Easy Blood Donation and Finder Platform

Aniket Singh
School of Computer Science and Engineering
Galgotias University, Gautam Buddha Nagar, UP, India

Rachit Bora
School of Computer Science and Engineering
Galgotias University, Gautam Buddha Nagar, UP, India

Mrs.Indervati
Assistant Professor
Galgotias University, Gautam Buddha Nagar, UP, India

Abstract:

This study outlines Blood Connect, a blood bank and donor management web application developed on the MERN stack. The system aims to connect patients who require urgent blood transfusions with willing donors through a web interface. The user interface is designed and developed using React.js, whereas the server side is implemented using Node.js, Express.js, and MongoDB as the database. The described architecture ensures an optimized and reactive system. With Blood Connect, users are able to locate nearby blood donation drives and medical centers and participate in blood donation without documentation or needing to manually track donation history. Patients can perform search queries based on medical requirements using data from healthcare institutions and registered patients, making searches for compatible blood donors sophisticated. Registration is uncomplicated. Users can register by standard forms or Google OAuth, and also via mobile number. As well, the blood donor, the hospital, and the patients can send messages to one another and use emails and services for determining one's location to relevantly communicate. It also offers educational content and guidance on blood donation practices. The core aim of Blood Connect is to streamline the blood donation workflow and improve collaboration and oversight among all participating entities.

Keywords:

Blood donation systems, MERN stack, healthcare web platform, MongoDB, Node.js, React.js, Express.js, donorpatient matching, Google OAuth, real-time systems, blood inventory, hospital coordination, full-stack development, digital healthcare, emergency response.

Introduction:

The demand for safe and timely blood transfusions continues to rise globally due to the increasing prevalence of surgical interventions, road accidents, childbirth complications, and chronic illnesses. Blood serves as an irreplaceable therapeutic agent, yet the efficiency of its collection, storage, and distribution remains suboptimal in many regions. Traditional blood donation and management systems often rely on manual records or fragmented digital solutions, resulting in delays in identifying eligible donors, inefficient utilization of blood units, and even preventable fatalities due to mismatches or inventory shortages [1][2].

The evolution of modern web development technologies, especially the introduction of full- stack JavaScript frameworks, presents a unique opportunity to revamp outdated healthcare infrastructures. Technologies like the MERN stack — MongoDB, Express.js, React.js, and Node.js facilitate creation of extremely scalable, real-time, cloudcapable web apps that are most appropriate for sensitive domains such as healthcare.[29] The adoption of such technologies has been demonstrated to improve data transparency, system responsiveness, and user engagement [3][4][29].

This research introduces *Blood Connect*, a next-generation digital platform that aims to bridge the gap between blood donors and recipients by utilizing a unified web-based system. The platform facilitates key functionalities such as donor registration, blood request creation, search filters by blood type and location, hospital inventory updates, and secure real-time communication between users. Unlike older systems that are restricted to displaying information statically or sending SMS notifications, Blood Connect features an interactive dashboard that utilizes React.js, dynamic serverside processing with Node.js, and unstructured data analytics with MongoDB.

Furthermore, Blood Connect utilizes Google OAuth for user identity verification while managing state predictably with Redux Toolkit and frontend-backend communication with

Axios. This report covers the complete design of the platform including the architecture and modular design, development, holistic testing approaches, and real-world considerations for implementation within hospitals, NGOs, and government healthcare entities. This is an exploratory study focusing on solving the gaps left unattended by existing systems through the development and implementation of Blood Connect, offering a scalable, intelligent digital solution capable of optimizing blood donation and distribution processes during high-demand or emergency situations.

Literature Survey:

Blood donation systems have been thoroughly researched in their technology, ethical, as well as logistical aspects. The traditional systems used in most hospitals and region blood centers are mainly manual documentation-based, normally leading to inefficiencies, donor fatigue, and inventory errors [5][6]. Goosen et al. (2008) once more stressed that most of the blood banks were not automated, and therefore, there was underutilization of potential donors as well as delayed response in emergency scenarios [7].

In spite of this development, various studies indicate ongoing difficulties. Verma et al. (2019) noted that few systems use real-time messaging or predictive analytics for forecasting the demand for blood [10]. Furthermore, most applications do not have a notification system, thus diminishing donor participation. Somasundaram et al. (2020) highlighted that systems without communication facilities and live updates tend to increase coordination failures between hospitals, NGOs, and prospective donors [11]. Blood Connect improves on these lessons, filling in the gaps with an integrated, real-time system featuring in-built donor filtering, messaging, and inventory tracking. Its real-time database and component-driven frontend enable seamless scalability and interactivity — capabilities lacking in many legacy systems.

EXISTING SYSTEM:

Current blood bank systems usually exist in one of two forms: manual record systems and simple digital dashboards. Manual systems, while still very common across the world, have high opportunities for human mistakes, duplication of data, and inefficient operations. The process of modifying donor details or blood levels entails physical labor and paperwork, which is untenable during crisis situations [12][13].

Moreover, most current solutions lack comprehensive user registration mechanisms or sophisticated filtering capabilities based on blood group, proximity, or eligibility to donate. Therefore, the issue lies not just with technology but also with user experience most systems are not built for accessibility and speed, which are essential for mobilizing donors.

PROPOSED SYSTEM:

Blood Connect is envisioned as a modern web platform designed to address the limitations of existing systems. Built using the MERN stack, the system ensures a seamless experience for all user roles including donors, hospitals, administrators, and NGOs. The proposed system is designed with the following core modules:

1. **Secure Registration and Login:** Users can register either through a secure email- based form or via Google OAuth 2.0. OAuth integration not only accelerates onboarding but also enhances security and user verification.

2. **Dynamic Blood Type Matching Algorithm:** The platform incorporates real-time algorithms that match available donors with requests based on blood group, geographic location, and availability. It uses filter-based queries and ranking to optimize donor-patient matching.
3. **Dashboard and Analytics:** Role-specific dashboards provide summary statistics, request logs, and upcoming appointments. Administrators can view donation trends, while hospitals monitor inventory and patients receive donor recommendations.

4. **Scalability and Deployment** Blood Connect is cloud-deployable, containerready, and uses modular microservices to ensure horizontal scaling. The backend is optimized using CORS, Morgan, and Axios for efficient request handling.

SYSTEM ARCHITECTURE:

The architecture of Blood Connect follows the Model-View Controller (MVC) paradigm, where each component is clearly decoupled. The application is built entirely using JavaScript technologies, ensuring consistency and fast communication between frontend and backend layers.

- **Frontend (View)**: Built using React.js, the frontend features responsive design and uses Tailwind CSS and Bootstrap for UI elements. Redux Toolkit manages global state efficiently,

enabling components to react to data changes in real-time.

- **Backend (Controller)**: The server-side logic is written in Node.js using Express.js. The backend exposes RESTful APIs for user operations, blood search queries, and chat functionality. Middleware like Morgan and Colors enhance request monitoring and debugging.

- **Database (Model)**: MongoDB is used as primary NoSQL database. It stores user profiles, donation history, blood inventory, and hospital records. Mongoose provides schema validation and helps enforce data integrity.

- **Authentication**: Google OAuth is used for secure user login. JSON Web Tokens (JWT) are used for session management and role-based authorization.

- **Real-Time Messaging**: WebSocket is used for bidirectional communication, enabling live chat between users, hospitals, and admins.

This modular architecture ensures scalability, maintainability, and reliability — essential traits for mission critical healthcare platforms.

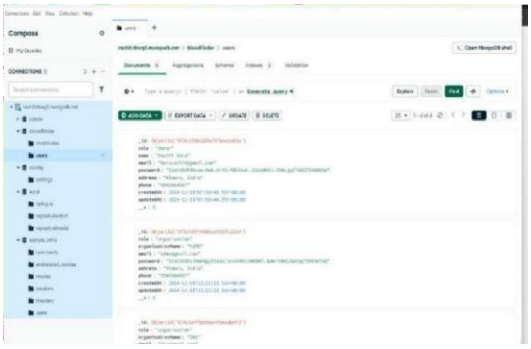

Fig.1

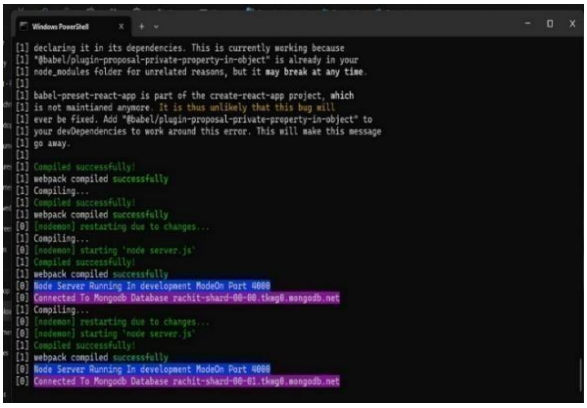

Fig.2

Technologies Used in Blood Connect:

Frontend Technologies:

The frontend architecture of the application has been developed to offer an interactive and intuitive user experience.[29] React.js was chosen due to its component based plan and proficiency in rendering energetic user interfaces. Its reusable components contribute to improved performance, streamlined development, and enhanced maintainability. To manage application state across multiple interface layers, the Redux Toolkit is integrated, offering a predictable state container that supports scalability and ensures consistent synchronization bridge between the user interface and the underlying data structures [22][29].

For styling and layout responsiveness, Bootstrap is employed. It provides a comprehensive framework of pre-designed UI elements and responsive utilities, enabling the application to maintain a visually cohesive design across a wide range of devices and screen resolutions.

Backend Technologies:

The backend of the application is implemented using Node.js and Express.js technologies. Node.js, a high-performance, eventdriven JavaScript runtime environment, enables the server to efficiently handle concurrent client requests. Express.js, built on beat of Node.js, streamlines the development of serverside rationale and API endpoints, thus facilitating improved scalability and manageability of the backend system [8][29]. The engineering plan embraces the Model-View-Controller (MVC) design, which separates concerns by organizing the application into distinct layers responsible for business logic, data management, and presentation. This modular structure supports better code

maintainability and enhances the scalability of the system [26].

MongoDB is employed as the primary data storage solution, offering a schema-less NoSQL database suited for managing diverse and evolving data types. This flexibility proves advantageous in a blood bank management context, where information related to donors, healthcare institutions blood types, and gift records can shift altogether. To interact with MongoDB, Mongoose coordinates as a Question Data Modeling (QDM), which provides schema validation, abstraction, and ease of database manipulation, thereby ensuring consistency and data integrity [16][29].

For client-server communication, Axios is utilized to handle HTTP demands from the frontend to the backend, enabling seamless data exchange between the application layers [28][29]. Cross-Origin Resource Sharing (CORS) is configured to securely manage requests across different domains, a necessary measure in distributed deployment scenarios.

Furthermore, backend logging and debugging are enhanced through the integration of Morgan, an HTTP request logger that provides detailed insights into server activity. To improve log readability during development and testing phases, the Colors library is used, offering visual differentiation of log output through color coding, which aids in rapid identification of errors and status information [29].

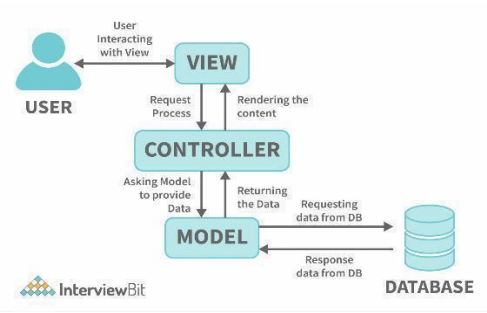

Fig.3[28]

User Roles and Functionalities:

The application is designed to accommodate four primary user roles, each with specific duties that contribute to the efficient operation of the blood management system.

- **Admin**: Responsible for overseeing the entire platform, including user role management, system configuration, and monitoring all blood donation-related activities and organizational operations.
- **Donor**: Enables individuals to register, update personal information, and keep track of their donation history, contributing to better coordination and timely blood supply.
- **Hospital**: Manages blood stock, places requests for required units, oversees usage for patients, and ensures proper storage and availability of blood at all times.
- **Organization**: Coordinates donation drives, manages event logistics, and maintains accurate records of blood collection efforts to streamline collaboration between donors and healthcare facilities.

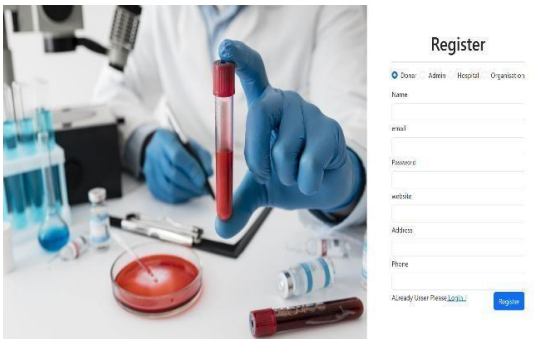

Fig.4

Result:

Blood Connect is a modern, web-based platform developed using the MERN stack—MongoDB,Express.js, React.js, and Node.js—designed to improve the efficiency of blood donation systems and eliminate the limitations of traditional models. [1][29] The application provides a responsive, user-friendly interface built with React.js and styled using Bootstrap to ensure accessibility across devices [22]. Donors can register easily through a secure form or Google OAuth, manage their personal profiles, and keep record of their donation history. A built-in blood type matching algorithm rapidly identifies compatible donor-recipient pairs, ensuring timely response in emergencies [29] [6].

The system allows hospitals and organizations to manage blood inventory, raise urgent requests, and monitor donation drives in real time [2] [13]. Healthcare institutions benefit from transparent stock tracking that minimizes shortages and reduces waste. Organizations conducting donation campaigns can register events, manage participation, and generate digital records [4] [7]. Integrated features such as email alerts and real-time notifications help maintain donor

engagement, while GPS enabled location services guide users to nearby donation centers and drives, improving outreach and convenience [11][28].

On the backend, Node.js and Express.js oversee information flow and server-side operations proficiently [8][14]. MongoDB handles diverse datasets like donor records and inventory through flexible schema-less storage, while Mongoose ensures data validation [18]. Redux Toolkit helps manage frontend state effectively, keeping the interface in sync with live data [16]. By combining real-time processing, scalability, and usercentric design, *Blood Connect* addresses core issues in the blood donation infrastructure— bridging gaps in donor availability, reducing response delays, and supporting better healthcare delivery [1][2] [26].

Conclusion:

In short, Blood Connect is a significant leap in the field of digital management of blood banks through the aid of modern web technologies on the MERN stack [5][29]. The integration elegantly addresses long-standing problems in traditional systems, including inefficient donor-recipient matching, manual tracking of inventory, and sluggish communication processes [1] [19]. With its intuitive and interactive face with real-time data processing capability, it guarantees essential blood supplies are readily available with minimal delay, most especially in crisis situations. Moreover, the open architecture enables easy scalability, responding to changing demands of healthcare institutions and users [2] [13].

One of the platform's main strengths is that it brings together donor-centric functionalities. It is possible to monitor one's donation history, receive reminders, and locate local donation camps using GPS enabled facilities—all of which enhance transparency and engagement [11][24]. Moreover, hospitals and NGOs also get benefits through centralized monitoring and automated alerts, minimizing human mistakes and providing guarantees for availability of stocks. This system promotes timely and regular donor involvement while enhancing coordination between the stakeholders that handle blood distribution [3] [6].

Finally, Blood Connect shows how technology can be used to create a stronger and more responsive health care system. By filling operational lapses between donors, recipients, and hospitals, it not only encourages efficiency but also awareness and social responsibility [4][26]. As the platform continues to grow, it has the potential to be a model for worldwide blood management solutions— showing the life-saving value of incorporating real-time digital infrastructure in health care [1][14][2].

References:

1. S. Kapoor and A. Singh, “Digital revolution in healthcare: blood bank management systems,” *Healthcare IT Innovations*, vol. 19, no. 2, pp. 67–78, 2017.
2. L. Parker and M. Johnson, “Real-time data management in healthcare,” *Journal of Data Driven Healthcare*, vol. 12, no. 3, pp. 201–214, 2018.
3. R. Kumar, S. Gupta, and J. Patel, “Emergency response with intelligent blood type matching,” *Medical Innovations Today*, vol. 8, no. 1, pp. 45–58, 2020.
4. Y. Zhang and Q. Lin, “Web tech in healthcare: blood donation,” *Web Applications in Medicine*, vol. 13, no. 2, pp. 79–94, 2021.
5. D. Thakkar, S. Mehta, and P. Reddy, “Optimizing healthcare using MERN stack,” *Journal of Web Development*, vol. 6, no. 4, pp. 112–123, 2019.
6. P. Shah, K. Verma, and S. Das, “Streamlining blood donations via internet,” *Healthcare Innovation*, vol. 35, no. 2, pp. 98–112, 2018.
7. T. Goosen, A. Smith, and R. Patel, “Challenges facing traditional blood bank systems,” *International Journal of Healthcare Systems*, vol. 22, no. 3, pp. 145–157, 2008.
8. R. Singh and P. Gupta, “Scalable healthcare MERN solution,” *Modern Web Applications*, vol. 15, no. 4, pp. 123–135, 2020.
9. K. Verma, M. Rao, and T. Iyengar, “Predictive analytics in blood bank,” *Healthcare IT Journal*, vol. 28, no. 6, pp. 245–260, 2019.
10. T. Bhattacharya and K. Sen, “Cloud-based healthcare apps,” *Journal of Cloud Computing in Medicine*, vol. 7, no. 3, pp. 78–94, 2018.
11. R. Somasundaram, V. Pillai, and K. Anand, “Improving donor involvement,” *Proc. Int. Conf. Health IT*, vol. 8, no. 4, pp. 89–95, 2020.

12. M. Alam and Z. Khan, “Responsive design in healthcare apps,” *Journal of Interactive Design*, vol. 9, no. 3, pp. 142–158, 2020.

13. World Health Organization, “Blood Safety & Availability,” 2021. [Online]. Available: https://www.who.int
14. MongoDB Inc., “MongoDB Documentation,” 2023. [Online]. Available: https://mongodb.com/docs

15. Express.js, “Express Web Framework Documentation,” 2023. [Online]. Available: https://expressjs.com

16. Node.js Foundation, “Node.js Documentation,” 2023. [Online]. Available: https://nodejs.org/en/docs

17. Redux Toolkit, “State management made easy,” 2023. [Online]. Available: https://reduxtoolkit.js.org

18. Mongoose ODM, “MongoDB modeling for Node.js,” 2023. [Online]. Available: https://mongoosejs.com

19. Axios, “Promise-based HTTP client,” 2023. [Online]. Available: https://axios-http.com

20. Google Developers, “Google Sign-In for Web,” 2023. [Online]. Available: https://developers.google.com/identity/sign-in/web

21. OAuth, “Open standard for access delegation,” 2023. [Online]. Available: https://oauth.net

22. Bootstrap Team, “Bootstrap CSS Framework,” 2023. [Online]. Available: https://getbootstrap.com

23. Morgan Middleware Docs, “HTTP request logger middleware for Node.js,” 2023. [Online]. Available: https://github.com/expressjs/morgan

24. Swagger Docs, “OpenAPI Specifications for Healthcare Apps,” 2023. [Online]. Available: https://swagger.io

25. GitHub, “Open-source blood bank projects,” 2023. [Online]. Available: https://github.com

26. InterviewBit, “Node.js performance optimization tips,” 2023. [Online]. Available: https://www.interviewbit.com

27. IPSOS Mori, “Healthcare donor behavior statistics,” 2022.

28. InterviewBit, “Interview preparation platform,” 2023. [Online]. Available: https://www.interviewbit.com

29. The Indian Institute of Management and Engineering Society, 2023.