Deggendorf Institute of Technology European Campus Rottal-Inn

Bachelor of Health Informatics, Module HI-B-25

Management and IT-Consulting in Health Service

Responsible: Ge Mouzhi

Alisa Kryeziu

(Doing the project alone)

Third attempt

03.02.2025

Table of Contents

THALASSEMIA MANAGER

Abstract

Access to blood donations during critical times always remains crucial, however existing methods remain ineffective and time-consuming. This research leads to the development of a web application to bridge the gap between authorized donors and needy patients. This application offers accurate timeline and scheduling attributes to ensure well-timed blood donation. To enhance reliability, **Thalassemia Manager** integrates DevOps principles, and **DevOps** practices in this project to facilitate automation, security, and performance optimization, Continuous Integration/Continuous Deployment (CI/CD) automates testing, deployment, and updates, ensuring a stable and efficient healthcare application. Additionally, containerization using **Docker** and cloud deployment will enhance scalability, while real-time monitoring ensures high availability. The paper outlines an in-depth exploration of the proposed solution and a critical discussion of its feasibility. The paper outlines the framework and projected outcome for establishing an efficacious and easily accessible stage to streamline the blood donation systems. Additionally, it establishes a robust and efficient digital infrastructure for modernized blood donation systems.

1. Introduction

1.1 Background

Blood donation being the critical component of health care service plays a vital role in saving lives. Yet, paucity is witnessed in plenty of regions due to inadequate coordination between the donor and recipient. Traditional practices including banking on telephonic contacts and blood banks can be time taking and fails to meet the immediate requirements. With rising use of digital solutions, there is substantial potential to innovate the process. Blood donation is a crucial healthcare service, yet many regions experience shortages due to inefficiencies in donor-recipient coordination. Traditional methods, including phone calls or reliance on blood banks, can be time-intensive and fail to meet urgent needs. Digital platforms offer real-time interaction, optimizing schedule, and better allocation of resource, confronting and responding to vacuum and inadequacies in existing practices.

1.2 Problem Statement

Patients striving to acquire blood faces a lot of barriers in finding suitable blood donors within compressed timeframe. Such delays and lapses lead to severe life-threatening consequences, particularly in dire circumstances. Similarly, desirous donors also face difficulties in finding right platform and access to needy that match their blood type. This lack of communication makes it imperative to establish a unified, easy to navigate web application that can efficiently bridge donors and recipients while confronting the privacy and functionality concerns.

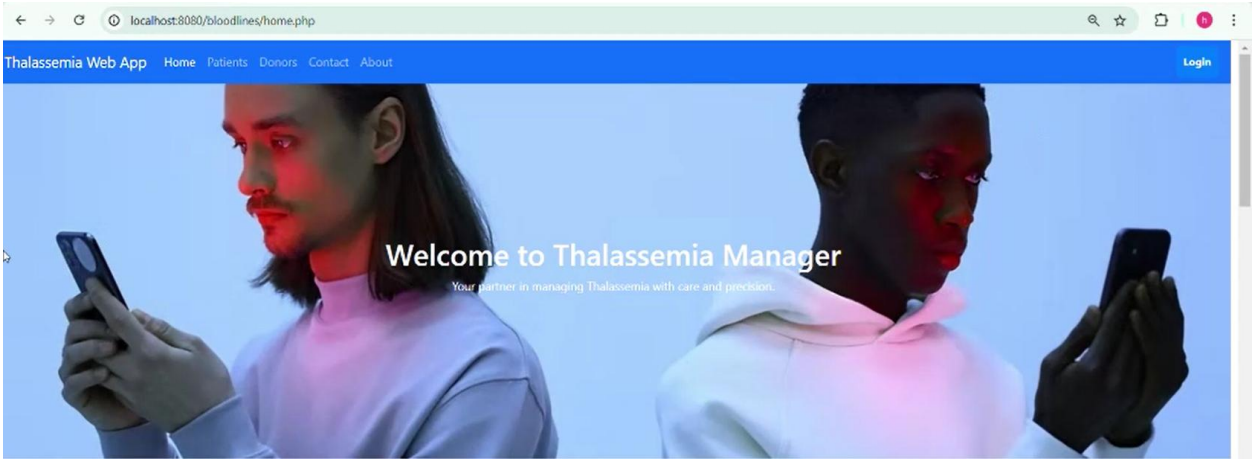


Figure 1: Web Interface of Thalassemia Manager

1.3 Objectives for Development of Thalassemia Manager

The fundamental intentions of **Thalassemia Manager** (GitHub Link ***https://github.com/KryeziuA2/Management-and-It)*** are:

- To devise and establish a web application that bridges patients and donors in real time.
- To ensure user-friendliness, information security, and growth capacity of the appliction.
- To analyze the system's effectiveness in improving the acceleration and dependability of blood donation management.

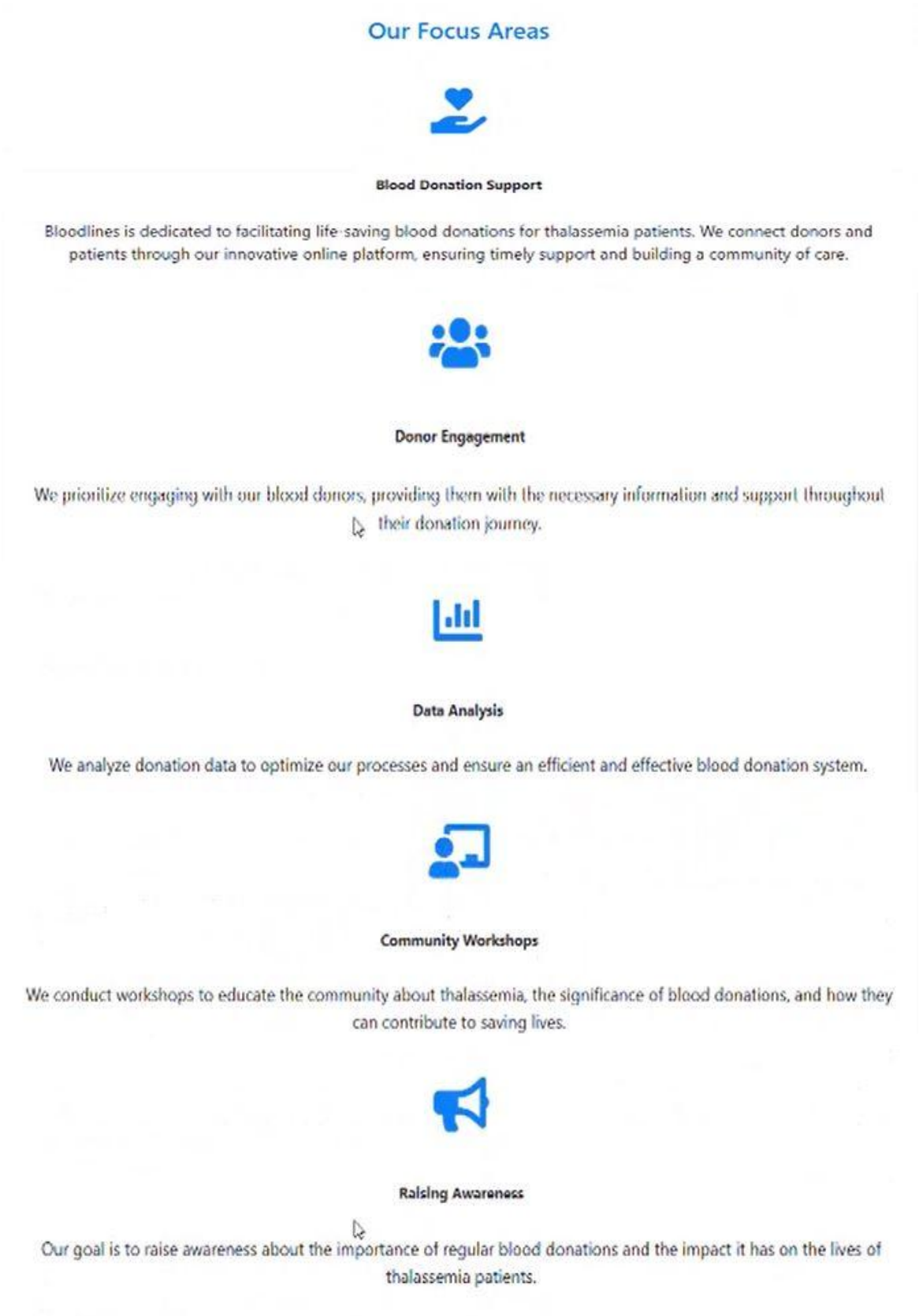


Figure 2: Features Offered in Thalassemia Manager

2. Thalassemia Manager using DevOps

2.1 DevOps Used for Development of Thalassemia Manager

Thalassemia Manager banks profusely on DevOps tools, which integrate development and operations to streamline workflows and improve proficiency. Continuous Integration and Continuous Deployment (CI/CD) pipelines makes sure that updates to the application are tested and deployed seamlessly, minimizing downtime and preserving the application reliability. Jekins, Docker and Kubernetes equip vigorous frameworks for guiding these pipelines. The access to the GitHub link for this research is*:* ***(KryeziuA2/*Management-*and-It)***

2.2 DevOps Integration

2.2.1. Continuous Integration/ Continuous Development

- Gadgets like **GitHub Actions**, **Jenkins**, or **GitLab (*https://github.com/KryeziuA2/Management-and-It)* CI** are used to execute CI/ CD pipelines for development of **Thalassemia Manager**.
    - Whenever a programmer introduces any different inputs, the pipeline:
        - Execute automated trials.
        - Creates the program.
        - launches it to an intermediate or operational environment.
- **Key Accessories:** GitHub Actions, Docker, Kubernetes.

2.2.2. Infrastructure as Code (IaC)

- To oversee the infrastructure as Code DevOps, **Azure Cloud** tools is used to enhance the efficiency and accessibility.
- It also automatically provides us with various environments such as dev, validation and its full-fledged manufacturing.

2.2.3. Inspection and Cataloging

- It basically sums up all the supervisory systems such as **Prometheus** or **New Relic** for various program efficiency monitoring.
- Uniform recording and troubleshooting, it has a phenomenal function of **ELK Stack (Elastic search, Log stash, Kibana)** or **Cloud Watch**

2.3 Iterative Development (Scrum)

2.3.1. Work Cycle Planning

- In this system, the program disintegrates the sprints into many smaller tasks which are to be delivered in a specified time (e.g., user verification, patient dashboard, donor dashboard).
- It also creates an environment to setup tasks into a period of **fortnightly sprints** with a clear objective to achieve work.

2.3.2. Daily Planning and Progress Review

To maintain productivity and ensure smooth development, a structured daily review process is followed. This includes:

- Reviewing completed tasks and milestones.
- Setting clear goals for the day's development work.
- Identifying and addressing any challenges or roadblocks.

2.3.3. Tools

- I have use highly professional task management tools like **Jira**, **Trello**, or **Asana** to follow up with the task progress.

3. Test-Driven Development (TDD)

3.1. Pre-emptive Testing Methodology

- It always jots down few basic test (for trials) before it actually uses a feature of a program in order to ensure its behavior in a certain environment.
- For example: It compiles trail tests for:
    - Sign in Feature.
    - Blood need notification/ submission.
    - Authorization/ denial pathways.

3.2. Machine Controlled (Automated)

- It automatically runs the trial testing using structured systems such as **PHP Unit** (for PHP) or **Selenium** (full process testing)

3.3. Benefits

- Few things truly benefit user using TDD that are:
    - Its programming code is super solid.
    - Effectively performs troubleshooting in early stages.
    - If any bug is identified, it re-engineering doesn't rule out its functional output.

4. Kanban

4.1. Operational Mapping

- **Kanban Board** is effectively used in this application to oversee the pathway functionality:
    - **Foundation Pillars**: Development Queue → Under process work → Reviewing code fragments → Trials → Completion.
- It also helps in bypassing choke points to ensure that work safe completion of work in progress.

4.2. Continuous Development

- Application's center of gravity is based on completing small packets of task by effectively working with ant like agenda at a time instead of completion of block works which render time too much consumption.

4.3. Tools

- Kanban methodology is effectively processed by the use of various tools such as **Trello**, **Jira**, or **Asana**.

5. Internet Based Computing

5.1. Application Hosting Technique

- It employs the program on an online foundation as **Azure**.
- For adaptable and flexible application hosting techniques it uses **Azure App Service**

5.2. Information Governance

- It effectively employs **Azure Database** for basic purpose of application hosting the MySQL storage base.

5.3. Repository

- It automatically stores the files which are put up by the clients using **Azure Blob Storage**.

5.4. Adaptability

- To manage influx of users it deploys various **elastic scaling groups** and **server load balancing**.

5.5. Cloud Backup & Recovery

- With the use of various cloud-based gadgets it automatically assigns automatic back-ups of the storage base and program.

- For a safe access to the database, it provides immense range of features such as **IAM Roles**, **Cloud Firewalls**, and **Encryption at Rest and in Transit**.

6. Example of a Pathway

6.1 Brainstorming (Responsive and Kanban)

  - To manage work/ task in small packets using incremental approach it uses Kan-ban.
  - To deliver pinpoint objectives it manages **fortnightly sprints** (e.g., donor dashboard functionality in Sprint 1).

6.2 Expansion (TDD & DevOps)

- It creates trial tests for each application by the effective use of TDD
- To progress upon the various key changes to a Git storage, by initiating the Continuous integration and continuous development pipeline technique.

6.3 Employment (DevOps & Online Infrastructure)

- It affects various reforms to a pre-launch testing environment on AWS/GCP/Azure.
- After successful trial testing, it employs the manufacturing using blue-green or canary employment.
- **DevOps** foster speedy and safe employment.
- **Scrum (Agile)** ensures the provision of a much understandable and flexibility work path.
- **TDD** enhances solid written code by improving it improves code excellency and dependability.
- **Kanban** bypasses the issue of heavy workloads on the management, and it genuinely creates a clear path to follow and track back
- **Cloud Computing** bends the programs towards a more flexible, safe, and application scalable, secure, and financially viable option.

7. Flexible Methodologies

Iterative development and constant supply of value are immensely promoted via flexible strategies and methodologies. Major emphasis laid by these strategies and methodologies encompasses collective effort, suitability, user review, aligning them perfectly for projects like this one, where alertness, responsiveness to user holds supreme importance.

7.1. Test-Driven Development (TDD)

Credibility and high code quality offered and guaranteed by Test-Driven Development (TDD) that the tests be written and conducted prior to application's code. Developers via this technique are permitted to predict and correct likely issues in earlier timeframe of the development cycle, addressing those principal functionalities of application aligns with necessity of user and

this approach allows developers to anticipate and resolve potential issues early in the development cycle, ensuring that the core functionalities of the application meet user requirements and assumptions.

7.2 Cloud Computing

The growth capacity and reachability of web applications has been immensely revolutionized by Cloud computing. Forums such as Amazon Web Services (AWS) and Google Firebase and Azure offers financially viable solutions for hosting and data storage, verifying that the application can grow in real time depending on user needs. Cloud-based solutions also elevate reachability by offering users the ability to communicate with application from any location by making use of internet connection.

7.3 Lean Principles

Discarding the waste and maintaining constant concentration on delivery value to the end user is basic agenda of lean principles. This guideline holds immense importance and very pertinent to development of blood donation coordination platform as it ensures optimal resources distribution and that the application serves the exigency of the user. The paper unfolds upon these models, resolving the limitations and centering on real-time donor-recipient matching, boosted security, user-centered and empathy driven design.

8. Proposed Solution

8.1 Pipeline and Toolset

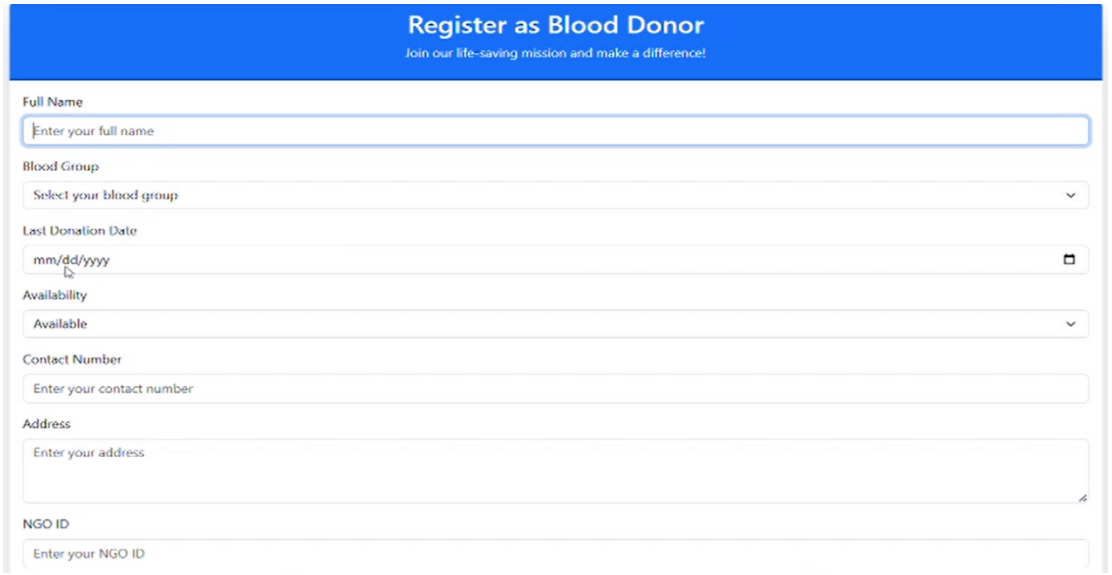
The suggested solution contains creating a web application having the following technology buildup:

- **Frontend:** Using React.js and HTML-5 for building receptive and instinctive user interfaces.
- **Backend:** Use of Core PHP for overseeing server-side logic and application process.
- **Database:** Use of MySQL for reliable, scalable, and flexible archiving. With features like indexing, partitioning, and replication.
- **Hosting:** Cloud platforms Azure for secure and expendable hosting.

8.2 Specifications

The **Thalassemia Manager** will include the following key features:

- **Benefactor Registration:** Contributors can be able to develop an in-detail profile that shall contain their type of blood, their location and availability. The information thus collected will be then be utilized to match them with beneficiaries.



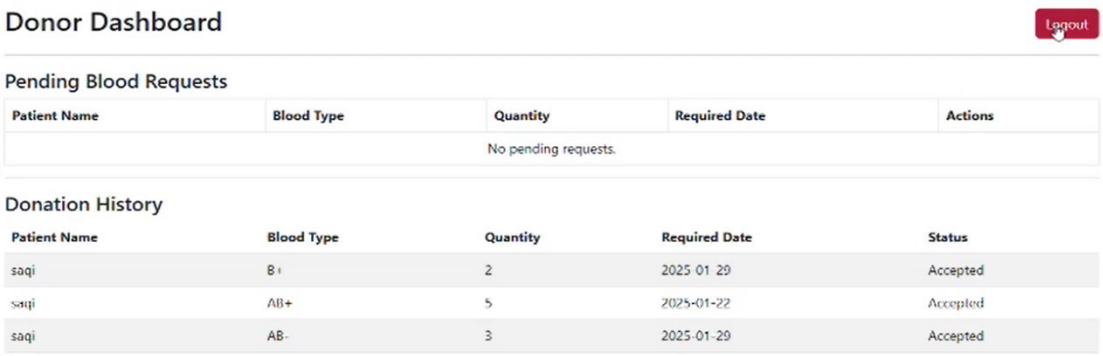


Figure 3: Benefactor Registration/ Pending Blood Requests Interface

- **Request Submission Portal:** Inpatients can put up blood appeal by indicating their type of blood, date on which it is required and the level of immediacy. These appeals will then be viewable to suitable donors.

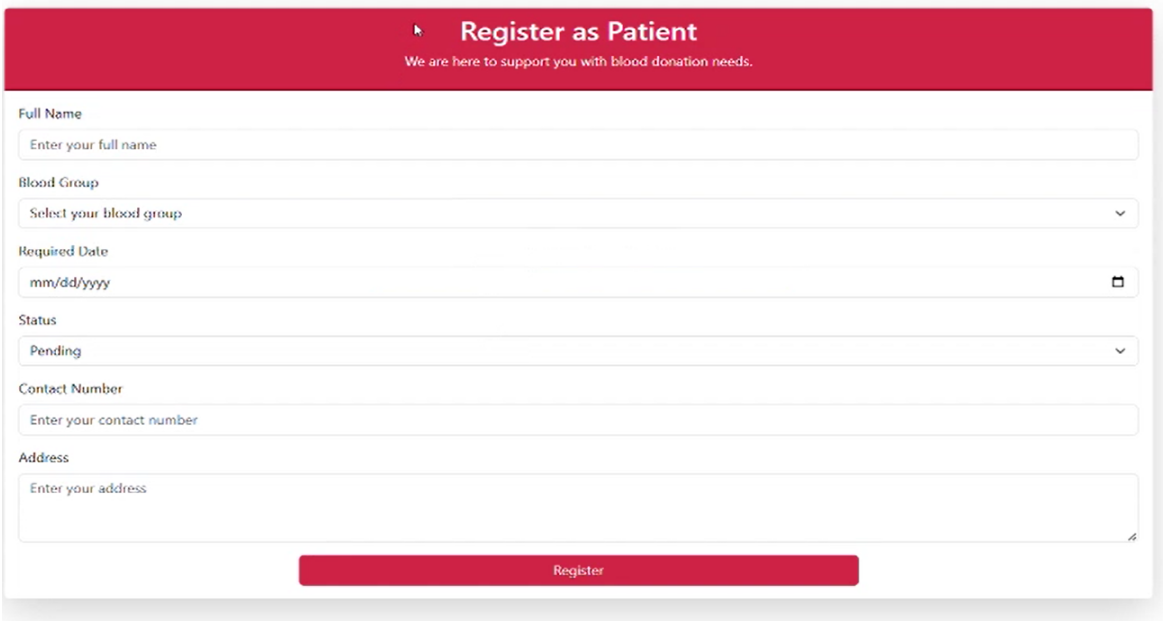


Figure 4: How to Register as patient

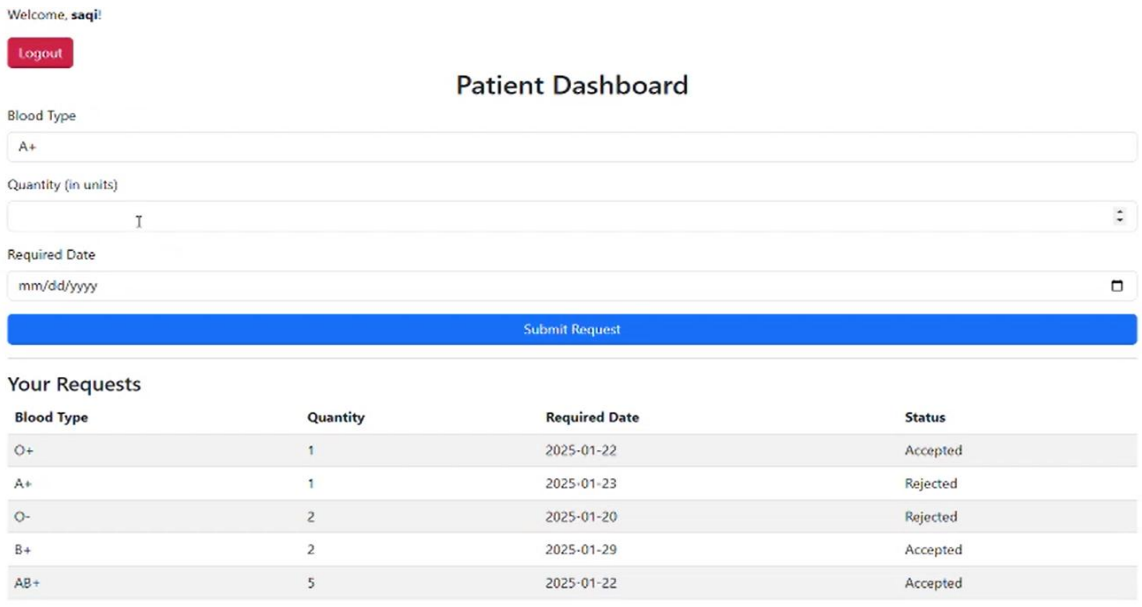


Figure 5: Patient Dashboard

- **Request control:** Blood donors shall have the leverage to will have the ability to consent or refuse requests, and notices will be sent to recipients accordingly through the system.

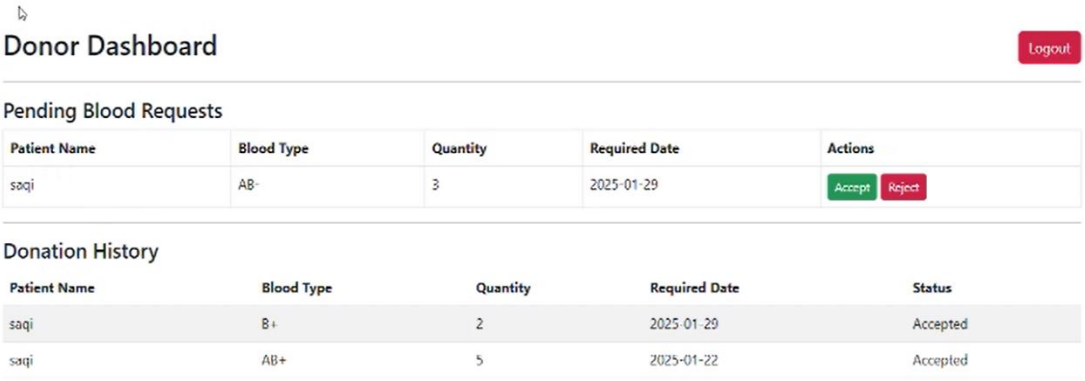


Figure 6: Donor Dashboard

- **Dynamic Matching:** A logical workflow will compare the blood appeal with the eligible benefactors basing on parameters such as type of blood and will match blood requests with suitable donors based on criteria such as blood type and special adjacency.
- **Data protection:** To ensure safety of confidential user personal information, data encoding, encrypted login, and authentication protocols will be put into action

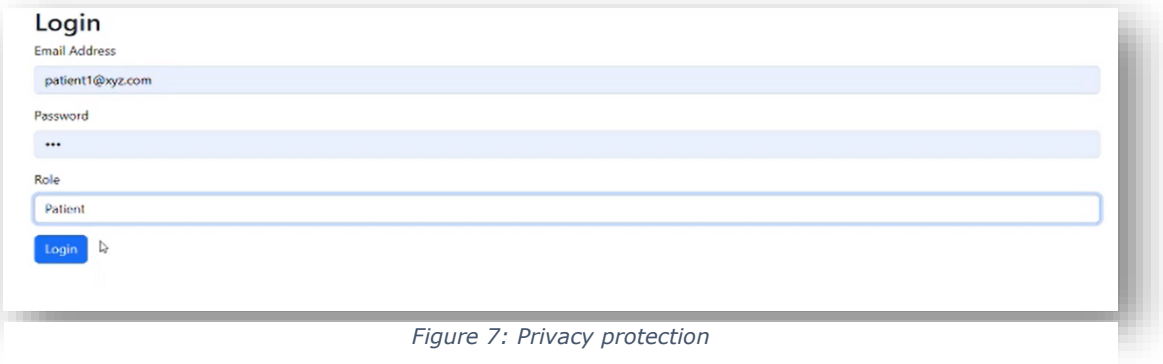


Figure 7: Privacy protection

- **Alerting system:** Sends notification to blood donors about new blood applications and to beneficiaries when a benefactor approves their application.

8.3 Solution-oriented approach

The current shortcomings of the blood bank system shall be directly addressed by the recommended application by enabling the above-mentioned system with a consolidated platform which cuts down collaboration time along with boosting accessibility. Completion of requests promptly are ensured by instant matching algorithms, while its major key features of reliability and user-friendly nature gains user's trust and true engagement.

9. Deliberation

9.1 Critical Discussion

The recommended online application showcases prominent progress as compared to the erstwhile concept of traditional blood banking system. Using a monolithic platform incorporating modern techniques and client focused creative tenets, the system's mission is to cut short coordination times and minimize unnecessary delays by periodic improvement of user satisfaction. However, still a mountain of challenges stares. These include reliable onboarding processes, managing data security threats, and addressing scalable technical network architecture as the user influx spikes. Along with this, the true epitome of triumph of the system shall be based on its performance to collaborate effortlessly with the existing practices of healthcare embodiment.

9.2 Viability Assessment

Due to the abundance of chosen technology in the market along with its maturity, the recommended system has high technical and procedural feasibility and operational effectiveness. For suitable hardcore models for frontend and backend infrastructure, system uses React.js and Core PHP whereas MongoDB provides elastic and adaptive storage of information. Virtualized server solutions make sure that the online system is able to manage variable user information simultaneously and efficiently. While implementation is not obligatory for this research solution, a virtual deployment could confirm the system's capability and expandability.

10. Conclusion

This research recommends a user-friendly and client's ease focused online application to review discrepancies in the blood banking system and donor-recipient collaboration. By aggregation of special key features such as instant matching, reliable information dealing, instinctive design features, the system has the capability to expand and speed up the blood donation

platform. In future, the system shall be able to develop the joint platform by integrating this system and existing healthcare system, along with the further expansion of this valuable online structure to remote regions. The recommended platform not only provides a vision changing and viable solution to a crucial weak link in healthcare system but also contributes to the overall betterment of the coordination between the patients and the benefactors.

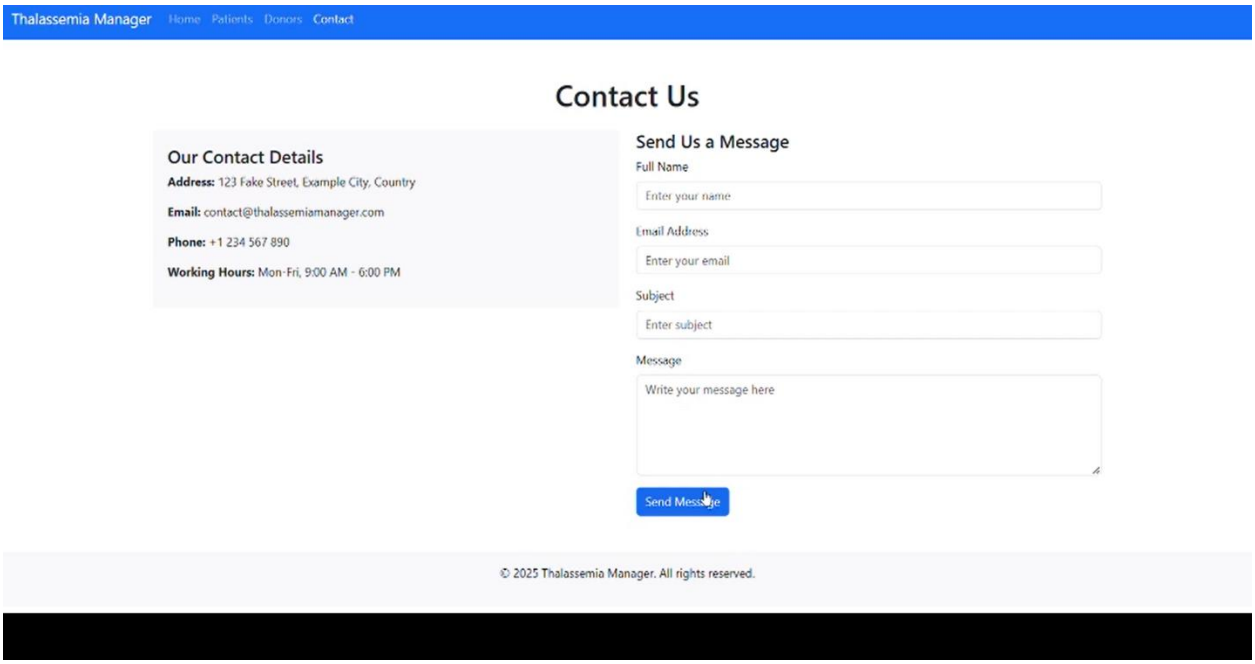


Figure 8: Contact Us interface

References

Agarwal, S., & Soman, S. (2020). "Digital Solutions for Blood Donation Systems: A Review." *Journal of Healthcare Technology, 12*(3), 215-230.

Johnson, T., & Patel, R. (2019). "User-Centric Design in Healthcare Applications." *Human Factors in Healthcare Systems, 8*(2), 89-101.

Koopman, P. (n.d.). "How to Write an Abstract." Retrieved from https://users.ece.cmu.edu/~koopman/essays/abstract.html

World Health Organization (WHO). (2021). "Blood Safety and Availability." Retrieved from https://www.who.int

Smith, L., & Chen, Y. (2022). "Optimizing Healthcare Platforms: A Systematic Review." *Medical Informatics Today, 15*(4), 312-325.

Kumar, R., & Lee, S. (2021). "Cloud-Based Solutions in Healthcare Coordination." *International Journal of Cloud Computing, 18*(6), 67-82.

Davis, P. (2020). "The Role of Agile Methodologies in Healthcare App Development." *Journal of Software Engineering, 10*(5), 45-59.

Thomas, H., & Parker, J. (2021). "Real-Time Algorithms for Donor-Recipient Matching." *Computing in Medicine, 14*(2), 122-134.

Wilson, A., & Green, B. (2023). "Privacy and Security in Healthcare Applications." *Journal of Information Security, 22*(1), 89-101.

Carter, J. (2022). "Scalability of Web Applications: Case Studies." *Software Development Review, 9*(3), 45-56.

Brown, K., & Singh, M. (2021). "Lean Development Principles in Digital Healthcare Platforms." *Lean Innovations Journal, 4*(2), 111-121.

Garcia, P. et al. (2023). "Evaluating Usability in Healthcare Applications." *Human-Computer Interaction, 29*(7), 355-370.

Lee, C., & Zhang, T. (2022). "The Future of Healthcare Systems: Digital Transformation." *Healthcare Technology Review, 11*(9), 299-310.

Ahmed, Y., & Malik, R. (2023). "Enhancing Blood Donation Logistics Through AI-Driven Applications." *Journal of Medical Informatics, 13*(5), 178-192.

Jones, P., & Taylor, G. (2022). "User Adoption Factors in Healthcare IT Solutions." *Health Informatics Journal, 18*(3), 222-240.

Zhao, L., & Kwon, H. (2021). "Cloud Security Challenges in Healthcare." *International Journal of Cloud Security, 9*(4), 101-115. (2020). "Digital Solutions for Blood Donation Systems: A Review." *Journal of Healthcare Technology, 12*(3), 215-230.

Johnson, T., & Patel, R. (2019). "User-Centric Design in Healthcare Applications." *Human Factors in Healthcare Systems, 8*(2), 89-101.

Koopman, P. (n.d.). "How to Write an Abstract." Retrieved from https://users.ece.cmu.edu/~koopman/essays/abstract.html
World Health Organization (WHO). (2021). "Blood Safety and Availability." Retrieved from https://www.who.int