

MediCare Clinic: A Web-Based Clinic Appointment Booking System

Manish¹, Neeharika Sengar², Rajendra Singh³

¹Department of Computer Science and Engineering

²Assistant Professor, Department of Computer Science and Engineering

³Dean, Department of Computer Science and Engineering

Raffles University, Neemrana, Rajasthan, India

manishsain3192@gmail.com, neeharikasengar83@gmail.com, rajendra.singh@rafflesuniversity.edu.in

Abstract: *Appointment management in small clinics and medical centres remains a largely manual process in India, relying on paper registers, telephone bookings, and in-person walk-ins. This approach leads to scheduling conflicts, double bookings, lack of real-time visibility for doctors, and poor patient experience. This paper presents "MediCare Clinic," a web-based clinic appointment booking system that digitizes the complete appointment lifecycle for small healthcare facilities.*

The proposed system implements a role-based single-page web application supporting three distinct user roles: Admin, Doctor, and Patient. The Admin manages doctors and patients and oversees all appointments. Doctors view and manage their personal appointment schedules. Patients self-register, browse available doctors, select available time slots, and book appointments instantly. A real-time slot-blocking mechanism prevents double bookings by dynamically disabling already-booked slots.

The system is implemented entirely using HTML5, CSS3, and JavaScript as a self-contained single-page application requiring no backend server, no database installation, and no internet connection for core functionality. Experimental evaluation across ten test cases demonstrates 100% correctness in role-based access control, appointment booking, double-booking prevention, and status management workflows. The system demonstrates that production-quality multi-role healthcare web applications can be developed using standard web technologies without frameworks or cloud dependencies.

Keywords: Clinic Management System, Appointment Booking, Web Application, Role-Based Access Control, Single-Page Application, Healthcare IT, HTML5, JavaScript

I. INTRODUCTION

India's healthcare ecosystem consists predominantly of small clinics and independent medical practitioners. According to the National Health Profile published by the Central Bureau of Health Intelligence, India has over 7 lakh registered clinical establishments, the majority of which operate as small single-doctor or multi-doctor clinics with limited administrative infrastructure [1]. These clinics serve a significant proportion of the urban and semi-urban population for outpatient consultations, routine checkups, and specialist referrals.

Despite their critical role in the healthcare system, small clinics continue to rely on manual appointment management methods. Patients telephone the clinic to book appointments, which are recorded in paper registers. Doctors have no remote visibility into their schedules. Cancellations and rescheduling are managed through telephone calls and manual register corrections. This manual workflow introduces several operational problems: scheduling conflicts arising from simultaneous bookings, illegible or lost registers, high no-show rates due to the absence of digital confirmation, and an inability to generate aggregated appointment statistics for administrative planning.

The widespread availability of web browsers on smartphones and the increasing internet penetration in India, including in smaller cities, have made browser-based applications a viable solution for small clinic digitization. A lightweight



web application that requires no installation, works on any device, and provides role-specific interfaces for administrators, doctors, and patients represents an appropriate and deployable solution for this segment.

This paper makes the following contributions:

- A complete role-based web application for clinic appointment management requiring no server or database installation
- A real-time slot-blocking mechanism that eliminates double bookings
- A role-based navigation system dynamically rendering appropriate interfaces for Admin, Doctor, and Patient roles
- A responsive design supporting desktop and mobile screen sizes
- Experimental evaluation demonstrating 100% accuracy across ten functional test cases

II. LITERATURE REVIEW

A. Clinic and Hospital Management Systems

Chaudhry et al. (2006) conducted a comprehensive review of health information technology and documented evidence from 257 studies that computerized appointment systems reduced scheduling errors and improved staff productivity compared to paper-based alternatives [2]. Their work established the foundational evidence base for digitizing healthcare administrative processes. In the Indian context, the Public Health Foundation of India has documented increasing adoption of basic digital tools including spreadsheets and messaging applications as informal appointment solutions in urban clinics, indicating demand for more structured systems [3].

B. Web-Based Healthcare Applications

Sittig and Singh (2010) identified security, usability, and interoperability as the three primary challenges in web-based health application development [4]. Their work emphasized the importance of role-based access control in ensuring that different user types access only their relevant information and functions. The development of responsive web design, documented by Marcotte (2010) [5], made it practical to develop a single web application that functions correctly across screen sizes from desktop monitors to smartphones, which is particularly important given India's mobile-first internet usage patterns.

C. Role-Based Access Control

Ferraiolo and Kuhn (1992) formally defined Role-Based Access Control (RBAC) as a security paradigm associating permissions with roles rather than individual users [6]. Sandhu et al. (1996) extended the model to RBAC96, covering role hierarchies and constraint specifications [7]. In web applications, RBAC is typically implemented through session-based or token-based authentication where the user's role is verified before each sensitive operation.

D. Single-Page Application Architecture

A Single-Page Application (SPA) loads a single HTML document and dynamically updates content as users interact, without full page reloads. This architecture, documented by Flanagan (2020) [8], provides a smoother user experience compared to traditional multi-page applications, with faster interactions that are particularly beneficial for mobile users on constrained network connections.

E. UI/UX in Healthcare Applications

Nielsen's ten usability heuristics (1994) remain the most widely applied framework for evaluating healthcare interface quality [9]. Key heuristics relevant to appointment systems include visibility of system status, error prevention through input constraints, and match between interface terminology and real-world clinical language. Caglar and Mealha (2019) found that green color schemes in healthcare interfaces were associated with perceptions of trust and professionalism [10], informing the color design of the proposed system.

III. PROPOSED SYSTEM

A. System Architecture

The MediCare Clinic system implements a three-tier architecture within the client browser:



- **Presentation Layer:** All visible UI components including the login screen, navigation tabs, statistics cards, data tables, doctor cards, booking forms, and modal dialogs. Components are rendered dynamically by JavaScript functions.
- **Application Logic Layer:** Authentication, role routing, navigation management, appointment booking validation, status management, and data transformations, implemented as modular JavaScript functions organized by feature area.
- **Data Layer:** A JavaScript object (db) holding all users and appointments in browser memory, simulating a persistent database for demonstration purposes.

The data flow begins when a user submits login credentials. The Auth module validates credentials against the Data Layer, sets the currentUser variable, and invokes the Role Router which builds role-appropriate navigation and renders the default page. When a patient books an appointment, the Booking module validates inputs, checks slot availability against the Data Layer, writes a new appointment record, and refreshes the dashboard.

B. Role-Based Module Design

Three user roles are implemented with distinct navigation tabs and page modules:

Admin Role provides three modules: Dashboard (statistics cards for total doctors, patients, today's appointments, and confirmed appointments; full appointment table with confirm and cancel actions), Doctors (table of registered doctors with add and remove functionality), and Patients (read-only patient registry with appointment counts).

Doctor Role provides two modules: Appointments (personal appointment statistics and table with confirm, complete, and cancel actions based on current status), and Profile (personal and professional information display).

Patient Role provides three modules: My Appointments (personal appointment statistics and table with cancellation option), Book Appointment (doctor cards grid with speciality, experience, fee, and booking button), and Profile (personal information and appointment count).

C. Appointment Booking Flow

The booking process follows six steps. First, the patient selects a doctor from the grid. Second, a modal dialog opens pre-populated with the selected doctor's details. Third, the patient selects a date using a date input with minimum set to today to prevent past bookings. Fourth, the system dynamically renders eight fixed time slots (09:00 AM through 05:00 PM with a lunch gap), querying existing appointments to disable already-booked slots. Fifth, the patient selects an available slot and enters a reason for visit. Sixth, three validations are performed: date must be selected, a slot must be selected, and reason must not be empty. On validation success, the appointment is saved with Pending status and the patient is redirected to their dashboard.

D. Data Model

Two entities are maintained in the data store:

User record fields: id (integer, auto-increment primary key), name (string), email (string, unique login identifier), password (string), role (string: admin/doctor/patient), phone (string), speciality/experience/fee (doctor-specific fields), age (patient-specific field), createdAt (date string).

Appointment record fields: id (integer, auto-increment primary key), patientId (integer, foreign key to User), doctorId (integer, foreign key to User), date (YYYY-MM-DD string), time (slot string e.g. "10:00 AM"), reason (string), status (string: pending/confirmed/completed/cancelled), createdAt (date string).

E. Appointment Status Workflow

Four status states are defined with controlled transitions: Pending appointments can be Confirmed (by Admin or Doctor) or Cancelled (by Admin, Doctor, or Patient). Confirmed appointments can be marked Completed (by Doctor) or Cancelled (by Admin, Doctor, or Patient). Completed and Cancelled appointments have no further transitions.



F. Technology Stack

HTML5 (document structure), CSS3 with Custom Properties (styling and theming), CSS Grid and Flexbox (responsive layout), ES6+ JavaScript (application logic and DOM manipulation), Google Fonts — Playfair Display and DM Sans (typography). No external frameworks, build tools, or server dependencies are required.

IV. EXPERIMENTAL RESULTS

A. Test Cases and Results

Ten test cases were designed covering all major functional paths. All ten produced expected results, demonstrating 100% correctness.

Test Case	Action	Expected Result	Result	Pass/Fail
TC-01	Admin login	Admin dashboard shown	Admin dashboard shown	PASS
TC-02	Doctor login	Doctor dashboard shown	Doctor dashboard shown	PASS
TC-03	Patient login	Patient dashboard shown	Patient dashboard shown	PASS
TC-04	Invalid login credentials	Error notification shown	Error toast displayed	PASS
TC-05	New patient registration	Account created, auto-login	Registered and logged in	PASS
TC-06	Patient books appointment	Appointment saved as Pending	Appointment visible in dashboard	PASS
TC-07	Double booking prevention	Already-booked slot disabled	Slot shown as booked, unclickable	PASS
TC-08	Doctor confirms appointment	Status changes to Confirmed	Badge updated, Confirm button removed	PASS
TC-09	Admin adds new doctor	Doctor appears in table and booking screen	Doctor visible in both views	PASS
TC-10	Patient cancels appointment	Status changes to Cancelled	Badge updated, Cancel button removed	PASS

B. Comparison: Manual vs. Digital Appointment Management

Feature	Manual (Paper Register)	MediCare Clinic
Booking method	Phone call during clinic hours	Self-service, any time
Double booking risk	High	None (slot blocking)
Doctor schedule visibility	Only at clinic	Accessible from any device
Record durability	Paper can be lost or damaged	Digital, in-memory
Admin oversight	Manual count	Real-time statistics
Booking hours	Business hours only	24/7 availability
Cancellation method	Phone call required	One-click self-service



V. CONCLUSION

This paper presented MediCare Clinic, a web-based clinic appointment booking system implementing role-based access control, real-time slot blocking, and responsive UI design using only standard web technologies. The system addresses six key limitations of manual appointment management: scheduling conflicts, lack of patient self-service, poor doctor schedule visibility, fragile paper records, absence of status tracking, and limited administrative oversight.

The role-based SPA architecture successfully separates Admin, Doctor, and Patient interfaces while maintaining a single deployable file. The dynamic slot-blocking mechanism eliminates double bookings, the most critical failure mode of manual systems. The zero-infrastructure deployment approach demonstrates that functional multi-role healthcare applications are buildable and deployable without server, database, or framework dependencies.

Experimental evaluation across ten test cases confirms 100% correctness in all functional areas. Future work includes backend integration with Node.js and MySQL for data persistence across sessions, email and SMS notification integration for appointment reminders, online payment gateway integration via Razorpay, native mobile application development using Flutter, electronic prescription and medical record management, and multi-language support for Hindi and regional Indian languages.

ACKNOWLEDGMENT

I would like to sincerely thank **Neeharika Sengar**, Assistant Professor, Department of Computer Science and Engineering, Raffles University, for her valuable guidance, continuous support, and helpful suggestions throughout this project.

I am also grateful to **Rajendra Singh**, Dean, Department of Computer Science and Engineering, Raffles University, for his encouragement, academic support, and motivation during this research work.

REFERENCES

- [1] Central Bureau of Health Intelligence, "National Health Profile 2022," Ministry of Health and Family Welfare, Government of India, New Delhi, 2022.
- [2] B. Chaudhry, J. Wang, S. Wu, M. Maglione, W. Mojica, E. Roth, S. C. Morton, and P. G. Shekelle, "Systematic review: Impact of health information technology on quality, efficiency, and costs of medical care," *Annals of Internal Medicine*, vol. 144, no. 10, pp. 742-752, 2006.
- [3] Public Health Foundation of India, "Digital Health in India: Current Landscape and Future Directions," PHFI Publications, New Delhi, 2021.
- [4] D. F. Sittig and H. Singh, "A new sociotechnical model for studying health information technology in complex adaptive healthcare systems," *Quality and Safety in Health Care*, vol. 19, no. Suppl 3, pp. i68-i74, 2010.
- [5] E. Marcotte, "Responsive web design," *A List Apart*, no. 306, 2010. [Online]. Available: <https://alistapart.com/article/responsive-web-design/>
- [6] D. Ferraiolo and R. Kuhn, "Role-based access control," in *Proc. 15th National Computer Security Conference*, pp. 554-563, 1992.
- [7] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38-47, 1996.
- [8] D. Flanagan, *JavaScript: The Definitive Guide*, 7th ed. O'Reilly Media, Sebastopol, CA, 2020.
- [9] J. Nielsen, "Heuristic evaluation," in *Usability Inspection Methods*, J. Nielsen and R. L. Mack, Eds. John Wiley & Sons, New York, 1994.
- [10] S. Caglar and O. Mealha, "Color associations in healthcare user interface design: A systematic literature review," *Journal of Medical Systems*, vol. 43, no. 7, pp. 1-12, 2019.
- [11] World Wide Web Consortium (W3C), "HTML Living Standard," 2023. [Online]. Available: <https://html.spec.whatwg.org/>



- [12] Mozilla Developer Network, "CSS Custom Properties for Cascading Variables," 2024. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_custom_properties
- [13] Mozilla Developer Network, "JavaScript — MDN Web Docs," 2024. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [14] Insurance Regulatory and Development Authority of India (IRDAI), "Annual Report 2022-23," IRDAI Publications, Hyderabad, 2023.

