ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

A Scalable Web-Based Attendance Management System with Integrated Analytics for Academic Decision Support

[1]V. Deepak Jaya Ram Yadav, [2]V. Madhu Sudhan, [3]V. Harsha Vardhan, [4]Y. Naveen, [5]B. Venkata Satya Umesh
[1,2,3,4,5]B. Tech CSE (AIDS) Students, Department of CSE, Sir C R Reddy College of Engineering, Eluru.

ABSTRACT

In educational institutions, effective attendance management is essential for tracking student involvement and assisting with academic decision-making. Conventional manual attendance tracking techniques are frequently laborious, prone to mistakes, and unable to yield insightful data. The design and implementation of a scalable web-based attendance management system with analytical capabilities integrated to improve accuracy and efficiency are presented in this work. The suggested system is built using a multi-tier design, in which a relational database guarantees structured data storage, the backend manages business logic using a lightweight web framework, and the frontend interface facilitates user interaction. Through a user-friendly interface, the system enables instructors to record attendance in real time while guaranteeing secure access through authentication procedures. In order to support data-driven academic decisions, it also includes analytics modules that calculate attendance percentages, spot absenteeism trends, and produce visual reports. By integrating visualization tools, users can properly interpret attendance trends and provide at-risk pupils with timely interventions. The technology greatly decreases administrative workload, increases data accuracy, and improves accessibility across many platforms, according to experimental study. Easy deployment and effective performance under normal usage situations are guaranteed by the lightweight design. The system offers a solid basis for upcoming improvements like predictive modelling and interaction with cutting-edge technology, even though its current concentration is on descriptive analytics. All things considered, the suggested solution provides a dependable, effective, and scalable method for updating attendance tracking in educational settings.

Keywords: Structural Equation Modelling, Random Forest, Classifier and Accuracy.

I. INTRODUCTION

The study uses a quantitative method based on structural equation modeling (SEM) to investigate how predictive analytics affects attendance systems. Predictive analytics greatly increases staff engagement and technological readiness, which in turn increases regular attendance and overall productivity, according to data gathered through structured questionnaires. The results demonstrate how data-driven strategies can enhance system performance and accountability. However, the study's limitations—such as its small sample size, reliance on self-reported data, and lack of validation in a variety of real-world settings—indicate the need for more research using larger datasets and cutting-edge technologies [1]. Using a Random Forest classifier trained on attributes like attendance, prior grades, study hours, and assignment scores, the study suggests a machine learning-based system for forecasting students' academic achievement. By classifying pupils into performance levels and achieving high accuracy (99%), the algorithm makes it possible to identify at-risk students early on. The results demonstrate the usefulness of predictive analytics in education by highlighting attendance and past academic records as important indicators of success. However, the study is constrained by the possibility of overfitting because of its high accuracy, reliance on the quality of the dataset, and lack of validation in a variety of educational settings, suggesting the need for more extensive testing and practical application [2].

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

The study uses a qualitative methodology that includes focus groups with school professionals and student interviews to examine school attendance issues (SAPs) among kids with social, emotional, and behavioral challenges (SEBD). The results show that effective measures for increasing attendance include trust-based relationships, well-coordinated support systems, flexible learning modalities, and clear communication. It also highlights different viewpoints, with students emphasizing emotional support and decision-making participation while professionals concentrate on procedural solutions. However, the study's qualitative design, small sample size, and concentration on particular educational environments may limit its applicability in more general scenarios [3].

The paper suggests an AI-driven architecture that uses machine learning models like Gradient Boosting (GB) and Gradient Boosting Machine (GBM) to create individualized competency recommendations based on academic achievement and attendance habits. Large-scale educational data from several colleges is analyzed as part of the methodology to assess model performance;

GB and GBM perform better than other approaches in terms of accuracy and predictive power.
The results show how important attendance is in determining competency-based learning pathways and how well academic and behavioral data may be integrated for individualized instruction. Nevertheless, the study is constrained by its reliance on sizable and organized datasets, possible difficulties in modifying the model for other educational systems, and the requirement for additional validation in a variety of real-world contexts [4].

By examining academic records, attendance information, and behavioral tendencies, the study suggests a hybrid machine learning framework that blends supervised and unsupervised methods to improve student management. The approach consists of adaptive mechanisms for tailored interventions, predictive modeling for academic success, and clustering for student segmentation. Compared to conventional methods, the results show increased precision and dependability in identifying kids who are at risk and assisting with data-driven decision-making. Nevertheless, the method's limitations include its dependence on thorough and high-quality datasets, possible implementation complexity, and difficulties using the model in various educational contexts [5].

II. LITERATURE SURVEY

Shrivastava et al., 2023 [6] presents an IoT-based RFID attendance monitoring system that enhances the efficiency and accuracy of attendance management compared to conventional manual methods. The methodology involves integrating RFID technology with IoT using hardware components such as an RFID reader and an Arduino ESP8266 microcontroller. Each student is assigned a unique RFID tag, which is scanned to automatically record attendance. The collected data is processed by the microcontroller and transmitted to a cloud-based platform (Adafruit.io) for real-time storage and monitoring. The system operates within a predefined area to ensure that attendance is marked only when the student is physically present. The key findings indicate that this automated approach significantly reduces time consumption, minimizes human errors, and prevents fraudulent practices such as proxy attendance to a certain extent. Furthermore, the system enables real-time tracking, accurate calculation of attendance percentages, and efficient report generation, thereby improving overall productivity and reliability in educational institutions.

Arulogun et al., 2013 [7] presents an RFID-based student attendance management system designed to address inefficiencies in manual attendance tracking, particularly in developing countries. The methodology involves the use of RFID technology, where students are assigned electronic tags (passive or active) that are detected by RFID readers to automatically record attendance during lectures. This wireless identification system enables fast and accurate data capture, which can be stored and utilized for analyzing classroom attendance patterns. The key findings indicate that the system significantly reduces time consumption associated with

manual attendance, improves accuracy, and provides valuable data for academic administration and decision-making. However, the system has certain limitations, including dependency on RFID hardware infrastructure, potential issues such as tag misplacement or misuse, and the lack of advanced authentication mechanisms to completely prevent proxy attendance. Additionally, scalability and implementation costs may pose challenges in large institutions.

Paul et al., 2025 [8] proposes a real-time attendance system that integrates face recognition and emotion detection using advanced machine learning techniques to enhance traditional attendance mechanisms. The methodology is based on a dual-path architecture, where ResNet-50 is employed for accurate facial recognition and Vision Transformer (ViT) is used for emotion detection, enabling simultaneous identity verification and emotional state analysis. A custom dataset is developed to support real-time performance requirements, and the system is integrated with a web-based platform to improve accessibility and scalability across various domains such as education, corporate environments, and healthcare. The key findings demonstrate high performance, achieving an accuracy of 0.87 for face recognition and 0.90 for emotion detection, along with an AUC-ROC score of 0.96, indicating strong reliability and precision. However, the system has certain limitations, including high computational complexity due to deep learning models, dependency on large and diverse datasets for optimal accuracy, and potential privacy concerns related to biometric data usage. Additionally, real-time performance may be affected in resource-constrained environments, and environmental factors such as lighting and facial occlusions can impact system accuracy. Ţălu, 2025 [9] presents a comprehensive review of Internet of Things (IoT)-based smart management systems in university environments, focusing on their potential to enhance administrative efficiency, improve student engagement, and modernize educational practices. The methodology follows a structured and systematic literature review approach, where relevant studies are selected from reputable academic databases based on predefined inclusion criteria. The review analyzes IoT applications across multiple domains such as smart classrooms, energy management, personalized learning, and attendance systems, incorporating insights from global case studies, including implementations at the Technical University of Cluj-Napoca. The key findings indicate that IoT integration significantly improves operational efficiency, enables datadriven decision-making, and enhances interactive learning experiences. However, the study also identifies several limitations, including concerns related to data privacy and security, challenges in system interoperability across different platforms, and high implementation and maintenance costs. These factors pose barriers to large-scale adoption and highlight the need for standardized frameworks and robust security mechanisms in IoT-based educational systems.

Butt et al., 2025 [10] proposes a hybrid framework that integrates machine learning (ML) techniques with rough set theory to enhance student management by identifying at-risk students and enabling personalized interventions. The methodology combines classification algorithms with rough set-based decision rules to analyze complex educational data, including academic performance, behavioral patterns, and student engagement levels. The system is evaluated using the Open University Learning Analytics Dataset (OULAD), where the ML-based layered approach detects patterns, anomalies, and risk indicators to support data-driven decision-making. The key findings demonstrate high performance, achieving an accuracy of 97.85% in predicting student outcomes and a precision of 94.62% in identifying students requiring support, outperforming traditional methods by approximately 15%. However, the framework has certain limitations, including dependency on high-quality and well-structured datasets, potential challenges in handling missing or inconsistent data, and increased computational complexity due to the hybrid model design. Additionally, the generalizability of the model may be limited across different educational contexts without proper adaptation and tuning.

Ahmed et al., 2025 [11] develops a Gamified Mobile Cloud-based Learning Management System (GMCLMS) to enhance student engagement and academic performance, particularly in an ObjectOriented Programming course. The methodology adopts a mixed approach, combining system development with a

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

quasi-experimental design to compare the performance of students using GMCLMS with those using traditional LMS platforms. The findings indicate that the gamified system significantly improves student engagement and learning outcomes, with higher performance observed in the experimental group, demonstrating the effectiveness of integrating gamification and cloud technologies in education. However, the study is limited by its focus on a specific course and sample group, potential implementation costs, and the need for broader validation across diverse subjects and educational environments.

Swiderski et al., 2025 [12] analyzes student attendance patterns in North Carolina by comparing pre-pandemic and post-pandemic data over multiple years to assess changes in absenteeism trends. The methodology involves longitudinal data analysis of student-level attendance records across different time periods and demographic groups. The findings reveal a significant increase in chronic absenteeism post-pandemic, with rates rising from 17% to 38%, and persistent absenteeism disproportionately affecting minority and high-poverty students. However, the study is limited by its focus on a specific geographic region, potential external socio-economic influences not fully captured, and limited exploration of underlying causal factors, indicating the need for broader and more comprehensive studies.

III. METHODOLOGY

A. System Architecture and Design Approach

The display layer, application layer, and data layer make up the multi-tier architecture used in the design of the suggested Web-Based Attendance Management System. A web-based user interface created using HTML, CSS, and JavaScript that is accessible on a variety of devices makes up the presentation layer. The Python Flask micro-framework, which manages routing, business logic, and request processing, is used to construct the application layer. A SQLite database that controls the persistent storage of student and attendance records supports the data layer. In addition to facilitating effective communication between system components, this division of responsibilities improves system scalability, maintainability, and security.

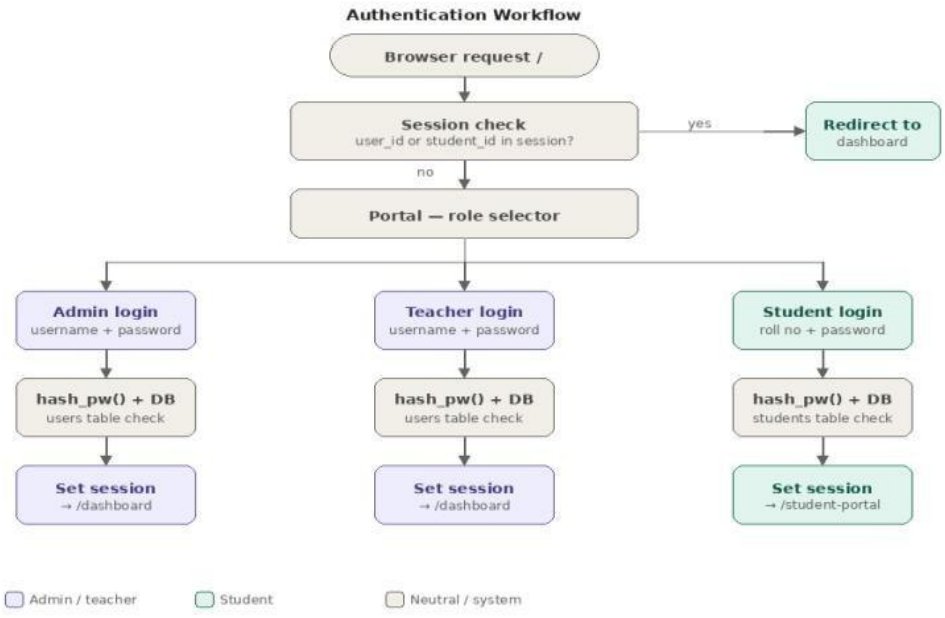


Fig 1. Arciterure

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

B. Development Methodology and Implementation Strategy

The system is created via a development approach based on the principles of an Agile methodology, which is modular and iterative. All functional components, from analytics and reporting to attendance and authentication, are developed separately and then added to the main application. RESTful endpoints supporting both GET and POST requests are defined using the methods provided by Flask routing. The use of lightweight frameworks and iterative testing during the entire process ensures fast development, flexible adjustments in the design, and successful debugging.

The principles of Agile software engineering were the key drivers behind the implementation of the modular and iterative development approach applied to design the Attendance Management System (AMS). The approach allowed meeting the evolving requirements of academic attendance systems through steady development increments, prompt response to stakeholders' feedback, and careful risk management throughout the entire project cycle.

Three key principles underpinned the development philosophy as a whole: (i) functional decomposition through modular architecture; (ii) integration iterations with progressive validation; and (iii) lightweight tools choice to ensure minimal environment dependency. All functional subsystems of the AMS, including authentication, students' management, attendance management, analytics, and reporting, were first developed as independent modules before being integrated into the main Flask application.

Table 1. Sprint Plan and Deliverables

Sprint	Focus Area	Key Deliverables	Status
1	Foundation & Auth	DB schema, init_db(), hash_pw(), login routes, session management	Complete
2	Core CRUD	Student add/delete, /mark attendance, /student detail, staff decorators	Complete
3	Analytics & Reporting	Dashboard aggregates, attendance log, eligibility report, threshold slider	Complete
4	Student Portal & Hardening	Student self-service portal, password change, admin user management, UI polish	Complete

C. Database Design and Data Management

A relational schema designed using SQLite forms the basis of the database design. With a special initialization routine, the application automatically initializes the database by creating any required tables such as 'students' and 'attendance'. Each attendance entry consists of details such as date and attendance along with a link to a particular student. To efficiently process the data, SQL is used to insert, query, and aggregate the data. Using this approach, reliable data storage, consistent data, and fast access to the attendance data for further processing are achieved.

During initialization of the application at startup, before handling any requests, the database is initialized using the init_db() function. This initialization ensures idempotent execution over multiple applications by making use of the CREATE TABLE IF NOT EXISTS guard on each of the relations. The implementation of schema evolution uses conditional ALTER TABLE statements where the current state of the column manifests itself with PRAGMA table_info() prior to any structure modification. This ensures zero-downtime schema evolution since no manual migrations are required and also the data is not needed to be exported or imported again for new releases.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

To guarantee that the app starts working without the requirement for any database initialization, two default accounts are created during its first run: administrator and class teacher, having their passwords saved as SHA-256 hashes.

All operations with the database are performed through parameterized queries using the sqlite3 package from the Python standard library. It automatically guards against all sorts of SQL injection attacks.

There are three primary categories of database actions that can be identified:

- Write actions involve executing UPDATE queries for changing the password, and INSERT queries for storing attendance incidents, and adding users and students.
- Reading actions involve executing JOIN queries for joining the student and attendance tables for detailed views and SELECT queries with WHERE clauses for fetching records.
- Aggregation actions involve scalar COUNT queries with filtering by date for generating dashboard statistics, as well as GROUP BY queries with conditional SUM queries (CASE WHEN status='Present' THEN 1 ELSE 0 END).

Table 2. Database Schema Summary

Relation	Primary Key	Key Attributes	Constraints
users	id (INTEGER)	username, password, role, full_name	Username UNIQUE NOT NULL
students	id (INTEGER)	name, roll_n dept, student_ password	roll_ no UNIQUE; student_ username UNIQUE
attendance	id (INTEGER)	student_id, date, status	FK student_ id → students(id)

D. Attendance Recording and Processing Mechanism

Using the attendance recording module, teachers can record attendance of each student for each session digitally. To make sure that the data entered is accurate and up-to-date, the attendance status like whether the student is present or not will be recorded along with time stamps. As far as the storage of this attendance status data, the data will be processed by backend logic and stored in the database while the Flask routes will be responsible for the updating and submitting process for attendance data.

During insertion and retrieving, all the dates are validated and formatted in ISO-8601 format (YYYY-MM-DD). Date strings were stored using Python's datetime in the /student/ and /studentportal pages of the student detail form. To convert the unformatted database dates in order to show the attendance history table entries in readable format, the functions date. From isoform at() and strftime() were utilized (for example Monday, April 14, 2025). The exception handling of date formatting loop ensures that corrupt legacy records will not crash the application but rather provide a smooth failover.

The staff will have the ability to see an attendance matrix in its entirety through the use of the /attendance-log endpoint, where all students enrolled will appear as rows while all dates documented will show up as columns. For the creation of such a matrix, a lookup table is created by performing a full-table scan on the attendance relation, where tuples containing (date, student_id) are mapped against their respective statuses. The structure ensures O(1) lookup for individual cells, helping avoid cell-by-cell queries on the database to ensure efficient processing times with growing datasets.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

E. Analytics and Decision Support Mechanism

The proposed solution includes an analytic component used for transforming unstructured data of student attendance into structured useful information. Threshold-based approach is employed to classify the students as safe, warning, and critical depending on the computation of attendance percentage via SQL aggregation function.

Student classification involves the use of the calculated attendance percentage in relation to the predefined threshold percentage value T. For instance, if a student is above T percentage attendance, then he is classified under the safe category while those who are below T but above T minus fifteen are put under warning.

The default value for T is seventy-five percent, but the user can adjust it via the range slider provided in the report interface. The three-stage approach is realized by the eligibility reporting module. Below are the classification criteria:

- Safe category – $P \geq T$ (qualified for the exam).
- Warning category: $(T - 15) \leq P < T$ (needs advice).
- Critical category: $P < (T - 15)$ (not eligible for test).

In order to make sure that the computation of the report does not involve any recalculation of prestored data, yet still ensures an accurate reflection of the selected policy, the thresholds are calculated dynamically according to the threshold values.

The ordering of students from least to most vulnerable is ensured within each band through sorting of students according to attendance %.Attendance data is transformed into numerical figures by the analytics component in order to allow evidence-based decisions on an academic level. Pattendance, formally defined as follows, serves as the basic indicator: P attendance is equal to (Npresent/Ntotal × 100) where Npresent stands for the number of sessions that were marked as "present" and Ntotal stands for the total number of sessions that were recorded for a particular student. The computation in the eligibility report is performed using SQL queries for aggregation purposes. Meanwhile, the student detail and portal views render percentages to the first digit using Python's round().

Table 3. Analytics Metrics and Computation Sources

Metric	Computation Layer	Data Source
Attendance percentage (P)	Application (Python)	attendance WHERE student_id = ?
Eligibility classification	Application (Python)	Derived from P and threshold T
Classes required (C^R)	Application (Python)	Closed-form formula on N_p and N_t
Dashboard totals	Database (SQL COUNT)	attendance GROUP BY status
Today's summary	Database (SQL COUNT)	attendance WHERE date = today
Per-student breakdown	Database (SQL GROUP BY)	students LEFT JOIN attendance

F. Reporting and Visualization Techniques

With an attendance reporting tool embedded in the system, structured attendance records are generated based on either a daily, monthly, or customizable time frame. Graphs and charts that aid in representing attendance patterns and distributions visually have been integrated through external libraries in order to accurately represent the attendance. These tools not only help the faculty/administrators in making informed decisions but also enable better data interpretation by users.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

The Chart.js, an open-source JavaScript charting library, which can be referenced using the content distribution network (CDN), is employed to create visual charts on the client-side. The various aspects of the attendance data are depicted using three different chart types:

- Total Attendance doughnut chart that has a slice of 68% and an animated entrance effect lasting 900 milliseconds, represents the total attendance to absence ratio for all the students and sessions combined.
- Current Day Attendance doughnut chart enables teachers to easily assess their performance in the current day's session by replicating the Total Attendance doughnut chart but within the context of the current day's calendar.
- Individual Student Attendance Grouped Bar Chart: It offers a clear comparison of individual attendance trends by depicting bars for each student registered on the portal, where the number of attendances is represented by green (#059669) and absences by red (#dc2626) colors.

Eligibility reports also serve as immediate visual cues in connection to each student's percentage concerning the threshold line, through the use of an additional visual aid in the form of progress bars embedded inside the table of classification, created using only CSS styles. Users may assess sensitivity analysis for eligibility rules without reloading the page, utilizing an interactive threshold slider input field (HTML range input, min-max value range 50-90%, steps of 5%), which updates the URL report parameters and triggers a re-classification process performed on the server side.

User Interface design is accomplished with responsive CSS grid layout, incorporating the DM Sans font from Google Fonts and CSS variables for color theme. All status colors used (for safe, warning, and critical statuses) have a contrast ratio that complies with the minimum WCAG 2.1 AA accessibility guidelines. To ensure a smooth transition effect in case of page updates, card elements have been designed with CSS keyframes animations for entering views. The animation durations are limited to 300 ms in order to prevent noticeable latency in time-sensitive faculty operations.

G. Testing and Validation Strategy

In order to ensure the accuracy and reliability of the system, comprehensive testing is done. Although the integration testing ensures that there will be seamless communication among parts like database queries and applications, the unit testing is performed on each module. The functional testing ensures that user activities like generating reports, marking attendance, and authentication process work fine. Performance and usability testing is done to check the responsiveness and usability of the system under normal operational conditions. The correctness of individual application logic parts was checked independently of the HTTP layer and the database via unit tests. Some of the key units tested include:

- hash_pw(): validated that the hexadecimal format of the 64-bit SHA-256 digest output was stable and matched the reference hashes from the National Institute of Standards and Technology's Secure Hash Standard.
- Eligibility logic flow: validated with boundary cases where P equals T, P equals T minus 15, and P equals zero, verifying proper allocation into warning, critical, and safe ranges, respectively.
- Formula for class requirement calculation: validated against personal computations of predicted outputs for selected parameter sets (N_p , N_t , T), including the special case where P equals T (where C^R should be zero).
- Login, staff, and admin decorators' behavior: ensured that unauthorized users lacking credentials or insufficient privileges would be routed to the appropriate destinations via the login, staff, and admin requirements, respectively.

  Integration tests were run using Flask's built-in test client, which allows simulated HTTP requests without network traffic. An in-memory SQLite database was created (via sqlite3.connect(':memory:')), initialized through the init_db() function, and populated with artificial attendance and student data for every test case. The following integration points were validated:

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

- POST /login/admin and POST /login/teacher: verified the presence of the session variable in case of valid credentials and absence of the session state in case of invalid credentials.
- POST /students: checked the successful execution of the INSERT statement as well as updates in the student list on the following GET request.
- POST /mark/: upon successful insertion of the attendance entry, redirect to /student/.
- POST /delete/: verified the cascading deletion of the student and corresponding attendance information. GET /eligibility-report?threshold=: checked categorization consistency in cases of 60, 75, and 85 threshold values.

A synthetic set of 50 students with 90 days of attendance entries (4,500 rows) was generated to benchmark page response times for both the dashboard and the eligibility report endpoints for the initial performance testing phase. The mean response times on average commodity hardware never exceeded 120 milliseconds, thus indicating that SQLite's query planner is capable of handling the aggregate load effectively in question at the desired scale. Three users were selected for walkthroughs representing an administrator, a teacher, and a student user for the purpose of usability evaluation.

Table 4. Test Coverage Summary

Test Level	Scope	Method	Outcome
Unit	hash_pw(), classifiers	Assertion-based	Pass — all reference vectors matched
Integration	Route ↔ SQLite	Flask test client	Pass — all CRUD operations verified
Functional	Full user journeys	Manual browser testing	Pass — all roles exercised
Performance	Dashboard, report (50 students, 90 days)	Timer-based measurement	Pass — <120 ms mean response
Usability	UI walkthrough	Structured user sessions	refinements

H. Deployment and Execution Environment

The package may be installed locally or hosted in a cloud environment to operate as a web server written in Flask. Since Flask and SQLite are light packages, there is no complex setup process needed. A main application starts the web server and creates the database for the system. Adaptability is ensured through flexible implementation, ensuring high accessibility, scalability, and maintainability in numerous operating environments.

The AMS is packaged as a standalone Python script with an ordinary __main__ guard controlling the start process. The application uses app.run() to start the Werkzeug development server after calling init_db(), which builds the database schema and populates default users. The program retrieves the PORT variable to determine the port number for running the server; otherwise, it uses port 5000 to execute without configuration and integrate easily into the cloud platform. The address binds to 0.0.0.0 to listen for incoming connections on all available IP addresses.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

The program uses a hard-coded development fallback to retrieve the application's secret key from the SECRET_KEY environment variable. Since the secret key is used to sign Flask's HMAC cookies, it should have a length of at least 128 bits for production use.

The AMS is versatile for many deployment targets due to the independence of the SQLite database and the lightweight nature of the Flask framework with its small memory footprint (approximately 6 MB). This table shows the available combinations:

Table 5. Supported Deployment Configurations

Environment	Infrastructure	Configuration Notes
Local development	Python 3.10+, pip	Run python app.py; database file created in working directory
On-premises institutional	Linux server, systemd	Set PORT and SECRET_KEY env vars; bind to internal network interface
Platform-as-a-Service (PaaS)	Heroku / Render / Railway	Procfile: web: python app.py; DATABASE path mapped to ephemeral storage or external volume
Containerised	Docker	FROM python:3.11-slim; COPY . .; RUN pip install flask; CMD ["python","app.py"]

Flask itself along with its transitively dependent packages (Werkzeug, Jinja2, Click, MarkupSafe, and associated risks) are the only runtime packages that can be installed by using the pip install flask command. The simplicity and reduction in risk in deployment operations are significantly improved due to the fact that no dependency on a database server, messaging broker, or other services exists. A requirements.txt file specifying exact versions of these dependencies for consistent builds within CI/CD pipelines should be considered.

I. Scalability Considerations and Future Infrastructure Path

The current design allows for the support of tens to low hundreds of concurrent users in a single institution/single server environment. If a deployment needs to be able to sustain writes from multiple faculty members, the recommendation is to switch over to client-server relational database management systems (PostgreSQL/MySQL using SQLAlchemy ORM). This is due to the limitations imposed by SQLite's write serialization scheme when there is a heavy number of writes being made on the database. Once that bottleneck has been alleviated, horizontal scaling will not be difficult since the Flask application does not have any form of user state outside of the database.

Containerization using Docker makes horizontal scaling a lot easier because Docker helps build images that can be orchestrated with Kubernetes or cloud-based PaaS solutions. In terms of TLS termination, static file caching, and requests rate limitation in production, a reverse proxy configuration (nginx or Caddy) is suggested to be deployed before the Flask application.

Due to the design principle of modularity of this application, only relevant parts can be maintained without major refactoring. The fact that all views have explicit database connection management means that issues related to the current state of the connection pool will not happen anymore. In case the application cannot be reached, a simple database artifact copy will be enough to make backups. Online backup functionality can be used in production with SQLite. Under data governance guidelines, it is recommended to have logging and auditing of the accesses made.

ISSN: 2582 - 6379
IJISEA Publications
International Journal for Interdisciplinary Sciences and Engineering Applications
IJISEA - An International Peer- Reviewed Journal
2026, Volume 7 Issue 2
www.ijisea.org

IV. RESULTS AND DISCUSSION

In assessing the effectiveness of the developed Web-Based Attendance Management System, the system was successfully deployed in an academic environment and tested. The system performed well regarding automating the process of taking attendance, thereby saving considerable amounts of time compared to the conventional paper-based method of taking attendance. Using a web interface, it became easy for lecturers to take attendance, and the data was saved automatically in the database. This helped ensure accuracy and accessibility of the data in real-time.

From the results obtained, the system takes attendance accurately due to minimized human errors, such as mistakes while entering data and missing data. Efficient storage and fast retrieval of attendance data was possible due to the incorporation of a structured database. Additionally, the system managed to generate attendance statistics and graphs, which enabled users to easily analyze trends and patterns of attendance data. This feature was particularly useful in monitoring students' attendance behavior.

The performance analysis showed stable functioning of the system under ordinary operational conditions with fast processing of input data, data searching, and generating reports. Using lightweight platforms like Flask and SQLite made it more convenient to deploy the system. Responding web interfaces were considered to be more convenient because they ensured that all data can be accessed from different devices.

Analytical functionality played an important role in providing value as the system transformed unprocessed attendance information into valuable information. Identification of students in need of assistance became possible with dynamic calculation of attendance rates and grouping of students using preset boundaries. It allows for improving decision-making processes and ensuring proactivity in dealing with attendance problems.

However, some limitations were identified while evaluating the system. In particular, the system uses manual data entry for tracking students' attendance, and although it is considered to be an improvement compared to the previous methods, users still make errors. Moreover, SQLite may limit scalability of the solution when many users are accessing it simultaneously. It also lacks advanced functionalities such as biometric authentication and machine-learning prediction capabilities.

In summary, the results indicate that through provision of a reliable, efficient, and easy-to-use method, the proposed system effectively addresses the weaknesses associated with the manual process. The system turns out to be a useful tool for academic tracking and decision-making purposes, particularly with the addition of analytics. For future enhancements of the system to increase efficiency, the focus should be on scalability, innovation, and use of advanced technologies.

REFERENCES

[1] Putri, I. M., Ramadhani, D. P., Indriyani, P., Aidah, E. N., & Cahyani, A. P. (2025). Predictive Analytics in attendance systems for employee productivity and accountability. *International Transactions on Education Technology (ITEE)*, *3*(2), 104–113. https://doi.org/10.33050/itee.v3i2.718

[2] Tripathi, A., & Reddy, K. (2025). Development of an Intelligent System for Predictive Analysis of Student Academic Performance Using Machine Learning Techniques. *Management*, 1110– 1117. https://doi.org/10.1109/icict64420.2025.11005033

[3] Enderle, C., Kotschy, L., Ricking, H., & Kreitz-Sandberg, S. (2025). Their voices matter: student and professional perspectives on overcoming school attendance problems in the context of social,

emotional, and behavioral difficulties. *Frontiers in Education*, *10*. https://doi.org/10.3389/feduc.2025.1627098

[4] Junaidi, Wahyono, T., & Sembiring, I. (2025). AI-driven competency recommendations based on attendance patterns and academic performance. *Computers and Education Artificial Intelligence*, *8*, 100423. https://doi.org/10.1016/j.caeai.2025.100423

[5] Men, Y., & Zhao, M. (2025). Hybrid machine learning techniques for improving student management and academic performance. *International Journal of Information and Communication Technology*, *26*(17), 93–108. https://doi.org/10.1504/ijict.2025.146666

[6] Shrivastava, A., Prasad, S. J. S., Yeruva, A. R., Mani, P., Nagpal, P., & Chaturvedi, A. (2023). IoT Based RFID Attendance Monitoring System of Students using Arduino ESP8266 & Adafruit.io on Defined Area. *Cybernetics & Systems*, *56*(1), 21–32. https://doi.org/10.1080/01969722.2023.2166243

[7] Arulogun, O. T., Olatunbosun, A., Fakolujo, O. A., & Olaniyi, O. M. (2013). *RFID-Based Students Attendance Management System*. http://irepo.futminna.edu.ng:8080/jspui/handle/123456789/8782

[8] Paul, J., Mitra, A., Kulhadiya, D., Pawar, T., Roy, A., & Sil, J. (2025). A Comprehensive Approach to Real-time Attendance Systems: Integrating Face Recognition and Emotion Detection and with Web Technologies. *Procedia Computer Science*, *258*, 3436–3446. https://doi.org/10.1016/j.procs.2025.04.600

[9] Ţălu, M. (2025). Exploring IoT applications for transforming university education: smart classrooms, student engagement, and innovations in teacher and student-focused technologies. *Buletin Ilmiah Sarjana Teknik Elektro*, *7*(1), 09–29. https://doi.org/10.12928/biste.v7i1.12361

[10] Butt, A. U. R., Ali, H., Asif, M., Alfraihi, H., Ishak, M. K., & Ammar, K. (2025). Enhancing student management through hybrid machine learning and rough set models: a framework for positive learning environments. *IEEE Access*, *13*, 80834–80846. https://doi.org/10.1109/access.2025.3565585

[11] Ahmed, H. M. M., El-Sabagh, H. A., & Elbourhamy, D. M. (2025). Effect of Gamified, Mobile, Cloud-Based Learning Management System (GMCLMS) on student engagement and achievement. *International Journal of Educational Technology in Higher Education*, *22*(1). https://doi.org/10.1186/s41239-025-00541-1

[12] S Swiderski, T., Fuller, S. C., & Bastian, K. C. (2025). Student-Level attendance patterns across three Post-Pandemic years. *Educational Evaluation and Policy Analysis*, *48*(1), 400–407. https://doi.org/10.3102/01623737251315715.