

## **Feladat: CRUD webalkalmazás (MySQL + reszponzív felület)**

### **1) Adatbázis létrehozása (MySQL)**

Hozd létre az alábbi adatbázist:

- Név: plants\_and\_animals
- Kolláció (collation): utf8mb4\_unicode\_ci

### **2) Tábla létrehozása**

Az adatbázison belül hozz létre 1 táblát a kiosztott témához tartozó néven.

A tábla kötelező mezői:

- id      azonosító (autoincrement pozitív, nem lehet null)
- created\_at    automatikusan megkapja a rekord létrehozásának idejét
- modified\_at   módosításkor frissüljön a rekord módosításának ideje, lehet null is.

A tábla tartalmazzon ezen felül a feladatban leírt ajánlott mezőket, amelyek alapján szűrést (filter/szelekció) lehet végezni.

A javasolt mezőnevektől el lehet térni, de a mezők típusa illeszkedjen a tárolt adathoz.

### **3) Megjelenítés (reszponzív weboldal)**

Készíts egy reszponzív weboldalt, ahol a tábla rekordjai megjelennek.

- A megjelenítés formája szabadon választható: táblázat / kártyák / lista stb.
- A megjelenítésből ki kell hagyni az alábbi mezőket:  
id, created\_at, modified\_at
- A következő műveleteknek a rekordhoz kötve kell megjeleníteniük:
  - Módosítás (Update)
  - Törlés (Delete)
- A következő funkciók legyenek kiemelten kezelve (jól látható, külön rész/gomb):
  - Új rekord felvétele (Create)
  - Szűrés (Filter / Szelekció)

### **4) Kötelező funkciók (CRUD)**

A táblán meg kell valósítani az összes CRUD műveletet:

1. Lekérdezés (Read) – összes rekord listázása
2. Lekérdezés szűréssel (Read + filter) – legalább 1–2 mező alapján szűrhető lista

3. Új rekord (Create) – űrlapról rögzítés
4. Módosítás (Update) – kiválasztott rekord szerkesztése
5. Törlés (Delete) – kiválasztott rekord törlése

Visszajelzések a felhasználónak

- Minden CRUD művelet után jelezd a felhasználónak, hogy sikeres volt-e.
- Sikertelenség esetén jelenjen meg egy rövid ok (pl. hibás adat, nem elérhető szerver, validációs hiba).

Megerősítés (biztonság)

- Update és Delete esetén a művelet előtt rá kell kérdezni: „Biztosan ezt akarja?”
- Egyszerű megoldás is elfogadott: alert() / confirm()

Űrlap kezelése

- Az űrlap lehet: Beágyazott form, vagy form egy modal ablakban.
- Fontos: a felhasználónak legyen lehetősége mégsem (mégse ment / mégse töröl).
- Bármilyen művelet indításakor meg kell akadályozni a párhuzamos indítást!

## 5) Használható technológiák (kötelező választás)

Minden kategóriából pontosan egyet kell választani:

Megjelenítés (UI): Bootstrap vagy Tailwind (minimális saját CSS megengedett)

Frontend: AngularJS / Angular / Vue / React / Svelte

Backend: PHP / Laravel / Express / SvelteKit / Python

### ELVÁRÁS:

Készíts igényes, átlátható, látványos felületet (reszponzív, rendezett elrendezés, használható gombok, egységes stílus). Tisztakód betartása. Hasonló munkáktól határolódj el! AI használata szigorúan tilos!

<b>ÉRTÉKELÉS:</b>	0 - 49%	elégtelen (1)
	50 - 62%	elégséges (2)
	63 - 75%	közepes (3)
	76 - 88%	jó (4)
	89 - 100%	jeles (5)