Student Management System - Documentation

# Problem Statement

The increasing need for efficient and organized educational systems has made manual student management outdated, error-prone, and time-consuming. Educational institutions require a digital system that can manage student registrations, course enrollments, and examination records in a seamless and structured manner.  
  
This project aims to create a console-based Student Management System in Java that supports two types of users—Admin and Student. Admins should be able to manage courses and subjects, while students can register, choose courses and subjects, take a basic exam, and view their results.

# Objectives

1. Implement Object-Oriented Programming (OOP) concepts in Java such as inheritance, encapsulation, and class interaction.

2. Provide an interface for Admins to:

- Add new courses.  
 - Add subjects under each course.  
 - View the list of registered students.  
 - View student exam results.

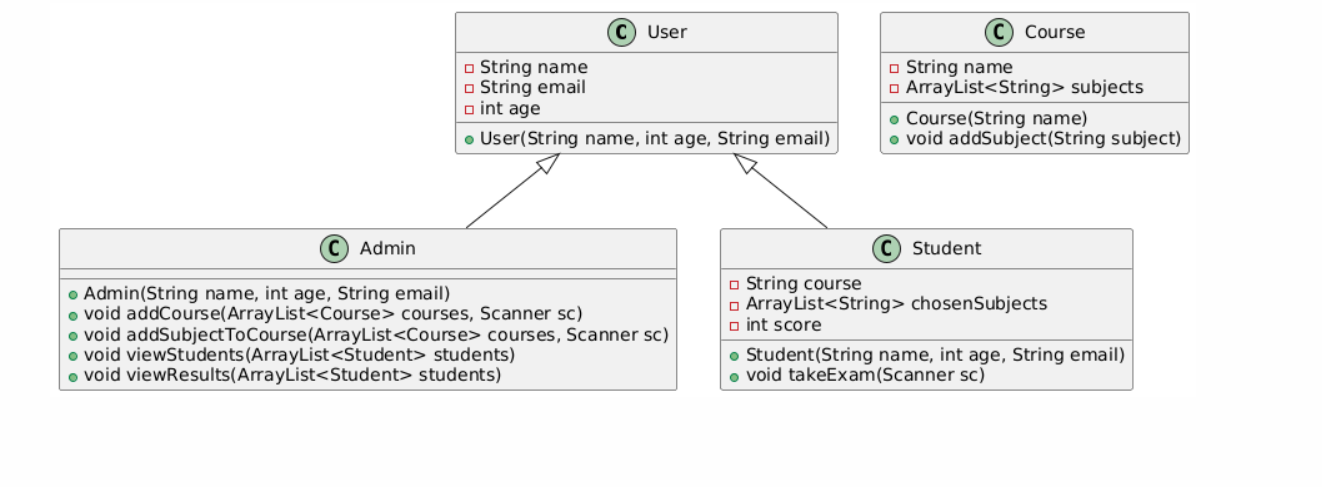
3. Allow Students to:

- Register with personal details.  
 - Select a course and subjects.  
 - Take a multiple-choice question (MCQ) based exam.  
 - View their exam results and pass/fail status.

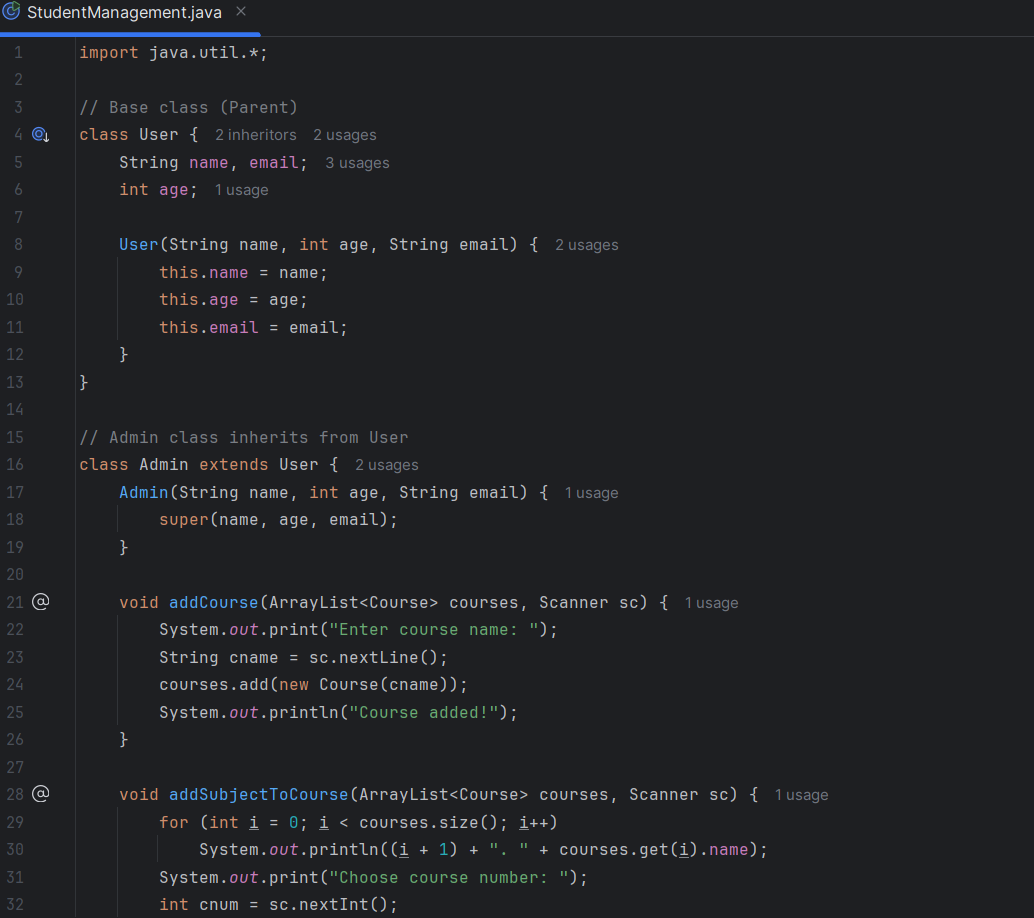
4. Ensure basic error handling like choosing available courses and taking the exam only once.

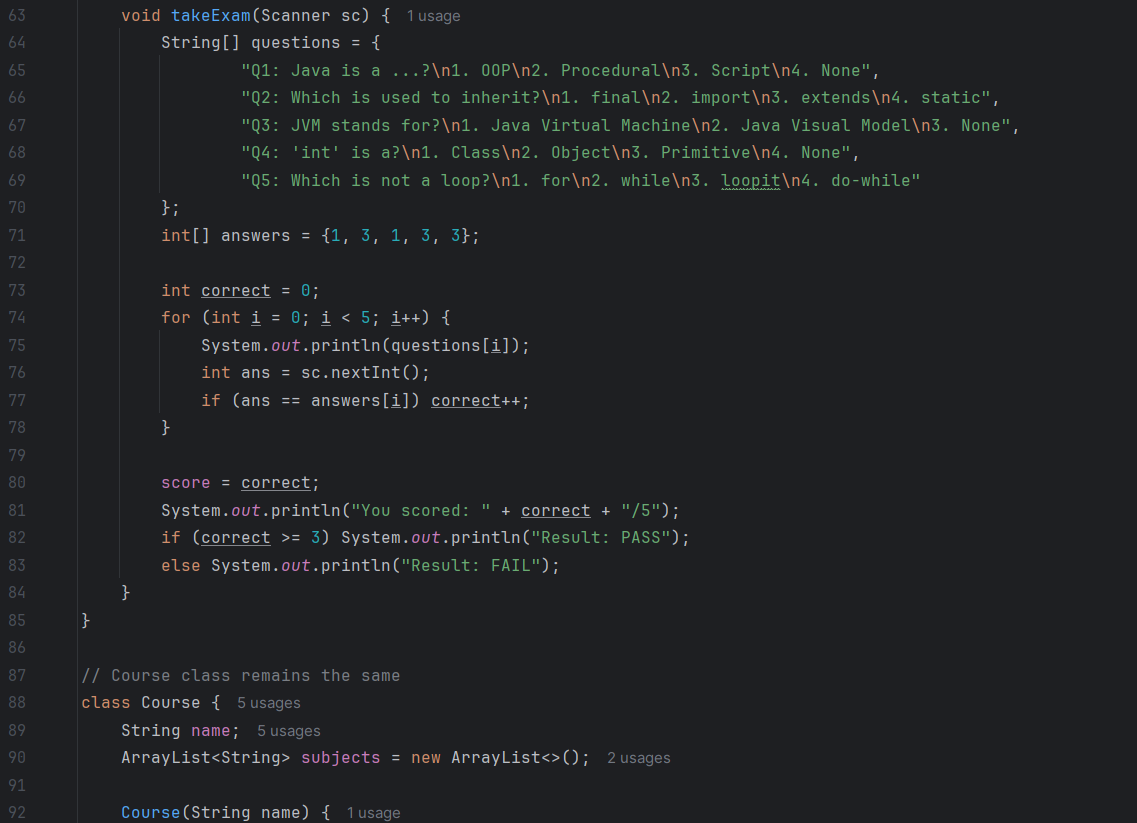
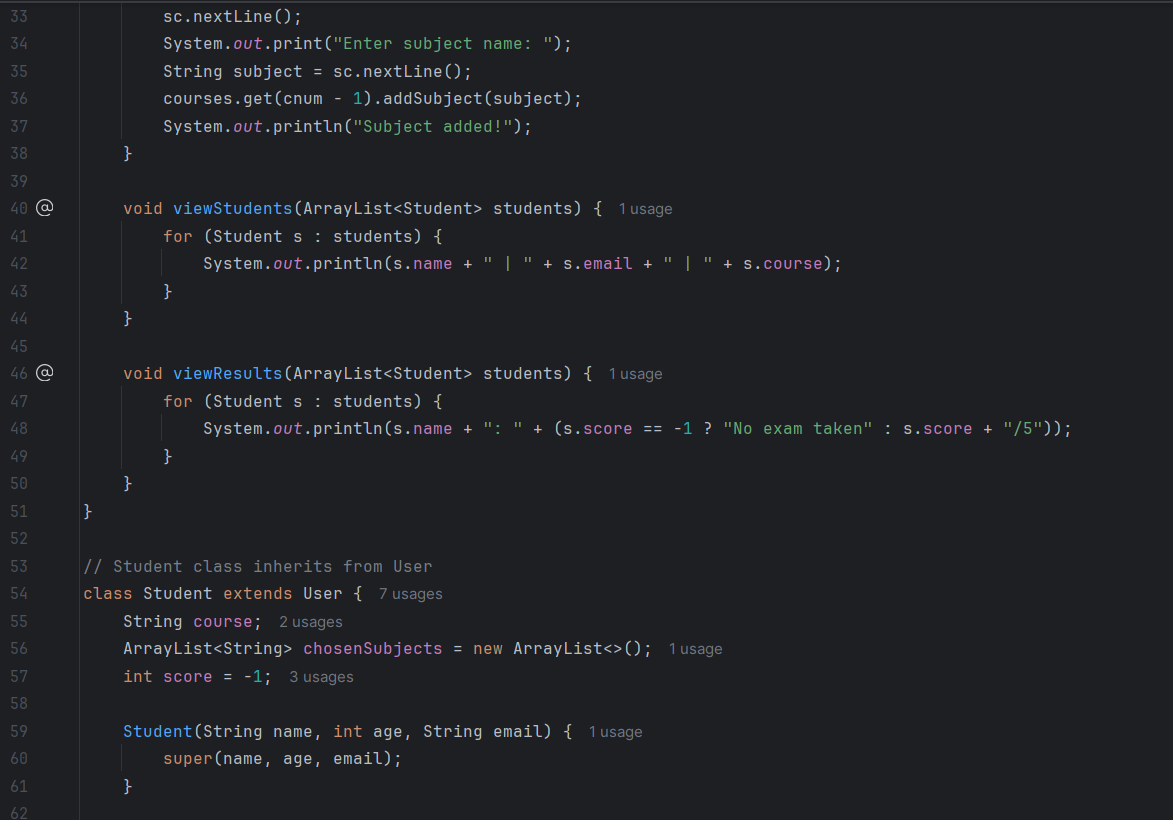
5. Make the application simple, interactive, and console-based for easy testing and use.

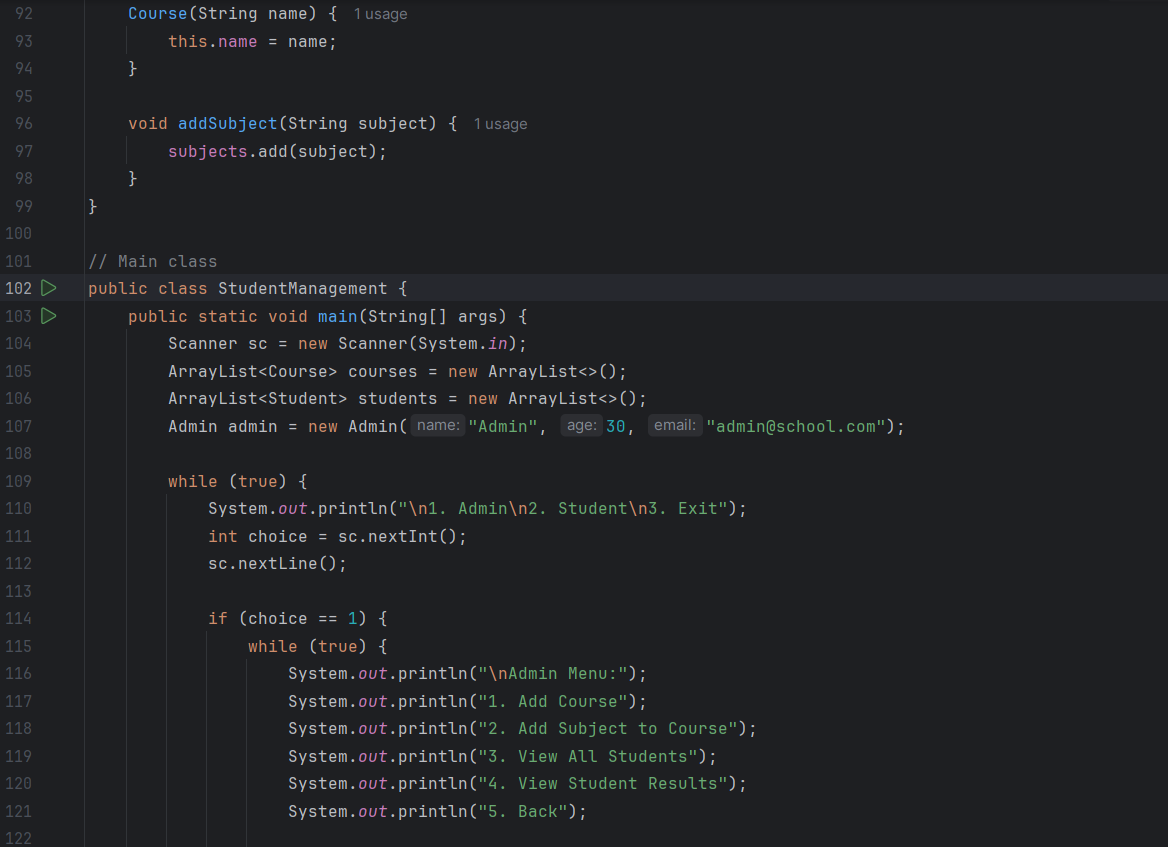
# Class Diagram



# Code

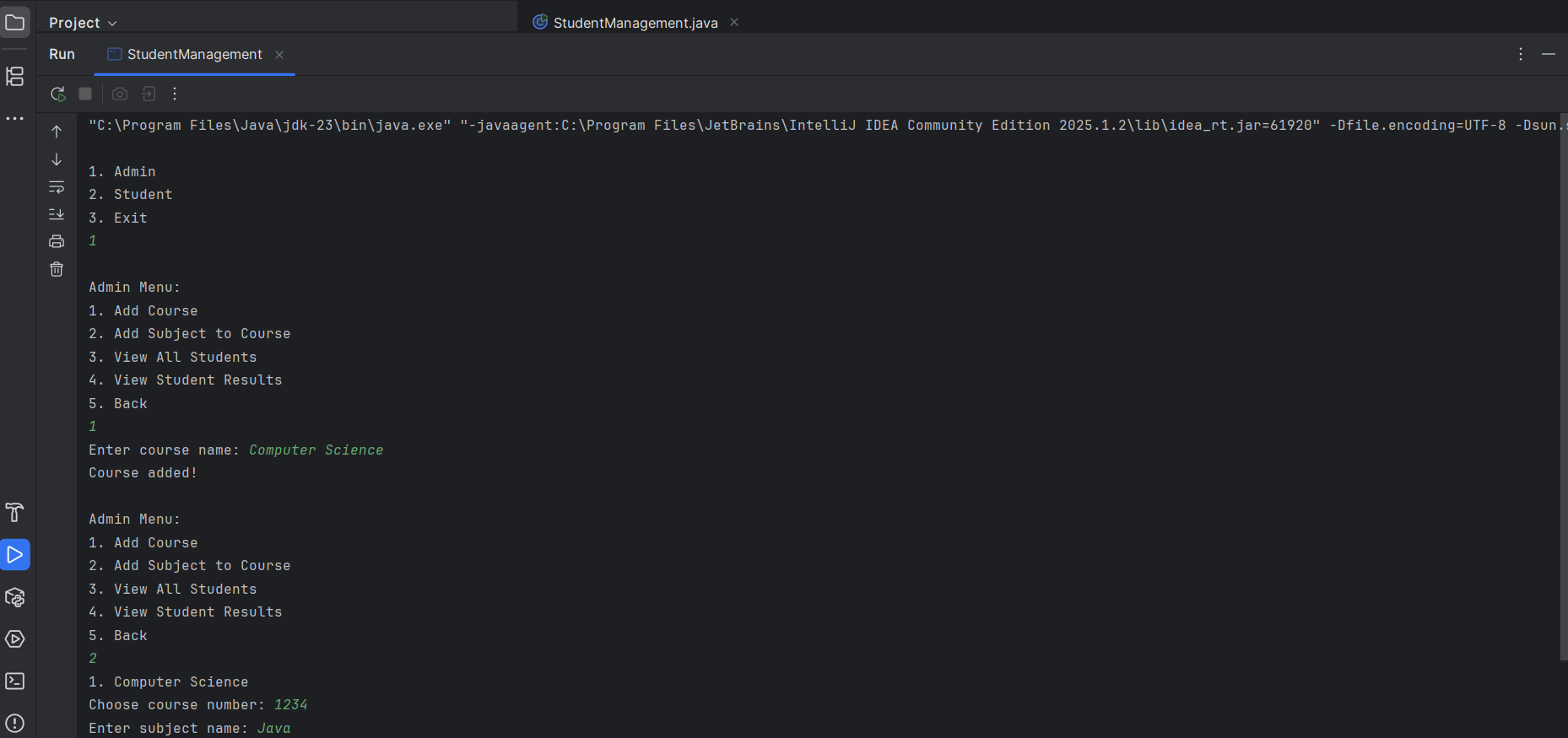


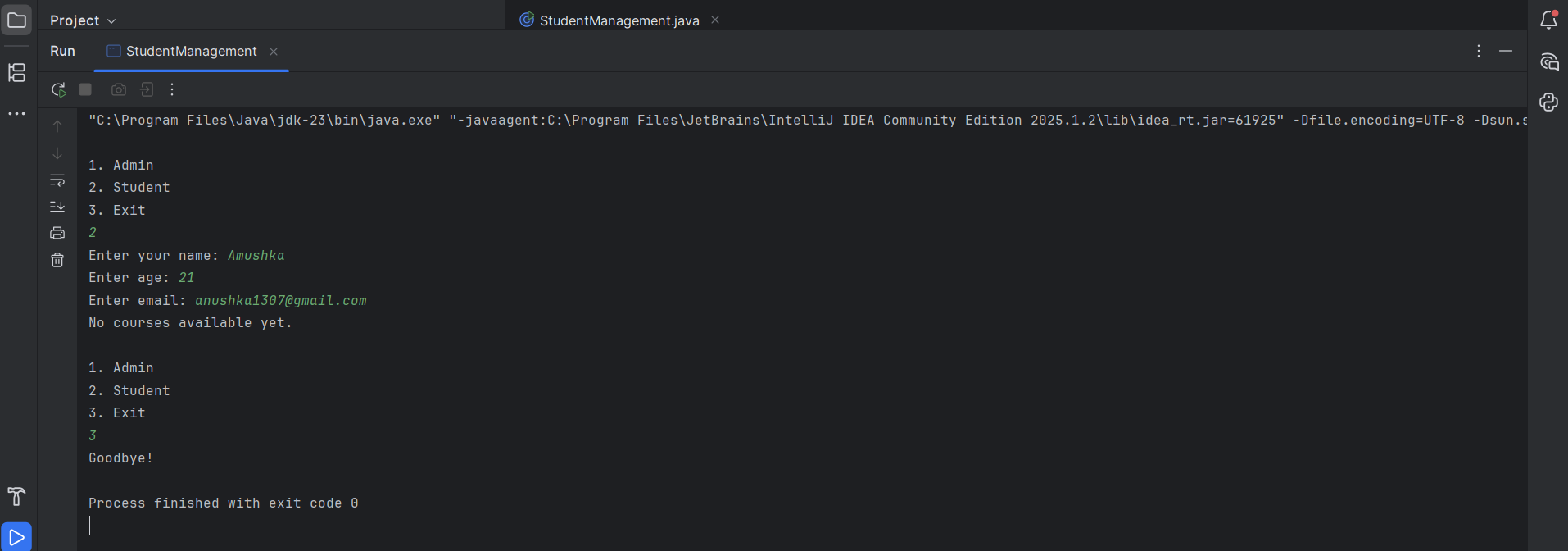






**Output**





# Conclusion

This project successfully demonstrates a basic Student Management System using Java and OOP principles. The program simulates real-life scenarios of course selection, subject management, and student evaluation. The use of inheritance through a base User class for both Admin and Student showcases re-usability and proper class design. It also improves code clarity and maintenance.  
  
The system can be extended in the future with advanced features like file/database storage, user authentication, and even a graphical interface (GUI) for enhanced usability. Overall, the project achieves its goals of learning, designing, and applying OOP concepts in a real-world mini project.