

BUDAPESTI MŰSZAKI SZAKKÉPZÉSI CENTRUM  
PETRIK LAJOS KÉT TANÍTÁSI NYELVŰ TECHNIKUM

SZOFTVERFEJLESZTŐ ÉS TESZTELŐ TECHNIKUS SZAKMA

# DogGo

**Készítette:** Kilián Marcell András, Korcsmáros Kristóf György, Takács Balázs Levente  
Budapest, 2021.

# 1 Tartalomjegyzék

DogGo .....	1
1 Tartalomjegyzék .....	2
Plágium nyilatkozat .....	3
2 Bevezetés .....	4
2.1 Téma .....	5
2.2 Témaválasztás indoklása .....	6
3 Fejlesztői dokumentáció .....	7
3.1 Funkciók .....	7
3.2 Használati eset diagram .....	8
3.3 Backend .....	9
3.3.1 Adatbázis terv .....	9
3.3.2 Adattáblák (doggodb) .....	10
3.3.3 Backend telepítésének lépései .....	12
3.3.4 API végpontok .....	13
3.4 DogGo - Web alkalmazás .....	25
3.4.1 Felületterv .....	25
3.4.2 Rendszerkövetelmények .....	27
3.5 Fejlesztői dokumentáció .....	28
3.5.1 Alkalmazott fejlesztői eszközök .....	28
3.5.2 Komponensek .....	29
3.5.3 Komponensek dokumentálása .....	30
3.5.4 Tesztelési dokumentáció .....	39
3.5.5 Továbbfejlesztési lehetőségek .....	44
3.6 Felhasználói dokumentáció .....	45
4 Project summary .....	51
5 Ábrajegyzék .....	52

## Plágium nyilatkozat

Alulírott Kilián Marcell András, Korcsmáros Kristóf György, Takács Balázs Levente kijelentjük, hogy ez a vizsgaremek saját tudásunk, önálló munkánk terméke. A vizsgaremek közös részeit pontosan jelöltük.

Budapest, 2021.

.....  
Kilián Marcell  
András

.....  
Korcsmáros Kristóf  
György

.....  
Takács Balázs  
Levente

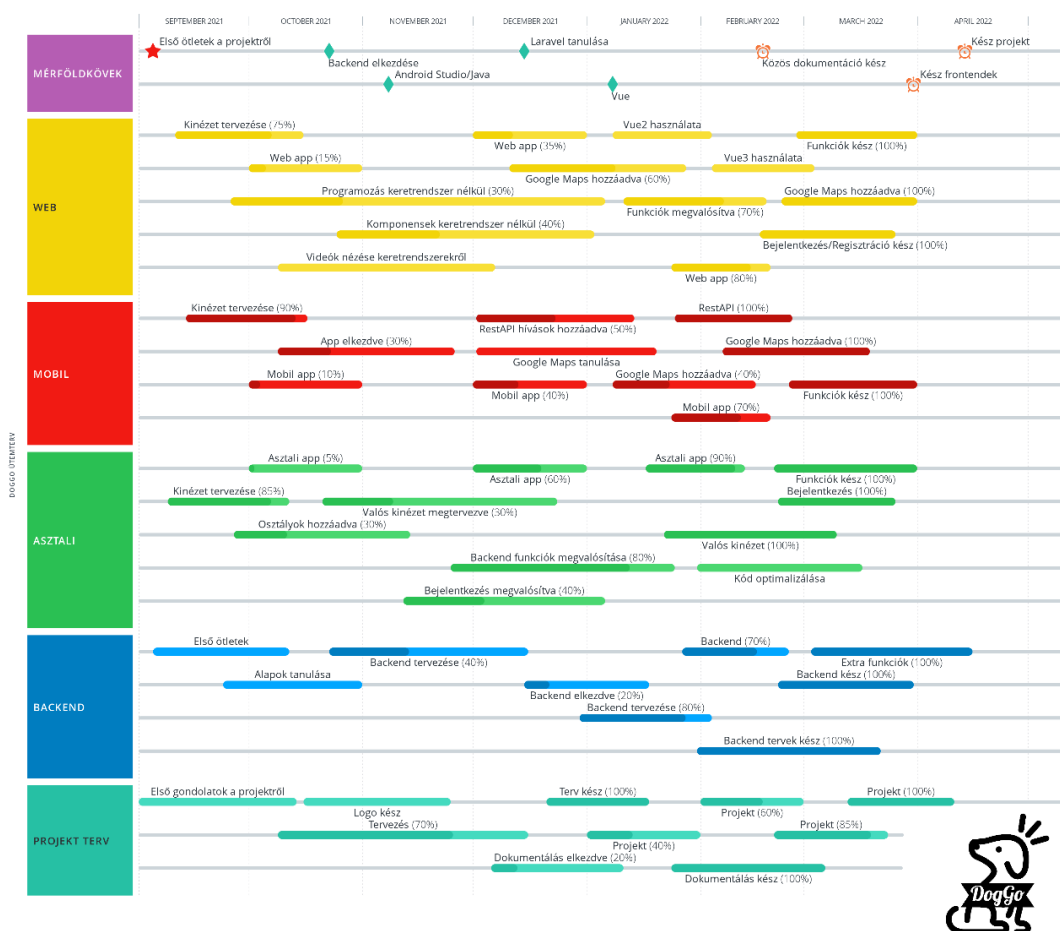
## 2 Bevezetés

Különböző forrásokból hallani, ismerősöktől, közösségi médiából azt, hogy egy adott helyre elvihetik kedvenceiket. Ennek ellenére mégsem tudunk sok véleményt meghallgatni, olvasni az adott helyekről. Ez az alkalmazás ennek a problémának a megoldására kínál lehetőséget.

Egy olyan alkalmazást készítünk, ahova az emberek leírhatják véleményüket a közelükben lévő, számukra jelentéssel bíró helyekről. Ha más ember is hallaná a pozitív tapasztalatokat az adott helyről, tudná, hogy milyen kutyás emberek járnak milyen helyekre, lehet, hogy ő is elmenne és kipróbálni a mások által megjelölt helyeket.

A másik fő célunk ezzel az alkalmazással az, hogy kutyás csoportokat, társaságokat hozzunk létre. A helyek értékelésével, leírásával az emberek kapcsolatba léphetnek egymással, motiválhatják egymást, hogy kimozduljanak otthonról kiskedvenceikkel.

Ezek a tulajdonságok, amelyek egyedivé teszik ezt a programot. Nincs másik program egyelőre, ami akár csak egy délutáni kutyasétáltatást összehozna más emberekkel.



1. ábra DogGo ütemterv

## 2.1 Téma

Téma a kutyasétáltatás. Mivel a választott témának egyedinek kell lennie, ezért olyan problémákat kerestünk a mindennapi alkalmazásokban, amikre egy alternatív program sincs. A kutyákat manapság nem lehet akárhová vinni, külön engedéllyel, bizonyos időközönként vagy egyáltalán nem lehet számos helyre menni velük. Emellett a közösségi médiában is sokszor látni olyat, amikor az emberek segítséget kérnek, hogy mégis hová vihetik a kutyáikat. Ezért lenne jó, egy olyan alkalmazás, ahol minden kutyasétáltatással kapcsolatos információ megtalálható egy adott helyhez, parkhoz.

Mit tud a szoftver? Egy térképen láthatunk megjelölt helyeket, amiket emberek tudnak az alkalmazáshoz rendelni, regisztrálás, bejelentkezés után. Az alkalmazást lehet látogatóként is használni, ebben az esetben, csak megnézhetjük, hogy milyen helyeket, milyen értékelésekkel jelöltek meg az emberek, alkalmas kutyasétáltatásra, azonban nem jelölhetnek meg helyeket.

A helymegjelöléshez tartoznak a képek, egy 5-ös skálán értékelések és a kommentek. Ha egy helyet alkalmasnak tartunk kutyasétáltatásra, akkor megjelölhetjük azt a pozíciót a térképen, hozzárendelhetünk képeket, megoszthatjuk, hogy mennyire tetszett az a hely egy 5-ös skálán és szöveges értékelést (kommentet) csatolhatunk a megjelölt pozícióhoz.

Rosszakaró emberek mindig is léteztek, mindenhol megjelennek. Az asztali alkalmazás azért jött létre, hogy adminisztrátor jogosultsággal szűrni lehessen ezeket a felhasználókat. Ha valaki rengeteg rossz, elfogadhatatlan helyet jelölne meg, mint például egy autópálya közepe, ahol nem feltétlenül biztonságos házikedvencünket sétáltatni, az adminisztrátor jogosultsággal tiltani tudjuk a felhasználókat.

Abban az esetben, ha egy felhasználó hibát észlel, továbbítani tudja a fejlesztőknek. A visszajelzés anonim, nem kell regisztrálni, bejelentkezni ennek a funkciónak a használatához. Az alkalmazás későbbi fejlesztése miatt adjuk hozzá ezt a funkciót az alkalmazáshoz.

## 2.2 Témaválasztás indoklása

Manapság a háztartások nagy részében hatalmas szerepet játszik a háziállat, azonban kiskedvenceink a legtöbb helyről ki vannak tiltva. Szerettünk volna olyan témát választani, amivel megtudjuk könnyíteni, azoknak az embereknek az életét, akik mindenhol a kutyájukkal mennének. Gondolkoztunk, hogy hogyan is lehetne ezt a problémát egy alkalmazás segítségével orvosolni, így kezdődött el a DogGo.

Számos esetleges megoldás jutott az eszünkbe, az egyik ilyen megoldás az lenne, hogy mi felsorolunk elterjedt, számunkra izgalmas, jó, kellemes helyeket, jelezve a felhasználóknak, hogy hova mehetnek a kiskedvencükkel sétálni. Azonban ezek a helyek három ember által kedvelt helyek lennének, attól, hogy egy helyet kedvel három ember, nem feltétlenül jelenti azt, hogy az emberek többsége is kedvelné. Ezért ezt a megoldást elvetettük. Rájöttünk, hogy minél több ember mutat számára kutyasétáltatásra alkalmas helyet, annál több ember fog, számára megfelelő helyet találni a sétáltatásra, ahova ő is elmenne kiskedvencét sétáltatni.

Emiatt jutottunk arra a megoldásra, hogy a felhasználók véleményét kell megjelenítenünk egy felületen, mivel minél több ember osztja meg a véleményét, annál több ember talál számára alkalmas helyet az alkalmazás segítségével. Minél nagyobb a választék a helyek közül annál több felhasználó használhatja az alkalmazást. Mivel a helyeket felhasználók adják hozzá, ők is valószínűleg megjelennek a számukra kellemes helyeken. Ezáltal motiválhatjuk a felhasználókat a kimozdulásra. Nagyobb eséllyel megy el sétálni egy ember, ha tudja, hogy beszélgethet, találkozhat másokkal. Ezeken felül, ha sikerülne kialakítani, az alkalmazás segítségével egy állatbarát közösséget, akkor egymást is ösztönöznék egy tartalmas sétára, találkozásra.

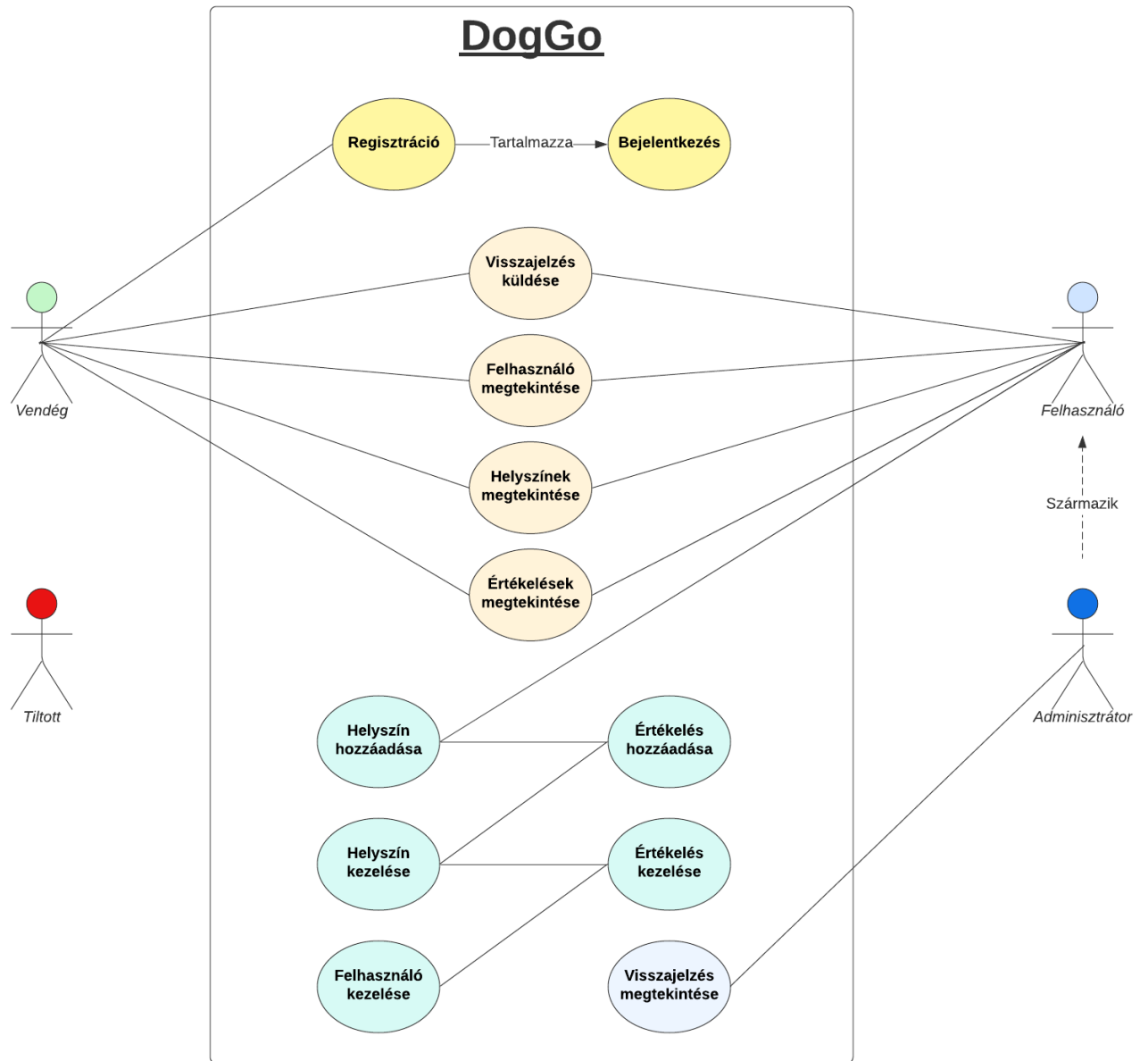
Ez az ötlet a fejlesztői csoport minden tagjának tetszett, a lelkesedést és egyetértést látva a projekt témája meg is született.

### 3 Fejlesztői dokumentáció

#### 3.1 Funkciók

	Látogató	Regisztrált felhasználó	Adminisztrátor
Regisztráció	✓	✗	✗
Bejelentkezés	✗	✓	✓
Visszajelzés küldése	✓	✓	✓
Hely megtekintése	✓	✓	✓
Értékelések megtekintése	✓	✓	✓
Hely hozzáadása	✗	✓	✓
Értékelés hozzáadása	✗	✓	✓
Saját hely kezelése	✗	✓	✓
Saját értékelés kezelése	✗	✓	✓
Összes hely kezelése	✗	✗	✓
Összes értékelés kezelése	✗	✗	✓
Felhasználók kezelése	✗	✗	✓
Visszajelzések kezelése	✗	✗	✓

## 3.2 Használati eset diagram



2. ábra: Használati eset diagram

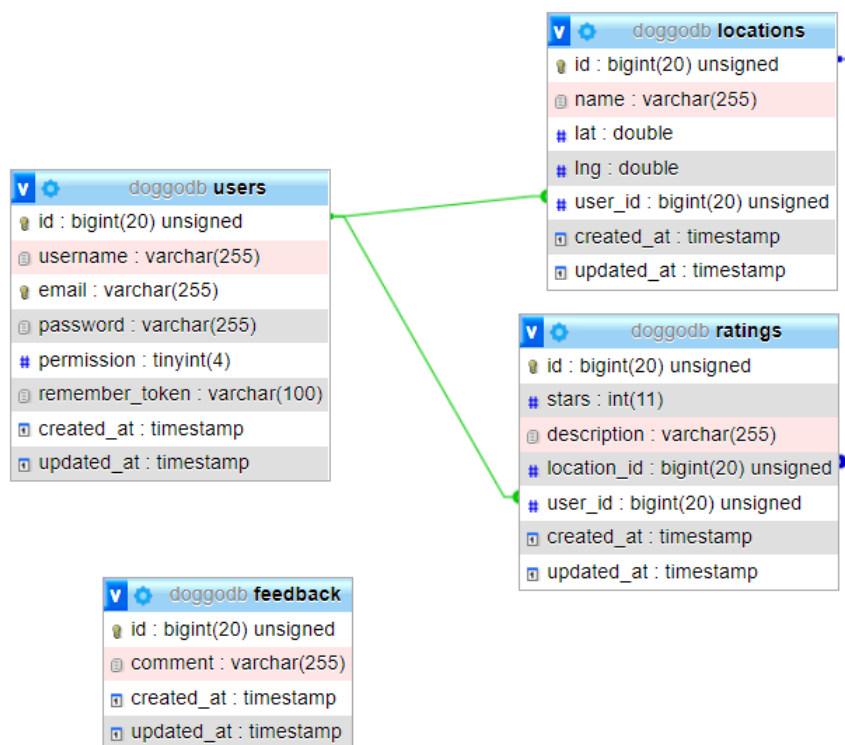


### 3.3 Backend

#### 3.3.1 Adatbázis terv

Az adatbázis az egyik legfontosabb rész a projekt működéséhez. Itt tároljuk a **felhasználók, helyek, értékelések** adatait és a **viisszajelzéseket**, amikkel fejleszteni, javítani tudjuk az alkalmazásunkat.

A projekt backend részét közösen csináljuk. Megbeszélünk egy időpontot és online folytatjuk a munkát. A táblák létrehozásával kezdtük a backend megvalósítását. Az adatbázis terv miatt egyszerű volt a táblák létrehozása. **Laravel** keretrendszert használtunk az elképzelt adatbázis megvalósításához. Néhány utasítással könnyen lehet a táblákat feltölteni *teszt adatokkal*, így egy sokkal *átláthatóbb* kezdetleges végeredményt látunk munkánk során.



3. ábra: Adatbázis terv

### 3.3.2 Adattáblák (doggodb)

users tábla

Oszlopnév	Típus	Hossz	Tartalma
<b>id</b>	BIGINT, PK, AI	20	Elsődleges kulcs
<b>username</b>	VARCHAR	5-20	Felhasználó felhasználóneve
<b>email</b>	VARCHAR	255	Felhasználó E-mail címe
<b>password</b>	VARCHAR	8-	Felhasználó jelszava
<b>permission</b>	TINYINT	4	Milyen jogosultsággal rendelkezik a felhasználó: 0 – default 1 – tiltva 2 – admin 3 – super admin
<b>created_at</b>	TIMESTAMP		Létrehozás dátuma

locations tábla

Oszlopnév	Típus	Hossz	Tartalom
<b>id</b>	BIGINT, PK, AI	20	Eldősleges kulcs
<b>name</b>	VARCHAR	5-40	Hely neve
<b>description</b>	VARCHAR	255	Hely leírása
<b>lat</b>	DOUBLE		Hely koordinátája (szélesség)
<b>lng</b>	DOUBLE		Hely koordinátája (hosszúság)
<b>user_id</b>	BIGINT, FK		Hivatkozás a users táblára

ratings tábla

Oszlopnév	Típus	Hossz	Tartalom
<b>id</b>	BIGINT, PK, AI	20	Elsődleges kulcs
<b>stars</b>	INT	1-5	Számos értékelés
<b>description</b>	VARCHAR	255	Szöveges értékelés
<b>location_id</b>	BIGINT, FK	20	Hivatkozás a locations táblára
<b>user_id</b>	BIGINT, FK	20	Hivatkozás a users táblára

feedback tábla

Oszlopnév	Típus	Hossz	Tartalom
<b>id</b>	BIGINT, PK, AI	20	Elsődleges kulcs
<b>comment</b>	VARCHAR	255	Visszajelzés szövege
<b>created_at</b>	TIMESTAMP		Létrehozás dátuma

### 3.3.3 Backend telepítésének lépései

Ahhoz, hogy elérjük az adatbázist, szükségünk lesz egy adatbázis kiszolgálóra. Ehhez legegyszerűbben használható a **XAMPP** telepítő csomag, melyet az alábbi linken tudunk letölteni:

<https://www.apachefriends.org/hu/download.html>

Telepítés után indítsuk el a **MySQL** kiszolgálót, ezután nyissuk meg a **phpmyadmin**-t és hozzunk létre egy adatbázist „doggodb” néven *utf8mb4\_hungarian\_ci* karakterkódolással.

A backend projekt megnyitása után, készítsünk egy másolatot a *.env.example* fájlról és nevezzük át *.env*-re. A fájlban írjuk át megfelelőre az adatbázis kapcsolat adatait.

Hajtsuk végre a következő utasításokat a konzolban:

1. composer install
2. php artisan key:generate --ansi
3. php artisan migrate
4. php db:seed
5. Indítsuk el a fejlesztői szerveret:
  6. php artisan serve

Teszteljük **Thunder Client** vagy **Postman** segítségével, hogy az alábbi URL megfelelő **JSON** adatot ad-e vissza:

<http://127.0.0.1:8000/api/users>

### 3.3.4 API végpontok

A program **backend** része úgy lett megvalósítva, hogy az adatot **JSON** formátumban adja és várja. Az adatbázis **factory**-k és **seeder**-ek segítségével, **tesztadatokkal** töltöttük fel.

Az megadott adatoknak meg kell felelnie az adatbázisban megadott feltételeknek (*adatbázis feltételek: 9-10. oldal*), a **JSON** példákban emiatt nincs *feltüntetve*, hogy mik a határértékek.

#### users végpont csoport

##### **GET /api/users**

Visszaadja a felhasználók adatait.

Például: GET <http://localhost:8000/api/users>

```
[
  {
    "id": 1,
    "username": "Dr. Quinten VonRueden",
    "email": "hlind@example.com",
    "permission": 1
  },
  {
    "id": 2,
    "username": "Rhea Schowalter",
    "email": "russel.hugh@example.org",
    "permission": 1
  }
]
```

##### **GET /api/users/{id}**

Visszaadja az adott ID-val rendelkező felhasználó adatait.

Például: GET <http://localhost:8000/api/users/5>

```
{
  "id": 5,
  "username": "Destinee Tillman",
  "email": "hagenes.irving@example.org",
  "permission": 1
}
```

## POST /api/users

Létrehoz egy új felhasználót a megadott adatokkal. Az ID-n kívül minden adat megadása kötelező. A jelszót **titkosítva** tároljuk el az adatbázisban, **bcrypt** titkosítást használunk, ez a laravel keretrendszer alapértelmezett titkosítási módszere.

Egy titkosított jelszó:

*\$2y\$10\$pGp/1o8W8qedKt5ChlRuX.pdl3WbwNAuKcicl41St0fIFX2aSGvC*

Sikeres hozzáadás esetén a létrehozott felhasználó adatai beleértve a generált ID-t visszaadja.

Például: POST <http://localhost:8000/api/users>

Bemenet:

```
{
  "username": "test",
  "email": "test@example.net",
  "password": "test",
  "permission": 1
}
```

Eredmény:

```
{
  "username": "test",
  "email": "test@example.net",
  "permission": 1,
  "id": 16
}
```

### **PUT /api/users/{id}**

Módosítja az adott ID-val rendelkező felhasználó adatait. Csak a módosítani kívánt adatokat kell megadni. Hogyha csak a felhasználónevet szeretnénk módosítani elég azt megadni:

Sikeres módosítás után visszaadja a módosított felhasználó adatait. Az ID nem módosítható.

Például: PUT <http://127.0.0.1:8000/api/users/5>

Bemenet:

```
{  
  "username": "test"  
}
```

Eredmény:

```
{  
  "id": 5,  
  "username": "test",  
  "email": "hagenes.irving@example.org",  
  "permission": 1  
}
```

### **DELETE /api/users/{id}**

Kitörli az adatbázisból az adott ID-val rendelkező felhasználót.

Például: DELETE <http://localhost:8000/api/users/1>

Eredmény:

Status: 204 No Content

### **Hibakezelés**

Hibás végpont esetén, vagy, ha az adatok nem felelnek meg a követelményeknek, a backend jelzi ezt.

A hibának megfelelő HTTP státuszkódot adja vissza (400-zal kezdődő), és a visszakapott JSON "message" tulajdonsága tartalmazza a hiba okát.

Például: GET <http://localhost:8000/api/users/9999> (nem létező id)

Status: 404 Not Found

```
{  
  "message": "A megadott azonosítóval nem található felhasználó"  
}
```

## **locations végpont csoport**

### **GET /api/locations**

Visszaadja a helyszínek adatait.

Például: GET <http://localhost:8000/api/locations>

```
[
  {
    "id": 1,
    "name": "Yasmine Oval",
    "lat": 44.240438,
    "lng": 84.739212,
    "user_id": 1
  },
  {
    "id": 2,
    "name": "Nicolette Trace",
    "lat": 14.781673,
    "lng": -24.293254,
    "user_id": 1
  }
]
```

### **GET /api/locations/{id}**

Visszaadja az adott ID-val rendelkező helyszín adatait.

Például: GET <http://localhost:8000/api/locations/5>

```
{
  "id": 5,
  "name": "Elmore Turnpike",
  "lat": -17.084211,
  "lng": -171.558352,
  "user_id": 1
}
```



## **POST /api/locations**

Létrehoz egy új helyszínt a megadott adatokkal. Az ID-n kívül minden adat megadása kötelező.

Sikeres hozzáadás esetén a létrehozott helyszín adatai beleértve a generált ID-t visszaadja.

Például: POST <http://127.0.0.1:8000/api/locations>

Bemenet:

```
{
  "name": "test",
  "lat": 44.240438,
  "lng": 84.739212,
  "user_id": 1
}
```

Eredmény:

```
{
  "id": 5,
  "name": "Elmore Turnpike",
  "lat": -17.084211,
  "lng": -171.558352,
  "user_id": 1
}
```

## **PUT /api/locations/{id}**

Módosítja az adott ID-val rendelkező helyszín adatait. Csak a módosítani kívánt adatokat kell megadni. Hogyha csak a nevet szeretnénk módosítani elég azt megadni:

Sikeres módosítás után visszaadja a módosított helyszín adatait. Az ID nem módosítható.

Például: PUT <http://127.0.0.1:8000/api/locations/5>

Bemenet:

```
{
  "name": "test"
}
```

Eredmény:

```
{
  "id": 5,
  "name": "test",
  "lat": -17.084211,
  "lng": -171.558352,
  "user_id": 1
}
```

## **DELETE /api/locations/{id}**

Kitörli az adatbázisból az adott ID-val rendelkező helyszínt.

Például: DELETE <http://localhost:8000/api/locations/1>

Eredmény:

Status: 204 No Content

## **Hibakezelés**

Hibás végpont esetén, vagy, ha az adatok nem felelnek meg a követelményeknek, a backend jelzi ezt.

A hibának megfelelő HTTP státuszkódot adja vissza (400-zal kezdődő), és a visszakapott JSON "message" tulajdonsága tartalmazza a hiba okát.

Például: GET <http://localhost:8000/api/locations/9999> (nem létező id)

Status: 404 Not Found

```
{  
  "message": "A megadott azonosítóval nem található helyszín."  
}
```

## **ratings végpont csoport**

### **GET /api/ratings**

Visszaadja az értékelések adatait.

Például: GET <http://localhost:8000/api/ratings>

```
[
  {
    "id": 1,
    "stars": 1,
    "description": "Quia similique corporis ratione placeat sed sequi.",
    "location_id": 1,
    "user_id": 1
  },
  {
    "id": 2,
    "stars": 3,
    "description": "Id quo facere tempore iste aliquid dolor.",
    "location_id": 1,
    "user_id": 1
  }
]
```

### **GET /api/ratings/{id}**

Visszaadja az adott ID-val rendelkező értékelés adatait.

Például: GET <http://localhost:8000/api/ratings/5>

```
{
  "id": 5,
  "stars": 3,
  "description": "Id perspiciatis consequatur dignissimos tempora.",
  "location_id": 1,
  "user_id": 1
}
```

## **POST /api/ratings**

Létrehoz egy új értékelést a megadott adatokkal. Az ID-n kívül minden adat megadása kötelező.

Sikeres hozzáadás esetén a létrehozott értékelés adatai beleértve a generált ID-t visszaadja.

Például: POST <http://127.0.0.1:8000/api/ratings>

Bemenet:

```
{
  "stars": 3,
  "description": "test",
  "location_id": 1,
  "user_id": 1
}
```

Eredmény:

```
{
  "stars": 3,
  "description": "test",
  "location_id": 1,
  "user_id": 1,
  "id": 16
}
```

## **PUT /api/ratings/{id}**

Módosítja az adott ID-val rendelkező értékelés adatait. Csak a módosítani kívánt adatokat kell megadni. Hogyha csak a szöveges értékelést szeretnénk módosítani elég azt megadni:

Sikeres módosítás után visszaadja a módosított értékelés adatait. Az ID nem módosítható.

Például: PUT <http://127.0.0.1:8000/api/ratings/5>

Bemenet:

```
{
  "description": "test"
}
```

Eredmény:

```
{
  "id": 5,
  "stars": 3,
  "description": "test",
  "location_id": 1,
  "user_id": 1
}
```

## **DELETE /api/ratings/{id}**

Kitörli az adatbázisból az adott ID-val rendelkező értékelést.

Például: DELETE <http://localhost:8000/api/ratings/1>

Eredmény:

Status: 204 No Content

## **Hibakezelés**

Hibás végpont esetén, vagy, ha az adatok nem felelnek meg a követelményeknek, a backend jelzi ezt.

A hibának megfelelő HTTP státuszkódot adja vissza (400-zal kezdődő), és a visszakapott JSON "message" tulajdonsága tartalmazza a hiba okát.

Például: GET <http://localhost:8000/api/ratings/9999> (nem létező id)

Status: 404 Not Found

```
{  
  "message": "A megadott azonosítóval nem található értékelés."  
}
```

### **GET /api/best\_rating**

Visszaadja a legjobb értékeléssel rendelkező helyszín átlag értékelését és nevét.

Például: GET [http://localhost:8000/api/best\\_rating](http://localhost:8000/api/best_rating)

```
{
  "name": "Sedrick Land",
  "atlag": "5.0"
}
```

### **GET /api/worst\_rating**

Visszaadja a legrosszabb értékeléssel rendelkező helyszín átlag értékelését és nevét.

Például: GET [http://localhost:8000/api/worst\\_rating](http://localhost:8000/api/worst_rating)

```
{
  "name": "Mossie Common",
  "atlag": "1.7"
}
```

### **GET /api/rating\_by\_user/{id}**

Visszaadja az adott ID-val rendelkező felhasználó értékeléseit.

Például: GET [http://localhost:8000/api/rating\\_by\\_user/1](http://localhost:8000/api/rating_by_user/1)

```
[
  {
    "id": 1,
    "stars": 2,
    "description": null,
    "location_id": 12,
    "user_id": 1
  },
  {
    "id": 2,
    "stars": 1,
    "description": "Doloremque quo iure assumenda sint quae blanditiis qui.",
    "location_id": 10,
    "user_id": 1
  }
]
```

### **GET /api/locations\_allowed**

Visszaadja az engedélyezett helyszínek adatait.

Például: GET [http://127.0.0.1:8000/api/locations\\_allowed](http://127.0.0.1:8000/api/locations_allowed)

```
[
  {
    "id": 2,
    "name": "Chadd Courts",
    "description": "As she said to Alice; and Alice was not much.",
    "lat": 5.633529,
    "lng": -161.676676,
    "allowed": true,
    "user_id": 1
  }
]
```

### **GET /api/locations\_not\_allowed**

Visszaadja a még nem engedélyezett helyszínek adatait.

Például: GET [http://127.0.0.1:8000/api/locations\\_not\\_allowed](http://127.0.0.1:8000/api/locations_not_allowed)

```
[
  {
    "id": 1,
    "name": "Lockman Plains",
    "description": "Alice. 'And where HAVE my shoulders got to.",
    "lat": 68.222227,
    "lng": -67.375361,
    "allowed": false,
    "user_id": 1
  }
]
```

### **GET /api/feedbacks\_read**

Visszaadja a már olvasott visszajelzések adatait.

Például: GET [http://127.0.0.1:8000/api/feedbacks\\_read](http://127.0.0.1:8000/api/feedbacks_read)

```
[
  {
    "id": 1,
    "comment": "Aperiam explicabo consectetur qui qui ea enim.",
    "read": true,
    "created_at": "2022-03-13T15:33:01.000000Z"
  }
]
```

### **GET /api/feedbacks\_not\_read**

Visszaadja a még nem olvasott visszajelzések adatait.

Például: GET [http://127.0.0.1:8000/api/feedbacks\\_not\\_read](http://127.0.0.1:8000/api/feedbacks_not_read)

```
[
  {
    "id": 3,
    "comment": "Molestiae molestiae totam dicta voluptate exercitationem.",
    "read": false,
    "created_at": "2022-03-13T15:33:01.000000Z"
  }
]
```

### **GET /api/read\_feedback\_count**

Visszaadja az olvasott visszajelzések számát.

Például: GET [http://127.0.0.1:8000/api/read\\_feedback\\_count](http://127.0.0.1:8000/api/read_feedback_count)

Eredmény: 7

### **GET /api/new\_feedback\_count**

Visszaadja az újonnan küldött visszajelzések számát.

Például: GET [http://127.0.0.1:8000/api/new\\_feedback\\_count](http://127.0.0.1:8000/api/new_feedback_count)

Eredmény: 8

### **POST /api/login**

Visszaadja a felhasználó **token**-jét, amivel megkaphatjuk a felhasználó adatait.

Például: POST <http://127.0.0.1:8000/api/login>

Eredmény: 1ljXgAZiSVq9o6gyFw6uDsH3niXSLFT5n9VGtU6Hk1



### 3.4 DogGo - Web alkalmazás

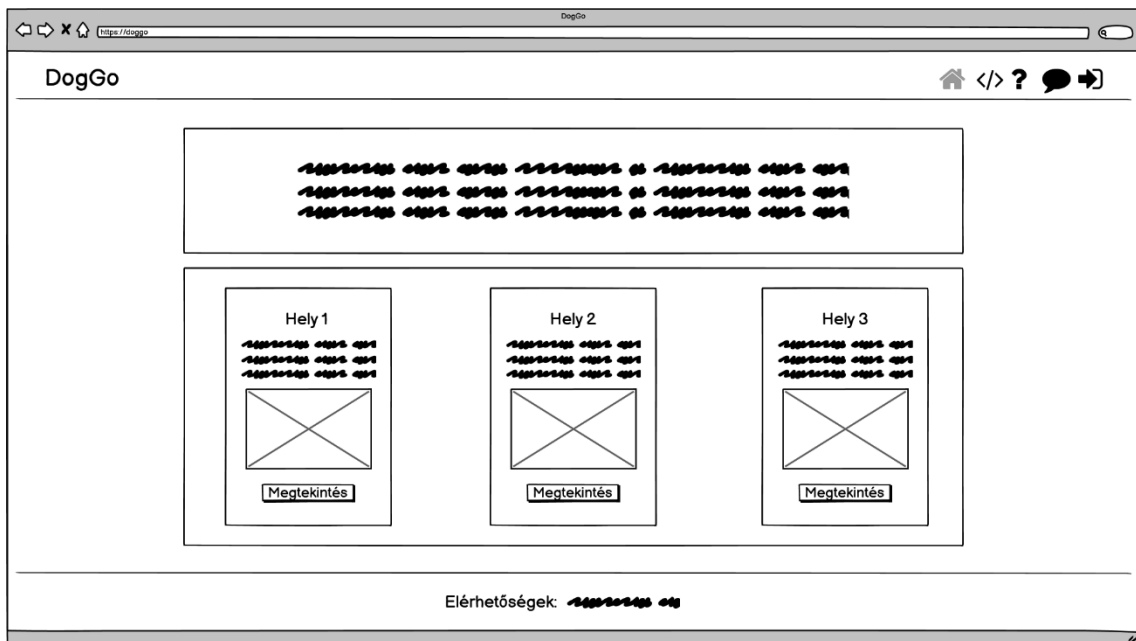
A weboldalon a felhasználók által kedvelt helyeket tekinthetjük meg és értékelhetjük azokat. Az általunk kedvelt helyeket is publikálhatjuk az alkalmazáson belül.

A webes alkalmazás lehetőséget nyújt az alapvető funkciókon kívül (regisztráció, bejelentkezés, hely és értékelés hozzáadása és kezelése) az admin jogosultsággal rendelkező felhasználóknak a helyek és értékelések kezelésére is.

#### 3.4.1 Felületterv

Az alkalmazás kinézetét igyekszem minél letisztultabbra, átláthatóbbra csinálni. A weboldal kinézetének tervét a **Balsamiq** nevű program segítségével készítettem el.

A főoldalon jelenítem meg az alkalmazás leírását és a helyeket. Az oldal tetején egy navigációs sávot láthatunk majd, az alján egy lábléce, ahol a fejlesztőkkel kapcsolatos, alapvető információkat láthatjuk.



4. ábra: Főoldal váza

Mivel a jelszót nem lehet egyelőre email üzenet használatával megváltoztatni, ezt a funkciót nem tettem bele az oldal vázába.

The image shows a web browser window with the address bar displaying 'https://doggo'. The page title is 'DogGo'. The main content area contains a registration form with the following fields: 'Felhasználónév:' (Username), 'Email:', 'Jelszó:' (Password), and 'Jelszó ismét:' (Repeat Password). Below these fields is a 'Regisztrálás' (Register) button. At the bottom right of the form area, there is a link that says 'Van már fiókom' (I already have an account).

5. ábra: Regisztrációs oldal váza

A bejelentkezés és regisztráció oldal közötti navigációt a gombok alatti részen valósítottam meg. A két oldal nagyon hasonló, középre terveztem egy dobozt, ahova a be-

The image shows a web browser window with the address bar displaying 'https://doggo'. The page title is 'DogGo'. The main content area contains a login form with the following fields: 'Felhasználónév:' (Username) and 'Jelszó:' (Password). Below these fields is a 'Bejelentkezés' (Login) button. At the bottom right of the form area, there is a link that says 'Nincs még fiókom' (I don't have an account yet).

6. ábra: Bejelentkezés oldal váza

meneti értékek helyes kitöltésével végrehajthatjuk a műveleteket.

### **3.4.2 Rendszerkövetelmények**

#### **Minimum rendszerkövetelmények**

- Google Chrome v57
- Microsoft Edge v79
- Opera v44
- Safari v11
- Firefox v76

#### **Ajánlott rendszerkövetelmények**

Legújabb verzió a felsorolt böngészők közül:

- Google Chrome
- Microsoft Edge
- Opera
- Safari
- Firefox

#### **A program telepítése**

Az alkalmazást nem lehet telepíteni. Az alkalmazás a használt böngészőn keresztül érthető el.

### **3.5 Fejlesztői dokumentáció**

#### **3.5.1 *Alkalmazott fejlesztői eszközök***

##### **Felhasznált jelölő, leíró és programozási nyelvek**

1. HTML
2. CSS
3. JavaScript
4. Vue

##### **Bővítmények és külső könyvtárak**

1. BootstrapVue
2. Font Awesome
3. Vuex
4. Vuelidate

##### **Keretrendszer, fejlesztői környezet**

1. Vue
2. Visual Studio Code

##### **Helyi webszerver**

1. XAMPP
2. Apache
3. MySQL phpMyAdmin

##### **Egyéb felhasznált programok**

1. Google chrome
2. Balsamiq
3. Microsoft Word
4. Microsoft PowerPoint
5. Microsot Paint

### 3.5.2 *Komponensek*

#### Komponensek függőségei

Az alkalmazás ikonjait **Font Awesome** segítségével valósítottam meg.

- "@fortawesome/fontawesome-svg-core": "^1.3.0"
- "@fortawesome/free-solid-svg-icons": "^6.0.0"
- "@fortawesome/vue-fontawesome": "^3.0.0-5"

A bemeneti értékek ellenőrzéséhez a **Vuelidate** könyvtárat használtam.

- "@vuelidate/core": "^2.0.0-alpha.35"
- "@vuelidate/validators": "^2.0.0-alpha.27"

A végpontokkal való kommunikálást az Axios használatával implementáltam.

- "axios": "^0.26.0"
- "vue-axios": "^3.4.1"
- "bootstrap": "^5.1.3"

Az alkalmazást a **Vue 3.0** használatával készítettem el.

- "vue": "^3.0.0"

Az alkalmazás továbbfejlesztésénél a felhasználó helyét megjeleníteném a térképen.

- "vue-browser-geolocation": "^1.8.0"

Az egyszerűbb navigálás érdekében a **Vue Router** kiegészítőt használom.

- "vue-router": "^4.0.14"

A felhasználó adatainak eltárolására a **Vuex** könyvtárat választottam.

- "vuex": "^4.0.2"

### 3.5.3 *Komponensek dokumentálása*

#### **Login komponens**

<b>Komponens leírása</b>	A komponens eltárolja a bejelentkezett felhasználó token-jét.
<b>Adattagok</b>	<b>Leírás</b>
registeredUser.username : String	A felhasználónév beviteli mezőbe írt szöveget tárolja el.
registeredUser.password : String	A jelszó beviteli mezőbe írt szöveget tárolja el.
token : String	A sikeresen bejelentkezett felhasználó token-jét tárolja el.
error : Boolean	Ez az érték mutatja, hogy hibás volt-e a bejelentkezés.
errorMessage : String	A hiba leírását tárolja el.
<b>Metódusok</b>	<b>Leírás</b>
login()	A metódus továbbítja a felhasználó által beírt adatokat a szervernek, helyes bejelentkezési adatok esetén a felhasználó token-jét elmenti. Helytelen adatok esetében hibaüzenetet ad vissza az eljárás.
mounted()	Megvizsgálja, hogy a token létezik-e, ha létezik, a főoldalra navigál minket.

### Register komponens

<b>Komponens leírása</b>	A komponens új felhasználók létrehozását valósítja meg.
<b>Adattagok</b>	<b>Leírás</b>
token : String	A regisztrált felhasználó token-jét tárolja el.
error : Boolean	Ez az érték mutatja, hogy hibás volt-e a regisztráció.
errorMessage : String	A hiba leírását tárolja el.
state.registerUser.username : String	A felhasználónév beviteli mezőbe írt szöveget tárolja el.
state.registerUser.email : String	Az email beviteli mezőbe írt szöveget tárolja el.
state.registerUser.password : String	A jelszó beviteli mezőbe írt szöveget tárolja el.
state.confirmPassword : String	A jelszó megerősítése beviteli mezőbe írt szöveget tárolja el.
<b>Metódusok</b>	<b>Leírás</b>
register()	Elküldi a beviteli mezőkbe írt adatokat a szervernek, ezekkel az adatokkal próbál egy új felhasználót létrehozni a program.
mounted()	Megvizsgálja, hogy a token létezik-e, ha létezik, a főoldalra navigál minket.

### Footer komponens

<b>Komponens leírása</b>	A weboldal láblécét jeleníti meg a komponens.
<b>Adattagok</b>	<b>Leírás</b>
-	-
<b>Metódusok</b>	<b>Leírás</b>
-	-

### Header komponens

<b>Komponens leírása</b>	A weboldal fejlécét jeleníti meg a komponens.
<b>Adattagok</b>	<b>Leírás</b>
-	-
<b>Metódusok</b>	<b>Leírás</b>
signOut()	Eltávolítja a token adattagot, a user adattagnak üres értéket állít be, visszavigágál minket a bejelentkezés oldalra.
getUserDatas()	A felhasználó token-jével elkéri a felhasználó adatait és eltárolja.
mounted()	Ha a token adattag létezik, meghívja a getUserDatas() metódust.

### Error komponens

Komponens leírása	Ezen a komponensen keresztül lehet az alkalmazás hibáit továbbítani a fejlesztőknek.
<b>Adattagok</b>	<b>Leírás</b>
message : String	A szöveges visszajelzését tárolja el a hibaüzenet elküldésének sikerességéről.
error : Integer	Ez az érték mutatja, hogy hibás volt-e a hibaüzenet elküldése.
sent : Boolean	Ez az érték mutatja, hogy elküldtük-e az üzenetet.
state.errorMessage.comment : String	A hibaüzenetet tárolja el.
<b>Metódusok</b>	<b>Leírás</b>
sendError()	A beviteli mezőbe írt hibaüzenetet próbálja meg elküldeni a szervernek.



## Locations komponens

<b>Komponens leírása</b>	Ez a komponens jeleníti meg a jóváhagyott helyeket.
<b>Adattagok</b>	<b>Leírás</b>
locations : Array	Ez a tömb tárolja el az engedélyezett helyeket.
bestRating : Array	A legjobb átlagosan értékelt hely adatait ez a tömb tárolja el.
editing : Boolean	Ez az érték jelzi, hogy egy adminisztrátori jogosultsággal rendelkező felhasználó szerkeszt-e egy helyet.
saving : Boolean	A többszöri mentést akadályozza meg.
admin : Integer	Az adminisztrátori jogosultság értékét tárolja el.
state.location.name : String	A hely neve beviteli mezőbe írt szöveget tárolja el.
state.location.lat : Integer	A hely szélessége beviteli mezőbe írt szöveget tárolja el.
state.location.lng : Integer	A hely hosszúsága beviteli mezőbe írt szöveget tárolja el.
state.location.description : String	A hely leírása beviteli mezőbe írt szöveget tárolja el.
state.location.allowed : Boolean	A hely engedélyezését tárolja el.
state.location.user_id : Integer	A hely felhasználójának azonosítóját tárolja el.
<b>Metódusok</b>	<b>Leírás</b>
loadData()	Ez a metódus kéri el a szerverről az engedélyezett helyek adatait és eltárolja a szerverről kapott helyeket.
checkLocationRating(id) : Integer	Átírányít minket a Ratings komponensre, az adott hely értékeléseit jeleníti meg.
bannLocation(id, locName) : Integer, String	Tiltja az id értékű helyet, a második paraméter a hely neve.
enableLocation(id, locName) : Integer, String	Engedélyezi az id értékű helyet, a második paraméter a hely neve.
editLocation(id) : Integer	Szerkeszti az id értékű helyet.
saveLocation()	Elmenti a szerkesztett helyet.
cancelLocation()	A szerkesztésből kilép az eljárás.
resetForm()	A beviteli mezőket alaphelyzetbe állítja.
mounted()	Meghívja a loadData() eljárást.

## MyLocations komponens

<b>Komponens leírása</b>	A saját helyeinket tudjuk kezelni ezzel a komponenssel.
<b>Adattagok</b>	<b>Leírás</b>
locations : Array	A saját helyeinket tárolja el.
add_new : Boolean	Azt mutatja, hogy éppen új helyet adunk-e hozzá az alkalmazáshoz.
saving : Boolean	A többszöri mentést akadályozza meg.
state.location.name : String	A hely neve beviteli mezőbe írt szöveget tárolja el.
state.location.lat : Integer	A hely szélessége beviteli mezőbe írt szöveget tárolja el.
state.location.lng : Integer	A hely hosszúsága beviteli mezőbe írt szöveget tárolja el.
state.location.description : String	A hely leírása beviteli mezőbe írt szöveget tárolja el.
state.location.allowed : Boolean	A hely engedélyezését tárolja el.
state.location.user_id : Integer	A hely felhasználójának azonosítóját tárolja el.
<b>Metódusok</b>	<b>Leírás</b>
changeUserId()	A metódus ellenőrzi, hogy a felhasználó be van-e jelentkezve, ha nincs bejelentkezve a felhasználó, a főoldalra navigálja a program.
loadData()	Ez az eljárás kéri és tárolja el a felhasználó helyeit, amiket publikált az alkalmazáson belül.
newLocation()	A bemeneti értékekkel megpróbál létrehozni egy helyet a metódus.
deleteLocation(id) : Integer	Az id azonosítójú helyet törli az eljárás.
editLocation(id) : Integer	Az id azonosítójú helyet szerkeszti a metódus, megjeleníti a beviteli mezőkben a hely adatait, módosíthatóvá teszi őket.
saveLocation()	Ez a metódus módosítja a hely adatait. Az új beviteli mezőbe írt adatokkal frissíti a kiválasztott helyet.
resetForm()	Alaphelyzetbe állítja a beviteli mezők értékét.
mounted()	Meghívja a changeUserId() metódust.

## Ratings komponens

<b>Komponens leírása</b>	Egy hely értékeléseit jeleníti meg a komponens.
<b>Adattagok</b>	<b>Leírás</b>
locationRatingId : Integer	A hely azonosítóját tárolja el. Azt mutatja, hogy melyik helynek kell az értékeléseit listázni.
ratings : Array	Ez a tömb tárolja el a hely értékeléseit.
locationRating : Array	A hely átlagos értékelését tárolja el.
locationData : Array	A hely adatait tárolja el.
message : String	Az értékelés hozzáadásának szöveges sikerességét jelzi az üzenet.
error : Integer	Ez az érték mutatja, hogy hibás volt-e az értékelés elküldése.
sent : Boolean	Ez az érték mutatja, hogy elküldtük-e az értékelést.
editing : Boolean	Azt mutatja, hogy éppen szerkesztjük-e az értékelésünket.
show : Boolean	Az értékelés közzétételének üzenetét jeleníti meg, illetve tünteti el.
id : Integer	Az értékelés szerkesztésekor, ez az érték jelenti az értékelés azonosítóját.
state.description : String	A szöveges értékelésünket tárolja el.
state.stars : Integer	A skálás értékelésünket tárolja el.
<b>Metódusok</b>	<b>Leírás</b>
checkEnabledLocation()	Ellenőrzi, hogy az értékelt hely engedélyezve van-e.
loadDatas()	Betölti a kiválasztott hely értékeléseit, átlagos értékelését és adatait.
addRating()	A mezőkben lévő adatokkal egy új értékelést ad a helyhez.
editRating(id) : Integer	Betölti a módosítani kívánt értékelésünk adatait a beviteli mezőkbe.
changeRating()	Az értékelésünket módosítja.
cancelEdit()	Kilép az értékelés szerkesztéséből és visszaállítja a beviteli mezők értékeit.
deleteRating(id) : Integer	Az id azonosítójú értékelésünket eltávolítjuk.
resetForm()	Visszaállítja a beviteli mezők értékeit.
mounted()	Meghívja a checkEnabledLocation metódust.

### Text komponens

<b>Komponens leírása</b>	A projekt alapvető leírását tartalmazza a komponens.
<b>Adattagok</b>	<b>Leírás</b>
-	-
<b>Metódusok</b>	<b>Leírás</b>
-	-

### Info komponens

<b>Komponens leírása</b>	A projekt részletesebb leírását jeleníti meg ez a komponens.
<b>Adattagok</b>	<b>Leírás</b>
-	-
<b>Metódusok</b>	<b>Leírás</b>
-	-

### UserData komponens

<b>Komponens leírása</b>	A felhasználó nevét és email címét módosíthatjuk.
<b>Adattagok</b>	<b>Leírás</b>
error : Boolean	Ez az érték mutatja, hogy hibás volt-e a felhasználó adatainak frissítése.
errorMessage : String	A szöveges visszajelzését tárolja el a felhasználó adatainak frissítéséről.
state.modifyUser.username : String	A felhasználónév beviteli mezőbe írt szöveget tárolja el.
state.modifyUser.email : String	Az email beviteli mezőbe írt szöveget tárolja el.
<b>Metódusok</b>	<b>Leírás</b>
change()	Ez a metódus frissíti a felhasználó adatait a beviteli mező értékeivel.
loadUserDatas()	Betölti a beviteli mezőkbe a felhasználó adatait.
mounted()	Meghívja a loadUserDatas() metódust.

### ErrorPage komponens

<b>Komponens leírása</b>	A hibajelentés oldalt jelenti ez a komponens.
<b>Adattagok</b>	<b>Leírás</b>
-	-
<b>Metódusok</b>	<b>Leírás</b>
-	-
<b>Komponensek</b>	Error

### HomePage komponens

<b>Komponens leírása</b>	A főoldalt jelenti ez a komponens.
<b>Adattagok</b>	<b>Leírás</b>
-	-
<b>Metódusok</b>	<b>Leírás</b>
-	-
<b>Komponensek</b>	Text, Locations, Footer

### InfoPage komponens

<b>Komponens leírása</b>	Az információs oldalt jelenti ez a komponens.
<b>Adattagok</b>	<b>Leírás</b>
-	-
<b>Metódusok</b>	<b>Leírás</b>
-	-
<b>Komponensek</b>	Info, Footer

### LoginPage komponens

<b>Komponens leírása</b>	A bejelentkezés oldalt jelenti ez a komponens.
<b>Adattagok</b>	<b>Leírás</b>
-	-
<b>Metódusok</b>	<b>Leírás</b>
-	-
<b>Komponensek</b>	Login

### MyLocationsPage komponens

<b>Komponens leírása</b>	A saját helyek oldalt jelenti ez a komponens.
<b>Adattagok</b>	<b>Leírás</b>
-	-
<b>Metódusok</b>	<b>Leírás</b>
-	-
<b>Komponensek</b>	MyLocations

### RatingsPage komponens

<b>Komponens leírása</b>	Az értékelés oldalt jelenti ez a komponens.
<b>Adattagok</b>	<b>Leírás</b>
-	-
<b>Metódusok</b>	<b>Leírás</b>
-	-
<b>Komponensek</b>	Ratings

### RegisterPage komponens

<b>Komponens leírása</b>	A regisztrálás oldalt jelenti ez a komponens.
<b>Adattagok</b>	<b>Leírás</b>
-	-
<b>Metódusok</b>	<b>Leírás</b>
-	-
<b>Komponensek</b>	Register

### UserDataPage komponens

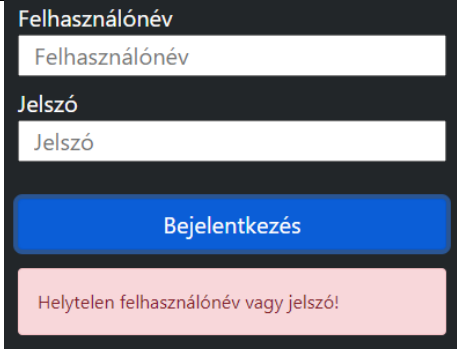
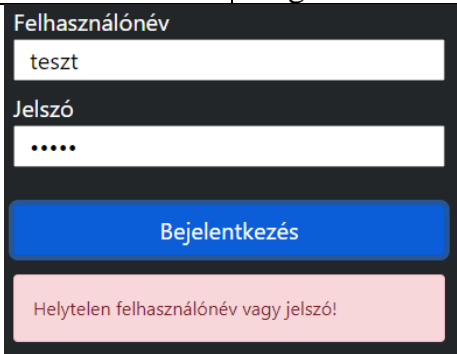
<b>Komponens leírása</b>	A felhasználó adatainak oldalát jelenti ez a komponens.
<b>Adattagok</b>	<b>Leírás</b>
-	-
<b>Metódusok</b>	<b>Leírás</b>
-	-
<b>Komponensek</b>	Userdata

### 3.5.4 Tesztelési dokumentáció

#### Regisztráció

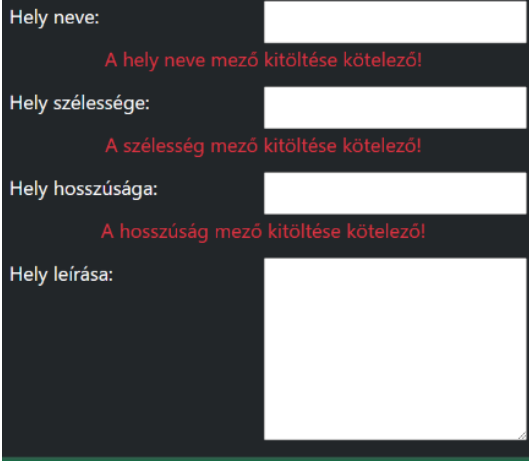
Művelet	Bemenet	Elvárt eredmény	Végeredmény
Üresen hagyott bemeneti mezők	Felhasználónév: - Email: - Jelszó: - Jelszó ismét: -	Minden bemeneti mező alatt hiba-üzenet jelenik meg	Minden bemeneti mező alatt hiba-üzenet jelenik meg
 <p>7. ábra: Regisztráció - Üresen kitöltött bemeneti mezők</p>			
Helytelen adatokkal kitöltött bemeneti mezők	Felhasználónév: t Email: t Jelszó: t Jelszó ismét: t	Minden bemeneti mező alatt megfelelő hibaüzenet jelenik meg, kivéve a jelszó ismét mező alatt	Minden bemeneti mező alatt megfelelő hibaüzenet jelenik meg, kivéve a jelszó ismét mező alatt
Már létező adatokkal kitöltött bemeneti mezők	Felhasználónév: <code>teszt_user_1</code> Email: <a href="mailto:teszt1@gmail.com">teszt1@gmail.com</a> Jelszó: <code>teszt123</code> Jelszó ismét: <code>teszt123</code>	Az oldalon "Ilyen felhasználó már létezik!" hibaüzenet jelenik meg	Az oldalon "Ilyen felhasználó már létezik!" hibaüzenet jelenik meg
Helyes adatokkal kitöltött bemeneti mezők	Felhasználónév: <code>teszt1</code> Email: <a href="mailto:teszt1@gmail.com">teszt1@gmail.com</a> Jelszó: <code>teszt123</code> Jelszó ismét: <code>teszt123</code>	Az alkalmazás a főoldalra navigál minket, a navigációs sáv jobb felső sarkában a felhasználónévünket láthatjuk	Az alkalmazás a főoldalra navigál minket, a navigációs sáv jobb felső sarkában a felhasználónévünket láthatjuk
 <p>8. ábra: Regisztráció - Felhasználó ikon</p>			

## Bejelentkezés

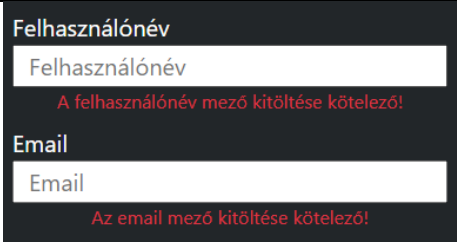
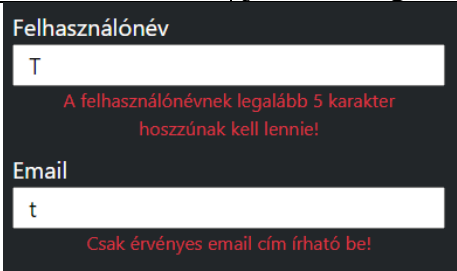
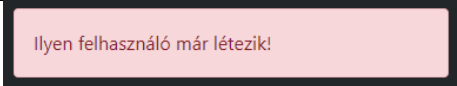
Művelet	Bemenet	Elvárt eredmény	Végeredmény
Üresen hagyott bemeneti mezők	Felhasználónév: - Jelszó: -	Az oldalon hiba- üzenet jelenik meg	Az oldalon hiba- üzenet jelenik meg
 <p>9. ábra: Bejelentkezés - Üresen kitöltött mezők</p>			
Helytelen adatokkal kitöltött bemeneti mezők	Felhasználónév: teszt Jelszó: teszt	Az oldalon hiba- üzenet jelenik meg	Az oldalon hiba- üzenet jelenik meg
 <p>10. ábra: Bejelentkezés - Helytelen adatokkal kitöltött mezők</p>			
Helyes adatokkal kitöltött bemeneti mezők	Felhasználónév: teszt1 Jelszó: teszt123	Az alkalmazás a főoldalra navigál minket, a navigá- ciós sáv jobb felső sarkában a felhasználónevün- ket láthatjuk	Az alkalmazás a főoldalra navigál minket, a navigá- ciós sáv jobb felső sarkában a felhasználónevün- ket láthatjuk



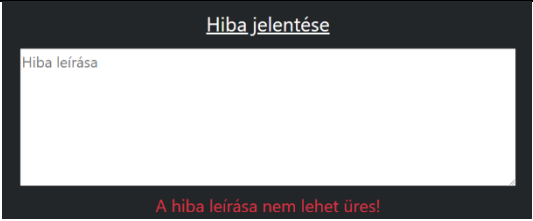
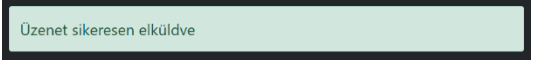
## Hely hozzáadása

Művelet	Bemenet	Elvárt eredmény	Végeredmény
Üresen hagyott bemeneti mezők	Hely neve: - Hely szélessége: - Hely hosszúsága: - Hely leírása: -	Minden bemeneti mező alatt megfelelő hibaüzenet jelenik meg, kivéve a Hely leírása mező alatt, mert nem kötelező a mező kitöltése	Minden bemeneti mező alatt megfelelő hibaüzenet jelenik meg, kivéve a Hely leírása mező alatt, mert nem kötelező a mező kitöltése
 <p>11. ábra: Hely hozzáadása - Üresen kitöltött mezők</p>			
Helytelen adatokkal kitöltött bemeneti mezők	Hely neve: T Hely szélessége: 1 Hely hosszúsága: 1 Hely leírása: Teszt	A Hely neve mező alatt hibaüzenet jelenik meg, mert nincs megfelelő adattal kitöltve	A Hely neve mező alatt hibaüzenet jelenik meg, mert nincs megfelelő adattal kitöltve
Helyes adatokkal kitöltött bemeneti mezők	Hely neve: Teszt Hely szélessége: 1 Hely hosszúsága: 1 Hely leírása: Teszt	A helyeink között megjelenik a hozzáadott hely	A helyeink között megjelenik a hozzáadott hely

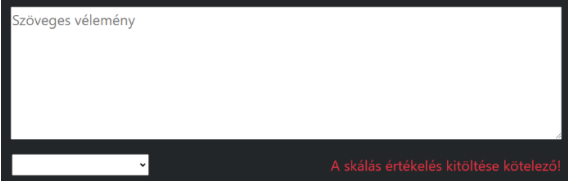
## Adataim módosítása

Művelet	Bemenet	Elvárt eredmény	Végeredmény
Üres bemeneti mezők	Felhasználónév: - Email: -	Minden bemeneti mező alatt megfelelő hibaüzenet jelenik meg	Minden bemeneti mező alatt megfelelő hibaüzenet jelenik meg
 <p>12. ábra: Adataim módosítása - Üresen kitöltött mezők</p>			
Helytelen adatokkal kitöltött bemeneti mezők	Felhasználónév: T Email: t	Minden bemeneti mező alatt megfelelő hibaüzenet jelenik meg	Minden bemeneti mező alatt megfelelő hibaüzenet jelenik meg
 <p>13. ábra: Adataim módosítása - Helytelen adatokkal kitöltött mezők</p>			
Már létező adatokkal kitöltött bemeneti mezők	Felhasználónév: admin Email: teszt@gmail.com	Az oldalon "Ilyen felhasználó már létezik!" hibaüzenet jelenik meg	Az oldalon "Ilyen felhasználó már létezik!" hibaüzenet jelenik meg
 <p>14. ábra: Adataim módosítása – Létező felhasználó hibaüzenet</p>			
Helyes adatokkal kitöltött bemeneti mezők	Felhasználónév: Test1 Email: teszt@gmail.com	Az alkalmazás a főoldalra navigál minket, a navigációs sáv jobb felső sarkában az új felhasználóvünket láthatjuk	Az alkalmazás a főoldalra navigál minket, a navigációs sáv jobb felső sarkában az új felhasználóvünket láthatjuk

## Hiba jelentése

Művelet	Bemenet	Elvárt eredmény	Végeredmény
Üres bemeneti mező	Hiba leírása: -	A bemeneti mező alatt megfelelő hibaüzenet jelenik meg	A bemeneti mező alatt megfelelő hibaüzenet jelenik meg
 <p>15. ábra: Hiba jelentése – Üresen kitöltött mezők</p>			
Helytelen adattal kitöltött bemeneti mező	Hiba leírása: (255 * t)	A bemeneti mező alatt megfelelő hibaüzenet jelenik meg	A bemeneti mező alatt megfelelő hibaüzenet jelenik meg
Helyes adattal kitöltött bemeneti mező	Hiba leírása: teszt	”Az üzenet sikeresen elküldve” üzenet jelenik meg az oldalon	”Az üzenet sikeresen elküldve” üzenet jelenik meg az oldalon
 <p>16. ábra: Hiba jelentése – Sikeresen publikált értékelés</p>			

## Értékelés hozzáadása

Művelet	Bemenet	Elvárt eredmény	Végeredmény
Üres bemeneti mezők	Szöveges vélemény: - Skálás értékelés: -	A bemeneti mezők alatt megfelelő hibaüzenet jelenik meg	A bemeneti mezők alatt megfelelő hibaüzenet jelenik meg
 <p>17. ábra: Értékelés hozzáadása - Üresen kitöltött mezők</p>			
Helytelen adatokkal kitöltött bemeneti mezők	Szöveges vélemény: 256 * t Skálás értékelés: -	A bemeneti mezők alatt megfelelő hibaüzenet jelenik meg	A bemeneti mezők alatt megfelelő hibaüzenet jelenik meg
Helyes adatokkal kitöltött bemeneti mezők	Szöveges vélemény: Teszt Skálás értékelés: 3	”Értékelés sikeresen közzétéve!” üzenet jelenik meg az oldalon	”Értékelés sikeresen közzétéve!” üzenet jelenik meg az oldalon

### **3.5.5 Továbbfejlesztési lehetőségek**

A projekt kezdetekor ezt a funkcióját meg szeretnénk volna valósítani a weboldalon, de sajnos a kevés idő miatt továbbfejlesztési lehetőség marad. A weboldalon a helyeket egy térképen terveztük megjeleníteni. A Google térképet terveztük használni a felhasználók által publikált helyek megjelenítéséhez, kezeléséhez.

Emellett továbbfejlesztési lehetőség még, a helyekhez csatolható kép. Sokkal nagyobb élményt nyújthatnánk a felhasználóknak, ha vizuálisan nagyobb örömforrást biztosítanánk az alkalmazás használata közben.

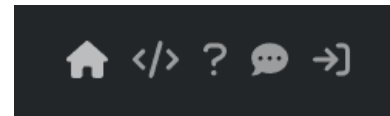
Előfordul, hogy elfelejtjük jelszavainkat. Emiatt fontos, hogy alkalmunk legyen módosítani ezt később is. A biztonságos megvalósítás miatt ezt email üzenet hitelesítésével valósítanám meg.

Számtalan helyzet van, amikor a felhasználó nem tudja eldönteni, hogy mi a hiba a weboldallal. Több visszajelzés a szerverről, töltési animációk is megkönnyebbíthetik a felhasználói élményt.

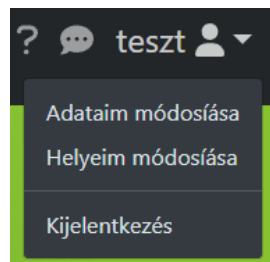
### 3.6 Felhasználói dokumentáció

Az alkalmazás megnyitásával a weboldal **főoldalát** láthatjuk. Az applikáció minden oldalán található egy *navigációs sáv*, a könnyebb navigálás érdekében.

Az alkalmazás tetején a navigációs sávot láthatjuk. A különböző ikonokra kattintva navigálhatjuk magunkat az oldalak között. A navigációk sorba a **főoldalt**, a **for-**



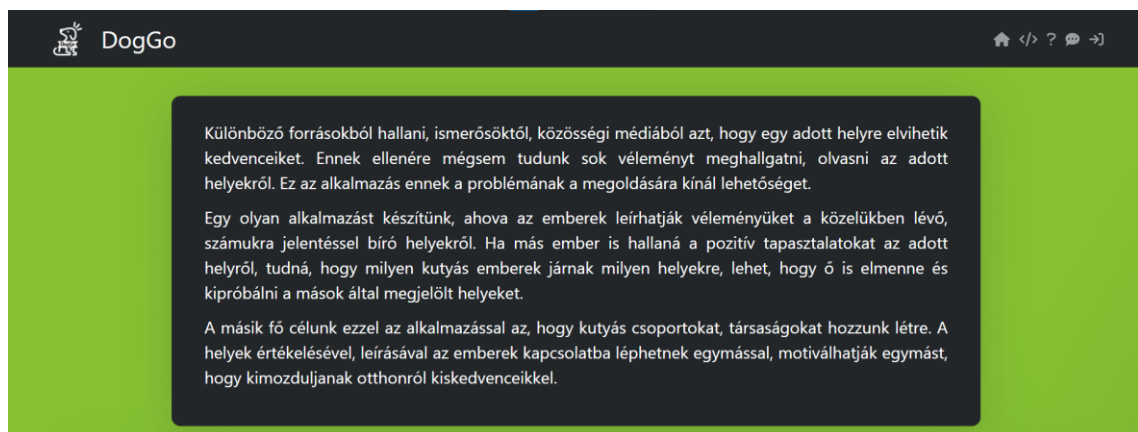
18. ábra: Navigációs sáv ikonjai



19. ábra: Felhasználó ikon

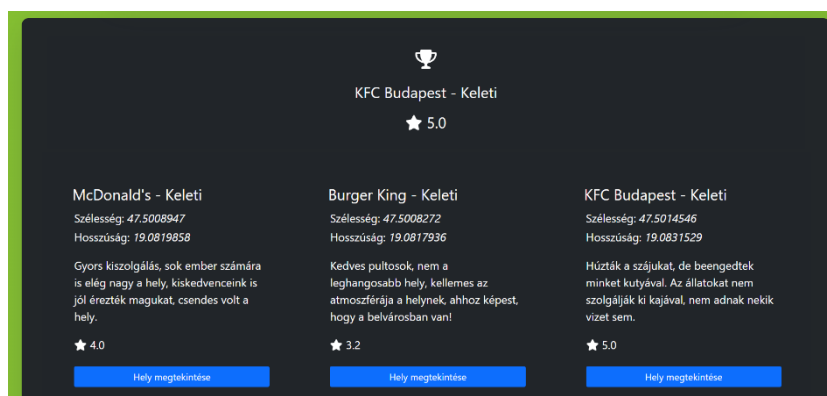
**ráskód** megtekintését, **információt** a weboldalról, a **hiba jelentést** és a **regisztrációt**, illetve **bejelentkezést** jelentik. Abban az esetben, ha bejelentkeztünk, a regisztráció ikon helyett a *saját felhasználónevi*nk jelenik meg, egy legördülő menüpontként, ahol a menüpontok sorba a felhasználó **adatainak** módosítását, a felhasználó **helyeinek** módosítását és a **kijelentkezést** jelentik.

A főoldal tetején az alkalmazás bemutatkozása található. Itt megérthetik a felhasználók, hogy miért is jött létre ez az alkalmazás, mi az alkalmazás alapvető célja.



21. ábra: Főoldal felső része

nálók, hogy miért is jött létre ez az alkalmazás, mi az alkalmazás alapvető célja.

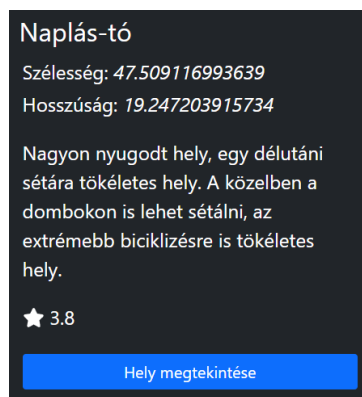


20. ábra: Megjelenített helyek

A bemutatkozás után az alkalmazás **helyeit** láthatjuk. A helyek tetején található a *legjobb értékeléseket* kapott hely. Ezt a helyet ajánlották leginkább a felhasználók.

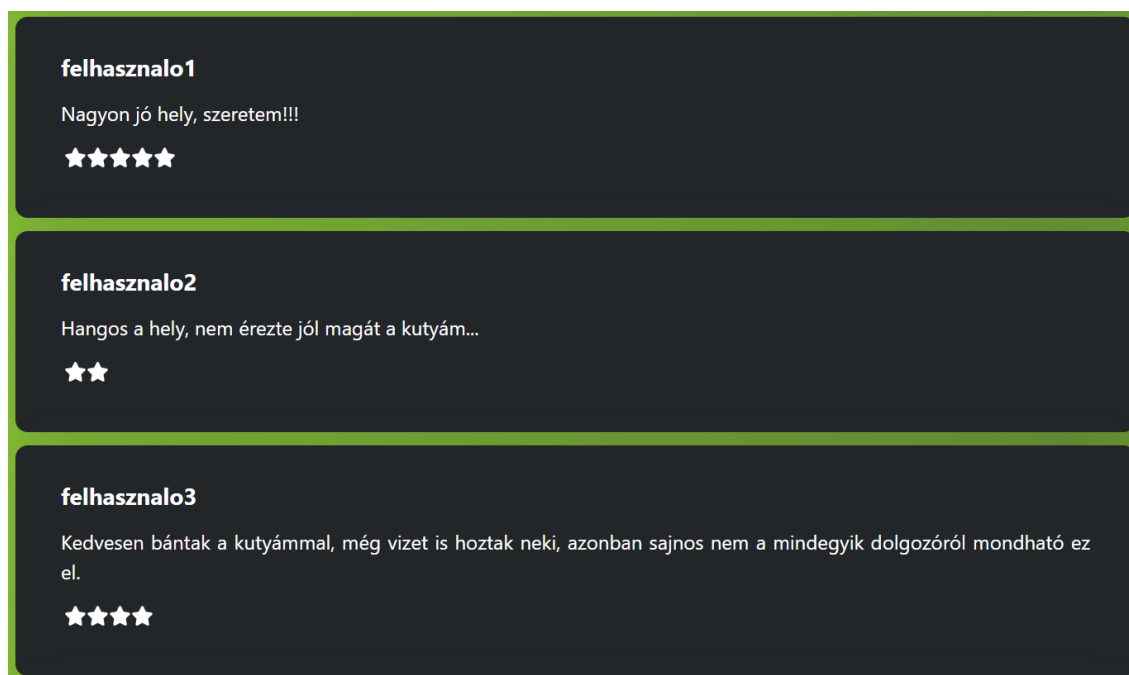
## Értékelések

Abban az esetben, ha a hely kapott legalább egy **értékelést**, a hely átlagos értékelését láthatjuk a hely tulajdonságai között (a hely leírása alatt). A helyek tulajdonságai sorba a hely **neve**, a hely **szélessége**, **hosszúsága**, a hely **leírása**, a hely **átlagos értékelés** és az **értékelések** megtekintése, amelyet a *"Hely megtekintése"* gombra kattintva tekinthetünk meg. Abban az esetben ha meg szeretnénk nézni egy hely értékeléseit, a gombra kattintva egy új oldalra navigál minket az alkalmazás.



22. ábra: Egy hely kinézete

A helyek *értékeléseinek megtekintéséhez* nem kell bejelentkezni, *vendég felhasználóként* is megtekinthetjük az értékeléseket. Ellenkező esetben, a helyeket csak akkor lehet értékelni, ha bejelentkezett felhasználóként használjuk az alkalmazást.



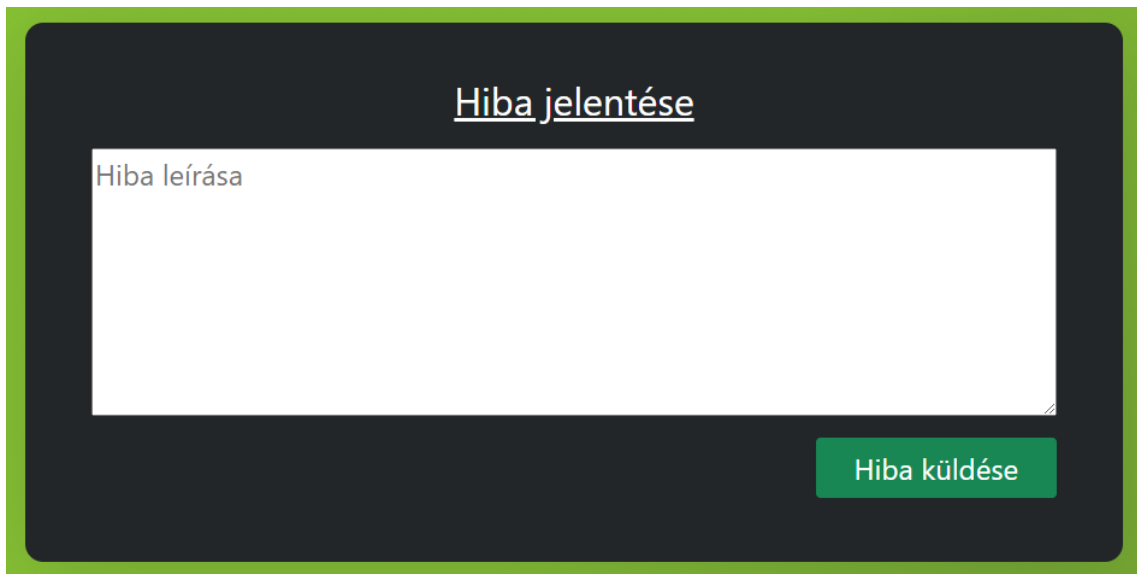
23. ábra: Egy hely értékelései

Egy értékelés tulajdonságai sorra a következők, a **felhasználónév**, azt jelenti, hogy melyik felhasználó értékelését láthatjuk, a **szöveges értékelés** és a **skálás értékelés**. A szöveges értékelés legfeljebb 255 karakter lehet, a skálás értékelés pedig egy egytől ötig tartó skálás értékelés.

Az értékelés *sikeres*, illetve *sikertelen* közzétételéről visszajelzést kapunk. Abban az esetben, ha nincs megfelelően kitöltve az értékelésünk, értesítést kapunk, csak úgy, mint minden beviteli mezőnél az alkalmazásban.

## Hiba jelentése

Abban az esetben ha valamilyen hibát észlel a felhasználó, lehetősége van ezt a hibát továbbítani a fejlesztőknek a *hibajelentés* fülön.



24. ábra: Hiba jelentése

A mező helyes kitöltése után, a hiba elküldéséről szintén visszajelzést kapunk. A hiba jelentéséhez nem kötelező a bejelentkezés.

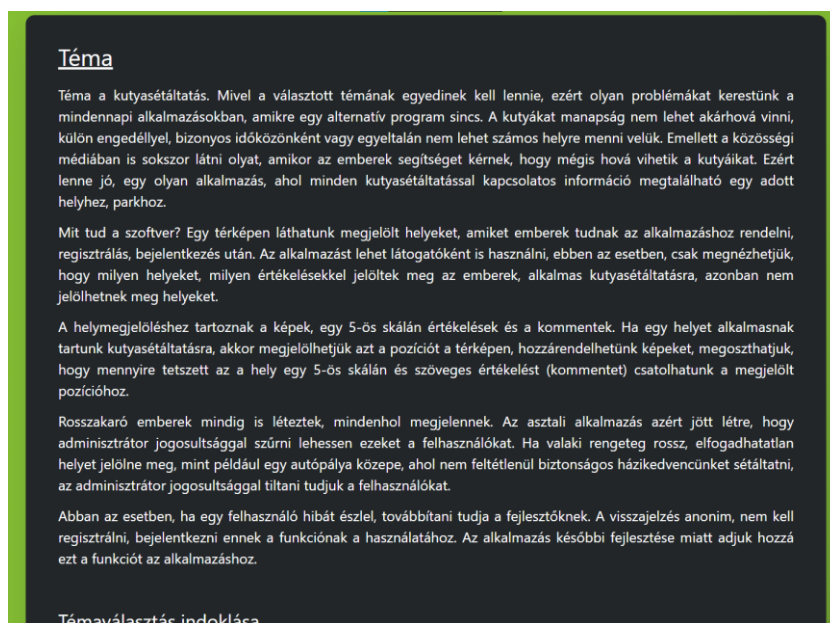


Üzenet sikeresen elküldve

25. ábra: Sikeres hibaüzenet küldése

## Információ

Az *információ* oldalon az alkalmazás részletesebb ismertetését láthatjuk, ezen az oldalon nincs semmilyen funkció a részletek elolvasásán kívül.



**Téma**

Téma a kutyasétáltatás. Mivel a választott témának egyedinek kell lennie, ezért olyan problémákat kerestünk a mindennapi alkalmazásokban, amikre egy alternatív program sincs. A kutyákat manapság nem lehet akárhová vinni, külön engedéllyel, bizonyos időközönként vagy egyáltalán nem lehet számos helyre menni velük. Emellett a közösségi médiában is sokszor látni olyat, amikor az emberek segítséget kérnek, hogy mégis hová vihetik a kutyáikat. Ezért lenne jó, egy olyan alkalmazás, ahol minden kutyasétáltatással kapcsolatos információ megtalálható egy adott helyhez, parkhoz.

Mit tud a szoftver? Egy térképen láthatunk megjelölt helyeket, amiket emberek tudnak az alkalmazáshoz rendelni, regisztrálás, bejelentkezés után. Az alkalmazást lehet látogatóként is használni, ebben az esetben, csak megnézhetjük, hogy milyen helyeket, milyen értékelésekkel jelölték meg az emberek, alkalmas kutyasétáltatásra, azonban nem jelölhetnek meg helyeket.

A helymegjelöléshez tartoznak a képek, egy 5-ös skálán értékelések és a kommentek. Ha egy helyet alkalmasnak tartunk kutyasétáltatásra, akkor megjelölhetjük azt a pozíciót a térképen, hozzárendelhetünk képeket, megoszthatjuk, hogy mennyire tetszett az a hely egy 5-ös skálán és szöveges értékelést (kommentet) csatolhatunk a megjelölt pozícióhoz.

Rosszakaró emberek mindig is léteztek, mindenhol megjelennek. Az asztali alkalmazás azért jött létre, hogy adminisztrátor jogosultsággal szűrni lehessen ezeket a felhasználókat. Ha valaki rengeteg rossz, elfogadhatatlan helyet jelölne meg, mint például egy autópálya közepe, ahol nem feltétlenül biztonságos házikedvencünket sétáltatni, az adminisztrátor jogosultsággal tiltani tudjuk a felhasználókat.

Abban az esetben, ha egy felhasználó hibát észlel, továbbítani tudja a fejlesztőknek. A visszajelzés anonim, nem kell regisztrálni, bejelentkezni ennek a funkciónak a használatához. Az alkalmazás későbbi fejlesztése miatt adjuk hozzá ezt a funkciót az alkalmazáshoz.

Témaválasztás indoklása

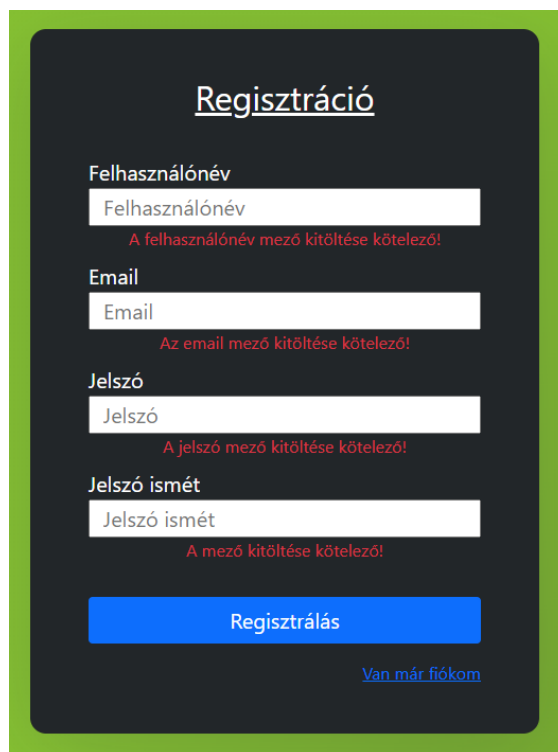
26. ábra: Az alkalmazás részletesebb leírása

## Regisztráció

A "regisztrálás" gombra kattintva, a regisztrációs oldalra kerül a felhasználó. Négy mező kitöltése *kötelező* a regisztráláshoz. Helyes beviteli értékek esetén, a regisztrálás gombra kattintva a regisztráció sikerességéről visszajelzést kapunk.



28. ábra: Regisztrációs oldal kinézete



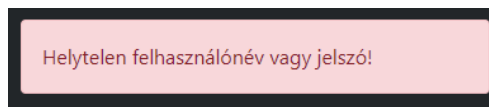
27. ábra: Regisztrációs hibajelzés

Sikeres regisztrálás esetén automatikusan bejelentkezik az alkalmazás, emiatt látjuk felhasználónevünket a fejléc jobb oldalán, sikeres regisztráció után. Ellenkező esetben, például, ha olyan felhasználót próbálunk létrehozni, amelyik már létezik, a következő hibaüzenet jelenik meg a regisztráció oldalán.

## Bejelentkezés

Abban az esetben, ha van már felhasználói fiókja és abba szeretne a felhasználó bejelentkezni, a "Van már fiókomban" gombra kattintva tud bejelentkezni a felhasználó.

A mezők helyes kitöltése után a "Bejelentkezés" gombra kattintva bejelentkezünk, helyesen kitöltött mezők esetében a navigációs sáv jobb oldalán a felhasználónevünket láthatjuk és értékeléseket és helyeket is hozzá tudunk adni az alkalmazáshoz. Ha a bejelentkezés sikertelen, hibaüzenet jelenik meg.



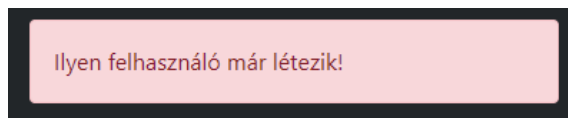
29. ábra: Hibás bejelentkezés



## Felhasználó adatainak módosítása

Abban az esetben ha módosítani kívánja adatait, a navigációs sávon a felhasználónévére kattintva, a legördülő menü *"Adataim módosítása"* menüpontjára kattintva a weboldal átnavigál minket az *adataim módosítása* oldalra. A **felhasználónév**, illetve az **email cím** módosítására van lehetőségünk. Az oldal betöltésekor a beviteli mezők értékei automatikusan betöltődnek. A mezők megfelelő kitöltése után a *"Módosítás"* gombra

30. ábra: Adataunk módosítása oldal



31. ábra: Létező felhasználó hibaüzenet

kattintva tudjuk módosítani az adatainkat. A módosítás sikeréről visszajelzést kapunk. Az adataink sikeres módosítása után a főoldalra navigál minket a program,

a navigációs sávon pedig az **új felhasználónevünket** láthatjuk.

Abban az esetben, ha sikertelen volt adataink módosítása, hibaüzenet jelenik meg az adataim módosítása oldalon.

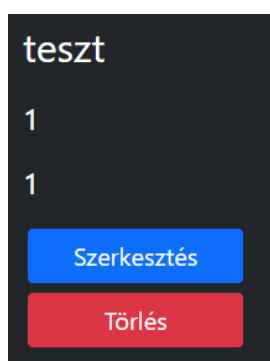
## Saját helyek

Ha magunk akarunk egy helyet hozzáadni az alkalmazáshoz, a *"Helyeim módosítása"* menüpontra kattintva ezt megtehetjük.

Az alkalmazás a **Saját helyek kezelése** oldalra navigál minket, itt találhatjuk azokat a helyeket, amiket mi adtunk az alkalmazáshoz. Ha mi is hozzá szeretnénk adni egy helyet a programhoz, a mezők megfelelő kitöltése után a *"Hely hozzáadása"* gombra kattintva ezt megvalósíthatjuk.

32. ábra: Hely hozzáadása

A hely publikálása után megváltoztathatjuk, illetve törölhetjük az általunk

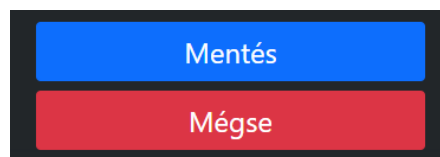


33. ábra: Egy általunk hozzáadott hely

hozzáadott helyek adatait. A "Szerkesztés" gombra kattintva módosíthatjuk a kiválasztott helyünk adatait,

ahol hozzáadtuk a helyünket, azokba a mezőkben betöltődnek a módosításra váró hely adatai. A hely adatainak javítása után, a "Mentés" gombra kattintva frissíthetjük helyünk adatait, ha meggondoltuk magunkat és mégsem akaruk a hely adatait módosítani a "Mégse" gombra kattintva kiléphetünk a hely

módosításából. A "Törlés" gombra kattintva a kiválasztott helyünket **törölhetjük**.



34. ábra: Helyeink módosítása

## 4 Project summary

Our project is about dog walking. We were thinking about creating something new to the world, what others have not seen yet. At the end we have finally created an idea that all the members of this team loved. This was the start of DogGo.

This idea is about an application where the users can add place markers on a map and let other users know why that place is good.

Fortunately it was easy to decide who will make which part of the project. Our cooperation is really powerful in my opinion. It went easily and fast to create the details of the project.

My part of the project is creating the web page. At the beginning of school year I was completely lost. I did not know what should I do. Fortunately I started to know why frameworks are important, thanks to my school, and I became able to start my part of the project. I have chosen a framework called Vue, because we were studying this framework in school.

Unfortunately I am not able to finish my project as i wanted because I do not have that much time left, so there are some opportunities for further development. One opportunity is for example the implementation of Google Maps.

It makes me really happy that my first bigger project was made with this group. I enjoyed all parts of this program with these people. I hope I will have the same kind of people at my workplace.

## 5 Ábrajegyzék

1. ábra DogGo ütemterv .....	4
2. ábra: Használati eset diagram.....	8
3. ábra: Adatbázis terv.....	9
4. ábra: Főoldal váza .....	25
5. ábra: Regisztrációs oldal váza .....	26
6. ábra: Bejelentkezés oldal váza .....	26
7. ábra: Regisztráció - Üresen kitöltött bemeneti mezők .....	39
8. ábra: Regisztráció - Felhasználó ikon .....	39
9. ábra: Bejelentkezés - Üresen kitöltött mezők .....	40
10. ábra: Bejelentkezés - Helytelen adatokkal kitöltött mezők.....	40
11. ábra: Hely hozzáadása - Üresen kitöltött mezők.....	41
12. ábra: Adataim módosítása - Üresen kitöltött mezők.....	42
13. ábra: Adataim módosítása - Helytelen adatokkal kitöltött mezők .....	42
14. ábra: Adataim módosítása – Létező felhasználó hibaüzenet .....	42
15. ábra: Hiba jelentése – Üresen kitöltött mezők .....	43
16. ábra: Hiba jelentése – Sikeresen publikált értékelés .....	43
17. ábra: Értékelés hozzáadása - Üresen kitöltött mezők.....	43
18. ábra: Navigációs sáv ikonjai .....	45
19. ábra: Felhasználó ikon.....	45
20. ábra: Megjelenített helyek.....	45
21. ábra: Főoldal felső része.....	45
22. ábra: Egy hely kinézete .....	46
23. ábra: Egy hely értékelései .....	46
24. ábra: Hiba jelentése .....	47
25. ábra: Sikeres hibaüzenet küldése .....	47
26. ábra: Az alkalmazás részletesebb leírása .....	47
27. ábra: Regisztrációs hibajelzés .....	48
28. ábra: Regisztrációs oldal kinézete.....	48
29. ábra: Hibás bejelentkezés.....	48
30. ábra: Adataunk módosítása oldal .....	49
31. ábra: Létező felhasználó hibaüzenet .....	49
32. ábra: Hely hozzáadása.....	49
33. ábra: Egy általunk hozzáadott hely .....	50
34. ábra: Helyeink módosítása .....	50