A

PROJECT REPORT ON

“BLOOD DONATION MANAGEMENT SYSTEM”

SUBMITTED BY

Ms. Sidra Ajaz Mulla

SUBMITTED TO

SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

IN PARTIAL FULFILLMENT OF DEGREE

MASTER OF COMPUTER APPLICATION [SEM II]

UNDER THE GUIDANCE OF

Dr. Neeta Raskar

Through,

SVIMS
Sadhu Vaswani Institute
of Management Studies
BELIEVE AND ACHIEVE

Sadhu Vasvani Institute of Management Studies for Girls,
Koregoan Park, Pune – 411001

2024 – 2025

DECLARATION BY STUDENT

To,
The Director,
SVIMS, Koregoan Park, Pune

I undersigned hereby declare that this project titled, “Blood Donation Management System ” written and submitted by me to SPPU, Pune in partial fulfilment of the requirement of the degree of MASTER OF COMPUTER APPLICATION (MCA I) under the guidance of Dr. Neeta Raskar, is my original work.

I further declare that to the best of my knowledge and belief , this project has not been submitted to this or any other University or Institution for the award of any Degree.

Place : Pune
Date : 22nd April 2025

Sidra Ajaz Mulla

ACKNOWLEDGEMENT

I extend my sincere gratitude to Dr. B.H. Nanwani, Dr. Neeta Raskar and Prof. Deepali Gavhane for allowing me to carry out the study and for their constant encouragement, valuable suggestions, and guidance during the research work.

I extend my special gratitude to my family members and friends who encouraged and motivated me to complete the project report.

Place : Pune

Date : 22nd April 2025

Sidra Ajaz Mulla

CHAPTER 1 : INTRODUCTION

1.1 CLIENT / ORGANIZATION PROFILE

A Blood Management System (BMS) is a critical healthcare information system designed to streamline and optimize the entire lifecycle of blood and blood products, from donation to transfusion. In today's healthcare landscape, ensuring the availability of safe and sufficient blood supplies is paramount for saving lives and supporting various medical procedures. However, managing blood effectively involves complex processes, including donor recruitment and screening, blood collection, processing, testing, storage, distribution, and ultimately, transfusion to patients.

Traditional, manual methods of managing blood are often prone to inefficiencies, errors, and wastage. A well-designed and implemented BMS addresses these challenges by leveraging technology to automate and integrate these processes. This leads to improved efficiency, enhanced safety, reduced costs, and better patient outcomes.

A Blood Management System project is therefore a vital undertaking for any healthcare organization or regional blood transfusion service. It represents a significant investment in infrastructure and technology aimed at modernizing blood banking practices and ultimately contributing to a safer and more efficient healthcare system. The successful implementation of such a system requires careful planning, stakeholder collaboration, and a thorough understanding of the specific needs and challenges of the target environment

1.2 NEED OF SYSTEM

The healthcare sector faces significant challenges in managing the demand and supply of blood. Traditional methods of blood donation tracking and inventory management often lead to inefficiencies such as shortages, overstocking, expired units, and data mismanagement. A **Blood Management System** addresses these issues through automation, accuracy, and real-time monitoring.

The need for a Blood Management System (BMS) arises from several critical factors in modern healthcare:

1. Ensuring Blood Safety and Quality:

- **Reducing Errors:** Manual processes in blood handling are prone to errors in labeling, cross-matching, and transfusion, which can have severe consequences. A BMS automates these processes, minimizing the risk of such errors.
- **Infection Control:** The system enforces stringent screening and testing protocols for donated blood, reducing the risk of transfusion-transmitted infections (TTIs) like HIV, Hepatitis B, and Hepatitis C.
- **Traceability:** A BMS provides a complete audit trail of each blood unit, from donor to recipient, allowing for swift identification and management in case of adverse reactions or contamination.
- **Proper Storage and Handling:** The system can monitor and manage storage conditions (temperature, humidity) and track expiration dates to maintain blood quality and prevent wastage.

2. Optimizing Blood Inventory Management:

- **Preventing Shortages and Wastage:** By accurately tracking blood inventory levels, a BMS helps ensure that the right blood components are available when needed, preventing critical shortages and minimizing wastage due to expiration.
- **Efficient Distribution:** The system can facilitate the efficient distribution of blood products to different departments or healthcare facilities based on demand.

- **Cost Reduction:** Minimizing wastage and optimizing inventory levels directly contribute to reducing operational costs for blood banks and hospitals.

3. Streamlining Operations and Improving Efficiency:

- **Automation of Tasks:** A BMS automates many manual tasks, such as donor registration, blood collection details, processing information, and report generation, freeing up staff for more critical activities.
- **Improved Communication:** The system can enhance communication and coordination between different departments involved in blood management, such as the blood bank, laboratory, and wards.
- **Faster Information Retrieval:** Electronic records allow for quick and easy access to donor information, blood inventory details, and patient transfusion history, saving time and improving decision-making.
- **Better Reporting and Analysis:** A BMS can generate comprehensive reports on various aspects of blood management, providing valuable insights for planning, resource allocation, and quality improvement.

4. Enhancing Donor Management:

- **Efficient Record Keeping:** The system maintains detailed records of blood donors, including their contact information, donation history, and eligibility status.
- **Improved Recruitment and Retention:** A BMS can facilitate communication with donors, schedule appointments, send reminders, and manage donor databases for effective recruitment and retention efforts.
- **Ensuring Donor Safety:** The system can track donor health information and donation frequency to ensure their well-being.

5. Meeting Regulatory Requirements:

- Blood banks and transfusion services must adhere to various national and international regulations and standards related to blood safety and quality. A BMS helps in maintaining compliance by enforcing standardized procedures and documentation.

In essence, a Blood Management System is not just a technological upgrade; it is a fundamental requirement for ensuring a safe, efficient, and sustainable blood supply that is crucial for effective healthcare delivery and saving lives. It addresses the inherent limitations and risks associated with manual blood banking practices, leading to better patient outcomes and a more robust healthcare system.

1.3 SCOPE AND FEASIBILITY OF WORK

The Blood Management System is designed to serve multiple stakeholders in the healthcare sector, including blood banks, hospitals, donors, and healthcare administrators. Its scope includes the automation and management of key operations related to blood donation and distribution. The system will:

- Maintain a centralized **database of donors**, including personal details, blood type, donation history, and eligibility.
- Track and manage **blood inventory**, including collection, testing, storage, and usage.
- Provide a **searchable interface** for locating required blood types quickly.
- Enable **scheduling and notification** systems for donor appointments and blood drives.
- Generate **reports and analytics** for monitoring usage, donor trends, and stock levels.
- Ensure **data security and privacy**, in compliance with health data regulations.

The system can be implemented at local, regional, or national levels, depending on the infrastructure and goals of the healthcare organization.

Feasibility of the Blood Management System

1. Technical Feasibility

- The system can be developed using widely available web and mobile technologies.
- Integration with existing hospital management systems and lab equipment is possible.
- Cloud-based deployment ensures scalability, data backup, and remote access.

2. Operational Feasibility

- The system simplifies the workflow for hospital staff and blood bank operators.
- With minimal training, users can adapt quickly due to an intuitive user interface.
- Real-time data access supports faster and more informed decision-making.

3. Economic Feasibility

- Reduces costs related to blood wastage, shortages, and manual errors.
- Long-term savings due to automation, efficient stock management, and reduced emergency procurement.
- Initial development and deployment costs are justified by the system's potential to save lives and resources.

4. Legal and Ethical Feasibility

- Can be designed to comply with health data protection laws (e.g., HIPAA, GDPR).
- Ensures donor confidentiality and ethical handling of sensitive information.

1.4 OPERATING SYSTEM – HARDWARE AND SOFTWARE

1. Server-Side Requirements

Component	**Specification**
Processor	Intel Xeon Quad Core or higher
RAM	Minimum 16 GB (32 GB recommended)
Storage	SSD – Minimum 500 GB (RAID recommended)
Operating System	Ubuntu Server / CentOS / Windows Server 2016+
Database	MySQL / PostgreSQL / MS SQL Server
Web Server	Apache / Nginx / IIS
Backend Tech	Node.js / PHP / Python / Java
Security	SSL Certificate, Firewall, Antivirus, OS Updates
Network	High-speed internet, secure access, static IP
Power Backup	UPS or Generator
Backup System	External drives or cloud-based backup solution

2. Client-Side Requirements

Component	**Specification**
Processor	Intel i3 or higher
RAM	Minimum 4 GB
Storage	Minimum 100 GB HDD/SSD
Operating System	Windows 10+ / Ubuntu / macOS
Web Browser	Google Chrome / Firefox / Microsoft Edge (latest)
Display	15” monitor or larger
Peripherals	Keyboard, Mouse, Printer, Barcode Scanner (optional)
Network	Stable LAN/Wi-Fi connection
Additional Software	PDF Reader, Office Suite

1.5 ARCHITECTURE OF THE SYSTEM

The architecture of the Blood Management System is designed to ensure **modularity, scalability, data integrity**, and **secure access**. A typical implementation follows a **three-tier architecture** (also known as **multi-tier**), separating the presentation, application, and data layers.

1. Layered Architecture

Layer	**Technology / Tools Used**	**Responsibilities**
Presentation Layer	Java Swing / JavaFX (within NetBeans)	Provides user interface for donors, admins, hospitals to interact with the system
Business Logic Layer	Core Java Classes and Methods	Handles processing, validation, and interaction between UI and database
Data Layer	MySQL Database (Connected via JDBC)	Stores and manages all application data (users, blood inventory, requests, etc.)

2. Key Components

Module	**Functionality**
Donor Management	Registration, login, donation tracking, eligibility check
Blood Inventory	Tracks blood types, quantity, expiry dates, and availability
Request Handling	Allows hospitals to request blood, tracks request status
Admin Controls	User management, report generation, system monitoring
Notification System	Sends alerts for low inventory, donor reminders (email/SMS integration optional)

1.6 DETAILDED DISCRIPTION OF TECHNOLOGY USED

➢ HTML (HyperText Markup Language):::

HTML (HyperText Markup Language) is the standard language used to create and design the structure of web pages. It acts as the backbone of every website, defining the content and layout that is displayed in web browsers.HTML is the standard language used to structure the content of web pages.

Key Features of HTML

- **Structure:**

Organizes content into meaningful sections like headers, footers, navigation bars, and main content areas.

- **Flexibility**:

Easily integrates with CSS for styling and JavaScript for interactivity.

- **Accessibility**:

Semantic elements improve usability for screen readers and other assistive technologies.

- **Cross-Browser Compatibility:**

Supported by all major web browsers, ensuring consistent user experiences.
HTML in Boutique Management System

- **HTML will be used to build:**

User Interfaces: Inventory management, sales tracking, customer management, and reports.
Forms: Data input for adding products, processing sales, and updating inventory.
Product Display: Showcases products with images, descriptions, and prices.

Purpose in the System

Page Structure:
Defines the layout and elements such as headers, footers, navigation bars, forms, and tables.
Ensures semantic structure for accessibility (e.g., <header>, <section>, <article>).
User Interaction:

Provides the foundation for interactive elements like buttons, forms, and links.

Integration:

Embeds multimedia (e.g., images and videos) for showcasing products in the boutique catalog.

- CSS (Cascading Style Sheets):::

CSS is used to style and enhance the appearance of HTML elements.

CSS (Cascading Style Sheets) is a language used to describe the presentation and layout of a web page. It works alongside HTML to style and format content, making websites visually appealing and easier to navigate.

Key Characteristics of CSS

- **Styling Language:**

Controls the look and feel of web pages, such as colors, fonts, and spacing.

- **Separation of Content and Design:**

Allows developers to keep HTML for structure and CSS for styling, promoting cleaner and more manageable code.

- **Cascading:**

Resolves styling conflicts by prioritizing rules based on specificity, importance, and source order.

- **How CSS Works**

CSS styles are applied to HTML elements using:

Inline CSS:

Directly within an HTML tag using the style attribute.

Internal CSS:

Defined within a <style> tag in the <head> section of an HTML document.

External CSS:

Written in a separate .css file and linked to the HTML document using the <link> tag.

Purpose in the System

Styling:

Adds colors, fonts, and background images to improve visual appeal.

Layout Design:

Manages the positioning of elements on the page using techniques like flexbox, grid, or CSS positioning.

Responsive Design:
Ensures compatibility with various devices (e.g., desktops, tablets, and mobile phones).
Theming:
Provides consistency across pages by defining reusable styles.

➢ PYTHON

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. It is widely used in web development, data science, machine learning, artificial intelligence, automation, and more.

Key Characteristics of Python

- **Simple and Readable:**

Python's syntax is clean and easy to understand, making it beginner-friendly.

- **Dynamically Typed:**

Variable types are determined at runtime, so you don't need to declare types explicitly.

- **Interpreted Language:**

Python code is executed line by line, allowing for immediate feedback during development.

- **Extensive Libraries:**

Python has a vast standard library and third-party modules for various applications, from data analysis to web development.

- **Cross-Platform:**

Python runs on multiple operating systems, including Windows, macOS, and Linux.

Applications of Python

Web Development:
Frameworks like Django and Flask enable the creation of dynamic websites.
Data Science:
Libraries such as Pandas, NumPy, and Matplotlib make Python ideal for data manipulation and visualization.
Machine Learning and AI:
Frameworks like TensorFlow, PyTorch, and scikit-learn power AI applications.

Automation and Scripting:
Automates file management, web scraping, and testing tasks.
Game Development:
Libraries like Pygame enable the development of 2D games.
Desktop GUI Applications:
Libraries like Tkinter, PyQt, or Kivy support GUI development.

➢ DJANGO

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It is well-suited for building web applications, including a Boutique Management System, because it provides built-in features like an admin interface, user authentication, and database handling.

Why Use Django for a Boutique Management System?

- **Rapid Development:**

Django allows developers to build a web application quickly due to its extensive set of pre-built features (e.g., authentication, admin interface).

- **Database Integration:**

Django includes an ORM (Object-Relational Mapping) system that allows easy interaction with the database. It handles tasks like adding, retrieving, and deleting records with minimal SQL.

- **Security:**

Django comes with built-in protections against common security threats like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

- **Scalability:**

Django is built to scale, meaning it can handle a growing boutique's needs, from a small shop to a larger enterprise system.

- **Admin Panel:**

Django’s admin panel provides a powerful interface for managing data and content, making it especially useful for boutique managers who need to update product information, view sales data, or manage employees.

Components of the Boutique Management System Using Django

Models:
The models.py file defines the data structures, representing the products, customers, orders, and other entities in the system.

Views:

Views handle the business logic of the application and interact with models to retrieve or update data.

Templates:

Django uses HTML templates to render dynamic content. The template system allows you to easily separate the user interface (UI) from business logic.

URLs:

Django uses a URL dispatcher to map URLs to corresponding views. This is essential for creating specific pages like product listings, order history, or customer management.

Forms:

Django forms handle user input, such as adding new products, registering customers, or processing orders.

CHAPTER 2 : PROPOSED SYSTEM

2.1 ARCHITECTURE OF THE SYSTEM

The proposed Blood Donation Management System aims to streamline and automate the process of blood donation, making it more efficient, transparent, and accessible. This system will serve as a centralized platform for managing donors, recipients, blood banks, and hospital interactions. It addresses the limitations of the current manual or semi-digital systems by offering a user-friendly and reliable digital solution.

Key Features of the Proposed System:

1. **Online Registration for Donors and Recipients:**
    - Donors can register their details, including blood group, contact information, and last donation date.
    - Recipients or hospitals can raise blood requests specifying blood group, quantity, and urgency.
2. **Search and Match Blood Donors:**
    - The system will automatically match available donors based on blood group and geographical proximity.
    - Notifications will be sent to eligible donors when there is a request.
3. **Blood Bank Management:**
    - Tracks the availability of different blood types in real-time.
    - Allows administrators to update blood stock levels and expiry dates.
4. **Appointment Scheduling:**
    - Enables donors to book appointments with blood banks or during donation camps.
    - Sends reminders and confirmations to ensure higher turnout.
5. **Admin Dashboard:**

    - Provides administrative control for monitoring donor activity, managing blood requests, and generating reports.
    - Ensures proper verification and approval of users and blood requests.

6. **Data Security and Privacy:**
    - Ensures the secure handling of user data with proper authentication and role-based access control.
    - Only authorized users can view or modify sensitive information.

7. **Reports and Analytics:**
    - Generates daily, weekly, and monthly reports on donations, requests, and inventory.
    - Helps blood banks and hospitals in planning and resource management.

Advantages of the Proposed System:

- Reduces manual errors and administrative workload.
- Improves donor engagement and response time.
- Provides a real-time view of blood availability.
- Helps save lives by ensuring timely access to blood.

2.2 OBJECTIVE OF SYSTEM

The primary objective of the **Blood Donation Management System** is to develop a centralized and digital platform that facilitates efficient management of blood donations and requests by connecting voluntary blood donors, blood banks, hospitals, and recipients in a seamless and organized manner.

The main objective of this project is to **develop an efficient, reliable, and user-friendly online system** that facilitates the process of blood donation and management. The system will act as a bridge between **donors, recipients, hospitals, and blood banks**, streamlining communication and making the process of donating and receiving blood faster and more effective.

Specific Objectives:

1. **To automate the process** of donor registration, blood request submission, and inventory management.
2. **To create a reliable database** of registered blood donors and their donation history.
3. **To enable real-time matching** of blood donors with recipients based on blood group and location.
4. **To manage and track blood stock** in various blood banks with accurate reporting.
5. **To increase donor participation** through reminders, notifications, and awareness.
6. **To reduce the time and effort** involved in finding suitable blood donors in emergency situations.
7. **To ensure secure access** and maintain the privacy of donor and patient information.

2.3 USER REQUIREMENT

The system will be used by four main types of users: **Donors, Recipients, Blood Bank Administrators,** and **System Admins**. Each user will have specific functionalities and expectations from the system.

1. Functional Requirements (What the system should do):

For Donors:

- Ability to register and create a profile with personal and medical details.
- View eligibility status based on last donation date and health conditions.
- Search and locate nearby blood donation camps or blood banks.
- Receive notifications or alerts for blood requests that match their blood type.
- Schedule or reschedule blood donation appointments.
- View donation history and receive reminders for future donations.

For Recipients / Hospitals:

- Register and create a profile to raise blood requests.
- Submit blood group, quantity required, and urgency level.
- Search available blood types in nearby blood banks.
- Track request status (pending, fulfilled, rejected).
- Receive alerts when a matching donor or stock is found.

For Blood Bank Staff:

- Log in with secure credentials to access admin dashboard.
- Manage blood inventory (add, update, or remove stock).
- View and process blood requests from hospitals or recipients.
- Approve or reject donor registrations based on eligibility.
- Generate reports on stock levels, donations, and requests.

For System Admin:

- Manage all user accounts (add, update, delete).
- Handle roles and permissions for different users.
- Monitor system logs and security activities.
- Generate system-wide reports and analytics.

2. Non-Functional Requirements (How the system should be):

- **Usability:** The system should have a user-friendly interface for easy navigation.
- **Reliability:** The platform should be stable with minimal downtime.
- **Performance:** The system must respond quickly, especially when searching for donors or stock.
- **Security:** Sensitive information (medical data, contact details) must be encrypted and securely stored.
- **Scalability:** The system should support multiple users simultaneously and allow expansion in future.
- **Accessibility:** Users should be able to access the system from mobile and desktop devices.

CHAPTER 4 : USER MANUAL

4.1 USER INTERFACE DESIGN (SCREENS ETC.)

4.2 LIMITATIONS

While the proposed Blood Donation Management System aims to improve and streamline the blood donation and management process, there are certain limitations that need to be considered:

1. Internet Dependency

- The system requires a stable internet connection for accessing features like donor search, request submission, and stock updates. In remote or underdeveloped areas, this could limit usability.

2. Data Accuracy Depends on Users

- The system relies on users (donors, admins, hospitals) to enter accurate and up-to-date information. Any false, incomplete, or outdated data may affect decision-making and emergency responses.

3. No Integration with National Health Databases

- Currently, the system may not be linked with official health records or government databases to automatically verify donor eligibility, health status, or donation frequency.

4. Limited Medical Screening Capability

- The system can only track donor eligibility based on self-declared information (like last donation date or age), not on actual medical tests like hemoglobin level, blood pressure, or infectious diseases.

5. Privacy Risks

- Despite best efforts in securing data, any online system faces the risk of data breaches if not properly maintained or updated against new threats.

6. No Offline Support

- Users cannot access system functionalities or update records without an internet connection, which limits usage in field camps or rural hospitals with poor connectivity.

7. Limited Language Support

- The default system may support only one or two major languages, limiting accessibility for users speaking regional or local dialects.

8. Real-time Donor Availability

- The system may show donors as eligible, but it does not guarantee that they are available or willing to donate at the time of need unless they manually confirm.

9. Initial User Training Required

- Admins and blood bank staff may require some initial training to properly use the system and handle features like inventory management and report generation.

4.3 FUTURE ENHANCEMENT

As technology evolves and user needs grow, the Blood Donation Management System can be upgraded and enhanced with new features to make it more efficient, accessible, and intelligent. Below are some proposed future enhancements:

1. Mobile Application Integration

- Develop dedicated Android and iOS apps to allow users to access all system features from their smartphones.
- Enable real-time notifications and location-based alerts via push notifications.

2. Real-Time Donor Tracking with GPS

- Integrate GPS to allow users to see live locations of nearby available donors and blood banks.
- Improve emergency responses by instantly notifying the closest eligible donors.

3. Integration with Health Systems / Aadhar (India-specific)

- Connect the system with government health databases (like Aadhar-linked health records) for automatic verification of donor identity and eligibility.
- Enhance data authenticity and avoid duplicate entries.

4. AI-Based Donor Matching

- Use artificial intelligence to recommend the best donors based on location, availability, blood group, donation history, and health data.

5. Multi-Language Support

- Introduce support for multiple regional languages to make the system accessible to users across different parts of the country or world.

6. Chatbot Support for Assistance

- Add an AI chatbot to assist users with FAQs, donor registrations, appointment bookings, and real-time support.

7. Blood Transport Tracking System

- Implement a module to track blood units during transport between banks and hospitals to ensure transparency and reduce delays.

8. Donor Reward and Loyalty System

- Introduce reward points, badges, or certificates to motivate and retain regular donors.
- Partner with NGOs or government programs to offer incentives.

9. Automated Health Check Integrations

- Link with health centers to automatically update donor vitals (like hemoglobin, BP) during donation.
- Prevent unfit donations and maintain a safe donor pool.

10. Offline Functionality

- Allow blood banks and mobile camps to collect and manage data offline and sync it once the internet is available.

These enhancements can significantly improve the overall impact of the system, promote regular donations, and save more lives through timely access to blood.

BIBLIOGRAPHY

ANNEXURE: Sample program code

Login.html

```
{% extends "base.html" %}

{% block title %}Hemosys | Log in{% endblock title %}

{% block content %}

<br>

<div class="formbg">

  <br><br><br>

  <div class="d-flex justify-content-center">

    <div class="card bg-light p-5 forms">

      <h2 class="text-dark d-flex justify-content-center">Log in</h2>

      <form action="/auth/login/" method="post">

        {% csrf_token %}

        <br/>

        <label for="id_username">Email</label>

        <div class="form-group">

                <input type="username" class="form-control p-2" 
id="id_username" name="username" required/>

        </div>
        <label for="id_password">Password</label>
        <div class="form-group">
                <input type="password" class="form-control p-2" id="id_password" name="password" required/>
        </div>
        <br/>
        <button type="submit" class="btn btn-dark">Log in</button>
        <br/><br/>
             <span>Forget Password? <a href="/auth/request-reset-email/">Reset</a></span>
        <br/>
        Not User? <a href="/auth/signup/">Sign up</a>
        <br/>
      </form>
    </div>
  </div>
</div>
{% endblock content %}
```

Request.html

```
{% extends "base.html" %}
{% block title %} Hemosys | Request Blood {% endblock title %}
{% block content %}

<div class="card bg-light p-5">
    <br>
    <div class="section-title">
        <h3>Request <span>Blood</span></h3>
    </div>

    <form action="/recipient/request-blood/" method="post"
enctype="multipart/form-data">
        {% csrf_token %}

        <div class="row">
          <div class="col-md-4">
          <label for="recipientname">Patient's name</label>
          <div class="form-group">
            <input type="text" class="form-control p-2" id="recipientname"
name="recipientname" required/>
          </div>
          </div>
          <div class="col-md-4">
          <label for="recipientemail">Guardian's email</label>
          <div class="form-group">
```

```
            <input type="email" class="form-control p-2" id="recipientemail"
name="recipientemail" required/>
          </div>
          </div>
          <div class="col-md-4">
            <label for="recipientphone">Guardian's contact</label>
            <div class="form-group">
              <input type="tel" class="form-control p-2" id="recipientphone"
name="recipientphone" required/>
            </div>
          </div>

        </div>
        <div class="row">
          <div class="col-md-4">
            <label for="recipientage">Patient's age</label>
            <div class="form-group">
              <input type="number" class="form-control p-2" id="recipientage"
name="recipientage" required/>
            </div>
          </div>
          <div class="col-md-4">
            <label for="recipientgender">Patient's gender</label>
            <div class="form-group">
              <select class="form-control" id="recipientgender"
name="recipientgender" requried>
                <option value="">Select gender</option>
        <option id="recipientgender" value="Male">Male</option>
        <option id="recipientgender" value="Female">Female</option>
       </select>
      </div>
     </div>
     <div class="col-md-4">
     <label for="recipientlocation">Patient's location</label>
     <div class="form-group">
      <input type="text" class="form-control p-2" id="recipientlocation"
name="recipientlocation" required/>
     </div>
     </div>
    </div>



    <div class="row">
    <div class="col-md-4">
     <label for="recipientcondition">Donation cause</label>
     <div class="form-group">
      <input type="text" class="form-control p-2" id="recipientcondition"
name="recipientcondition" required/>
     </div>
    </div>
    <div class="col-md-4">
    <label for="recipientblood">Patient's blood group</label>
    <div class="form-group">
```

```
          <select class="form-control" id="recipientblood" name="recipientblood"
requried>
            <option value="">Select blood group</option>
            <option id="bloodA+" value="A+">A+</option>
            <option id="bloodA-" value="A-">A-</option>
            <option id="bloodB+" value="B+">B+</option>
            <option id="bloodB-" value="B-">B-</option>
            <option id="bloodAB+" value="AB+">AB+</option>
            <option id="bloodAB-" value="AB-">AB-</option>
            <option id="bloodO+" value="O+">O+</option>
            <option id="bloodO-" value="O-">O-</option>
          </select>
        </div>
        </div>
        <div class="col-md-4">
        <label for="recipientdonationtype">Donation type</label>
        <div class="form-group">
          <select class="form-control" id="recipientdonationtype"
name="recipientdonationtype" requried>
            <option value="">Select type</option>
            <option id="red-blood" value="Red Blood">Red Blood</option>
            <option id="platelet" value="Platelet">Platelet</option>
            <option id="plasma" value="Plasma">Plasma</option>
          </select>
        </div>
        </div>
        </diV>
```

```
    <div class = "row">
      <div class="col-md-4">
        <label for="recipientdonationquantity">Donation quantity</label>
        <div class="form-group">
          <input type="number" class="form-control p-2" id="recipientdonationquantity" name="recipientdonationquantity" placeholder = "Bag" required/>
        </div>
      </div>
      <div class="col-md-4">
        <label for="recipientdate">Donation date</label>
        <div class="form-group">
          <input type="date" class="form-control p-2" id="recipientdate" name="recipientdate" required/>
        </div>
      </div>
      <div class="col-md-4">
        <label for="recipienttime">Donation time</label>
        <div class="form-group">
          <input type="time" class="form-control p-2" id="recipienttime" name="recipienttime" required/>
        </div>
      </div>
    </div>

    <br>
```

```
        <div class="d-flex justify-content-center">
          <button type="submit" class="btn btn-dark" style="background-color: #e43c5c">Submit</button>
        </div>
      </form>
</div>

{% endblock content %}
```