

SZAKDOLGOZAT

Boda Gábor Máté

Gollmer Henrik

Nagy Zsolt

Szoftverfejlesztő és tesztelő

2024-2025

**VÁCI SZAKKÉPZÉSI CENTRUM
BORONKAY GYÖRGY MŰSZAKI
TECHNIKUM ÉS GIMNÁZIUM**



MenhelyMágus

Készítette:

Boda Gábor Máté

Konzulens:

Gollmer Henrik

Gyombolai Ferenc Péter

Nagy Zsolt

1. Hallgatói nyilatkozat

Alulírottak, ezúton kijelentjük, hogy a szakdolgozat saját, önálló munkánk, és korábban még sehol nem került publikálásra. Szakdolgozatunk a Váci Szakképzési Centrum Boronkay György Műszaki Technikum és Gimnázium Szoftverfejlesztő és tesztelő technikus képzésén készítettük. Tudomásul vesszük, hogy szakdolgozatunkat a Váci Szakképzési Centrum Boronkay György Műszaki Technikum és Gimnázium tárolja.

.....
Boda Gábor Máté

.....
Gollmer Henrik

.....
Nagy Zsolt



2. Konzultációs lap

Vizsgálók neve:

Szakdolgozat címe:

Program által nyújtott szolgáltatások:

- Különböző menhelyek állatainak központi nyilvántartása
- Felhasználók számára legideálisabb állat megkeresése

Sorszám	A konzultáció időpontja	A konzulens aláírása
1.	2024. október 11.	
2.	2024. november 15.	
3.	2024. december 13.	
4.	2025. január 17.	
5.	2025. február 14.	
6.	2025. március 14.	

A szakdolgozat beadható:

A szakdolgozatot átvettem:

Vác, 2025.

Vác, 2025.

.....

Konzulens

.....

A szakképzést folytató
intézmény felelőse



3. Tartalomjegyzék

1. Hallgatói nyilatkozat.....	3
2. Konzultációs lap	4
3. Tartalomjegyzék.....	5
4. Témaválasztás	9
4.1. Kezdeti ötletelések.....	9
4.2. Állatmenhely	9
4.3. Való életbeli probléma megoldása	9
4.4. Nem állatmenhely vagyunk.....	10
4.5. Miért hasznos?	10
5. Munkamegosztás, tervek, spec.	11
5.1. Munkamegosztás	11
5.2. Fejlesztési időhatárokkal kapcsolatos tervek	12
5.3. Feladatspecifikáció	12
6. Fejlesztői dokumentáció	15
6.1. Fejlesztői környezet	15
6.1.1. Visual Studio.....	15
6.1.2. Visual Studio Code.....	15
6.1.3. XAMPP	15
6.1.4. Paint.NET	16
6.1.5. GitHub	16
6.1.6. phpMyAdmin	16
6.1.7. Google Chrome Developer Console	17
6.2. Használt prog. nyelvek, technológiák	18
6.2.1. HTML	18
6.2.2. CSS	18



6.2.3.	JavaScript	18
6.2.4.	PHP	19
6.2.5.	C#.....	19
6.2.6.	WPF.....	19
6.2.7.	JSON	20
6.3.	Adatbázis	21
6.3.1.	Adatbázissal kapcsolatos információk.....	21
6.3.2.	Adatbázis táblái.....	23
6.4.	Szerepkörök és jogosultságaik	42
6.4.1.	Fiók nélküli látogatók – vendégek.....	42
6.4.2.	Felhasználók.....	42
6.4.3.	Alkalmazottak	43
6.4.4.	Adminok.....	43
6.5.	Alkalmazások felépítése, szerkezete, felületek bemutatása 45	
6.5.1.	Szétválasztás.....	45
6.5.2.	PHP apiszerver.....	45
6.5.3.	Webalkalmazás	46
6.5.4.	Asztali alkalmazás	47
6.6.	Funkciók bemutatása	50
6.6.1.	Fiókkezelés.....	50
6.6.2.	Állatok megtekintése	51
6.6.3.	Alkalmazott eszközök.....	52
6.6.4.	Admin eszközök	52
6.6.5.	Felhasználói eszközök (Web)	53



6.7.	Osztályok kapcsolati diagramja	55
6.7.1.	PHP apiszerver	55
6.7.2.	C# Asztali alkalmazás	56
6.8.	Input adatok helyességének ellenőrzése, megkötések	57
6.8.1.	Nem megfelelő formátum	57
6.8.2.	Használatban lévő E-mail cím.....	57
6.8.3.	Kitöltetlen beviteli mezők	58
6.9.	Biztonság	59
6.9.1.	Hitelesítés	59
6.9.2.	Jelszó titkosítás	61
7.	Felhasználói dokumentáció.....	63
7.1.	Rendszerkövetelmények	63
7.1.1.	Webes felület	63
7.1.2.	Asztali alkalmazás	63
7.1.3.	Kiszolgáló	63
7.2.	Telepítési útmutató	64
7.2.1.	Asztali alkalmazás	64
7.2.2.	Szerver	64
7.3.	Webes felület gyakorlati bemutatása	65
7.3.1.	Webes felület	65
7.3.2.	Asztali alkalmazás	71
8.	Tesztelés	76
8.1.	Új admin hozzáadása	76
8.2.	Bejelentkezés.....	77
8.3.	Menhelymódosítás	77
8.4.	Jogosultsághiány	79



8.4.1. WEB	79
8.4.2. Postman.....	79
9. Konklúziók.....	81
9.1. Tanultak.....	81
9.2. Továbbfejlesztési irányok	81
9.2.1. Új opciók hozzáadása:	81
9.2.2. Adatbázis lefordítása:	81
9.2.3. Jobb képfeltöltés:	81
9.2.4. Örökbefogadások megtekintése:	82
10. Irodalomjegyzék/Használt oldalak.....	83
10.1. Könyvek	83
10.2. Használt oldalak	83
11. Mellékletek	84



4. Témaválasztás

4.1. Kezdeti ötletelések

A legelején az alapötlet, amiben mind meg tudtunk egyezni, az az volt, hogy valamilyen webshop féleséget szeretnénk csinálni: lehessen böngészni különböző termékek között, ezeket lehessen megvenni. Az egyetlen probléma ezzel csak az volt, hogy a konkrét webshop témában nem tudtunk megegyezni.

4.2. Állatmenhely

Körülbelül 1 órányi ötletelés után az egyikőnk felvetette az ötletet, hogy az oldalunk lehetne egy állatörökbefogadó oldal. Tehát az oldalunkon valóban lehet böngészni és lényegében „vásárolni” is. A böngészendő elemek viszont azok az állatok lesznek, amelyeket nem vásárolni, hanem örökbe lehet fogadni az oldalra tévedő felhasználóknak.

4.3. Való életbeli probléma megoldása

A vizsgaremek feladat ismertetésekor azt hallottuk, hogy annak, amit csinálunk hasznosnak kell lennie, és egy valós életbeli problémát kell megoldania.

Bár az az ötlet, hogy mi egy állatmenhely leszünk nagyon tetszett mindannyiunknak, nem éreztük azt, hogy ez az ötlet eléggé egyedi, és eléggé hasznos lenne az emberek számára. Az állatmenhelyek remekek, de így is van belőlük rengeteg, mi ettől valami egyedibbet szerettünk volna csinálni, szóval tovább ötleteltünk.



4.4. Nem állatmenhely vagyunk

Az idő teltével, és rengeteg közös ötletelés után eldöntöttük, hogy a vizsgáremekünkben nem egy állatmenhelyet fogunk csinálni, de az alap témát nem változtattuk meg, maradtunk az állat örökbefogadásnál, csak egy kis csavarral.

Projektünk egy olyan cég, amely állatmenhelyeket köt össze. Weboldalunk és asztali alkalmazásunk nem a saját állatainkat jelenítik meg, hanem más magyarországi menhelyek saját állatait.

4.5. Miért hasznos?

Internetes utánanéзések, és kutakodások után felfedeztük, hogy rengeteg olyan menhely van Magyarországon, amelyek bár rendelkeznek telephellyel, de nincsen nekik saját weboldaluk. Legtöbb esetben csak Facebook oldallal rendelkeznek, viszont ott természetesen nincsenek kilistázva az adott menhely állatai. Emiatt a kelleténél több időt kell egy érdeklődőnek befektetnie a számára tökéletes állat megtalálásához.

Mi egy helyen jelenítjük meg partner menhelyeink állatait, és az alkalmazottjaik számára saját fiókokat biztosítunk, amellyel tudják kezelni az állataikat, illetve az állataikhoz tartozó örökbefogadási kérelmeket.



5. Munkamegosztás, tervek, spec.

5.1. Munkamegosztás

Az elvégzésre váró feladatokat megpróbáltuk igazságosan (egyenlő módon) felosztani egymás között. A munka során mind ugyanannyit dolgoztunk, nem fordult elő, hogy valaki kiemelkedően sokat dolgozott volna a többiekhez képest.

Boda Gábor Máté kapta a **backend** megírásának feladatát. Ez tartalmazta az adatbázishoz való kapcsolódás megvalósítását, az összes repository, az autentikáció és a REST API megírását.

Gollmer Henrik vállalta el az **asztali alkalmazást**. Ez tartalmazta egy esztétikus, könnyen használható asztali alkalmazás megtervezését, és annak minden funkciójának megvalósítását.

Nagy Zsolt feladata lett a **Webalkalmazás** megírása. Megtervezte a weboldal kinézetét, felépítését, elrendezését, minden funkcióját megírta, mindezt olyan módon, hogy szinte bármely kijelző méreten is (350 pixel felett) szép lett az oldal.

Máté, Henrik és Zsolt is közösen dolgoztak az **adatbázis** tervezésében, kialakításában, és annak adatokkal történő feltöltésében. Közös munka szüleménye a teljes **dokumentáció**, és a **PowerPoint prezentációnk** is.

Természetesen, amikor valamelyik egyéni feladat problémát okozott az egyikünknek, akkor közösen ötleteltünk rajta, és próbáltuk megoldani, addig a saját, aktuális munkánkat szünetelve.



5.2. Fejlesztési időhatárokkal kapcsolatos tervek

A legfőbb cél nem más volt minthogy februárban már teljesen kész legyünk a három fő egyéni feladattal (backend, web, asztali), annak érdekében, hogy onnantól kezdve el tudjunk kezdeni dolgozni a dokumentációnkon, meg azzal párhuzamosan kisebb hibákat keresni, és azokat kijavítani.

Szerencsére a tervünk sikerült, februárban készen lettünk az előbb felsorolt feladatokkal.

A leghatékonyabb hibakeresés a befejezés utáni időszakban lépett fel: számtalan apró hibát találtunk, amelyet addig nem vettünk észre, és azokat gyorsan kijavítottuk közösen.

Egynéhány apró funkció is ebben az időszakban jutott eszünkbe, amelyek még nem voltak benne a projektben, ezeket is ekkor implementáltuk.

5.3. Feladatspecifikáció

Backend:

- Meg kell valósítani autentikációt, hogy mindenki csak azt tudja elvégezni, amire neki jogosultsága van. 3 féle fiók fajta létezik, mindnek máshoz van joga.
 - Admin
 - Alkalmazott
 - Sima felhasználó
- Az állatok adatai egy központi szerveren legyenek eltárolva, hogy minden felületről elérhetők legyenek. Ehhez a szerverhez tudjunk csatlakozni, és kommunikálni vele.
- REST API interfész használata
 - GET: adatlekérés
 - POST: adatfeltöltés



- PUT: adatmódosítás
- DELETE: adattörlés

Webalkalmazás:

- Fiókkezelés
 - Bejelentkezés
 - Regisztráció (felhasználóként)
 - Személyes adatok módosítása
 - Fióktörlés
- Állatokat lehessen megtekinteni
 - Mozaik nézetben és egyenként is
 - Felhasználói fiókról örökbe lehessen fogadni, és állatot kedvencek közé adni.
- Bármilyen típusú fiókkal be lehessen lépni.
- A felhasználó beállíthassa a preferenciáit az állattal kapcsolatban.
 - A felhasználó tudja kilistázni az állatokat, és azok legyenek előrébb feldobva, amelyek a legmegfelelőbbek a számára.
 - A listázás alapértelmezetten a preferenciák alapján történjen (a szűrő értékei úgy állítódnak be, hogy igazodjanak a preferenciákhoz), de a preferenciáit keresés közben a felhasználó módosíthassa (frissül a szűrő, és a kilistázott állatok), és ha úgy kívánja, akkor elmentheti új preferenciáját (felülírja a régebbit).
- Biztosítson lehetőséget az alkalmazottaknak az állatok és az örökbefogadások kezelésére.
- Biztosítson lehetőséget az adminoknak az alkalmazottak, a menhelyek, illetve a felhasználók kezelésére.



– A nyelv legyen választható

- Magyar
- Angol
- Német

– Legyen több téma

- Világos téma
- Sötét téma

Asztali alkalmazás:

– Kizárólag adminisztrációs célokra szolgál, tehát csak adminisztrációs jogokkal rendelkező fiókkal lehessen belépni, azaz admin, illetve alkalmazotti fiókkal.

– Fiókkezelés

- Bejelentkezés
- Regisztráció (alkalmazottként)
- Személyes adatok módosítása
- Fióktörlés

– Az alkalmazottaknak biztosítson lehetőséget az állatok és az örökbefogadások kezelésére.

– Az adminoknak biztosítson lehetőséget az alkalmazottak, a menhelyek, illetve a felhasználók kezelésére.



6. Fejlesztői dokumentáció

6.1. Fejlesztői környezet

6.1.1. Visual Studio

A Visual Studio egy fejlesztői környezet (IDE), amelyet elsősorban .NET alkalmazások fejlesztésére használnak. Támogatja a C#, C++, Visual Basic, F# és más nyelveket, így rendkívül sokoldalú. A program kód szerkesztését, hibakeresést, tesztelést és verziókezelést tesz lehetővé, mindezt integrált eszközökkel. Az IDE a Microsoft által kifejlesztett, és gyakran használják vállalati szintű alkalmazások fejlesztésére, webfejlesztésre és mobilalkalmazások készítésére is. Emellett plugin-ok és kiegészítők révén könnyen testre szabható.

6.1.2. Visual Studio Code

A Visual Studio Code (VS Code) egy könnyű, nyílt forráskódú kódszerkesztő, amely különböző programozási nyelvekhez (például JavaScript, PHP, C++) kínál támogatást. Bár nem teljes értékű IDE, számos hasznos funkcióval rendelkezik, mint például hibakeresés, verziókezelés (Git integrációval), kódkiegészítés, és szintaxis kiemelés. A kódszerkesztő moduláris, így a felhasználók különböző bővítményeket telepíthetnek, hogy testre szabják a munkakörnyezetüket. A Visual Studio Code gyors és ideális könnyű vagy közepes méretű fejlesztési projektekhez.

6.1.3. XAMPP

A XAMPP egy ingyenes, nyílt forráskódú fejlesztői környezet, amely lehetővé teszi, hogy könnyen telepítsünk egy Apache szervert, MySQL adatbázist, PHP-t és Perl-t egy Windows, Linux vagy Mac gépre. Elsősorban webfejlesztéshez használják, mivel egyszerűsíti a helyi



szerver környezet létrehozását. A XAMPP ideális azok számára, akik webes alkalmazásokat fejlesztenek, mivel gyorsan beállíthatják a szükséges komponenseket anélkül, hogy manuálisan konfigurálniuk kellene a szerveret.

6.1.4. Paint.NET

A Paint.NET egy ingyenes, könnyen használható képszerkesztő program, amely elsősorban Windows operációs rendszeren fut. Bár nem rendelkezik olyan fejlett funkciókkal, mint az Adobe Photoshop, a Paint.NET elegendő eszközt biztosít az alapvető képszerkesztési feladatokhoz, például fényképek vágásához, színek módosításához, rétegek kezeléséhez és effektek hozzáadásához. A program egyszerűségének és gyorsaságának köszönhetően ideális azok számára, akik gyors és könnyű képszerkesztési megoldást keresnek.

6.1.5. GitHub

A GitHub egy webes platform, amely a Git verziókezelő rendszert használja kódok tárolására, kezelésére és megosztására. Fejlesztők és csapatok használják, hogy közösen dolgozhassanak projekteken, és könnyen nyomon követhessék a kódváltoztatásokat. A GitHub lehetővé teszi a kód verzióinak nyilvántartását, a hibajegyek és feladatok kezelését, valamint a dokumentációk tárolását.

6.1.6. phpMyAdmin

A phpMyAdmin egy webalapú alkalmazás, amelyet MySQL és MariaDB adatbázisok kezelésére használnak. A felhasználók könnyen végezhetnek adatbázis műveleteket, például táblák létrehozását, adatbevitelét, lekérdezések futtatását és adatbázis-migrációkat. A phpMyAdmin grafikus felhasználói felületet biztosít, amely leegyszerűsíti az adatbázis kezelési feladatokat, így nem szükséges parancssori ismeret.



6.1.7. Google Chrome Developer Console

A Google Chrome Developer Console a Google Chrome böngésző fejlesztői eszköze, amely segít a webfejlesztőknek hibakeresésben, a weboldalak teljesítményének optimalizálásában, és a front-end fejlesztési munkák során. Az eszköz lehetővé teszi a HTML, CSS és JavaScript kódok gyors módosítását és tesztelését, valamint az oldalak hibáinak naplózását. Továbbá, a konzol segítségével nyomon követhetők a hálózati kérések, az alkalmazás teljesítménye, és különböző biztonsági beállítások is ellenőrizhetők.



6.2. Használt prog. nyelvek, technológiák

6.2.1. HTML

HTML (HyperText Markup Language) egy jelölőnyelv, amelyet weboldalak létrehozására használnak. A HTML lehetővé teszi a tartalom strukturálását a weboldalon, például címek, bekezdések, listák, képek és hivatkozások használatával. A dokumentum elemei "tagek" közé vannak ágyazva, mint például `<h1>`, `<p>`, ``, stb. A HTML önállóan nem hajt végre semmilyen műveletet, csupán a tartalom és struktúra meghatározására szolgál. A HTML-t más webfejlesztési technológiákkal, mint például CSS-sel és JavaScript-tel kombinálva lehet dinamikus és vizuálisan vonzó weboldalakot létrehozni.

6.2.2. CSS

CSS (Cascading Style Sheets) a weboldalak vizuális megjelenésének formázására szolgáló nyelv. A CSS segítségével meghatározhatjuk a színt, betűtípust, elrendezést, térközöket és sok más megjelenítési beállítást. A CSS lehetővé teszi, hogy a HTML elemeket szépen formázzuk, anélkül, hogy megváltoztatnánk azok struktúráját. A stíluslapokat külön fájlokban is tárolhatjuk, így a kód karbantartása és újrahasználata könnyebbé válik. Az alkalmazott stílusok az úgynevezett "cascading" elv szerint működnek, vagyis az alacsonyabb szintű szabályok felülírhatók magasabb szintű szabályokkal.

6.2.3. JavaScript

A JavaScript egy dinamikus, objektumorientált programozási nyelv, amelyet elsősorban weboldalak interaktivitásának megvalósítására használnak. Segítségével a weboldalak dinamikusan reagálhatnak a felhasználói interakciókra, például kattintásokra, űrlapok kitöltésére vagy akár időzített események kezelésére. A JavaScript képes DOM



manipulációra (a HTML dokumentum módosítása), eseménykezelésre és adatfeldolgozásra. A JavaScript a weboldalak front-end fejlesztésének alapvető eszköze.

6.2.4. PHP

A PHP (PHP: Hypertext Preprocessor) egy szerveroldali szkriptnyelv, amelyet dinamikus weboldalak és alkalmazások készítésére használnak. A PHP lehetővé teszi a weboldalak adatbázisokkal való interakcióját, felhasználói adatok feldolgozását, űrlapok kezelését és más dinamikus tartalmak generálását. A PHP-t a szerver futtatja, mielőtt a HTML kódot visszaküldi a böngészőnek, így a felhasználó számára statikusnak tűnhet a weboldal, de valójában dinamikusan generálódik.

6.2.5. C#

A C# egy modern, objektumorientált programozási nyelv, amelyet a Microsoft fejlesztett ki, és amelyet a .NET platformon való alkalmazásfejlesztéshez használnak. A C# erőteljes és rugalmas, amely lehetővé teszi az asztali alkalmazások, webalkalmazások és játékok fejlesztését. A nyelv szintaxisa tiszta és könnyen olvasható, és számos beépített funkcióval rendelkezik, mint például a memória kezelés és az objektum-orientált programozás támogatása. A C# széles körben használatos üzleti alkalmazások és vállalati rendszerek fejlesztésére.

6.2.6. WPF

A WPF (Windows Presentation Foundation) a Microsoft által kifejlesztett grafikus alrendszer, amelyet asztali alkalmazások fejlesztésére használnak. A rendszer XAML (Extensible Application Markup Language) alapú, amely egy deklaratív jelölőnyelv, és lehetővé teszi a felhasználói felület gyors tervezését. A WPF támogatja az animációkat, adatkötést, és különböző megjelenítési módokat, mint



például 2D és 3D grafikák, valamint multimédiás tartalmak integrálását.

6.2.7. JSON

A JSON (JavaScript Object Notation) egy könnyen olvasható és írható adatcsere formátum, amelyet széles körben használnak a webalkalmazások adatainak tárolására és továbbítására. A JSON egyszerű szintaxist használ, amely kulcs-érték párokból áll, így könnyen értelmezhető mind emberek, mind programok számára. A JSON-t gyakran alkalmazzák API-k, adatbázisok, konfigurációs fájlok és más adatcserét igénylő rendszerek esetén. Könnyen integrálható JavaScript-tel, de más programozási nyelvekben is támogatott.



6.3. Adatbázis

6.3.1. Adatbázissal kapcsolatos információk

Adatbázis neve: menhelymagus

Adatbázis illesztése: utf8mb4_general_ci

Adatbázismotor: InnoDB

- MySQL verzió: 15.1
- MariaDB verzió: 10.4.32

Adatbázist létrehozó SQL parancs:

```
CREATE DATABASE IF NOT EXISTS menhelymagus  
CHARACTER SET utf8mb4  
COLLATE utf8mb4_general_ci;
```





6.3.2. Adatbázis táblái

6.3.2.1. Az „admins” tábla

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett	Megjegyzések	Extra
<input type="checkbox"/>	1 id 	int(11)			Nem	Nincs		AUTO_INCREMENT
<input type="checkbox"/>	2 name	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/>	3 email	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/>	4 passwordHash	varchar(95)	latin2_hungarian_ci		Nem	Nincs		



A MenhelyMágushoz tartozó adminok adatait tárolja.

Táblához tartozó SQL parancs:

```
CREATE TABLE `admins` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `passwordHash` varchar(95) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;  
  
ALTER TABLE `admins` ADD PRIMARY KEY (`id`);
```



6.3.2.2. Az „adoptions” tábla

	#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett
<input type="checkbox"/>	1	animalId 	int(11)			Nem	Nincs
<input type="checkbox"/>	2	userId 	int(11)			Nem	Nincs
<input type="checkbox"/>	3	pending	tinyint(1)			Nem	Nincs

A menhelyek örökbefogadásait tartalmazza:

- örökbefogadott állat ID
- örökbefogadó felhasználó ID
- függőben van-e az örökbefogadás: 0 (false) vagy 1 (true)

Táblához tartozó SQL parancs:

```
CREATE TABLE `adoptions` (  
  `animalId` int(11) NOT NULL,  
  `userId` int(11) NOT NULL,  
  `pending` tinyint(1) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;  
  
ALTER TABLE `adoptions`  
  ADD KEY `animalId` (`animalId`,`userId`),  
  ADD KEY `userId` (`userId`);  
  
ALTER TABLE `adoptions`  
  ADD CONSTRAINT `adoptions_ibfk_1` FOREIGN KEY (`animalId`) REFERENCES `animals` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `adoptions_ibfk_2` FOREIGN KEY (`userId`) REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```



6.3.2.3. Az „ animals „ tábla

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett	Megjegyzések	Extra
<input type="checkbox"/>	1 id 🔑	int(11)			Nem	Nincs		AUTO_INCREMENT
<input type="checkbox"/>	2 shelterId 🔑	int(11)			Nem	Nincs		
<input type="checkbox"/>	3 birthDate	date			Nem	Nincs		
<input type="checkbox"/>	4 entryDate	date			Nem	Nincs		
<input type="checkbox"/>	5 speciesId 🔑	int(11)			Nem	Nincs		
<input type="checkbox"/>	6 name	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/>	7 gender	varchar(1)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/>	8 neutered	tinyint(1)			Nem	Nincs		
<input type="checkbox"/>	9 healthy	tinyint(1)			Nem	Nincs		
<input type="checkbox"/>	10 weight	double			Nem	Nincs		
<input type="checkbox"/>	11 cuteness	int(11)			Nem	Nincs		
<input type="checkbox"/>	12 childFriendliness	int(11)			Nem	Nincs		
<input type="checkbox"/>	13 sociability	int(11)			Nem	Nincs		
<input type="checkbox"/>	14 exerciseNeed	int(11)			Nem	Nincs		
<input type="checkbox"/>	15 furLength	int(11)			Nem	Nincs		
<input type="checkbox"/>	16 docility	int(11)			Nem	Nincs		
<input type="checkbox"/>	17 housebroken	tinyint(1)			Nem	Nincs		
<input type="checkbox"/>	18 description	text	latin2_hungarian_ci		Nem	Nincs		

A menhelyek összes állatát tartalmazza.

Táblához tartozó SQL parancs:

```
CREATE TABLE `animals` (
  `id` int(11) NOT NULL,
  `shelterId` int(11) NOT NULL,
  `birthDate` date NOT NULL,
  `entryDate` date NOT NULL,
  `speciesId` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `gender` varchar(1) NOT NULL,
  `neutered` tinyint(1) NOT NULL,
  `healthy` tinyint(1) NOT NULL,
  `weight` double NOT NULL,
```



```
`cuteness` int(11) NOT NULL,  
`childFriendliness` int(11) NOT NULL,  
`sociability` int(11) NOT NULL,  
`exerciseNeed` int(11) NOT NULL,  
`furLength` int(11) NOT NULL,  
`docility` int(11) NOT NULL,  
`housebroken` tinyint(1) NOT NULL,  
`description` text NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;
```

```
ALTER TABLE `animals`  
  
  ADD PRIMARY KEY (`id`),  
  
  ADD KEY `shelterId` (`shelterId`,`speciesId`),  
  
  ADD KEY `speciesId` (`speciesId`);
```

```
ALTER TABLE `animals`  
  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=292;
```

```
ALTER TABLE `animals`  
  
  ADD CONSTRAINT `animals_ibfk_1` FOREIGN KEY (`speciesId`) REFERENCES  
  `species` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,  
  
  ADD CONSTRAINT `animals_ibfk_2` FOREIGN KEY (`shelterId`) REFERENCES  
  `shelters` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```



6.3.2.4.A „breedjunction” tábla

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett
<input type="checkbox"/> 1	animalId 	int(11)			Nem	Nincs
<input type="checkbox"/> 2	breedId 	int(11)			Nem	Nincs



Az állatokhoz tartozó fajtáknak a kapcsolótáblája: összeköti az állatot a fajtával/fajtákkal.

Táblához tartozó SQL parancs:

```
CREATE TABLE `breedjunction` (  
  `animalId` int(11) NOT NULL,  
  `breedId` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;  
  
ALTER TABLE `breedjunction`  
  ADD KEY `animalId` (`animalId`,`breedId`),  
  ADD KEY `breedId` (`breedId`);  
  
ALTER TABLE `breedjunction`  
  ADD CONSTRAINT `breedjunction_ibfk_1` FOREIGN KEY (`breedId`)  
REFERENCES `breeds` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `breedjunction_ibfk_2` FOREIGN KEY (`animalId`)  
REFERENCES `animals` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```



6.3.2.5.A „breeds” tábla

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett	Megjegyzések	Extra
<input type="checkbox"/> 1	id 	int(11)			Nem	Nincs		AUTO_INCREMENT
<input type="checkbox"/> 2	name	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 3	speciesId 	int(11)			Nem	Nincs		

A fajtákat (ID + név), illetve az fajuknak az ID-jét tartalmazza.

Táblához tartozó SQL parancs:

```
CREATE TABLE `breeds` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  `speciesId` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;  
  
ALTER TABLE `breeds`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `speciesId` (`speciesId`);  
  
ALTER TABLE `breeds`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=76;  
  
ALTER TABLE `breeds`  
  ADD CONSTRAINT `breeds_ibfk_1` FOREIGN KEY (`speciesId`) REFERENCES  
  `species` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```



6.3.2.6. Az „ employees „ tábla

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett	Megjegyzések	Extra
<input type="checkbox"/> 1	id 🔑	int(11)			Nem	Nincs		AUTO_INCREMENT
<input type="checkbox"/> 2	name	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 3	email	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 4	passwordHash	varchar(95)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 5	shelterId 🔑	int(11)			Igen	NULL		



A menhelyekhez tartozó alkalmazottak adatait tartalmazza.

Táblához tartozó SQL parancs:

```
CREATE TABLE `employees` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `passwordHash` varchar(95) NOT NULL,  
  `shelterId` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;  
  
ALTER TABLE `employees`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `shelterId` (`shelterId`);  
  
ALTER TABLE `employees`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=19;  
  
ALTER TABLE `employees`  
  ADD CONSTRAINT `employees_ibfk_1` FOREIGN KEY (`shelterId`) REFERENCES `shelters` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```



6.3.2.7.A „favourites” tábla

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett
<input type="checkbox"/> 1	animalId 	int(11)			Nem	Nincs
<input type="checkbox"/> 2	userId 	int(11)			Nem	Nincs



A felhasználók kedvenc állatait tárolja.

Táblához tartozó SQL parancs:

```
CREATE TABLE `favourites` (  
  `animalId` int(11) NOT NULL,  
  `userId` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;  
  
ALTER TABLE `favourites`  
  ADD KEY `animalId` (`animalId`,`userId`),  
  ADD KEY `userId` (`userId`);  
  
ALTER TABLE `favourites`  
  ADD CONSTRAINT `favourites_ibfk_1` FOREIGN KEY (`userId`) REFERENCES  
  `users` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `favourites_ibfk_2` FOREIGN KEY (`animalId`)  
  REFERENCES `animals` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```



6.3.2.8.A „ habitatjunction ” tábla

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett
<input type="checkbox"/> 1	animalId 	int(11)			Nem	Nincs
<input type="checkbox"/> 2	habitatId 	int(11)			Nem	Nincs


Az állatokhoz tartozó élőhelyeknek a kapcsolótáblája: összeköti az állatot az élőhellyel/élőhelyekkel.

Táblához tartozó SQL parancs:

```
CREATE TABLE `habitatjunction` (  
  `animalId` int(11) NOT NULL,  
  `habitatId` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;  
  
ALTER TABLE `habitatjunction`  
  ADD KEY `habitatId` (`habitatId`),  
  ADD KEY `animalId` (`animalId`);  
  
ALTER TABLE `habitatjunction`  
  ADD CONSTRAINT `habitatjunction_ibfk_1` FOREIGN KEY (`habitatId`)  
REFERENCES `habitats` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `habitatjunction_ibfk_2` FOREIGN KEY (`animalId`)  
REFERENCES `animals` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```



6.3.2.9.A „habitats” tábla

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett	Megjegyzések	Extra
<input type="checkbox"/>	1 id 	int(11)			Nem	Nincs		AUTO_INCREMENT
<input type="checkbox"/>	2 name	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/>	3 description	varchar(255)	latin2_hungarian_ci		Nem	Nincs		


Az élőhelyeket (ID + név + leírás) tartalmazza.

Táblához tartozó SQL parancs:

```
CREATE TABLE `habitats` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  `description` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;  
  
ALTER TABLE `habitats`  
  ADD PRIMARY KEY (`id`);  
  
ALTER TABLE `habitats`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=10;  
  
ALTER TABLE `habitatjunction`  
  ADD CONSTRAINT `habitatjunction_ibfk_1` FOREIGN KEY (`habitatId`) REFERENCES `habitats` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `habitatjunction_ibfk_2` FOREIGN KEY (`animalId`) REFERENCES `animals` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```



6.3.2.10. A „ preferences ” tábla

	#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett
<input type="checkbox"/>	1	userId 	int(11)			Nem	Nincs
<input type="checkbox"/>	2	age	int(11)			Nem	Nincs
<input type="checkbox"/>	3	ageWeight	int(11)			Nem	Nincs
<input type="checkbox"/>	4	weight	int(11)			Nem	Nincs
<input type="checkbox"/>	5	weightWeight	int(11)			Nem	Nincs
<input type="checkbox"/>	6	cuteness	int(11)			Nem	Nincs
<input type="checkbox"/>	7	cutenessWeight	int(11)			Nem	Nincs
<input type="checkbox"/>	8	childFriendliness	int(11)			Nem	Nincs
<input type="checkbox"/>	9	childFriendlinessWeight	int(11)			Nem	Nincs
<input type="checkbox"/>	10	sociability	int(11)			Nem	Nincs
<input type="checkbox"/>	11	sociabilityWeight	int(11)			Nem	Nincs
<input type="checkbox"/>	12	exerciseNeed	int(11)			Nem	Nincs
<input type="checkbox"/>	13	exerciseNeedWeight	int(11)			Nem	Nincs
<input type="checkbox"/>	14	furLength	int(11)			Nem	Nincs
<input type="checkbox"/>	15	furLengthWeight	int(11)			Nem	Nincs
<input type="checkbox"/>	16	docility	int(11)			Nem	Nincs
<input type="checkbox"/>	17	docilityWeight	int(11)			Nem	Nincs

A felhasználók preferenciáit tartalmazza.

Táblához tartozó SQL parancs:



```
CREATE TABLE `preferences` (
  `userId` int(11) NOT NULL,
  `age` int(11) NOT NULL,
  `ageWeight` int(11) NOT NULL,
  `weight` int(11) NOT NULL,
  `weightWeight` int(11) NOT NULL,
  `cuteness` int(11) NOT NULL,
```



```
`cutenessWeight` int(11) NOT NULL,  
`childFriendliness` int(11) NOT NULL,  
`childFriendlinessWeight` int(11) NOT NULL,  
`sociability` int(11) NOT NULL,  
`sociabilityWeight` int(11) NOT NULL,  
`exerciseNeed` int(11) NOT NULL,  
`exerciseNeedWeight` int(11) NOT NULL,  
`furLength` int(11) NOT NULL,  
`furLengthWeight` int(11) NOT NULL,  
`docility` int(11) NOT NULL,  
`docilityWeight` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;  
  
ALTER TABLE `preferences`  
  ADD PRIMARY KEY (`userId`);  
  
ALTER TABLE `preferences`  
  ADD CONSTRAINT `preferences_ibfk_1` FOREIGN KEY (`userId`)  
  REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```



6.3.2.11. A „ provideablehabitats ” tábla

	#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett
<input type="checkbox"/>	1	userId 	int(11)			Nem	Nincs
<input type="checkbox"/>	2	habitatId 	int(11)			Nem	Nincs

Az felhasználókhöz tartozó biztosítható élőhelyeknek a kapcsolótáblája: összeköti az felhasználót az élőhellyel/élőhelyekkel.

Táblához tartozó SQL parancs:

```
CREATE TABLE `provideablehabitats` (  
  `userId` int(11) NOT NULL,  
  `habitatId` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;  
  
ALTER TABLE `provideablehabitats`  
  ADD KEY `userId` (`userId`,`habitatId`),  
  ADD KEY `habitatId` (`habitatId`);  
  
ALTER TABLE `provideablehabitats`  
  ADD CONSTRAINT `provideablehabitats_ibfk_1` FOREIGN KEY (`userId`)  
  REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `provideablehabitats_ibfk_2` FOREIGN KEY (`habitatId`)  
  REFERENCES `habitats` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```



6.3.2.12. A „shelters” tábla

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett	Megjegyzések	Extra
<input type="checkbox"/> 1	id 	int(11)			Nem	Nincs		AUTO_INCREMENT
<input type="checkbox"/> 2	name	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 3	city	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 4	street	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 5	houseNumber	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 6	telephoneNumber	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 7	email	varchar(255)	latin2_hungarian_ci		Nem	Nincs		

A MenhelyMágus partner menhelyeinek adatait tartalmazza.

Táblához tartozó SQL parancs:

```
CREATE TABLE `shelters` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  `city` varchar(255) NOT NULL,  
  `street` varchar(255) NOT NULL,  
  `houseNumber` varchar(255) NOT NULL,  
  `telephoneNumber` varchar(255) NOT NULL,  
  `email` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;  
  
ALTER TABLE `shelters`  
  ADD PRIMARY KEY (`id`);  
  
ALTER TABLE `shelters`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;
```



6.3.2.13. A „species” tábla

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett	Megjegyzések	Extra
<input type="checkbox"/> 1	id	int(11)			Nem	Nincs		AUTO_INCREMENT
<input type="checkbox"/> 2	name	varchar(255)	latin2_hungarian_ci		Nem	Nincs		

A fajokat (ID + név) tartalmazza.

Táblához tartozó SQL parancs:

```
CREATE TABLE `species` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;  
  
ALTER TABLE `species`  
  ADD PRIMARY KEY (`id`);  
  
ALTER TABLE `species`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=12;
```



6.3.2.14. A „tokens” tábla

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett	Megjegyzések	Extra
<input type="checkbox"/> 1	id 	int(11)			Nem	Nincs		AUTO_INCREMENT
<input type="checkbox"/> 2	userId 	int(11)			Nem	Nincs		
<input type="checkbox"/> 3	userType	varchar(255)	utf8mb4_general_ci		Nem	Nincs		
<input type="checkbox"/> 4	token	varchar(255)	utf8mb4_general_ci		Nem	Nincs		

A bejelentkezési tokenek adatait tartalmazza:

- Token ID
- Felhasználó ID (akihez a token tartozik)
- userType (megadja, hogy milyen típusú fiókhoz tartozik)
- Token (password_hash() használatával)

Táblához tartozó SQL parancs:

```
CREATE TABLE `tokens` (  
  `id` int(11) NOT NULL,  
  `userId` int(11) NOT NULL,  
  `userType` varchar(255) NOT NULL,  
  `token` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;  
  
ALTER TABLE `tokens`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `userId` (`userId`);  
  
ALTER TABLE `tokens`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=60;
```



6.3.2.15. A „ users ” tábla

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett	Megjegyzések	Extra
<input type="checkbox"/> 1	id 🔑	int(11)			Nem	Nincs		AUTO_INCREMENT
<input type="checkbox"/> 2	email	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 3	passwordHash	varchar(95)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 4	name	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 5	telephoneNumber	varchar(11)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 6	city	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 7	street	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/> 8	houseNumber	int(11)			Nem	Nincs		
<input type="checkbox"/> 9	wantsEmail	tinyint(1)			Igen	NULL		

A MenhelyMágusra regisztrált felhasználók adatait tartalmazza.

Táblához tartozó SQL parancs:

```
CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `email` varchar(255) NOT NULL,
  `passwordHash` varchar(95) NOT NULL,
  `name` varchar(255) NOT NULL,
  `telephoneNumber` varchar(11) NOT NULL,
  `city` varchar(255) NOT NULL,
  `street` varchar(255) NOT NULL,
  `houseNumber` int(11) NOT NULL,
  `wantsEmail` tinyint(1) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;

ALTER TABLE `users`
  ADD PRIMARY KEY (`id`);

ALTER TABLE `users`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;
```



6.3.2.16. A „vaccinejunction” tábla

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett
<input type="checkbox"/>	1 animalId 	int(11)			Nem	Nincs
<input type="checkbox"/>	2 vaccineId 	int(11)			Nem	Nincs

Az állatokhoz tartozó oltásoknak a kapcsolótáblája: összeköti az állatot az oltással/oltásokkal.

Táblához tartozó SQL parancs:

```
CREATE TABLE `vaccinejunction` (  
  `animalId` int(11) NOT NULL,  
  `vaccineId` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;  
  
ALTER TABLE `vaccinejunction`  
  ADD KEY `animalId` (`animalId`,`vaccineId`),  
  ADD KEY `vaccineId` (`vaccineId`);  
  
ALTER TABLE `vaccinejunction`  
  ADD CONSTRAINT `vaccinejunction_ibfk_1` FOREIGN KEY (`vaccineId`) REFERENCES `vaccines` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `vaccinejunction_ibfk_2` FOREIGN KEY (`animalId`) REFERENCES `animals` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```



6.3.2.17. A „vaccines” tábla

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett	Megjegyzések	Extra
<input type="checkbox"/>	1 id 	int(11)			Nem	Nincs		AUTO_INCREMENT
<input type="checkbox"/>	2 name	varchar(255)	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/>	3 description	text	latin2_hungarian_ci		Nem	Nincs		
<input type="checkbox"/>	4 speciesId 	int(11)			Nem	Nincs		

Az oltásokat (ID + név + leírás), illetve a hozzájuk tartozó faj ID-jét tartalmazza.

Táblához tartozó SQL parancs:

```
CREATE TABLE `vaccines` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  `description` text NOT NULL,  
  `speciesId` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_hungarian_ci;  
  
ALTER TABLE `vaccines`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `speciesId` (`speciesId`);  
  
ALTER TABLE `vaccines`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=13;  
  
ALTER TABLE `vaccines`  
  ADD CONSTRAINT `vaccines_ibfk_1` FOREIGN KEY (`speciesId`) REFERENCES `species` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```



6.4. Szerepkörök és jogosultságaik

Projektünkbe számos különböző szerepkör van, mindegyik különböző jogosultságokkal rendelkezik.

6.4.1. Fiók nélküli látogatók – vendégek

A MenhelyMágus weboldalát azok is böngészhetik, akik nincsenek bejelentkezve, persze csak korlátozott módon.

- Bejelentkezés
- Regisztráció
- Állatok böngészése
- Állatok szűrése szűrők alapján
- Állatok keresése preferenciák alapján

Az asztali alkalmazást csak admin vagy alkalmazotti fiókkal lehet használni, így vendégek nem használhatják.

6.4.2. Felhasználók

Regisztrációt követően, ha valaki rendelkezik fiókkal, akkor hasonló a jogköre, mint a vendégeknek, de kicsit bővül.

- Kijelentkezés
- Személyes adatok módosítása
- Fiók törlése
- Állatok böngészése
- Állatok szűrése szűrők alapján
- Állatok keresése preferenciák alapján
- Felhasználói preferencia mentése
- Állat kedvencekhez adása
- Állat örökbefogadása

Az asztali alkalmazást csak admin vagy alkalmazotti fiókkal lehet használni, így felhasználók nem használhatják.



6.4.3. Alkalmazottak

- Kijelentkezés
- Személyes adatok módosítása
- Fiók törlése
- Állatok böngészése
- Állatok szűrése szűrők alapján
- Állatok keresése preferenciák alapján
- Új állat hozzáadása
- Állat módosítása
- Állat törlése
- Örökbefogadások kezelése
 - Örökbefogadás elutasítása
 - Örökbefogadás jóváhagyása

Az alkalmazottak a weboldalt és az asztali alkalmazást egyaránt használhatják.

6.4.4. Adminok

- Kijelentkezés
- Személyes adatok módosítása
- Fiók törlése
- Állatok böngészése
- Állatok szűrése szűrők alapján
- Állatok keresése preferenciák alapján
- Adminok kezelése
 - Új admin hozzáadása (*Csak asztali verzióban*)
- Alkalmazottak kezelése
 - Alkalmazotti menhely ID módosítása
 - Alkalmazott törlése
- Felhasználók kezelése
 - Felhasználó törlése



- Menhelyek kezelése
 - Új menhely hozzáadása
 - Menhely adatainak módosítása
 - Menhely törlése

Az adminok a weboldalt és az asztali alkalmazást egyaránt használhatják.



6.5. Alkalmazások felépítése, szerkezete, felületek bemutatása

6.5.1. Szétválasztás

Az alkalmazásunk három elkülöníthető részből áll: egy API-szerverből, egy webes frontendből, és egy asztali alkalmazásból. A weboldal könnyű hozzáférést biztosít a felhasználók számára, akár mobileszközről is. Az asztali alkalmazás egy robosztus felületet biztosít a mindennapi alkalmazotti használatához. Az API szerver pedig központi adattárolást biztosít, amit könnyű elérni bármilyen kliensből.

6.5.2. PHP apiszerver

6.5.2.1. DB:

Ősosztály, azaz ebből öröklődik a legtöbb másik osztály. Szükséges az adatbázishoz való csatlakozáshoz.

6.5.2.2. BaseRepository:

A DB osztály implementációja. Tartalmazza az olyan alap metódusokat, amelyekre több táblának is szüksége van. Ilyen az összes rekord lekérése, egy bizonyos azonosítójú rekord lekérése, a legutolsó azonosító lekérése, rekordok száma, törlés azonosító alapján, új adat hozzáadása, és adat módosítása.

6.5.2.3. Egyéb repositoryk:

A BaseRepositoryra épülnek fel, és az adott táblához tartozó speciális lekérdezéseket tartalmazzák. Többek között egy hitelesítési lépést szúrnak be a lekérdezések elé vagy automatikusan összekapcsolnak táblákat.



6.5.2.4. Authenticate:

Ellenőrzi a kérést kezdeményezőt, hogy van e jogosultsága az adott művelet elvégzésére. Erre részletesebben kitérünk a [Biztonság](#) bekezdésben.

6.5.2.5. Request:

Megnézi a kérés típusát és útvonalát, és attól függően meghívja a szükséges repository metódusát. Ennek a visszatérési értékétől függően pedig eldönti a visszaküldendő üzenet állapotkódját. Ezekkel az adatokkal végül megívja a Response.php response függvényét.

6.5.2.6. Response:

Az állapotkódhoz hozzárendeli a hozzátartozó üzenetet, majd az adatokkal együtt, json formátumban visszaküldi a kliensnek.

6.5.3. Webalkalmazás

6.5.3.1. PHP fájlok

Ezen fájlok egy része tényleges oldal és `<!DOCTYPE html>`-el kezdődik, ilyen a main.php, adminTools.php, adoptions.php, animal.php, animals.php, create.php, edit.php, employeeTools.php, manageEmployees.php, manageShelters.php, és manageUsers.php. A többi fájl csak includeolva van a többi fájlban belül, mivel többször használható kódrészletek. Ilyen az oldal fej és lábléce, illetve a különböző felugró ablakok

6.5.3.2. JS fájlok

Funkció szerint szét vannak választva különböző fájlokra, így mindig csak a szükségesek lesznek meghívva.

6.5.3.3. CSS fájlok

A minden oldalon használt stíluslap a style.css, minden más itt is szét van szedve a funkciók szerint.



6.5.4. Asztali alkalmazás

6.5.4.1. Fő logikát tartalmazó fájlok

App: A xaml fájl a vezérlők globális stílusait és erőforrásait tartalmazza. A „code-behind” pedig a vezérlőkhöz tartozó és egyéb segédfüggvényeket foglalja magába.

Mainwindow: Az alkalmazás alapja, tartalmazza a menürendszert és logikáját, a MainContent elemet, amelyen a különböző UserControl-ok jelennek meg. Emellett itt lesznek megjelenítve a sikert/hibát jelző, és megerősítést kérő felugró elemek.

Apiservice: A backendhez való kapcsolatot teremti meg. Parametrizált függvényekkel könnyíti meg a különböző végpontok elérését. Automatikusan hozzáadja a felhasználó tokenjét a kéréshez, és a GET metódusú lekérdezések deszerializálják a visszakapott JSONt egy tetszőleges típusú objektummá.

6.5.4.2. Classes mappa

Ebben a mappában található osztályok az alkalmazás alapvető **domain modelljét** képviselik. Ezek az osztályok meghatározzák a különböző entitások szerkezetét, és elsősorban objektumok létrehozására és adatkezelésre szolgálnak. Lehetővé teszik az **adatok deszerializálását** JSON-ból, így biztosítva az adatok zökkenőmentes átalakítását JSON és C# objektumok között.

6.5.4.3. Controls mappa

Itt találhatóak a saját készítésű vezérlőink, mint egy legördülő lista, amelyen egyszerre több elem is kijelölhető (MultiSelectComboBox), vagy a telefonszámokat validáló szövegdoboz (PhoneNumber).



6.5.4.4. Converters mappa

Csak az ImagePathConverter fájl tartalmazza, amely egy szöveges adatból létrehozza egy kép elérési útvonalát (például az állat neméből csinál megfelelő ikont).

6.5.4.5. Account mappa

Tartalmazza fiók menüpont oldalait, mint a bejelentkezést, regisztrációt és személyes adatok változtatását

6.5.4.6. Admin mappa

AddAdmin: Létrehoz egy új admin jogosultságú fiókot a megadott adatok alapján.

ListEmployees: Kiírja egy DataGrid segítségével az összes alkalmazottat. Ki tudja szűni az összes olyan felhasználót, amelynek nincsen hozzárendelt menhelye, és keresni alkalmazott neve szerint. Egy sorra kattintva megjelennek az alkalmazott részletei és a menhelyhez tartozó műveletek gombjai. A törlés gomb egy megerősítő ablak elfogadása után kitörli az alkalmazottat, és a menhely változtatása gomb a **LinkEmployee** oldalra irányít át.

LinkEmployee: Látszódik az alkalmazott neve és email címe, de ezek az adatok nem változtathatók. A menhely ID-jének megadásával hozzá lehet adni vagy meg lehet változtatni az alkalmazott menhelyét. Helytelen ID nem adható meg, de a mező üresen hagyható, ezzel jelezve, hogy az alkalmazottnak nincsen menhelye.

AddShelter: Létrehoz egy új állatmenhelyet a megadott adatok alapján.

ListShelters: Egy DataGrid-ben jeleníti meg az összes menhelyet. Tud szűrni város, és menhely neve szerint. Egy sorra kattintva megjelennek a menhely részletei és a menhelyhez tartozó műveletek



gombjai. A törlés gomb egy megerősítő ablak elfogadása után kitörli a menhelyet, a módosítás gomb átirányít az **EditShelter** oldalra, az alkalmazottak kilistázása pedig a **ListEmployees** oldalra, de csak az adott menhelyben dolgozó alkalmazottakra szűrve.

EditShelter: A létrehozáshoz hasonló felépítésű oldal, amely bemeneti mezői már fel vannak töltve a módosítandó menhely jelenlegi adataival.

6.5.4.7. Employee mappa

AddAnimal: Létrehoz egy új állatot a megadott adatok alapján.

ListAnimals: Egy DataGrid-ben jeleníti meg az összes az alkalmazott menhelyéhez tartozó állatot. Tud szűrni faj, és név alapján. Egy sorra kattintva megjelennek az állat részletei és az állathoz tartozó műveletek gombjai. A törlés gomb egy megerősítő ablak elfogadása után kitörli a menhelyet, a módosítás gomb átirányít az **EditAnimal** oldalra.

EditAnimal: A létrehozáshoz hasonló felépítésű oldal, amely bemeneti mezői már fel vannak töltve a módosítandó állat jelenlegi adataival.

AdoptAnimals: Egy DataGrid-ben kiírja az összes öröbefogadási kérelmet: az örökbefogadandó állat adatait és az örökbefogadási kérelmet benyújtó felhasználó adatait. Az alkalmazottnak ilyenkor lehetősége van elfogadni vagy elutasítani a kérelmet.



6.6. Funkciók bemutatása

6.6.1. Fiókkezelés

6.6.1.1. Bejelentkezés

A bejelentkezéshez egy email cím, illetve egy jelszó szükséges. Ezek megadása után a backend létrehoz egy token, amelyet POST requesttel feltölt az adatbázisba. Ezen token fog segíteni a későbbiekben a felhasználó hitelesítéséhez.

A webalkalmazásba bármilyen felhasználóként be lehet jelentkezni, míg az asztali alkalmazásba csak adminként, illetve alkalmazottként.

6.6.1.2. Regisztráció

A regisztráció az asztali alkalmazásban alkalmazottként, míg a webalkalmazásban felhasználóként történik. A szükséges adatok megadása után a backend létrehozza a fiókot, és egy token (ez a token a regisztráció utáni automatikus beléptetés során jön létre), amelyet feltölt az adatbázisba.

A regisztráció után automatikusan megtörténik a bejelentkezés a létrehozott fiókba.

6.6.1.3. Személyes adatok módosítása

A személyes adatok módosítása oly módon zajlik le, hogy amely beviteli mezők üresen vannak hagyva, ott nem történik módosítás, ahol viszont be van írva valami, ott történik. Például, ha egy felhasználó minden adatának mezejére beír valamit, viszont az email cím mezőt üresen hagyja, akkor az esetben az email adatnál nem történik módosítás az adatbázisban, míg a többinél pedig igen.

Az adatmódosítás PUT requesttel történik.



6.6.1.4. Fiók törlése

A művelet során törlődik a fiók, illetve a hozzá tartozó minden token az adatbázisból.

A fiók törlése egy DELETE requesttel történik.

6.6.2. Állatok megtekintése

6.6.2.1. Állatok kilistázása

Amennyiben egy felhasználói fiókkal van valaki bejelentkezve, akkor az ő személyes preferenciái alapján történik meg alapértelmezetten a listázás, máskülönben preferenciaszámolás nélkül.

A webalkalmazásban minden állat megtekinthető bárki számára, míg az asztali alkalmazásban csak akkor listázhatóak, ha alkalmazott van bejelentkezve, és csak azok az állatok fognak megjelenni, akik az alkalmazott menhelyéhez tartoznak.

A lekérdezés a backendben egy POST requesttel történik. Azért nem GET-et használtunk, mert a GET-be nem lehet body-t rakni.

6.6.2.2. Szűrés & Preferencia (WEB)

A szűrés és a preferencia abban segít, hogy a weboldalt látogatók könnyen megtalálhassák a számukra legmegfelelőbb állatot. A szűrő, és a preferencia beállítása után az állatokat kilistázó POST request body-jába kerülnek a beállított értékek.

A szűrés segítségével kevesebb állat jelenik meg (például csak nőstények), míg a preferencia a megjelenített állatok megjelenési sorrendjét módosítja (például a fiatal állatokat dobja be előre).



6.6.2.3. Felhasználói preferencia frissítése (WEB)

A weboldalon, ha valaki felhasználói fiókkal van bejelentkezve, akkor az ő személyes preferenciái töltenek be alapértelmezetten. Ezen preferenciák módosítása után frissítheti azokat a felhasználó.

A frissítés a backendben PUT requesttel történik.

6.6.3. Alkalmazott eszközök

6.6.3.1. Állatkezelés

Az alkalmazott saját menhelyének állatait tudja kezelni. Jogában áll módosítani (PUT), törölni (DELETE) azokat, illetve hozzá is adhat menhelyéhez új állatot (POST).

6.6.3.2. Örökbefogadások kezelése

Az alkalmazott megtekintheti a menhelyéhez tartozó függőben lévő örökbefogadásokat, és azokat jóváhagyhatja (PUT), vagy törölheti (DELETE).

Elfogadáskor a függőben lévőségük módosul az adatbázisban, és az örökbefogadás véglegesen el lesz fogadva. Törléskor az örökbefogadási kérelem törlődik az adatbázisból.

6.6.4. Admin eszközök

6.6.4.1. Admin hozzáadása (Asztali)

Az asztali alkalmazásba belépett adminok számára megnyílik rá a lehetőség, hogy újabb adminokat adjanak hozzá az adatbázishoz. A művelet a backendben POST requesttel történik.



6.6.4.2. Alkalmazottak kezelése

Az adminok tudják módosítani az alkalmazottakhoz tartozó menhelyet (PUT), illetve tudják törölni őket az adatbázisból (DELETE).

6.6.4.3. Felhasználók kezelése (Web)

Felhasználók törlésére van lehetősége az adminoknak a webalkalmazásban. A művelet a backendben DELETE requesttel történik.

6.6.4.4. Menhelyek kezelése

Az eddig létező menhelyek adatait tudják az adminok módosítani (PUT), illetve tudják törölni őket (DELETE). Létre is tudnak hozni egy teljesen új menhelyet, amely POST requesttel történik.

6.6.5. Felhasználói eszközök (Web)

6.6.5.1. Örökbefogadás

A felhasználó tud örökbefogadási kérést küldeni abban az esetben, amennyiben az állat élőhelyei közül legalább egyet biztosítani tud. A kérés elküldése POST requesttel történik.

6.6.5.2. Felhasználói kedvencek

A felhasználó hozzá tud adni állatokat a kedvencei közé, az állatnál lévő csillag ikonra kattintással (POST). Amennyiben a csillagra újra rányom, az állat törlődik a kedvencek közül (DELETE).

Annak, hogy egy állat kedvenc, nincsen semmi konkrét funkcionalitása, csak annyi, hogy a szűrőben lehet úgy szűrni, hogy csak a kedvenceket mutassa az oldal, így azokat az állatokat, akiket elrakott kedvencekbe a felhasználó, a későbbiekben könnyen megtalálhatja.



6.6.5.3. Weboldal személyre szabása (Web)

A webalkalmazásban van rá lehetőség, hogy azt a felhasználó személyre szabja. Ennek két módja van:

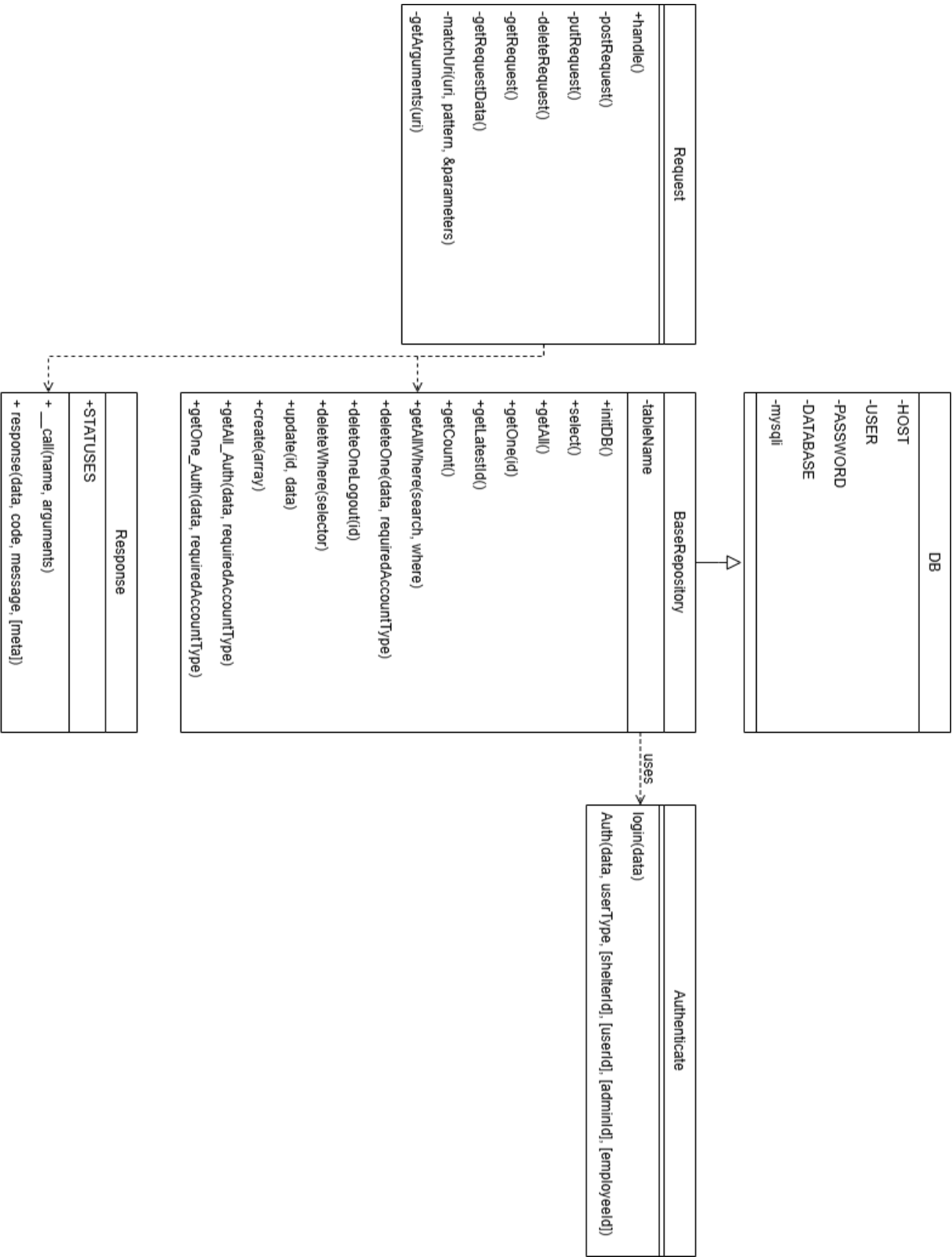
- Az oldalt látogatók választhatnak három nyelv (magyar, angol, illetve német) közül.
- Van rá lehetőség, hogy a sötét és a világos téma között lehessen váltani.

A nyelv, illetve témabeállítások localStorage-ben elmentődnek.

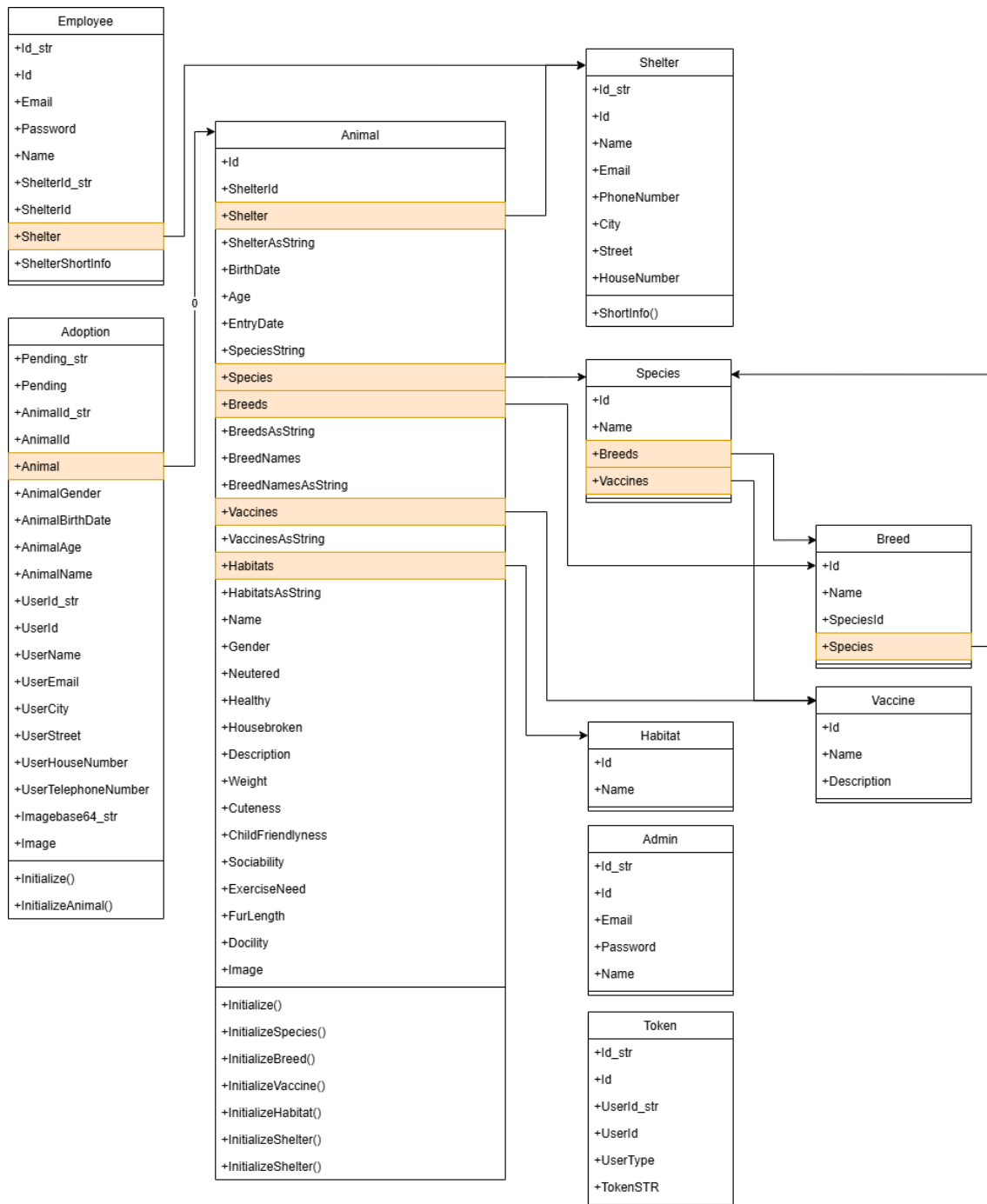


6.7. Osztályok kapcsolati diagramja

6.7.1. PHP apiszerver



6.7.2. C# Asztali alkalmazás



6.8. Input adatok helyességének ellenőrzése, megkötések

Projektünk rengeteg funkcióval rendelkezik, ezek legtöbbje pedig igényli, hogy legyen valamilyen bemenet.

Az oldal/asztali alkalmazás használói által begépelt adatok, mind egytől egyig le vannak ellenőrizve. Amennyiben a felhasználó egy adatot/adatokat rosszul ad meg, akkor arról figyelmeztetve lesz, és mindaddig amíg nem adja meg azokat helyesen, addig az adott művelet nem lesz elvégezve.

Egy felhasználó által begépelt szöveg többféle okból is lehet hibás.

6.8.1. Nem megfelelő formátum

Nem megfelelő formátumú adatot felvinni nem lehet az asztali alkalmazásban, illetve a weben sem.

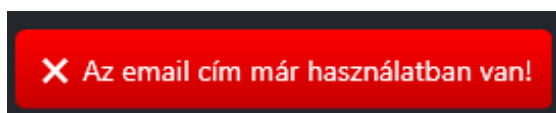
Rossz formátumú adat lehetne például egy olyan telefonszám, amely betűkből áll, vagy egy állat tömege, amely szintúgy betűkből áll. Ezen beviteli mezőknél korlátoztuk, hogy milyen karaktereket lehet bele írni, így, ha a felhasználó esetlegesen szánt szándékkal helytelen beviteli adatokat szeretne itt megadni, akkor sem tudna.

6.8.2. Használatban lévő E-mail cím

Felhasználói fiók és új menhely létrehozásakor, illetve azok módosításakor szükséges megadni E-mail címet. Ilyenkor, ha esetleg olyan E-mail cím lenne megadva, amely már használatban van, akkor arról a felhasználót értesítjük.



Értesítés a weben



Értesítés az asztali alkalmazásban

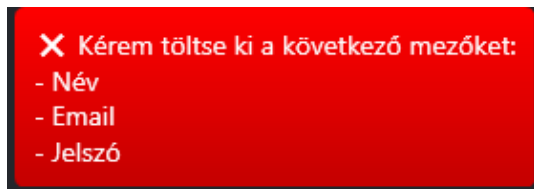


6.8.3. Kitöltetlen beviteli mezők

Amennyiben a felhasználó nem tölt ki minden adatot, amely szükséges egy művelet végrehajtásához, akkor az esetben értesítést kap róla, és a művelet mindaddig nem futhat le, amíg ki nem tölt minden beviteli mezőt.



Értesítés a weben



Értesítés az asztali alkalmazásban



6.9. Biztonság

6.9.1. Hitelesítés

A backend biztosítja, hogy mindenki csak azt tudja elvégezni, amire neki jogosultsága van. Például egy alkalmazottként bejelentkezett ember nem fog tudni adminokat törölni, vagy menhelyeket hozzáadni, de viszont a saját menhelye állatainak kezelésére van lehetősége.

Minden olyan repositoryban, amelynél szükséges hitelesítés, ott az Authenticate.php-ből az „**Auth**” nevű függvény meg van hívva, és csak akkor futnak le az egyes műveletek, hogyha ez a függvény true-t ad vissza, azaz az adott felhasználónak valóban van jogosultsága az adott művelet elvégzésére.

Authenticate.php : Auth()

- Visszaadja, hogy az adott felhasználó jogosult-e az adott művelet elvégzésére
 - Paraméterben megkapja az éppen bejelentkezve lévő felhasználó tokenjét, és az alapján eldönti, hogy van-e jogosultsága a műveletre vagy nincs.
 - Figyeli, hogy a bejelentkezett felhasználó típusa egyezik-e az elvárttal.
 - Elvárt: admin | Bejelentkezve: alkalmazott -> false
 - Elvárt: admin | Bejelentkezve: admin -> true
 - Egyes esetekben figyeli (a fiók típuson felül), hogy a felhasználó ID egyezik-e az elvárttal. Ez például akkor fontos, amikor egy felhasználó személyes adatot kíván módosítani, ilyenkor csak a saját adatait módosíthatja, más felhasználókét pedig nem.
 - Elvárt ID: 6 | Bejelentkezett ID : 1 -> false
 - Elvárt ID: 6 | Bejelentkezett ID : 6 -> true



- Egyes esetekben figyelni, hogy az adott alkalmazott melyik menhelyhez tartozik (ID). Ennek oka például, hogy egy állat adatait, csak az módosíthassa/törölje, aki ahhoz a menhelyhez tartozik, ahová az állat is.
 - Elvárt ID: 2 | Bejelentkezett ID: 2 -> true
 - Elvárt ID: 2 | Bejelentkezett ID: 8 -> false
- Abban az esetben, hogyha a hitelesítés sikertelen volt, akkor a backend 401-es kódot („Unauthorized”) ad vissza.

Példa sikertelen hitelesítésre:

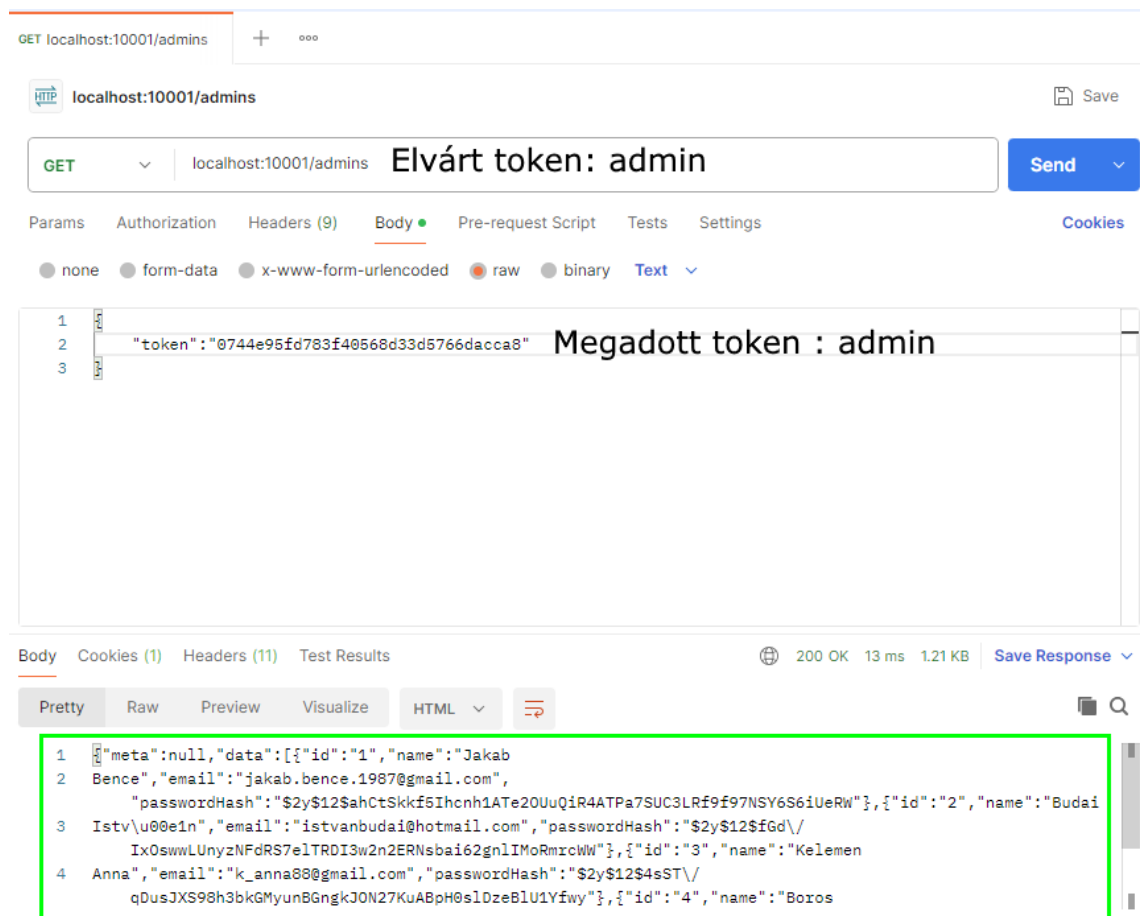
The screenshot shows a REST client interface with a GET request to `localhost:10001/admins`. The request body is `Elvárt token: admin`. The response status is `401 Unauthorized` with a response time of `32 ms` and a size of `529 B`. The response body is displayed in JSON format:

```
1 {"meta":null,"data":["Unauthorized"],"message":"Unauthorized","code":401}
```

The response body is highlighted with a red box. The interface also shows tabs for Params, Authorization, Headers (9), Body, Pre-request Script, Tests, and Settings. The Body tab is selected, and the response is shown in a Pretty view.



Példa sikeres hitelesítésre:



GET localhost:10001/admins

Elvárt token: admin

Megadott token : admin

```
1 [{"meta":null,"data":[{"id":"1","name":"Jakab
2 Bence","email":"jakab.bence.1987@gmail.com",
3 "passwordHash":"$2y$12$ahCtSkkf6Ihcnh1ATe20UuQIR4ATPa7SUC3LRf9f97NSY6S6iUeRW"},{"id":"2","name":"Budai
4 Istv\u00e9n","email":"istvanbudai@hotmail.com","passwordHash":"$2y$12$fGd\
5 IxOswwLUnyzNFdRS7e1TRDI3w2n2ERNsbai62gnlIMoRmrcWW"},{"id":"3","name":"Kelemen
6 Anna","email":"k_anna88@gmail.com","passwordHash":"$2y$12$4sST\
7 qDusJXS98h3bkGMyunBGngk3JON27KuABpH0s1DzeBlU1Yfwy"},{"id":"4","name":"Boros
```

6.9.2. Jelszó titkosítás

Regisztrációnál a jelszavak titkosítva/hashelve lesznek, ezzel tettük igazán biztonságossá projektünket. Ha phpMyAdminba belépve megnézi valaki az összes felhasználó adatát, akkor nem fogja látni a rendes jelszavukat, csak a titkosított formájukat.

A Jelszavak titkosítása a `password_hash()` segítségével történik PHP-ben. Ez egyirányú módon titkosítja a jelszavakat, azaz nem lehet visszafejteni egy hashből, hogy mi a jelszó.

A titkosított jelszavakat ellenőrizni (például bejelentkezésnél) a `password_verify()` segítségével lehet, amely paraméterben megkapja a jelszót, illetve a titkosított jelszót, és visszaadja, hogy a jelszó egyezik-e a titkosított jelszóval.



id	name	email	passwordHash
1	Jakab Bence	jakab.bence.1987@gmail.com	\$2y\$12\$ahCtSkkf5lhcnh1ATe2OUuQiR4ATPa7SUC3LRf9f97N...
2	Budai István	istvanbudai@hotmail.com	\$2y\$12\$fGd/lxOswwLUnyzNFdRS7eITRDI3w2n2ERNsbai62gn...
3	Kelemen Anna	k_anna88@gmail.com	\$2y\$12\$4sST/qDusJXS98h3bkGMyunBGngkJON27KuABpH0sID...
4	Boros László	boroslaciii@gmail.com	\$2y\$12\$m2vqQaSDLqKHpCMQskELYOSQ8kbS/ptmSTk4SniL9Re...
5	Bogdán Julcsa	b.julcsa9@citromail.hu	\$2y\$12\$PXR7WP/6CpDVzujj5dLk8e0junzWcC5xvZDmEjurMvK...

Titkosított jelszavak az adatbázisban

```
if (isset($data['passwordHash'])) {
    $data['passwordHash'] = password_hash($data['passwordHash'], PASSWORD_DEFAULT);
}
```

Jelszótitkosítás a backendben

```
if (password_verify($data['password'], $user['passwordHash']))
{
```

Jelszó ellenőrzés a backendben



7. Felhasználói dokumentáció

7.1. Rendszerkövetelmények

7.1.1. Webes felület

A program használatához ajánlott böngészők a következők:

- Chrome v130 vagy nagyobb
- Firefox v130.0 vagy nagyobb
- Microsoft Edge v128 vagy nagyobb
- Opera v113 vagy nagyobb
- Safari v17 vagy nagyobb

A legjobb felhasználói élményhez asztali módban legalább 860x600px, mobil módban 375x650px képernyőméret ajánlott.

7.1.2. Asztali alkalmazás

A program használatához, Windows 10 vagy Windows 11 operációs rendszer szükséges. A minimum szükséges képernyőméret 850x500px. A program nem igényel jelentős számítási kapacitást, bármilyen e századbeli számítógép megfelel, de internetkapcsolatra van szüksége.

7.1.3. Kiszolgáló

A szerver futtatásához Windows 10 vagy Windows 11 rendszerű számítógép szükséges. A program futásához szükség van egy MySql adatbázis-kiszolgálóra, PHP-ra és Composer-re.



7.2. Telepítési útmutató

7.2.1. Asztali alkalmazás

Az asztali alkalmazás „telepítéséhez” csomagoljuk ki a 'futtatandó fájlok mappája'/'MenhelyMagus kezelo.zip' archívumot. Az ebben található végrehajtható fájl ezután önállóan futtatható.

7.2.2. Szerver

A szerver telepítéséhez feltételezzük, hogy a szükséges függőségek már telepítve vannak és a PHP, valamint a Composer is szerepel a PATH-ban.

Először töltsük le a projektet! A 'server' mappába navigálva egy futtassuk le a '*composer dump-autoload*' parancsot, hogy megoldja a függőségeket.

Az adatbázis-kiszolgálón hozzuk létre a '*menhelymagus*' adatbázist, majd importáljuk a '*database/menhelymagus.sql*' fájlt, ami létrehozza a táblákat, kulcsokat, valamint feltölti adattal.

A futtatáshoz kövessük a 7.2.2.1. pontban található lépéseket.

7.2.2.1. Szerver indítása

Győződjünk meg róla, hogy az adatbázis-kiszolgáló elérhető!

Az API kiszolgáló és a webszerver két külön folyamat, amiket a '*server/start.bat*', valamint az 'Állatmenhely WEB/start.bat' fájlokkal tudunk elindítani.

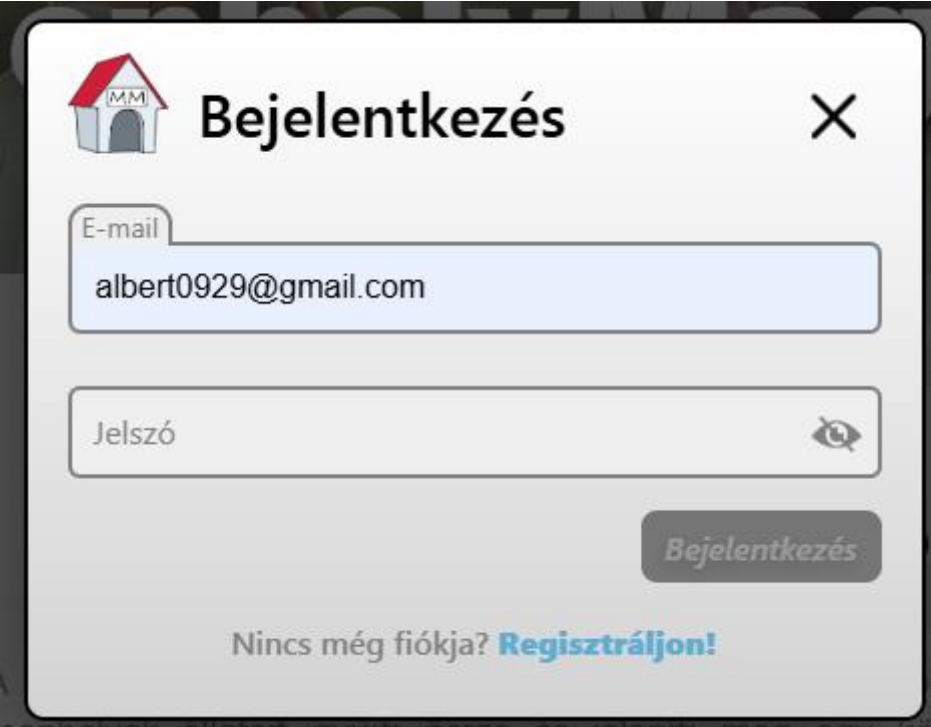


7.3. Webes felület gyakorlati bemutatása

7.3.1. Webes felület

7.3.1.1. Bejelentkezés

Állatokat bejelentkezés nélkül is böngészhetünk, azonban beállításaink és kedvenc állataink elmentéséhez érdemes bejelentkezni. Örökbefogadáshoz, valamint a menhelyek és a rendszer kezeléséhez pedig szükséges a bejelentkezés.




The screenshot shows a login window titled "Bejelentkezés" (Login) with a close button (X) in the top right corner. In the top left corner, there is a small icon of a house with "MM" on it. The form contains two input fields: "E-mail" with the value "albert0929@gmail.com" and "Jelszó" (Password) with a toggle icon (an eye with a slash) on the right. Below the password field is a button labeled "Bejelentkezés". At the bottom of the form, there is a link that says "Nincs még fiókja? [Regisztráljon!](#)".




7.3.1.2. Állatok oldal

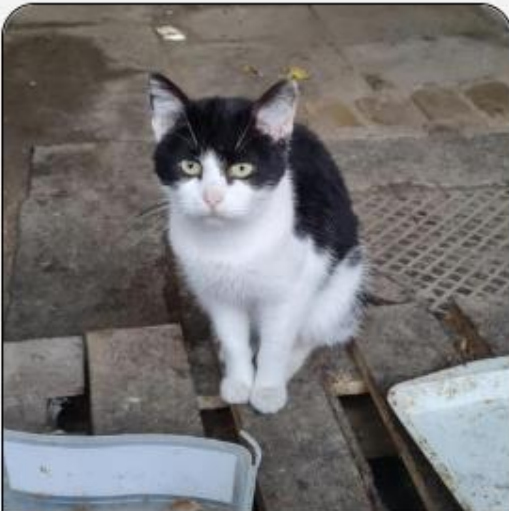
Az oldal törzsén kis kártyákon jelennek meg az állatok főbb jellemzőikkel. A kártyákon a következő információk láthatók: név, kor, nem, helyszín és néhány jellemző tulajdonság.




Pajti 2 ♂ ★
📍 Göd
Ivartalanított Gyerekbarát



Bodza 2 ♀ ★
📍 Győr
Lusta Fiatal



Bambusz 1 ♀ ★
📍 Budapest
Lusta Magának való



Drazsé 4 ♂ ★
📍 Göd
Társaslény Ivartalanított



7.3.1.3. Szűrő és preferenciák

A számunkra tökéletes állat kiválasztásához szűrőt és preferenciát állíthatunk be az állatok oldalán.

Szűrő

Nem

☒ ☐ ☐

Ivartalanított

☒ ☒ ☒

Szobatiszta

☒ ☒ ☒

Kedvenc-e

☐

Faj

Macska

Fajta

Fajta

Menhely

Helyszín

Állat élőhelye

Élőhely

A szűrőben kötelező megkötéseket tehetünk, például, hogy az állat legyen szobatiszta. A rendszer ezután a *Szűrés!* gombra kattintva csak az ennek megfelelő állatokat fogja megjeleníteni.

A preferenciák '*Érték*' beállításával megadhatjuk milyen tulajdonságú állatnak örülnénk. A mellette található '*Súly*' csúszkával pedig megadhatjuk, hogy mennyire fontos az adott tulajdonság.

A mintán például egy apró, kis mozgásigényű állatot keresünk, de

Preferenciák

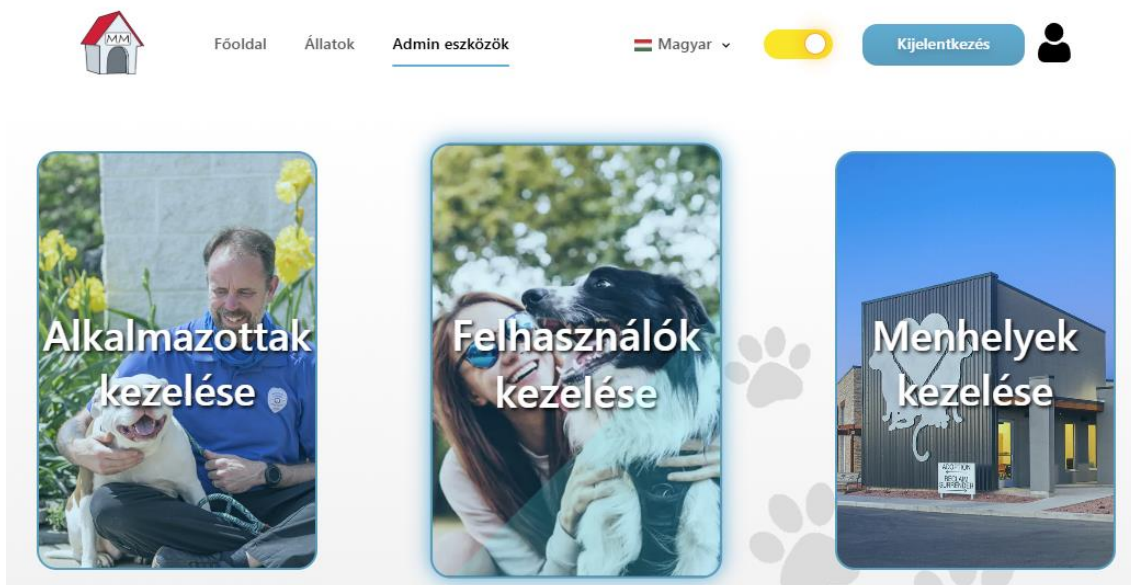
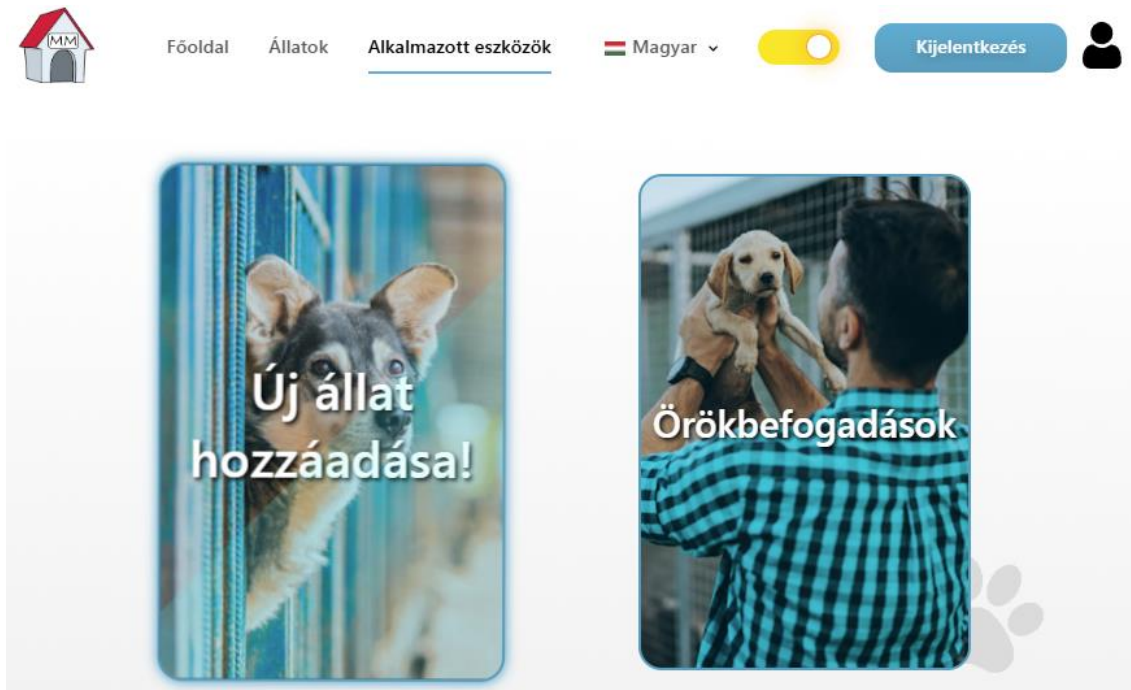
	Érték	Súly
Életkor	<input type="checkbox"/>	<input type="checkbox"/>
Tömeg	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Cukiság	<input type="checkbox"/>	<input type="checkbox"/>
Gyerekbarát	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Társaslény	<input type="checkbox"/>	<input type="checkbox"/>
Mozgásigény	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Szőrhossz	<input type="checkbox"/>	<input type="checkbox"/>
Tanulékony	<input type="checkbox"/>	<input type="checkbox"/>

legfőképpen legyen gyerekbarát. A rendszer mindet figyelembe véve rangsorolva jeleníti meg az állatokat.



7.3.1.4. Alkalmazott és admin eszközök

A megfelelő jogosultsággal rendelkező fiókkal belépve egy új fül jelenik meg a navigációs soron. A képekre kattintva továbbléphetünk az adott funkciók oldalaira.



7.3.1.5. Állat hozzáadása

Az állatok hozzáadása az alábbi felületen történik. A Leírás és oltásokon kívül minden mező kitöltése kötelező. A képfeltöltő menü a képre kattintva nyílik meg. Ismeretlen születési dátumnál érdemes hó/01-et, esetleg év/jan. 01-et megadni. Keverékek esetén a legjellemzőbb fajtá(ka)t adjuk meg!

Az állat módosítása oldal ehhez hasonlóan működik.

Buksi

Kutya 1 Fajta

éééé. hh. nn. 2025. 03. 13.

Budapest Súly (kg)

Élőhely 1 Oltás

- ☐ Bordetella
- ☐ Canine Influenza
- ☐ DHPPi
- ☐ Leptospirosis
- ☐ Lyme-kór
- ☒ Veszettség

Cukiság

Gyerekbarát

Társaslény

Mozgásigény

Szőrhossz

Tanulékonyság

Egészséges ☐

Ivartalanított ☐

Szobatiszta ☒

Állat hozzáadása!



7.3.1.6. Admin eszközök Menhelyek kezelése


Az admin eszközök egyike a menhelyek kezelése. Itt egy listában látszik az összes menhely és a velük végrehajtható műveletek. A szerkesztés gombra kattintva megjelenik a szerkesztési ablak.

A többi admin eszköz ehhez hasonlóan működik.

Menhelyek táblázata

ID	Név		
4	SzívHang Allatotthon		
5	Új Esély Otthon		
6	Pusztta-Tanya		
7	Veresi hullőház		
8	A Remény Útja		
9	Védett Mancsok		

[Új menhely](#)

 **Menhelyadatok** ×

Név

Csepeli Állatotthon

E-mail

allat.otthon@csepel.hu

Város

Budapest

Utca

Reggel utca

Házzszám

48

Telefonszám

06209724532

Módosítások elvetése

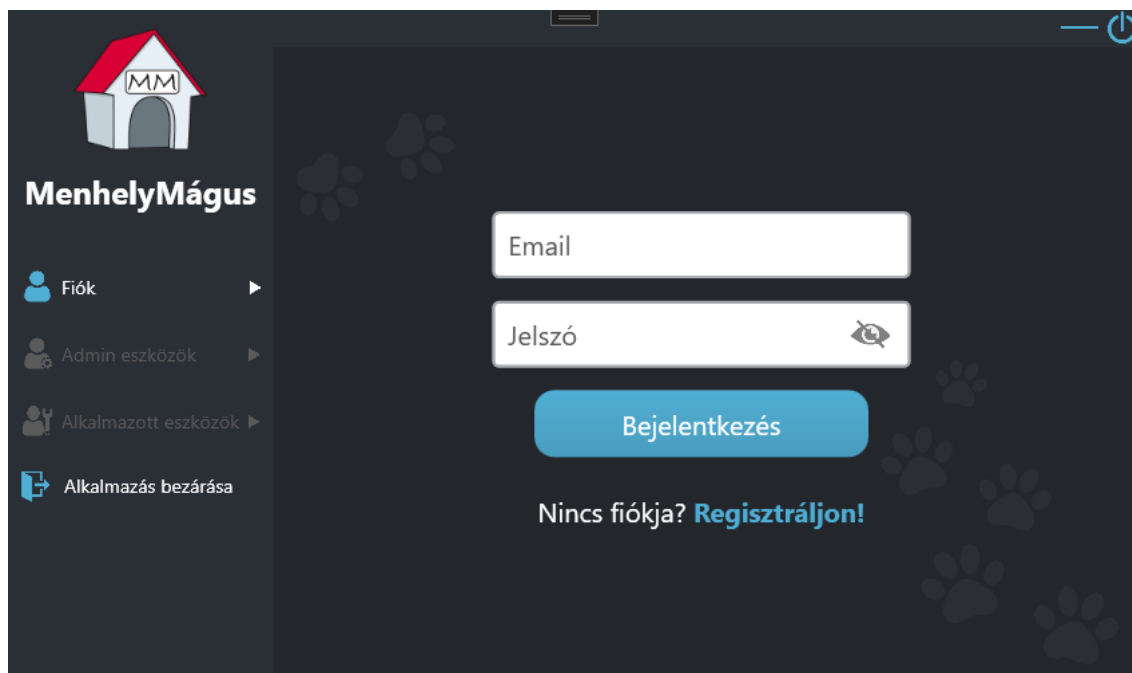
Módosítások mentése



7.3.2. Asztali alkalmazás

7.3.2.1. Belépés

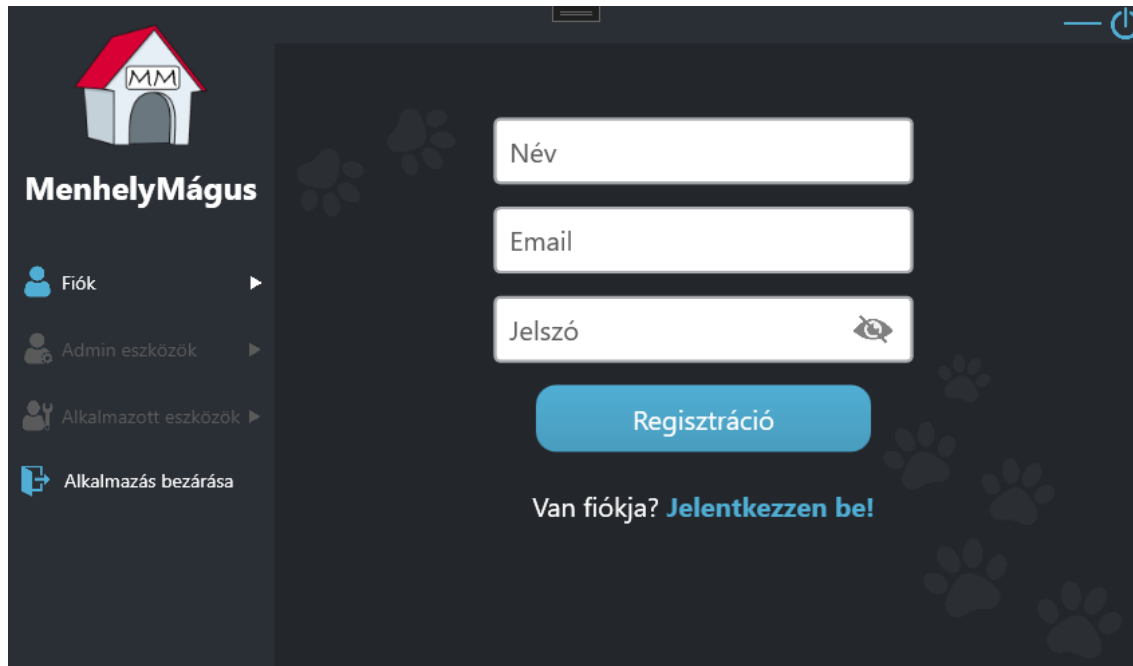
Az alkalmazás használatához alkalmazotti vagy rendszergazdai jogosultságú fiók szükséges.



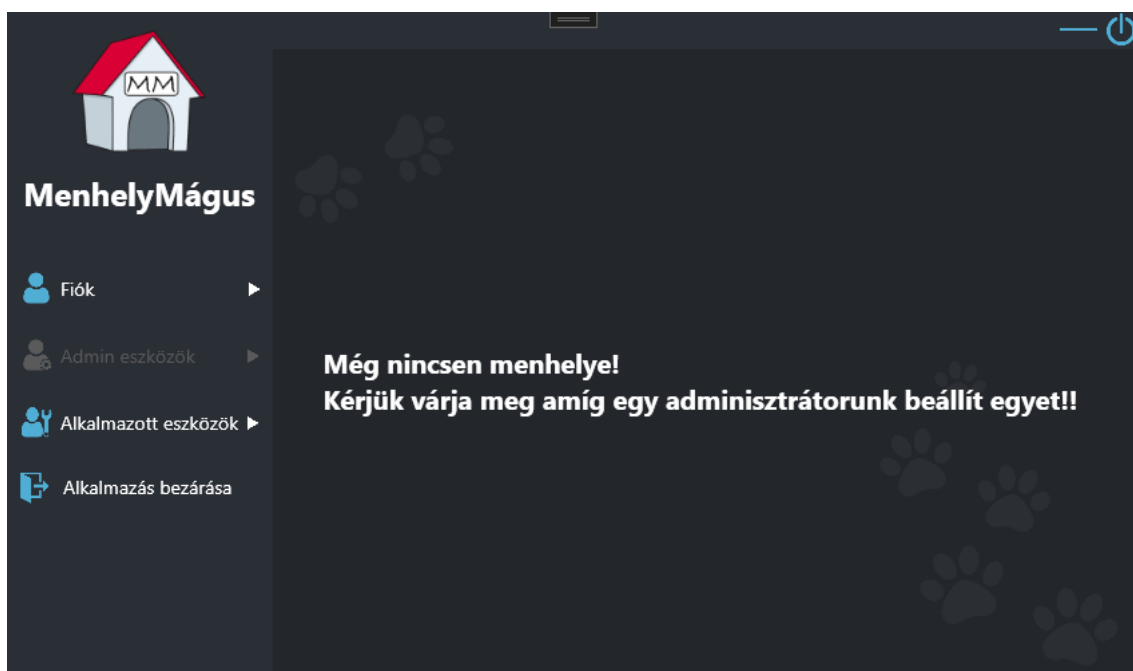
7.3.2.2. Regisztráció

Az asztali alkalmazásban kizárólag alkalmazottként lehet regisztrálni. Új admin felvételéhez lásd 7.3.2.6.

Regisztráció után még nem tudunk azonnal műveleteket végrehajtani. Előtte egy adminnak hozzá kell rendelnie a menhelyünkhöz.

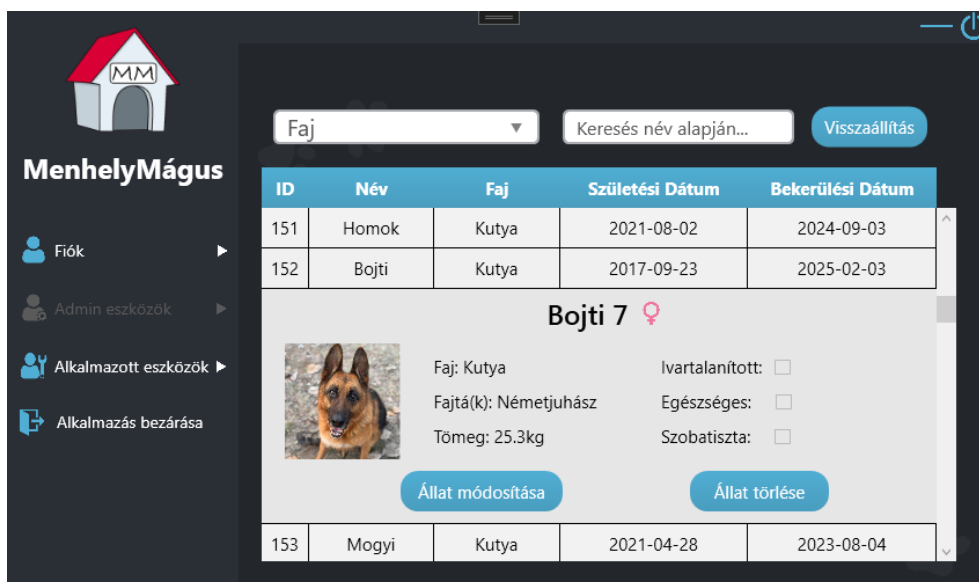


The screenshot shows the registration interface of the MenhelyMágus application. On the left is a dark sidebar with a house icon containing 'MM' and the title 'MenhelyMágus'. Below the title are four menu items: 'Fiók' (Account), 'Admin eszközök' (Admin tools), 'Alkalmazott eszközök' (Employee tools), and 'Alkalmazás bezárása' (Close application). The main area has a dark background with paw prints. It contains three white input fields labeled 'Név' (Name), 'Email', and 'Jelszó' (Password) with an eye icon for toggling visibility. Below these is a blue 'Regisztráció' button. At the bottom, it says 'Van fiókja? [Jelentkezzen be!](#)'.



7.3.2.3. Állatok listája

Ebben a listában tekinthetjük meg a menhelyünkhöz tartozó állatokat. Az adott sorra kattintva részletesebb információt kaphatunk az állatról.



The screenshot shows the 'MenhelyMágus' application interface. On the left is a sidebar with the logo (a house with 'MM') and navigation links: 'Fiók', 'Admin eszközök', 'Alkalmazott eszközök', and 'Alkalmazás bezárása'. The main area displays a table of animals with columns: ID, Név, Faj, Születési Dátum, and Bekerülési Dátum. The table lists three animals: Homok (ID 151, born 2021-08-02, entered 2024-09-03), Bojti (ID 152, born 2017-09-23, entered 2025-02-03), and Mogyi (ID 153, born 2021-04-28, entered 2023-08-04). Below the table, the details for 'Bojti 7' are shown, including a photo of a German Shepherd, breed (Kutya), breed details (Németjuhász), weight (25.3kg), and checkboxes for spayed, healthy, and vaccinated. Buttons for 'Állat módosítása' and 'Állat törlése' are present.

ID	Név	Faj	Születési Dátum	Bekerülési Dátum
151	Homok	Kutya	2021-08-02	2024-09-03
152	Bojti	Kutya	2017-09-23	2025-02-03
153	Mogyi	Kutya	2021-04-28	2023-08-04

Bojti 7 ♀

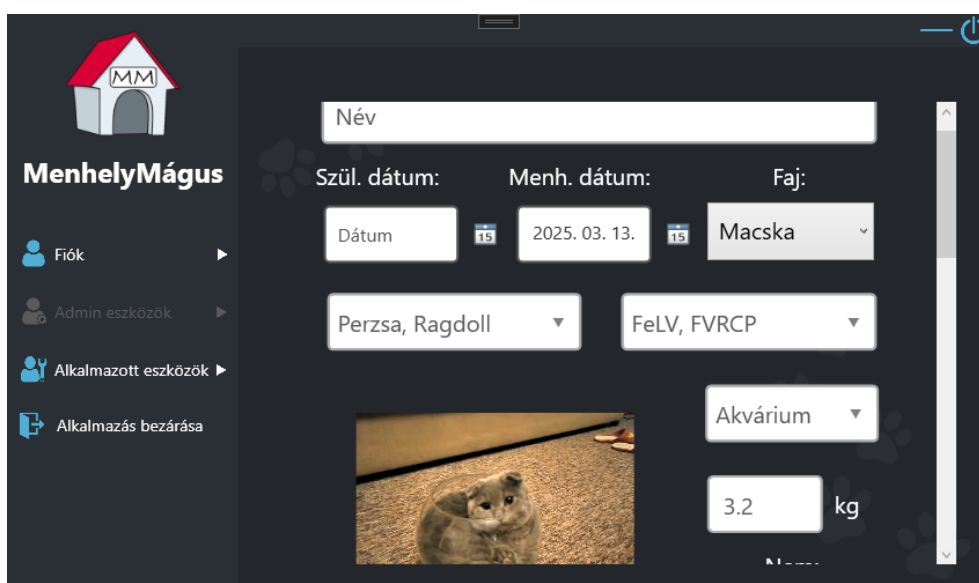
Faj: Kutya
Fajtá(k): Németjuhász
Tömeg: 25.3kg

Ivartalanított: ☐
Egészséges: ☐
Szobatiszta: ☐

Állat módosítása Állat törlése

7.3.2.4. Állat hozzáadása

Az állatok hozzáadása az alábbi felületen történik. A Leírás és oltásokon kívül minden mező kitöltése kötelező. A képfeltöltő menü a képre kattintva nyílik meg



The screenshot shows the 'MenhelyMágus' application interface for adding a new animal. The sidebar is the same as in the previous screenshot. The main area contains a form with the following fields: 'Név' (Name), 'Szül. dátum:' (Birth date) with a calendar icon, 'Menh. dátum:' (Adoption date) with a calendar icon, 'Faj:' (Species) with a dropdown menu showing 'Macska', 'Perzsa, Ragdoll' (Breed) with a dropdown menu, 'FeLV, FVRCP' (Vaccinations) with a dropdown menu, 'Akvárium' (Aquarium) with a dropdown menu, and a weight field showing '3.2 kg'. There is also a photo upload area with a placeholder image of a cat.

Név

Szül. dátum: Menh. dátum: Faj:

Dátum 2025. 03. 13. Macska

Perzsa, Ragdoll FeLV, FVRCP

Akvárium


3.2 kg



7.3.2.5. Örökbefogadások

Itt egy listában láthatók az örökbefogadásra váró állatok. A sorra kattintva megnézhetjük, illetve elfogadhatjuk a kérést.

The screenshot shows the 'MenhelyMágus' application interface. On the left is a sidebar with a house icon and the text 'MenhelyMágus'. Below it are menu items: 'Fiók', 'Admin eszközök', 'Alkalmazott eszközök', and 'Alkalmazás bezárása'. The main area has a search bar 'Keresés név alapján...' and a 'Visszaállítás' button. Below is a table with columns: 'Állatnév', 'Örökbefogató név', 'Örökbefogató email', and 'Születési Dátum'. The first row shows 'Zsóka', 'Bálint Anna', 'annabalint0303@gmail.com', and '2022-01-01'. Below the table, the details for 'Zsóka 3' are shown, including a photo of a dog and the adopter's information: 'Név: Bálint Anna', 'E-mail cím: annabalint0303@gmail.com', 'Telefonszám: 06208358138', and 'Lakcím: Göd, Mátyás Király utca 30'. There are two buttons: 'Örökbefogadás elutasítása' and 'Örökbefogadás elfogadása'. At the bottom, another row of the table shows 'Bojti', 'Veres Mónika', 'veresmoncsika@gmail.com', and '2017-09-23'.

Állatnév	Örökbefogató név	Örökbefogató email	Születési Dátum
Zsóka	Bálint Anna	annabalint0303@gmail.com	2022-01-01
Zsóka 3  Örökbefogató adatai: Név: Bálint Anna E-mail cím: annabalint0303@gmail.com Telefonszám: 06208358138 Lakcím: Göd, Mátyás Király utca 30			
Bojti	Veres Mónika	veresmoncsika@gmail.com	2017-09-23

7.3.2.6. Admin hozzáadása

Új admin fiókot csak egy már meglévő admin fiókban lehet létrehozni. **Az itt megadott jelszó ideiglenes,** belépés után módosítani kell!

The screenshot shows the 'Admin hozzáadása' form in the 'MenhelyMágus' application. The sidebar is the same as in the previous screenshot. The main area has the title 'Admin hozzáadása' and three input fields: 'Email', 'Jelszó', and 'Név'. Below the fields is a blue button labeled 'Admin hozzáadása!'. The background has a dark theme with paw print patterns.



7.3.2.7. Alkalmazottak kezelése

Az admin az alkalmazottak adatait egy listában látja, ahol az egyes sorokra kattintva több információt is megtudhat róluk. Itt található a *'Menhely változtatása'* gomb, amivel át lehet helyezni, vagy az új alkalmazottakat hozzárendelni egy menhelyhez.

A menhely ID-t beírva megjelenik a menhely neve, üres értéket megadva pedig el lehet távolítani a hozzárendelt menhelyt.

ID	Név	E-mail	Menhely ID
1	Tóth Albert	albert0929@gmail.com	1
2	Szabó Julcsi	szabojulcsi02@gmail.com	1
3	Rézműves Csabika	rezmuves.csabika@gmail.com	2
4	Orsós Rómeó	orsosromeo22@hotmail.com	2

Alkalmazott menhelyének adatai: Név: Újpesti Menhely
Város: Budapest
Telefonszám: 06209728562
E-mail: menhelyujpest@freemail.hu

Alkalmazotti menhely ID változtatása

szabojulcsi02@gmail.com

Szabó Julcsi

1

Név: Újpesti Menhely
Város: Budapest
Telefonszám: 06209728562
E-mail: menhelyujpest@freemail.hu

Menhely ID változtatása! Mégsem

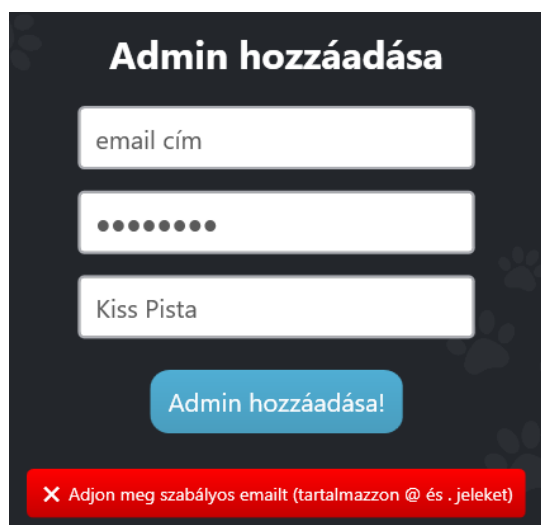


8. Tesztelés

Az alábbiakban egy új adminisztrátor felvételének, bejelentkezésének, illetve jogosultságainak tesztelését fogjuk szemléltetni.

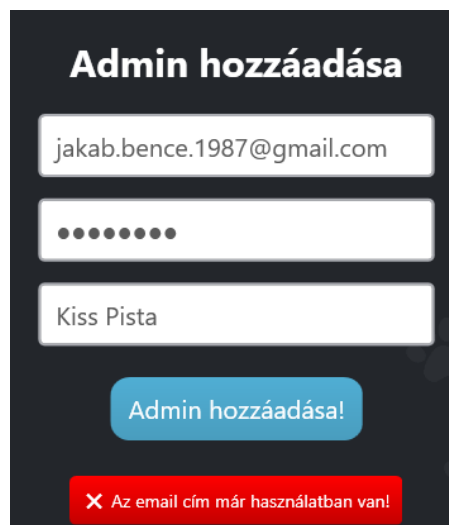
8.1. Új admin hozzáadása

Új adminisztrátori fiókot az asztali alkalmazásból lehet létrehozni, erre a weboldalon nincs lehetőség.



The screenshot shows a dark-themed form titled "Admin hozzáadása". It contains three input fields: "email cím" (containing "email cím"), a password field (containing "••••••••"), and a name field (containing "Kiss Pista"). Below the fields is a blue button labeled "Admin hozzáadása!". At the bottom, a red error message bar displays: "X Adjon meg szabályos emailt (tartalmazzon @ és . jeleket)".

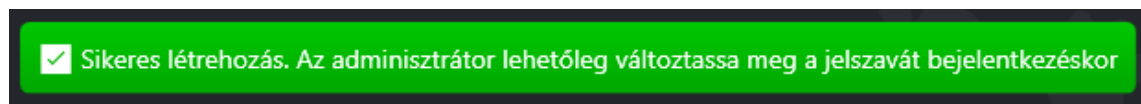
Érvénytelen E-mail cím



The screenshot shows the same "Admin hozzáadása" form. The "email cím" field now contains "jakab.bence.1987@gmail.com". The other fields and the blue button remain the same. The red error message bar at the bottom now displays: "X Az email cím már használatban van!".

Használt E-mail cím

Miután helyes adatokat adtunk meg, egy zöld üzenet jelenik meg, amely közli az új admin felvételének sikerességét.

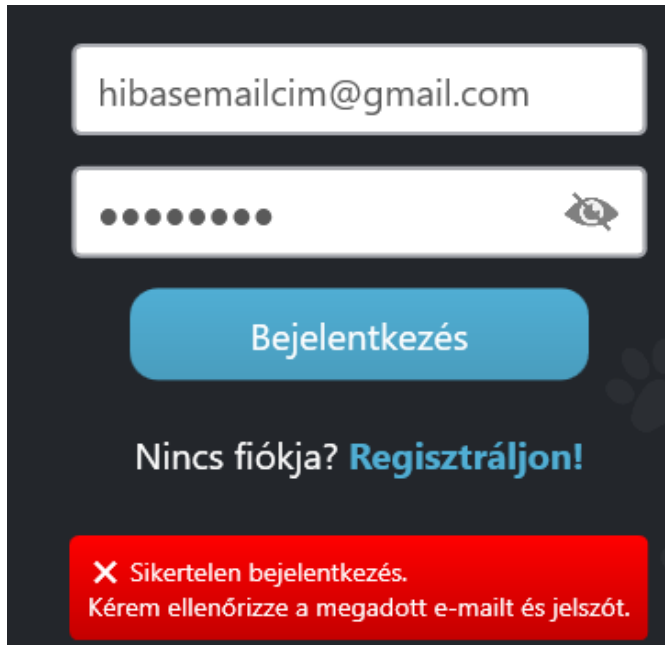


Üzenet (plusz egy biztonsági jó tanács) a sikeres felvétel után



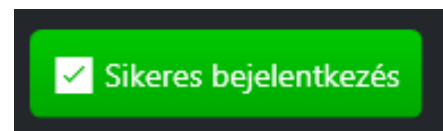
8.2. Bejelentkezés

Létrehozás után már sikeresen be lehet jelentkezni az újonnan létrehozott fiókba a weboldalon, és az asztali alkalmazásban egyaránt.



A login form on a dark background. It has two input fields: the first contains the email 'hibasemailcim@gmail.com', and the second contains masked characters '.....' with a toggle icon on the right. Below the fields is a blue button labeled 'Bejelentkezés'. Underneath the button is the text 'Nincs fiókja? [Regisztráljon!](#)'. At the bottom, a red error message box contains a red 'X' icon and the text: 'Sikertelen bejelentkezés. Kérem ellenőrizze a megadott e-mailt és jelszót.'

Hibás bejelentkezési adatok



A green success message box with a green checkmark icon and the text: 'Sikeres bejelentkezés'.

*Sikeres bejelentkezés
esetében megjelenő
üzenet*

8.3. Menhelymódosítás

Ezután újonnan létrehozott adminunk számára megnyílik a lehetőség rá, hogy az admin eszközök menüpontra kattintson.

Az újonnan létrehozott fiókkal próbáljunk meg egy létező menhelyet módosítani. Mint mindenhol, természetesen a menhelymódosítás esetében is lehet hibás input adatokat megadni.



Menhely módosítása

Újpesti Menhely

b.julcsa9@citromail.hu

Budapest 06209728562

Petőfi Sándor utca 63.

Menhely módosítása! Mégsem

✗ Az email cím már használatban van!

Már használatban lévő E-mail cím

Menhely módosítása

Újpesti Menhely

menhelyujpest@freemail.hu

Budapest 06209728562

Utca 63.

Menhely módosítása! Mégsem

✗ Kérem töltsse ki a következő mezőket:
- Utca

Hiányzó adat(ok)



 **Menhely sikeresen módosítva.**

*Sikeres módosítás esetében
megjelenő üzenet*

8.4. Jogosultsághiány

8.4.1. WEB

Amennyiben a weboldalon olyan oldalt ír be a keresőbe az új admin, amelyhez nincs jogosultsága, az oldal tájékoztatást ad róla, majd átdobja a főoldalra.



Jogosultsághiányról tájékoztató üzenet

8.4.2. Postman

Amennyiben az új admin Postmanben (vagy egy ahhoz hasonló programban) olyan műveleteket próbál végrehajtani, amelyekhez nincs joga (például örökbefogadásokat kezelni), akkor az esetben „Unauthorized” (kód: 401) hibát kap.



HTTP <http://localhost:10001/adoptions/1/1>

DELETE <http://localhost:10001/adoptions/1/1> **Alkalmazotti művelet (örökbefogadás elutasítása)**

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary **Text** ▼

```
1  
2 "token":"e7f8f3c3ea4c2279d0fccc2729be1e3b"  
3
```

Az új admin tokenje

Body Cookies (1) Headers (12) Test Results

Pretty Raw Preview Visualize

```
"meta":null,"data":[],"message":"Unauthorized","code":401
```

Jogosultságihiányról való tájékoztatás Postmanben



9. Konklúziók

9.1. Tanultak

Projektünk során nagyon sok dolgot tanultunk, és sok új tapasztalatot szereztünk. Megtanultunk effektíven csapatban dolgozni egy nagyobb projekten. Nem lehet egyedül dolgozni, és szükség van mások visszajelzésére, véleményére. A csapatmunka elősegítése érdekében fontos volt megismerni a gites verziókezelést is, ami lehetővé tette a párhuzamos munkát, nagyobb komplikációk nélkül. Most csináltunk először egy olyan nagyobb projektet, amiben szükség volt különböző jogosultságú felhasználók szétválasztására, egy autentikációs rendszer kialakítására. Új volt a WPF használata is az asztali alkalmazáshoz, eddig mindenkinek csak Windows Forms applikációkkal volt tapasztalata.

9.2. Továbbfejlesztési irányok

9.2.1. Új opciók hozzáadása:

Egy adminisztrátornak legyen módja új választási lehetőségeket hozzáadni az adatbázishoz, illetve a meglévőket módosítani, vagy törölni. Ilyenek lehetnek új állatfajok, fajták, oltások és élőhelyek.

9.2.2. Adatbázis lefordítása:

Jelenleg a weboldal frontendje van lefordítva a magyar nyelv mellett angolra és németre. A jövőben célunk, hogy az adatbázis tartalma is mindhárom nyelvben elérhető legyen.

9.2.3. Jobb képfeltöltés:

Állat hozzáadásánál jelenleg bármilyen méretarányú kép megadható, de megjelenítésnél mindig 1:1 arányhoz közelít. Ezt egy alkalmazott



bár tudhatja, és ennek megfelelően tölthet fel képeket, de jobb lenne, ha a feltöltésnél egyből lehetőséget adunk a kép kivágására.

9.2.4. Örökbefogadások megtekintése:

Jelenleg még nincsen lehetősége egy felhasználónak sem a függőben lévő örökbefogadási kérelmeit, sem a már jóváhagyott örökbefogadásait megtekinteni.



10. Irodalomjegyzék/Használt oldalak

10.1. Könyvek

- Robert C. Martin: Tiszta kód - Az agilis szoftverfejlesztés kézikönyve
- Leonard Richardson és Mike Amundsen: RESTful Web APIs

10.2. Használt oldalak

- [W3Schools](#)
- [ChatGPT](#)
- [Stack Overflow](#)
- [Reddit](#)
- [CodeProject](#)
- [YouTube](#)
- [CodePen](#)
- [WPF Tutorial](#)



11. Mellékletek

[MenhelyMágus GitHub](#)

