from flask import Flask, request, jsonify, render\_template

import sqlite3

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

# Database helper function to fetch questions by category

def get\_questions(category, limit=10):

    try:

        conn = sqlite3.connect('quiz.db')

        cursor = conn.cursor()

        cursor.execute("SELECT \* FROM questions WHERE category = ? ORDER BY RANDOM() LIMIT ?", (category, limit))

        rows = cursor.fetchall()

        conn.close()

        # Format questions into a list of dictionaries

        questions = []

        for row in rows:

            questions.append({

                "id": row[0],

                "category": row[1],

                "question\_text": row[2],

                "options": [row[3], row[4], row[5], row[6]],

                "correct\_option": row[7]

            })

        return questions

    except Exception as e:

        return {"error": str(e)}

# Homepage route - this should return an HTML page if you're doing frontend work

@app.route('/')

def home():

    return render\_template('index.html')  # Serve the HTML file from templates folder

# Endpoint to fetch questions based on the difficulty level (category)

@app.route('/get\_questions', methods=['GET'])

def fetch\_questions():

    category = request.args.get('category', 'C')  # Default to Category C

    questions = get\_questions(category)

    # Handle case when there are no questions or an error

    if "error" in questions:

        return jsonify({"error": questions["error"]}), 500

    return jsonify(questions)

# Endpoint to submit answers and calculate score

@app.route('/submit\_answers', methods=['POST'])

def submit\_answers():

    try:

        data = request.json

        user\_answers = data.get('answers', [])

        category = data.get('category', 'C')

        if not user\_answers:

            return jsonify({"error": "No answers submitted"}), 400

        score = 0

        for answer in user\_answers:

            question\_id = answer.get('question\_id')

            selected\_option = answer.get('selected\_option')

            if not question\_id or not selected\_option:

                return jsonify({"error": "Invalid answer format"}), 400

            # Verify the answer from the database

            conn = sqlite3.connect('quiz.db')

            cursor = conn.cursor()

            cursor.execute("SELECT correct\_option FROM questions WHERE id = ?", (question\_id,))

            correct\_option = cursor.fetchone()

            conn.close()

            if correct\_option:

                correct\_option = correct\_option[0]

                if selected\_option == correct\_option:

                    score += 1

            else:

                return jsonify({"error": f"Question {question\_id} not found"}), 404

        # Adjust difficulty level based on score

        if score >= 8:

            next\_category = 'C'

        elif score >= 5:

            next\_category = category  # Stay in the current category

        else:

            next\_category = 'A'

        return jsonify({"score": score, "next\_category": next\_category})

    except Exception as e:

        return jsonify({"error": str(e)}), 500

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

    app.run(debug=True)