BLOOD DONATION MANAGEMENT SYSTEM

A MINI PROJECT REPORT

Submitted by

BHUVANESHWARI M **220701046**

DEEPTHI P **220701059**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM

CHENNAI-602105

2023 - 24

BONAFIDE CERTIFICATE

Certified that this project report "**BLOOD DONATION MANAGEMENT SYSTEM**"

is the bonafide work of **"BHUVANESHWARI M (20701046), DEEPTHI P**

(220701059)"

who carried out the project work under my supervision.

Submitted for the Practical Examination held on _____

SIGNATURE

Dr.R.SABITHA
Professor and II Year Academic Head,
Computer Science and Engineering,
Rajalakshmi Engineering College
Thandalam, Chennai - 602 105

SIGNATURE

Mrs.S.Vinothini
Assistant Professor ,
Computer Science and Engineering,
Rajalakshmi Engineering College,
Thandalam, Chennai - 602 105

INTERNAL EXAMINER

EXTERNAL EXAMIER

ABSTRACT:

The Blood Donation Management System is a user-friendly tool that helps manage donor information and their contact details. This system makes it easy to find and contact blood donors directly when there is a need for blood. It also allows users to check the availability of different blood types.

By organizing donor details and blood stock information, the system ensures that those in need can quickly and efficiently find and obtain the required blood. This helps to save time and potentially save lives by improving the connection between donors and recipients.This system contains features that allow users to check the availability of various blood types, it simplifies the process of matching donors with recipients. By organizing donor details and maintaining up-to-date blood stock information, the system ensures quick and efficient access to the required blood supplies. Additionally, the encourages a steady supply of blood, and fosters a well-coordinated network of donors, ultimately improving the overall efficiency and responsiveness of blood donation services.

The Blood Donation Management System features an intuitive admin dashboard that serves as a centralized hub for managing all donor and recipient information. Administrators can easily access and update a comprehensive list of donors, including their contact details, blood type, donation history, and availability. The system also allows for efficient management of recipient details, such as their contact information, medical history, and specific reasons for requiring blood transfusions. This ensures that all pertinent information is readily available and up-to-date.

TABLE OF CONTENTS

Chapter 1 INTRODUCTION

1.1 INTRODUCTION

Blood donation is a critical component of healthcare services, providing essential support in medical emergencies, surgeries, and for patients with chronic conditions requiring regular transfusions. The Blood Donation Bank Management System is designed to facilitate the efficient management of blood donations, streamline donor information, and improve the accessibility of blood when needed. This system serves as a bridge between donors and recipients, ensuring that the process of blood donation and distribution is as seamless and effective as possible. The Blood Donation Management System offers a comprehensive suite of features that cater to the needs of healthcare providers, donors, and recipients. At its core, the system maintains a reliable and detailed database of donor information, including contact details, blood types, donation history, and availability status. This organized repository of information allows healthcare administrators to quickly identify and reach out to suitable donors during times of urgent need.

Developed using SQL, PHP, HTML, CSS, and JavaScript, the Blood Donation Bank Management System leverages these technologies to create a robust, user-friendly platform. SQL provides the backbone for the database management system, ensuring efficient storage and retrieval of donor and blood inventory information. PHP is utilized for server-side scripting, enabling dynamic content and interactions within the application. HTML and CSS form the foundation of the user interface, delivering a structured and visually appealing design, while JavaScript adds interactivity and enhances the overall user experience.

This report will detail the development process, system architecture, and the technologies employed in creating the Blood Donation Bank Management System. The objective is to demonstrate how the integration of these technologies results in a comprehensive solution that not only meets but exceeds the requirements for efficient blood donation management, ultimately leading to improved service delivery and patient outcomes.

1.2 OBJECTIVES

- To create a centralized database for managing donor information and blood inventory.
- To enable easy and direct contact with donors in case of emergency blood requirements.
- To provide real-time information on the availability of different blood types at various locations.
- To ensure compliance with health regulations and safety standards for blood donation and storage.
- To improve coordination between blood banks, donors, and healthcare institutions.

1.3 MODULES

1.3.1 Donor Registration Module

The Donor Registration Module is designed to capture and store detailed information about each donor. This module collects essential data such as the donor's name, contact information, blood type, medical history, and past donation records. It ensures that all donor information is accurately entered and securely stored in the database. This module includes an intuitive form that guides users through the registration process, making it easy for donors to input their details. Additionally, it verifies the eligibility of donors based on predefined criteria, ensuring that only eligible donors are registered.

1.3.2 Blood Type Search Functionality

The Blood Type Search Functionality is crucial for locating specific blood types within the inventory. This module allows administrators and users to search for available blood units based on blood type. It provides real-time updates on the availability of different blood types, helping administrators manage stock efficiently and ensuring that the right type of blood is available when needed. This module includes functionalities for filtering search results by location, blood bank, and availability status, making it easier to quickly find the required blood type in emergency situations.

1.3.3 Donor Contact Module

The Donor Contact Module facilitates direct communication between the blood bank administrators and donors. It stores and manages donor contact information, enabling administrators to send notifications and reminders for upcoming donation events, blood shortages, or eligibility for the next donation. This module supports various communication channels such as email, SMS, and phone calls, ensuring that donors can be reached promptly. It also includes features for scheduling appointments and sending thank-you messages, fostering a positive relationship between the blood bank and donors.

1.3.4 Recipient Registration and Donor Matching Module

The Recipient Registration and Donor Matching Module is designed to register recipients who need blood transfusions and match them with suitable donors. This module collects detailed information about recipients, including their medical history, blood type, and specific blood needs. It then uses this information to find compatible donors from the database, ensuring a reliable match. This module also prioritizes matches based on urgency and criticality, ensuring that those in the most critical need are attended to first. By facilitating efficient and accurate matching, this module helps save lives and improve patient outcomes.

1.3.5 Admin Dashboard Module

The Admin Dashboard Module serves as a centralized hub for administrators to manage all aspects of the blood donation system. This module provides a comprehensive overview of donor registrations, blood inventory levels, recipient information, and ongoing communications. It includes tools for generating reports, monitoring system performance, and tracking key metrics such as donation rates and blood usage. The dashboard is designed with an intuitive interface, allowing administrators to easily navigate through different sections, access detailed records, and perform administrative tasks efficiently. It also features role-based access control to ensure that sensitive information is only accessible to authorized personnel.

1.3.6 Database Module

The Database Module is the backbone of the Blood Donation Bank Management System, responsible for storing and managing all data related to donors, recipients, blood inventory, and transactions. This module ensures data integrity, security, and efficient retrieval through structured queries and indexing. It uses SQL for database management, supporting complex queries and reporting needs.

Chapter 2 SURVEY OF TECHNOLOGIES

2.1 SOFTWARE DESCRIPTION

The Blood Donation Bank Management System utilizes a combination of technologies to ensure a robust and efficient system. The backend is supported by a relational database management system (RDBMS), while the frontend is designed for user-friendly interaction. Middleware technologies are used to enable seamless communication between the backend and frontend.

2.2 LANGUAGES

The system is primarily developed using PHP as the backend programming language for server-side scripting, providing dynamic content generation and database connectivity.

2.2.1 PHP

Role: PHP, or "PHP: Hypertext Preprocessor," is the primary backend programming language used for server-side scripting in the Blood Donation Bank Management System.

Usage: PHP is embedded within HTML to create dynamic web pages that interact with the database. When a PHP script is requested by the client, it is parsed and executed on the server, allowing for real-time data retrieval, updates, and interactions.

Advantages:

- **Database Connectivity**: It offers robust support for database operations, particularly with MySQL, facilitating efficient data management.
- **Server-Side Execution**: PHP code is executed on the server, ensuring secure and fast processing before sending HTML output to the client.

2.2.2 SQL

Role: SQL (Structured Query Language) is used for managing and manipulating the relational database that stores all data**.**

Usage: All information related to donors, blood inventory, and transactions in the system are stored and implemented as a relational schema using SQL.

Advantages:

- **Efficient Data Management**: SQL allows for efficient querying, updating, and management of large datasets.

2.2.3 HTML

Role : HTML (Hyper Text Markup Language) is the standard markup language used to create the web pages of the Blood Donation Bank Management System.

Usage: It structures the content, allowing for the embedding of forms, images, and other media.

Advantages:

- **Compatibility:** HTML is supported by all web browsers, ensuring that the system can be accessed universally.
- **Integration**: HTML allows embedding of other technologies such as CSS and JavaScript to enhance the functionality and appearance of web pages.

2.2.4 CSS

Role: CSS (Cascading Style Sheets) is used for styling the HTML elements

Usage: Defining the visual presentation of the Blood Donation Bank Management System's web pages.

Advantages:

- **Design Consistency:** CSS ensures a consistent look and feel across all web pages, improving the user experience.
- **Responsive Design:** CSS supports responsive design techniques, ensuring that the system is accessible on various devices with different screen sizes.

2.2.5 JavaScript

Role: JavaScript (JS) is used for client-side scripting in the Blood Donation Bank Management System.

Usage: It enables interactive features and dynamic content updates without requiring a page reload.

Advantages:

- **Interactivity:** JavaScript enhances user interaction by enabling features like form validation, dynamic content updates, and interactive elements.
- **Versatility:** JavaScript can be used across different environments, including server-side (Node.js), making it a versatile tool for developers.

Chapter 3 REQUIREMENTS AND ANALYSIS

3.1 REQUIREMENT SPECIFICATION

3.1.1 Functional Requirements

User Authentication and Authorization

- User Registration and Login: Allow users to register, create accounts, and log in securely.
- Role-Based Access Control: Assign specific permissions based on user roles (donor, recipient, admin).

Donor Registration and Profile Management

- Profile Creation and Update: Enable donors to create and update their profiles with personal and medical information.
- Appointment Scheduling: Allow donors to schedule, reschedule, and cancel donation appointments.

Blood Type Search Functionality

- Search Capability: Allow users to search for available blood units based on blood type.
- Real-Time Updates: Provide up-to-date information on blood type availability.

Recipient Registration and Donor Matching

- Recipient Data Collection: Register recipients by collecting detailed information including medical history and specific blood needs.
- Donor Matching: Match recipients with suitable donors based on compatibility and urgency.

Admin Dashboard Module

- Comprehensive Overview: Provide an overview of donor registrations, blood inventory levels, and recipient information.
- Management Tools: Include tools for generating reports, monitoring system performance, and tracking key metrics.

3.1.2 Non-Functional Requirements

Security

- Data Encryption: Ensure encryption of sensitive data both in transit and at rest.
- Compliance: Adhere to relevant health data regulations and standards (e.g., HIPAA).

Performance

- Scalability: Design the system to handle increasing numbers of users and transactions efficiently.
- Response Time: Ensure prompt responses to user actions and queries.

Reliability

- Availability: Guarantee high system availability with minimal downtime.
- Data Backup: Regularly back up data to prevent loss and ensure data recovery.

Usability

- User-Friendly Interface: Provide an intuitive and easy-to-navigate interface for all user roles.
- Accessibility: Ensure the system is accessible to users with disabilities, following web accessibility standards.

Maintainability

- Modular Design: Use a modular architecture to facilitate easy maintenance and future upgrades.
- Comprehensive Documentation: Provide detailed documentation for users and developers.

Interoperability

- System Integration: Enable integration with other health management systems, laboratories, and inventory systems.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements:

- Desktop PC or Laptop: A reliable desktop PC or laptop to host the Blood Donation Bank Management System.
- Processor: Intel® Core TM i3-6006U CPU @ 2.00GHz or equivalent for efficient processing.
- RAM: 4.00 GB RAM to handle concurrent user requests and database operations.
- System Architecture: 64-bit operating system, x64 based processor for optimal performance.
- Monitor Resolution: 1024 x 768 monitor resolution for clear display of the system interface.
- Input Devices: Keyboard and Mouse for user interaction.
- Server with high processing power and ample storage capacity
- Reliable network infrastructure

Software Requirements:

- Operating System: Windows 10
- Code editor :Visual Studio Code
- Front End: HTML, CSS, JavaScript
- Back End: PHP, MySQL
- Middleware: XAAMP (Apache, MySQL, PHP, Perl)
- Version Control: Git

3.3 ARCHITECTURE DIAGRAM

A visual diagram that provides an overall view of the blood bank management system, identifying the external entities that interact with the system and the major data flows between these entities and the system.

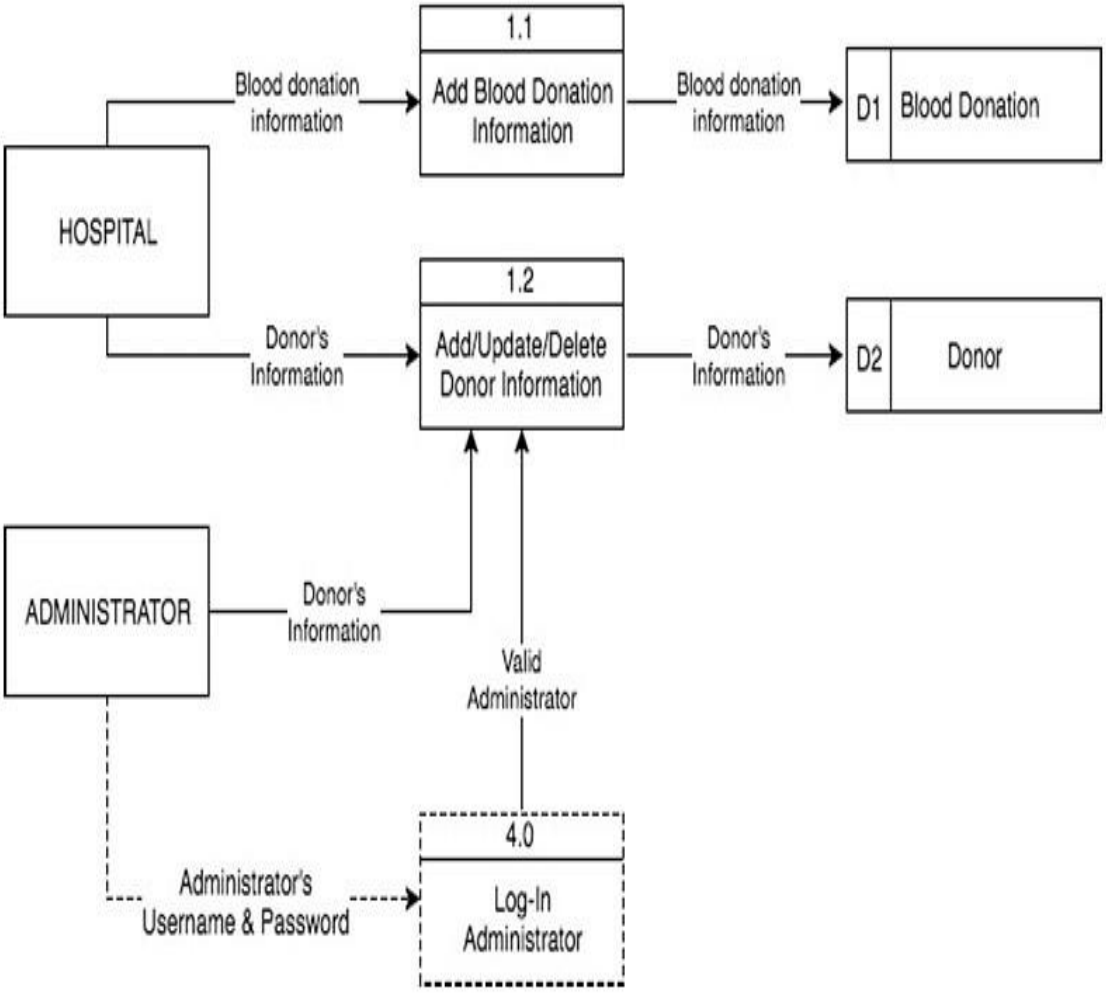

Fig 3.3 Architecture diagram

3.4 ER DIAGRAM

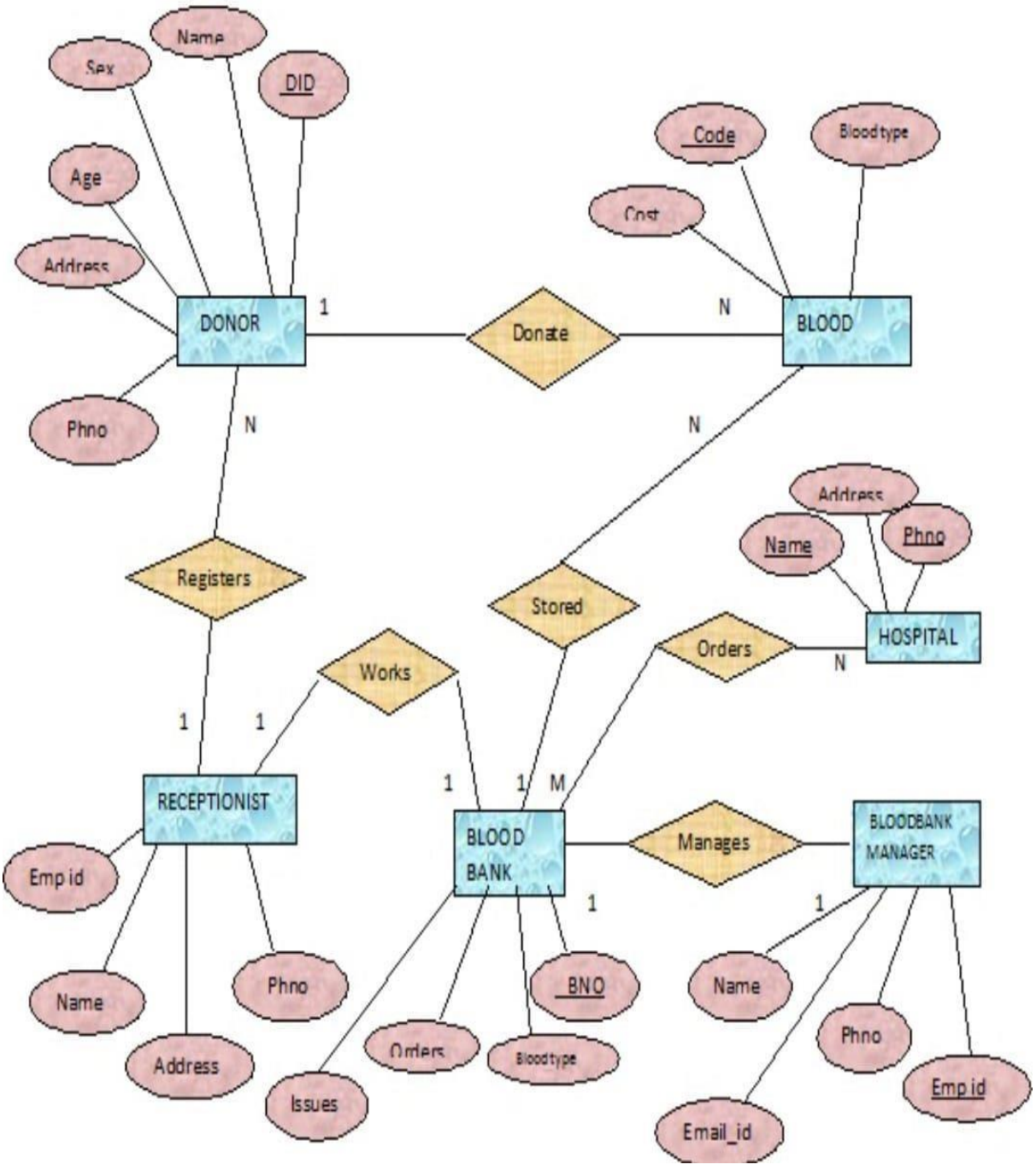

Fig 3.4 ER diagram

3.5 NORMALISATION

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. It involves dividing a database into two or more tables and defining relationships between the tables. The steps to normalize a database table for Blood Bank Management System are as follows.

Raw Database

Attribute	Datatype	Example Value
donor_name	VARCHAR(50)	JOHN
recipient_name	VARCHAR(50)	ALICE
donor_location	VARCHAR(50)	NY
Donor_phone_no	NUMBER(10,2)	9785044002
recipient_location	VARCHAR(50)	LA
Recipient_phone_no	NUMBER(10,2)	1234567890
donor_blood_group	VARCHAR(50)	A+
blood_types	VARCHAR(50)	A+,A-,B+,B-,O+,O-
reason_for_blood	VARCHAR(50)	Surgery

First Normal Form (1NF)

To achieve 1NF, we need to ensure that each cell contains only a single value and each record is unique. We'll separate the blood_types and phone number column into individual rows

donor _name	recipient _name	Donor _location	recipient _location	Recipient Phone_no	Donor Phone_no	donor_ blood_ group	Blood _type	reason_fo r_blood
JOHN	ALICE	NY	NY	1234567890	7890643562	A+	A+	SURGERY
JOHN	ALICE	NY	NY	0987654321	5637299058	A-	O+	SURGERY
JANE	BOB	LA	LA	5467342190	4563789190	B+	O-	ACCIDENT
JANE	ALICE	LA	LA	1237895456	5637820	B-	O+	ACCIDENT
……	……	…….	……	…….	…….	……	…..	…….

Second Normal Form (2NF) with Phone Numbers

To achieve 2NF, we need to ensure that all non-key attributes are fully functionally dependent on the primary key. We'll decompose the table into multiple tables to remove partial dependencies. We also include phone numbers in the respective tables.

Donor's Table

Donor_id	Donor_name	Donor_location	Donor_blood_group	donor_phone
1	John	NY	A+	0987654321
2	Jane	SF	O-	1234567890
…..	…..	……	…..	…..

Recipient's Table

Recipient_id	Recipient_name	Recipient_location	recipient_phone
1	Alice	LA	456372890
2	Bob	NY	890674524
……	…….	…….	……..

Blood Donation Table

donation_id	donor_id	recipient_id	blood_type	reason_for_blood
1	1	1	A+	Surgery
2	1	1	O+	Surgery
3	2	2	O-	Accident

Chapter 4 PROGRAM CODE

1. ADD DONOR PAGE

```
    <?php include 'session.php'; ?>
<html>

<head>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/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/3.4.1/js/bootstrap.min.js"></scr
ipt>

<style>

#sidebar{position:relative;margin-top:-20px}
#content{position:relative;margin-left:210px}
@media screen and (max-width: 600px) {
  #content {
    position:relative;margin-left:auto;margin-right:auto;
  }
}
</style>
</head>

<body style="color:black">
  <?php
  include 'conn.php';
    if (isset($_SESSION['loggedin']) && $_SESSION['loggedin'] == true) {
    ?>
<div id="header">
<?php $active="add"; include 'header.php';
?>
</div>
```

```
<div id="sidebar">
<?php include 'sidebar.php'; ?>

</div>
<div id="content">
  <div class="content-wrapper">
    <div class="container-fluid">
      <div class="row">
        <div class="col-md-12 lg-12 sm-12">

          <h1 class="page-title">Add Donor</h1>
        </div>
      </div>
      <hr>
      <form name="donor" action="save_donor_data.php" method="post">
      <div class="row">
      <div class="col-lg-4 mb-4"><br>
      <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"><br>
      <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"><br>
      <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"><br>
      <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"><br>
      <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"><br>
      <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>
      <br>
      <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> <br>
      <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" onclick="popup()"></div>
        </div>
      </div>
    </form>

      </div>
      </div>
      </div>
      <?php
    } else {
```

```
      echo '<div class="alert alert-danger"><b> Please Login First To Access
Admin Portal.</b></div>';
      ?>
      <form method="post" name="" action="login.php" class="form-
horizontal">
        <div class="form-group">
          <div class="col-sm-8 col-sm-offset-4" style="float:left">

            <button class="btn btn-primary" name="submit" type="submit">Go to
Login Page</button>
          </div>
        </div>
      </form>
   <?php }
    ?>
    <script>
    function popup() {
      alert("Data added Successfully.");
    }
    </script>
</body>
</html>
```

2 . ADMIN DASHBOARD

```
<html>
<head>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/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/3.4.1/js/bootstrap.min.js"></scr
ipt>

<style>

#sidebar{position:relative;margin-top:-20px}
```

```
#content{position:relative;margin-left:210px}
@media screen and (max-width: 600px) {
 #content {
   position:relative;margin-left:auto;margin-right:auto;
 }
}


.block-anchor {
          color:red;

          cursor: pointer;
       }
</style>
</head>

<body style="color:black;" >

 <?php
 include 'conn.php';
  include 'session.php';
  if (isset($_SESSION['loggedin']) && $_SESSION['loggedin'] == true) {
  ?>

<div id="header">
<?php include 'header.php';
?>
</div>
<div id="sidebar">
<?php
$active="dashboard";
include 'sidebar.php'; ?>

</div>
<div id="content">

 <div class="content-wrapper">
  <div class="container-fluid">
   <div class="row">
    <div class="col-md-12 lg-12 sm-12">
     <h1 class="page-title">Dashboard</h1>
    </div>
   </div>
   <hr>
```

```
    <div class="row">
      <div class="col-md-12">
        <div class="row">
          <div class="col-md-3">
            <div class="panel panel-default panel-info" style="border-
radius:50px;">
              <div class="panel-body panel-info bk-primary text-light"
style="background-color:#D6EAF8; border-radius:50px">
                <div class="stat-panel text-center">
                  <?php
                    $sql =" SELECT * from donor_details ";
                    $result=mysqli_query($conn,$sql) or die("query failed.");
                    $row=mysqli_num_rows($result);

                  ?>


                  <div class="stat-panel-number h1"><?php echo $row?></div>
                  <div class="stat-panel-title text-uppercase">Blood Donors
Available </div>
                  <br>
                    <button class="btn btn-danger" onclick="window.location.href =
'donor_list.php';">
                      Full Detail <i class="fa fa-arrow-right"></i>
                    </button>


                </div>
              </div>

            </div>
          </div>

          <div class="col-md-3">
            <div class="panel panel-default panel-info" style="border-
radius:50px;">
              <div class="panel-body panel-info bk-primary text-light"
style="background-color:#ABEBC6;border-radius:50px;">
                <div class="stat-panel text-center">
                  <?php
                    $sql1 =" SELECT * from contact_query ";
                    $result1=mysqli_query($conn,$sql1) or die("query failed.");
                    $row1=mysqli_num_rows($result1);
```

```
                ?>


                <div class="stat-panel-number h1 "><?php echo $row1?></div>
                <div class="stat-panel-title text-uppercase"> All User Queries
</div>
                <br>
                <button class="btn btn-danger" onclick="window.location.href =
'query.php';">
                  Full Detail <i class="fa fa-arrow-right"></i>
                </button>
              </div>
            </div>

          </div>
        </div>

        <div class="col-md-3">
          <div class="panel panel-default panel-info" style="border-
radius:50px;">
            <div class="panel-body panel-info bk-primary text-light"
style="background-color:#E8DAEF ;border-radius:50px; ">
              <div class="stat-panel text-center">
                <?php
                  $sql2 ="SELECT * from contact_query where query_status=2 ";
                  $result2=mysqli_query($conn,$sql2) or die("query failed.");
                  $row2=mysqli_num_rows($result2);

                ?>


                <div class="stat-panel-number h1 "><?php echo $row2 ?></div>
                <div class="stat-panel-title text-uppercase"> Pending Queries
</div>
                <br>
                <button class="btn btn-danger" onclick="window.location.href =
'pending_query.php';">
                  Full Detail <i class="fa fa-arrow-right"></i>
                </button>
              </div>
            </div>

          </div>
        </div>
```

```
        </div>
      </div>
    </div>
  <?php
 } else {
     echo '<div class="alert alert-danger"><b> Please Login First To Access Admin Portal.</b></div>';
     ?>
     <form method="post" name="" action="login.php" class="form-horizontal">
       <div class="form-group">
         <div class="col-sm-8 col-sm-offset-4" style="float:left">

           <button class="btn btn-primary" name="submit" type="submit">Go to Login Page</button>
         </div>
       </div>
     </form>
 <?php }
  ?>

</body>
</html>
```

Chapter 5 RESULTS AND DISCUSSION

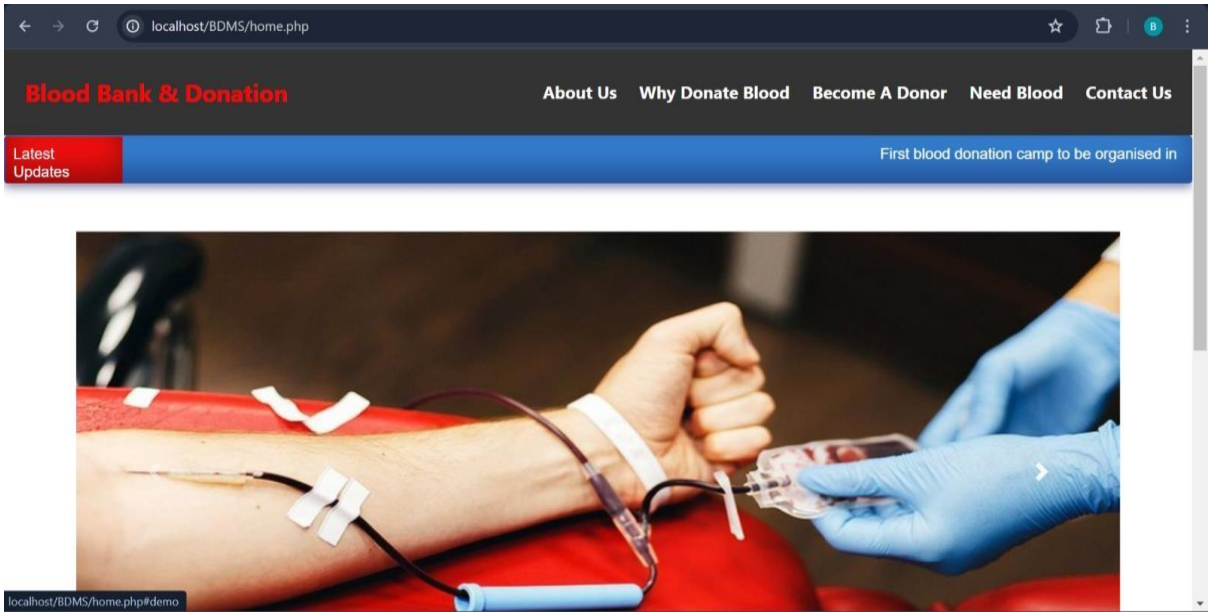

Fig 5.1 DONOR AND RECIPIENT HOME PAGE

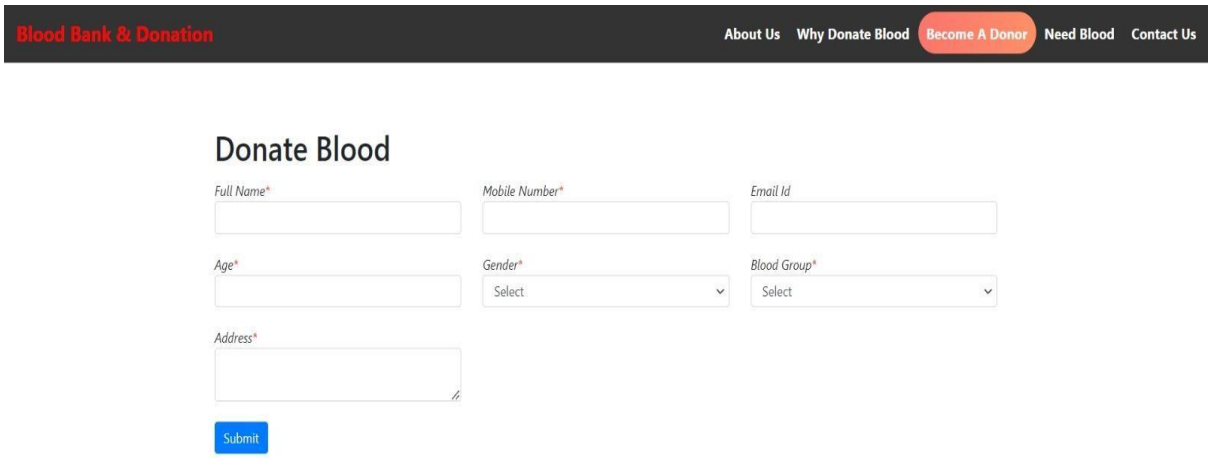

Fig 5.2 DONOR REGISTRATION

Fig 5.3 RECIPIENT REGISTRATION

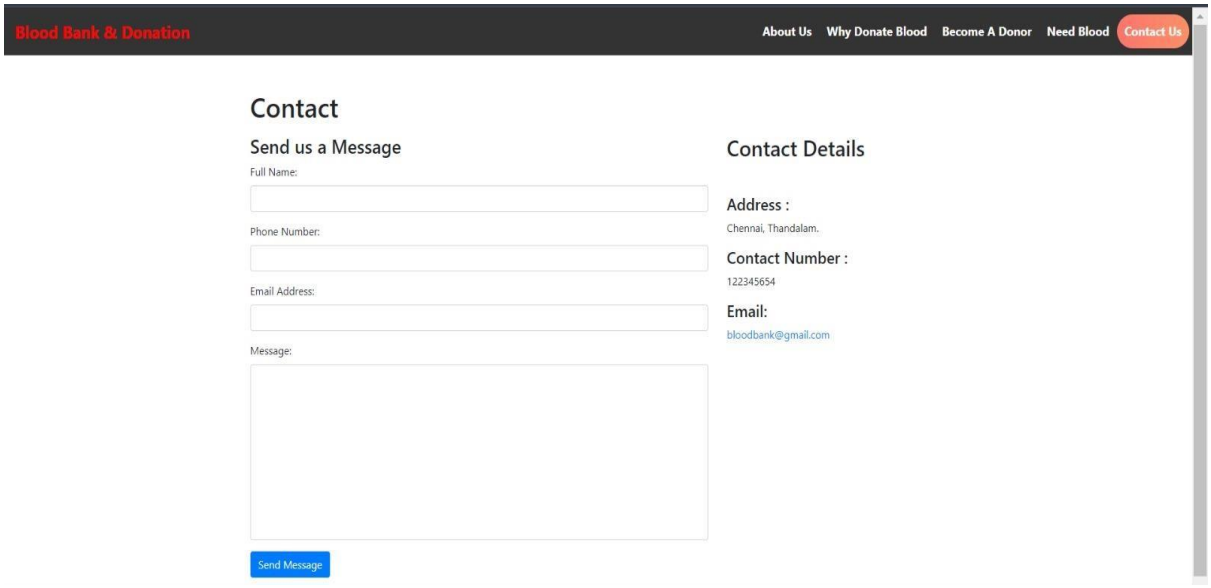

Fig 5.4 CONTACT PAGE

ADMIN LOGIN

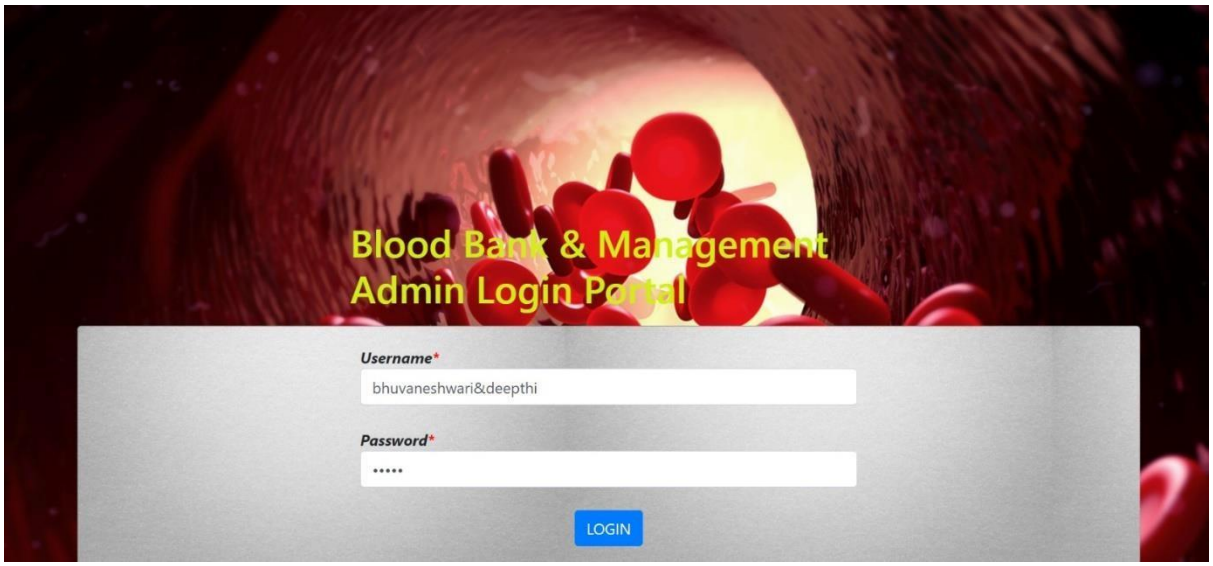

Fig 5.5 CONTACT PAGE

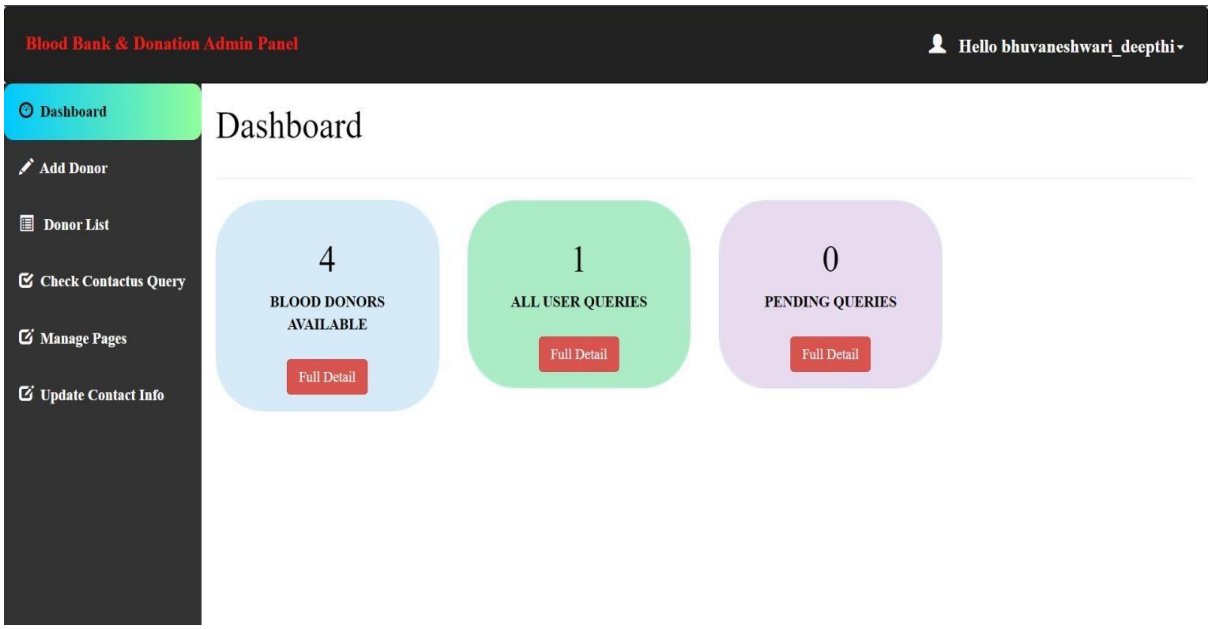

Fig 5.6 ADMIN DASHBOARD

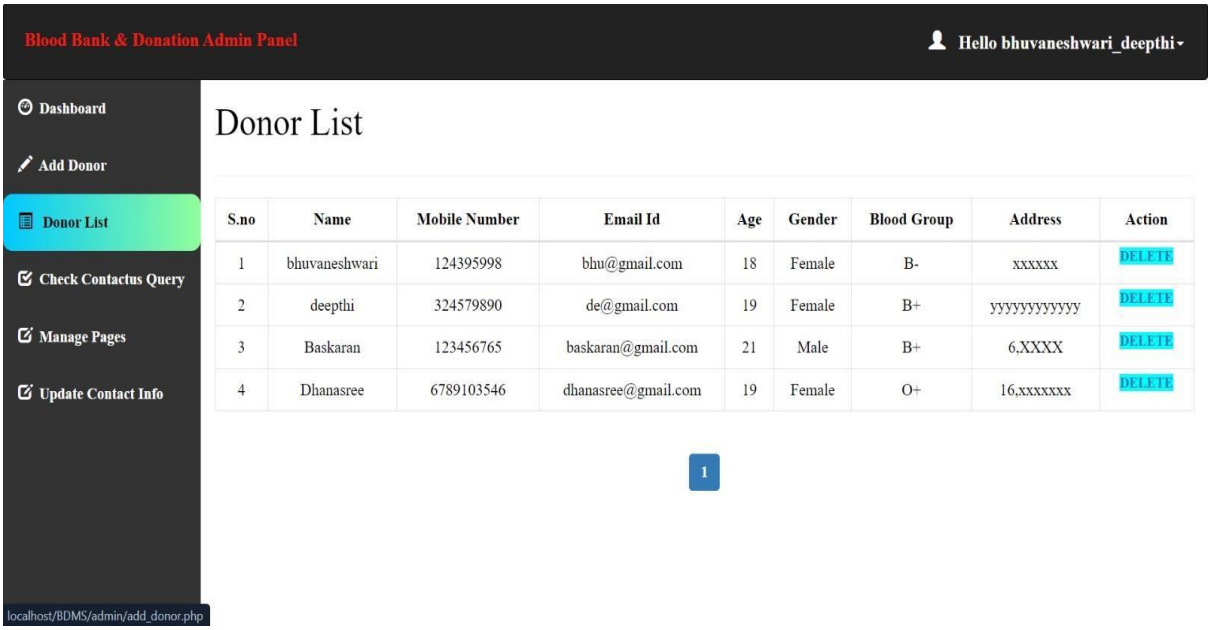

Fig 5.7 ADMIN RECORDS

Chapter 6 CONCLUSION

6.1 Conclusion

The development of the Blood Donation Management System marks a significant stride in optimizing blood donation processes and improving accessibility to life-saving resources. By harnessing a diverse array of contemporary web technologies, the system fulfills its primary objective of expediting blood donation procedures and minimizing response times, thus enriching the overall experience for both donors and recipients.

Utilizing PHP and SQL establishes a robust backbone for secure server-side processing and efficient database management. HTML, CSS, and JavaSceipt contribute to crafting a responsive and visually appealing user interface, while Incorporating modal dialogs further refines the system's usability by providing seamless and focused interaction during donation procedures.

With comprehensive functional requirements, the system facilitates critical operations such as donor registration, blood inventory management, recipient matching, and communication facilitation. Moreover, non-functional requirements ensure the system's performance, security, scalability, reliability, and maintainability, guaranteeing uninterrupted service delivery and user satisfaction.

In summary, the Blood Donation Management System effectively tackles the challenges of blood donation administration by leveraging an integrated suite of technologies. It not only meets operational demands but also establishes a benchmark for efficiency and user-centricity in healthcare services. The system's design and implementation offer a scalable and adaptable solution capable of meeting evolving needs while consistently delivering exceptional performance and enhancing public health outcomes.

Chapter 7 REFERENCES

7.1 REFERENCES

[1] https://web.ku.ac.th/saranaroo/chap5a.htm

[2] http://porcheckin.com/2014/05/blood-การบรจิ าคเลอื ด/

[3] https://www.redcross.or.th/forum/16095

[4] https://www.blood.co.uk/the-donation-process/furtherinformation/tests-we-carryout/

[5] Blood Bank Management System M Sai Tarun1, S Ravi kishan2, Shaik Azaad Suraz Basha3, Shaik Raj Ahammad4, U Chandrasekhar5, Neha Bagga6 Department of Computers Science and Engineering Lovely Professional University Jalandhar, India