

```

from random import randint

from flask import Flask,render_template,request,url_for,redirect

from flask_mail import *

import ibm_db


dictionaryForEmailDonor={}

def printDonorData(conn):

    sql = "SELECT * FROM donorregistration"

    out = ibm_db.exec_immediate(conn, sql)

    document = ibm_db.fetch_assoc(out)

    while document != False:

        dictionaryForEmailDonor.update({document['EMAIL']:document['PASSWORD']})

        document = ibm_db.fetch_assoc(out)


dictionaryForEmailRecipient={}

def printRecipientData(conn):

    sql = "SELECT * FROM recipientregistration"

    out = ibm_db.exec_immediate(conn, sql)

    document = ibm_db.fetch_assoc(out)

    while document != False:

        dictionaryForEmailRecipient.update({document['EMAIL']:document['PASSWORD']})

        document = ibm_db.fetch_assoc(out)


dictionaryForEmailIncharge={}

def printInchargeData(conn):

    sql = "SELECT * FROM inchargeregistration"

    out = ibm_db.exec_immediate(conn, sql)

    document = ibm_db.fetch_assoc(out)

    while document != False:

        dictionaryForEmailIncharge.update({document['EMAIL']:document['PASSWORD']})

        document = ibm_db.fetch_assoc(out)

```

```

def
insertDonorData(conn,username,firstname,lastname,age,gender,blood,email,phonenumber,passwo
rd,district,pincode):

    sql="INSERT INTO
donorregistration(username,firstname,lastname,age,gender,blood,email,phonenumber,password,di
strict,pincode) VALUES
('{}','{}','{}',{},{},'{}','{}','{}','{}','{}').format(username,firstname,lastname,age,gender,blood,email,ph
onenumber,password,district,pincode)

    out = ibm_db.exec_immediate(conn,sql)

    print('Number of affected rows : ',ibm_db.num_rows(out),"\\n")

```

```

def
insertRecipientData(conn,username,firstname,lastname,age,gender,blood,email,phonenumber,pass
word,district,pincode):

    sql="INSERT INTO
recipientregistration(username,firstname,lastname,age,gender,blood,email,phonenumber,password
,district,pincode) VALUES
('{}','{}','{}',{},{},'{}','{}','{}','{}','{}').format(username,firstname,lastname,age,gender,blood,email,ph
onenumber,password,district,pincode)

    out = ibm_db.exec_immediate(conn,sql)

    print('Number of affected rows : ',ibm_db.num_rows(out),"\\n")

```

```

def insertInchargeData(conn,hospitalname,email,phonenumber,password,district,address,pincode):

    sql="INSERT INTO
inchargeregistration(hospitalname,email,phonenumber,password,district,address,pincode) VALUES
('{}','{}',{},{},'{}','{}','{}').format(hospitalname,email,phonenumber,password,district,address,pincode)

    out = ibm_db.exec_immediate(conn,sql)

    print('Number of affected rows : ',ibm_db.num_rows(out),"\\n")

```

```

def deleteDonorData(conn,email):

    sql = "DELETE FROM donorregistration WHERE email={}".format(email)

    out = ibm_db.exec_immediate(conn, sql)

    print('Number of affected rows : ', ibm_db.num_rows(out), "\\n")

```

```

def deleteRecipientData(conn,email):

```

```

sql = "DELETE FROM recipientregistration WHERE email={}".format(email)

out = ibm_db.exec_immediate(conn, sql)

print('Number of affected rows : ', ibm_db.num_rows(out), "\n")

```

```
def deleteInchargeData(conn,email):
```

```

    sql = "DELETE FROM inchargeregistration WHERE email={}".format(email)

    out = ibm_db.exec_immediate(conn, sql)

    print('Number of affected rows : ', ibm_db.num_rows(out), "\n")

```

```
try:
```

```

    conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=0c77d6f2-5da9-48a9-81f8-
86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31198;SECURITY=SSL;SSLS
erverCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=bjn03696;PWD=ef96tLjX2VjzaCPX;
", "", "")

    print("Db connected")

```

```
except:
```

```
    print("Error")
```

```
app=Flask(__name__)
```

```
app.config['MAIL_SERVER']='smtp.gmail.com'
```

```
app.config['MAIL_PORT']=465
```

```
app.config['MAIL_USERNAME'] = 'gowrishankarj1110@gmail.com'
```

```
app.config['MAIL_PASSWORD'] = 'bfylpfroomoghhxi'
```

```
app.config['MAIL_USE_TLS'] = False
```

```
app.config['MAIL_USE_SSL'] = True
```

```
mail = Mail(app)
```

```
otp = randint(000000,999999)
```

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

```
def index():
```

```
    return render_template("indexpage.html")
```

```
@app.route("/inchargelogin",methods=['POST','GET'])
```

```
def inchargelogin():
```

```
    if request.method=="POST":
```

```
        printInchargeData(conn)
```

```
        text=request.form['text']
```

```
        password=request.form['password']
```

```
        try:
```

```
            if dictionaryForEmailIncharge[text] == password:
```

```
                return redirect(url_for('inchargehome'))
```

```
        except:
```

```
            return "invalid email or password"
```

```
    return render_template("inchargelogin.html")
```

```
@app.route("/inchargeregister",methods=['POST','GET'])
```

```
def inchargeregister():
```

```
    if request.method=="POST":
```

```
        hospitalname = request.form['hospitalname']
```

```
        email = request.form['email']
```

```
        phonenumber=request.form['phonenumber']
```

```
        district=request.form['district']
```

```
        pincode=request.form['pincode']
```

```
        address=request.form['address']
```

```
        password = request.form['password']
```

```
        insertInchargeData(conn,hospitalname,email,phonenumber,password,district,address,pincode)
```

```
        return redirect(url_for('inchargeauthentication',email=email))
```

```
    return render_template('inchargeregister.html')
```

```
@app.route("/recipientlogin",methods=['POST','GET'])
```

```
def recipientlogin():
```

```
    if request.method=="POST":
```

```

printRecipientData(conn)

text=request.form['text']

password=request.form['password']

try:

    if dictionaryForEmailRecipient[text] == password:

        return redirect(url_for('recipienthome'))

except:

    return "invalid email or password"

return render_template("recipientlogin.html")

```

```

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

```

```

def recipientregister():

```

```

    if request.method=="POST":

        username = request.form['username']

        firstname=request.form['firstname']

        lastname=request.form['lastname']

        gender=request.form['gender']

        age=request.form['age']

        blood=request.form['blood']

        email = request.form['email']

        phonenumber=request.form['phonenumber']

        district=request.form['district']

        pincode=request.form['pincode']

        password = request.form['password']

```

```

insertRecipientData(conn,username,firstname,lastname,age,gender,blood,email,phonenumber,password,district,pincode)

```

```

    return redirect(url_for('recipientauthentication',email=email))

    return render_template('recipientregister.html')

```

```

@app.route("/inchargeauthentication/<email>",methods=['GET','POST'])

```

```

def inchargeauthentication(email):

```

```
msg = Message('OTP',sender = 'gowrishankarj1110@gmail.com', recipients = [email])
msg.body = str(otp)
mail.send(msg)
return render_template('inchargeauthentication.html',email=email)
```

```
@app.route('/inchargevalidate',methods=["POST"])
```

```
def inchargevalidate():
```

```
    user_otp = request.form['otp']
```

```
    email=request.form['email']
```

```
    email.lower()
```

```
    if otp == int(user_otp):
```

```
        return render_template("inchargelogin.html")
```

```
    else:
```

```
        deleteInchargeData(conn,email)
```

```
        return "<h3>failure</h3>"
```

```
@app.route("/recipientauthentication/<email>",methods=['GET','POST'])
```

```
def recipientauthentication(email):
```

```
    msg = Message('OTP',sender = 'gowrishankarj1110@gmail.com', recipients = [email])
```

```
    msg.body = str(otp)
```

```
    mail.send(msg)
```

```
    return render_template('recipientauthentication.html',email=email)
```

```
@app.route('/recipientvalidate',methods=["POST"])
```

```
def recipientvalidate():
```

```
    user_otp = request.form['otp']
```

```
    email=request.form['email']
```

```
    email.lower()
```

```
    if otp == int(user_otp):
```

```
        return render_template("recipientlogin.html")
```

```
    else:
```

```
deleteRecipientData(conn,email)
```

```
return "<h3>failure</h3>"
```

```
@app.route("/donorauthentication/<email>",methods=['GET','POST'])
```

```
def donorauthentication(email):
```

```
    msg = Message('OTP',sender = 'gowrishankarj1110@gmail.com', recipients = [email])
```

```
    msg.body = str(otp)
```

```
    mail.send(msg)
```

```
    return render_template('donorauthentication.html',email=email)
```

```
@app.route('/donorvalidate',methods=["POST"])
```

```
def donorvalidate():
```

```
    user_otp = request.form['otp']
```

```
    email=request.form['email']
```

```
    email.lower()
```

```
    if otp == int(user_otp):
```

```
        return render_template("donorlogin.html")
```

```
    else:
```

```
        deleteDonorData(conn,email)
```

```
        return "<h3>failure</h3>"
```

```
@app.route("/donorlogin",methods=['POST','GET'])
```

```
def donorlogin():
```

```
    if request.method=="POST":
```

```
        printDonorData(conn)
```

```
        text=request.form['text']
```

```
        password=request.form['password']
```

```
    try:
```

```
        if dictionaryForEmailDonor[text] == password:
```

```
            return redirect(url_for('donorhome'))
```

```
    except:
```

```
        return "invalid email or password"
    return render_template("donorlogin.html")
```

```
@app.route("/donorregister",methods=['POST','GET'])
```

```
def donorregister():
```

```
    if request.method=="POST":
```

```
        username = request.form['username']
```

```
        firstname=request.form['firstname']
```

```
        lastname=request.form['lastname']
```

```
        gender=request.form['gender']
```

```
        age=request.form['age']
```

```
        blood=request.form['blood']
```

```
        email = request.form['email']
```

```
        phonenumber=request.form['phonenumber']
```

```
        district=request.form['district']
```

```
        pincode=request.form['pincode']
```

```
        password = request.form['password']
```

```
insertDonorData(conn,username,firstname,lastname,age,gender,blood,email,phonenumber,password,district,pincode)
```

```
        return redirect(url_for('donorauthentication',email=email))
```

```
    return render_template('donorregister.html')
```

```
@app.route("/donorhome")
```

```
def donorhome():
```

```
    return "donor home page"
```

```
@app.route("/recipienthome")
```

```
def recipienthome():
```

```
    return "recipient home page"
```

```
@app.route("/inchargehome")
```



```
def inchargehome():  
    return "incharge home page"
```

```
if __name__ == "__main__":  
    app.run(debug=True)
```