**ATTENDANCE MANAGEMENT SYSTEM**

**A PROJECT REPORT**

**CSA0937-PROGRAMMING IN JAVA AND ANALYSIS**

***Submitted by***

**N.Vamsi Vardhan Reddy**

**(192110209)**

***In partial fulfilment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING IN**

**COMPUTER SCIENCE**

****

**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA NAGAR, THANDALAM,**

**SIMATS, CHENNAI – 602105.**

**MARCH 2024**

**BONAFIDE CERTIFICATE**

This is to certify that the project report entitled “Attendance Management System” submitted by “N.Vamsi Vardhan Reddy (192110209)”, to Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, is a record of bonafide work carried out by him/her under my guidance. The project fulfills the requirements as per the regulations of this institution and in my appraisal meets the required standards for submission.

**Dr.M.Buvanesvari**

Professor

Department of Knowledge

Engineering,

Saveetha School of Engineering

SIMATS, Chennai – 602 105

Internal examiner External Examiner

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTENTS** | **PAGE NO** |
| 1 | ABSTRACT | 4 |
| 2 | INTRODUCTION | 5 |
| 3 | DESCRIPTION | 6 |
| 4 | SYSTEM REQUIREMENTS | 7 |
| 5 | EXISTING WORK | 8 |
| 6 | PROPOSED WORK | 9 |
| 7 | TECHNOLOGY USED | 10 |
| 8 | USE CASE DIAGRAM | 11 |
| 9 | SOURCE CODE | 13 |
| 10 | SCREENSHOTS(OUTPUTS) | 18 |
| 11 | CONCLUSION&FUTURE ENHANCEMENTS | 21 |
| 12 | REFERENCES | 22 |

**ABSTRACT**

The evolution of technology has significantly impacted various aspects of organizational management, including the traditional method of attendance tracking. This abstract presents an innovative solution: an Automated Attendance Management System (AMS) designed to streamline and enhance the efficiency of attendance monitoring in diverse settings. The proposed system leverages cutting-edge technologies such as biometrics, RFID, and cloud computing to overcome the limitations of manual attendance systems.

The Automated Attendance Management System aims to eliminate the tedious and error-prone processes associated with manual attendance tracking. By implementing biometric authentication methods such as fingerprint recognition or facial recognition, the system ensures accurate and secure identification of individuals. Additionally, RFID technology enables seamless tracking of attendance without direct physical contact, promoting a hygienic and convenient solution, especially in the context of the global health landscape.

The integration of cloud computing facilitates real-time data synchronization and accessibility, enabling administrators and stakeholders to monitor attendance remotely. The system also incorporates advanced analytics tools, providing valuable insights into attendance patterns, trends, and potential areas for improvement. Such data-driven insights empower organizations to make informed decisions, optimize resource allocation, and enhance overall operational efficiency.

**INTRODUCTION**

The aims and objectives of an Attendance Management System (AMS) project typically revolve around addressing specific challenges in attendance tracking and promoting efficiency, accuracy, and convenience.

* Enable real-time monitoring of attendance.
* Provide timely insights for prompt decision-making.
* Implement cloud-based storage for remote accessibility.
* Facilitate access to attendance data from anywhere, promoting flexibility.
* Develop analytics tools for in-depth attendance data analysis.
* Generate meaningful insights for informed decision-making.
* Design user-friendly interfaces for administrators and end-users.
* Ensure widespread adoption by simplifying attendance management processes.
* Integrate the system with existing organizational platforms.

**DESCRIPTION**

The background of an Attendance Management System project involves an understanding of the existing challenges and inefficiencies in manual attendance tracking processes, prompting the need for an automated solution. Common elements in the background of an AMS project include.

Traditional attendance tracking relies on manual methods, such as paper-based sign-in sheets, which are time-consuming, error-prone, and susceptible to inaccuracies.Manual attendance systems are vulnerable to errors, including unintentional mistakes in data entry and fraudulent practices like proxy attendance, leading to inaccurate records.

Managing attendance manually consumes significant administrative resources, particularly in large organizations, resulting in increased operational costs and potential delays in decision-making.Traditional methods lack the capability for real-time monitoring, making it challenging for administrators to promptly address attendance-related issues or make informed decisions based on current data. With sensitive attendance data being handled manually, concerns about data privacy and security become pertinent, necessitating a more secure and controlled approach.The advancements in biometrics, and cloud computing technologies offer opportunities to revolutionize attendance tracking, providing more accurate, efficient, and secure alternatives.As organizations expand, manual attendance tracking becomes increasingly impractical. An automated system is needed to handle the growing workforce and diverse attendance management requirements.Events such as global health crises may highlight the need for contactless attendance systems, promoting the adoption of technology-driven solutions to ensure a safe and hygienic work environment.

**SOFTWARE REQUIREMENTS**

* Server
* 4 GB RAM or more
* Depending on the size of the database, 50 GB or more
* Client (for web-based systems):
* Modern web browser with JavaScript enabled
* Software Requirements:
* Linux (e.g., Ubuntu, CentOS) or Windows Server
* Windows, macOS, Linux
* Database Management System:
* MySQL, PostgreSQL, MongoDB, or other databases based on system architecture
* Web Server
* Apache, Nginx
* Depends on the technology stack (e.g., PHP, Python, Java, .NET)
* Frameworks (if applicable):
* Laravel, Django, Spring, .NET Core, etc.
* Internet connection (for cloud-based systems or remote access)
* Local area network (LAN) for on-premises systems
* Firewalls and security measures to protect data
* User authentication and authorization
* Regular security updates and patches

**EXISTING**

* A comprehensive school management system that includes attendance tracking for students.
* Provides features for managing student information, grades, and communication.
* A cloud-based student information system with attendance tracking capabilities.
* Offers features for grading, communication, and collaboration.
* A mobile and web-based solution for schools and colleges.
* Includes attendance management, timetable creation, and communication tools.
* An open-source Learning Management System (LMS) that may include attendance tracking features.
* Designed for educational content delivery and collaboration.
* A web-based platform for school management.
* Offers attendance tracking, grade management, and communication tools.
* A cloud-based school management system with attendance tracking capabilities
* A school management system with attendance tracking, timetable creation, and academic management features.
* Offers a cloud-based solution for educational institutions.

**PROPOSED :**

* [Attendance is carried out in handwritten registers](https://itsourcecode.com/fyp/existing-system-chapter-4-doc-attendance-monitoring-system/).
* [It requires more human effort and is prone to errors such as misplaced sheets and illegible handwriting](https://leadschool.in/blog/student-attendance-management-system-decoding-the-scope-objectives/).
* [It can be a tedious job to maintain the record](https://itsourcecode.com/fyp/existing-system-chapter-4-doc-attendance-monitoring-system/).
* [The system requires more human effor](https://itsourcecode.com/fyp/existing-system-chapter-4-doc-attendance-monitoring-system/)t.
* [It streamlines the recording of attendance by automating the process3](https://leadschool.in/blog/student-attendance-management-system-decoding-the-scope-objectives/).
* [It eradicates the need for manual paperwork](https://leadschool.in/blog/student-attendance-management-system-decoding-the-scope-objectives/).
* [It offers real-time insights into student attendance trends3](https://leadschool.in/blog/student-attendance-management-system-decoding-the-scope-objectives/).
* [Administrators, teachers, and parents have digital access to attendance reports and student attendance records](https://leadschool.in/blog/student-attendance-management-system-decoding-the-scope-objectives/).
* [It reduces the time and effort required for attendance-taking while ensuring accuracy](https://leadschool.in/blog/student-attendance-management-system-decoding-the-scope-objectives/).
* [The proposed system covers the problem of using the manual system such as an unsecured record of attendance, lots of paper works, and inaccuracy of the data inputs](https://itsourcecode.com/fyp/proposed-solution-chapter-5-attendance-monitoring-system-doc-2019/).
* [This proposed system will benefit the school for its development and improvement by using a new technology and improved system for the day-to-day process of attendance](https://itsourcecode.com/fyp/proposed-solution-chapter-5-attendance-monitoring-system-doc-2019/).

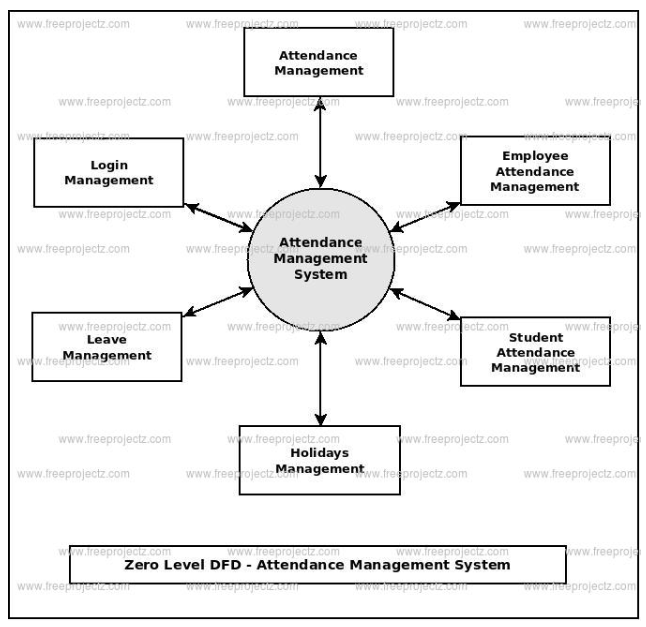
**TECHNOLOGY USED**

An attendance management system implemented solely in Java utilizes a robust technology stack for seamless tracking and organization of attendance data. The backend logic can be crafted with Java, employing frameworks like Spring Boot for efficient development and management of business logic.

For data storage and retrieval, a Java application might integrate with relational databases such as MySQL or PostgreSQL, or even leverage non-relational databases like MongoDB, depending on specific requirements. The user interface, developed using JavaServer Faces (JSF) or other Java-based UI frameworks, ensures a responsive and user-friendly experience. Security features are implemented using Java's built-in capabilities and additional libraries if needed.

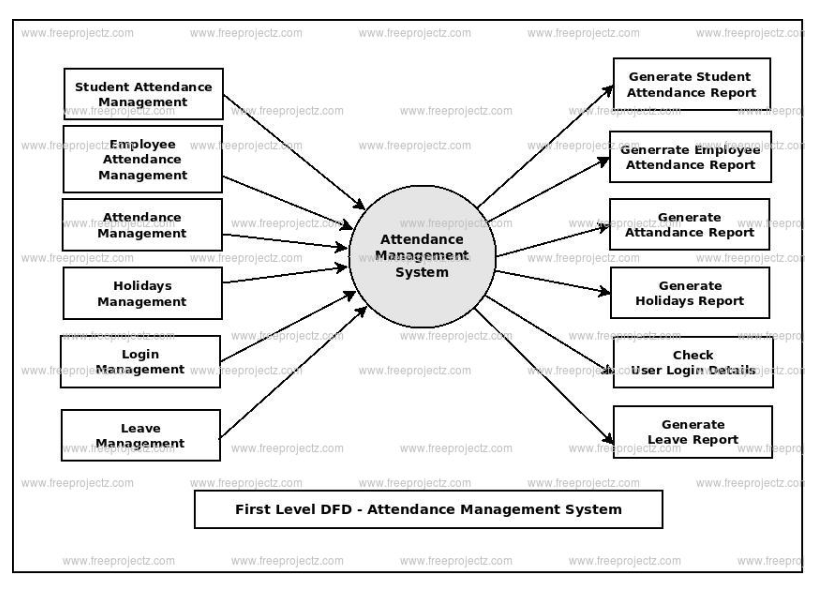
The system can leverage Java's multithreading capabilities for concurrent processing, enhancing performance. Integration with other systems or services can be achieved through Java APIs or web services. Mobile applications may be developed using Java for Android platforms. Overall, a Java-centric approach ensures a robust and scalable architecture for an attendance management system.

**USE CASE DIAGRAMS**

****

Attendance Management System shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an extemal agent, and which together provide all of the functionality of the Attendance Management System system as a whole. It also identifies internal data stores of Authentication, Login, Leave, Holidays, Student Attendance that must be present in order for the Attendance system to do its job, and shows the flow of data between the various parts of Attendance, Student Attendance, Login, Authentication, Leave of the system. DFD Level 1 provides a more detailed breakout of pieces of the 1st level DFD. You will highlight the main functionalities of Attendance.

**DATA FLOW DIAGRAM OF STUDENT**



Admin logins to the system and manage all the functionalities of Attendance Management System

Admin can add, edit, delete and view the records of Attendance, Student Attendance, Leave.

Admin can manage all the details of Employee Attendance, Holidays, Login

Admin can also generate reports of Attendance, Employee Attendance, Student Attendance, Holidays, Leave, Login

Admin can search the details of Employee Attendance, Leave, Login

Admin can apply different level of filters on report of Attendance, Holidays, Leave

Admin can tracks the detailed information of Employee Attendance, Student Attendance, Holidays.

**SOURCE CODE:**

import javax.swing.\*;

import java.awt.event.\*;

import java.util.\*;

import java.text.\*;

class Student {

String name;

int rollNumber;

boolean isPresent;

public Student(String name, int rollNumber) {

this.name = name;

this.rollNumber = rollNumber;

this.isPresent = true;

}

}

class AttendanceManagementSystem extends JFrame implements ActionListener {

Map<String, String> userCredentials = new HashMap<>();

Map<Integer, Student> students = new HashMap<>();

JLabel usernameLabel, passwordLabel, dateLabel;

JTextField usernameField;

JPasswordField passwordField;

JButton loginButton, displayButton;

private boolean isLoggedIn = false;

public AttendanceManagementSystem() {

userCredentials.put("abrar", "abrar786");

students.put(101, new Student("Abrar", 192124048));

students.put(102, new Student("Nagendra", 192124050));

students.put(103, new Student("Nawaz", 192124051));

students.put(104, new Student("Naveen", 192124010));

students.put(105, new Student("Bhaskar", 192110578));

usernameLabel = new JLabel("Username:");

usernameLabel.setBounds(20, 20, 80, 20);

add(usernameLabel);

usernameField = new JTextField();

usernameField.setBounds(100, 20, 200, 20);

add(usernameField);

passwordLabel = new JLabel("Password:");

passwordLabel.setBounds(20, 50, 80, 20);

add(passwordLabel);

passwordField = new JPasswordField();

passwordField.setBounds(100, 50, 200, 20);

add(passwordField);

loginButton = new JButton("Login");

loginButton.setBounds(20, 80, 120, 30);

loginButton.addActionListener(this);

add(loginButton);

displayButton = new JButton("Display Students");

displayButton.setBounds(150, 80, 150, 30);

displayButton.addActionListener(this);

displayButton.setEnabled(false); // Initially disabled

add(displayButton);

dateLabel = new JLabel();

dateLabel.setBounds(20, 110, 300, 20);

add(dateLabel);

setSize(350, 180);

setLayout(null);

setVisible(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

// Update date label

updateDate();

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == loginButton) {

String username = usernameField.getText();

String password = new String(passwordField.getPassword());

if (userCredentials.containsKey(username) && userCredentials.get(username).equals(password)) {

JOptionPane.showMessageDialog(this, "Login successful.");

clearFields();

isLoggedIn = true;

displayButton.setEnabled(true);

} else {

JOptionPane.showMessageDialog(this, "Invalid username or password. Please try again.");

isLoggedIn = false;

displayButton.setEnabled(false);

}

} else if (e.getSource() == displayButton && isLoggedIn) {

takeAttendanceInput();

displayAttendanceSummary();

} else {

JOptionPane.showMessageDialog(this, "Please login first.");

}

}

private void clearFields() {

usernameField.setText("");

passwordField.setText("");

}

private void takeAttendanceInput() {

for (Student student : students.values()) {

String input;

boolean validInput;

do {

input = JOptionPane.showInputDialog(this, "Is " + student.name + " present? (yes/no)");

validInput = input != null && (input.equalsIgnoreCase("yes") || input.equalsIgnoreCase("no"));

if (!validInput) {

JOptionPane.showMessageDialog(this, "Invalid input. Please enter 'yes' or 'no'.");

}

} while (!validInput);

student.isPresent = input.equalsIgnoreCase("yes");

}

}

private void displayAttendanceSummary() {

StringBuilder presentStudents = new StringBuilder("Present Students:\n");

StringBuilder absentStudents = new StringBuilder("Absent Students:\n");

int presentCount = 0;

int absentCount = 0;

int totalStudents = students.size(); // Total number of students

// Get the current date

DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

Date date = new Date();

String currentDate = dateFormat.format(date);

for (Student student : students.values()) {

if (student.isPresent) {

presentStudents.append(student.name).append("\n");

presentCount++;

} else {

absentStudents.append(student.name).append("\n");

absentCount++;

}

}

String summaryMessage = "Date: " + currentDate + "\n\n" +

presentStudents.toString() + "\n" + absentStudents.toString() + "\n"

+ "Total Present: " + presentCount + "\nTotal Absent: " + absentCount

+ "\nTotal Students: " + totalStudents; // Include total students

JOptionPane.showMessageDialog(this, summaryMessage, "Attendance Summary", JOptionPane.PLAIN\_MESSAGE);

}

private void updateDate() {

DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

Date date = new Date();

dateLabel.setText("Date: " + dateFormat.format(date));

}

public static void main(String[] args) {

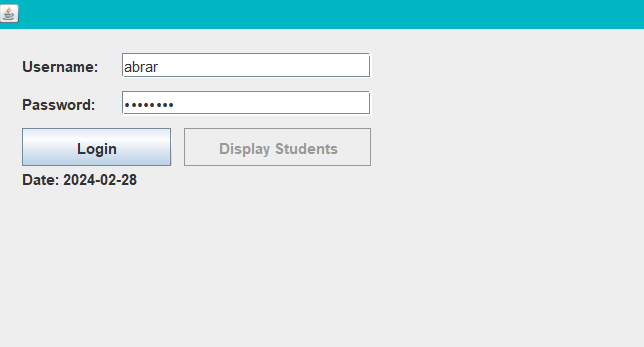
new AttendanceManagementSystem();

}

}

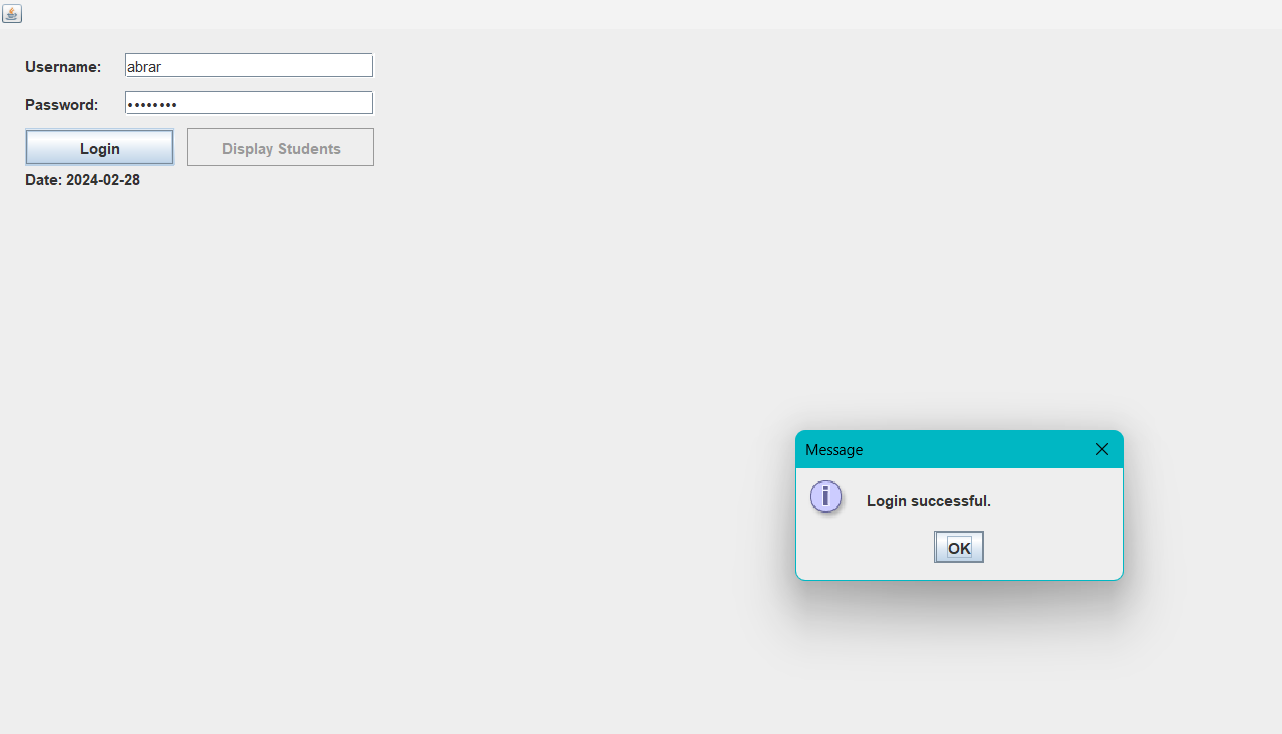
**OUTPUT:**

**Fig 1** The interface of an attendance app typically features a login screen with fields for entering the username and password, ensuring secure and personalized access. The username field allows users to input their unique identification, often assigned by the system administrator.

****

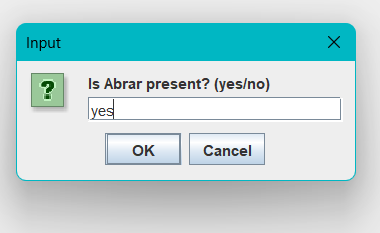
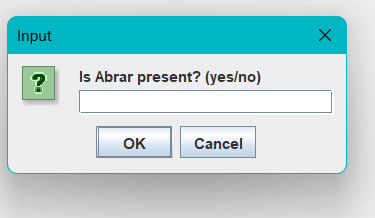
**Fig 1**

**Fig 2** After a successful login into the attendance app, users are seamlessly transitioned into the main interface, unlocking a myriad of functionalities tailored to meet their specific needs.

****

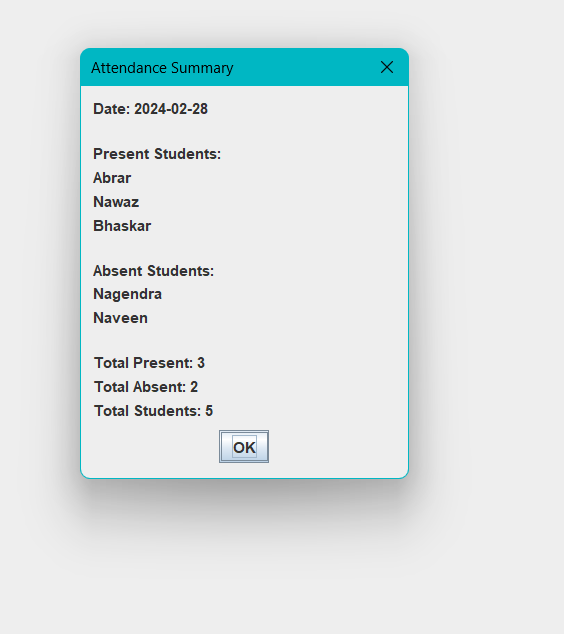
**Fig 2**

**Fig 3** To mark attendance present or absent

****

**Fig 3**

**Fig 4** The system computes the number of present and absent individuals based on the attendance records for a specific class session or a defined time period. The interface may display a dashboard or a specific attendance management page where they can view the overall attendance statistics.

****

**Fig 4**

**CONCLUSION**

The implementation of the proposed Attendance Management System marks a significant leap forward from the limitations of manual attendance tracking. By automating the process, integrating advanced technologies such as biometrics and RFID, and leveraging cloud computing for real-time access, the system addresses the inefficiencies, inaccuracies, and security concerns associated with traditional methods. The enhanced user interfaces, analytics tools, and notification systems contribute to a more streamlined, efficient, and secure attendance management process.

The shift towards automation not only improves the accuracy of attendance records but also provides a foundation for data-driven decision-making. The system's scalability ensures adaptability to organizational growth, making it a sustainable solution for evolving workforce needs. The integration with existing systems facilitates seamless data flow across different organizational departments, promoting consistency and reducing manual efforts.

**FUTURE ENHANCEMENT**

Explore the integration of Internet of Things (IoT) devices for more comprehensive and context-aware attendance tracking, allowing for additional data points and insights.

Implement machine learning algorithms to analysis historical attendance data and predict future attendance trends, enabling proactive management strategies.

Further develop and optimize the mobile application, providing additional features such as geolocation-based attendance marking for field-based employees.

Investigate the use of blockchain technology to enhance the security of attendance records, ensuring tamper-proof and transparent data management. Integrate the Attendance Management System with smart office technologies for a cohesive workplace experience, including automated room access based on attendance.

**REFERENCES**

extension://fheoggkfdfchfphceeifdbepaooicaho/html/site\_status\_block\_page.html

* <https://code-projects.org/attendance-management-system-in-java-with-source-code/>
* <https://sourcecodehero.com/attendance-management-system-project-in-java-with-source-code/>
* <https://itsourcecode.com/free-projects/java-projects/attendance-management-system-project-in-java-with-source-code/>
* https://myprojectideas.com/online-attendance-management-system-java-project/