PROJECT TITLE

STUDENT MANAGEMENT SYSTEM

PROJECT REPORT

Submitted by

S SATHISH

Subject

Python Programming

Submitted to

Mr. Rohan

Institute Name

Besant Technologies

Year

2025

Place

Bangalore, Karnataka

PROJECT OVERVIEW

The Student Management System is a simple Python program that runs in the console.
It allows us to add, view, search, update, and delete student records.
At first, the data is stored only while the program is running, but we can also use JSON files to save the data so it is not lost when the program is closed.

PREREQUISITES

- Python 3.8 or higher installed
- Any code editor like VS Code or PyCharm
- (Optional) Git for version control

PROJECT STRUCTURE

- **Student_mgmt.py** → Main program file
- **Data. Json** → File used to store student data (created automatically when program runs)
- **README.md** → Documentation file

DATA MODEL

Each student record has the following fields:

- Roll number (unique)
- Name
- Grade
- Age (optional)

Example of one record:

{"roll no": "283", "name": "Sathish", "grade": "A", "age": "21”}

All students are stored in a list called students.

FEATURES

- Add new student (with validation so roll number is unique)
- View all students in a proper table format
- Search student by roll number or name
- Update student details (can skip fields to keep old values)
- Delete student (with confirmation)
- Save and load data using JSON so records remain after closing the program

HOW TO RUN THE PROGRAM

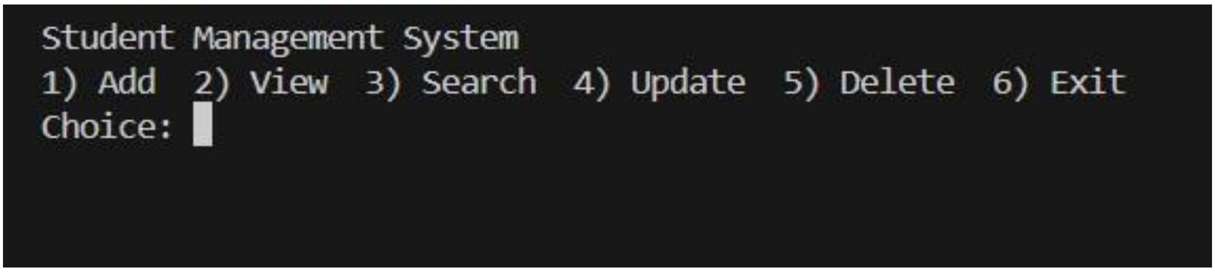
1. Open the project folder in terminal or command prompt.
2. Run the program using:
3. python student_mgmt.py
4. The menu will appear with options:
5. 1) Add
6. 2) View
7. 3) Search
8. 4) Update
9. 5) Delete
10. 6) Exit
11. Choose the required option and follow the instructions on the screen.

TESTING CHECKLIST

- Add a student and then view the list → record should appear.
- Try adding a student with the same roll number → should be rejected.
- Search for a roll number that exists → record should be shown.
- Update a record and press Enter to keep some fields unchanged.
- Delete a record and confirm → record should be removed.
- Close the program and reopen → records should still be there (if JSON is used).

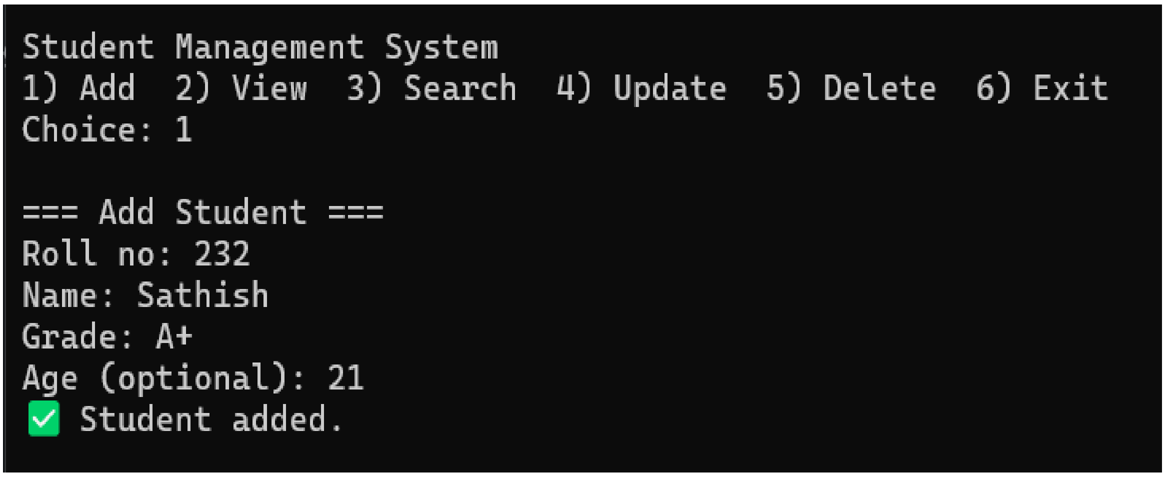
SCREENSHOTS:

MENU WITH OPTIONS:



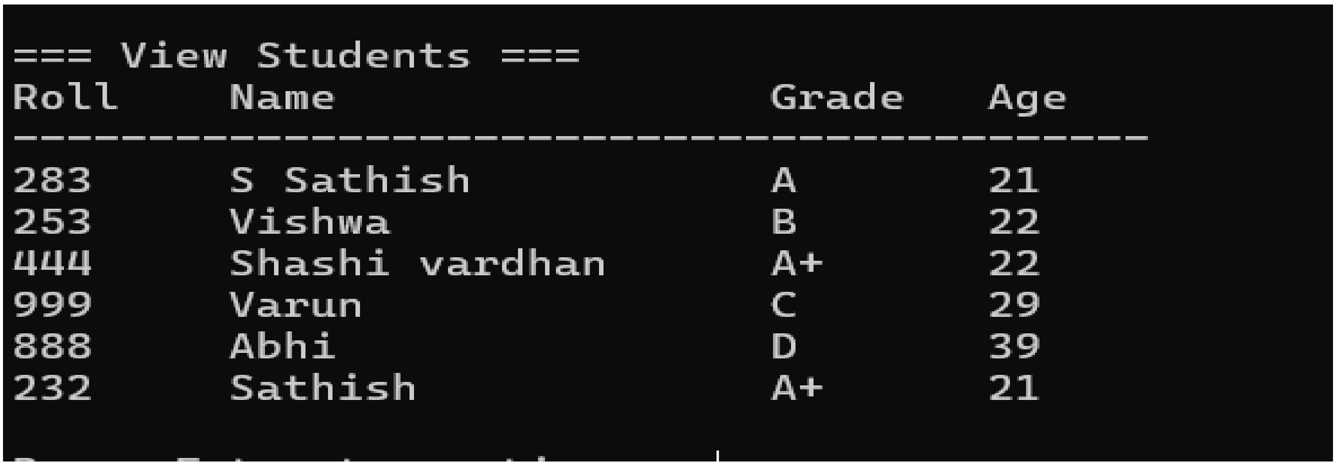
```
Student Management System
1) Add  2) View  3) Search  4) Update  5) Delete  6) Exit
Choice: 
```

ADDING A STUDENT:



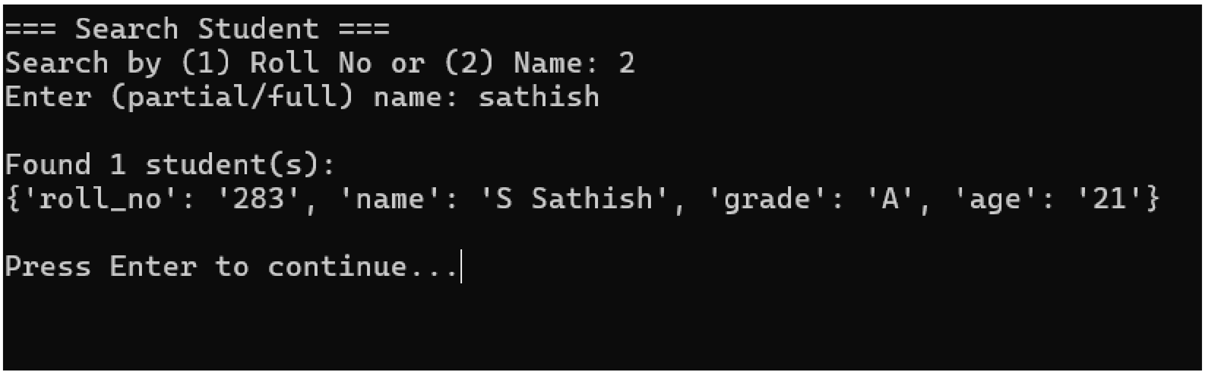
```
Student Management System
1) Add  2) View  3) Search  4) Update  5) Delete  6) Exit
Choice: 1

=== Add Student ===
Roll no: 232
Name: Sathish
Grade: A+
Age (optional): 21
✅ Student added.
```

VIEWING STUDENT LIST:



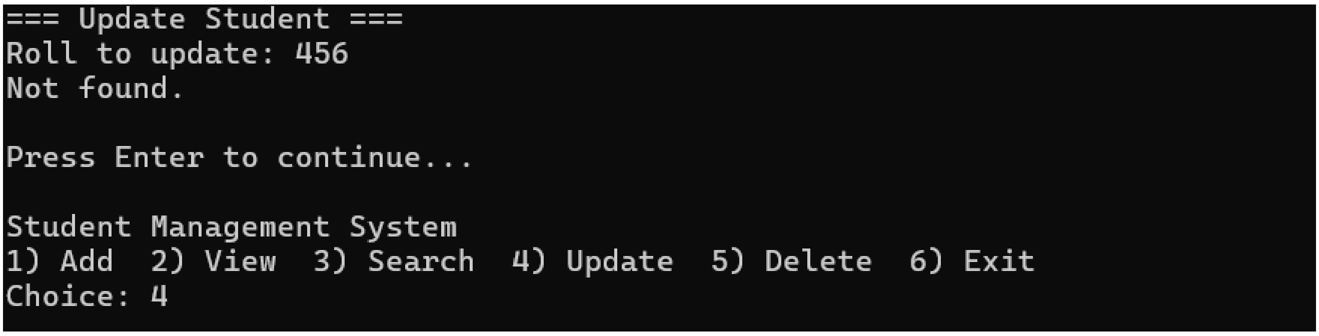
```
=== View Students ===
Roll       Name                Grade    Age
-------------------------------------------
283        S Sathish           A        21
253        Vishwa              B        22
444        Shashi vardhan      A+       22
999        Varun               C        29
888        Abhi                D        39
232        Sathish             A+       21
```

SEARCHING A RECORD:



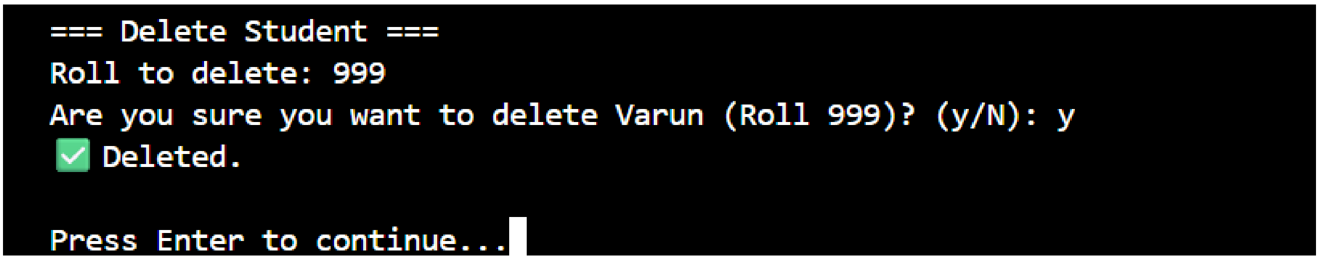
```
=== Search Student ===
Search by (1) Roll No or (2) Name: 2
Enter (partial/full) name: sathish

Found 1 student(s):
{'roll_no': '283', 'name': 'S Sathish', 'grade': 'A', 'age': '21'}

Press Enter to continue...
```

UPDATING A RECORD:



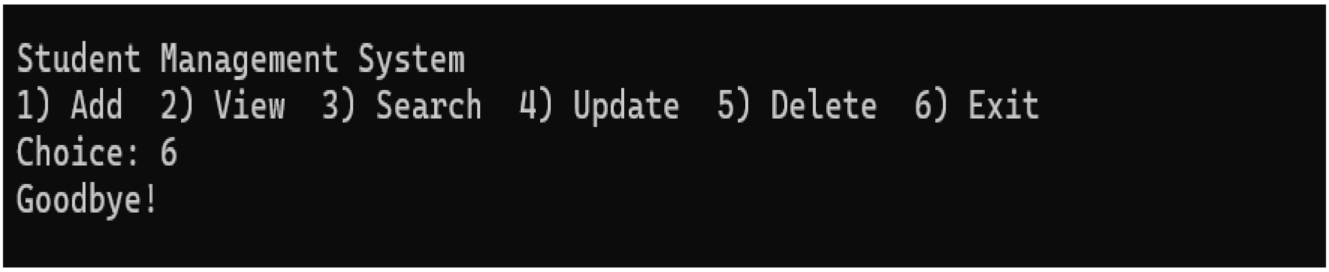
```
=== Update Student ===
Roll to update: 456
Not found.

Press Enter to continue...

Student Management System
1) Add  2) View  3) Search  4) Update  5) Delete  6) Exit
Choice: 4
```

DELETING A STUDENT:



```
=== Delete Student ===
Roll to delete: 999
Are you sure you want to delete Varun (Roll 999)? (y/N): y
✅ Deleted.

Press Enter to continue...
```

EXIT:



```
Student Management System
1) Add  2) View  3) Search  4) Update  5) Delete  6) Exit
Choice: 6
Goodbye!
```

What I Learned

- How to use lists and dictionaries in Python
- How to make menu-based programs
- Input validation and error handling
- Reading and writing data with JSON files
- Building a small but complete project from scratch

Future Improvements

- Allow searching by partial name
- Add sorting options (by roll number or name)
- Add a simple web version (Flask)