STUDENT RECORD MANAGEMENT

INTRODUCTION:

The **Student Record Management System** is a streamlined application designed to facilitate the efficient handling of student data. Built using Python and the ‘tkinter’ library, this system provides a graphical user interface (GUI) for managing key student information, including roll numbers, names, and scores. It offers functionalities like adding new student records, searching for existing ones, sorting the records by name or score, and displaying all records in a structured tabular format.

PROBLEM DOMAIN:

Managing student records can become cumbersome when handling large datasets manually. Tasks such as adding new records, searching for specific students, sorting by attributes (like name or score), and maintaining a consistent user interface require automation and systematic organization.

EXPECTED DOMAIN:

The application serves as a lightweight, interactive system for educators, administrative staff, or anyone needing a quick and efficient way to manage student data. This application is intended for educational institutions, small coaching centers, or personal use to manage student records without requiring a full-fledged database or web interface.

REQUIREMENTS:

1. **Core Functionalities**:
   - Add student records with `roll number`, `name`, and `score`.
   - Search student records using either roll number or name.
   - Sort student records by name or score (ascending order).
   - Display all student records in a tabular format
2. **User Interface**:
   - A GUI that allows users to interact with the system intuitively.
   - Text entry fields for data input and buttons for executing actions.
   - A table to visually display student records.
3. **Input Validation**:
   - Validate that roll numbers and scores are integers.
   - Ensure proper error handling and provide feedback for invalid inputs.

DATASTRUCTURES:

- List
- Dictionary
- Searching
- Sorting

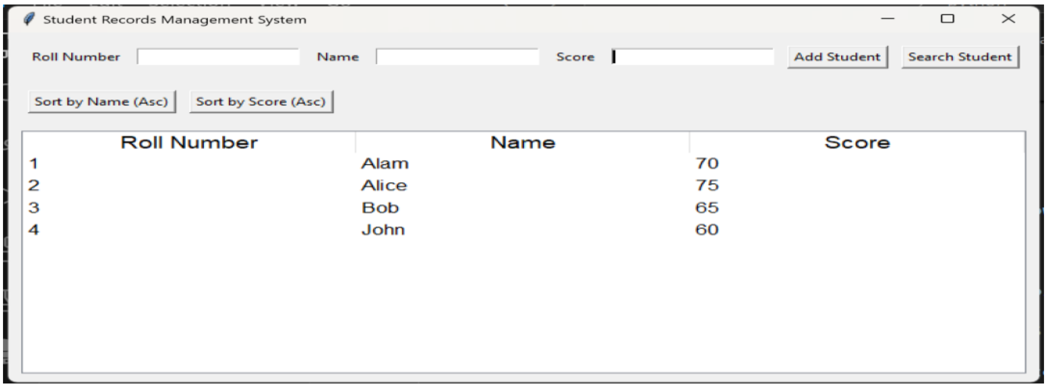
METHODOLOGY:

- **Data Storage**:
  - Use a Python list of dictionaries to store student records. Each dictionary represents a student's details.
- **GUI Components**:
  - Use tkinter frames to organize the layout.
  - Employ Entry widgets for user input and Button widgets for executing actions.
  - Implement a Treeview widget to display records in a tabular format.
- **Functions**:
  - Add_students: Adds a new student to the list and displays a success message.
  - Search_students: Searches the list for a student using roll number or name and displays the result.
  - Sort_students: Sorts the student list based on name or score.
  - Display_students : Updates the table to reflect the current student list.
- **Event Handlers**:
  - Functions like handle_add_students and handle_sort_students tie GUI elements to the underlying data manipulation logic.
- **Error Handling**:
  - Use `try-except` blocks to manage invalid inputs and ensure the program runs smoothly.

CONCLUSION:

The student records management system provides a simple and effective solution for managing small-scale student data. It combines a user-friendly interface with essential data operations like adding, searching, sorting, and displaying records

OUTPUT:



Roll Number	Name	Score
1	Alam	70
2	Alice	75
3	Bob	65
4	John	60