

PROJECT TITLE

STUDENT MANAGEMENT SYSTEM

PROJECT REPORT

Submitted by

Ganta Naga Venkata Sowmya

Subject

Python Programming

Submitted to

Mr. Rohan

Institute Name

Besant Technologies

Year

2025

Place

Bangalore, Karnataka

GitHub

Project Description

The Student Management System is a Python-based application designed to keep track of student records efficiently. It provides options to add, update, search, view, and delete student details in an easy-to-use console interface. To prevent data loss, the program also supports saving and retrieving information using JSON files.

Requirements

- Python 3.8 or later
 - Any IDE or text editor (VS Code, PyCharm, etc.)
 - (Optional) Git for version control
-

Project Files

- **student_mgmt.py** → Main source code file
 - **data.json** → Stores all student records
 - **README.md** → Documentation and usage guide
-

Data Structure

Each student is stored as a dictionary with the following fields:

- Roll Number (unique identifier)
- Name
- Grade
- Age (optional)

Example of one record:

```
{"roll no": 501, "name": "Sowmya", "grade": "A", "age": 22}
```

All records are maintained in a list called *students*.

Key Features

- Add new student with unique roll number validation
- Display all records in a structured table format

- Search student details using roll number or name
 - Update existing records (with option to retain old values)
 - Delete records with confirmation
 - Save/load data using JSON to maintain persistence
-

How to Run

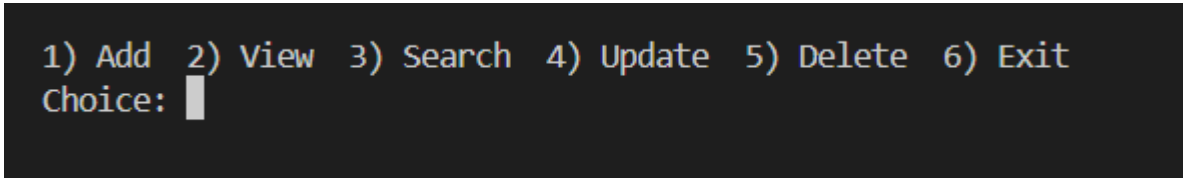
1. Open terminal inside the project folder
 2. Run:
 3. `python student_mgmt.py`
 4. Select an option from the menu (Add, View, Search, Update, Delete, Exit)
-

Testing Points

- Ensure duplicate roll numbers are rejected
- Check search results for both roll number and name
- Verify updates retain previous values if fields are skipped
- Confirm deletion after user confirmation
- Close and reopen program → data should remain available via JSON

Screenshots:

Menu with options:



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

Adding a student:

```
1) Add  2) View  3) Search  4) Update  5) Delete  6) Exit
Choice: 1
Roll no: 504
Name: Shashi
Grade: B
Age: 23
✅ Student added.
```

Viewing student list:

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

| Roll | Name | Grade | Age |
|------|---------|-------|-----|
| 501 | Sowmya | A | 22 |
| 502 | Bhavani | A | 21 |
| 503 | Ajay | B | 22 |
| 504 | Shashi | B | 23 |

Searching a record:

```
1) Add  2) View  3) Search  4) Update  5) Delete  6) Exit
Choice: 3
Enter roll no to search: 501
Found: {'roll_no': '501', 'name': 'Sowmya', 'grade': 'A', 'age': '22'}
```

Updating a record:

```
1) Add  2) View  3) Search  4) Update  5) Delete  6) Exit
Choice: 4
Enter roll no to search: 503
Found: {'roll_no': '503', 'name': 'Ajay', 'grade': 'B', 'age': '22'}
Name [Ajay]: Ajay Reddy
Grade [B]: A
Age [22]: 22
✅ Student updated.
```

Deleting a student:

```
1) Add 2) View 3) Search 4) Update 5) Delete 6) Exit
Choice: 5
Enter roll no to search: 504
Found: {'roll_no': '504', 'name': 'Shashi', 'grade': 'B', 'age': '23'}
Delete? (y/N): y
Deleted.
```

Exit:

```
1) Add 2) View 3) Search 4) Update 5) Delete 6) Exit
Choice: 6
Goodbye 🙋
```

What I Learned

- Hands-on practice with lists and dictionaries
 - Menu-driven program design
 - Input validation and exception handling
 - File handling with JSON
 - End-to-end project development skills
-

Future Enhancements

- Enable partial name search
- Add sorting (by roll number, name, grade)
- Extend project into a web app using Flask