**Szakképesítés neve: Szoftverfejlesztő és -tesztelő**

**OKJ száma: 54 522**

**ÁllatRadar**

Állatkereső és -megtaláló Oldal

2024-2025

**Készítette**:

Sóki Zalán

Páskuj Csaba

Bodrogi Zétény

# **Bevezetés és Célok**

Az ÁllatRadar weboldal egy innovatív digitális megoldás, amelyet az elveszett és talált állatok nyilvántartására, valamint a gazdák és a kisállatok újraegyesítésének elősegítésére fejlesztettünk. Az alkalmazás könnyen használható felülettel rendelkezik, amely lehetővé teszi a felhasználók számára, hogy gyorsan regisztrálhassák elveszett háziállataikat vagy a megtalált állatokat, és így maximalizálják az újraegyesítés esélyeit. A projekt megvalósítása mögött valós társadalmi probléma áll: hazánkban évente több ezer háziállat tűnik el, és a hagyományos keresési módszerek - mint a közösségi média bejegyzések vagy a papír alapú hirdetések - hatékonysága korlátozott. Az ÁllatRadar egy központosított, könnyen kereshető és szűrhető adatbázist kínál, amely összeköti az elveszett állatok gazdáit és a segíteni kívánó embereket. A fejlesztés során kiemelt figyelmet fordítottunk a felhasználói élményre, a keresési folyamat optimalizálására és a biztonságos adatkezelésre. A modern technológiai megoldások - React alapú felhasználói felület, Express.js backend és strukturált adatbázis - biztosítják a gyors és megbízható működést. Az ÁllatRadar célja nem pusztán egy technológiai megoldás létrehozása, hanem egy olyan közösségi platform kialakítása, amely valódi segítséget nyújt a bajba jutott állatoknak és gazdáiknak. A rendszer egyszerű regisztrációs folyamatot, részletes keresési lehetőségeket és közvetlen kapcsolatfelvételi csatornát biztosít a felhasználók között, minimalizálva az elveszett állatok megtalálásához szükséges időt. A weboldalunk küldetése, hogy:

* Hatékony eszközt adjon az állattulajdonosok kezébe elveszett kisállataik megtalálásához
* Platformot biztosítson a megtalált állatok bejelentésére
* Centralizált adatbázist kínáljon, amely növeli az újraegyesítés esélyeit
* Támogassa az állatbarát közösségek kialakulását és együttműködését
* Hosszú távon csökkentse az otthon nélkül maradt állatok számát

# **A. Felhasználói dokumentáció**

## **1. A program általános specifikációja**

Az ÁllatRadar egy modern webalapú alkalmazás, amely az elveszett és talált állatok nyilvántartását, valamint a gazdák és állatok újraegyesítését segíti elő. A rendszer felhasználóbarát felületen keresztül teszi lehetővé a felhasználók számára az elveszett állatok bejelentését, a talált állatok regisztrálását, és az adatbázisban való keresést különböző paraméterek alapján. Az alkalmazás főbb funkciói:

* Biztonságos felhasználói regisztráció és bejelentkezés
* Elveszett állatok regisztrálása részletes adatokkal és képekkel
* Talált állatok bejelentése helyszín és egyéb információk megadásával
* Fejlett keresési lehetőségek az állatok adatbázisában
* Felhasználók közötti közvetlen kommunikáció az állatok újraegyesítése érdekében
* Adminisztrátori funkciók a bejegyzések és felhasználók kezeléséhez
* Reszponzív felhasználói felület, amely minden eszközön megfelelően működik

Az ÁllatRadar célja, hogy egy könnyen használható, hatékony platformot biztosítson, amely összekötő kapocs a segítségre szoruló állatok és az őket segíteni kívánó emberek között.

## **2. Rendszerkövetelmények**

**Hardver követelmények**

Az ÁllatRadar webalapú alkalmazás, amely minimális hardverigénnyel rendelkezik, hiszen a feldolgozás nagy része a szerveren történik.

**Felhasználói oldalon:**

* Bármely eszköz (számítógép, tablet, okostelefon) internet kapcsolattal és webböngészővel
* Minimum 2GB RAM (a böngésző megfelelő működéséhez)
* Ajánlott: 4GB RAM és modern processzor a gördülékeny élmény érdekében
* Minimum 1024x768 képernyőfelbontás (reszponzív design, de ennél kisebb felbontáson nehezebben használható)
* Kamera (opcionális, de ajánlott az állatok fotóinak feltöltéséhez)

**Szoftver követelmények**

**Operációs rendszer:**

* Bármely modern operációs rendszer (Windows 10+, macOS 10.13+, Linux, Android 8.0+, iOS 12.0+)

**Webböngésző (az alábbiak legfrissebb verziója):**

* Google Chrome 90+
* Mozilla Firefox 88+
* Safari 14+
* Microsoft Edge 90+
* Opera 76+

**Egyéb követelmények:**

* JavaScript engedélyezve
* Cookie-k engedélyezve
* Stabil internetkapcsolat
* Frissített böngésző, amely támogatja a modern CSS tulajdonságokat és JavaScript funkciókat

## **3. A program telepítése**

Mivel az ÁllatRadar egy webalapú alkalmazás, a végfelhasználóknak nem szükséges telepíteniük semmilyen szoftvert. Az alkalmazás használatához elegendő egy kompatibilis webböngészővel rendelkező eszköz és internetkapcsolat. A fejlesztői környezet telepítéséhez és a lokális futtatáshoz az alábbi lépések szükségesek:

**1. Előfeltételek telepítése:**

* Node.js (16.x vagy újabb verzió)
* npm (Node Package Manager) vagy yarn
* Git verziókezelő rendszer

**2. Repository klónozása:**

git clone https://github.com/username/allatRadar.git

cd allatRadar

**3. Backend telepítése és indítása:**

cd Back

npm install

npx prisma generate

npm start

**4. Frontend telepítése és indítása:**

cd ../Front

npm install

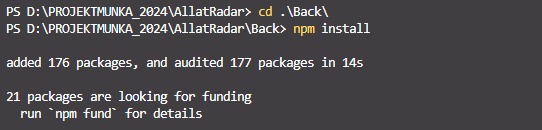
npm run dev

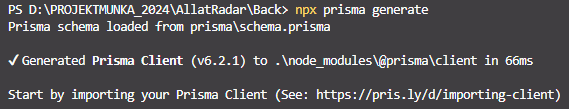
**5. A weboldal elérése:**

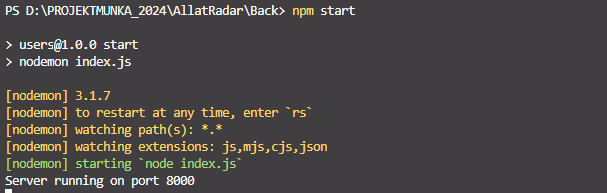
A sikeres telepítés és indítás után a weboldal elérhető lesz a böngészőben a következő címen:

<http://localhost:5173>

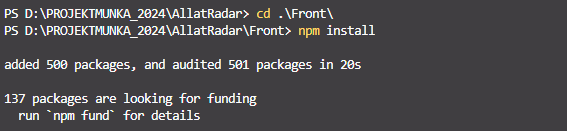
**Backend Telepítés**

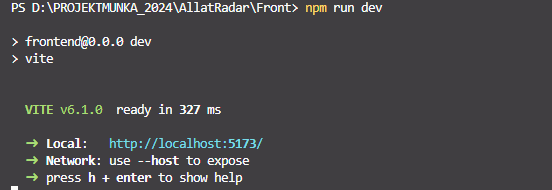






**Frontend Telepítés**





## **4. A program használatának részletes leírása**

### **Regisztráció és bejelentkezés**

**Regisztráció:**

1. Nyissa meg a weboldalt a böngészőjében
2. Kattintson a "Regisztráció" gombra a navigációs sávban
3. Töltse ki a regisztrációs űrlapot a kért adatokkal:

* Felhasználónév
* E-mail cím (érvényes e-mail formátumban)
* Jelszó
* Telefonszám

1. Kattintson a "Regisztráció" gombra az űrlap alján
2. Sikeres regisztráció esetén átirányításra kerül a bejelentkezési oldalra



**Bejelentkezés:**

1. Nyissa meg a weboldalt a böngészőjében
2. Kattintson a "Bejelentkezés" gombra a navigációs sávban
3. Adja meg a felhasználónevét és jelszavát
4. Kattintson a "Bejelentkezés" gombra
5. Sikeres bejelentkezés esetén átirányításra kerül a főoldalra

### **Elveszett állat regisztrálása**

1. Jelentkezzen be a fiókjába
2. Kattintson az "**Elveszett állat regisztrálása**" gombra a navigációs sávban
3. Töltse ki az űrlapot a következő adatokkal:

* **Állatfaj (kötelező):** pl. kutya, macska, madár stb..
* **Állatkategória:** további pontosítás, pl. fajtája
* **Eltűnés dátuma (kötelező):** mikor veszett el az állat
* **Microchip szám (ha van):** a beültetett azonosító chip száma
* **Kisállat neme:** hím/nőstény/ismeretlen
* **Kisállat színe:** az állat fő színe(i)
* **Kisállat mérete:** kicsi/közepes/nagy
* **Eltűnés helyszíne (kötelező):** minél pontosabb cím vagy helymeghatározás
* **Egyéb információk:** különleges ismertetőjegyek, körülmények

1. **Töltsön fel egy képet az állatról (kötelező)** - válasszon egy jó minőségű, felismerhető képet
2. Kattintson az "Adatfelvitel" gombra
3. A rendszer visszaigazolja a sikeres regisztrációt, és tájékoztatja, hogy a bejegyzés adminisztrátori jóváhagyásra vár

****

### **Talált állat regisztrálása**

1. Jelentkezzen be a fiókjába
2. Kattintson a "Talált állat regisztrálása" gombra a navigációs sávban
3. Töltse ki az űrlapot a következő adatokkal:

* **Állatfaj (kötelező):** pl. kutya, macska, madár
* **Állatkategória:** további pontosítás, pl. fajtája (ha felismerhető)
* **Megtalálás dátuma (kötelező):** mikor találta meg az állatot
* **Kisállat színe:** az állat fő színe(i)
* **Kisállat mérete:** kicsi/közepes/nagy
* **Megtalálás helyszíne (kötelező):** minél pontosabb cím vagy helymeghatározás
* **Egyéb információk**: különleges ismertetőjegyek, körülmények

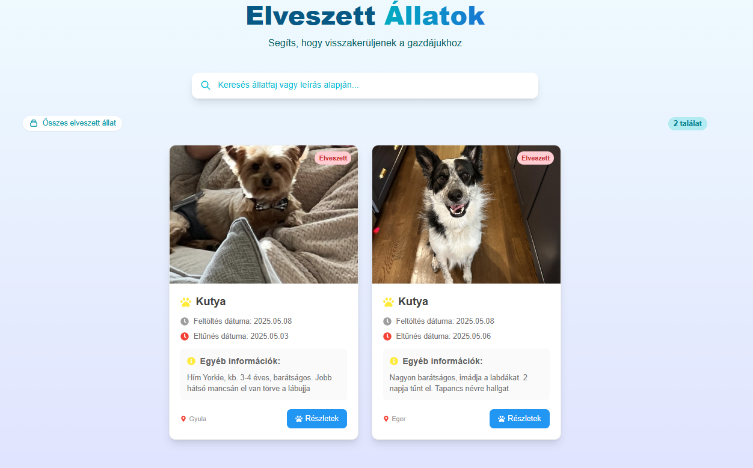
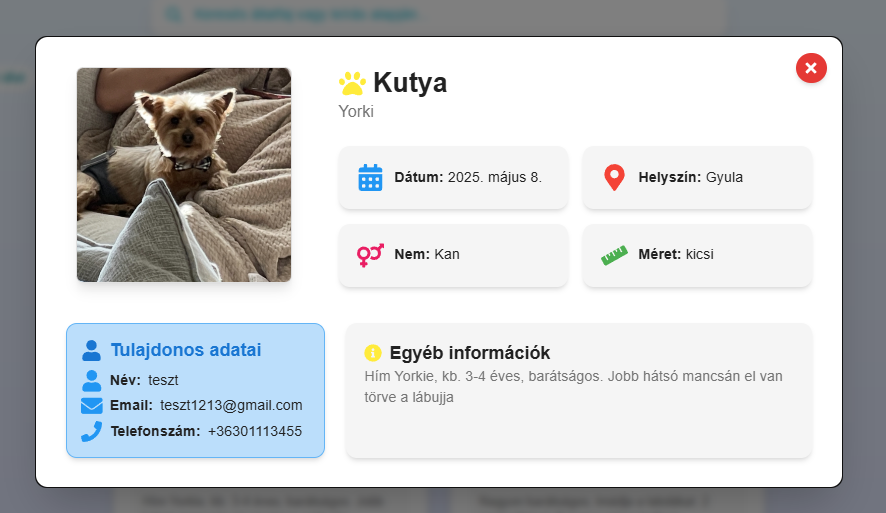
1. Töltsön fel egy képet az állatról (kötelező) - válasszon egy jó minőségű, felismerhető képet
2. Kattintson az "Adatfelvitel" gombra
3. A rendszer visszaigazolja a sikeres regisztrációt, és tájékoztatja, hogy a bejegyzés adminisztrátori jóváhagyásra vár



### **Keresés az állatok között**

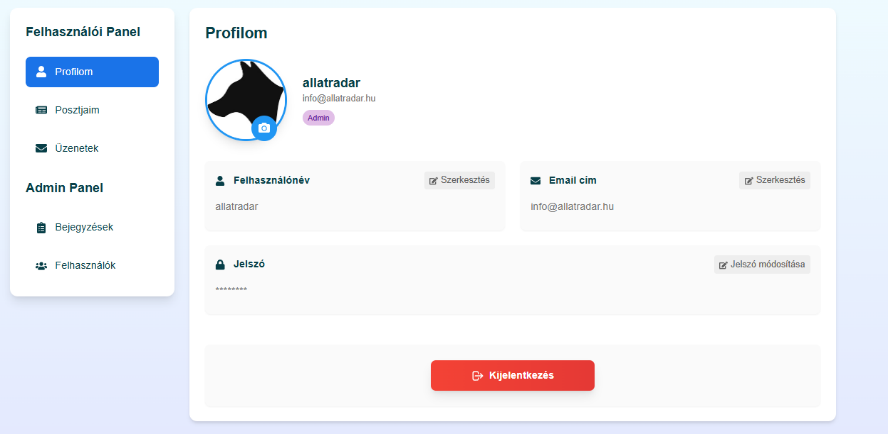
1. Kattintson az "Összes elveszett" vagy "Megtalált" gombra a navigációs sávban
2. Használja a keresési és szűrési opciókat a listázott állatok szűkítéséhez:

* Keresés állatfaj alapján (pl. kutya, macska)

1. Kattintson egy állatra a részletes adatok megtekintéséhez
2. A részletes nézetben láthatja az állat összes adatát és a tulajdonos elérhetőségeit
3. Kapcsolatba léphet a tulajdonossal a megadott elérhetőségeken keresztül

### **Felhasználói profil kezelése**

1. Jelentkezzen be a fiókjába
2. Kattintson a "Profilom" gombra a navigációs sávban
3. A profil oldalon a következő műveleteket végezheti el:

* Személyes adatok módosítása
* Jelszó megváltoztatása
* Profilkép feltöltése vagy módosítása
* Saját bejegyzések megtekintése
* Üzenetek kezelése

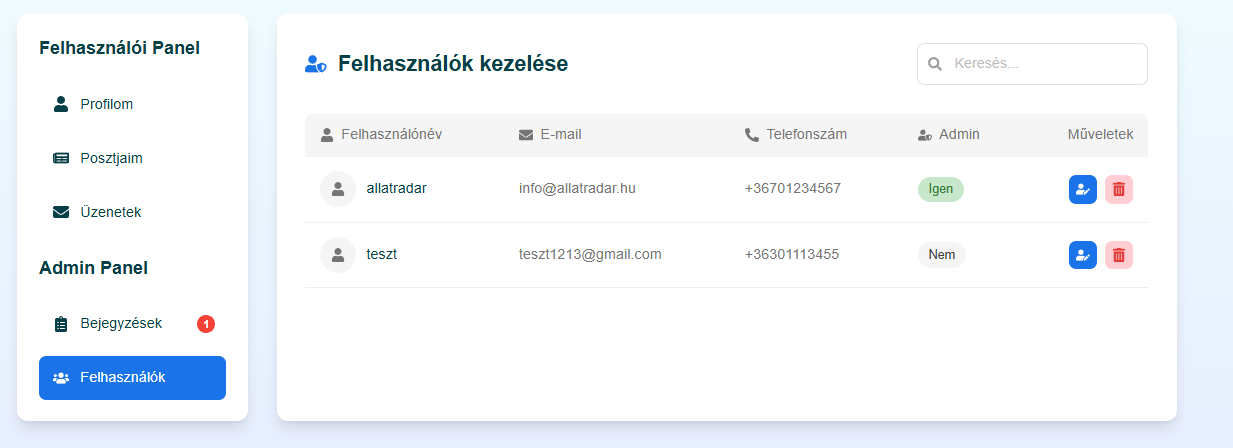
### **Adminisztrátori funkciók**

1. Jelentkezzen be egy adminisztrátori fiókkal
2. Kattintson az "Admin Panel" gombra a navigációs sávban
3. Az admin panelen két fő szekció található:

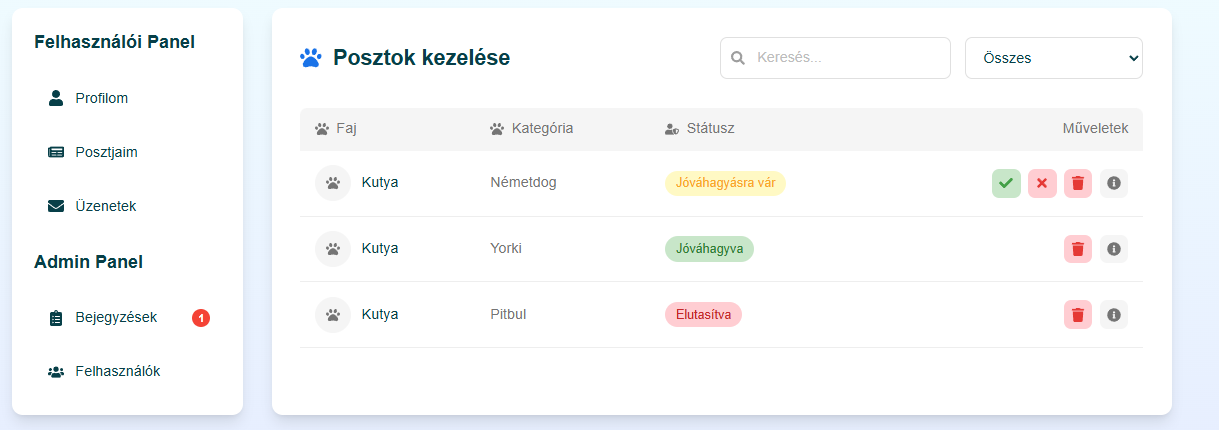
* **Felhasználók kezelése**
* **Állatok / Posztok kezelése**

1. **Felhasználók kezelése:**

* Felhasználók listájának megtekintése
* **Felhasználói adatok szerkesztése** (e-mail, felhasználónév, telefonszám, admin jogosultság)
* **Felhasználók törlése**



1. **Állatok / Posztok kezelése:**

* Várakozó, jóváhagyott és elutasított állatok listájának megtekintése
* **Állatok** **részletes adatainak megtekintése**
* **Állatok jóváhagyása vagy elutasítása**
* Elutasítás esetén indoklás megadása
* **Állatok törlése**

# **B. Fejlesztői dokumentáció**

## **1. Témaválasztás indoklása**

Az állatmenhely weboldal fejlesztésének ötlete egy valós, társadalmi problémára adott válaszként született meg. Magyarországon évente több ezer háziállat tűnik el, és a hagyományos keresési módszerek - közösségi média bejegyzések, plakátok kihelyezése - gyakran nem elég hatékonyak. A témaválasztás fő motivációja az volt, hogy létrehozzunk egy centralizált, könnyen használható digitális platformot, amely segíti az elveszett állatok és gazdáik újraegyesítését. A projekt lehetőséget adott a modern webes technológiák gyakorlati alkalmazására, miközben valós társadalmi értéket teremt. Az állatmentésben résztvevő szervezetek és önkéntesek munkájának támogatása, valamint az állattulajdonosok segítése olyan cél, amely nemcsak szakmai kihívást jelentett, hanem személyes elégedettséget is nyújtott a fejlesztői csapat számára.

## **2. Az alkalmazott fejlesztői eszközök**

A projekt fejlesztése során a következő eszközöket és technológiákat alkalmaztuk:

**Frontend fejlesztés**

* **React**: Modern JavaScript könyvtár a dinamikus felhasználói felület kialakításához
* **Vite**: Gyors frontend build eszköz, hot module replacement funkcióval
* **Tailwind CSS**: Utility-first CSS keretrendszer a reszponzív és modern felhasználói felület kialakításához
* **React Router**: Kliensoldali útvonalkezelés a single-page alkalmazásban
* **React Icons**: Ikonkészlet a vizuális elemek gazdagításához
* **React Toastify**: Felhasználói visszajelzések és értesítések megjelenítésére
* **React Datepicker**: Dátumválasztó komponens implementálására
* **Axios**: HTTP kérések kezelésére a frontend és backend között
* **React Leaflet**: Térképes megjelenítés az állatok eltűnési és megtalálási helyeinek vizualizálásához
* **Context API**: Alkalmazás szintű állapotkezelésre (felhasználói bejelentkezés, témakezelés)

**Backend fejlesztés**

* **Node.js**: Szerver oldali JavaScript futtatókörnyezet
* **Express.js**: Web alkalmazás keretrendszer a RESTful API kialakításához
* **Prisma ORM**: Adatbázis-kezelő objektum-relációs leképező
* **SQLite**: Könnyű, fájl-alapú adatbázis-kezelő rendszer
* **JWT (JSON Web Token)**: Felhasználói hitelesítés és jogosultságkezelés
* **Argon2**: Modern, biztonságos jelszó titkosító algoritmus
* **Multer**: Fájlfeltöltés kezelésére (állatok képeinek tárolása)
* **Express CSP Header**: Content Security Policy implementálása

**Fejlesztői környezet és eszközök**

* **Visual Studio Code**: Kódszerkesztő
* **Git**: Verziókezelő rendszer
* **npm**: Csomagkezelő a függőségek kezelésére
* **ESLint**: Kód minőségellenőrző és formázó eszköz
* **Postman**: API tesztelésre

## **3. Adatmodell leírása**

Az ÁllatRadar adatmodellje három fő entitást tartalmaz, amelyek között relációs kapcsolatok állnak fenn:

A képen szöveg, képernyőkép, szám, szoftver látható

Lehet, hogy az AI által létrehozott tartalom helytelen.

**Felhasználó (User)**

* **id:** Egyedi azonosító (autoincrement egész szám)
* **email:** E-mail cím (egyedi érték)
* **username:** Felhasználónév
* **password:** Jelszó (titkosított formában tárolva)
* **phonenumber:** Telefonszám
* **admin:** Admin jogosultság jelzője ("true"/"false" string)
* **profilePicture:** Profilkép elérési útja (opcionális)
* **animals:** Kapcsolat az Állat entitással (egy-több kapcsolat)

**Állat (Animal)**

* **id:** Egyedi azonosító (autoincrement egész szám)
* **talalt\_elveszett:** Állat státusza ("talált" vagy "elveszett")
* **allatfaj:** Az állat faja (pl. kutya, macska)
* **kategoria:** Az állat kategóriája/fajtája (opcionális)
* **datum:** Feltöltés dátuma
* **elveszesIdeje:** Az eltűnés vagy megtalálás pontos ideje (opcionális)
* **neme:** Az állat neme (alapértelmezett: "Ismeretlen")
* **szin:** Az állat színe
* **meret:** Az állat mérete
* **egyeb\_info:** Egyéb információk (opcionális)
* **helyszin:** Az eltűnés vagy megtalálás helyszíne
* **filePath:** Az állatról készült kép elérési útja
* **visszakerult\_e:** Visszakerült-e már az állat a gazdájához
* **chipszam:** Microchip szám (opcionális, nagy egész szám)
* **elutasitva:** Az állat bejegyzésének státusza (üres, "jóváhagyva", "elutasítva")
* **elutasitasoka:** Elutasítás esetén az indoklás
* **visszajelzes:** Visszajelzés (alapértelmezett: üres string)
* **userId:** A felhasználó azonosítója, aki a bejegyzést létrehozta (idegen kulcs)

**Üzenet (Message)**

* **id:** Egyedi azonosító (autoincrement egész szám)
* conten**t:** Az üzenet tartalma
* **createdAt:** Az üzenet létrehozásának időpontja
* **senderId:** A küldő felhasználó azonosítója
* **receiverId:** A címzett felhasználó azonosítója

**Kapcsolatok**

* **Felhasználó - Állat**: Egy-Több kapcsolat. Egy felhasználóhoz több állat is tartozhat, de egy állat csak egy felhasználóhoz.
* **Felhasználók közötti üzenetek**: Az üzenet modell tárolja a küldő és a címzett azonosítóját, így lehetővé téve az adminoknak a poszt elutasításának indokát jelezni üzenetben a felhasználók számára

## **4. Részletes feladatspecifikáció, algoritmusok**

**Felhasználói regisztráció folyamata**

1. Adatok bekérése a felhasználótól (frontend)
2. Adatok validálása a kliens oldalon (frontend)
3. Adatok küldése a szervernek (API kérés)
4. Szerver oldali validáció:

* **E-mail formátum ellenőrzése**
* **Felhasználónév ellenőrzése**

1. E-mail egyediségének ellenőrzése az adatbázisban
2. Jelszó titkosítása Argon2 algoritmussal
3. Felhasználó adatainak mentése az adatbázisban
4. JWT token generálása és visszaküldése a kliensnek
5. Sikeres regisztráció jelzése a felhasználónak

**Bejelentkezési folyamat**

**1. Felhasználó\_Keresése:**

#### SQL: SELECT \* FROM users WHERE username = [bemeneti\_felhasználónév]

**2. Ellenőrzés, hogy létezik-e a felhasználó**

HA felhasználó **NEM** létezik AKKOR:

* VISSZATÉRÉS "Hibás felhasználónév vagy jelszó" hibaüzenettel

**VÉGE**

**3. Jelszó ellenőrzése**

JelszóEgyezik = ELLENŐRZÉS argon2.verify(felhasználó.jelszó, beérkezett jelszó)

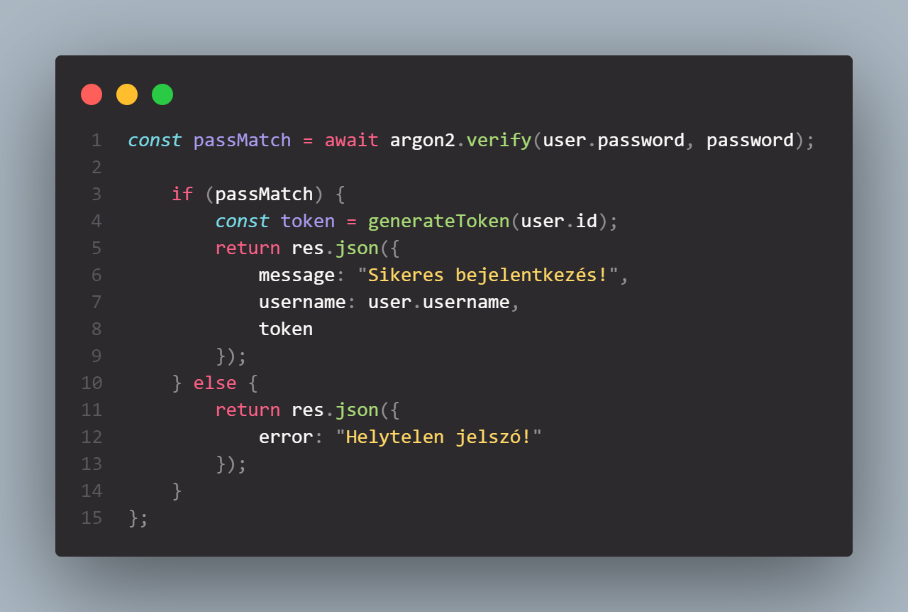
**HA jelszóEgyezik = FALSE**

* + - VISSZATÉRÉS "Hibás felhasználónév vagy jelszó" hibaüzenettel

**HA jelszóEgyezik = TRUE**

* + - VISSZATÉRÉS "Sikeres Bejelentkezés" üzenettel

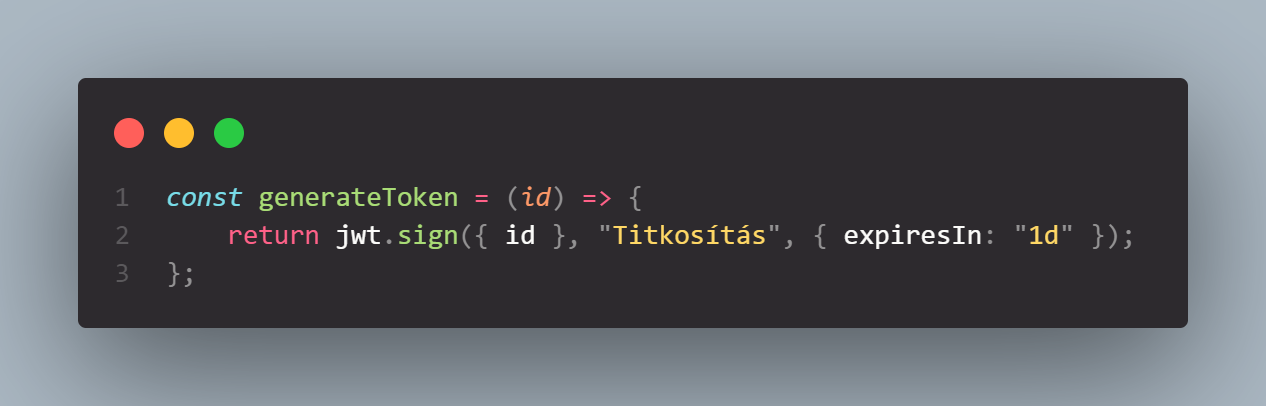
** VÉGE**



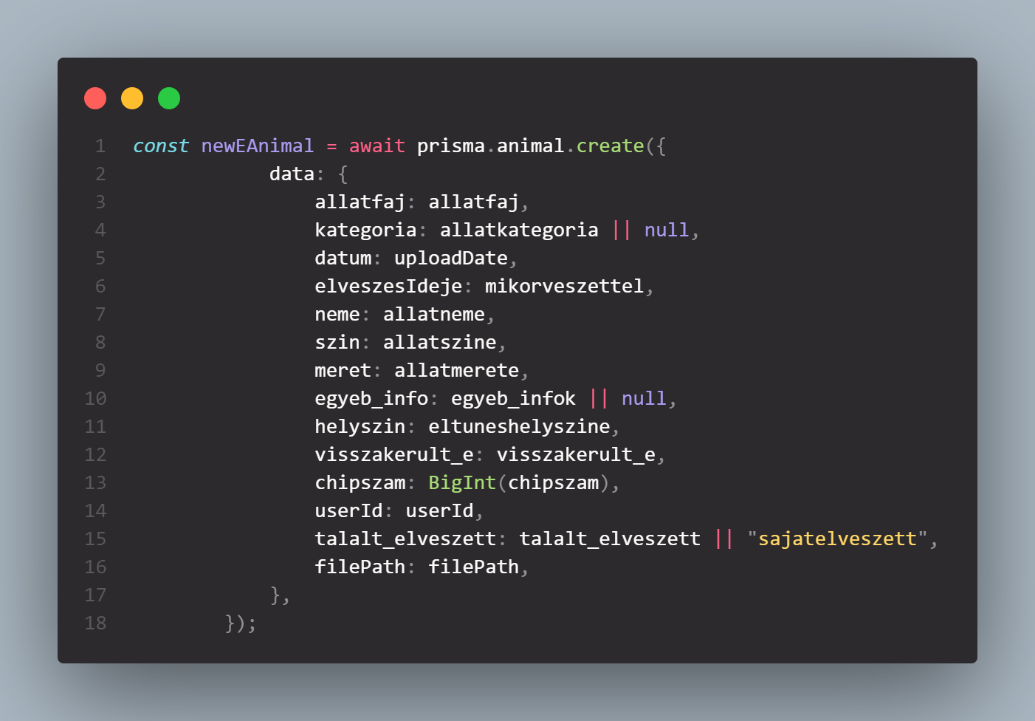
**4. JWT token generálása**

token = JWT.sign({

id: felhasználó.id,

 }, titkosKulcs, lejáratiIdő)

**Állat regisztrálása (elveszett vagy talált)**

1. Adatok bekérése a felhasználótól (frontend)
2. Kép feltöltése
3. Adatok validálása a kliens oldalon
4. Autentikáció ellenőrzése (JWT token)
5. Adatok küldése a szervernek (multipart/form-data)
6. Kép mentése a szerveren a Multer middleware segítségével
7. Állat adatainak validálása a szerveren
8. Állat adatainak mentése az adatbázisban "várakozó" státusszal
9. Sikeres regisztráció visszaigazolása

**FUNCTION állatkereső(szűrőFeltételek)**

**1. Alapvető lekérdezés felépítése**

#### SQL: "SELECT \* FROM Animal WHERE elutasitva != 'elutasitva'"

**2. Szűrés típus alapján (elveszett/talált)**

HA szűrőFeltételek.típus **NEM üres** AKKOR

lekérdezés += " AND talalt\_elveszett = " + szűrőFeltételek.típus

VÉGE

**3. Lekérdezés végrehajtása**

eredmények = VÉGREHAJT lekérdezés

**4. Képek elérési útjának korrigálása**

MINDEN eredmény AZ eredmények-ben

##### eredmény.filePath = szerverCím + "/" + eredmény.filePath

**VÉGE**

5. Visszatérés az eredményekkel

##### VISSZATÉRÉS eredmények

**VÉGE**

**Adminisztrátori jóváhagyás folyamata**

1. Adminisztrátor bejelentkezik
2. Várakozó posztok lekérdezése az adatbázisból
3. Bejegyzés részletes adatainak megtekintése
4. Jóváhagyási döntés meghozatala

* Jóváhagyás esetén: státusz módosítása **"jóváhagyva"-ra**
* Elutasítás esetén: státusz módosítása **"elutasítva"-ra**, elutasítási ok megadása

1. Adatbázis frissítése a döntés alapján



## **5. Tesztelési dokumentáció**

A tesztelés során különböző teszteseteket vizsgáltunk a rendszer megfelelő működésének biztosítása érdekében. A tesztelést mind fekete doboz (black box), mind fehér doboz (white box) módszerekkel végeztük.

### **1. Teszteset: Felhasználói regisztráció hibás adatokkal**

**Teszt célja:** Ellenőrizni a regisztrációs űrlap validációs funkcióit hibás adatok esetén.

**Tesztelési módszer:** Fekete doboz tesztelés

**Teszt lépései:**

1. A regisztrációs űrlap megnyitása
2. Hibás adatok megadása:

* **Érvénytelen e-mail cím formátum (hiányzó @ karakter)**
* **Telefonszám betűkkel**

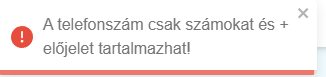
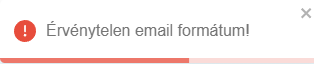
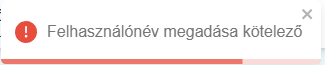
1. Regisztráció gomb megnyomása

**Várt eredmény:** Az űrlap nem engedi a beküldést, és a megfelelő hibaüzenetek jelennek meg minden hibás mező alatt.

**Tényleges eredmény:** Az űrlap megfelelően validálta a beviteli mezőket, és a várt hibaüzenetek megjelentek az érvénytelen mezők alatt:

* **"Kérjük adjon meg egy érvényes e-mail címet"**
* **"A telefonszám csak számokat tartalmazhat"**

**Következtetés:** A frontend validáció megfelelően működik.



### **Teszteset: Elveszett állat regisztrálása és adminisztrátori jóváhagyás**

**Teszt célja:** Ellenőrizni az elveszett állat regisztrálási folyamatát és az adminisztrátori jóváhagyási mechanizmust.

**Tesztelési módszer:** Fehér doboz tesztelés

**Teszt lépései:**

1. Bejelentkezés egy normál felhasználói fiókkal
2. Az "Elveszett állat regisztrálása" oldal megnyitása
3. Adatok kitöltése és kép feltöltése:

* Állatfaj: "kutya"
* Kategória: "labrador"
* Eltűnés dátuma: 2023-04-15
* Egyéb adatok kitöltése

1. Adatfelvitel gomb megnyomása
2. Kijelentkezés
3. Bejelentkezés egy adminisztrátori fiókkal
4. Admin panel megnyitása
5. Várakozó bejegyzések megtekintése
6. Az új bejegyzés jóváhagyása

**Várt eredmény:**

* A felhasználó sikeresen bejelenti az elveszett állatot
* A bejegyzés "várakozó" státusszal kerül be az adatbázisba
* Az adminisztrátor látja a várakozó bejegyzést
* Jóváhagyás után a bejegyzés megjelenik a nyilvános listában

**Tényleges eredmény:**

* Az elveszett állat adatai sikeresen mentésre kerültek az adatbázisban
* A feltöltött kép megfelelően tárolódott a szerveren
* Az adminisztrátori felületen megjelent a várakozó bejegyzés
* A jóváhagyás után a bejegyzés azonnal megjelent a nyilvános listában
* A felhasználói felületen megfelelően megjelentek az állat adatai és képe

**Következtetés:** Az elveszett állat regisztrálási folyamata és az adminisztrátori jóváhagyási mechanizmus megfelelően működik. A feltöltött adatok helyesen tárolódnak, és a jogosultságkezelés is a tervezett módon működik.

### **3. Teszteset: Keresési funkció tesztelése**

**Teszt célja:** Ellenőrizni a keresési és szűrési funkciók hatékonyságát és pontosságát.

**Tesztelési módszer:** Fekete doboz tesztelés

**Teszt lépései:**

1. A "Talált állatok" oldal megnyitása
2. Keresési feltételek beállítása:

* Állatfaj: "macska"
* Helyszín: "Debrecen"
* Időszak: az elmúlt 30 nap

1. Keresés indítása
2. Eredmények ellenőrzése

**Várt eredmény:**

* A rendszer csak azokat a macskákat jeleníti meg, amelyeket Debrecenben találtak az elmúlt 30 napban.

**Tényleges eredmény:**

* A rendszer helyesen szűrte az adatbázist, és csak a keresési feltételeknek megfelelő találatokat jelenítette meg. A megjelenített állatok mindegyike macska volt, a helyszín Debrecen vagy annak környéke, és mindegyiket az elmúlt 30 napon belül találták.

**Következtetés:** A keresési és szűrési funkciók megfelelően működnek, az adatbázis lekérdezések helyesen adják vissza a szűrt eredményeket.

### **4. Teszteset: Reszponzív megjelenítés különböző eszközökön**

**Teszt célja:** Ellenőrizni, hogy a weboldal megfelelően jelenik-e meg és működik-e különböző képernyőméretekkel rendelkező eszközökön.

**Tesztelési módszer:** Fekete doboz tesztelés

**Teszt lépései:**

1. A weboldal megnyitása különböző eszközökön:

* Asztali számítógép (1920x1080 felbontás)
* Tablet (iPad Air, 1640x2360 felbontás)
* Mobiltelefon (iPhone 12, 1170x2532 felbontás)

1. A főoldal betöltésének és megjelenítésének ellenőrzése minden eszközön
2. Navigáció a különböző oldalak között minden eszközön
3. Űrlapok kitöltése és beküldése minden eszközön

**Várt eredmény:** A weboldal minden eszközön megfelelően jelenik meg, az elemek átrendeződnek a képernyőméretnek megfelelően, és minden funkció használható minden eszközön.

**Tényleges eredmény:**

* Asztali számítógépen: Minden elem megfelelően jelent meg, a teljes tartalom jól látható és használható volt.
* Tableten: Az oldal elrendezése megfelelően alkalmazkodott a kisebb képernyőhöz, a menü és a tartalom átrendeződött, de minden funkció elérhető és használható maradt.
* Mobiltelefon: A navigációs menü hamburger menüre váltott, a tartalom oszlopos elrendezésre váltott, az űrlapok és gombok mérete megfelelően nőtt a könnyebb használhatóság érdekében.

**Következtetés:** A weboldal reszponzív kialakítása megfelelően működik, a Tailwind CSS keretrendszer segítségével az alkalmazás minden eszközön jól használható. Kisebb optimalizációk szükségesek a nagyon kis képernyőméretű eszközökön való megjelenítéshez.

### 5. Teszteset: Adatbiztonsági funkciók tesztelése

**Teszt célja:** Ellenőrizni a webalkalmazás biztonsági funkcióit, különös tekintettel a jogosultságkezelésre.

**Tesztelési módszer:** Fehér doboz tesztelés

**Teszt lépései:**

1. Nem bejelentkezett felhasználóként próbálni elérni a védett oldalakat (pl. "Profilom", "Adminisztráció")
2. Bejelentkezés normál felhasználói fiókkal
3. Kísérlet adminisztrátori funkciókat elérni a böngésző URL-sávjának manipulálásával
4. JWT token manipulálása a localStorage-ben

**Várt eredmény:** A rendszer minden esetben megtagadja a hozzáférést a jogosulatlan műveletekhez és oldalakhoz.

**Tényleges eredmény:**

* Nem bejelentkezett felhasználóként a védett oldalak helyett a bejelentkezési képernyő jelent meg
* Normál felhasználóként az adminisztrátori oldalak elérési kísérletére a rendszer "Hozzáférés megtagadva" üzenetet adott
* A JWT token manipulálására a rendszer érvénytelen token üzenettel válaszolt, és a felhasználót kijelentkeztette

**Következtetés:** A jogosultságkezelés mind a frontend, mind a backend oldalon megfelelően működik. Az adminisztrátori útvonalak védettek a jogosulatlan hozzáférés ellen, és a token-alapú autentikáció hatékonyan biztosítja a biztonságos működést.

# Továbbfejlesztési lehetőségek

A jelenlegi rendszer működőképes és hatékony, de számos továbbfejlesztési lehetőséget azonosítottunk, amelyek növelhetik a funkcionalitást és a felhasználói élményt:

1. **Automatikus értesítési rendszer**: Felhasználók értesítése e-mailben vagy push értesítésekkel, amikor:

* Új, a keresési feltételeiknek megfelelő állat kerül regisztrálásra
* Adminisztrátori jóváhagyás vagy elutasítás történik
* Üzenet érkezik más felhasználótól

1. **Fejlett térképes funkciók**:

* Hőtérkép az eltűnések gyakori helyszíneiről
* Közelben lévő állatorvosok és állatmenhelyek megjelenítése
* Területi szűrési lehetőség kör vagy sokszög rajzolásával

1. **Intelligens keresés és egyeztetés**:

* Gépi tanulási algoritmusok implementálása a hasonló állatok automatikus párosítására
* Képfelismerés a hasonló kinézetű állatok azonosítására
* Fuzzzy keresés a hasonló leírások megtalálására

1. **Közösségi funkciók bővítése**:

* Kommentelési lehetőség a bejegyzésekhez
* Megosztási funkciók közösségi platformokon
* Önkéntesek bevonása: regisztrációs lehetőség azoknak, akik segítenének a keresésben

1. **Mobilalkalmazás fejlesztése**:

* Natív iOS és Android alkalmazások létrehozása
* Offline használati lehetőségek
* Kamera integráció a gyorsabb állat regisztráláshoz
* Push értesítések

1. **Statisztikai elemzések és riportok**:

* Részletes statisztikák a felhasználók, elveszett és megtalált állatok számáról
* Sikerességi ráta mérése és vizualizálása
* Adminisztrátorok számára riportok generálása

1. **Többnyelvű támogatás**:

* Magyar mellett angol és esetleg más nyelvek hozzáadása
* Automatikus fordítási lehetőség a bejegyzésekhez

1. **QR-kód rendszer**:

* Egyedi QR-kód generálása minden regisztrált állathoz
* Nyomtatható QR-kódos biléták, amelyek megkönnyítik az állat azonosítását megtalálás esetén

1. **Szakértői támogatás integrálása**:

* Állatorvosi tanácsadás integrálása a platformba
* Viselkedési szakértők bevonása az elveszett állatok keresési stratégiájának kidolgozásába

1. **Mikroadományozási rendszer**:

* Lehetőség mikroadományok gyűjtésére elveszett állatok megtalálásának támogatásához
* Jutalmak felajánlásának lehetősége megtalálók számára

Ezek a fejlesztési irányok jelentősen bővíthetik az alkalmazás funkcionalitását és hatékonyságát, javíthatják a felhasználói élményt, és növelhetik a platform társadalmi hatását.

# Irodalomjegyzék, forrásmegjelölés

1. React hivatalos dokumentáció: https://reactjs.org/docs/getting-started.html
2. Node.js dokumentáció: https://nodejs.org/en/docs/
3. Express.js útmutató: https://expressjs.com/en/guide/routing.html
4. Prisma ORM dokumentáció: https://www.prisma.io/docs/
5. JWT autentikáció implementálása: https://jwt.io/introduction/
6. Tailwind CSS dokumentáció: https://tailwindcss.com/docs
7. React Router dokumentáció: https://reactrouter.com/
8. Argon2 jelszó titkosító könyvtár: https://github.com/ranisalt/node-argon2
9. Multer dokumentáció: https://github.com/expressjs/multer
10. React Context API használata: https://reactjs.org/docs/context.html
11. Vite build eszköz: https://vitejs.dev/guide/
12. Axios HTTP könyvtár: https://axios-http.com/docs/intro
13. React Toastify könyvtár: https://fkhadra.github.io/react-toastify/introduction/
14. Content Security Policy implementáció: https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP
15. React Leaflet térképkezelő: https://react-leaflet.js.org/
16. ESLint kód minőségellenőrző: https://eslint.org/docs/latest/
17. Jovanović, G., & Jagodić, D. (2022). Web application security in modern development environments. Applied Sciences, 12(3), 1648.
18. Fielding, R. T., & Taylor, R. N. (2000). Architectural styles and the design of network-based software architectures. University of California, Irvine Doctoral dissertation.
19. Nielsen, J. (2020). 10 Usability Heuristics for User Interface Design. Nielsen Norman Group.
20. Sandoval, K. (2019). Best Practices for Authentication and Authorization for REST APIs. Nordic APIs.

Tartalomjegyzék

[Bevezetés és Célok 2](#_Toc197559193)

[A. Felhasználói dokumentáció 3](#_Toc197559194)

[1. A program általános specifikációja 3](#_Toc197559195)

[2. Rendszerkövetelmények 3](#_Toc197559196)

[3. A program telepítése 5](#_Toc197559197)

[4. A program használatának részletes leírása 7](#_Toc197559198)

[Regisztráció és bejelentkezés 7](#_Toc197559199)

[Elveszett állat regisztrálása 8](#_Toc197559200)

[Talált állat regisztrálása 9](#_Toc197559201)

[Keresés az állatok között 10](#_Toc197559202)

[Felhasználói profil kezelése 10](#_Toc197559203)

[Adminisztrátori funkciók 11](#_Toc197559204)

[B. Fejlesztői dokumentáció 12](#_Toc197559205)

[1. Témaválasztás indoklása 12](#_Toc197559206)

[2. Az alkalmazott fejlesztői eszközök 12](#_Toc197559207)

[3. Adatmodell leírása 14](#_Toc197559208)

[4. Részletes feladatspecifikáció, algoritmusok 16](#_Toc197559209)

[5. Tesztelési dokumentáció 21](#_Toc197559210)

[1. Teszteset: Felhasználói regisztráció hibás adatokkal 22](#_Toc197559211)

[2. Teszteset: Elveszett állat regisztrálása és adminisztrátori jóváhagyás 23](#_Toc197559212)

[3. Teszteset: Keresési funkció tesztelése 24](#_Toc197559213)

[4. Teszteset: Reszponzív megjelenítés különböző eszközökön 25](#_Toc197559214)

[5. Teszteset: Adatbiztonsági funkciók tesztelése 26](#_Toc197559215)

[Továbbfejlesztési lehetőségek 27](#_Toc197559216)

[Irodalomjegyzék, forrásmegjelölés 29](#_Toc197559217)