

## SPRINT DELIVERABLE-2

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

Sprint-2: Donor confirmation mail

Recipient registration

**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'] = 'urpndwlpsfkwzdkc'
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 'password' in
request.form:
        email = request.form['email']
        password = request.form['password']
```

```

        stmt = ibm_db.prepare(conn, 'SELECT * FROM donors WHERE email = ? AND
password = ?')
        ibm_db.bind_param(stmt,1,email)
        ibm_db.bind_param(stmt,2,password)
        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('/register', methods = ['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST':
        name = request.form['name']
        bloodgroup = request.form['blood']
        email = request.form['email']
        phonenumber = request.form['phone']
        password = request.form['password']
        address = request.form['address']
        gender = request.form['gender']
        age = request.form['age']
        district = request.form['district']
        state = request.form['state']
        illness = request.form['illness']
        sql= "SELECT * FROM donors WHERE name = ?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,name)
        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]+', name):
    msg = 'name must contain only characters and numbers !'
elif not name or not password or not email:
    msg = 'Please fill out the form !'
else:
    insert_sql = "INSERT INTO donors VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"

    stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(stmt, 1, name)
    ibm_db.bind_param(stmt, 2, bloodgroup)
    ibm_db.bind_param(stmt, 3, email)
    ibm_db.bind_param(stmt, 4, phonenumber)
    ibm_db.bind_param(stmt, 5, password)
    ibm_db.bind_param(stmt, 6, address)
    ibm_db.bind_param(stmt, 7, gender)
    ibm_db.bind_param(stmt, 8, age)
    ibm_db.bind_param(stmt, 9, district)
    ibm_db.bind_param(stmt, 10, state)
    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('/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']
        page = request.form['page']
        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, pname)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        # msgg = Message('Hello', sender = 'sindhu1634gmail.com', recipients =
[ademail])

```

```

# 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'^@+@[^@]+\.[^@]+', 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 recipients VALUES(?, ?, ?, ?, ?, ?, ?,
?)"

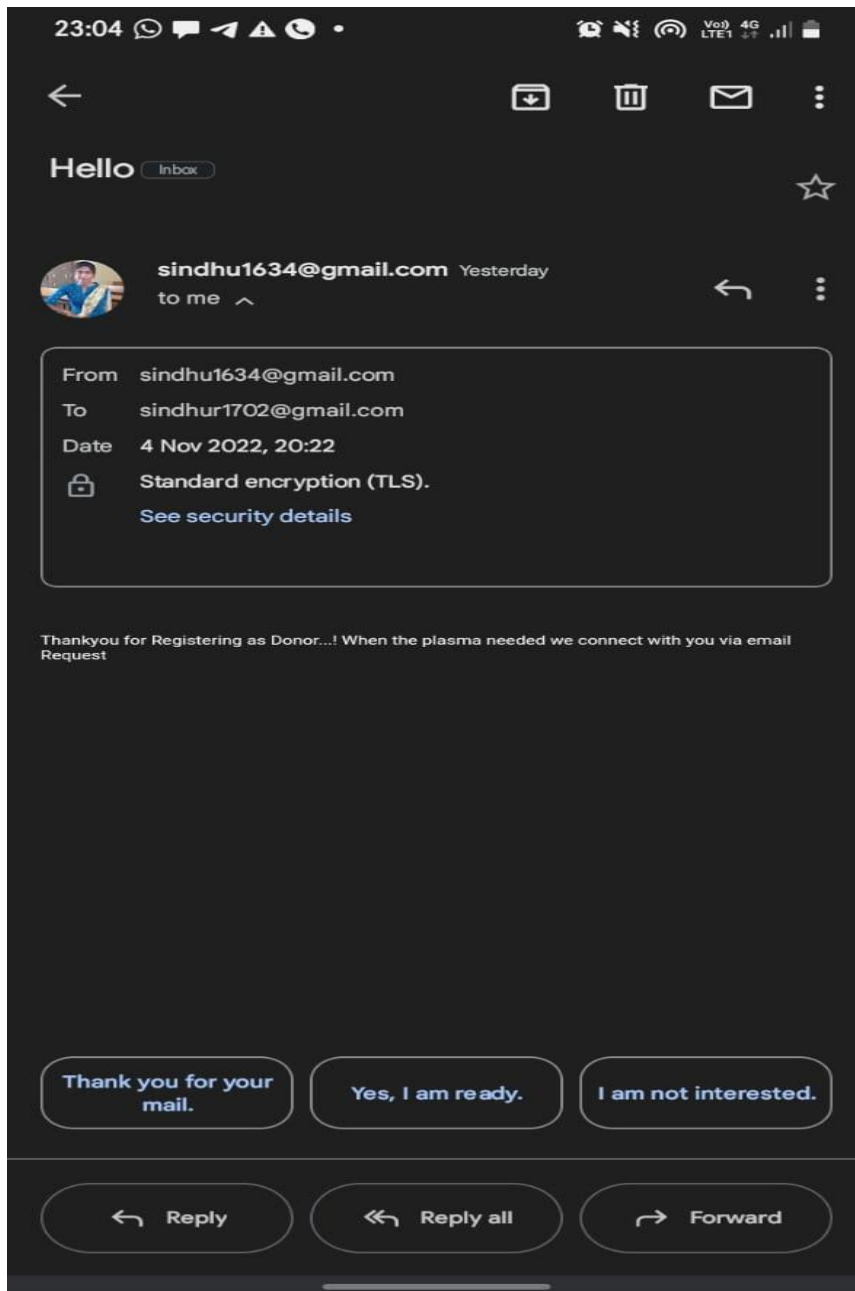
    stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(stmt, 1, adname)
    ibm_db.bind_param(stmt, 2, ademail)
    ibm_db.bind_param(stmt, 3, pname)
    ibm_db.bind_param(stmt, 4, pblood)
    ibm_db.bind_param(stmt, 5, page)
    ibm_db.bind_param(stmt, 6, trequest)
    ibm_db.bind_param(stmt, 7, pphone)
    ibm_db.bind_param(stmt, 8, paddress)
    ibm_db.execute(stmt)
    msg = 'You have successfully registered !'
return render_template('reregister.html', msg = msg)
if __name__ == '__main__':
    app.run(debug= True)

```

## OUTPUT:

The screenshot displays a web browser window with the address bar showing '127.0.0.1:5000/register'. The main content is a 'Donor Registration' form. The form has a dark blue background with white text and input fields. The fields and their values are as follows:

Field Label	Value
Full Name	sindhu
Blood Group	A+
Email	sindhur1702@gmail.com
Password	...
Phone Number	1234567890



templates/Recipient.html:

```
<!DOCTYPE html>
<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='rr.css')
  } }">
```

```
</head>
<body>

<form action="{ { url_for('recipient') } }" method="post">
  <div class="msg"> { { msg } }</div>
  <h3>Recipient Registration</h3>

  <label for="adname">ADMIN ORGANISATION </label>
  <input type="text" name="adname" placeholder="Enter your Organisation
name..." id="adname">

  <label for="ademail">ADMIN EMAIL</label>
  <input type="text" name="ademail" placeholder="Enter your email"
id="ademail">

  <label for="pname">PATIENT'S NAME </label>
  <input type="text" name="pname" placeholder="Enter Patient's name"
id="pname">

  <label for="pblood">PATIENT'S BLOOD GROUP</label>
  <input type="text" name="pblood" placeholder="Enter Patient's blood Group"
id="pblood">

  <label for="page">PATIENT'S AGE</label>
  <input type="number" name="page" placeholder="Enter age" id="page">

  <label for="trequest">TYPES OF REQUEST</label>
  <input type="text" name="trequest" placeholder="Enter whether
Emergency/routine/urgent" id="trequest">

  <label for="pphone">PHONE NUMBER</label>
  <input type="text" name="pphone" placeholder="Enter Phone Number"
id="pphone">

  <label for="paddress">ADDRESS</label>
  <textarea type="text" name="paddress" placeholder="Enter Address"
id="paddress">
</textarea>

  <a><button href="/recipient">SUBMIT</button></a>

</form>
</body>
</html>
```

## Static/rr.css:

```
*{
  padding: 15px;
  margin: 0;
  box-sizing: border-box;
  font-family: sans-serif;
}

body{
  display: flex;

  justify-content: center;
  align-items: center;
  background-color: #031237;
}

.msg{
  text-align: center;
  font-size: 20px;
  color: #e5e5e5;
}

.container{
  display: flexbox;
  flex-direction: row;
  flex-wrap: wrap;
  position: relative;
  width: 100%;
  max-width: 850px;
  padding: 30px;
  margin: 0 28px;
}

form{
  margin: -75px;

  background-color: rgba(255,255,255,0.13);
  position: absolute;
  transform: translate(-50%,-50%);
  top: 143%;
  left: 55%;
  border-radius: 10px;
  backdrop-filter: blur(10px);
  border: 2px solid rgba(255,255,255,0.1);
  box-shadow: 0 0 40px rgba(8,7,16,0.6);
  padding: 50px 76px;
```

```
}  
form *{  
    font-family: 'Poppins', sans-serif;  
    color: #ffffff;  
    letter-spacing: 0.5px;  
    outline: none;  
    border: none;  
}  
form h3{  
    font-size: 32px;  
    font-weight: 500;  
    line-height: 42px;  
    text-align: center;  
}  
  
label{  
    display: block;  
    margin-top: 30px;  
    font-size: 16px;  
    font-weight: 500;  
}  
  
input{  
    display: block;  
    height: 50px;  
    width: 100%;  
    background-color: rgba(255,255,255,0.07);  
    border-radius: 3px;  
    padding: 0 10px;  
    margin-top: 8px;  
    font-size: 14px;  
    font-weight: 300;  
}  
  
textarea{  
    display: block;  
    height: 150px;  
    width: 100%;  
    background-color: rgba(255,255,255,0.07);  
    border-radius: 3px;  
    padding: 0 10px;  
    margin-top: 8px;  
    font-size: 14px;  
    font-weight: 300;  
}  
::placeholder{  
    color: #e5e5e5;
```



```
}  
button{  
  margin: 100px -2px;  
  width: 100%;  
  background-color: #ffffff;  
  color: #080710;  
  padding: 15px 0;  
  font-size: 18px;  
  font-weight: 600;  
  border-radius: 10px;  
  cursor: pointer;  
}  
  
@media(max-width: 600px){  
  .container{  
    min-width: 280px;  
  }  
  
  .user-input-box{  
    margin-bottom: 12px;  
    width: 100%;  
  }  
  
  .user-input-box:nth-child(2n){  
    justify-content: space-between;  
  }  
  
  .gender-category{  
    display: flex;  
    justify-content: space-between;  
    width: 100%;  
  }  
  
  .main-user-info{  
    max-height: 380px;  
    overflow: auto;  
  }  
  
  .main-user-info::-webkit-scrollbar{  
    width: 0;  
  }  
}
```

## Web.html (index page):

```
<!DOCTYPE html>
<html lang="en">

<head>
  <title>Webpage Design</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='res.css') }}">
</head>

<body>

  <div class="main">
    <div class="navbar">
      <CENTER>
        <div class="icon">
          <h2 class="logo">PLASMA WORLD!</h2>

        </div>
      </CENTER>

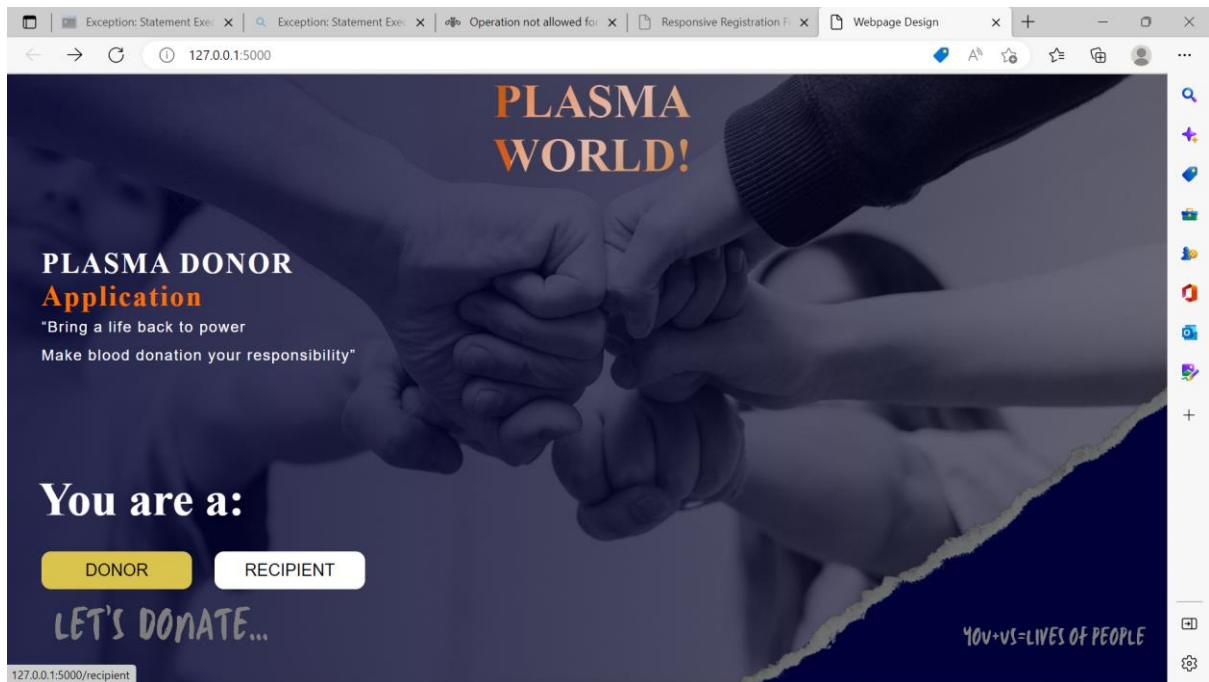
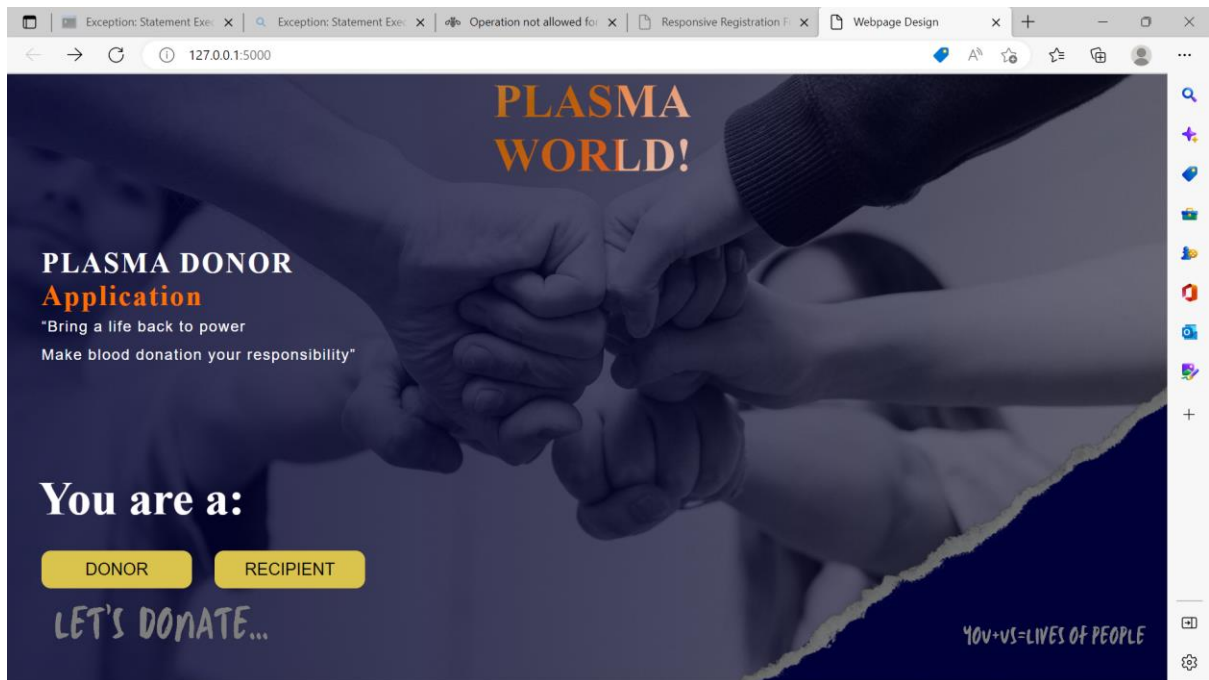
    </div>
    <div class="content">
      <h1>PLASMA DONOR <br><span>Application</span> <br></h1>
      <p class="par"> “Bring a life back to power
        <br>Make blood donation your responsibility”
        <div class="you"><br><b> You are a:</b></div>

      </p>
      <div>
        <button class="cn"><a href="/login">DONOR</a></button>
        <button class="cn1"><a
href="/recipient">RECIPIENT</a></button>
      </div>

    </div>
  </div>
</div>
</div>
</div>
  <script src="https://unpkg.com/ionicons@5.4.0/dist/ionicons.js"></script>
</body>

</html>
```

## OUTPUT:



Exception: Statement Exe: X Exception: Statement Exe: X Operation not allowed for: X Responsive Registration F: X Registration Form X

127.0.0.1:5000/recipient

## Recipient Registration

ADMIN ORGANISATION

VM hospital

ADMIN EMAIL

sindhur1702@gmail.com

PATIENT'S NAME

rahul

PATIENT'S BLOOD GROUP

Exception: Statement Exe: X Exception: Statement Exe: X Operation not allowed for: X Responsive Registration F: X Registration Form X

127.0.0.1:5000/recipient

PATIENT'S BLOOD GROUP

A+

PATIENT'S AGE

34

TYPES OF REQUEST

urgent

PHONE NUMBER

1234567890

ADDRESS

tvf

Exception: Statement Exe: X Exception: Statement Exe: X Operation not allowed for: X Responsive Registration F: X Registration Form X

127.0.0.1:5000/recipient

PHONE NUMBER

1234567890

ADDRESS

tv

SUBMIT

Exception: Statement Exe: X Exception: Statement Exe: X Operation not allowed for: X Responsive Registration F: X Registration Form X

127.0.0.1:5000/recipient

You have successfully registered !

## Recipient Registration

ADMIN ORGANISATION

Enter your Organisation name...

ADMIN EMAIL

Enter your email

PATIENT'S NAME

Enter Patient's name

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTSSequencesApplication objects

MPF67194.RECIPIENTS

Back

Export to CSV

ADNAME	ADMAIL	PNAME	PBLOOD	PAGE	TREQUEST	PPHONE	PADDRESS
VM hospital	sindhur1702@gmail.com	rahul	A+	34	urgent	1234567890	tvf
asd	sindhu1634@gmail.com	cvbn	A+	67	urgent	9384243611	sss

WhatsApp Image...jpeg

Show all