

## DEVELOPMENT OF A SECURE AND SCALABLE RECEPTIONIST MODULE FOR A HOSPITAL MANAGEMENT SYSTEM

Patel Alok Rakeshkumar\*<sup>1</sup>

\*<sup>1</sup>Reyna Solutions LLP, Vadodara, Gujarat, India.

DOI : <https://www.doi.org/10.56726/IRJMETS71493>

### ABSTRACT

In today's digital healthcare ecosystem, efficient hospital management solutions are essential to streamline frontline operations and improve patient engagement. This paper presents the design and development of a secure and modular Receptionist Module as part of HealthBox – A Complete Hospital Management System. Built using the MERN stack (MongoDB, Express.js, React.js, Node.js), the system offers functionalities including appointment booking, patient EMR management, to-do task scheduling, and session-based authentication. The paper outlines the system architecture, design methodology, core modules, and implementation details. A literature review of recent research validates the technological approach and addresses areas for future enhancement.

**Keywords:** Hospital Management System, Receptionist Module, MERN Stack, Secure Authentication, Node.js, React.js.

### I. INTRODUCTION

Healthcare institutions are rapidly adopting digital solutions to manage patient data, appointments, billing, and communication between departments. Receptionists are the first point of contact and need tools that are intuitive, secure, and connected with other healthcare modules. The aim of this project was to build a full-featured Receptionist Module with modern technologies and role-specific functionalities.

### II. LITERATURE REVIEW

Several researchers have proposed frameworks for hospital management using various technologies. A Django-based hospital system with HTML/CSS UI was presented in [1], while [2] introduced an IoT-enabled infrastructure for patient tracking. [3] proposed AI-driven chatbot integration, and [4] developed a full-stack web-based HMS. In [5], a cloud-based design using React and MongoDB was emphasized. These works validated our technology choice and influenced design considerations such as modularity, file handling, and dynamic appointment filtering.

### III. SYSTEM OVERVIEW AND METHODOLOGY

The Receptionist Module is built using the MERN stack and integrates with other system modules like Doctor, Patient, and EMR. The backend is powered by Express.js with session-based authentication using express-session. React.js is used on the frontend for dynamic UI components and responsive interfaces. Mongoose ODM is used for database schema management and validation. Multer handles secure file uploads, and AWS S3 is used for file storage. The application was deployed on AWS EC2 with NGINX and PM2.

#### Core Modules

1. To-Do Task System: Allows CRUD operations with filters by status and priority.
2. Appointment Booking: Supports OPD/IPD slots, doctor/patient mapping, and report upload.
3. Today's Appointments: Displays appointments with status dropdown, invoice filters, and avatars.
4. EMR Management: Includes patient history, document uploads, and filter/search options.
5. Profile Management: Receptionist profile editing with image handling, validation, and secure login.

### IV. IMPLEMENTATION & RESULTS

The module was successfully deployed with interactive UI, real-time updates, and secure route protection. Receptionists can now manage appointments and patient data efficiently. UI validation using Formik/Yup, performance-optimized API responses, and AWS-based deployment ensured scalability and reliability.

## V. DISCUSSION

The research papers reviewed validated the use of modern stacks like React, MongoDB, and session authentication. Compared to Django or IoT-centric systems, our solution is highly modular and scalable. Integration with other modules like Doctors and Patients adds value across departments. One limitation was the lack of mobile support which can be explored in future updates.

## VI. CONCLUSION

This project successfully delivered a robust Receptionist Module integrated into HealthBox. Through the use of modern tools and best practices, the system ensures efficient management of tasks, appointments, and EMR data for hospital receptionists. The research reviewed supported the design and highlighted future enhancements including voice-enabled chatbots and analytics.

## VII. REFERENCES

- [1] Design and Implementation of an E-hospital Management system using Django and CSS, IEEE, 2023. <https://ieeexplore.ieee.org/document/10112962/>
- [2] Hospital Management System Based on IoT, IEEE, 2023. <https://ieeexplore.ieee.org/document/10040306/>
- [3] AI Based Chatbot for Hospital Management System, IEEE, 2022. <https://ieeexplore.ieee.org/document/10007105/>
- [4] Web Based Hospital Management System, IEEE, 2024. <https://ieeexplore.ieee.org/document/10541355/>
- [5] Design and Development of Smart Hospital Management and Location Tracking System for People using Internet of Things, IEEE, 2024. <https://ieeexplore.ieee.org/document/10652750/>