

PROJECT DEVELOPMENT

SPRINT 3 – Source code

DATE	14 NOVEMBER 2022
Team ID	PNT2022TMID30039
Project name	Customer care registry

```
from flask import Flask,render_template,request,url_for,session,redirect
from flask_mysqlldb import MySQL
from sendmail import
sendemail,forget_password_mail,updated_password_mail,solve_mail
import json
import ibm_db
import re
from random import randint
from datetime import date

app = Flask(__name__)
# http://remotemysql.com/

# dsn_hostname = "b0aebb68-94fa-46ec-a1fc-
1c999edb6187.c3n41cmd0nqn timer 39u98g.databases.appdomain.cloud"
# dsn_uid = "dmt13873"
# dsn_pwd = "740yZ1Yq8Uj2E4qm"
# dsn_database = 'bludb'
# dsn_port = 31249
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-a1fc-
1c999edb6187.c3n41cmd0nqn timer 39u98g.databases.appdomain.cloud;PORT=31249;SECURITY=SSL;SSLServerCertificate=src/DigiCertGlobalRootCA.crt;UID=dmt13873;PWD=740yZ1Yq8Uj
2E4qm",'','') # type: ignore
print(conn)
print("connection successful...")

# database configuration
# app.config['MYSQL_HOST'] = 'sql12.freesqldatabase.com'
# app.config['MYSQL_USER'] = 'sql12552843'
# app.config['MYSQL_PASSWORD'] = 'zWIzHmXNi8'
# app.config['MYSQL_DB'] = 'sql12552843'
app.secret_key = "super secret key"
```

```

# mysql = MySQL(app)

@app.route('/')
def home():
    today = date.today()
    current_date = today.strftime('%d/%m/%Y')
    if "google_token" in session:
        session["current_date"] = current_date
        return render_template('home.html')
    if "username" in session:
        session["current_date"] = current_date
        return render_template('home.html')
    return render_template('index.html')

# manually registration
@app.route('/register',methods=["POST"])
def register():
    if request.method == 'POST':
        name = request.form['uname']
        mail = request.form['mail']
        pwd = request.form['pwd']
        cpwd = request.form['confirmpwd']
        if not re.match(r'^@[^@]+\.[^@]+', mail):
            msg = 'Invalid email address !'
            return render_template('index.html',signupmsg=msg)
        if pwd != cpwd:
            msg = 'Please enter correct confirm password'
            return render_template('index.html',signupmsg=msg)
        # check account is exists or not
        # cursor = mysql.connection.cursor()
        rCheckQuery = ''
        result = ibm_db.exec_immediate(conn,f"SELECT * FROM customerdeatils WHERE
email LIKE '{mail}'")
        # cursor.execute('SELECT * FROM customerdeatils WHERE email LIKE %
s',[mail])
        # existing_user = cursor.fetchone()
        # cursor.close()
        existing_user = ibm_db.fetch_row(result)
        #exits
        if existing_user:
            msg = 'Account already exists please login.'
            return render_template('index.html',signupmsg = msg)
        # not exists

        # cursor = mysql.connection.cursor()

```

```

        # cursor.execute('INSERT INTO customerdeatils VALUES(null,% s,% s,%
s)',(name,mail,pwd))
        # mysql.connection.commit()
        # cursor.close()
        regInsertQuery = f"INSERT INTO customerdeatils (username,email,passwd)
VALUES('{name}','{mail}','{pwd}')"
        insertflag = ibm_db.exec_immediate(conn,regInsertQuery)
        msg = 'Your registration successfully completed.'
        # send mail
        sendemail(mail,'Account_creation')

    return render_template('index.html',signupmsg = msg)
# admin page
@app.route('/admin/<which>')
def admin(which):
    if which == 'customers':
        # cursor = mysql.connection.cursor()
        result = ibm_db.exec_immediate(conn,'SELECT * FROM customerdeatils')
        data = []
        while ibm_db.fetch_row(result):
            temp =
[ibm_db.result(result,0),ibm_db.result(result,1),ibm_db.result(result,2),ibm_db.r
esult(result,3)]
            data.append(temp)
        return render_template('admin.html',customers=data,complaints=None)
    if which == 'complaints':
        # cursor = mysql.connection.cursor()
        result = ibm_db.exec_immediate(conn,'SELECT * FROM complaints')
        data = []
        while ibm_db.fetch_row(result):
            temp =
[ibm_db.result(result,0),ibm_db.result(result,1),ibm_db.result(result,2),ibm_db.r
esult(result,3),ibm_db.result(result,4),ibm_db.result(result,5)]
            data.append(temp)
        return render_template('admin.html',customers=None,complaints=data)
# admin delete
@app.route('/Delete/<type>/<id>')
def Delete(type,id):
    if type == 'customers':
        # cursor = mysql.connection.cursor()
        result = ibm_db.exec(conn,f'DELETE FROM customerdeatils WHERE id =
"{id}"')
        # mysql.connection.commit()
        # cursor.close()
        return redirect(url_for('admin',which='customers'))

```

```

    if type == 'complaints':
        # cursor = mysql.connection.cursor()
        result = ibm_db.exec_immediate(conn,f'DELETE FROM complaints WHERE id =
{id}')
        # mysql.connection.commit()
        # cursor.close()
        return redirect(url_for('admin',which='complaints'))
# manually login
@app.route('/login',methods=['POST','GET'])
def login():
    if request.method == 'POST':
        mail = request.form['mail1']
        password = request.form['pwd1']
        # login is admin or not
        if mail == "admin" and password == 'admin@1810':
            return redirect(url_for('admin',which='customers'))
        # check account is exists or not
        # cursor = mysql.connection.cursor()
        query = "SELECT * FROM customerdeatils WHERE email=? AND passwd=?"
        stmt = ibm_db.prepare(conn, query) # type:ignore
        ibm_db.bind_param(stmt,1,mail) # type:ignore
        ibm_db.bind_param(stmt,2,password) # type:ignore
        ibm_db.execute(stmt) # type:ignore
        user = ibm_db.fetch_assoc(stmt) # type:ignore
        print(user,password)
        #exists
        if user:
            session["username"] = user['USERNAME']
            session['mail'] = mail
            return
render_template('home.html',username=session["username"],mail=session["mail"])
        else:
            msg = 'mail or password is not valid.'
            return render_template('index.html',signinmsg=msg)
    if request.method == "GET":
        return redirect(url_for('home'))

# logout method
@app.route('/logout')
def logout():
    if "username" in session:
        session.pop("username")
    if "google_token" in session:
        session.pop("google_token")
        session.pop("mail")

```

```

    if "mail" in session:
        session.pop("mail")
    return redirect(url_for('home'))

# complaint register
@app.route('/complaint',methods=['POST'])
def complaint():
    if request.method == 'POST':
        complaint_name = request.form['complaint_name']
        name = request.form['name']
        mail = request.form['email']
        against_person = request.form['against_person']
        date = request.form["date"]
        des = request.form['complaint_des']
        # cursor = mysql.connection.cursor()
        if not name == session["username"] or not mail == session["mail"]:
            msg = "please don't change username and mail."
            return render_template('home.html',msg=msg)
        result = ibm_db.exec_immediate(conn,f"INSERT INTO complaints
(username,email,against_person,des,date,solved)
VALUES('{name}','{mail}','{against_person}','{des}','{date}','{0}')"
        # mysql.connection.commit()
        # cursor.close()
        sendemail(mail,'complaint_creation')
        msg = 'Complaint registered you check out complaints section.'
        return render_template('home.html',msg=msg)

# show complaints and progress
@app.route('/showcomplaints')
def showcomplaints():
    # cursor = mysql.connection.cursor()
    # cursor.execute("SELECT * FROM complaints WHERE username= % s AND email=%
s",(session["username"],session["mail"]))
    # details = cursor.fetchall()
    # cursor.close()
    query = "SELECT * FROM complaints WHERE username=? AND email=?"
    stmt = ibm_db.prepare(conn, query) # type:ignore
    ibm_db.bind_param(stmt,1,session["username"]) # type:ignore
    ibm_db.bind_param(stmt,2,session['mail']) # type:ignore
    ibm_db.execute(stmt)
    data = []
    while ibm_db.fetch_row(stmt):
        temp =
[ibm_db.result(stmt,0),ibm_db.result(stmt,1),ibm_db.result(stmt,2),ibm_db.result(
stmt,3),ibm_db.result(stmt,4),ibm_db.result(stmt,5),ibm_db.result(stmt,6)]

```

```

        print(temp)
        data.append(temp)
    return render_template('complaints.html',complaints=data)

# update complaint
@app.route('/solve',methods=["POST"])
def solve_complaint():
    if request.method == "POST":
        c_id = request.form['c_id']

        print(c_id)
        # cursor = mysql.connection.cursor()
        # cursor.execute("UPDATE complaints SET solved = % s WHERE id = %
s",('1',c_id,))
        query = "UPDATE complaints SET solved = '1' WHERE id = ?"
        # mysql.connection.commit()
        stmt = ibm_db.prepare(conn, query) # type:ignore
        ibm_db.bind_param(stmt,1,c_id) # type:ignore
        ibm_db.execute(stmt)
        detail = ibm_db.result(stmt,0)
        print(detail)
        # cursor.execute("SELECT * FROM complaints WHERE id = % s",[c_id])
        query2 = "SELECT * FROM complaints WHERE id = ?"
        stmt1 = ibm_db.prepare(conn, query2) # type:ignore
        ibm_db.bind_param(stmt1,1,c_id) # type:ignore
        ibm_db.execute(stmt1)
        details = ibm_db.result(stmt1,0)
        # cursor.close()
        print(details)
        # solve_mail(session['mail'],'user')
        return redirect(url_for('showcomplaints'))
    return redirect(url_for('showcomplaints'))

# # admin agent allot
# @app.route('/solve_admin',methods=["POST"])
# def solve_admin():
#     if request.method == "POST":
#         c_id = request.form['c_id']
#         # cursor = mysql.connection.cursor()
#         cursor.execute("SELECT * FROM complaints WHERE id = % s",[c_id])
#         query = "SELECT * FROM complaints WHERE id = ?"
#         details = cursor.fetchone()
#         cursor.close()
#         solve_mail(details[3],'admin')
#         return redirect(url_for('admin',which='complaints'))

```

```

#     return redirect(url_for('admin',which='complaints'))
# remove complaint
@app.route('/dismiss',methods=["POST"])
def dismiss_complaint():
    if request.method == "POST":
        c_id = request.form["c_id"]
        # cursor = mysql.connection.cursor()
        # cursor.execute("DELETE FROM complaints WHERE id = % s",[c_id])
        # mysql.connection.commit()
        # cursor.close()
        query = "DELETE FROM complaints WHERE id = ?"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt,1,c_id) # type:ignore
        ibm_db.execute(stmt)
        return redirect(url_for('showcomplaints'))
    return redirect(url_for('showcomplaints'))
# send otp in user mail id
@app.route('/send_otp',methods=["POST","GET"])
def send_otp():
    if request.method == "POST":
        mail = request.form["mail"]
        cursor = mysql.connection.cursor()
        cursor.execute("SELECT * FROM customerdeatils WHERE email = % s",[mail])
        temp = cursor.fetchone()
        cursor.close()
        if not temp:
            return render_template('forget.html',type='otp',msg1='Your account
doesn\'t exist please register')
        otp = randint(10 ** 5,10**6)
        forget_password_mail(mail,otp)
        session["otp"] = otp
        return
render_template('forget.html',type='update_password',tempmail=mail)
# forget password method
@app.route('/forgetpassword/<type>',methods=["POST","GET"])
def forgetpassword(type):
    if type == 'otp':
        return render_template('forget.html',type=type)
    if request.method == "POST":
        mail = request.form["mail"]
        otp = request.form["otp"]
        pwd = request.form["password"]
        c_pwd = request.form["con_pwd"]
        print(otp,session['otp'])
        if not pwd == c_pwd:

```

```
        msg = 'Please Enter Password properly'
        return render_template('forget.html',type='updatePassword',msg=msg)
    if not otp == str(session['otp']):
        msg = "Your OTP is Incorrect."
        return render_template('forget.html',type='updatePassword',msg=msg)
    cursor = mysql.connection.cursor()
    cursor.execute("UPDATE customerdeatils SET passwd = % s WHERE email = %
s",(pwd,mail))
    mysql.connection.commit()
    cursor.close()
    msg = 'password updated successfully'
    updated_password_mail(mail)
    return render_template('forget.html',type='updatePassword',msg=msg)

if __name__ == '__main__':
    app.run(host = '0.0.0.0',port = 8080,debug=True)
```