BLOOD DONOR NOTIFICATION SYSTEM

A DESIGN PROJECT REPORT

Submitted by

ABINESH R (811722104005)

ANBAZHAGAN P (811722104011)

HARIBHARATHI R (811722104047)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

NOV/DEC,2024

BLOOD DONOR NOTIFICATION SYSTEM

A DESIGN PROJECT REPORT

Submitted by

ABINESH R (811722104005)

ANBAZHAGAN P (811722104011)

HARIBHARATHI R (811722104047)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

NOV/DEC,2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(AUTONOMOUS)
SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report titled **"BLOOD DONOR NOTIFICATION SYSTEM"** is the Bonafide work of the students ABINESH.R (811722104005), ANBAZHAGAN.P (811722104011), HARIBHARATHI.R (811722104047) who carried out the project under my supervision. Certified further, that to the best of my knowledge the workreported herein does not form part of any other project report or dissertation on thebasis of which a degree or award was conferred on an earlier occasion on this orany other candidate.

SIGNATURE

Dr.A.Delphin Carolina Rani, M.E, Ph.D**.,**

HEAD OF THE DEPARTMENT
PROFESSOR

Department of CSE

K. Ramakrishnan College of Technology

(Autonomous)

Samayapuram – 621 112

SIGNATURE

Mrs.A.THENMOZHI,M.E.,

ASSISTANT PROFESSOR

Department of CSE

K. Ramakrishnan College of Technology

(Autonomous)

Samayapuram – 621 112

Submitted for the viva-voce examination held on …….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"BLOOD DONOR NOTIFICATION SYSTEM"** is the result of original workdone by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of **BACHELOR OF ENGINEERING**.

Signature

ABINESH R

ANBAZHAGAN P

HARIBHARATHI R

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and in-debt to our institution "**K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)**", for providing us with the opportunity to do this project.

We are glad to credit honorable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project andoffering adequate duration in completing our project.

We would like to thank **Dr. N. VASUDEVAN, M.Tech, Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

We heartily thanks to **Dr. A.DELPHIN CAROLINA RANI, M.E, Ph.D.,** Head of the Department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express my deep and sincere gratitude to my beloved project guide **Mrs. A.THENMOZHI, M.E..,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for **her** incalculable suggestions, creativity, assistance andpatience which motivated me to carry out this project.

I render my sincere thanks to Course Coordinator and other staff membersfor providing valuable information during the course.

I wish to express my special thanks to the officials and Lab Technicians ofour departments who rendered their help during the period of the work progress.

ABSTRACT

The BLOOD DONOR NOTIFICATION SYSTEM is a comprehensive platform developed to enhance the efficiency of blood donation processes and optimize blood inventory management. It serves as a centralized solution for managing donor information, tracking blood stocks, scheduling donations, and ensuring the availability of blood to meet medical demands in a timely manner. Key features of the system include user registration, real-time inventory management, donor eligibility screening, donation history tracking, and appointment scheduling. Additionally, the system incorporates donor geolocation, allowing blood banks to identify and notify nearby donors during urgent situations, enabling faster response times for critical needs.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION :

API	-	Application Programming Interface
DB	-	Database
SMS	-	Short Message Service
UI	-	User Interface
EX	-	User Experience
OTP	-	One Time Password
IOT	-	Internet of Things
DOB	-	Date Of Birth
BG	-	Blood Group
BCC	-	Blind carbon Copy
DON	-	Donation

CHAPTER 1

INTRODUCTION

The demand for a reliable, efficient, and technology-driven BLOOD DONOR NOTIFICATION SYSTEM has never been greater. Blood transfusions are critical in a wide range of medical procedures, yet blood banks and healthcare institutions frequently encounter challenges related to blood shortages, donor engagement, and inventory management. To address these needs, advanced technologies, including Artificial Intelligence (AI), are increasingly being integrated into healthcare solutions to improve responsiveness, accuracy, and operational efficiency.

Machine learning algorithms are applied to analyze donation patterns and donor behaviors, identifying high-risk periods and proactively reaching out to eligible donors. Another core feature of the system is the integration of geolocation with AI-based notifications. Using donor location data, the system identifies and alerts nearby eligible donors in case of emergencies, enhancing response times and increasing donation rates in critical situations. AI algorithms personalize donor notifications based on each individual's history, eligibility status, and location, creating a targeted approach to engage donors effectively. Additionally, AI-driven chatbots provide 24/7 support, answering common donor questions, guiding them through the registration process, and even helping them schedule appointments.

Overall, the AI-enhanced BLOOD DONOR NOTIFICATION SYSTEM addresses the pressing need for efficient, automated, and intelligent solutions in healthcare. By leveraging the latest advancements in AI technology. Through innovation and AI integration, this system aims to transform blood bank operations, optimize blood availability, and ultimately save lives.

1.1 PROJECT OVERVIEW

The BLOOD DONOR NOTIFICATION SYSTEM is a comprehensive platform designed to streamline and optimize the management of blood donations, inventory, and donor engagement. This project leverages modern technologies, including Artificial Intelligence (AI) and geolocation, to address critical challenges faced by blood banks and healthcare institutions. The system aims to maintain adequate blood supplies, engage donors effectively, and ensure rapid response during emergencies.

Key features of the system include donor registration, eligibility screening, and real-time blood inventory management. Predictive analytics are used to forecast demand, preventing shortages and minimizing wastage.

The system also generates detailed reports on blood trends, donor behavior, and inventory, aiding decision-making and improving operations. Data security is a top priority, with the platform implementing robust measures to ensure compliance with privacy standards. Accessible via both mobile and web platforms, this project aims to provide a user-friendly and reliable solution for managing blood donation activities, ultimately supporting life-saving healthcare services.

1.2 PROBLEM STATEMENT

Blood banks and healthcare institutions face persistent challenges in managing blood supplies, engaging donors, and ensuring timely response to urgent medical needs. These challenges include frequent blood shortages, inefficient inventory management, difficulties in retaining regular donors, and delayed emergency responses. Traditional manual systems often result in human errors, mismanagement of blood stocks, and limited communication with donors, leading to both wasted blood and potential shortfalls during critical times.

Additionally, there is a lack of personalized communication, making it hard to keep donors engaged and informed about upcoming opportunities and urgent needs. The absence of real-time data and geolocation-based notifications hinders the ability to mobilize eligible donors quickly during emergencies. Moreover, maintaining the security and privacy of sensitive donor information is a major concern, especially with increasing data breaches and privacy regulations.

1.2.1 GOALS

The main goal of the BLOOD DONOR NOTIFICATION SYSTEM is to improve the efficiency and effectiveness of blood donation processes, ensuring a reliable and continuous supply of blood for medical needs. The current system faces multiple challenges, such as inefficient inventory tracking, inadequate communication with donors, and slow response times during emergencies, which can lead to critical shortages or wasted supplies.

1.3 OBJECTIVE OF THE PROJECT

- Optimize Blood Inventory
- Increase Donor Engagement
- Enhance Emergency Response
- Ensure Data Security
- Provide Detailed Reports and Analytics
- Streamline Appointment Scheduling
- Improve Communication Between Stakeholders
- Support Scalability and Flexibility
- Automate Donor Eligibility Screening
- Minimize Blood Wastage
- Improve Donor Retention Rates
- Facilitate Mobile and Web Access

1.4 SCOPE OF THE PROJECT

The scope of the BLOOD DONOR NOTIFICATION SYSTEM project focuses on automating and optimizing the processes involved in blood donation, inventory management, and donor engagement. It includes the development of a centralized platform that serves as a hub for donors, blood banks, and healthcare providers. Key features include donor registration and management, eligibility screening, real-time inventory tracking, donation scheduling, geolocation-based notifications, and automated reminders. The system will provide reporting and analytics tools for tracking trends, donor engagement, and stock levels. Additionally, it will ensure data security through encryption and access control. The project will be accessible via both mobile and web interfaces.

Key features within the scope include:

- **Donor Registration and Management**: A user-friendly interface for donors to register, manage their profiles, and track their donation history.
- **Eligibility Screening**: An automated system to assess donor eligibility based on predefined health criteria.
- **Inventory Tracking**: Real-time management of blood stocks by type and expiration date, along with predictive analytics for demand forecasting.
- **Appointment Scheduling**
  Allows donors to book, modify, or cancel donation appointments online or via a mobile app. Includes reminders and location-based options for convenience.
- **Notification and Alerts**
  Sends reminders for upcoming appointments, eligibility updates, and urgent requests for specific blood types during shortages or emergencies.
- **Integration with Health Records**
  Links with electronic health records (EHR) to retrieve donor health information securely and ensure compliance with eligibility criteria.

CHAPTER 2
LITERATURE REVIEW

2.1 TITLE: DEVELOPMENT OF BLOOD DONOR NOTIFICATION SYSTEM

AUTHORS: DR. MICHAEL JOHNSON

YEAR: 2023

The study conducted by Dr. Michael Johnson in 2023 explores the **Development of BLOOD DONOR NOTIFICATION SYSTEM** with a focus on the integration of modern technology to enhance the efficiency, accuracy, and responsiveness of blood banks. This research combines elements of Computer Science and Electrical and Electronics Engineering (EEE) to address common challenges faced in traditional blood donation systems. The research underscores the role of IoT (Internet of Things) technology in developing a more automated and reliable system for blood management.

Johnson and Lee examine the use of IoT sensors for real-time monitoring of blood storage conditions, such as temperature, humidity, and blood shelf life, ensuring that the stored blood remains in safe conditions. These sensors are designed to automatically alert staff if any parameter goes out of the specified range, reducing the risk of blood spoilage due to storage mishandling. Their study also delves into the role of geolocation technology in enhancing the efficiency of donor management. The authors demonstrate how GPS and geolocation modules can identify and notify eligible donors within a specific radius during emergency situations. This approach not only increases the responsiveness of the blood bank system but also ensures that critical blood supplies are obtained faster during urgent medical cases.

2.2 TITLE: DESIGN AND IMPLEMENTATION OF BLOOD DONOR NOTIFICATION SYSTEM

AUTHOR: DR. DAVID MILLER

YEAR:2021

The research conducted by Dr. David Miller in 2021 focuses on the Design and Implementation of Blood Donor Notification System with an emphasis on integrating modern technologies from the field of Electrical and Electronics Engineering (EEE) to improve the efficiency, security, and reliability of blood donation processes.

Their study explores the challenges of traditional blood donor management systems, such as manual errors, inefficient communication with donors, and lack of real-time data access. To address these issues, Miller and Roberts have proposed a system that leverages IoT (Internet of Things) and AI-driven solutions. The research showcases how IoT devices can be implemented to automate blood inventory tracking, monitor storage conditions, and maintain accurate records of blood donations. These IoT sensors provide real-time data on blood storage parameters like temperature and expiration dates, automatically alerting staff if any issues are detected.

Another significant aspect covered in their research is the integration of reliable communication systems to connect blood banks, healthcare centers, and donors. They discuss the use of secure communication protocols to transmit data, ensuring that sensitive donor information is protected and privacy is maintained. Additionally, the system is designed to send automated notifications to donors, including reminders for upcoming donations, eligibility alerts, and urgent requests for specific blood types, enhancing donor engagement. The proposed system revolutionizes the blood donation process by ensuring that blood stocks are managed efficiently and communication with donors is seamless. The use of IoT and AI not only automates routine tasks but also provides predictive insights, improving resource allocation and emergency preparedness.

2.3 TITLE: ENHANCING BLOOD DONOR NOTIFICATION THROUGH TECHNOLOGY INTEGRATION

AUTHOR: DR. RACHEL CARTER

YEAR:2020

In 2020, Dr. Rachel Carter conducted a comprehensive study on Enhancing Blood Donor Management Through Technology Integration, focusing on the application of advanced technologies from Electrical and Electronics Engineering (EEE) to modernize and improve blood donor management systems. Her research highlights the transformative potential of technology in increasing the efficiency, accuracy, and responsiveness of blood donation processes.

Dr. Carter's study emphasizes the use of **Internet of Things (IoT)** devices to enhance real-time monitoring of blood storage conditions. By integrating IoT sensors, blood banks can continuously track temperature, humidity, and expiration dates, ensuring that stored blood remains viable. These sensors automatically alert staff if storage conditions deviate from the required parameters, reducing the risk of blood spoilage due to improper handling.

The research also focuses on the application of **Artificial Intelligence (AI)** in donor management. Dr. Carter demonstrates how AI algorithms can be utilized to analyze donation trends, predict blood demand, and manage inventory levels. The AI-driven system can suggest the best times to organize donation drives based on historical data, optimize donor scheduling, and send targeted notifications to potential donors, improving overall engagement and reducing shortages. Overall, Dr. Rachel Carter's 2020 study presents a comprehensive analysis of how technology can be integrated into blood donor management systems. This proposed system represents a significant step forward in modernizing blood donor management, leveraging cutting-edge technology to address existing limitations while paving the way for innovative future applications.

2.4 TITLE: A COMPARATIVE STUDY OF BLOOD DONOR NOTIFICATION SYSTEMS

AUTHOR: DR. JONATHAN EVANS

YEAR: 2021

In 2021, Dr. Jonathan Evans conducted a study titled A Comparative Study of Blood Donor Management Systems, focusing on the evaluation of different technological approaches to blood donor management, particularly through the lens of Electrical and Electronics Engineering (EEE). His research aimed to identify the strengths and weaknesses of traditional versus modern, technology-driven systems in managing blood donations effectively.

Dr. Evans's study compares **manual blood donor management systems** with those that incorporate advanced technologies like **IoT (Internet of Things)**, **AI (Artificial Intelligence)**, and **cloud-based platforms**. The research highlights that traditional systems often suffer from inefficiencies, including slow data processing, manual errors, and lack of real-time updates, which can lead to outdated inventory information and missed donor engagement opportunities.

In contrast, modern systems utilize **AI algorithms** for predictive analytics, allowing for better inventory management and demand forecasting. Dr. Evans emphasizes how AI can help anticipate blood shortages, optimize donor recruitment efforts, and improve the timing of blood drives based on historical donation patterns. Dr. Evans concludes that while traditional systems can handle basic donor management, technology-integrated solutions provide a significant improvement in efficiency, accuracy, and scalability. Dr. Evans's comparative analysis underscores the transformative potential of modern systems in revolutionizing blood donor management. By presenting a clear contrast between traditional and advanced methods, his study serves as a compelling case for adopting technology-driven solutions in blood banks and healthcare systems worldwide. The study highlights the limitations of manual systems and demonstrates how modern approaches significantly enhance the efficiency and reliability of blood donation processes.

CHAPTER 3

EXISTING SYSTEM

The current systems used for managing blood banks and donor information are often outdated and heavily reliant on manual processes. In many traditional setups, blood banks maintain donor records, blood inventory, and donation schedules using spreadsheets or paper-based logs. This manual data entry approach can be prone to human error, causing inconsistencies and inaccurate records. In addition, communication with donors is typically done through phone calls or basic SMS notifications, leading to delays in contacting potential donors during emergencies. Inventory management is often inefficient, as the tracking of blood types, expiry dates, and availability is done manually, increasing the risk of outdated or expired blood stocks.

The Disadvantage of Existing System are:

- Manual Data Entry Errors
- Time-Consuming Communication
- Lack of Real-Time Updates
- Limited Integration
- Inefficient Inventory Management
- No Predictive Analytics
- Security Vulnerabilities
- Low Donor Engagement.

CHAPTER 4

PROPOSED SYSTEM

The proposed BLOOD DONOR NOTIFICATION SYSTEM aims to create an efficient, automated, and user-friendly platform for managing blood donation processes. This system will leverage web technologies like HTML, CSS, PHP, and SQL to build a centralized database that streamlines donor registration, blood inventory management, and blood requests. The system will provide a secure interface for donors to register, track their donation history, and receive notifications about upcoming donation events. The blood bank staff will have access to tools for managing donor records, tracking blood storage, and generating reports. This system will also facilitate communication between blood banks and donors through automated email notifications, reducing manual work and enhancing accuracy. Real-time inventory tracking will allow blood banks to maintain an optimal stock level, minimizing wastage and ensuring that emergency requirements are promptly met. Additionally, the integration of data visualization tools will provide insights into donor trends and blood demand patterns, supporting better decision-making and efficient resource allocation.

The Advantages of Proposed System are:

- Automated donor and blood inventory management
- Real-time blood availability tracking
- Easy donor registration and information updates
- Efficient communication with donors via notifications
- Enhanced data accuracy and reduced manual errors
- Centralized database for streamlined operations
- Quick access to donor history and blood stock report.

CHAPTER 5
SYSTEM ARCHITECTURE

The system architecture of the BLOOD DONOR NOTIFICATION SYSTEM is designed to provide a structured and efficient workflow for managing blood donations, inventory, and donor interactions. The architecture is built around a centralized database that stores all critical information, including donor details, blood type inventory, donation history, and blood requests. The front-end interface, developed using HTML, CSS, and JavaScript, allows users—both donors and blood bank staff—to interact with the system through a web browser. The back-end server, powered by PHP, handles all business logic, data processing, and communication with the database using SQL.

The system includes modules for donor registration, blood inventory tracking, notifications, and reporting, with data visualization tools integrated for better insights. A secure authentication mechanism ensures that only authorized users can access sensitive information. The architecture also supports real-time updates and notifications, enabling blood banks to send alerts to donors about donation events or urgent blood requirements. The system is hosted on a local or cloud-based server, using XAMPP as the development environment for easy deployment and management. This architecture is scalable, allowing for future enhancements like AI-based predictive analytics and integration with external healthcare systems.

The system's core is a centralized relational database designed to store and manage critical information like donor profiles, blood type inventory, donation history, and blood requests. Optimized schema for fast retrieval, built-in backup mechanisms, and scalability to accommodate growing data needs. Built with HTML for structure, CSS for styling, and JavaScript for interactivity, ensuring a responsive and user-friendly experience. Provides a seamless interface for donors to register, book appointments, and view their donation history, while staff can manage inventory and donor interactions.

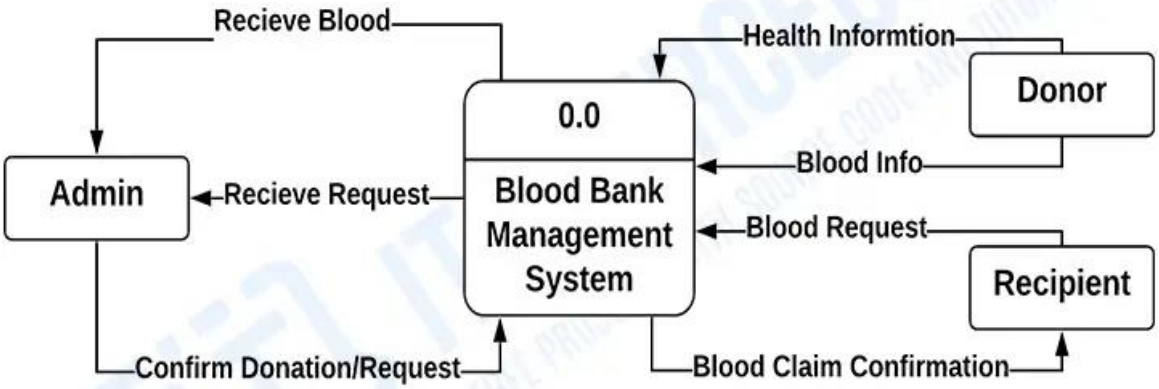


Fig : 5 System Design

Components:

- Admin: This block represents the administrative section of the system. It likely handles tasks such as:
- Managing user accounts
- Tracking blood inventory
- Generating reports
- Overseeing the overall system
- Blood Bank Management System (BBMS): This is the core of the system, responsible for managing blood donations, requests, and distribution. It interacts with both Donors and Recipients.
- Donor: This block represents individuals who donate blood. The system likely collects and stores information about donors, such as their blood type, medical history, and contact details.
- Recipient: This block represents individuals who receive blood transfusions. The system manages their medical information, blood type requirements, and transfusion history.

5.1 DATA FLOW DIAGRAM

The Data Flow Diagram (DFD) for the BLOOD DONOR NOTIFICATION SYSTEM illustrates the flow of information between the system's various components, highlighting how data moves from donors, administrators, and users to the database and back. In the first level, the DFD begins with external entities, such as donors and blood bank staff, who interact with the system through a user interface. Donors can register, update their details, and request or schedule donations. The system processes incoming data, such as blood donation records, blood type availability, and donor eligibility, and stores it in the database.

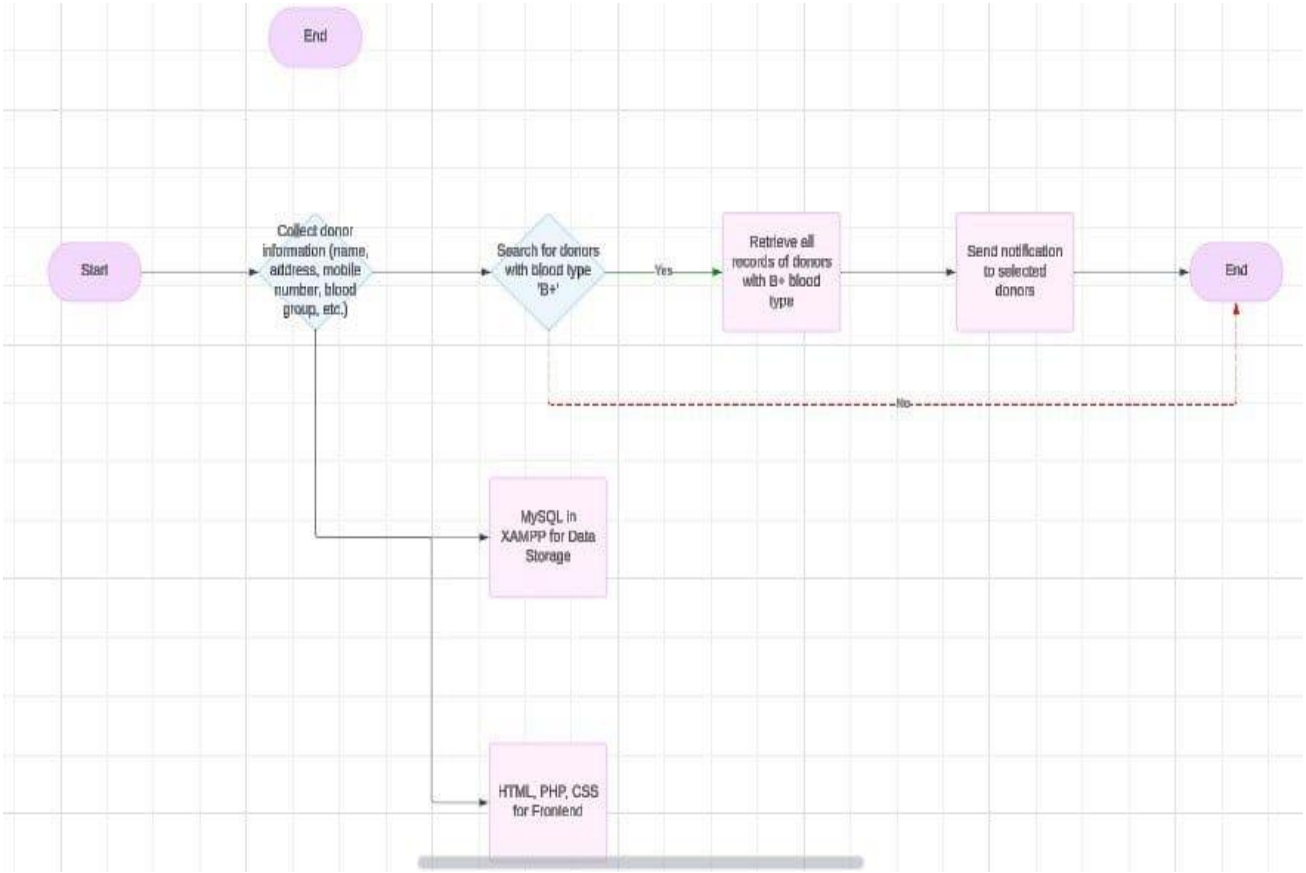


Fig 5.1 DFD Process

5.1.1 ADMIN LOGIN:

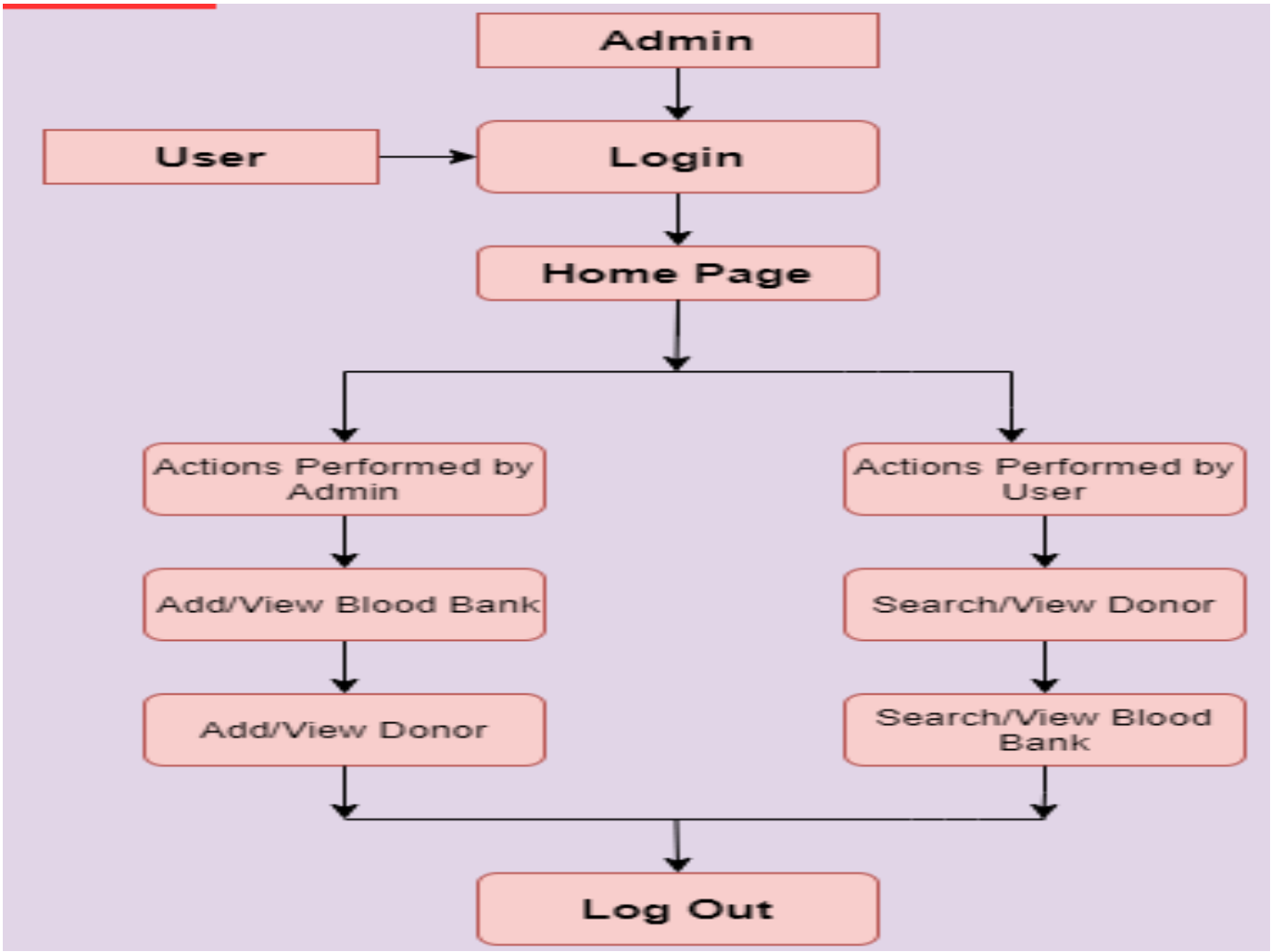


Fig: 5.1.1 Admin Process

5.1.2 UML DIAGRAM:

Unified Modeling Language (UML) is a standardized visual modeling language used in software engineering to visualize the design of a system. UML helps in specifying, visualizing, constructing, and documenting the artifacts of software systems. It provides a general-purpose, developmental, and intuitive way to represent the structure and behavior of a system**.**

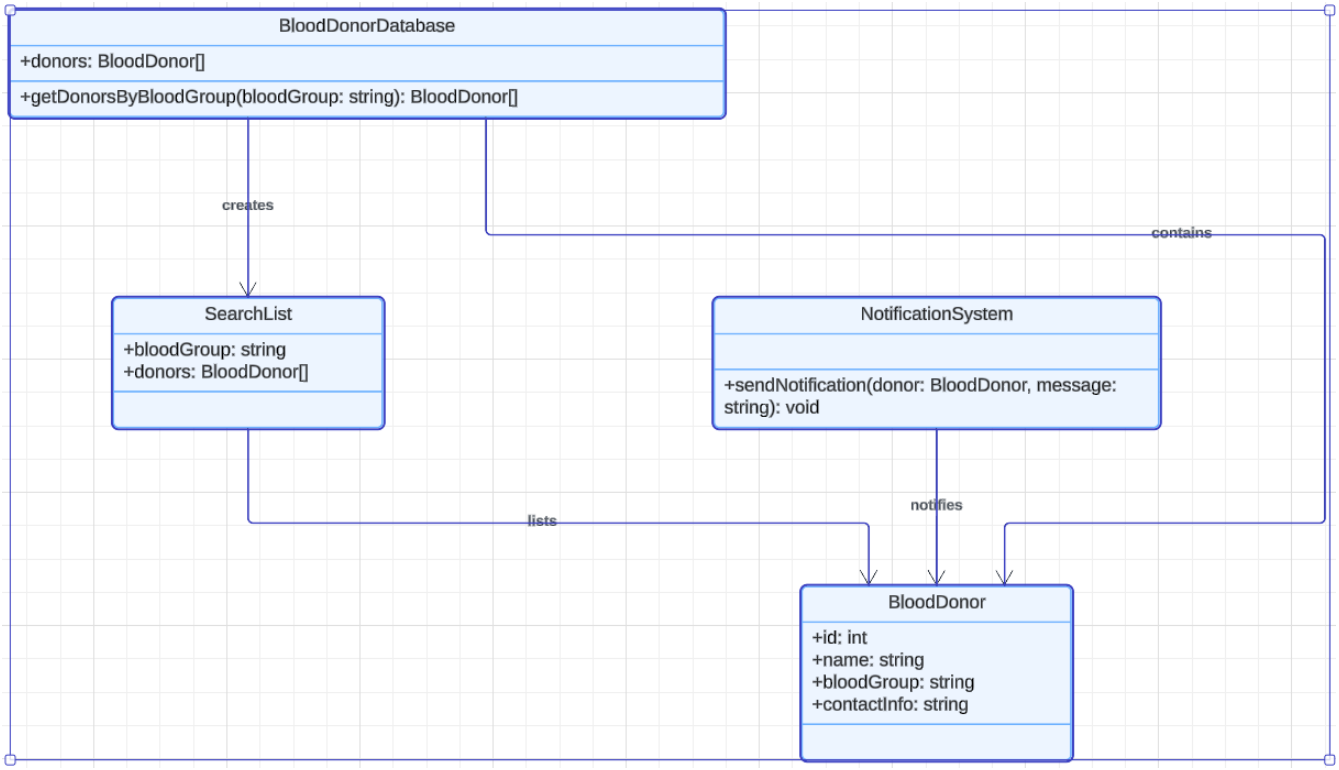


Fig: 5.1.2 UML Diagram

5.2 USE CASE DIAGRAM

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (Admin) should orcan perform in collaboration with one or more external users of the system (donor). A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

Each of these use cases explains how the system handles the actions or scenarios requested by the user.

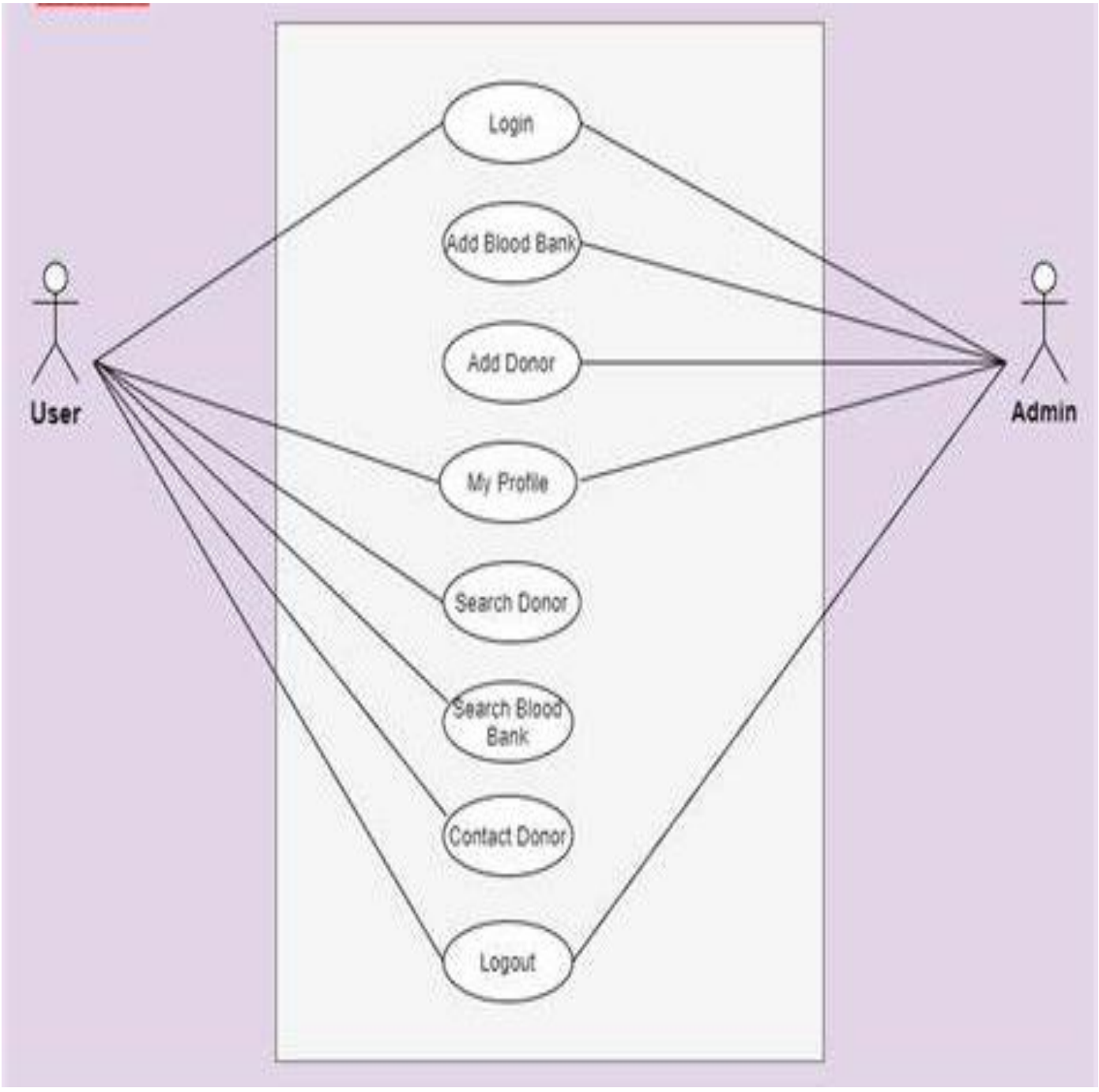


Fig 5.2 Use Case Diagram

5.3 ACTIVITY DIAGRAM

An activity diagram visually presents a series of actions or flow of control ina system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modelled can be sequential and concurrent.

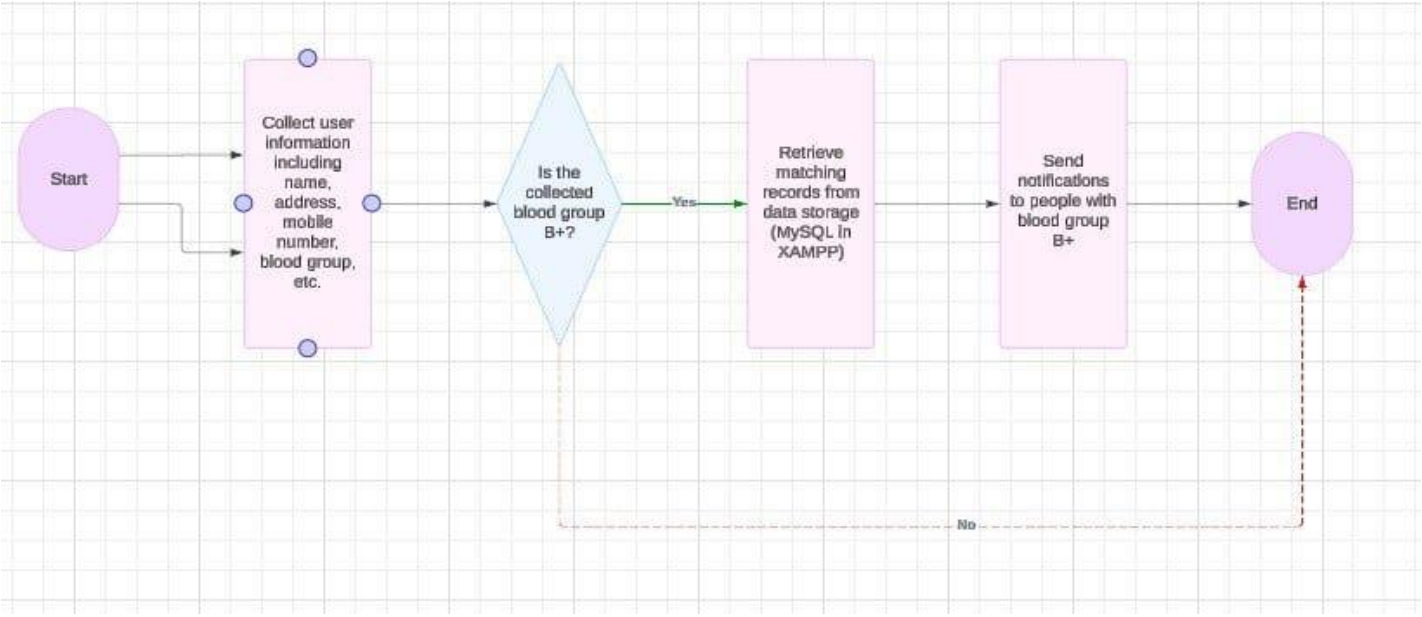


Fig 5.3 Activity Diagram

5.4 SEQUENCE DIAGRAM

The BLOOD DONOR NOTIFICATION SYSTEM is designed to streamline and automate the process of managing blood donations, donor registrations, and blood inventory for blood banks. The system provides a user-friendly platform for donors to register, update their information, and schedule blood donations, while also allowing blood bank staff to efficiently manage donor records and monitor blood stocks.

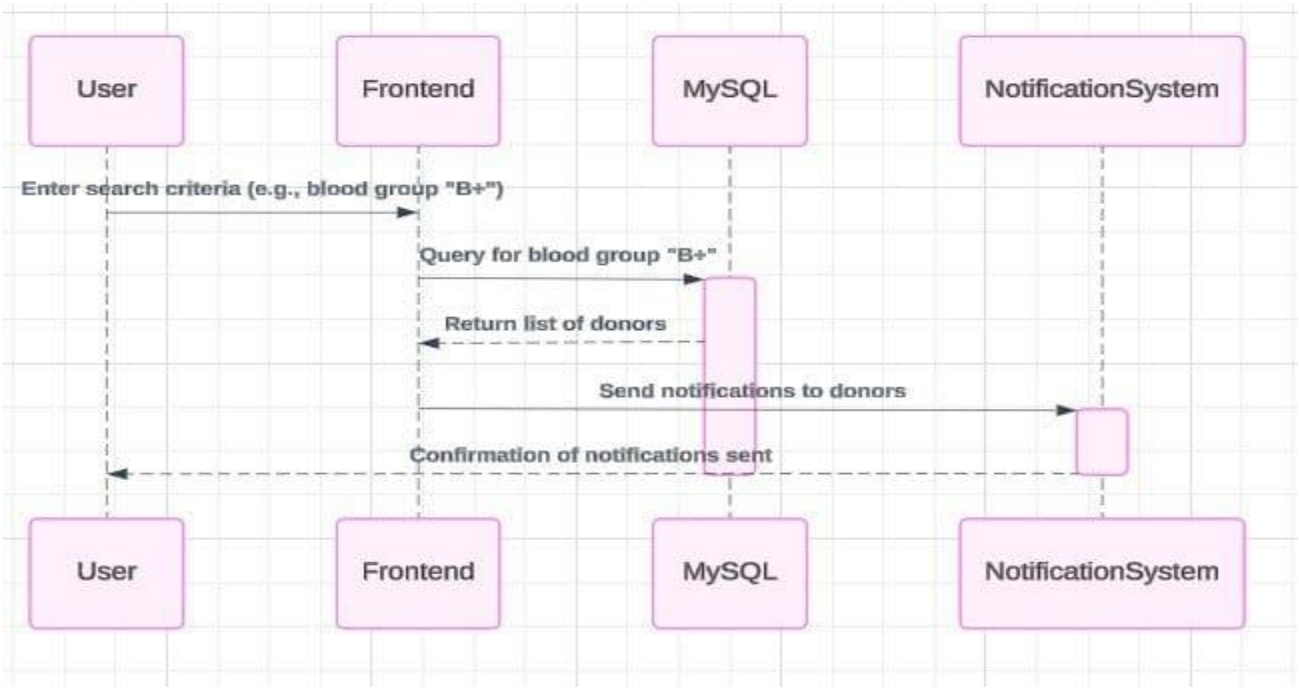


Fig 5.4 Sequence Diagram

5.4 DATABASE DESIGN:

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates.

5.5.1 DONOR ACCOUNT

Field Name	Data Type	Description
Name	Varchar(100)	Donor's name
Mobile No	Varchar(15)	Donor's mobile number
Email id	Varchar(150)	Donor mail id
Age	Int	Donor age
Blood Group	Varchar(5)	Donor's blood group type
Address	Text	Donor's address

Table 5.5.1 Donor account creation

5.5.2 ADMIN LOGIN

Field Name	Data Type	Description
User name	Varchar(150)	Admin user name
Password	Varchar(150)	Admin account password

TABLE 5.5.2 Administrator Login

5.5.3 AVAILABLE BLOOD BANK

Field Name	Data Type	Description
Name	Varchar(100)	Donor's name
Mobile No	Varchar(15)	Donor's mobile number
Email id	Varchar(150)	Donor mail id
Age	Int	Donor age
Gender	Varchar(5)	Donor's blood group type
Blood Group	Varchar(5)	Donor's blood group type
Address	Text	Donor's address
Action	String	perform delete operation

TABLE 5.5.3 Available Blood Bank

CHAPTER 6

SYSTEM REQUIREMENTS

6.1 HARDWARE REQUIREMENTS

- Operating system : Windows 11
- Coding Language : HTML, CSS XAMPP SQL AND PHP
- Tool : VS CODE
- Database : MySQL

6.2 SOFTWARE REQUIREMENTS

- System : Pentium I7
- Hard Disk : 500 GB
- Monitor : 15 VGA Color
- Mouse : Logitech
- Ram : 16GB

6.3 HARDWARE DESCRIPTION

6.3.1 WINDOW 11

Windows 11 is a major release of Microsoft's Windows NT operating system.It is the direct successor Windows 10 was made available for download via MSDNand TechNet, as a free upgrade for retail copies of Windows 8 and Windows 8.1 users via the Windows Store, and to Windows 7 users via Windows Update.

Windows 11 receives new builds on an ongoing basis, which are available at no additional cost to users, in addition to additional test builds of Windows 11, which are available to Windows Insiders. Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as securitypatches, over their ten-year lifespan of extended support In June 2021.

6.3.2 HTML

HTML, or Hyper Text Markup Language is the standard markup language used to create web pages. It is a combination of Hypertext and Markup language. The Hypertext defines the link between web pages, and Markup defines the text document within tags to structure the web pages. This language annotates text so that machines can understand and manipulate it accordingly. HTML is human-readable and uses tags to define what manipulation has to be done on the text.

6.3.3 CSS

Java is one of the most popular and widely used programming language and a platform that was developed by James Gosling in the year 1982. It is based on the concept of Object-oriented Programming. A platform is an environment in that develops and runs programs written in any programming language. Java is a high-level, object-oriented, secure, robust, platform-independent, multithreaded, and portable programming language.

Creating Java projects helps sharpen your skills and boosts your confidence as a developer. It provides practical application of theoretical knowledge. Building a portfolio showcasing completed projects empowers you for job interviews, giving you solutions, code, apps, and projects to display to recruiters.

6.3.4 PHP

JavaScript is the most powerful and versatile web programming language. It is used for making the websites interactive. JavaScript helps us add features like animations, interactive forms and dynamic content to web pages.

JavaScript is a programming language used for creating dynamic content on websites. It is a lightweight, cross-platform and single-threaded programming language. JavaScript is an interpreted language that executes code line by line providing more flexibility. It is a commonly used programming language to create dynamic and interactive elements in web applications. It is easy to learn.

6.3.5 HTML5

Hyper Text Markup Language (HTML) is the basic scripting language used by web browsers to render pages on the world wide web. It provides the foundation for all web pages by defining the layout, organization, and presentation of text, images, and multimedia.

Markup Language: HTML is a markup language used to structure content on the web by using tags and attributes to define elements and their relationships. HTML paragraphs are a fundamental part of creating structured text on a webpage. They are used to break up large blocks of text, making content more readable and accessible. Here's an in-depth explanation of paragraphs and related elements in HTML

6.4 DATABASE AND SERVER SETUP

The step-by-step instructions for setting up the database and server for the BLOOD DONOR NOTIFICATION SYSTEM using **XAMPP**, **PHP**, **MySQL**, **HTML**, and **CSS**:

6.4.1 XAMPP INSTALLATION

- Download XAMPP from the official website.
- Install XAMPP on your system. This will include Apache (web server), PHP, and MySQL.
- Launch the XAMPP Control Panel and start the **Apache** and **MySQL** services.

6.4.2 DATABASE CREATION

- Open your web browser and navigate to **phpMyAdmin** by going to: http://localhost/phpmyadmin.
- Click on the **'New'** button in the left sidebar to create a new database.
- Name your database, for example, blood_bank_system.
- Click on **'Create'**.

6.4.3 APACHE SERVER CONFIGURATION

- Go to the XAMPP installation directory (usually C:\xampp on Windows).
- Navigate to the htdocs folder inside xampp. This folder will act as the root directory for your project files.
- Create a new folder for your project. For example, create a folder named blood_bank_system.
- In the XAMPP installation folder, navigate to apache\conf.
- Open the file httpd.conf with a text editor (like Notepad++ or Visual Studio Code).

  Key configuration steps

```
DocumentRoot "C:/xampp/htdocs"
<Directory "C:/xampp/htdocs">
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

6.4.4 PHP CONFIGURATION

The PHP configuration for the BLOOD DONOR NOTIFICATION SYSTEM involves setting up a secure database connection using PHP and MySQL. The system uses a config.php file to manage database connectivity, where server parameters like hostname, username, password, and database name are specified. This file is crucial for creating a persistent connection between the PHP scripts and the MySQL database. In this project, user interactions such as donor registration and login are handled by PHP scripts that capture form inputs, validate them, and store them in relevant database tables.

6.5 COMMUNICATION TOOLS

6.5.1 SMS GATEWAY API

For integrating SMS notifications into the BLOOD DONOR NOTIFICATION SYSTEM, an SMS Gateway API is essential. The SMS Gateway API allows the system to send automated text messages to donors, blood bank administrators, and recipients for various purposes such as donation reminders, appointment confirmations, and emergency blood requests. Here's a breakdown of how to use an SMS Gateway API.

Choosing an SMS Gateway API

- **Twilio**: One of the most popular SMS gateway services, offering a simple API to send and receive SMS globally.
- **Nexmo (Vonage)**: Provides reliable SMS messaging services with easy-to-use API documentation.
- **Textlocal**: Offers a robust API for sending SMS, ideal for regional communications.
- **Plivo**: A scalable SMS API service suitable for larger volumes of messages.
- **Fast2SMS** (India-specific): An affordable and easy-to-integrate SMS gateway for local messaging in India.

Twilio API Setup Example

For this example, we'll use Twilio, a commonly used SMS API.

Step-by-Step Integration

1. Create an Account:

- Sign up at Twilio and get an account SID, Auth Token, and Twilio phone number.

2. Install PHP SDK (if using PHP):

- Use Composer to install the Twilio PHP SDK.

```
composer require twilio/sdk
```

SMS Use Cases in Project

- **Donor Registration Confirmation**: Send an SMS when a user registers as a donor.
- **Donation Reminder**: Send reminders to donors for upcoming blood donation.
- **Emergency Alerts**: Notify potential donors in a specific area during emergencies.
- **Appointment Scheduling**: Confirm and remind about donation appointments.
- **Inventory Updates**: Inform the blood bank staff when the inventory is running low.

6.5.2 EMAIL SERVICES

Email services into the BLOOD DONOR NOTIFICATION SYSTEM, you can use SMTP (Simple Mail Transfer Protocol) or an email service API. This allows the system to send automated emails to donors, blood bank administrators, and recipients for various purposes, such as donation confirmations, reminders, alerts for emergency blood needs, and notifications about events. Below is a step-by-step guide for implementing email services:

1. Email Service Options

- **SMTP (Simple Mail Transfer Protocol)**: Standard email protocol using a mail server (like Gmail, Yahoo, Outlook).
- **PHPMailer**: A popular PHP library for sending emails via SMTP with more features.
- **Third-Party Email APIs**:
    - **SendGrid**
    - **Mailgun**
    - **Amazon SES (Simple Email Service)**
    - **Mailjet**

2. Using PHP Mailer for Sending Emails

PHP Mailer is a widely-used library in PHP to send emails via SMTP, providing better security, error handling, and attachment management.

Step-by-Step Integration

1. Install PHP Mailer

- Use Composer to install PHP Mailer in your project:

```
composer require phpmailer/phpmailer
```

3. Using a Third-Party Email API (SendGrid Example)

If you prefer using an email API like **SendGrid**, follow the instructions below:

1. Create a SendGrid Account:

- Sign up at SendGrid and get an API key.

2. Install SendGrid Library:

- Install SendGrid's PHP library using Composer:

```
composer require sendgrid/sendgrid
```

4. Email Use Cases in Project

- **Donor Registration Confirmation**: Send an email when a user registers as a donor.
- **Blood Donation Confirmation**: Email confirmation to donors after successful blood donation.
- **Reminders for Blood Donation Events**: Send reminders for upcoming events or donation campaigns.
- **Urgent Appeal Notifications:** Broadcast urgent appeals for rare blood types or emergency situations requiring large quantities of blood.
- **Donor Eligibility Updates:** Notify donors when they become eligible to donate again, based on the recommended time interval between donations.
- **Notifications for Blood Requests**: Alert specific groups of donors based on blood type during emergency needs.
- **Account Recovery**: Send password reset links or account recovery emails to use and phone number.

CHAPTER 7
SYSTEM TESTING

System testing ensures that the BLOOD DONOR NOTIFICATION SYSTEM works as a whole, integrating all components and modules to verify that the system meets specified requirements. It involves checking the functionality, performance, and reliability of the system, ensuring that users can manage donor registrations, schedule donations, search blood availability, receive notifications, and perform all other functions without errors.

7.1 TESTING STEPS

- Requirement Analysis
- Test Planning
- Test Case Development
- Environment Setup
- Test Execution
- Defect Reporting
- Retesting & Regression Testing
- Test Closure

7.2 TYPES OF TESTS

7.2.1 UNIT TESTING

- **Objective**: Verify that individual components or modules of the system function correctly.
- **Application**: Test each feature of the project separately, such as:

    - **Donor Registration Module**: Check if donor data is saved correctly.
    - **Blood Search Functionality**: Ensure the search filter works for blood types.
    - **Notification System**: Test if email/SMS notifications are triggered for specific events.

- **Tools**: PHP Unit for PHP-based applications.
- **Scope**: Detect bugs at an early stage by focusing on small pieces of code.

7.2.2 SYSTEM TESTING

- **Objective**: Validate that the integrated system performs as a whole and meets the specified requirements.
- **Application**: Evaluate the end-to-end functionality, such as:

    - **Login and Authentication**: Ensure users can log in and manage their accounts securely.
    - **Blood Donation Scheduling**: Test the scheduling feature for donors.
    - **Database Operations**: Verify that donor data is correctly stored, updated, and retrieved.
    - **Notification System**: Test if all notifications are triggered appropriately for different actions (registrations, confirmations, emergencies).

- **Scope**: Ensure that all components work together as expected.

7.2.3 USER ACCEPTANCE TESTING

- **Objective:** Verify that the system meets user expectations and requirements.
- **Application:** Perform real-world testing by involving actual end-users, such as:
  - **Ease of Use:** Check if users can easily navigate the system and find relevant information.
  - **Performance:** Validate that the system is fast and responsive under typical usage scenarios.
  - **Correctness:** Test if all functionalities are accurate according to the user's needs (e.g., proper donor data management, correct blood request details).
- **Scope:** Confirm the system is ready for deployment and meets business needs.

7.2.4 SECURITY TESTING

- **Objective:** Identify vulnerabilities and ensure that the system is secure from threats.
- **Application:** Test security-related aspects, including:
  - **User Authentication:** Ensure that only registered users can access sensitive areas.
  - **Data Encryption:** Verify that sensitive data (like passwords, donor information) is securely encrypted.
  - **Injection Vulnerabilities:** Check for SQL injection risks in forms and inputs.
  - **Access Control**: Ensure proper permissions for different user roles (Admin, Donor, User).
  - **API Security:** Test if API endpoints (for SMS, Email) are secure from unauthorized access.
- **Scope:** Protect sensitive donor information and ensure the system is compliant with data privacy regulations.

CHAPTER 8

CONCLUSION AND FUTURE WORK

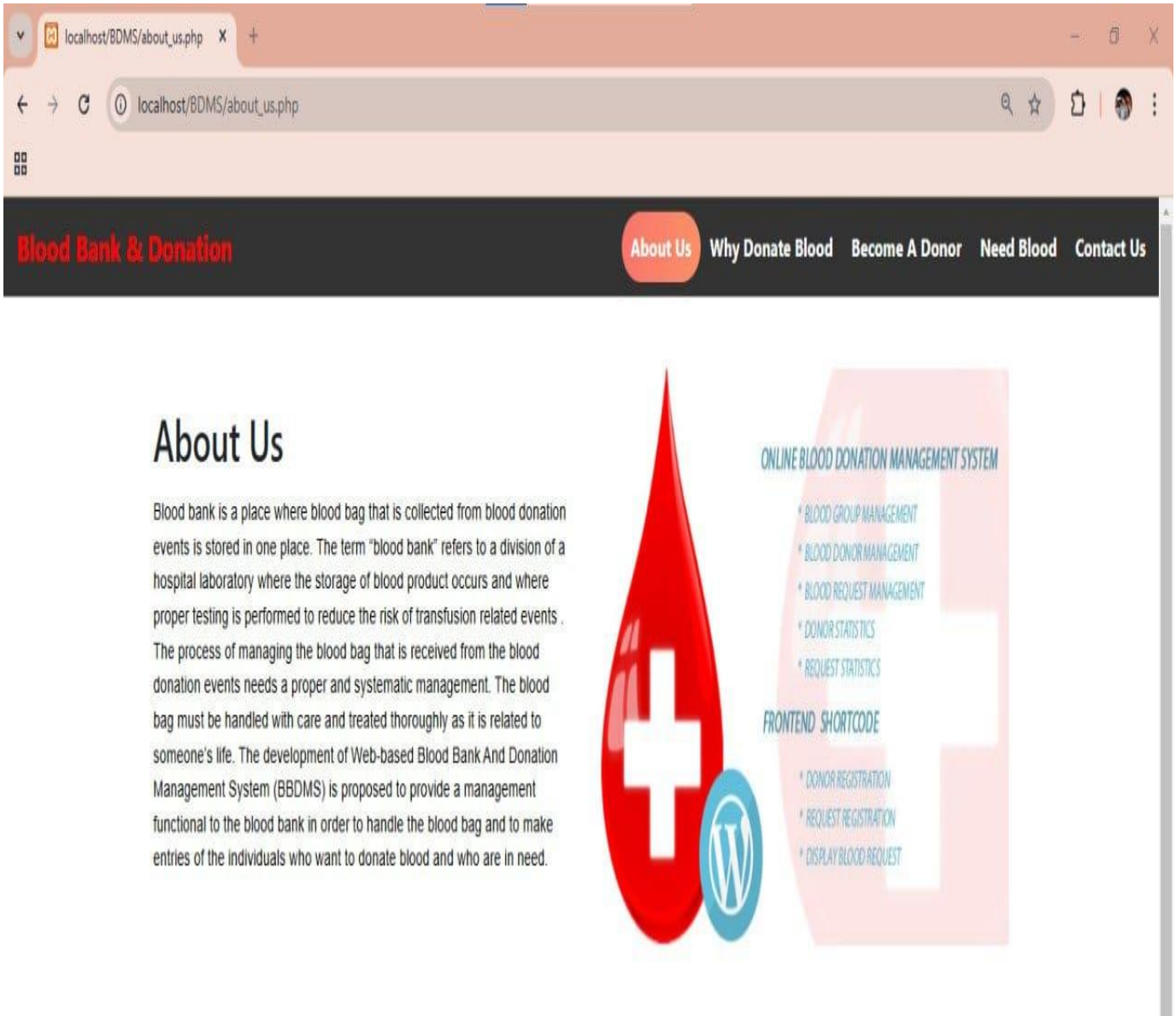
The BLOOD DONOR NOTIFICATION SYSTEM effectively streamlines the process of blood donation, making it more efficient and accessible for both donors and blood banks. By leveraging technology, the system automates tasks such as registration, blood stock management, and notifications, reducing manual errors and enhancing communication. This system provides a reliable platform that ensures blood is available when needed, ultimately contributing to saving lives and improving the overall donation experience.

Future enhancements will focus on expanding the system's capabilities through advanced technologies. This includes integrating machine learning for predictive blood demand analysis, developing a mobile application for greater accessibility, and improving security features to protect sensitive donor data. Additional features like multilingual support, blockchain for blood traceability, and gamification could further enhance user engagement and system reliability, making the platform even more efficient and impactful.

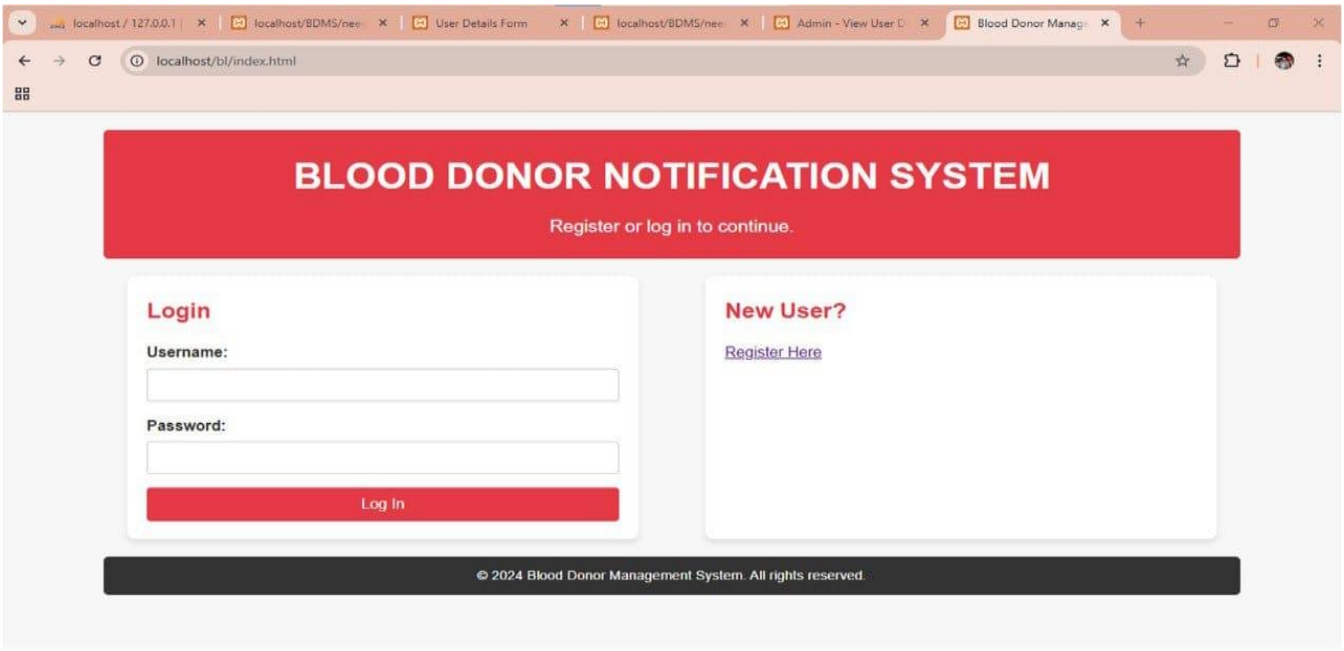
The **Blood Donor Notification System** is a vital solution that transforms the traditional blood donation process into a streamlined and technology-driven workflow. The system successfully automates critical functions such as donor registration, eligibility screening, inventory management, and notifications. These automated processes significantly reduce human errors and save valuable time for both donors and blood banks. By providing real-time updates and robust communication channels, the system ensures that blood is readily available during emergencies, ultimately saving lives and contributing to public health infrastructure. Additionally, the user-friendly design and centralized database make it a reliable and efficient platform for managing blood donation activitie

APPENDIX A

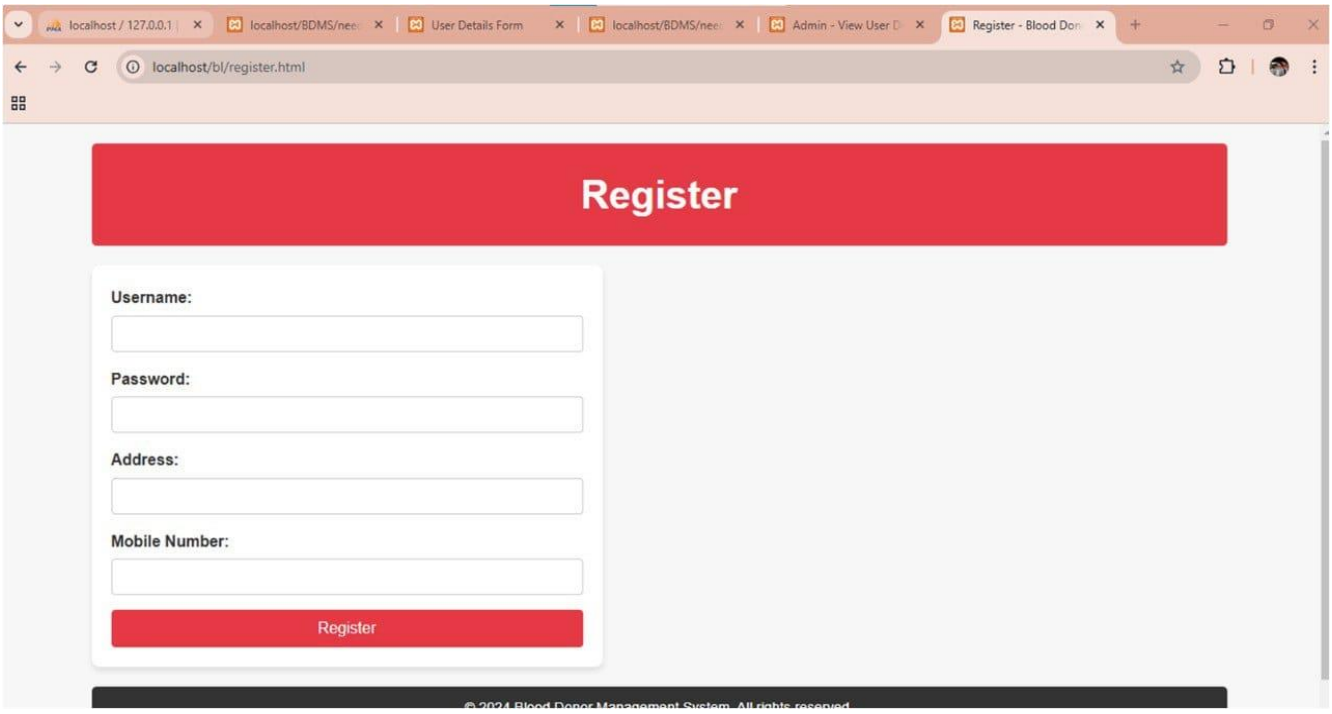
IDLE OPEN WINDOW



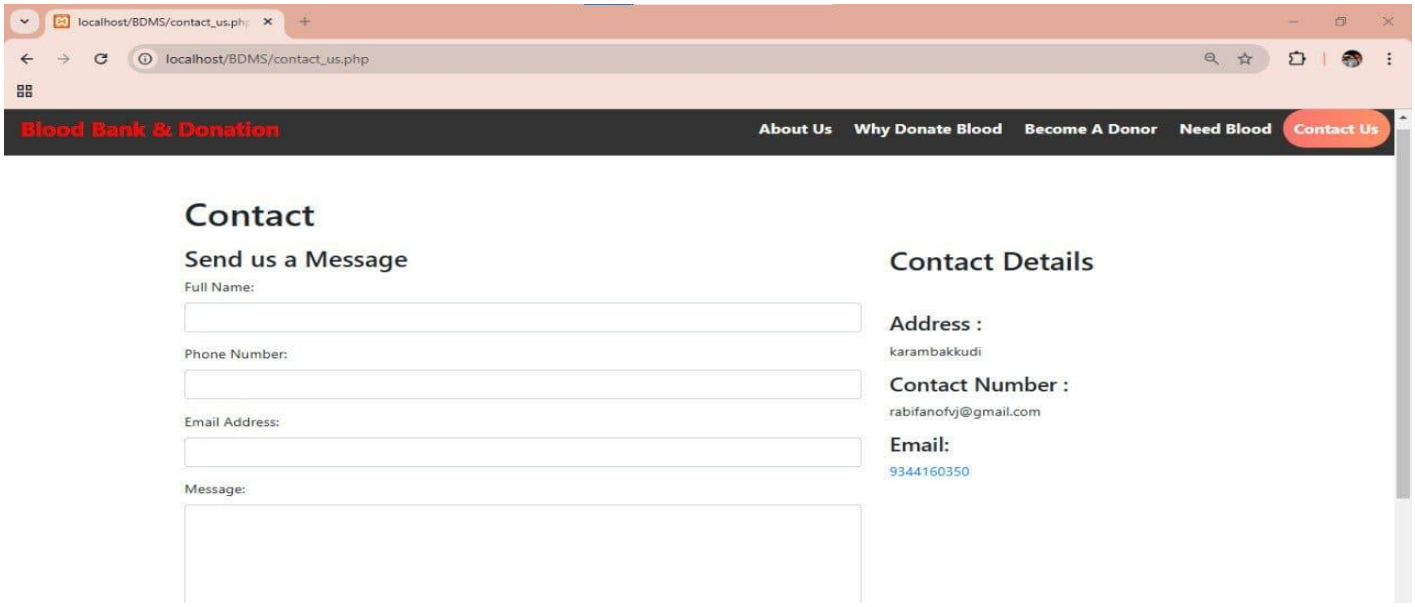
ADMIN LOGIN PAGE



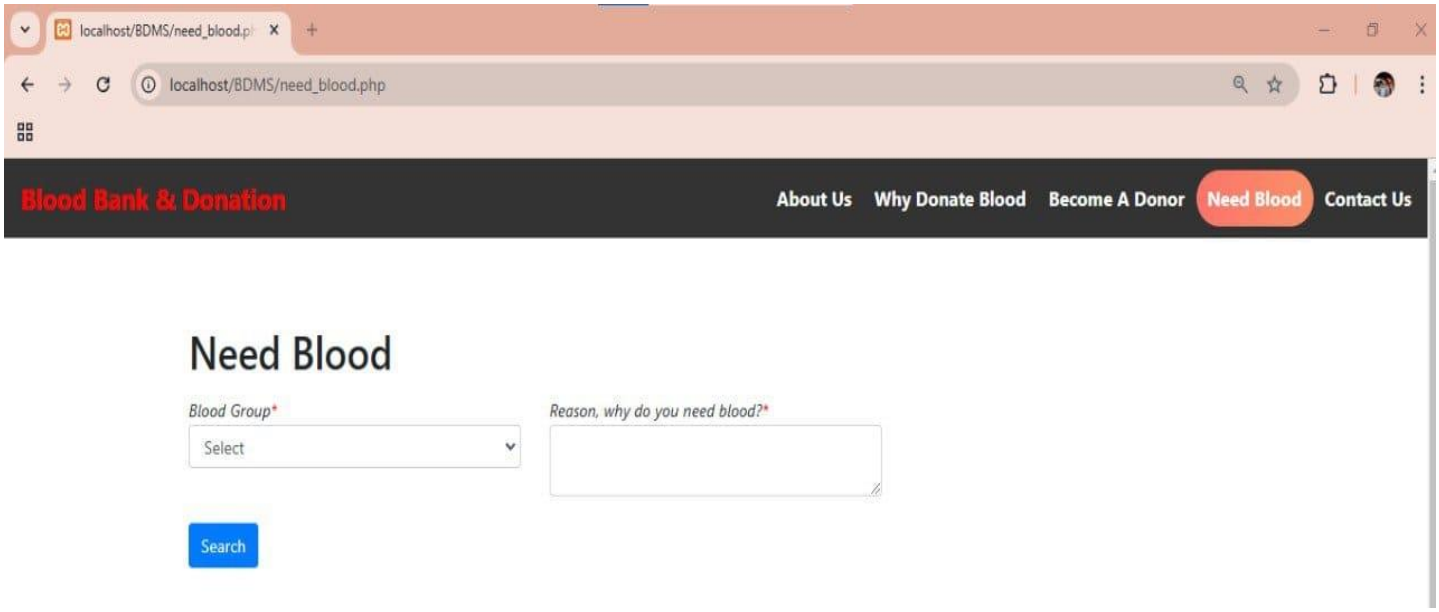
ADMIN REGISTER PAGE



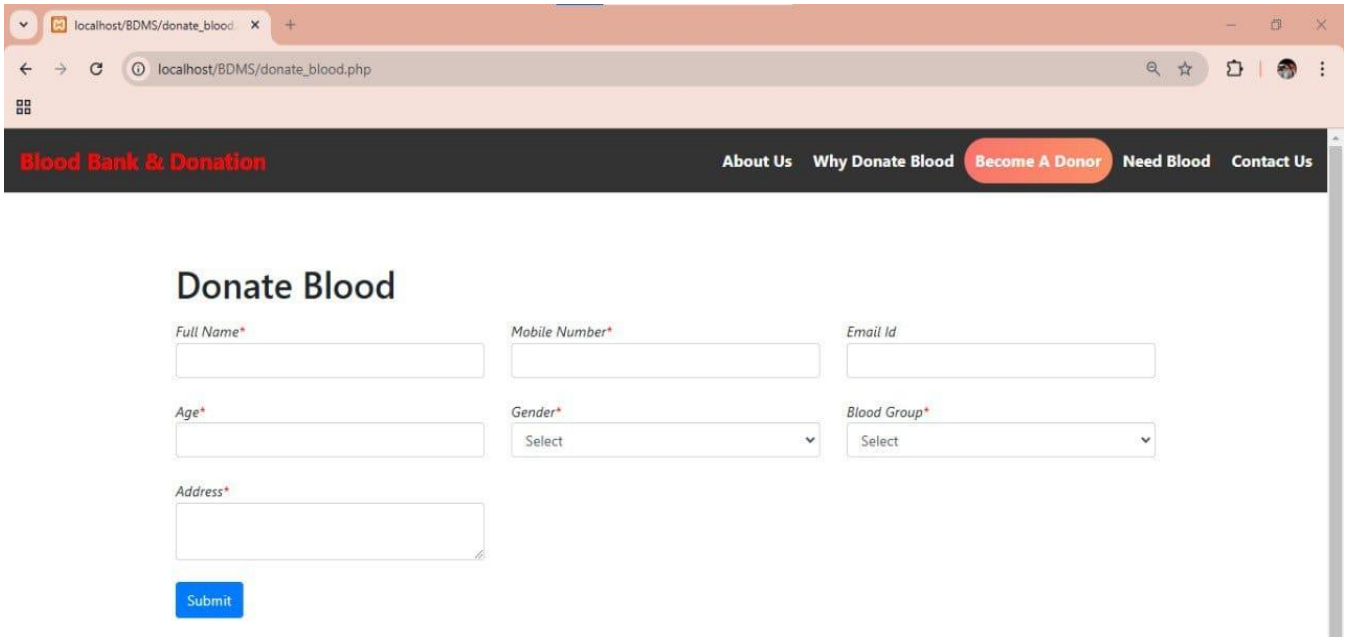
CONTACT DETAILS



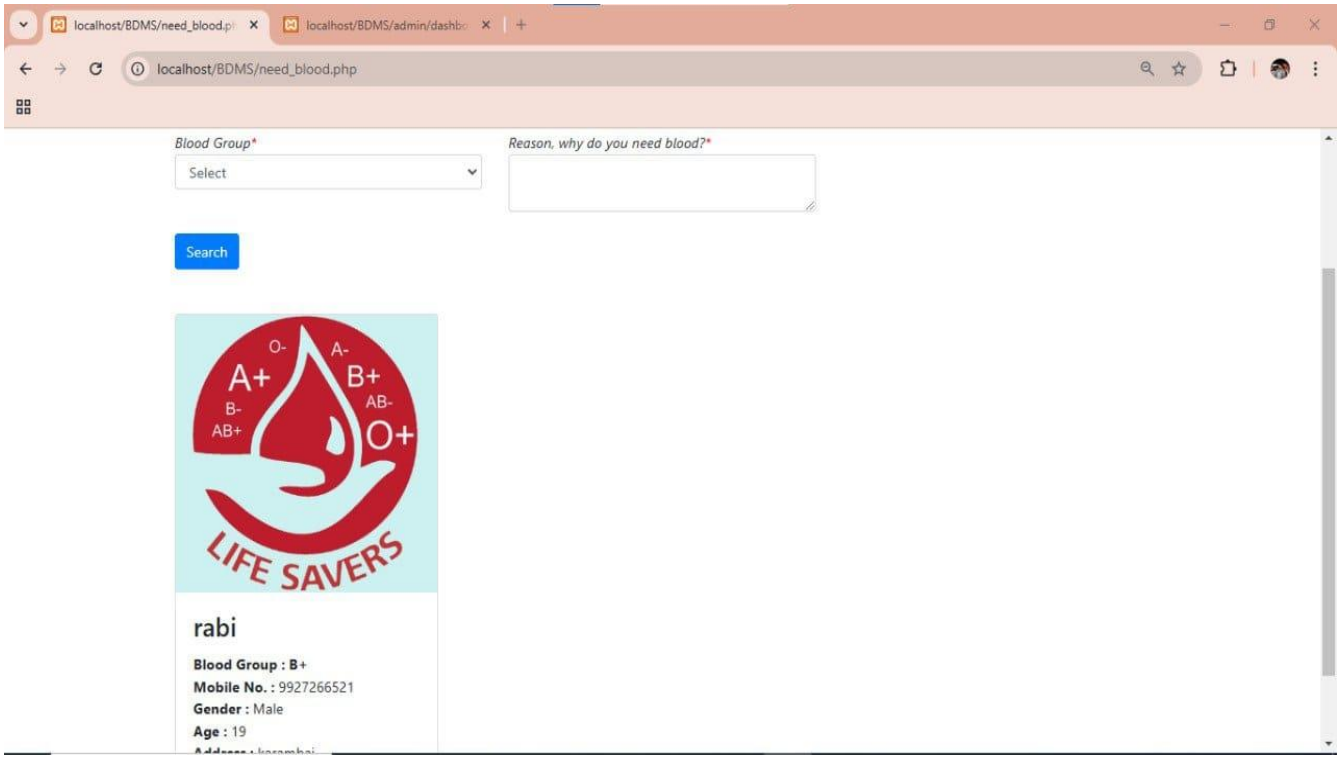
BLOOD NEED REQUEST

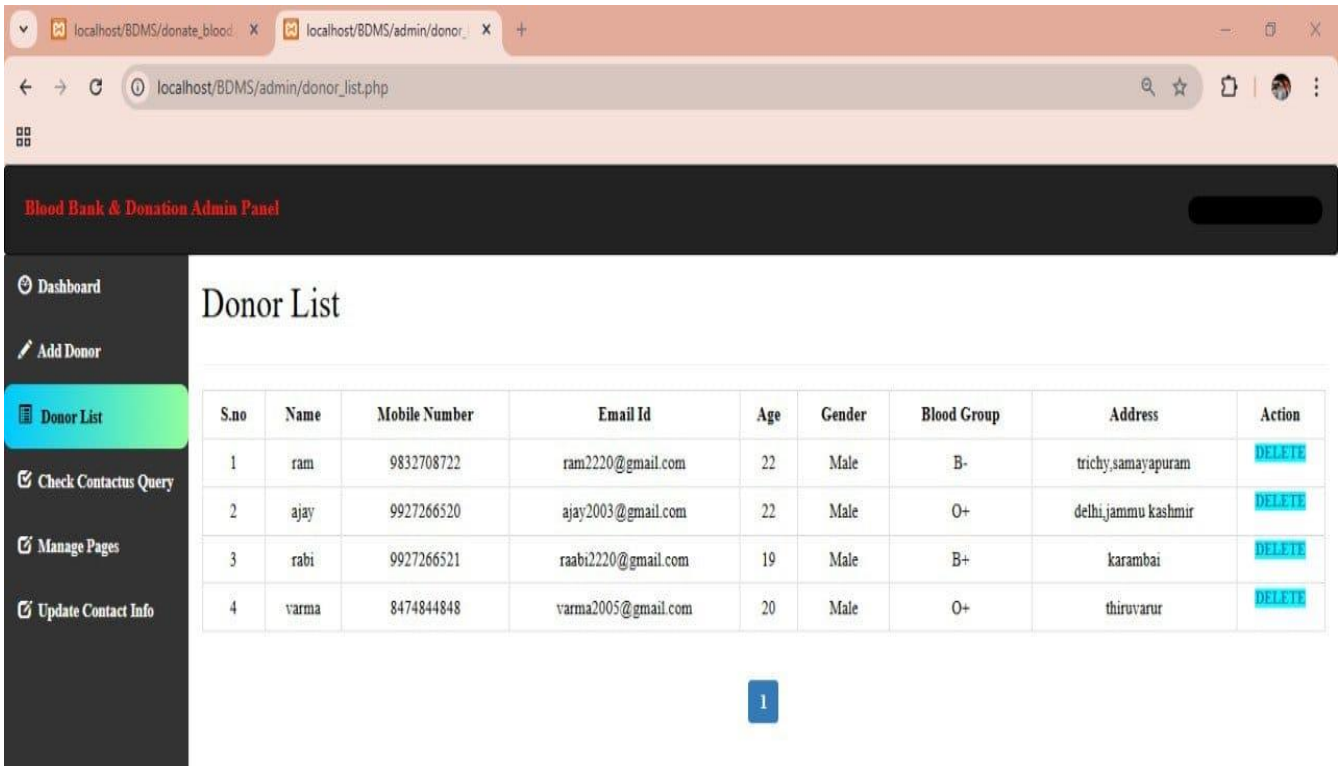


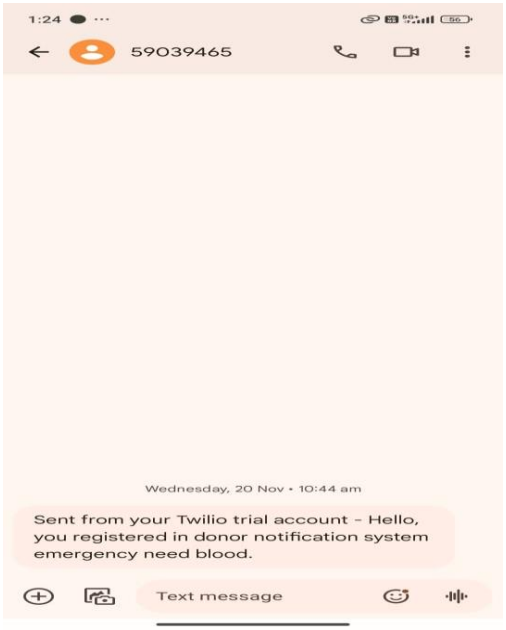
DONATE BLOOD



BLOOD DONAR LIST




localhost/BDMS/donate_blood
localhost/BDMS/admin/donor_
localhost/BDMS/admin/donor_list.php
Blood Bank & Donation Admin Panel
Dashboard
Add Donor
Donor List
Check Contactus Query
Manage Pages
Update Contact Info
Donor List
S.no Name Mobile Number Email Id Age Gender Blood Group Address Action
1 ram 9832708722 ram2220@gmail.com 22 Male B- trichy,samayapuram DELETE
2 ajay 9927266520 ajay2003@gmail.com 22 Male O+ delhi,jammu kashmir DELETE
3 rabi 9927266521 raabi2220@gmail.com 19 Male B+ karambai DELETE
4 varma 8474844848 varma2005@gmail.com 20 Male O+ thiruvarur DELETE
1


1:24
59039465
Wednesday, 20 Nov · 10:44 am
Sent from your Twilio trial account - Hello, you registered in donor notification system emergency need blood.
Text message

APPENDIX B

SOURCE CODE:

Create_database.php:

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

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
<style>
</style>
</head>

<body>
<div class="header">
<?php
$active="home";
include('head.php'); ?>

</div>
<?php include'ticker.php'; ?>

  <div id="page-container" style="margin-top:50px; position: relative;min-height: 84vh;   ">
    <div class="container">
    <div id="content-wrap"style="padding-bottom:75px;">
  <div id="demo" class="carousel slide" data-ride="carousel">

    <!-- Indicators -->
    <ul class="carousel-indicators">
      <li data-target="#demo" data-slide-to="0" class="active"></li>
      <li data-target="#demo" data-slide-to="1"></li>
    </ul>

    <!-- The slideshow -->
    <div class="carousel-inner">
```

```
    <div class="carousel-item active">
      <img src="image\_107317099_blooddonor976.jpg" alt="image\_107317099_blooddonor976.jpg" width="100%" height="500">
    </div>
    <div class="carousel-item">
      <img src="image\Blood-facts_10-illustration-graphics_canteen.png" alt="image\Blood-facts_10-illustration-graphics_canteen.png" width="100%" height="500">
    </div>

  </div>

  <!-- Left and right controls -->
  <a class="carousel-control-prev" href="#demo" data-slide="prev">
    <span class="carousel-control-prev-icon"></span>
  </a>
  <a class="carousel-control-next" href="#demo" data-slide="next">
    <span class="carousel-control-next-icon"></span>
  </a>
 </div>
<br>
    <h1 style="text-align:center;font-size:45px;">Welcome to BloodBank & Donor Management System</h1>
<br>
    <div class="row">
      <div class="col-lg-4 mb-4">
        <div class="card">
          <h4 class="card-header card bg-info text-white" >The need for blood</h4>

            <p class="card-body overflow-auto" style="padding-left:2%;height:120px;text-align:left;">
              <?php
                include 'conn.php';
                $sql=$sql= "select  from pages where page_type='needforblood'";
                $result=mysqli_query($conn,$sql);
                if(mysqli_num_rows($result)>0)   {
                   while($row = mysqli_fetch_assoc($result)) {
                     echo $row['page_data'];
                   }
                 }

              ?>
            </p>
        </div>
      </div>
      <div class="col-lg-4 mb-4">
        <div class="card">
          <h4 class="card-header card bg-info text-white">Blood Tips</h4>
```

```
            <p class="card-body overflow-auto" style="padding-left:2%;height:120px;text-align:left;">
              <?php
                include 'conn.php';
                $sql=$sql= "select  from pages where page_type='bloodtips'";
                $result=mysqli_query($conn,$sql);
                if(mysqli_num_rows($result)>0)   {
                    while($row = mysqli_fetch_assoc($result)) {
                      echo $row['page_data'];
                    }
                  }

              ?>
            </p>

              </div>
        </div>
        <div class="col-lg-4 mb-4">
           <div class="card">
              <h4 class="card-header card bg-info text-white" >Who you could Help</h
```

Need_blood.php

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

<head>

   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
   <meta name="description" content="">
   <meta name="author" content="">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
<style>
</style>
</head>

<body>
<div class="header">
<?php
$active="home";
include('head.php'); ?>
```

```
</div>
<?php include'ticker.php'; ?>

  <div id="page-container" style="margin-top:50px; position: relative;min-height: 84vh;   ">
    <div class="container">
    <div id="content-wrap"style="padding-bottom:75px;">
  <div id="demo" class="carousel slide" data-ride="carousel">

    <!-- Indicators -->
    <ul class="carousel-indicators">
      <li data-target="#demo" data-slide-to="0" class="active"></li>
      <li data-target="#demo" data-slide-to="1"></li>
    </ul>

    <!-- The slideshow -->
    <div class="carousel-inner">
      <div class="carousel-item active">
        <img  src="image\_107317099_blooddonor976.jpg"  alt="image\_107317099_blooddonor976.jpg"
width="100%" height="500">
      </div>
      <div class="carousel-item">
        <img  src="image\Blood-facts_10-illustration-graphics_canteen.png"  alt="image\Blood-facts_10-
illustration-graphics_canteen.png" width="100%" height="500">
      </div>

    </div>

    <!-- Left and right controls -->
    <a class="carousel-control-prev" href="#demo" data-slide="prev">
      <span class="carousel-control-prev-icon"></span>
    </a>
    <a class="carousel-control-next" href="#demo" data-slide="next">
      <span class="carousel-control-next-icon"></span>
    </a>
  </div>
<br>
       <h1  style="text-align:center;font-size:45px;">Welcome  to  BloodBank  &  Donor  Management
System</h1>
<br>
       <div class="row">
          <div class="col-lg-4 mb-4">
             <div class="card">
                <h4 class="card-header card bg-info text-white" >The need for blood</h4>

                   <p     class="card-body     overflow-auto"     style="padding-left:2%;height:120px;text-
align:left;">
                     <?php
                       include 'conn.php';
```

```
                $sql=$sql= "select  from pages where page_type='needforblood'";
                $result=mysqli_query($conn,$sql);
                if(mysqli_num_rows($result)>0)   {
                    while($row = mysqli_fetch_assoc($result)) {
                      echo $row['page_data'];
                    }
                  }

                ?>
               </p>
    </div>
</div>
<div class="col-lg-4 mb-4">
    <div class="card">
        <h4 class="card-header card bg-info text-white">Blood Tips</h4>

        <p class="card-body overflow-auto" style="padding-left:2%;height:120px;text-align:left;">
          <?php
            include 'conn.php';
            $sql=$sql= "select  from pages where page_type='bloodtips'";
            $result=mysqli_query($conn,$sql);
            if(mysqli_num_rows($result)>0)   {
                while($row = mysqli_fetch_assoc($result)) {
                  echo $row['page_data'];
                }
              }

           ?>
         </p>

            </div>
</div>
<div class="col-lg-4 mb-4">
    <div class="card">
        <h4 class="card-header card bg-info text-white" >Who you could Help</h
```

Donate_blood.php

```
<html>

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</head>

<body>
<?php
$active ='donate';
 include('head.php') ?>

<div id="page-container" style="margin-top:50px; position: relative;min-height: 84vh;">
  <div class="container">
  <div id="content-wrap" style="padding-bottom:50px;">
<div class="row">
    <div class="col-lg-6">
       <h1 class="mt-4 mb-3">Donate Blood </h1>
      </div>
</div>
<form name="donor" action="savedata.php" method="post">
<div class="row">
<div class="col-lg-4 mb-4">
<div class="font-italic">Full Name<span style="color:red"></span></div>
<div><input type="text" name="fullname" class="form-control" required></div>
</div>
<div class="col-lg-4 mb-4">
<div class="font-italic">Mobile Number<span style="color:red"></span></div>
<div><input type="text" name="mobileno" class="form-control" required></div>
</div>
<div class="col-lg-4 mb-4">
<div class="font-italic">Email Id</div>
<div><input type="email" name="emailid" class="form-control"></div>
</div>
</div>
<div class="row">
<div class="col-lg-4 mb-4">
<div class="font-italic">Age<span style="color:red"></span></div>
<div><input type="text" name="age" class="form-control" required></div>
</div>
<div class="col-lg-4 mb-4">
<div class="font-italic">Gender<span style="color:red"></span></div>
<div><select name="gender" class="form-control" required>
<option value="">Select</option>
<option value="Male">Male</option>
<option value="Female">Female</option>
</select>
</div>
</div>
<div class="col-lg-4 mb-4">
<div class="font-italic">Blood Group<span style="color:red"></span></div>
```

```
<div><select name="blood" class="form-control" required>
  <option value=""selected disabled>Select</option>
  <?php
    include 'conn.php';
    $sql= "select  from blood";
    $result=mysqli_query($conn,$sql) or die("query unsuccessful.");
  while($row=mysqli_fetch_assoc($result)){
   ?>
   <option value=" <?php echo $row['blood_id'] ?>"> <?php echo $row['blood_group'] ?> </option>
  <?php } ?>
</select>
</div>
</div>
</div>
<div class="row">
<div class="col-lg-4 mb-4">
<div class="font-italic">Address<span style="color:red"></span></div>
<div><textarea class="form-control" name="address" required></textarea></div></div>
</div>
<div class="row">
  <div class="col-lg-4 mb-4">
  <div><input type="submit" name="submit" class="btn btn-primary" value="Submit"
style="cursor:pointer"></div>
  </div>
</div>
</div>
</div>
<?php include('footer.php') ?>
</div>
</body>
</html>
```

Contact_us.php

```
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<meta name="description" content="">
<meta name="author" content="">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</head>
```

```
<body>
<?php $active ='contact';
include 'head.php'; ?>
<?php
if(isset($_POST["send"])){
$name=$_POST['fullname'];
$number=$_POST['contactno'];
$email=$_POST['email'];
$message=$_POST['message'];
$conn=mysqli_connect("localhost","root","","blood_donation") or die("Connection error");
$sql= "insert into contact_query (query_name,query_mail,query_number,query_message) values('{$name}','{$number}','{$email}','{$message}')";
$result=mysqli_query($conn,$sql) or die("query unsuccessful.");
echo '<div class="alert alert-success alert_dismissible"><b><button type="button" class="close" data-dismiss="alert">&times;</button></b><b>Query Sent, We will contact you shortly. </b></div>';
}?>

<div id="page-container" style="margin-top:50px; position: relative;min-height: 84vh;">
<div class="container">
<div id="content-wrap" style="padding-bottom:50px;">
<h1 class="mt-4 mb-3">Contact</h1>
<div class="row">
<div class="col-lg-8 mb-4">
<h3>Send us a Message</h3>
<form name="sentMessage"  method="post">
<div class="control-group form-group">
<div class="controls">
<label>Full Name:</label>
<input type="text" class="form-control" id="name" name="fullname" required>
<p class="help-block"></p>
</div>
</div>
<div class="control-group form-group">
<div class="controls">
<label>Phone Number:</label>
<input type="tel" class="form-control" id="phone" name="contactno"  required >
</div>
</div>
<div class="control-group form-group">
<div class="controls">
<label>Email Address:</label>
<input type="email" class="form-control" id="email" name="email" required>
</div>
</div>
<div class="control-group form-group">
<div class="controls">
<label>Message:</label>
<textarea rows="10" cols="100" class="form-control" id="message" name="message" required maxlength="999" style="resize:none"></textarea>
```

```
</div>
</div>
<button type="submit" name="send"  class="btn btn-primary">Send Message</button>
</form>
</div>
<div class="col-lg-4 mb-4">
<h2>Contact Details</h2>
<?php
include 'conn.php';
$sql= "select  from contact_info";
$result=mysqli_query($conn,$sql);
if(mysqli_num_rows($result)>0)   {
while($row = mysqli_fetch_assoc($result)) { ?>
<br>
<p>
<h4>Address :</h4><?php echo $row['contact_address']; ?>
</p>
<p>
<h4>Contact Number :</h4><?php echo $row['contact_phone']; ?>
</p>
<p>
<h4>  Email: </h4><a href="#"><?php echo $row['contact_mail']; ?></a>
</a></b>
</p>
<?php }
} ?>
</div>
</div>
<!-- /.row -->


</div>
</div>
<?php include 'footer.php' ?>
</div>
</body>

</html>
```

About_us.php

```
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<meta name="description" content="">
<meta name="author" content="">
```

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>
<style>
</style>
<body>


<?php
$active ='about';
include('head.php');

?>


<div id="page-container" style="margin-top:50px; position: relative;min-height: 84vh;">
<div class="container">
<div id="content-wrap" style="padding-bottom:50px;">
<div class="row">
<div class="col-lg-6">
<h1 class="mt-4 mb-3">About Us</h1>
<p> <?php
include 'conn.php';
$sql=$sql= "select  from pages where page_type='aboutus'";
$result=mysqli_query($conn,$sql);
if(mysqli_num_rows($result)>0)   {
while($row = mysqli_fetch_assoc($result)) {
echo $row['page_data'];
}
}

?>
</p>

</div>
<div class="col-lg-6">
<img class="img-fluid rounded" src="image\banner_590x300.jpg" style="height:400px" alt="error"  >
</div>
</div>
</div></div>

<?php include('footer.php')
?>
</div>
</body>
```

</html>

JAVASCRIPT

```
var bkExtend = function() {
var A = arguments;
if (A.length == 1) {
A = [this, A[0]];
}
for (var B in A[1]) {
A[0][B] = A[1][B];
}
return A[0];
};

function bkClass() {}
bkClass.prototype.construct = function() {};

bkClass.extend = function(C) {
var A = function() {
if (arguments[0] !== bkClass) {
return this.construct.apply(this, arguments);
}
};
var B = new this(bkClass);
bkExtend(B, C);
A.prototype = B;
A.extend = this.extend;
return A;
};

var bkElement = bkClass.extend({
construct: function(B, A) {
if (typeof B == "string") {
B = document.createElement(B);
}
B = $BK(B);
return B;
},
appendTo: function(A) {
A.appendChild(this);
return this;
},
appendBefore: function(A) {
A.parentNode.insertBefore(this, A);
return this;
},
```

```
addEvent: function(B, A) {
bkLib.addEvent(this, B, A);
return this;
},
setContent: function(A) {
this.innerHTML = A;
return this;
},
pos: function() {
var C = 0, curtop = 0;
var B = obj = this;
if (obj.offsetParent) {
do {
C += obj.offsetLeft;
curtop += obj.offsetTop;
} while (obj = obj.offsetParent);
}
var A = (!window.opera) ? parseInt(this.getStyle("border-width") || this.style.border) || 0 : 0;
return [C + A, curtop + A + this.offsetHeight];
},
noSelect: function() {
bkLib.noSelect(this);
return this;
},
parentTag: function(A) {
var B = this;
do {
if (B && B.nodeName && B.nodeName.toUpperCase() == A) {
return B;
}
B = B.parentNode;
} while (B);
return false;
},
hasClass: function(A) {
return this.className.match(new RegExp("(\\s|^)nicEdit-" + A + "(\\s|$)"));
},
addClass: function(A) {
if (!this.hasClass(A)) {
this.className += " nicEdit-" + A;
}
return this;
},
removeClass: function(A) {
if (this.hasClass(A)) {
this.className = this.className.replace(new RegExp("(\\s|^)nicEdit-" + A + "(\\s|$)"), " ");
}
return this;
},
```

```
setStyle: function(A) {
var B = this.style;
for (var C in A) {
switch (C) {
case "float":
B.cssFloat = B.styleFloat = A[C];
break;
case "opacity":
B.opacity = A[C];
B.filter = "alpha(opacity=" + Math.round(A[C]  100) + ")";
break;
case "className":
this.className = A[C];
break;
default:
B[C] = A[C];
}
}
return this;
},
getStyle: function(A, C) {
var B = (!C) ? document.defaultView : C;
if (this.nodeType == 1) {
return (B && B.getComputedStyle) ? B.getComputedStyle(this, null).getPropertyValue(A) :
this.currentStyle[bkLib.camelize(A)];
}
},
remove: function() {
this.parentNode.removeChild(this);
return this;
},
setAttributes: function(A) {
for (var B in A) {
this[B] = A[B];
}
return this;
}
});

var bkLib = {
isMSIE: (navigator.appVersion.indexOf("MSIE") != -1),
addEvent: function(C, B, A) {
(C.addEventListener) ? C.addEventListener(B, A, false) : C.attachEvent("on" + B, A);
},
toArray: function(C) {
var B = C.length, A = new Array(B);
while (B--) {
A[B] = C[B];
}
```

```
return A;
},
noSelect: function(B) {
if (B.setAttribute && B.nodeName.toLowerCase() != "input" && B.nodeName.toLowerCase() !=
"textarea") {
B.setAttribute("unselectable", "on");
}
for (var A = 0; A < B.childNodes.length; A++) {
bkLib.noSelect(B.childNodes[A]);
}
},
camelize: function(A) {
return A.replace(/\-(.)/g, function(B, C) {
return C.toUpperCase();
});
},
inArray: function(A, B) {
return (bkLib.search(A, B) != null);
},
search: function(A, C) {
for (var B = 0; B < A.length; B++) {
if (A[B] == C) {
return B;
}
}
return null;
},
cancelEvent: function(A) {
A = A || window.event;
if (A.preventDefault && A.stopPropagation) {
A.preventDefault();
A.stopPropagation();
}
return false;
},
domLoad: [],
domLoaded: function() {
if (arguments.callee.done) {
return;
}
arguments.callee.done = true;
for (var i = 0; i < bkLib.domLoad.length; i++) {
bkLib.domLoad[i]();
}
},
onDomLoaded: function(A) {
this.domLoad.push(A);
if (document.addEventListener) {
document.addEventListener("DOMContentLoaded", bkLib.domLoaded, null);
```

```
} else if (bkLib.isMSIE) {
document.write("<style>.nicEdit-main p { margin: 0; }</style><script id=_ie_onload defer " +
((location.protocol == "https:") ? "src='javascript:void(0)'" : "src=//0") + "></script>");
$BK("_ie_onload").onreadystatechange = function() {
if (this.readyState == "complete") {
bkLib.domLoaded();
}
};
}
window.onload = bkLib.domLoaded;
}
};

function $BK(A) {
if (typeof A == "string") {
A = document.getElementById(A);
}
return (A && !A.appendTo) ? bkExtend(A, bkElement.prototype) : A;
}

var bkEvent = {
addEvent: function(A, B) {
if (B) {
this.eventList = this.eventList || {};
this.eventList[A] = this.eventList[A] || [];
this.eventList[A].push(B);
}
return this;
},
fireEvent: function(A) {
if (this.eventList && this.eventList[A]) {
for (var i = 0; i < this.eventList[A].length; i++) {
this.eventList[A][i]();
}
}
return this;};}
```

REFERENCES

[1] A. Kumar, R. Sharma, and S. Soni, "Blood Donor Notification System using Mobile Application," Proceedings of the 2022 IEEE International Conference on Healthcare Informatics (ICHI), pp. 215-220, August 2022, doi: 10.1109/ICHI53548.2022.9234987.

[2] M. R. Ali, S. M. S. Ahmed, and S. T. Niazi, "Design and Implementation of a Blood Donation Management System," IEEE Access, vol. 10, pp. 35842-35850, 2022, doi: 10.1109/ACCESS.2022.3142497.

[3] V. Gupta, S. Tiwari, and P. Singh, "Smart Blood Donation System for Emergency Situations," Proceedings of the 2021 IEEE International Conference on Smart Healthcare (ICSH), pp. 189-194, November 2021, doi: 10.1109/ICSH53783.2021.9561792.

[4] S. P. Gohil, S. Patel, and S. Mehta, "A Real-time Blood Donor Notification and Matching System," IEEE Transactions on Biomedical Engineering, vol. 68, no. 6, pp. 1774-1780, June 2021, doi: 10.1109/TBME.2020.2999034.

[5] A. R. Dey, M. Islam, and K. Roy, "Developing an Automated Blood Donation Notification System Using Cloud Computing," IEEE Cloud Computing, vol. 8, no. 1, pp. 35-42, January 2020, doi: 10.1109/CC.2020.2968397.

[6] R. V. Shankar and N. K. Prasad, "An Integrated Blood Donation Notification System with a Web and Mobile Application," Proceedings of the 2019 IEEE International Conference on Artificial Intelligence and Computer Engineering (ICAICE), pp. 45-50, December 2019, doi: 10.1109/ICAICE48760.2019.8984543.

[7] J. Sharma, V. Kumar, and A. K. Mishra, "Blood Donation Alert System for Hospitals Using IoT," IEEE Transactions on Industrial Informatics, vol. 17, no. 5, pp. 33143322, May 2021, doi: 10.1109/TII.2020.2965671

[8] T. B. Rajan, N. P. S. Kumar, and R. Kumar, "An IoT-based Blood Donation and Notification System for Healthcare Management," IEEE Access, vol. 9, pp. 23456-23464, 2021, doi: 10.1109/ACCESS.2021.3056511.

[9] S. A. Raza, A. S. Khan, and M. I. Khalid, "Development of a Blood Donor Notification System Using SMS and Mobile App," Proceedings of the 2018 IEEE International Conference on Communication, Control, and Computing Technologies (ICCCCT), pp. 134-139, July 2018, doi: 10.1109/ICCCCT.2018.8882368.

[10] R. G. Kumar and P. V. Rao, "Blood Donor Awareness System with Automated Notification Mechanism," IEEE Transactions on Information Technology in Biomedicine, vol. 16, no. 4, pp. 925-930, April 2019, doi: 10.1109/TITB.2018.2923874.