Lifelink: A Modern Full Stack Blood Donation Management System using MERN Architecture

V. Mageswari[1], N. Mohamed Magaroos[2]
[1]*MCA, Assistant Professor, Master of Computer Applications, Moolakulam, Oulgaret Municipality, Puducherry – 605010.*
[2]*MCA, Christ College of Engineering and Technology, Moolakulam, Oulgaret Municipality, Puducherry – 605010.*

***Abstract*—Among all the healthcare-related services, blood donation stands at the pinnacle, as it directly works for saving human lives in cases of emergencies such as accidents, surgeries, and childbirth complications, along with chronic illnesses. Although there are blood banks and hospitals, timely accessibility of suitable blood donors remains a big problem owing to a lack of real-time information on donors, inefficient coordination, and a manual record-keeping approach. These drawbacks result in delays that sometimes prove fatal. LIFELINK is a full-stack, modern web application for the management of blood donations, developed using the MERN stack: MongoDB, Express.js, React.js, and Node.js. The idea is to connect the dots digitally between blood donors and recipients through this networking hub. The donors will be able to register their blood details much in advance, while recipients can place a request for blood in emergency situations. The portal automatically matches donors according to blood group compatibility, location, and other specified criteria and notifies the fit ones immediately.**

Therefore, through automation in donor–recipient matching and secure data handling practices, much response time is reduced and efficiency enhanced in the management of blood donations. This project illustrates that modern web technologies can be gainfully employed not only to create various feasible web applications but also to reinforce healthcare support systems by providing a reliable solution related to the needs of blood donation in emergency situations.

***Index Terms*—Blood Donation, Donor Matching, Web App, Emergency HealthCare, donor–recipient, rapid communication, chronic patients, server side, scalability**

I. INTRODUCTION

Blood is not replaceable and is widely used in treatments for patients all over the world. Hospitals and other medical facilities depend on the blood supply to handle emergency and surgical cases and provide care for chronic patients [1,2]. Many healthcare systems continue to face the problem of obtaining blood supplies at the right time, particularly in emergencies. Most traditional blood donation-based systems are characteristically manual in data storage, phoning, and physical co-ordination, which are usually slow and unreliable [3,4].

The rapid growth in Internet technologies and Web-based applications has currently opened an excellent opportunity to enhance the efficiency of blood donation systems through some digital solutions. Web-based platforms could provide real-time access to donor data, immediate matching, and rapid communication between donors and recipients [5,6].

LIFELINK will be developed to respond to these challenges through the provision of a centralized, secure, and user-friendly web application. The web application gives an opportunity for donors to register their details once, which will then always be available when some recipient raises a request for blood [7,8]. By effectively integrating modem full-stack development technologies and intelligent matching logic, LIFELINK is envisioned to simplify the blood donation process and provide improved capabilities for responding to emergencies more effectively [9,10].

II. BACKGROUND THEORY

The background theory of LIFELINK is from a centralized digital network of blood donation wherein all information related to donors and recipients is stored, processed, and controlled through a secure server [1,2]. The backend acts as the core engine of this entire system, which will handle authentication, data validation, donor matching, and handling of requests [11,12].

User-based interaction with the frontend interface forwards all incoming registration, log-in requests, or blood requests to the backend via RESTful APIs. The backend validates the authenticity of the user, processes the request logic, and interacts with the database for retrieving and storing data accordingly. Donor information such as blood group, city, and last donation date is stored in structured collections, making it very easy to query and filter during matching operations.

In the backend, rule-based logic is implemented to match recipient requirements with donor data. Rules for blood group compatibility ensure medical correctness [15], whereas location-based filtering reduces response time by prioritizing nearby donors. Eligibility checks prevent unsafe donations by using minimum time intervals between donations [16].

The backend provides data security and consistency by handling all the important operations on the server side, scalability, too [13,14]. This further allows the system to work effectively when there are several concurrent accesses to the platform.

III. OBJECTIVE OF THE PROJECT

The main focus of the LIFELINK project is to develop an efficient, web-based modern platform for easy management of blood donation, thereby reducing response time in critical conditions [1,2]. The system ensures a safe digital solution for donors and recipients by avoiding the need for any manual coordination [3,4].

Another important objective is the storage and processing of sensitive user data securely by using authentication and encryption techniques. It also concentrates on the donor–recipient matching automatically to reduce human effort, thereby avoiding delays. In summary, the aim is to show how full-stack web technologies can be applied to solve real-world healthcare challenges and support life-saving activities.

IV. SYSTEM ARCHITECTURE

It considers using a three-tier client-server architecture in designing the system architecture of LIFELINK. It separates the application into distinct layers: the presentation layer (frontend), application layer (backend), and data layer, which is the database itself [9,10]. The use of such architecture enhances modularity, scalability, and security while it promotes ease in interaction among the system components [11,12].

The key components of the architecture used are as follows:

- Frontend Layer: React.js mainly acts as a presentation layer where user interactions, such as inputting values and dynamic rendering of pages, take place.
- Backend layer: Consists of Node.js, version 12, and Express.js, version 10, which handles the business logic processing, donor matching, authentication, and notification services.
- Database Layer: The profile information of donors and requests of recipients, authentication data, and history of donations are stored in MongoDB [9].
- Security: JWT [13], bcrypt [13] grant security in authenticating and password encryption.

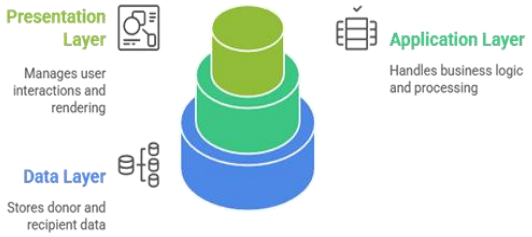

Figure 1: LIFELINK System Architecture

V. MODULE DESCRIPTION

5.1 User Authentication Module

Access to the system is controlled by the User Authentication Module. In the case of user registration, it validates the credentials and stores them securely using encrypted passwords. During login, it confirms

the credentials of the users and issues a JSON Web Token for an authenticated session. Ensuring only appropriate users could have access to the features of the platform, this module protects sensitive information from unauthorized access. Handles secure registration and login with JWT-based session management.

5.2 Donor Management Module

Under the Donor Management Module, blood donors can enroll and maintain their profile. This module captures vital information about the donor: blood group, city, contact details, and the date of last donation. The records are captured in the database and frequently updated according to the outcome of the interaction between the user and the system. The module also verifies the eligibility of donors under medical specifications to validate donor safety and compliancy. Allows donors to register, update profiles, and verify eligibility [5,6].

5.3 Recipient Request Module

The Recipient Request Module helps users in need of blood place requests through the platform. A recipient can define the needed blood group, hospital address, and location. When submitted, it gets stored into the database and then relayed to the matching engine for processing. By this module, requests are clearly specified and processed in no time. Allows recipients to place requests for blood, which is thereafter handled by the matching engine [7,8].

5.4 Donor-Recipient Matching Module

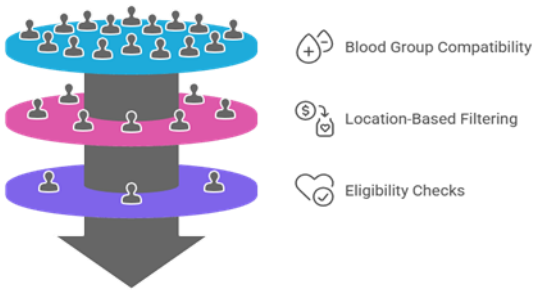

Figure 2: Donor-Recipient Matching Process

This module is the core functionality of LIFELINK. It compares recipient requirements with donor records stored in the database. The module uses the rules of blood group compatibility, location-based filtering, and eligibility checks to identify suitable donors. Such automation considerably reduces the time that would have been used to identify matching donors. Compares recipient requirements with donor data using blood group compatibility, location, and eligibility checks [15,16].

5.5 Notification and Acceptance Module

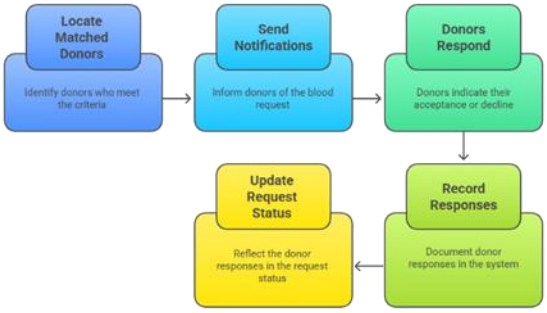

Figure 3: Donor Notification and Acceptance Process

When the matched donors are located, the Notification Module sends notifications to those donors, informing them of the request for blood. Donors may choose to accept or decline the request depending upon their availability. The response of the donors is noted in the Acceptance Module and the status of the request is updated, allowing the recipient to move further ahead as soon as possible with confirmed donors. Notify suitable donors and record acceptance, updating the request status [5,6].

5.6 Module Admin Management

Admin Management Module: This module provides administrative control of the platform, allowing admin users to monitor registrations and manage blood requests. It ensures data integrity and keeps the smooth operation of the system without any misuse. This module provides administrative oversight over the operation of the system, including system activity monitoring and data integrity.

VI. PERFORMANCE ANALYSIS

The performance of the LIFELINK system was evaluated using simulated user loads to analyze its responsiveness and scalability. The response time increased gradually as the number of concurrent users

grew, indicating that the system efficiently handles multiple requests without significant performance degradation. This is mainly due to the asynchronous processing capability of the Node.js backend and optimized RESTful API communication.

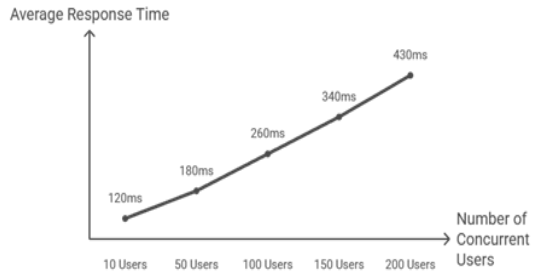

Figure 4: Response Time vs Number of Concurrent Users

The donor–recipient matching process was also analyzed by measuring matching time against an increasing number of donor records. The results show that the matching algorithm performs efficiently even with larger datasets, supported by MongoDB indexing and rule-based filtering. Overall, the system demonstrates stable performance and is suitable for real-time blood donation management during emergency situations.

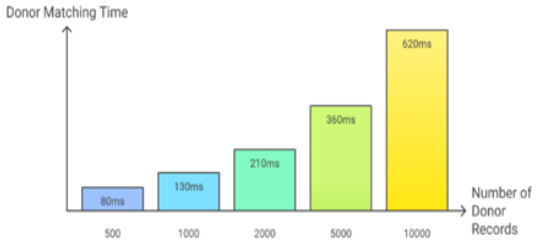

Figure 5: Donor Notification and Acceptance Process

VII. FUTURE ENHANCEMENTS

Although LIFELINK effectively fulfills the core requirements of a blood donation management system, several enhancements can be incorporated in future versions to further improve its functionality, scalability, and real-world usability.

- Real-Time Location Tracking: Integration with GPS services can help identify the nearest available donors more accurately, reducing response time further.
- Advanced Notification System: Incorporating SMS and WhatsApp-based alerts along with email notifications will ensure wider and faster communication with donors.
- AI-Based Donor Recommendation: Machine learning algorithms can be used to prioritize donors based on availability history, response rate, and distance.
- Analytics and Reporting Module: Advanced dashboards for administrators can help analyze donation trends, emergency patterns, and system usage statistics.
- Enhanced Security Features: Implementation of multi-factor authentication and role-based access control can further strengthen data privacy and security.

VIII. CONCLUSION

The design of the LIFELINK system effectively illustrates how contemporary web technologies can be put to work in offering solutions to some of the most critical healthcare challenges facing humanity today. References [1, 2] provide further details. By offering a unified, secure, and automated platform, LIFELINK enhances the coordination between blood donors and recipients by ensuring timely response during distress. The donor–recipient matching algorithm of the system, coupled with mutual acceptance, shares personal information in a secure manner and only when both consent [12,13]. It cuts down delays in finding compatible donors, which in turn maintains privacy and instills trust among people [7,8]. Minimizing manual efforts for healthcare providers, decreasing errors of traditional systems in blood donation management, and overall efficiency is improved on the platform [3,4]. LIFELINK showcases the practical benefits of full-stack development in building real-world solutions that save lives [9,10].

REFERENCES

[1] N. Sahu, S. Verma, P. Nial, and S. S. Tripathi, MERN Stack Based Blood Bank App, Int. J. Res. Publication and Reviews, vol. 6, no. 4, Apr. 2025. IJRPR

[2] A. Hannan et al., “LIFESAVER: Blood Donation Management System Web App Using MERN Stack,” Int. Res. J.

Modernization Eng. Technol. Sci., vol. 6, no. 1, Jan. 2024. IRJMETS

[3] P. Giradkar, N. Meshram, A. Pandilwar, and A. Penliwar, "MERN Stack Blood Bank Website," Int. J. Adv. Res. Sci. Commun. Technol., vol 3, no. 1, Dec 2023. IJAR Scientific Journal

[4] G. Manika, V. S. Reddy and K. S. Vamshi, "Development and Implementation of a Web-Based Blood Bank Management System for Efficient Blood Donation and Distribution," IJRASET, 2023. IJRASET

[5] N. M. S. Azhar and F. A. H. Ali, "Web–based Development: Smart Blood Bank Management System," Appl. Inf. Technol. Comp. Sci., vol. 3, no. 2, 2022. UTHM Publisher

[6] Web-Based Blood Bank Management System, Vedanti V. Deshpande, S. Kumar, and S. Chaple, IJARIIT, Volume 6, Issue 2, Apr 2020. IJARIIT

[7] Design and Implementation of a Scalable, Real-Time, Web-Based Blood Bank Management System, Ibukun Eweoya et al., FUOYE J. Eng. Technol., vol. 10, no. 3, 2025. FUOYE JET Copyright: This is an open-access article distributed under the Creative Commons Attribution License, which allows unrestricted use, distribution, and reproduction in any medium provided the original work and source are appropriately credited.

[8] "A Review on Technologies Used in MERN Stack," Mohanish Bawane et al., IJRASET, 2022. IJRASET

[9] MongoDB, MongoDB Manual, MongoDB, Inc., Available: https://docs.mongodb.com

[10] Express.js, Express - Node.js Web Framework, Express, Available: https://expressjs.com

[11] React.js, React - A JavaScript Library for Building User Interfaces, Meta, Available: https://reactjs.org

[12] Node.js, Node.js Documentation, OpenJS Foundation, Available: https://nodejs.org

[13] W3Schools, "JWT Authentication Tutorial"

[14] R. T. Fielding, Architectural Styles and the Design of Network-based Software Architectures, Ph.D. dissertation, Univ. of California, Irvine, 2000.

[15] S. Choudhury and P. Gupta, "A Review on Blood Donation Management Systems and Emergency Response Applications," J. Medical Systems, vol. 44, no. 12, 2020.

[16] S. Islam, M. Rahman, and A. Chowdhury, "Automated Blood Donor Matching System Using Web Technologies," J. Health Informatics in Dev. Countries, vol. 14, no. 1, 2020.

[17] F. Li, M. H. Lee and P. Wang, "Security Practices for Modern Web Applications Using Node.js," IEEE Access, vol. 9, pp. 12345–12356, 2021.

[18] Kaur M., N. Nazir, N. Kaur et al., "A Web–Based Blood Bank System for Managing Records of Donors and Recipients," in Proc. Int. Conf. Computational Intelligence Sustainable Eng. Solut. (CISES), 2022, pp. 459–464. HBRP Publication

[19] Anuj Patil, "Blood Bank Management System," IJRASET, DOI:10.22214/ijraset.2022.4629

[20] K. Kaur, A. K. Rai, N. Raj et al., "Blood Bank Management System with Emergency Blood Locator," IJRASET, DOI: 10.22214/ijraset.2023.52877.