

MancsTárs

Szili Bettina, Rákóczi Viktória, Petrás Kata Magdolna

Technológiák: Angular, Java, WildFly, MySQL

Dátum: 2025. Április

Tartalomjegyzék.....	1
Feladat rövid ismertetése	3
Bevezetés.....	3
Cél.....	3
Használt technológiák.....	4
Programozási nyelvek.....	4
Adatbázis	4
Entity-relációs diagram:.....	4
Egyéb technológia.....	5
Adatbázis	5
Adatbázis rövid magyarázat	5
Fejlesztési módszertan	5
Scrum	5
Program működésének a részletes leírása	6
Program részletes leírása	6
Tesztelés	7
Tesztelési fajták	7
Jelentőség	7
Használt technológia.....	7
Frontend	7
Backend.....	8
Összegzés és továbbfejlesztési lehetőségek	8
Tapasztalatok	8
Továbbfejlesztési lehetőségek	8

Feladat rövid ismertetése

Bevezetés

A MancsTárs menhelyen élő állatoknak segít szerető otthonra találni azzal, hogy ad a menhelyeknek egy nagyobb platformot, aminek segítségével minél több potenciális gazdát tudnak elérni, hiszen sokszor az vehető észre, hogy szinte alig találhatóak meg online, illetve ha mégis, akkor is csak nehézkesen. A lehetséges gazdáknak pedig sokkal könnyebb lenne, hogyha minden menhely megtalálható lenne egy helyen, hiszen egy örökbefogadási rendszer elég körülményes tud lenni, mivel sokszor rengeteget kell utazniuk, hogy meglátogassanak egy menhelyet, hiszen nem mindig található egy a közelben és van, hogy néha többet is meg kell látogatniuk. Sokszor az is észrevehető, hogy az emberek és az állatok szükséglete mindezek után nem találkozik, ami ahhoz is vezethet, hogy az állatokat visszaviszik a menhelyre. A Mancstárs ezekre a problémákra próbál megoldást találni.

Cél

MancsTárs nevezetű projektnek az a célja a menhelyen lévő állatok tudjanak találni maguknak egy megfelelő otthont, ahol gondoskodnak róluk és nem hagyják el őket. Az oldalon minden állathoz tartozik egy kártya, ami olyan információkat tartalmaz róluk, amik könnyebben elősegítik azt, hogy a hozzájuk legjobban illő személyhez kerüljenek, hiszen többek között a fizika tulajdonságaik mellett, a személyiségükről is található leírás, egy pontozási rendszer alapján, így a jövőbeli örökbe fogadó könnyen kitudja választani azt a kisállatot, ami számára a legmegfelelőbb. Ezzel elkerülve azt, hogy a kisállatok és az emberek is csalódással zárják az örökbefogadási folyamatot.

Használt technológiák

Programozási nyelvek

Frontend-keretrendszerként az Angular 19-et és a Bootstrap-et használtuk, TypeScript nyelven, illetve az Angular komponenseinek és nézeteinek a kialakítását HTML-lel és CSS-el készítettük el.

Backend fejlesztéshez és az objektumorientált programozáshoz: A Java programnyelvet választottuk, mert számunkra ez a program nyelv sokkal kényelmesebbnek bizonyult.

Adatbázis-kezelésre a MySQL-t használtuk.

REST API-k használatával kérjük le és jelenítjük meg az adatbázisból érkező állatok és menhelyek adatait.

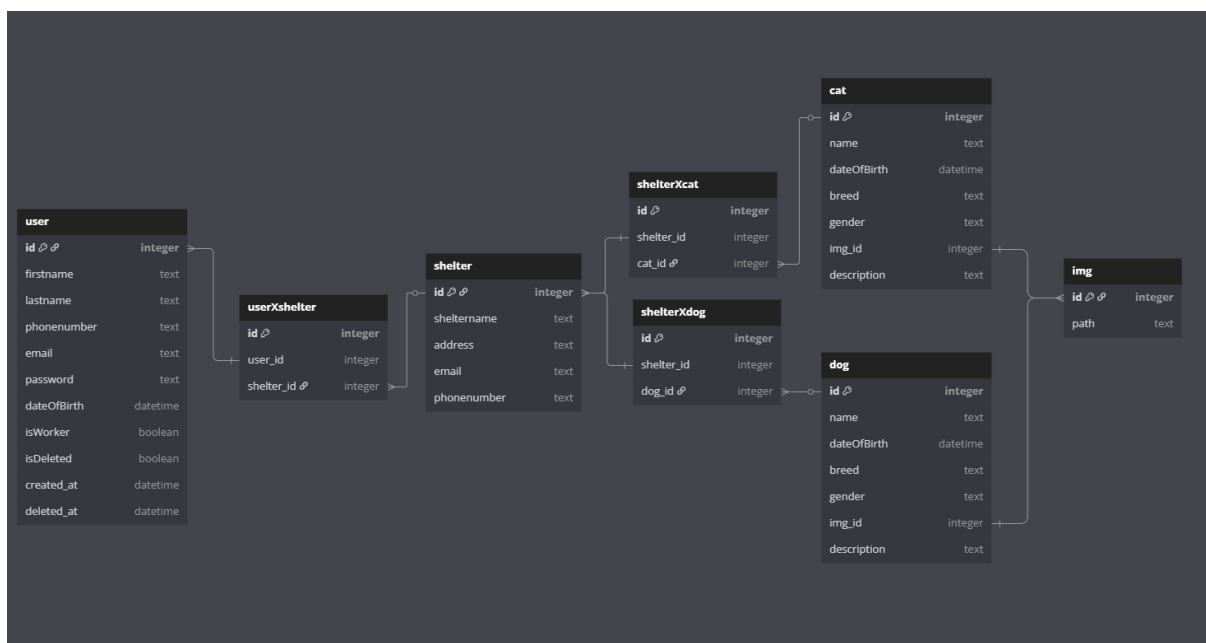
JWT (JSON Web Token) alapú autentikáció segítségével védjük le a felhasználóink személyes adatait, mert a felhasználói biztonság és elégedettség a két fő prioritásunk.

Fejlesztői környezetek, illetve a csomagkezelők pedig az alábbiak voltak: Visual Studio Code, Node.js és Netbeans.

Adatbázis

Relációs adatbázist használunk, amelyet a MySQL kezel, aminek segítségével könnyebben tudjuk az adatokat táblába rendezni, definiálni a köztük lévő kapcsolatot, valamint az adatok lekérdezése is hatékonyabb, a hozzá tartozó szerveret pedig a MAMP segítségével futtatjuk.

Entity-relációs diagram:



Egyéb technológia

Postman-t és Wildfly-t használtunk a REST API teszteléséhez és futtatásához, Bootstrap-et az oldal reszponzivitásának megvalósításához.

Csapatmunkához, legfőképp a Jira-t használtuk, verzió kezeléshez pedig GitHubot használtuk, továbbá az egymással való kommunikációra és a dolgok megosztásához a Google Drive-ot és a Discord-ot vettük igénybe.

A projektünknek a design terve Figmában készült el.

Adatbázis

Adatbázis rövid magyarázat

Adatbázisba először feltöltöttük adatokkal a kutya, macska, felhasználó, menhely táblát utána írtunk hozzá eljárásokat, amik lekérlik az összes kutya, macska, felhasználó és menhely adatait, utána megcsináltuk, hogy új adatokat tudjanak felvenni és ki is tudjanak törölni az álatokkal kapcsolatban az adminok, illetve frissíteni az adatokat. Új jelszó kérést is csináltunk, illetve regisztrálni és bejelentkezni is lehet.

Fejlesztési módszertan

Scrum

A fejlesztés során az Agil megközelítést alkalmaztuk, ezen belül is a Scrum módszertant. Ez tette lehetővé, hogy rugalmas módon haladjunk előre.

A csapatunk három főből állt, és a frontend, backend, adatbázis és a dokumentációs feladatokat osztottuk fel egymás között, még pedig úgy, hogy mindenki vállalta azt, amihez értett vagy fejlődni szeretett volna benne.

A fejlesztést havi sprintekre bontottuk, Minden hónap elején összeállítottuk a sprint célkitűzéseit, meghatároztuk, hogy ilyen feladatokat kell előtérben helyeznünk.

A feladatokat a Jira-ban vezettük, itt hoztuk létre a feladatokat, határidőket állítottunk be, jelöltük a feladatok aktuális állapotát, és követtük a haladásunkat A Jira átláthatóvá tette a csapatunk számára, hogy ki min dolgozott éppen, mi van még hátra és mi készült el.

Habár nem mindennap tartottunk megbeszéléseket, hetente legalább egyszer röviden átbeszéltük, hogy ki mivel haladt, milyen akadályba ütközött ilyen feladatokra fog fókuszálni az elkövetkezendő napokban.

A forráskódot a GitHub segítségével kezeltük, így hatékonyan tudtunk együttműködni hiszen mindenki külön ágon tudott dolgozni, majd a legvégén pull request-tel mindent sikeresen átintegráltunk a fő ágba.

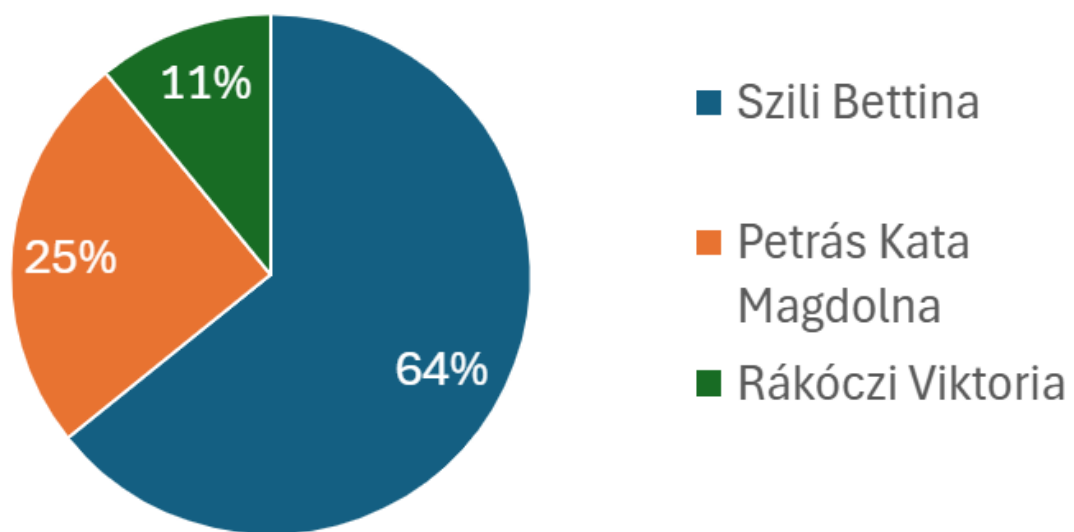
A csapat felosztás pedig a következőképpen nézett ki:

Szili Bettina dolgozott a frontenden, webdesignon, illetve ő volt a csapat project menedzsere,

Petrás Kata Magdolna készítette az adatbázist, illetve az oldalon látható tartalmak is neki köszönhetőek,

Rákóczi Viktória pedig a Backend részért volt felelős.

Órák Eloszlása Csapattagonként



Program működésének a részletes leírása

Program részletes leírása

Ahogy betölt az oldal megjelennek a regisztrációra és bejelentkezésre való opciók. Ha még nincs fiókod, akkor tudsz regisztrálni, és ha sikerül regisztrálnod, akkor be tudsz lépni. Sikeres belépést követően a főoldalra irányít át, ahol található a menüsáv. A menüsávon belül lehet választani, hogy kutyát vagy macskát akarsz örökbe fogadni. Ha nem gépen vagy laptopon vagy, akkor telefonon a hamburger menüt (három horizontális csík a sarokban)

lenyitva kiadja, hogy mik a választási lehetőségeid. Amint kiválasztottad milyen állatot akarsz akkor rá tudsz menni a képükre, ahol felugrik egy részletes leírás az állatról. Ha megtetszik az állat és örökre akarnád fogadni akkor van egy menhely opció és meg tudod nézni, hogy melyik a legközelebbi menhely hozzád, ellátva elérhetőségekkel, hogy örökre tudd fogadni. Ha sikeres az adoptáció, akkor haza viheted az állatod, és biztosíthatod neki egy kényelmesebb életet tele szeretettel és törődéssel.

Tesztelés

Tesztelési fajták

A programozásban két féle tesztelési módszert különítünk el az egyik az automatizált teszt a másik a manuális teszt. Az automatizált teszteket általában olyan funkciók tesztelésére használjuk, amelyeket így egyszerűbben és gyorsabban le tudjuk tesztelni és esetleges bugokat felfedezni, ami eddig nem szűrt szemet. A manuális tesztelés ugyan lassabb, de egyszerűbb elkészíteni.

Jelentőség

Azért nagyon jelentősek a tesztelések, hogy a hibákat még azelőtt észre vegyük, hogy a kész szoftver termékünk üzembe helyezésre kerül, így növeljük annak a minőségét.

Használt technológia

A manuális teszt katalógust Excel táblában készítettük el.

Frontend

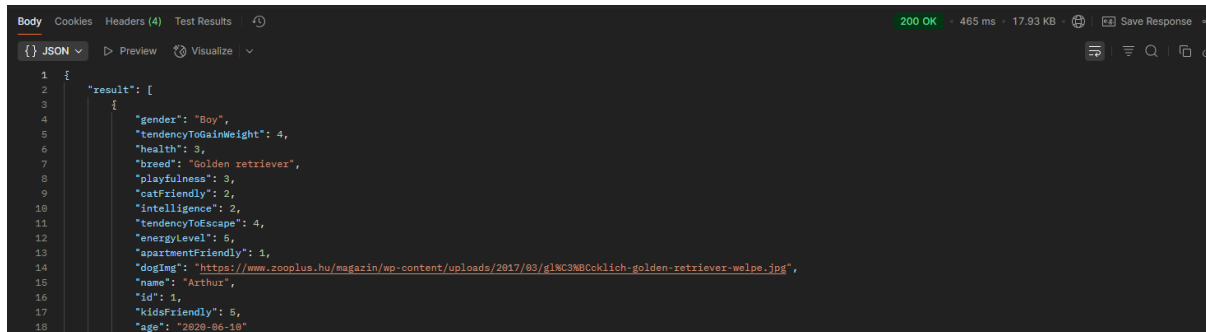
A frontendet kizárólag manuálisan teszteltük, amit a Manuális Tesztelés PDF fájlban lehet megtekinteni részletesebben.

19	scrum-79	regiszter függvények tesztelése	1. Postman megnyitása 2. "POST" kérésbe beírjuk az alábbi linket: http://127.0.0.1:8080/Mancs_Tars-1.0-SNAP/webresources/user/register 3. Megadjuk a Body-n belül raw (JSON) formátumban a megadott user felhasználónevét, email címét és jelszavát 4. API kérés küldése	Lekérés után 200-as HTTP-kódot kapunk vissza	<input type="checkbox"/>	Model-réteg nem éri el az adatbázisból érkező adatokat, így az EntityManager sem tudja listává alakítani azokat
20	scrum-102	regiszter tesztelése frontenden	1. Projekt megnyitása böngészőben 2. "Register" gomb megnyomása 3. Új fiók létrehozása	A "Register" gomb megnyomása után átirányít minket a főoldal "dogs" almenüjére	<input checked="" type="checkbox"/>	
21	scrum-80	getAllDog api tesztelése	1. Projekt megnyitása böngészőben 2. Bejelentkezés egy tetszőleges fiókkal 3. "dogs" almenüre érkezés	A kutyák adatai megjelennek az oldalon	<input checked="" type="checkbox"/>	
22	scrum-81	getAllCat api tesztelése	1. Projekt megnyitása böngészőben 2. Bejelentkezés egy tetszőleges fiókkal 3. "cats" almenü megnyitása	A macskák adatai megjelennek az oldalon	<input checked="" type="checkbox"/>	

Backend

A backendet Postman segítségével teszteltük, aminek eredményeit úgyszintén a Manuális Tesztelés PDF fájlban lehet megtekinteni részletesebben.

Tesztelt endpointok: GET/api/dogs, GET/api/shelters, GET/api/cats



Összegzés és továbbfejlesztési lehetőségek

Tapasztalatok

Nem minden esetben ment jól a kommunikáció, de mindig igyekeztük az ebből adódó vitákat megbeszélni, hogy a munkafolyamatokat ez ne befolyásolja. Rengetegszer elakadtunk, de mindig igyekeztünk segíteni egymásnak. A kezdetek nagyon nehézkesek voltak, de ki szerettük volna hozni a lehető legtöbbet ebből a projectből.

Továbbfejlesztési lehetőségek

Szeretnénk létrehozni egy olyan algoritmust, ami annyit tesz, hogy miután egy felhasználó regisztrál az oldalunkra, akkor szembe találja majd magát egy kitöltendő tesztel, ami többek között olyan kérdéseket tartalmaz, hogy mit keresnek egy kisállatban, illetve amik segítenek megérteni az ő személyiségüket is egy picit, aztán ennek az eredményének a segítségével ajánljunk neki az oldalunk állatokat.