

STUDENT MANAGEMENT SYSTEM

NAME OF THE STUDENTS :

ROLL NO. :

M.HEMANTH KUMAR

23BFA02L36

S.M.TOUHID BASHA

23BFA02L53

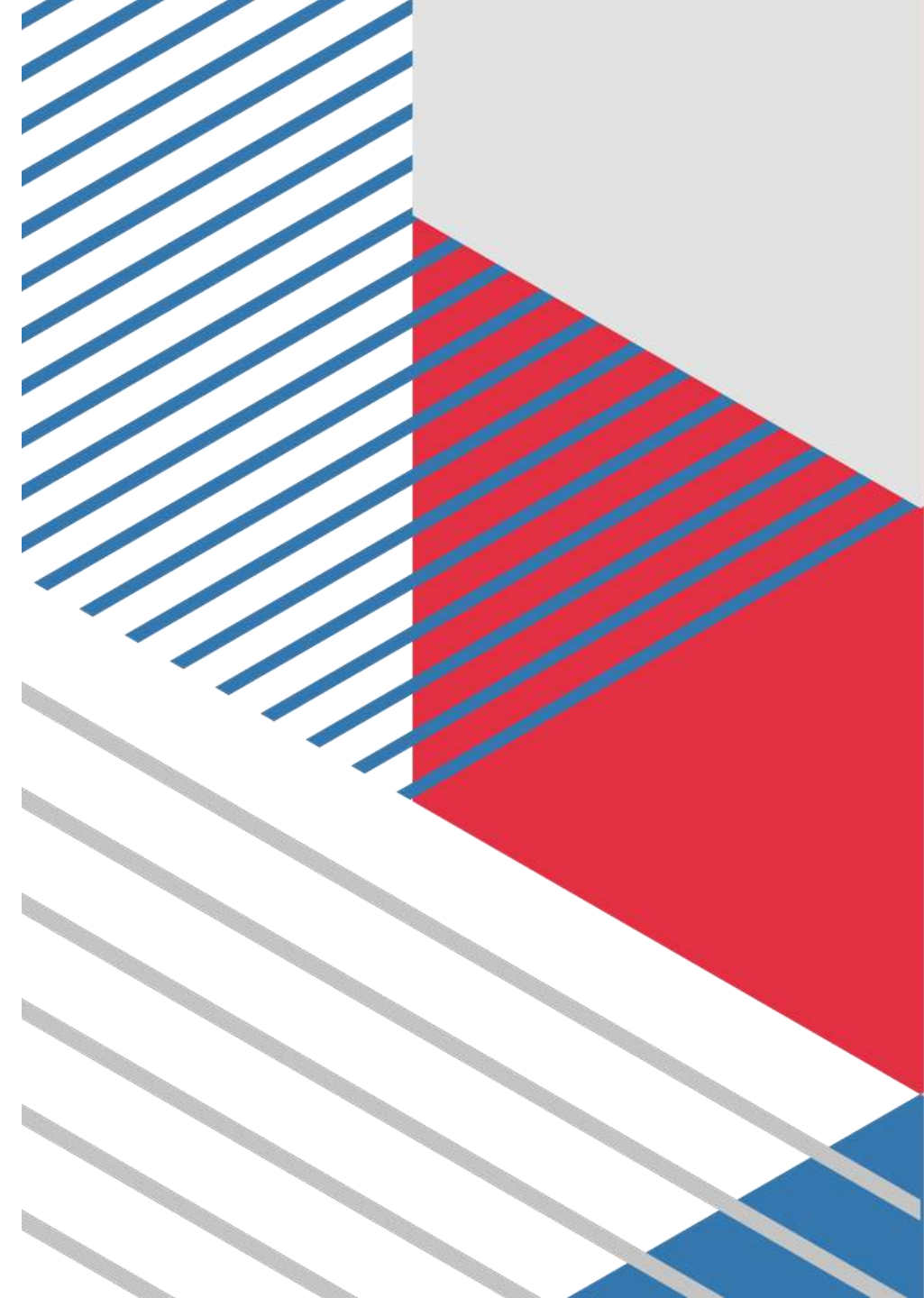


INTRODUCTION

A Student Management System (SMS) is a software application designed to manage and organize student-related data effectively and efficiently. Leveraging data structures in its implementation ensures optimized performance for storing, retrieving, and manipulating this data.

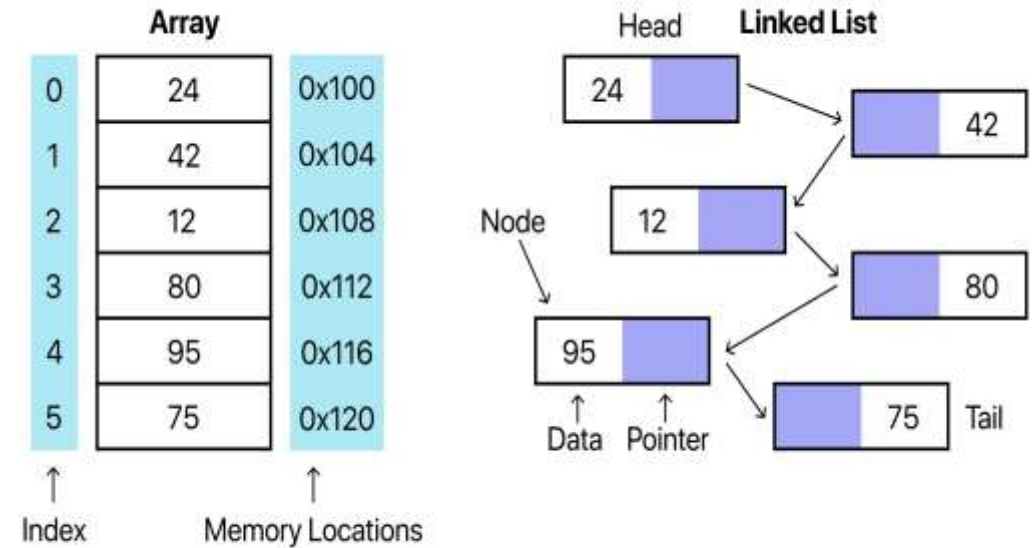
Key Features of SMS Using Data Structures

1. Student Records Storage
2. Search and Retrieval
3. sorting Data
4. Attendance Tracking
5. Grade Management



DATA STRUCTURE USED : LINKED LIST

A **linked list** is an ideal choice for implementing a Student Management System (SMS) due to its ability to dynamically adjust its size. Unlike arrays, which have a fixed size and can waste memory, linked lists allocate memory as needed for each student record. This makes them well-suited for handling fluctuating student data, especially in institutions where enrollment numbers frequently change..





CRUD(Create , Read, Update, Delete)

Create: The "Create" operation is used to add new student records to the system. This involves gathering information such as student ID, name, age, class, grades, and contact details. The data is then stored in an appropriate data structure, such as a linked list, hash table, or database.

Example:

- Adding a new student when they enroll in the institution.

READ: The "Read" operation retrieves and displays student information. Efficient searching techniques using data structures like binary search trees or hash tables ensure quick access to the data.

Example:

- Viewing the details of a specific student by their unique ID



UPDATE: The "Update" operation modifies existing student data. This could involve editing a student's grades, contact information, or any other details. Ensuring data consistency and integrity during the update process is vital.

Example:

- Updating a student's contact number.
- Changing a student's class after a promotion.

4.DELETE: The "Delete" operation removes student records from the system. This is especially useful for handling cases like student graduation, transfer, or withdrawal. Using data structures like linked lists, the deletion can be performed efficiently without disrupting the system.

Example:

- Removing a student from the system upon graduation.
- Deleting incomplete or duplicate records.

Conclusion

Provide a brief summary of your presentation.
Remind the audience what you covered in the previous slides.

Questions & answers

Invite questions from the audience.





Resources

List the resources you used for your research:

- Source #1
- Source #2
- Source #3

Thank you



THANK YOU

