

Gyulai SZC Szigeti Endre Technikum és Szakképző Iskola

Az ágazat megnevezése: Informatika és távközlés

A szakma megnevezése: Szoftverfejlesztő és -tesztelő

A szakma azonosító száma: 5 0613 12 03

VIZSGAREMEK

Erdőkilátó Állatkert és Szabadidő Park

Nagy Tibor
2/14

Tóth Tamás Levente
2/14

Kolozsvári László Krisztián
2/14

5520 Szeghalom, Ady Endre utca 3.
2023.

Tartalomjegyzék

Bevezetés.....	3
Fejlesztői dokumentáció.....	4
Tesztdokumentáció	19
Felhasználói dokumentáció.....	24
Összefoglalás	27

Bevezetés

A fejlesztők teljes célja, hogy a látogatók számára teljeskörű információt nyújtson a képzeletünkben lévő állatkert és szabadidőparkról, hogy kedvet kapjanak arra, ha egyszer megvalósul a tervünk, akkor minél többen látogassanak el korhatártól függetlenül.



A weboldal célja, hogy a mi képzeletünkben lévő állatkertet és szabadidőparkot mutassa virtuálisan, illetve odafigyel a közösség fejlesztésére, a természetvédelemre és a sportra is egyaránt. Az **állatkert** a fő attrakciója, amely a park területének az 1/4-ét tölti ki.

Az **állatkertben** számos kontinensekről származó állatfaj található itt, amelyeket naponta több száz látogató csodál meg a weboldalunkon. Az állatok földrészenként csoportosított kifutókban tekinthetők meg.

A **tópart** a kikapcsolódásra vágyó látogatóknak a park strandja a legideálisabb hely. A partszakaszon egy ételbár üzemel reggeltől késő délutánig. A tó fürdésre, a tópart sátorozásra, kempingezésre és táborútrakásra is ideális, természetesen a szigorú szabályok betartása mellett.

Az **erdő** a parkerdőben kitaposott ösvényeken haladva számos facsoportokat tekinthetnek meg a látogatók. Az erdő jelentős része is alkalmas sátorozásra. A kerékpárral érkező látogatóknak lehetőségük van kipróbálni a krosszpályát, melynek domborzata földkupacokból kialakított ugratókból állnak. A rámpák mérete a 1 métert is meghaladják. Az erdőben szintén tudnak a résztvevők kempingelni, sátorozni, de van lehetőség lakókocsis kirándulásra is.

Az **étterem** a park étterme a tölgyerdő mélyén az állatkert és a tópart látványától elszigetelve helyezkedik el, azonban az épület teraszairól a körülötte lévő erdő elképesztő látványt biztosít a vendégeinek.

A **szabadidőpark**, mely az állatkert mellett található közvetlen, számos programmal várja az érdeklődőket. A vendégek különböző programokat tudnak igénybe venni. A programok korosztálytól függetlenül, a kicsiktől az egészen nagy korosztályig megtalálják a magukhoz illő programokat. Legyen szó gyalogtúráról, biciklizésről, jógáról, de akár a tó körül lévő röplabdáról vagy a vízibiciklizésről.

Fejlesztői dokumentáció

A program elkészítésére azért volt szükség, hogy a felhasználók információt gyűjtsenek az oldalak tartalmából. A hasonló weboldalakhoz képest a mi oldalunkon megjelennek az állatok is részletesebben és a programokról is kiterjesztett leírást nyújt az oldal. Idő hiányában nem volt lehetőség arra, hogy online is tudjon a felhasználó bejelentkezni programokra. A későbbiekben még erre sor kerülhet.

A program elkészítéséhez Windows 10 Pro szoftververziójú munkaeszközöket használtunk.



Processzor: Intel(R) Core(TM) i5-10400 CPU @ 2.90GHz 2.90 GHz

Memória (RAM) mérete: 8,00 GB

Rendszer típusa: 64 bites operációs rendszer, x64-alapú processzor

Ha a felhasználó eszköze hasonló paraméterekkel rendelkezik, akkor elérhető számára az weboldal.

A fejlesztés során használt szoftverek:



A frontend oldal elkészítéséhez a React keretrendszert használtuk.

Rövid leírás a React keretrendszerről:

- React egy nyílt forrású JavaScript könyvtár, amely lehetővé teszi a felhasználói felületek (UI) építését. A React az egyik legnépszerűbb frontend keretrendszer, amelyet a legtöbb fejlesztő használ a modern webalkalmazások fejlesztéséhez.
- A React egy deklaratív megközelítést alkalmaz a felhasználói felületek építéséhez, amely azt jelenti, hogy a fejlesztők csak azt határozzák meg, hogy a felhasználói felületnek milyen állapotokban kell lennie, és a React

automatikusan kezeli a felületi elemek állapotának frissítését az adott változásoknak megfelelően.

- A React az egyes alkalmazási funkciókhoz komponenseket használ, amelyek kis, önálló egységek, amelyeket könnyen lehet újrahasznosítani. Az egyes komponensek állapotát és tulajdonságait egy egyszerű API segítségével lehet kezelni.
- A React előnyei közé tartozik a gyors és hatékony alkalmazásfejlesztés, a könnyű újrahasznosítás és a könnyen érthető kód. A React használatával fejlesztők nagyobb rugalmasságot élveznek a felhasználói felület megtervezésében és fejlesztésében, valamint segíti a fejlesztőket a kód hatékonyabb kezelésében és karbantartásában.

A backend oldal elkészítéséhez a Node.js Sequelize keretrendszert használtuk.

Rövid leírás a Sequelize keretrendszerről:

- Sequelize egy Node.js alapú ORM (Object-Relational Mapping) keretrendszer, amely lehetővé teszi a Node.js alkalmazások számára, hogy könnyedén kommunikáljanak az adatbázisokkal. A Sequelize főként a relációs adatbázisokhoz tervezett, mint például a MySQL, Postgres, SQLite és MSSQL.
- A Sequelize segítségével a Node.js alkalmazások képesek lehetnek az adatbázisokkal történő műveletek végrehajtására, mint például az adatok lekérdezése, beillesztése, frissítése és törlése. A Sequelize lehetővé teszi az alkalmazások számára, hogy a modelleket használják az adatbázis táblák ábrázolására, amelyek tartalmazzák a mezőket és azok típusait, amelyeket az adatbázisban tárolnak.
- A Sequelize lehetővé teszi az adatbázisok szinkronizálását is az alkalmazás modelleivel, így az adatbázis táblák automatikusan létrejönnek, amikor az alkalmazás elindul, és frissítve lesznek, amikor a modellek módosulnak. Emellett a Sequelize támogatja a tranzakciókat és a lekérdezési eredmények lapozását, amelyek lehetővé teszik az alkalmazások számára, hogy hatékonyabban dolgozzanak az adatbázisokkal.
- A Sequelize további előnye, hogy nagy rugalmasságot és konfigurálhatóságot kínál. A felhasználók számos beállítási lehetőséget használhatnak az alkalmazásuk igényei szerint, és testreszabhatják a Sequelize-t az adatbázisok számos különböző típusához. Emellett a Sequelize támogatja a middleware-ek használatát, amelyek lehetővé teszik az alkalmazások számára, hogy kiterjesszék a Sequelize működését és az adatbázisműveletek előtt és után történő módosításokat végezzenek.

Az adatbázis-kezeléshez a dbForge Studio adatbázis-kezelő szoftvert használtuk.

Rövid leírás a dbForge Studio alkalmazásról:

- A dbForge Studio egy adatbázis-kezelő eszköz, amelyet a Devart fejlesztett ki. A szoftver támogatja az összes nagyobb adatbázis-szervert, beleértve az Oracle-t, a SQL Server-t, a MySQL-t, a PostgreSQL-t, a SQLite-ot, az InterBase-t és a Firebird-ot. Az eszköz lehetővé teszi az adatbázisok tervezését, szerkesztését, karbantartását és a lekérdezések futtatását.
- A dbForge Studio kiterjedt funkciókészlete számos eszközt és lehetőséget kínál, amelyek megkönnyítik az adatbázis-adminisztrátorok és fejlesztők munkáját. Például lehetőség van az adatbázis-séma létrehozására, a táblák és mezők létrehozására, a relációk definiálására, az adatok importálására és exportálására, a lekérdezések írására és futtatására, az adatok vizualizációjára, a szkriptek végrehajtására és sok más feladatra.
- A dbForge Studio könnyen használható felhasználói felülete és intuitív kezelőfelülete lehetővé teszi a felhasználók számára, hogy gyorsan és hatékonyan dolgozzanak az adatbázisaikkal. A szoftver számos hasznos funkciót tartalmaz, például kódszerkesztőt, szerkesztési tippeket, automatikus kódkiegészítést, kódgenerálást és sok másat.
- A dbForge Studio teljes körű adatbázis-kezelő eszközként szolgál, amely segíti az adatbázisok hatékony és biztonságos kezelését, így nagyszerű választás azoknak az adatbázis-adminisztrátoroknak és fejlesztőknek, akik hatékony és megbízható adatbázis-kezelő eszközt keresnek.

Hogy miért ezeket a keretrendszereket választottuk?

A React keretrendszer egy nyílt forráskódú JavaScript keretrendszer, amelyet a Facebook fejlesztett ki és tart karban. A React-et az alábbi okok miatt használják a fejlesztők:

1. Hatékonyabb felhasználói felületek készítése: A React keretrendszer lehetővé teszi a fejlesztők számára, hogy hatékonyabban és gyorsabban hozzanak létre nagyobb és összetettebb felhasználói felületeket.
2. Komponens alapú felépítés: A React segítségével a fejlesztők könnyen felépíthetnek és újra felhasználhatnak
3. kisebb részekből álló komponenseket, amelyeket könnyedén össze lehet fűzni nagyobb alkalmazásokká.

4. Virtuális DOM: A React alkalmazza a virtuális DOM-ot, amely lehetővé teszi, hogy hatékonyan kezelje a nagyszámú adatelemet, és a változások automatikus frissítése miatt javítja az alkalmazás teljesítményét.
5. Rugalmas és bővíthető: A React rugalmas és bővíthető keretrendszer, amelynek segítségével a fejlesztők számos bővítményt és kiegészítőt használhatnak, amelyek segítségével még hatékonyabbá és gyorsabbá tehetik az alkalmazásokat.
6. Nagy közösség: A React nagy és aktív közösséggel rendelkezik, amely rengeteg hasznos dokumentációval és segítséggel szolgál a fejlesztőknek. Ezen felül, rengeteg fejlesztő és vállalat használja a React-et, így a keretrendszer folyamatosan fejlődik és javul.

Összességében, a React keretrendszer hatékony és rugalmas megoldást nyújt a felhasználói felületek készítésére, és segítségével a fejlesztők gyorsan és hatékonyan tudnak dolgozni a nagyobb és összetettebb alkalmazásokkal.

A Sequelize egy Node.js alapú ORM (Object-Relational Mapping) keretrendszer, amely lehetővé teszi a Node.js alkalmazások számára, hogy adatbázisokkal kommunikáljanak és adatokat tároljanak. Az ORM megkönnyíti az adatbázis kezelést és csökkenti az ismétlődő kódmennyiséget.

Az Sequelize használata több előnnyel jár a Node.js backend fejlesztők számára:

1. Támogatja több adatbázis típusát: Az Sequelize támogatja a MySQL, PostgreSQL, SQLite és MSSQL adatbázisokat, így az alkalmazások sokféle adatbázis típussal kommunikálhatnak.
2. Egyszerű adatbázis-kezelés: Az Sequelize egyszerű és hatékony módja az adatbázis-kezelésnek. Az ORM segítségével a fejlesztők könnyedén hozzáadhatnak, frissíthetnek, törölhetnek vagy lekérdezhetnek adatokat az adatbázisból.
3. Biztonságos adatbázis-kapcsolat: Az Sequelize lehetővé teszi az alkalmazások számára, hogy biztonságosan kommunikáljanak az adatbázissal SSL vagy TLS titkosítással.
4. Egységes adatmodell: Az ORM lehetővé teszi a fejlesztők számára, hogy egységes adatmodellt használjanak az alkalmazás egészében, ami megkönnyíti a fejlesztést és a karbantartást.

5. Aszinkron működés: Az Sequelize lehetővé teszi az aszinkron működést, ami jelentősen javítja az alkalmazás teljesítményét és skálázhatóságát.
6. Támogatja a migrációkat: Az ORM támogatja a migrációkat, ami lehetővé teszi az adatbázis sémájának változtatását a fejlesztési folyamat során, anélkül hogy az adatokat elveszítenénk.

Összességében a Sequelize egy hatékony és kényelmes módja az adatbázis-kezelésnek a Node.js alkalmazásokban. Az ORM lehetővé teszi a fejlesztők számára, hogy egyszerűen és hatékonyan kezeljék az adatokat, miközben javítják az alkalmazás teljesítményét és skálázhatóságát.

A dbForge Studio egy olyan adatbázis-kezelő és adatbázis-fejlesztő eszköz, amely segít az adatbázisok tervezésében, karbantartásában és optimalizálásában. Az alkalmazás számos funkcióval rendelkezik, amelyek megkönnyítik az adatbázis-fejlesztők és az adatbázis-adminisztrátorok munkáját.

Néhány olyan funkció, amelyekért érdemes lehet használni a dbForge Studio-t:

1. Adatbázis-tervező: Az alkalmazás segítségével könnyen tervezhetünk adatbázisokat, melyek tartalmazzák az adatok tábláit, mezőit, kulcsait és kapcsolatait.
2. Adatbázis-szerkesztő: A dbForge Studio lehetővé teszi az adatok szerkesztését, frissítését és törlését az adatbázisban. Az adatok könnyen navigálhatóak és szerkeszthetőek, és a változtatások automatikusan mentésre kerülnek.
3. Adatbázis-optimalizálás: Az alkalmazás segítségével könnyen optimalizálhatjuk az adatbázis teljesítményét. Az alkalmazás számos funkcióval rendelkezik, amelyek segítenek az adatbázis-indexek, táblák és lekérdezések optimalizálásában.
4. Adatbázis-exportálás és importálás: A dbForge Studio lehetővé teszi az adatok importálását és exportálását az adatbázisok között. Az adatok könnyen átjárhatóak különböző adatbázisok között, így az adatok áttöltése könnyebb és gyorsabb lesz.
5. Adatbázis-monitorozás: Az alkalmazás lehetővé teszi az adatbázis-monitorozást, így figyelemmel kísérhetjük az adatbázisok teljesítményét és hatékonyságát. Az adatbázis-monitorozás segít az esetleges problémák időben történő felismerésében és megoldásában.

Összességében, a dbForge Studio egy hatékony eszköz az adatbázis-fejlesztők és adatbázis-adminisztrátorok számára, amely megkönnyíti az adatbázis-tervezést, -karbantartást és -optimalizálást, valamint lehetővé teszi az adatok hatékonyabb átültetését és kezelését.

Az adatbázis felépítését az yEd Graph Editor alkalmazással terveztük. yEd Graph Editor egy ingyenes, platformfüggetlen diagram szerkesztő szoftver, amely lehetővé teszi a felhasználók számára, hogy professzionális minőségű diagramokat hozzanak létre. A programot az yWorks GmbH fejleszti.

A yEd Graph Editor számos előre elkészített sablont és eszköztárat tartalmaz, amelyek segítségével a felhasználók könnyen létrehozhatnak olyan diagramokat, mint például áramköri rajzok, hálózati diagramok, UML diagramok, folyamatábrák és még sok más.

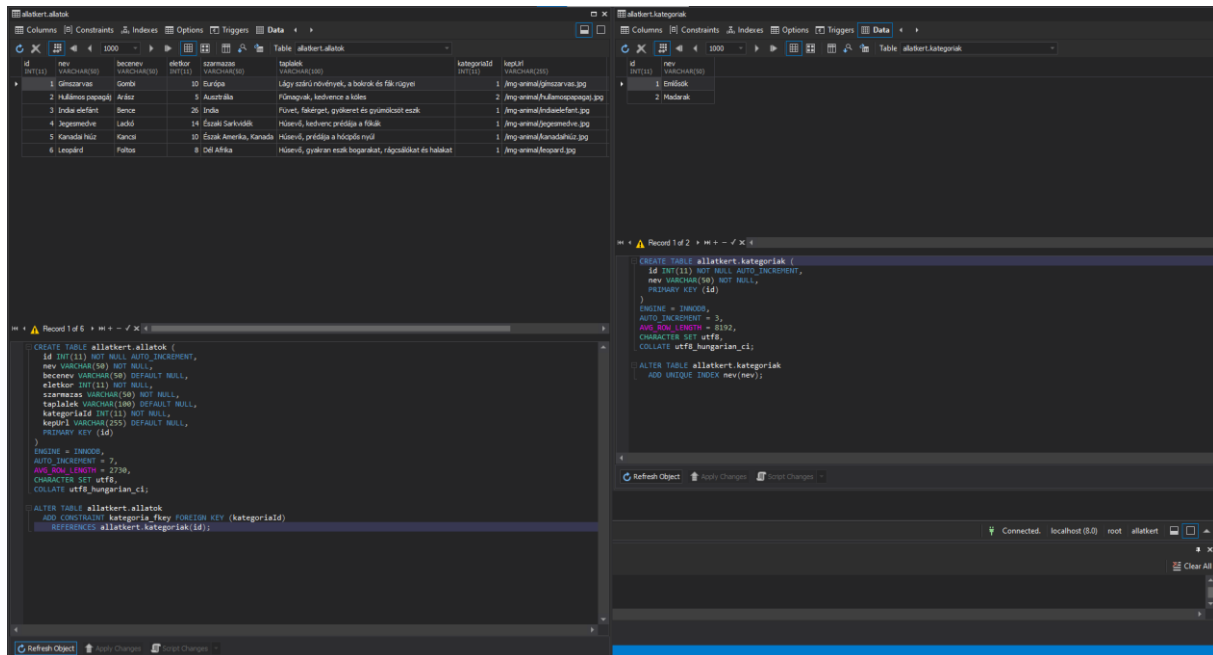
A program lehetővé teszi az adatok importálását és exportálását több formátumban, beleértve az XML, SVG, PDF, JPG és PNG formátumokat. A felhasználók egyedi beállításokat hajthatnak végre, például a színek, betűméretek és a vonalstílusok testreszabását.

Az yEd Graph Editor interaktív funkciókat is kínál, például a diagramok animált megjelenítését, amely lehetővé teszi a felhasználók számára, hogy áttekintést nyerjenek a diagramok összetettségéről és azok kapcsolatairól.

A program magas szintű használhatóságot biztosít, mivel egyszerűen navigálható, és intuitív felhasználói felülettel rendelkezik. Az yEd Graph Editor letöltése és használata ingyenes, és elérhető Windows, Mac OS X és Linux operációs rendszerekhez.

Részletesebben az adatbázisról:

- állatok tábla – tartalmazza az állatok legjellemzőbb adatait
- kategóriák tábla – tartalmazza az állatoknak a rendszertani adatait



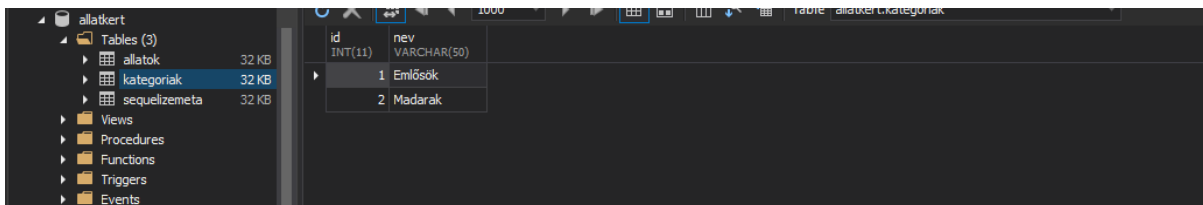
Az állatok adattábla szerkezete, felépítése:

- id: az állat egyedi azonosítója
- nev: az állat neve
- becenev: az állat beceneve
- eletkor: az állat életkora
- szarmazas: az állat származása
- taplalek: az állat tápláléka
- kategoriald: azonosító kapcsolat a kategoriak táblához
- kepUrl: fénykép URL az állathoz

id	nev	becenev	eletkor	szarmazas	taplalek	kategoriald	kepUrl
1	Gímszarvas	Gombi	10	Európa	Lágy szárú növények, a bokrok és fák rügyei	1	/img-animal/gimszarvas.jpg
2	Hüllámos papagáj	Arász	5	Ausztrália	Fűmagvak, kedvence a koles	2	/img-animal/hullamospapagaj.jpg
3	Indiai elefánt	Bence	26	India	Füvet, fakérget, gyökeret és gyümölcsöt eszik	1	/img-animal/indiaielefant.jpg
4	Jegesmedve	Ladkó	14	Északi Sarkvidék	Hűsévő, kedvenc prédája a fókák	1	/img-animal/jegesmedve.jpg
5	Kanadai hiúz	Kancsi	10	Észak Amerika, Kanada	Hűsévő, prédája a hócipő nyúl	1	/img-animal/kanadahiuz.jpg
6	Leopárd	Foltos	8	Dél Afrika	Hűsévő, gyakran eszik bogarakat, rágcsálókat és halakat	1	/img-animal/leopard.jpg

A kategóriák adattábla szerkezete, felépítése:

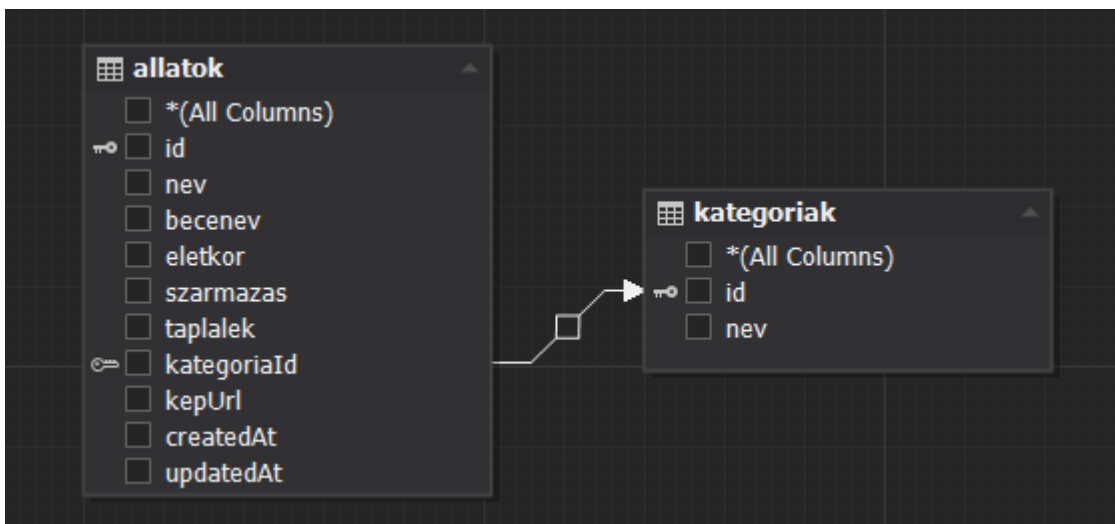
- id: a kategória egyedi azonosítója
- nev: a kategória neve



id	nev
1	Emlősök
2	Madarak

Az adattáblák közötti kapcsolat:

- az allatok tábla kategoriald azonosítója kapcsolódik a kategoriak tábla id egyedi azonosítójához
- ez a kapcsolat adja meg, hogy az állat milyen törzsbe tartozik

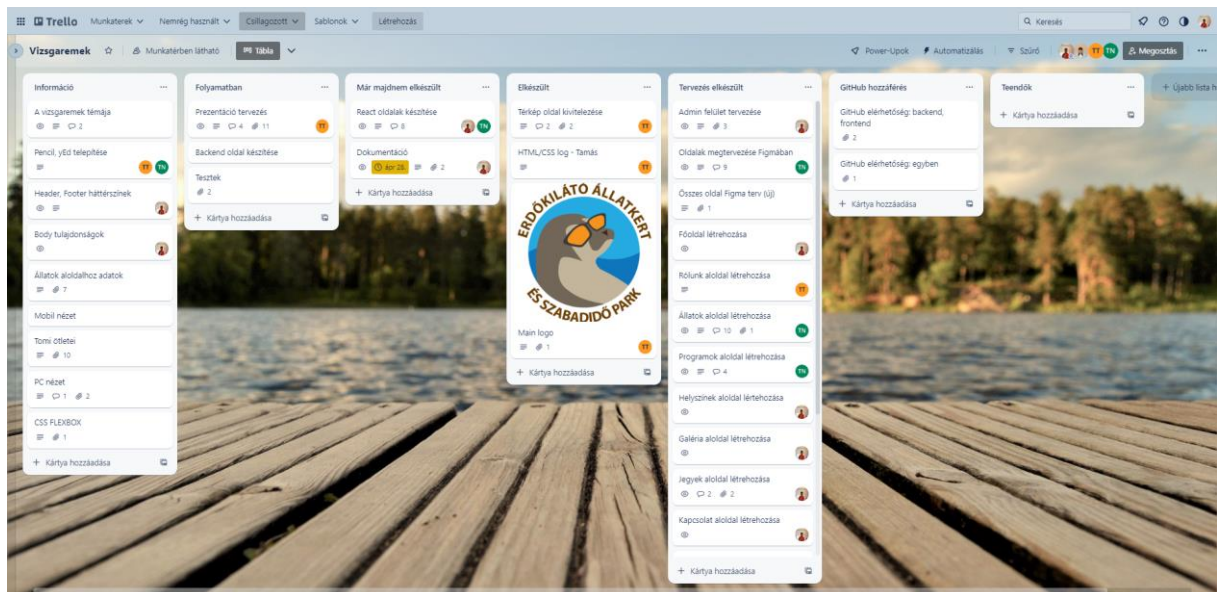


A fejlesztési folyamatokat a következő projektmenedzsmenttel követtük:

- A Trello egy webalapú projektmenedzsment és együttműködési eszköz, amely vizuális megközelítést használ a csapatoknak a feladatok megszervezéséhez és rangsorolásához. A projekteket jelző „táblák” és „kártyák” rendszerét használja, amelyek a projekten belüli egyes feladatokat vagy elemeket képviselik. A felhasználók létrehozhatnak, áthelyezhetnek és rangsorolhatnak kártyákat a táblán, valamint hozzárendelhetik azokat a csapattagokhoz, határidőket adhatnak hozzá, és fájlokat csatolhatnak. A Trello olyan funkciókat is tartalmaz, mint az ellenőrző listák, címkék és megjegyzések, amelyek segítségével további kontextusokat és információkat adhatnak a feladatokhoz. A Trello-t széles körben használják a különböző iparágakban működő csapatok, beleértve a szoftverfejlesztést, a marketinget és az oktatást.

A saját projektmenedzsmentünk elérhetősége:

 <https://trello.com/b/o28HyLqs/vizsgaremek>



Mielőtt a teljes projektnek elkezdtük volna a kódolását, a Figma alkalmazás segítségével elkezdtük megtervezni az oldalakat elsősorban csak egyszerűen, aztán volt olyan oldal, amelyhez készült prototípus tervezés is. A tervezések 3 méretben készültek, hogy minden eszközhöz kompatibilisek legyenek.

Figma egy kollaboratív, felhőalapú design platform, amely lehetővé teszi a tervezők számára, hogy együtt dolgozzanak projektjeiken, és könnyen megoszthassák munkájukat másokkal. A Figma alkalmazás lehetővé teszi a felhasználók számára, hogy létrehozzanak, szerkesszenek és megosszanak vektoros grafikákat, felhasználói felületeket, weboldalakat és mobilalkalmazásokat.

Az alkalmazásban található számos eszköz és funkció, mint például a tervezőeszközök (pl. toll, négyzet, vonal, szöveg stb.), a rétegezési lehetőségek, a betűtípusok és színválasztók, valamint a prototípuskészítő funkció, amely lehetővé teszi a felhasználók számára, hogy dinamikus interaktív prototípusokat hozzanak létre.

A Figma használata nagyon egyszerű és intuitív, és lehetővé teszi a tervezők számára, hogy gyorsan és hatékonyan dolgozzanak együtt projekten, míg a platform biztonságosan tárolja és megőrzi a munkájukat. A Figma számos funkciója és előnye miatt a tervezők széles körben használják, és sokan a tervezésben használt alapvető eszköznek tekintik.

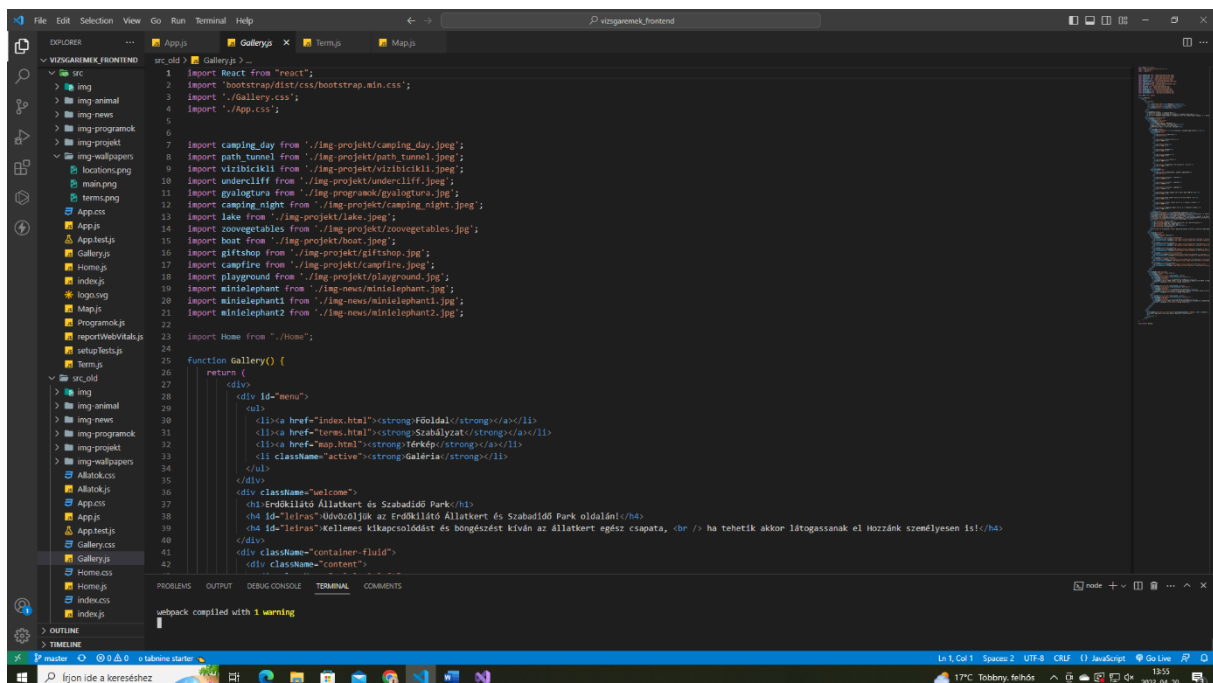
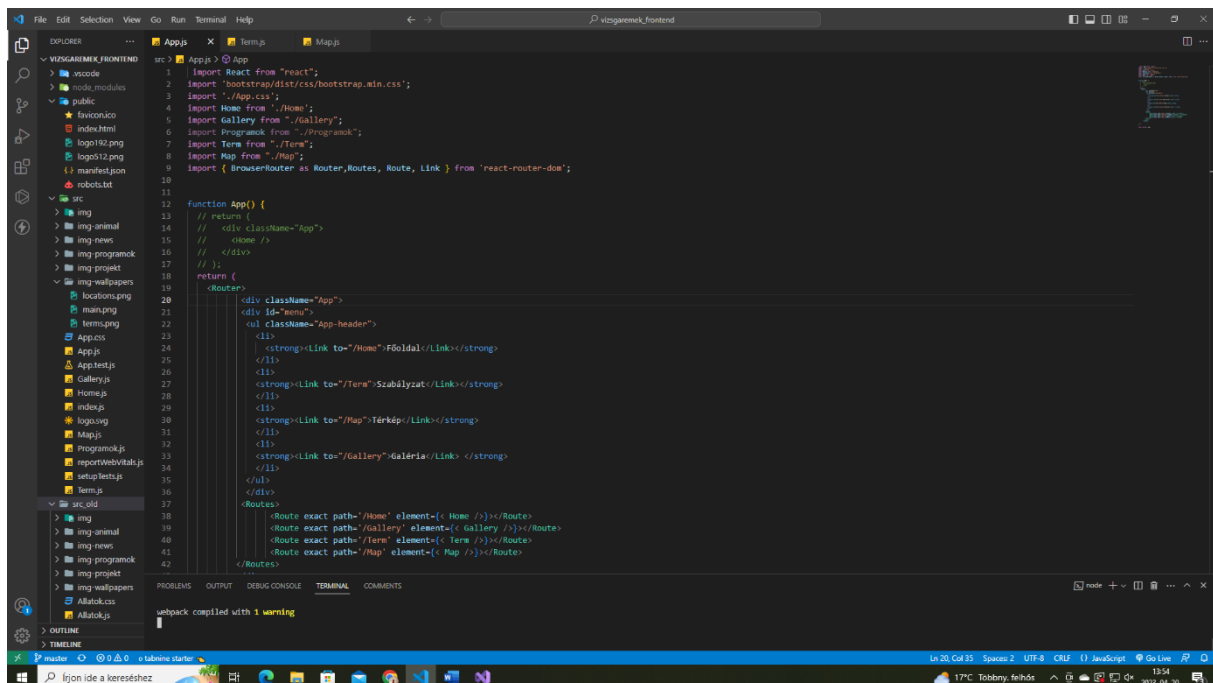
Néhány részlet a tervezés közben:



Az oldalakat első lépésként statikus oldalakként (HTML5, CSS3) készítettük el, hogy rövidebb idő alatt lássuk, hogy hogyan is kell majd átalakítanunk a végleges projekthez.

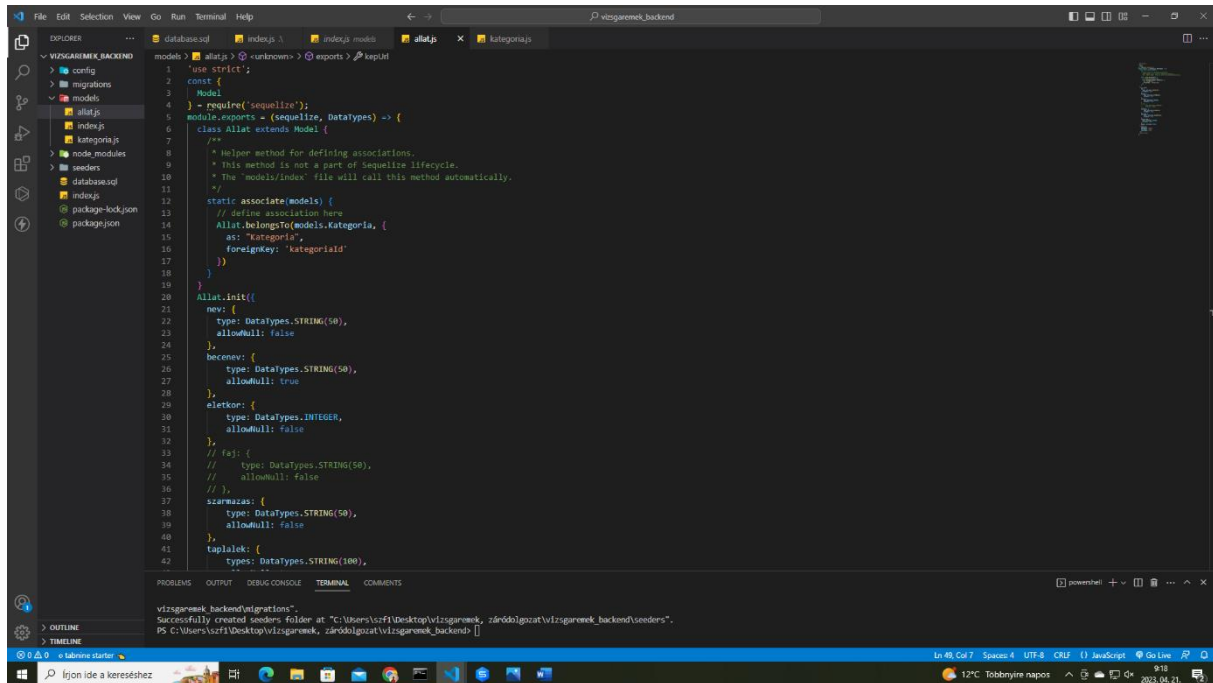


Frontend oldal készítése közben készült kódrészletek. Fontos volt számunkra, hogy minden rész rögzítve legyen az esetleges hibák elkerülése végett. Emiatt többször készítettünk képernyőfelvételek, illetve a https://github.com/kolozsvarikrisztian/allatkert-vizsgaremek_frontend.git közös repository volt az, ahol a folyamatokat dátum szerint tudtuk követni.

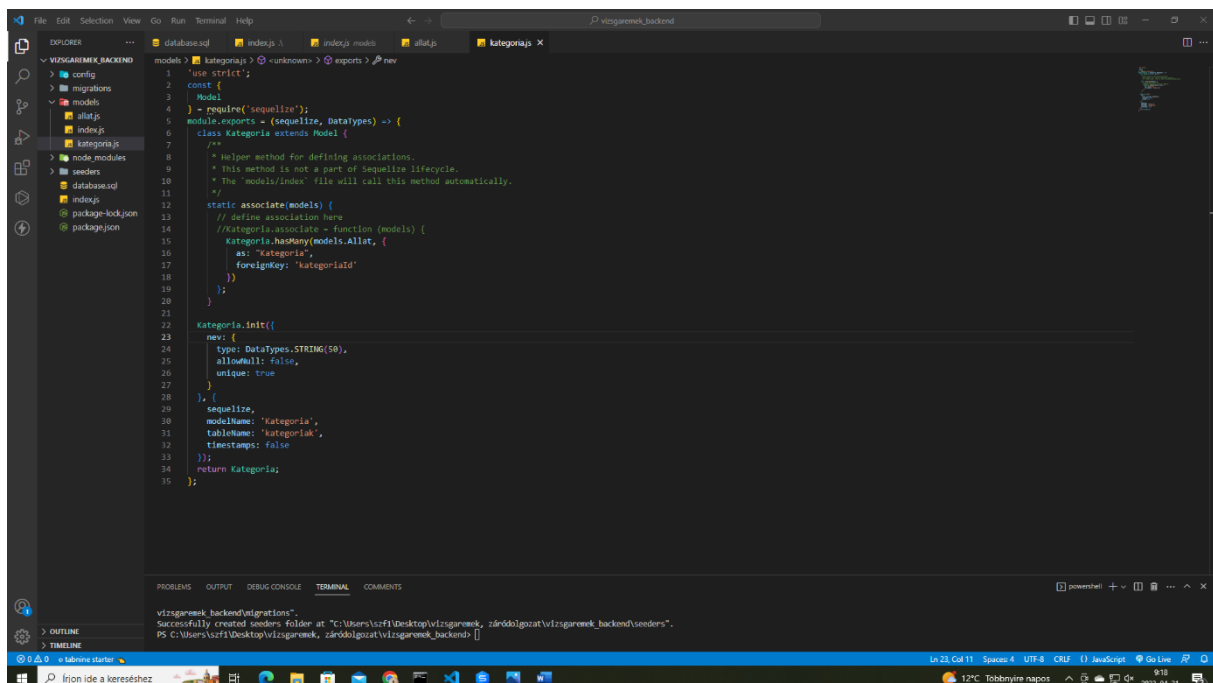


A közös backend oldal fejlesztéséhez a https://github.com/kolozsvarikrisztian/allatkert-vizsgaremek_backend.git közös repository volt nagy segítségünkre.

A backend oldal közben készült kódrészletek:



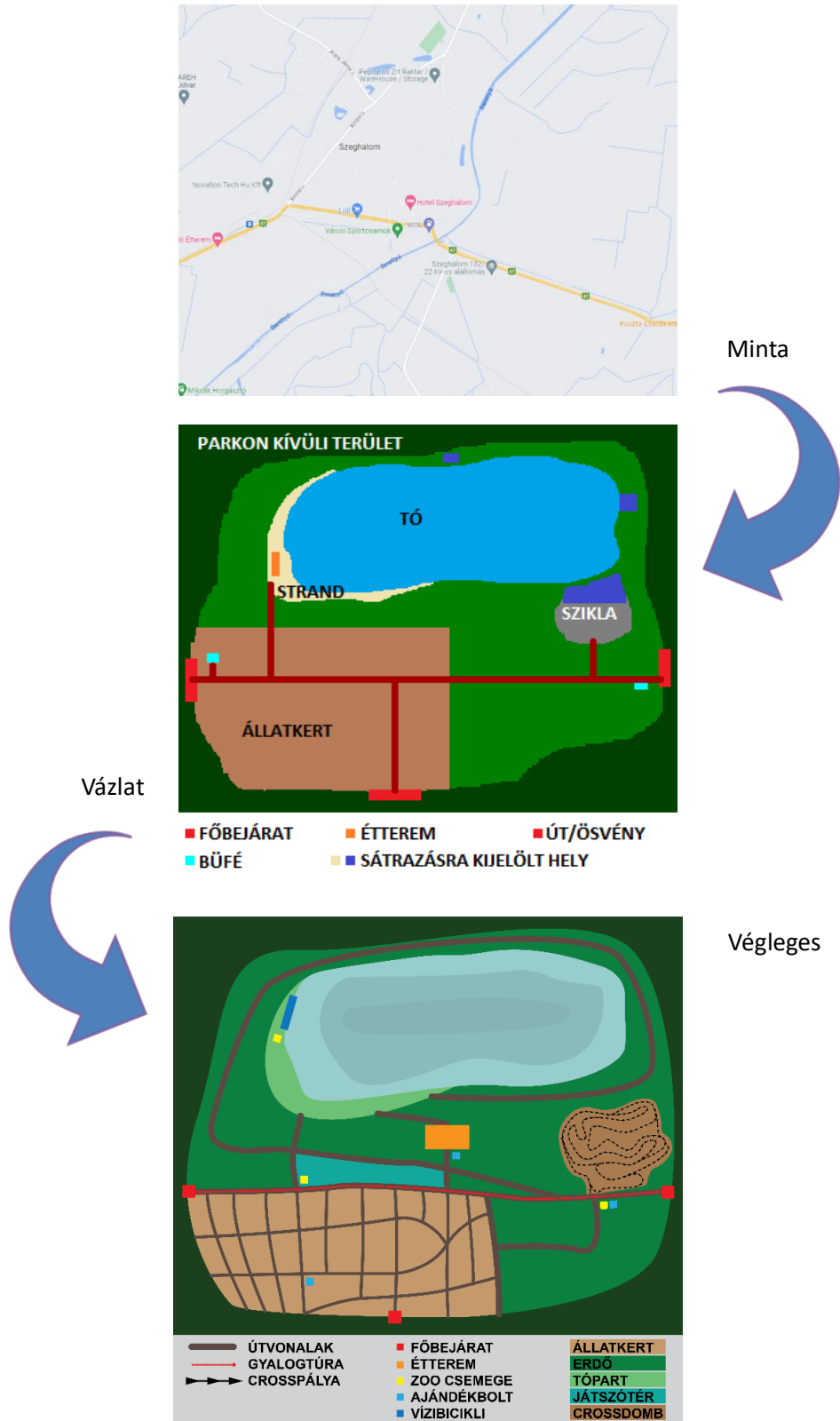
```
1 'use strict';
2 const {
3   Model
4 } = require('sequelize');
5 module.exports = (sequelize, DataTypes) => {
6   class Allat extends Model {
7     /**
8      * Helper method for defining associations.
9      * This method is not a part of Sequelize lifecycle.
10      * The 'models/index' file will call this method automatically.
11      */
12     static associate(models) {
13       // define association here
14       Allat.belongsTo(models.Kategoria, {
15         as: 'Kategoria',
16         foreignkey: 'kategoriaId'
17       })
18     }
19   }
20   Allat.init({
21     nev: {
22       type: DataTypes.STRING(50),
23       allowNull: false
24     },
25     becenev: {
26       type: DataTypes.STRING(50),
27       allowNull: true
28     },
29     eletkor: {
30       type: DataTypes.INTEGER,
31       allowNull: false
32     },
33     faj: {
34       type: DataTypes.STRING(50),
35       allowNull: false
36     },
37     szarmazas: {
38       type: DataTypes.STRING(50),
39       allowNull: false
40     },
41     taplakok: {
42       type: DataTypes.STRING(100),
```



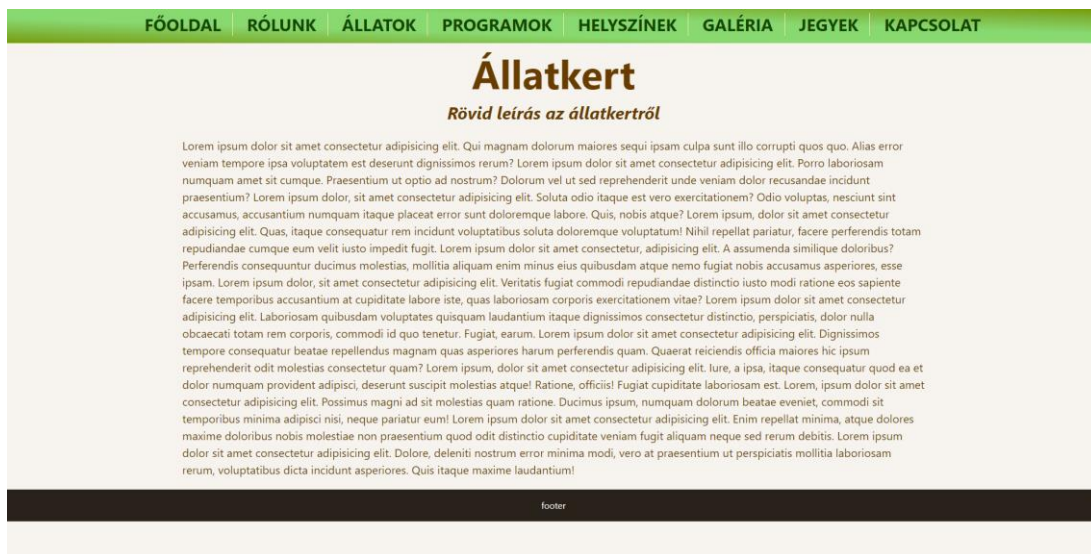
```
1 'use strict';
2 const {
3   Model
4 } = require('sequelize');
5 module.exports = (sequelize, DataTypes) => {
6   class Kategoria extends Model {
7     /**
8      * Helper method for defining associations.
9      * This method is not a part of Sequelize lifecycle.
10      * The 'models/index' file will call this method automatically.
11      */
12     static associate(models) {
13       // define association here
14       Kategoria.hasMany(models.Allat, {
15         as: 'Kategoria',
16         foreignkey: 'kategoriaId'
17       })
18     }
19   }
20   Kategoria.init({
21     nev: {
22       type: DataTypes.STRING(50),
23       allowNull: false,
24       unique: true
25     },
26     becenev: {
27       type: DataTypes.STRING(50),
28       allowNull: true
29     },
30     eletkor: {
31       type: DataTypes.INTEGER,
32       allowNull: false
33     },
34     faj: {
35       type: DataTypes.STRING(50),
36       allowNull: false
37     },
38     szarmazas: {
39       type: DataTypes.STRING(50),
40       allowNull: false
41     },
42     taplakok: {
43       type: DataTypes.STRING(100),
```


A térkép oldal készítése közben, elsőnek csak a Google Térképről kivágott mintaképet használtunk a térkép ábrázolására, aztán az később elkészült a saját elképzelésünk alapján grafikai alkalmazással.

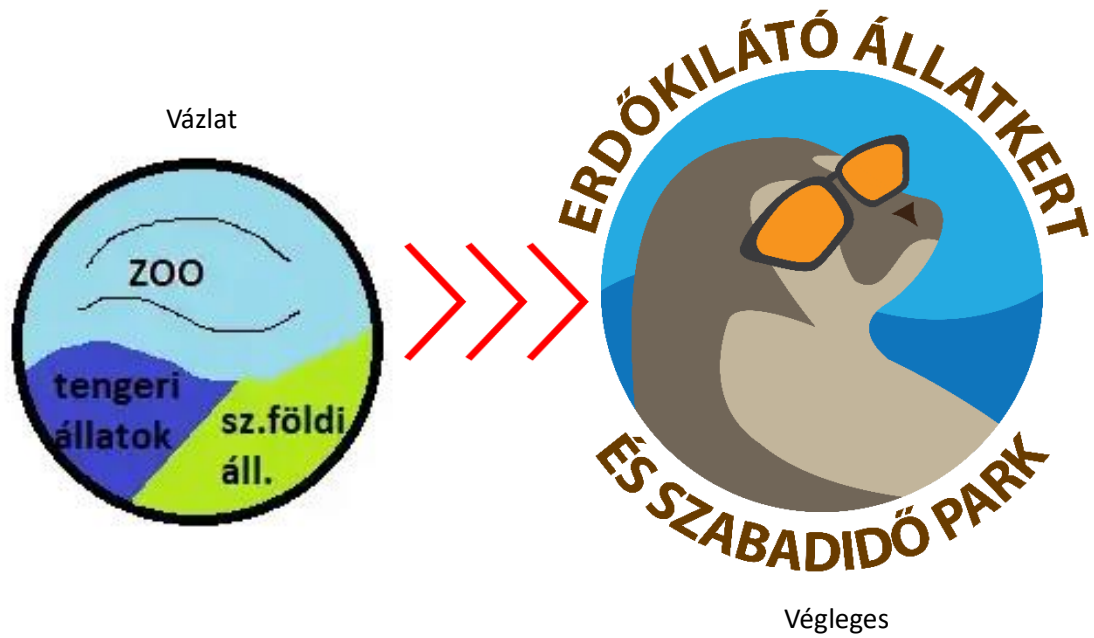
Folyamatábra a térkép készítéséről:



Folyamatábra a verziófrissítésekről:



Logónak a tervezése és az elkészült logó:



Teszt dokumentáció

A React tesztek általában a React alkalmazások vagy komponensek funkcionalitását és megjelenését tesztelik. A tesztek általában automatikusan futnak, és két fő típusra oszthatók: egységtesztek és integrációs tesztek.

Az egységtesztek a React komponensek vagy funkciók egyes részeit tesztelik, például a komponens állapotát vagy a funkció visszatérési értékét. Az egységtesztek segítségével ellenőrizhető, hogy a komponensek és funkciók helyesen működnek-e és helyes adatokat adnak-e vissza.

Az integrációs tesztek az egyes komponensek együttműködését és a rendszer egészének működését tesztelik. Az integrációs tesztek az egyes komponensek általános interakcióját, valamint a külső függőségeket, mint például az API-kat és az adatbázisokat is ellenőrzik.

A React tesztek futtatásához számos tesztelési keretrendszer áll rendelkezésre, például a Jest vagy az Enzyme. Az ilyen tesztelési keretrendszerek lehetővé teszik a tesztek egyszerű és hatékony létrehozását, futtatását és eredményeinek elemzését.

Összességében a React tesztek nagyon fontosak az alkalmazások minőségének biztosításához, és a fejlesztőknek érdemes részletesen megvizsgálniuk a tesztelési keretrendszereket és eszközöket a hatékony és megbízható tesztelés érdekében.

A React alkalmazás tesztjéről készült mentések:

```
FAIL src/test/Home.test.js
  x home page main parts (1830 ms)
```

```
  • home page main parts
```

```
    TestingLibraryElementError: Unable to find an accessible element with the role "img" and name "logo"
```

```
Test Suites: 1 failed, 1 total
Tests:       1 failed, 1 total
Snapshots:   0 total
Time:        5.415 s, estimated 6 s
Ran all test suites.
```

```
PASS src/test/Home.test.js
  ✓ home page main parts (730 ms)
```

```
Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        4.154 s
Ran all test suites.
```

FAIL src/test/Home.test.js
✓ home page main parts (727 ms)
✗ loading animals (1045 ms)

- loading animals

Unable to find role="heading" and name "Gímszarvas"

Test Suites: 1 failed, 1 total
Tests: 1 failed, 1 passed, 2 total
Snapshots: 0 total
Time: 5.681 s
Ran all test suites.

PASS src/test/Home.test.js
✓ home page main parts (735 ms)
✓ loading animals (283 ms)

Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 4.226 s, estimated 5 s
Ran all test suites.

A backend Node.js Sequelizee tesztjéről készült mentések:

FAIL tests/routes.test.js

animals API

✗ should show all animals (43 ms)

• animals API > should show all animals

expect(received).toEqual(expected) // deep equality

Expected: 200

Received: 404

```
5 |         it('should show all animals', async () => {
6 |             const res = await request(app).get('/api/allatok')
> 7 |             expect(res.statusCode).toEqual(200);
    |                                     ^
8 |             expect(res.body).toHaveLength(6);
9 |         })
10 |
```

at Object.toEqual (tests/routes.test.js:7:32)

Test Suites: 1 failed, 1 total

Tests: 1 failed, 1 total

Snapshots: 0 total

Time: 1.75 s, estimated 3 s

Ran all test suites.

PASS tests/routes.test.js

animals API

✓ should show all animals (183 ms)

Test Suites: 1 passed, 1 total

Tests: 1 passed, 1 total

Snapshots: 0 total

Time: 1.669 s, estimated 2 s

Ran all test suites.

FAIL tests/routes.test.js

animals API

✓ should show all animals (146 ms)

✗ should show a animal (10 ms)

• animals API › should show a animal

expect(received).toEqual(expected) // deep equality

Expected: 200

Received: 404

```
11 |         it('should show a animal', async () => {
12 |             const res = await request(app).get('/api/allatok/3')
> 13 |             expect(res.statusCode).toEqual(200);
    |                                     ^
14 |             expect(res.body.nev).toBe('Indiai elefánt');
15 |         })
16 |     })
```

at Object.toEqual (tests/routes.test.js:13:32)

Test Suites: 1 failed, 1 total

Tests: 1 failed, 1 passed, 2 total

Snapshots: 0 total

Time: 1.786 s, estimated 2 s

Ran all test suites.

PASS tests/routes.test.js

animals API

✓ should show all animals (190 ms)

✓ should show a animal (20 ms)

Test Suites: 1 passed, 1 total

Tests: 2 passed, 2 total

Snapshots: 0 total

Time: 1.693 s, estimated 2 s

Ran all test suites.

FAIL tests/routes.test.js

animals API

- ✓ should show all animals (146 ms)
- ✓ should show a animal (25 ms)
- ✗ should show a animal with category (13 ms)

• animals API > should show a animal with category

TypeError: Cannot read properties of undefined (reading 'nev')

```
18 |         const res = await request(app).get('/api/allatok/3')
19 |         expect(res.statusCode).toEqual(200);
> 20 |         expect(res.body.Kategoria.nev).toBe('Emlősök');
    |                                     ^
21 |     })
22 | })
```

at Object.nev (tests/routes.test.js:20:35)

Test Suites: 1 failed, 1 total

Tests: 1 failed, 2 passed, 3 total

Snapshots: 0 total

Time: 1.817 s, estimated 2 s

Ran all test suites.

PASS tests/routes.test.js

animals API

- ✓ should show all animals (189 ms)
- ✓ should show a animal (21 ms)
- ✓ should show a animal with category (14 ms)

Test Suites: 1 passed, 1 total

Tests: 3 passed, 3 total

Snapshots: 0 total

Time: 1.727 s, estimated 2 s

Ran all test suites.

Felhasználói dokumentáció

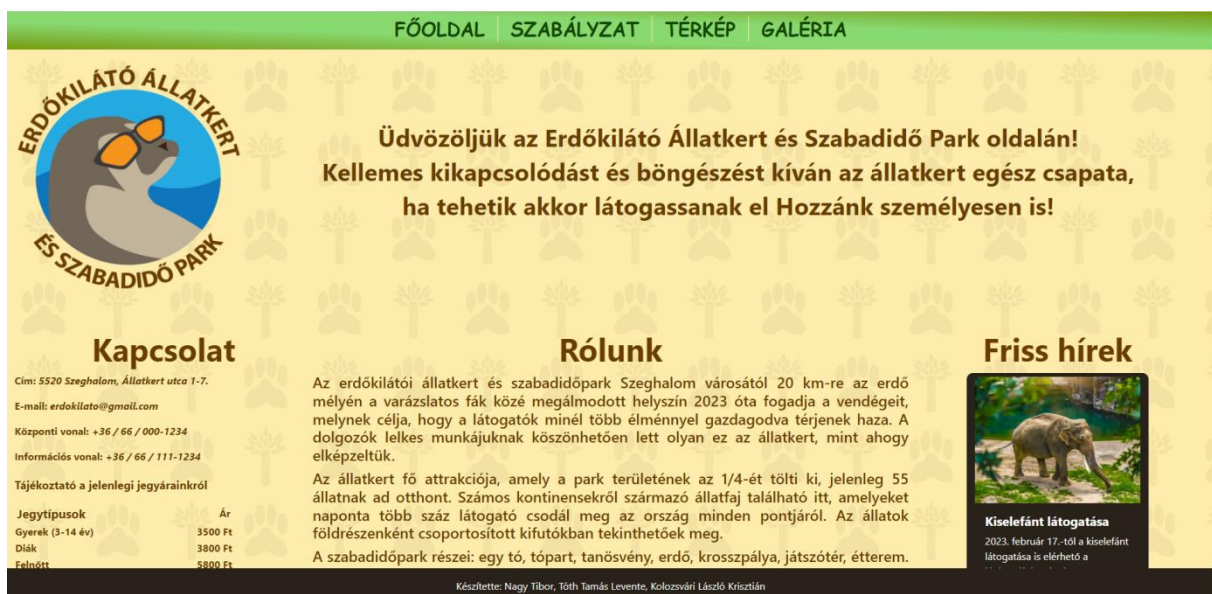
A felhasználó egy állatkert és szabadidő parkról szóló weboldalt tart a kezében, mely részletesen, virtuálisan mutatja be az elképzelt tervünket.

Ahhoz, hogy a felhasználó meg tudja tekinteni a weboldalt szüksége lesz valamilyen eszközre.

Az lehet:

- Windows alapú PC, laptop
- macOS alapú PC, laptop
- Mobileszközökön is elérhető az oldal:
 - IOS alapú iPhone, iPad
 - Android alapú mobiltelefonok, tabletek

Ahogy elindítjuk az alkalmazást a főoldal jelenik meg. A főoldal három részre osztozik, bal oldalt az elérhetőségek, kapcsolat, illetve a jegyekkel kapcsolatos információs adatok jelennek meg. Rövid információ a „Rólunk” cím alatt jelenik meg.



A szabályzat földre lépve megjelennek az állatkertünk és szabadidőparkunk szabályai, amelyet szigorúan be kell tartani, hogy az állatkertet és szabadidőparkot használatba tudják venni.

FŐLDAL
SZABÁLYZAT
TÉRKÉP
GALÉRIA

Kapcsolat

Cím: 5520 Szeghalom, Állatkert utca 1-7.

E-mail: erdokilato@gmail.com

Központi vonal: +36 / 66 / 000-1234

Információs vonal: +36 / 66 / 111-1234

Tájékoztató a jelenlegi jegyárainkról

Jegytípusok	Ár
Gyerek (3-14 év)	3500 Ft
Diák	3800 Ft
Felnőtt	5800 Ft
Nyugdíjas	3800 Ft
70 év felett	2000 Ft
Fogyatékkal élők részére (1 fő + kísérő)	2000 Ft

Kedvezményes, csoportos jegytípusok	Ár
2 felnőtt + 1 gyermek	12500 Ft
2 felnőtt + 2 gyermek	15500 Ft
2 felnőtt + 3 gyermek	17500 Ft
Óvodás csoport (15 fő után 2 kísérő 300 Ft / fő):	1000 Ft
Diák csoport (15 fő után 2 kísérő 300 Ft / fő):	2500 Ft
Felnőtt csoport (15 fő, az ár	4800 Ft

Az Erdőkilátó Állatkert és Szabadidő Park szabályai

Az állatkertre vonatkozó szabályok:

- Az állatkertben lévő állatokat kizárólag csak a helyszínen megvásárolt ZOO csemegével szabad etetni, az otthonról hozott ételekkel, maradékokkal szigorúan TILOS!
- Az állatok simogatására csak a kijelölt helyszíneken van lehetőség. A kiskorú gyermekek csak szülői felügyelettel simogathatják meg az állatainkat!
- 12 év alatt kizárólag csak szülői felügyelettel látogatható az állatkert!
- A park teljes területén a szemételezés, rongálás, a berendezések szabálytalan használata TILOS!
- Dohányozni, gyújtóeszközt használni csak, az arra kijelölt, táblával jelzett helyen szabad, egyébként SZIGORÚAN TILOS!

A szabadidőparkra vonatkozó szabályok:

- A szabadidőparkba való belépéssel, a jegyek megvásárlásával, a szolgáltatások igénybevételével tudomásul veszed a házirendet, és magadra nézve kötelezően elfogadod az ismertetett biztonsági szabályokat is. E szabályokat minden esetben be kell tartanod és másokkal is be kell tartatnod.
- Szolgáltatásainkat mindenki a saját felelősségére veheti igénybe, saját és mások testi épségére vigyázva! Kiskorú vendégeink testi épségéért, valamint az állatok esetlegesen okozott balesetektől/károktól a szülő/gondviselő visel felelősséget, még abban az esetben is, ha nincs jelen.
- Kérjük a szülőket és kiskorúkat, hogy a kiskorúak felügyeletéről gondoskodjanak, mert a bekövetkezett balesetektől és sérülésektől felelősséget nem vállalunk!
- A park teljes területén a szemételezés, rongálás, a berendezések szabálytalan használata TILOS!
- Dohányozni, gyújtóeszközt használni csak, az arra kijelölt, táblával jelzett helyen szabad, egyébként SZIGORÚAN TILOS!
- A szabadidőpark használata csak saját felelősségre használható!

Az erdőben kirándulók csak a kijelölt tűzrakóhelyeken gyűjthetnek tüzet az alábbi szabályok betartása mellett:

- a tűz meggyújtása előtt a tűzrakóhely környékét meg kell tisztítani a levelektől, fűszálaktól annak érdekében, hogy a tűz ne terjedhessen át az erdőre;
- a tüzet nem szabad felügyelet nélkül hagyni, de a még ki nem hűlt paraszat, hamu sem;
- amint feltámad a szél, a tüzet el kell oltani; gondoskodni kell arról, hogy készenlétben legyen a tűz oltására alkalmas anyag, eszköz (pl. víz, homok, lapát);
- az égetés befejezése után a tüzet gondosan el kell oltani, meg kell győződni arról, hogy tényleg elaludt, a hamura távozás előtt lehetőleg földet kell szórni.

Friss hírek

Kiselefánt látogatása

2023. február 17.-től a kiselefánt látogatása is elérhető a látogatóink számára!

[Bővebb információ](#)

A metszet mentve a vágólápra.

Válassza ki a kép megtekintéséhez és megosztásához.

Készítette: Nagy Tibor, Tóth Tamás Levente, Kolozsvári László Krisztián

A térkép földre kattintva egy elképzelt térkép jelenik meg, amelyen fel vannak tüntetve az állatok helyei, illetve a park rész is. A térkép után, ha legörgetünk, akkor láthatjuk a programjaink és a helyszínek részt, a helyszínekről részletes leírást találunk, a programoknak pedig megjelenik az időpont, a helyszíne, az is, hogy ki foglalkozik azon foglalkozáson a résztvevőkkel.

FŐLDAL
SZABÁLYZAT
TÉRKÉP
GALÉRIA

Kapcsolat

Cím: 5520 Szeghalom, Állatkert utca 1-7.

E-mail: erdokilato@gmail.com

Központi vonal: +36 / 66 / 000-1234

Információs vonal: +36 / 66 / 111-1234

Tájékoztató a jelenlegi jegyárainkról

Jegytípusok	Ár
Gyerek (3-14 év)	3500 Ft
Diák	3800 Ft
Felnőtt	5800 Ft
Nyugdíjas	3800 Ft
70 év felett	2000 Ft
Fogyatékkal élők részére (1 fő + kísérő)	2000 Ft

Kedvezményes, csoportos jegytípusok	Ár
2 felnőtt + 1 gyermek	12500 Ft
2 felnőtt + 2 gyermek	15500 Ft
2 felnőtt + 3 gyermek	17500 Ft
Óvodás csoport (15 fő után 2 kísérő 300 Ft / fő):	1000 Ft
Diák csoport (15 fő után 2 kísérő 300 Ft / fő):	2500 Ft

Térkép

Friss hírek

Kiselefánt látogatása

2023. február 17.-től a kiselefánt látogatása is elérhető a látogatóink számára!

[Bővebb információ](#)

Elnevezték az elefántot

A látogatók által beküldött nevek közül első helyen végzett a *Simi* név a kiselefánt számára!

Készítette: Nagy Tibor, Tóth Tamás Levente, Kolozsvári László Krisztián

A galéria földre kattintva jelennek meg az általunk készített fényképek az állatkertről és a szabadidőparkról. A fényképek folyamatosan frissülnek, ahogy bővül az állatkert állománya, illetve a szabadidőparkban történt események során készült képek is ezen az oldalon jelennek meg.

FŐOLDAL

SZABÁLYZAT

TÉRKÉP

GALÉRIA

Kapcsolat

Cím: 5520 Szeghalom, Állatkert utca 1-7.
E-mail: erdokilato@gmail.com
Központi vonal: +36 / 66 / 000-1234
Információs vonal: +36 / 66 / 111-1234
Tájékoztató a jelenlegi jegyárainkról

Jegytípusok	Ár
Gyerek (3-14 év)	3500 Ft
Diák	3800 Ft
Felnőtt	5800 Ft
Nyugdíjas	3800 Ft
70 év felett	2000 Ft
Fogyatékkal élők részére(1 fő + kísérő)	2000 Ft

Kedvezményes, csoportos jegytípusok	Ár
2 felnőtt + 1 gyermek	12500 Ft
2 felnőtt + 2 gyermek	15500 Ft
2 felnőtt + 3 gyermek	17500 Ft
Óvodás csoport (15 fő után 2 kísérő 300 Ft / fő):	1000 Ft
Diák csoport (15 fő után 2 kísérő 300 Ft / fő):	2500 Ft

Galéria

Friss hírek

Kiselefánt látogatása
2023. február 17.-től a kiselefánt látogatása is elérhető a látogatóink számára!
[Bővebb információ](#)

Elnevezték az elefántot
A látogatók által beküldött nevekből első helyen végzett a *Sissi* név! Így a kiselefánt azt a *Sissi* nevet kapta.

Készítette: Nagy Tibor, Tóth Tamás Levente, Kolozsvári László Krisztián

Az oldal teljesen szabadon elérhető, regisztrációra nincsen szükség ahhoz, hogy a felhasználó betekintést nyerjen a weboldalra.

Szinte minden felhasználó számára a legfontosabb rész a kapcsolat, illetve a információ a jegyárakról. Itt tudnak tájékoztatást nyerni a jelenlegi árakról. Ez a sáv minden oldalnak a bal oldalán elérhető, a könnyű navigálás érdekében. A másik fontos rész a fizetési lehetőségek, hogy a látogatók fel legyenek készülve.

Kapcsolat

Cím: 5520 Szeghalom, Állatkert utca 1-7.
Központi vonal: +36 / 66 / 000-1234
Információs vonal: +36 / 66 / 111-1234
Tájékoztató a jelenlegi jegyárainkról

Jegytípusok	Ár
Gyerek (3-14 év)	3500 Ft
Diák	3800 Ft
Felnőtt	5800 Ft
Nyugdíjas	3800 Ft
70 év felett	2000 Ft
Fogyatékkal élők részére(1 fő + kísérő)	2000 Ft

Kedvezményes, csoportos jegytípusok

	Ár
2 felnőtt + 1 gyermek	12500 Ft
2 felnőtt + 2 gyermek	15500 Ft
2 felnőtt + 3 gyermek	17500 Ft
Óvodás csoport (15 fő után 2 kísérő 300 Ft / fő):	1000 Ft
Diák csoport (15 fő után 2 kísérő 300 Ft / fő):	2500 Ft
Felnőtt csoport (15 fő, az ár fejenként értendő):	4800 Ft
Nyugdíjas csoport (15 fő, az ár fejenként értendő):	2800 Ft

Kedvezményes jegytípus
Kérjük vegyék igénybe a fent megemlített kedvezményes jegytípusok valamelyikét, ha Önök jogosultak rá.

Csoportos kedvezmény
Amennyiben Önök valamilyen intézményből (iskola, óvoda, közintézmény) érkeznek, és a csoportjuk létszáma meghaladja a 15 főt, akkor jogosultak a fent megemlített csoportos kedvezményekre.

A csoportoknak előzetes bejelentkezés, illetve időpontegyeztetés szükséges azért, hogy elkerüljük a csoportok ütközését, és ezáltal kialakult kellemetlenségeket.

Fizetési lehetőségek

- Készpénz (HUF)
- Bankkártyás fizetés (Mastercard, Visa, American Express)
- Apple Pay, Simple Pay
- SZÉP kártya (OTP, MKB, K&H)
- Ajándékkártya

A számlaigényét kérjük, hogy előre jelezze. A számlát a pénztárnál tudja majd átvenni.

26

Összefoglalás

A projektmunkánk összefoglalva jól sikerült, idő hiányában még lesz majd rajta mit módosítani, átalakítani, természetesen a jövőben szeretnénk majd foglalkozni a bővítésével.

A csapat szerint jól hasznosulna, ha a valóságban is lenne egy ilyen, az átlátása egyszerű, jól lehet benne tájékozódni. Reméljük, hogy ez az egész egyszer megvalósul, bár egy igen nagy projekt lenne.

Bővítésre, későbbi kiegészítésre, ami már nem fért bele az időbe:

- programjavaslatok a felhasználótól
- módosítás, új állat felvitele „admin” bejelentkezéssel
- jelentkezni online a programokra
- bővíteni a hírek listát, ha a bővebb információ gombra kattintunk, akkor részletesen jelenjen meg teljes képernyőn