

CENTRALIZED BLOOD BANK

A Project Report

Submitted for the partial fulfillment for the award of the degree of

B.Tech (IT) 3rd Year

Submitted by

Deeksha Tiwari (2216820)

Diksha Rawat (2216823)

Kritika Bhandari (2216848)

Under the supervision of

Dr Roopesh Kumar



Department of Computer Science

Banasthali Vidyapith

Session: 2024-25

Certificate

Certified that following students have carried out the project work titled "**Centralized Blood Bank**" from **1 August 2024 to 31st March 2025** for the award of the **Bachelor of Technology** from **Banasthali Vidyapith** under my supervision. The thesis/report embodies result of original work and studies carried out by students themselves and the contents of the thesis/report do not form the basis for the award of any other degree to the candidates or to anybody else.

Member 1: 2216820. Deeksha Tiwari IT

Member 2: 2216823. Diksha Rawat IT

Member 3: 2216848. Kritika Bhandari IT

Dr. Roopesh Kumar


5/4/25

Designation: Asst. Professor

Place: Banasthali Vidyapith

Date: 5-4-25

ABSTRACT

The Centralized Blood Bank Management System is designed to streamline the process of blood donation, storage, and distribution through an integrated digital platform. This system aims to bridge the gap between blood donors, recipients, and blood banks by providing real-time data on blood availability, donor registrations, and request fulfillment.

The system incorporates key functionalities such as blood bank registration, user account management, donor and recipient tracking, blood inventory management, and emergency request handling. By centralizing data, the platform enhances efficiency, reduces wastage, and ensures timely availability of blood for patients in need.

This report outlines the design, development, and implementation of the system, focusing on its Entity-Relationship (ER) diagram, database structure, user interface, and security measures. The system is built to support multi-user access, secure authentication, and role-based authorization to prevent unauthorized data access.

By adopting this digital solution, the blood bank network can significantly improve accessibility, transparency, and responsiveness, ultimately saving more lives through effective blood management.

ACKNOWLEDGEMENT

We, the undersigned, express our sincere gratitude to all those who contributed to the successful completion of our group project on Centralized Blood Bank Management System.

First and foremost, we extend our heartfelt thanks to our mentor **Dr. Roopesh Kumar, Assistant Professor, Department of Computer Science, Banasthali Vidyapith** for their continuous guidance, valuable feedback, and encouragement throughout the project. Their insights and expertise have been instrumental in shaping our understanding and execution of this system.

We also extend our sincere appreciation to our coordinators, **Dr. Deepak Kumar** and **Dr. Neelam Sharma**, for their unwavering support, insightful suggestions, and guidance throughout the project. Their mentorship provided us with the clarity and direction we needed to move forward with confidence.

We are also grateful to **Banasthali Vidyapith** for providing us with the necessary resources and a collaborative environment that facilitated our research and development.

A special note of appreciation goes to our team members, who worked tirelessly, shared ideas, and dedicated their time and effort to ensure the successful completion of this project. The teamwork, coordination, and perseverance displayed by each member played a crucial role in achieving our objectives.

Lastly, we would like to thank our friends for their constant support and motivation. Their encouragement kept us focused and determined to complete this project successfully.

Team Members:

1. [DEEKSHA TIWARI]
2. [DIKSHA RAWAT]
3. [KRITIKA BHANDARI]

BANASTHALI VIDYAPITH

2024-2025

TABLE OF CONTENTS

● Introduction	1
● Requirement Analysis (SRS)	1-14
o Requirement specification	1-3
o H/w and S/w Requirements	3-4
o Feasibility Study	4-7
o Product Functions	7-12
o Use – Case diagram	13
● System Design (SDS)	15-26
o High-level Design	15-17
o ER Diagram/Class Diagrams	18-19
o Database Design	20-22
o Data flow diagrams/Activity Diagrams	23
o Flowcharts/Sequence Diagrams	24-26
● Coding	27-52
● Testing	53
● User Interfaces	54-55
● Appendices	55
● References	55

The purpose of a centralized blood bank management system is to efficiently manage and streamline the collection, storage, and distribution of blood and its components. It ensures real-time data availability across multiple blood banks, enhancing coordination and reducing wastage. The system provides transparency, traceability, and accessibility to both donors and patients, ensuring safe and timely access to blood. Additionally, it helps in maintaining accurate donor records, managing inventory, monitoring blood quality, and facilitating easy access to information, thus improving overall operational efficiency.

REQUIREMENT ANALYSIS (SRS)

2. OVERALL DESCRIPTION

This section will give an overview of the entire system. System will be explained in its context to show how the system will work and introduce the basic functionality of it. It will describe all the stakeholders who will access to the system and what functionality is available for each type. The constraints and assumptions for the system will explained.

2.1 PRODUCT PERSPECTIVE

The **Centralized Blood Bank Management System (CBBMS)** is designed as a web-based that integrates multiple blood banks, hospitals, and donors into a single, unified system. This system is intended to replace the existing fragmented and manual methods of blood donation management, making it easier to track, monitor, and manage blood inventory at various locations.

The system acts as an intermediary between **blood donors** and **blood banks** providing real-time information and facilitating the secure and efficient communication. The centralized system ensures that hospitals have access to required blood types quickly while providing blood banks with up-to-date stock management capabilities.

The CBBMS will be accessible by various users including blood bank administrators, hospital staff, and individual donors. Each user group will have specific access rights and features tailored to their needs, ensuring the system is both functional and secure. It will have a modular architecture to allow future scalability, such as adding new blood banks, integrating with other healthcare systems, and supporting emergency blood requests.

The product will interact with a relational database system (such as MySQL) to store and manage data such as donor details, blood inventory, blood bank locations, and hospital requests. The web application will follow standard client-server architecture, with a front-end developed using technologies like HTML5, CSS3, and JavaScript, and a backend using a robust server-side framework like Javascript.

1. Donor Registration and Management

- **Donor Registration:** Individuals can register by providing personal details (name, age, address, contact details) and medical information such as blood type and eligibility criteria (health conditions, donation frequency).

2. Blood Inventory Management

- **Add/Update Inventory:** Blood banks can add and update details of available blood units, including blood type, quantity, collection date, and expiry date.

3. Blood Requests and Fulfillment

- **Blood Request Submission:** Hospitals or medical institutions can submit requests for specific blood types and quantities through the system.

4. Search Functionality

- **Find Blood Banks:** Users can search for blood banks by location, availability of specific blood types.
- **Search by Blood Type:** Hospitals and other users can search for available blood units based on the blood type required (A+, B-, AB+, etc.), allowing for quick location of required stocks.

6. User Roles and Access Control

- **Admin User:** The system administrator will have full control over the system, including user management and overall system maintenance.
- **Blood Bank Administrator:** Responsible for managing blood stock, fulfilling requests, and coordinating with hospitals.
- **User:** Can submit and track blood requests, view inventory.
- **Donor:** Receive notifications and update profile.

8. Data Security and Privacy

- **Data Encryption:** Sensitive data, including donor and patient information, will be encrypted to ensure compliance with privacy laws such as GDPR and HIPAA.

9. Multi-Platform Accessibility

- **Web Interface:** Accessible via major browsers on desktops and mobile devices.

10. Disaster Recovery

- **Data Backup:** Regular backups of all system data will be stored securely to ensure that the system can be restored in case of a failure or data loss.

User-Friendly Interface: Design an intuitive interface for staff and donors to navigate the system easily.

Automated Notifications and Alerts: Send automatic notifications for requests.

2.3 USER CLASSES AND CHARACTERISTICS

Donors:

- **Characteristics:** Individuals willing to donate blood. They range from frequent donors to first-time donors, often with limited technical expertise.

Blood Bank Administrators:

- **Characteristics:** Personnel responsible for managing blood inventories. They have moderate to advanced technical skills and are familiar with medical regulations for blood collection and storage.

Hospital Staff/Medical Institutions:

- **Characteristics:** Medical professionals who request blood for patients. They need reliable and timely access to blood supplies and have moderate technical skills.

System Administrator:

- **Characteristics:** IT professionals with advanced technical knowledge responsible for maintaining the system's integrity, performance, and security.

2.4 OPERATING ENVIRONMENT

Hardware Requirements:

Client Hardware Requirements

These requirements apply to the computers used by blood bank staff, doctors, and administrators to access the system.

- **Processor (CPU):** Intel Core i5 or AMD equivalent, 2.0 GHz or higher
- **RAM:** 8 GB or higher (16 GB recommended for more intensive usage or multitasking)
- **Storage:**
 - Minimum: 250 GB SSD or HDD
 - Recommended: 512 GB SSD for faster data access
- **Display:**
 - 15-inch monitor or larger

- Minimum resolution: 1920x1080 pixels (Full HD)
- **Graphics:** Integrated Graphics (dedicated GPU is not typically required unless the system has a heavy GUI or visual analytics)
- **Network Interface:**
 - Wired Ethernet (10/100/1000 Mbps) or
 - Wireless (Wi-Fi 802.11 b/g/n/ac)

Server hardware requirements:

Processor (CPU)

- Minimum: Intel Xeon E5 or equivalent (quad-core, 2.0 GHz or higher)

Memory (RAM)

- Minimum: 8 GB

Storage

- Minimum: 500 GB HDD (Hard Disk Drive)

Operating System Compatibility

- Windows Server

Database Server

- Minimum: MySQL

Software Requirements:

- **Operating System (Server):** Windows Server 2016
- **Operating System (Client):** Windows 10
- **Database:** MySQL
- **Web Server:** HTTP server
- **Programming Languages:** Javascript
- **Client-Side:** HTML5, CSS3, and JavaScript for the user interface.
- **Server-Side:** Javascript, JSON, HTTP(Node.js)

Security:

- HTTPS encryption for data transmission.

2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

Regulatory Compliance: The system must adhere to national and international regulations for blood donation, storage, and distribution, such as the World Health Organization (WHO) guidelines, FDA regulations, and local health laws.

Data Security: The system must comply with data protection standards (e.g., HIPAA, GDPR) to ensure the privacy and security of donor and patient information.

Scalability: The architecture should support future expansion, accommodating multiple blood banks and users during emergency scenarios without degradation in performance.

Technology Stack: Should use modern, scalable technologies such as secure databases, and responsive frameworks.

2.6 USER DOCUMENTATION

The **Centralized Blood Bank Management System (CBBMS)** streamlines blood donation and distribution. It allows **donors** to register, schedule donations, and track their history. **Blood bank administrators** can manage blood inventory by adding, updating, and monitoring blood stocks. **Hospitals** can request blood units.

To use the system:

- **Donors:** Register by providing details such as name, contact information, and blood type. You can receive requests for your blood type.
- **Blood Banks:** Log in to update inventory and monitor blood stock levels.
- **Hospitals/Users:** Submit blood requests based on patient needs.

2.7 ASSUMPTIONS AND DEPENDENCIES

ASSUMPTIONS

Internet Access: All blood banks, hospitals, and users must have reliable internet access to utilize the system effectively.

User Competency: Users (donors, administrators, hospital staff) are assumed to have basic digital literacy and can navigate web.

Data Accuracy: Blood banks and hospitals will provide accurate and up-to-date information regarding blood stocks and requests.

Compliance: The system will adhere to relevant health and safety regulations, including data privacy laws (e.g., GDPR, HIPAA).

6

System Integration: Blood banks and hospitals are assumed to integrate with the system's APIs or data formats for smooth data exchange.

Scalability: The system is designed to scale to handle increasing data volumes and user loads, especially during emergencies.

DEPENDENCIES:

Internet Connectivity: Reliable internet access is essential for real-time data synchronization and communication between blood banks, hospitals, and donors.

Database Management System: Requires a robust database system (e.g., MySQL) for storing and managing blood inventory, donor information, and transaction records.

Compliance with Regulations: Must adhere to health and safety regulations (e.g., HIPAA, GDPR) for handling sensitive medical data and ensuring data privacy.

Server Infrastructure: Requires scalable server infrastructure to handle varying loads and ensure high availability, particularly during peak times.

3. EXTERNAL INTERFACE REQUIREMENTS

3.1 USER INTERFACE

Donor Interface: Allows registration, scheduling donations, viewing donation history, and receiving notification.

Blood Bank Administrator Interface: Manages blood inventory (add/update/remove), tracks expiry dates, and processes blood requests from hospitals.

Hospital Interface: Places blood requests.

System Administrator Interface: Oversees overall system operations, manages user accounts, and ensures system performance.

3.2 HARDWARE INTERFACES

The **hardware interface requirements** for the Blood Bank Management System (CBBMS) involve interaction with devices necessary for system operation. Key requirements :

- **Server:** A robust server with sufficient storage and processing power to host the CBBMS database and application.
- **Printers:** For printing blood donation reports, labels, and receipts.

- **Network Devices:** Routers and switches for reliable internet connectivity to support real-time data access and updates across multiple locations.
- **Backup Devices:** External hard drives.

3.3 SOFTWARE INTERFACES

The **Software Interfaces** requirements for the Blood Bank Management System (CBBMS) define how the system interacts with external software components. The system will interface with:

- **Database Management System (DBMS):** The CBBMS must integrate with a relational database (e.g., MySQL) for storing donor, recipient, and blood inventory data.
- **API Integration:** The system will provide RESTful APIs for communication with hospital management systems and other blood banks to share blood availability data.
- **Authentication Services:** Integration with third-party authentication systems for secure user access.

3.3 COMMUNICATION INTERFACES

HTTP/HTTPS

- **Purpose:** Web communication protocol for client-server interaction.

FTP/SFTP

- **Purpose:** File transfer protocol (secure).

4.SYSTEM FEATURES

Donor Management: Register donors, manage profiles.

Blood Inventory Management: Monitor blood stock levels and update availability in real-time.

Hospital Blood Requests: Hospitals can request blood.

Search & Notifications: Search for blood availability and receive notifications for donation requests making.

2.2 PRODUCT FUNCTIONS

Donor Registration and Management: Users can create donor profiles by providing personal details, blood type, and contact information.

Blood Inventory Management: Blood banks can add, update, and monitor blood stock, including blood type, quantity, and expiration date.

Hospital Blood Requests: Hospitals can search for specific blood types, place requests.

to ensure secure and appropriate system usage.

4.1 SYSTEM FEATURE 1

4.1.1 DESCRIPTION AND PRIORITY

1. Donor Registration and Management

- **Description:** The system will allow users to register as blood donors by providing personal details, contact information, blood type, medical history, and availability. The system will store and manage these details in a centralized database.
- **Priority:** High

2. Blood Request and Reservation

- **Description:** Hospitals and patients can request specific blood groups through the system.
- **Priority:** High

3. Blood Compatibility and Cross-Matching

- **Description:** The system will ensure that the requested blood units are compatible with the recipient's blood type. It will manage and store cross-matching test results to verify compatibility.
- **Priority:** Medium

4. Hospital/Clinic Registration and Management

- **Description:** The system will allow hospitals and clinics to register and access the centralized blood inventory.
- **Priority:** High
- **Priority:** Medium

5. User Roles and Permissions

- **Description:** The system will support different user roles (e.g., administrator, donor, hospital staff) with varying access permissions. Each role will have tailored access to specific functionalities based on their needs.
- **Priority:** Medium

6. Data Backup and Recovery

- **Description:** Regular backup and recovery procedures will be in place to prevent data loss and

- **Priority:** High

4.1.2 STIMULUS/RESPONSE SEQUENCES

Donor Registration & Management:

- **Stimulus:** User (admin or staff) inputs donor details (personal information, medical history, blood type).
- **Response:** System validates and stores donor data, assigns a unique donor ID, and generates confirmation notification.

Blood Request Handling:

- **Stimulus:** Hospital or individual requests blood.
- **Response:** System checks availability, processes the request, generates delivery details.

4.1.3 FUNCTIONAL REQUIREMENTS

Donor Registration and Management: The system allows for the registration of new donors, capturing their personal details, medical history, and eligibility status. It can update donor information and track their donation history.

Blood Inventory Management. It tracks inventory levels, blood expiration dates, and can issue alerts for low stock or expiring units.

Blood Request and Fulfillment: Hospitals and healthcare centers can request specific blood types through the system. The system processes requests, checks availability, and generates reports for blood delivery or pickup.

User Roles and Security: The system offers role-based access for different users like administrators, lab technicians, and donors to ensure data security.

5. OTHER NON-FUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE REQUIREMENTS

Response Time: The system should provide a response time of less than 2 seconds for user queries and data retrieval. Critical operations, such as blood request processing and donor registration, must complete within 5 seconds.

Scalability: The CBBMS should support up to 1,000 concurrent users without degradation in performance. It must be scalable to handle increasing loads by adding additional servers or resources as needed.

System Availability: The CBBMS must have an uptime of 99.9%, ensuring high availability.

5.2 SAFETY REQUIREMENTS

Data Protection:

- **Confidentiality:** Implement encryption protocols (e.g., AES-256) for sensitive data such as donor and patient information.
- **Access Control:** Enforce role-based access controls (RBAC) to ensure only authorized personnel can access or modify data.

System Integrity:

- **Data Backup:** Regular automated backups must be performed to prevent data loss due to system failures or breaches.
- **Data Validation:** Implement stringent data validation checks to prevent erroneous or malicious data entry.

Compliance:

- **Regulatory Standards:** Ensure compliance with relevant regulations such as HIPAA (Health Insurance Portability and Accountability Act) for data protection and blood bank standards.

Secure Facilities:

Data centers and server rooms must be physically secured with restricted access.

- **Environmental Controls:** Implement environmental controls (e.g., temperature and humidity monitoring) to protect hardware and blood storage systems.

Emergency Procedures:

- **Incident Response Plan:** Develop and maintain an incident response plan to address data breaches or system failures swiftly.
- **Disaster Recovery:** Establish and regularly test disaster recovery procedures to ensure continuity of service in the event of catastrophic events.

5.3 SECURITY REQUIREMENTS

Access Control: Role-based access to restrict sensitive data based on user roles (e.g., admin, donor, recipient).

Data Encryption: Encrypt data in transit and at rest to protect against unauthorized access.

Authentication and Authorization: Implement strong password policies and multi-factor authentication (MFA) for system access.

Data Integrity: Ensure data accuracy and protection against tampering or corruption through validation

5.4 SOFTWARE QUALITY ATTRIBUTES

Reliability: Ensures consistent system performance and data integrity, even under heavy load.

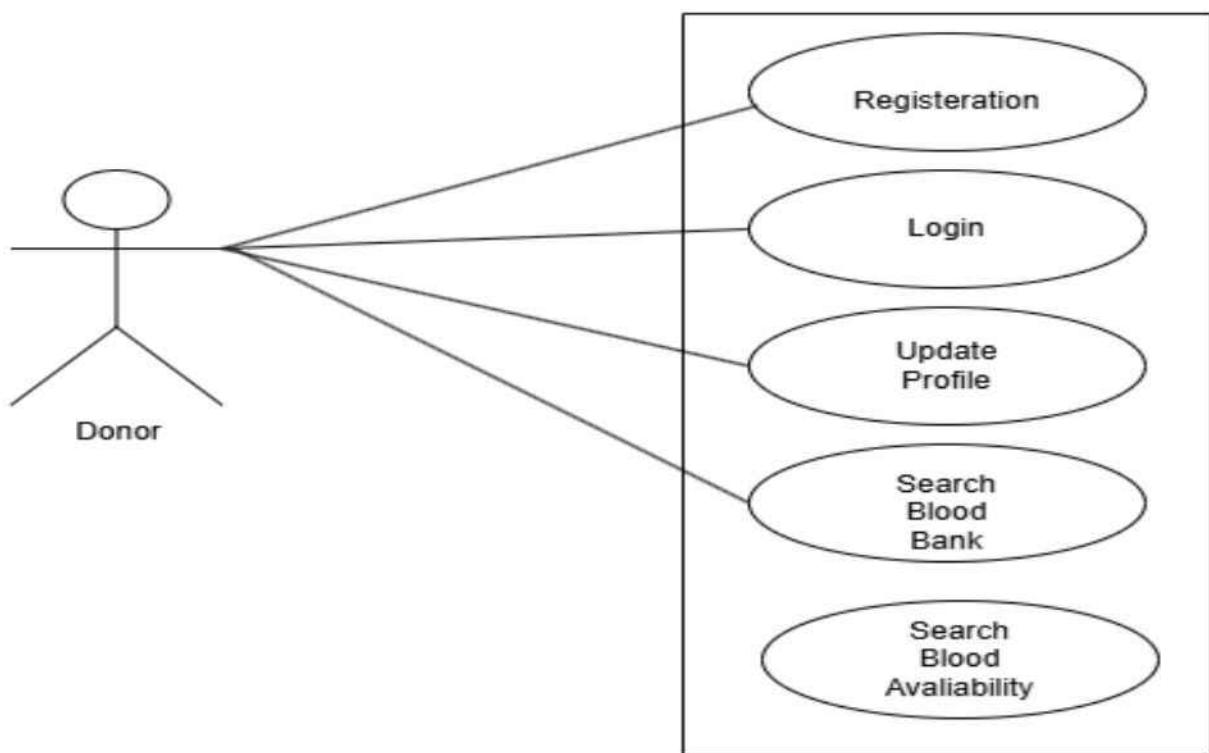
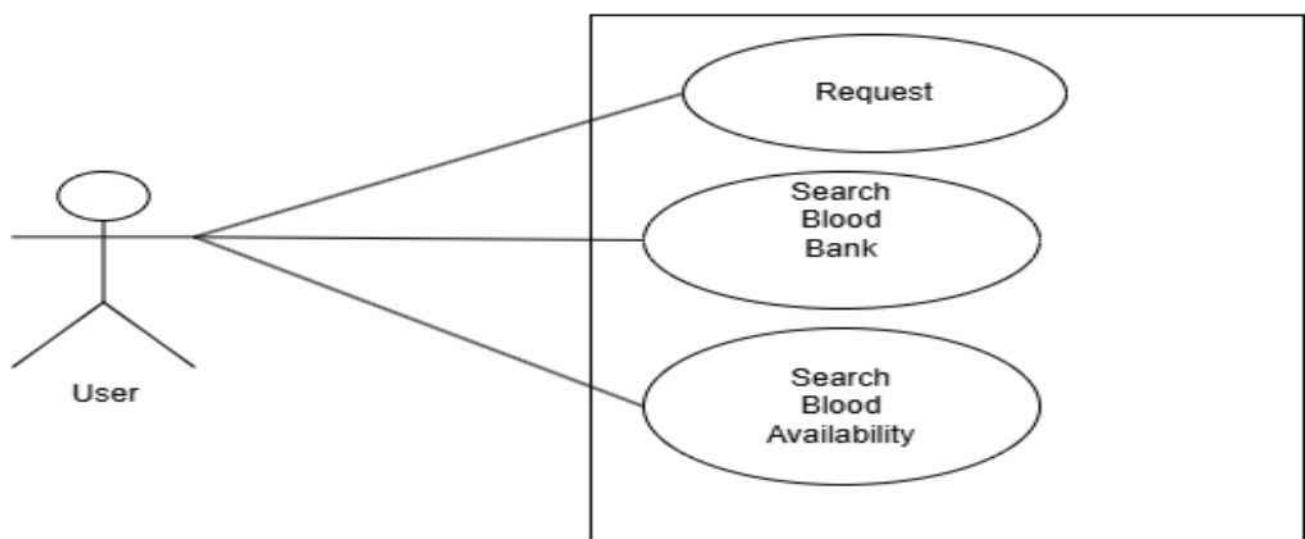
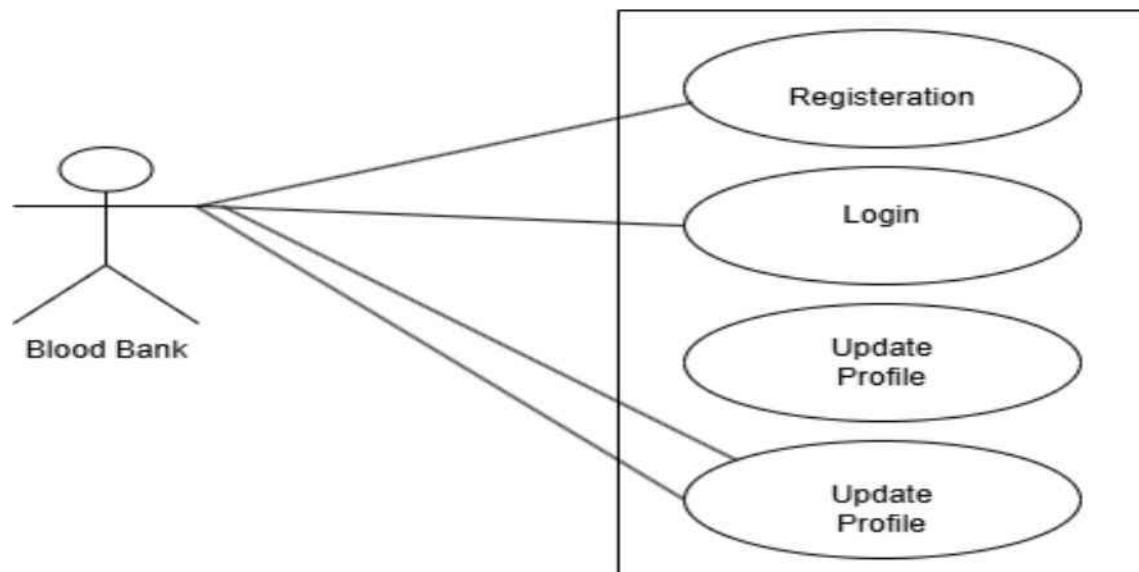
Availability: Provides high uptime to support 24/7 blood management and emergency needs.

Scalability: Accommodates growing data and user demands without performance degradation.

Security: Protects sensitive donor and patient information with robust access controls and encryption.

Usability: Offers an intuitive interface for users of varying technical skills.

Maintainability: Facilitates easy updates and bug fixes with modular design and clear documentation.



6 OTHER REQUIREMENTS: NILL

1. INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

Provide a Detailed Design Blueprint: The SDS outlines the system's architecture, modules, and components. It defines how the system will be constructed and integrated to meet the functional and non-functional requirements.

Ensure Clear Understanding among Stakeholders: By specifying the design and structure of the system, the SDS ensures that developers, testers, project managers, and other stakeholders have a shared understanding of how the system will work.

Serve as a Reference for Developers: The SDS serves as a guide for developers during the coding phase. It describes the system's components, database schema, API endpoints, and workflows to be followed in development.

Facilitate System Maintenance and Future Enhancements: The document provides a clear, structured overview of the system design, making it easier to maintain, troubleshoot, and expand upon the system in the future.

Support Consistent and Efficient Implementation: The SDS helps to ensure that the system is implemented in a consistent manner, adhering to the specified standards and best practices, reducing ambiguity, and minimizing errors.

Guide Testing and Validation: The design specification in the SDS document also serves as a reference point for testing teams to ensure that the system meets the required functionality and performance criteria.

1.2 SCOPE OF THE DEVELOPMENT PROJECT

Track and manage blood inventory across multiple blood banks.

Allow hospitals to request blood units based on patient requirements.

Enable donors to register and schedule donations.

Ensure compliance with medical and safety standards.

Provide real-time reports on blood stocks, donation schedules, and requests

Be accessible via web-based and mobile platforms for ease of access.

1.3 DEFINITION ACRONYMS AND ABBREVIATIONS

Definitions

- Blood Bank:** An organization or facility that collects, stores, and manages blood and blood products for use in medical treatments.

- **Donor:** An individual who voluntarily provides blood or blood components for donation.
- **Recipient:** A patient or individual who receives blood or blood products for medical treatment.
- **Inventory:** The collection of available blood types and quantities stored in the blood bank.
- **Request:** A formal demand for blood or blood products submitted by hospitals or clinics.
- **Allocation:** The process of assigning available blood to fulfill requests based on criteria such as type and urgency.

Acronyms

- **BBMS:** Blood Bank Management System
- **API:** Application Programming Interface
- **CRUD:** Create, Read, Update, Delete
- **UI:** User Interface
- **UX:** User Experience

Abbreviations

- **O+:** Blood Type O Positive
- **A-:** Blood Type A Negative
- **B+:** Blood Type B Positive
- **ABO:** A system for classifying blood based on the presence or absence of antigens A and B
- **RH:** Rhesus factor, which determines whether blood is positive or negative for the Rh antigen

1.4 REFERENCES

[A Blood Bank Management System | IEEE Conference Publication | IEEE Xplore](#)

[Blood Management System | IEEE Conference Publication | IEEE Xplore](#)

[\(PDF\) BLOOD MANAGEMENT SYSTEM \(researchgate.net\)](#)

[research paper 2.docx - Google Docs \(ijnrd.org\)](#)

1.5 OVERVIEW OF DOCUMENT

The Centralized Blood Bank Management System is designed to streamline and enhance the management of blood donations, inventory, and distribution across multiple blood banks. This system aims to improve the efficiency of blood collection, testing, storage, and allocation, ensuring timely and accurate blood supply to hospitals and clinics.

1. **Unified Inventory Management:** Centralize blood inventory tracking, including real-time updates and reduce wastage.
2. **Donor Management:** Maintain detailed records of donors, and facilitate appointment scheduling and communication.
3. **Request :** Efficiently handle blood requests from healthcare facilities, match them with available inventory.
4. **Security and User Management:** Implement robust security measures and manage user roles and permissions to safeguard data and ensure proper access controls.
5. **Integration:** Seamlessly interface with external systems such as Electronic Health Records (EHR) and national blood registries to enhance data sharing and compliance.

2. SYSTEM ARCHITECTURE AND DESCRIPTION

2.1 OVERVIEW OF MODULES/COMPONENTS

Frontend: Web-based interface for donors and blood bank staff.

Backend: RESTful API for managing data operations, secure data storage, and processing requests.

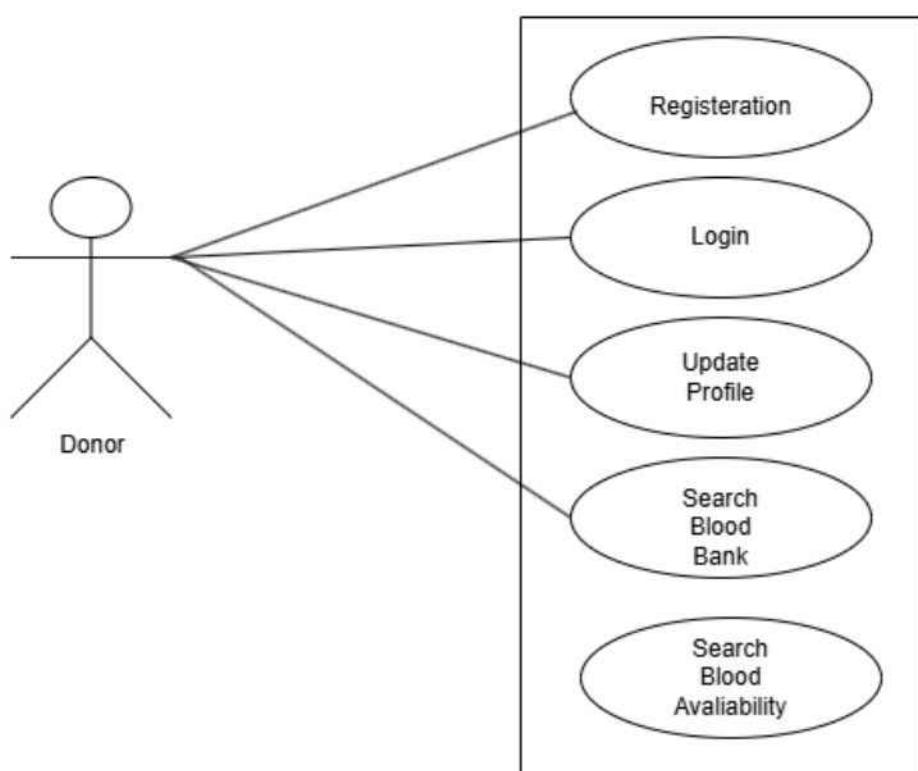
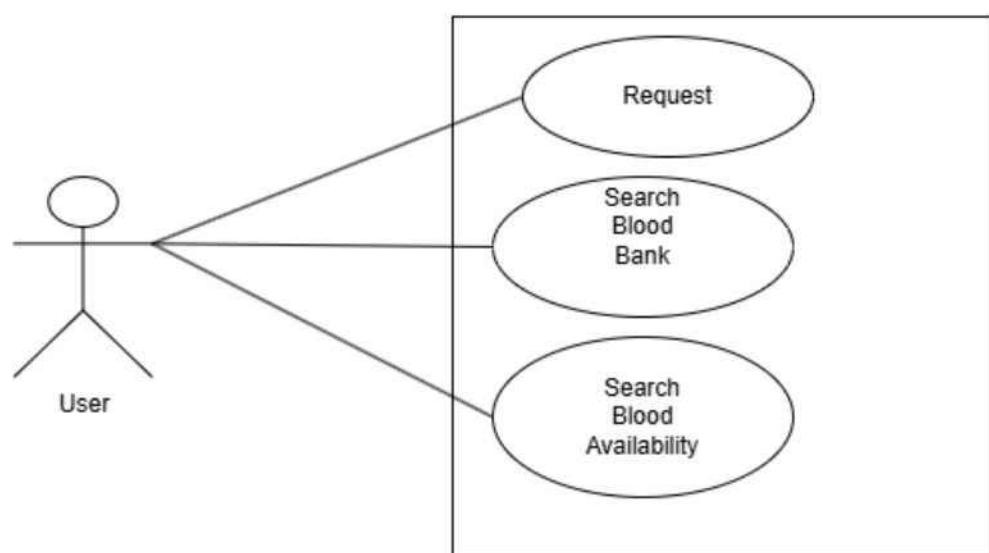
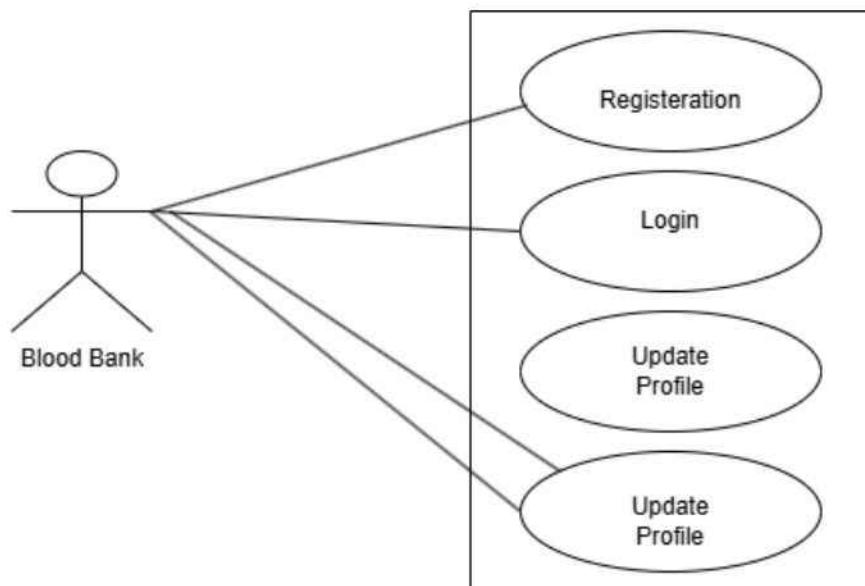
Database: Relational database (MySQL) for storing donor information, blood inventory, and transaction history.

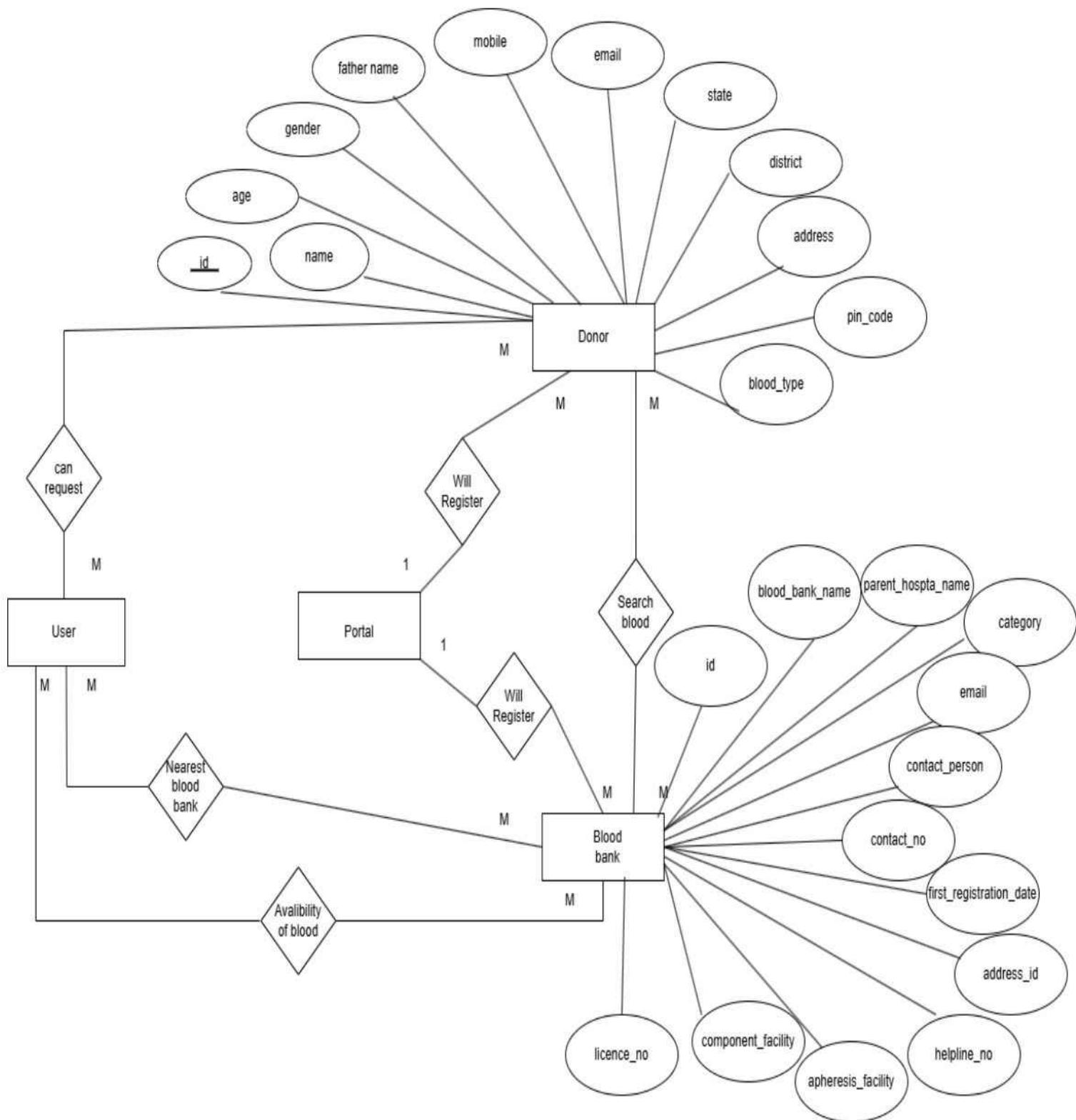
External Services: Integration with SMS/Email services for notifications, hospital management systems for requests.

2.2 STRUCTURE AND RELATIONSHIPS

USECASE DIAGRAM

18





BLOOD_COMPONENTS TABLE

Column Name	Data Type	Description
id	INT	(Primary key) Not Null
blood_bank_email	VARCHAR	(Foreign key) Not Null
blood_type	ENUM	Not Null
plasma_units	INT	Not Null
platelets_units	INT	Not Null
wbc_units	INT	Not Null
rbc_units	INT	Not Null

BLOOD TYPES

Column Name	Data Type	Description
bloodtype_id	INT	(Primary key) Not Null
bloodtype	VARCHAR	Not Null

BLOOD BANK TABLE

Column Name	Data Type	Description
id	INT	(primary key) Not Null
blood_bank_name	VARCHAR	Not Null
parent_hospital_name	VARCHAR	-
category	ENUM('Govt', 'Private')	Not Null
email	VARCHAR	Not Null (unique)
contact_person	VARCHAR	Not Null
contact_no	VARCHAR	Not Null
first_registration_date	DATE	-
licence_no	VARCHAR	Not Null(Unique)
Component_facility	ENUM('yes', 'No')	Not Null
apheresis_facility	ENUM('yes', 'No')	Not Null
helpline_no	INT	-
address_id	INT	Foreign key

BLOODBANK ACCOUNTS

Column Name	Data Type	Description
id	INT	(Primary key) Not Null
bloodbank_id	INT	(Foreign key) Not Null
email	VARCHAR	Not Null
password	VARCHAR	Not Null
created_at	TIMESTAMP	-

BLOODBANK ADDRESS

Column Name	Data Type	Description
id	INT	(Primary key) Not Null
state	VARCHAR	Not Null
district	VARCHAR	Not Null
city	VARCHAR	-
postal_address	TEXT	-

BLOOD STOCK

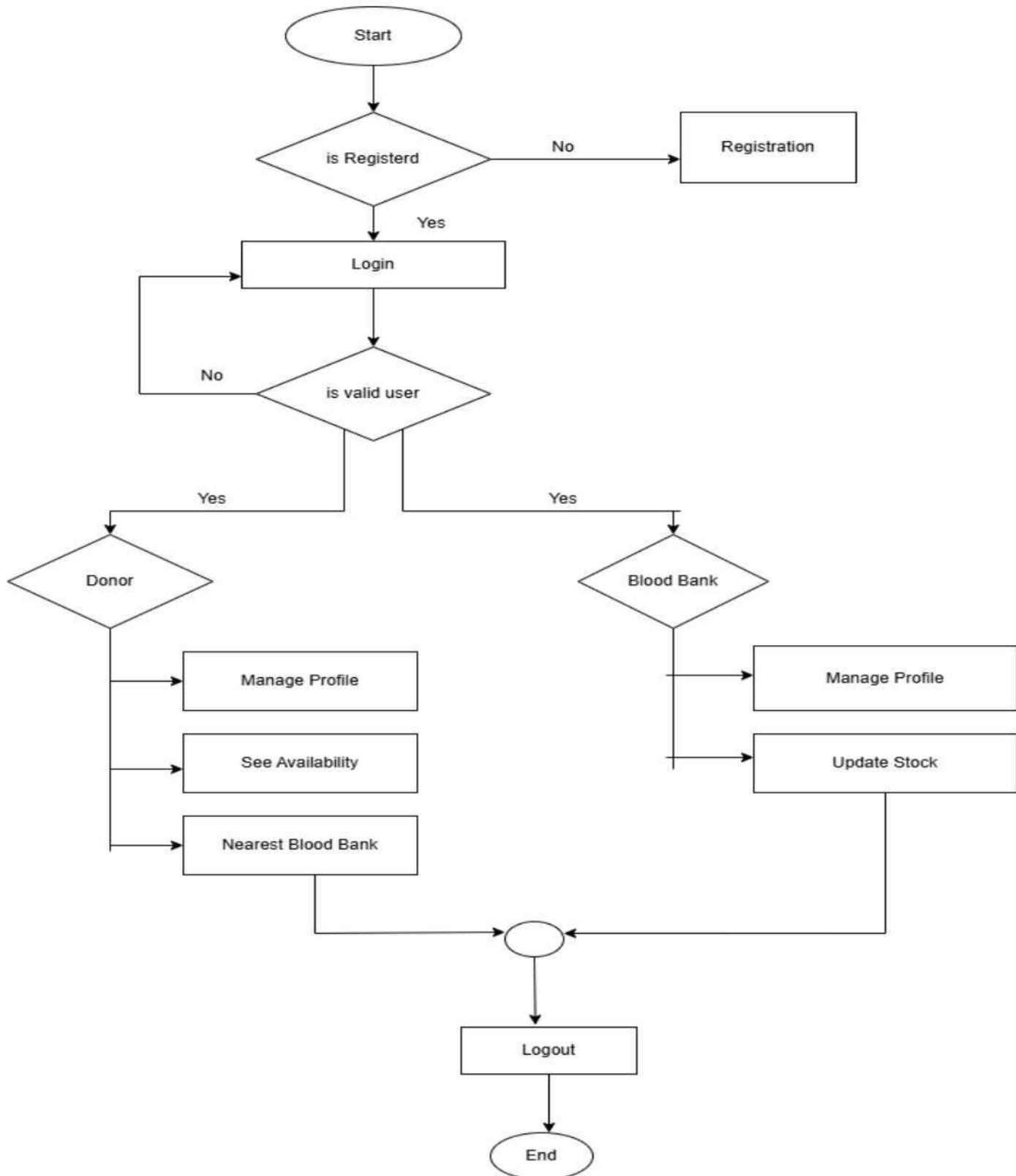
Column Name	Data Type	Description
BloodStockID	INT	(Primary key) Not Null
BloodBankID	INT	-
BloodType	ENUM('A+', 'A-', 'B')	Not Null
PlasmaUnits	INT	-
RedBloodCellsUnits	INT	-
WhiteBloodCellsUnits	INT	-
LastUpdated	TIMESTAMP	-

DONORLOGIN TABLE

Column Name	Data Type	Description
id	INT	(Primary key) Not Null
email	VARCHAR	(Foreign Key)Not Null (Unique)
password	VARCHAR	Not Null
reset_token	INT	-
created_at	TIMESTAMP	-

DONORS TABLE

Column Name	Data Type	Description
id	INT	(primary key) Not Null
name	VARCHAR	Not Null
age	INT	Not Null
gender	VARCHAR	Not Null
father_name	VARCHAR	-
mobile	VARCHAR	(Unique) Not Null
email	VARCHAR	(Unique) Not Null
state	VARCHAR	Not Null
district	VARCHAR	Not Null
address	TEXT	-
pin_code	VARCHAR	Not Null
blood_type	VARCHAR	Not Null



Complicated Navigation: If the interface is hard to navigate, users like donors, blood bank staff, or hospitals may struggle to find what they need. The design should be simple and easy to understand.

Too Much Information on One Screen: Showing too much data at once can overwhelm users, especially when managing things like donor information, blood stock, and hospital requests. The information should be organized in a way that's easy to follow.

Not Mobile-Friendly: Many users will want to access the system from their phones or tablets. If the design doesn't work well on smaller screens, it can frustrate mobile users. The system should be responsive, meaning it works well on any device.

Poor Data Entry Validation: If the system doesn't properly check for mistakes when users enter information, errors can happen. For example, incorrect donor details or wrong blood inventory data can lead to problems down the line.

Lack of Accessibility: The system should be designed for people of all abilities, including those with visual or physical disabilities. Following accessibility guidelines will make the system easier to use for everyone.

3.DETAILED DESCRIPTION OF COMPONENTS

3.1 COMPONENT TEMPLATE DESCRIPTION

Component 1 (Donor management)

Identification: Manages all aspects related to blood donors within the Blood Bank Management System.

Type: Core component

Purpose: To streamline donor registration, track donation history, and maintain donor health records.

Function: Facilitates donor sign-up, maintains donor profiles, manages eligibility and scheduling of donations, and ensures accurate record-keeping.

Subordinates: Includes donor registration forms, health screening modules, and scheduling systems.

Dependencies: Relies on medical history data, blood collection modules, and inventory management for up-to-date information and scheduling.

Interfaces:

- **User Interface:** Donors interact through web to register and manage profiles.
- **Admin Interface:** Administrators manage and review donor information.
- **Database Interface:** Stores donor records, health details, and donation history.

Resources: Requires a database for storing donor information, secure access controls, and

Processing: Handles registration, updates donor details, schedules donations, and checks eligibility. It processes data to maintain accurate records and generate alerts for upcoming donation opportunities.

Data: Includes personal details (name, contact info), medical history, donation records.

Component 2 (Inventory management)

Identification: Inventory Management Module.

Type: Core module responsible for managing the blood stock.

Purpose: To track, monitor, and manage blood units, ensuring the availability of different blood types and maintaining their quality and safety.

Function:

- Record and update blood stock levels.
- Monitor blood types (A, B, AB, O).
- Categorize blood units as available, quarantined.

Subordinates:

- Request Management (uses inventory data for blood requests).

Dependencies:

- Relies on the Blood Collection .

Resources: Database storing blood type, quantity, collection date.

Processing:

Updates during stock additions or removals.

Data: Blood group, quantity, collection, availability status.

Component 3 (Request Management)

Identification:

- Name: Request Management
- Module: Blood Bank Management System

Type: Core functional module handling blood requests from doctors/hospitals.

Subordinates:

- Inventory Management: Ensures blood stock availability.

- User Roles: users like doctors can request blood.

Resources:

- Data: Blood type, stock status, patient details.
- Users: Doctors, blood bank staff.

Component 4(Notification and Alerts)

Identification:

Module: Notifications and Alerts

System: Blood Bank Management System

Type: Functional Component

Function: Sends notifications to different users (donors, doctors, administrators) via email, SMS.

Dependencies:

Relies on data from the Inventory Management module.

Requires User Role Management for sending role-specific alerts.

Resources:

Data from blood inventory, donor records, request logs.

Communication tools (email).

4. REUSE AND RELATIONSHIPS TO OTHER PRODUCTS

Not Required!!!

5. DESIGN DECISIONS AND TRADEOFFS

Not Required!!!!

6.PSEUDOCODE FOR COMPONENTS

Not Available in this Document.

7.APPENDICES

Null.

CODING

Home page Frontened Code

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset=""UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>E-rakt Portal</title>
    <link
      href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;600&family=Poppins:wght@400;600&display=swap" rel="stylesheet">
    <style>
      /* Global styles*/
      body {
        font-family: 'Open Sans', sans-serif;
        margin: 0;
        padding: 0;
        background-color: #f9f9f9;
      }
      .header {
        display: flex;
        align-items: center;
        justify-content: center; /* Centers the heading horizontally */
        position: relative; /* Allows positioning of the logo */
        height: 80px;
        background-color: #e74c3c;
        color: white;
        padding: 0 20px;
        box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
      }
      .logo {
        position: absolute; /* Positions the logo independently */
        left: 20px; /* Aligns the logo to the leftmost corner */
      }
    </style>
  </head>
  <body>
    <div class="header">
      
      <h1>E-rakt Portal</h1>
    </div>
    <div>
      <p>Welcome to E-rakt Portal! Your one-stop solution for all your e-commerce needs. We offer a wide range of products across various categories, including Electronics, Home Appliances, Clothing, and more. Our mission is to provide you with a seamless shopping experience, backed by excellent customer service and competitive pricing. Explore our website today and discover why we're the go-to destination for online purchases. If you have any questions or need assistance, please don't hesitate to contact us. Happy shopping!</p>
    </div>
  </body>
</html>

```

```
.logo img {  
    height: 50px; /* Adjusts the logo size */  
}
```

```
.heading {  
    margin: 0;  
    font-size: 24px;  
    font-weight: bold;  
}
```

```
/* Navigation styles */
```

```
nav ul {  
    display: flex;  
    list-style: none;  
    gap: 15px;  
    margin: 0;  
    padding: 0;    }
```

```
nav ul li a {  
    text-decoration: none;  
    color: #333;  
    font-weight: bold;  
}
```

```
/* Navigation section*/
```

```
* {  
    padding: 0;  
    margin: 0;  
    font-family:'Poppins', sans-serif;  
}  
ul {  
    list-style:none;  
    background:#333;  
}  
ul li {
```

```
display: inline-block;
position: relative;
}

ul li a {
    display: block;
    padding: 10px 10px;
    color:#FFF;
    text-decoration: none;
    text-align: center;
    font-size: 20px;
}

ul li ul.dropdown li {
    display: block;
}

ul li ul.dropdown {
    width: 100%;
    background: #808080;
    position: absolute;
    z-index: 999;
    display: none;
}

ul li a:hover {
    background: #463f3f;
}

ul li:hover ul.dropdown {
    display: block;
}

/* Hero Section */

.hero {
    background-image: url('https://www.explorejeffersonpa.com/wp-content/uploads/sites/3/2024/04/blood-donation.jpg');
    background-size: cover;
    background-position: center;
    color: white;
    text-align: center;
```

```
padding: 100px 20px;  
}  
  
.hero h2 {  
    font-size: 48px;  
    font-family: 'Poppins', sans-serif;  
    margin-bottom: 20px;  
}  
  
.hero p {  
    font-size: 24px;  
    margin-bottom: 30px;  
}  
  
.hero .cta-button {  
    background-color: #e74c3c;  
    color: white;  
    padding: 15px 30px;  
    font-size: 18px;  
    text-decoration: none;  
    border-radius: 5px;  
}  
  
.hero .cta-button:hover {  
    background-color: #c0392b;  
}  
  
/* About Section */  
  
.about {  
  
    text-align: center;  
    padding: 60px 20px;  
}  
  
.about h2 {  
    font-size: 36px;  
    color: #c0392b;  
}  
  
.about p {  
    font-size: 18px;  
    color: #666;  
    margin-bottom: 30px;
```

```
}

/* container */

.container {
    display: flex;
    align-items: center;
    justify-content: space-between;
    gap: 700px;
}

.container img {
    width: 45%; /* Image ka size */
    max-width: 60%;
    border-radius: 10px;
}

.container video {
    width: 45%; /* Video ka size */
    max-width: 100%;
    border-radius: 10px;
}

/* Services Section */

.services {
    background-image: url('https://sbtcup.org/Icon/Spinner-1s-294px.gif');
    background-size: 50%;
    background-position: center;
    background-color: #f4f4f4;
    padding: 60px 20px;
}

.services h2 {
    text-align: center;
    font-size: 36px;
    color: #e74c3c;
    margin-bottom: 40px;
}

.service-item {
    display: flex;
    justify-content: space-around;
    gap: 20px;
}
```

```
}

.service-card {
    background-color: white;
    padding: 30px;
    border-radius: 10px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
    width: 30%;
    text-align: center;
}

.service-card h3 {
    font-size: 24px; color: #e74c3c;
    margin-bottom: 20px;
}

.service-card p {
    font-size: 16px;
    color: #666;
}

/* Contact Section */

.contact {
    text-align: center;
    padding: 60px 20px;
    background-color: #e74c3c;
    color: white;
}

.contact h2 {
    font-size: 36px;
    margin-bottom: 30px;
}

.contact p {
    font-size: 18px;
    margin-bottom: 20px;
}

.contact a {
    background-color: white;
    color: #e74c3c;
    padding: 15px 30px;
```

```
font-size: 18px;  
text-decoration: none;  
border-radius: 5px;  
}  
.contact a:hover {  
background-color: #f9f9f9;  
}  
/* Footer */  
  
footer {  
background-color: #333;  
color: white;  
text-align: center;  
padding: 20px 0;  
}  
.footer-nav ul {  
list-style-type:none;  
padding:0;  
margin:0;  
display:flex;  
justify-content:center;  
gap: 30px;  
}  
.footer-nav li {  
display: inline-block;  
}  
.footer-link {  
text-decoration: none;  
color: white;  
font-size: 16px;  
padding: 10px 20px;  
display:inline-block;  
border-radius: 5px;  
transition: background-color 0.3s ease;  
}  
.footer-link:hover {  
background-color: #007bff;
```

```
}

</style>

</head>

<body>

<header class="header">

<div class="logo">
    
</div>

<h1 class="heading">Central Blood Bank</h1>

</header>

<ul>
    <li><a href="home.html">Home</a></li>
    <li>
        <a href="#">About us</a>
        <ul class="dropdown">
            <li><a href="abouttt.html">About us</a></li>
            <li><a href="factt.html">Facts</a></li>
            <li><a href="gallery.html">Photo gallery</a></li>
            <li><a href="videogallery.html">Video gallery</a></li>
            <li><a href="Looking.html">Know Us</a></li>
            <li><a href="FAQS1.html">FAQs</a></li>
        </ul>
    </li>
    <li>
        <a href="#">Looking for blood</a>
        <ul class="dropdown">
            <li><a href="BloodAvailability1.html">Blood Availability</a></li>
            <li><a href="bloodbankdirectory.html">Blood Bank Directory</a></li>
        </ul>
    </li>
</ul>
```

```

</li>
<li>
  <a href="#">Donor Details</a>
  <ul class="dropdown">
    <li><a href="register1.html">Donor register</a></li>
    <li><a href="donorlogin.html">Donor login</a></li>

  </ul>
</li>
<li><a href="request.html">Request Blood</a></li>
<li><a href="#">Blood Bank Login</a>
  <ul class="dropdown">
    <li><a href="eraktportallogin.html">E-rakt Portal Login</a></li>
    <li><a href="bloodbankregistration.html ">Add Your Blood Bank</a></li>
  </ul>
</li>

</ul>

<section class="hero">
  <h2>Save a Life, Donate Blood</h2>
  <p><i><marquee behaviour="alternate" direction="right">Be a hero. Your donation can save lives today!</marquee></i></p>
  <a href="donorlogin3.html" class="cta-button">Donate Now</a>
</section>

<section class="about">
  <h2><u>Donation Chart</u></h2>
  <hr>
  
</section>

  <p style="text-align:center ; font-size:40px; color: #e74c3c"><b><u>Things To Remember While Donating Blood</u></b><p>

<section class="container">
  <p style="background-color:rgb(48, 42, 42)">
    
    <video autoplay muted loop>

```

Your browser does not support the video tag

</video></p>

</section>

<section class="services">

<h2><u>Our Services</u></h2>

<hr>

<div class="service-item">

<div class="service-card">

<h3>Blood Donation</h3>

<p>Join us in donating blood to save lives and help those in critical need.</p>

</div>

<div class="service-card">

<h3>Emergency Support</h3>

<p>We provide immediate blood supply for emergency cases and surgeries.</p>

</div>

<div class="service-card">

<h3>Community Outreach</h3>

<p>We educate communities on the importance of regular blood donation.</p>

</div>

</div>

</section>

<section class="contact" id="contact">

<h2>Contact Us</h2>

<p>Have questions or want to learn more about blood donation? Get in touch with us!</p>

<p>Contact no.-0000000000</p>

<p>Address-bbbbbbbb</p>

Email Us

</section>

<footer>

<nav class="footer-nav">

```

<ul>
    <li><a href="home.html" class="footer-link">Home</a></li>
    <li><a href="abouttt.html" class="footer-link">About us</a></li>
    <li><a href="gallery.html" class="footer-link">Photo gallery</a></li>
    <li><a href="videogallery.html" class="footer-link">Video gallery</a></li>
    <li><a href="BloodAvailability1.html" class="footer-link">Blood Availability</a></li>
    <li><a href="bloodbankdirectory.html" class="footer-link">Blood Bank Directory</a></li>
    <li><a href="donorlogin3.html" class="footer-link">Donor Login</a></li>
    <li><a href="request.html" class="footer-link">Request blood</a></li>
    <li><a href="eraktportallogin.html" class="footer-link">e raktportallogin</a></li>

</ul>
<hr>
<p>&copy; 2025 Central Blood Bank | All Rights Reserved</p>
</nav>

</footer>
</body>
</html>

```

(Frontened Code for Blood Bank Directory)

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Nearest Blood Bank Search</title>
    <link rel="stylesheet" href="Bloodbankdirectory.css">
</head>
<body background="bloodbankdirectory.jpg">
<header class="header">
    <div class="logo">
        
    </div>
    <h1 class="heading">Central Blood Bank</h1>
</header>

```

```

<ul>
  <li><a href="home.html">Home</a></li>
  <li>
    <a href="#">About us</a>
    <ul class="dropdown">
      <li><a href="abouttt.html">About us</a></li>
      <li><a href="factt.html">Facts</a></li>
      <li><a href="gallery.html">Photo gallery</a></li>
      <li><a href="videogallery.html">Video gallery</a></li>
      <li><a href="Looking.html">Know Us</a></li>
      <li><a href="FAQS1.html">FAQs</a></li>

    </ul>
  </li>
  <li>
    <a href="#">Looking for blood</a>
    <ul class="dropdown">
      <li><a href="BloodAvailability1.html">Blood Availability</a></li>
      <li><a href="bloodbankdirectory.html">Blood Bank Directory</a></li>
    </ul>
  </li>
  <li>
    <a href="#">Donor Details</a>
    <ul class="dropdown">
      <li><a href="register1.html">Donor register</a></li>
      <li><a href="donorlogin.html">Donor login</a></li>

    </ul>
  </li>
  <li><a href="request.html">Request Blood</a></li>
  <li><a href="#">Blood Bank Login</a>
    <ul class="dropdown">
      <li><a href="eraktportallogin.html">E-rakt Portal Login</a></li>
      <li><a href="bloodbankregistration.html ">Add Your Blood Bank</a></li>
    </ul>
  </li>
</ul>

```

```
<div class="container">
  <h1>Nearest Blood Bank (BB) / Blood Storage Unit (BSU)</h1>
  <div class="search-section">
    <div class="dropdowns">
      <select id="state" class="dropdown" onchange="populateDistricts()">
        <option value="">Select State</option>
        <option value="Uttar Pradesh">Uttar Pradesh</option>
        <option value="Maharashtra">Maharashtra</option>
        <option value="Rajasthan">Rajasthan</option>
        <option value="Bihar">Bihar</option>
        <option value="Karnataka">Karnataka</option>
      </select>
      <select id="district" class="dropdown">
        <option value="">Select District</option>
      </select>
      <input type="text" id="name" placeholder="Blood Bank or Hospital Name" class="dropdown">
      <button id="search-btn" class="btn">Search</button>
    </div>
    <div class="geolocation-warning" style="display: none;">
      User denied the request for Geolocation.
    </div>
  </div>

  <div class="legend">
    <label>
      <input type="checkbox" id="govtBloodBanks">
      <span>Government Blood Banks</span>
    </label>
    <label>
      <input type="checkbox" id="otherBloodBanks">
      <span>Other Blood Banks</span>
    </label>
  </div>

  <div class="result-section">
    <div class="show-entries">
```

```
<label for="entries">Show
<select id="entries">
  <option value="4">4</option>
  <option value="8">8</option>
  <option value="12">12</option>
</select>
entries
</label>
</div>

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Name</th>
<th>Address</th>
<th>Phone</th>
<th>Email</th>
<th>Category</th>
<th>Component Facility</th>
<th>Apheresis Facility</th>
</tr>
</thead>
<tbody id="table-body">
<tr>
<td colspan="8">No data available in table</td>
</tr>
</tbody>
</table>

</div>
</div>
</div>

<footer>
```

</footer>

<script>

```
function populateDistricts() {  
    const stateDistricts = {  
        "Uttar Pradesh": ["Lucknow", "Varanasi", "Agra"],  
        "Maharashtra": ["Mumbai", "Pune", "Nagpur"],  
        "Rajasthan": ["Jaipur", "Udaipur", "Jodhpur"],  
        "Bihar": ["Patna", "Gaya", "Muzaffarpur"],  
        "Karnataka": ["Bangalore", "Mysore", "Hubli"]  
    };  
}
```

```
let state = document.getElementById("state").value;
```

```
let districtDropdown = document.getElementById("district");
```

```
districtDropdown.innerHTML = '<option value="">Select District</option>';
```

```
if (stateDistricts[state]) {  
    stateDistricts[state].forEach(district => {  
        let option = document.createElement("option");  
        option.value = district;  
        option.textContent = district;  
        districtDropdown.appendChild(option);  
    });  
}  
}  
</script>
```

```
<script src="directoryscript1.js"></script>
```

```
</body>
```

```
</html>
```

```
# CSS
```

```
/* General Styles */
```

```
body {
```

```
    font-family: 'Open Sans', sans-serif;
```

```
margin: 0;  
padding: 0;  
background-color: #f9f9f9;  
}  
.header {  
display: flex;  
align-items: center;  
justify-content: center; /* Centers the heading horizontally */  
position: relative; /* Allows positioning of the logo */  
height: 80px;  
background-color: #e74c3c;  
color: white;  
padding: 0 20px;  
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
}
```

```
.logo {  
position: absolute; /* Positions the logo independently */  
left: 20px; /* Aligns the logo to the leftmost corner */  
}
```

```
.logo img {  
height: 50px; /* Adjusts the logo size */  
}
```

```
.heading {  
margin: 0;  
font-size: 24px;  
font-weight: bold;  
}  
/* Navigation styles */
```

```
nav ul {  
display: flex;  
list-style: none;  
gap: 15px;  
margin: 0;
```

```
padding: 0;      }

nav ul li a {
    text-decoration: none;
    color: #333;
    font-weight: bold;
}

/* Navigation section*/
* {
    padding: 0;
    margin: 0;
    font-family:'Poppins', sans-serif;
}
ul {
    list-style:none;
    background:#333;
}
ul li {
    display: inline-block;
    position: relative;
}
ul li a {
    display: block;
    padding: 10px 10px;
    color:#FFF;
    text-decoration: none;
    text-align: center;
    font-size: 20px;
}
ul li ul.dropdown li {
    display: block;
}
ul li ul.dropdown {
    width: 100%;
    background: #808080;
```

```
position: absolute;  
z-index: 999;  
display: none;  
}  
  
ul li a:hover {  
background: #463f3f;  
}  
  
ul li:hover ul.dropdown {  
display: block;  
}  
  
/* General Styles */  
  
body {  
font-family: Arial, sans-serif;  
margin: 0;  
padding: 0;  
background-color: #f9f9f9;  
}
```

```
.container {  
max-width: 1200px;  
margin: 30px auto;  
background-color: #fff;  
padding: 20px;  
border-radius: 10px;  
box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);  
}
```

```
h1 {  
text-align: center;  
color: #b22222;  
margin-bottom: 20px;  
}
```

```
/* Search Section */  
  
.search-section {  
margin-bottom: 20px;
```

```
}
```

```
.dropdowns {
```

```
    display: flex;
```

```
    gap: 10px;
```

```
    justify-content: center;
```

```
    flex-wrap: wrap;
```

```
}
```

```
.dropdown {
```

```
    padding: 10px;
```

```
    font-size: 16px;
```

```
    border: 1px solid #ccc;
```

```
    border-radius: 4px;
```

```
    min-width: 200px;
```

```
}
```

```
.btn {
```

```
    padding: 10px 20px;
```

```
    font-size: 16px;
```

```
    border: none;
```

```
    background-color: #b22222;
```

```
    color: white;
```

```
    border-radius: 4px;
```

```
    cursor: pointer;
```

```
}
```

```
.btn:hover {
```

```
    background-color: #a11d1d;
```

```
}
```

```
.geolocation-warning {
```

```
    text-align: center;
```

```
    color: #555;
```

```
    font-size: 14px;
```

```
    margin-top: 10px;
```

```
}
```

```
/* Legend */  
.legend {  
    display: flex;  
    justify-content: center;  
    gap: 20px;  
    margin-bottom: 10px;  
}  
  
.legend input {  
    margin-right: 5px;  
}  
  
/* Results Section */  
.result-section {  
    margin-top: 20px;  
}  
  
.show-entries {  
    text-align: left;  
    margin-bottom: 10px;  
}  
  
table {  
    width: 100%;  
    border-collapse: collapse;  
}  
  
thead {  
    background-color: #b22222;  
    color: white;  
}  
  
th, td {  
    padding: 10px;  
    border: 1px solid #ddd;  
    text-align: center;  
}
```

```
tbody tr:nth-child(even) {  
    background-color: #f9f9f9;  
}  
  
.pagination {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    margin-top: 10px;  
}  
  
.pagination-buttons button {  
    padding: 5px 10px;  
    font-size: 14px;  
    background-color: #b22222;  
    color: white;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
}  
  
.pagination-buttons button:disabled {  
    background-color: #ccc;  
    cursor: not-allowed;  
}  
/* Footer */  
footer {  
    background-color: #333;  
    color: white;  
    text-align: center;  
    padding: 10px;  
    position: fixed;  
    bottom: 0;  
    width: 100%;
```

```
}
```

```
javascript
```

```
document.getElementById('searchBtn').addEventListener('click', () => {  
    const state = document.getElementById('state').value;  
    const district = document.getElementById('district').value;  
    const bloodGroup = document.getElementById('bloodGroup').value;  
    const bloodType = document.getElementById('bloodType').value;
```

```
if (!state && !district) {
```

```
    alert('Please select a state or district to search.');
```

```
    return;
```

```
}
```

```
// Sample data
```

```
const sampleResults = [
```

```
{
```

```
    id: 1,
```

```
    bloodBank: 'City Blood Bank',
```

```
    category: 'Government',
```

```
    availability: 'Available',
```

```
    lastUpdated: '2025-01-16',
```

```
    type: 'Whole Blood',
```

```
},
```

```
{
```

```
    id: 2,
```

```
    bloodBank: 'Global Blood Center',
```

```
    category: 'Private',
```

```
    availability: 'Limited',
```

```
    lastUpdated: '2025-01-15',
```

```
    type: 'Plasma',
```

```
},
```

```
];
```

```
const resultBody = document.getElementById('resultBody');
```

```
resultBody.innerHTML = '';
```

```

sampleResults.forEach((result, index) => {
  const row = document.createElement('tr');
  row.innerHTML =
    `<td>${index + 1}</td>
    <td>${result.bloodBank}</td>
    <td>${result.category}</td>
    <td>${result.availability}</td>
    <td>${result.lastUpdated}</td>
    <td>${result.type}</td>
  `;
  resultBody.appendChild(row);
});

});

```

(Backened Code for blood bank)

Db.js

```

import mysql from 'mysql2/promise';
const pool = mysql.createPool({
  host: 'localhost',
  port: '3306',
  user: 'root',
  password: 'AaBbCcflyhigh123',
  database: 'deekshadb1',
  // Adjust connectionLimit as needed
  connectionLimit: 10
});
pool.getConnection()
  .then(() => {
    console.log('Connected to MySQL Database (Pool).');
  })
  .catch((error) => {
    console.error('Database connection failed:', error);
    throw error;
  });
export default pool;

```

Backened code for blood bank directory controllers file

```

import { getBloodBankDirectories } from "../models/bloodbankdirectoryModel.js";
export const fetchBloodBankDirectories = async (req, res) => {
  try {
    const { state, district, searchName } = req.query;

    if (!state || !district) {
      return res.status(400).json({ message: "State and district are required." });
    }

    const bloodBanks = await getBloodBankDirectories(state, district, searchName);
    res.json(bloodBanks);
  } catch (error) {
    res.status(500).json({ message: "Internal Server Error", error });
  }
};

```

Backened code for blood bank directory models file

```

import pool from "../config/db.js";
export const getBloodBankDirectories = async (state, district, searchName) => {
  try {
    let query = `

      SELECT
        b.id, b.blood_bank_name, b.parent_hospital_name, b.category, b.email,
        b.contact_person, b.contact_no, b.first_registration_date, b.licence_no,
        b.component_facility, b.apheresis_facility, b.helpline_no,
        a.state, a.district, a.city, a.postal_address

      FROM bloodbank b
      JOIN bloodbankaddress a ON b.address_id = a.id
      WHERE a.state = ? AND a.district = ?

    `;

    let params = [state, district];

    if (searchName) {
      query += ` AND (b.blood_bank_name LIKE ? OR b.parent_hospital_name LIKE ?)`;
      params.push(`%${searchName}%`, `%${searchName}%`);
    }
  }
}

```

```

const [rows] = await pool.query(query, params);
return rows;
} catch (error) {
  console.error("Database Query Error:", error);
  throw error;
}
};


```

Backened code for blood bank directory routes file

```

import express from "express";
import { fetchBloodBankDirectories } from "../controllers/bloodbankdirectoryController.js";
const router = express.Router();
router.get("/bloodbankdirectory", fetchBloodBankDirectories);
export default router;


```

Backend Code for JS File

```

import express from 'express';
import cors from 'cors';
import bloodBankRoutes from './routes/bloodBankRoutes.js';
import authRoutes from './routes/authRoutes.js';
import donorRoutes from './routes/donorRoutes.js';
import bloodbankdirectoryRoutes from './routes/bloodbankdirectoryRoutes.js';
import donorAuthRoutes from './routes/donorAuthRoutes.js';
import donorProfileRoutes from './routes/donorProfileRoutes.js';
import bloodbankProfileRoutes from './routes/bloodbankProfileRoutes.js';
import bloodComponentRoutes from './routes/bloodComponentRoutes.js';

import session from "express-session";
import pool from './config/db.js'; // The pool import

const app = express();

// Middleware
app.use(cors());
app.use(express.json()); // For parsing JSON bodies
app.use(session({ secret: "supersecret", resave: false, saveUninitialized: true }));


```

```
app.use('/bloodbank', bloodBankRoutes);
app.use("/api/auth", authRoutes);
app.use("/api", bloodbankdirectoryRoutes);
app.use("/api/donorAuth", donorAuthRoutes);
app.use('/', donorProfileRoutes);
app.use("/bloodbankprofile", bloodbankProfileRoutes);
app.use("/bloodcomponents", bloodComponentRoutes);
```

```
//app.use('/api/donors', donorRoutes);
```

```
app.use('/donors', donorRoutes);
```

```
// Optional: Test the pool connection on server startup
```

```
(async () => {
  try {
    // Make a quick test query to ensure the pool is connected
    const [rows] = await pool.query('SELECT 1 + 1 AS result');
    console.log('MySQL Pool Connected Successfully. Test result:', rows[0].result);
  }
});
```

```
// Start the server only after a successful test
```

```
const PORT = process.env.PORT || 5000;
const HOST = process.env.HOST || '172.20.133.238'; // Set Static IP Address
```

```
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
```

```
} catch (error) {
  console.error('Failed to connect to DB:', error);
  process.exit(1); // Exit process with failure
})());
```

1. Sample Test Cases

Test Case ID	Feature	Test Input	Expected Output	Actual Output	Status
TC_01	User Registration	Valid data	User registered successfully	User registered successfully	Pass
TC_02	Login	Invalid password	Show error message	Error displayed	message Pass
TC_03	Blood Search	B+ group	List of blood banks with B+	Correct list shown	Pass
TC_04	Blood Request	Request to bank with stock = 0	Show “Blood available”	“Blood not available” shown	Pass
TC_05	Form Submission (Donor)	Empty number	contact Show error	validation Error shown	Pass
TC_06	Database Connectivity	Form submitted	Data saved in database	Data saved	Pass

2. Bugs Found and Fixes

Bug ID	Description	Status
BUG_01	Login accepted empty password	Fixed: Added validation
BUG_02	Blood search not case-insensitive	Fixed: Used LOWER() in SQL query
BUG_03	Donor age not validating correctly	Fixed: Added age range validation

3. Tools Used

- **Browser Developer Tools** – For HTML/CSS debugging
- **Manual Testing** – All modules tested by hand with multiple scenarios



Donation Chart



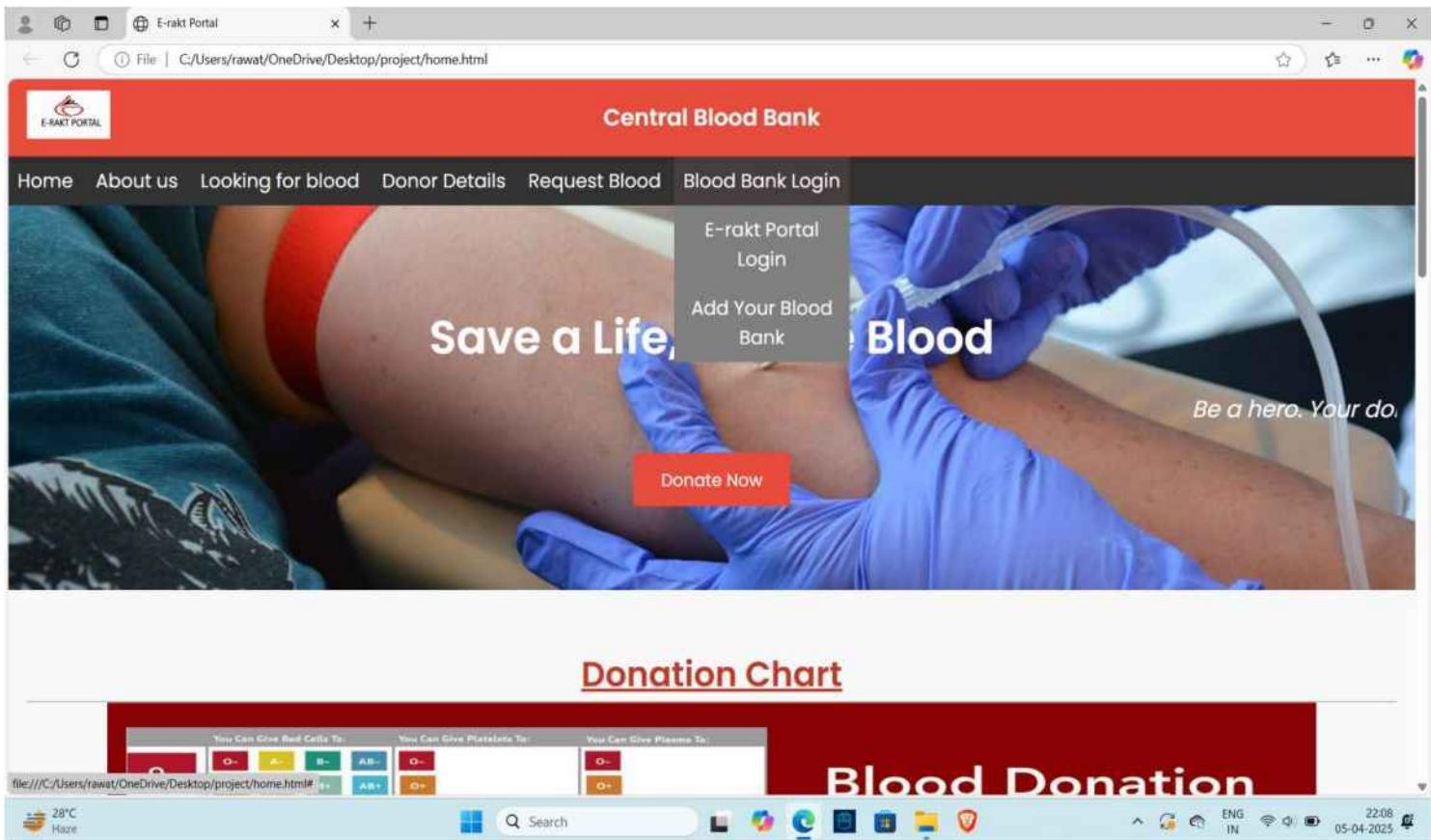
Blood Donation



Donation Chart



Blood Donation



APPENDICES

NILL

REFERENCES

[A Blood Bank Management System | IEEE Conference Publication | IEEE Xplore](#)

[Blood Management System | IEEE Conference Publication | IEEE Xplore](#)

[\(PDF\) BLOOD MANAGEMENT SYSTEM \(researchgate.net\)](#)

[research paper 2.docx - Google Docs \(ijnrd.org\)](#)