A

Project Report

On

Blood Bank & Donor Management System

By :

Kayum Parmar(BCA)

Semester – 5

Project Guide :

PROF. SHREYA DOSHI

Submitted To :

Geetanjali College Of Computer

Science And Commerce (B.B.A). Rajkot.

Academic Year : 2024-2025

ACKNOWLEDGEMENT

I Am Happy To Submit My Idea Of " Blood Bank & Donor Management System " Application In Saurashtra University, Rajkot For BCA Degree In Computer Branch.

We take this occasion to thank God, almighty for blessing us with his grace and taking our Endeavour to a successful culmination. We extend our sincere and heartfelt thanks to our esteemed guide, PROF. SHREYA DOSHI for providing us with the right guidance and advice at the crucial junctures and for showing us the right way. We would like to thank the other faculty members also, at this occasion. Last but not the least, we would like to thank friends for the support and encouragement they have given us during the course of our work.

INTRODUCTION

System Development:

The process of building systems has always been complex with system becoming larger, the costs and complexities get multiplied. So the need for better methods for developing systems is widely recognized to be effective and the applied model should meet a few basic requirements.

- The model should be structured and cover the entire system development process from feasibility study to programming, testing and implementation.
- The model should utilize established methods and techniques like database designs, normalizations and structured programming techniques.
- The model should consist of building blocks, which define tasks, results and interfaces.
- The model should separate the logical system from the physical system.
- Documentation should be a direct result of the development work and should be concise, precise and as non-redundant as possible.

Based on the above requirements of the system model, system study has been made. Various methodologies have been applied for system study, evolving design documents, data modeling, input screen design and report design.

Project:

The persons who like to donate blood registers in my site. The persons in need of blood searches for the persons having the same blood group and within the city. If he found a donor in his city then he gets the total details of the donor, if he doesn't find any donor then he is given the contact numbers and addresses of the Life Saving Contact Persons for major cities. In this projects we use PHP and Mysql and it contains two modules i.e Admin and Donor.

Admin Module

Dashboard: In this section, admin can view all the details in brief like total blood group listed, registered donor list and total enquiries received.

Blood Group: In this section, admin can manage blood group(Add/ Delete).

Donor List: In this section, admin can view list of donor and have right to delete and hide the detail of donor.

Manage Contact us Query: In this section, admin can manage query which is received by users.

Manage Pages: In this section, admin can website pages.

Update Contact info: In this section, admin can update the contact details of website.

Request Received by Donor: In this section, admin can view the request of blood which is received by donor.

Admin can also update his profile, change the password and recover the password.

Donor Module

Home: Its is welcome page for users and donor. If any users want to donate the blood they must register with us.

About Us: Users can view the about us page.

Contact Us: Users can contact with admin the through contact us page.

Donor List: Users can view and contact with donor.

Search Donor: Users can search the donor according to city and blood group.

Registered Users(Donor)

Home: Its is welcome page for users and donor. If any users want to donate the blood they must register with us.

About Us: Users can view the about us page.

Contact Us: Users can contact with admin the through contact us page.

Donor List: Users can view and contact with donor.

Search Donor: Users can search the donor according to city and blood group.

My Account:

- Profile
- Request Received
- Logout

Donor can also update his profile, change the password and recover the password.

Software Environment

The environment specifies the software tools and frameworks required for development, operation, and maintenance of the BBDMS application.

Technologies Used:

- **HTTP**: Communication protocol for transferring data over the web.
- **PHP**: Server-side scripting language for dynamic content generation.
- **MySQL**: Relational database management system.
- **HTML**: Markup language for creating web page structure.
- **JavaScript**: Client-side scripting for interactive features.

Technology Breakdown:

1. HTTP

- **Definition**: Hypertext Transfer Protocol is a stateless, connectionless protocol used to communicate between a client (web browser) and a server. It operates over TCP/IP and follows a request/response paradigm.
- **Operation**: HTTP opens a connection, sends a request, receives a response, and then closes the connection. This is efficient for supporting many users but incurs performance penalties with every new connection.

2. PHP

- **What is PHP?**: PHP is a widely-used server-side scripting language ideal for building dynamic web applications.
- **Capabilities**:
    - Dynamic content generation.
    - Handling form data.
    - Managing user authentication and session control.
    - Working with databases such as MySQL.
    - File operations (create, read, write).

- **Why PHP?**: It is open-source, cross-platform, and compatible with many web servers and databases.

3. MySQL

- **What is MySQL?**: A popular, open-source relational database management system, known for being reliable, secure, and compatible with many applications.
- **Capabilities**:
    - Data Definition Language (DDL): Used to define structures (tables, databases).
    - Data Manipulation Language (DML): Used to manage data (insert, delete, update).
    - Data Control Language (DCL): Used to control access to data (grant, revoke).
- **Why MySQL?**: It is fast, secure, and supports a wide range of platforms.

4. HTML

- **Definition**: HyperText Markup Language is used to structure web pages.
- **Why HTML?**: It ensures the application is accessible across different platforms and browsers.
- **Features**:
    - Supports multimedia, hyperlinks, and forms.
    - Works well with CSS and JavaScript for enhanced functionality.

5. JavaScript

- **What is JavaScript?**: A lightweight, object-based scripting language for creating dynamic, client-side content.
- **Capabilities**:
    - Enhances web interactivity.
    - Handles client-side form validation, event handling, and DOM manipulation.

- **Why JavaScript?**: It’s platform-independent and works across various browsers, providing rich user experiences.

Cascading Style Sheets (CSS):

CSS is used to ensure a consistent look and feel throughout the website. By separating content from presentation, it allows for easier updates and better design consistency.

Conclusion:

This combination of technologies ensures that the BBDMS system is highly scalable, efficient, and user-friendly. The system can manage large volumes of data and users while providing dynamic interactions and secure database management.

WORKING ENVIRONMENT

Hardware Configuration:

Processor	:	P III 700 MHz
RAM	:	64 MB RAM
Hard Disk Drive	:	20 GB HDD
Keyboard	:	104 keys
Mouse	:	Logitech Mouse
Monitor	:	15” digital color monitor
Display Type	:	VGA

Software Configuration:

Operating System	:	Windows xp, Vista,7 or any one
Web server	:	Apache
Web Browser	:	Chrome/Mozilla/Internet Explorer5.0
Designing Tool	:	Dreamweaver3.0, HTML
Server Side Scripting	:	PHP
Client Side Scripting	:	MYSQL

System Design

The **System Design** phase is crucial in defining the architecture, components, modules, interfaces, and data that the system will handle. The design serves as the blueprint for system development, ensuring a clear pathway from requirements to implementation. Below is a refined system design breakdown for BBDMS:

---

1. Introduction to System Design

- **System Design**: The blueprint that specifies the structure, behavior, and overall functioning of a system. This phase follows system analysis and bridges the gap between analysis and implementation.
- **Key Concepts**:
    - **Abstraction**: Simplifies complex systems by focusing on the essential features while hiding the intricate details.
    - **Modularity**: Breaks down the system into smaller, manageable modules with defined interfaces, which simplifies debugging, testing, and maintenance.
    - **Verification**: Ensures the design satisfies both the requirements analysis and the end-user needs.

---

2. Essential Design Factors

- **Reliability**: The system should perform as expected under normal conditions, adhering strictly to specifications.
- **Extensibility**: The system should be adaptable to changes in requirements or specifications.
- **Reusability**: Modular design should allow parts of the system to be reused in other projects.

---

3. Conceptual and Detailed Design

Conceptual Design (High-Level Design):

- Specifies the **externally observable characteristics** of the system, such as:
    - User interfaces
    - External data sources and destinations
    - Functional characteristics
    - System flow and data handling

Detailed Design (Low-Level Design):

- Focuses on the **internal structure** of the system, providing details for:
    - Module interactions
    - Data handling and storage mechanisms
    - Error handling, security, and performance tuning

---

4. Preliminary Design

- Focuses on an overall picture of the system's modules and sub-modules.
- **Tools for Preliminary Design**:
    - **Data Flow Diagrams (DFD)**: Illustrate the flow of data through the system.
    - **Entity-Relationship (ER) Diagrams**: Represent the database's logical structure.
- Modules and sub-modules are designed with the principles of **Cohesion** (keeping related functionality together) and **Coupling** (minimizing dependencies between modules).

---

5. Code Design

- **Purpose**: To identify and retrieve information efficiently.
- **Characteristics**:
    - Codes should be **unique**, concise, and expandable.
    - They should follow a standard format for ease of storage and retrieval.
    - Codes should be easily understandable, versatile, stable, and meaningful.

---

6. Input Design

- The primary objectives of input design:
    - **Cost-effective input methods**.
    - **Accuracy** of data entry.
    - **User-friendliness**: The system should be easy to use for the staff.
- **User Interface**: Input screens are designed to be intuitive, minimizing the risk of incorrect data entry.
    - **Pushbuttons and shortcuts**: Enhance the user experience and simplify operations.

---

7. Output Design

- **Outputs** communicate the results of system processing to users or external entities.
- **Types of Outputs**:
    - **External Outputs**: Delivered to users or entities outside the organization, like donor reports or statistics.
    - **Internal Outputs**: Used by staff within the organization for internal decision-making or operations.
    - **Operation Outputs**: Include system logs, usage statistics, and error reports used by system administrators.

- **Interactive Outputs**: Involve real-time communication between users and the system, such as live data reports or user prompts.

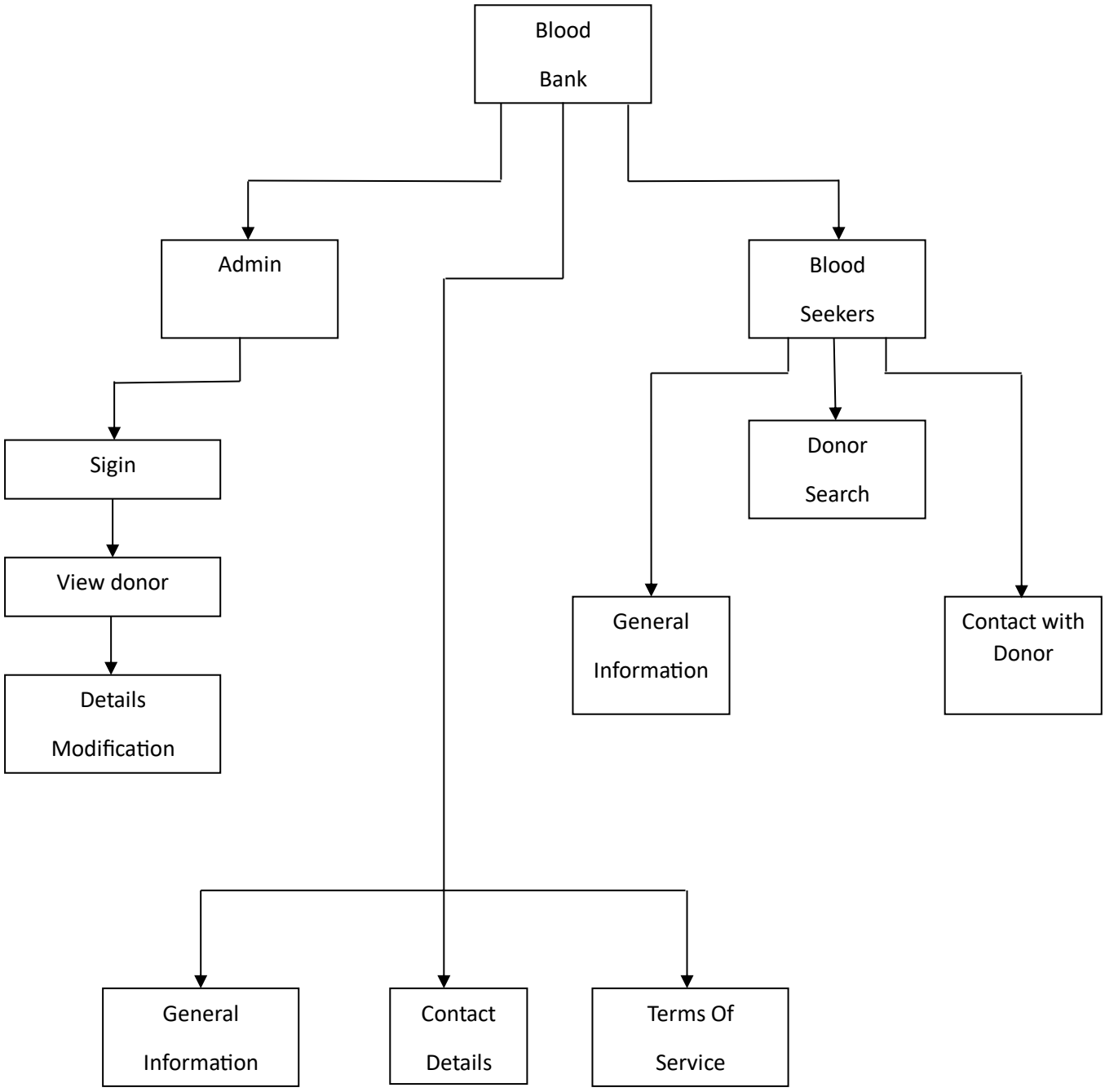
---

8. Design Concepts Recap

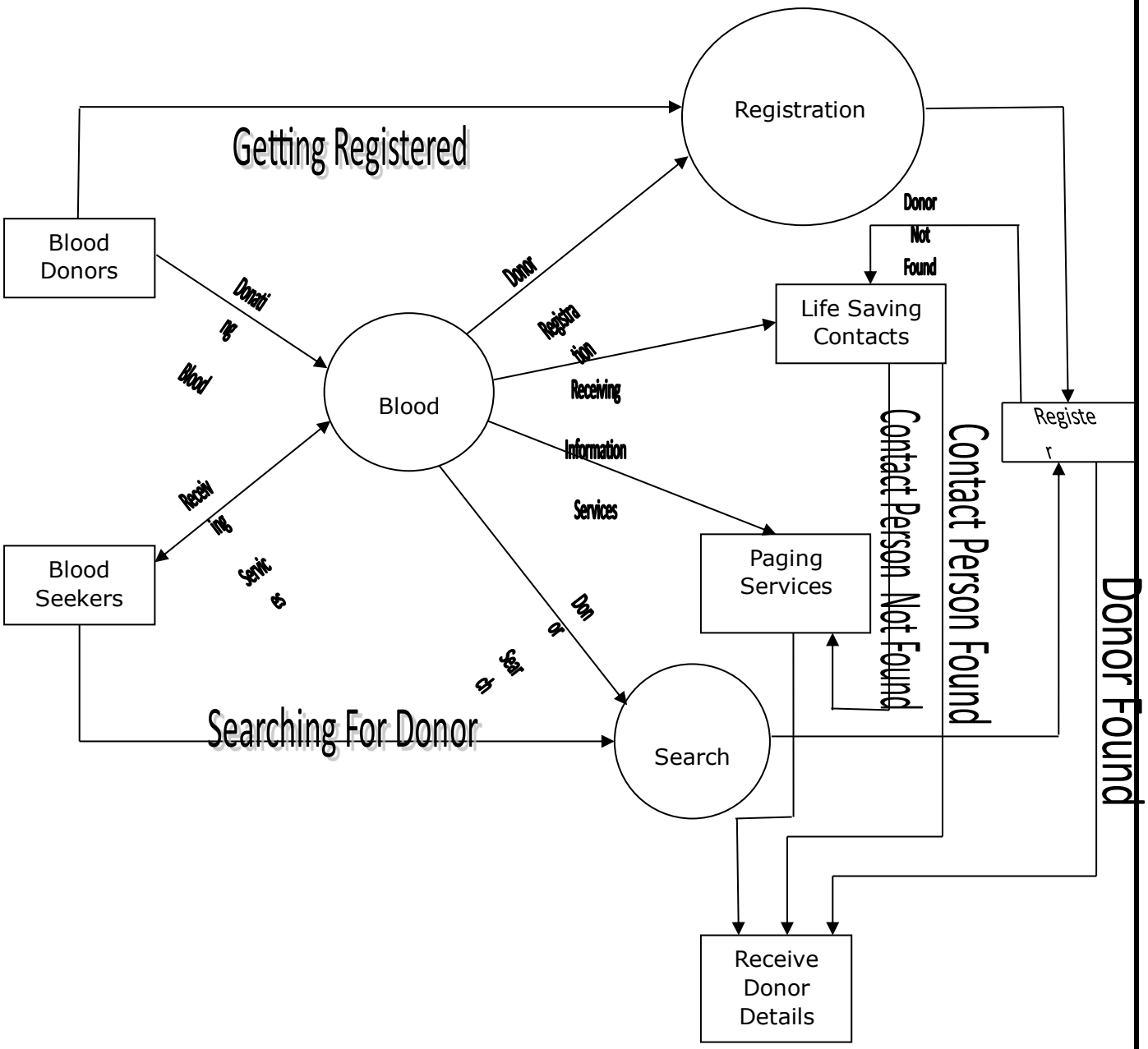
- **Abstraction**: Simplifies complex systems by focusing on the key functionality.
- **Modularity**: Divides the system into independent modules to improve maintainability.
- **Verification**: Ensures the system meets the requirements and functions correctly during implementation.

System Design Tools :

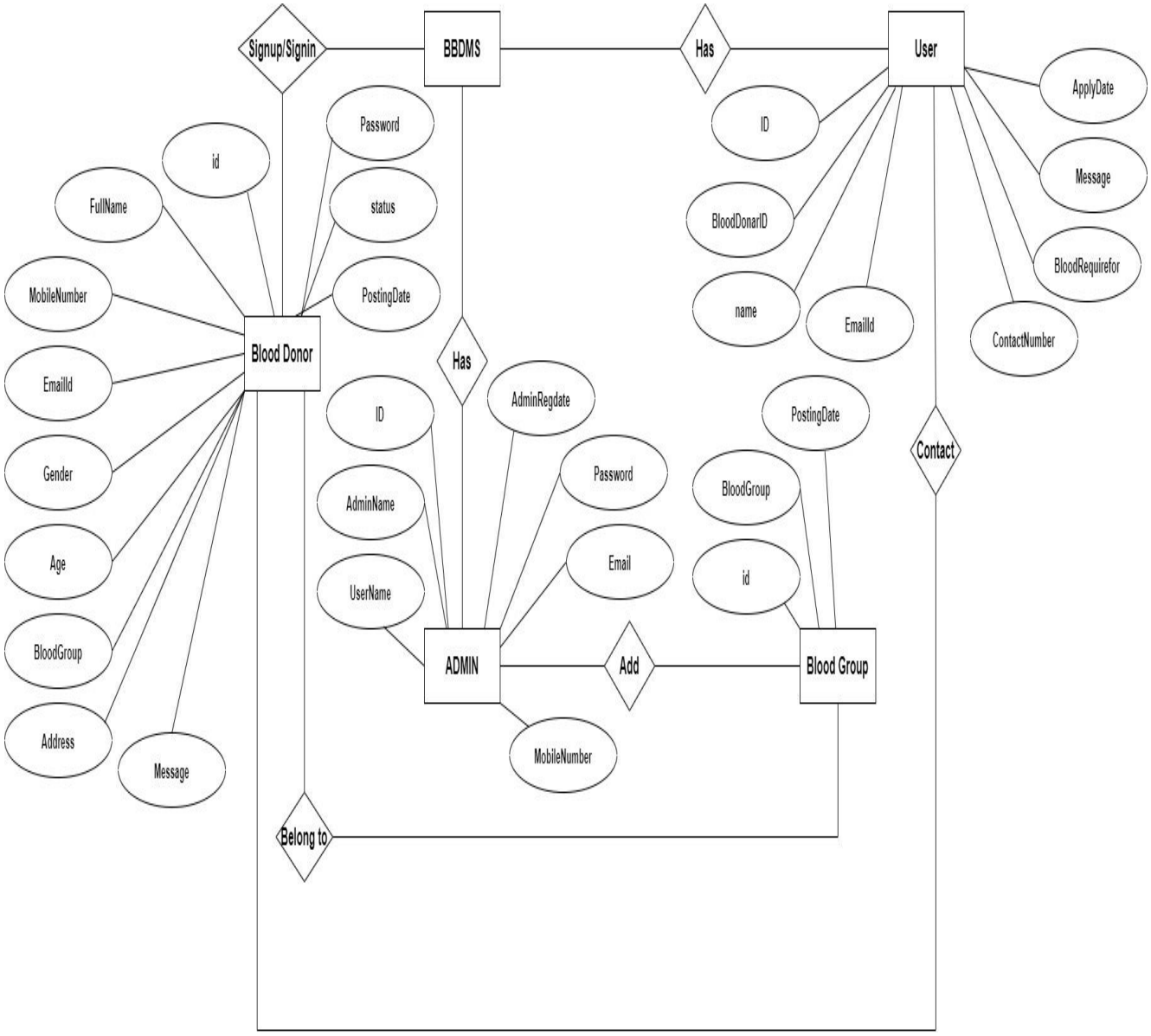
SYSTEM FLOW DIAGRAM



Data flow Diagrams :



ER Diagram :


Signup/Signin
BBDMS
Has
User
ApplyDate
Message
BloodRequirefor
ContactNumber
EmailId
name
BloodDonarID
ID
Password
status
PostingDate
id
FullName
MobileNumber
EmailId
Gender
Age
BloodGroup
Address
Message
Blood Donor
Has
ID
AdminRegdate
Password
Email
AdminName
UserName
ADMIN
MobileNumber
Add
Blood Group
PostingDate
BloodGroup
id
Contact
Belong to

Data Dictionary

- **Admin Table :**

#	Name	Type	Collation	Null	Default	Extra
1	**id**	int(11)		No	None	AUTO_INCREMENT
2	**AdminName**	varchar(50)	utf8mb4_general_ci	No	None	
3	**UserName**	varchar(50)	utf8mb4_general_ci	No	None	
4	**MobileNumber**	varchar(11)	utf8mb4_general_ci	No	None	
5	**Email**	varchar(50)	utf8mb4_general_ci	No	None	
6	**Password**	varchar(50)	utf8mb4_general_ci	No	None	

- **Blood-Group:**

#	Name	Type	Collation	Null	Default	Extra
1	**id**	int(11)		No	None	AUTO_INCREMENT
2	**AdminName**	varchar(50)	utf8mb4_general_ci	No	None	
3	**UserName**	varchar(50)	utf8mb4_general_ci	No	None	

- **Blooddonars Table:**

#	Name	Type	Collation	Null	Default	Extra
1	**id**	int(11)		No	NULL	AUTO_INCREMENT
2	**fname**	varchar(50)	utf8mb4_general_ci	Yes	NULL	
3	**number**	varchar(50)	utf8mb4_general_ci	Yes	NULL	
4	**email**	varchar(11)	utf8mb4_general_ci	Yes	NULL	
5	**age**	varchar(50)		Yes	NULL	
6	**gender**	varchar(50)	utf8mb4_general_ci	Yes	NULL	
7	**blood_group**	varchar(50)	utf8mb4_general_ci	Yes	NULL	
8	**date-of-birth**	date		Yes	NULL	
9	**address**	varchar(500)	utf8mb4_general_ci	Yes	NULL	
10	**date-time**	datetime(6)		Yes	NULL	
11	**state**	varchar(50)	utf8mb4_general_ci	Yes	NULL	
12	**password**	varchar(50)	utf8mb4_general_ci	Yes	NULL	

- **Bloodrequirer Table:**

#	Name	Type	Collation	Null	Default	Extra
1	**id**	int(11)		No	None	AUTO_INCREMENT
2	**BloodDonarid**	varchar(50)		Yes	Null	
3	**name**	varchar(50)	utf8mb4_general_ci	Yes	Null	
4	**email**	varchar(11)	utf8mb4_general_ci	Yes	Null	
5	**number**	varchar(50)	utf8mb4_general_ci	Yes	Null	
6	**bloodrequirefor**	varchar(50)	utf8mb4_general_ci	Yes	Null	
7	**message**	mediumtext	utf8mb4_general_ci	Yes	Null	
8	**ApplyDate**	timestamp		Yes	current_timestamp()	

- **Contactus Table:**

#	Name	Type	Collation	Null	Default	Extra
1	id	int(11)		No	None	AUTO_INCREMENT
2	**name**	varchar(50)	utf8mb4_general_ci	No	None	
3	**number**	varchar(10)	utf8mb4_general_ci	No	None	
4	**email**	varchar(50)	utf8mb4_general_ci	No	None	
5	**message**	text		No	None	
6	**Posting_Date**	timestamp	utf8mb4_general_ci	No	current_timestamp()	

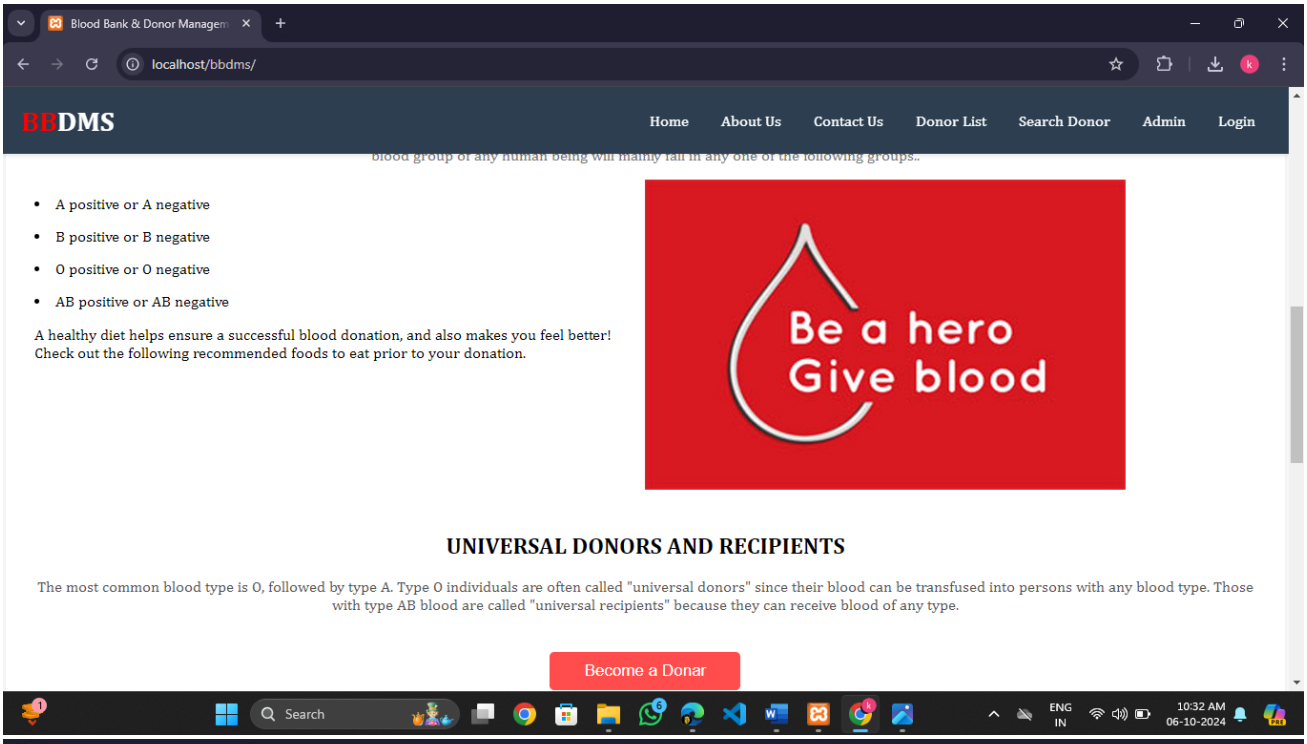
- **Contactusinfo Table:**

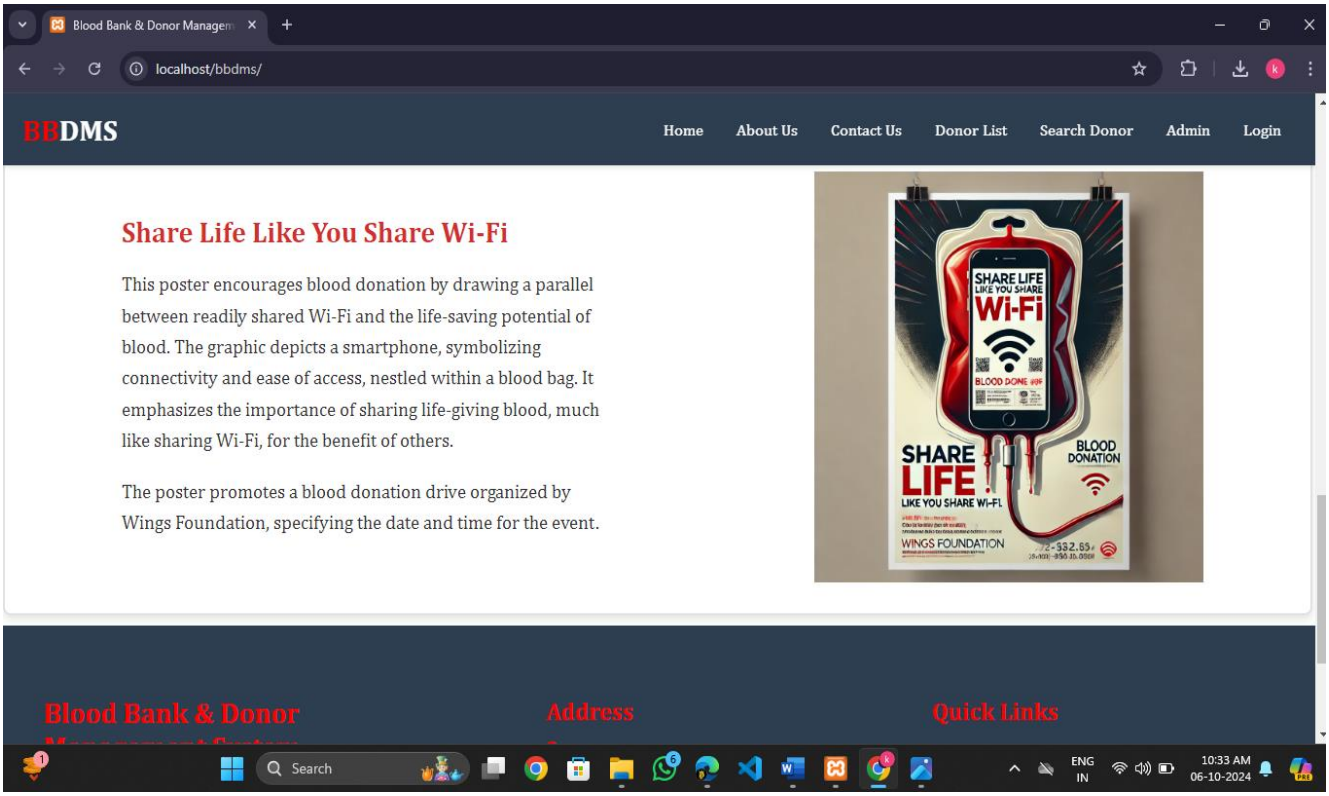
#	Name	Type	Collation	Null	Default	Extra
1	id	int(11)		No	None	AUTO_INCREMENT
2	**Address**	varchar(50)	utf8mb4_general_ci	No	None	
3	**Number**	varchar(50)	utf8mb4_general_ci	No	None	
4	**Email**	varchar(50)	utf8mb4_general_ci	No	None	

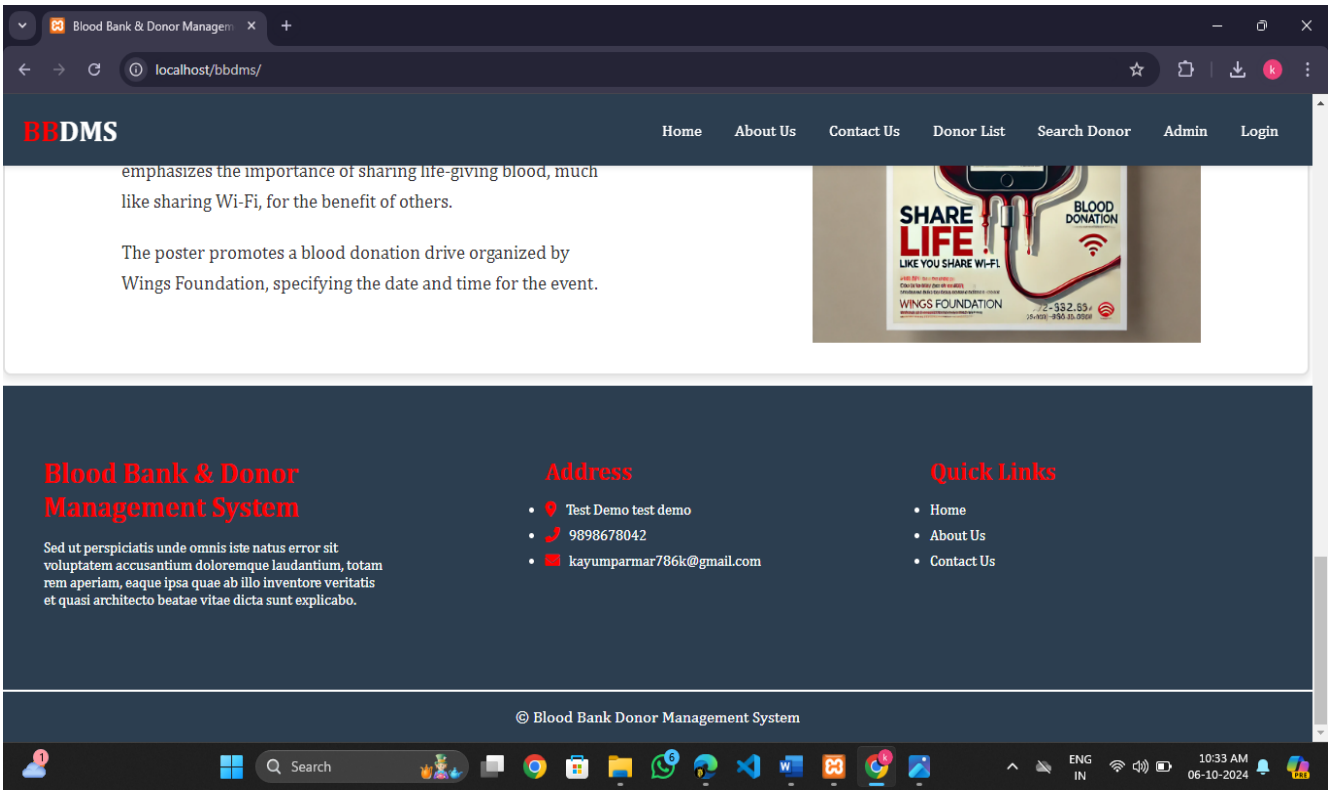
SCREENS (Project Screen Shots)

User Penel:

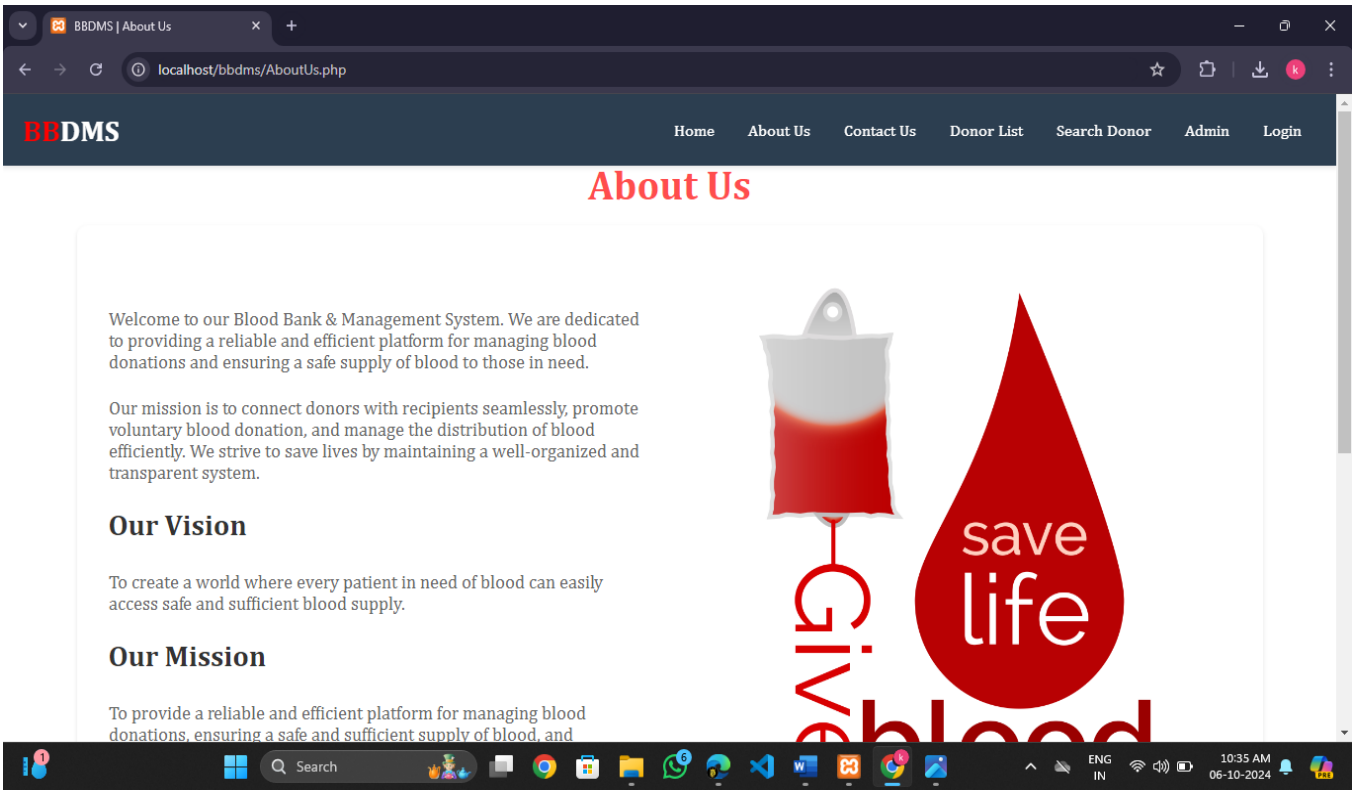
Home Page :

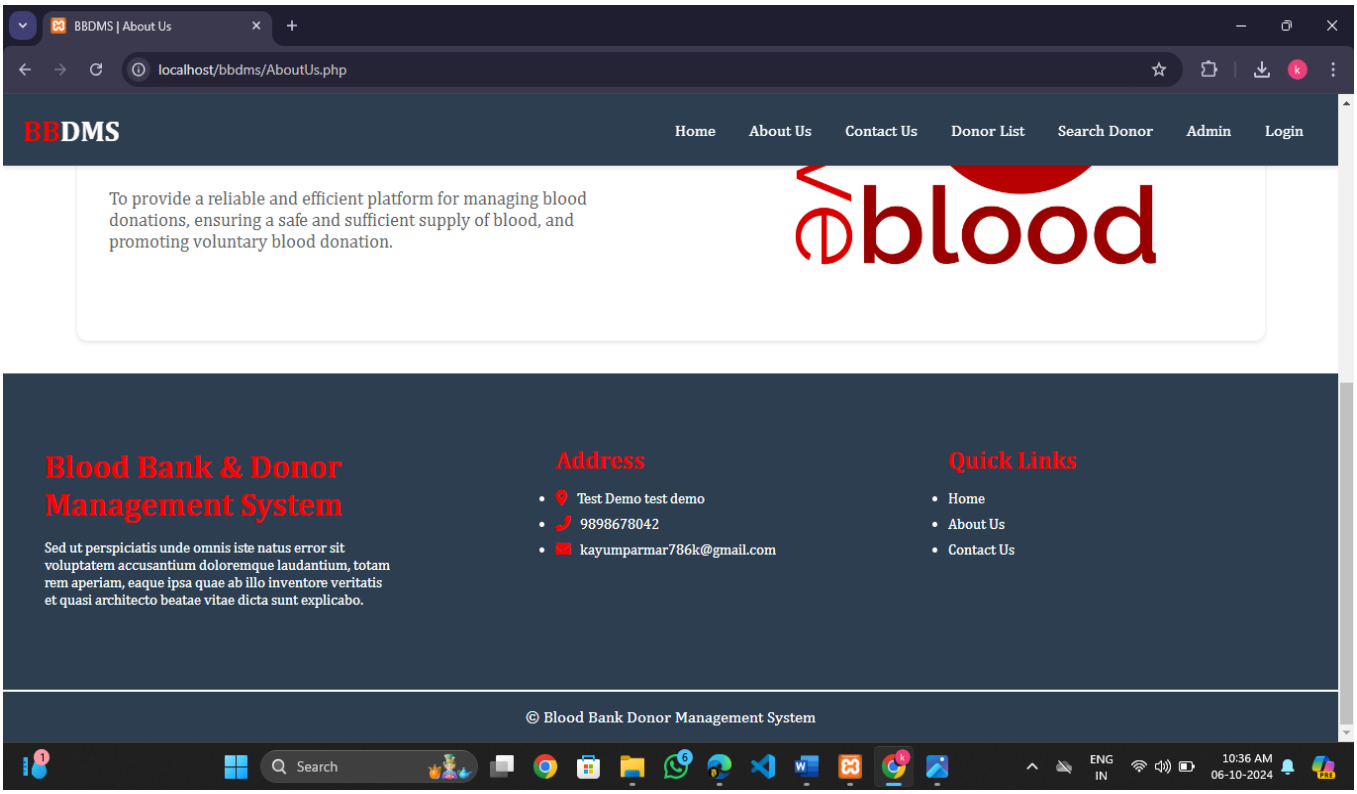



Blood Bank & Donor Managem
localhost/bbdms/
BBDMS
Home
About Us
Contact Us
Donor List
Search Donor
Admin
Login
Share Life Like You Share Wi-Fi
This poster encourages blood donation by drawing a parallel between readily shared Wi-Fi and the life-saving potential of blood. The graphic depicts a smartphone, symbolizing connectivity and ease of access, nestled within a blood bag. It emphasizes the importance of sharing life-giving blood, much like sharing Wi-Fi, for the benefit of others.
The poster promotes a blood donation drive organized by Wings Foundation, specifying the date and time for the event.
SHARE LIFE LIKE YOU SHARE Wi-Fi
BLOOD DONE
SHARE LIFE
LIKE YOU SHARE WI-FI
BLOOD DONATION
WINGS FOUNDATION
Blood Bank & Donor
Address
Quick Links
Search
ENG IN
10:33 AM 06-10-2024

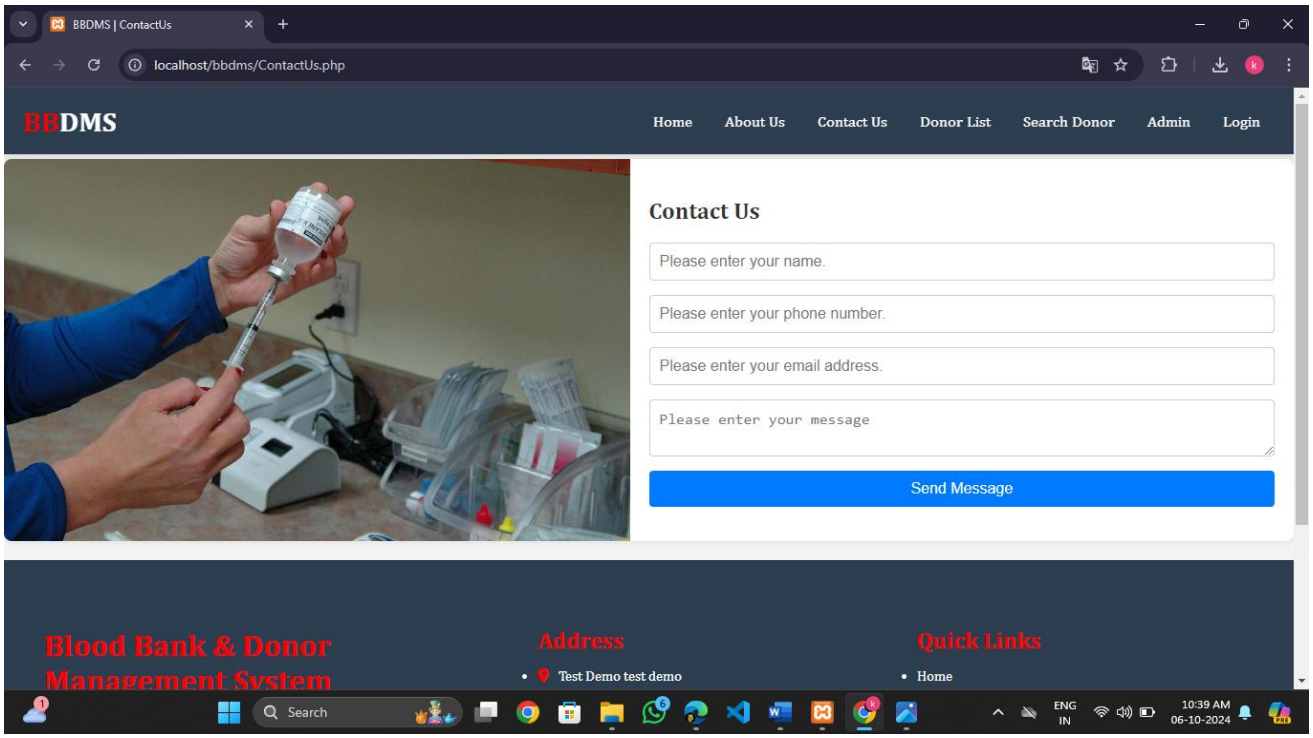

Blood Bank & Donor Managem
localhost/bbdms/
BBDMS
Home
About Us
Contact Us
Donor List
Search Donor
Admin
Login
emphasizes the importance of sharing life-giving blood, much like sharing Wi-Fi, for the benefit of others.
The poster promotes a blood donation drive organized by Wings Foundation, specifying the date and time for the event.
SHARE LIFE LIKE YOU SHARE WI-FI
BLOOD DONATION
WINGS FOUNDATION
Blood Bank & Donor Management System
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
Address
Test Demo test demo
9898678042
kayumparmar786k@gmail.com
Quick Links
Home
About Us
Contact Us
© Blood Bank Donor Management System
Search
ENG IN
10:33 AM 06-10-2024

About Us Page :

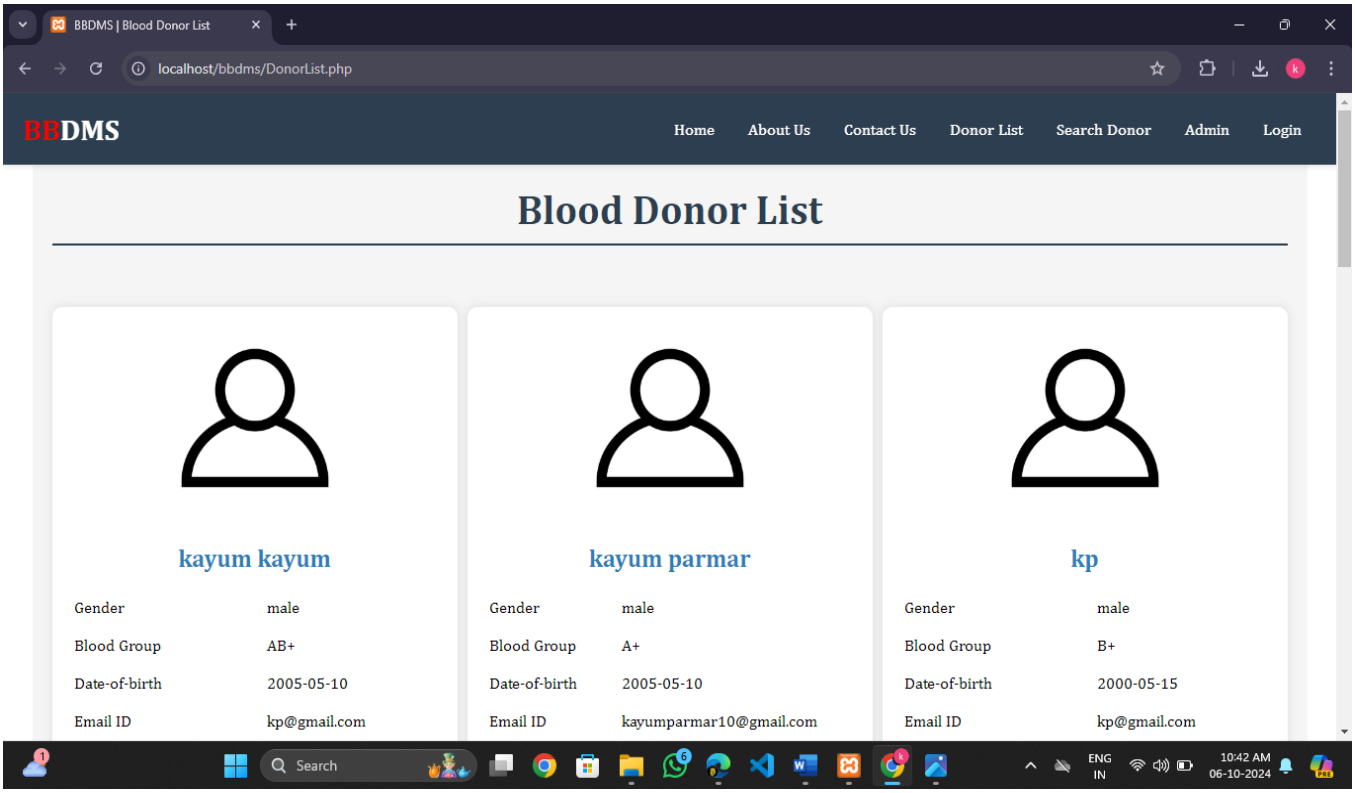


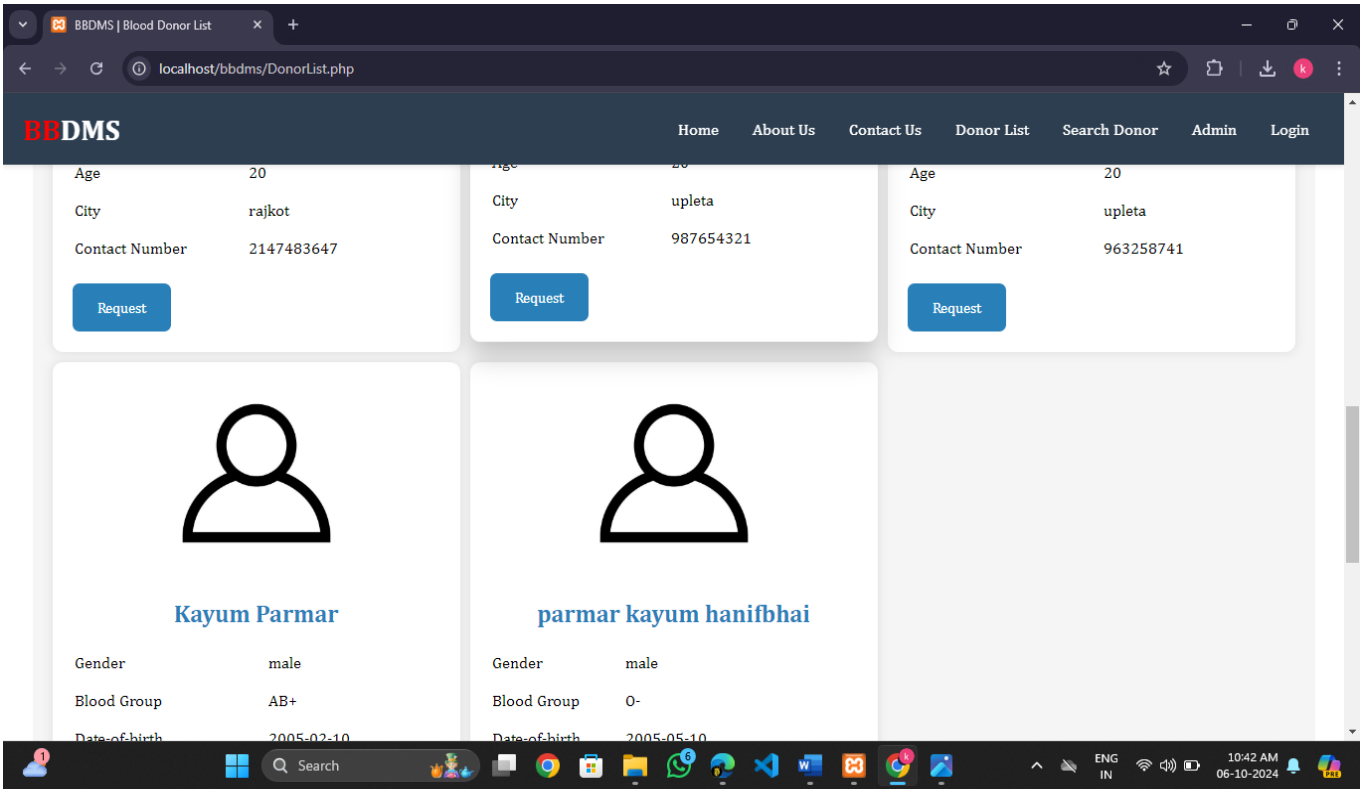


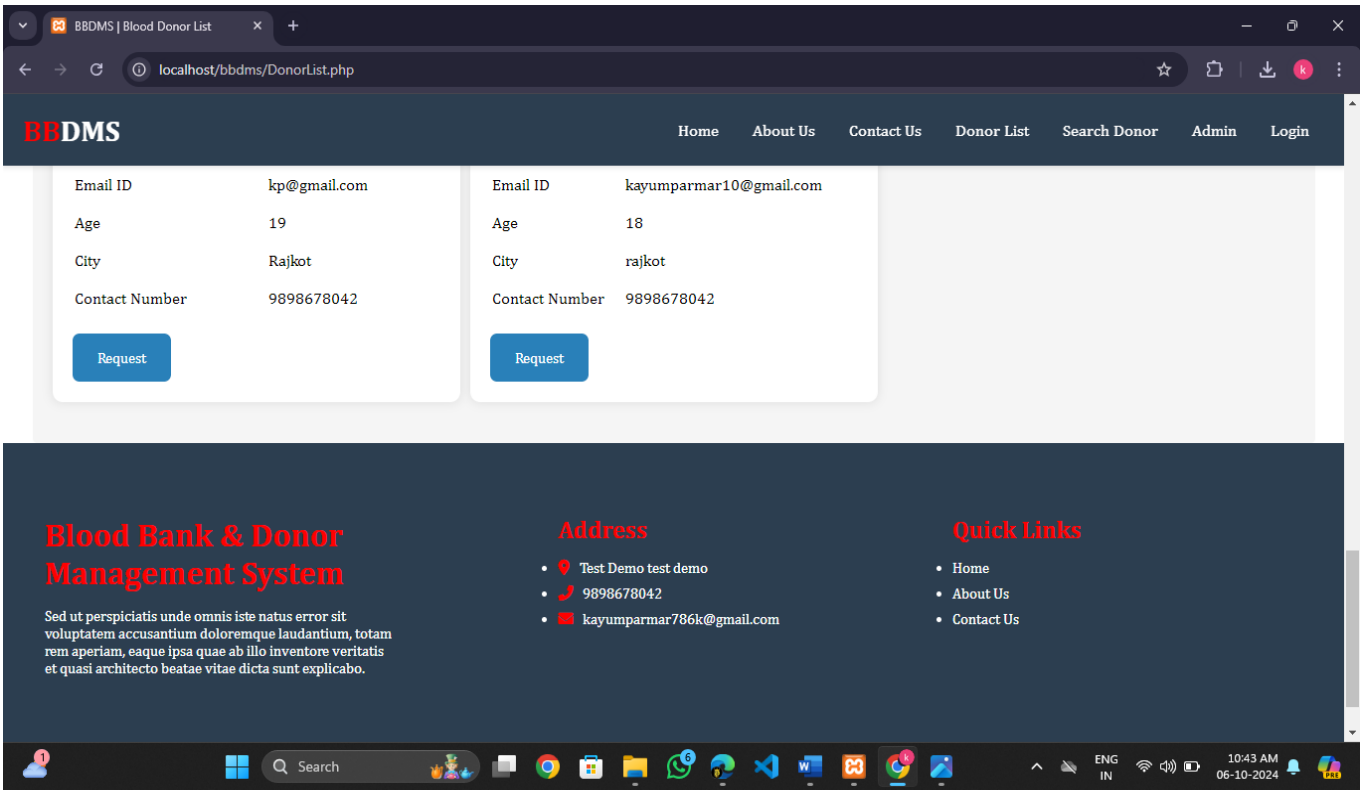
Contact Us Page :



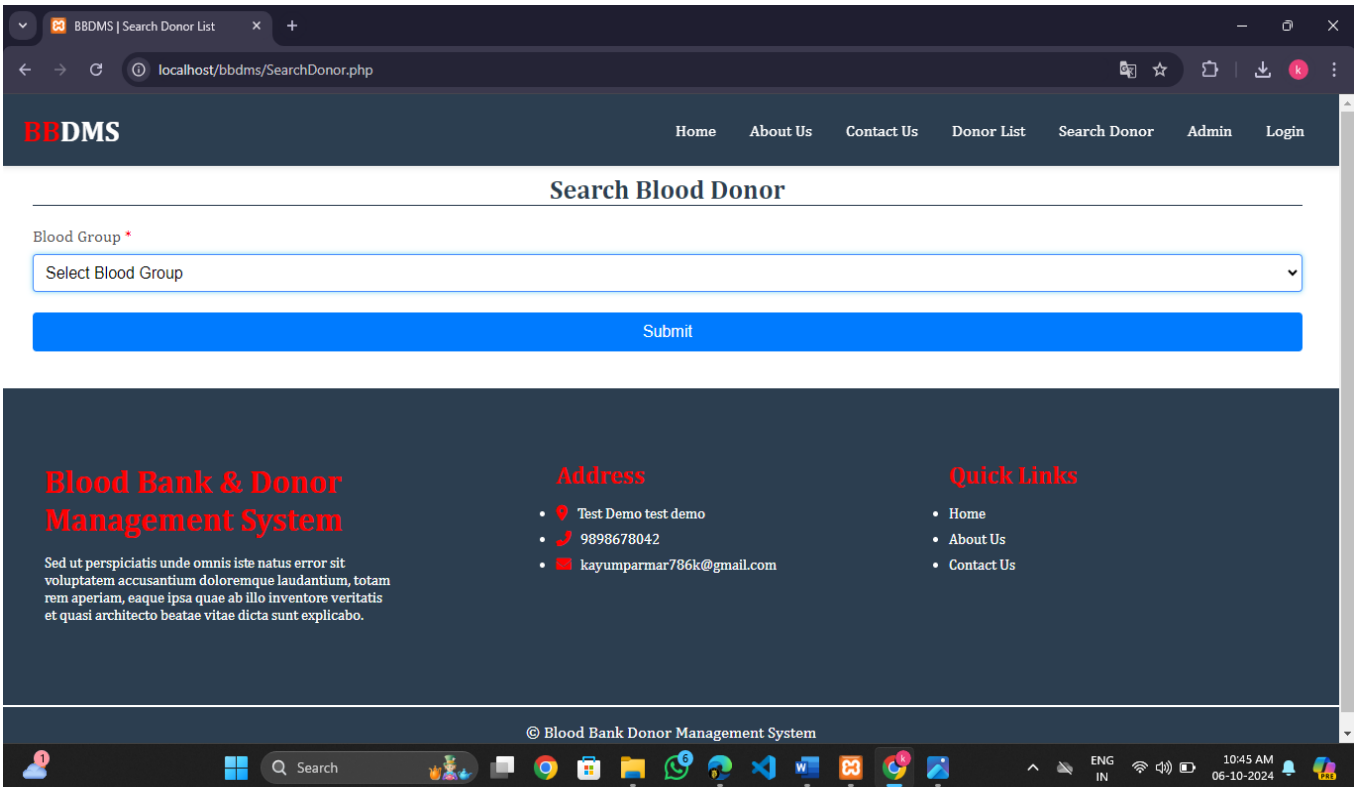
Donor List :



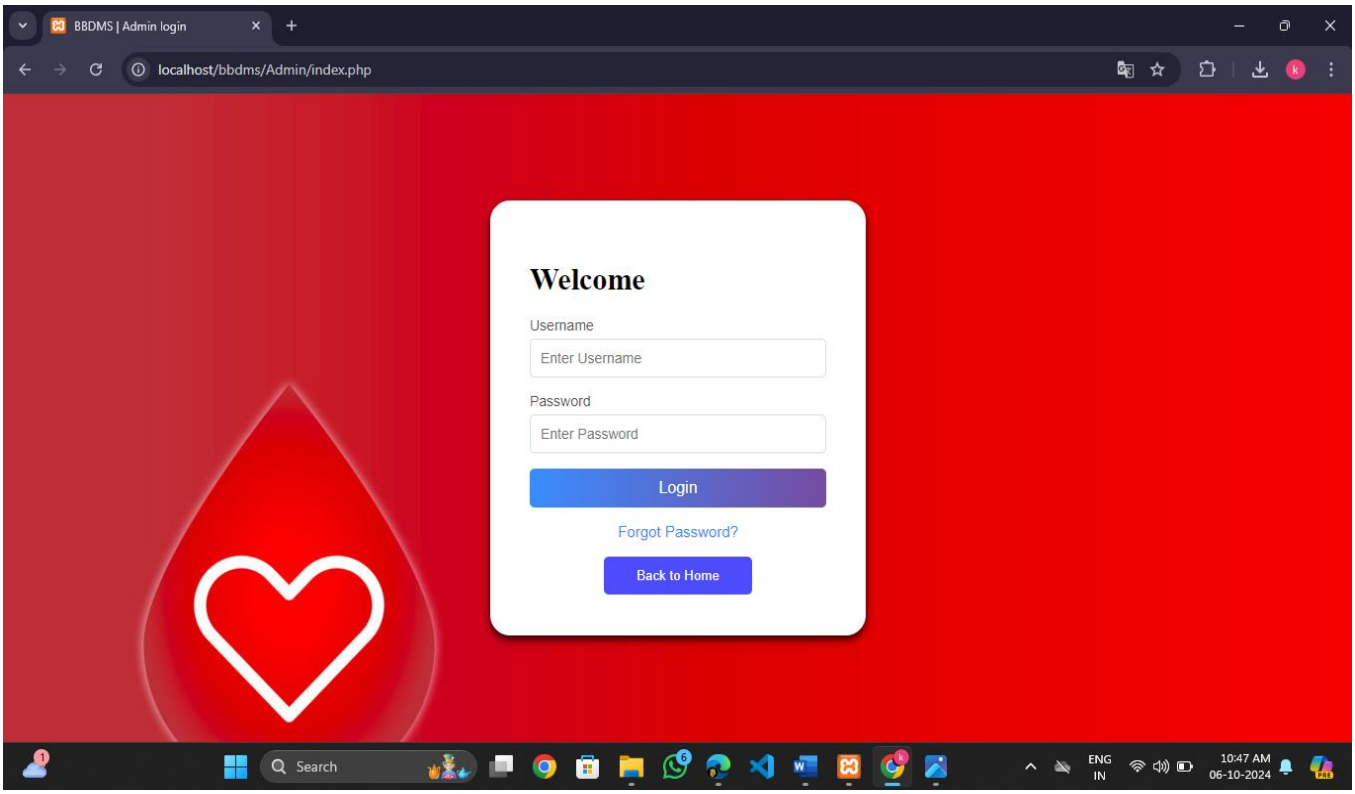

BBDMS | Blood Donor List
localhost/bbdms/DonorList.php
BBDMS
Home
About Us
Contact Us
Donor List
Search Donor
Admin
Login
Age 20
City rajkot
Contact Number 2147483647
Request
City upleta
Contact Number 987654321
Request
Age 20
City upleta
Contact Number 963258741
Request
Kayum Parmar
Gender male
Blood Group AB+
parmar kayum hanifbhai
Gender male
Blood Group O-
Search
ENG IN
10:42 AM 06-10-2024


BBDMS | Blood Donor List
localhost/bbdms/DonorList.php
BBDMS
Home
About Us
Contact Us
Donor List
Search Donor
Admin
Login
Email ID kp@gmail.com
Age 19
City Rajkot
Contact Number 9898678042
Request
Email ID kayumparmar10@gmail.com
Age 18
City rajkot
Contact Number 9898678042
Request
Blood Bank & Donor Management System
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
Address
Test Demo test demo
9898678042
kayumparmar786k@gmail.com
Quick Links
Home
About Us
Contact Us
Search
ENG IN
10:43 AM 06-10-2024

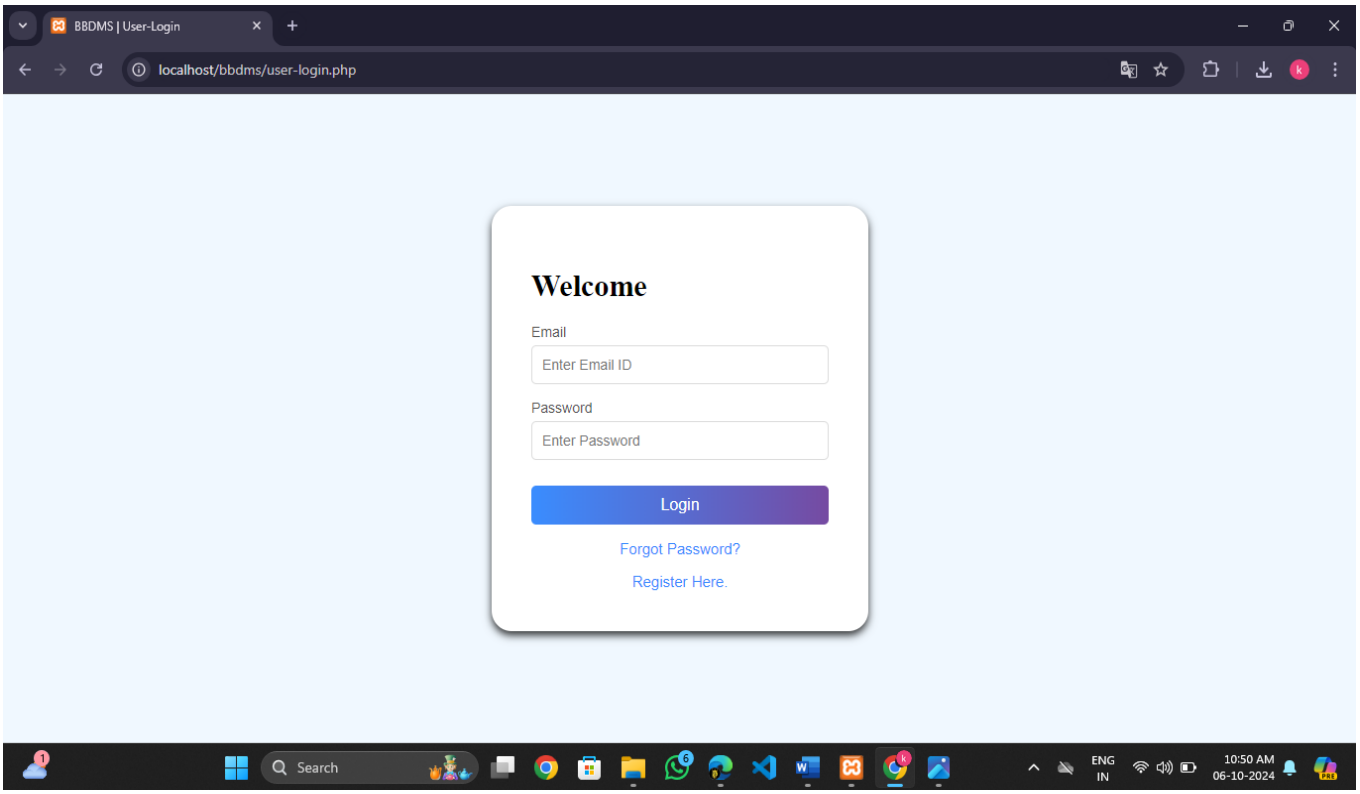
Search Donor :



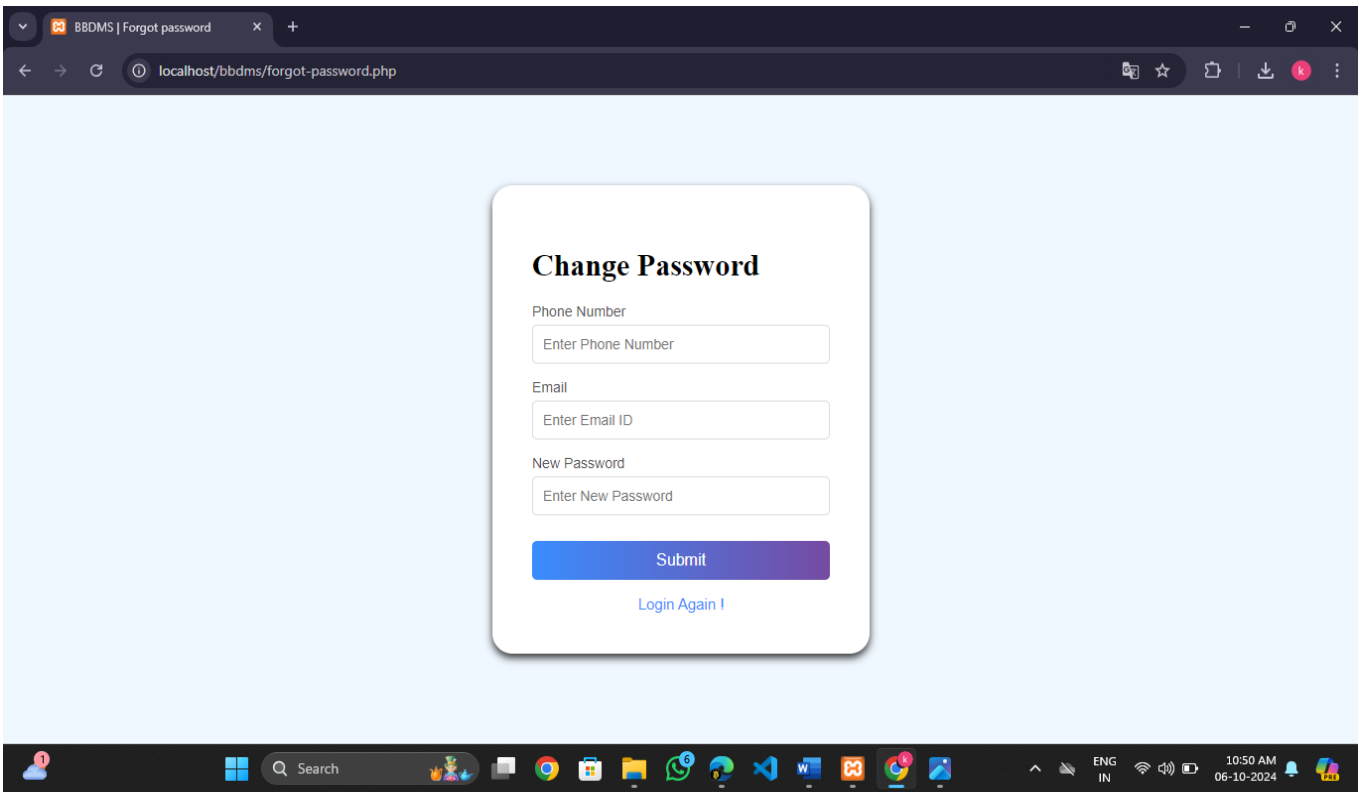
Admin Login Page :



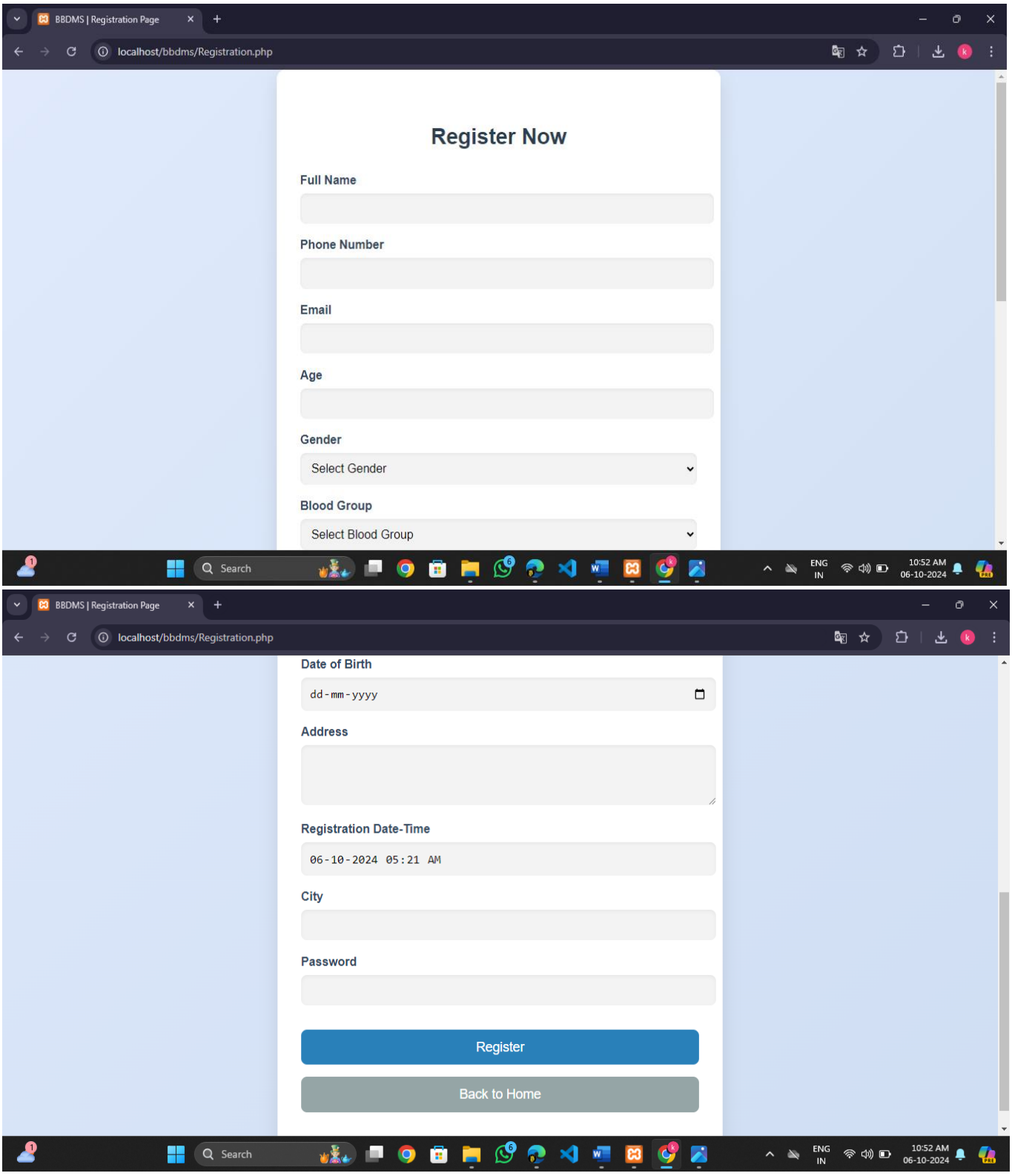
User Login Page :



Change password Page :

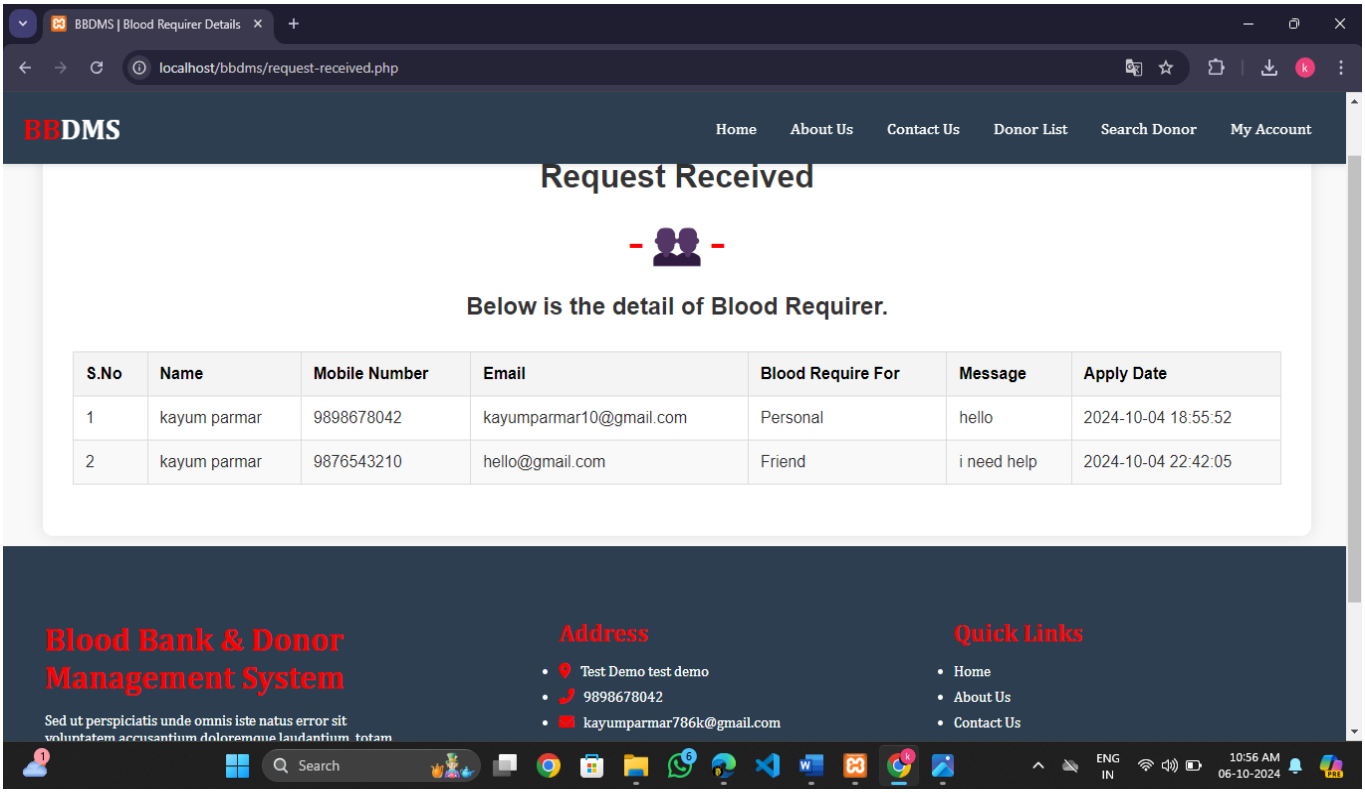


Register User Page :

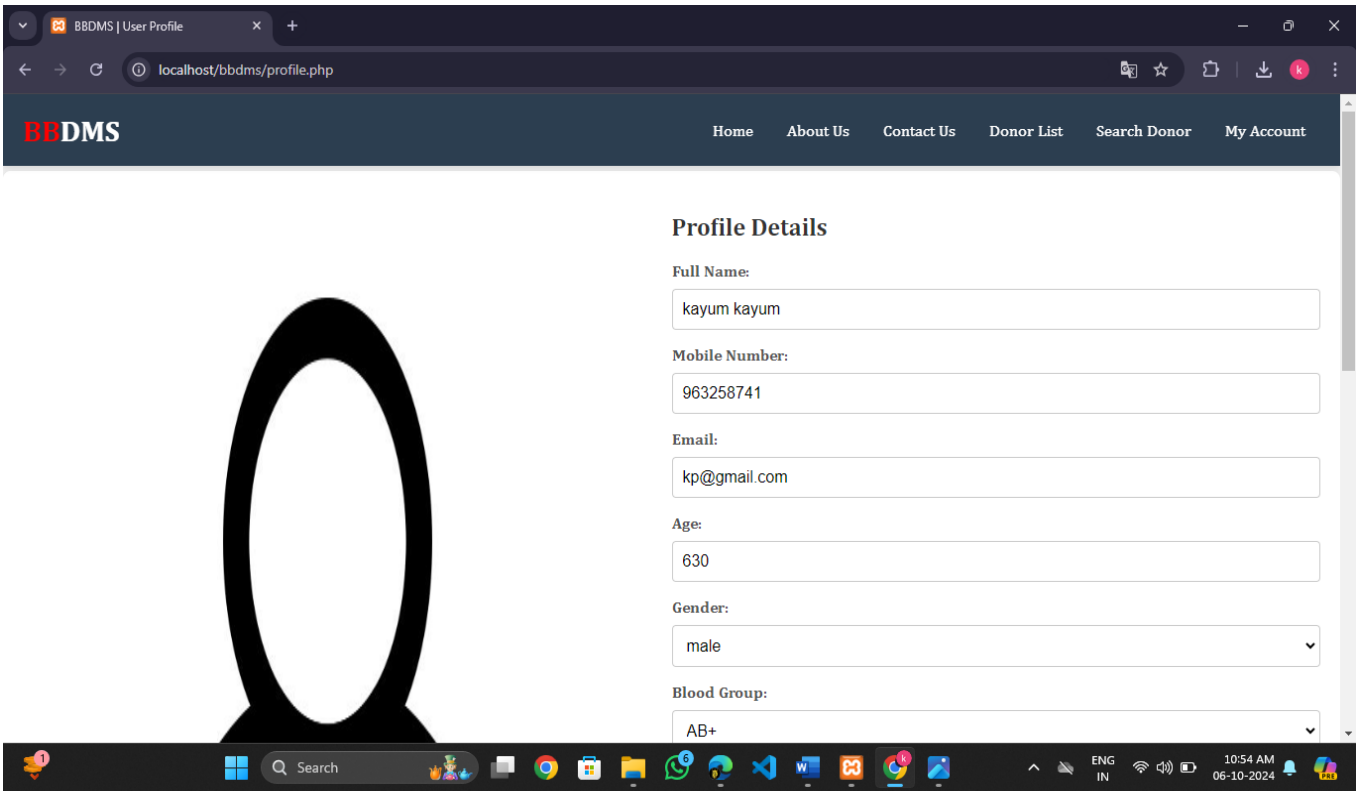


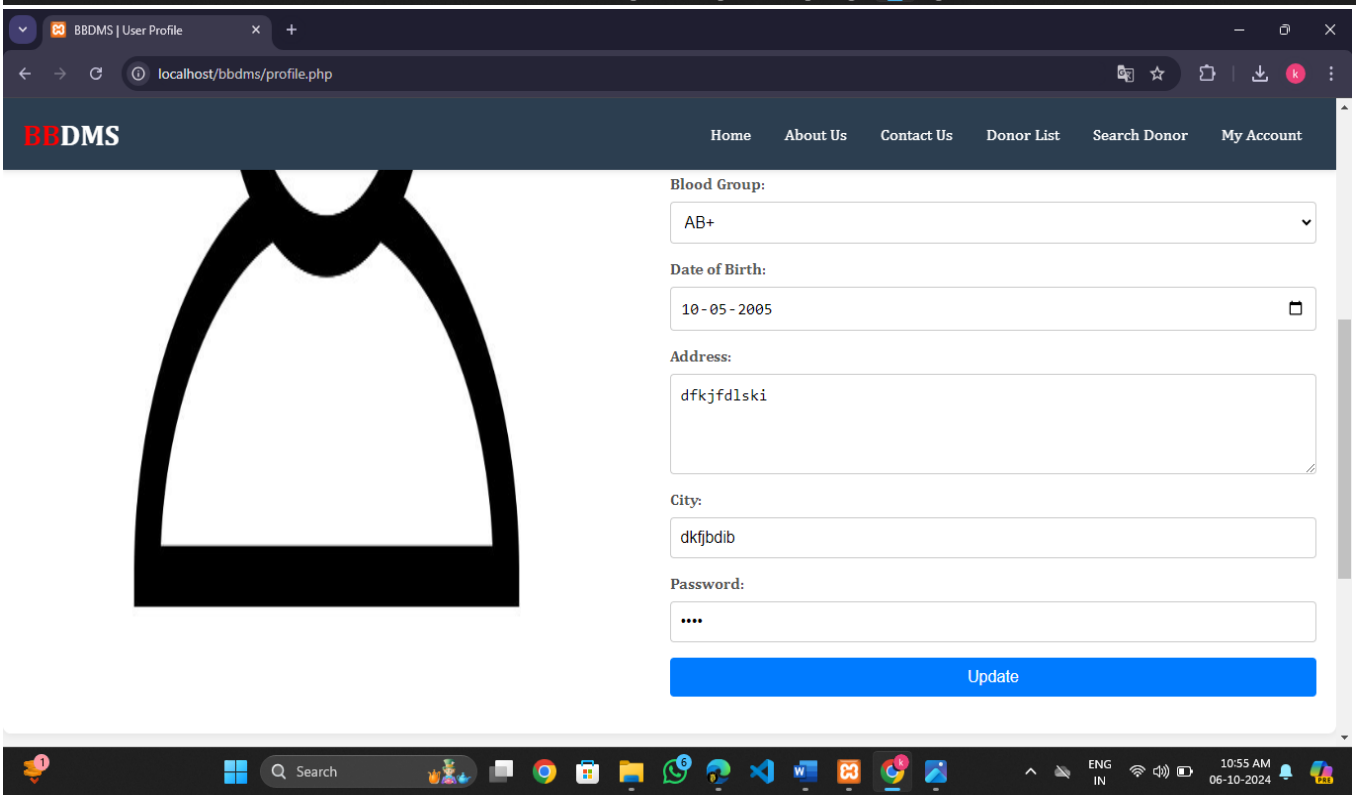
User Login Home Page :

User Request Page :



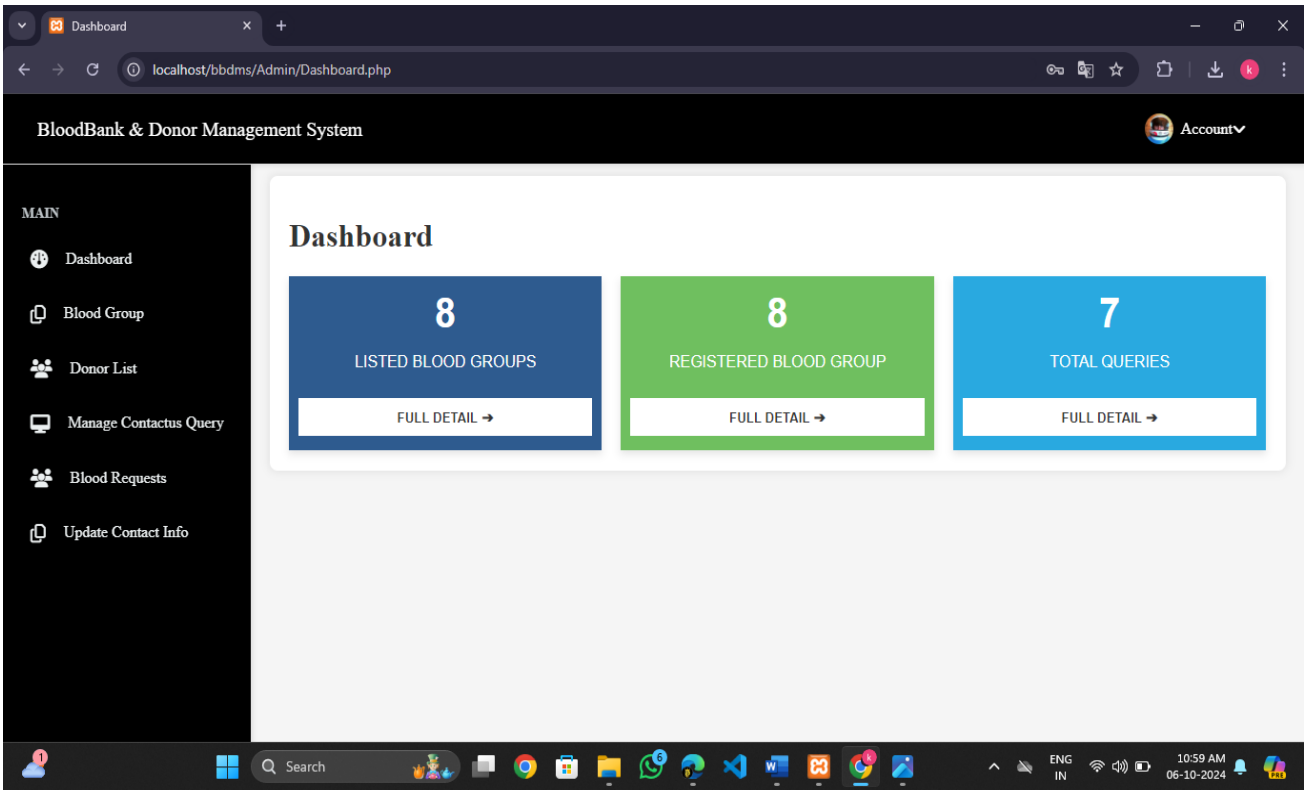
User Profile Page :



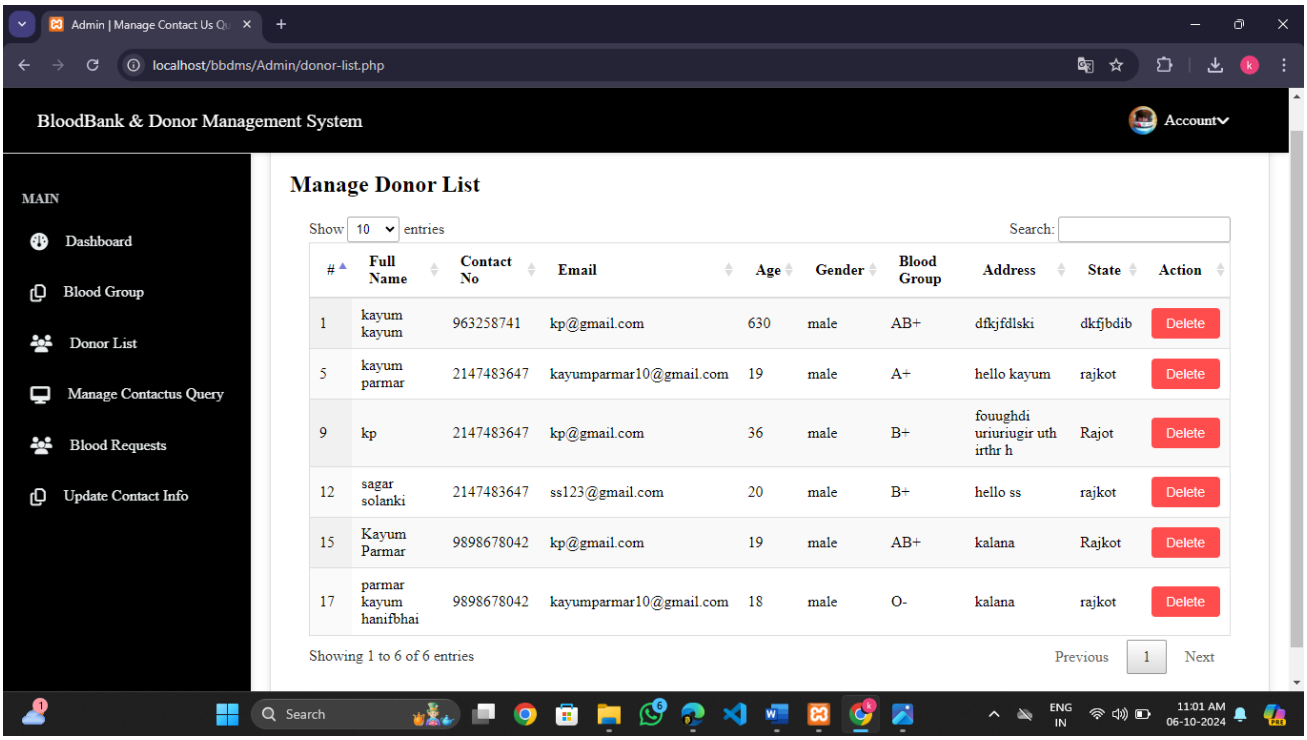


Admin Penel :

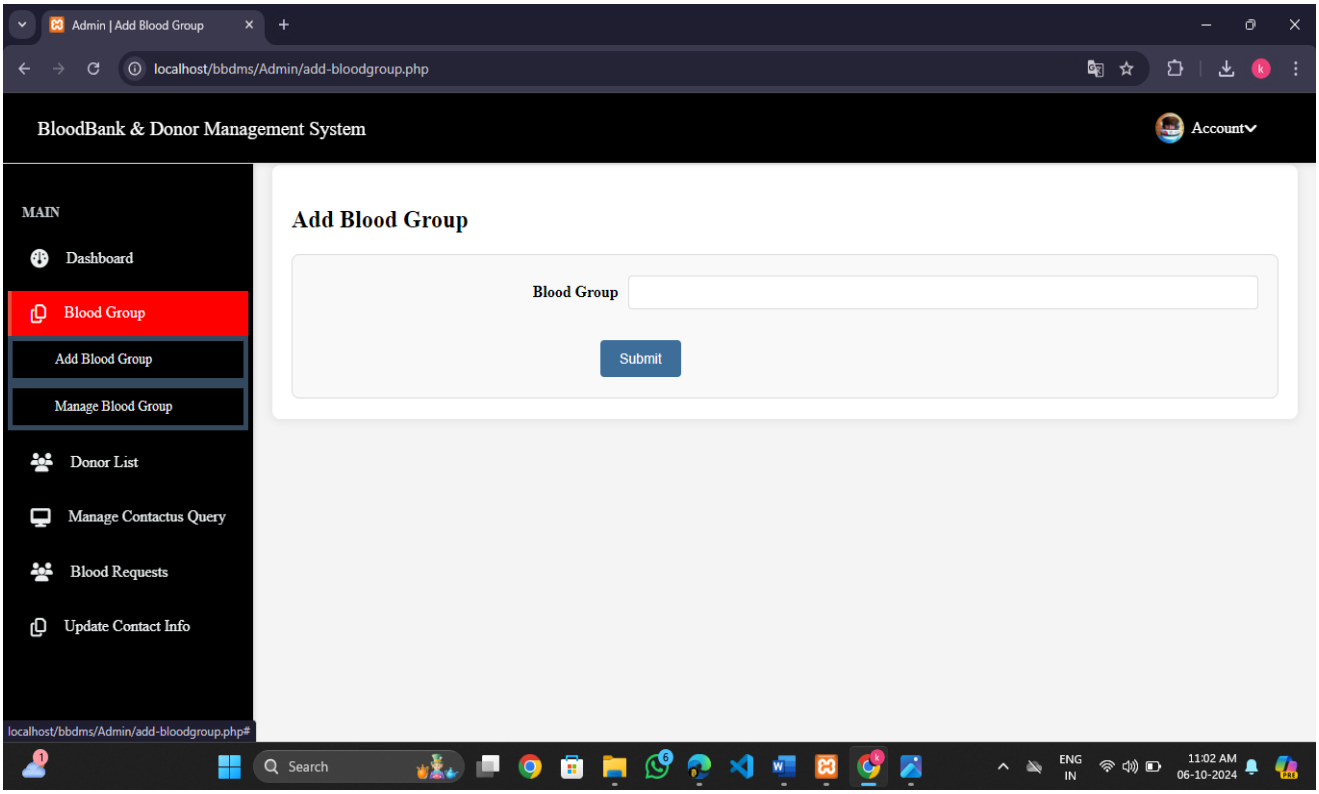
Admin Dashboard Page :

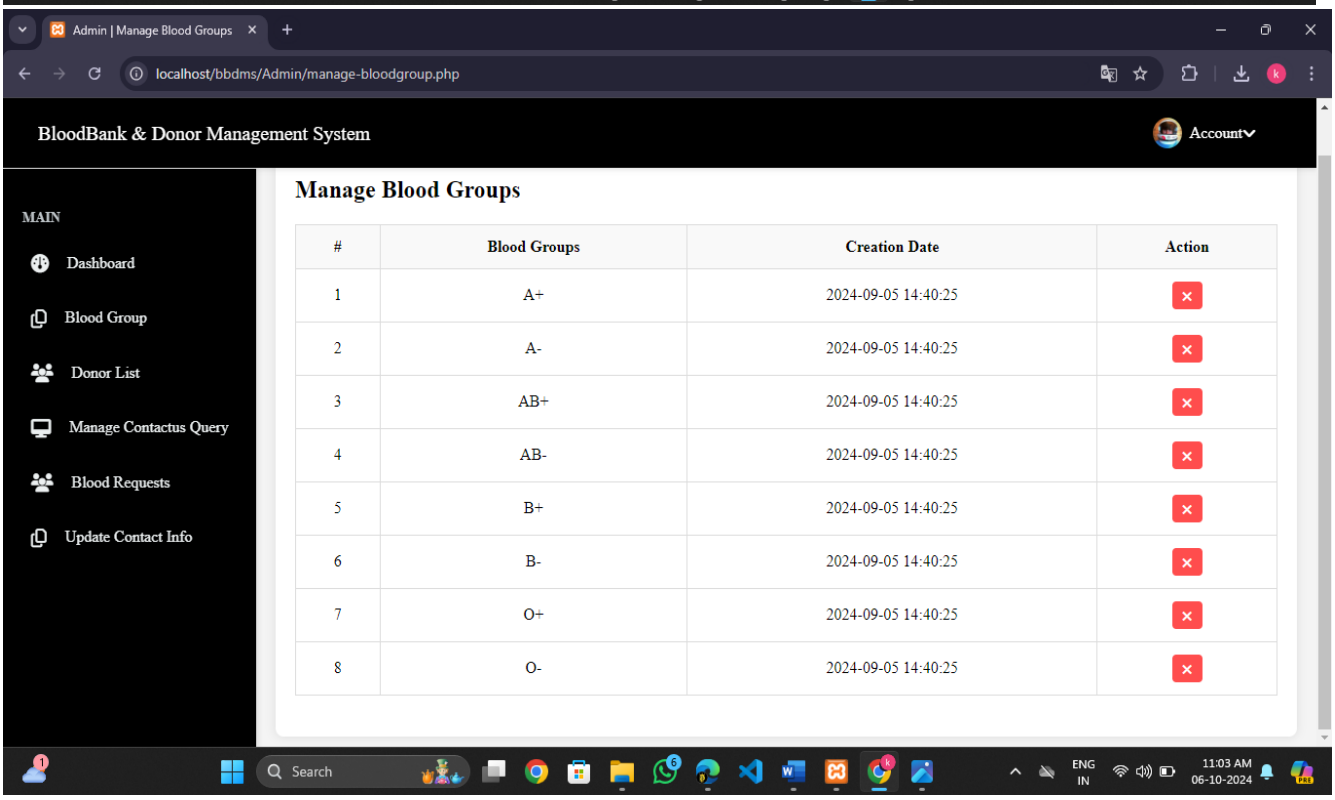


Donor List Page :

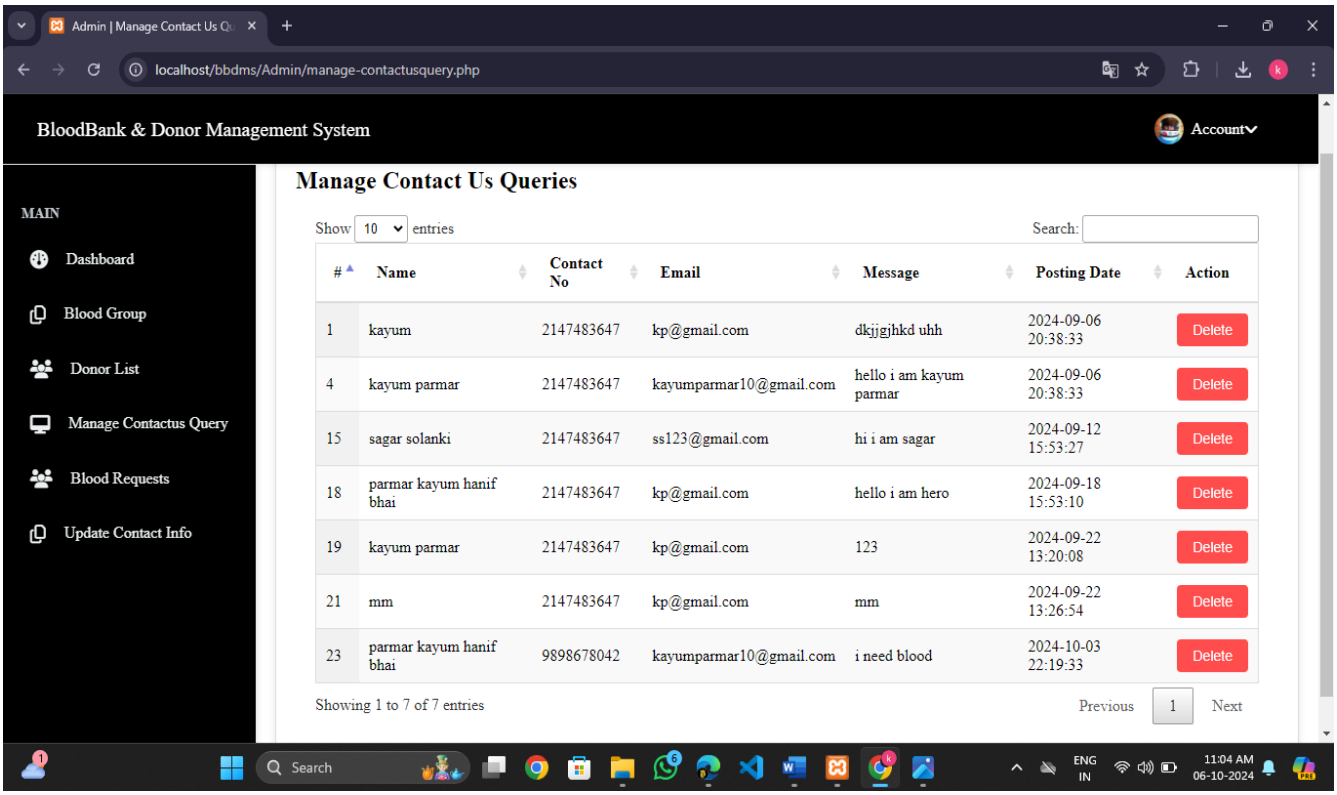


Blood Group Add And Manage :

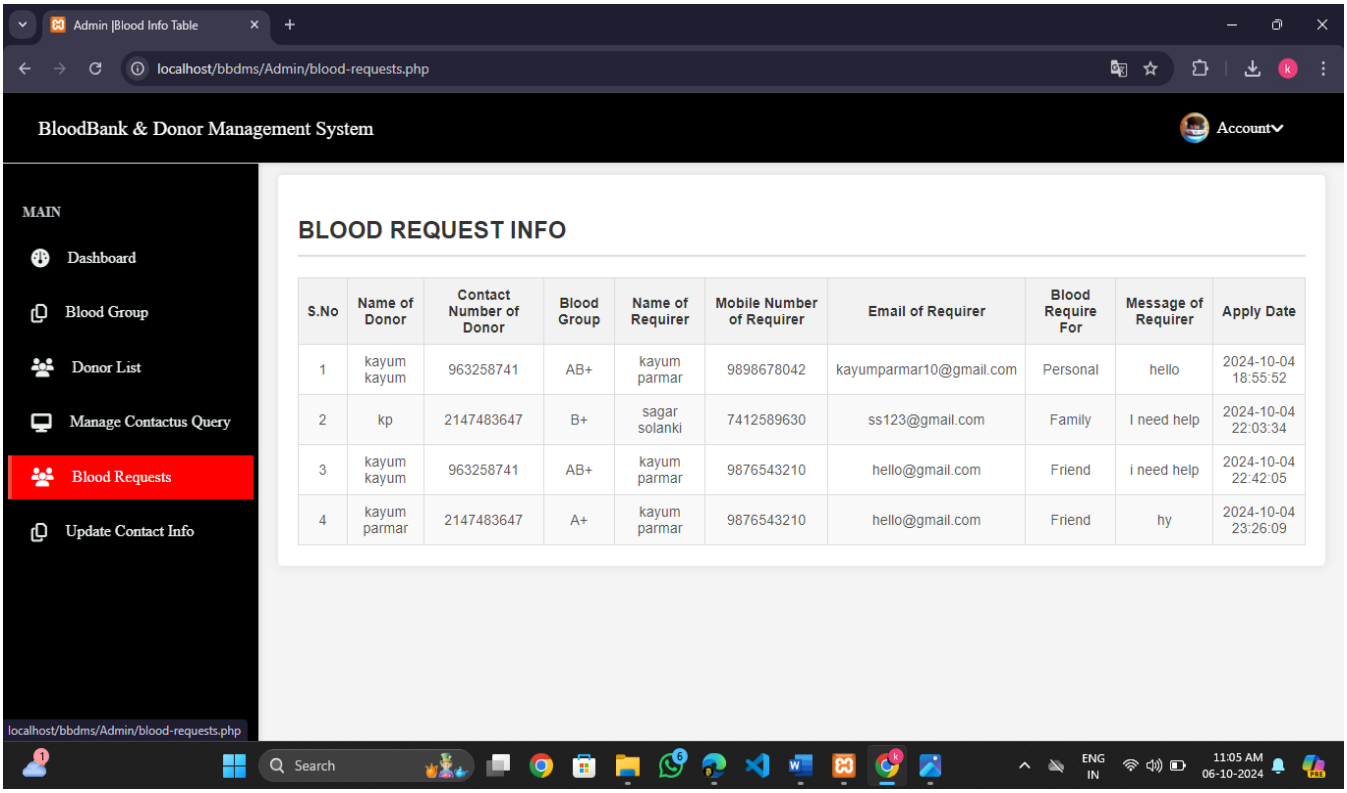




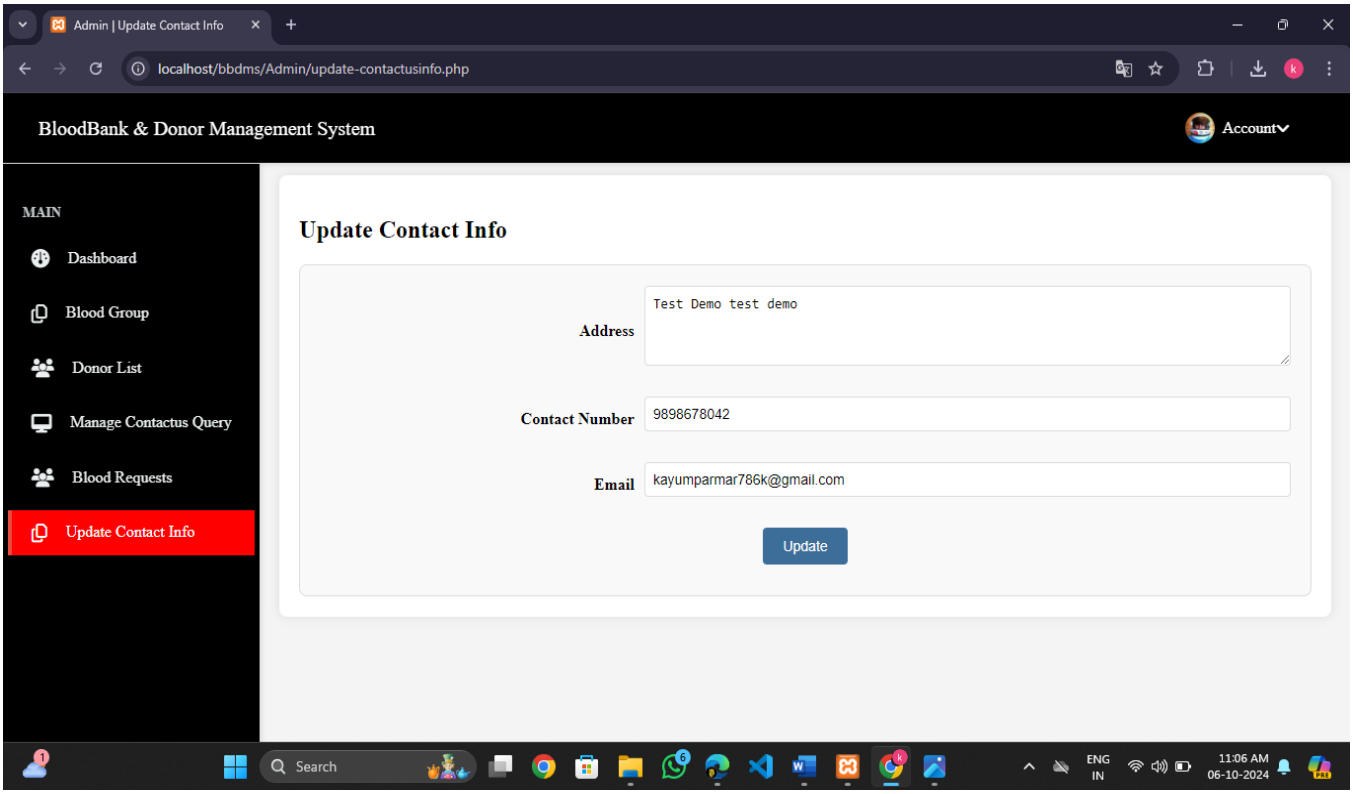
Manage Contact Us Queries Page :



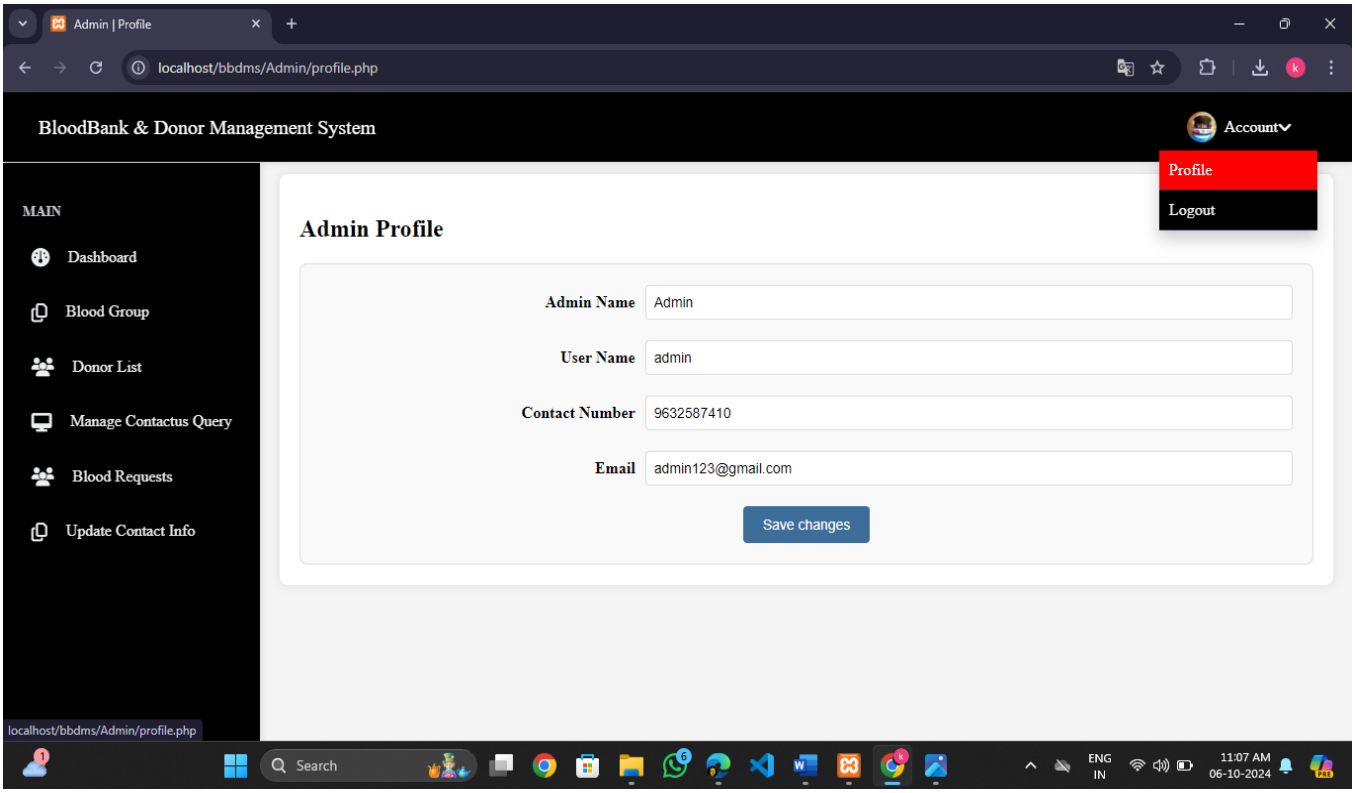
Blood Request Info Page :



Update Contact Info :



Admin Profile Page :



SYSTEM TESTING

System testing is the phase before system implementation, where the system is thoroughly tested to ensure it is error-free and meets all necessary requirements. During this stage, test data is used to identify errors and implement corrections. The reports generated during testing are reviewed and approved by the users. The system is designed to be user-friendly, with features such as online help to assist users as needed.

Test Plan:

A **test plan** is a detailed document outlining the scope, strategy, and schedule of the testing process. It also identifies the test items and personnel responsible for different testing activities. The test plan for a project is critical in ensuring that all aspects of the system are thoroughly evaluated before it goes live.

Major Testing Activities:

The test plan covers several key activities:

- **Test units**: Identifying the individual components or units of the system to be tested.
- **Features to be tested**: Specific system features that need to be evaluated.
- **Testing approach**: Describes the methods and strategies that will be used during testing.
- **Test deliverables**: Outlines what documentation and results are to be produced during the testing process.
- **Schedule**: Specifies the timeline and deadlines for each stage of testing.
- **Personnel allocation**: Identifies the individuals responsible for carrying out the testing activities.

Test Units:

Testing is conducted at two levels: **Unit Testing** and **System Testing**.

Unit Testing:

Unit testing focuses on individual components or functions of the system to ensure they work as intended. The key units tested include:

- **Validating user requests**: Ensures that requests made by users are processed correctly.
- **Validating user input**: Checks that the system accepts and processes input data accurately.
- **Exception handling**: Tests how the system responds to errors or exceptions during operation.

System Testing:

System testing is conducted after unit testing to ensure that the integrated system functions as expected. Key aspects tested during system testing include:

- **Integration of programs**: Verifies that all programs and components are integrated correctly.
- **Full system functionality**: Ensures that the entire system operates as intended once all components are working together.

Other Testing Strategies:

Alpha Testing:

Alpha testing is conducted at the developer's site, often with the customer present. The software is tested in a controlled environment, where the developer can monitor the user's interaction with the system, identifying any issues or usability problems. Errors and other problems encountered during alpha testing are recorded for correction.

Beta Testing:

Beta testing takes place at the customer's site and involves end-users using the software in a real-world environment. Unlike alpha testing, the developer is not present during beta testing. Users report any issues they encounter back to the

developer. This testing phase provides valuable feedback from actual users, which is used to make further improvements before the final product is released to the broader customer base.

Test Deliverables:

The testing process results in several important documents, which serve as evidence that the system has been thoroughly tested:

- **Unit Test Report**: A report documenting the results of unit testing for each component of the system.
- **Test Case Specification for System Testing**: A detailed document outlining the test cases used during system testing, which must be reviewed before testing begins.
- **System Testing Report**: A report summarizing the findings of system testing, including any issues found and corrective actions taken.
- **Error Report**: A log of any errors or bugs encountered during the testing process, along with their status (e.g., resolved or pending).

These documents are essential for ensuring transparency in the testing process and providing a record of the system's performance during testing.

IMPLEMENTATION AND EVALUATION

During the software-testing phase each module of software is thoroughly tested for bugs and for accuracy of output. The system developed is very user-friendly and the detailed documentation is also given to the user as online help wherever necessary. The implementation phase normally ends with the formal test involving all the components.

The entire system was developed using the PHP, HTML, JavaScript, Personal Web Server, and MYSQL as back end. The HTML is used to design the web page. The Personal Web Server is used to understand the client's request and to send response to them. The JAVASCRIPT are used for client-side validations so that the user can enter only appropriate input in the input fields. The MYSQL is the back end tool where the database resides.

Hence the design of the entire system is user-friendly and simple the implementation has been quite easy.

Future Enhancement:

As there was a little number of contact person's information given, some people may face difficulty in getting blood fast. So i like to gather more information regarding the contact persons in other cities as well as villages and will provide much more services for the people and help everyone with humanity.

BIBLIOGRAPHY

Reference Books and journal

- ✓ PHP and MySQL web development
- ✓ The complete reference PHP

Referring online manual from website

- www.php.net
- www.tutorialpoints.com
- www.w3school.com