**SAPIENTIA ERDÉLYI MAGYAR TUDOMÁNYEGYETEM**

**CSÍKSZEREDAI KAR**

**GAZDASÁGI INFORMATIKA SZAK**

**DIPLOMADOLGOZAT**

**ESEMÉNY LÉTREHOZÓ WEBes applikáció  
(Lecrevent)**

**Végzős hallgató:**

**Vántsa Alpár**

**Témavezető:**

**Dr. Pál László, egyetemi docens**

**2023**

**Román kivonat**

**Creator de evenimente aplicație web**

Fusce a metus eu diam varius congue nec nec sapien. Vestibulum orci tortor, sollicitudin ac euismod non, placerat ac augue. Nunc convallis accumsan justo. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Donec malesuada vehicula lectus, viverra sodales ipsum gravida nec. Integer gravida nisi ut magna mollis molestie. Nullam pharetra accumsan sagittis. Proin tristique rhoncus orci, eget vulputate nisi sollicitudin et. Quisque lacus augue, mollis non mollis et, ullamcorper in purus. Morbi et sem orci. Praesent accumsan odio in ante ullamcorper id pellentesque mauris rhoncus. Duis vitae neque dolor.

Duis sed purus at eros bibendum cursus nec a nulla. Donec turpis quam, ultricies id pretium sit amet, gravida eget leo.

Fusce a metus eu diam varius congue nec nec sapien. Vestibulum orci tortor, sollicitudin ac euismod non, placerat ac augue. Nunc convallis accumsan justo. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Donec malesuada vehicula lectus, viverra sodales ipsum gravida nec. Integer gravida nisi ut magna mollis molestie. Nullam pharetra accumsan sagittis. Proin tristique rhoncus orci, eget vulputate nisi sollicitudin et. Quisque lacus augue, mollis non mollis et, ullamcorper in purus. Morbi et sem orci. Praesent accumsan odio in ante ullamcorper id pellentesque mauris rhoncus. Duis vitae neque dolor.

Duis sed purus at eros bibendum cursus nec a nulla. Donec turpis quam, ultricies id pretium sit amet, gravida eget leo.

Fusce a metus eu diam varius congue nec nec sapien. Vestibulum orci tortor, sollicitudin ac euismod non, placerat ac augue. Nunc convallis accumsan justo. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Donec malesuada vehicula lectus, viverra sodales ipsum gravida nec. Integer gravida nisi ut magna mollis molestie. Nullam pharetra accumsan sagittis. Proin tristique rhoncus orci, eget vulputate nisi sollicitudin et. Quisque lacus augue, mollis non mollis et, ullamcorper in purus.

**Angol kivonat**

**Event creator web application**

Fusce a metus eu diam varius congue nec nec sapien. Vestibulum orci tortor, sollicitudin ac euismod non, placerat ac augue. Nunc convallis accumsan justo. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Donec malesuada vehicula lectus, viverra sodales ipsum gravida nec. Integer gravida nisi ut magna mollis molestie. Nullam pharetra accumsan sagittis. Proin tristique rhoncus orci, eget vulputate nisi sollicitudin et. Quisque lacus augue, mollis non mollis et, ullamcorper in purus. Morbi et sem orci. Praesent accumsan odio in ante ullamcorper id pellentesque mauris rhoncus. Duis vitae neque dolor.

Duis sed purus at eros bibendum cursus nec a nulla. Donec turpis quam, ultricies id pretium sit amet, gravida eget leo.

**Tartalomjegyzék**

**[1.](#_Toc534978338)****[Bevezető](#_Toc534978338)** [5](#_Toc534978338)

**[2.](#_Toc534978339)****[Használt technológiák](#_Toc534978339)** [6](#_Toc534978339)

**[2.1. ASP NET Core](#_Toc534978340)** [6](#_Toc534978340)

**[2.2.](#_Toc534978341)****[Javascript - Vue.js - Vuetify](#_Toc534978341)** [6](#_Toc534978341)

**[3.](#_Toc534978342)****[Alkalmazás bemutatása](#_Toc534978342)** [7](#_Toc534978342)

**[3.1.](#_Toc534978343)****[Programozási környezet](#_Toc534978343)** [7](#_Toc534978343)

**[3.2.](#_Toc534978344)****[Adatbázis](#_Toc534978344)** [7](#_Toc534978344)

**[3.3.](#_Toc534978345)****[Az alkalmazás fontosabb funkciói](#_Toc534978345)** [8](#_Toc534978345)

**[3.3.1.](#_Toc534978346)****[Login](#_Toc534978346)** [8](#_Toc534978346)

**[3.3.2.](#_Toc534978347)****[Regisztráció](#_Toc534978347)** [9](#_Toc534978347)

**[4.](#_Toc534978348)****[Következtetések](#_Toc534978348)** [9](#_Toc534978348)

**[Irodalomjegyzék](#_Toc534978349)** [10](#_Toc534978349)

# **Bevezető**

Székelyföld sokszínű és változatos, ez nem csak az emberekre, a környezetre és a hétköznapi életre mondható el, hanem a sportra is. Az itt élő emberek igencsak változatosak, a földrajzi helyzetnek köszönhetően, sokféle sportolási lehetőség adott. A sportnak kitűnő összekovácsoló ereje van, összehozza az embereket, új barátságok szövődnek, emellett egészségen és fitten tarthatjuk magukat. Az online tér egy kítűnő hely, ahol az emberek megismerkedhetnek, összekapcsolódhatnak, az alkalmazásom segítségével az online térben lehetőséget kapnak a felsználók egymás megismerésére, majd ezt követően a való életben találkozhatnak és érezhetik jól magukat.

Projektem legfőbb célja, hogy összehozza az embereket a különböző események által és akár életre szóló barátságokat, élményeket szerezzenek a webes applikáció segítségével. Noha ott van a Facebook, Instagram és még számtalan applikáció amely által a felhasználók értesülhetnek különböző eseményekről, de minden ilyen és az elöbb említett applikációból hiányzik valami, vagy éppenséggel átláthatatlan a felület. Éppen ezért készítettem egy olyan applikációt, amely kizárólag csak az eseményekre összpontosít, emellett pedig egyszerű és átlátható felhasználóbarát felülettel rendelkezik amellyel rengeteg idő megspórolható. Más szempontból mindenkivel előfordult már, hogy a barátai valami miatt éppen nem tudtak részt venni az aznapi focizáson, vagy egyéb közös programon. Az alkalmazás az ilyen esetekre is remek megoldás lehet, hiszen egy esemény létrehozásával máris lehetőséget adunk valakinek, hogy csatlakozzon és ezáltal pótolja a hiányzó helyet.  
 A felhasználóknak egy egyszerű regisztrációt követően lehetőségük van életkor illetve távolság szerint szűrniük a különböző eseményeket, ezáltal a felhasználót nem fogják “zavarni” az olyan események amelyekre amúgy sem lenne lehetősége eljutni, vagy esetlegesen nem az általa preferált korosztály vesz részt az eseményen. Az esemény létrehozásakor a felhasználónak lehetősége van beállítani a szabad helyek számát, ezáltal elkerülhetőek a különböző kellemetlenségek a jövőben.

A dolgozatom készítése során rengeteg új dolgot sikerült megtanulnom, emellett a tanult dolgokat is sikerült jobban elmélyítenem. Ide sorolnám a C# ASP NET Core-t [1] amely által a backendet hoztam létre, illetve a Vue.js [2] keretrendszert amely jelentősen megkönnyítette a felhasználói felület elkészítését. Ezen modern technológiák segítségével egy modern és gyors weboldalt sikerült létrehoznom, amely minden korosztálynak megfelelő segítséget nyújthat új emberek megismerésében és egy jobb, egészségesebb élet kialakításában.

# **Használt technológiák**

A weboldal létrehozásához különböző technológiákat használtam. A front-end, amely az applikáció azon része amellyel a felhasználó közvetlenül találkozik. Ennek alapját a Vue.js Javascript [2] keretrendszer alkotja, melyhez társítottam a Vuetify-t [3] a Vue.js legismertebb felületi keretrendszeré. A felhasználói felület tervezéséhez a Figma tervező programját használtam. A weboldal back-end része, ami egyben a rendszer müködtetője is, ASP NET Core [1] rendszerben született meg. Emellett még használtam, de nem utolsó sorban XAMPP-t [4] amely segítségével kapcsolat jött létre az API illetve az adatbázis között, illetve a Github-nak [5] köszönhetően folyamatos verziókövetést használtam a fejlesztés során, ezáltal a későbbi hibákat egyszerűbben tudtam kijavítani.

## **ASP NET Core**

A weboldalam fejlesztésében az ASP NET Core Web API 7 [1] modelljét használtam, ami akkor még a legújabb verziónak számított, jelenleg a 8-as verzió mondható a legújabbnak. Ez egy keretrendszer az ASP NET Core-hoz [1], amely lehetővé tette a kommunikációt mint a klins, mint a szerver oldallal HTTP protokollt használva. Rengeteg beépített funkciója van, mint pl. a hitelesítő, de session-öket illetve automatikus adatbáziskapcsolódást is lehetővé teszi.

Az ASP NET Core MVC [6] keretrendszer segítségével az alkalmazás ezen részén is átláthatóbb, könnyebben érthető kód jön létre. Az MVC a Model-View-Controller (modell - nézet - vezérlő ) komponensekből áll. A modellben tárolódnak és kezelődnek az adatok, amelyek lehetnek file-okban tárolt adatok, vagy adatbázisból származó adatok. A nézetnek az a szerepe, hogy megjelenítse a különböző adatokat, illetve biztosítja az alkalmazás felhasználói felületét amellyel a felhasználó közvetlenül kapcsolatba kerül, itt jelennek meg az adatok és itt lép interakcióba a felhasználó a weboldallal. A vezérlő összekapcsolja a nézetet és a modellt. Legfőbb feladata a válaszok visszaküldése és a kérések kezelése.

Ahhoz, hogy kapcsolatot jöjjön létre a C# és MySql között, a C# Entity Framework [7] keretrendszert használtam. A keretrendszernek köszönhetően zökkenőmentesen kapcsolatot tudtam teremteni a MySql és az .Net applikációm között. Előnye az, hogy a kódban végzett változtatások automatikus változást hoznak az adatbázisban is egyaránt. További előnynek mondható, hogy a verziókövetésre van lehetőség, ezáltal könnyen lekövethető az adatbázisban történt változások.

A felhasználó adatai megörzése érdekében Json Web Tokeneket (JWT) [8] használtam. Ezeknek a tokeneknek a felhasználók azonosításában van szerepük, ezáltal a rendszer megjegyzi egy a már előre bejelentkezett felhasználó adatait. Egy token 3 részből tevődik össze, ezen típusok a következők: fejléc (header) amely információt tárol a token típusáról (typ) és az algoritmusról (alg), adatok (payload) itt azon adatok szerepelnek, amelyeket előre meghatározva a backend-en beállítottam pl. felhasználónév, email cím, életkor, stb.. Az utolsó típusa az aláírás (signature) az algoritmus alkalmazásával keletkezik egy kód, amelyet tárol és ennek a segítségével ellenőri a token valódiságát.

## **Javascript - Vue.js - Vuetify**

A Javascript egy objektumorientált gyengén típusú programozási nyelv, amely napjaink egyik elengedhetetlen kelléke a webfejlesztések során. Általa a fejlesztők dinamikus és interaktív weboldalakat hozhatnak létre, emellett különböző funkciókat, adatkezelést vagy akár animációkat is hozzáaadhatnak a weboldalhoz, amelyek megkérdőjelezhetetlenül növelik a felhasználói élményt. A Javascript kódja futtatható a böngészőben, illetve NodeJs [9] segítségével akár a saját operációs rendszerünken is futtathatóak a scriptek.

A Vue.js [2] a JavaScript egyik legismertebb keretrendszere. Legfőbb feladata, hogy megkönnyítse a fejlesztő dolgát, mivel a natív JavaScript-ben leprogramozni különféle feladatokat jelentősen több időt vesznek igénybe. A fejlesztés során bármely változás a kódban, valós idejű változásként jelenik meg a fejlesztő képernyőjén, amely igencsak megkönnyíti a fejlesztést, ez annak köszönhető, hogy a Vue.js [2] virtuális DOM-ot használ. A Route-oknak köszönhetően több oldalas weboldalakat is lehetőségünk van létrehozni, a komponensek segítségével pedig olyan elemeket hozhatunk létre, amelyek előreláthatólag tudunk, hogy többszőr is felfogunk használni (pl. header, footer).

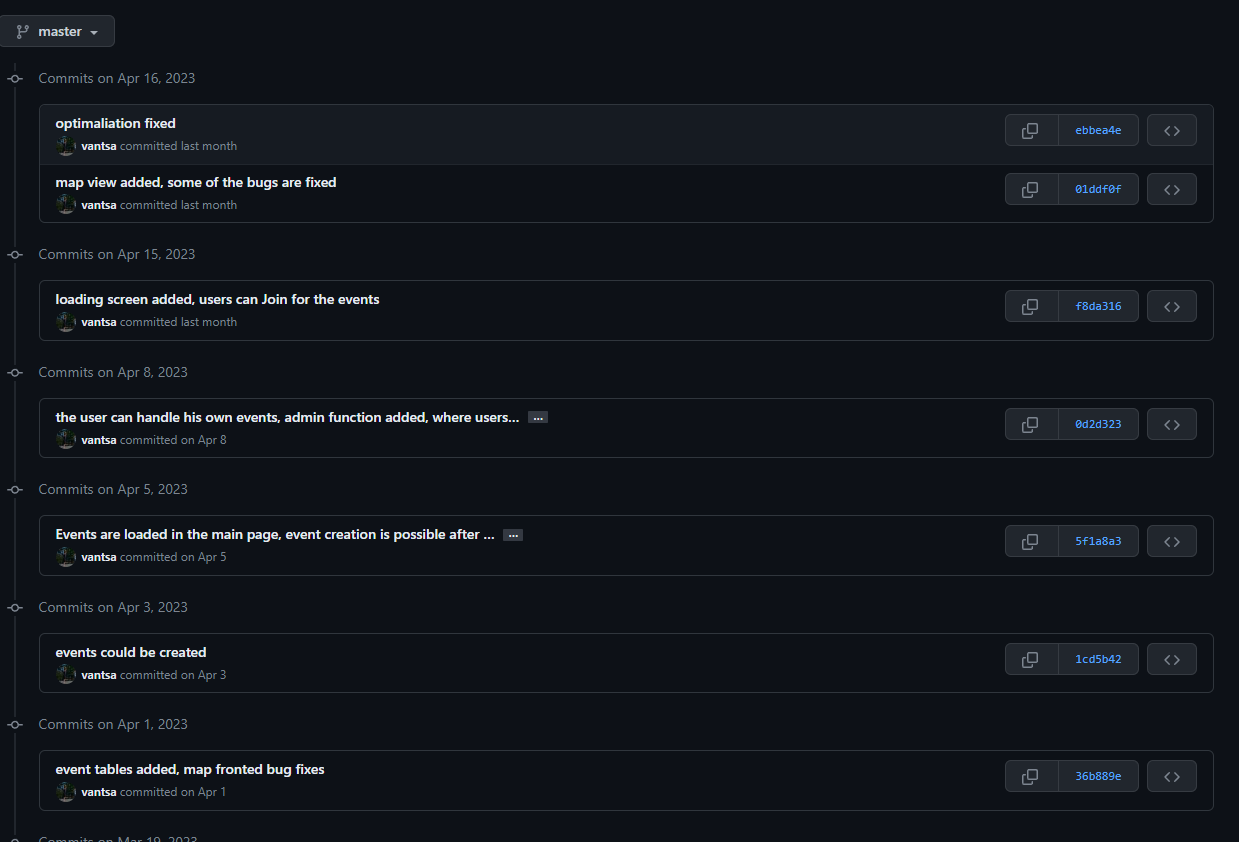
A Vuetify [3] egy komponenskönyvtár amelyet a Vue.js-hez használunk. Lehetővé teszi a fejlesztő számára, hogy korszerű, gyors weboldalt készítsen rövid idő alatt. A Vuetify [3] előre elkészített komponenseket tartalmaz (gombok, kártyák, űrlapok, stb.). Ezen komponensek szabadon felhasználhatóak és elérhetőek, illetve ami a legfontosabb a fejlesztők saját igényeik szerint személyre szabhatják azokat. Ezek az előre legyártott komponensek rendelkeznek különféle attribútomokkal, amelyek segítségével könnyebben és gyorsabban személyre szabható az adott komponens.

## **MySql/Xampp**

A MySql [10] egy adatbáziskezelő rendszer amely az egyik legelterjedtebb a piacon, a webes alkalmazások egyik alapvető pillére. Legfontosabb feladata az adatok tárolása, emellett különböző müveleteket végezhetőek mint például adatok törlése, frissítése, lekérdezése, beszúrása.   
 Az Xampp [4] egy nyílt forrású webes szerveralkalmazás, amely biztosítja a MqSql megfelelő működését a fejlesztő saját rendszerén. Meghatározó elemei az Apache webszerver, MariaDB, és a PHP nyelv futtatható rendszere. Annak ellenére, hogy az Xampp [4] legfőképpen PHP nyelvben megírt kódokra van kifejlesztve, az ASP .NET Correal való összekapcsolás során semmilyen problémába nem ütköztem.

## **Github**

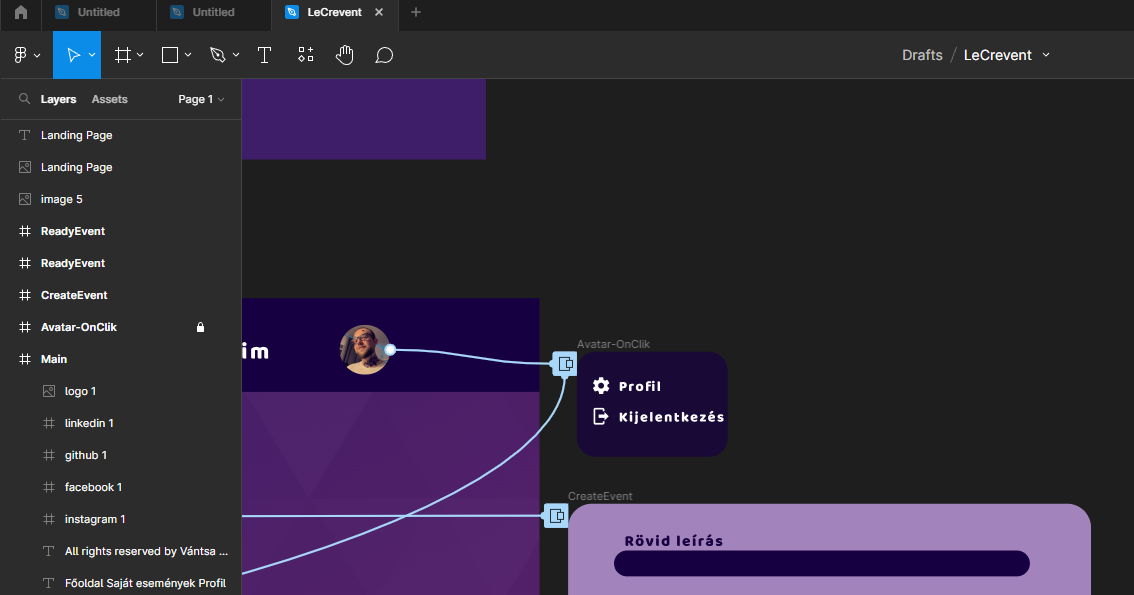
A Github [5] egy olyan felhő alapú szolgáltatás amely lehetőve teszi a fejlesztők számára a saját kódjaik tárolását illetve kezelését, emellett lehetővé teszi, hogy több ember egyszerre dolgozzon ugyanazon a projekten egyidőben, anélkül, hogy problémába ütköznének. Legjelentősebb funkciója a verziókövetés, amely lehetővé teszi a fejlesztők számára a kód felügyeletét illetve a változások követését. A Github [5] lehetőséget ad arra, hogy a meglévő kódot duplikáljuk egy külön ágra, majd a kívánt fejlesztés végrehajtásával, a főszálra visszavezetni a kódot, amely tartalmaz minden szükséges dolgot. Fontos megjegyezni, az adatok felhőben tárolódnak így bármilyen esetleges számítógép meghibásodás során nem érhet kódveszteség. A melléket ábrán (2.1 Ábra) látható az általam használt verziókövetes.



2.1. Ábra. Github verziókövetés

## **Figma**

A Figma [11] egy felhő alapú grafikus felületek tervezésére alkalmas platform. Legfőbb célja, a webes és mobilapplikációk megtervezésének leegyszerűsítése. Kezelőfelülete egyszerű, használata pár óra gyakorlás után teljesen elegendő egy kezdő webdesigner számára. Fontosnak tartottam egy tervezői eszköz használatát az applikációm megtervezéséhez, mivel ezáltal a késöbbiekben már egy kész elképzelés alapján tudtam a weboldalamat fejleszteni, ezáltal elkerültem a késöbbiekben felmerülő bizonytalanságokat az alkalmazásom felhasználói felületével kapcsolatban.

  
2.2. Ábra. Tervezés a Figma rendszerében

## **Node Package Manager (NPM)**

A Node Package Manager [12] egy könyvtárrendszer, amely lehetővé teszi különböző JavaScript csomagok telepítését és használatát a saját kódunkban ezáltal gyorsabb és egyszerübb is a fejlesztés.

Az applikációm fejlesztése során a következő csomagokat használtam:

* vue-axios [13] : api hivások kezelése, kapcsolatteremtés frontend-backend között
* vue-cors [14]: erőforrás kezelése, személyreszabott fejléc engedélyezése
* Eslint [15]: jelzést ad a potenciális hibákról (formázás, stílus, logika)
* jwt-decode [16]: a JWT-ben tárolt adatok dekódolására szolgál
* vue-leaflet [17]: egyszerű és korszerű térképkezelő, amely lehetővé tette a térkép megjelenítését
* vue-leaflet-markercluster: lehetővé tette pontok megjelőlését a térképen
* vue-router [18]: útvonalkezelésére szolgál
* vue-the-mask [19]: form-ok kitöltésénél formázást és korlátozásokat ellenőriz
* Vuetify [3]: Vue.js alapú előre legyártott komponensek könyvtára

# **Alkalmazás bemutatása**

Egy esemény szervező alkalmazásnak elengedhetlen része, hogy adatbázisban tárolódjanak a különféle adatok, ezért is volt szükség az említett technológiák használatára. Egy ilyen típusú alkalmazásnak úgy gondolom kötelezően tartalmaznia kell regisztrációhoz kötött azonosítást. A következőkben az alkalmazás részletes bemutatása következik, ahol külön kitérek a fejlesztés során felmerült nehézségekre, érdekességekre, illetve olyan dolgokra amelyek nélkül nem jöhetett volna létre az alkalmazás.

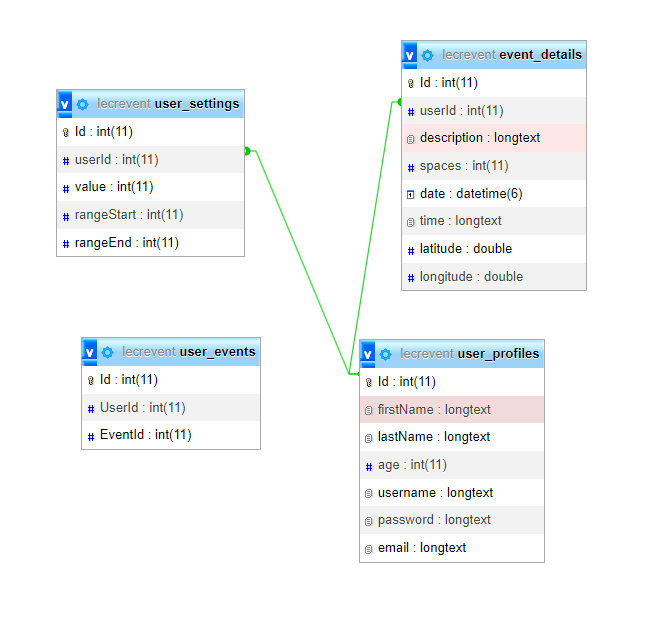
## **Programozási környezet**

Az alkalmazás fejlesztése kezdetlegesen csak és kizárólag a Visual Studio Code fejlesztői környezetében történt. Számomra az első pillanattól kezdődően egyértelmű volt, hogy ebben a kódszerkesztőben fogok dolgozni. Az egyik legfontosabb szempont számomra az az átláthatósága az adott környezetnek, amely a fent említett Visual Studio Code-nak az egyik legprofibban kivitelezett rendszere van, emellett számtalan bővítmény telepíthető amelyek megkönnyítették a fejlesztés során a dolgomat. Ilyen bővítmények például a szintaxisok kiemelése, automatikus kódkiegészítés, hibakeresés. Tovább a Git támogatásnak köszönhetően könnyen és gyorsan feltölthető a kód Github felhőjébe.

Mint említettem kezdetlegesen csak a Visual Studio Code környezetét használtam, de az idő múlásával, a tesztelések számának növekedésével kénytelen voltam a backend fejlesztését (ASP .NET Core) átvinnem a Microsoft által fejlesztett Microsoft Visual Studióba amely eredetileg a .NET Core kódszerkesztője. Az elöbbi kódszerkesztő csupán annyiban különbözik a Visual Studio Code-tól, hogy egy teljes körű fejlesztő környezet, amelyben mobil, windows, web és felhőalapú alkalmazások fejlesztése lehetséges.

## **Adatbázis**

A 3.1 Ábrán látható az alkalmazásom adatbázisa, amely 4 táblából áll. Ezekben a táblákban tárolódnak a különböző adatok, mint például a felhasználók adatai, és beállításai illetve az események adatai. Az alkalmazás fejlesztése során különböző triggereket is használtam, amelyeket a táblák bemutatása után részletesebben kibontok.



3.1. Ábra. Az alkalmazás adatbázis sémája

A ‘user\_profiles’ tábla tartalmazza a felhasználó adatait amelyet a regisztráció során adott meg. Az Id mező automatikusan növekvő (Auto Increment) ezáltal nem áll fent a lehetősége az ismétlődésnek. A felhasználónév és az email egyedi, ez biztosítja azt, hogy egy felhasználó csak egy email címről tudnak regisztálni az alkalmazásra. A jelszó titkosítva van, ezáltal a felhasználón kivül senki más nem tudhatja a jelszavát. A felhasználó életkora egy int típusú adat amelyet a felhasználó születési dátumként ad meg, majd ezt követően int tipussá alakítva mentődik el az adatbázisban.

|  |  |  |
| --- | --- | --- |
| **Attríbutum neve** | **Típus** | **Attríbutum szerepe** |
| id | int, primary, unique | Általános ID |
| firstName | varchar | Felhasználó vezetékneve |
| lastName | varchar | Felhasználó keresztneve |
| age | int | Felhasználó életkora |
| username | varchar, unique | Felhasználónév |
| password | varchar | Felhasználó jelszava(hash) |
| email | varchar, unique | Felhasználó email címe |

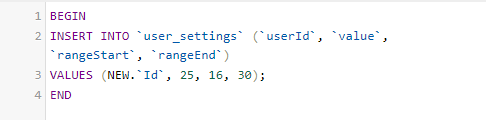
3.1. Táblázat user\_profiles tábla

A ‘user\_settings’ tábla tartalmazza a felhasználó beállításait. Ebben a táblában levő értékeket a felhasználó bejelentkezés után tudja frissíteni. A userId segítségével tudhatjuk, hogy melyik felhasználó beállításai fognak változni, vagy éppen melyik felhasználó beállítását látjuk. A ‘value’ érték meghatározza azt a kilométeres körzetet, amelyben a felhasználó szeretné látni a közelendő eseményeket. A ‘rangeStart’ illetve ‘rangeEnd’ attríbutumok egy olyan intervallum alső és felső határai amelyek által a felhasználónak lehetősége van beállítani azt, hogy mely az a korosztály akinek szeretné az eseményeit látni. Ezen adatok a felhasználó számára személyre szabhatóak.

|  |  |  |
| --- | --- | --- |
| **Attríbutum neve** | **Típus** | **Attríbutum szerepe** |
| id | int, primary | Általános ID |
| userId | int, foreign key | Felhasználó ID-je |
| value | int | Maximális km körzet |
| rangeStart | int | Minimális életkor |
| rangeEnd | int | Maximális életkor |

3.2. Táblázat user\_settings tábla

Regisztrációt követően egy trigger segítségével beállítja a default(alap) értékeket az adott felhasználónak.



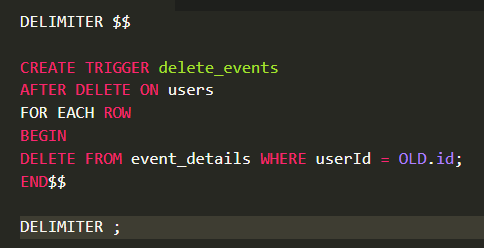
3.2. Ábra. Regisztrációt követő INSERT trigger

Az ‘event\_details’ tábla tartalmazza az eseményeket és az azokhoz tartozó adatokat. A ‘userId’ által tudjuk melyik felhasználó hozta létre az adott eseményt. A ‘description’ szolgál az esemény pontosabb ismertetésére. A ‘spaces’ arra szolgál, hogy megmutatja hány férőhely van az adott eseményre, ez az adat dinamikusan változik amikor egy felhasználó csatlakozik az eseményhez, vagy éppen visszalép az adott eseményről. A ‘date’ és a ‘time’ attríbutumok szolgálnak az esemény pontos idejének meghatározására. A ‘latitude’ és ‘longitude’ adatok a térkép által meghatározott koordinátákat tárolják ,ezek által a Leaflet könnyedén tudja menteni illetve lekérni az esemény helyszínének koordinátáit.

|  |  |  |
| --- | --- | --- |
| **Attríbutum neve** | **Típus** | **Attríbutum szerepe** |
| Id | int, primary | Általános ID |
| userId | int, foreign | Felhasználó ID-je |
| description | longtext | Esemény címe |
| spaces | int | Elérhető helyek száma |
| date | datetime | dátum |
| time | longtext | Pontos idő (óra:perc) |
| latitude | double | szélesség |
| longitude | double | hosszúság |

* 1. Táblázat user\_details tábla

Egy felhasználó esetleg törlése során, minden felhasználó tartozó adat törlődik, beleértve az összes eseményt amelyet az adott felhasználó hozott létre. Ennek a kivitelezéséhez ‘DELETE’ triggert használtam amely a mellékelt (3.3. Ábra) ábrán is látható.



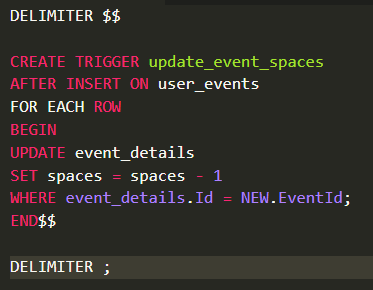
3.3. Ábra. Felhasználó törlését követő DELETE trigger

A ‘user\_events’ tábla tartalmazza mindazokat a felhasználokat és eseményeket amelyekhez csatalkozott már felhasználó. Ennek a táblának köszönhetően lehet tudni, hogy hány elérhető hely van egy adott eseményhez, vagy éppen azt, hogy az adott felhasználó csatlakozott-e már az adott eseményhez

|  |  |  |
| --- | --- | --- |
| **Attríbutum neve** | **Típus** | **Attríbutum szerepe** |
| Id | int, primary | Általános ID |
| UserId | int | Felhasználó ID-je |
| EventId | int | Esemény ID-je |

* 1. Táblázat user\_events tábla

Ha a ‘user\_events’ táblába egy új adat kerül beszúrásra akkor az ‘event\_details’ táblában levő ‘spaces’ mező frissül. Egy új adat beszúrása azt jelenti, hogy egy felhasználó csatlakozott egy eseményhez, így az adott esemény azonosító alapján a ‘spaces’ mező értéke csökkeni fog eggyel, természetesen törlés esetén ez az érték növekszik eggyel. A spaces mező értéke soha nem lehet negatív szám. Ezeknek a müveleteknek a megvalósításához triggereket használtam, a mellékelt (3.4 Ábra) egy ilyen trigger kódját mutatja be.



3.3. Ábra. Beszúrás után ‘spaces’ érték változása

A triggerek segítségével még színesebbé és gyorsabbá tudtam tenni úgy az alkalmazásom mint a hozzá tartozó adatbázist. Az ASP .NET Core-ban használt Entity Framework [7] keretrendszernek köszönhetően a folyamatos kódváltozások backenden dinamikusan megjelentek az adatbázisban is egyaránt.

## **Az alkalmazás fontosabb funkciói**

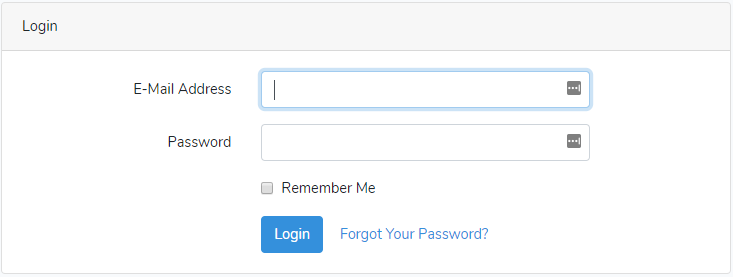
Fusce a metus eu diam varius congue nec nec sapien. Vestibulum orci tortor, sollicitudin ac euismod non, placerat ac augue. Nunc convallis accumsan justo.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

### **Login**

Fusce a metus eu diam varius congue nec nec sapien. Vestibulum orci tortor, sollicitudin ac euismod non, placerat ac augue. Nunc convallis accumsan justo.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

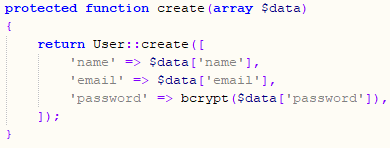


3.2.Ábra. Login ablak

### **Regisztráció**

Fusce a metus eu diam varius congue nec nec sapien. Vestibulum orci tortor, sollicitudin ac euismod non, placerat ac augue. Nunc convallis accumsan justo.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.



3.3.Ábra. Új felhasználó létrehozása

# **Következtetések**

Fusce a metus eu diam varius congue nec nec sapien. Vestibulum orci tortor, sollicitudin ac euismod non, placerat ac augue. Nunc convallis accumsan justo. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Donec malesuada vehicula lectus, viverra sodales ipsum gravida nec. Integer gravida nisi ut magna mollis molestie. Nullam pharetra accumsan sagittis. Proin tristique rhoncus orci, eget vulputate nisi sollicitudin et. Quisque lacus augue, mollis non mollis et, ullamcorper in purus. Morbi et sem orci.

Praesent accumsan odio in ante ullamcorper id pellentesque mauris rhoncus. Duis vitae neque dolor. Duis sed purus at eros bibendum cursus nec a nulla. Donec turpis quam, ultricies id pretium sit amet, gravida eget leo.

# **Irodalomjegyzék**

1. Az ASP.NET Core dokumentációjahttps://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0
2. Ekler Péter, [Fehér Marcell](https://www.libri.hu/szerzok/feher_marcell.html), [Forstner Bertalan](https://www.libri.hu/szerzok/forstner_bertalan.html), [Kelényi Imre](https://www.libri.hu/szerzok/kelenyi_imre.html). (2012): *Android-alapú szoftverfejlesztés*, Szak Kiadó Kft., Bicske.
3. Ferrucci, F., Salza, P., Sarro, F. (2018): *[Using Hadoop MapReduce for Parallel Genetic Algorithms: A Comparison of the Global, Grid and Island Models](https://www.mitpressjournals.org/doi/full/10.1162/evco_a_00213),* Evolutionary Computation, Vol. 26, pp. 535–567.
4. Nixon, R. (2018): *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*, [O'Reilly](http://csikszereda.qulto.ro/ro/results/-/results/publisher/O/solr?p_auth=Fm0hnmWe) Media, USA.
5. Sklar, D. and Trachtenberg, A. (2014): *PHP cookbook*, [O'Reilly](http://csikszereda.qulto.ro/ro/results/-/results/publisher/O/solr?p_auth=rgBPwg8X) Media, USA.
6. Stauffer, M. (2017): *Laravel Up and Running*, O’Reilly Media, USA.
7. Ullman, J.D., <http://infolab.stanford.edu/~ullman/focs/ch07.pdf>, 2019.03.15.
8. https://kinsta.com/knowledgebase/what-is-github/