IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 11, April 2025
Impact Factor: 7.67

A Hybrid Hospital Framework Using HTML, CSS3, REACT and Mangodb for Real-Time Assistance and Data Management

Ashwin Anil Shelke, Deep Shelke, Trushna Shelar, Tejas Shende,
Shaikh Roman, Mrs. Shubhnagi Kshirsager
Department of Computer Engineering
Dr. DY Patil College of Engineering and Innovation, Pune, India

Abstract: *The healthcare industry is rapidly going digital to improve efficiency, security, and patient care. This paper proposes a hybrid hospital management system that combines HTML5, CSS3, React.js, and MongoDB to enable real-time support and efficient data management. The system has an AI-driven chatbot for quick support, role-based access control (RBAC) for security, and workflow automation for appointments, prescriptions, and patient history. Analyses indicate 62% quicker patient registration, 37% less no-show for appointments, and 85% satisfaction of doctors, establishing its effectiveness. The architecture fills the gap between old HMS and contemporary AI-based systems, providing an easily scalable, user-friendly health management system.*

Keywords: Hospital Management System, AI Chatbot, React.js, MongoDB, Real-Time Data Processing, Role-Based Access Control

I. INTRODUCTION

Healthcare facilities are challenged with data management, appointment scheduling, and realtime communication. Conventional Hospital Management Systems (HMS) tend to have noninteractive interfaces, lack AI support, and are not scalable, resulting in inefficiencies.

This study suggests a hybrid framework that integrates:

- Front-end: HTML5, CSS3, React.js (dynamic UI).
- Back-end: Node.js, Express (RESTful APIs).
- Database: MongoDB (NoSQL for scalability).
- AI Chatbot: NLP-based real-time support.

1. Design a secure, role-based HMS with real-time updates.
2. Implement an AI chatbot for real-time user support.
3. Streamline workflows (appointments, prescriptions, records).
4. Assess efficiency, security, and user satisfaction.

The system was pilot tested in a 250-bed hospital, demonstrating strong improvements in workflow speed, accuracy of data, and user experience.

II. LITERATURE REVIEW

2.1 Conventional HMS Drawbacks

- Patel & Patel (2018) ascertained legacy HMS are affected by:
- Poor UI/UX → Low adoption. no\tNo realtime support → Manual mistakes.
- Limited interoperability → Data silos.

DOI: 10.48175/568

2.2 Role-Based Access Control (RBAC)

- Kumar & Singh (2020) demonstrated RBAC improves security with streamlined workflows.

2.3 AI in Healthcare

- Chatterjee et al. (2020): Chatbots cut down consultation time by 30%.
- Zhang & Wang (2017): AI enhances appointment scheduling and diagnostics.

2.4 Research Gap

- Current HMS do not have AI integration, resulting in manual inefficiencies. This research combines HMS + AI chatbots + real-time data processing for an integrated solution.

III. METHODOLOGY

3.1 System Architecture

We employed a three-tiered model:

- Presentation Layer (React.js) → Responsive UI.
- Application Layer (Node.js) → Business logic & chatbot.
- Data Layer (MongoDB) → Encrypted patient records.

3.2 Role-Based Access Control Role

- Permissions
- Admin
- Full system control

3.3 Objectives

Doctor

Patient records, prescriptions

Receptionist

Appointments, billing

Nurse

Vital signs, care plans

3.4 AI Chatbot Integration

Python, TensorFlow, NLTK were used to build it.

Functions:

FAQs (hospital policies, visiting hours).

Appointment booking (automated reminders).

Basic triage (symptom checker).

IV. TECHNOLOGY STACK

Component

Technology

Front-End

React.js, HTML5, CSS3

Back-End

Node.js, Express

Database

MongoDB

Security
JWT, Bcrypt

V. IMPLEMENTATION

5.1 Core Modules

1. User Management → Secure login (JWT).
2. Patient Records → Digital history storage.
3. Appointments → Automated scheduling.
4. Prescriptions → E-prescriptions with validation.
5. Chatbot → 24/7 assistance.

5.2 Key Interfaces

- Admin Dashboard: User management, analytics.
- Doctor Portal: Patient history, e-prescriptions.
- Chatbot Widget: Present on every page.

5.3 Real-Time Features

- Live appointment updates (WebSockets).
- Instant chatbot responses (NLP).
- Auto-sync patient records (MongoDB).

VI. RESULTS & DISCUSSION

6.1 Performance Metrics

Metric	Improvement
Patient Registration	62% faster
Appointment No-Shows	37% reduction
Prescription Errors	49% fewer
Chatbot Query Resolution	78% success rate

6.2 User Feedback

85% of doctors claimed smoother workflow.
82% of patients found the chatbot useful.

6.3 Challenges

Staff training took considerable time.
Chatbot had difficulty with sophisticated medical jargon.
Legacy system migration involved data cleansing.

VII. CONCLUSION & FUTURE WORK

7.1 Conclusion

The hybrid model increases efficiency, security, and user experience by combining HMS + AI + real-time data.

7.2 Future Enhancements

- Hands-free voice-enabled chatbot.
- Predictive analytics for patient inflow.
- Blockchain for tamper-proof records.

REFERENCES

[1]. Patel, M. & Patel, S. (2018). *Survey on HMS Software*. JSEA.

[2]. Kumar, R. & Singh, P. (2020). *RBAC in Healthcare*. JMI.

[3]. Zhang, L. & Wang, H. (2017). *AI in Hospital Management*. IEEE.

DOI: 10.48175/568