```
from flask import Blueprint, jsonify, request
from backend import conn
from backend.auth_middleware import token_required
import ibm_db

user = Blueprint("user", __name__)


@user.route("/skills", methods=["GET", "POST", "DELETE"])
@token_required
def manage_skills(current_user):
    # Get user_id of current user
    user_id = current_user['USER_ID']

    # Handle GET request
    if request.method == 'GET':
        skills = []

        sql = f"select name from skills where user_id={user_id}"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.execute(stmt)
        dict = ibm_db.fetch_assoc(stmt)

        # Iterate over all the results and append skills to the array
        while dict != False:
            skills.append(dict['NAME'])
            dict = ibm_db.fetch_assoc(stmt)

        return jsonify({"skills": skills}), 200

    # Get the skills from the request
```

```
if not ('skills' in request.json):
    return jsonify({"error": f"All feilds are required!"}), 409

skills = request.json['skills']

# If no skills are provided then return empty array
if skills == []:
    return jsonify({"skills": []}), 200

# Handle POST request
if request.method == "POST":
    # Prepare the SQL statement to insert multiple rows
    values = ''
    for i in range(len(skills)):
        if i == 0:
            values += 'values'
        values += f"('{skills[i]}',{user_id})"
        if i != len(skills)-1:
            values += ','
    sql = f"insert into skills(name,user_id) {values}"
    stmt = ibm_db.prepare(conn, sql)
    status = ibm_db.execute(stmt)

    if status:
        return jsonify({"message": "Updated skills successfully!"}), 200
    else:
        jsonify({"error": "Something went wrong!!"}), 409

# Handle DELETE request
if request.method == 'DELETE':
    values = ""
```

```
        for i in range(len(skills)):
            values += f"'{skills[i]}'"
            if i != len(skills)-1:
                values += ','
        sql = f"delete from skills where name in ({values})"
        stmt = ibm_db.prepare(conn, sql)
        status = ibm_db.execute(stmt)
        if status:
            return jsonify({"message": "Deleted skills successfully!"}), 200
        else:
            jsonify({"error": "Something went wrong!!"}), 409


@user.route('/profile', methods=["POST"])
@token_required
def update_user_info(current_user):
    user_id = current_user['USER_ID']
    update_fields = ['name', 'phone_number']
    for feild in update_fields:
        if not (feild in request.json):
            return jsonify({"error": f"All feilds are required!"}), 409
    name = request.json['name']
    phone_number = request.json['phone_number']
    sql = f"update users set name='{name}',phone_number='{phone_number}' where user_id={user_id}"
    stmt = ibm_db.prepare(conn, sql)
    status = ibm_db.execute(stmt)
    if status:
        return jsonify({"name": name, "phone_number": phone_number}), 200
    else:
        jsonify({"error": "Something went wrong!!"}), 409
```