



PES UNIVERSITY

JACKFRUIT MINIPROJECT on **Academic Performance analysis tool**

Submitted by

TEAM	NAME	SRN
Member - 1	Akshay kumar B	PES1UG25AM467
Member - 2	Deepak	PES1UG25EC321
Member - 3	Dinesh Moolimani	PES1UG25CS644
Member - 4	Dharnendra	PES1UG25CS641
Member - 5	Chetan Ganiger	PES1UG25BT051

(Aug-Jan 25-26)

under the guidance of

AMBIKA K
Asst. Prof
Department of CSE
PES University

FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
PROGRAM B.TECH

- 1. PROBLEM STATEMENT :**
Academic performance analysis tool

2. APPROCH/METHEDOLOGY DATA STRUCTURE USED

- Student data is generated automatically for all students.
- A login system is used to display each student's profile and marks.
- Total marks, percentage and CGPA are calculated for every student.
- The topper is identified by comparing total marks.
- Subject averages are computed from all student records.
- A password-change option is provided for users.
- A simple menu-driven loop controls all operations.

DATA STRUCTURES USED:

- List- to store subjects, names and departments.
- Dictionary- to store student profiles, marks and passwords.
- Loops- for generating data and showing menu options.
- Functions- to handle login, topper, averages, password changes.

3. SAMPLE INPUT OUTPUT

```
===== Academic Performance System =====
1. Login & View Result
2. Check Class Topper
3. Subject Average Marks
4. Change Password
5. Exit
Enter choice: 1
Enter your Name: Arun
Enter Password: arun@123

✅ Login Successfull!

----- STUDENT PROFILE -----
Name       : Arun
Age        : 18
Department : CSE
-----

----- RESULT -----
Maths      : 72
Python     : 71
Electronics : 76
Mechanics  : 86
Chemistry  : 54

Total Marks : 359 / 500
Percentage  : 71.8 %
CGPA        : 7.18
-----
```

4. CHALLENGES FACED

- Ensuring correct calculation for totals, percentage and CGPA.
- Handling login, password changes and input errors.
- Identifying the topper and computing subject averages accurately.

5. SCOPE FOR IMPROVEMENT

- Store data permanently using a database instead of inmemory storage.
- Allow teachers to edit marks or add new subjects.
- Provide detailed analytics like rank lists and performance charts.
- Improve security with stronger passwords and OTP verification.

6. SCREEN SHORTS OF THE EXECUTABLE CODE WITH OUTPUT

```

1  import random
2
3  # SUBJECT LIST
4  subjects = ["Maths", "Python", "Electronics", "Mechanics", "Chemistry"]
5
6  # DATABASES
7  students = {} # Stores profile: name, age, dept
8  student_marks = {} # Stores marks & CGPA
9  passwords = {} # Stores all passwords
10
11 # RANDOM DATA FOR 50 STUDENTS
12 names_pool = [
13     "Arjun", "Rahul", "Riya", "Sneha", "Karan", "Aditi", "Nikhil", "Meera", "Rohit", "Kavya",
14     "Vikram", "Ishita", "Pooja", "Sanjay", "Varun", "Devika", "Ananya", "Manoj", "Tarun", "Sahana",
15     "Ajay", "Divya", "Abhishek", "Harsh", "Tanvi", "Gaurav", "Nisha", "Vishal", "Ragini", "Aman",
16     "Yash", "Pranav", "Chandan", "Sameer", "Vijay", "Tejas", "Suresh", "Rakesh", "Pavan", "Darshan",
17     "Lakshmi", "Preeti", "Neha", "Shruti", "Deepak", "Sudeep", "Hemant", "Rashmi", "Kiran", "Bhavana"
18 ]
19
20 departments = ["CSE", "ECE", "ME", "Civil", "AI/ML"]
21
22 def generate_marks():
23     return {sub: random.randint(35, 100) for sub in subjects}
24
25 def calculate_cgpa(marks):
26     return round(sum((m / 10) for m in marks.values()) / len(marks), 2)
27
28 # Create 50 random students
29 for i in range(200):
30     name = random.choice(names_pool)
31     sid = name.lower()
32
33     age = 18
34     dept = random.choice(departments)
35     pwd = f"{sid}@123"

```

```

36
37     students[sid] = {"Name": name, "Age": age, "Department": dept}
38     passwords[sid] = pwd
39
40     marks = generate_marks()
41     total = sum(marks.values())
42     percent = round((total / 500) * 100, 2)
43     cgpa = calculate_cgpa(marks)
44
45     student_marks[sid] = {
46         "Name": name,
47         "Marks": marks,
48         "Total": total,
49         "Percentage": percent,
50         "CGPA": cgpa
51     }
52
53 # FUNCTION: CLASS TOPPER
54 def get_topper():
55     topper_id = max(student_marks, key=lambda x: student_marks[x]["Total"])
56     t = student_marks[topper_id]
57
58     print("\n===== CLASS TOPPER =====")
59     print(f"Name      : {t['Name']}")
60     print(f"Total      : {t['Total']} / 500")
61     print(f"Percentage : {t['Percentage']}%")
62     print(f"CGPA       : {t['CGPA']}")
63     print("=====\n")
64
65 # FUNCTION: SUBJECT AVERAGE
66 def show_subject_average():
67     print("\n===== SUBJECT AVERAGES =====")
68     for sub in subjects:
69         avg = sum(student_marks[s]["Marks"][sub] for s in student_marks) / len(student_marks)
70         print(f"{sub:<12} : {round(avg, 2)}")
71     print("=====\n")
72
73 # FUNCTION: CHANGE PASSWORD
74 def change_password():
75     print("\n===== CHANGE PASSWORD =====")
76     user_id = input("Enter your Student ID (letters only): ").strip().lower()
77
78     # Auto-create if not exists
79     if user_id not in students:
80         students[user_id] = {
81             "Name": user_id.capitalize(),
82             "Age": 18,
83             "Department": random.choice(departments)
84         }
85         passwords[user_id] = f"{user_id}@123"
86
87     old_pwd = input("Enter OLD password: ")
88
89     if old_pwd != passwords.get(user_id):
90         print("\n❌ Incorrect old password!\n")
91         return
92
93     new_pwd = input("Enter NEW password: ")
94     confirm = input("Confirm NEW password: ")
95
96     if new_pwd != confirm:
97         print("\n❌ Password mismatch!\n")
98         return
99
100     passwords[user_id] = new_pwd
101     print("\n✅ Password updated successfully!\n")
102
103 # MAIN SYSTEM LOOP
104 while True:

```

```

105 print("==== Academic Performance System =====")
106 print("1. Login & View Result")
107 print("2. Check Class Topper")
108 print("3. Subject Average Marks")
109 print("4. Change Password") # Swapped
110 print("5. Exit") # Swapped
111
112 choice = input("Enter choice: ")
113
114 if choice == "1":
115     name = input("Enter your Name: ").strip()
116     student_id = name.lower()
117     password = input("Enter Password: ")
118
119     correct_pwd = passwords.get(student_id, f"{student_id}@123")
120
121     if password != correct_pwd:
122         print("\n❌ Incorrect password!\n")
123         continue
124
125     print("\n✅ Login Successful!")
126
127     # Auto-add profile if not exists
128     if student_id not in students:
129         students[student_id] = {
130             "Name": name,
131             "Age": 18,
132             "Department": random.choice(departments)
133         }
134         passwords[student_id] = f"{student_id}@123"
135
136     # Create marks if new
137     if student_id not in student_marks:
138         marks = generate_marks()
139         total = sum(marks.values())
140         percent = round((total / 500) * 100, 2)
141         cgpa = calculate_cgpa(marks)
142
143         student_marks[student_id] = {
144             "Name": name,
145             "Marks": marks,
146             "Total": total,
147             "Percentage": percent,
148             "CGPA": cgpa
149         }
150
151     # DISPLAY PROFILE
152     print("\n----- STUDENT PROFILE -----")
153     print(f"Name : {students[student_id]['Name']}")
154     print(f"Age : {students[student_id]['Age']}")
155     print(f"Department : {students[student_id]['Department']}")
156     print("-----\n")
157
158     # DISPLAY RESULT
159     print("----- RESULT -----")
160     for sub, sc in student_marks[student_id]["Marks"].items():
161         print(f"{sub:<12}: {sc}")
162
163     print("\nTotal Marks :", student_marks[student_id]["Total"], "/ 500")
164     print("Percentage :", student_marks[student_id]["Percentage"], "%")
165     print("CGPA :", student_marks[student_id]["CGPA"])
166     print("-----\n")
167
168 elif choice == "2":
169     get_topper()
170
171 elif choice == "3":
172     show_subject_average()
173
174 elif choice == "4": # Now this is CHANGE PASSWORD
175     change_password()
176
177 elif choice == "5": # Now this is EXIT
178     print("\nThank you for using the system!")
179     break
180
181 else:
182     print("\nInvalid choice!\n")

```



```

===== Academic Performance System =====
1. Login & View Result
2. Check Class Topper
3. Subject Average Marks
4. Change Password
5. Exit
Enter choice: 1
Enter your Name: Arun
Enter Password: arun@123

```

✓ Login Successful!

```

----- STUDENT PROFILE -----
Name       : Arun
Age        : 18
Department : CSE

```

```

----- RESULT -----
Maths      : 72
Python     : 71
Electronics : 76
Mechanics  : 86
Chemistry  : 54

Total Marks : 359 / 500
Percentage  : 71.8 %
CGPA       : 7.18

```

```

===== Academic Performance System =====
1. Login & View Result
2. Check Class Topper
3. Subject Average Marks
4. Change Password
5. Exit
Enter choice: 2

```

```

===== CLASS TOPPER =====
Name       : Kiran
Total      : 441 / 500
Percentage : 88.2%
CGPA       : 8.82

```

```

===== Academic Performance System =====
1. Login & View Result
2. Check Class Topper
3. Subject Average Marks
4. Change Password
5. Exit
Enter choice: 3

```

```

===== SUBJECT AVERAGES =====
Maths      : 69.8
Python     : 63.1
Electronics : 67.08
Mechanics  : 68.16
Chemistry  : 62.64

```

```
===== Academic Performance System =====
1. Login & View Result
2. Check Class Topper
3. Subject Average Marks
4. Change Password
5. Exit
Enter choice: 4

===== CHANGE PASSWORD =====
Enter your Student ID (letters only): charan
Enter OLD password: charan@123
Enter NEW password: charan_123
Confirm NEW password: charan_123

✅ Password updated successfully!
```

```
===== CHANGE PASSWORD =====
Enter your Student ID (letters only): charan
Enter OLD password: charan@123
Enter NEW password: charan_123
Confirm NEW password: charan_123

✅ Password updated successfully!

===== Academic Performance System =====
1. Login & View Result
2. Check Class Topper
3. Subject Average Marks
4. Change Password
5. Exit
Enter choice: 5
```

7. GITHUB – LINK OF ALL FOUR TEAM MEMBERS

- Member-1 GitHub - <https://github.com/akshaykumarb345-beep/Academic-analysis-tool>
- Member-2 GitHub - <https://github.com/DeepakRajgit/Academic-performance-analysis-tool>
- Member-3 GitHub - <https://github.com/dineshmoolimani/AcademicPerformanceEnalysisTool.git>
- Member-4 GitHub – <https://github.com/Dharnendra14/Academic-performance-analysis-tool.git>
- Member-5-Github - <https://github.com/chetanganiger072-pixel/Academic-performance-analysis-tool.git>

