**DogGo – Web**

Kilián Marcell András

**Bevezetés**

Mért választottam a weboldal elkészítését?

A webfejlesztés régóta érdekel, barátaim, ismerőseim megmutatták nekem, hogy mit is sikerült összehozniuk különböző keretrendszerek segítségével. Érdekelt, hogy hogyan kezdődik el egy ilyen projekt megvalósítása, mik a nehézségek egy ilyen oldal elkészítése során, hogyan zajlik a megvalósítás menete, milyen tervekhez igazodnak a tervezők.

Pár hónapja még azt sem tudtam mi az a keretrendszer, annyit tudtam róla, hogy hasznos egy webolal elkészítésében. Elképesztő élmény volt nézni ismerősöket, barátokat, ahogy csinálják a programjukat egy keretrendszer segítségével, kellemes élmény volt számomra, hogy nem értettem a kódjukat, mert még csak ennyire sem láttam, hogy mennyire széleskörű is a programozás. Emiatt szerettem volna egyszer én is érteni egy hasonló programot, amit keretrendszer segítségével hoztam létre.

Nekem ezelőtt nem volt alkalmam, ötletem és nagyon időm sem egy hasonló, nagyobb projekt elkészítéséhez. Közös megegyezés alapján én készítem el végül az alkalmazásunk webes részét. Szerencsémre most kipróbálhatom magam, hogy mennyire is tetszik ez az ága a programozásnak.

Terv/Weboldal kinézete

A weboldal tetején egy navigációs sávot képzeltünk el a többiekkel, a sáv bal oldalán a logónk, jobb oldalán sorba az egyéb opciók, például bejelentkezés, forráskód megtekintése, weboldal megosztása. Egy egyszerű sáv, viszont megkönnyebbíti a felhasználók számára a navigálást az alkalmazáson belül.

A főoldal törzse az alkalmazás bemutatkozásával, leírásával kezdődik. Az alkalmazás leírása után képeket terveztünk beilleszteni, hogy milyen is az alkalmazás, milyen közösségek jöhetnek létre az alkalmazás használatával, azonban átgondoltuk és eldöntöttük, annak érdekében, hogy a *vendég* felhasználók is részesülhessenek valamennyi élményben, amit az alkalmazás nyújt, az adatbázisban szereplő helyeket jelenítjük meg a leírás után.

A helyek megjelenítéséhez a *Google Maps* térképét akarjuk használni, ehhez létre kell hozni egy *Google Maps API kulcsot* a developers.google.com oldalon. Azért a Google térképét választottuk, mert Javascript segítségével könnyen lekérdezhető az aktuális helyünk, könnyű megjeleníteni különböző helyeket rajta és egyszerűen formázhatjuk azokat.

Az oldal aljára egy láblécet terveztünk beilleszteni, ahol megjelenítenénk az elérhetőségeinket, egyéb velünk kapcsolatos információkat, az alkalmazás logóját.

Megvalósítás

A tanév elején nagyon elveszettnek éreztem magam. Azt sem tudtam megmondani, hogy milyen eszközöket szeretnék használni a projekt megvalósításához. Igaz, elkezdtem videókat nézni, legtöbbet a *YouTube*-on, ahol emberek felépítenek egy teljes alkalmazást, lemásolnak alkalmazásokat, megcsinálják ugyan azt az alkalmazást, csak más-más keretrendszerekkel. Ezek után még inkább elveszettnek éreztem magam. Nem értettem, hogy miket és hogyan csinálnak azok az emberek, csak a végeredményt láttam, úgy éreztem, hogy hiába szánok időt ennek a megértésére, nem érek el vele semmit.

Az idei tanév alatt, számomra az első nagyobb lépés, amit a vizsgaremek elkészítéséhez tettünk, az a *Laravel* keretrendszer tanulásának elkezdése volt. Ezt a keretrendszert használjuk a projekt **backend** részének megvalósításához. Ahogy elkezdtük csinálni a projektünket, kicsit jobban megértettem, hogy mi is az a keretrendszer, hogy miért olyan hasznos, miért és hogyan is épül fel.

Újra elkezdtem videókat nézni különböző alkalmazások klónozásáról. A legtöbb fejlesztő a **React** Javascript keretrendszert használta ezekben a videókban. Mi is egy Javascript keretrendszert tanultunk eddig idén, a **Vue** Javascript keretrendszert.

Miért a Vue?

A **Vue** Javascript keretrendszer olyan *feature*-öket tartalmaz, nyújt számomra, amiket már használtam, ismerek. Ezek az előnyök például a többoldalas appok könnyű építése, a *routing*, *komponensek*, adatok *validálása*, számtalan hasznos csomag (*package*), *third party plugin* tartozik a keretrendszerhez (*Vuex, Vuelidate, Bootstrap Vue*). Ezek felett a *Laravel* keretrendszerrel könnyen lehet használni, szinkronizálva van a két keretrendszer.

Ezen felül ez az első frontend keretrendszer, amit idén tanítottak/tanítanak nekünk. Az első órán nem értettem semmit, nem tudtam, hogy mi is az a *kompones*, miért hasznosak, hogyan épül fel egy Vue alkalmazás, milyen parancsokat kell beírni a program tesztelésére, megjelenítésére. Ahogy az idő telik egyre több és több hasznos funkciót fedezek fel a keretrendszerben. Több ismerősöm is mondta, hogy egy könnyen értelmezhető keretrendszer a Vue, tanulásnak jó, ahhoz, hogy jobban megértsem a keretrendszerek felépítését, jó gyakorlás.

Tudtam, hogy ha elakadok akkor kérhetek segítséget, mert a környezetemben lévő emberek is használták/használják ezt a keretrendszert. Ezeknek a tényezőknek az összessége miatt választottam ezt a keretrendszert, ennek a vizsgaremeknek az elkészítésére

Fejlesztői dokumentáció

**Alkalmazott feljesztői 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. Microsoft Word
3. Microsoft PowerPoint
4. Microsot Paint

Komponensek:

Komponensek függőségei:

* "@fortawesome/fontawesome-svg-core": "^1.3.0"
* "@fortawesome/free-solid-svg-icons": "^6.0.0"
* "@fortawesome/vue-fontawesome": "^3.0.0-5"
* "@vuelidate/core": "^2.0.0-alpha.35"
* "@vuelidate/validators": "^2.0.0-alpha.27"
* "axios": "^0.26.0"
* "bootstrap": "^5.1.3"
* "vue": "^3.0.0"
* "vue-axios": "^3.4.1"
* "vue-browser-geolocation": "^1.8.0"
* "vue-router": "^4.0.14"
* "vuex": "^4.0.2"

Komponensek dokumentálása:

|  |  |
| --- | --- |
| Login komponens | |
| **Komponens leírása** | A komponens eltárolja a bejelentkezni próbáló felhasználó token-jét. |
| **Adattagok** | **Leírás** |
| 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. |
| **Metódusok** | **Leírás** |
| 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 | |
| **Komponens leírása** | A komponens új felhasználók létrehozását valósítja meg. |
| **Adattagok** | **Leírás** |
| 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. |
| **Metódusok** | **Leírás** |
| 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 | |
| **Komponens leírása** | A weboldal láblécét jeleníti meg a komponens. |
| **Adattagok** | **Leírás** |
| - | - |
| **Metódusok** | **Leírás** |
| - | - |

|  |  |
| --- | --- |
| Header komponens | |
| **Komponens leírása** | A weboldal fejlécét jeleníti meg a komponens. |
| **Adattagok** | **Leírás** |
| - | - |
| **Metódusok** | **Leírás** |
| signOut() | Eltávolítja a token adattagot, a user adattagnak üres értéket állít be, visszanavigál 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. |
| **Adattagok** | **Leírás** |
| 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.errorMsg.comment : String | A hibaüzenetet tárolja el. |
| **Metódusok** | **Leírás** |
| sendError() | A beviteli mezőbe írt hibaüzenetet próbálja meg elküldeni a szervernek. |

|  |  |
| --- | --- |
| Locations komponens | |
| **Komponens leírása** | Ez a komponens jeleníti meg a jóváhagyott helyeket. |
| **Adattagok** | **Leírás** |
| 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. |
| **Metódusok** | **Leírás** |
| loadData() | Ez a metódus kéri el a szervertől az engedélyezett helyek adatait és eltárolja a szervertől kapott helyeket. |
| checkLocationRating(id) : Integer | Átirányít minket a Ratings komponensre, az adott hely értékeléseit jeleníti meg. |
| deleteLocation(id) : Integer | Tiltja az id értékével rendelkező helyet. |
| editLocation(id) : Integer | Szerkeszti az id értékével rendelkező helyet. |
| saveLocation() | Elmenti a szerkesztett helyet. |
| cancelLocation() | A szerkesztésből kilép az eljárás. |
| resetForm() | A beviteli mezőket alaphelyzetbe állítja. |
| mounted() | Meghívja a loadData() eljárást, betölti az engedélyezett helyeket. |

|  |  |
| --- | --- |
| MyLocations komponens | |
| **Komponens leírása** | A saját helyeink kezelésére jött létre ez a komponens, a saját helyeinket tudjuk szerkeszteni, illetve helyeket hozzáadni az alkalmazáshoz. |
| **Adattagok** | **Leírás** |
| 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. |
| **Metódusok** | **Leírás** |
| 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 | |
| **Komponens leírása** | Egy hely értékeléseit jeleníti meg a komponens. |
| **Adattagok** | **Leírás** |
| 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 tűnteti 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. |
| **Metódusok** | **Leírás** |
| loadDatas() | Betölti a kiválasztott hely értékeléseit, az átlagos értékelését és adatait, emellett eltárolja ezeket az értékeket. |
| addRating() | A mezőkben lévő adatokkal egy új értékelést ad a helyhez. |
| editRating(id) : Integer | Betölti a módosítani kívánt értékelésünk adatait a beviteli mezőkbe. |
| changeRating() | A kiválasztott értékelésünk módosítása után ez az eljárás frissíti, módosítja a hozzászólásunkat. |
| 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() | Betölti az értékeléseket, ha nincs kiválasztott értékelés, visszanavigál minket a főoldalra. |

|  |  |
| --- | --- |
| Text komponens | |
| **Komponens leírása** | A projekt alapvető leírását tartalmazza a komponens. |
| **Adattagok** | **Leírás** |
| - | - |
| **Metódusok** | **Leírás** |
| - | - |

|  |  |
| --- | --- |
| Info komponens | |
| **Komponens leírása** | A projekt részletesebb leírását jeleníti meg ez a komponens. |
| **Adattagok** | **Leírás** |
| - | - |
| **Metódusok** | **Leírás** |
| - | - |

|  |  |
| --- | --- |
| UserData komponens | |
| **Komponens leírása** | A felhasználó nevét és email címét módosíthatjuk. |
| **Adattagok** | **Leírás** |
| 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. |
| **Metódusok** | **Leírás** |
| 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 | |
| **Komponens leírása** | A hibajelentés oldalt jelenti ez a komponens. |
| **Adattagok** | **Leírás** |
| - | - |
| **Metódusok** | **Leírás** |
| - | - |
| **Komponensek** | Error |

|  |  |
| --- | --- |
| HomePage komponens | |
| **Komponens leírása** | A főoldalt jelenti ez a komponens. |
| **Adattagok** | **Leírás** |
| - | - |
| **Metódusok** | **Leírás** |
| - | - |
| **Komponensek** | Text, Locations, Footer |

|  |  |
| --- | --- |
| InfoPage komponens | |
| **Komponens leírása** | Az információs oldalt jelenti ez a komponens. |
| **Adattagok** | **Leírás** |
| - | - |
| **Metódusok** | **Leírás** |
| - | - |
| **Komponensek** | Info, Footer |

|  |  |
| --- | --- |
| LoginPage komponens | |
| **Komponens leírása** | A bejelentkezés oldalt jelenti ez a komponens. |
| **Adattagok** | **Leírás** |
| - | - |
| **Metódusok** | **Leírás** |
| - | - |
| **Komponensek** | Login |

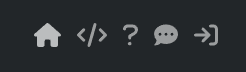
|  |  |
| --- | --- |
| MyLocationsPage **komponens** | |
| **Komponens leírása** | A saját helyek oldalt jelenti ez a komponens. |
| **Adattagok** | **Leírás** |
| - | - |
| **Metódusok** | **Leírás** |
| - | - |
| **Komponensek** | MyLocations |

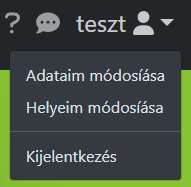
|  |  |
| --- | --- |
| RatingsPage komponens | |
| **Komponens leírása** | Az értékelés oldalt jelenti ez a komponens. |
| **Adattagok** | **Leírás** |
| - | - |
| **Metódusok** | **Leírás** |
| - | - |
| **Komponensek** | Ratings |

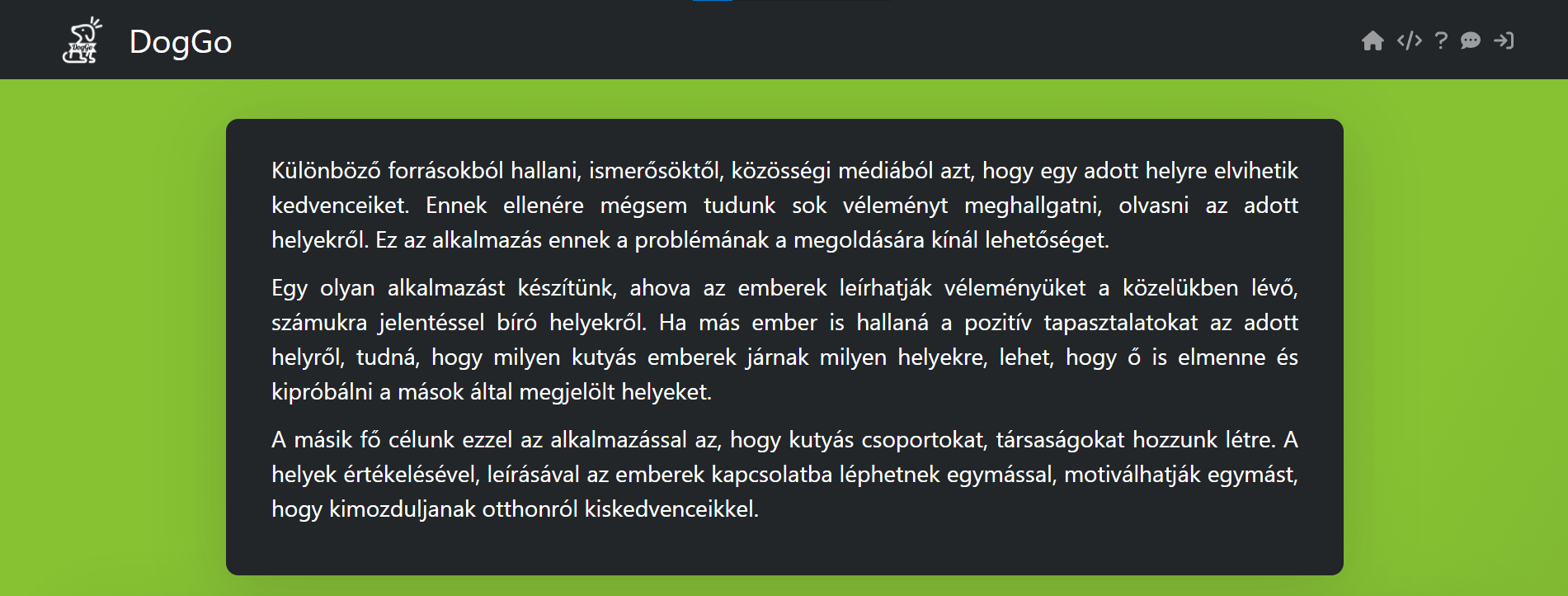
|  |  |
| --- | --- |
| RegisterPage komponens | |
| **Komponens leírása** | A regisztrálás oldalt jelenti ez a komponens. |
| **Adattagok** | **Leírás** |
| - | - |
| **Metódusok** | **Leírás** |
| - | - |
| **Komponensek** | Register |

|  |  |
| --- | --- |
| UserDataPage komponens | |
| **Komponens leírása** | A felhasználó adatainak oldalát jelenti ez a komponens. |
| **Adattagok** | **Leírás** |
| - | - |
| **Metódusok** | **Leírás** |
| - | - |
| **Komponensek** | Userdata |

Felhasználói dokumentáció

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

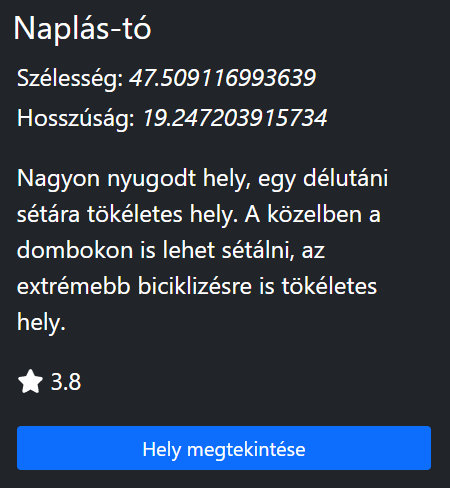
Az alkalmazás tetején a navigációs sávot láthatjuk. A különböző ikonokra kattintva navigálhatjuk magunkat az oldalak között. A navigációk sorba a **főoldalt**, a **forráskód** megtekintését, **információt** a weboldalról, a **hiba jelentést** és a **regisztrációt**, illetve **bejelentkezést** jelentik. Abban az esetben, ha bejelentkeztünk, a regisztráció ikon helyett a *saját felhasználónevünk* 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.

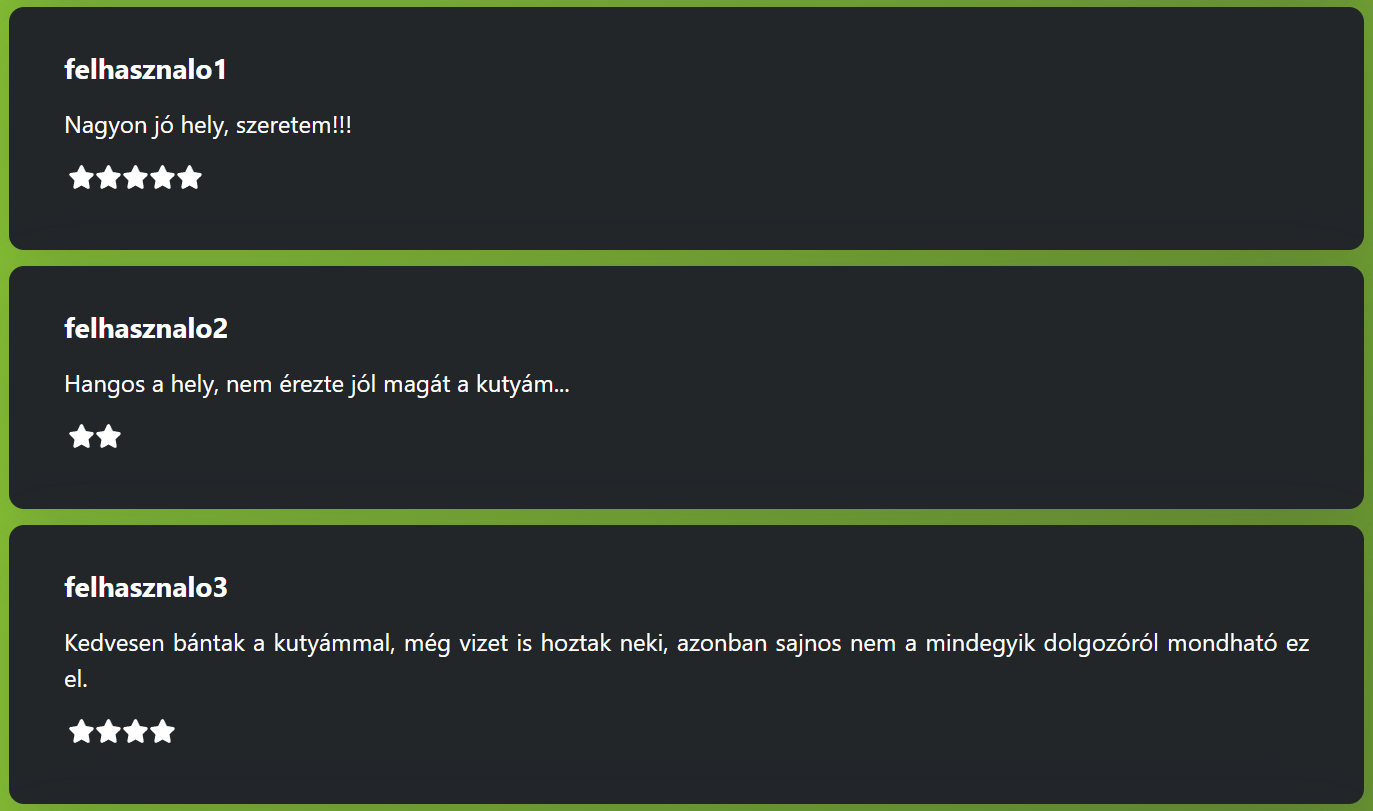
A bemutatkozás után az alkalmazás **helyei** találhatóak. A helyek tetején található a *legjobb értékeléseket* kapott hely. Ezt a helyet ajánlották leginkább a felhasználók, alatta található a többi hely.

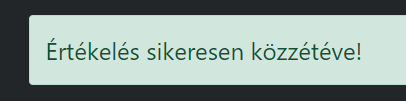
Mindegyik hely rendelkezik egy szélességgel és hosszúsággal, ez jelzi a helyek koordinátáit.

**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.

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.

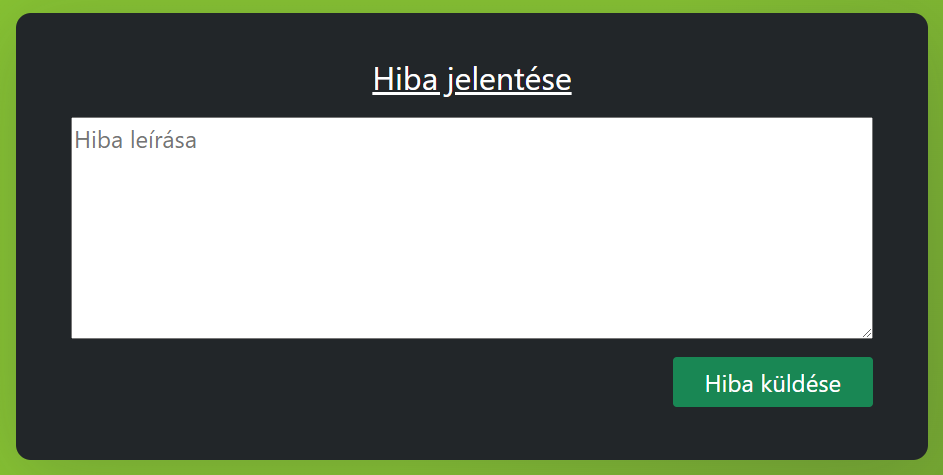


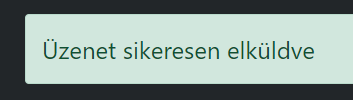
Egy értékelés tulajdonságai sorra a következők, a **felhasználónév**, azt jelenti, hogy melyik felhasználó értékelését láthatjuk, a **szöveges értékelés** és a **skálás értékelés**. A szöveges értékelés legfeljebb 255 karakter lehet, a skálás értékelés pedig egy egytől ötig tartó skálás értékelés.

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**

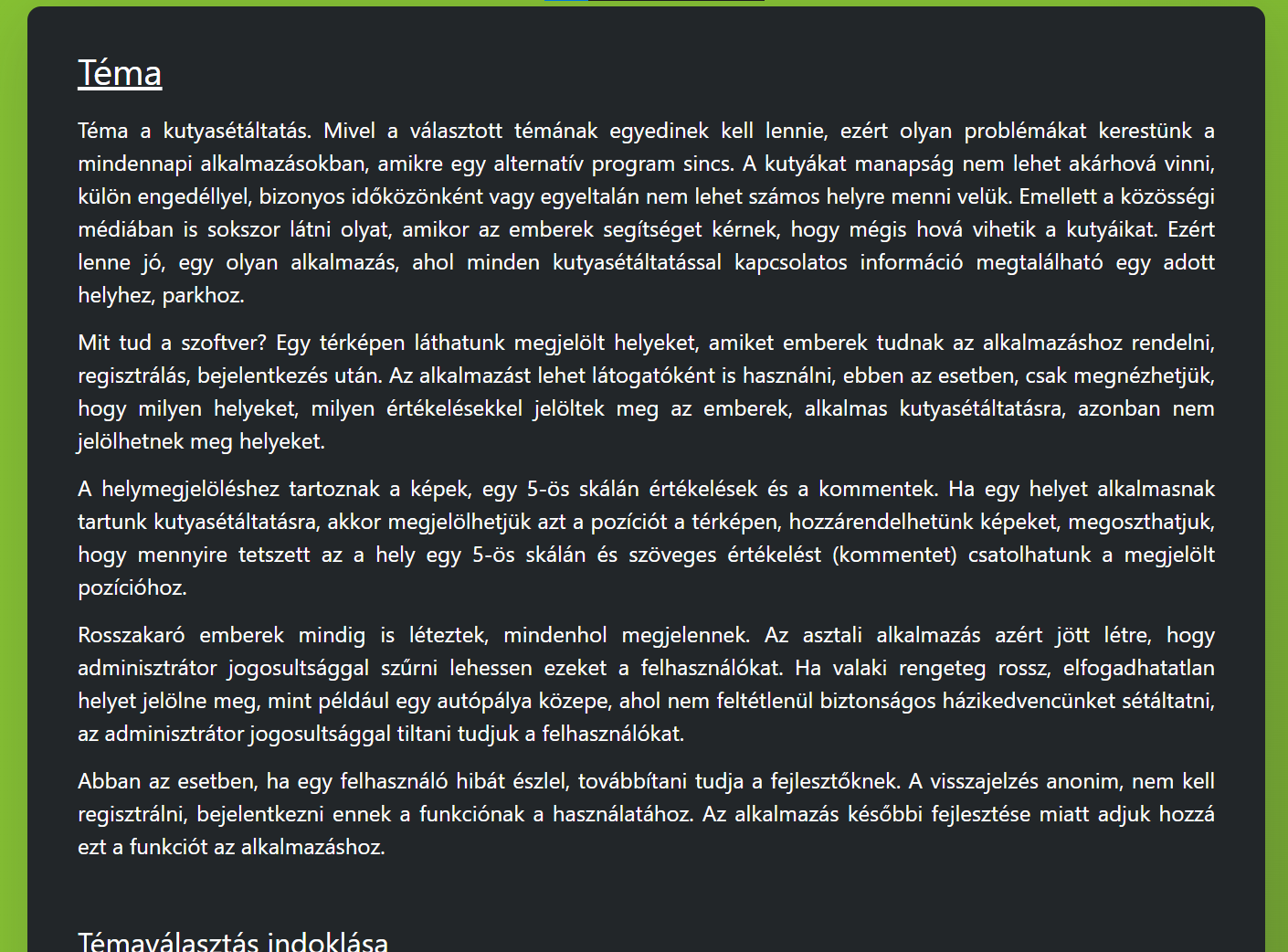
Abban az esetben ha valamilyen hibát észlel a felhasználó, lehetősége van ezt a hibát továbbítani a fejlesztőknek a *hibajelentés* fülön.



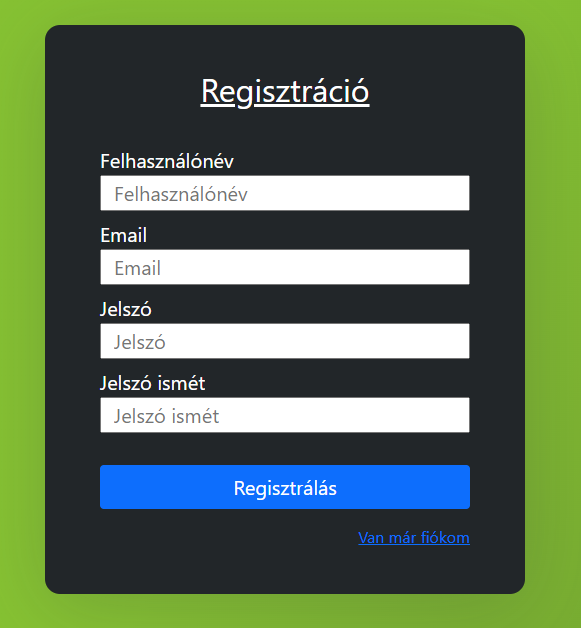
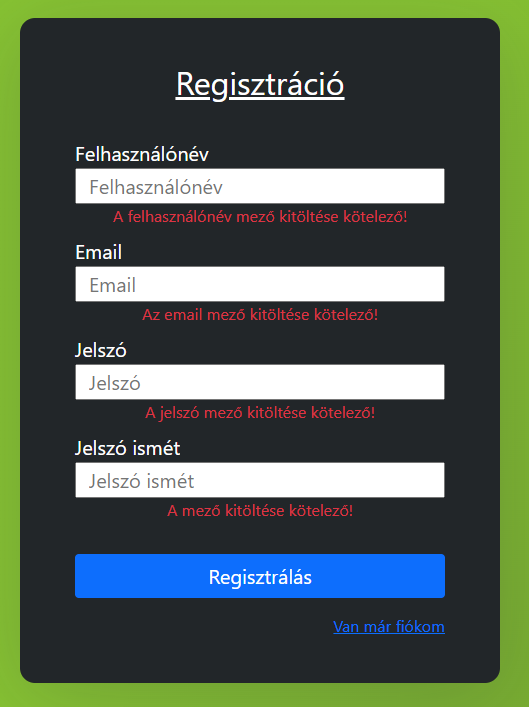
A mező helyes kitöltése után, a hiba elküldéséről szintén visszajelzést kapunk.

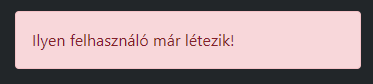
A hiba jelentéséhez nem kötelező a bejelentkezés.

**Információ**

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

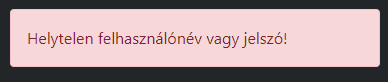
**Regisztráció**

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.

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.

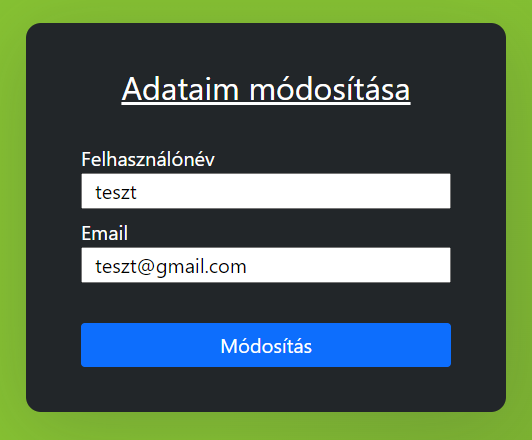
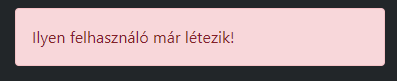
**Bejelentkezés**

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

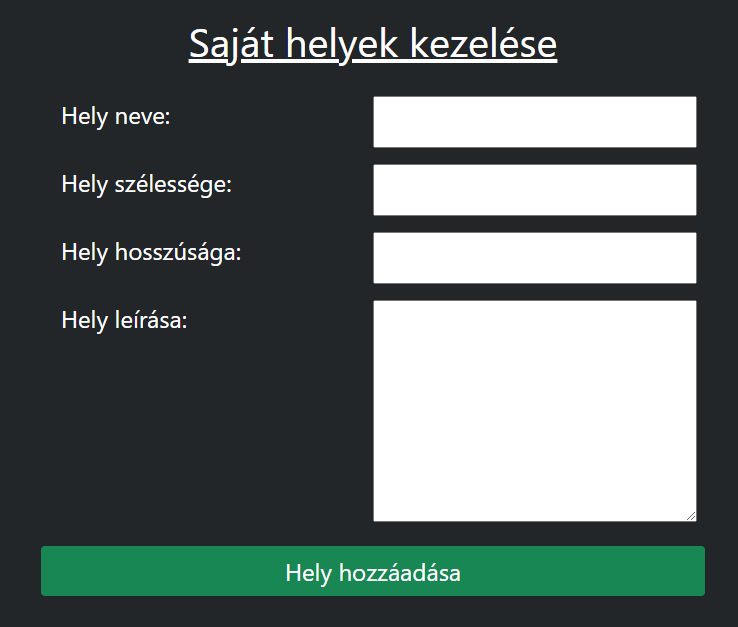
A mezők helyes kitöltése után a *”Bejelentkezés”* gombra kattintva bejelentkezünk, ha helyesen adtuk meg az adatainkat, ebben az esetben a navigációs sáv jobb oldalán a felhasználónevünket láthatjuk és értékeléseket és helyeket is hozzá tudunk adni az alkalmazáshoz.

Ha a bejelentkezés sikertelen, hibaüzenet jelenik meg. A beírt adatok ilyenkor nem vesznek el, könnyen lehet a beviteli mezőket módosítani.

**Felhasználó adatainak módosítása**

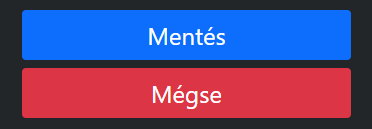
Abban az esetben ha módosítani kívánja adatait, a navigációs sávon a felhasználónevére kattintva, a legördülő menü *”Adataim módosítása”* menüpontjára kattintva a weboldal átnavigál minket az *adataim módosítása* oldalra. A **felhasználónév**, illetve az **email cím** módosítására van lehetőségünk. Az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

A hely publikálása után megváltoztathatjuk, illetve törölhetjük az általunk 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**.