

MenhelyTérkép adatbázis:

Adatbázis típusa: NoSQL MongoDB

Milyen Collection-ok (táblák) lesznek:

Menhelyek: ezen a táblán tároljuk a menhelyek információit (pl: név, cím, telefonszám, és a regisztrációs adatai...)

Felhasználók: ebben a táblában tároljuk a sima (kereső) felhasználók adatait

Állat típusok listája: ez nekünk kell, hogy címkéket adhassunk hozzá a menhelyekhez (pl: Budaörsi állatmenhely: címke: kutya, macska)

Képek listája: ebben a táblában tároljuk a képek url-jét

Naptár: ebben a táblában tároljuk az eseményeket, amelyeket a menhelyek szerveznek



Adminok: Ez tárolja az admin felhasználókat akik a WPF asztali alkalmazásba betudnak lépni

Bővebben:


- **Menhelyek**
 - Id // menhelyek azonosító száma, (Mongo tölti ki)
 - Név // a menhely neve, (STRING, required, unique: true)
 - Cím // a menhely fizikai címe, (STRING, required)
 - Telefonszám // a menhely telefonszáma, (STRING, required)
 - E-mail cím // a menhely emailés elérhetősége, (STRING, required, unique: true)
 - Jelszó // jelszó a menhelynek, hogy betudjon lépni a fiókjába, (STRING, required)
 - Értékelés // értékelési érték, (NUMBER, required, default: 0)
 - Oldal link // a menhely valódi oldalának linkje, (STRING, required)
 - Leírás // a menhely saját leírása magáról, (STRING, required)
 - Cimkék // a hozzáadott cimkék id-a (ARRAY, default: [])
 - Jóváhagyott // csak a jóváhagyott menhelyek listázhatók ez amiatt fontos, (BOOLEAN, required, default: false)
- **Felhasználók**
 - Id // felhasználók azonosítója, (Mongo tölti ki)
 - Név // felhasználó teljes neve, (STRING, required)
 - E-mail // felhasználó email címe, (STRING, required, unique: true)
 - Telefonszám // felhasználó telefonszáma, (STRING)
 - Jelszó // felszó a fiókhoz, (STRING, required)
 - Kedvelt lista // felhasználó ide menti a kedvelt menhelyek id-at, (ARRAY, default: [])
- **Állat típusok listája**
 - Id // állatok azonosítója (NUMBER, mongo tölti ki)
 - Típus // pl: kutya, macska... (STRING, required)

- Képek listája
 - Id // képek azonosítója, (mongo automatikusan létrehozza)
 - Kép url //Helyileg hol a kép (STRING, required)
 - Menhely id //összekötni a menhellyel (NUMBER, required)
- Naptár
 - Id // esemény azonosítója (NUMBER, mongo automatikusan létrehozza)
 - Menhely id //menhelyek azonosítója (NUMBER, required)
 - Esemény neve //esemény neve ami megjelenik a naptárnál (STRING, required)
 - Dátum // esemény időpontja (DATE, required)
 - Leírás //rövid leírás a programról (STRING, required)
- Adminok
 - Id // esemény azonosítója (NUMBER, mongo automatikusan létrehozza)
 - felhasználonev //admin felhasználóneve (STRING, required)
 - jelszo //admin jelszava hashed(STRING, required)

MongoDB modellek:

Backend > models >  menhelyek-model.js >  menhelyekModel >  menhelycime

```
1  const mongoose = require('mongoose');
2
3  const menhelyekModel = mongoose.Schema({
4    menhelyneve:{
5      type: String,
6      default: "",
7      required: true
8    },
9    email:{
10     type: String,
11     default: "",
12     required: true
13   },
14   telefonszam:{
15     type: String,
16     default: "",
17     required: true
18   },
19   jelszo:{
20     type: String,
21     default: "",
22     required: true
23   },
24   menhelycime:{
25     type: String,
26     default: "",
27     required: true
28   },
29   oldallink:{
30     type: String,
31     default: "",
32     required: true
33   },
34   leiras:{
35     type: String,
36     default: "",
37     required: true
38   },
39   jovahagyva:{
40     type: Boolean,
41     default: false,
42     required: true
43   },
44 });
45
46 module.exports = mongoose.model('Menhelyek', menhelyekModel);
```

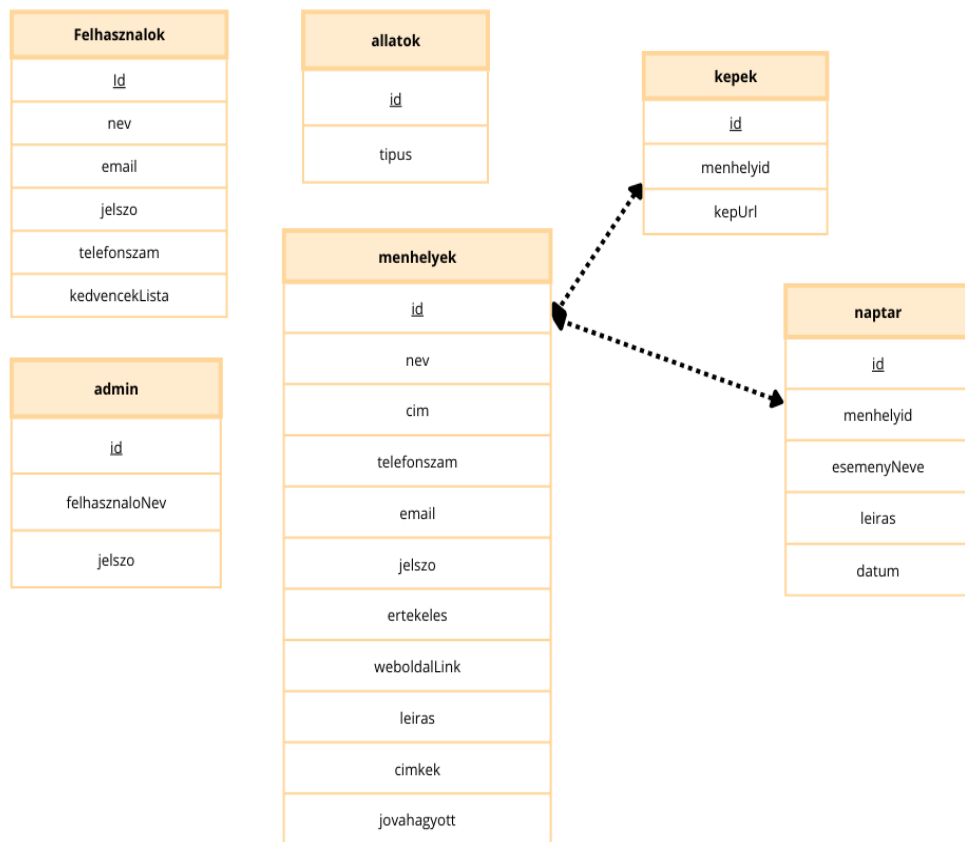
Backend > models >  felhasznalok-model.js > ...

```
1  const mongoose = require('mongoose');
2
3  const FelhasznalokScheme = mongoose.Schema({
4    nev:{
5      type: String,
6      default: "",
7      required: true
8    },
9    email:{
10     type: String,
11     default: "",
12     required: true
13   },
14   telefonszam:{
15     type: String,
16     default: "",
17     required: true
18   },
19   jelszo:{
20     type: String,
21     default: "",
22     required: true
23   }
24 });
25
26 module.exports = mongoose.model('Felhasznalok', FelhasznalokScheme);
```

Backend > models >  adminok-model.js > ...

```
1  const mongoose = require('mongoose');
2
3  const AdminokScheme = mongoose.Schema({
4    felhasznalonev:{
5      type: String,
6      default: '',
7      required: true
8    },
9    jelszo:{
10     type: String,
11     default: '',
12     required: true
13   }
14 });
15
16 module.exports = mongoose.model('Adminok',AdminokScheme);
```

EER diagramm:



Ek diagramm:

