

Code Cache: Attendance & Geofencing System

Prabin Kumar Das

Student, Dept. of CSE,
GIFT Autonomous, Bhubaneswar

Soumya Prakash Behera

Student, Dept. of CSE,
GIFT Autonomous, Bhubaneswar

Dr. Soumendra Prasad Rout

Assistant Professor, Dept. of CSE,
GIFT Autonomous, Bhubaneswar

Abstract— The rapid advancement of digital workplace technologies and web-based security systems has significantly increased the demand for intelligent, real-time attendance and personnel tracking solutions. Traditional attendance systems often rely on manual entry or simple biometric devices that lack location verification, geospatial intelligence, and multi-layered security enforcement. This paper presents the design and implementation of Code Cache, a high-security, real-time Attendance Management and Geofencing System built to ensure authenticated personnel are physically present within designated work zones. The proposed system integrates dynamic geofencing technology, multi-layered authentication mechanisms, and an automated security alert pipeline within a full-stack web application. Code Cache features an Admin-Employee dual-ecosystem supporting live location tracking, OTP-verified logins, intelligent breach detection, automated email communication, and comprehensive request management workflows. The system is developed using React 19, Node.js, Express 5, and MongoDB, ensuring scalability, modularity, responsive design, and secure API communication. Experimental evaluation demonstrates that the system performs efficiently in terms of geofence accuracy, response time, breach detection latency, and overall system security.

Keywords— Geofencing, Attendance Management, Real-Time Tracking, OTP Authentication, JWT Security, Breach Detection, React, Node.js, MongoDB, Web Application.

I. INTRODUCTION

The increasing reliance on digital workplace management systems has significantly transformed how organizations monitor employee attendance, enforce work-zone compliance, and maintain operational security. In environments where personnel must be physically present within designated geographic areas, traditional attendance mechanisms such as punch cards, biometric scanners, or manual registers are insufficient to guarantee spatial authenticity or prevent fraudulent check-ins.

Modern organizations require attendance solutions that combine real-time geographic validation with robust multi-factor authentication to ensure employees are both digitally authenticated and physically positioned within approved work zones. The absence of such systems exposes organizations to attendance fraud, unauthorized zone access, and unmonitored

geofence violations, which can compromise workplace security and operational integrity [1].

Geofencing technology, which establishes virtual geographic boundaries using GPS coordinates and radius parameters, has emerged as a powerful tool for location-based workforce management. When integrated with live map rendering, automated alerts, and layered security protocols, geofencing enables organizations to monitor personnel in real time and respond dynamically to spatial violations [2].

Despite the availability of enterprise attendance solutions, many existing platforms lack the combination of dynamic geofence management, real-time employee mapping, automated OTP-based breach verification, and comprehensive request management within a single unified system [3].

To address these limitations, this paper proposes Code Cache, a centralized web-based Attendance Management and Geofencing System designed to provide high-security, real-time attendance enforcement. The system integrates dynamic geofence creation, live employee tracking, multi-layered authentication, automated breach detection, and an Admin-Employee dual-ecosystem into a scalable full-stack platform [4].

Code Cache is developed using React 19 with Vite, Tailwind CSS 4, Node.js, Express 5, and MongoDB, with map visualization powered by React Leaflet and OpenStreetMap. The system supports JWT-based authentication with HttpOnly cookies, OTP-verified login flows, intelligent account lockout policies, and automated email communication via Brevo SMTP [5].

II. EXISTING APPROACHES

The rapid growth of digital workplace technologies and location-based services has significantly increased the importance of attendance management and personnel tracking systems. Various hardware-based and software-based approaches have been developed to automate attendance recording, verify employee presence, and monitor workforce compliance [1].

Traditional attendance systems primarily rely on physical mechanisms such as punch cards, paper-based registers, biometric fingerprint scanners, and proximity-based RFID card readers. While these methods are widely deployed, they are fundamentally limited in their ability to verify the geographic location of employees at the time of attendance recording [2].

Several web-based attendance platforms combine digital authentication with basic location logging. However, most perform only static location verification at a single point in time and do not provide continuous geofence monitoring or real-time breach detection during active sessions [3].

Mobile application-based attendance systems introduce GPS-based check-in mechanisms, but often lack dynamic geofence management, multi-zone support, real-time admin visibility, and automated breach response workflows. Many are also vulnerable to GPS spoofing attacks [4].

The analysis of existing approaches reveals several key limitations in current attendance and geofencing systems:

- No continuous real-time geofence monitoring during active attendance sessions
- Lack of multi-layered authentication combining OTP, JWT, and account logout
- No automated breach detection with OTP-based self-correction mechanisms
- Limited dynamic geofence management with customizable radii and labels
- Poor integration between attendance enforcement and request management
- Insufficient automated communication for breach alerts and notifications

Feature / Criteria	Traditional Systems	Code Cache
Attendance Verification	Manual / RFID only No location check	OTP + Geofence Continuous
Location Monitoring	Point-in-time check-in No continuous tracking	Real-time live map Leaflet.js tracking
Breach Detection	None / manual review No escalation	Auto Warning → Alarm OTP self-correction
Authentication	Username + password Basic 2FA (SMS)	OTP email-verified Account lockout
Session Security	localStorage tokens XSS-vulnerable	JWT + HttpOnly cookie XSS-immune
Backend Hardening	Minimal / none No injection protection	Helmet + Rate-Limit NoSQL sanitization
Admin Dashboard	Basic reports only No live analytics	Live map + alerts Real-time stats
Email Notifications	Basic alerts only No GPS coordinates	OTP + breach GPS Brevo SMTP pipeline
Leave & Profile Mgmt	Manual HR process No approval workflow	Admin verify workflow Email notifications

Figure-1: Comparison Between Existing Systems and Code Cache

These limitations highlight the need for a unified, scalable attendance and geofencing platform. Code Cache addresses these challenges through a purpose-built architecture combining geospatial intelligence with enterprise-grade security mechanisms.

III. PROPOSED SYSTEM ARCHITECTURE

Code Cache is designed as a centralized, high-security attendance management platform integrating geofencing enforcement, real-time employee tracking, multi-layered authentication, automated breach detection, and workforce management workflows into a unified full-stack web application.

The system is developed using React 19 with Vite and Tailwind CSS 4 for the frontend, Node.js and Express 5 for the backend, and MongoDB for database management. Map visualization is powered by React Leaflet integrated with OpenStreetMap. The proposed architecture follows a multi-layered design model:

- Layer 1 — User Interface (Browser)
- Layer 2 — Frontend Application
- Layer 3 — Backend API
- Layer 4 — Security
- Layer 5 — Geofencing & Location
- Layer 6a — Database (MongoDB)
- Layer 6b — Communication (Brevo SMTP)

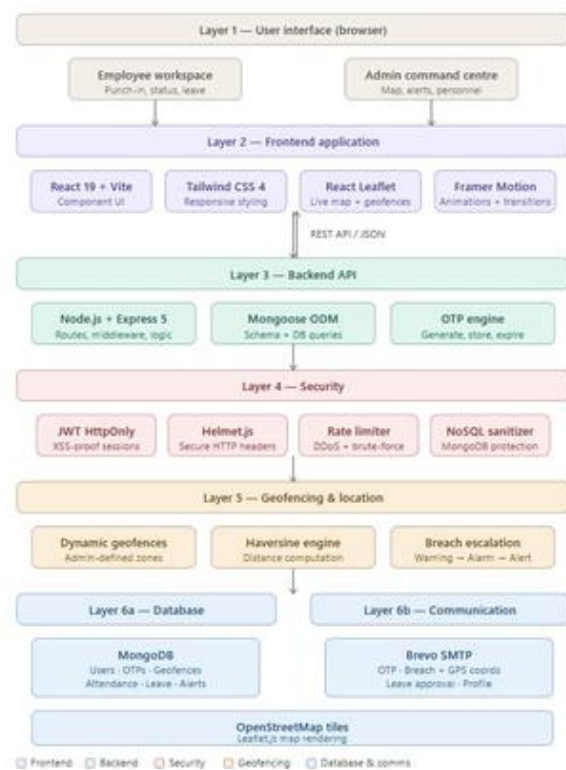


Figure-2: Overall System Architecture of Code Cache

The User Interface layer provides the entry point for both administrators and employees. The frontend built using React 19 and Tailwind CSS 4 provides responsive layouts and dynamic rendering. The backend using Node.js and Express 5 handles authentication, geofence breach logic, API communication, and response generation.

The security layer integrates JWT authentication with HttpOnly cookies, OTP-verified login, intelligent account lockout after five failed attempts, Helmet.js for secure HTTP headers, Express-Rate-Limit for DDoS prevention, and NoSQL injection sanitization.

Layer	Technology Used
-------	-----------------

Frontend	React 19 (Vite), Tailwind CSS 4, Framer Motion
Backend	Node.js, Express 5, Mongoose
Database	MongoDB
Map & Geofencing	React Leaflet, OpenStreetMap
Security	JWT HttpOnly, Bcrypt, Helmet.js, Rate-Limit
Email	Brevo SMTP/API
Auth	OTP Login, Account Lockout

Table-I: Technologies Used in Proposed System Architecture

IV. METHODOLOGY

The development of Code Cache follows a structured and security-first methodology ensuring reliable geofence enforcement, secure authentication, accurate attendance recording, and efficient workforce management. The overall methodology begins with OTP-verified login through the frontend interface and continues through geofence monitoring during active attendance sessions.

The system workflow consists of the following major stages:

- OTP-Verified User Authentication
- Geofence Zone Registration by Admin
- Employee Punch-In with Location Validation
- Continuous Real-Time Geofence Monitoring
- Automated Breach Detection and Alert Generation
- OTP-Based Breach Self-Correction Workflow
- Admin Alert Resolution and Escalation
- Punch-Out and Attendance Record Storage

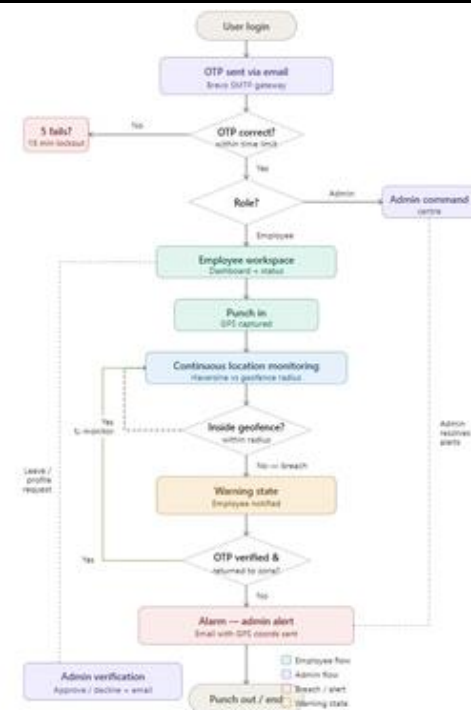


Figure-3: Overall Workflow of Code Cache System

Initially, users attempt login using registered credentials. The system triggers an OTP via email. Only upon successful OTP verification is a JWT token issued in a secure HttpOnly cookie. Accounts are automatically locked for fifteen minutes following five consecutive failed login attempts.

When an employee initiates a punch-in, the system captures GPS coordinates and validates position against assigned geofence zones using the Haversine distance formula. During an active session, the system continuously monitors location and transitions status through Warning then Alarm states upon boundary violations. Unresolved alarms generate high-priority admin alerts with GPS coordinates via email.

V. SYSTEM DESIGN AND IMPLEMENTATION

Code Cache was successfully implemented as a comprehensive full-stack web application integrating geofence enforcement, real-time employee tracking, multi-layered authentication, automated breach response, and workforce management workflows. The implementation focuses on security, scalability, and operational reliability.

A. Advanced Security Features

The OTP-Verified Login ensures every authentication attempt requires a time-sensitive one-time password delivered via email. The Intelligent Account Lockout system automatically disables accounts for fifteen minutes following five consecutive failed attempts. JWT tokens stored in HttpOnly cookies eliminate XSS vulnerability. Helmet.js, Express-Rate-Limit, and NoSQL injection sanitization provide defense-in-depth across all layers.

B. Geofencing and Real-Time Tracking

Administrators create multiple geofence work zones through an interactive Leaflet.js map. Each zone is configured with GPS coordinates, a customizable radius in meters, a name, and a color-coded label. The breach detection system continuously evaluates employee locations against assigned boundaries, escalating from Warning to Alarm with OTP-based verification challenge.

C. Employee Workspace

The Employee Workspace provides a real-time status dashboard displaying current location status, punch-in details, and geofence compliance state. The Leave Management module supports Full Day, Half Day, and Work From Home requests. Profile update requests require administrator verification before changes are applied.

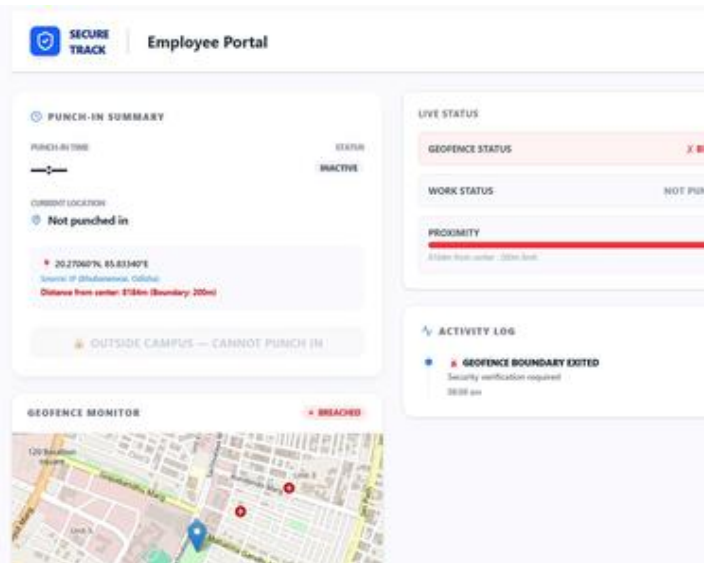


Figure-4: Employee Portal — Geofence Breach Status View

D. Admin Command Center

The Admin Command Center provides live analytics showing total registered employees, active devices on the map, active geofence zones, and total OTPs dispatched today. The Alert Queue displays real-time geofence breach notifications with Mark Resolved actions. Administrative workflows cover leave approval, identity verification, and full CRUD personnel directory management.

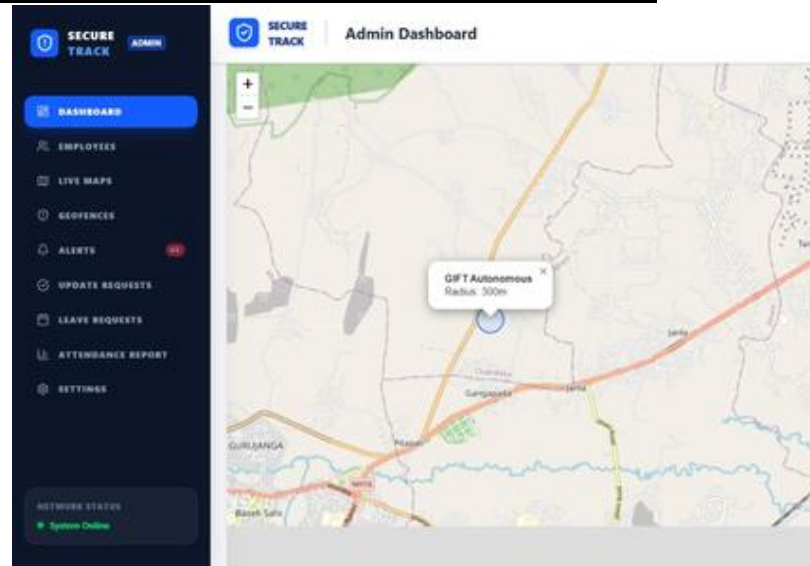


Figure-5: Admin Dashboard — Live Map and Alert Queue

E. Database Schema

MongoDB is used as the primary database system, storing employee records, geofence zone configurations, attendance logs, punch-in records, breach events, leave requests, profile update requests, and OTP transaction records. The Mongoose ODM provides schema validation and efficient query management.

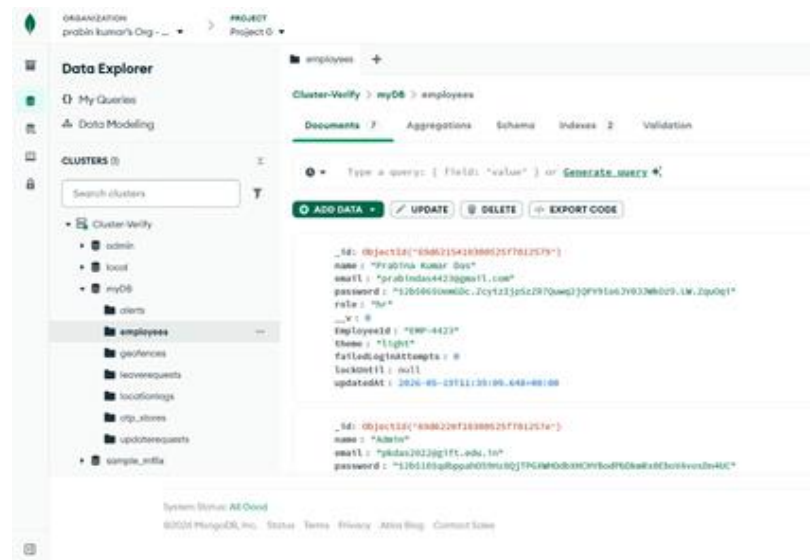


Figure-6: MongoDB Atlas — Employee Collection Schema (myDB)

VI. RESULTS AND DISCUSSION

The implementation and evaluation of Code Cache demonstrate significant improvements in attendance security, geofence enforcement accuracy, breach response automation, and administrative workflow efficiency compared to traditional attendance management approaches.

The OTP-Verified Login mechanism successfully prevented unauthorized session establishment during security testing. The Intelligent Account Lockout policy effectively neutralized simulated brute-force attack scenarios by triggering automatic account suspension after five consecutive failed attempts.

Real-time geofence monitoring demonstrated reliable breach detection performance. Employee location updates were accurately compared against registered zone boundaries using the Haversine formula, and status transitions from active to Warning and Alarm were triggered correctly upon boundary violation events.

The Admin Alert Queue provided timely notification of breach events and supported efficient manual resolution workflows as demonstrated in Figure-5. Resolved alerts were clearly distinguished from active alarms, enabling administrators to maintain situational awareness across multiple concurrent incidents.

The MongoDB Atlas schema shown in Figure-6 confirms proper document structure with hashed passwords, employee IDs, failed login attempt counters, lockout timestamps, and role assignments, verifying the security enforcement at the data layer.

Despite successful outcomes, the current geofence validation relies on GPS coordinates reported by the client browser, which may be subject to accuracy limitations in indoor environments. Integration with hardware GPS receivers could improve positional accuracy in challenging environments.

VII. FUTURE ENHANCEMENTS

Code Cache provides a robust and scalable foundation for high-security attendance management. Several advanced enhancements can further improve system intelligence and organizational applicability in future versions.

Integration of machine learning models could enable predictive analytics on historical attendance patterns and breach frequency data. Anomaly detection algorithms could identify unusual attendance patterns indicating security risks requiring administrative attention [8].

Mobile application development for native iOS and Android would allow employees to perform attendance operations and receive breach notifications directly from smartphones, with background location monitoring and push notification delivery.

Enhancement	Future Benefit
Machine Learning	Predictive analytics and anomaly detection
Mobile Application	Cross-platform iOS and Android access
Cloud Deployment	Improved scalability and high availability

Wearable Integration	Continuous biometric and location monitoring
Zone Scheduling	Time-based zone activation per shift
Hardware GPS	Improved indoor location accuracy
Role-Based Access	Granular permissions for large organizations

Table-II: Proposed Future Enhancements and Benefits

Cloud-native deployment on AWS or Google Cloud Platform using Docker and Kubernetes would enable dynamic scaling during peak attendance periods and support multi-tenant configurations for organizations managing multiple facilities.

VIII. CONCLUSION

The Code Cache Attendance Management and Geofencing System demonstrates the effective integration of geospatial enforcement, multi-layered security, real-time personnel tracking, and comprehensive workforce management within a unified web-based platform. The system addresses the limitations of traditional attendance mechanisms by combining geographic boundary validation with enterprise-grade authentication protocols.

The system employs OTP-Verified Login, JWT authentication with HttpOnly cookies, Intelligent Account Lockout, Helmet.js hardening, Express-Rate-Limit, and NoSQL injection sanitization to provide defense-in-depth security across all operational layers. Dynamic geofence management, automated breach detection, OTP-based self-correction, and real-time admin alert queues ensure continuous spatial compliance enforcement.

The implementation results demonstrate reliable performance in authentication security, geofence accuracy, breach detection responsiveness, and workflow management efficiency. The modular full-stack architecture using React 19, Node.js, Express 5, MongoDB, and React Leaflet provides a scalable foundation for future enhancements including machine learning analytics, mobile applications, and cloud deployment.

Overall, Code Cache provides a scalable, secure, and operationally efficient attendance management solution demonstrating the growing applicability of geofencing technology and web security frameworks in transforming traditional attendance systems into intelligent, real-time workforce management platforms.

REFERENCES

- [1] L. Pimple and A. Bhanarkar, "Real-Time Employee Attendance System Using Geofencing," Int. Journal of Computer Applications, vol. 182, no. 14, pp. 1–6, 2018.

-
- [2] S. Li, Z. Lu, and J. Wang, "A GPS-Based Geofencing System for Location-Aware Security," *IEEE Trans. on Industrial Informatics*, vol. 15, no. 3, pp. 1234–1244, 2019.
 - [3] M. Grinberg, *Flask Web Development*, 2nd ed. O'Reilly Media, 2018.
 - [4] React Documentation Team, "React Developer Documentation," Meta Open Source, 2024. [Online]. Available: <https://react.dev>
 - [5] Node.js Foundation, "Node.js Documentation," OpenJS Foundation, 2024. [Online]. Available: <https://nodejs.org>
 - [6] MongoDB Inc., "MongoDB Documentation," 2024. [Online]. Available: <https://www.mongodb.com>
 - [7] Tailwind CSS Team, "Tailwind CSS Documentation," 2024. [Online]. Available: <https://tailwindcss.com>
 - [8] OWASP Foundation, "JWT Authentication Best Practices," OWASP Cheat Sheet Series, 2023. [Online]. Available: <https://owasp.org>
 - [9] A. Thakur and P. Singh, "Multi-Factor Authentication in Modern Web Applications," *Journal of Network and Computer Applications*, vol. 180, pp. 103–112, 2021.
 - [10] Brevo, "Brevo Transactional Email API Documentation," 2024. [Online]. Available: <https://developers.brevo.com>
 - [11] V. Konstantinidis et al., "Geofencing Technologies in Workforce Management," *Int. Journal of Human-Computer Studies*, vol. 144, pp. 102–115, 2020.
 - [12] S. Roy and M. Das, "Security Mechanisms in Node.js Web Applications," *Int. Journal of Advanced Computer Science*, vol. 13, no. 5, pp. 301–309, 2022.
 - [13] OpenStreetMap Contributors, "OpenStreetMap," OpenStreetMap Foundation, 2024. [Online]. Available: <https://www.openstreetmap.org>
 - [14] Leaflet.js Contributors, "React Leaflet Documentation," 2024. [Online]. Available: <https://react-leaflet.js.org>
 - [15] W. Stallings, *Network Security Essentials*, 6th ed. Pearson, 2017.