# DATA STRUCTURE

# Course project update

## Topic:Student Mark Analysis System

Student Mark Analysis System

Project Update

Submitted By:

Submitted By,

**Swaliha CA ASI23CA061**

**Vikhneshwar ASI23CA062**

**Vishwesh V Shenoy ASI23CA063**

**Zayed Bin Hassan ASI23CA064**

**Amal Babu ASI23CA035**

**Project Overview**

The Student Mark Analysis System is designed to efficiently manage and analyze student marks, enabling users to add, update, search, sort, and delete student records. The system provides a structured approach to tracking academic performance, ensuring accurate data management and insightful analysis. A menu-based user interface enhances user interaction and experience.

Key Features of the System

*Student Registration:*

Add new student records with attributes such as Name, Student ID, Course, Marks, and Enrollment Date.

*Mark Update:*

Allow users to update marks for students, including adjustments for re-evaluations or extra credits.

*Sorting*:

The system can sort student records based on various attributes, such as Name, Student ID, or Marks.

*Search:*

Users can search the student database using attributes such as Name, Student ID, or Course.

*Data Saving & Loading*:

Student marks data can be saved to a text file and retrieved upon restarting the system.

**Data Structures Used**

*Linked Lists:*

Used to store student records, where each node contains the details of a student.

*Pointer Array:*

Each node's pointer is stored in an array, facilitating efficient searching and sorting operations.

**Algorithms Used**

*Linear Search:*

Utilized for searching through unsorted student records.

*Binary Search:*

Employed for searching within sorted student records.

*Merge Sort*:

Used for sorting student records based on selected attributes.

*Bubble Sort*:

Implemented for maintaining sorted order during insertion operations.

System Design

Menu

The user interface consists of a menu that offers various options, managed through switch cases. The menu continues to loop until the user opts to exit the program.

**Student Registration**

Create a new node for the student.

Add necessary data to the node.

Link the new node to the previous node.

If auto-sorting is enabled, use bubble sort to maintain order.

**Mark Update**

Search for the student record to be updated.

Update the marks as required.

If the sorted attribute changes, re-sort the array.

**Student Deletion**

Search for the student to be deleted.

Remove the student record from the list.

Update the pointer array accordingly.

**Sorting**

Select an attribute for sorting.

Use merge sort to efficiently organize the records.

**Searching**

Choose the attribute to search by.

Check if the array is sorted based on the selected attribute.

If sorted, perform a binary search.

If unsorted, use linear search.

**Data Saving**

Open the save file using file handling.

Write each student’s data to the file sequentially.

**Conclusion**

The Student Mark Analysis System aims to provide an efficient and user-friendly platform for managing and analyzing student academic performance. By utilizing appropriate data structures and algorithms, the system ensures quick data retrieval and processing, contributing to better academic management and insights for educators and students alike

