International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03235
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

DOCTO APPLICATION USING MERN STACK

Ansh Sharma [1], Anurag Malik [2], Ayush Singh [3], Basu Kumar [4], Mohd Samir[5]
[1,3,4,5]B.Tech Scholar, Computer Science and Engineering, Moradabad Institute of Technology, Moradabad , India [2]Associate Professor, Computer Science and Engineering, Moradabad Institute of Technology, Moradabad, India [1]anshwork9@gmail.com , [2] anurag_malik@rediffmail.com
[3]ayush080804.rks@gmail.com , [4]basukumar94103@gmail.com , [5]mohdsamir30102001@gmail.com

Abstract: - In the digital age, the demand for healthcare services through online platforms has increased significantly. This research paper presents a comprehensive analysis and development of a doctor appointment booking application integrated with a medicine purchasing feature using the MERN (MongoDB[9], Express.js, React[7], Node.js[8]) stack [1]. This application enhances user convenience by providing a seamless platform to book doctor appointments and purchase prescribed medicines online. The paper elaborates on the architecture, implementation methodology, features, future scalability, and data flow diagrams to provide a deeper technical insight.

Key Words: — MERN Stack [1], Doctor Appointment System, Online Medicine Purchase, Healthcare Application, Full-Stack Development, Telemedicine [2]

1. INTRODUCTION

Access to timely medical care and medication is crucial for maintaining public health. In traditional healthcare systems, patients often face significant challenges such as long waiting times, manual appointment scheduling errors, and delays in accessing necessary medicines. These inefficiencies not only inconvenience patients but also place additional burdens on healthcare providers. Furthermore, the reliance on physical visits for booking appointments and purchasing medications increases the time and effort required for both patients and medical staff.

As healthcare demands grow, there is a pressing need for technological solutions that can streamline these processes. The advent of digital platforms provides an opportunity to improve healthcare delivery by offering patients a more efficient and accessible way to book appointments and receive medications. Our proposed system addresses these challenges by leveraging the MERN (MongoDB[9], Express.js, React[7], Node.js[8]) stack [1] to deliver a comprehensive, integrated solution.

The system provides users with a seamless experience, allowing them to book doctor appointments, upload prescriptions, and order medicines—all within a single platform. This integration reduces the need for multiple visits, minimizes human error, and enhances operational efficiency. The user-friendly interface ensures that patients can easily navigate the system, while secure data handling through advanced authentication methods (JWT[12]) safeguards sensitive information. With the rise of telemedicine and online health services, there is a growing demand for iintegrated solutions that not only facilitate virtual consultations but also manage medication orders .Our system bridges this gap by automating critical processes such as appointment scheduling, prescription verification, and order tracking. This automation enhances patient satisfaction by reducing wait times and ensures that prescriptions are accurately processed and delivered.

The proposed solution not only improves the patient experience but also assists healthcare providers by offering real-time updates on patient appointments and medication requests. This digital transformation in healthcare delivery enhances operational efficiency, reduces administrative overhead, and provides a scalable framework capable of accommodating future innovations and increasing patient demands.

2. LITERATURE REVIEW

Several studies have explored the implementation of telemedicine platforms. Past research highlights the importance of user-centric designs and the need for secure data handling. Existing systems lack integration between appointment booking and medicine purchasing. Our system addresses this gap by offering an all-in-one healthcare solution.

Previous systems primarily focused on either appointment booking or medication delivery but lacked an integrated approach. By leveraging the MERN stack[1], we ensure a robust, scalable, and secure system for comprehensive healthcare management. According to WHO [19], telemedicine is a growing need in modern healthcare, especially post-pandemic, which our system aims to support.

3. PROBLEM STATEMENT

Patients face challenges such as long wait times for appointments, difficulty accessing prescriptions, and the inconvenience of visiting physical pharmacies. There is a need for a system that enables patients to

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03235
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

book doctor appointments and order medicines through a unified platform.

The absence of a centralized platform results fragmented services and increased processing time. Our proposed solution bridges these gaps by providing an integrated ecosystem that enhances user satisfaction and operational efficiency.

4. OBJECTIVES

- Develop a secure and efficient doctor appointment booking system.
- Enable online purchase and delivery of prescribed medicines.
- Ensure data privacy and security compliance.
- Provide an intuitive and responsive user interface.
- Implement an efficient order tracking and management system.
- Ensure real-time synchronization between appointment scheduling and order management.

5. SYSTEM ARCHITECTURE

5.1 TECHNOLOGY STACK

- Frontend: React[7] (for dynamic and responsive UI)
- Backend: Node.js[8] with Express (for API development)
- Database: MongoDB[9] (for storing user, doctor, and order information)
- Authentication: JWT[12] (for secure user authentication)
- Payment Integration: Stripe[20] or Razorpay[5] (for medicine purchase)

5.2 SYSTEM COMPONENTS

1. **User Interface Layer:** Built using React, this layer provides users with a seamless experience for booking appointments and ordering medicines. It includes dynamic forms, real-time updates, and user- friendly dashboards.
2. **API Layer:** Developed using Express.js, it handles all client- server interactions, including user authentication, appointment scheduling, and medicine order management.
3. **Database Layer:** MongoDB[9] stores all the core information such as user profiles, doctor details, appointments, and medicine orders. It ensures scalability and quick retrieval of data while supporting complex queries.
4. **Payment Gateway**: Integrated with Stripe[20] or Razorpay[5], enabling secure transactions for medicine purchases and maintaining transaction logs.

5.3 SYSTEM WORKFLOW

1. **User Authentication:** Secure login and registration for patients and doctors using JWT[12].
2. **Doctor Appointment:** Users can search for available doctors, view profiles, and book available slots.
3. **Medicine Purchase**: Users can upload prescriptions and place medicine orders, which are verified and processed.
4. **Order Management:** Users can track medicine orders in real-time from order placement to delivery.
5. **Admin Panel:** Manages doctor availability, prescription verification, and order dispatching.

5.4 Data Flow Diagram (DFD) Level 0

DFD (Context Diagram)

This diagram represents the entire system's interaction with external entities:

- User → Book Appointment, Order Medicine
- Doctor → Manage Appointments, Update Availability
- Admin → Verify Prescriptions, Manage Orders
- Payment Gateway → Process Payments**(fig.1)**

(These Names and photographs shown are only for representational purpose)

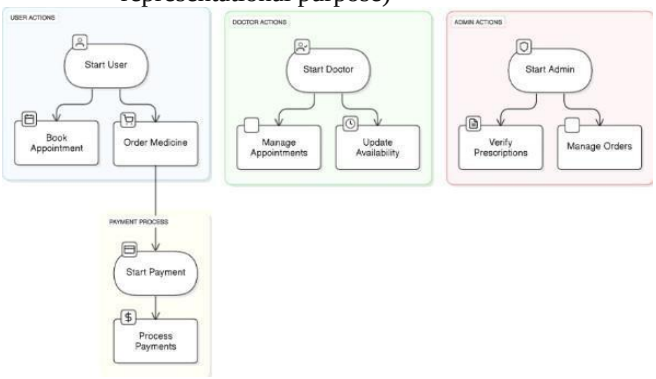

Fig. 1 Overall structure

Level 1 DFD (Detailed System Flow)

1. **User Module:**
   - Register/Login
   - Book Appointment
   - Order Medicine
   - Track Orders
2. **Doctor Module**
   - Manage Appointments
   - View Patient Details
3. **Admin Moduled:**
   - Verify Medicine Orders
   - Process and Ship Medicines

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03235
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

4. Payment Module:

- Initiate Payment
- Confirm Payment Status**(fig.2)**

(These Names and photographs shown are only for representational purpose).

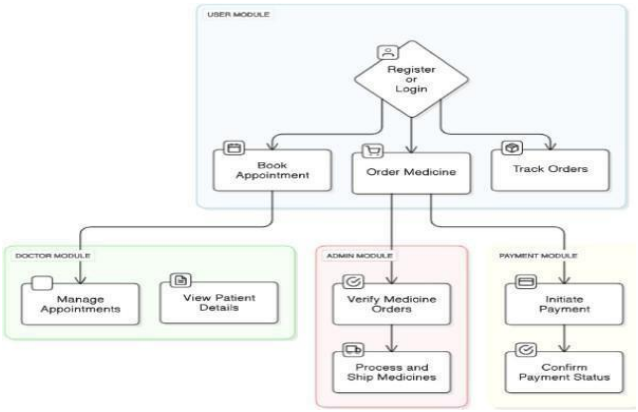

Fig. 2 Overall structure

Level 2 DFD (Detailed System Flow)

1. **User Requests:**

- Input: Registration details, Appointment details, Prescription
- Output: confirmation, Payment receipt, Order status

2. **Doctor Updates:**

- Input: Availability schedule, Patient notes
- Output: availability, history

3. **Admin Verification:**

- Input: Updated Appointment Prescription, Payment confirmation
- Output: Order approval, Shipment initiation**(fig.3)**

(The Names and photographs shown are only for representational purpose).

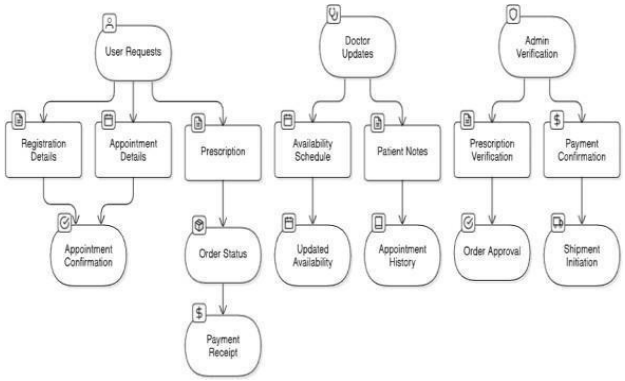

Fig. 3. Overall Strucure Of a System

6. METHODOLOGY AND IMPLEMENTATION

6.1. Frontend Development

- Built using React[7] for a modular and interactive interface.
- React[7] Router for navigation between booking, order tracking, and dashboards.
- Dynamic forms and real-time status updates for experience.
- Styled using Tailwind CSS[10] to ensure a responsive and clean design.
- Accessibility standards from W3C [15] were followed to ensure the application is usable for individuals with disabilities.

6.2. Backend Development

- RESTful APIs created using Node.js[8] and Express.
- MongoDB[9] used for scalable and flexible data storage.
- Secure endpoints for handling sensitive information and user authentication.
- Utility functions from Lodash [21] were used for efficient data manipulation and array processing in the backend logic.

6.3. User Authentication

- JWT[12] ensures secure login and session.
- Password encryption using bcrypt for enhanced security.
- Firebase [16] can be used as an alternative authentication method or for integrating real-time notifications.

6.4. Medicine Purchase Workflow

- Users upload prescriptions
- Admin verifies and processes orders.
- Payment is handled through integrated

6.5. Appointment Booking

- Users select available slots based on realtime doctor availability.
- Automated notifications for booking confirmations and reminders.

6.6. Implementation

The implementation process involved a structured approach across all application modules to ensure functionality, security, and user-friendliness.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03235
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

1. User Registration Authentication:

- Implemented secure JWT[12] authentication for users and doctors.
- User credentials are encrypted with bcrypt and securely stored in MongoDB[9].
- Integrated role-based access control (User, Doctor, Admin) to manage permissions
- Frontend includes input validation to prevent SQL injection and XSS attacks. **(fig.4**)
  (These Names and photographs shown are only for representational purpose).

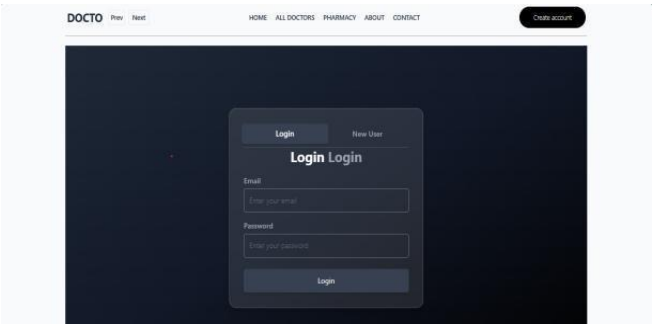

Fig. 4 Login Page

2. Doctor Appointment Scheduling:

- Users can view doctor profiles and select available appointment slots.
- Appointment requests are updated in realtime in the MongoDB[9] database
- Available time slots include:
  Morning: 9:00 AM - 10:00 AM, 10:30 AM - 11:30AM Afternoon:1:00 PM - 2:00 PM, 2:30 PM - 3:30 PM Evening: 5:00 PM - 6:00 PM, 6:30 PM - 7:30 PM.**(fig.5)**
  (These Names and photographs shown are only for representational purpose).

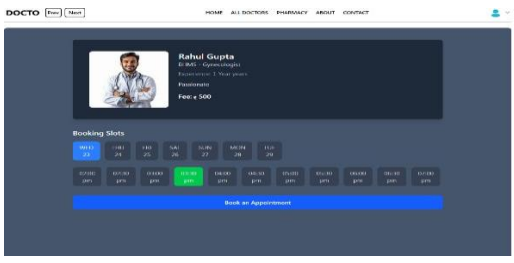

Fig. 5 Doctor Appointment Scheduling

3. Medicine Ordering System:

- Users upload prescriptions which are stored securely.
- Admin verifies prescriptions and processes orders.
- Users receive real-time notifications for payment confirmation and order updates.
- Images such as prescriptions or doctor profile pictures are stored and managed using Cloudinary [4] for optimized delivery and storage.
- Payment is integrated using Stripe or Razorpay[5] APIs for smooth transactions**.(fig.6)**

(These Names and photographs shown are only for representational purpose)

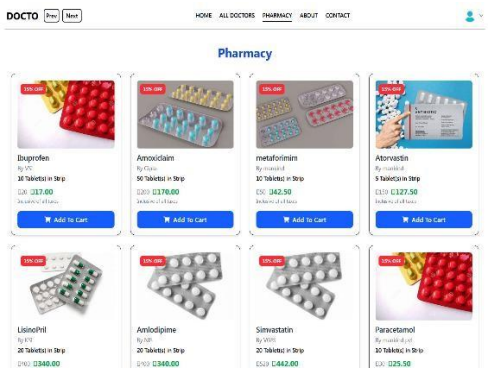

Fig. 6. Medicine Ordering Page

4. Admin Panel:

- Admin users can manage doctor availability, verify prescriptions, and order statuses.
- Custom dashboards provide analytics on appointment bookings and purchases.
  Monitor and update medicine order statuses. **(fig.7)**
  (These Names and photographs shown are only for representational purpose)

Fig.7. Admin Panel

5. Testing And Deployment

- Each module was tested for performance, security, and reliability.
- The application is deployed on a cloud platform ensuring scalability and high availability.
- Integration tests ensured smooth communication between frontend, backend, and database

- CI/CD pipelines were implemented using Jenkins[18] to automate testing and deployment processes.
- Docker [17] containers were used to ensure consistent development environments and ease of deployment across platforms.
- API endpoints were thoroughly tested using Postman [13] to ensure reliability and correct responses under different conditions.

7. RESULTS AND DECLARATION

The developed system successfully meets the objectives:

- Efficiency: Reduced appointment scheduling time
- Convenience: One-stop platform for consultations and medicine orders.
- Security: Encrypted user data and secure transactions.
- Scalability: Flexible architecture allowing future feature integration.

8. CONCLUSION

The doctor appointment booking and medicine purchase application provide a comprehensive, user- friendly solution to modern healthcare needs. Retrieved from Future enhancements may include AI-based diagnosis suggestions, multilingual support, and integration with wearable health devices.

9. REFERENCES

1. Mern, M. (2022). Full-Stack Development with MERN. Journal of Web Development.
2. Kumar, A., & Sharma, P. (2021). "Telemedicine and Its Future." International Journal of Healthcare Technology.
3. Mozilla. (2023). Web Security and Authentication Principles. MDN Web Docs. Retrieved from https://developer.mozilla.org
4. Cloudinary. (n.d.). Image and Video Management Platform. Retrieved from https://cloudinary.com
5. Razorpay. (n.d.). Online Payment Gateway for India. Retrieved from https://razorpay.com.
6. OpenAI. (2023). ChatGPT: AI Language Model for Code Assistance. Retrieved from https://openai.com/chatgpt.
7. React Documentation. (n.d.). React – A JavaScript Library for Building User Interfaces. Retrieved from https://reactjs.org.
8. Node.js & Express Docs. (n.d.). Backend Framework for Scalable Applications. https://expressjs.com
9. MongoDB. (n.d.). Flexible NoSQL Database. from Retrieved https://www.mongodb.com
10. Tailwind CSS Docs. (n.d.). Utility- First CSS Framework. Retrieved from https://tailwindcss.com.
11. Socket.IO. (n.d.). Real-Time Engine for Web Apps. Retrieved from https://socket.io (For enabling real-time features in future chat or consultation modules)
12. JWT.io. (n.d.). JSON Web Tokens Introduction and Implementation. Retrieved from https://jwt.io/introduction.
13. Postman. (n.d.). API Platform for Building and Testing APIs. Retrieved from https://www.postman.com.
14. GitHub. (n.d.). Version Control and Collaboration Platform. Retrieved from https://github.com.
15. W3C. (2023). Web Accessibility Standards. Retrieved from https://www.w3.org/WAI/
16. Google. (n.d.). Firebase Authentication Documentation. Retrieved from https://firebase.google.com/docs/aut
17. Docker. (n.d.). What is Docker?. Retrieved from https://www.docker.com/resources/whatcontai ner/
18. Jenkins. (n.d.). The leading open-source automation server. Retrieved from https://www.jenkins.io
19. WHO. (2022). Telemedicine: Opportunities and developments in Member States. World Health Organization. Retrieved from https://www.who.int/publications
20. Stripe. (n.d.). Online payments infrastructure for the internet. Retrieved from https://stripe.com

21. Lodash Documentation. (n.d.). A modern JavaScript utility library delivering modularity, performance & extras. Retrieved from https://lodash.com