



Vas Megyei Szakképzési Centrum  
Nádasy Tamás Technikum és Kollégium

# **PROJEKTFELADAT**

**Thrive Tribe**

**Csánki Adrián, Somogyi Bálint András,  
Szár Máté**

**Konzulens:  
Balics Gábor**

**2024**

# Nyilatkozat

Alulírott, Csánki Adrián, Somogyi Bálint András, Szár Máté kijelentjük, hogy a Thrive Tribe című projektfeladat kidolgozása a saját munkánk, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtuk fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeink közös munkánk, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásunk szerint hitelesek.

Csepreg, 2024.04.19.

---

hallgató

---

hallgató

---

hallgató

# Kivonat

## Thrive Tribe

Szakdolgozatunk célja, hogy támogassuk azokat az egyéneket, akik fizikai átalakulásra törekednek, kihasználva egy online platform lehetőségeit. Azon korban, ahol a virtuális tér a személyes fejlődés legfőbb terepe, elengedhetetlen egy digitális menedékhely létrehozása, amely bölcsességeket és gyakorlati eszközöket kínál az életmódváltáshoz.

Az online platform lehetőségeinek kiaknázása lehetővé teszi, hogy az életmódváltásra vágyók számára könnyen elérhetővé váljanak a releváns információk és eszközök az egészséges étkezéssel kapcsolatban. Ez a digitális menedékhely nemcsak egy hely, ahol az emberek tájékozódhatnak és inspirációt meríthetnek az életmódváltáshoz, hanem aktív támogatást is nyújthat számukra ezen az úton.

Nehéz megtalálni a megfelelő információkat, amik alapján el tudják kezdeni az életmód változásukat az emberek. Ezért az oldalunkon igyekszünk a mai napig a legösszefoglalóbb és leghitelesebb tájékoztatást nyújtani, hogy az látogatók minden szükséges ismerettel rendelkezzenek az egészséges életmód elindításához. Az oldalon az egészséges táplálkozásról, valamint az edzők által összeállított edzéstervekből informálódhat.

# **Abstract**

## Thrive Tribe

Our thesis aims to support individuals striving for physical transformation by leveraging the possibilities of an online platform. In an era where virtual space is the primary arena for personal development, establishing a digital sanctuary is essential, offering wisdom and practical tools for lifestyle changes.

Harnessing the opportunities of the online platform enables easy access to relevant information and resources for those desiring lifestyle changes, particularly concerning healthy eating. This digital refuge serves not only as a place for people to navigate and find inspiration for lifestyle changes but also provides active support along their journey.

Finding the right information to initiate lifestyle changes can be challenging for individuals. Therefore, our website strives to provide the most comprehensive and credible information available to ensure that visitors have all the necessary knowledge to kickstart a healthy lifestyle. Visitors can learn about healthy eating and access workout plans curated by trainers on the website.

# Tartalomjegyzék

1.	Bevezetés.....	7
2.	Hasonló weboldalak .....	8
3.	Felhasználói dokumentáció .....	9
3.1.	Hardver és Szoftver igény .....	10
3.2.	Belépés és regisztráció.....	10
3.3.	Főoldal és tartalma .....	11
3.4.	Funkciók ismertetése .....	13
3.4.1.	Főoldal.....	14
3.4.2.	Edző fül .....	14
3.4.3.	Edzés fül .....	15
3.4.4.	Étrend fül .....	18
3.4.5.	Térkép fül .....	20
3.4.6.	Profil szerkesztés .....	21
3.4.7.	Admin lehetőségek .....	23
4.	Fejlesztői dokumentáció .....	24
4.1.	Az adatbázis táblái és kapcsolatai .....	24
4.1.1.	Users tábla .....	26
4.1.2.	UserRole tábla .....	26
4.1.3.	Role tábla.....	26
4.1.4.	Training tábla.....	27
4.1.5.	CoachExperience tábla .....	27
4.1.6.	CityCentrum tábla .....	27
4.1.7.	Diet tábla .....	28
4.1.8.	Food tábla .....	28

4.1.9.	food_role tábla .....	28
4.1.10.	Location tábla .....	28
4.1.11.	Map tábla .....	29
4.2.	Model View Controller (MVC) .....	30
4.3.	Felhasználók jogkezelése .....	32
4.4.	API hívások .....	33
4.5.	Jelszó titkosítás és módosítás .....	36
4.6.	Regex .....	36
4.7.	Teszt dokumentáció .....	37
5.	A weboldal design-ja .....	39
6.	Összefoglalás .....	40
6.1.	A szakdolgozat célja .....	40
6.2.	Megvalósítás .....	40
6.3.	Fejlesztési lehetőségek .....	41
7.	Irodalomjegyzék .....	42
8.	Mellékletek .....	43
8.1.	Programkódok .....	43
8.1.1.	Jelszó titkosítás .....	43
8.1.2.	Jelszó módosítás .....	44
8.1.3.	Access_Component .....	45
8.1.4.	Regex .....	47

# 1. Bevezetés

A 2024-es záróvizsgánk beadandójához egy olyan weboldalt választottunk, amiben az egyének változásokat hozhatnak a mindennapi életükben. Fő célja nem csupán az egyén személyes fejlődési útjának elősegítése, hanem az egészség mélyebb megértése és egy összetartóbb közösség formálása. A választásunk azért erre a projektre esett, mivel mindannyiunknak fontos az egészség és a weboldalt mások számára könnyedén el lehet érni.

Edzőként lehetőség van edzéstervek készítésére, amelyek minden felhasználó számára elérhetőek. Ezek által segítséget nyújthat a felhasználóknak, akik fejlődést szeretnének elérni a fitness életükben. Tanulmányaik, tapasztalataik és szakterületük meghatározásával könnyítik meg a felhasználóknak, hogy megtalálják a nekik megfelelő edzőt.

Felhasználóként hozzáférhet a különböző étrendekhez, használhatja az edzéseket, megtekintheti az edzőket, a térkép segítségével edzőtermeket és edző parkokat tud keresni a közelében.

A szoftver fejlesztéséhez a 'Next.js'-ben szerzett ismereteinket és a MySQL adatbázis-kezelési készségeinket használtuk fel.

- A weboldalhoz a WebStorm<sup>1</sup> programozási környezetet használtuk 'TypeScript'nyelven írtuk, illetve a Laravel [\[6\]](#) alkalmaztuk.
- Laragont<sup>2</sup> a lokális környezet kialakításához.
- Az adatbázis tervezéséhez Figmát [\[4\]](#) használtunk, az elkészítéséhez pedig a Laragonon belüli HeidiSQL segített.
- Az adatok tárolása MySQL szerverben történik.
- A programban használt képeket, illetve ikonokat az [irodalomjegyzékben](#) jelölt oldalról és a Figma weboldal segítségével készítettük.

---

<sup>1</sup> A WebStorm egy fejlett fejlesztői környezet (IDE) beépített hibakereső, tesztelő és verziókezelő eszközökkel.

<sup>2</sup> A Laragon egy könnyen használható fejlesztői környezet, amely lehetővé teszi a webserverek, adatbázisok és programozási nyelvek gyors telepítését és konfigurálását webfejlesztési projektekhez.

## 2. Hasonló weboldalak

Jelenleg nagyon kevés hasonló weboldal található az interneten, pedig egy több funkciót magába foglaló oldal mindenki számára kedvezőbb. Ez egy olyan webhely, ami kombinálja más weboldalak tulajdonságait, amik az interneten találhatóak.

A felhasználó és edző bejelentkezés vagy regisztrációt követően megtalálhatják ezeket a fűleket és el is érhetik azokat. Az edzők akár edzéstervet hozhatnak létre, amikből a felhasználókat segíthetik.

Más oldalak csak egy diéta ismertetésével vagy edzésterv bemutatásával foglalkozik, de mi több tulajdonságot szeretnénk egy weblapon alkalmazni. Ez a portál egy letisztult és egyszerűbb áttekintést nyújt a felhasználó számára, így egy helyen mindent megtalálhatnak, hogy elkezdjék életüket egy egészségesebb irányba vinni, ez volt a fő indokunk a téma választásában.



### 3. Felhasználói dokumentáció

A programunk célja, hogy segítsen a felhasználóknak a fizikai céljaik elérésében és a táplálkozás terén.

Lehetőségeink felhasználóink:

- Saját profil szerkesztése
- Edzéstervek olvasása
- Étrendek megtekintése
- Edzőtermek keresése a Térképen
- Böngészheti az elérhető edzőket

Lehetőségeink edzőként:

- Edzéstervek létrehozás
- Saját edző profil létrehozása
- Saját profil szerkesztése
- Edzőtermek keresése a Térképen
- Étrendek megtekintése
- Adatmódosítás

Lehetőségeink adminként:

- Adatok módosítása és törlése az Admin fülön

### 3.1. Hardver és Szoftver igény

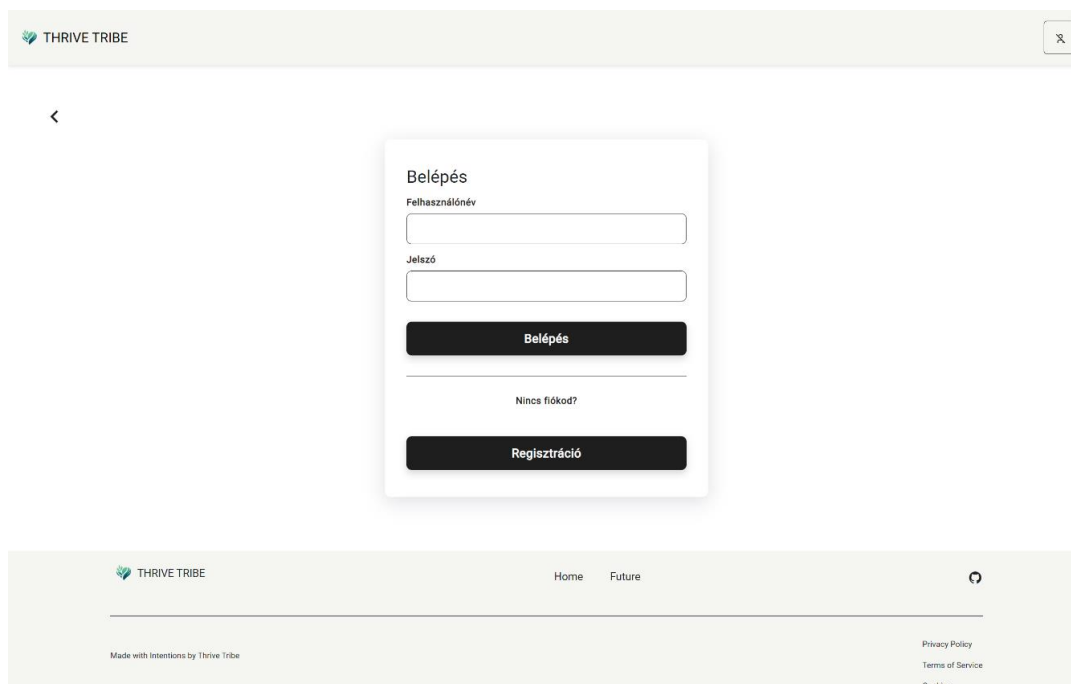
A következő specifikáció ajánlott:

- Legalább 2GHz-es processzor
- 16Gb szabad memória
- Windows 7 vagy annál újabb operációs rendszer
- 1.5Gb szabad tárhely
- Minimum 8.1-es PHP verzió
- A Node.js-nek legalább a 20-as verziójára van szükség
- Nélkülözhetetlen egy backend webszerver (esetlegesen backend migrációk futtatása)
- Frontend npm script futtatása

### 3.2. Belépés és regisztráció

A regisztrációhoz navigálnunk kell a jobb felső sarokban található ikonra, melyre kattintva megnyílik a regisztrációs felület. Itt lehetőségünk van regisztrálni, legyen szó egyszerű felhasználói vagy edzői fiókról. Ha edzőként szeretnénk regisztrálni, a felületen található „Edző vagy?” feliratú csúszkát kell aktiválnunk.

A belépéshez szükséges megadni a megfelelő felhasználónév és jelszó páros az adott mezőkbe. Ezután a „Belépés” gombra kattintva, vagy az „Enter” billentyű lenyomásával tovább lépünk. Ekkor visszajutunk a főoldalra, ahol már láthatóvá válnak az elérhető fülek, melyek segítségével könnyedén navigálhatunk az oldal különböző szolgáltatásai között.



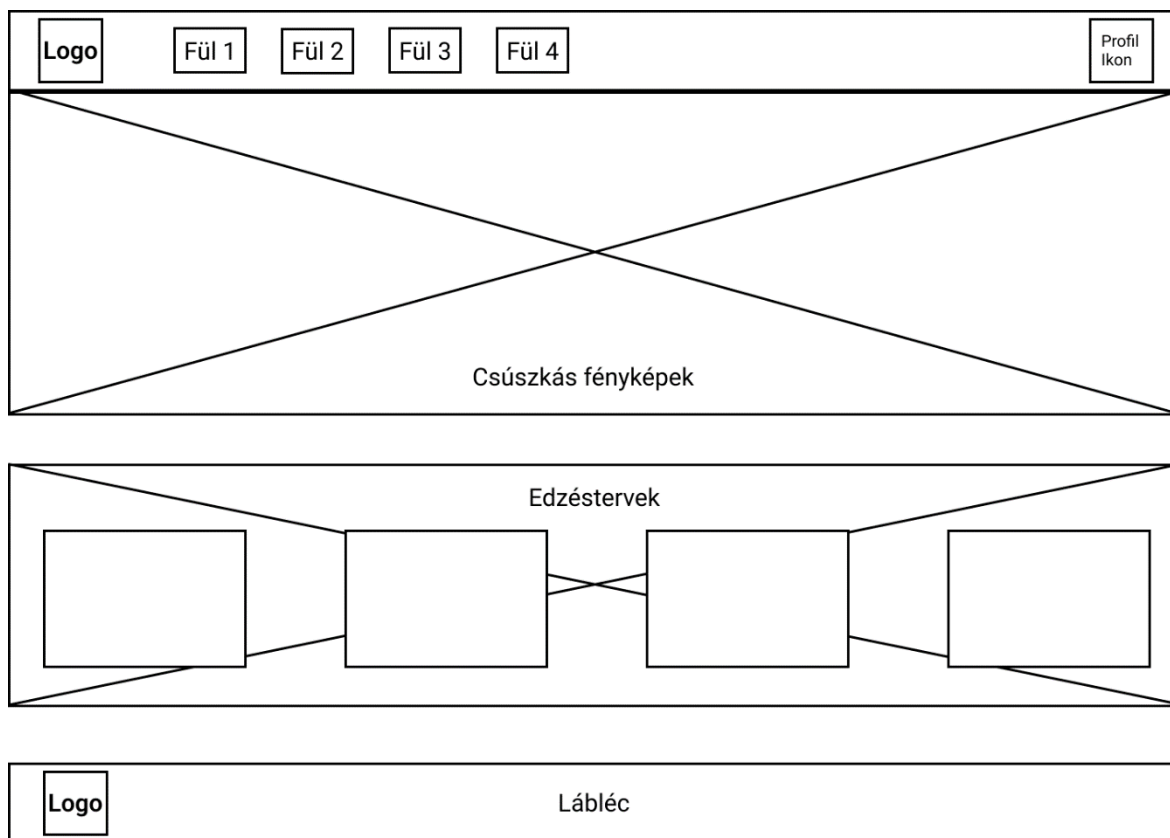
*1. ábra a belépés oldaláról*

### 3.3. Főoldal és tartalma

Az oldal felső részén található a navigációs menüsor, amelyen a logó, az elérhető fülek, valamint a bejelentkezésre szolgáló ikon helyezkedik el. A fülek minden további oldalon láthatóak, így könnyen vissza lehet térni a kívánt részhez.

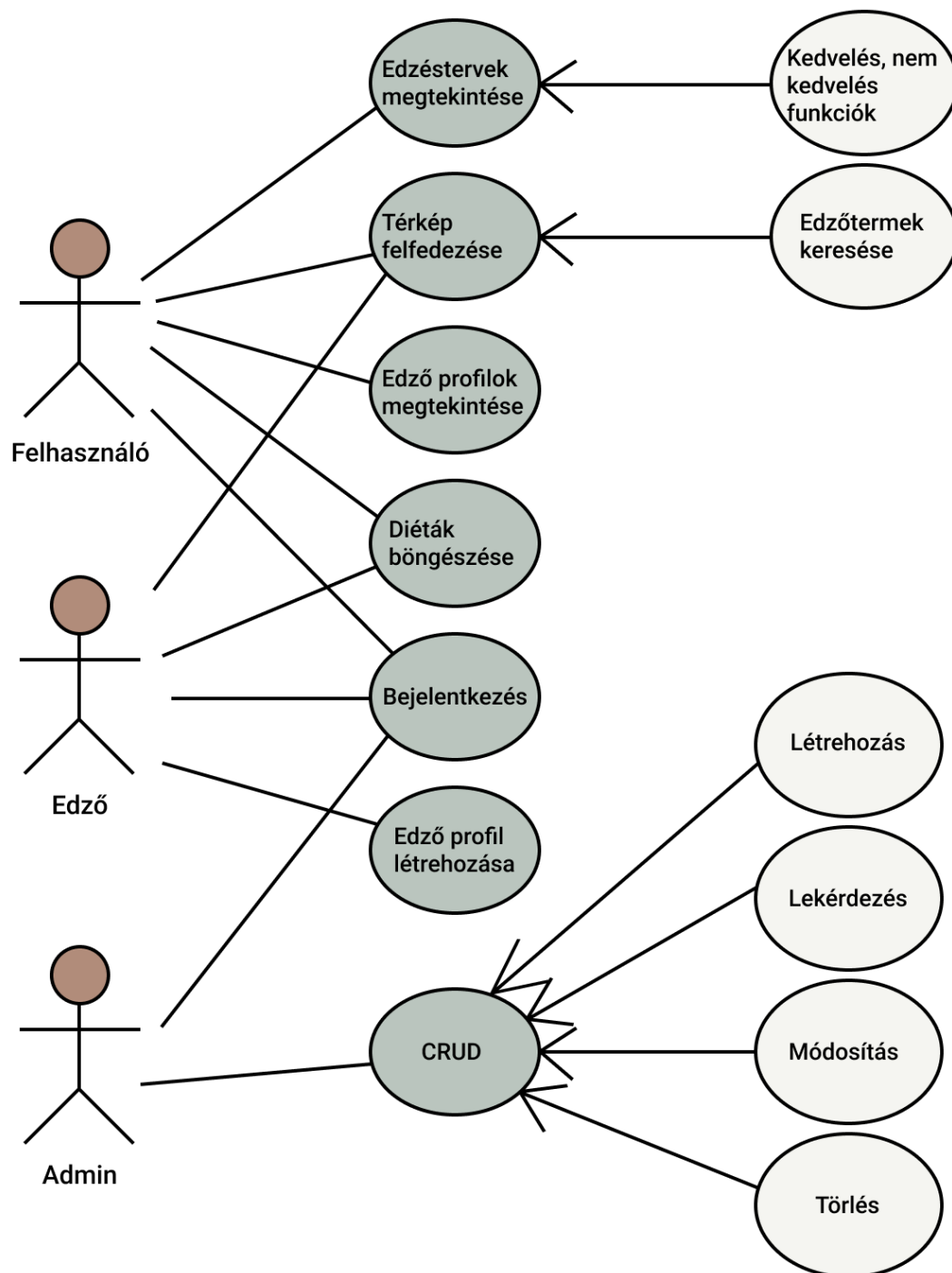
A fejléc alatt elhelyezkedik egy úgynevezett „körhinta”, ahol három kép követi egymást. Ezek a képek inspiráló motivációs szövegeket hordoznak, hogy lendületet adjanak az oldal felhasználóinak. Ez alatt található meg, négy különböző kártyában a legnépszerűbb edzéstervek, hogy könnyen választhassanak az aktuális edzésprogramok közül.

Az oldal alján található a lábléc, mely tartalmazza a logót, illetve két opciót: a "Home"-t és a "Future"-t. A Home-ra kattintva visszaviszi a felhasználókat a főoldalra, a Future-re kattintva, pedig a weboldal fejlesztési lehetőségei jellennek meg egy új aloldalon. Egy Github ikon található még a jobb sarokban, ahol elérhető a projekt. Végül, itt helyezkedik el a szlogenünk is, hogy inspiráljon és közelebb hozza a közösséget.



*2. ábra a főoldal dizájn tervéről*

### 3.4. Funkciók ismertetése



3. Use Case Diagram

### 3.4.1. Főoldal

A főoldal alapértelmezetten nem tartalmaz minden funkciót, ezek eléréséhez először be kell jelentkeznünk.

Miután sikeresen bejelentkeztünk, a fejlécben megjelenik az összes fül, függetlenül a felhasználó típusától, kivéve az adminisztrátorét, aki csak az „Admin” fület látja.

Lejjebb görgetve választhatunk a legjobb négy edzésterv egyike közül, és ha rájuk kattintunk, az adott edzésterv oldalára irányít az minket, feltéve, hogy be vagyunk jelentkezve. A Home fül visszarepít minket a főoldalra, a Future pedig a jövőbeli fejlesztések oldalára vezet minket. A GitHub ikon megnyomásával pedig az adattár oldalára jutunk el, ahol további információkat találhatunk a projekttel kapcsolatban.

### 3.4.2. Edző fül

Rákattintva az „Edzők” fülre az edzőket találjuk.

Felhasználóként az edzők által létrehozott profilokat tekinthetjük meg, így kedvünkre válogathatunk a különböző edzők között. Itt egy rövid bevezető vár minket, amiben leírjuk miért a Thrive Tribe edzőit érdemes választani. A fülön létrehozhatják saját edző profiljaikat az edzők és hirdetés céljából tudásukat oszthatják meg a többi felhasználóval. Itt minden fontos információt megtudhatunk az edzőről. Kezdve a nevével, az életkorát, végzettségeit, készségeit és tapasztalatait. Nem mellesleg az elérhetőségeit, ami egy telefonszámot és egy e-mail címet tartalmaz.

Az edzők által létrehozott profilok böngészése rendkívül hasznos lehet, mert lehetőséget nyújt arra, hogy alaposan megismerje az egyes edzőket, mielőtt döntést hozna a felhasználó, hogy kivel veszi fel a kapcsolatot.

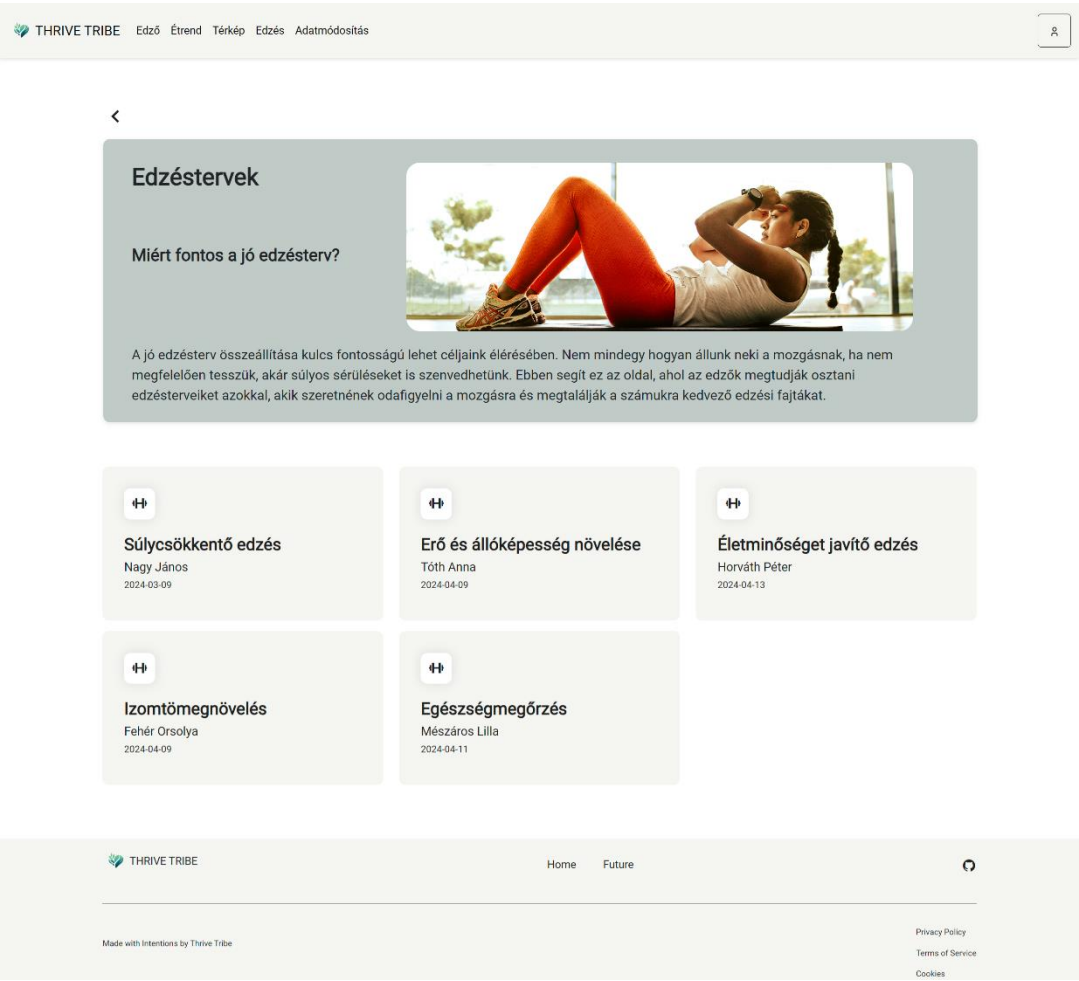


4. ábra az Edző oldalról

### 3.4.3. Edzés fül

Az „Edzés” fülre kattintva az edzéstervek széles választékát találjuk megjelenítve. Egy újabb bevezető fogad minket, mely hangsúlyozza a megfelelő edzéstervek fontosságát a célok elérésében és a sérülések megelőzésében. Az oldal lehetőséget kínál az edzők számára, hogy megoszthassák terveiket azokkal a felhasználókkal, akik rendszeresen foglalkoznak testmozgással vagy éppen azoknak, akik fizikai változásokat szeretnének elérni az életükben. Ezáltal az edzők segíthetnek megtalálni az ideális edzésmódszereket az érdeklődők számára.

Itt böngészhetünk az összes elérhető edzésterv között, ahol láthatóak azok címei, szerzőik és szerkesztési dátumaik. Amennyiben kiválasztunk egy edzéstervet, részletes leírást kapunk róla, valamint megtekinthetjük a „like” és „dislike” gombok számait is, amelyek segítenek mérlegelni az adott tervezet hatékonyságát azáltal, hogy mutatják, hány felhasználó találta hasznosnak azt.



5. ábra az Edzéstervek oldalról



THRIVE TRIBE

Edző

Étrend

Térkép


Edzés

Adatmódosítás

Edzés

Súlycsökkentés

Súlycsökkentő edzés



1. Bemelegítés (5 perc kardiógépen majd 5 perc gimnasztika, teljes testet átmozgató torna, fejkörzésekkel, vállkörzésekkel, karkörzésekkel, törzshajlításokkal, törzskörzésekkel, lábhajlításokkal, csípőkörzésekkel stb.) 2. Vállból nyomás-sétáló kitorrés szuperszett: Úlj egy támlás padra és a súlyzót magas tartásban fogd meg. Innen engedd le vállig, majd ismét nyújtsd vissza. 3X15 ismétlést végezz. Ha kész egy sorozat, akkor végezz sétáló kitorrés, 20 lépéssel, majd jöhetsz az újabb vállból nyomás. Tárogatás vízszintes padon – hasprés szuperszett Feküdj egy vízszintes padra, tartsd a súlyzókat magad előtt függőlegesen tartott nyújtott karban. Engedd a karokat két oldalra, vállmagasságban tartva, majd emeld vissza. Ebből végezz 3x15 ismétlést, az egyes sorozatok után pedig fekéldj a talajra és végezz 20 hasprés. Mellhez húzás – padra lépés szuperszett Úlj a lehúzó gépbe, fogd meg a fogantyút szélesen és húzd le egyenes háttal, a vállak lent tartásával a mellizmod felső részéhez, majd engedd vissza. 3X15 ismétlés a cél, a pihenőidőben pedig egy legalább térdmagasságú padra lépegsz fel és le, mindkét lábbal 10x elkezdve a folyamatot. Bicepsz hajlítás rúddal-lábmelés szuperszett A bicepszhajlításnál figyelj arra, hogy a könyököd ne mozduljon előre és hátra, a tested előtt nyújtva, vállszéles fogásból indítsd a mozdulatot, majd hajlítsd a könyököt és ismét nyújtsd ki. 3X15 ismétlést végezz, a sorozatok között pedig könyöktámaszos tolózkodó gépnél végezz 10-15 térdhúzást Tricepsz nyújtás csigán egyenes rúddal- lábtoló Végezz 3x15 tricepsz nyújtást rúddal, a sorozatok között pedig fekéldj a lábtoló gépbe és végezz 20 lábhajlítást, minél mélyebbre. Végezhetesz egy lábás lábtoló gyakorlatot is, úgy 15-15 ismétlés menjen mindkét lábra 3. Nyújtás és Relaxáció (10 perc): Lassú, mély nyújtó gyakorlatok végzése az izmok lazítására és a rugalmasság javítására. Légzőgyakorlatok és rövid meditáció a mentális és érzelmi kiegyensúlyozottságért.

1

1

- Nagy János

2024-03-09

THRIVE TRIBE

Home

Future

Made with Intentions by Thrive Tribe

Privacy Policy

Terms of Service

Cookies

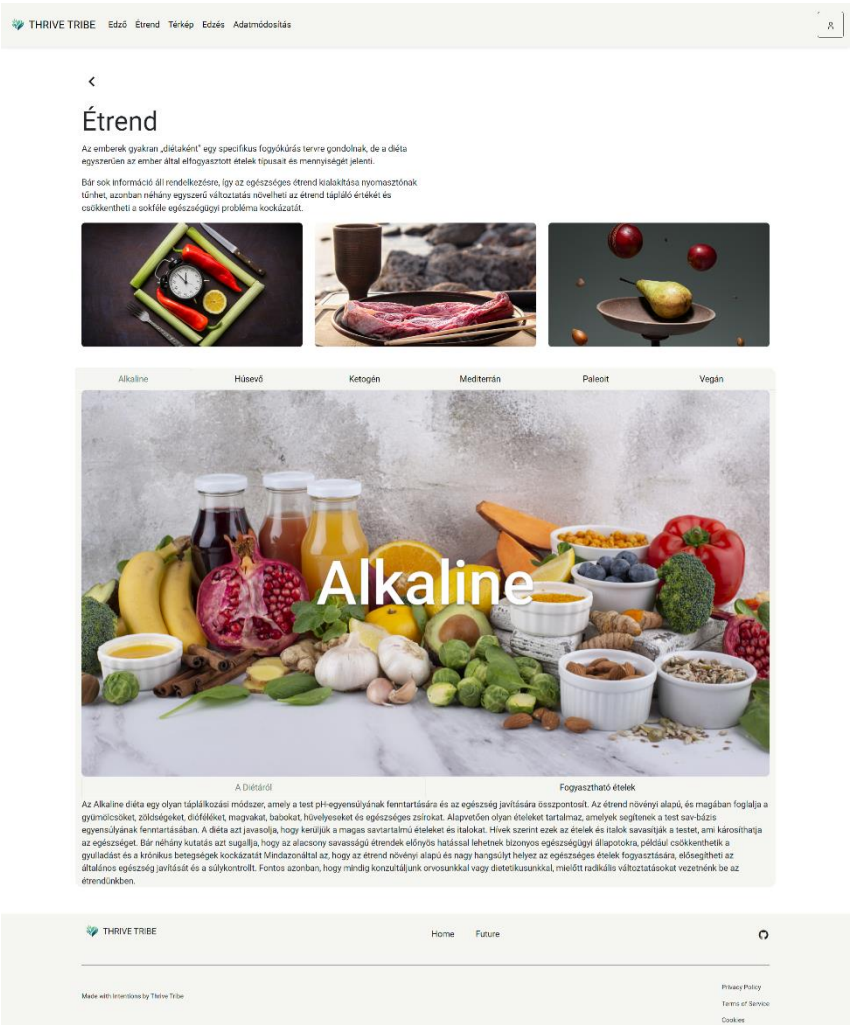
6. ábra egy edzéstervről

### **3.4.4. Étrend fül**

Amikor rákattintunk az „Étrend” fülre, az étrend típusokat láthatjuk megjelenve a képernyőn.

Az étrend típusok egyesével elkülönítve vannak „tabokban”, amelyekre kattintva könnyen navigálhatunk. Ezekben a tabokban található egy kép az étrend nevével, valamint további két tab-ban részletes leírás olvasható az adott étrendről, illetve egy másikban felsorolásra kerülnek az étrendhez tartozó fogyasztható ételek is.

Ez az oldal kiemelkedően fontos szerepet tölt be, hiszen segítséget nyújt azoknak, akik az ideális étrendet keresik mindennapi életük során. Ezáltal hozzájárulhatnak egy egészségesebb életmód kialakításához és kitűzött célok eléréséhez, miközben egy teljes körű tájékoztatást nyújtunk az étkezési lehetőségekről és azok hatásairól is.

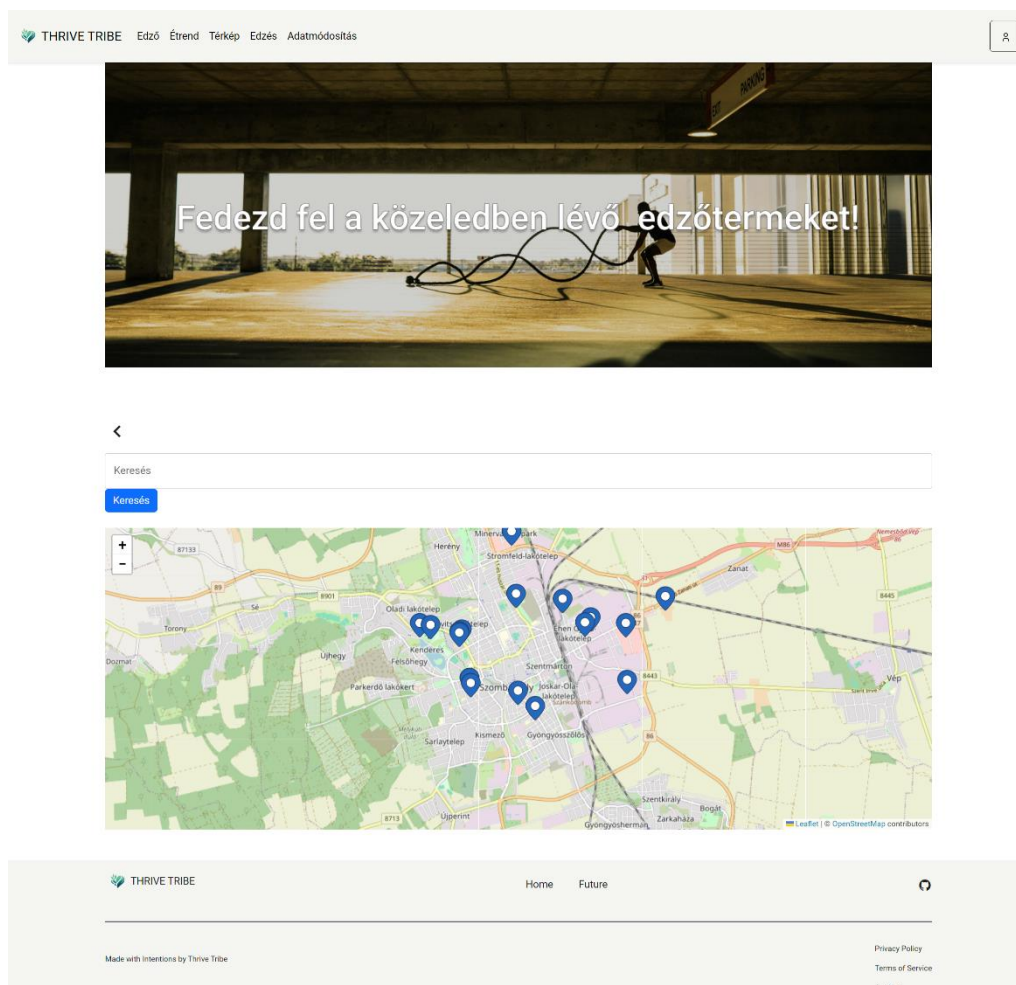


7. ábra az Étrend oldalról

### 3.4.5. Térkép fül

A "Térkép" fülre való kattintást követően a felhasználó a térképet látja megjelenni. Ebben a menüpontban egy kép fogadja, mely alatt elhelyezkedik a kereső és maga a térkép. Ezen az oldalon lehetőség nyílik az edzőtermek keresésére Vas Vármegye területén belül. A térkép funkcióit kihasználva tudunk közelíteni, ugyanakkor hatékonyabb lehet a város nevének beírása a keresősávba, ami azonnal átnavigál minket a kiválasztott város központjára, ahol megjelennek az edzőtermek.


Az edzőtermet egy helyjelölő jelzi a térképen, amelyre kattintva bővebb információkat találhatunk az adott létesítményről, beleértve a nevét, pontos címét és nyitvatartási idejét, amely a hét minden napjáról nyújt tájékoztatást. Az edzőtermek elhelyezkedése koordináták alapján van meghatározva a térképen, ami lehető legpontosabban van megadva.




8. ábra a Térkép oldalról

### 3.4.6. Profil szerkesztés

Minden egyes fióknak megadatik a lehetőség arra, hogy módosítsa saját adatait. Ezek közé tartoznak a vezetéknev, keresztnév, e-mail cím, életkor, súly és magasság. Fontos megjegyezni, hogy a felhasználónév nem változtatható meg, hiszen ez minden felhasználó számára egyedi azonosítót jelent.


**THRIVE TRIBE**

[Edző](#)
[Étrend](#)
[Térkép](#)
[Edzés](#)
[Adatmódosítás](#)



Adatmódosítás

Vezetéknév

Teszt

Keresztnév

Elek

Email

user1@gmail.com

Életkor

18

Súly (kg)


70

Magasság (cm)


190

[Jelszó módosítása!](#)

Mentés


**THRIVE TRIBE**

[Home](#)
[Future](#)



Made with Intentions by Thrive Tribe

[Privacy Policy](#)  
[Terms of Service](#)  
[Cookies](#)

9. ábra az Adatmódosító oldalról

### 3.4.7. Admin lehetőségek

Az admin -nak lehetősége van módosítani, létrehozni, illetve törölni az adatokat, kivéve az ételek és az étrendek tabból amikben csak létrehozni nem tud. Ezt mind az admin fülön teheti meg.

Miután az admin bejelentkezett egyedül csak az admin fület láthatja és erre a fülre lépve az oldalon lévő összes adatot kezelheti, ami az oldalhoz, illetve felhasználókhoz tartoznak.

Ezek az adatok szintén egy tab-ban vannak meghatározva. Ezek nem mások, mint a felhasználók, jogok, edzéstervek, térkép és a hozzá tartozó fokok a városközpontokhoz, ételek, étrendek, edzők, városközpontok.

A létrehozás gomb a már meglévő adatok felett helyezkedik el és ezzel gombbal is hozhatjuk létre az adott tabhoz tartozó adatokat.

A már létrehozott adatok mellett találjuk a Szerkesztés és Törlés gombokat. A szerkesztés gombbal szerkeszthetjük az ott lévő adatok, illetőleg törölhetjük is azokat a törlés gombbal.

THRIVE TRIBE Admin

<

Felhasználók
Jogok
Edzéstervek
Térkép
Fokok
Ételek
Étrendek
Edzők
Városok
Üzenőfal

Létrehozás

ID	Email	FirstName	LastName	Height	Weight	Age	Phone	FavouriteDietId	FavouriteTrainingId	CoachExperienceId	LastLogin	
31	admin1@gmail.com	Admin1	Admin1			15	+36308238625					<div>Szerkesztés</div> <div>Törlés</div>
38	user1@gmail.com	Elek	Teszt	190	70	18						<div>Szerkesztés</div> <div>Törlés</div>
39	nagy.janos1977@gmail.com	János	Nagy			47	+36304448910			37		<div>Szerkesztés</div> <div>Törlés</div>
40	kovacs_eva1989@gmail.com	Éva	Kovács			35	+36207439175			38		<div>Szerkesztés</div> <div>Törlés</div>
41	anna2.toth@outlook.com	Anna	Tóth			28	+36700315503			39		<div>Szerkesztés</div> <div>Törlés</div>
42	horvath_peterx2@gmail.com	Péter	Horváth			32	+36208661089			40		<div>Szerkesztés</div> <div>Törlés</div>
43	orsolya.feher98@gmail.com	Orsolya	Fehér			26	+36301147203			41		<div>Szerkesztés</div> <div>Törlés</div>
44	lilla17mesz@gmail.com	Lilla	Mészáros			34	+36204315477			42		<div>Szerkesztés</div> <div>Törlés</div>

10. ábra az Admin felületről

## 4. Fejlesztői dokumentáció

### 4.1. Az adatbázis táblái és kapcsolatai

Az alkalmazáshoz tartozó adatok tárolására balint-ntszki nevű adatbázis nyújt lehetőséget.

Az adatbázisnak a lokális szerveren kell futnia jelszó nélkül. Jelenleg az adatbázis 9 táblából és közöttük 2 kapcsolatból áll. Továbbá alapból tartalmaz adatokat, amelyek szükségesek a működésének teszteléséhez.

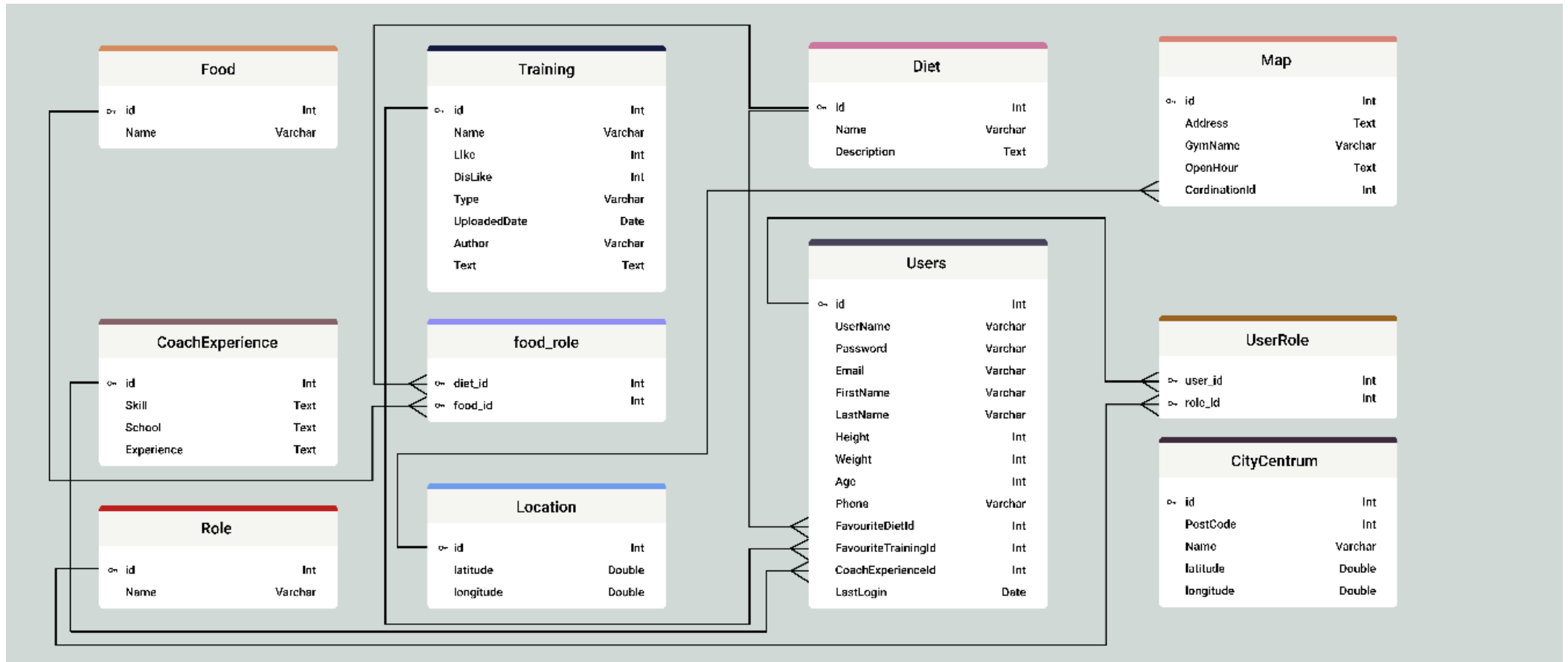
A Food tábla és a Diet tábla közötti kapcsolatot egy food\_role nevű kapcsolótábla segítségével valósítjuk meg.

A Users tábla és a Role tábla közötti kapcsolatot egy user\_role nevű kapcsolótábla segítségével állítjuk fel.

A Map tábla kapcsolatban van a Location táblával, melyben tároljuk az edzőtermek koordinátáit, így lehetőség van azok helyét pontosan meghatározni és megjeleníteni.

A Users tábla kapcsolódik a CoachExperience táblához, mely tartalmazza az edzők profiljait, így lehetőség van az edzők tapasztalatainak és a felhasználói profiljuknak a tárolására és kezelésére. Ez a kapcsolat segít abban, hogy az alkalmazásban megjeleníthessük az edzők szakmai tapasztalatait és profilinformációit.





11. ábra az adatbázis struktúrájáról

#### 4.1.1. Users tábla

→ A felhasználók tárolására szolgáló tábla.

Mezőnév	Típus	Megjegyzés
<b>id</b>	Int, PK, NN, AI	Felhasználói azonosító
<b>UserName</b>	Varchar, NN	Felhasználónév
<b>Password</b>	Varchar, NN	Jelszó
<b>Email</b>	Varchar, NN	Email cím
<b>FirstName</b>	Varchar, NN	Keresztnév
<b>LastName</b>	Varchar, NN	Vezetéknév
<b>Height</b>	Int	Magasság
<b>Weight</b>	Int	Testsúly
<b>Age</b>	Int	Életkor
<b>Phone</b>	Varchar	Telefonszám
<b>FavouriteDietId</b>	Int, FK	Kedvenc diéta azonosítója
<b>FavouriteTrainingId</b>	Int, FK	Kedvenc edzés azonosítója
<b>CoachExperienceId</b>	Int, FK	Edző tapasztalat azonosítója
<b>LastLogin</b>	DATE	Utolsó belépés

#### 4.1.2. UserRole tábla

→ A felhasználó és felhasználó jogai kapcsolattábla.

Mezőnév	Típus	Megjegyzés
<b>user_id</b>	Int, PK, NN	Felhasználó azonosítója
<b>role_id</b>	Int, PK, NN	Jog azonosítója

#### 4.1.3. Role tábla

→ A felhasználói jogok tárolására szolgáló tábla.

Mezőnév	Típus	Megjegyzés
<b>id</b>	Int, PK, NN, AI	Jog azonosítója
<b>Name</b>	Varchar, NN	Jog megnevezése

#### 4.1.4. Training tábla

→ Az edzéstervek szolgáló tábla.

Mezőnév	Típus	Megjegyzés
<b>id</b>	Int, PK, NN, AI	Edzésterv azonosítója
<b>Name</b>	Varchar, NN	Edzés neve
<b>Text</b>	Text, NN	Edzés szövege
<b>Like</b>	Int	Kedvelés
<b>DisLike</b>	Int	Nem kedvelés
<b>Type</b>	Varchar, NN	Típus
<b>UploadedDate</b>	Date, NN	Feltöltés dátuma
<b>Author</b>	Varchar, NN	Szerző

#### 4.1.5. CoachExperience tábla

→ Az edző profil tárolására szolgáló tábla.

Mezőnév	Típus	Megjegyzés
<b>id</b>	Int, PK, NN, AI	Edző tapasztalat azonosítója
<b>Skill</b>	Text	Készség
<b>School</b>	Text	Végzettség
<b>Experience</b>	Text	Tapasztalat

#### 4.1.6. CityCentrum tábla

→ A városközpontok tárolására szolgáló tábla.

Mezőnév	Típus	Megjegyzés
<b>id</b>	Int, PK, NN, AI	Városközpont azonosítója
<b>PostCode</b>	Int, NN	Irányítószám
<b>Name</b>	Varchar, NN	Város neve
<b>latitude</b>	Double, NN	Földrajzi szélesség
<b>longitude</b>	Double, NN	Földrajzi hosszúság

#### 4.1.7. Diet tábla

→ Az étrendek tárolására szolgáló tábla.

Mezőnév	Típus	Megjegyzés
<b>id</b>	Int, PK, NN, AI	Diéta azonosítója
<b>Name</b>	Varchar, NN	Diéta megnevezése
<b>Description</b>	Text, NN	Leírás

#### 4.1.8. Food tábla

→ Az ételek tárolására szolgáló tábla.

Mezőnév	Típus	Megjegyzés
<b>id</b>	Int, PK, NN, AI	Étel azonosítója
<b>Name</b>	Varchar, NN	Étel neve

#### 4.1.9. food\_role tábla

→ Az ételek és az étrendek kapcsoló táblája.

Mezőnév	Típus	Megjegyzés
<b>diet_id</b>	Int, PK, NN	Diéta azonosítója
<b>food_id</b>	Int, PK, NN	Étel azonosítója

#### 4.1.10. Location tábla

→ A koordináták tárolására szolgáló tábla.

Mezőnév	Típus	Megjegyzés
<b>id</b>	Int, PK, NN, AI	Elhelyezkedés azonosítója
<b>latitude</b>	Double, NN	Földrajzi szélesség
<b>longitude</b>	Double, NN	Földrajzi hosszúság

#### 4.1.11. Map tábla

→ Az edzőtermek adatainak tárolására szolgáló tábla.

Mezőnév	Típus	Megjegyzés
<b>id</b>	Int, PK, NN, AI	Edzőterem azonosítója
<b>Address</b>	Text, NN	Edzőterem címe
<b>GymName</b>	Varchar, NN	Név
<b>OpenHour</b>	Text, NN	Nyitvatartás
<b>CordinationId</b>	Int, FK, NN	Elhelyezkedés azonosítója

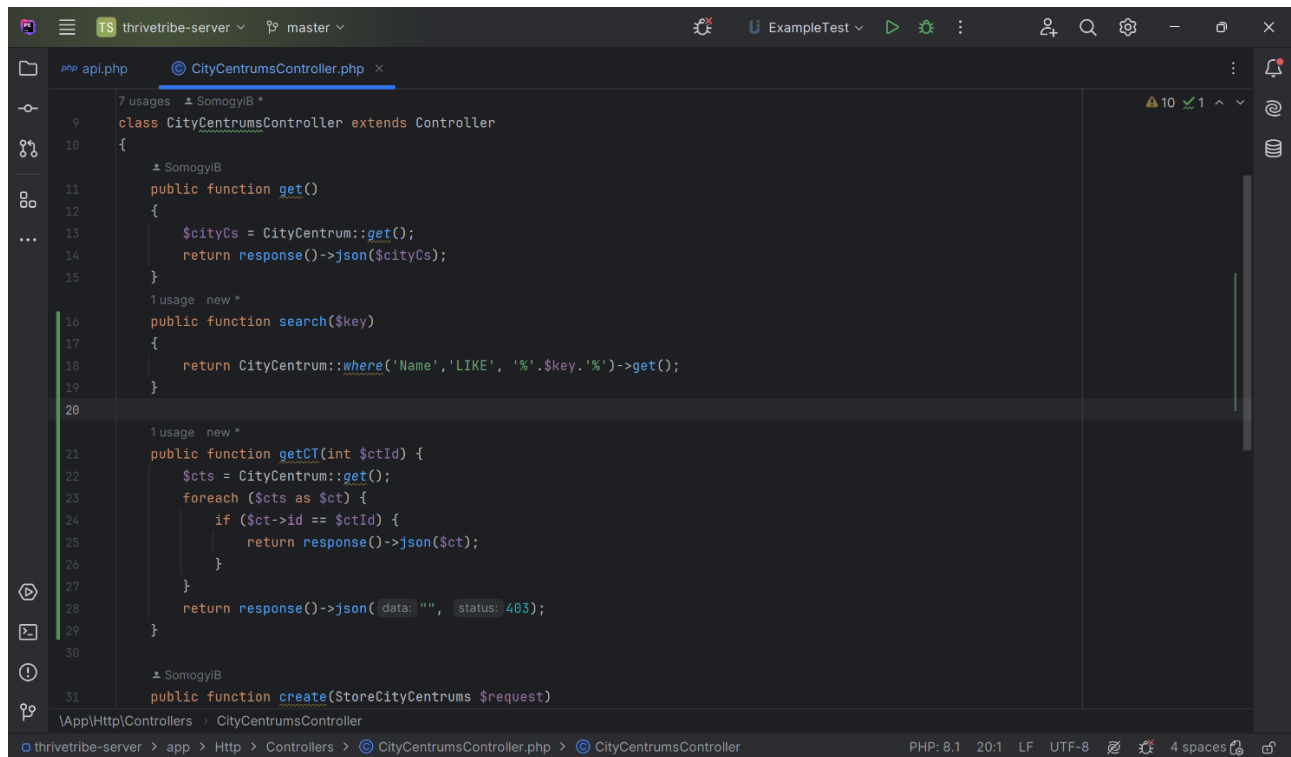
## 4.2. Model View Controller (MVC)

A Model View Controller, magyarul Model Nézet Vezérlő egy olyan séma, amely az alkalmazásadatokat és a vezérlési logikát három különálló komponensre – modellre, nézetre és vezérlőre – választja szét, így az egyes összetevők egymástól függetlenül módosíthatók.

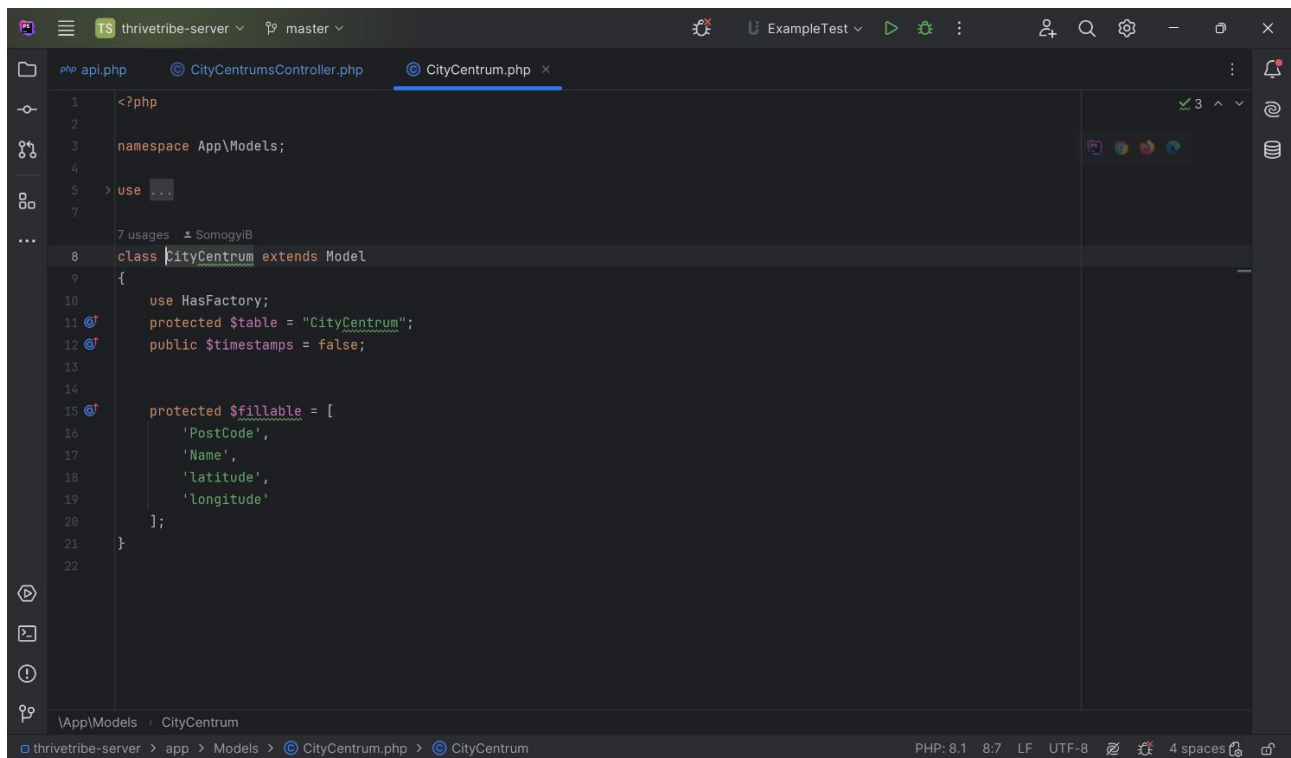
- A modell (Modell) adatokat szolgáltat, és állapotának megváltoztatásával válaszol a vezérlőparancsokra.
- A Nézet (View) felelős azért, hogy a modelladatokat megjelenítse a felhasználó számára a modellváltozások hatására.
- A Vezérlő (Controller) értelmezi a felhasználó tevékenységét, értesítve a modellt a változtatások szükségességéről.

A szakdolgozat backendjéhez Laravel-t használtunk, ami az MVC architektúrára épül. Erre a példánk a következő:

- A CityCentrum modell a városközpontok adatainak tárolására szolgál. A \$fillable tömbben felsorolod azokat az attribútumokat, amelyeket lehet módosítani a modellben.
- A CityCentrumsController vezérlő pedig a városközpontokkal kapcsolatos kéréseket kezeli. Például, a get függvény lekéri az összes városközpontot, a create függvény új városközpontot hoz létre, az update függvény frissíti egy meglévő városközpont adatait, míg a delete függvény törli a városközpontot.
- View: A felhasználói felületet React segítségével hozzuk létre TypeScript nyelven



12. ábra a CityCentrumController-ről



13. ábra a CityCentrum-ról

### 4.3. Felhasználók jogkezelése

Az `isValidView` függvény megvizsgálja, hogy a `userType` paraméter által meghatározott felhasználótípus jogosult-e az adott `viewname` nézet megtekintésére. Ehhez össze kell hasonlítani a `viewname`-et a megfelelő felhasználótípus által megtekinthető nézetek listájával.

Az `isValidView` függvénynek a következő bemeneti paraméterekkel rendelkezik:

- `viewname`: Az az oldal neve, amely megnyitásra került.
- `userType`: Az a felhasználó típusa, akinek a jogosultságát ellenőrizni szeretnénk.

A nézetek tartalmazzák azt, hogy melyik oldalt láthatják:

- `adminNezetek`: Az adminisztrátor típusú felhasználók által megtekinthető oldalak neveit tartalmazó lista.
- `edzoNezetek`: Az edző típusú felhasználók által megtekinthető oldalak neveit tartalmazó lista.
- `felhasznaloNezetek`: A felhasználó típusú felhasználók által megtekinthető oldalak neveit tartalmazó lista.

Az `isValidView` függvény a következőképpen működik:

1. Ellenőrizze, hogy a `userType` értéke melyik típusba tartozik: `admin`, `edző` vagy `felhasználó`.
2. Ha az adott `userType` típusú felhasználó az adminisztrátor, ellenőrizze, hogy a `viewname` szerepel-e az `adminNezetek` listában.
3. Ha az adott `userType` típusú felhasználó az edző, ellenőrizze, hogy a `viewname` szerepel-e az `edzoNezetek` listában.
4. Ha az adott `userType` típusú felhasználó a felhasználó, ellenőrizze, hogy a `viewname` szerepel-e a `felhasznaloNezetek` listában.

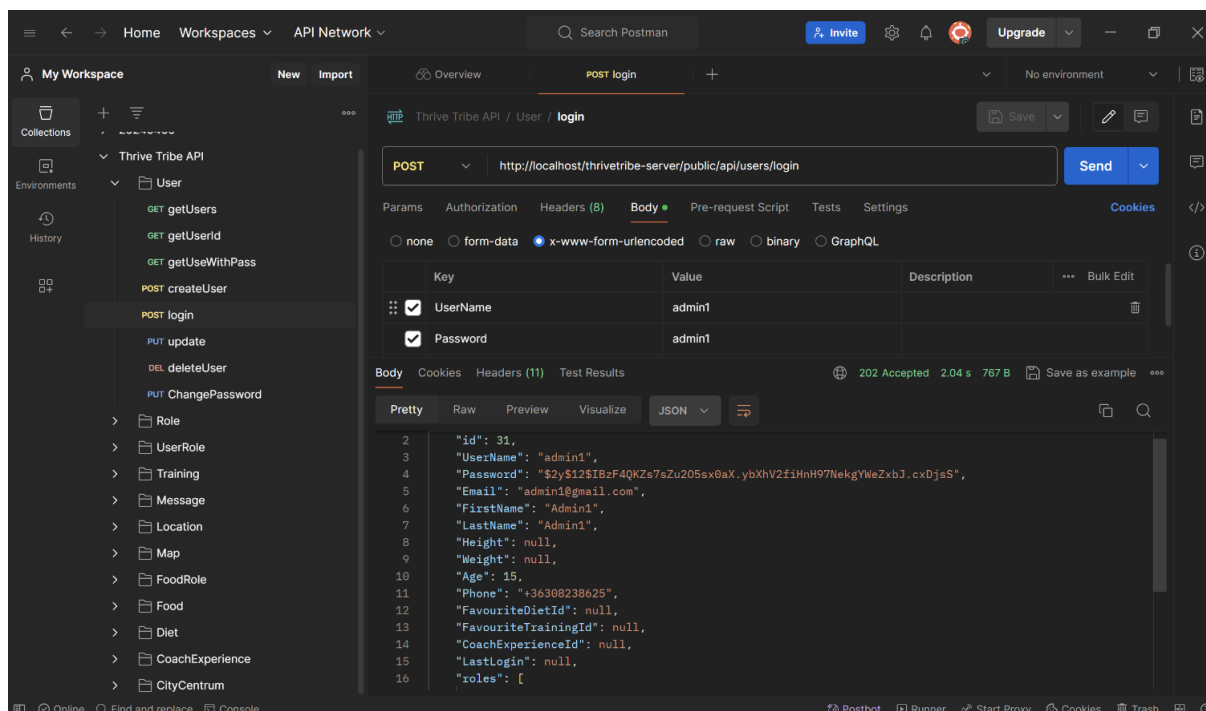


5. Ha a viewname szerepel a megfelelő nézetlistában, akkor a függvény true értékkel tér vissza, jelezve, hogy a felhasználó jogosult az adott nézet megtekintésére. Ellenkező esetben false értékkel tér vissza.

## 4.4. API hívások

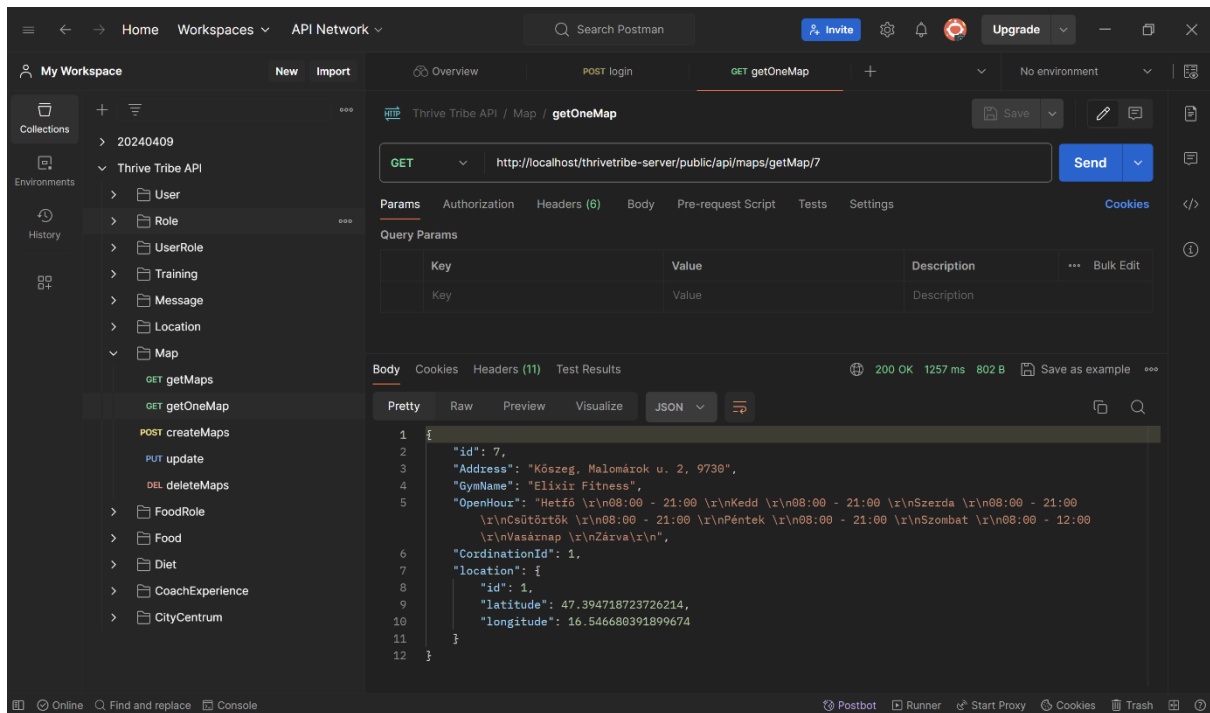
Az API (Application Programming Interface) hívások olyan kéréseket jelentenek, amelyeket egy alkalmazás küld egy szervernek vagy egy szolgáltatásnak az adatok eléréséhez vagy módosításához.

- http protokollt használjuk.
- localhost: Ez a számítógép neve vagy IP címe, ahol a szerver fut. A localhost egy speciális hivatkozás, amely mindig a saját számítógépünket jelöli.
- thrive-tribe-server: Ez a szerverünk neve, amely az API-t szolgáltatja.
- public: Ez általában az a könyvtár, ahol a publikus fájlok találhatóak az alkalmazásban.
- api: Ez az az elérési út, ahol az API-hoz tartozó végpontok találhatóak az alkalmazásban.
- users: Ez az az erőforrás, amelyre az API kérést fog kapni vagy válaszolni fog.
- login: Ez az az eljárás vagy művelet, amelyet az API végrehajt. Az általunk vizsgált helyzetben egy bejelentkezési folyamatról van szó, ahol a felhasználóknak felhasználónevet és jelszót kell megadniuk a belépéshez.



14. ábra a login API hívásunkról

- http protokollt használjuk.
- localhost: Ez a számítógép neve vagy IP címe, ahol a szerver fut. A localhost egy speciális hivatkozás, amely mindig a saját számítógépünket jelöli.
- thrivetribe-server: Ez a szerverünk neve, amely az API-t szolgáltatja.
- public: Ez általában az a könyvtár, ahol a publikus fájlok találhatóak az alkalmazásban.
- api: Ez az az elérési út, ahol az API-hoz tartozó végpontok találhatóak az alkalmazásban.
- maps: Ez az az erőforrás, amelyre az API kérést fog kapni vagy válaszolni fog.
- getMap: Ez az az eljárás vagy művelet, amelyet az API végrehajt.
- 7: Ez egy paraméter, amely azonosítja azt a konkrét térképet, amelyet le szeretnénk kérdezni. Ebben az esetben a 7-es azonosítójú térképet kérjük le.



15. ábra egy edzőterem adatainak a lekérdezéséről

## 4.5. Jelszó titkosítás és módosítás

A `'create'` függvény felelős az új felhasználók létrehozásáért az adatbázisban. Ennek a függvénynek a fontos része a jelszó biztonságos tárolása, amihez a Laravel keretrendszer `Hash::make()` metódusát használjuk.

A `Hash::make()` metódus segítségével a felhasználó által megadott jelszót hash-eljük, vagyis egy egyirányú titkosítási folyamaton keresztül átalakítjuk egy úgynevezett. hash értéké. Ez azért fontos, mert a hash érték nem fordítható vissza az eredeti jelszóra, így még akkor sem lehet látni a felhasználó által megadott jelszót, ha valaki hozzáfér az adatbázishoz.

A `'PasswordChange'` függvény felelős a felhasználók jelszavainak módosításáért. A `$user` változóban eltároljuk az adott felhasználó objektumát. A funkció az átadott `$request` objektum alapján megváltoztatja a felhasználó jelszavát. Ellenőrzi, hogy a megadott régi jelszó megfelel-e a felhasználó jelenlegi jelszavának. Ellenőrzi, hogy az új jelszavak megegyeznek-e. Ha minden ellenőrzés sikeres volt, a funkció megváltoztatja a felhasználó jelszavát az új jelszóra.

## 4.6. Regex

Az informatikában a reguláris kifejezés egy szövegek mintázatait leíró kód, amit szövegekben való keresésre, illetve annak formai ellenőrzésére, módosítására használnak.

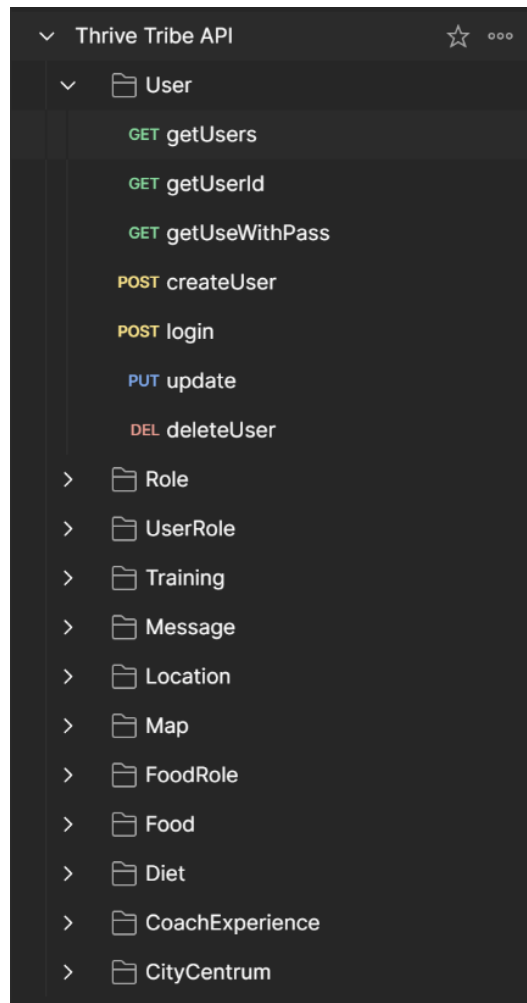
A reguláris kifejezéseket gyakran használják programozás során például arra, hogy a felhasználó által beírt adatokról eldöntsék, megfelelő formátumú-e, azaz például egy szabályos e-mail címet vagy telefonszámot adott-e meg egy űrlapon. Reguláris kifejezésekkel egyetlen sorban megoldhatóak olyan feladatok, amiket nélkülük csak egy hosszabb program segítségével lehetne elvégezni.

A mi esetünkben a telefonszámnál használtuk, melyet mi saját magunk írtunk meg. Ez esetben az oldalon a helyes telefonszám formátum +36-al kezdődik, és utána még kilenc darab szám van.

## 4.7. Tesztdokumentáció

A fejlesztés során rendszeres tesztelés történt, mely párhuzamosan zajlott a fejlesztési folyamattal. Miután elkészült a program, többször is kipróbáltuk annak működését különböző helyzetekben, hogy biztosak lehessünk a stabilitásában és funkcionalitásában. A teszteket saját laptopjainkon végeztük el, biztosítva ezzel a környezet kontrollált és stabil voltát.

A tesztek során az API-khoz történő kérések ellenőrzésére és validálására a Postman alkalmazást használtuk. Ennek segítségével gördülékenyen futtattuk és ellenőriztük az összes API hívást, biztosítva ezzel azok helyes működését. A docs mappában megtalálhatóak a Postman tesztek alatt használt elérési útvonalak is. Ezt a fájlt a Thrive Tribe API-postman\_collection.json néven találhatók.



16. ábra a Postman alkalmazásból

## 5. A weboldal design-ja

Már a weboldal ötletének kitalálásakor tudtuk milyen kinézetet akarunk neki készíteni. Legyen világos témájú, egyszerű, letisztult és könnyen kezelhető. A szürke alapszín is ekkor lett egyértelmű. A kinézetet folyamatosan készítettük a kódolással párhuzamosan, mivel gyakran ez is befolyásolta azt, hogy mi és hogyan fog működni az egyes füleken. A felhasznált ikonokat, képeket az internetről letöltve, majd Figmában átszerkesztve, átszínezve használtuk fel.

A logónkat a Microsoft BingAI-ának segítségével generáltattuk, ami egy zöldes szívet ábrázol, ezzel is az egészség fontosságát sugallva. A weboldal a nevét a témájáról kapta, egy közösség, ami fejlődni szeretne. Ezen felül az angol elnevezések egyre elterjedtebbek, így úgy gondoltuk, hogy ez lesz a legmegfelelőbb az oldalunk számára.



*17. ábra a Thrive Tribe logójáról*

## 6. Összefoglalás

### 6.1. A szakdolgozat célja

A szakdolgozatunk célja egy olyan weboldal volt, amellyel a felhasználók egészségesebb életmódjukra való törekvéseiket egy oldalon megtalálhatják.

Egy olyan online platform létrehozása, amely segítséget nyújt az életmódváltásra törekedőknek. Ennek a digitális menedékhelynek az alapvető célja a releváns információk és eszközök biztosítása az egészséges étkezés és életmód terén, valamint aktív támogatás nyújtása az embereknek ezen az úton.

Így időt spórolva más oldalak keresését illetően.

### 6.2. Megvalósítás

Első sorban a weboldal dizájnolását kezdtük el megtervezni, amihez a Figma programot használtuk. A kivitelezés közben kirajzolódott bennünk az oldal letisztultsága és egyszerű kezelhetősége. A kezdeti kódolási fázisokban a frontend megírása volt a cél, mert láttuk, hogy több időt fog igénybe venni, mint a backend-é. Így a főoldalnak álltunk neki, melyet a Regisztrációs/Belépési rész dizájnja követett. Ez után következett a backend kódolása és ezt követően a két folyamat párhuzamosan haladt előre a projekt készülésekor.

Ezt követően az adminisztrációs felület tervezésével már konkrét elképzeléseink voltak arról, hogy milyen funkciókat kívánunk biztosítani az adminisztrátor számára. Az implementálásnál adtunk neki lehetőséget módosításra, törlésre és létrehozásra. Utolsó előtti lépésekben a backendnek, illetve a frontendnek az összeköttetését valósítottuk meg, a különböző navigációs fűlekhez kivéve az regisztráció/belépés kivételével elkészült az összes kapcsolat. Végezetül egy átfogó ellenőrzést hajtottunk végre az oldalon, hogy megbizonyosodjunk a dizájn és felhasználói interakciók működőképességéről.



### 6.3. Fejlesztési lehetőségek

A tervezett fejlesztéseink egy olyan felhasználói felületet céloznak meg, amely még inkább személyre szabott és interaktívva teszi az edzéseket és az edzőkkel való kommunikációt.

Első lépésként minden felhasználó számára lehetővé lehet tenni a profilkép beállítását, hogy a közösség még személyesebb legyen. Ezáltal mindenki könnyebben azonosítható lesz, ami segíthet az edzőknek és a felhasználóknak abban, hogy közelebb kerüljenek egymáshoz.

Az üzenőfal egy másik kiemelt funkció lehet, ahol a felhasználók és az edzők szabadon kommunikálhatnak. Ez lehetőséget ad a felhasználóknak arra, hogy kérdéseket tegyenek fel az edzőknek, vagy épp megosszák egymással a tapasztalataikat és a motivációjukat. Az edzők pedig használhatják az üzenőfalat arra, hogy közvetlenül kapcsolatba lépjenek a felhasználókkal, tanácsokat adjanak nekik, vagy akár témákat boncolgassanak, amelyek érdekesek lehetnek az egész közösség számára.

A fejlesztések közé a kedvenc edzéseket egy külön ablakban is be lehet építeni, ami megkönnyíti a felhasználók számára az edzéstervek tárolását és hozzáférését. Amikor valaki megtalálja a kedvenc edzését, egyszerűen elmentheti azt a profilon belül, és könnyen visszatérhet hozzá később, anélkül, hogy végig kellene keresnie az összes elérhető edzéstervet. Ezáltal időt és energiát takaríthatnak meg, és könnyebben fókuszálhatnak az edzésekre.

Végül az automatizálás. Az automatizálásnak köszönhetően az Étrend fülön belül könnyedén létre lehet hozni új tabokat, így egyszerűbben és gyorsabban tud új diétákat és ételeket összeállítani az admin.

Ezek a tervezett fejlesztések mind azt célozzák meg, hogy még inkább összekössék és támogassák egymást az edzést végzők és az edzők között, valamint segítsék a felhasználókat abban, hogy hatékonyabban és élvezetesebben ériék el a céljaikat az edzések során.

## 7. Irodalomjegyzék

Fénykép források:

- [1] <https://unsplash.com/>
- [2] <https://www.freepik.com/>
- [3] <https://www.bing.com/images/create>

Tervezéshez és szerkesztéshez:

- [4] <https://www.figma.com/>

Programozási problémák megoldásai:

- [5] <https://stackoverflow.com>
- [6] <https://laravel.com/docs/11.x>
- [7] <https://learning.postman.com/>
- [8] <https://www.tiny.cloud/docs/tinymce/latest/react-cloud/>
- [9] <https://ihateregex.io/>
- [10] <https://codesandbox.io/>
- [11] <https://www.npmjs.com/>
- [12] <https://react-bootstrap.netlify.app/docs/getting-started/introduction>

Ikonok és matricák:

- [13] <https://react-icons.github.io/react-icons/https://boxicons.com/>

Az oldalon lévő információk:

- [14] <https://www.hsph.harvard.edu/nutritionsource>
- [15] <https://www.ncbi.nlm.nih.gov/pmc/articles>

Felhasznált szakirodalom:

- [16] <https://codegym.cc/hu/quests/lectures/hu.questservlets.level14.lecture02>
- [17] <https://kiszervezettmarketing.hu/weboldal-keszites/api-jelentese/>
- [18] <https://lexiq.hu/regularis-kifejezes>

## 8. Mellékletek

### 8.1. Programkódok

#### 8.1.1. Jelszó titkosítás

```
public function create(StoreUsers $request)
{
    $request->validated();
    $requestArray = $request->all();
    $roles = $request->get("roles");
    unset($requestArray["roles"]);
    $requestArray["Password"] = Hash::make($request->newPassword);
    $created = Users::create($requestArray);
    $created->roles()->attach($roles);
    return response()->json($created);
}
```

### 8.1.2. Jelszó módosítás

```
public function PasswordChange(Users $user, Request $request)
{
    $log = $request->all();
    $newPassword1 = $log["newPassword1"];
    $newPassword2 = $log["newPassword2"];

    //$random = Hash::make($newPassword1);
    //$user->Password = $random;

    if(Hash::check($log["Password"], $user->Password)){
        if($newPassword1 == $newPassword2){
            $pass = Hash::make($newPassword1);
            $user->Password = $pass;
            $user->save();
            return response()->json("OK", 204);
        } else {
            return response()->json("A két jelszó NEM egyezik!", 401);
        }
    } else {
        return response()->json("Nem az a régi jelszód!", 400);
    }
}
```

### 8.1.3. Access\_Component

```
import React from 'react';
import DietTabs from "@components/Diet/DietTabs";

function isValidView(viewname: string, userType: number):boolean {
    //viewname - az oldal neve, mely megnyitásra került - a név azonos kell, hogy legyen az
    //alábbi tömbökben szereplőkkel
    //userType - átadásra került a belogolt felhasználó azonosítója
    interface Nezetek {
        name: string;
    }
    const adminNezetek : Nezetek[] = [
        { name: "admin"},
        { name: "citycenrumCRUD"},
        { name: "coachCRUD"},
        { name: "dietCRUD"},
        { name: "foodCRUD"},
        { name: "locationCRUD"},
        { name: "mapCRUD"},
        { name: "roleCRUD"},
        { name: "trainingCRUD"},
        { name: "userCRUD"}
    ];
    const edzoNezetek : Nezetek[] = [
        { name: "coachcreate"},
        { name: "trainingcreate"},
        { name: "diet"},
        { name: "map"},
        { name: "datamodification"},
    ]
}
```

```

    { name: "passwordchange"}
  ];
const felhasznaloNezetek : Nezetek[] = [
  { name: "coach"},
  { name: "diet"},
  { name: "map"},
  { name: "training"},
  { name: "detailedtraining"},
  { name: "datamodification"},
  { name: "passwordchange"}
];

if(userType == 1){
  const talalatAdmin = adminNezetek.find(Nezetek => Nezetek.name == viewname);
  if(talalatAdmin != undefined){
    return true;
  }else{
    //nem admin jog
    return false;
  }
}else if (userType == 5){
  const talalatEdzo = edzoNezetek.find(Nezetek => Nezetek.name == viewname);
  if(talalatEdzo != undefined){
    return true;
  }else{
    //nem edzo jog
    return false;
  }
}else if (userType == 4){

```

```

const talalatFelhasz = felhasznaloNezetek.find(Nezetek => Nezetek.name ===
viewname);
    if(talalatFelhasz !== undefined){
        return true;
    }else{
        //nem felhasz jog
        return false;
    }
} else {
    //ha egyik user kategóriában sincs, tehát nincs belépve, akkor nem lát semmit a
vizsgált oldalakon
    return false;
}
}

export default isValidView;

```

#### 8.1.4. Regex

```

...
<form onSubmit={ (event) => {
    event.preventDefault();
    const form = event.target;
    const regex = /^[+]?([36]{2})?[0-9]{3}?[0-9]{6,6}$/;
    let match = regex.test(form.phone.value);
    if(match) {} else {alert("Nem megfelelő a telefonszám!");}
    if(ellenorzott || match) {
        axios.post("http://localhost/thrivetribe-server/public/api/users/create",
        {
            ...

```

```
        Phone: form.phone.value,  
        ...  
    }).then((res) => {  
        alert("Sikeres felvitel!");  
    }).catch((res) => {  
        alert("Már van ilyen felhasználónév!");  
    })  
    } else {}  
    }}>  
...  
<label>Telefonszám/Mobilszám</label>  
    <div className="custom-inputRegistration">  
        <input type="phone" name="phone"/>  
    </div>  
...  
<Button className="registration" type="submit">Mentés</Button>  
</form>  
...
```