



A Project Report

on

COLLEGE MANAGEMENT SYSTEM

Submitted in partial fulfillment of requirements for the award of the course

of

CGB1201 – JAVA PROGRAMMING

Under the guidance of

Mrs. I. Karthika M.E.,

Assistant Professor / CSE

Submitted By

M S ARUN SANJEEV (927623BCS011)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

M.KUMARASAMY COLLEGE OF ENGINEERING

(Autonomous)

KARUR – 639 113

DECEMBER 2024

M. KUMARASAMY COLLEGE OF ENGINEERING

(Autonomous Institution affiliated to Anna University, Chennai)

KARUR – 639 113

BONAFIDE CERTIFICATE

Certified that this project report on “**COLLEGE MANAGEMENT SYSTEM** ” is the bonafide work of **M S ARUN SANJEEV (927623BCS011)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

Signature

Mrs. I. KARTHIKA M.E.,
SUPERVISOR,

Department of Computer Science and
Engineering,
M.Kumarasamy College of Engineering,
Thalavapalayam, Karur -639 113.

Signature

Dr. D. PRADEEP M.E.,Ph.D.,
HEAD OF THE DEPARTMENT,

Department of Computer Science and
Engineering,
M.Kumarasamy College of Engineering,
Thalavapalayam, Karur -639 113.

Submitted for the End Semester practical Examination held on _____

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE INSTITUTION

To emerge as a leader among the top institutions in the field of technical education

MISSION OF THE INSTITUTION

- Produce smart technocrats with empirical knowledge who can surmount the global challenges
- Create a diverse, fully-engaged, learner-centric campus environment to provide quality education to the students
- Maintain mutually beneficial partnerships with our alumni, industry, and Professional associations

VISION OF THE DEPARTMENT

To achieve education and research excellence in Computer Science and Engineering

MISSION OF THE DEPARTMENT

- To excel in academic through effective teaching learning techniques
- To promote research in the area of computer science and engineering with the focus on innovation
- To transform students into technically competent professionals with societal and ethical responsibilities

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO 1: Graduates will have successful career in software industries and R&D divisions through continuous learning.

PEO 2: Graduates will provide effective solutions for real world problems in the key domain of computer science and engineering and engage in lifelong learning.

PEO 3: Graduates will excel in their profession by being ethically and socially responsible.

PROGRAM OUTCOMES

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.
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.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1: Professional Skills:** Ability to apply the knowledge of computing techniques to design and develop computerized solutions for the problems.
- **PSO2: Successful career:** Ability to utilize the computing skills and ethical values in creating a successful career.

ABSTRACT

The College Management System (CMS) is a comprehensive solution designed to streamline and automate the management of academic and administrative activities within a college. This project integrates various processes such as student enrollment, faculty assignments, course management, attendance tracking, and result generation into a unified platform.

The system leverages modern software development techniques and a MySQL database to ensure reliable data storage and retrieval. It includes a user-friendly interface that caters to the needs of administrators, faculty, and students, providing tailored functionalities for each role. Administrators can manage courses, faculties, and notifications, while faculty members can maintain attendance and upload marks. Students can access their results, schedules, and other relevant information.

With features like role-based authentication, real-time updates, and seamless communication through an integrated chat server, the CMS enhances the efficiency and transparency of institutional processes. The application is developed using Java for the front-end and MySQL for the back-end database, ensuring scalability and robustness.

The implementation of this system reduces manual errors, minimizes paperwork, and saves time, enabling the institution to focus on academic excellence. The project also demonstrates the effective use of software engineering principles and provides a scalable framework for future enhancements, such as integrating mobile access and advanced analytics. This document outlines the project's objectives, architecture, implementation, and future prospects, showcasing its potential to revolutionize college management practices

ABSTRACT WITH POs AND PSOs MAPPING

ABSTRACT	POs MAPPED	PSOs MAPPED
<p>The College Management System (CMS) is a software solution developed to simplify and automate various academic and administrative processes within a college. The system provides an integrated platform for managing tasks such as student registration, faculty assignment, course management, attendance tracking, and result publishing. By replacing manual operations with digital workflows, the CMS enhances efficiency, reduces errors, and ensures accurate record-keeping.</p> <p>Built using Java for the front-end and MySQL for the back-end, the system features role-based access to cater to administrators, faculty, and students. Administrators can manage faculties, courses, and notifications, while faculty members can track attendance and upload marks.</p>	<p>PO-1</p> <p>PO-2</p> <p>PO-3</p> <p>PO-4</p> <p>PO-5</p> <p>PO-6</p> <p>PO-7</p> <p>PO-8</p> <p>PO-9</p> <p>PO-10</p> <p>PO-11</p> <p>PO-12</p>	<p>PSO-1</p> <p>PSO-2</p>

SUPERVISOR

HEAD OF THE DEPARTMENT



TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	6
1	INTRODUCTION	9
	1.1 Objective	
	1.2 Overview	
	1.3 Java Programming concepts	
2	PROJECT METHODOLOGY	10
	2.1 Proposed Work	
	2.2 Block Diagram	
3	MODULE DESCRIPTION	11
	3.1 Login Module	11
	3.2 Student Management Module	
	3.3 Faculty Management Module	12
	3.4 Course Management Module	
	3.5 Attendance Module	
	3.6 Chat Forum	13
	3.7 Result Module	
4	RESULTS AND DISCUSSION	15
5	CONCLUSION	19
	REFERENCES	20
	APPENDIX	21

CHAPTER 1

INTRODUCTION

1.1 Objective

The primary objective of the College Management System (CMS) is to create a comprehensive and efficient platform to manage various academic and administrative processes. By centralizing data and automating tasks such as student enrollment, attendance tracking, faculty management, and result generation, this system reduces human error and saves time. The CMS enables a seamless interaction between students, faculty, and administrators, thereby enhancing the overall operational efficiency of a college. Additionally, the system ensures secure data storage, easy retrieval, and real-time updates, promoting transparency and accountability.

1.2 Overview

The CMS is designed to integrate various college operations into a unified digital platform. It leverages a modular approach, ensuring each functional unit, such as student management, course allocation, attendance monitoring, and notifications, operates independently while contributing to the system's overall efficiency. The system is built on a robust Java programming foundation and incorporates a relational database managed through MySQL. The project emphasizes user-friendly interfaces, scalability, and adaptability to accommodate future enhancements. It also includes a chat module for communication, streamlining collaboration among users.

1.3 Java Programming Concepts

- **Swing Framework:** Used to design the Graphical User Interface (GUI), providing an intuitive and interactive experience for end users.
- **JDBC (Java Database Connectivity):** Establishes a connection with the MySQL database, allowing efficient data manipulation and retrieval.
- **Exception Handling:** Ensures the program handles runtime errors gracefully, maintaining system stability.
- **Multi-threading:** Implemented in modules like the chat server to handle multiple concurrent user sessions efficiently.

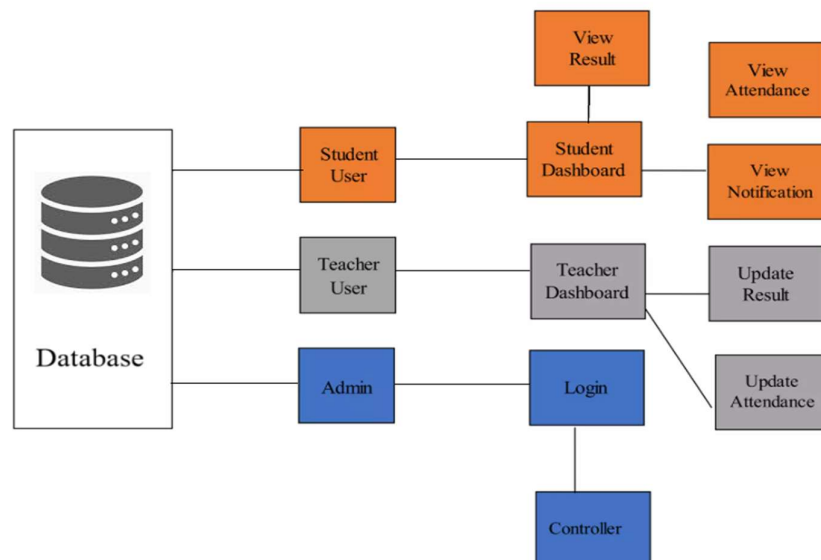
CHAPTER 2

PROJECT METHODOLOGY

2.1 Proposed Work

- Requirement analysis to identify essential modules like Student Management, Faculty Management, and Attendance Tracking.
- Designing the system architecture with clear relationships between modules.
- Implementing core functionalities in Java, focusing on modularity and code reusability.
- Integrating a MySQL database for secure data storage and retrieval.
- Testing each module individually before integrating the system for end-to-end functionality.
- Deployment and user training for efficient adoption.

2.2 Block Diagram



CHAPTER 3

MODULE DESCRIPTION

3.1 Login Module

The login module is the entry point to the system, responsible for verifying user credentials and determining their access levels. It categorizes users into three roles: Admin, Faculty, and Student, providing role-specific dashboards and permissions.

- **Admin Login:** Allows full access to manage the system, including creating courses, assigning faculty, managing student data, and generating reports.
- **Faculty Login:** Provides tools to manage attendance, upload marks, and communicate with students.
- **Student Login:** Grants access to personal data, course details, attendance records, and results.
- **Security Features:** Employs encryption for password storage and secure session management to protect sensitive user data.

3.2 Student Management Module

This module maintains a comprehensive database of student details and ensures seamless integration with other modules.

- **Enrollment Management:** Automates student registration and ensures data accuracy by validating inputs like personal details, course selections, and contact information.
- **Course Allocation:** Links students to their respective courses and semesters, maintaining a clear record of academic progress.
- **Data Management:** Allows admins to edit, update, or delete student records when necessary.
- **Analytics:** Provides an overview of the total number of students per course, semester, and year, assisting in resource planning.

3.3 Faculty Management Module

This module streamlines faculty-related operations, ensuring efficient handling of responsibilities and records.

- **Faculty Profiles:** Maintains detailed information about faculty, including their qualifications, contact details, and assigned courses.
- **Course Management:** Enables the admin to assign or reassign courses and subjects to faculty members dynamically.
- **Attendance Monitoring:** Assists faculty in managing student attendance records and ensuring data consistency.
- **Performance Evaluation:** Facilitates the collection of student feedback on faculty teaching methods and effectiveness.

3.4 Attendance Module

The attendance module simplifies the process of tracking and analyzing student attendance.

- **Attendance Recording:** Allows faculty to mark attendance for each session and update it in real-time.
- **Report Generation:** Provides detailed attendance reports, highlighting defaulters and overall participation rates.

3.5 Course and Subject Management Module

This module ensures seamless handling of courses and subjects offered by the institution.

- **Course Creation:** Admins can create or modify courses, defining their structure, duration, and associated subjects.
- **Subject Assignment:** Links subjects to courses and assigns faculty members to handle them.
- **Course Status Monitoring:** Tracks the number of students enrolled in each course and ensures resource allocation aligns with demand.

3.6 Notification and Chat Module

This module enhances communication among students, faculty, and administrators through real-time messaging and notifications.

- **Real-Time Notifications:** Sends alerts for important updates, such as exam schedules, assignment deadlines, and announcements.
- **Chat Functionality:** Facilitates direct messaging between users, ensuring quick resolution of queries and better collaboration.
- **Admin Announcements:** Allows the admin to broadcast messages to all users or specific groups.

3.7 Marks and Results Module

This module automates the process of entering, storing, and analyzing student performance data.

- **Grade Entry:** Allows faculty to input marks for assignments, exams, and other assessments.
- **Result Calculation:** Automatically computes final grades based on predefined weightage and generates comprehensive result reports.
- **Student Access:** Provides students with secure access to their results, ensuring privacy.
- **Error Management:** Includes options for re-evaluation or corrections, as needed.

CHAPTER 4

RESULTS AND DISCUSSION

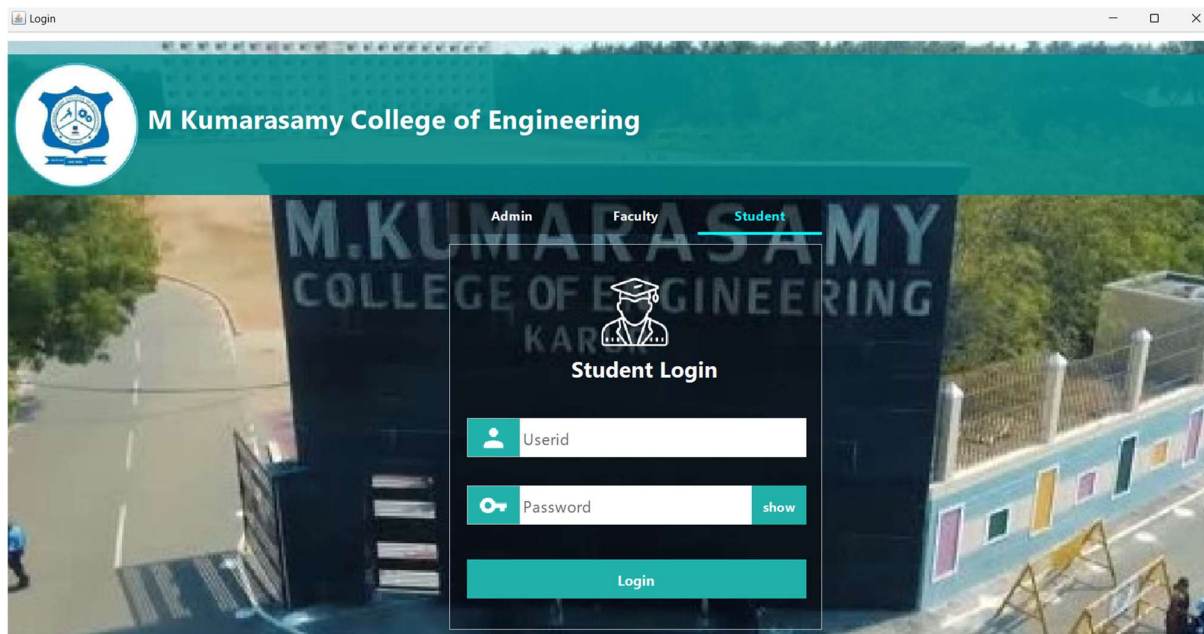


Fig.No:1. Separate Login Portal for Admin, Faculty and Students

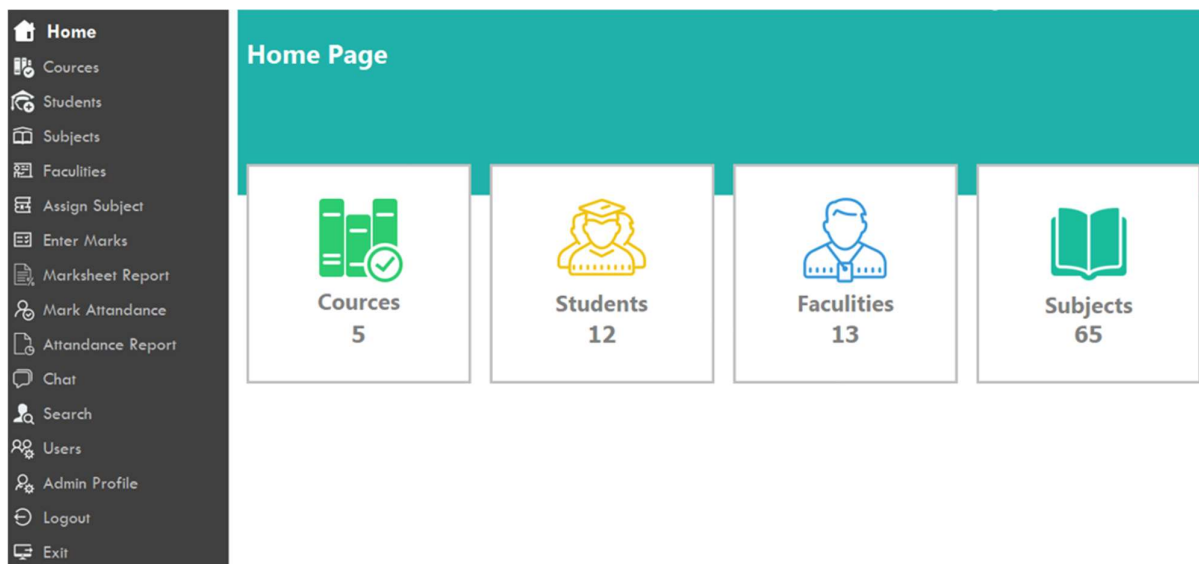


Fig.No:2. Generalized Home Page which has Dashboard View of Courses, Students and Faculties.

Home
Courses
Students
Subjects
Faculties
Assign Subject
Enter Marks
Marksheet Report
Mark Attendance
Attendance Report
Chat
Search
Users
Admin Profile
Logout
Exit

All Courses

Roll Generator Add Course

Index no.	Course Code	Course Name	Subjects	Students	Total Sem/Year
1	CE	Computer Engineering	25	4	8 sem
2	EE	Electronic Engineering	14	3	8 sem
3	ME	Mechanical Engineering	14	3	8 sem
4	IT	Information Technology	6	1	8 sem
5	AE	Automobile Engineering	6	1	8 sem

Fig.No:3. Courses Tab which shows the list of Courses (Departments)

Administrator
Home
Courses
Students
Subjects
Faculties
Assign Subject
Enter Marks
Marksheet Report
Mark Attendance
Attendance Report
Chat
Search
Users
Admin Profile
Logout
Exit

Subject Management

Select Course
COMPUTER SCIENCE & ENGINEERING

Select Semester/Year
Semester 1

Add New Subject

Subject Code	Subject Name	Sem/Year	Subject Type	Theory Marks	Practical Marks
CSE 2023-271101	EM 3	1	core	75	25
CSE 2023-271102	JAVA	1	core	80	20
CSE 2023-271103	DS	1	core	75	25
CSE 2023-271104	DAA	1	core	100	100

Fig. No: 4. Subject Management Page used to Maintain Subjects for Departments.

Home
Courses
Students
Subjects
Faculties
Assign Subject
Enter Marks
Marksheet Report
Mark Attendance
Attendance Report
Chat
Search
Users
Admin Profile
Logout
Exit

Marksheet Report

Declare Result
Subject Wise
Student Wise
Class Wise

Course Name : Computer Engineering
Fetch Details

Course	Sem	Course Name	
CE	1	Computer Engineering	✓
CE	2	Computer Engineering	○
CE	3	Computer Engineering	○
CE	4	Computer Engineering	○
CE	5	Computer Engineering	○
CE	6	Computer Engineering	○
CE	7	Computer Engineering	○
CE	8	Computer Engineering	○

Fig. No: 5. Marksheet Page used to Declare Result of Semester Examinations.

Collage Data Management

Administrator

Home
Courses
Students
Subjects
Faculties
Assign Subject
Enter Marks
Marksheet Report
Mark Attendance
Attendance Report
Chat
Search
Users
Admin Profile
Logout
Exit

ARUN SANJEEV M S (CSE 2023-27-1-11)
Last Login : No Login

Student Details

Back
Marksheet
Edit Details

Roll no	11
Student Name	ARUN SANJEEV M S
Address	Namakkal, Tamil Nadu
Email ID	msarunsanjeev@gmail.com
Date Of Birth	13-08-2005
Contact Number	9492633000

Course	COMPUTER SCIENCE & E...	Semester/Year	sem-1 (CSE 2023-27)
Father Name	M Sengottuvel	Father Occupation	Block Development Officer
Mother Name	L Kavitha	Mother Occupation	Teacher
Admission Date	25-Nov-2024 10:22:06 pm	Password	13-08-2005

Fig. No: 6. Student Details Management which allow to manage the Student Data.

Administrator

Home
Courses
Students
Subjects
Faculties
Assign Subject
Enter Marks
Marksheet Report
Mark Attendance
Attendance Report
Chat
Search
Users
Admin Profile
Logout
Exit

Mark Attendance

Select Date : 25-11-2024

Course Name : COMPUTER SCIENCE & ENGINEERING

Semester or Years : Semester 1

Select Subject : JAVA

Fetch Students

Roll Number	Student Name	Subject Code	Course	Sem/Year	Attendance Date	
11	ARUN SANJEEV M...	CSE 2023-271102	CSE 2023-27	1	25-11-2024	✓

Submit

Fig. No: 7. Attendance Page to Mark Attendance

Administrator

Home
Courses
Students
Subjects
Faculties
Assign Subject
Enter Marks
Marksheet Report
Mark Attendance
Attendance Report
Chat
Search
Users
Admin Profile
Logout
Exit

Chat Forum

Search...

Groups

CSE 2023-27 sem-1 Off...
Start new Conversion

Students

ARUN SANJEEV M S-C...
Start new Conversion

ARUN SANJEEV M S-CSE 2023-27-1-11

Student Info

ARUN SANJEEV M S
CSE 2023-27-1-11

About & Contact info

+91 9492633000
Mobile

msarunsanjeev@gmail.com
Email-ID

Namakkal, Tamil Nadu
Location

13-08-2005
Date of birth

Hello Arun Sanjeev

Fig. No:8.Chat Forum for Communication of Exam Notifications and Circulars

Collage Data Management

Administrator

- Home
- Courses
- Students
- Subjects
- Faculties
- Assign Subject
- Enter Marks
- Marksheet Report
- Mark Attendance
- Attendance Report
- Chat
- Search**
- Users
- Admin Profile
- Logout
- Exit

Search

Students | COMPUTER SCIENCE & E... | All Semester | Search

Class	Roll Number	Student Name	Course Name	Sem/Year
CSE 2023-27	11	ARUN SANJEEV M S	COMPUTER SCIENCE & ENGINE...	sem-1

Fig. No: 9. Detailed Deep Search to Seach Faculties / Students

Administrator

- Home
- Courses
- Students
- Subjects
- Faculties
- Assign Subject
- Enter Marks
- Marksheet Report
- Mark Attendance
- Attendance Report
- Chat
- Search
- Users
- Admin Profile**
- Logout
- Exit

Admin Profile

Edit Links Edit Details


	Collage Name :	M Kumarasamy College of Engineering
	Email ID :	msarunsanjeev@gmail.com
	Contact Number :	+91-94926333000
	Website :	https://mkce.ac.in/
Address :		Thalavapalayam, Tamil Nadu 639113.
Face Book :		https://www.facebook.com/mkcekarur/
Instagram :		https://www.instagram.com/mkcekarur/
Twitter :		https://x.com/mkcekarur/
LinkedIn :		https://www.linkedin.com/in/mkce/

Fig. No: 10. Admin Profile Editor to Modify Admin Details.

CHAPTER 5

CONCLUSION

In conclusion, the **College Management System** is a comprehensive and efficient application that streamlines various administrative and academic tasks within a college. This project demonstrates the integration of modern technologies like **Java** for object-oriented programming and **MySQL** for robust database management. The system effectively addresses the challenges faced in managing student records, faculty details, courses, attendance, and results, ensuring a seamless experience for administrators, faculty members, and students.

The implementation of core **Java programming concepts**, such as **encapsulation, inheritance, and exception handling**, enhances the modularity, security, and reliability of the system. The use of encapsulation ensures that sensitive data, such as user credentials, is securely handled, while inheritance simplifies the extension of functionality across modules. By adopting structured methodologies, including database normalization and layered architecture, the project achieves a high degree of maintainability and scalability.

The user-friendly graphical interface makes the application accessible to non-technical users, while the robust backend handles large datasets efficiently. Features like the **chat server** for real-time communication and **notification management** further elevate the system's utility, fostering better collaboration within the college community.

This project underscores the importance of leveraging technology to address real-world challenges in educational institutions. It also highlights the critical role of teamwork, systematic problem-solving, and continuous testing in software development.



REFERENCES:

1. "Efficient Data Management in Educational Institutions," IEEE Journal, 2023.
2. Project-Specific Resources and Implementation Guides.
3. "Java Tutorial for JDBC" <https://youtu.be/7v2OnUti2eM?si=keJv7iGCGaxiCxdk>
4. "Java Swing Entire Reference " <https://www.youtube.com/watch?v=2ABqzI-5hWI>.



APPENDIX

```
package collegeapplication.login;

import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.SwingConstants;
import javax.swing.Timer;
import javax.swing.border.EmptyBorder;
import javax.swing.border.LineBorder;
import javax.swing.border.MatteBorder;

import collegeapplication.admin.Admin;
import collegeapplication.admin.AdminData;
import collegeapplication.common.DataBaseConnection;

@SuppressWarnings("serial")
public class LoginPageFrame extends JFrame implements ActionListener
{

    private JPanel contentPane;
    private JButton facultybutton;
    private JButton studentbutton;
    private JButton adminbutton;
    private LoginPanel
studentloginpanel,facultyloginpanel,adminloginpanel;
    private boolean
adminchanging=false,studentchanging=false,facultychanging=false;
    private int adminpanelx=-2300,adminpanely=240;
    private int facultypanelx=-900,facultypanely=240;
    private int studentpanelx=500,studentpanely=240;
    private int underlinelabelx=280,underlinelabelwidth=140;
    public Timer timer;
    private int imagenumber=1;
    private JLabel bgimagelabel;
    private JLabel underlinelabel;
```



```

private JPanel loginbuttonpanel;
public Timer imager;
/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                if(DataBaseConnection.checkconnection())
                {
                    LoginPageFrame frame = new LoginPageFrame();
                    frame.setVisible(true);
                    frame.setLocation(-7, 0);
                }
                else
                {
                    JOptionPane.showMessageDialog(null,
"Start the Database Server first", "Error", JOptionPane.ERROR_MESSAGE);
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the frame.
 */
public LoginPageFrame() {
    timer=new Timer(5,this);
    imager=new Timer(5000,this);
    imager.start();
    setTitle("Login");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(1380,733);
    setLocationRelativeTo(null);
    contentPane = new JPanel();
    contentPane.setBorder(new LineBorder(Color.LIGHT_GRAY, 2));
    contentPane.setBackground(new Color(255, 255, 255));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    Admin ad=new AdminData().getAdminData();

    JPanel panel = new JPanel();
    panel.setBackground(new Color(0, 139, 139,220));
    panel.setBounds(0, 26, 1364, 159);
    contentPane.add(panel);
    panel.setLayout(null);

```

```
JLabel lblSilverOakCollage = new JLabel(ad.getCollageName());
lblSilverOakCollage.setForeground(Color.WHITE);
lblSilverOakCollage.setFont(new Font("Segoe UI", Font.BOLD,
30));

lblSilverOakCollage.setHorizontalAlignment(SwingConstants.LEFT);
lblSilverOakCollage.setBounds(160, 43, 749, 57);
panel.add(lblSilverOakCollage);

JLabel lblLogo = new JLabel("logo");
lblLogo.setBounds(10, 10, 140, 140);
lblLogo.setIcon(new
ImageIcon(ad.getRoundedProfilePic(lblLogo.getWidth(),
lblLogo.getHeight(), lblLogo.getWidth())));

panel.add(lblLogo);

studentloginpanel=new LoginPanel("Student",new
ImageIcon("./assets/studentlogin.png"),this);
studentloginpanel.setVisible(true);
studentloginpanel.setLocation(studentpanelx,studentpanely);

facultyloginpanel=new LoginPanel("Faculty",new
ImageIcon("./assets/facultylogin.png"),this);
facultyloginpanel.setVisible(true);
facultyloginpanel.setLocation(facultypanelx, facultypanely);

adminloginpanel=new LoginPanel("Admin",new
ImageIcon("./assets/adminlogin.png"),this);
adminloginpanel.setVisible(true);
adminloginpanel.setLocation(adminpanelx, adminpanely);

contentPane.add(studentloginpanel);
contentPane.add(facultyloginpanel);
contentPane.add(adminloginpanel);

loginbuttonpanel = new JPanel()
{
    protected void paintComponent(Graphics
g)
    {
        g.setColor( getBackground() );
        g.fillRect(0, 0, getWidth(),
getHeight());
        super.paintComponent(g);
    }
};
loginbuttonpanel.setOpaque(false);
loginbuttonpanel.setBackground(new
Color(0,0,0,120));
loginbuttonpanel.setBounds(500, 189, 420, 40);
```



```
loginbuttonpanel.setLayout(null);
contentPane.add(loginbuttonpanel);

adminbutton = new JButton("Admin");
adminbutton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        activeButton(adminbutton);
        disableButton(facultybutton);
        disableButton(studentbutton);
        adminchanging=true;
        studentchanging=false;
        facultychanging=false;
        timer.start();
    }
});
this.buttonStyle(adminbutton);
adminbutton.setBounds(0, 0, 140, 35);
loginbuttonpanel.add(adminbutton);

facultybutton = new JButton("Faculty");
facultybutton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        activeButton(facultybutton);
        disableButton(studentbutton);
        disableButton(adminbutton);
        facultychanging=true;
        adminchanging=false;
        studentchanging=false;
        timer.start();
    }
});
this.buttonStyle(facultybutton);
facultybutton.setBounds(140, 0, 140, 35);
loginbuttonpanel.add(facultybutton);

studentbutton = new JButton("Student");
studentbutton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        activeButton(studentbutton);
        disableButton(facultybutton);
        disableButton(adminbutton);
        studentchanging=true;
        adminchanging=false;
        facultychanging=false;
        timer.start();
    }
});
studentbutton.setBounds(280, 0, 140, 35);
this.buttonStyle(studentbutton);
loginbuttonpanel.add(studentbutton);
```




```
        activeButton(studentbutton);

        underlinelabel = new JLabel("");
        underlinelabel.setBorder(new MatteBorder(3, 0, 0, 0,
(Color) Color.CYAN));
        underlinelabel.setBounds(underlinelabelx, 37,
underlinelabelwidth, 4);
        loginbuttonpanel.add(underlinelabel);

        bgimagelabel = new JLabel("image");
        bgimagelabel.setBounds(0, 11, 1380, 683);
        contentPane.add(bgimagelabel);
        this.setBackgroundImage();

    }

    //
    // protected void disposethis() {
    //     // TODO Auto-generated method stub
    //     this.dispose();
    // }

    @Override
    public void actionPerformed(ActionEvent e)
    {
        // TODO Auto-generated method stub
        if(!adminchanging&&!studentchanging&&!facultychanging)
        {

            imagenumber++;
            if(imagenumber>5)
            {
                imagenumber=1;
            }
            this.setBackgroundImage();
        }

        if(adminchanging)
        {
            if(adminpanelx==500)
            {
                adminchanging=false;
                timer.stop();
            }
            else
            {

                adminpanelx+=50;
                studentpanelx+=50;
                facultypanelx+=50;
            }
        }
    }
}
```



```
        underlinelabelx-=5;
    }
}
else if(facultychanging)
{
    if(facultypanelx==500)
    {
        facultychanging=false;
        timer.stop();
    }
    else
    {
        if(facultypanelx>500)
        {
            adminpanelx-=50;
            studentpanelx-=50;
            facultypanelx-=50;
            underlinelabelx+=5;
        }
        else
        {
            adminpanelx+=50;
            studentpanelx+=50;
            facultypanelx+=50;
            underlinelabelx-=5;
        }
    }
}
else if(studentchanging)
{
    if(studentpanelx==500)
    {
        studentchanging=false;
        timer.stop();
    }
    else
    {
        adminpanelx-=50;
        studentpanelx-=50;
        facultypanelx-=50;
        underlinelabelx+=5;
    }
}

studentloginpanel.setLocation(studentpanelx,studentpanely);
facultyloginpanel.setLocation(facultypanelx,
facultypanely);
adminloginpanel.setLocation(adminpanelx, adminpanely);
underlinelabel.setLocation(underlinelabelx,
underlinelabel.getY());
this.repaint();
```



```
}
public void buttonStyle(JButton button)
{
    button.setFocusable(true);
    button.setForeground(Color.WHITE);
    button.setFont(new Font("Segoe UI", Font.BOLD, 15));
    button.setBorder(new EmptyBorder(0,0,0,0));
    button.setBackground(Color.black);
    button.setFocusPainted(false);
    button.setBorderPainted(false);
    button.setOpaque(false);

}
public void activeButton(JButton button)
{
    button.setForeground(Color.cyan);

}
public void disableButton(JButton button)
{
    button.setForeground(Color.white);
}
public void setBackgroundImage()
{
    try {
        System.out.println(imagenumber);
        Image image=ImageIO.read(new
File("../assets//backgroundimage"+imagenumber+".jpg"));
        bgimagelabel.setIcon(new
ImageIcon(image.getScaledInstance(bgimagelabel.getWidth(),
bgimagelabel.getHeight(), Image.SCALE_SMOOTH)));

    } catch (IOException e) {

        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.SwingConstants;
import javax.swing.Timer;
import javax.swing.border.EmptyBorder;
import javax.swing.border.LineBorder;
import javax.swing.border.MatteBorder;

import collegeapplication.admin.Admin;
import collegeapplication.admin.AdminData;
import collegeapplication.common.DataBaseConnection;
```



```
@SuppressWarnings("serial")
public class LoginPageFrame extends JFrame implements ActionListener
{

    private JPanel contentPane;
    private JButton facultybutton;
    private JButton studentbutton;
    private JButton adminbutton;
    private LoginPanel
studentloginpanel, facultyloginpanel, adminloginpanel;
    private boolean
adminchanging=false, studentchanging=false, facultychanging=false;
    private int adminpanelx=-2300, adminpanely=240;
    private int facultypanelx=-900, facultypanely=240;
    private int studentpanelx=500, studentpanely=240;
    private int underlinelabelx=280, underlinelabelwidth=140;
    public Timer timer;
    private int imagenumber=1;
    private JLabel bgimagelabel;
    private JLabel underlinelabel;
    private JPanel loginbuttonpanel;
    public Timer imagetimer;
    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    if(DataBaseConnection.checkconnection())
                    {
                        LoginPageFrame frame = new LoginPageFrame();
                        frame.setVisible(true);
                        frame.setLocation(-7, 0);
                    }
                    else
                    {
                        JOptionPane.showMessageDialog(null,
"Start the Database Server first", "Error", JOptionPane.ERROR_MESSAGE);
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public LoginPageFrame() {
        timer=new Timer(5, this);
        imagetimer=new Timer(5000, this);
        imagetimer.start();
    }
}
```



```

        setTitle("Login");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(1380,733);
        setLocationRelativeTo(null);
        contentPane = new JPanel();
        contentPane.setBorder(new LineBorder(Color.LIGHT_GRAY, 2));
        contentPane.setBackground(new Color(255, 255, 255));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        Admin ad=new AdminData().getAdminData();

        JPanel panel = new JPanel();
        panel.setBackground(new Color(0, 139, 139,220));
        panel.setBounds(0, 26, 1364, 159);
        contentPane.add(panel);
        panel.setLayout(null);

        JLabel lblSilverOakCollage = new JLabel(ad.getCollageName());
        lblSilverOakCollage.setForeground(Color.WHITE);
        lblSilverOakCollage.setFont(new Font("Segoe UI", Font.BOLD,
30));

        lblSilverOakCollage.setHorizontalAlignment(SwingConstants.LEFT);
        lblSilverOakCollage.setBounds(160, 43, 749, 57);
        panel.add(lblSilverOakCollage);

        JLabel lblLogo = new JLabel("logo");
        lblLogo.setBounds(10, 10, 140, 140);
        lblLogo.setIcon(new
        ImageIcon(ad.getRoundedProfilePic(lblLogo.getWidth(),
        lblLogo.getHeight(), lblLogo.getWidth())));

        panel.add(lblLogo);

        studentloginpanel=new LoginPanel("Student",new
        ImageIcon("./assets/studentlogin.png"),this);
        studentloginpanel.setVisible(true);
        studentloginpanel.setLocation(studentpanelx,studentpanely);

        facultyloginpanel=new LoginPanel("Faculty",new
        ImageIcon("./assets/facultylogin.png"),this);
        facultyloginpanel.setVisible(true);
        facultyloginpanel.setLocation(facultypanelx, facultypanely);

        adminloginpanel=new LoginPanel("Admin",new
        ImageIcon("./assets/adminlogin.png"),this);
        adminloginpanel.setVisible(true);
        adminloginpanel.setLocation(adminpanelx, adminpanely);

```



```

contentPane.add(studentloginpanel);
contentPane.add(facultyloginpanel);
contentPane.add(adminloginpanel);

loginbuttonpanel = new JPanel()
{
    protected void paintComponent(Graphics
g)
    {
        g.setColor( getBackground() );
        g.fillRect(0, 0, getWidth(),
getHeight());
        super.paintComponent(g);
    }
};
loginbuttonpanel.setOpaque(false);
loginbuttonpanel.setBackground(new
Color(0,0,0,120));
loginbuttonpanel.setBounds(500, 189, 420, 40);
loginbuttonpanel.setLayout(null);
contentPane.add(loginbuttonpanel);

adminbutton = new JButton("Admin");
adminbutton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        activeButton(adminbutton);
        disableButton(facultybutton);
        disableButton(studentbutton);
        adminchanging=true;
        studentchanging=false;
        facultychanging=false;
        timer.start();
    }
});
this.buttonStyle(adminbutton);
adminbutton.setBounds(0, 0, 140, 35);
loginbuttonpanel.add(adminbutton);

facultybutton = new JButton("Faculty");
facultybutton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        activeButton(facultybutton);
        disableButton(studentbutton);
        disableButton(adminbutton);
        facultychanging=true;
        adminchanging=false;
        studentchanging=false;
        timer.start();
    }
});

```



```

        this.buttonStyle(facultybutton);
        facultybutton.setBounds(140, 0, 140, 35);
        loginbuttonpanel.add(facultybutton);

        studentbutton = new JButton("Student");
        studentbutton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                activeButton(studentbutton);
                disableButton(facultybutton);
                disableButton(adminbutton);
                studentchanging=true;
                adminchanging=false;
                facultychanging=false;
                timer.start();
            }
        });
        studentbutton.setBounds(280, 0, 140, 35);
        this.buttonStyle(studentbutton);
        loginbuttonpanel.add(studentbutton);
        activeButton(studentbutton);

        underlinelabel = new JLabel("");
        underlinelabel.setBorder(new MatteBorder(3, 0, 0, 0,
(Color)Color.CYAN));
        underlinelabel.setBounds(underlinelabelx, 37,
underlinelabelwidth, 4);
        loginbuttonpanel.add(underlinelabel);

        bgimagelabel = new JLabel("image");
        bgimagelabel.setBounds(0, 11, 1380, 683);
        contentPane.add(bgimagelabel);
        this.setBackgroundImage();

    }

    //
    // protected void disposethis() {
    //     // TODO Auto-generated method stub
    //     this.dispose();
    // }

    @Override
    public void actionPerformed(ActionEvent e)
    {
        // TODO Auto-generated method stub
        if(!adminchanging&&!studentchanging&&!facultychanging)
        {

            imagenumber++;
            if(imagenumber>5)

```



```
{
    imagenumber=1;

}
this.setBackgroundImage();
}

if(adminchanging)
{
    if(adminpanelx==500)
    {
        adminchanging=false;
        timer.stop();
    }
    else
    {
        adminpanelx+=50;
        studentpanelx+=50;
        facultypanelx+=50;
        underlinelabelx-=5;
    }
}
else if(facultychanging)
{
    if(facultypanelx==500)
    {
        facultychanging=false;
        timer.stop();
    }
    else
    {
        if(facultypanelx>500)
        {
            adminpanelx-=50;
            studentpanelx-=50;
            facultypanelx-=50;
            underlinelabelx+=5;
        }
        else
        {
            adminpanelx+=50;
            studentpanelx+=50;
            facultypanelx+=50;
            underlinelabelx-=5;
        }
    }
}
else if(studentchanging)
{
    if(studentpanelx==500)
    {
        studentchanging=false;
        timer.stop();
    }
}
```




```
        }
        else
        {

            adminpanelx-=50;
            studentpanelx-=50;
            facultypanelx-=50;
            underlinelabelx+=5;

        }
    }

    studentloginpanel.setLocation(studentpanelx,studentpanely);
    facultyloginpanel.setLocation(facultypanelx,
facultypanely);
    adminloginpanel.setLocation(adminpanelx, adminpanely);
    underlinelabel.setLocation(underlinelabelx,
underlinelabel.getY());
    this.repaint();

}
public void buttonStyle(JButton button)
{
    button.setFocusable(true);
    button.setForeground(Color.WHITE);
    button.setFont(new Font("Segoe UI", Font.BOLD, 15));
    button.setBorder(new EmptyBorder(0,0,0,0));
    button.setBackground(Color.black);
    button.setFocusPainted(false);
    button.setBorderPainted(false);
    button.setOpaque(false);

}
public void activeButton(JButton button)
{
    button.setForeground(Color.cyan);

}
public void disableButton(JButton button)
{
    button.setForeground(Color.white);
}
public void setBackgroundImage()
{
    try {
        System.out.println(imagenumber);
        Image image=ImageIO.read(new
File("../assets//backgroundimage"+imagenumber+".jpg"));
        bgimagelabel.setIcon(new
ImageIcon(image.getScaledInstance(bgimagelabel.getWidth(),
bgimagelabel.getHeight(), Image.SCALE_SMOOTH)));
    }
}
```



```
        } catch (IOException e) {  
  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        }  
    }  
}
```