VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018

A Dbms Mini Project Report

on

" BLOOD BANK MANAGEMENT "

Submitted in partial fulfillment of the requirements for the V semester and award of the degree of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi

Submitted by:

KASHISH	*1RN20CS064*
POOJITHA C	*1RN20CS098*

Under the Guidance of:

Mrs. S MAMATHA JAJUR
Assistant Professor
Dept. of CSE

Department of Computer Science and Engineering
(NBA Accredited for academic years 2018-19, 2019-20, 2020-22)
RNS Institute of Technology
Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560 098
2022-2023

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

(NBA Accredited for academic years 2022-25)

CERTIFICATE

Certified that the mini project work entitled **“BLOOD BANK MANAGEMENT”** has been successfully carried out by **”KASHISH”** bearing USN **”1RN20CS064”** and **“POOJITHA C”** bearing USN **”1RN20CS098”**, bonafide students of **”RNS Institute of Technology”** in partial fulfillment of the requirements for the 5th semester of **”Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University”**, Belagavi, during academic year 2022-2023. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the DBMS laboratory requirements of 5th semester BE, CSE.

Signature of the Guide **Mrs. S Mamatha Jajur** Assistant Professor Dept. of CSE	Signature of the HoD **Dr. Kiran P** Professor & HOD Dept. of CSE	Signature of the Principal **Dr. M K Venkatesha** Principal

External Viva:

Name of the Examiners Signature with Date

1.

2.

Acknowledgement

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

We are grateful to Management and **Dr. M K Venkatesha**, Principal, RNSIT, Bangalore, for his support towards completing this mini project.

We would like to thank **Dr. Kiran P**, Professor & Head, Department of Computer Science & Engineering, RNSIT, Bangalore, for his valuable suggestions and expert advice.

We deeply express our sincere gratitude to our guide **Mrs. S Mamatha Jajur**, Assistant Professor, **Dr. H R Shashidara**, Associate Professor, Department of CSE, RNSIT, Bangalore, for their able guidance, regular source of encouragement and assistance throughout this project.

We would like to thank all the teaching and non-teaching staff of Department of Computer Science & Engineering, RNSIT, Bengaluru-98 for their constant support and encouragement.

Abstract

This project deals with the ‘Blood Bank Management System’. The system is used for daily blood bank activities such as registering patient, maintaining employee details, donor details and total units of blood details. It is very difficult to do this process manually. Hence it is recommended to computerize this process by developing the relative software as the world is turning into information and technology.The Blood Bank management system will speed up the process of keeping records of patients and donors in an efficient manner.

Nowadays, the network plays an import role in people’s life. In the process of the improvement of the people’s living standard, people’s demands of the life’s quality and efficiency is more higher, the traditional blood bank system inconvenience gradually emerge, and the online blood bank management has gradually emerged useful in public.

Contents

Chapter 1

Introduction

1.1 Overview of Database Management

The DBMS manages the data; the database engine allows data to be accessed, locked and modified; and the database schema defines the database's logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform data administration procedures. The DBMS supports many typical database administration tasks, including change management, performance monitoring and tuning, security, and backup and recovery. Most database management systems are also responsible for automated rollbacks and restarts as well as logging and auditing of activity in databases and the applications that access them.

- Data independence and efficient access
- data integrity and security
- Reduced application development time

1.2 History of Database Management

Data is a collection of facts and figures. The data collection was increasing day to day and they needed to be stored in a device or a software which is safer.

Charles Bachman was the first person to develop the Integrated Data Store (IDS) which was based on network data model for which he was inaugurated with the Turing Award (The most prestigious award which is equivalent to Nobel prize in the field of Computer Science.). It was developed in early 1960's.

In the late 1960's, IBM (International Business Machines Corporation) developed the Integrated Management Systems which is the standard database system used till date in many places. It was developed based on the hierarchical database model. It was during the year 1970 that the relational database model was developed by Edgar Codd. Many of the database models we use today are relational based. It was considered the standardized database model from then.

The relational model was still in use by many people in the market.Later during the same decade (1980's), IBM developed the Structured Query Language (SQL) as a part of R project. It was declared as a standard language for the queries by ISO and ANSI. The Transaction Management Systems for processing transactions was also developed by James Gray for which he was felicitated the Turing Award.

Further, there were many other models with rich features like complex queries, datatypes to insert images and many others. The Internet Age has perhaps influenced the data models much more. Data models were developed using object oriented programming features, embedding with scripting languages like Hyper Text Markup Language (HTML) for queries. With humongous data being available online, DBMS is gaining more significance day by day.

The history of databases is a rich one, stretching as far back as the advent of the computer as we know it today. Databases have grown alongside computers, and changed immensely since their inception in the 1960s.

1.3 Applications of DBMS

The applications of DBMS management can be divided into four major areas:

- Display of information
- Design
- Simulation and animation
- User interfaces

Chapter 2

Requirement Analysis

2.1 Hardware Requirements

- The Hardware requirements are very minimal and the program can be run on most of the machines. Processor : Intel Atom® processor or Intel® Core™ i3 processor
- Processor Speed : 2.4 GHz
- RAM : 1 GB Storage Space : 40 GB Monitor Resolution : 1024*768 or 1336*768 or 1280*1024

2.2 Software Requirements

- Operating System - Windows 10 or later
- Text Editor - Atom
- SQL - SQLite3 Browser - Any one which supports HTML and CSS Additional tool - DB Browser

Chapter 3

Functional Requirements

3.1 Major entities

Online Blood Bank management system is to provide services for the people who are in need of blood by getting help from the donors who are interested in donating blood for the people. There are seven main modules in this system.

Admin

Donors

Donor Registration

Modifying Donor Information

Acceptors

Donor Search

Life Saving Contacts

3.2 End User Requirements

1. Main Goals:

• Our motto is to develop a software program for managing the Blood Bank process related to donors and patient accounts and to keep each and every track about their data like various processes efficiently.

• Hereby, our main objective is the ease of access of important data considering how conviniently we can keep track record of patient details.

2. Ease of access:

• The details can be easily added, deleted or updated without any hassle.

• Our software will perform and fulfill all the tasks that any administrator would desire.

3. Saving Lookup Time:

• The person looking for the information doesn't need to go through the whole database to do small operation

Chapter 4

Database Design

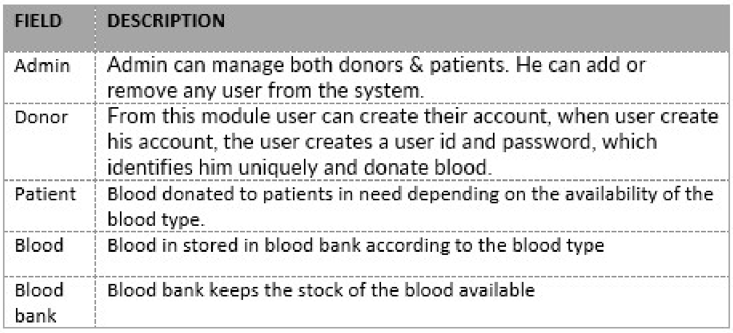
4.1 Entities,Attributes and Relationships

The database, called data, will have five tables admin, donor, blood, patient and blood bank. Each will hold information about either the patient or donor. The tables will be linked through a foreign key. The product table has the following fields:

4.2 Identify Major entities, attributes and relationships

- Login page to give access to admins.
- Adding donor details by the admin.
- Changing ID and password by admin and user. which stores the details of donors and patients.
- Admin can check all details but not password and modify them.
- Admin can delete the entities from the table.

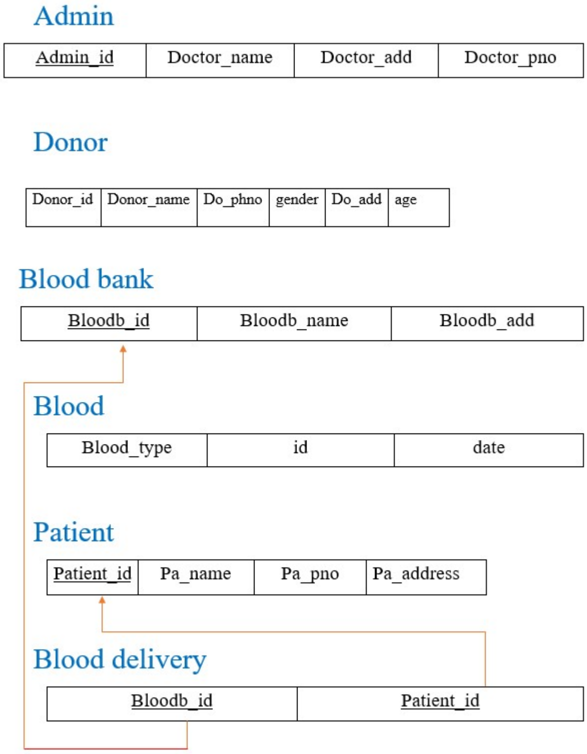
Figure 4.1: Entity Table



FIELD	DESCRIPTION
Admin	Admin can manage both donors & patients. He can add or remove any user from the system.
Donor	From this module user can create their account, when user create his account, the user creates a user id and password, which identifies him uniquely and donate blood.
Patient	Blood donated to patients in need depending on the availability of the blood type.
Blood	Blood in stored in blood bank according to the blood type
Blood bank	Blood bank keeps the stock of the blood available

4.3 Schema Diagram

Figure 4.2: Relational Schema



Admin

Admin_id	Doctor_name	Doctor_add	Doctor_pno

Donor

Donor_id	Donor_name	Do_phno	gender	Do_add	age

Blood bank

Bloodb_id	Bloodb_name	Bloodb_add

Blood

Blood_type	id	date

Patient

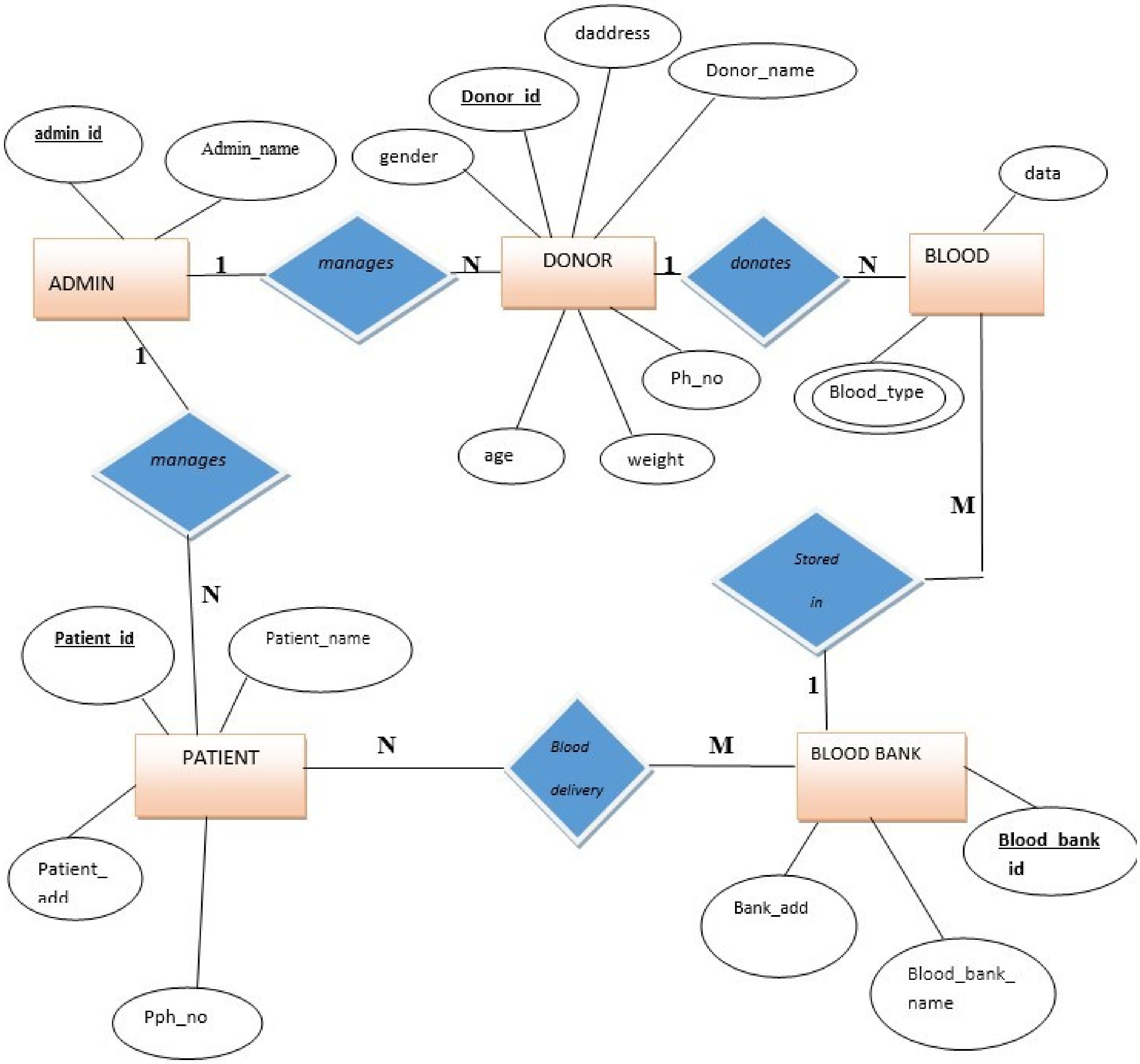
Patient_id	Pa_name	Pa_pno	Pa_address

Blood delivery

Bloodb_id	Patient_id

4.4 ER Diagram

Figure 4.3: Entity Relationship Diagram



Chapter 5

Description of Tools And Technologies

5.1 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as ¡img /¿ and ¡input /¿ introduce content into the page directly. Others such as ¡p¿...¡/p¿ surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

5.2 A CSS Framework

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Bootstrap is a free and open-source front-end web framework used for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap is the second most-starred project on GitHub, with more than 111,600 stars and 51,500 forks.

5.3 Django

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. Django is written in Python, which runs on many platforms. That means that you are not tied to any particular server platform, and can run your applications on many flavors of Linux, Windows, and macOS.

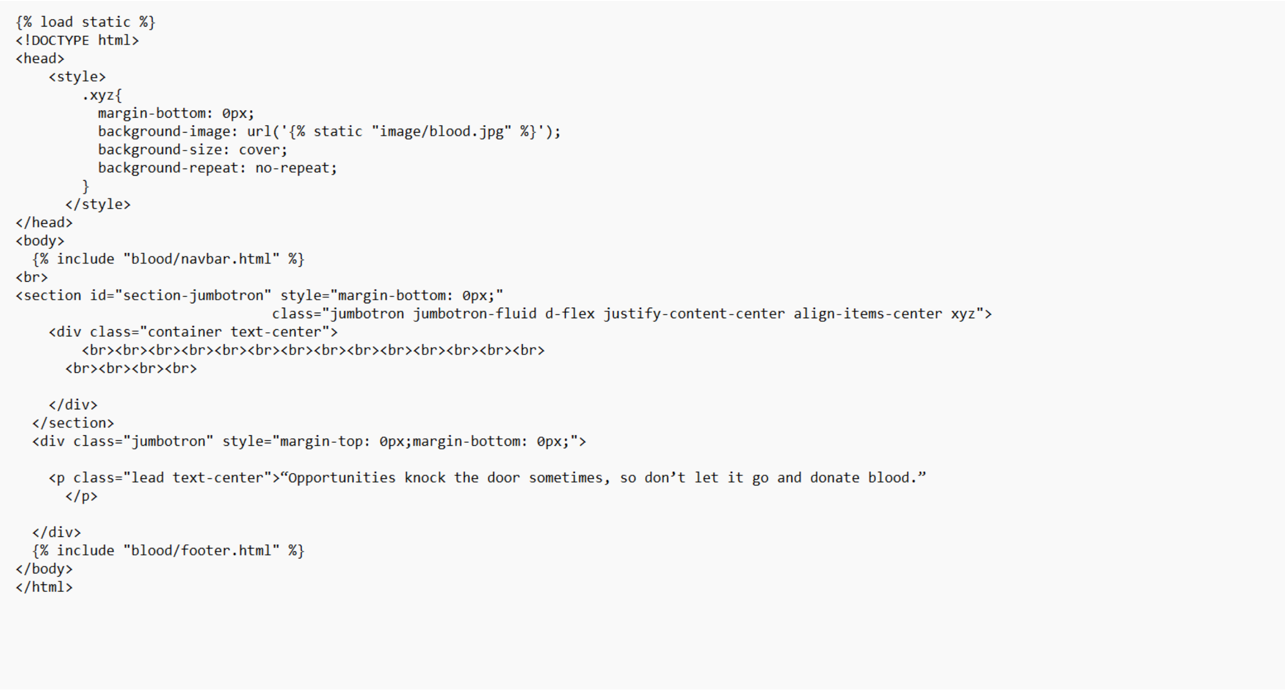
Django is a great addition to projects that need to handle large volumes of content (e.g., media files), user interactions or heavy traffic, or deal with complex functions or technology (e.g., machine learning). Yet it is simple enough for smaller-scale projects, or if you intend to scale your project to a much higher level. That’s why Django is used by so many companies that vary in size and goals.

Chapter 6

Implementation

6.1 Index page

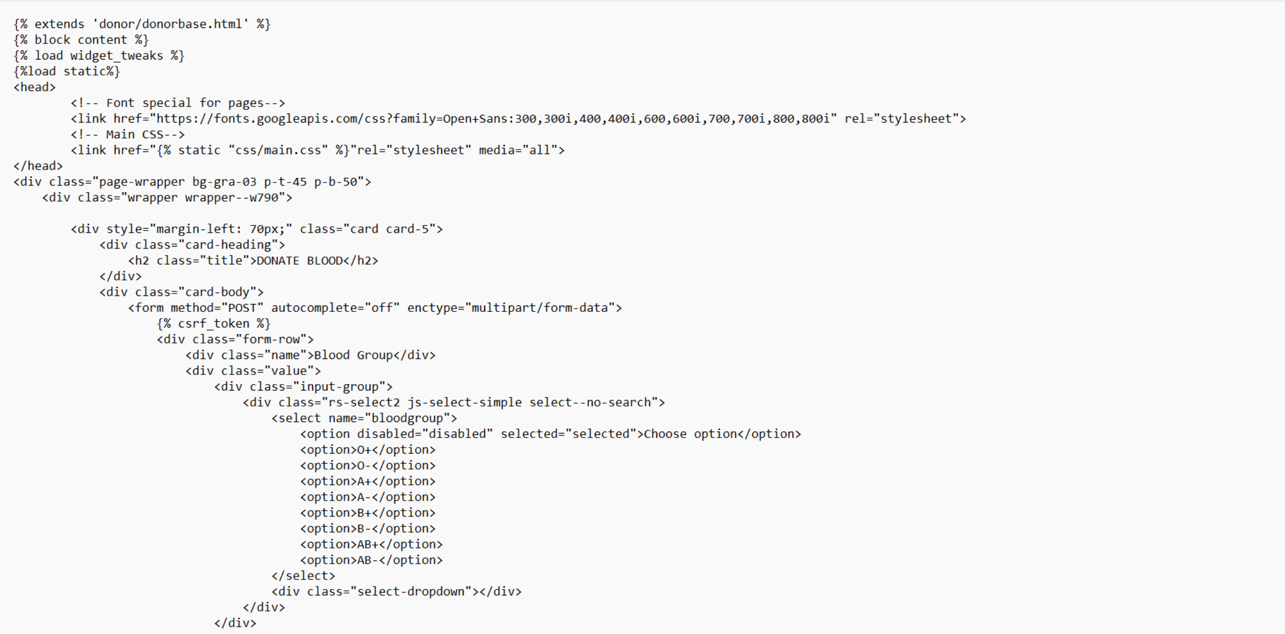
Figure 6.1: Index Page



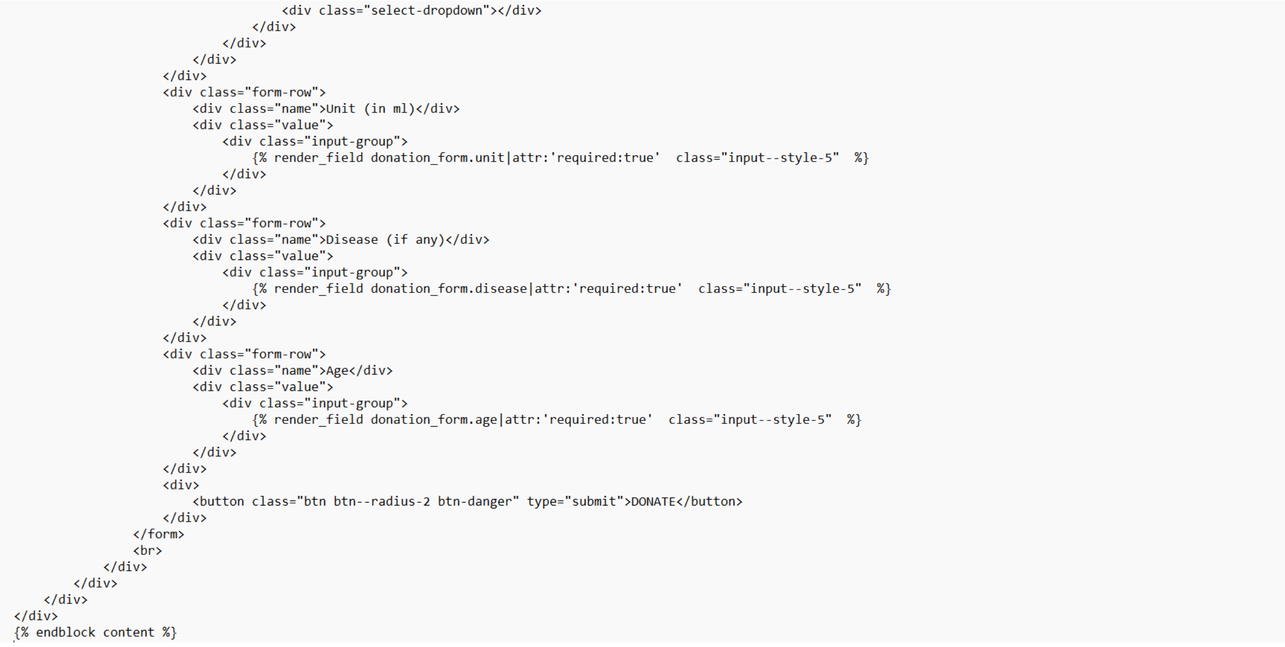
```
{% load static %}
<!DOCTYPE html>
<head>
    <style>
        .xyz{
          margin-bottom: 0px;
          background-image: url('{% static "image/blood.jpg" %}');
          background-size: cover;
          background-repeat: no-repeat;
        }
      </style>
</head>
<body>
  {% include "blood/navbar.html" %}
<br>
<section id="section-jumbotron" style="margin-bottom: 0px;"
                               class="jumbotron jumbotron-fluid d-flex justify-content-center align-items-center xyz">
    <div class="container text-center">
        <br><br><br><br><br><br><br><br><br><br><br><br><br><br>
      <br><br><br><br>

    </div>
  </section>
  <div class="jumbotron" style="margin-top: 0px;margin-bottom: 0px;">

    <p class="lead text-center">“Opportunities knock the door sometimes, so don’t let it go and donate blood.”
      </p>

  </div>
  {% include "blood/footer.html" %}
</body>
</html>
```

6.2 Donate blood page

Figure 6.2: Donate blood page



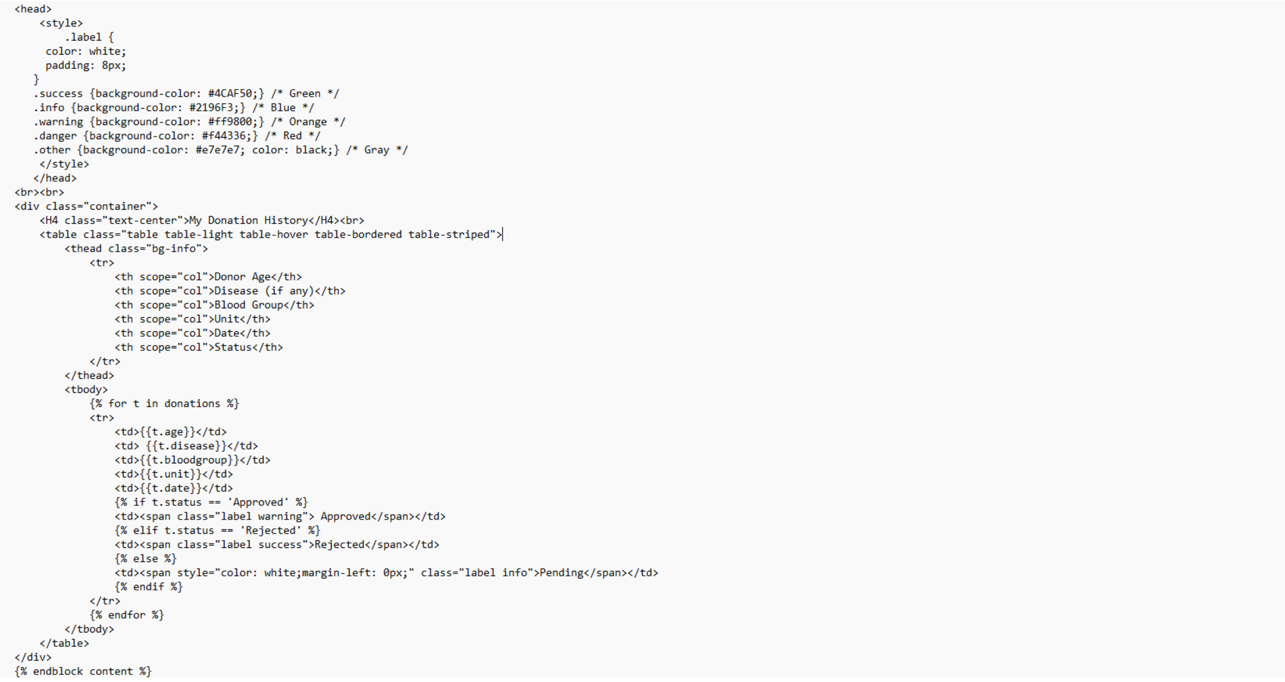
```
{% extends 'donor/donorbase.html' %}
{% block content %}
{% load widget_tweaks %}
{%load static%}
<head>
        <!-- Font special for pages-->
        <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i,800,800i" rel="stylesheet">
        <!-- Main CSS-->
        <link href="{% static "css/main.css" %}"rel="stylesheet" media="all">
</head>
<div class="page-wrapper bg-gra-03 p-t-45 p-b-50">
    <div class="wrapper wrapper--w790">

        <div style="margin-left: 70px;" class="card card-5">
            <div class="card-heading">
                <h2 class="title">DONATE BLOOD</h2>
            </div>
            <div class="card-body">
                <form method="POST" autocomplete="off" enctype="multipart/form-data">
                    {% csrf_token %}
                    <div class="form-row">
                        <div class="name">Blood Group</div>
                        <div class="value">
                            <div class="input-group">
                                <div class="rs-select2 js-select-simple select--no-search">
                                    <select name="bloodgroup">
                                        <option disabled="disabled" selected="selected">Choose option</option>
                                        <option>O+</option>
                                        <option>O-</option>
                                        <option>A+</option>
                                        <option>A-</option>
                                        <option>B+</option>
                                        <option>B-</option>
                                        <option>AB+</option>
                                        <option>AB-</option>
                                    </select>
                                    <div class="select-dropdown"></div>
                                </div>
                            </div>
```



```
                                    <div class="select-dropdown"></div>
                                </div>
                            </div>
                        </div>
                    </div>
                    <div class="form-row">
                        <div class="name">Unit (in ml)</div>
                        <div class="value">
                            <div class="input-group">
                                {% render_field donation_form.unit|attr:'required:true'  class="input--style-5"  %}
                            </div>
                        </div>
                    </div>
                    <div class="form-row">
                        <div class="name">Disease (if any)</div>
                        <div class="value">
                            <div class="input-group">
                                {% render_field donation_form.disease|attr:'required:true'  class="input--style-5"  %}
                            </div>
                        </div>
                    </div>
                    <div class="form-row">
                        <div class="name">Age</div>
                        <div class="value">
                            <div class="input-group">
                                {% render_field donation_form.age|attr:'required:true'  class="input--style-5"  %}
                            </div>
                        </div>
                    </div>
                    <div>
                        <button class="btn btn--radius-2 btn-danger" type="submit">DONATE</button>
                    </div>
                </form>
                <br>
            </div>
        </div>
    </div>
</div>
{% endblock content %}
```

6.3 Donation history

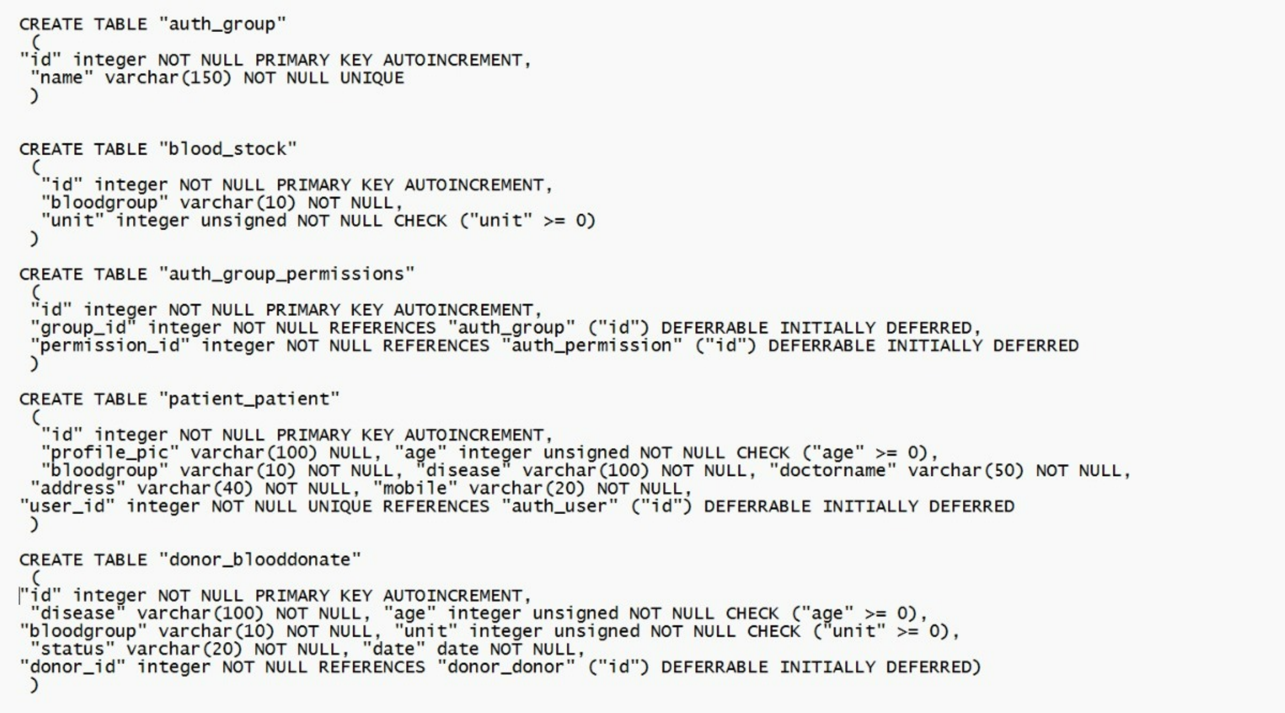
Figure 6.3: Donate blood page



```
<head>
    <style>
        .label {
     color: white;
     padding: 8px;
   }
   .success {background-color: #4CAF50;} /* Green */
   .info {background-color: #2196F3;} /* Blue */
   .warning {background-color: #ff9800;} /* Orange */
   .danger {background-color: #f44336;} /* Red */
   .other {background-color: #e7e7e7; color: black;} /* Gray */
    </style>
   </head>
<br><br>
<div class="container">
    <H4 class="text-center">My Donation History</H4><br>
    <table class="table table-light table-hover table-bordered table-striped">
        <thead class="bg-info">
            <tr>
                <th scope="col">Donor Age</th>
                <th scope="col">Disease (if any)</th>
                <th scope="col">Blood Group</th>
                <th scope="col">Unit</th>
                <th scope="col">Date</th>
                <th scope="col">Status</th>
            </tr>
        </thead>
        <tbody>
            {% for t in donations %}
            <tr>
                <td>{{t.age}}</td>
                <td> {{t.disease}}</td>
                <td>{{t.bloodgroup}}</td>
                <td>{{t.unit}}</td>
                <td>{{t.date}}</td>
                {% if t.status == 'Approved' %}
                <td><span class="label warning"> Approved</span></td>
                {% elif t.status == 'Rejected' %}
                <td><span class="label success">Rejected</span></td>
                {% else %}
                <td><span style="color: white;margin-left: 0px;" class="label info">Pending</span></td>
                {% endif %}
            </tr>
            {% endfor %}
        </tbody>
    </table>
</div>
{% endblock content %}
```

6.4 Create statements MySQL

Figure 6.4: Create statements

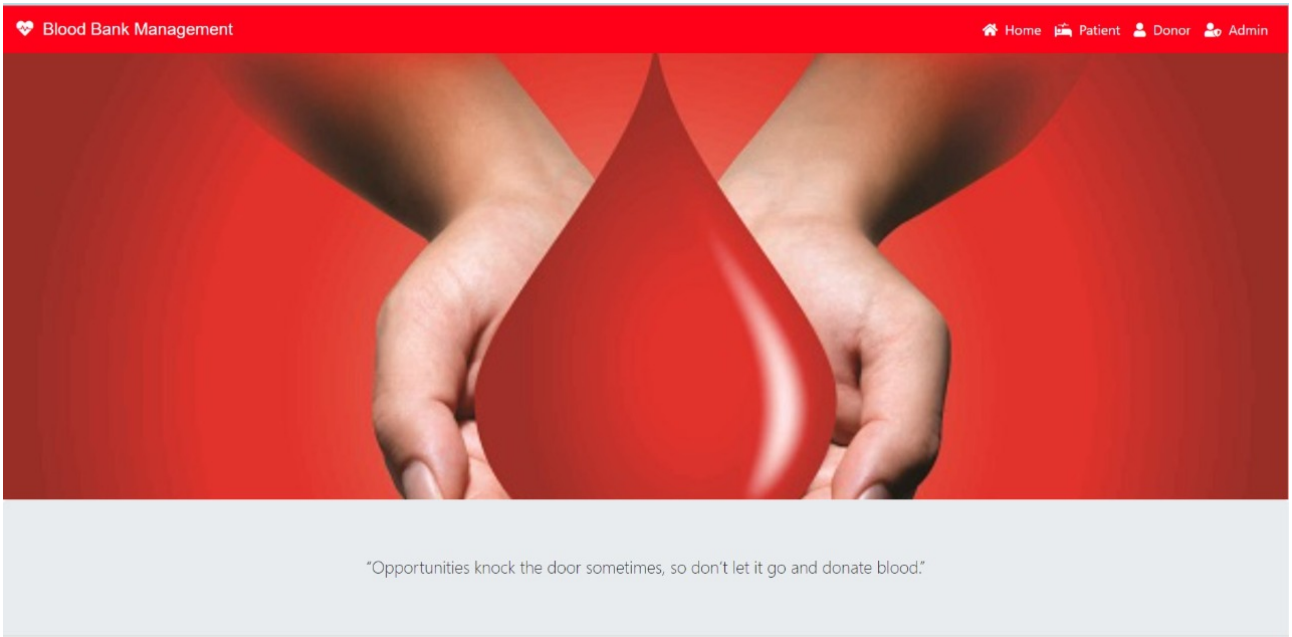


```
CREATE TABLE "auth_group"
 (
"id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
 "name" varchar(150) NOT NULL UNIQUE
 )


CREATE TABLE "blood_stock"
 (
  "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
  "bloodgroup" varchar(10) NOT NULL,
  "unit" integer unsigned NOT NULL CHECK ("unit" >= 0)
 )

CREATE TABLE "auth_group_permissions"
 (
 "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
 "group_id" integer NOT NULL REFERENCES "auth_group" ("id") DEFERRABLE INITIALLY DEFERRED,
 "permission_id" integer NOT NULL REFERENCES "auth_permission" ("id") DEFERRABLE INITIALLY DEFERRED
 )

CREATE TABLE "patient_patient"
 (
  "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
  "profile_pic" varchar(100) NULL, "age" integer unsigned NOT NULL CHECK ("age" >= 0),
  "bloodgroup" varchar(10) NOT NULL, "disease" varchar(100) NOT NULL, "doctorname" varchar(50) NOT NULL,
 "address" varchar(40) NOT NULL, "mobile" varchar(20) NOT NULL,
"user_id" integer NOT NULL UNIQUE REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED
 )

CREATE TABLE "donor_blooddonate"
 (
"id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
 "disease" varchar(100) NOT NULL, "age" integer unsigned NOT NULL CHECK ("age" >= 0),
"bloodgroup" varchar(10) NOT NULL, "unit" integer unsigned NOT NULL CHECK ("unit" >= 0),
 "status" varchar(20) NOT NULL, "date" date NOT NULL,
"donor_id" integer NOT NULL REFERENCES "donor_donor" ("id") DEFERRABLE INITIALLY DEFERRED)
 )
```

Chapter 7

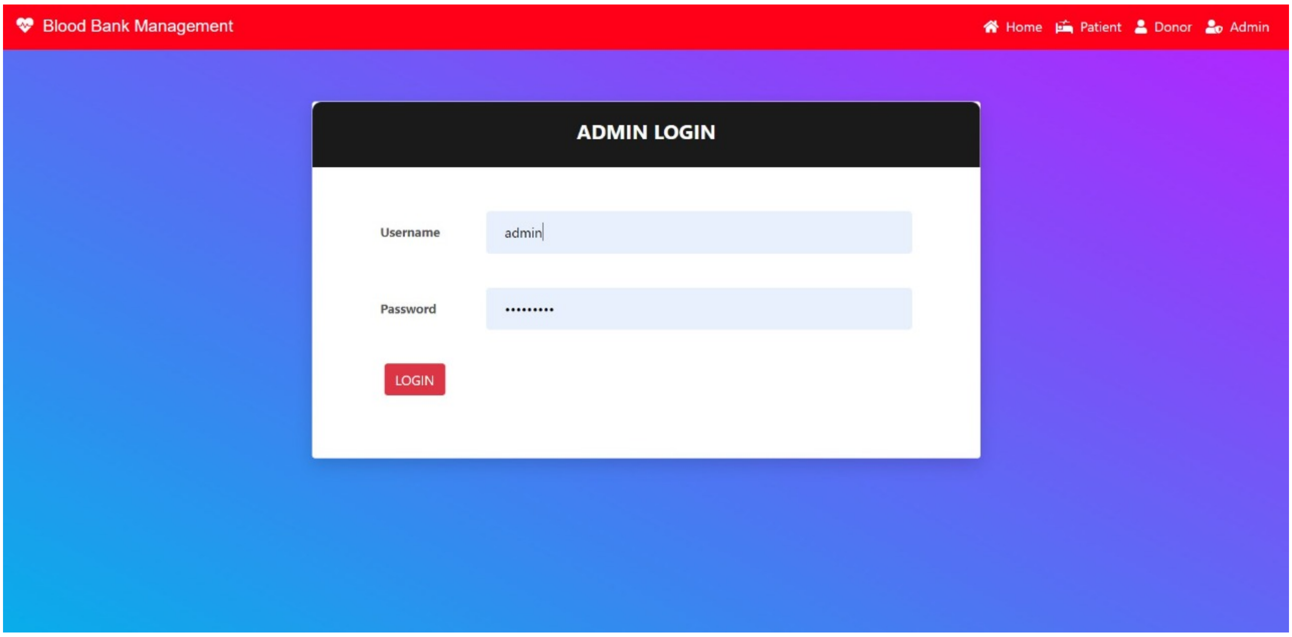
Results & Snapshots

Figure 7.1: Home Page



"Opportunities knock the door sometimes, so don't let it go and donate blood."

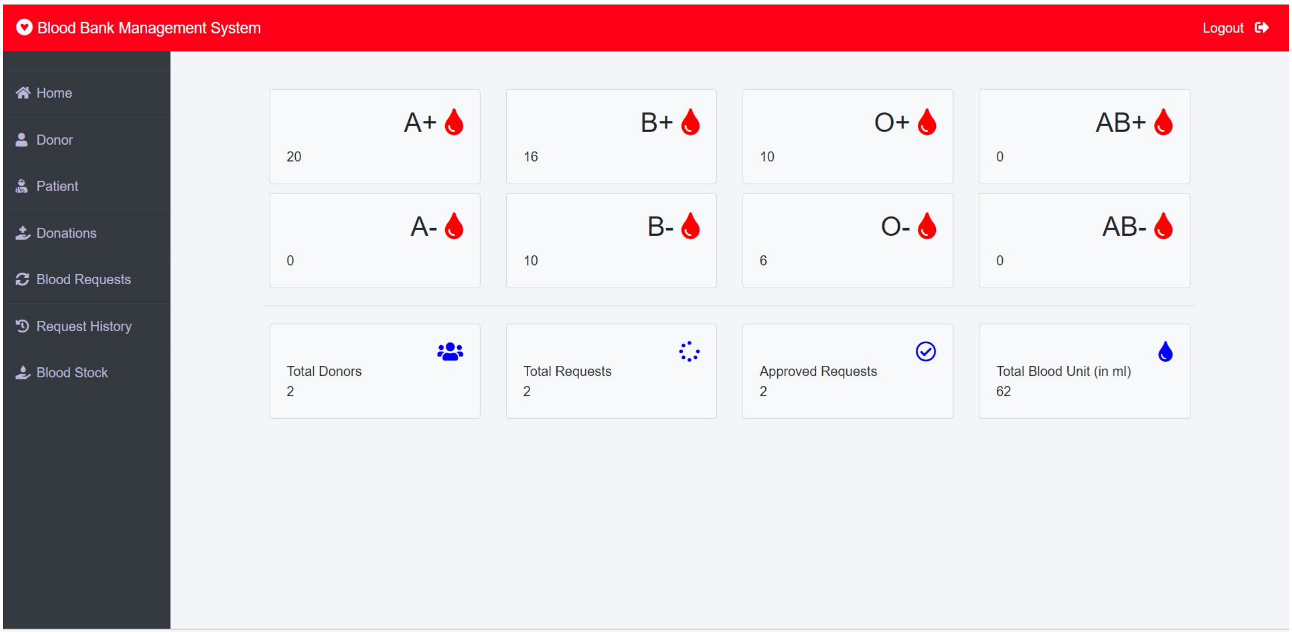
Figure 7.2: Admin Login Page



Username: admin

Password: •••••••••

Figure 7.3: Admin Access



A+ 20

B+ 16

O+ 10

AB+ 0

A- 0

B- 10

O- 6

AB- 0

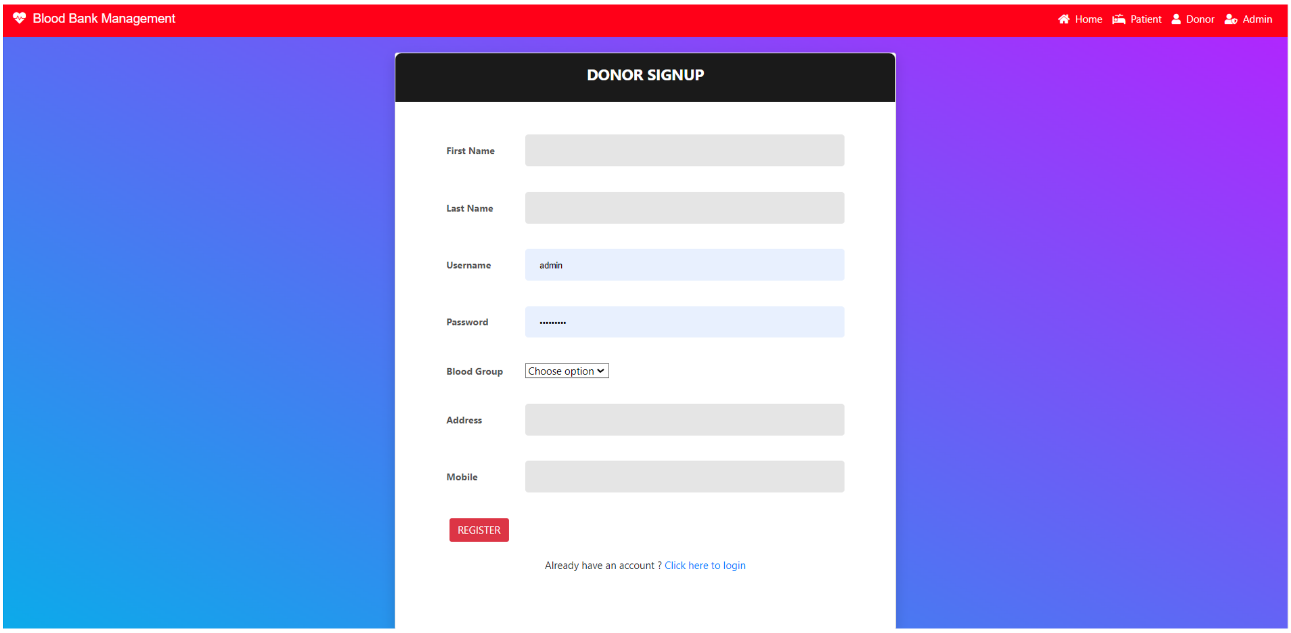
Total Donors 2

Total Requests 2

Approved Requests 2

Total Blood Unit (in ml) 62

Figure 7.4: Donor signup

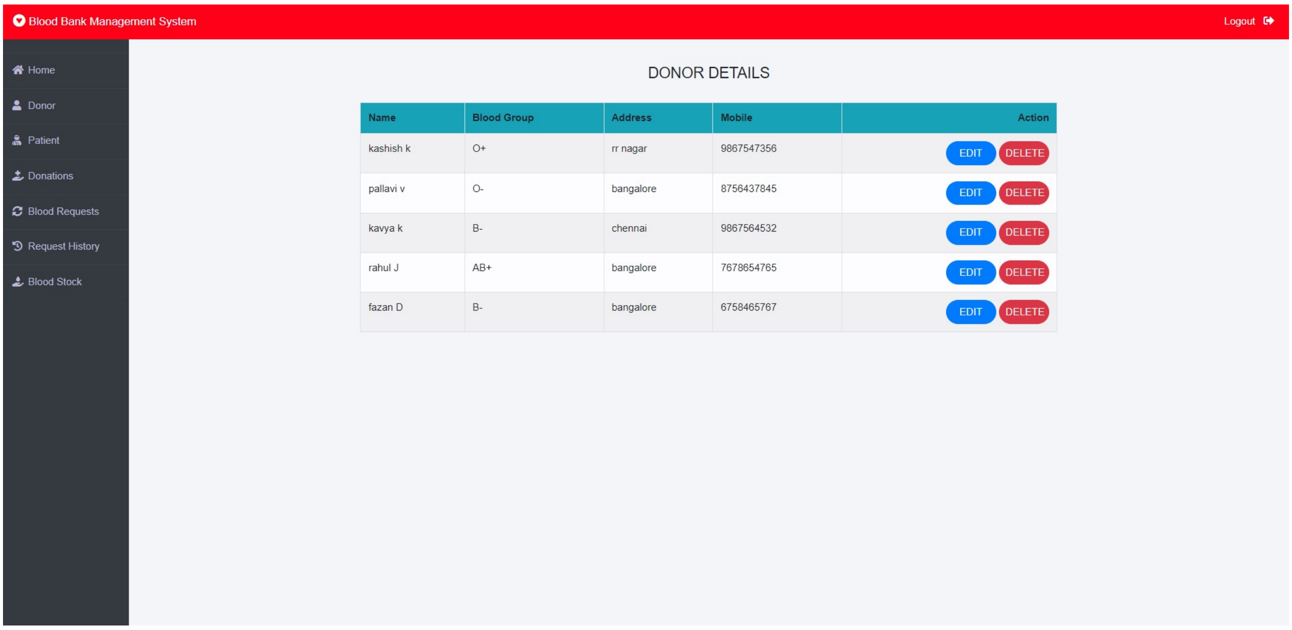


DONOR SIGNUP

First Name	
Last Name	
Username	admin
Password	•••••••••
Blood Group	Choose option
Address	
Mobile	

Already have an account ? Click here to login

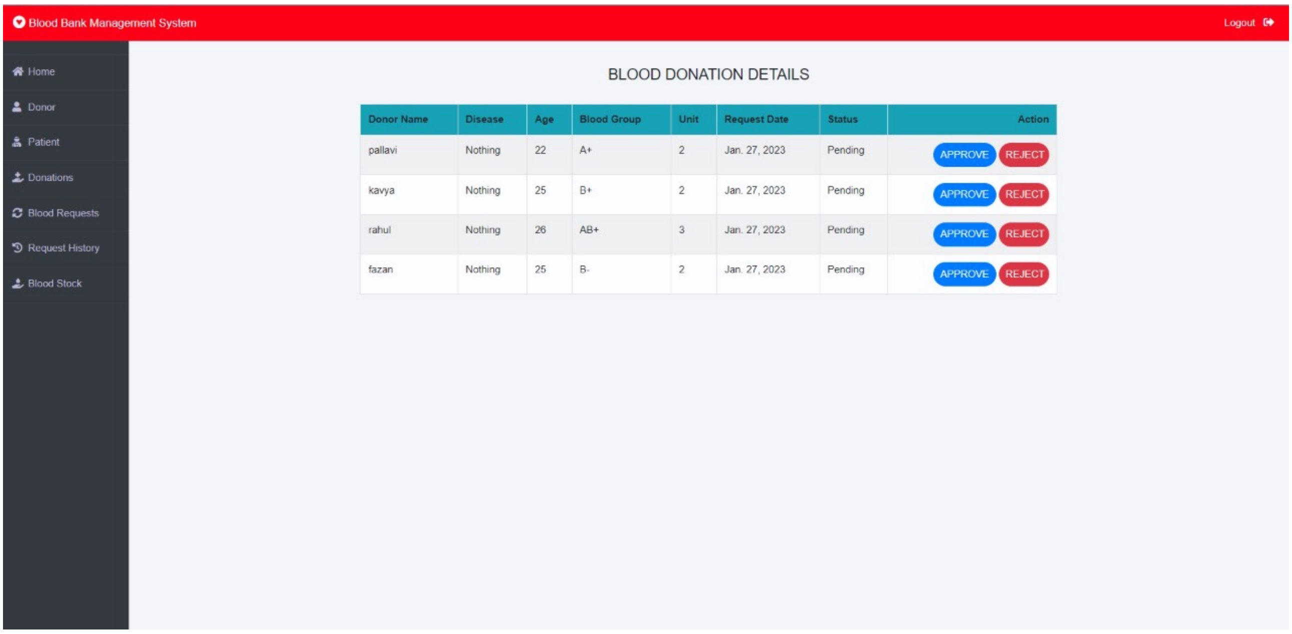
Figure 7.5: Donor Details



DONOR DETAILS

Name	Blood Group	Address	Mobile	Action
kashish k	O+	rr nagar	9867547356	EDIT DELETE
pallavi v	O-	bangalore	8756437845	EDIT DELETE
kavya k	B-	chennai	9867564532	EDIT DELETE
rahul J	AB+	bangalore	7678654765	EDIT DELETE
fazan D	B-	bangalore	6758465767	EDIT DELETE

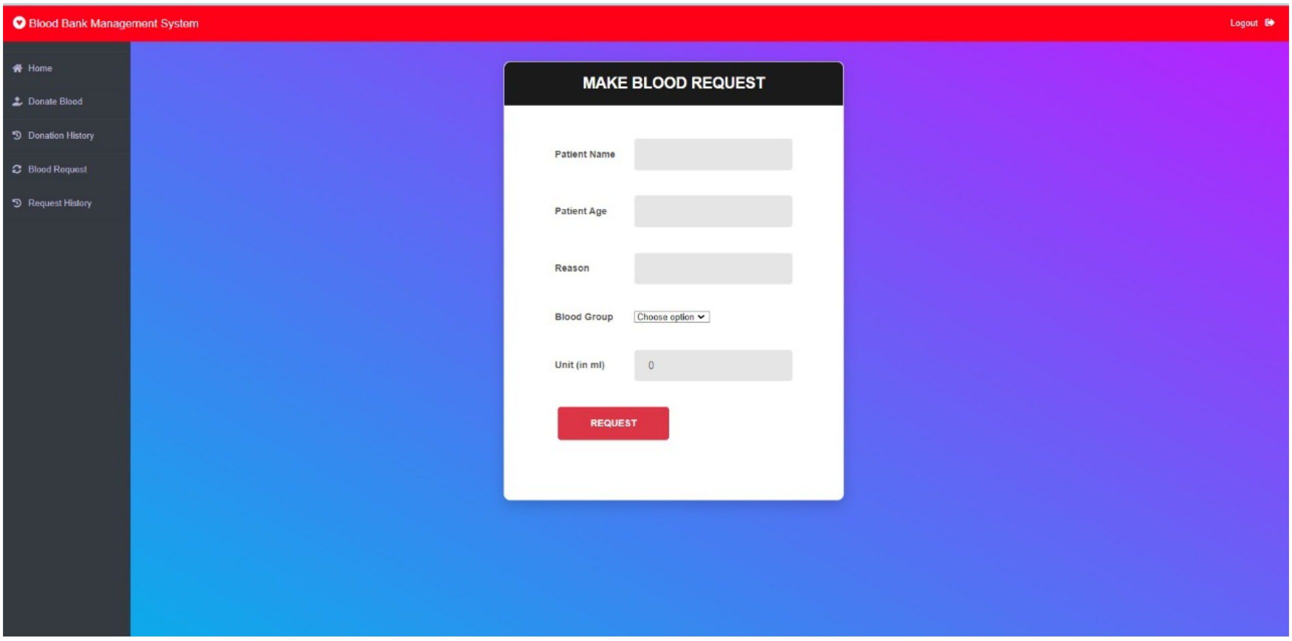
Figure 7.6: blood donation details



BLOOD DONATION DETAILS

Donor Name	Disease	Age	Blood Group	Unit	Request Date	Status	Action
pallavi	Nothing	22	A+	2	Jan. 27, 2023	Pending	APPROVE REJECT
kavya	Nothing	25	B+	2	Jan. 27, 2023	Pending	APPROVE REJECT
rahul	Nothing	26	AB+	3	Jan. 27, 2023	Pending	APPROVE REJECT
fazan	Nothing	25	B-	2	Jan. 27, 2023	Pending	APPROVE REJECT

Figure 7.7: blood request



MAKE BLOOD REQUEST

Patient Name

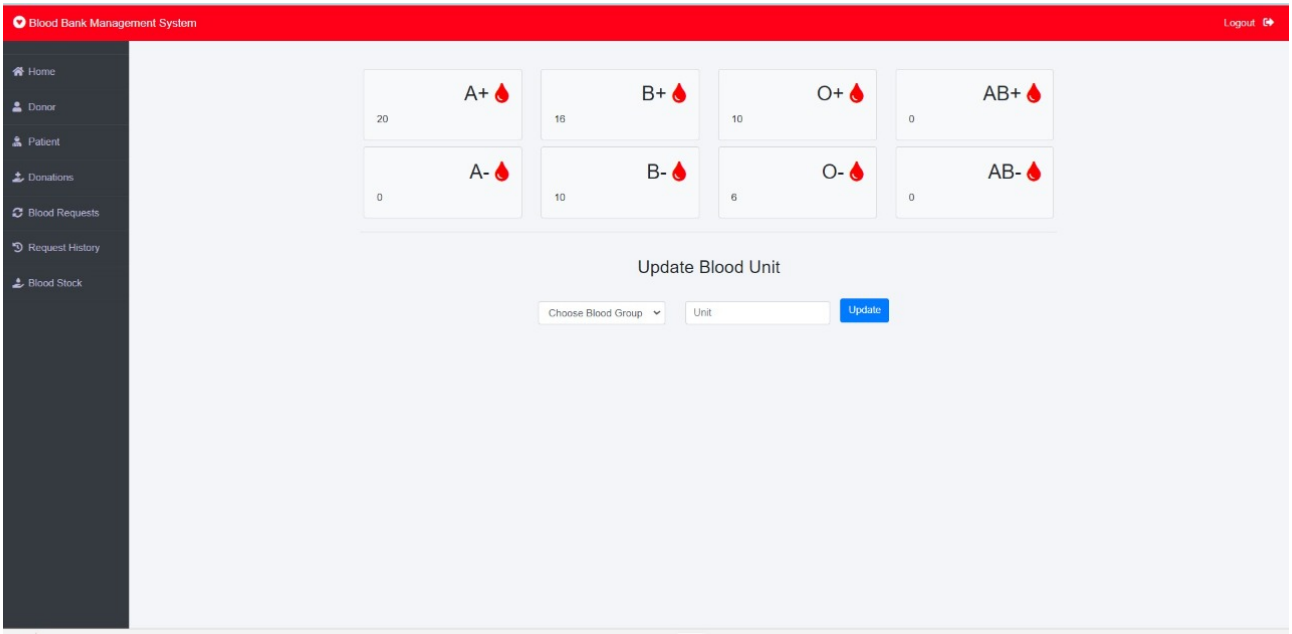
Patient Age

Reason

Blood Group: Choose option

Unit (in ml): 0

Figure 7.8: blood stock



A+	B+	O+	AB+
20	16	10	0
A-	B-	O-	AB-
0	10	6	0

Update Blood Unit

Chapter 8

Conclusion & Future Enhancements

8.1 Conclusion

After implementing the application it will contain the advantages were incomparable to the present contemporary systems used by company. The most admirable feature founded was its simplicity in terms of application to the user but its highly beneficial outputs can't be ignored. The users will be highly benefited after using the system.It is hoped that this project will help the future developers to modify and implement the system. After modifying some techniques of the programs, it will give us the best performance as our requirements. The project will be very useful for the users.This proposed Blood Bank Management System gives a reliable platform for both donors and acceptors.This DBMS system is a web-based application that helps to minimize human errors and problems pertaining to data redundancy.

The project provides simple retrieval techniques and easy updating and deletion operations thus helping inefficient maintenance of records.

8.2 Future Enhancements

Software development is never – ending process and continues the life of the software as per the changing needs of the user from time to time. The project is no doubt has been developed keeping in mind easy modification and enhancement that may be required from time to time. We are trying to cover all their existing system for blood return of all blood groups but due to shortage of time we become unable to include many things. Only with a little more doing it is possible to design the formats for those returns. Moreover, an on-line system will be more helpful to the organization. All these can be considered to be future scope for this project.

References

[1] " J D Ullman *""Principles of Database Systems"*

[2] " Elmasri Ramez and Navathe Shamkant","*"Fundamentals of Database System"* ", 2nd Edition, Pearson Education, 2001.

[3] Learning Web Design: Jennifer Robins, *" Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics* ,

[4] @online MySql,`https://www.techtarget.com/`

[5] @online HTML AND CSS , `https://www.codecademy.com/learn/learn-html`