# Frontend programozás - Kilátók

A következő feladatban a magyarországi kilátókat bemutató alkalmazást kell készítenie, ahol a látogatóknak lehetőségük van értékelni az egyes kilátókat. A webalkalmazást a kiadott leírás szerint kell elkészítenie, a felhasználói felületek kialakítását minták segítik, melyeket **nem kell** követnie a megoldásában.

A backend API szerver futtatásához a források mappa backend.zip állományát csomagolja ki a projektkönyvtárába, majd indítsa el a szervert a **Parancssor** (cmd.exe) ablakból a backend mappa aktuálissá tétele után a node backend.js parancs kiadásával. Az API szervert a http://localhost:3000 címen éri el. A **Parancsor** ablakot ne zárja be és ne jelöljön benne ki semmit a feladat megoldása során!

A http://localhost:3000/docs URL címen segítséget kap a webszerveren elérhető REST API függvényekkel kapcsolatban:

A képen szöveg, képernyőkép, szoftver, Weblap látható

Automatikusan generált leírás

Az elérhető REST API függvények:

* /api/locations

Metódus: GET

Válasz: (200-as státuszkóddal) Az adatbázisban található **hegységek** adatai

Válasz minta:

[

{

"id": 1,

"locationName": "Börzsöny"

},

{

"id": 2,

"locationName": "Bükk"

},

{

"id": 3,

"locationName": "Cserhát"

},

{

"id": 4,

"locationName": "Mátra"

},

{

"id": 5,

"locationName": "Visegrádi-hegység"

},

{

"id": 6,

"locationName": "Zempléni-hegység"

}

]

* /api/locations/{locationName}/viewpoints

Metódus: GET

Válasz: (200-as státuszkóddal) Az adatbázisban a {locationName} nevű hegységben található kilátók részletes adatai.

Válasz minta:

[

{

"viewpointName": "Avasi kilátó",

"mountain": "Avas",

"locationId": 2,

"height": 72.2,

"description": "Tévétoronyként és kilátóként egyaránt ...",

"built": "1963.05.04",

"imageUrl": " https://nits68.github.io/static/viewpoints/avas.jpg"

},

...

{

"viewpointName": "Petőfi-kilátó",

"mountain": "Bálvány",

"locationId": 2,

"height": 18.5,

"description": "A Bükk-vidék legmagasabban fekvő ...",

"built": "2016.06.22",

"imageUrl": "https://vizsga2023.cluster.jedlik.eu/images/petofi.jpg"

}

]

* /api/viewpoints

Metódus: GET

Válasz: (200-as státuszkóddal) Az adatbázisban található összes kilátó neve és a hegy (**nem hegység**) neve, ahol a kilátó található

Válasz minta:

[

{

"id": 8,

"viewpointName": "Avasi kilátó",

"mountain": "Avas"

},

...

{

"id": 4,

"viewpointName": "Várhegyi kilátó",

"mountain": "Várhegy"

},

{

"id": 14,

"viewpointName": "Vörös-kő kilátó",

"mountain": "Vörös-kő"

}

]

* /api/rate

Metódus: POST

A kérés törzsében kell a szerver felé elküldeni a kilátó értékelésének adatait.

Minta kérés body-ra:

{

"viewpointId": 15,

"rating": 8,

"email": "kiss.dora@mail.hu",

"comment": "Nagyon szép kilátás!"

}

Sikeres rögzítés esetén a válasz: (201-es státuszkóddal):

{

"count": 1,

"average": 8

}

Sikertelen rögzítés esetén (400-as státuszkóddal):

Hibaüzenet szövege

**Feladatok**

Az Ön által választott JavaScript keretrendszer (Angular, VueJs) valamelyikével készítse el az alábbi feladatokat! A feladatok elkészítését forrásállományok segítik, melyek a választott technológiának megfelelő alkönyvtárban találhatók a források mappában.

1. Készítse el az alkalmazás menüjeként szolgáló navigációs sávot az alábbi minta alapján:

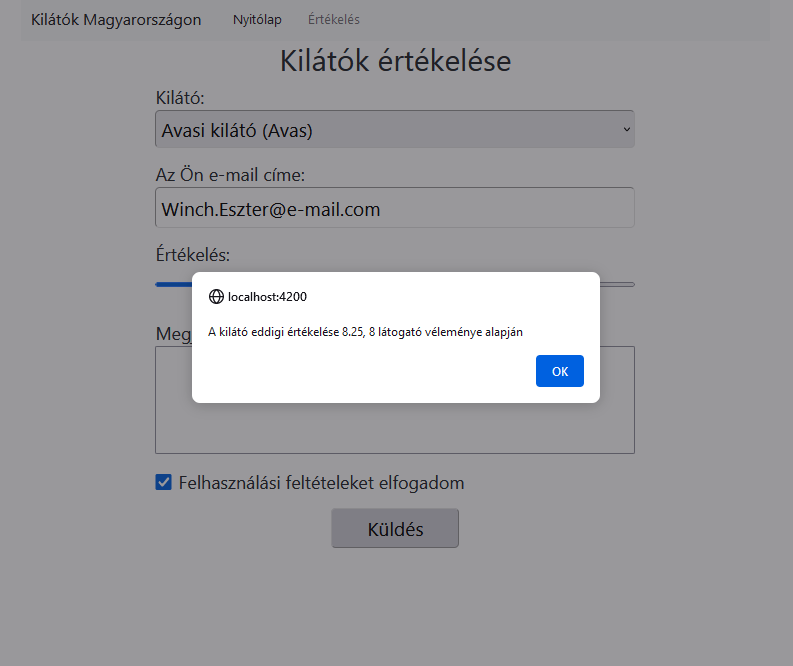


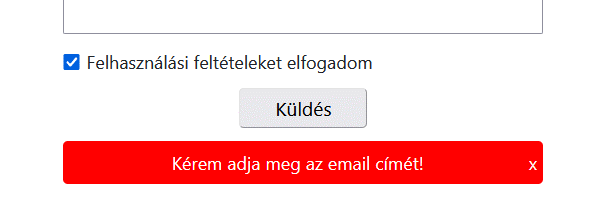
1. Hozzon létre egy oldalt, ami a webalkalmazás nyitó oldala lesz!
   1. Az oldal tetején lévő – a **hegységeket** tartalmazó – lenyíló lista lehetőségeinek feltöltéséhez használja a /api/locations URL-en található REST API függvényt!
   2. A kiválasztott hegységben lévő kilátók adatait a kiszolgáló szerver /api/locations/{locationName}/viewpoints URL-jén található REST API függvénnyel kérdezze le! (*Az URL-ben szereplő {locationName} helyére a kiválasztott hegység nevét illessze be!)*
   3. Az adatokat a kiadott mintához hasonlóan jelenítse meg. A megjelenítés során használhatja a források könyvtárban található állományokat.
   4. A kezdőoldalon a „Nyitólap” felirathoz állítsa be, hogy arra kattintva navigáljon a program erre az oldalra!

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

1. Hozzon létre egy újabb oldalt, amelyen a kilátók értékelését tudja elvégezni:
   1. Az oldal létrehozása során használhatja a források könyvtárban szereplő állományokat.
   2. Az összes kilátót tartalmazó lenyíló listához az elemeket a szerver /api/viewpoints URL-jéről töltse le!
   3. A „Küldés” gombra kattintva az űrlapon szereplő adatokat küldje el a szerver /api/rate URL-re az API leírás alapján!
   4. Sikeres küldés esetén jelenítse meg a szerver által visszaadott adatokat: *eddig hányan értékelték a kilátót, és jelenleg mennyi az átlagos értékelése*, majd navigáljon át a kilátók leírását tartalmazó oldalra.
   5. Sikertelen küldés esetén jelenítse meg a backend szervertől kapott hibaüzenetet!
   6. Állítsa be ennek az oldalnak az eléréséhez a /rating útvonalat!
   7. A kezdőoldalon az „Értékelés” felirathoz állítsa be, hogy arra kattintva navigáljon át a /rating útvonalra!





1. Forráskódját a választott technológiának megfelelően adja le:
   1. Tömörítés előtt törölje a felesleges állományokat és mappákat! **Ügyeljen arra, hogy feladatmegoldást tartalmazó mappát/állományt ne töröljön!**
   2. A forráskódját tömörítse be Vezetéknév\_Keresztnév\_frontend.zip néven!