import datetime  
import warnings  
  
from flask import Flask, jsonify, render\_template, request  
from keras.models import load\_model  
import numpy as np  
import pymysql  
  
from config import db\_name, host, logger, model\_name, password, port, user  
  
warnings.filterwarnings("ignore")  
  
app = Flask(\_\_name\_\_)  
  
try:  
 connection = pymysql.connect(  
 host=host,  
 port=port,  
 user=user,  
 password=password,  
 database=db\_name,  
 cursorclass=pymysql.cursors.DictCursor  
 )  
 logger.info("Успешное подключение к БД.")  
 try:  
 with connection.cursor() as cursor:  
 create\_table\_query = "CREATE TABLE requests (id int AUTO\_INCREMENT," \  
 "user\_id varchar(32) ," \  
 "req\_date datetime," \  
 "vector varchar(256)," \  
 "action\_list varchar(32), " \  
 "PRIMARY KEY (id));"  
 cursor.execute(create\_table\_query)  
 connection.commit()  
 logger.info("Успешное создание таблицы")  
 except Exception as exc:  
 logger.error(f"Таблица уже создана. Ошибка: {exc}")  
except Exception as exc:  
 logger.error(f"Не удалось подключиться к БД. Ошибка: {exc}")  
  
  
def insert\_values(user\_id, req\_date, vector, test\_actions):  
 try:  
 with connection.cursor() as cursor:  
 insert\_query = "INSERT INTO requests (user\_id, req\_date, vector, action\_list) VALUES (%s, %s," \  
 "%s, %s);"  
 cursor.execute(insert\_query, (user\_id, req\_date.strftime('%Y-%m-%d %H:%M:%S'), vector, test\_actions))  
 connection.commit()  
 except Exception as exc:  
 logger.error(f"Ошибка при добавлении данных: {exc}")  
  
  
def select\_all():  
 try:  
 with connection.cursor() as cursor:  
 cursor.execute('select \* from requests')  
 dataset = cursor.fetchall()  
 for data in dataset:  
 data.update({'req\_date': datetime.datetime.strftime(data.get('req\_date'), '%Y-%m-%d %H:%M:%S')})  
 connection.commit()  
 return dataset  
 except Exception as exc:  
 logger.error(f"Ошибка при выборке данных: {exc}")  
  
  
@app.route('/')  
def first\_page():  
 return render\_template('index.html')  
  
  
@app.route('/staff', methods=['GET'])  
def get\_staff():  
 dataset = select\_all()  
 for data in dataset:  
 logger.debug(data)  
 return jsonify(dataset)  
  
  
@app.route('/staff', methods=['POST'])  
def get\_action():  
 output = {}  
 action\_list = []  
 model = load\_model(model\_name)  
 req = request.json  
 for data in req.keys():  
 if "action" in data:  
 for el in req[data].split(' '):  
 action\_list.append(el)  
 embedding\_output = model.predict(action\_list)  
 kumulative\_sum = np.zeros(len(embedding\_output[0]))  
  
 for every\_action\_embedding in embedding\_output:  
 kumulative\_sum += every\_action\_embedding  
 for el in req.keys():  
 if "id" in el:  
 output['user\_id'] = req[el]  
 output['actions\_list'] = action\_list  
 output['vector'] = kumulative\_sum.tolist()  
 logger.debug(type(output['vector']))  
 output['date'] = datetime.datetime.now()  
 insert\_values(str(output['user\_id']), output['date'],  
 ' '.join(str(el) for el in output['vector']), ' '.join(str(el) for el in output['actions\_list']))  
 return jsonify(output)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run()