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