NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES - FAST

Project Report CS-2005
Database System

Teacher: Sir Shoaib Raza
Section: BS(CS)-5F
Semester: 5th semester Fall 2023
Blood Bank Management System

Member 1: 21K-4568 Muhammad Ali
Member 2: 21K-3175 Syed Muhammad Taqi Baquer
Member 3: 21K-3267 Muhammad Saad

Introduction

In an era where technological advancements play a pivotal role in transforming healthcare systems, we have tried to enhance the efficiency and accessibility of blood donation services. Built on the powerful trio of Node.js, HTML, CSS, and JavaScript, this innovative system redefines the way we approach blood donation, effortlessly connecting donors, recipients, and administrators in a user-friendly digital environment.

Problem

Despite the critical importance of blood donation, the existing blood bank management systems has a loop-hole. Traditional approach is usually slow to address the dynamic nature of blood supply and demand. The absence of a unified platform often results in communication gaps between donors, recipients, and administrators, leading to potential delays in critical situations. The need of for centralized blood bank is evident keeping in mind the loop-holes of traditional blood bank system. Providing user friendly interface which interacts with user taking input for donor, recipient and admin.

Objectives

1. **System overview:** Blood bank contains admin who will monitor donor and recipient.
2. **Technological frameworks:** Node.js, HTML, CSS, and JavaScript is used in creating a robust and user-friendly system.
3. **User roles and features:** Donor will register as a donor and recipient who is need of blood will register as a recipient and admin will accept or reject accordingly.
4. **Data management and security:** Ensures robust measure to secure sensitive information of donor and recipient which is only visible to admin.
5. **User interface and experience:** An intuitive interface for seamless user interaction and smooth data entry.
6. **Insightful reporting:** Holds a record and maintain logs which contains all the transactions.

Methodologies

We designed a user-friendly interface that accommodates the diverse needs of donors, recipients, and administrators. This approach aimed to make interactions intuitive and efficient, promoting widespread system adoption. The project integrated Agile methodologies for feasible development Usability testing played a crucial role in refining the user interface and overall system functionality, we as a team member reviewed it as a user and worked on the improvement.

System and Environment:

1. **Database Interface**
   The system interfaces with a MySQL database server via XAMPP, hosted locally. The database server manages donor recipient and admin and is facilitated through SQL queries, with communication streamlined using the Node.js MySQL library.

2. **Frontend Interface**
   The frontend of the system is crafted using HTML, CSS, and JavaScript, providing an intuitive user interface. This interface communicates with the backend through API calls, utilizing JavaScript's fetch API for seamless HTTP requests. The system ensures compatibility with contemporary web standards, leveraging HTML5, CSS3 and JavaScript.

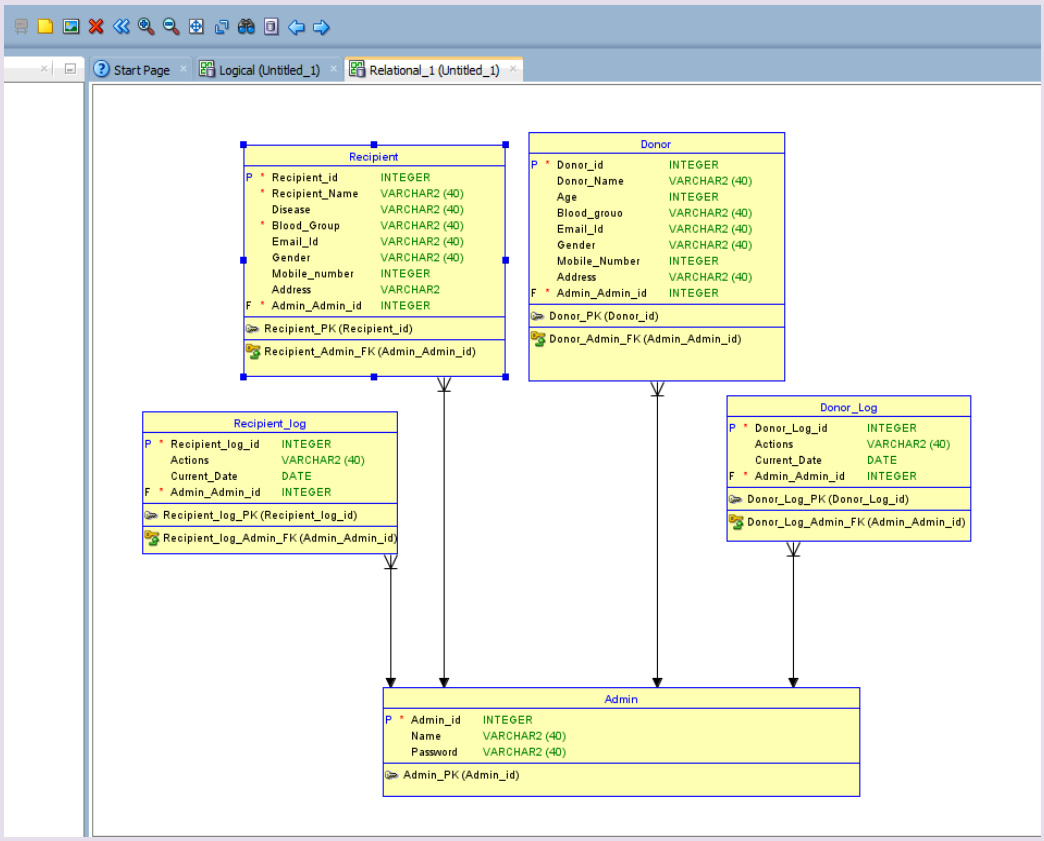
3. **Backend interface**
   The backend infrastructure is developed using JavaScript and Node.js, a runtime for server-side JavaScript. The system exposes RESTful APIs that the frontend utilizes for various functionalities like data retrieval, donor registration, recipient registration, updating (accept or reject).

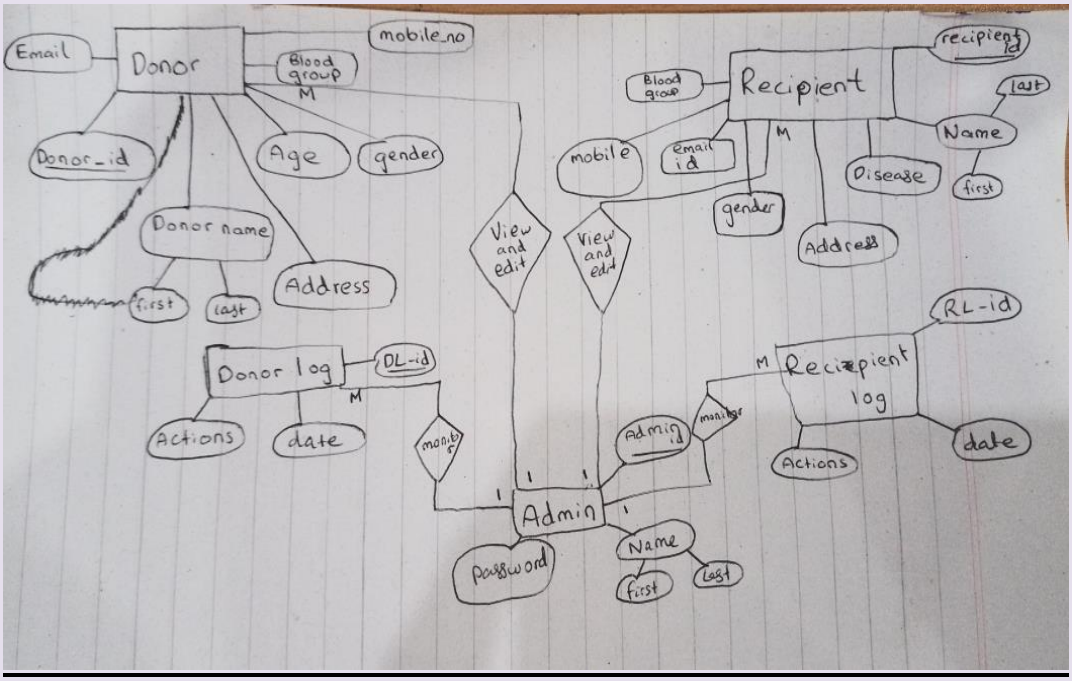
Communications Interfaces

1. **API Communication:** Communication between the frontend and backend of the Blood Bank Management System occurs through RESTful APIs. The system exchanges data in JSON format, adhering to REST principles to ensure stateless communication. This approach not only enhances scalability but also promotes maintainability, allowing for efficient interaction between different systems components.
2. **Security measures:** To safeguard the confidentiality and integrity of data during communication, the Blood Bank Management System implements HTTPS for all interactions between the client and server. SSL/TLS protocols are employed to provide robust encryption, ensuring the secure transmission of sensitive information. This security measure is fundamental in maintaining the trustworthiness of the blood bank system.

3. **Data transferring rates:** The system is optimized for efficient data transfer rates between the frontend and backend, ensuring a responsive user experience. Designed to handle concurrent requests and responses, the Blood Bank Management System prioritizes swift and reliable communication.

ER Diagram


Start Page
Logical (Untitled_1)
Relational_1 (Untitled_1)
Recipient
P * Recipient_id INTEGER
* Recipient_Name VARCHAR2 (40)
Disease VARCHAR2 (40)
* Blood_Group VARCHAR2 (40)
Email_Id VARCHAR2 (40)
Gender VARCHAR2 (40)
Mobile_number INTEGER
Address VARCHAR2
F * Admin_Admin_id INTEGER
Recipient_PK (Recipient_id)
Recipient_Admin_FK (Admin_Admin_id)
Donor
P * Donor_id INTEGER
Donor_Name VARCHAR2 (40)
Age INTEGER
Blood_grouo VARCHAR2 (40)
Email_Id VARCHAR2 (40)
Gender VARCHAR2 (40)
Mobile_Number INTEGER
Address VARCHAR2 (40)
F * Admin_Admin_id INTEGER
Donor_PK (Donor_id)
Donor_Admin_FK (Admin_Admin_id)
Recipient_log
P * Recipient_log_id INTEGER
Actions VARCHAR2 (40)
Current_Date DATE
F * Admin_Admin_id INTEGER
Recipient_log_PK (Recipient_log_id)
Recipient_log_Admin_FK (Admin_Admin_id)
Donor_Log
P * Donor_Log_id INTEGER
Actions VARCHAR2 (40)
Current_Date DATE
F * Admin_Admin_id INTEGER
Donor_Log_PK (Donor_Log_id)
Donor_Log_Admin_FK (Admin_Admin_id)
Admin
P * Admin_id INTEGER
Name VARCHAR2 (40)
Password VARCHAR2 (40)
Admin_PK (Admin_id)


Email
Donor
mobile_no
Blood group
M
Donor_id
Age
gender
Donor name
first
last
Address
Donor log
DL-id
M
Actions
date
monitor
View and edit
View and edit
Admin
Password
Name
first
last
Admin_id
monitor
Recipient
recipient id
Blood group
last
Name
mobile
Email id
M
Disease
first
gender
Address
RL-id
M
Recipient log
Actions
date

Normalization

1: Functional Dependencies:

Admin_id ---> A_name, A_password

Donor_id ---> D_name , D_email, D_age, D_gender, D_bloodgroup, D_mobileno, D_address

Recipient_id ---> R_name , R_email, R_disease, R_gender, R_bloodgroup, R_mobileno, R_address

DonorLog_id ---> DL_action, DL_date

RecipientLog_id ---> RL_action, RL_date

2: Primary Keys:

Admin_id, Donor_id, Recipient_id, DonorLog_id, RecipientLog_id

1. **Admin Table (Assuming each admin has a unique ID):**

   Admin (Admin_id, A_name, A_password)

   This table is already in 1NF and 2NF since there are no repeating groups or partial dependencies. Admin_id serves as the primary key.

2. **Donor Table:**

   Donor (Donor_id, D_name, D_email, D_age, D_gender, D_bloodgroup, D_mobileno, D_address)

   This table is already in 1NF as each attribute contains atomic values. To achieve 2NF, we need to make sure that there are no partial dependencies. The primary key is Donor_id.

3. **Recipient Table:**

   Recipient (Recipient_id, R_name, R_email, R_disease, R_gender, R_bloodgroup, R_mobileno, R_address)

   This table is already in 1NF as each attribute contains atomic values. To achieve 2NF, we need to make sure that there are no partial dependencies. The primary key is Recipient_id.

4. **DonorLog Table:**

   DonorLog (DonorLog_id, Donor_id, DL_action, DL_date)

This table is already in 1NF as each attribute contains atomic values. To achieve 2NF, we need to make sure that there are no partial dependencies. The primary key is DonorLog_id.

5. **RecipientLog Table:**

   RecipientLog (RecipientLog_id, Recipient_id, RL_action, RL_date)

   This table is already in 1NF as each attribute contains atomic values. To achieve 2NF, we need to make sure that there are no partial dependencies. The primary key is RecipientLog_id.

   Now, let's check for 3NF:

   In each table, there should be no transitive dependencies. That means every non-prime attribute must be non-transitively dependent on every superkey.

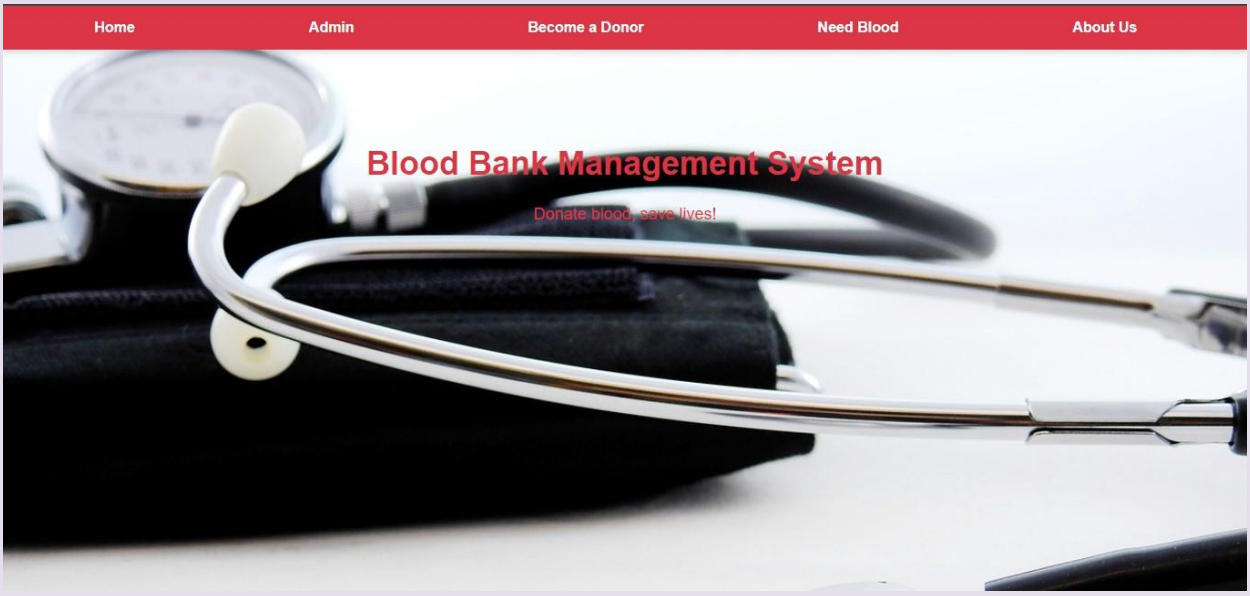
   No transitive dependencies are present in the current tables, so they

   are in 3NF.

   The tables are now normalized up to 3NF.

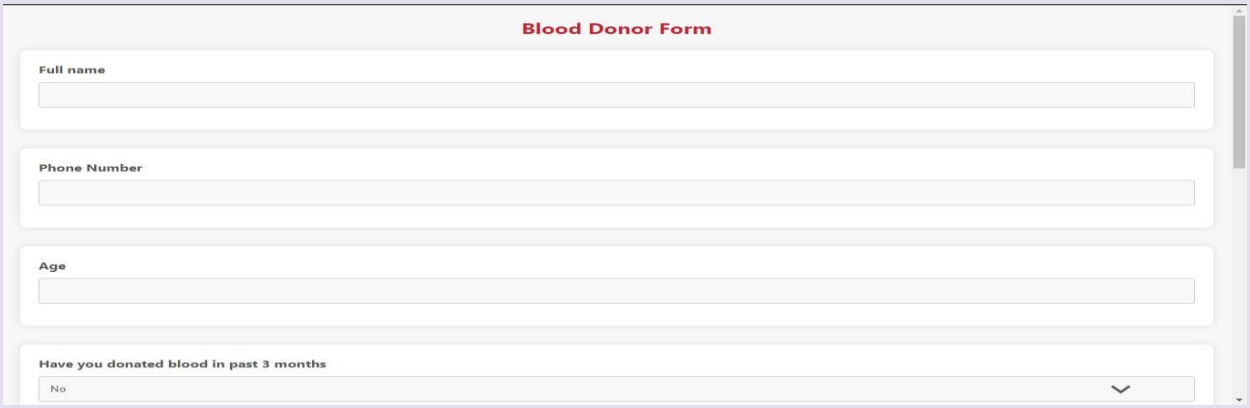
Project Output

User Home Page:



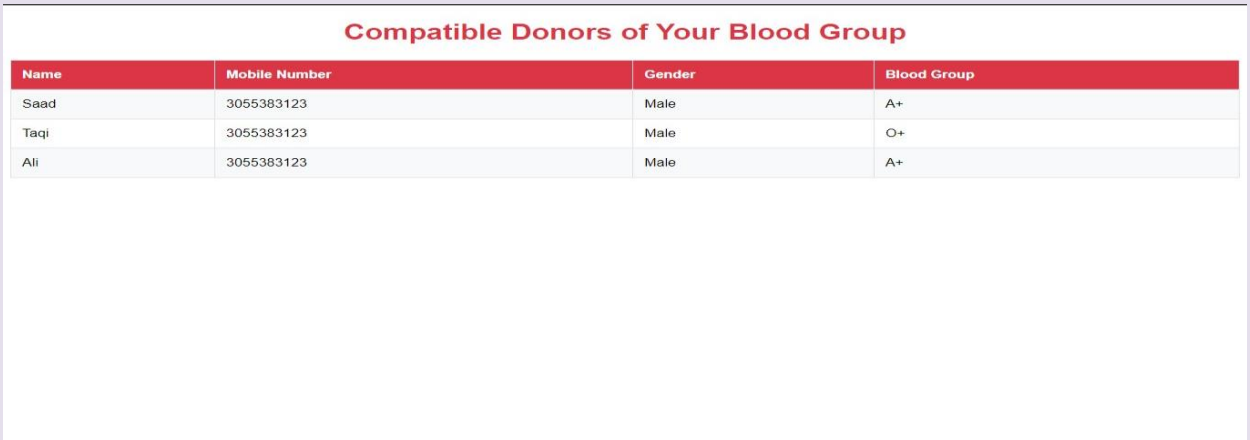
About Us:

Donor Form:



Recipient Form:

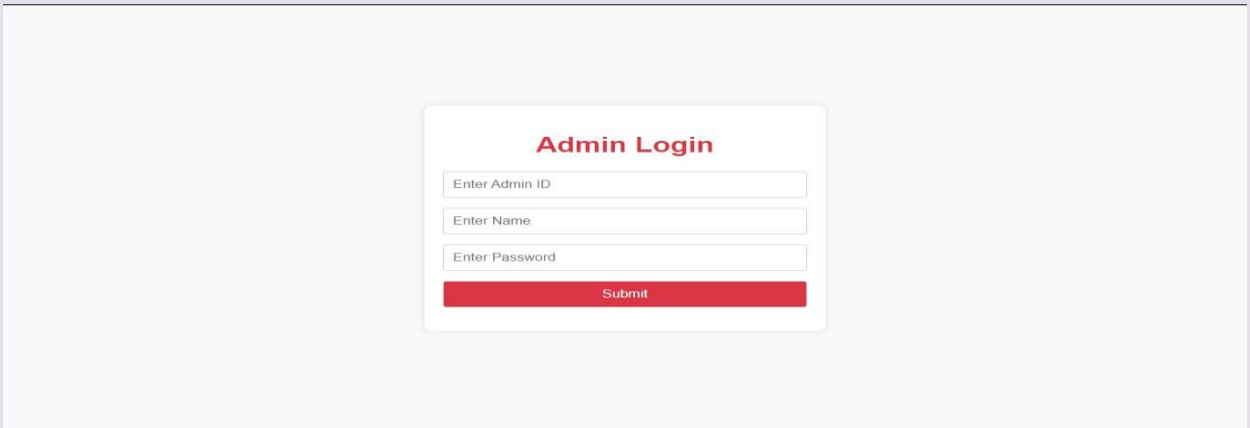
Compatible Donor of your Blood:



Compatible Donors of Your Blood Group

Name	Mobile Number	Gender	Blood Group
Saad	3055383123	Male	A+
Taqi	3055383123	Male	O+
Ali	3055383123	Male	A+

Admin Login Page:



Donor List:

Donors List

S.No	Name	Mobile Number	Age	Period	Question	Gender	Disease	Blood Group	Date	Action
15	Saad	3055383123	18	no	yes	Male	no	A+	Thu Dec 07 2023 08:38:04 GMT+0500 (Pakistan Standard Time)	Delete
16	Taqi	3055383123	20	no	yes	Male	no	O+	Thu Dec 07 2023 08:39:58 GMT+0500 (Pakistan Standard Time)	Delete
17	Ali	3055383123	20	no	yes	Male	no	A+	Thu Dec 07 2023 08:52:45 GMT+0500 (Pakistan Standard Time)	Delete

Recipient List:

Recipient List

S.No	Name	Mobile Number	Gender	Reason	Blood Group	Date	Action
97	taqi	3055383123	Male	None	A+	Thu Dec 07 2023 08:38:16 GMT+0500 (Pakistan Standard Time)	Delete
98	Saad	3055383123	Male	other	AB+	Thu Dec 07 2023 08:40:29 GMT+0500 (Pakistan Standard Time)	Delete
99	Saad	3055383123	Male	other	A+	Thu Dec 07 2023 08:45:26 GMT+0500 (Pakistan Standard Time)	Delete
100	Zain	3043383648	Male	other	O-	Thu Dec 07 2023 08:53:54 GMT+0500 (Pakistan Standard Time)	Delete
101	Anas	3055383123	Male	medical reason	AB+	Thu Dec 07 2023 08:54:49 GMT+0500 (Pakistan Standard Time)	Delete

Recipient Logs:

Recipient Logs

S.No	Recipient ID	Action	Date	Action
143	99	deleted	Thu Dec 07 2023 08:56:40 GMT+0500 (Pakistan Standard Time)	RollBack

Donor Logs:

Donor Logs

S.No	Donor ID	Action	Date	Action
22	17	deleted	Thu Dec 07 2023 08:57:12 GMT+0500 (Pakistan Standard Time)	RollBack