



SRINIVASA EDUCATIONAL SOCIETY'S

## **PACE INSTITUTE OF TECHNOLOGY & SCIENCES** **(AUTONOMOUS)**

Approved by AICTE, UGC, New Delhi & Govt. of Andhra Pradesh | Permanently Affiliated to JNTUK, Kakinada, A.P.

**ACCREDITED BY NAAC WITH 'A' GRADE | ACCREDITED BY NBA**

An ISO 9001 : 2008 Certified Institution | 'A' Grade Engineering College by Government of A.P.

NH-16, Near Valluramma Temple, ONGOLE - 523 272, A.P., Contact No.: 08592 278315, 9581456310 | [www.pace.ac.in](http://www.pace.ac.in)

### **DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING** **(2023-2024)**

## **Internship and mini project based on python programming with Data Engineer**

**Project title: Scholastic Achievement Dashboard**

**In accordance with requirement of degree of  
BACHELOR OF TECHNOLOGY  
In  
ELECTRICAL AND ELECTRONICS ENGINEERING**

**Submitted by:**

**Deepika Manchikalapati**  
**21KQ1A0217**

**Date: 11-06-2024**

# Scholastic Achievement Analytics

## Abstract:

This abstract presents a literature review on the use of a large array of data about students and courses that was collected by institutions and learning analytics to improve students success. The purpose of this paper is to provide a brief overview of how academic analytics has been used in educational institutions, what tools are available, and how institution can predict student performance and achievement. In addition, the study will discuss its applications, goals, examples, and why instructors want to make use of academic analytics. Finally, this will propose an intelligent recommendation intervention to improve students' achievement that will be based on two outcomes, performance as measured by final grade, and students' information data such as attendance, subject, marks, roll numbers, total, highest marks and lowest marks like that. And to store all this data by using python programming language.

## Description:

Scholastic achievement refers to the measurable performance of students in their educational pursuits, often assessed through grades, standardized test scores, and other academic indicators. This concept is central to understanding how effectively students are learning and progressing through their educational journeys. This concept is crucial for understanding how well students are learning and advancing through their educational pathways. Key factors influencing scholastic achievement include parental involvement, teaching quality, school environment, and individual student characteristics. Higher family income and parental education levels generally provide better educational resources and support, leading to improved academic performance. Active parental involvement, effective teaching strategies, a safe and inclusive school environment, and intrinsic student motivation are also essential components. Recommendations for improving scholastic achievement include differentiated instruction by educators, investment in educational infrastructure and teacher training by policymakers. By addressing these interconnected factors, stakeholders can work towards creating an educational system that supports and nurtures the scholastic achievement of all students.

## Requirements:

To ensure a comprehensive system for tracking and promoting scholastic achievement, various requirements need to be addressed across multiple domains.

### Student Data Management:

- Store personal details (name, age, grade, contact information).
- Maintain academic records (grades, performance, subjects).
- Ensure data security and privacy for student information.
- Implement access controls and encryption mechanisms.
- Ensure compatibility with existing educational systems and platforms.
- Support data import/export in standard formats.

- Choose a suitable technology stack (e.g., Python for backend, React for frontend).
- Use a robust database system (e.g. pandas).

### Approach:

To design a system that tracks and promotes scholastic achievement, we can adopt a modular approach leveraging Python for backend development, ensuring scalability, flexibility, and ease of maintenance. Below is a detailed approach outlining key steps and modules necessary for the implementation.

**Application Layer:** The backend logic implemented in Python, using frameworks like pandas and functions

**Features:** Manage student details, academic records, health records.

**Implementation:** Create models for student data, academic records..

### Explanation:

• **import pandas as pd:** Import the pandas library, which is useful for data manipulation and analysis.

□ **s = []:** Initialize an empty list s to store the information of each student.

□ **def data(name, roll, subject, marks, total):**: Define a function data that takes student details as parameters.

□ **student = {...}:** Create a dictionary student containing the student's name, roll number, subjects, marks, and total marks.

□ **s.append(student):** Append the student dictionary to the list s.

**n = int(input()):** Read an integer input n, which represents the number of students.

□ **for i in range(n):**: Loop n times to gather information for each student.

□ **name = input("enter student name: "):** Prompt and read the student's name.

□ **roll = int(input("enter roll number: "):** Prompt and read the student's roll number as an integer.

□ **subject = list(input("enter subject name: ").split()):** Prompt and read the subject names, split them by spaces, and convert them to a list.

□ **marks = list(map(int, input("enter student marks: ").split())):** Prompt and read the student's marks, split them by spaces, convert each mark to an integer, and store them in a list.

□ **total = sum(marks):** Calculate the total marks by summing up the marks list.

□ **data(name, roll, subject, marks, total):** Call the data function to store the student's information.

□ **df = pd.DataFrame(s)**: Convert the list `s` (which contains dictionaries of student information) into a pandas DataFrame `df`.

□ **print(df)**: Print the DataFrame to display the student information in a tabular format.

**b = []**: Initialize an empty list `b` to store the total marks of each student.

□ **for j in range(n)::** Loop `n` times to process each student's total marks.

□ **a = s[j]['total']**: Extract the total marks of the `j`-th student from the list `s`.

□ **b.append(a)**: Append the total marks to the list `b`.

□ **print('max total=:', max(b))**: Print the maximum total marks from the list `b`.

□ **print('min total=:', min(b))**: Print the minimum total marks from the list `b`.

□ **b.sort()**: Sort the list `b` in ascending order.

□ **print('sort total=:', b)**: Print the sorted list of total marks.

□ **print(set(b))**: Print the set of unique total marks from the list `b`.

**c = []**: Initialize an empty list `c` to store the marks lists of each student.

□ **for k in range(n)::** Loop `n` times to process each student's marks.

□ **d = s[k]['marks']**: Extract the marks list of the `k`-th student from the list `s`.

□ **c.append(d)**: Append the marks list to the list `c`.

□ **for p in range(n)::** Loop `n` times to analyze each student's marks.

□ **print(f'maximum score of student{p+1}:', max(c[p]))**: Print the maximum mark for the `p`-th student.

□ **print(f'minimum score of student{p+1}:', min(c[p]))**: Print the minimum mark for the `p`-th student.

□ **print(f'total score of student{p+1}:', sum(c[p]))**: Print the total marks for the `p`-th student.

□ **print(f'average score of student{p+1}:', sum(c[p]) / len(c[p]))**: Print the average marks for the `p`-th student.

□ **c[p].sort()**: Sort the marks list of the `p`-th student in ascending order.

□ **print(f'sort of all marks {p+1}:', c[p])**: Print the sorted marks list of the `p`-th student.

❑ **print(f'score of each subject without duplicates:', set(c[p])):** Print the unique marks of the p-th student.

**m = []:** Initialize an empty list m to store marks for each subject.

❑ **for j in range(len(marks)):** Loop over the number of subjects.

❑ **for i in range(n):** Loop n times to process each student's marks for the j-th subject.

❑ **m.append(s[i]['marks'][j]):** Append the marks of the i-th student for the j-th subject to the list m.

❑ **print(f'maximum marks in subject {j+1}:', max(m)):** Print the maximum marks for the j-th subject.

❑ **m.clear():** Clear the list m to prepare for the next subject's marks.

```
*project.py - C:\Users\my lap\OneDrive\Desktop\PYTHON\project.py (3.12.1)*
File Edit Format Run Options Window Help
import pandas as pd
s=[]
def data(name,roll,subject,marks,total):
    student={
        'name':name,
        'rollnumber':roll,
        'subject':subject,
        'marks':marks,
        'total':total
    }
    s.append(student)
n=int(input())
for i in range(n):
    name=input("enter student name: ")
    roll=int(input("enter roll number: "))
    subject=list(input("enter subject name: ").split())
    marks=list(map(int,input("enter student marks: ").split()))
    total=sum(marks)
    data(name,roll,subject,marks,total)
df=pd.DataFrame(s)
print(df)
b=[]
for j in range(n):
    a=s[j]['total']
    b.append(a)
    print('max total:',max(b))
    print('min total:',min(b))
    b.sort()
    print('sort total:',b)
    print(set(b))
```

```

    print(set(v))
c=[]
for k in range(n):
    d=s[k]['marks']
    c.append(d)
for p in range(n):
    print(f'maximum score of student{p+1}:',max(c[p]))
    print(f'minimum score of student{p+1}:',min(c[p]))
    print(f'total score of student{p+1}:',sum(c[p]))
    print(f'average score of student{p+1}:',sum(c[p])/len(c[p]))
    c[p].sort()
    print(f'sort of all marks {p+1}:',c[p])
    print(f'score of each subject without duplicates:',set(c[p]))
m=[]
for j in range(len(marks)):
    for i in range(n):
        m.append(s[i]['marks'][j])
        print(f'maximum marks in subject {j+1}:',max(m))
        m.clear()

```

Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

```

>>> ===== RESTART: C:\Users\P00JITHA\AppData\Local\Programs\Python\Python311\proj\data3.py =====
20
enter student name:deepika
enter roll number:17
enter marks:25 35 46
enter student name:navya
enter roll number:13
enter marks:43 56 24
enter student name:sruthi
enter roll number:09
enter marks:76 58 49
enter student name:rahithya
enter roll number:29
enter marks:68 47 63
enter student name:hema
enter roll number:21
enter marks:55 83 49
enter student name:jyothi
enter roll number:22
enter marks:45 36 28
enter student name:jahnavi

```

enter student name:jahnavi

enter roll number:02

enter marks:45 32 65

enter student name:pooji

enter roll number:01

enter marks:54 37 48

enter student name:sravani

enter roll number:12

enter marks:34 75 64

enter student name:ramya

enter roll number:10

enter marks:36 57 39

enter student name:kavya

enter roll number:24

enter marks:58 64 39

enter student name:bhavana

enter roll number:03

enter marks:68 59 30

enter student name:devika

enter roll number:18

enter marks:23 56 48

enter student name:anu

enter roll number:04

enter marks:36 48 59

enter student name:divya



```

enter student name:divya
enter roll number:15
enter marks:24 56 86
enter student name:naveena
enter roll number:16
enter marks:24 75 47
enter student name:naga
enter roll number:20
enter marks:25 36 47
enter student name:meghana
enter roll number:14
enter marks:65 48 39
enter student name:jaya
enter roll number:11
enter marks:34 56 73
enter student name:kala
enter roll number:23
enter marks:54 32 67

```

	name	rollnumber	marks	total
0	deepika	17	[25, 35, 46]	106
1	navya	13	[43, 56, 24]	123
2	sruthi	9	[76, 58, 49]	183
3	rahithya	29	[68, 47, 63]	178
4	hema	21	[55, 83, 49]	187
5	jyothi	22	[45, 36, 28]	109
6	jahnavi	2	[45, 32, 65]	142
7	pooji	1	[54, 37, 48]	139
8	sravani	12	[34, 75, 64]	173
9	ramya	10	[36, 57, 39]	132
10	kavya	24	[58, 64, 39]	161
11	bhavana	3	[68, 59, 30]	157
12	devika	18	[23, 56, 48]	127
13	anu	4	[36, 48, 59]	143
14	divya	15	[24, 56, 86]	166
15	naveena	16	[24, 75, 47]	146
16	naga	20	[25, 36, 47]	108
17	meghana	14	[65, 48, 39]	152
18	jaya	11	[34, 56, 73]	163
19	kala	23	[54, 32, 67]	153



```
max total=: 187
min total=: 106
sort total=: [106, 108, 109, 123, 127, 132, 139, 142, 143, 146, 152, 153, 157, 161, 163, 166, 173, 178, 183, 187]
{132, 139, 142, 143, 146, 152, 153, 157, 161, 163, 166, 173, 178, 183, 187, 106, 108, 109, 123, 127}
maximum score of student1: 46
minimum score of student1: 25
total score of student1: 106
average score of student1: 35.33333333333336
sort of all marks 1: [25, 35, 46]
score of each subject without duplicates: {25, 35, 46}
maximum score of student2: 56
minimum score of student2: 24
total score of student2: 123
average score of student2: 41.0
sort of all marks 2: [24, 43, 56]
score of each subject without duplicates: {24, 56, 43}
maximum score of student3: 76
minimum score of student3: 49
total score of student3: 183
average score of student3: 61.0
sort of all marks 3: [49, 58, 76]
score of each subject without duplicates: {49, 58, 76}
maximum score of student4: 68
```

```
maximum score of student4: 68
minimum score of student4: 47
total score of student4: 178
average score of student4: 59.33333333333336
sort of all marks 4: [47, 63, 68]
score of each subject without duplicates: {68, 63, 47}
maximum score of student5: 83
minimum score of student5: 49
total score of student5: 187
average score of student5: 62.33333333333336
sort of all marks 5: [49, 55, 83]
score of each subject without duplicates: {49, 83, 55}
maximum score of student6: 45
minimum score of student6: 28
total score of student6: 109
average score of student6: 36.33333333333336
sort of all marks 6: [28, 36, 45]
score of each subject without duplicates: {28, 45, 36}
maximum score of student7: 65
minimum score of student7: 32
total score of student7: 142
average score of student7: 47.33333333333336
sort of all marks 7: [32, 45, 65]
score of each subject without duplicates: {32, 65, 45}
maximum score of student8: 54
```

```
minimum score of student8: 37
total score of student8: 139
average score of student8: 46.33333333333336
sort of all marks 8: [37, 48, 54]
score of each subject without duplicates: {48, 37, 54}
maximum score of student9: 75
minimum score of student9: 34
total score of student9: 173
average score of student9: 57.666666666666664
sort of all marks 9: [34, 64, 75]
score of each subject without duplicates: {64, 34, 75}
maximum score of student10: 57
minimum score of student10: 36
total score of student10: 132
average score of student10: 44.0
sort of all marks 10: [36, 39, 57]
score of each subject without duplicates: {57, 36, 39}
maximum score of student11: 64
minimum score of student11: 39
total score of student11: 161
average score of student11: 53.666666666666664
sort of all marks 11: [39, 58, 64]
score of each subject without duplicates: {64, 58, 39}
maximum score of student12: 68
minimum score of student12: 30
```



```
total score of student12: 157
average score of student12: 52.33333333333336
sort of all marks 12: [30, 59, 68]
score of each subject without duplicates: {59, 68, 30}
maximum score of student13: 56
minimum score of student13: 23
total score of student13: 127
average score of student13: 42.33333333333336
sort of all marks 13: [23, 48, 56]
score of each subject without duplicates: {48, 56, 23}
maximum score of student14: 59
minimum score of student14: 36
total score of student14: 143
average score of student14: 47.666666666666664
sort of all marks 14: [36, 48, 59]
score of each subject without duplicates: {48, 59, 36}
maximum score of student15: 86
minimum score of student15: 24
total score of student15: 166
average score of student15: 55.33333333333336
sort of all marks 15: [24, 56, 86]
score of each subject without duplicates: {24, 56, 86}
maximum score of student16: 75
minimum score of student16: 24
total score of student16: 146
```

```
total score of student16: 146
average score of student16: 48.666666666666664
sort of all marks 16: [24, 47, 75]
score of each subject without duplicates: {24, 75, 47}
maximum score of student17: 47
minimum score of student17: 25
total score of student17: 108
average score of student17: 36.0
sort of all marks 17: [25, 36, 47]
score of each subject without duplicates: {25, 36, 47}
maximum score of student18: 65
minimum score of student18: 39
total score of student18: 152
average score of student18: 50.666666666666664
sort of all marks 18: [39, 48, 65]
score of each subject without duplicates: {48, 65, 39}
maximum score of student19: 73
minimum score of student19: 34
total score of student19: 163
average score of student19: 54.333333333333336
sort of all marks 19: [34, 56, 73]
score of each subject without duplicates: {56, 73, 34}
maximum score of student20: 67
minimum score of student20: 32
total score of student20: 153
```

```
total score of student20: 153  
average score of student20: 51.0  
sort of all marks 20: [32, 54, 67]  
score of each subject without duplicates: {32, 67, 54}  
maximum marks in subject 1: 49  
maximum marks in subject 2: 64  
maximum marks in subject 3: 86
```

#### Conclusion:

The provided code effectively demonstrates the management and analysis of student marks using Python and the pandas library