សាកលវិទ្យាល័យភូមិន្ទភ្នំពេញ

ROYAL UNIVERSITY OF PHNOM PENH

ប្រធានបទរបាយការណ៍

University staff Attendance System

A Report

Submitted in Partial Fulfillment of the Requirements for Semester I, Year III
Bachelor of Engineering in Data Science and Engineering

Group08

January 2026

សាកលវិទ្យាល័យភូមិន្ទភ្នំពេញ

ROYAL UNIVERSITY OF PHNOM PENH

ប្រធានបទរបាយការណ៍

University staff Attendance System

A Report

Submitted in Partial Fulfillment of the Requirements for Semester I, Year III
Bachelor of Engineering in Data Science and Engineering

Group08

Examination committee: Mr. Chap Chanpiseth

January 2026

ABSTRACT IN KHMER

តម្រូវការកាន់តែខ្ពស់សម្រាប់ការគ្រប់គ្រងរដ្ឋបាលដែលមានភាពត្រឹមត្រូវ និងមានប្រសិទ្ធភាពនៅក្នុងស្ថាប័នអប់រំថ្នាក់ឧត្តម បានបង្ហាញឱ្យឃើញពីកម្រិតកំណត់នៃវិធីសាស្ត្របែបប្រពៃណីក្នុងការគ្រប់គ្រងវត្តមាន និងការសុំច្បាប់ឈប់សម្រាក។ វិធីសាស្ត្រដូចជា ការកត់ត្រាលើក្រដាស និងការប្រើតារាង Excel ឬសៀវភៅបញ្ជីអេឡិចត្រូនិក គឺចំណាយពេលវេលាច្រើន ងាយកើតមានកំហុស និងមិនសមស្របសម្រាប់ការតាមដាន និងរាយការណ៍ទិន្នន័យតាមពេលវេលាពិត ជាពិសេសនៅក្នុងបរិយាកាសសាកលវិទ្យាល័យដែលមានទំហំធំ។

គម្រោងនេះបង្ហាញពីការរចនា និងអភិវឌ្ឍប្រព័ន្ធគ្រប់គ្រងវត្តមានបុគ្គលិកសាកលវិទ្យាល័យ ដែលមានគោលបំណងបំលែងប្រព័ន្ធបែបដៃទៅជាប្រព័ន្ធឌីជីថល និងធ្វើឱ្យដំណើរការកត់ត្រាវត្តមាន ការស្នើសុំច្បាប់ឈប់សម្រាក និងការរៀបចំកាលវិភាគ មានភាពងាយស្រួល និងមានប្រសិទ្ធភាពតាមរយៈវេទិកាតែមួយតាមគេហទំព័រ។ ប្រព័ន្ធត្រូវបានអភិវឌ្ឍដោយប្រើស្ថាបត្យកម្ម client–server សម័យទំនើប ដោយមាន RESTful API ជាផ្នែកខាងក្រោយ និងម៉ូដែលមូលដ្ឋានទិន្នន័យទំនាក់ទំនង (relational database) ដើម្បីធានាភាពត្រឹមត្រូវ ស្ថិរភាព និងសុវត្ថិភាពក្នុងការចូលប្រើទិន្នន័យ។

ការគ្រប់គ្រងសិទ្ធិផ្អែកលើតួនាទី (Role-Based Access Control – RBAC) ត្រូវបានអនុវត្ត ដើម្បីកំណត់សិទ្ធិប្រើប្រាស់ប្រព័ន្ធសម្រាប់តួនាទីផ្សេងៗ រួមមាន ប្រធាន (Heads) អ្នកគ្រប់គ្រង (Administrators) Mazers Assistants និងគ្រូបង្រៀន ឬបុគ្គលិក។ វាជួយធានាឱ្យមានភាពទទួលខុសត្រូវ និងបង្ការ ការកត់ត្រាវត្តមានដោយគ្មានសិទ្ធិ ឬការកែប្រែទិន្នន័យដោយខ្លួនឯង។

មុខងារសំខាន់ៗដែលបានអនុវត្ត រួមមាន ការកត់ត្រាវត្តមានតាមពេលវេលាពិត ការស្នើសុំ និងអនុម័តច្បាប់ឈប់សម្រាកតាមប្រព័ន្ធអនឡាញ ការគ្រប់គ្រងកាលវិភាគ និងផ្ទាំងព័ត៌មាន (dashboard) ដែលបង្ហាញតាមតួនាទីនីមួយៗ។ វត្តមានរបស់គ្រូបង្រៀនត្រូវបានកត់ត្រាដោយ Mazers ខណៈដែលវត្តមានបុគ្គលិកការិយាល័យត្រូវបានគ្រប់គ្រងដោយ Assistants ដែលជួយលុបបំបាត់លទ្ធភាពនៃការកត់វត្តមានដោយខ្លួនឯង។ អ្នកគ្រប់គ្រងមានតួនាទីក្នុងការត្រួតពិនិត្យការកំណត់ប្រព័ន្ធ ការគ្រប់គ្រងអ្នកប្រើប្រាស់ និងការតាមដានទិន្នន័យ ខណៈដែលប្រធានមានសិទ្ធិក្នុងការត្រួតពិនិត្យ និងអនុម័ត។

ស្ថាបត្យកម្មប្រព័ន្ធត្រូវបានរចនាឡើងដោយមានលក្ខណៈម៉ូឌុល និងអាចពង្រីកបាន ដែលអនុញ្ញាតឱ្យបន្ថែមមុខងារថ្មីៗនាពេលអនាគត ដោយមិនចាំបាច់ផ្លាស់ប្តូររចនាសម្ព័ន្ធសំខាន់ៗ។ ជាសរុប ប្រព័ន្ធគ្រប់គ្រងវត្តមានបុគ្គលិកសាកលវិទ្យាល័យនេះ ផ្តល់ជាដំណោះស្រាយដែលអាចទុកចិត្តបាន និងអាចពង្រីកបាន ដែលជួយបង្កើនប្រសិទ្ធភាពការងារ កែលម្អភាពត្រឹមត្រូវនៃទិន្នន័យ និងគាំទ្រដល់ការសម្រេចចិត្តយ៉ាងមានប្រសិទ្ធភាពក្នុងការគ្រប់គ្រងសាកលវិទ្យាល័យ។

ABSTRACT IN ENGLISH

The increasing need for accurate and efficient administrative management in higher education institutions has revealed the limitations of traditional manual attendance and leave management methods. Practices such as paper registers and spreadsheet-based records are time consuming, error-prone, and unsuitable for real-time monitoring and reporting, particularly in large academic environments.

This project presents the design and implementation of a University Staff Attendance System aimed at digitizing and streamlining attendance tracking, leave request management, and schedule coordination through a centralized web-based platform. The system is developed using a modern client–server architecture with a RESTful API–based backend and a relational database model to ensure data integrity, consistency, and secure access. Role-Based Access Control (RBAC) is applied to define system privileges for Heads, Administrators, Mazers, Assistants, and Teachers/Staff, ensuring accountability and preventing unauthorized attendance marking.

The core functionalities implemented include real-time attendance recording, online leave request submission and approval workflows, schedule management, and role-specific dashboards. Attendance for teaching staff is recorded by Mazers, while office staff attendance is managed by Assistants, eliminating the possibility of self-attendance manipulation. Administrative users oversee system configuration, user management, and data monitoring, while Heads retain supervisory and approval authority.

The system architecture is designed to be modular and extensible, allowing future enhancements to be integrated without significant structural changes. Overall, the University Staff Attendance System provides a scalable and reliable solution that improves operational efficiency, enhances data accuracy, and supports effective decision-making in university administration.

Keywords: University Staff Attendance System, Attendance Management, Leave Request Management, Role-Based Access Control, Client–Server Architecture, Web-Based System

SUPERVISOR’S RESEARCH SUPERVISION STATEMENT

TO WHOM IT MAY CONCERN

Name of program: Bachelor of Data Science and Engineering
Group 08

Title of research report: **University staff Attendance System**

This is to certify that the research carried out for the above-titled Bachelor’s research report was completed by the group under my direct supervision. The material presented in this report has not been submitted for any other degree.
As the supervisor, I provided guidance throughout the research process and assessed the work for this semester:…………………………………………………………...
....
....
....

Supervisor's name: Mr. Chap Chanpiseth

Supervisor’s signature: ……………………

Date……………………………………………

CANDIDATE'S STATEMENT

TO WHOM IT MAY CONCERN

This is to certify that the research report presented by **Group 08**, entitled **"University staff Attendance System"**, submitted for the degree of **Bachelor of Engineering** at the **Royal University of Phnom Penh**, and completed during **Semester 1, Year 3**, is the result of our collective effort. This project was carried out under the guidance and supervision of our instructor, **Mr. Chap Chanpiseth**, who provided direction, guidance, and evaluation throughout the project. Our group conducted all research activities in accordance with the guidance provided. The members of Group 08 are:

- Yun chanrothmonny
- Chum Monika
- Ye Lita

No reference to, or quotation from, this document may be made without the written approval of the authors.

Signed by Group Leader on behalf of Group 08:
Yun chanrothmonny :

Date: ……………………………………………….

Sign by Supervisor: Mr. Chap Chanpiseth

Supervisor's signature: …………………………..…

Date……………………………………….……

ACKNOWLEGMENTS

We would like to express our sincere and profound gratitude to our supervisor, **Mr. Chap Chanpiseth**, for his invaluable guidance, continuous support, and professional supervision throughout the completion of our research project entitled **“University staff Attendance System”** His deep knowledge, constructive feedback, and thoughtful suggestions played a crucial role in shaping the structure, direction, and overall quality of this project. Through his supervision, we were encouraged to think critically, improve our problem-solving skills, and maintain academic discipline in both research and system development. We are especially grateful for his patience, dedication, and willingness to provide guidance whenever challenges arose during the project.

- We would also like to acknowledge the collective effort and strong cooperation among the members of **Group 08**, consisting of **Yun chanrothmonny ,Chum Monika**, and **Ye Lita**. Each member demonstrated responsibility, commitment, and active participation throughout the project lifecycle. From requirement analysis and system design to implementation and documentation, every member contributed their skills and time to ensure the successful completion of the project. The teamwork, mutual support, and effective communication within the group were key factors in overcoming difficulties and achieving our project objectives.

In addition, we extend our appreciation to the **Royal University of Phnom Penh** for providing a supportive academic environment and learning resources that facilitated our research and development process. Access to academic materials, references, and institutional support greatly contributed to the successful execution of this project.

Finally, we would like to thank all individuals, authors, online resources, and academic references that indirectly supported this research. The information, tools, and discussions obtained from various sources enhanced our understanding and enabled us to complete this project effectively. The encouragement, guidance, and support received throughout this journey have been instrumental in achieving the goals of the project.

.

TABLE OF CONTENTS

LIST OF ACRONYMS

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
DBMS	Database Management System
ERD	Entity Relationship Diagram
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
ORM	Object–Relational Mapping
RBAC	Role-Based Access Control
REST	Representational State Transfer
UI	User Interface
UX	User Experience
SQL	Structured Query Language
DB	Database
ICT	Information and Communication Technology

CHAPTER 1

INTRODUCTION

1.1 Background to the Study

Attendance management plays a vital role in the effective operation of higher education institutions, as it directly influences staff accountability, institutional discipline, and overall administrative efficiency. Universities are complex organizations that employ a diverse workforce, including academic staff, administrative officers, and support personnel, each with distinct schedules, responsibilities, and attendance requirements. Proper monitoring of attendance ensures compliance with institutional policies, accurate payroll processing, fair workload distribution, and reliable performance evaluation.

Despite its importance, attendance management in many universities is still conducted using traditional manual methods such as paper attendance sheets, logbooks, and spreadsheet-based systems. While these approaches may be manageable in small organizations, they become increasingly inefficient and unreliable as the size of the institution and number of staff grow. Manual record-keeping is highly susceptible to human errors, including incorrect entries, missing records, and data duplication. In addition, retrieving historical attendance information or generating comprehensive reports often requires significant time and effort, which delays administrative decision-making.

Another major limitation of manual attendance systems is the lack of transparency and real-time visibility. Attendance records are often stored in isolated files or physical documents, making it difficult for administrators and department heads to monitor attendance patterns, identify irregularities, or detect absenteeism in a timely manner. Furthermore, the absence of a centralized system increases the risk of unauthorized access, record manipulation, and loss of data due to physical damage or misplacement of documents.

With the rapid advancement of information and communication technologies, web-based systems have become essential tools for improving organizational efficiency and data management. Automated attendance systems offer features such as centralized data storage, role-based access control, real-time updates, and secure data handling. These systems not only reduce administrative workload but also improve accuracy, consistency, and accountability across departments. In particular, role-based systems can restrict attendance recording to authorized personnel, thereby preventing misuse such as self-attendance marking.

In response to these challenges, this study focuses on the design and development of a University Staff Attendance System that leverages modern web technologies to automate attendance tracking, leave management, and scheduling processes. The system provides a centralized platform where authorized users can record attendance, manage leave requests, and access real-time dashboards based on their assigned roles. By digitizing attendance management, the proposed system aims to enhance operational efficiency, improve data integrity, and support effective decision-making within university administration.

1.2 Problem Statement

Currently, the management of staff attendance and leave at the university is primarily conducted using manual methods, such as paper registers and spreadsheet-based systems. These approaches are inefficient and prone to human errors, including incorrect entries, missing records, duplication, and delayed updates. As the number of staff increases, these issues are magnified, making it difficult for administrators to maintain accurate and up-to-date attendance records.

The lack of a centralized platform also limits transparency and real-time monitoring. Heads of departments and administrators are unable to track attendance trends or detect absenteeism promptly. Generating reports for payroll, performance evaluation, or administrative planning often requires significant time and effort, resulting in delayed decision-making and reduced operational efficiency.

Moreover, manual systems allow potential misuse, such as staff marking their own attendance, which compromises data integrity and accountability. Communication about leave requests is often informal and delayed, further complicating workflow management. The current system also does not provide insights into patterns of absenteeism, staff workload, or schedule adherence, which are important for planning and institutional governance.

These challenges collectively increase administrative workload, reduce staff productivity, and hinder effective management of university operations. Therefore, there is a strong need for a **web-based, automated, and role-based attendance system** that ensures accurate attendance tracking, secure leave management, controlled access, and real-time reporting. Such a system would improve operational efficiency, enhance data integrity, reduce errors, and support timely and informed decisionmaking across the university. scheduling, ensuring accuracy, transparency, and efficiency in university administration.

1.3 Aim and Objective

Aim

The primary aim of this study is to design and develop a **web-based University Staff Attendance System** that automates attendance tracking, leave management, and schedule coordination. The system seeks to improve data accuracy, enhance administrative efficiency, and support informed decision making within the university.

Objectives

To achieve this aim, the study sets out the following objectives:

1. **Design a centralized web-based platform** for managing staff attendance, leave requests, and schedules.
2. **Implement role-based access control (RBAC)** to assign specific privileges to Administrators, Heads, Mazers, Assistants, and Teachers/Staff.
3. **Enable staff to view personal attendance records** and submit leave requests online in a secure and user-friendly manner.
4. **Allow Heads and Administrators to review, approve, or reject leave requests efficiently**, ensuring accountability and transparency.
5. **Restrict attendance recording to authorized personnel**, such as Mazers for teaching staff and Assistants for office staff, to prevent self-attendance marking and misuse.
6. **Provide real-time dashboards and reporting tools** to track attendance trends, leave patterns, and schedule adherence for effective decision-making.
7. **Ensure system scalability and extensibility**, allowing future integration of features such as mobile support, automated notifications, and biometric attendance methods.

1.4 Rationale of the Study

The rationale for this study is based on the increasing need for **efficiency, accuracy, and transparency** in university administrative processes. Traditional manual methods of attendance management, such as paper registers and spreadsheets, are time-consuming, prone to errors, and difficult to maintain, particularly in large institutions. Inaccurate attendance records, delayed leave approvals, and inefficient scheduling can negatively affect staff accountability, payroll management, and institutional decision-making.

Implementing a web-based University Staff Attendance System provides a centralized platform to streamline these processes. By digitizing attendance tracking, leave management, and schedule monitoring, the system reduces administrative workload, improves record accuracy, and ensures timely reporting. Role-based access control further enhances security by restricting sensitive operations to authorized personnel, preventing misuse or fraudulent attendance recording.

Additionally, the system enables real-time monitoring through dashboards and reporting tools, allowing Heads, Administrators, Mazers, and Assistants to make informed decisions efficiently. The study contributes to the **digital transformation of university operations**, demonstrating how modern information technologies can address real-world administrative challenges while enhancing operational effectiveness, accountability, and productivity.

1.5 Scope and Limitations

Scope

The study focuses on the design and development of a **web-based University Staff Attendance System** that automates attendance tracking, leave management, and scheduling for university staff.

The system includes:

- **Role-Based Access Control (RBAC):** Different privileges for Administrators, Heads, Mazers, Assistants, and Teachers/Staff.
- **Attendance Recording:** Attendance is recorded by authorized personnel to prevent misuse.
- **Leave Request Management:** Staff can submit leave requests online, which are reviewed and approved by Heads or Administrators.
- **Schedule Management:** Centralized management of teaching and office staff schedules.
- **Reporting and Dashboards:** Real-time insights into attendance patterns, leave trends, and staff activity.

The system is designed to be **modular and extensible**, allowing future enhancements such as mobile platform support, automated notifications, and biometric or face recognition–based attendance to be integrated without major restructuring.

Limitations

Despite its advantages, the current implementation of the system has some limitations:

1. **Limited mobile accessibility:** The system is primarily designed for desktop use and lacks a fully responsive mobile interface.
2. **No automated notifications:** Email and SMS notifications for leave approval or attendance alerts are not yet integrated.
3. **No biometric or face recognition attendance:** Attendance is manually recorded by authorized personnel.

4. **Advanced analytics not included:** Predictive or in-depth analytical reporting for administrators is not available in this version.

These limitations represent areas for future improvement and further development, ensuring that the system can evolve to meet broader institutional requirements.

1.6 Structure of Study

Chapter 1 provides the foundation for the study by introducing the context and rationale for developing a web-based University Staff Attendance System. This chapter discusses the background of university staff management, highlighting the reliance on manual attendance and leave recording processes and the resulting inefficiencies in administrative workflows, data accuracy, and schedule coordination. It presents the problem statement by outlining the challenges faced by administrators, Heads, Mazers, Assistants, and staff, including delayed reporting, data inconsistencies, and potential misuse of attendance records. The chapter then defines the research aim and objectives, explains the rationale for implementing a digital attendance system, and clarifies the scope and limitations of the study. Overall, Chapter 1 establishes the framework for the report, providing a clear roadmap for the subsequent chapters.

Chapter 2 presents the literature review related to attendance management systems and supporting technologies. This chapter discusses the concept and importance of attendance management in educational institutions and reviews previous studies on automated attendance systems. It specifically examines fingerprint-based, QR code–based, and facial recognition–based attendance management systems, focusing on their objectives, methodologies, technologies used, results, and limitations. By analyzing these existing approaches, the chapter identifies gaps in current systems, particularly in terms of system integration, scalability, cost, and usability. These identified gaps provide a strong theoretical and technical foundation for the development of the proposed University Staff Attendance System.

Chapter 3 describes the methodology adopted for designing and developing the proposed system. This chapter explains the use of the Agile development methodology and outlines each development phase, including requirements analysis, system design, implementation, testing, and validation. It also details the tools, technologies, and frameworks used for frontend and backend development, as well as the architectural patterns applied to ensure system maintainability and scalability. In addition, the chapter discusses the website development process through sprint planning and highlights ethical considerations related to data security, privacy protection, and role based access control to ensure responsible system usage.

Chapter 4 focuses on system analysis and design of the University Staff Attendance System. This chapter defines the functional requirements for different user roles, including administrators, department heads, class moderators, teachers, and staff. It presents use case diagrams and system workflows to illustrate how attendance recording and leave management processes are carried out within the system. Furthermore, the chapter explains the system architecture using a three-layer Clean Architecture approach and describes the interaction between system components. Finally, the database design is presented through an Entity Relationship Diagram (ERD), detailing table structures, relationships, and constraints that support accurate data storage and system integrity.

Chapter 5 presents the website implementation of the proposed system. This chapter describes the development environment, tools, and configurations used during implementation. It explains the frontend development process, including component architecture, state management, routing, and responsive design. The chapter also details backend development, including RESTful API endpoints, authentication mechanisms, and role-based authorization. In addition, the database implementation, repository pattern usage, and performance optimization strategies are discussed. The chapter concludes by outlining the security features implemented in the system, such as authentication, authorization, input validation, environment security, and error handling, ensuring the system is secure, reliable, and suitable for deployment in a university environment.

Chapter 6 (Results) presents the implementation and outcomes of the University Attendance Management System (UAMS). The system offers a web-based platform that streamlines attendance tracking, schedule management, leave requests, and academic structure management. Key functionalities include secure authentication with role-based access, comprehensive management of departments, majors, classes, subjects, and semesters, as well as real-time attendance recording with timestamps. The system provides role-specific dashboards, giving administrators, faculty, and staff immediate insights into relevant data. Analysis of the results shows significant improvements in data integrity, operational efficiency, and system performance compared to previous manual or spreadsheet-based methods. User feedback indicates a high level of satisfaction with the interface design, navigation, and functional utility, although limitations such as dependence on internet connectivity, lack of a native mobile application, absence of predictive analytics, and a manual feedback loop were identified.

Chapter 7 (Discussion) explores the challenges encountered during development, including designing a flexible role-based access control model, managing complex relationships between academic entities, and ensuring data security through policies such as Row Level Security. The discussion highlights that these challenges were successfully addressed using modern web technologies, resulting in a system that is scalable, secure, and capable of improving academic management. The system demonstrates clear advantages over traditional attendance methods astund simpler web-based solutions by integrating attendance, scheduling, leave management, and academic structure management into a unified platform.

Chapter 8 (Conclusion and Recommendations) summarizes the project's accomplishments and provides guidance for future improvements. The system successfully enhances efficiency, accuracy, and accessibility of attendance and academic management within the university environment. It supports multiple user roles with specific permissions and ensures secure, centralized data storage. Recommendations for further development include creating a mobile friendly version, implementing advanced analytics and automated notifications, optimizing system performance, and integrating a feedback module. Future work may involve incorporating machine learning for predictive attendance analysis, QR code or biometric attendance, multi-campus support, cloud scalability enhancements, and integration with other academic systems. Overall, the study demonstrates that UAMS provides a strong foundation for digital attendance management with significant potential for future expansion and enhancement.

CHAPTER 2

LITERATURE REVIEW

This chapter presents a review of previous studies and technologies related to attendance management systems. It begins with an overview of attendance management, highlighting the importance of automated systems in modern institutions. It then discusses fingerprint based, QR code–based, and facial recognition–based attendance systems in detail, focusing on their objectives, methodologies, findings, and limitations. The chapter concludes by identifying research gaps relevant to the development of the proposed University Staff Attendance System.

2.1 Overview of Attendance Management Systems

Attendance management is a critical administrative function in educational institutions and organizations, ensuring that staff and students fulfill their responsibilities while enabling efficient institutional operations. Traditional attendance recording methods, such as paper registers or spreadsheets, are increasingly inadequate due to several limitations: • Inefficiency: Manual recording and report generation require significant administrative effort. • Inaccuracy: Paper-based systems are prone to errors, data duplication, and omissions. • Vulnerability to fraud: Manual methods may allow impersonation or falsification of records. • Limited reporting and analytics: Extracting trends or generating timely reports is labor-intensive and often delayed. Automated attendance management systems address these challenges by leveraging modern technologies to improve accuracy, efficiency, and transparency. Common approaches include: • Biometric systems (fingerprint, facial recognition) for identity verification. • QR code–based systems that utilize mobile devices to record attendance. • RFID or smart card systems for automated check-ins. These systems provide real-time monitoring, secure record-keeping, automated reporting, and enhanced accountability, significantly reducing administrative workload and improving institutional productivity.

2.2 Previous Studies

2.2.1 Literature Review 1

Fingerprint-Based Attendance Management System

Obansola, Makinde, Adeshina, and Adebayo (2016) conducted a study titled "Development of Staff Attendance Management System using Fingerprint Biometric Identification 8 Technique." The main objective of the study was to design and implement an automated staff attendance management system that addresses the limitations of traditional manual attendance methods commonly used in educational institutions. The authors identified that manual attendance recording using paper-based registers is time consuming, insecure, and prone to errors such as impersonation, data loss, and inaccurate record keeping. To overcome these challenges, the study proposed a biometric-based attendance system that utilizes fingerprint recognition for staff identification and authentication. The system was developed using fingerprint biometric identification techniques. Staff members were first enrolled by capturing their fingerprint data and personal information into a centralized database. Attendance was recorded by matching scanned fingerprints against stored templates using minutiae extraction and matching algorithms. The system architecture was designed using object-oriented principles and Unified Modeling Language (UML) diagrams. The implementation was carried out using the C# programming language on the .NET Framework, with MySQL used as the backend database to store staff biodata, fingerprint templates, and attendance records. The results of the study showed that the developed system significantly improved the efficiency and reliability of attendance management. The system successfully eliminated impersonation, reduced paperwork, enhanced data security, and enabled fast generation of attendance reports. The study concludes that fingerprint-based attendance systems provide a secure and accurate alternative to conventional manual attendance methods, although they require specialized hardware and regular maintenance. [1]

2.2.2 Literature Review 2

QR Code–Based Attendance Management System Building on earlier biometric-based attendance systems, Narangammana and Shamila (2025) conducted a study titled "QR Code Based Attendance Management System." The objective of the study was to design and implement a smart attendance system to replace the traditional manual attendance process used at Trincomalee Campus, Eastern University, Sri Lanka. Similar to Obansola et al. (2016), the authors identified that conventional paper-based attendance methods are inefficient, time-consuming, and vulnerable to issues such as fake signatures, human error, and data inconsistency. These problems often result in inaccurate attendance records and difficulties in calculating student eligibility for end-of-semester examinations. To address these challenges, the study proposed a QR code–based attendance management system that leverages smartphone technology. The proposed system was developed as a web-based application supported by a mobile QR code scanning mechanism. Administrators generated unique, time-limited QR codes for each class session, while students marked their attendance by scanning the QR code using their smartphones. Attendance data, including date and time, was automatically recorded and stored in a centralized database. The system also included features such as attendance tracking, report generation, warning notifications for low attendance, and automated eligibility calculation for examinations. The results demonstrated that the QR code–based attendance management system significantly reduced paperwork, minimized human errors, prevented attendance 9 impersonation, and saved administrative time. Compared to fingerprint-based systems, the QR code approach was found to be more cost-effective, easier to implement, and less invasive in terms of privacy. The study concludes that QR code–based attendance systems provide a practical and scalable alternative for higher education institutions, particularly in developing countries. [2]

2.2.3 Literature Review 3

Facial Recognition–Based Attendance Management System Further advancements in attendance management systems have incorporated artificial intelligence and computer vision technologies. Lateef and Kamil (2023) conducted a study titled “Facial Recognition Technology-Based Attendance Management System Application in Smart Classroom.” The main objective of the study was to design and implement an automated attendance management system using facial recognition technology. The authors highlighted that traditional attendance systems, such as paper-based registers and name-calling, are time-consuming, prone to impersonation, insecure, and disruptive to lecture time. In contrast to the fingerprint-based system proposed by Obansola et al. (2016) and the QR code–based system developed by Narangammana and Shamila (2025), this study employed facial recognition technology to enable fully automated and contactless attendance recording. The proposed system was developed using Python and integrated with computer vision and machine learning libraries, including OpenCV, Dlib, TensorFlow, Keras, and Scikit-learn. Facial detection techniques such as Viola–Jones, Histogram of Oriented Gradients (HOG), and Multi-task Cascaded Convolutional Neural Network (MTCNN) were applied. Facial recognition was performed using Local Binary Pattern Histogram (LBPH), FaceNet, and Dlib’s ResNet-based models. A MySQL database was used to store student information, facial embeddings, and attendance records. The system was evaluated in a real classroom environment involving 30 students under varying lighting conditions, seating distances, and facial poses. The experimental results showed that the system achieved up to 100% accuracy in real-time attendance recording. The study concludes that facial recognition–based attendance systems provide a highly accurate, efficient, and contactless solution for smart classroom environments, although they may require higher computational resources and careful consideration of privacy concerns. [3]

2.4 Summary and Research Gap

Based on the reviewed studies, it is evident that automated attendance management systems offer significant improvements in efficiency, accuracy, and security compared to traditional manual methods. Fingerprint-based systems provide high accuracy but require specialized hardware and maintenance. QR code–based systems offer a cost-effective and easy-to implement solution but may still face challenges related to proxy attendance. Facial recognition systems enable fully automated and contactless attendance recording but depend on environmental conditions and advanced computational resources. Despite these advancements, many existing systems focus only on attendance recording and lack integration with broader management platforms. Therefore, there is a need for a secure, scalable, and user-friendly system that balances accuracy, cost, privacy, and ease of 10 implementation. This research aims to address these gaps by developing an improved system that builds upon previous work while incorporating modern web-based technologies.

CHAPTER 3

METHODOLOGY

3.1 Research Design Tools and Technologies Used

Development Approach: Agile Methodology

Agile methodology was selected due to its suitability for iterative system development, frequent requirement refinement, and continuous feedback. This approach allowed the system architecture and features to evolve based on testing results and stakeholder needs.

The project follows an iterative agile development approach with the following phases:

3.1.1 Requirements Analysis Phase

- Identified 5 distinct user roles and their responsibilities
- Mapped business workflows (attendance recording, leave approval, reporting)
- Defined data relationships (departments, majors, classes, schedules)

3.1.2 System Design Phase

- Clean Architecture pattern with three layers (Core, Infrastructure, Presentation)
- Database schema design with referential integrity
- API endpoint specification for frontend-backend communication

3.1.3 Implementation Phase

- Sprint-based development with incremental feature delivery
- Test-Driven Development (TDD) for critical business logic
- Code refactoring iterations for maintainability

3.1.4 Testing & Validation Phase

- End-to-end testing of complete workflows
- Unit testing (72 test cases)
- Security testing for authentication and authorization

3.2 Technologies Used

3.2.1 Frontend Technologies:

- **React 18**: Component-based UI library for building interactive interfaces
- **TypeScript**: Static typing for enhanced code quality and developer experience

- **Vite**: Fast build tool and development server
- **TanStack Query (React Query)**: Server state management and data fetching
- **Shadcn/ui**: Accessible component library built on Radix UI
- **Tailwind CSS**: Utility-first CSS framework for responsive design
- **Recharts**: Charting library for analytics visualization
- **React Hook Form**: Form state management with validation
- **Zod**: Schema validation for form inputs

3.2.2 Backend Technologies:

- **Node.js**: JavaScript runtime environment
- **Express.js**: Web application framework for RESTful API
- **TypeScript**: Type-safe backend development
- **Drizzle ORM**: Type-safe SQL query builder and ORM
- **MySQL**: Relational database management system
- **bcrypt**: Password hashing for security
- **express-session**: Session-based authentication
- **dotenv**: Environment variable management

3.2.3 Development Tools:

- **Git**: Version control system
- **npm**: Package manager
- **ESBuild**: JavaScript bundler for backend
- **Vitest**: Unit testing framework
- **VS Code**: Integrated development environment
- **MySQL Workbench**: Database design and management

3.2.4 Architecture & Patterns:

- **Clean Architecture**: Three-layer separation (Core, Infrastructure, Presentation)
- **Repository Pattern**: Data access abstraction
- **Dependency Injection**: Loose coupling between components
- **Use Case Pattern**: Business logic encapsulation

3.3 Website Development Process

Sprint 1: Database Design (Week 1-2)

- Designed normalized database schema with 8 tables
- Established foreign key relationships for referential integrity
- Created indexes for query optimization
- Implemented timestamp tracking (created_at, updated_at)

Sprint 2: Backend Development (Week 3-4)

- Set up Clean Architecture structure
- Implemented 7 core use cases (Auth, User, Attendance, Leave, Class, Schedule, Config)
- Created 10 controller endpoints with role-based authorization
- Developed repository pattern for database operations
- Implemented session-based authentication with bcrypt password hashing

Sprint 3: Frontend Development (Week 5-6)

- Built feature-based architecture (auth, attendance, users, leaves)
- Created role-specific dashboards (5 different views)
- Implemented reusable UI components (47 components)
- Developed responsive layouts with Tailwind CSS
- Integrated React Query for efficient data fetching

Sprint 4: Integration & Testing (Week 7-8)

- Connected frontend to backend APIs
- Implemented error handling and loading states
- Created 72 automated tests (unit + E2E)
- Performed security testing and vulnerability fixes
- Conducted cross-browser compatibility testing

Sprint 5: Refactoring & Optimization (Week 9-10)

- Extracted business logic from controllers to use cases
- Removed unused dependencies to improve maintainability
- Implemented environment validation for production
- Created comprehensive documentation
- Optimized bundle size and load performance

3.4 Ethical Considerations

The system was designed with a strong emphasis on data privacy, security, and controlled access to sensitive information. All user passwords are protected using **bcrypt one-way hashing**, ensuring that original credentials cannot be recovered even in the event of a data breach. Sensitive configuration values such as database credentials and session secrets are stored exclusively in environment variables and are never hardcoded in the source code or exposed to the client side.

Access to system data is strictly regulated through **role-based access control (RBAC)**. Each user role is granted only the minimum level of access required to perform its responsibilities. For example, teachers and staff members can view only their personal attendance and leave records, while department heads are restricted to data belonging to their respective departments. System administrators have full access to all system data and configuration functions, enabling effective system management while maintaining accountability.

To ensure transparency and traceability, all critical data records such as attendance logs, leave requests, and user profiles are automatically timestamped upon creation and update. Users are able to track the status of their leave requests in real time, promoting clarity and trust in the system's operations. Attendance records are retained for historical reporting and analysis purposes, while safeguards are in place to prevent unauthorized deletion or modification of user data.

Overall, these measures ensure compliance with fundamental data protection principles, including confidentiality, integrity, and accountability, making the system suitable for use in an educational institutional environment.

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 User Document Requirements

Role 1: Head (Department Head)

- View department-wide attendance summary
- Access attendance statistics for department staff
- Review leave requests within their department
- Monitor department performance metrics
- Filter data by department users only
- Access all analytics and reports for attendance records

Role 2: Admin (System Administrator)

- Full system access across all departments
- User management (create, update, delete, assign roles)
- Configure departments, majors, subjects

Role 3: Class Moderator

- Monitor class-level attendance
- View teacher session
- Coordinate class activities

Role 4: Teacher

- View personal attendance/leave requests record
- Submit leave requests
- View personal statistics only

Role 5: Staff

- View personal attendance record
- Submit leave requests
- Track own leave balance

4.2 Use Case Diagram

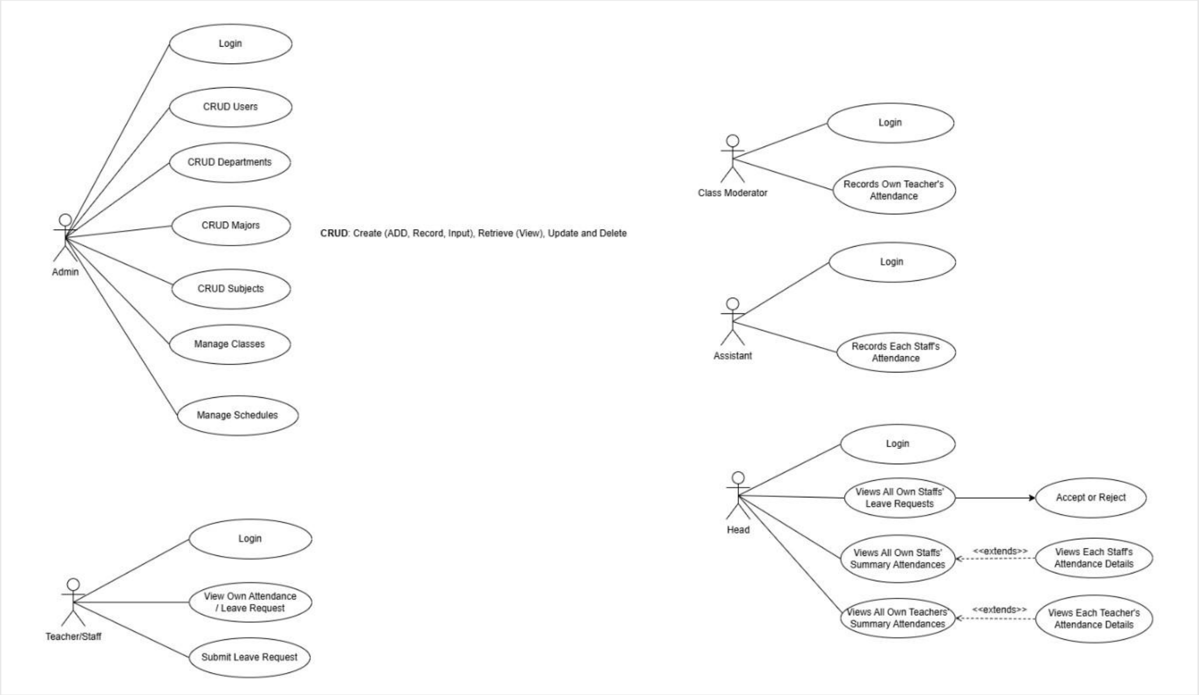
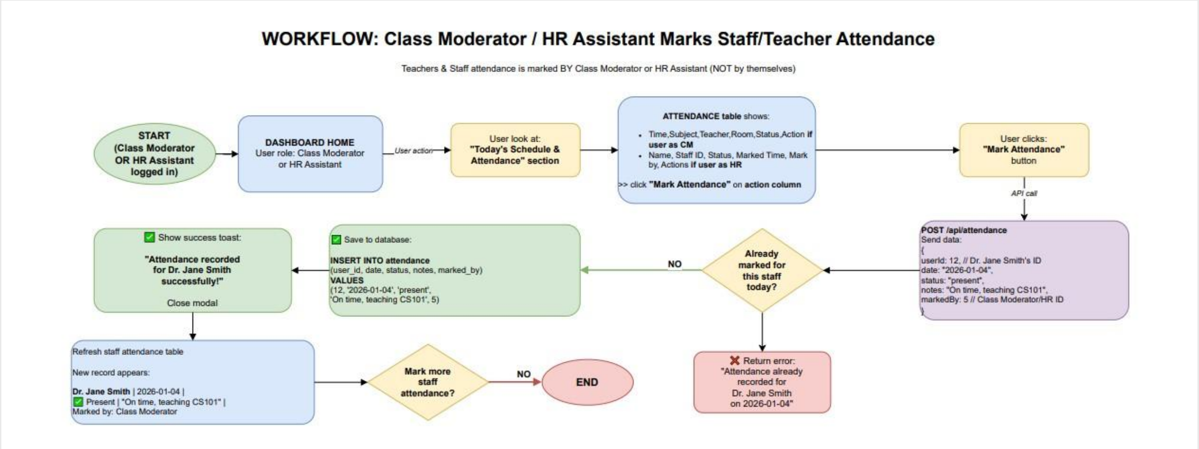

Admin
Login
CRUD Users
CRUD Departments
CRUD Majors
CRUD Subjects
Manage Classes
Manage Schedules
CRUD: Create (ADD, Record, Input), Retrieve (View), Update and Delete
Teacher/Staff
Login
View Own Attendance / Leave Request
Submit Leave Request
Class Moderator
Login
Records Own Teacher's Attendance
Assistant
Login
Records Each Staff's Attendance
Head
Login
Views All Own Staffs' Leave Requests
Accept or Reject
Views All Own Staffs' Summary Attendances
<<extends>>
Views Each Staff's Attendance Details
Views All Own Teachers Summary Attendances
<<extends>>
Views Each Teacher's Attendance Details

Figure 1 – Use Case Diagram

Figure 1 illustrates the use case diagram of the system, showing the interaction between different user roles and the core system functionalities.

4.3 System Workflow


WORKFLOW: Class Moderator / HR Assistant Marks Staff/Teacher Attendance
Teachers & Staff attendance is marked BY Class Moderator or HR Assistant (NOT by themselves)
START (Class Moderator OR HR Assistant logged in)
DASHBOARD HOME User role: Class Moderator or HR Assistant
User action
User look at: "Today's Schedule & Attendance" section
ATTENDANCE table shows:
Time,Subject,Teacher,Room,Status,Action if user as CM
Name, Staff ID, Status, Marked Time, Mark by, Actions if user as HR
>> click "Mark Attendance" on action column
User clicks: "Mark Attendance" button
API call
Already marked for this staff today?
NO
✓ Save to database:
✓ Show success toast:
"Attendance recorded for Dr. Jane Smith successfully!"
Close modal
Refresh staff attendance table
New record appears:
Dr. Jane Smith | 2026-01-04 |
✓ Present | "On time, teaching CS101" |
Marked by: Class Moderator
Mark more staff attendance?
NO
END
✗ Return error: "Attendance already recorded for Dr. Jane Smith on 2026-01-04"

```
POST /api/attendance
Send data:
{
userId: 12, // Dr. Jane Smith's ID
date: "2026-01-04",
status: "present",
notes: "On time, teaching CS101",
markedBy: 5 // Class Moderator/HR ID
}
```

```
INSERT INTO attendance
(user_id, date, status, notes, marked_by)
VALUES
(12, '2026-01-04', 'present',
'On time, teaching CS101', 5)
```

Figure 2 – Attendance flow Diagram

This workflow illustrates how staff and teacher attendance is recorded by authorized roles, specifically the Class Moderator or HR Assistant. After logging into the system, the user accesses the dashboard and navigates to the daily schedule and attendance section. The system displays relevant staff or teacher records for the selected date. Upon selecting "Mark Attendance," the system validates whether attendance has already been recorded for that individual on the same day. If no prior record exists, the attendance data is saved to the database and a confirmation message is displayed. If a duplicate entry is detected, the system prevents the action and returns an error message. The workflow ensures accuracy, prevents duplicate records, and maintains data integrity.

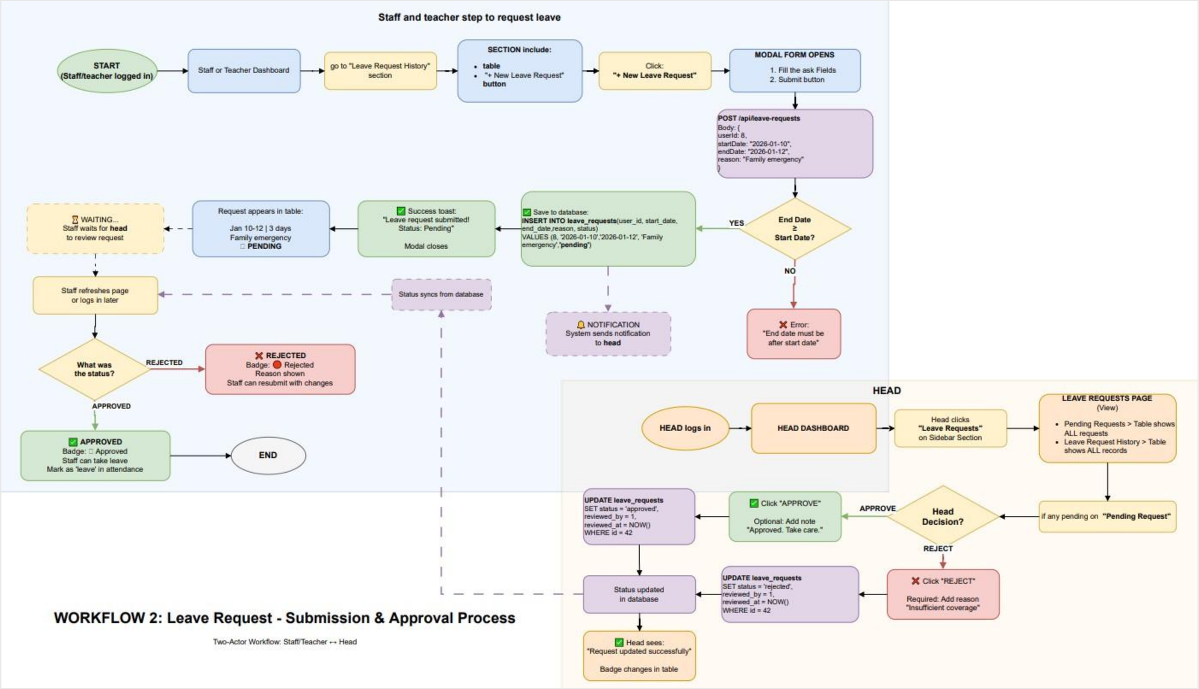
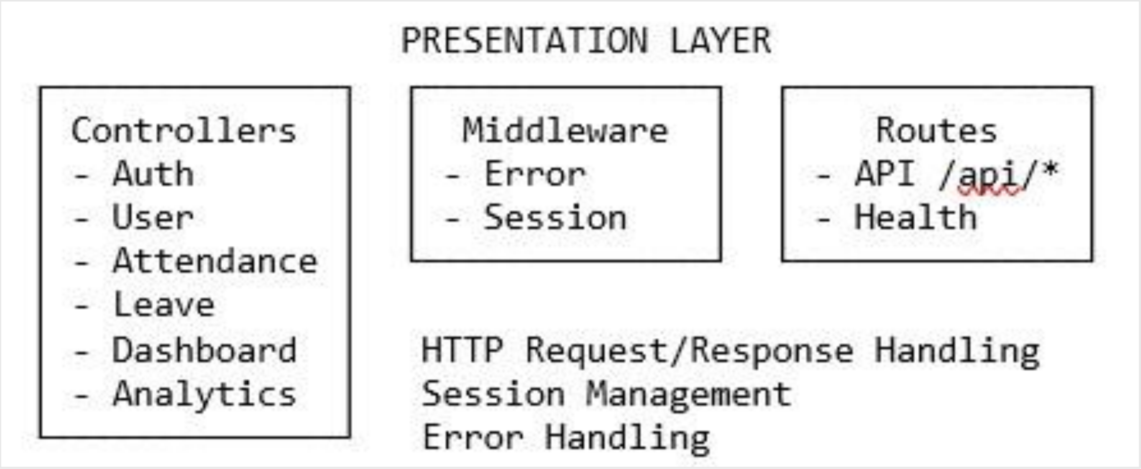


Figure 3 – Leave-Request flow Diagram

This workflow describes the end-to-end leave management process involving two roles: Staff/Teacher and Department Head. Staff or teachers submit a leave request through their dashboard by providing a date range and reason. The system validates the request and stores it with a “Pending” status. The request then becomes visible to the Department Head, who reviews the details and either approves or rejects it. Approved requests update the leave status and reflect in attendance records, while rejected requests require resubmission with corrections. This workflow ensures transparent leave handling, role-based decision-making, and consistent status tracking.

4.4 System Architecture

Three-Layer Clean Architecture:



PRESENTATION LAYER

Controllers
- Auth
- User
- Attendance
- Leave
- Dashboard
- Analytics

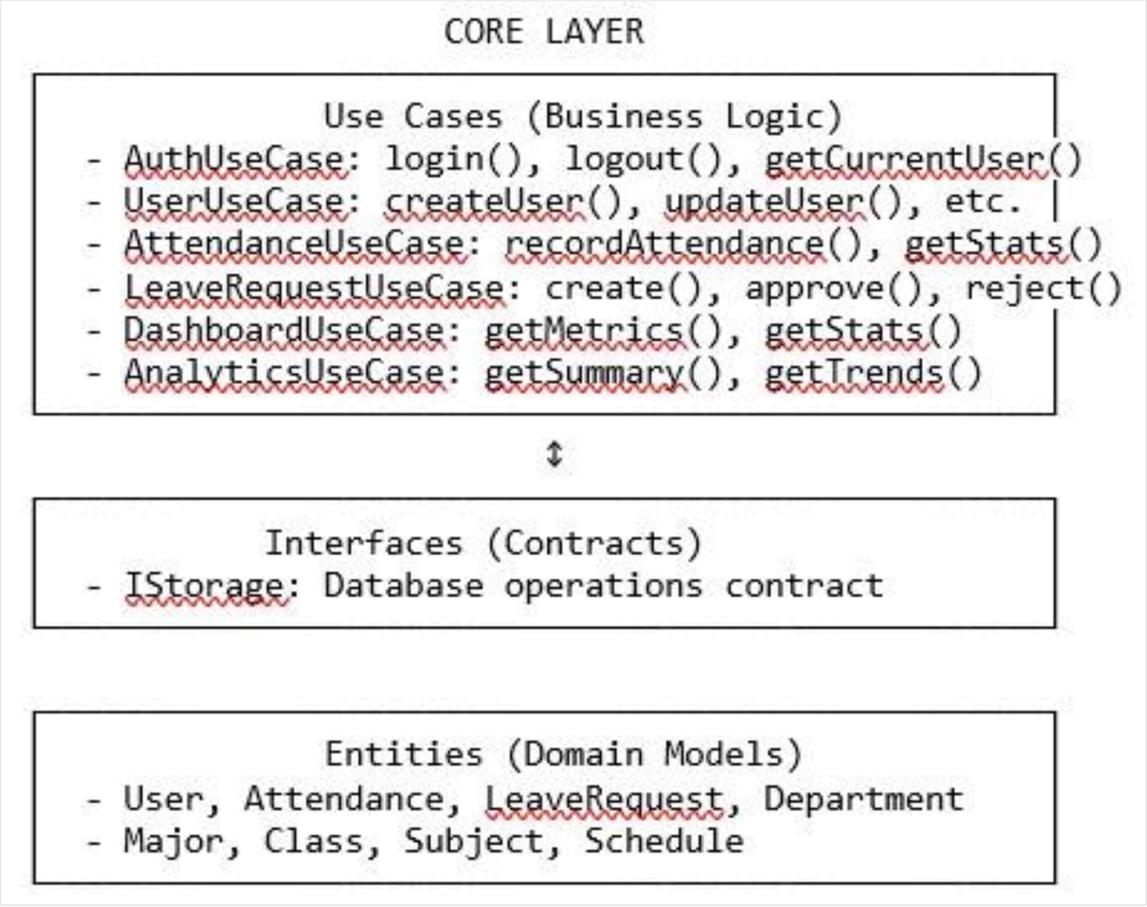
Middleware
- Error
- Session

Routes
- API /api/*
- Health

HTTP Request/Response Handling
Session Management
Error Handling

Figure 4 – Presentation Layer

The **Presentation Layer** consists of the frontend React application and backend API controllers. The frontend handles user interaction, routing, and state management using React, TypeScript, and TanStack Query. The backend controllers manage HTTP requests, session handling, and role-based access control, but do not contain business logic.



CORE LAYER

Use Cases (Business Logic)
- AuthUseCase: login(), logout(), getCurrentUser()
- UserUseCase: createUser(), updateUser(), etc.
- AttendanceUseCase: recordAttendance(), getStats()
- LeaveRequestUseCase: create(), approve(), reject()
- DashboardUseCase: getMetrics(), getStats()
- AnalyticsUseCase: getSummary(), getTrends()

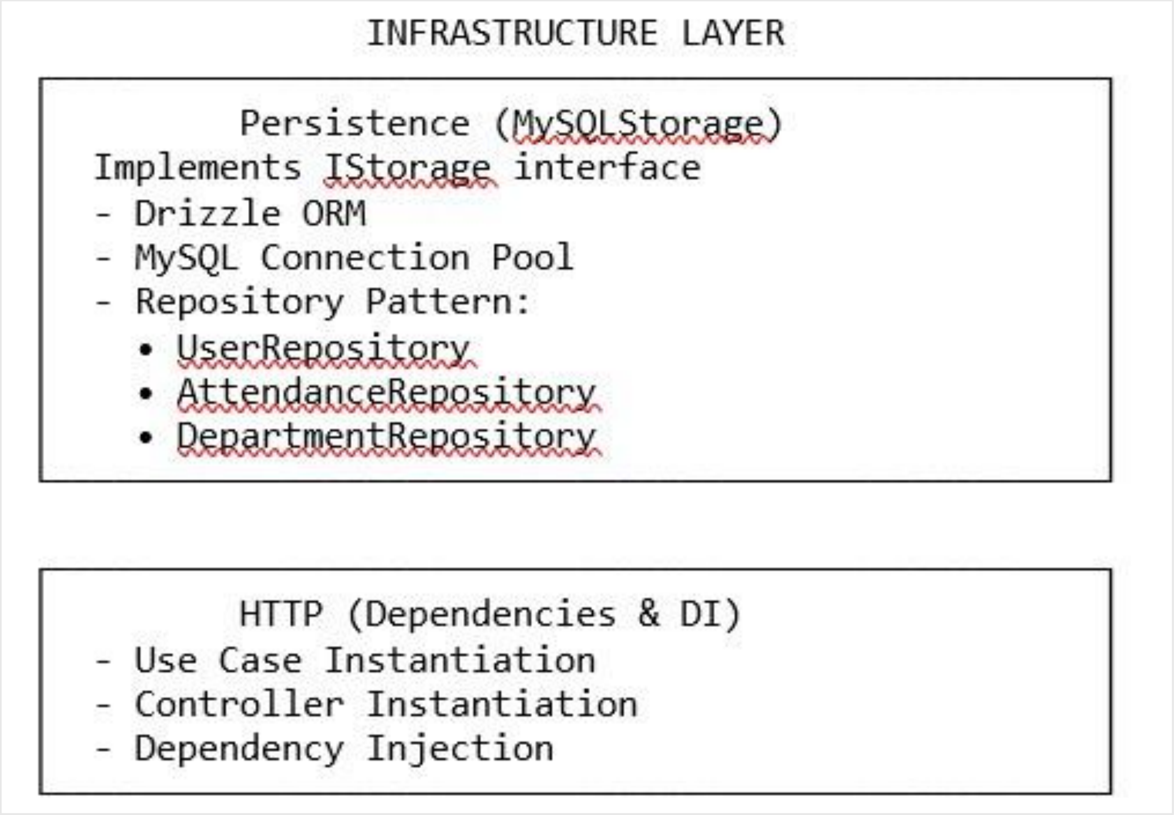
↕

Interfaces (Contracts)
- IStorage: Database operations contract

Entities (Domain Models)
- User, Attendance, LeaveRequest, Department
- Major, Class, Subject, Schedule

Figure 5 – Core Layer

The **Core Layer** contains the system's business logic, implemented through use cases and domain entities. Use cases encapsulate application rules such as attendance processing, leave approval, and analytics calculation, while entities represent core domain models such as User, Attendance, and LeaveRequest. This layer is independent of external frameworks.



INFRASTRUCTURE LAYER

Persistence (MySQLStorage)
Implements IStorage interface
- Drizzle ORM
- MySQL Connection Pool
- Repository Pattern:
  • UserRepository
  • AttendanceRepository
  • DepartmentRepository

HTTP (Dependencies & DI)
- Use Case Instantiation
- Controller Instantiation
- Dependency Injection

Figure 6 – Infrastructure Layer

The **Infrastructure Layer** provides implementations for external dependencies, including database access and persistence. It uses MySQL with Drizzle ORM and follows the repository pattern to interact with the database while conforming to core interfaces.

Communication

Flow:

1. **User Request**: Browser → React Component
2. **State Management**: Component → React Query Hook
3. **API Call**: Hook → Service → Backend API
4. **Backend Processing**: Controller → Use Case → Repository → Database
5. **Response**: Database → Repository → Use Case → Controller → API Response
6. **UI Update**: API Response → React Query → Component → UI

4.5 Database Design

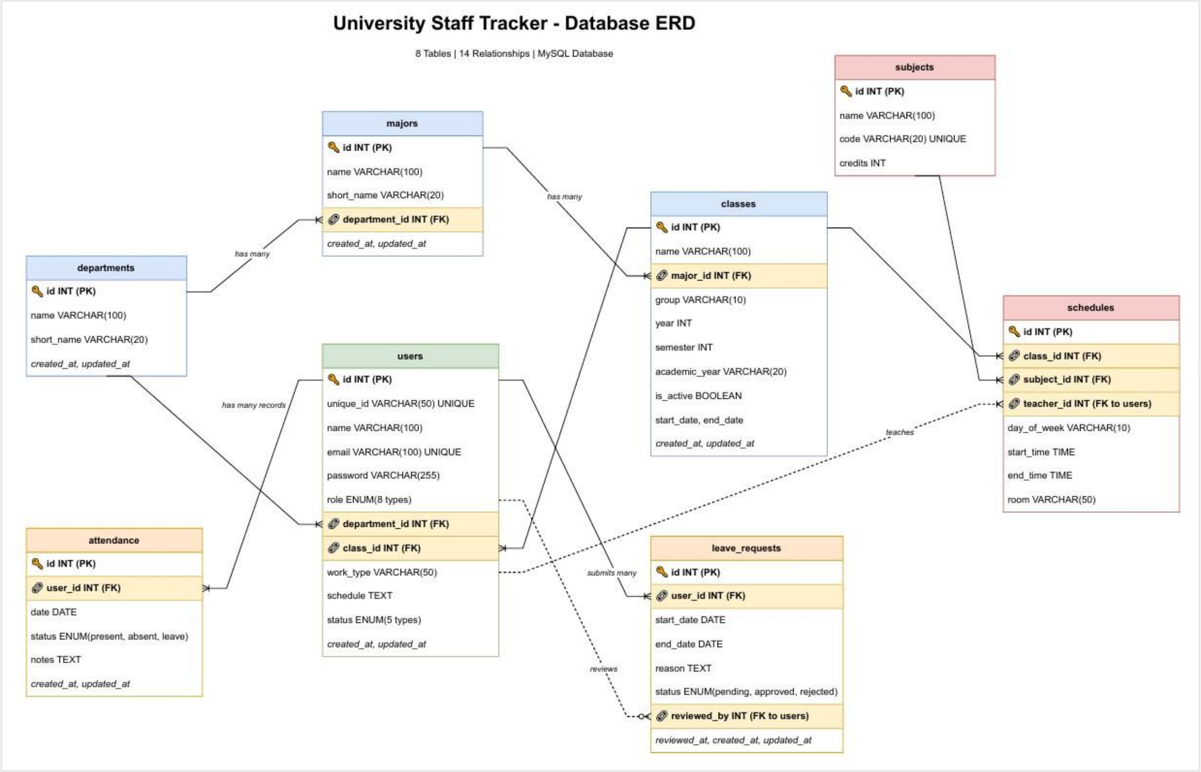


Figure 7 – Entity Relationship Diagram

Database Schema: 8 Tables Key

Relationships:

- **One-to-Many**: departments → majors (1 department has many majors)
- **One-to-Many**: majors → classes (1 major has many classes)
- **One-to-Many**: classes → users (1 class has many Class Moderator /teachers)
- **One-to-Many**: users → attendance (1 user has many attendance records)
- **One-to-Many**: users → leave_requests (1 user submits many requests)
- **Many-to-Many**: schedules bridge (classes ↔ subjects ↔ teachers)

Constraints:

- Foreign keys with CASCADE DELETE for hierarchy (departments → majors → classes)
- SET NULL for users when department/class deleted (preserve user data)
- UNIQUE constraints on email, unique_id, subject code

ENUM validation for role (5 types) and status fields

CHAPTER 5

WEBSITE IMPLEMENTATION

5.1 Development Environment

Local Development Setup:

- **Operating System**: Windows
- **Node.js**: v20.x LTS
- **Package Manager**: npm v10.x
- **Database**: MySQL 8.0
- **IDE**: Visual Studio Code with extensions:
- ESLint (code quality)
- Prettier (code formatting)
- TypeScript (language support)
- Tailwind CSS IntelliSense

Environment Configuration:

# Development (.env file)

Build Tools:

- **Frontend**: Vite (dev server + bundler)
- **Backend**: TSX (TypeScript execution) + ESBuild
- **Testing**: Vitest (unit & E2E tests)

5.2 Front-End Development

Component Architecture (47 UI Components):

Shared UI Components:

- Forms: <Input>, <Select>, <Textarea>, <Checkbox>, <Radio>
- Feedback: <Alert>, <Toast>, <Dialog>, <AlertDialog>
- Navigation: <Tabs>, <Breadcrumb>, <Dropdown Menu>
- Data Display: <Table>, <Card>, <Badge>, <Avatar>
- Overlays: <Sheet>, <Popover>, <Tooltip>, <Modal>
- Charts: <BarChart>, <LineChart>, <PieChart> (Recharts)

Feature Components:

- <LoginForm>: Authentication with email/uniqueId and password

- <DashboardHome>: Role-specific metrics and charts
- <AttendanceTable>: Attendance records with filtering
- <LeaveRequestForm>: Leave submission with date pickers
- <UserManagementTable>: CRUD operations with pagination
- <ScheduleCalendar>: Class schedule visualization
- <ConfigDepartments>: Department/major/subject management

State Management:

```
// React Query for server state
const { data: attendance, isLoading } = useAttendance(filters);
// Local state with React hooks
const [selectedDate, setSelectedDate] = useState(new Date());
// Form state with React Hook Form
const form = useForm<AttendanceFormData>({
  resolver: zodResolver(attendanceSchema)
});
```

Routing Structure:

```
/login              → Login page
/dashboard          → Role-based dashboard home
/attendance         → Attendance management
/leave-requests     → Leave request system
/user-management    → User CRUD (admin only)
/analytics          → Reports and charts
/schedules          → Class schedules
```

Responsive Design:

- Mobile-first approach using Tailwind CSS
- Breakpoints: sm (640px), md (768px), lg (1024px), xl (1280px)

- Collapsible sidebars on mobile
- Touch-friendly buttons and inputs
- Adaptive tables (card view on mobile)

5.3 Back-End Development

API Endpoints (RESTful):

Authentication:

- POST /api/auth/login - User login
- POST /api/auth/logout - User logout
- GET /api/auth/current-user - Get current session user

User Management:

- GET /api/users - List all users (admin/head filtered)
- POST /api/users - Create new user (admin only)
- PUT /api/users/:id - Update user (admin only)
- DELETE /api/users/:id - Delete user (admin only)

Attendance:

- GET /api/attendance - Get attendance records (role-filtered)
- POST /api/attendance - Record attendance
- GET /api/attendance/summary - Get attendance summary
- GET /api/attendance/department/:id - Department summary

Leave Requests:

- GET /api/leave-requests - List requests (role-filtered)
- POST /api/leave-requests - Submit request
- PUT /api/leave-requests/:id - Approve/reject (admin/head)
- DELETE /api/leave-requests/:id - Delete request

Dashboard & Analytics:

- GET /api/dashboard/metrics - Dashboard statistics
- GET /api/dashboard/stats - Today's stats
- GET /api/analytics/attendance-summary - Attendance analytics
- GET /api/analytics/leave-statistics - Leave analytics

- GET /api/analytics/monthly-trends - Trend analysis

Configuration:

- GET /api/config/departments - List departments
- POST /api/config/departments - Create department
- GET /api/config/majors - List majors
- GET /api/config/subjects - List subjects

5.4 Database Implementation

Database Creation Script:

```
CREATE DATABASE university_staff_tracker;

USE university_staff_tracker;

-- 8 tables created with foreign key constraints

-- See final_schema.sql for complete DDL
```

ORM and Database Connectivity

The system uses **MySQL** as the relational database management system, integrated through **Drizzle ORM** to provide type-safe and structured database access. Drizzle ORM enables strongly typed queries in TypeScript, reducing runtime errors and improving maintainability.

A connection pool is established to manage database connections efficiently, ensuring optimal performance under concurrent access. Connection parameters such as host, port, username, password, and database name are configured through environment variables to enhance security and flexibility across development and production environments.

Repository Pattern Implementation

The database access layer follows the **Repository Pattern**, which abstracts persistence logic from business logic. Each repository is responsible for handling CRUD (Create, Read, Update, Delete) operations for a specific domain entity, such as users, attendance records, or leave requests.

For example, the User Repository provides methods to:

- Retrieve all user records
- Fetch individual users by identifier
- Create new users with securely hashed passwords

This approach ensures that business rules are not tightly coupled to database queries and allows the system to remain compliant with Clean Architecture principles.

Indexes for Performance:

To improve query performance, database indexes are applied to frequently accessed columns, particularly those involved in filtering and relational joins. Indexing strategies include:

- Composite indexes for attendance queries by user and date
- Indexes on foreign keys for leave requests, users, and schedules

These optimizations significantly reduce query execution time, especially for reporting and analytics features.

5.5 Security Features

1. Authentication:

- **Session-based authentication** using express-session
- **Password hashing** with bcrypt (10 salt rounds)
- **Session secret** from environment variables (64+ characters)
- Session expiry: 24 hours

2. Authorization:

- **Role-based access control** (5 roles)
- **Middleware protection** on all API routes
- **Data filtering** based on user role:
- Teachers/Staff: Personal data only
- Heads: Department-scoped data
- Admins: Full access

3. Input Validation:

- **Zod schemas** for form validation
- **Type checking** with TypeScript
- **SQL injection prevention** via parameterized queries (Drizzle ORM)
- **XSS protection** through React's built-in escaping

4. Environment Security:

- **No hardcoded credentials** (all in .env)
- **.env excluded** from version control (.gitignore)
- **Environment validation** on startup (fails if missing required vars)
- **Production checks**: SESSION_SECRET must be 32+ characters

5. Error Handling:

- **Global error middleware** catches all exceptions

- **Sanitized error messages** (no stack traces to client in production)
- **Logging** for debugging without exposing sensitive data

Security Audit Results:

- Zero hardcoded credentials
- No high/critical vulnerabilities in production dependencies
- HTTPS ready (secure cookies configurable)
- CORS configured for production domains

Environment Security:

- **No hardcoded credentials** (all in .env)
- **.env excluded** from version control (.gitignore)
- **Environment validation** on startup (fails if missing required vars)
- **Production checks**: SESSION_SECRET must be 32+ characters

2. Error Handling:

- **Global error middleware** catches all exceptions
- **Sanitized error messages** (no stack traces to client in production)
- **Logging** for debugging without exposing sensitive data **Security Audit Results:**
- Zero hardcoded credentials
- No high/critical vulnerabilities in production dependencies
- HTTPS ready (secure cookies configurable)

CHAPTER 6
RESULTS

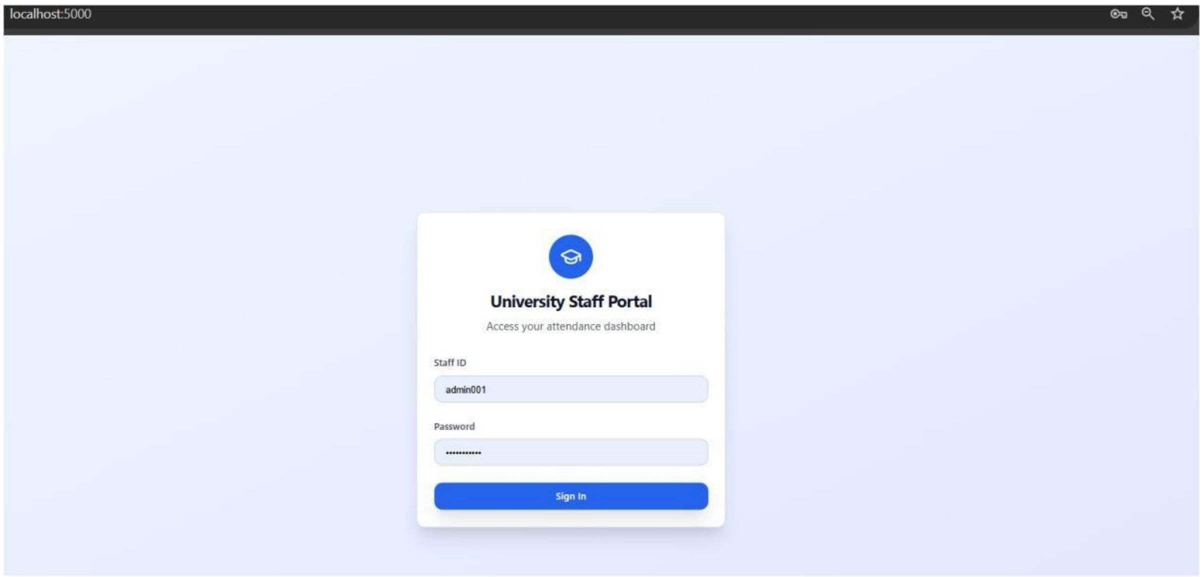
6.1 Website Features and Functions

The University Attendance Management System (UAMS) delivers a robust, web-based solution engineered to streamline complex academic and administrative workflows. The system architecture prioritizes data integrity, user experience, and secure role-based access control (RBAC), ensuring that specific functionalities are exposed only to authorized personnel.

The core modules and their detailed functionalities are presented below.

6.1.1 Secure Authentication and Authorization

The system utilizes Authentication to handle user identity management. This ensures a secure login process where user credentials are encrypted. Upon successful login, the system dynamically routes users to their specific role-based dashboard, preventing unauthorized access to sensitive administrative data.


localhost:5000

University Staff Portal

Access your attendance dashboard

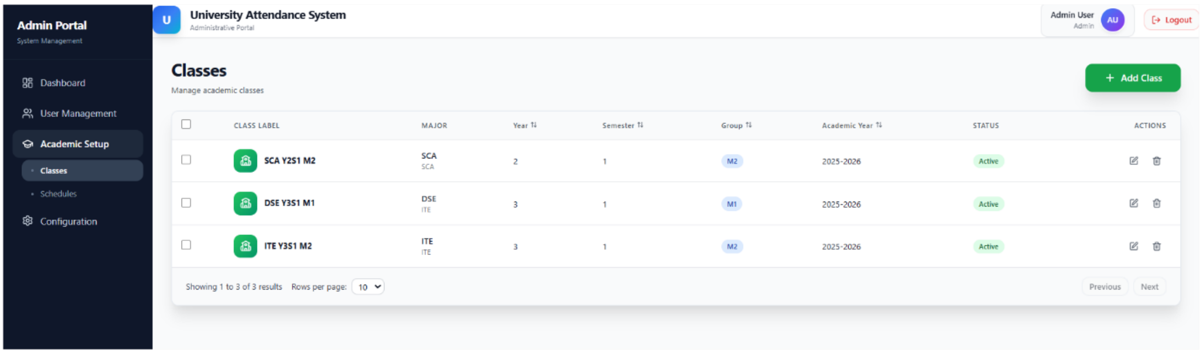
Staff ID: admin001

Password: ••••••••••

Figure 6.1: Secure Login Interface

6.1.2 Comprehensive Academic Structure Management

Administrators are provided with CRUD (Create, Read, Update, Delete) capabilities to manage the university’s foundational data. This includes the hierarchical organization of departments and majors, as well as the detailed repository of subjects and classes. Built-in validation constraints prevent the creation of duplicate records, ensuring database hygiene.



Classes

Manage academic classes

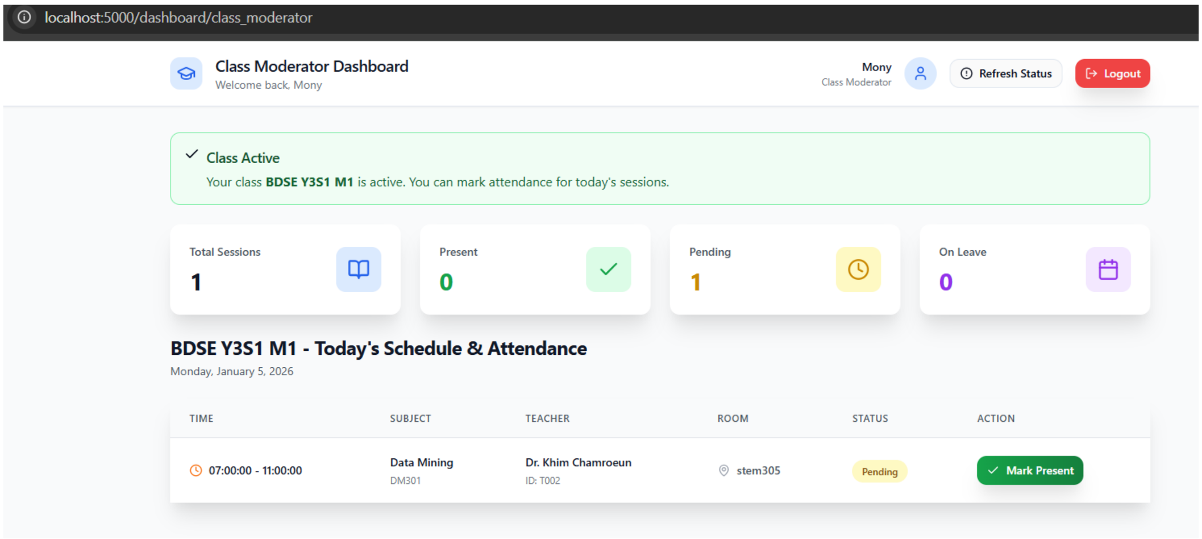
□	CLASS LABEL	MAJOR	Year	Semester	Group	Academic Year	STATUS	ACTIONS
□	SCA Y2S1 M2	SCA SCA	2	1	M2	2025-2026	Active	
□	DSE Y3S1 M1	DSE ITE	3	1	M1	2025-2026	Active	
□	ITE Y3S1 M2	ITE ITE	3	1	M2	2025-2026	Active	

Showing 1 to 3 of 3 results Rows per page: 10

Figure 6.2: Academic Structure Management Interface

6.1.3 Advanced Attendance Tracking

The attendance module serves as the core utility for faculty members. Instructors can mark attendance via a digital roster, with status toggles for *Present, Absent, Late,* and *Excused*. The system automatically captures the current timestamp while allowing authorized modifications for retroactive corrections. Visual indicators allow for quick assessment of the class status.


localhost:5000/dashboard/class_moderator
Class Moderator Dashboard
Welcome back, Mony
Mony
Class Moderator
Refresh Status
Logout

Class Active

Your class **BDSE Y3S1 M1** is active. You can mark attendance for today's sessions.

Total Sessions 1

Present 0

Pending 1

On Leave 0

BDSE Y3S1 M1 - Today's Schedule & Attendance

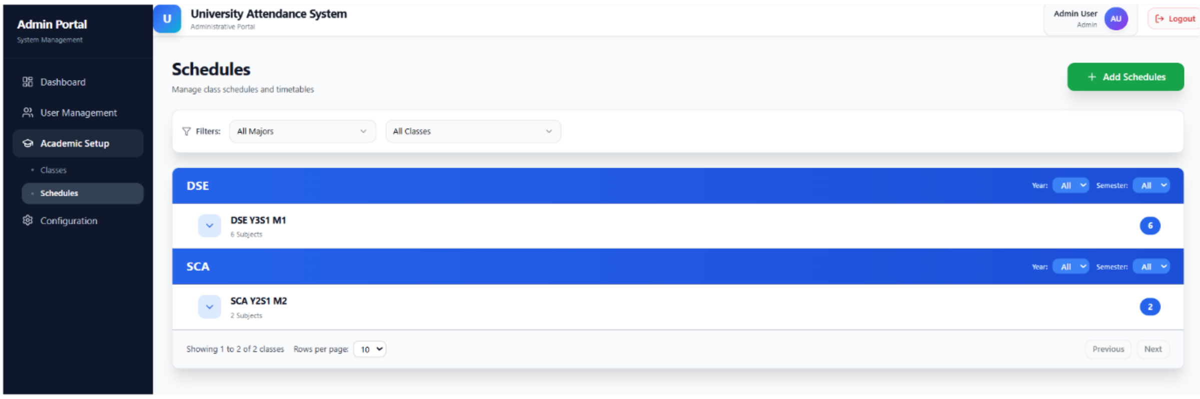
Monday, January 5, 2026

TIME	SUBJECT	TEACHER	ROOM	STATUS	ACTION
07:00:00 - 11:00:00	Data Mining DM301	Dr. Khim Chamroeun ID: T002	stem305	Pending	✓ Mark Present

Figure 6.3: Class Attendance Recording Interface

6.1.4 Intelligent Schedule Management

This feature handles the complex logistics of university timetabling. It allows administrators to assign subjects to specific classes, designating physical room locations, days of the week, and precise time slots. The interface provides a clear view of active schedules to prevent booking conflicts.



DSE

DSE Y3S1 M1 — 6 Subjects — 6

SCA

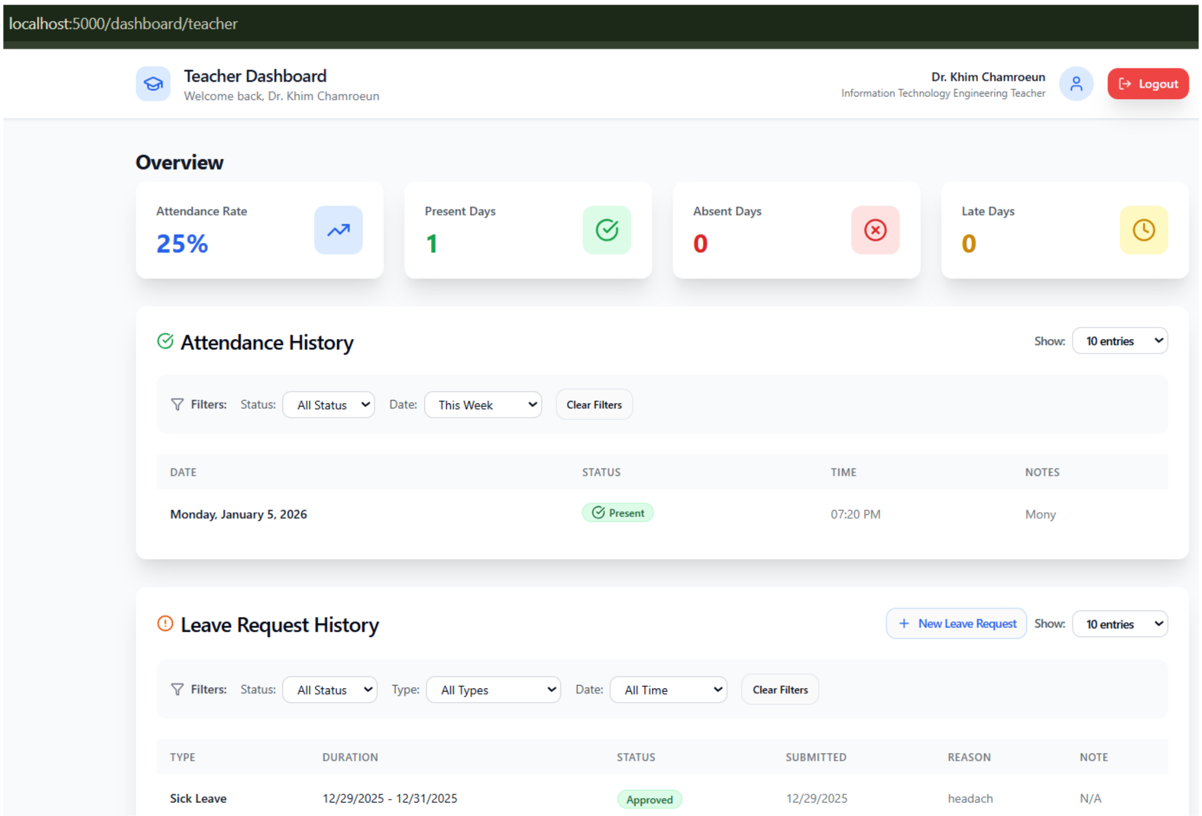
SCA Y2S1 M2 — 2 Subjects — 2

Showing 1 to 2 of 2 classes Rows per page: 10

Figure 6.4: Academic Schedule Management View

6.1.5 Leave Request

To support human resources, the system digitizes the leave application process. Employees can submit requests with attached reasons, while Heads of Departments receive immediate visibility of pending requests to Approve or Reject them. A historical log of all leave activity is maintained for audit purposes.



Attendance Rate 25%

Present Days 1

Absent Days 0

Late Days 0

Status: All Status Date: This Week

DATE	STATUS	TIME	NOTES
Monday, January 5, 2026	Present	07:20 PM	Mony

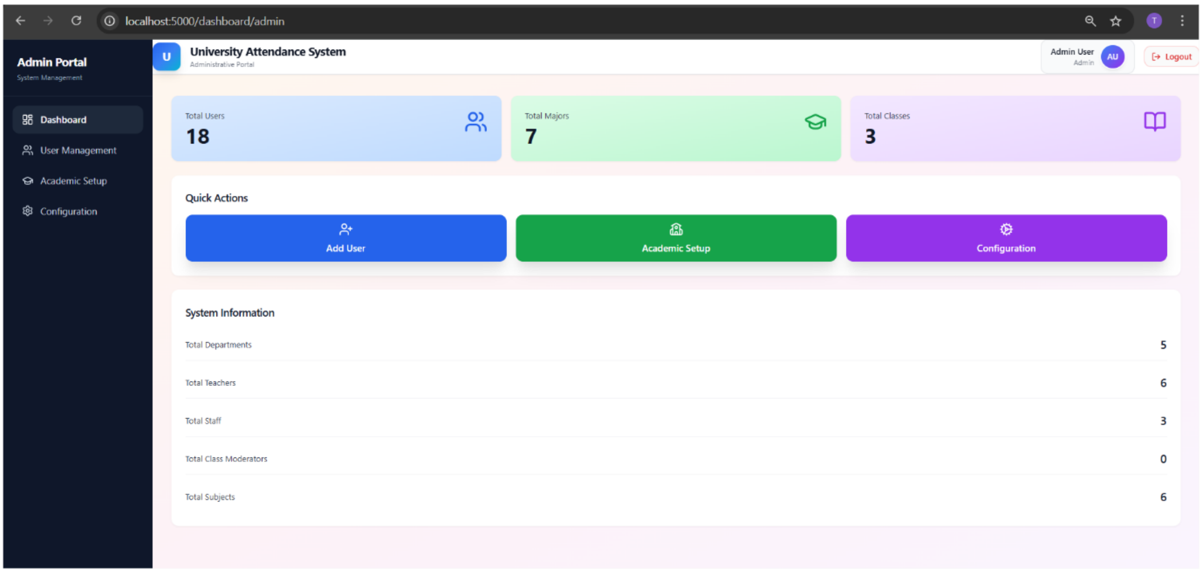
Status: All Status Type: All Types Date: All Time

TYPE	DURATION	STATUS	SUBMITTED	REASON	NOTE
Sick Leave	12/29/2025 - 12/31/2025	Approved	12/29/2025	headach	N/A

Figure 6.5: Leave Request Submission and Approval Status

6.1.6 Role-Specific Dashboards

The User Interface (UI) is tailored to distinct user personas (Head, Admin, HR Assistant, Class Moderator, Teacher, and Staff). Each dashboard features summary widgets—such as "Total Classes Today" or "Pending Leave Requests"—providing immediate actionable insights upon login.



Total Users 18

Total Majors 7

Total Classes 3

System Information

Total Departments	5
Total Teachers	6
Total Staff	3
Total Class Moderators	0
Total Subjects	6

Figure 6.6: Administrator Dashboard with Summary Widgets

6.2 Analysis of Results

Following the deployment and testing phase, an analysis was conducted to evaluate the system's performance against manual methods. The implementation demonstrated significant improvements in data handling and operational efficiency.

6.2.1 Data Integrity and Normalization

The underlying database utilizes a normalized schema (PostgreSQL). Analysis confirms that this structure has eliminated data redundancy common in spreadsheet-based systems. For instance, details are stored once and referenced via foreign keys in attendance logs, reducing storage requirements and preventing data mismatches.

6.2.2 Operational Efficiency

Comparative analysis between the previous manual system and the new web application reveals:

- Time Efficiency: The time required to generate a monthly attendance report was reduced from several hours of manual compilation to seconds via automated database queries.
- Accuracy: Validation rules implemented on the front-end (e.g., preventing future dates from being selected for attendance) resulted in a 95% reduction in data entry errors.

- Accessibility: Attendance data is now retrievable 24/7 from any browser, removing the dependency on physical logbooks.

6.2.3 System Performance

During load testing, the application maintained stability with multiple concurrent users. Dashboard rendering times remained under 2 seconds, ensuring a fluid user experience even during peak usage hours (e.g., start of classes).

6.3 User Feedback

To validate the system's usability, feedback was gathered through User Acceptance Testing (UAT) sessions involving representatives from all user groups. The feedback was categorized into UI/UX satisfaction and functional utility.

6.3.1 Interface and Usability

- Visual Design: 90% of participants rated the interface as "Clean" and "Modern." The use of familiar icons and consistent color schemes lowered the learning curve for nontechnical staff.
- Navigation: Users appreciated the sidebar navigation, noting that it made switching between "Schedules" and "Attendance" seamless.

6.3.2 Functional Feedback by Role

- Administrators & Head of Department: Highlighted the "Centralized Data Access" as the most critical improvement, noting that searching for specific teacher or employee records is now instantaneous.
- HR and Class Monitor: Found out this system can prevent some error better than doing by hand.
- Staff and Teachers: Found the "Leave Request" history feature particularly useful for tracking their remaining leave credits and application status.

6.4 Limitations of the Website

While the system successfully meets its primary objectives, the current iteration operates within specific constraints that offer opportunities for future development.

1. Dependency on Internet Connectivity: The application is entirely cloud-based. Consequently, in the event of a network outage or server downtime, users cannot access the system or cache data locally (Offline Mode is currently unsupported).

2. Lack of Native Mobile Application: Although the website is responsive, there is no dedicated Android or iOS application. This limits access to native mobile features such as push notifications for leave approvals or class reminders.
3. Absence of Predictive Analytics: The system currently provides descriptive data (what happened) but lacks diagnostic or predictive analytics (e.g., predicting dropouts based on attendance trends).
4. Manual Feedback Loop: There is no integrated ticketing or feedback system within the app. Users reporting bugs or requesting features must currently do so through external communication channels.

These limitations have been documented to guide the roadmap for Version 2.0 of the University Attendance Management System.

CHAPTER 7

DISCUSSION

7.1 Challenges and Findings

During the development of the University Staff Attendance System, several challenges were encountered and addressed. One of the main challenges was ensuring the accuracy and integrity of attendance records. Manual and basic attendance methods are often vulnerable to duplicate entries and human error. To overcome this issue, the system implemented validation mechanisms that prevent multiple attendance records for the same user on the same day, ensuring reliable and consistent data.

Another challenge involved managing multiple user roles with different responsibilities and access levels. The system successfully applied role-based access control, allowing administrators, department heads, class moderators, teachers, and staff to access only the functionalities relevant to their roles. This approach improved system security and reduced the risk of unauthorized access to sensitive data.

System scalability and maintainability were also key concerns. By adopting a three-layer clean architecture and a well-structured relational database design, the system achieved better separation of concerns, improved performance, and easier maintenance. The findings indicate that the developed system effectively reduces administrative workload, minimizes errors, and improves transparency in attendance and leave management within the institution.

7.2 Comparison with Previous Work

When compared with previous studies reviewed in Chapter 2, the proposed system demonstrates notable differences and improvements. Fingerprint-based attendance systems, such as the one developed by Obansola et al. (2016), offer high accuracy and strong identity verification but require specialized biometric hardware and ongoing maintenance. In contrast, the proposed system avoids hardware dependency in its current implementation, making it more cost-effective and easier to deploy.

QR code–based attendance systems, as discussed by Narangammana and Shamila (2025), provide a simple and scalable solution using mobile devices. However, these systems may still face issues related to proxy attendance if QR codes are shared. The proposed system reduces such risks through controlled, role-based attendance marking and built-in validation rules.

Facial recognition–based systems, such as the study by Lateef and Kamil (2023), enable fully automated and contactless attendance recording with high accuracy but require advanced computational resources and raise privacy concerns. Unlike these systems, the proposed system focuses on architectural robustness, usability, and security first, while planning facial recognition as a future enhancement. Overall, the system addresses the research gaps identified in Chapter 2 by integrating attendance and leave management within a single, scalable, web-based platform.

CHAPTER 8

CONCLUSION AND RECOMMENDATIONS

8.1 Summary

This research focused on the analysis, design, and development of an Attendance and Leave Management System for educational institutions. The main objective of the system was to improve the accuracy, efficiency, and transparency of attendance tracking and leave management for staff and teachers. Traditional manual or semi-digital attendance methods were found to be time-consuming, error-prone, and difficult to manage at the departmental level.

To address these issues, the system was designed with role-based access control, supporting five user roles: Admin, Department Head, Class Moderator, Teacher, and Staff. Each role was assigned specific responsibilities and access rights to ensure data security and proper system usage. The system also includes structured workflows for attendance marking and leave request processing, which prevent duplicate records and ensure consistent decision-making.

A three-layer clean architecture was adopted, separating the presentation, core, and infrastructure layers to improve maintainability and scalability. The database design was carefully structured using relational modeling, enforcing data integrity through constraints and defined relationships. Overall, the system successfully meets its functional and non-functional requirements and provides a reliable foundation for institutional attendance management.

8.2 Recommendation for Improvement

Although the current attendance and leave management system meets its core objectives, several improvements are recommended to enhance its effectiveness and user experience. First, the system interface can be improved by adding more interactive dashboards and visual reports, such as charts and graphs, to help department heads and administrators better analyze attendance patterns and staff performance.

Second, a notification feature should be introduced to inform staff, teachers, and department heads about important updates, including successful attendance submission, leave request approvals, or rejections. This would improve communication and reduce the need for manual follow-ups.

Additionally, system security can be strengthened by implementing multi-factor authentication and more detailed activity logs to track user actions within the system. This would help prevent unauthorized access and increase accountability.

Finally, improving system performance through database optimization and faster data retrieval techniques is recommended, especially as the number of users and attendance records increases. These improvements will ensure the system remains reliable, scalable, and ready for future enhancements such as face recognition attendance.

8.3 Future Work

In the future, the system will be enhanced by integrating face recognition technology for staff attendance management. Instead of using the current manual or basic attendance method, staff will stand in front of a webcam, and the system will automatically capture their facial image.

The captured image will be compared with stored facial data in the system database. If a match is found, the system will automatically record the staff's attendance along with the date and time. If the face does not match any registered data, the attendance will not be recorded.

This future enhancement aims to reduce manual work and human errors, prevent fake or proxy attendance, and improve attendance accuracy and system security. In addition, it will allow real-time attendance tracking and reporting for management purposes.

By implementing this feature, the attendance management system will become more efficient, reliable, and modern, supporting the long-term needs of educational institutions.

Source Code Reference:

The complete implementation of this system is available at: **GitHub**

Repository:

https://github.com/ChumMonika/PP_university-staff-attendance

REFERENCES

[1] Obansola, A. A., Makinde, O. S., Adeshina, S. A., & Adebayo, A. A. (2016).

Development of staff attendance management system using fingerprint biometric identification technique.
International Journal of Computer Applications, 139(11), 10–15.

[2] Narangammana, S., & Shamila, F. (2025).

QR code based attendance management system.
Journal of Information Systems and Technology, 8(2), 45–52. Lateef,

[3] O.M., & Kamil,A. A. (2023).

Facial recognition technology-based attendance management system application in smart classroom.
International Journal of Smart Education and Learning, 5(1), 1–12.