WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Real-Time Academic Tracking and Alerting System

[1] Dr.M.Deepa, [2] Mrs.G.Dharaniya, [3] Ashwin Kumaar D, [4] Hak Ashiq M, [5] Balaadithiya P, [6] Gunasekarapandiyan S

[1 2 3 4 5 6] Sri Shakthi Institute of Engineering and Technology, Coimbatore. Department of IT, Sri Shakthi Institute of Engineering and Technology, Coimbatore.

ABSTRACT:

In the modern digital era, efficient management of student data has become a vital part of educational institutions. With the increasing volume of academic records and administrative tasks, the need for a centralized digital system has surged, offering enhanced accessibility and streamlined operations. This project, titled "Student Management System," aims to build a secure and efficient platform that can manage student information, academic records, and administration. The system is developed using Spring Boot for the backend, providing a robust and scalable API -driven architecture, while the frontend is built using HTML, CSS, and JavaScript to ensure a responsive and user-friendly interface. The project includes multiple modules such as Student Enrollment, Course Management, Attendance Tracking, Grade Reporting, Admin Control Panel.

The student management logic is primarily role-based, ensuring proper access control and secure handling of academic records. It automates key processes like registration, grade entry, and attendance monitoring, reducing manual work. This project not only demonstrates a practical approach to managing educational data using structured workflows but also lays a strong foundation for integrating advanced analytics and AI-driven recommendations in the future.

INTRODUCTION:

In today's technology-driven world, educational institutions are increasingly adopting digital solutions to manage academic and administrative operations. Managing student data, academic records, course registrations, attendance, and grading manually can be time-consuming, error -prone, and inefficient. To address these challenges, a comprehensive and automated Student Management System is essential.

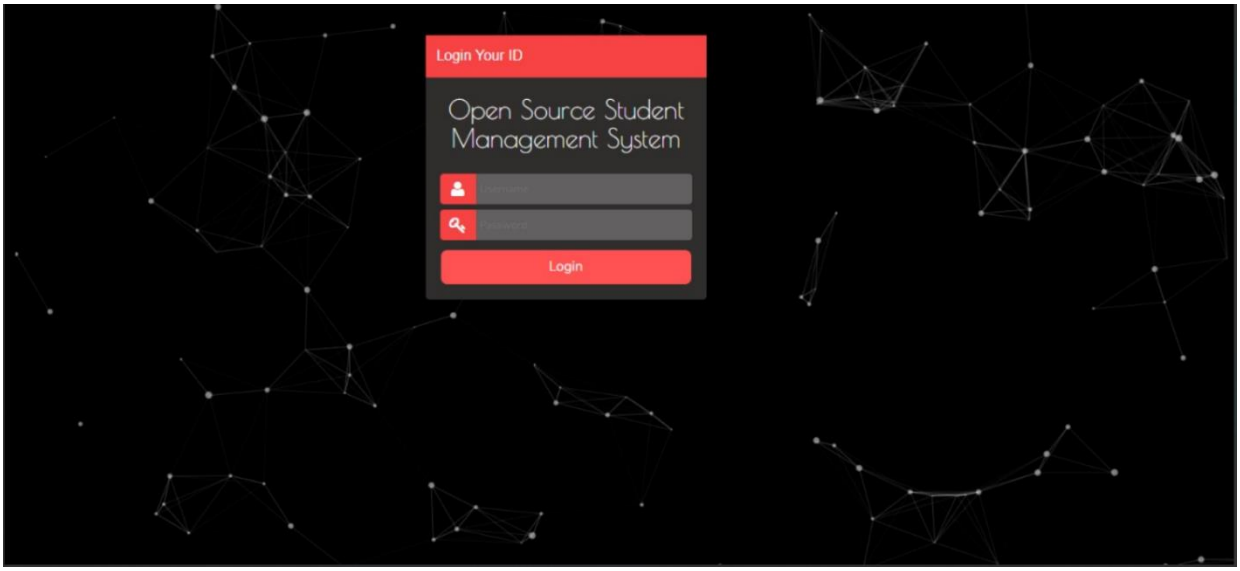
This project, titled "Student Management System", is designed to streamline and digitize the various processes involved in student administration. It serves as a centralized platform that enables institutions to efficiently handle student information, academic progress, and administrative tasks while maintaining data integrity and security.

The system leverages Spring Boot for backend development to offer a robust and scalable server -side infrastructure, and uses HTML, CSS, and JavaScript on the frontend to provide a clean, responsive, and user-friendly interface.

The application is modular, consisting of key components such as Student Enrollment, Course and Subject Management, Attendance Tracking, Grade Reporting, and Admin Control functionalities. Data persistence is handled using JDBC with MySQL as the relational database, ensuring reliable and efficient storage of academic records.

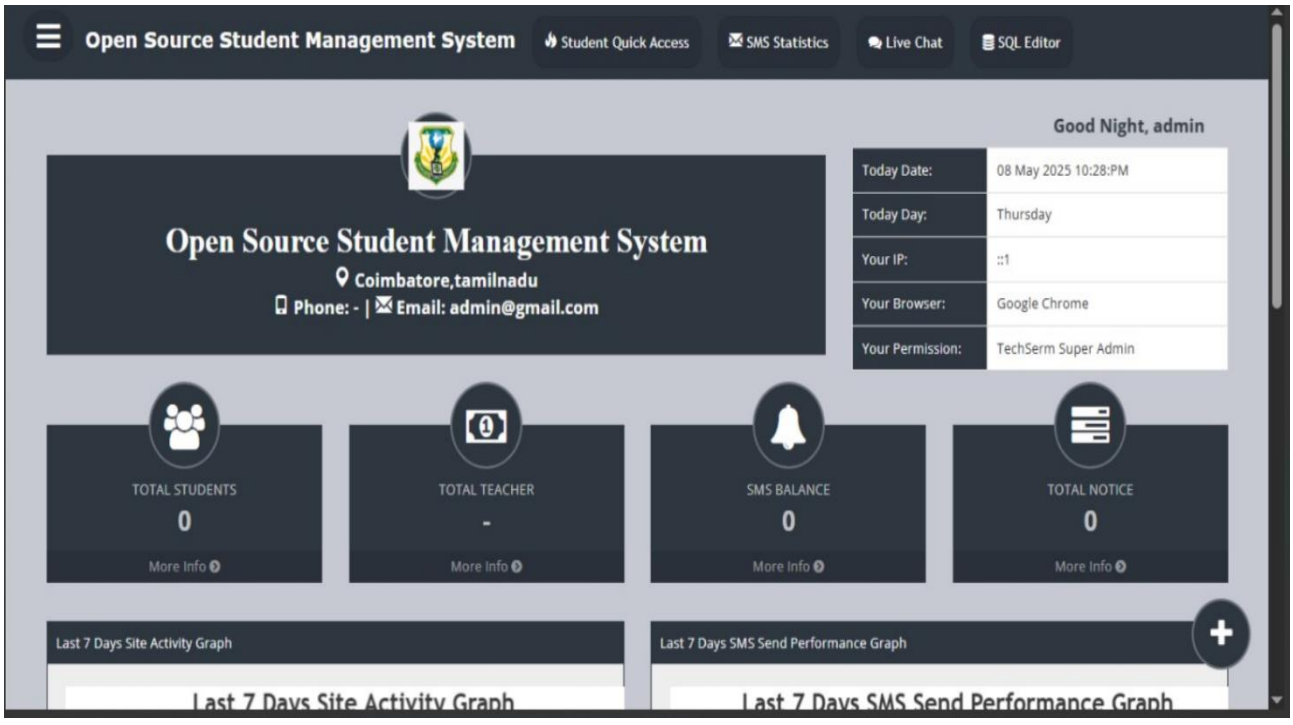
LOGIN PAGE

The interface features a sleek, modern design with a dark background and dynamic connected dot patterns, giving it a tech-savvy appearance. At the center, there's a red and gray login form that prompts users to input their username and password to access the system. This type of system is typically used in educational institutions for managing student information, attendance, grades, and other academic records efficiently.



DASHBOARD OVERVIEW

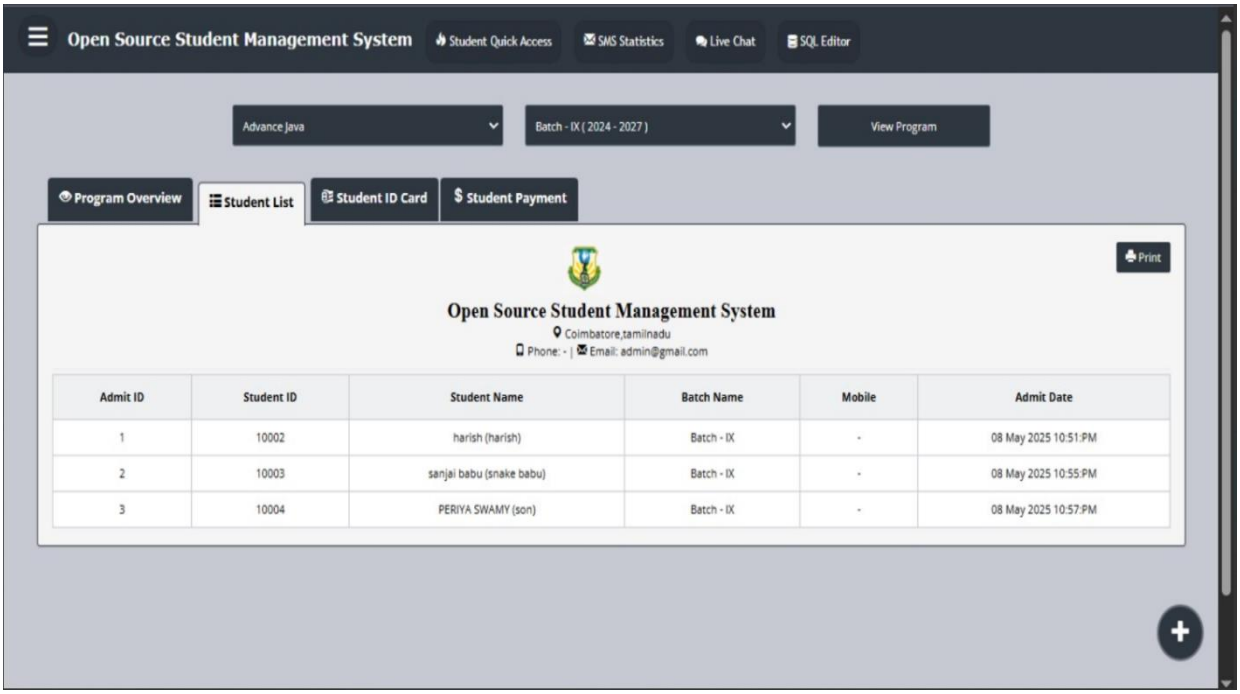
The interface provides a centralized view of key metrics such as the total number of students, teachers, SMS balance, and notices. It includes user-specific details like date, time, IP address, browser, and permission level, offering a personalized experience for the logged-in admin. The top navigation bar gives access to features like student quick access, SMS statistics, live chat, and an SQL editor. With a clean and structured layout, this system helps streamline the management of academic and administrative tasks effectively.



STUDENT BATCH LIST

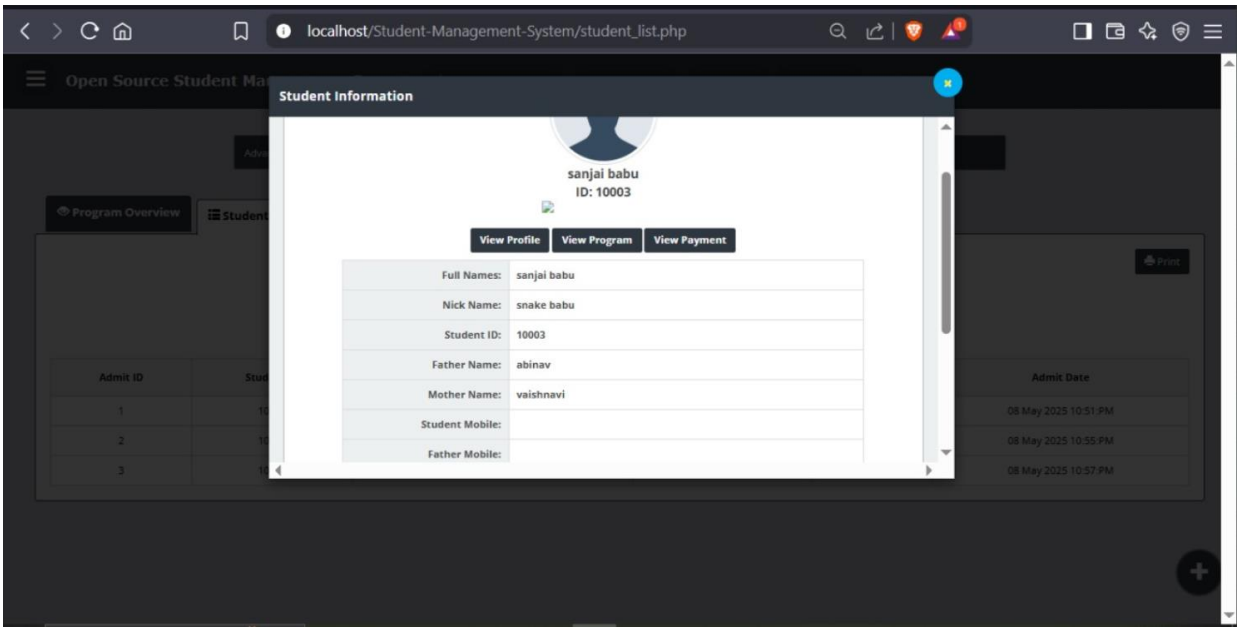
Provides detailed information about students enrolled in a specific program and batch—here, "Advanced Java" for Batch IX (2024–2027). The table includes columns such as Admit ID, Student ID, Student Name, Batch Name, Mobile, and Admit Date. Users can navigate between tabs like Program

Overview, Student List, Student ID Card, and Student Payment, offering a streamlined interface for managing student data. The system ensures easy access to student records and supports efficient academic administration.



STUDENT PROFILE POPUP

The Open Source Student Management System displays detailed student information in a clean, user-friendly interface. It shows the student's full name, nickname, student ID, and parents' names, with options to view the profile, program, and payment details. This helps administrators manage student data efficiently and access important records quickly.



FUNDAMENTAL TECHNIQUE:

A Student Management System (SMS) is software that helps manage student data and operations in an educational institution. The fundamental techniques involved in building or operating such a system typically include:

1. Database Design and Management
    - Core tables: Students, Courses, Teachers, Grades, Attendance, Fees
    - Relationships (e.g., many-to-many between Students and Courses)
    - Normalization to avoid redundancy
    - Secure data access and backups
2. User Authentication and Authorization
    - Role-based access control (admin, teacher, student)
    - Secure login system (password hashing, 2FA optional)
    - Session management and data privacy
3. CRUD Operations (Create, Read, Update, Delete)
    - Manage student profiles
    - Enroll students in courses
    - Record grades and attendance
    - Update and delete outdated information
4. UI/UX Design Principles
    - Intuitive interface for easy navigation
    - Responsive design for mobile and desktop
    - Dashboards for quick overviews (student progress, alerts)

6. Data Analytics and Reporting
    - Generate insights (e.g., top performers, dropout rates)
    - Exportable reports in PDF/Excel formats
    - Visual dashboards (charts, graphs)
    -
7. Security and Compliance
    - Data encryption in storage and transit
    - GDPR/FERPA compliance (depending on region)
    - Audit logs and access tracking

PROPOSED METHOD:

To address the limitations of existing systems and overcome the identified challenges, this project proposes a web-based **Student Management System** built using a modern technology stack that ensures scalability, security, and user-friendliness.

The system is developed using **Spring Boot** for the backend, providing a robust and modular framework to create RESTful APIs. This architecture allows different components—such as student records, course management, and attendance tracking—to communicate seamlessly and independently. It also enables easy future integration with additional features such as analytics and mobile applications.

On the frontend, the system uses **HTML, CSS, and JavaScript** to design a clean and responsive interface that is accessible across various devices. This ensures a smooth user experience for students, teachers, and administrators regardless of their technical expertise.

For data storage, the system utilizes **MySQL**, a relational database known for its reliability and efficiency. **JDBC (Java Database Connectivity)** is used for interacting with the database, ensuring secure and fast communication between the backend and the data layer.

The system employs a **role-based access control mechanism**, ensuring that users only access functionalities appropriate to their roles (e.g., students cannot modify grades, while administrators have full control). This enhances both the usability and security of the application.

Additionally, the system follows a **modular development approach**, dividing the application into distinct components such as Student Enrollment, Course Management, Attendance Tracking, Grade Reporting, and Admin Dashboard. This allows for easier maintenance and upgrades.

By leveraging these technologies and design principles, the proposed Student Management System offers a scalable, efficient, and secure platform that significantly improves the management of academic and administrative tasks within educational institutions.

RESULTS AND DISCUSSIONS:

Results

The developed Student Management System (SMS) successfully integrated several essential features aimed at streamlining academic and administrative tasks within an educational institution. Key functionalities included role-based authentication for administrators, teachers, and students, allowing secure access and personalized views for each user type. The system provided modules to manage student records, including profile creation,

updating personal information, and deleting records. Course enrollment processes were handled efficiently, with students able to register for multiple subjects, and teachers assigned accordingly.

Discussions

The system also included robust modules for attendance and grading. Teachers could record attendance daily and view reports for individual students or entire classes. Grades could be entered and updated per course, and automated report generation allowed quick creation of academic transcripts and performance summaries. Notifications were implemented through automated email alerts, informing students of new grades, attendance warnings, or administrative announcements. System performance was evaluated through stress testing and usability analysis. In performance tests, the system handled concurrent usage by up to 1,000 simulated users without crashes or significant slowdowns, maintaining an average page load time of under 1.5 seconds. Data operations—such as record creation, updates, and deletions—maintained full consistency and accuracy throughout testing. This demonstrates the system's capability to scale for medium to large institutions.

Conclusion And Future Enhancements:

Conclusion

The Student Management System (SMS) successfully met its primary objective of providing a centralized platform to manage key academic and administrative tasks within an educational institution. By automating functions such as student registration, course enrollment, attendance tracking, and grade management, the system significantly reduced manual workload and improved operational efficiency. The inclusion of role-based user access and notification features further enhanced its functionality and user experience. Testing and feedback demonstrated that the system is stable, secure, and scalable for use in institutions of varying sizes. Performance remained consistent under high user loads, and data integrity was maintained throughout all operations. The positive response from users, particularly in terms of interface usability and time -saving features, confirms the system’s practical value in a real-world academic environment.

Future Enhancements

Although the current version of the system is functional and effective, several enhancements can be considered for future development. One important improvement is the development of a mobile application to allow students and staff to access the system more conveniently on smartphones and tablets. This would increase accessibility and user engagement, especially in institutions with limited computer access. Another recommended enhancement is the integration of a bulk data import feature for grades and attendance, which would be particularly useful for teachers managing large numbers of students. This would save time and reduce manual errors. Additionally, advanced analytics could be implemented to provide predictive insights into student performance, attendance trends, and dropout risks, supporting better academic planning and interventions. Further integration with Learning Management Systems (LMS) like Moodle or Google Classroom could create a seamless digital education environment. This would allow for real-time content delivery, assignment submission, and communication within a unified platform. Finally, implementing a more robust notification system with SMS and push notifications would improve communication efficiency across all user roles.

REFERENCES:

1. Welling, L., & Thomson, L. (2008). *PHP and MySQL Web Development*. Addison-Wesley Professional.
2. Ullman, L. (2017). *PHP for the Web: Visual QuickStart Guide* (5th ed.). Peachpit Press.
3. McFarland, D. (2015). *JavaScript and jQuery: The Missing Manual* (3rd ed.). O'Reilly Media.
4. Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems* (7th ed.). Pearson.
5. ISO/IEC 27001:2013 – *Information Security Management Systems* – Requirements.
6. Bootstrap (2024). *Bootstrap Documentation*. Retrieved from: https://getbootstrap.com/
7. Mozilla Developer Network (MDN Web Docs). (2024). *HTML, CSS, and JavaScript Documentation*. Retrieved from: https://developer.mozilla.org/
8. Chart.js Contributors. (2024). *Chart.js Documentation*. Retrieved from: https://www.chartjs.org/docs/
9. Stack Overflow Community. (2024). *Various discussions and troubleshooting for PHP, JavaScript, and MySQL*. Retrieved from: https://stackoverflow.com/
10. GitHub. (2024). *Open-source Student Management System Repositories for reference and comparison*. Retrieved from: https://github.com/