



**JAIN**  
DEEMED-TO-BE UNIVERSITY

SCHOOL OF  
COMPUTER  
SCIENCE AND IT

**School of Computer Science and Information Technology**

**Department of CS & IT**

**IV Semester**

**SUBJECT CODE: 23BCA4C03**

**SUBJECT NAME: PYTHON PROGRAMMING**

**Activity - 2**

**Python Project: Academic Tracker and Resource  
Recommender**

**Specialization: MACT**

**Date of Submission: 15 – 04 - 2025**

**Submitted By:**

Manan Hemani	23BCAR0485
Om Chitha Swaroop	23BCAR0458

**Faculty In-Charge:**

**Mr. Sahabzada Betab Badar**

# Index

Sl. No.	Contents	Page No.
1	Problem Statement	3
2	Dataset Description	3 – 4
3	Coding Part	4 – 5
4	Example Insights	6 – 7
5	Chart Diagram	8
6	Analysis	8 – 9
7	Future Aspect	9
8	Conclusion	10
9	Reference	10

## 1. Problem Statement

In the current educational landscape, students often face challenges in monitoring their academic performance and identifying the areas where they need improvement. Despite the availability of test scores and assignment marks, many students struggle to understand what those numbers imply about their learning progress. Moreover, there is usually a lack of personalized support in recommending appropriate learning materials that align with each student's unique needs and capabilities.

Traditional systems used in schools and colleges provide only static data like grades and attendance records, without offering meaningful insights or suggestions. As a result, students who are underperforming may continue to do so without any clear direction for improvement. Teachers also face difficulties in giving personalized attention to each student due to time constraints and large classroom sizes.

The Academic Tracker aims to address these issues by offering a smart solution that not only tracks academic progress but also provides resource suggestions tailored to individual student needs. This system analyzes student data such as test scores, assignment completion rates, attendance percentages, and study hours to give a comprehensive view of each student's performance. Based on this analysis, it recommends learning resources like video lectures, textbooks, websites, or productivity tools to help the student improve.

By adopting this approach, the Academic Tracker seeks to make academic tracking more meaningful, insightful, and student-centered, ultimately helping students become more aware and responsible for their learning journey.

## 2. Dataset Description

The project has 2 CSV files which contains the data, they are described below:

### 1. Users.csv

This file stores detailed academic data for each student. It is dynamically created and updated as students input their subject-wise performance. Each row in the file represents a single subject entry for a particular student.

Column Name	Description
Student Name	Name of the student (capitalized and title-cased).
Subject	Name of the subject (e.g., Python, Java, DSA, etc.).
Score (%)	Percentage score obtained in the subject.
Attendance (%)	Overall attendance percentage of the student.
Overall (%)	Average percentage across all subjects.
Weak Subject	Indicates whether the subject is a weak area for the student (Yes/No).

## 2. Resource.csv

This is a static file containing personalized learning recommendations based on student performance and attendance. It is preloaded with suggested resources and conditions for when they should be displayed.

Column Name	Description
Subject	Subject name for which the recommendation is applicable.
Conditions	Logical condition based on score and attendance (e.g., score<33 & attendance<75).
Recommendation	The actual advice, study tip, or resource link for the student.

## 3. Coding Part

**GitHub:** <https://github.com/manan-hemani/Academic-Tracker>

Implementation of the Academic Tracker and Resource Recommender system was done by employing the Python programming language, taking advantage of core libraries like **Pandas**, **NumPy**, and **Matplotlib**. The project is constructed

based on **Object-Oriented Programming (OOP)** principles to promote modularity, readability, and maintainability. The entire core functionality is wrapped within a single class Academic Tracker involving data input, progress calculation, data storage, visualization, and generation of recommendations. This facilitates direct separation of concerns and simplicity in extension in the future.

The system starts with the user interaction in which the student enters his/her name and academic information such as subjects, marks obtained, total marks, and overall attendance percentage. The data is checked and stored in relevant instance variables. Percentage score in each subject is determined by dividing the marks obtained by total marks and then multiplying by 100. After entering all the subject scores, the system computes the overall academic performance as a mean of all the subject percentages using the NumPy library.

Once progress has been calculated, the information is stored to a CSV file (**users.csv**) in a formatted manner with Pandas. Each subject for a student is kept on a single row for improved structure and querying. If the CSV file exists, then the new data is added to the existing file to keep a historical record of submissions. The platform also has functionality to identify and show weak subjects—those in which the score is less than a specified value (e.g., 33%).

For graphical analysis, the project utilizes Matplotlib to create a bar chart representing subject-wise scores. The chart contains labeled axes, a threshold line to mark the pass mark, and titles for improved interpretability. This graphic aid enables students to easily determine subjects in which they might require improvement.

Last, the system picks from another CSV file, **resources.csv**, containing predefined suggestions related to performance and attendance conditions. According to the student's score and attendance, pertinent personalized advice and resources (such as tutorials, practice sites, or motivational advice) are presented. The logic dynamically corresponds conditions like "**score < 33 and attendance < 75**" to provide feedback that suits the situation, supporting targeted learning.

As a whole, the coding aspect merges file manipulation, data processing, visualization, and conditional logic into an intuitive and smart study support system. Applying OOP means that the system is maintainable and simple to extend, e.g., including login support, connecting databases, or developing a GUI later.

## 4. Example Insight

To better illustrate how the Academic Tracker and Resource Recommender functions, the following screenshots demonstrate the key stages of the project workflow along with corresponding outputs:

- Welcome Screen and Student Input

The first screenshot shows the welcome message and prompts the user to enter their name. If the student is new, the system proceeds to collect subject-wise scores and attendance percentage. This step ensures personalized data collection for every individual student.

```
Welcome to academic Tracker and Resource Recommender
*****

Your Name:abc

User not found. Let's enter your details.

Enter the No.of.Subject:2
Enter Subject 1: python
Enter the marks you scored in Python: 65
Enter the total marks in Python: 100
Enter Subject 2: java
Enter the marks you scored in Java: 75
Enter the total marks in Java: 100
Enter your Overall Attendance Percentage:77
```

- Entered Details Confirmation

After entering all subject marks and attendance, the system displays a summary of the entered data and asks for confirmation. This verification step prevents data entry mistakes and enhances the accuracy of the tracker.

```
Your Entered Details:

Python: 65.0%
Java: 75.0%
Attendance: 77.0%

Type yes to confirm or no to re-enter your details:yes

Details Confirmed and Saved.
```

- Calculating Progress and Report Table

Once confirmed, the program calculates the student's overall performance. It generates a detailed table showing subject names, corresponding percentages, attendance, overall percentage, and whether a subject is considered weak (based on the threshold criteria). This

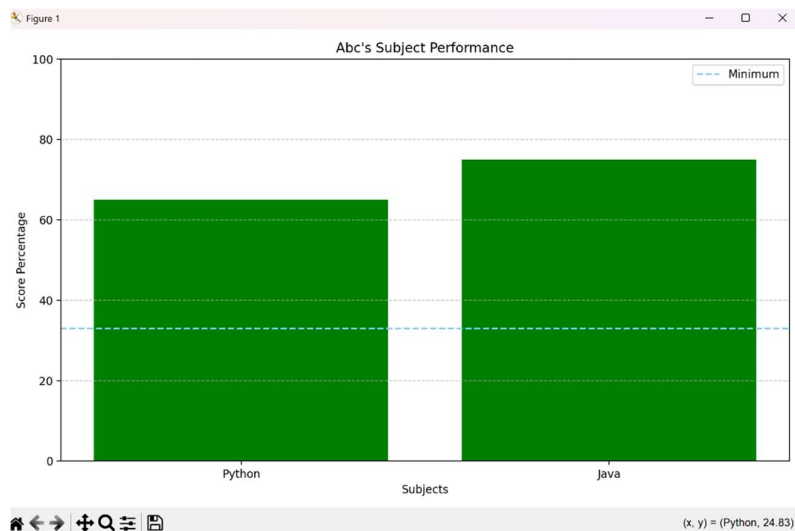
tabular format makes it easy to interpret academic performance at a glance.

```
Calculating your Progress....
Your Updated Academic Report:
```

Student Name	Subject	Score (%)	Attendance (%)	Overall (%)	Weak Subject
Abc	Python	65.0	77.0	70.0	No
Abc	Java	75.0	77.0	70.0	No

- Performance Graph Visualization

The bar chart represents the student's scores across all subjects. The inclusion of a red dashed line for the pass threshold allows for immediate identification of weak areas. This visual feedback complements the data table and appeals to visual learners.



- Personalized Recommendations

The final screenshot shows the personalized subject-wise recommendations. These are dynamically generated based on score and attendance combinations and aim to help students improve their academic standing. The resources may include links to tutorials, coding platforms, or advice like attending remedial sessions.

```
Personalized Recommendations:

Python: Great job! Try building projects from here: https://realpython.com/
Java: Excellent work! Practice with real coding challenges: https://www.hackerrank.com/domains/tutorials/10-days-of-java
```

## 5. Chart Diagram for Example Insight



A sample chart diagram representing student performance across subjects. It visually highlights weak areas (scores below the pass threshold i.e. 33%) and offers insights into subject-wise strengths and weaknesses.

## 6. Analysis

The "**Academic Progress Tracker and Resource Recommender**" project aims to fill a fundamental gap for students to evaluate their academic performance in an organized and meaningful way. By means of an easy-to-use Python-based interface, students are able to enter their marks and attendance in different subjects and obtain instant, data-driven feedback. The project employs a combination of data storage, processing, and visualization methods to present a comprehensive picture of a student's progress.

The system calculates such things as subject-wise percentage marks, academic percentage overall, and weak subjects (those below a set threshold). It also assesses attendance, which is an important factor in overall academic well-being. If there is low attendance, the system takes it along with subject scores to provide customized feedback.



One of the strongest features is the inclusion of a recommendation engine. Depending on certain conditions (like low scores, low attendance, or both), it delivers personalized study resources to the student. This ensures that students identify not only their weak areas but also receive directed guidance to address them.

Data is retained indefinitely in CSV files for student advancement (users.csv) as well as course material (resources.csv). Historical tracking and eventual growth into web-enabled dashboards or analytics software are supported by this.

Technically, the project applies fundamental programming concepts like file handling, loops, conditionals, dictionaries, and Object-Oriented Programming (OOP) concepts. It also applies libraries like pandas for data manipulation, NumPy for numerical computation, and Matplotlib for visualization.

Overall, the project shows how coding can be used to create smart, helpful systems within the education system. It combines simplicity with usefulness, and hence it is the perfect solution for students who wish to monitor and improve their own academic experience on their own.

## **7. Future Aspect**

The Academic Tracker system can be further enhanced with the following advanced features:

- **AI-Powered Recommendations:**  
Integrate machine learning models to provide even more accurate and personalized resource suggestions based on learning patterns and student behavior.
- **Mental Health and Well-being Insights:**  
Include a module for students to self-assess stress levels, enabling early interventions and better support systems.
- **Mobile App Version:**  
Launch a user-friendly mobile app so students can track their performance and access study materials on the go.
- **Predictive Analytics:**  
Use data trends to predict students at risk of failing or dropping out and alert teachers for early action.
- **Graphical User Interface:**  
Transforming the command line code to GUI in python using tkinter.

## 8. Conclusion

The Academic Tracker is an innovative and practical solution designed to bridge the gap between academic performance and actionable learning improvement. Through detailed data analysis and intelligent suggestions, the system empowers students to take ownership of their education by providing clear insights into their strengths and areas needing development.

It not only benefits students by offering personalized recommendations but also assists teachers and parents in understanding academic trends and making informed decisions. By tracking parameters like test scores, attendance, assignment completion, and study habits, the Academic Tracker goes beyond traditional report cards and creates a dynamic feedback loop for continuous improvement.

In the future, the platform can be enhanced with AI-driven insights, gamification features, mobile app integration, and mental well-being tracking to further support holistic student development. Overall, Academic Tracker stands as a powerful tool that can transform educational experiences by promoting smart learning, timely support, and better outcomes for every student.

## 9. Reference

- Python Documentation
  - [www.python.org](http://www.python.org)
- NumPy Documentation
  - [www.numpy.org](http://www.numpy.org)
- Pandas Documentation
  - <https://pandas.pydata.org/docs/>
- Matplotlib Documentation
  - [www.matplotlib.org](http://www.matplotlib.org)
- Stack Overflow
  - [www.stackoverflow.com](http://www.stackoverflow.com)
- W3schools
  - [www.w3schools.com](http://www.w3schools.com)