1. Introduction

This report presents the development of a command-line Student Gradebook Application written in Python. The project aims to manage student course data, store information persistently, and calculate weighted GPA.

2. Objectives

- Create a CLI application that manages course information.
- Store data permanently using JSON.
- Support adding, editing, deleting, and viewing courses.
- Calculate weighted GPA accurately.

3. System Design

3.1 Architecture

Programming language: Python

Storage: JSON file

Application is divided into individual functions:

- add_course()
- update_course()
- delete_course()
- view_courses()
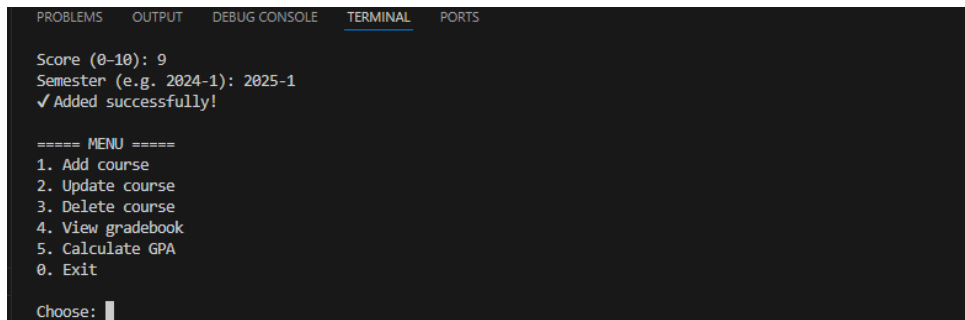- calculate_gpa()

3.2 Data Format

Example gradebook.json:

[

  {

    "code": "MATH101",

    "name": "Calculus",

    "credits": 3,

    "semester": "2024-1",

    "score": 9
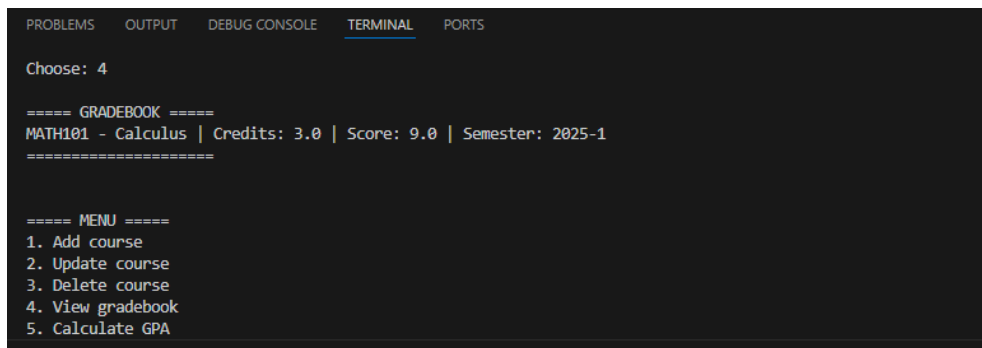
  }

]

4. Features and Screenshots

Add Course

View Courses

Update/Delete

```
Choose: 2
Enter course code to update: MATH101
Editing [Calculus]
New course name: & "E:/hoc code/python.exe" c:/my_flask_app/main.py
New credits: 4
New semester: 2025-2
New score (0-10): 10
✓Updated!
```

```
Choose: 3
Enter course code to delete: MATH101
✓Deleted!

===== MENU =====
1. Add course
2. Update course
3. Delete course
4. View gradebook
5. Calculate GPA
0. Exit

Choose: ▌
```

Calculate GPA

```
0. Exit

Choose: 5
📌 Overall GPA: 9.0

===== GPA BY SEMESTER =====
2025-2: 9.0
==========================
```

5. Testing

- Input duplicate course code → system rejects it.

- Empty gradebook → system prints "No courses found".

- Data remains after restarting → persistent storage works.

6. Known Issues

- No GUI (because this is Extra A only).

- User must enter correct numeric values or app may crash (can be improved with validation).

7. Conclusion

The CLI application fulfills the requirements of Extra A. It manages course data, stores information using JSON, and calculates weighted GPA. The project can be expanded next with GUI (Extra C) or web interactivity.