Dưới đây là phiên bản cập nhật đầy đủ cho file app.py của bạn, mình đã bổ sung thêm 2 route xử lý sửa và xóa câu hỏi, giữ nguyên phần hiện có, bạn chỉ cần thay thế toàn bộ app.py hiện tại bằng đoạn này:

from flask import Flask, render\_template, request, jsonify, redirect, url\_for, abort

from flask\_sqlalchemy import SQLAlchemy

from sqlalchemy import text

from datetime import datetime

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

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///quiz\_results.db'

app.config['SQLALCHEMY\_ENGINE\_OPTIONS'] = {"connect\_args": {"timeout": 10}}

app.config['SQLALCHEMY\_TRACK\_MODIFICATIONS'] = False

db = SQLAlchemy(app)

class Question(db.Model):

id = db.Column(db.Integer, primary\_key=True)

question\_text = db.Column(db.String(200), nullable=False)

answer\_a = db.Column(db.String(100), nullable=False)

answer\_b = db.Column(db.String(100), nullable=False)

answer\_c = db.Column(db.String(100), nullable=False)

answer\_d = db.Column(db.String(100), nullable=False)

correct\_answer = db.Column(db.String(1), nullable=False)

class QuizResult(db.Model):

id = db.Column(db.Integer, primary\_key=True)

name = db.Column(db.String(100), nullable=False)

start\_time = db.Column(db.DateTime, nullable=False)

score = db.Column(db.Integer, nullable=False)

with app.app\_context():

db.create\_all()

# Nếu muốn bật WAL mode, có thể bật sau khi tạo bảng:

# with db.engine.connect() as conn:

# conn.execute(text('PRAGMA journal\_mode=WAL;'))

@app.route('/')

def index():

questions = Question.query.all()

return render\_template('index.html', questions=questions)

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

def submit():

data = request.json

name = data.get('name')

score = data.get('score')

start\_time\_str = data.get('start\_time')

if not name or score is None or not start\_time\_str:

return jsonify({'error': 'Thiếu dữ liệu'}), 400

try:

start\_time = datetime.strptime(start\_time\_str, "%Y-%m-%d %H:%M:%S")

except:

return jsonify({'error': 'Sai định dạng ngày giờ'}), 400

result = QuizResult(name=name, score=score, start\_time=start\_time)

db.session.add(result)

db.session.commit()

return jsonify({'message': 'Lưu kết quả thành công'})

@app.route('/admin/questions')

def admin\_questions():

questions = Question.query.order\_by(Question.id.asc()).all()

return render\_template('admin\_questions.html', questions=questions)

@app.route('/admin/questions/add', methods=['GET', 'POST'])

def add\_question():

if request.method == 'POST':

question\_text = request.form.get('question\_text', '').strip()

answer\_a = request.form.get('answer\_a', '').strip()

answer\_b = request.form.get('answer\_b', '').strip()

answer\_c = request.form.get('answer\_c', '').strip()

answer\_d = request.form.get('answer\_d', '').strip()

correct\_answer = request.form.get('correct\_answer', '').strip().upper()

if not all([question\_text, answer\_a, answer\_b, answer\_c, answer\_d, correct\_answer]):

return render\_template('add\_question.html', error="Vui lòng nhập đầy đủ thông tin!")

if correct\_answer not in ['A', 'B', 'C', 'D']:

return render\_template('add\_question.html', error="Đáp án đúng phải là A, B, C hoặc D.")

try:

new\_q = Question(

question\_text=question\_text,

answer\_a=answer\_a,

answer\_b=answer\_b,

answer\_c=answer\_c,

answer\_d=answer\_d,

correct\_answer=correct\_answer

)

db.session.add(new\_q)

db.session.commit()

except Exception as e:

db.session.rollback()

return render\_template('add\_question.html', error=f"Lỗi khi lưu câu hỏi: {str(e)}")

return redirect(url\_for('admin\_questions'))

return render\_template('add\_question.html')

@app.route('/admin/questions/edit/<int:question\_id>', methods=['GET', 'POST'])

def edit\_question(question\_id):

question = Question.query.get\_or\_404(question\_id)

if request.method == 'POST':

question\_text = request.form.get('question\_text', '').strip()

answer\_a = request.form.get('answer\_a', '').strip()

answer\_b = request.form.get('answer\_b', '').strip()

answer\_c = request.form.get('answer\_c', '').strip()

answer\_d = request.form.get('answer\_d', '').strip()

correct\_answer = request.form.get('correct\_answer', '').strip().upper()

if not all([question\_text, answer\_a, answer\_b, answer\_c, answer\_d, correct\_answer]):

return render\_template('edit\_question.html', question=question, error="Vui lòng nhập đầy đủ thông tin!")

if correct\_answer not in ['A', 'B', 'C', 'D']:

return render\_template('edit\_question.html', question=question, error="Đáp án đúng phải là A, B, C hoặc D.")

try:

question.question\_text = question\_text

question.answer\_a = answer\_a

question.answer\_b = answer\_b

question.answer\_c = answer\_c

question.answer\_d = answer\_d

question.correct\_answer = correct\_answer

db.session.commit()

except Exception as e:

db.session.rollback()

return render\_template('edit\_question.html', question=question, error=f"Lỗi khi cập nhật câu hỏi: {str(e)}")

return redirect(url\_for('admin\_questions'))

return render\_template('edit\_question.html', question=question)

@app.route('/admin/questions/delete/<int:question\_id>', methods=['POST'])

def delete\_question(question\_id):

question = Question.query.get\_or\_404(question\_id)

try:

db.session.delete(question)

db.session.commit()

except Exception as e:

db.session.rollback()

# Có thể xử lý lỗi ở đây nếu muốn

return redirect(url\_for('admin\_questions'))

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

app.run(debug=True)

**Tóm tắt:**

* Đã thêm 2 route /admin/questions/edit/<int:question\_id> để sửa câu hỏi.
* Đã thêm route /admin/questions/delete/<int:question\_id> xử lý xóa câu hỏi.
* Đảm bảo gọi render\_template với file edit\_question.html khi sửa.
* Giữ nguyên các route và chức năng hiện tại.

Bạn chỉ việc lưu lại, khởi động lại Flask, kiểm tra các chức năng thêm, sửa, xóa trong trang quản trị.

Nếu cần hỗ trợ thêm hoặc có lỗi phát sinh, cứ báo nhé!