



COLLEGE MANAGEMENT SYSTEM



A PROJECT REPORT

Submitted by

GOWSIC PRABHA K (8115U23EC029)

in partial fulfillment of requirements for the award of the course

EGB1201 - JAVA PROGRAMMING

in

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

SAMAYAPURAM – 621 112

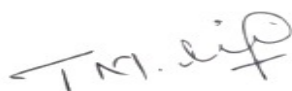
DECEMBER - 2024

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

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on “**COLLEGEMANAGEMENTSYSTEM**” is the bonafide work of **GOWSIC PRABHA K (8115U23EC029)** who carried out the project work during the academic year 2024 - 2025 under my supervision.



SIGNATURE

Dr. T. M. NITHYA, M.E., Ph.D.,

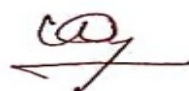
HEAD OF THE DEPARTMENT

ASSOCIATE PROFESSOR

Department of CSE

K.Ramakrishnan College of
Engineering (Autonomous)

Samayapuram-621112.



SIGNATURE

Mr. V. KUMARARAJA, M.E., (Ph.D.),

SUPERVISOR

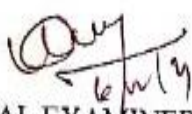
ASSISTANT PROFESSOR

Department of CSE

K.Ramakrishnan College of Engineering
(Autonomous)

Samayapuram-621112.

Submitted for the viva-voce examination held on 6.12.24



INTERNAL EXAMINER



EXTERNAL EXAMINER

DECLARATION

I declare that the project report on “**COLLEGE MANAGEMENT SYSTEM** ” is the result of original work done by us and best of our knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **EGB1201 - JAVA PROGRAMMING**.

K. Gowsic Prabha

Signature

GOWSIC PRABHA K

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution “**K.Ramakrishnan College of Engineering (Autonomous)**”, for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN , B.E.**, for having provided for 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 to our project and offering adequate duration in completing our project.

I would like to thank **Dr. D. SRINIVASAN, B.E, M.E., Ph.D.**, Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. T. M. NITHYA, M.E.,Ph.D.**, Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mr.V.KUMARARAJA, M.E., (Ph.D.)**, Department of **COMPUTER SCIENCE AND ENGINEERING**, for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to 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 progress.

VISION OF THE INSTITUTION

To achieve a prominent position among the top technical institutions.

MISSION OF THE INSTITUTION

- M1: To bestow standard technical education par excellence through state of the art infrastructure, competent faculty and high ethical standards.
- M2: To nurture research and entrepreneurial skills among students in cutting edge technologies.
- M3: To provide education for developing high-quality professionals to transform the society.

VISION OF DEPARTMENT

To create eminent professionals of Computer Science and Engineering by imparting quality education.

MISSION OF DEPARTMENT

M1: To provide technical exposure in the field of Computer Science and Engineering through state of the art infrastructure and ethical standards.

M2: To engage the students in research and development activities in the field of Computer Science and Engineering.

M3: To empower the learners to involve in industrial and multi-disciplinary projects for addressing the societal needs.

PROGRAM EDUCATIONAL OBJECTIVES

Our graduates shall

PEO1: Analyse, design and create innovative products for addressing social needs.

PEO2: Equip themselves for employability, higher studies and research.

PEO3: Nurture the leadership qualities and entrepreneurial skills for their successful career.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Apply the basic and advanced knowledge in developing software, hardware and firmware solutions addressing real life problems.
- **PSO2:** Design, develop, test and implement product-based solutions for their career enhancement.

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 analyze 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.

ABSTRACT

The College Management System is a Java-based application designed to streamline and automate various academic and administrative processes within a college. It allows administrators to manage students, faculty, and courses effectively while maintaining a centralized database. Faculty members can handle tasks such as managing attendance, entering grades, and accessing assigned courses. Students can register for courses, pay fees, and view grades, ensuring transparency and accessibility. The application also includes a Fees Module that processes payments, generates receipts, and provides real-time status updates. The system offers intuitive menus for seamless navigation between modules and role-based access control to maintain data security. Built with Java, it demonstrates modular design, robust data handling, and user-friendly functionality, making it a comprehensive tool for efficient college management. The project showcases the integration of core programming concepts, modular design, and interactive menu-driven functionality in a practical application. Moreover, the project emphasizes scalability and future enhancements, such as integrating cloud storage for data synchronization, automating grading systems, and introducing advanced reporting tools. By reducing manual processes and offering a centralized management platform, the College Management System significantly enhances productivity and transparency, contributing to the institution's overall efficiency. The system serves as a modern solution to traditional management challenges in educational institutions, aligning with contemporary technological trends to deliver a reliable, scalable, and user-centric platform.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The College Management System is a Java-based application that helps colleges manage academic and administrative tasks effectively. Users can access different modules for specific roles: administrators can manage students, faculty, and courses; faculty can track attendance and enter grades; students can register for courses, pay fees, and view grades; and the Fees module handles payment tracking and receipts. With role-based functionalities and intuitive navigation, the system ensures smooth operation and data security. This project highlights modular design, data management, and user-friendly interface development in a practical context.	PO1 -3 PO2 -3 PO3 -3 PO4 -3 PO5 -3 PO6 -3 PO7 -3 PO8 -3 PO9 -3 PO10 -3 PO11-3 PO12 -3	PSO1 -3 PSO2 -3

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

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	
1	INTRODUCTION	
	1.1 Objective	
	1.2 Overview	
	1.3 Java Programming concepts	
2	PROJECT METHODOLOGY	
	2.1 Proposed Work	
	2.2 Block Diagram	
3	MODULE DESCRIPTION	
	3.1 User Registration Module	
	3.2 User Authentication Module	
	3.3 Role-Based Access Control (RBAC) Module	
	3.4 Survey Creation Module	
	3.5 Survey Distribution module	
	3.6 Survey Response Module	
	3.7 Result Analysis Module	
4	CONCLUSION & FUTURE SCOPE	
	4.1 Conclusion	
	4.2 Future Scope	
	APPENDIX A (SOURCE CODE)	
	APPENDIX B (SCREENSHOTS)	
	REFERENCES	

CHAPTER 1

INTRODUCTION

1.1 Objective

The College Management System project is developed to streamline and automate various academic and administrative processes within a college. The main objective is to provide a comprehensive tool that allows administrators, faculty, and students to manage and access key college activities efficiently. The application aims to facilitate tasks such as managing student and faculty records, tracking course enrollments, handling attendance, and processing fee payments. Additionally, it provides real-time updates on students' academic progress and financial status, ensuring seamless communication and transparency. Users can also access detailed summaries and reports to better monitor and manage college-related activities. Additionally, the project seeks to promote transparency and accessibility by offering user-friendly interfaces for all stakeholders. Students can view profiles, register for courses, and pay fees, while faculty can manage academic responsibilities such as attendance and grades.

1.2 Overview

The College Management System is a desktop application built using Java, designed to streamline and automate various academic and administrative tasks within a college. The application allows administrators to manage student and faculty records, courses, and fee payments, while faculty members can track attendance and enter grades. Students can register for courses, view grades, and track their fee payments. The system dynamically updates and displays information, ensuring that users have real-time access to relevant data. With an intuitive graphical user interface (GUI), the program enables seamless navigation

between different modules, providing administrators, faculty, and students with a user-friendly experience. The system also incorporates role-based access control to maintain data security and ensures that the appropriate features are accessible to each user type .

1.3 Java Programming Concepts

The College Management System project utilizes several fundamental Java programming concepts to create a robust and interactive application. The project is based on Object-Oriented Programming (OOP) principles, where the core logic is encapsulated in classes such as Admin, Faculty, Student, and Fees. These classes contain fields and methods responsible for managing student and faculty records, courses, attendance, grades, and fee payments. The College Management System application uses Swing to build the graphical user interface (GUI) with components like JFrame, JPanel, and JButton. These Swing components allow for a flexible, interactive layout, providing an intuitive user experience. Event handling in the application is managed using listeners such as ActionListener and MouseListener to capture and process user inputs. This ensures that the program reacts to user interactions like clicking buttons or entering text. The application employs layout managers like GridLayout, BorderLayout, and FlowLayout to arrange GUI components efficiently. These layouts help ensure the user interface is adaptable to different screen sizes and orientations. The program also uses data types like String, int, and ArrayList to store and process student and faculty information, course details, and fee records. Constants are used to define budget limits, making the code more flexible and easier to update. Exception handling uses try-catch blocks to prevent crashes from invalid input. It ensures non-numeric or incorrectly formatted entries are handled properly.

CHAPTER 2

PROJECT METHODOLOGY

2.1 Proposed Work

The proposed work for the College Management System focuses on developing a comprehensive software solution to streamline and automate key administrative and academic processes within an educational institution. The system will include modules for managing student admissions, profiles, attendance, fee payments, and academic records. It will also cater to faculty needs such as attendance tracking, course assignment, and leave management. Additionally, the system will integrate examination scheduling, result generation, library management, and optional features like hostel management. A robust communication portal will facilitate notifications and updates via email or SMS, ensuring smooth interaction between stakeholders. Built with a secure user authentication system, the solution will provide role-based access for administrators, faculty, students, and parents. Advanced analytics and reporting features will support informed decision-making and enhance operational efficiency. This system will significantly reduce manual work, improve accuracy, and ensure a user-friendly experience for all users.

2.1.1 User Registration and Authentication

This module is a critical component of the College Management System, ensuring secure access and personalized experiences for all users. This module allows students, faculty, administrators, and parents to register with unique credentials such as email IDs or phone numbers. During registration, user roles are assigned to define access permissions. A secure authentication system, incorporating encryption and multi-factor authentication (MFA), safeguards

user data and system integrity. Features include login, logout, password recovery, and session management to enhance usability. Administrators can monitor and manage user accounts, ensuring compliance with institutional policies. This module provides a reliable and secure gateway to the system's features, ensuring a seamless and secure experience for all stakeholders.

2.1.2 Role-Based Access Control (RBAC)

This system in the College Management System is designed to ensure secure and efficient management of user privileges based on their roles within the institution. It categorizes users into specific roles such as administrators, faculty, students, and parents, granting them access only to the functionalities and data relevant to their responsibilities. For instance, administrators can manage all modules, faculty can access academic and attendance records, students can view their profiles and results, and parents can monitor student performance. This structured access control enhances data security, minimizes unauthorized access, and simplifies system maintenance. By aligning permissions with user responsibilities, RBAC ensures operational efficiency and protects sensitive institutional information.

2.1.3 Survey Creation and Management

This module in the College Management System enables administrators and faculty to design, distribute, and analyze surveys effectively. This module facilitates the creation of customizable surveys for various purposes, such as gathering student feedback, assessing course effectiveness, or conducting general opinion polls. Users can design surveys with multiple question formats, including multiple-choice, short answers, and ratings. The system ensures seamless distribution to targeted groups, such as students, faculty, or parents, through the portal or email notifications. Real-time data collection and reporting allow for quick analysis of responses, enabling data-

driven decision-making. This module promotes engagement, enhances feedback mechanisms, and supports continuous improvement across the institution.

2.1.4 Survey Distribution and Response

The **Survey Distribution and Response** module in the College Management System ensures efficient dissemination and collection of survey data. Administrators and faculty can distribute surveys to specific user groups, such as students, faculty, or parents, through the system's portal, email, or SMS notifications. The module provides options for setting deadlines, reminders, and tracking participation rates. Respondents can easily access and complete surveys using user-friendly interfaces on web or mobile platforms. The system records responses securely and in real-time, ensuring data integrity and confidentiality. Automated acknowledgment of submissions and real-time analytics enable quick insights, fostering effective feedback utilization and decision-making.

2.1.5 Survey Result Analysis

The **Survey Result Analysis** module in the College Management System provides comprehensive tools to evaluate survey responses and derive actionable insights. It aggregates data in real-time, presenting it through visualizations such as charts, graphs, and summary reports for easy interpretation. Administrators and faculty can filter results by categories, demographics, or specific questions to focus on relevant trends and feedback. Advanced analytics, including sentiment analysis and statistical summaries, help identify patterns and areas for improvement. This module ensures that survey outcomes are effectively utilized for decision-making, enhancing academic programs, administrative processes, and overall stakeholder satisfaction within the institution.

2.1.6 Security Features

The **Security Features** of the College Management System are designed to safeguard sensitive data and ensure a secure user experience. The system employs robust measures, including encrypted data storage, secure login with multi-factor authentication (MFA), and role-based access control (RBAC) to prevent unauthorized access. It integrates regular data backups and secure data transmission protocols like HTTPS to protect against data loss and interception. Advanced measures such as intrusion detection, activity logging, and session timeouts enhance monitoring and risk mitigation. Compliance with data protection regulations ensures the privacy and integrity of user information, making the system reliable and secure for all stakeholders.

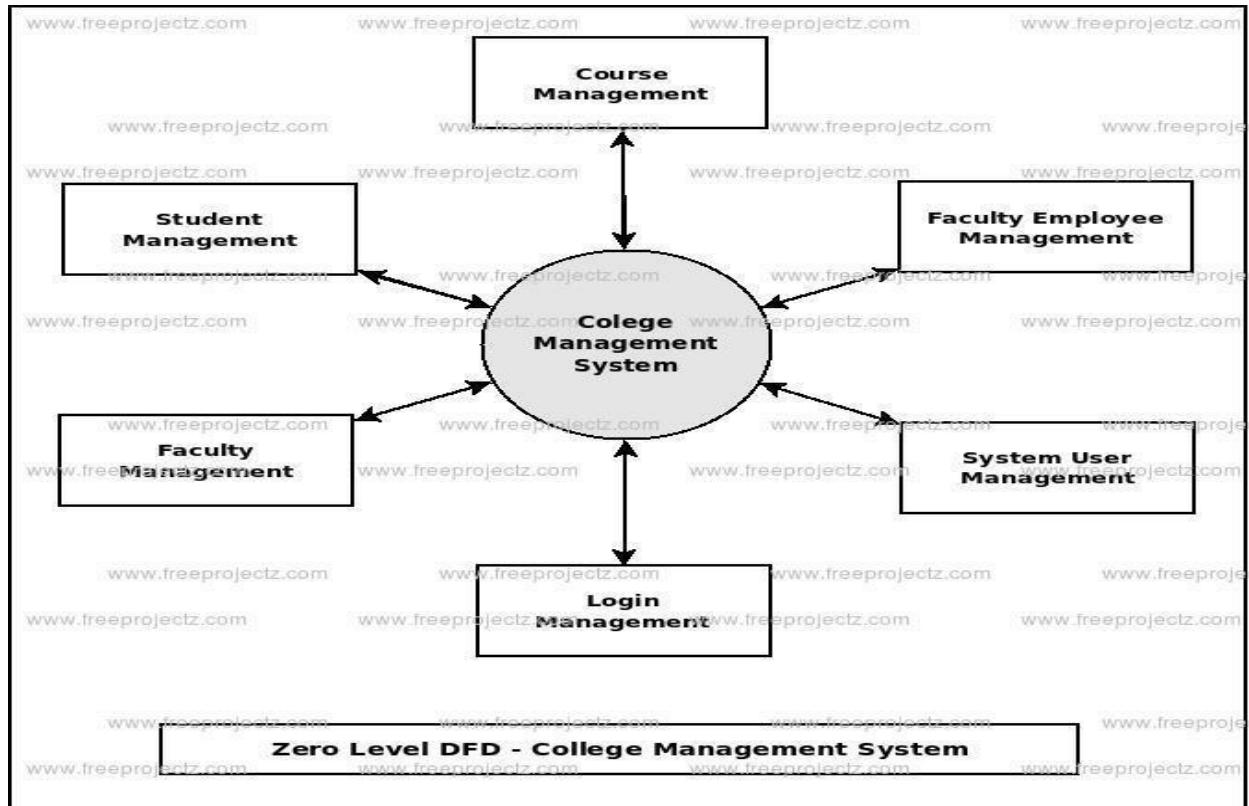
2.1.7 User Interface (UI) Development

The **User Interface (UI) Development** for the College Management System focuses on creating a visually appealing, intuitive, and responsive design to enhance user experience. The UI is tailored to cater to diverse user roles, including administrators, faculty, students, and parents, ensuring easy navigation and access to relevant features. Modern design principles and frameworks such as Bootstrap or Material UI are employed to deliver consistency and accessibility across devices. Features like dashboards, quick-access menus, and interactive elements simplify task management. Special attention is given to usability, with clear layouts, contextual help, and error messages to guide users effectively. This user-centric design ensures seamless interaction and boosts system adoption rates.

2.1.8 Future Enhancements

The Future Enhancements for the College Management System aim to expand its capabilities, improve user experience, and incorporate emerging technologies. Future upgrades may include integration with Artificial Intelligence (AI) for predictive analytics, such as student performance forecasting and personalized learning recommendations. Adding mobile app support for both Android and iOS will provide users with more flexibility and on-the-go access. Blockchain technology could be incorporated for secure, transparent academic record management. Furthermore, AI-driven chatbots for automated customer support and advanced communication features will enhance user engagement. Integration with external platforms like learning management systems (LMS) or career services could further streamline administrative tasks, making the system more comprehensive and adaptable to evolving educational needs.

2.2 Block Diagram



CHAPTER 3

MODULE DESCRIPTION

1. Course Management

Description:

This module handles the management of all courses offered by the college. It includes functionalities for course creation, modification, and deletion. Faculty members can assign courses to students, and students can register for courses through the system. It also tracks course schedules, room assignments, and course prerequisites, ensuring a smooth and organized course delivery system.

2. Student Management

Description:

This module deals with the management of student data, including personal details, academic records, course enrollments, attendance, and performance tracking. It allows students to update their profiles, view their grades, and check the status of their academic progress. Administrators and faculty can access detailed student reports and generate academic transcripts and certificates.

3. Faculty Management

Description:

This module manages faculty-related information, including faculty profiles, department assignments, teaching schedules, and performance evaluations. Faculty members can update their details, manage their class schedules, and access their teaching assignments. The module allows administrators to monitor faculty performance and generate reports.

4. Faculty Employee Management

Description:

This module focuses on managing faculty employment-related information, such as contracts, salary details, benefits, and employment history. It tracks faculty tenure, promotions, and appraisals.

5. Login Management

Description:

The Login Management module handles user authentication and access control for all users (students, faculty, administrators). It ensures secure login using credentials such as username and password. This module supports role-based access, where different users are granted varying levels of system access depending on their role (admin, student, faculty).

6. System User Management

Description:

This module is responsible for managing the creation, modification, and deletion of user accounts for the entire system. It includes defining roles (such as admin, faculty, student), setting permissions, and ensuring that users have access only to the relevant sections of the system. It tracks user activity and manages user privileges to maintain system security and integrity.

7. College Management System (Core)

Description:

The central module that integrates all other modules. It acts as the hub of the College Management System, ensuring seamless interaction between various components like course management, student management, faculty management, and login management. This module provides a user-friendly interface for interacting with the system and ensures data consistency across all sections.

CHAPTER 4

CONCLUSION AND FUTURE SCOPE

4.1 Conclusion:

The College Management System is an essential tool designed to streamline and automate the various administrative and academic processes within a college or educational institution. By integrating modules such as user registration, role-based access control, student management, fee management, attendance tracking, examination management, and more, the system facilitates smooth and efficient operations across departments. This system enhances collaboration between students, faculty, administrators, and parents, offering a centralized platform for real-time communication, information sharing, and decision-making. The automation of routine tasks reduces administrative workload, minimizes errors, and ensures consistency in handling data, allowing staff to focus on more strategic activities. Moreover, the system's secure login, data encryption, and role-based access ensure that sensitive information is protected and accessible only to authorized users. The ability to generate reports, track performance, and monitor progress empowers administrators to make data-driven decisions that contribute to the overall growth and success of the institution.

4.2 Future Scope

The future scope of the College Management System lies in its ability to continuously evolve and integrate advanced technologies to enhance the educational experience. With the rise of mobile apps, AI, and machine learning, future systems will offer personalized learning experiences, predictive analytics, and more efficient administrative processes. Integrating blockchain technology for secure academic record management, along with cloud-based solutions, will ensure scalability, security, and real-time access. Additionally, IoT integration can create smart campuses, while enhanced analytics and reporting tools will empower administrators with data-driven insights. The system may also incorporate external platform integrations and advanced communication tools to foster better collaboration among students, faculty, and staff. As these technologies develop, the College Management System will become a more robust, user-centric solution, contributing to improved institutional efficiency and student success.

APPENDIX A

```
package collegemanagementsystem;

Import java.util.Scanner;

Import java.util.ArrayList;

//Main Class

Public class CollegeManagementSystem

{ Public static void main(String[] args) { Scanner scanner = new Scanner(System.in);
Admin admin = new Admin();

Faculty faculty = new Faculty(); Student student = new Student();


System.out.println("Welcome to the College Management System");
System.out.println("1. Admin\n2. Faculty\n3. Student\n4. Exit");

Int choice = scanner.nextInt();


Switch (choice)

{ Case 1: Admin.adminMenu(); Break;

Case 2:

Faculty.facultyMenu(); Break;


Case 3:

Student.studentMenu(); Break;

Case 4:

System.out.println("Exiting the system. Goodbye!"); Break;

Default:

System.out.println("Invalid choice!");

}
```

```
Scanner.close();
```

```
}
```

```
}
```

```
//Admin Class Class Admin {
```

```
ArrayList<Student> students = new ArrayList<>(); ArrayList<Faculty> faculty = new  
ArrayList<>(); ArrayList<String> courses = new ArrayList<>(); Fees fees = new  
Fees();
```

```
Void adminMenu() {
```

```
Scanner scanner = new Scanner(System.in); Int choice;
```

```
Do {
```

```
System.out.println("\nAdmin Menu:");
```

```
System.out.println("1. Manage Students\n2. Manage Faculty\n3. Manage Courses\n4.  
View Fees\n5. Exit");
```

```
Choice = scanner.nextInt(); Scanner.nextLine();
```

```
Switch (choice)
```

```
{ Case 1: manageStudents(scanner); break;
```

```
case 2:
```

```
manageFaculty(scanner); break;
```

```
case 3:
```

```
manageCourses(scanner); break;
```

```
case 4:
```

```
viewFees(scanner); break;
```

```
case 5:
```

```
System.out.println("Exiting Admin Menu."); Break;
```


Default:

```
System.out.println("Invalid choice!");
```

```
}
```

```
} while (choice != 5);
```

```
}
```

Void manageStudents(Scanner scanner)

```
{ System.out.println("Enter student name to add:"); String studentName =  
scanner.nextLine(); Students.add(new Student(studentName));  
System.out.println("Student added: " + studentName);
```

```
}
```

Void manageFaculty(Scanner scanner)

```
{ System.out.println("Enter faculty name to add:"); String facultyName =  
scanner.nextLine(); Faculty.add(new Faculty(facultyName));  
System.out.println("Faculty added: " + facultyName);
```

```
}
```

Void manageCourses(Scanner scanner)

```
{ System.out.println("Enter course name to add:");String course = scanner.nextLine();  
Courses.add(course); System.out.println("Course added: " + course);
```

```
}
```

Void viewFees(Scanner scanner)

```
{ Fees.viewFeeDetails();
```

```
}  
}
```

```
//Faculty Class Class Faculty
```

```
{String name;
```

```
Faculty() { } Faculty(String name) {
```

```
This.name = name;
```

```
}
```

```
Void facultyMenu()
```

```
{ System.out.println("\nFaculty Menu:");
```

```
System.out.println("1. View Courses\n2. Manage Attendance\n3. Enter Grades\n4. Exit");
```

```
//Implement functionalities here as needed.
```

```
}
```

```
}
```

```
//Student Class Class Student
```

```
{String name;
```

```
Fees fees = new Fees();
```

```
Student() { }
```

```
Student(String name)
```

```
{ This.name = name;
```

```
}
```

```
Void studentMenu() {
```

```
Scanner scanner = new Scanner(System.in); Int choice;
```

```
Do {
```

```
System.out.println("\nStudent Menu:");
```

```
System.out.println("1. View Profile\n2. Register for Courses\n3. Pay  
Fees\n4. View Grades\n5. Exit");
```

```
Choice = scanner.nextInt(); Scanner.nextLine();
```

```
Switch (choice)
```

```
{ Case 1: viewProfile(); break;
```

```
case 2:
```

```
registerCourses(); break;
```

```
case 3:
```

```
payFees();
```

```
break; case 4:
```

```
viewGrades(); break;
```

case 5:

```
System.out.println("Exiting Student Menu."); Break;
```

Default:

```
System.out.println("Invalid choice!");
```

```
}
```

```
} while (choice != 5);
```

```
}
```

```
Void viewProfile() {
```

```
System.out.println("Student Profile: " + name);
```

```
}
```

```
Void registerCourses()
```

```
{ System.out.println("Enter course to register:");
```

```
//Register course logic
```

```
}
```

```
Void payFees()
```

```
{ Fees.payFees ();
```

16

}

Void viewGrades() {

//Show grades functionality

}

}

//Fees Class Class Fees {

Double amountDue = 5000; // Example amount, this can be dynamic

Void payFees() {

System.out.println("Paying fees. Amount due: \$" + amountDue);

//Payment logic, here just a mock System.out.println("Payment Successful!");
amountDue = 0;

}

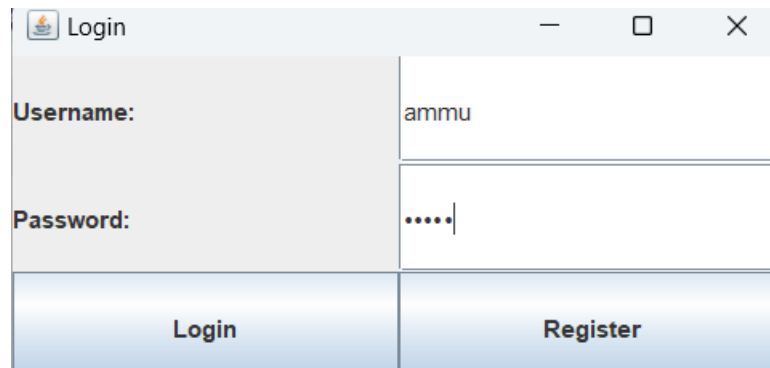
Void viewFeeDetails() {

System.out.println("Total Fees Due: \$" + amountDue);

}

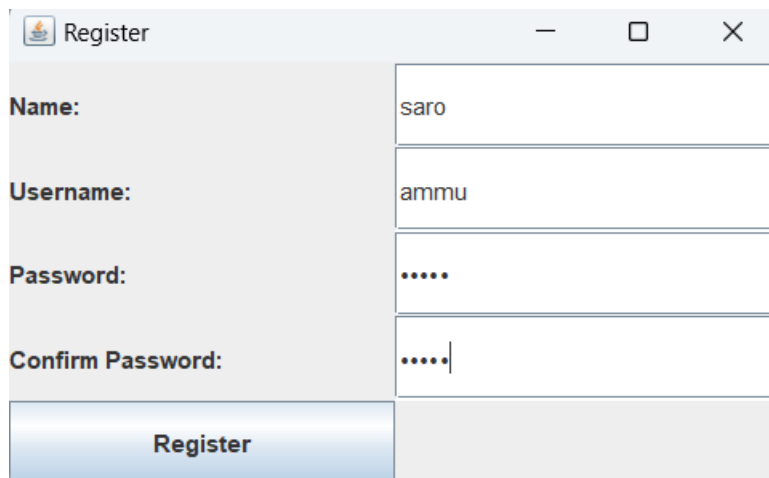
}

APPENDIX B



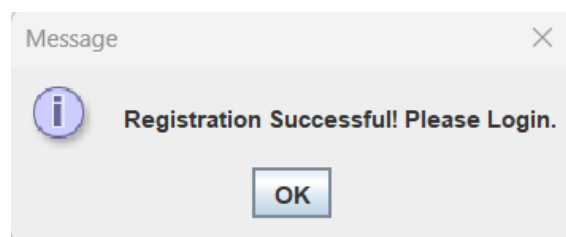
A screenshot of a Java Swing window titled "Login". The window has a standard title bar with minimize, maximize, and close buttons. The main area is divided into two sections. The first section is labeled "Username:" and contains a text field with the value "ammu". The second section is labeled "Password:" and contains a password field with five dots. Below these fields are two buttons: "Login" and "Register".

Username:	ammu
Password:
Login	Register




A screenshot of a Java Swing window titled "Register". The window has a standard title bar with minimize, maximize, and close buttons. The main area is divided into four sections. The first section is labeled "Name:" and contains a text field with the value "saro". The second section is labeled "Username:" and contains a text field with the value "ammu". The third section is labeled "Password:" and contains a password field with five dots. The fourth section is labeled "Confirm Password:" and contains a password field with five dots. Below these fields is a single button labeled "Register".

Name:	saro
Username:	ammu
Password:
Confirm Password:
Register	



A screenshot of a Java Swing message dialog box titled "Message". The dialog has a standard title bar with a close button. The main area contains an information icon (a lowercase 'i' inside a circle) followed by the text "Registration Successful! Please Login." Below the text is an "OK" button.

Message

 Registration Successful! Please Login.

OK

REFERENCES:

- [1] **Sommerville, I. (2011).** Software Engineering (9th ed.). Addison-Wesley.
- [2] **Pressman, R. S. (2014).** Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill.
- [3] **Bansal, A., & Sharma, R. (2017).** "Development of College Management System using PHP and MySQL." International Journal of Computer Applications, 166(8), 30-35.
- [4] **Bhatia, S., & Mehta, A. (2019).** "Design and Implementation of an Automated College Management System." International Journal of Computer Science and Technology, 14(2), 101-107.
- [5] **Kumar, S., & Singh, R. (2016).** "A Survey on College Management Systems: Issues and Challenges." International Journal of Computer Science & Information Technology, 8(4), 213-218.
- [6] **"Student Management System: A Review."** International Journal of Advanced Research in Computer Science and Software Engineering, 8(5), 53-56.
- [7] **"Building a College Management System using Cloud Computing."** International Journal of Advanced Computer Science and Applications, 9(4), 112-120.