PROJECT TITLE

STUDENT MANAGEMENT SYSTEM

PROJECT REPORT

Submitted by

SOMALA SHASAHIKALA

Subject

Python Programming

Submitted to

Mr. Rohan

Institute Name

Besant Technologies

Year

2025

Place

Bangalore, Karnataka

PROJECT SUMMARY

The Student Management System is a Python console program that helps you keep track of student information easily. You can add new students, update their details, search for students, see all records, and delete any wrong or old data. The system saves all the information in JSON files, so nothing gets lost when you close the program. It's a simple and reliable way to manage student records.

PREREQUISITES

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

PROJECT STRUCTURE

- Student_mgmt.py →Main script handling student management operations.
- Data. Json → Auto-created JSON file storing student records.
- README.md → Documentation with usage instructions and project details.

DATA MODEL

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": 200, "name": "Shashikala", "grade": "A", "age": 22}

All records are maintained in a list called students.

KEY FEATURES

- Add students with Unique Roll Numbers
- Show all Student Records in an Organized Table
- Find Students by Roll Number or Name
- Modify Existing Student information
- Remove Student Records with Confirmation
- Data Storage and Retrieval Using JSON

STEPS TO RUN THE APPLICATION

- Navigate to the project folder using terminal or command prompt
- Start the program with the command:

  Python student_mgmt.py

- A menu will be displayed containing the following choice:

  a. Add
  b. View
  c. Search
  d. Update
  e. Delete
  f. Exit

- Pick an option and proceed an guided by the system

TESTING POINTS

- 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
Roll no: 201
Name: Hima
Grade: A
Age (optional): 21
Student added.
```

Viewing student list:

```
Student Management System
1) Add  2) View  3) Search  4) Update  5) Delete  6) Exit
Choice: 2
Roll       Name                 Grade Age
-----------------------------------------
201        Hima                 A     21
202        Sowmya               A+    22
203        Lavanya              B     24
```

Searching a record:

```
1) Add  2) View  3) Search  4) Update  5) Delete  6) Exit
Choice: 3
Enter roll no to search: 201
Found: {'roll_no': '201', 'name': 'Hima', 'grade': 'A', 'age': '21'}
```

Updating a record:

```
1) Add  2) View  3) Search  4) Update  5) Delete  6) Exit
Choice: 4
Roll to update: 202
Name [Sowmya]: Sowmya Reddy
Grade [A+]: B
Age [22]: 25
Student updated.
```

Deleting a student:

```
1) Add  2) View  3) Search  4) Update  5) Delete  6) Exit
Choice: 5
Roll to delete: 201
Are you sure you want to delete? (y/N): y
Student deleted.
```

Exit:

```
1) Add  2) View  3) Search  4) Update  5) Delete  6) Exit
Choice: 6
Data saved. Goodbye!
```

What I learned

- Practical experience in working with python lists and dictionaries
- Implementing strong input checks and handling runtime errors
- Improved problem-solving and debugging skills
- Enhanced logical thinking for program flow control
- Gaining confidence in developing a complete project from scratch

---

FUTURE IMPROVEMENT

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