

# WEBPROGRAMOZÁS III. GY.

(LBT\_IM744G2)

Beadandó feladat dokumentáció

Kovács Norbert  
ANOXWJ

2021. június 6.

# Tartalomjegyzék

<b>1. Weboldal koncepciója</b>	<b>3</b>
1.1. Technikai részletek . . . . .	3
1.2. Elképzelés . . . . .	3
<b>2. Adatbázis tervezése</b>	<b>4</b>
2.1. ER modell . . . . .	4
2.2. Relációs adatmodell . . . . .	5
2.3. SQL parancsok . . . . .	5
2.3.1. Táblák létrehozása . . . . .	5
<b>3. Összefoglaló</b>	<b>5</b>

# 1. Weboldal koncepciója

## 1.1. Technikai részletek

A beadandó feladatot, az órán is alkalmazott *xampp* szerverrel kívánom megvalósítani, *apache netbeans* fejlesztőkörnyezettel. A Mysql kezelését a videóban is használt, *phpmyadmin* oldalon intézem. Nem fontos részlet, a megvalósítás szempontjából, de a Windows operációs rendszer, virtuális gépként fut.

(Követelmény: A beadandó feladat PHP MVC kereterendszerben készül!)

- OS: Windows 10 Pro (21H1)
- Kernel: 10.0.19041.1023
- Mysql: 10.4.19-MariaDB
- PHP: 8.0.6
- Apache Netbeans IDE: 12.3
- CodeIgniter: 3.1.11

## 1.2. Elképzelés

A weboldal egy *fiktív* szervezet köré épül, aminek a feladata az *állategészségügy*, és *állatvédelem*. A fiktív szervezetnek több városban lesznek épületei országszerte. Egy városban több különféle épület is tartozhat a szervezethez. Például tartozhat állatmenhely, állatorvosi rendelő, raktárpépület stb. Ezeknek az épületeknek vannak hozzájuk tartozó egységeik. Egy menhely állatok tárolására alkalmas kennelekkel, egy rendelő vizsgálóval, váróval, egy raktár szobákkal, szektorokkal rendelkezik stb.

Hasonlóképpen az egyetemi példafeladathoz, ahol

Kampusz -> Épület -> Terem

Pl.: Eger -> C épület -> 124-es terem,

felosztás található, itt is három rétegre kívánom felosztani a problémát, mégpedig:

Telep -> Épület -> 'Szoba'

Pl.: Miskolc -> Iroda -> 1-es iroda

vagy

Eger -> Egri állatmenhely -> 3-as Kennel.

A weboldal célja, tehát hogy a szervezet irányítani tudja a különböző városaiban található épületcsoportokat, pénzmozgásokat, alkalmazottakat stb. A fiktív szervezetet, az előző beadandóm állatotthonából fejleszttem tovább, részben idő spórolás céljából, részben a hasonló koncepcióból adódóan. Szervezet neve: **Kék mancs**.

Az *elképzelés* szerint, a felhasználókat több rétegben kezeljük majd. A weboldal nem a nyilvánosság felé közvetített oldal, hanem belső rendszerért felelős. Ettől függetlenül szükség lesz, telepenként admin, és felhasználói felületre.

## 2. Adatbázis tervezése

Szükségünk lesz, egy **épület** entitásra, egy '**szoba**' entitásra. Ennek a megnevezése, nem teljesen fedi le a pontos entitást, hiszen minden épületben ez mást reprezentál, azonban szükséges eltárolnunk, hogy melyik állat melyik kennelhez van rendelve, vagy hogy melyik alkalmazott melyik irodában dolgozik stb. Szükségünk lesz egy **állat** entitásra is, hogy eltároljuk a különféle állatokat, amik a szervezethez köthetők. Tárolnunk kell az **alkalmazottakat** is, és az **örökbefogadók** személyeit. Amennyiben lesz elég idő a fejlesztés során, az alkalmazottak **képzettségi mátrixát** is szeretném tárolni, ami egy gyakorlatból átemelt problémára nyújt megoldást.

*(A megvalósítandó projekthez tartozó adatmodell legalább 5 forgalmi adattáblát tartalmaz (a külső modulok által használt táblákon kívül).)*

### 2.1. ER modell

Az ER modell tervezése közben vegyük szemügyre nagyobb részletességgel az entitásokat, majd szemléltessük grafikusán. A grafikus szemléltetésre a *Draw.io* asztali alkalmazását használom.

#### Épület

Az **Épület** entitás kapcsán, tárolnunk kell a *megnevezését*, a *várost*, és félreértések elkerülése érdekében a *típusát*. Általánosságban kijelenthetjük, hogy egy városban, több azonos típusú épület is előfordulhat (két iroda, két raktár), viszont ugyan abban a városban, két azonos megnevezésű épület nem lehet. Ettől függetlenül, az egyszerűség kedvéért egy *azonosító* mezővel fogjuk ellátni az épületeket. A következőkben részletezett **szoba** entitás minden esetben csak egy épülethez tartozik.

*Egy szoba, csak egy épülethez tartozhat, de egy épülethez több szoba (1:N)  
Egy épülethez, több alkalmazott, de egy alkalmazott, egy épülethez rendelt. (1:N)*

#### Szoba

Egy **szoba**, egy épület alegysége. Tárolnunk kell a *kapacitását*. Attól függetlenül, hogy egy épület meghatározza, hogy milyen típusú szobákkal rendelkezik, ettől függetlenül, lehetőséget kell biztosítanunk a variációknak. Például egy Állatotthon is rendelkezhet kenneleken kívül váróval stb. Ezért tárolnunk kell a szoba *típusát* is.

*Egy szobához több állat/alkalmazott tartozhat, de egy állat/alkalmazott csak egy szobához. (1:N)*

#### Állat

Egy állat esetén, tárolnunk kell a *nevét*, *típusát*, *Születési évét* és *nemét*. Ezenkívül kap egy azonosítómezőt, és el kell tárolnunk melyik szobához tartozik.

#### Alkalmazott

Az alkalmazottak esetén, tároljuk el az alkalmazott teljes *nevét*, *születési dátumát*, alkalmazotti *jogviszonyát*, weboldalon használható *felhasználónevét*, *jelszavát*, és az *épületet* amihez kirendelt.

#### Örökbefogadó

Egy örökbefogadó személy kapcsán, el kell tárolnunk az *nevét*, *életkorát*, *lakhelyét*, jelenleg birtokolt *háziállatok számát*, és hogy *javasolt-e örökbefogadásra*.

## 2.2. Relációs adatmodell

## 2.3. SQL parancsok

### 2.3.1. Táblák létrehozása

## 3. Összefoglaló