|  |
| --- |
| import smtplib, ssl |
|  | ## email.mime subclasses |
|  | from email.mime.multipart import MIMEMultipart |
|  | from email.mime.text import MIMEText |
|  | ## The pandas library is only for generating the current date, which is not necessary for sending emails |
|  | import pandas as pd |
|  | import os |
|  | from datetime import datetime |
|  |  |
|  | from flask import Flask |
|  | from turtle import st |
|  | from flask import Flask, render\_template, request, redirect, url\_for, session |
|  | from markupsafe import escape |
|  | import ibm\_db |
|  |  |
|  | conn = ibm\_db.connect("DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32286;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=nmm36298;PWD=zmbdjcoUXumahPoj",'','') |
|  |  |
|  | app = Flask(\_\_name\_\_) |
|  |  |
|  | picsFolder = os.path.join('static','pics') |
|  | app.config['UPLOAD\_FOLDER'] = picsFolder |
|  | var\_list = [] |
|  |  |
|  | @app.route('/') |
|  | def home(): |
|  | img = os.path.join(app.config['UPLOAD\_FOLDER'],'diabetic-retinopathy-illustration\_42265-59.webp') |
|  | img1 = os.path.join(app.config['UPLOAD\_FOLDER'],'ibmregister.jpg') |
|  | img2 = os.path.join(app.config['UPLOAD\_FOLDER'],'ibmlogin.jpg') |
|  | return render\_template('ibmprojecthome.html',imgg=img,imgg1=img1,imgg2=img2) |
|  |  |
|  | @app.route('/login') |
|  | def login(): |
|  | if request.method == 'POST' : |
|  | NAME = request.form['NAME'] |
|  | PASSWORD = request.form['PASSWORD'] |
|  | sql = "SELECT \* FROM users WHERE NAME =? AND PASSWORD=?" |
|  | stmt = ibm\_db.prepare(conn, sql) |
|  | ibm\_db.bind\_param(stmt,1,NAME) |
|  | ibm\_db.bind\_param(stmt,2,PASSWORD) |
|  | ibm\_db.execute(stmt) |
|  | account = ibm\_db.fetch\_assoc(stmt) |
|  | print (account) |
|  | print(fdfghg) |
|  | if account: |
|  | session['loggedin'] = True |
|  | session['id'] = account['USERNAME'] |
|  | userid= account['USERNAME'] |
|  | session['username'] = account['USERNAME'] |
|  | msg = 'Logged in successfully !' |
|  | print("sss") |
|  | # return render\_template('register.html', regImg=loginImg) |
|  | return redirect(url\_for('predict')) |
|  | else: |
|  | msg = 'Incorrect username / password !' |
|  | print(nj) |
|  | #return render\_template('login.html', msg = msg,loginImg=loginImg) |
|  |  |
|  | img3 = os.path.join(app.config['UPLOAD\_FOLDER'],'ibmregister.png') |
|  | return render\_template('ibmprojectlogin.html',imgg3=img3) |
|  |  |
|  |  |
|  | @app.route('/register') |
|  | def register(): |
|  | # img = os.path.join(app.config['UPLOAD\_FOLDER'],'diabetic-retinopathy-illustration\_42265-59.webp') |
|  | # img1 = os.path.join(app.config['UPLOAD\_FOLDER'],'ibmregister.jpg') |
|  | # img2 = os.path.join(app.config['UPLOAD\_FOLDER'],'ibmlogin.jpg') |
|  | img3 = os.path.join(app.config['UPLOAD\_FOLDER'],'ibmregister.png') |
|  | return render\_template('ibmprojectregister.html',imgg3=img3) |
|  |  |
|  | @app.route('/predict') |
|  | def predict(): |
|  | img4 = os.path.join(app.config['UPLOAD\_FOLDER'],'ibmpredict.jpg') |
|  | return render\_template('ibmprojectprediction.html',imgg4=img4) |
|  |  |
|  | @app.route('/addrec',methods=['POST','GET']) |
|  | def addrec(): |
|  | if request.method == 'POST' : |
|  | NAME = request.form['NAME'] |
|  | EMAIL = request.form['EMAIL'] |
|  | PASSWORD = request.form['PASSWORD'] |
|  | sql = "SELECT \* FROM Users 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) |
|  | if account: |
|  | msg = 'Account already exists !' |
|  |  |
|  | else: |
|  |  |
|  | var\_list.append(NAME) |
|  | var\_list.append(EMAIL) |
|  | var\_list.append(PASSWORD) |
|  | bodytemp = r"D:\\ibmme\\templates\\email.html" |
|  | with open(bodytemp, "r", encoding='utf-8') as f: |
|  | html= f.read() |
|  |  |
|  |  |
|  | # Set up the email addresses and password. Please replace below with your email address and password |
|  | email\_from = '1911047@nec.edu.in' |
|  | epassword = 'jnfulpwymmsjvpqo' |
|  | email\_to = EMAIL |
|  |  |
|  | # Generate today's date to be included in the email Subject |
|  | date\_str = pd.Timestamp.today().strftime('%Y-%m-%d') |
|  |  |
|  | # Create a MIMEMultipart class, and set up the From, To, Subject fields |
|  | email\_message = MIMEMultipart() |
|  | email\_message['From'] = email\_from |
|  | email\_message['To'] = email\_to |
|  | email\_message['Subject'] = f'Report email - {date\_str}' |
|  |  |
|  | # Attach the html doc defined earlier, as a MIMEText html content type to the MIME message |
|  | email\_message.attach(MIMEText(html, "html")) |
|  | # Convert it as a string |
|  | email\_string = email\_message.as\_string() |
|  |  |
|  | # Connect to the Gmail SMTP server and Send Email |
|  | context = ssl.create\_default\_context() |
|  | with smtplib.SMTP\_SSL("smtp.gmail.com", 465, context=context) as server: |
|  | server.login(email\_from, epassword) |
|  | server.sendmail(email\_from, email\_to, email\_string) |
|  | return render\_template('notify.html') |
|  |  |
|  | return render\_template('ibmprojectlogin.html') |
|  |  |
|  |  |
|  | @app.route('/confirm') |
|  | def confirmation(): |
|  | insert\_sql = "INSERT INTO Users (NAME, EMAIL, PASSWORD) VALUES (?,?,?)" |
|  | prep\_stmt = ibm\_db.prepare(conn, insert\_sql) |
|  | ibm\_db.bind\_param(prep\_stmt, 1, var\_list[0]) |
|  | ibm\_db.bind\_param(prep\_stmt, 2, var\_list[1]) |
|  | ibm\_db.bind\_param(prep\_stmt, 3, var\_list[2]) |
|  | ibm\_db.execute(prep\_stmt) |
|  | return render\_template('ibmprojectlogin.html') |
|  |  |
|  | @app.route('/logout') |
|  | def logout(): |
|  | session.pop('loggedin', None) |
|  | session.pop('id', None) |
|  | session.pop('username', None) |
|  | regImg = os.path.join(app.config['UPLOAD\_FOLDER'],'Plasma\_login.png') |
|  | return render\_template('login.html',regImg=regImg) |
|  |  |
|  |  |
|  | # main driver function |
|  | if \_\_name\_\_ == '\_\_main\_\_': |
|  | # run() method of Flask class runs the application |
|  | # on the local development server. |
|  | app.run() |

IBM