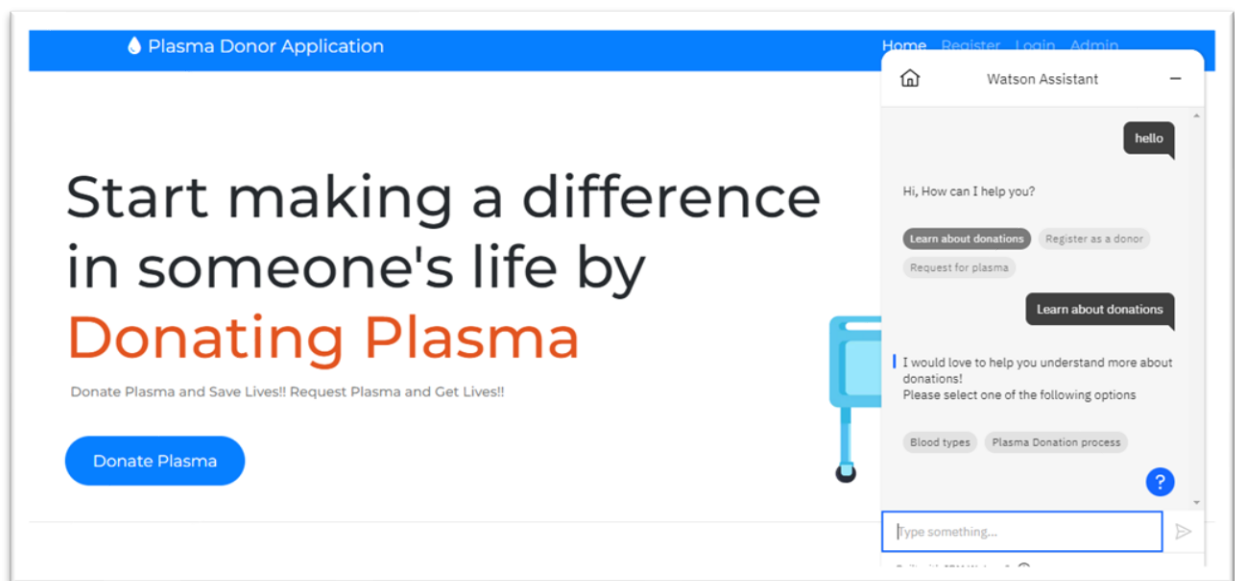


# PROJECT DEVELOPMENT PHASE

## SPRINT-2

Date	9 November 2022
Team ID	PNT2022TMID27090
Project Name	Plasma Donor Application

### INTEGRATING CHATBOT:



## LOGIN :

Plasma Donor Application

[Home](#) [Register](#) [Login](#)

LogIn using UserName and Password

User Name

Enter UserName

Password

Enter Password

[Forget Password](#)

LogIn

Don't have an account [Register here](#)

## USER PAGE :

Plasma Donor Application

[Home](#) [Register](#) [Login](#) [Admin](#)

Welcome : Sharath Kumar!!

Logged in successfully !

Donate Plasma

Request Plasma

Log Out

## REGISTER AS DONOR :

Plasma Donor Application

Home Register Login Admin

Donating Plasma

Full Name

Enter your Name

Mobile Number

Enter your Mobile No.

Email

Enter your Email

Age

Enter your Age

Gender

Select your Gender

Blood Group

Select your blood group

Area

Enter your Area Name

City

Enter your City Name

District

Enter your District Name

☐ I agree the terms and conditions

Reset

Submit

## REQUEST FOR PLASMA:

Plasma Donor Application

Home Register Login Admin

Request for Plasma

Full Name

Enter your Full Name

Email

Enter your Email

Mobile Number

Enter your Mobile No.

Hospital Name

Enter Hospital Nmae

Age

Enter your Age

Gender

Select your Gender

Blood Group

Select your blood group

Area

Enter your Area Name

City

Enter your City Name

District

Enter your District Name

☐ I agree the terms and conditions

Close

Request

# CONNECTION WITH IBM DATABASE:

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

NYB16417.LOGIN

Back

Export to CSV

USERNAME	USERMAIL	USERCONTACT	PASSWORD
NandhiniL	nandhini02lak@gmail.com	1234	nandhini
NandhiniLakshmpathy	nandhini02lak@gmail.com	1234	dass
Nandhinilak	nandhini02lak@gmail.com	12345678	nan
Sharath	nandhiniakshmpathy1307@gmail.com	7334567898	Nandhy
Sharath Kumar	nandhini02lak@gmail.com	7339286572	123456
Sharathtest	chinnukool72@gmail.com	1234	sharath

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

NYB16417.REQUEST2

Back

Export to CSV

DRMAIL	HOSPITALNAME	RECNAME	RECMOBILE	RECMAIL	RECAGE	RECGENDER	RECBLOODGROUP	RECAREA	RECCITY
chinnukool72@gmail.com	kmc	nandhini	123345	chinnukool72@gmail.com	21	Male	A+	chennai	chennai
nandhini@gmail.com	kmc	nandhinil	123456	nandhini@gmail.com	21	Female	A-	chennai	chennai











```
File Edit Selection View Go Run Terminal Help
sprint2

EXPLORER
  sprint2
    template
    base.html
    donor.html
    home.html
    login.html
    register.html
    request.html
    signs.html
    success.html
    user_profile.html
    app.py

success.html X
template > success.html > ...
1 {% extends 'base.html' %}
2
3 <!--title tag-->
4 {% block title %}
5 <title>Plasma-Success</title>
6 {% endblock %}
7
8 {% block link %}
9 {% endblock %}
10
11 <!--User Content-->
12 {% block content %}
13
14
15
16 <div class="container-fluid">
17 <div class="text-center">
18
19 <style>
20
21 .hide {
22   display: none;
23 }
24 .myDIV:hover + .hide {
25   display: block;
26   color: red;
27   text-align: top;
28 }
29
30 </style>
31
32
33
34
35 <div class="container">
36 <div class="row p-1">
37 <div class="col-sm-12">
38 <div class="card" style="margin-top: 70px;">
39 <div class="msg">{{ msg }}</div>
40 <div class="card-body">
41
42
43 <div class="myDIV">
44
45  <div class="hide">You are a real superhero!</div>
48
49 <h2>Thank you for donating plasma.</h2>
50 </div>
51 </div>
52 </div>
53 </div>
54
55
56
57 {% endblock %}
58
```

```
File Edit Selection View Go Run Terminal Help
sprint2

EXPLORER
  sprint2
    template
    base.html
    donor.html
    home.html
    login.html
    register.html
    request.html
    signs.html
    success.html
    user_profile.html
    app.py

user_profile.html X
template > user_profile.html > ...
1 {% extends 'base.html' %}
2
3 <!--title tag-->
4 {% block title %}
5 <title>Plasma-Profile</title>
6 <style>
7
8 .profile-area{
9   padding: 70px 0;
10  border: 2px solid #23226;
11  margin-right: 190px;
12  margin-left: 190px;
13  margin-top: 50px;
14
15  .card{
16    box-shadow: 0 0 30px #23226;
17    overflow: hidden;
18    border-radius: 15px;
19    margin-top: 30px;
20    background-image: linear-gradient(to top right, #23226 49%, #23226 49%, #23226 51%, #23226 51%);
21    height: 200px;
22
23    .card-hover{
24      border-color: #23226;
25      transform: rotate(1deg);
26      transition: .7s;
27    }
28    .main-text, .card-body{
29      text-align: center;
30      padding-top: 60px;
31
32    .btn{
33      padding: 10px 10px;
34      margin-bottom: 20px;
35    }
36  }
37 </style>
38 {% endblock %}
39
40 {% block link %}
41
42 <h2 class="page-header text-center" style="margin-top: 30px; color: black">Your P
43
44
45 <!--User Content-->
46 {% block content %}
47 <div class="container">
48 <h2 class="text-danger" style="margin-top: 50px;">Welcome : {{session['username']}}
49 </div>
50 <div class="profile-area">
51 <div class="container">
52 <div class="msg">{{ msg }}</div>
53 <div class="row">
54
55 <div class="col-12 col-md-6 col-lg-6">
56 <div class="card">
57 <div class="main-text card-body">
58 <div><a href="{{ url_for('register') }}" class="btn btn-light">Donate P
59 </div>
60 </div>
61 </div>
62 <div class="col-12 col-md-6 col-lg-6">
63 <div class="card">
64 <div class="main-text card-body">
65 <div><a href="{{ url_for('donorlist') }}" class="btn btn-light">Request
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 <div class="container">
73 <a href="{{ url_for('logout') }}" class="btn btn-danger">Log Out/</div>
74
75
76 {% endblock %}
77
```

```
static > # home.css > {} @media only screen and (max-width: 500px) {
1  @media only screen and (max-width: 500px) {
2      /* For mobile phones: */
3      .landing, .landing-text h1, .landing-text h3, .landing-text .btn, .img, .landing-text .btn a {
4          width: 100%;
5          height: auto;
6      }
7  }
8  .landing {
9      margin-top: 100px;
10     margin-left: 50px;
11     margin-right: 50px;
12 }
13 .landing-text h1 {
14     font-size: 65px;
15 }
16 .landing-text h3 {
17     margin: 6px;
18     font-size: 15px;
19     line-height: 1.8;
20     color: #ffffff;
21     margin-right: 20px;
22 }
23 .landing-text .btn {
24     width: 200px;
25     margin-top: 30px;
26     padding: 14px 20px 12px 20px;
27     background-color: #007bff;
28     border-radius: 45px;
29     text-align: center;
30 }
31 .landing-text .btn a {
32     font-size: 18px;
33     color: #ffff;
34 }
35 .img {
36     float: right;
37 }
38 }
```

```
static > # style.css > *
1  * {
2      font-family: 'Montserrat', sans-serif;
3  }
4  #icon {
5      color: white;
6      padding-right: 2px;
7  }
8  }
```

```
◆ app.py > ...
1 from flask import Flask, render_template, request, session, redirect, url_for
2 import ibm_db
3 import sendgrid
4 import os
5 import re
6 from sendgrid.helpers.mail import Mail
7
8 app = Flask(__name__)
9 app.secret_key = "1"
10
11 conn = ibm_db.connect("DATA")
12
13 @app.route("/")
14 def index():
15     return render_template("index.html")
16
17 @app.route("/home")
18 def home_page():
19     return render_template("home.html")
20
21
22 @app.route("/login", methods=['POST'])
23 def login():
24     global userid
25     msg = ''
26     if request.method == 'POST':
27         username = request.form['username']
28         password = request.form['password']
29         sql = "SELECT * FROM user WHERE username = ? AND password = ?"
30         stmt = ibm_db.prepare(conn, sql)
31         ibm_db.bind_param(stmt, 1, username)
32         ibm_db.bind_param(stmt, 2, password)
33         ibm_db.execute(stmt)
34         account = ibm_db.fetch_row()
35         print(account)
36         if account:
37             session['logged_in'] = True
38             session['id'] = account[0]
39             session['username'] = account[1]
40             msg = 'logged in successfully'
41             return redirect(url_for("home_page"))
42         else:
43             msg = 'Incorrect username or password'
44             return render_template("login.html", msg=msg)
45
46
47 ◆ app.py > ...
48 # After login
49 # After login
50 @app.route('/afterlogin')
51 def afterlogin():
52     return render_template("afterlogin.html")
53
54 #-----
55
56 @app.route('/signin', methods=['POST'])
57 def signin():
58     msg = ''
59     if request.method == 'POST':
60         username = request.form['username']
61         usermail = request.form['email']
62         password = request.form['password']
63         sql = "SELECT * FROM user WHERE username = ? AND password = ?"
64         stmt = ibm_db.prepare(conn, sql)
65         ibm_db.bind_param(stmt, 1, username)
66         ibm_db.bind_param(stmt, 2, password)
67         ibm_db.execute(stmt)
68         account = ibm_db.fetch_row()
69         print(account)
70         if account:
71             msg = 'Account already exists'
72             return render_template("signin.html", msg=msg)
73         else:
74             msg = 'Invalid email'
75             return render_template("signin.html", msg=msg)
76         else:
77             insert_sql = "INSERT INTO user (username, email, password) VALUES (?, ?, ?)"
78             prep_stmt = ibm_db.prepare(conn, insert_sql)
79             ibm_db.bind_param(prepare_stmt, 1, username)
80             ibm_db.bind_param(prepare_stmt, 2, usermail)
81             ibm_db.bind_param(prepare_stmt, 3, password)
82             ibm_db.execute(prepare_stmt)
83             msg = 'You have successfully registered'
84             return render_template("signin.html", msg=msg)
85         # mailtest(user)
86         return render_template("signin.html", msg=msg)
87     elif request.method == 'GET':
88         msg = 'Please fill the form'
89         return render_template("signin.html", msg=msg)
90
91
92 @app.route("/register")
93
94 ◆ app.py > ...
95
96 @app.route("/register")
97 def register():
98     return render_template("register.html")
99
100
101 @app.route("/addonor", methods=['POST'])
102 def addonor():
103     if request.method == 'POST':
104         name = request.form['name']
105         mobile = request.form['mobile']
106         email = request.form['email']
107         age = request.form['age']
108         gender = request.form['gender']
109         blood = request.form['blood']
110         area = request.form['area']
111         city = request.form['city']
112         district = request.form['district']
113
114         sql = "SELECT * FROM donor WHERE name = ? AND mobile = ? AND email = ? AND age = ? AND gender = ? AND blood = ? AND area = ? AND city = ? AND district = ?"
115         stmt = ibm_db.prepare(conn, sql)
116         ibm_db.bind_param(stmt, 1, name)
117         ibm_db.bind_param(stmt, 2, mobile)
118         ibm_db.bind_param(stmt, 3, email)
119         ibm_db.bind_param(stmt, 4, age)
120         ibm_db.bind_param(stmt, 5, gender)
121         ibm_db.bind_param(stmt, 6, blood)
122         ibm_db.bind_param(stmt, 7, area)
123         ibm_db.bind_param(stmt, 8, city)
124         ibm_db.bind_param(stmt, 9, district)
125         ibm_db.execute(stmt)
126         account = ibm_db.fetch_row()
127         print(account)
128         if account:
129             msg = 'Donor already exists'
130             return render_template("addonor.html", msg=msg)
131         else:
132             insert_sql = "INSERT INTO donor (name, mobile, email, age, gender, blood, area, city, district) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)"
133             prep_stmt = ibm_db.prepare(conn, insert_sql)
134             ibm_db.bind_param(prepare_stmt, 1, name)
135             ibm_db.bind_param(prepare_stmt, 2, mobile)
136             ibm_db.bind_param(prepare_stmt, 3, email)
137             ibm_db.bind_param(prepare_stmt, 4, age)
138             ibm_db.bind_param(prepare_stmt, 5, gender)
139             ibm_db.bind_param(prepare_stmt, 6, blood)
140             ibm_db.bind_param(prepare_stmt, 7, area)
141             ibm_db.bind_param(prepare_stmt, 8, city)
142             ibm_db.bind_param(prepare_stmt, 9, district)
143             ibm_db.execute(prepare_stmt)
144             msg = 'Donor successfully added'
145             return render_template("addonor.html", msg=msg)
146
147
148 ◆ app.py > ...
149
150 @app.route('/request_page')
151 def request_page():
152     msg = ''
153     if request.method == 'POST':
154         drmail = request.form['drmail']
155         hospitalname = request.form['hospitalname']
156         recname = request.form['recname']
157         recmobile = request.form['recmobile']
158         recage = request.form['recage']
159         recgender = request.form['recgender']
160         recbloodgroup = request.form['recbloodgroup']
161         recarea = request.form['recarea']
162         reccity = request.form['reccity']
163         recdistrict = request.form['recdistrict']
164         sql = "SELECT * FROM request WHERE drmail = ? AND hospitalname = ? AND recname = ? AND recmobile = ? AND recage = ? AND recgender = ? AND recbloodgroup = ? AND recarea = ? AND reccity = ? AND recdistrict = ?"
165         stmt = ibm_db.prepare(conn, sql)
166         ibm_db.bind_param(stmt, 1, drmail)
167         ibm_db.bind_param(stmt, 2, hospitalname)
168         ibm_db.bind_param(stmt, 3, recname)
169         ibm_db.bind_param(stmt, 4, recmobile)
170         ibm_db.bind_param(stmt, 5, recage)
171         ibm_db.bind_param(stmt, 6, recgender)
172         ibm_db.bind_param(stmt, 7, recbloodgroup)
173         ibm_db.bind_param(stmt, 8, recarea)
174         ibm_db.bind_param(stmt, 9, reccity)
175         ibm_db.bind_param(stmt, 10, recdistrict)
176         ibm_db.execute(stmt)
177         account = ibm_db.fetch_row()
178         print(account)
179         if account:
180             msg = 'Request already exists'
181             return render_template("request_page.html", msg=msg)
182         else:
183             insert_sql = "INSERT INTO request (drmail, hospitalname, recname, recmobile, recage, recgender, recbloodgroup, recarea, reccity, recdistrict) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
184             prep_stmt = ibm_db.prepare(conn, insert_sql)
185             ibm_db.bind_param(prepare_stmt, 1, drmail)
186             ibm_db.bind_param(prepare_stmt, 2, hospitalname)
187             ibm_db.bind_param(prepare_stmt, 3, recname)
188             ibm_db.bind_param(prepare_stmt, 4, recmobile)
189             ibm_db.bind_param(prepare_stmt, 5, recage)
190             ibm_db.bind_param(prepare_stmt, 6, recgender)
191             ibm_db.bind_param(prepare_stmt, 7, recbloodgroup)
192             ibm_db.bind_param(prepare_stmt, 8, recarea)
193             ibm_db.bind_param(prepare_stmt, 9, reccity)
194             ibm_db.bind_param(prepare_stmt, 10, recdistrict)
195             ibm_db.execute(prepare_stmt)
196             msg = 'Request successfully added'
197             return render_template("request_page.html", msg=msg)
198
199
200 ◆ app.py > ...
201
202 @app.route('/logout')
203 def logout():
204     session.clear()
205     return redirect(url_for("home_page"))
206
207
208 if __name__ == '__main__':
209     app.run(debug=True)
```