A

Project Report On

**Blood Bank**

**(Blood Bank Management System)**

**By**

**Bhavin Sonagara (CE-135)(20CEUOG058)**

**Sparsh Sheth (CE-127)(20CEUOS109)**

**B. Tech CE, Semester VI**

**Subject: System Design Practice**

**Guided By:**

Prof. Apurva A Mehta

Assistant Professor

Dept. of Comp. Engineering



**Faculty of technology**

**Department of Computer Engineering**

**Dharmsinh Desai University**



**Faculty of Technology**

**Department of Computer Engineering**

**Dharmsinh Desai University**

**CERTIFICATE**

This is to certify that the practical/term work carried out in the subject of **Software Design Practice** and recorded in this journal is the bonafide work of

**Bhavin Sonagara , ID No: 20CEUOG058**

**Sparsh Sheth, ID No: 20CEUOS109**

of B.Tech semester VI in the brach of **Computer Engineering** during the academic year **2022-2023**.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Prof. Apurva A Mehta  Assistant Professor,  Dept. of Computer Engg.,  Faculty of Technology,  Dharmsinh Desai University,  Nadiad |  | Dr. C.K. Bhensdadia  Head,  Dept. of Computer Engg.,  Faculty of Technology,  Dharmsinh Desai University,  Nadiad |

**Acknowledgement**

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us during this project. This project would have been materialized without the help from many who asked us good questions and rescued from various red tape crisis.

Theoretical knowledge is of no importance if one doesn’t know the way of its implementation. We are thankful to our institute that provided us an opportunity to apply our theoretical knowledge through the project. We feel obliged in submitting this project as part of our curriculum.

We would like to take the opportunity to express our humble gratitude to our guide **Prof. Apurva A Mehta**, under whom we undertook our project. His constant guidance and willingness to share his vast knowledge made us enhance our knowledge and helped us to complete the assigned tasks to perfection. Without his effort, support and an astonishing testing ability this project may not have succeeded.

With Sincere Regards,

Bhavin Sonagara

Sparsh Sheth

**Table Of Content**

|  |  |  |
| --- | --- | --- |
| Sr .No | Topics | Page No |
| 1 | Abstract | V |
| 2 | Introduction | 1 |
| 3 | Software Requirement Specification | 3 |
| 4 | Design | 6 |
| 5 | Implementation Details | 13 |
| 6 | Testing | 17 |
| 7 | Screenshots | 18 |
| 8 | Conclusion | 24 |
| 9 | Limitation and Future Extension | 25 |
| 10 | Bibliography | 26 |

**1.Abstract**

Blood Buddies is a website primarily for connecting Donor, Receiver and Bloodbank . Bloodbank add a blood stock based on they get from the donor. Receiver search a blood stock present In the nearest bloodbank and add a request for a blood. Bloodbank can accept the request and give a blood to the receiver . Donor can donate a blood to the bloodbank and at the blood donation camp.

**1.1 Purpose**  
The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirement with respect to end users. Also, we shall predict and sort out how we hope this system will be used in order  
to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the system develops.

In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals.This document describes the project’s target audience, its user interface and software requirements. It defines how our clients, team and end user see the system and its functionality.

**1.2 Scope**Primarily, the scope of the system pertains to any donor who wants to donate a blood or any receiver who wants to find a blood.

The remaining sections of this document provide a general description, including characteristics of the users of this project, the functional and data requirements of the system.

**2. Introduction**

**2.1 Brief Introduction**

The blood buddies system is a software application designed to manage the activities of a blood bank, including the registration of donors, testing of blood samples, storage and inventory management of blood units, and distribution of blood to hospitals and clinics. The system helps blood banks to maintain accurate and up-to-date records of donors and blood units, track the availability of blood types, and ensure that blood is distributed to those who need it in a timely and efficient manner.

By using a blood buddies system, blood banks can streamline their operations, reduce errors and inefficiencies, and ultimately save lives by ensuring that blood is available when and where it is needed.

**2.2 Tools / Technologies Used**

Technologies :

• HTML5  
• CSS3  
• Bootstrap  
• ReactJS  
• Express for NodeJS  
• MongoDB

Tool :

• Visual Studio Code  
• GitHub

Platforms:

• Atlas Development Server

**3. Software Requirement Specifications**

**3.1 Types of Users**

1. Donor  
2. Receiver

3. Blood bank

4. Admin

**3.2 System Functional Requirements**

R1: Manage Bloodstock:

Description : Blood bank can manage blood stock increment blood stock as well as decrement blood stock. Blood bank can take a blood from a donor and it can give to the receiver.

R.1.1 : Add a blood stock

Input : Enter a details of blood and user like blood group, birthdate and last donation date  
Output : give a greeting message.

R.1.2: Remove blood stock

Input : Accept the user request for a blood if blood is present else decline request.

Output : give a status of a request.

R2: Manage Blood bank

Description: Admin can add blood bank based on the request of the user. Admin verify the details of the blood bank and add blood bank.

R.2.1: Add Blood bank

Input: Accept a request of user of user

output : Greeting message.

R.2.2 Delete Blood bank

Input : click on the delete button

Output: Greeting message

R.3 : Manage Users

R.3.1: Login system :  
Description :- If a user already has an account then he/she needs to login by entering his username and password. If user is new then he need to register(sign up)  
Input :- user details  
Output :- confirmation message.

R.3.2 Sign up :  
Description: If a user does not have an account then he has to first make an account by providing details such as username, password, mobile number , email id and all details will be stored in the database.  
Input:-user details  
Output:-confirmation message

R.3.3: Recover account :  
Description :- if the user has already an account and he forgets the account password he can recover the account.  
Input :- Need to provide registered mail id.  
Output :- User will get OTP on provided mail

R.3.4: Update user details  
Description:- User can update his own details.  
Input:- all new details of that user need to change.  
Output:- confirmation page.

R.4: Manage Request

Description: Blood bank can manage a request(application)

of bloodstock and admin can mange a request of the user for the bloodbank.

R.4.1: Accept a request

Input: Click on the accept request.

Output: Status of the request is changed

R.4.2: Decline the request

Input: Click on the decline button

Output: Status of the request is changed

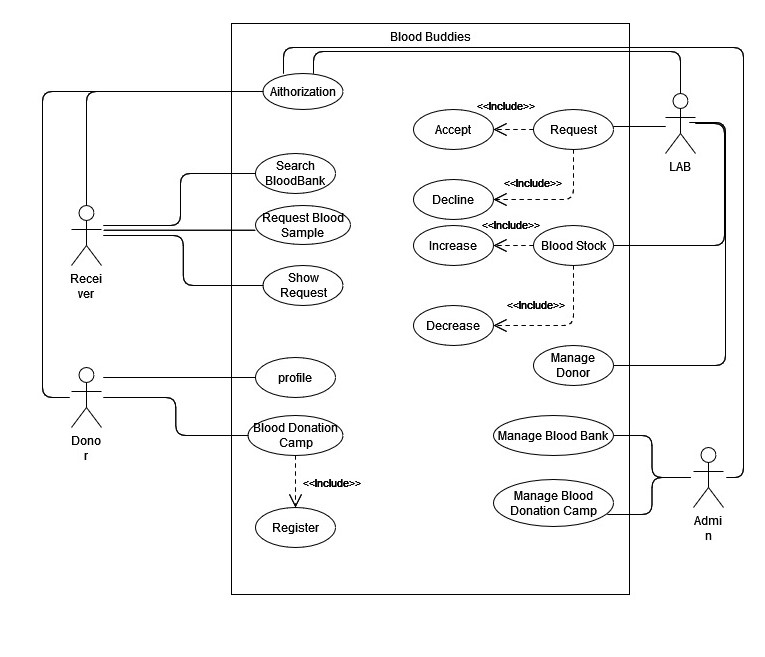
* 1. **Other Non-functional Requirements**
     1. **Usability:**  
        • The system must be easy to use by users such that they do not need to read an extensive amount of manuals.  
        • The options of the system must be easily navigable by the users with buttons that are easy to understand.
     2. **Reliability:**  
        • The system should not update the data in any database for any failed processes.  
        • The system is able to update and delete information which is provided by system user very easily.

3.3.3 **Performance:**

• The system must complete updating the databases options successfully every time the user requests such a process.  
• All the functions of the system must be available to the user every time the system is turned on.

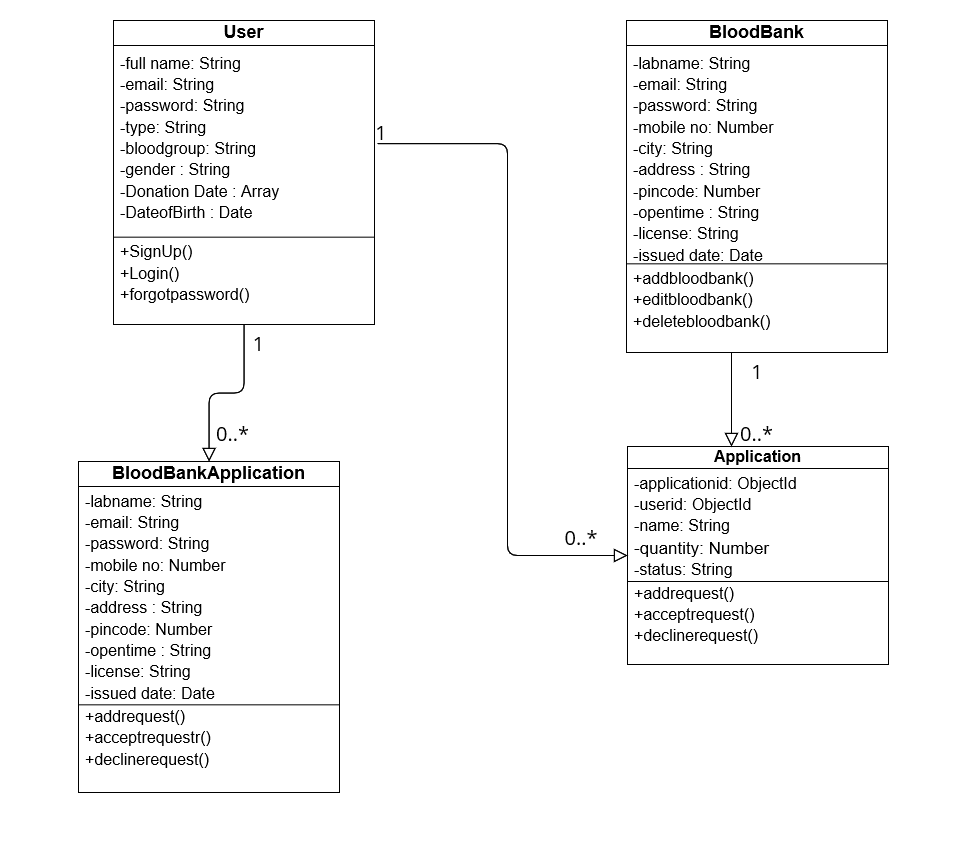
1. **Design**

**4.1 Use Case Diagram**

****

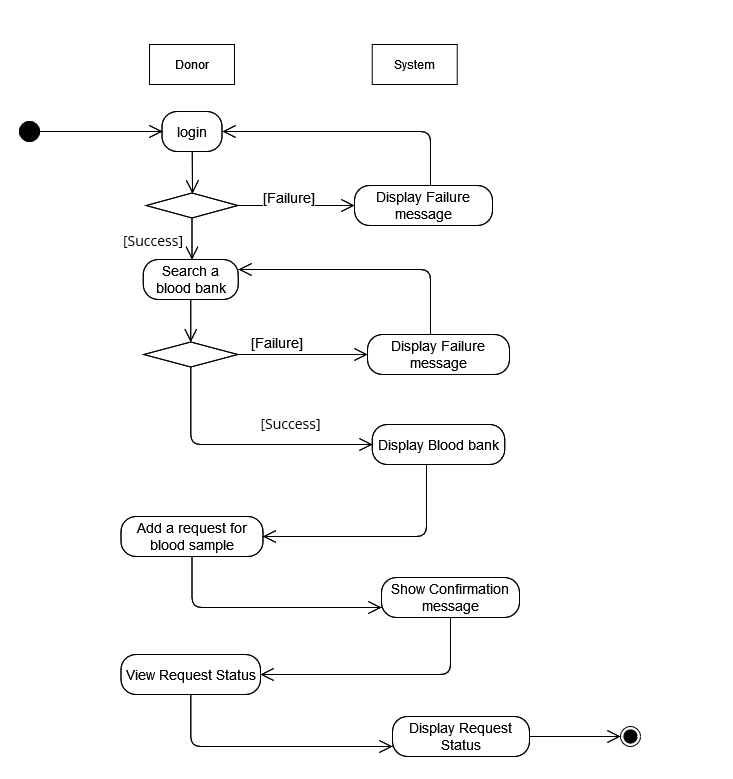
**Use Case Diagram of Blood Bank**

**4.2 Class Diagram**

****

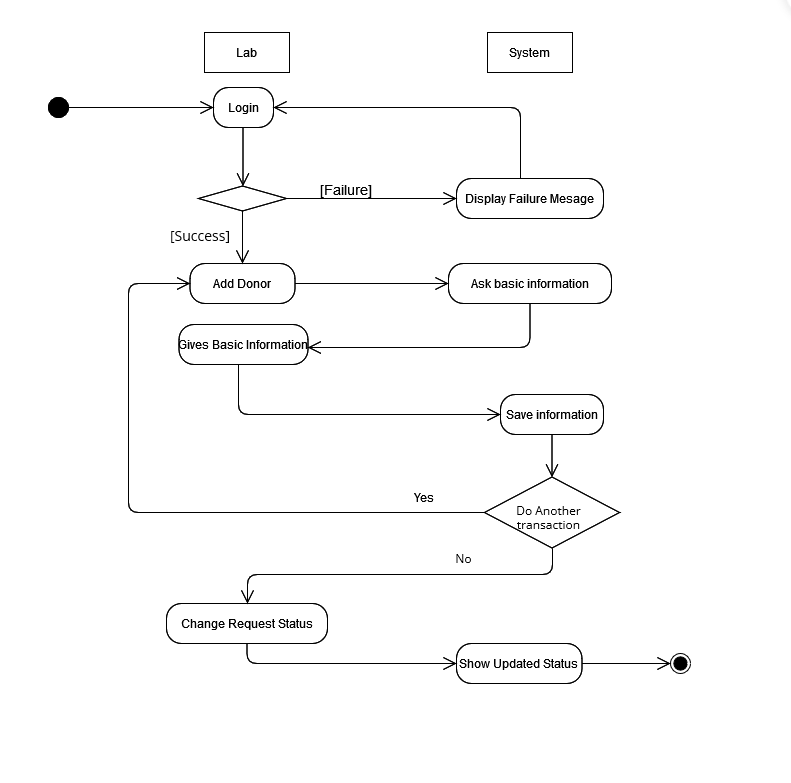
**Class Diagram of Blood Bank**

* 1. **Activity Diagram**
     1. **Activity Diagram of User**

****

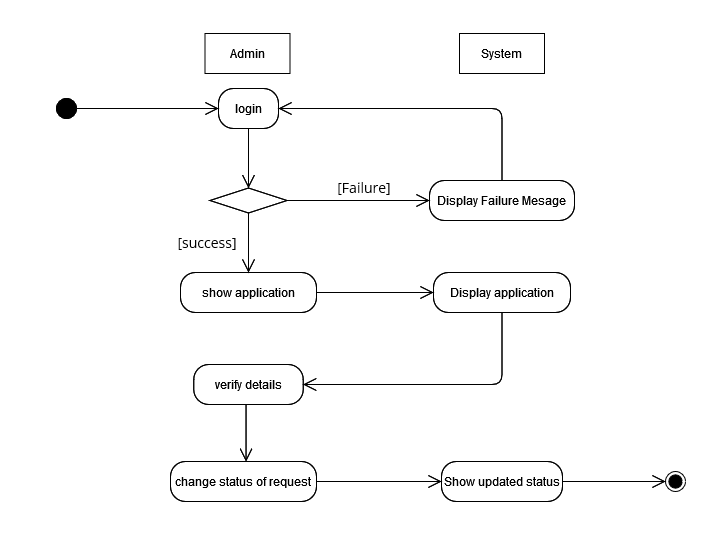
**Activity Diagram of User**

* + 1. **Activity Diagram of Lab user**

****

**Activity Diagram of Lab User**

* + 1. **Admin Activity Diagram**

****

**Activity Diagram of Admin**

**4.4 Data Dictionary**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User** | | | | | | |
| Sr  no | Field  Name | Data  Type | Required | Unique | Referred  Table | Description |
| 1 | Full name | String | yes | No | - | Must be full name |
| 2 | email | String | yes | Yes | - |  |
| 3 | password | String | yes | No | - |  |
| 4 | type | String | no | No | - | Default:  ‘user’ |
| 5 | bloodgroup | String | yes | No | - |  |
| 6 | gender | String | yes | No | - |  |
| 7 | Donation  date | Array | no | No | - | Bloodbank name and date object |
| 8 | Dateofbirth | Date | yes | No | - |  |

**4.4.1 User Model**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Blood Bank** | | | | | | |
| Sr.  no | Field  Name | Data  Type | Required | Unique | Referred  Table | Description |
| 1 | labname | String | yes | no | - |  |
| 2 | email | String | yes | yes | - |  |
| 3 | address | String | yes | no | - |  |
| 4 | password | String | yes | no | - |  |
| 5 | mobile\_no | Number | yes | no | - |  |
| 6 | City | String | yes | no | - |  |
| 7 | pincode | Number | yes | no | - |  |
| 8 | open time | String | yes | no | - |  |
| 9 | license | String | yes | no |  |  |
| 10 | Issued date | Date | yes | no | - |  |

**4.4.2 Blood Bank Model**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Application** | | | | | | |
| Sr.  no | Field  Name | Data  Type | Required | Unique | Referred  Table | Description |
| 1 | applicationid | ObjectId | yes | yes | - | Moongoose.  Schema.  Types.  ObjectId |
| 2 | userid | ObjectId | yes | yes | - | Moongoose.  Schema.  Types.  ObjectId |
| 3 | name | String | yes | no | - |  |
| 4 | quantity | Number | yes | no | - |  |
| 5 | status | String | yes | no | - | Default:  ‘Applied’ |

**4.4.3 Application Model**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Blood Bank Application** | | | | | | |
| Sr.  no | Field  Name | Data  Type | Required | Unique | Referred  Table | Description |
| 1 | labname | String | yes | no | - |  |
| 2 | email | String | yes | yes | - |  |
| 3 | address | String | yes | no | - |  |
| 4 | password | String | yes | no | - |  |
| 5 | mobile\_no | Number | yes | no | - |  |
| 6 | City | String | yes | no | - |  |
| 7 | pincode | Number | yes | no | - |  |
| 8 | open time | String | yes | no | - |  |
| 9 | license | String | yes | no |  |  |
| 10 | Issued date | Date | yes | no | - |  |

**4.4.4 Blood Bank Application model**

**5. Implementation Details**

The Project is divided into two parts: Client Side and Server Side.  
Client Side is implemented using ReactJS. It has all the required Components and  
Routing.

The Server Side is implemented using NodeJS, ExpressJS and mongoose.

MongoDB is used for data storage.

**5.1 Modules Description :**

The system consists of 3 main modules namely  
1. User Module  
2. Blood Bank Module

3. Admin Module  
Each module consists of several methods to implement the required  
functionality. Implementation is done using ReactJS, NodeJS, Express JS and  
MongoDB (for data storage).

**User Module :**  
Profile : In this section, User can create/update profile with all the  
necessary details or view profile if already created.

Blood bank : In this section, User can search the blood bank based his/her blood group required.

Application : In this section, User can add a request for a blood group to particular blood bank and user can see a application status.

**Blood Bank Module :**  
Profile : In this section, Blood Bank User can create/update profile with all the  
necessary details or view profile if already created.

Blood Stock : In this section, user can add a donor and blood stock to the blood bank and based on the request decrease a blood stock.

Application : In this section, User can show all application of the receiver and a change a status of the request.

**Admin Module :**

Blood Bank : in this section, admin can add blood bank based on the request of the user and manage a blood bank.

Application: In this section, admin can change a status of blood bank add application.

**5.2 Function Prototype :**

**API :**

Search Blood bank

user0Router.get("/search/:key1/:key2",auth, async(req,res) => {

    …

})

Request Blood Sample

user0Router.post('/user0/req-blood-sample', auth, async (req,res) => {

    …

})

Show Application

user0Router.get("/user0/show-application/",auth,async(req,res) => {

…

})

Add Blood Sample

labRouter.post("/lab/add-bloodsample/", auth, async (req, res) => {

    …

})

Accept Application

labRouter.patch("/lab/accept-application/:key", auth, async (req, res) => {

    …

})

Add Blood Donor

labRouter.post("/lab/add-blooddonor", auth, async (req, res) => {

    …

})

Add Blood Bank

adminRouter.post("/admin/add-bloodbank", async (req, res) => {

    …

})

Show All Blood Bank

adminRouter.get('/admin/fetchallbloodbank', async (req,res) => {

    …

})

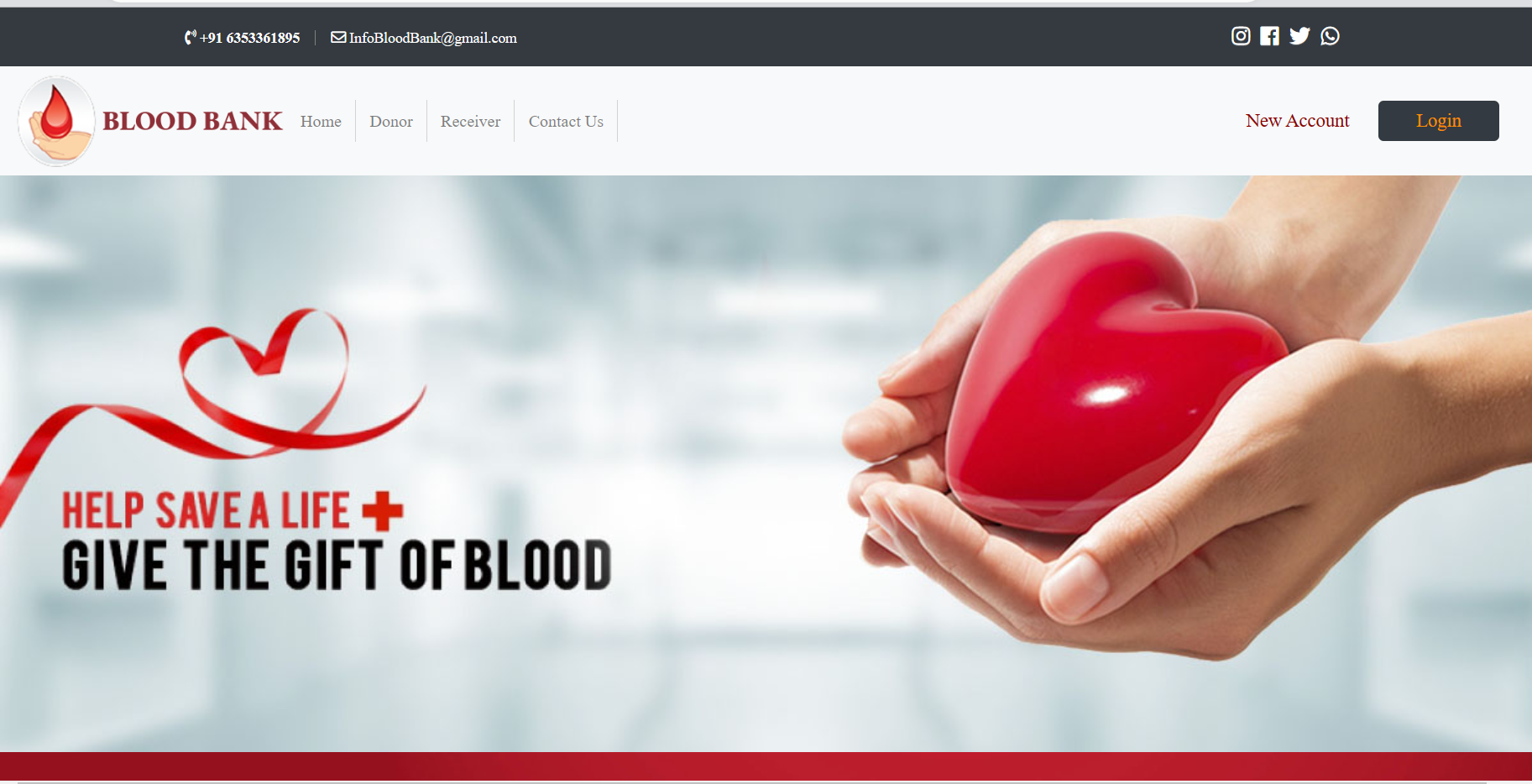
**6. Testing**

6.1 Testing Method:

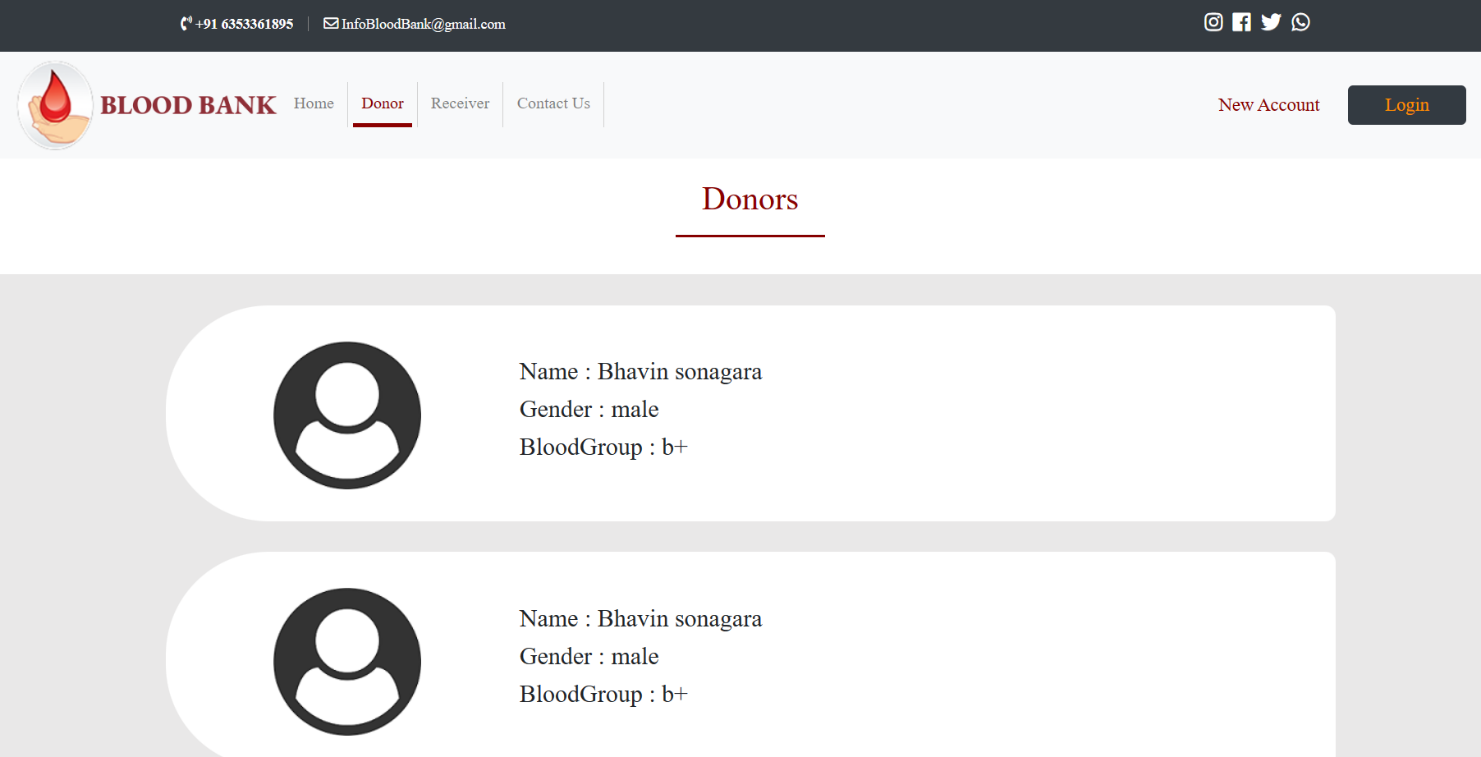
Manual Testing has been performed

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Test Scenario** | **Expected Result** | **Actual result** | **Status** |
| 1 | Login with Correct Credentials | User should able to Log-in | Success message : “Logged-In Successfully” | pass |
| 2 | Login with incorrect Credentials | User should not able to Log-in | Error message : “Sorry invalid credentials! ” | pass |
| 3 | Search for blood bank | Shows blood bank | Shows blood bank on the board | pass |
| 4 | Without Login search for a blood bank | Can’t display blood bank | Shows error message : “Login required” | pass |
| 5 | Request a blood sample | Application should a created and show status | Shows success message : “application created” | pass |
| 6 | Change application status | Application status should be change and receiver get a blood sample | Application status changed successfully and receiver get a blood sample | pass |
| 7 | Blood stock not present and accept application | Application status should be change to decline and receiver not get a blood sample | Shows error : “requested blood sample is less than the present blood sample” | pass |
| 8 | Update profile | Profile should be updated | Success message : “Profile updated successfully” | pass |
| 9 | Logged-in as a user and use route of admin | Error-404 Page | redirect 404-Not Found Page | pass |
| 10 | logout | User should be logged out and restricted from the system until next login. | User is successfully logged out and not able to access the system without signing again | pass |

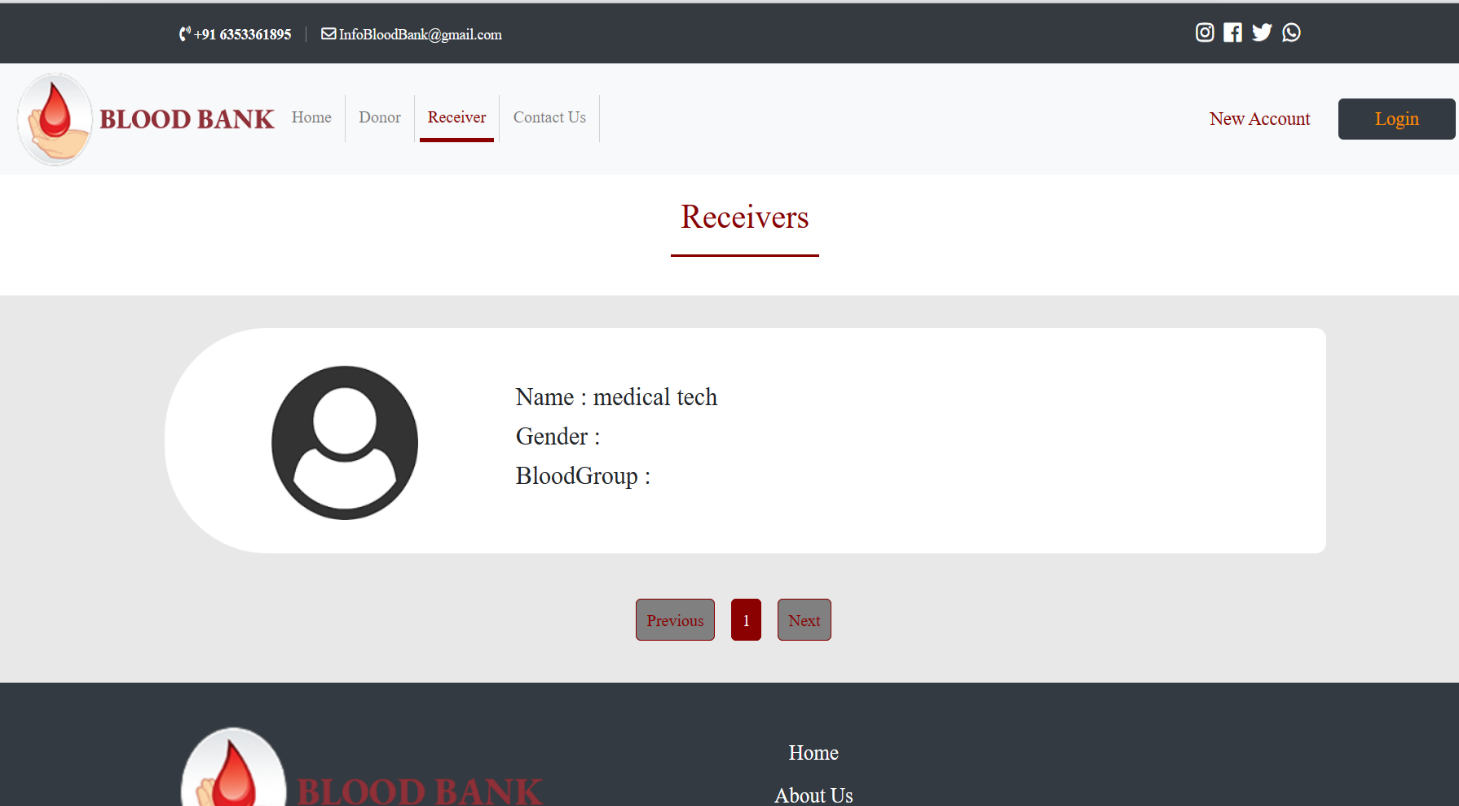
**7. Screenshots**



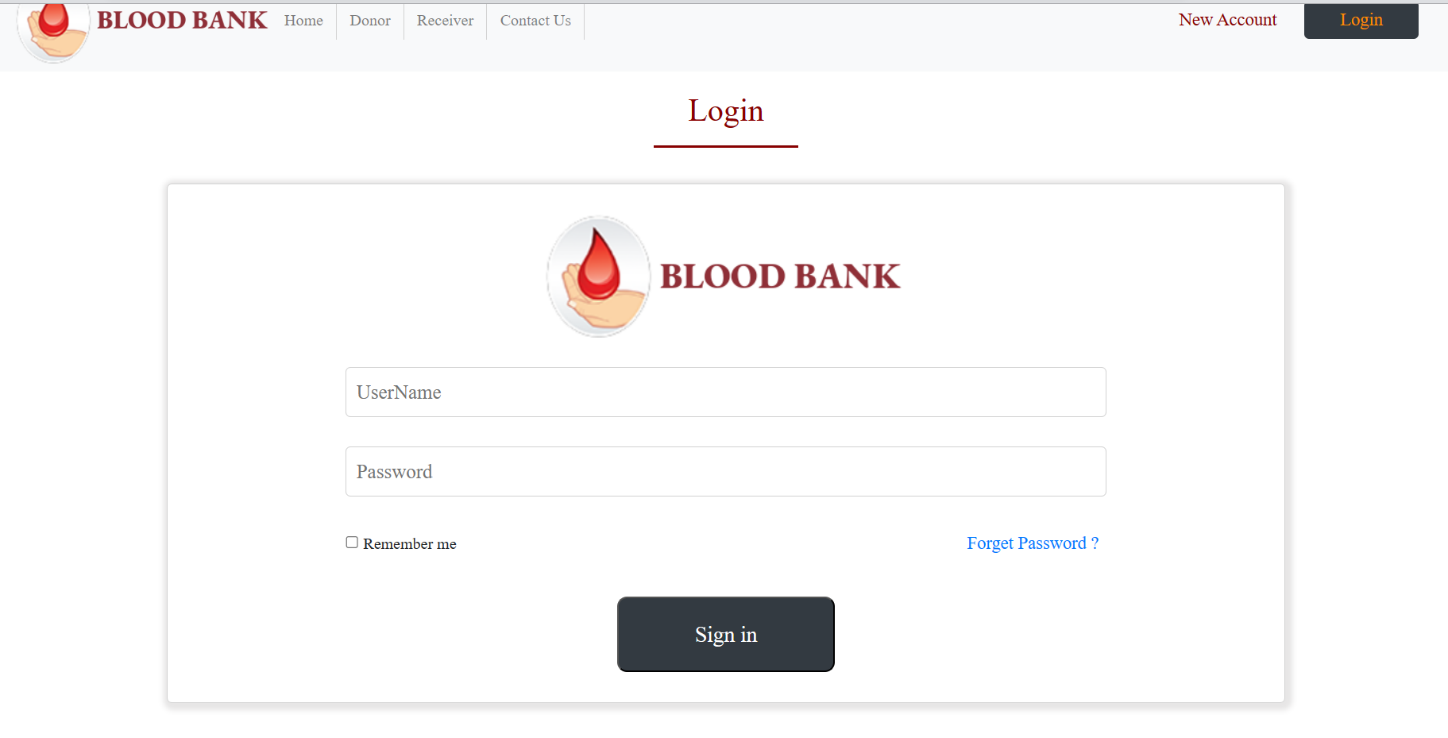
**7.1 Home Page**



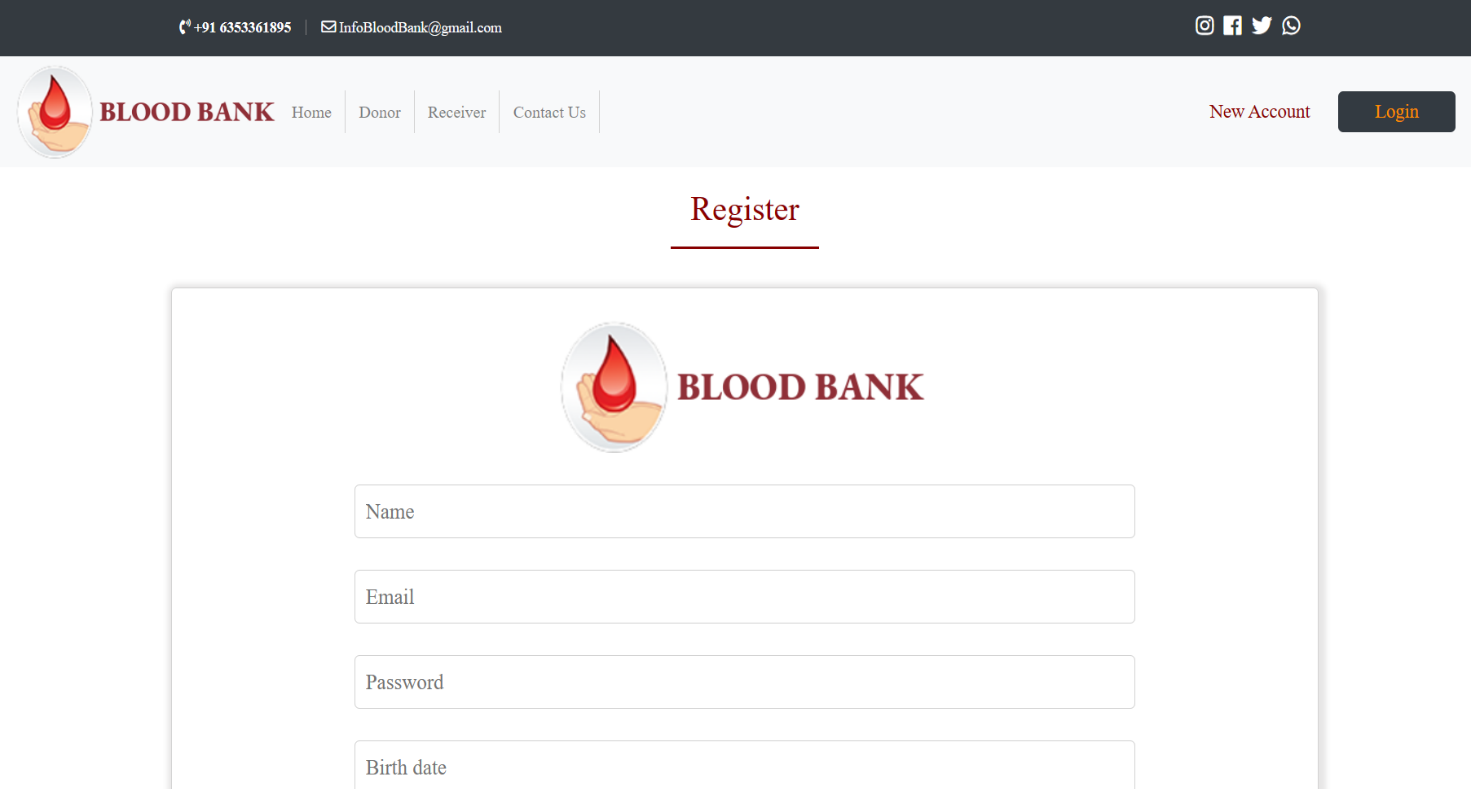
**7.2 Display Donors**



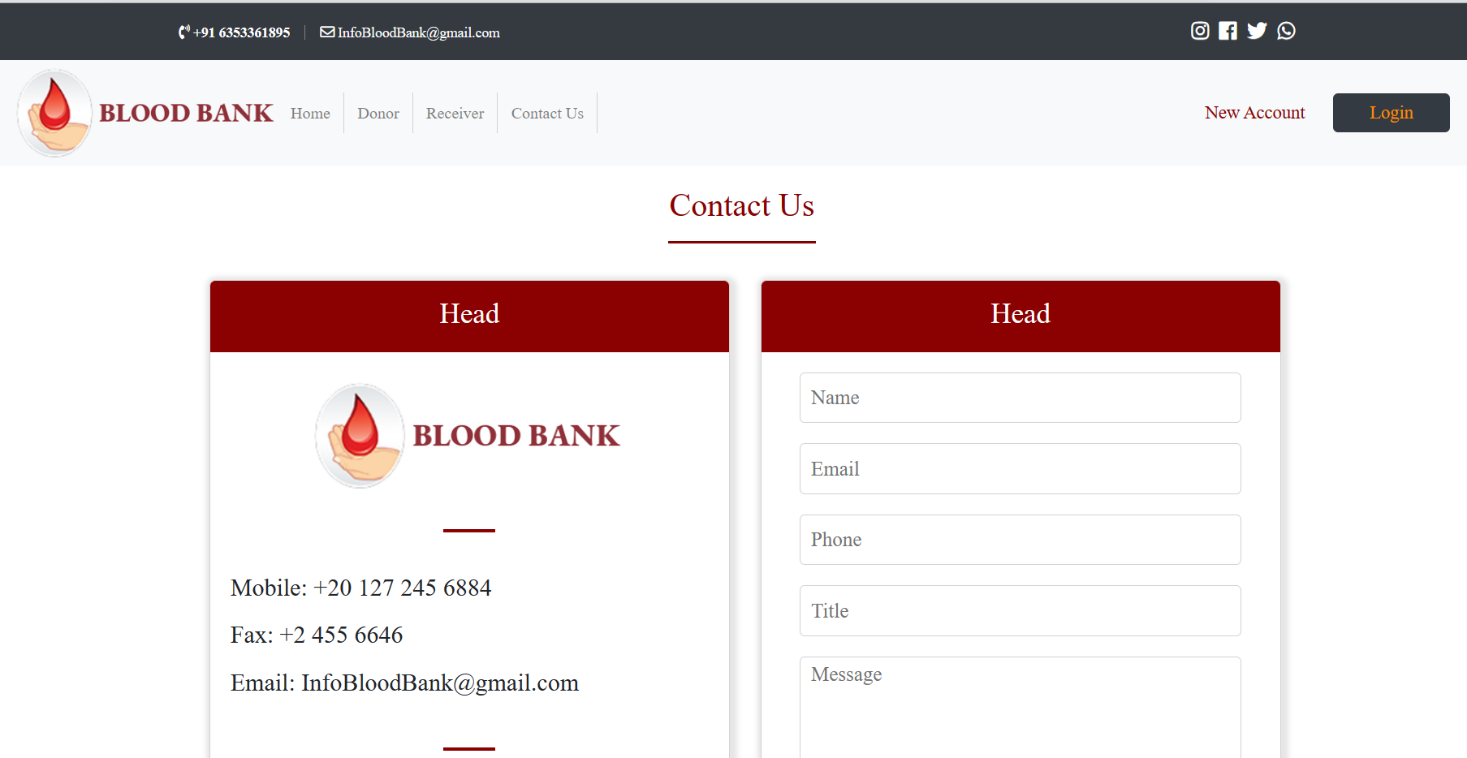
**7.3 Display Receiver**



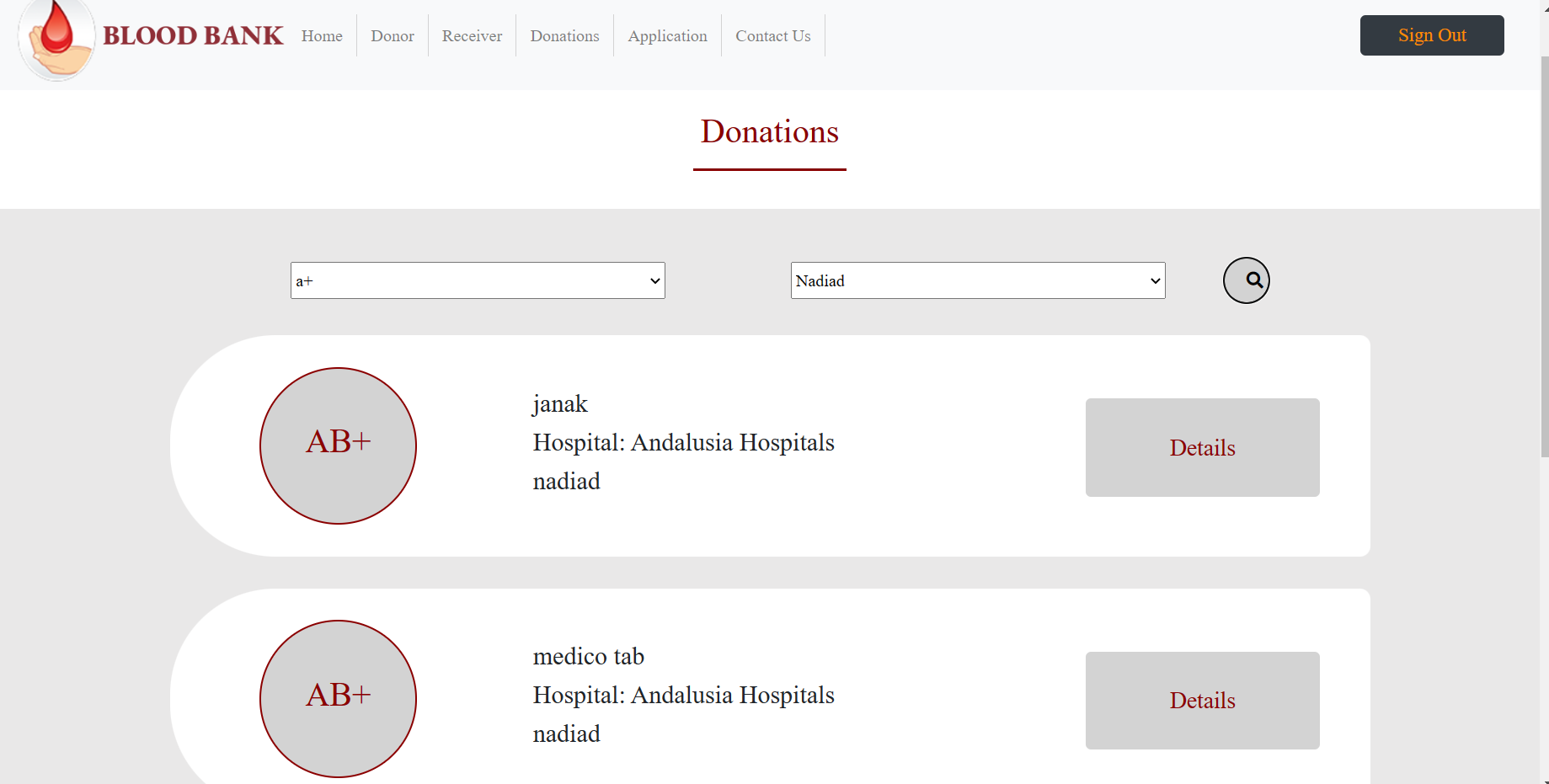
**7.4 Login Page**



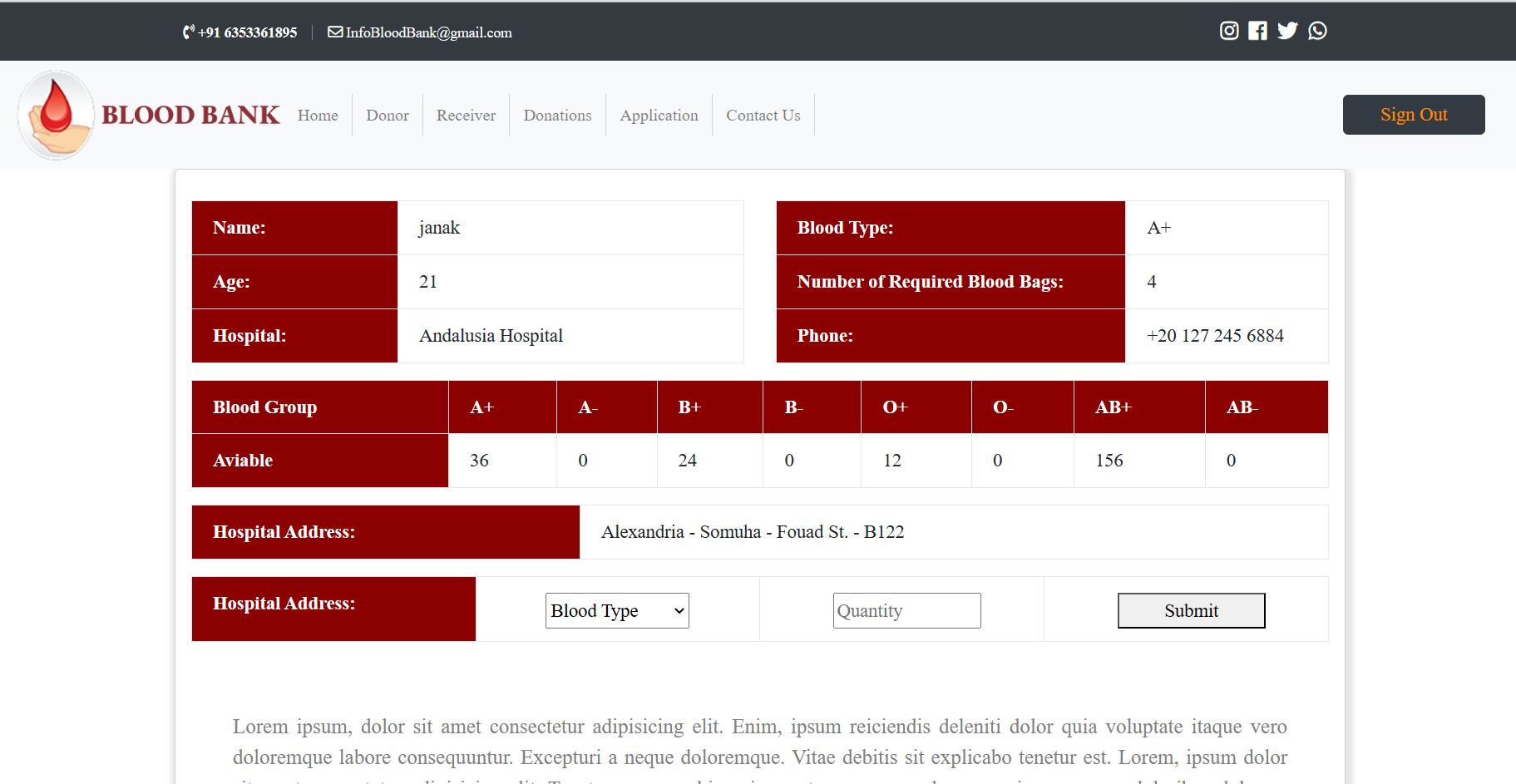
**7.5 Register Page**



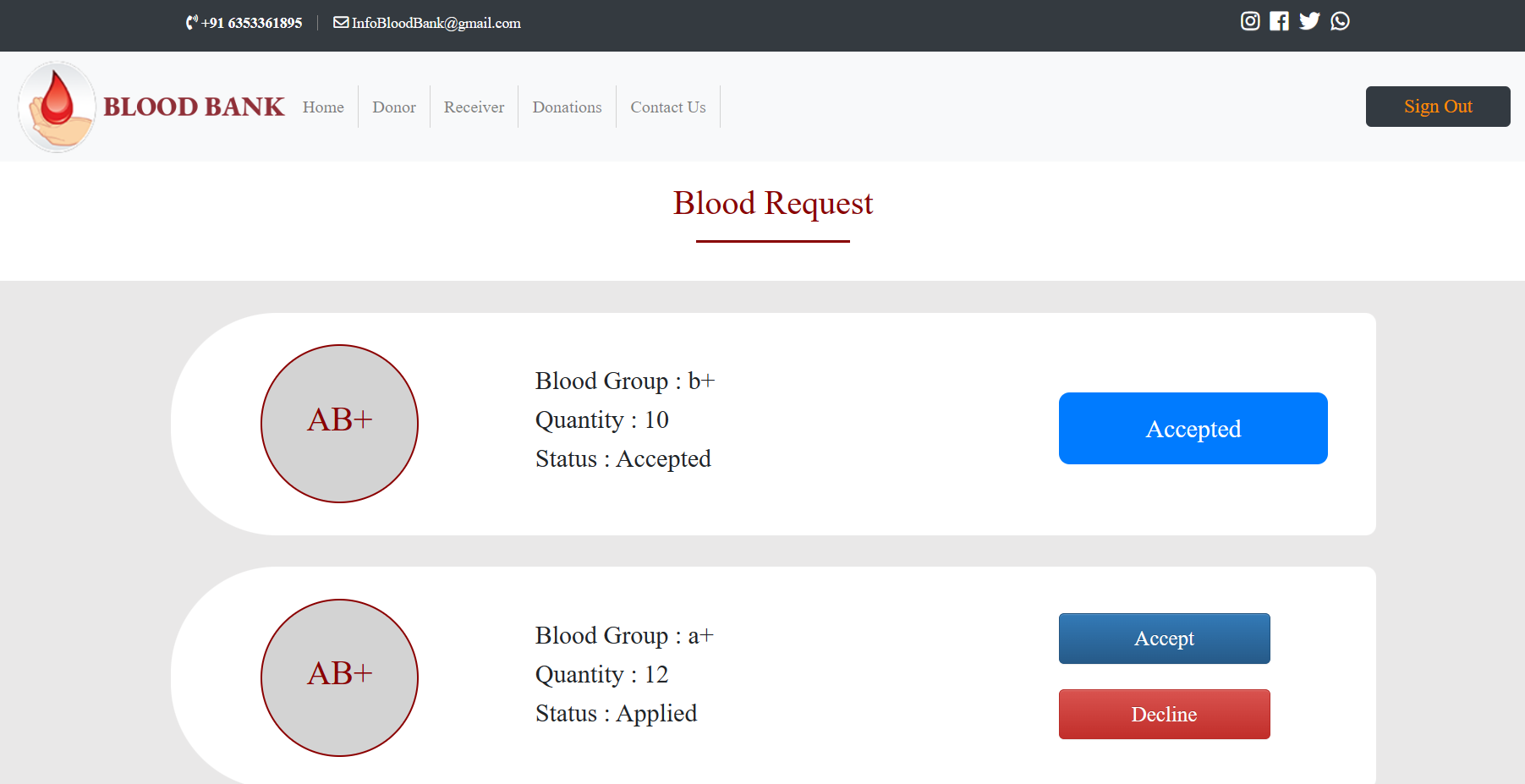
**7.6 Contact us Page**



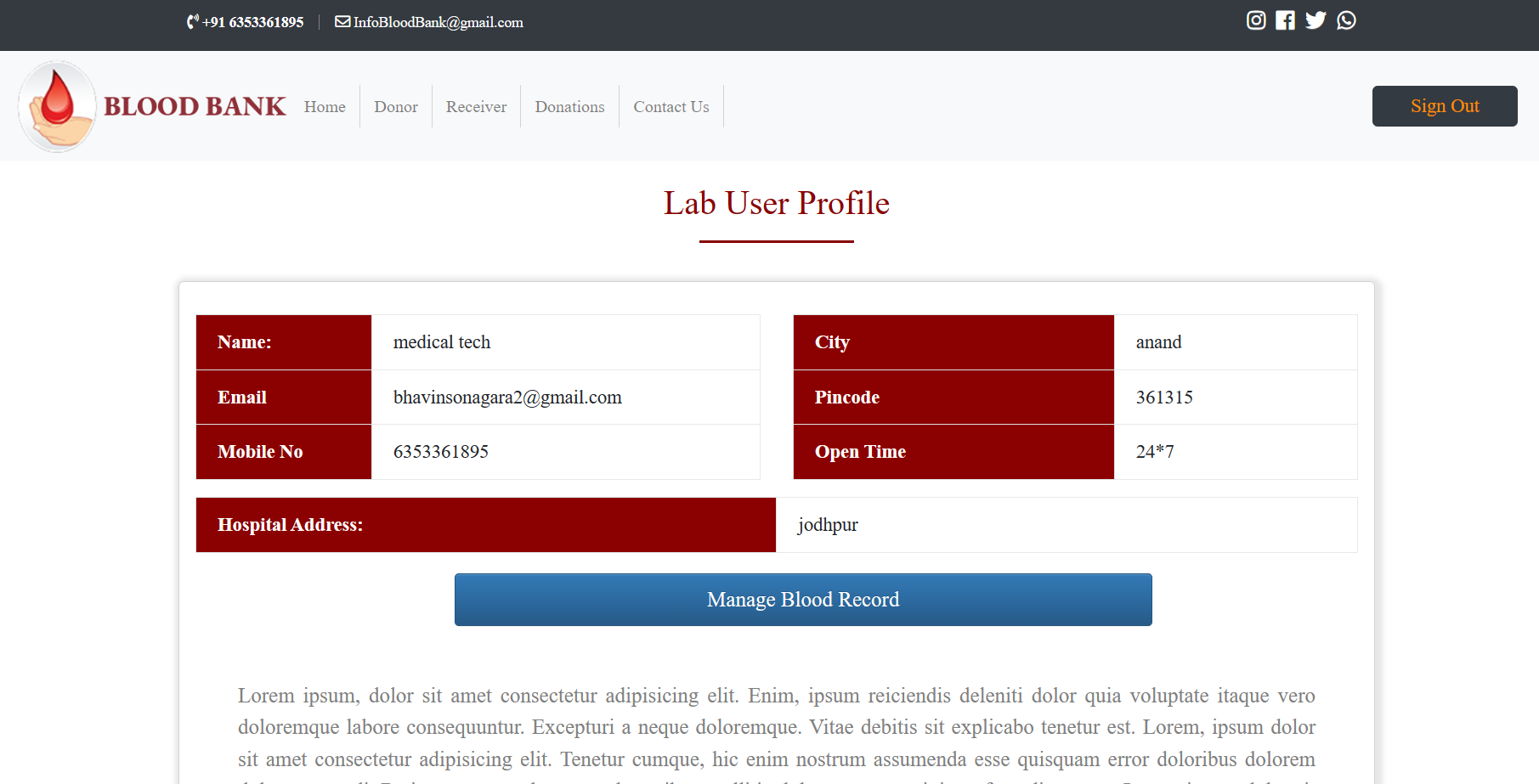
**7.7 Search Page**



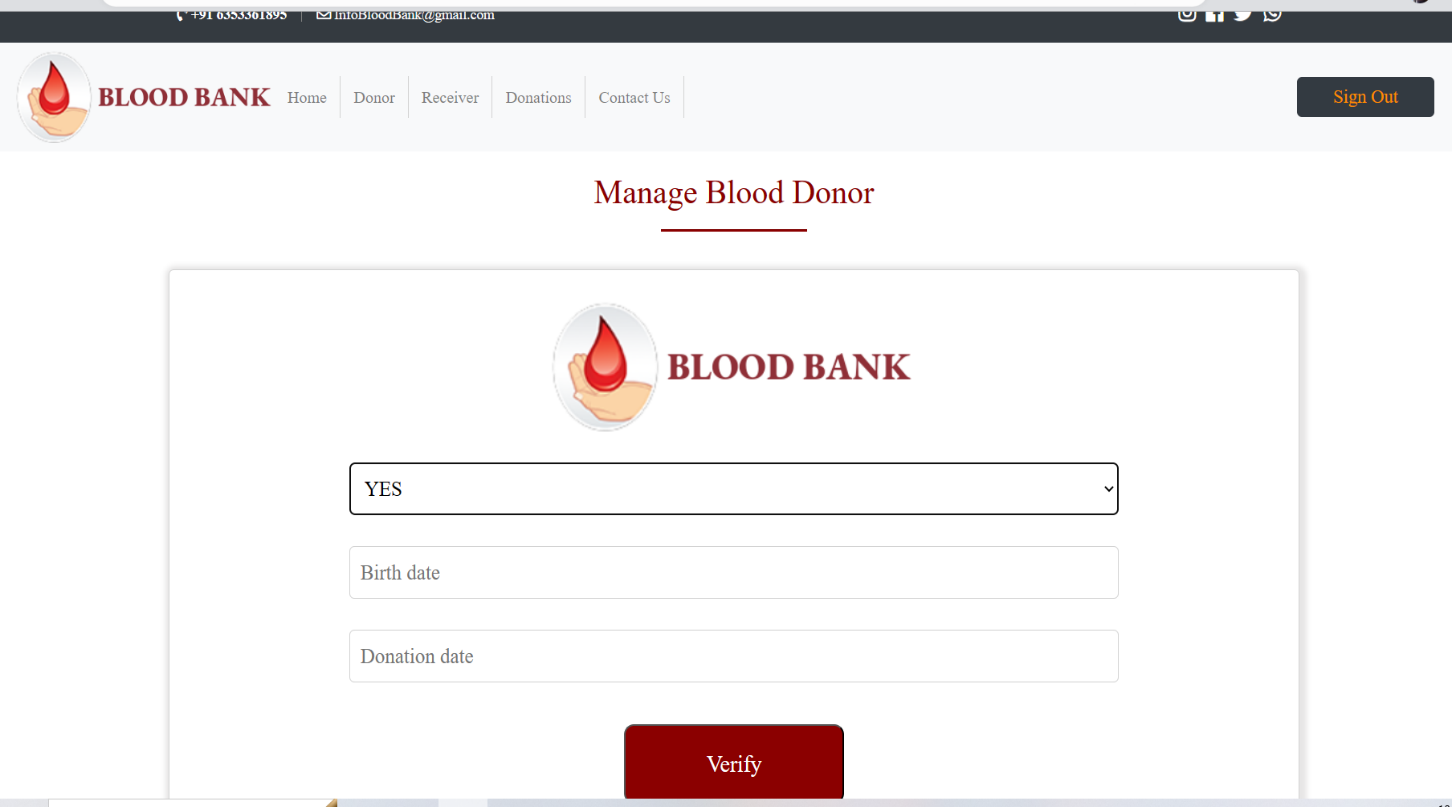
**7.8 Lab Details Page**



**7.9 Blood Request Page**



**7.10 Lab User Profile Page**



**7.11 Manage Donor Page**

**8. Conclusion**

The functionalities implemented in the system were done after understanding  
all the system modules according to the requirements.

Functionalities that are successfully implemented in the system are :

* Authentication based on JWT token and Email-Verification
* Session Management
* Forgot Password
* Logout

User Functionalities:

* Create/Update/View Profile
* Search blood bank
* Create request
* Show request status

Blood Bank Functionalities:

* Create/Update/View Profile
* Increase/Decrease blood stock
* Change a request status
* Show donor/receiver

Admin Functionalities:

* Create/Update/View Profile
* Add/delete blood bank
* Show donor/receiver
* Add/delete blood donation camp

**9. Limitations and Future Extensions**

9.1 Limitations:

Currently, we have not implemented chatting functionality between donor and user due to lack of time and our site is not that much responsive.

9.2 Future Extensions:

System can be extended to implement chatting functionality between  
user and donor.

It can also be extended to make it fully responsive or to change UI part to make  
it more user friendly.

**10. Bibliography**

10.1 Websites :

“React Tutorial.” W3Schools Online Web Tutorials,  
<https://www.w3schools.com/REACT/DEFAULT.ASP>

“NodeJS Tutorial.” W3Schools Online Web Tutorials,  
<https://www.w3schools.com/nodejs/>

“Mongoose v6.2.9: Schemas.” Mongoose ODM v6.2.9,  
<https://mongoosejs.com/docs/guide.html>

“Express Routing.” Express - NodeJS Web Application Framework,  
<https://expressjs.com/en/guide/routing.html>

“Stack Overflow - Where Developers Learn, Share, & Build Careers.”  
Stack Overflow, <https://stackoverflow.com/>

10.2 Youtube Channels :

“ReactJS Tutorials in Hindi - YouTube.” YouTube, CodeWithHarry  
<https://youtube.com/playlist?list=PLu0W_9lII9agx66oZnT6IyhcMIbUMNMdt>

“Login Page with React and NodeJS + Email Verification and Forgot Password. -  
YouTube.” YouTube, ToThePointCode  
<https://youtube.com/playlist?list=PLk8gdrb2DmChrL50moKeFNAVnFqZ3GLF7>