BACSE101 Problem Solving using Python

PROJECT REPORT

on

<Report Card Generator>

Prepared by

<Member 1 :Daivik> - <25BDS0272>
<Member 2 : Aaryavardhan> - <25BDS0283>
<Member 3 : Pradyumnan> - <25BDS0258>
<Member 3 : Malhar> - <25BDS0247>

Under the supervision of

<Jay Prajapati>

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering
Vellore Institute of Technology, Vellore.

October 27, 2025

Table of Contents

Abstract

*Our Python project is a comprehensive **Report Card Generation System** designed to efficiently manage and display student information in a structured and visually appealing manner.*

*The system is divided into four key modules — **Person**, **Academic**, **Library**, and **Main** — each serving a specific purpose.*

The Personal module stores and manages basic student details such as name, roll number, and contact information.

The Academic module handles subject-wise marks, grades, and performance analysis to generate an accurate report card.

The Library module maintains records of books issued, due dates, and fines, offering a complete overview of a student's library activity.

Finally, the Main module integrates data from all other modules and presents it in an organized, user-friendly interface that enhances readability and usability.

Together, these modules create a robust, modular, and efficient system that simplifies student record management and report generation for educational institutions.

1. Introduction

1.1 Domain Information

- This project falls under the domain of **Education and Student Management Systems**.
  It focuses on automating the process of generating student report cards, calculating grades, and maintaining academic records efficiently.
- It also gives the record of the books published and if any fine is applicable.
- The project helps in reducing manual errors, improving record-keeping, and providing quick access to student performance data.

1.2 Software Libraries Used

- **re** – Used for input validation and checking correct formats using regular expressions
- **math** - Used for performing mathematical calculations such as averages, percentages, and grade computation.

1.3 Contributions by Team Members

- Aaryavardhan : Created the “Person” module
- Malhar : Created the “library” module
- Daivik : Created the “academic” module
- Pradyumnan : Created the “main” file where the main code is written.

1.4 Challenges Faced

- Defining what all information is needed in a report card.
- Making the loops for calculating the information.

2. Problem Statement and Objectives

Problem Statement : The manual preparation of student report cards is time-consuming and prone to errors.

This project, **Student Report Card Generator**, automates the process by allowing users to enter marks and automatically calculate the **total**, **average**, and **grade** for each student efficiently.It also takes the information of the books issued by a student and if there is any fine to be paid.

Objectives:

- To develop a Python-based system that automates the generation of student report cards.
- To allow entry of the **number of subjects** and **marks** for each subject to calculate **total**, **average**, and **grade** automatically.
- To record **library details**, including the number and names of books issued and any **fine** to be paid.
- To reduce manual effort, minimize calculation errors, and make student record management more efficient and accurate.

3. Implementation

3.1 User Input System:

Allows users to enter the number of subjects and input marks for each subject.

CODE :

Academic module -
```
for i in range(n):
    subject = input(f"Enter subject {i+1} name: ")
        mark = float(input(f"Enter marks obtained in {subject} (out of 100): "))
    subjects.append(subject.title())
    marks.append(mark)
```
Library module -
```
for i in range(count):
    book = input(f"Enter book {i+1} name: ")
    issued_books.append(book.title())

    fine = float(input("Enter fine amount (if any): ₹"))
    return issued_books, fine
```

3.2 Automatic Calculation:

Calculates **total marks**, **average**, and **grade** automatically based on the entered data.

CODE :

```
def calculate_grade(avg):
    """Return grade based on average marks."""
    if avg >= 90:
        return "A+"
    elif avg >= 80:
        return "A"
    elif avg >= 70:
        return "B"
    elif avg >= 60:
        return "C"
    elif avg >= 50:
        return "D"
    else:
        return "F"
```

3.3 Error Handling :

Detects invalid inputs such as letters instead of numbers and prompts the user to re-enter correct information.

3.4 Library Management:

Records the **number of books issued**, their **names**, and calculates any **fine** to be paid by the student.

4. Demo Screenshots

Academic module :

```python
# academic.py
import math

def calculate_grade(avg):
    """Return grade based on average marks."""
    if avg >= 90:
        return "A+"
    elif avg >= 80:
        return "A"
    elif avg >= 70:
        return "B"
    elif avg >= 60:
        return "C"
    elif avg >= 50:
        return "D"
    else:
        return "F"

def get_academic_info():
    """Take subject details, calculate average and grade."""
    subjects = []
    marks = []

    n = int(input("Enter number of subjects: "))

    for i in range(n):
        subject = input(f"Enter subject {i+1} name: ")
        mark = float(input(f"Enter marks obtained in {subject} (out of 100): "))
        subjects.append(subject.title())
        marks.append(mark)

    avg = sum(marks) / n
    grade = calculate_grade(avg)

    academic_data = list(zip(subjects, marks))
    return academic_data, avg, grade
```

Library Module :

```python
# library.py

def get_library_info():
    """Collect library information."""
    issued_books = []
    count = int(input("Enter number of books issued: "))

    for i in range(count):
        book = input(f"Enter book {i+1} name: ")
        issued_books.append(book.title())

    fine = float(input("Enter fine amount (if any): ₹"))
    return issued_books, fine
```

Person Module :

```python
# person.py
import re

def validate_name(name):
    """Check if name contains only alphabets and spaces."""
    return bool(re.match(r"^[A-Za-z\s]+$", name))

def get_student_info():
    """Get and validate basic student details."""
    name = input("Enter student name: ")
    while not validate_name(name):
        name = input("Invalid name. Please enter only alphabets: ")

    roll_no = input("Enter roll number: ")
    dept = input("Enter department: ")

    return {"Name": name.title(), "Roll No": roll_no, "Department": dept}
```

Main Module :

```python
# main.py
from person import get_student_info
from academic import get_academic_info
from library import get_library_info

def display_report():
    print("\n====== STUDENT INFORMATION SYSTEM ======\n")

    student = get_student_info()
    academic_data, avg, grade = get_academic_info()
    issued_books, fine = get_library_info()

    print("\n---------- STUDENT DETAILS ----------")
    for key, value in student.items():
        print(f"{key}: {value}")

    print("\n---------- ACADEMIC PERFORMANCE ----------")
    print(f"{'Subject':<20}{'Marks':<10}")
    print("-" * 30)
    for sub, mark in academic_data:
        print(f"{sub:<20}{mark:<10.2f}")
    print("-" * 30)
    print(f"Average: {avg:.2f}")
    print(f"Grade: {grade}")

    print("\n---------- LIBRARY RECORD ----------")
    if issued_books:
        print("Books Issued:")
        for b in issued_books:
            print(f" - {b}")
    else:
        print("No books issued.")
    print(f"Total Fine: ₹{fine:.2f}")

    print("\n========================================")
    print("Student Information generated successfully!\n")

if __name__ == "__main__":
    display_report()
```

Result :

```
---------- STUDENT DETAILS ----------
Name: Daivik
Roll No: 25BDS0272
Department: Data Science

---------- ACADEMIC PERFORMANCE ----------
Subject             Marks
------------------------------
Maths               90.00
Python              90.00
Chemistry           90.00
Basic Engineering   90.00
------------------------------
Average: 90.00
Grade: A+

---------- LIBRARY RECORD ----------
Books Issued:
 - Chem Reference Book
 - Math Reference Book
Total Fine: ₹0.00

==========================================
Student Information generated successfully!
```

5. Conclusion

The **Student Report Card Generator** project successfully automates the process of managing student academic and library records.

It allows users to enter subject marks, calculate the total, average, and grade, and also record details of books issued along with any fines.

By replacing manual calculations with an automated system, the project minimizes human errors, saves time, and ensures accuracy in student performance evaluation.

Overall, it provides an efficient, reliable, and user-friendly solution for maintaining student information in educational institutions.