

PROJECT REPORT

On

Doctor Appointment App (DocEase)

Submitted in partial fulfillment of the requirement for the Course BEE
(22CS026) of

COMPUTER SCIENCE AND ENGINEERING B.E.

Batch-2022 in Nov -2024



Under the Guidance of :

Ms. Preenu Mittan

Submitted By:

Bhuvesh Mittal

2210991450

Daanushi Sharma

2210991469

**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING CHITKARA UNIVERSITY PUNJAB**

CERTIFICATE

This is to be certified that the project entitled “DocEase”, an online doctor appointment app has been submitted for the Bachelor of Computer Science Engineering at Chitkara University, Punjab during the academic semester July 2024- December 2024 is a bona fide piece of project work carried out by “Bhuvesh Mittal (2210991450) and Daanushi Sharma (2210991469)” towards the partial fulfillment for the award of the course Integrated Project (CS 203) under the guidance of “Ms. Preenu Mittan” and supervision.

Sign. of Project Guide :

Ms. Preenu Mittan

CANDIDATE'S DECLARATION

We , Bhuvesh Mittal (2210991450) and Daanushi Sharma (2210991469) G-18, B.E.-2022 of the Chitkara University, Punjab hereby declare that the Integrated Project Report entitled “DocEase” is an original work and data provided in the study is authentic to the best of our knowledge. This report has not been submitted to any other Institute for the award of any other course.

Bhuvesh Mittal
2210991450

Daanushi Sharma
2210991469

Place: Chitkara University

Date: November 26, 2024

ACKNOWLEDGMENT

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior and acts during the course of study.

We express our sincere gratitude to all for providing me an opportunity to undergo Integrated Project as a part of the curriculum.

We are thankful to "Ms. Preenu Mittan" for her support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

We also extend our sincere appreciation to “Ms. Preenu Mittan” who provided his valuable suggestions and precious time in accomplishing our integrated project report.

Lastly, we would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to-day experience and received lots of suggestions that improve our quality of work.

Bhuvesh Mittal

2210991450

Daanushi Sharma

2210991469

Table of Contents

1. Abstract
2. Introduction to the project
 - 2.1 Background
 - 2.2 Problem Statement
3. Software and Hardware Requirement Specification
 - 3.1 Methods
 - 3.2 Programming/Working Environment
 - 3.3 Requirements to run the application
4. Data Flow Diagram(DFD)
5. Database
 - 5.1 Database Analysis
 - 5.2 Database Implementation
 - 5.3 Database Design
6. Program's Structure Analyzing and GUI Constructing (Project Snapshots)
 - 6.1 Program Structure Analysis
 - 6.2 GUI Construction (Project Snapshots)
 - 6.3 GUI Design Considerations
7. Code-Implementation and Database Connections (If any)
8. Future Scope
9. Conclusion
10. Bibliography/References

1. Abstract

This project is an online doctor appointment system built using modern web technologies, including Node.js, Express.js, React, Tailwind CSS, and MongoDB. The primary objective is to provide a seamless, hassle-free platform for booking and managing doctor appointments for patients, doctors, and administrators.

The system includes three fully implemented user roles:

- **Admin Dashboard:** Administrators can efficiently manage the platform, add new doctors, store their profiles in MongoDB, and display key details such as doctor names, specializations, and availability. Admins can also oversee appointment bookings, payment statuses, and platform operations.
- **Doctor Dashboard:** Doctors can manage their schedules, view upcoming appointments, mark appointments as completed, update their availability and profiles, and track earnings and payment statuses.
- **User Dashboard:** Users can create accounts, log in, explore a list of doctors, view profiles, filter doctors by specialty, book appointments for specific dates and times, cancel appointments if needed, and make secure payments via Razorpay integration.

The project is fully dynamic and designed to simplify healthcare appointment management. It offers a clean, scalable, and user-friendly interface to meet modern healthcare needs. This system represents a complete, practical solution for connecting patients and doctors, demonstrating the potential to make a meaningful real-world impact on healthcare accessibility and management.

2. Introduction

2.1 Background

The rapid advancement of technology has revolutionized various industries, with healthcare being one of the most profoundly impacted sectors. The integration of digital tools into healthcare management has paved the way for online platforms that simplify administrative tasks and enhance communication between patients, doctors, and administrators. These platforms play a crucial role in improving appointment booking processes, managing doctor schedules, and offering a seamless experience for users.

Online doctor appointment systems are comprehensive solutions designed to handle various tasks, such as storing detailed doctor profiles, managing availability schedules, facilitating user-friendly appointment booking, and enabling secure payment processes. By centralizing these functionalities, such systems eliminate manual work, improve accessibility, and enhance the overall user experience through real-time updates and streamlined workflows.

In today's world, where there is a growing emphasis on accessible healthcare and the rapid adoption of online services, these systems have become indispensable. They contribute significantly to improving patient care, optimizing resource management, and increasing operational efficiency.

Our Online Doctor Appointment System leverages advanced web technologies to meet these demands, offering a complete and dynamic platform that caters to the needs of patients, doctors, and administrators. The system demonstrates how technology can transform healthcare delivery, making it more efficient, accessible, and user-friendly.

2.2 Problem Statement

Traditional methods of managing doctor appointments often rely on manual processes, phone calls, and paper-based records, resulting in inefficiencies, delays, and errors. Patients frequently face challenges in finding available doctors and booking appointments conveniently. Administrators encounter difficulties in managing schedules, maintaining accurate records, and ensuring smooth communication. Similarly, doctors often lack a centralized system to efficiently track their appointments and update their availability.

This project addresses the critical gap created by the absence of a unified platform that integrates the requirements of patients, doctors, and administrators. Without such a system, data becomes fragmented, communication remains inefficient, and the overall process of appointment management becomes unnecessarily cumbersome and time-consuming.

The specific challenges this project aims to solve include:

- **How can a centralized platform streamline the appointment booking process for patients while ensuring efficiency and accessibility?**
- **What features are essential to cater to the unique needs of administrators, doctors, and patients effectively?**

These challenges have guided the design and development of our Online Doctor Appointment System. By leveraging modern web technologies, this project offers a comprehensive, dynamic, and user-friendly solution, addressing the pain points of traditional appointment management methods while enhancing the overall experience for all stakeholders.

2.3 Research Questions

The implementation of an online doctor appointment system raises several key questions that guide its development:

- **How can a centralized platform improve the efficiency of healthcare appointment management?**

A centralized system consolidates critical functions like doctor profiles, availability management, and appointment booking into one platform. This ensures real-time updates, reduces errors, and enhances communication between patients, doctors, and administrators. For example, when administrators update a doctor's availability, patients can immediately see the changes while booking appointments.

- **What are the essential features of an online doctor appointment system?**

An effective system should include:

- Admin Dashboard: Manage doctor profiles, update schedules, and oversee appointment bookings.
- Doctor Dashboard (planned): View schedules, update availability, and manage patient appointments.
- User Dashboard (planned): Create accounts, browse doctors by specialization, check availability, and book appointments.

These features ensure seamless interaction between all stakeholders, making appointment management efficient and user-friendly.

2.4 Previous Research and Methodology

The development of online healthcare platforms has been explored extensively, with a focus on user-centered design, scalability, and real-time data management. Many systems highlight the importance of intuitive interfaces and secure data handling to ensure a positive user experience.

Our project adopts a similar methodology, leveraging Node.js, Express.js, React, MongoDB, and Tailwind CSS to create a modern and scalable system. The focus is on delivering tailored dashboards for administrators, doctors, and patients. The design emphasizes usability, efficient data handling, and meeting the unique needs of each user group.

3. Software and Hardware Requirement Specification

3.1 Methods

The development of the Online Doctor Appointment System follows an agile methodology. This approach allows for iterative progress, flexibility, and regular feedback during development. The main steps involved include:

- **Requirement Gathering:** Collaborating with stakeholders (patients, doctors, and administrators) to understand their needs and define system functionalities.
- **Design:** Creating wireframes and layout designs for the admin, doctor, and user dashboards.
- **Development:** Writing the application code using Node.js and Express.js for the backend, React for the frontend, and Tailwind CSS for responsive design.
- **Testing:** Conducting both manual and automated testing to ensure each feature works correctly, focusing on booking, scheduling, and data management.

3.2 Programming/Working Environment

The project is developed using a full-stack JavaScript environment with the following tools and technologies:

- **Backend:**
 - Node.js: A server-side runtime for building the backend.
 - Express.js: A web application framework for creating REST APIs and managing routes.
- **Frontend:**
 - React: A JavaScript library for building dynamic and interactive user interfaces.
 - Tailwind CSS: A utility-first CSS framework for rapid UI development.
- **Database:**
 - MongoDB: A NoSQL database for storing doctor profiles, user information, and appointment details.
- **Development Tools:**
 - **Visual Studio Code:** A code editor with extensions for Node.js, React, and Tailwind development.
 - **Git:** A version control system for tracking changes and enabling collaboration.
 - **Postman:** A tool for testing RESTful APIs and backend endpoints.

3.3 Requirements to Run the Application

Software Requirements:

- Node.js (Version 14.x or higher) and npm (Node Package Manager).
- Express.js and other required Node.js packages installed via npm.
- React and Tailwind CSS for the frontend.
- A modern web browser (e.g., Chrome, Firefox, Edge) to access the application.
- MongoDB: Installed locally or accessible via a cloud database service (e.g., MongoDB Atlas).
- Operating System: The app is compatible with Windows, macOS, and Linux.

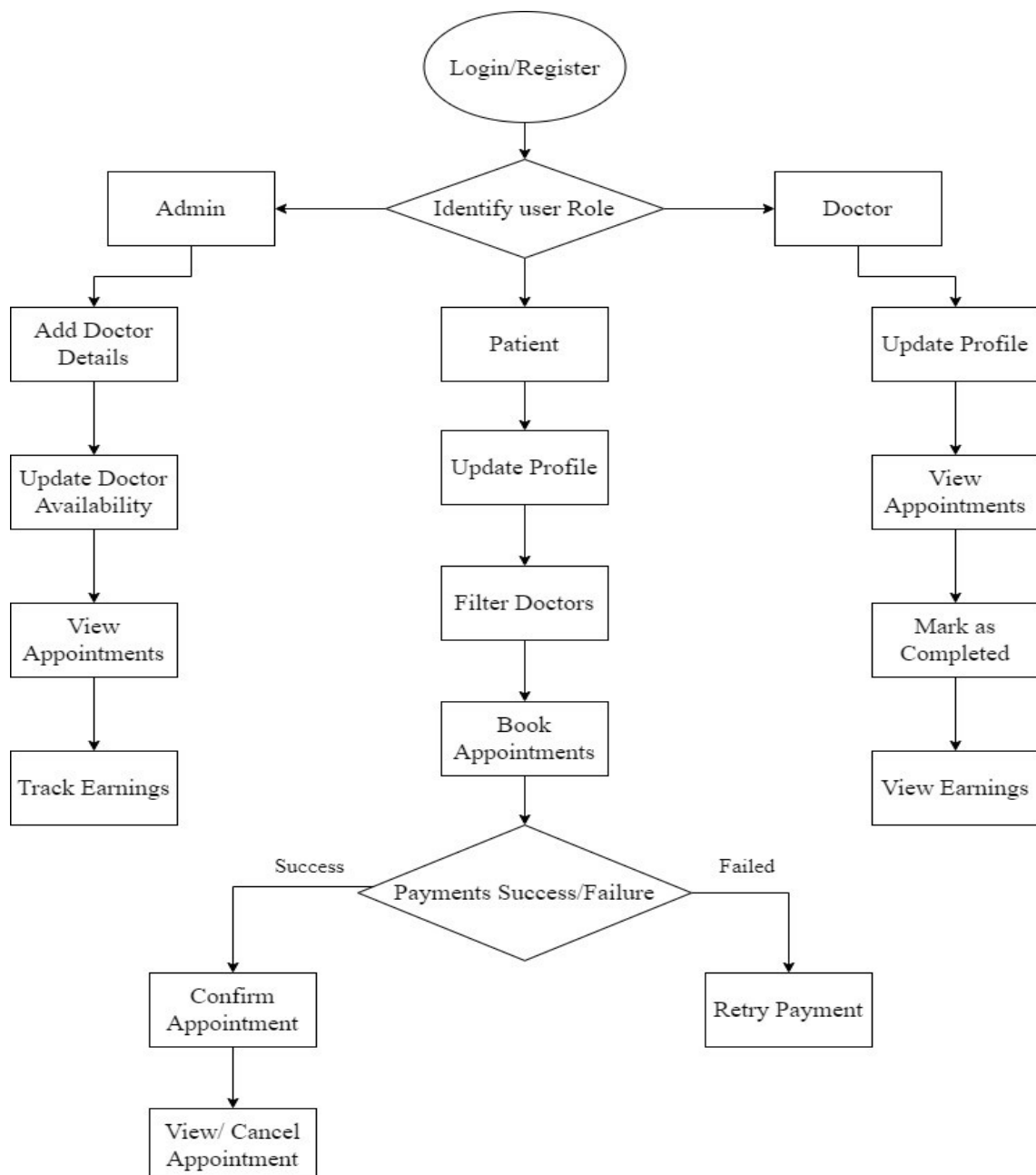
Hardware Requirements:

- **Server:**
 - Minimum 2 GB RAM and 20 GB storage for hosting the backend and database.
 - Internet-connected server for real-time accessibility.
- **Client Machines:**
 - A standard PC, laptop, tablet, or mobile device with an internet connection.
 - Minimum specs: 2 GB RAM and a modern web browser for running the application.

- **Network:**

- A stable and reliable internet connection to enable seamless interactions between the server and client.

4. Data Flow Diagram (DFD)



5. Database Analysis, Design, and Implementation

5.1 Database Analysis

In the Online Doctor Appointment System, the database plays a central role in managing critical information related to patients, doctors, and administrators. The database analysis focuses on identifying data requirements, their relationships, and the architecture needed for efficient implementation using MongoDB.

Data Requirements:

- **Doctor Data:** Includes personal and professional information such as name, email, phone number, specialization, availability, clinic address, and schedule.
- **Patient Data:** Contains user account details (name, email, phone number, gender, and password) and booking history (appointment date, time, doctor, and status).
- **Appointment Data:** Tracks appointment details, including doctor ID, patient ID, booking date, time, and status (e.g., confirmed, canceled).

Data Relationships:

- **Doctors and Appointments:** One-to-many relationship, as a doctor can have multiple appointments.
- **Patients and Appointments:** One-to-many relationship, since a patient can book appointments with different doctors.
- **Administrators and System Management:** Administrators manage doctor profiles and oversee appointments, requiring proper data access and control mechanisms.

5.2 Database Design

The database design for the system revolves around creating efficient collections and defining relationships within MongoDB to facilitate seamless data management.

Collections:

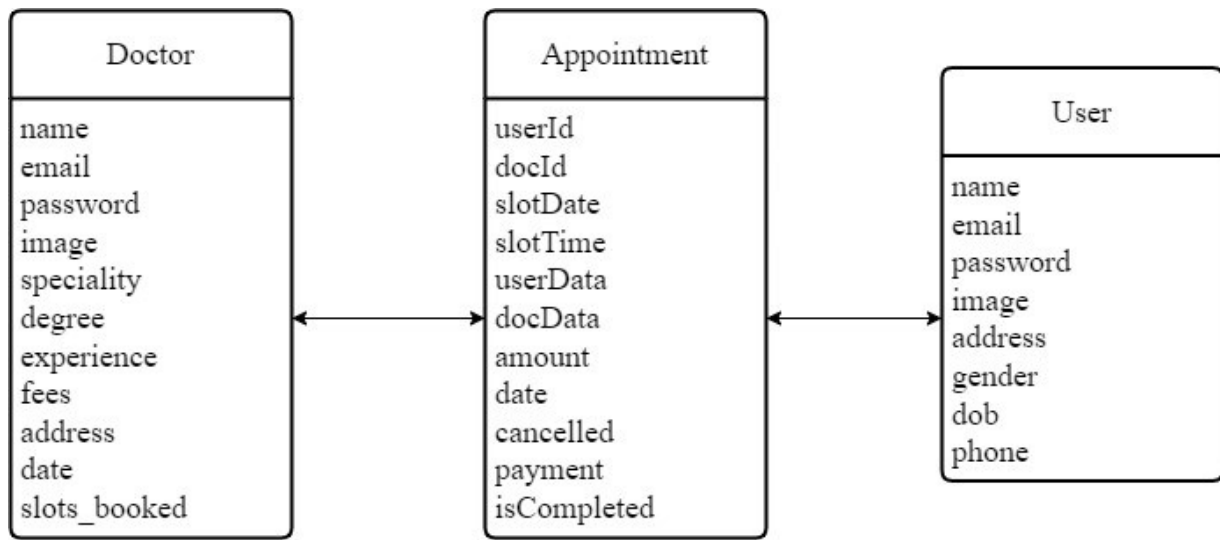
- **Doctors Collection:**
 - Fields: doctorId, name, email, phone, specialization, phone, availability, schedule, clinicAddress.
 - Relationships: Linked to the appointments collection by doctorId.
- **Patients Collection:**
 - Fields: patientId, name, email, phone, gender, password, bookingHistory.
 - Relationships: Linked to the appointments collection by patientId.

- **Appointments Collection:**

- Fields: appointmentId, doctorId, patientId, date, time, status.
- Relationships: References doctorId and patientId to connect doctors and patients.

Indexes:

Indexes will be implemented on commonly queried fields such as doctorId, patientId, appointmentId and email to improve query performance and data retrieval.



5.3 Database Implementation

The implementation phase will involve setting up MongoDB, defining schemas, and integrating the database with the application using Mongoose.

Steps for Implementation:

1. **MongoDB Setup:** Install and configure MongoDB locally or on a cloud platform like MongoDB Atlas.
2. **Schema Definition:**
 - Create schemas for doctors, patients, appointments, and admins using Mongoose.
 - Define relationships between collections where necessary.
3. **Data Integration:**
 - Populate the database with dummy data for testing and demonstration purposes.
 - Ensure data is clean, consistent, and conforms to the schema definitions.

4. Data Access Layer:

- Develop a robust data access layer using Mongoose to handle CRUD operations.
- Implement controllers to interact with the database for adding, retrieving, updating, and deleting records.

5. Testing and Optimization:

- Test the database to verify data integrity and relationships.
- Optimize queries and schema designs to enhance performance.

6. Program's Structure Analysis and GUI Construction (Project Snapshots)

6.1 Program Structure Analysis

The Online Doctor Appointment System is designed to include several interconnected modules, each serving a specific purpose to streamline the booking and management of doctor appointments. The structure of the system is divided into distinct sections for patients, doctors, and administrators. Below is an analysis of the key components of the program's structure:

Patient/User Dashboard:

- **Profile:** Displays personal information, including name, email, and contact details.
- **Appointments:** Shows the list of past, upcoming, and canceled appointments, along with appointment status.
- **Doctor Search:** Allows patients to search for doctors based on specialization, availability, and location.
- **Book Appointment:** Provides an interface for patients to select a doctor, date, and time to schedule an appointment.
- **Booking History:** Displays the history of all previous appointments, including status and details.

Doctor Dashboard:

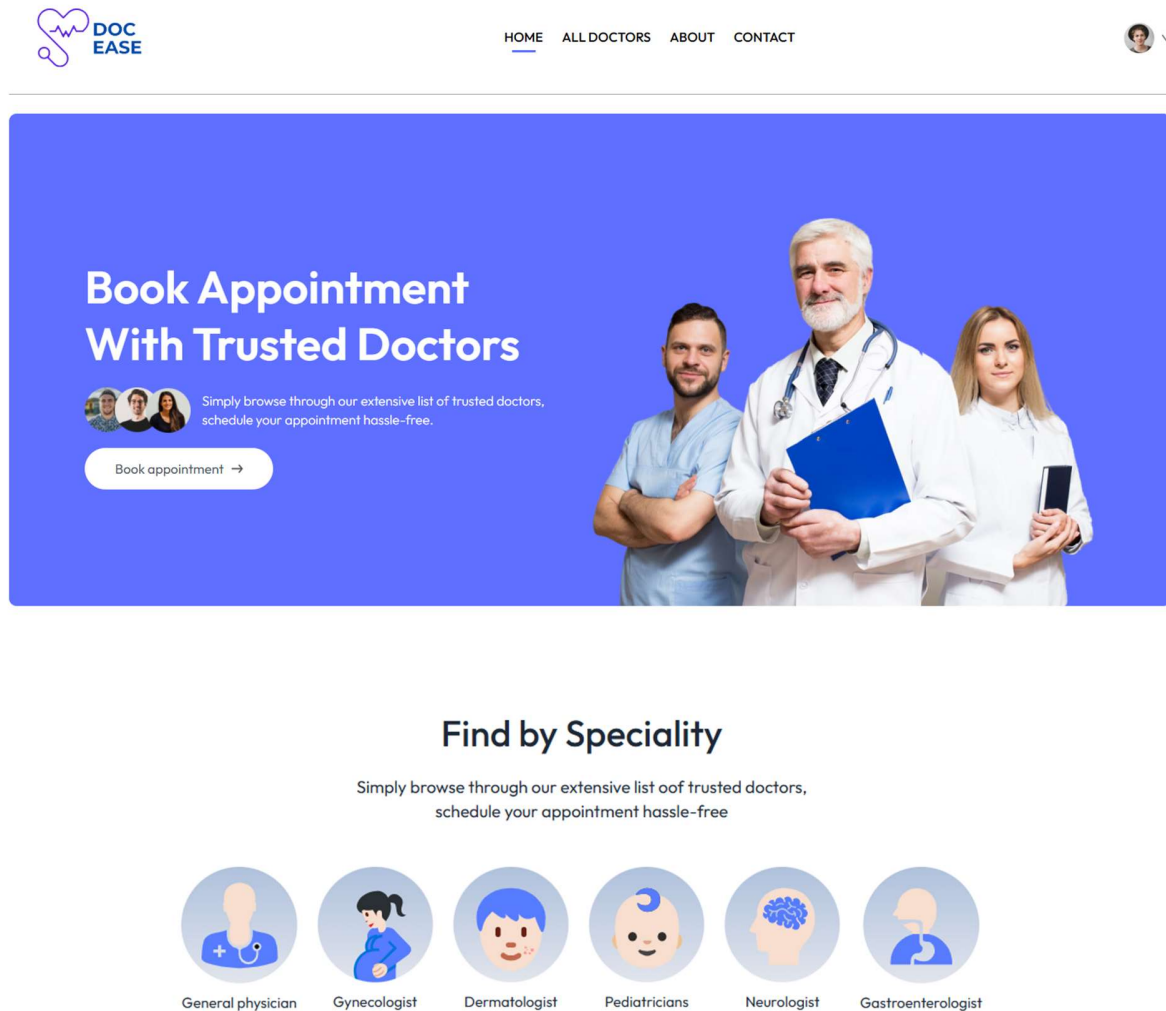
- Profile: Displays doctor's personal details, specialization, contact information, and clinic address.
- Appointments: Shows the list of appointments for the doctor, including both upcoming and past ones.
- Schedule Management: Allows doctors to set and update their availability for appointments.
- Patient Information: Displays information about patients attending the doctor's appointments.
- Appointment Management: Allows doctors to confirm, reschedule, or cancel appointments.

Admin Dashboard:

- Doctor Management: Enables admins to add, update, and remove doctor profiles. Admins can also view a list of all registered doctors along with their availability and specialization.
- Appointment Management: Provides an overview of all appointments made on the platform, including their status (confirmed, canceled, etc.), and the ability to manage and reschedule them.











This structure ensures a smooth flow of information between users and administrators while maintaining distinct functionalities for each user role. Each dashboard provides specific tools and access, ensuring a personalized and efficient experience for patients, doctors, and admins.

6.2 GUI Construction (Project Snapshots)



Top Doctors to Book

Simply browse through our extensive list of trusted doctors.

 ● Available Dr. Richard James General physician	 ● Available Dr. Emily Larson Gynecologist	 ● Available Dr. Sarah Patel Dermatologist	 ● Available Dr. Christopher Lee Pediatricians	 ● Available Dr. Jennifer Garcia Neurologist	 ● Available Dr. Andrew Williams Neurologist
 ● Available Dr. Christopher Davis General physician	 ● Available Dr. Timothy White Gynecologist	 ● Available Dr. Ava Mitchell Dermatologist	 ● Available Dr. Jeffrey King Pediatricians		

Book Appointment With 100+ Trusted Doctors

Create account



Lorem ipsum dolor sit amet consectetur adipisicing elit. Iure consectetur itaque, incididunt quam architecto amet ducimus, nobis cupiditate ratione aspernatur magnam doloribus? Iste queraat possimus expedita fugiat molestiae fuga consequatur.

COMPANY

Home
About us
Contact us
Privacy policy

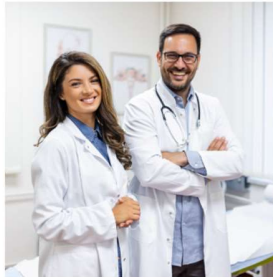
GET IN TOUCH

+91-98765-43210
docease@gmail.com

© 2024 DocEase. All rights reserved.

5.2.1 Home Page

ABOUT US



Welcome to Prescripto, your trusted partner in managing your healthcare needs conveniently and efficiently. At Prescripto, we understand the challenges individuals face when it comes to scheduling doctor appointments and managing their health records.

Prescripto is committed to excellence in healthcare technology. We continuously strive to enhance our platform, integrating the latest advancements to improve user experience and deliver superior service. Whether you're booking your first appointment or managing ongoing care, Prescripto is here to support you every step of the way.

Our Vision

Our vision at Prescripto is to create a seamless healthcare experience for every user. We aim to bridge the gap between patients and healthcare providers, making it easier for you to access the care you need, when you need it.

WHY CHOOSE US

EFFICIENCY:

Streamlined appointment scheduling that fits into your busy lifestyle.

CONVENIENCE:

Access to a network of trusted healthcare professionals in your area.

PERSONALIZATION:

Tailored recommendations and reminders to help you stay on top of your health.

5.2.2 About Page

Browse through the doctors specialist.

General physician


Gynecologist

Dermatologist


Pediatricians

Neurologist


Gastroenterologist




● Available
Dr. Richard James
General physician



● Available
Dr. Emily Larson
Gynecologist




● Available
Dr. Sarah Patel
Dermatologist




● Available
Dr. Christopher Lee
Pediatricians




● Available
Dr. Jennifer Garcia
Neurologist




● Available
Dr. Andrew Williams
Neurologist




● Available
Dr. Christopher Davis
General physician




● Available
Dr. Timothy White
Gynecologist



● Available
Dr. Ava Mitchell
Dermatologist



● Available
Dr. Jeffrey King
Pediatricians



● Available
Dr. Zoe Kelly
Pediatricians



● Available
Dr. Patrick Harris
Neurologist

5.2.3 All Doctors



Bhuvesh Mittal

CONTACT INFORMATION

Email id: bhuveshmittal@gmail.com
Phone: +91 1234567890
Address: Ansal Orchard County, Sector 115
 Royal Estate 2, Sector 20

BASIC INFORMATION




Gender: Male
Birthday: 2000-01-20

Edit

5.2.4 My Profile



My Appointments

	Dr. Richard James General physician Address: 17th Cross, Richmond Circle, Ring Road, London Date & Time: 2, Dec, 2024 7:00 PM	Pay Online Cancel appointment
	Dr. Emily Larson Gynecologist Address: 27th Cross, Richmond Circle, Ring Road, London Date & Time: 2, Dec, 2024 7:00 PM	Pay Online Cancel appointment
	Dr. Sarah Patel Dermatologist Address: 37th Cross, Richmond Circle, Ring Road, London Date & Time: 2, Dec, 2024 7:00 PM	Pay Online Cancel appointment

5.2.5 My Appointments

 Dashboard

 Appointments

 Add Doctor

 Doctors List

Add Doctor



Upload doctor picture

Doctor name

Name

Speciality

General physician

Doctor Email

Email

Education

Education

Doctor Password

Password

Address

address 1

address 2

Experience

1 Year

Fees

fees

About Doctor

write about doctor

Add doctor

5.2.6 Add doctors by admin

 Dashboard

 Appointments

 Add Doctor

 Doctors List

All Doctors



Dr. Richard James
General physician

☒ Available



Dr. Emily Larson
Gynecologist

☒ Available



Dr. Sarah Patel
Dermatologist

☒ Available



Dr. Christopher Lee
Pediatricians

☒ Available



Dr. Jennifer Garcia
Neurologist

☒ Available



Dr. Andrew Williams
Neurologist

☒ Available



Dr. Christopher Davis
General physician

☒ Available



Dr. Timothy White
Gynecologist

☒ Available



Dr. Ava Mitchell
Dermatologist

☒ Available



Dr. Jeffrey King
Pediatricians

☒ Available



Dr. Zoe Kelly
Pediatricians

☒ Available



Dr. Patrick Harris
Neurologist

☒ Available

5.2.7 Doctors List

A screenshot of the Admin Login form. It has a title 'Admin Login' in blue. Below it are two input fields: 'Email' and 'Password'. A blue 'Login' button is positioned below the password field. At the bottom, there is a link 'Doctor Login? Click here'.

5.2.8 Admin Login

6.3 GUI Design Considerations

The Online Doctor Appointment System focuses on providing a user-friendly and efficient interface through the following design principles:

- **User Experience (UX):** Intuitive dashboards tailored to user roles ensure seamless navigation and task completion.
- **Responsive Design:** The interface adapts to various screen sizes for a consistent experience across desktop and mobile devices.
- **Consistency:** Uniform design elements, including colors, typography, and reusable components, create a cohesive appearance.
- **Accessibility:** Adherence to web accessibility standards ensures usability for individuals with disabilities, including features like keyboard navigation and high-contrast visuals.
- **Role-Specific Dashboards:** Each dashboard is optimized for its target users, focusing on key tasks like appointment booking for patients and schedule management for doctors.

These principles ensure a streamlined, visually appealing, and accessible system for all users.

7. Code-Implementation and Database Connections (If any)

```
1  import mongoose from "mongoose";
2
3  const connectDB = async () =>{
4
5      mongoose.connection.on('connected', ()=> console.log("DataBase Connected"))
6
7      await mongoose.connect(`${process.env.MONGODB_URI}/docease`)
8  }
9
10 export default connectDB;
```

```
1  MONGODB_URI='mongodb+srv://doceaseadmin:docease@cluster0.ewy66.mongodb.net'
2  CLOUDINARY_NAME='duarhkbwp'
3  CLOUDINARY_API_KEY='619321447624451'
4  CLOUDINARY_SECRET_KEY='MAGuRegpdeJvbipN09j33Q2ZHg'
5  ADMIN_EMAIL='admin@docease.com'
6  ADMIN_PASSWORD='qwerty123'
7  JWT_SECRET='docease'
```

6.1 Database Connection

8. Future Scope

The Online Doctor Appointment System, while functional and robust, has immense potential for further enhancements. The following areas have been identified for future development.

1. Advanced Analytics and Reporting:

- Developing detailed analytics tools for administrators to monitor appointment trends, doctor availability, and payment data.
- Integrating visual dashboards to present insights, enabling data-driven decision-making and system optimization.

2. Mobile Application Development:

- Creating mobile applications for patients, doctors, and administrators to provide seamless access to all features on the go.
- Adding real-time push notifications for appointment confirmations, reminders, and updates.

3. Enhanced Security Measures:

- Further strengthening security with multi-factor authentication, advanced data encryption, and comprehensive role-based access controls.
- Conducting regular security audits to maintain the integrity and confidentiality of user data.

4. Integration with External Systems:

- Expanding the system's capabilities by integrating telemedicine tools for virtual consultations.
- Synchronizing doctor schedules with external calendar systems, such as Google Calendar, to streamline time management.

5. Improved User Experience (UX):

- Continuously refining the user interface based on feedback to ensure a smoother and more intuitive experience.
- Introducing personalized features such as patient health history summaries, doctor availability alerts, and customized recommendations.

By focusing on these areas, the system can evolve into an even more comprehensive and user-friendly platform, addressing the growing demands of patients, doctors, and administrators while remaining adaptable to future needs.

9. Conclusion

The development of the Online Doctor Appointment System marks a significant step toward streamlining healthcare management by providing a centralized platform for patients, doctors, and administrators. Built using Node.js, Express.js, React, MongoDB, and Tailwind CSS, the system integrates various functionalities to enhance user convenience and operational efficiency.

The system's design simplifies critical processes, such as patient registration, appointment booking, and doctor management, through dedicated dashboards for Admin, Doctor, and User (Patient) roles. Patients can effortlessly create accounts, browse available doctors, and book appointments based on availability and specialization, while doctors can manage schedules and update availability. The admin dashboard allows for seamless management of doctor profiles, appointment monitoring, and user data, ensuring the platform remains efficient and up-to-date.

While the current implementation focuses on core functionalities, the project has a clear path for future growth. The planned enhancements, including advanced reporting features, enhanced security measures, and a mobile application, will further refine the system's usability and scalability. Database integration with MongoDB has already laid the foundation for efficient data management and storage, enabling real-time updates and complex queries.

The responsive and intuitive interface ensures accessibility across various devices, making the platform convenient for all users. Future efforts will also focus on strengthening security through encryption, multi-factor authentication, and regular system audits to protect sensitive user data.

In conclusion, this project establishes a robust framework for improving healthcare access and management. With planned upgrades and continuous user feedback, the system is well-positioned to evolve into a comprehensive and scalable solution for modern healthcare needs.

10. References

1. *Express - Node.js web application framework*. (n.d.). Expressjs.com. Retrieved November 27, 2024, from <https://expressjs.com/>
2. GreatStack [@GreatStackDev]. (n.d.). *Build full stack doctor appointment booking system using MERN stack | complete react JS project*. Youtube. Retrieved November 27, 2024, from <https://www.youtube.com/watch?v=eRTTIS0zaW8>
3. Groom, J. (2016). *JSON Web Tokens - jwt.io*. <https://jwt.io/>
4. *Image and video upload, storage, optimization and CDN*. (2024, February 6). Cloudinary. <https://cloudinary.com/>
5. *MongoDB: The developer data platform*. (n.d.). MongoDB. Retrieved November 27, 2024, from <https://www.mongodb.com/>
6. *Npm: Express*. (n.d.). Npm. Retrieved November 27, 2024, from <https://www.npmjs.com/package/express>
7. *Rapidly build modern websites without ever leaving your HTML*. (n.d.). Tailwindcss.com. Retrieved November 27, 2024, from <https://tailwindcss.com/>
8. *React*. (n.d.). React.dev. Retrieved November 27, 2024, from <https://react.dev/>
9. *React tutorial*. (n.d.). W3schools.com. Retrieved November 27, 2024, from <https://www.w3schools.com/REACT/default.asp>
10. *Run JavaScript everywhere*. (n.d.). Nodejs.org. Retrieved November 27, 2024, from <https://nodejs.org/en>