

SPRINT DELIVERABLE-3

Team Id	PNT2022PMID49593
Project Name	PLASMA DONOR APPLICATION
Name	SINDHU R
Roll Number	950019104043

Sprint 3- Request sends to the Donor

App.py

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
from flask_mail import Mail, Message
import re
app = Flask(__name__)
mail = Mail(app) # instantiate the mail class

# configuration of mail
app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'sindhu1634@gmail.com'
app.config['MAIL_PASSWORD'] = 'urpndwlpfkwzdkc'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)
app.secret_key='a'
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-
a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;SECURIT
Y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=mpf67194;PWD=iCG3PvOYv
DiHMrq8",'','')
@app.route('/')
def index():
    return render_template('web.html')
@app.route('/login', methods=['GET', 'POST'])
def login():
    global userid
    msg = ''
    if request.method == 'POST' and 'email' in request.form and 'password1' in
request.form:
        email = request.form['email']
        password1 = request.form['password']
        stmt = ibm_db.prepare(conn,'SELECT * FROM donor WHERE email = ? AND
password1 = ?')
        ibm_db.bind_param(stmt,1,email)
```

```

        ibm_db.bind_param(stmt,2,password1)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            session['loggedin'] = True
            session['email'] = account['EMAIL']
            msg = 'Logged in successfully !'
            return render_template('web.html', msg = msg)
        else:
            msg = 'Incorrect username / password !'
            return render_template('login.html', msg = msg)
@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return redirect(url_for('login'))

@app.route('/send_email', methods = ['POST'])
def send():
    if 'blood' in request.form:
        type = request.form['blood']
        stmt = ibm_db.prepare(conn, 'SELECT email FROM donor WHERE blood = ?')
        ibm_db.bind_param(stmt,1,type)
        ibm_db.execute(stmt)
        tb = ibm_db.fetch_tuple(stmt)
        mails = []
        while tb != False:
            mails.append(tb[0])
            tb = ibm_db.fetch_tuple(stmt)

        msg = Message('Blood Request', sender='sindhu1634gmail',
recipients=mails)
        msg.body = "Here the " + request.form['blood'] + " Blood Requested"
        mail.send(msg)
        return render_template('display.html', state="SENT")

    return 'Please Provide Blood in Form'
@app.route('/display', methods =['GET', 'POST'])
def display():
    if 'id' in session:
        if 'blood' in request.form:
            type = request.form['blood']
            stmt = ibm_db.prepare(conn, 'SELECT name1 FROM donor WHERE blood =
?')

            ibm_db.bind_param(stmt,1,type)
            ibm_db.execute(stmt)
            tb = ibm_db.fetch_tuple(stmt)

```

```

        data = []
        while tb != False:
            data.append(tb[0])
            tb = ibm_db.fetch_tuple(stmt)

        return render_template('display.html', data=data,
blood=request.form['blood'], state="NOTSENT")
        return redirect('reclogin')
    return 'Not Authed'

@app.route('/register', methods =['GET', 'POST'])

def register():
    msg = ''
    if request.method == 'POST':
        name1 = request.form['name1']
        blood = request.form['blood']
        email = request.form['email']
        phone = request.form['phone']
        password1 = request.form['password1']
        address1 = request.form['address1']
        gender = request.form['gender']
        age1 = request.form['age1']
        district = request.form['district']
        state1 = request.form['state1']
        illness = request.form['illness']
        sql= "SELECT * FROM donor WHERE name1 = ?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,name1)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        msgg = Message(
            'Hello',
            sender ='sindhu1634gmail.com',
            recipients = [email]
        )
        msgg.body = 'Thankyou for Registering as Donor...! When the plasma
needed we connect with you via email Request'
        mail.send(msgg)

    if account:
        msg = 'Account already exists !'
    elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
        msg = 'Invalid email address !'
    elif not re.match(r'[A-Za-z0-9]+', name1):
        msg = 'name must contain only characters and numbers !'
    elif not name1 or not password1 or not email:
        msg = 'Please fill out the form !'

```

```

        else:
            insert_sql = "INSERT INTO
donor(name1,blood,email,phone,password1,address1,gender,age1,district,state1,i
llness) VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
            stmt = ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(stmt, 1, name1)
            ibm_db.bind_param(stmt, 2, blood)
            ibm_db.bind_param(stmt, 3, email)
            ibm_db.bind_param(stmt, 4, phone)
            ibm_db.bind_param(stmt, 5, password1)
            ibm_db.bind_param(stmt, 6, address1)
            ibm_db.bind_param(stmt, 7, gender)
            ibm_db.bind_param(stmt, 8, age1)
            ibm_db.bind_param(stmt, 9, district)
            ibm_db.bind_param(stmt, 10, state1)
            ibm_db.bind_param(stmt, 11, illness)
            ibm_db.execute(stmt)
            msg = 'You have successfully registered !'
        return render_template('register.html', msg = msg)

@app.route('/reclogin', methods =['GET', 'POST'])
def reclogin():
    global userid
    msg = ''
    if request.method == 'POST' and 'pphone' in request.form :
        pphone = request.form['pphone']
        stmt = ibm_db.prepare(conn,'SELECT * FROM recipient WHERE pphone = ?')
        ibm_db.bind_param(stmt,1, pphone)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            session['id'] = True
            session['id'] = account['ID']
            msg = 'Logged in successfully !'
            return render_template('list.html', msg = msg)
        else:
            msg = 'Incorrect username!'
    return render_template('reclogin.html', msg = msg)

#recipient registration

@app.route('/recipient', methods =['GET', 'POST'])
def recipient():
    msg = ''
    if request.method == 'POST':
        adname = request.form['adname']
        ademail = request.form['ademail']
        pname = request.form['pname']

```

```

pblood = request.form['pblood']
page1 = request.form['page1']
trequest = request.form['trequest']
pphone = request.form['pphone']
paddress = request.form['paddress']
sql= "SELECT * FROM recipients WHERE pname = ?"
stmt = ibm_db.prepare(conn,sql)
ibm_db.bind_param(stmt,1,ademail)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)

if account:
    msg = 'Account already exists !'
elif not re.match(r'^@+@[^@]+\.[^@]+', ademail):
    msg = 'Invalid email address !'
elif not re.match(r'[A-Za-z0-9]+', pname):
    msg = 'name must contain only characters and numbers !'
elif not adname or not ademail:
    msg = 'Please fill out the form !'
else:
    insert_sql = "INSERT INTO
recipient(adname,ademail,pname,pblood,page1,trequest,pphone,paddress)
VALUES(?, ?, ?, ?, ?, ?, ?, ?)"
    stmt = ibm_db.prepare(conn,insert_sql)
    ibm_db.bind_param(stmt, 1, adname)
    ibm_db.bind_param(stmt, 2, ademail)
    ibm_db.bind_param(stmt, 3, pname)
    ibm_db.bind_param(stmt, 4, pblood)
    ibm_db.bind_param(stmt, 5, page1)
    ibm_db.bind_param(stmt, 6, trequest)
    ibm_db.bind_param(stmt, 7, pphone)
    ibm_db.bind_param(stmt, 8, paddress)
    ibm_db.execute(stmt)
    msg = 'You have successfully registered !'
    return render_template('list.html')
return render_template('recregister.html', msg = msg)
if __name__ == '__main__':
    app.run(debug= True)

```

List.html

```

<!DOCTYPE html>
<head>
    <meta charset="utf-8" />
    <title>Registration Form</title>
    <meta name="viewport" content="width=device-width,
        initial-scale=1.0"/>

```

```

        <link rel="stylesheet" href="{{ url_for('static', filename='list.css')
    }}">

</head>
<body>
    <form action="{{ url_for('display') }}" method="post">

        <label for="blood">Blood Group</label><br>
        <input type="text" id="blood" name="blood"><br>
        <div class="btn">
            <button>Donors list</button>
        </div>
    </form>
</body>

```

Display.html

```

<div>
    <div>Name</div>

    {% for val in data %}
    <div>{{val}}</div>
    {% endfor %}

    {% if state == "SENT" %}
    <div>Email Sent Successfully </div>
    {% else %}
    <form action="/send_email" method="post">
        <input hidden name="blood" value="{{blood}}"/>
        <button type="submit">Send</button>
    </form>
    {% endif %}
    </tbody>
</table>
</div>

```

List.css

```
.btn{  
  margin-bottom: auto;  
  padding-top: 45px;  
  text-align: center;  
}
```

OUTPUT:







