**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

**Algoritmusok és Alkalmazásaik Tanszék**

**Online animátor jelentkezési rendszer fejlesztése webes környezetben**

Szerző: Témavezető:

Pipicz Petra Dr. Szabó László Ferenc

Programtervező informatikus BSc. Egyetemi docens, habilált doktor

**Budapest, 2025**

Tartalomjegyzék

[1. Bevezetés 1](#_Toc196506514)

[1.1. Motiváció 1](#_Toc196506515)

[1.2. Az alkalmazás 1](#_Toc196506516)

[2. Felhasználói dokumentáció 2](#_Toc196506517)

[2.1. Az alkalmazás rövid ismertetése 2](#_Toc196506518)

[2.2. Az alkalmazás használata 2](#_Toc196506519)

[2.2.1. Felhasználói hozzáférés 2](#_Toc196506520)

[2.2.2. Bejelentkezés 3](#_Toc196506521)

[2.2.3. Regisztráció 4](#_Toc196506522)

[2.2.4. Kezdőlap 6](#_Toc196506523)

[2.2.4.1. Események 6](#_Toc196506524)

[2.2.4.2. Jelentkezés eseményre 7](#_Toc196506525)

[2.2.4.3. Esemény törlése 7](#_Toc196506526)

[2.2.4.4. Részletek megtekintése 7](#_Toc196506527)

[2.2.4.5. Jelentkezett események 8](#_Toc196506528)

[2.2.5. Profil 10](#_Toc196506529)

[2.2.5.1. Profil adatok módosítása 11](#_Toc196506530)

[2.2.5.2. Jelszó megváltoztatása 11](#_Toc196506531)

[2.2.6. Felhasználók 12](#_Toc196506532)

[2.2.7. Új esemény 13](#_Toc196506533)

[3. Fejlesztői dokumentáció 15](#_Toc196506534)

[3.1. Specifikáció 15](#_Toc196506535)

[3.1.1. Feladatleírás 15](#_Toc196506536)

[3.1.2. Működési logika 16](#_Toc196506537)

[3.1.3. Fejlesztői környezetek 16](#_Toc196506538)

[3.1.4. Felhasználói felület specifikáció 17](#_Toc196506539)

[3.2. Architekrúra 17](#_Toc196506540)

[3.2.1. Frontend 17](#_Toc196506541)

[3.2.2. Backend 18](#_Toc196506542)

[3.3. Szoftverek telepítése 18](#_Toc196506543)

[3.3.1. Szoftver- és hardverkörnyezet 19](#_Toc196506544)

[3.3.2. Telepítés 19](#_Toc196506545)

[3.4. Alkalmazás felépítése 20](#_Toc196506546)

[3.4.1. Felhasználói esetek 20](#_Toc196506547)

[3.4.2. Felhasználói történetek 22](#_Toc196506548)

[3.4.3. Adatbázis 29](#_Toc196506549)

[3.4.3.1. Az adatbázis felépítése 29](#_Toc196506550)

[3.4.4. Backend 31](#_Toc196506551)

[3.4.4.1. Végpontok 32](#_Toc196506552)

[3.4.4.2. Middlewares 40](#_Toc196506553)

[3.4.4.3. Utils 41](#_Toc196506554)

[3.4.4.4. Database 41](#_Toc196506555)

[3.4.5. Frontend 41](#_Toc196506556)

[3.4.5.1. Komponensek 41](#_Toc196506557)

[3.4.5.2. Auth 43](#_Toc196506558)

[3.4.5.3. Edit 43](#_Toc196506559)

[3.4.5.4. Event-details 44](#_Toc196506560)

[3.4.5.5. Forget-password 45](#_Toc196506561)

[3.4.5.6. Header 46](#_Toc196506562)

[3.4.5.7. Home 47](#_Toc196506563)

[3.4.5.8. New-date 48](#_Toc196506564)

[3.4.5.9. Password-reset 49](#_Toc196506565)

[3.4.5.10. Profile 50](#_Toc196506566)

[3.4.5.11. Shared 50](#_Toc196506567)

[3.4.5.12. Users 50](#_Toc196506568)

[3.5. Tesztelési terv 51](#_Toc196506569)

[4. Összefoglalás 54](#_Toc196506570)

[4.1. Konklúzió 54](#_Toc196506571)

[4.2. További fejlesztési lehetőségek 54](#_Toc196506572)

[5. Irodalomjegyzék 56](#_Toc196506573)

[6. Ábra-, és táblázatjegyzék 57](#_Toc196506574)

[6.1. Ábrajegyzék 57](#_Toc196506575)

[6.2. Táblázatjegyzék 58](#_Toc196506576)

# Bevezetés

Az animátorok az ELTE Informatikai Karának önkéntes hallgatói közösségét alkotják, akik elsősorban a kari rendezvények szervezéséért, lebonyolításáért és felügyeletéért felelősek. Tevékenységük kiterjed többek között a Társas- és Szerepjátékos Hétvégére (TSzH), amely során a társasjátékozás kerül a középpontba. Kiemelt szerepet vállalnak továbbá a Gólyatábor és a gólyahéten megrendezett programok lebonyolításában is. Ez a két esemény különösen fontos, hiszen céljuk, hogy az elsőéves hallgatók minél gyorsabban beilleszkedhessenek az egyetemi közösség életébe.

## Motiváció

A témaválasztásom mögött személyes és szakmai motiváció egyaránt áll. Egyetemi éveim kezdete óta aktívan részt veszek az animátori közösség munkájában, így testközelből tapasztaltam meg a jelenlegi jelentkezési rendszer hiányosságait. Úgy éreztem, elérkezett az idő egy modernebb, felhasználóbarátabb megoldás kidolgozására, amely egyszerűsíti és átláthatóbbá teszi az animátorok eseményekre való jelentkezését.

Szakmai szempontból a webfejlesztés világa mindig is közel állt hozzám. Egy gyorsan fejlődő digitális világban a webes alkalmazások egyre meghatározóbb szerepet töltenek be mindennapi életünkben. A vállalkozások, szolgáltatók és közösségek számára is elengedhetetlenné vált az online jelenlét, amelyhez jól működő, könnyen használható webes felületekre van szükség. Ez is hozzájárult ahhoz, hogy szakdolgozatom témájaként egy dinamikus webalkalmazás fejlesztése mellett döntöttem, melynek elkészítése során értékes tapasztalatokat szereztem a felhasznált technológiák alkalmazásában.

## Az alkalmazás

A cél egy olyan alkalmazás létrehozása volt, amely egyszerűbbé és gyorsabbá teszi az animátorok számára az eseményekre történő jelentkezést, valamint megkönnyíti az animátorkoordinátorok számára a jelentkezett önkéntesek kezelését. A megvalósításhoz a Node.js-ben elérhető könyvtárakat használtam, köztük az Angular és az Express keretrendszereket. Ezek választását az is indokolta, hogy jelenleg is ezekkel a technológiákkal dolgozom a munkám során, így a fejlesztés során szerzett tapasztalatok közvetlenül hozzájárultak a meglévő tudásom bővítéséhez.

# Felhasználói dokumentáció

Ebben a fejezetben az alkalmazás rövid ismertetése után, részletezem a hozzáférést a felhasználók számára. Ezt követően bemutatásra kerül képernyőképekkel szemléltetve az alkalmazás használata lépésről lépésre, eközben pedig részletezem a különböző funkciókat. Emellett felhívom a figyelmet az esetleges hibákra is, amelyek megjelenhetnek az alkalmazás használata közben.

## Az alkalmazás rövid ismertetése

Az alkalmazás alapvetően egy időpontfoglaló webes applikáció, amely lehetővé teszi az animátorok és koordinátorok számára az események hatékony kezelését. Az animátorok bejelentkezés után jelentkezhetnek a még elérhető eseményekre, illetve legfeljebb 24 órával a kezdési időpont előtt le is mondhatják azokat. A felhasználók számára továbbá lehetőség nyílik a saját eseményeik Google Naptárba történő exportálására, amennyiben csatlakoztatták Google-fiókjukat az alkalmazáshoz.

Általános felhasználóként megtekinthetők és szerkeszthetők a saját profiladatok. Koordinátorok és adminisztrátorok számára elérhető az események részletes megtekintése, a regisztrált felhasználók listája, valamint új események létrehozása. Az adminisztrátorok ezen felül jogosultak események törlésére, illetve felhasználói jogosultságok módosítására is.

## Az alkalmazás használata

### Felhasználói hozzáférés

Jelen állapotában az alkalmazás nem elérhető publikus webszerverről, ugyanis a szakdolgozat keretében csak lokális számítógépről futtatható.

Az alkalmazás eléréséhez a következő teendőket kell végrehajtani:

* A felhasználó nyisson meg egy lokális gépen található böngészőt.
* Az alkalmazás elérési útvonala: <http://localhost:4200/>

A telepítési útmutatóról a későbbiekben esik szó a Szoftverek telepítése fejezetben.

### Bejelentkezés

A program elindítását követően, ha a felhasználó nincs bejelentkezve akkor a ’Bejelentkezés’ felületre jut.

A képen Betűtípus, szöveg, Grafika, embléma látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A megfelelő email cím, jelszó megadása után a ’Bejelentkezés’ funkciógombra kattintva véglegesíthető. Amíg a mezők formailag nem megfelelően vannak megadva, addig a gomb nem elérhető.

Helytelen email és/vagy jelszó megadása esetén a ’**Hibás felhasználónév vagy jelszó!**’ hibaüzenet jelenik meg. Helytelen formátumú email cím megadása esetén a ’**Az email formátuma nem megfelelő**.’ hibaüzenet jelenik meg. Helytelen formátumú jelszó megadása esetén a ’**A jelszónak tartalmaznia kell legalább egy nagybetűt, egy számot és legalább 8 karakternek kell lennie.**’ hibaüzenet jelenik meg.

A képen szöveg, képernyőkép, Betűtípus, embléma látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

1. ábra - Bejelentkezés

**Elfelejtett jelszó**

Ha a felhasználó elfelejtette jelszavát, akkor az ’Elfelejtett jelszó’ funkciógombra kattintva kezdeményezheti. Az új jelszó kérését követően megjelenik az ’Új jelszó igénylése’ felület:

A képen szöveg, képernyőkép, Betűtípus, Acélkék látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

2. ábra - Új jelszó igénylése

A ’Mégsem’ funkciógombra kattintva a rendszer visszanavigálja a felhasználót a ’Bejelentkezés’ felületre.

Formailag megfelelő és létező email cím megadása után a ’Küldés’ funkciógombra kattintva indítható el a folyamat. Küldés után a felhasználó kap egy emailt, amely tartalmazza az új jelszavát.

* A kiküldött email tárgya: ’Password Reset’ (Jelszó visszaállítás)
* A kiküldött email szövege: ’Your new password is: <jelszó>’ (Az új jelszavad: <jelszó>)

**Sikeres bejelentkezés:**

30 perc után a rendszer automatikusan kilépteti a felhasználót, és ismét a ’Bejelentkezés’ felületre navigál.

Sikeres bejelentkezést követően a fejlécen megjelenik a bejelentkezett felhasználó neve, és mellette a ’Kijelentkezés’ funkciógomb.

### Regisztráció

Ha a felhasználónak még nincsen fiókja akkor a ’Bejelentkezés’ felületről tud átnavigálni a ’Regisztráció’ felületre. Ezt két helyen is megteheti:

A képen szöveg, képernyőkép, Betűtípus, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

3. ábra - Regisztráció gombok

Átirányítás után megjelenik egy regisztrációs űrlap, melynek helyes kitöltésével a felhasználó regisztrálni tud az applikációba.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

4. ábra - Regisztráció

A mezők kitöltése után a ’Regisztráció’ funkciógombra kattintva véglegesíthető az űrlap. Véglegesítés után a rendszer a ’Bejelentkezés’ felületre navigál.

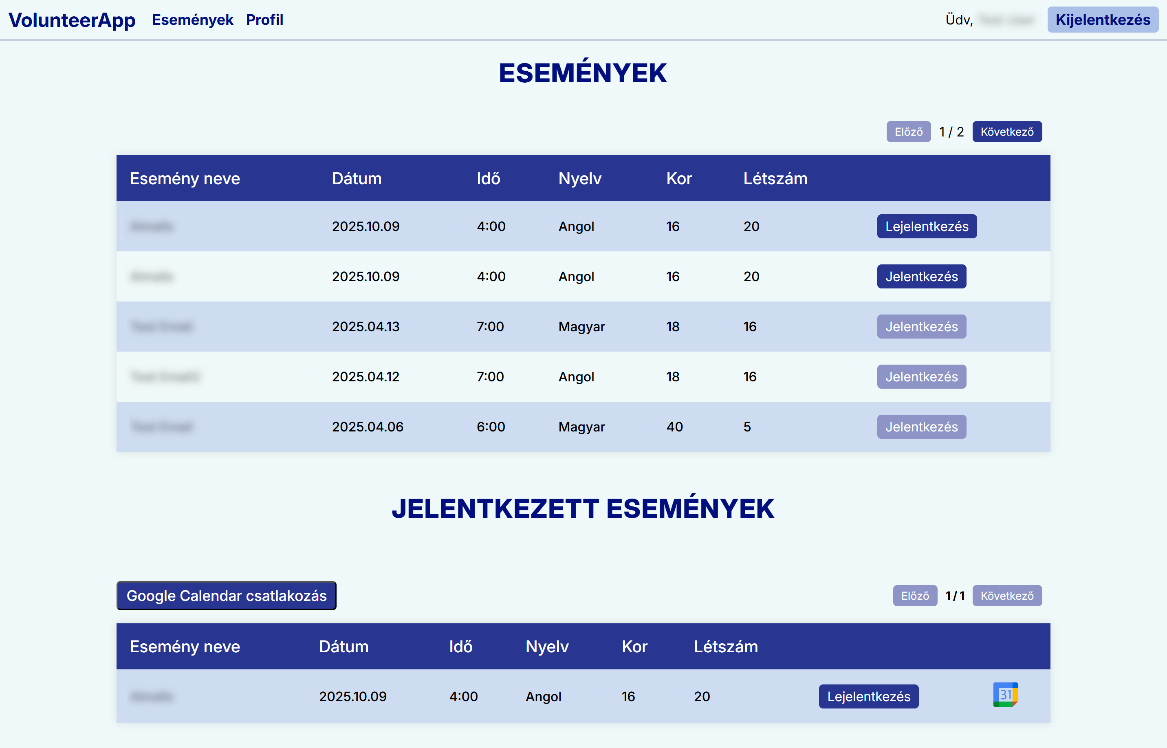
Amíg a megadott adatok nem megfelelőek formailag addig a ’Regisztráció’ gomb inaktív.

1. táblázat - Regisztrációs űrlap mezőinek validációja

|  |  |
| --- | --- |
| Név | * Mező kitöltése kötelező * Min. 4, max. 50 karakter hosszú   Megjelenő hibaüzenetek:   * **Mező kitöltése kötelező.** * **A névnek 4 és 50 karakter között kell lennie.** |
| Telefonszám | * Mező kitöltése kötelező * Formátuma: +36301234567 vagy 06301234567   Megjelenő hibaüzenetek:   * **Mező kitöltése kötelező.** * **A telefonszám érvénytelen! Pl.: +36201234567 vagy 061234567.** |
| Email cím | * Mező kitöltése kötelező * Formátuma: [teszt@teszt.hu](mailto:teszt@teszt.hu)   Megjelenő hibaüzenetek:   * **Mező kitöltése kötelező.** * **Az email formátuma nem megfelelő.** |
| Jelszó | * Mező kitöltése kötelező * Min 8 karakter, legalább egy nagybetű és egy szám   Megjelenő hibaüzenetek:   * **Mező kitöltése kötelező.** * **A jelszónak tartalmaznia kell legalább egy nagybetűt, egy számot és legalább 8 karakternek kell lennie.** |
| Jelszó megerősítése | * Mező kitöltése kötelező * Meg kell egyeznie a jelszóval   Megjelenő hibaüzenetek:   * **Mező kitöltése kötelező.** * **A két jelszó nem egyezik.** |

### Kezdőlap

Bejelentkezés után a felhasználót a rendszer a kezdőlapra navigálja, ahol megtekinthetőek az animátorok számára elérhető események, illetve a felhasználó saját eseményei is. Ezt a felületet minden bejelentkezett felhasználó eléri.



5. ábra - Kezdőoldal

#### Események

Az ’Események’ táblázatban láthatóak az egyes események adatai, amelyekre a felhasználók jelentkezhetnek dátum szerint csökkenő sorrendben.

A jobb felső sarokban egy lapozó található. A ’Következő’ funkciógombbal előre léphet a felhasználó az események között. Az ’Előző’ funkciógombbal visszaléphet az események között. A két gomb között látható a jelenlegi ’oldal / összes oldal.

A táblázatban a következő oszlopok találhatóak:

* **Esemény neve**
* **Dátum**: Az esemény dátuma
* **Idő**: Az esemény kezdő időpontja
* **Nyelv**: Milyen nyelvű csoport érkezik
* **Kor**: Az érkező csoport korosztálya
* **Létszám**: Az érkező csoport létszáma
* **Jelentkezés/Lejelentkezés** funkciógomb
* A képen kör, szimbólum látható

  Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen. funkciógomb:
  + kattintásra átirányít a jelentkezett animátorok listájára
  + csak koordinátor és admin láthatja
* A képen szimbólum, embléma, piros, kör látható

  Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen. funkciógomb:
  + esemény törlése
  + csak admin láthatja

#### Jelentkezés eseményre

A ’Jelentkezés’ funkciógombra kattintva a felhasználó sikeresen jelentkezik az eseményre amire szeretne. A fel- és lejelentkezésről mind az animátor, mind pedig a koordinátorok értesülnek emailben. Az email tartalmazza az esemény nevét és idejét, illetve az animátor nevét is.

* A kiküldött email tárgya: ’Új jelentkezés eseményre’ / ’Lejelentkezés eseményről’
* A kiküldött email szövege:

’Új jelentkezés érkezett az eseményre:’ / ’Lejelentkezés történt eseményről: "<esemény neve>" (2025.10.09 4:00)’

’Jelentkező neve: <felhasználó neve>’

Ha az esemény időpontja előtt 24 órával vagyunk nem lehet lemondani azt, így a ’Jelentkezés/Lejelentkezés’ funkciógomb inaktívvá válik. Ha elmúlt az esemény időpontja a gomb szintén nem elérhető.

Az animátor az esemény előtt 12 órával kap egy emlékeztető emailt a rendezvény nevével és idejével.

* A kiküldött email tárgya: ’Esemény jelentkezés megerősítése’
* A kiküldött email szövege: ’Sikeresen jelentkeztél a(z) "<esemény neve>" eseményre, amely <esemény dátuma és ideje> kerül megrendezésre.’

#### Esemény törlése

Csak adminisztrátori jogosultsággal rendelkező felhasználó éri el a törlés funkciógombot. Kattintással lehetősége van kitörölni az adott eseményt, ekkor a táblázatból automatikusan eltűnik a kitörölt esemény.

#### Részletek megtekintése

Csak koordinátor és admin jogosultsággal rendelkező felhasználó éri el a A képen kör, szimbólum látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen. funkciógombot. Kattintásra átnavigálja a felhasználót a ’Jelentkezett felhasználók’ felületre.

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

6. ábra - Jelentkezett felhasználók

A felületen megjelennek a felhasználók nevei, email címük, illetve telefonszámuk.

A ’← VISSZA A FŐOLDALRA’ funkciógombra kattintva a rendszer visszanavigálja a felhasználót a kezdőoldalra.

Ha még senki nem jelentkezett az adott eseményre az alábbi felület látható:

A képen szöveg, képernyőkép, Betűtípus, embléma látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

7. ábra - Nem található jelentkezett felhasználó

#### Jelentkezett események

Jelentkezés után az adott esemény megjelenik a ’Jelentkezett események’ táblázatban.

Ha nincsen a felhasználónak saját eseménye a következő felület jelenik meg:

A képen szöveg, képernyőkép, Betűtípus, embléma látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

8. ábra - Nem található jelentkezett esemény

Ha vannak eseményeink, akkor egy táblázat jelenik meg a következő mezőkkel:

* **Esemény neve**
* **Dátum**: Az esemény dátuma
* **Idő**: Az esemény kezdő időpontja
* **Nyelv**: Milyen nyelvű csoport érkezik
* **Kor**: Az érkező csoport korosztálya
* **Létszám**: Az érkező csoport létszáma
* **Lejelentkezés** funkciógomb
* A képen szimbólum, embléma, Betűtípus, Grafika látható

  Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen. funkciógomb: exportálás Google Calendar-ba

A táblázat felett a jobb oldalon egy lapozó található. A ’Következő’ funkciógombbal előre léphet a felhasználó az események között. Az ’Előző’ funkciógombbal visszaléphet az események között. A két gomb között látható a jelenlegi oldal/összes oldal.

A bal oldalon a ’Google Calendar csatlakozás’ funkciógomb található. Kattintásra a szerver átnavigál a Google saját bejelentkezési felületére. Sikeres bejelentkezés után a rendszer visszanavigál a kezdőoldalra.

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

9. ábra - Csatlakozás Google Calendar-hoz

Sikeres csatlakozás után, a felhasználónak lehetősége van a saját eseményeit kiexportálni Google Naptárba a sor végén található naptár ikonra kattintva.



10. ábra - Exportálás Google Naptárba

### Profil

A bejelentkezett felhasználó ezen a felületen megtekintheti és módosíthatja saját adatait.

A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

11. ábra – Profil

A ’← VISSZA A FŐOLDALRA’ funkciógombra kattintva a rendszer visszanavigálja a felhasználót a kezdőoldalra.

#### Profil adatok módosítása

A ’Módosítás’ funkciógombra kattintva, a rendszer átnavigál az ’Adatok módosítása felületre’. Ezen az oldalon a felhasználó módosíthatja a nevét, telefonszámát és az email címét.

A ’← VISSZA A PROFIL OLDARA’ funkciógombra kattintva a rendszer visszanavigálja a felhasználót a ’Profil’ oldalra.

A ’Mégsem’ funkciógombra kattintva a felhasználó visszalép módosítás nélkül a ’Profil’ oldalra.

A ’Mentés’ funkciógombra kattintva a rendszer elmenti a módosított adatokat, majd üzenetet küld a sikerességről: ’Profil sikeresen módosítva!’. Mentés után a rendszer visszanavigál a ’Profil’ oldalra.



12. ábra - Adatok módosítása

#### Jelszó megváltoztatása

A ’Jelszó megváltoztatása’ funkciógombra kattintva a rendszer átirányít a ’Jelszó megváltoztatása felületre.

A ’Mégsem’ funkciógombra kattintva a felhasználó visszalép módosítás nélkül a ’Profil’ oldalra.

A jelszó megváltoztatásához szükséges megadni a felhasználó jelenlegi jelszavát, illetve az új jelszót. Az új jelszó nem egyezhet meg a régivel. Az új jelszónak tartalmaznia kell legalább egy nagy betűt és egy számot, illetve legalább 8 karakter hosszúnak kell lennie.

A ’Módosítás’ funkciógombra kattintva a módosítások elmentődnek, majd a rendszer visszanavigál a ’Profil’ oldalra. Ha úgy kattint a gombra, hogy nincsenek helyesen kitöltve a mezők a ’Minden mező kitöltése kötelező!’ hibaüzenetet kapja a felhasználó.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

13. ábra - Jelszó megváltoztatása

### Felhasználók

A ’Felhasználók’ felület csak koordinátor és adminisztrátor jogosultsággal rendelkező felhasználók számára érhető el. A felületen látható az összes eddigi regisztrált felhasználó.



14. ábra - Felhasználók

Az adminisztrátor módosíthatja az egyes felhasználók jogosultságait: Admin, Koordinátor, Animátor. Ha nincsen semmi bejelölve, akkor Animátor az illető. Az Admin és a Koordinátori szerepkör attól függ, hogy melyik oszlop van bejelölve. Ez úgy módosítható, ha először kiveszi a felhasználó a jelölést, majd átkattint a másik jelölőnégyzetbe. Ezzel meg is változott a felhasználó szerepköre.

### Új esemény

Az ’Új esemény’ felület csak koordinátor és adminisztrátori jogosultsággal rendelkező felhasználók számára érhető el. A felületen új eseményeket lehet létrehozni.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

15. ábra - Új esemény hozzáadása

A ’← VISSZA A FŐOLDALRA’ funkciógombra kattintva a rendszer visszanavigálja a felhasználót a kezdőoldalra.

A mezők helyes kitöltése után a ’Hozzáadás’ funkciógombra kattintva véglegesíthető az űrlap. Véglegesítés után a rendszer a kezdőlapra navigál.

2. táblázat - Új esemény hozzáadása űrlap mezőinek validációja

|  |  |
| --- | --- |
| Esemény neve | * Mező kitöltése kötelező * Min 5, max 50 karakter hosszú lehet   Megjelenő hibaüzenetek:   * **Mező kitöltése kötelező.** * **A név 5 és 50 karakter között lehet.** |
| Esemény ideje | * Mező kitöltése kötelező * Dátum formátum: ÉÉÉÉ.HH.NN   Megjelenő hibaüzenetek:   * **Mező kitöltése kötelező.** |
| Idő | * Mező kitöltése kötelező * Szám formátum: min 1, max 24   Megjelenő hibaüzenetek:   * **Mező kitöltése kötelező.** * **Az óra 1 és 24 között kell, hogy legyen.** |
| Létszám | * Mező kitöltése kötelező * Szám formátum: min 1   Megjelenő hibaüzenetek:   * **Mező kitöltése kötelező.** * **A létszám legalább 1 fő kell legyen.** |
| Korosztály | * Mező kitöltése kötelező * Szám formátum: min 1   Megjelenő hibaüzenetek:   * **Mező kitöltése kötelező.** * **A korosztály legalább 1 éves kell legyen.** |
| Magyar | * Kitöltése opcionális * Üresen: Angol nyelvű csoport * Bejelölve: Magyar nyelvű csoport |

# Fejlesztői dokumentáció

Ebben a fejezetben a fejlesztői dokumentáció kerül bemutatásra. Elsőként az alkalmazás specifikációját ismertetem, amely tartalmazza a projekt célkitűzéseit, a fejlesztéshez használt környezetek leírását, valamint a működési logikát. Ezt követően részletesen bemutatom a szoftver architektúráját, különös tekintettel a frontend oldalon alkalmazott MVVM és MVC tervezési mintákra. A telepítési útmutató után áttekintem a rendszer egyes rétegeinek felépítését, kitérve arra, hogy az adatbázis miért nem klasszikus táblastruktúrát használ, milyen végpontokat biztosít a backend, illetve hogyan épülnek fel a frontend komponensei.

## Specifikáció

### Feladatleírás

A fejlesztés célja egy olyan webes alkalmazás megalkotása, amely támogatja az animátorok és koordinátorok közti hatékony időpontkezelést. A program lehetőséget biztosít az eseményekre való jelentkezésre, azok lemondására, az események exportálására Google naptárba. A felhasználó adatainak megtekintésére és módosítására, illetve új események létrehozására.

A rendszerben három fő szerepkört különböztetünk meg:

* Animátor:
  + Regisztráció és bejelentkezés után elérheti a saját jelentkezési felületét.
  + Jelentkezhet az elérhető eseményekre, illetve azokat le is mondhatja legkésőbb az esemény előtt 24 órával.
  + Értesítést kap emailben az események sikeres fel, és lejelentkezéséről, valamint emlékeztetőt kap az esemény előtt 12 órával.
  + Saját eseményeit exportálhatja saját Google Naptárába.
  + Adatait módosíthatja.
* Koordinátor:
  + Megtekintheti az egyes eseményekre jelentkezett felhasználókat.
  + Új eseményeket hozhat létre.
  + Megtekintheti az eddig regisztrált felhasználók adatait.
* Adminisztrátor:
  + Eseményeket törölhet.
  + A felhasználóknak jogosultságokat tud adni
  + Teljes jogosultság az alkalmazás eléréséhez

### Működési logika

Az alkalmazás főbb funkcionális egységei:

* Felhasználókezelés: regisztráció, bejelentkezés, adatok szerkesztése
* Eseménykezelés: események létrehozása, törlése (koordinátori és adminisztrátori jogosultsággal)
* Jelentkezéskezelés: animátorok jelentkezése és lemondása az egyes eseményekre
* Értesítések: automatikus email küldése jelentkezés és lemondás esetén, valamint emlékeztető email küldése az esemény előtt
* Google Calendar integráció: az animátorok jelentkezései exportálhatóak saját naptárukba

### Fejlesztői környezetek

A fejlesztés az alábbi környzetben történt:

* Visual Studio Code 1.99.1
* Frontend fejlesztés: Node.js v22.13.1, Framework: Angular CLI 19.1.6
* Backend fejlesztés: Node.js v22.13.1, Framework: Express 11.1.0
* Adatbázis: Firebase Realtime Database

A fejlesztő, illetve a kész programot futtató hardver:

* Angular CLI, Express
  + AMD Ryzen 5 5600H
  + 3.30 GHz
  + 16,0 GB RAM
  + Windows 11 Education 64 bit OS

### Felhasználói felület specifikáció

A felhasználói felülettel szembeni elvárás, hogy átlátható, letisztult designnal rendelkezzen, biztosítsa az egyszerű navigációt az alkalmazáson belül. Hibák esetén mindig egyértelmű visszajelzést kapjon a felhasználó.

## Architekrúra

### Frontend

Mind a Model-View-Controller (MVC), mind a Model-View-ViewModel (MVVM) architektúrális mintákat használja az Angular, és ezek a webes alkalmazások strukturálásának alapvető tervezési elveiként szolgálnak.

**MVC:**

* Model (M): A model az alkalmazás adatait és üzleti logikáját reprezentálja. Az Angularban ezt gyakran olyan osztályok képezik, amelyek kezelik az adatokat, műveleteket hajtanak végre, és kapcsolatba lépnek az API-kal.
* View (V): A View az alkalmazás felhasználói felülete (UI). Az Angularban a nézetet HTML-sablonok alkotják, amelyek meghatározzák, hogy a modellből származó adatokat hogyan kell megjeleníteni a felhasználó számára. A nézet felelős az adatok megjelenítéséért, illetve a felhasználói adatbevitelek rögzítéséért.
* Contorrel (C): A hagyományos MVC-ben a Controller kapcsolóként működik a modell és a nézet között. Fogadja a felhasználó bemenetét a Nézettől, feldolgozza azt, kapcsolatba lép a Modellel az adatok lekérdezése vagy frissítése érteképen, majd ennek megfelelően frissíti a Nézetet. Az Angularban a Controller szerepét nagyrészt Komponensek helyettesítik.

**MVVM:**

* Model (M): Ahogy az MVC architektúrában is, Angularban a Model szerepét olyan szolgáltatások (service-ek) vagy osztályok töltik be, amelyek az adatok kezeléséért felelnek
* View (V):Az Angularban a template-ek töltik be a View szerepét: ők jelenítik meg az adatokat és rögzítik a felhasználói beviteleket. Az adatkötés, az interpoláció és a direktívák segítségével a nézet összekapcsolódik a ViewModel-lel (komponens), így biztosítva az adatok kétirányú áramlását a logika és a megjelenítés között.
* ViewModel (VM): Angularban a ViewModel szerepét gyakran a komponensek és a hozzájuk tartozó TypeScript osztályok jelenítik meg. A komponensosztály tartalmazza az alkalmazás logikáját, és a szolgáltatásokkal (service-ekkel) kommunikálva kérdezi le vagy módosítja az adatokat. Olyan tulajdonságokat és metódusokat tesz elérhetővé, amelyekre a nézet (template) hivatkozni tud. Az Angularban ezt az összekapcsolást adatkötés (data binding) segítségével valósítjuk meg, amely lehetővé teszi az adatok zavartalan áramlását a ViewModel (komponens) és a View (template) között. [1]

### Backend

Az alkalmazás szerveroldali logikáját a Node.js környezeten belül futó Express.js keretrendszer valósítja meg. Az egyes funkciók (authentikáció, események, profil, google naptár) külön fájlokba, mappákba szervezve találhatók. Ez megkönnyíti az átláthatóságot és a fejlesztési folyamatot.

Főbb architekturális elemek:

* Router: A /routes mappában rendezheti a különböző REST végpontok deklarációit. Minden egységhez tartozik egy külön router fájl (pl. auth.routes.js, event.routes.js)
* Controllers: Az interfész-adapterek, vagy más néven adapterréteg, tartalmazzák a vezérlőket (controller-eket), API-kat és gateway-eket. Ez a réteg irányítja a kommunikáció áramlását a külső komponensek és a rendszer háttér rétege (backend) között
* Middleware: A middleware egy olyan függvény, amely hozzáfér a kérés objektumhoz (request), a válasz objektumhoz (response) és a következő middleware-hez (next). Ilyenek például: authMiddleware.js: ellenőrzi a jwt token érvényességét, rolesMiddleware.js: ellenőrzi jogosultságok alapján a hozzáférést az egyes kérésekhez [2],[3]

## Szoftverek telepítése

Ebben a fejezetben ismertetésre kerülnek az alkalmazás futtatásához szükséges szoftver- és hardverkövetelmények, majd részletesen bemutatom a szükséges szoftverek telepítésének és az alkalmazás elindításának lépéseit.

### Szoftver- és hardverkörnyezet

Szükséges hardver:

* Operációs rendszer: Windows 10, 11
* Ram: 16 GB

Szükséges szoftverek:

* Node.js v23.11.0 (<https://nodejs.org/en>)

### Telepítés

A link, ahonnan a projekt elérhető: <https://github.com/ppipicz24/Szakdolgozat>. Miután a felhasználó megnyitotta az oldat, töltse le ZIP formátumban a kódot. Letöltés után csomagolja ki a tömörített fájlt.

A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

16. ábra - Projekt letöltése GitHubról

Ha még nincsen telepítve a Node.js a számítógépére akkor azt megteheti a hivatalos weboldalról: <https://nodejs.org/en>.

Ha már telepítve van a felhasználó számítógépére a Node.js, akkor a következő lépéseket kell megtennie:

1. Nyissa meg a PowerShellt rendszergazdagént és engedélyezze a scriptek futtatását:
   * Keresse meg a PowerShell-t a Start menüben, majd *jobb klikk → ’Futtatás rendszergazdaként’*.
   * Írja be ezt a parancsot: ’*Set-ExecutionPolicy RemoteSigned’*
   * Nyomjon Entert, majd válaszoljon ’Y’-nal

Ezzel a beállítással engedélyezi, hogy a saját gépén lévő scriptek futtathatóak legyenek.

1. Nyissa meg a projekt mappáját és navigáljon a ’*Szakdolgozat-main/Backend*’ mappába. Kattintson jobb gombbal a mappába, és kattintson a ’Megnyitás a terminálban’ lehetőségre.
   * Írja be ezt a parancssort: ’*npm i’*
   * Indítsa el a szervert a ’*nodemon*’ paranccsal
2. Lépjen vissza a projekt mappájába és navigáljon a ’*Szakdolgozat-main/FrontEnd/volunteerApp’*. Kattintson jobb gombbal a mappába, és kattintson a „Megnyitás a terminálban” lehetőségre.
   * Írja be ezt a parancssort: ’*npm i’*
   * Indítsa el az applikációt az ’*npm start*’ paranccsal
3. Nyissa meg a böngészőt (Chrome, Edge, Firefox, Safari): <http://localhost:4200/>

## Alkalmazás felépítése

### Felhasználói esetek

A felhasználói eset (use-case) diagram szemléletesen mutatja be, hogy az egyes szerepkörök (aktorok) milyen funkciókat képesek végrehajtani az alkalmazásban. Az átláthatóság érdekében a diagram két részletben kerül bemutatásra.

A képen diagram, rajz, szöveg, vázlat látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

17. ábra - Felhasználói esetek diagrammja

A kezdő oldal megnyitása eset \*-gal van jelölve. Ez az a pont, ahol a jogosultságok elválnak. Innentől 3 féle szerepkör van megkülönböztetve: animátor, koordinátor, admin. A 32. ábrán mind a három szerepkörhöz kapcsolódó funkciók találhatóak.

A képen szöveg, diagram, rajz, vázlat látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

18. ábra - Felhasználói esetek szétválása

A Felhasználó oldal megnyitása \*-gal van jelölve. Ezt a funkciót mind a koordinátor, mind az admin végre tudja hajtani, azonban a jogosultságkezelésre csak az admin jogosult.

Ebből az ábrából is látszik, hogy az adminisztrátornak van a legtöbb jogköre, míg a koordinátornak valamivel kevesebb. Az animátoroknak a saját adatai módosításán kívül semmilyen más módosító/törlő funkcióra nincsen ráhatása.

### Felhasználói történetek

A felhasználói történetek részletesebb betekintést nyújtanak az alkalmazás felületén elérhető események, funkciók és navigációs lehetőségek működésébe. Ezek a történetek bemutatják a felhasználók által végrehajtott tevékenységek sorozatát, valamint azok lehetséges kimeneteleit különböző szituációkban.

3. táblázat - Felhasználói történetek

|  |  |  |
| --- | --- | --- |
| AS A | Felhasználó |  |
| I WANT TO | Bejelentkezni | |
| 1 | GIVEN | Rendelkezik regisztrált e-mail címmel és jelszóval |
| WHEN | Adatok megadása helyesen és megnyomja a 'Bejelentkezés' gombot |
| THEN | Sikeres hitelesítés után a rendszer belépteti a felhasználót |
| 2 | GIVEN | Helytelen e-mail cím vagy jelszó megadása |
| WHEN | Megnyomja a 'Bejelentkezés' gombot |
| THEN | A rendszer hibaüzenetet jelenít meg a hibás adatokról |
| 3 | GIVEN | Elfelejtette a jelszót |
| WHEN | Elfelejtett jelszó' linkre kattint, megadja az e-mail-címét |
| THEN | A rendszer feldob egy ablakot, ahol új jelszó adható meg |
| 4 | GIVEN | Megtekinti a 'Bejelentkezés' oldalt |
| WHEN | - |
| THEN | Megjelenik egy űrlap, ahol látható a 'Felhasználónév', 'Jelszó' szövegdobozok, az 'Elfelejtett jelszó' link és a 'Bejelentkezés' gomb |

|  |  |  |
| --- | --- | --- |
| AS A | Új felhasználó |  |
| I WANT TO | Regisztrálni | |
| 1 | GIVEN | Nem rendelkezik fiókkal a rendszerben |
| WHEN | Helyesen megadja az adatait, majd rákattint a 'Regisztráció' gombra |
| THEN | Létrejön a fiók és a rendszer a 'Bejelentkezés' oldalra navigál |
| 2 | GIVEN | Helytelen adatok megadása |
| WHEN | - |
| THEN | A 'Regisztráció' gomb inaktív, a rendszer validációs hibát ad vissza |
| 3 | GIVEN | Megad egy jelszót a regisztráció során |
| WHEN | Gyenge erősségű a jelszó |
| THEN | A 'Regisztráció' gomb inaktív, a rendszer validációs hibát ad vissza |
| 4 | GIVEN | Megtekinti a 'Regisztráció' oldalt |
| WHEN | - |
| THEN | Megjelenik egy űrlap, ahol látható a 'Felhasználónév', 'Név', 'E-mail cím', 'Jelszó', 'Jelszó megerősítése', 'Telefonszám' szövegdobozok és a 'Regisztráció' gomb |

|  |  |  |
| --- | --- | --- |
| AS A | Felhasználó |  |
| I WANT TO | Kijelentkezni | |
| 1 | GIVEN | Bejelentkezett felhasználó |
| WHEN | Kijelentkezés' gombra kattint |
| THEN | A rendszer kijelentkezteti a felhasználót, megjelenik a 'Bejelentkezés' oldal |

|  |  |  |
| --- | --- | --- |
| AS A | Bejelentkezett felhasználó |  |
| I WANT TO | A kezdőoldalon lenni | |
| 1 | GIVEN | Megtekinti a kezdő oldalt |
| WHEN | - |
| THEN | Megjelenik a fejléc: VolunteerApp, Események, Profil, Üdv, <felhasználó neve>, Kijelentkezés  Megjelenik az 'Eseményeim' felület a már létrehozott események részleteivel  Megjelenik a 'Jelentkezett eseményeim' felület |
| 2 | GIVEN | Megtekinti az események táblázatot |
| WHEN | Nincsenek események |
| THEN | Még nincsenek események' felirat jelenik meg |
| 3 | GIVEN | Megtekinti az események táblázatot |
| WHEN | Vannak események |
| THEN | A táblázatban a következő oszlopok láthatóak: Esemény neve, Dátum, Idő, Nyelv, Kor, Létszám, Jelentkezés/Lejelentkezés gomb |
| 4 | GIVEN | Megtekinti a jelentkezett események táblázatot |
| WHEN | Nincsenek események |
| THEN | Megjelenik a 'Még nem jelentkeztél egy eseményre sem' felirat |
| 5 | GIVEN | Megtekinti a jelentkezett események táblázatot |
| WHEN | Vannak események |
| THEN | A táblázatban a következő oszlopok láthatóak: Esemény neve, Dátum, Idő, Nyelv, Kor, Létszám, Lejelentkezés gomb, Naptár ikon |
| 6 | GIVEN | Jelentkezett esemény |
| WHEN | Az 'Előző' vagy 'Következő' gombra kattint |
| THEN | Megjelenik az előző vagy a következő oldal az események listájából |

|  |  |  |
| --- | --- | --- |
| AS A | Bejelentkezett koordinátor, admin |  |
| I WANT TO | A kezdőoldalon lenni | |
| 1 | GIVEN | Megtekinti a kezdő oldalt |
| WHEN | - |
| THEN | Megjelenik a fejléc: VolunteerApp, Események, Profil, Felhasználók, Új esemény, Üdv, <felhasználó neve>, Kijelentkezés; Megjelenik az 'Eseményeim' felület a már létrehozott események részleteivel; Megjelenik a 'Jelentkezett eseményeim' felület |
| 2 | GIVEN | Megtekinti az események táblázatot |
| WHEN | Vannak események |
| THEN | A táblázatban a következő oszlopok láthatóak: Esemény neve, Dátum, Idő, Nyelv, Kor, Létszám, Jelentkezés/Lejelentkezés gomb, Részletek ikon |

|  |  |  |
| --- | --- | --- |
| AS A | Bejelentkezett admin |  |
| I WANT TO | A kezdőoldalon lenni | |
| 1 | GIVEN | Megtekinti az események táblázatot |
| WHEN | Vannak események |
| THEN | A táblázatban a következő oszlopok láthatóak: Esemény neve, Dátum, Idő, Nyelv, Kor, Létszám, Jelentkezés/Lejelentkezés gomb, Törlés gomb, Részletek ikon |
| 2 | GIVEN | Esemény amire jelentkezni lehet |
| WHEN | A törlés gombra kattint |
| THEN | Kitörlődik az esemény az adatbázisból |

|  |  |  |
| --- | --- | --- |
| AS A | Bejelentkezett felhasználó |  |
| I WANT TO | Eseményre jelentkezni, lejelentkezni | |
| 1 | GIVEN | Esemény amire jelentkezni lehet |
| WHEN | Megnyomja a 'Jelentkezés' gombot |
| THEN | A 'Jelentkezés' gomb helyett megjelenik a 'Lejelentkezés' gomb, a rendszer eltárolja a jelentkezést és megjelenik a 'Jelentkezett eseményeim' rész alatt |
| 2 | GIVEN | Jelentkezett esemény |
| WHEN | Megnyomja a 'Lejelentkezés' gombot |
| THEN | A 'Lejelentkezés' gomb helyett megjelenik a 'Jelentkezés' gomb, a rendszer törli az adatbázisból és a 'Jelentkezett eseményeim' részből a jelentkezést |
| 3 | GIVEN | Időpont amire jelentkezni lehet/Jelentkezett időpont |
| WHEN | Megnyomja a 'Jelentkezés'/'Lejelentkezés' gombot |
| THEN | Kap egy emailt a jelentkezés/lejelentkezés sikerességéről |
| 4 | GIVEN | Jelentkezett esemény |
| WHEN | 12 órával az esemény időpontja előtt van |
| THEN | Kap egy emlékeztető emailt az esemény idejéről |

|  |  |  |
| --- | --- | --- |
| AS A | Bejelentkezett felhasználó |  |
| I WANT TO | Google Calendarba exportálni eseményt | |
| 1 | GIVEN | Jelentkezett esemény |
| WHEN | A 'Google Calendar csatlakozás' gombra kattint |
| THEN | A rendszer átnavigál a Google saját bejelentkezési oldalára |
| 2 | GIVEN | Csatlakoztatva van a felhasználó a Google Calendarhoz |
| WHEN | A naptár ikonra kattint |
| THEN | A rendszer egy Google API hívás után exportálja Google Calendarba a megadott időpontokat |

|  |  |  |
| --- | --- | --- |
| AS A | Bejelentkezett koordinátor, admin |  |
| I WANT TO | Megtekinteni az eseményre jelentkezett animátorokat | |
| 1 | GIVEN | Események a táblázatban |
| WHEN | Rákattint a részletek ikonra |
| THEN | Átnavigál a rendszer a 'Jelentkezett felhasználók' felületre |
| 2 | GIVEN | Megtekinti a felületet |
| WHEN | - |
| THEN | Láthatóak a jelentkezett felhasználók adatai |

|  |  |  |
| --- | --- | --- |
| AS A | Bejelentkezett felhasználó |  |
| I WANT TO | A ’Profil’ oldalon lenni | |
| 1 | GIVEN | Kezdő oldalon van |
| WHEN | Rákattint a 'Profil' menüre |
| THEN | A rendszer átnavigál a 'Profil' oldalra |
| 2 | GIVEN | Megtekinti a ’Profil’ oldalt |
| WHEN | - |
| THEN | Megjelennek a felhasználó adatai, a 'Módosítás' gomb és a 'Jelszó megváltoztatása' gomb |
| 3 | GIVEN | Megtekinti a profil adatokat |
| WHEN | A 'Módosítás' gombra kattint |
| THEN | Megjelenik az adatok módosítására szolgáló oldal |
| 4 | GIVEN | Az 'Adatok módosítás' oldal megjelenik |
| WHEN | - |
| THEN | Szerkeszteni tudja az adatait |
| 5 | GIVEN | Megtekinti a profil adatokat |
| WHEN | A 'Jelszó megváltoztatása' gombra kattint |
| THEN | Megjelenik a jelszó módosítására szolgáló oldal |
| 6 | GIVEN | A 'Jelszó megváltoztatása' oldal megjelenik |
| WHEN | - |
| THEN | Megváltoztathatja a jelszavát a régi és az új jelszó megadásával |

|  |  |  |
| --- | --- | --- |
| AS A | Bejelentkezett koordinátor |  |
| I WANT TO | A 'Felhasználók' oldalon lenni | |
| 1 | GIVEN | Kezdő oldalon van |
| WHEN | Rákattint a 'Felhasználók' menüre |
| THEN | A rendszer átnavigál a 'Felhasználók' oldalra |
| 2 | GIVEN | Megtekinti az oldalt |
| WHEN | - |
| THEN | Megjelenik egy táblázat a már jelentkezett felhasználók adatival |
| 3 | GIVEN | Megtekinti a táblázatot |
| WHEN | - |
| THEN | A táblázatban a következő oszlopok láthatóak: Név, Email, Telefonszám |

|  |  |  |
| --- | --- | --- |
| AS A | Bejelentkezett admin |  |
| I WANT TO | A 'Felhasználók' oldalon lenni | |
| 1 | GIVEN | Megtekinti az oldalt |
| WHEN | - |
| THEN | Megjelenik egy táblázat a már jelentkezett felhasználók adatival |
| 2 | GIVEN | Megtekinti a táblázatot |
| WHEN | - |
| THEN | A táblázatban a következő oszlopok láthatóak: Név, Email, Telefonszám, Admin, Koordinátor |
| 3 | GIVEN | Felhasználó a táblázatban |
| WHEN | Az 'Admin' vagy a 'Koordinátor' oszlopban levő jelölőnégyzetre kattint |
| THEN | Megváltozik a felhasználó jogosultsága |

|  |  |  |
| --- | --- | --- |
| AS A | Bejelentkezett koordinátor, admin |  |
| I WANT TO | A kezdőoldalon lenni | |
| 1 | GIVEN | Kezdő oldalon van |
| WHEN | Rákattint az 'Új esemény' menüre |
| THEN | A rendszer átnavigál az 'Új esemény hozzáadása' oldalra |
| 2 | GIVEN | Megtekinti az oldalt |
| WHEN | - |
| THEN | Megjelenik az űrlap, ahol megadhatóak a következő adatok: Esemény neve, Esemény ideje, Idő, Létszám, Korosztály, Magyar |
| 3 | GIVEN | Megtekinti az oldalt |
| WHEN | Helyesen megadja az adatokat, majd rákattint a 'Hozzáadás' gombra |
| THEN | Létrejön az új esemény és a rendszer visszanavigál a kezdőoldalra  Az új esemény megjelenik az események táblázatban |

### Adatbázis

Az alkalmazás adatbázis része Firebase Realtime Database-ben tárolódik. Ez egy felhőben tárolt adatbázis. Az adatokat JSON formátumban tárolja, és valós időben szinkronizál minden csatlakozott klienssel.

Lehetővé teszi, hogy tartalmas, kooperatív alkalmazásokat hozzon létre, mivel biztonságos hozzáférést biztosít az adatbázishoz közvetlenül az ügyféloldali kódból. Az adatok helyben tárolódnak, és a valós idejű események még offline állapotban is folyamatosan működnek, így a felhasználónak rugalmas szolgáltatást nyújtanak. [4],[5]

#### Az adatbázis felépítése

Minden adat JSON objektumként tárolódik. Az adatbázisra úgy is lehet gondolni, mint egy felhőben tárolt JSON-fára. Az SQL-adatbázissal ellentétben nincsenek táblák vagy rekordok. A hozzáadott adatok egy fa-struktúrában kerülnek tárolásra, ahol minden csomópontnak (node) saját kulcsa van. Ez a kulcs lehet egyedi azonosító (pl. felhasználói ID), vagy automatikusan generált, például a push() metódus segítségével hozzáadott.

A képen szöveg, képernyőkép, képernyő látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

19. ábra - Az adatbázis felépítése

**Events:** Az események adatait tartalmazza

* age: number, az érkező korosztálya
* createdAt: automatikusan létrehozott számsor létrehozáskor
* date: string, az esemény dátuma
* id: string, az esemény id-ja
* isFull: boolean, megtelt-e az esemény
* isHungarian: boolean, magyar vagy angol-e a csoport
* name: string, az esemény neve
* numberOfPeople: number, az érkező csoport létszáma
* time: number, az esemény időpontja

Esemény törlése lehetséges adminisztrátorként, ekkor a „kapcsolótáblából” is kitörlődnek a hozzá tartozó adatok

**Users:** A felhasználók adatait tartalmazza

* createdAt: automatikusan generált számsor létrehozáskor
* email: string, a felhasználó email címe
* googleCalendar
  + accessToken: string, hozzáférés a Google Api-hoz
  + connected: boolean, csatlakozott-e már a felhasználó az adatbázishoz vagy sem
  + connectedAt: automatikusan generált számsor csatlakozáskor
  + refreshToken: string, accessTokent helyettesítő token
* id: string, a felhasználó id-ja
* name: string, a felhasználó neve
* password: string, a felhasználó jelszava Hashelve
* phoneNumber: string, a felhasználó jelszava
* role: a felhasználó jogosultsága

Felhasználó törlése nem lehetséges az adatbázisból

**userEvents:** A User és az Events táblát kapcsolja össze. Itt tárolódik, hogy a felhasználók milyen eseményekre jelentkeztek.

* eventId: string, az esemény id-ja
* id: string, a jelentkezett esemény id-ja
* registeredAt: automatikus generált számsor, amikor a felhasználó jelentkezett az eseményre
* reminderSent: boolean, a felhasználó kapott-e már emlékeztető emailt
* userId: string, a felhasználó id-ja
* user\_event: string, userId\_eventId

Törölhető adat az adatbázisból abban az esetben, ha egy esemény kitörlődik, vagy ha a felhasználó lejelentkezett az adott eseményről.

### Backend

Ahogyan az már korábban is említésre került, az alkalmazás backendje Node.js Express Frameworkével került elkészítésre. A végpontok külön szegmensekre lettek bontva funkcionalitás alapján: autentikáció (auth.routes.js), események (events.routes.js), felhasználókezelés (users.routes.js), Google Calendar (googleCalendar.routes.js). A végpontok hozzáférését szabályozó függvények a middlewarek között találhatóak.

#### Végpontok

**auth.routes.js:**

* POST /auth/register:
  + A regisztrációért felelős végponthívás.
  + A háttérben le is validálja a beérkező adatokat, hogy azok megfelelőek-e az elvárásnak, ellenőrzi szerepel-e már az adott felhasználó az adatbázisban.
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Request body:
    - name – név
    - email – email cím
    - password – jelszó
    - phoneNumber – telefonszám
    - role – szerepkör
  + Response messages:
    - 201: User registered successfully – sikeres regisztráció
    - 400: All fields are required – minden mező kitöltése kötelező
    - 400: Name must be at least 3 characters long – a név legalább 3 karakter hosszú kell legyen
    - 400: Invalid email format – érvénytelen email formátum
    - 400: Password must be at least 8 characters long, contain 1 uppercase letter and 1 number – a jelszó legalább 8 karakter hosszú kell legyen, tartalmaznia kell legalább 1 nagybetűt és 1 számot
    - 400: Email already exists – az email már létezik
    - 500: Server error, <hibaüzenet> - belső szerver hiba
* POST /auth/login:
  + A bejelentkezésért felelős végponthívás.
  + A háttérben le is validálja a beérkező adatokat, hogy azok megfelelőek-e az elvárásnak, a felhasználó létezik-e az adatbázisban.
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Request body:
    - email – email cím
    - password – jelszó
  + Response messages:
    - 200: Login successful – sikeres bejelentkezés
    - 400: Email and password are required – az email és a jelszó megadása kötelező
    - 400: Invalid email or password – érvénytelen email vagy jelszó
    - 500: Server error, <hibaüzenet> - belső szerver hiba
* POST /auth/logout:
  + A kijelentkezésért felelős végponthívás.
  + Csak kijelentkezteti a felhasználót, nem validál semmit.
  + Response messages:
    - 200: Logout successful – sikeres kijelentkezés
* POST /auth/forgot-password:
  + Az új jelszó kiküldéséért felelős végponthívás.
  + Ha a felhasználó új jelszót igényel, akkor egy random generált jelszót kap emailben, és módosítja az adatbázist.
  + Ellenőrzi, hogy létezik-e az email cím az adatbázisban.
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Request body:
    - email – email cím
  + Response messages:
    - 201: New password sent to your email – az új jelszó elküldve az emailre
    - 400: Email is required – email kitöltése kötelező
    - 400: Invalid email format – érvénytelen email formátum
    - 404: User not found – a felhasználó nem található
    - 500: Server error, <hibaüzenet> - belső szerver hiba
* POST /auth/new-password:
  + Az új jelszó beállításáért felelős végponthívás.
  + Ha a felhasználó új jelszót állít be magának módosítja az adatbázist, illetve validálja a beérkező adatokat.
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Request body:
    - oldPassword – régi jelszó
    - newPassword – új jelszó
  + Response messages:
    - 201: Password updated successfully – sikeres jelszó frissítés
    - 400: Old password, new password and userId are required – a régi és az új jelszó kitöltése kötelező
    - 400: Invalid old password – érvénytelen az új jelszó
    - 400: Password must be at least 8 characters long, contain 1 uppercase letter and 1 number - a jelszó legalább 8 karakter hosszú kell legyen, tartalmaznia kell legalább 1 nagybetűt és 1 számot
    - 500: Server error, <hibaüzenet> - belső szerver hiba

**events.routes.js:**

* POST /events:
  + Az új események létrehozásáért felelős végponthívás.
  + A háttérben le is validálja a beérkező adatokat, hogy azok megfelelőek-e az elvárásnak.
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Request body:
    - name – esemény neve
    - date – esemény dátuma
    - time – esemény ideje
    - numberOfPeople – csoport lészáma
    - age – csoport átlagéletkora
    - isHungarian – magyar-e a csoport
    - isFull – megtelt-e a jelentkezett animátorok száma vagy sem.
  + Response messages:
    - 201: Event created successfully – az esemény sikeresen létrehozva
    - 400: All fields are required – az összes mező kitöltése kötelező
    - 400: Name must be at least 5 and maximum 50 characters long – a név legalább 5, legfeljebb 50 karakter hosszú kell legyen
    - 400: Number of people must be at least 1 – a létszámnak legalább 1-nek kell lennie
    - 400: Age must be at least 1 – a kornak legalább 1-nek kell lennie
    - 400: Date must be in the future – a dátumnak jövőbeninek kell lennie
    - 400: Time must be between 0 and 24 – az időnek 0 és 24 között kell lennie
    - 400: isFull must be true or false – az isFull igaz vagy hamis kell legyen
    - 500: Server error, <hibaüzenet> - belső szerver hiba
* GET /events:
  + Az események lekéréséért felelős végponthívás.
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Response messages:
    - 200: <esemény adatai>
    - 500: Server error, <hibaüzenet> - belső szerver hiba
* DELETE /events/:id:
  + Az események törléséért felelős végponthívás.
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Request:
    - params.id – az esemény azonosítója
  + Response messages:
    - 200: Event and related registrations deleted successfully – az esemény és a hozzá tartozó kapcsolatok törlése sikeres
    - 404: Event not found – az esemény nem található
    - 500: Server error, <hibaüzenet> - belső szerver hiba
* POST /events/:id/register:
  + Az eseményekre jelentkezésért felelős végponthívás.
  + Ha a felhasználó jelentkezett egy eseményre akkor az bekerül a „kapcsoló táblába”.
  + Jelentkezés után az animátor és a koordinátor emailben értesül a jelentkezésről.
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Request:
    - params.id – az esemény azonosítója
    - user.id – a felhasználó azonosítója
  + Response messages:
    - 200: Successfully registered for the event – sikeres jelentkezés
    - 400: User already registered for this event – a felhasználó már regisztrált az eseményre
    - 404: Event not found – az esemény nem található
    - 500: Server error, <hibaüzenet> - belső szerver hiba
* DELETE/events/:id/unregistered:
  + Az eseményről való lejelentkezésért felelős végponthívás.
  + Ha a felhasználó törölte a jelentkezést, a „kapcsoló táblából” kitölődnek a hozzá tartozó adatok.
  + A felhasználó emailt kap a sikeres lejelentkezésről.
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Request:
    - params.id – az esemény azonosítója
    - user.id – a felhasználó azonosítója
  + Response messages:
    - 200: Successfully unregistered from event – A lejelentkezés sikeresen megtörtént
    - 404: User is not registered for this event – a felhasználó nem jelentkezett az eseményre
    - 500: Server error, <hibaüzenet> - belső szerver hiba
* GET /events/send-reminder:
  + Automatikus email küldésért felelős végponthívás.
  + 12 órával az esemény kezdete előtt a felhasználó emailben emlékeztető üzenetet kap.
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Response messages:
    - 500: Server error, <hibaüzenet> - belső szerver hiba
* GET /events/my-events:
  + A felhasználó saját eseményei lekérdezéséért felelős végponthívás
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Request:
    - user.id – a felhasználó azonosítója
  + Response messages:
    - 200: <regisztrált események>
    - 500: Server error, <hibaüzenet> - belső szerver hiba
* GET /events/:id/registered-users:
  + Az adott eseményekre feljelentkezett felhasználók lekérdezéséért felelős végponthívás
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Request:
    - params.id – az esemény azonosítója
  + Response messages:
    - 200: <jelentkezett felhasználók>
    - 500: Server error, <hibaüzenet> - belső szerver hiba

**googleCalendar.routes.js:**

Ahhoz, hogy elérhetővé váljanak a Google API-k OAuth 2.0 protokollt használ a hitelesítéshez és engedélyezésekhez.

Az OAuth 2.0 egy nyílt szabványú *engedélyezési* protokoll, amelyet arra használnak, hogy harmadik féltől származó alkalmazások biztonságosan hozzáférhessenek egy felhasználó adataihoz anélkül, hogy közvetlenül megszereznék a felhasználó jelszavát. A Google API-k (például Google Calendar, Gmail, Google Drive stb.) használatához gyakran szükséges az OAuth 2.0 alapú hitelesítés, hogy a felhasználó engedélyezze az alkalmazás számára az adataihoz való hozzáférést. [6],[7],[8]

* GET /google/auth/google:
  + Google bejelentkezésre átirányításért felelős végponthívás
  + Request:
    - query.state – eltárolja a felhasználó azonosítóját és az eredeti útvonalat ahová majd visszatér a program
* GET /google/auth/google/callback:
  + Bejelentkezés után hívódik meg, innen kapja meg a backend a további adatokat: refresh token, access token
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Request:
    - query.code: a google-től kapott hitelesítési kódot tárolja el
    - query.state: a felhasználótól kapott redirect útvonalat tárolja
  + Response messages:
    - 302: <redirectUrl> - a visszatérési url
    - 400: userId missing from state – a felhasználó
    - 500: Hiba a Google hitelesítés során
* POST /google/export-calendar:
  + Az esemény exportálásáért felelős végponthívás
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Request body:
    - eventId – az esemény azonosítója
    - access\_token – a Google hitelesítés során visszakapott token, csak 1 óráig érvényes, ezzel kap a backend hozzáférést a Google API-khoz
    - refresh\_token – a lejárt access\_token helyett kér másikat, hosszú élettartamú
  + Response messages:
    - 200: Sikeresen exportálva a Google Naptárba
    - 400: eventId és access\_token szükséges
    - 400: refresh\_token szükséges
    - 404: Event not found – az esemény nem elérhető
    - 500: Szerverhiba <hibaüzenet>

**users.routes.js:**

* GET /users:
  + A regisztrált felhasználók lekérdezéséért felelős végponthívás
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Response messages:
    - 200: <felhasználók>
    - 500: Server error, <hibaüzenet> - belső szerver hiba
* GET /users/profile:
  + A felhasználó profil adatai lekérdezéséért felelős végponthívás
  + A felmerülő hibákról a felhasználó érthetően értesítve van.
  + Request:
    - user.id – a felhasználó azonosítója
  + Response messages:
    - 200: <felhasználó adatai>
    - 404: User not found – a felhasználó nem létezik
    - 500: Server error, <hibaüzenet> - belső szerver hiba
* PATCH /users/profile
  + A felhasználó adatainak módosításáért felelős végponthívás
  + A felmerülő hibákért a felhasználó érthetően értesítve van.
  + A háttérben le is validálja a beérkező adatokat, hogy azok megfelelőek-e az elvárásnak, és módosítja az adatbázist.
  + Request:
    - user.id – a felhasználó azonosítója
    - body:
      * name – felhasználó neve
      * phoneNumber – a felhasználó telefonszáma
      * email – a felhasználó email címe
  + Response messages:
    - 200: Profile updated successfully – A profiladatok sikeresen frissültek
    - 400: At least one field is required for update – legalább egy mező kitöltése kötelező
    - 400: Invalid email format – nem megfelelő az email formátuma
    - 404: User not found - a felhasználó nem létezik
    - 500: Server error, <hibaüzenet> - belső szerver hiba
* PATCH /users/:id/role:
  + A felhasználók szerepköreinek megváltoztatásáért felelős végponthívás
  + A felmerülő hibákért a felhasználó érthetően értesítve van.
  + A háttérben le is validálja a beérkező adatokat, hogy azok megfelelőek-e az elvárásnak, és módosítja az adatbázist.
  + Request:
    - params.id – a felhasználó azonosítója
    - body: role – szerepkör
  + Response messages:
    - 200: User role updated to <szerepkör> - A felhasználó szerepköre frissült
    - 400: Invalid role provided – érvénytelen szerepkör
    - 404: User not found – a felhasználó nem létezik
    - 500: Server error, <hibaüzenet> - belső szerver hiba

#### Middlewares

**authenticateToken:**

* Ellenőrzi, hogy az adott végponthívás bejelentkezett felhasználóval történik-e
* Lekéri a headerből a felhasználó tokenjét, ehhez json web tokent használ.
* Request:
  + headers[”authorization”]
* Response messages:
  + 401: Access denied – hozzáférés megtagadva
  + 403: Invalid token – érvénytelen token

**isAdminOrCoordinator:**

* Ellenőrzi, hogy van-e admin vagy koordinátori jogosultsága a felhasználónak
* Request:
  + user.id – a felhasználó azonosítója
* Response messages:
  + 403: Access denied. Admin or Coordinator role required – Hozzáférés megtagadva. Admin vagy koordinátori jogosultság szükséges
  + 500: Server error, <hibaüzenet> - belső szerver hiba

**isAdmin:**

* Ellenőrzi, hogy van-e admin jogosultsága a felhasználónak
* Request:
  + user.id – a felhasználó azonosítója
* Response messages:
  + 403: Access denied. Admin role required – Hozzáférés megtagadva. Admin jogosultság szükséges.
  + 500: Server error, <hibaüzenet> - belső szerver hiba

#### Utils

**emails.js:**

* Itt történik meg maga az email küldés, beállítódik a küldő felhasználó emailje, illetve az is, hogy kinek küldi az emailt.
* Ehhez Nodemailer-t használ az alkalmazás, amely a Node,js egy könyvtára. Arra tervezték, hogy egyszerűen és hatékonyan lehessen e-maileket küldeni. Nem szükséges külső szolgáltatás, mivel képes közvetlenül SMTP-n (Simple Mail Transfer Protocol) keresztül kapcsolódni a kiszolgálókhoz. [9]

#### Database

**firebase.js:**

* Itt történik meg az adatbázis importálása firebase-ből
* 3 részre osztódnak az adatok: users, events, userEvents

### Frontend

A korábbiakban már szóba került, hogy az alkalmazás frontend része a Node.js Angular könyvtárával készült. Az Angular szintaxisa a TypeScript és a HTML nyelv kombinációjával jön létre.

#### Komponensek

Azzal, hogy komponenseket hozunk létre, megnövekszik az újra felhasználás a kódban, mivel azokat felhasználhatjuk más komponenseken belül is.

A stílusok nem a HTML-ben találhatóak, hanem egy erre létrehozott CSS fájlban, ami a komponens nevéhez hasonló. Azokat a stílusokat, amiket ebben a fájlban definiálunk, csak az adott komponensre lesznek érvényesek.

A Service-k összekötik a backendet a frontenddel. A Service-en belül hívódnak meg egyes függvényeken keresztül a komponenshez tartozó végpontok, majd azokat átadódnak a frontend irányába.

A képen szöveg, képernyőkép, Betűtípus, képernyő látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

20. ábra - backend összekapcsolása a frontenddel (auth.service.ts)

A Component-ek megkapják a Service-től az adatokat, majd a metódusok segítségével a HTML-ben található funkciókat működésre bírják.

A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

21. ábra - Service befecskendezése a komponensbe (auth.component.ts)

Angularban a navigációért egy beépített Router könyvtár felel, ami a böngésző címsorába beírt URL-t adja át a komponensnek. A *routerLink=””* megadásával adhatjuk meg, hogy az oldal hova navigáljon.

A képen szöveg, képernyőkép, Betűtípus, sor látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

22. ábra - Navigálás routerLink használatával (header.component.html)

#### Auth

A bejelentkezés és a regisztráció működéséért és megjelenítéséért felelős komponens. Ezen kívül az *auth.service.ts* felel a felhasználói adatok kezeléséért is.

A komponens fő metódusai:

* onSwitchModel*():* kattintásra a bejelentkezés és a regisztráció között vált.
* *onForgetPassword():* kattintásra a ’/forget-password’ linkre navigál.
* onSubmit(NgForm): kattintásra a bejelentkezésnél/regisztrációnál megadott adatok elküldésre kerülnek a backend irányába.

A képen szöveg, képernyőkép, diagram, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

23. ábra - Auth komponens-diagramja

#### Edit

A felhasználói adatok módosításáért felelős komponens. Megjeleníti az űrlapot, ahol a módosítás végrehajtható.

A komponens fő metódusa:

* onSubmit(NgForm): kattintásra a felhasználói adatok módosításánál megadott adatok elküldésre kerülnek a backend irányába.

A képen szöveg, képernyőkép, diagram, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

24. ábra - Edit komponens-diagramja

#### Event-details

Eseményenként kezeli a felhasználókat. Egy ’Jelentkezett felhasználók’ oldalon mutatja meg a már korábban jelentkezett animátorokat az adott eseményre.

A komponens fő metódusa:

* loadRegisteredUser(string): betölti azokat a felhasználókat, akik a „kapcsolótábla” alapján jelentkeztek az adott eseményre.

A képen szöveg, képernyőkép, Betűtípus, dokumentum látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

25. ábra – Event-details komponens-diagramja

#### Forget-password

Az új jelszó igényléséért felelős komponens. Megjeleníti az elfelejtett jelszóhoz tartozó űrlapot, ahol az új jelszó igénylése lehetséges.

A komponens fő metódusa:

* onSubmit(): kattintásra elküldi a kérelmet a backend felé, hogy új jelszót igényelt a felhasználó az email címe megadásával.

A képen szöveg, képernyőkép, Betűtípus, diagram látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

26. ábra - Forget-password komponens-diagramja

#### Header

A fejléc megjelenítéséért és az oldalak közötti navigálásáért felelős komponens. A komponens leellenőrzi, hogy be van-e jelentkezve a felhasználó. Ettől függően változik a felirat a fejlécen.

A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

27. ábra - Fejléc felépítése HTML-ben

#### Home

A főoldal az események megjelenítéséért felelős komponens. A legkomplexebb komponens, hiszen itt megjelenik a komponensek újrafelhasználhatósága, így minden komponens service-e megjelenik. Itt történik meg az eseményekre való jelentkezés, a Google Naptárhoz csatlakozás, az események törlése és a részletek megtekintése is.

A komponens fő metódusai:

* eventPastCheck(EventModel): Ellenőrzi, hogy az adott esemény időpontja elmúlt-e már a jelenhez képest.
* checkGoogleCalendarConnection(): Ellenőrzi, hogy a felhasználó csatlakozott-e a Google Calendarhoz vagy sem.
* handleGoogleRedirect(): beállítja a localStorage-ban a Google-től kapott tokeneket csatlakozás után.
* exportGoogleToGoogle(string): kattintásra exportálja az eseményt a Google Naptárba.
* onClickApply(string): kattintásra megtörténik a jelentkezés.
* deleteEvent(string): kattintásra a felhasználó kitörli az eseményt.
* eventDetails(string): kattintásra átnavigál a ’Jelentkezett felhasználók’ oldalra.
* onConnectGoogleCalendar(): kattintásra átnavigál a Google autentikációra.

A képen szöveg, képernyőkép, Betűtípus, diagram látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

28. ábra - Home komponens-diagramja

#### New-date

Az új esemény létrehozásáért felelős komponens. Megjelenít egy űrlapot, aminek a kitöltésével új esemény létrehozása lehetséges.

A komponens fő metódusa:

* onSubmit(ngForm): kattintásra elküldi a kérelmet a backend felé, hogy új esemény jöjjön létre az adatok megadásával.

A képen szöveg, képernyőkép, diagram, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

29. ábra - New-date komponens-diagramja

#### Password-reset

Új jelszó generálásáért felelős komponens. Megjelenít egy űrlapot, aminek a kitöltésével új jelszó létrehozása lehetséges.

A komponens fő metódusa:

* onSubmit(): kattintásra elküldi a kérelmet a backend felé, hogy új jelszó jöjjön létre az adatok megadásával.

A képen szöveg, képernyőkép, Betűtípus, diagram látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

30. ábra - Password-reset komponens-diagramja

#### Profile

A felhasználó profil adatainak megjelenítéséért felelős komponens.

A képen szöveg, képernyőkép, Betűtípus, diagram látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

31. ábra - Profile komponens-diagramja

#### Shared

A hibaüzenetek megjelenítéséért felelős komponens. A többi komponens használja arra, hogy könnyedén tudja megjeleníteni a hibaüzeneteket.

#### Users

A regisztrált felhasználók megjelenítéséért felelős komponens. A már korábban regisztrált felhasználók megjelennek az adataikkal együtt az oldalon. Ezen felül az adminnak itt van lehetősége megváltoztatni a felhasználók jogosultságait is.

A képen szöveg, képernyőkép, diagram, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

32. ábra - Users komponens-diagramja

## Tesztelési terv

Az alkalmazás tesztelése manuálisan történt. A fejlesztés során minden funkció implementálása közben rendszeresen ellenőriztem a helyes működést, hogy az esetleges hibák mielőbb azonosításra és javításra kerülhessenek. A fejlesztési folyamat lezárását követően egy átfogó tesztelési jegyzőkönyvet készítettem, amely tartalmazza a releváns teszteseteket. Ezt követően a dokumentáció alapján manuálisan elvégeztem a tesztelést. Az egyes tesztesetek és a hozzájuk tartozó elvárt eredmények az alábbi táblázatban találhatók.

4. táblázat - Tesztelési tervek

|  |  |  |
| --- | --- | --- |
| **Oldal/Funkció** | **Teszteset** | **Elvárt eredmény** |
| Bejelentkezés | Bejelentkezés helyes adatokkal | Sikeres belépés a rendszerbe |
|  | Bejelentkezés helytelen adatokkal | Hibaüzenet jelenik meg |
|  | Elfelejtett jelszó kezelése | Új jelszó megadására lehetőség |
|  | Bejelentkezés oldal megtekintése | Megjelenik az űrlap a megfelelő elemekkel |
| Regisztráció | Regisztráció helyes adatokkal | Fiók létrejön, navigálás a bejelentkezéshez |
|  | Regisztráció helytelen adatokkal | A 'Regisztráció' gomb inaktív, validációs hiba |
|  | Gyenge jelszó megadása regisztrációkor | A 'Regisztráció' gomb inaktív, validációs hiba |
|  | Regisztrációs oldal megtekintése | Megjelenik az űrlap a megfelelő mezőkkel: Név, Telefonszám, Email, Jelszó, Jelszó megerősítése |
| Kijelentkezés | Kijelentkezés | Kijelentkeztet és 'Bejelentkezés' oldalra visz |
| Kezdőoldal | Kezdőoldal megtekintése | Megjelenik a fejléc: ’Események’, ’Profil’, ’Kijelentkezés’, és az események felület |
|  | Események táblázat üres | 'Még nincsenek események' felirat jelenik meg |
|  | Események táblázat eseményekkel | Megjelennek az események az adatokkal, mellette a Jelentkezés gomb |
|  | Jelentkezett események táblázat üres | 'Még nem jelentkeztél egy eseményre sem' felirat |
|  | Jelentkezett események táblázat adatokkal | Megjelennek az események az adatokkal |
|  | Lapozás az események között | Megjelenik a következő oldal az események között |
|  | Koordinátor/Admin kezdőoldal megtekintése | Bővített fejléc: ’Események’, ’Profil’, ’Kijelentkezés’, ’Felhasználók’, ’Új esemény’, és események táblázat kibővítve: esemény részletei gomb, esemény törlése gomb adminként |
|  | Esemény törlése adminként | Esemény törlése az adatbázisból |
| Eseményre jelentkezés /lejelentkezés | Eseményre jelentkezés | Sikeres jelentkezés, megjelenik a lejelentkezés gomb, megjelenik a ’Jelentkezett események’ felületen az esemény |
|  | Lejelentkezés eseményről | Jelentkezés törlése, visszaáll a 'Jelentkezés' gomb |
|  | Jelentkezési/lejelentkezési értesítés | E-mail értesítés küldése |
|  | Esemény emlékeztető e-mail | Emlékeztető e-mail küldése |
|  | Esemény részletek megtekintése koordinátorként | Jelentkezett felhasználók megtekintése koordinátorként |
| Google Calendar integráció | Google Calendar csatlakozás | Navigálás Google bejelentkezési oldalra |
|  | Esemény exportálása Naptárba | Esemény exportálása a naptárba |
| Profil | Profil oldal megtekintése | Profil adatok megjelenítése |
|  | Profil adatok módosítása | Az adatok módosítása valamely mező megadásával: Név, Email, Telefonszám |
|  | Jelszó módosítása | Jelszó megváltoztatása a régi és az új jelszó megadásával |
| Felhasználók kezelése (Koordinátor) | Felhasználók oldal megtekintése koordinátorként | Felhasználói adatok táblázat megjelenítése |
| Felhasználók kezelése (Admin) | Felhasználók oldal megtekintése adminként | Felhasználói adatok + jogosultságok táblázat |
|  | Felhasználó jogosultságának módosítása | Jogosultság változtatása |
| Új esemény létrehozása | Új esemény létrehozása | Új esemény megjelenik a táblázatban |

# Összefoglalás

## Konklúzió

A szakdolgozat célja egy online, animátorok és koordinátorok közötti időpontfoglalást támogató, valós idejű visszajelzéseket biztosító webes alkalmazás fejlesztése volt, Angular frontend és Node.js alapú backend felhasználásával. Az alkalmazás lehetőséget biztosít események létrehozására, azok megtekintésére, valamint az eseményekre való jelentkezésre és lemondásra. A rendszer támogatja a Google Calendar integrációt is, így a felhasználók egyszerűen exportálhatják saját eseményeiket személyes naptárukba.

A projekt során mélyebb betekintést nyertem a webes alkalmazásfejlesztés különböző aspektusaiba: a jogosultságkezelés, valós idejű adatfrissítés, komponensalapú fejlesztés és REST API kialakítás mind a megvalósítás részét képezték. A Firebase szolgáltatásait használtam az adatok tárolására és hitelesítésre, amely lehetővé tette az adatbázis valós idejű frissítését és a felhasználók közötti gyors kommunikációt.

A felhasználói dokumentáció részletes útmutatót nyújt az alkalmazás funkcióinak használatához, míg a fejlesztői dokumentáció segítséget biztosít más fejlesztők számára az alkalmazás struktúrájának és technológiai hátterének megértéséhez. Ennek köszönhetően az alkalmazás jövőbeli bővítése és karbantartása hatékonyan megvalósítható.

## További fejlesztési lehetőségek

Az alkalmazás számos továbbfejlesztési lehetőséget rejt magában, amelyek hozzájárulhatnak a felhasználói élmény és a funkcionalitás bővítéséhez.

Elsődlegesen érdemes lenne lehetőséget biztosítani az események és a felhasználók szűrésére, ezáltal támogatva a gyorsabb és hatékonyabb böngészést. További kényelmi funkcióként megvalósítható lenne a táblázatok rendezhetősége név szerint ábécérendben, illetve időpont alapján növekvő vagy csökkenő sorrendben.

A rendszer platformfüggetlenségének bővítése érdekében célszerű lenne mobilnézet kialakítása, hosszabb távon pedig akár egy dedikált mobilalkalmazás fejlesztése is.

A funkcionalitás továbbfejlesztéseként bevezethető lenne eseményenként egy maximális létszámkorlát, amely megakadályozza, hogy az eseményekre korlátlan számú animátor jelentkezhessen.

A Google Calendar integráció kapcsán érdemes lenne kezelni a token lejáratát oly módon, hogy érvénytelen token esetén ismét megjelenjen a „Csatlakozás Google Calendarhoz” gomb, míg érvényes token esetén ez a lehetőség ne legyen elérhető a felhasználó számára.

Végül, a felhasználói adatokat érintő műveletek — például esemény törlése vagy adat módosítása — esetén célszerű lenne megerősítő felugró ablak bevezetése, amely biztosítja, hogy a felhasználó valóban végre akarja hajtani a műveletet.

# Irodalomjegyzék

1. *Medium, MVC and MVVM Patterns in Angular* (2023)

Link: <https://medium.com/front-end-world/mvc-and-mvvm-patterns-in-angular-7397e0bc7b07>, (utolsó megtekintés dátuma: 2025.04.16.)

1. *MERLINO.agency, Clean Architecture in ExpressJS Applications (NodeJS)* (2023)

Link: <https://merlino.agency/blog/clean-architecture-in-express-js-applications>, (utolsó megtekintés dátuma: 2025.04.16.)

1. *Horváth Győző, Alkalmazások fejlesztése*

Link: <https://webprogramozas.inf.elte.hu/alkfejl/02.html>, (utolsó megtekintés dátuma: 2025.04.16.)

1. *Firebase dokumentáció, Firebase Realtime Database* (2025)

Link: <https://firebase.google.com/docs/database#realtime-database>, (utolsó megtekintés dátuma: 2025.04.16.)

1. *Firebase dokumentáció, Structure Your Database* (2025)

Link: <https://firebase.google.com/docs/database/admin/structure-data>, (utolsó megtekintés dátuma: 2025.04.16.)

1. *Google Identity, Using OAuth 2.0 to Access Google APIs* (2025)

Link: <https://developers.google.com/identity/protocols/oauth2>, (utolsó megtekintés dátuma: 2025.04.22.)

1. *Google Identity, Using OAuth 2.0 for Web Server Applications* (2025)

Link: <https://developers.google.com/identity/protocols/oauth2/web-server>, (utolsó megtekintés dátuma: 2025. 04. 22.)

1. *Datatracker, The OAuth 2.0 Authorization Framework* (2012)

Link: <https://datatracker.ietf.org/doc/html/rfc6749>, (utolsó megtekintés dátuma: 2025. 04. 22.)

1. *Nodemailer dokumentáció*

Link: <https://nodemailer.com/>, (utolsó megtekintés dátuma: 2025. 04. 22.)

# Ábra-, és táblázatjegyzék

## Ábrajegyzék

[1. ábra - Bejelentkezés 3](#_Toc196506577)

[2. ábra - Új jelszó igénylése 3](#_Toc196506578)

[3. ábra - Regisztráció gombok 4](#_Toc196506579)

[4. ábra - Regisztráció 5](#_Toc196506580)

[5. ábra - Kezdőoldal 6](#_Toc196506581)

[6. ábra - Jelentkezett felhasználók 8](#_Toc196506582)

[7. ábra - Nem található jelentkezett felhasználó 8](#_Toc196506583)

[8. ábra - Nem található jelentkezett esemény 9](#_Toc196506584)

[9. ábra - Csatlakozás Google Calendar-hoz 10](#_Toc196506585)

[10. ábra - exportálás Google Calendarba 10](#_Toc196506586)

[11. ábra – Profil 10](#_Toc196506587)

[12. ábra - Adatok módosítása 11](#_Toc196506588)

[13. ábra - Jelszó megváltoztatása 12](#_Toc196506589)

[14. ábra - Felhasználók 12](#_Toc196506590)

[15. ábra - Új esemény hozzáadása 13](#_Toc196506591)

[16. ábra - Projekt letöltése GitHubról 19](#_Toc196506592)

[17. ábra - Felhasználói esetek diagrammja 21](#_Toc196506593)

[18. ábra - Felhasználói esetek szétválása 22](#_Toc196506594)

[19. ábra - Az adatbázis felépítése 30](#_Toc196506595)

[20. ábra - backend összekapcsolása a frontenddel (auth.service.ts) 42](#_Toc196506596)

[21. ábra - Service befecskendezése a komponensbe (auth.component.ts) 42](#_Toc196506597)

[22. ábra - Navigálás routerLink használatával (header.component.html) 42](#_Toc196506598)

[23. ábra - Auth komponens-diagramja 43](#_Toc196506599)

[24. ábra - Edit komponens-diagramja 44](#_Toc196506600)

[25. ábra – Event-details komponens-diagramja 45](#_Toc196506601)

[26. ábra - Forget-password komponens-diagramja 46](#_Toc196506602)

[27. ábra - Fejléc felépítése HTML-ben 46](#_Toc196506603)

[28. ábra - Home komponens-diagramja 48](#_Toc196506604)

[29. ábra - New-date komponens-diagramja 49](#_Toc196506605)

[30. ábra - Password-reset komponens-diagramja 49](#_Toc196506606)

[31. ábra - Profile komponens-diagramja 50](#_Toc196506607)

[32. ábra - Users komponens-diagramja 51](#_Toc196506608)

## Táblázatjegyzék

[2.1. táblázat - Regisztrációs űrlap mezőinek validációja 5](#_Toc196506609)

[2.2. táblázat - Új esemény hozzáadása űrlap mezőinek validációja 13](#_Toc196506610)

[3.1. táblázat - Felhasználói történetek 23](#_Toc196506611)

[3.2. táblázat - Tesztelési tervek 51](#_Toc196506612)