# Розширення: підтримка типів OA (відкрита відповідь) і MA (відповідність)

Цей документ доповнює базовий посібник. Додаємо два типи: OA — відкрита відповідь; MA — встановити відповідність. Нижче — CSV-приклади, скрипт імпорту та оновлення Flask/HTML.

## 1) CSV для OA (відкрита відповідь)

Файл `questions\_oa.csv`. Поле `correct\_text` може містити кілька допустимих варіантів, розділених крапкою з комою `;`. Пробіли та регістр ігноруються, кома замінюється на крапку.

### questions\_oa.csv (приклад)

subject,year,topic,text,correct\_text,source  
math,2024,Алгебра,"Спростіть 6x + 12 / 3","2x+4; 2x + 4","НМТ"  
math,2024,Степені,"Обчисліть 2^3 \* 2^4","2^7; 128","НМТ"  
ukr,2024,Орфографія,"Напишіть слово з правильним наголосом (кава)","кава","власна база"

(Файл: questions\_oa.csv)

## 2) CSV для MA (відповідність)

Файл `questions\_ma.csv`. Кожне завдання MA займає кілька рядків з однаковим `code` — це ідентифікатор одного питання. Для кожного рядка вказуємо пару `left` → `right` і однаковий `pair\_key` (A,B,C...).

### questions\_ma.csv (приклад)

code,subject,year,topic,text,left,right,pair\_key,source  
Q1,ukr,2024,Наголос,"Установіть відповідність: слово — наголос","квартал","кварта́л","A","НМТ"  
Q1,ukr,2024,Наголос,"Установіть відповідність: слово — наголос","удар","уда́р","B","НМТ"  
Q1,ukr,2024,Наголос,"Установіть відповідність: слово — наголос","колос","ко́лос","C","НМТ"  
Q2,math,2024,Властивості степенів,"Установіть відповідність виразу та результату","x^2\*x^3","x^5","A","НМТ"  
Q2,math,2024,Властивості степенів,"Установіть відповідність виразу та результату","a^3\*a^2","a^5","B","НМТ"

(Файл: questions\_ma.csv)

## 3) Скрипт імпорту OA та MA у SQLite

Скрипт читає `questions\_oa.csv` і `questions\_ma.csv`, створює записи в таблиці `questions`, у `match\_pairs` додає пари. Запустіть після базового імпорту SC або незалежно — він не видаляє існуючі дані.

### import\_oa\_ma.py

import os, csv, sqlite3  
  
BASE\_DIR = os.path.dirname(os.path.abspath(\_\_file\_\_))  
DB\_PATH = os.path.join(BASE\_DIR, "questions.db")  
  
con = sqlite3.connect(DB\_PATH)  
cur = con.cursor()  
  
# ===== OA =====  
oa\_path = os.path.join(BASE\_DIR, "questions\_oa.csv")  
if os.path.exists(oa\_path):  
 with open(oa\_path, newline="", encoding="utf-8") as f:  
 rdr = csv.DictReader(f)  
 added\_oa = 0  
 for r in rdr:  
 subject = r["subject"].strip()  
 year = int(r["year"]) if r["year"].strip() else None  
 topic = (r.get("topic") or "").strip() or None  
 text = r["text"].strip()  
 correct\_text = r["correct\_text"].strip()  
 source = (r.get("source") or "").strip() or None  
 cur.execute("""INSERT INTO questions  
 (subject,year,topic,type,text,correct\_index,correct\_text,explanation,source)  
 VALUES (?,?,?,?,?,?,?,?,?)""",  
 (subject,year,topic,"OA",text,None,correct\_text,None,source))  
 added\_oa += 1  
 print(f"OA додано: {added\_oa}")  
  
# ===== MA =====  
ma\_path = os.path.join(BASE\_DIR, "questions\_ma.csv")  
if os.path.exists(ma\_path):  
 with open(ma\_path, newline="", encoding="utf-8") as f:  
 rdr = csv.DictReader(f)  
 grouped = {}  
 for r in rdr:  
 code = r["code"].strip()  
 grouped.setdefault(code, []).append(r)  
  
 added\_ma = 0  
 for code, rows in grouped.items():  
 head = rows[0]  
 subject = head["subject"].strip()  
 year = int(head["year"]) if head["year"].strip() else None  
 topic = (head.get("topic") or "").strip() or None  
 text = head["text"].strip()  
 source = (head.get("source") or "").strip() or None  
 # питання MA  
 cur.execute("""INSERT INTO questions  
 (subject,year,topic,type,text,correct\_index,correct\_text,explanation,source)  
 VALUES (?,?,?,?,?,?,?,?,?)""",  
 (subject,year,topic,"MA",text,None,None,None,source))  
 qid = cur.lastrowid  
 for r in rows:  
 left = r["left"].strip()  
 right = r["right"].strip()  
 key = r["pair\_key"].strip()  
 cur.execute("""INSERT INTO match\_pairs  
 (question\_id,left\_text,right\_text,pair\_key)  
 VALUES (?,?,?,?)""", (qid,left,right,key))  
 added\_ma += 1  
 print(f"MA додано (завдань): {added\_ma}")  
  
con.commit()  
con.close()  
print("Готово: імпорт OA/MA завершено.")

(Файл: import\_oa\_ma.py)

## 4) Оновлення Flask: показ і перевірка SC, OA, MA

Замініть/оновіть логіку отримання питання та перевірки відповіді. Тепер додаток підтягує опції для SC, пари для MA, та порівнює текст для OA.

### app.py (фрагменти: завантаження питання та перевірка)

import os, sqlite3, random  
from flask import Flask, render\_template, request, session, redirect, url\_for  
  
app = Flask(\_\_name\_\_)  
app.secret\_key = "nmt-secret"  
  
BASE\_DIR = os.path.dirname(os.path.abspath(\_\_file\_\_))  
DB\_PATH = os.path.join(BASE\_DIR, "questions.db")  
  
def db():  
 con = sqlite3.connect(DB\_PATH); con.row\_factory = sqlite3.Row; return con  
  
def get\_random\_question(subject=None):  
 con = db(); cur = con.cursor()  
 if subject:  
 cur.execute("SELECT \* FROM questions WHERE subject=? ORDER BY RANDOM() LIMIT 1", (subject,))  
 else:  
 cur.execute("SELECT \* FROM questions ORDER BY RANDOM() LIMIT 1")  
 q = cur.fetchone()  
 if not q:  
 con.close(); return None  
 q = dict(q)  
 if q["type"] == "SC":  
 cur.execute("SELECT idx, text FROM choices WHERE question\_id=? ORDER BY idx", (q["id"],))  
 q["options"] = cur.fetchall()  
 elif q["type"] == "MA":  
 cur.execute("SELECT left\_text, right\_text, pair\_key FROM match\_pairs WHERE question\_id=?", (q["id"],))  
 pairs = cur.fetchall()  
 left = [{"text": p["left\_text"], "key": p["pair\_key"]} for p in pairs]  
 right = [{"text": p["right\_text"], "key": p["pair\_key"]} for p in pairs]  
 random.shuffle(right)  
 q["left"], q["right"] = left, right  
 con.close()  
 return q  
  
@app.route("/quiz", methods=["GET","POST"])  
def quiz():  
 feedback = None  
 if request.method == "POST" and session.get("q"):  
 q = session["q"]  
 qt = q["type"]  
 session["total"] = session.get("total",0) + 1  
  
 if qt == "SC":  
 chosen = int(request.form["answer"])  
 if chosen == q["correct\_index"]:  
 session["score"] = session.get("score",0) + 1  
 feedback = ("✅ Правильно!", True)  
 else:  
 feedback = (f"❌ Неправильно. Правильна: {q['correct\_index']}", False)  
  
 elif qt == "OA":  
 user = request.form.get("answer","").strip().lower().replace(" ", "")  
 user = user.replace(",", ".")  
 valids = [v.strip().lower().replace(" ", "").replace(",", ".") for v in (q["correct\_text"] or "").split(";")]  
 if user and user in valids:  
 session["score"] = session.get("score",0) + 1  
 feedback = ("✅ Правильно!", True)  
 else:  
 feedback = (f"❌ Неправильно. Допустимі варіанти: {q['correct\_text']}", False)  
  
 elif qt == "MA":  
 correct = 0  
 total\_pairs = len(q["left"])  
 for i, left in enumerate(q["left"]):  
 chosen\_key = request.form.get(f"match\_{i}")  
 if chosen\_key == left["key"]:  
 correct += 1  
 if correct == total\_pairs:  
 session["score"] = session.get("score",0) + 1  
 feedback = (f"✅ Усі пари правильні ({correct}/{total\_pairs})", True)  
 else:  
 feedback = (f"ℹ️ Частково правильно: {correct}/{total\_pairs})", False)  
  
 subject = session.get("subject")  
 q = get\_random\_question(subject)  
 session["q"] = q  
 stats = {"score": session.get("score",0), "total": session.get("total",0)}  
 return render\_template("quiz.html", q=q, feedback=feedback, stats=stats)

(Файл: app.py (оновлено))

## 5) Оновлення шаблону quiz.html (умови на тип питання)

### templates/quiz.html (фрагмент)

<!doctype html>  
<html lang="uk">  
<meta charset="utf-8">  
<title>НМТ — Тренажер</title>  
<link rel="stylesheet" href="{{ url\_for('static', filename='styles.css') }}">  
<body class="container">  
 <h1>НМТ — Тренажер</h1>  
 <p>Рахунок: {{ stats.score }}/{{ stats.total }}</p>  
  
 {% if feedback %}  
 <div class="{{ 'alert-ok' if feedback[1] else 'alert-bad' }}">{{ feedback[0] }}</div>  
 {% endif %}  
  
 {% if q %}  
 <div class="card">  
 <h2>{{ q.text }}</h2>  
  
 {% if q.type == 'SC' %}  
 <form method="post">  
 {% for opt in q.options %}  
 <button class="btn" name="answer" value="{{ opt['idx'] }}">  
 <b>{{ opt['idx'] }}.</b> {{ opt['text'] }}  
 </button>  
 {% endfor %}  
 </form>  
  
 {% elif q.type == 'OA' %}  
 <form method="post">  
 <input class="input" type="text" name="answer" placeholder="Ваша відповідь">  
 <button class="btn primary">Перевірити</button>  
 </form>  
  
 {% elif q.type == 'MA' %}  
 <form method="post">  
 {% for i, left in enumerate(q.left) %}  
 <div class="row">  
 <span class="pill">{{ i+1 }}</span> {{ left.text }}  
 <select name="match\_{{ i }}" class="select">  
 <option value="">—</option>  
 {% for r in q.right %}  
 <option value="{{ r.key }}">{{ r.text }}</option>  
 {% endfor %}  
 </select>  
 </div>  
 {% endfor %}  
 <button class="btn primary" style="margin-top:10px">Перевірити</button>  
 </form>  
 {% endif %}  
 </div>  
 {% else %}  
 <p>Питань поки немає.</p>  
 {% endif %}  
</body>  
</html>

(Файл: templates/quiz.html)

## 6) Трохи стилів для MA/OA

### static/styles.css (доповнення)

input.input{width:100%;padding:10px;border:1px solid #cbd5e1;border-radius:8px;margin:8px 0}  
.select{margin-left:10px;padding:6px}  
.row{display:flex;align-items:center;gap:10px;margin:6px 0}  
.pill{display:inline-block;background:#e5e7eb;border-radius:999px;padding:2px 8px;font-weight:700}

(Файл: styles.css)

## 7) Команди запуску імпорту OA/MA

### Запуск імпорту

Windows (PowerShell):  
py import\_oa\_ma.py  
  
Mac / Linux (Terminal):  
python3 import\_oa\_ma.py

(Файл: термінал)

Після імпорту перезапустіть Flask-додаток. Тепер у тренажері випадково з'являтимуться питання всіх трьох типів: SC, OA, MA.