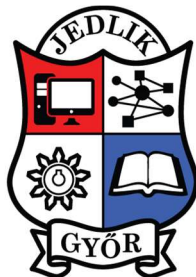




győri szakképzési centrum

Jedlik Ányos  
Gépipari és Informatikai  
Technikum és Kollégium



9021 Győr, Szent István út 7.

+36 (96) 529-480

+36 (96) 529-448

OM: 203037/003

jedlik@jedlik.eu

www.jedlik.eu

# Záródolgozat feladatkiírás

Tanuló(k) nevei: Csizmazia Ádám, Dernóczi Balázs, Sipos Roland  
Képzés: Nappali munkarend  
Szak: 5 0613 12 03 Szoftverfejlesztő és tesztelő technikus




## A záródolgozat címe: GO EVENT!

Konzulens: Bólya Gábor  
Beadási határidő: 2023.04.27.

---

**Módos Gábor**  
Igazgató

## Konzultációs lap

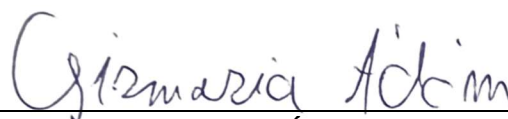
	A konzultáció		Konzulens aláírása
	ideje	témája	
1.	2022.09.15.	Témaválasztás és specifikáció	
2.	2023.03.01.	Záródolgozat készültségi fokának értékelése	
3.	2023.04.19.	Dokumentáció véglegesítése	

## Tulajdonosi nyilatkozat

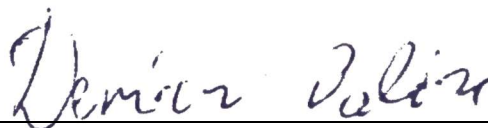
Ez a dolgozat a saját munkánk eredménye. Dolgozatunk azon részeit, melyeket más szerzők munkájából vettünk át, egyértelműen megjelöltük.

Ha kiderülne, hogy ez a nyilatkozat valótlan, tudomásul vesszük, hogy a szakmai vizsgabizottság a szakmai vizsgáról kizár minket és szakmai vizsgát csak új záródolgozat készítése után tehetünk.

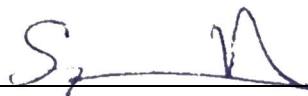
Győr, 2023. április 27.



Csizmazia Ádám



Dernóczi Balázs



Sipos Roland

**GO EVENT!**  
DOKUMENTÁCIÓ

Dokumentációt készítette: Sipos Roland

## TARTALOM

1.	Bevezetés .....	5
2.	Szoftver adatai.....	6
2.1	Szoftver neve.....	6
2.2	Szoftver arculata, külső megjelenése.....	6
2.3	Szoftver funkciói.....	7
2.4	Szoftver követelményei.....	8
3.	Fejlesztési ciklus, munkamegosztás .....	9
4.	Alkalmazott technológiák.....	10
4.1	Adatbázis .....	11
4.2	Backend .....	12
4.2.1	Szerveroldali npm csomagok.....	12
4.2.2	Mappaszerkezet .....	13
4.2.3	Adatbázis kapcsolat kialakítása .....	14
4.2.4	Controller-ek .....	14
4.2.5	Middleware-ek .....	15
4.2.6	Útvonalak .....	16
4.2.7	Biztonság .....	16
4.3	Frontend.....	17
4.3.1	Npm csomagok a frontendben.....	17
4.3.2	Komponensek.....	17
4.3.3	Útvonalak .....	18
4.3.4	Reszponzivitás .....	18
5.	Felhasználói kézikönyv .....	19
5.1	Oldal használata bejelentkezés nélkül .....	19
5.2	Bejelentkezés általános felhasználói fiókkal .....	20
5.3	Adminisztrátori hozzáférés .....	22
6.	Fejlesztői dokumentum.....	24
6.1	Fejlesztői környezet futtatása .....	24
6.2	API-k használata .....	24
6.2.1	GET .....	25
6.2.2	POST .....	26
6.2.3	PUT .....	27
6.2.4	DELETE .....	27
7.	Publikálás, fejlesztési lehetőségek .....	29
8.	Források.....	30

## 1. BEVEZETÉS

Mai világban az IT számos területen megjelenik, ilyen például a logisztika, a közlekedés, gazdaság vagy éppen a rendezvények szervezésében. Az informatika, és az informatikai rendszerek bevonásának segítségével automatizálni lehet az emberi folyamatokat, és jövőbeli előrejelzést lehet készíteni a korábban felvitt adatokból.

Csapatunk erre fókuszálva alkotta meg a GO EVENT! appot, melynek célja, hogy jellemzően ingyenes eseményeket közvetítsen a felhasználóknak. Ehhez egy webes alkalmazás fejlesztése vált indokolttá. Az alkalmazás lehetővé teszi a felhasználók számára, hogy jelezzék részvételi szándékukat az adott eseményen, ezzel segítve az esemény rendezőinek abban, hogy megbecsüljék a résztvevők számát és igényeit. Az eseményre jelentkezők pedig biztosak lehetnek abban, hogy a belépési jogukat biztosítják, ha a férőhelyek betelnek.

Az alkalmazás felhasználóbarát felülettel rendelkezik, amely könnyen kezelhető és kényelmes. Emellett az alkalmazás mobilbarát is (hisz manapság többen használjuk a mobil készülékeinket), így a felhasználók bármikor és bárhol elérhetik az adott esemény információit és a jelentkezési lehetőséget.

Az eseményekre történő online jelentkezés lehetővé tétele nemcsak a rendezvény szervezőinek könnyíti meg a tervezést és a koordinációt, hanem a felhasználói számára is nagy előnyökkel jár. Az online jelentkezés lehetősége növeli a felhasználók kényelmét, és csökkenti az esetleges kellemetlenségeket, amelyek akkor merülnének fel, ha a helyszínen már megtelt a rendezvény kapacitása.

Az alkalmazás tehát valódi értéket nyújt mind a rendezvény szervezői, mind pedig a felhasználók számára.

## 2. SZOFTVER ADATAI

### 2.1 SZOFTVER NEVE

Az első benyomás rendkívül fontos, és ezt a benyomást a névválasztás nagymértékben meghatározza, különösen a szellemi termékek esetében. A név azonosítója a terméket vagy szolgáltatást, amelyre az emberek emlékeznek, és amelynek alapján felismerik és megkülönböztetik azokat a hasonló termékektől vagy szolgáltatásoktól.

Az új szellemi termék, a GO EVENT! névvel rendelkezik, amelynek kiemelt marketing szerepe van. A név megfelel a 2022-ben használt trendeknek, és azon célcsoportok számára is vonzó, akik érdeklődnek az események, találkozók és különféle rendezvények iránt. A GO EVENT! névvel kapcsolatban fontos megemlíteni, hogy a név hangzása dinamikus és modern, ami megragadja a felhasználók figyelmét és emlékezetébe vési magát. Az egyszerűsége és rövidegsége miatt a név könnyen megjegyezhető, így az emberek könnyedén társítani tudják az adott termékkel vagy szolgáltatással.

Ezenkívül a GO EVENT! név illeszkedik az alkalmazás funkcióihoz, amelyek célja, hogy az emberek számára megkönnyítsék az események megtalálását és részvételét. Az alkalmazás biztosítja a felhasználói számára, hogy a legújabb és legnépszerűbb eseményekről értesüljenek, valamint könnyen és gyorsan jelentkezni tudjanak azokra. A GO EVENT! név jól tükrözi az alkalmazás célját, amelynek lényege, hogy az embereknek segítsen az adminisztráció során.

Összességében tehát a GO EVENT! név jelentősége nagy, mivel lehetővé teszi a termék azonosítását, megragadja a felhasználók figyelmét, és jól tükrözi az alkalmazás célját és funkcióit.

### 2.2 SZOFTVER ARCULATA, KÜLSŐ MEGJELENÉSE

A Webapp külső megjelenése alapértelmezetten sötét színű, erre azért esett a választásunk, mert a saját tervezésű logónk színéhez ez illett a legjobban, továbbá a szemünket is kíméli, azáltal, hogy nem fehér színű a kijelző a böngészés során, továbbá a mobil eszközeink így kevesebb energiát fogyasztanak.

Az arculatban a sötét színű webes felületen a ciánkék színű logó dominál mely teljes mértékben saját tervezésünk.

A logóban található „O” betű belsejében egy „play” gombot szimbolizál, amely arra ösztönzi weboldalra érkező látogatókat, hogy jelentkezzenek eseményekre az alkalmazás használatával.

A leggyakrabban használt betűtípusok közé tartoznak a modern megjelenést adó Bahnschrift és Arial fontok.



1. ábra Szellemi termék logója

## 2.3 SZOFTVER FUNKCIÓI

Az általunk fejlesztett szoftver számos olyan funkcióval rendelkezik, amelyek nélkülözhetetlenek az oldal céljának eléréséhez. A funkciók könnyen csoportosíthatók és áttekinthetők.

### 2.3.1 Autentikációhoz köthető funkciók

#### Regisztráció

A regisztráció során a felhasználó bekerül a GO EVENT! rendszerébe.

#### Bejelentkezés

A bejelentkezés során a felhasználó adatai kényelmi szempontot figyelembe véve a rendszer böngészőjében tároljuk. A bejelentkezést követően lehet jelentkezni az eseményre, valamint elérni a felhasználói oldalt.

#### Kijelentkezés

A kijelentkezés során az adatok törlődnek a böngészőből.

#### Elfelejtett jelszó

Elfelejtett jelszó esetén elég csak megadni a regisztráció során megadott email címet, és a rendszer generál egy kódot, ami segítségével be lehet lépni a felületre, ezt ajánlott módosítani a fiók menüpont alatt lévő jelszó módosítással.

#### Jelszó módosítás

A jelszó módosítás az általunk ismert jelszó frissítése, itt mezőt kell kötelezően kitölteni, a régi jelszót, valamint az új jelszót kétszer, ezt azért kérjük, hogy biztosan ismerjük, amit begépelünk.

### 2.3.1 Felhasználó szintű funkciók

#### Jelentkezés az eseményre

Ez a pont bejelentkezés után érhető el, és elég egy kattintás, és jelentkezési szándékunk részvétellé változott, melyről kapunk egy visszaigazoló emailt, mely tartalmazza az esemény fő adatait, valamint a belépéshez szükséges QR kódot. A jelentkezés feltétele, hogy az esemény korhatári kritériuma teljesüljön, valamint, hogy a helyszín kapacitása biztosított legyen.

#### Eseményre való jelentkezés lemondása

A fiókban található az adataink mellett a jelentkezett eseményeinket is, ahol a „Lemondom” gomb is megtalálható, erre kattintva törlődik az igényünk a rendszerből melyről kap a felhasználó egy emailt is.

#### Fiók törlése

A fiókunkat tudjuk törölni, ha nem vagyunk elégedettek az oldallal, valamint, ha a programok nem szimpatikusak, ezzel a fiókadatok törlődnek az adatbázisból.

### 2.3.2 Adminisztrátor szintű funkciók

#### Új esemény létrehozása

A szükséges adatok megadásával új eseményt tudunk létrehozni. A házzám megadása opcionális, ha például egy olyan helyről van szó melynek nincsen házzáma, ilyen lehet egy tér.

#### Új adminisztrátor felvétele

Lehetőség van felvenni adminisztrátort, aki elérheti az admin felületet és ezzel létrehozhat eseményt, vagy új admin fiókot, vagy éppen törölhet.

#### Esemény törlése

Táblázat jobb oldali részén lehet a kiválasztott eseményt törölni.

#### Felhasználó törlése

A felhasználók táblázatban lehetőség van törölni az egyes felhasználókat.

#### Táblázatok exportálása

A kívánt táblázatot le tudjuk menteni PDF/Xslx formátumban.

### 2.3.3 Jogosultság nélküli funkciók

#### Kapcsolati űrlap küldés

A név, email cím, tárgy és üzenet mezők kitöltésével üzenetünk továbbításra kerül a GO EVENT! adminisztrátorai felé.

#### Weboldal fordítása

Oldalunkon gondoltunk (elsősorban a szomszédos országok miatt) a többnyelvűségre, így a weboldal alján lehetőség van a kiválasztott nyelvre való fordítással.

## 2.4 SZOFTVER KÖVETELMÉNYEI

Az applikáció használatához szükséges, hogy a felhasználó rendelkezzen egy olyan eszközzel, amely képes a szoftver futtatására, legyen az asztali vagy mobileszköz, és stabil internetkapcsolattal rendelkezzen. Emellett a regisztrációhoz, illetve a jelentkezéshez szükséges, hogy a felhasználónak legyen egy érvényes email címe, amelyhez hozzáférése van, mivel a rendszer a felhasználók személyes adatainak védelme érdekében az email címet használja a regisztráció és az azonosítás során, és oda küld visszaigazoló emaileket.



### 3. FEJLESZTÉSI CIKLUS, MUNKAMEGOSZTÁS

Csapatunk 2022 szeptemberében alakult meg, azonban az ötlet még nem volt teljesen kidolgozva, így az első hónapot a csapatunk azzal töltötte, hogy tisztázzuk a szoftver célját és funkcióit, valamint ezen információk birtokában megtervezzük az adatbázist, és kiválasszuk a használni kívánt technológiákat.

Miután októberben sikerült tisztázni a szoftverünk célját és funkcióit, a csapatunk a tervezésre és az implementációra koncentrált. Számos egyeztetés során elkészült az adatbázisunk, ahol tároljuk a felhasználóink és programjaink adatait.

Az új év beköszöntével, 2023 januárjában a backend fejlesztésére összpontosítottunk, hogy össze tudjuk kapcsolni az adatbázist és a szoftvert. A backend fejlesztése során finomhangoltuk az adatbázist, hogy az még hatékonyabb legyen.

Ezt követően március elején kezdetét vette a frontend megalkotása, és annak összekötése a backenddel, ami áprilisra el is készült. Ekkor a projektünk már egyaránt képes volt szerveroldali megjelenéssel (EJS) és külsős alkalmazásként is működni, API kérések segítségével.

A fennmaradó időszakot pedig az esetleges hibák javítására, optimalizálásra, továbbá dokumentációra fordítottuk, végül 2023.04.27.-én projektünket befejeztük, és azt késznek nyilvánítva beadtuk.

A munkát úgy osztottuk fel, hogy mindenki olyan feladatot kapjon, amelyhez a legjobban ért, és amiben jobb a szaktudása. Miután eldőlték a funkciók, lefektettük azon dolgokat, mint például a változóneveket, útvonalakat, ami hatással lehet a fejlesztésre, így nyugodtan tudtunk dolgozni a projekten. Ennek értelmében az alábbi felosztás történt:

<b>Csizmazia Ádám:</b>	Frontend megalkotása Optimalizálási folyamatok
<b>Dernóczi Balázs:</b>	Adatbázis tervezése, és megalkotása + hosztolása Tesztelés GitHub fiók kezelése
<b>Sipos Roland:</b>	Backend szerver megalkotása Weboldal külső megjelenésének tervezése Dokumentáció Publikáció

Összességében a munkát jól sikerült felosztani, hisz mindenki arányosan dolgozott a projektben.

## 4. ALKALMAZOTT TECHNOLÓGIÁK

Az általunk használt technológiák közé tartozik a node.js, vue.js és MySQL. A node.js lehetővé teszi a JavaScript használatát szerveroldalon, így az alkalmazások hatékonyabbak és egyszerűbbek lehetnek. A VUE.JS egy felkapott frontend keretrendszer melyet kiegészítettünk a Bootstrap és AOS nevű kiegészítővel. Az adatok hatékony tárolására és kezelésére a MySQL adatbázis-kezelő rendszert használtunk. A fejlesztői környezetünk a Visual Studio Code volt, amely egy hatékony kódszerkesztőt biztosít, valamint a phpMyAdmin segített az adatbázisok könnyű kezelésében. Az alkalmazásunk fejlesztéséhez az XAMPP keresztplatformos szoftvercsomagot használtuk, amely tartalmazza a PHP-t és az Apache szerveret, valamint más szükséges eszközöket a szerveroldali alkalmazásunkhoz.



A fenti alkalmazásokon túl, a Microsoft OneDrive, és a GitHubot használtuk a fájlok felhőben való tárolására:

GO EVENT! - 2/14F B group

Overview Repositories 4 Projects 1 Packages Teams People 4 Settings

Find a repository... Type Language Sort New repository

- Hosted-version-Frontend** (Public) — CSS MIT 0 stars 0 forks 0 pulls Updated 20 hours ago
- Fullstack-version-with-EJS-and-VUE** (Public) — HTML MIT 0 stars 0 forks 0 pulls Updated 20 hours ago  
Ez a repository azt a verziót tartalmazza, ahol a szerveroldal képes egyéni működésre, de emellett a VUE app is képes működni.
- Hosted-version-Backend** (Public) — JavaScript MIT 0 stars 0 forks 1 pull Updated yesterday
- Development-documents** (Public) — MIT 0 stars 0 forks 0 pulls Updated 3 days ago

2. ábra GitHub csoportunk repositorjai

## 4.1 ADATBÁZIS

Az adatbázisunk tervezése során alaposan átgondoltuk, hogy milyen adatokat kell tárolnunk, és ennek megfelelően kialakítottuk az adatbázis struktúráját. Figyelembe vettük az esetleges bővítési lehetőségeket is, például egy jövőbeli modul beépítésével az előadók külön táblában való tárolását, vagy akár egy mobilalkalmazás fejlesztését az egyedi belépőkódok használatára. Döntésünk a MySQL adatbázisszerver mellett szakértelmünk és az általunk használt Node.js backend által támogatott npm csomagok miatt született.

Az adatokat tartalmazó mezők számának optimalizálása érdekében több mezőben korlátoztuk a felvitt adatok karakterszámát, figyelembe véve az igényeket és a rendelkezésre álló helyet. Érdekeség, hogy a locations táblában nem köteleztük az adminisztrátorokat a házszám megadására, mivel az esemény helyszínének jelölése a valós cím helyett egyes esetekben könnyebben felismerhető. (pl: Focipálya)

Az adatbázisunk 4 fő táblát, 2 kapcsolótáblát és összesen 29 mezőt tartalmaz, amelyek az alábbi adatokat tárolják:

### USERS

ID	A felhasználó egyedi azonosítója	(auto increment)
NAME	A felhasználó neve	
PASSWORD:	A felhasználó jelszava	(64 karakteres hash)
GENDER	A felhasználó neve	(Statisztikai adat)
PERMISSION	A felhasználó jogosultsági szintje	
BIRTHDATE	A felhasználó életkora	
EMAIL	A felhasználó email címe	
NATIONALITY:	A felhasználó állampolgársága	(Statisztikai adat)

### EVENTPROPERTIES

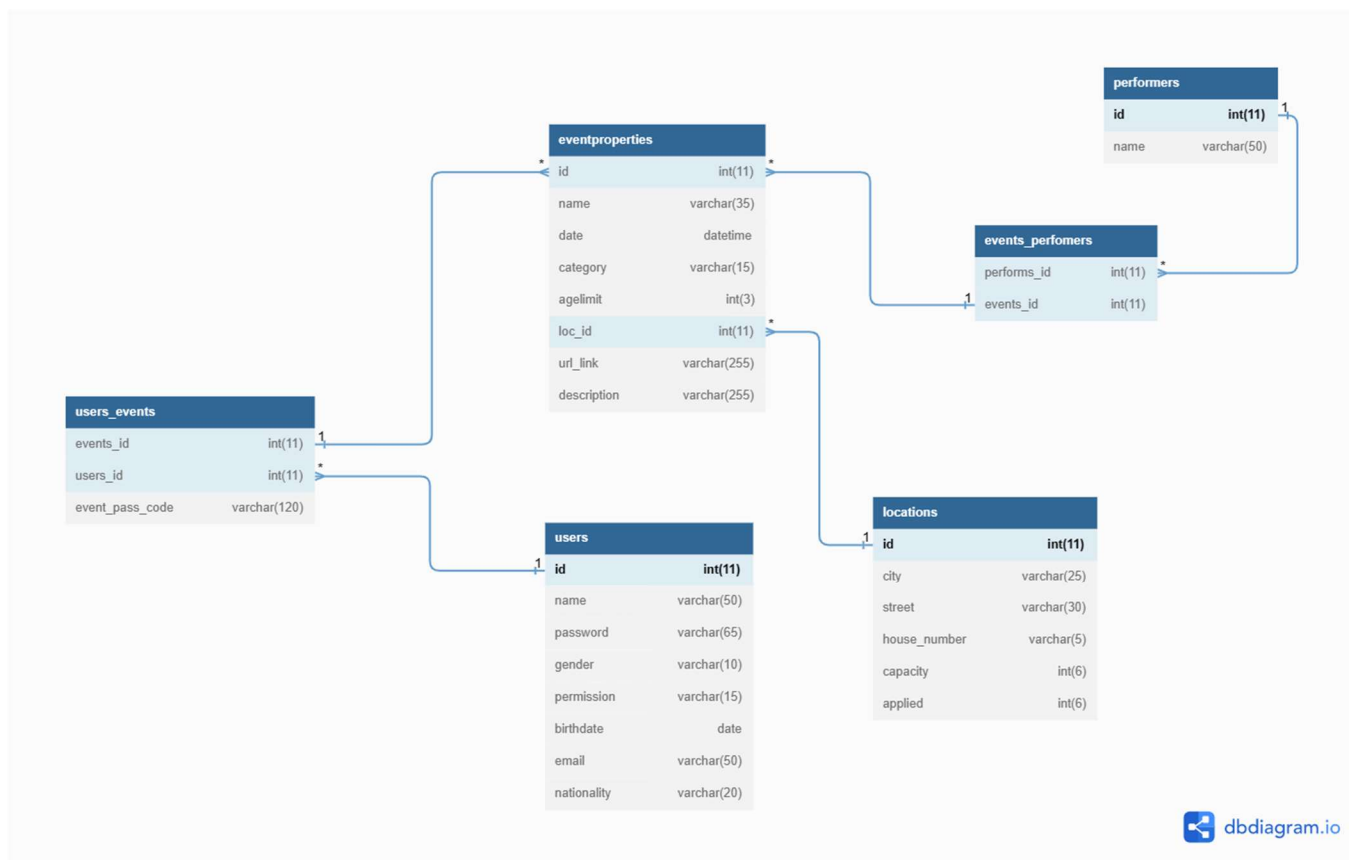
ID	Az esemény azonosítója	(auto increment)
NAME	Az esemény neve	
DATE	Az esemény megrendezésének dátuma	
CATEGORY	Az esemény kategóriája	
AGELIMIT	Az esemény korhatár besorolása	
LOC_ID	Az esemény helyszíne	
URL_LINK	Az eseményhez tartozó kép URL címe	
DESCRIPTION	Az esemény leírása	

### LOCATIONS

ID	A helyszín azonosítója	(auto increment)
CITY	A helyszín címe (város)	
STREET	A helyszín címe (utca)	
HOUSE_NUMBER	A helyszín címe (házszám)	(Lehetséges null érték)
CAPACITY	A helyszín kapacitása	
APPLIED	Jelentkezettek száma	

## PERFORMERS

ID A fellépő azonosítója (auto increment)  
 NAME A fellépő neve



3. ábra adatbázis kapcsolatok

## 4.2 BACKEND

Szakdolgozatunkban backend technológiának a Node.js-t választottuk, döntésünknel közrejátszott, hogy e keretrendszert ismerjük jobban, ebben kaptunk több elméleti oktatást, és népszerű a 2022-es trendek alapján, hisz magas szintű skálázhatóság, gyors válaszidő, egyszerű telepítés jellemzi, emellett könnyen integrálható adatbázisokkal.

### 4.2.1 SZERVEROLDALI NPM CSOMAGOK

Számos csomagot felhasználtunk a projektben, ezek nélkülözhetetlenek az oldal működéséhez, valamint nagyban megkönnyítették a programozást, mert előre definiált függvényeket használhattunk. Ezek megválasztása többnyire videók, vagy GitHub források során történt, ahol inspirálódhattunk, és felmértük melyik csomag lenne a legalkalmasabb számunkra, ezek alapján a következő csomagokat integráltuk a projektünkbe:

NPM PACK	VERZIÓ	FELADATA
async	^3.2.4	Aszinkron műveletek végrehajtására szolgáló segédprogramokat tartalmazó JavaScript könyvtár.
axios	^1.3.3	HTTP kérések végrehajtására szolgáló JavaScript könyvtár, amely egyszerű interfészt kínál a Promise API-hoz.
bcrypt	^5.1.0	Hash függvényeket és tartalmazó könyvtár a jelszavak biztonságos tárolásához.
body-parser	^1.20.1	HTTP kérések testének olvasására szolgáló könyvtár.
cookie-parser	^1.4.6	HTTP sütik olvasására és írására szolgáló könyvtár.
cors	^2.8.5	Cross-Origin Resource Sharing (CORS) funkciók bekapcsolására szolgáló könyvtár.
dotenv	^16.0.3	Környezeti változók betöltésére szolgáló könyvtár a .env fájlból.
ejs	^3.1.8	JavaScript sablonmotor az adatok megjelenítéséhez a HTML oldalakon.
express	^4.18.2	Webes alkalmazások készítéséhez szolgáló keretrendszer.
express-flash	^0.0.2	A közvetlen HTTP kérés során megjelenítendő üzeneteket kezelő könyvtár.
express-session	^1.17.3	Munkamenetkezelő könyvtár, amely lehetővé teszi a munkamenetek tárolását az Express alkalmazásokban.
method-override	^3.0.0	HTTP metódusok felülírására szolgáló middleware.
mysql	^2.18.1	MySQL adatbázis-kezelő rendszerhez készült Node.js modul.
nodemailer	^6.9.1	Email küldésre szolgáló könyvtár.
passport	^0.6.0	Felhasználó-hitelesítő middleware a Node.js-hez
passport-local	^1.0.0	Passport middleware a helyi felhasználói hitelesítéshez.
pool	^0.4.1	MySQL adatbázis-kezelő rendszerhez készült Node.js modul az adatbázis-összeköttes pool-okhoz
qrcode	^1.5.1	QR-kódok generálására szolgáló könyvtár
uuid	^9.0.0	Egyedi azonosítókat generáló könyvtár.

#### 4.2.2 MAPPASZERKEZET

Mappaszerkezetünk kialakításakor szem előtt tartottuk a logikus felépítést így az alábbi mappák találhatóak meg:

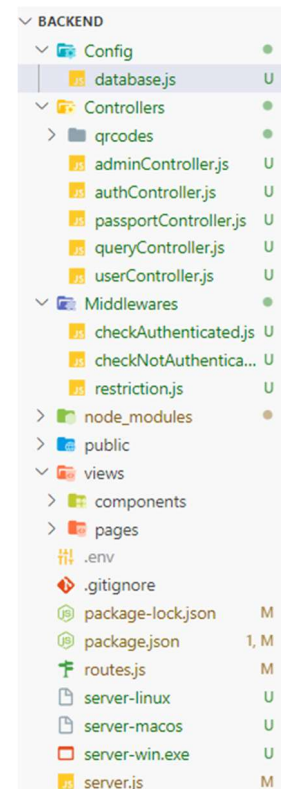
**Config:** Konfigurációs mappa

**Controllers:** Függvényeket tartalmazó mappa

**Middlewares:** Middlewareket tartalmazó mappa

**Public:** Itt találhatóak a fájlok például: képek, js, css fájlok, amik az oldal megjelenését szabályozzák

**Views:** Ebben tároljuk az EJS által használt oldalak és komponenseket



### 4.2.3 ADATBÁZIS KAPCSOLAT KIALAKÍTÁSA

Az adatbázis kapcsolat a backend és a mysql szerver között a mysql npm csomag segítségével jön létre, melyet a dotenv környezeti változókkal együtt definiáljuk a kapcsolatot:

```
1  const mysql = require('mysql');
2
3  // Adatbázis-kapcsolat létrehozása
4  const connection = mysql.createConnection({
5    host: process.env.DB_HOST,
6    user: process.env.DB_USER,
7    password: process.env.DB_PASSWORD,
8    database: process.env.DB_NAME
9  });
10
11 module.exports = connection;
```

4. ábra Adatbázis kapcsolat kialakítása környezeti változókkal

### 4.2.4 CONTROLLER-EK

A Controllerjeink jól elkülöníthetők a funkciók besorolása alapján, az alábbi táblázat bemutatja, hogy milyen függvények találhatók az alábbi controllerek alatt, a függvény neve utal a funkciójára, hisz fontos a clean-code más néven „tisztá kód” alapelv betartása.

FÁJL	FÜGGVÉNY NEVE	Függvény leírása
<b>adminController.js</b> (admin folyamatok)	AddNewEvent()	Új esemény rögzítése
	AddNewAdmin()	Új adminisztrátor rögzítése
	deleteEvent(id)	Esemény törlése
<b>userController.js</b> (általános fiók folyamatok)	contactForm()	Kapcsolati űrlap küldése
	applyToLocation()	Jelentkezés egy eseményre
	cancelApplication()	Eseményre jelentkezés lemondása
	deleteUserById(id)	Felhasználói fiók törlése ID alapján
<b>queryController.js</b> (API kérések, és adatok lekérdezése adatbázisból)	NextEventContent()	A következő esemény helye, ideje, és neve
	AllEvents()	Összes esemény
	getUsers()	Összes felhasználó adata
	getAppliedEvents()	Felhasználó eseményei
	eventsByCategories()	Események kategóriánként
	eventsByAge()	Események korhatár szerint
	eventPass()	Esemény belépőkód ellenőrzése
<b>authController.js</b> (autentikációval kapcsolatos folyamatok)	ArchivedEvents()	Archivált események
	signup()	Regisztráció
	refreshpassword()	Ismert jelszó frissítése, módosítása
<b>passportController.js</b>	forgotpassword()	Elfelejtett jelszó esetén új igénylése
	initialize()	EJS Autentikáció

### 4.2.5 MIDDLEWARE-EK

Szerverünkön 3 middleware található, ami az útvonalak korlátozását végzi, mind EJS mind API szinten, ezeket a külön fájlokban helyeztük el.

Middlewares >  checkAuthenticated.js > ...

```
1 // Ez a Middleware az EJS-ről érkező kéréseket vizsgálja,  
2 // és csak azokra a route-okra enged, amit csak akkor lehet  
3 // látogatni, ha a felhasználó be van jelentkezve.  
4  
5  
6 module.exports = (req, res, next) => {  
7   if(req.isAuthenticated()){  
8     return next()  
9   }  
10  res.redirect("/")  
11  };
```

Middlewares >  checkNotAuthenticated.js > ...

```
1 // Ez a Middleware az EJS-ről érkező kéréseket vizsgálja,  
2 // és csak azokra a route-okra enged, amit csak akkor lehet  
3 // látogatni, ha a felhasználó nincs bejelentkezve.  
4  
5 module.exports = (req, res, next) => {  
6   if(req.isAuthenticated()){  
7     return res.redirect("/")  
8   }  
9   next()  
10  };
```

Middlewares >  restriction.js > ...

```
1 // Külsős alkalmazások engedélyezése  
2 const allowedOrigins = ['http://localhost:5173', 'http://127.0.0.1:5173',  
3 'https://goeventhungary.netlify.app',];  
4  
5 module.exports = (req, res, next) => {  
6   if(allowedOrigins.includes(req.headers.origin)) {  
7     return next();  
8   } else {  
9     return res.status(403).send('Hozzáférés megtagadva!');  
10  }  
11  };
```

Példa egy API korlátozásra, amit csak a rendszerünkben regisztrált app érhet el:

```
// Új esemény létrehozása (POST)  
router.post("/docs/newevent",restriction, async (req, res, next) => {  
  await AddNewEvent(connection)(req, res);  
});
```



### 4.2.6 ÚTVONALAK

Alábbi útvonalak biztosítanak interakciót a backend szerveren, EJS végponton:

/	Főoldal
/adatvedelem	Adatvédelmi oldal
/userpage	Felhasználói oldal
/adminpage	Adminisztrációs felület
/docs	Fejlesztői dokumentáció (API)

További API-s útvonalak, http kérések listája a 6.2 API-k használata fejezetben található.

### 4.2.7 BIZTONSÁG

Az oldalunkon bizalmasan kezeljük a felhasználók adatait, így mi nem tudjuk lekérdezni a jelszavát mert az a bcrypt segítségével hashként mentjük adatbázisunkba. Felhasználóink adatait pedig böngészőbe mentjük a kijelentkezésig.

```
// Ha az e-mail cím még nem szerepel az adatbázisban, akkor a felhasználói adatokat hozzáadjuk
const hashedPassword = await bcrypt.hash(req.body.password, 10);
connection.query(`INSERT INTO users (name, email, password, nationality, gender, birthdate, permission) VALUES (
  "${req.body.name}",
  "${req.body.email}",
  "${hashedPassword}",
  "${req.body.citizenship}",
  "${req.body.gender}",
  "${req.body.birthday}",
  "user"
)`, (err) => {
  if (err) {
    console.error('Hiba a regisztráció során: ' + err.stack);
    return res.redirect("/");
  }
});
```

5. ábra Jelszó titkosítás a bcrypt segítségével

Felhasználó által megadott jelszót egyből a átadjuk a hash függvénynek, ami 10-es szózással dolgozik ami már elegendő biztonságot ad annak feltörése ellen.

```
const user = result[0];
bcrypt.compare(password, user.password, (err, isMatch) => {
  if (err) {
    console.log(err);
    res.status(500).json({ message: "Hiba történt a jelszó ellenőrzése közben." });
  } else if (!isMatch) {
    // Hibás email, vagy jelszó
    res.status(401).json({ message: "Hibás email vagy jelszó.", email });
  } else {
    // Sikeres bejelentkezés, user adatainak elküldése a kliensnek
    res.status(200).json({ user });
  }
});
```

6. ábra Jelszó vizsgálat a compare függvénnyel

A fenti kódrészlet pedig az azonosítást mutatja be, a compare függvény összehasonlítja az általunk begépett jelszót, és az adatbázisban tárolt hash-hez viszonyítja, egyezés során pedig továbbítja a kliensoldalnak az adatainkat.



## 4.3 FRONTEND

### 4.3.1 NPM CSOMAGOK A FRONTENDBEN

Teljesség igénye nélkül, a frontendben is alkalmaztunk Npm csomagokat, melyek az alábbi funkciót biztosítják a webappunk számára:

CSOMAG NEVE	VERZIÓ	FELADATA
aos	^2.3.4	Animált megjelenést kölcsönöz görgetés során
bootstrap	^5.0.0	Bootstrap keretrendszer integrálását végzi.
jspdf	^2.5.1	Tartalmak konvertálását végzi pdf formátumba
axios	^1.3.3	Kérést küld a backend szerver felé
vue-router	^4.1.6	Útvonalak, végpontokat kezel.
vuex	^4.1.0	Globális változók alkalmazása/használatát biztosítja

### 4.3.2 KOMPONENSEK

Az átláthatóság érdekében, fontos a komponensek használata, hisz így az egyes tartalom blokkok szétválaszthatók, és a kód áttekintése is egyszerűbbé válik, így számos komponensre osztottuk fel az oldalt.

Az alábbi táblázatban, a komponens neve és tartalmának általános bemutatása található:

KOMPONENS NEVE	TARTALMA
about.vue	Rövid tájékoztatást ad az oldalról
access_denied.vue	Jogosultság hiányában e komponens tájékoztatja a felhasználót
actuallyevents.vue	Aktuális események blokkja
admin_menu.vue	Admin felületen lévő gombok + rögzítési form
admin_tables.vue	Admin felületen lévő táblázatok (Events, Users)
contactform.vue	Kapcsolati űrlap és elérhetőség
footerbar.vue	Weboldal lábléce, tartalmazza a weboldal fordítását más nyelvre
gallery.vue	A főoldalon lévő képek komponense
homescreen.vue	Kezdőképernyő, rajta a következő eseménnyel és visszaszámlálóval
login_signup.vue	Bejelentkezés / regisztrációs pop-up ablak
map.vue	Térkép szekció
refreshpassword.vue	Jelszó módosítása, a userpage oldalon
user_applicant.vue	Felhasználó által regisztrált események tartalma
user_datafield.vue	Felhasználó adatai

Ezen komponensek összesen 7 oldalon vannak alkalmazva.

### 4.3.3 ÚTVONALAK

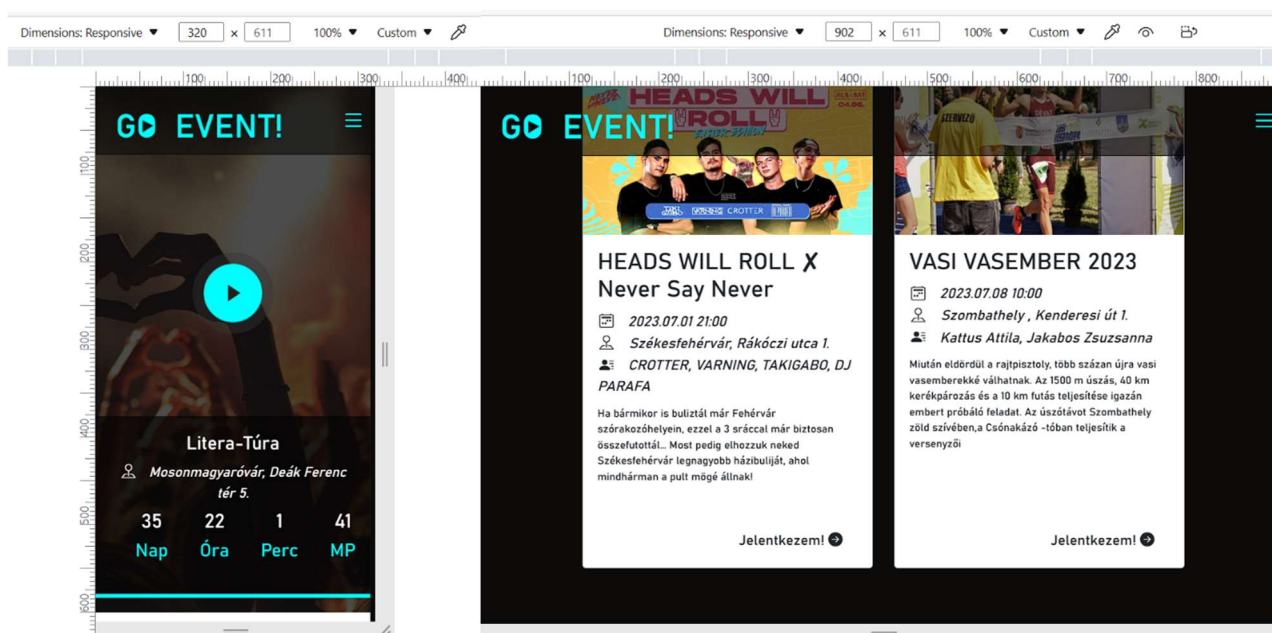
Útvonalakat, route-okat próbáltuk úgy összeállítani, hogy minél oldalváltással legyen használható az oldal, így az input mezők nagy részét Pop-up ablakban helyeztünk el, például a bejelentkezést, regisztrációt, jelszó módosítást vagy adminisztráció részről esemény, vagy admin rögzítést. A végpontokat Vue-router segítségével deklaráltuk a projektbe. Az alábbi route-ok léteznek a GO EVENT! felületén:

OLDAL NEVE	VÉGPONT	KORLÁTOZÁS
Home (Főoldal)	/	-
Adatvédelmi oldal	/adatvedelem	-
Fejlesztői dokumentum (API)	/docs	-
Felhasználói oldal	/userpage	Csak bejelentkezés esetén látogatható
Adminisztrációs oldal	/adminpage	Bejelentkezés, és admin szerepkör esetén

Ezen kívül vannak speciális tájékoztató oldalaink, amely akkor töltődnek be, ha nem létezik az oldal, vagy nincs hozzá jogunk meglátogatni, ilyen a „404 Nem található oldal!”, vagy a „Nincs jogosultsága megtekinteni” oldal.

### 4.3.4 RESZPONZIVITÁS

Kiemelt szerepet kapott a reszponzív design kialakítása a webalkalmazásunk fejlesztésében. A *Nemzeti média és hírközlési hatóság* által készült statisztikák azt mutatják, hogy a mobiltelefonok használata egyre inkább elterjedt, ezért nagyon fontos, hogy az alkalmazásunk ne csak asztali számítógépeken, hanem mobil eszközökön is jól működjön. A Bootstrap osztályok használatával sikerült biztosítani, hogy az alkalmazásunk optimálisan működjön mobiltelefonokon is, ezáltal elkerülve a felhasználók elvesztését és javítva az alkalmazásunk általános értékelését.



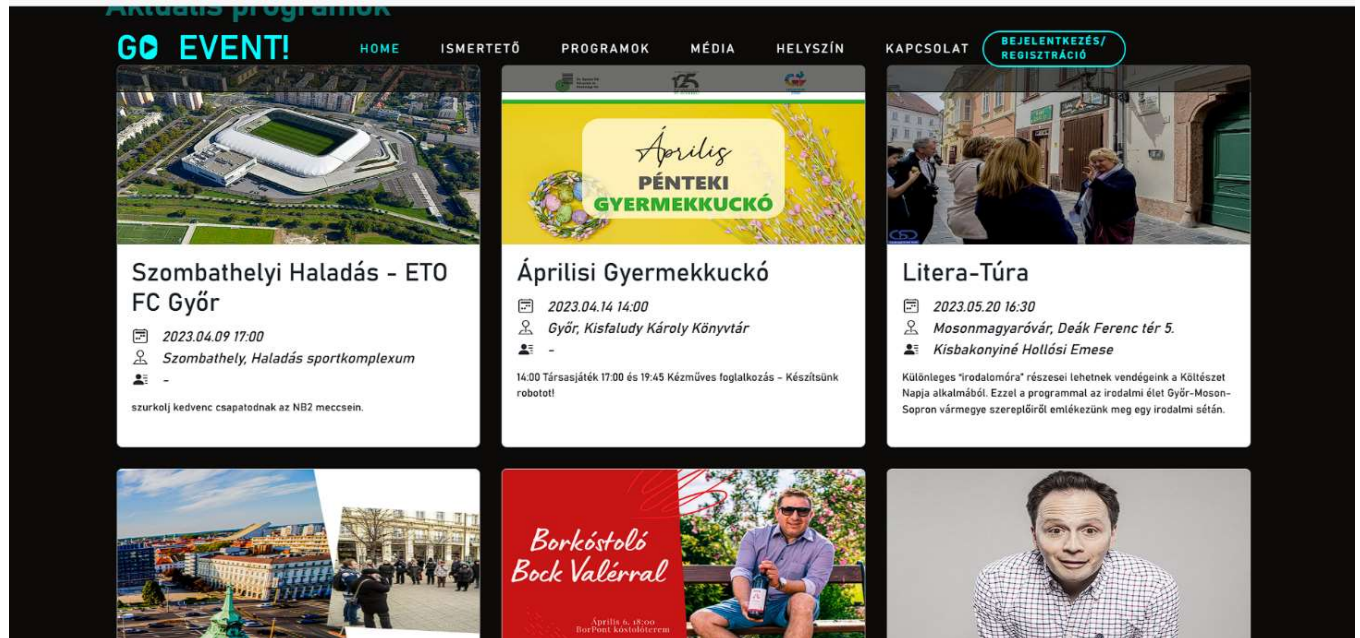
7. ábra Mobil megjelenés

8. ábra Programok megjelenése Tablet nézetben

## 5. FELHASZNÁLÓI KÉZIKÖNYV

### 5.1 OLDAL HASZNÁLATA BEJELENTKEZÉS NÉLKÜL

Az oldalon lehetővé tettük, a programok és események böngészését és azok szűrését kategóriánként.



9. ábra Bejelentkezés nélkül is megtekinthetőek a programok

Az üzenetküldési lehetőség segítségével egyszerűen és kényelmesen tudunk üzenetet küldeni az alkalmazás üzemeltetőinek. Nem szükséges külön e-mailt írni, csak ki kell tölteni a főoldal lap alján lévő kapcsolati űrlapot és elküldeni. Ezzel a funkcióval gyorsan és hatékonyan tudunk kapcsolatba lépni az alkalmazás üzemeltetőivel, ha bármilyen kérdésünk, vagy problémánk, javaslatunk van.

KÉRDÉSED VAN? VEDD FEL VELÜNK A KAPCSOLATOT!

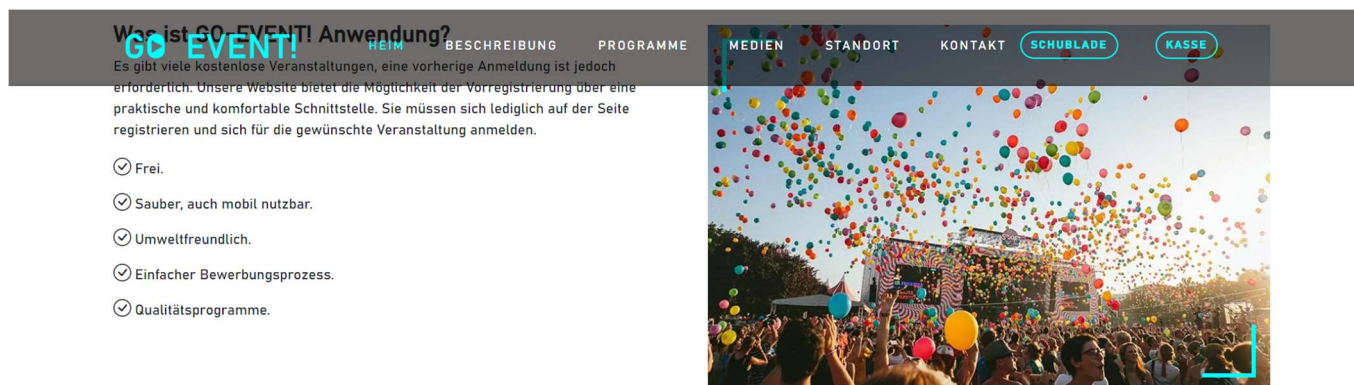
## Kapcsolat

KÜLDÉS

**GO EVENT!**  
9022  
GYŐR  
Szent István út 1.  
Telefonszám: +36 1 511 511  
E-mail: goeventhungary@gmail.com

10. ábra Kapcsolati űrlap

Az oldalunk alján elérhető egy fordító modul, amely segítségével könnyedén átállítható az oldal nyelve a felhasználó igényeinek megfelelően. A Google fordító rendszerével történő integráció révén a weboldal szövege azonnal átalakul a kiválasztott nyelvre.



11. ábra App német nyelven

Elérhető oldalak bejelentkezés nélkül: Home, Adatvédelmi oldal, Fejlesztői dokumentáció

## 5.2 BEJELENTKEZÉS ÁLTALÁNOS FELHASZNÁLÓI FIÓKKAL

A bejelentkezés esetén a programokon túl leadhatjuk jelentkezési igényünket, ennek folyamata a következő:

- 1. lépés:** Kattintson a Bejelentkezés/Regisztráció gombra a menüben
- 2. lépés:** Válassza ki, hogy belépne, vagy regisztrálna oldalunkra

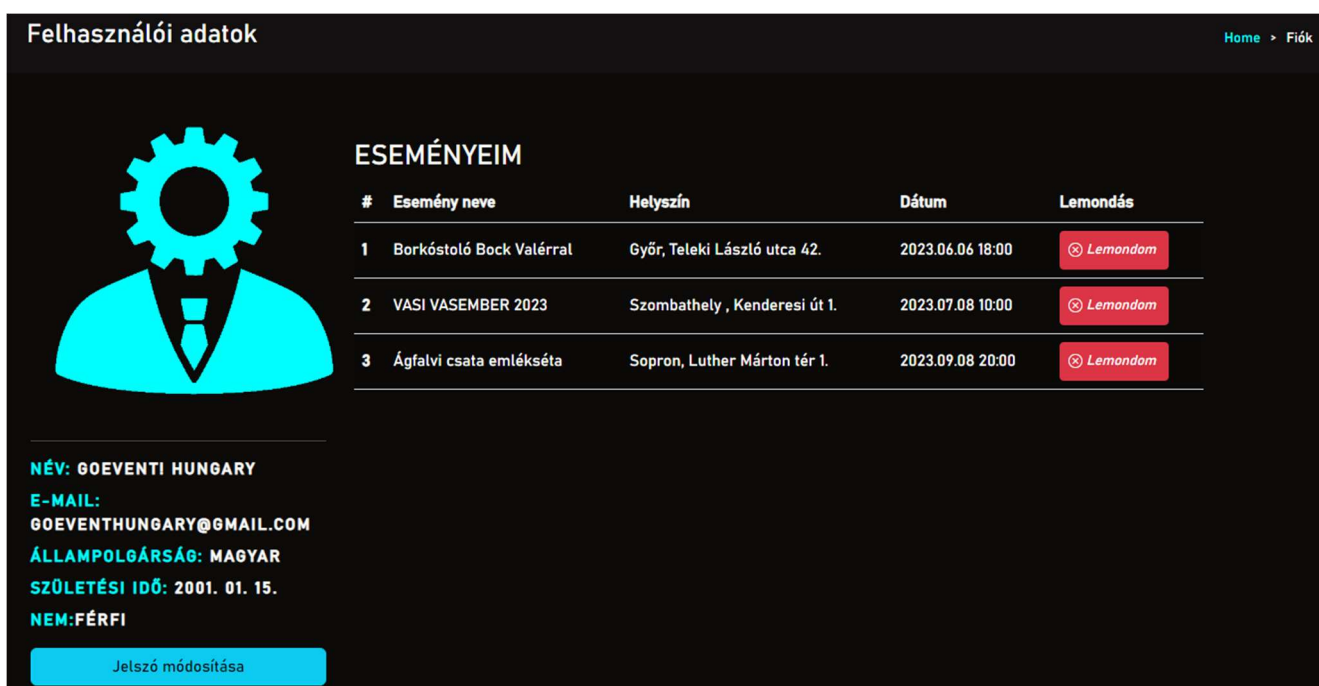
12. ábra Pop-up ablak, ahol regisztrálni, és bejelentkezni lehet oldalunkra

3. **lépés:** A megfelelő adatok bevitele során megtörténik a bejelentkezés, és új gombok jelennek meg a menüben, valamint az eseményeknél aktiválódik a „Jelentkezem” gomb, ahol jelentkezhetünk kedvenc programjainkra.



13. ábra Bejelentkezést követően lehet elérni egyes funkciókat

4. **lépés:** Amennyiben módosítani szeretnénk jelszónkat, vagy le szeretnénk mondani programjainkat, akkor azt a „Fiók” menüpontban tehetjük meg.



14. ábra Felhasználói fiók megjelenése

A megfelelő esemény sorában lévő „Lemondom” gombra kattintva jelentkezési szándékunk törlődik, így más személy jelentkezhet az eseményre.



További lehetőségként jelenik meg ezen oldalon a jelszó módosítása, ami akkor lehet hasznos, ha biztonsági okból változtatni szeretné belépési jelszavát. Végül itt tudjuk törölni a fiókunkat is véglegesen.

Az alkalmazásból való kijelentkezés során az adatok törlődnek a böngészőből, ezért legközelebb újra be kell jelentkezni.

### 5.3 ADMINISZTRÁTORI HOZZÁFÉRÉS

Adminisztrátori fiók megjelenése hasonló, ugyanúgy tudunk jelentkezni eseményekre, és azokat le is tudjuk mondani, azonban hozzáférünk az összes esemény és felhasználó adataihoz az /adminpage oldalon, melyet csak akkor érhetünk el, ha minket úgy regisztráltak szintén admin fiókból.

Adminisztráció						
<div>    Felhasználók listája +Új esemény/admin </div> <div>Home &gt; Adminisztrációs felület</div>						
#	ESEMÉNY NEVE	IDŐPONT	HELYSZÍN	JK. SZÁMA	KORHATÁR	MŰVELETEK
1	Litera-Túra	2023. 05. 20. 18:30	Mosonmagyaróvár, Deák Ferenc tér 5.	130 / 1 (1%)	16+	
2	RÁBARANGOLÁS	2023. 05. 31. 19:00	Győr, Baross út, Csónakos szobor	120 / 0	16+	
3	Fermentálás workshop	2023. 06. 03. 19:00	Fertőszéplak, Petőfi Sándor utca 1/A.	10 / 0	-	
4	Borkóstoló Bock Valérral	2023. 06. 06. 20:00	Győr, Teleki László utca 42.	400 / 2 (1%)	18+	
5	Janklovics Péter önálló	2023. 06. 17. 20:00	Mosonmagyaróvár, Erkel Ferenc utca 14.	450 / 1 (0%)	18+	
6	HEADS WILL ROLL X Never Say Never	2023. 07. 01. 23:00	Székesfehérvár, Rákóczi utca 1.	1200 / 1 (0%)	16+	
7	VASI VASEMBER 2023	2023. 07. 08. 12:00	Szombathely, Kenderesi út 1.	3000 / 0	-	
8	A gyáva c. színdarab	2023. 07. 10. 19:00	Szombathely, Ady tér 3.	200 / 0	16+	
9	Ágfalvi csata emlékséta	2023. 09. 08. 22:00	Sopron, Luther Márton tér 1.	70 / 3 (4%)	-	
10	Winter Fest	2024. 11. 30. 20:00	Dunaújváros, Kossuth utca 123.	1000 /	16+	

15. ábra Programok listája az admin felületen

Alapértelmezetten az események fő adatait látjuk, de a fenti „admin menü” -ben könnyen tudunk váltani, hogy a felhasználók adatait is meg tudjuk tekinteni.

Adminisztráció								
<div>    Események listája +Új esemény/admin </div> <div>Home &gt; Adminisztrációs felület</div>								
#	NÉV	EMAIL CÍM	ÁLLAMPOLGÁRSÁG	NEM	SZÜLETÉSI IDŐ	ÉLETKOR	JOGOSULTSÁG	TÖRLÉS
1	FEKETE KLAUDIA	feketeklaudia01@gmail.com	Magyar	Nő	2007. 04. 15.	16 éves	user	
2	GOEVENT! HUNGARY	goeventhungary@gmail.com	Magyar	Férfi	2001. 01. 15.	22 éves	admin	
3	DERNÓCZI BALÁZS	dernoczi.balazs@goevent.hu	Magyar	Férfi	2001. 10. 20.	22 éves	admin	
4	KISS MAJA	kissmaja2005@gmail.com	Magyar	Nő	2005. 12. 09.	18 éves	user	
5	FERENCZY KATA	ferenczykata@smr.hu	Magyar	Nő	2000. 06. 22.	23 éves	user	
6	SOÓS LIZA	soosliza0712@freemail.hu	Magyar	Nő	2004. 07. 12.	19 éves	user	
7	KOVÁCS PÉTER	kvts.peter92@gmail.com	Magyar	Férfi	1992. 10. 04.	31 éves	user	
8	LUKAS NEUMÜLLER	lukasneumuller2000@gmx.at	Osztrák	Férfi	2000. 11. 15.	23 éves	user	

16. ábra Felhasználók listája

Ezeket az adatokat a fenti gombok segítségével exportálni tudjuk, valamint tudjuk frissíteni is. Továbbá a megfelelő sor összetartozó adatait tudjuk törölni az adatbázisból.

A megfelelő adatok megadásával az esemény vagy az adminisztrátor rögzítése megtörténik, az esemény a főoldalon megjelenik, az új adminisztrátor pedig hozzáférhet a fiókhöz.

Végül az utolsó gombra kattintva („+ Új esemény/ admin”) tudunk új eseményt, vagy adminisztrátort rögzíteni a GO EVENT! rendszerébe, utóbbi felvétele nagyban hasonlít az általános regisztrációhoz, azonban itt a rendszer admin jogot biztosít a bejelentkezés során.

×

ÚJ ESEMÉNY LÉTREHOZÁSA

ÚJ ADMINISZTRÁTOR HOZZÁADÁSA

Esemény neve

Város 

Utca / hely neve

Hsz.

Maximális létszám

Fellépők nevei vesszővel tagolva

Esemény leírása

Adja meg az URL címét az illusztrációs képnek

Zene (koncert)

Korhatár nélküli

ESEMÉNY IDEJE:

éééé. hh. nn. --:--

Rögzítés

×

ÚJ ESEMÉNY LÉTREHOZÁSA

ÚJ ADMINISZTRÁTOR HOZZÁADÁSA

Név

Email cím

Jelszó

Jelszó megegyeszer

Magyar

☒ FÉRFI
☐ NŐ
☐ NEM ADOM MEG

SZÜLETÉSI IDŐ:

éééé. hh. nn.

Admin felvétele

17. ábra Rögzítés menüpont az admin felületen

## 6. FEJLESZTŐI DOKUMENTUM

### 6.1 FEJLESZTŐI KÖRNYEZET FUTTATÁSA

Az általunk megküldött csomag két részre bontható, az egyik tartalmazza a localhost, azaz a helyi futtatáshoz alkalmas csomagot, valamint az internetre publikált verzió tartalmát, ami kis mértékben tér el a másiktól.

Az alábbi módon tudjuk a GO EVENT! felületeit megnyitni:

#### ADATBÁZIS

1. lépés: Mysql server indítása DbForge, vagy Xampp segítségével. (mi Xampp-al dolgoztunk)
2. lépés: Szükséges dump futtatása / alkalmazása (*Goevent\_app\_db\_dump*)

#### BACKEND (Szerveroldal)

3. lépés: Visual studio code megnyitása
4. lépés: **node server.js**, majd ctrl + kattintás a linkre. ( <http://localhost:5172/> )

#### FRONTEND (Kliensoldal)

5. lépés: Visual studio code-ban kliens server mappa tallózása ( *File>New window > Frontend* )
6. lépés: Terminal ablakban (ctrl+ö) **npm i**, majd **npm run dev** parancs alkalmazása
7. lépés: URL megnyitása ctrl + kattintással ( <http://localhost:5173/> )

Funkciók tesztelésére szánt user fiókok:

SZEREPKÖR	EMAIL CÍM	JELSZÓ
Admin jogú fiók	goeventhungary@gmail.com	goevent_ABR
User jogú fiók	soosliza0712@freemail.hu	soosliza0712

*valamennyi usernél az adatbázisban az a logika, hogy a @ jel előtti szöveg a jelszó.*

### 6.2 API-K HASZNÁLATA

Alkalmazásunk biztosítja, hogy külsős alkalmazások hozzáférjenek az adatbázis bizonyos adataihoz, ezt két kategóriára bonthatjuk, az adatok érzékenysége szerint:

#### 1. Alapadatok biztosítása

Itt azon adatok érhetők el, melyek bejelentkezés nélkül is elérhetőek, így biztosítva, hogy programjainkat közzé tudják tenni más platformokon.

#### 2. Bizalmas adatok, és interakciók biztosítása

Itt jellemzően olyan külsős partnerek kapnak hozzáférést, amelyek együttműködnek a projektben, és céljuk, hogy integráltan együttműködjenek a GO EVENT! -tel és eseményeket vagy felhasználókat regisztráltassanak saját oldalukon a GO EVENT! rendszerében, mint egy Franchise hálózat, más oldalakról érkező adatok kerülnek fel egy adatbázisban, mégis a weboldal kinézete, a partner döntése maradhat.



### 6.2.1 GET

#### NextEventContent()

- Feladata: Lekérdezhetjük a következő eseményt
- Válasz: Esemény neve, helyszíne és ideje
- Jogosultság: Korlátozás nélkül használható
- Elérése: docs/nextevent

#### AllEvents()

- Feladata: Az összes esemény kilistázása
- Válasz: events tömb
- Jogosultság: Korlátozás nélkül használható
- Elérése: docs/allevnts

#### eventsByAge()

- Feladata: A lekérdezhetjük a jövőbeli eseményeket korhatár szerint
- Válasz: visszaadja a keresett érték objektumát
- Jogosultság: Korlátozás nélkül használható
- Elérése: docs/eventagelimit/:agelimit

#### eventsByCategories()

- Feladata: A lekérdezhetjük a jövőbeli eseményeket kategóriájuk szerint
- Válasz: Visszaadja a keresett érték objektumát
- Jogosultság: Korlátozás nélkül használható
- Elérése: docs/eventcategory/:category

#### eventPass()

- Feladata: Ellenőrizhetjük, hogy az belépőkód érvényes-e, és hogy kihez tartozik
- Válasz: Visszaadja azon fontos adatokat, amiből meghatározható a jogosultság
- Jogosultság: Csak rögzített partner használhatja ezen funkciót
- Elérése: docs/eventpass/:pass\_code

#### getUsers()

- Feladata: Megkaphatjuk az összes regisztrált tag adatát
- Válasz: users tömb
- Jogosultság: Csak rögzített partner használhatja ezen funkciót
- Elérése: docs/allusers

#### getAppliedEvents()

- Feladata: Lekérdezhetjük a felhasználó regisztrált eseményeit ID-ja alapján.
- Válasz: Visszaadja a keresett érték objektumát
- Jogosultság: Csak rögzített partner használhatja ezen funkciót
- Elérése: docs/userapplied/:user\_id

**ArchivedEvents()**

- Feladata: Lekérdezés pillanatához viszonyított korábbi események kilistázása
- Válasz: Visszaadja a keresett érték objektumát
- Jogosultság: Korlátozás nélkül használható
- Elérése docs/archive

**6.2.2 POST****newevent()**

- Feladata: Új eseményt tudunk rögzíteni
- Jogosultság: Csak rögzített partner használhatja ezen funkciót
- Elérése docs/newevent

**signup()**

- Feladata: Új általános hozzáférésű fiók regisztrálása
- Jogosultság: Korlátozás nélkül használható
- Elérése docs/signup

**newadmin()**

- Feladata: Új adminisztrátor rögzítése
- Jogosultság: Csak rögzített partner használhatja ezen funkciót
- Elérése docs/newadmin

**sendForm()**

- Feladata: Kapcsolati űrlap adatainak továbbítása
- Jogosultság: Korlátozás nélkül használható
- Elérése docs/sendForm

**applyToLocation()**

- Feladata: ID alapján jelentkezés egy eseményre
- Jogosultság: Csak rögzített partner használhatja ezen funkciót
- Elérése docs/applyToLocation

**cancelApplication()**

- Feladata: Eseményre való jelentkezés törlése
- Jogosultság: Csak rögzített partner használhatja ezen funkciót
- Elérése docs/newevent

**login()**

- Feladata: Felhasználó azonosítása, és adatainak elküldése
- Jogosultság: Csak rögzített partner használhatja ezen funkciót
- Elérése docs/login

### 6.2.3 PUT

#### **forgotPassword()**

- Feladata: Elfelejtett jelszó esetén új jelszó generálása, és küldése emailben.
- Jogosultság: Csak rögzített partner használhatja ezen funkciót
- Elérése docs/forgotpassword

#### **changePassword()**

- Feladata: Általunk ismert jelszó frissítése
- Jogosultság: Csak rögzített partner használhatja ezen funkciót
- Elérése docs/refreshpassword

### 6.2.4 DELETE

#### **deleteEvent()**

- Feladata: Esemény törlése ID-ja alapján.
- Jogosultság: Csak rögzített partner használhatja ezen funkciót
- Elérése docs/deleteEvent/:id

#### **deleteUser()**

- Feladata: Felhasználó törlése ID-ja alapján.
- Jogosultság: Csak rögzített partner használhatja ezen funkciót
- Elérése: docs/deleteUser/:id

GO EVENT! - API Service 3.0.0 OAS3

## User process

GET	/docs/allusers	Összes felhasználó	▼
GET	/docs/userapplied/{user_id}	Felhasználók regisztrált eseményei	▼
POST	/docs/sendForm	Kapcsolatfelvételi űrlap küldése	▼
DELETE	/docs/deleteUser/{id}	Felhasználó törlése	▼

## Event informations

GET	/docs/allevnts	Összes esemény	▼
GET	/docs/nexthevent	Következő esemény	▼
GET	/docs/eventcategory/{categories}	Az események kategóriák szerinti szűrése	▼
GET	/docs/archive	Archívum	▼
GET	/docs/eventagelimit/{agelimit}	Az események korhatár szerinti szűrése	▼
GET	/docs/eventpass/{pass_code}	Belépőkód ellenőrzés	▼

## Authentication

POST	/docs/signup	Regisztráció	▼
POST	/docs/login	Bejelentkezés	▼ 🔒
PUT	/docs/forgotpassword	Elfelejtett jelszó	▼
PUT	/docs/refreshPassword	Jelszóváltogatás	▼

## Admin process

POST	/docs/newadmin	Admin regisztráció	▼ 🔒
POST	/docs/newevent	Új esemény létrehozása	▼ 🔒
DELETE	/docs/deleteEvent/{id}	Esemény törlése	▼

## Apply / Cancel application

POST	/docs/applyToLocation	Jelentkezés egy eseményre	▼
POST	/docs/cancelApplication	Visszamondás	▼ 🔒

## Schemas

Eventproperties >
ForgotPassword >
ChangePassword >
ArchivedEvent >

Bővebb információért kattintson az alábbi linkre:

[Swagger UI \(GO EVENT! App\)](#)

## 7. PUBLIKÁLÁS, FEJLESZTÉSI LEHETŐSÉGEK

Utolsó hónapban, sikerült a webes appjainkat az interneten közzétenni, így bárki elérheti ezeket. Tapasztalatunk az, hogy az ingyenes tárhelyek szerverei lassabbak, és picit a felhasználói élmény rovására ment. A „deployolás” különböző oldalakon történt, backend részről a *cyclic.sh* nevű oldal tűnt megbízhatónak, frontend téren pedig a *netlify.com* oldalon lett nyilvánosságra hozva. Az adataink kezdetekben a *freedb.tech* oldalon, majd később az iskolánktól kapott adatbázis szerveren vannak eltárolva.

- **Backend szerver:** <https://goeventapiservice.cyclic.app/>
- **Frontend szerver:** <https://goeventhungary.netlify.app/>

A publikálás után az alábbi lehetőségek merültek fel a projekt folytatásaként:

- A fellépők menedzselése modul lehetővé tenné az esemény szervezőinek, hogy könnyen és hatékonyan kezeljék a fellépőket. Ez a modul tartalmazhatna olyan funkciókat, mint például a fellépők adatainak kezelése, a fellépési idők beosztását, a felmerülő költségek nyomon követése és a szerződések kezelését.
- Egy mobilalkalmazás fejlesztése a webes apphoz számos előnnyel járna, amelyek számottevően javíthatják a felhasználói élményt és fokozhatja az alkalmazás hatékonyságát. Az emailben kapott QR-kódok beolvasása lehetővé tenné a felhasználók számára, hogy egyszerűbben és biztonságosabban jussanak be az eseményekre, és így könnyebbé válna a belépés. Ez különösen fontos a COVID-19 világjárvány idején, amikor a kontaktmentesség és az egészség védelme kiemelten fontos. Az ilyen típusú alkalmazások iránt egyre nagyobb az igény, és jelentős előnyökkel járnak a rendezvényszervezők és a felhasználók számára egyaránt.
- Egy térképnézet segítségével a felhasználók láthatnák az események helyszíneit. Ez jelentősen javíthatja az alkalmazás használatának kényelmét, mivel a felhasználók azonnal megtalálhatják a legközelebbi eseményeket, amelyek érdekelhetik őket.
- A Google fiókos bejelentkezés lehetősége hasznos lehet azoknak a felhasználóknak, akik már rendelkeznek Google-fiókkal. Ez kényelmes és egyszerű lehetőséget kínál a felhasználók számára, hogy belépjenek az alkalmazásba, és elkerüli a szükségtelen jelszó-emlékeztetők használatát.
- Az online fizetési lehetőségek növekvő népszerűsége miatt, a bankkártyás fizetési mód bevezetése az alkalmazásban lehetővé tenné a felhasználók számára, hogy egyszerűen és biztonságosan vásároljanak belépőket és egyéb termékeket az alkalmazásban. Ez jelentősen javíthatja a felhasználói élményt, és növelheti az alkalmazás által kezelt tranzakciók számát.

Összességében, az új funkciók bevezetése jelentős előnyöket és segítséget kínálnának a felhasználók számára, valamint hozzájárulna az alkalmazás sikeresebbé tételében.

## 8. FORRÁSOK

**Asaolu Elijah, D. L.** (2022. 10 04). <https://stackabuse.com/integrating-mysql-with-nodejs-applications/>.

**Bosco, T.** (2023. 03 14). <https://morioh.com/p/8d60d900f824>.

**Kantar-Hoffmann kutatócég.** (2023. 02 02).

[https://nmhh.hu/cikk/235951/Internethasznalati\\_szokasok\\_digitalis\\_media\\_es\\_tartalomfogyasztas](https://nmhh.hu/cikk/235951/Internethasznalati_szokasok_digitalis_media_es_tartalomfogyasztas). Nmhh.

**Netlify.** (2023. 04 09). <https://docs.netlify.com/get-started/>.

**npmjs.com.** (2022. 11 09). <https://www.npmjs.com/package/bcrypt>.

**SZTE TTIK.** (2023. 03 05). <http://www.inf.u-szeged.hu/~tarib/javascript/vuejs.html#az-elseo-vue-js-alkalmazasunk>.

**Vuejs.org.** (2023. 02 27). <https://vuex.vuejs.org/>.

**Wikipédia.** (2023. 04 11). <https://hu.wikipedia.org/wiki/XAMPP>.

Köszönjük, a győri Jedlik Ányos gépipari és informatikai technikum és kollégium tanárainak a rendelkezésre állását, és segítőkészségét, kiemelve Bólya Gábor tanár úr támogatását!