VITYARTHI PROJECT

TEACHER’S GRADEBOOK

SUBMITTED BY: AARAV SCIJU
REGN NO.: 25BCE10099

AARAV SCIJU

25BCE10099

INTRODUCTION

This is a simple tool designed to help teachers manage student records efficiently. It automates the calculation of grades, generates report cards, and provides a visual graph for analysis.

This tool allows teachers to view class rosters, analyse individual student performance, and visualize subject marks.

PROBLEM STATEMENT

Teachers have to manually calculate student marks by hand. This takes a lot of time and often leads to calculation mistakes. Teachers need a simple and automated way to manage student grades without using pen and paper.

FUNCTIONAL REQUIREMENTS

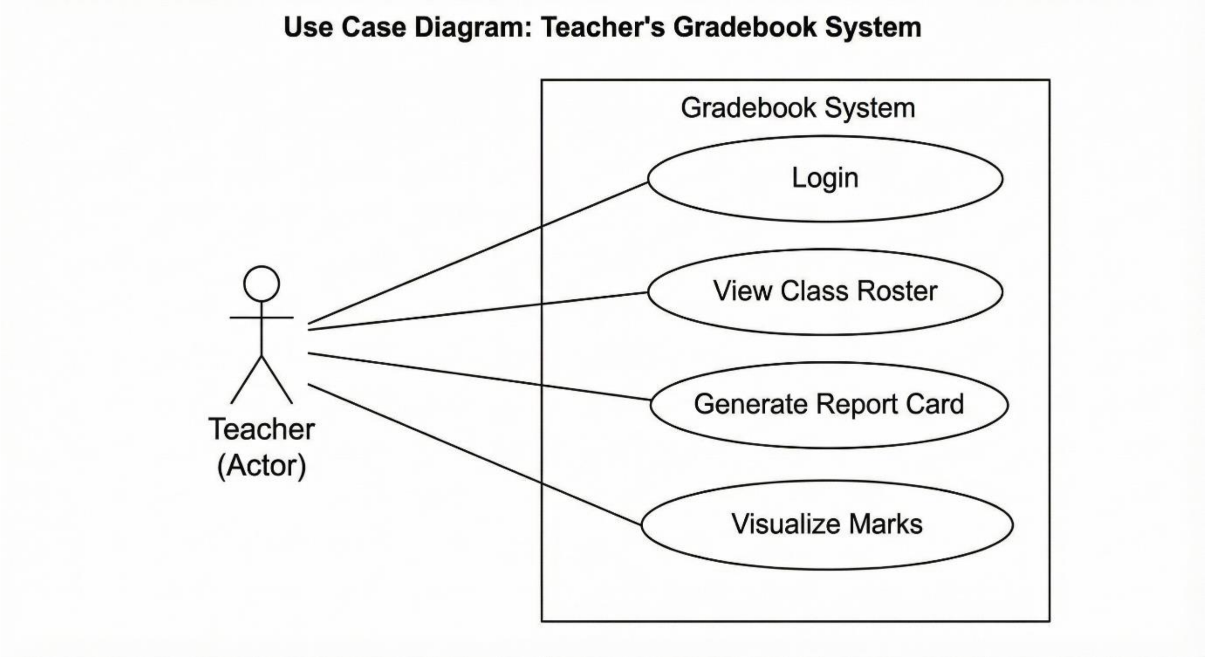
1. **Secure Access:** A password login system that restricts access to authorized users only.
2. **Class Roster Management:** A feature to list all students.
3. **Report Generation:** Takes a student's name and calculates Total Marks, Average, and Letter Grade.
4. **Visualization:** Uses matplotlib to generate a bar chart of subject marks.

NON-FUNCTIONAL REQUIREMENTS

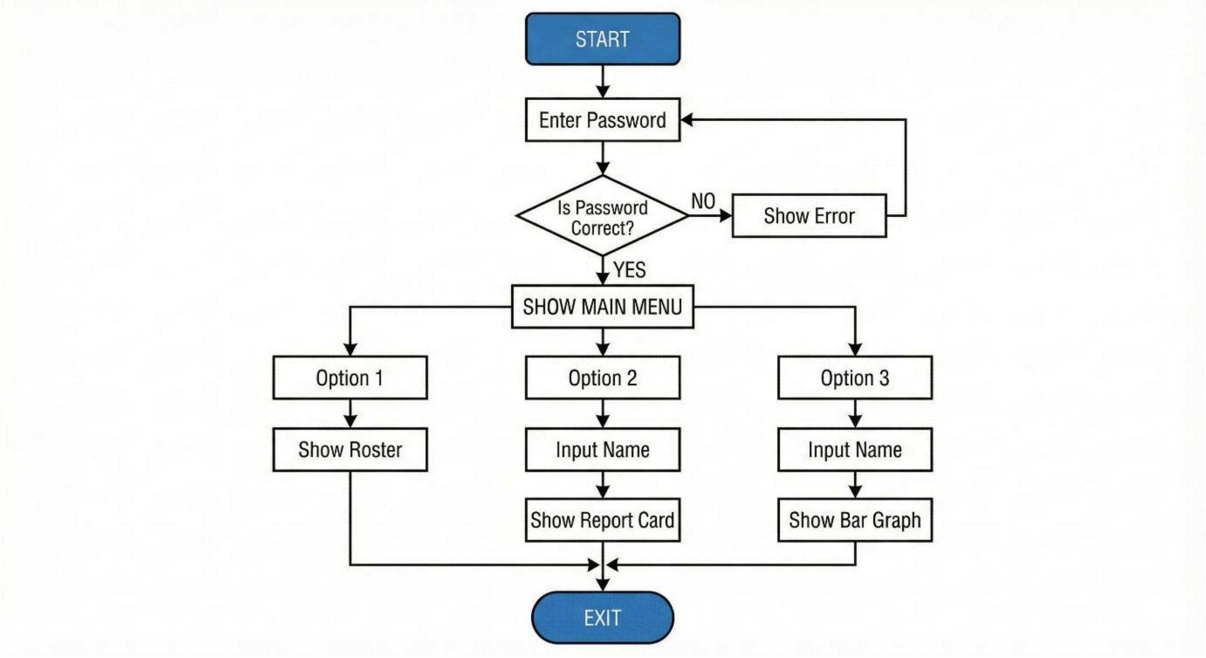
1. **User-Friendly Interface:** This project is a Command Line Interface (CLI) with a clear, numbered menu structure. Users are guided by intuitive prompts.
2. **Data Validation:** When generating reports, the system verifies if a student exists in the database before attempting calculations. If a student is not found, a descriptive error message is displayed instead of a code traceback.
3. **Error Handling:** Input case-sensitivity has been taken care of while inputting the name. Also prevented it from crashing if incorrect menu number was chosen.

DESIGN DIAGRAMS

1. Use Case Diagram:



2． Workflow Diagram:



IMPLEMENTATION

1. main.py: This file is the entry point which contains the main menu logic and also the password protection system

2. data.py: This file contains the in-memory database where you have to manually add/remove data

3. reports.py: This file contains the report card function and the calculation logic for average, pass/fail and grade

4. roster.py: This file contains the roster function which displays a list of all students in the class

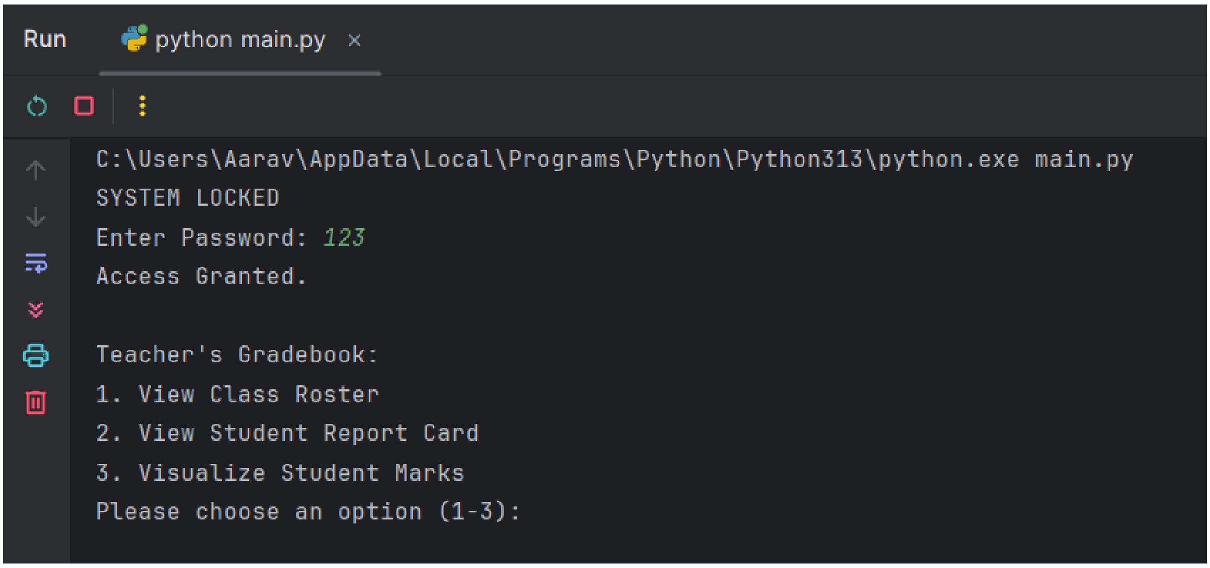
5. visualize.py: This file uses the matplotlib library to create a function which plots a student's marks across various subjects

TESTING APPROACH

1. **Test Roster**: Run the program and select Option **1**. Verify that a list of student names appears.
2. **Test Report Card**: Run the program and select Option **2**. When asked for a name, type **Aarav**. Verify that you see a list of subjects, marks, average, and pass/fail status.
3. Test Visualization: Run the program and select Option **3**. Type **Aarav**. A new window should pop up displaying a bar chart.

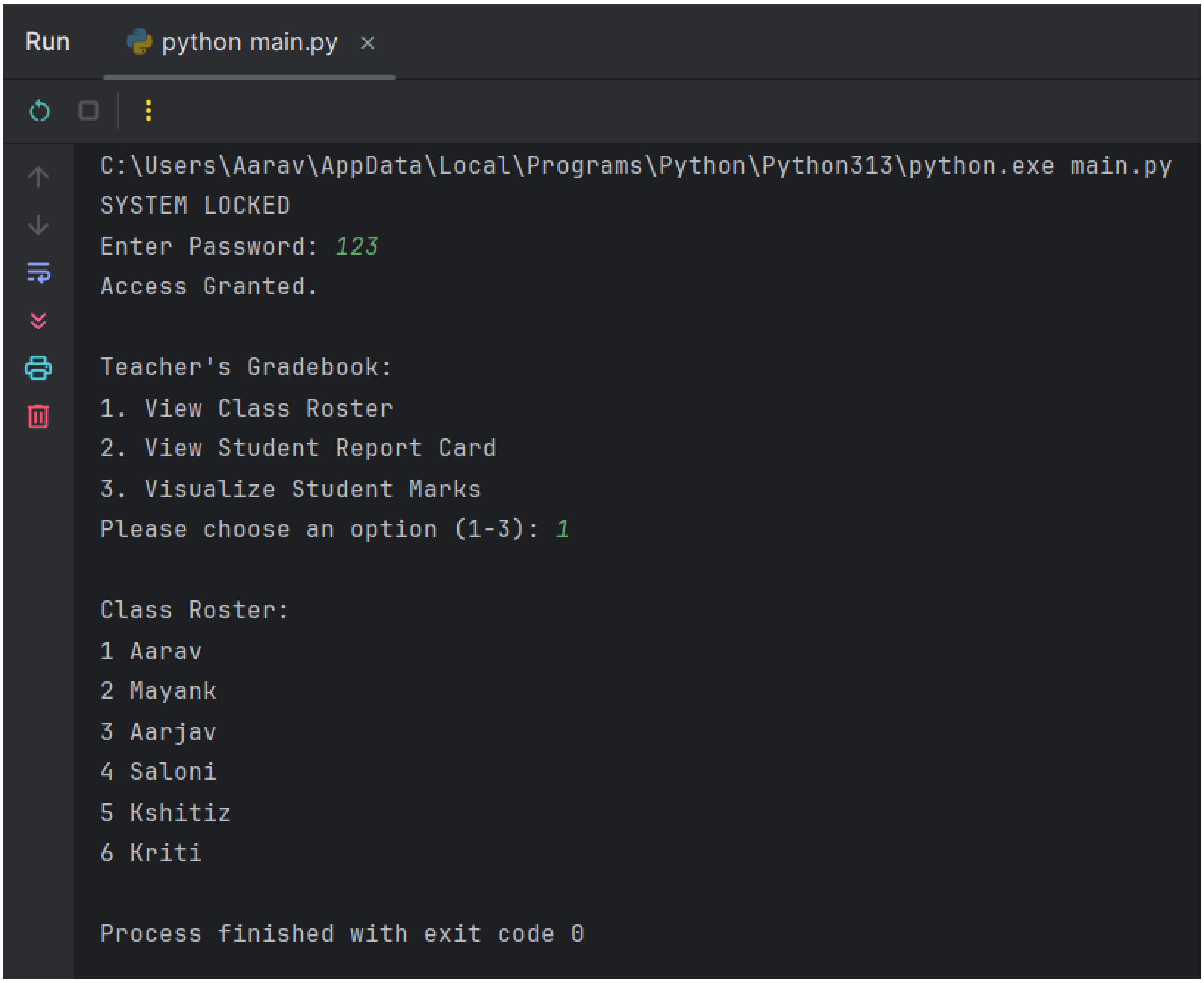
SCREENSHOTS

1. Main Menu and Login:



```
C:\Users\Aarav\AppData\Local\Programs\Python\Python313\python.exe main.py
SYSTEM LOCKED
Enter Password: 123
Access Granted.

Teacher's Gradebook:
1. View Class Roster
2. View Student Report Card
3. Visualize Student Marks
Please choose an option (1-3):
```

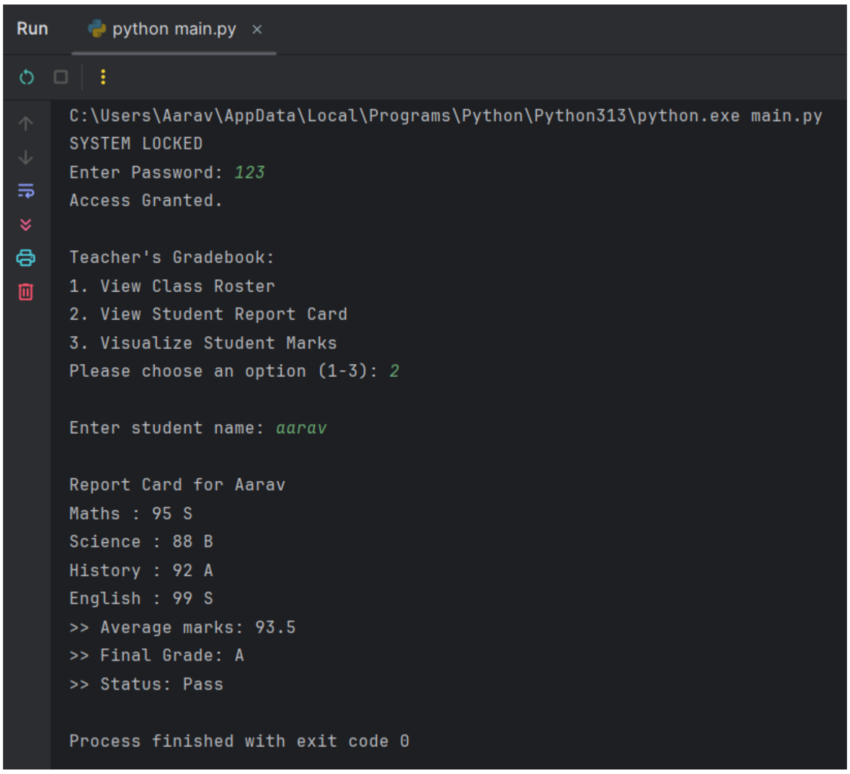
2. Class Roster:



```
C:\Users\Aarav\AppData\Local\Programs\Python\Python313\python.exe main.py
SYSTEM LOCKED
Enter Password: 123
Access Granted.

Teacher's Gradebook:
1. View Class Roster
2. View Student Report Card
3. Visualize Student Marks
Please choose an option (1-3): 1

Class Roster:
1 Aarav
2 Mayank
3 Aarjav
4 Saloni
5 Kshitiz
6 Kriti

Process finished with exit code 0
```

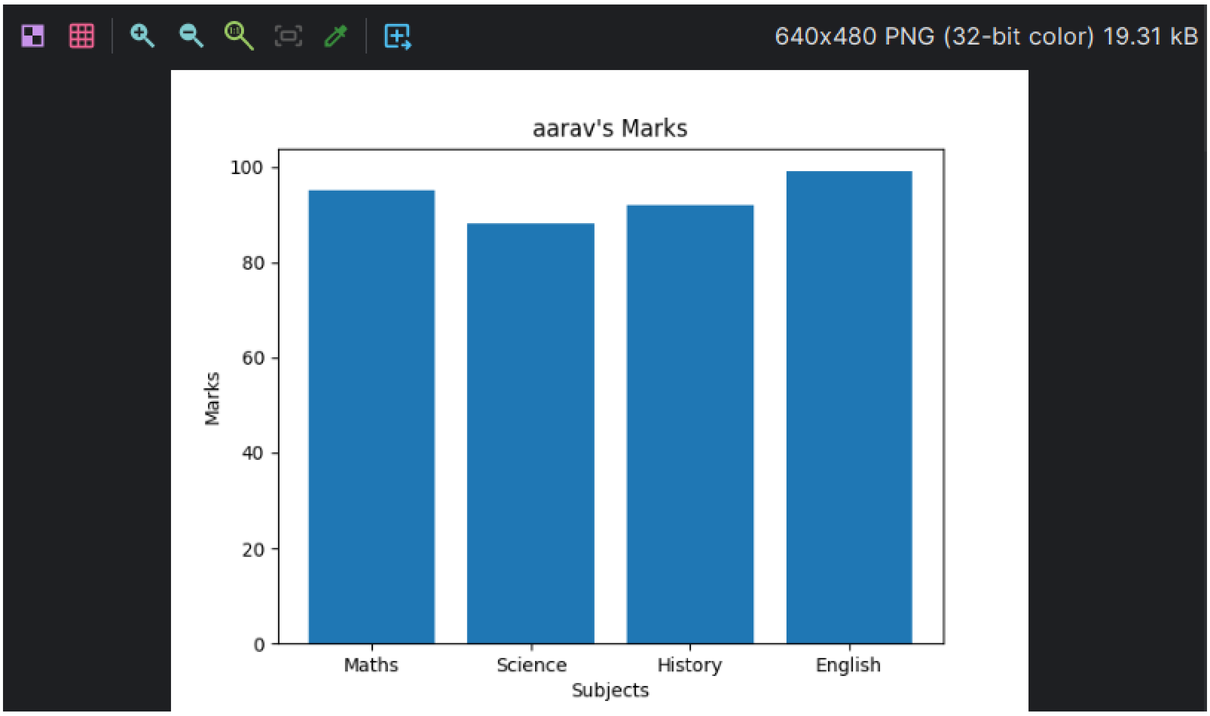
SCREENSHOTS

3. Report Card:

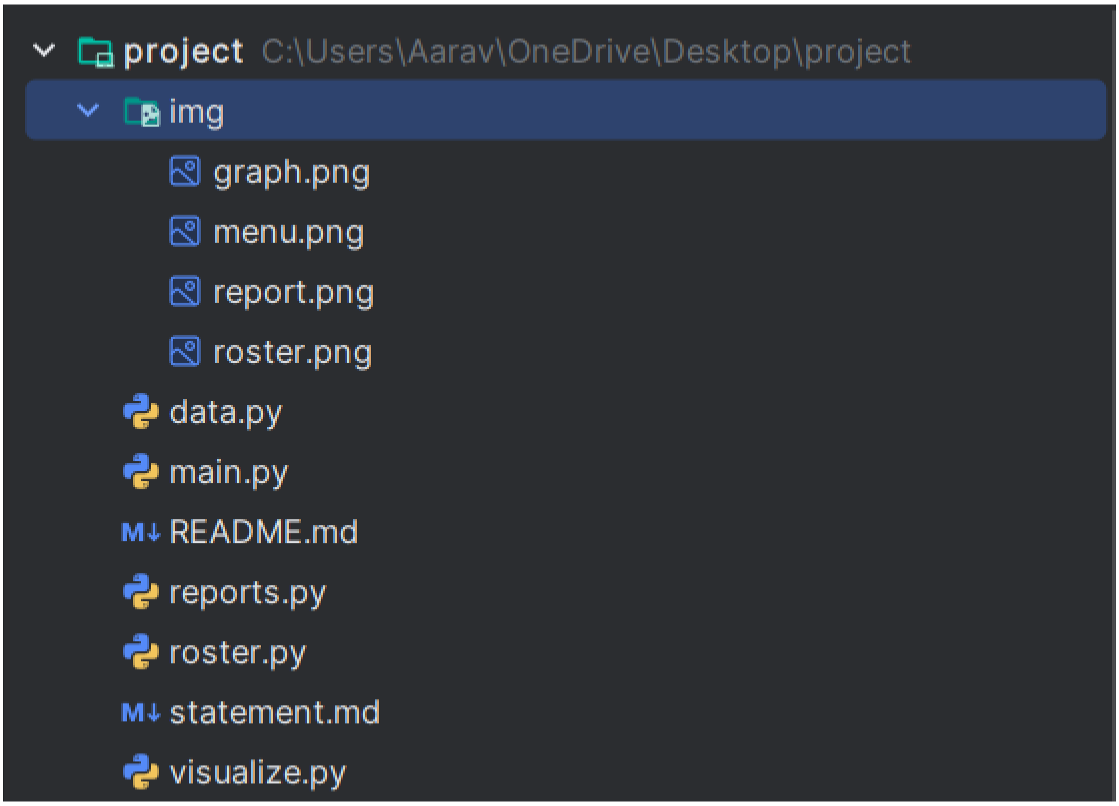


```
C:\Users\Aarav\AppData\Local\Programs\Python\Python313\python.exe main.py
SYSTEM LOCKED
Enter Password: 123
Access Granted.

Teacher's Gradebook:
1. View Class Roster
2. View Student Report Card
3. Visualize Student Marks
Please choose an option (1-3): 2

Enter student name: aarav

Report Card for Aarav
Maths : 95 S
Science : 88 B
History : 92 A
English : 99 S
>> Average marks: 93.5
>> Final Grade: A
>> Status: Pass

Process finished with exit code 0
```

4. Bar Graph:



FILE STRUCTURE



```
project  C:\Users\Aarav\OneDrive\Desktop\project
  img
    graph.png
    menu.png
    report.png
    roster.png
  data.py
  main.py
  README.md
  reports.py
  roster.py
  statement.md
  visualize.py
```

FUTURE ENHANCEMENTS

1. **Data saving:** Implementing CSV or SQL storage so data is
2. **GUI:** Replacing the Command Line Interface with a graphical window.

TECHNOLOGIES/TOOLS USED

1. Python3 (Programming language)

2. PyCharm (IDE)

3. Git, GitHub and GitKraken (Version Control)

REFERENCES

1. Matplotlib

   Matplotlib documentation

   Intro to Computational Physics (Module 1)

2. Markdown

   Markdown Guide