# K L UNIVERSITY

**FRESHMAN ENGINEERING DEPARTMENT**

## A Project Based Lab Report On

### Student Management System

### SUBMITTED BY:

ID NUMBER NAME

2300030244 GURIVI REDDY YASASWINIRIVI REDDY YASASWIN 2300030570 RACHAPUDI SAI SRE E

**UNDER THE ESTEEMED GUIDANCE OF**

**TANYA GANGULY**

**ASSOCIATE PROFESSOR**



### KL UNIVERSITY

Green fields, Vaddeswaram – 522 502 Guntur Dt., AP, India.

# DEPARTMENT OF BASIC ENGINEERING SCIENCES

# 

### CERTIFICATE

This is to certify that the project based laboratory report entitled “STUDENT MANAGEMENT SYSTEM BY LINKED LIST” submitted by GIRIVI REDDY YASASWINI bearing Regd. No. 2300030244, 2300030570 to the **Department of COMPUTER SICENCE ENGINEERING , KL University** in partial fulfillment of the requirements for the completion of a project based Laboratory in “DATA STRUCTURES(23SC1202)” course in I B Tech II Semester, is a Bonafede record of the work carried out by him/her under my supervision during the academic year 2023 – 2024.

PROJECT SUPERVISOR HEAD OF THE DEPARTMENT

TANYA GANGULY Dr. D. HARITHA

# ACKNOWLEDGEMENTS

It is great pleasure for us to express my gratitude to our honorable President.

**Sri. Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project-based laboratory report.

We express sincere gratitude to our Director **Dr. A. Jagdeesh** for his administration towards our academic growth.

We express sincere gratitude to our Coordinator and HOD-BES **Dr. D. Haritha** for her leadership and constant motivation provided in successful completion of our academic semester. We record it as my privilege to deeply thank you for providing us with the efficient faculty and facilities to make our ideas into reality.

We express my sincere thanks to our project supervisor TANYA GANGULYfor her novel association of ideas, encouragement, appreciation, and intellectual zeal which motivated us to venture this project successfully.

Finally, we are pleased to acknowledge our indebtedness to all those who devoted themselves directly or indirectly to making this project a success.

**Name:** GIRIVI REDDY YASASWINI RACHAPUDI SAI SREERIVI

**ID No:** 2300030244

2300030570

**ABSTRACT**

This project aims to develop a Student Management System for efficiently managing student information within a department. The system utilizes dynamic memory allocation and linked lists to handle student records effectively. Each student record consists of fields such as first name, last name, course code, and grade.

The Student Management System offers the following functionalities:

1. Insertion of Student Record: Enables the addition of new student records to the system.

2. Searching for Student Record: Facilitates the retrieval of student records based on specified criteria.

3. Updating of Student Record: Allows for the modification of existing student records with updated information.

4. Deletion of Student Record: Provides the capability to remove student records from the system.

Efficient management of student information is crucial for academic institutions, and this system streamlines the process by providing a structured approach to handle student data effectively.

In addition to these core functionalities, the Student Management System ensures data integrity and security by implementing appropriate access controls. It provides administrators with the ability to authenticate users and assign appropriate privileges, thus safeguarding sensitive student information from unauthorized access. Furthermore, the system incorporates error handling mechanisms to gracefully manage unexpected scenarios, ensuring robustness and reliability in operation.

By leveraging modern software development techniques and technologies, this Student Management System aims to enhance the efficiency and effectiveness of student information management within educational institutions, ultimately contributing to a more streamlined and organized academic environment.

# INDEX

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGE NO** |
| 1 | Introduction | 6 |
| 2 | Aim of the Project | 7 |
| 2.1 | Advantages & Disadvantages | 8 |
| 2.2 | Future Implementation | 8 |
| 3 | Software & Hardware Details | 9 |
| 4 | Data Flow Diagram | 10-11 |
| 5 | Algorithm for each module | 12 |
| 6 | Implementation | 13-15 |
| 7 | Integration and System Testing | 16-18 |
| 8 | Conclusion | 19 |

**INTRODUCTION**Welcome to our Student Information System, designed to streamline the management of student data within academic departments. This system

offers a user-friendly interface and essential functionalities for administrators to efficiently handle student records.Our Student Information System allows administrators to seamlessly add, update, display, and delete student details. With dynamic memory allocation using linked lists, the system optimizes memory usage and ensures

scalability as student data grows.Key features of our system include:• Add Student Details: Easily input student information, including first name, last name, course code, and grade.• Update Student Records: Modify existing student records with updated information as needed.• Display Student Details: Access comprehensive student information, organized and presented for easy reference.• Delete Student Records: Remove outdated or unnecessary student records from the system.Utilizing structured language and efficient programming tech niques, our system is designed for ease of learning and use. It produces efficient programs and can be compiled on various computer platforms.As a valuable addition, our system ensures data integrity and security.

Access controls are implemented to safeguard student information, and error handling mechanisms are in place to manage unexpected scenarios effectively.

# AIM

The primary objective of developing the Student Information System (SIS) is to enhance the management of student data within academic departments. This system aims to provide efficient and organized means for storing, updating, and accessing student records.Key goals of the project include:1. Efficient Data Management: The system facilitates the storage of student information in a structured format, including first and last names, course codes, and grades. Utilizing linked

lists for dynamic memory allocation ensures optimal resource utilization and scalability.2.Streamlined Operations: By automating tasks such as adding, updating, displaying, and deleting student records, the SIS reduces manual effort for administrative staff. This automation minimizes errors and improves the overall efficiency of managing student data.3.Enhanced Accessibilit: With anytime, anywhere access to student information, the system caters to the needs of academic staff, enabling quick retrieval of data as required. This accessibility promotes effective decision-making and timely responses to inquiries.4. Promotion of Academic Excellence: By providing comprehensive student records, the SIS supports academic departments in monitoring student progress and performance.

This promotes academic excellence by enabling educators to identify areas for improvement and provide targeted support to students.5.Data Security and Integrit: Ensuring the confidentiality and integrity of student data is a priority. The SIS implements robust security measures to safeguard sensitive information, preventing unauthorized access or tampering of records.In summary, the Student Information System aims to modernize and streamline the management of student data, fostering efficiency, accessibility, and security within academic institutions.

**ADVANTAGES**• Efficient Resource Utilization: Our Student Information System optimizes organizational resources by eliminating the need for paper-based record-keeping, contributing to a more eco-friendly environment.• Accuracy and Precision: The system ensures accurate and faultless billing calculations, minimizing errors in student record management.**DISADVANTAGES**

• Internet Requirement: Users need internet access to utilize the system fully, which may pose limitations in environments with restricted connectivity.

•While our Student Information System offers several advantages, such as efficient resource utilization and accurate record management, it is important to acknowledge its dependency on internet connectivity as a potential limitation**FUTURE IMPLEMENTATION**Advanced Student Interaction: Future versions of our Student Information System will prioritize enhancing student interaction and engagement. Features like personalized student profiles, messaging systems for communication between students and faculty, and interactive dashboards displaying academic progress and upcoming events will be implemented to foster a more dynamic learning environment.

# SYSTEM REQUIREMENTS

## SOFTWARE REQUIREMENTS:

The major software requirements of the project are as follows: Language: Turbo-C.

Operating system**:** Windows 10 or more.

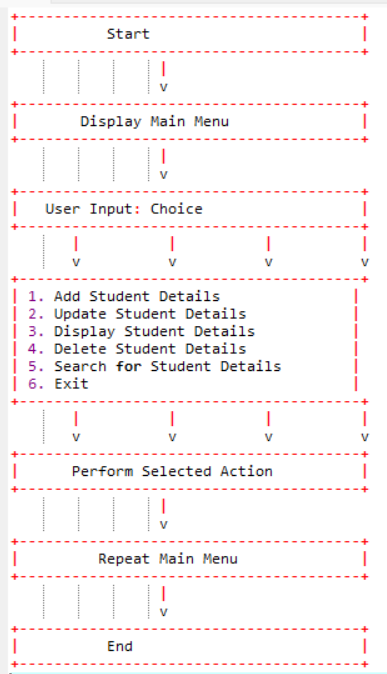
Technical requirements: Monitor, CPU, Keyboard, Mouse, etc...

## HARDWARE REQUIREMENTS:

The hardware requirements that map towards the software are as follows: RAM: 8 GB.

Processor: i5/i7.

**Flow Chart**

****

]

# ALGORITHM

1. Start

2. Read the operation to perform (add, update, display, delete, search, exit).

3. Perform the selected operation.

- Add Student Details:

- Read First Name, Last Name, Course Code, and Grade.

- Allocate memory for a new node.

- Assign input values to the node.

- Insert the node at the end of the linked list.

- Display success message.

- Update Student Details:

- Read the First Name and Second Name of the student to update.

- Traverse the linked list to find the student.

- If found, prompt for the field to update (First Name, Last Name, Course Code, or Grade).

- Update the selected field with the new value.

- Display success message.

- Display Student Details:

- If the linked list is empty, display a message indicating no student details.

- Traverse the linked list and display each student's details.

- Delete Student Details:

- Prompt for the method of deletion (by First Name or Second Name).

- Read the corresponding name to delete.

- Traverse the linked list to find the student.

- If found, delete the student node.

- Display success message.

- Search Student Details:

- Prompt for the method of search (by First Name or Second Name).

- Read the corresponding name to search.

- Traverse the linked list to find the student.

- If found, display the student's details.

- If not found, display a message indicating the student was not found.

- Exit: Terminate the program.

4. Repeat from step 2 until the user chooses to exit.

5. Stop

# IMPLEMENTATION

# 

# A screenshot of a computer program Description automatically generated

# A screenshot of a computer Description automatically generated

# A screenshot of a computer program Description automatically generated

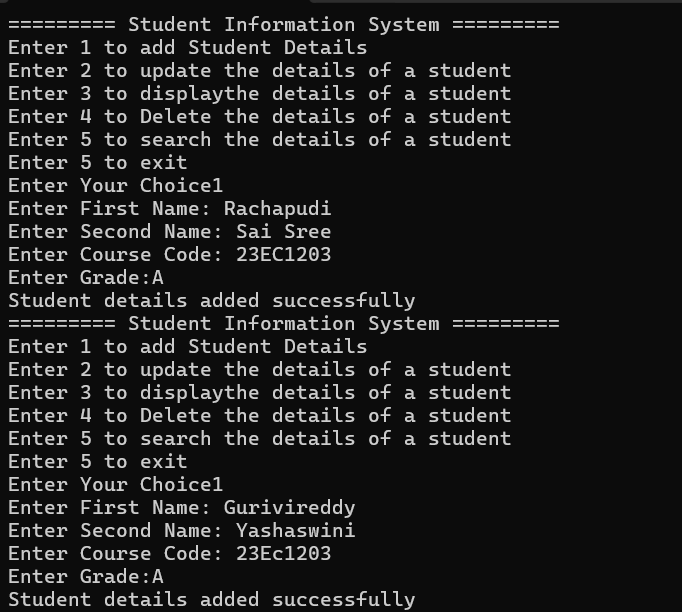
# A computer screen shot of a code Description automatically generated

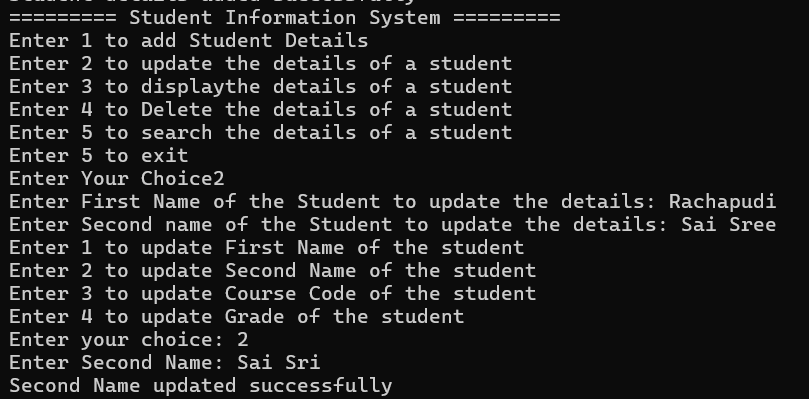
# A screenshot of a computer program Description automatically generated

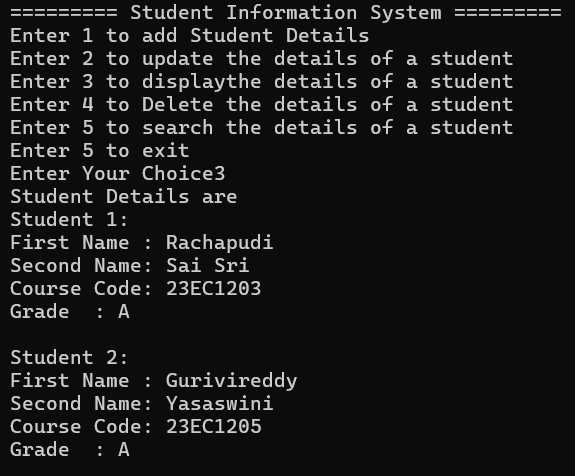
# A screenshot of a computer program Description automatically generated

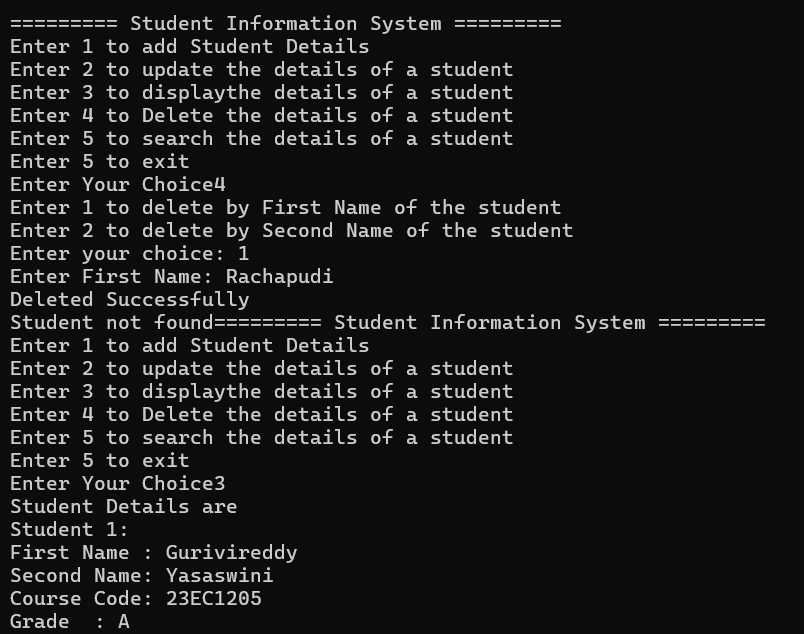
**INTEGRATION AND SYSTEM TESTING**

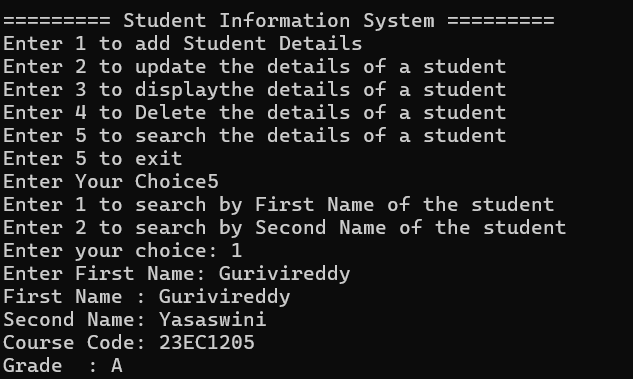
**OUTPUTS**

**Screen Shots:**









# CONCLUSION

Conclusion:

1. Modular Structure:

- The code is well-structured and organized into functions for adding, updating, displaying, and deleting student records. This modular approach enhances readability, maintenance, and code reuse.

2. Dynamic Memory Allocation:

- The program utilizes dynamic memory allocation for creating new nodes to store student information. This allows for efficient memory usage as memory is allocated only when needed.

3. Linked List Implementation:

- The student records are managed using a linked list data structure. This choice offers flexibility in adding, updating, and deleting records, as well as efficient traversal and searching of records.

4. User Interaction:

- The program provides a user-friendly interface through which users can select various operations such as adding, updating, displaying, deleting, and searching student records. Each operation is clearly labeled and prompts the user for necessary inputs.

5. Error Handling:

- The code includes basic error handling mechanisms, such as checking for empty lists before deletion and searching for records that may not exist. However, further enhancements could be made to improve error detection and reporting.

6. Limited Searching:

- The search functionality currently allows searching only by first name or second name. Enhancements could be made to allow searching by other criteria such as course code or grade.

7. Input Handling:

- The code employs scanf for input, which may lead to potential buffer overflow issues. Using functions like fgets for input along with proper buffer handling can enhance the robustness of the program.

Overall, the provided code forms a solid foundation for a Student Information System, but further improvements and refinements can be made to enhance its functionality, robustness, and user experience.

19