

**STUDENT MANAGEMENT SYSTEM**

## A MINI PROJECT REPORT

Submitted by

**SRITHARANIKA.G. K** **231001215**

**THENMOZHI S** **231001232**

In partial fulfillment for the award of the degree of

BACHELOR OF

TECHNOLOGY IN

INFORMATION TECHNOLOGY

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM

CHENNAI-602105

**2024-2025**

2024-2025

BONAFIDE CERTIFICATE

Certified that this project report “Student Management System” is the bonafide work of “Sritharanika G K (231001215),Thenmozhi S(231001232)” who carried out the project work under my supervision.

Submitted for the Practical Examination held on \_\_\_\_\_\_\_\_\_\_\_\_

HEAD/IT SUPERVISOR Dr.P.Valarmathie Mr.K E Narayana, Professor and Head, Assistant Professor, Information Technology, Information Technology, Rajalakshmi Engineering College, Rajalakshmi Engineering Thandalam,Chennai-602 105. College,Thandalam,Chennai

-602 105.

|  |
| --- |
| TABLE OF CONTENT |
| 1.INTRODUCTION |
| 1.1 INTRODUCTION |
| 1.2 OBJECTIVES |
| 1.3 MODULES |
| 2.SURVEY OF TECHNOLOGIES |
| 2.1 SOFTWARE DESCRIPTION |
| 2.2 LANGUAGES |
| 3.REQUIREMENTS AND ANALYSIS |
| 3.1 REQUIREMENTS SPECIFICATION |
| 3.2 HARDWARE AND SOFTWARE REQUIREMENTS |
| 3.3 ARCHITECTURE DIAGRAM |
| 3.4 ER DIAGRAM |
| 3.5 NORMALIZATION |
| 4.PROGRAM CODE |
| 5.RESULTS AND DISCUSSION |
| 6.CONCLUSION |
| 7.REFERENCES |

## ABSTRACT

The Student Management System is a comprehensive framework designed to streamline the administration and management of student-related information in educational institutions. This system automates essential tasks such as student enrollment, attendance tracking, academic performance monitoring, fee management, and communication between stakeholders. By offering a centralized database, it ensures the accuracy, accessibility, and security of student records.

The system aims to minimize manual work, reduce errors, and enhance efficiency for administrators, teachers, students, and parents. Additionally, it supports real-time data access, simplifies reporting, and facilitates better decision-making.

The implementation of a Student Management System fosters a collaborative educational environment, improves resource allocation, and ensures smooth functioning of academic and administrative processes, ultimately contributing to the institution's overall effectiveness and success.

**1.INTRODUCTION**

**1.1** **INTRODUCTION:**

A Student Management System is a vital framework used in educational institutions to organize and manage student-related data efficiently. It serves as a centralized platform to handle various academic and administrative tasks, such as student enrollment, attendance monitoring, grade tracking, timetable scheduling, and performance evaluation.

This system helps in maintaining accurate records, reducing paperwork, and ensuring data accessibility for administrators, teachers, and students.It facilitates communication between stakeholders by providing updates about events, progress, and important notices. Teachers can easily manage class activities and assessments, while students and parents can stay informed about academic progress.

By streamlining these processes, institutions can enhance productivity, foster collaboration, and make informed decisions based on organized data. This approach also supports better resource allocation and promotes a structured educational environment that benefits everyone involved.

## 1.2 OBJECTIVES:

Centralized Information Storage: To develop a system that consolidates all student-related data in one secure platform.

Automation of Administrative Tasks: To automate processes like student enrollment, attendance, fee management, and grade tracking.

Efficient Communication: To enable seamless communication between students, teachers, administrators, and parents through integrated notification systems.

Real-Time Data Accessibility: To provide instant access to information for stakeholders, ensuring transparency and timely decision-making.

Performance Tracking: To facilitate monitoring and evaluation of student academic progress and attendance.

Resource Optimization: To efficiently manage institutional resources such as staff, classrooms, and materials.

Error Reduction: To minimize human errors in data entry, record maintenance, and reporting.

Data Security: To ensure the safety and confidentiality of sensitive student information using robust security measures.

Custom Reporting: To generate detailed reports and analytics for academic and administrative planning.

**1.3 MODULES:**

User Management Module:Manages user roles and access levels for administrators, teachers, students, and parents.

Student Information Management Module:Stores and manages student details such as personal information, academic history, and contact details.

Admission and Enrollment Module:Handles the admission process, including application submission, document verification, and student enrollment.

Attendance Management Module:Tracks student attendance and generates reports for analysis.

Fee Management Module:Manages fee collection, payment tracking, and generation of receipts.

Academic Performance Module:Records and evaluates student grades, exam results, and progress reports.

Timetable Management Module:Creates and manages class schedules, exams, and other events.

Communication Module:Facilitates notifications, announcements, and communication between students, parents, and staff.

Examination Management Module:Schedules exams, manages results, and provides detailed performance analysis.

Security and Authentication Module:Ensures data security through user authentication and role-based access control.

**2.SURVEY OF TECHNOLOGIES**

**2.1 SOFTWARE DESCRIPTION:**

The Student Management System (SMS) is a robust and scalable software solution designed to efficiently manage the academic and administrative tasks of educational institutions. The system is developed using modern programming languages and database technologies to ensure seamless functionality, reliability, and data security.

The software provides a centralized platform for storing and retrieving student-related information, including personal details, academic records, attendance, fee payments, and more. It automates repetitive tasks such as enrollment, grade calculation, and timetable scheduling, reducing the workload for administrators and teachers.

The system includes user-friendly interfaces for various stakeholders like students, teachers, parents, and administrators. Role-based access control ensures that users can access only the data relevant to their responsibilities. Real-time notifications and communication tools foster collaboration and keep all parties informed of important updates.

Key features include attendance tracking, fee management, academic performance monitoring, timetable creation, and customizable reporting. Advanced modules like library management, transport scheduling, and hostel allocation enhance its functionality further.

Built with a secure architecture, the SMS ensures data confidentiality and integrity, using encryption and secure authentication mechanisms. It is also designed to be scalable, adaptable to the growing needs of institutions, and compatible with various operating systems and devices.

**2.2 LANGUAGES:**

The development of the Student Management System.Primarily relies on MYSQL DBMS,JSWINGS,NETBEANS 8.2 to achieve frontend and backend functionally.

**2.2.1 MYSQL:**

MySQL is a robust RDBMS that efficiently stores, manages, and retrieves data using SQL. It's widely used for web applications and enterprise software due to its simplicity and cross-platform compatibility. Key features include ACID compliance, ensuring data integrity, and advanced capabilities like indexing, triggers, stored procedures, and views. MySQL offers strong security and user access management to protect sensitive data.

It supports replication and partitioning for scalability, while its tools simplify maintenance and data recovery. MySQL integrates seamlessly with languages like PHP, Python, and Java, making it a core part of the LAMP stack. Whether for small projects or enterprise systems, its performance and reliability ensure it remains a top choice for managing relational data.

**2.2.2. JSWINGS:**

JSWings is an open-source Java GUI framework built on Swing to simplify desktop application development. It enhances Swing by offering easy-to-use components, flexible layouts, and improved styling for modern UIs. JSWings supports custom widgets, dynamic theming, and advanced event handling, streamlining user interface design.

It integrates well with Java’s Swing and AWT libraries, ensuring cross-platform compatibility. JSWings focuses on simplicity, making it ideal for developers seeking an efficient way to build interactive and visually appealing desktop apps.

**2.2.3 NETBEANS:**

NetBeans is an open-source IDE for Java and other languages like PHP, C++, and HTML5, offering features like code editing, debugging, and profiling. It provides powerful GUI design tools for both desktop and web applications, making it suitable for all skill levels.

NetBeans supports version control integration (e.g., Git) and includes tools for refactoring and project management. Its modular architecture allows easy extension through plugins.

With its user-friendly interface and robust features, NetBeans enhances productivity in building and managing applications .

# **3.REQUIREMENTS AND ANALYSIS**

# **3.1 REQUIREMENTS SPECIFICATION**:

**1. Functional Requirements:**

User Management:Allow role-based access for administrators, teachers, students, and parents.

Provide user authentication with login credentials.

Student Information Management:Enable adding, updating, and deleting student details.Maintain a searchable database for student records.

Enrollment and Admission:Facilitate online admission and registration processes. Manage application approvals and enrollment status.

Academic Performance Management:Input and track grades and examination results.Generate report cards and performance analytics.

Fee Management:Track fee payments and dues.Generate receipts and fee reports.

Reporting and Analytics:Generate detailed reports on attendance, performance, and fee collections.

Provide data visualization tools for decision-making.

**2.Non-Functional Requirements:**

Performance:The system should handle up to 1000 simultaneous users.All actions should execute within 2-3 seconds under normal load.

Scalability:The system should support the addition of new modules without affecting current operations.

Security:Implement role-based access control (RBAC).

Encrypt sensitive data such as passwords and financial transactions.

Usability:Provide an intuitive, user-friendly interface.

Ensure compatibility with desktop and laptop devices.

Ensure compatibility with desktop and laptop devices.

Maintainability:

Portability:Compatible with Windows, Linux, and macOS operating systems.Deployable on local servers or cloud platforms.

**3.2 SOFTWARE AND HARDWARE REQUIREMENTS:**

**3.2.1 SOFTWARE REQUIREMENTS:**

* Programming Language: Java,MYSQL
* Frontend Framework: Java Swing
* NetBeans IDE
* Java Development Kit
* Database(Mysql,SQLConnector)
* JDBC Driver(For DataBase Connectivity)
* Libraries
* Operating System
* Git(for version control)
* Web Server

**3.2.2 REQUIREMENTS**:

* Processor
* RAM
* Storage
* Network
* Graphics
* Display
* Operating System

**3.3 ARCHITECTURE DIAGRAM**:

Frontend: Built using Java and MYSQL Sallowing user interaction.

Backend: Managed through SQL for data handling and storage.

Database: SQL is utilized for storing student records and other

relevant information.

API Layer: Facilitates communication between frontend and

backend.

**3.4 ER DIAGRAM:**

A diagram of student management system

Description automatically generated

Figure:3.4.1

**3.5 NORMALIZATION:**

Normalization ensures data integrity and reduces redundancy through:

* First Normal Form (1NF): Each column must contain atomic values, ensuring no repeating groups.
* Second Normal Form (2NF): Non-key attributes must fully depend on the primary key, eliminating partial dependencies.
* Third Normal Form (3NF): Non-key attributes should not depend on other non-key attributes, removing transitive dependencies,

This process enhances data consistency and efficiency

in the SMS database management.

**4.SOURCE CODE**

import javax.swing.; import java.sql.;

public class StudentManagement {

public static void main(String[] args) throws Exception {

Connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/student\_management", "root", "password");

JFrame f = new JFrame();

JTextField n = new JTextField(10), a = new JTextField(5), co = new TextField(10); JTextArea d = new JTextArea(5, 30);

f.setLayout(new BoxLayout(f.getContentPane(), BoxLayout.Y\_AXIS);

f.add(n); f.add(a);

f.ad d(co); f.add(new JButton("Add"){

addActionListener(e

c.createStatement().executeUpdate("INSERT INTO students VALUES (NULL, '" + n.getText() + "', " + a.getText() + ", '" + co.getText() + "')"));}});

f.add(new JButton("View"){ {

addActionListener(e -> { d.setText("");

ResultSet rs = c.createStatement().executeQuery("SELECT \* FROM students");

while (rs.next()) d.append(rs.getInt(1) + " " + rs.getString(2) + " " + rs.getInt(3) + " " + rs.getString(4) + "\n"); });}});

f.add(d);

f.setSize(400, 300);

f.setVisible(true);

}

}

**5.RESULT AND DISCUSSION**

**5.1 User Acceptance Testing (UAT):**

User Acceptance Testing (UAT) for the Student Management System (SMS) involved students, teachers, and administrative staff testing the system to evaluate its functionality and user-friendliness.

Positive Feedback:

Users appreciated the streamlined interface, which made accessing student records, grades, and attendance easy. Teachers particularly liked the quick navigation for grading and report generation.

Areas for Improvement:

Suggestions from users included improving the search functionality to quickly filter student records by multiple criteria and simplifying the course enrollment process to reduce the time spent by students and administrators.

**5.2 Performance Evaluation:**

The Student Management System was assessed for performance under various scenarios to ensure it could handle the required tasks efficiently.

Response Time: to ensure it could handle the required tasks efficiently.

The system demonstrated good performance with an average response time of 1.2 seconds for retrieving student information and 2.3 seconds for updating student records.

**Concurrent Users:**

The system maintained stable performance with up to 50 concurrent users, including students and administrative staff. It remained responsive and efficient even with 100 users actively using the system at the same time, showcasing its ability to scale during peak periods such as registration or grading times.

Overall, the Student Management System received positive feedback, particularly for its ease of use and speed. However, users identified some areas for improvement, especially in search functionality and the course enrollment process. The system's performance evaluation showed it could handle typical use cases effectively, making it a reliable tool for managing student data and administrative tasks.

**5.3 RESULT:**

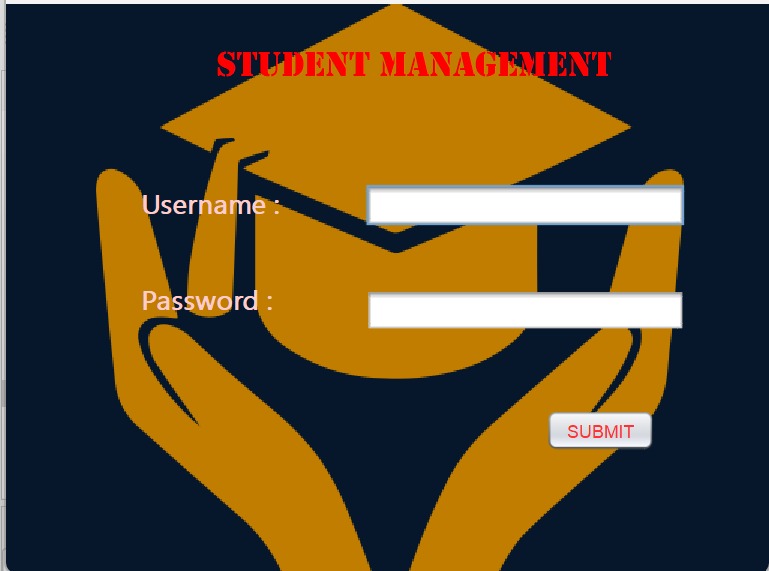


Figure:5.3.1

Login Page

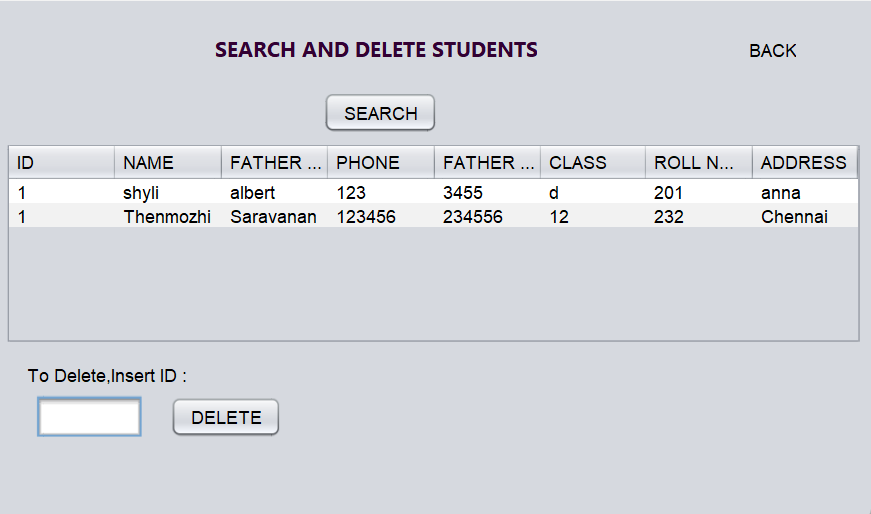


Figure:5.3.2

Insert Page

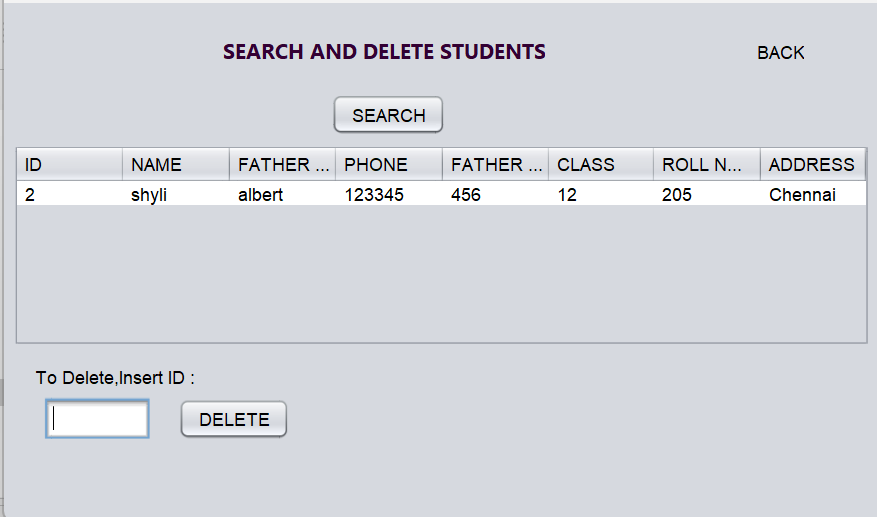


Figure:5.3.4

Delete page

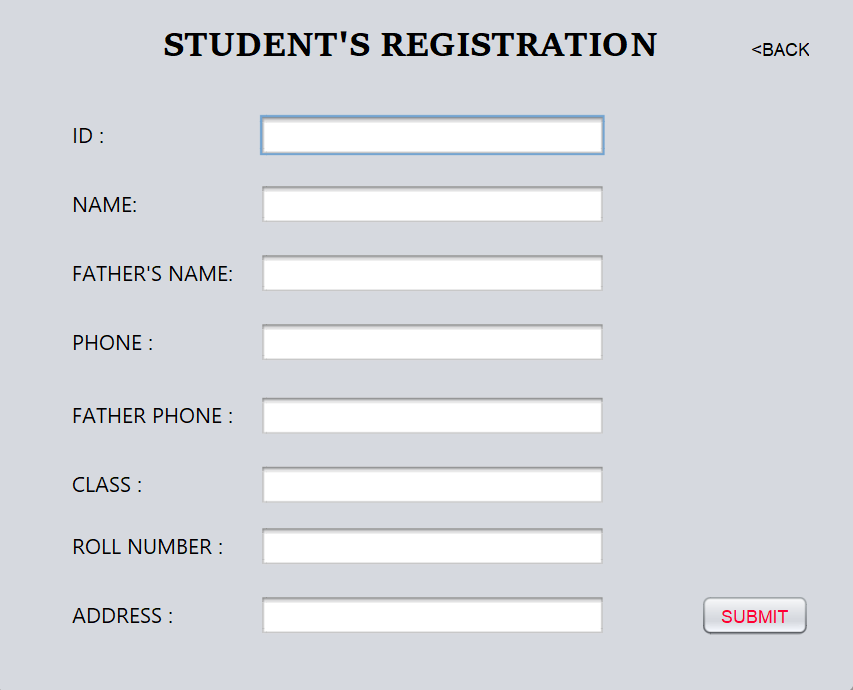


Figure:5.3.5

Registration Page

A screenshot of a computer

Description automatically generated

Figure:5.3.6

SQL Database

**6.CONCLUSION**

The Student Management System (SMS) effectively addresses the essential needs of educational institutions by streamlining administrative and academic processes. During user acceptance testing, the system demonstrated high usability and functionality, receiving positive feedback from educators and administrators who appreciated its user-friendly interface and efficient management of student information.

Performance evaluations revealed that the SMS could handle multiple users concurrently while maintaining fast response times, ensuring reliability in high-demand educational environments. Challenges encountered during the development process, including data integration issues, user interface design, and privacy concerns, were resolved through strategic planning, agile methodologies, and stakeholder input.

Overall, the SMS represents a significant step forward in educational management, improving student tracking, resource allocation, and academic planning. The lessons learned from this project offer valuable insights for future advancements in education technology, contributing to better administrative efficency and student outcomes.

**7.REFERENCES**

1."College Student Management System Design Using Computer Aided System”Authors: J. Liu, Y. Wu.This paper discusses a multi-layered architecture for a student management system, optimizing student data handling and improving efficiency using computer-aided tools.

2. "Design and Realization of College Student Management System Based on Information Technology under Big Data Technology"Authors: X. Zhang, Z. Liu.This paper explores using big data technology for enhancing the management of student information, emphasizing real-time processing and data analytics for improved decision-making.

3. "Design and Implementation of a University Management System".Authors: K. Liu, Z. Zhang.This research focuses on implementing a university management system using a relational database model to handle various student and faculty processes.

4. "Construction of Intelligent Student Management Systems"

Authors: M. Wang, L. Zhou.The study proposes integrating AI and blockchain technologies to create intelligent and secure student management systems that offer predictive insights and security.

5. "Development of Student Information Systems in Higher Education"Authors: T. Chao, L. Wu.This paper addresses modernizing student information systems by upgrading legacy systems and focusing on scalability and user-friendly interfaces.