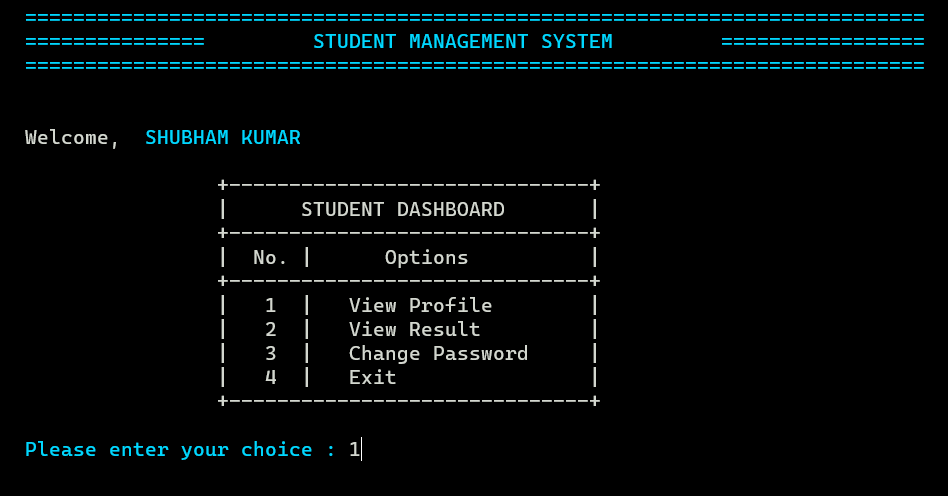
* Here Teacher can add marks of only that student which belongs to their department.



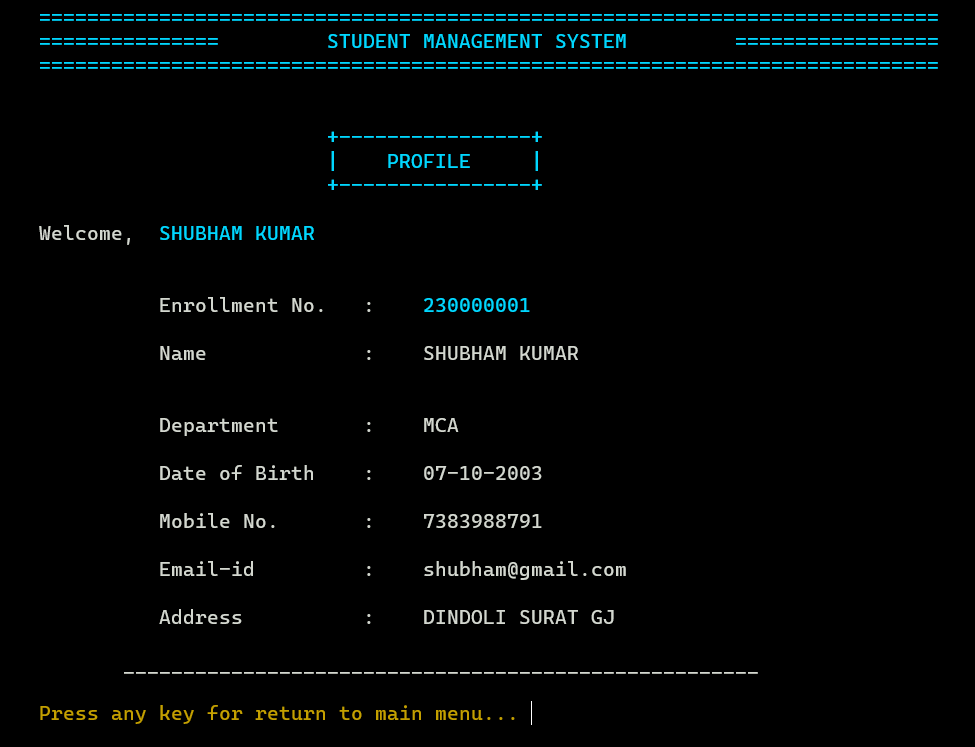
**[Fig. 5.1.28 Change Login Password(Teacher)]**



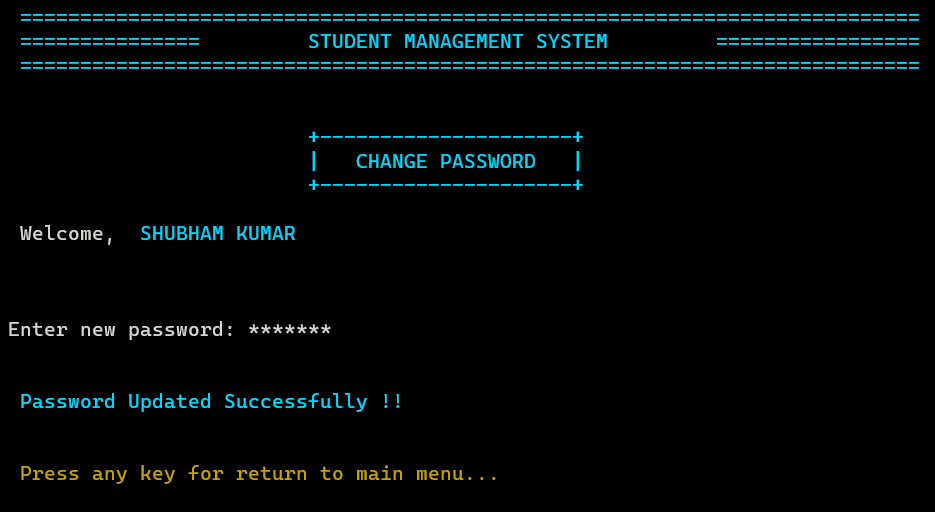
**[Fig. 5.1.29 Student Login]**



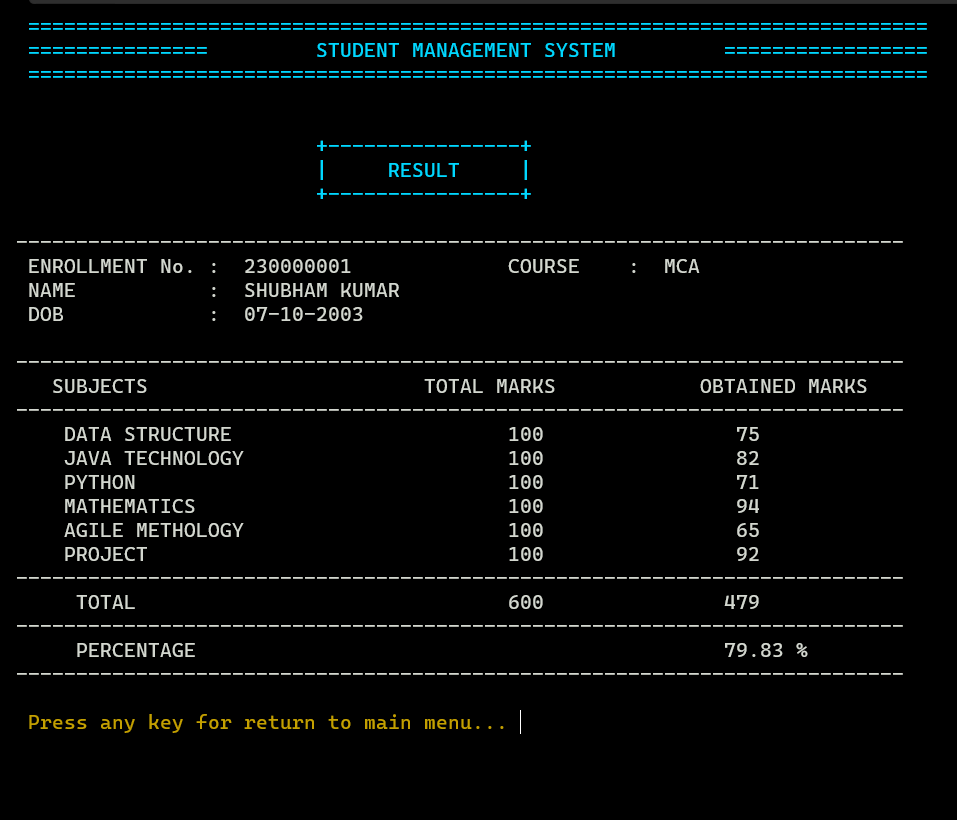
**[Fig. 5.1.30 Student Dashboard]**



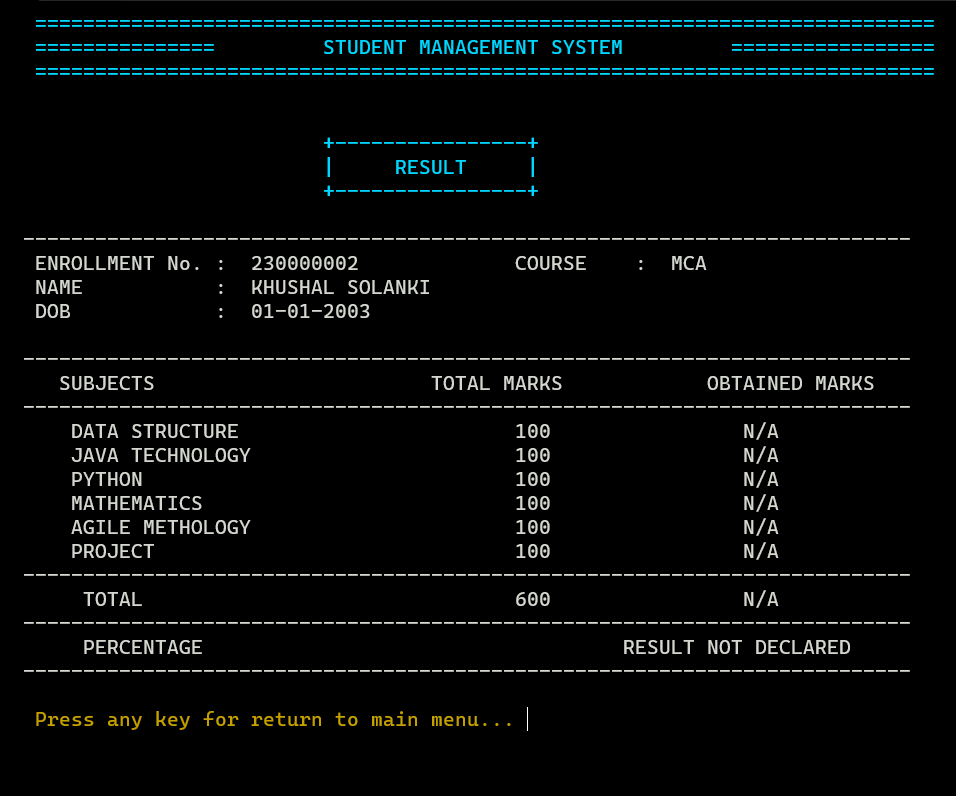
**[Fig. 5.1.31 Student Profile]**



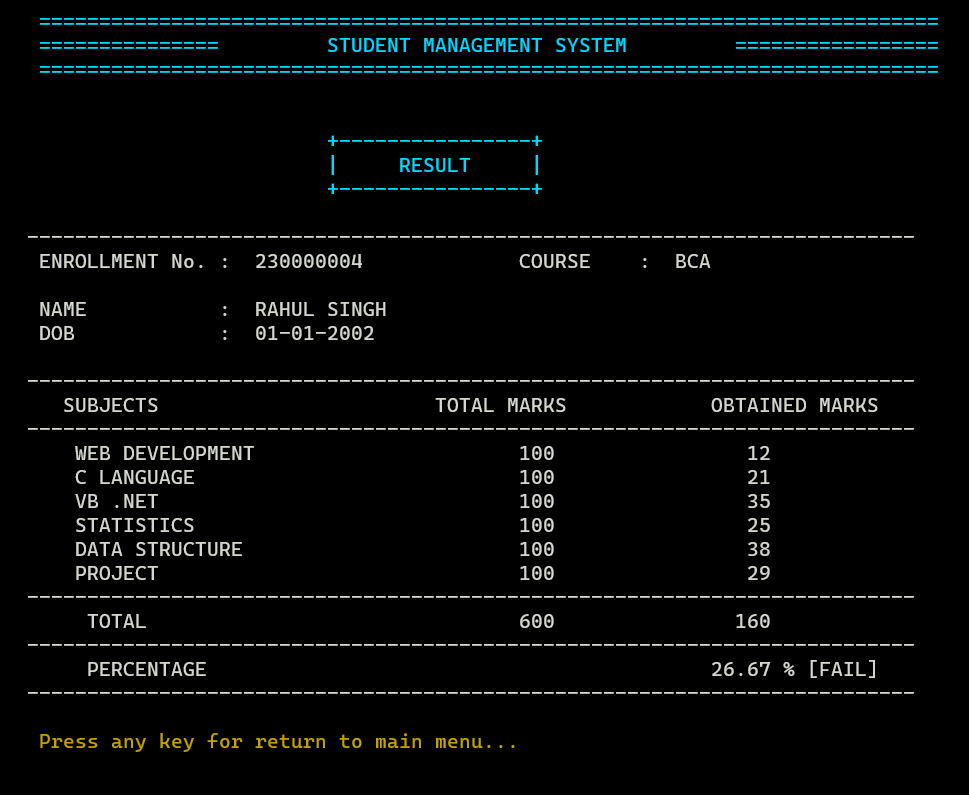
**[Fig. 5.1.32 Change Login Password (Student)]**



**[Fig. 5.1.33 View Result (Pass)]**



**[Fig. 5.1.34 View Result (Not Declared)]**



**[Fig. 5.1.35 View Result (Fail)]**

**6. References**

* <https://www.cprogramming.com/>
* <https://www.youtube.com/watch?v=ZIe_j8xAkU4>
* <https://www.geeksforgeeks.org/c-programming-language/>
* <https://www.w3schools.com/c/index.php>