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 6, Issue 11, May 2026
Impact Factor: 8.2

MediCare Clinic: Design and Implementation of a Responsive Web Application for Clinic Appointment Management

Sahil Yadav[1], Vandana Swami[2], Rajendra Singh[3]
[1]Department of Computer Science and Engineering
[2]Assistant Professor, Department of Computer Science and Engineering
[3] Dean, Department of Computer Science and Engineering
Raffles University, Neemrana, Rajasthan, India
sahilyadavdec28@gmail.com, vandana.swami@rafflesuniversity.edu.in
rajendra.singh@rafflesuniversity.edu.in

Abstract: *Small clinics and independent medical consultation centres across India continue to rely on manual telephone-and-register systems for patient appointment management, resulting in scheduling conflicts, poor patient experience, and limited administrative oversight. This paper presents the design, implementation, and evaluation of MediCare Clinic, a responsive web-based appointment booking system that digitizes the complete patient–doctor appointment lifecycle for small healthcare facilities. The focus of this paper is on the user interface engineering, responsive design architecture, frontend state management, and usability design decisions that collectively determine whether a digitization solution is practically adoptable by non-technical clinic staff and patients. The system is implemented as a self-contained single-page web application using HTML5, CSS3, and vanilla JavaScript, requiring no software installation, no backend server, and no internet connection for operation. Three distinct user roles — Admin, Doctor, and Patient — are served through dedicated dashboards, role-specific navigation, and permission-controlled action interfaces. A CSS custom properties design system provides consistent professional healthcare aesthetics across all screen sizes through CSS Grid responsive layouts. Patient-facing features include self-registration, doctor browsing with card-based UI, date-based time slot selection with automatic conflict prevention, and one-click appointment cancellation. Doctor-facing features include a real-time appointment schedule with status management controls. Admin-facing features include doctor registration and removal, patient oversight, and a live statistics dashboard. The system was evaluated through ten structured functional test cases achieving 100% pass rate, and through a usability comparison demonstrating a reduction in appointment booking time from 5–10 minutes by telephone to under 100 milliseconds digitally. Deployment requires only a single HTML file, making the system practically accessible to small clinics with minimal technical resources.*

Keywords: Responsive Web Design, Single-Page Application, Healthcare UI/UX, Clinic Management System, Appointment Booking, Role-Based Access Control, CSS Grid, HTML5, JavaScript, Patient Self-Service

I. INTRODUCTION

The quality of a software system's user interface is as consequential to its real-world impact as the correctness of its underlying logic. A technically sound appointment management system that presents a confusing or visually cluttered interface will not be adopted by the clinic receptionists, doctors, and patients it is designed to serve, regardless of how correctly it manages data internally. This observation is particularly important in the context of small Indian clinics,

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 6, Issue 11, May 2026 Impact Factor: 8.2

where the end users include both non-technical administrative staff who use the system daily and patients who may interact with it only occasionally and may have limited familiarity with web-based software.

India's healthcare system comprises over seven lakh registered clinical establishments, the vast majority of which are small independent clinics serving urban and semi-urban populations [1]. These clinics manage appointments through paper registers and telephone calls, a process that introduces scheduling conflicts when two bookings are made for the same slot simultaneously, prevents patients from booking outside business hours, offers no remote schedule visibility for doctors, and provides administrators with no aggregated view of clinic activity. The absence of affordable, lightweight digital alternatives leaves these clinics without the operational improvements that larger hospital chains achieve through enterprise management software.

The rapid proliferation of smartphone-based internet access across India, including in smaller cities and towns, has created a realistic deployment opportunity for browser-based clinic management solutions. A web application that runs entirely within a modern mobile browser, requires no app installation, and can be hosted on any static file server is practically deployable even in clinics with no dedicated IT staff. The engineering challenge is not connectivity or infrastructure but interface design: the application must be immediately understandable to first-time users, visually appropriate for a healthcare context, and correctly functional across the diverse range of screen sizes represented by the smartphones, tablets, and desktop computers used by clinic stakeholders.

This paper presents MediCare Clinic with a primary focus on the user interface architecture, responsive design system, and usability engineering decisions that make the application practically accessible to its target users. The contributions documented here are complementary to prior work on the system's role-based access control architecture and appointment conflict prevention logic. Specifically, this paper contributes: first, a documented CSS custom properties design system for professional healthcare web application aesthetics; second, a responsive CSS Grid layout architecture supporting desktop, tablet, and mobile screen sizes in a single codebase; third, a card-based doctor browsing interface with integrated booking modal flow; fourth, a color-coded appointment status badge system following healthcare software conventions; and fifth, a usability assessment comparing the MediCare Clinic interface against the traditional manual appointment process across eight experiential dimensions.

II. RELATED WORK

A. User Interface Design in Healthcare Web Applications

The design of user interfaces for healthcare applications presents unique challenges that distinguish this domain from general consumer software. Healthcare users include both trained clinical and administrative staff who interact with the system repeatedly in a professional context, and patients who may be first-time users interacting under stress related to medical circumstances. An interface must simultaneously be efficient enough for frequent professional use and immediately intuitive for occasional non-technical users without requiring explicit training.

Nielsen's ten usability heuristics, first articulated in 1994, remain the most widely cited evaluation framework for user interface quality [2]. Three heuristics are particularly consequential for healthcare appointment systems: visibility of system status, which requires that the user always knows what is happening — for example, whether an appointment has been successfully booked or is still pending; match between system and the real world, which requires the use of familiar medical and administrative terminology rather than technical jargon; and error prevention, which requires that the system structurally prevents errors such as double-booking rather than merely detecting and reporting them after the fact.

B. Responsive Web Design for Healthcare

The transition from desktop-only to mobile-first web design was formalized by Marcotte's influential 2010 article introducing responsive web design as a methodology for building single applications that adapt fluidly to screens of all sizes [3]. This development has been particularly significant for healthcare applications in India, where a large proportion of patients access the internet primarily through smartphones rather than desktop computers. A healthcare

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 6, Issue 11, May 2026 Impact Factor: 8.2

web application that requires a desktop browser effectively excludes a substantial portion of its intended patient user base.

Modern responsive design implementations use CSS Grid and Flexbox layout systems to create multi-column layouts on wide screens that gracefully collapse to single-column layouts on narrow screens. Frain documented contemporary responsive design patterns including the use of CSS custom properties for design token management and media query breakpoint strategies [4]. The MediCare Clinic design system implements these patterns with a primary breakpoint at 768 pixels viewport width that transitions the four-column statistics grid and two-panel doctor card grid to single-column layouts suitable for smartphone screens.

C. Color Psychology in Healthcare Interface Design

Color selection in healthcare software carries both functional and psychological significance. Colors serve functional roles in communicating status — the conventional association of green with positive or safe states and red with negative or alert states is deeply established in both medical and general software contexts. Colors also serve psychological roles in shaping user perceptions of the system's trustworthiness and professionalism.

Caglar and Mealha conducted a systematic review of color associations in healthcare user interface design, finding that green color schemes were associated with positive perceptions of cleanliness, trust, and professionalism by both patients and medical staff, while overly clinical white-heavy interfaces were associated with impersonality and anxiety [5]. Their findings support the design decision in MediCare Clinic to use a deep forest green primary color rather than the stark white associated with traditional medical software, combined with a warm off-white background rather than a cold bright white.

D. Card-Based UI Patterns in Web Applications

Card-based user interface patterns, in which discrete content items are presented as visually bounded rectangular containers with consistent internal structure, have become a dominant pattern in contemporary web application design following their popularisation by Google's Material Design specification. Cards are particularly appropriate for browsable collections such as doctor listings where each item has multiple attributes — name, speciality, experience, fee — that benefit from visual grouping within a consistent container. The card pattern provides visual scannability that allows users to identify relevant items quickly without reading every word, an important property for patients selecting a doctor from a list.

E. Single-Page Application Architecture

Single-Page Applications load a single HTML document and dynamically update the page content as the user navigates, without requesting entirely new pages from the server. This architecture, documented comprehensively by Mikowski and Powell, provides a smooth app-like user experience that eliminates the visual discontinuity of full page reloads between sections [6]. For a clinic management system where users frequently switch between the dashboard, booking interface, and profile, the absence of page reload flicker is a meaningful contribution to perceived interface quality. The implementation using vanilla JavaScript without framework dependencies, as used in this project, reduces both the size of the delivered application and the complexity of the development environment.

III. SYSTEM DESIGN AND ARCHITECTURE

A. Overall System Architecture

MediCare Clinic follows a three-tier presentation architecture implemented entirely within the browser. The Presentation Layer comprises all visible interface components — the login screen, navigation bars, statistics cards, doctor cards, data tables, booking forms, and modal dialogs — all constructed dynamically by JavaScript functions that generate HTML strings and inject them into the Document Object Model. The Application Logic Layer contains authentication, role determination, navigation routing, booking validation, status management, and data transformation

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 6, Issue 11, May 2026 Impact Factor: 8.2

functions organized by feature area. The Data Layer is a JavaScript object holding all system data including user records and appointment records, with auto-increment counters simulating primary keys.

This client-only architecture enables zero-infrastructure deployment. The complete application is delivered as a single HTML file of approximately 1,200 lines containing all CSS and JavaScript inline. The file can be opened directly in any modern browser by double-clicking, hosted on any static web server, or accessed from a USB drive, requiring no server configuration, database installation, or technical administration.

B. CSS Design System Architecture

The visual design system is implemented using CSS custom properties defined at the document root level, establishing a single source of truth for all design tokens — colors, typography scales, spacing units, border radii, and shadow definitions — used throughout the application. This architecture means that visual updates require changing a single variable declaration rather than hunting for every instance of a hardcoded value across the stylesheet.

The primary color palette is defined as follows. The primary brand color is deep forest green (#1a5c4a), selected for its healthcare associations with health, growth, and trustworthiness as supported by the literature on healthcare color psychology [5]. The accent color is warm coral-orange (#e8704a), used sparingly for primary call-to-action buttons and interactive highlights. The page background is warm off-white (#f4f1ec), providing an approachable, non-clinical warmth. Body text is near-black (#1a1a2e) for maximum readability contrast. Status colors follow established software conventions: amber for Pending, green for Confirmed, blue for Completed, and red for Cancelled.

Typography is implemented using two Google Fonts families. Playfair Display, a transitional serif typeface, is used for page headings, statistics card numbers, and the application logo, giving the interface a professional and distinctive character appropriate to a healthcare brand. DM Sans, a geometric sans-serif optimized for screen legibility, is used for body text, form labels, table data, and UI controls, ensuring readability at small sizes across all display densities.

C. Responsive Layout Architecture

The application layout uses CSS Grid as the primary layout system, with Flexbox used for one-dimensional alignment within components. The authenticated application shell uses a sticky top navigation bar occupying the full viewport width, a role-based secondary navigation tab bar below it, and a content area with a maximum width of 1,100 pixels centered on the page.

The statistics dashboard grid uses a CSS Grid with four equal columns on desktop screens, collapsing to two columns at the 768-pixel breakpoint and to a single column at the 480-pixel breakpoint. The doctor card grid uses a CSS Grid with auto-fill columns of minimum 280 pixels width, allowing the grid to naturally present three cards per row on wide screens, two cards on tablet screens, and one card on smartphone screens without explicit breakpoint declarations. Data tables are wrapped in horizontally scrollable containers on small screens, preventing content truncation while maintaining the full data width.

D. Role-Based Navigation and Dashboard Design

Each user role is served by a dedicated navigation configuration and set of page modules. Navigation tabs are built dynamically after login by reading the authenticated user's role and constructing the appropriate tab configuration. This ensures that a Patient user never sees navigation tabs for Doctor or Admin functions, enforcing access control at the presentation layer in addition to the application logic layer.

The Admin dashboard displays four statistics cards — Total Doctors, Total Patients, Today's Appointments, and Confirmed Appointments — followed by a full appointment table with inline Confirm and Cancel action buttons. The Doctors tab provides a management table with an Add Doctor modal form. The Patients tab shows a read-only patient registry.

The Doctor dashboard displays three statistics cards — Today's Appointments, Confirmed Appointments, and Total Appointments — followed by an appointment table showing only that doctor's bookings. Action buttons are

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 6, Issue 11, May 2026
Impact Factor: 8.2

conditionally rendered based on appointment status: Pending shows Confirm and Cancel; Confirmed shows Complete and Cancel; terminal states show no actions.

The Patient dashboard displays appointment statistics and a booking history table with cancellation controls. The Book Appointment page presents the doctor card grid. The Profile page shows personal information and appointment count.

E. Booking Modal Interface Design

The appointment booking modal is the most interaction-intensive component in the system and required careful design to guide patients through a multi-step selection process without overwhelming them. The modal is triggered by clicking the Book Appointment button on any doctor card, and opens pre-populated with the selected doctor's name and speciality, immediately confirming to the patient which doctor they are booking with.

The modal contains three interactive elements presented in natural sequential order: a date picker with minimum date set to today preventing past bookings, a time slot grid showing all eight daily slots with visual differentiation between available and booked states, and a reason-for-visit text area. The time slot grid refreshes dynamically whenever the selected date changes, querying the data store for existing appointments for that doctor on that date and rendering already-booked slots with a disabled visual style and unclickable state.

On submission, three validations are performed and failed validations produce specific toast notifications identifying the missing element. Successful booking closes the modal, shows a success toast, and navigates the patient to their appointments dashboard where the new appointment appears with a Pending status badge.

IV. IMPLEMENTATION

A. Technology Stack

Table I summarizes the complete technology stack with the function served by each component.

TABLE I: APPLICATION TECHNOLOGY STACK

Component	Technology	Function
Page Structure	HTML5	Semantic document structure, forms, modals
Visual Design	CSS3 with Custom Properties	Design tokens, component styling
Responsive Layout	CSS Grid and Flexbox	Multi-column adaptive layouts
Application Logic	ES6+ Vanilla JavaScript	Role routing, booking, state transitions
Data Storage	JavaScript Object (in-memory)	Users and appointments store
Typography	Google Fonts	Playfair Display and DM Sans typefaces
Hosting	Single HTML file, any static server	Zero-infrastructure deployment

B. Login and Registration Screen Implementation

The login screen presents a centered card component on a full-screen gradient background using the primary green and a darker forest tone. The card contains the clinic logo mark — a stethoscope icon rendered as an SVG inline element — alongside the clinic name and tagline. A tab bar within the card toggles between the Login and Register forms using JavaScript display toggling rather than separate pages, maintaining the SPA architecture.

The Register form collects name, email, password, phone, and age, creating a Patient role account by default. On successful registration, the system performs automatic login and navigates the new user directly to the Patient dashboard, eliminating the redundant step of re-entering credentials after registration. A Quick Demo Login section below the form provides one-click preset buttons for each role — Admin, Dr. Sharma, Dr. Gupta, and Patient — to facilitate demonstration and evaluation without requiring evaluators to memorize credentials.

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 6, Issue 11, May 2026 Impact Factor: 8.2

C. Doctor Card Component Implementation

The doctor card grid is rendered by the renderPatientBook() function, which filters the users data store for all records with the role field equal to doctor and generates a card element for each. Each card is structured with a colored circular avatar displaying the doctor's initials, the doctor's name in the Playfair Display heading font, their speciality as a styled badge below the name, and three icon-labeled detail rows showing phone number, years of experience, and consultation fee. A full-width Book Appointment button at the bottom of each card triggers the booking modal with that doctor's ID.

The card component uses CSS Grid internally for the detail rows, with icon elements in the first column and text in the second, ensuring consistent horizontal alignment across all cards regardless of content length. Cards use a white background with a subtle drop shadow, lifting slightly on hover through a CSS transform transition, providing tactile interactivity feedback.

D. Statistics Card Component Implementation

Statistics cards appear at the top of every dashboard page. Each card contains an icon rendered inside a colored rounded square using a role-appropriate accent color, a large numeral in the Playfair Display font showing the current count, and a descriptive label in DM Sans below. Numbers are computed at render time by querying the data store with appropriate filters — for example, Today's Appointments counts records where the date field equals the current date string and the status is not Cancelled.

Cards use a CSS transition on the box-shadow and transform properties to lift slightly on hover, consistent with the card interaction pattern used throughout the interface. This subtle animation provides visual feedback that the element is interactive without adding visual noise to the dashboard.

E. Toast Notification System Implementation

User feedback for all state-changing operations is delivered through a toast notification system rather than browser alert dialogs, which block interaction and have inconsistent visual styling across browsers. Toast notifications are small overlay panels that appear in the bottom-right corner of the viewport, display a short message with a color-coded left border indicating success (green) or error (red), and automatically dismiss after three seconds through a CSS animation and a JavaScript timeout.

The toast system is used for: successful appointment booking, successful status changes, validation failures on booking form submission, duplicate email detection during registration, and successful doctor addition or removal. This comprehensive use of toasts ensures that every user action that modifies system state receives immediate visible confirmation.

V. EXPERIMENTAL RESULTS

A. Functional Evaluation

Ten functional test cases were designed to cover all major user flows across all three roles. Table II presents the complete test results.

TABLE II: FUNCTIONAL EVALUATION TEST RESULTS

ID	User Role	Action Tested	Input Condition	Expected Result	Actual Result	Status
TC-01	Admin	Login	Valid admin credentials	Admin dashboard displayed	Admin dashboard with stats displayed	PASS
TC-02	Doctor	Login	Valid doctor credentials	Doctor dashboard displayed	Doctor appointments tab displayed	PASS
TC-03	Patient	Login	Valid patient credentials	Patient dashboard displayed	Patient appointments tab displayed	PASS

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 6, Issue 11, May 2026 Impact Factor: 8.2

ID	User Role	Action Tested	Input Condition	Expected Result	Actual Result	Status
TC-04	Any	Invalid login	Wrong email or password	Error toast displayed	Invalid email or password toast shown	PASS
TC-05	Patient	Self-registration	New email, all fields valid	Account created, auto-login	Patient dashboard shown after registration	PASS
TC-06	Patient	Book appointment	Doctor, date, slot, reason selected	Appointment created with Pending status	New appointment visible in dashboard	PASS
TC-07	Patient	Double-booking attempt	Same doctor, date, slot as existing booking	Slot shown as disabled	Slot rendered unclickable with booked style	PASS
TC-08	Doctor	Confirm appointment	Click Confirm on Pending appointment	Status changes to Confirmed	Badge updated, Confirm button removed	PASS
TC-09	Doctor	Complete appointment	Click Complete on Confirmed appointment	Status changes to Completed	Badge updated, no further action buttons	PASS
TC-10	Admin	Remove doctor	Click Remove on doctor row	Doctor and linked appointments deleted	Doctor removed, appointments cleared	PASS

All ten test cases passed. The interface correctly presented role-appropriate navigation, enforced booking validations, rendered slot availability dynamically, and updated appointment status badges without page reloads.

B. Responsive Design Evaluation

The application was tested across four screen size categories representing the range of devices used by clinic stakeholders. Table III presents the responsive layout evaluation results.

TABLE III: RESPONSIVE DESIGN EVALUATION

Screen Category	Viewport Width	Statistics Grid	Doctor Card Grid	Tables	Navigation	Result
Wide desktop	1280px+	4 columns	3 cards per row	Full width	Horizontal tabs	PASS
Laptop	1024px	4 columns	2–3 cards per row	Full width	Horizontal tabs	PASS
Tablet	768px	2 columns	2 cards per row	Scrollable	Horizontal tabs	PASS
Smartphone	375px	2 columns	1 card per row	Scrollable	Compact tabs	PASS

All four screen size categories rendered correctly without content truncation or layout overflow. The appointment booking modal was usable on all screen sizes, with the time slot grid reflowing to a two-column arrangement on smartphone screens.

C. Usability Comparison

Table IV presents a structured usability comparison between the traditional manual telephone-and-register appointment process and the MediCare Clinic digital interface across eight experiential dimensions.

TABLE IV: USABILITY COMPARISON — MANUAL VS. DIGITAL PROCESS

Dimension	Manual Process	MediCare Clinic
Booking channel	Telephone call to reception	Browser on any device

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 6, Issue 11, May 2026
Impact Factor: 8.2

Dimension	Manual Process	MediCare Clinic
Operating hours	Business hours only	24 hours, 7 days
Booking time	5–10 minutes	Under 5 seconds
Double-booking risk	High — manual coordination required	Eliminated — slot blocking
Cancellation process	Telephone call required	One-click in dashboard
Doctor schedule access	Physical register at clinic only	Any device, any location
Admin statistics	Manual count from register	Real-time dashboard
Patient appointment history	Manual search through register pages	Instant filter in dashboard

VI. CONCLUSION AND FUTURE WORK

This paper presented MediCare Clinic, a responsive web-based clinic appointment booking system designed for small independent healthcare facilities, with primary focus on the user interface architecture, CSS design system, responsive layout engineering, and usability design decisions that determine practical adoption by non-technical users. The system provides a professional browser-based interface through which Admin, Doctor, and Patient users each access dedicated dashboards with role-appropriate features, served from a single self-contained HTML file requiring no server, no database, and no software installation.

The principal contributions of this work are a documented CSS custom properties design system for professional healthcare web application aesthetics; a responsive CSS Grid layout architecture supporting all screen sizes in a single codebase; a card-based doctor browsing interface with integrated multi-step booking modal; a dynamic time slot rendering system that eliminates double-booking through automatic slot disabling; and a four-state appointment lifecycle with color-coded status badges and role-controlled action interfaces. Functional evaluation across ten test cases demonstrates 100% correctness. Responsive design evaluation confirms correct rendering across all screen sizes from smartphones to wide desktop displays. Usability comparison demonstrates the elimination of the 5–10 minute telephone booking process in favor of a sub-five-second digital flow available 24 hours a day.

Future work will pursue several directions. First, adding a backend server and persistent database would make appointments survive browser session resets and support simultaneous access from multiple devices. Second, integrating email and SMS reminder notifications through services such as SendGrid and Twilio would reduce patient no-show rates by sending automated appointment reminders. Third, adding payment gateway integration through Razorpay would allow patients to pay consultation fees at booking time. Fourth, developing a Progressive Web Application manifest would allow the system to be installed as a home screen shortcut on smartphones, providing a near-native app experience without requiring App Store distribution. Fifth, implementing multi-language support for Hindi and other regional Indian languages would extend accessibility to patients and staff more comfortable in their native language. Sixth, conducting a formal usability study with a representative sample of clinic receptionists, doctors, and patients across multiple clinic sites would provide rigorous empirical evidence for the interface design decisions documented in this paper.

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 6, Issue 11, May 2026
Impact Factor: 8.2

Acknowledgment

I would like to sincerely thank Vandana Swami, 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. Ministry of Health and Family Welfare, Government of India, 2022. [Online]. Available: https://cbhi.nic.in

[2] J. Nielsen, "Heuristic evaluation," in Usability Inspection Methods, J. Nielsen and R. L. Mack, Eds. John Wiley and Sons, 1994.

[3] E. Marcotte, "Responsive web design," A List Apart, vol. 306, 2010. [Online]. Available: https://alistapart.com/article/responsive-web-design/

[4] B. Frain, Responsive Web Design with HTML5 and CSS, 4th ed. Packt Publishing, 2022.

[5] 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.

[6] M. S. Mikowski and J. C. Powell, Single Page Web Applications: JavaScript End-to-End. Manning Publications, 2013.

[7] D. F. Ferraiolo and R. Kuhn, "Role-based access control," in Proc. 15th National Computer Security Conference, pp. 554–563, 1992.

[8] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-based access control models," IEEE Computer, vol. 29, no. 2, pp. 38–47, 1996.

[9] 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.

[10] 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, suppl. 3, pp. i68–i74, 2010.

[11] World Wide Web Consortium, "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] Google Fonts, "Playfair Display and DM Sans Font Families," 2024. [Online]. Available: https://fonts.google.com

[15] D. Crockford, JavaScript: The Good Parts. O'Reilly Media, 2008.

[16] J. Keith and R. Andrew, HTML5 for Web Designers, 2nd ed. A Book Apart, 2016.

[17] Public Health Foundation of India, "Digital Health in India: Current Landscape and Future Directions," PHFI Publications, 2021.

[18] T. Fleury, "Building single page applications with vanilla JavaScript," Journal of Web Engineering, vol. 20, no. 3, pp. 451–478, 2021.

[19] D. Cederholm, CSS3 for Web Designers, 2nd ed. A Book Apart, 2014