

Project Report
on
Life Line Donor's



**Ganpat
University**
॥ विद्यया समाजोत्कर्षः ॥

**U.V. Patel
College of
Engineering**



at
U. V. Patel College of Engineering

Project Guide :
Prof. Avnesh Joshi

Prepared By:
Mr. Setu Patel (22012011090)
Mr. Ramendra Rao (22012021081)
Mr. Aryan Soni (22012021099)
Mr. Vasu Viroja (22012021026)

**B.Tech Semester VI
Information Technology**

January-June 2025

Submitted to,
Department of Information Technology
U.V. Patel College of Engineering
Ganpat University, Ganpat Vidyanagar, Mehsana - 384012

U.V. PATEL COLLEGE OF ENGINEERING



**Ganpat
University**

॥ विद्या समाजोत्कर्षः ॥

U.V. Patel
College of
Engineering



17/05/25

C E R T I F I C A T E

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr. Setu Patel student of **B.Tech. Semester VI (Computer Engineering)** has completed his full semester Capstone Project-II titled "**Life Line Donor's**" satisfactorily in partial fulfillment of the requirement of Bachelor of Technology degree of Computer Engineering at Ganpat University, Ganpat Vidyanagar, Mehsana in the year 2024-2025.

Project Guide

Sign

Prof. Avnesh Joshi

Dr. Paresh M. Solanki

Head, Computer Engineering

U.V. PATEL COLLEGE OF ENGINEERING



**Ganpat
University**
॥ विद्यया समाजोत्कर्षः ॥

U.V. Patel
College of
Engineering



17/05/25

C E R T I F I C A T E

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr. Ramendra Rao of **B.Tech. Semester VI (Information Technology)** has completed his full semester Capstone Project-II titled "**Life Line Donor's**" satisfactorily in partial fulfillment of the requirement of Bachelor of Technology degree of Information Technology at Ganpat University, Ganpat Vidyanagar, Mehsana in the year 2024-2025.

Project Guide

Sign

Prof. Avnesh Joshi

Dr. Devang S. Pandya

Head, Information Technology

U.V. PATEL COLLEGE OF ENGINEERING



**Ganpat
University**

॥ विद्या समाजोत्कर्षः ॥

**U.V. Patel
College of
Engineering**



17/05/25

C E R T I F I C A T E

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr. Aryan Soni student of **B.Tech. Semester VI (Information Technology)** has completed his full semester Capstone Project-II titled “**Life Line Donor’s**” satisfactorily in partial fulfillment of the requirement of Bachelor of Technology degree of Information Technology at Ganpat University, Ganpat Vidyanagar, Mehsana in the year 2024-2025.

Project Guide

Sign
Prof. Avnesh Joshi

Dr. Devang S. Pandya
Head, Information Technology

U.V. PATEL COLLEGE OF ENGINEERING



**Ganpat
University**

॥ विद्या समाजोत्कर्षः ॥

**U.V. Patel
College of
Engineering**



17/05/25

C E R T I F I C A T E

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr. Vasu Viroja student of **B.Tech. Semester VI (Information Technology)** has completed his full semester Capstone Project-II titled “**Life Line Donor’s**” satisfactorily in partial fulfillment of the requirement of Bachelor of Technology degree of Information Technology at Ganpat University, Ganpat Vidyanagar, Mehsana in the year 2024-2025.

Project Guide

Sign

Prof. Avnesh Joshi

Dr. Devang S. Pandya

Head, Information Technology

ACKNOWLEDGEMENT

It is a genuine pleasure to express Our profound gratitude and deep regards to our guide "**Prof. Avnesh Joshi**" for his exemplary guidance, monitoring and constant encouragement. We would like to express our special thanks to GUNI who gave us the golden opportunity to do this wonderful project on the topic "**Life Line Donor's**", which helped us in doing a lot of research and we came to know about so many new things.

Very special thanks to the entire professors especially **Prof. Avnesh Joshi**, who helped us to create such a system. We are very grateful to **Ganpat University** and to our college **Department of Computer Engineering** in which we are going to submit our project.

With Sincere regard from,

Setu Patel

Ramendra Rao

Aryan Soni

Vasu Viroja

ABSTRACT

The Blood Management System project is designed to facilitate the entire blood donation process, from donor registration to blood distribution. It provides tools for blood donors to manage their profiles, schedule donations, and track their history. For hospitals and blood banks, it offers efficient management of blood inventory, real-time tracking of demand and availability, and coordination of transfusions. This system improves communication between donors, blood banks, and hospitals, ensuring timely access to blood for patients in need while optimizing resource management for healthcare providers.

In a world where the demand for a safe and timely blood supply remains critical, the need for an efficient **Life Line Donor's** becomes indispensable. Therefore, we have prepared such a project which will further improve the blood donation process.

Our project is to create a user-friendly Web application solution, That facilitates the efficient management of blood operations. The system aims to enhance transparency, accuracy, and speed, ensuring that the right blood type is available when needed.

We have taken care of the project so that all given information is true, and the project is a smooth-running project without any errors or problems. We are very apologies if any wrong information is given.

INDEX

1. INTRODUCTION.....	01
1.1 Project Overview.....	01
1.2 Project background.....	01
1.3 Purpose.....	01
1.3.1 Problem Statement.....	01
1.3.2 Problem Solution.....	01
2. PROJECT SCOPE.....	02
3. FEASIBILITY ANALYSIS.....	04
3.1 Technical Feasibility.....	04
3.2 Time schedule feasibility.....	04
3.3 Operational Feasibility.....	04
3.4 Implementation Feasibility.....	04
3.5 Economic Feasibility.....	05
4. HARDWARE AND SOFTWARE REQUIREMENT.....	06
4.1 Hardware Requirements.....	06
4.2 Software Requirements.....	06
5. PROCESS MODEL.....	07
5.1 Project Vision.....	07
5.2 Stakeholders.....	07
5.3 Backlog Creation.....	07
5.4 Agile Sprints.....	07
5.5 Role of Agile Model.....	08
5.6 Workflow.....	08
5.7 Tools for collaboration.....	08
5.8 Key Agile Practices.....	08
6. PROJECT PLAN.....	09
7. SYSTEM DESIGN.....	10
7.1 Use Case Diagram.....	10
7.2 Class Diagram.....	11
7.3 Sequence Diagram.....	13
7.4 Activity Diagram.....	14
7.5 E-R Diagram.....	15
7.6 List of Tables.....	16
7.7 Table Design.....	16
7.8 Data Dictionary.....	17
7.9 Design Strategy 2-tier & 3-Tier.....	19
7.10 Data Flow Diagram.....	21
7.10.1 Data Flow Diagram Level-0.....	21
7.10.2 Data Flow Diagram Level-1.....	22
7.10.3 Data Flow Diagram Level-2.....	23
7.11 User Interface Design.....	25
8. IMPLEMENTATION DETAILS.....	32
8.1 Algorithm and Flowchart of Implementation.....	32
8.1.1 Algorithm.....	32
8.1.2 Flowchart of Implementation.....	33

8.2 Actual Program Code.....	34
9. TESTING.....	69
9.1 User Module.....	69
9.2 Admin Module.....	69
10 USER MANUAL.....	71
10.1 Installation Steps.....	71
10.2 Screen short With Explanation.....	72
11 CONCLUSION AND FEATURE WORK.....	79
11.1 Conclusion.....	79
11.2 Feature Work.....	79
12 ANNEXURE.....	81
12.1 Glossary of team and abbreviations.....	81
12.2 References.	81
12.3 About tools and technology.....	82
12.4 About UVPCE College.....	82

LIST OF FIGURES

Figure 1	Use-Case Diagram.....	10
Figure 2	Class Diagram.....	11
Figure 3	Sequence Diagram.....	13
Figure 4	Activity Diagram.....	14
Figure 5	E-R Diagram.....	15
Figure 6	DFD Level-0.....	21
Figure 7	DFD Level-1.....	22
Figure 8	DFD Level-2.....	23
Figure 9	Home Page.....	26
Figure 10	Login Page.....	27
Figure 11	About Us Page.....	27
Figure 12	Sign up Page.....	28
Figure 13	Reset Password Page.....	29
Figure 14	Look blood Page.....	29
Figure 15	Want to donate Blood Page.....	30
Figure 16	Admin Login Page.....	30
Figure 17	Contact Us Page.....	31
Figure 18	Admin Panel Page.....	31
Figure 19	Flowchart of Implementation.....	33
Figure 20	Sign Up Page.....	72
Figure 21	Login Page.....	73
Figure 22	Looking for a blood Page.....	74
Figure 23	You want to donate blood Page.....	76
Figure 24	About Us Page.....	77

LIST OF TABLES

Table 1	Project Plane.....	09
Table 2	Hospitals Tables.....	16
Table 3	Registration Table.....	16
Table 4	Contact_messages Table.....	17
Table 5	Donors Table.....	17
Table 6	Blood_Orders Table.....	17
Table 7	User Module Table.....	69
Table 8	Admin Module Table.....	69

1. INTRODUCTION

1.1 Project Overview

A Life Line Donor's Website is an online platform designed to streamline and facilitate the process of blood donation by connecting blood donors, recipients, hospitals, and blood banks. For hospitals and blood banks, the website helps manage blood inventory, process requests for blood, and ensure timely delivery of Life-saving resources.

1.2 Project Background

Blood is a critical resource for medical emergencies, surgeries, accidents, and chronic medical conditions. The timely availability of safe blood can save lives. However, maintaining an efficient and responsive blood donation system has always been a challenge, particularly in large communities or regions with uneven access to medical services. A Life Line Donor's aims to address these challenges by streamlining the collection, storage, and distribution of blood units.

1.3 Purpose

1.3.1 Problem Statement

A hospital's blood bank system often faces issues with tracking blood inventory, matching donor types with patient needs, and maintaining up-to-date donor information. Manual processes can lead to errors, delays in acquiring the required blood type, and communication gaps between donors and hospitals, ultimately risking patient health.

1.3.2 Problem solution

The Blood Management System website automates the process of blood inventory tracking, donor registration, and blood requests. It allows real-time monitoring of available blood types, sends notifications to donors based on demand, and helps hospitals quickly match and request the correct blood type, improving efficiency and saving lives.

2. PROJECT SCOPE

Understanding the Core Objectives:

A blood management system aims to streamline the process of blood donation, storage, and distribution. Its primary objectives include:

- Efficient Blood Collection: Simplifying the donation process for donors and ensuring accurate blood type identification.
- Safe Blood Storage: Implementing robust storage protocols to maintain blood quality and prevent contamination.
- Effective Blood Distribution: Optimizing the allocation of blood products to hospitals based on patient needs and availability.
- Accurate Record Keeping: Maintaining comprehensive records of blood donations, storage, and distribution for traceability and compliance.

Key Features and Functionality:

- Donor Management:
 - Online registration and scheduling
 - Health assessment and eligibility criteria
 - Donation history tracking
- Blood Collection:
 - Barcoding or other unique identification for blood units
 - Real-time tracking of blood donations
 - Quality control measures
- Blood Storage:
 - Inventory management system
 - Temperature monitoring and alarms
 - Expiration date tracking
- Blood Distribution:
 - Request management from healthcare facilities
 - Allocation based on blood type, patient needs, and availability
 - Transportation logistics tracking
- Reporting and Analytics:
 - Detailed reports on blood donations, storage, and distribution
 - Statistical analysis to identify trends and patterns
 - Compliance tracking and reporting
- Integration with Healthcare Systems:
 - Interoperability with hospital information systems (HIS)
 - Data exchange for patient records and blood transfusion information
- Security and Data Privacy:
 - Robust security measures to protect sensitive patient data
 - Compliance with relevant data privacy regulations (e.g., HIPAA, GDPR)

Project Scope Boundaries:

- Out-of-Scope Features:
 - Developing a blood testing laboratory
 - Implementing a blood transfusion monitoring system
 - Creating a blood donation awareness campaign
- Assumptions:
 - Existing infrastructure for blood collection, storage, and distribution

- Availability of trained personnel
- Compliance with local and national regulations

Potential Challenges and Mitigation Strategies:

- Data Accuracy: Implement data validation and quality control measures.
- System Integration: Ensure compatibility with existing healthcare systems and address technical challenges.
- Security Risks: Employ strong security measures and regularly update systems to protect sensitive data.
- Regulatory Compliance: Stay updated on relevant regulations and ensure compliance through regular audits.

3. FEASIBILITY ANALYSIS

3.1 Technical feasibility

- Infrastructure Requirements:
 - Identify the hardware and software needed to support the system.
 - Evaluate integration with existing hospital or healthcare systems (e.g., EMR/EHR systems).
- Technology Availability:
 - Assess the availability of suitable technology (e.g., cloud-based platforms, IoT devices for real-time tracking).
- Scalability:
 - Determine whether the system can scale to manage large volumes of data or expand to other regions or institutions.
- Data Security:
 - Evaluate encryption and data protection mechanisms to ensure patient and donor confidentiality.

3.2 Time schedule feasibility

- Timeline for Development and Implementation:
 - Estimate time for each development phase: requirements gathering, design, testing, deployment, and training.
- Deadlines and Constraints:
 - Assess if the project can meet critical deadlines (e.g., launching before peak demand seasons).

3.3 Operational Feasibility

- Stakeholder Acceptance:
 - Determine if staff, donors, and patients will accept and adopt the new system.
 - Conduct surveys or focus groups to gauge stakeholder opinions.
- Process Integration:
 - Evaluate how the system fits into existing workflows, such as donor management, blood testing, and transfusion processes.
- Training and Support:
 - Identify the training required for healthcare workers and technical support needed post-deployment.

3.4 Implementation Feasibility

- I. Planning Phase:
Conduct stakeholder meetings, feasibility studies, and define objectives.
- II. Development Phase:
Build or procure the system, integrate with existing infrastructure, and conduct initial tests.
- III. Pilot Testing:
Deploy in a controlled environment to identify and address issues.
- IV. Training:
Train staff on system use, troubleshooting, and workflows.
- V. Full Deployment:

- Roll out across all departments or facilities with ongoing support.
- VI. Monitoring and Feedback:
Continuously monitor performance and gather user feedback for iterative improvements.

3.5 Economic Feasibility

1. Cost Analysis:
 - Initial investment: hardware, software development, and installation costs.
 - Recurring costs: maintenance, upgrades, training, and support.
2. Cost-Benefit Analysis:
 - Potential savings from reduced blood wastage and improved inventory management.
 - Enhanced revenue opportunities through better service and reduced penalties for shortages.
3. Funding Sources:
 - Assess the availability of funding through government grants, private investment, or public-private partnerships.

4. HARDWARE AND SOFTWARE REQUIREMENT

4.1 Hardware Requirements

- **Processor** : While an Intel Core i3 is sufficient for a smaller-scale system, a more powerful processor like an Intel Core i5 or i7 might be beneficial for handling larger datasets and complex operations.
- **RAM** : 4GB of RAM should be adequate for most tasks, but consider increasing it to 8GB or more if the system is expected to handle a large number of concurrent users or complex calculations.
- **Storage** : 128GB of storage should be sufficient for a basic system, but if the system is expected to store a large amount of data (e.g., historical records, images), a larger storage capacity might be required.

4.2 Software Requirements

- **Frontend** : HTML, CSS, and JavaScript are the essential tools for building the user interface. Consider using a frontend framework like React or Angular to improve development efficiency and maintainability.
- **Backend** : PHP for backend development. The best choice depends on the team's expertise and the specific requirements of the system.
- **Database** : PhpMyAdmin is a good choice for storing structured data, but there are other options available, such as NoSQL databases like MongoDB or Cassandra. The choice of database depends on the nature of the data and the specific requirements of the system.

Additional Considerations:

- **Operating System** : While Windows is a common choice, Linux distributions like Ubuntu or CentOS are also popular options for server environments due to their stability and security.
- **Web Server** : A web server like Apache or Nginx is required to serve the frontend application.
- **Development Tools** : Consider using a development environment like Visual Studio Code or IntelliJ IDEA to improve productivity and streamline development.
- **Version Control** : Use a version control system like Git to track changes to the codebase and collaborate with other developers.
- **Testing and Quality Assurance** : Implement a comprehensive testing strategy to ensure the quality and reliability of the system.

5. PROCESS MODEL

Developing an agile model for a **Lifeline Donors website** involves organizing the project into flexible, iterative cycles to ensure continuous improvement and timely delivery. Here's an agile model framework tailored for the development:

5.1 Product Vision

Create a website that facilitates seamless interactions between blood donors, recipients, and donation centers, ensuring timely and efficient blood donation services.

5.2 Stakeholders

- **Product Owner:** Represents the donors, recipients, and admins.
- **Scrum Master/Agile Coach:** Ensures the team adheres to agile principles.
- **Development Team:** Designers, developers, and testers.
- **End Users:** Donors, recipients, and medical staff.

5.3 Backlog Creation

Product Backlog (High-Level Features):

1. **User Registration & Authentication**
 - Donor and recipient accounts.
 - Integration with social logins.
2. **Search & Match Functionality**
 - Search for blood donors by location, blood type, and availability.
3. **Donation Requests**
 - Ability to create, view, and respond to blood requests.
4. **Notifications & Alerts**
 - SMS/Email alerts for new requests or donation campaigns.
5. **Donation Tracker**
 - History of donations for donors and recipients.
6. **Content Management**
 - Informational pages about blood donation benefits, FAQs, and safety guidelines.
7. **Admin Dashboard**
 - Manage users, donations, and campaigns.
8. **Reports and Analytics**
 - Insights on donation trends, active users, etc.

5.4 Agile Sprints

Divide the project into **2-3 week sprints** focusing on a subset of features.

Example Sprint Plan:

- **Sprint 1:**
 - Basic website skeleton.
 - User registration and login system.
- **Sprint 2:**
 - Search and match functionality.
 - Donation request module.

- **Sprint 3:**
 - Notifications and alerts.
 - Donation tracker.
- **Sprint 4:**
 - Admin dashboard and content management.
 - Integration testing.
- **Sprint 5:**
 - Final testing, performance optimization, and deployment.

5.5 Roles in Agile Model

- **Scrum Master:** Facilitates daily stand-ups, ensures team focus, resolves blockers.
- **Product Owner:** Manages backlog priorities, gathers user feedback.
- **Development Team:** Creates designs, writes code, and tests functionalities.

5.6 Workflow

Daily Stand-Up (15 min):

- What was done yesterday?
- What will be done today?
- Any blockers?

Sprint Planning:

Define tasks and objectives for the sprint, estimate timelines.

Sprint Review:

Demo completed features, gather feedback from stakeholders.

Retrospective:

Review what went well, what didn't, and plan improvements for the next sprint.

5.7 Tools for Collaboration

- **Project Management:** Jira, Trello, or Asana.
- **Version Control:** Git/GitHub.
- **Design:** Figma or Adobe XD.
- **Communication:** Slack, Microsoft Teams.

5.8 Key Agile Practices

1. **User Stories:** Write actionable user stories, e.g., "*As a donor, I want to view my donation history so I can track my contributions.*"
2. **Incremental Delivery:** Deploy features incrementally for continuous user feedback.
3. **Continuous Integration:** Regularly integrate code into a shared repository with automated testing.
4. **Retrospective Meetings:** Adapt based on lessons learned.

6. PROJECT PLANE

Sr. No.	Major Activities	Estimated Duration (weeks)
1	Requirements Gathering	3 weeks
2	UI/UX Design	3 weeks
3	Database Design & Setup	2 weeks
4	User Registration & Profile Creation	2 weeks
5	Personalized Recommendations Algorithm	4 weeks
6	Supplement & Meal Plan Database	3 weeks
7	Progress Tracking Feature Development	3 weeks
8	Search & Filtering Functionality	2 weeks
9	Feedback & Ratings System	2 weeks
10	Notifications & Reminders	2 weeks
11	Secure Payment Processing Integration	2 weeks
12	Multi-Platform Accessibility	4 weeks
13	Data Privacy & Security Implementation	2 weeks
14	Testing & Quality Assurance	3 weeks
15	User Onboarding Development	2 weeks
16	Deployment & Launch	1 weeks

Table 1 Project Plane

7. SYSTEM DESIGN

7.1 Use case Diagram

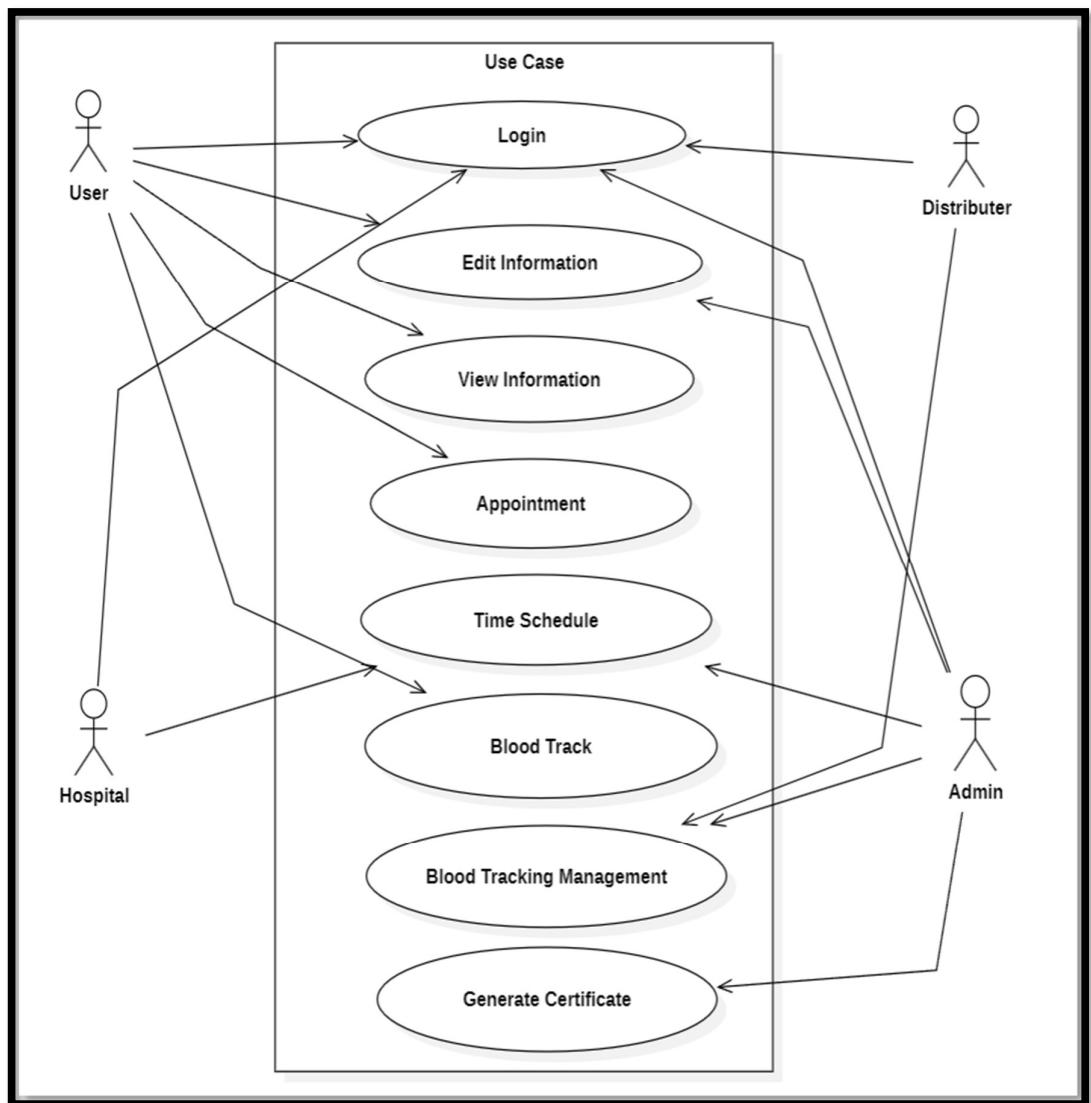


Figure 1 Use-case Diagram

Use case diagram provides a foundational understanding of the key functionalities and interactions within a blood bank management system. Further analysis and refinement would be necessary to fully capture the system's requirements and capabilities.

Key Use Cases :

- **Login:** Allows users to access the system.
- **Edit Information:** Enables users to update their personal information.
- **View Information:** Provides users with access to their own information and potentially other relevant data.
- **Appointment:** Facilitates scheduling appointments for blood donation.

- **Time Schedule:** Allows for viewing and managing time schedules related to appointments or operations.
- **Blood Track:** Enables tracking and monitoring of blood units.
- **Blood Tracking Management:** Provides administrative functions for managing blood tracking processes.
- **Generate Certificate:** Generates certificates or documentation related to blood donation or other activities.

Actor Interactions

- **User:** Can log in, edit information, view information, and schedule appointments.
- **Distributor:** May have access to certain aspects of blood tracking or distribution.
- **Admin:** Has administrative privileges to manage blood tracking, generate certificates, and potentially other functions.
- **Hospital:** Interacts with the system to request blood units or access information related to blood availability.

7.2 Class Diagram

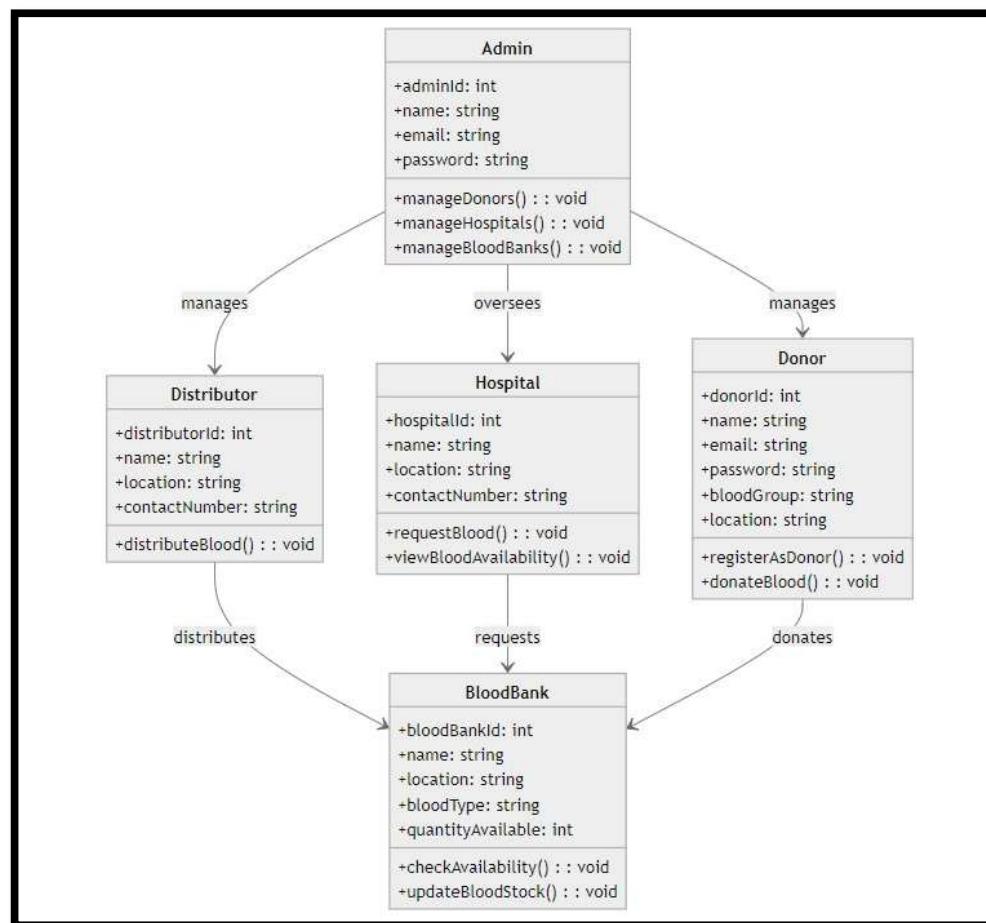


Figure 2 Class Diagram

The diagram you provided is a use case diagram that illustrates the interactions between users (actors) and a system for managing blood donations and distribution.

Actors:

- Donor: The individual who wishes to donate blood.
- Admin: The person responsible for managing the overall system.
- Distributor: The entity tasked with delivering blood to hospitals.
- Hospital: The institution that requires blood for medical purposes.

Use Cases:

- sendDonationRequest(): The use case represents the action taken by a donor to initiate the donation process.
- manageRequests(): The use case represents the actions taken by the admin to manage blood donation requests, including logging requests, updating the requests database, and coordinating with the distributor.
- manageDonations(): The use case represents the actions taken by the admin to manage blood donations, including updating the donations database and coordinating with the distributor.
- informDistributor(): The use case represents the action taken by the admin to notify the distributor about available blood for distribution.
- distributeBlood(): The use case represents the action taken by the distributor to deliver blood to hospitals.
- requestBlood(): The use case represents the action taken by a hospital to request blood for medical purposes.

Relationships:

- Include : The "manageRequests" and "manageDonations" use cases include the "informDistributor" use case, indicating that the admin must inform the distributor as part of managing requests and donations.
- Extend : The "manageRequests" and "manageDonations" use cases extend the "Admin" actor, indicating that the admin is responsible for performing these actions.

Databases :

- DonationsDB: Stores information about blood donations.
- RequestsDB: Stores information about blood requests.

This diagram provides a visual representation of the system's functionality and the interactions between its components. It helps to understand the system's workflow and identify potential areas for improvement.

7.3 Sequence Diagram

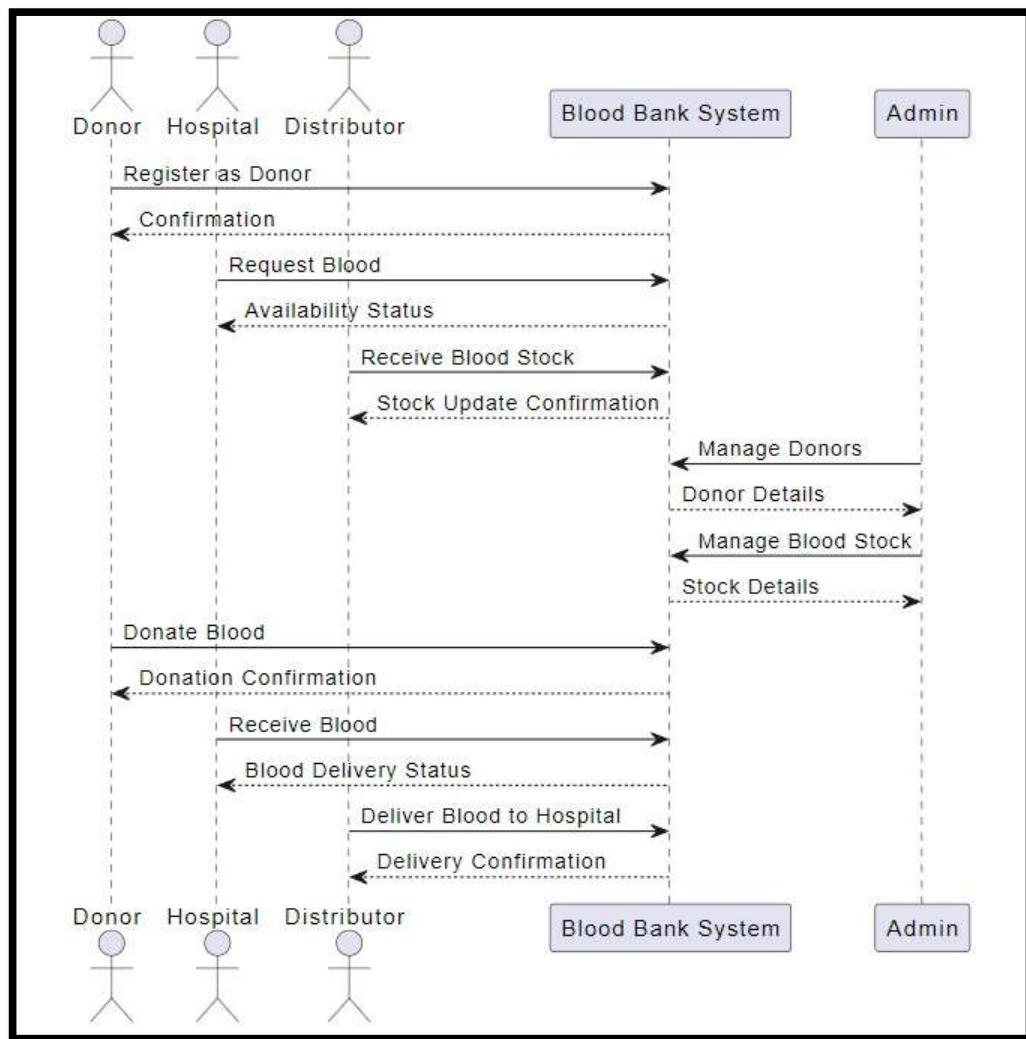


Figure 3 Sequence Diagram

The provided sequence diagram illustrates the workflow involved in blood donation and distribution. Here's a breakdown of the steps:

1. **Donor Sends Donation Request:** The process begins when a donor sends a request to donate blood. This request could be made through an online portal, mobile app, or directly to the organization handling blood donations.
2. **Admin Logs Donation Request:** The admin receives the donation request and logs it into the Donations DB. This database stores information about all donation requests.
3. **Admin Requests Blood:** Based on the blood type and availability, the admin requests blood from the Distributor. This request is sent to the Distributor.
4. **Distributor Logs Blood Request:** The Distributor receives the blood request and logs it into the RequestsDB. This database stores information about all blood requests.

5. Admin Informs Distributor to Deliver Blood: The admin informs the Distributor to deliver the requested blood to the Hospital.
6. Blood Delivered: The Distributor delivers the blood to the Hospital. This completes the process of blood donation and distribution.

This sequence diagram provides a clear visualization of the interactions between the different roles involved in the process, from the donor to the hospital, ensuring efficient and organized blood donation and distribution.

7.4 Activity Diagram

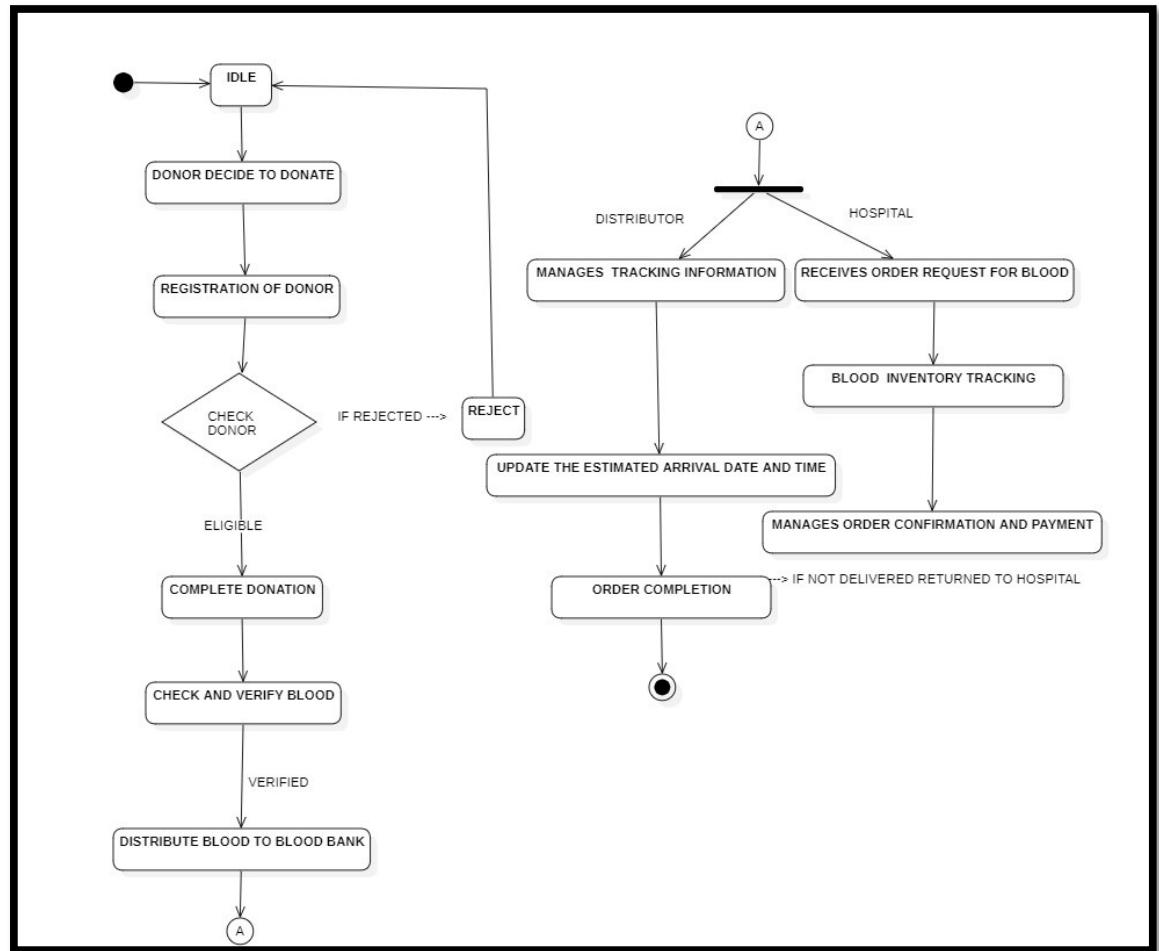


Figure 4 Activity Diagram

This diagram provides a clear visualization of the workflow involved in blood donation, from the initial decision by the donor to the final distribution of blood to the blood bank.

Key Stages and Decision Points:

1. **Idle:** The system starts in an idle state, waiting for a donor to initiate the process.

2. **Donor Decides to Donate:** A potential donor decides to donate blood and proceeds to the next step.
3. **Registration of Donor:** The donor's information is registered, including personal details and medical history.
4. **Check Donor:** The donor's eligibility for donation is assessed based on various criteria, such as health conditions and blood type.
5. **If Rejected:** If the donor is not eligible, the process ends with a rejection.
6. **Eligible:** If the donor is eligible, they proceed to the next step.
7. **Complete Donation:** The donor completes the blood donation process.
8. **Check and Verify Blood:** The donated blood is tested and verified for quality and safety.
9. **Verified:** If the blood is verified, it proceeds to the next step.
10. **Distribute Blood to Blood Bank:** The verified blood is distributed to the blood bank for storage and future use.

Parallel Processes:

1. **Distributor:** While the donor completes the donation process, the distributor manages tracking information related to blood units.
2. **Hospital:** The hospital receives order requests for blood and tracks blood inventory.

Order Completion: The system manages order confirmation, payment, and delivery. If the order is not delivered, it is returned to the hospital.

7.5 E-R Diagram

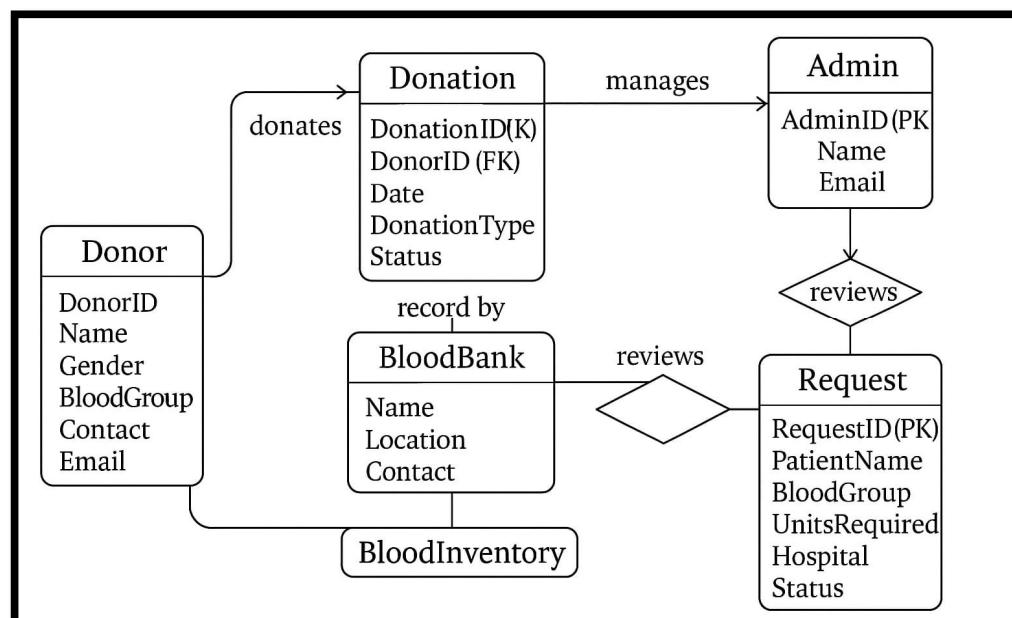


Figure 5 E-R Diagram

It represents the data structure of the Lifeline Blood Donors system. It includes key entities such as Donor, Donation, BloodBank, BloodInventory, Request, and Admin. Each entity contains relevant attributes, and relationships between them define how data flows—donors make donations, blood banks manage inventory, hospitals raise requests, and admins monitor and approve system activities. The design supports efficient tracking and coordination of blood donation operations.

7.6 List of Tables

1. Hospitals Table : Stores information about hospitals registered in the system.
2. Registration Table : Stores information of users who register (hospital staff, admin, etc.)
3. Contact_messages Table : Stores messages submitted via the contact form.
4. Donor's Table : Stores information about blood donors.
5. Blood_orders : Stores information about blood orders placed by hospitals through the system.

7.7 Table design

Sr No.	Column_name	Data_type
1	District	Varchar(16)
2	City Name	Varchar(14)
3	Hospital Name	Varchar(31)
4	A+	Varchar(3)
5	A-	Varchar(3)
6	B+	Varchar(3)
7	B-	Varchar(3)
8	AB+	Varchar(3)
9	AB-	Varchar(3)
10	O+	Varchar(3)
11	O-	Varchar(3)

Table 2 Hospitals Table

Sr No.	Column Name	Data Type
1	Id	Int(11)
2	UserName	Varchar(255)
3	EmailID	Varchar(255)
4	Mobile Number	Varchar(15)
5	Address	Varchar(255)
6	Password	Varchar(255)

Table 3 Registration Table

Sr No.	Column Name	Data Type
1	Id	Int(11)
2	Username	Varchar(100)
3	Email	Varchar(150)
4	Message	Text
5	Submitted_at	timestamp

Table 4 Contact_messages Table

Sr No.	Column Name	Data Type
1	Id	Int(11)
2	Name	Varchar(100)
3	Age	Int(11)
4	Mobile	Varchar(15)
5	Hospital	Varchar(255)
6	DateTime	Varchar(100)
7	Submitted_at	Timestamp

Table 5 Donors Table

Sr No.	Column Name	Data Type
1	Order_id	Int(11)
2	Customer_name	Varchar(100)
3	Mobile_number	Varchar(15)
4	Hospital_name	Varchar(100)
5	a_plus	Int(11)
6	a_minus	Int(11)
7	b_plus	Int(11)
8	b_minus	Int(11)
9	ab_plus	Int(11)
10	ab_minus	Int(11)
11	o_plus	Int(11)
12	o_minus	Int(11)
13	Order_date	Datetime
14	status	Varchar(20)

Table 6 Blood_orders

7.8 Data Dictionary

1. Hospitals Tables

District : District where the hospital is located

City Name : City where the hospital is located

Hospital Name : Name of the hospital

A+ : Units of A+ blood available

A- : Units of A- blood available

B+ : Units of B+ blood available

B- : Units of B- blood available

AB+ : Units of AB+ blood available

AB- : Units of AB- blood available

O+ : Units of O+ blood available

O- : Units of O- blood available

2. *Registration Table*

Id : Unique ID for each user

User Name : Full name or username of the user

Email ID : User's email address (must be unique)

Mobile Number : Contact number of the user

Address : User's complete address

Password : User's password (should be stored hashed)

3. *Contact_messages Table*

Id ; Unique ID for each message

Username : Name of the person who submitted the message

Email : Email address of the sender

Message : The actual content of the message

Submitted_at : Date and time the message was submitted

4. *Donors Table*

Id : Unique ID for each donor entry

Name : Full name of the donor

Age : Age of the donor

Mobile : Contact number of the donor

Hospital : Name of the hospital where the donation was made

Date Time : Date and time of the donation

Submitted_at : Date and time the entry was submitted into the system

5. *Blood_orders*

Order_id : Unique identifier for each order (Primary Key).

Customer_name : Name of the customer placing the blood order.

Mobile_number : Customer's mobile number.

Hospital_name : Name of the hospital requesting blood units.

a_plus : Number of A+ blood units ordered.

a_minus : Number of A- blood units ordered.

b_plus : Number of B+ blood units ordered.

b_minus : Number of B- blood units ordered.

ab_plus : Number of AB+ blood units ordered.

ab_minus : Number of AB- blood units ordered.

o_plus : Number of O+ blood units ordered.

o_minus : Number of O- blood units ordered.

Order_date : Date and time when the order was placed.

Status : Current status of the order (e.g. Pending, Approved, Rejected).

7.9 Design strategy 2-tier And 3-tier

⇒ 2 tier strategy

Tier 1: Client (Presentation Layer)

Purpose : To interact with end-users (donors, hospitals, admin).

Users and Actions :

- Donor
 - Register themselves (name, age, blood group, contact info)
 - Update profile
- Hospital
 - Request for blood units
 - Search for available blood groups

Technologies :

- HTML, CSS, JavaScript, PHP

Tier 2: Server (Data Layer)

Purpose : To store and manage all the application's data.

Tables in Database :

- Donors
 - ID, Name, Age, Gender, Blood Group, Contact Info
- Blood Inventory
 - Blood Group, Unit Available
- Hospitals
 - Name, Contact, Address
- Requests
 - Blood Group

Technologies :

- PhpMyAdmin

How It Works:

The client app sends a request (e.g., search for A+ blood).

The server (database) responds with data (e.g., available units, donor list).

All logic like validation or calculations is handled on the client side.

⇒ 3 tier Strategy

Tier 1: User interface (Presentation Layer)

Purpose : To interact with end-users

Components :

- Web Pages

Users and Actions :

- Donor
 - Register themselves (name, age, blood group, contact info)
 - Update profile
- Hospital
 - Request for blood units
 - Search for available blood groups

Technologies :

- HTML, CSS, JavaScript, PHP

Tier 2: Application Tier (Business logic Layer)

Purpose : Processes user requests, applies rules and coordinates data.

Responsibilities:

- Authenticate users (login/register)
- Match donors with blood requests
- Handle donor notifications (e.g., via email/SMS)
- Validate and process blood request forms
- Handle search and filter operations

Technologies :

- PHP
- PhpMyAdmin

Tier 3 : Data Tier (Database Layer)

Purpose : This is where all the data is stored and managed.

Responsibilities :

- Store donor details, blood groups, hospital data
- Track available blood units by type and location
- Maintain logs of blood requests and donations

Technologies :

- PhpMyAdmin

Tables :

- Hospitals Table
- Registration Table
- Contact_messages Table
- Donor's Table

7.10 Data Flow Diagram

7.10.1 Data Flow Diagram Level-0

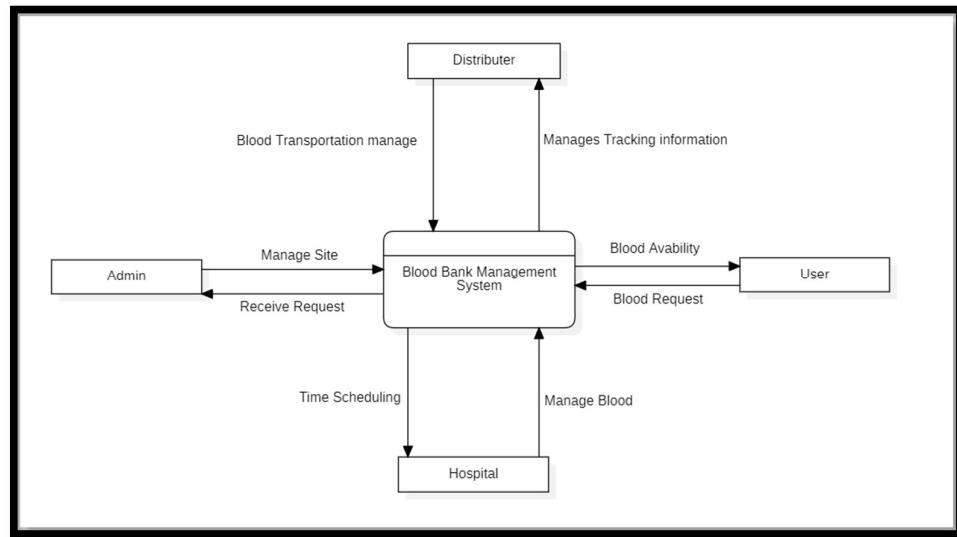


Figure 6 DFD Level-0

This diagram provides a high-level overview of the key components and interactions within a blood bank management system. Further analysis and refinement would be necessary to fully capture the system's requirements and capabilities.

Key Components and Interactions:

- **Blood Bank Management System:** The central hub of the system, responsible for managing various aspects of blood bank operations.
- **Admin:** Oversees the system, manages sites, receives requests, and schedules time.
- **Distributor:** Manages blood transportation, tracks information, and ensures blood availability.
- **User:** Likely refers to blood donors or recipients who interact with the system to schedule appointments or view information.
- **Hospital:** Requests blood from the blood bank and receives the requested units.

Specific Responsibilities:

- **Admin:**
 - Manages the physical locations or sites of the blood bank.
 - Receives and processes requests for blood from hospitals.
 - Schedules appointments for blood donation or other activities.
- **Distributor:**
 - Organizes and manages the transportation of blood units to hospitals.

- Tracks information related to blood transportation and delivery.
- Ensures that blood is available to meet the needs of hospitals.
- **User:**
 - Likely interacts with the system to schedule appointments for blood donation or view their own information.
- **Hospital:**
 - Requests blood units from the blood bank based on patient needs.
 - Receives the requested blood units for patient treatment.

7.10.2 Data Flow Diagram Level-1

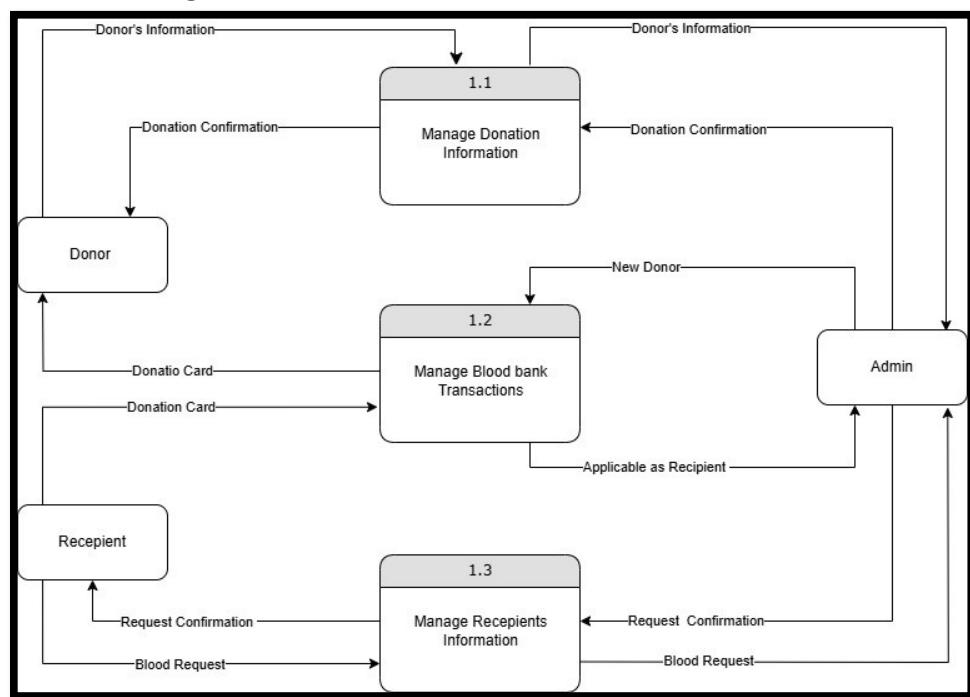


Figure 7 DFD Level-1

The diagram illustrates the process of blood donation and distribution in a hypothetical system. Here's a breakdown of the components and their interactions:

Actors:

- **Donor:** The individual who wishes to donate blood.
- **Admin:** The person responsible for managing the donation and request processes.
- **Distributor:** The entity tasked with delivering blood to hospitals.
- **Hospital:** The institution that requires blood for medical purposes.

Components:

- Send Donation Request: The action taken by the donor to initiate the donation process.
- Logs Request: The action taken by the admin to record the donation request.
- Informs: The action taken by the admin to notify the distributor about the available blood.
- Delivers Blood: The action taken by the distributor to transport the blood to the hospital.
- Donations Database: A database that stores information about blood donations.
- Requests Database: A database that stores information about blood requests.

Process Flow:

1. Donor sends a donation request: The donor sends a request to the admin indicating their desire to donate blood.
2. Admin logs the request: The admin records the donation request in the donations database.
3. Admin informs the distributor: The admin notifies the distributor about the available blood.
4. Distributor delivers blood to the hospital: The distributor transports the blood from the donor to the hospital.
5. Hospital receives blood: The hospital receives the donated blood for medical use.

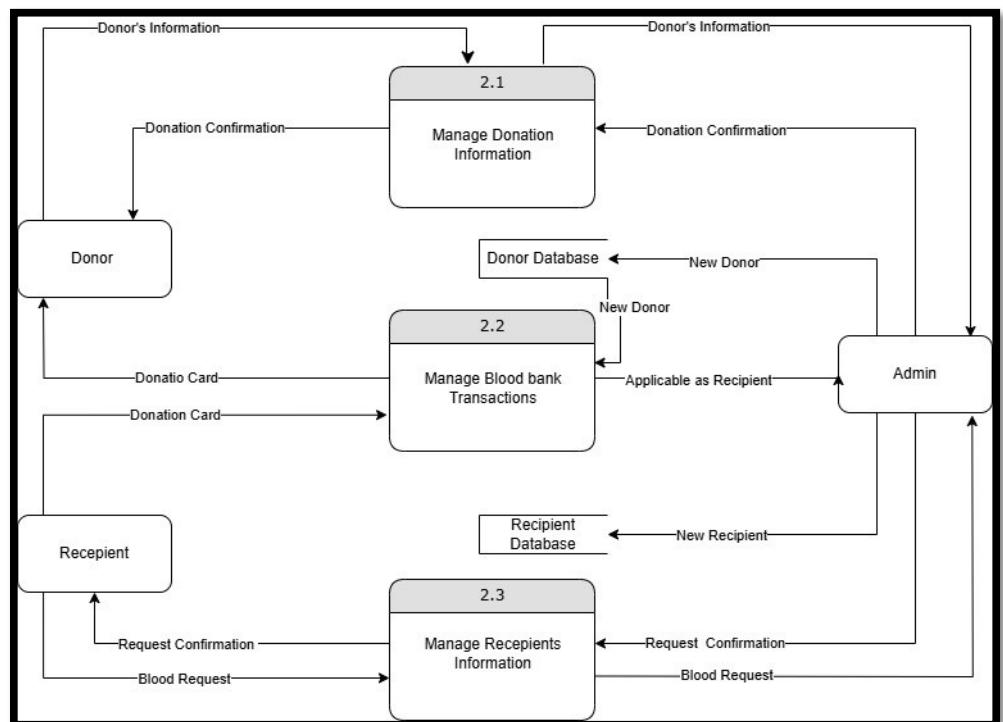
7.10.3 Data Flow Diagram Level-2

Figure 8 DFD Level-2

The diagram illustrates a system for managing blood donations and distribution. Here's breakdown of its components and their interactions:

Actors:

- Donor : The individual who wishes to donate blood.
 - Distributor : The entity tasked with delivering blood to hospitals.
 - Hospital : The institution that requires blood for medical purposes.
 - Admin : The person responsible for managing the overall system.
- Processes:
- Donation Process: The sequence of activities involved in a donor donating blood.
 - Distribution Process: The sequence of activities involved in delivering blood to hospitals.
 - Request Process: The sequence of activities involved in a hospital requesting blood.
 - Admin Process: The sequence of activities involved in managing the system, including logging requests, updating databases, and coordinating the donation and distribution

Databases:

- Donations Database: Stores information about blood donations.
- Requests Database: Stores information about blood requests.
- Users Database: Stores information about all users of the system, including donors, distributors, and hospital representatives.

Interactions:

- Donor: Initiates the donation process by sending a donation request.
- Donation Process: Records the donation and updates the donations database.
- Distributor: Receives information about available blood and distributes it to hospitals.
- Distribution Process: Updates the donations and requests databases to reflect the distribution.
- Hospital: Requests blood by initiating the request process.
- Request Process: Updates the requests database and informs the admin about the request.
- Admin: Manages the system by logging requests, updating databases, and coordinating the donation and distribution processes.

7.11 User Interface Design

The screenshot displays the user interface of a mobile application for blood donation. At the top, there is a navigation bar with the following items:

- Lifeline Donor's (Logo)
- About Us
- Looking for Blood
- Want to Donate Blood
- Login
- Admin Login
- Contact Us

Below the navigation bar, a banner displays the text "Upcoming Blood Donation Camps : 23 July, 2025 | Time : 1:0".

The main content area features a diagram illustrating blood compatibility. It shows a central blood bag labeled "O (-ve)" with the text "can donate" next to it. Below the bag, a horizontal line connects nine white human icons, each representing a different blood type. The blood types are labeled as follows:

- A+
- B-
- B+
- O-
- AB-
- AB+
- O+
- A-

In the bottom right corner of the main content area, there is a small logo for "Lifeline Donor's" with the text "Lifeline Donor's" and "Blood Donation App" below it.

At the bottom of the screen, there is a callout box titled "Compatible Blood Type Donors" with the sub-instruction "Find out which blood types are compatible for donation and transfusion." The callout contains a table:

Blood Type	Can Donate To	Can Receive From
A+	A+, AB+	A+, A-, O+, O-
A-	A+, A-, AB+, AB-	A-, O-
B+	B+, AB+	B+, B-, O+, O-
B-	B+, B-, AB+, AB-	B-, O-
AB+	AB+	All Blood Types
AB-	AB+, AB-	A-, B-, AB-, O-

O+	O+, A+, B+, AB+	O+, O-
O-	All Blood Types	O-

TYPES OF DONATION

The average human body contains about five liters of blood, which is made of several cellular and non-cellular components such as **Red blood cell**, **Platelet**, and **Plasma**.

Each type of component has its unique properties and can be used for different indications. The donated blood is separated into these components by the blood center, and one donated unit can save up to four lives depending on the number of components separated from your blood.

GuideLine for donors

What is it?

Blood collected straight from the donor into a blood bag and mixed with an anticoagulant is called whole blood. This collected whole blood is then centrifuged and red cells, platelets, and plasma are separated. The separated red cells are mixed with a preservative to be called packed red blood cells.

Who can donate?

You need to be 18-65 years old, weigh 45kg or more, and be fit and healthy.

User For?

Correction of severe anemia in a number of conditions and blood loss in case of childbirth, surgery, or trauma settings.

Lasts For?

Red cells can be stored for 42 days at 2-6 degrees Celsius.

How long does it take to donate?

15-30 minutes to donate, including the pre-donation check-up.

How often can I donate?

Male donors can donate again after 90 days, and female donors can donate again after 120 days.

[Find Nearest Blood Bank To Donate](#)

Looking for Blood

[Blood Availability](#)
[Blood Bank Directory](#)

Want to Donate Blood

[Blood Donation Camps](#)
[Donor Login](#)

About Us

[About Lifeline Donors](#)
[Contact Us](#)

[Terms & Conditions](#) | [Privacy Policy](#) | [Accessibility Statement](#) | [Site Map](#)

Figure 9 Home Page

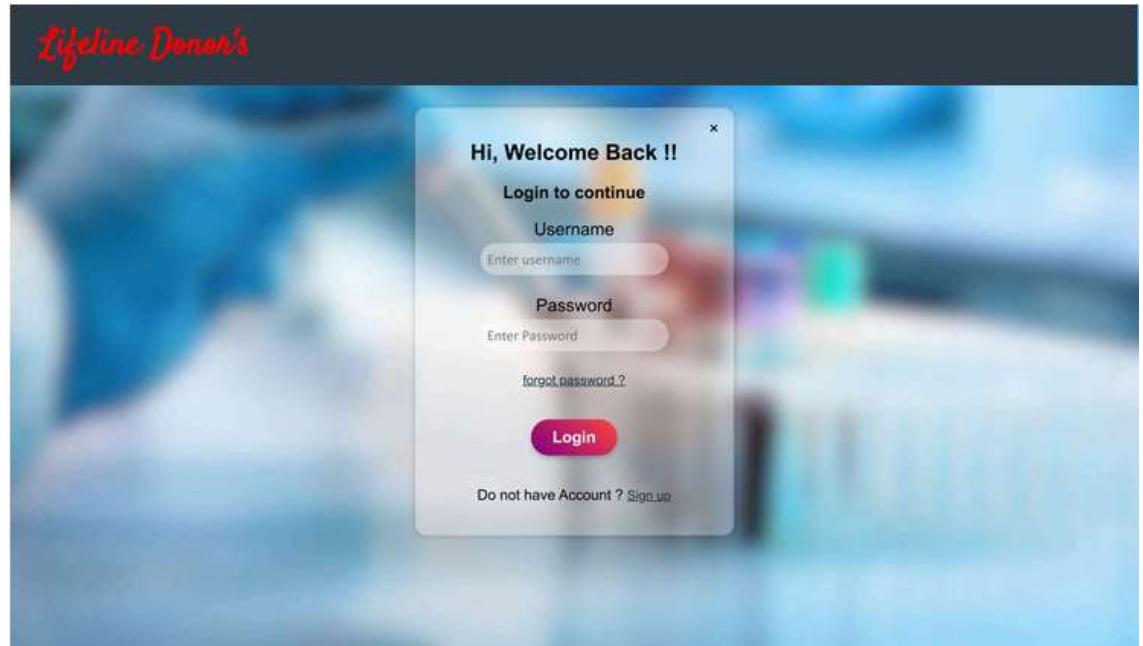


Figure 10 Login Page

The image shows the 'About Us' page of the Lifeline Donor's website. The top left features the 'Lifeline Donor's' logo. A central white rectangular box contains the heading 'About Us' in bold red font. Below the heading, a paragraph describes the platform as a digital bridge between blood donors and recipients, with the mission of ensuring safe and compatible blood is available on time. Another paragraph details services like blood donation registration and finding nearest blood banks. A third paragraph highlights the team's commitment to modern technology and saving lives. At the bottom of the box is the text 'Join our mission and become a part of a life-saving movement!' followed by a purple 'Back to Home Page' button. At the very bottom of the page, a small copyright notice reads '© 2025 Lifeline Donor's. All Rights Reserved.'

Figure 11 About Us Page

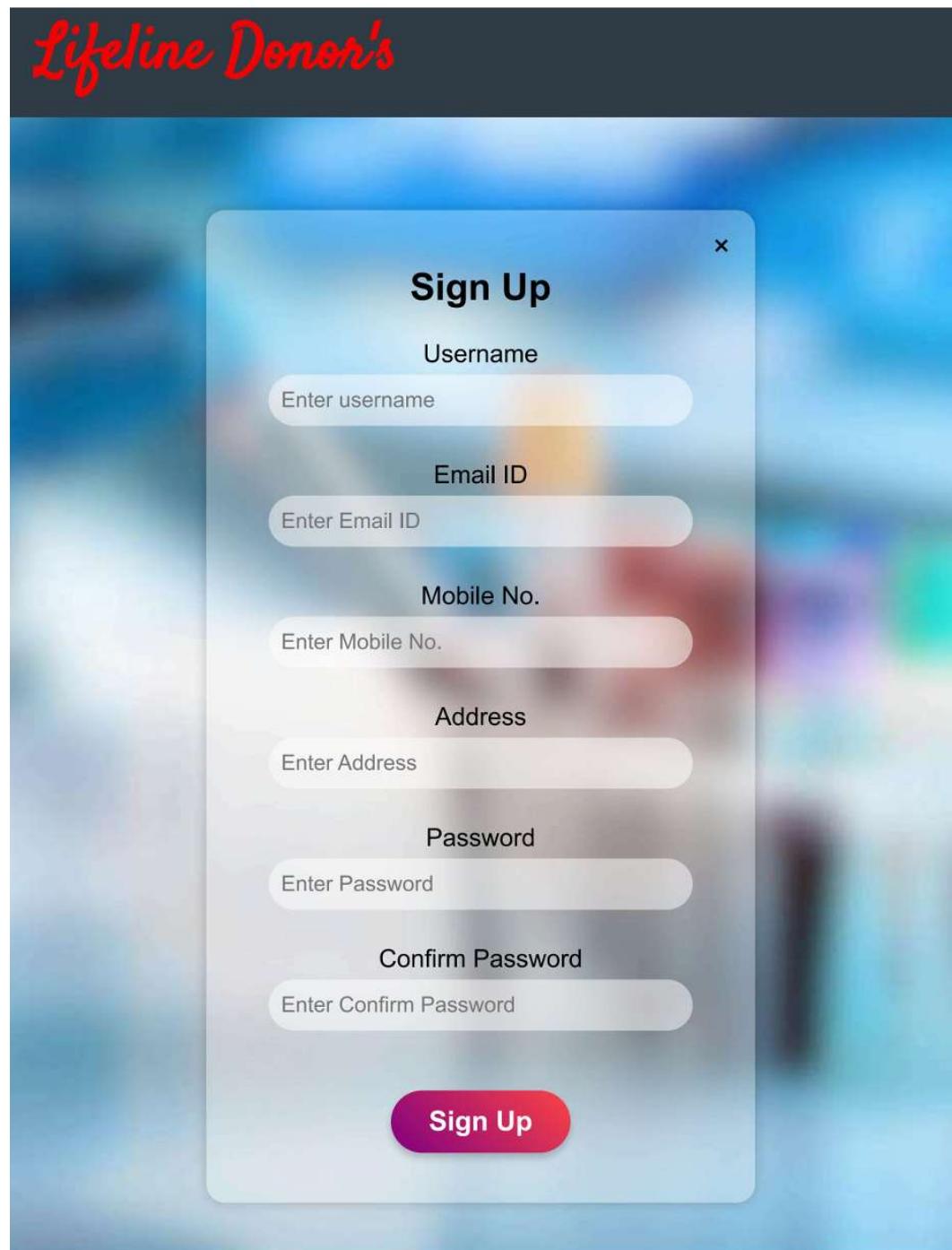


Figure 12 Sign Up Page

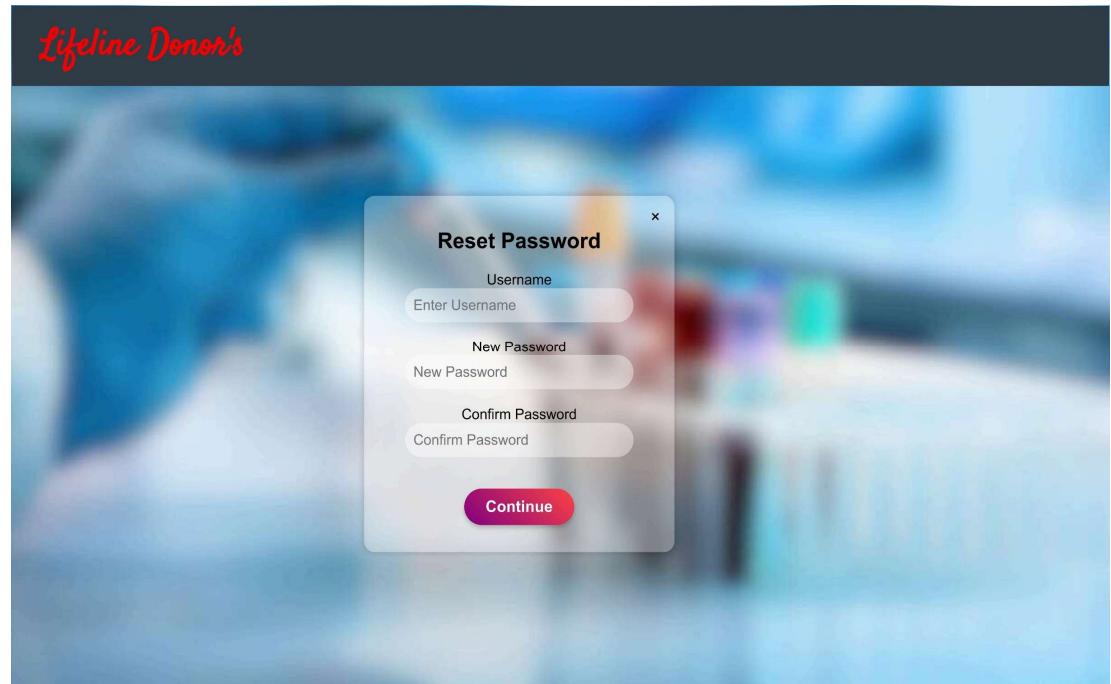


Figure 13 Reset Password Page

A screenshot of the "Lifeline Donor's" application showing the "Blood Storage Unit (BSU)" section. At the top, there are dropdown menus for "District" (Select District) and "City" (Select City), followed by a "Search" button. Below this is a table with columns for "Hospital Name" and blood types (A+, A-, B+, B-, AB+, AB-, O+, O-) with a "Select Hospital" column at the end. A message "No data available" is shown. Below this is another table with columns for "Hospital Name" and blood types (A+, A-, B+, B-, AB+, AB-, O+, O-). At the bottom, there is a search bar ("Enter name"), a mobile number input field ("Enter Mobile No."), and an "Order Now" button.

Figure 14 Look Blood Page

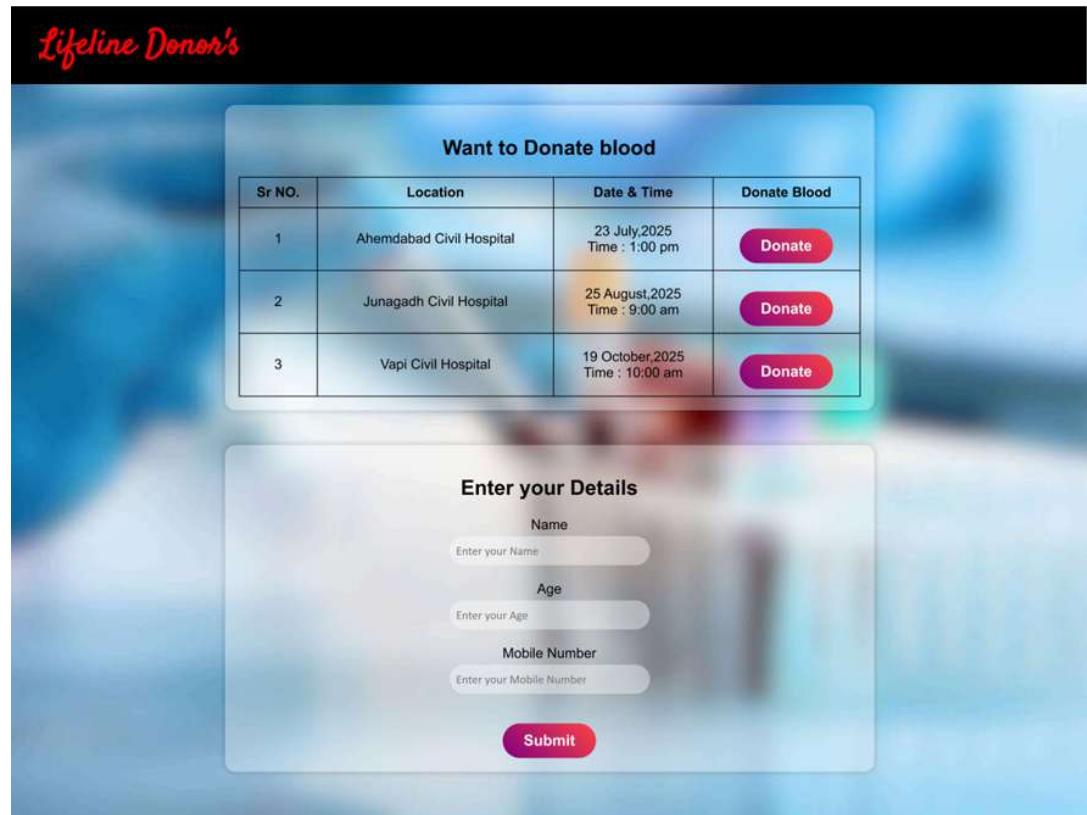


Figure 15 Want to donate Blood Page

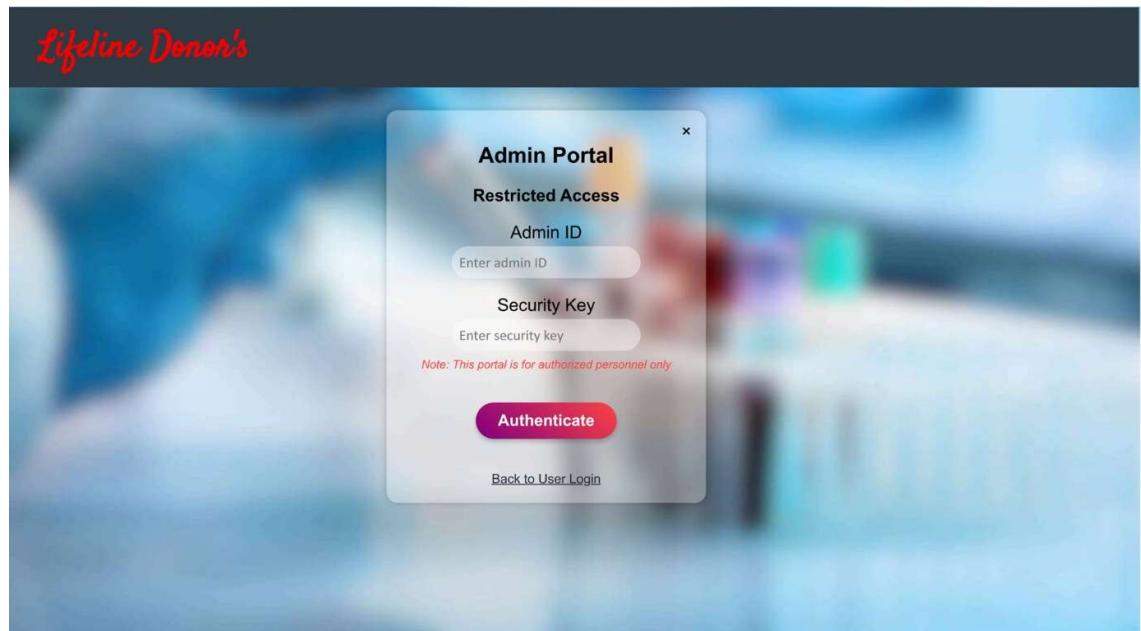


Figure 16 Admin Login Page

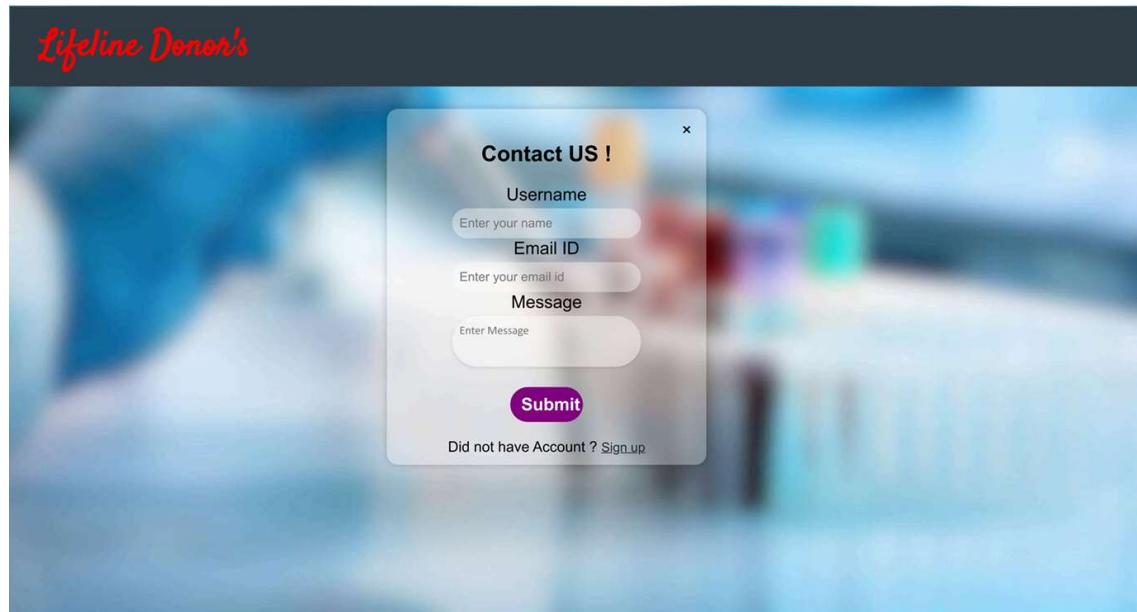


Figure 17 Contact Us Page

Figure 18 Admin Panel Page

8. IMPLEMENTATION DETAILS

8.1 Algorithm and Flowchart of Implementation

8.1.1 Algorithm

- Start : The process begins with the user accessing the system.
- User Registration/Login : Users (donors or recipients) must register or log in to the system to continue.
- Fill Donation Form / Blood Request : Once logged in, users can either fill out a blood donation form or raise a blood request depending on their role.
- Submit Request/Input : The form is submitted. Based on the type of request, data is either stored directly or sent to the admin for verification.
- Store Donor/Request Data : If the submission includes basic donor or request information, it is stored in the system database.
- Admin Login : The admin accesses the system to handle the backend operations, including reviewing the submitted requests.
- Admin Verifies Donors & Requests : The admin verifies the authenticity and availability of donors and validates recipient requests.
- Approve/Reject Requests : The admin decides whether to approve or reject the submitted requests.
- If Rejected → Notify Donor/Receiver via Email/SMS : If a request is rejected, the system sends a notification to the donor or receiver through SMS or email.
- If Approved → Update Blood Inventory : If the request is approved, the system updates the blood inventory accordingly (e.g., adding or removing units).
- Store Data in History/Logs : All approved or rejected actions, inventory updates, and interactions are logged in the system for future reference.
- End : The process ends here, awaiting the next interaction.

8.1.2 Flowchart of Implementation

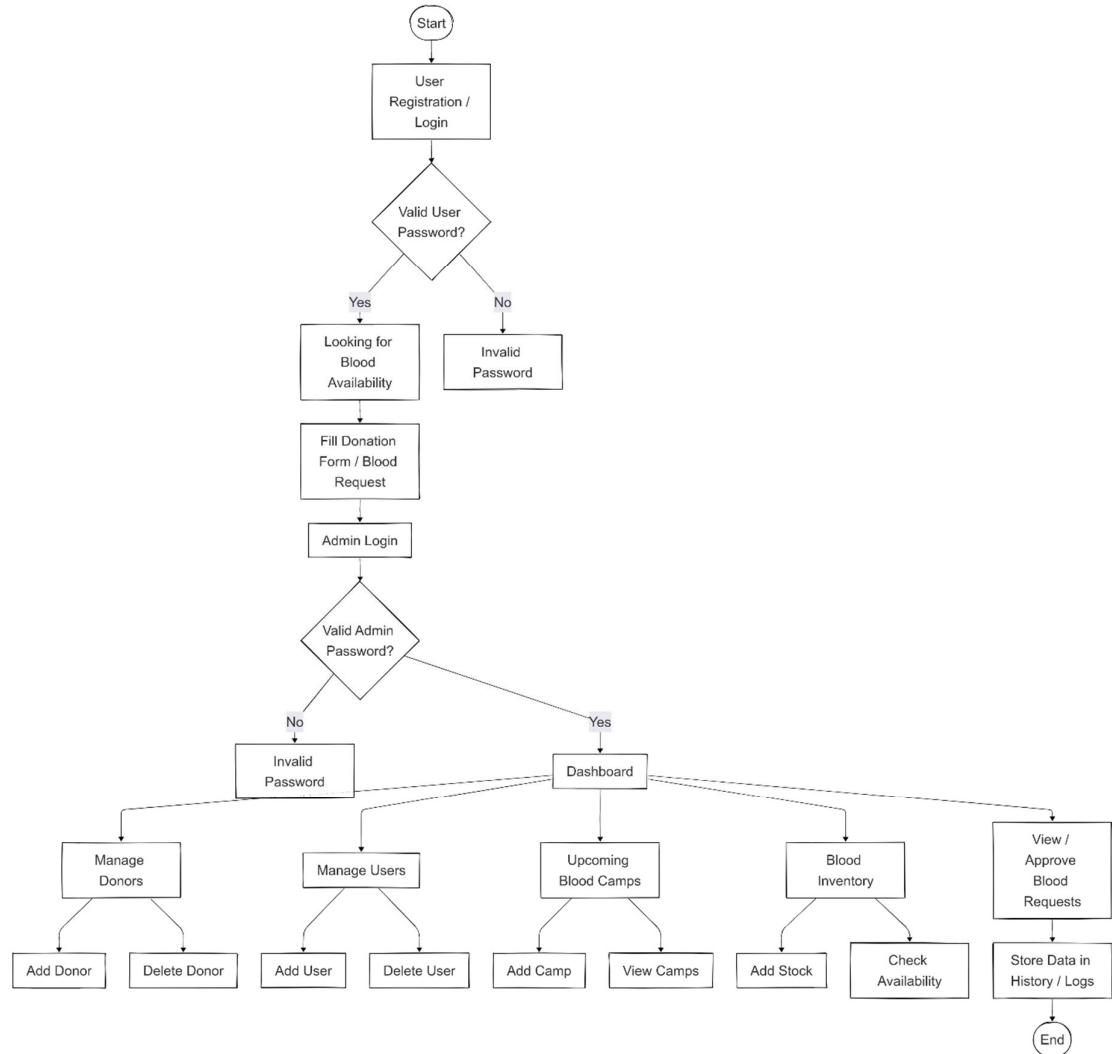


Figure 19 Flowchart of Implementation

This flowchart represents the working of a Blood Bank Management System, divided into two main panels: User Panel and Admin Panel.

The process begins with the user registration or login. If the user provides a valid password, they can proceed to check blood availability and fill out either a donation or a blood request form. If the password is invalid, the user is redirected back.

Next, access to the Admin Panel is granted through an admin login. If the admin password is correct, the system displays a dashboard with several functionalities:

- Manage Donors: Add or delete donor records.
- Manage Users: Add or delete users in the system.
- Upcoming Blood Camps: Add new blood camp details or view existing ones.
- Blood Inventory: Add blood stock and check current availability.

- View/Approve Blood Requests: Review requests and store approved entries in the system history/logs.

If the admin password is incorrect, the system redirects to the home page. The flow ends after successful operations or termination of any path.

This diagram clearly visualizes the logical flow and role-based functionalities within the blood bank system.

8.2 Actual Program Code

9.2.1 Home Page Code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Blood Donation</title>
    <link
        href="https://fonts.googleapis.com/css2?family=Satisfy&display=swap"
        rel="stylesheet">
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f8f9fa;
            text-align: center;
            margin: 0;
            padding: 20px;
        }

        marquee{
            height: 40px;
            font-size: 25px;
        }

        .donation-types {
            background: #fff;
            padding: 30px;
            border-radius: 10px;
            box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
            width: 80%;
            margin: auto;
        }

        .title {
            color: red;
            font-size: 24px;
            text-transform: uppercase;
        }
    </style>

```

```
        }
```

```
.description {
    font-size: 16px;
    color: #333;
    margin-bottom: 20px;
}
```

```
.tabs {
    display: flex;
    justify-content: center;
    gap: 10px;
    margin-bottom: 20px;
}
```

```
.tab-btn {
    padding: 10px 15px;
    border: none;
    cursor: pointer;
    background: #eee;
    border-radius: 5px;
    font-size: 16px;
}
```

```
.tab-btn.active {
    background: red;
    color: white;
    font-weight: bold;
}
```

```
.tab-content {
    display: none;
    text-align: left;
    margin-top: 10px;
}
```

```
.tab-content.active {
    display: block;
}
```

```
h3 {
    color: red;
    font-size: 18px;
}
```

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

```
        color: #333;
    }

    .donate-btn {
        background: red;
        color: white;
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        font-size: 16px;
        cursor: pointer;
        margin-top: 20px;
    }

    .donate-btn:hover {
        background: darkred;
    }

body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background: url('background.png') no-repeat center center fixed;
    background-size: cover;
}

.compatibility {
    text-align: center;
    padding: 30px;
    background-color: rgba(255, 255, 255, 0.9);
    border-radius: 10px;
    margin: 20px auto;
    width: 80%;
    box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.2);
}

.compatibility h2 {
    color: red;
    font-size: 28px;
}

.compatibility p {
    font-size: 16px;
    color: #333;
    margin-bottom: 20px;
}
```

```

table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 10px;
}

th,td {
    border: 2px solid red;
    padding: 10px;
    text-align: center;
    font-size: 18px;
}

th {
    background-color: red;
    color: white;
}

td {
    background-color: #f8d7da;
    font-weight: bold;
}

tr:hover {
    background-color: #f1c5c7;
}

.logo {
    font-size: 50px;
    font-weight: bold;
    margin-left: 20px; /* Moves logo to the left */
    font-family: 'Satisfy', cursive;
    /*color: (rgb(126,53,53),rgb(165, 18, 165));*/
    background: linear-gradient(to right,rgb(245,2,2),rgb(245, 2,
2));
    /* background: linear-gradient(45deg, rgb(68, 52, 238), rgba(81,
37, 223, 0.648));*/
    -webkit-background-clip: text;
    -webkit-text-fill-color: transparent;
}

/* Navbar */
nav {
    background-color: rgb(0, 0, 0);
    background: linear-gradient(45deg,#2f3b45,#2f3b45);

    padding: 20px;
}

```

```
        display: flex;
        align-items: center;
        justify-content: space-between;
    }

    nav ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
        display: flex;
    }

    nav ul li {
        margin: 0 20px;
    }

    nav ul li a {
        color: white;
        text-decoration: none;
        font-weight: bold;
        background: #1A237E;
        padding: 10px 15px;
        border-radius: 5px;
        transition: transform 0.2s ease-in-out, background 0.2s ease-in-out;
        display: inline-block;
    }

    /* Hover Effect */
    nav ul li a:hover {
        background: #424242;
        transform: scale(1.2);
    }

    /* Click Effect */
    nav ul li a:active {
        background: rgb(255, 77, 77);
        transform: scale(1.3);
    }

    /* Banner */
    .banner {
        text-align: center;
        color: rgb(0, 0, 0);
        padding: 50px 0;
        position: relative;
    }
```

```
.slider {  
    width: 100vw;  
    max-width: 900px;  
    height: 600px;  
    margin: auto;  
    overflow: hidden;  
    position: relative;  
}  
  
.slides img {  
    width: 100%;  
    height: 100%;  
    object-fit: cover;  
    flex-shrink: 0;  
}  
  
/* Slider Container */  
.slider {  
    width: 100vw;  
    max-width: 900px;  
    height: 500px;  
    margin: auto;  
    position: relative;  
    overflow: hidden;  
}  
  
/* Slides */  
.slides {  
    display: flex;  
    width: 100%;  
    transition: transform 0.5s ease-in-out;  
}  
  
/* Images */  
.slides img {  
    width: 100%;  
    height: 100%;  
    object-fit: cover;  
    flex-shrink: 0;  
}  
  
/* Navigation Arrows */  
.prev,  
.next {  
    position: absolute;  
    top: 50%;  
}
```

```
        transform: translateY(-50%);  
        background-color: rgba(0, 0, 0, 0.5);  
        color: white;  
        border: none;  
        padding: 15px;  
        cursor: pointer;  
        font-size: 18px;  
        transition: all 0.3s ease-in-out;  
        border-radius: 50%;  
    }  
  
/* Left Arrow */  
.prev {  
    left: -50px;  
    /* Move outside the slider */  
}  
  
/* Right Arrow */  
.next {  
    right: -50px;  
    /* Move outside the slider */  
}  
  
/* Hover Effect */  
.prev:hover,  
.next:hover {  
    background-color: rgba(255, 77, 77, 0.8);  
    transform: scale(1.2);  
    box-shadow: 0px 0px 10px rgba(255, 77, 77, 0.9);  
}  
  
/* Dots */  
.slider-dots {  
    text-align: center;  
    position: absolute;  
    width: 100%;  
    bottom: 10px;  
}  
  
.slider-dots span {  
    height: 12px;  
    width: 12px;  
    margin: 0 5px;  
    background-color: gray;  
    border-radius: 50%;  
    display: inline-block;  
    transition: background 0.3s ease-in-out;
```

```
        cursor: pointer;
    }

    /* Active Dot */
.slider-dots span.active {
    background-color: red;
}

/* Footer */
footer {
    background-color: black;
    color: white;
    padding: 20px;
    text-align: center;
}

.footer-links {
    display: flex;
    justify-content: space-around;
    flex-wrap: wrap;
}

.footer-links div {
    margin: 10px;
}

.footer-links ul {
    list-style-type: none;
    padding: 0;
}

.footer-links ul li a {
    color: white;
    text-decoration: none;
    position: relative;
    transition: color 0.3s;
}

.footer-links ul li a::after {
    content: "";
    position: absolute;
    left: 0;
    bottom: -3px;
    width: 0;
    height: 2px;
    background: white;
    transition: width 0.3s ease-in-out;
```

```
}
```

```
.footer-links ul li a:hover::after {  
    width: 100%;  
}
```

```
.term {  
    font: smaller;  
}
```

```
@media screen and (max-width: 768px) {  
    .main {  
        width: 95%;  
        margin: 80px auto;  
        padding: 1.5rem;  
    }
```

```
    h1 {  
        font-size: 1.8rem;  
    }
```

```
    h3 {  
        font-size: 1.2rem;  
    }
```

```
    .form-group label {  
        font-size: 1rem;  
    }
```

```
    .form-group input {  
        padding: 10px;  
    }
```

```
    .btn {  
        width: 100%;  
        padding: 10px;  
    }  
}
```

```
@media screen and (max-width: 480px) {  
    .main {  
        margin: 60px auto;  
        padding: 1rem;  
    }
```

```
    h1 {  
        font-size: 1.5rem;
```

```

        }

.form-group {
    margin-bottom: 1rem;
}

</style>
</head>

<body>

<header>
    <nav>
        <div class="logo">Lifeline Donor's</div> <!-- Lifeline Donors on Left
-->
        <ul>
            <li><a href="aboutus.html">About Us</a></li>
            <li><a href="lookblood.html">Looking for Blood</a></li>
            <li><a href="wanttodonate.php">Want to Donate Blood</a></li>
            <li><a href="loginpage.html">Login</a></li>
            <li><a href="adminlogin.html">Admin Login</a></li>
            <li><a href="contactus.html">Contact Us</a></li>
        </ul>
    </nav>
</header>

<marquee behavior="scroll" direction="width" bgcolor="Lightyellow"
scrollamount="15">Upcoming Blood Donation Camps : 23 July, 2025 | Time :
1:00 PM | Ahemdabad Civil Hospital

    ♦ 25 August, 2025 | Time : 9:00 AM | Junagadh Civil
Hospital

    ♦ 19 October, 2025 | Time : 10:00 AM | Vapi Civil Hospital
</marquee>

<section class="banner">
    <div class="slider">
        <button class="prev" onclick="moveSlide(-1)">←</button>
        <div class="slides">
            
            
            
            
            
        </div>
    </div>
</section>
```

```

        </div>
        <button class="next" onclick="moveSlide(1)">&#10095;</button>
    </div>
</section>

<section class="compatibility">
    <h2>Compatible Blood Type Donors</h2>
    <p>Find out which blood types are compatible for donation and
transfusion.</p>
    <table>
        <tr>
            <th>Blood Type</th>
            <th>Can Donate To</th>
            <th>Can Receive From</th>
        </tr>
        <tr>
            <td>A+</td>
            <td>A+, AB+</td>
            <td>A+, A-, O+, O-</td>
        </tr>
        <tr>
            <td>A-</td>
            <td>A+, A-, AB+, AB-</td>
            <td>A-, O-</td>
        </tr>
        <tr>
            <td>B+</td>
            <td>B+, AB+</td>
            <td>B+, B-, O+, O-</td>
        </tr>
        <tr>
            <td>B-</td>
            <td>B+, B-, AB+, AB-</td>
            <td>B-, O-</td>
        </tr>
        <tr>
            <td>AB+</td>
            <td>AB+</td>
            <td>All Blood Types</td>
        </tr>
        <tr>
            <td>AB-</td>
            <td>AB+, AB-</td>
            <td>A-, B-, AB-, O-</td>
        </tr>
        <tr>
            <td>O+</td>
        </tr>
    </table>

```

```

<td>O+, A+, B+, AB+</td>
<td>O+, O-</td>
</tr>
<tr>
<td>O-</td>
<td>All Blood Types</td>
<td>O-</td>
</tr>
</table>
</section>

<section class="types-of-donation">
<link rel="stylesheet" href="styles.css">
<section class="donation-types">
<h2 class="title">TYPES OF DONATION</h2>
<p class="description">
    The average human body contains about five liters of blood, which
    is made of several cellular and
    non-cellular components such as
    <strong>Red blood cell, Platelet,</strong> and
    <strong>Plasma.</strong>
</p>
<p class="description">
    Each type of component has its unique properties and can be used
    for different indications. The donated
    blood is separated into these components
    by the blood center, and one donated unit can save up to four lives
    depending on the number of
    components separated from your blood.
</p><br>

<h2 style="color: red;">GuideLine for donors</h2>

<!-- Tab Content -->
<div class="tab-content active" id="red-blood">
<h3>What is it?</h3>
<p>Blood collected straight from the donor into a blood bag and
mixed with an anticoagulant is called
whole blood. This collected whole blood is then centrifuged and
red cells, platelets, and plasma are
separated. The separated red cells are mixed with a preservative to
be called packed red blood
cells.</p>

<h3>Who can donate?</h3>

```

<p>You need to be 18-65 years old, weigh 45kg or more, and be fit and healthy.</p>

<h3>User For?</h3>
<p>Correction of severe anemia in a number of conditions and blood loss in case of childbirth, surgery, or trauma settings.</p>

<h3>Lasts For?</h3>
<p>Red cells can be stored for 42 days at 2-6 degrees Celsius.</p>

<h3>How long does it take to donate?</h3>
<p>15-30 minutes to donate, including the pre-donation check-up.</p>

<h3>How often can I donate?</h3>
<p>Male donors can donate again after 90 days, and female donors can donate again after 120 days.</p>

</div>

<div class="tab-content" id="plasma">
<h3>What is it?</h3>
<p>Plasma is the liquid portion of blood that carries nutrients, hormones, and waste products. It is separated from whole blood and can be frozen for later use.</p>

<h3>Who can donate?</h3>
<p>Individuals aged 18-65, weighing 50kg or more, and in good health.</p>

<h3>User For?</h3>
<p>Used for patients with clotting disorders, liver disease, and severe infections.</p>

<h3>Lasts For?</h3>
<p>Frozen plasma can be stored for up to one year.</p>

<h3>How long does it take to donate?</h3>
<p>30-45 minutes, including plasma separation.</p>

<h3>How often can I donate?</h3>
<p>Every 14 days, up to 24 times a year.</p>

</div>

<div class="tab-content" id="platelet">
<h3>What is it?</h3>

<p>Platelets are tiny blood cells that help form clots to stop bleeding. They are separated from whole blood or collected directly by apheresis.</p>

<h3>Who can donate?</h3>
<p>Healthy individuals aged 18-60, weighing 50kg or more, with no recent medications affecting platelets.</p>

<h3>User For?</h3>
<p>Critical for cancer patients, organ transplant recipients, and individuals with blood disorders.</p>

<h3>Lasts For?</h3>
<p>Platelets can be stored for only 5-7 days.</p>

<h3>How long does it take to donate?</h3>
<p>45-90 minutes.</p>

<h3>How often can I donate?</h3>
<p>Every 7-14 days, up to 24 times a year.</p>

<button class="donate-btn" onclick="window.location.href='nbloodbank.html';">Find Nearest Blood Bank To Donate</button>

</section>

</section>

<footer>

<div class="footer-links">

<div>

<h3>Looking for Blood</h3>

Blood Availability

Blood Bank Directory

</div>

<div>

<h3>Want to Donate Blood</h3>

Blood Donation Camps

Donor Login

```

        </ul>
    </div>
    <div>
        <h3>About Us</h3>
        <ul>
            <li><a href="aboutus.html">About Lifeline Donors</a></li>
            <li><a href="contactus.html">Contact Us</a></li>
        </ul>
    </div>
    </div>
    <p class="term" style="color: red;">Terms & Conditions | Privacy Policy
    | Accessibility Statement | Site Map</p>
</footer>

<script>
let index = 0;
function moveSlide(step) {
    const slides = document.querySelector(".slides");
    const totalSlides = slides.children.length;
    index = (index + step + totalSlides) % totalSlides;
    slides.style.transform = `translateX(-${index * 100}%)`;
}

setInterval(() => moveSlide(1), 5000);
</script>

</body>

</html>

```

9.2.2 Signup Page Code

```

<!DOCTYPE html>
<html lang="en">

    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Sign Up Page</title>
        <link
            href="https://fonts.googleapis.com/css2?family=Satisfy&display=swap"
            rel="stylesheet">

        <style>
            .logo {
                font-size: 50px;
                font-weight: bold;

```

```
margin-left: 20px;
font-family: 'Satisfy', cursive;
background: linear-gradient(to right, rgb(245, 2, 2), rgb(245, 2, 2));
-webkit-background-clip: text;
-webkit-text-fill-color: transparent;
}

body {
    font-family: Arial, sans-serif;
    background-color: #f8f9fa;
    text-align: center;
    margin: 0;
    padding: 0;
    background: url('background.png') no-repeat center center fixed;
    background-size: cover;
}

nav {
    background: linear-gradient(45deg, #2f3b45, #2f3b45);
    padding: 15px;
    margin: -15px;
    display: flex;
    align-items: center;
    justify-content: space-between;
}

.close {
    position: absolute;
    right: 20px;
    top: 15px;
    font-size: 24px;
    cursor: pointer;
    color: black;
    transition: color 0.3s ease;
    z-index: 1;
}

.close:hover {
    color: #ff4444;
}

.main {
    background-color: rgba(255, 255, 255, 0.3);
    backdrop-filter: blur(10px);
    padding: 20px;
    border-radius: 15px;
    width: 400px;
```

```

        margin: 90px auto;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
    }

    input {
        width: 80%;
        padding: 10px;
        margin-top: 5px;
        border: none;
        border-radius: 100px;
        background: rgba(255, 255, 255, 0.5);
    }

    .error {
        color: red;
        font-size: 0.9em;
        margin-top: 5px;
    }

    .bt {
        background: linear-gradient(45deg, #800080, #ff4444);
        color: white;
        padding: 12px 30px;
        border: none;
        border-radius: 25px;
        font-size: 1.1rem;
        font-weight: bold;
        cursor: pointer;
        transition: all 0.3s ease-in-out;
        margin: 1rem 0;
        box-shadow: 0 4px 6px rgba(0, 0, 0, 0.2);
    }

    .bt:hover {
        background: linear-gradient(45deg, #ff4444, #800080);
        transform: scale(1.05);
        box-shadow: 0 8px 15px rgba(0, 0, 0, 0.3);
    }

```

</style>

</head>

<body>

<header>

<nav>

<div class="logo">Lifeline Donor's</div>

</nav>

```

        </header>

        <section class="main">
            <div class="First">
                <h2>Sign Up</h2>

                <span class="close" onclick="window.location.href='Home Page
Code.html'">&times;</span>

                <form id="signupForm" onsubmit="return handleSubmit(event)">
                    <label>Username</label><br>
                    <input type="text" id="Username" name="Username"
placeholder="Enter username">
                    <div class="error" id="usernameError"></div><br>

                    <label>Email ID</label><br>
                    <input type="text" id="EmailId" name="EmailId"
placeholder="Enter Email ID">
                    <div class="error" id="emailError"></div><br>

                    <label>Mobile No.</label><br>
                    <input type="text" id="MobileNum" name="MobileNum"
placeholder="Enter Mobile No.">
                    <div class="error" id="mobileError"></div><br>

                    <label>Address</label><br>
                    <input type="text" id="Address" name="Address"
placeholder="Enter Address">
                    <div class="error" id="addressError"></div><br>

                    <label>Password</label><br>
                    <input type="password" id="Password" name="Password"
placeholder="Enter Password">
                    <div class="error" id="passwordError"></div><br>

                    <label>Confirm Password</label><br>
                    <input type="password" id="ConfirmPassword"
name="ConfirmPassword" placeholder="Enter Confirm Password">
                    <div class="error" id="confirmPasswordError"></div><br>

                <button class="bt" type="submit">Sign Up</button>
            </form>
        </div>
    </section>

    <script>
        function validateForm() {

```

```

let isValid = true;

// Clear all error messages
document.querySelectorAll('.error').forEach(e => e.innerHTML = "");

// Get form values
const username = document.getElementById('Username').value.trim();
const email = document.getElementById('EmailId').value.trim();
const mobile = document.getElementById('MobileNum').value.trim();
const address = document.getElementById('Address').value.trim();
const password = document.getElementById('Password').value;
const confirmPassword =
    document.getElementById('ConfirmPassword').value;

// Username validation
if (username === "") {
    document.getElementById('usernameError').innerHTML =
    "Username is required.";
    isValid = false;
} else if (username.length < 3) {
    document.getElementById('usernameError').innerHTML =
    "Username must be at least 3 characters.";
    isValid = false;
}

// Email validation
const emailPattern = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$/;
if (email === "") {
    document.getElementById('emailError').innerHTML = "Email is required.";
    isValid = false;
} else if (!emailPattern.test(email)) {
    document.getElementById('emailError').innerHTML = "Invalid email format.";
    isValid = false;
}

// Mobile validation
if (mobile === "") {
    document.getElementById('mobileError').innerHTML = "Mobile number is required.";
    isValid = false;
} else if (!/^d{10}\$/test(mobile)) {
    document.getElementById('mobileError').innerHTML = "Mobile number must be 10 digits.";
    isValid = false;
}

```

```

        }

        // Address validation
        if (address === "") {
            document.getElementById('addressError').innerHTML = "Address
is required.";
            isValid = false;
        }

        // Password validation
        if (password === "") {
            document.getElementById('passwordError').innerHTML =
"Password is required.";
            isValid = false;
        } else if (password.length < 6) {
            document.getElementById('passwordError').innerHTML =
>Password must be at least 6 characters.";
            isValid = false;
        }

        // Confirm password validation
        if (confirmPassword === "") {
            document.getElementById('confirmPasswordError').innerHTML =
"Confirm password is required.";
            isValid = false;
        } else if (password !== confirmPassword) {
            document.getElementById('confirmPasswordError').innerHTML =
"Passwords do not match.";
            isValid = false;
        }

        return isValid;
    }

    function redirectToHome() {
        window.location.href = "index.html";
    }

    async function handleSubmit(event) {
        event.preventDefault(); // Stop normal form submission

        if (validateForm()) {
            const formData = new
FormData(document.getElementById('signupForm'));

            try {
                const response = await fetch('connect.php', {

```

```

        method: 'POST',
        body: formData
    });

    if (response.ok) {
        alert("Your account has been created successfully!");
        window.location.href = "loginpage.html";
    } else {
        const error = await response.text();
        alert("Error: " + error);
    }
} catch (error) {
    alert("Network error: " + error);
}
}
}
}
</script>
</body>
</html>

```

9.2.3 Look Blood Page

```

<?php
session_start();
if (!isset($_SESSION['user_id'])) {
    header("Location: loginpage.html"); // redirect to login if not logged in
    exit();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Looking Blood</title>
    <link
        href="https://fonts.googleapis.com/css2?family=Satisfy&display=swap"
        rel="stylesheet">
    <style>
        .logo {
            font-size: 50px;
            font-weight: bold;
            margin-left: 20px;
            font-family: 'Satisfy', cursive;
            background: linear-gradient(to right, rgb(245, 2, 2), rgb(245, 2, 2));
            -webkit-background-clip: text;

```

```
-webkit-text-fill-color: transparent;
}
nav {
    background-color: black;
    padding: 20px;
    margin: -20px;
    display: flex;
    align-items: center;
    justify-content: space-between;
}
body {
    font-family: Arial, sans-serif;
    background: url('background.png') no-repeat center center fixed;
    background-size: cover;
    margin: 0;
    padding: 20px;
    text-align: center;
}
.main1, .main2 {
    background-color: rgba(255, 255, 255, 0.3);
    backdrop-filter: blur(10px);
    padding: 20px;
    border-radius: 15px;
    width: 90%;
    max-width: 900px;
    margin: 50px auto;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
}
.main3 {
    background-color: rgba(255, 255, 255, 0.3);
    backdrop-filter: blur(10px);
    padding: 20px;
    border-radius: 15px;
    width: 100%;
    max-width: 900px;
    margin: 50px auto;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
}
.order-confirmation {
    background-color: rgba(255, 255, 255, 0.3);
    backdrop-filter: blur(10px);
    padding: 20px;
    border-radius: 15px;
    width: 90%;
    max-width: 900px;
    margin: 20px auto;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
```

```

        }

.form-control, select, .user-input {
    font-family: Calibri;
    width: 30%;
    padding: 10px;
    margin: 5px auto;
    border: none;
    border-radius: 50px;
    background: rgba(255, 255, 255, 0.402);
}

.q{
    width: 50px;
}
.form-control, select {
    font-family: Calibri;
    width: 30%;
    padding: 10px;
    margin: 5px auto;
    border: none;
    border-radius: 50px;
    background: rgba(255, 255, 255, 0.402);
}
.bt {
    background: linear-gradient(45deg, #800080, #ff4444);
    color: white;
    padding: 12px 30px;
    border: none;
    border-radius: 25px;
    font-size: 1.1rem;
    font-weight: bold;
    cursor: pointer;
    transition: all 0.3s ease-in-out;
    margin-top: 1rem;
}
.bt:hover {
    background: linear-gradient(45deg, #ff4444, #800080);
    transform: scale(1.05);
}
table {
    width: 100%;
    border-collapse: collapse;
}
th, td {
    border: 1px solid black;
    padding: 10px;
}

```

```

        }
    </style>
</head>
<body>
<header>
    <nav>
        <div class="logo">Lifeline Donor's</div>
    </nav>
</header>

<section class="main1">
    <h2>Blood Storage Unit (BSU)</h2>
    <label>District</label><br>
    <select id="district" class="form-control" onchange="updateCities()">
        <option>Select District</option>
    </select><br>

    <label>City</label><br>
    <select id="city" class="form-control">
        <option>Select City</option>
    </select><br>

    <button class="bt" onclick="filterResults()">Search</button>
</section>

<section class="main2">
    <table>
        <thead>
            <tr>
                <th>Hospital Name</th>
                <th>A+</th>
                <th>A-</th>
                <th>B+</th>
                <th>B-</th>
                <th>AB+</th>
                <th>AB-</th>
                <th>O+</th>
                <th>O-</th>
                <th>Select Hospital</th>
            </tr>
        </thead>
        <tbody id="bloodBankTable">
            <tr><td colspan="10">No data available</td></tr>
        </tbody>
    </table>
</section>

```

```

<section class="main3">
  <table>
    <thead>
      <tr>
        <th>Hospital Name</th>
        <th>A+</th>
        <th>A-</th>
        <th>B+</th>
        <th>B-</th>
        <th>AB+</th>
        <th>AB-</th>
        <th>O+</th>
        <th>O-</th>
      </tr>
    </thead>
    <tbody id="bloodBankTable">
      <tr>
        <td id="selectedHospitalName"></td>
        <td><input class="q" type="number" name="quantity"
style="font-family: Calibri; width: 60%; padding: 10px; margin-top:
5px; border: none; border-radius: 50px; background:
rgb(255,255,255,0.402);"></td>
        <td><input class="q" type="number" name="quantity"
style="font-family: Calibri; width: 60%; padding: 10px; margin-top:
5px; border: none; border-radius: 50px; background:
rgb(255,255,255,0.402);"></td>
        <td><input class="q" type="number" name="quantity"
style="font-family: Calibri; width: 60%; padding: 10px; margin-top:
5px; border: none; border-radius: 50px; background:
rgb(255,255,255,0.402);"></td>
        <td><input class="q" type="number" name="quantity"
style="font-family: Calibri; width: 60%; padding: 10px; margin-top:
5px; border: none; border-radius: 50px; background:
rgb(255,255,255,0.402);"></td>
        <td><input class="q" type="number" name="quantity"
style="font-family: Calibri; width: 60%; padding: 10px; margin-top:
5px; border: none; border-radius: 50px; background:
rgb(255,255,255,0.402);"></td>
        <td><input class="q" type="number" name="quantity"
style="font-family: Calibri; width: 60%; padding: 10px; margin-top:
5px; border: none; border-radius: 50px; background:
rgb(255,255,255,0.402);"></td>
        <td><input class="q" type="number" name="quantity"
style="font-family: Calibri; width: 60%; padding: 10px; margin-top:
5px; border: none; border-radius: 50px; background:
rgb(255,255,255,0.402);"></td>
    
```

```

        <td><input class="q" type="number" name="quantity"
style="font-family: Calibri; width: 60%; padding: 10px; margin-top:
5px; border: none; border-radius: 50px; background:
rgb(255, 255, 255, 0.402);"></td>
      </tr>
    </tbody>
  </table>

  <button class="bt" id="orderBtn" name="purchase">Order
Now</button>
</section>

<section class="order-confirmation" id="orderConfirmation">
  <h2>Confirm Your Order</h2>
  <div>
    <h3>Hospital: <span id="confirmHospitalName"></span></h3>
    <table>
      <thead>
        <tr><th>Blood Group</th><th>Quantity</th></tr>
      </thead>
      <tbody id="orderSummary"></tbody>
    </table>
  </div>
  <div style="margin: 20px 0;">
    <input type="text" id="customerName" class="user-input"
placeholder="Your Name">
    <br>
    <input type="tel" id="customerMobile" class="user-input"
placeholder="Mobile Number">
  </div>
  <button class="bt" id="confirmOrderBtn">Confirm Order</button>
  <button class="bt" onclick="hideConfirmation()">Cancel</button>
</section>

<script>
  const districtData = {
    "Ahmedabad": ["Ahmedabad City", "Bavla", "Daskroi",
      "Dhandhuka", "Dholera", "Dholka", "Mandal", "Sanand", "Viramgam"],
    "Amreli": ["Amreli", "Bagasara", "Dhari", "Jafrabad", "Khambha",
      "Kukavav", "Lathi", "Rajula", "Savarkundla"],
    "Anand": ["Anand", "Borsad", "Petlad", "Khambhat", "Sojitra",
      "Tarapur", "Umreth", "Anklav"],
    "Aravalli": ["Bayad", "Bhiloda", "Dhansura", "Malpur", "Meghraj",
      "Modasa"],
    "Banaskantha": ["Amirgadh", "Bhabhar", "Danta", "Deesa",
      "Dhanera", "Dantiwada", "Kankrej", "Lakhani", "Palanpur", "Suigam",
      "Tharad", "Vadgam", "Vav"]
  }
</script>

```

"Bharuch": ["Amod", "Ankleshwar", "Bharuch", "Hansot",
 "Jambusar", "Jhagadia", "Netrang", "Vagra", "Valia"],
 "Bhavnagar": ["Bhavnagar", "Botad", "Gadhada", "Ghogha",
 "Jesar", "Mahuva", "Palitana", "Sihor", "Talaja", "Umrala"],
 "Botad": ["Botad", "Gadhada", "Barvala", "Ranpur"],
 "Chhota Udaipur": ["Bodeli", "Chhota Udaipur", "Kavant",
 "Naswadi", "Pavi Jetpur", "Sankheda"],
 "Dahod": ["Dahod", "Devgad Baria", "Dhanpur", "Fatepura",
 "Garbada", "Limkheda", "Santrampur", "Jhalod"],
 "Dang": ["Ahwa", "Subir", "Waghai"],
 "Devbhoomi Dwarka": ["Bhanvad", "Dwarka", "Khambhalia",
 "Kalyanpur"],
 "Gandhinagar": ["Dehgam", "Gandhinagar", "Kalol", "Mansa"],
 "Gir Somnath": ["Gir Gadhada", "Kodinar", "Sutrapada", "Talala",
 "Una", "Veraval"],
 "Jamnagar": ["Dhrol", "Jamjodhpur", "Jamnagar", "Jodiya",
 "Kalavad", "Lalpur"],
 "Junagadh": ["Bhesan", "Junagadh", "Keshod", "Malia",
 "Manavadar", "Mangrol", "Mendarda", "Vanthali", "Visavadar"],
 "Kheda": ["Kheda", "Kapadvanj", "Kathlal", "Matar",
 "Mehmedabad", "Mahudha", "Nadiad", "Thasra", "Virpur"],
 "Kutch": ["Abdasa", "Anjar", "Bhachau", "Bhuj", "Gandhidham",
 "Lakhpat", "Mandvi", "Mundra", "Nakhatrana", "Rapar"],
 "Mahisagar": ["Balasinor", "Kadana", "Khanpur", "Lunawada",
 "Santrampur", "Virpur"],
 "Mehsana": ["Becharaji", "Jotana", "Kadi", "Kheralu", "Mehsana",
 "Satlasana", "Unjha", "Vijapur", "Visnagar"],
 "Morbi": ["Halvad", "Morbi", "Maliya", "Tankara", "Wankaner"],
 "Narmada": ["Dediapada", "Garudeshwar", "Nandod", "Sagbara",
 "Tilakwada"],
 "Navsari": ["Chikhli", "Gandevi", "Jalalpore", "Khergam",
 "Navsari", "Vansda"],
 "Panchmahal": ["Ghoghamba", "Godhra", "Halol", "Jambughoda",
 "Kalol", "Morwa Hadaf", "Shehera"],
 "Patan": ["Chanasma", "Harij", "Patan", "Radhanpur", "Sami",
 "Sidhpur"],
 "Porbandar": ["Kutiyana", "Porbandar", "Ranavav"],
 "Rajkot": ["Dhoraji", "Gondal", "Jamkandorna", "Jasdan", "Jetpur",
 "Kotda Sangani", "Lodhika", "Morbi", "Rajkot", "Upleta", "Wankaner"],
 "Sabarkantha": ["Bayad", "Dhansura", "Himatnagar", "Idar",
 "Khedbrahma", "Poshina", "Prantij", "Talod", "Vijaynagar"],
 "Surat": ["Bardoli", "Choryasi", "Kamrej", "Mahuva", "Mandvi",
 "Mangrol", "Olpad", "Palsana", "Umarpada"],
 "Surendranagar": ["Chotila", "Dhrangadhra", "Halvad", "Lakhtar",
 "Limbdi", "Muli", "Patdi", "Sayla", "Thangadh", "Wadhwan"],
 "Tapi": ["Dolvan", "Nizar", "Songadh", "Uchchhal", "Valod",
 "Vyara"],
 "Vadodara": ["Vadodara", "Baroda", "Bopal", "Naroda", "Rajkot", "Talod", "Vadnagar", "Vapi"],
 "Vapi": ["Vapi", "Vadodara", "Baroda", "Bopal", "Naroda", "Rajkot", "Talod", "Vadnagar"]

```

        "Vadodara": ["Dabhoi", "Karjan", "Padra", "Savli", "Vadodara",
    "Waghodia"],
    "Valsad": ["Dharampur", "Kaprada", "Pardi", "Umbergaon",
    "Valsad", "Vapi"]
};

window.onload = () => {
    const district = document.getElementById("district");
    for (let key in districtData) {
        const opt = document.createElement("option");
        opt.value = key;
        opt.textContent = key;
        district.appendChild(opt);
    }
};

function updateCities() {
    const city = document.getElementById("city");
    city.innerHTML = '<option>Select City</option>';
    const selected = document.getElementById("district").value;
    if (districtData[selected]) {
        districtData[selected].forEach(c => {
            let opt = document.createElement("option");
            opt.value = c;
            opt.textContent = c;
            city.appendChild(opt);
        });
    }
}

function filterResults() {
    let district = document.getElementById("district").value;
    let city = document.getElementById("city").value;

    if (district === "Select District" || city === "Select City") {
        alert("Please select district and city");
        return;
    }

    fetch(`lookblood.php?district=${encodeURIComponent(district)}&cit
y=${encodeURIComponent(city)}`)
        .then(res => res.json())
        .then(data => {
            const table = document.getElementById("bloodBankTable");
            table.innerHTML = "";

            if (data.length === 0) {

```

```

        table.innerHTML = `<tr><td colspan="9">No data
found</td></tr>`;
        return;
    }

    data.forEach(row => {
        table.innerHTML += `
<tr>
    <td>${row["Hospital Name"]}</td>
    <td>${row["A+"]}</td>
    <td>${row["A-"]}</td>
    <td>${row["B+"]}</td>
    <td>${row["B-"]}</td>
    <td>${row["AB+"]}</td>
    <td>${row["AB-"]}</td>
    <td>${row["O+"]}</td>
    <td>${row["O-"]}</td>
    <td><button class='bt'
onclick="selectHospital('${row['Hospital Name']}')">Purchase
Blood</button></td>
    </tr>
`;
    });
})
.catch(error => {
    console.error("Fetch error:", error);
});
}

function selectHospital(name) {
    document.getElementById("selectedHospitalName").textContent = name;
    document.querySelector(".main3").scrollIntoView({ behavior: "smooth"
});
}

document.getElementById("orderBtn").addEventListener("click", () => {
    const hospital =
document.getElementById("selectedHospitalName").textContent;
if (!hospital) {
    alert("Please select a hospital first.");
    return;
}

const inputs = document.querySelectorAll(".main3 input.q");
const bloodTypes = ["A+", "A-", "B+", "B-", "AB+", "AB-", "O+", "O-"];
let order = [];

```

```

inputs.forEach((input, i) => {
  const qty = parseInt(input.value) || 0;
  if (qty > 0) {
    order.push({ bloodType: bloodTypes[i], quantity: qty });
  }
});

if (order.length === 0) {
  alert("Please select at least one unit.");
  return;
}

// Fill confirmation section
document.getElementById("confirmHospitalName").textContent = hospital;
const summary = document.getElementById("orderSummary");
summary.innerHTML = "";
order.forEach(item => {
  summary.innerHTML +=
`<tr><td>${item.bloodType}</td><td>${item.quantity}</td></tr>`;
});

document.getElementById("orderConfirmation").scrollIntoView({
behavior: "smooth"
});

document.getElementById("confirmOrderBtn").addEventListener("click",
function() {
  const name = document.getElementById("customerName").value;
  const mobile = document.getElementById("customerMobile").value;
  const hospital =
document.getElementById("confirmHospitalName").textContent;

  if (!name || !mobile) {
    alert("Please fill in all fields");
    return;
  }

  // Get quantities from the order summary table
  const quantities = {};
  const rows =
document.getElementById("orderSummary").getElementsByName("tr");
  for (let row of rows) {
    const cells = row.getElementsByName("td");
    if (cells.length === 2) {
      quantities[cells[0].textContent] = parseInt(cells[1].textContent) || 0;
    }
  }
}
);

```

```

// Create form data
const formData = new FormData();
formData.append('name', name);
formData.append('mobile', mobile);
formData.append('hospital', hospital);
formData.append('quantities', JSON.stringify(quantities));

// Send data to server
fetch('store_order.php', {
    method: 'POST',
    body: formData
})
.then(response => response.json())
.then(data => {
    if (data.success) {
        alert("Your blood purchase request has been placed Successfully");
        window.location.href = "Home Page Code.html";
    } else {
        alert("Error placing order: " + (data.error || "Unknown error"));
    }
})
.catch(error => {
    console.error('Error:', error);
    alert("Error placing order. Please try again.");
});
});

function hideConfirmation() {
    document.getElementById("orderConfirmation").style.display = "none";
}

</script>

</body>
</html>

```

9.2.4 Want to Donate Blood Page Code

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Looking Blood</title>

```

```
<link
  href="https://fonts.googleapis.com/css2?family=Satisfy&display=swap"
  rel="stylesheet">
<style>
  .logo {
    font-size: 50px;
    font-weight: bold;
    margin-left: 20px;
    font-family: 'Satisfy', cursive;
    background: linear-gradient(to right, rgb(245, 2, 2), rgb(245, 2, 2));
    -webkit-background-clip: text;
    -webkit-text-fill-color: transparent;
  }
  nav {
    background-color: black;
    padding: 20px;
    margin: -20px;
    display: flex;
    align-items: center;
    justify-content: space-between;
  }
  body {
    font-family: Arial, sans-serif;
    background: url('background.png') no-repeat center center fixed;
    background-size: cover;
    margin: 0;
    padding: 20px;
    text-align: center;
  }
  .main1, .main2{
    background-color: rgba(255, 255, 255, 0.3);
    backdrop-filter: blur(10px);
    padding: 20px;
    border-radius: 15px;
    width: 90%;
    max-width: 900px;
    margin: 50px auto;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
  }
  .main3{
    background-color: rgba(255, 255, 255, 0.3);
    backdrop-filter: blur(10px);
    padding: 20px;
    border-radius: 15px;
    width: 100%;
    max-width: 900px;
    margin: 50px auto;
  }
```

```
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
    }
.q{
    width: 50px;
}
.form-control, select {
    font-family: Calibri;
    width: 30%;
    padding: 10px;
    margin: 5px auto;
    border: none;
    border-radius: 50px;
    background: rgba(255, 255, 255, 0.402);
}
.bt {
    background: linear-gradient(45deg, #800080, #ff4444);
    color: white;
    padding: 12px 30px;
    border: none;
    border-radius: 25px;
    font-size: 1.1rem;
    font-weight: bold;
    cursor: pointer;
    transition: all 0.3s ease-in-out;
    margin-top: 1rem;
}
.bt:hover {
    background: linear-gradient(45deg, #ff4444, #800080);
    transform: scale(1.05);
}
table {
    width: 100%;
    border-collapse: collapse;
}
th, td {
    border: 1px solid black;
    padding: 10px;
}
</style>
</head>
<body>
<header>
<nav>
    <div class="logo">Lifeline Donor's</div>
</nav>
</header>
```

```

<section class="main1">
  <h2>Want to Donate blood</h2>
  <table>
    <thead>
      <tr>
        <th>Sr NO.</th>
        <th>Location</th>
        <th>Date & Time</th>
        <th>Donate Blood</th>
      </tr>
    </thead>
    <tbody id="bloodBankTable">
      <tr>
        <td>1</td>
        <td>Ahemdabad Civil Hospital</td>
        <td>23 July,2025 <br>Time : 1:00 pm</td>
        <td><button class="bt" onclick="showDonationInfo('Ahemdabad Civil Hospital', '23 July, 2025 - 1:00 pm')">Donate</button></td>
      </tr>
      <tr>
        <td>2</td>
        <td>Junagadh Civil Hospital</td>
        <td>25 August,2025 <br>Time : 9:00 am</td>
        <td><button class="bt" onclick="showDonationInfo('Junagadh Civil Hospital', '25 August, 2025 - 9:00 am')">Donate</button></td>
      </tr>
      <tr>
        <td>3</td>
        <td>Vapi Civil Hospital</td>
        <td>19 October,2025 <br>Time : 10:00 am</td>
        <td><button class="bt" onclick="showDonationInfo('Vapi Civil Hospital', '19 October, 2025 - 10:00 am')">Donate</button></td>
      </tr>
    </tbody>
  </table>
</section>

<section class="main3">
  <h2>Enter your Details</h2>
  <!-- AJAX populated info -->
  <div id="donationDetailsBox" style="margin-bottom: 20px;"></div>

  <form action="Showinfo.php" method="post">
    <input type="hidden" name="hospital" id="hospitalInput">
    <input type="hidden" name="datetime" id="datetimeInput">

```

```

<label>Name</label><br>
<input type="text" placeholder="Enter your Name" name="uname"
required
    style="font-family: Calibri; width: 30%; padding: 10px; margin-
top: 5px; border: none; border-radius: 50px; background:
rgb(255,255,255,0.402);"><br><br>
<label>Age</label><br>
<input type="number" placeholder="Enter your Age" name="age"
min="18" max="60" required
    style="font-family: Calibri; width: 30%; padding: 10px; margin-
top: 5px; border: none; border-radius: 50px; background:
rgb(255,255,255,0.402);"><br><br>
<label>Mobile Number</label><br>
<input type="number" placeholder="Enter your Mobile Number"
name="mno" required
    style="font-family: Calibri; width: 30%; padding: 10px; margin-
top: 5px; border: none; border-radius: 50px; background:
rgb(255,255,255,0.402);"><br><br>
    <button class="bt" type="submit">Submit</button>
</form>
</section>

<script>
    function showDonationInfo(hospitalName, dateTime) {
        // Populate visible details
        const detailsBox = document.getElementById("donationDetailsBox");
        detailsBox.innerHTML =
            `<p><strong>Hospital:</strong> ${hospitalName}</p>
             <p><strong>Date & Time:</strong> ${dateTime}</p>
            `;
        // Populate hidden inputs
        document.getElementById("hospitalInput").value = hospitalName;
        document.getElementById("datetimeInput").value = dateTime;

        // Scroll to form section
        detailsBox.scrollIntoView({ behavior: "smooth" });
    }
</script>
</body>
</html>

```

9. TESTING

9.1 User Module

Test Case Number	Test Case	Expected Result	Status
TC-01	User account creation	Account has been Sucessfully	✓
TC-02	User Login	Login sucessfully	✓
TC-03	Mobile Number in a Valid Format	Mobile Number must be 10 digit	✓
TC-04	Password in a Valid Format	Password Must be of 8 characters	✓
TC-05	Reset Password	Password Changed sucessfully	✓
TC-06	Looking for a Blood Availability	Display the Available Stock of Blood Bottles	✓
TC-07	User can request for a Blood bottles	Your blood request send sucessfully	✓
TC-08	User can register for blood donation camp	Your blood donation request has been send to admin	✓
TC-09	User can Contact us	Your message has been send Sucessfully	✓

Table 7 User Module Table

9.2 Admin Module

Test Case Number	Test Case	Expected Result	Status
TC-10	Admin Login	Admin login has been authenticated successfully and redirect to admin dashboard	✓
TC-11	Security Key Validation	Invalid Security key	✓
TC-12	Manage users	Admin can delete and Add new user	✓
TC-13	Manage donors	Admin can delete donors	✓
TC-14	Blood Inventory	Admin can Add hospital and update the blood availability stocks	✓
TC-15	Manage Camps	Admin can schedule an Event of upcoming blood camps	✓

TC-16	Manage blood Request	Admin can view/ approve the blood request	✓
TC-17	Report analysis	Admin can generate report in PDF format with the data of total registered users, registered donors, total blood units	✓

Table 8 Admin Module Table

10.USER MANUAL

10.1 Installation Steps (Used Software/Project/Deployment steps of System)

Required Software Tools:

- Visual Studio Code
- Php v 8.2.12
- XAMPP Control Panel v 3.3.0
- Any Browser like Chrome, Microsoft Edge, etc.
- Github Desktop
- Internet Connectivity

1. Download the Visual Code.
 - Download and install Visual Studio Code from the official Website :
<https://code.visualstudio.com/download>
 - And Setup the Visual Studio Code.
2. Download the PHP 8.2.12
<https://www.php.net/downloads.php>
3. Download XAMPP Control Panel v 3.3.0
 - Download and install XAMPP Control Panel from official Website :
<https://www.apachefriends.org/download.html>
 - And Setup the XAMPP Control Panel.
4. Use Localhost in PhpMyAdmin And Localhost for Run Php File.
<http://localhost/phpmyadmin/>
http://localhost/vasu/Lifeline_Donors-main/Lifeline_Donors-main/
5. Download Github Desktop for develop project in team using Create repository.
 - Get the project source Code from the version control system (Github) or shared Link.
 - And use the Github Desktop App. Download and install from official Website :
<https://desktop.github.com/download/>
 - Open the project folder in Visual Studio Code.

10.2 Sanpshorts with explanation

Snapshot 1 : Signup Page

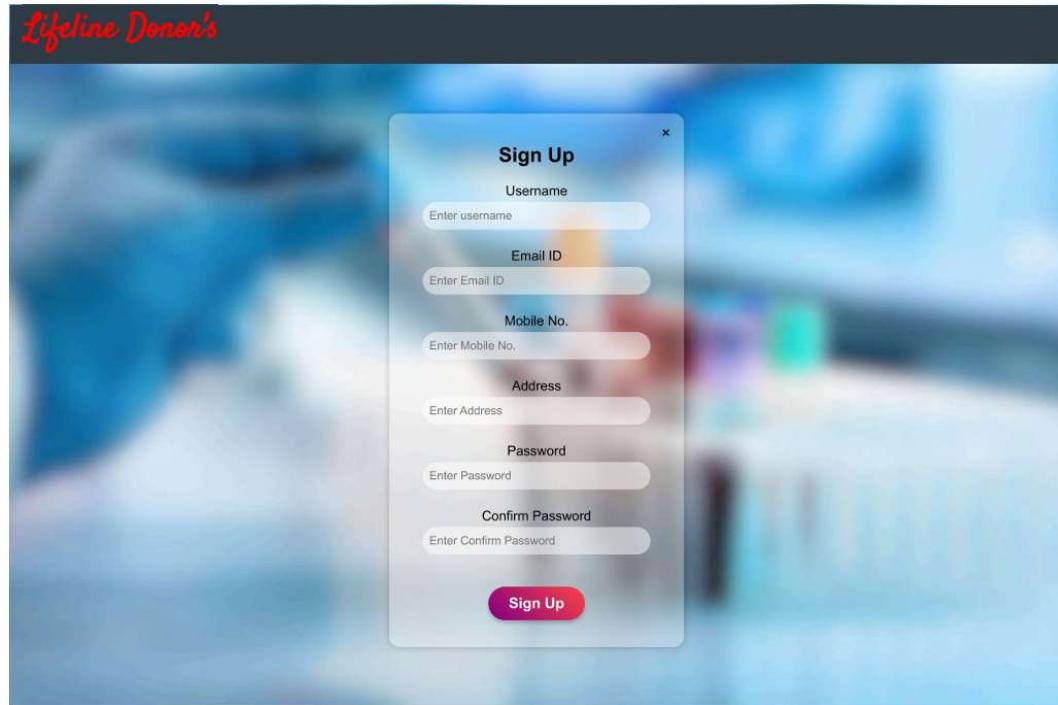


Figure 20 Sign up Page

For new users to register with the **Lifeline Donor's** blood bank management system. It prompts the user to input essential credentials, specifically a username, email ID, mobile number, address, and password to create a new account.

The user interface (UI) is crafted using **HTML and CSS**, with input fields for capturing user details. The password and confirm password fields include features such as **text masking** to enhance data security during entry.

A **Sign Up button** is provided to trigger the registration process. The design of the form adopts a **glassmorphism theme**, maintaining a modern and visually engaging look that aligns with the app's branding using gradients, transparency, and custom fonts.

PhpMyAdmin play crucial roles in handling and storing user credentials. When the user submits the form, Php processes the input and insert the data into the PhpMyAdmin database using insert statement.

If registration is successful, the user receives a confirmation message and can then log in to the platform. If the input is invalid or the email already exists, appropriate error messages are displayed, prompting correction. This ensures only valid data is stored in the system.

- **Security Features:**

The password fields are masked, and data is securely stored in the backend using PHP.

- **Database Integration:**

User information is inserted into a users table using PHP and MySQL in phpMyAdmin.

- **Form Validation:**

Basic HTML5 validation ensures required fields are not left blank; further validation can be added in PHP or JavaScript.

Snapshot 2 : Login Page

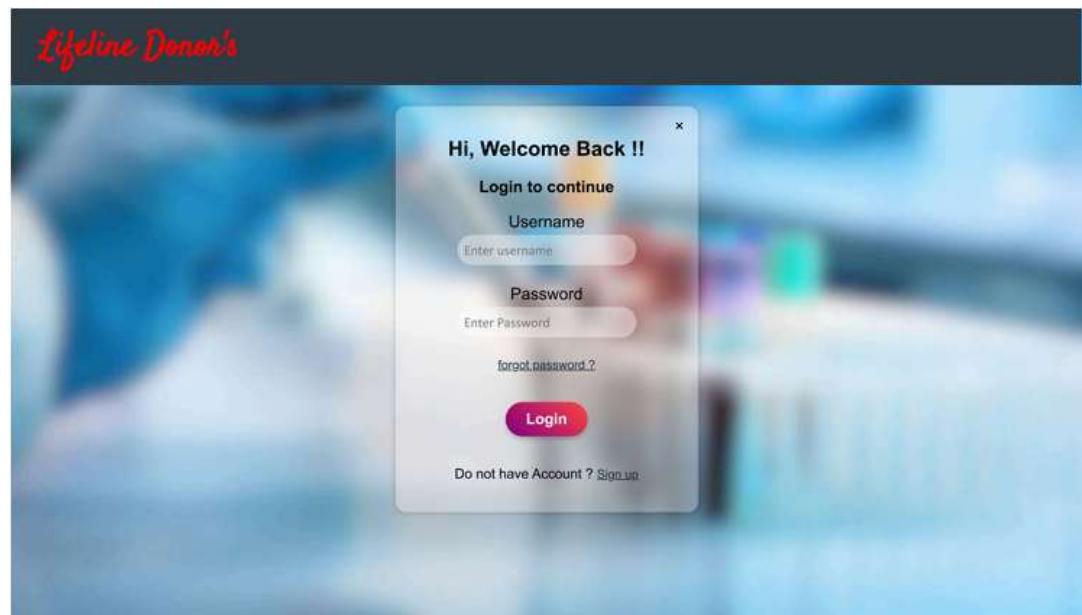


Figure 21 Login Page

The Login Screen serves as the secure access point for users to log in to the Lifeline Donor's system. It ensures that only **registered users** can interact with key functionalities such as viewing blood availability, placing requests, or registering for blood donation.

This screen includes input fields for:

- **Username**
- **Password**

Both fields are designed using standard **HTML input tags** styled with modern **glassmorphism aesthetics** for a clean, minimal UI. The **password field** uses masking to hide input for security purposes.

Below the fields:

- A “**forgot password?**” link allows users to recover their account (future enhancement).

- A “**Login**” button triggers backend **PHP code** that verifies credentials by querying the **MySQL database** (e.g., users table).

If the username and password match, the user is redirected to the main dashboard or home page. Otherwise, an error message prompts re-entry.

New users can click the “**Sign up**” link at the bottom, redirecting them to the registration form.

User Authentication:

Validates login credentials against the database using secure PHP scripts.

Secure Input Fields:

Password input is masked; form uses basic frontend validation.

Navigation Options:

Users can recover passwords or register for a new account directly from this screen.

Database Interaction:

PHP SELECT queries are used to verify the entered credentials from the users table.

Snapshot 3 : Looking for a Blood

The screenshot shows two screens of a mobile application. The top screen is titled "Blood Storage Unit (BSU)" and contains fields for "District" (with dropdown placeholder "Select District") and "City" (with dropdown placeholder "Select City"), followed by a "Search" button. Below this is a table with columns for "Hospital Name" and blood types (A+, A-, B+, B-, AB+, AB-, O+, O-), with a "Select Hospital" column at the end. A message "No data available" is displayed. The bottom screen is titled "Looking for a blood" and contains a table with columns for "Hospital Name" and blood types (A+, A-, B+, B-, AB+, AB-, O+, O-). It includes input fields for "Name" (placeholder "Enter name") and "Mobile No." (placeholder "Enter Mobile No."), and a "Order Now" button at the bottom.

Figure 22 Looking for a blood Page

The Blood Storage Unit (BSU) screen

is designed to help users **search and request available blood units** based on their location. It acts as the main interface for accessing real-time blood stock data in hospitals across various cities and districts.

The top section of the screen contains **dropdown menus** for selecting the **District** and **City**. Once a selection is made, the user can click the **Search** button to retrieve available data. This dynamically displays a **hospital-wise blood availability table** showing quantities for each blood group (A+, A-, B+, B-, AB+, AB-, O+, O-).

Below the search result, users can view another panel where they can select a hospital and request specific blood units. The user interface provides **radio buttons or selectors** to choose blood types and enter personal details like **Name** and **Mobile Number**.

The “**Order Now**” button submits the request. This interaction is powered by **PHP and MySQL**, where:

- Blood availability is fetched using SQL SELECT queries based on city/district filters.
- Blood orders are processed using INSERT queries into an **orders** table.

The design maintains a **glassmorphism visual style** for consistency, using soft backgrounds, rounded edges, and vibrant buttons to enhance usability and appearance.

Location-Based Filtering:

Users can narrow their search using district and city dropdowns to locate nearby blood banks.

Request Functionality:

Users can place orders by selecting a hospital and entering minimal personal details.

Snapshot 4 : You Want to Donate Blood

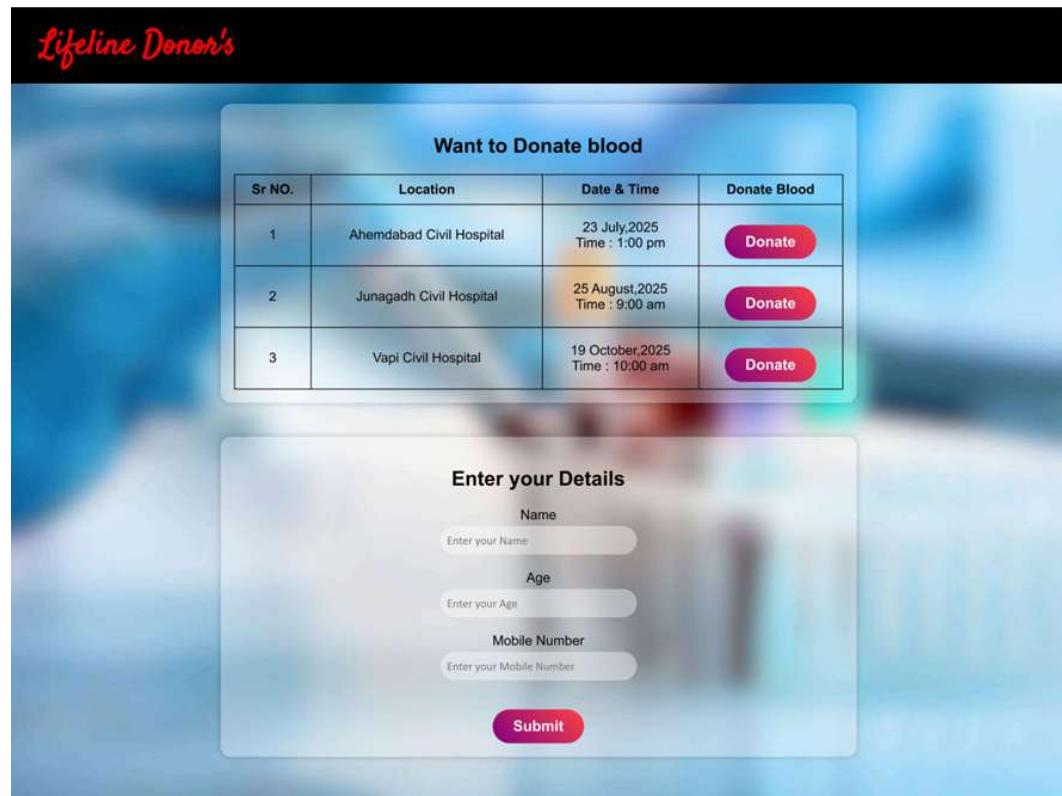


Figure 23 You Want to donate blood Page

The "Want to Donate Blood" screen is designed to allow volunteers to register as blood donors by choosing a hospital, date, and time convenient for their donation. This feature encourages community participation and makes donor registration simple and efficient.

The top section displays a data table listing:

- Serial Number
- Location (Hospital Name)
- Date & Time of the donation event
- A Donate button for each entry

The user selects their preferred hospital and donation time by clicking the Donate button next to the desired row. Upon clicking, the screen scrolls or directs the user to the form section below.

The form includes fields to enter the donor's Name, Age, and Mobile Number. These inputs are necessary for confirming the donor's identity and ensuring they meet donation criteria. The Submit button finalizes the registration process.

The data entered is then sent to a PHP script, which stores the information in a MySQL database table (e.g., donors or donation_registrations). Backend validation checks for valid input (e.g., age limit, non-empty fields).

The user interface follows the glassmorphism style, ensuring a seamless and visually appealing experience with translucent panels and gradient buttons.

Hospital Event Selection:

Donors can choose from a list of upcoming blood donation events at different locations.

Simple Registration Form:

Basic form captures necessary donor information to facilitate contact and eligibility.

Snapshot 5 : About Us Page

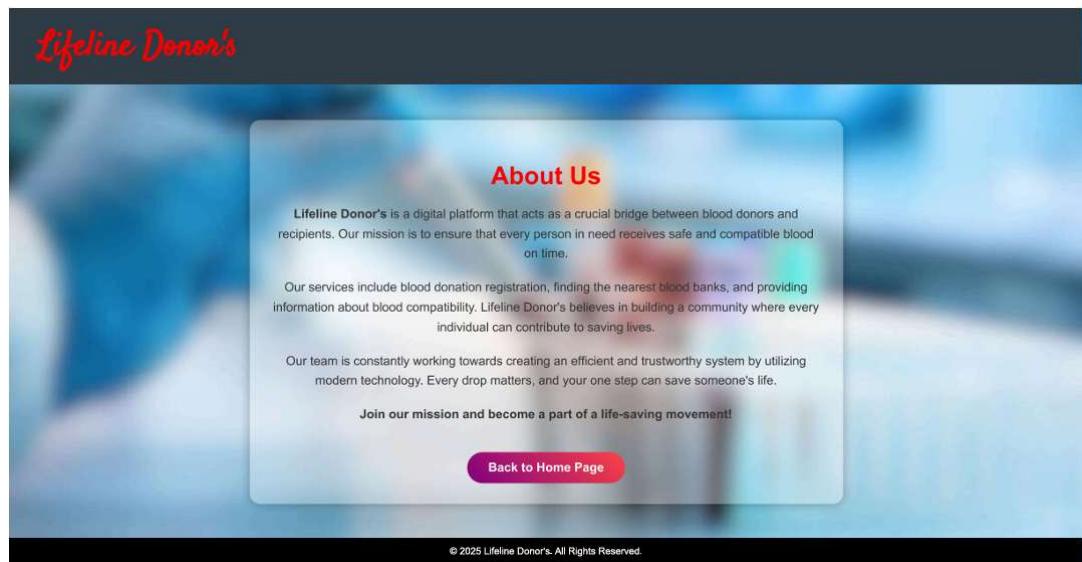


Figure 24 About Us Page

The **About Us** section offers users a clear and concise understanding of the mission and vision behind the **Lifeline Donor's** platform. It serves to inform users about the purpose of the system and build trust by highlighting the team's goals and values.

This screen includes:

- A **brief introduction** to the platform's role as a bridge between blood donors and recipients.
- A **mission statement** emphasizing timely and safe access to compatible blood.
- A summary of **core services** such as:
 - Blood donation registration
 - Locating the nearest blood banks

- o Information on blood group compatibility

The UI is built with a clean, modern glassmorphic design, making it user-friendly and aesthetically aligned with the rest of the application. A “**Back to Home Page**” button allows users to easily navigate back to the main interface.

Purpose:

Educes users on the platform's importance in saving lives by facilitating blood donation and requests.

Content Focus:

Mission, services provided, and commitment to technology-driven solutions.

Navigation:

Simple button interface returns the user to the homepage.

Design Consistency:

Matches the overall application theme, enhancing user experience.

Encouragement Message:

Motivates users to be part of a life-saving movement by joining the community.

11.CONCLUSION AND FEATURE WORK

11.1 Conclusion

A well-implemented blood management system is crucial for ensuring the efficient and safe distribution of blood products to patients in need. By streamlining the processes involved in blood donation, storage, and distribution, these systems can help to improve patient outcomes and reduce the risk of blood-borne diseases.

Key benefits of a blood management system include:

- Improved efficiency : Automated processes and real-time tracking can reduce manual labor and errors, leading to more efficient operations.
- Enhanced safety: Robust security measures and quality control protocols can help to prevent contamination and ensure the safety of blood products.
- Better patient outcomes : Timely access to blood products can improve patient outcomes and reduce mortality rates.
- Cost savings : By optimizing inventory management and reducing waste, blood management systems can help to reduce costs.
- Improved compliance : These systems can help organizations to comply with relevant regulations and standards.

11.2 Feature Work

Blood management systems have evolved significantly, but there's still room for innovation. Here are some potential future directions:

1. Enhanced Donor Recruitment and Retention:

- Personalized Campaigns: Utilize data analytics to create targeted campaigns that resonate with specific donor demographics.
- Incentive Programs: Explore innovative incentive structures beyond traditional blood drives and gift cards.
- Donor Loyalty Programs: Reward repeat donors with exclusive benefits and recognition.

2. Improved Inventory Management:

- Predictive Analytics: Employ machine learning to forecast blood needs based on historical data, seasonal trends, and local events.
- Real-time Tracking: Implement IoT-enabled technologies to monitor blood unit conditions throughout the supply chain.
- Automated Blood Typing: Explore advancements in rapid, automated blood typing techniques to reduce errors and expedite transfusion processes.

3. Enhanced Blood Safety and Quality:

- Advanced Screening Technologies: Invest in cutting-edge screening methods to detect emerging blood-borne pathogens and diseases.
- Blockchain Technology: Utilize blockchain to ensure the integrity and traceability of blood products from donation to transfusion.

- Personalized Transfusions: Explore the potential of personalized blood products tailored to specific patient needs, such as rare blood types or specific disease conditions.

4. Integration with Electronic Health Records (EHRs):

- Seamless Data Exchange: Facilitate seamless data sharing between blood management systems and EHRs to improve patient care coordination and reduce the risk of adverse events.
- Automated Blood Ordering: Enable automated blood orders to be placed directly from EHRs, streamlining the transfusion process.

5. Mobile Applications:

- Donor Self-Service: Develop mobile apps that allow donors to schedule appointments, track their donation history, and receive reminders.
- Emergency Blood Requests: Create platforms for patients or their representatives to request blood in emergency situations.

6. Artificial Intelligence (AI) and Machine Learning:

- Predictive Modeling : Use AI to predict blood shortages, optimize inventory management, and improve donor recruitment strategies.

12.ANNEXURE

12.1 Glossary of terms and abbreviations

API (Application Programming Interface): A set of rules that allow different software components to communicate with each other.

Database: A structured collection of data that can be accessed, managed, and updated electronically.

ER Diagram (Entity-Relationship Diagram): A graphical representation of data entities and their relationships within a database.

UI/UX (User Interface/User Experience): The design and interaction of a product, focusing on how users interact with it.

Blood tracking: A system that monitors the movement and availability of blood units from donation to transfusion.

Order tracking: A process of monitoring the status of blood orders from the time they are placed to the time they are fulfilled.

Blood donors information: Data collected from individuals who donate blood, including personal details, medical history, and donation records.

Blood donation drive updates: Information about upcoming blood donation drives, including location, date, time, and any special requirements.

12.2 References

Impact Life

<https://www.bloodcenter.org/>

Friend2support

<https://www.friends2support.org>

Eraktkosh

https://eraktkosh.mohfw.gov.in/BLDAHIMS/bloodbank/transactions/bbp_ublicindex.html

Gujarat state council for blood transfusion

<https://gscbt.net/>

Redcross Blood

<https://www.redcrossblood.org/>

Ublood

<https://ublood.com/>

12.3 About tools and technology

1. Development Tools
 - o Visual Studio Code
 - o XAMPP Control Panel
 - o PhpMyAdmin Page

2. Programming Languages
 - o Frontend Development
 - HTML5
 - CSS3
 - JavaScript Frameworks
 - o Backend Development
 - PHP
 - o Database
 - PhpMyAdmin

12.4 About UVPCE College

At Ganpat University-U.V. Patel College of Engineering (GUNI-UVPCE), our mission is to guide and inspire students to achieve technical excellence and meet the ever-evolving demands of industry, the nation, and the global community. We aim to develop a generation of professionals who are technically adept and driven to utilize their knowledge for the betterment of society.

Our core objectives include:

Providing comprehensive guidance for the holistic development of students' personalities.

Imparting technical and need-based education through extensive training programs.

Shaping future professionals to tackle challenges with resilience.

Fostering a sense of belonging and collaboration within the engineering community.

Establishment and Vision

GUNI-UVPCE, established in September 1997, is a constituent college of Ganpat University. Located on a sprawling 25-acre campus at Ganpat Vidyanagar, it was founded to deliver top-tier education and training in Engineering and Technology to cater to the demands of industries locally and globally. The college is named after Shri Ugarchandbhai Varanasibhai Patel, a prominent industrialist, in recognition of his generous contributions.

Infrastructure and Academic Offerings

The college boasts six modern buildings equipped with state-of-the-art classrooms, laboratories, seminar halls, and libraries. Its advanced facilities include high-speed internet through a 1 Gbps fiber link, satellite-based education centers, and robust departmental infrastructure. The institute offers a

wide range of undergraduate, postgraduate, and Ph.D. programs, fostering an environment conducive to research and innovation.

Industry Collaboration and Placement

The college maintains strong ties with leading industries, ensuring practical exposure and excellent placement opportunities for students. These partnerships support student training, hands-on experience, and R&D collaboration. Additionally, the institute promotes entrepreneurial growth through incubation centers and startup initiatives.

Centers of Excellence and Specialized Training

GUNI-UVPCE hosts several Centers of Excellence to equip students with cutting-edge skills, including:

Bosch-Rexroth Centre for Automation Technologies.

Centre for Additive Manufacturing (3D Printing).

EC-Council University Centre for Cyber Security Technologies.

Energy Innovation Centre (EIC).

These centers provide students with access to advanced technologies, ensuring they are well-prepared for emerging industry trends.