from flask import Flask, render\_template, request, redirect, session, url\_for, flash

import mysql.connector

from flask import request

from datetime import datetime

app = Flask(\_\_name\_\_)

app.secret\_key = 'your\_secret\_key'

def get\_db\_connection():

    connection = mysql.connector.connect(

*host*='localhost',

*port*=3307,

*user*='root',

*password*='',

*database*='qlbida'

    )

    return connection

*# ĐĂNG KÝ TÀI KHOẢN*

@app.route('/register', *methods*=['GET', 'POST'])

def register():

    if request.method == 'POST':

        username = request.form['username']

        password = request.form['password']

        full\_name = request.form['full\_name']

        phone\_number = request.form['phone\_number']

        address = request.form['address']

        role\_id = request.form['role\_id']  *# role\_id là một số nguyên*

        conn = get\_db\_connection()

        if conn.is\_connected():

            try:

*# Kiểm tra xem tên đăng nhập đã tồn tại chưa*

                cur = conn.cursor()

                cur.execute("SELECT id FROM account WHERE username = %s", (username,))

                account = cur.fetchone()

                if account:

                    flash('Tên đăng nhập đã tồn tại, vui lòng chọn tên đăng nhập khác.', 'error')

                else:

*# Thêm tài khoản mới vào cơ sở dữ liệu*

                    cur.execute("INSERT INTO account (username, password, full\_name,phone\_number,address  , role\_id) VALUES (%s, %s, %s, %s,%s,%s)",

                                (username, password, full\_name,phone\_number,address  , role\_id))

                    conn.commit()

                    flash('Đăng ký tài khoản thành công!', 'success')

                    return redirect(url\_for('login'))

            except mysql.connector.Error as e:

                print(e)

                flash('Đã xảy ra lỗi khi đăng ký tài khoản.', 'error')

            finally:

                cur.close()

                conn.close()

        else:

            flash('Không thể kết nối đến cơ sở dữ liệu.', 'error')

    return render\_template('register.html')

@app.route('/')

def index():

    return render\_template('index.html')

*# ĐĂNG NHẬP*

@app.route('/login', *methods*=['GET', 'POST'])

def login():

    if request.method == 'POST':

        username = request.form['username']

        password = request.form['password']

        conn = get\_db\_connection()

        if conn.is\_connected():

            try:

                cur = conn.cursor()

                cur.execute("SELECT a.id, a.username, a.role\_id, r.name FROM account a JOIN role r ON a.role\_id = r.id WHERE username = %s AND password = %s", (username, password))

                account = cur.fetchone()

                if account:

                    role\_id = account[2]

                    session['loggedin'] = True

                    session['id'] = account[0]

                    session['username'] = account[1]

                    session['role\_id'] = role\_id

                    session['role\_name'] = account[3]

                    if role\_id == 1:  *# Admin role\_id*

                        session['loggedin\_admin'] = True

                        session.pop('loggedin\_nhanvien', None)  *# Xóa nếu có*

                        flash('Đăng nhập thành công admin!', 'success')

                        return redirect(url\_for('admin'))

                    elif role\_id == 2:  *# Nhân viên role\_id*

                        session['loggedin\_nhanvien'] = True

                        session.pop('loggedin\_admin', None)  *# Xóa nếu có*

                        flash('Đăng nhập thành công nhân viên!', 'success')

                        return redirect(url\_for('nhanvien'))

                    else:

                        flash('Không có quyền truy cập.', 'error')

                else:

                    flash('Tên đăng nhập hoặc mật khẩu không đúng.', 'error')

            except mysql.connector.Error as e:

                print(e)

                flash('Đã xảy ra lỗi khi đăng nhập.', 'error')

            finally:

                cur.close()

                conn.close()

        else:

            flash('Không thể kết nối đến cơ sở dữ liệu.', 'error')

    return render\_template('login.html')

*# ADMIN từ ĐĂNG NHẬP*

@app.route('/admin')

def admin():

    if 'loggedin\_admin' in session:

        conn = get\_db\_connection()

        if conn.is\_connected():

            try:

                cur = conn.cursor()

                cur.execute("SELECT \* FROM account WHERE id = %s", (session['id'],))

                account = cur.fetchone()

                return render\_template('admin.html', *account*=account)

            except mysql.connector.Error as e:

                print(e)

            finally:

                cur.close()

                conn.close()

        return render\_template('admin.html')

    else:

        return redirect(url\_for('login'))

*# NHÂN VIÊN từ ĐĂNG NHẬP*

@app.route('/nhanvien')

def nhanvien():

    if 'loggedin\_nhanvien' in session:

        conn = get\_db\_connection()

        if conn.is\_connected():

            try:

                cur = conn.cursor()

                cur.execute("SELECT \* FROM account WHERE id = %s", (session['id'],))

                account = cur.fetchone()

                return render\_template('nhanvien.html', *account*=account)

            except mysql.connector.Error as e:

                print(e)

            finally:

                cur.close()

                conn.close()

        return render\_template('nhanvien.html')

    else:

        return redirect(url\_for('login'))

*# ĐĂNG XUẤT*

@app.route('/logout')

def logout():

    session.pop('loggedin\_nhanvien', None)

    session.pop('loggedin\_admin', None)

    session.pop('id', None)

    session.pop('username', None)

    session.pop('role\_id', None)

    flash('Bạn đã đăng xuất thành công!', 'success')

    return redirect(url\_for('login'))

*# ------------------QUẢN LÝ BÀN-----------------------*

@app.route('/quanlybillards')

def quanlybillards():

    connection = get\_db\_connection()

    cursor = connection.cursor(*dictionary*=True)

    cursor.execute("SELECT \* FROM banbida1")

    tables = cursor.fetchall()

    cursor.close()

    connection.close()

    return render\_template('quan\_li\_ban\_bia/banbia.html', *tables*=tables)

@app.route('/chitiethoadon/<int:ma\_ban>')

def chitiethoadon(*ma\_ban*):

    connection = get\_db\_connection()

    cursor = connection.cursor(*dictionary*=True)

*# Fetch the invoice details*

    cursor.execute("SELECT \* FROM hoadon WHERE MaBan = %s ORDER BY NgayTao DESC LIMIT 1", (*ma\_ban*,))

    hoadon = cursor.fetchone()

    if not hoadon:

        cursor.close()

        connection.close()

        return redirect(url\_for('quanlybillards'))

*# Fetch the ordered products*

    cursor.execute("""

        SELECT sp.TenSP, cto.SoLuong, sp.Gia

        FROM chitietorder cto

        JOIN sanpham sp ON cto.MaSP = sp.MaSP

        WHERE cto.MaHoaDon = %s

    """, (hoadon['MaHoaDon'],))

    ordered\_products = cursor.fetchall()

*# Fetch the list of all products*

    cursor.execute("SELECT \* FROM sanpham")

    products = cursor.fetchall()

    cursor.close()

    connection.close()

    return render\_template('quan\_li\_ban\_bia/invoice\_details.html', *hoadon*=hoadon, *ordered\_products*=ordered\_products, *products*=products)

@app.route('/add\_to\_order', *methods*=['POST'])

def add\_to\_order():

    ma\_hoa\_don = request.form['MaHoaDon']

    ma\_sp = request.form['MaSP']

    so\_luong = int(request.form['SoLuong'])

    connection = get\_db\_connection()

    cursor = connection.cursor()

    try:

*# Check if the product is already in the order*

        cursor.execute("SELECT \* FROM chitietorder WHERE MaHoaDon = %s AND MaSP = %s", (ma\_hoa\_don, ma\_sp))

        order\_item = cursor.fetchone()

        if order\_item:

*# If the product is already in the order, update the quantity*

            cursor.execute("UPDATE chitietorder SET SoLuong = SoLuong + %s WHERE MaHoaDon = %s AND MaSP = %s", (so\_luong, ma\_hoa\_don, ma\_sp))

        else:

*# Otherwise, insert a new record*

            cursor.execute("INSERT INTO chitietorder (MaHoaDon, MaSP, SoLuong) VALUES (%s, %s, %s)", (ma\_hoa\_don, ma\_sp, so\_luong))

*# Update product quantity in sanpham table*

        cursor.execute("UPDATE sanpham SET SoLuong = SoLuong - %s WHERE MaSP = %s", (so\_luong, ma\_sp))

        connection.commit()

        cursor.close()

        connection.close()

        return redirect(url\_for('chitiethoadon', *ma\_ban*=request.form['MaBan']))

    except Exception as e:

        print(f"Error: {e}")

        cursor.close()

        connection.close()

        return redirect(url\_for('error\_page'))

@app.route('/tao\_hoadon', *methods*=['POST'])

def tao\_hoadon():

    MaBan = request.form.get('MaBan')

*# Ensure the table exists and is available*

    connection = get\_db\_connection()

    cursor = connection.cursor(*dictionary*=True)

    cursor.execute("SELECT \* FROM banbida1 WHERE MaBan = %s AND TinhTrang = 'Trống'", (MaBan,))

    table = cursor.fetchone()

    if table:

*# Insert new invoice*

        cursor.execute(

            "INSERT INTO hoadon (MaBan, ThoiGianBatDau) VALUES (%s, NOW())",

            (MaBan,)

        )

        connection.commit()

*# Get the new invoice ID*

        new\_invoice\_id = cursor.lastrowid

*# Update table status to "Có khách"*

        cursor.execute(

            "UPDATE banbida1 SET TinhTrang = 'Có khách' WHERE MaBan = %s",

            (MaBan,)

        )

        connection.commit()

        cursor.close()

        connection.close()

*# Redirect to the invoice details page*

        return redirect(url\_for('chitiethoadon', *ma\_ban*=MaBan))

    cursor.close()

    connection.close()

    return redirect(url\_for('quanlybillards'))

@app.route('/ket\_thuc\_hoadon', *methods*=['POST'])

def ket\_thuc\_hoadon():

    MaHoaDon = request.form.get('MaHoaDon')

    connection = get\_db\_connection()

    cursor = connection.cursor()

    try:

*# Update ThoiGianKetThuc*

        cursor.execute("UPDATE hoadon SET ThoiGianKetThuc = NOW() WHERE MaHoaDon = %s", (MaHoaDon,))

*# Calculate and update ThanhTienThoigianchoi (in seconds)*

        cursor.execute("UPDATE hoadon SET ThanhTienThoigianchoi = TIMESTAMPDIFF(SECOND, ThoiGianBatDau, NOW()) \* 100 WHERE MaHoaDon = %s", (MaHoaDon,))

*# Calculate TongTien*

        cursor.execute("SELECT SUM(chitietorder.SoLuong \* sanpham.Gia) AS TongTien FROM chitietorder JOIN sanpham ON chitietorder.MaSP = sanpham.MaSP WHERE chitietorder.MaHoaDon = %s", (MaHoaDon,))

        tong\_tien = cursor.fetchone()[0]

        cursor.execute("UPDATE hoadon SET ThanhTienOrder = %s, TongTien = ThanhTienThoigianchoi + ThanhTienOrder WHERE MaHoaDon = %s", (tong\_tien, MaHoaDon,))

*# Update table status to 'Trống'*

        cursor.execute("UPDATE banbida1 SET TinhTrang = 'Trống' WHERE MaBan IN (SELECT MaBan FROM hoadon WHERE MaHoaDon = %s)", (MaHoaDon,))

        connection.commit()

        cursor.close()

        connection.close()

*# Redirect to select\_member page with invoice\_id*

        return redirect(url\_for('select\_member', *invoice\_id*=MaHoaDon))

    except Exception as e:

*# Handle exceptions here*

        print(f"Error: {e}")

        cursor.close()

        connection.close()

*# Redirect to an error page or handle accordingly*

        return redirect(url\_for('error\_page'))

@app.route('/select\_member/<int:invoice\_id>')

def select\_member(*invoice\_id*):

    connection = get\_db\_connection()

    cursor = connection.cursor(*dictionary*=True)

*# Fetch invoice details*

    cursor.execute("SELECT \* FROM hoadon WHERE MaHoaDon = %s", (*invoice\_id*,))

    hoadon = cursor.fetchone()

    if not hoadon:

        cursor.close()

        connection.close()

        return redirect(url\_for('index'))  *# Redirect to index if invoice not found*

*# Fetch list of members*

    cursor.execute("SELECT \* FROM member")

    members = cursor.fetchall()

    cursor.close()

    connection.close()

    return render\_template('select\_member.html', *hoadon*=hoadon, *members*=members)

@app.route('/update\_invoice\_member', *methods*=['POST'])

def update\_invoice\_member():

    MaHoaDon = request.form.get('MaHoaDon')

    MaMember = request.form.get('MaMember')

    connection = get\_db\_connection()

    cursor = connection.cursor()

*# Update MaMember in hoadon table*

    cursor.execute("UPDATE hoadon SET MaMember = %s WHERE MaHoaDon = %s", (MaMember, MaHoaDon))

    connection.commit()

    cursor.close()

    connection.close()

    return redirect(url\_for('quanlybillards'))

*#THÊM BÀN*

@app.route('/themban', *methods*=['GET', 'POST'])

def them\_ban():

    if request.method == 'POST':

        ten\_ban = request.form['ten\_ban']

        loai\_bida = request.form['loai\_bida']

        tinh\_trang = request.form['tinh\_trang']

        connection = get\_db\_connection()

        cursor = connection.cursor()

        insert\_query = "INSERT INTO banbida1 (TenBan, LoaiBida, TinhTrang) VALUES (%s, %s, %s)"

        cursor.execute(insert\_query, (ten\_ban, loai\_bida, tinh\_trang))

        connection.commit()

        cursor.close()

        connection.close()

        flash('Bàn mới đã được thêm thành công!', 'success')

        return redirect(url\_for('quanlybillards'))

    return render\_template('quan\_li\_ban\_bia/them\_ban.html')

*#XÓA BÀN*

@app.route('/xoaban/<int:ma\_ban>', *methods*=['GET', 'POST'])

def xoa\_ban(*ma\_ban*):

    connection = get\_db\_connection()

    cursor = connection.cursor()

    cursor.execute("SELECT \* FROM banbida1 WHERE MaBan = %s", (*ma\_ban*,))

    table = cursor.fetchone()

    if table:

        delete\_query = "DELETE FROM banbida1 WHERE MaBan = %s"

        cursor.execute(delete\_query, (*ma\_ban*,))

        connection.commit()

        flash(f'Bàn có ID {*ma\_ban*} đã được xóa thành công!', 'success')

    else:

        flash(f'Bàn có ID {*ma\_ban*} không tồn tại.', 'error')

    cursor.close()

    connection.close()

    return redirect(url\_for('quanlybillards'))

*#----------------QUẢN LÝ NHÂN VIÊN----------------------*

@app.route('/quanlynhanvien')

def quanlynhanvien():

    keyword = request.args.get('keyword', '')

    connection = get\_db\_connection()

    cursor = connection.cursor(*dictionary*=True)

    if keyword:

        cursor.execute("SELECT a.\* FROM account a JOIN role r ON a.role\_id = r.id WHERE r.id = 2 AND (a.full\_name LIKE %s OR a.phone\_number LIKE %s)", ('%' + keyword + '%', '%' + keyword + '%'))

    else:

        cursor.execute("SELECT a.\* FROM account a ")

    nhanviens = cursor.fetchall()

    cursor.close()

    connection.close()

    return render\_template('quan\_li\_nhan\_vien/nhanvien.html', *nhanviens*=nhanviens)

*#THÊM NHÂN VIÊN*

@app.route('/themnhanvien', *methods*=['GET', 'POST'])

def them\_nhan\_vien():

    if request.method == 'POST':

        username = request.form['username']

        password = request.form['password']

        full\_name = request.form['full\_name']

        phone\_number = request.form['phone\_number']

        address = request.form['address']

        role\_id = 2

        connection = get\_db\_connection()

        cursor = connection.cursor()

        insert\_query = "INSERT INTO account (username, password,full\_name,phone\_number,address,role\_id) VALUES (%s, %s, %s, %s, %s, %s)"

        cursor.execute(insert\_query, (username, password,full\_name,phone\_number,address,role\_id))

        connection.commit()

        cursor.close()

        connection.close()

        flash('Nhân viên mới đã được thêm thành công!', 'success')

        return redirect(url\_for('quanlynhanvien'))

    return render\_template('quan\_li\_nhan\_vien/them\_nhan\_vien.html')

*#SỬA NHÂN VIÊN*

@app.route('/sua\_nhan\_vien/<int:id>', *methods*=['GET', 'POST'])

def sua\_nhan\_vien(*id*):

    connection = get\_db\_connection()

    cursor = connection.cursor(*dictionary*=True)

    if request.method == 'POST':

        username = request.form['username']

        password = request.form['password']

        full\_name = request.form['full\_name']

        phone\_number = request.form['phone\_number']

        address = request.form['address']

        update\_query = """

            UPDATE account

            SET username = %s, password = %s, full\_name = %s, phone\_number = %s, address = %s

            WHERE id = %s

        """

        cursor.execute(update\_query, (username, password, full\_name, phone\_number, address, *id*))

*# Nếu có bảng lương, cập nhật thông tin lương của nhân viên tại đây*

*# Ví dụ: update\_salary(id, salary\_info)*

        connection.commit()

        flash('Thông tin nhân viên đã được cập nhật!', 'success')

        return redirect(url\_for('quanlynhanvien'))

    cursor.execute("SELECT \* FROM account WHERE id = %s", (*id*,))

    nhanvien = cursor.fetchone()

    cursor.close()

    connection.close()

    return render\_template('quan\_li\_nhan\_vien/sua\_nhan\_vien.html', *nhanvien*=nhanvien)

*# XÓA NHÂN VIÊN*

@app.route('/xoanhanvien/<int:id>', *methods*=['GET', 'POST'])

def xoa\_nhan\_vien(*id*):

    connection = get\_db\_connection()

    cursor = connection.cursor()

    try:

*# Kiểm tra và xóa các bản ghi liên quan trong bảng salary*

        delete\_salary\_query = "DELETE FROM luong WHERE account\_id = %s"

        cursor.execute(delete\_salary\_query, (*id*,))

*## Kiểm tra và xóa các bản ghi liên quan trong bảng penalty*

        delete\_penalty\_query = "DELETE FROM ca\_lam\_viec WHERE account\_id = %s"

        cursor.execute(delete\_penalty\_query, (*id*,))

*# Tiếp tục xóa nhân viên từ bảng account*

        delete\_query\_account = "DELETE FROM account WHERE id = %s"

        cursor.execute(delete\_query\_account, (*id*,))

        connection.commit()

        flash(f'Nhân Viên có ID {*id*} đã được xóa thành công!', 'success')

    except Exception as e:

        flash(f'Đã xảy ra lỗi khi xóa nhân viên: {str(e)}', 'error')

        connection.rollback()

    finally:

        cursor.close()

        connection.close()

    return redirect(url\_for('quanlynhanvien'))

*# -------------------QUẢN LÝ SẢN PHẨM-----------------------*

*#----------------QUẢN LÝ SẢN PHẨM----------------------*

@app.route('/quanlysanpham')

def quanlysanpham():

    keyword = request.args.get('keyword', '')

    connection = get\_db\_connection()

    cursor = connection.cursor(*dictionary*=True)

    if keyword:

        cursor.execute("SELECT \* FROM sanpham WHERE (TenSP LIKE %s)", ('%' + keyword + '%',))

    else:

        cursor.execute("SELECT \* FROM sanpham")

    sanphams = cursor.fetchall()

    cursor.close()

    connection.close()

    return render\_template('quan\_li\_san\_pham/sanpham.html', *sanphams*=sanphams)

*#THÊM SẢN PHẨM*

@app.route('/themsanpham', *methods*=['GET', 'POST'])

def them\_san\_pham():

    if request.method == 'POST':

        TenSP = request.form['TenSP']

        Gia = request.form['Gia']

        SoLuong = request.form['SoLuong']

        connection = get\_db\_connection()

        cursor = connection.cursor()

        insert\_query = "INSERT INTO sanpham (TenSP, Gia, SoLuong) VALUES (%s, %s, %s)"

        cursor.execute(insert\_query, (TenSP, Gia, SoLuong))

        connection.commit()

        cursor.close()

        connection.close()

        flash('Sản phẩm mới đã được thêm thành công!', 'success')

        return redirect(url\_for('quanlysanpham'))

    return render\_template('quan\_li\_san\_pham/them\_san\_pham.html')

*#SỬA SẢN PHẨM*

@app.route('/suasanpham/<int:MaSP>', *methods*=['GET', 'POST'])

def sua\_san\_pham(*MaSP*):

    connection = get\_db\_connection()

    cursor = connection.cursor(*dictionary*=True)

    if request.method == 'POST':

        TenSP = request.form['TenSP']

        Gia = request.form['Gia']

        SoLuong = request.form['SoLuong']

        update\_query = "UPDATE sanpham SET TenSP = %s, Gia = %s, SoLuong = %s WHERE MaSP = %s"

        cursor.execute(update\_query, (TenSP, Gia, SoLuong, *MaSP*))

        connection.commit()

        flash('Thông tin sản phẩm đã được cập nhật!', 'success')

        return redirect(url\_for('quanlysanpham'))

    cursor.execute("SELECT \* FROM sanpham WHERE MaSP = %s", (*MaSP*,))

    sanpham = cursor.fetchone()

    cursor.close()

    connection.close()

    return render\_template('quan\_li\_san\_pham/sua\_san\_pham.html', *sanpham*=sanpham)

@app.route('/xoasanpham/<int:MaSP>', *methods*=['GET', 'POST'])

def xoa\_san\_pham(*MaSP*):

    connection = get\_db\_connection()

    cursor = connection.cursor()

    cursor.execute("SELECT \* FROM sanpham WHERE MaSP = %s", (*MaSP*,))

    sanpham = cursor.fetchone()

    if sanpham:

        delete\_query = "DELETE FROM sanpham WHERE MaSP = %s"

        cursor.execute(delete\_query, (*MaSP*,))

        connection.commit()

        flash(f'Sản phẩm có ID {*MaSP*} đã được xóa thành công!', 'success')

    else:

        flash(f'Sản phẩm có ID {*MaSP*} không tồn tại.', 'error')

    cursor.close()

    connection.close()

    return redirect(url\_for('quanlysanpham'))

*#----------------QUẢN LÝ KHÁCH HÀNG----------------------*

@app.route('/quanlykhachhang')

def quanlykhachhang():

    keyword = request.args.get('keyword', '')

    connection = get\_db\_connection()

    cursor = connection.cursor(*dictionary*=True)

    if keyword:

        cursor.execute("SELECT \* FROM member WHERE (TenMember LIKE %s OR SoDienThoai LIKE %s)", ('%' + keyword + '%', '%' + keyword + '%'))

    else:

        cursor.execute("SELECT \* FROM member")

    khachhangs = cursor.fetchall()

    cursor.close()

    connection.close()

    return render\_template('quan\_li\_khach\_hang/khachhang.html', *khachhangs*=khachhangs)

*#THÊM KHÁCH HÀNG*

@app.route('/themkhachhang', *methods*=['GET', 'POST'])

def them\_khach\_hang():

    if request.method == 'POST':

        TenMember = request.form['TenMember']

        DiaChi = request.form['DiaChi']

        SoDienThoai = request.form['SoDienThoai']

        Email = request.form['Email']

        connection = get\_db\_connection()

        cursor = connection.cursor()

        insert\_query = "INSERT INTO member (TenMember, DiaChi, SoDienThoai,Email) VALUES (%s, %s, %s, %s)"

        cursor.execute(insert\_query, (TenMember, DiaChi, SoDienThoai,Email))

        connection.commit()

        cursor.close()

        connection.close()

        flash('Khách Hàng mới đã được thêm thành công!', 'success')

        return redirect(url\_for('quanlykhachhang'))

    return render\_template('quan\_li\_khach\_hang/them\_khach\_hang.html')

*#SỬA KHÁCH HÀNG*

@app.route('/sua\_khach\_hang/<int:MaMember>', *methods*=['GET', 'POST'])

def sua\_khach\_hang(*MaMember*):

    connection = get\_db\_connection()

    cursor = connection.cursor(*dictionary*=True)

    if request.method == 'POST':

        TenMember = request.form['TenMember']

        DiaChi = request.form['DiaChi']

        SoDienThoai = request.form['SoDienThoai']

        Email = request.form['Email']

        update\_query = "UPDATE member SET TenMember = %s, DiaChi = %s, SoDienThoai = %s, Email = %s WHERE MaMember = %s"

        cursor.execute(update\_query, (TenMember, DiaChi, SoDienThoai,Email, *MaMember*))

        connection.commit()

        flash('Thông tin khách hàng đã được cập nhật!', 'success')

        return redirect(url\_for('quanlykhachhang'))

    cursor.execute("SELECT \* FROM member WHERE MaMember = %s", (*MaMember*,))

    khachhangs = cursor.fetchone()

    cursor.close()

    connection.close()

    return render\_template('quan\_li\_khach\_hang/sua\_khach\_hang.html', *khachhang*=khachhangs)

*#XÓA KHÁCH HÀNG*

@app.route('/xoakhachhang/<int:MaMember>', *methods*=['GET', 'POST'])

def xoa\_khach\_hang(*MaMember*):

    connection = get\_db\_connection()

    cursor = connection.cursor()

    cursor.execute("SELECT \* FROM member WHERE MaMember = %s", (*MaMember*,))

    khachhang = cursor.fetchone()

    if nhanvien:

        delete\_query = "DELETE FROM member WHERE MaMember = %s"

        cursor.execute(delete\_query, (*MaMember*,))

        connection.commit()

        flash(f'Khách Hàng có ID {*MaMember*} đã được xóa thành công!', 'success')

    else:

        flash(f'Khách Hàng có ID {*MaMember*} không tồn tại.', 'error')

    cursor.close()

    connection.close()

    return redirect(url\_for('quanlykhachhang'))

*#----------------------------*

@app.route('/phan\_cong\_viec')

def phan\_cong\_viec():

    conn = get\_db\_connection()

    cursor = conn.cursor()

    cursor.execute("""

        SELECT

            clv.id,

            a.full\_name,

            clv.cong\_viec,

            clv.ngay\_ca\_lam,

            clv.gio\_bat\_dau,

            clv.gio\_ket\_thuc,

            p.ly\_do,

            p.so\_tien\_phat

        FROM ca\_lam\_viec clv

        JOIN account a ON clv.account\_id = a.id

        LEFT JOIN phat p ON clv.phat\_id = p.id

    """)

    thong\_tin = cursor.fetchall()

    cursor.close()

    conn.close()

    return render\_template('quan\_li\_nhan\_vien/phan\_cong\_viec.html', *thong\_tin*=thong\_tin)

@app.route('/them\_cong\_viec', *methods*=['GET', 'POST'])

def them\_cong\_viec():

    if request.method == 'POST':

        account\_id = request.form['account\_id']

        cong\_viec = request.form['cong\_viec']

        ngay\_ca\_lam = request.form['ngay\_ca\_lam']

        gio\_bat\_dau = request.form['gio\_bat\_dau']

        gio\_ket\_thuc = request.form['gio\_ket\_thuc']

        try:

            conn = get\_db\_connection()

            cursor = conn.cursor()

*# Thực hiện INSERT vào bảng ca\_lam\_viec*

            query = "INSERT INTO ca\_lam\_viec (account\_id, cong\_viec, ngay\_ca\_lam, gio\_bat\_dau, gio\_ket\_thuc) VALUES (%s, %s, %s, %s, %s)"

            cursor.execute(query, (account\_id, cong\_viec, ngay\_ca\_lam, gio\_bat\_dau, gio\_ket\_thuc))

            conn.commit()

            flash('Thêm công việc mới thành công', 'success')

        except mysql.connector.Error as e:

            print(f'Error: {e}')

            flash('Thêm công việc mới thất bại', 'error')

        finally:

            cursor.close()

            conn.close()

        return redirect(url\_for('phan\_cong\_viec'))

*# Lấy danh sách tài khoản để hiển thị trong dropdown*

    conn = get\_db\_connection()

    cursor = conn.cursor()

    cursor.execute("SELECT id, full\_name FROM account")

    accounts = cursor.fetchall()

    cursor.close()

    conn.close()

    return render\_template('quan\_li\_nhan\_vien/them\_cong\_viec.html', *accounts*=accounts)

@app.route('/sua\_cong\_viec/<int:id>', *methods*=['GET', 'POST'])

def sua\_cong\_viec(*id*):

    if request.method == 'POST':

        account\_id = request.form['account\_id']

        cong\_viec = request.form['cong\_viec']

        ngay\_ca\_lam = request.form['ngay\_ca\_lam']

        gio\_bat\_dau = request.form['gio\_bat\_dau']

        gio\_ket\_thuc = request.form['gio\_ket\_thuc']

        phat\_id = request.form['phat\_id'] if request.form['phat\_id'] else None

        try:

            conn = get\_db\_connection()

            cursor = conn.cursor()

*# Thực hiện UPDATE vào bảng ca\_lam\_viec*

            query = """

                UPDATE ca\_lam\_viec

                SET account\_id = %s, cong\_viec = %s, ngay\_ca\_lam = %s, gio\_bat\_dau = %s, gio\_ket\_thuc = %s, phat\_id = %s

                WHERE id = %s

            """

            cursor.execute(query, (account\_id, cong\_viec, ngay\_ca\_lam, gio\_bat\_dau, gio\_ket\_thuc, phat\_id, *id*))

            conn.commit()

            flash('Cập nhật công việc thành công', 'success')

        except mysql.connector.Error as e:

            print(f'Error: {e}')

            flash('Cập nhật công việc thất bại', 'error')

        finally:

            cursor.close()

            conn.close()

        return redirect(url\_for('phan\_cong\_viec'))

*# Lấy thông tin công việc để hiển thị trong form*

    conn = get\_db\_connection()

    cursor = conn.cursor()

    cursor.execute("SELECT account\_id, cong\_viec, ngay\_ca\_lam, gio\_bat\_dau, gio\_ket\_thuc, phat\_id FROM ca\_lam\_viec WHERE id = %s", (*id*,))

    cong\_viec = cursor.fetchone()

    cursor.execute("SELECT id, full\_name FROM account")

    accounts = cursor.fetchall()

    cursor.execute("SELECT id, ly\_do FROM phat")

    phat = cursor.fetchall()

    cursor.close()

    conn.close()

    return render\_template('quan\_li\_nhan\_vien/sua\_cong\_viec.html', *id*=*id*, *cong\_viec*=cong\_viec, *accounts*=accounts, *phat*=phat)

@app.route('/xoa\_cong\_viec/<int:id>', *methods*=['POST'])

def xoa\_cong\_viec(*id*):

    if request.method == 'POST':

        try:

            conn = get\_db\_connection()

            cursor = conn.cursor()

*# Thực hiện DELETE từ bảng ca\_lam\_viec*

            query = "DELETE FROM ca\_lam\_viec WHERE id = %s"

            cursor.execute(query, (*id*,))

            conn.commit()

            flash('Xóa công việc thành công', 'success')

        except mysql.connector.Error as e:

            print(f'Error: {e}')

            flash('Xóa công việc thất bại', 'error')

        finally:

            cursor.close()

            conn.close()

        return redirect(url\_for('phan\_cong\_viec'))

*# Các xử lý khác khi request.method là GET hoặc những phương thức khác*

    return redirect(url\_for('phan\_cong\_viec'))

@app.route('/tim\_kiem\_cong\_viec', *methods*=['GET', 'POST'])

def tim\_kiem\_cong\_viec():

    if request.method == 'POST':

        search\_term = request.form['search\_term']

        conn = get\_db\_connection()

        cursor = conn.cursor()

*# Tìm kiếm các bản ghi trong bảng ca\_lam\_viec*

        query = """

            SELECT a.full\_name, clv.cong\_viec, clv.ngay\_ca\_lam, clv.gio\_bat\_dau, clv.gio\_ket\_thuc, p.ly\_do, p.so\_tien\_phat, clv.id

            FROM ca\_lam\_viec clv

            JOIN account a ON clv.account\_id = a.id

            LEFT JOIN phat p ON clv.phat\_id = p.id

            WHERE a.full\_name LIKE %s OR clv.cong\_viec LIKE %s OR p.ly\_do LIKE %s

        """

        cursor.execute(query, (f"%{search\_term}%", f"%{search\_term}%", f"%{search\_term}%"))

        thong\_tin = cursor.fetchall()

        cursor.close()

        conn.close()

        return render\_template('quan\_li\_nhan\_vien/tim\_kiem\_cong\_viec.html', *thong\_tin*=thong\_tin)

    return render\_template('quan\_li\_nhan\_vien/tim\_kiem\_cong\_viec.html')

*# Route để tính tổng lương và tổng số tiền phạt theo tháng/năm*

@app.route('/tong\_luong\_thang')

def tong\_luong\_thang():

    conn = get\_db\_connection()

    cur = conn.cursor()

    cur.execute("""

        SELECT

            l.id,

            l.account\_id,

            l.thang,

            l.nam,

            l.luong\_co\_ban,

            l.thuong,

            (l.luong\_co\_ban + IFNULL(l.thuong, 0) - IFNULL(p.so\_tien\_phat, 0)) AS tong\_luong,

            IFNULL(SUM(p.so\_tien\_phat), 0) AS tong\_so\_tien\_phat\_thang

        FROM luong l

        JOIN ca\_lam\_viec clv ON l.account\_id = clv.account\_id

        LEFT JOIN phat p ON clv.phat\_id = p.id AND MONTH(clv.ngay\_ca\_lam) = l.thang AND YEAR(clv.ngay\_ca\_lam) = l.nam

        GROUP BY l.id, l.account\_id, l.thang, l.nam, l.luong\_co\_ban, l.thuong;

    """)

    tong\_luong\_thang = cur.fetchall()

    cur.close()

    conn.close()

    return render\_template('quan\_li\_nhan\_vien/tong\_luong\_thang.html', *tong\_luong\_thang*=tong\_luong\_thang)

*# Route để tính tổng lương và tổng số tiền phạt theo năm*

@app.route('/tong\_luong\_nam')

def tong\_luong\_nam():

    conn = get\_db\_connection()

    cur = conn.cursor()

    cur.execute("""

SELECT

    l.account\_id,

    a.full\_name,

    l.nam,

    SUM(DISTINCT l.luong\_co\_ban) AS tong\_luong\_co\_ban,

    SUM(DISTINCT IFNULL(l.thuong, 0)) AS tong\_thuong,

    SUM(DISTINCT IFNULL(p.so\_tien\_phat, 0)) AS tong\_so\_tien\_phat\_nam,

    SUM(DISTINCT l.luong\_co\_ban + IFNULL(l.thuong, 0) - IFNULL(p.so\_tien\_phat, 0)) AS tong\_luong

FROM luong l

JOIN account a ON l.account\_id = a.id

LEFT JOIN (

    SELECT

        clv.account\_id,

        YEAR(clv.ngay\_ca\_lam) AS nam,

        SUM(p.so\_tien\_phat) AS so\_tien\_phat

    FROM ca\_lam\_viec clv

    LEFT JOIN phat p ON clv.phat\_id = p.id

    GROUP BY clv.account\_id, YEAR(clv.ngay\_ca\_lam)

) p ON l.account\_id = p.account\_id AND l.nam = p.nam

GROUP BY l.account\_id, l.nam

ORDER BY l.account\_id, l.nam;

    """)

    tong\_luong\_nam = cur.fetchall()

    cur.close()

    conn.close()

    return render\_template('quan\_li\_nhan\_vien/tong\_luong\_nam.html', *tong\_luong\_nam*=tong\_luong\_nam)

*# MAIN*

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(*debug*=True)