Pannon Egyetem

Műszaki Informatikai Kar

Villamosmérnöki és Információs Rendszerek Tanszék

Programtervező informatikus BSc

**SZAKDOLGOZAT**

**Edzőterem menedzsment webalkalmazás fejlesztése**

**Selyem Péter Ferenc**

Témavezető: Nagy Zsuzsanna

2025



Hallgatói nyilatkozat

Alulírott Selyem Péter Ferenc hallgató (Neptun kód: WOKEUE) kijelentem, és a dolgozat feltöltésével egyidejűleg nyilatkozom, hogy az Edzőterem menedzsment webalkalmazás fejlesztése című szakdolgozatot (a továbbiakban: dolgozat) a Pannon Egyetem Műszaki Informatikai Kar Villamosmérnöki és Információs Rendszerek Tanszékén készítettem a programtervező informatikus oklevél megszerzése érdekében.

Kijelentem, hogy a dolgozatban csak a megadott és hivatkozott forrásokat használtam fel, és ezekre a vonatkozó idézési szabályok szerint hivatkoztam.

Nyilatkozom, hogy a dolgozat érdemi része saját szellemi alkotásom eredménye, és azt más intézményben, szakon, vagy felsőfokú képesítés megszerzésére nem nyújtottam be. Tudomásul veszem, hogy a plágium vagy szerzői jogsértés esetén a dolgozatom elutasításra kerülhet, és ellenem fegyelmi eljárás indulhat. Tudomásul veszem továbbá, hogy szerzői jogsértés esetén az Egyetem jogosult a dolgozat elérhetőségét korlátozni, valamint eltávolítani a dokumentumot a dolgozatok tárolására szolgáló, a témát vezető szervezeti egység által meghatározott elektronikus zárt rendszerből.

Tudomásul veszem továbbá, hogy a Pannon Egyetem a dolgozat eredményeit saját céljaira eltérő írásbeli megállapodás hiányában a Pannon Egyetem Szellemi Tulajdon Kezelési Szabályzatában foglaltaknak megfelelően szabadon felhasználhatja.

Nyilatkozom, hogy a dolgozat elkészítése során mesterséges intelligencia eszközöket *használtam /nem használtam[[1]](#footnote-1).*

Nyilatkozom, hogy a dolgozat elkészítése során az alábbi táblázatban feltüntetett mesterséges intelligencia eszközöket kizárólag a kutatási, illetve fejlesztési feladat támogatására használtam fel, az érdemi munka, elemzés és következtetések teljes mértékben saját szellemi alkotásomat képezik.

Példa a táblázat kitöltésére:

|  |  |  |  |
| --- | --- | --- | --- |
| **Alkalmazott technológia** | **Alkalmazás módja** | **Előállított tartalom** | **MI használat aránya** |
| GPT-4o (OpenAI) | szöveges leírás technológiákhoz | 2 fejezet | 20% |
| DeepSeek | Hibajavítási segédlet |  | 5% |
| GPT-4o | Hibajavítási segédlet |  | 10% |
|  |  |  |  |

Dátum: Veszprém, 2025. május 6.

*Selyem Péter Ferenc*

Témavezetői nyilatkozat

Alulírott Nagy Zsuzsanna témavezető kijelentem, hogy az Edzőterem menedzsment webalkalmazás fejlesztése című dolgozatot *Selyem Péter Ferenc* a Pannon Egyetem Villamosmérnöki és Információs Rendszerek Tanszékén készítette programtervező informatikus végzettség megszerzése érdekében.

Kijelentem, hogy a dolgozat védésre bocsátását engedélyezem.

Dátum: Veszprém, 2025. május 6.

*Nagy Zsuzsanna*

Köszönetnyilvánítás

A hallgató köszönetet nyilvánít mindazoknak, akiktől (elméleti, gyakorlati, erkölcsi stb.) segítséget kapott.

Tartalmi összefoglaló

A szakdolgozatom egy edzőterem menedzsment webalkalmazás megtervezéséről és megvalósításáról szól. Az alkalmazás ötlete 2024 nyarán jutott eszembe, amikor elkezdtem komolyan edzőterembe járni és önmagamat fejleszteni. A terembe járva láttam, hogy még mindig a régimódi papírbérletet használják.

A webalkalmazáshoz elkészítettem a felhasználói felületet és annak a kiszolgáló szerverét, amelyekkel lehetősége van a felhasználónak jegyet és bérletet vásárolni, edzésekre jelentkezni, megnézni milyen hivatalos edzők járnak a terembe, továbbá lehetősége van az edzései naplózására is.

A webalkalmazás megvalósításához a Visual Studio-t és a Visual Studio Code-ot használtam, mivel ezeket már volt szerencsém használni, valamint mindkettő az egyik legjobb és legelterjedtebb a maga területén. Az alkalmazást C# nyelven írtam a .Net keretrendszert használva, mivel ezek modernek és elterjedtek a hasonló alkalmazások megvalósításához. Emellett a .Net keretrendszerben sok segítséget és bővítményt lehet találni, így ezt találtam a legalkalmasabbnak a feladat backend részéhez. Adatbázis kezeléshez az SQLite alkalmazást választottam, mert ingyen elérhető és tökéletes a feladatra. A frontend részhez pedig a jól megszokott és bevált HTML, CSS és JavaScript nyelveket használtam, és mellette a Bootstrap-et az egyszerűbb, esztétikusabb megvalósításért és a jobb felhasználói élményért.

**Kulcsszavak:** [4-6 kulcsszó felsorolása, vesszővel elválasztva]

Abstract

Abstract in English

**Keywords:** [list 4-6 keywords]

Tartalomjegyzék

[Jelölésjegyzék 11](#_Toc197366435)

[1. Bevezetés 12](#_Toc197366436)

[2. Felhasznált eszközök és technológiák 13](#_Toc197366437)

[1.1. Visual Studio 13](#_Toc197366446)

[1.2. C# programozási nyelv 13](#_Toc197366466)

[1.3. ASP.NET Core 14](#_Toc197366467)

[1.4. Visual Studio Code 14](#_Toc197366468)

[1.5. Bootsrap 15](#_Toc197366469)

[2. Hasonló oldalak vizsgálata 16](#_Toc197366472)

[2.1. Flex Gym 16](#_Toc197366473)

[2.2. Planet Fitness 17](#_Toc197366479)

[3. Rendszerterv 19](#_Toc197366481)

[3.1. Funkciók 20](#_Toc197366482)

[3.1.1. Regisztráció 20](#_Toc197366483)

[3.1.2. Belépés 20](#_Toc197366484)

[3.1.3. Jegyvásárlás 20](#_Toc197366485)

[3.1.4. Jegy használat 20](#_Toc197366486)

[3.1.5. Edzés naplózás 20](#_Toc197366487)

[3.1.6. Jegyek kezelése 21](#_Toc197366488)

[3.1.7. Felhasználók kezelése 21](#_Toc197366489)

[3.1.8. Külön edzések 21](#_Toc197366490)

[4. Adatbázis 21](#_Toc197366492)

[5. Megvalósítás 24](#_Toc197366494)

[5.1. Oldal Térkép 24](#_Toc197366495)

[5.2. Főoldal 24](#_Toc197366498)

[5.3. Jegyek oldal 28](#_Toc197366499)

[5.4. Edzések oldal 29](#_Toc197366500)

[5.5. Edzők oldal 29](#_Toc197366501)

[Rólunk oldal 30](#_Toc197366502)

[5.6. 30](#_Toc197366511)

[Profil 31](#_Toc197366512)

[5.7. 31](#_Toc197366513)

[5.8. Admin oldal 35](#_Toc197366514)

[6. Összefoglalás 39](#_Toc197366515)

[Irodalomjegyzék 40](#_Toc197366516)

[Ábrajegyzék 45](#_Toc197366517)

[Táblázatjegyzék 46](#_Toc197366518)

Jelölésjegyzék

# Bevezetés

A szakdolgozatom egy edzőterem menedzsment webalkalmazás megtervezéséről és megvalósításáról szól. Az alkalmazás ötlete 2024 nyarán jutott eszembe, amikor kezdtem komolyan edzőterembe járni és önmagamat fejleszteni. A terembe járva láttam, hogy még mindig a régimódi papírbérletet használják. A videókban, amiket megnéztem, hogy informálódjak afelől, hogy mit hogyan is kéne csinálni az edzés alatt, láttam sokféle edzőtermet. Voltak olyanok, ahol egy olyan beléptető kapu van, ahol be kell szkennelni a bérletet, hogy beengedjen. Ezzel kapcsolatban beszéltem másokkal, hogy mennyivel egyszerűbb és jobb lenne egy olyan terem, ahol online lehetne rendezni a jegyeket és bérleteket. Emellett, a terem a videóban látotthoz hasonló beléptető rendszerrel rendelkezne.

A feladatom egy olyan egy könnyen használható webalkalmazás fejlesztése, amihez készítettem egy felhasználói felületet és egy kiszolgáló szervert. Ezen a felületen a felhasználó tud regisztrálni és ez után van lehetősége bejelentkezni és jegyeket vagy bérleteket venni, edzésekre jelentkezni, megnézni milyen hivatásos edzők vannak a nyilvántartásban, és ha szeretné naplózhatja az edzéseit is. Ha egy edző jelentkezik be, akkor neki még lehetősége van külön edzéseket is kiírni. Adminok tudnak ezek felett még új jegyeket felvinni vagy meglévőket módosítani, felhasználók jogait változtatni vagy épp törölni egyfelhasználót.

A Felhasznált eszközök és technológiák című fejezetben leírom, hogy miket használtama webalkalmazásom megvalósításához.

A Hasonló oldalak fejezeben az összeszedett két oldalról írok, amiket megvizsgáltam, hogy a jelenleg létező weblapokat hogyan valósították meg.

A Rendszerterv fejezetben kifejtem, hogy milyen funkciókat is tartalmaz a webalkalmazásom.

Ezt követően az adatbázisról szóló fejezet következik és hogy abban hogyan vannak tárolva az adatok.

Végül a Megvalósítás elnevezésű fejezetben a funkciók megvalósításáról található leírás.

# Felhasznált eszközök és technológiák

## Visual Studio

A backend megvalósításához a Visual Studio-t használtam, ami a Microsoft által fejlesztett integrált fejlesztőkörnyezet (IDE). Ezt széles körben használják különféle szoftverek fejlesztésére, beleértve az asztali, mobil és webes alkalmazásokat is. Ez az eszköz különösen népszerű a .NET fejlesztések során, például C#, ASP.NET és más Microsoft technológiák használatakor, hiszen átfogó eszközkészletet kínál a fejlesztés minden szakaszára; a kódírástól kezdve a hibakeresésen át egészen az alkalmazások telepítéséig. A Visual Studio támogatja a különböző nyelveket és platformokat és megkönnyíti a fejlesztést, ezáltal ideális választás kezdő és tapasztalt fejlesztők számára is.



## C# programozási nyelv

Az általam választott programozási nyelv a C#, amivel van egy kevés tapasztalatom és azt mondják, hogy ez az egyik legjobb nyelv webalkalmazás fejlesztésre. Emellett sok segédanyag található az interneten hozzá, ezért erre a nyelvre esett a választásom.

A C# egy általános célú, modern és objektumorientált programozási nyelv, amelyet a Microsoft fejlesztett ki 2000-ben, és része a .NET ökoszisztémának. A célja, hogy egyszerre legyen könnyen használható a kezdő programozók számára, és hatékony eszköz a professzionális fejlesztők kezében, akik komplex alkalmazásokat készítenek.

A C# szintaxisa tiszta és intuitív. Ez lehetővé teszi a fejlesztők számára, hogy gyorsan elsajátítsák a nyelvet, különösen, ha már rendelkeznek előzetes tapasztalattal más programozási nyelvekkel. A C# teljes mértékben támogatja az objektumorientált programozást (OOP), amely lehetővé teszi a fejlesztők számára, hogy moduláris, újra felhasználható és karbantartható kódot hozzanak létre.

A C#-ot eredetileg a Windows ökoszisztéma számára tervezték, de a .NET Core (és az újabb .NET verziók) megjelenésével a nyelv platformfüggetlenné vált, így támogatja a Windows, macOS és Linux rendszereket is. Az ASP.NET Core segítségével a C# ideális választás webalkalmazások és API-k fejlesztésére - csakúgy, mint asztali alkalmazások, mobilalkalmazások vagy játékok fejlesztésére.

## ASP.NET Core

Az előbb is emlegetett ASP.NET a Microsoft által fejlesztett nyílt forráskódú, nagy teljesítményű keretrendszer webes alkalmazások fejlesztésére. Az ASP.NET segítségével dinamikus, modern, skálázható és biztonságos webalkalmazások, API-k, valamint valós idejű szolgáltatások fejleszthetők különféle platformokon (pl. Windows, macOS és Linux rendszereken).

Az ASP.NET Core az egyik leggyorsabb webes keretrendszer, köszönhetően a fejlett optimalizálási technikáknak, például az aszinkron feldolgozásnak, a Kestrel webszervernek és a hatékony memóriahasználatnak. ASP.NET-ben lehetőség van különböző csomagokat becsatolni a programunkhoz, amik megkönnyítik a fejlesztést.

## Visual Studio Code

A frontend megvalósításához a Visual Studio Code nevű programot használom, ami egy nyílt forráskódú, könnyűsúlyú, de rendkívül erőteljes kódszerkesztő és fejlesztőkörnyezet (IDE), amelyet a Microsoft fejlesztett. 2015-ben jelent meg, és gyorsan az egyik legnépszerűbb eszközzé vált a fejlesztők körében, köszönhetően a sokoldalúságának, bővíthetőségének és gyorsaságának.

A Visual Studio Code támogatja a különféle programozási nyelveket és fejlesztési környezeteket, mint például JavaScript, Python, C#, Java, HTML/CSS, és még sok más. Az Extension Marketplace segítségével rengeteg bővítményt lehet hozzáadni, amik segítik a fejlesztést akár csak annyival, hogy átláthatóbbá vagy színesebbé teszik a kódot. Vannak olyan bővítmények is, amelyek a kód írása közben próbálják kitalálni, hogy mit akarunk és javaslatokat adnak, amiket, ha elfogadunk, akkor ki sem kell írni az akár több soros parancsokat.

Én az Extensionök közül használtam a HTML EndTag Labels elnevezésüt, amely a különböző html tagek után odaírja hogy mit is zár be és milyen classok és id-k tartoznak ahoz az elemhez.

Használtam még a Live Server elnevezésü kiegészítőt is, ami szerintem az egyik legfontosabb kiegészítő amit mindenkinek ajánlok, hiszen ezzel egy lokális szerveren van a weboldal ezálltal reálisabban viselkedik és mellé még minden mentés után egyből frissül az oldal, ezzel megkönnyítve a fejlesztést.

## Bootsrap

A Bootstrap egy népszerű, nyílt forráskódú front-end keretrendszer, ami lehetővé teszi, hogy gyorsan és hatékonyan építsünk fel vizuálisan tetszetős és mobilbarát felületeket HTML, CSS és JavaScript kombinációjával. Az előre definiált stílusok és komponensek segítségével időt takaríthatunk meg, miközben a webes szabályokat is betartjuk. Ezeket egyszerűen csak be kell csatolni a HTML fájlba és már használhatjuk is a weboldalunk fejlesztésére.

# Hasonló oldalak vizsgálata

## Flex Gym

A Flex Gym oldala egy magyar példa hasonló weblapra. Nekem nem tetszik, hogy sok a kép a főoldalon (1. ábra), de viszont vannak jó megoldásaik. Tetszik, hogy a nyitvatartás megtalálható az oldal alján egyszerűen, de szerintem mehetne egy menüpont alá, az elérhetőségekkel együtt, amik viszont alig látszódnak a menüben.

A képen szöveg, Emberi arc, ember, személy látható

Automatikusan generált leírásA jegyárak egy menüpont alatt vannak, amit hasonlóan oldanék meg én is, de a főoldalra kitenném 1-2 gyakoribb jegy árát. Online vett jegyeket, ha minden igaz, e-mailben küldik el. Ezt én inkább az oldalon oldanám meg, a profilban, ahol egy „jegyeim” menüpont lenne megtalálható. Az „edzőink” (2. ábra) menüpontban tetszik, hogy felsorolja az edzőket és a bemutatkozásaikat. Szerintem, ha lenne pár vélemény emberektől, akiket edzettek, akkor jobb lenne.

1. ábra: A Flex Gym főoldala

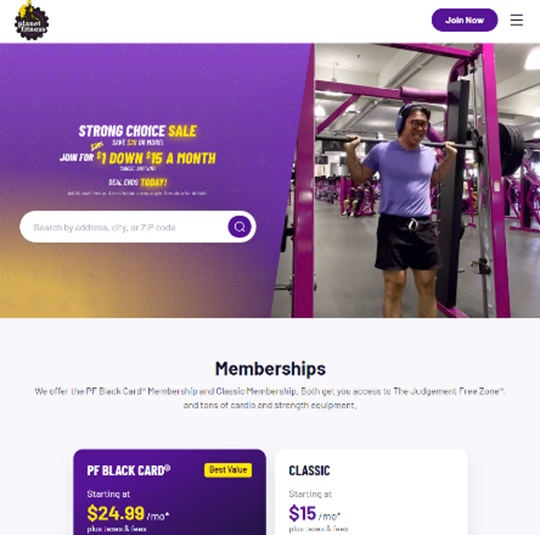
A képen szöveg, Emberi arc, nő, ruházat látható

Automatikusan generált leírás

2. ábra: A Flex Gym „Edzőink” oldala

## Planet Fitness

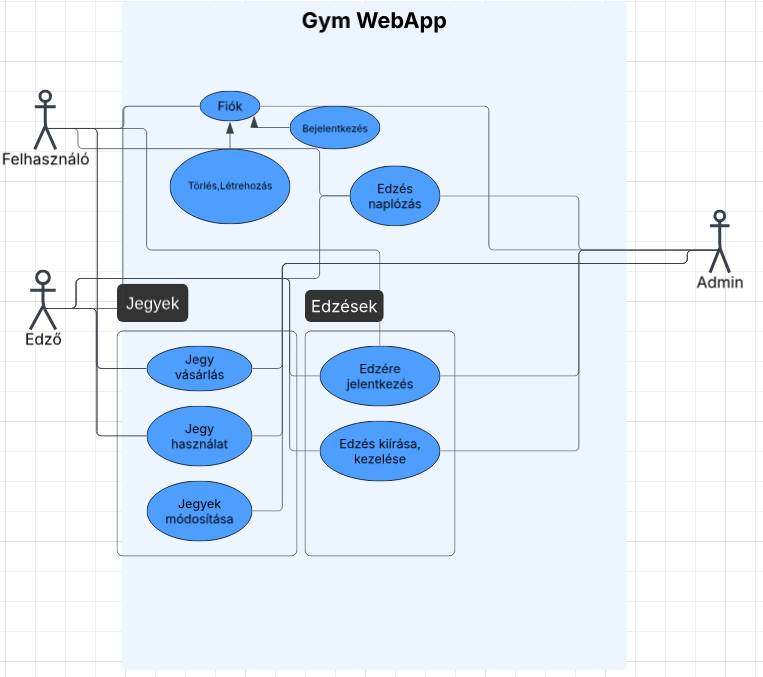
A Planet Fitness egy külföldi példa, aminek a főoldalán (3. ábra) kint vannak az árak. Ezáltal egyből lehet látni, hogy mennyivel kell számolni, ha oda akar menni az ember. Szerintem itt is sok képet tettek ki az oldalra. Legalábbis sok helyet foglalnak el. Viszont a nyitvatartás eléréséhez először ki kell keresni, hogy melyik terembe tervezünk menni és ott lesz leírva, ami érthető. Szerintem egy átlagos nyitvatartást írhattak volna egy könnyebben elérhető helyre.



3. ábra: A Planet Fitness főoldala

# Rendszerterv

Az alábbi ábrán (4. ábra) látható az alkalmazás Use Case diagramja.



4. ábra: Az alkalmazás Use Case diagramja

A webalkalmazásban háromféle **felhasználói jogosultság és szerepkör** található:

* **Felhasználó**: tud fiókot létrehozni, ebbe be- és kijelentkezni és tudja törölni a saját fiókját. Tud még edzést naplózni, jegyeket venni és felhasználni, továbbá kiírt edzésekre jelentkezni.
* **Edző:** az alap felhasználói képességek mellett tud edzéseket kiírni. Neki meg kell adni a telefonszámát, milyen edzés típusra szakosodott és képet is feltölthet magáról.
* **Admin:** az edzői jogokon kívül tud még másik személy fiókját törölni, jegyeket módosítani, új jegy típust hozzáadni, edzéseket kiírni és kezelni és mások jogosultságát kezelni.

## Funkciók

### Regisztráció

Egy új felhasználó tud regisztrálni és azután tud csak belépni és elírni az oldal többi funkcióját

### Belépés

A felhasználó be tud lépni, és ezek után érheti el a további funkciókat

### Jegyvásárlás

Mikor a jegyeknél rákattint a vásárlás gombra, egy külön ablak ugrik fel, ahol meg kell adnia a bankkártya adatait és ezek után, tudja megvenni a kiválasztott jegyet.

### Jegy használat

A felhasználó a profilján megtalálja a „Jegyeim” gombot, amivel előhozza a már megvásárolt jegyeit és itt lehetősége van rányomni a felhasználás gombra, amivel kap egy belépőkódot, amivel bejuthat a terembe.

### Edzés naplózás

A felhasználó profilján található egy „Edzés naplózás” gomb, amivel előhoz további három gombot. Az egyikkel egy új elvégzett gyakorlatot tud felvinni, a másikkal a már felvitt gyakorlatait tudja kilistázni.

### Jegyek kezelése

Az Admin joggal rendelkezők el tudják érni az admin oldalt, ahol ki tudják listázni az összes létező jegy típust és lehetőségük van ezeknek a szerkesztésükre vagy egy új jegytípus felvitelére.

### Felhasználók kezelése

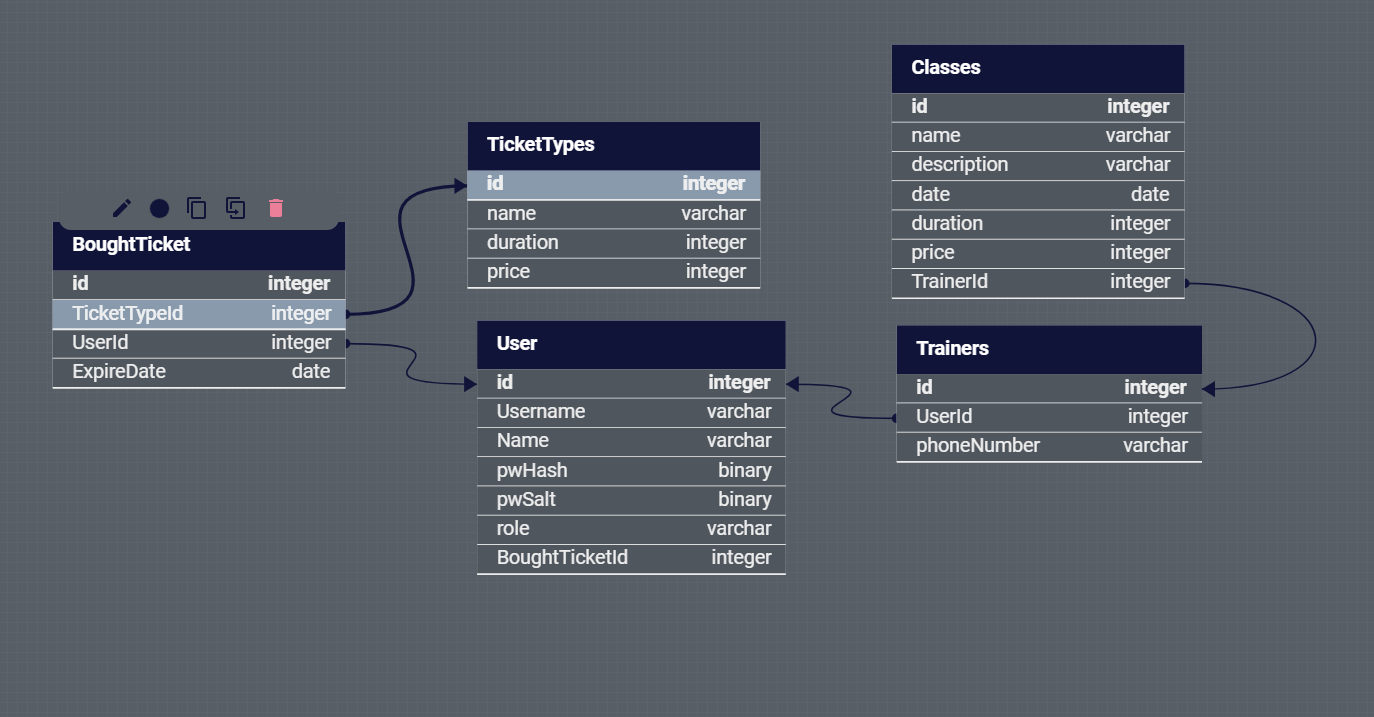
Az Admin joggal rendelkezők az admin oldalon ki tudják listázni az összes regisztrált felhasználót és tudják módosítani a jogosultságukat vagy esetlegesen törölni a profiljuk.

### Külön edzések

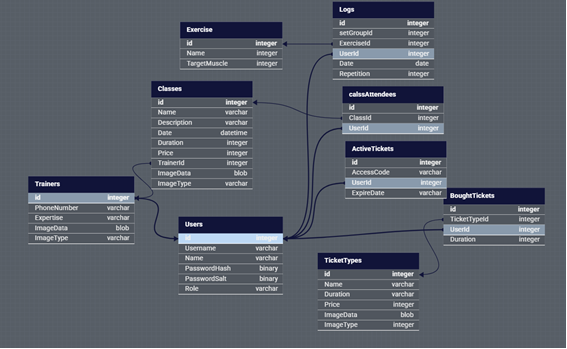
Edzőknek és Adminoknak lehetőségük van külön edzéseket kiírni amire a felhasználók tudnak jelentkezni.

# Adatbázis

Az adatokat egy adatbázisban tárolom, ami kezdetileg 5 táblából állt (5. ábra), de a mostani változat már 9 táblát foglal magába (6. ábra).



5. ábra: Az adatbázis kezdetleges modellje



6. ábra: Az adatbázis jelenlegi modellje

A User táblában kerül eltárolásra a felhasználók:

* teljes neve,
* felhasználó neve, amit a bejelentkezéshez használ
* jelszava titkosított formában

A **Trainers** tábla tartalmazza az edző:

* Telefonszámát
* Szakosodását
* Képét
* Felhasználó azonosítót a User táblából

A **Classes** táblában van tárolva a kiírt edzések adatai, amik az:

* Edzés megnevezése
* Leírás az edzésről
* Edzés időpontja
* Edzés időtartama
* Ára
* Edzéshez rendelt edző azonosítója a Trainers táblából
* Kép

**ClassAttendees** Táblában tárolom, hogy ki melyik edzésre jelentkezett

**TicketTypes** táblában találhatók a létrehozott jegytípusoknak a:

* Megnevezése
* Jegy érvényességi ideje (jegytől függően dátum vagy használat szám)
* Ára
* Képe

**BoughtTickets** táblában vannak a megvásárolt jegyek tárolva a következő módon:

* jegy típusának azonosítója A TicketTypes táblából
* A felhasználó azonosítója, aki megvette a jegyet
* A visszamaradt idő/használat

**ActiveTickets** táblában a jelenleg aktív jegyek vannak eltárolva:

* A belépési kóddal
* Megvett jegy azonosítója a BoughtTicket táblából
* Jegy tulajdonosának azonosítójával a User táblából
* És a jegy lejárati ideje

**Logs** táblában tárolódik a felhasználók álltal felvitt edzések:

* Csoportosító kulcsa
* Adott gyakorlat azonosítója az Exercise táblából
* A felhasználó azonosítója a User táblából, aki felvitte az edzést
* A gyakorlat dátuma
* Gyakorlat ismétlésszáma

Az **Exercise** táblában vannak eltárolva a létező gyakorlatok és hogy milyen izmokat dolgoznak.

Az adatbázis kezeléséhez, adatok módosításához és eléréséhez ORM-et használtam, mert ez megkönnyíti az adatbázis létrehozását és a műveletek végrehajtását, hiszen egyszerű osztályokat (7. ábra) kell létrehozni és ezekből építi fel az adatbázis.

A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

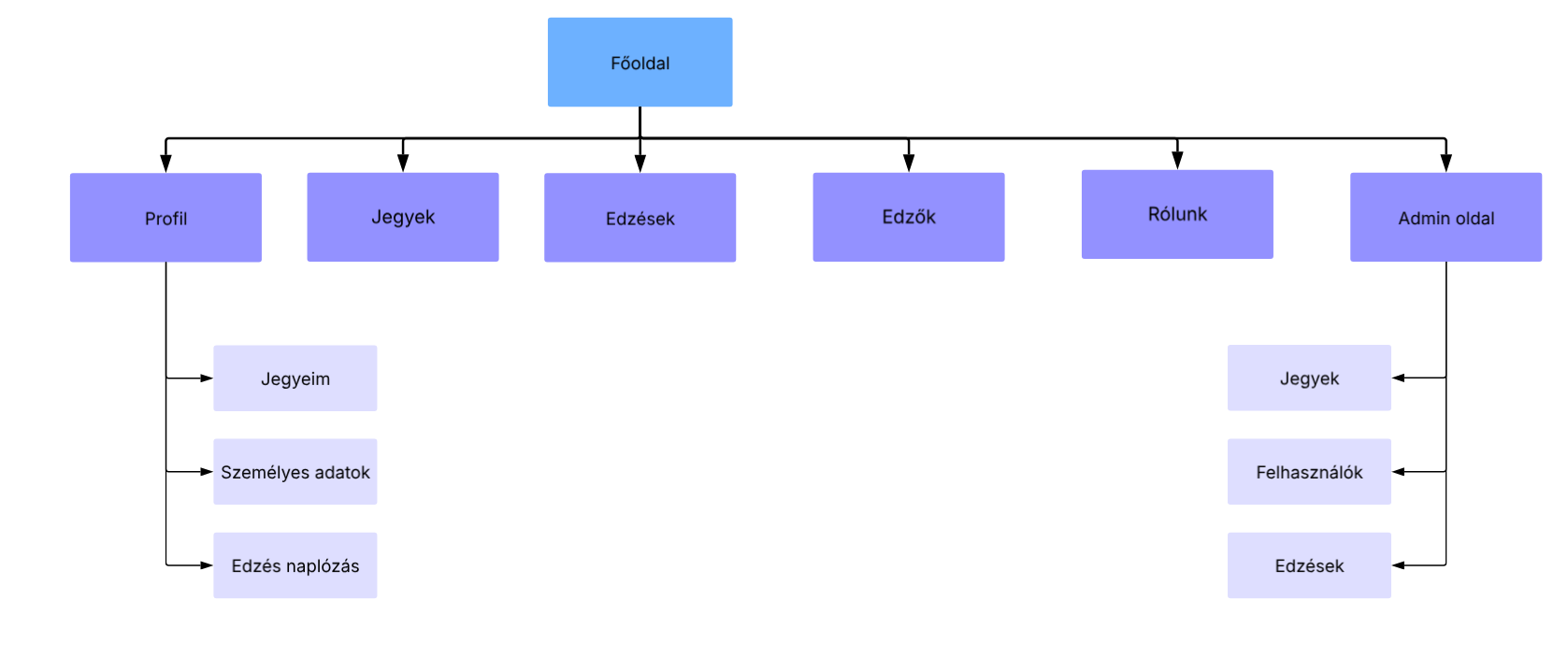
8. ábra: Adatbázis összerakása

7. ábra: Osztály adatbázishoz

# Megvalósítás

## Oldal Térkép

Az oldal térképe az alábbi ábrán (9. ábra) látható.



9. ábra: Oldaltérkép

## Főoldal

A Főoldalról (8. ábra) minden külön oldal elérhető egyszerűen. A felső navigációs sávból elérhetőek a Jegyek, Edzések, Edzők és rólunk oldalak, valamit a terem nevére kattintva a főoldal és a sáv jobb oldalán az emberkére kattintva előjön egy legördülő lista további opciókkal. Ez a navigációs sáv minden oldalon megtalálható.

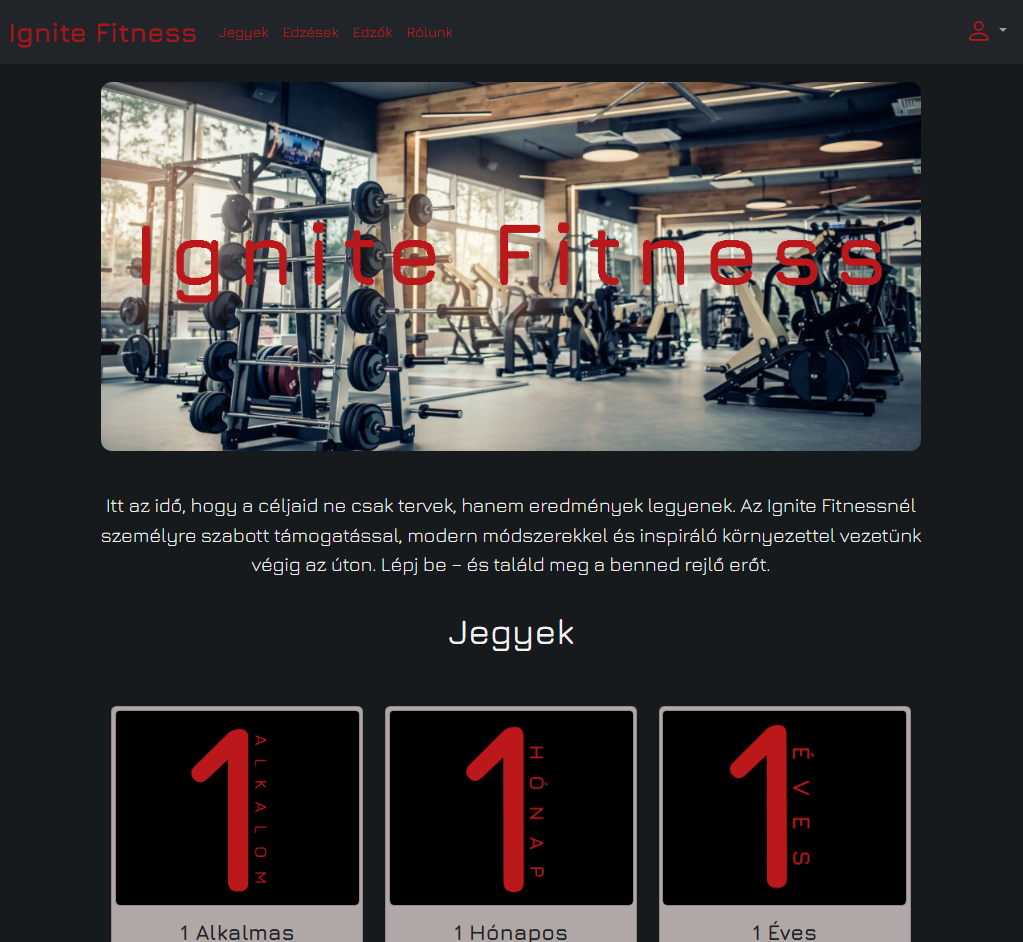
A Főoldalon még megtalálható a jegy típusokból, edzésekből és edzőkből három-három opció, mindegyik sávnál egy külön gombbal, ami elvezet ezeknek a saját oldalára.

Mindegyik oldalt reszponzívra csináltam, hogy kisebb kijelzőkön is használható legyen, ehhez a Bootstrap keretrendszer rácsos megoldását használtam, ami sorok és oszlopok definiálásán alapul.

Reszponzívitás miatt az előbb említett jegy, edző és edzés sávokat úgy alakítottam ki, hogy kis képernyőn átmegy lapozható listává, amit oldalra lehet görgetni, ehhez a Bootstrapnek a carousel modulját használtam, amivel szépen meg lehetett ezt valósítani A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.és jó iránytást is adnak mellé (10.ábra).

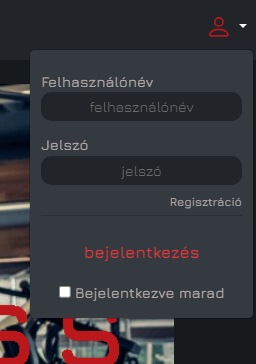
10. ábra: Jegy megjelenítésének kódja



11. ábra: Főoldal

A jobb fenti ember ikonra kattintva előjön egy kis lista, ahol eleinte bejelentkezni tudunk vagy elérni a regisztrációs oldalt (12.ábra ). Bejelentkezést követően itt tudjuk elérni a profilunkat (13.ábra). Az admin joggal rendelkezők innen érik el az Admin oldalt. Valamint egy gyorsgomb is található itt, ami a megvásárolt jegyeinkhez visz egyből.



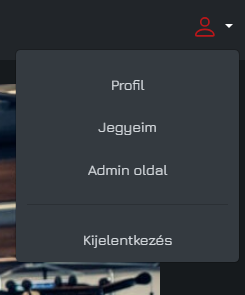


12. ábra: Bejelentkező ablak

Teszteléshez lett felvéve 17 felhasználó, jelenleg mindenkinek a jelszava megegyezik a felhasználónevével:

1. táblázat: Felvett felhasználók

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | Felhasználónév | Név | Jelszó | Jogosultság |
| 3 | Gaborka | Kis Gábor | Gaborka | Trainer |
| 4 | Mat12 | Liks Máté | Mat12 | User |
| 5 | admin | admin | admin | Admin |
| 6 | Jancsi | Kelemen János | Jancsi | User |
| 7 | Pancsi | Kiss Panna | Pancsi | User |
| 8 | Halas | Halász Gábor | Halas | User |
| 9 | Ádámbátyám | Nagy Ádám | Ádámbátyám | User |
| 10 | Zolo | Varga Zoltán | Zolo | Trainer |
| 11 | Zsozso | Balogh Zsófia | Zsozso | User |
| 12 | Dorka | Gál Dorottya | Dorka | Trainer |
| 13 | Mészi | Mészáros Judit | Mészi | User |
| 14 | Marcsi | Orsós Mária | Marcsi | Trainer |
| 15 | Julcsi | Tóth Julianna | Julcsi | User |
| 16 | Boro | Boros Réka | Boro | Trainer |
| 17 | Spider-man | Peter Parker | Spider-man | Trainer |



## Jegyek oldal

A jegyek oldalon megtalálható az összes jegy és bérlet opció, amik közül a felhasználó választhat és vásárolhat. Amikor a felhasználó rámegy erre az oldalra a weboldal küld egy kérést a szervernek, és az visszaküldi neki az összes jegy típus adatát, amit Javascript segítségével megjelenít az oldal. Vásárlás gomba kattintva előjön egy felugró ablak, ahol megadjuk a bankkártya adatainkat (jelenleg nem kell rendes adatoknak lenniük) és a Fizetés gombra nyomva meg is vettük a jegyet és megtalálhatjuk ezt a jegyeinknél.

A jegyek lekérdezése úgy történik, hogy a kérést elküldi szervernek, ahol a TicketControllerben az egyik metódus azt fogja (.ábra) ezek után az a TicektService egyik metódusát hívja meg(.ábra) amitől megkapja a jegyek listáját és ezt a listát küldi vissza a weboldalnak.

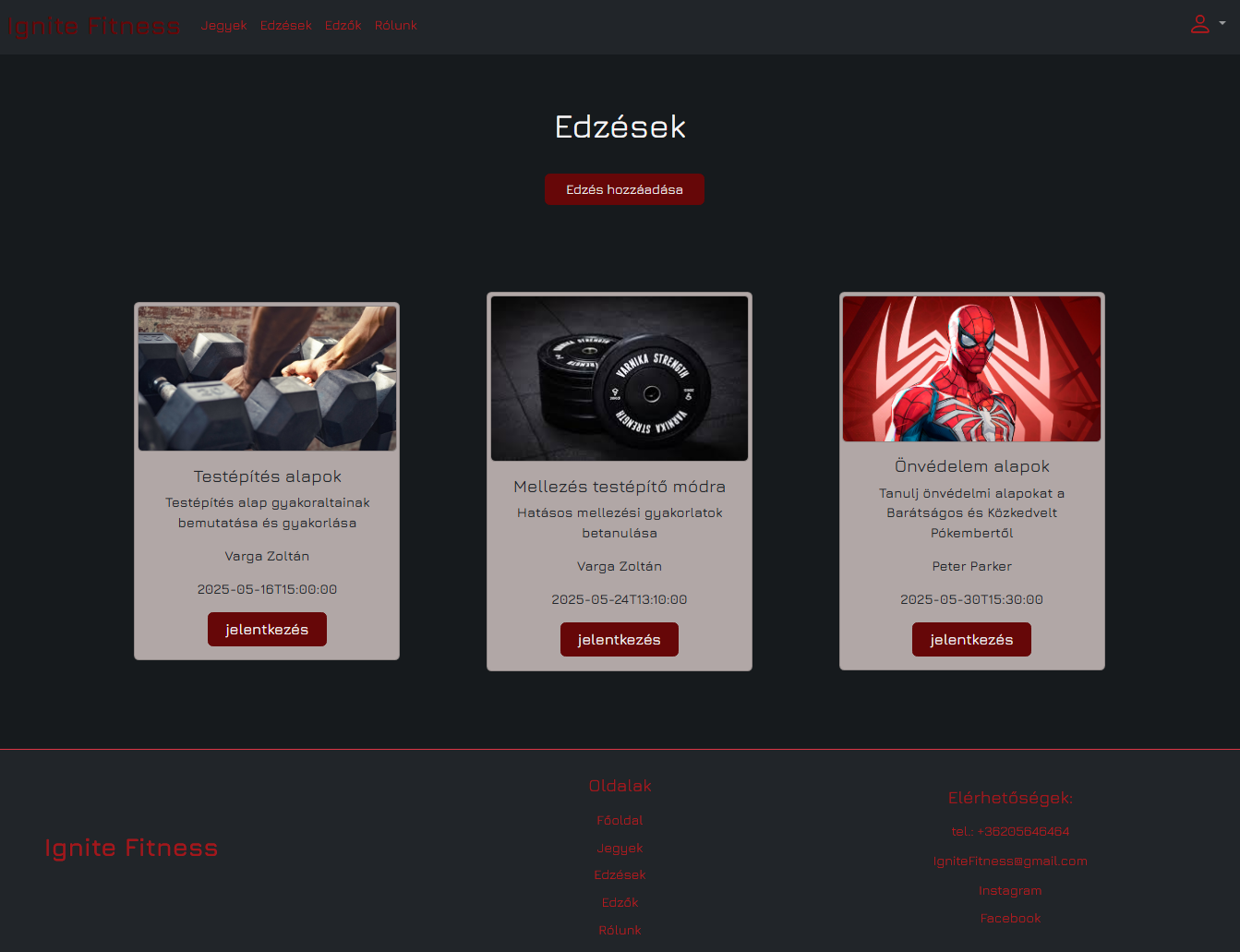
A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen. A TicketService beli metódus lekérdezi az adatbázisból a jegyeket, ezután ezeket egyesével átalakítja megfelelő típusura a Mapper segítségével, ezt beleteszi egy listába és visszaadja a listát a Controllernek.

## Edzések oldal

Az Edzések oldalon az összes jelenleg kiír edzés megtalálható és a jelentkezési opció is rájuk, aminek megnyomásával feliratkozunk az edzésre, ha van „Edzés jegy” -ünk, ha nincs, akkor szól, hogy nem rendelkezünk jeggyel. Az Edzőknek megjelenik külön egy „Edzés hozzáadása” gomb is, amire kattintva előjön egy form ami segítségével fel tudnak vinni új edzéseket.



## Edzők oldal

Edzők oldalon megtalálható az összes regisztrált edző és nekik az elérhetőségük, ha beállították azt maguknak a profiljukon.

## Rólunk oldal

A Rólunk oldalon találhatóak az elérhetőségek, a nyitvatartás valamit a terem címe.

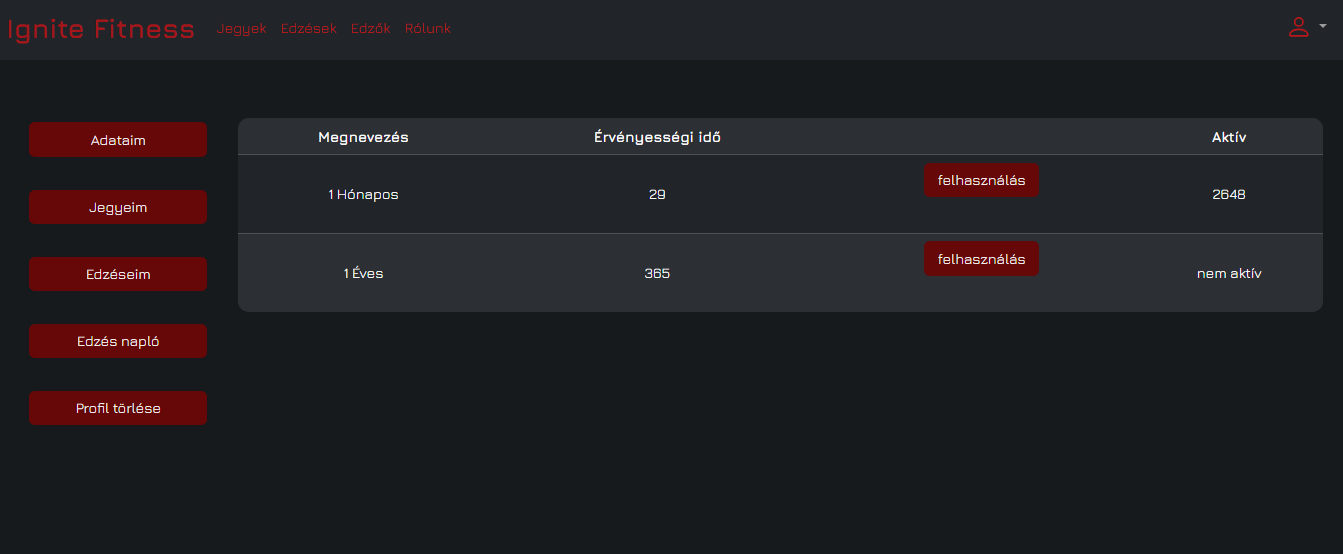
## Profil

Profilunkon eleinte öt gombot találunk, ami az „Adataim”, „Jegyeim”, „Edzéseim”, „Edzés napló”, „Profil törlése”. Az utóbbira kattintva megkérdezi az oldal, hogy biztos törölni akarjuk-e a profilt, ha igenre nyomunk, akkor törlődik a teljes profilunk.

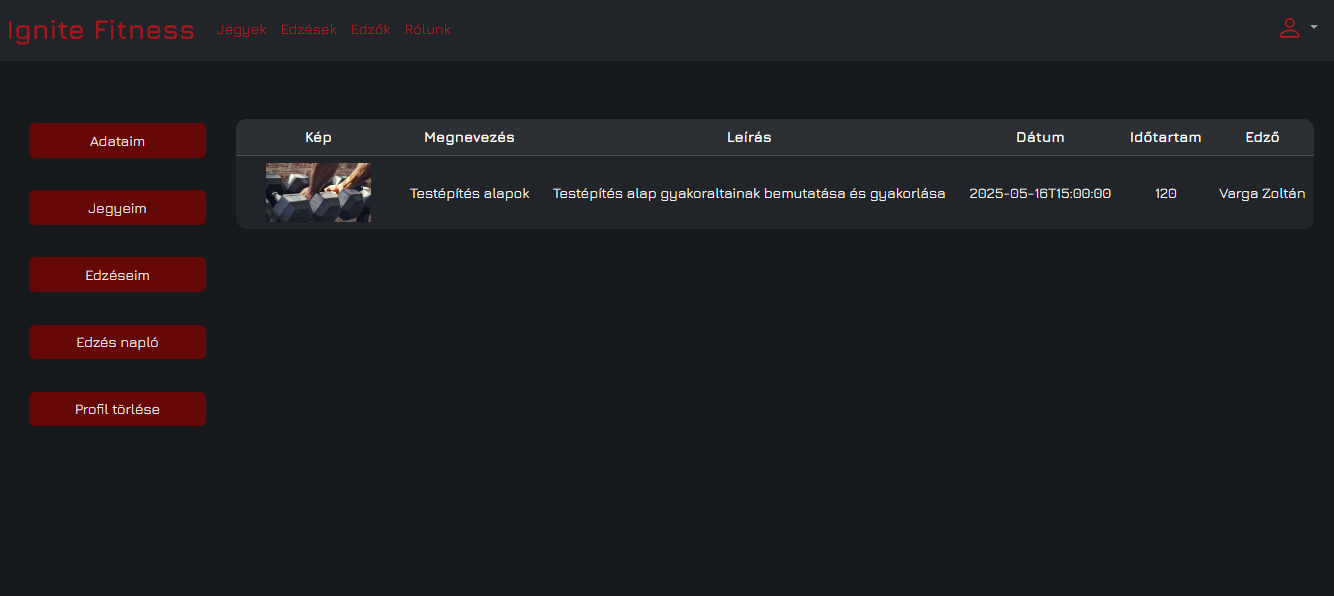
Adataimra kattintva kettő féle táblázat jöhet elő, sima felhasználóknak csak a felhasználó nevüket és a nevüket tartalmazza, edzőknek viszont tartalmazza a telefonszámukat, képesítésüket és a képüket is mellé.

Az adatok megjelenítését kódban úgy oldottam, meg hogy alapból a sima felhasználónak a tábálaztából indulok ki, ezután megnézi a program, hogy a felhasználó edző-e, és ha igen akkor teszi hozzá a kellő sorokat.



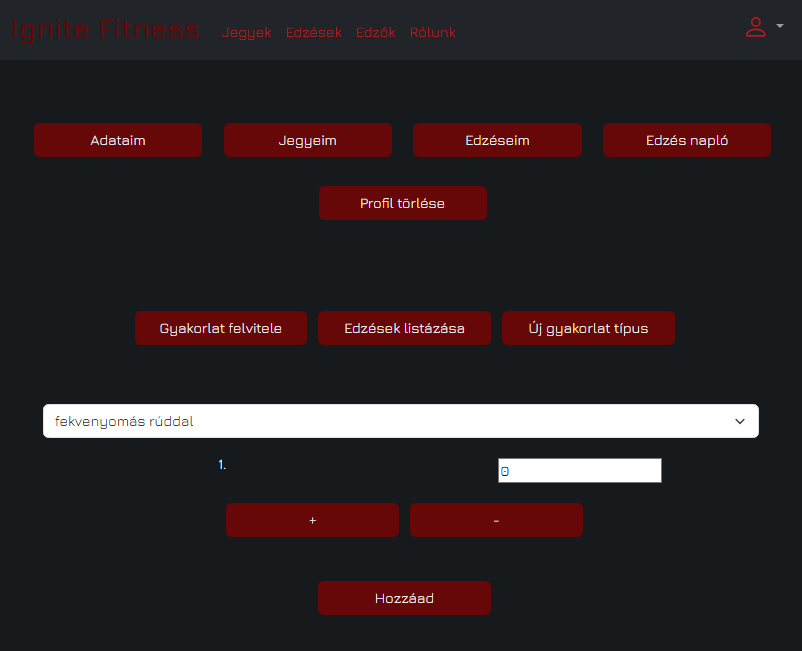
A „Jegyeim” gombbal előhozható a megvásárolt jegyeink listája, itt tudjuk őket felhasználni, amikor kapunk egy 4 számjegyű kódot, amivel beléphetünk a terembe.

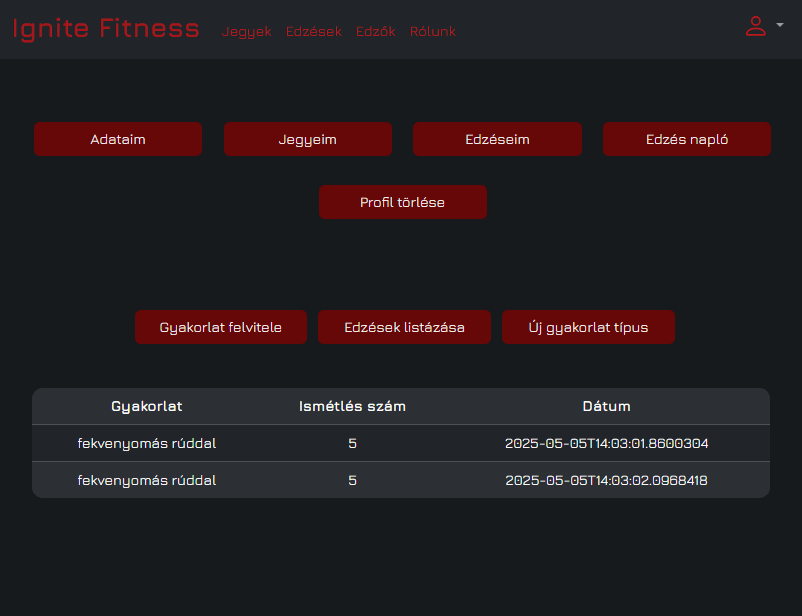
A táblázat megjelenítésénél lefut egy függvény, ami lekéri az aktív jegyeinket, majd ezeknek a hozzáférési kódját beteszi a táblázat megfelelő helyére.

Valamint az „Edzéseim” gombot választva előjön egy táblázat az edzésekkel, amikre jelentkeztünk.

Edzés napló gombbal előhozunk három új gombot a „Gyakorlat felvitele”, „Edzések listázása” és az „Új gyakorlat típus” megnevezéssel.

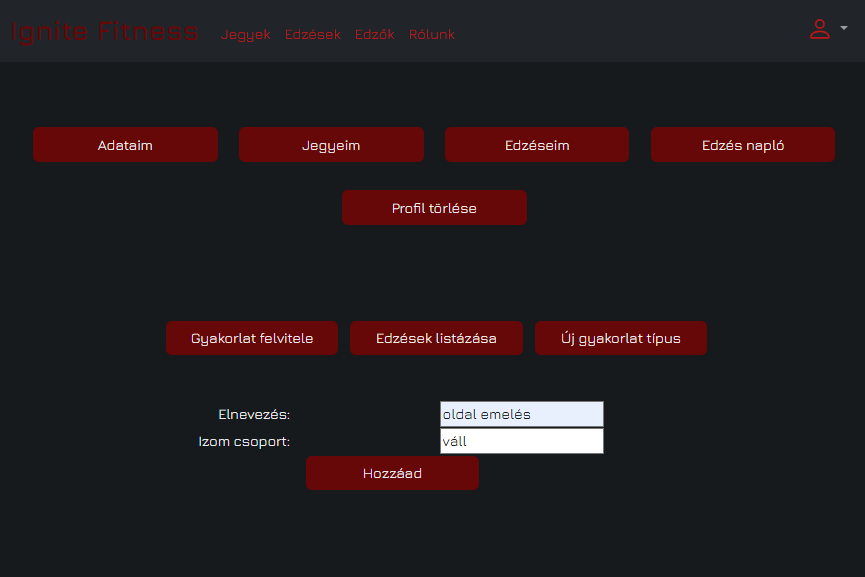
A „Gyakorlat felvitele” gombbal előjön egy legördülő lista, ahol választhatunk gyakorlat típust és alatta megadhatjuk, hogy mennyiszer csináltunk mennyi ismétlést és végül a „Hozzáad” gombbal menthetjük (.ábra).

A plusz és mínusz gombokkal j sort adhatunk hozzá, amit úgy csináltam meg, hogy a már beírt adatok maradjanak meg benne.



Edzések listázása gombbal a már felvitt edzéseinket listázhatjuk ki. (23. ábra)

„Új gyakorlat típus” gombbal felvihetünk új gyakorlatokat, amiket esetleg nem találunk a listában,



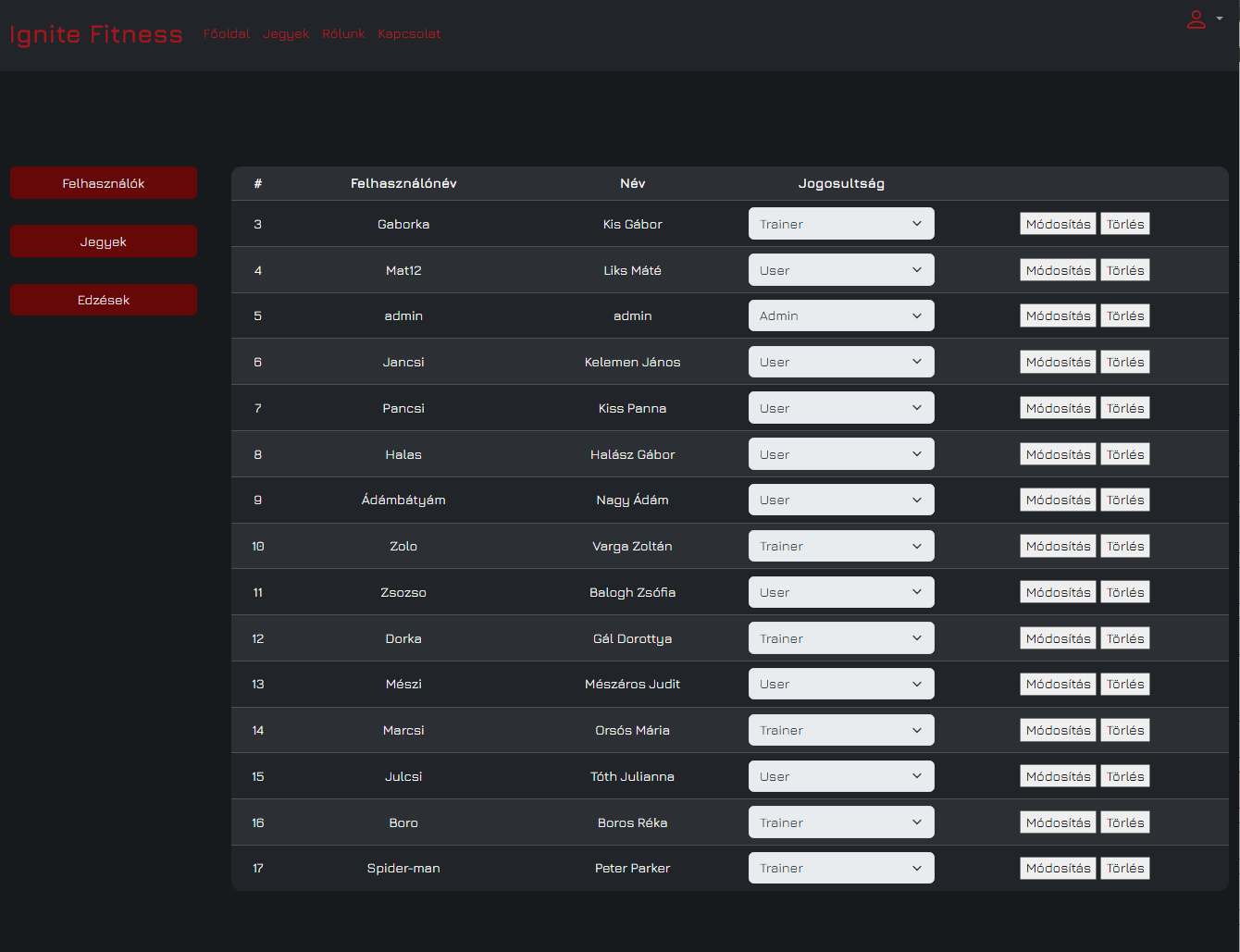
## Admin oldal

Az Admin joggal rendelkezők az Admin oldalra átmenve 3 gombbot találnak „Felhasználók”, „Jegyek” és „Edzések” felirattal.

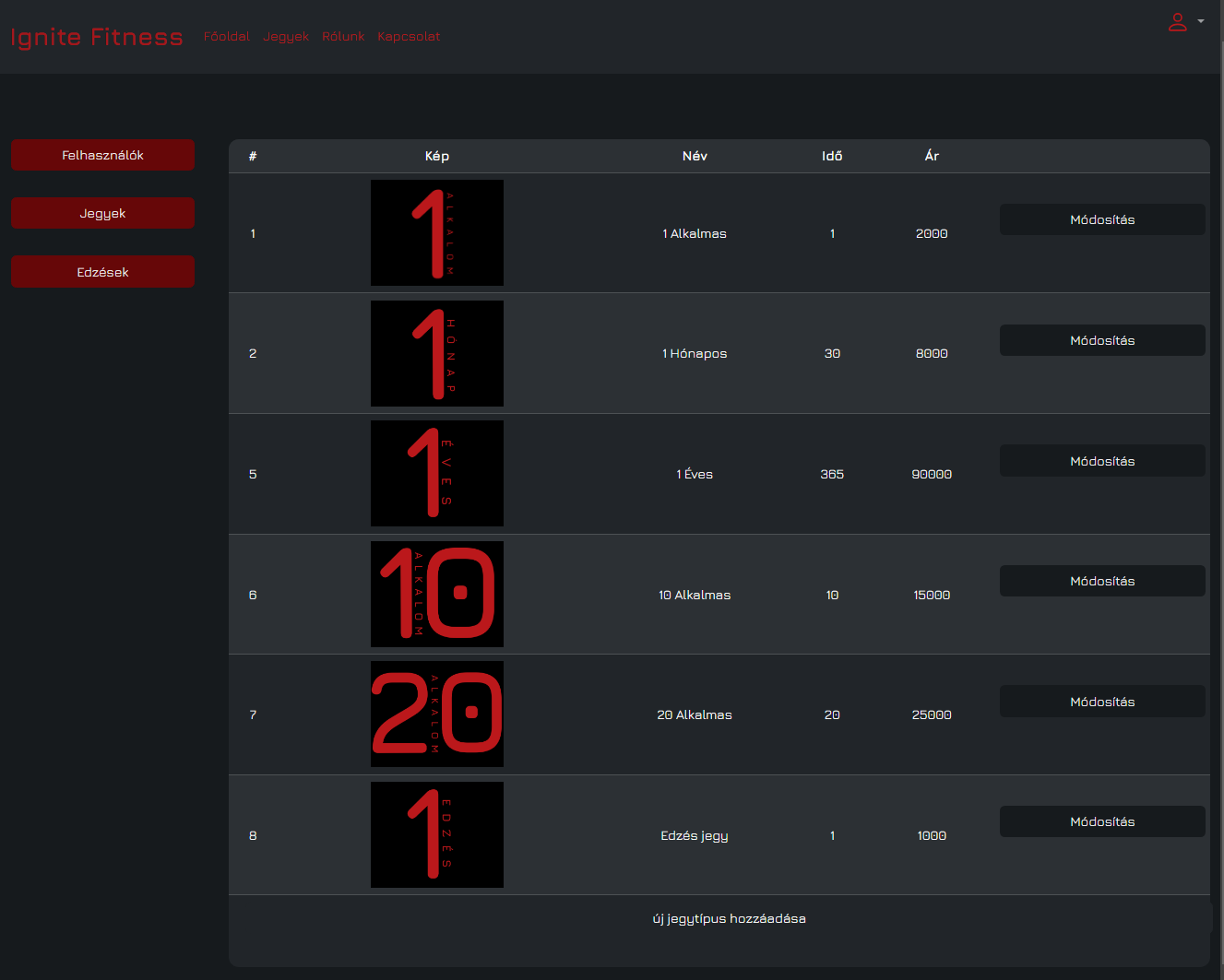
A „Felhasználók” gombra nyomva kapnak egy listát az összes regisztrált felhasználóról és itt tudják módosítani a jogaikat, valamint törölni a fiókjukat.

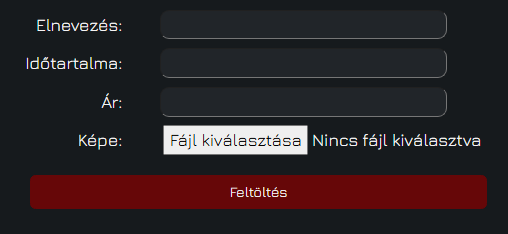
Jog módosításnál, ha Edzőre módosulna akkor fel lesz véve az edzőtáblába az adott személy (.ábra), eleinte egy alap képpel, telefonszám és képesítés nélkül, ezt nekik kell kitölteni a profiljukon.

A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

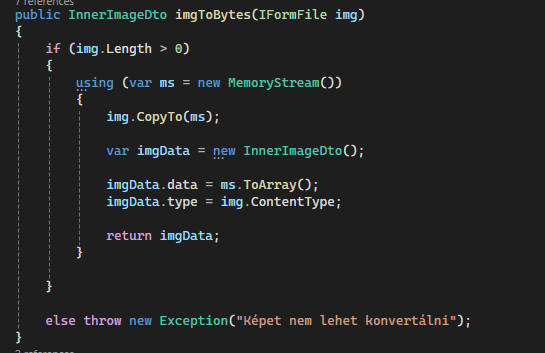
„Jegyek” gombbal előhozhatják az éppen létező jegytípusokat, ezeket tudják módosítani, valamit tudnak új jegyet felvenni az „új jegytípus hozzáadása” opcióval.

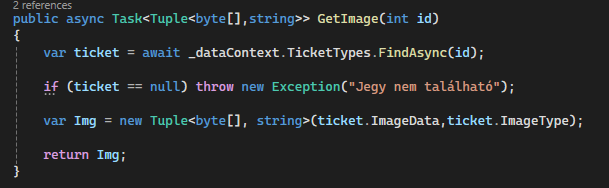


Új jegy típust felvitelénél egy form jelenik meg a táblázat alatt minek a segítségével az admin könnyen felvihet új jegyeket.

Valamit az „Edzések” gombbal az éppen kiírt edzéseket listázza ki egy táblázatban, valamint tud felvinni új edzéseket is, ha erre lenne szükség.



A képek tárolását a következő kóddal oldottam meg, hol az látható, hogy a metódus kap egy fájlt és ezt átalakítja eltárolható formátumba és visszaadja ezt az adatot és a fájl típusát, aminek a segítségével vissza lehet adni a képet.

Képek visszaadását azt úgy oldottam meg, hogy minden controllernek van egy getImage metódusa, aminek az elérését adott jeggyel vagy edzéssel megkapja a weboldal és ettől le tudja kérni a képet az adott adathoz. Ez a metódus visszaad egy byte-string párt, amit a controller a weboldalnak elküld File ként és ott megjeleníthető lesz.

A típusok közti váltást a szerveren az AutoMapper nevezetű csomag könnyítetté meg. Ez egyszerűbbé teszi a különböző egyedi típusok közti konverziót, csupán egy kis konfigurációt igényel használat elött.

A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

# Összefoglalás

Edzőterem menedzsment webalkalmazás fejlesztése című szakdolgozatom keretében megnéztem hasonló weboldalakat, amik azt kínálják, amit én kitaláltam témámnak és megnéztem, hogy ők hogyan valósították meg ezeket és kigyűjtöttem, ami tetszik és nem tetszik ezek weboldalak megvalósításában.

Ezekután összegyűjtöttem milyen technológiákat szeretnék használni az én a webalkalmazásom megvalósításához és milyen technológiák lennének a legalkalmasabbak. Amint ez megvolt, összetettem egy részletesebb tervet, hogy mit és hogyan kéne megvalósítani. a tervezés után jött a webalkalmazás létrehozása, amihez csináltam egy kiszolgáló szervert és hozzá egy felhasználói felületet.

Nyilván még maradt sok tovább fejlesztési lehetőség, mint például fizetéshez egy fizető API integrálása, Edzés naplózást átdolgozni felhasználó barátabbá és jobban használhatóvá. Jelszavak tárolását biztonságosabbá tenni Argon 2 használatával.

Irodalomjegyzék

|  |  |
| --- | --- |
| [1] | T. De Smedt and W. Daelemans, “Pattern for python.,” *The Journal of Machine Learning Research,* vol. 13, no. 1, pp. 2063-2067, 2012. |
| [2] | „Záróvizsga információk,” [Online]. Available: https://mik.uni-pannon.hu/index.php/hu/oktatas/zarovizsga.html. [Hozzáférés dátuma: 04 03 2022]. |
| [3] | D. J. etherall és A. S. Tanenbaum, Computer networks, Pearson Education, 2013. |

Mellékletek

Mappaszerkezet

**GymWebapp**

│

├───frontend

│ │ aboutUs.html

│ │ adminPage.html

│ │ classes.html

│ │ index.html

│ │ profil.html

│ │ purchase.html

│ │ Registration.html

│ │ tickets.html

│ │ trainers.html

│ │

│ ├───bootstrap-5.3.5-dist

│ │ ├───css

│ │ └───js

│ │ └───Jura

│ ├───JS

│ │ admin.js

│ │ api.js

│ │ auth.js

│ │ exerciseLog.js

│ │ fetching.js

│ │ main.js

│ │ profile.js

│ │ reg.js

│ │

│ └───Style

│ adminStyle.css

│ loginReg.css

│ profile.css

│ regStyle.css

│ style.css

│ tickets.css

│

├───GymWebapp

│ │ GymWebapp.sln

│ │

│ ├───.vs

│ │ ├───GymWebapp

│ │ │ ├───config

│ │ │ │ applicationhost.config

│ │ │

│ │ └───ProjectEvaluation

│ └───GymWebapp

│ │ appsettings.Development.json

│ │ appsettings.json

│ │ Gym.db

│ │ Gym.db-shm

│ │ Gym.db-wal

│ │ GymWebapp.csproj

│ │ GymWebapp.csproj.user

│ │ GymWebapp.http

│ │ Program.cs

│ │

│ │ └───Release

│ │ └───net8.0

│ ├───Controllers

│ │ AuthController.cs

│ │ ClassController.cs

│ │ LoggingController.cs

│ │ TicketController.cs

│ │ UserController.cs

│ │

│ ├───Mapper

│ │ MapperConfig.cs

│ │

│ ├───Middlewares

│ │ ExceptionMiddleware.cs

│ │

│ ├───Migrations

│ │ 20250108122030\_Init.cs

│ │ 20250108122030\_Init.Designer.cs

│ │ 20250117124843\_BoughtTicket\_Change.cs

│ │ 20250117124843\_BoughtTicket\_Change.Designer.cs

│ │ 20250117133709\_ActiceTicketTable.cs

│ │ 20250117133709\_ActiceTicketTable.Designer.cs

│ │ 20250129144714\_imageTicketTest.cs

│ │ 20250129144714\_imageTicketTest.Designer.cs

│ │ 20250206130353\_PicturesAndTrainerExpertiseAdded.cs

│ │ 20250206130353\_PicturesAndTrainerExpertiseAdded.Designer.cs

│ │ 20250224115939\_ClassAttendeeTable.cs

│ │ 20250224115939\_ClassAttendeeTable.Designer.cs

│ │ 20250402114201\_Exercise\_log.cs

│ │ 20250402114201\_Exercise\_log.Designer.cs

│ │ 20250427123523\_activeTicketRework.cs

│ │ 20250427123523\_activeTicketRework.Designer.cs

│ │ 20250427123850\_activeTicketRework2.cs

│ │ 20250427123850\_activeTicketRework2.Designer.cs

│ │ 20250501093026\_classNoPrice.cs

│ │ 20250501093026\_classNoPrice.Designer.cs

│ │ DataContextModelSnapshot.cs

│ │

│ ├───Model

│ │ │ DataContext.cs

│ │ │

│ │ ├───Data

│ │ │ ActiveTicket.cs

│ │ │ BougthTicket.cs

│ │ │ Class.cs

│ │ │ ClassAttendee.cs

│ │ │ Exercise.cs

│ │ │ Log.cs

│ │ │ TicketType.cs

│ │ │ Trainer.cs

│ │ │ User.cs

│ │ │

│ │ └───Dtos

│ │ ClassDto.cs

│ │ ImageDto.cs

│ │ LogDto.cs

│ │ TicketDto.cs

│ │ UserDto.cs

│ │

│ ├───obj

│ │ ├───Debug

│ │ │ └───net8.0

│ │ └───Release

│ │ └───net8.0

│ ├───Properties

│ │ launchSettings.json

│ │

│ └───Services

│ AuthService.cs

│ ClassService.cs

│ ImgService.cs

│ LoggingService.cs

│ TicketService.cs

│ UserService.cs

Ábrajegyzék

[1. ábra: A Flex Gym főoldala 16](#_Toc197409580)

[2. ábra: A Flex Gym „Edzőink” oldala 17](#_Toc197409581)

[3. ábra: A Planet Fitness főoldala 18](#_Toc197409582)

[4. ábra: Az alkalmazás Use Case diagramja 19](#_Toc197409583)

[5. ábra: Az adatbázis kezdetleges modellje 21](#_Toc197409584)

[6. ábra: Az adatbázis jelenlegi modellje 22](#_Toc197409585)

[8. ábra: Adatbázis összerakása 24](#_Toc197409586)

[7. ábra: Osztály adatbázishoz 24](#_Toc197409587)

[9. ábra: Oldaltérkép 25](#_Toc197409588)

[10. ábra: Jegy megjelenítésének kódja 26](#_Toc197409589)

[11. ábra: Főoldal 27](#_Toc197409590)

[12. ábra: Bejelentkező ablak 28](#_Toc197409591)

[11. ábra 29](#_Toc197409592)

[10. ábra 29](#_Toc197409593)

[13. ábra: Vásárló felugró ablak 30](#_Toc197409594)

[12. ábra: Jegyek oldal 30](#_Toc197409595)

Táblázatjegyzék

[1. táblázat:Felvett felhasználók 27](#_Toc196905914)

1. A megfelelő rész aláhúzandó. [↑](#footnote-ref-1)