**STUDENT MANAGEMENT SYSTEM**

**A PROJECT REPORT**

***Submitted by***

# JHA SHUBHAMKUMAR (ET23MTCA020)

# SHAH RUCHITKUMAR (ET23MTCA060)

# SOLANKI KHUSHALKUMAR (ET23MTCA062)

Under the Guidance Of

Prof. Preksha Parmar

***In fulfillment for the subject Software Project-1 (MTCA16106)***

# Master of Computer Application

At

**Sarvajanik College Of Engineering & Technology, Surat**

**Sarvajanik University, Surat**

**December, 2023**

|  |  |  |
| --- | --- | --- |
| SCET ARCHITECTURE | **Sarvajanik Education Society**  **Sarvajanik College of Engineering & Technology, Surat.**  **Master of Computer Application Department**  **Academic Year 2023-24** | A description... |

Date: 22/12/2023

# CERTIFICATE

This is to certify that the project entitled “STUDENT MANAGEMENT SYSTEM” has been submitted by JHA SHUBHAMKUMAR (ET23MTCA020), SHAH RUCHITKUMAR (ET23MTCA060), SOLANKI KHUSHALKUMAR (ET23MTCA062) towards fulfillment for the subject Software Project – I (MTCA16106) in 1st Semester of MCA of Sarvajanik University, Surat during the academic year 2023-24.

|  |  |
| --- | --- |
| **Guide Name: Prof. Preksha Parmar** | |
| **(Guides’ Signature)** | **Prof. Gayatri Kapadia Head, M.C.A. Department** |

|  |
| --- |
| **Examiners’ Signature:**  **1.** |
| **2.** |

**Acknowledgement**

We take this opportunity to express our sincere gratitude to all those who helped us in various capacities in undertaking this project and devising the report.

We are honored to express our sense of gratitude to our project guide **Prof.** **Preksha Parmar**. Her unparalleled knowledge, unwavering moral fiber with her invaluable expertise, provided an extraordinary level of support, guiding us seamlessly through the complexities of this project.

Our sincere thanks are extended to **Prof. Gayatri Kapadia**, the Head of Department, Master of Computer Application, for providing us all the necessary facilities for the successful completion of our project. Her encouragement and support were crucial in navigating the challenges inherent in our project, and we are truly grateful for her commitment to our academic endeavors.

In addition, we would like to convey our heartfelt appreciation to our friends and contemporaries for their invaluable co-operation and compliance throughout the project.

JHA SHUBHAMKUMAR

SHAH RUCHITKUMAR

SOLANKI KHUSHALKUMAR

|  |  |
| --- | --- |
| **Index** | **Page. No.** |
| 1. Abstract | 1 |
| 2. Software Specification | 2 |
| 3. Scope &Objectives | 3 |
| 1. DataBase Design    1. UseCase Diagram    2. Activity Diagram    3. ER Diagram | 4 - 8 |
| 1. System Design    1. screenshot | 9 |
| 6. References | 31 |

**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.

Moreover, the Student Management System emphasizes adaptability by incorporating a responsive design, making it accessible across various devices to ensures a consistent and user-friendly experience.

The System incorporates advanced security measures to safeguard sensitive student information. User authentication and access control mechanisms are implemented to ensure that only authorized personnel can access specific modules and perform designated actions. This robust security layer helps maintain confidentiality and data privacy, aligning with the highest standards of information security within educational environments.

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:**

* The Student Management System is designed to operate smoothly across all major platforms like Windows, macOS and Linux.
* Compatibility with diverse operating systems ensures accessibility for a broad user base.
* Whether users are on Windows, macOS or Linux the system guarantees a consistent and reliable performance.

|  |  |
| --- | --- |
| **Operating system** | * Microsoft windows 7 or above * Ubuntu 16.04 LTS or above * MacOS 12 or above |

**Hardware Configuration:**

**Processor:**

* The Student Management System is optimized for a minimum Intel Core i3 processor, ensuring responsive performance during runtime.

|  |  |
| --- | --- |
| **Processor** | Intel i3 or higher |

**RAM:**

* A minimum of 4 GB RAM is recommended to provide a smooth and efficient user experience while handling files and processing data.

|  |  |
| --- | --- |
| **RAM** | 4GB or higher |

**Storage:**

* The system requires approximately 128 GB of storage space, accommodating the application's executable files and data storage for student and teacher’s records.

|  |  |
| --- | --- |
| **Storage** | 128 GB or higher |

**Development Environment:**

* The Student Management System is designed to be developed and compiled using the Dev C++, Turbo C or Visual Studio integrated development environments (IDEs).

|  |  |
| --- | --- |
| **Development Environment** | Dev C++, Turbo C or Visual Studio |

**Programming Language:**

* This system is implemented in the C programming language, ensuring compatibility with the compilers provided by Dev C++ or Turbo C.

|  |  |
| --- | --- |
| **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.

In the context of the Student Management System, the aim is to provide administration staff with a convenient and straightforward method for maintaining records through the command-line interface. This eliminates the need for maintaining physical files or books with the goal of enhancing accessibility.

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

**4. Database Design**

**File System for Data Storage**

* The Student Management System uses file system for storing and managing student and teacher’s data.

**File Structure:**

|  |
| --- |
| **Course** |
| Course ID |
| Course Name |
| Fees |
| Subjects [6] |

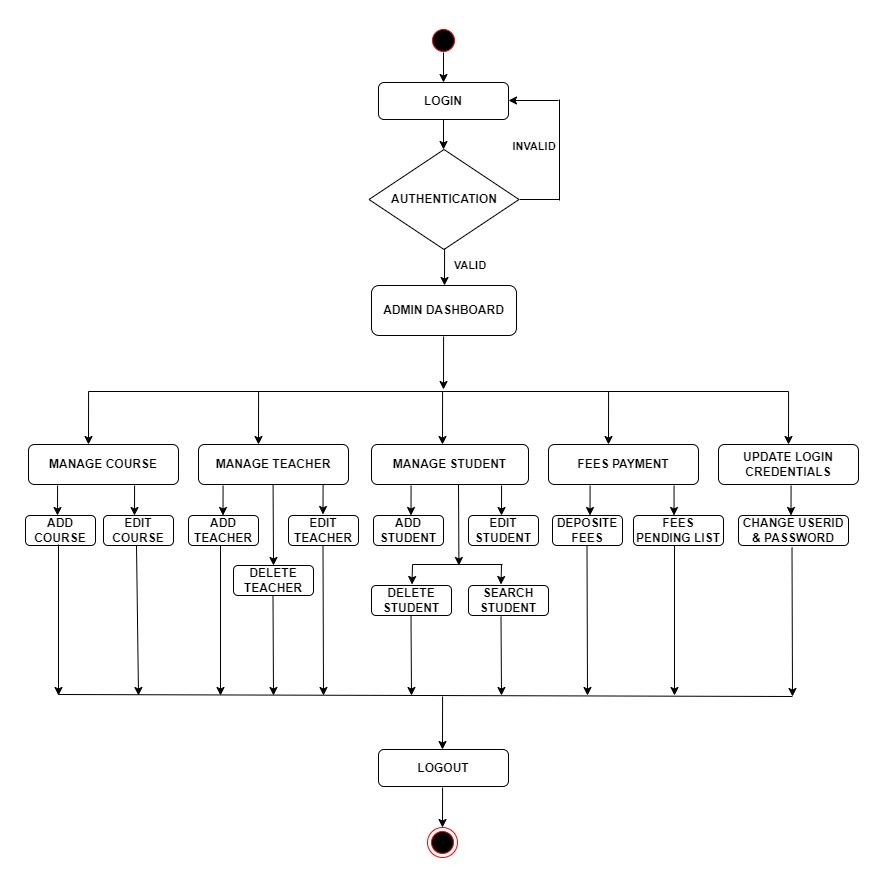
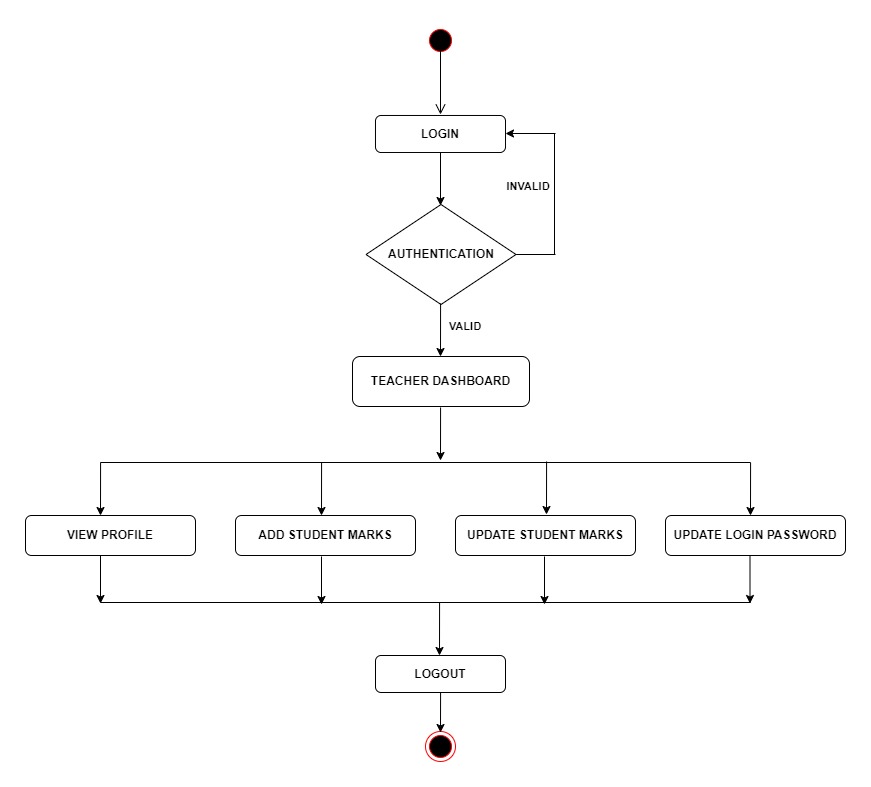
|  |
| --- |
| **Teacher** |
| Teacher ID |
| Name |
| Department |
| Date of Joining |
| Salary |
| Mobile No |
| Email |
| Password |

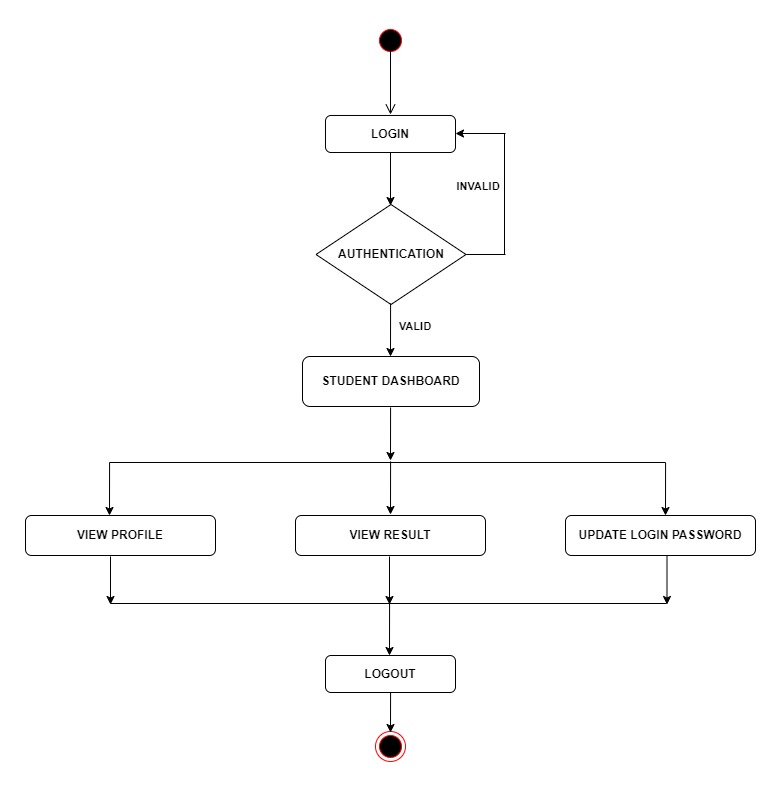
|  |
| --- |
| **Student** |
| Enrollment No |
| Name |
| Date of Birth |
| Department |
| Mobile No |
| Address |
| Email |
| Password |
| Fee |
| Marks [Subjects] [6] |

**4.1 Use Case Diagram**

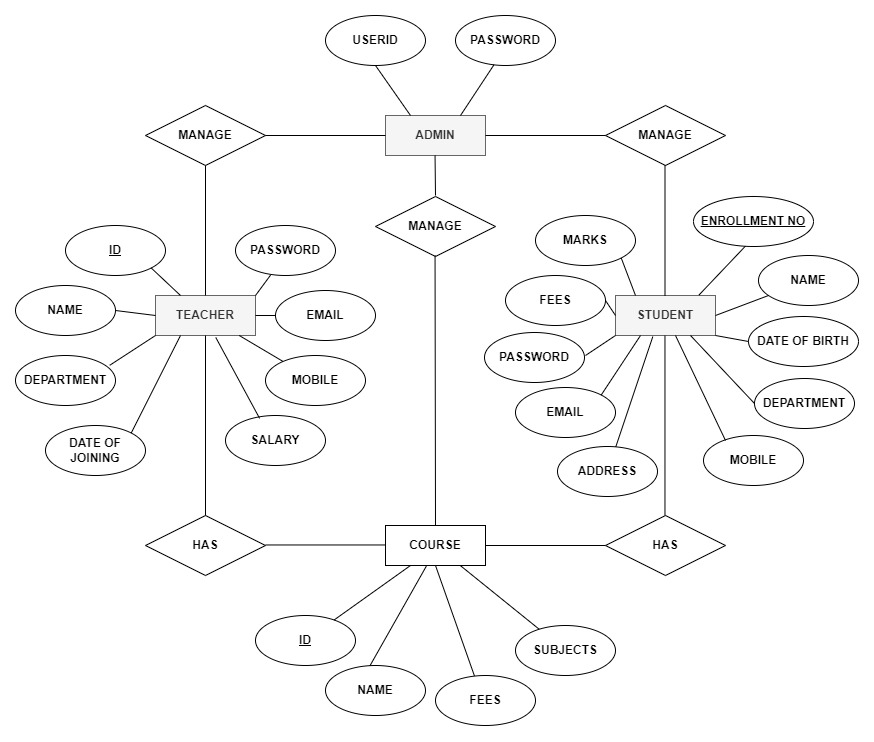
****

**4.2 Activity Diagram**

* **ADMIN **
* **TEACHER**
* **STUDENT**

****

**4.3 ER Diagram**

****

**5. System Design**

In the system design phase of the Student Management System (SMS), our focus is on crafting a robust architecture that seamlessly integrates various modules to fulfill the diverse requirements of educational institutions. The design is structured to optimize efficiency, ensure data integrity, provide a user-friendly experience, and allow for scalability.

The Student Module forms the foundational element for managing student information, encompassing vital personal details, enrollment information, and contact details. In lieu of a relational database model, the system adopts a file-based approach in C programming. The file handling mechanism is thorough implemented to efficiently organize and manage student data, allowing for quick retrieval and modification as needed. This design choice aligns with the project's goals of simplicity and resource efficiency, providing an effective means of storing and accessing student information within the system.

The Admin Module is designed to handle course-related information, student enrollments, marksheet, and fee payments. The system facilitates the dynamic recording of academic data while ensuring accuracy and consistency. This module contributes to a comprehensive academic profile for each student.

The User Interface Module is engineered to provide a user-friendly experience for administrators, teachers, and students. By incorporating responsive design principles, the system ensures accessibility across various devices. Robust user authentication and access control mechanisms are integrated to safeguard sensitive information, allowing authorized personnel to perform specific actions.

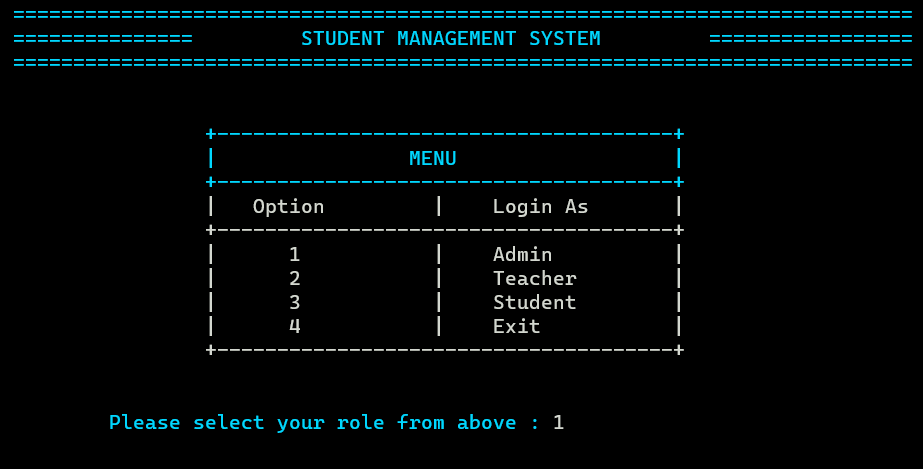
Ensuring data integrity and system reliability is paramount. Input validation mechanisms are implemented to validate user inputs and prevent data inconsistencies. Error handling strategies are also embedded to address unforeseen issues promptly, maintaining a stable and dependable system.

The choice of the C programming language aligns with our goal of efficiency and versatility. Leveraging file handling for database management ensures an effective implementation that balances performance with simplicity. The system design emphasizes scalability, allowing for seamless expansion to accommodate future data growth and evolving educational needs. This modular and scalable design sets the foundation for a maintainable system that can be easily updated and enhanced in response to changing requirements.

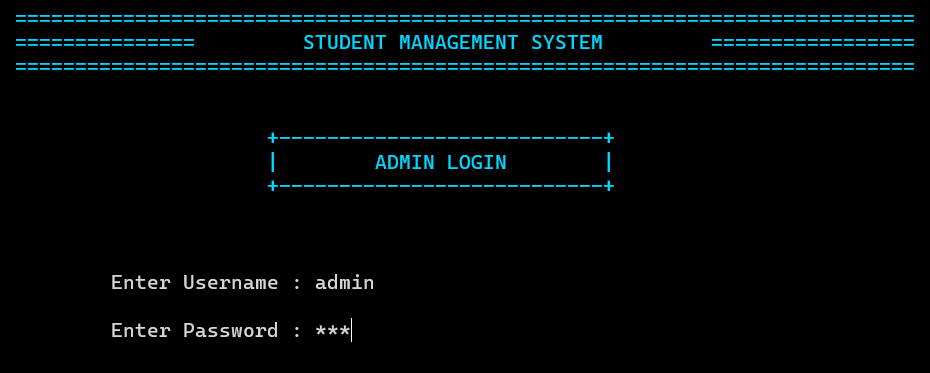
In summary, the system design for the Student Management System integrates key modules aiming to optimize functionality, user experience, and reliability.

**5.1 Screenshot**

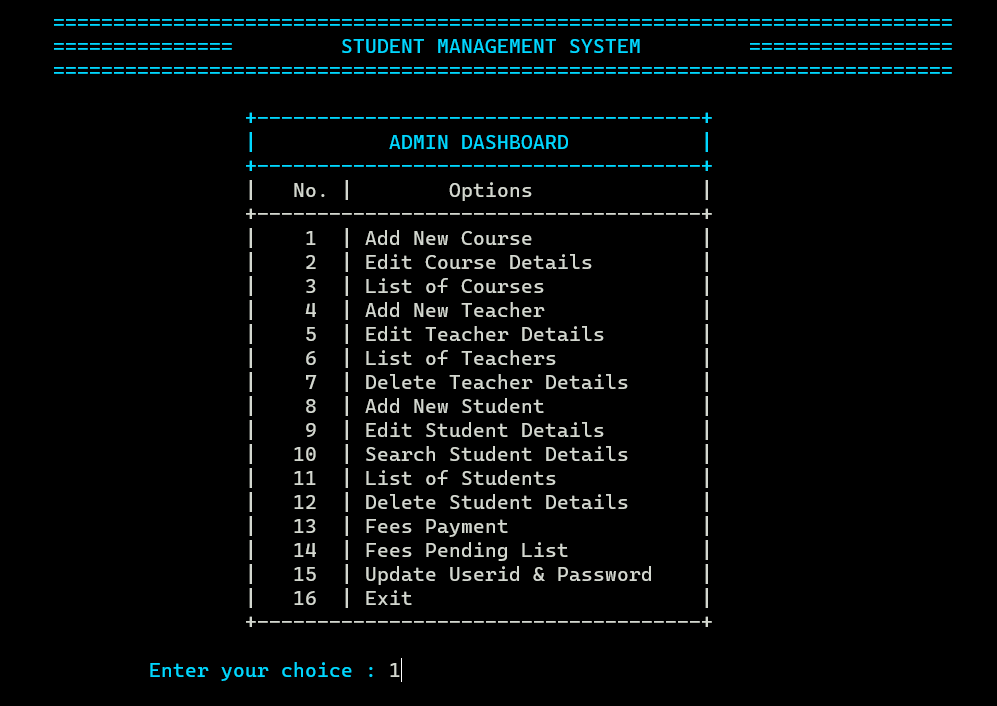
**Welcome Page**



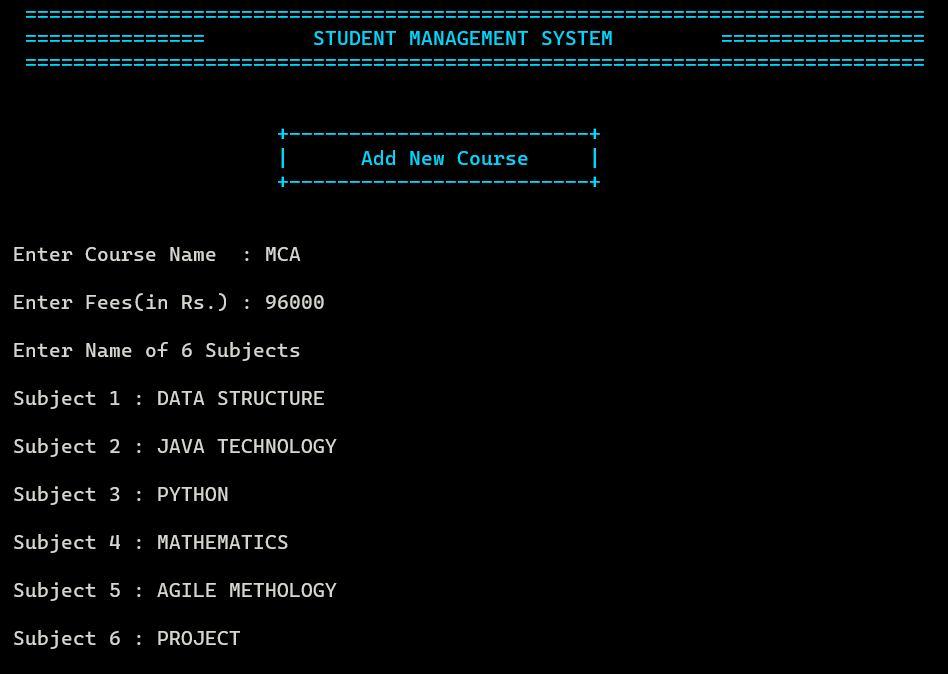
**Admin Login**



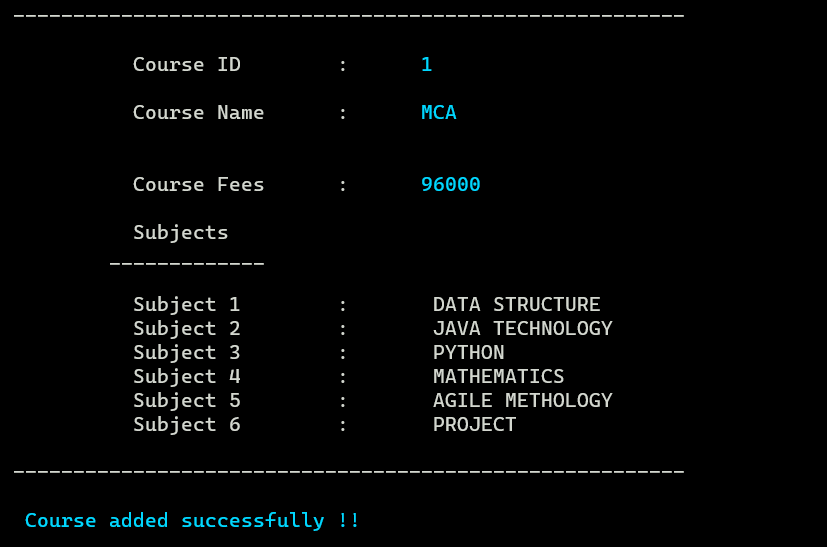
**Admin Dashboard**



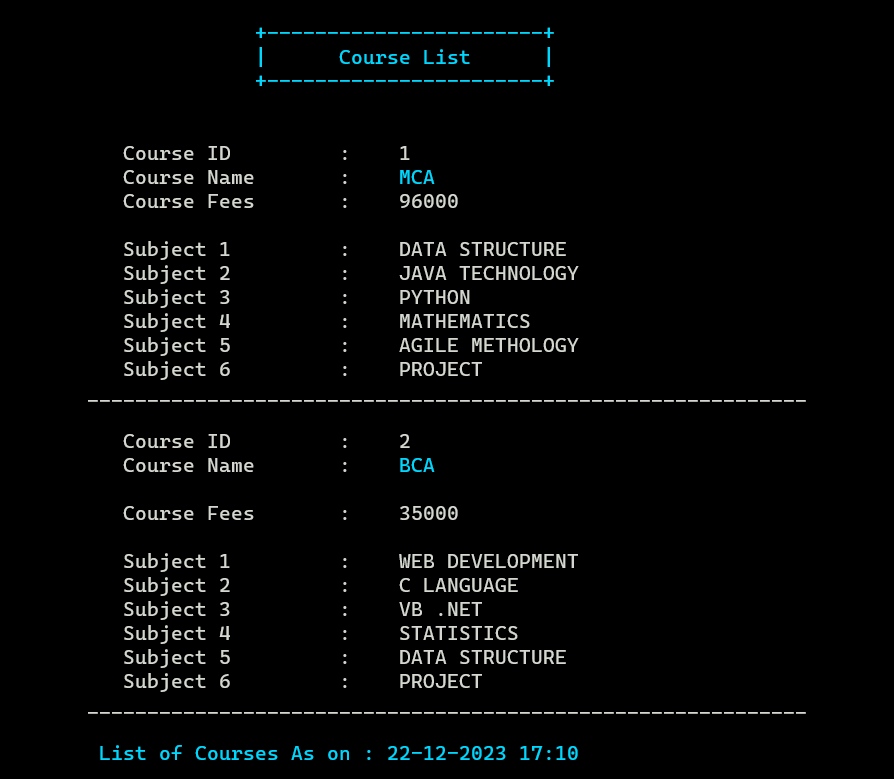
**Add New Course**



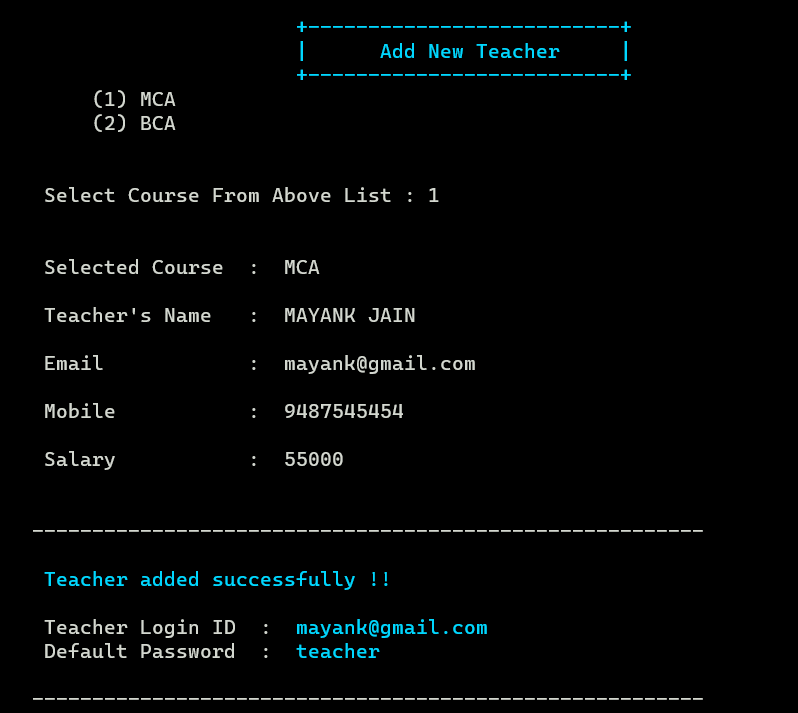
**Course Added**



**Course List**



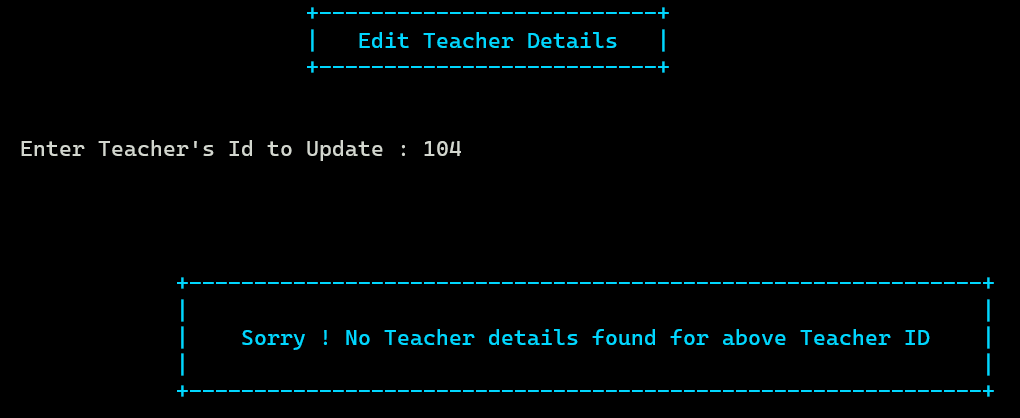
**Add New Teacher**

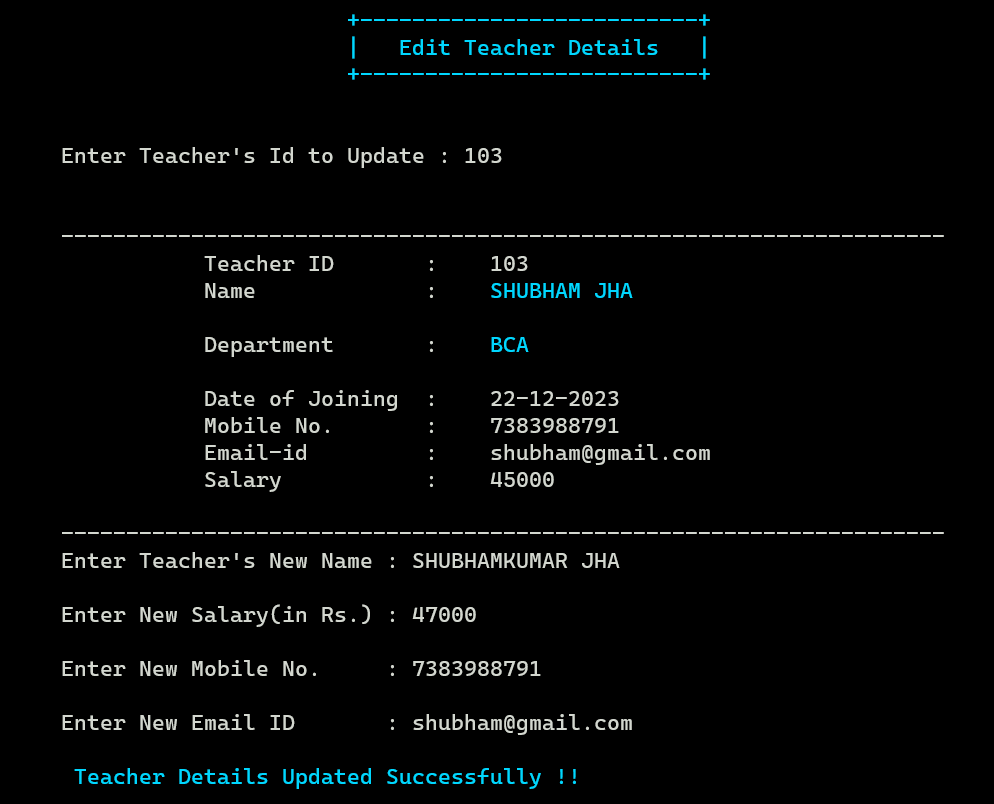


**Teacher’s List**

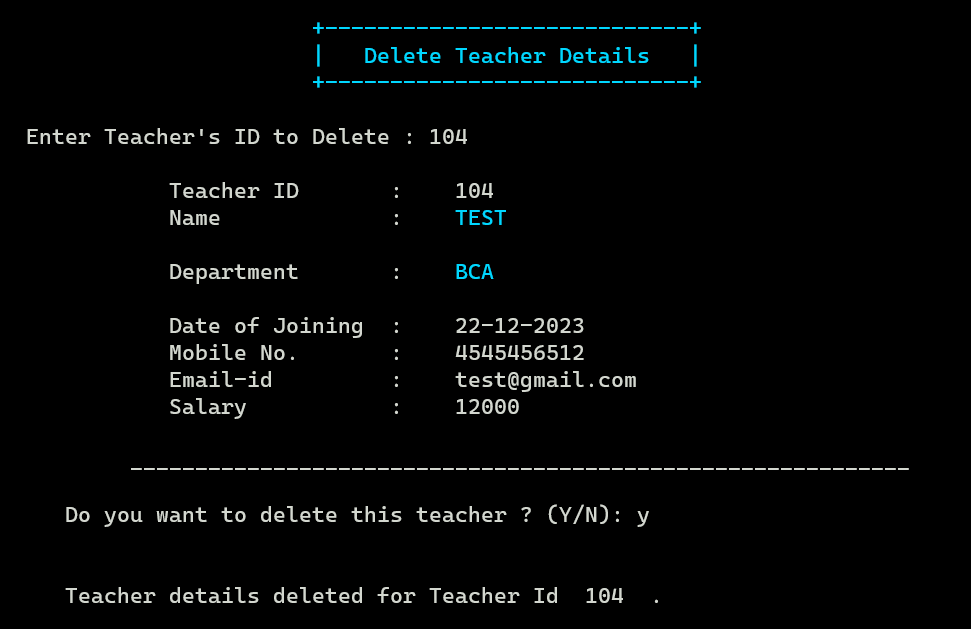


**Edit Teacher Details**

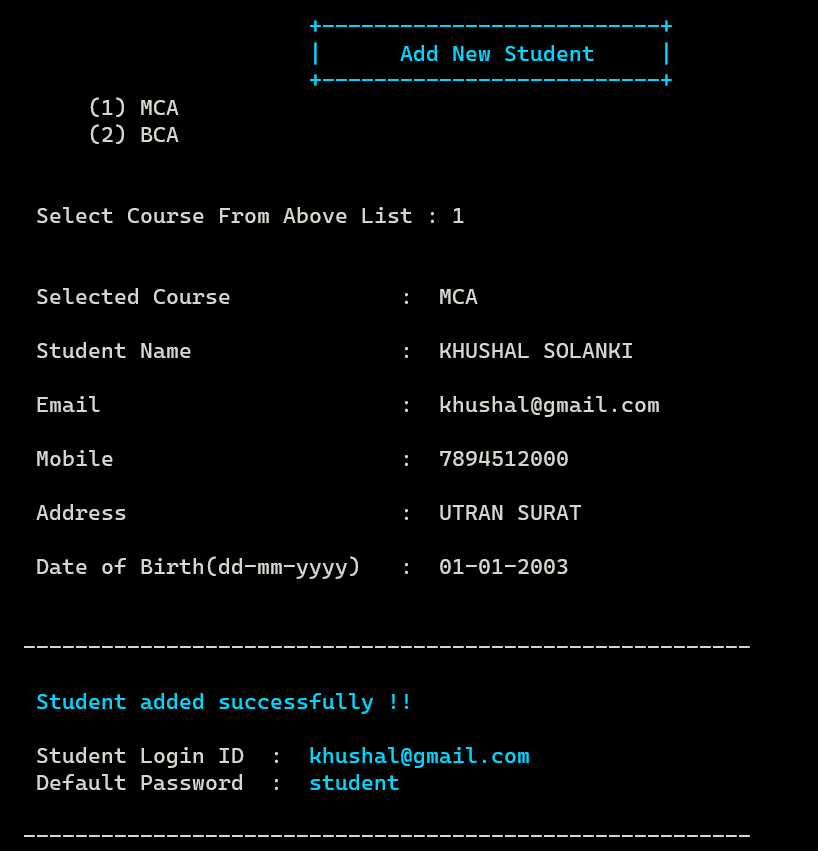




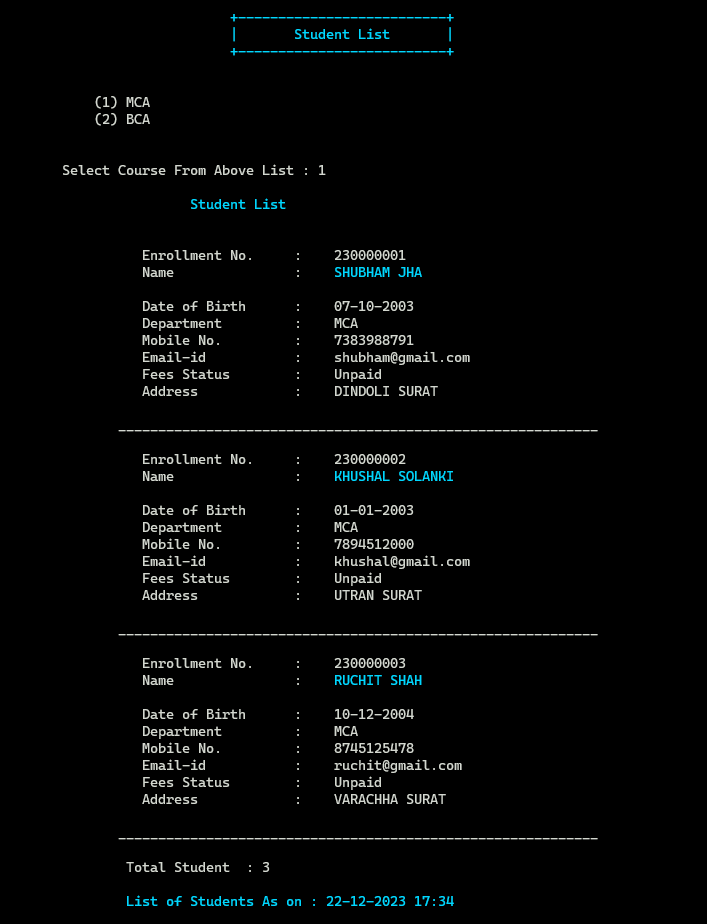
**Delete Teacher Details**



**Add New Student**



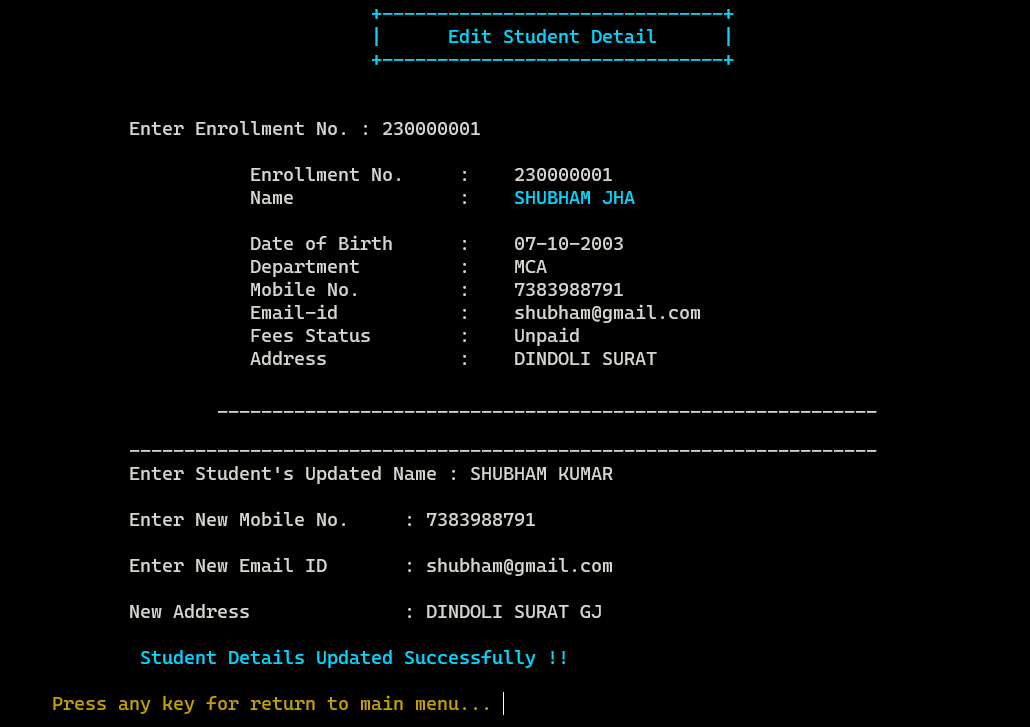
**Student List (Course wise)**

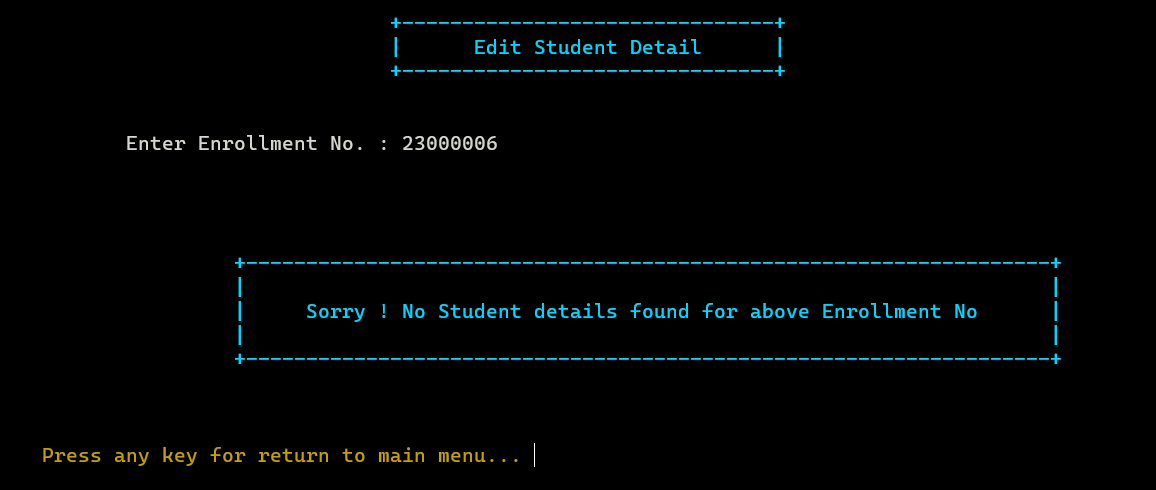


**Student List (Course wise)**

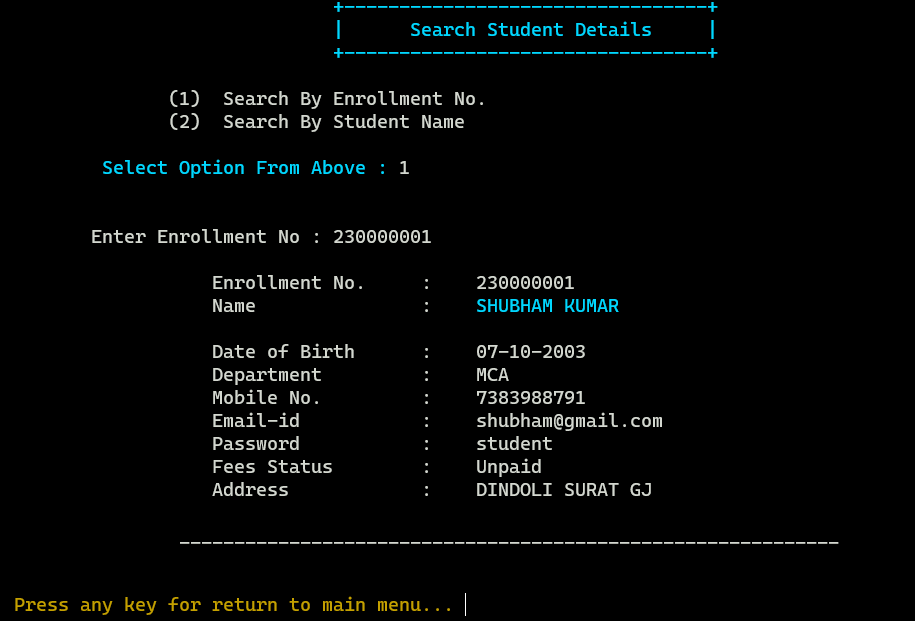


**Edit Student Detail**

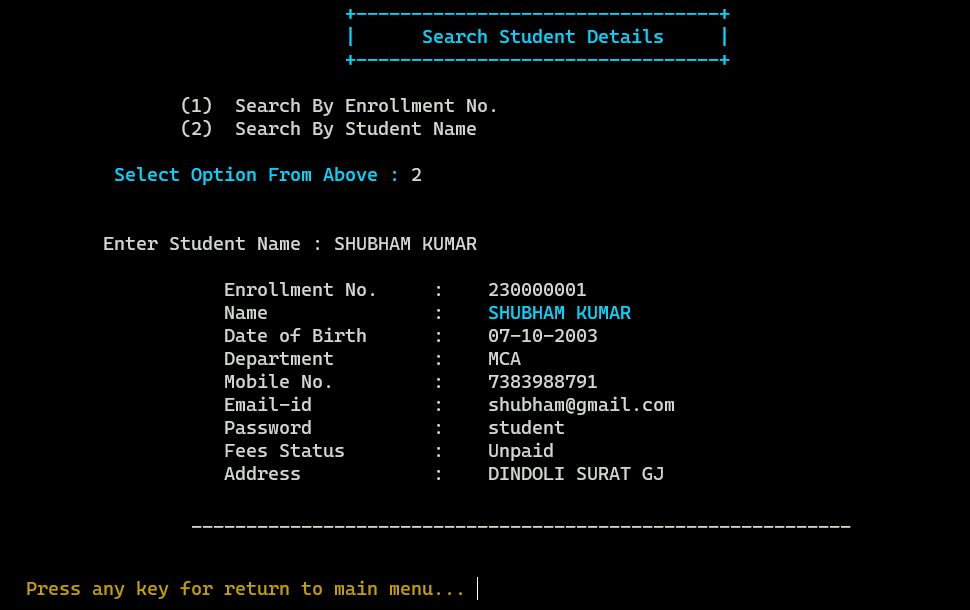




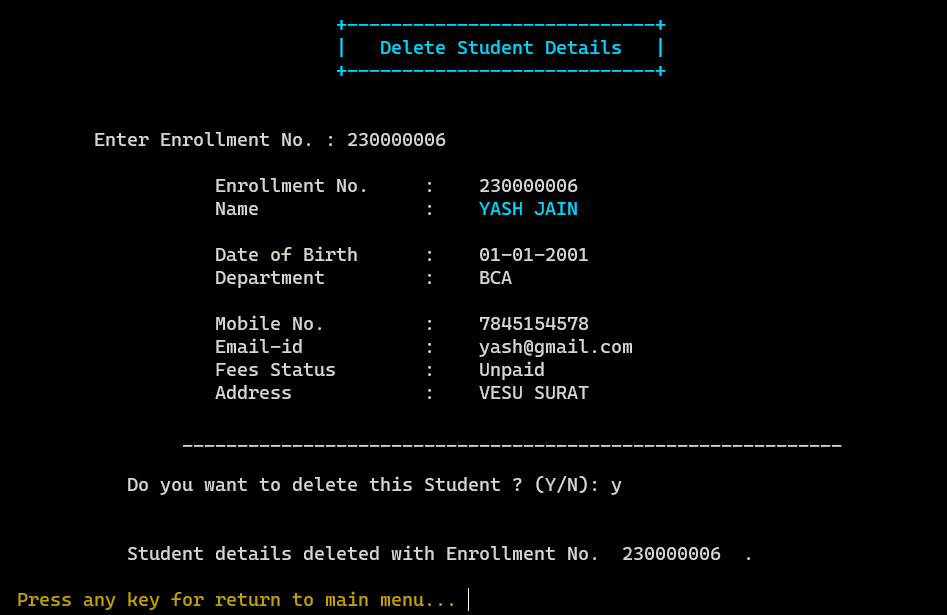
**Search Student By Enrollment No.**

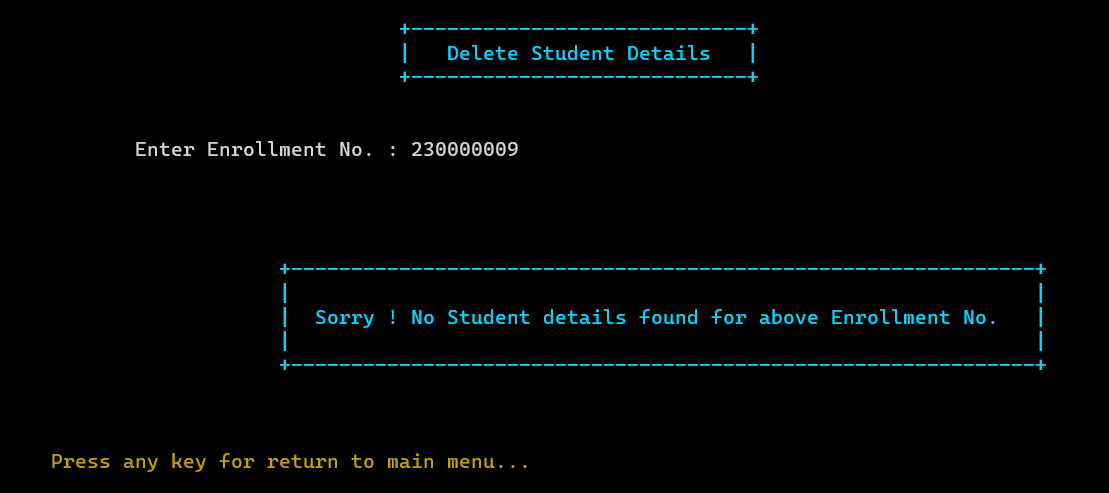


**Search Student By Name**

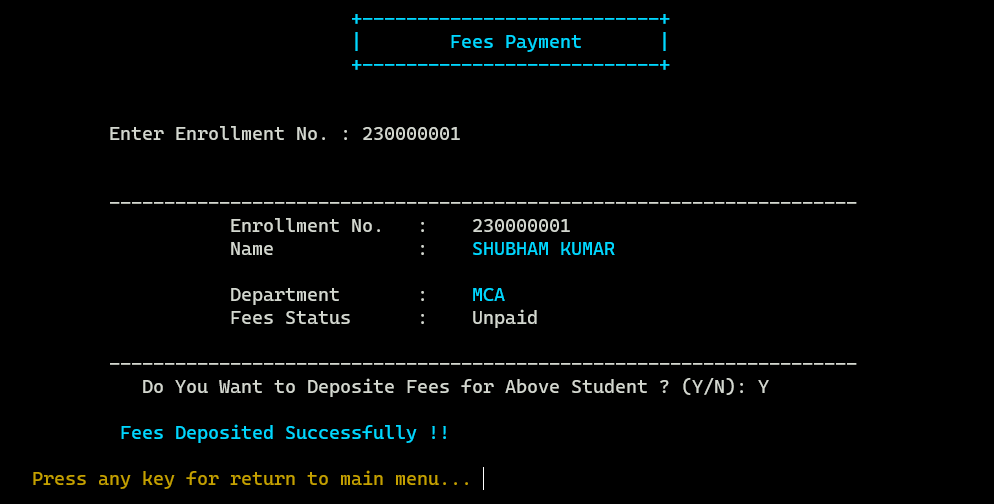


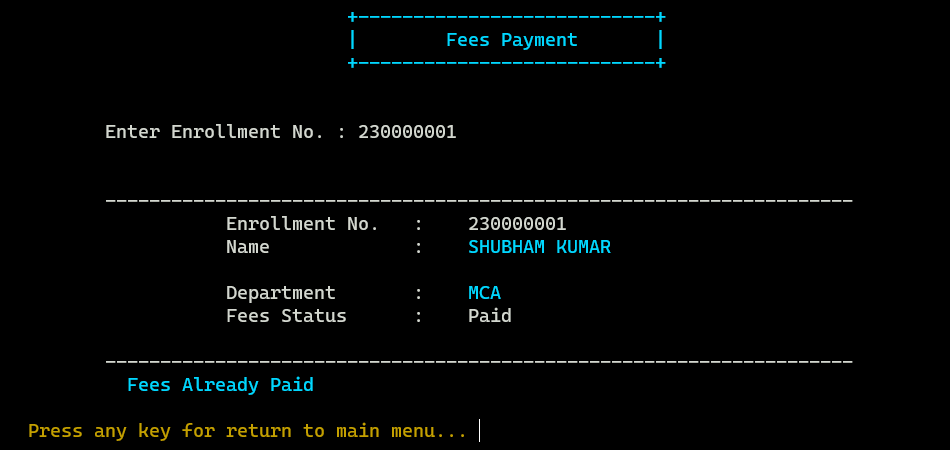
**Delete Student Detail**



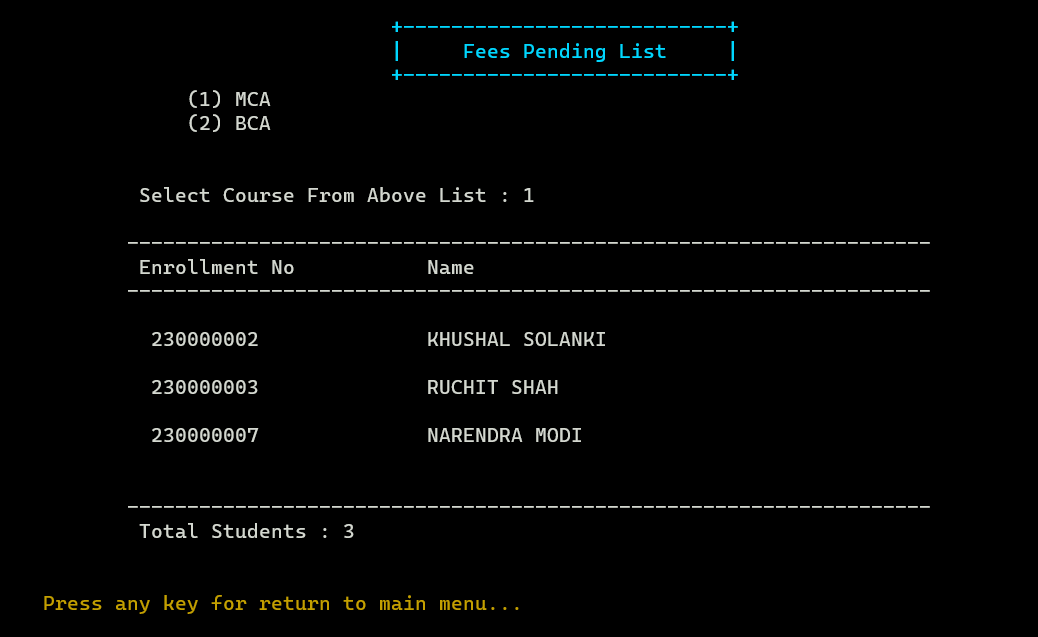


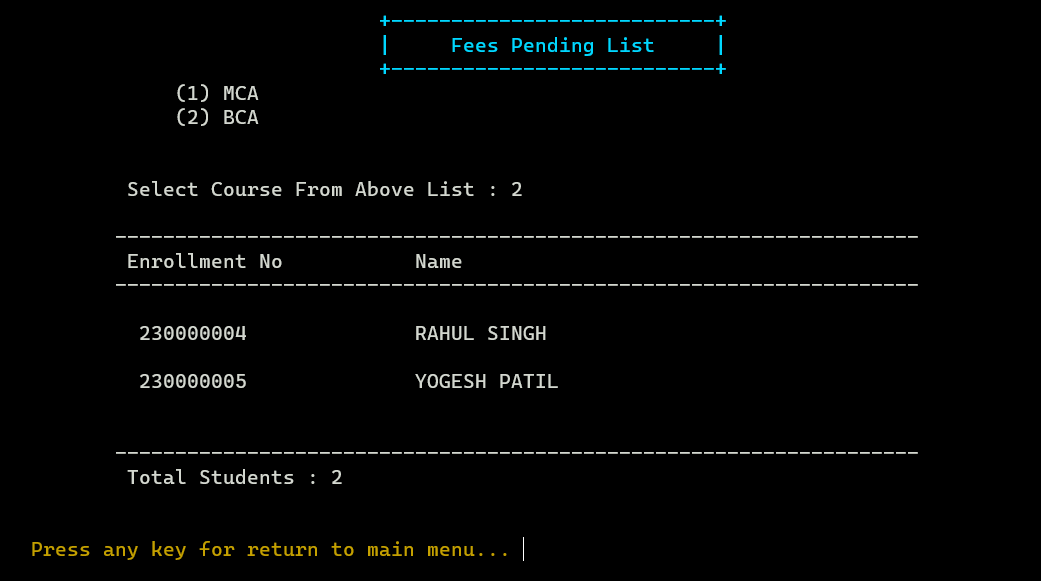
**Fees Payment**



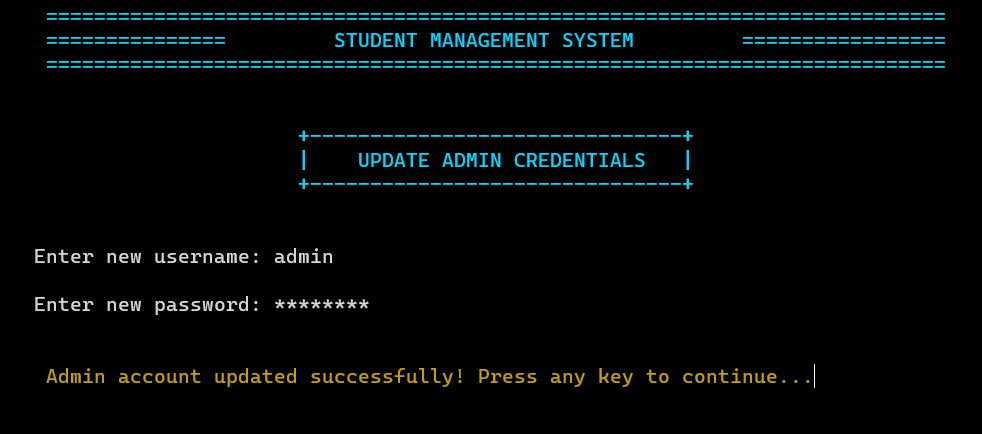


**Fees Pending List (Course wise)**

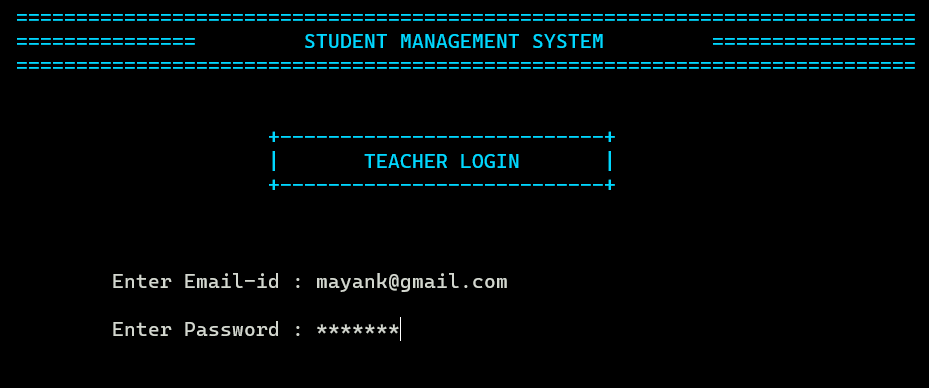




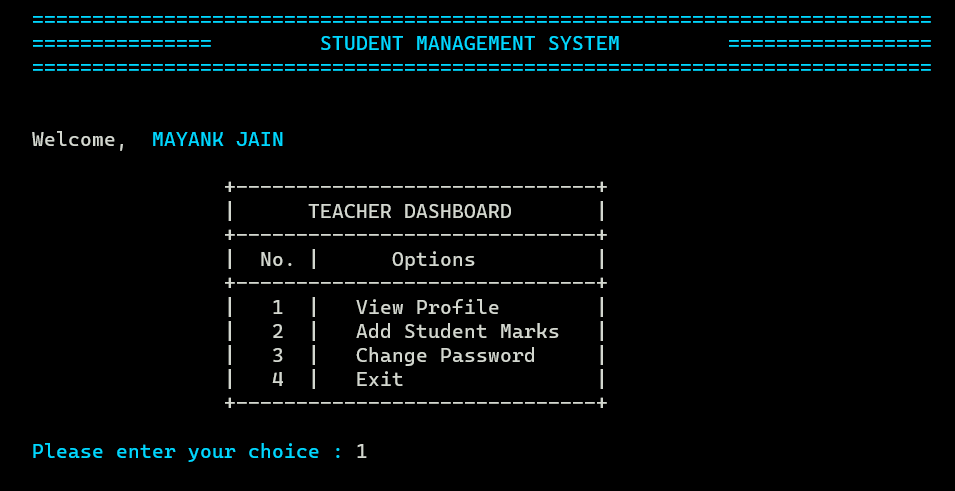
**Update Admin Credentials**



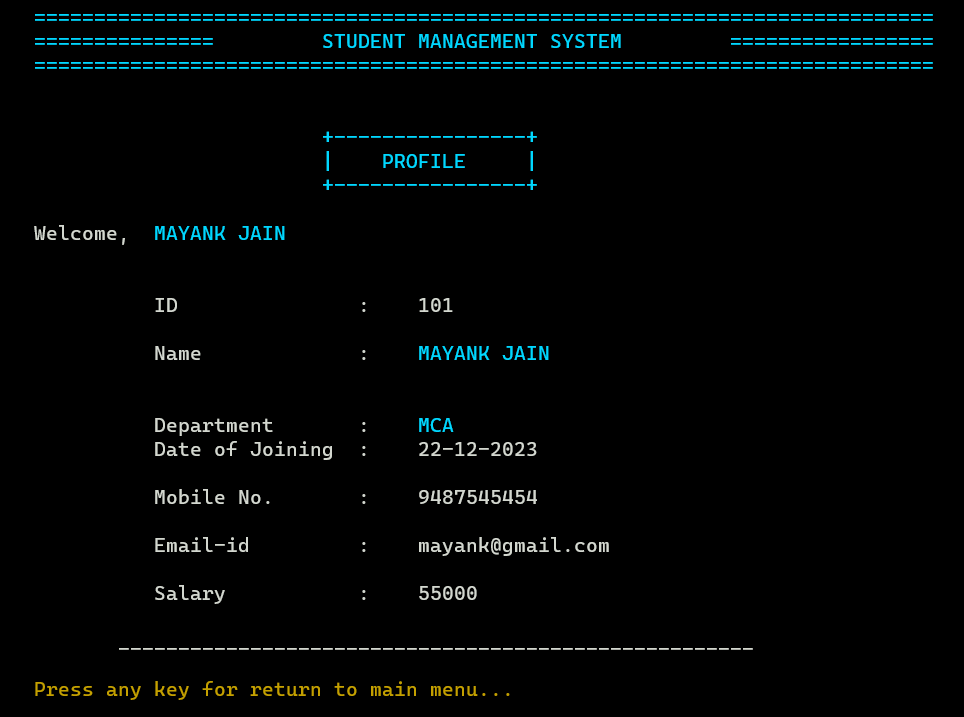
**Teacher Login**



**Teacher Dashboard**



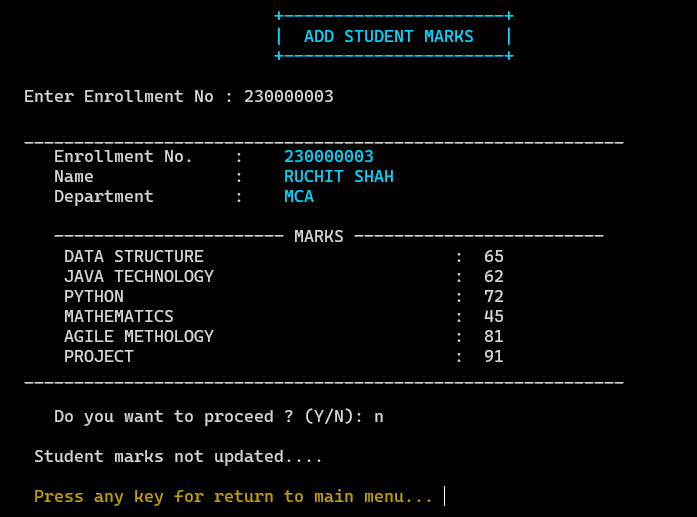
**Teacher Profile**



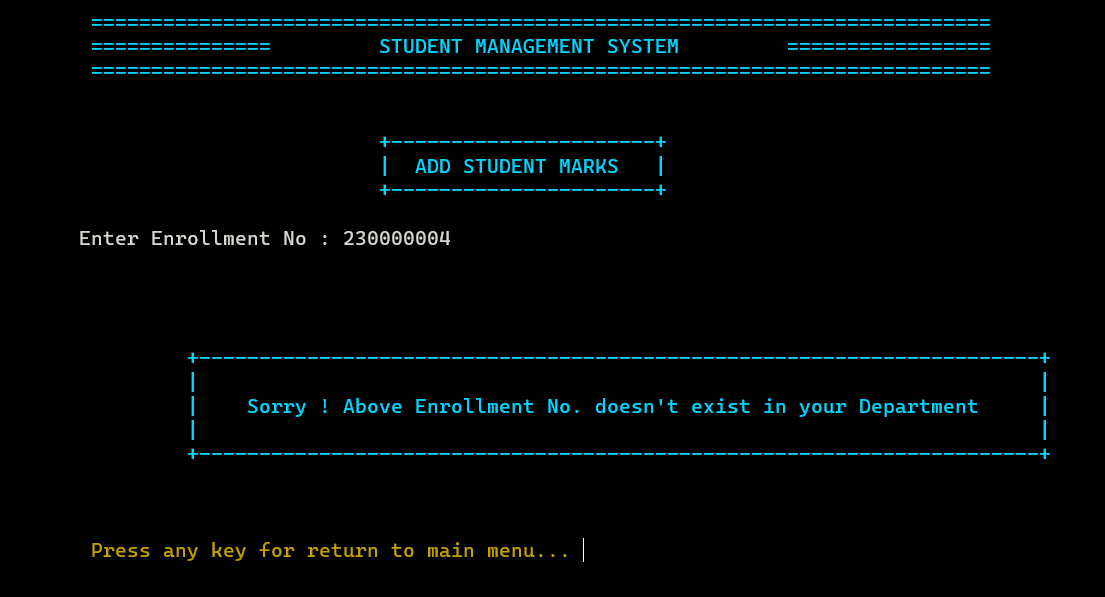
**Add Student Marks**



**Update Student Marks**

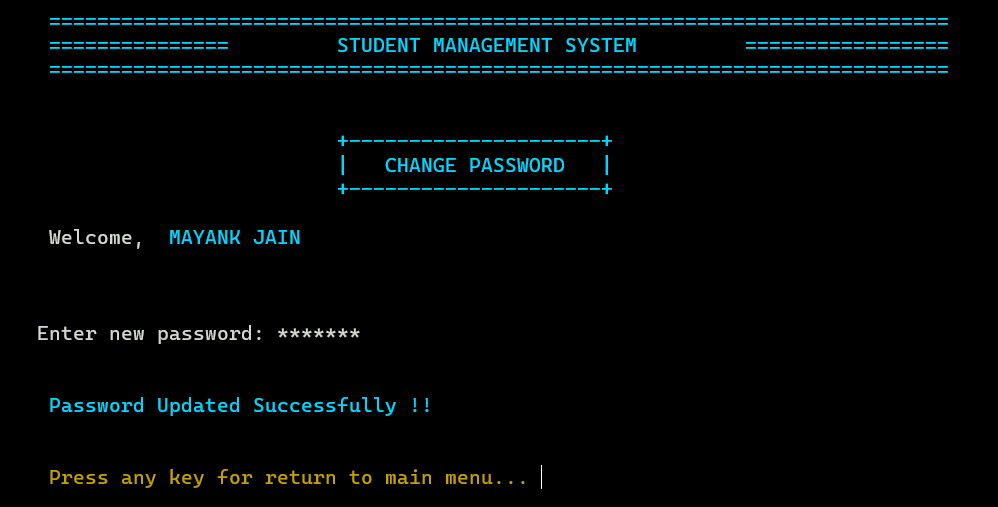


**Add Student Marks**

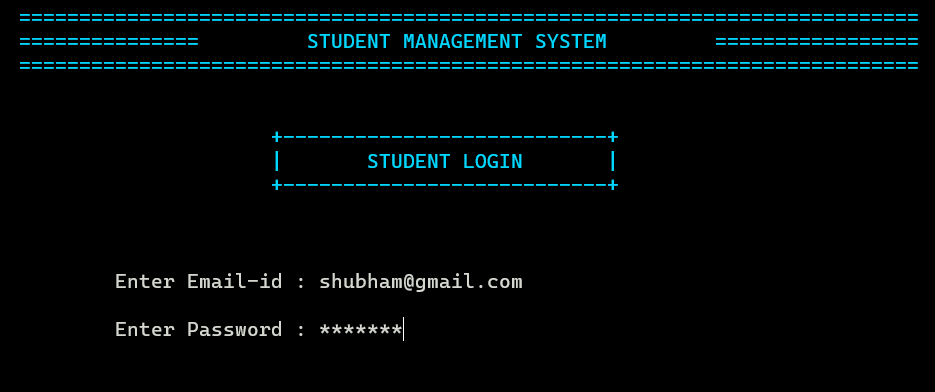


* Here Teacher can add marks of only that student which belongs to their department.

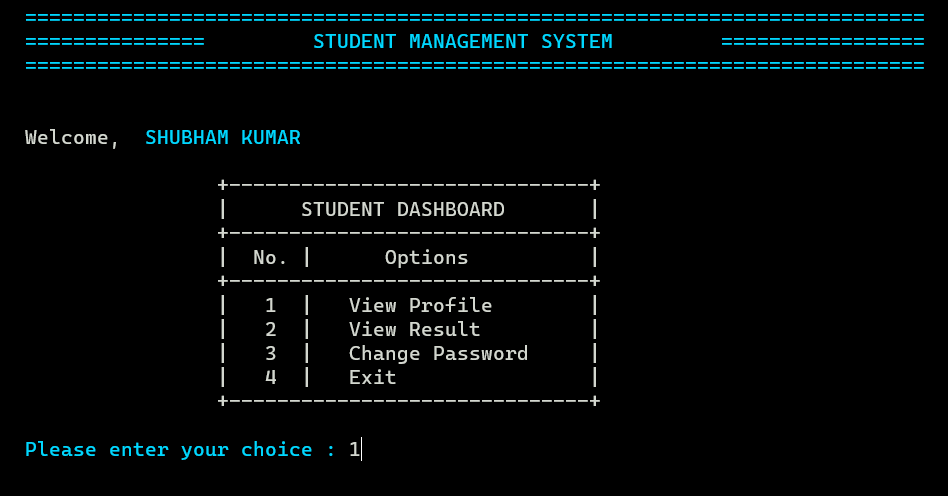
**Change Login Password**



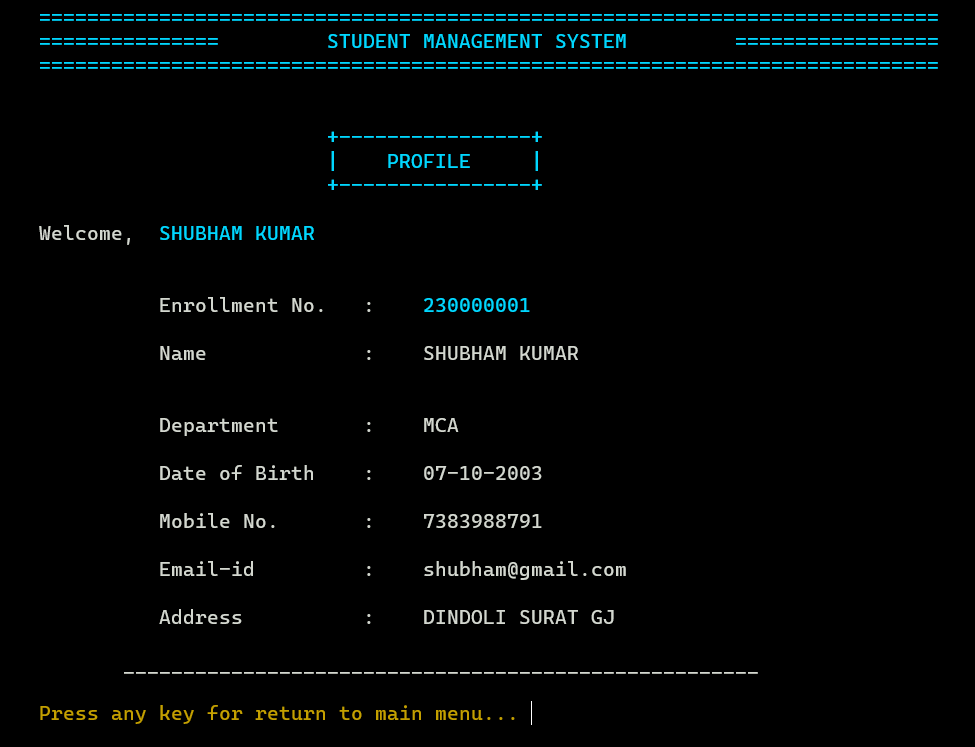
**Student Login**



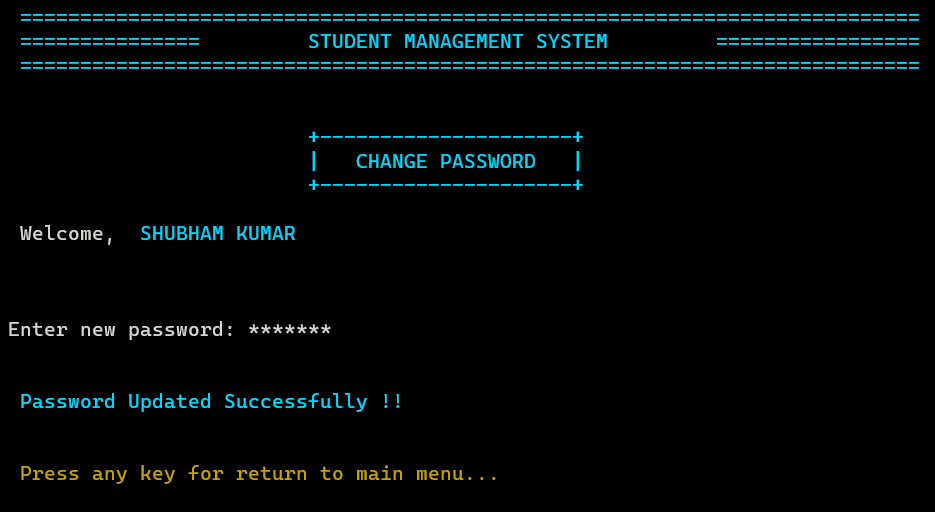
**Student Dashboard**



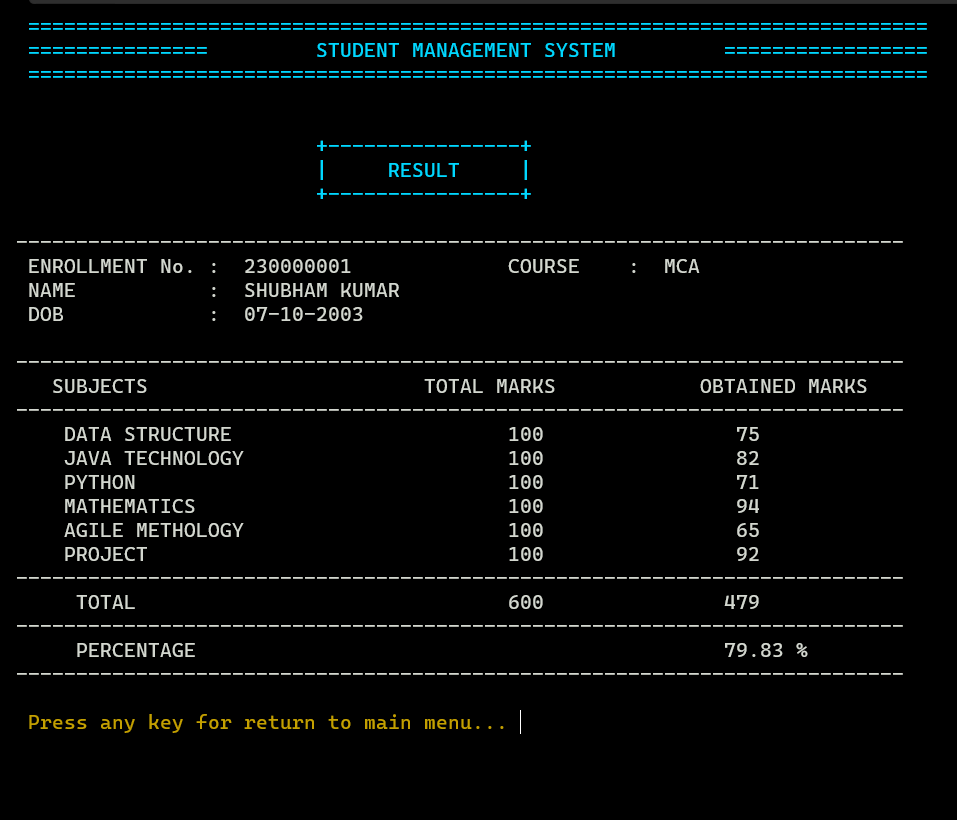
**Student Profile**



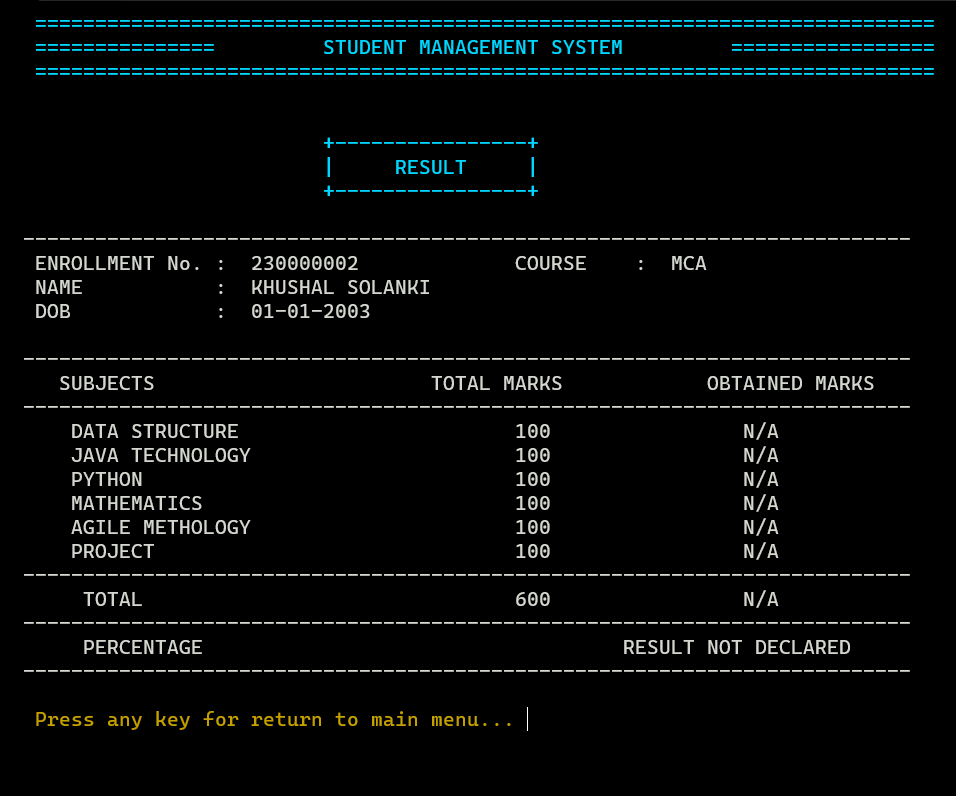
**Change Login Password**



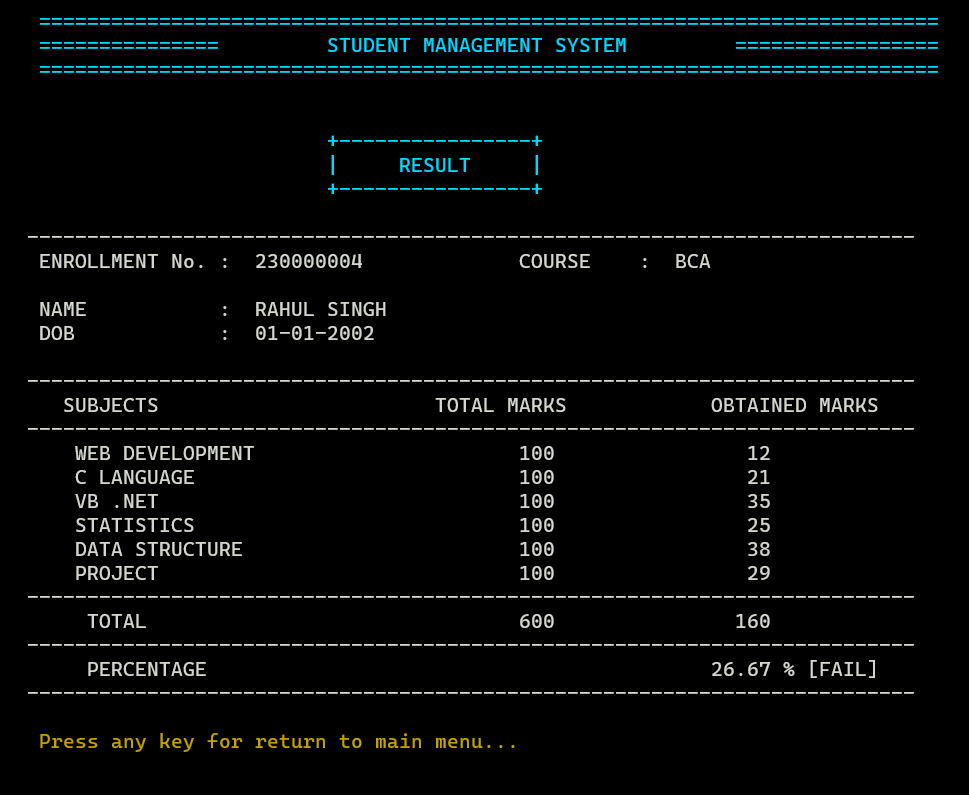
**View Result**



**View Result**



**View Result**



**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>