



BLOOD DONATION MANAGEMENT SYSTEM USING CURD OPERATION

Nayanjyoti Sahu, Sulochana Sahu, Prof. Smruti Smaraki Sarangi, Computer Science and Engineering, BPUT, Bhubaneswar, INDIA. Email:- Nayanjyoti.sahu2020@gift.edu.in

ABSTRACT

The blood donation management system is a web application that allows users to manage blood donations and donors. The system is built using Node.js, Express, and MongoDB, and uses CRUD (create, read, update, delete) operations to manage data.

The system includes the following features:

Authentication: Users can sign up and log in to the system using their email and password. The system uses bcrypt for password hashing and JSON Web Tokens for authentication.

Donor management: Users can create, read, update, and delete donor records. Each donor record includes the donor's name, contact information, blood type, and donation history.

Donation management: Users can create, read, update, and delete donation records. Each donation record includes the donor's name, the date of the donation, and the amount of blood donated.

Search and filter: Users can search for donors and donations by name, blood type, or date.

Google OAuth2 authentication: Users can sign up and log in to the system using their Google account. The system is designed to be easy to use and maintain, and includes error handling and validation to ensure data integrity. The system can be deployed to a web server or cloud platform, and can be accessed from any device with an internet connection.

Overall, the blood donation management system is a powerful tool for managing blood donations and donors, and can help save lives by ensuring that blood is available when and where it is needed..

I. INTRODUCTION

Rudhiram, it's a Sanskrit word that means blood, It is a web app that acts as an intermediate between the donor and recipient of blood. According to research in every 2 seconds, a person needs blood in India. Besides vital need, getting the right blood at the right time is also important to save a life. In today's life, we came across many situations where we need blood for the survival of the patient, it could be an emergency accident, delivery of a baby, or any other possibility. In all these situations we need blood on an urgent basis to save the life of the person. This is where Rudhiram comes into play, the web app links all the hospitals, blood banks, blood camps, and users with their detailed information on a single platform which we called "Rudhiram". With the help of this application as a platform where any person who is willing to donate blood or any person who needs blood (patient) can help each other. The application keeps the track of the blood available in the hospital or the blood bank, as soon as there is any update in the hospital database, it will also be reflected on the application with the help of API. The application can keep donor information and the stock of blood in the hospitals and the blood bank. The paper presents a dynamic full-stack [MERN] web-application with the use of the following technology:

1. React.js: It's a JavaScript framework used to create the front-end (UI/UX) of Rudhiram ^[1]. Implemented the v17.0.1 with new tools like Hooks etc.
2. Redux: A Predictable State Container for JS Apps ^[2]. It's used to create a centralized store for storing states of the app.
3. Express.js (Node v12.18.2): Express is a minimal and flexible Node.js web application framework ^[3]. It's used for the development of the back-end and API of Rudhiram.
4. MongoDB (v4.4): A NoSQL database with JSON-like document schemas ^[4]. Used for implementing a database for Rudhiram. Robo3t a MongoDB GUI is used for implementing the database.

The system has 5 modules: Admin module, Hospital, Donor, Blood Camps, and Recipient module

respectively. The admin module has access to every movement of the application. The Hospital or blood bank can register to the application, after registration, they need to share their database with the application, and the admin will verify the data, and then its database will get integrated with our database which eventually reflects on the front-end. The donor can register himself on the application and can search nearby hospitals to donate blood and can generate a ticket as an appointment. The recipient has to register to the application and after registration, he needs to provide the required blood group which helps our algorithm to find the nearest hospitals with the same or compatible blood. Then the recipient can contact the user or hospital for the blood. Rudhiram is a fusion of modern technologies and algorithms which provide a well-built blood donor and hospital management system to serve mankind.

II. LITERATURE REVIEW

1. “A survey on Blood Bank management system” by professor Animesh Tayal, Harshad Gahre, Akshay Patel, Sagar Jog, Pratik Jain, Jaya Dhawale ^[5]. They have developed a website and an android app by which a solution to the ever-growing requirements of blood supply can be fulfilled. They have used JSP, Bootstrap & Java for the frontend & MYSQL for the backend
2. “CBBR Centralised Blood Bank Repository” by Ibrahim Fawze Akar, Tukur Anas Mohammad, Mohammad Ismail Z. They have developed a centralized web-based system using HTML5/CSS & JSP on WWW Platform ^[6]. H2 Database is used by them and has hosted it on Apache Web Server. This also supports transactions on both acceptors and donor's side, an acceptor can pay for the transfusion fee and Centers fee and a donor can also be paid for his services.
3. “Blood Donation Management System” by KM Akkas Ali, Israt Jahan, Md. Ariful Islam, Md. Shafa-at pravez^[7] have developed a web-enabled and mobile-based application to maintain day to day transaction in the blood bank. It creates an e- information about the donor and the organization that is related to donating the blood. They have used ASP.net for the entire front-end and SQL server 2008 for the backend.
4. “Implementation of Blood Donation Application using android Smartphone” by Monika Mandole, Pradnya Jagtap, Prachi Mhaske, Sonali Vidhate ^[8]. They have developed an android application that is designed to store, process, retrieve and analyze the information concerned with the admin. They have used Java, JDK, GCM, database in their project.

III. METHODOLOGY

This paper represents a blood donation management system using modern technologies with a step-by-step methodology as discussed below.

a. Concept & Design:

With the rise in React framework, the front-end becomes seamlessly fast, more dynamic, responsive, and user-friendly. But still, a lot of websites use aged technologies, this problem is a big interruption for a blood management system where rapid blood transfusion happens. The paper presents the full stack [MERN] blood management system with dynamic API support

a.1 Stakeholders of the application

1. Users (Donors): The person who wants to donate blood. First-time users need to sign-up with an e-mail address and password or by using OAuth, after sign-up, he/she will receive an email where they update their profile for further consideration for blood donation. Admin will verify their details and add them to the database as donors with respective blood types. A regular user can directly login using register e-mail and they can easily choose the nearest blood donation camp to schedule a blood donation visit as per their convenience with ease as it's a simple and paper-free process.
2. Users (Patients): The person who needs blood in an emergency. Users also have login/sign-up they have to follow the same step as the users (Donor) did. After login they can go to the blood fetch page created in the application where they need to provide the details of blood type and area after

pressing the search button, they will get the list of hospitals/users with available compatible blood type. Then they can contact them to get the blood they can also contact us to help them in the delivery process.

3. *Hospitals*: Any hospitals can register with us by providing proper detail and after verification, they get registered. In an emergency, they can contact other hospitals/user for blood. The registered hospital gets showed up on Rudhiram as a blood bank if they have compatible blood available.
4. *Blood Camps*: Any blood camp can register with us by providing proper detail and after verification, they get registered. The registered camp gets showed up on Rudhiram while users search for blood camps near them.

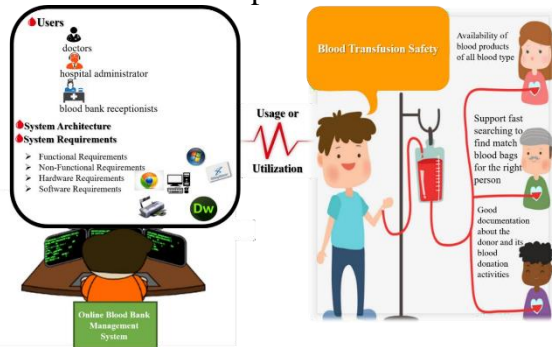


Figure 1: Technical architecture of the proposed system

5. *Administrator*: Admin manages and supervises the working of every department and layer of Rudhiram. Admin control, create, update, delete and verify the records of users, hospitals, and blood camps as per need.

All of these different modules are connected together as React components which in turn connected with Redux a state management tool. Redux provides a centralized store and the whole app is connected with this store, some components can update the state in-store and some can receive the state of the components and hence the data can easily be shared between different components of the app.

API requests are handled by the Express server created using Node JS. Database connectivity is created with MongoDB using Express and all of the NoSQL queries are executed and sent back to the client in JSON format which is rendered smoothly by React. It's also used for searching algorithms handling the compatible blood type HTTP request query and hospitals or blood camps in user state. Express is also used for session management and cookies. All of the data is being stored in the database using Axios and HTTP requests. This data can be used further for medical and research related to blood. E-mail, phone call, location service is provided to the end-user of registered hospitals and donors.

The paper presents the app development in 2 parts first client-side development i.e., React/Redux application, and server-side development i.e., Express (node.js) application both developed on VS code editor on a Linux machine (Ubuntu OS). Then finally both apps are deployed on the same server (Heroku) using distributed client-server computing technology providing a proxy of server in a client application.

b. Implementation & Method:

The paper describes the detailed process of building the web application in a different module such as front-end i.e., client-side development, back-end i.e., server-side development, and then finally deployment on Heroku distributed server.

b.1 Front-End or Client-Side Development:

React JS is intensively used for the creation of the front-end with CSS and some HTML-5. Reacts component tool is used for the creation of different custom components such as Home.jsx, Main.jsx, Loin.jsx, SignUp.jsx, BloodDonate.jsx, BloodFtech.jsx, etc which are interlinked well to provide a smooth user experience. CSS and JavaScript are used for making the app more responsive for different size devices. Material-UI ^[9] and React Bootstrap ^[10] are used for special effects,

buttons, forms, cards, etc. React Router Dom is used for creating different routes and links which in turn makes the application seamlessly fast, that a user gets a native app-like experience on the browser as there is no flashy blank page between route transitions. Redux Centralized store is used for data connectivity within separate components of React. Redux provides a dispatcher that dispatches actions on the store for state updation which in turn provides consistent data for the whole app. Axios is used in a client for making an API request to the server. JSON format is used for exchanging data between client and server. Many other npm packages and react library are used to provide better performance and user experience.

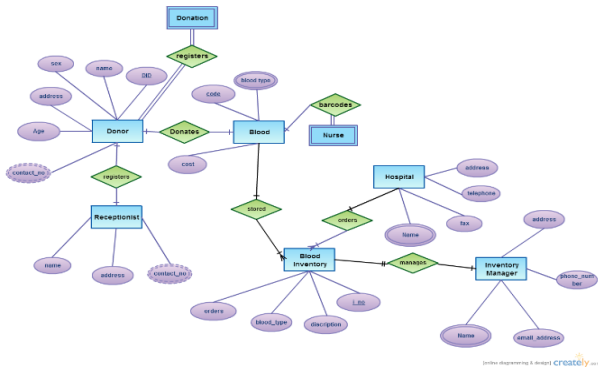


Figure 2: Class Diagram for Bloodlink App.

b.2 Back-End or Server-Side Development:

Express is used for the creation of the back-end with Node. REST full API is created with support for get, post, delete, and update routes using Express. The API is used to server the HTTP request by the client for the transaction of data with the database. Passport and npm package is used for session management, cookies management, and password encryption (salting + hashing). Google OAuth 2.0 is implemented using passport strategy for google. Passport local strategy for authenticating client locally and session management using serializing and de-serializing tokens. A different algorithm is used for answering different queries, java-script in Express is used for this purpose. Express is also connected with python sell to provide data analysis and reports for further research and medical purpose. All the back-end side code is written in java-script and Express so to handle precise API requests and warnings of deprecated npm packages.

MongoDB is used as the database for Rudhiram for the following reasons as its flexible, dynamic, and easy to user-friendly work environment for development. MongoDB Atlas is used to deploy and scale the database in the cloud. It's free (community server only) having Multi-cloud data distribution (75+ cloud regions), secure for sensitive data, and is designed for developer productivity and optimal performance [4]. The database is divided into 4 collections hospitals, blood-camps, users, articles. Each collection has its data, function, and usage according to need.

Hospital collection is used to provide the list of registered hospitals with their details and available types of blood to the recipient of blood searching on the app. Blood Camps collection is used to provide registered blood-camps details, location, and time so that a donor can schedule a visit for donation. Users collection contains the list of all the registered users with us and having all the details so that a person in an emergency can contact them. Articles collection contains the list of blood donation and blood-related articles for general purpose. NoSQL JSON format is primarily used for storing and transporting data between the app and the MongoDB Atlas while Express provides the platform for the same.

Finally, the app is deployed on Heroku a PaaS that enables to build, run and operate applications entirely in the cloud [11]. Using a free server provided to students.

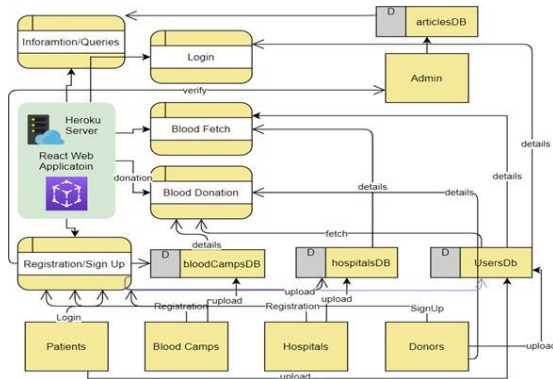


Figure 3: Data flow Diagram for Bloodlink App.

IV. IMPLEMENTATION AND RESULTS

The project is deployed and hosted on Heroku public server and anyone can visit this application at <https://rudhiram.herokuapp.com/>. A detailed guide is given bellow about how the app works for different users and different purposes.

Step 1: Go the <https://rudhiram.herokuapp.com/> from any device of your choice. It's the main page of the app.

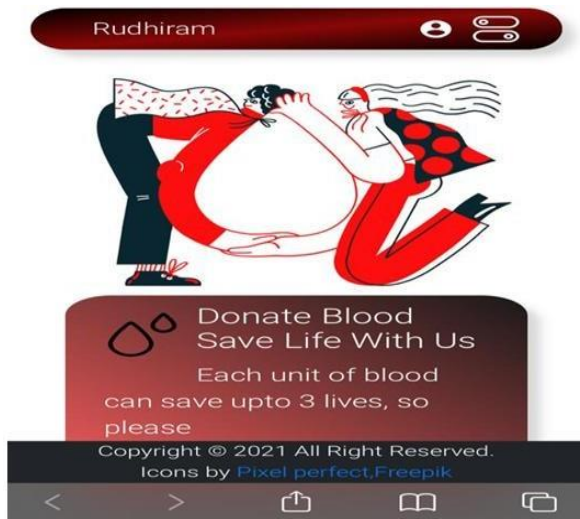


Figure 4: Mobile View of Main Page

Step 2: In step 2 the user of the app have to go the <https://rudhiram.herokuapp.com/reg/bloodcamp> and <https://rudhiram.herokuapp.com/reg/hospital> to register blood-camp and hospital respectively



Figure 5: View of the registration page for hospitals/blood-camps.

Step 3: Login/Signup page for users for verification and upload of details.

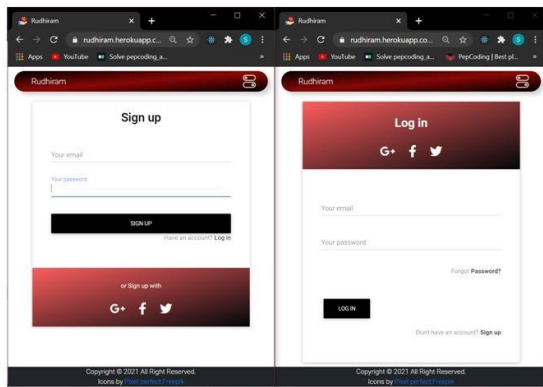


Figure 6: LOGIN/SIGNUP page for the user

Step 4: After Login/signup user will enter into the main page of the application which contains links for blood donation and blood fetch. Available at <https://rudhiram.herokuapp.com/main>.

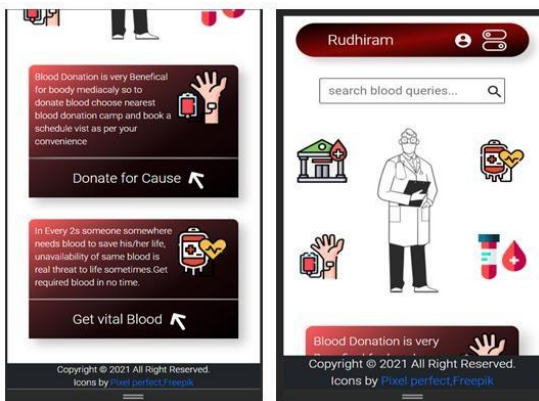


Figure 7: View of the main page of the application

Step 5: Click Donate for Cause button for blood donation at the nearest blood camp in your city. Click Get Vital Blood to receive blood in case of emergence from the nearest hospital or a registered donor.

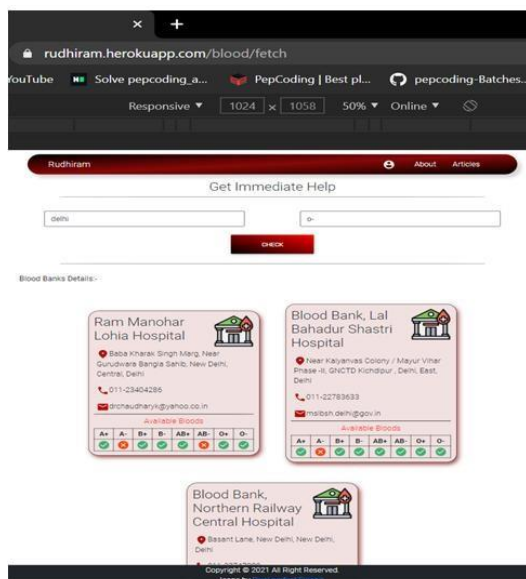


Figure 8: View of donation and blood fetch page of the application



Step 6: Now if you need blood you can just contact the hospital or donor. If you want to donate blood then you need to click on schedule visit and selected a respective day when you can donate blood after you will get the mail and appreciation ticket.

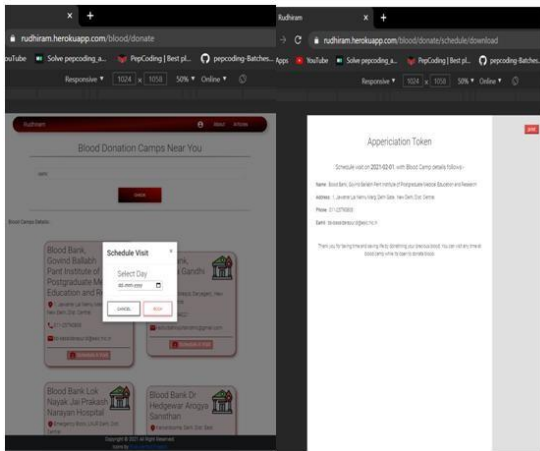


Figure 9: Scheduling Appointments and generating Tokens

Step 7: You can visit your profile for updating and checking your booking history etc. finally you can logout and close the window.

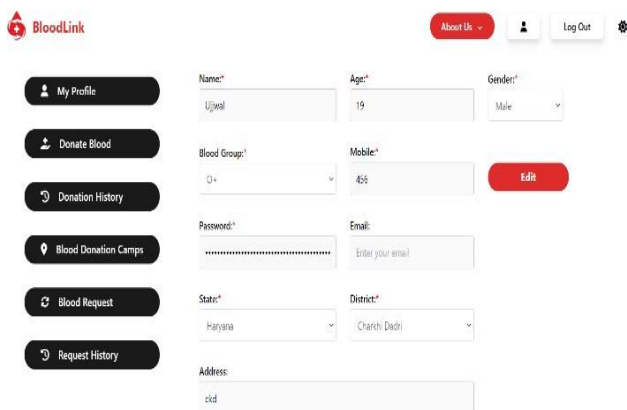


Figure 10: user profile page showed only after login

Robo3t is used as GUI for MongoDB a which contains four collections each having its own use in the application.

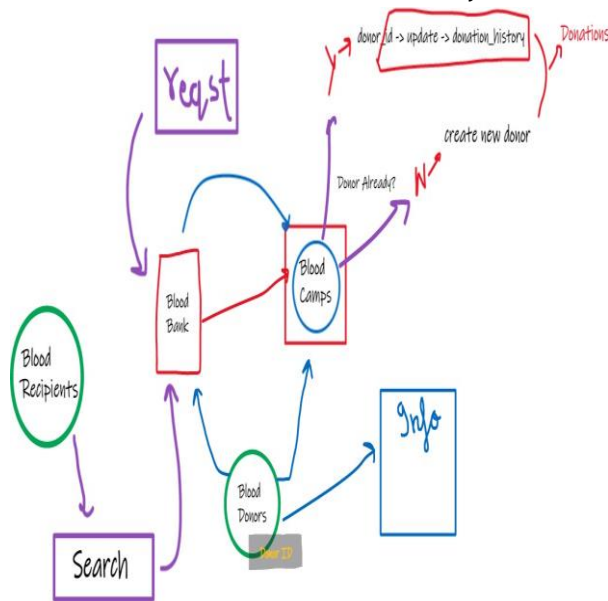


Figure 11: Database collections representation

V. FUTURE WORK

Although the project provides a modern solution for Blood bank and donor management system still there is a chance of improvement. Rudhiram can be further improved by implementing a fast-searching algorithm like Dijkstra, A*, etc. to show results faster. Live blood transaction tracking can be added so to improve the transparency of the blood transfusion process. By applying data analytics, we will get the information that which blood group is rarely available in which area. SMS facilities can be added to notify users and hospitals about the events and process. An E-mail newsletter can be created to increase awareness among citizens for regular blood donation and its benefits as saving lives in emergencies.

6 CONCLUSIONS

The project mainly focuses on the issue of Blood availability when someone needs it on an urgent basis and when someone wants to donate blood and don't have proper guidance for the same. The Blood Donation Management System is a 24x7 system that is essential for different kinds of people like blood donation system personnel, doctors, donors, recipients, and other general users [7]. Rudhiram provides a very efficient and advanced way for blood transfusion between hospitals/donors and patients. The purpose of the project is to simplify and automate the process of searching for blood and to maintain a centralized record of blood donors, recipients, blood donation camps, and the availability of stocks of blood in hospitals and blood banks.

ACKNOWLEDGMENT

The authors are grateful to Bhagwan Parshuram Institute of Technology, New Delhi for providing the platform, resources, and support for this project.

REFERENCES

- [1]. React JS framework available at <https://reactjs.org/>.
- [2]. Redux State management tool available at <https://redux.js.org/>.
- [3]. Node (Express.js) for back-end available at <https://nodejs.org/en/>.
- [4]. MongoDB NoSQL database available at <https://www.mongodb.com/>.
- [5]. A Survey on Blood Bank Management System by Prof. Animesh Tayal, Harshad Gahare, Akshay Patel, Sagar Jog, Pratik Jain Dhawale Department of Computer Science & Engineering S. B. Jain Institute of Technology, Management and Research, Nagpur. IJRTER



ISSN: 2455-1457 available at [<https://www.ijrter.com/papers/volume-3/issue-1/blood-bank-management-system.pdf>].

[6]. “CBBR Centralised Blood Bank Repository” by Ibrahim Fawze Akar, Tukur Anas Mohammad, Mohammad Ismail Z. IJISE Vol.

3 (No.1), April, 2015 available at [<https://www.ftms.edu.my/journals/pdf/IJISE/Apr2015/85-97.pdf>].

[7]. Blood Donation Management System” by KM Akkas Ali, Israt Jahan, Md. Ariful Islam, Md. Shafa-at pravez. AJER Volume-4, Issue- 6, pp-123-136 available at [[http://www.ajer.org/papers/v4\(06\)/O04601230136.pdf](http://www.ajer.org/papers/v4(06)/O04601230136.pdf)].

[8]. “Implementation of Blood Donation Application using android Smartphone” by Monika Mandole, Pradnya Jagtap, Prachi Mhaske, Sonali Vidhate. Available at IJARIIT Vol 3, Issue 6 available at [<https://www.ijariit.com/manuscripts/v3i6/V3I6-1396.pdf>].

[9]. Material-UI for styling available at <https://material-ui.com/>.

[10]. React Bootstrap for styling available at <https://react-bootstrap.netlify.app/>.

[11]. Heroku server is used for hosting, available at <https://dashboard.heroku.com/>.