Blood Drive Engine

PROJECT CLOSURE REPORT

SUBMITTED BY

Aatish Khengre	**PRN ID 220970920001**
Pranit Dhengale	**PRN ID 220970920029**
Rajat Patil	**PRN ID 220970920031**
RohitKumar Bilagude	**PRN ID 220970920033**

Under the guidance of
Dr. Ritu Khosla

In partial fulfilment of the requirements for the award of
POST GRADUATE DIPLOMA IN ADVANCED COMPUTING

Centre for Development of Advanced Computing
NIT Silchar campus, Silchar - 788010.
Assam (India)

CERTIFICATE

This is to certify that the project titled **BLOOD DRIVE ENGINE** is a record of the bonafide work done by the following students and submitted in partial fulfillment of the requirements for the award of **Post Graduate Diploma in Advanced Computing** by Center For Development Of Advanced Computing, Silchar, Assam, during the academic year 2022 - 2023.

Atish Khengre
220970920001

Pranit Dhengale
220970920029

Rajat Patil
220970920031

RohitKumar Bilagude
220970920033

Dr. Ritu Khosla
CDAC Silchar

Mr. Alok Dey
CDAC Silchar

Date: 13th March, 2023

ACKNOWLEDGMENTS

We take this opportunity to thank our guide and mentor Dr. Ritu Khosla. It is because of his support and guidance that we have been able to complete this project successfully.

A sincere word of thanks to Dr.Ritu Khosla who has monitored our work and has been a source of inspiration to us at all times.

A heartfelt gratitude to the entire faculty of CDAC without whom we would never have been able to learn new technologies and apply them into this project and complete it successfully.

Atish Khengre
220970920001

Pranit Dhengale
220970920029

Rajat Patil
220970920031

RohitKumar Bilagude
220970920033

Center For Development of Advanced Computing
Silchar, Assam

Date: 13th March, 2023

CONTENTS

ABSTRACT

Blood donation and transfusion has been an ever-serious issue and the shortage of blood throughout the world has caused many people to lose their life. The lack of a centralized system for blood donation is majorly responsible for those losses. Now in the era of online and digital processes, the conventional methods of collecting blood are absolute. An automated system is required to manage the centers and to showcase the information to the interested parties. We have developed a website that singlehandedly solves all these issues related to blood donation and reception. We have designed a SQLite database as an integral part of the integrated framework to store blood donation data in a centralized database for analytical processing. The proposed system would enable people to register as a donor to make themselves available whenever in need of their blood type

The main goal of the Blood Drive Engine project is to monitor Blood Bank data, Blood stock, Donor List. It manages the Donor, Blood stock data. The project is entirely administrative and therefore access is guaranteed only to the administrator. The project's aim is to develop an application system to minimize the manual work for Blood Bank, Donor, Blood Group management. It monitors all of the Blood Group information, Blood supply and Donor list.

Chapter 1
INTRODUCTION

1.1 Preamble

The Blood Drive Engine is to create an E-information about the donor and organization that are related to donating the blood. Through this any person who is interested in donating the blood can register himself in the same way if any organization wants to register itself with these sites can also register. Moreover if any general consumer wants to make request blood online he can also take the help of this site. Admin is the main authority who can do addition, decision and modification if required.

1.2 Project Description

The project is aimed to developing an online blood donation information. The entire project has been developed keeping in view of the distributed client serving computing technology in mind

The blood bank drive is to create an E-information about the donor and organization that are related to donating the blood. The project has been planned to be having the view of distributed architecture with centralized storage of database this storage of data has been planned using the SQL server and all the user interface has been designed using the JSP technology. The database connectivity is planned using the "SQL connection" methodology. The standards of security and data protective mechanism have been given a big choice for proper usage.

1.3 Overview

This Project is online web based project. Today you can easily connect with anything through Internet services. So online platform is the best choice for our project. Blood drive engine aims serving for human welfare. We have all the information you will need. Many people are here for you, to help you, willing to donate blood for you anytime. We have done all the job, rest is yours. Search the blood group you need. You can help us by registering on blood drive engine. If you are willing to donate your blood when needed. As a proud member of blood drive engine and a responsible human being you can help someone in need, So donate blood in online. Person who need to donate blood may register on our website with the help of user name and password.

Through this system it will be easier to find a donor for exact blood type and easy to build the connection between the donor and the blood bank authorities. The main intent of building this software is to formal the procedure of blood donation and motivate donors in order to donate the blood.

1.4 Project Goals and Scope

The goal is to get the blood for patient on time and easily. No need to check or contact with particular blood bank. The user will simply login and get information about availability of the blood within short time. User can get that information on the registered number through SMS. This software will improve the blood management system operation of the blood bank. The scope is to centralized repository of blood or particular blood bank. Basically the aim of software to manage the particular type of blood within the time and easily so there will not be any panic situation at that movement. Also our website receives useful data related to blood donation and reception in an integrated manner and helps in making intense situation better. In terms of coordination and fast reaction (that gets to be most imperative to spare the most extreme number of casual it is). Our website can also be used in camps as it asks for data to understand eligibility. Inner future work, we plan to explorer the generic mathematical model along with different donor groups using data mining tools and analytics. We are also focusing on implementing an application based on the website that will further help with connectivity to wider masses.

Chapter 2
BACKGROUND THEORY

2.1 Background and Motivation

As we know as we know the importance of human blood for human being on time. If the patient require blood in emergency and it is not available in that place then we should search it in different places of cities. This will take a lot of time. And possibility of unavailability of required blood. So the effort of searching in particular blood bank will become useless. And also wastage off time too
The motivation behind developing this web application is to check the availability of particular blood.

2.2 Literature Review

The current blood bank storage system is focused on files data and knowledge about the blood donors and recipients are stored in documents and archives. Data and information processing becomes difficult and time-consuming as a result of this. All tests of blood donation and transfusion are recorded on physical papers as well. This makes information helpless to blunders and human errors which in turn puts human lives in peril. Another underlying problem with this framework is destitute productivity. The sheer time-consuming method of recovering blood, be it donor or recipient information takes a lot of effort. The information retrieval being such a time-consuming process makes it very hard for hospitals to save lives at crucial times. Information Security & Information backup is another additional point to consider as the papers and records are effortlessly stolen or misplaced. This makes it an untrustworthy framework.

The goal behind our project has been to provide a platform that has all the information regarding blood donation, registered donors, which may in turn help in providing fast blood delivery. We have put our efforts into researching all about blood management systems and practices and have used the knowledge in making our project the best of what it could be. Every blood donation management system is required to accomplish some basic tasks. It has to have a mechanism for information exchange to be made available for donors, receptors, and other stakeholders. It must also ensure that the information regarding the blood inventory status of different stakeholders such as blood banks, hospitals are made available. It was important for us to find the faults in the existing system so that we can find the solutions to the flaws and implement them in our project.

2.3 Problem Statement

In present scenario searching for blood donors can take place through blood bank centers or by toll free numbers. So far it is a time taken process. Because it is having lots of manual work. It is waste of time to go to blood bank if the blood of particular group is not available and most of the time user has to wait in queue.

Chapter 3
METHODOLOGY

3.1 JAVA

Java is a **programming language** and a **platform**. Java is a high level, robust, object-oriented and secure programming language.Java was developed by *Sun Microsystems* (which is now the subsidiary of Oracle) in the year 1995. *James Gosling* is known as the father of Java. Before Java, its name was *Oak*. Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to Java.

Platform: Any hardware or software environment in which a program runs, is known as a platform. Since Java has a runtime environment (JRE) and API, it is called a platform.

3.2 JSP

Java Server Pages (JSP) is a technology that allows developers to create dynamic web pages using a combination of HTML, XML, and Java code. JSP pages are executed on a web server, and the resulting output is sent to the client's web browser. JSP provides a way to easily access Java code and objects from within a web page, simplifying the creation of dynamic web pages. JSP pages are typically used in conjunction with Java servlets, which handle data processing and client requests. JSP is part of the Java EE platform and is supported by most web servers and servlet containers.

3.3 JDBC

JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database. There are four types of JDBC drivers:

3.4 Node.JS

NodeJS is an open-source and cross-platform runtime environment built on Chrome's V8 JavaScript engine for executing JavaScript code outside of a browser. You need to recollect that NodeJS isn't a framework, and it's not a programing language. It provides an event-driven, non-blocking (asynchronous) I/O and cross-platform runtime environment for building highly scalable server-side applications using JavaScript.

3.5 HTML/CSS

HTML is an acronym which stands for **Hyper Text Markup Language** which is used for creating web pages and web applications. Let's see what is meant by Hypertext Markup Language, and Web page.

Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable.

CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page. It describes how a webpage should look.

3.6 MYSQL

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by **Oracle Company**.

Our MySQL tutorial includes all topics of MySQL database that provides for how to manage database and to manipulate data with the help of various SQL queries. These queries are: insert records, update records, delete records, select records, create tables, drop tables, etc. There are also given MySQL interview questions to help you better understand the MySQL database.

3.7 VS_CODE

VS Code will have deep remote development. You can connect to a container running a different OS and use any VS Code plugins, linting, debugging for that environment.

3.8 My SQL Library

MySQL Connectors provide connectivity to the MySQL server for client programs. APIs provide low-level access to MySQL resources using either the classic MySQL protocol or X Protocol.

3.9 TWILIO LIBRARY

Server-side helper libraries (or Server-side SDKs) make it easy for you to use Twilio's REST APIs, generate TwiML, and perform other common server-side programming tasks. These helper libraries are available in a variety of popular server-side programming languages.

Twilio's OpenAPI specification empowers you with a broad set of developer tooling, ranging from Postman collections to API mocking to automatic client generation in over 40 programming languages.

Twilio's JavaScript SDKs are used in the browser to create video conversations, make VoIP phone calls, or implement real-time omnichannel chat. Get started with the SDK you need.

This makes it much more convenient to use Twilio in your preferred programming language, without writing complex methods and functions. As an example, it is now possible to programmatically build a strongly-typed Rust library for Twilio’s API method

The various hardware and software configurations and requirements have been mentioned in detail in this section to provide an effective insight into the developmentfor this project.

3.10 External Interfaces

The external interfaces include the cloud data storage of Twilio Library APIs for dataacquisition, extraction and cleaning and certain requirement specific interfaces that may be implemented further upon updating the software or while tweaking the system

3.11 User Interfaces

GUI and Help Issues: The system will have user friendly and interactive screens and help will be available at all levels like site map or online help. List or Index of scripts will be provided for reducing data entry errors. The user has to select the correct parameters from the list wherever provided.

Security Issues
Users at different levels in the hierarchy will have restricted access to information. Onlythe System/Server Administrator will do administrative and maintenance work and update the information. Various security measures like password protection, etc. will also be provided.

3.12 Hardware Requirements Hardware requirements is as follows:

- Ram : 8GB
- Hard Drive : 512GB
- Processor : AMD RYZEN 5

3.13 Software Requirements Software requirements is as follows :

- Operating Systems : Windows 10 or Windows 11.
- Technology : JSP, JAVA , JDBC HTML,CSS
  IDE : Eclipse IDE

Chapter 4
CODE

Connectionprovider.jsp

```
package Project_pp;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;

public class ConnectionProvider {


public static Connection getCon() {


		try {

			Class.forName("com.mysql.jdbc.Driver");
			Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/bloodbank","root","8685");
			return con;


		} catch (Exception e) {
					System.out.println(e);
					return null;
		}
		}

}
```

addnewDonor.jsp

```
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="javax.naming.spi.DirStateFactory.Result"%>
<%@page import="Project_pp.ConnectionProvider"%>

<%@page import="java.sql.Connection"%>
<%@include file="header.html"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
 <link rel="stylesheet" href="style.css" type="text/css" media="screen">
<style>
input[type="text"], input[type="password"], input[type="email"], select,input[type="number"]
{
   border: none;
   background:silver;
   height: 50px;
   font-size: 16px;
            padding:15px;
            width:60%;
            border-radius: 25px;
            margin-left:20%;
}
h2,h1
{
            margin-left:20%;
}
hr
{
width:60%;
}
</style>
</head>
<body>



<%
            String msg= request.getParameter("msg");
             if("valid".equals(msg)){
                        %>
                        <center ><font color="red" size="5"></font> Successfully Updated</center>
                       <%}



             if("invalid".equals(msg)){
                        %>
                        <center ><font color="red" size="5"></font>Some thing went wrong ! Try again</center>
                       <%} %>
```

```
<%
int id=1;
try{
        Connection con= ConnectionProvider.getCon()


        Statement st=con.createStatement();
        //Connection con1=ConnectionProvider.getCon();

        ResultSet rs= st.executeQuery("select max(id) from donor");


        if(rs.next()){
                id=rs.getInt(1);
                id=id+1;

        }%>
        <div class="container" >


        <h1 style="color:red;">Donor ID:<% out.println(id);%> </h1>


        <%
}
catch(Exception e){

}


%>
<div class="container">
<form action="addNewDonorAction.jsp" method="post">
<input type="hidden" name="id" value="<% out.println(id); %>">

<h2>Name</h2>
<input type="text" placeholder="Enter Your Name" name="name">
<hr>
<h2>Father Name</h2>
<input type="text" placeholder="Enter Your Father Name" name="father">
<hr>
<h2> Mother Name</h2>
<input type="text" placeholder="Enter Your  Mother Name" name="mother">
<hr>
<h2>Mobile Number</h2>
<input type="Number" placeholder="Enter Mobile Number" name="mobileno">
<hr>
<h2>Gender</h2>

<select name="gender">
<option value="male">Male</option>
<option value="Female">Female</option>
<option value="Others">Others</option>
</select>
<hr>
<h2>Email</h2>
<input type="email" placeholder="Enter Your Email " name="email">
<hr>
<h2>BloodGroup</h2>
```

```
<select name="bloodgroup">
<option value="A+">A+</option>
<option value="A-">A-</option>
<option value="B+">B+</option>



<option value="O+">O+</option>
<option value="O-">O-</option>
<option value="AB+">AB+</option>
<option value="AB-">AB-</option>



</select>
<h2>Addresses</h2>
<input type="text" placeholder="Enter Your Addresses" name="addresses">
<br>
<center><button type="Submit" class="button">Save </button></center>

</form>



  </div>

<br>
<br>
<br>
<br>
<h3><center>Privacy Policy ::2023  </center></h3>

</body>
</html>
```

addNewDonorAction.jsp

```
<%@page import="java.sql.PreparedStatement"%>
<%@page import="Project_pp.ConnectionProvider"%>

<%@page import="java.sql.Connection"%>
<%

String id=request.getParameter("id");
String name=request.getParameter("name");
String father=request.getParameter("father");
String mother=request.getParameter("mother");
String mobileno=request.getParameter("mobileno");
String gender=request.getParameter("gender");
String email=request.getParameter("email");
String bloodgroup=request.getParameter("bloodgroup");
String addresses=request.getParameter("addresses");


try{
```

```
            Connection con=ConnectionProvider.getCon();
 PreparedStatement ps= con.prepareStatement("insert into donor  values (?,?,?,?,?,?,?,?,?)");
            ps.setString(1, id);
            ps.setString(2, name);
            ps.setString(3, father);




            ps.setString(4, mother);
            ps.setString(5, mobileno);
            ps.setString(6, gender);
            ps.setString(7, email);
            ps.setString(8, bloodgroup);
            ps.setString(9, addresses);
            ps.executeUpdate();
            response.sendRedirect("addNewDonor.jsp?msg=valid");

}
catch(Exception e){


            response.sendRedirect("addNewDonor.jsp?msg=invalid");

}

%>
```

Adminlogin.jsp

```
<!DOCTYPE html>
<html lang="en">
<head>
 <link rel="stylesheet" href="style.css" type="text/css" media="screen">
<style>
input[type="text"], input[type="password"], input[type="submit"]
{
   border: none;
   background:silver;
   height: 50px;
   font-size: 16px;
            margin-left:35%;
            padding:15px;
            width:33%;
            border-radius: 25px;
}
</style>
</head>
<body>
<div class="header">
  <a href="#default" class="logo"><img class="logo" src="index.jpg"></a>
  <div class="header-right">
    <a href="index.jsp">Home</a>
    <a class="active" href="adminLogin.jsp">Admin Login</a>
  </div>
</div>
<body>
<div class="container">
```

```
<br>
<br>
		<%
		String msg=request.getParameter("msg");
		 if("true".equals(msg)){
			%>
			<center ><font color="red" size="5"></font>Invalid Username/Password...</center>
			<%} %>




<form action="adminLoginAction.jsp" method="post">
<div class="form-group">
<center><h2> Admin Username</h2></center>
<input type="text" placeholder="Enter Your Username" name="Username" required>


<center><h2>Admin Password</h2></center>
<input type="text" placeholder="Enter Your Password" name="Password" required>
<center><button type="Submit" class="button">Submit</button></center>


</div>


</form>

</div>
<br>
<br>
<br>
<br>
<h3><center>Privacy Policy ::2023  </center></h3>
</body>
</html>
```

Adminloginaction.jsp

```
<%
String Username=request.getParameter("Username");
String Password=request.getParameter("Password");
if("admin".equals(Username)&& "admin".equals(Password)){
		response.sendRedirect("home.jsp?msg=valid");

}
else{
		response.sendRedirect("adminLogin.jsp?msg=invalid");

}
%>
```

deleteDonor.jsp

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

<%

String id= request.getParameter("id");




try{
		Connection con= ConnectionProvider.getCon();
		Statement st=con.createStatement();
		st.executeUpdate("delete from donor where id="+id+"");
		response.sendRedirect("editDeleteList.jsp?msg=deleted");

}catch(Exception e){
		response.sendRedirect("editDeleteList.jsp?msg=invalid");


}


%>
```

editdeleteList.jsp

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

<%@include file="header.html"%>

<html>
<head>
<style>
#customers {
  font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
  border-collapse: collapse;
  width: 95%;
}

#customers td, #customers th {
  border: 1px solid #ddd;
  padding: 8px;
}

#customers tr:nth-child(even){background-color: #f2f2f2;}

#customers tr:hover {background-color: #ddd;}
```

```
#customers th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #4CAF50;
  color: white;
}
</style>
</head>
<body>
<br>
<%



String msg=request.getParameter("msg");
if("valid".equals(msg))
{%>


<center><font color="red" size="5">Succesfully Updated</font></center>
<%
}
%>
<%

if("invalid".equals(msg))
{%>
<center><font color="red" size="5">Something Went Wrong Try Again</font></center>
<%
}
%>
<%

if("deleted".equals(msg))
{%>
<center><font color="red" size="5">Succesfully Deleted</font></center>
<%
}
%>
<center>
<table id="customers">
<tr>
<td>ID</td>
<td>Name</td>
<td>Father Name</td>
<td>Mother Name</td>
<td>Mobileno</td>
<td>Gender</td>
<td>Email</td>
<td>Bloodgroup</td>
<td>Addresses</td>
<td>Edit</td>
<td>Delete</td>


</tr>
<tr>
<%
try
{
```

```
        Connection con=ConnectionProvider.getCon();
        Statement st=con.createStatement();
        ResultSet rs=st.executeQuery("Select *from donor");
        while(rs.next())
        {
%>
<td><%=rs.getInt(1) %></td>
<td><%=rs.getString(2) %></td>
<td><%=rs.getString(3) %></td>
<td><%=rs.getString(4) %></td>
<td><%=rs.getString(5) %></td>
<td><%=rs.getString(6) %></td>
<td><%=rs.getString(7) %></td>
<td><%=rs.getString(8) %></td>
<td><%=rs.getString(9) %></td>




<td><a href="updateDonor.jsp?id=<%=rs.getString(1)%>">Edit</a></td>
<td><a href="deleteDonor.jsp?id=<%=rs.getString(1)%>">Delete</a></td>
</tr>
<%
        }
        }


catch(Exception e)
{
        System.out.println(e);
}
%>
</table>
</center>
<br>
<br>
<br>
<br>
<h3><center>Privacy Policy :: 2023</center></h3>
</body>
</html>
```

Header.html

```
<!DOCTYPE html>
<html>
<head>
 <link rel="stylesheet" href="style.css" type="text/css" media="screen">
<style>

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
```

```
}

li {
  float: left;
  border-right:1px solid #bbb;
}

li:last-child {
  border-right: none;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

li a:hover:not(.active) {
  background-color: #111;
}

.active {
  background-color: #4CAF50;
}
</style>
</head>
<body>




<ul>
  <li><a href="home.jsp">Home</a></li>
  <li><a href="addNewDonor.jsp">Add New Donor</a></li>
  <li><a href="editDeleteList.jsp">Edit,Delete & List of Donor Details</a></li>
    <li><a href="manageStock.jsp">Manage Stock</a></li>
  <li><a href="requestForBlood.jsp">Request for Blood</a></li>
  <li><a href="requestCompleted.jsp">Request Completed</a></li>
  <li style="float:right"><a href="adminLogin.jsp">Logout</a></li>
</ul>

</body>
</html>
```

Home.jsp

```
<%@include file="header.html"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<style>
img{
width:100%;
height:609px;
```

```
}</style>
</head>
<body>
<br>
<div style="max-width:100%">
  <img class="mySlides"  src="wel3.jpg" >
  <img class="mySlides"  src="admin33.jpg" >
   <img class="mySlides"  src="pop.jpg" >
</div>



<script>
var myIndex = 0;
carousel();

function carousel() {
    var i;
    var x = document.getElementsByClassName("mySlides");
    for (i = 0; i < x.length; i++) {
       x[i].style.display = "none";
    }
    myIndex++;
    if (myIndex > x.length) {myIndex = 1}
    x[myIndex-1].style.display = "block";
    setTimeout(carousel, 2000);
}
</script>
<br>
<h3><center>Privacy Policy :: 2023  </center></h3>
</body>
</html>
```

Index.jsp

```
<head>
 <link rel="stylesheet" href="style.css" type="text/css" media="screen">
<style>
.mySlides {display:none;}

input[type="text"], input[type="mail"]
{
    border: none;
    background:silver;
    height: 50px;
    font-size: 16px;
            margin-left:2%;
            padding:15px;
}
</style>
</head>
<body>

<div class="header">
  <a href="#default" class="logo"><img class="logo" src="index.jpg"></a>
  <div class="header-right">
    <a class="active" href="index.jsp">Home</a>
   <a href="loginaction.jsp">Login</a>
    <a href="adminLogin.jsp">Admin Login</a>
  </div>
```

```
  <center><font color="red" size="5"><h2>Blood Drive Engine</h2></font></center>
</div>

<div style="max-width:100%">
  <img class="mySlides"  src="slide12.jpg" >
  <img class="mySlides"  src="slide13.jpg" >
</div>
<script>
var myIndex = 0;
carousel();

function carousel() {
    var i;
    var x = document.getElementsByClassName("mySlides");
    for (i = 0; i < x.length; i++) {
       x[i].style.display = "none";
    }
    myIndex++;
    if (myIndex > x.length) {myIndex = 1}
    x[myIndex-1].style.display = "block";
    setTimeout(carousel, 3000); // Change image every 3 seconds
}
</script>




<body>
  <br>
  <%
  String msg=request.getParameter("msg");
  if("valid".equals(msg))
  {
            %>
            <center><font color="red" size="5">Form submitted sucessfully.</font></center>
            <%
  }


  %>
  <%
  String login=request.getParameter("login");
  if("true".equals(login))
  {
            %>
           <script >alert("Login Sucessfully") </script>
            <%
  }
  %>


  <br>
  <br>
  </div>
</div>

<div class="row1">
<div class="container">

    </tbody>
```

```
  </table>

</div>
</div>
<h3><center>Helpline No : 345664</center></h3>

</body>
</html>
```

Index1.jsp

```
>
 <link rel="stylesheet" href="style.css" type="text/css" media="screen">
<style>
.mySlides {display:none;}

input[type="text"], input[type="mail"]
{
    border: none;
    background:silver;
    height: 50px;
    font-size: 16px;
            margin-left:2%;
            padding:15px;



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

<div class="header">
  <a href="#default" class="logo"><img class="logo" src="index.jpg"></a>
  <div class="header-right">
    <a class="active" href="index.jsp">Home</a>
   <!--<a href="loginaction.jsp">Login</a>!-->
   <!-- <a href="adminLogin.jsp">Admin Login</a>!-->
  </div>
</div>




<div style="max-width:100%">
  <img class="mySlides"  src="slide44.jpg" >
  <img class="mySlides"  src="slide56.jpg" >
</div>

<script>
var myIndex = 0;
carousel();

function carousel() {
    var i;
    var x = document.getElementsByClassName("mySlides");
    for (i = 0; i < x.length; i++) {
```

```
    x[i].style.display = "none";
  }
  myIndex++;
  if (myIndex > x.length) {myIndex = 1}
  x[myIndex-1].style.display = "block";
  setTimeout(carousel, 3000); // Change image every 3 seconds
}
</script>


<body>
 <br>
 <%
 String msg=request.getParameter("msg");
 if("true".equals(msg))
 {
            %>
            <script >alert("Form Submitted Sucessfully") </script>
            <%
 }
 %>
 <%
 String login=request.getParameter("login");
 if("true".equals(login))
 {
            %>
           <script >alert("Login Sucessfully") </script>
            <%
 }
 %>




 <%

 if("invalid".equals(msg))
 {
            %>
            <center><font color="red" size="5">Invalid Data !! Try Again...</font></center>
            <%
 }
 %>
<center><h1>Enter Your Details To Request for Blood</h1></center>
<form action="indexFormAction.jsp" method="post">
<center>
<input type="text" name="name" placeholder="Enter Name" required>
<input type="text" name="mobileno" placeholder="Enter mobile Number" required>
<input type="mail" name="email" placeholder="Enter Email Address" required>
<input type="text" name="bloodgroup" placeholder="Enter Blood Group" required>




<button class="button1"><Span>Submit</Span></button>
</center>
</form>

 <br>
 <br>
 </div>
```

```
</div>

<div class="row1">
<div class="container">
<br>
<br>
   </tbody>
  </table>
</div>
</div>
<h3><center>Helpline No : 345664</center></h3>

</body>

</html>
```

indexFormAction.jsp

```
<%@page import= "Project_pp.ConnectionProvider"%>
<%@page import= "java.sql.*"%>
<%
String name=request.getParameter("name");
String mobileno=request.getParameter("mobileno");
String email=request.getParameter("email");
String bloodgroup=request.getParameter("bloodgroup");
String status = "pending";
String sms=request.getParameter("sms");
try{
        Connection con=ConnectionProvider.getCon();
        PreparedStatement ps=con.prepareStatement("insert into bloodrequest values(?,?,?,?,?,?)");
        ps.setString(1,name);
        ps.setString(2,mobileno);



        ps.setString(3,email);
        ps.setString(4,bloodgroup);
        ps.setString(5,status);
        ps.setString(6, sms);
        ps.executeUpdate();
        response.sendRedirect("index1.jsp?msg=true");

}
catch(Exception e)
{
        response.sendRedirect("login.jsp?msg=invalid");
}

%>
```

Login.jsp

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

```
 <link rel="stylesheet" href="style.css" type="text/css" media="screen">
<style>
input[type="text"], input[type="password"], input[type="submit"]
{
   border: none;
   background:silver;
   height: 50px;
   font-size: 16px;
            margin-left:35%;
            padding:15px;
            width:33%;
            border-radius: 25px;
}
</style>
</head>
<body>
<div class="header">
  <a href="#default" class="logo"><img class="logo" src="index.jpg"></a>
  <div class="header-right">
    <a href="index.jsp">Home</a>
     <!--  <a href="login.jsp" >Login</a>-->
    <!--   <a class="active" href="adminLogin.jsp">Admin Login</a>-->
  </div>
</div>
<body>
<div class="container">
<br>
<br>


                    <%
 String Register=request.getParameter("Register");
 if("true".equals(Register))
 {
             %>
            <script >alert("Register Sucessfully") </script>
             <%
 }
 %>


<form action="loginaction.jsp" method="post">
<div class="form-group">
<center><h2>User Id</h2></center>
<input type="text" placeholder="Enter Your User Id" name="UserId" required>


<center><h2>Password</h2></center>
<input type="text" placeholder="Enter Your Password" name="Password" required>
<center><button type="Submit" class="button">Login</button></center>
 <center> <a href="regir.jsp" color="blue">Create New User ?</a></center>
</div>

</form>
</div>




<br>
<br>
<br>
<br>
```

```
<h3><center>Privacy Policy ::2023  </center></h3>
</body>
</html>
```

loginAction.jsp

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

<%
String UserId=request.getParameter("UserId");
String Password=request.getParameter("Password");


try
{
        System.out.println("inside loginaction.jsp"+UserId+" "+Password);
        Connection con= ConnectionProvider.getCon();
        boolean isSucess=false;
        Statement st= con.createStatement();
        ResultSet rs= st.executeQuery("select * from registerdata where ID='"+UserId+"' and
password='"+Password+"'");

        while(rs.next())
        {
                isSucess=true;
        }
        if(isSucess){

                response.sendRedirect("index1.jsp?msg=valid&login=true");

        }

        else{
                response.sendRedirect("login.jsp?msg=invalid&login=false");

        }
}
catch(Exception e){
        System.out.println(e);
        e.printStackTrace();
}
%>
```

manageStock.jsp

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


<%@include file="header.html"%>



<html>
<head>
<link rel="stylesheet" href="style.css" type="text/css" media="screen">
<style>
input[type="text"], input[type="password"], input[type="submit"],select
{
    border: none;
    background:silver;
    height: 50px;
    font-size: 16px;
            margin-left:35%;
            padding:15px;
            width:33%;
            border-radius: 25px;
}

#customers {
  font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
  border-collapse: collapse;
  width: 55%;
}

#customers td, #customers th {
  border: 1px solid #ddd;
  padding: 8px;
}




#customers tr:nth-child(even){background-color: #f2f2f2;}

#customers tr:hover {background-color: #ddd;}

#customers th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #4CAF50;
  color: white;
}
</style>
</head>
<body>
<div class="container">
<br>
```

```
<%
String msg= request.getParameter("msg");
if("invalid".equals(msg))
{
%>
<center> <font color="red" size="5">Some thing went wrong!! try again..</font></center>

<%} %>


<%

if("valid".equals(msg))
{
%>
<center> <font color="red" size="5">Data Updated sucessfully</font></center>

<%} %>


<form action="manageStockAction.jsp" method="post">
<div class="form-group">
<center><h2>Select Blood Group</h2></center>
<select name="bloodgroup">
<option value="A+">A+</option>
<option value="A-">A-</option>
<option value="B+">B+</option>
<option value="B-">B-</option>
<option value="O+">O+</option>
<option value="o-">O-</option>
<option value="AB+">AB+</option>
<option value="AB-">AB-</option>
</select>



<center><h2>Select Increase/Decrease Blood Group</h2></center>
<select name="incdec">
<option value="inc">Increase</option>
<option value="dec">Decrease</option>
</select>









<center><h2>Units</h2></center>
<input type="text" placeholder="Enter Units" name="units">
<center><button type="submit"  class="button">Save</button></center>
</div>
</form>
<br>
<center>
<table id="customers">
<tr>
<th>Blood Group</th>
<th>Units</th>
</tr>
<tr>
```

```
<%
try{
Connection con = ConnectionProvider.getCon();

Statement st= con.createStatement();
ResultSet rs= st.executeQuery("select * from stock ");
while(rs.next())
{
%>
<td> <%=rs.getString(1) %></td>
<td> <%=rs.getString(2) %></td>
</tr>




<%
}
}
catch(Exception e)
{
        System.out.println(e);

        }
%>



</table>
</center>
<br>
<br>
<br>
<br>
<h3><center>Privacy Policy :: 2023 </center></h3>
</body>
</html>
```

manageStockAction.jsp

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

<%



String bloodgroup=request.getParameter("bloodgroup");
String incdec=request.getParameter("incdec");
String units = request.getParameter("units");
int units1=Integer.parseInt(units);
try
{

Connection con= ConnectionProvider.getCon();
```

```
Statement st= con.createStatement();
if(incdec.equals("inc"))

st.executeUpdate("update stock set units=units+'"+units1   +"'   where bloodgroup='"+bloodgroup+"'");
else
        st.executeUpdate("update stock set units=units-'"+units1   +"'   where bloodgroup='"+bloodgroup+"'");
response.sendRedirect("manageStock.jsp?msg.valid");

}
catch(Exception e)




{

        System.out.println(e);
        response.sendRedirect("manageStock.jsp?msg.invalid");

}

%>
```

Regir.jsp

```
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="Project_pp.ConnectionProvider"%>
<%@page import="java.sql.Connection"%>



<!DOCTYPE html>
<html lang="en">
<head>
 <link rel="stylesheet" href="style.css" type="text/css" media="screen">
<style>
input[type="text"], input[type="password"],select , input[type="submit"]
{
   border: none;
   background:silver;
   height: 50px;
   font-size: 16px;
        margin-left:35%;
        padding:15px;
        width:33%;
        border-radius: 25px;
}

</style>
</head>
<body>
<div class="container">

<div class="header">
  <a href="#default" class="logo"><img class="logo" src="index.jpg"></a>
  <div class="header-right">
```

```
   <!--  <a class="active" href="index.jsp">Home</a>-->
  <!-- <a href="loginaction.jsp">Login</a> -->
    <a href="adminLogin.jsp">Admin Login</a>
  </div>
</div>

<br>
<br>




            <%
            String msg=request.getParameter("msg");
             if("invalid".equals(msg)){
                        %>
                        <center ><font color="blue" size="5"></font>Invalid Details...</center>
                       <%} %>
            <%

             if("valid".equals(msg)){
                        %>

                        <center ><font color="red" size="5"></font>Successfully Registered...</center>
                       <%} %>



<%
int id=1;
try{
            Connection con= ConnectionProvider.getCon();

            Statement st=con.createStatement();

            ResultSet rs=st.executeQuery("select max(id) from registerdata");

            if(rs.next()){
                        id=rs.getInt(1);
                        id=id+1;
            }%>
            <div class="con" >


            <h2 style="color:red;">Form No :<% out.println(id);%> </h2>




            <%
}
catch(Exception e){

}


%>


<form action="regisaction.jsp" method="post">
```

```
 <center> <h1>* Request Registeration Form *</h1> </center>

<center><h3><font color="blue">Name</font></h3></center>
<input type="text" name="name" placeholder= "Enter Your Name" size="15" required />


<center><h3><font color="blue">Mobile Number</font></h3></center>
<input type="text" name="mobileno" placeholder="Enter Your Mobile Number" size="10" required>

<center><h3><font color="blue">UserId</font></h3></center>
<input type="text" name="Id" placeholder="Enter Your Id" size="15"required />

<center><h3><font color="blue">Password</font></h3></center>
  <input type="password" placeholder="Enter Your Password" name="password" required>

<center><h3><font color="blue">Select Gender</font></h3></center>
 <select name="gender">
<option value="male">Male</option>
<option value="Female">Female</option>
<option value="Others">Others</option>
</select>
<br>
<br>
 <center><button type="submit" class="registerbtn">Register</button></center>


  </div>

</div>
</form>
</body>
</html>
```

regisAction.jsp

```
<%@page import="java.sql.PreparedStatement"%>
<%@page import="Project_pp.ConnectionProvider"%>

<%@page import="java.sql.Connection"%>


<%


String name=request.getParameter("name");
String mobileno=request.getParameter("mobileno");
String Id=request.getParameter("Id");
String gender=request.getParameter("gender");
String password=request.getParameter("password");




try{
```

```
        Connection con=ConnectionProvider.getCon();
 PreparedStatement ps= con.prepareStatement("insert into registerdata values (?,?,?,?,?)");



        ps.setString(1, name);
        ps.setString(2, mobileno);
        ps.setString(3, Id);
        ps.setString(4, gender);
        ps.setString(5, password);

        ps.executeUpdate();
        response.sendRedirect("login.jsp?msg=valid&&Register=true");

}
catch(Exception e){

        System.out.println(e);
        response.sendRedirect("regir.jsp?msg=invalid");

}

%>
```

requestCompleated.jsp

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


<%@include file="header.html"%>
<html>
<head>
<style>
#customers {
  font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
  border-collapse: collapse;
  width: 95%;
}

#customers td, #customers th {
  border: 1px solid #ddd;
  padding: 8px;
}



#customers tr:nth-child(even){background-color: #f2f2f2;}

#customers tr:hover {background-color: #ddd;}
```

```
#customers th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #4CAF50;
  color: white;
}


</style>
</head>
<body>
<br>
<center>


<table id="customers">
<tr>
<th>Name</th>
<th>mobilenumber</th>
<th>Email</th>
<th>Blood Group</th>
</tr>
<tr>
<%
try
{
        Connection con= ConnectionProvider.getCon();
        Statement st= con.createStatement();
        ResultSet rs= st.executeQuery("select * from bloodrequest where status='completed'");
        while(rs.next())
        {

%>
<td> <%=rs.getString(1) %> </td>
<td> <%=rs.getString(2) %> </td>
<td> <%=rs.getString(3) %> </td>
<td> <%=rs.getString(4) %> </td>
</tr>
<%
        }
}
catch(Exception e){
        System.out.println(e);
}
%>

</table>

</center>
<br>
<br>
<br>
<br>
<h3><center>Privacy Policy:: 2023 </center></h3>
</body>
</html>
```

requestForBlood.jsp

```
<%@page import="Project_pp.ConnectionProvider" %>
<%@page import="java.sql.*" %>
<%@include file="header.html"%>
<html>
<head>
<style>



#customers {
  font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
  border-collapse: collapse;
  width: 95%;
}

#customers td, #customers th {
  border: 1px solid #ddd;
  padding: 8px;
}

#customers tr:nth-child(even){background-color: #f2f2f2;}

#customers tr:hover {background-color: #ddd;}

#customers th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #4CAF50;
  color: white;
}
</style>
</head>
<body>
<br>
<center>
<table id="customers">
<tr>
<th>Name</th>
<th>Mobile Number</th>
<th>Email</th>
<th>Blood Group</th>
<th>Done</th>
<th>Delete</th>
</tr>
<tr>
<%
try
{
          Connection con=ConnectionProvider.getCon();
          Statement st=con.createStatement();
          ResultSet rs=st.executeQuery("select * from bloodrequest where status='pending'");
          while(rs.next())
          {
%>
<td><%=rs.getString(1) %></td>
<td><%=rs.getString(2) %></td>
<td><%=rs.getString(3) %></td>
<td><%=rs.getString(4) %></td>
```

```
<td><a href="requestForBloodDone.jsp?mobileno=<%=rs.getString(2)%>">Done</a></td>
<td><a href="requestForBloodDelete.jsp?mobileno=<%=rs.getString(2)%>">Delete</a></td>
</tr>
<%



        }
        }
catch(Exception e)



{
        System.out.println(e);
        }
%>
</table>

</center>
<br>
<br>
<br>
<br>
<h3><center>All Right Reserved :: 2023</center></h3>
</body>
</html>
```

requestBloodDelete.jsp

```
<%@page import="Project_pp.ConnectionProvider" %>
<%@page import="java.sql.*" %>
<%
String mobileno=request.getParameter("mobileno");
try
{
        Connection con=ConnectionProvider.getCon();
        Statement st=con.createStatement();
        st.executeUpdate("delete from bloodrequest where mobileno='"+mobileno+"'");
        response.sendRedirect("requestForBlood.jsp");
        }
catch(Exception e)
{
        response.sendRedirect("requestForBlood.jsp");
        }
%>
```

requestBloodDone.jsp

```
<%@page import="Project_pp.ConnectionProvider" %>
<%@page import="java.sql.*" %>
<%
String mobileno=request.getParameter("mobileno");
try
{



		Connection con=ConnectionProvider.getCon();
		PreparedStatement ps=con.prepareStatement("update bloodrequest set status='completed' where
mobileno=?");
		ps.setString(1,mobileno);
		ps.executeUpdate();
		response.sendRedirect("requestForBlood.jsp");



		}
catch(Exception e)
{
		response.sendRedirect("requestForBlood.jsp");
		}
%>
```

updateDonor.jsp

```
<%@ page import="Project_pp.ConnectionProvider" %>
<%@page import="java.sql.*" %>
<%@include file="header.html"%>


<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
 <link rel="stylesheet" href="style.css" type="text/css" media="screen">
<style>
input[type="text"], input[type="password"], input[type="email"], select,input[type="number"]
{
   border: none;
   background:silver;
   height: 50px;
   font-size: 16px;
		padding:15px;
		width:60%;
		border-radius: 25px;
		margin-left:20%;
}
h2,h1
{
		margin-left:20%;
}
```

```
hr
{
width:60%;
}
</style>
</head>
<body>
<%
String id=request.getParameter("id");
try
{
	Connection con=ConnectionProvider.getCon();
	Statement st=con.createStatement();
	ResultSet rs=st.executeQuery("select *from donor where id='"+id+"'");
	while(rs.next())
	{
%>



<div class="container">
<form action="updateDonorAction.jsp" method="post">
<input type="hidden" name="id" value="<%out.println(id);%>">
<h2>Name</h2>
<input type="text" value="<%=rs.getString(2)%>" name="name">
<hr>
<h2>Father Name</h2>
<input type="text" value="<%=rs.getString(3)%>" name="father">
<hr>
<h2>Mother Name</h2>
<input type="text" value="<%=rs.getString(4)%>" name="mother">
<hr>
<h2>Mobile Number</h2>
<input type="number" value="<%=rs.getString(5)%>" name="mobilenumber">
<hr>
<h2>Email</h2>
<input type="email" value="<%=rs.getString(7)%>" name="email">
<hr>
<h2>Addresses</h2>
<input type="text" value="<%=rs.getString(9)%>" name="addresses">
<br>
<center><button type="submit" class="button">Save</button></center>
</form>
</div>
<%
	}
	}
catch(Exception e)
{
	System.out.println(e);
}
%>


<br>
<br>
<br>
<br>
<h3><center>Privacy Policy :: 2023  </center></h3>

</body>
```

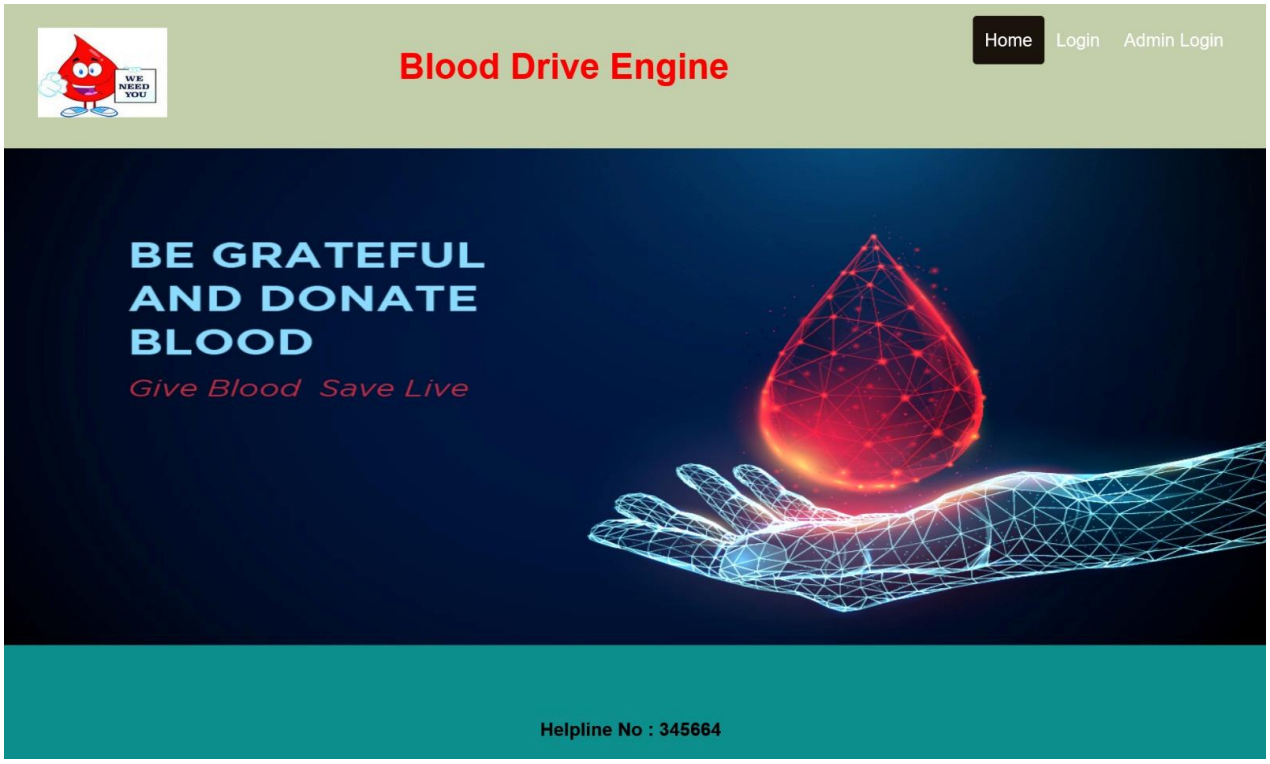
```
</html>
```

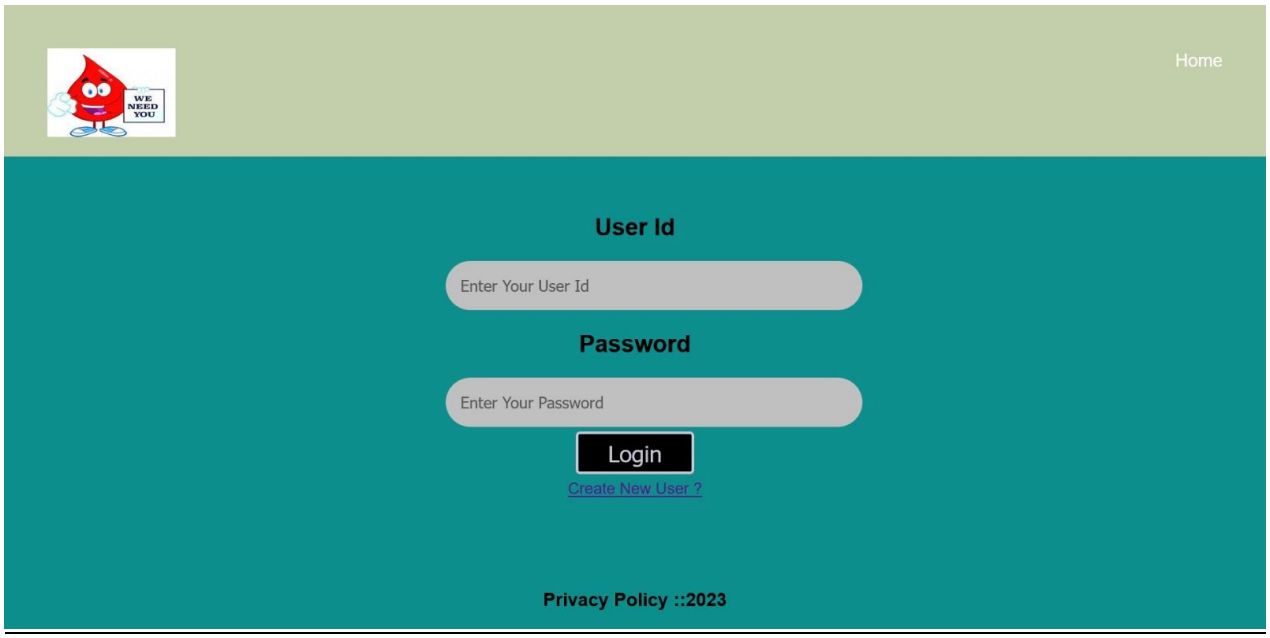
updateDonorAction.jsp

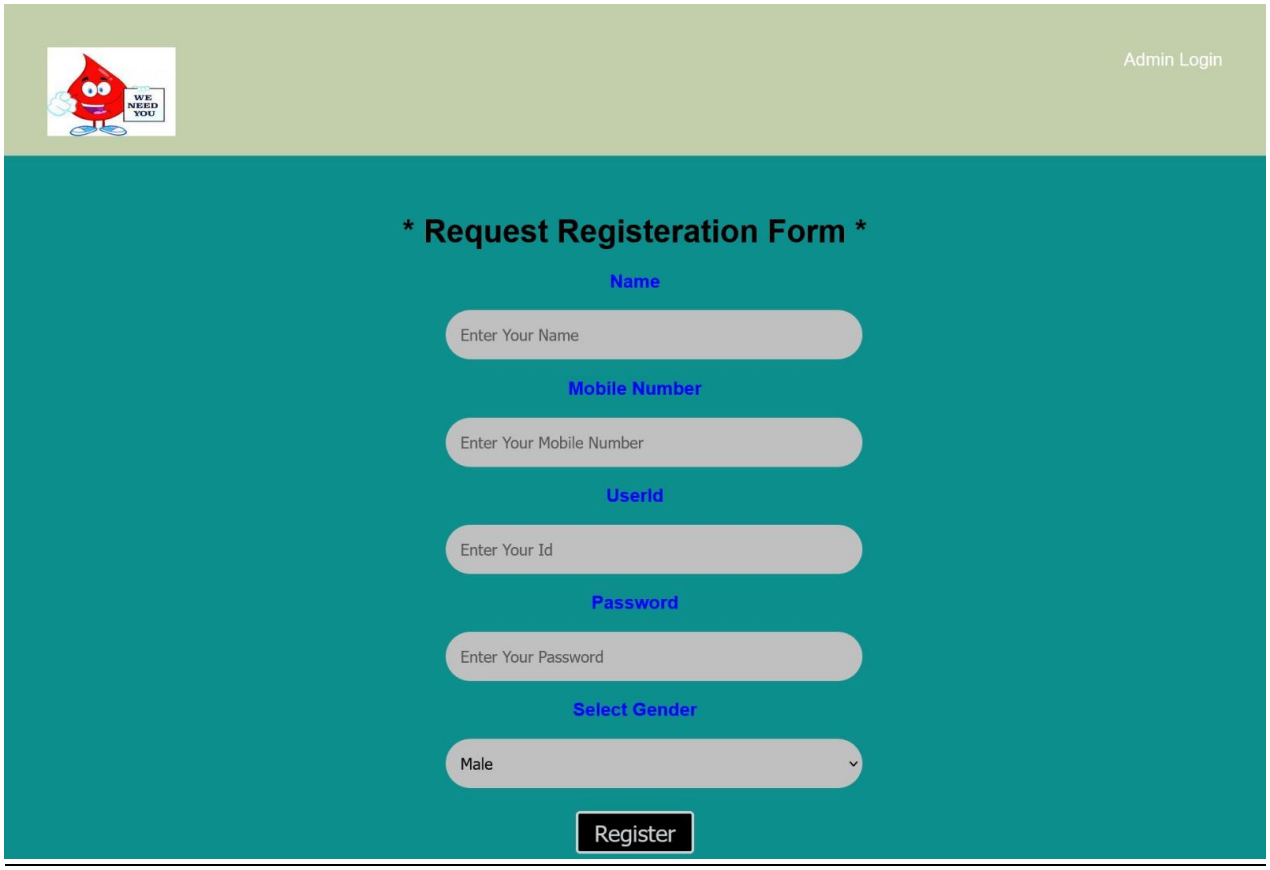
```
<%@ page import="Project_pp.ConnectionProvider" %>
<%@page import="java.sql.*" %>

<%

String id= request.getParameter("id");
String name= request.getParameter("name");
String father= request.getParameter("father");
String mother= request.getParameter("mother");
String mobileno= request.getParameter("mobileno");
String email= request.getParameter("email");
String addresses= request.getParameter("addresses");



try{

	Connection con=ConnectionProvider.getCon();
	PreparedStatement ps=con.prepareStatement("update donor set
name=?,father=?,mother=?,mobileno=?,email=?,addresses=? where id=?");
	ps.setString(1, name);
	ps.setString(2, father);
	ps.setString(3, mother);
	ps.setString(4, mobileno);
	ps.setString(5, email);
	ps.setString(6, addresses);
	ps.setString(7, id);
	ps.executeUpdate();
	response.sendRedirect("editDeleteList.jsp?msg=valid");



}catch(Exception e){
	System.out.println(e);
	response.sendRedirect("editDeleteList.jsp?msg=invalid");
}

%>
```

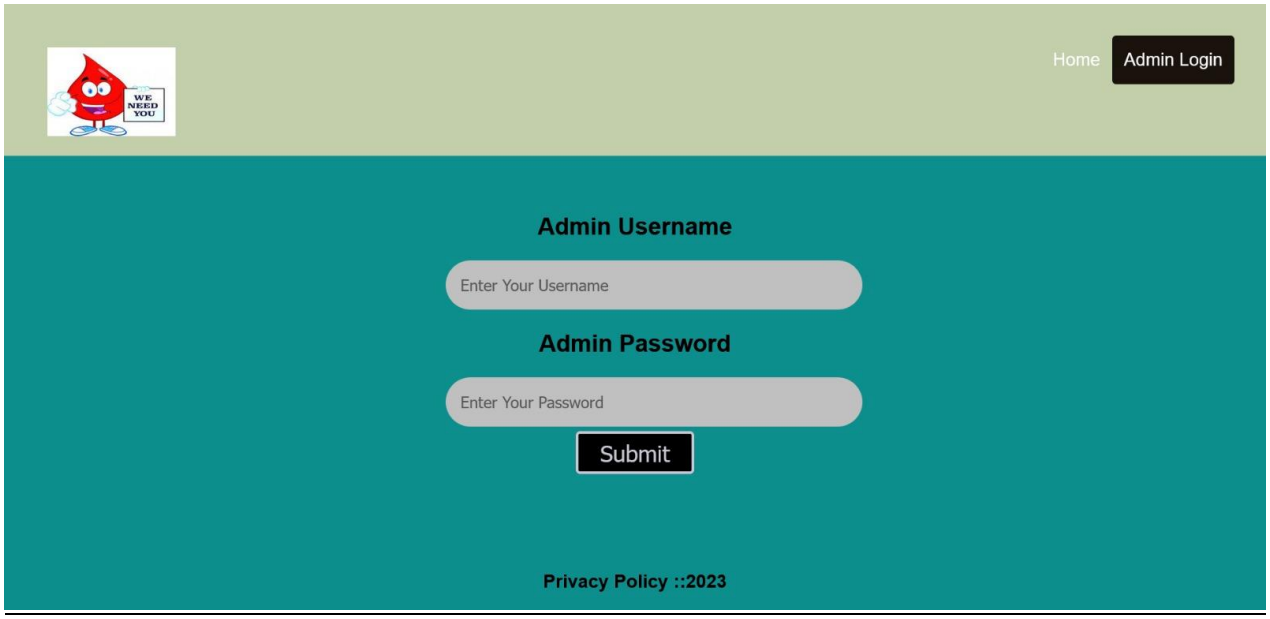
Useregister.jsp

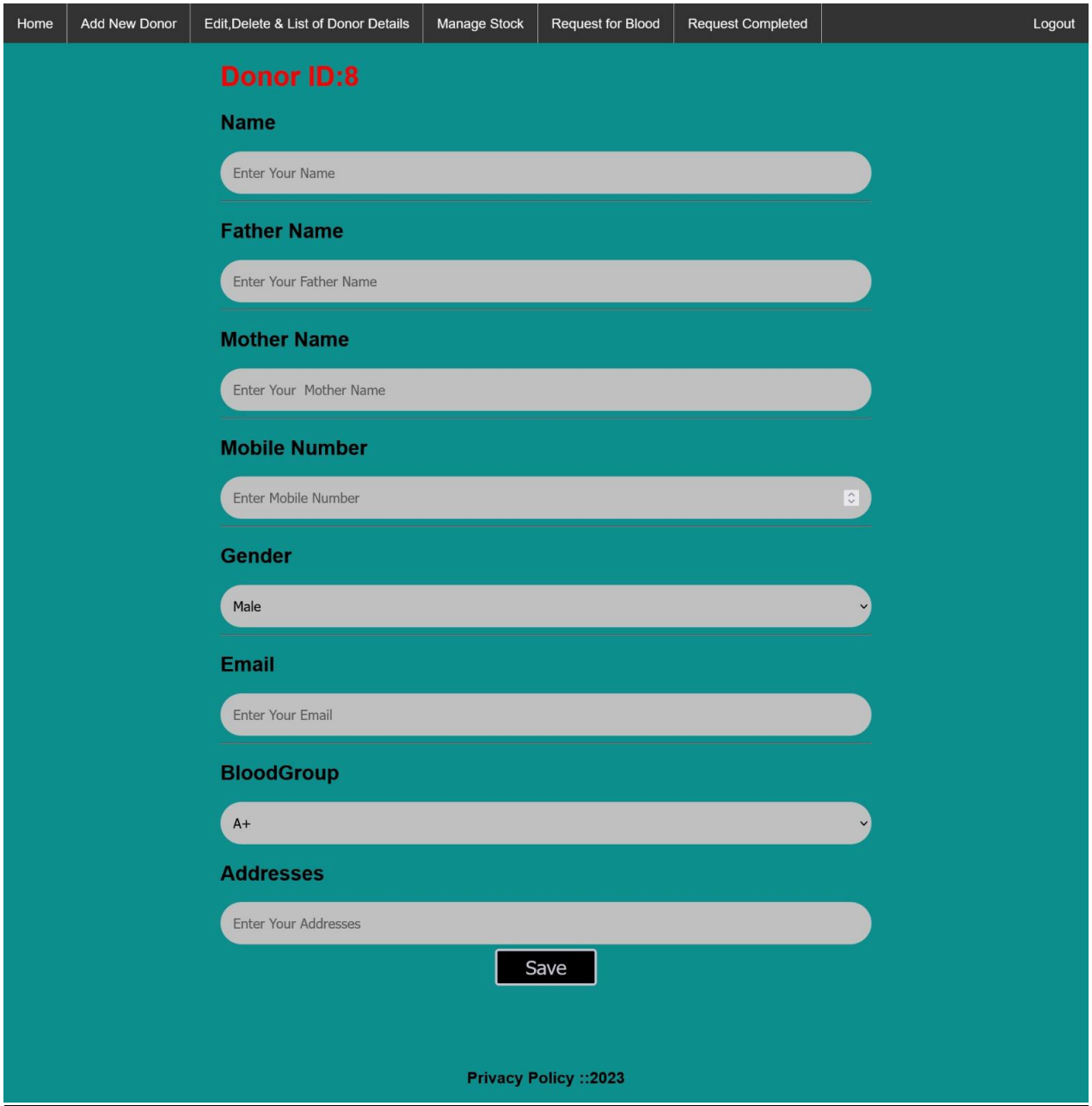
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title> </title>
</head>
<body>



 <form action="Register" method="post">
 <table>
 <tr>
 <td>User Name</td>
 <td><input type="text" name="UserName"></td>


 </tr>
 <tr>
  <td>Password</td>
 <td><input type="password" name="password"></td>
 </tr>
 <tr>


  <td>Email</td>
  <td><input type="text" name="email"></td>
  </tr>
  <tr>
  <td>Phone</td>
  <td><input type="text" name="mobileno"></td>
  </tr>
  <tr>
  <td>Submit</td>
  <td><input type="submit" value="register"></td>
  </tr>
  </table>
  </form>
```


Blood Drive Engine
Home
Login
Admin Login
BE GRATEFUL AND DONATE BLOOD
Give Blood Save Live
Helpline No : 345664

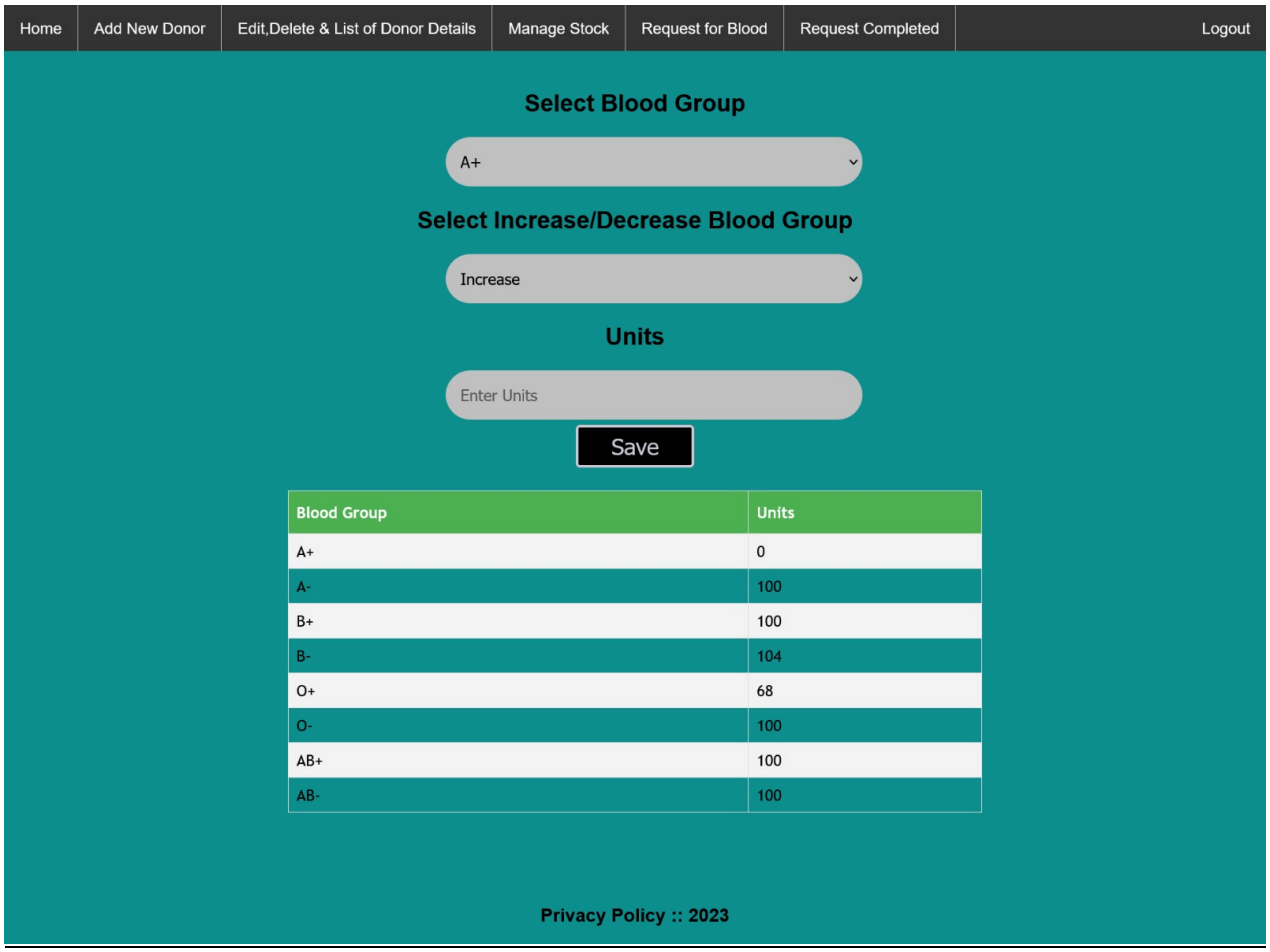

Home
User Id
Enter Your User Id
Password
Enter Your Password
Login
Create New User ?
Privacy Policy ::2023


Admin Login
* Request Registeration Form *
Name
Enter Your Name
Mobile Number
Enter Your Mobile Number
UserId
Enter Your Id
Password
Enter Your Password
Select Gender
Male
Register


Home
Admin Login
Admin Username
Enter Your Username
Admin Password
Enter Your Password
Submit
Privacy Policy ::2023


Home
Add New Donor
Edit,Delete & List of Donor Details
Manage Stock
Request for Blood
Request Completed
Logout
Donor ID:8
Name
Enter Your Name
Father Name
Enter Your Father Name
Mother Name
Enter Your Mother Name
Mobile Number
Enter Mobile Number
Gender
Male
Email
Enter Your Email
BloodGroup
A+
Addresses
Enter Your Addresses
Save
Privacy Policy ::2023


Home
Add New Donor
Edit,Delete & List of Donor Details
Manage Stock
Request for Blood
Request Completed
Logout
ID Name Father Name Mother Name Mobileno Gender Email Bloodgroup Addresses Edit Delete
6 lalith lee leena null male pranit@gmail.com A+ oop Edit Delete
7 shivani shinde praful ketki 908764533 Female shivani23@gnail.com O- pune solapur Edit Delete
Privacy Policy :: 2023


Home
Add New Donor
Edit,Delete & List of Donor Details
Manage Stock
Request for Blood
Request Completed
Logout
Select Blood Group
A+
Select Increase/Decrease Blood Group
Increase
Units
Enter Units
Save
Blood Group
Units
A+ 0
A- 100
B+ 100
B- 104
O+ 68
O- 100
AB+ 100
AB- 100
Privacy Policy :: 2023

Chapter 5
CONCLUSION

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in JSP web based application and SQL Server, but also about all handling procedure related with "Blood Drive Engine". It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently

Chapter 6
REFERENCES

[1] Sibing CT. Existing and recommended legislative framework for national blood transfusion policy. Global Journal of Transfusion Medicine. 2017 Jul 1:212) 80.

[2] Sinha S. Seth T. Colah RB, Battles AH. Haemoglobinopathies in India: estimates of blood requirements and treatment costs for the decade 2017-2020. Journal of community genetics. 2020 Jan;11(1):30-45

(3) Kulshreshtha V, Maheshwari DS. The blood donation centre Management Information System in India. International Journal of Engineering Research & Android applications (UERA) SSN:2248-9622

[4] Priya P. Saranya V. Shabana S. Subramani K. The optimization of blood donor information and management system by Technopedia. International Journal of Innovative Research in Science, Engineering and Technology. 2014 Feb;3(1)

[5] A Study on Blood Bank Management System" by A. Clemen Teena, K. Sankar and S. Kannan, Department of MCA, Bharath University, Selaiyur, Chennai-73, Tamil Nadu, India

[6] Gupta N. Gawande R., Thengadi N. MBB: A Life Saving Application. International Journal For Research in Emerging Science And Technology. 2015 Mar 2(1) 326-30.