Chapter 1

Introduction

This chapter provides a foundational overview of the project, including the problem statement, objectives, and anticipated outcomes.

1.1 Introduction

The availability of blood is a critical aspect of public health, yet many hospitals and blood banks face challenges in managing inventory and connecting with potential donors in a timely manner. The absence of a centralized, real-time system can lead to shortages and delays during emergencies. This project, BloodLink, addresses this problem by developing a web-based platform to efficiently manage the entire blood donation ecosystem. By creating a unified digital hub, we can streamline operations, reduce administrative burden, and ultimately facilitate a more responsive and efficient blood supply chain.

1.2 Motivation

The computational motivation for this project stems from the need for a dynamic and organized system to replace manual or fragmented data management. Manually tracking blood inventory, donor details, and requests is prone to human error and inefficiency. A web-based application provides a single source of truth for all stakeholders. The platform allows for real-time tracking of blood inventory, quick access to donor information, and streamlined blood request processes, ultimately saving time and lives. Furthermore, the development process provides an opportunity to apply principles of software engineering, database design, and user interface development to a real-world problem, which is a valuable learning experience.

1.3 Objectives

The primary objectives of this project are:

1. To design and implement a relational database to store comprehensive data on donors, recipients, blood inventory, and donation history.
2. To develop an administrative dashboard for authorized personnel to perform CRUD (Create, Read, Update, Delete) operations on all data entities.
3. To create user-friendly forms for public users to register as donors or submit blood requests.
4. To ensure the system's data integrity and security, preventing unauthorized access and data corruption.
5. To make the platform responsive and accessible from various devices, including desktops, tablets, and smartphones.

1.4 Feasibility Study

The feasibility of creating a Blood Donation Management System is high. The project utilizes well-established and widely supported open-source technologies such as PHP for the backend and MySQL for the database. A literature review of similar projects and existing blood bank management software confirms that the proposed features are technically achievable and meet a real-world need. The project is

also economically feasible, as it uses free and open-source software, requiring minimal infrastructure costs. The project is operationally feasible as it provides a system that is easy to use and integrates seamlessly into the workflow of a blood bank or hospital.

1.5 Gap Analysis

Existing solutions often lack a centralized, real-time interface or are proprietary systems limited to specific organizations. Many small-scale blood banks or hospital departments still rely on manual record-keeping or simple spreadsheets, which are inefficient and error-prone. This project fills the gap by providing a simple, localized, and open-source platform that can be adapted by these organizations to better connect with their community of donors and recipients. The system is designed to be easily deployable and customizable, offering a more robust alternative to manual processes.

1.6 Project Outcome

The project outcome is a fully functional web application that provides a comprehensive administrative dashboard for managing blood donation data. The system allows for efficient data entry, retrieval, and management, leading to improved operational efficiency and a better-managed blood supply chain. The project also serves as a proof of concept, demonstrating the practical application of software engineering principles to solve a pressing societal problem.

Chapter 2

Proposed Methodology/Architecture

This chapter details the system's design, from the overall architecture to the underlying data structure and user interface.

2.1 Requirement Analysis & Design Specification

This section outlines the functional and non-functional requirements that guided the design of the system.

2.1.1 Overview

The system requires a robust backend to handle database interactions and a responsive frontend for user input and data display. The primary functional requirements include managing donors, recipients, blood requests, donation history, and blood inventory. Non-functional requirements include security, performance, and usability. The system must be able to handle a moderate number of users and transactions without significant latency.

2.1.2 System Design

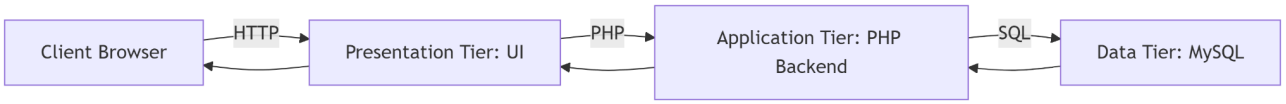
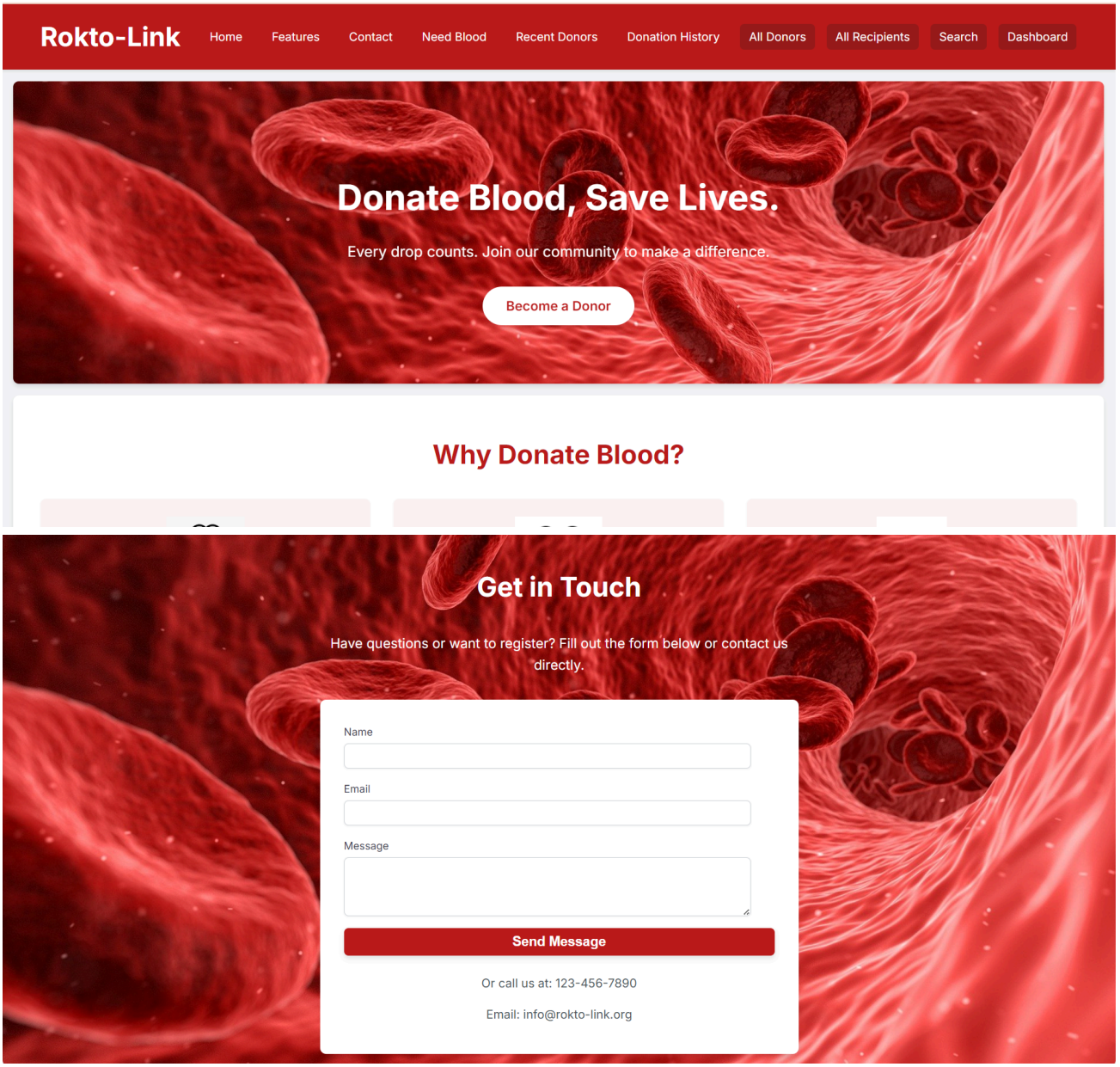


Figure 2.1: System Design diagram

The system employs a standard client-server architecture, using a three-tier model. The presentation tier is the user interface, the application tier is the PHP backend, and the data tier is the MySQL database. The user's browser acts as the client, sending HTTP requests to the server. The server-side logic is handled by PHP scripts, which process the requests, interact with the MySQL database using standard SQL queries, and generate dynamic HTML content. This modular approach ensures that each component can be developed and maintained independently.

2.1.3 UI Design

The user interface is designed to be clean, intuitive, and responsive. It uses a modern CSS framework for consistent styling and adapts to different screen sizes, providing a good user experience on both desktop and mobile devices. The administrative dashboard is a single-page application that uses dynamic form sections to manage different data entities, minimizing page reloads and improving user flow. The design prioritizes readability and ease of navigation to ensure administrators can perform their tasks quickly and accurately.


Rokto-Link
Home
Features
Contact
Need Blood
Recent Donors
Donation History
All Donors
All Recipients
Search
Dashboard
Donate Blood, Save Lives.
Every drop counts. Join our community to make a difference.
Become a Donor
Why Donate Blood?

Get in Touch
Have questions or want to register? Fill out the form below or contact us directly.
Name
Email
Message
Send Message
Or call us at: 123-456-7890
Email: info@rokto-link.org

Recent Donors

Donor Name	Blood Group	Last Donation
Alice Johnson	B+	2025-07-22
Jane Smith	A-	2025-07-10
John	O+	2024-06-15

Recent Donation History

Donation Date	Donor Name	Blood Group	Units Donated
2025-07-22	Alice Johnson	B+	10
2025-07-10	Jane Smith	A-	1
2025-07-03	Alice Johnson	AB+	1

Top Needed Blood Groups (Urgent Requests)

A+

Units Needed: 5

A-

Units Needed: 5

O-

Units Needed: 2

Recent Donors

Donor Name	Blood Group	Last Donation
Alice Johnson	B+	2025-07-22

Why Donate Blood?

Life-Saving Impact

Your single donation can save up to three lives. It's a simple act with profound impact.

For Emergencies

Blood is constantly needed for accident victims, surgeries, and chronic illnesses.

Community Health

Maintaining a healthy blood supply is crucial for the well-being of our entire community.

Our Features

Search Blood Data

Home Dashboard

Search Options

Search In: Donors
Keyword Search: Enter name, email, hospital
Blood Group: Any
City (Donors): Enter city

Search

Search Results (Donors)

Id	Name	Email	Phone	Blood Group	Last Donation Date	Address	City	State	Zip Code	Is Available	Created At
3	Alice Johnson	alice.j@example.com	555-123-4567	B+	2025-08-08		Chittagong			1	2025-07-31 13:45:46
2	Jane Smith	jane.smith@example.com	987-654-3210	A-	2025-08-09		Dhaka			1	2025-07-31 13:45:46
1	John	john.doe@example.com	123-456-7890	O+	2025-07-31		Dhaka			1	2025-07-31 13:45:46

Admin Dashboard

Quick Actions

Add New Donor | Add New Recipient | Add New Donation | Add New Blood Request | Hide Add Forms

Add New Blood Request

Recipient: Select Recipient
Blood Group: Select Blood Group
Units Needed:
Status: Pending

Add Blood Request

Quick Actions

Add New Donor | Add New Recipient | Add New Donation | Add New Blood Request | Hide Add Forms

Add New Recipient

Name:
Email:
Phone:
Blood Group: Select Blood Group
Reason for Blood:
Hospital Name:
City:
State:

Add Recipient

Quick Actions

Add New Donor | Add New Recipient | Add New Donation | Add New Blood Request | Hide Add Forms

Add New Donor

Name:

Email:

Phone:

Blood Group: Select Blood Group

Last Donation Date: mm/dd/yyyy

Address:

City:

State:

Zip Code:

Is Available for Donation?

Add Donor

Donation History

Id	Donor Id	Donation Date	Blood Group	Units Donated	Notes	Created At	Actions
12	Jane Smith	2025-08-09	A+	4		2025-08-01 23:25:15	Edit Delete
11	Alice Johnson	2025-08-08	AB+	7		2025-08-01 23:24:55	Edit Delete
10	Jane Smith	2025-08-07	O-	2		2025-08-01 23:24:30	Edit Delete
13	John	2025-07-31	O-	2		2025-08-02 23:00:44	Edit Delete
9	Alice Johnson	2025-07-22	B+	10		2025-07-31 15:33:20	Edit Delete
7	Jane Smith	2025-07-10	A-	1	gnbghngn	2025-07-31 14:01:54	Edit Delete

2	Mr. Sharma	sharma.r@example.com	444-555-6666	AB+	Transfusion for chronic illness	General Hospital	Chittagong	2025-07-31 13:45:46	Edit D

Blood Requests

Id	Recipient Id	Blood Group	Units Needed	Request Date	Status	Actions
6	Baby Alex	A-	5	2025-07-31 15:37:55	urgent	Edit Delete
5	Baby Alex	A+	5	2025-07-31 14:47:28	urgent	Edit Delete
1	Baby Alex	O-	2	2025-07-31 13:45:46	urgent	Edit Delete

Donation History

Donors

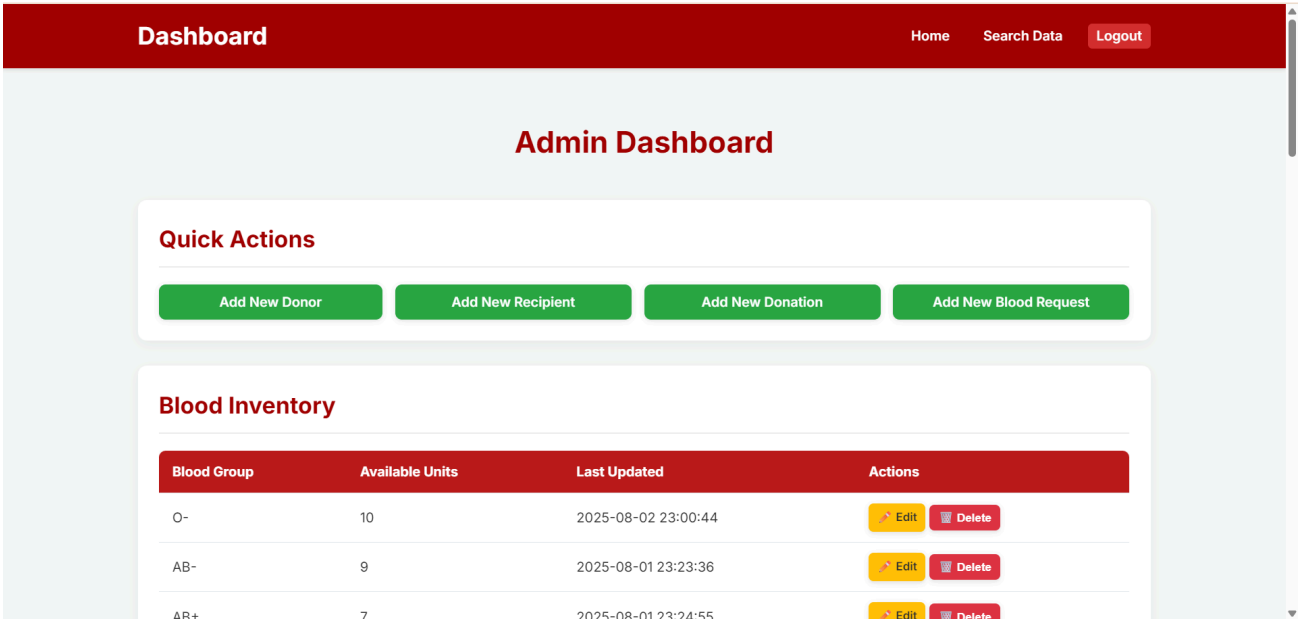
Id	Name	Email	Phone	Blood Group	Last Donation Date	Address	City	State	Zip Code	Is Available	Crea At
1	John	john.doe@example.com	123-456-7890	O+	2025-07-31		Dhaka			Yes	2025 07-3 13:45
2	Jane Smith	jane.smith@example.com	987-654-3210	A-	2025-08-09		Dhaka			Yes	2025 07-3 13:45
3	Alice Johnson	alice.j@example.com	555-123-4567	B+	2025-08-08		Chittagong			Yes	2025 07-3 13:45

Recipients

Id	Name	Email	Phone	Blood Group	Reason	Hospital Name	City	State	Created At	Actions

Blood Inventory

Blood Group	Available Units	Last Updated	Actions
O-	10	2025-08-02 23:00:44	Edit Delete
AB-	9	2025-08-01 23:23:36	Edit Delete
AB+	7	2025-08-01 23:24:55	Edit Delete
B-	3	2025-08-01 23:23:11	Edit Delete
A-	4	2025-08-01 23:23:20	Edit Delete
A+	4	2025-08-01 23:25:15	Edit Delete



2.2 Overall Project Plan

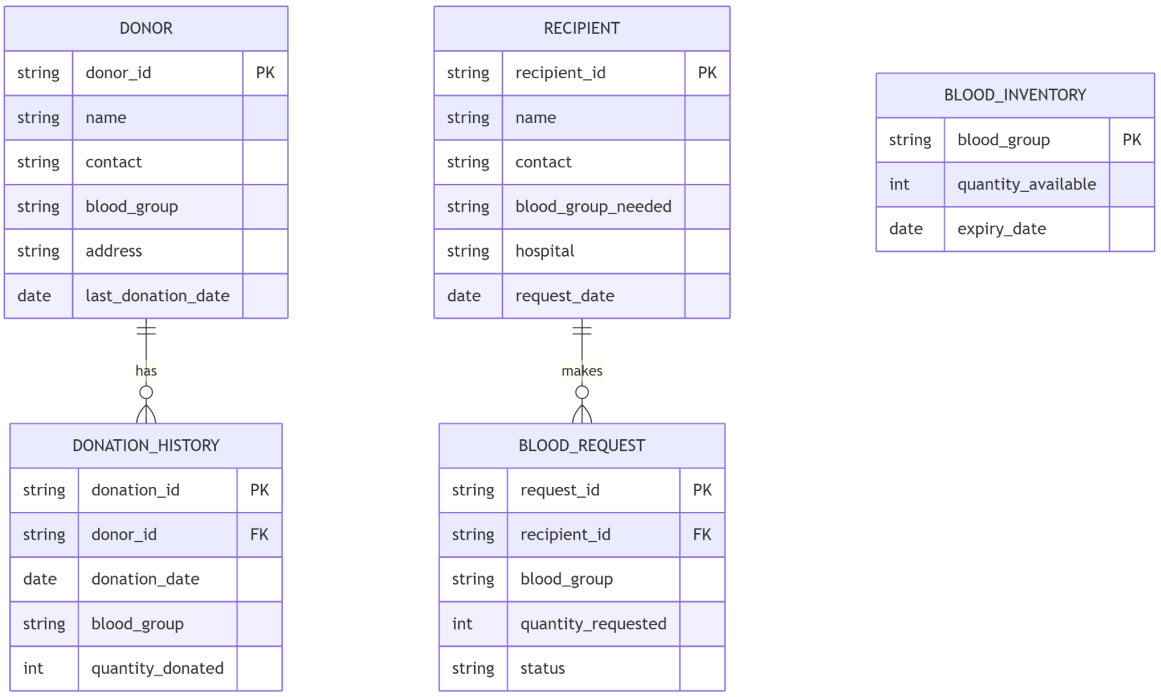
The project was developed following an agile methodology, with individual team members taking ownership of specific modules. This allowed for parallel development and frequent integration of components to ensure a cohesive final product. The project was broken down into a series of sprints, with each sprint focusing on a specific feature set (e.g., donor management, blood request forms, etc.). This iterative process allowed for continuous feedback and refinement of the system.

2.3 Entity-Relationship (ER) Diagram

The ER diagram of the database schema shows the relationships between the main entities. It consists of the following entities:

Donors: Stores personal information about blood donors.
Recipients: Stores information about individuals receiving blood.
Blood_Inventory: Tracks the available units of each blood group.
Donation_History: Records each blood donation event.
Blood_Requests: Records requests for blood from recipients.

The relationships are as follows:
A Donor can have many Donation_History records (one-to-many).
A Recipient can have many Blood_Requests (one-to-many).
A Donation_History record is linked to a specific Donor (many-to-one).
A Blood_Request is linked to a specific Recipient (many-to-one).
The Blood_Inventory table is a standalone table tracking inventory by blood_group.



2.4 Database Schema

SQL

```
CREATE TABLE `donors` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `email` varchar(255) DEFAULT NULL,
  `phone` varchar(20) DEFAULT NULL,
  `blood_group` enum('A+','A-','B+','B-','AB+','AB-','O+','O-') NOT NULL,
  `last_donation_date` date DEFAULT NULL,
  `address` text DEFAULT NULL,
  `city` varchar(100) DEFAULT NULL,
  `state` varchar(100) DEFAULT NULL,
  `zip_code` varchar(20) DEFAULT NULL,
  `is_available` tinyint(1) NOT NULL DEFAULT 1,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  PRIMARY KEY (`id`),
  UNIQUE KEY `email` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `recipients` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `email` varchar(255) DEFAULT NULL,
  `phone` varchar(20) DEFAULT NULL,
  `blood_group` enum('A+','A-','B+','B-','AB+','AB-','O+','O-') NOT NULL,
  `reason` text DEFAULT NULL,
  `hospital_name` varchar(255) DEFAULT NULL,
  `city` varchar(100) DEFAULT NULL,
  `state` varchar(100) DEFAULT NULL,
```

```
    `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
    PRIMARY KEY (`id`),
    UNIQUE KEY `email` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `blood_inventory` (
    `blood_group` enum('A+','A-','B+','B-','AB+','AB-','O+','O-') NOT NULL,
    `available_units` int(11) NOT NULL DEFAULT 0,
    `last_updated` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
    PRIMARY KEY (`blood_group`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `donation_history` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `donor_id` int(11) NOT NULL,
    `donation_date` date NOT NULL,
    `blood_group` enum('A+','A-','B+','B-','AB+','AB-','O+','O-') NOT NULL,
    `units_donated` int(11) NOT NULL,
    `notes` text DEFAULT NULL,
    `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
    PRIMARY KEY (`id`),
    FOREIGN KEY (`donor_id`) REFERENCES `donors` (`id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `blood_requests` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `recipient_id` int(11) NOT NULL,
    `blood_group` enum('A+','A-','B+','B-','AB+','AB-','O+','O-') NOT NULL,
    `units_needed` int(11) NOT NULL,
    `request_date` date NOT NULL,
    `status` enum('pending','fulfilled','urgent','cancelled') NOT NULL DEFAULT 'pending',
    `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
    PRIMARY KEY (`id`),
    FOREIGN KEY (`recipient_id`) REFERENCES `recipients` (`id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Chapter 3

Implementation and Results

This chapter covers the project's implementation details, from the programming language to the final outcomes.

3.1 Implementation

The project was implemented using PHP for server-side scripting and MySQL for the database. The frontend was built with HTML and CSS, with a focus on responsiveness and an intuitive user experience. The core of the system is the dashboard.php file, which serves as the administrative interface. It uses a single-page approach with conditional rendering of forms and tables, allowing for inline editing and a smooth user experience. The system's functionality is driven by a series of PHP functions that handle all database interactions, including insertion, updating, and deletion of records based on form submissions.

3.2 Performance Analysis

The system is designed to be lightweight and efficient. By using a relational database with appropriate indexing on primary keys, data retrieval operations are fast. The front-end code is streamlined to minimize client-side processing, and the use of a single-page design for the dashboard reduces the number of server requests. The PHP backend is optimized to handle concurrent requests efficiently, ensuring that the system remains responsive even with multiple administrators using it at the same time. While scalability for a nationwide system would require further optimization, the current architecture is well-suited for a localized, smaller-scale deployment.

3.3 Results and Discussion

The project was implemented using PHP for server-side scripting and MySQL for the database. The frontend was built with HTML and CSS, with a focus on responsiveness and an intuitive user experience. The core of the system is the dashboard.php file, which serves as the administrative interface. It uses a single-page approach with conditional rendering of forms and tables, allowing for inline editing and a smooth user experience. The system's functionality is driven by a series of PHP functions that handle all database interactions, including insertion, updating, and deletion of records based on form submissions.

Chapter 4

Engineering Standards and Mapping

This chapter addresses the broader context of the project, including its impact, team collaboration, and alignment with program outcomes.

4.1 Impact on Society, Environment and Sustainability

This section details the broader impact of the project beyond its technical implementation.

4.1.1 Impact on Life

The project's primary impact is on human life. By creating a more efficient and responsive system for managing blood resources, the platform can help ensure that blood is available when and where it is needed most. This can reduce delays in emergency medical situations, potentially saving lives and improving patient outcomes.

4.1.2 Impact on Society & Environment

From a societal perspective, the project promotes civic engagement by making it easier for individuals to register as blood donors. It provides a centralized resource that can be a catalyst for community-based blood drives. The environmental impact is minimal, as it is a purely digital solution, reducing the need for paper-based record-keeping. The system's sustainability lies in its ability to be maintained and developed with open-source technologies, making it a cost-effective and long-lasting solution.

4.1.3 Ethical Aspects

The system handles sensitive personal and medical data. Therefore, the ethical aspects of data privacy and security are paramount. The project design adheres to ethical considerations by not collecting any data that is not necessary for its core function and by ensuring that the database is designed with basic security measures. Future work will focus on implementing more robust security and authentication.

4.1.4 Sustainability Plan

The project is built on open-source technologies, making it inherently sustainable. It does not rely on proprietary software or expensive licenses. The code is well-documented, allowing for future developers to easily understand and build upon the existing system. This ensures that the project can evolve and be maintained over the long term, adapting to new needs and technologies.

4.2 Project Management and Team Work

The project was a collaborative effort by three team members, each with a distinct area of responsibility. We adopted a decentralized approach to project management, where each member was given autonomy over their designated module.

- ☐ **1st team mate:** Took the lead on implementing all functionalities related to the blood inventory table. This included developing the UI for displaying inventory, implementing the CRUD operations for managing units, and ensuring data integrity with other related tables.
- ☐ **2nd team mate:** Was responsible for the management of donors and recipients. Her contribution involved designing and implementing the registration forms and the dashboard's tables and CRUD functionality for these two entities.
- ☐ **3rd team mate :** Focused on the blood request and donation history modules. This included

creating the forms for new blood requests, developing the display tables for donation records, and implementing the associated CRUD logic.

A cost analysis for this project suggests a minimal budget, as all software used is free. The revenue model for a future, larger-scale application could involve a subscription fee for hospitals or organizations to use the platform. An alternate budget would involve purchasing a pre-built commercial solution, which would be significantly more expensive and less customizable. The rationale for our approach is the cost-effectiveness and flexibility of open-source development.

4.3 Complex Engineering Problem

This section provides a mapping of the project's problem-solving and activity categories.

4.3.1 Mapping of Program Outcome

In this section, we provide a mapping of the problem and provided solution with targeted Program Outcomes (PO's).

Table 4.1: Justification of Program Outcomes

PO's	**Justification**
PO1	The project demonstrates an application of knowledge of mathematics, science, and engineering principles to solve the complex problem of blood resource management. This includes applying principles of database design, software architecture, and user interface development.
PO2	The project involved identifying, formulating, and analyzing the engineering problem of blood donation logistics and developing a functional solution. This included analyzing user requirements and translating them into a working system.
PO3	The project's design phase required the development of a system that meets the needs of various stakeholders (donors, recipients, administrators), demonstrating the ability to design system components and processes.

4.3.2 Complex Problem Solving

In this section, we provide a mapping with problem solving categories. For each mapping add subsections to put rationale (Use Table 4.2). For P1, you need to put another mapping with Knowledge profile and rational thereof.

Table 4.2: Mapping with complex problem solving.

EP1 Dept of Knowledge	**EP2** Range of Conflicting Requirements	**EP3** Depth of Analysis	**EP4** Familiarity of Issues	**EP5** Extent of Applicable Codes	**EP6** Extent Of Stakeholder Involvement	**EP7** Inter-dependence
✅	✅	✅	✅	✅		

4.3.3 Engineering Activities

In this section, provide a mapping with engineering activities. For each mapping add subsections to put rationale (Use Table 4.3).

Table 4.3: Mapping with complex engineering activities.

EA1 Range of resources	**EA2** Level of Interaction	**EA3** Innovation	**EA4** Consequences for society and environment	**EA5** Familiarity
The project uses a wide range of open-source resources, including PHP, MySQL, and various web development tools.	The system involves a high level of interaction between the user, the application, and the database, requiring careful design of user interfaces and data flow.	The innovation lies in providing an accessible, open-source solution that can be easily adopted by a wide range of small organizations, filling a market gap.	The project has a high impact on society by improving the efficiency of blood donation. The environmental impact is minimal.	The team was highly familiar with the technologies and tools used in the project, which streamlined the development process.

Chapter 5

Conclusion

This chapter provides a final summary of the project, its limitations, and potential future improvements.

5.1 Summary

The BloodLink project successfully created a functional web-based system for managing blood donation data. The application provides a centralized and efficient platform for administrators to handle records of donors, recipients, donations, and blood requests. It effectively addresses the stated objectives and provides a solid foundation for a more comprehensive blood management solution. The project demonstrates the team's ability to apply software engineering principles to solve a real-world problem and has a positive societal impact.

5.2 Limitation

The current system has several limitations. It lacks a robust user authentication system for public users, meaning that donor and recipient information is submitted without a formal login. The public-facing forms also do not have integrated real-time validation for duplicate entries or advanced input sanitization. Additionally, the system does not include features for automated notifications or a sophisticated algorithm for matching donors to urgent requests.

5.3 Future Work

Future enhancements for this project could include:

1. Implementing a secure user authentication system for donors and recipients, allowing them to manage their own profiles and donation history.
2. Developing a real-time notification system using push notifications or SMS to alert available donors of urgent blood requests.
3. Adding a more advanced search and filtering capability to the dashboard, allowing administrators to find donors based on location, last donation date, and other criteria.
4. Integrating a graphical representation of the blood inventory to provide a more visual and intuitive overview of available resources.
5. Developing a mobile application version of the platform to improve accessibility for donors and recipients.

References

[1] Jon Kleinberg and Eva Tardos. *Algorithm design*. Pearson Education India, 2006.