



# **Dharmsinh Desai University, Nadiad**

Faculty of Technology, Department of Computer Engineering

B. Tech. CE Semester - VI  
Subject: (CE - 621) System Design Practice

Project Title: **Blood Donation**

**By:**

Prince Jodhani :- CE049  
Id :- 17CEUON062

**Guided By:**

Prof. Sheetal S. Shah



# **Dharmsinh Desai University, Nadiad**

Faculty of Technology, Department of Computer Engineering

## **CERTIFICATE**

This is to certify that System Design Practice entitled “**Blood Donation**” is the bonafied report of work carried out by

**1) Prince Jodhani (17CEUON062)**

Of Department of Computer Engineering ,Semester VI , academic year 2019-2020, under our supervision and guidance.

---

### **Guide**

**Prof. Sheetal S. Shah**  
Assistant Professor  
Computer Engineering,  
Dharmsinh Desai University,  
Nadiad

### **HOD**

**Dr. C. K. Bhensdadia**  
Head of Department  
Computer Engineering,  
Dharmsinh Desai University,  
Nadiad

# *Table of Contents*

<b>1 Abstract.....</b>	<b>4</b>
<b>2 Introduction .....</b>	<b>5</b>
2.1 Project Details: Brief Introduction.....	5
2.2 Technology and Tools Used.....	5
<b>3 Software Requirement Specifications .....</b>	<b>6</b>
3.1 Scope .....	6
3.2 System Functional Requirements .....	6
3.3 Other Non-Functional Requirements .....	7
<b>4 Design .....</b>	<b>9</b>
4.1 Usecase Diagram.....	9
4.2 Class Diagram .....	9
4.3 Sequence Diagram.....	10
4.4 Activity Diagram .....	10
4.5 E-R Diagram.....	11
4.6 Data Dictionary.....	12
<b>5 Implementation Details .....</b>	<b>13</b>
5.1 Activities Description .....	13
<b>6 Testing .....</b>	<b>15</b>
6.1 Testing Method .....	15
6.2 Test Cases .....	15
<b>7 Screen-shots of the System .....</b>	<b>17</b>
<b>8 Conclusion .....</b>	<b>20</b>
<b>9 Limitations of System .....</b>	<b>21</b>
<b>10 Future Extensions to the System.....</b>	<b>21</b>
<b>11 Bibliography .....</b>	<b>22</b>

---

## **Abstract**

---

As we all know how blood is important to save other people's life and importance in our life also. So, for donating blood is more important for all peoples. From this aspect I decided to create one web application. By using this app people can register to donate and request blood very easily and it saves time.

---

# Introduction

---

## 2.1. Brief Introduction



Raktadan is an web application. Through our app donor can donate blood or he/her can also raise request for blood by using our application. By using our app anyone can organize blood donation camp by filling form of donate blood. If we have shortage of blood then we can send mail to our active donors. Anyone can get help of how to organize camp through our app. And if any patient want to get blood then he/she have to fill up the form of request.

## 2.2. Tools/Technologies Used

Technologies : Angular,

Node,

Mongo DB,

HTML, CSS, Bootstrap 4

Tools : Visual Studio Code

---

## 3. Software Requirement Specifications

---

### 3.1 Product Scope

This web application is time consuming for who have to donate blood,request blood or organize blood donation camp.

#### Types of Users:-

1. End user (Donor or Patient)

### 3.2 System Functional Requirements

#### *1) End user (Donor):-*

##### R.1.1: Signup

Description :- Any user can create their account in our app by filling up signup form.

Inputs :- Write Personal Details.

Output :- Account created successfully.

##### R.1.2: Login

Description :- User can login in our app using their email & password.

Input :- Enter valid details.

Output : You are logged in successfully.

##### R.1.3: View Pages

Description :- User can view our pages and details of our application and about how to organize camp.

Inputs :- Click on appropriate tab.

Output :- Content of selected tab.

**R.1.4: Request Blood**

Description :- Patient can send request of appropriate blood.

Inputs :- Enter details properly.

Output :- Your request sent successfully.

**R.1.5: Organize Camp**

Description :- User can organize camp at any location.

Inputs :- User enter city, mobile number, no. of people, date.

Output :- Data submitted.

**R.1.6: Donate Blood**

Description :- User can donate blood at nearest blood center and we send details of that blood center on his/her mail address.

Inputs :- Enter Your Details.

Output :- Send mail with details of blood center.

**R.1.7 : Contact us**

Description :- User can contact our team for any inquiries or queries.

Inputs :- User can write their query.

Output :- Send mail with solution.

**R.1.8 : Delete Account**

Description :- User can delete his/her account.

Inputs :- User Selection.

Output :- Account will be deleted.

**3.3 Other Non-functional Requirements****(1) Performance**

Performance The system must be interactive and the delays involved must be less. So in every action – response of the system, there are no immediate delays. In case of opening App components, of popping error messages and saving the settings or sessions there is delay much below 7-8 seconds.

## **(2) Safety**

User details should be securely stored to the server.

## **(3) Database**

System requires to access users data fastly to maintain the performance.

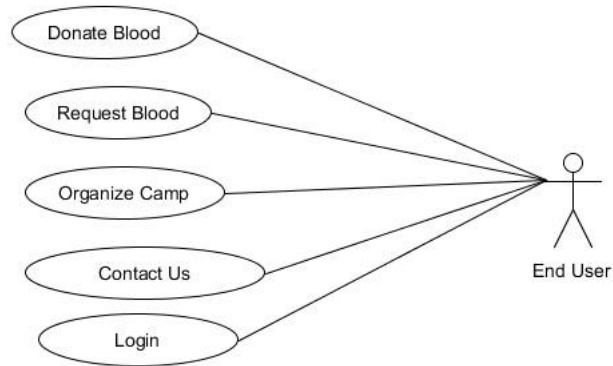


---

## 4 Design

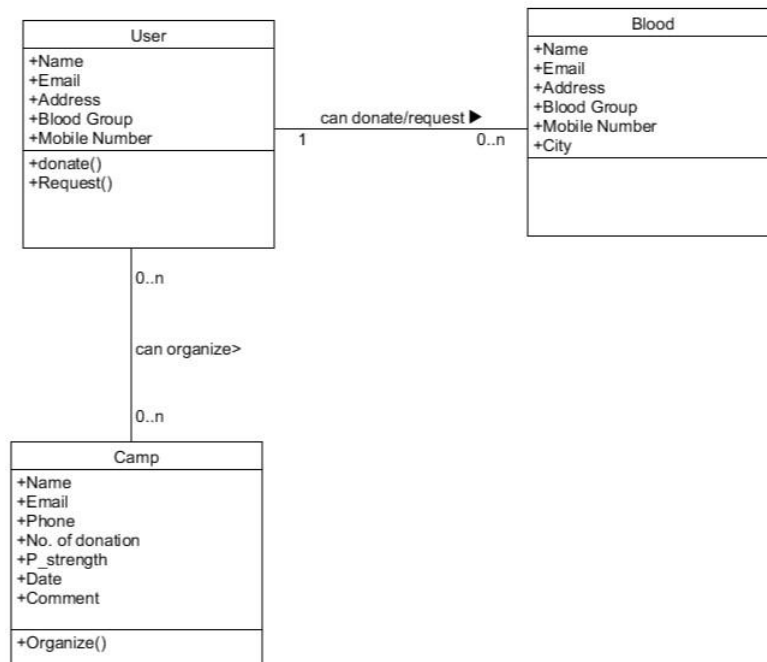
---

### 1.1 Use Case Diagrams

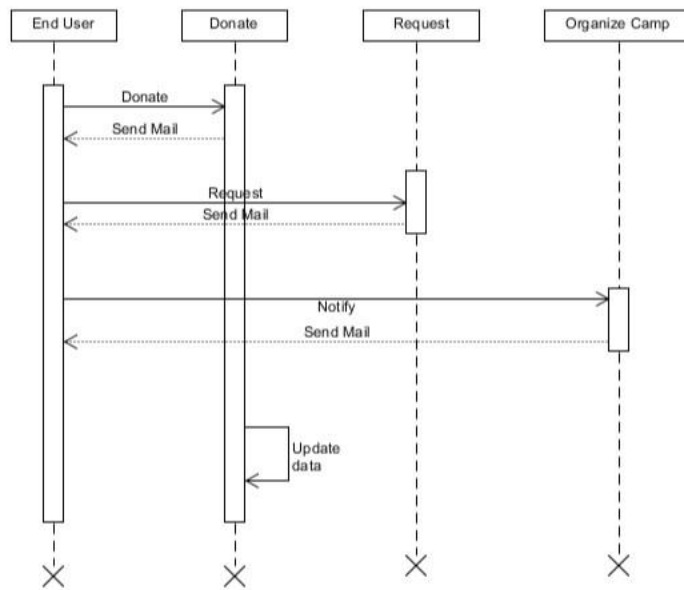


---

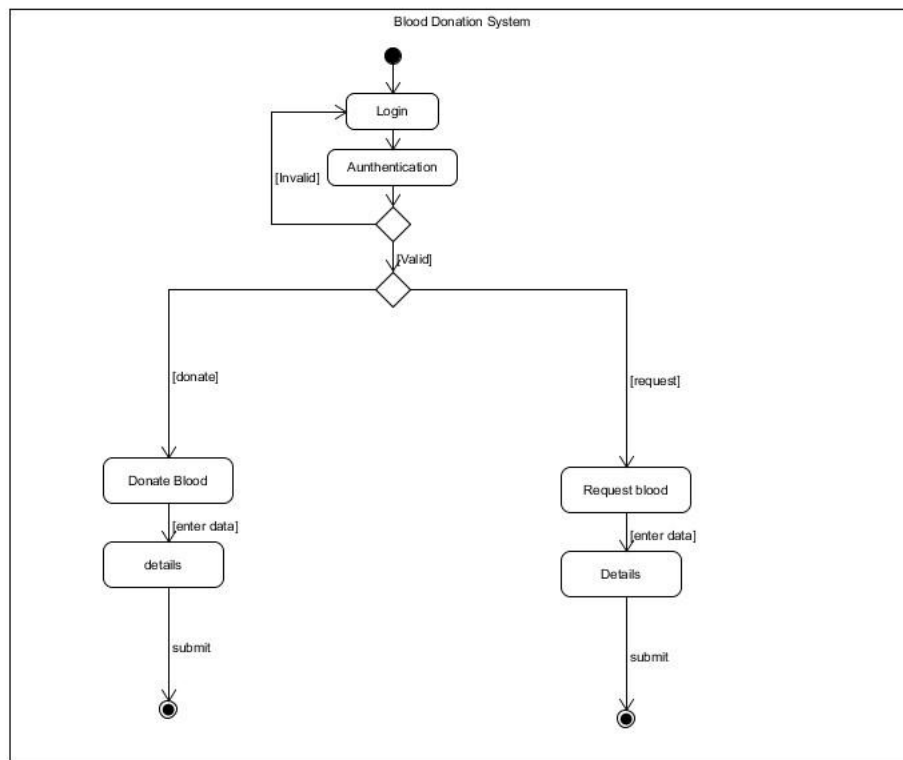
### 1.2 Class Diagrams



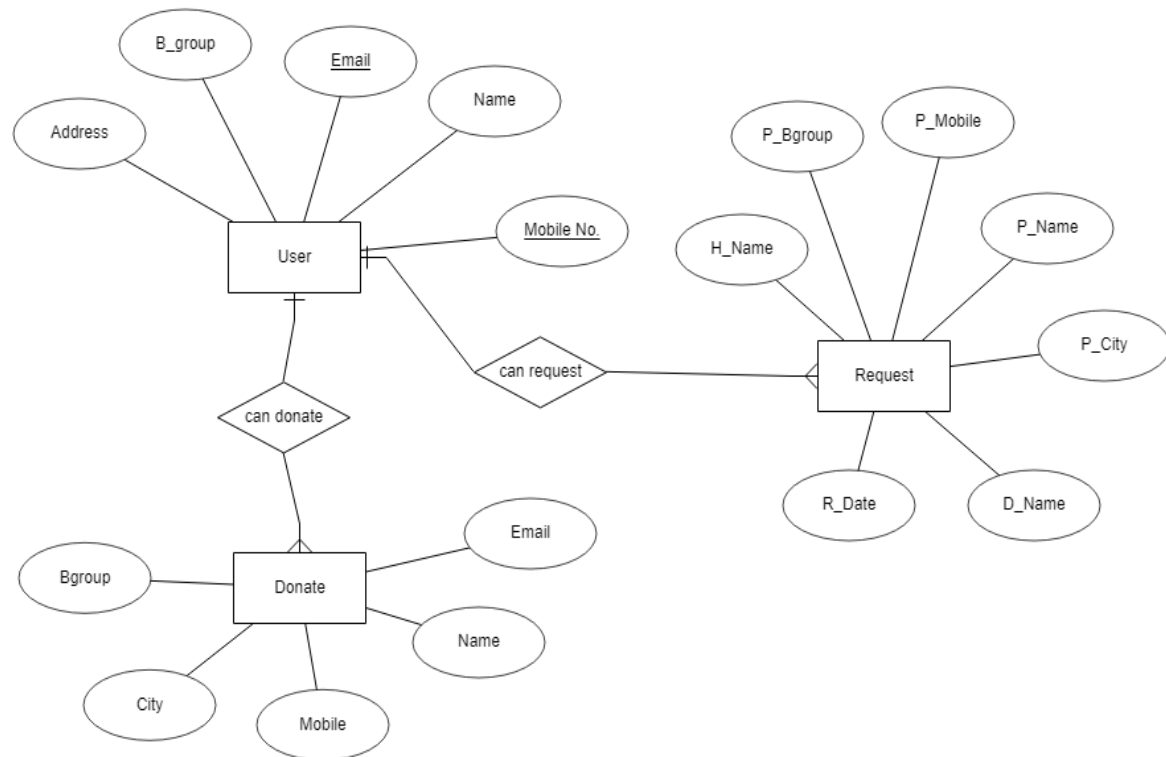
## 1.3 Sequence Diagrams



## 1.4 Activity Diagrams



## 1.5 E-R Diagrams



## 1.6 Data Dictionary

### Donor / Patient

Sr. No	Name	Data Type	Width	Required	Unique	PK/FK
1	Email	VARCHAR2	40	YES	YES	PK
2	Name	VARCHAR2	50	YES	-	-
3	Blood group	VARCHAR2	10	YES	-	-
4	Address	VARCHAR2	100	YES	YES	-
5	Mobile no	NUMBER	15	YES	YES	-
6	Password	VARCHAR2	20	YES	YES	-

### Request

Sr. No	Name	Data Type	Width	Required	Unique	PK/FK
1	P_Name	VARCHAR2	40	YES	-	PK
2	P_Number	NUMBER	15	YES	YES	PK
3	P_Bgroup	VARCHAR2	10	YES	-	-
4	P_City	VARCHAR2	15	YES	-	-
5	H_Name	VARCHAR2	50	YES	-	-
6	D_Name	VARCHAR2	20	YES	-	-
7	R_date	DATE	10	YES	-	-

### Organize Camp

Sr. No	Name	Data Type	Width	Required	Unique	PK/FK
1	Name	VARCHAR2	50	YES	-	-
2	Email	VARCHAR2	20	YES	YES	PK
3	Phone	NUMBER	15	YES	YES	PK
4	No. of donation	NUMBER	5	YES	-	-
5	P_strength	NUMBER	5	YES	-	-
6	Date	DATE	20	YES	-	-
7	Comment	VARCHAR2	100	YES	-	-

---

## 2

## Implementation Details

---

### 5.1 Activities Description

#### □ **Sign Up Activity:**

This activity is used to store user's data to the database and enables the user to login to the system. All the fields in this module contains required validations and it uses user model to store data.

Input : User's Informations

Output : User Registered and redirect to login page

Processing : Validating user's data and then storing them to database

#### □ **Login Activity:**

This module takes users credentials and then verifies it with registered users , if user details is not match with database then user can't login.If user is not registered then he/she redirect to signup activity.

Input: User Credentials

Output: Logging user.

Processing: Verifying user credentials with the database.

□ **Home Activity:**

This activity is our home page which will load after 7-8 seconds of launching the app .

From here user can registered & login (if already registered), can donate & request blood.

Input : User Selection

Output :Corresponding response

□ **Donate Blood Activity:**

This activity takes details from user and send it into nearest blood center and then after confirmation by blood bank we send details of bank to user via email.

Input : User Selection

Output : Correspondingresponse

□ **Request Blood Activity:**

This activity takes details from user and their blood details send it to active donors and after confirm by donor we send both(requester & donor) to blood bank.

Input : User Selection

Output : Correspondingresponse

### 3

## Testing

### 6.1 Testing Method:

We have performed Black - box testing for the testing purpose.

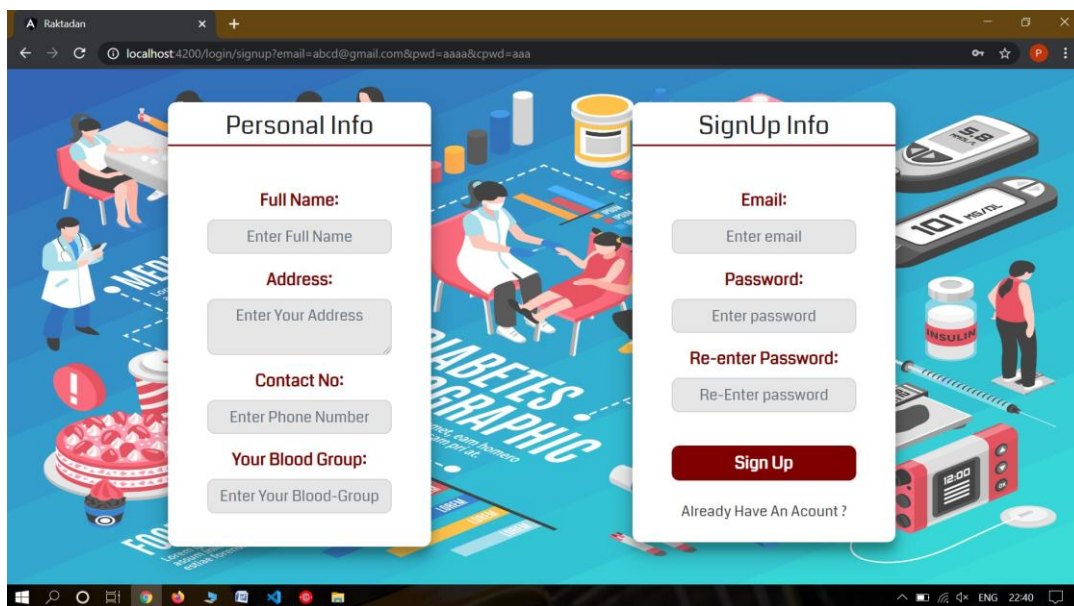
### 6.2 Test Cases:

For Registration:

Email: email address.(we put validation for email)if user forget to write @ or . or other requirement then it can occur error.

Password: both password field(password & Confirm password) are not matching then data of all field are removed otherwise it indirect to login page.

(I write Password:aaaa & Confirm password:aaa)

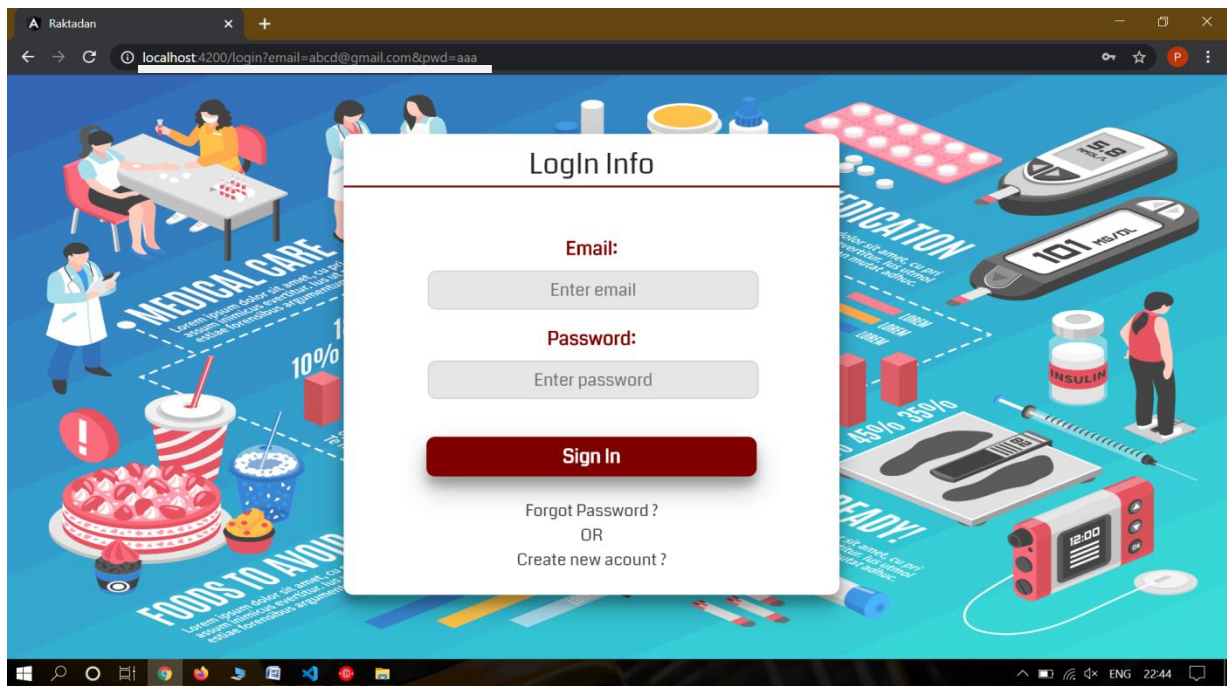
The screenshot shows a web browser window with two overlapping registration forms. The 'Personal Info' form on the left has fields for Full Name, Address, Contact No, and Your Blood Group. The 'SignUp Info' form on the right has fields for Email, Password, and Re-enter Password, along with a 'Sign Up' button and a link for 'Already Have An Account?'. The background features a medical-themed illustration with a doctor, a patient, and various medical devices like a glucose meter and insulin pump.

For Login:

Email: email address.(we put validation for email)if user forget to write @ or . or other requirement then it can occur error.

Password: both password field(password & Confirm password) are not matching then data of all field are removed otherwise it indirect to home page.

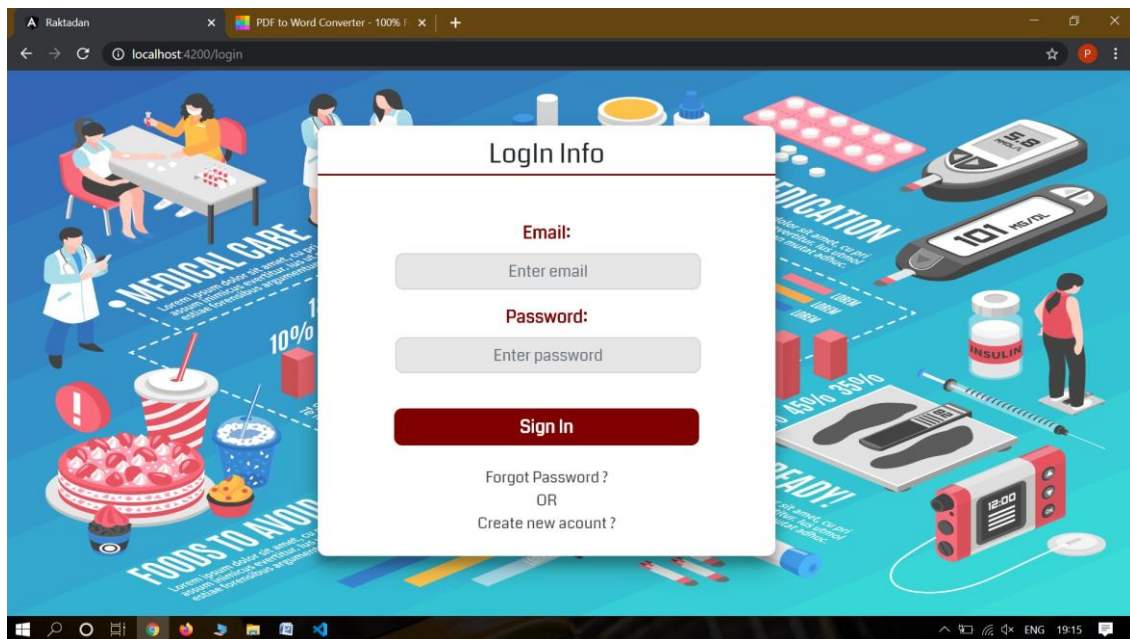
Email:abcd@gmail.com  
Password:aaaa





## 4 Screenshots

### (1) Login



A screenshot of a web browser displaying the login page of a medical application. The browser's address bar shows 'localhost:4200/login'. The page features a colorful, isometric illustration of medical professionals and patients. Overlaid on this is a white 'Login Info' form. The form contains fields for 'Email' and 'Password', each with a placeholder text 'Enter email' and 'Enter password' respectively. Below these fields is a red 'Sign In' button. At the bottom of the form, there are links for 'Forgot Password?', 'OR', and 'Create new account?'.

Browser tabs: Raktadan, PDF to Word Converter - 100% F

Address bar: localhost:4200/login

Form Title: Login Info

Fields:

- Email: Enter email
- Password: Enter password

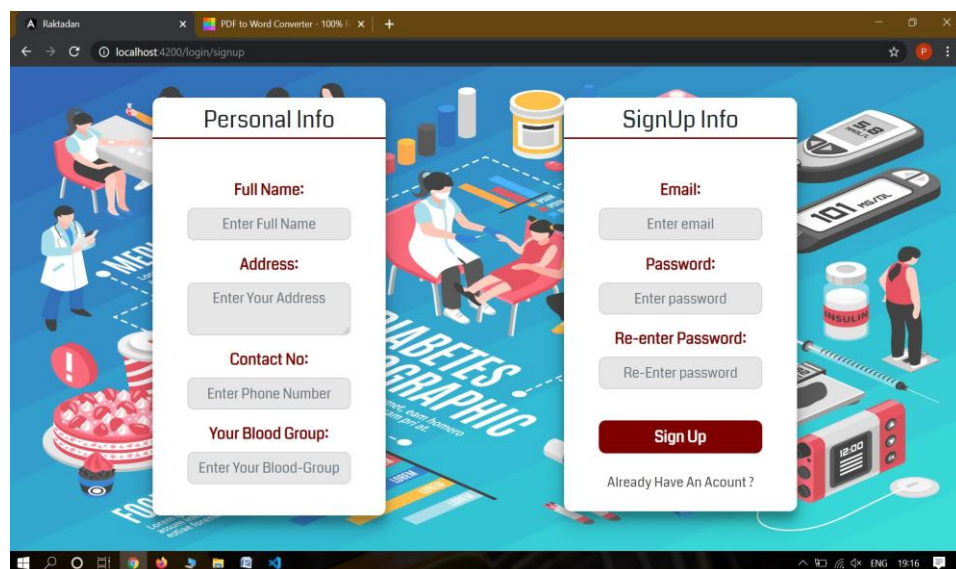
Buttons:

- Sign In

Links:

- [Forgot Password?](#)
- OR
- [Create new account?](#)

### (2) Signup



A screenshot of a web browser displaying the signup page of a medical application. The browser's address bar shows 'localhost:4200/login/signup'. The page features the same colorful, isometric illustration as the login page. Overlaid on this are two white forms: 'Personal Info' and 'SignUp Info'. The 'Personal Info' form has fields for 'Full Name', 'Address', 'Contact No.', and 'Your Blood Group', each with a placeholder text. The 'SignUp Info' form has fields for 'Email', 'Password', and 'Re-enter Password', each with a placeholder text. Below the 'Re-enter Password' field is a red 'Sign Up' button. At the bottom of the 'SignUp Info' form, there is a link for 'Already Have An Account?'.

Browser tabs: Raktadan, PDF to Word Converter - 100% F

Address bar: localhost:4200/login/signup

Forms:

**Personal Info**

- Full Name: Enter Full Name
- Address: Enter Your Address
- Contact No: Enter Phone Number
- Your Blood Group: Enter Your Blood-Group

**SignUp Info**

- Email: Enter email
- Password: Enter password
- Re-enter Password: Re-Enter password

Buttons:

- Sign Up

Links:

- [Already Have An Account?](#)

### (3) Request Blood

**Request Blood**

Patient Name:

Patient Number:

Blood Group:

Type blood group from the following types:  
A+, AB+, A-, B+, B-, O+, O-

City:

City:

Hospital Name & Address:

Doctor Name:

When Required?:

**Request**

## (4) Organize Camp

The screenshot shows a web browser window with the URL `localhost:4200/contact`. The browser's address bar and tabs are visible at the top. The website's navigation bar includes links: Home, About, Organize Camp, About Blood, Contact Us, Sponsor Us, and Raise Request. The main content area features a dark red background with a pattern of red blood cells. On the left, the text 'RED BLOOD CELLS VECTOR' is displayed. On the right, there is a form with the following fields and labels:

- Name: Enter your name
- Email: Enter email
- Phone: Enter your mobile no.
- Estimated no. of donation: Enter value
- People Strength: Enter people strength
- Date: dd-mm-yyyy
- Additional Comment: Enter your comments

A red 'Submit' button is located at the bottom of the form.

## (5) Mailing code

```
var transporter = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    user: 'pjodhani0712@gmail.com',
    pass: 'REDACTED'
  }
});

var mailOptions = {
  from: 'pjodhani0712@gmail.com',
  to: newuserModel.email,
  subject: 'Hello' + ", " + newuserModel.name,
  text: 'Your Account for RAKTADAN is registered succesfully!\n Thank you for registered.\nHave
};

transporter.sendMail(mailOptions, function (error, info) {
  if (error) {
    console.log(error);
  } else {
    console.log('Email sent: ' + info.response);
  }
});
```

---

## Conclusion

---

The functionality implemented in the system was done after understanding all the system modules according to the requirements.

Functionalities that are successfully implemented in the system are:

- User Registration containing all the necessary validations on fields.
- Login
- Logout
- Donate Blood
- Organize Camp
- Raise Request
- Contact us
- Home Page

We have learned new technology of mailing user and we use it in our app. Also we successfully learnt node js, angular js & MongoDB.

---

## **Limitations and Future Enhancements**

---

### **▢ Limitations**

- Currently user can not organize camp because we have to call user for confirmation.(call or send messages are paid currently so... )

### **▢ Functionalities not Implemented**

- Showing up active & recent donor list

### **▢ Future Extension**

- Text Message functionality.
- List of active donors and recent donors.

---

# Bibliography

---

▯ **Websites:**

1. <https://stackoverflow.com/questions/> - For Solving errors & questions.

▯ **Useful Links:**

1. <https://www.w3schools.com/>
2. <https://www.npmjs.com/>
3. <https://fonts.google.com/>