# PLASMA DONAR MANAGEMENT

## 1. INTRODUCTION

### 1.1 PROJECT OVERVIEW

During the COVID 19 crisis, the requirement of plasma became a high priority and the donor count has become low. Saving the donor information and helping the needy by notifying the current donors list, would be a helping hand. In regard to the problem faced, an application is to be built which would take the donor details, store them and inform them upon a request.

The main goal of our project is to design a user-friendly web application that is like a scientific vehicle from which we can help reduce mortality or help those affected by COVID19 by donating plasma from patients who have recovered without approved antiretroviral therapy planning for a deadly COVID19 infection, plasma therapy is an experimental approach to treat those COVID-positive patients and help them recover faster. Therapy, which is considered reliable and safe. If a particular person has fully recovered from COVID19, they are eligible to donate their plasma.

As we all know, the traditional methods of finding plasma, one has to find out for oneself by looking at hospital records and contacting donors have been recovered, sometimes may not be available at home and move to other places. In this type of scenario, the health of those who are sick becomes disastrous. Therefore, it is not considered a rapid process to findplasma. The main purpose of the proposed system, the donor who wants to donate plasma can simply upload their covid19 traced certificate and can donate the plasma to the blood bank, the blood bank can apply for the donor and once the donor has accepted the request, the blood bank can add the units they need and the hospital can also send the

request to the blood bank that urgently needs the plasma for the patient and can take the plasma from the blood bank.

## 1.2 PURPOSE

The scope of the work is plasma donor management scheme IBM cloud web management scheme is deployed for all admin details ,user details.

# 2 LITERATURE SURVEY

## 2.1 EXISTING PROBLEM

| S.NO | TITLE | AUTHOR | ABSTRACT |
|---|---|---|---|
| 1. | Blood bag – A web application tomanage all blood donation and transfusion processes | Rehab S. Ali<br>Tamer F. Hafez<br>Ali Badawey Ali<br>Nadia Abd-Alsabour | Many lives could be lost due to the difficulty in obtaining a proper blood bag, Therefore, this work aims to help citizens execute their needs for a safe and reliable blood group by searching for and locating a specific blood group. In this paper, we illustrate the problem of the blood bags shortage which is represented in the uncontrolled blood banks and parallel markets, lack of awareness and confidence, disappearance of the rare blood groups, and the difficulty in finding a specific blood group. Hence, we proposed theBlood Bag web-based application that is connected to a centralized database to gather and organize the data from all blood banks and blood donation campaigns. The proposed application organizes and controls the whole critical processes related to blood donation, testing and storage of blood bags, and delivering it to the patient. |
| 2. | mHealth – Blood donation application using android smart phone | Muhammad FahimHalil Ibrahim Cebe Jawab Rasheed Farzad Kiani | mHealth is new horizons for health that offers healthcare services by utilizing the mobile devices and communication technologies. In health care services, blood donation is a complex process and consumes time to find some donor who has the compatibility of blood group with the patient. We developed android based blood donation application as mHealth solutions to establish a connection between the requester and donor at anytime and anywhere. The objective of this application is to provide the information about the requested blood and number of available donors around those localities. It assists the requester to broadcast the message |
| 3. | Web based online blood donationsy stem | Rohit | This paper depicts a high level program to close the hole between blood givers and individuals needing blood. The Online Blood donation Administration Framework application is an approach to synchronize blood donation centers with emergency clinics with the assistance of the Web. It is a web application |

| | | | |
|---|---|---|---|
| | | | where enlisted clinics can check the accessibility of the necessary Bloodand can send a blood solicitation to the closestblood donation center or comparable contributor as per the blood and can be controlled online through where fundamental.Blood donation center can likewise send a solicitation to another blood donation center that isn't accessible. Anybody willing to give blood can be found at the closest blood donation center utilizing the Android Bank The xecutivees Framework. Blood donation center can be followed utilizing maps. The Android application is simply accessible to benefactors to look for blood gifts and ask blood donation centers and clinics to search out blood donation centers and close by givers. |
| 4. | Developing a Plasma donor application using Function-as-a-service in AWS | Aishwarya RGowri | A plasma is a liquid portion of the blood, over55% of human blood is plasma. Plasma is used to treat various infectious diseases and itis one of the oldest methods known as plasmatherapy. Plasma therapy is a process where blood is donated by recovered patients in order to establish antibodies that fights the infection. In this project plasma donor application is being developed by using AWSservices. The services used are AWS Lambda, API gateway, DynamoDB, AWS Elastic Compute Cloud with the help of these AWS services, it eliminates the need of configuring the servers and reduces the infrastructural costs associated with it and helps to achieve<br>serverless computing. For instance, during COVID 19crisis the requirement for<br>plasma |

## 2.2 REFERENCES


[1] Danic B. Comprendre le don : l'apport des sciences humaines à l'activité de prélèvement. Transfus Clin Biol 2003;10:146–50,

[2] Bednall TC, Bove LL. Donating blood: a meta-analytic review of selfreported motivators and deterrents. Transfus Med Rev 2011;25:317–34,

[3] Bagot KL, Murray AL, Masser BM. How can we improve retention of the first-time donor? A systematic review of the current evidence. Transfus Med Rev 2016;30:81–91

[4] Bandura A. Self-efficacy: toward a unifying theory of behavioral change. Psychol Rev 1977;84:191–215,

[5] Robert P. Global plasma demand in 2015. Pharm Policy Law 2009:359–67,

[6] Cataldo JF, Cohen E, Morganti JB. Motivation of voluntary plasmapheresis donors. Transfusion (Paris) 1976;16:375–9

[7] Bagot KL, Bove LL, Masser BM, Bednall TC, Buzza M. Perceived deterrents to being a plasmapheresis donor in a voluntary, nonremunerated environment. Transfusion (Paris) 2013;53:1108–19

[8] Bagot KL, Masser BM, White KM. A novel approach to increasing inventory with the current panel: increasing donation frequency by asking for a different blood product. Transfusion (Paris) 2015;55:1294–302,

[9] Bove LL, Bednall T, Masser B, Buzza M. Understanding the plasmapheresis donor in a voluntary, nonremunerated environment. Transfusion (Paris) 2011;51:2411–24

[10] Charbonneau J, Cloutier M-S, Carrier É. Whole blood and apheresis donors in Quebec, Canada: demographic differences and motivations to donate. Transfus Apher Sci 2015;53:320–8,

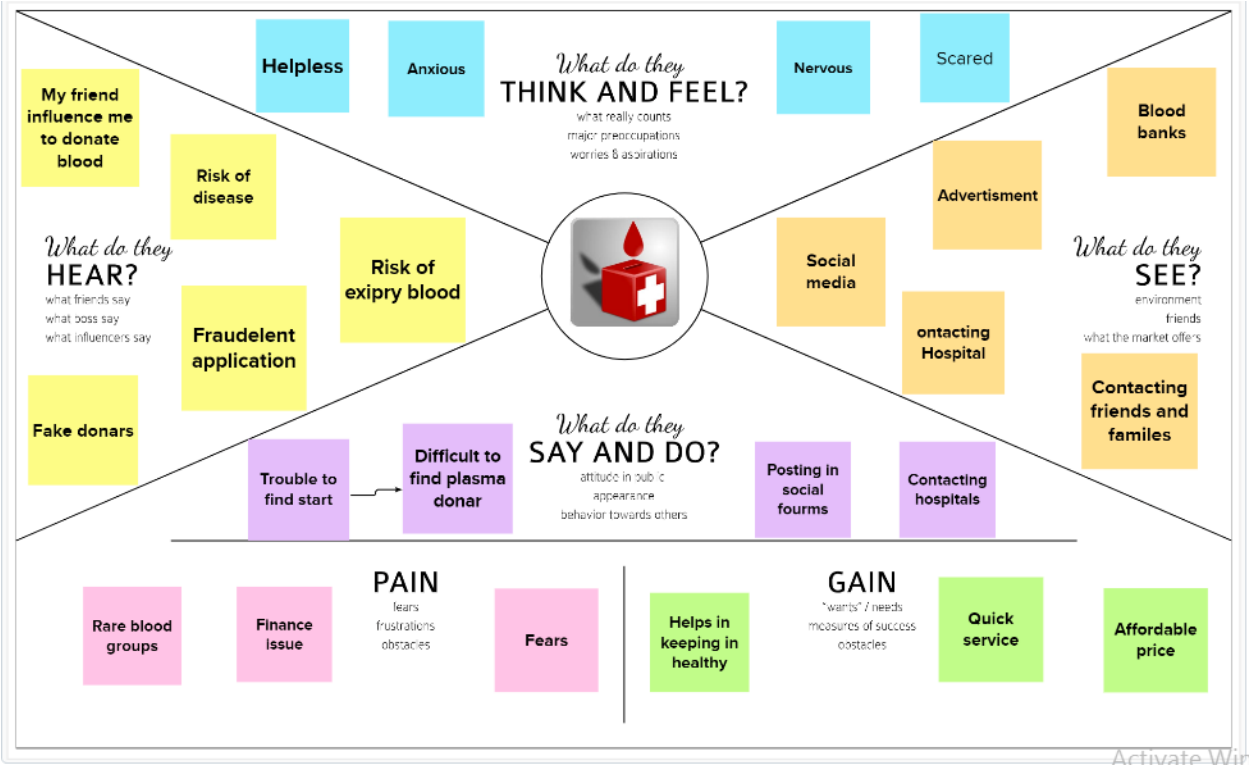
## 2.3 PROBLEM STATEMENT DEFINITION

During the COVID 19 crisis, the requirement of plasma became a high priority and the donor count has become low. Saving the donor information and helping the needy by notifying the current donors list, would be a helping hand. In regard to the problem faced, an application is to be built which would take the donor details, store them and inform them upon a request.

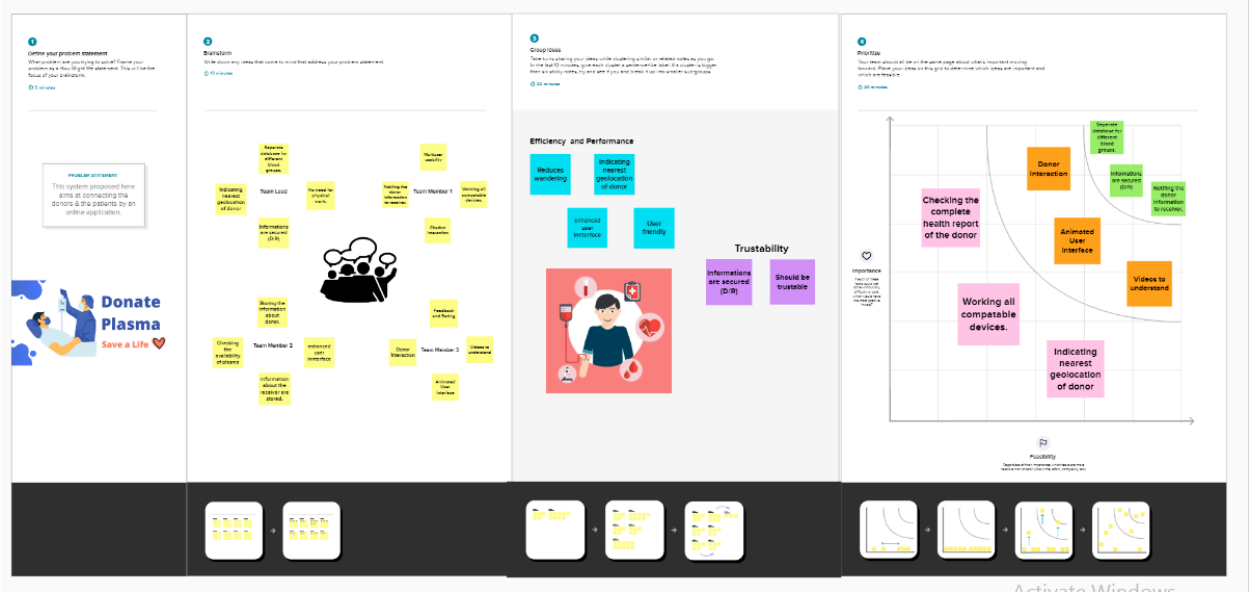
Plasma is a liquid portion of the blood, over 55% of human blood is plasma. Plasma is used to treat various infectious diseases and it is one of the oldest methods known as plasma therapy. Plasma therapy is a process where blood is donated by recovered patients in order to establish antibodies that fights the infection. For instance, during COVID 19 crisis the requirement for plasma increased drastically as there were no vaccination found in order to treat the infected patients, with plasma therapy the recovery rates where high but the donor count was very low and in such situations it was very important to get the information about the plasma donors. Saving the donor information and notifying about the current donors would be a helping hand as it can save time and help the users to track down the necessary information about the donors.

# 3. IDEATION & PROPOSED SOLUTION

## 3.1 Empathy Map Canvas:




## 3.2 Ideation & Brainstorming:



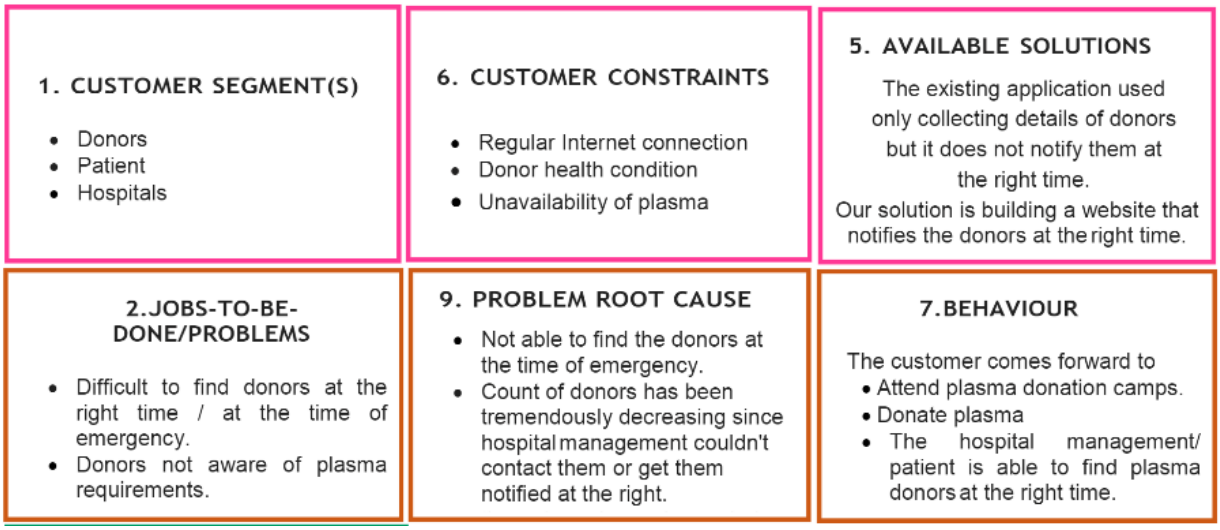
### 3.3 Proposed Solution:

Project team shall fill the following information in proposed solution template.

| S.NO. | PARAMETER | DESCRIPTION |
|---|---|---|
| 1 | Problem Statement (Problem to be solved) | Plasma Donor identifying and providing it to the required peoples. |
| 2 | Idea / Solution description | Plasma is a liquid portion of the blood, over 55% of human blood is plasma. Plasma is used to treat various infectious diseases and it is one of the oldest methods known as plasma therapy. Plasma therapy is a process where blood is donated by recovered patients in order to establish antibodies that fight the infection. In this project plasma donor application is being developed by using AWS services. |
| 3 | Novelty / Uniqueness | Plasma Donor Application by integrating it with IBM cloud. |
| 4 | Social Impact / Customer Satisfaction | Effect of donor motivation on donor satisfaction and loyalty is variable due to the influence of common donor ship attitudes prevailing in donor population, impact of social marketing programs, focused on promotion of donor commitment and deliberate donor ship. Thus, we have predicted that effect of |

| | | |
|---|---|---|
| | | donor motivation on donor relationship satisfaction and loyalty change |
| 5 | Business Model (Revenue Model) | • Can collaborate with plasma donor agencies.<br>• Can collaborate with Hospitals. |
| 6 | Scalability of the Solution | **PLASMA DONOR APPLICATION**<br>The results are represented with the scalability of different logins in any application. It tells the systematic process of each user login scalability in a systematic way. |

### 3.4 Problem Solution fit:



### 3. TRIGGERS

Blood donation improves or saves lives and enhances social solidarity. It is also influenced by increasing deaths due to unavailability of plasma at required times.

### 4.EMOTIONS: BEFORE/AFTER

**Before:**

Patient/ hospital find it hard to get a right resource to get plasma leaving them upset.

**After:**

The donors and customers have a feeling of satisfaction.

### 10. YOUR SOLUTION

Creating website which will provide information about available donors and plasma. If not available, the customer will be notified when plasma is available.

### 8.CHANNELS OF BEHAVIOUR

**Online:**

Can use the website to find donors.

**Offline:**

Can use the record maintain by the hospital.

# REQUIREMENT ANALYSIS

## 4.1 Functional requirement:

Following are the functional requirements of the proposed solution.

| FR NO. | NON-FUNCTIONAL REQUIREMENT | DESCRIPTION |
|---|---|---|
| NFR-1 | **Usability** | Must have a good looking User friendly interface. |
| NFR-2 | **Security** | It must be secured with the proper username and password. |
| NFR-3 | **Reliability** | The system should be made in such a way that it is reliable in its operations and for securing the sensitive details. |
| NFR-4 | **Performance** | Users should have a proper Internet Connection. |
| NFR-5 | **Availability** | The system including the online and offline components should be available 24/7. |
| NFR-6 | **Scalability** | The application has the ability to handle growing numbers of users and load without compromising on performance and causing disruptions to user experience. |

**Non-functional Requirements:**

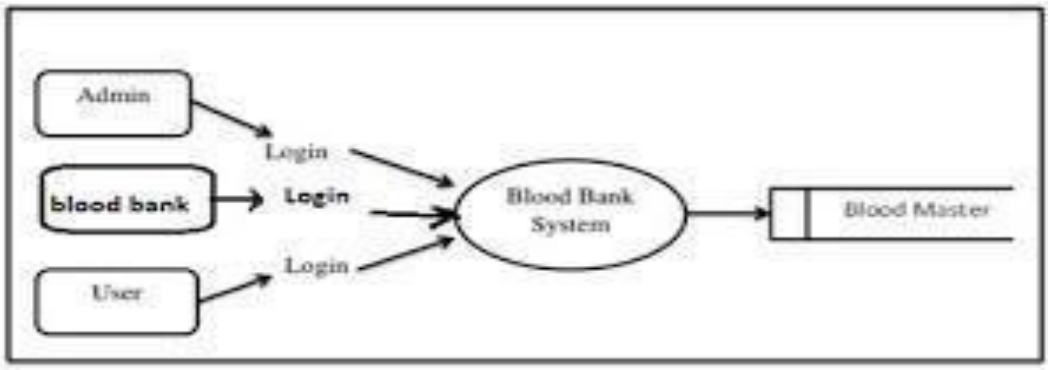
Following are the non-functional requirements of the proposed solution.

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Registration through Form (WebApp) |
| FR-2 | User Confirmation | Confirmation via Email<br>Confirmation via OTP |
| FR-3 | Certification | After the donor donates plasma, we will give them a certificate of appreciation and authentication. |
| FR-4 | Statistical data | The availability of plasma is given in the page as stats, which will be helpful for the users. |
| FR-5 | User Plasma Request | Users can request to donate plasma by filling out the request form on the page.<br>Once the request is submitted, they will get an email |
| FR-6 | Searching /reporting requirements | Users can use the search bar to look up information about camps and other topics. |
| FR-7 | Virtual Assistants | A virtual assistant is a software agent that can carry out tasks or provide services on behalf of a person in response to commands or inquiries.<br>When users enter their inquiries, the system will respond with pertinent information about plasma and details of plasma donation. |

# PROJECT DESIGN:

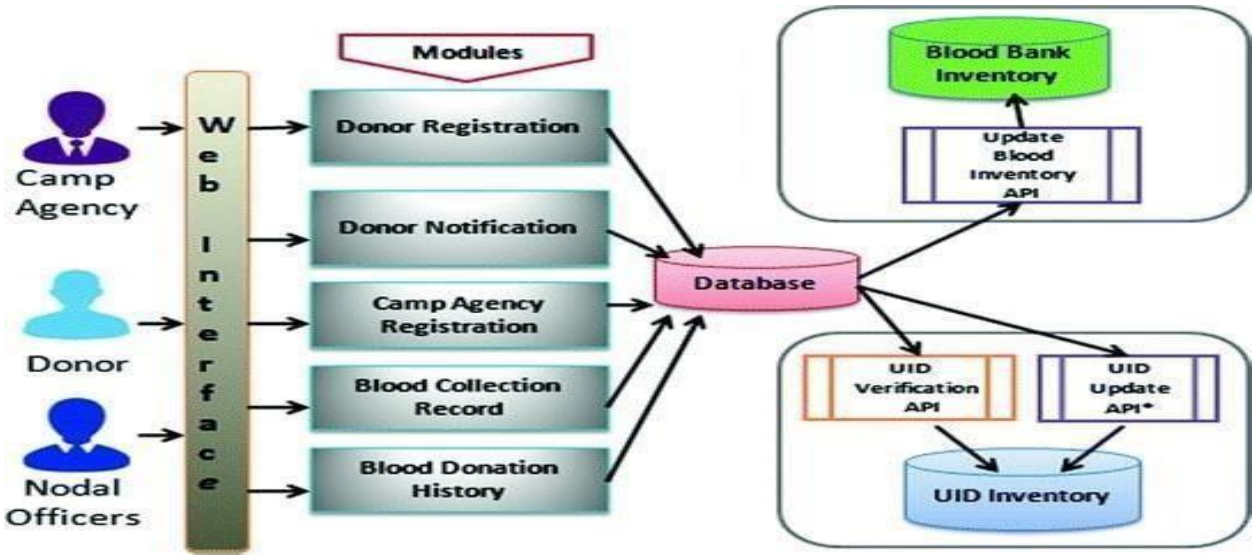
## 5.1 Data Flow Diagrams:

### Plasma Donor Application Data Floe Diagram




**LEVEL 0**


**LEVEL 1**

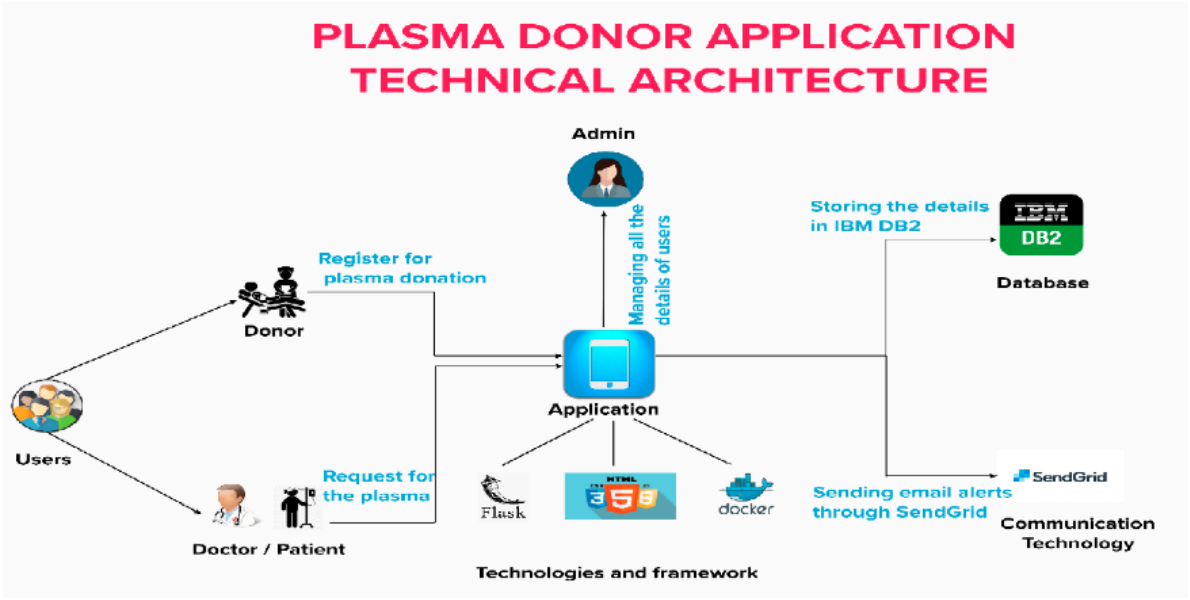

Modules
Camp Agency
Donor
Nodal Officers
Web Interface
Donor Registration
Donor Notification
Camp Agency Registration
Blood Collection Record
Blood Donation History
Database
Blood Bank Inventory
Update Blood Inventory API
UID Verification API
UID Update API*
UID Inventory

**LEVEL 2**

## 5.2 Solution & Technical Architecture:

## Solution Architecture:

## Technical Architecture:




## 5.3 User Stories:

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| Donor | App Registration | USN-1 | As a user, I can register for the application byentering my email, password, and confirming my password. | I can access my account /dashboard | High | Sprint-1 |
| | Login | USN-2 | As a user, I can log into the application by entering email & password | I can receive confirmation email & click confirm | High | Sprint-2 |
| | Register For Donate | USN-3 | As a user, I can log into the application andfind the current bank to donate plasma and confirm my booking | I can register & access thedashboard with Facebook Login | Medium | Sprint-3 |
| patient/doct | Find the bank | USN-4 | As a patient, I can directly | I can access my account / | High | Sprint-1,2 |

| | | | | | | |
|---|---|---|---|---|---|---|
| or | | | access the application and find the plasma available bank | dashboard | | |
| | Request for plasma | USN-5 | As a user, I can enter into the application andfind the current bank and request for plasma and state the emergency | I can register & access the dashboard with FacebookLogin | Medium | Sprint-3 |
| Administrator | Maintain the applications | USN-6 | As Administrator I can log into the application by entering email & password and maintaining details for users | I can access my account /dashboard | High | Sprint-3 |
| | Connect The Bank With Users | USN-7 | As Administrator, i can hold the good communication between bank and user | I can access my account / dashboard | Low | Sprint-4 |
| | Maintain Database | USN-8 | As Administrator i can hold the exact detailsof donor and patient and also bank for requesting and available of plasma | I can access my account /dashboard | Medium | Sprint-4 |
| Plasma Bank | Connect The Bank With Users | USN-7 | As Bank, i can hold the good communication between Administrator and user | I can access my account / dashboard | Medium | Sprint-3 |
| | Maintain Database | USN-8 | As Bank i can hold the exact details of donorand patient and also bank for requesting and available of plasma | I can access my account /dashboard | High | Sprint-4 |
| BOT | Help the user my bot mesg in appliation | USN-9 | As AI bot, i can hold the good communication between bank and user also help the user | I can access my account /dashboard | Medium | Sprint-4 |

# PROJECT PLANNING & SCHEDULING

## 6.1 Sprint Planning & Estimation:

| Sprint | Functional Requirement(Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Registration | USN-1 | A user can register for the application by entering their email, password, andconfirming the password. | 3 | High | Nithishkumar<br>Boopathi rajan<br>Kamalesh kannan |
| Sprint-1 | Email verification | USN-2 | A user will receive confirmation email once they have registered for the application. | 3 | High | Nithishkumar<br>Gunasekar<br>Kamalesh kannan |
| Sprint-1 | | USN-3 | A user can registe r for the applic ation throug h | 2 | Medium | Nithishkumar<br>Boopathi rajan |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | Googl<br>e | | | |
| Sprint-1 | Login | USN-4 | A user can log into the application by entering email & password | 2 | High | Nithishkumar<br>Kamalesh<br>kannan |
| Sprint-1 | Donor Profile | USN-5 | A user is able to register themselves asverified plasma donor. | 3 | High | Nithishkumar<br>Boopathi<br>rajan<br>Kalyanapriya<br>n |
| Sprint-2 | Virtual Certificate | USN-6 | A user will get a virtual donor certificate after a verified successful plasma donation. | 2 | Medium | Nithishkuma<br>r Boopathi<br>rajan<br>Kalyanapriy<br>an |

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-2 | Plasma Request | USN-7 | A verified clinic is able to make a plasma request in the application | 3 | High | Nithishkumar Gunasekar Kamalesh kannan |
| Sprint-2 | Verification of Donor's details | USN-8 | We the administrators will verify the details provided by the donorsso only the genuine donors are able to use the application | 2 | Medium | Boopathi rajan Kamalesh kannan |
| Sprint-3 | Accept the donation request | USN-9 | A user and a registered donor will geta notification to accept the plasma request for their specific blood type. | 3 | High | Nithishkumar Boopathi rajan Kamalesh kannan |
| Sprint-3 | Communication Channel | USN-10 | A patient is able to communicate with the donor personally within the application. | 3 | Medium | Nithishkumar Boopathi rajan Kalyanapriyan |
| Sprint-3 | | USN-11 | A user and a registered donor is ableto share their location with the recipient after accepting their plasmarequest. | 3 | Medium | Nithishkumar Boopathi rajan Kamalesh kannan |
| Sprint-3 | Administrator | USN-12 | An admin will store the registered donor's details after | 3 | High | Nithishkumar kalyanapriyan Kamalesh kannan |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | verification into the database. | | | |
| Sprint-4 | Support | USN-13 | A user is able to ask basic question related to plasma donation with thehelp of chat-bot.s | 2 | Medium | Nithishkumar Boopathi rajan Gunasekar |

## 6.2 Sprint Delivery Schedule:

| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned EndDate)** | **Sprint Release Date (Actual)** |
|---|---|---|---|---|---|---|
| Sprint-1 | 18 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 18 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 20 | 05 Nov 2022 |
| Sprint-3 | 18 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 18 | 12 Nov 2022 |
| Sprint-4 | 18 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 18 | 19 Nov 2022 |

## 6.3 Reports from JIRA:

**Velocity:**

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) | Average Velocity(AV) = Sprint duration /velocity |
|---|---|---|---|---|---|---|---|
| Sprint-1 | 18 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 18 | 29 Oct 2022 | 3 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 20 | 05 Nov 2022 | 3.33 |
| Sprint-3 | 18 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 18 | 12 Nov 2022 | 3 |
| Sprint-4 | 18 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 18 | 19 Nov 2022 | 3 |

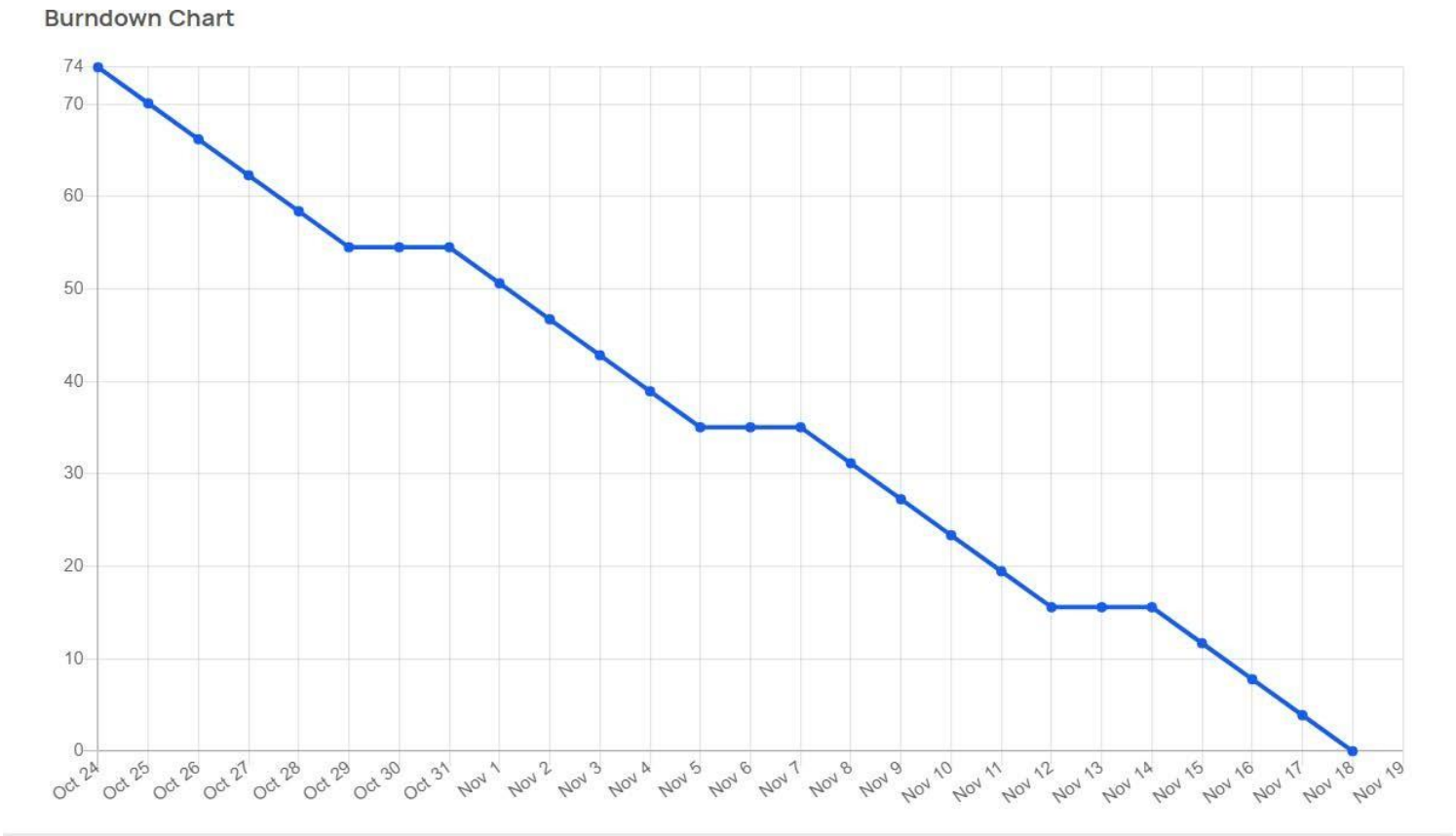
Total number of days = sprint 1 + sprint 2 + sprint 3 + sprint 4 = 6 + 6 + 6 + 6 = 24

Total number of story points = 18 + 20 + 18 + 18 = 74

Average velocity per sprint = 74 / 24

~= 3.083333

= 3




**Estimated Burn down Chart**

## 7. CODING & SOLUTIONING:

**CODING**:

```
from flask import Flask, render_template, flash, request,session

from flask import Flask, render_template, request, jsonify

import datetime

import re

import ibm_db

import pandas

import ibm_db_dbi

from sqlalchemy import create_engine

engine = create_engine('sqlite://',

                       echo = False)

dsn_hostname = "9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud"

dsn_uid = "slg84898"

dsn_pwd = "sQLhssDgMcqDZ0uR"

dsn_driver = "{IBM DB2 ODBC DRIVER}"

dsn_database = "bludb"
```

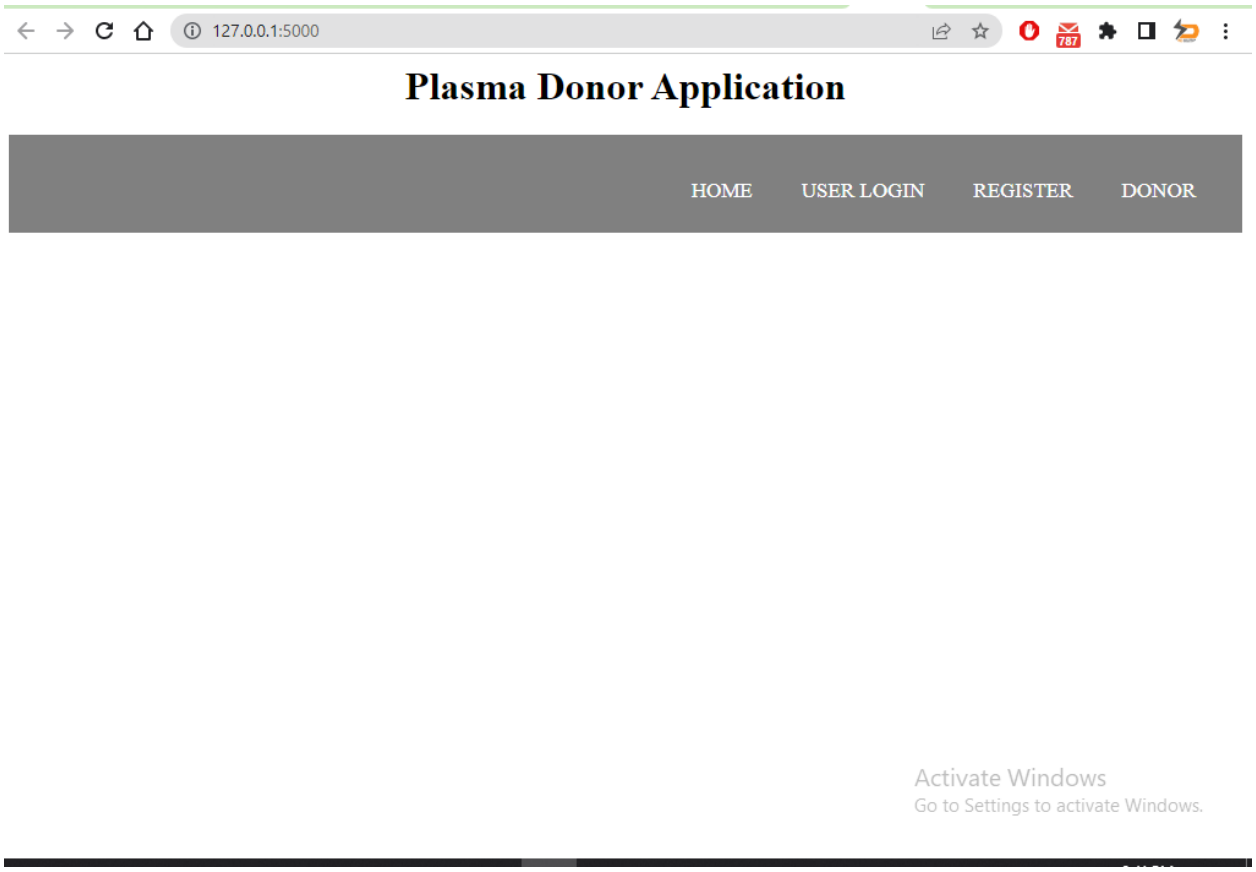
```
dsn_port = "32459"

dsn_protocol = "TCPIP"

dsn_security = "SSL"

dsn = (

    "DRIVER={0};"

    "DATABASE={1};"

    "HOSTNAME={2};"

    "PORT={3};"

    "PROTOCOL={4};"

    "UID={5};"

    "PWD={6};"

    "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname, dsn_port, dsn_protocol, dsn_uid, dsn_pwd,dsn_security)

try:

    conn = ibm_db.connect(dsn, "", "")

    print ("Connected to database: ", dsn_database, "as user: ", dsn_uid, "on host: ", dsn_hostname)

except:

    print ("Unable to connect: ", ibm_db.conn_errormsg() )

app = Flask(__name__)
```

app.config.from_object(__name__)

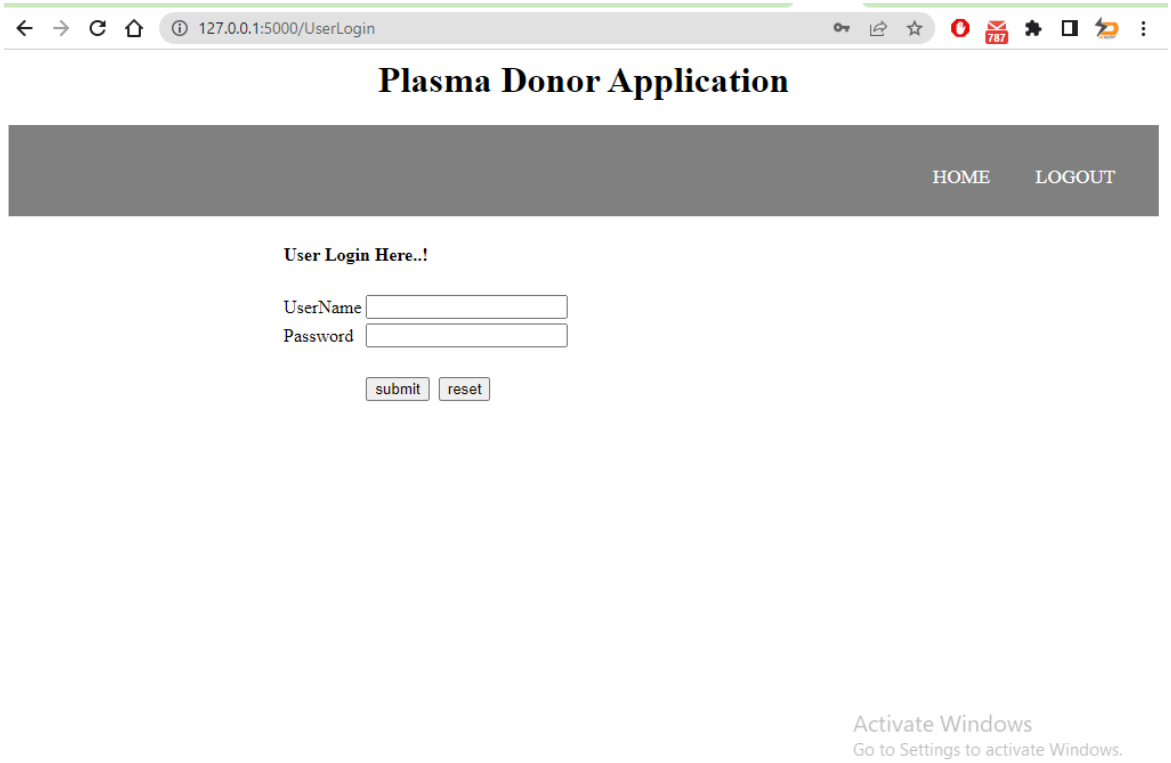
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'

**SCREENSHOT:**

**HOME PAGE:**

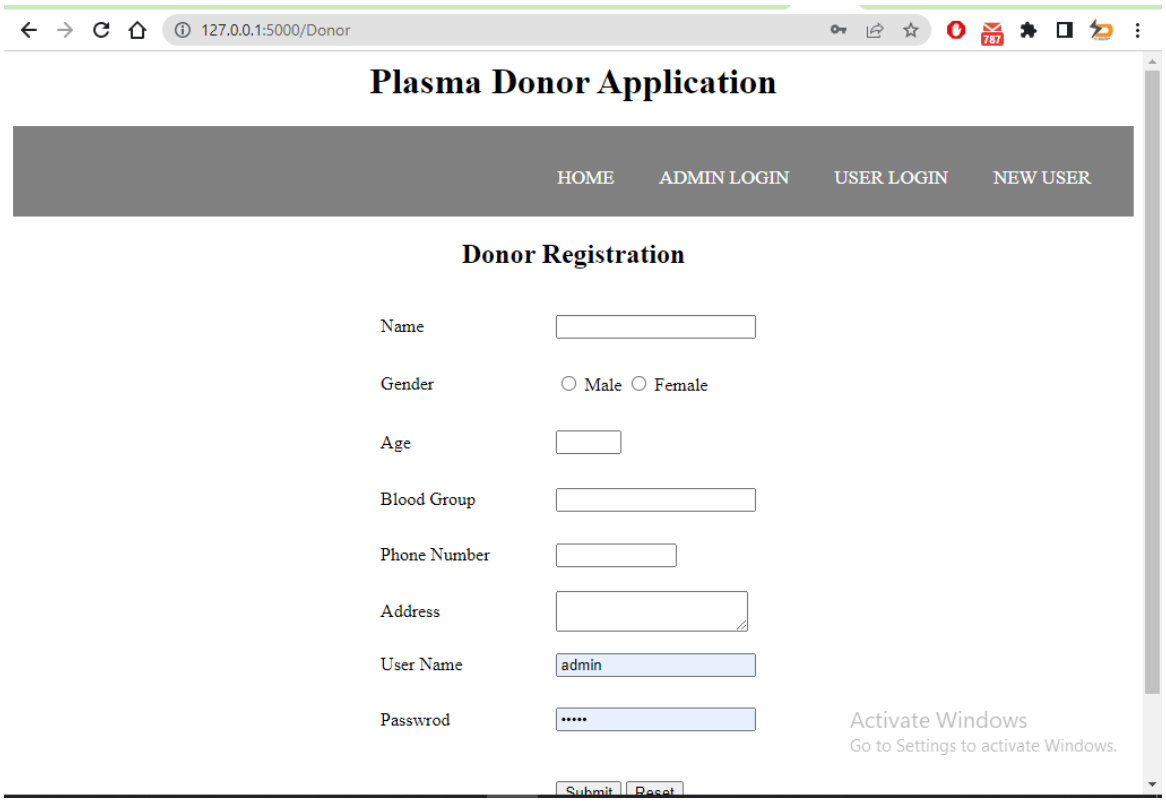


## USER LOGIN:

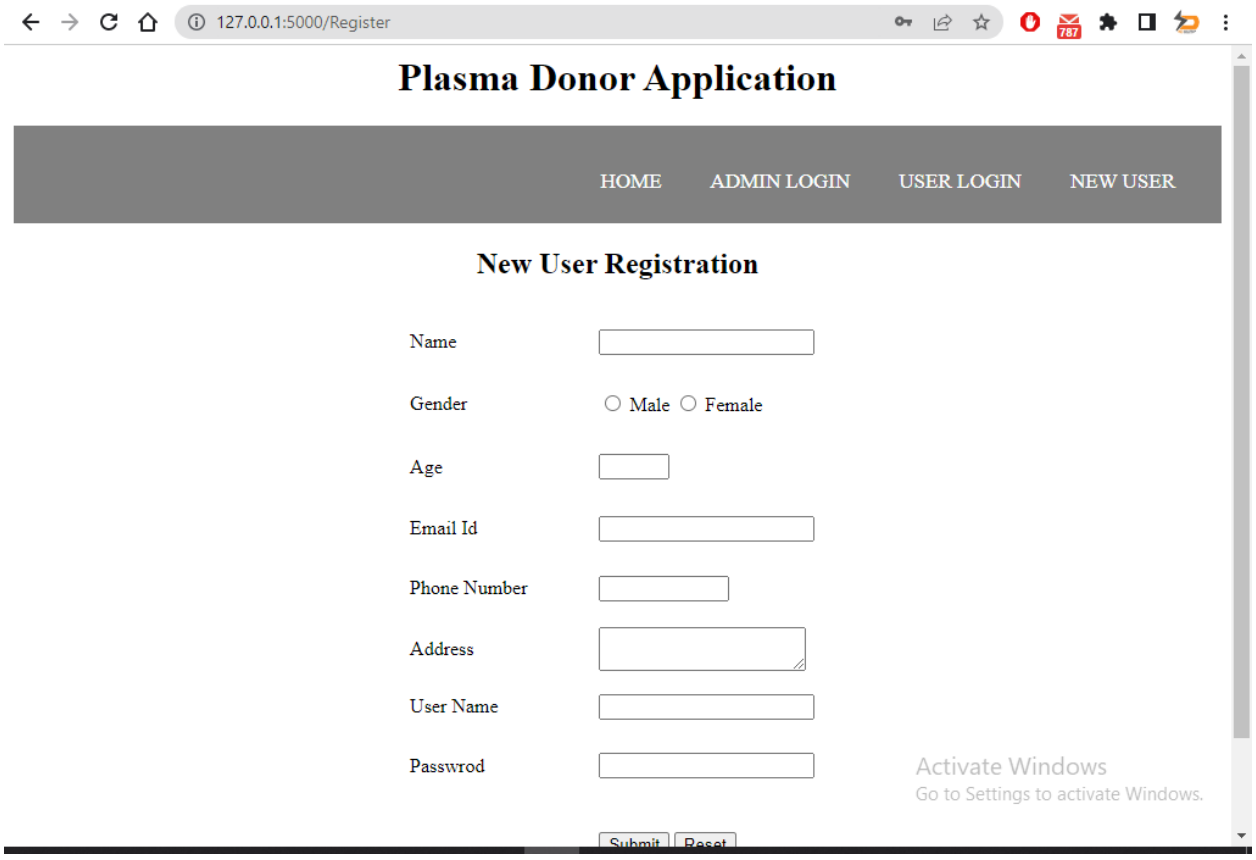



Activate Windows
Go to Settings to activate Windows.

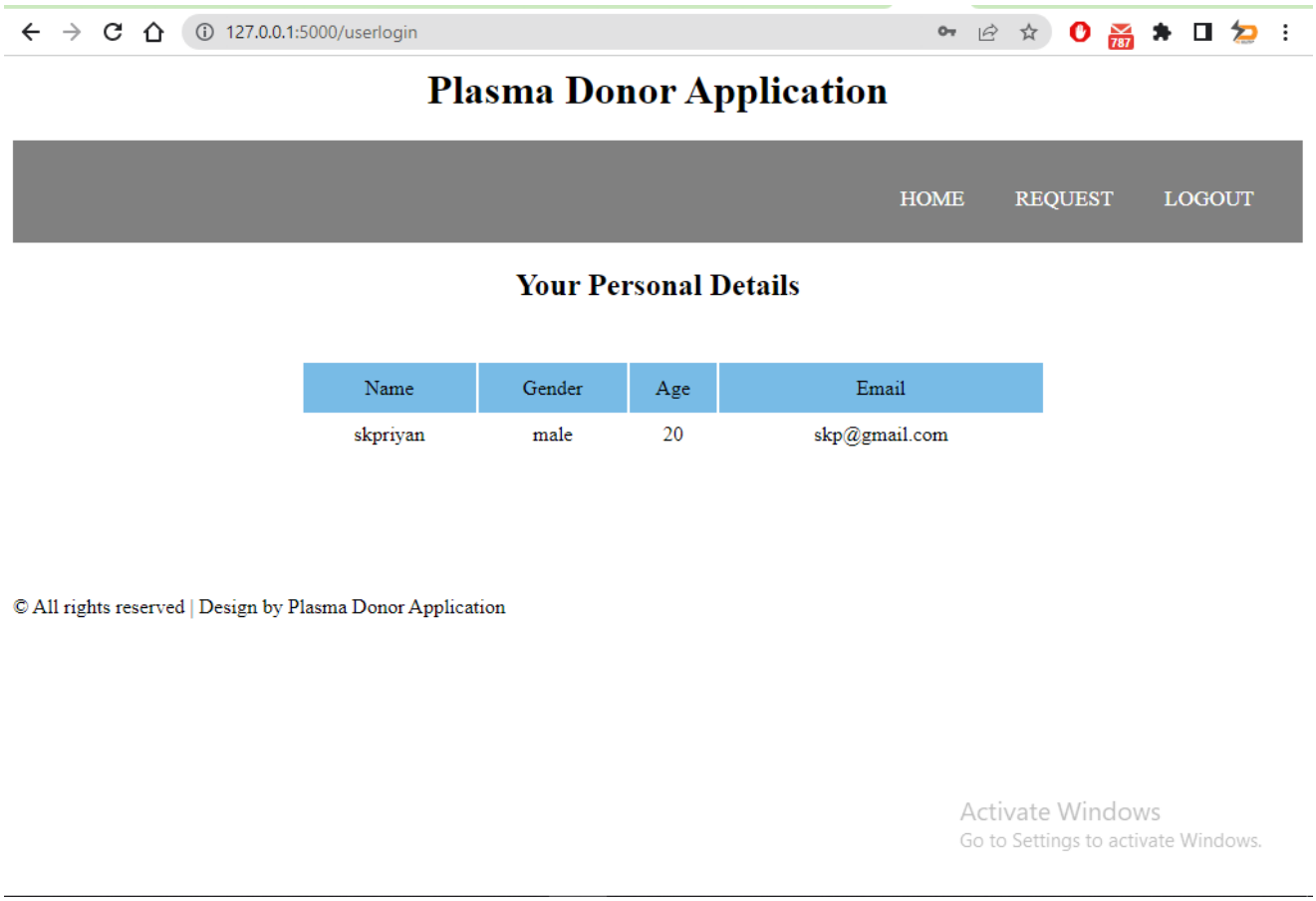
# DONOR REGISTRATION



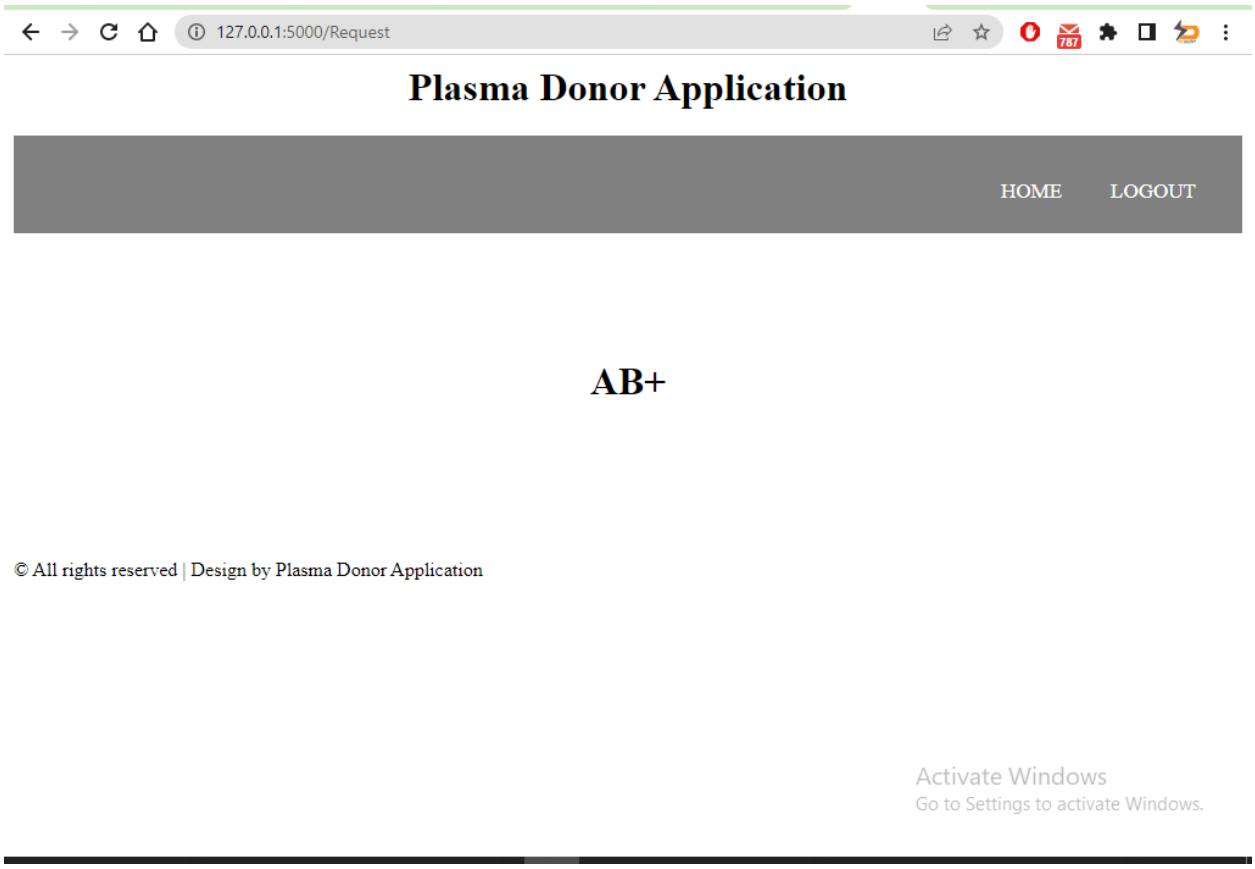
# NEW USER REGISTRATION



# YOUR PERSONAL DETAILS



## RESULT


127.0.0.1:5000/Request
Plasma Donor Application
HOME
LOGOUT
AB+
© All rights reserved | Design by Plasma Donor Application
Activate Windows
Go to Settings to activate Windows.

# 8 TESTING

## 8. SYSTEM 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, sub-assemblies, 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 test. Each test type addresses a specific testing requirement.

## 8.1 TYPES OF TESTS

### 8.1.1 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.

### 8.1.2 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 satisfaction, as shown by successfully 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.

### 8.1.3 Functional test

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:

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

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

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, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### 8.1.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test.

System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

### 8.1.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

### 8.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

## 8.2 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

### 8.2.1 Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

### 8.2.2 Test objectives

- All field entries must work properly.

- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

### 8.2.3 Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

## 8.3 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

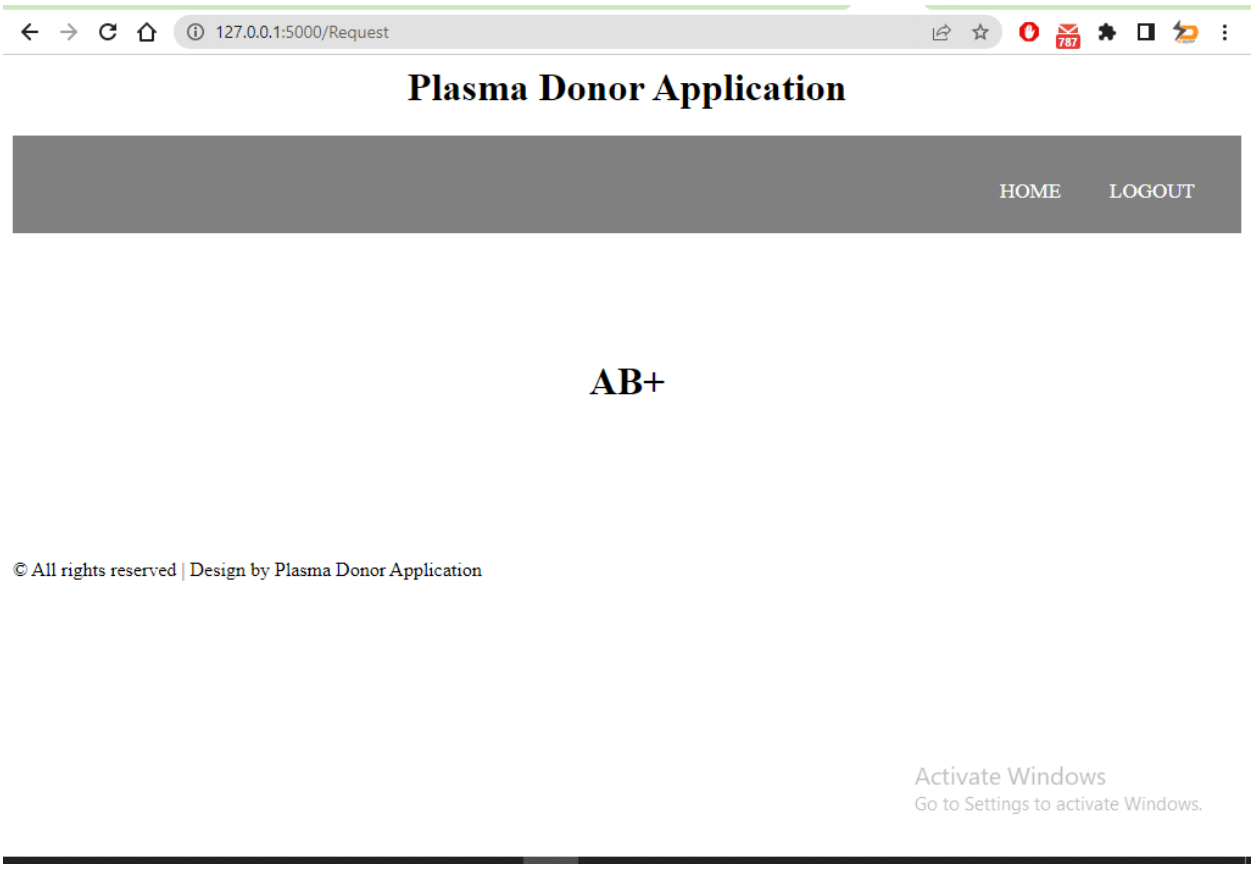
**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

## 8.4 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

# 9 RESULTS


127.0.0.1:5000/Request
Plasma Donor Application
HOME
LOGOUT
AB+
© All rights reserved | Design by Plasma Donor Application
Activate Windows
Go to Settings to activate Windows.

## 10 ADVANTAGES & DISADVANTAGES:

1.user friendly

2. Identify and formulate the problem

3. Identify the requirements, objective, and preferences

4. Determine project plan

5. Determine the feasibility of plan

## 11. CONCLUSION :

We design the plasma donor project for web based application using flask technology Plasma is the most essential thing to save a life. By donating blood, we can save many lives. It is also important to remember that any one of us may need blood at some point in our lives; making blood donation is an essential duty of our citizenry. In today's world where people are busy with their lifestyle and those who are eager to donate blood but are not able to, can plan to donate blood by sitting at home just by one click with our application This application will make revolutionary changes to the medical system as people will be able to donate blood and serve mankind. It can also help people to know about the benefits about blood donation and that their small contribution can help one person to save his/her lives as soon as possible in a quick and well managed manner.

## 12. FUTURE WORK

In future work, in this time of scarcity of blood donors, this study illustrates that a donor's sense of satisfaction is immediate but evanescent and may play a critical factor on his or her intent to return for future donations. Better customer service must therefore be included in the development of robust recruitment and retention efforts, as blood centers prepare for the challenge of a decreased donor base in the face of new testing recommendations and requirements for reducing the risk of TRALI and Chagas disease.25–28 As new interventions are implemented, additional studies of both donor satisfaction and motivation are needed to assess areas of strengths and weaknesses for ongoing quality improvement.

## 13. APPENDIX:

**Source Code:**

```
from flask import Flask, render_template, flash, request,session

from flask import Flask, render_template, request, jsonify

import datetime

import re

import ibm_db

import pandas

import ibm_db_dbi

from sqlalchemy import create_engine

engine = create_engine('sqlite://',

                       echo = False)

dsn_hostname = "9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud"

dsn_uid = "slg84898"

dsn_pwd = "sQLhssDgMcqDZ0uR"

dsn_driver = "{IBM DB2 ODBC DRIVER}"

dsn_database = "bludb"

dsn_port = "32459"
```

```
dsn_protocol = "TCPIP"

dsn_security = "SSL"

dsn = (

    "DRIVER={0};"

    "DATABASE={1};"

    "HOSTNAME={2};"

    "PORT={3};"

    "PROTOCOL={4};"

    "UID={5};"

    "PWD={6};"

    "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname, dsn_port, dsn_protocol, dsn_uid, dsn_pwd,dsn_security)

try:

    conn = ibm_db.connect(dsn, "", "")

    print ("Connected to database: ", dsn_database, "as user: ", dsn_uid, "on host: ", dsn_hostname)

except:

    print ("Unable to connect: ", ibm_db.conn_errormsg() )

app = Flask(__name__)

app.config.from_object(__name__)
```

```
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'

@app.route("/")
def homepage():
    return render_template('index.html')

@app.route("/AdminLogin")
def AdminLogin():
    return render_template('AdminLogin.html')

@app.route("/Register")
def Register():
    return render_template('Register.html')

@app.route("/UserLogin")
def UserLogin():
    return render_template('UserLogin.html')

@app.route("/Donor")
def Donor():
    return render_template('Donor.html')

@app.route("/RNewUser", methods=['GET', 'POST'])
def RNewUser():
    if request.method == 'POST':
```

```
        name1 = request.form['name']
        gender1 = request.form['gender']
        Age = request.form['age']
        email = request.form['email']
        address = request.form['address']
        pnumber = request.form['phone']
        uname = request.form['uname']
        password = request.form['psw']
        conn = ibm_db.connect(dsn, "", "")
        insertQuery = "INSERT INTO regtb VALUES ('" + name1 + "','" + gender1 + "','" + Age + "','" + email + "','" + pnumber + "','" + password + "','" + uname + "','" + address + "')"
        insert_table = ibm_db.exec_immediate (conn, insertQuery)
        print(insert_table)
 return render_template('userlogin.html')
@app.route("/RNewDonor", methods=['GET', 'POST'])
def RNewDonor():
    if request.method == 'POST':
        name1 = request.form['name']
        gender1 = request.form['gender']
```

```
        Age = request.form['age']
        blood = request.form['bgrp']
        address = request.form['address']
        pnumber = request.form['phone']
        uname = request.form['uname']
        password = request.form['psw']
        conn = ibm_db.connect(dsn, "", "")
        insertQuery = "INSERT INTO dotb VALUES ('" + name1 + "','" + gender1 + "','" + Age + "','" + blood + "','" + pnumber + "','" + password + "','" + uname + "','" + address + "')"
        insert_table = ibm_db.exec_immediate (conn, insertQuery)
        print(insert_table)
    return render_template('userlogin.html')
@app.route("/Request")
def Request():
    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * from dotb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('Employee_Data',
```

```
            con=engine,
            if_exists='append')
    # run a sql query
    print(engine.execute("SELECT * FROM Employee_Data").fetchall())
    return render_template('ViewProduct.html', data=engine.execute("SELECT * FROM Employee_Data").fetchall())
@app.route("/userlogin", methods=['GET', 'POST'])
def userlogin():
    error = None
    if request.method == 'POST':
        username = request.form['uname']
        password = request.form['password']
        session['uname'] = request.form['uname']
        conn = ibm_db.connect(dsn, "", "")
        pd_conn = ibm_db_dbi.Connection(conn)
        selectQuery = "SELECT * from regtb where uname='" + username + "' and password='" + password + "'"
        dataframe = pandas.read_sql(selectQuery, pd_conn)
        if dataframe.empty:
            data1 = 'Username or Password is wrong'
```

```
        return render_template('goback.html', data=data1)
    else:
        print("Login")
        selectQuery = "SELECT * from regtb where uname='" + username + "' and password='" + password + "'"
        dataframe = pandas.read_sql(selectQuery, pd_conn)
        dataframe.to_sql('Employee_Data',
                con=engine,
                if_exists='append')
        # run a sql query
        print(engine.execute("SELECT * FROM Employee_Data").fetchall())
        return render_template('UserHome.html', data=engine.execute("SELECT * FROM Employee_Data").fetchall())
def main():
    app.run(debug=True, use_reloader=True)
if __name__ == '__main__':
    main()
```