A

Major Project

On

INSTANT PLASMA DONOR RECIPIENT CONNECTOR WEB APPLICATION

Submitted to

Jawaharlal Nehru Technological University, Hyderabad

In Partial fulfillment of the requirements for the award of Degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

J.TEJASWINI (187R1A05K6)

U.SREEJA (187R1A05M9)

M.SANJAY (197R5A0524)

Under the esteemed guidance of

G.KALPANA DEVI

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**INSTANT PLASMA DONOR RECIPIENT CONNECTOR WEB APPLICATION"** being submitted by **J.TEJASWINI (187R1A05K6), U.SREEJA (187R1A05M9), M.SANJAY (197R5A0524)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of Bonafide work carried out by him/her under our guidance and supervision during the year 2021-22.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Mrs. G.Kalpana Devi
(Assistant Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Mrs. G.Kalpana Devi,** Assistant Professor, for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **, Mr. A. Uday Kiran, Mr. J. Narasimha Rao, Dr. T. S. Mastan Rao, Mrs. G. Latha, Mr. A. Kiran Kumar** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

J.TEJASWINI (187R1A05K6)
U.SREEJA (187R1A05M9)
M.SANJAY (197R5A0524)

ABSTRACT

The world is suffering from COVID-19 crisis, and we have not found any vaccine yet. But there is another scientific way from which we can help to lower the death ratio or help the COVID-19 affected person is by donating plasma from recovered patients. With no approved antiviral treatment plan for the deadly COVID-19 infection, plasma therapy is an experimental approach to treat COVID-19 positive patients and help them recover faster. The therapy is considered to be safe and promising. If a particular person is fully recovered from COVID-19, he/she is eligible to donate their plasma. In the proposed system, donors who need to donate plasma can donate by uploading COVID-19 certificate and blood bank can view donors and can raise requests to donors and the hospital can register/login and can search for the plasma, they can raise requests to blood bank and can get the plasma.

LIST OF FIGURES

LIST OF SCREENSHOTS

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1.INTRODUCTION

1.INTRODUCTION

1.1 PROJECT SCOPE

The scope of this project is the donor who wants to donate plasma can donate by uploading covid-19 certificate and blood bank can view donors and can raise requests to donors and the hospitals can register/login and can raise requests to blood bank and can get the plasma from blood bank.

1.2 PROJECT PURPOSE

The purpose is that due to COVID-19, People need emergency plasma from recovered people to donate plasma in order to recover faster. Our project can give a solution to this problem, Where a person can donate plasma by uploading a recovered certificate to the blood bank.

1.3 PROJECT FEATURES

The main feature of this project is that hospital can check the availability of plasma from the blood bank and they can make a request to the blood bank and get the plasma so that immediate treatment can be done to the patients. A blood bank acts like an interface between the hospital and donor.Hospital and donor are getting the required information from the blood bank.

2.SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

System Analysis is an important phase in the system development process. The System is studied to the minute details and analysed. The system analyst plays an important role as an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified.Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

The solution proposed here aims to solve these disadvantages of the existing system, by developing a user-friendly web application that will help to lower the death ratio and help the COVID-19 people to recover faster by donating plasma from recovered patients, It will work efficiently at emergency situations.

2.2 EXISTING SYSTEM

People have to find them physically by visiting hospitals register books and reaching out to recovered donors homes. Sometimes they will be not available at their places and will be went on work. In this type of scenarios, diseased persons health gets more worsened. This is an expensive and will not work as effectively in emergency situations.

2.2.1 DISADVANTAGES OF EXISTING SYSTEM

- Tedious work
- Expensive
- Requires more man power
- Time consuming

2.3 PROPOSED SYSTEM

In the proposed system, a donor who wants to donate plasma can simply upload his/her covid-19 recovered certificate and can donate plasma to the blood bank. Blood bank after verifying donor certificate can raise a request to the donor once the donor accepts a request they can add the required amount of units they need.Hospital can send requests to blood bank who need emergency plasma to patients and collect the plasma from the blood bank.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Immediate solutions
- Saves time and energy
- Saves money
- Ease of finding

2.4 FEASIBILITY STUDY

- The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.
- Three key considerations involved in the feasibility analysis are:
  1.Economic Feasibility
  2.Technical Feasibility
  3.Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

- This study is carried out to check the economic impact that the system will have on the organization.The amount of fund that the company can pour into the research is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.4.2 TECHNICAL FEASIBILITY

- This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 SOCIAL FEASIBILITY

- The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed as he is the final user of the system.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

1. **Processor - I3/Intel Processor**
2. **Ram - 4 GB (min)**
3. **Hard Disk - Above 4Gb**

2.5.2 SOFTWARE REQUIREMENTS:

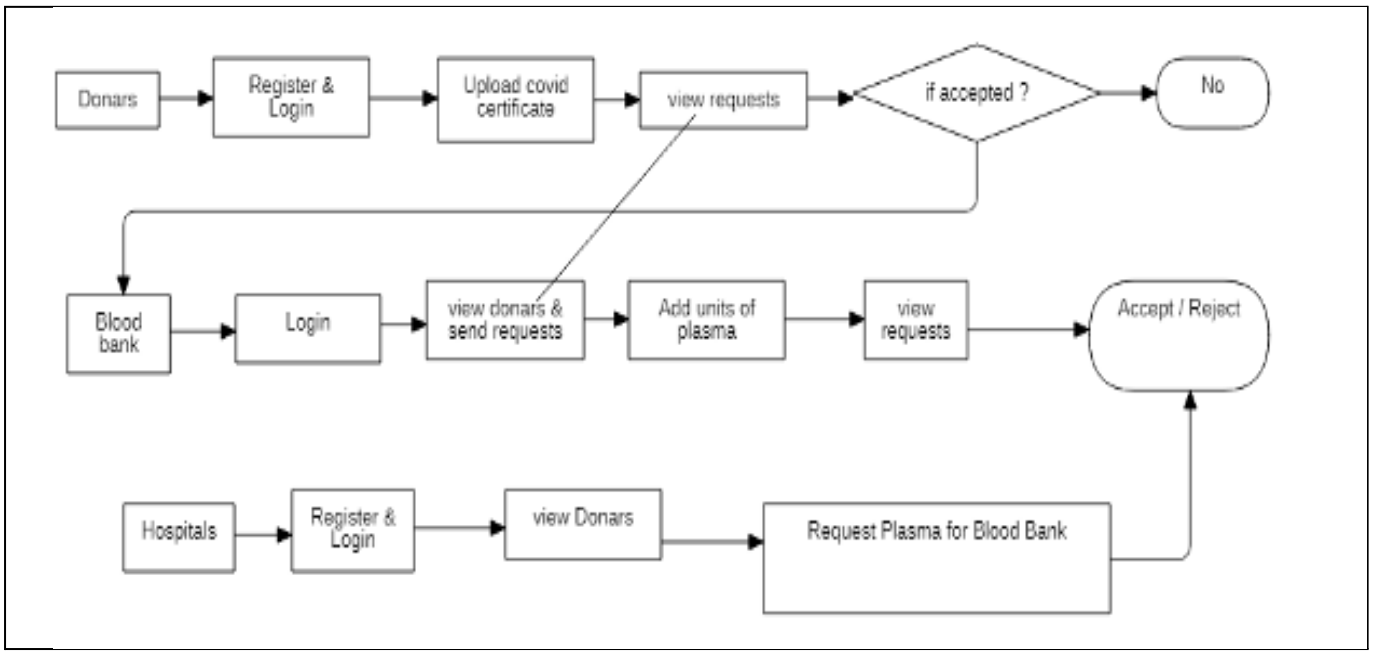
- The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.
- The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

1. Python 3.7 version

3.ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCHITECTURE:



3.1 Project Architecture of Instant Plasma Donor Recipient Connector Web Application

3.2 MODULES DESCRIPTION:

1.Donor:

Login:

Donors can login with their credentials and donors who need registration can register and login.

Upload Certificate:

Donors can upload covid-19 negative certificate after they login.

View Profile:

Donors can view their profile i.e., name, certificate, blood group, age.

View Requests:

Donors can view requests sent by blood bank and donors can either accept/reject the requests.

2.Blood bank:

Login:
Blood bank can login with their credentials.

View Donors:
Blood bank can view all the donors available and can send request to particular donor.

View Status:
Blood bank can view the status of their requests sent by the hospital and can accept/reject a request.

View Feedback:
Blood bank can view feedback sent by hospital.

View Chats:
Blood bank can view chats from the hospital and can give reply to the hospital.

3.Hospital:

Login:
Hospital can login with their valid credentials and hospital who need registration can register and login.

View:
Hospital can view all the blood groups available.

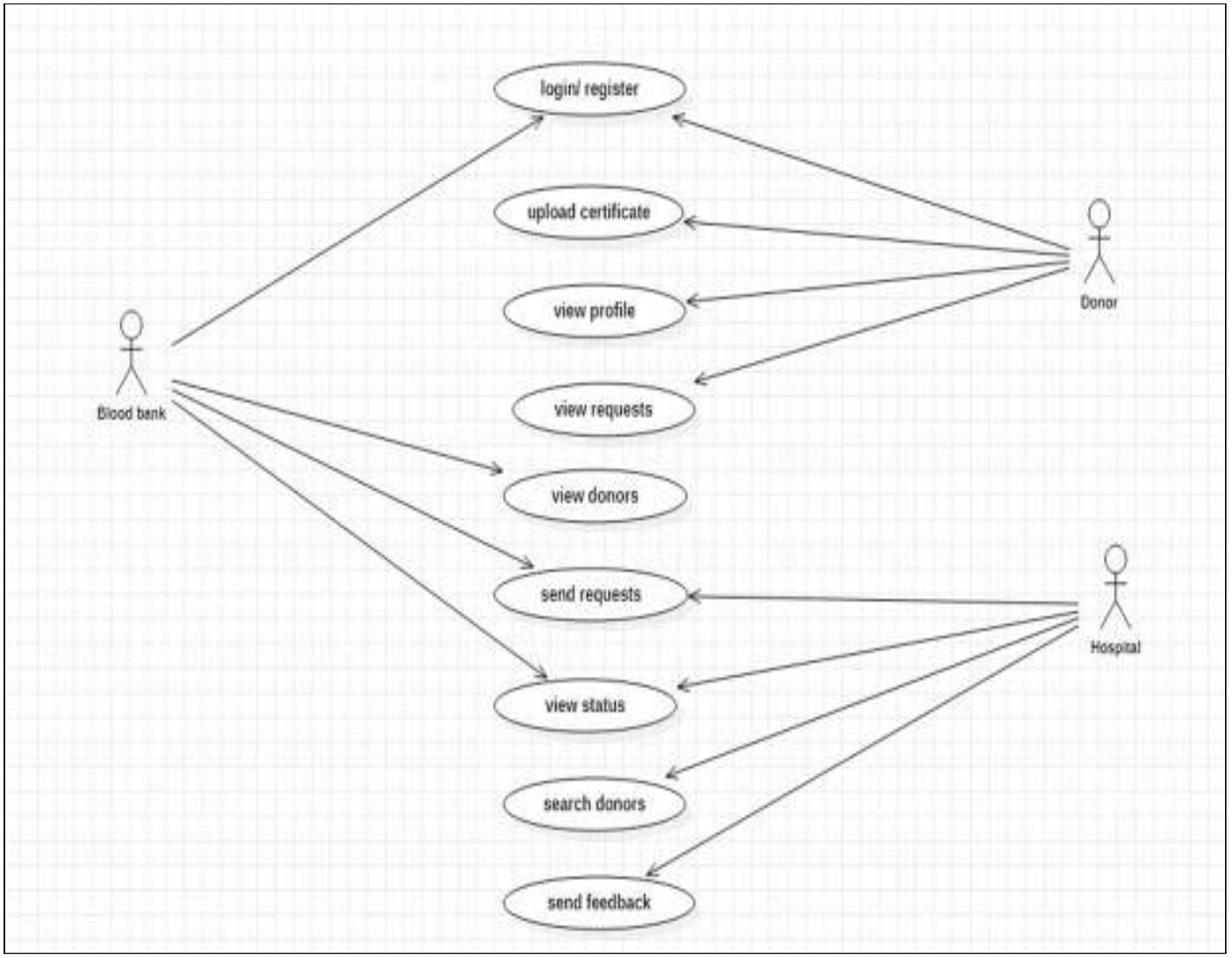
Search;
Hospital can search the blood group and can send request to blood bank.

View Status:
Hospital can view the status of their request.

Send Feedback:
Hospital can send feedback to the blood bank.

3.2 USE CASE DIAGRAM:

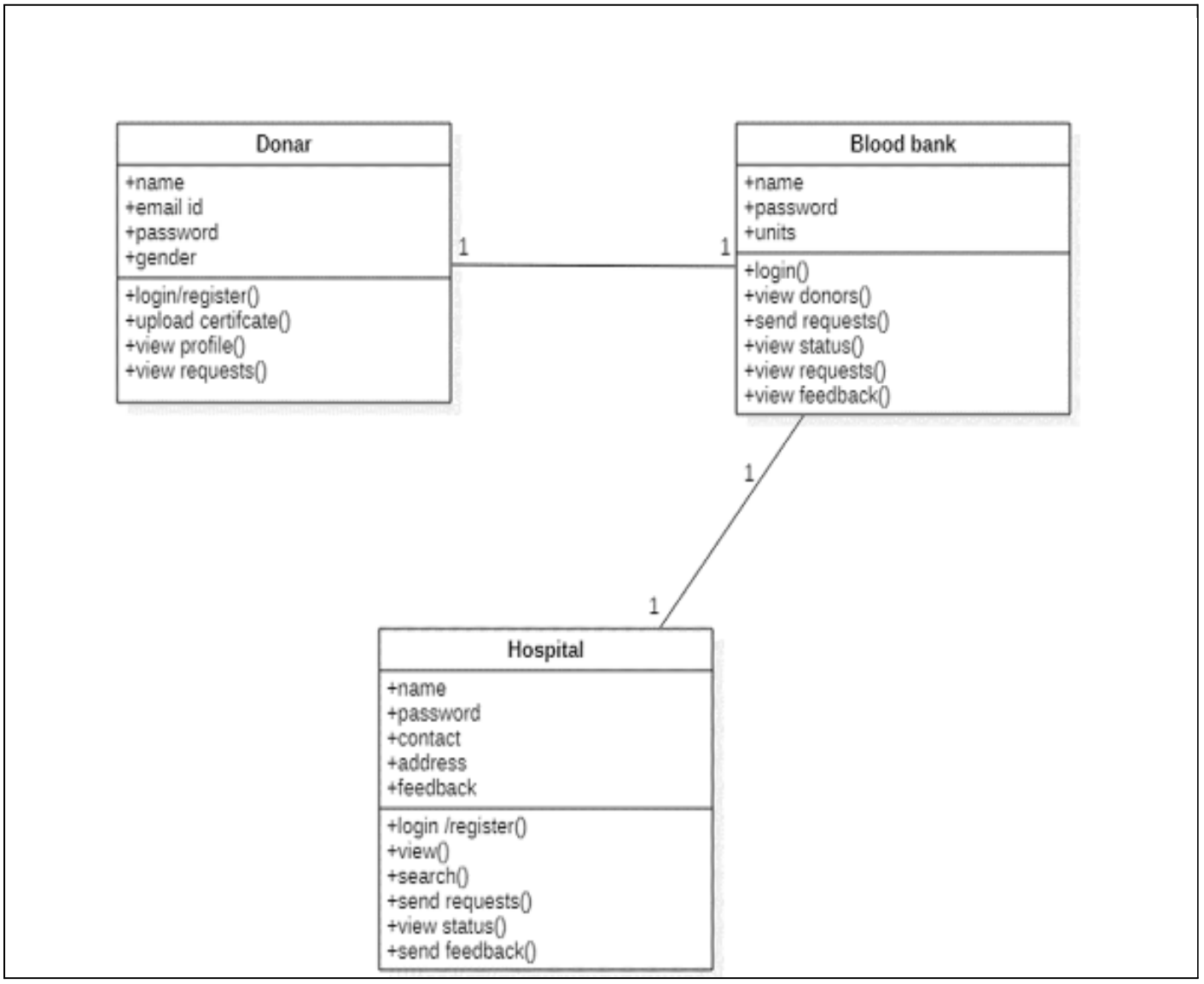
Use Case diagram purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases, It shows what system functions are performed for which actor.



3.2 Use Case Diagram For Blood Bank, Donor and Hospital

3.3 CLASS DIAGRAM:

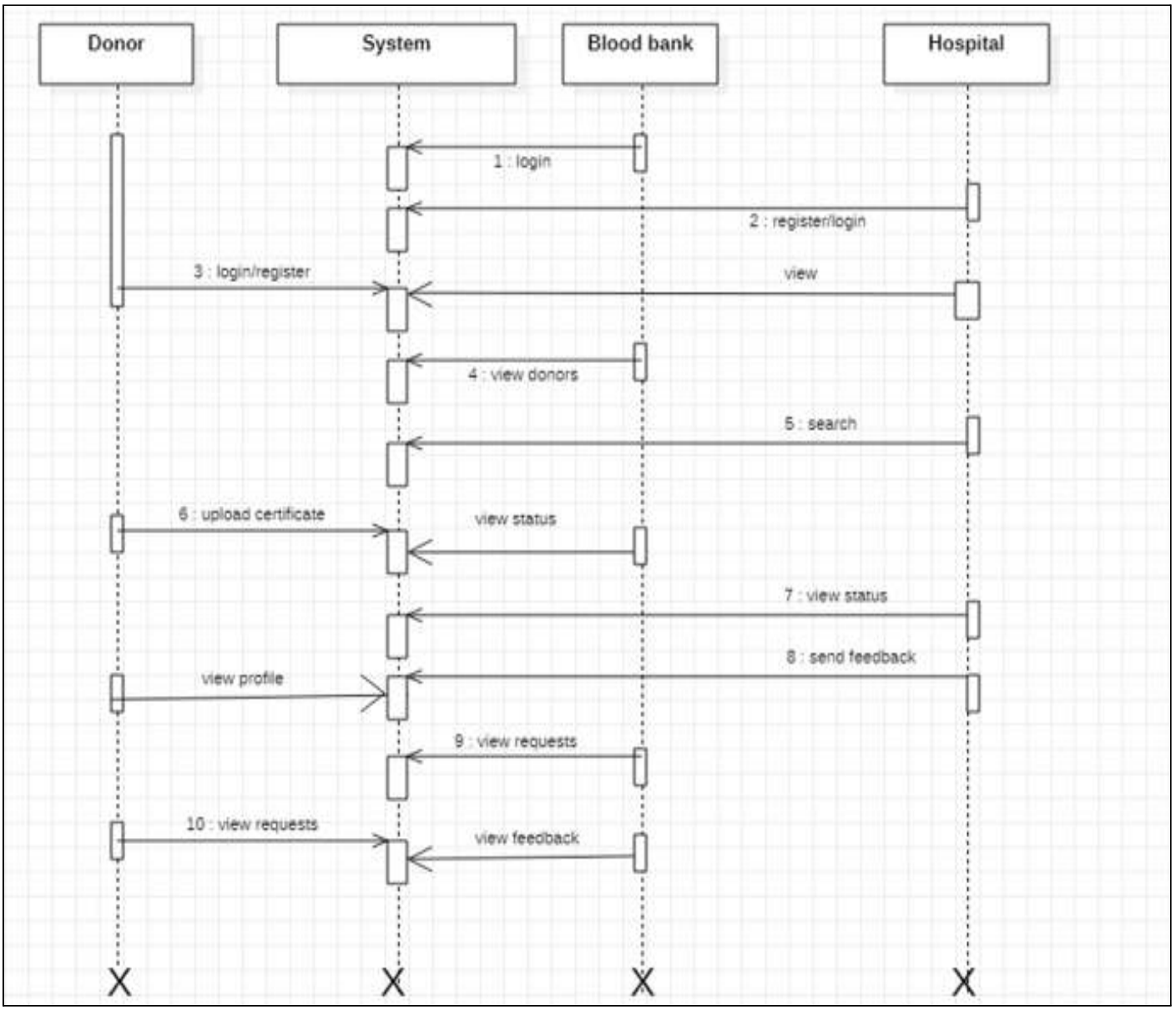
Class Diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations(or methods), and the relationship among the classes.



3.3 Class Diagram For Blood Bank, Donor and Hospital

3.4 SEQUENCE DIAGRAM:

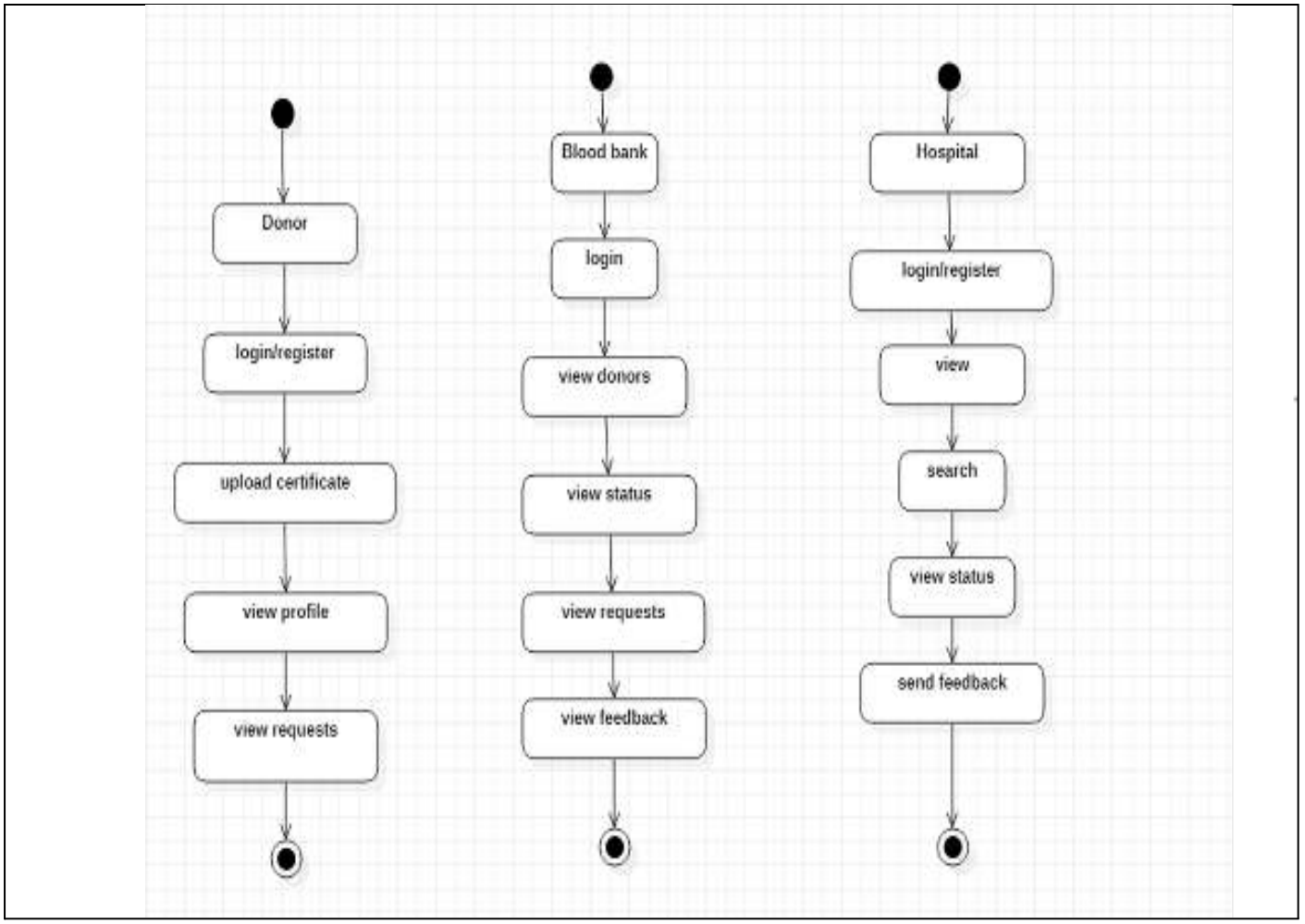
A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams and timing diagrams.



3.4 Sequence Diagram For Blood Bank, Donor and Hospital

3.5 ACTIVITY DIAGRAM :

Activity diagrams are graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



3.5 Activity Diagram For Blood Bank, Donor and Hospital

4.IMPLEMENTATION

4.IMPLEMENTATION

4.1 Sample Code

```
from flask import Flask, make_response, render_template, request, send_file, redirect, url_for
from werkzeug.utils import secure_filename
from db_functions import register_donor, user_loginDb, get_user_profileDb, check_if_email_exists,
set_fileDb, login_bbankDb, bbank_view_donorsDb, update_bbank_donor_status,
get_donor_blood_requestDb, add_bloodDb, h_loginDb, h_registerDb, get_bloodDb, request_blood,
show_blood_reqDb,
get_all_hosp_reqDb,update_hosp_reqDb,hsp_req_feedbackDb,get_hsp_feedDb,get_donor_mail_id
import os
import smtplib

UPLOAD_FOLDER = './cert_uploads/'
app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
smtp_email = 'tejaswinijalli888@gmail.com'
smtp_password = 'wqgofvugcosgghqx'

def send_mail(email_id,userId):
    try:
        subject = "Blood Bank"
        body = "You have a new request from Blood Bank"
        msg = f"Subject: {subject}\n\n{body}\nhttp://127.0.0.1:5000/donor-home?userId={userId}."
        conn = smtplib.SMTP("smtp.gmail.com", 587)
        conn.starttls()
        conn.login(smtp_email, smtp_password)
        conn.sendmail(smtp_email, email_id, msg)
        conn.close()
    except Exception as e:
        print("Mail sending failed",e)

# ################## donor route ##################


def get_cert_loc(userId):
    files_list = os.listdir("./cert_uploads/")
    for x in files_list:
        find = f"{userId}_"
        if x.find(find) == 0:
            return x
    return ''


@app.route("/donorLogin", methods=['GET', 'POST'])
```

```
@app.route("/donor-login", methods=["GET", 'POST'])
@app.route("/donorRegister", methods=["GET", 'POST'])
def user_login_page():
    if request.method == "POST":
        if request.path == "/donorRegister":
            username = request.form.get("username")
            email_id = request.form.get("emailId")
            password = request.form.get('password')
            confirmPassword = request.form.get("confirmPassword")
            gender = request.form.get("gender")
            dob = request.form.get("dob")
            age = request.form.get('age')
            contact = request.form.get('contact')
            address = request.form.get("address")
            bgroup = request.form.get("bgroup")

            if password != confirmPassword:
                return render_template("donor_login.html", message="password does not match")

            if check_if_email_exists(email_id):
                return render_template("donor_login.html", message="email already exists")

            if not (username and email_id and password and gender and dob and age and contact and address and
bgroup):
                return redirect(url_for("user_login_page"))

            register_donor(username, email_id, password, gender,
                           dob, bgroup, age, contact, address)

        elif request.path == '/donor-login':
            emailId = request.form.get("email")
            password = request.form.get("password")
            if not (emailId and password):
                return render_template("donor_login.html", message="invalid user or password")
            user = user_loginDb(emailId, password)
            if user:
                return redirect(url_for('donor_home', userId=user[0][0]))

    return render_template("donor_login.html")


@app.route('/donor-home')
def donor_home():
    args = request.args.to_dict()
    userId = args.get('userId')
    user = get_user_profileDb(userId)
```

```
    status = get_donor_blood_requestDb(userId)
    a = get_cert_loc(userId)
    if user is None:
        return redirect(url_for("user_login_page"))
    user = list(user)
    if user[2]:
        user[2] = request.url_root + "cert-file/" + a
    return render_template('donor_home.html', user=user, cert=a, status=status)


@app.route("/upload-cert", methods=['POST'])
def upload_cert():
    userId = request.args.get('userId')
    if 'file' not in request.files:
        return redirect(f"http://127.0.0.1:5000/donor-home?userId={userId}")
    file = request.files['file']
    if file.filename == '':
        return redirect(f"http://127.0.0.1:5000/donor-home?userId={userId}")
    filename = secure_filename(str(file.filename or ""))
    file.save(os.path.join(
        app.config['UPLOAD_FOLDER'], userId + "_" + filename))
    set_fileDb(filename, userId)
    return redirect(f"http://127.0.0.1:5000/donor-home?userId={userId}")


@app.route("/cert-file/<path:filename>")
def get_cert_file(filename):
    return send_file('./cert_uploads/'+filename, mimetype="image/png")


# ################### donor routes ends ###################


# +++++++++++++++++++ blood bank routes starts +++++++++++++++++++

@app.route("/b-banklogin", methods=['GET', 'POST'])
def b_bank_login():
    if request.method == "POST":
        admin_mail = request.form.get("email")
        password = request.form.get("password")
        if not(admin_mail and password):
            return render_template("b_login.html", message="invalid mail or password")
        user = login_bbankDb(admin_mail.strip(), password.strip())
        if user:
            return redirect(url_for("b_home"))
        return redirect(url_for("b_bank_login", s="true"))
```

```
    s = request.args.get("s")
    message = ''
    if s:
        message = "Invalid username or password"
    return render_template("b_login.html", message=message)


@app.route("/b-home")
def b_home():
    users = bbank_view_donorsDb()
    hosp_req = get_all_hosp_reqDb() # [(2, 'B+', 2, None, 'pending', 1)]
    return render_template("b_home.html", users=users, hosp_req=hosp_req)


@app.route("/req-donor", methods=["POST"])
@app.route("/donor-reply", methods=["POST"])
@app.route("/reply_hsp_req", methods=["POST"])
def req_donor():

    userId = request.args.get("userId")
    status = request.args.get("status")
    update_bbank_donor_status(status, userId)
    if request.path == '/req-donor':
        emailId = get_donor_mail_id(userId)
        if emailId:
            send_mail(emailId, userId)
        return redirect(url_for('b_home'))
    elif request.path == "/reply_hsp_req":
        s = request.args.get('s')
        r_id = request.args.get("r_id")
        quan = request.args.get("quan")
        if s and r_id:
            if s == "approve":
                update_hosp_reqDb(r_id,"completed",int(quan))
            elif s == "reject":
                update_hosp_reqDb(r_id, "rejected")
    if request.path == "/donor-reply":
        return redirect(url_for("donor_home", userId=userId))
    return redirect(url_for("b_home"))


@app.route("/add-blood", methods=["POST"])
def add_blood():
    quan = request.form.get("quantity")
    userId = request.args.get('userId')
    bgroup = request.args.get("bgroup").upper()
```

```
    blood_map = {"1": "A+",  "2": "B+", "3": "O+",  "4": "AB+",
                 "5": "A-",  "6": "B-",  "7": "O-", "8": "AB-"}
    if bgroup.isdigit():
        bgroup = blood_map[bgroup]
    else:
        bgroup = "unknown"

    update_bbank_donor_status("completed", userId)
    add_bloodDb(bgroup, quan)
    return redirect(url_for("b_home"))

@app.route("/b_feedback")
def b_feedback():
    r_id = request.args.get("r_id")
    data = get_hsp_feedDb(r_id)
    return render_template("b_feedback.html", feedback=data)


# ++++++++++++++++++ blood bank routes ends ++++++++++++++++++


# ++++++++++++++++++ hospital routes starts ++++++++++++++++++


@app.route("/h-login", methods=['GET', "POST"])
def h_login():
    if request.method == "POST":
        h_name = request.form.get("h_name")
        pwd = request.form.get("password")
        post_type = request.args.get("type")

        if post_type == "register":
            contact = request.form.get("contact")
            address = request.form.get("address")
            if not (h_name and pwd and contact and address):
                return render_template("h_login.html", message="Fill all the fields")
            h_registerDb(h_name, pwd, contact, address)
            return redirect(url_for("h_login"))
        else:
            if not (h_name and pwd):
                return render_template("h_login.html", message="Invalid hospital name or password")
            h = h_loginDb(h_name, pwd)
            if h:
                return redirect(url_for("h_home", userId=h[0]))
            return render_template("h_login.html", message="Invalid hospital name or password")
```

```
    return render_template("h_login.html", message="")


@app.route("/h-home")
def h_home():
    userId = request.args.get('userId')
    users = (1,)
    all_groups = get_bloodDb(userId)
    blood_req = show_blood_reqDb(userId)
    return render_template("h_home.html", all_groups=all_groups, users=users, h_id=userId,
blood_req=blood_req)


@app.route("/req-quan", methods=['POST'])
def request_quan():
    blood_map = {"1": "A+",  "2": "B+", "3": "O+",  "4": "AB+",
                 "5": "A-",  "6": "B-",  "7": "O-", "8": "AB-"}
    quan = request.form.get("quantity")
    bgroup = request.args.get("bgroup")
    h_id = request.args.get('h_id')
    bgroup = blood_map[bgroup]
    if h_id and bgroup and quan:
        request_blood(quan=quan, bgroup=bgroup, h_id=h_id)
    return redirect(url_for("h_home", userId=h_id))

@app.route("/h-feedback", methods=["GET", "POST"])
def feedback_view():
    r_id = request.args.get("r_id")
    u_id = request.args.get('u_id')
    if not (r_id and u_id):
        resp = make_response('<h1>Page Not Found</h1>')
        resp.status_code  = 404
        return resp
    if request.method == "POST":
        feed = request.form.get("feedback")
        if not feed:
            return redirect(url_for("feedback_view", r_id=r_id, u_id=u_id))
        hsp_req_feedbackDb(feed,r_id=r_id, u_id=u_id)
        return redirect(url_for("h_home",userId=u_id))

    return render_template("h_feedback.html")



# ++++++++++++++++++ hospital routes end ++++++++++++++++++
```

```
if __name__ == "__main__":
    app.run(debug=True)
```

5.RESULTS

5.RESULTS

5.1 SCREENSHOTS

5.1.1:Donor Register Page

If You are the new user then you need to register by giving user details like Full Name, Date of Birth, Email Address, Contact No, Gender, Age, Blood Group, Address, Create Password and Confirm Password.

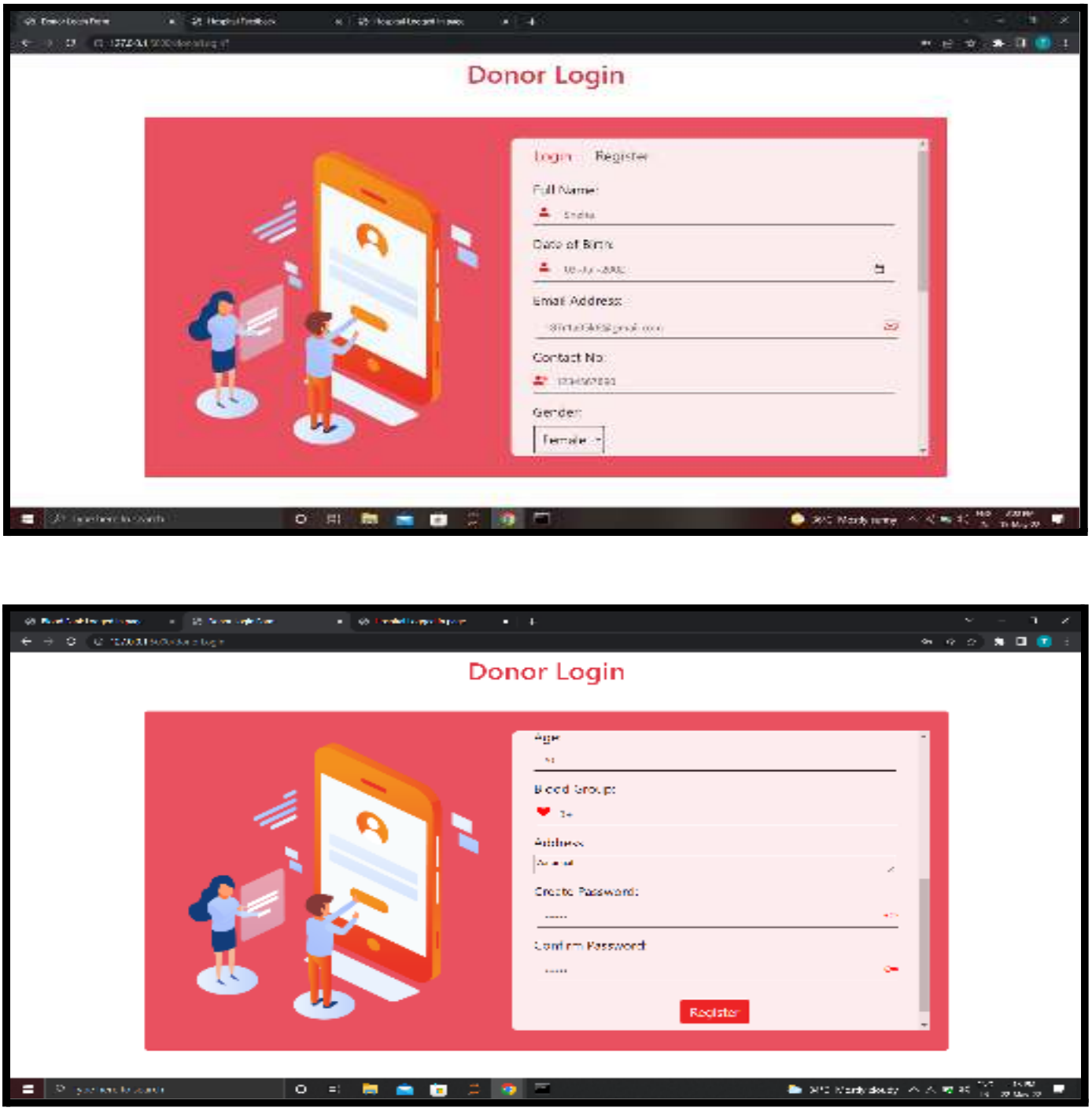


Figure 5.1.1:Donor Register Page

5.1.2:Donor Login Page

If you are the existing user, you can login to the webpage by using Email Address and Password given at the time of register.

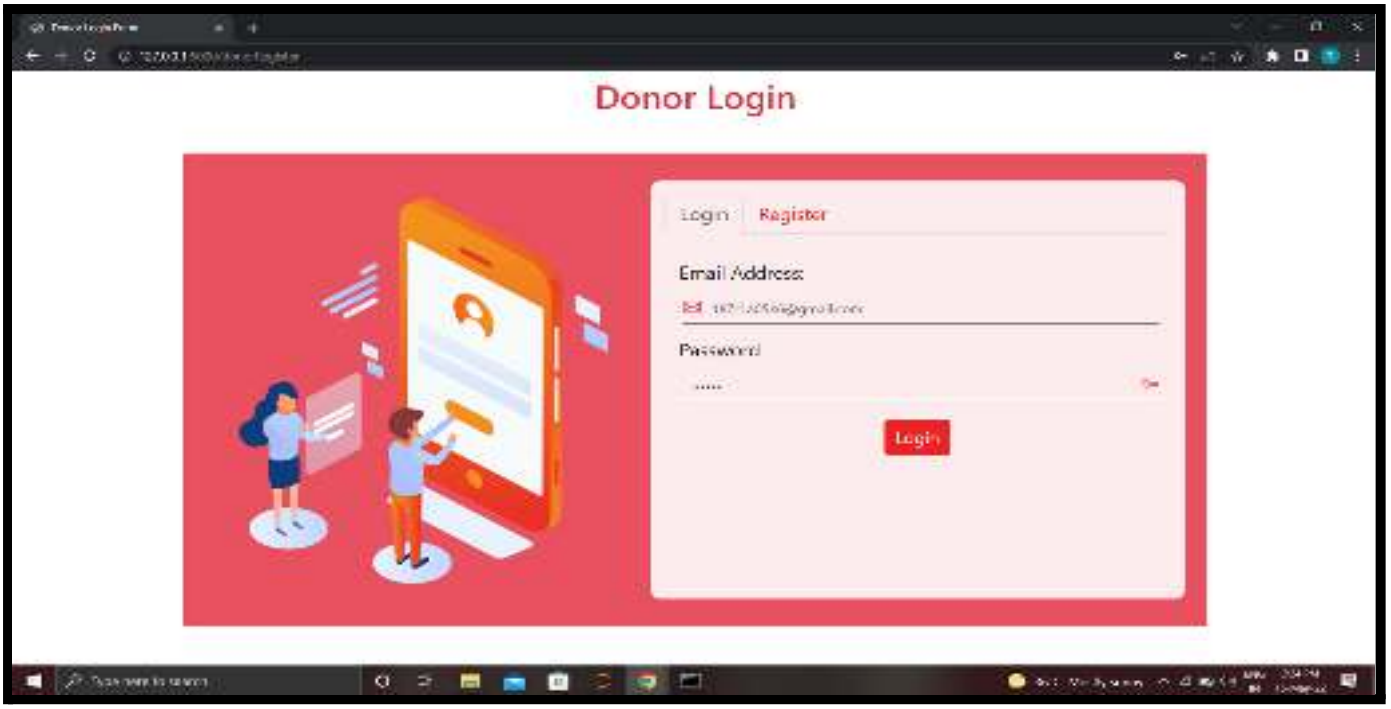


Figure 5.1.2:Donor Login Page

5.1.3:Donor Certificate Page Before Uploading Certificate

Donor has to submit the COVID-19 recovery certificate before donating plasma to the blood bank.

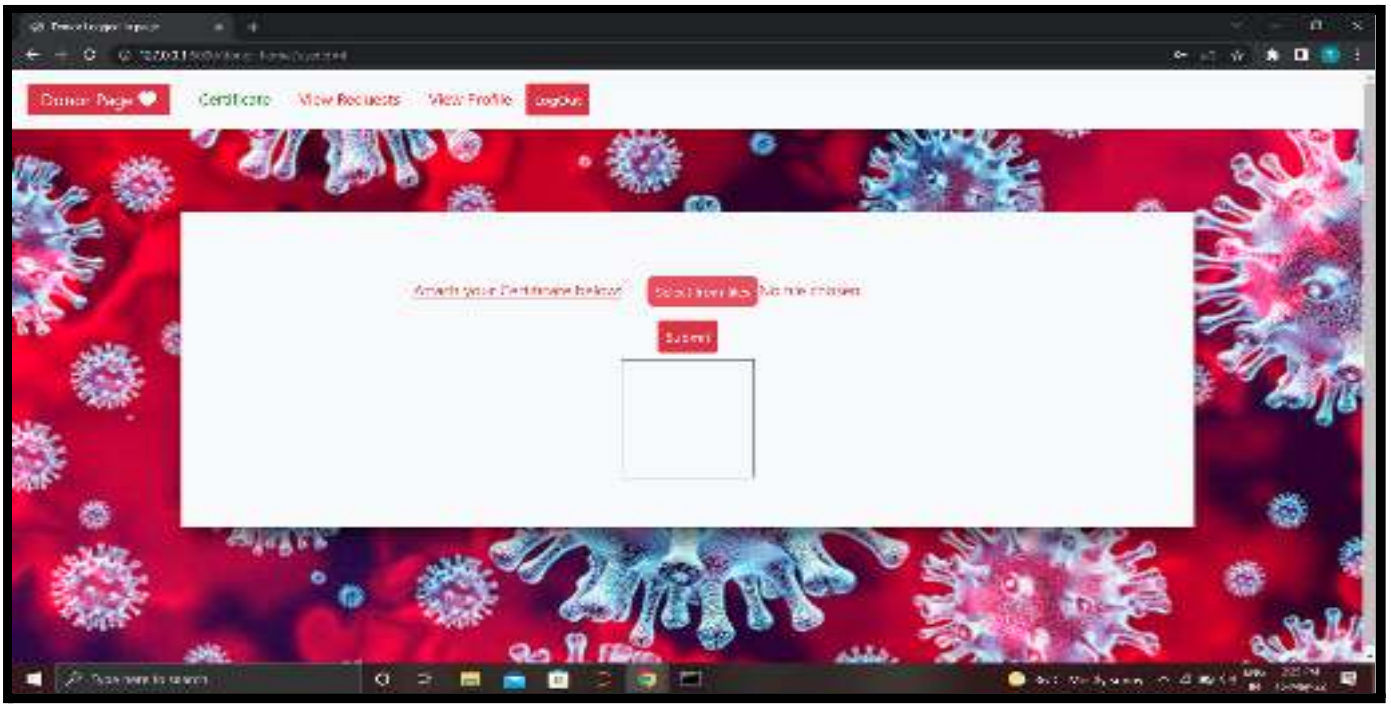


Figure 5.1.3:Donor Certificate Page Before Uploading Certificate

5.1.4:Donor Requests Page Before Uploading Certificate

Donor can view requests from the blood bank.In the below figure there are no requests because the user has not uploaded their recovery certificate.

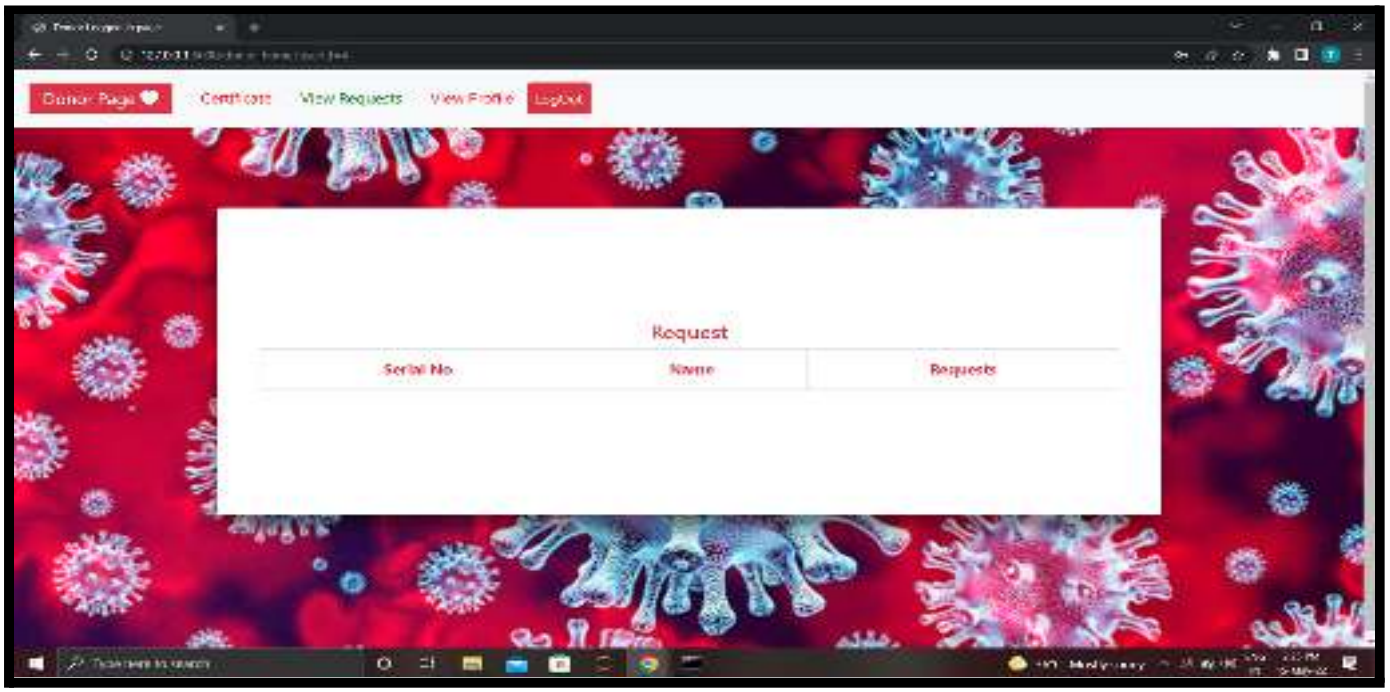


Figure 5.1.4:Donor Requests Page Before Uploading Certificate

5.1.5:Donor Profile Page Before Uploading Certificate

Donor can view their details like Serial Number of Donor, Name, Certificate, Blood Group, Age.

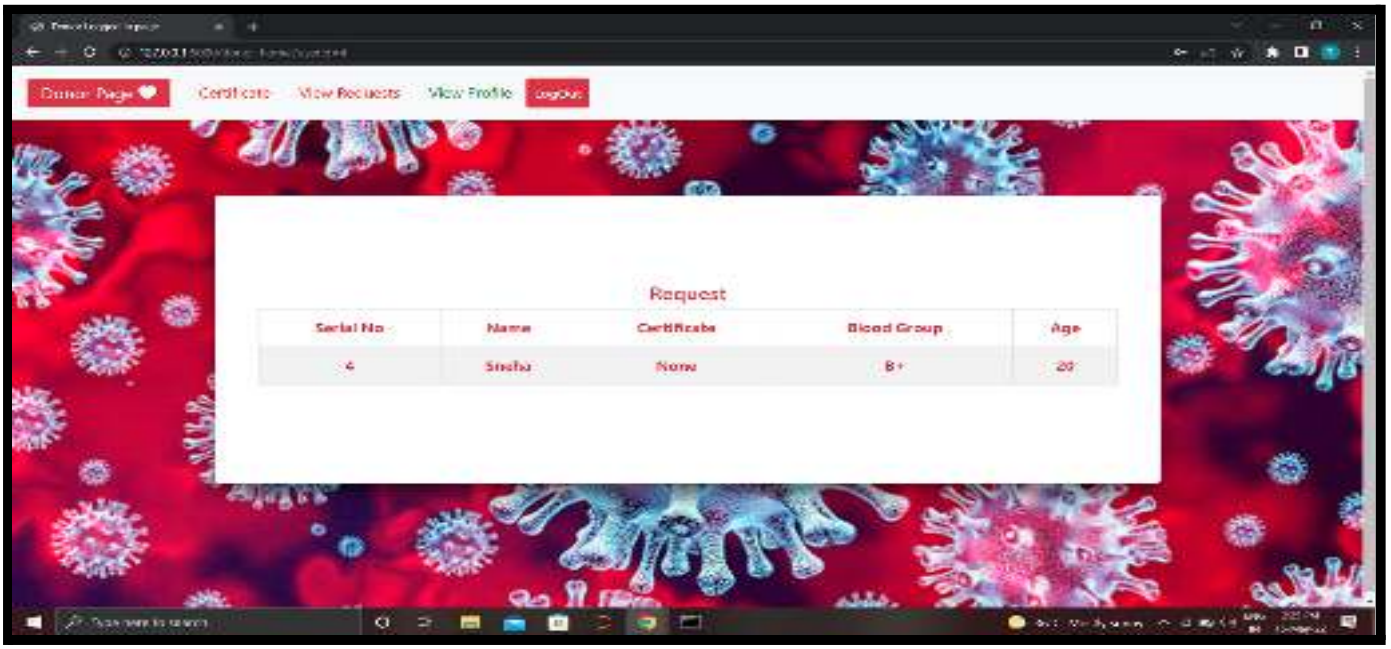


Figure 5.1.5: Donor Profile Page Before Uploading Certificate

5.1.6:Donor Certificate Page After Uploading Certificate

As shown in figure 5.3 user can choose select from files and they can upload COVID-19 recovered certificate by clicking on submit button. Once the file is submitted it will display the message as shown in below figure:

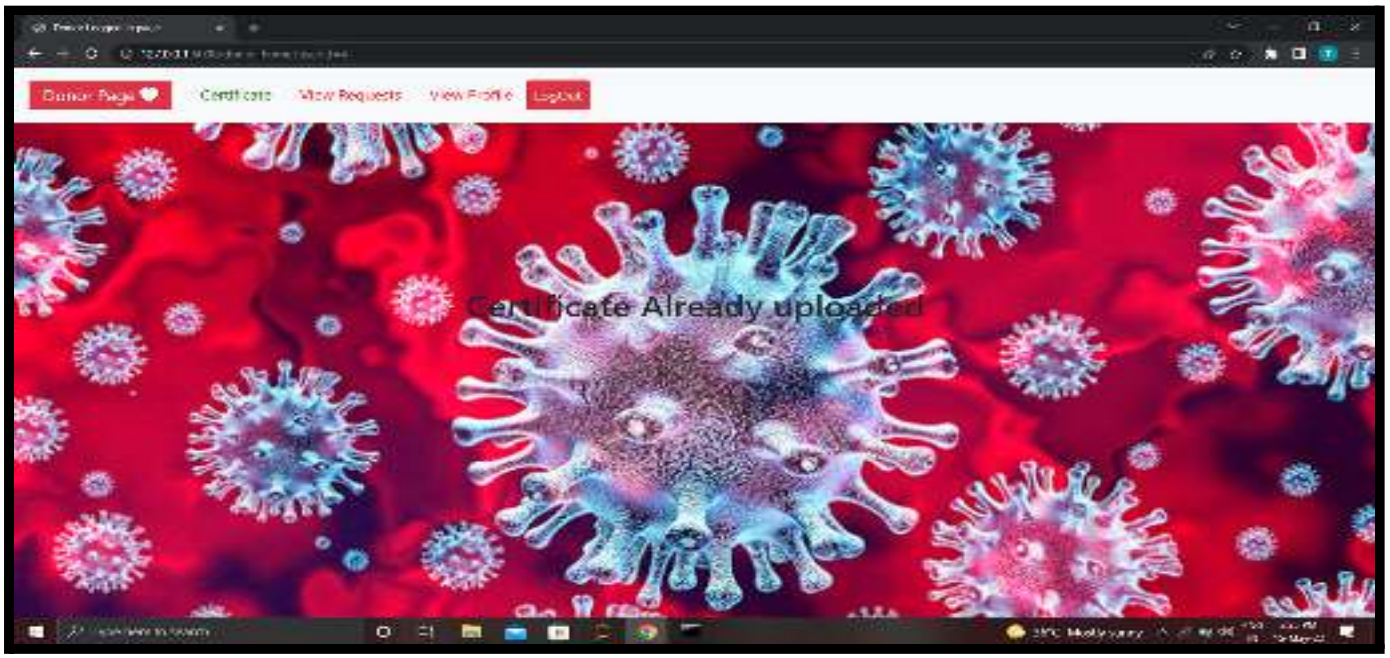


Figure 5.1.6:Donor Certificate Page After Uploading Certificate

5.1.7:Donor Profile Page After Uploading Certificate

User can view the certificate they uploaded by clicking on Download cert from view profile page.

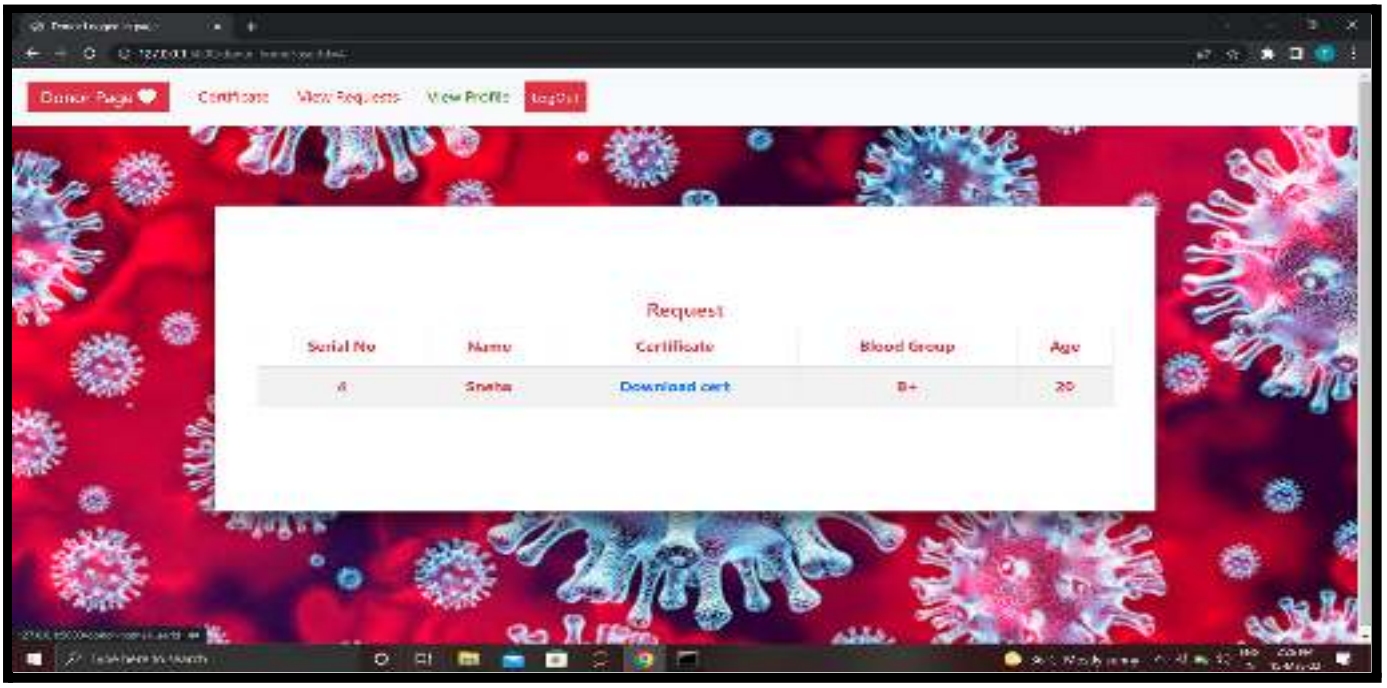


Figure 5.1.7:Donor Profile Page After Uploading Certificate

5.1.8:Blood Bank Login Page

There is only one admin for the blood bank they can login with their Email Address and Password.

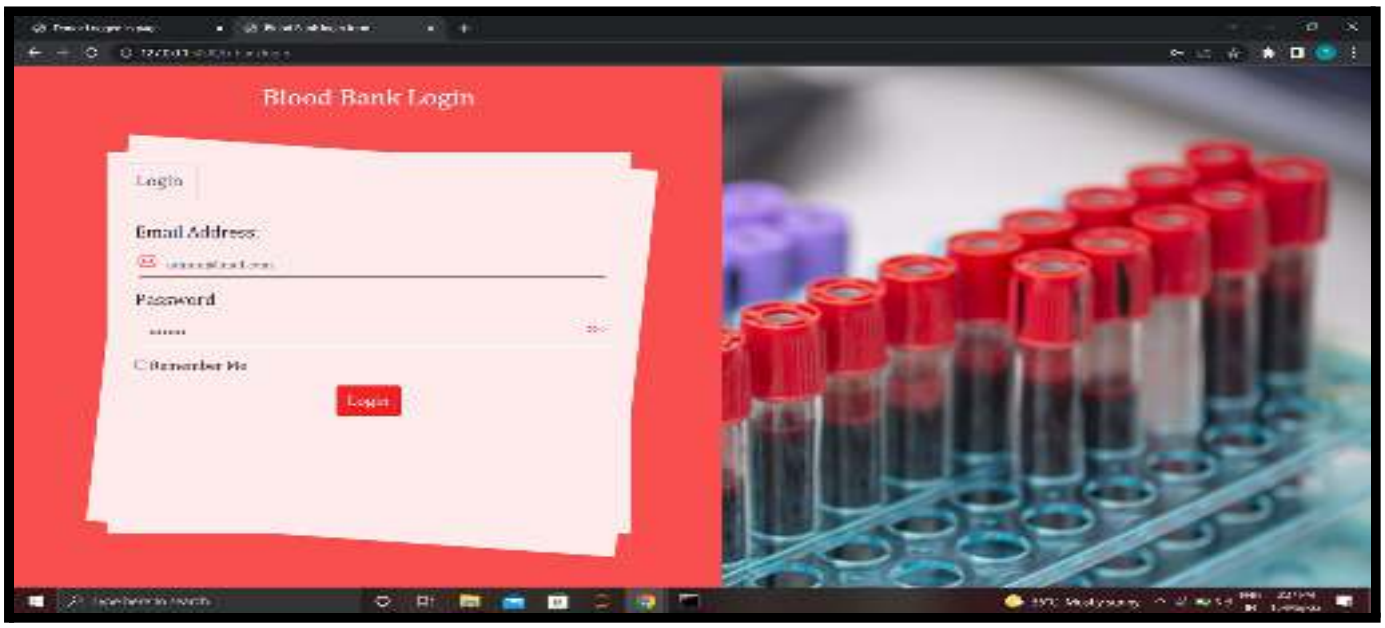


Figure 5.1.8:Blood Bank Login Page

5.1.9:Blood Bank View Donors Page

Blood Bank can view the donors who uploaded the certificate in the donors page and also blood bank can search for the particular blood group that they require and raise a request to the donors by clicking on request in the column Request Blood.

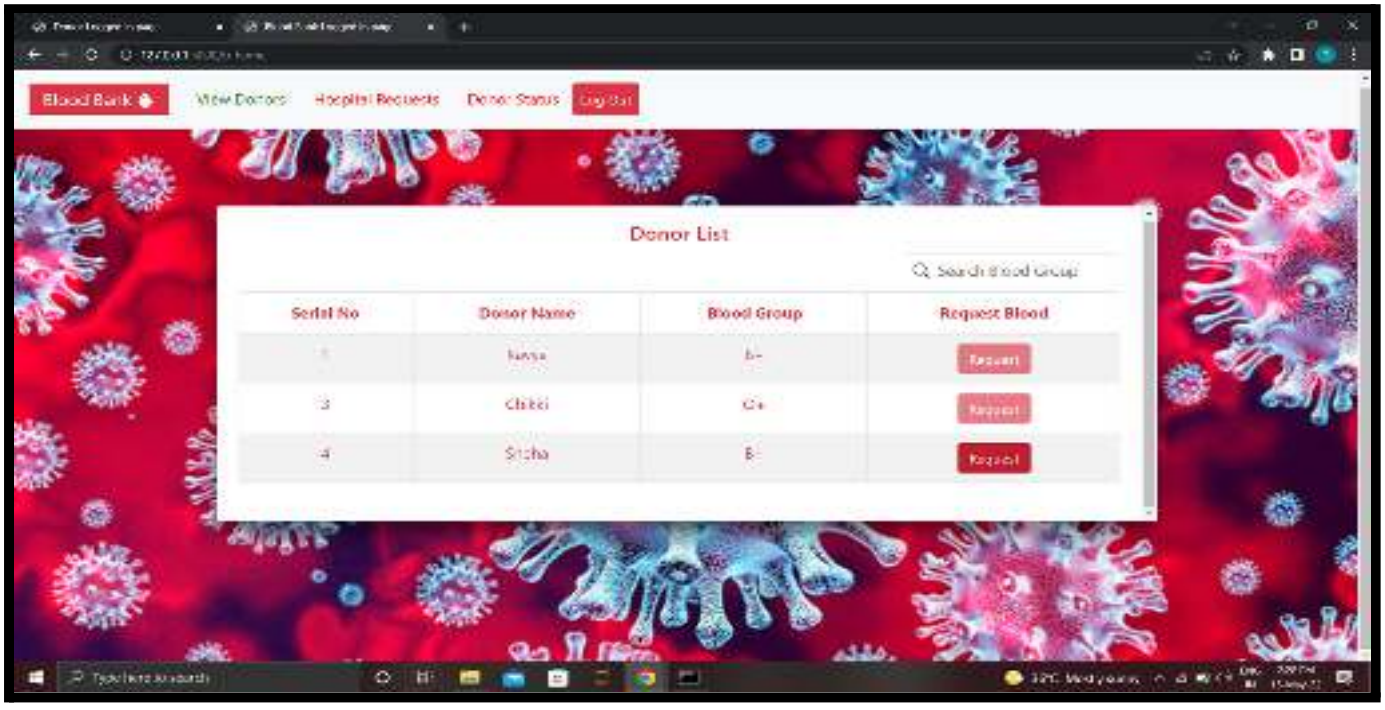


Figure 5.1.9:Blood Bank View Donors Page

5.1.10:Blood Bank Alert Message to Donor

Once the Blood Bank raises a request to the donor an alert message is sent to the mail of the donor.

Figure 5.1.10:Blood Bank Alert Message to Donor

5.1.11:Donor View Requests Page

Through the mail the donor will get to know that they have a request from blood bank and they can login to the donor page and they will find a request from blood bank they had a choice of accepting or rejecting the request as shown below.

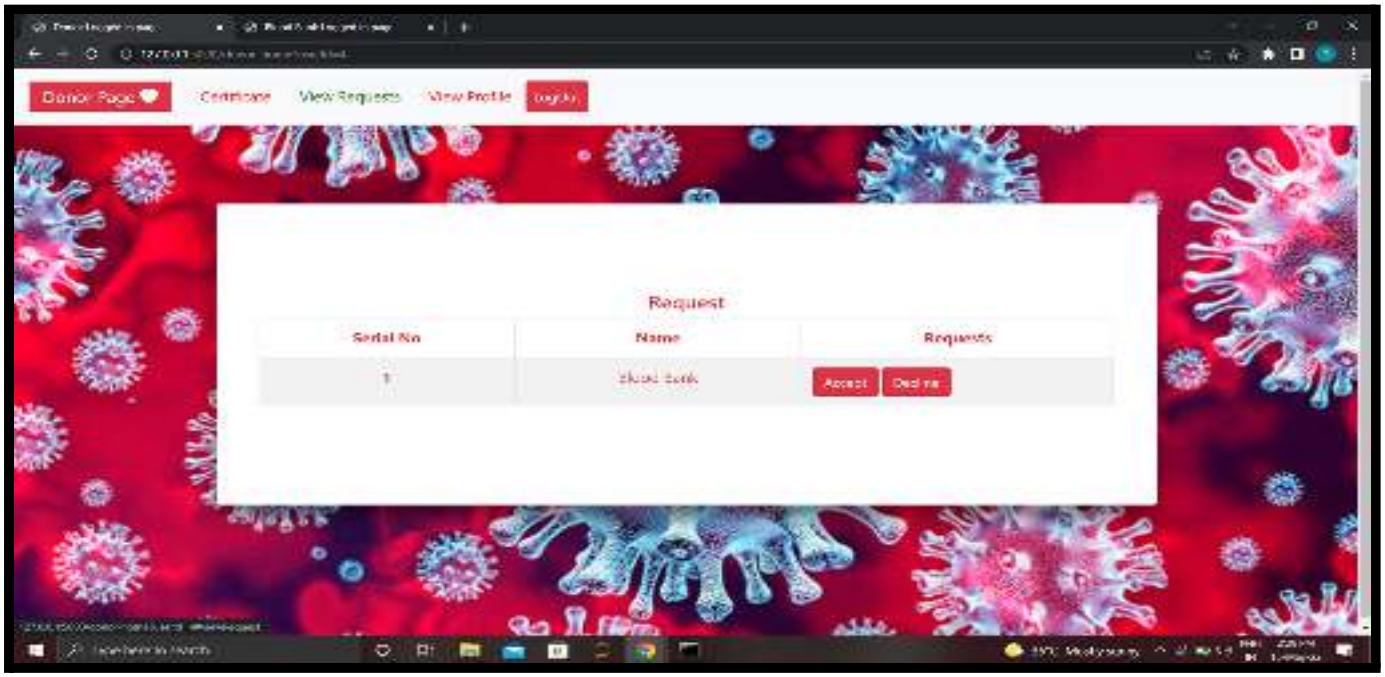


Figure 5.1.11:Donor View Requests Page

5.1.12:Donor View Donor Status Page

If the donor has not given any response then it will show the status as pending, if the donor has accepted the request blood bank can add the units of blood and approve it and whenever hospital makes a request and they collected plasma from blood bank it will show status as completed.

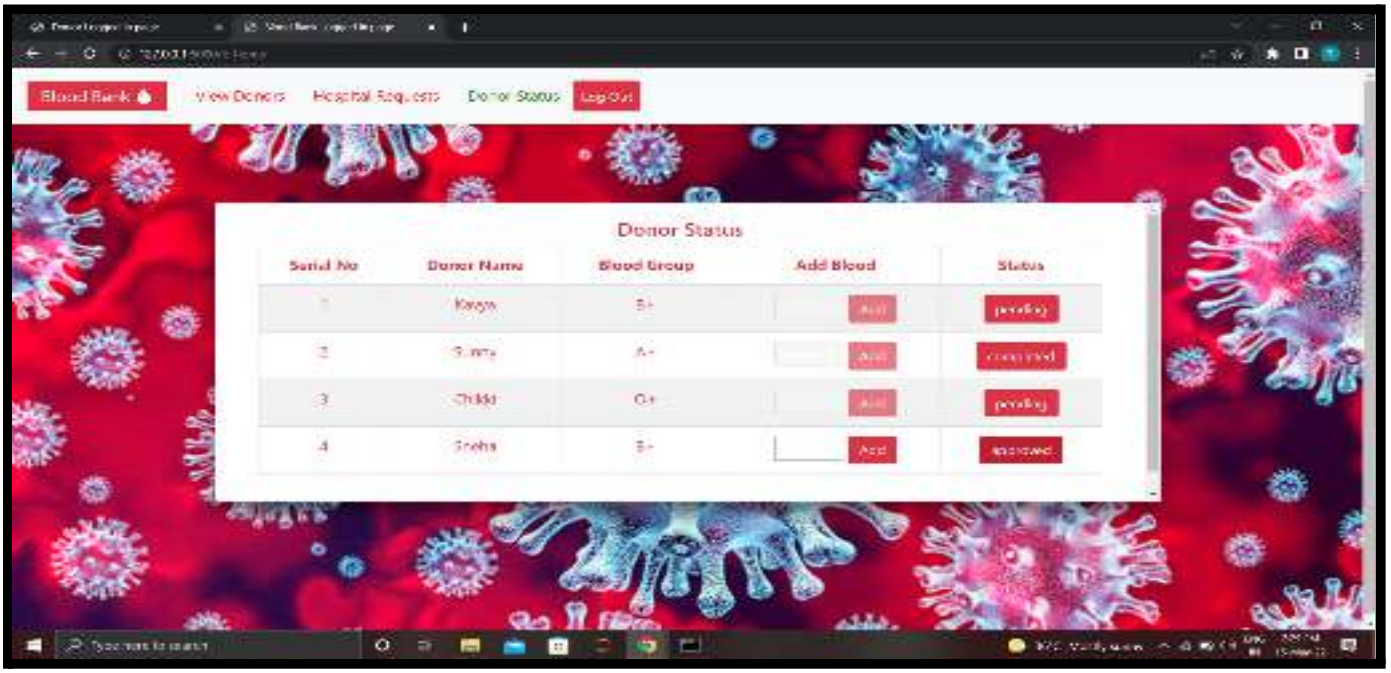


Figure 5.1.12:Donor View Donor Status Page

5.1.13:Hospital Register Page

Different hospitals can register and raise a request to the blood bank.They can give the hospital name, Contact, Address and create password.

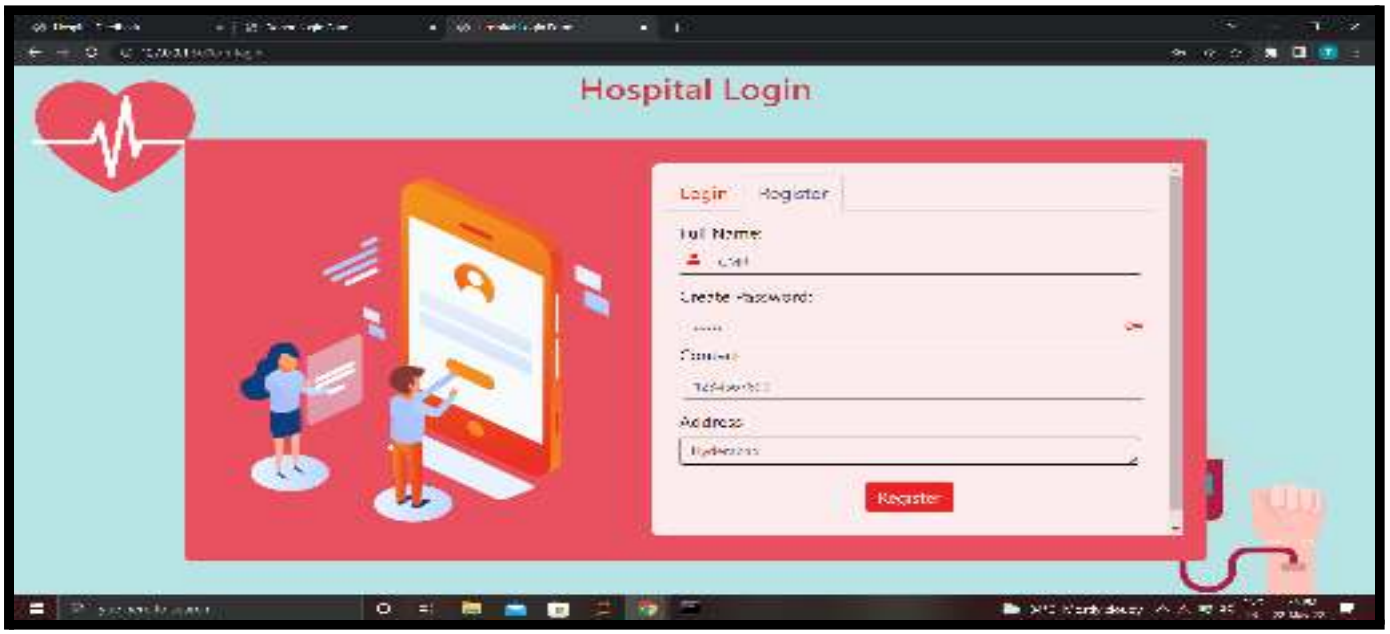


Figure 5.1.13:Hospital Register Page

5.1.14:Hospital Login Page

Hospitals can login with the name and password.

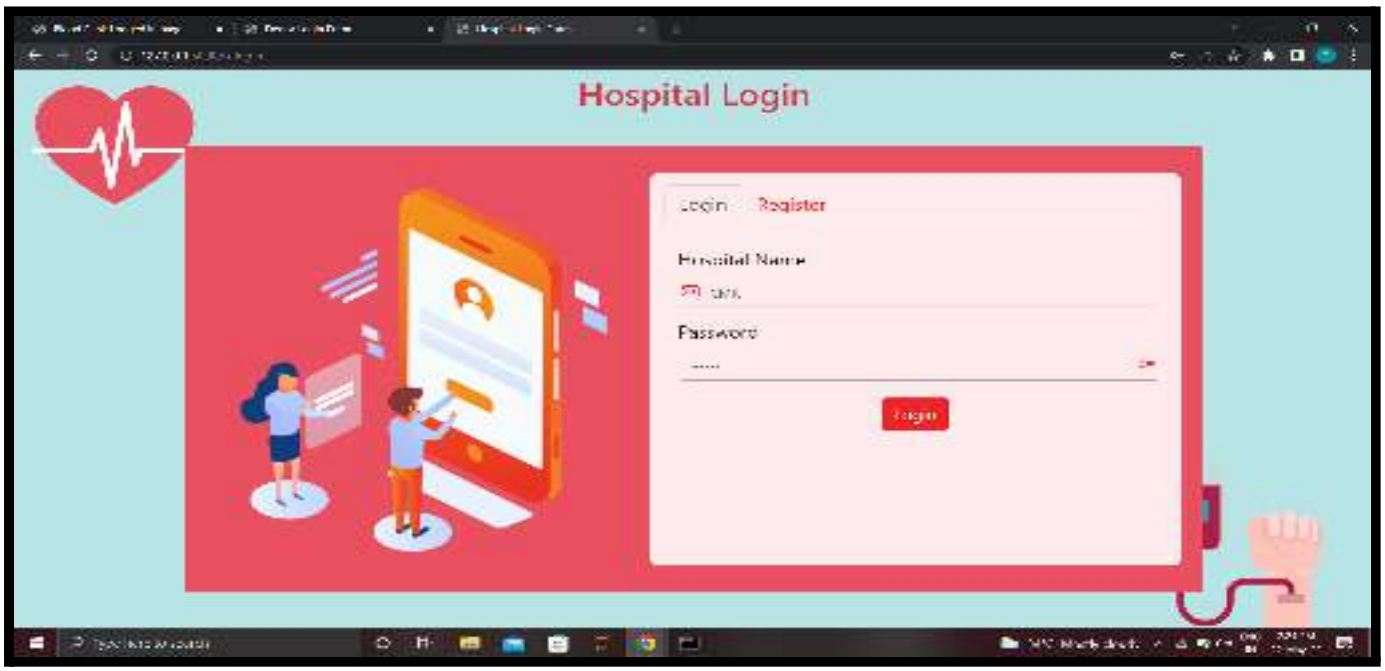


Figure 5.1.14:Hospital Login Page

5.1.15:Hospital View Page

Here the Hospital can view the available blood group from the blood bank and they select the number of units they require and can raise a request to the blood bank.

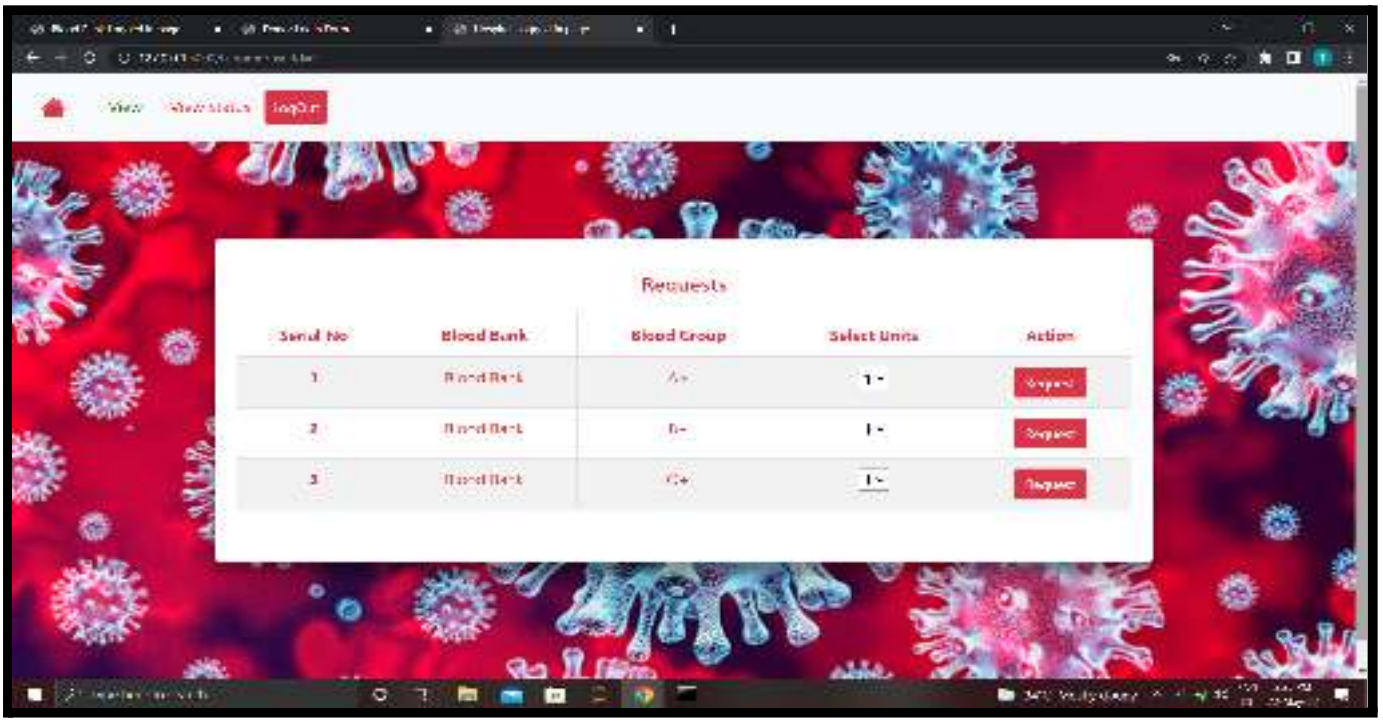


Figure 5.1.15:Hospital View Page

5.1.16:Hospital View Status Page Before Blood Bank Approves

Once a Hospital makes a request to the blood bank they can check their status whether the blood bank has accepted or rejected. If it has not responded it will show the status as pending.

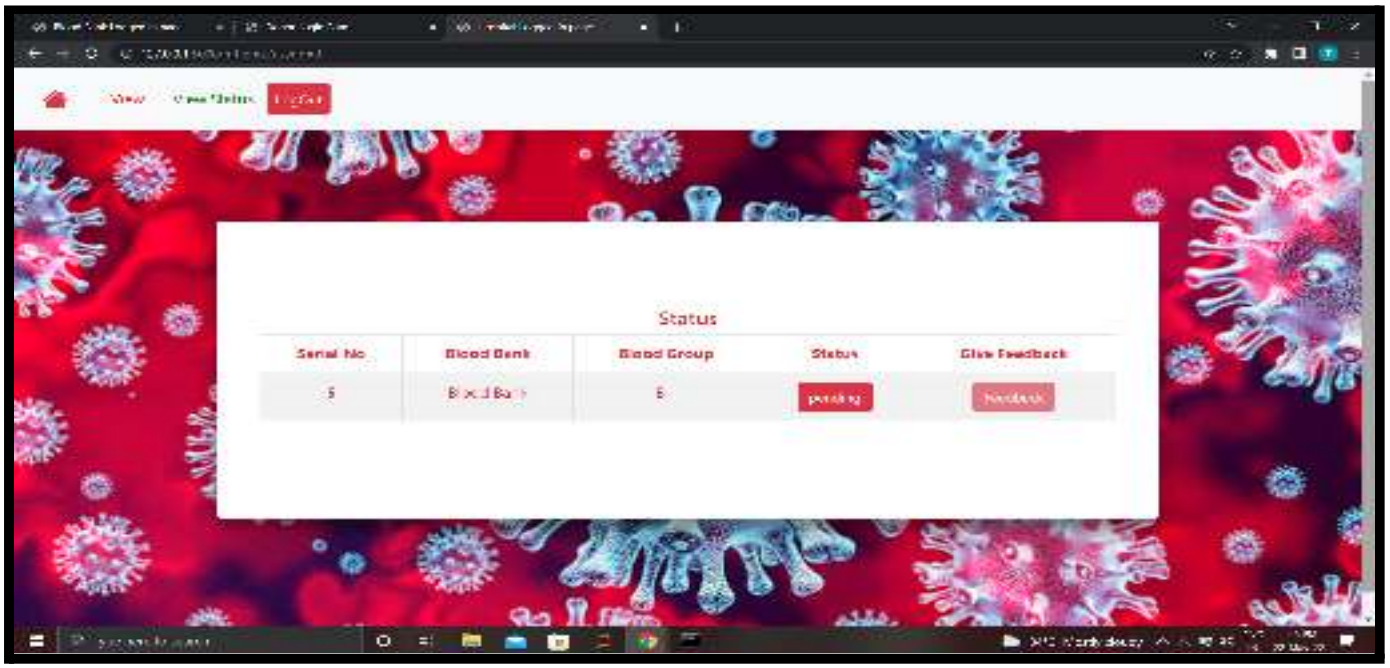


Figure 5.1.16:Hospital View Status Page Before Blood Bank Approves

5.1.17:Blood Bank Hospital Requests Page

Whenever,Hospital makes a request to the Blood Bank they can check the requests from the Hospital Requests and Blood Bank will either approve or reject it.

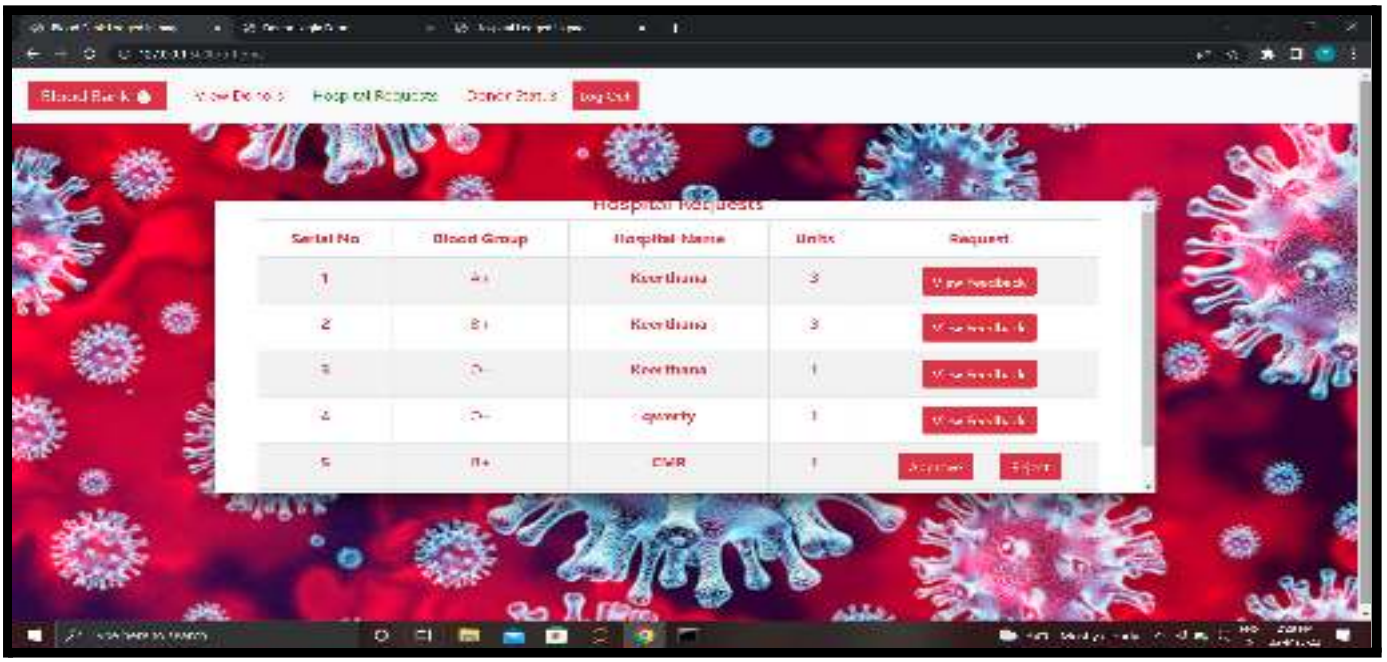


Figure 5.1.17:Blood Bank Hospital Requests Page

5.1.18:Hospital View Status Page After Blood Bank Approves

If the blood bank approves the request it will show the status as completed and the hospital can give the feedback by clicking on the feedback.

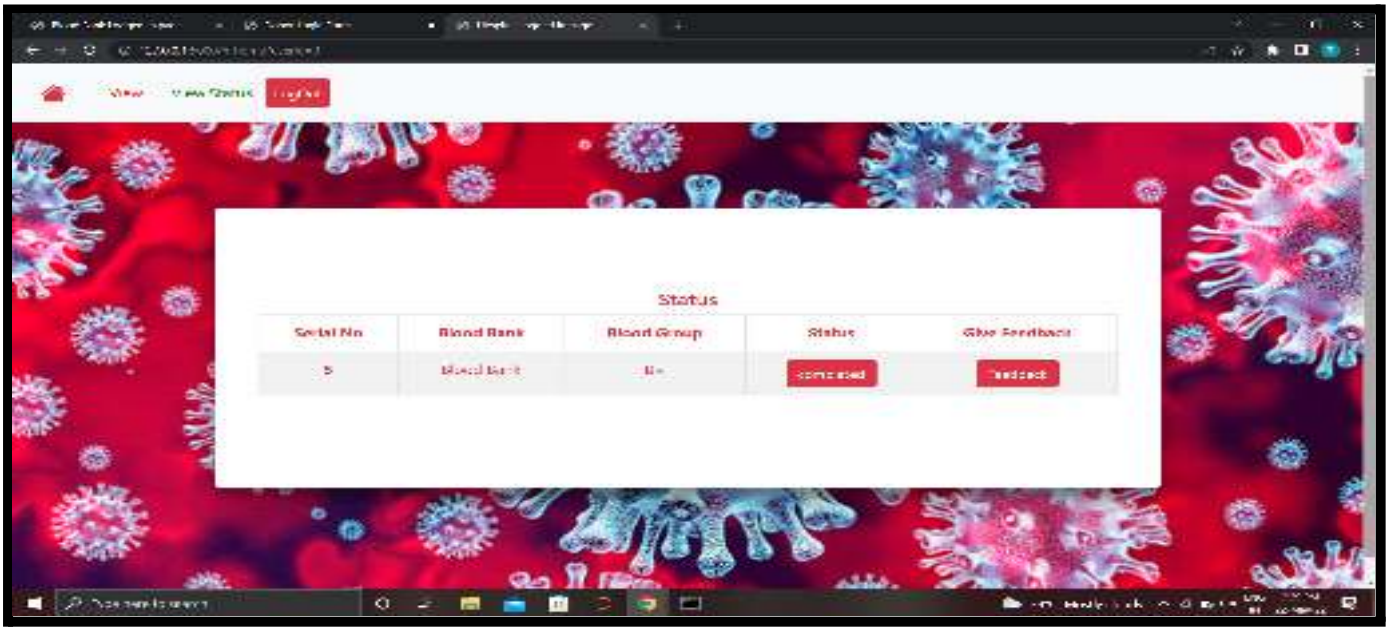


Figure 5.1.18:Hospital View Status Page After Blood Bank Approves

5.1.19:Hospital Feedback Page

Hospital can share their experience through feedback. Feedback can be given in the form of chats.

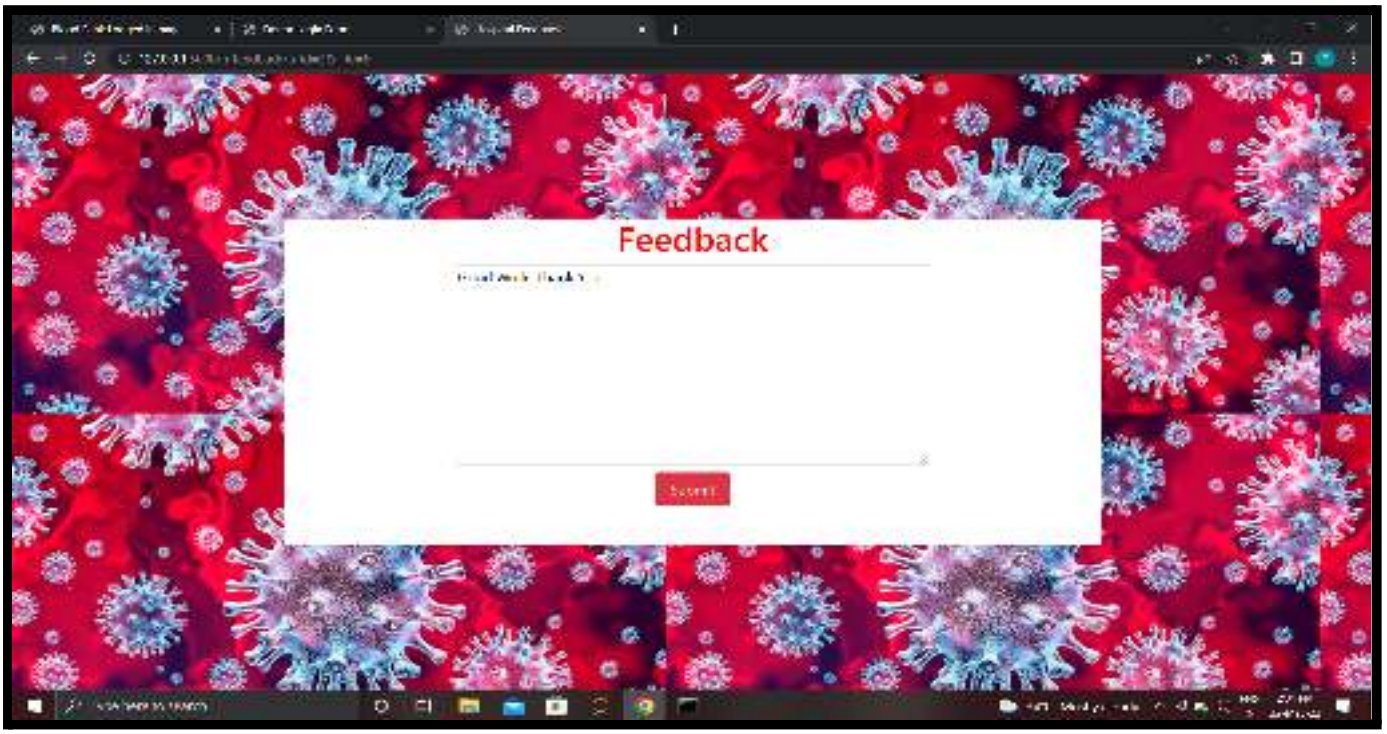


Figure 5.1.19:Hospital Feedback Page

5.1.20:Blood Bank View Feedback Page

Blood Bank can view feedback from the hospital requests page once they give the approval to the hospital's request.

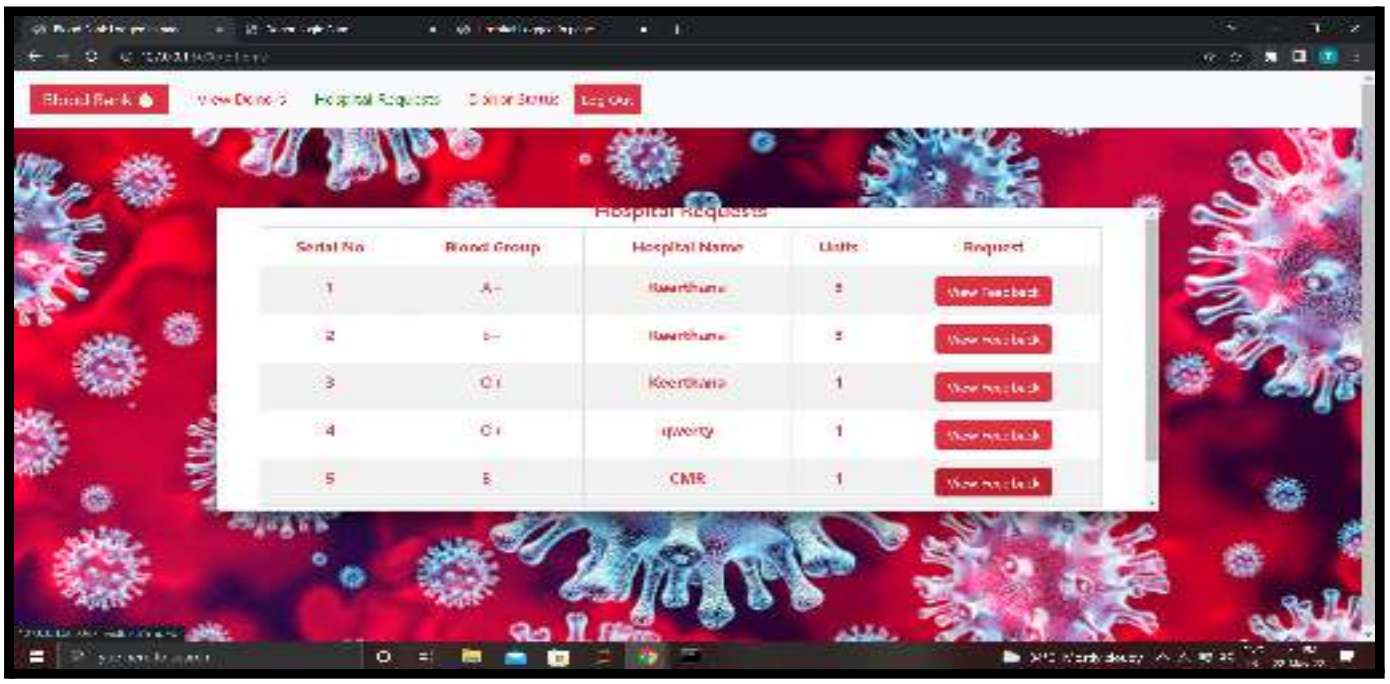


Figure 5.1.20:Blood Bank View Feedback Page

5.1.21:Blood Bank Can View Feedback

After clicking on view feedback as shown in figure 5.20 Blood Bank can view the feedback given by the hospital.

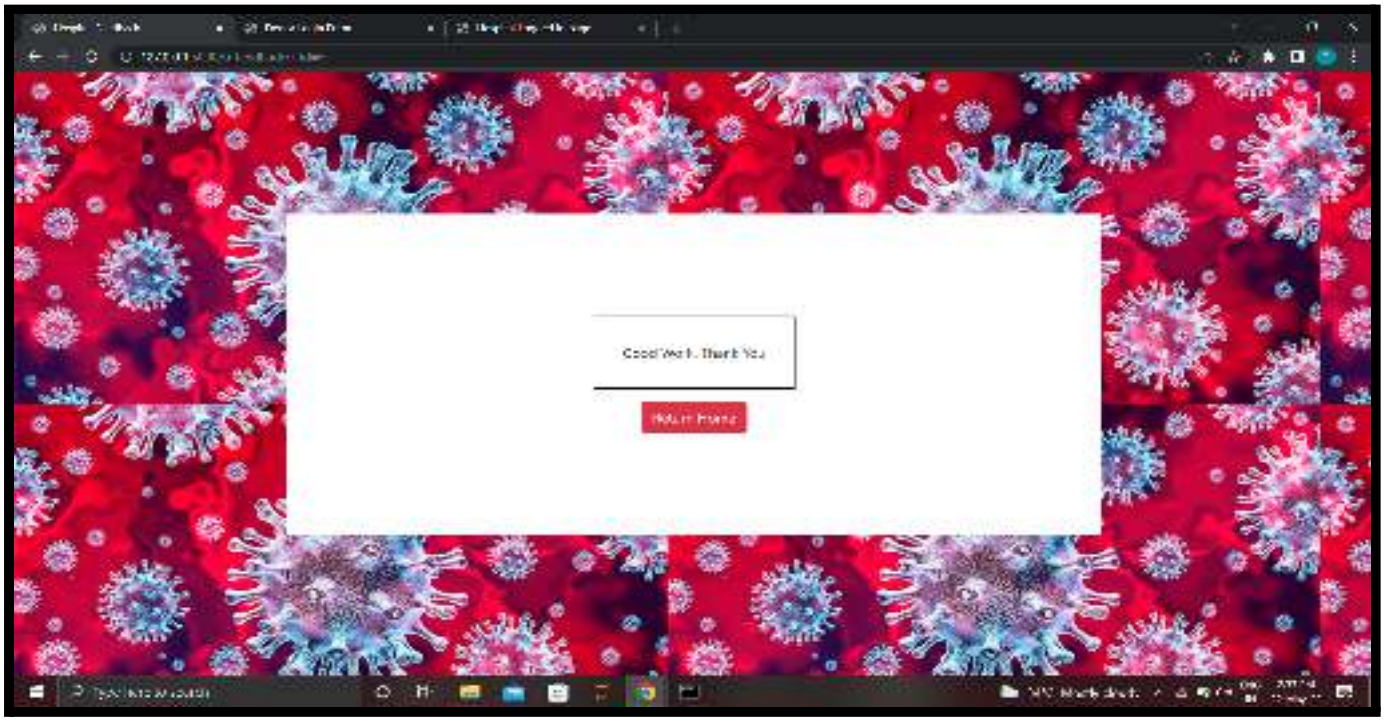


Figure 5.1.21:Blood Bank Can View Feedback

5.1.22:Hospital Update Feedback page

Hospital can update the feedback which they sent to the blood bank.Through this feedback they can also send what blood groups they require more.

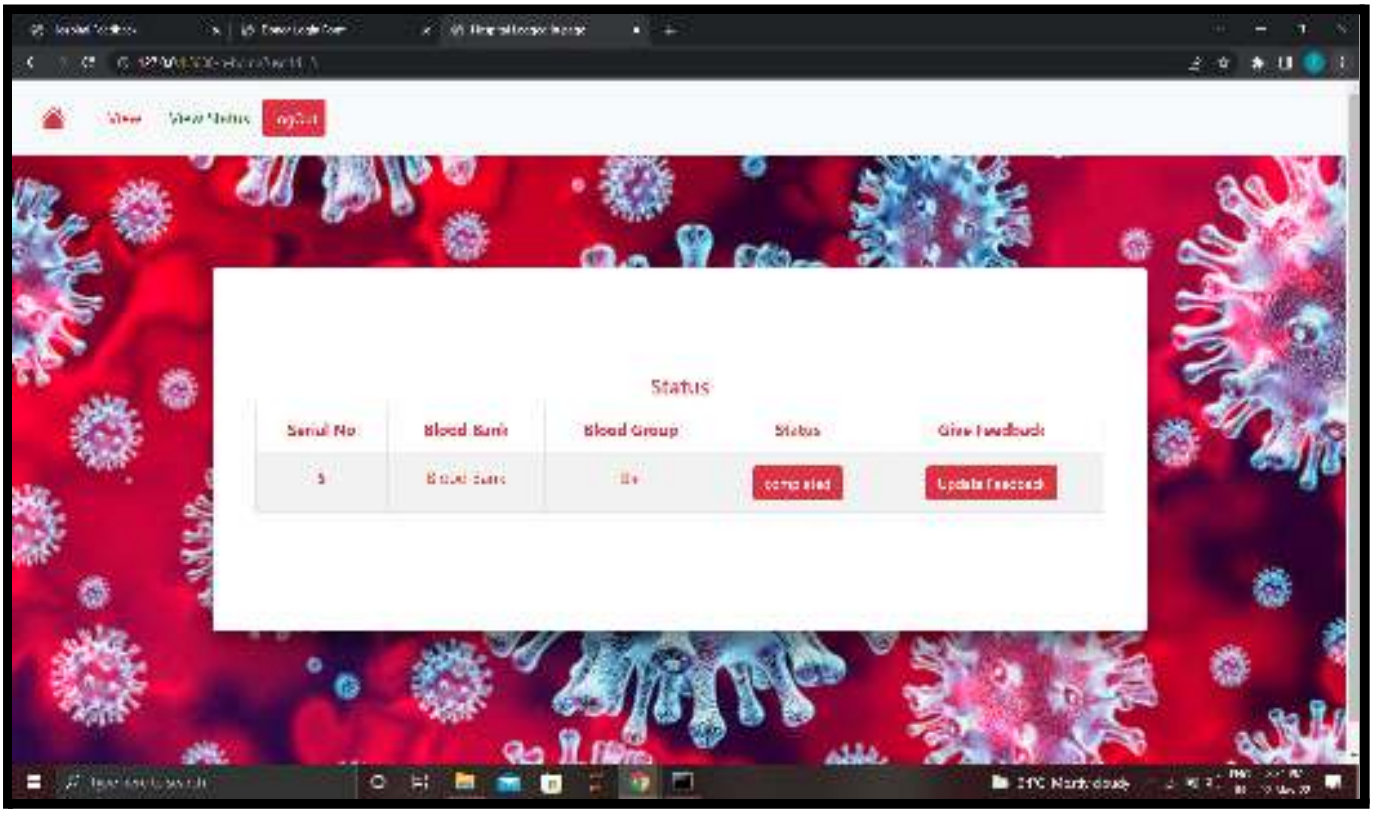


Figure 5.1.22:Hospital Update Feedback page

6.TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

- Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
- Functional testing is centered on the following items:

  1.Valid Input : identified classes of valid input must be accepted.

  2.Invalid Input : identified classes of invalid input must be rejected.

  3.Functions : identified functions must be exercised.

  4.Output :identified classes of application outputs must be exercised.

  5.Systems/Procedures : interfacing systems or procedures must be invoked.
- Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

6.3 TEST CASES

Test	Description	Action	Expected Result	Actual Result	Result
Register	Donor,Hospital and blood bank admin have to register before in order to perform required operations.	Check whether the given details are sufficient.	Valid:Login Page Invalid:Error message	Valid:Login Page Invalid:Error message	Pass
Login	User has to login using credentials given at the time of registration	Check if Email and Password are valid	Valid:Home page Invalid:Error Message	Valid:Home page Invalid:Error Message	Pass
Upload Certificate	Donor has to upload the COVID-19 recovery certificate	Show whether the certificate is uploaded or not	It displays the message certificate is uploaded if user submitted the file.If user has not submitted any file it shows the option select the file and submit	It displays the message certificate is uploaded if user submitted the file.If user has not submitted any file it shows the option select the file and submit	Pass
View Requests	Blood Bank and Donors can view requests.	It shows the requests that they are sending and getting from a particular webpage.	Donor:Once the donor uploads a certificate they can view the requests from the blood bank. Blood Bank:blood can make requests to donor and they can view requests from Hospitals.	Donor:Once the donor uploads a certificate they can view the requests from the blood bank. Blood Bank:blood can make requests to donor and they can view requests from Hospitals..	Pass
View Status	Blood Bank and Hospitals can view there status regarding their requests.	It shows the status of there requests whether it is approved or rejected.	Blood Bank:After sending requests to the donor, blood bank can check there status whether it is approved or rejected. Hospital:After sending requests to the blood bank, Hospital can check there status whether it is approved or rejected.	Blood Bank:After sending requests to the donor, blood bank can check there status whether it is approved or rejected. Hospital:After sending requests to the blood bank, Hospital can check there status whether it is approved or rejected.	Pass

7.CONCLUSION

7.CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

Plasma is the yellow liquid part of the blood that contains antibodies, Antibodies are proteins made by the body in response to infection. People who have fully recovered from COVID-19 for atleast two weeks are encouraged to consider donating plasma, Which may help to save the lives of other patients. Because you fought the infection, your plasma now contains COVID-19 antibodies. These antibodies provided one way for your immune system to fight the virus when you were sick, so your plasma may be able to be used to help others fight off the disease. Individuals must have a prior diagnosis of COVID-19 documented by a laboratory test and meet other donor qualifications. Individuals must have complete resolution of symptoms for atleast 14 days prior to donation. A negative lab test for active COVID-19 disease is not necessary to qualify for donation.

7.2 FUTURE SCOPE

In future this type of process is very helpful and useful to patients who need emergency plasma. At present the world is suffering from COVID-19 crisis, and we haven’t found any vaccine yet. So,this type of process is useful in future also.

8.BIBLIOGRAPHY

8.BIBLIOGRAPHY

8.1 REFERENCES

[1]Dennis O'Neil(1999). "Blood Components".Palomar College. Archived from the original on June 5,2013.

[2]Tuskegee University(May 29, 2013)."Chapter 9 Blood".tuskegee.edu. Archived from the original on December 28, 2013.

[3]"Ways to Keep Your Blood Plasma Healthy". Archived from the original on November 1, 2013.Retrieved November 10, 2011.

[4]Jump up to Maton, Anthea; Jean Hopkins; Charles Wiliam McLaughlin; Susan Johnson; Maryanna Quon Warner LaHart; David LaHart; Jill D. Wright (1993), Human Biology and Health, Englewood Cliffs, New Jersey,USA.

[5]The Physics Factbook - Density of Blood.

[6]Basic Biology(2015)."Blood cells".

[7]Elkassabany NM, Meny GM, Doria RR, Marcucci C (2008). "Green Plasma Revisited". Anesthesiology 108(4);

[8]"19th WHO Model List of Essential Medicines(April 2015)"(PDF). WHO April 2015. Retrieved May 10, 2015

[9]Tripathi S, Kumar V,Prabhakar A, Joshi S, Agarwal A(2015)."Passive blood plasma separation at the microscale; a review of design principles and microdevices". J.Micromech, Microeng 25(8); 083001.

[10]Guo, Weijin; Hansson, Jonas; van der wijngaart, Wouter(2020)."Synthetic Paper Separates Plasma from Whole Blood with Low Protein Loss".Analytical Chemistry.92(9): 6194-6199.

[11]Mani A, Poornima AP, Gupta D(2019) "Greenish discoloration of plasma: Is it really a matter of concern?", Asian Journal of Transfusion Science.

[12]Starr, Douglas P. (2000), Blood: An Epic History of Medicine and Commerce. New York:Quill.

8.2 WEBSITES

- https://www.tutorialspoint.com/python/index.html
- https://www.javatpoint.com/python-tutorial

8.3 GITHUB Link

- **https://github.com/JalliTejaswini/Instant-Plasma-Donor-Recipient-Connector-Web-Application-**