School Management System

Objective

The main objective of this project is to develop a School Management System that:

- Manages student information.
- Allows adding, viewing, searching, and deleting records.
- Stores data in a CSV file for persistence.
- Demonstrates basic file handling, structured data management, and simple user interaction.

:

Algorithm

1. Start
2. Initialize an array `students[]` of type Student
3. Set `studentCount` = 0

4. Load student data from `students.csv` into `students[]`
   - Open `students.csv` in read mode
   - For each line in the file:
     - Split data by commas
     - Assign values to corresponding fields of `Student`
     - Add student to `students[]`
     - Increment `studentCount`
   - Close the file

5. Display Main Menu:
   1. Add Student
   2. View Students
   3. Search Student by Roll Number
   4. Delete Student by Roll Number
   5. Exit

6. Get user input for choice

7. If choice = 1 (Add Student):
   - Check if `studentCount` < MAX_STUDENTS
     - If true, prompt user to enter student details (Roll No, Name, Age, Class, Fees Status, Parent's Name, Mobile, Address)
     - Add the student to `students[]`
     - Increment `studentCount`
     - Save data to `students.csv` (open file in write mode, write all student data)
     - Display "Student added successfully"
   - Else, display "Student limit reached"

8. If choice = 2 (View Students):
   - If `studentCount` = 0, display "No students available"
   - Else, loop through `students[]` and display each student's details

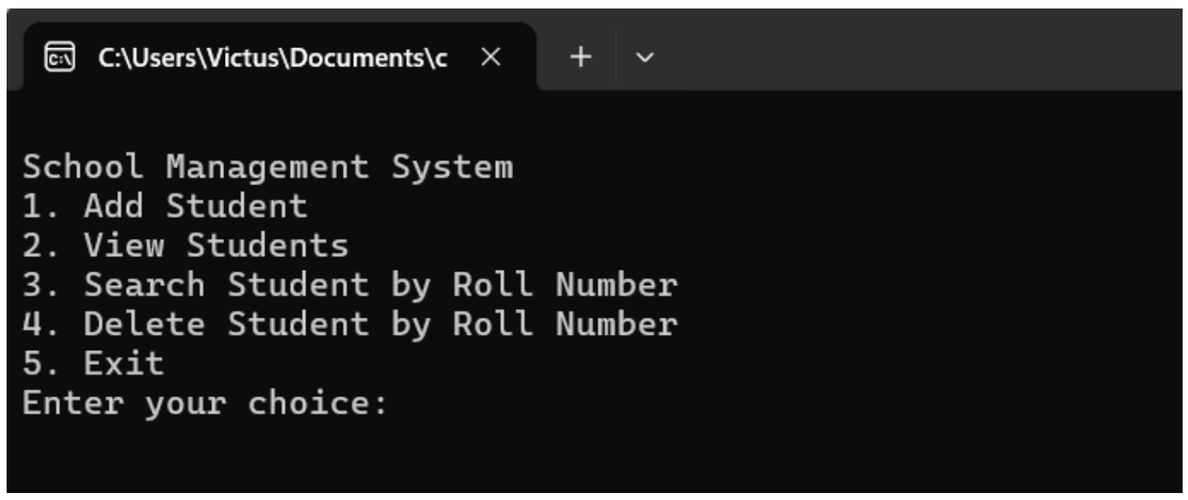
9. If choice = 3 (Search Student by Roll Number):
   - Get roll number from user
   - Loop through `students[]`:
     - If roll number matches, display student's details
     - Else, display "Student not found"

10. If choice = 4 (Delete Student by Roll Number):
    - Get roll number from user
    - Loop through `students[]`:
      - If roll number matches:
        - Shift all students after the deleted one to the left
        - Decrease `studentCount`
        - Save updated data to `students.csv`
        - Display "Student deleted"
    - Else, display "Student not found"

11. If choice = 5 (Exit):
    - Save all student data to `students.csv` (open file in write mode, write all student data)
    - Exit program

12. End

Output



Implementation:

To implement the School Management System (SMS) project in C, there are several important programming tokens and concepts you'll use. These include:

1. Data Structures (for storing student details)
2. File Handling (to read and write data to CSV)
3. Loops (for iterating through student records)
4. Conditional Statements (for making decisions like adding, searching, and deleting records)
5. Input and Output Functions (to interact with the user)

Below is a breakdown of the critical programming concepts (tokens) and how they are used in the implementation.

Conclusion

This School Management System provides a fundamental framework for managing student information. It demonstrates key concepts such as structured data, file handling, and simple CRUD operations (Create, Read, Update, Delete) in C. The system is designed to be simple yet scalable for further enhancements. It provides a reliable and easy-to-use interface for school administrators to maintain student records, with data persistence handled via CSV files.

Name:
Kashyap Thummar
Enrollment No. :
12402040501033

Name:
Rudra Kachhela
Enrollment No. :
12402040501054

Name:
Pavan Rana
Enrollment No. :
12402040501051