Student Management System

Python Project

Divyanshi Gupta 25BAI10841
Department of Computer Science
Engineering (AI & ML)
Vellore Institute of Technology
Bhopal, Madhya Pradesh
divyanshi.25bai10841@vitbhopal.ac.in

Abstract*:* this is simple student management system which is terminal based to manage data of student effectively it uses basic python programming. And it performs basic File Input output operations like add student, remove student, Update student delete student. And save this data to the text file students.txt.

Introduction

A Student Management System is a tool designed to maintain student information in an organized way. In any institution, managing records manually becomes difficult as data grows. This project provides a simple command-line window that stores student details such as name, roll number, and marks using Python Programming..

Objective

The main objectives of this project are:

- Manage student data
- Understanding input output operations using Python
- Store data permanently in a text file
- Provide a simple and user-friendly menu-driven interface
- Reduce manual workload in handling student records

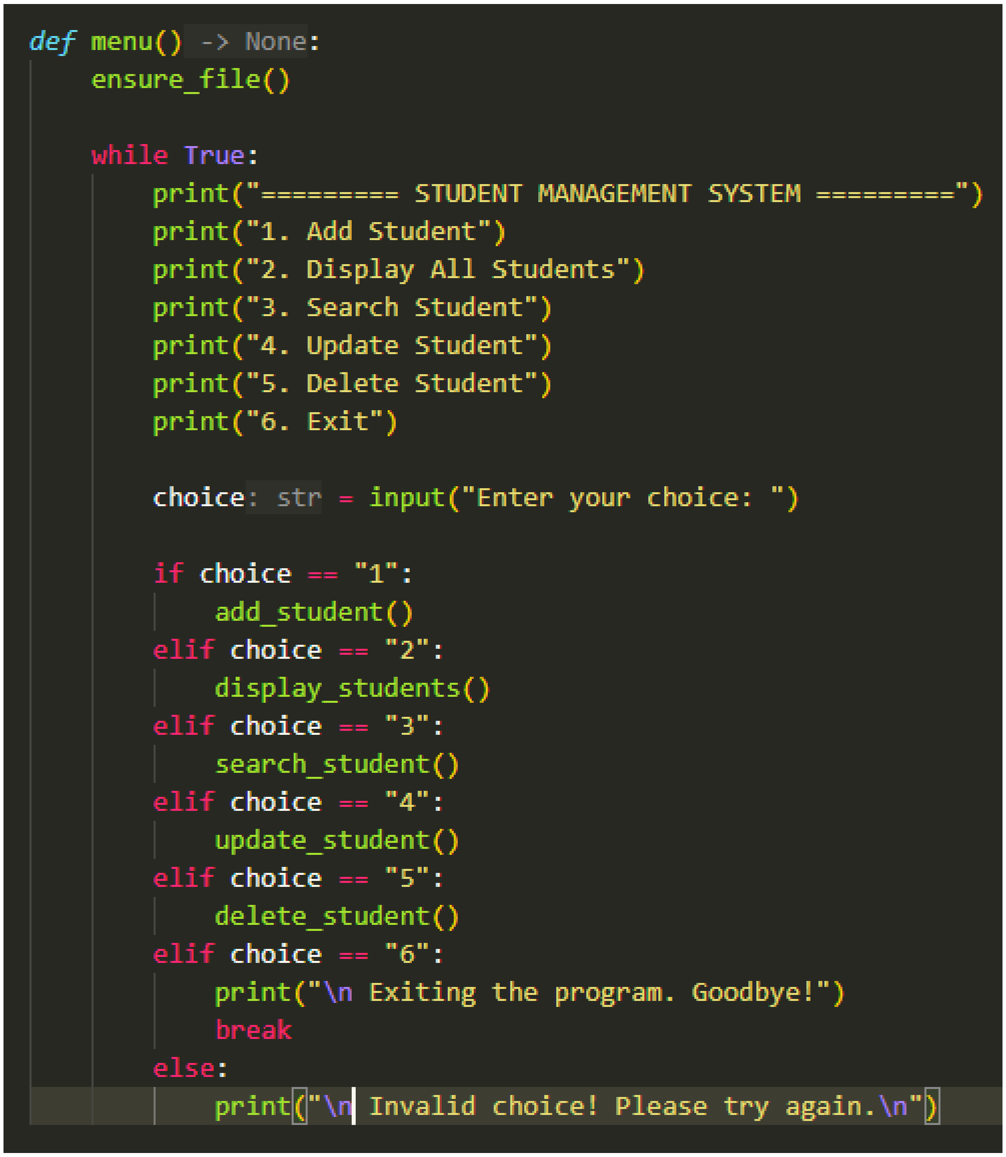
Implementation

The system works in a step-by-step process:

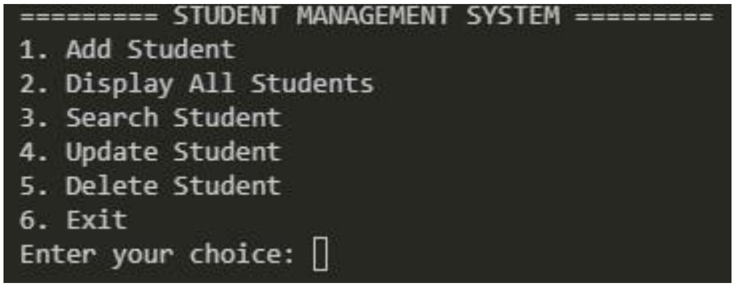
1. The program displays a menu with different options
2. The user chooses an operation (e.g., Add Student)
3. The system performs the selected operation
4. Student information is stored in the text file students.txt
5. File handling is used to read, write, update, or delete records
6. The system continues to run until the user exits

Working

- Main Menu

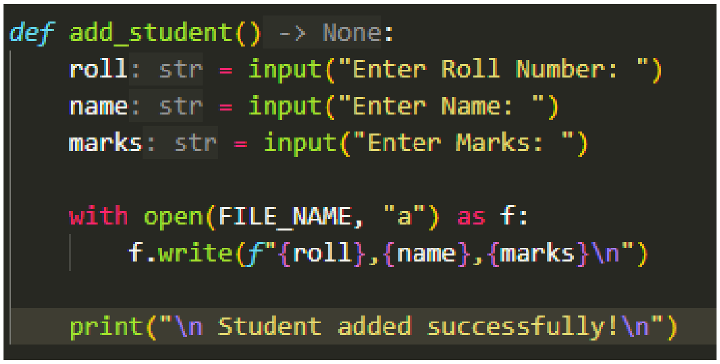


```
def menu() -> None:
    ensure_file()

    while True:
        print("========= STUDENT MANAGEMENT SYSTEM =========")
        print("1. Add Student")
        print("2. Display All Students")
        print("3. Search Student")
        print("4. Update Student")
        print("5. Delete Student")
        print("6. Exit")

        choice: str = input("Enter your choice: ")

        if choice == "1":
            add_student()
        elif choice == "2":
            display_students()
        elif choice == "3":
            search_student()
        elif choice == "4":
            update_student()
        elif choice == "5":
            delete_student()
        elif choice == "6":
            print("\n Exiting the program. Goodbye!")
            break
        else:
            print("\n Invalid choice! Please try again.\n")
```

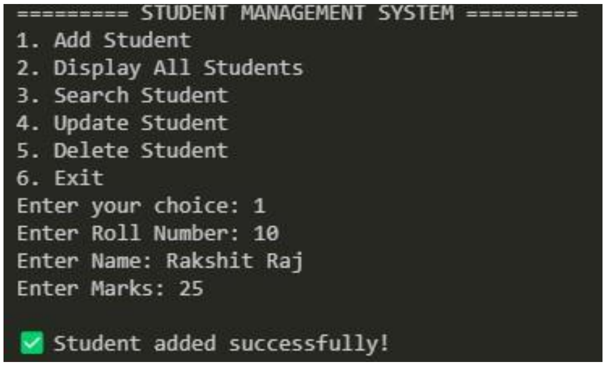


```
========= STUDENT MANAGEMENT SYSTEM =========
1. Add Student
2. Display All Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice: 
```

- Add new student records
    - This adds new student

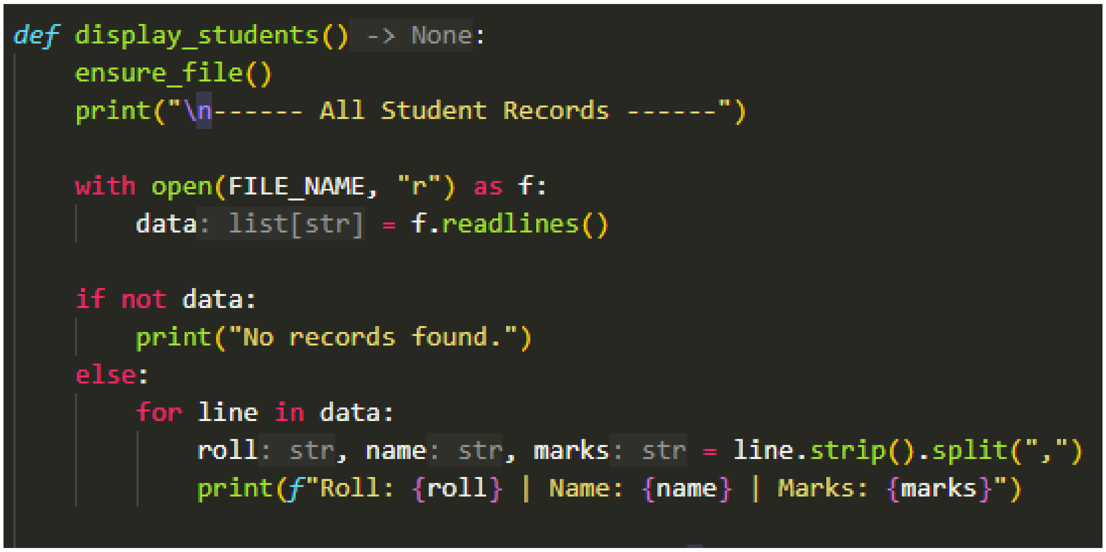


```python
def add_student() -> None:
    roll: str = input("Enter Roll Number: ")
    name: str = input("Enter Name: ")
    marks: str = input("Enter Marks: ")

    with open(FILE_NAME, "a") as f:
        f.write(f"{roll},{name},{marks}\n")

    print("\n Student added successfully!\n")
```

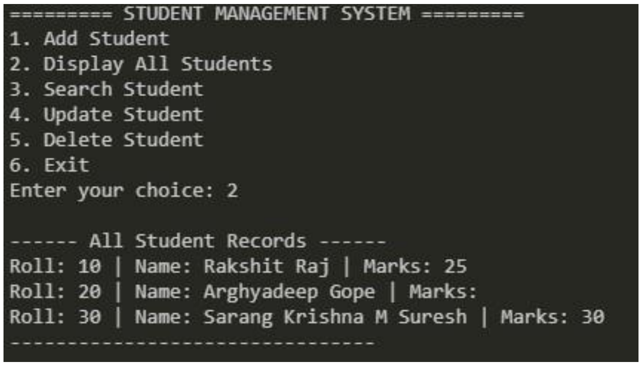


```
========= STUDENT MANAGEMENT SYSTEM =========
1. Add Student
2. Display All Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice: 1
Enter Roll Number: 10
Enter Name: Rakshit Raj
Enter Marks: 25

✅ Student added successfully!
```

- Search student by roll number
    - This Function Search the name of student

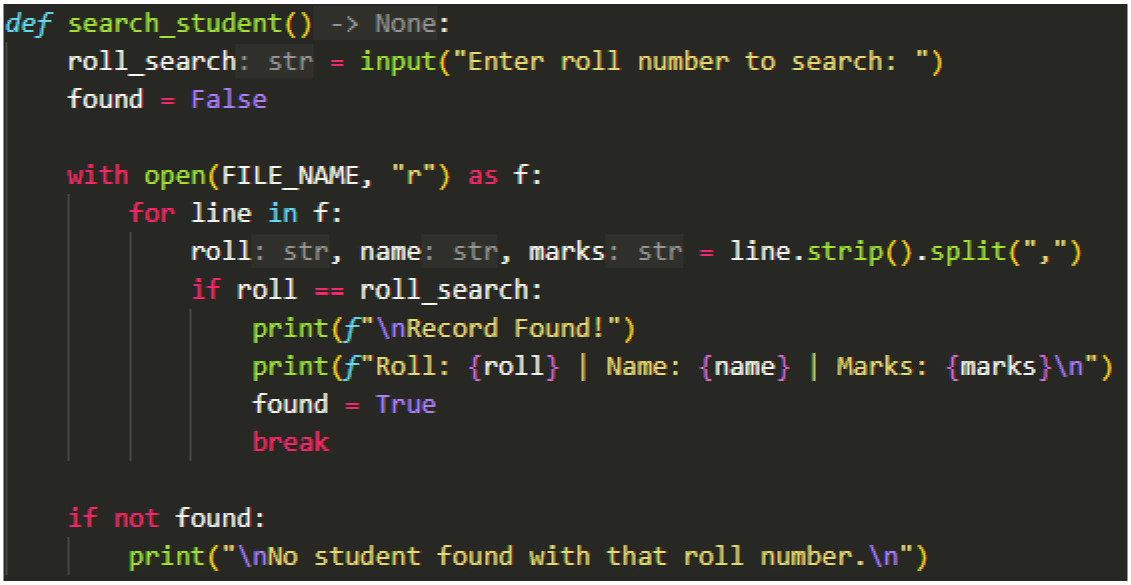


```python
def display_students() -> None:
    ensure_file()
    print("\n------ All Student Records ------")

    with open(FILE_NAME, "r") as f:
        data: list[str] = f.readlines()

    if not data:
        print("No records found.")
    else:
        for line in data:
            roll: str, name: str, marks: str = line.strip().split(",")
            print(f"Roll: {roll} | Name: {name} | Marks: {marks}")
```

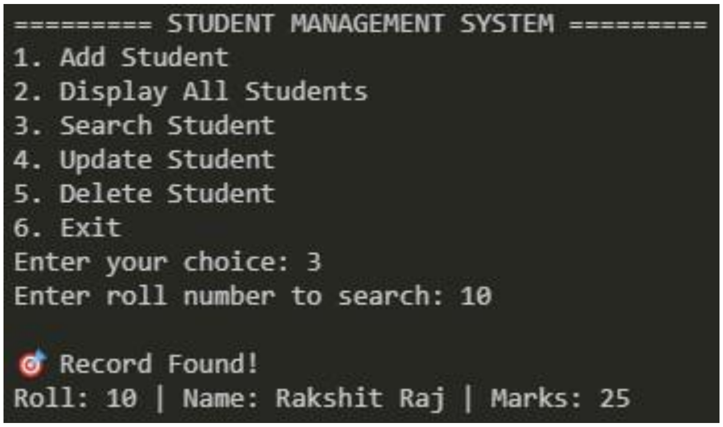


```
========= STUDENT MANAGEMENT SYSTEM =========
1. Add Student
2. Display All Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice: 2

------ All Student Records ------
Roll: 10 | Name: Rakshit Raj | Marks: 25
Roll: 20 | Name: Arghyadeep Gope | Marks:
Roll: 30 | Name: Sarang Krishna M Suresh | Marks: 30
--------------------------------
```

- Update student details
    - This function updates the student

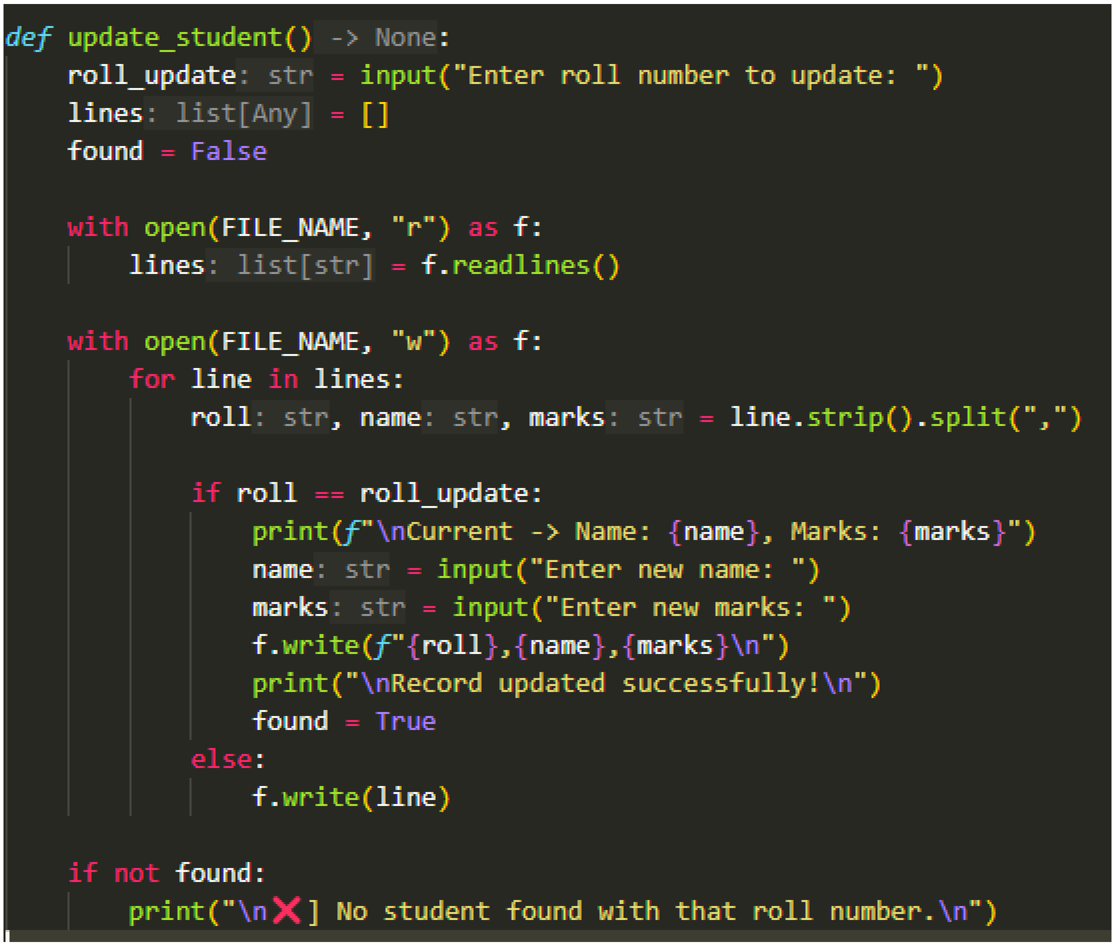


```python
def search_student() -> None:
    roll_search: str = input("Enter roll number to search: ")
    found = False

    with open(FILE_NAME, "r") as f:
        for line in f:
            roll: str, name: str, marks: str = line.strip().split(",")
            if roll == roll_search:
                print(f"\nRecord Found!")
                print(f"Roll: {roll} | Name: {name} | Marks: {marks}\n")
                found = True
                break

    if not found:
        print("\nNo student found with that roll number.\n")
```

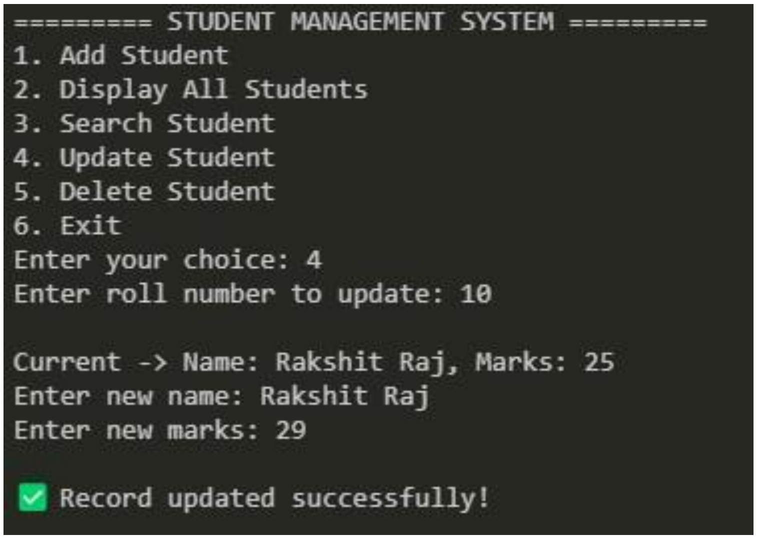


```
========= STUDENT MANAGEMENT SYSTEM =========
1. Add Student
2. Display All Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice: 3
Enter roll number to search: 10

🎯 Record Found!
Roll: 10 | Name: Rakshit Raj | Marks: 25
```

- Delete student record
    - This function delete the record of student

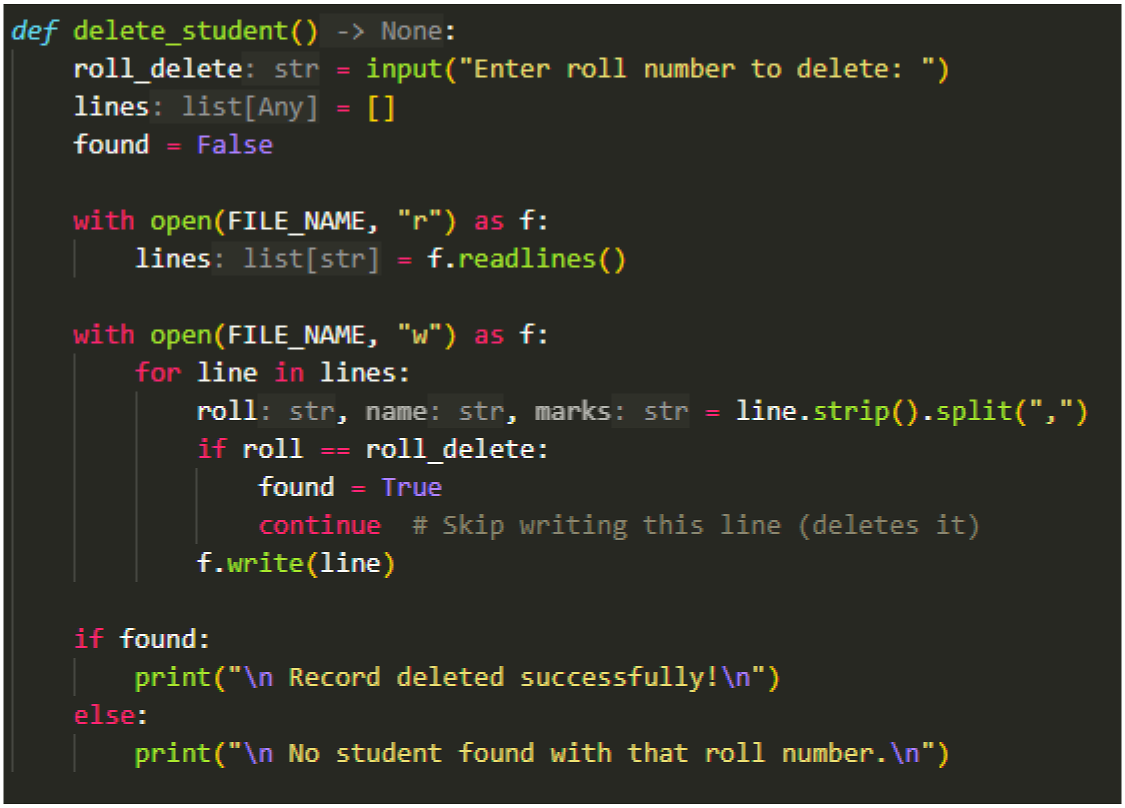


```
def update_student() -> None:
    roll_update: str = input("Enter roll number to update: ")
    lines: list[Any] = []
    found = False

    with open(FILE_NAME, "r") as f:
        lines: list[str] = f.readlines()

    with open(FILE_NAME, "w") as f:
        for line in lines:
            roll: str, name: str, marks: str = line.strip().split(",")

            if roll == roll_update:
                print(f"\nCurrent -> Name: {name}, Marks: {marks}")
                name: str = input("Enter new name: ")
                marks: str = input("Enter new marks: ")
                f.write(f"{roll},{name},{marks}\n")
                print("\nRecord updated successfully!\n")
                found = True
            else:
                f.write(line)

    if not found:
        print("\n❌] No student found with that roll number.\n")
```

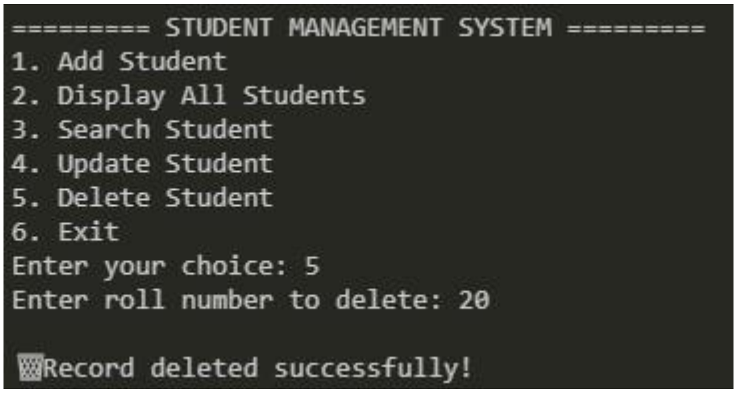


```
========= STUDENT MANAGEMENT SYSTEM =========
1. Add Student
2. Display All Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice: 4
Enter roll number to update: 10

Current -> Name: Rakshit Raj, Marks: 25
Enter new name: Rakshit Raj
Enter new marks: 29

✅ Record updated successfully!
```

- Display all records
    - For displaying all records

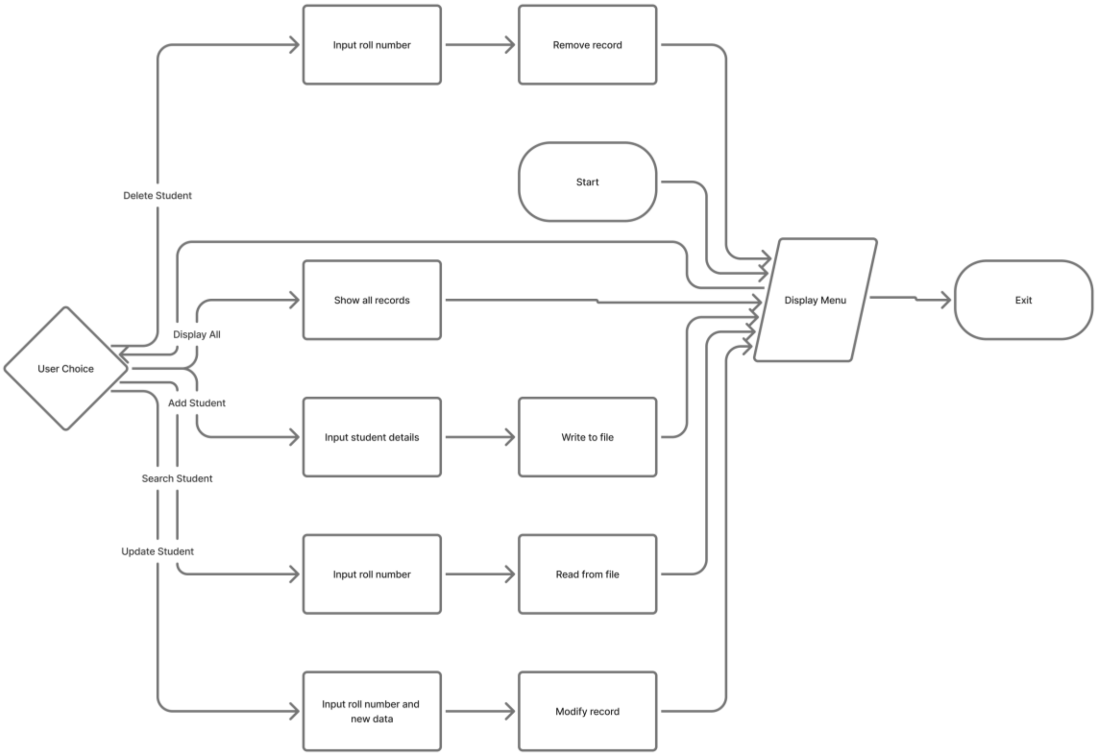


```python
def delete_student() -> None:
    roll_delete: str = input("Enter roll number to delete: ")
    lines: list[Any] = []
    found = False

    with open(FILE_NAME, "r") as f:
        lines: list[str] = f.readlines()

    with open(FILE_NAME, "w") as f:
        for line in lines:
            roll: str, name: str, marks: str = line.strip().split(",")
            if roll == roll_delete:
                found = True
                continue  # Skip writing this line (deletes it)
            f.write(line)

    if found:
        print("\n Record deleted successfully!\n")
    else:
        print("\n No student found with that roll number.\n")
```



```
========= STUDENT MANAGEMENT SYSTEM =========
1. Add Student
2. Display All Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice: 5
Enter roll number to delete: 20

🗑Record deleted successfully!
```

FlowChart



Conclusions

The Student Management System hence successfully achieves its objectives by providing a simple way to manage student records. The system automates the basic operations, reducing the manual effort, and ensures accurate information. This project also helps beginners understand the concepts of Python programming and file handling in a real-world scenario.

Reference

- Python Documentation
- Class Notes
- Online Tutorials
- W3Schools / GeeksForGeeks