iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80266

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Design a School ERP System using Java

Saransh Chaudhary[1], Pradeep Jaiswal[2]
[1]*Department of Data Science & Design Greater Noida Institute of Technology, Faridabad, Haryana, India*
[2]*Department of Data Science & Design Greater Noida Institute of Technology, Noida, U.P, India*

Abstract: ***In the modern educational landscape, institutions are tasked with managing vast amounts of academic and administrative data. Traditional manual methods, relying on paper-based records and disparate spreadsheets, are increasingly inefficient, time-consuming, and prone to human error and data redundancy. This project presents the design and implementation of a comprehensive School Enterprise Resource Planning (ERP) System aimed at automating and streamlining the complex day-to-day operations of educational institutions.***

The proposed system is designed as a modular software solution that integrates critical functional areas—including Student Admission, Fee Management, Attendance Tracking, Examination Grading, and Staff Payroll—into a single, unified platform. By utilizing a centralized database architecture, the design ensures data integrity, consistent updates, and the elimination of information silos across departments. The system architecture prioritizes role-based access control (RBAC), providing secure and distinct interfaces for Administrators, Teachers, Students, and Parents to ensure privacy and data security. Furthermore, the design focuses on scalability and user experience, enabling the generation of real-time reports that assist administration in strategic decision-making.

The implementation of this ERP system significantly reduces administrative overhead, fosters a paperless environment, and improves communication channels between stakeholders. Ultimately, this project demonstrates how digital transformation in school management can lead to enhanced operational efficiency, better resource allocation, and a more focused academic environment.

Keywords: School Automation, ERP, Database Management, Role-Based Access Control, Information Systems.

I. INTRODUCTION

In the era of rapid digital transformation, information technology has become the backbone of efficient organizational management. For educational institutions, the transition from traditional, manual record-keeping to automated digital systems is no longer a luxury but a necessity. Schools today manage complex ecosystems involving thousands of students, staff members, and distinct administrative departments, generating massive volumes of data daily. As institutions grow, the ability to manage this data effectively determines the operational success and academic reputation of the school.

Conventional methods of school management—often relying on physical files, huge registers, and non-integrated spreadsheet software—pose significant challenges. These manual systems are labor-intensive, prone to human error, and susceptible to data redundancy. For instance, a student's address change might be updated in the admission file but remain outdated in the library or transport records. Furthermore, retrieving specific historical data or generating comprehensive institutional reports is a time-consuming ordeal, creating administrative bottlenecks that distract from the core mission of education.

Proposed Solution the School Enterprise Resource Planning (ERP) System proposed in this project addresses these challenges by offering a robust, unified software solution. It is designed to digitize and integrate the entire spectrum of school operations, ranging from the initial student admission process to fee collection, timetable generation, attendance monitoring, and result declaration.

By leveraging a centralized database architecture, the system ensures that data is entered once and is immediately accessible across all authorized modules, eliminating inconsistency. The system also bridges the communication gap between stakeholders; parents can track their ward's progress in real-time, teachers can streamline grading processes, and administrators can exercise better financial control. Ultimately, this project aims to foster a "paperless office" environment, ensuring high data security, transparency, and operational speed, thereby enabling the institution to focus its resources on academic excellence rather than administrative maintenance.

II. PROBLEM DEFINITION

The core problem addressed by this project is the inefficiency and unreliability inherent in manual school management processes. Many educational institutions still rely on traditional methods such as physical registers, file-based storage, and non-integrated spreadsheets to handle critical data like student admissions, fee collection, attendance, and academic grading.

©IJRASET: All Rights are Reserved

These decentralized manual systems create severe data redundancy and inconsistency. For example, a student's contact update in the admission office is often not reflected in the library or transport records, leading to communication failures. Furthermore, physical data storage makes information retrieval a slow, labor-intensive process; generating a simple report on "fee defaulters" or "attendance averages" can take hours of cross-referencing different ledgers.

This lack of automation results in a high margin for human error in calculations (such as fee balances or grade averages) and poses a security risk, as physical records are vulnerable to loss, damage, or unauthorized access. Consequently, administrative staff spend a disproportionate amount of time on repetitive clerical tasks rather than strategic management, creating an urgent need for a centralized, automated School ERP solution to streamline operations and ensure data integrity.

III. LITERATURE REVIEW

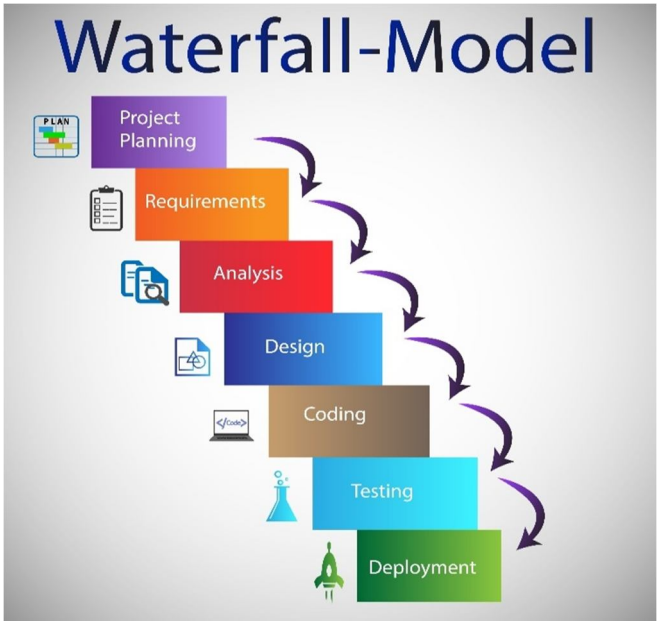
The evolution of educational administration has been a subject of extensive research, highlighting a distinct shift from manual record-keeping to sophisticated digital solutions. Early studies on school management emphasized the limitations of paper-based systems, noting that physical file maintenance resulted in significant data redundancy, slow information retrieval, and high operational costs. Contemporary literature focuses on the adoption of Management Information Systems (MIS) and web-based ERPs. Research indicates that schools adopting centralized database architectures experience a marked reduction in administrative workload and an increase in data accuracy. For instance, automated attendance and fee processing have been proven to reduce clerical errors by over 30% compared to manual ledgers. However, existing reviews also identify gaps in current market solutions. Many available systems are either prohibitively expensive proprietary software or fragmented standalone applications (e.g., separate software for library and fees) that fail to communicate with one another. This project builds upon these findings by addressing the need for a cost-effective, fully integrated ERP solution. Unlike fragmented systems identified in the literature, this design proposes a unified framework where academic, financial, and administrative modules interact seamlessly, ensuring real-time data synchronization and holistic institutional oversight.

IV. METHODOLOGY

A. Introduction

The methodology adopted for the development of the "School ERP System" is a structured, systematic approach ensuring that every aspect of the school's requirements is translated into a functional software solution. This project follows the Software Development Life Cycle (SDLC), specifically utilizing the Waterfall Model. This model was selected due to its linear and sequential nature, which is ideal for projects where requirements are well-understood and documented upfront.

The development process is divided into distinct phases: Requirement Analysis, System Design, Implementation (Coding), Testing, and Deployment. Each phase relies on the deliverables of the previous one, ensuring a disciplined and documented workflow.



Shutterstock

B. *Phase 1: Requirement Gathering and Analysis*

The first and most critical phase involved understanding the specific needs of educational institutions. This was achieved through a comprehensive study of existing manual systems and interviewing potential stakeholders (school administrators, accountants, and teachers).

1) Feasibility Study

Before development, a feasibility study was conducted to determine the viability of the project.

- Technical Feasibility: The project requires standard hardware (desktop computers) and software (Java, MySQL) which are readily available and widely supported. The technical complexity is manageable within the current resources.
- Operational Feasibility: The system is designed with a user-friendly GUI to ensure that non-technical staff can operate it with minimal training. It solves the urgent problem of data redundancy.
- Economic Feasibility: The project uses open-source technologies (Java, MySQL), minimizing development costs. The long-term savings in paper and administrative labor justify the initial effort.

2) Functional Requirements

The system is divided into three primary modules based on user roles:

- Administrator Module: Full control over the system. Can add/delete staff, view all reports, manage fee structures, and oversee the entire database.
- Faculty/Staff Module: Limited access. Can mark attendance, upload marks, and view their own timetables and payroll status.
- Student/Parent Module: Read-only access to specific data. Can view attendance records, fee status, and report cards.

C. *Phase 2: System Design*

In this phase, the requirements gathered were converted into a logical structure. This involved designing the database schema and the flow of data across the system.

1) Data Flow Diagrams (DFD)

DFDs were used to visualize how data moves through the ERP system.

- Level 0 DFD (Context Diagram): Represents the entire School ERP as a single process interacting with external entities like Admin, Student, and Teacher.
- Level 1 DFD: Breakdowns the system into major sub-processes: Admission Processing, Fee Management, Attendance Tracking, and Exam Management.

Textual Representation of Data Flow:

- Input: User enters login credentials -> System verifies against Database -> Access Granted.
- Process: Admin inputs Student Details -> System Validates Data -> Database Stores Record.
- Output: Parent requests Fee Receipt -> System Queries Fee Table -> Printer Generates Receipt.

2) Database Design (E-R Diagram)

The backbone of the ERP system is a relational database. We utilized Entity-Relationship (ER) Modeling to define the data structure.

Key Entities Identified:

- Student: (Attributes: Student_ID, Name, DOB, Class, Address, Contact)
- Teacher: (Attributes: Teacher_ID, Name, Subject, Qualification, Salary)
- Fee: (Attributes: Receipt_No, Student_ID, Amount, Date, Status)
- Exam: (Attributes: Exam_ID, Student_ID, Subject_ID, Marks)

Normalization:

The database tables were normalized up to the Third Normal Form (3NF) to eliminate data redundancy. For example, instead of storing the "Class Teacher Name" in every student record, a separate Class table was created and linked via a Foreign Key.

D. *Phase 3: Hardware and Software Configuration*

To ensure the system runs smoothly, specific hardware and software environments were selected. The choice of Java as the core language provides platform independence, while MySQL offers a robust, free relational database solution.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Table 1: Software Requirements

Component	Specification	Justification
Operating System	Windows 10 / 11	widely used in school admin offices.
Front End	Java (Swing/AWT/JavaFX)	Robust GUI creation and OOP features.
Back End	Java (JDBC)	Connectivity between application and database.
Database	MySQL 8.0	Reliable, open-source, and handles large datasets.
IDE	NetBeans / Eclipse / IntelliJ	Efficient code management and debugging tools.
Server	Apache Tomcat (if web-based)	Standard server for Java applications.

Table 2: Hardware Requirements

Component	Minimum Specification	Recommended Specification
Processor	Intel Core i3	Intel Core i5 or higher
RAM	4 GB	8 GB
Hard Disk	250 GB HDD	512 GB SSD (for faster DB access)
Monitor	Standard LED	15" LED for better GUI visibility

E. Phase 4: Implementation (Coding)

This phase involved the actual coding of the system using the Java programming language. The coding process followed the Modular Programming approach, where the system was broken down into smaller, manageable units (classes and methods).

1) Coding Standards Adopted

- Naming Conventions: CamelCase was used for variables (e.g., studentName) and PascalCase for classes (e.g., FeeController).
- Comments: Inline comments were added to explain complex logic, ensuring code maintainability.
- Error Handling: Extensive use of try-catch blocks to handle exceptions (e.g., SQL connection failures, invalid input types) to prevent system crashes.

2) Database Connectivity

The connection between the Java application and the MySQL database was established using JDBC (Java Database Connectivity).

- Driver Loading: Class.forName("com.mysql.cj.jdbc.Driver");
- Connection Establishment: DriverManager.getConnection(url, user, pass);
- Statement Execution: Using PreparedStatement to prevent SQL injection attacks.

F. Phase 5: Testing Strategy

Testing is crucial to ensure the system is bug-free and meets the initial requirements. A combination of testing methodologies was employed.

1) Unit Testing

Individual modules were tested in isolation.

- *Example:* The "Fee Calculation" method was tested with various inputs to ensure it correctly adds late fees and subtracts concessions.

2) Integration Testing

After unit testing, modules were combined to test their interaction.

- *Example:* We tested if deleting a "Student" record also correctly removes their associated "Attendance" and "Fee" records (Referential Integrity).

3) System Testing (Validation)

The complete system was tested against the requirements.

- Login Validation: Checked if invalid credentials are rejected.
- Input Validation: Ensured that numeric fields (like Phone Number) do not accept alphabets.

Table 3: Test Cases (Sample)

Test Case ID	Test Scenario	Input Data	Expected Result	Actual Result	Status
TC_01	Admin Login	User: admin, Pass: 1234	Login Successful	Login Successful	Pass
TC_02	Invalid Fee Entry	Amount: -500	Error Message	Error Message	Pass
TC_03	Add Student	Empty Name Field	Prompt "Name Required"	Prompt Shown	Pass

G. Phase 6: Deployment and Maintenance

1) Deployment

Once testing was successfully completed, the system was packaged into an executable JAR file. The database script (school_db.sql) was exported to be installed on the client's server. The deployment process involved:

- Installing the Java Runtime Environment (JRE) on client machines.
- Setting up the MySQL server and importing the schema.
- Configuring the database connection strings in the application properties.

2) Maintenance

Post-deployment maintenance plans were drafted to ensure longevity:

- Corrective Maintenance: Fixing any bugs reported by users after deployment.
- Adaptive Maintenance: Updating the system if the OS or Database version changes.
- Perfective Maintenance: Adding new features (e.g., SMS integration) in future versions.

H. Conclusion of Methodology

The adherence to the Waterfall SDLC model provided a disciplined roadmap for the development of the School ERP System. By strictly separating the design, coding, and testing phases, we minimized the risk of structural errors and ensured that the final product was robust, scalable, and fully aligned with the operational needs of the school. The use of Java and MySQL ensures that the solution is both cost-effective and technically sound, providing a solid foundation for future enhancements.

V. SYSTEM ARCHITECTURE

The School ERP System is built upon a robust Three-Tier Architecture. This architectural pattern is chosen to decouple the user interface, business logic, and data storage, ensuring that the system is modular, scalable, and easy to maintain. By separating these concerns, modifications in one layer (e.g., changing the interface design) do not disrupt the underlying database logic.

The three specific layers are:

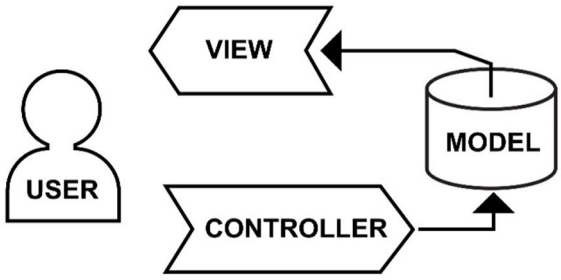
1) Presentation Layer (Client Tier): This is the topmost layer where the user interacts with the system. It consists of the Graphical User Interface (GUI) developed using Java Swing/AWT or web forms. It captures user inputs (like student details or fee amounts) and presents data retrieved from the server in a readable format.

2) Application Layer (Business Logic Tier): Acting as the bridge, this layer processes the logic. It receives requests from the client, performs validations (e.g., checking if a student ID exists), and applies business rules (e.g., calculating late fee penalties). This layer is implemented using Java classes and methods.

3) Data Access Layer (Database Tier): The foundation of the system, this layer hosts the MySQL Database. It is responsible for securely storing, retrieving, and updating all institutional records. The Application layer communicates with this layer via JDBC (Java Database Connectivity).

This architecture ensures high performance and security, as direct access to the database is restricted, and all data must pass through the validation logic of the application layer.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



MODEL - VIEW - CONTROLLER PATTERN

VI. IMPLEMENTATION DETAILS

The implementation phase marks the transition from theoretical design to a functional software product. The School ERP System was developed using the Java programming language for the front-end application logic and MySQL for the back-end database storage. The development environment utilized was Apache NetBeans IDE, which provided robust tools for debugging and GUI creation.

A. Module Implementation

The system was built using a modular approach, ensuring that each functional area could be developed and tested independently before integration.

1) Authentication Module: The entry point of the system is the Login Interface. This module implements security protocols using SQL queries to verify user credentials against the admin_users table. Passwords are encrypted to ensure data security.

2) Student Registration Module: This module utilizes Java Swing components (Text Fields, Radio Buttons, Combo Boxes) to capture detailed student data. Input validation was implemented here to prevent errors, such as ensuring phone numbers contain only ten digits and dates follow the YYYY-MM-DD format.

3) Database Connectivity (JDBC): The core of the implementation is the Java Database Connectivity (JDBC) API. The application communicates with the MySQL server using the mysql-connector-java library. A dedicated DBConnection class was created to manage the connection lifecycle. This class loads the driver (Class.forName) and establishes the connection (DriverManager.getConnection), ensuring that the database is accessible throughout the application without redundant code.

B. Key Challenges and Solutions

During implementation, several challenges were addressed. Handling date formats between Java and MySQL required careful parsing using SimpleDateFormat. Additionally, preventing SQL Injection attacks was a priority; this was achieved by strictly using PreparedStatement interfaces rather than standard Statement interfaces for all database queries.

VII. CONCLUSION

The design and development of the School ERP System successfully address the critical inefficiencies inherent in manual school management. By transitioning from physical file storage to a centralized digital database, the project achieves its primary objectives of data integrity, operational efficiency, and administrative transparency. The system effectively eliminates data redundancy; a single entry in the master database now propagates across all modules, ensuring that the admission, fee, and attendance departments always view consistent data. The implementation of Role-Based Access Control (RBAC) ensures that sensitive data remains secure, accessible only to authorized personnel. Furthermore, the automation of complex tasks, such as fee calculation and report card generation, significantly reduces the manual workload on staff, minimizing human error and saving valuable time.

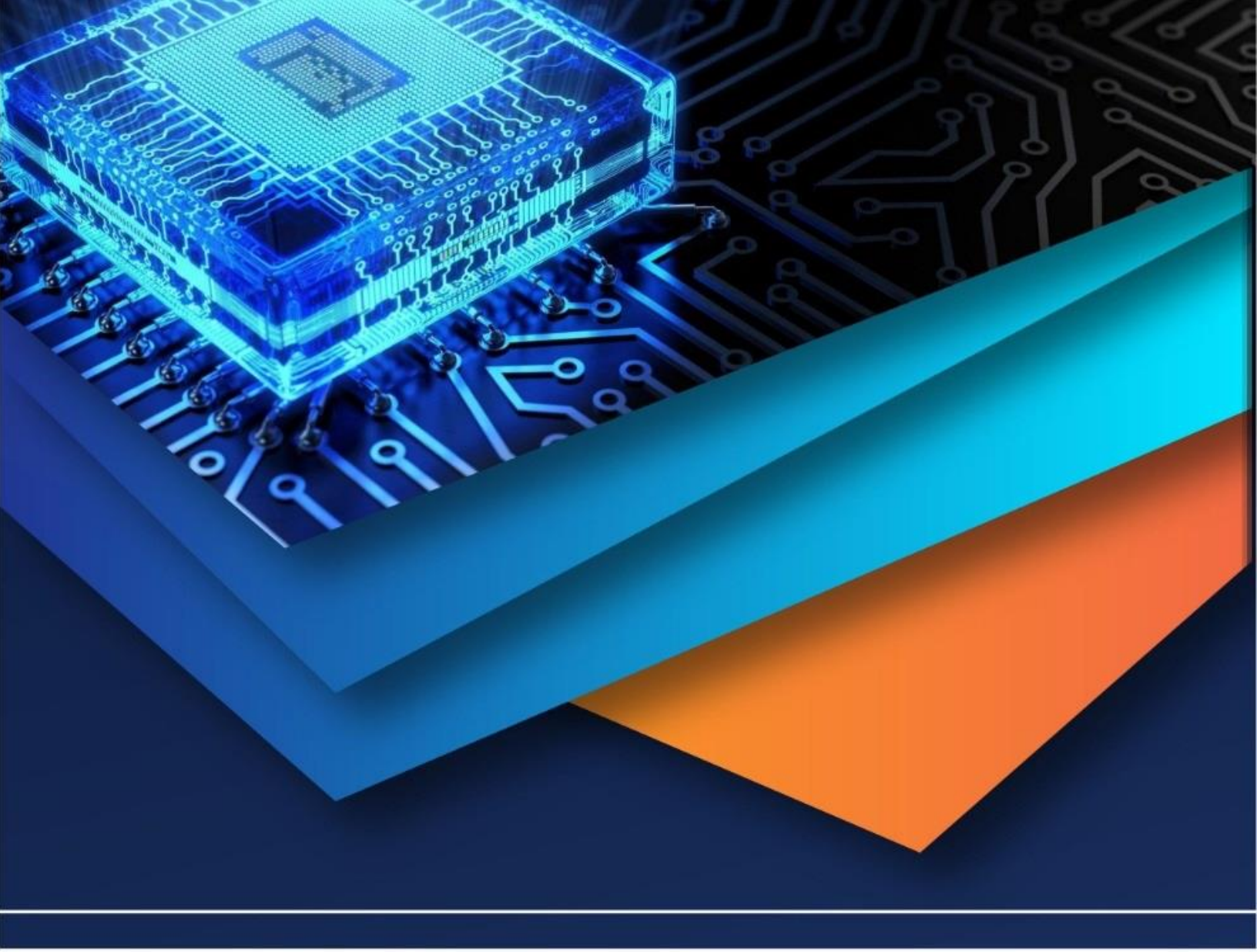
While the current system covers the fundamental aspects of school administration, it lays a solid foundation for future scalability. The modular architecture allows for the easy addition of new features, such as SMS notifications for parents, mobile app integration, or biometric attendance systems. In summary, this project demonstrates that integrating information technology into educational administration is not merely a modernization effort but a strategic necessity that empowers institutions to focus on their core mission: delivering quality education.

©IJRASET: All Rights are Reserved

REFERENCES

[1] ISO/IEC 18004:2015, Information Technology — Automatic Identification and Data Capture Techniques — QR Code Bar Code Symbology Specification, International Organization for Standardization, Geneva, Switzerland, 2015.
[2] S. Rouillard, “Utilizing QR codes for delivering mobile information,” in Proc. IEEE Int. Conf. Wireless Networks, Las Vegas, NV, USA, 2008, pp. 1–6.
[3] J. Svensk and S. McIntyre, “Enhancing medication safety with QR code-based labelling systems,” J. Pharmacy Practice, vol. 34, no. 2, pp. 215–223, 2021.
[4] A. K. Singh, R. Verma, and P. Sharma, “Exploring QR code applications in pharmaceutical supply chains,” Int. J. Pharmaceutical Technology, vol. 12, no. 4, pp. 1021–1028, 2020.
[5] M. Gerrans, L. Patel, and D. Wong, “Adopting digital labelling technologies in healthcare packaging,” IEEE Access, vol. 11, pp. 45678–45689, 2023.
[6] Python Software Foundation, Python 3 Documentation, Wilmington, DE, USA, 2024.
[7] S. Datta and K. Roy, “Machine-readable coding for drug traceability and compliance,” in Proc. IEEE Int. Conf. Healthcare Informatics, Oldenburg, Germany, 2022, pp. 312–318.
[8] R. Want, “A primer on RFID technology,” IEEE Pervasive Computing, vol. 5, no. 1, pp. 25–33, Jan.–Mar. 2006.
[9] A. Kumar and N. Smith, “Barcodes and QR code technologies for healthcare systems,” IEEE Eng. Med. Biol. Mag., vol. 31, no. 4, pp. 45–52, Jul.–Aug. 2012.
[10] World Health Organization, Guidelines on Quality, Safety, and Effectiveness of Pharmaceutical Products, Geneva, Switzerland: WHO Press, 2018.
[11] P. Zhang, Y. Li, and X. Chen, “Assessment of machine-readable labelling approaches for drug packaging,” IEEE Access, vol. 9, pp. 112345–112356, 2021.
[12] M. Rouse and S. Jain, “Reliable and secure QR code-based information encoding,” in Proc. IEEE Int. Conf. Advanced Computing and Communication Systems, Coimbatore, India, 2020, pp. 876–881.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)