

Student Marks Management & Analysis System

- Student Marks Management & Analysis System using Python.
- Name - Navonil Ganguli
- Year - 4th year
- College - Netaji Subhash Engineering College

Project Overview

1.A Python program to:

- Store student names and marks
- Calculate total, average, and grade
- Display a ranked list of students

2.Uses lists, dictionaries, functions, sorting, and file handling

3.Demonstrates data processing & analysis concepts

Project Objectives

- Learn Python data structures (list, dictionary).
- Use functions for modular programming.
- Implement sorting & aggregation.
- Store and retrieve data from a file.

Table Structure

Data will be stored in Python as:

```
students = [  
    {"name": "Alice", "marks": [85, 78, 92]},  
    {"name": "Bob", "marks": [72, 88, 91]},  
]
```

Fields:

- **name** – Student's Name
- **marks** – List of marks in subjects
- **Computed:** total, average, grade

Code: Data Input & Storage

```
students = []
```

```
def add_student(name, marks):  
    students.append({"name": name, "marks": marks})
```

```
add_student("Alice", [85, 78, 92])  
add_student("Bob", [72, 88, 91])  
add_student("Charlie", [90, 80, 95]))  
);
```

Code: Grade Calculation

```
def calculate_grade(avg):  
    if avg >= 90:  
        return "A+"  
    elif avg >= 75:  
        return "A"  
    elif avg >= 60:  
        return "B"  
    else:  
        return "C"
```

Code: Processing & Ranking

for student in students:

```
    total = sum(student["marks"])
```

```
    avg = total / len(student["marks"])
```

```
    grade = calculate_grade(avg)
```

```
    student["total"] = total
```

```
    student["avg"] = avg
```

```
    student["grade"] = grade
```

```
students.sort(key=lambda x: x["total"],  
reverse=True)
```

Output Display

```
print("Name | Total | Average | Grade")
for s in students:
    print(f"{s['name']} | {s['total']} | {s['avg']:.2f} | {s['grade']}")
```

Sample Output:

Name	Total	Average	Grade
Charlie	265	88.33	A
Alice	255	85.00	A
Bob	251	83.67	A

File Handling

```
with open("student_report.txt", "w") as f:  
    for s in students:  
        f.write(f"{s['name']},{s['total']},{s['avg']:.2f},  
{s['grade']}\\n")
```

Conclusion

- Covers **data input, processing, and storage**
- Uses **functions, sorting, file I/O**
- Easily extendable for:
 1. More subjects
 2. GUI using Tkinter
 3. Database integration