Állatmenhely vizsgaprojekt

Fejlesztői-, és felhasználói dokumentáció

**BZSH Külkereskedelmi Technikum 2024/2025 tanév**

**A fejlesztői csapat:**

**Csöndesné Krepsz Andrea**

**Dobay Viktória Célia**

**Törgyekes Klaudia**

Tartalomjegyzék

**Bevezetés2**

**Fejlesztői dokumentáció4**

**Felhasználói dokumentáció14**

**Fejlesztési lehetőségek a továbbiakban18**

**Tesztelés19**

## **Bevezetés**

**1.1 Projekt áttekintése**

Ez a dokumentáció egy állatmenhely webalkalmazás frontend és backend fejlesztését mutatja be. Választott projektünk az állatmenhely weboldal fejlesztése volt, amely kutyák és macskák örökbefogadását segíti elő. A felhasználók regisztrálhatnak az oldalon, kedvencnek jelölhetik az állatokat, és időpontot foglalhatnak a menhelyek látogatására, hogy személyesen megismerkedjenek a kiválasztott házikedvencekkel.

**1.2 Projekt célja**

A fejlesztett program célja, hogy megkönnyítse az érdeklődők számára az örökbefogadási folyamatot, biztosítva egy egyszerű és felhasználóbarát élményt a kereséstől kezdve egészen az örökbefogadásig. Továbbá lehetővé tegye a látogatók számára a menhelyen örökbefogadható kutyák és macskák megtekintését, a számukra érdekesnek tartott állatok kedvencek közé emelését, személyes találkozáshoz időpontfoglalást, valamint az adminisztrátorok számára az adatkezelést. A rendszer Laravel backendre és Angular frontendre épül.

## A weboldal főbb funkciói:

* **Állatok listázása:** A felhasználók lekérdezhetik az örökbefogadható állatok listáját, majd regisztrációt és bejelentkezést követően mennyiségi korlátozás nélkül kedvencek közé tehetnek különböző állatokat, illetve személyes találkozóhoz időpontot is foglalhatnak.
* **Kedvenc hozzáadása**: Az érdeklődők kedvencnek jelölhetik az általuk kiválasztott állatokat, hogy később könnyen rátaláljanak.
* **Időpontfoglalás**: A felhasználók időpontot foglalhatnak a menhelyek látogatására, hogy személyesen is találkozhassanak a kiszemelt állatokkal.
* **Regisztráció és bejelentkezés**: Az oldal lehetőséget biztosít a felhasználók számára a regisztrációra és a bejelentkezésre, így azok hozzáférhetnek a személyre szabott funkciókhoz, például a kedvenc állatokhoz és a foglalási előzményeikhez.
* **Admin felület**: A menhelyek adminisztrátorai kezelhetik az állatok listáját, az időpontokat, a felhasználókat és az örökbefogadókat (itt nem csak az oldalra regisztrált felhasználók szerepelnek, hanem akik ténylegesen örökbefogadóként jelentkeztek), valamint a sikeres örökbefogadásokat is tudják rögzíteni/kezeleni.

A cél az volt, hogy a felhasználók számára egy átlátható és jól működő platformot biztosítsunk az örökbefogadási folyamatok lebonyolítására.  Az állatok listázásán kívül minden funkció csak regisztrációt és bejelentkezést követően érhető el tekintettel arra, hogy valóban azok a felhasználók használják a programot, akik igénybe szeretnék venni az állatmenhely szolgáltatásait.

## 1.4 Technológiai stack

**1.4.1 Frontend: Angular**

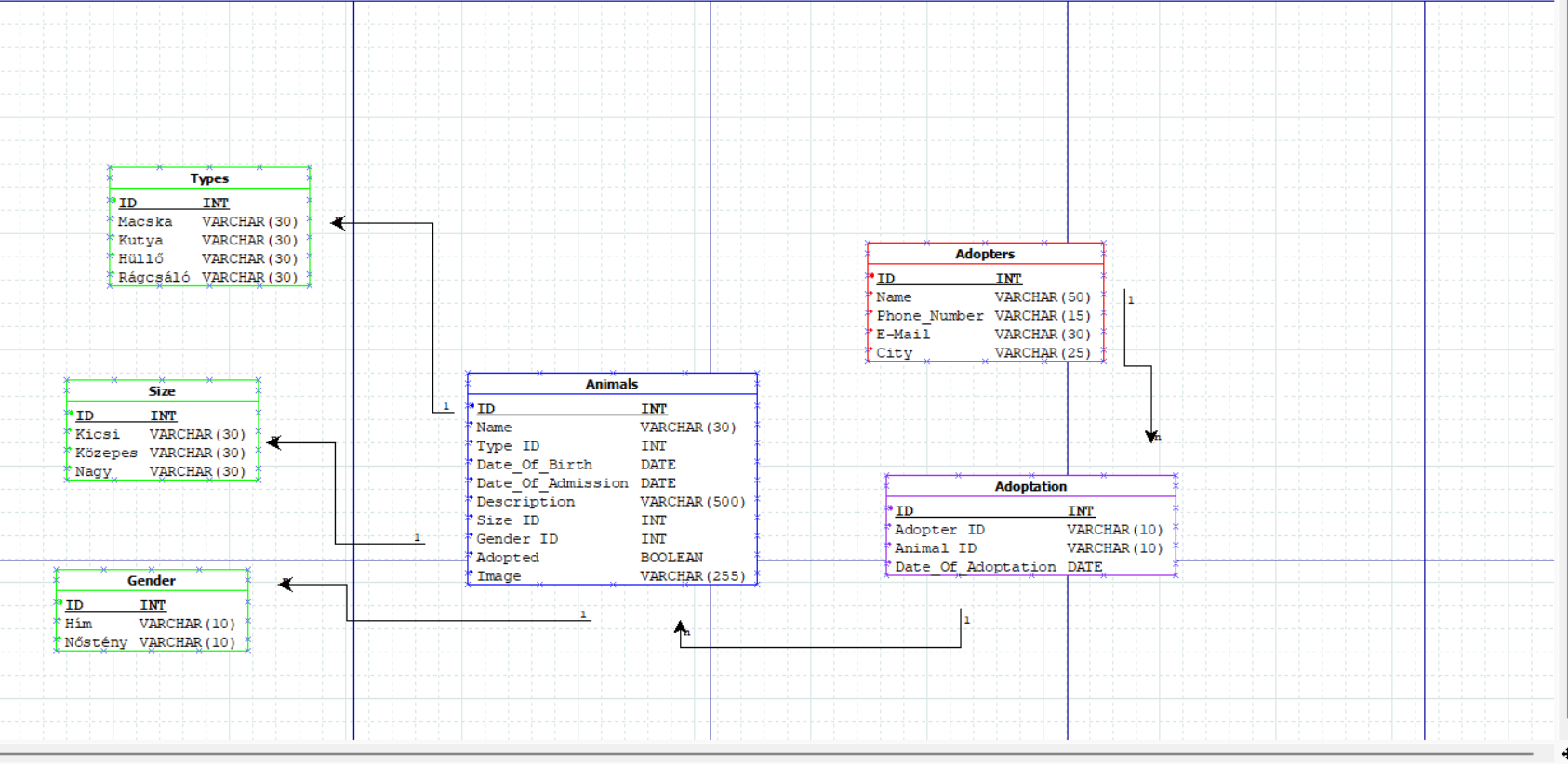
A frontend fejlesztéséhez az Angular keretrendszert választottuk, mivel ez biztosítja a könnyen kezelhető, komponens alapú fejlesztést. Az Angular a weboldal dinamikus felületének létrehozásához, a felhasználói interakciók kezeléséhez és az adatok valós idejű frissítéséhez biztosít alapot.

**1.4.2 Backend: Laravel**

A backend fejlesztéséhez az Laravel PHP keretrendszert választottuk, mivel erős eszközkészletet biztosít a webalkalmazások gyors fejlesztéséhez. Laravel biztosítja a RESTful API-k egyszerű létrehozását, adatbázis-kezelést, valamint a migrációk és modellek segítségével hatékony adatkezelést.

**1.4.3 Egyéb használt technológiák: Dia, MySQL, XAMPP**

Első körben az adatbázis megtervezéséhez a Dia nevű programot használtunk, hogy egy adatmodellt létrehozzunk. Az első terv szerint a kutyák és a macskák két különböző adattáblába kerültek volna, azonban ez szükségtelenné vált, hiszen törekedtünk arra, hogy minél kevesebb adattáblából megoldható legyen a program fejlesztése. Az alábbi ábra szemlélteti, hogy milyen kapcsolatban állnak egymással a különböző táblák, itt szerepelnek az állatok, az örökbefogadók, illetve az örökbefogadások.

****

A fenti adatmodell volt a kiinduló pont a program elkészítésében. A képen láthatóak a különböző kapcsolatok az adatbázis tervezése során.

## **Fejlesztői dokumentáció**

## 2. Funkciók részletes leírása

### 2.1. Regisztráció és bejelentkezés

Felhasználók regisztrálhatnak és bejelentkezhetnek.

A sikeres bejelentkezés után Bearer token kerül eltárolásra localStorage-ban.

A bejelentkezés után a felhasználó szerepe (isAdmin mező: 0, 1, 2) határozza meg a jogosultságokat.

### 2.2. Felhasználói jogosultságok

**Sima felhasználó (isAdmin = 0):**

Állatok böngészése

Időpont foglalása

Kedvencekhez adás és eltávolítás, saját kedvencek listájának betöltése

Saját foglalások lekérdezése, módosítása, törlése

**Admin (isAdmin = 1):**

Állatok kezelése (CRUD)

Örökbefogadók és örökbefogadások kezelése (

Saját időpontfoglalások (új időpont foglalása meglévő listázása, módosítása törlése)

Saját kedvencek kezelése (kijelölés és törlés)

Bármely felhasználó által lefoglalt időpont lekérdezése és törlése

**Superuser (isAdmin = 2):**

Minden admin és sima felhasználó által elérhető funkció

Felhasználók jogosultságainak kezelése

### 2.3. Állatkezelés

Állatok listázása, szerkesztése, új állat felvétele és törlése.

Legördülő menük: fajta, méret, nem.

Módosítás gomb csak szerkesztés után aktív.

### 2.4. Kedvencek kezelése

Bejelentkezett felhasználók kedvencekhez adhatnak állatokat.

Nem bejelentkezett felhasználó esetén modális bejelentkezési felület jelenik meg.

### 2.5. Időpontfoglalás

Időpontfoglalás csak hétköznapokon, 8:00 és 20:00 között, fél vagy egész órában.

Ha nem bejelentkezett felhasználó próbál foglalni, a rendszer bejelentkezésre irányít.

### 2.6. Admin általi időponttörlés

Ehhez a funkcióhoz admin vagy superuser jogosultság szükséges.

Az admin a deleteanyappointment/{id} útvonalon keresztül törölhet időpontot.

A törlésről a felhasználó automatikus értesítést (emailt) kap.

### 2.7. Örökbefogadások és örökbefogadók

Adatok listázása, módosítása, új rekordok felvétele.

Admin felület külön aloldalakon keresztül érhető el.

## 3. Navigációs térkép

### Nyilvános oldalak:

Főoldal

Regisztráció / Bejelentkezés

Örökbefogadható listája

### Bejelentkezett felhasználó:

Saját profil menüpont alatt az alábbi almenüket látja, ahol a lekérdezés mellett törlési, illetve az időpontok esetében módosítási lehetősége van.

Kedvenceim

Foglalásaim

### Admin/Superuser:

Admin felület menüpontjai

Állatok kezelése

Felhasználók kezelése

Örökbefogadók kezelése

Ürökbefogadások kezelése

Időpontok kezelése

Jogosultságkezelés

## 4. Backend API végpontok

**Publikus:**

php

MásolásSzerkesztés

Route::post("/register", [UserController::class, "register"]);

Route::post("/login", [UserController::class, "login"]);

Route::get("/search", [SearchController::class, "searchAnimals"]);

Route::get("/animals", [AnimalController::class, "getAnimals"]);

Route::get("/animal", [AnimalController::class, "getAnimal"]);

Route::get("/adoptable", [AnimalController::class, "getAdoptableAnimals"]);

Route::get("/types", [TypeController::class, "getTypes"]);

Route::get("/genders", [GenderController::class, "getGenders"]);

Route::get("/sizes", [SizeController::class, "getSizes"]);

Route::get("/mail", [MailController::class, "sendMail"]);

**Authentikált felhasználóknak:**

php

MásolásSzerkesztés

Route::middleware("auth:sanctum")->group(function() {

Route::post("/admin", [UserController::class, "giveAdmin"]);

Route::post("/logout", [UserController::class, "logout"]);

Route::get("/user", [UserController::class, "getUser"]);

Route::get("/users", [UserController::class, "getUsers"]);

Route::put("/updateuser", [UserController::class, "updateUser"]);

Route::post("/newanimal", [AnimalController::class, "newAnimal"]);

Route::put("/updateanimal", [AnimalController::class, "updateAnimal"]);

Route::delete("/deleteanimal/{id}", [AnimalController::class, "destroyAnimal"]);

Route::get("/appointments", [AppointmentController::class, "getAppointments"]);

Route::get("/anyappointments", [AppointmentController::class, "getAnyAppointments"]);

Route::post("/newappointment", [AppointmentController::class, "newAppointment"]);

Route::put("/updateappointment/{id}", [AppointmentController::class, "updateAppointment"]);

Route::delete("/deleteappointment/{id}", [AppointmentController::class, "destroyAppointment"]);

Route::delete("/deleteanyappointment/{id}", [AppointmentController::class, "destroyAnyAppointment"]);

Route::post("/addfavorite", [FavoriteController::class, "addFavorite"]);

Route::get("/getfavorites", [FavoriteController::class, "getUserFavorites"]);

Route::delete("/removefavorite/{id}", [FavoriteController::class, "removeFavorite"]);

Route::get("/adopter", [AdopterController::class, "getAdopter"]);

Route::get("/adopters", [AdopterController::class, "getAdopters"]);

Route::post("/newadopter", [AdopterController::class, "newAdopter"]);

Route::put("/updateadopter", [AdopterController::class, "updateAdopter"]);

Route::get("adoptions", [AdoptionController::class, "getAdoptions"]);

Route::post("newadoption", [AdoptionController::class, "newAdoption"]);

Route::put("updateadoption", [AdoptionController::class, "updateAdoption"]);

});

## 5. Technológiai stack

* **Frontend:** Angular 18, Bootstrap, TypeScript
* **Backend:** Laravel, PHP 8+, Laravel Sanctum (auth)
* **Adatbázis:** MySQL
* **Külső szolgáltatások:** Email küldés (Laravel Mail), képmegjelenítés

## 6. Fejlesztési környezet

* **Frontend:** http://localhost:4200
* **Backend:** http://localhost:8000
* Minden adatkommunikáció REST API-n keresztül történik.
* Token-alapú hitelesítés.

## 7. Függelék

### A. Navigációs térkép felhasználói jogosultságok szerint

A webalkalmazás felhasználói különböző szerepkörökhöz kötődően eltérő menüpontokat és funkciókat érhetnek el:

#### Nem bejelentkezett felhasználó:

* Főoldal
* Állatok listázása és adatlap megtekintése
* Regisztráció és bejelentkezés menüpont
* Keresés (állat neve vagy típusa szerint)

#### Bejelentkezett felhasználó (isAdmin = 0):

* Elér minden publikus oldalt
* Profil menü megfelelő almenüpontja alatt kedvencnek jelölt állatok megtekintése, törlés a kedvencek közül, lefoglalt idpontjainak kezelése (lekérdezés, módosítás, törlés)
* Kedvencek kezelése (hozzáadás, eltávolítás, listázás)
* Időpontfoglalás űrlap
* Örökbefogadás indítása

#### Admin (isAdmin = 1):

* Minden felhasználói funkció
* Admin menüpont
* Állatok kezelése (valamennyi állat lekérdezése, új állat hozzáadása, meglévő módosítása, törlése)
* Örökbefogadók, örökbefogadások kezelése
* Lefoglalt időpontok listázása és törlése
* Jogosultságkezelés (csak superusernek)

#### Superuser (isAdmin = 2):

* Minden admin funkció
* Plusz lehetőség: admin jogosultság adása más felhasználóknak

Az alkalmazás dinamikusan jeleníti meg a menüpontokat a bejelentkezett felhasználó szerepköre alapján.

Összefoglaló táblázat az egyes jogosultságok birtokában elérhető funkciókról

| Funkció / Jogosultság | Regisztráció és bejelentkezés nélkül | Bejelentkezett felhasználó | Admin | Superuser |
| --- | --- | --- | --- | --- |
| Regisztráció, bejelentkezés |  | – | – | – |
| Örökbefogadható állatok listázása, adatlap megtekintés |  |  |  |  |
| Saját profil alatti menük elérése | – |  |  |  |
| Kedvencek kezelése (kiválasztás és törlés) | – |  |  |  |
| Időpontfoglalás | – |  |  |  |
| Saját lefoglalt időpont kezelése (lekérdezés, módosítás, törlés) | – |  |  |  |
| Állatok hozzáadása / szerkesztése | – | – |  |  |
| Állatok törlése | – | – |  |  |
| Örökbefogadók és örökbefogadások kezelése(lekérdezés, felvitel, módosítás, törlés) | – | – |  |  |
| Bármely felhasználó által lefoglalt időpontok listázása, törlése | – | – |  |  |
| Admin jogosultság kiosztása | – | – | – |  |

### B. API végpontok

A REST API végpontjai hitelesítés szerint vannak szétválasztva. Lásd a 4. pontban („Backend API végpontok”) részletesen.

### C. Példa email értesítés – Admin által törölt időpont esetén

Az alábbi sablont használja a rendszer, ha egy adminisztrátor töröl egy felhasználó által foglalt időpontot:

**Tárgy:** Értesítés időpont törléséről

Tisztelt Felhasználó!

Az Ön által lefoglalt időpontot (**2025. április 18. 14:00**) az állatmenhely adminisztrátora törölte.

Kérjük, látogasson el weboldalunkra, és válasszon új időpontot.

Köszönjük megértését!

Üdvözlettel:  
Az Állatmenhely csapata

## **FELHASZNÁLÓI DOKUMENTÁCIÓ**

## 

## Funkciók és használat

Az alkalmazás alább részletezett funkciói lehetővé teszik a felhasználók számára, hogy kutyákat és macskákat találjanak, regisztráljanak, időpontot foglaljanak, és kedvenc állataikat kezeljék. Az alkalmazás a Laravel backend és az Angular frontend segítségével valósul meg.

**1.1 Állatok listázása és kedvencek**

A felhasználók könnyen listázhatják az elérhető kutyákat és macskákat a menhelyen. A saját profiljukba beállíthatnak kedvencként korlátozás nélkül állatokat, hiszen reális, hogy a valódi életben több állat is megtetszik valakinek és sokáig keresgéli a megfelelő négylábú társat.

* 1. **Állatok részletes profilja**

A felhasználók kattintással megtekinthetik az egyes állatok részletes profilját, amely tartalmazza az állat nevét, típusát, nemét, leírását és egyéb fontos információkat. Az állatok profiljai dinamikusan töltődnek be a backendről, és az adatok Angular komponens segítségével jelennek meg.

* 1. **Regisztráció és bejelentkezés**

A regisztráció és bejelentkezés lehetőséget ad a felhasználók számára, hogy saját fiókot hozzanak létre és bejelentkezzenek. A felhasználói jogosultságok lehetővé teszik a felhasználók számára, hogy kedvenceket jelöljenek, időpontot foglaljanak, és egyéb interakciókat végezzenek az oldalon.

* 1. **Felhasználói fiók létrehozása**

A regisztrációs űrlap segítségével a felhasználók egyszerűen regisztrálhatnak az oldalon. Az űrlap tartalmazza a felhasználó nevét, email címét és jelszavát, amelyet a backend segítségével mentünk el a Laravel adatbázisban. A sikeres regisztráció után a felhasználó bejelentkezhet a rendszerbe.

* 1. **Bejelentkezés és felhasználói jogosultságok**

A bejelentkezési folyamat során a felhasználó megadja az email címét és a jelszavát. A rendszer hitelesíti a felhasználót, és hozzáférést biztosít az adott fiókhoz, valamint a megfelelő jogosultságokhoz. Az Angular segítségével biztosítjuk a felhasználói élményt, miközben a backend Laravel biztosítja a hitelesítést és az adatkezelést. A bejelentkezés sikeressége után a felhasználó hozzáférést kap a kedvencekhez és az időpontfoglalás lehetőségéhez.

* 1. **Időpontfoglalás**

A felhasználók számára lehetőség van időpontot foglalni egy állat megtekintésére. Az időpontfoglalási funkció az alkalmazás egyik kulcsfontosságú eleme, mivel lehetővé teszi, hogy a felhasználók előre meghatározott időpontok között válasszanak.

**1.6.2 Időpontok elérhetősége**

A rendszerben a felhasználók elérhetik az állatok megtekintésére szabad időpontokat. Az Angular frontend segítségével a felhasználók egy egyszerű űrlapon választhatják ki a kívánt dátumot és időpontot. A backend pedig biztosítja, hogy az időpontok csak akkor választhatóak, ha azok elérhetők, így elkerülhetőek az ütközések.

**1.6.3 Időpont törlése az admin által**

Amikor egy felhasználó kiválasztott időpontját az adminoknak valamilyen okból (pl. a kiválasztott állatot időközben örökbefogadták) törölni kell, akkor erről a rendszer a felhasználót e-mailben értesíti.

* 1. **Kedvenc állat hozzáadása a profilhoz**

A felhasználók lehetőséget kapnak arra, hogy kedvenc állatokat jelöljenek a profiljukban. Ezzel könnyen nyomon követhetik, mely állatokat találták érdekesnek, és később gyorsan hozzáférhetnek hozzájuk, és akár lefoglalhatják a személyes találkozó időpontját. Ha mégsem szeretnének egy állatot a felhasználók, a korábban kedvencnek kiválasztott állat könnyen eltávolítható a listából.

Ehhez a tartalmi adatokat a Laravel backend és a feltöltött MySQL adatbázis biztosítja, az Angular frontend pedig az interakciós élményt.

## UI/UX

A weboldal kialakításának célja egy letisztult, könnyen használható és reszponzív felület biztosítása, amely megkönnyíti a felhasználók számára az állatok profiljának megtekintését, az időpontfoglalást és az egyéb interakciókat. A tervezés során a felhasználóbarát navigációra is figyelmet fordítottunk.

* 1. **Weboldal dizájn és navigáció**

A weboldal főbb részei:

**Főoldal** – bemutatja az oldalt, kiemeli a legfrissebb örökbefogadható állatokat.

**Állatok listázása** – az örökbefogadható állatok listája.

**Állat profilolja** – részletes információk egy adott állatról, képekkel és örökbefogadási lehetőségekkel.

**Időpontfoglalás** – online lehetőség személyes látogatás vagy örökbefogadási találkozó foglalására.

**Felhasználói fiók** – a regisztrált felhasználók számára lehetőség a korábbi foglalások, kedvencek és értesítések kezelésére.

* 1. **Reszponzív felület (mobil és desktop nézetek)**

A weboldal reszponzív kialakítása biztosítja, hogy az oldal minden eszközön (mobil, tablet, desktop) megfelelően működjön.

**2.2.1 Mobilbarát kialakítás főbb szempontjai**

* Egyszerűsített menü (hamburger ikon mobilon)
* Nagyméretű gombok az érintőképernyős használathoz
* Képek és szövegek dinamikus méretezése
* Kényelmes görgetési élmény
  + 1. **Asztali nézet**
* Teljes navigációs menü a felső sávban
* Rácsos elrendezés az állatprofilok megjelenítéséhez

* + 1. **Formák és interakciók (időpontfoglalás, regisztráció)**

A weboldalon a következő interaktív elemek találhatók:

**Regisztráció és bejelentkezés**

* Az oldalra érvényes (egyedi) felhasználónévvel, e-mailel és jelszóval lehet regisztrálni.

**Kedvencek jelölése**

* Minden állat külön kártyával rendelkezik, ahol a megfelelő gomb megnyomásával kedvencek közé emelhető.
* A kedvencek összesített megtekintésére és a kedvencek törlésére a profil menü kedvenceim almenüben van lehetőség.

**Időpontfoglalás**

* A felhasználók kiválaszthatják a számukra megfelelő időpontokat, és online foglalhatnak.
* Foglalás visszaigazolása a sikeres foglalást követően azonnal megtörténik, a profil menü foglalásaim almenü alatt nyomonkövethető és módosítható.

* + 1. **Állatok profiljának megjelenítése**

Minden állatnak saját adatlapja van, amely a következő információkat tartalmazza:

* **Kép** – fénykép az állatról.
* **Alapadatok** – név, faj, vélelmezett születési idő, menhelyre bekerülés dátuma, méret.
* **Leírás** – egy rövid történet az állatról és a személyiségéről.
* **Egészségügyi információk** – oltások, különleges igények.
* **Kedvencek és időpont foglalás** – közvetlen lehetőség az érdeklődés jelzésére.

A felhasználók a kedvenc állataikat hozzárendelhetik a saját fiókjukba, így később könnyen visszatérhetnek hozzájuk.

## **Jövőbeli fejlesztések**

Tervezett funkciók

**1. Chat funkció (Örökbefogadók és menhelyek között)**

* Beépített üzenetküldő rendszer a felhasználók és a menhelyek számára.
* Automatikus válaszok és gyors kérdések (pl. „Mikor elérhető az állat?”).
* Moderálás és biztonsági szűrők.

**2. Állatok értékelése és visszajelzések**

* Azok a felhasználók, akik egy adott állatot megismertek, azonban mégsem sikerült az örökbefogadás, visszajelzést adhatnak az állatról (pl. „Nagyon kedves és játékos kutya”).
* Más érdeklődők számára segít jobban megismerni az állat személyiségét.
* Menhelyek számára hasznos információ az örökbefogadások utókövetéséhez.

**3. Szűrési lehetőségek**

* Szűrés az állatok fajtája vagy külleme - (pl. németjuhász kutya, cirmos macska ), mérete és kora alapján.
* Viselkedési jellemzők szerinti szűrés (pl. „barátságos más állatokkal”, „nyugodt”, „aktív”).
* Kiemelt mentett keresések és értesítések az új, releváns állatokról.

**4. Örökbefogadás nyomon követése**

* A felhasználók nyomon követhetik az örökbefogadás állapotát.
* Értesítések a következő lépésekről és a szükséges dokumentumokról.

**5. Örökbefogadók szerkesztése**

* Pl. adatváltozás esetén

**6. Mobilalkalmazás fejlesztése**

* Natív iOS és Android alkalmazás, amely gyorsabb és kényelmesebb hozzáférést biztosít.
* Push értesítések új állatok megjelenésekor vagy foglalás emlékeztetőkhöz.

## **Tesztelés**

A tesztelést manuálisan végeztük a különböző teszteseteket egy Microsoft Excel táblázatban rögzítettük és megmutatja, hogy hogyan viselkedik a program különböző bevitelek során. Pl. regisztrációkor, milyen hibaüzenet jelenik meg, ha nem megfelelőek a bevitt adatok.

Ezt a táblázatot backend mellett a Githubon lehet elérni tesztelés mappában. A tesztelés Windows 11-es operációs rendszeren lett elvégezve egy laptopon keresztül.

Természetesen a program fejlesztése közben is történt tesztelés pl. az Insomnia program segítségével a backend ellenőrizhető volt, hogy pl. valóban megjelennek-e az állatok stb. amíg a Frontend fejlesztése folyamatban volt.

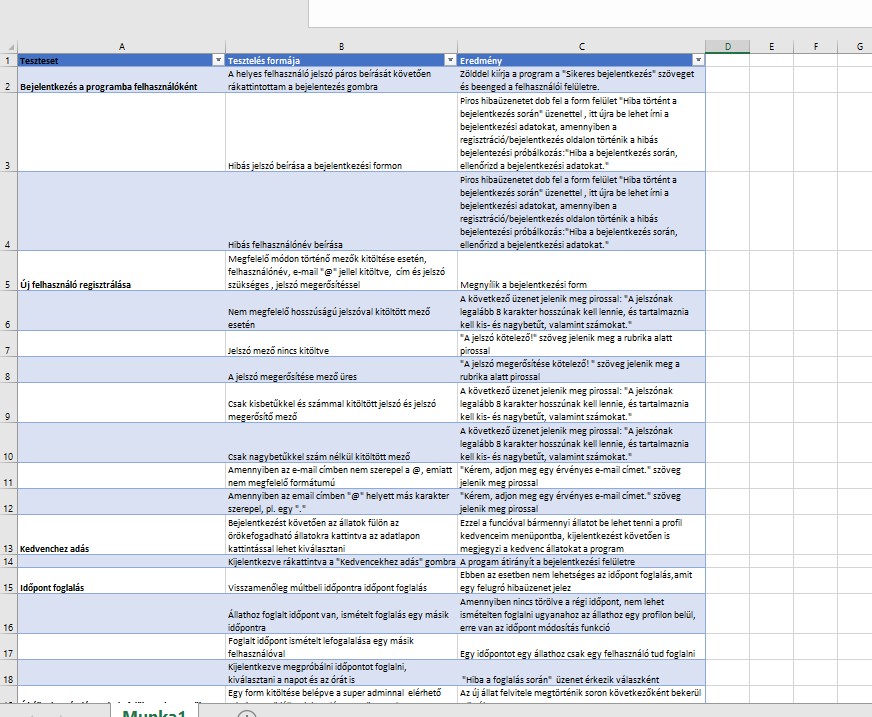
Amennyiben a program végső felhasználókhoz kerül a továbbfejlesztést követően egy terheléses tesztelés elvégzése is szükségessé válik, azonban az iskolai projekt keretében ennek nem láttam szükségességét, mivel egyszerre nem használja több felhasználó a programunkat.

A fent említett táblázatban 22 féle teszteset került manuálisan kipróbálásra és részletezésre került a táblában, továbbá fotódokumentáció is megtalálható hozzá a Githubon ami a következő linkeken keresztül elérhető:

https://github.com/KlauVikiAndi/allatmenhely/blob/main/tesztesetek\_allatmenhely.xlsx

<https://github.com/KlauVikiAndi/allatmenhely/blob/main/tesztesetek_kepernyokepek.zip>

Az alábbi kép szemlélteti a táblázat egy részét.



Köszönjük a figyelmet.