ISSN No. 0976-5697

VOLUME 17 SPECIAL ISSUE 1, JULY 2026

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Centralised Digital Platform for Comprehensive student activity Record in HEIs

Ms Ayushi Ghill
Department of Computer Science and Engineering
Geetanjali Institute of Technical Studies(GITS)
Udaipur, India
Ayudhanu17@gmail.com

Pradhum Sharma
Department of Computer Science and Engineering
Geetanjali Institute of TechnicalStudies(GITS)
Udaipur, India
pradhumsharma59@gmail.com

Pawandeep Kour
Department of Computer Science and Engineering
Geetanjali Institute of TechnicalStudies(GITS)
Udaipur,India
pawandeepkour011@gmail.com

Prerak Ameta
Department of Computer Science and Engineering
Geetanjali Institute of TechnicalStudies(GITS)
Udaipur,India
prerakkc0611@gmail.com

***Abstract*:** Higher Educational Institutions (HEIs) currently face significant challenges in maintaining accurate, transparent, and verifiable student information. Traditional methods of record-keeping rely heavily on manual data entry, paper-based certificates, and unstructured data handling, which frequently results in inefficiency, data fragmentation, and a lack of real-time accessibility. This paper presents the design, development, and evaluation of a Centralized Digital Platform designed to automate and unify student activity records. The proposed system utilizes a three-tier architecture to provide a seamless ecosystem supporting role-based dashboards for students, faculty, administrators, and recruiters. By integrating features such as automated attendance, QR-based certificate verification, and centralized demographic reporting, the system significantly improves transparency and secure data management. Experimental results from pilot testing demonstrate a reduction in administrative task time by up to 50% and an improvement in data accuracy from 40% to over 95%. This research highlights how centralized, technology-driven solutions can modernize academic management, align with outcome-based education standards, and establish a foundation for scalable digital transformation in HEIs.

***Keywords*:** Centralized Digital Platform; Higher Education Institutions; Student Activity Records; Digital Transformation; Three-Tier Architecture; Role-Based Access Control.

I. INTRODUCTION

The effective management of student data and administrative processes is fundamental to the successful functioning of modern Higher Education Institutions (HEIs). Historically, traditional paper-based systems were the standard for handling student records, class schedules, and parental communications. However, these methods are no longer sufficient to meet the demands of rapid institutional growth and modern educational frameworks. Outdated manual systems inherently lead to fragmented data, redundant processes, and a high risk of data loss, making it incredibly difficult to manage tasks like student enrollment, grading, and tracking extracurricular achievements.

As institutions transition toward Outcome-Based Education (OBE) and adhere to strict accreditation frameworks (such as NAAC and NBA in India), the need for comprehensive and reliable digital records has become critical. The absence of centralized visibility creates bottlenecks for administrators generating reports, faculty tracking attendance, and recruiters attempting to verify student achievements during campus placements.

To address these systemic inefficiencies, this research proposes a Centralized Digital Platform for Comprehensive Student Activity Records. This system is engineered to automate administrative tasks and centralize the management of both academic and extracurricular data. By digitizing data workflows, the platform aims to reduce human error, streamline communication between stakeholders, and provide real-time, data-driven insights. This paper details the architectural design, implementation methodology, and experimental results of deploying such a system in a real-world educational setting.

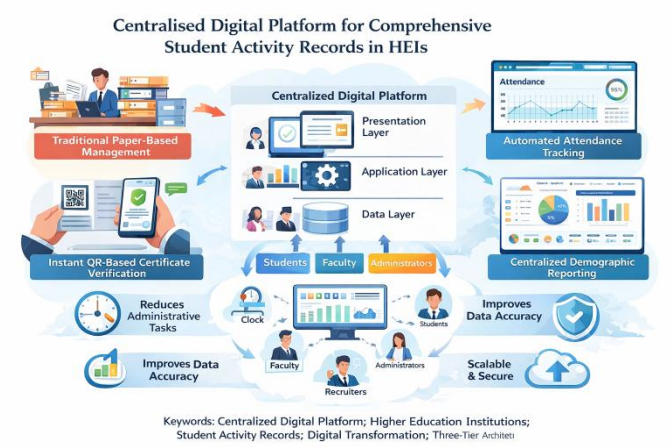

Figure 1. Activity Records in HEIs

Figure 2. Various Subject Activity of Digital Platform

II. LITERATURE REVIEW

A. Traditional Record Management and Inefficiencies

Research into traditional paper-based student record systems reveals significant vulnerabilities regarding data integrity and administrative efficiency. Baseline studies indicate that manual systems often suffer from severe discrepancies; in some cases, up to 60% of student records contain inaccuracies due to mismatched documentation. Furthermore, physical records are highly susceptible to damage or misplacement, with audits showing that approximately 30% of historical student records in traditional setups are lost or irretrievable over time.

B. Digital Solutions in Education

The integration of Information Systems in education has proven pivotal in enhancing data accessibility and workflow management. Studies show that digital Student Information Systems (SIS) play a vital role in improving stakeholder communication and administrative workflows. While commercial platforms like OpenSIS and PowerSchool offer comprehensive features such as attendance tracking and gradebook management, they often present financial and customization barriers for smaller to medium-sized institutions. Consequently, many resource-constrained schools resort to basic tools like shared Excel spreadsheets, which lack scalability, automated processes, and robust security.

C. Advanced Integration and Authentication

Recent advancements have proposed integrating Artificial Intelligence (AI) and secure digital verification into student management. Applications incorporating tools like face-recognition for attendance and QR-code generation for tamper-proof certificates have been conceptualized to reduce manual dependencies. Furthermore, AI-driven chatbots are increasingly utilized to handle routine queries regarding academic schedules and placement statistics, significantly reducing the administrative burden on faculty. However, gaps remain in creating a single, interoperable platform that combines these advanced features into a cohesive, user-friendly ecosystem for HEIs.

III. METHODOLOGY

A. System Architecture

The proposed Centralized Digital Platform is built upon a robust three-tier architecture model. This design separates the system into distinct layers for user interaction, business logic, and data management, ensuring high scalability, maintainability, and ease of updates.

- **Presentation Layer (UI/UX Layer):** This tier serves as the direct point of interaction for users (Students, Faculty, Administrators, Recruiters). It is developed using responsive web technologies (HTML5, CSS3, JavaScript, and frameworks like Bootstrap) to ensure accessibility across both desktop and mobile devices. It is responsible for displaying intuitive dashboards, interactive charts, and data input forms.
- **Application Layer (Business Logic Layer):** Serving as the core of the system, this layer processes all business rules. It handles user authentication, role-based routing, record manipulation (CRUD operations), and report generation. Depending on institutional preference, this layer utilizes server-side technologies such as Node.js or Python (Flask) alongside secure middleware.
- **Data Layer (Database Layer):** This layer is dedicated to the secure storage and rapid retrieval of all institutional data, including user credentials, student demographics, and activity logs. A relational database like MySQL or a NoSQL database like MongoDB is utilized to ensure data integrity and flexible schema modeling.

B. Development Lifecycle

The system development followed the structured Waterfall Model of the Software Development Life Cycle (SDLC).

- **Requirements Gathering:** Data was collected via stakeholder interviews and surveys to identify user needs and system specifications.

- **System Design:** System architecture, database schemas, and wireframes were fully mapped out before coding commenced.
- **System Development:** The actual coding of frontend interfaces and backend APIs was executed.
- **Testing and Quality Assurance:** Comprehensive unit, integration, and system testing were conducted to identify and resolve bugs, ensuring the software met functional requirements.
- **Deployment and Maintenance:** The system was launched on a live server, allowing end-users to interact with the platform and provide feedback for continuous improvement.

C. Core Modular Design

To ensure comprehensive management of student activities, the system is divided into specific functional modules:

- **User Authentication and Authorization:** Implements Role-Based Access Control (RBAC) to differentiate permissions between Administrators, Faculty, Students, and Recruiters. It utilizes secure token-based authentication (e.g., JWT) and password encryption (e.g., bcrypt or AES).
- **Student Record Management:** Allows authorized faculty to add, update, and manage student demographic and academic data. Students can upload their extracurricular certificates and course forms directly to the platform.
- **Verification and Security:** Certificates uploaded by students trigger a digital verification request to the faculty. Upon approval, the system generates a tamper-proof QR code, allowing recruiters to instantly validate the authenticity of the document during placements.
- **Analytics and Reporting:** Administrators are provided with automated dashboards generating visual reports (charts and graphs) on class demographics, enrollment tracking, and overall performance trends.

IV. IMPLEMENTATION

The implementation phase translates the theoretical system design into a functional, highly responsive web application with robust human-computer interaction. This phase encompasses the setup of the development environment, the coding of the technology stack, and rigorous testing protocols to ensure system reliability.

A. Hardware and Software Requirements

For the effective development and operation of the Centralized Digital Platform, specific minimum system requirements were established.

- **Hardware:** The system requires an Intel-based computer with at least 4GB of RAM, a minimum of 100GB hard disk space, and an uninterrupted power supply (UPS) to prevent data corruption during development.
- **Software:** Development was conducted on a Windows Operating System (Windows 7 or higher) for fast processing. Visual Studio Code was utilized as the primary Integrated Development Environment (IDE) due to its powerful built-in tools, Git support, and real-time syntax error detection. Postman was used for testing HTTP GET and POST methods, and Google Chrome was the preferred web browser for interface testing.

B. Technology Stack

The platform was engineered using a modern, scalable technology stack tailored to handle complex student data and multi-stakeholder access.

- **Frontend (Presentation Layer):** The user interfaces were built using HTML5, CSS3, and JavaScript. To ensure a mobile-first, highly responsive design across various devices (desktops, tablets, and smartphones), the Bootstrap framework was heavily utilized. Interactive data visualization for administrative dashboards was powered by libraries like Chart.js.
- **Backend (Application Layer):** The server-side logic and RESTful APIs were developed using Python and the Flask web framework. Flask, a micro-framework based on the Werkzeug WSGI toolkit and Jinja2 templating engine, was chosen for its simplicity, extensibility, and efficient handling of routing and HTTP requests. (Note: The architecture is also adaptable to Node.js and Express.js for environments requiring highly asynchronous processing).
- **Database (Data Layer):** MongoDB, a NoSQL document-oriented database, was deployed to manage the institution's data. Unlike traditional relational databases, MongoDB stores information in flexible, JSON-like documents, making it highly scalable and capable of handling rapid changes to workloads and large volumes of unstructured data (such as extracurricular certificates and dynamic student profiles).

C. Implementation of Core Modules

- **Authentication and Security:** The system implements strict Role-Based Access Control (RBAC). Passwords and sensitive user data are encrypted using secure hashing algorithms (like bcrypt) before being stored in the database, ensuring high data security.
- **Automation Features:** For modules requiring advanced validation, algorithms convert uploaded images into face embeddings for automated attendance tracking using libraries like face-api.js. Additionally, SHA-based hash generation is implemented to create tamper-proof QR codes upon the approval of student certificates by faculty.

D. System Testing

Testing is a critical part of the software development lifecycle, intended to identify flaws, remove coding errors, and ensure the system performs according to specifications. The system was subjected to the following testing phases:

1. **Unit Testing:** The first level of testing involved isolating individual functions, methods, and components (such as the login and search modules) to verify their correctness during the coding phase.
2. **Integration Testing:** Different modules were combined and tested as a unified entity. This phase verified that individual units communicated with each other properly and exposed any defects in data exchange between the frontend interfaces and the backend database.
3. **System Testing:** Conducted on the fully integrated platform, this test evaluated the application's overall compliance with the functional and technical requirement specifications. Exhaustive test data was used to ensure the system handled edge cases and errors correctly.
4. **Acceptance Testing:** As a final step before production deployment, end-users (students and faculty representatives) evaluated the application in a real-world scenario to confirm that business flows were seamless and the software met all their initial expectations.

Figure 3. Requirements and Implementation

V. RESULTS

The system was deployed and evaluated in a real-world educational setting through a comprehensive pilot program.

A. Improved Data Accuracy and Integrity

Prior to the implementation of the centralized system, baseline audits revealed that 60% of student records contained discrepancies, and 30% of historical data was lost due to paper-based vulnerabilities. Post-implementation results demonstrated a dramatic improvement. The introduction of standardized data entry, automated validation, and real-time updates increased data accuracy to over 95%. Furthermore, the transition to secure, centralized digital storage with automated backups reduced the risk of record loss to less than 5%.

B. Enhancement of Administrative Efficiency

The automation of routine administrative workflows yielded significant time savings. Processes that previously required hours of manual labor, such as enrollment management, attendance tracking, and report generation, were optimized. The system reduced the overall time required for administrative tasks by up to 50%. Specifically, in the context of certificate validation, faculty verification time was reduced by 90% due to the digitized approval workflow.

C. Stakeholder Usability and Verification Speed

The platform significantly improved the experience for external stakeholders, particularly recruiters. The implementation of QR-based authentication completely eliminated the need for manual cross-checking of physical portfolios. Recruiters were able to successfully scan and verify student documents in less than 5 seconds. Additionally, the integration of automated querying tools (like chatbots) successfully handled hundreds of routine queries, drastically reducing faculty dependency for basic information retrieval.

VI. DISCUSSION

A. System Strengths

The experimental results underscore the transformative potential of the Centralized Digital Platform. The most prominent strength of the system is the establishment of a "single source of truth" for student data. By eliminating data silos, the platform ensures that administrators, faculty, and students are all interacting with synchronized, up-to-date information. The generation of verified digital identity profiles serves as a comprehensive academic passport, deeply enhancing student employability and recruiter trust. Furthermore, the platform's cost-effectiveness and customization capabilities make it highly suitable for HEIs operating with limited technical resources or budgets.

B. Limitations and Challenges

Despite its successes, the deployment of such a comprehensive digital ecosystem presents certain challenges. A primary limitation is network dependency; real-time verification and dashboard updates require consistent internet connectivity, which can be an obstacle in regions with poor infrastructure. Additionally, transitioning staff from decades-old manual systems to a digital platform often meets with initial resistance and requires significant investment in user training and change management. System security also remains an ongoing concern; while encryption and RBAC are implemented, centralized databases are lucrative targets for cyber threats, necessitating rigorous and continuous security audits.

C. Real-World Deployment Considerations

For widespread deployment, HEIs must ensure that the platform aligns with national digital frameworks. In India, for instance, aligning the platform with the National Education Policy (NEP) 2020 and initiatives like Digital India promotes evidence-based, data-driven educational environments. Deploying the system on scalable cloud architecture ensures

24/7 availability and allows the platform to handle rapid increases in student enrollment without degrading performance.

VII. CONCLUSION AND FUTURE SCOPE

The Centralized Digital Platform for Comprehensive Student Activity Records successfully addresses the core administrative and data management challenges faced by modern Higher Education Institutions. By transitioning from fragmented, paper-based methods to a secure, three-tier web application, the system drastically reduces human error, prevents data loss, and eliminates administrative redundancies. The integration of automated workflows, role-based dashboards, and instant QR verification not only empowers educators to make data-driven decisions but also bridges the trust gap between academia and industry recruiters.

Looking ahead, there are several promising avenues for future research and enhancement. First, integrating the platform with national academic repositories (such as DigiLocker) could facilitate nationwide, seamless academic verification. Second, embedding advanced predictive analytics and machine learning models could allow institutions to forecast student performance, identify at-risk students, and deploy proactive academic interventions. Finally, expanding the system into a dedicated mobile application would further enhance accessibility, allowing students and parents to monitor academic progress and institutional communications in real-time, from anywhere.

VIII. ACKNOWLEDGMENT

I would like to express my sincere gratitude to the faculty and administration of my institution for their continuous guidance and support throughout this research project. Special thanks to the IT department for providing the necessary infrastructure to conceptualize and evaluate the proposed system architecture.

IX. REFERENCE

[1] T. Anderson and J. Lee, "Challenges in traditional record management systems in education," *Journal of Educational Technology*, vol. 41, no. 2, pp. 110-124, 2019.

[2] Ministry of Education, Government of India, "National Education Policy (NEP) 2020," Policy Report, 2020.

[3] L. Brown and D. Wilson, "Challenges in implementing SRS for small institutions," *Journal of Educational Technology*, vol. 21, no. 4, pp. 234-245, 2019.

[4] J. J. Garrett, *The elements of user experience: User-centered design for the web*, 2nd ed. Pearson Education, 2010. [5] S. K. Singh and P. Kumar, "AI-Driven Automation in Higher Education: A Systematic Review," *International Journal of Educational Technology*, vol. 14, no. 3, pp. 112-128, 2022.

[5] M. Tanwar and R. Sharma, "Blockchain-Enabled Academic Certificate Verification," *Journal of Information Security and Applications*, vol. 63, pp. 1-10, 2021.

[6] W. W. Royce, "Managing the development of large software systems," *Proceedings of IEEE WESCON*, pp. 1-9, 1970.