

MEDIQUEUE: HOSPITAL QUEUE AND APPOINTMENT MANAGEMENT SYSTEM

Muthukumar M

Master of Computer Applications, Nehru Institute of Information Technology and Management, Tamil Nadu, India .
Reg. No: 721524622037 | Batch: 2024 –2026 – MENTOR –Mr. T.Thiyagarajan,Department of Computer Applications

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

ABSTRACT

MediQueue – Hospital Queue Management and Appointment Scheduling System is a web-based healthcare management platform designed to improve patient flow, appointment scheduling, and queue monitoring in hospitals and clinics. The system provides a centralized environment that connects patients, doctors, receptionists, and administrators, reducing waiting time and improving the efficiency of healthcare services. The system is developed using HTML, CSS, JavaScript, Node.js, Express.js, SQLite, and Socket.IO technologies. It includes modules such as user authentication, patient management, doctor management, appointment scheduling, real-time queue tracking, notifications, and administrative dashboards. Role-based access control ensures secure and efficient management of healthcare activities. The key feature of MediQueue is its real-time queue management system, which allows patients and hospital staff to monitor token status, queue position, and appointment updates instantly using Socket.IO technology. The platform enhances communication, reduces manual workload, and improves transparency in hospital operations. The system also provides dashboard analytics and reporting features that help administrators analyze appointments, patient records, doctor availability, and queue performance. The system contributes towards creating a more organized, transparent, and efficient healthcare environment.

I. INTRODUCTION

MediQueue is a web-based Hospital Queue Management and Appointment Scheduling System designed to simplify and improve the healthcare service process. The system provides a digital platform for managing patient appointments, monitoring queues, and improving communication between patients, doctors, and hospital staff.

The MediQueue system addresses the common challenges faced by hospitals, such as long waiting times, inefficient appointment handling, and lack of real-time communication between patients and healthcare staff. By replacing traditional manual queue management processes with a digital platform, the system improves operational efficiency and provides a smoother experience for both patients and healthcare providers.

The application is designed with a responsive and user-friendly interface that allows users to access healthcare services easily through different roles. Patients can book appointments, view queue status, and receive notifications, while doctors and administrators can manage schedules, patient information, and daily activities effectively. The integration of real-time updates ensures that users receive accurate information without delays.

MediQueue demonstrates how modern web technologies can be utilized to enhance healthcare management systems. The project provides a scalable solution that can be further extended with advanced features such as mobile application support, online consultations, AI-based queue prediction, and cloud-based data management.

II. RELATED WORK

Healthcare queue management and appointment scheduling systems have gained significant attention with the advancement of digital healthcare technologies. Several existing systems have been developed to reduce patient waiting time, improve appointment handling, and enhance communication between healthcare providers and patients. Traditional hospital management systems mainly focus on maintaining patient records and appointment details but often lack real-time queue monitoring capabilities.

Previous research has explored automated appointment scheduling systems that allow patients to book consultations digitally and reduce dependency on manual registration processes. These systems improve accessibility and convenience but may not provide effective solutions for managing patient flow after arrival at healthcare facilities. Real-time queue management solutions have been introduced to overcome the limitations of conventional token-based systems. Technologies such as web sockets and real-time communication frameworks enable instant updates regarding queue status, patient tokens, and doctor availability. These approaches improve transparency and help hospitals optimize their workflow. Modern healthcare platforms also incorporate dashboards and analytics features to support administrative decision-making. By analyzing appointment records, patient flow, and resource utilization, healthcare organizations can improve operational efficiency. However, many existing systems lack integration between appointment scheduling, queue monitoring, notifications, and role based management. MediQueue combines these essential features into a unified healthcare management platform. The system focuses on providing real-time queue tracking, automated appointment scheduling, secure user management, and efficient communication between patients and healthcare staff. By integrating these functionalities, MediQueue aims to provide a more organized, transparent, and patient-friendly healthcare experience.

III. SYSTEM OVERVIEW

This project titled MediQueue is a web-based Hospital Queue Management and Appointment Scheduling System developed to provide an efficient digital solution for managing patient appointments, hospital queues, and healthcare operations. The system creates a centralized platform that connects patients, doctors, receptionists, and administrators, enabling smooth coordination and effective management of hospital activities.

The system allows patients to register, log in, book appointments, and monitor their queue status in real time. Doctors can manage their schedules, view patient details, and update consultation status, while receptionists can handle appointments and queue operations. Administrators have access to dashboards that help monitor users, appointments, doctor availability, and overall system performance. MediQueue is developed using modern web technologies such as HTML, CSS, JavaScript, Node.js, Express.js, SQLite, and Socket.IO. The frontend provides a responsive and user-friendly interface, while the backend manages application logic, database operations, and real-time communication. Socket.IO enables instant queue updates, allowing patients and staff to receive live information without refreshing the page. The system reduces manual workload, minimizes patient waiting time, and improves transparency in healthcare services. With features such as authentication, role-based access control, notification management, and analytical dashboards.

IV. MODULE DESCRIPTION

1. User Authentication Module

This module provides secure login and authentication functionalities for all users of the system. It verifies user credentials, manages user sessions, and implements role-based access control to ensure that administrators, doctors, receptionists, and staff members can access only the features relevant to their responsibilities.

2. Patient Management Module

The Patient Management Module is responsible for maintaining patient information in a centralized database. It allows hospital staff to add, update, view, and manage patient records, including personal details, contact information, and medical-related information. This module helps in organizing patient data efficiently and reducing manual record-keeping.

3.Doctor Management Module

This module manages doctor-related information such as doctor profiles, specializations, departments, availability status, and schedules. It enables administrators to maintain accurate doctor records and helps patients book appointments with the appropriate healthcare professional.

4.Appointment Scheduling Module

The Appointment Scheduling Module allows patients and hospital staff to book, modify, and manage appointments. It helps avoid scheduling conflicts by checking doctor availability and maintaining appointment records systematically. This module ensures efficient allocation of consultation slots.

5.Queue Management Module

The Queue Management Module is the core component of the system. It automatically generates token numbers for patients and manages the consultation queue. The module tracks patient status and ensures a smooth flow of consultations while minimizing waiting times.

6.Real-Time Queue Tracking Module

This module uses Socket.IO technology to provide real-time updates regarding queue positions and token status. Patients can monitor their queue progress instantly without refreshing the application, improving transparency and reducing uncertainty.

7.Notification Management Module

The Notification Module sends important updates related to appointments, queue progression, doctor availability, and system alerts. It improves communication between patients and healthcare providers and ensures that users remain informed about relevant activities.

8.Dashboard and Analytics Module

This module provides administrators with statistical information and performance metrics through dashboards and reports. It displays data such as total appointments, patient count, doctor availability, queue status, and other operational insights that support decision-making and resource management.

V. METHODOLOGY

The development of MediQueue follows a systematic methodology that focuses on analyzing, designing, developing, testing, and maintaining a web-based Hospital Queue Management and Appointment Scheduling System. Initially, the requirements of patients, doctors, receptionists, and administrators are analyzed to identify the limitations of traditional manual queue management systems and define the necessary functionalities such as user authentication, appointment scheduling, queue monitoring, notifications, and dashboard management. Based on the collected requirements, the system architecture, database structure, user interface design, and module workflow are designed to ensure efficient data handling and smooth interaction between users. The system is then developed using modern web technologies including HTML, CSS, JavaScript, Node.js, Express.js, SQLite, and Socket.IO, where the frontend provides a responsive user interface and the backend manages business logic, database operations, security, and real-time communication. The implementation includes different modules for patient management, doctor management, appointment scheduling, queue tracking, and administrative control. After development, the system undergoes various testing processes such as functional testing, integration testing, and usability testing to verify accuracy, performance, and reliability. The real-time queue update mechanism and appointment scheduling features are carefully validated to ensure seamless operation. Finally, the system is prepared for deployment with continuous maintenance and future enhancements to improve healthcare service efficiency and provide a scalable solution for hospitals and clinics. The development of MediQueue follows a systematic methodology that focuses on analyzing, designing, developing, testing, and maintaining a web-based Hospital Queue Management and Appointment Scheduling System. The system architecture is designed to support scalability and adaptability for different healthcare environments. The implementation process follows continuous evaluation and improvement to ensure better user experience and operational efficiency.

Initially, the requirements of patients, doctors, receptionists, and administrators are analyzed to identify the limitations of traditional manual queue management systems and define the necessary functionalities such as user authentication, appointment scheduling, queue monitoring, notifications, and dashboard management. Based on the collected requirements, the system architecture, database structure, user interface design, and module workflow are designed to ensure efficient data handling and smooth interaction between users. The system is then developed using modern web technologies including HTML, CSS, JavaScript, Node.js, Express.js, SQLite, and Socket.IO, where the frontend provides a responsive user interface and the backend manages business logic, database operations, security, and real-time communication. The implementation includes different modules for patient management, doctor management, appointment scheduling, queue tracking, and administrative control. After development, the system undergoes various testing processes such as functional testing, integration testing, and usability testing to verify accuracy, performance, and reliability. The real-time queue update mechanism and appointment scheduling features are carefully validated to ensure seamless operation. Finally, the system is prepared for deployment with continuous maintenance and future enhancements to improve healthcare service efficiency and provide a scalable solution for hospitals and clinics. The methodology ensures that MediQueue provides a dependable platform for managing hospital activities effectively.

VI. REQUIREMENTS SPECIFICATION

The requirement specification defines the functional and non-functional requirements necessary for the development of the MediQueue – Hospital Queue Management and Appointment Scheduling System. The main objective of the system is to provide an efficient digital platform for managing patient appointments, hospital queues, and communication between patients, doctors, receptionists, and administrators. The requirements are identified based on the limitations of traditional manual healthcare management systems and the need for a faster, more organized, and transparent solution.

Software Requirements

MediQueue is a web-based healthcare management system designed to automate patient appointments, doctor schedules, and real-time queue monitoring. The system reduces waiting time by providing digital appointment booking, live queue updates, and efficient communication between patients, doctors, and hospital administrators..

Software Component	Specifications
HTML5	Used for creating the structure and content of web pages. Supports semantic markup for better accessibility and SEO.
CSS3	Used for styling and designing the user interface. Supports responsive design through Flexbox, Grid, and Media Queries.
JavaScript (ES6+)	Used for client-side scripting and adding interactivity to web pages. Supports DOM manipulation, events, and AJAX.
Node.js	JavaScript runtime environment used for server-side development. Provides an event-driven, non-blocking I/O model.
Express.js	Web framework for Node.js used to build RESTful APIs and handle routing, middleware, and HTTP requests.
SQLite	Lightweight relational database used for storing application data such as users, appointments, queue details, and system records.
Socket.IO	Real-time communication library that enables bi-directional communication between the server and clients for live queue updates and notifications.
Bootstrap 5	CSS framework used for designing responsive, mobile-first user interfaces with pre-built components.
Nodemon	Development tool used to automatically restart the server whenever changes are detected in the source code.

Hardware Requirements

The MediQueue – Hospital Queue Management and Appointment Scheduling System requires hardware components that support smooth development, deployment, and daily operation. The system should provide sufficient processing power, storage, and network connectivity to handle patient records, appointment scheduling, queue updates, and communication between users.

HARDWARE REQUIREMENTS	SPECIFICATIONS
Processor (CPU)	Intel Core i5 or equivalent (2.3 GHz or higher)
RAM	8 GB or higher
Storage	256 GB SSD or higher
Monitor	14-inch or above, HD (1366 x 768) or higher resolution
Keyboard	Standard USB Keyboard
Mouse	Standard USB Mouse
Network Connection	High-speed Internet (LAN/Wi-Fi)
Operating System	Windows 10/11 or Linux (64-bit)
Server (for Deployment)	Intel Core i5/i7, 16 GB RAM, 512 GB SSD or higher
Backup	External Hard Drive / Cloud Backup
Printer (Optional)	Laser or Inkjet Printer
Display for Queue (Optional)	LED Monitor / Smart TV (32 inch or above)
UPS Power Backup (Optional)	1000VA or higher

VII. SYSTEM IMPLEMENTATION

IMPLEMENTATION PHASES

System Implementation is the process of converting the designed system into a fully functional application. The MediQueue system was implemented using modern web technologies to provide an efficient hospital queue and appointment management solution.

1. Frontend Implementation

The user interface was developed using HTML, CSS, and JavaScript. The frontend provides interactive screens for user authentication, doctor management, patient registration, appointment scheduling, queue tracking, notifications, and dashboard analytics. The interface is designed to be user-friendly and responsive for easy navigation.

2. Backend Implementation

The backend was developed using Node.js and Express.js. It handles business logic, API requests, authentication, data processing, and communication between the frontend and database. RESTful APIs were created to manage users, doctors, patients, appointments, queues, notifications, and dashboard reports.

3. Database Implementation

The system uses SQLite as the database management system with the better-sqlite3 library. The database stores all application data, including user accounts, doctor details, patient records, appointments, queue information, and notifications. Primary and foreign keys are used to maintain relationships between tables.

4. Authentication and Security Implementation

Security was implemented using JWT (JSON Web Tokens) for user authentication and bcryptjs for password encryption. Role-based access control ensures that only authorized users can access specific system functionalities based on their assigned roles.

5. Real Time Queue Implementation

Socket.IO was integrated to provide real-time queue updates. When queue information changes, all connected users receive instant updates without refreshing the page. This feature improves communication and reduces patient waiting uncertainty.

VIII. SAMPLE OUTPUT

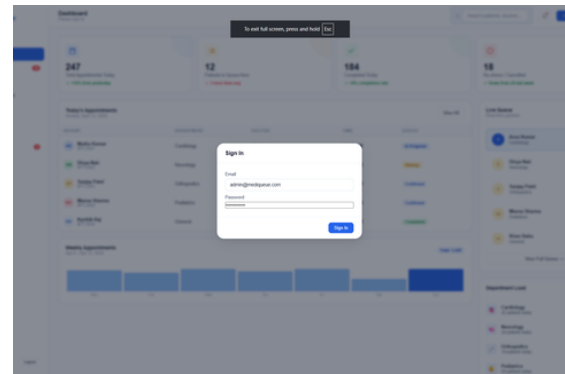


Fig -1 Login Page

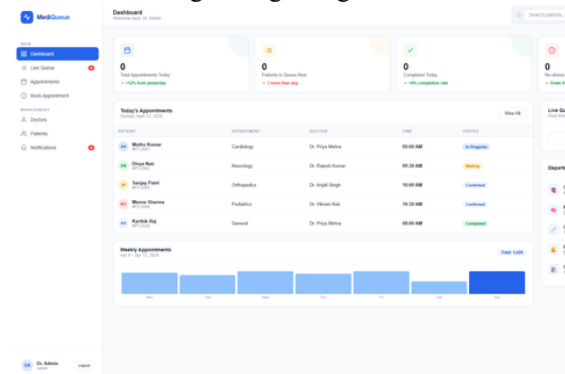


Fig -2 Dashboard Page

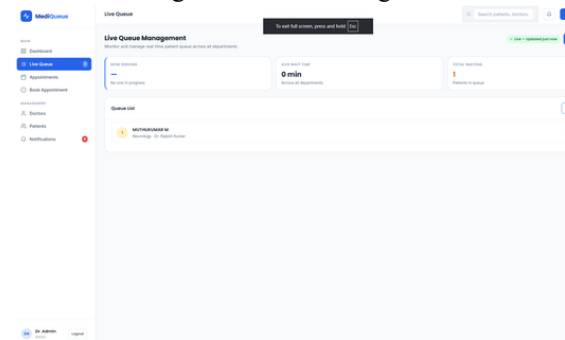


Fig -3 Live Queue

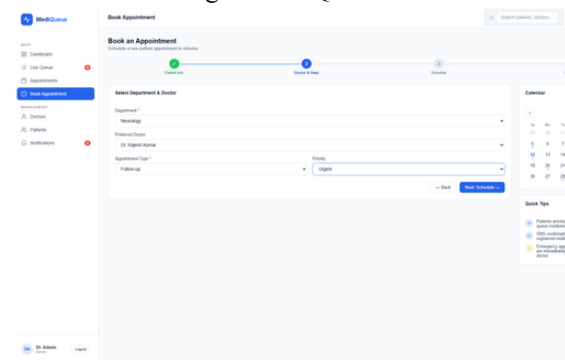


Fig -4 Book Appointments

Appointment ID	Patient Name	Doctor Name	Appointment Date	Status
1001	John Doe	Dr. John Doe	Apr 15, 10:30 AM	Completed
1002	Jane Smith	Dr. Jane Smith	Apr 15, 11:00 AM	Pending
1003	Mike Johnson	Dr. Mike Johnson	Apr 15, 11:30 AM	Completed
1004	Sarah Lee	Dr. Sarah Lee	Apr 15, 12:00 PM	Pending
1005	David Kim	Dr. David Kim	Apr 15, 12:30 PM	Completed
1006	Emily White	Dr. Emily White	Apr 15, 1:00 PM	Pending
1007	Robert Brown	Dr. Robert Brown	Apr 15, 1:30 PM	Completed
1008	Lisa Green	Dr. Lisa Green	Apr 15, 2:00 PM	Pending
1009	James Black	Dr. James Black	Apr 15, 2:30 PM	Completed
1010	Maria Garcia	Dr. Maria Garcia	Apr 15, 3:00 PM	Pending

Fig -6 Appointments

IX. CONCLUSIONS

The MediQueue – Hospital Queue and Appointment Management System was developed to provide a modern and efficient solution for managing hospital appointments and patient queues. The system successfully replaces manual processes with a digital platform that simplifies patient registration, doctor management, appointment scheduling, and real-time queue tracking. This helps reduce patient waiting time, improves resource utilization, and enhances the overall efficiency of hospital operations. The project was implemented using HTML, CSS, JavaScript, Node.js, Express.js, SQLite, JWT, bcryptjs, and Socket.IO. These technologies enabled the development of a secure, reliable, and responsive web application with real-time communication and role-based access control. The dashboard analytics and notification features further assist hospital administrators in monitoring daily activities and making informed decisions. Testing confirmed that all major modules function correctly and meet the specified requirements. The system provides accurate data management, secure authentication, and real-time updates, making it suitable for practical hospital environments. Overall, MediQueue demonstrates how modern web technologies can be effectively applied to improve healthcare services and hospital workflow. With future enhancements such as mobile applications, AI-based waiting time prediction, cloud deployment, and telemedicine support, the system can be further expanded into a comprehensive healthcare management solution.

ACKNOWLEDGMENT

I express my sincere gratitude to my institution and the Department of Master of Computer Applications (MCA) for providing me with the opportunity to undertake the project "MediQueue – Hospital Queue Management and Appointment Scheduling System." This project has been an invaluable learning experience that enhanced my technical knowledge and practical skills in software development..

REFERENCES

1. Sommerville, I. (2016). Software Engineering (10th Edition). Pearson Education.
2. Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach (9th Edition). McGraw-Hill Education.
3. MDN Web Docs. (2026). Retrieved from <https://developer.mozilla.org>
4. OWASP Foundation – Web Application Security. (2026). Retrieved from <https://owasp.org>
5. Duckett, J. (2014). JavaScript and jQuery: Interactive Front-End Web Development. Wiley.
6. Nixon, R. (2021). Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 (6th Edition). O'Reilly Media.
7. Freeman, E., Robson, E., Sierra, K., & Bates, B. (2021). Head First Design Patterns (2nd Edition). O'Reilly Media.
8. Express.js Official Documentation. (2026). Retrieved from <https://expressjs.com>
9. SQLite Documentation. (2026). Retrieved from <https://www.sqlite.org>
10. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). Database System Concepts (7th Edition). McGraw-Hill Education.