BLOOD BANK MANAGEMENT SYSTEM

A

Skill Oriented Course Web Application / Project report submitted in the partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

By

DURISETI.LAVANYA SRI **22761A0579**

Under the esteemed guidance of

Mr. N. SrinivasaRao

Sr. Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING

LAKIREDDY BALIREDDY COLLEGE OF ENGINEERING

(AUTONOMOUS)

Accredited by NAAC with 'A' Grade& NBA (Under Tier - I),

An ISO 21001:2018,14001:2015,50001:2018 Certified Institution

Approved by AICTE, New Delhi and Affiliated to JNTUK, Kakinada

L.B. REDDY NAGAR, MYLAVARAM, NTR DIST., A.P.-521 230.

2022-2026

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING (AUTONOMOUS)

Accredited by NAAC with 'A' Grade& NBA (Under Tier - I),
An ISO 21001:2018,14001:2015,50001:2018 Certified Institution
Approved by AICTE, New Delhi and Affiliated to JNTUK, Kakinada
L.B. REDDY NAGAR, MYLAVARAM, NTR DIST., A.P.-521 230.

Department of
COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Skill Oriented Course Web Application / Project entitled "**BLOOD BANK MANAGEMENT SYSTEM**" is being submitted by

DURISETI.LAVANYA SRI **22761A0579**

in partial fulfillment of the requirements for the award of degree of **B.Tech** in **Computer Science & Engineering** from **Jawaharlal Nehru Technological University Kakinada** is a record of bonafide work carried out by her at **Lakireddy Bali Reddy College of Engineering (Autonomous).**

The results embodied in this Skill Oriented Course Web Application / Project report have not been submitted to any other university or institute for the award of any degree or diploma.

PROJECTGUIDE
Mr. N. SrinivasaRao

HEAD OF THEDEPARTMENT
Dr. D.Veeraiah

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I would like to thank **Mr. N. SrinivasaRao**, Sr. Assistant Professor, CSE department for the encouragement and support in carrying out this Skill Oriented Course Web Application /Project.

I also take the privilege to record my thanks to **Dr. D. Veeraiah**, Professor, Head of the Department of CSE whose encouragement, cooperation, and valuable support crown my success.

I express my thanks to the support given by management in completing my WebApplication / Project. I also express my sincere gratitude & deep sense of respect to **Dr. K. Appa Rao**, Principal for making us available all the required assistance and his supportand inspiration to carry out this Skill Oriented Course Web Application / Project in the Institute.

I am thankful to the teaching and non-teaching staff of CSE department for their direct as well as indirect help in my Skill Oriented Course Web Application / Project.

I am elated to avail myself to this opportunity to express my deep sense of gratitude to my parents.

DURISETI.LAVANYA SRI (22761A0579)

DECLARATION

I am here to declare that the project entitled **"BLOOD BANK MANAGEMENT SYSTEM"** work done by me. I certify that the work contained in the report is original and has been done by me under the guidance of my supervisor. The work has not been submitted to any other institute in preparing for any degree or diploma. I have followed the guidelines provided by the institute in preparing the report. I have confirmed to the norms and guidelines given in the Ethical Code of Conduct of the Institute. Whenever I have used materials (data, theoretical analysis, figures and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owner of the sources, whenever necessary**.**

Signature of the student: D.Lavanya Sri

DURISETI.LAVANYA SRI (22761A0579)

ABSTRACT

The Blood Bank Management System (BBMS) project aims to create an efficient and user-friendly platform for managing blood donation, storage, and distribution processes. With the increasing demand for blood transfusions in medical emergencies and routine medical procedures, effective management of blood resources becomes crucial. This project endeavors to streamline the entire workflow associated with blood donation, inventory management, donor registration, and blood request fulfillment.

The BBMS system provides a comprehensive solution to address the challenges faced by blood banks and hospitals in managing their blood inventory. It includes modules for donor registration, blood screening, inventory management, blood request processing, and donor management. Through a user-friendly interface, both donors and administrators can efficiently interact with the system, thereby enhancing the overall efficiency and transparency of blood bank operations.

Moreover, the BBMS system facilitates seamless communication between blood banks, hospitals, and donors through automated notifications and alerts. Hospitals can submit blood requests online, and the system automatically matches the request with available donors or blood units, streamlining the process of blood distribution. Overall, the Blood Bank Management System project aims to revolutionize blood bank operations, ultimately saving lives by ensuring timely access to safe and compatible blood units.

LIST OF CONTENTS

LIST OF ABBREVIATIONS

1. HTML-Hypertext Markup Language
2. CSS-Cascading Style Sheets
3. JS-JavaScript
4. JDBC – Java Database Connectivity
5. JSP – Java Server pages
6. XML- Extensible Markup Language
7. ODBC-Open Database Connectivity
8. SQL- Structured Query Language
9. API-Application Programming Interface
10. HTTP- Hypertext Transfer Protocol
11. URL-Uniform Resource Locator

INTRODUCTION

Overview of the Project:

Blood Bank Management System (BBMS) is a crucial tool in the healthcare sector designed to streamline the process of blood donation, storage, and distribution. With the constant demand for blood in medical emergencies, surgeries, and treatments, efficient management of blood resources becomes paramount. BBMS serves as a centralized platform that facilitates the coordination between blood banks, hospitals, donors, and recipients, ensuring timely access to safe and compatible blood units.

The primary objective of BBMS is to automate and optimize the various aspects of blood banking operations, thereby enhancing efficiency, accuracy, and transparency. By leveraging technology, BBMS simplifies donor registration, blood screening, inventory management, and blood request processing. It enables blood banks to maintain comprehensive records of donors, their blood types, donation history, and eligibility criteria, allowing for quick identification of suitable donors when needed.

Furthermore, BBMS incorporates advanced inventory management features to monitor blood levels, storage conditions, and expiration dates. This helps minimize wastage, ensure the availability of blood units, and adhere to regulatory standards for blood storage and handling. Additionally, BBMS facilitates seamless communication between blood banks and healthcare facilities through automated notifications and alerts, expediting the process of blood distribution and transfusion.

In summary, the implementation of a Blood Bank Management System revolutionizes blood banking practices by replacing manual and paper-based processes with efficient and technology-driven solutions. It not only improves the overall operational efficiency of blood banks but also plays a critical role in saving lives by ensuring the timely availability of safe and compatible blood for patients in need.

Feasibility Study:

A feasibility study of a Blood Bank Management System (BBMS) would typically assess the practicality and viability of implementing such a system within a specific context, such as a healthcare institution or a blood bank facility. Here's how a feasibility study for a BBMS might be conducted:

1. **Technical Feasibility:**

This aspect evaluates whether the technical requirements for implementing the BBMS can be met. It involves assessing factors such as the availability of necessary hardware and software infrastructure, compatibility with existing systems, and the technical expertise required for development and maintenance. Additionally, considerations regarding data security, scalability, and integration capabilities with other healthcare systems need to be addressed.

2. **Operational Feasibility:**

Operational feasibility examines whether the proposed BBMS can be effectively integrated into the existing workflow of the blood bank facility without disrupting day-to-day operations. It involves evaluating factors such as the ease of use of the system, training requirements for staff members, and the impact on productivity and efficiency. Assessing stakeholder acceptance and willingness to adapt to new processes is also essential in determining operational feasibility.

3. **Financial Feasibility:**

Financial feasibility involves analyzing the costs associated with developing, implementing, and maintaining the BBMS compared to the potential benefits and cost savings it offers. This includes initial investment costs for software development, hardware procurement, training expenses, as well as ongoing operational and maintenance costs. A cost-benefit analysis is conducted to determine whether the anticipated benefits, such as improved efficiency, reduced wastage, and enhanced patient care, justify the investment in the BBMS.

4. Legal and Regulatory Feasibility:

Legal and regulatory feasibility assesses whether the proposed BBMS complies with relevant laws, regulations, and industry standards governing blood banking and healthcare data management. This includes ensuring compliance with patient privacy laws (e.g., HIPAA), blood safety regulations, and data protection requirements. Addressing legal and regulatory concerns upfront is crucial to avoid potential legal liabilities and ensure the integrity and confidentiality of patient information.

5. **Schedule Feasibility:**

Schedule feasibility evaluates whether the BBMS can be developed, implemented, and deployed within a reasonable timeframe to meet the needs and expectations of stakeholders. This involves setting realistic project timelines, identifying potential bottlenecks or challenges that may cause delays, and developing a comprehensive project plan with clear milestones and deliverables.

TECHNOLOGIES USED

HTML:

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages. Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.

As its name suggests, HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

HTML Document Structure:

In its simplest form, following is an example of an HTML document:

```
<html>
    <head>
        <title>This is document file</title>
    </head>
    <body>
        All body related tags are present here
    </body>
</html>
```

HTML Text Formatting Tags:

Bold Text tag:

**
..</br>**Anything that appears within
...</br> element, is displayed in bold. Strong Text tag: **...**

Anything that appears within **...** element is displayed as important text. Italic Tag:<i> ..</*i*>.

<i>tag is used to display the content in italic.

Emphasized Text tag: ...</*em*>

Anything that appears within ...</*em*> element is displayed as emphasized text.

Underline tag:<u> ...</u>:

The HTML <u> tag is used to underline a text.

Marked Text tag: <mark>...</mark>

Anything that appears with-in <mark> ...</mark> element, is displayed as marked with yellow ink.

Strike Text tag: <strike>…</strike> or~~…~~

Anything that appears within <strike> ..</strike> or ~~…~~ element is displayed with strikethrough, which is a thin line through the text

Computer Code tag: <code>...</code>

Any programming code to appear on a Web page should be placed inside <code>...</code> tags. Usually the content of the element is presented in a monospaced font, just like the code in most programming books.

Short Quotations tag: <q> ... </q>

The <q> ...</q> element is used when you want to add a double quote within a sentence

Linking Documents: <a>.... (Hyper Links)

A link is specified using HTML tag <a> . This tag is called anchor tag and anything between the opening <a> tag and the closing tag becomes part of the link and a user can click that part to reach to the linked

document. Following is the simple syntax to use <a> tag.

Syntax:

Link Txt

Insert Image:<img>...</img> You can insert any image in your web page by using <img>tag. Following is the simple syntax to use this

tag.

CSS:

CSS is a language that describes the style of an HTML document. CSS describes how HTML elements should be displayed. CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS Syntax:

- A CSS rule-set consists of a selector and a declaration block:

```
h1 {color:blue;font-size:12px;}
```

Grouping Selectors:

If you have elements with the same style definitions, like this:

```
h1
{
text-align: center;
color: red;
}
h2
{
text-align: center;
color: red;
}
p
{
text-align: center;
color: red;
}
```

A CSS comment starts with /* and ends with */.

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

Inline Styles:

style="color: blue; margin-left: 30px ;"

Internal Style Sheet:

<style>

Body

{

Background-color: : linen;

}

h1 {

color: maroon;

margin-left: 40px;

}

External Style Sheet:

<link rel=”stylesheet” type=”text/css” href=”mystyle.css”>

JavaScript:

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with objectoriented capabilities.

➢ Script means small piece of code.

➢ Java script you can easily create interactive web pages.

➢ It is designed to add interactivity to HTML pages.

➢ Scripting languages are two kinds one is client-side other one is servers-side scripting.

In general client-side scripting is used for verifying simple validation at client side, server-side scripting is used for database verifications. VBScript, java script and J script are examples for client-side scripting and ASP,JSP, servlets etc.are examples of server-side scripting.

Web pages are two types:

1.Static web page

2.Dynamic webpage

➢ Static web page where there is no specific interaction with the client

➢ Dynamic web page which is having interactions with client and as well as validations can be added. Simple HTML script is called static web page, if you add script to HTML page it is called dynamic page.

➢ Java script code as written between <script> </script> tags.

➢ Java script ignores white space

➢ Java script is case sensitive language

➢ Script program can save as either. Js or. Html

The syntax of the script tag is as follows:

<script language=””scripting language name””>

</script>

The language attribute specifies the scripting language used in the script. Both Microsoft internet explorer and Netscape navigator use java script asthe default scripting language. The script tag may be placed in either the head or the body or the body HTML document.

EVENT HANDLING:

➢ Events are triggers that call one of your function.

➢ An event could be action Such as clicking on a button or placing your mouse over an image.

for example we will use the onclick event for starting our form validation scripts, and the on mouse overevent for creating graphics images that change when you place your cursor over them.

1.Scripts can respond to user inter actions

2.Change the page according i.e. add dynamism to the page

3.It makes web applications more responsive and user-friendly

JAVASCRIPT Form Validation:

It is important to validate the form submitted by the user because it can have inappropriate Values. So, validation is must to authenticate user. JavaScript provides facility to validate the form on the client-side so data processing will be faster than server-side validation. Most of the web developers prefer JavaScript form validation. Through JavaScript, we can validate name, password, email, date, mobile numbers and more fields.

JDBC:

JDBC is a Java API that is used to connect and execute the query to the database. JDBC API uses JDBC drivers to connect to the database. JDBC API can be used to access tabular data stored into any relational database.

Java Database Connectivity with 6 Steps:

There are 6 stepsto connect any java application with the database using JDBC.

These steps are as follows:

- Importing JDBC package
- Load/Register the Driver class
- Establish the connection
- Create statement o Execute queries
- Close connection

Importing JDBC Package:

Syntax:

```
import java.sql.*;
```

Loading/Registering the driver class:

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

Registering:

```
Oracle.jdbc.driver.OracleDriver od=new oracle.jdbc.driver.OracleDriver();

DriverManager.registerDriver(od);
```

Establishing the connection:

```
Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","password");
```

Creating the statement:

```
Statement stmt=con.createStatement();
```

Executing the queries:

```
ResultSet rs=stmt.executeQuery("select * from emp");

while(rs.next())

{

        System.out.println(rs.getInt(1)+" "+rs.getString(2));

}
```

Closing connection:

```
con.close();
```

Servlets:

Servlet technology is used to create web application (resides at server side and generates dynamic web page).

Servlet API:

- The javax.servlet and javax.servlet.http packagesrepresent interfaces and classes for servlet API.
- The javax.servlet package contains many interfaces and classes that are used by the servlet or web container. These are not specific to any protocol.

Steps to create a servlet example

There are given 6 stepsto create a servlet example. These steps are required for all the servers.

The servlet example can be created by three ways:

1. By implementing Servlet interface,

2. By inheriting GenericServlet class, (or)

3. By inheriting HttpServlet class

Here, we are going to use apache tomcat server in this example. The steps are as follows:

1. Create a directory structure

2. Create a Servlet

3. Compile the Servlet

4. Create a deployment descriptor

5. Start the server and deploy the project

6. Access the servle

Methods of GenericServlet class:

There are many methods in GenericServlet class. Some of them are given below.

1. public void init(ServletConfig config) is used to initialize the servlet.

2. public abstract void service(ServletRequest request, ServletResponse response) provides service for the incoming request. It is invoked at each time when user requests for a servlet.

3. public void destroy() is invoked only once throughout the life cycle and indicates that servlet is being destroyed.0

4. public ServletConfig getServletConfig() returns the object of ServletConfig.

5. public String getServletInfo() returns information about servlet such as writer, copyright, version etc.

6. public void init() it is a convenient method for the servlet programmers, now there is no need to call super.init(config)

HttpServlet class:

The HttpServlet class extends the GenericServlet class and implements Serializable interface. It provides http specific methods such as doGet, doPost, doHead etc.

Methods of HttpServlet class:

There are many methods in HttpServlet class. Some of them are given below:

1. public void service(ServletRequest req,ServletResponse res) dispatches the request to the protected service method by converting the request and response object into http type.

2. protected void service(HttpServletRequest req, HttpServletResponse res) receives the request from the service method, and dispatches the request to the doXXX() method depending on the incoming http request type.

3. protected void doGet(HttpServletRequest req, HttpServletResponse res) handles the GET request. It is invoked by the web container.

4. protected void doPost(HttpServletRequest req, HttpServletResponse res) handles the POST request. It is invoked by the web container.

5. protected void doHead(HttpServletRequest req, HttpServletResponse res) handles the HEAD request. It is invoked by the web container.

RequestDispatcher in Servlet:

The RequestDispatcher interface provides the facility of dispatching the request to another resource it may be html, servlet or jsp. This interface can also be used to include the content of another resource also. It is one of the way of servlet collaboration.

public void forward(ServletRequest request,ServletResponse response)throws ServletException, java.io.IOException

public void include(ServletRequest request,ServletResponse response)throws ServletException, java.io.IOException

Example of using getRequestDispatcher method

RequestDispatcher rd=request.getRequestDispatcher("myservlet");

//myservlet is the url-pattern of the servlet

rd.forward(request, response);//method may be include or forward

Cookies in Servlet:

Types of Cookie There are 2 types of cookies in servlets.

1. Non-persistent cookie

2. Persistent cookie

Non-persistent cookie:

It is valid for single session only. It is removed each time when user closes the browser.

Persistent cookie:

It is valid for multiple session . It is not removed each time when user closes the browser. It is removed only if user logout or signout.

Creating a Cookie:

Let's see the simple code to create cookie.

Cookie ck=new Cookie("user","admin");//creating cookie object

response.addCookie(ck);//adding cookie in the response

deleting a Cookie:

Cookie ck=new Cookie("user","");//deleting value of cookie

ck.setMaxAge(0);//changing the maximum age to 0 seconds

response.addCookie(ck);//adding cookie in the response

getting Cookies:

Cookie ck[]=request.getCookies();for(int i=0;i<ck.length;i++){

out.print(""+ck[i].getName()+" "+ck[i].getValue());//printing name and value of cookie }

JSP:

JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to servlet because it provides more functionality than servlet such as expression language, jstl etc.

Running a simple JSP Page:

Follow the following steps to execute this JSP page:

o Start the server

o put the jsp file in a folder and deploy on the server

o visit the browser by the url http://localhost:8090/folder/programname.jsp

JSP scriptlet tag:

<% java source code %>

JSP expression tag:

<%= statement %>

JSP Declaration Tag:

<%! Fields or method declration %>

JSP Implicit Objects:

1)JSP out implicit object:

<% out.print("Welcome to jsp"); %>

2)JSP request implicit object:

<% String name=request.getParameter("Uname"); %>

3)JSP response implicit object:

<% response.sendRedirect("http://www.google.com"); %>

4)JSP config implicit object:

String std=config.getInitParameter("student");

5)JSP application implicit object:

String std=application.getInitParameter("student");

6)exception implicit object:

<form name="form" action="welcome.jsp">

Error.jsp:

<%@ page isErroePage="true" %>

The exception is: <%= exception %>

JSP directive elements:

<%@ directive attribute="value" %>

JSP page directive:

<%@ page attribute=”value” %>

Import:

The import attribute is used to import class,interface or all the members of a package.It is similar to import keyword in java class or interface.

<%@ page import=”java.util.Date” %>

errorPage:

<%@ page errorPage=”myerror.jsp” %>

isErrorPage:

<%@ page isErrorPage=”true” %>

Jsp Include Directive:

<%@ include file=”resourceName” %>

JSP Action Tags/ Elements:

Syntax of jsp:forward action tag without parameter:

<jsp:forward page=”relationURL|<%+ expression %>”/>

Syntax of jsp:forward action tag with parameter:

<jsp:forward page=”relativeURL| <%= expression %>”>

<jsp:param name=”parametername” value=”parametervalue |<%= expression %>” />

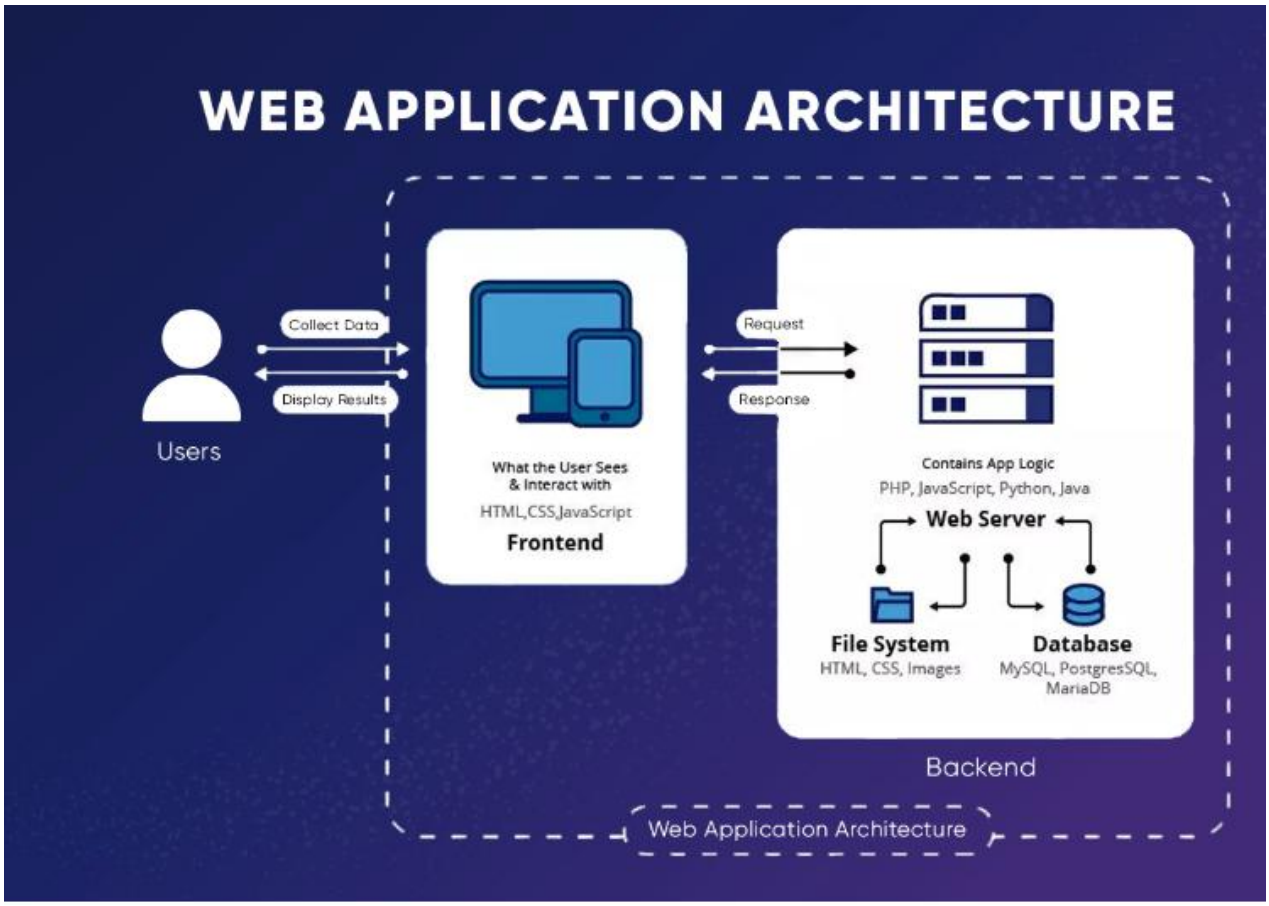
</jsp:forward>

jsp:include action tag:

Syntax:

<jsp:include page=”location of the page”/>

WEB APPLICATION ARCHITECTURE



CODING & IMPLEMENTATION

HTML CODE

doner_login.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Doner Login</title>
   <link rel="stylesheet" type="text/css" href="doner_login_style.css">
</head>
<body>
        <div class="main">
        <ul>
                <img
src="https://us.123rf.com/450wm/marfuah/marfuah2311/marfuah231102880/217391417-blood-care-logo-template-design-vector-emblem-design-concept-creative-symbol-icon.jpg?ver=6"
width="45px" height="55px"/>
                <li>Blood Bank System</li>
                </ul>
                <hr/>
        </div>
   <div class="container" >
      <form name="loginForm" onsubmit="return validateForm();" action="Doner_login.jsp"
method="POST">
                        <a href="home.html">HOME</a>
                        <h1>DONER Login</h1>
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" >

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" ><br/>

        <button type="submit">Login</button><br>
```

```
                    <p>Does not an account?</p><a href="doner_register.html">Click here to
register</a>
      </form>
   </div>
        <p>Copy  Rights  @ D.Lavanya Sri, 22761A0579.</p>
        <script src="doner_login.js"></script>
</body>
</html>
```

doner_register.html:

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <title>Doner Registration</title>
        <link rel="stylesheet" href="doner_register.css">
        <script src="doner_register.js"></script>
 </head>
 <body>

        <div class="high">
                <ul>
                <img
src="https://us.123rf.com/450wm/marfuah/marfuah2311/marfuah231102880/217391417-blood-
care-logo-template-design-vector-emblem-design-concept-creative-symbol-icon.jpg?ver=6"
width="45px" height="55px"/>

                <li>Blood Bank</li>
                <li><a href="home.html">HOME</a></li>
                </ul>
        </div>
        <div class="low">
         <img
src="https://t4.ftcdn.net/jpg/01/02/04/83/240_F_102048335_IKcEMB9ZaUONmXTTk3egcYTKi
K2V0GX8.jpg" width="35px" height="45px"/>Doner Registration
```

```
    </div>
    <div class="container" >

<form name="loginForm"  action="register.jsp" onsubmit="return validate();">
    <label>Name:</label><br>
    <input type="text" name="name"/><br>

    <label>Email:</label><br>
    <input type="email" name="email"/><br>

                    <label>Username:</label><br>
    <input type="text"  name="uname" /><br>

                    <label>Password:</label><br>
    <input type="password"  name="pwd" /><br>

                    <label>Age:</label><br>
    <input type="number"  name="age"/><br>

                    <label>Choose Gender:</label><br>
                    <input type="radio" name="gender" value="male">:Male
                    <input type="radio" name="gender" value="female">:Female<br>

                     <label>Phone Number:</label><br>
    <input type="number" name="phno" /><br>

                     <label>Address:</label><br>
    <input type="text"  name="adder" /><br>

                    <label>State:</label><br>
    <input type="text"  name="state" /><br>
```

```
                    <label>Choose Your Blood Group:</label><br>
                    <input type="radio" name="group" value="A+">:A+ve
                    <input type="radio" name="group" value="A-">:A-ve<br>
                    <input type="radio" name="group" value="B+">:B+ve
                    <input type="radio" name="group" value="B-">:B-ve <br>
                    <input type="radio" name="group" value="O+">:O+ve
                    <input type="radio" name="group" value="O-">:O -ve<br>
                    <input type="radio" name="group" value="AB+">:AB+ve
                    <input type="radio" name="group" value="AB-">:AB-ve<br>
        <button type="submit">Submit</button>
                    <button type="reset">Reset</button>
      </form>
   </div>

        <p>Copy  Rights  @  D.Lavanya Sri, 22761A0579.</p>
 </body>
</html>
```

CSS CODE

doner_login_style.css:

```
.container
        {
   width: 400px;
        height:390px;
   padding:20px;
        margin: 50px auto;
        background-color:white;
        max-width: 400px;
   border: 4px solid black;
        font-family: sans-serif;
```

```
}
body
{
        background-image:url("https://media.istockphoto.com/id/1320162065/vector/blood-cells-red-erythrocytes.jpg?s=612x612&w=0&k=20&c=rX8kGwiOsHW-V-BOal3Cf6ADKdjx_U5snDdykya0jbs=");
        width:100%;
        height:80%;
        background-size:cover;
}
 a
{
        text-decoration:none;
        font-size:18px;
        }
 h1 {
   text-align: center;
   margin-bottom: 20px;
        color:black;
}
 form {
   margin-bottom: 20px;
}

label {
   display: block;
   margin-bottom: 15px;
        color:black;
}
 input {
   width: 70%;
   padding: 5px;
```

```
        height:20px;
    border: 2px solid ;
        border-radius:10px 10px 10px 10px;
}

 button {
    background-color: blue;
    color: white;
    padding: 15px;
        margin-top:35px;
    cursor: pointer;
        font-size:20px;
        border-radius:20px 20px 20px 20px;
}
.container p{
        text-align:left;
        color:black;
        }
 img{
                padding-top:5px;
                padding-left:500px;
}
.main li{
        text-align:center;
        color:white;
        display:inline-block;
        font-size:35px;
        padding-bottom:5px;
        }
.main{
        width:100%;
```

```
        height:90px;


        }
p{
 text-align:center;
 color:white;
 }
```

doner_register.css:

```
.container {
    width: 700px;
        height:400px;
    border: 10px ;
        margin: 0 auto;
}
form {
    margin-bottom: 40px;
        background-color:Lavender;
        border: 1px solid ;
        padding:10px;


}
input {

    border: 2px solid ;
        border-color:green;
        margin-bottom:20px;
        margin-right:20px;
        height:20px;
        position:center;
        padding-left:50px;
        padding-right:100px;
```

```
        padding-bottom:10px;
        margin-left:60px;

}
label{
padding:20px 0px 10px 10px;
        margin-left:10px;
        }

button{
        width:90px;
        height:50px;
        margin-left:70px;
        margin-bottom:10px;
         border: 2px solid ;
        border-color:green;

        }
.low{
        background-color:red;
        width:700px;
        height:60px;
        font-family:courier;
        text-align:center;
        color:white;
        margin-left:354px;
        border: 2px solid ;
        border-color:black;
        font-size:50px;
        margin: 0 auto;
        }
```

```
.high{
                width:100%;
                height:70px;
                background-color:red;
                color:white;
                margin-bottom:15px;
                }

.high  ul li{
                color:white;
                font-size:25px;
                padding-bottom:5px;
                display:inline-block;
                padding-right:500px;
        }
img{
        padding-top:5px;
        padding-left:0px;
        padding-right:0px;
        }
a{
        display:inline-block;
        color:white;
        text-decoration:none;
}
 p{
        text-align:center;
        color:green;
        margin-top:540px;
}
```

doner_profile.css:

```
table.styled-table {
        width: 40%;
        margin: 0 auto; /* Center the table */
        border:1px;
        background-color: Lavender;
    }
    table.styled-table th, table.styled-table td {
        border: 1px solid #ddd;
        padding: 8px;
    }
    table.styled-table th {
        background-color: #4CAF50; /* Header background color */
        color: black;
    }
    p {
        text-align: center;
        color: green;
    }
```

donate_blodd.css:

```
form{
                    width:400px;
                    height:400px;
                    border:1px solid;
                    margin:0 auto;
            }
        .form-group {
            margin-bottom: 15px;
        }
```

```
.form-group label {
    display: inline-block;
    width: 150px; /* Adjust as needed */
    text-align: right;
    margin-right: 10px;
                padding-bottom:25px;
}
.form-group input {
    width: 200px; /* Adjust as needed */
}
.form-group input[type="submit"] {
    width: auto;
                height:30px;
                color:white;
                background-color:red;
    margin-left: 110px; /* Width of label + margin-right of label */
}
        h2{
                color:white;
                background-color:black;
                text-align:center;
                }
                p
                {
                text-align:center;
                color:green;
                }
```

reg_display1.css:

```
.table-container {
                display: flex;
                justify-content: center;
```

```
}

table {
   border:1px solid;
   width: 400px; /* Adjust table width as needed */
}

th, td {
   border: 1px solid #dddddd;
   text-align: left;
   padding: 8px;
}

th {
   background-color: Lavender;
}
            h2{
            text-align:center;
            }
            p{
text-align:center;
color:green;
margin-top:400px;
}
```

JAVASCRIPT CODE

doner_login.js:

```
var flag=0;
                    function validateForm()
                    {
```

```
var pass = document.loginForm.password.value;
var checkSpecial = /[*@!#%&]+/.test(pass);
var checkUpper = /[A-Z]+/.test(pass);
var checkLower = /[a-z]+/.test(pass);
var checkNum = /[0-9]+/.test(pass);
var user =document.loginForm.username.value;
var letters=/^[a-zA-Z0-9_\s]+$/;
if(pass=="" || pass=="null" && user=="" || user=="null")
{
        alert("Please Enter the username and password");
        return false;
}
if(pass=="" || pass=="null")
{
        alert("please enter ur password");
        document.loginForm.password.focus();
        return false;
        }
if(pass.length<8)
{
        alert("The password should be minimum 8 characters");
        document.loginForm.password.focus();
        return false;
}
```

```
if(checkSpecial &&checkLower && checkUpper &&checkNum)
{
	flag=1;
}

if(flag==0)
{
	alert("Password should be the combination of atleast one lowercase letter, uppercase letter,Number & special symbols like ($, & #)");
	document.loginForm.password.focus();
	return false;
}
if(user=="" || user=="null")
{
	alert("Plz enter your user name");
	document.loginForm.username.focus();
	return false;
}
if(user.length<6)
{
	alert("The User name should be minimum 6 characters");
	document.loginForm.username.focus();
	return false;
}
```

```
                    if(!user.match(letters))
                    {
                        alert("Plz enter valid user name");
                        document.loginForm.username.focus();
                        return false;
                    }
    }
```

doner_register.js:

```
var flag=0;
    function validate()
    {
        var name=document.loginForm.name.value;
        var alpha=/[A-Za-z]+/;
        var user=document.loginForm.uname.value;
        var letters=/^[a-zA-Z0-9_\s]+$/;

        var pass = document.loginForm.pwd.value;
        var checkSpecial = /[*@!#%&]+/.test(pass);
        var checkUpper = /[A-Z]+/.test(pass);
        var checkLower = /[a-z]+/.test(pass);
        var checkNum = /[0-9]+/.test(pass);

        var age=document.loginForm.age.value;
        var num=/[0-9]+/;
```

```
var add=document.loginForm.adder.value;
var alphas=/[A-Za-z]+/;

var sta=document.loginForm.state.value;
var alphas1=/[A-Za-z]+/;




if (!alpha.test(name))
{
        alert("Please enter a valid name containing only alphabetic characters.");
        return false;
}
if(!letters.test(user))
{
        alert("Please enter the username with alphbates and numbers");
        return false;
}
if(user.length<6)
{
        alert("The User name should be minimum 6 characters");
        document.loginForm.uname.focus();
        return false;
}
```

```
if(checkSpecial && checkLower && checkUpper && checkNum)
{
	flag=1;
}
if(flag==0)
{
	alert("Please enter the Password with combination of atleast one lowercase letter, uppercase letter,Number & special symbols like ($, & #)");
	document.loginForm.pwd.focus();
	return false;
}
if(pass.length<8)
{
	alert("The password should be minimum 8 characters");
	document.loginForm.pwd.focus();
	return false;
}

if (!num.test(age))
{
	alert("Please enter a valid age");
	return false;
}

if (!alphas.test(add))
```

```
		{
			alert("Please enter the Address which contain only alphabets.");
			return false;
		}
		if (!alphas1.test(sta))
		{
			alert("Please enter the State which contain only alphabets.");
			return false;
		}
	}
```

JSP CODE

Doner_login.jsp:

```
<%@ page import="java.sql.*" %>
<jsp:include page="doner_home.html" />
<%
	String un=request.getParameter("username");
	try
	{
		Class.forName("oracle.jdbc.driver.OracleDriver");
		Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","lavanya");
		String qry="select * from Doner_reg  where uname=?";
		PreparedStatement pstmt=con.prepareStatement(qry);
		pstmt.setString(1,un);
		ResultSet rs=pstmt.executeQuery();
```

```
                if (rs.next()) {
   session.setAttribute("user", un);
   out.println("<br><br><br><h2 style ='text-align:center;'>Welcome: " + un + "</h2>");
}
                else{
                        out.print("<h3 style='text-align:center; color:white;'>invalid user name & password</h2>");
%>
<jsp:forward page="doner_login.html" />
<%
}
                }

        catch(Exception e)
        {
                out.print(e);
        }
%>
<p>Copy  Rights  @ D.Lavanya Sri, 22761A0579.</p>
<style>
p{
text-align:center;
color:green;
margin-top:400px;
}
```

```
</style>
```

doner_profile.jsp:

```
<%@ page import="java.sql.*"%>
<jsp:include page="doner_home.html" />
<br>
<br>
<br>

<%
String uname = (String)session.getAttribute("user");
if(uname != null) {
   out.print("<br><br><div style='text-align:center;'>This is your PROFILE<br><br>Your are Mr/Ms. " + uname + "</div>");

   try {
      Class.forName("oracle.jdbc.driver.OracleDriver");
      Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "system", "lavanya");
      String query = "SELECT * FROM Doner_reg WHERE UNAME = ?";
      PreparedStatement pstmt = con.prepareStatement(query);
      pstmt.setString(1, uname);
      ResultSet rs = pstmt.executeQuery();

      if (rs.next()) {
         String name = rs.getString("NAME");
         String email = rs.getString("EMAIL");
```

```
            String pwd = rs.getString("PWD");
            int age = rs.getInt("AGE");
            String gen = rs.getString("GENDER");
            long phno = rs.getLong("PHNO");
            String adder = rs.getString("ADDRESS");
            String state = rs.getString("STATE");
            String group = rs.getString("B_GROUP");
%>
<html>
<head>
<link rel="stylesheet" href="doner_profile.css">
</head>
<table class="styled-table" border="1" cellspacing="0">
    <tr>
        <th>NAME</th>
        <td><%= name %></td>
    </tr>
    <tr>
        <th>EMAIL</th>
        <td><%= email %></td>
    </tr>
    <tr>
        <th>UNAME</th>
        <td><%= uname %></td>
    </tr>
```

```
<tr>
    <th>PWD</th>
    <td><%= pwd %></td>
</tr>
<tr>
    <th>AGE</th>
    <td><%= age %></td>
</tr>
<tr>
    <th>GENDER</th>
    <td><%= gen %></td>
</tr>
<tr>
    <th>PHNO</th>
    <td><%= phno %></td>
</tr>
<tr>
    <th>ADDRESS</th>
    <td><%= adder %></td>
</tr>
<tr>
    <th>STATE</th>
    <td><%= state %></td>
</tr>
<tr>
```

```
        <th>B_GROUP</th>
        <td><%= group %></td>
    </tr>
</table>
</html>
<%
        } else {
            out.println("Please login first");
        %>
<jsp:include page="Doner_login.html" />
<%
        }
    } catch(Exception e) {
        out.print(e);
    }
}
%>


<p>Copy Rights @ D.Lavanya Sri, 22761A0579.</p>
```

Donate_blood.jsp:

```
<jsp:include page="doner_home.html" />
<br>
<br>
<br>
<html>
```

```
<head>
    <title>Login Form</title>
    <link rel="stylesheet" href="donate_blood.css">


</head>
<body>
    <form action="blood_donate_reg.jsp" method="post">
            <h2>DONATE BLOOD</h2>
        <div class="form-group">
            <label>Blood Group:</label>
            <select name="options">
                                <option name="choose">Choose Blood Group</option>
                                <option name="A+VE">A+VE</option>
                                <option name="A-VE">A-VE</option>
                                <option name="B+VE">B+VE</option>
                                <option name="B-VE">B-VE</option>
                                <option name="O+VE">O+VE</option>
                                <option name="O-VE">O-VE</option>
                                <option name="AB+VE">AB+VE</option>
                                <option name="AB-VE">AB-VE</option>
                                </select>
                    </div>
        <div class="form-group">
            <label>Unit(in ml):</label>
            <input type="number"  name="unit">
```

```
        </div>
        <div class="form-group">
                        <label>Disease(if any):</label>
           <input type="text" name="dis"/>
        </div>
                <div class="form-group">
                        <label>Age:</label>
           <input type="number" name="age"/>
        </div>
                <div class="form-group">
           <input type="submit" value="DONATE">
        </div>
     </form>
          <p>Copy  Rights  @ D.Lavanya Sri, 22761A0579.</p>
</body>
</html>
```

reg_display1.jsp:

```
<%@ page import="java.sql.*" %>
<jsp:include page="Admin_home.html" />
<br>
<br>
<br>
<!DOCTYPE html>
<html>
<head>
```

```
    <title>Admin Home</title>
    <link rel="stylesheet" href="reg_display1.css">


</head>
<body>
        <h2>The Registered Detaills</h2>
    <div class="table-container">
                <br>
        <table>
            <tr>
                <th>NAME</th>
                <th>EMAIL</th>
                <th>UNAME</th>
                <th>PWD</th>
                <th>AGE</th>
                <th>GENDER</th>
                <th>PHNO</th>
                <th>ADDRESS</th>
                <th>STATE</th>
                <th>B_GROUP</th>
            </tr>
            <%
            try {
                Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
    Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "system", "lavanya");


    Statement stmt=con.createStatement();
        String qry="select * from Doner_reg";
        ResultSet rs=stmt.executeQuery(qry);



    while (rs.next()) {
%>
<tr>
    <td><%= rs.getString(1) %></td>
    <td><%= rs.getString(2) %></td>
    <td><%= rs.getString(3) %></td>
    <td><%= rs.getString(4) %></td>
    <td><%= rs.getInt(5) %></td>
    <td><%= rs.getString(6) %></td>
    <td><%= rs.getLong(7) %></td>
    <td><%= rs.getString(8) %></td>
    <td><%= rs.getString(9) %></td>
    <td><%= rs.getString(10) %></td>
</tr>
<%
    }
    rs.close();
```

```
            stmt.close();
            con.close();
        } catch (Exception e) {
            out.print(e);
        }
        %>
      </table>
   </div>
        <p>Copy  Rights  @ D.Lavanya Sri, 22761A0579.</p>
</body>
</html>
```

register.jsp:

```
<%@ page import="java.sql.*" %>

<%

        String name=request.getParameter("name");
        String email=request.getParameter("email");
        String uname=request.getParameter("uname");
        String pwd=request.getParameter("pwd");
        int age=Integer.parseInt(request.getParameter("age"));
        String gen=request.getParameter("gender");
        long phno=Long.parseLong(request.getParameter("phno"));
        String adder=request.getParameter("adder");
        String state=request.getParameter("state");
```

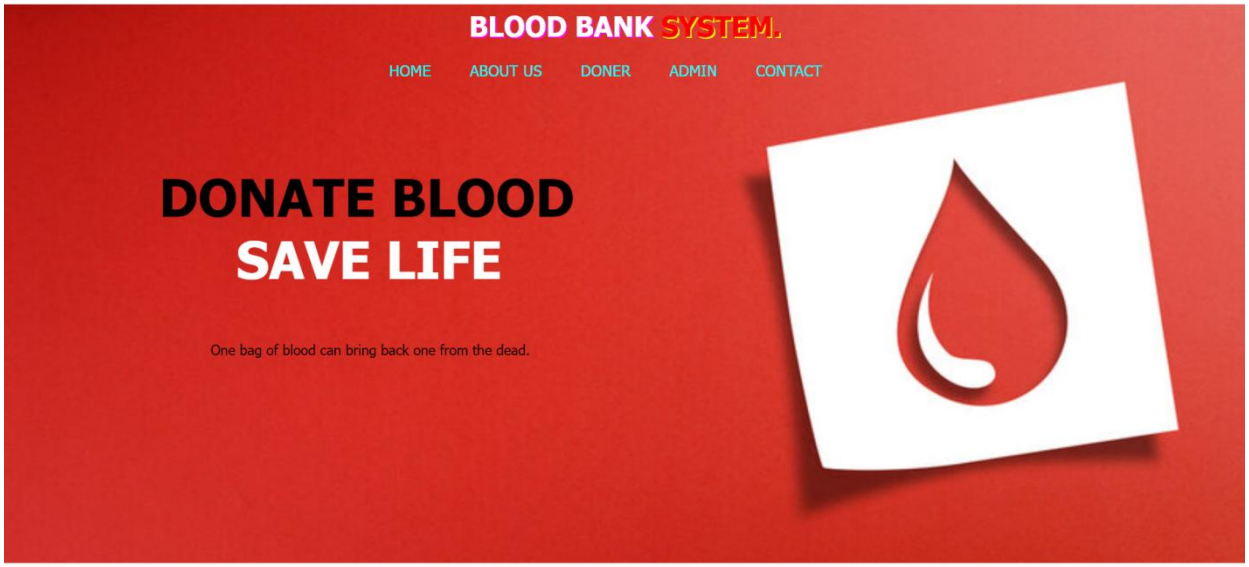
```
String group=request.getParameter("group");
try
{
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","lavanya");

        String qry = "insert into Doner_reg values(?,?,?,?,?,?,?,?,?,?)";
        PreparedStatement pst=con.prepareStatement(qry);
        pst.setString(1,name);
        pst.setString(2,email);
        pst.setString(3,uname);
        pst.setString(4,pwd);
        pst.setInt(5,age);
        pst.setString(6,gen);
        pst.setLong(7,phno);
        pst.setString(8,adder);
        pst.setString(9,state);
        pst.setString(10,group);
        pst.executeUpdate();
        out.print("You are successfully Registerd");
        out.print("<br>");
        out.print("<a href='doner_login.html'>LOGIN</a>");
        con.close();
}
```

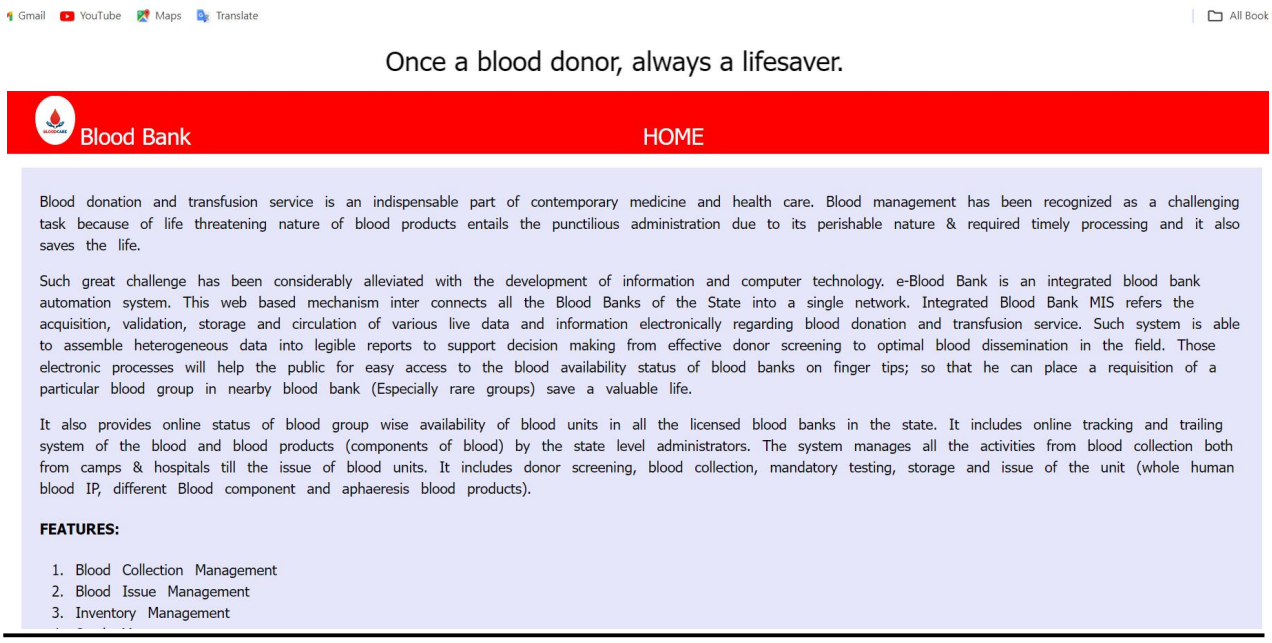
```
        catch(Exception e)
        {
                out.print(e);
        }
%>
<p>Copy  Rights  @ D.Lavanya Sri, 22761A0579.</p>
<style>
p{
text-align:center;
color:green;
margin-top:400px;
}
</style>
```

RESULTS

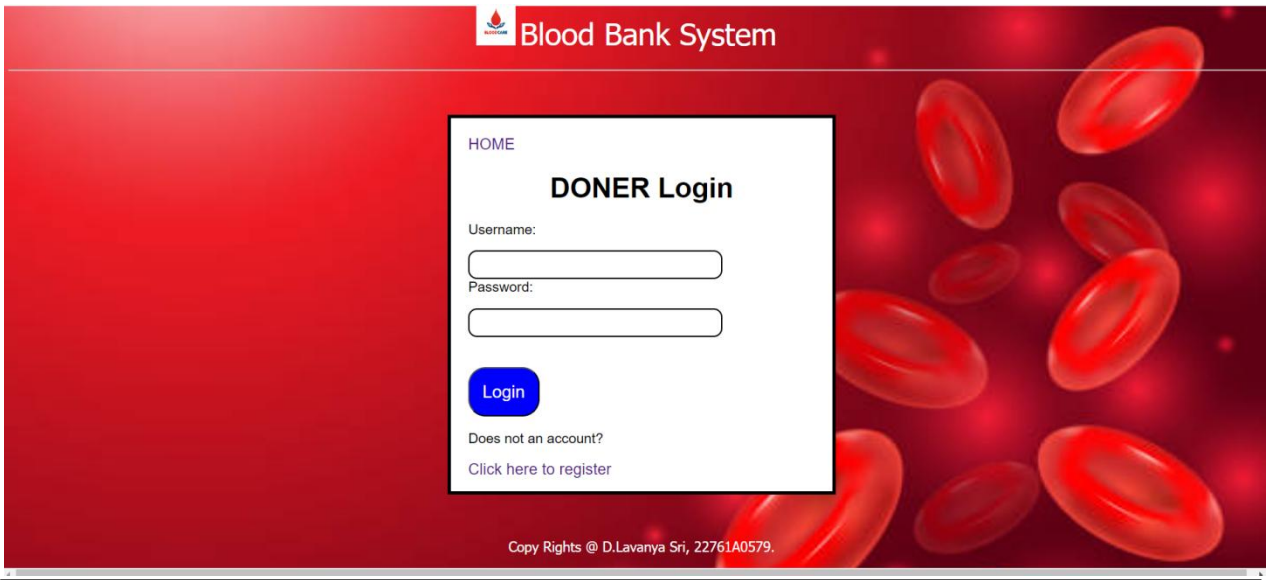
HOME:



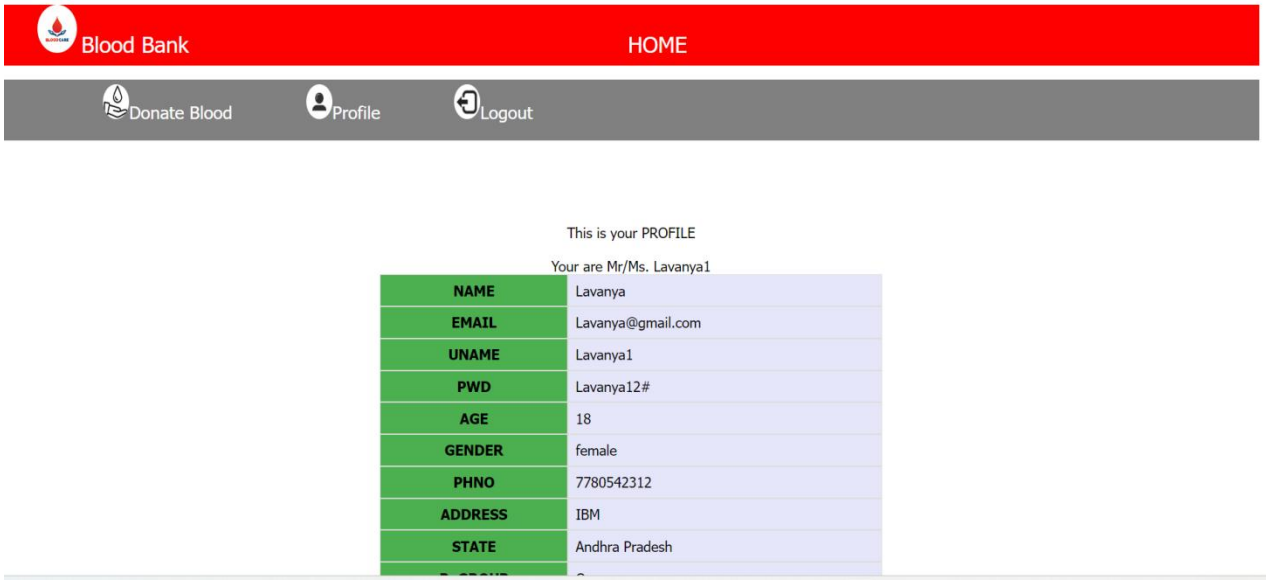
ABOUT_US:



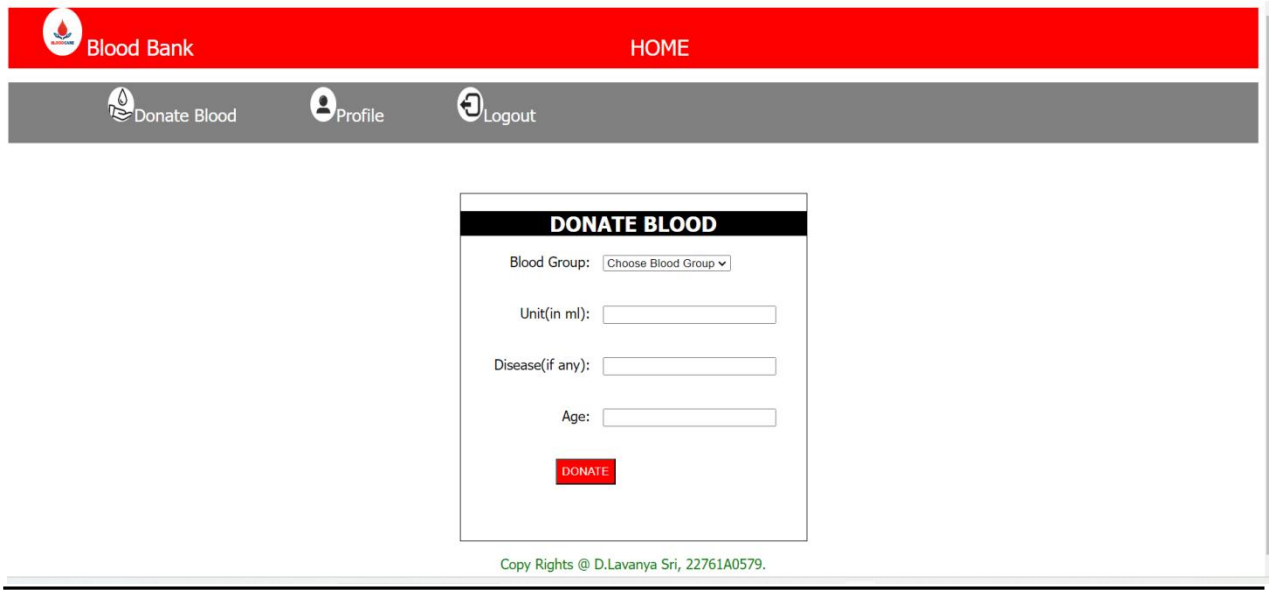
DONER_LOGIN:



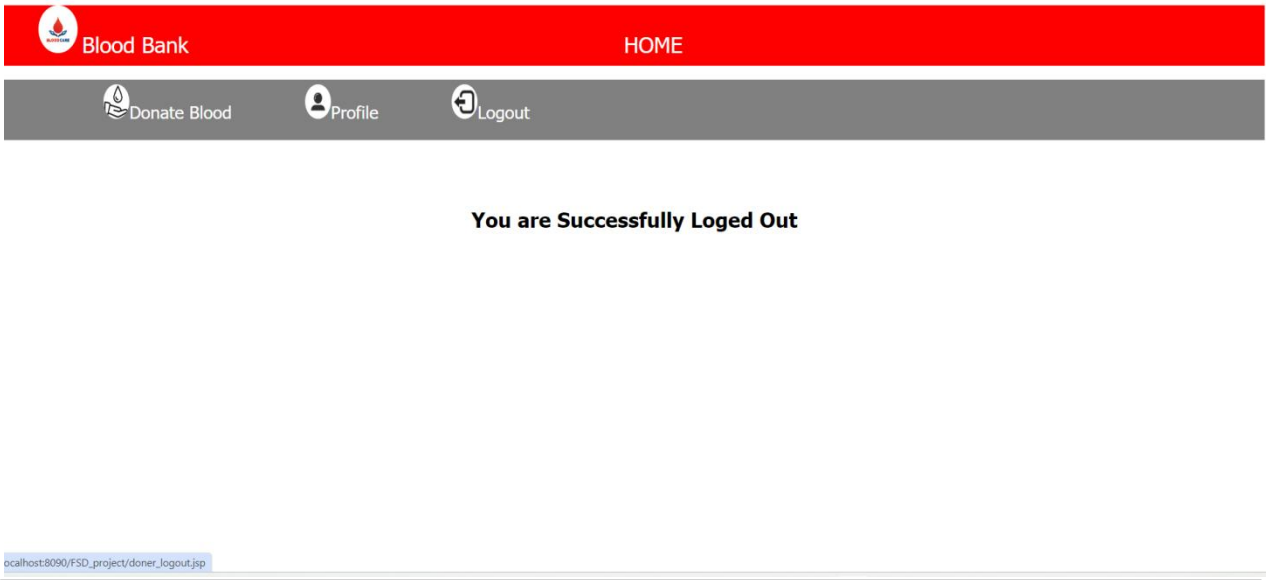
DONER_PROFILE:



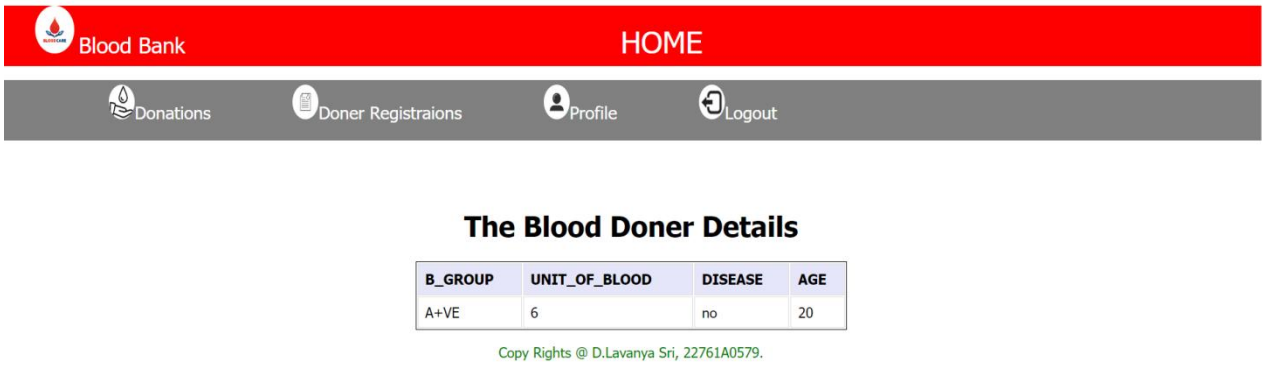
DONATE_BLOOD:



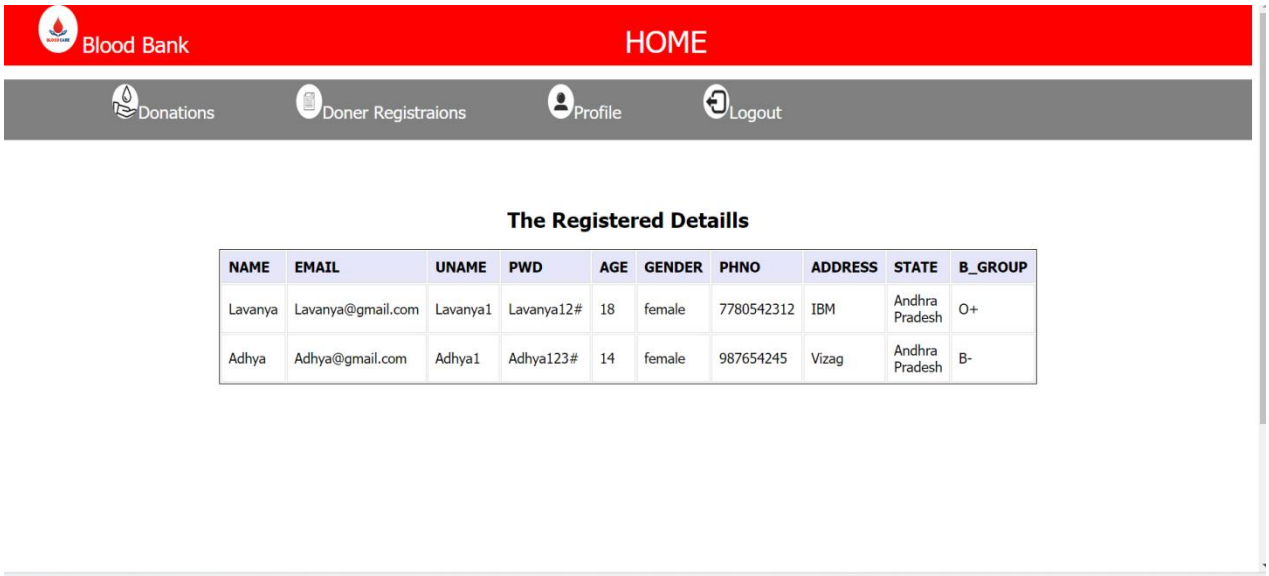
DONER_LOGOUT:



DONATIONS:



DONER_REGISTRATIONS:



CONTACT_US:

Blood Bank HOME

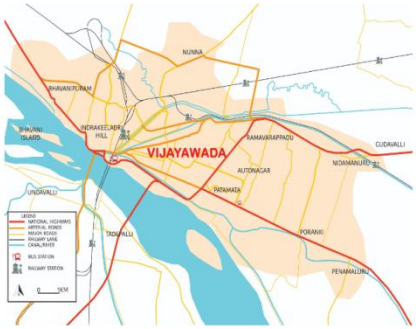
Contact Us
National Health Mission

Department of Health & Family Welfare Annex Building of SIHFW
Phone No. - 0674-2392480/88

.....

State Blood Transfusion Council, Andhra Pradesh

Health & family Welfare Department, Govt. Of AP GS Raju Street, Gandhi Nagar, Vijayawada - 520003 Call on 0866-2573669 Toll Free No:1800 425 1234



CONCLUSION

In conclusion, the implementation of a blood bank management system offers a transformative solution to enhance the efficiency, accuracy, and accessibility of blood inventory management. By leveraging technological advancements such as automated inventory tracking, real-time donor management, and predictive analytics, blood banks can optimize their operations and ensure a steady supply of blood products to meet the needs of patients. This not only streamlines the process of blood collection, storage, and distribution but also improves patient outcomes by minimizing delays in transfusion procedures and reducing the risk of blood shortages.

Furthermore, the integration of a blood bank management system facilitates better coordination between blood banks, hospitals, and other healthcare facilities. Through centralized data management and communication channels, stakeholders can easily collaborate, share information, and coordinate blood donations and transfusions more effectively. This interoperability not only enhances the overall resilience of the healthcare system but also fosters a sense of community engagement and support for blood donation initiatives. As a result, the blood bank management system serves as a vital tool in promoting public health and saving lives by ensuring the availability of blood products when and where they are needed most.

In summary, the adoption of a blood bank management system represents a significant advancement in healthcare technology that promises to revolutionize the way blood services are delivered and managed. By harnessing the power of data-driven insights and automation, blood banks can overcome logistical challenges, improve resource allocation, and ultimately save more lives. However, continuous innovation, stakeholder collaboration, and adherence to regulatory standards are essential to maximizing the benefits of this technology and addressing evolving needs in blood banking and transfusion medicine.

REFERENCES

1. https://www.redcrossblood.org/hosting-a-blood-drive/manage-my-blood-drive/blood-drive-coordinator-toolkit/drive-management-system-overview.html
2. https://www.eraktkosh.in/
3. https://blooddonor.jharkhand.gov.in/
4. http://www.netbloodbank.com/