

Medicare: A Doctor Appointment Application System

Kartik Koul
Dept. of Information Technology
Maharaja Agrasen
Institute of Technology
Sector 22, Rohini, Delhi, 110086
karrtik549gmail.com

Depansu Sharma
Dept. of Information Technology
Maharaja Agrasen
Institute of Technology Sector 22,
Rohini, Delhi, 110086
depanshu.sharma9873@gmail.com

Tejna Khosla
Dept. of Information Technology
Maharaja Agrasen
Institute of Technology Sector 22,
Rohini, Delhi, 110086
tejnakhosla@mait.ac.in

K.C. Tripathi

Dept. of Information Technology
Maharaja Agrasen Institute of Technology
Sector 22, Rohini, Delhi, 110086
kctripathi@mait.ac.in

M.L. Sharma

Dept. of Information Technology
Maharaja Agrasen Institute of Technology
Sector 22, Rohini, Delhi, 110086
mlsharma@mait.ac.in

ABSTRACT

The field of doctor appointment scheduling is currently experiencing a significant transformation, utilizing advanced technologies to enhance user experiences. Traditional manual processes, such as depending on receptionists for scheduling appointments, come with inherent limitations that hinder overall user satisfaction. This study aims to tackle these challenges by introducing an advanced Doctor Appointment System, eliminating the drawbacks associated with manual scheduling and ensuring a seamless and patient-focused experience. In traditional appointment scheduling, issues like limited personalization, delayed response times, and a lack of real-time information arise when communicating preferences and expectations to receptionists. In contrast, the Doctor Appointment System, developed on the MERN stack, employs cutting-edge technologies to overcome these limitations. The constraints of traditional scheduling, such as restricted hours and limited expertise, are overcome by the 24/7 accessibility and comprehensive information provided by the MERN stack appointment system. Patients now have the convenience of scheduling appointments, accessing medical information, and managing their healthcare needs. This system establishes a platform that delivers personalized, real-time, and user-friendly healthcare solutions, empowering patients to take control of their appointments efficiently and

pleasantly. This research endeavors to evaluate the effectiveness of this innovative approach and shed light on its potential to reshape the future of doctor appointment systems.

Keyword - Appointment Booking, User Reviews, Secure Authentication, User Accounts.

I. INTRODUCTION

The healthcare industry is experiencing a rapid evolution, marked by a growing dependence on digital platforms to enhance user experiences. This paper addresses the development of a doctor appointment system based on the MERN stack. It presents the design, development, and evaluation of a comprehensive appointment scheduling website built on MongoDB, Express.js, React, and Node.js. The system incorporates crucial features, including secure user authentication, a review and rating system, practitioner ratings, and seamless communication using the Context API. MongoDB is utilized for data storage, Node.js for server-side scripting, React for the user interface, and JWT (JSON Web Token) for secure user authentication. The research assesses the system's functionality, performance, and user satisfaction, offering insights into the potential of MERN stack technologies in the healthcare domain. This paper delves into the complex realm of modern web development, specifically focusing on the creation and

evaluation of a doctor appointment system. As the digital era reshapes the way healthcare appointments are scheduled and managed, the integration of advanced technologies becomes crucial. This study addresses the limitations inherent in traditional appointment scheduling methods and proposes a solution by harnessing the capabilities of the MERN stack. The central issue revolves around improving user experiences in healthcare appointment scheduling, recognizing the need for a more streamlined, personalized, and technologically advanced platform. The literature review contextualizes the current state of web development within the healthcare sector, acknowledging historical contributions that have shaped the industry. Historical milestones, such as the evolution of online healthcare scheduling platforms, lay the foundation for understanding the progress made in the field. The review underscores the limitations of manual appointment scheduling and outdated web technologies, emphasizing the growing demand for more sophisticated and user-friendly systems. By examining existing literature, this paper aims to position itself within the state-of-the-art knowledge, identifying gaps that the proposed MERN stack-based Doctor Appointment System seeks to address.

The study stands out by introducing the MERN stack as an innovative and integrated solution for web development in the healthcare industry. It highlights the significant differences between traditional methodologies and the proposed framework, emphasizing the advantages of MongoDB for efficient data storage, Express.js and Node.js for robust server-side scripting, and React for creating dynamic and responsive user interfaces. The literature review serves as a foundation for introducing the unique contribution of this paper – a comprehensive examination of a MERN stack-based doctor appointment system, with a detailed analysis of its features, functionality, and user experience.

II. LITERATURE REVIEW

In the dynamic environment of healthcare technology, contemporary appointment scheduling platforms have been influenced by theoretical frameworks emphasizing patient-centric design, collaborative community dynamics, and the integration of advanced

technologies. Analysing these frameworks offers insights into the evolution of healthcare technology and sets the stage for a comparative examination with the innovative features of our doctor appointment system.

Timely Information:

Advantage: Conventional systems in healthcare often rely on static information, pamphlets, or periodic updates. The real-time update feature in our system ensures that patients receive the latest and most pertinent information instantly, enabling them to take advantage of time-sensitive opportunities.

Personalization:

Advantage: Unlike traditional healthcare systems that provide generic information to all patients, the personalized nature of our system tailors updates based on individual preferences and medical history. This targeted approach enhances patient engagement and satisfaction.

III. METHODOLOGY

The basis of this research is the careful design and implementation of the doctor appointment system website, leveraging the MERN group, to ensure a seamless and secure user experience. This Methods section provides clarity and insight into the processes that support platform quality by describing the methods used to obtain the results.

System Architecture:

Description: The essence of this methodology is to delineate the system architecture for the Doctor Appointment System. This encompasses comprehensive insights into the data storage mechanism employed by MongoDB, the server-side logic processing by Express.js, the user interface rendering by React, and the runtime environment management by Node.js.[] Furnish an intricate depiction of the organizational structure within the MERN stack.

Purpose: This serves as the groundwork for comprehending how diverse technologies collaborate to create a streamlined and proficient doctor appointment system.

Database Design and Implementation:

Description: Details the database design process of how MongoDB is tailored to handle diverse types of data pertinent to the healthcare domain []. It explores implementation concepts, encompassing data structures and indexing strategies.

Goal: Enhancing transparency in the database configuration to facilitate efficient data retrieval and storage, thereby contributing to the overall performance and responsiveness of the doctor appointment system.

User Authentication Using JWT:

Description: This method introduces the use of user authentication using JSON Web Tokens (JWT). This includes token creation, verification, and secure transmission of credentials.

Purpose: To demonstrate the effectiveness of the authentication system, to ensure that users use appropriate security measures and to increase trust in the platform.

React responsive user interface:

Description: Provides a comprehensive overview of the design and implementation process of the user interface using React. It illustrates the application of object-oriented models, the utilization of virtual DOM, and the efficient integration of state management. Additionally, detailed information about API specifications is presented.

Goal: Striving for transparency in user interface design, emphasizing a user-centric approach that prioritizes interactivity and clarity on travel websites.

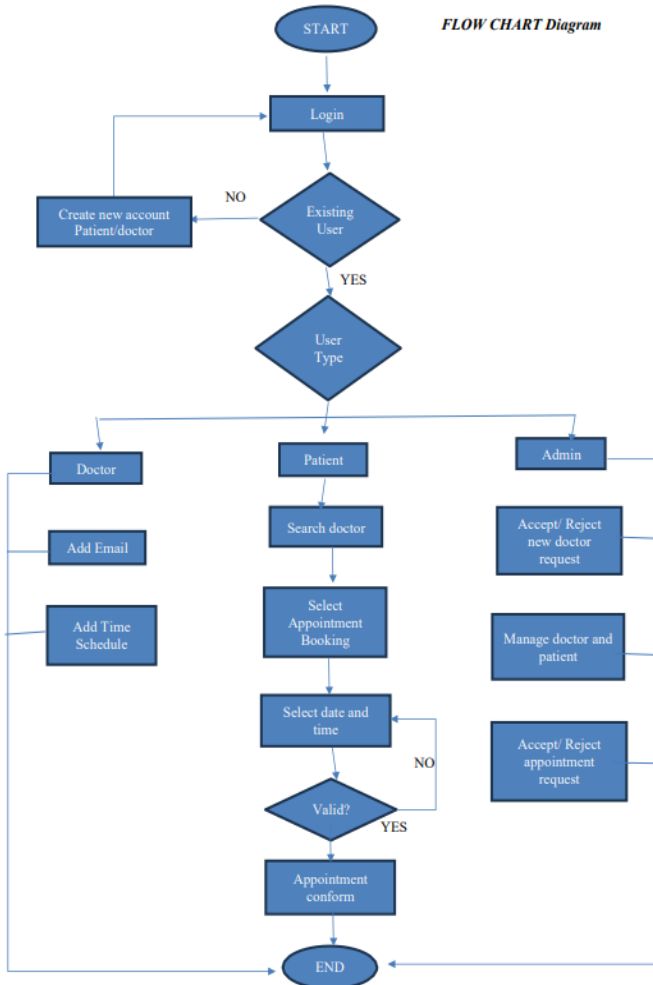
IV. SYSTEM STUDY AND ANALYSIS

Existing System:

- All tasks are handled manually.
- In the manual appointment booking system, patients need to visit the healthcare facility.
- Inquire about available appointments, book a slot, make the payment, and then collect a receipt.
- Challenging to keep track of patient details related to appointments and payments.
- Record-keeping is done in a register.
- Appointment details are logged in a notebook.
- Advertising is done through local newspapers or in the local market.
- Limited availability of appointment scheduling for a specific area or individual.

Proposed System:

- Developing a web-based application for our institution.
- Offering a search feature for customers.
- Generating various types of reports.
- Providing an online appointment booking system with an online payment option for patients.
- Furnishing package details for patients.
- Allowing customers to cancel appointments with a deduction of 15% from the refunded amount.



MongoDB Atlas:

Description: MongoDB Atlas is a cloud-based database service provided by MongoDB. It is a fully managed and scalable NoSQL database solution that allows developers to store and retrieve data in flexible, JSON-like documents.

Key Features:

- Automated scaling of database clusters.
- Ensuring secure data storage through encryption and authentication measures.
- Continuous real-time monitoring and analytics for optimizing performance.

Tech Stack Role: MongoDB Atlas functions as the central data repository for the Doctor Appointment System, storing crucial information like patient profiles, medical practitioners' details, appointment schedules, and patient reviews.

Express.js:

Description: Express.js is a web application framework for Node.js. It simplifies the process of building robust and scalable web applications by providing a set of features and tools for routing, middleware, and HTTP request handling.

Key Features:

- Middleware support for handling HTTP requests.
- Routing capabilities for defining endpoints.
- Integration with various template engines for dynamic content generation.

Tech Stack Role: In the context of the Doctor Appointment System, Express.js is employed to build the backend server. It manages routing, executes business logic, and facilitates communication with the MongoDB database.

React:

Description: React is a JavaScript library for building user interfaces. [] It allows developers to create reusable UI components that efficiently update and render when the underlying data changes, providing a more dynamic and responsive user experience.

Key Features:

- Component-based architecture for modular development.
- Virtual DOM for efficient rendering and updates.
- Declarative syntax for defining UI components.

Tech Stack Role: React is utilized to construct the frontend of the Medicare- Doctor Appointment System, crafting a dynamic and interactive user interface. This interface facilitates tasks such as scheduling appointments, accessing medical information, and managing user interactions, ensuring a seamless and user-friendly experience.

Node.js:

Description: Node.js is a JavaScript runtime environment that allows developers to execute server-side JavaScript. It is designed to be lightweight, efficient, and scalable, making it suitable for building server-side applications.

Key Features:

- Non-blocking, event-driven architecture.
- Package management with npm (Node Package Manager).
- Extensive ecosystem of libraries and modules.

Tech Stack Role: Within the Doctor Appointment System, Node.js functions as the runtime environment for the backend. It facilitates server-side scripting and facilitates communication between the frontend and the MongoDB database, ensuring the system's smooth operation.

Postman:

Description: Postman is a popular API development and testing tool that simplifies the process of building, testing, and documenting APIs. It provides a user-friendly interface for sending HTTP requests and inspecting responses.

Key Features:

- API request creation and testing.
- Automated testing and scripting.
- Collaboration and API documentation.

Tech Stack Role: Postman is employed in the development and testing phases of the Doctor Appointment System to validate API endpoints, assess data interactions, and verify the correct operation of backend services.

Context API:

Description: Context API is a feature in React that allows the passing of data through the component tree without explicitly passing props at every level. It simplifies state management in large

applications by creating a global state accessible to all components.

Key Features:

- Centralized state management.
- Avoids prop drilling.
- Facilitates easy sharing of state between components.

Tech Stack Role: In the Doctor Appointment System's frontend, Context API is utilized to oversee global state management. This ensures a streamlined and organized sharing of data among various React components, encompassing aspects like user authentication status, selected medical practitioners, and appointment details.

V. CONCLUSION & FUTURE SCOPE

The proposed online appointment system has been successfully implemented using Android Studio for application development, while the website is crafted using HTML and PHP. The project's tasks are categorized into distinct modules, with data exchange facilitated through APIs between the website and the Android application. The introduced system is not only efficient but also boasts a user-friendly interface.

Looking ahead, future enhancements will involve incorporating admin and doctor modules within the Android application. This expansion aims to allow doctors to register on the application and perform various tasks seamlessly. Furthermore, administrators will have the option to manage patient and doctor details directly through the app, eliminating the need for exclusive reliance on the website.

To discourage misuse and ensure commitment from users, a potential addition to the system is the introduction of a payment mechanism. Charging a nominal fee during the appointment booking process can help filter out users with no genuine intention of seeking medical attention.

Continuing with future directions, there is a plan to refine the patient's module further. This includes

implementing features such as appointment reminders and the ability to save appointment dates directly to the user's calendar, enhancing the overall user experience.

VI. REFERENCES

1. Petrikoglou A. and H. Kaskalis, T. Full Stack Web Development Teaching: Current

Status and a New Proposal. DOI: 10.5220/0008066202180225 In Proceedings of the

15th International Conference on Web Information Systems and Technologies

(WEBIST 2019), ISBN: 978-989-758-386-5. Available at

<https://pdfs.semanticscholar.org/a0e8/b48fd6c64bd9b3fdc7cbdbc3f5f672d7ce14.pdf>

2. Bafna S. A., Dutonde P. D., Mamidwar S. S., Korvate M. S., & Shirbhare P. D.

(2022). Review on Study and Usage of MERN Stack for Web Development.

International Journal of Research in Advanced Science and Engineering

Technology, 10(2), ISSN: 2321-9653. IC Value: 45.98. SJ Impact Factor: 7.538.

Available at <https://www.ijraset.com/best-journal/study-and-usage-of-mern-stackfor-web-development>.

3. React.dev Quick Start – React. Available at: <https://react.dev/learn> [Online;

Released on 29 May 2013].

4. Rimal A., Metropolia University of Applied Sciences. Developing a Web

Application on NodeJS and MongoDB using ES6 and Beyond. Available At

https://www.theseus.fi/bitstream/handle/10024/159951/Rimal_Aashis.pdf

[Released on 25 January 2019]

5. MDN Web Docs About JavaScript. Available at:

<https://developer.mozilla.org/en-US/docs/Web/JavaScript> [Online; Released

on 2005].

6. Node.js Introduction to Node.js. Available at: <https://nodejs.dev/en/learn>

[Online; Released on 27 May, 2009].

7. Bawane M., Gawande I., Joshi V., Nikam R., & Bachwani S. A. (2022). A Review

on Technologies used in MERN stack. International Journal of Research in

Advanced Science and Engineering Technology, 10(1), ISSN: 2321-9653. IC

Value: 45.98. SJ Impact Factor: 7.538. Available at

<https://www.ijraset.com/best-journal/a-review-on-technologies-used-in-mernstack>.

8. Razorpay Docs Payment Gateway. Available at:

<https://razorpay.com/docs/partners/aggregators/partner-auth/payment-gateway>

[Online; Released on 2014]

9. Rappin, Noel (2021). Modern CSS with Tailwind flexible styling without the fuss.

31

Raleigh. ISBN 978-1-68050-857-4. OCLC 1277046918.

10. Gerchev, Ivaylo (2022). Tailwind CSS. Sebastopol: O'Reilly Media. ISBN 978-1-

0981-4099-1. OCLC 1314257318.

11. Chourasia, Rawnak (2023-03-08). "Convert Class Component to Function(Arrow)

Component - React". Code Part Time. Retrieved 2023-08-15.

12. "Consider re-licensing to AL v2.0, as RocksDB has just done". Github. Retrieved

2017-08-18.

13. "Node.js Foundation Combines Node.js and io.js Into Single Codebase in New

Release". nodejs.org. 14 September 2015. Retrieved 28 January 2016.

14. Foundation, Node js. "Installing Node.js via package manager". Node.js. Retrieved 19

May 2019.

15. Connolly, Daniel (18 August 1992). "document type declaration subset for Hyper

Text Markup Language as defined by the World Wide Web project". CERN.

Retrieved 24 October 2010.

16. Berners-Lee, Tim; Connolly, Daniel (June 1993). "Hypertext Markup Language

(HTML) Internet-Draft version 1.2". IETF IIR Working Group. Retrieved 18

September 2010.

17. "Node.js Foundation to shepherd Express Web framework". Infoworld. IDG. 10

February 2016. Retrieved 11 February 2016.

18. "Mean.io: The Friendly & Fun Javascript Fullstack for your next web application".

Archived from the original on 13 June 2019. Retrieved 15 July 2019.

19. Microsoft Internet Explorer 3.0 Beta Now Available". microsoft.com. Microsoft.

May 29, 1996. Archived from the original on November 24, 2020. Retrieved July

16, 2016.

20. "Usage statistics of JavaScript libraries for websites". w3techs.com. Archived

from the original on 2012-05-26. Retrieved 2021-04-09