

COLLEGE MANAGEMENT SYSTEM



A PROJECT REPORT

Submitted by

MUKIL SUBRAMANI S G(8115U23AM029)

in partial fulfillment of requirements for the award of the course

CGB1201 - JAVA PROGRAMMING

in

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

DECEMBER - 2024

**K. RAMAKRISHNAN COLLEGE OF ENGINEERING
(AUTONOMOUS)**

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on “**COLLEGE MANAGEMENT SYSTEM**” is the bonafide work of **MUKIL SUBRMANI S G (8115U23AM029)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE

Dr. B.KIRAN BALA B.Tech.,M.E.,
M.B.A.,Ph.D., M.I.S.T.E.,U.A.C.E.E.,
IAENG

HEAD OF THE DEPARTMENT,

Department of Artificial Intelligence
and Machine Learning,

K.Ramakrishnan College of
Engineering (Autonomous)

Samayapuram,Trichy–621112.

SIGNATURE

Mrs. GEETHA M.E,

ASSISTANT PROFESSOR

Department of Artificial Intelligence
and Data Science,

K. Ramakrishnan College of
Engineering (Autonomous)

Samayapuram,Trichy–621112.

Submitted for the viva-voce examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **“COLLEGE MANAGEMENT SYSTEM”** is the result of original work done by me and best of my knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201 - JAVA PROGRAMMING**.

Signature

MUKIL SUBRAMANI S G

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in debtedness to our institution, **“K.RAMAKRISHNAN COLLEGE OF ENGINEERING(Autonomous)”**, for providing me with the opportunity to do this project. I extend our sincere acknowledgment and appreciation to the esteemed and honorable Chairman, **Dr. K. RAMAKRISHNAN, B.E.**, for having provided the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director , **Dr.S. KUPPUSAMY, MBA, Ph.D.**, for forwarding our project and offering an adequate duration to complete it. I would like to thank **Dr. D. SRINIVASAN, M.E., Ph.D., FIE., MIIW., MISTE., MISAE., C. Engg.**, Principal, who gave the opportunity to frame the project to full satisfaction.

Dr. B. KIRAN BALA, B.Tech., M.E., M.B.A., Ph.D., M.I.S.T.E., U.A.C.E.E., IAENG, Head of the Department of Artificial Intelligence and Machine Learning, for providing his encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs. P. GEETHA, M.E.**, Department of Artificial Intelligence and Data Science, for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work .

INSTITUTE VISION AND MISSION

VISION OF THE INSTITUTE:

To achieve a prominent position among the top technical institutions.

MISSION OF THE INSTITUTE:

M1: To be show standard technical education excellence through state of the art infrastructure, competent faculty and high ethical standards.

M2: To nurture research and entrepreneurial skills among students in cuttingedge technologies.

M3: To provide education for developing high-quality professionals to transformthe society.

DEPARTMENT VISION AND MISSION

DEPARTMENT OF CSE(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

Vision of the Department

To become a renowned hub for Artificial Intelligence and Machine Learning Technologies to produce highly talented globally recognizable technocrats tomeet Industrial needs and societal expectations.

Mission of the Department

M1: To impart advanced education in Artificial Intelligence and MachineLearning, Built upon a foundation in Computer Science and Engineering.

M2: To foster Experiential learning equips students with engineering skills toTackle real-world problems.

M3: To promote collaborative innovation in Artificial Intelligence, machine

Learning, and related research and development with industries.

M4: To provide an enjoyable environment for pursuing excellence while upholding Strong personal and professional values and ethics.

Programme Educational Objectives (PEOs):

Graduates will be able to:

PEO1: Excel in technical abilities to build intelligent systems in the fields of Artificial Intelligence and Machine Learning in order to find new opportunities.

PEO2: Embrace new technology to solve real-world problems, whether alone or As a team, while prioritizing ethics and societal benefits.

PEO3: Accept lifelong learning to expand future opportunities in research and Product development.

Programme Specific Outcomes (PSOs):

PSO1: Ability to create and use Artificial Intelligence and Machine Learning Algorithms, including supervised and unsupervised learning, reinforcement Learning, and deep learning models.

PSO2: Ability to collect, pre-process, and analyze large datasets, including data Cleaning, feature engineering, and data visualization..

PROGRAM OUTCOMES(POs)

Engineering students will be able to:

1.Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2.Problem analysis: Identify, formulate, review research literature, and an

alyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

3.Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations

4.Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5.Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

6.The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

7.Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8.Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9.Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10.Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11.Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12.Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The **College Management System** is designed to streamline and automate essential academic and administrative processes in a college. This system enables efficient management of user records (students and faculty), courses, attendance tracking, and grade assignments. Traditional methods of handling such tasks often involve manual recordkeeping, which is time-consuming, error-prone, and inefficient. The College Management System provides a digital solution to overcome these challenges. It allows administrators to manage users and courses while providing tools for faculty to record attendance and grades. Key features include user and course management, real-time attendance tracking, and the ability to generate and view grade reports.

The system is implemented using Java programming, leveraging object-oriented principles such as encapsulation, inheritance, and polymorphism for scalability and maintainability. Future enhancements may include integrating a database for persistent storage, providing role-based authentication, and enabling online portals for students and faculty to access their information. Overall, the project demonstrates the power of programming in solving real-world challenges effectively.

ABSTRACT WITH POs AND PSOs MAPPING

ABSTRACT	POs MAPPED	PSOs MAPPED
<p>The College Management System provides a digital solution to overcome these challenges. It allows administrators to manage users and courses while providing tools for faculty to record attendance and grades. Key features include user and course management, real-time attendance tracking, and the ability to generate and view grade reports.</p>	<p>PO1 -3</p> <p>PO2 -3</p> <p>PO3 -3</p> <p>PO4 -3</p> <p>PO5 -3</p> <p>PO6 -3</p> <p>PO7 -3</p> <p>PO8 -3</p> <p>PO9 -3</p> <p>PO10 -3</p> <p>PO11-3</p> <p>PO12 -3</p>	<p>PSO1 -3</p> <p>PSO2 -3</p>

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	
1	INTRODUCTION	1
	1.1 Objective	1
	1.2 Overview	1
	1.3 Java Programming concepts	2
2	PROJECT METHODOLOGY	4
	2.1 Proposed Work	4
	2.2 Block Diagram	5
3	MODULE DESCRIPTION	6
	1 3.1 User Management Module	6
	3.2 Course Management Module	6
	3.3 Attendance Management Module	7
	3.4 Grade Management Module	7
	3.5 Reports and Analytics Module	7
4	CONCLUSION & FUTURE SCOPE	9
	4.1 Conclusion	9
	4.2 Future Scope	9
	APPENDIX A (SOURCE CODE)	11
	APPENDIX B (SCREENSHOTS)	18
	REFERENCES	20

CHAPTER 1

INTRODUCTION

1.1 Objective

The College Management System aims to design and implement a software solution that automates key academic and administrative operations in a college. Traditional manual methods for managing users, courses, attendance, and grades are not only time-consuming but also prone to human error. This system aims to provide a reliable and efficient digital platform to overcome these challenges and ensure that college processes are streamlined.

The system is built with the goal of improving operational efficiency, enhancing data accuracy, and reducing administrative workload. By automating processes like user and course management, attendance tracking, and grade recording, this system allows administrators and faculty to focus more on academic activities. With its scalable and modular design, the system lays the foundation for future enhancements, such as online student portals, database integration, and role-based access controls.

1.2 Overview

The College Management System is an automated platform designed to simplify and enhance the management of college operations. It serves as a comprehensive solution for handling user records (students and faculty), creating and managing courses, recording attendance, and maintaining grades. This program provides a structured way to store, retrieve, and process data, ensuring that all essential college operations

are handled efficiently. Its intuitive menu-driven interface allows users to interact seamlessly with the system.

The system is implemented using Java programming and employs the principles of ObjectOriented Programming (OOP). The use of dynamic data structures such as ArrayList ensures that the system can handle real-time operations effectively. The modular design of the program makes it adaptable for future improvements, such as integrating persistent storage and advanced analytics. This project demonstrates how Java programming can be utilized to develop a practical and impactful application for educational institutions.

1.3 Java Programming Concepts

- The College Management System is developed using core Java programming concepts, with a strong emphasis on Object-Oriented Programming (OOP) principles. Classes such as User, Course, Attendance, and Grade represent real-world entities, encapsulating their attributes and behaviors. Encapsulation ensures that sensitive data (like IDs and names) is securely stored within these classes, accessible only through getter and setter methods. This approach protects the integrity of the data and promotes reusability of code. The modular design of the system, achieved through OOP, allows for easy expansion to include new functionalities in the future.
- The program employs essential control structures such as loops (for, while) and conditional statements (if, switch) to handle user input and execute the corresponding logic. Exception handling is implemented to gracefully manage invalid inputs, such as non-numeric data for IDs, ensuring the robustness of the system. Data is stored and managed using Java's dynamic data structures like ArrayList for users, courses, and attendance records. These structures enable efficient addition, deletion, and retrieval of data, enhancing the overall performance of the system.

- String handling is another crucial aspect of the program, used extensively for displaying menus, formatted outputs, and messages to the user. The modularized code, with methods like `manageUsers()` and `manageCourses()`, adheres to Java's best practices, ensuring clean, maintainable, and scalable code. By integrating these Java programming concepts, the system showcases how foundational programming techniques can solve real-world problems in an educational context.

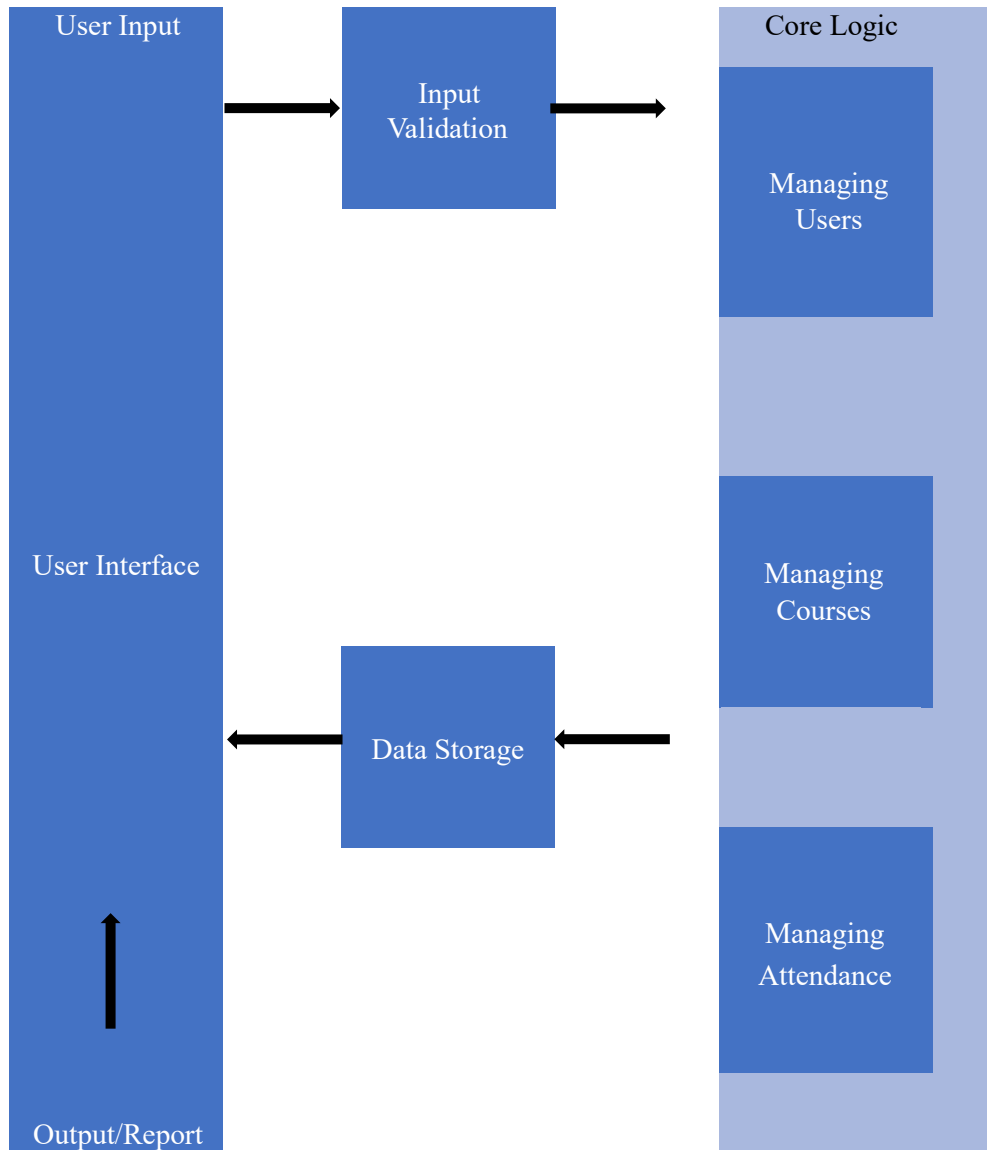
CHAPTER 2

PROJECT METHODOLOGY

2.1 Proposed Work

- The College Management System is designed to provide an efficient and automated solution for managing the essential operations of a college. The proposed system replaces traditional manual methods with a digital platform that allows administrators and faculty to manage users, courses, attendance, and grades. The system features a menu-driven interface that simplifies user interaction and ensures that tasks like adding users, recording attendance, and assigning grades are completed efficiently and accurately.
- The system is built using Java, leveraging the principles of Object-Oriented Programming (OOP) for scalability and maintainability. Data is stored in dynamic structures like ArrayList and HashMap, enabling the program to handle real-time operations effectively. Input validation ensures the accuracy of data, while exception handling provides robustness by preventing system crashes due to invalid inputs. These features make the system a reliable and practical solution for colleges.
- Future enhancements include integrating a database for persistent data storage, enabling online portals for students and faculty, and implementing role-based access control for improved security. The modular structure of the program ensures that these enhancements can be incorporated seamlessly. Overall, the proposed work aims to create a system that not only simplifies administrative tasks but also provides a solid foundation for future technological advancements.

2.2 Block Diagram



CHAPTER 3

MODULE DESCRIPTION

3.1 User Management Module

The User Management Module plays a central role in the College Management System by enabling administrators to efficiently manage the institution's members. This module allows for the addition of new users, including both students and faculty, with each user assigned a unique ID to ensure data integrity. Administrators can record details such as the user's name and role, which can either be "Student" or "Faculty." Additionally, this module includes functionality to view a complete list of registered users, providing a clear and organized overview of all users in the system. This ensures easy access to information and simplifies the management of user data.

3.2 Course Management Module

The Course Management Module provides a comprehensive system for creating and maintaining the institution's course offerings. Faculty or administrators can create new courses by assigning them unique IDs and descriptive names, ensuring that courses are distinct and easily identifiable. This module also offers a feature to view all available courses, making it simple for administrators and faculty to keep track of current offerings. By providing a structured way to manage course data, this module ensures that information is accurate, up-to-date, and readily accessible whenever required.

3.3 Attendance Management Module

The Attendance Management Module is designed to track and monitor student participation in classes. Faculty members can use this module to record attendance for students in specific courses, marking them as either "Present" or "Absent" based on their participation. This module also allows administrators and faculty to view attendance records, both for individual students and entire courses. By providing a clear overview of attendance trends, this module helps faculty identify students with low attendance and take proactive measures to address potential issues, such as academic counseling or parental notifications.

3.4 Grade Management Module

The Grade Management Module streamlines the process of recording and managing student grades. Faculty can assign grades to students for specific courses, ensuring that each student's academic performance is accurately recorded. Grades are stored in a structured format, making them easily retrievable for future analysis or reporting. This module helps faculty evaluate individual student performance and identify areas where students may need additional support. It also aids administrators in generating academic reports and monitoring overall performance trends within the institution.

3.5 Reports and Analytics Module

The Reports and Analytics Module enhances decision-making by generating detailed reports and insights based on the data stored in the system. This module provides attendance summaries that highlight participation trends across courses, helping faculty and administrators identify students with irregular attendance. Grade reports offer

insights into the overall academic performance of students and courses, making it easier to identify top-performing students or those at risk of failing. Additionally, the module generates statistical summaries, such as the number of active courses, the total number of users, and trends in academic performance. These analytics empower the administration and faculty to make data-driven decisions for improving academic outcomes and operational efficiency.

CHAPTER 4

CONCLUSION & FUTURE SCOPE

4.1 CONCLUSION

The College Management System is an efficient and robust solution for managing essential academic and administrative operations in a college. By automating tasks like user management, course creation, attendance tracking, and grade reporting, the system addresses the inefficiencies of traditional manual methods. Its menu-driven interface and modular design ensure that users can interact with the system intuitively and that administrators can manage their tasks effectively.

The system's scalability allows it to adapt to future enhancements, such as integrating databases or developing online portals for students and faculty. By utilizing Java programming concepts and following best practices in software development, this project demonstrates the potential of technology to simplify complex processes and improve operational efficiency in educational institutions.

4.2 FUTURE SCOPE

The College Management System is designed with scalability and future enhancements in mind. In its current form, the system provides a reliable and efficient solution for managing users, courses, attendance, and grades. However, the system can be expanded further to meet evolving institutional needs. Integrating a database for persistent storage would ensure that user data, attendance records, and grades are retained even after the system is restarted, enhancing its reliability for long-term usage.

Future developments could include the addition of role-based access control, where students, faculty, and administrators have different levels of access to system functionalities. An online portal could also be developed, allowing students to view their attendance and grades remotely, while faculty could manage records more conveniently. By incorporating these features, the system could evolve into a comprehensive digital platform for managing all aspects of college operations.

APPENDIX A

(SOURCE CODE)

```
import
java.util.ArrayList;
import java.util.List;
import
java.util.Scanner;

// Class to represent a user (student or
faculty) class User {
    int id;
    String name, role;

    User(int id, String name, String
role) {        this.id = id;
this.name = name;        this.role =
role;
    }
}

// Class to represent a course
class Course {
    int id;
    String name;
```

```
    Course(int id, String
name) {      this.id = id;
this.name = name;
    }
}
```

```
// Class to
```

```
    represent
```

```
    attendance
```

```
    class
```

```
    Attendance {
```

```
        int studentId;
```

```
        int courseId;
```

```
        String status;
```

```
    Attendance(int studentId, int courseId, String
status) {      this.studentId = studentId;
this.courseId = courseId;      this.status =
status;
    }
}
```

```
// Class to represent
```

```
grades class Grade {
```

```
    int studentId;    int
```

```
    courseId;    String
```

```
    grade;
```

```
    Grade(int studentId, int courseId, String
grade) {      this.studentId = studentId;
```

```
this.courseId = courseId;    this.grade =  
grade;  
    }  
}
```

```
// Main College Management System class public class  
CollegeManagementSystem {    private static List<User> users  
= new ArrayList<>();    private static List<Course> courses =  
new ArrayList<>();    private static List<Attendance>  
attendanceRecords = new ArrayList<>();    private static  
List<Grade> grades = new ArrayList<>();    private static int  
userIdCounter = 1;    private static int courseIdCounter = 1;
```

```
    public static void main(String[] args) {  
Scanner scanner = new  
Scanner(System.in);    int choice;
```

```
        System.out.println("Welcome to the College Management System!");
```

```
do {  
    System.out.println("\nMain Menu:");  
    System.out.println("1. Manage Users");  
    System.out.println("2. Manage Courses");  
    System.out.println("3. Manage Attendance");  
    System.out.println("4. Manage Grades");  
    System.out.println("5. Exit");  
System.out.print("Enter your choice: ");  
choice = scanner.nextInt();
```



```

        switch (choice)
        {
            case 1:

manageUsers(scanner);
break;          case 2:

manageCourses(scanner);
break;          case 3:

manageAttendance(scanner);
break;          case 4:

manageGrades(scanner);
break;          case 5:

                System.out.println("Exiting... Thank
you!");          break;

            default:

                System.out.println("Invalid choice. Please try again.");
        }
    } while (choice != 5);

    scanner.close();
}

private static void manageUsers(Scanner scanner) {
    System.out.println("\nUser Management:");
    System.out.println("1. Add User");
    System.out.println("2. View
Users");    System.out.print("Enter

```

```

your choice: ");    int choice =
scanner.nextInt();
scanner.nextLine(); // Consume newline

    switch (choice) {
case 1:
        System.out.print("Enter user name: ");
        String name = scanner.nextLine();
        System.out.print("Enter role
(Student/Faculty): ");    String role =
scanner.nextLine();
        users.add(new User(userIdCounter++,
name, role));    System.out.println("User
added successfully!");    break;    case
2:
        System.out.println("User
List:");    for (User user :
users) {
            System.out.println("ID: " + user.id + ", Name: " + user.name + ", Role:
" + user.role);
        }
        break;
default:
        System.out.println("Invalid choice.");
    }
}

private static void manageCourses(Scanner scanner) {
    System.out.println("\nCourse Management:");
    System.out.println("1. Add Course");

```

```

        System.out.println("2. View
Courses");    System.out.print("Enter
your choice: ");    int choice =
scanner.nextInt();
scanner.nextLine(); // Consume newline

        switch (choice) {
case 1:
            System.out.print("Enter course name:
");    String courseName =
scanner.nextLine();
            courses.add(new Course(courseIdCounter++,
courseName));    System.out.println("Course added
successfully!");    break;    case 2:
            System.out.println("Course
List:");    for (Course course :
courses) {
                System.out.println("ID: " + course.id + ", Name: " + course.name);
            }
            break;
default:
            System.out.println("Invalid choice.");
        }
    }

    private static void manageAttendance(Scanner scanner) {
        System.out.println("\nAttendance
Management:");    System.out.print("Enter
student ID: ");    int studentId =
scanner.nextInt();

```

```

System.out.print("Enter course ID: ");    int
courseId = scanner.nextInt();
scanner.nextLine(); // Consume newline

    System.out.print("Enter attendance status
(Present/Absent): ");    String status = scanner.nextLine();
    attendanceRecords.add(new Attendance(studentId, courseId, status));
    System.out.println("Attendance recorded successfully!");
}

private static void manageGrades(Scanner scanner) {
    System.out.println("\nGrade
Management:");
    System.out.print("Enter student ID: ");
    int studentId = scanner.nextInt();
    System.out.print("Enter course ID: ");
    int courseId = scanner.nextInt();
    scanner.nextLine(); // Consume newline
    System.out.print("Enter grade: ");
    String grade = scanner.nextLine();
    grades.add(new Grade(studentId, courseId, grade));
    System.out.println("Grade recorded successfully!");
}
}

```

APPENDIX B (SCREENSHOTS)

```
Welcome to the College Management System!
```

```
Main Menu:
```

1. Manage Users
2. Manage Courses
3. Manage Attendance
4. Manage Grades
5. Exit

```
Enter your choice: 1
```

```
User Management:
```

1. Add User
2. View Users

```
Enter your choice: 1
```

```
Enter user name: Sara
```

```
Enter role (Student/Faculty): Student
```

```
User added successfully!
```

```
Welcome to the College Management System!
```

```
Main Menu:
```

1. Manage Users
2. Manage Courses
3. Manage Attendance
4. Manage Grades
5. Exit

```
Enter your choice: 2
```

```
Course Management:
```

1. Add Course
2. View Courses

```
Enter your choice: 1
```

```
Enter course name: Java
```

```
Course added successfully!
```

Welcome to the College Management System!

Main Menu:

1. Manage Users
2. Manage Courses
3. Manage Attendance
4. Manage Grades
5. Exit

Enter your choice: 3

Attendance Management:

Enter student ID: 29

Enter course ID: 2

Enter attendance status (Present/Absent): Present

Attendance recorded successfully!

Welcome to the College Management System!

Main Menu:

1. Manage Users
2. Manage Courses
3. Manage Attendance
4. Manage Grades
5. Exit

Enter your choice: 4

Grade Management:

Enter student ID: 29

Enter course ID: 2

Enter grade: 0

Grade recorded successfully!

REFERENCES

Deitel, H. M., & Deitel, P. J. (2017). Java: How to Program (11th ed.). Pearson. This book provides a comprehensive guide to Java programming, covering object-oriented principles, GUI development, and exception handling, which were crucial for the development of this system.

Schildt, H. (2019). Java: The Complete Reference (11th ed.). McGraw-Hill Education.

Schildt's reference book was instrumental in understanding Java syntax, AWT, and other fundamental concepts utilized in this project.

Oracle. (n.d.). Java Documentation. Retrieved from <https://docs.oracle.com/en/java/> The official Java documentation from Oracle was used to reference classes and methods related to AWT components, event handling, and general Java functionality.