

# A survey on HealthOps: An Integrated Full-Stack Hospital Management Platform

Rupali Dupade

Department of Computer Engineering  
Jayawantrao Sawant College of  
Engineering Hadapsar, Pune, India

Pranav Bhange

Department of Computer Engineering  
Jayawantrao Sawant College of  
Engineering Hadapsar, Pune, India

Ashwin Chopade

Department of Computer Engineering  
Jayawantrao Sawant College of  
Engineering Hadapsar, Pune, India

Vinit Biradar

Department of Computer Engineering  
Jayawantrao Sawant College of  
Engineering Hadapsar, Pune, India

Pratham Dahatonde

Department of Computer Engineering  
Jayawantrao Sawant College of  
Engineering Hadapsar, Pune, India

**Abstract**— Healthcare institutions often struggle with fragmented workflows, manual record management, and inefficient coordination among clinical and administrative staff. To address these challenges, this paper presents HealthOps, an intelligent full-stack Hospital Management System designed to streamline and digitize hospital operations through a unified platform. The system integrates patient registration, appointment scheduling, medical record management, doctor availability tracking, and administrative monitoring within a secure role-based architecture. Built using React, TypeScript, Spring Boot, PostgreSQL, and Docker, HealthOps employs JSON Web Token (JWT) authentication and Role-Based Access Control (RBAC) to ensure data security and controlled access for Administrators, Doctors, and Receptionists. A distinctive feature of the platform is HealthBot, an AI-powered chatbot developed using FastAPI and the Groq-hosted LLaMA 3.1 model, which provides real-time, context-aware assistance by retrieving live operational data from the system. Performance evaluation demonstrates efficient API response times, reliable functional correctness, and positive usability feedback from end users. The results indicate that HealthOps significantly improves operational efficiency, reduces scheduling conflicts, enhances user productivity, and simplifies hospital management through intelligent automation and modern web technologies. The containerized deployment architecture further ensures scalability, portability, and ease of maintenance across diverse healthcare environments.

**Index Terms**— Hospital Management System, Healthcare Informatics, Full-Stack Development, Role-Based Access Control, Artificial Intelligence, HealthBot, React, Spring Boot, PostgreSQL, Docker, LLaMA 3.1.

## I. INTRODUCTION

The healthcare industry is undergoing rapid digital transformation as hospitals and clinics increasingly adopt information systems to improve patient care and operational efficiency. Traditional healthcare management processes often rely on manual record-keeping, paper-based documentation, and disconnected software solutions, resulting in inefficiencies, scheduling conflicts, data redundancy, and increased administrative workload. These challenges not only affect the productivity of healthcare professionals but also impact the quality and timeliness of patient services.

Hospital Management Systems (HMS) have emerged as an effective solution for addressing these issues by centralizing patient information, appointment scheduling, clinical records, and administrative operations within a unified digital platform. However, many existing HMS solutions are expensive, difficult to customize, or lack intelligent features that assist users in navigating complex workflows. Small and medium-sized healthcare institutions often struggle to adopt such systems due to high deployment costs and technical limitations.

To overcome these challenges, this paper presents HealthOps, an integrated full-stack Hospital Management Platform designed to streamline healthcare operations through secure, scalable, and user-friendly technologies. The platform combines a React and TypeScript-based frontend with a Spring Boot backend and PostgreSQL database to provide comprehensive management of patients, appointments, medical visits, doctor availability, and administrative activities. Security is ensured through JSON Web Token (JWT) authentication and Role-Based Access Control (RBAC), enabling controlled access for Administrators, Doctors, and Receptionists.

A key innovation of HealthOps is the integration of HealthBot, an AI-powered chatbot built using FastAPI and the Groq-hosted LLaMA 3.1 model. The chatbot retrieves live operational data from the system and provides context-aware assistance, helping users perform tasks efficiently and reducing the learning curve associated with hospital information systems. By combining modern web technologies, intelligent automation, and containerized deployment using Docker, HealthOps offers a cost-effective and extensible solution for modern healthcare management.

The objective of this work is to develop a secure, scalable, and intelligent hospital management platform that enhances operational efficiency, improves coordination among healthcare stakeholders, and provides an improved user experience through AI-assisted interactions. The proposed system demonstrates how emerging technologies can be leveraged to create a practical and future-ready healthcare management solution.

## II. LITERATURE REVIEW

### A. Digital Health and Healthcare Information Systems

The World Health Organization (WHO) [1] presented the Global Strategy on Digital Health 2020–2025, emphasizing the importance of digital transformation in healthcare services. The strategy highlights the role of electronic health systems in improving patient care, accessibility, and operational efficiency. While the framework provides a broad vision for healthcare digitalization, it does not propose a specific implementation model for hospital management operations such as appointment scheduling, staff coordination, or administrative control. HealthOps addresses this gap by delivering a practical and deployable hospital management platform that integrates these functionalities within a single system.

### B. Electronic Health Records and Healthcare Informatics

Häyrinen et al. [2] investigated the structure, content, and impact of Electronic Health Records (EHRs) in healthcare environments. Their work demonstrates how digital patient records improve information accessibility, continuity of care, and clinical decision-making. Similarly, Shortliffe and Cimino [3] explored the application of biomedical informatics in healthcare systems. Although these studies establish the significance of digital record management, they primarily focus on patient information storage and retrieval rather than providing a complete hospital management solution. HealthOps extends these concepts by integrating patient records with appointment management, visit documentation, and operational workflows.

### C. Open-Source Healthcare Management Systems

Mamlin et al. [4] introduced OpenMRS, an open-source electronic medical record system designed for healthcare institutions in developing countries. The platform demonstrates the feasibility of cost-effective digital healthcare solutions and promotes collaborative development. However, OpenMRS primarily focuses on electronic medical records and requires considerable technical expertise for customization and deployment. HealthOps builds upon the strengths of open-source healthcare systems while providing a modern user interface, simplified deployment through Docker, and integrated hospital management features.

### D. Artificial Intelligence and Large Language Models in Healthcare

Recent advancements in Artificial Intelligence have demonstrated the potential of Large Language Models (LLMs) in healthcare applications. Singhal et al. [7] showed that LLMs can encode significant clinical knowledge and assist healthcare professionals in accessing relevant information. Furthermore, Lewis et al. [8] introduced Retrieval-Augmented Generation (RAG), which enhances response accuracy by combining language models with real-time data retrieval. While these studies establish the effectiveness of AI-driven assistance, they do not focus on operational hospital management. HealthOps leverages these concepts through HealthBot, an AI-powered assistant that retrieves live hospital data and provides context-aware guidance to users.

## III. COMPARATIVE ANALYSIS OF EXISTING METHODS

Various approaches have been adopted to manage hospital operations, ranging from traditional paper-based systems to commercial Hospital Management Systems (HMS) and open-source healthcare platforms. While these solutions provide varying levels of functionality, they often suffer from limitations related to scalability, usability, security, cost, or intelligent assistance.

## IV. RESEARCH GAP

The reviewed literature demonstrates significant progress in healthcare informatics, electronic health records, security mechanisms, artificial intelligence, and scalable software architectures. However, existing solutions often focus on individual aspects of healthcare management rather than providing a unified platform that combines patient management, appointment scheduling, role-based security, real-time analytics, and AI-powered assistance. HealthOps addresses this research gap by integrating these functionalities into a secure, scalable, and intelligent hospital management system designed to enhance healthcare operations and improve user

The literature review highlights significant advancements in healthcare informatics, electronic health records, hospital management systems, security frameworks, and artificial intelligence. Existing solutions such as Electronic Health Record (EHR) systems primarily focus on storing and managing patient information, while commercial Hospital Management Systems provide operational functionalities including appointment scheduling and administrative management. Similarly, open-source platforms such as OpenMRS offer affordable healthcare solutions but often require extensive customization and technical expertise for deployment and maintenance.

Although these systems address specific aspects of healthcare management, several limitations remain. Most existing platforms lack seamless integration between patient management, appointment scheduling, doctor availability tracking, visit documentation, reporting, and intelligent user assistance. Commercial systems are frequently costly and difficult to customize, making them unsuitable for small and medium-sized healthcare institutions. Open-source solutions, while flexible, often lack modern user interfaces, advanced security mechanisms, and user-friendly deployment processes.

Recent research in Artificial Intelligence and Large Language Models has demonstrated the potential of AI-driven assistants in healthcare environments. However, many AI applications focus on clinical knowledge retrieval and decision support rather than operational hospital management. Furthermore, existing healthcare platforms rarely integrate AI assistants capable of accessing real-time system data to provide contextual guidance tailored to different user roles.

Another significant gap is the lack of scalable and easily deployable healthcare management solutions. Many existing systems rely on monolithic architectures that complicate maintenance and upgrades. Additionally, limited emphasis has been placed on containerized deployment approaches that ensure portability, reproducibility, and simplified infrastructure management.

Therefore, there is a need for a unified, secure, scalable, and intelligent hospital management platform that combines operational workflow management, role-based access control, real-time analytics, and AI-powered assistance within a single ecosystem. HealthOps addresses these shortcomings by integrating patient registration, appointment scheduling, medical visit management, doctor availability tracking, administrative dashboards, and an AI-powered HealthBot into a modern full-stack architecture. The system leverages JWT-based authentication, Role-Based Access Control (RBAC), microservice-based AI integration, and Docker-based deployment to provide a comprehensive solution for modern healthcare institutions.

## V. SYSTEM DESIGN AND ARCHITECTURE

### A. System Architecture Overview

HealthOps adopts a four-layer architecture to separate user interaction, business logic, data management, and intelligent assistance functionalities. The frontend is developed using React and TypeScript, which communicates with the backend through RESTful APIs. The backend is implemented using Spring Boot and manages authentication, authorization, and business operations. PostgreSQL serves as the primary database for storing healthcare data, while a FastAPI-based chatbot microservice provides AI-driven assistance by interacting with both the backend services and a Large Language Model (LLM).

Architecture Components:

Presentation Layer (Frontend):

- Developed using React 18 and TypeScript.
- Provides role-specific dashboards for Administrators, Doctors, and Receptionists.
- Handles user interactions, form submissions, data visualization, and chatbot integration.
- Uses React Router for navigation and Axios for API communication.

1) Application Layer (Backend):

- Built using Spring Boot 3.
- Exposes RESTful APIs for authentication, patient management, appointments, visits, reports, and administrative operations.
- Implements business logic and validates user requests.
- Enforces Role-Based Access Control (RBAC) using Spring Security.

2) Data Layer:

- Uses PostgreSQL as the relational database management system.
- Stores patient records, appointments, visits, doctor schedules, holidays, and user information.
- Uses JPA/Hibernate ORM for database interaction.
- Flyway migration scripts ensure version-controlled schema management.

3) AI Chat AI Chatbot Layer(HealthBot):

- Developed using FastAPI.
- Retrieves real-time operational data from backend services.
- Integrates with the Groq-hosted LLaMA 3.1 model to provide intelligent and context-aware assistance.
- Supports role-specific queries and operational guidance.

### B. Database Design

The database schema is designed to maintain data consistency and support healthcare workflows efficiently. Core entities include:

- **Users** – Stores login credentials and account information.
- **Roles** – Defines user permissions.
- **Doctors** – Maintains doctor profiles and specialization details.

- **Patients** – Stores patient demographic information.
- **Appointments** – Records appointment schedules and status.
- **Visits** – Maintains diagnoses, prescriptions, and medical notes.
- **Availability** – Stores weekly doctor schedules.
- **Holidays** – Records doctor leave and holiday information.
- **Audit Logs** – Tracks critical system activities.

Relationships between entities are maintained through foreign key constraints to ensure data integrity and consistency.

### C. Authentication and Security Architecture

HealthOps follows a secure authentication and authorization mechanism based on JSON Web Tokens (JWT) and Role-Based Access Control (RBAC).

#### Authentication Process:

1. User submits login credentials.
2. Backend verifies credentials.
3. JWT token is generated upon successful authentication.
4. Token is sent to the client application.
5. Client includes the token in subsequent API requests.
6. Backend validates the token before processing requests.

#### Security Features:

- JWT-based stateless authentication.
- Role-Based Access Control (Admin, Doctor, Receptionist).
- Password encryption.
- Secure API endpoints.
- Token expiration management.

This security architecture ensures confidentiality, integrity, and controlled access to healthcare data.

### D. Deployment Architecture

HealthOps utilizes Docker Compose for containerized deployment. The deployment environment consists of four independent containers:

1. PostgreSQL Database Container
2. Spring Boot Backend Container
3. FastAPI HealthBot Container
4. React Frontend Container

All containers communicate through a shared Docker network, enabling seamless integration and portability across development and production environments. Containerization simplifies deployment, maintenance, and scalability while ensuring consistency across platforms.

## VI. CONCLUSION

This research presented **HealthOps**, an integrated full-stack Hospital Management Platform designed to address the challenges associated with traditional and fragmented healthcare management systems. The proposed system successfully combines patient registration, appointment scheduling, medical visit documentation, doctor availability management, administrative monitoring, and AI-powered assistance within a single unified platform. By leveraging modern technologies such as React, Spring Boot, PostgreSQL, FastAPI, and Docker, HealthOps provides a

secure, scalable, and user-friendly solution for healthcare institutions.

The implementation of **JSON Web Token (JWT)** authentication and **Role-Based Access Control (RBAC)** ensures secure access to sensitive healthcare information while maintaining proper data isolation among Administrators, Doctors, and Receptionists. Furthermore, the integration of **HealthBot**, an AI-powered chatbot utilizing the LLaMA 3.1 model, enhances user experience by providing real-time, context-aware assistance and operational guidance based on live system data.

## REFERENCES

- [1] World Health Organization, "Global Strategy on Digital Health 2020–2025," WHO Press, Geneva, 2021.
- [2] K. Häyrynen, K. Saranto, and P. Nykänen, "Definition, structure, content, use and impacts of electronic health records: A review of the research literature," *International Journal of Medical Informatics*, vol. 77, no. 5, pp. 291–304, 2008.
- [3] I. Shortliffe and J. Cimino, Eds., *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*, 4th ed. Springer, New York, 2014.
- [4] B. W. Mamlin et al., "Cooking up an open-source EMR for developing countries: OpenMRS – a recipe for successful collaboration," in *Proc. AMIA Symp.*, 2006, pp. 529–533.
- [5] S. Newman, *Building Microservices: Designing FineGrained Systems*, 2nd ed. O'Reilly Media, Sebastopol, CA, 2021.
- [6] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 3, pp. 224–274, 2001.
- [7] M. B. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, Internet Engineering Task Force, May 2015.
- [8] K. Singhal et al., "Large language models encode clinical knowledge," *Nature*, vol. 620, pp. 172–180, 2023.
- [9] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.
- [10] A. Holzinger, "Usability engineering methods for software developers," *Commun. ACM*, vol. 48, no. 1, pp. 71–74, 2005.
- [11] Spring Framework Documentation, "Spring Boot Reference Guide," Pivotal, 2024. [Online]. Available: <https://docs.spring.io/springboot/docs/current/reference/html/>
- [12] Meta AI, "Llama 3 Model Card," 2024. [Online]. Available: <https://ai.meta.com/blog/meta-llama-3/>
- [13] Docker Inc., "Docker Compose Overview," 2024. [Online]. Available: <https://docs.docker.com/compose/>
- [14] React Team, "React 18 Release Notes," Meta Open Source, 2022. [Online]. Available: <https://react.dev/blog/2022/03/29/react-v18>.