from flask import Flask, jsonify, request  
import mysql.connector  
from mysql.connector import Error  
  
app = Flask(\_\_name\_\_)  
  
def create\_connection():  
 connection = None  
 try:  
 print('Connexion à la base de données...')  
 connection = mysql.connector.connect(  
 host='localhost',   
 user='root',  
 password='password',  
 database='videos\_db'  
 )  
 except Error as e:  
 print(f"The error '{e}' occurred")  
  
 return connection  
  
# Liste toutes les vidéos dans la base de données  
@app.route('/videos', methods=['GET'])  
def lister\_videos():  
 connection = create\_connection()  
 cursor = connection.cursor(dictionary=True)  
 query = "SELECT \* FROM videos ORDER BY ordre"  
 cursor.execute(query)  
 videos = cursor.fetchall()  
 cursor.close()  
 connection.close()  
 return jsonify(videos)  
  
# Marquer une vidéo comme étant en cours de lecture et enregistrer le temps de début de lecture  
@app.route('/video/current', methods=['POST'])  
def marquer\_video\_courante():  
 data = request.json  
 connection = create\_connection()  
 cursor = connection.cursor()  
 query = """  
 INSERT INTO video\_courant (id\_video, debut\_video, fin\_video, temps\_jouer)  
 VALUES (%s, NOW(), NULL, 0)  
 """  
 cursor.execute(query, (data['id\_video'],))  
 connection.commit()  
 cursor.close()  
 connection.close()  
 return jsonify({"success": True}), 201  
  
# Mettre à jour les informations de la vidéo courante lorsqu'elle est terminée de jouer.  
# Enregistre le temps de lecture  
@app.route('/video/current', methods=['PUT'])  
def terminer\_video\_courante():  
 data = request.json  
 connection = create\_connection()  
 cursor = connection.cursor()  
 query = """  
 UPDATE video\_courant  
 SET fin\_video = NOW(), temps\_jouer = %s  
 WHERE id\_video = %s AND fin\_video IS NULL  
 """  
 cursor.execute(query, (data['temps\_jouer'], data['id\_video']))  
 mettre\_a\_jour\_stats(data['id\_video'], data['temps\_jouer'])  
 connection.commit()  
 cursor.close()  
 connection.close()  
 return jsonify({"success": True})  
  
# Mettre à jour les statistiques pour une vidéo donnée pour la journée en cours  
# Si une entrée pour cette vidéo et cette journée existe déjà, elle est mise à jour (la vidéo a déjà été jouée aujourd'hui)  
# Sinon, une nouvelle entrée est créée (la vidéo n'a pas encore été jouée aujourd'hui)  
def mettre\_a\_jour\_stats(id\_video, temps\_jouer):  
 connection = create\_connection()  
 cursor = connection.cursor()  
   
 # Vérifier si une entrée pour aujourd'hui et cette vidéo existe déjà  
 query = """  
 SELECT id\_nb FROM nb\_video\_jour  
 WHERE date\_jour = CURDATE() AND id\_video = %s  
 """  
 cursor.execute(query, (id\_video,))  
 result = cursor.fetchone()  
   
 if result:  
 # Mise à jour de l'entrée existante  
 query = """  
 UPDATE nb\_video\_jour  
 SET nb\_jouer = nb\_jouer + 1, temps\_total = temps\_total + %s  
 WHERE id\_nb = %s  
 """  
 cursor.execute(query, (temps\_jouer, result[0]))  
 else:  
 # Création d'une nouvelle entrée  
 query = """  
 INSERT INTO nb\_video\_jour (date\_jour, id\_video, nb\_jouer, temps\_total)  
 VALUES (CURDATE(), %s, 1, %s)  
 """  
 cursor.execute(query, (id\_video, temps\_jouer))  
   
 connection.commit()  
 cursor.close()  
 connection.close()  
   
# Obtenir les statistiques pour la journée en cours  
# Retourne les statistiques par vidéo et les statistiques globales  
@app.route('/stats/jour', methods=['GET'])  
def obtenir\_stats\_jour():  
 connection = create\_connection()  
 cursor = connection.cursor(dictionary=True)  
   
 # Requête pour les statistiques par vidéo  
 query\_stats\_par\_video = """  
 SELECT id\_video, SUM(nb\_jouer) AS nb\_jouer, SUM(temps\_total) AS temps\_total  
 FROM nb\_video\_jour  
 WHERE date\_jour = CURDATE()  
 GROUP BY id\_video  
 """  
 cursor.execute(query\_stats\_par\_video)  
 stats\_par\_video = cursor.fetchall()  
  
 # Requête pour les statistiques totales  
 query\_stats\_globales = """  
 SELECT SUM(nb\_jouer) AS nb\_jouer\_total, SUM(temps\_total) AS temps\_total  
 FROM nb\_video\_jour  
 WHERE date\_jour = CURDATE()  
 """  
 cursor.execute(query\_stats\_globales)  
 stats\_globales = cursor.fetchone()   
  
 cursor.close()  
 connection.close()  
   
 reponse = {  
 'stats\_par\_video': stats\_par\_video,  
 'stats\_globales': stats\_globales  
 }  
   
 return jsonify(reponse)