

CLINIC APPOINTMENT BOOKING SYSTEM

MINI PROJECT REPORT

Submitted by

ATHIRA VIJU (CHN22CS033)

to

*APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of Degree in
Computer Science And Engineering*



**DEPARTMENT OF COMPUTER ENGINEERING
COLLEGE OF ENGINEERING CHENGANNUR, ALAPPUZHA**

APRIL 2025

**DEPARTMENT OF COMPUTER ENGINEERING
COLLEGE OF ENGINEERING CHENGANNUR
ALAPPUZHA**



CERTIFICATE

*This is to certify that the project report titled **MEDICARE** is a bonafide record of the **CSD 334 MINIPROJECT** presented by **ATHIRA VIJU (CHN22CS033)**, Sixth Semester Computer Engineering students, under my guidance and supervision. This project is submitted in partial fulfillment of the requirements for the award of the degree **Bachelor of Technology** of APJ Abdul Kalam Technological University.*

Smt. LEYA G

Project Guide
Assistant Professor
Dept. of Computer Engineering

Smt. GEETHA.S

Project Coordinator
Assistant Professor
Dept. of Computer Engineering

Prof. GOPAKUMAR G

Head of the Dept.
Associate Professor
Dept. of Computer Engineering

DECLARATION

I undersigned hereby declare that the project report “**Clinic Appointment Booking System**”, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under the supervision of **Smt.Geetha S**, Assistant Professor, Department of Computer Engineering. This submission represents my ideas in own words, and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma, or similar title of any other University.

ATHIRA VIJU CHN22CS033

Place: Chengannur

Date : 07/04/25

ACKNOWLEDGEMENT

This work would not have been possible without the support of many people. First and foremost, we give thanks to Almighty God who gave me the inner strength, resources, and ability to complete my project successfully.

We would like to thank **Prof. (Dr.) Hari. V S**, The Principal College of Engineering Chengannur, for providing the best facilities and atmosphere for the project completion and presentation. Special thanks also goes to HOD **Prof. Gopakumar G**, Associate Professor, Department of Computer Engineering, for his exceptional support, guidance, and encouragement throughout the project. We would also like to thank our project coordinator **Smt. Geetha S**, Assistant Professor, Department of Computer Engineering, **Smt. Betty James**, Assistant Professor, Department of Computer Engineering, **Smt. Leya G**, Assistant Professor, Department of Computer Engineering who also took on the role of our project guide, for her extended help and support during the project.

We would like to thank our dear friends and faculties for extending their cooperation and encouragement throughout the project work, without which we would never have completed the project this well. Thank you all for your love and also for being very understanding.

ATHIRA VIJU CHN22CS033

ABSTRACT

The Clinic Appointment System is an efficient and user-friendly software solution designed to streamline appointment management in a healthcare environment. It provides a centralized platform for patients, doctors, and administrators to interact effectively. Patients can register, search for doctors, view available slots, and seamlessly book appointments. Administrators can manage their schedules, access patient appointments, and update their availability. The system ensures a smooth workflow by incorporating modules for appointment booking, notifications, and feedback management. It leverages modern technologies, including a responsive user interface, a secure backend, and a relational database for reliable data storage and retrieval. Additional features such as email notifications, role-based access control, and patient history management enhance its functionality. By automating routine tasks, the Clinic Appointment System reduces administrative workload, minimizes scheduling errors, and improves patient satisfaction, making it an essential tool for modern healthcare facilities.

CONTENTS

Declaration	i
Acknowledgement	ii
Abstract	iii
List of Figures	vii
Abbreviations	viii
1 INTRODUCTION	1
1.1 Project Area	1
1.2 Objectives	2
1.3 Problem Definition	3
1.3.1 Existing System	3
1.4 Limitations	4
1.4.1 Problem Statement	4
1.4.2 Proposed System	5
1.4.2.1 Real-time Scheduling	5
1.4.2.2 Automated OP Number Generation	5
1.4.2.3 Interactive Chatbot Support	5
1.4.2.4 Admin Panel	5
1.4.2.5 Email Confirmation System	5
1.4.2.6 Slot Management	6
1.4.2.7 Appointment Management	6
2 LITERATURE REVIEW	7
2.1 Patient Appointment and Scheduling System (2019)	7
2.2 Design and Implementation of Clinic Appointment Registration System (2013)	7
2.3 Online Doctor Appointment Booking System (2022)	8
2.4 Doctor Appointment Online Booking System (2018)	8

3 PROJECT DESIGN	10
3.1 System Architecture	10
3.2 Data Flow Diagram	12
3.2.1 User Query Processing	12
3.2.2 Database Retrieval	12
3.3 Entity-Relationship (ER) Diagram	12
3.3.1 Key Entities & Attributes	13
3.3.2 Relationships	13
3.4 Dataflow Diagram (Level 0)	14
3.5 Dataflow Diagram (Level 1)	14
3.5.1 Actors & Entities	15
3.5.2 Flow of Data	15
3.6 Use Case Diagram	15
3.7 Technology Stack	16
3.7.1 Front End	16
3.7.2 Back End	17
3.8 IMPLEMENTATION	17
3.9 RESULTS AND DISCUSSION	19
3.9.1 Home Page	19
3.9.2 Chatbot	20
3.9.3 About Page	21
3.9.4 Departments Page	21
3.9.5 Department wise doctor filtering	22
3.9.6 Doctor Profile	22
3.9.7 Book Appointment	23
3.9.8 OP Generation	23
3.9.9 Admin Login Page	24
3.9.10 Admin Dashboard	24
3.9.11 Admin-Appointments	25
3.9.12 Manage Doctor Profile Page	25
3.9.13 Add Doctor via Admin Panel	26
3.9.14 Doctor Duty Management Page	26
3.9.15 Add Schedule for Doctor Duty Management	27

3.9.16	Delete Schedule	27
3.9.17	Appointment Confirmation Mail	28
3.9.18	Database: Doctor's Table	28
3.9.19	Database: Appointment Table	28
3.9.20	Database: Department Table	29
3.9.21	Database: Availability Table	29
3.10	Advantages	29
3.11	Limitations	30
4	CONCLUSION AND FUTURE SCOPE	32
4.1	Future Scope	32
4.1.1	Integration with Telemedicine Services	32
4.1.2	AI-Driven Doctor Recommendations	32
4.1.3	Multi-Clinic and Multi-Specialty Support	32
4.1.4	Secure Payment Gateway Integration	32
4.1.5	Advanced Chatbot with Voice Support	32
4.1.6	Integration with Electronic Health Records (EHR)	33
REFERENCES		34

LIST OF FIGURES

3.1	System Architecture	10
3.2	ER Diagram	13
3.3	Data Flow Diagram Level 1	14
3.4	Use case Diagram	16
3.5	Home Page	19
3.6	Selecting department - doctor Page	20
3.7	Selecting department - doctor Page	20
3.8	About Page	21
3.9	Departments Page	21
3.10	Dept based doctor filtering	22
3.11	Doctor Profile Page	22
3.12	Book Appointment Page	23
3.13	OP Generation Page	23
3.14	Admin Login Page	24
3.15	Admin Dashboard Page	24
3.16	Admin-Appointment	25
3.17	Manage doctor profile Page	25
3.18	Admin Dashboard: Add Doctor	26
3.19	Admin Dashboard: Doctor Duty management	26
3.20	Doctor Duty management : Add Schedule	27
3.21	Schedule Added	27
3.22	Doctor Duty management : Delete Schedule	27
3.23	Confirmation mail	28
3.24	Doctor's Table	28
3.25	Appointment Table	28
3.26	Department Table	29
3.27	Availability Table	29

ABBREVIATIONS

1. OP: Outpatient Number
2. UI: User Interface
3. DB: Database
4. API: Application Programming Interface
5. CRUD: Create, Read, Update, Delete
6. CSS: Cascading Style Sheets
7. HTML – HyperText Markup Language
8. JS – JavaScript

CHAPTER 1

INTRODUCTION

In today's fast-paced world, healthcare facilities face challenges in managing patient appointments efficiently. Traditional methods of booking appointments often lead to long waiting times, scheduling conflicts, and administrative burdens. A Clinical Booking Appointment System offers a digital solution to streamline the appointment scheduling process, ensuring convenience for both patients and healthcare providers. By integrating technology into healthcare management, the system enhances accessibility, reduces manual errors, and improves overall patient satisfaction. With the growing need for automation in medical services, an online appointment booking system provides a structured and efficient way to manage consultations, making healthcare services more organized and patient friendly.

1.1 Project Area

The Clinical Booking Appointment System is part of the healthcare IT solutions domain, which focuses on using digital advancements to enhance the efficiency and accessibility of medical services. In today's fast-paced world, traditional appointment booking methods rely heavily on manual registrations, paper-based records, and long waiting times, causing inconvenience to both patients and healthcare staff. As technology continues to evolve, healthcare facilities must adopt modern solutions that streamline processes and improve patient care. Our project aims to bridge this gap by providing a web-based appointment scheduling system that simplifies the booking process while ensuring smooth clinic operations.

One of the primary challenges in traditional appointment booking is the lack of real-time availability tracking. Patients often face uncertainty when trying to secure an appointment, leading to frustration and inefficiencies in clinic workflow. With the proposed system, users can easily check doctor availability, select a suitable time slot, and book an appointment—all in just a few clicks. This not only reduces patient waiting times but also allows clinics to manage schedules effectively without the risk of double bookings or mismanagement.

Apart from appointment booking, another critical aspect of the system is clinic administration and management. Clinics often struggle with organizing patient records, managing doctor schedules, and handling leave requests. Our system incorporates an admin panel where clinic staff can

oversee appointments, adjust schedules, and update doctor availability. This ensures that both patients and clinic administrators benefit from a well-organized and structured system that minimizes errors and optimizes daily operations.

Additionally, to enhance user interaction and support, the system includes a chatbot assistant that provides instant responses to common patient queries. Whether users need information about available doctors, booking procedures, or clinic operating hours, the chatbot offers a convenient and efficient way to get answers. By integrating this feature, the system not only automates appointment scheduling but also improves accessibility for users, making healthcare services more patient-friendly and digitally advanced.

1.2 Objectives

The primary objective of this project is to develop an intuitive, efficient, and user-friendly online appointment booking system that overcomes the limitations of traditional methods. By integrating modern web technologies, the system ensures a seamless experience for both patients and clinic administrators. The following objectives define the core functionalities and advantages of the system:

1. Enabling online appointment booking: Patients can conveniently view doctor availability, select a suitable time slot, and confirm appointments without physically visiting the clinic. This reduces waiting times and eliminates the hassle of long queues.

2. Automated OP number generation: For new patients, the system automatically generates an Outpatient (OP) number, reducing the burden of manual registration and streamlining the patient onboarding process.

3. Interactive chatbot support: A smart chatbot is integrated into the system to assist patients with common queries related to clinic details, doctor availability, and appointment booking procedures.

4. User-friendly navigation: The website is designed with an intuitive navigation bar that includes sections such as Departments, Doctors, About, and Home, enabling users to quickly find the information they need. Clicking on these sections smoothly scrolls to the corresponding page, enhancing user experience.

5. Email confirmation system: Once an appointment is successfully booked, the system automatically sends a confirmation email to the patient containing all relevant appointment details.

6. Appointment slot management: Each time slot is limited to a maximum of two patients,

ensuring an organized and evenly distributed schedule. This prevents overbooking and allows doctors to manage their time efficiently.

7. Efficient clinic management: The system includes a dedicated admin panel that allows clinic staff to manage doctor schedules, handle leave requests, and track appointments with ease. This ensures smooth clinic operations and optimal utilization of resources.

8. Doctor Filtering by Department : Finding the right doctor can be difficult, especially in multi-specialty clinics. Our system allows patients to filter doctors based on their department, such as dentistry, general physician, or dermatology. This feature helps users quickly locate the right specialist.

1.3 Problem Definition

The healthcare industry, particularly small and medium-sized clinics, has struggled with outdated and inefficient appointment booking systems. Traditional methods typically rely on manual registration, either through phone calls or in-person visits, which results in significant delays, errors, and overbooking. These systems often lack real-time scheduling, making it difficult for patients to secure an appointment at their preferred time. The absence of a streamlined process also leads to confusion among patients and clinic staff, contributing to longer waiting times, miscommunication, and overall dissatisfaction.

Additionally, without an automated system, clinic staff is burdened with managing patient records, doctor schedules, and appointment details manually, further increasing the likelihood of mistakes and administrative inefficiencies. This not only hampers the clinic's ability to provide timely care but also affects patient trust and satisfaction.

Therefore, the primary problem is the lack of a cohesive, real-time system that can integrate appointment scheduling with doctor availability, patient records, and clinic operations. This gap calls for a modern, web-based solution that addresses the pain points of manual systems and enhances both patient experience and clinic management efficiency. The solution should be able to reduce human error, ensure optimal use of resources, and create a seamless interaction between patients and healthcare providers.

1.3.1 Existing System

In many clinics, the traditional appointment booking system relies heavily on manual registrations, where patients either visit the clinic in person or call receptionists to schedule an appointment. This

method often leads to long waiting times, misplaced records, and inefficiencies in managing doctor schedules. Additionally, receptionists must manually record patient details, increasing the risk of human errors and overlapping appointments.

While online booking systems have been introduced in some clinics, they often lack real-time scheduling features, leading to double bookings and appointment conflicts. Doctors are frequently overbooked, preventing them from dedicating sufficient time to each patient, which can compromise the quality of healthcare provided. Moreover, these systems typically lack efficient administrative controls, making it difficult for clinic staff to effectively manage doctor availability, patient records, and cancellations.

Without a structured and intelligent system, clinics face operational inefficiencies that result in patient dissatisfaction, increased workload for staff, and poor resource utilization.

1.4 Limitations

- **Manual Booking Process** – Traditional methods require patients to visit the clinic or call reception, leading to long waiting times.
- **Overlapping Appointments** – Lack of real-time scheduling results in double bookings, causing appointment conflicts.
- **Inefficient Time Management** – Doctors may be overbooked, reducing the time they can dedicate to each patient.
- **Limited Administrative Control** – Clinic staff cannot efficiently manage doctor availability, patient records, or schedule changes.
- **No Real-Time Updates** – Patients cannot check live doctor availability, leading to uncertainty and last-minute schedule issues.
- **High Risk of Human Errors** – Manual data entry increases the chance of mismanagement, leading to misplaced records and scheduling mistakes.
- **Longer Waiting Times** – Due to poor scheduling, patients often wait for extended periods before seeing a doctor.

1.4.1 Problem Statement

This project aims to develop a web-based appointment booking system that enables patients to easily schedule appointments with real-time access to doctor availability and time slots. It will automate

tasks like patient registration and appointment confirmation, reducing staff workload and enhancing clinic efficiency.

1.4.2 Proposed System

The proposed solution is an intuitive, web-based appointment booking system designed to automate and streamline clinic appointment processes. The system will feature:

1.4.2.1 Real-time Scheduling

- Patients can view available time slots for their preferred doctor and book appointments instantly without waiting for confirmation from clinic staff.
- This ensures that there are no conflicts or overbookings, improving efficiency and reducing patient wait times.

1.4.2.2 Automated OP Number Generation

- New patients will automatically receive an Outpatient (OP) number during the booking process.
- This eliminates the need for manual entry and reduces the chances of errors in record-keeping.

1.4.2.3 Interactive Chatbot Support

- A chatbot integrated into the system will answer common questions regarding the clinic, doctors, and available appointment slots.
- This feature will assist patients in navigating the system and help them book appointments efficiently.

1.4.2.4 Admin Panel

- The system will include a secure admin panel, allowing clinic staff to manage doctor schedules, view patient bookings, and monitor appointment statuses.
- Admins can manage appointments based on availability, ensuring smooth clinic operations.

1.4.2.5 Email Confirmation System

- Once an appointment is successfully booked, patients will receive an automatic email confirmation.
- The email will contain all relevant details, such as appointment time, doctor name, and date,

ensuring clear communication.

1.4.2.6 Slot Management

- Each time slot will be able to accommodate a maximum of two appointments.
- This ensures better time management and prevents overbooking, allowing doctors to attend to each patient effectively.

1.4.2.7 Appointment Management

- Clinic administrators will be able to view and manage all appointments in real-time.
- They can also generate reports to track appointments, patient visits, and doctor availability, improving clinic efficiency.

This solution aims to optimize the workflow of the clinic, reduce human error, and offer patients a hassle-free experience when booking appointments. By automating key processes, the system will free up valuable administrative time, allowing the clinic to focus on providing high-quality care.

CHAPTER 2

LITERATURE REVIEW

2.1 Patient Appointment and Scheduling System (2019)

The research, conducted by Akinode, John Lekan, and Oloruntoba S.A, introduces a web-based system that leverages database management, online scheduling, and real-time appointment tracking. The system provides a structured and efficient way for patients to book appointments online, eliminating the need for manual scheduling and reducing human errors. By automating the process, hospitals and clinics can better manage patient inflow and reduce long waiting times. The system also integrates features such as automated notifications and reminders, ensuring patients do not forget their scheduled appointments. Additionally, it allows doctors to manage their schedules dynamically, making it easier to optimize consultation timings and prevent overbooking.

Despite its advantages, the system relies heavily on a stable internet connection, which can be a drawback in areas with poor network coverage. Technical issues such as software bugs, server downtimes, or system crashes can disrupt appointment management, causing delays in patient care. The initial setup and maintenance costs may be high, particularly for smaller clinics that lack the necessary resources. Furthermore, storing patient records in an online database raises concerns about data security and privacy. Without proper encryption and authentication protocols, sensitive patient information could be vulnerable to cyber threats, necessitating stringent security measures to safeguard data integrity.

2.2 Design and Implementation of Clinic Appointment Registration System (2013)

Authored by Xiuju Zhan and Xiufeng Liu, contribute a Browser-Server (B/S) model that streamlines the patient registration and appointment booking process. The system allows patients to register and schedule appointments through an online interface, reducing the burden on hospital administrative staff and minimizing paperwork. Real-time tracking of appointment data ensures that doctors and hospital management can efficiently allocate resources and manage patient flow. The system also helps eliminate duplicate bookings and scheduling conflicts, leading to a smoother appointment process. By digitizing patient records, hospitals can maintain an organized and centralized database

for better record-keeping.

However, system failures or unexpected technical glitches could lead to data loss, disrupting hospital operations. To prevent such issues, clinics and hospitals need robust backup solutions and IT support. Additionally, the system might be challenging for elderly patients or individuals unfamiliar with technology, as they may struggle to navigate the online booking process without assistance. Another limitation is that the system primarily focuses on internal clinic appointments and does not support external consultations, making it less adaptable for multi-specialty hospitals that collaborate with external healthcare providers. Implementing such a system requires training and awareness programs to ensure that both patients and medical staff can utilize it effectively.

2.3 Online Doctor Appointment Booking System (2022)

Authored by Geeta and Shivagona Patil introduces an object-oriented appointment booking system with three primary components: doctors managing their schedules, patients booking appointments and accessing prescriptions, and administrators overseeing all activities. The system is designed to offer structured booking, personalized access, and smooth coordination among patients, doctors, and admins. Patients can conveniently book, cancel, or reschedule their appointments while receiving real-time updates through email or SMS notifications. The system also allows users to download appointment confirmations in PDF format, ensuring they have a record for future reference. Doctors can update their availability dynamically, preventing scheduling conflicts and reducing the risk of overbooking.

Despite its benefits, the system's complexity poses maintenance challenges, requiring regular updates and technical support. If administrators are slow to respond, appointment confirmations may be delayed, leading to inconvenience for patients. The system also raises concerns about data privacy, as administrators have broad access to patient records, which could pose a security risk if not managed properly. Integration with existing hospital management software might be difficult, requiring additional customization and investment. While the system improves overall efficiency, it demands a well-structured workflow to ensure smooth operations and avoid mismanagement of appointments.

2.4 Doctor Appointment Online Booking System (2018)

Authored by Ms. Sanjeevani P. Avhale, Ms. Wrushali R. Ajabe, and Ms. Pallavi A., presents a mobile-based appointment booking application that incorporates features such as user authentication, area-based doctor filtration, booking requests, confirmations, and notifications. The app

categorizes doctors based on their specialty and location, helping patients find the most suitable healthcare provider in their vicinity. Real-time appointment updates, reminders, and notifications ensure that patients do not miss their scheduled visits. Additionally, the system allows doctors to manage their schedules efficiently, enabling structured appointment bookings and reducing last-minute cancellations. By digitizing the booking process, the app enhances the accessibility and convenience of healthcare services.

However, the system requires installation on a mobile device, which may be a barrier for patients who are not familiar with mobile applications or do not own smartphones. The app is highly dependent on internet connectivity, and any network disruptions can affect appointment scheduling. Regular updates and system maintenance are necessary to ensure smooth operation, requiring ongoing technical support. Furthermore, manual doctor approvals for appointments may introduce delays, affecting the efficiency of the system. Data security is another significant concern, as mobile applications are more vulnerable to cyber threats, making it essential to implement strong encryption and authentication measures to protect patient information.

CHAPTER 3

PROJECT DESIGN

3.1 System Architecture

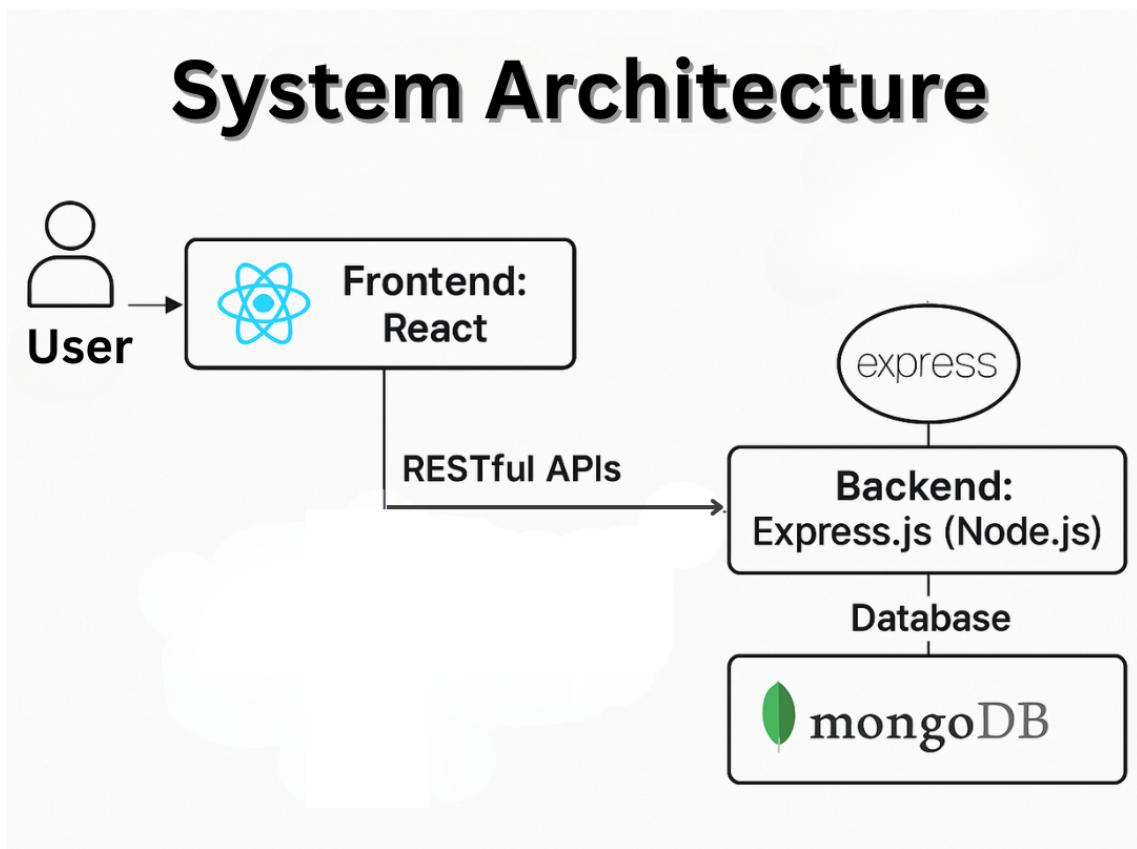


Figure 3.1: System Architecture

The system architecture follows a MERN (MongoDB, Express.js, React, Node.js) stack, ensuring a structured and efficient framework for the Clinic Appointment Booking System.

- **Frontend (React.js):** The user interacts with the system through a React.js interface, providing a seamless and dynamic experience.

- **Backend (Express.js with Node.js):** The backend is built using Express.js, a lightweight framework running on Node.js, which handles user requests and business logic.
- **Database (MongoDB):** The system utilizes MongoDB, a NoSQL database, to store and manage patient details, appointment records, doctor schedules, and other essential data.
- **Communication:** The frontend communicates with the backend via RESTful APIs, ensuring smooth data exchange and efficient processing.

This architecture enhances scalability, improves performance, and ensures a user-friendly experience for both patients and administrators.

3.2 Data Flow Diagram

3.2.1 User Query Processing

- When a patient initiates a request (e.g., booking an appointment, checking doctor availability), the system captures the input through the frontend (**React.js**).
- The request is sent via RESTful APIs to the backend (**Express.js with Node.js**), where it is processed.
- The system validates the request, checking for required details such as patient information, preferred doctor, available time slots, and department.
- If the request is incomplete, an error message is returned to the user; otherwise, it proceeds to the database retrieval step.

3.2.2 Database Retrieval

- Once the request is validated, the backend queries the MongoDB database to retrieve relevant information.
- For appointment booking, the system checks the Availability table to confirm if the doctor is free at the requested time.
- If the doctor is available, an appointment entry is created, and the details are stored in the Appointments table.
- The system updates doctor schedules to reflect the new booking.
- Confirmation details (including appointment time, doctor details, and email notification) are generated and sent back to the user.

3.3 Entity-Relationship (ER) Diagram

The Entity-Relationship (ER) Diagram defines the database structure, outlining how different entities relate to each other.

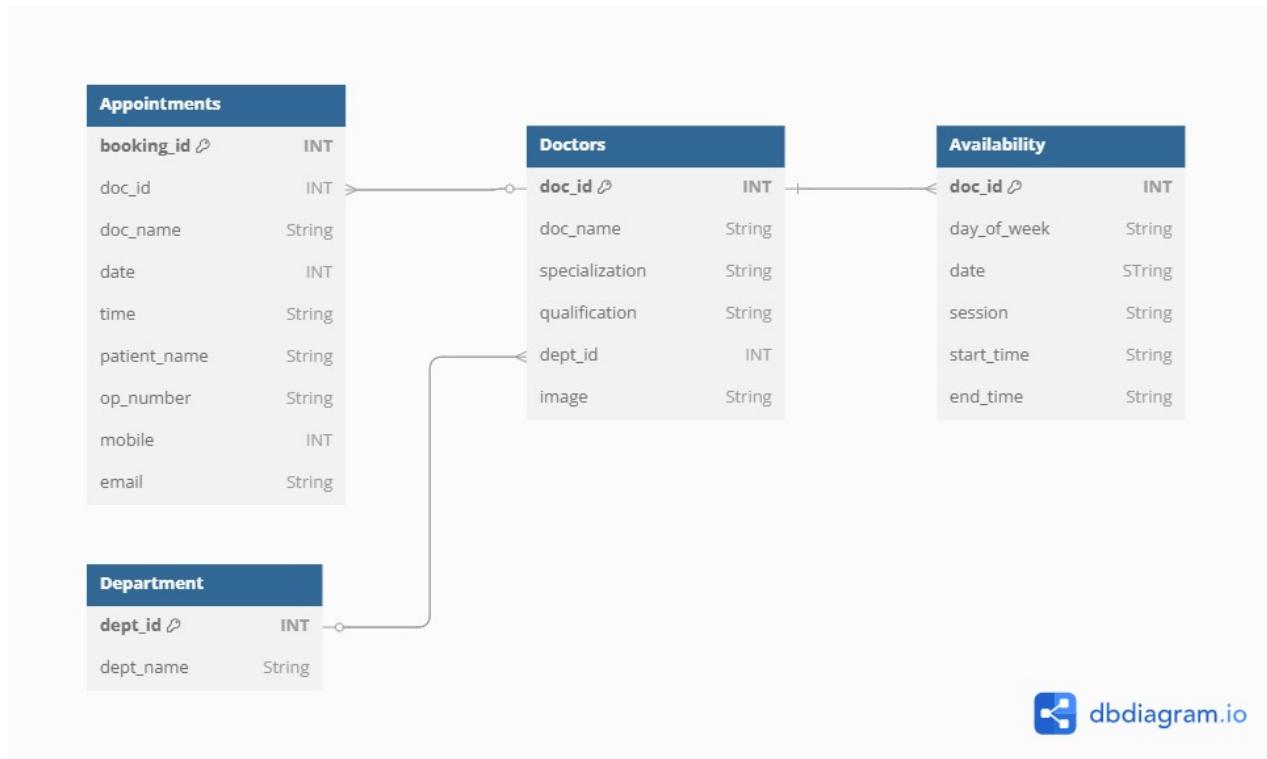


Figure 3.2: ER Diagram

3.3.1 Key Entities & Attributes

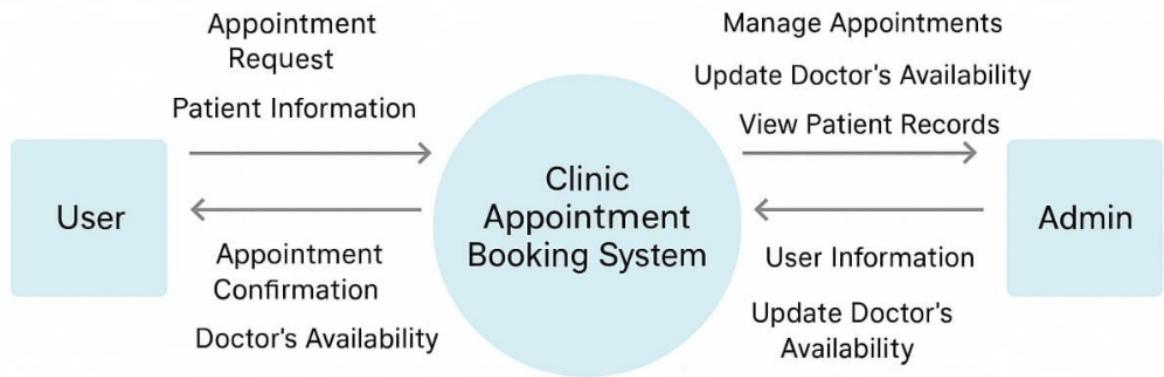
- **Appointments**: Stores details such as booking ID, doctor ID, patient details, date, and time.
- **Doctors**: Contains doctor information, including specialization, qualifications, and department affiliation.
- **Availability**: Defines doctors' available time slots, including session details and working hours.
- **Department**: Manages different hospital departments with department IDs and names.

3.3.2 Relationships

- **Appointments** are linked to Doctors through **doc_id**.
- **Doctors** are associated with Departments, ensuring proper classification.
- **Availability** is mapped to Doctors, allowing real-time schedule updates.

The schema design optimizes data storage, maintains relationships efficiently, and supports seamless appointment scheduling.

3.4 Dataflow Diagram (Level 0)



3.5 Dataflow Diagram (Level 1)

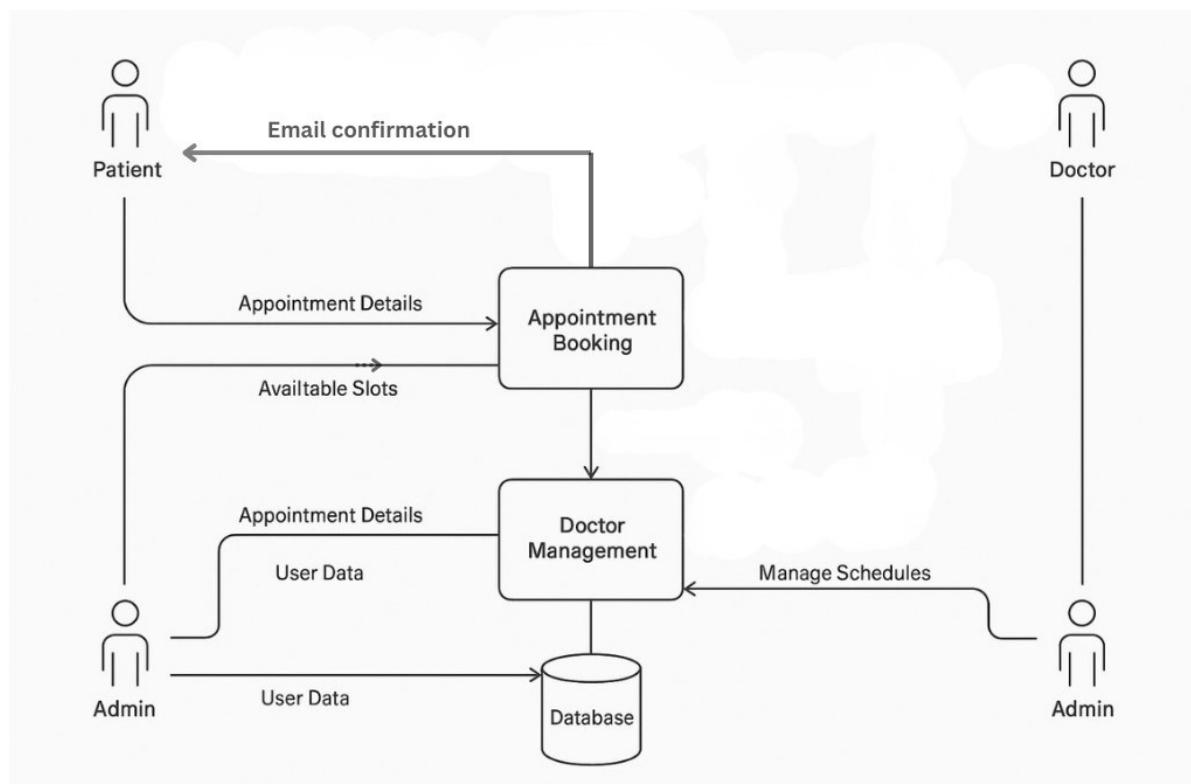


Figure 3.3: Data Flow Diagram Level 1

The Data Flow Diagram (DFD) represents how information moves within the system, ensuring smooth processing and efficient data management.

3.5.1 Actors & Entities

- **Patient:** Initiates the appointment booking process by providing details.
- **Doctor:** Reviews and manages appointment schedules.
- **Admin:** Maintains and updates user records, ensuring data consistency.

3.5.2 Flow of Data

- **Appointment Booking:** Patients enter their details and select a doctor.
- **Doctor Management:** The system cross-checks doctor availability and assigns a time slot.
- **Database Interaction:** Patient and appointment data are stored securely.
- **Email Confirmation:** The system sends a notification to confirm the appointment.

The DFD ensures that each process within the system is structured, enhancing efficiency and minimizing scheduling conflicts.

3.6 Use Case Diagram

The **Use Case Diagram** illustrates the key interactions between users and the system, highlighting different functionalities.

- **Actors:**
 - **User (Patient):** Can view available doctors and departments, book an appointment, receive confirmation, view their schedule, and interact with the chatbot for assistance.
 - **Admin (Clinic Management):** Manages doctor availability, accesses appointment details, and handles chatbot interactions.
- **Key Features:**
 - **Appointment Booking:** Patients can select doctors, choose a convenient time slot, and book appointments.
 - **Doctor Availability Management:** Admin ensures efficient scheduling by managing doctors' working hours and availability.

- **Chatbot Integration:** Patients can interact with the chatbot for quick assistance in scheduling and inquiries.

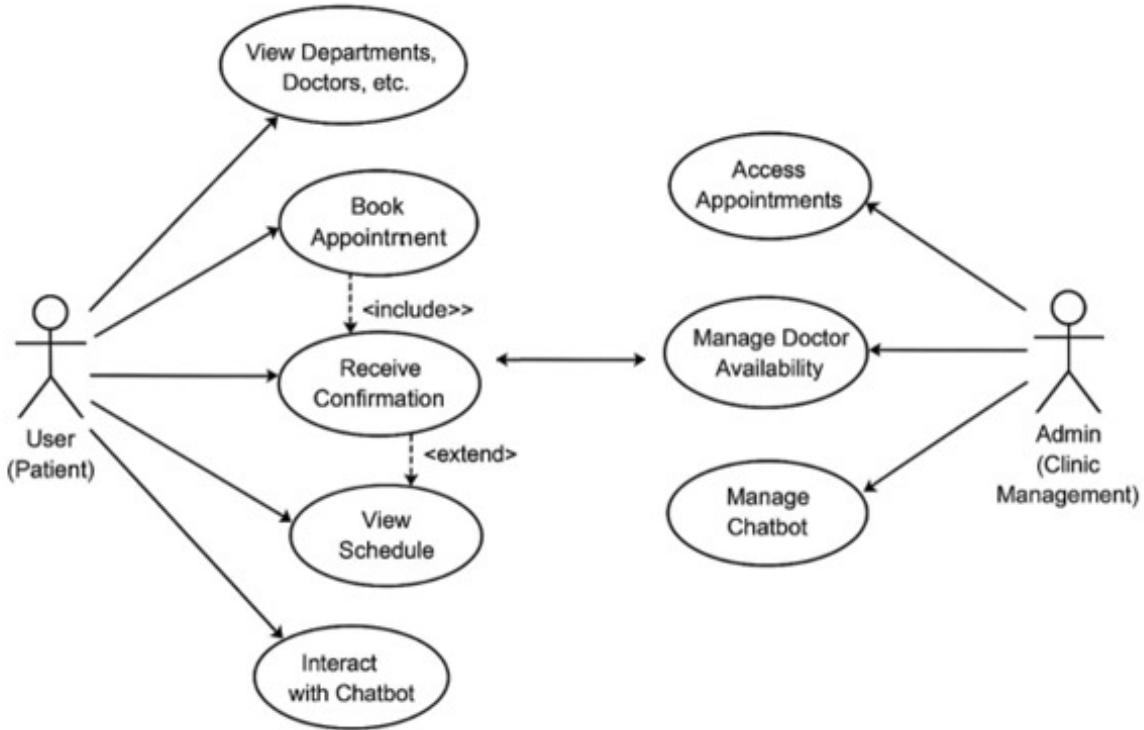


Figure 3.4: Use case Diagram

This diagram ensures a clear understanding of system interactions and streamlines the appointment scheduling process.

3.7 Technology Stack

3.7.1 Front End

The frontend of this project is built using modern web technologies to ensure a seamless user experience:

- 1. React.js** – A JavaScript library for building dynamic and interactive user interfaces. React ensures fast rendering and efficient state management, enhancing the overall responsiveness of the application.
- 2. HTML5** – Provides the structural foundation of the web pages, ensuring semantic and well-organized content.

3. CSS3 – Used for styling and layout, making the website visually appealing and responsive across different devices. **4. React Router** – Enables smooth navigation between different sections of the website, including departments, doctors, and contact pages.

5. Axios – A promise-based HTTP client used for making API requests to fetch and send data between the frontend and backend.

3.7.2 Back End

The backend of this project is designed to handle data storage, user authentication, and appointment management efficiently:

1. Node.js – A runtime environment that allows JavaScript to be executed on the server side, making the backend lightweight and scalable.

2. Express.js – A fast and minimalistic web framework for Node.js, used to create API endpoints and handle requests efficiently.

3. MongoDB – A NoSQL database that stores patient records, appointment details, and doctor schedules in a flexible and scalable manner.

4. Mongoose – An ODM (Object-Document Mapping) library for MongoDB, simplifying database operations and schema management.

5. Nodemailer – Used for sending automated confirmation emails to patients after successful appointment bookings.

6. JWT (JSON Web Token) – Ensures secure authentication and authorization for admin access and protected routes.

3.8 IMPLEMENTATION

Phase 1: Project Setup & Environment Configuration

- Set up a Git repository for version control.
- Install required dependencies for frontend and backend.
- Set up MongoDB.

Phase 2: Database Design & Backend APIs

Define MongoDB Schema & Models

- **Doctor:** Contains doctor details and available slots.
- **Appointment:** Tracks bookings and their status.
- **Availability:** Tracks available slots.
- **Departments:** Manages department details.

Develop API Endpoints in Express.js

- **Authentication APIs:** Login and Register with JWT authentication.
- **Doctor Management APIs:** Add, List, and Manage doctors.
- **Appointment APIs:** Book an appointment and check availability.

Phase 3: Frontend Development in React

Develop UI Components

- **Department List**
- **Doctor List & Profile Page:** Fetch and display doctors with availability.
- **Appointment Booking Page:** Form to select date and time.
- **Admin Dashboard:** Manage doctors and schedules.

Integrate APIs with React using Axios

- Fetch doctors and available slots.
- Submit appointment booking requests.

Phase 4: Testing & Deployment

Testing

- Unit testing for APIs using Jest/Supertest.
- UI testing using React Testing Library.

Deployment

- Deploy **Frontend** on Vercel/Netlify.

- Deploy **Backend** on Render/AWS/Heroku.
- Deploy **MongoDB** on MongoDB Atlas.

Phase 5: Enhancements & Optimization

- Real-time appointment status updates .
- Payment gateway integration for online consultations.
- Video Consultation.

3.9 RESULTS AND DISCUSSION

3.9.1 Home Page

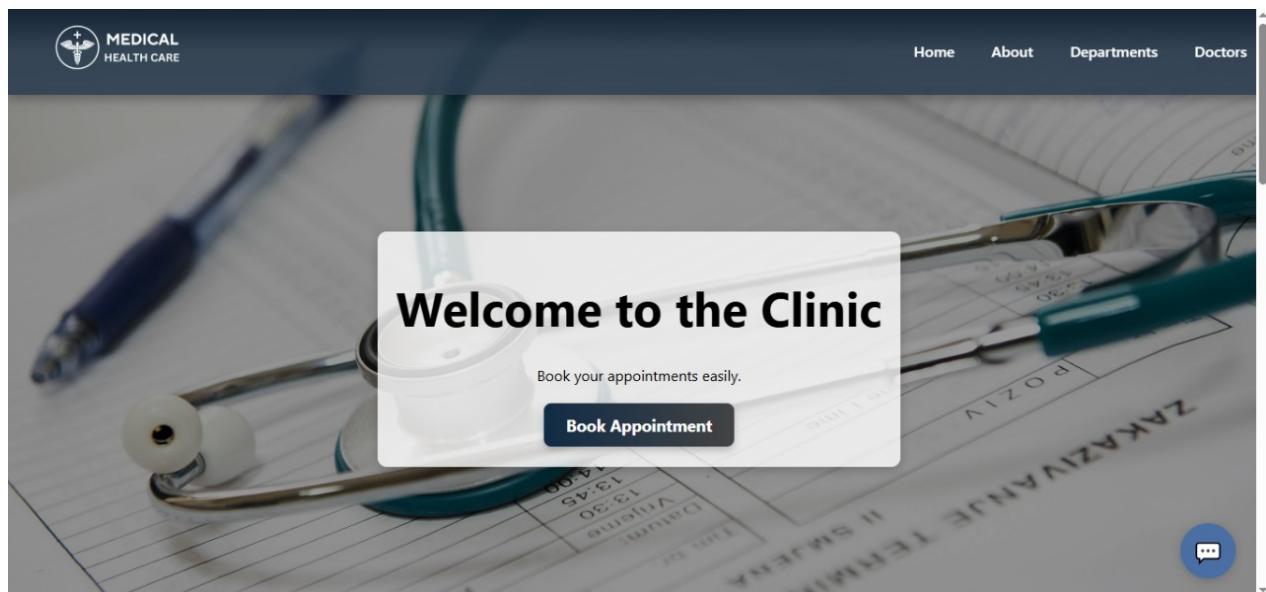


Figure 3.5: Home Page

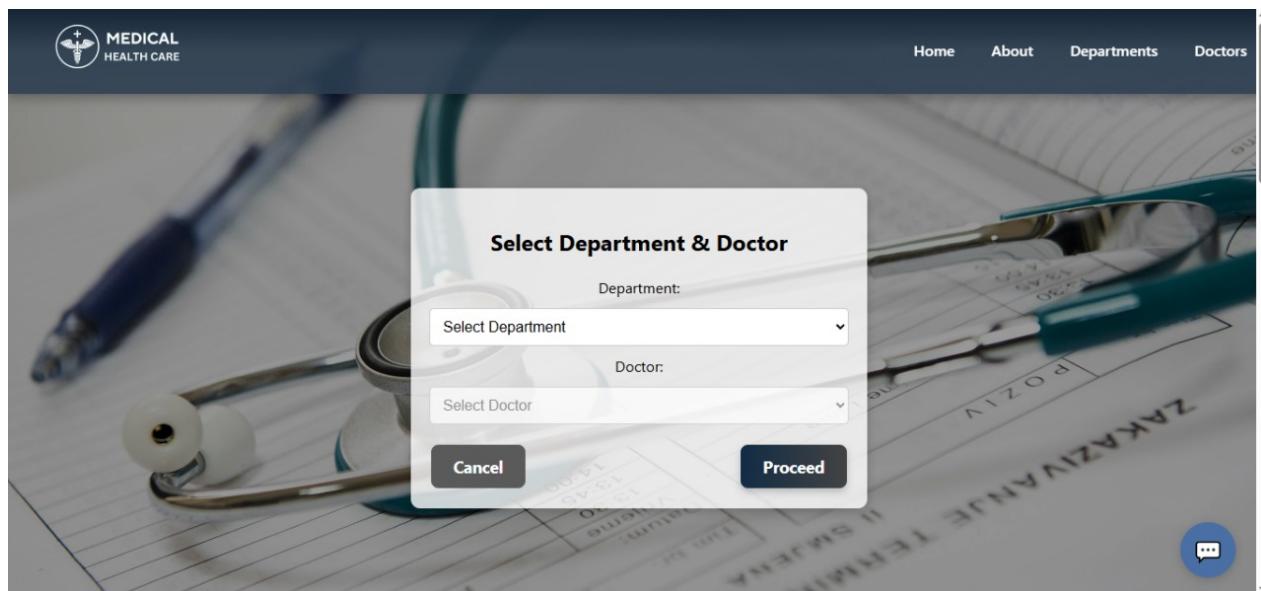


Figure 3.6: Selecting department - doctor Page

3.9.2 Chatbot

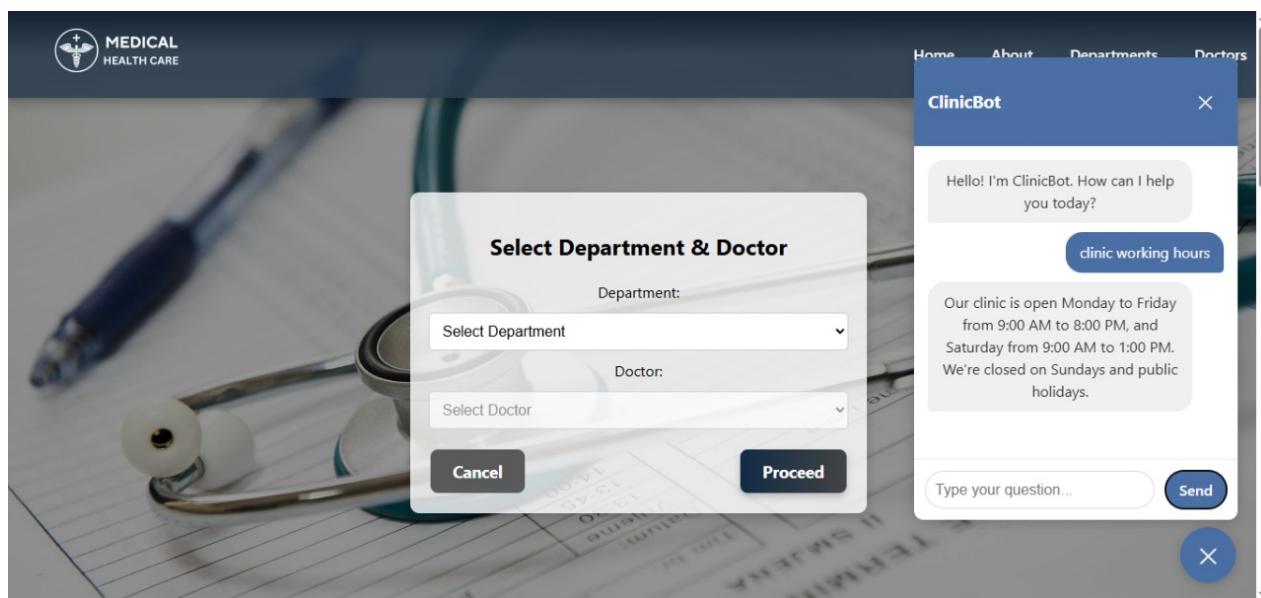


Figure 3.7: Selecting department - doctor Page

3.9.3 About Page

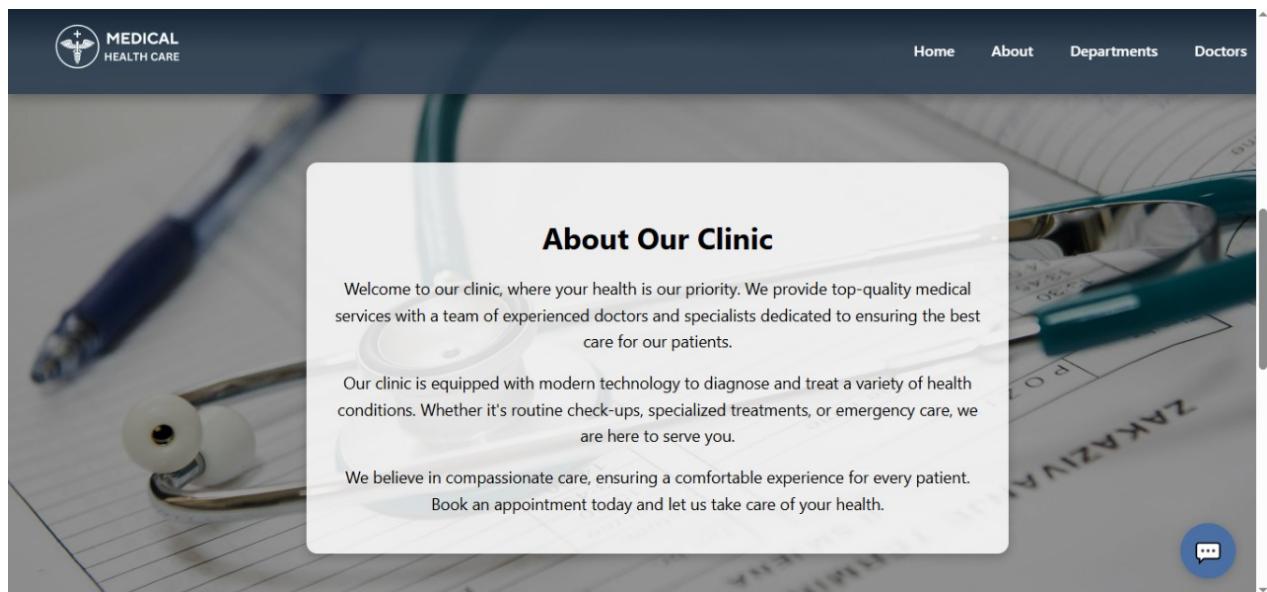


Figure 3.8: About Page

3.9.4 Departments Page

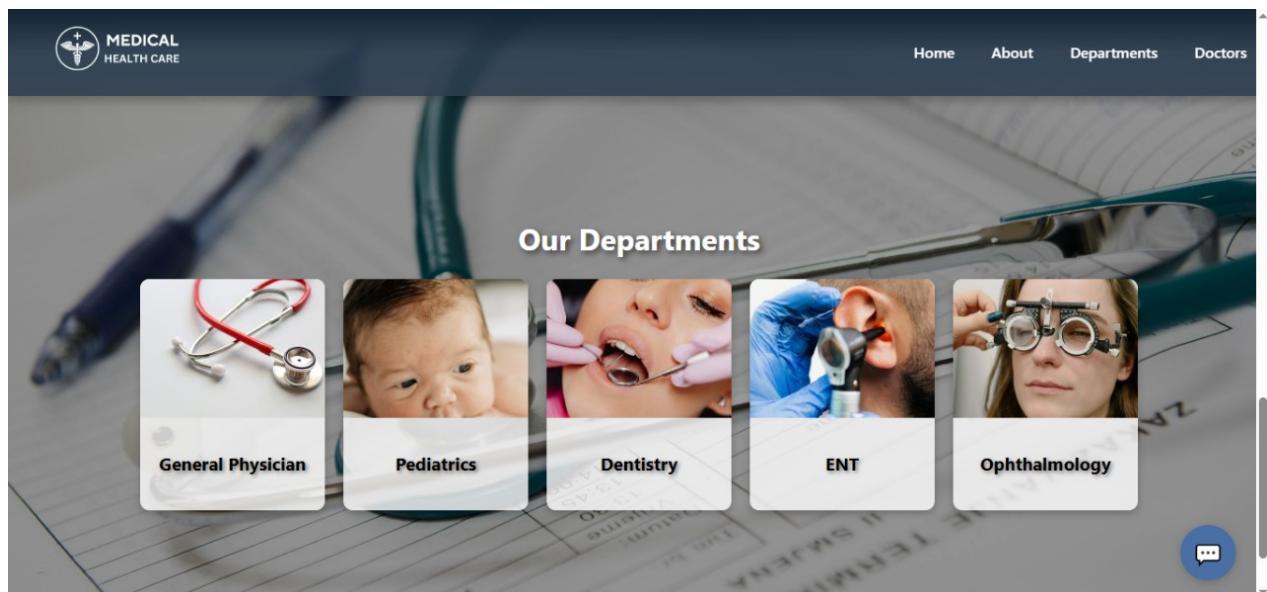


Figure 3.9: Departments Page

3.9.5 Department wise doctor filtering

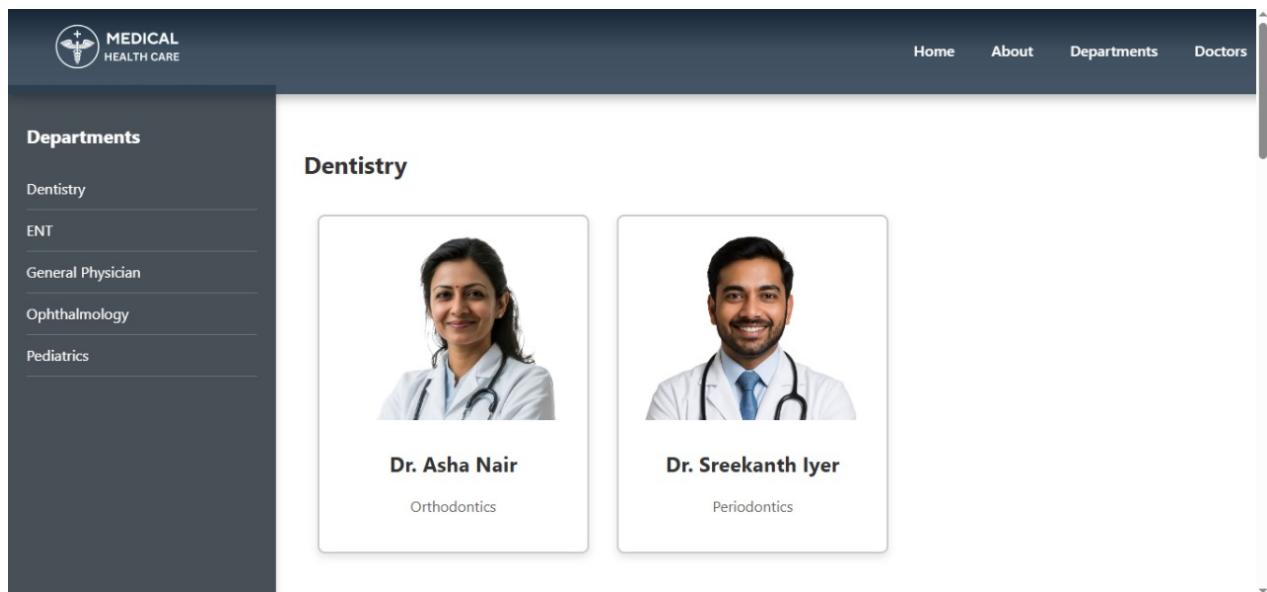


Figure 3.10: Dept based doctor filtering

3.9.6 Doctor Profile

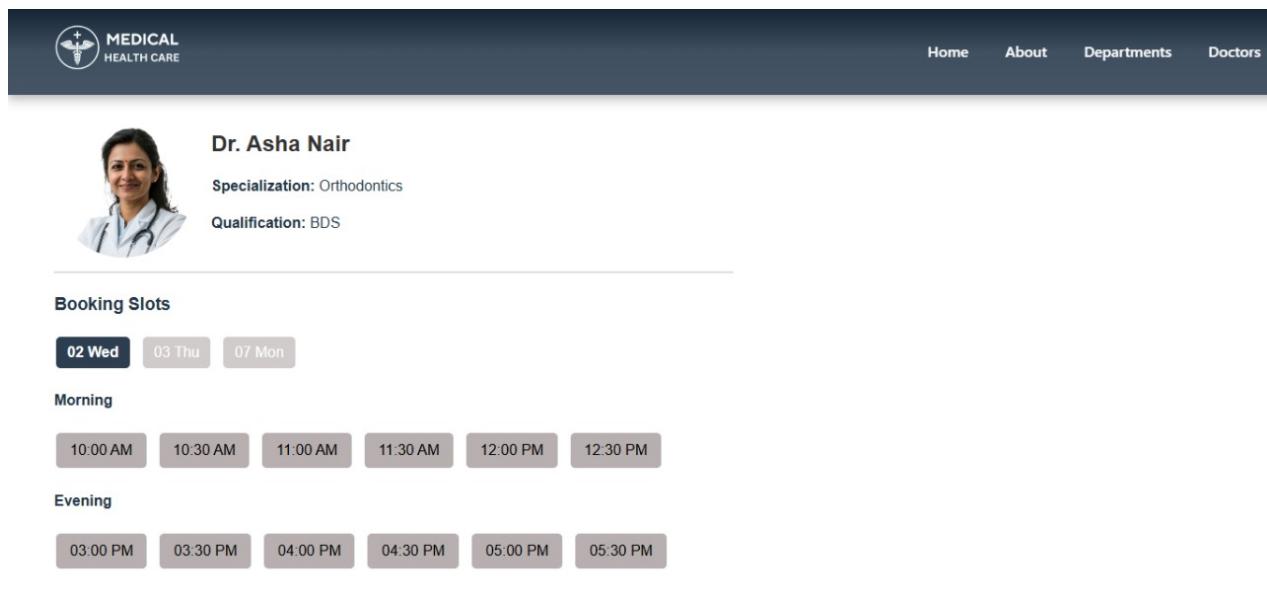


Figure 3.11: Doctor Profile Page

3.9.7 Book Appointment

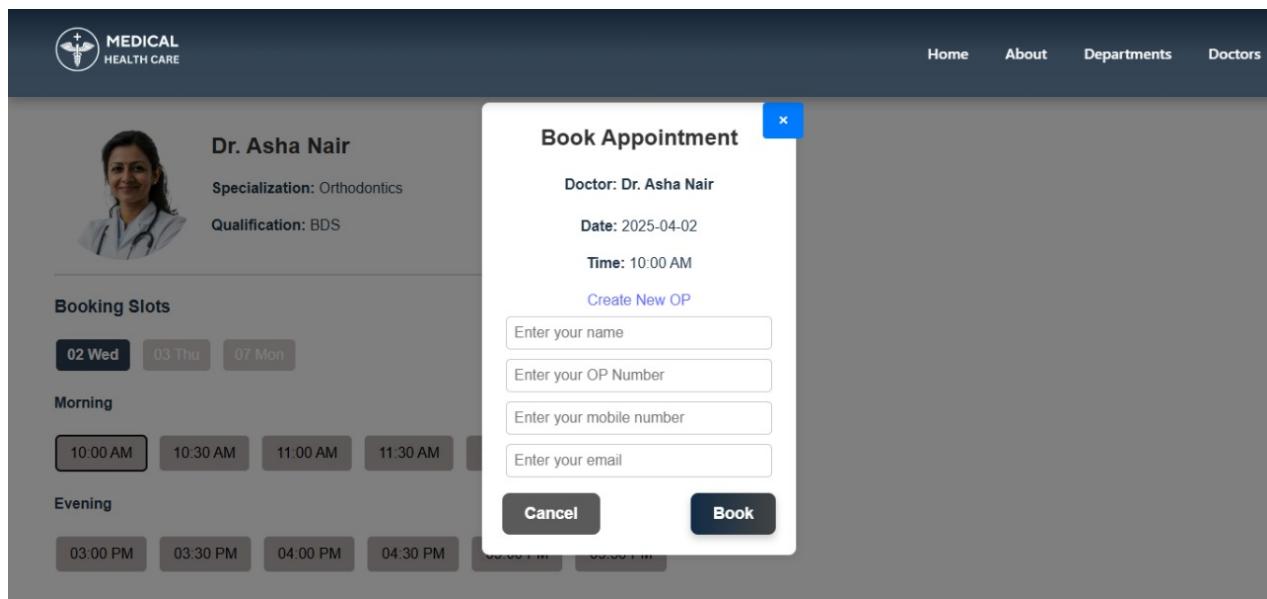


Figure 3.12: Book Appointment Page

3.9.8 OP Generation

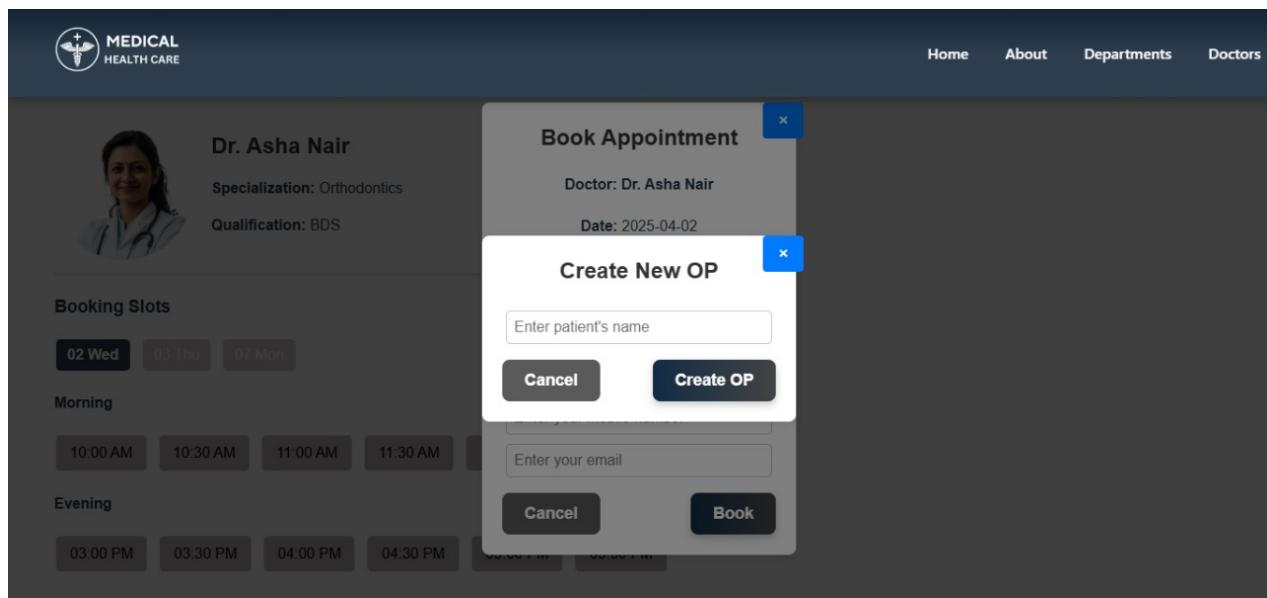


Figure 3.13: OP Generation Page

3.9.9 Admin Login Page

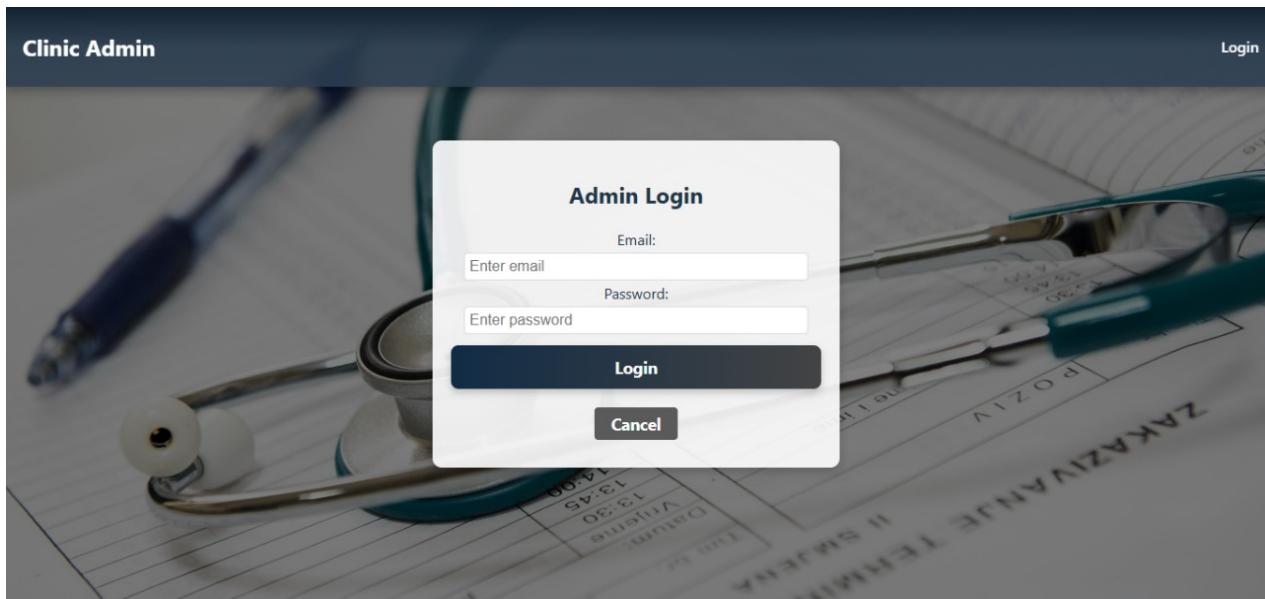


Figure 3.14: Admin Login Page

3.9.10 Admin Dashboard

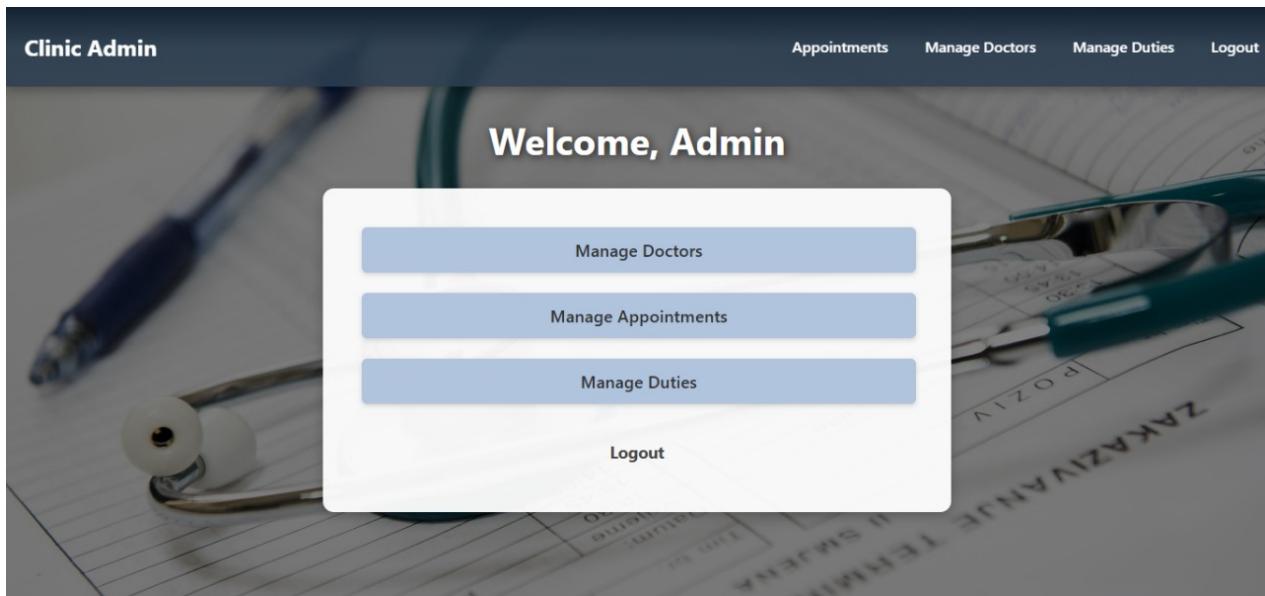


Figure 3.15: Admin Dashboard Page

3.9.11 Admin-Appointments

The screenshot shows the 'Admin - Appointments' section of the Clinic Admin interface. At the top, there are filter inputs for Date (dd-mm-yyyy), Doctor (Filter by doctor), Patient (Filter by patient), and a 'Clear Filters' button. Below the filters is a table with columns: Doctor, Date, Time, Patient, OP Number, Mobile, and Email. The table contains five rows of appointment data.

Doctor	Date	Time	Patient	OP Number	Mobile	Email
Dr. Asha Nair	27/01/2025	09:00 AM	abc	123	9876543210	abc@gmail.com
Dr. Asha Nair	27/01/2025	09:00 AM	pqr	101	9078563412	pqr@gmail.com
Dr. Sneha Mathew	20/03/2025	09:00 AM	ABC	1669	9777667888	abc@gmail.com
Dr. Sneha Mathew	20/03/2025	09:30 AM	xyz	1502	2191479882	gfdw@gji.com
Dr. Asha Nair	02/04/2025	10:00 AM	Ramesh	321	9112412213	ramesh@gmail.com

Figure 3.16: Admin-Appointment

3.9.12 Manage Doctor Profile Page

The screenshot shows the 'Manage - Doctor Profile' section of the Clinic Admin interface. At the top, there is a '+ Add Doctor' button. Below it is a table with columns: Image, Doctor Name, Specialization, Qualification, and Actions. The table lists four doctors: Dr. Ameer Ali, Dr. Asha Nair, Dr. Hari Sankar, and Dr. Kavya Warrier, each with a small profile picture and edit/cancel icons in the Actions column.

Image	Doctor Name	Specialization	Qualification	Actions
	Dr. Ameer Ali	Vision Therapy	MBBS, MS	
	Dr. Asha Nair	Orthodontics	BDS	
	Dr. Hari Sankar	Sinus Surgery	MBBS, MS	
	Dr. Kavya Warrier	Audiology	MBBS, MS	

Figure 3.17: Manage doctor profile Page

3.9.13 Add Doctor via Admin Panel

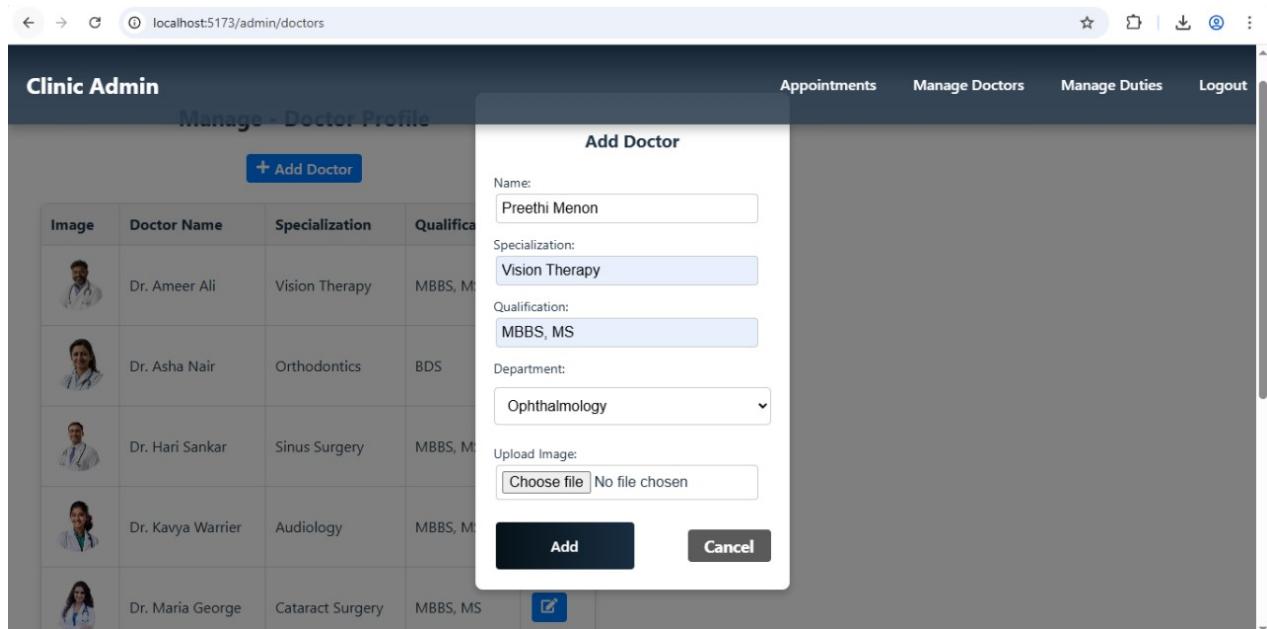


Figure 3.18: Admin Dashboard: Add Doctor

3.9.14 Doctor Duty Management Page

Doctor Name		Department	Action	
Dr. Asha Nair		Orthodontics	<button>Hide Schedule</button>	
<button>+ Add New Schedule</button>				
Session	Start Time	End Time	Date	Actions
Morning	10:00	13:00:00	02/04/2025	<button>Edit</button> <button>Delete</button>
Evening	15:00:00	18:00:00	02/04/2025	<button>Edit</button> <button>Delete</button>
Morning	09:00	12:00	03/04/2025	<button>Edit</button> <button>Delete</button>
Morning	09:00	12:00	07/04/2025	<button>Edit</button> <button>Delete</button>
Evening	16:00:00	19:00:00	07/04/2025	<button>Edit</button> <button>Delete</button>

Figure 3.19: Admin Dashboard: Doctor Duty management

3.9.15 Add Schedule for Doctor Duty Management

The screenshot shows the 'Doctor Duty Management' section of the 'Clinic Admin' application. At the top, there are navigation links: Appointments, Manage Doctors, Manage Duties, and Logout. Below the header, a table lists a single doctor entry: Dr. Asha Nair (Orthodontics). A 'Hide Schedule' button is visible next to her name. A blue button '+ Add New Schedule' is located below the table. The main area is titled 'Add New Schedule' and contains fields for Date (05-04-2025), Start Time (09:00 AM), End Time (12:00 PM), and Session (Morning). At the bottom are 'Save' and 'Cancel' buttons.

Figure 3.20: Doctor Duty management : Add Schedule

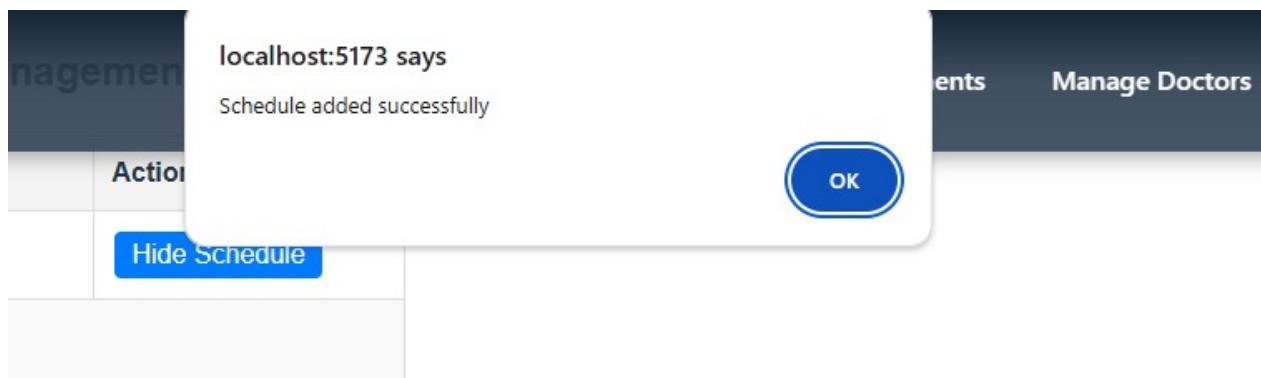


Figure 3.21: Schedule Added

3.9.16 Delete Schedule

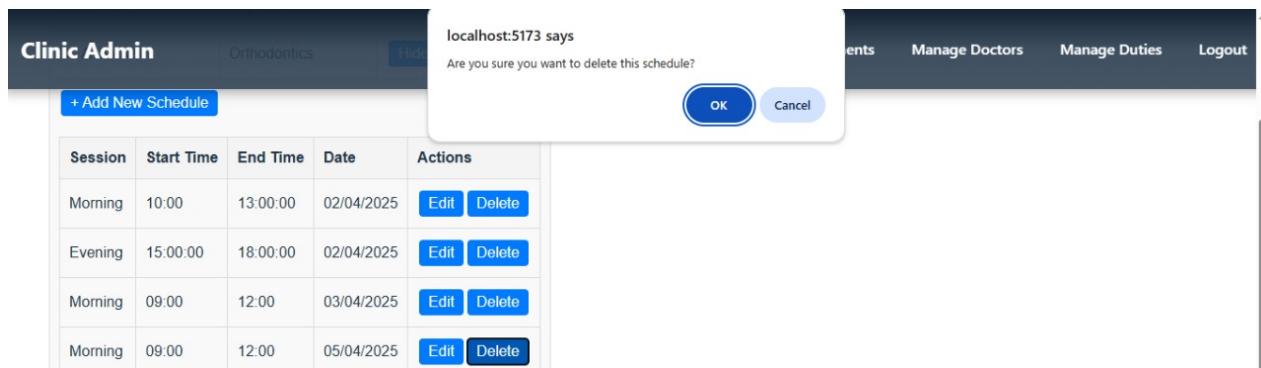


Figure 3.22: Doctor Duty management : Delete Schedule

3.9.17 Appointment Confirmation Mail

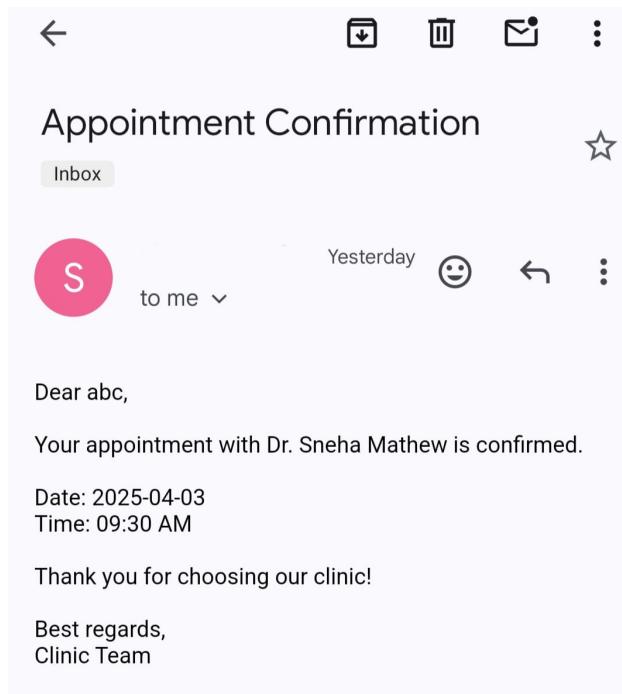


Figure 3.23: Confirmation mail

3.9.18 Database: Doctor's Table

	<code>* doctors</code>	<code>_id ObjectId</code>	<code>doc_id Int32</code>	<code>doc_name String</code>	<code>specialization String</code>	<code>qualification</code>	<code>dept_id Int32</code>	<code>image String</code>
1	<code>ObjectId('679b135293e21d..</code>	<code>1</code>		"Dr. Asha Nair"	"Orthodontics"	"BDS"	<code>1</code>	"asha_nair.png"
2	<code>ObjectId('679b135293e21d..</code>	<code>2</code>		"Dr. Sneha Mathew"	"Pediatric Care"	"MBBS, MD"	<code>2</code>	"sneha_mathew.png"
3	<code>ObjectId('679b135293e21d..</code>	<code>3</code>		"Dr. Kavya Warrier"	"Audiology"	"MBBS, MS"	<code>3</code>	"kavya_warrier.png"
4	<code>ObjectId('679b135293e21d..</code>	<code>4</code>		"Dr. Rajesh Pillai"	"Internal Medicine"	"MBBS, MD"	<code>4</code>	"rajesh_pillai.png"
5	<code>ObjectId('679b135293e21d..</code>	<code>5</code>		"Dr. Maria George"	"Cataract Surgery"	"MBBS, MS"	<code>5</code>	"maria_george.png"
6	<code>ObjectId('679b135293e21d..</code>	<code>6</code>		"Dr. Sreekanth Iyer"	"Periodontics"	"BDS, MDS"	<code>1</code>	"sreekanth_iyer.png"
7	<code>ObjectId('679b135293e21d..</code>	<code>7</code>		"Dr. Susan Thomas"	"Neonatal Care"	"MBBS, DCH"	<code>2</code>	"susan_thomas.png"
8	<code>ObjectId('679b135293e21d..</code>	<code>8</code>		"Dr. Hari Sankar"	"Sinus Surgery"	"MBBS, MS"	<code>3</code>	"hari_sankar.png"
9	<code>ObjectId('679b135293e21d..</code>	<code>9</code>		"Dr. Meera Das"	"Family Medicine"	"MBBS"	<code>4</code>	"meera_das.png"
10	<code>ObjectId('679b135293e21d..</code>	<code>10</code>		"Dr. Ameer Ali"	"Vision Therapy"	"MBBS, MS"	<code>5</code>	"ameer_ali.png"

Figure 3.24: Doctor's Table

3.9.19 Database: Appointment Table

	<code>* appointments</code>	<code>_id ObjectId</code>	<code>doc_id Int32</code>	<code>doc_name String</code>	<code>date String</code>	<code>time String</code>	<code>patient_name String</code>	<code>op_number String</code>	<code>mobile String</code>	<code>email String</code>
1	<code>ObjectId('67c86783305ed1..</code>	<code>1</code>		"Dr. Asha Nair"	"2025-01-27"	"09:00 AM"	"abc"	"123"	"9876543210"	"abc@gmail.com"
2	<code>ObjectId('67c87ced3a5541..</code>	<code>1</code>		"Dr. Asha Nair"	"2025-01-27"	"09:00 AM"	"pqr"	"101"	"9878563412"	"pqr@gmail.com"
3	<code>ObjectId('67dbadec627ac25..</code>	<code>2</code>		"Dr. Sneha Mathew"	"2025-03-28"	"09:00 AM"	"ABC"	"1609"	"9777667888"	"abc@gmail.com"
4	<code>ObjectId('67dbaca2f27ac25..</code>	<code>2</code>		"Dr. Sneha Mathew"	"2025-03-28"	"10:00 AM"	"varsha k"	"123"	"9827163463"	"kumarvarsha2907@gmail.c.."
5	<code>ObjectId('67dbad3e27ac25..</code>	<code>2</code>		"Dr. Sneha Mathew"	"2025-03-28"	"09:00 AM"	"devu"	"12323"	"2124413413"	"devusuntha28@gmail.com"
6	<code>ObjectId('67dbb822c04588..</code>	<code>2</code>		"Dr. Sneha Mathew"	"2025-03-28"	"09:30 AM"	"xyz"	"1502"	"2191479882"	"gfdw@gji.com"

Figure 3.25: Appointment Table

3.9.20 Database: Department Table

departments			
	_id ObjectId	dept_id Int32	dept_name String
1	ObjectId('679b0f8493e21d...')	1	"Dentistry"
2	ObjectId('679b0f8493e21d...')	2	"Pediatrics"
3	ObjectId('679b0f8493e21d...')	3	"ENT"
4	ObjectId('679b0f8493e21d...')	4	"General Physician"
5	ObjectId('679b0f8493e21d...')	5	"Ophthalmology"

Figure 3.26: Department Table

3.9.21 Database: Availability Table

availability							
	_id ObjectId	doc_id Int32	day_of_week String	date String	session String	start_time String	end_time String
1	ObjectId('67b19c46a586a2...')	1	"Mon"	"2025-04-07"	"Morning"	"09:00"	"12:00"
2	ObjectId('67b19c46a586a2...')	1	"Mon"	"2025-04-07"	"Evening"	"16:00:00"	"19:00:00"
3	ObjectId('67b19c46a586a2...')	1	"Wed"	"2025-04-02"	"Morning"	"10:00"	"13:00:00"
4	ObjectId('67b19c46a586a2...')	1	"Wed"	"2025-04-02"	"Evening"	"15:00:00"	"18:00:00"
5	ObjectId('67b19c46a586a2...')	2	"Tue"	"2025-04-08"	"Morning"	"08:30:00"	"11:30:00"
6	ObjectId('67b19c46a586a2...')	2	"Tue"	"2025-04-08"	"Evening"	"17:00:00"	"20:00:00"
7	ObjectId('67b19c46a586a2...')	2	"Thu"	"2025-04-03"	"Morning"	"09:00:00"	"12:00:00"
8	ObjectId('67b19c46a586a2...')	2	"Thu"	"2025-04-03"	"Evening"	"18:00:00"	"21:00:00"
9	ObjectId('67e9902f6cab07...')	9	"Mon"	"2025-04-07"	"Morning"	"09:00:00"	"12:00"

Figure 3.27: Availability Table

3.10 Advantages

- **Convenience for Patients**

The system allows patients to book appointments anytime, anywhere, eliminating the need for physical visits to the clinic just for scheduling. Users can easily select their preferred doctor and time slot. Additionally, new patients can generate an OP number automatically, making the registration process seamless and hassle-free.

- **Efficient Appointment Management**

The booking system ensures that each time slot can accommodate a maximum of two patients, preventing overcrowding. Patients can instantly check slot availability, reducing wait times and confusion. Once the booking is confirmed, users receive a confirmation email, making the process transparent and reliable.

- **User-Friendly Navigation**

The website features a structured navigation bar that includes sections like Home, Departments, Doctors, About, etc. When clicked, each section smoothly scrolls to the respective page, making navigation easy and intuitive. This enhances the overall user experience and ensures that information is readily accessible.

- **Admin Panel for Better Management**

The clinic management team has access to a **dedicated admin panel** where they can monitor and manage appointment details. They can update doctors' schedules, mark duty leaves, and maintain accurate records of patient bookings. This reduces manual workload and enhances operational efficiency.

- **Automated Email Notifications**

Upon successful booking, an email confirmation is sent to the patient, ensuring they have a record of their appointment details. This reduces the chances of missed appointments and enhances communication between the clinic and patients.

- **Integrated Chatbot for Basic Support**

The chatbot assists users by addressing queries related to clinic details, appointment scheduling, doctor availability, and more. This reduces the dependency on staff for answering common questions and provides basic essential support to users.

- **Reduces Administrative Workload**

The system automates scheduling and data management, reducing the burden on clinic staff. It eliminates the need for manual appointment handling, freeing up resources for patient care. Doctors and administrators can focus more on medical services instead of scheduling issues.

- **Saves Time for Patients and Doctors**

The system minimizes waiting periods by ensuring structured appointment slots. Patients don't need to stand in long queues, improving their overall experience. Doctors get a well-organized schedule, optimizing their consultation hours.

3.11 Limitations

1. **No Emergency Booking Feature**

The system is designed for scheduled appointments and does not support emergency cases or urgent medical bookings, which may require an alternative manual intervention.

2. Limited AI Capabilities in Chatbot While the chatbot can handle basic queries regarding appointment scheduling and clinic details, it may not provide accurate responses for complex medical inquiries, requiring human intervention.

3. Scalability Concerns If the clinic expands significantly or the number of concurrent users increases beyond expected limits, the system may require performance optimizations or infrastructure upgrades.

4. Dependency on Internet Connectivity The system requires a stable internet connection for both patients and clinic staff. In areas with poor connectivity, users may face difficulties in booking or managing appointments.

CHAPTER 4

CONCLUSION AND FUTURE SCOPE

The Clinic Appointment Booking System enhances efficiency, reduces administrative workload, and improves patient satisfaction by offering a seamless and automated booking process. By leveraging AI-powered chatbots, real-time scheduling, and notifications, the system optimizes healthcare service delivery. With its user-friendly interface and secure framework, it provides a reliable solution for both patients and healthcare providers, ensuring a smooth and organized appointment management process.

4.1 Future Scope

The proposed system has several potential enhancements that can further improve efficiency, accessibility, and functionality in the future.

4.1.1 Integration with Telemedicine Services

This will enable virtual consultations, allowing patients to connect with doctors remotely, reducing the need for physical visits.

4.1.2 AI-Driven Doctor Recommendations

Artificial Intelligence can be leveraged to recommend doctors based on patient history and preferences, enhancing personalized healthcare.

4.1.3 Multi-Clinic and Multi-Specialty Support

The system can be expanded to support multiple clinics and specialties, making it accessible to a broader user base.

4.1.4 Secure Payment Gateway Integration

Incorporating a secure online payment system will allow patients to pay for consultations seamlessly, ensuring hassle-free transactions.

4.1.5 Advanced Chatbot with Voice Support

Enhancing the chatbot with voice capabilities and multilingual assistance will improve user experience and accessibility.

4.1.6 Integration with Electronic Health Records (EHR)

By linking with EHR systems, patient records can be efficiently managed, enabling comprehensive healthcare data storage and retrieval.

REFERENCES

- [1] Ms. Sanjeevani P. Avhale, Ms. Wrushali R. Ajabe, Ms Pallavi A. "Doctor Appointment Online Booking System "-2018
<https://ijcrt.org/papers/IJCRT1812133.pdf>
<https://ijcrt.org/papers/IJCRT1812133.pdf>
- [2] Ms .Geeta, Ms.Shivagona Patil
"Patient Appointment and Scheduling System" - 2019
<https://www.scirp.org/pdf/eng2013122011373865.pdf>
- [3] Mr.. Xiuju Zhan, Mr. Xiufeng Liu
"Design and Implementation of Clinic Appointment Registration System m" - 2013
https://www.researchgate.net/publication/332864696_Design_and_Implementation_of_a_Patient_Appointment_and_Scheduling_System
- [4] Mr. Kamal Acharya
"CLINIC MANAGEMENT SYSTEM PROJECT REPORT-2021"
https://www.researchgate.net/publication/382947045_CLINIC_MANAGEMENT_SYSTEM_PROJECT_REPORT
- [5] Mr. Miguel Boety Bopabote
"Clinic Management System Project Documentation -2014"
https://www.researchgate.net/publication/382947045_CLINIC_MANAGEMENT_SYSTEM_PROJECT_REPORT