

SPRINT-2

Date	5 October 2022
Team ID	PNT2022TMID49558
Project name	Plasma donor web application

CODE:

Reg.html:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Recipient Registration page</title>
```

```
    <link rel="stylesheet" href="style.css">
```

```
  </head>
```

```
    <style>
```

```
body{
```

```
  background-image: url(https://images.pexels.com/photos/6074921/pexels-photo-6074921.jpeg?cs=srgb&dl=pexels-alena-shekhovtcova-6074921.jpg&fm=jpg&_gl=1*1jzc4xw*_ga*MTkxMDQwNjUxNS4xNjY3Mjk3NDk1*_ga_8JE65Q40S6*MTY2NzQ4NDI3MS4yLjEuMTY2NzQ4NTA4NC4wLjAuMA..);
```

```
background-repeat: no-repeat; }
```

```
</style>
```

```
<body>
```

```
<div class="wrap">
```

```
<h2 class="head">RECIPIENT REGISTER FORM</h2>
```

```
<div class="sub-main">
```

```
  <form action="#" method="post">
```

```
<input placeholder="Username" name="Username" type="text" required="">
<br> <input placeholder="Age" name="Age" type="text" required="">
<br><input placeholder="Gender" name="Gender" type="text" required="">
<br><input placeholder="Blood group" name="Blood group" type="text" required="">
<br>
<input placeholder="Phone Number" name="Number" type="text" required="">
<br><input placeholder="Address" name="Address" type="text" required="">
<br><input placeholder="Email" name="mail" type="text" required="">
<br><input placeholder="Password" name="Password" type="password" required="">
<br><input placeholder="Confirm Password" name="Password" type="password" required="">
<br><input placeholder="Height" name="Height" type="text" required="">
<br><input placeholder="Weight" name="Weight" type="text" required="">
<br><br>
<button><input type="submit" value="Register"></form></button>
</form>
</div>
<div class="clear"></div>
</div></body>
</html>
```

Login.html:

```
<Html>
<head>
<title>Login form</title>
```

```
<style>
body {
  background-image: url(login.jpg);
  background-repeat: no-repeat;
  background-position: right;}
.main {
  background-color: lightpink;
  position: absolute;
  top: 50%;
  right: 60%;
  transform: translate(-50%, -50%);
  background: #0aeff0;
  box-shadow: 0 0 5px rgba(0 0 0 /.5), 0 0 25px rgba(0 0 0 /.5), 0 0 50px rgba(0 0 0 /.5), 0
0 100px #03e9f4
width: 400px;
padding: 60px;
background: rgba(0 0 0 /.9);
box-sizing: border-box;
border-radius: 10px;
} .login{
  margin-left: auto;
  margin-right: auto;
}
</style></head>
<h1 class="word" style="padding:95px";> DONOR/RECIPIENT LOGIN</h1>
```

```
<body>

<br>

<div class="main">

    <div class="login">

        <form action="{{ url_for('login') }}" method="post">

            <br>

            <center>

                <label style="color:#fff;">Username:</label>

                <input type="text" name="Username" size="15" /> <br> <br>

                <label style="color:#fff;"> Password: </label>

                <input type="text" name="Password" size="15" />

                <br> <br> <br>

                <button>

                    <input type="submit" value="LOGIN" />

                </button>

            </center>

        </form>

    </div>

</div>

</body>

</html>
```

App.py:

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
```

```

app = Flask(__name__)
app.secret_key = 'a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=125f9f61-9715-46f9-9399-
c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30426;SECURITY=S
SL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=thy82883;PWD=DtTOZYv3UCHUf
P15", "", "")
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/login', methods =['GET', 'POST'])
def login():
    global userid
    msg = ""
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
        username = request.form['username']
        password = request.form['password']
        stmt = ibm_db.prepare(conn,'SELECT * FROM donor WHERE username = ? AND
password = ?')
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            session['loggedin'] = True
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'
            return render_template('index.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':
        username = request.form['username']
        age = request.form['age']
        bloodgroup = request.form['bloodgroup']
        lastdonatedate= request.form['lastdonatedate']
        gender = request.form['gender']
        phone= request.form['phone']
        address= request.form['address']
        email = request.form['email']
        password = request.form['password']
        confirmpassword = request.form['confirmpassword']
        height= request.form['height']
        weight = request.form['weight']
        sql = "SELECT * FROM donor WHERE username = ? "
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        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]+', username):
            msg = 'Username must contain only characters and numbers !'
        elif not username or not password or not confirmpassword:
            msg = 'Please fill out the form !'
        else:
            insert_sql = "INSERT INTO donor VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
            stmt = ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(stmt, 1, username )
            ibm_db.bind_param(stmt, 2, age)
            ibm_db.bind_param(stmt, 3, bloodgroup)
            ibm_db.bind_param(stmt, 4, lastdonatedate )
            ibm_db.bind_param(stmt, 5, gender )
            ibm_db.bind_param(stmt, 6, phone )
            ibm_db.bind_param(stmt, 7, address )
            ibm_db.bind_param(stmt, 8, email )

```

```

        ibm_db.bind_param(stmt, 9, password )
        ibm_db.bind_param(stmt, 10, confirmpassword )
        ibm_db.bind_param(stmt, 11, height)
        ibm_db.bind_param(stmt, 12, weight )
        ibm_db.execute(stmt)

        msg = 'You have successfully registered !'
    elif request.method == 'POST':
        msg = 'Please fill out the form !'

    return render_template('donor.html', msg = msg)
@app.route('/registers', methods =['GET', 'POST'])
def registers():
    msg = "
    if request.method == 'POST':
        username = request.form['username']
        age = request.form['age']
        bloodgroup = request.form['bloodgroup']
        gender = request.form['gender']
        phone= request.form['phone']
        address= request.form['address']
        email = request.form['email']
        password = request.form['password']
        confirmpassword = request.form['confirmpassword']
        height= request.form['height']
        weight = request.form['weight']
        sql = "SELECT * FROM recipient WHERE username = ? "
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        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]+', username):
            msg = 'Username must contain only characters and numbers !'
        elif not username or not password or not confirmpassword:
            msg = 'Please fill out the form !'
    
```

```

else:
    insert_sql = "INSERT INTO recipient VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
    stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(stmt, 1, username )
    ibm_db.bind_param(stmt, 2, age)
    ibm_db.bind_param(stmt, 3, bloodgroup)
    ibm_db.bind_param(stmt, 4, gender )
    ibm_db.bind_param(stmt, 5, phone )
    ibm_db.bind_param(stmt, 6, address )
    ibm_db.bind_param(stmt, 7, email )
    ibm_db.bind_param(stmt, 8, password )
    ibm_db.bind_param(stmt, 9, confirmpassword )
    ibm_db.bind_param(stmt, 10, height)
    ibm_db.bind_param(stmt, 11, weight )
    ibm_db.execute(stmt)

    msg = 'You have successfully registered !'
elif request.method == 'POST':
    msg = 'Please fill out the form !'

    return render_template('recipient.html', msg = msg)
if __name__ == '__main__':
    app.run(debug = True)
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re

app = Flask(__name__)
app.secret_key = 'a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30426;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=thy82883;PWD=DtTOZYv3UCHUfP15", "", "")
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/login', methods =['GET', 'POST'])
def login():
    global userid

```



```

msg = "
if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
    username = request.form['username']
    password = request.form['password']
    stmt = ibm_db.prepare(conn,'SELECT * FROM donor WHERE username = ? AND
password = ?')
    ibm_db.bind_param(stmt,1,username)
    ibm_db.bind_param(stmt,2,password)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    if account:
        session['loggedin'] = True
        session['username'] = account['USERNAME']
        msg = 'Logged in successfully !'
        return render_template('index.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':
        username = request.form['username']
        age = request.form['age']
        bloodgroup = request.form['bloodgroup']
        lastdonatedate= request.form['lastdonatedate']
        gender = request.form['gender']
        phone= request.form['phone']
        address= request.form['address']
        email = request.form['email']
        password = request.form['password']
        confirmpassword = request.form['confirmpassword']

```

```

height= request.form['height']
weight = request.form['weight']
sql = "SELECT * FROM donor WHERE username = ? "
stmt = ibm_db.prepare(conn,sql)
ibm_db.bind_param(stmt,1,username)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
print(account)
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]+', username):
    msg = 'Username must contain only characters and numbers !'
elif not username or not password or not username:
    msg = 'Please fill out the form !'
else:
    insert_sql = "INSERT INTO donor VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
    stmt = ibm_db.prepare(conn,insert_sql)
    ibm_db.bind_param(stmt, 1, username )
    ibm_db.bind_param(stmt, 2, age)
    ibm_db.bind_param(stmt, 3, bloodgroup)
    ibm_db.bind_param(stmt, 4, lastdonatedate )
    ibm_db.bind_param(stmt, 5, gender )
    ibm_db.bind_param(stmt, 6, phone )
    ibm_db.bind_param(stmt, 7, address )
    ibm_db.bind_param(stmt, 8, email )
    ibm_db.bind_param(stmt, 9, password )
    ibm_db.bind_param(stmt, 10, confirmpassword )
    ibm_db.bind_param(stmt, 11, height)
    ibm_db.bind_param(stmt, 12, weight )
    ibm_db.execute(stmt)

    msg = 'You have successfully registered !'
elif request.method == 'POST':
    msg = 'Please fill out the form !'

    return render_template('donor.html', msg = msg)
@app.route('/registers', methods=['GET', 'POST'])
def registers():

```

```

msg = "
if request.method == 'POST':
    username = request.form['username']
    age = request.form['age']
    bloodgroup = request.form['bloodgroup']
    gender = request.form['gender']
    phone= request.form['phone']
    address= request.form['address']
    email = request.form['email']
    password = request.form['password']
    confirmpassword = request.form['confirmpassword']
    height= request.form['height']
    weight = request.form['weight']
    sql = "SELECT * FROM recipient WHERE username = ? "
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print(account)
    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]+', username):
        msg = 'Username must contain only characters and numbers !'
    elif not username or not password or not confirmpassword:
        msg = 'Please fill out the form !'
    else:
        insert_sql = "INSERT INTO recipient VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
        stmt = ibm_db.prepare(conn,insert_sql)
        ibm_db.bind_param(stmt, 1, username )
        ibm_db.bind_param(stmt, 2, age)
        ibm_db.bind_param(stmt, 3, bloodgroup)
        ibm_db.bind_param(stmt, 4, gender )
        ibm_db.bind_param(stmt, 5, phone )
        ibm_db.bind_param(stmt, 6, address )
        ibm_db.bind_param(stmt, 7, email )
        ibm_db.bind_param(stmt, 8, password )
        ibm_db.bind_param(stmt, 9, confirmpassword )
        ibm_db.bind_param(stmt, 10, height)

```

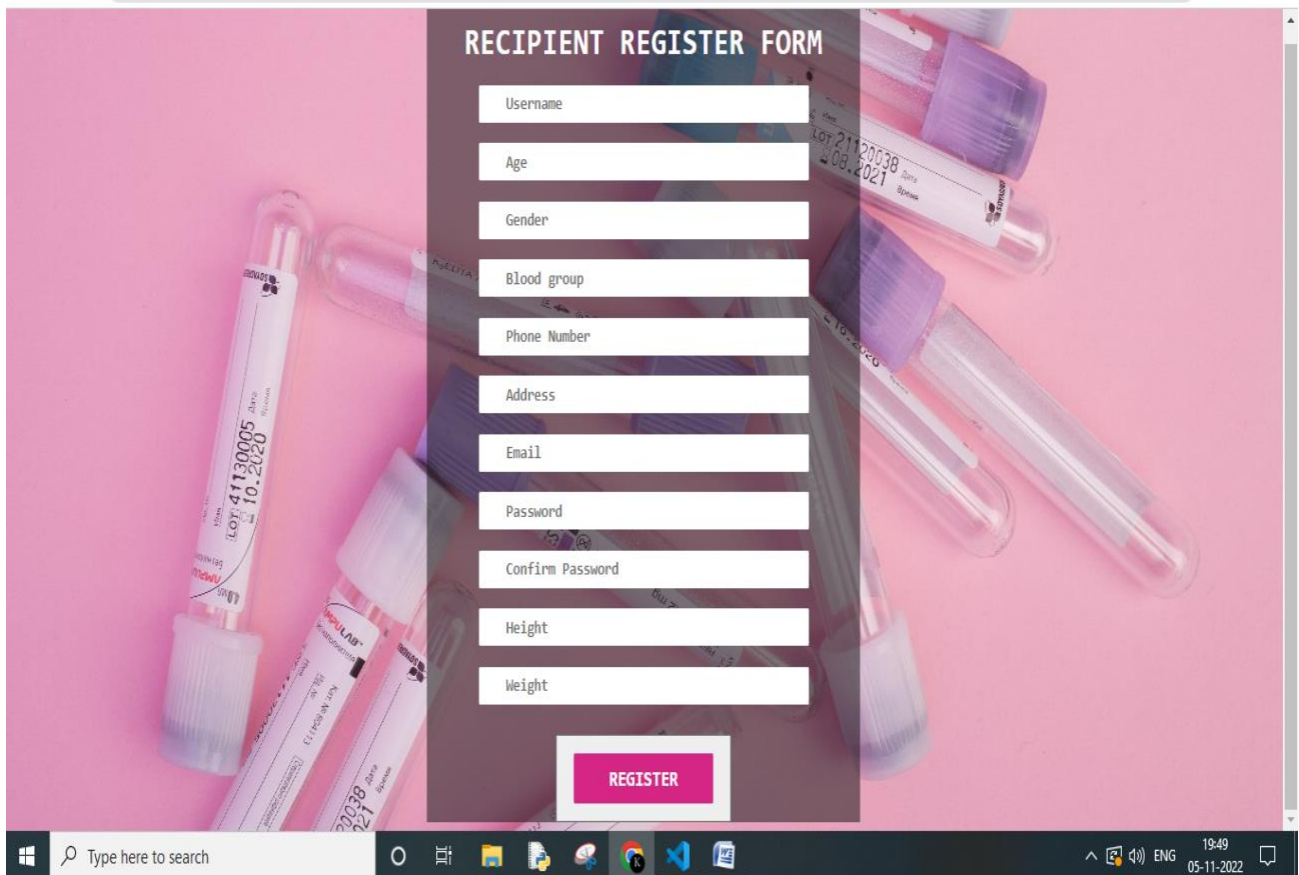
```
ibm_db.bind_param(stmt, 11, weight )
ibm_db.execute(stmt)

msg = 'You have successfully registered !'
elif request.method == 'POST':
    msg = 'Please fill out the form !'

return render_template('recipient.html', msg = msg)
if __name__ == '__main__':
    app.run(debug = True)
```

OUTPUT

Recipient registration page:



RECIPIENT REGISTER FORM

Username

Age

Gender

Blood group

Phone Number

Address

Email

Password

Confirm Password

Height


Weight

REGISTER

RECIPIENT REGISTER FORM

Recipient login page:

DONOR/RECIPIENT LOGIN



DATEBASE for Recipient:

IBM Db2 on Cloud

Load DataLoad History**Tables**ViewsIndexesAliasesMQTsSequencesApplication objects

THY82883.RECIPIENT

Back

Export to CSV

USERNAME	AGE	BLOODGROUP	GENDER	PHONE	ADDRESS	EMAIL	PASSWORD	CONFIRMPASSWORD	HEIGHT	WEIGHT
evanj	20	B+	female	9466574301	852D1,Thavazhakuzhivilai,kottadai,kuzhithurai,kanyakumari,629163	anur@gmail.com	123	123	123	23
kamini	18	B+	female	9080706050	north car street	kamini@gmail.com	kavi	kavi	120	30

Sprint-2 EMAIL ALERT CONFIRMATION TO DONOR

Mail.py:

```
# importing libraries

from flask import Flask

from flask_mail import Mail, Message

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'] = 'karpagamboothanathan@gmail.com'

app.config['MAIL_PASSWORD'] = 'nmydiidhksvvojid'

app.config['MAIL_USE_TLS'] = False

app.config['MAIL_USE_SSL'] = True

mail = Mail(app)

# message object mapped to a particular URL '/'

@app.route("/")

def index():

    msg = Message(

        'you have registered sucessfully',

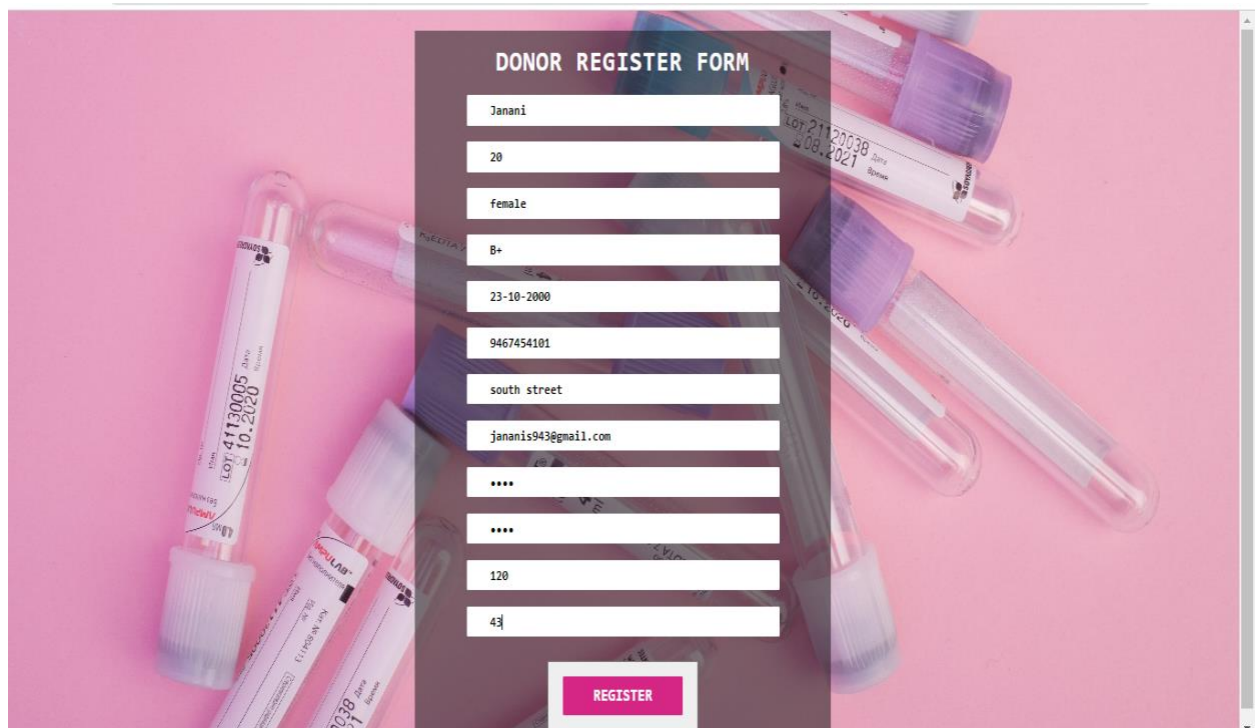
        sender='karpagamboothanathan@gmail.com',

        recipients = ['jananis943@gmail.com']

    )
```

```
msg.body = 'welcome !! to plasma donor web application'  
mail.send(msg)  
return'Sent'  
if __name__ == '__main__':  
    app.run(debug = True)
```

DONOR Registration page:



The image shows a web application interface for donor registration. The background is a pink surface with several blood test tubes. Overlaid on this is a dark grey rectangular form titled "DONOR REGISTER FORM" in white capital letters. The form contains a series of white input fields for the following information: Name (filled with "Janani"), Age (filled with "20"), Gender (filled with "female"), Blood Type (filled with "B+"), Date of Birth (filled with "23-10-2000"), Phone Number (filled with "9467454101"), Address (filled with "south street"), Email (filled with "jananis943@gmail.com"), Password (indicated by four dots), Confirm Password (indicated by four dots), Height (filled with "120"), and Weight (filled with "43"). At the bottom of the form is a pink button with the word "REGISTER" in white capital letters.

Database donor table:

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

THY82883.DONOR

Back

Export to CSV

USERNAME	AGE	BLOODGROUP	LASTDONATEDATE	GENDER	PHONE	ADDRESS	EMAIL	PASSWORD	CONFIRMPASSWORD
Anu	20	B+	23-10-2000	female	9467454101	852D1,Thavazhakuzhi vilai ,kottadai,kuzhithurai,kan yakumari,629163	anur231001@gmail.com	123	123
Janani	20	B+	23-10-2000	female	9467454101	south street	jananis943@gmail.com	kali	kali
ammu	20	B+	23-10-2000	female	9467454101	852D1,Thavazhakuzhi vilai ,kottadai,kuzhithurai,kan yakumari,629163	anur@gmail.com	123	123

Mail Alert:

you have registered sucessfully

mail.google.com/mail/u/0/#inbox/FMfcgzGgRQFJSzhDbxFJnhgKvSDBFMT

Compose

Inbox 1,901

Starred

Snoozed

Sent

Drafts

More

Labels

Search mail

you have registered sucessfully

1 of 2,310

you have registered sucessfully

karpagamboothanathan@gmail.com

welcome !! to plasma donor web application

21:06 (13 minutes ago)

karpagamboothanathan@gmail.com

to me

welcome !! to plasma donor web application

21:12 (7 minutes ago)

karpagamboothanathan@gmail.com

to me

welcome !! to plasma donor web application

21:12 (7 minutes ago)

Reply Forward