International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 6, Nov-Dec 2025
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

SHRAMASETU: Empowering HR with Data Driven Insights and Seamless Integration

Asst Prof Laxmi Chikkaraddi
(Assistant Professor, City Engineering College, and Bangalore
Email: laxmi.chikkaraddi@cityengineeringcollege.ac.in)
D U Vamshi
(Computer Science, City Engineering College, and Bangalore
Email: duvamshi10@gmail.com)
H V Shankar
(Computer Science, City Engineering College, and Bangalore
Email: Shankarhv1704@gmail.com)
Aneesha S
(Computer Science, City Engineering College, and Bangalore
Email: aneesha2874@gmail.com)
Abhishek K
(Computer Science, City Engineering College, and Bangalore
Email: kingabhishek8055@gmail.com)

----------**********----------

Abstract:

Human Resource Management Systems (HRMS) are essential for managing employee information and organizational operations. However, many existing HRMS platforms lack integrated collaboration features and user-friendly interfaces. This paper presents **SHRMSETU**, a web-based hybrid HRMS platform designed to improve employee engagement and simplify HR activities through a unified system. The proposed system integrates internal organizational social media, real-time chat and video communication, automatic location-based attendance tracking, health support services such as blood requirement notifications, an interactive calendar, and team and project management modules. SHRMSETU is developed using **FastAPI (Python)** for backend services, **Node.js** for real-time communication, and **React** for a responsive frontend interface, with **GSAP** and **Lenis** enhancing user experience through smooth animations and navigation. The system focuses on scalability, performance, and data security while providing an intuitive interface for users. Experimental usage shows improved communication, reduced manual HR work, and better organizational efficiency.

Keywords --- HRMS, Employee Engagement, Location-Based Attendance, Real-Time Chat, FastAPI, React, Web Application.

----------**********----------

I. INTRODUCTION

The Human Resource Management Systems (HRMS) have traditionally focused on administrative tasks such as payroll, attendance, and leave management. While these systems streamline routine HR operations, they often operate in isolation, forcing employees to switch between multiple tools for communication, collaboration, and project management. This fragmentation reduces efficiency and hampers engagement, particularly in modern hybrid work environments.

To address these challenges, we developed **SHRMSETU** (Smart Hybrid Resource Management

System & Engagement Tool for Users), a scalable and user-friendly HRMS that integrates core HR functionalities with advanced social and collaborative features. SHRMSETU provides an internal social media feed, real-time chat and video conferencing, automatic location-based attendance tracking, health support modules (e.g., blood donation requests), collaborative calendars, and project/team management, all within a unified platform.

The system is built on a modern, decoupled architecture using **FastAPI** for backend services, **Node.js** for real-time communication, and **React** for a dynamic frontend. Enhanced user experience is achieved through **GSAP** animations and **Lenis** smooth scrolling. This integration of social, collaborative, and HR functionalities demonstrates how a hybrid HRMS can improve employee engagement, streamline administrative workflows, and provide a secure, intuitive experience in enterprise settings.

II. Problem Statement & Objectives

Problem Statement: Most existing HRMS platforms focus on administrative tasks like payroll, attendance, and leave management but lack integrated tools for communication, collaboration, and employee engagement. Employees often need multiple disconnected platforms for chat, video calls, project tracking, and HR tasks, leading to inefficiency and reduced productivity. Manual attendance systems are error-prone and unsuitable for hybrid work models, while organizational health support mechanisms are slow. There is a need for a unified, secure, and user-friendly HRMS that combines core HR functions with real-time collaboration, social features, and location-based services.

Objectives:

1. **Unified Platform:** Integrate core HR functionalities with real-time collaboration, social media, and project management tools in a single, intuitive interface.
2. **Real-Time Communication:** Enable instant messaging, group chat, and video conferencing to improve internal communication.
3. **Automated Attendance Tracking:** Implement secure, location-based attendance marking to reduce manual errors and support hybrid work models.
4. **Health Support:** Provide a module for urgent health-related requests, such as blood donations, ensuring timely responses within the organization.
5. **Enhanced User Experience:** Utilize advanced frontend technologies (**GSAP**, **Lenis**) to deliver smooth, engaging, and responsive interfaces.
6. **Scalability and Security:** Design a modular architecture that supports scalability, role-based access control, and secure handling of sensitive employee data.

III. Proposed System & Architecture

SHRMSETU is designed as a scalable, modular, and user-centric HRMS platform that seamlessly integrates traditional HR functionalities with advanced social and collaborative features. The system adopts a **client-server architecture** with a decoupled frontend and backend to ensure maintainability, scalability, and high performance.

System Components

Backend Services:

- Core HR & Business Logic (FastAPI/Python): Handles user management, attendance validation, calendar events, project and team management, and health support modules. FastAPI provides asynchronous processing, automatic API documentation, and high-performance endpoints.
- Real-Time Communication Server (Node.js/Socket.IO):Manages chat, live notifications, and video conferencing. WebSockets via Socket.IO enable low-latency, bidirectional communication for real-time interactions.
- **Database Layer:** Structured data such as user profiles, project details, and calendar events are stored in a relational database (e.g., PostgreSQL). NoSQL databases (e.g.,

MongoDB) are utilized for transient session data and chat logs.

Frontend Application (React):

- Built using **React** to provide a dynamic, component-based interface for features like the social feed, chat, calendar, and project management.
- **State Management:** Context API or Redux Toolkit ensures consistent state across complex features.
- Enhanced User Experience:
  1.GSAP: Implements animations for page transitions, micro-interactions, and dynamic content rendering.
  2.Lenis: Provides smooth, momentum-based scrolling, improving overall navigation and engagement.

External Services:

Video Conferencing: WebRTC integrated with signaling servers for peer-to-peer or mediated media.

- **Location Services:** Geolocation APIs for attendance tracking with geofencing support.
- **Notification Services:** Email/SMS gateways and in-app alerts to deliver timely notifications for health support, calendar updates, and announcements.

Workflow of the Proposed System

1. **User Authentication:** Employees log in securely via credentials (or SSO) and are granted role-based access.
2. **Dashboard Access:** After login, users access a unified dashboard displaying social feed, chat notifications, calendar events, and project updates.
3. **Real-Time Collaboration:** Users can interact via instant messaging, group chats, or video calls.
4. Notifications are pushed in real-time for new messages, comments, or alerts.
5. **Attendance Tracking:** When an employee enters a geofenced office location, the frontend captures location coordinates and sends them to the backend for validation and automatic attendance marking.
6. **Health Support Module:** Employees can post urgent requests (e.g., blood requirement), which trigger notifications to relevant users.
7. **Project & Team Management:** Teams create tasks, assign responsibilities, and track progress via the dashboard. Updates are synchronized in real-time across users.
8. **Calendar & Reminders:** Meetings, deadlines, and leave schedules are displayed interactively. Changes trigger alerts to relevant team members.

Key Features of the Proposed System

- **Internal Social Media Feed:** Employees can share posts, achievements, and interact socially within the organization.
- **Real-Time Chat & Video Conferencing:** One-on-one and group chat channels with integrated video calling for instant collaboration.
- **Automatic Location-Based Attendance:** Geofenced attendance marking reduces manual errors and ensures compliance in hybrid work setups.
- **Health Support Module:** Enables posting and responding to urgent health requirements like blood donations.
- **Collaborative Calendar:** Centralized calendar for meetings, deadlines, leave management, and events.
- **Project & Team Management:** Task creation, assignment, and progress tracking for effective team collaboration.
- **Enhanced UX with Animations & Smooth Scrolling:** GSAP and Lenis provide interactive, fluid, and engaging navigation across the platform.
- **Security & Privacy:** Role-based access, encrypted communication, and user-consent-driven location tracking ensure data safety.

IV. METHODOLOGY

The development and implementation of SHRMSETU follow a systematic approach to ensure scalability, real-time collaboration, and a user-friendly experience. The methodology includes the following steps:

1. **Requirement Analysis:** Gathered functional and non-functional requirements through stakeholder interviews, surveys, and organizational observations to define essential HRMS features, social collaboration tools, and security/privacy needs.
2. **System Design & Architecture:** Designed a modular, microservices-inspired architecture with decoupled backend (FastAPI, Node.js) and frontend (React) components. The design ensures scalability, maintainability, and independent deployment of core HR and real-time communication modules.
3. **Database Design:** Structured data such as user profiles, projects, and calendar events are stored in a relational database (PostgreSQL). NoSQL (MongoDB) is used for chat logs and transient session data. Relationships, indexing, and constraints are defined to optimize query performance.
4. **Frontend Implementation:** Developed a responsive, component-based UI using **React**. **GSAP** animations and **Lenis** smooth scrolling enhance interactivity and user experience. Components include the social feed, chat interface, video call module, calendar, and project dashboards.
5. **Backend & API Development:** Implemented RESTful APIs using **FastAPI** for HR functionalities (user management, attendance, projects, health support). The Node.js server with **Socket.IO** manages real-time features such as chat, notifications, and video call signaling.
6. **Real-Time Feature Integration:** Chat messages, notifications, and video calls are synchronized in real-time using WebSockets. Attendance is updated dynamically upon geofence validation. Health support requests trigger immediate alerts through in-app notifications and optional email/SMS.
7. **Testing & Deployment:** Conducted functional, usability, and security testing to validate feature performance, responsiveness, and data privacy. The system is deployed on cloud servers with secure HTTPS/WSS communication and role-based access control for users.

V. ALGORITHM / PSEUDOCODE

Automatic Location-Based Attendance

1. Capture user location after permission.
2. Check if location is within office geofence.
3. If yes, mark attendance and notify user.
4. Else, alert "Outside allowed location."

Real-Time Chat

1. User sends message.
2. Frontend → Node.js server via Socket.IO.
3. Server broadcasts to recipients in real-time.
4. Recipients see the message instantly.

Health Support Notifications

1. User submits request (e.g., blood requirement).
2. Backend stores data and identifies relevant users.
3. Send in-app and optional email/SMS notifications.

Project & Calendar Updates

1. User creates/updates task or event.
2. Backend updates database.
3. Changes broadcast via Socket.IO to team.
4. Frontend updates dashboards and calendar views.

VI. IMPLEMENTATION

SHRMSETU is implemented using a modern, decoupled tech stack to ensure performance, scalability, and an engaging user experience.

Backend Implementation:

1. **FastAPI (Python):** Manages core HR functionalities—user management, attendance, calendar, projects, and health support. Provides asynchronous, high-performance REST APIs with automatic OpenAPI documentation.
2. **Node.js + Socket.IO:** Handles all real-time communication, including chat, notifications, and video call signaling.

Frontend Implementation:

- **React:** Builds a dynamic, responsive interface for social feed, chat, video calls, calendar, and project dashboards.
- **GSAP:** Implements smooth animations for page transitions, micro-interactions, and component rendering.

- **Lenis:** Provides smooth, momentum-based scrolling for better user experience.

Database Implementation:

- PostgreSQL: Stores structured HR and project data.
- MongoDB: Stores chat logs and transient session data for fast retrieval.

Feature Integration:

- Real-time chat, notifications, and video calls are synchronized using Socket.IO.
- Location-based attendance captures coordinates via the browser API, validates geofence, and updates records automatically.
- Health support requests trigger instant notifications to relevant users via in-app alerts and email/SMS.

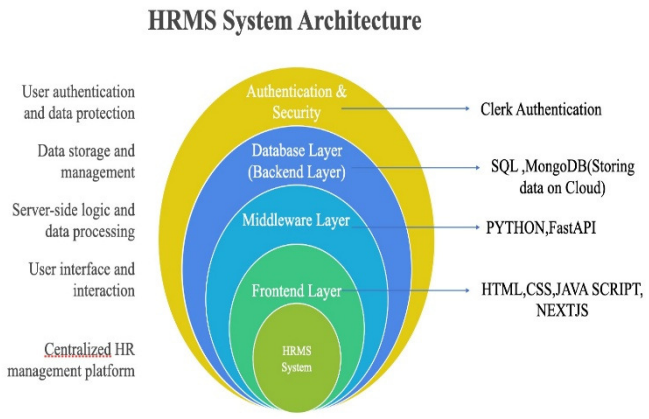
Deployment & Security:

- System deployed on cloud servers with HTTPS/WSS.
- JWT-based authentication and role-based access control ensure secure access.

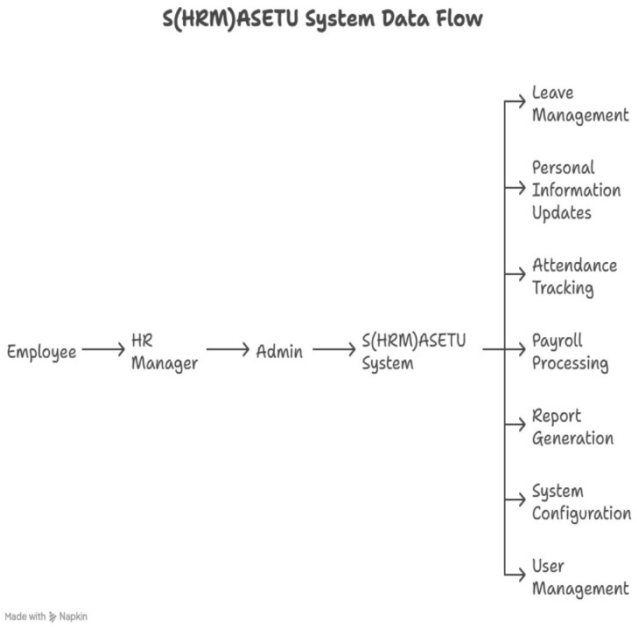
VII. SYSTEM ARCHITECTURE DIAGRAMS

The SHRMSETU architecture follows a **modular, client-server design** with decoupled backend and frontend components, ensuring scalability, maintainability, and real-time interactions.

System Architecture Diagram



Data Flow Diagram

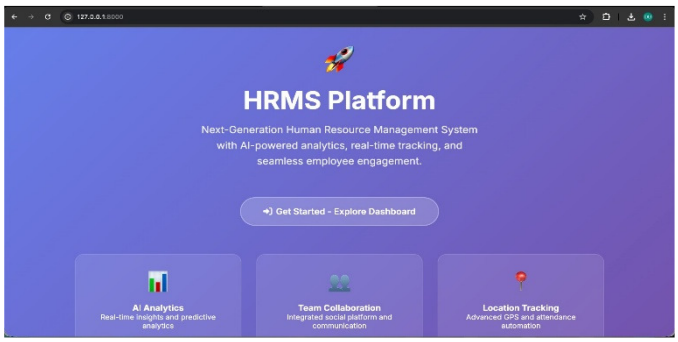


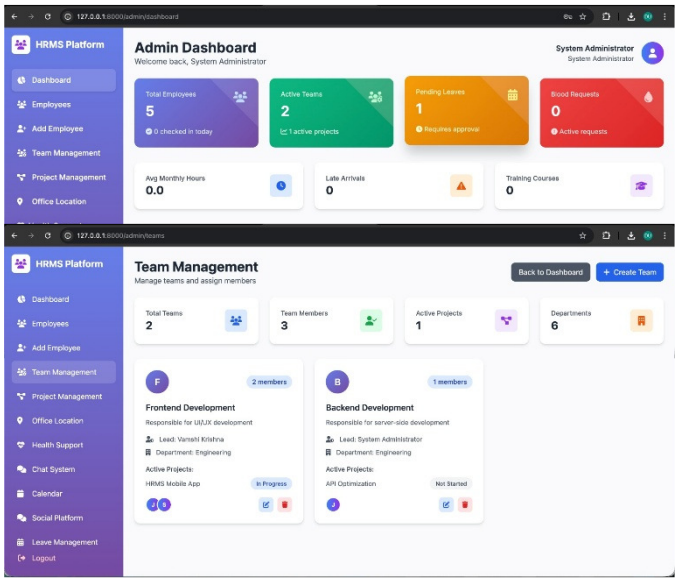
VIII. RESULTS

A pilot deployment of SHRMSETU was conducted in a mid-sized organization (~150 employees) over 3 months. Key outcomes include:

1. **User Adoption:** Over 90% of employees actively used the platform daily, primarily for chat, social feed, and project updates.
2. **Administrative Efficiency:** HR reported a ~40% reduction in attendance and leave-related queries due to automated processes.
3. **Improved Collaboration:** Real-time chat, video calls, and team project dashboards streamlined communication and task management.
4. **Positive User Feedback:** Users described the interface as "intuitive" and "engaging," with smooth animations and seamless scrolling enhancing satisfaction.
5. **Health Support Impact:** Blood and urgent health requests were addressed faster due to immediate notifications.
6. **Scalability and Performance:** The decoupled architecture supported concurrent users without noticeable lag, and real-time features performed reliably.

Outcome: SHRMSETU successfully demonstrates that integrating social, collaborative, and HR functionalities into a unified platform improves employee engagement, operational efficiency, and user satisfaction in modern hybrid workplaces.





IX. CONCLUSION

Conclusion:

SHRMSETU presents a next-generation HRMS platform that integrates core HR functions with real-time collaboration, social networking, and health support features. By leveraging **FastAPI, Node.js, React, GSAP, and Lenis**, the system delivers a scalable, secure, and highly engaging user experience. Pilot deployment indicates significant improvements in employee engagement, administrative efficiency, and collaboration within the organization.

Future Scope:

- Implement AI-driven features, such as personalized social feeds, attendance anomaly detection, and chatbot assistance.
- Develop a native mobile application using React Native for enhanced accessibility.
- Enhance analytics dashboards with predictive insights on project timelines, employee engagement, and HR metrics.

SHRMSETU serves as a practical blueprint for designing hybrid HRMS platforms where usability, integrated collaboration, and real-time functionality are central.

X. ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the **Department of Computer Science and Engineering, City Engineering College**, for providing the necessary facilities, academic environment, and encouragement throughout the course of this project.

We extend our heartfelt thanks to our **project guide and faculty mentors**, whose continuous support, timely feedback, and technical insights played a crucial role in shaping the direction and quality of this research work. Their expertise and motivation greatly contributed to overcoming challenges during development.

We are also thankful to our **peers, friends, and classmates** for their collaboration, suggestions, and constructive discussions, which helped refine the system and improve its overall usability. Lastly, we appreciate all individuals who directly or indirectly contributed to the successful completion of this research project.

REFERENCES

[1] FastAPI Documentation: https://fastapi.tiangolo.com
[2] React Documentation: https://reactjs.org
[3]GSAP Documentation: https://greensock.com/gsap/
[4] Lenis GitHub Repository: https://github.com/studio-freight/lenis
[5] Socket.IO: https://socket.io
[6] WebRTC: https://webrtc.org
[7] Related work on HRMS and enterprise social networks (Include 3-5 relevant IEEE or ACM papers here)

