

Vizsgaremek 2024/2025	3
1. Dizájn	4
2. Fejlesztői dokumentáció	5
2.1 Backend	6
2.1.1 PostgreSQL adatbázis és használata Dockerrel	8
2.1.2 Adatbázis Struktúra +ER diagram	10
2.1.3 Projektstruktúra [BACKEND]	14
2.1.3.1 Scripts	16
2.1.3.2 Common	17
2.1.3.3 Docs	18
2.1.4 Swagger dokumentáció	21
2.1.5 Authentikáció és Authorizáció	23
2.2 Frontend	26
2.2.1 Projektstruktúra [FRONTEND]	28
2.2.2 Indítás Mobilról	30
2.2.3 Seedelés által generált teszt fiókok	31
3. Felhasználói dokumentáció	32
3.1 Regisztráció/Bejelentkezés	33
3.1.1 Regisztráció	34
3.1.2 Bejelentkezés	36
3.2 Navigációs sáv	38
3.2.1 gombok	40
3.3 Főoldal/dashboard	43
3.3.1 Napi edzés	44
3.3.2 Heti edzés	46
3.3.3 Naptár	47
3.3.4 Rekordok	49
3.3.5 Statisztika	50
3.4 Edzések	51
3.4.1 Edzések Oldal	52
3.4.2 Egy edzés részletei Oldal	54
3.4.3 Aktív edzés	56
3.5 Edzéstervezek	60
3.5.1 Edzéstervezek Oldal	61
3.5.2 Egy edzésterv oldal	64
3.6 Gyakorlatok	66
3.6.1 Gyakorlatok Oldal	67
3.6.2 Egy gyakorlat Oldal	70
3.7 Profil	73
3.7.1 Profil Oldal	74
3.7.2 Kedvenc edzések	75
3.8 Admin	76
3.8.1 Gyakorlat Szerkesztése Oldal	77
3.8.2 Új gyakorlat hozzáadása	78
3.8.3 Admin felület	79

3.9 Statisztikák	81
3.9.1 Statisztikák oldal	82
3.9.2 Rekordok Oldal	85
4. Teszt Dokumentáció	86
4.1 Edzés Teszt esetek	87
4.2 Izomcsoporthoz való hozzáférés tesztje	89
4.3 Felhasználó Teszt esetek - UserService és UsersController	91
4.4 UserGyakorlatController és UserGyakorlatService Teszt Dokumentáció	95
4.5 GyakorlatController és GyakorlatService Teszt Dokumentáció	99
4.6 Frontend manuális tesztelés	102
4.7 e2e teszt	113

Vizsgaremek 2024/2025

Bevezetés

A **RepVault** projekt célja, hogy egy egyszerűen használható, ingyenes edzéskövető applikációt hozzunk létre, amely lehetővé teszi a felhasználók számára edzéseik hetente történő naplózását és nyomon követését.

A projekt dokumentációja egy központi dokumentumtárból, a **Confluence** platform segítségével készült, amely támogatja az átlátható és rendszerezett fejlesztési folyamatot.

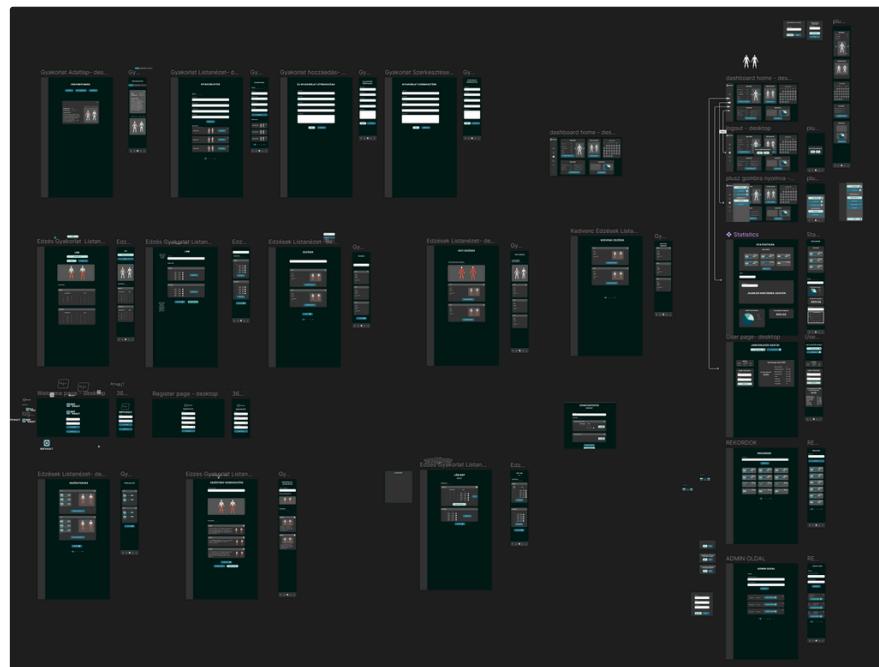
1. Dizájn

A projekt tervezése és prototípusának elkészítése a **Figma** segítségével zajlott, mely lehetővé tette a gyors és rugalmas megoldások kidolgozását. A képen látható, hogy a végső layout jelentősen kibővült, a különféle képernyőméretekhez és funkciókhoz igazodva. A tervezési folyamat során folyamatosan bővítettük a felületeket, így az oldalak és komponensek száma fokozatosan nőtt, és mára egy komplex, átlátható struktúrát alkotnak.

A Figma használatának előnyei közé tartozik:

- **Együttműködés:** valós időben többen is szerkeszthették a felületet.
- **Verziókövetés:** bármikor visszaléphetünk korábbi mentésekhez.
- **Prototípus készítés:** interaktív kattintható prototípusokat hozhattunk létre, melyek segítségével könnyebb volt a folyamat végén a funkciók tesztelése és bemutatása.

Ennek eredményeként a kialakult tervek jól átláthatók, könnyen módosíthatók és stabil alapot nyújtanak az alkalmazás fejlesztéséhez. A különféle képernyők az alkalmazás teljes funkcionalitását átfogják, és a felhasználók visszajelzései alapján is folyamatosan finomíthatók.





2. Fejlesztői dokumentáció

Ez a dokumentáció a *RepVault™* webalkalmazás fejlesztői szempontból releváns működését mutatja be, és részletes útmutatást nyújt a rendszer helyi környezetben történő elindításához, valamint a fejlesztési környezet konfigurálásához.



2.1 Backend

Backend setupolása [🔗](#)

- klónozzuk le a repository-t githubról.

```
1 git clone https://github.com/Almos-Pal/Vizsgaremek-Backend/
```

- nyissuk meg a mappát

```
1 cd .\Vizsgaremek-Backend\
```

- töltük le a dependenciákat

```
1 npm install
```

a **.env.example-ból** másoljuk át az adatokat egy **.env** fájlba majd cseréljük ki a DATABASE_URL-t az alább megadott verzióra:

```
1 DATABASE_URL="postgresql://user:repvault_is_awesome@localhost:5432/repvault"
```

i Érdemes hosszabb titkosított kulcsokat beadni a `JWT_SECRET` és `JWT_REFRESH_TOKEN_KEY` értékek helyére, mint amit a `.env.example` tartalmaz.

Ezután az adatbázis elérése érdekében indítsuk el a Docker Desktop programot, majd a korábban használt terminál ablakban az alábbi kód segítségével indítsuk el a konténerünket.

```
1 docker-compose up -d
```

i A projekt adatbázisáról és a docker használatáról részletesebben a külön arról szóló részben tudhat meg többet.

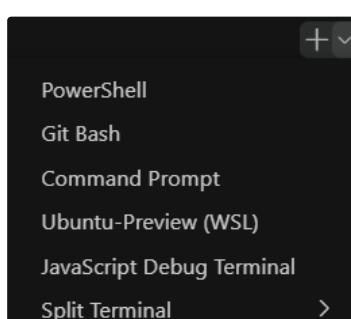
- Ahhoz, hogy az adatbázis elkészüljön, futtassuk az alábbi parancsot, ami szinkronizálja a sémát.

```
1 npx prisma db push
```

A seedeléshez létre hoztunk egy bash fájlt, amivel egyszerűen le lehet futtatni a projekthez kellő adatokat,-Ezek működéséről egy későbbi fejezetben lesz szó részletesebben

Visual studio code-ot használva futtathatunk bash parancsokat

1. Nyissuk meg a terminált
2. Kattintsunk a képen látható plusz gomb melletti lefelé ikonra



3. Válasszuk ki a "Git Bash" lehetőséget

4. lépjünk bele a scripts könyvtárba `cd Vizsgaremek-Backend/scripts/`

5. futtassuk a seedeléseket. `bash run-seeds.sh` (így legenerálunk pár teszt adatot és az oldal működéséhez kellő kötelező táblákat)

ez után futtassuk a szervert

```
1 npm run start:dev
```



2.1.1 PostgreSQL adatbázis és használata Dockerrel

Ez a projekt PostgreSQL adatbázist használ, amely egy Docker-konténerben fut. Ennek a megoldásnak a célja, hogy egyszerű, gyors és platformfüggetlen módot biztosítson a fejlesztéshez szükséges adatbázis környezet elindítására.

Előfeltételek

Először győződjön meg róla, hogy az alábbi szoftverek telepítve vannak a számítógépére:

- Docker Desktop
- Docker Compose (Docker Desktop részeként automatikusan elérhető)

Az adatbázis elindítása

1. Klónozza a projekt backend repository-ját

```
1 git clone https://github.com/Almos-Pal/Vizsgaremek-Backend.git
2 cd Vizsgaremek-Backend
```

2. Indítsa el az adatbázist (és a pgAdmin felületet) a következő parancssal:

```
1 docker-compose up -d
```

Ez a parancs:

- Elindít egy PostgreSQL adatbázist (`db` nevű szolgáltatásként)
- Elindítja a pgAdmin felületet (`pgadmin` nevű szolgáltatásként)
- Létrehoz egy `pgdata` nevű Docker volument, amely az adatokat tartósan tárolja

Az adatbázi elérhetőségei

Paraméter	Érték
Host	localhost
Port	5432
Adatbázis	repvault
Felhasználónév	user
Jelszó	repvault_is-awesome

Ezek az értékek a `docker-compose.yml` fájlban is megtalálhatóak, a `db` szolgáltatásnál.

Kapcsolódás pgAdmin-on keresztül (opcionális)

Amennyiben szükségesnek találja, az adatbázist a böngészőn keresztül, vizuálisan is lehet kezelni, a pgAdmin felület használatával

- Nyissa meg a következő URL-t: <http://localhost:5050>

- Jelentkezzen be:
 - **Email:** admin@example.com
 - **Jelszó:** admin
- Object Explorer oldalon a “Servers”-re jobb egér kattintással megnyíló felületen, kattintson a Register > Server menü pontokra. Ezután
 - **Név:** például repvault
 - **Host name/address:** db
 - **Port:** 5432
 - **Maintenance database:** repvault
 - **Username:** user
 - **Password:** repvault_is_awesome
 - Pipálja be a “Save password” opciót
 - Ezután nyomjon a Mentés/Save gombra

Az adatbázis leállítása

A konténerek leállításához használja az alábbi parancsot:

```
1 docker-compose down
```

Ez a parancs leállítja a konténereket, azonban az adatokat megtartja a pgdata volume-on belül.

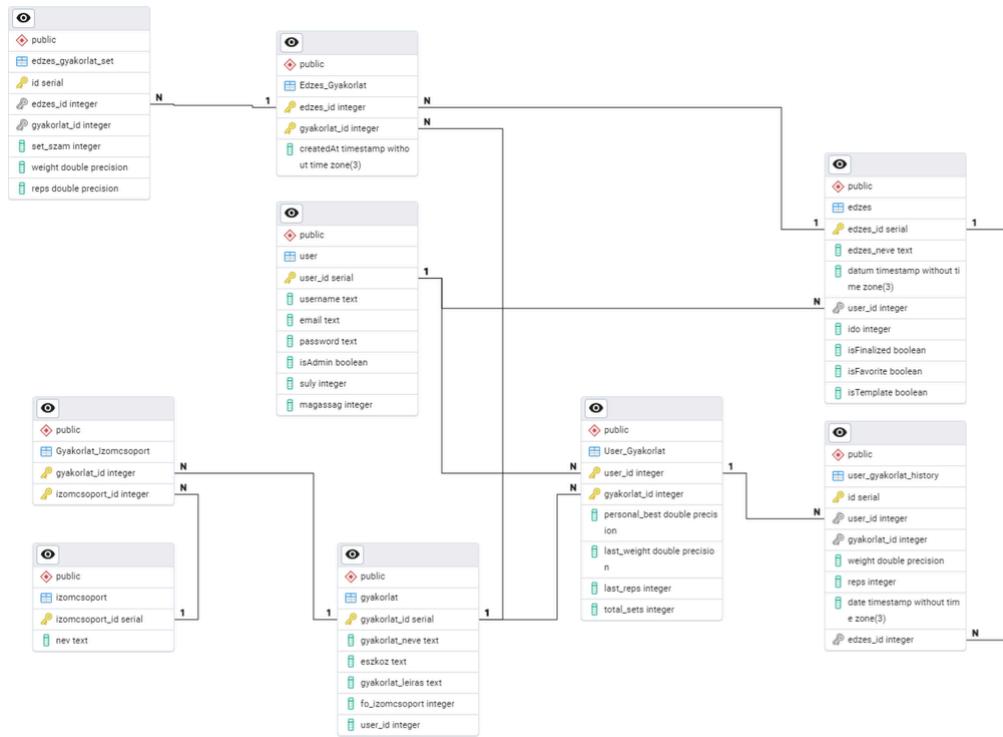
Adatbázis alaphelyzetbe hozása

Amennyiben teljesen új, tiszta adatbázisra van szüksége:

```
1 docker-compose down -v
```

Ez a parancs **leállítja és törli a konténereket**, valamint **eltávolítja az adatokat** is a volume-ból.

2.1.2 Adatbázis Struktúra +ER diagram



ER diagram

User tábla

Oszlop neve	Tárolt adat típusa	Egyéb
user_id	Egész szám	automatikusan generálódik
username	Szöveg	
email	Szöveg	egyedi
password	Szöveg	
gyakorlatok	Egy a többhöz kapcsolat a user_Gyakorlat táblával	
edzesek	Egy a többhöz kapcsolat az Edzes táblával	
isAdmin	Logikai	alapértelmezetten (false)
suly	Egész szám	nem kötelező
magasság	Egész szám	nem kötelező

Gyakorlat tábla ↗

Oszlop neve	Tárolt adat típus	Egyéb
gyakorlat_id	Egész szám	automatikusan generálódik
gyakorlat_neve	Szöveg	
eszköz	Szöveg	nem kötelező
gyakorlat_leiras	Szöveg	nem kötelező
fo_izomcsoport	Egész szám	nem kötelező
users	Egy a többhöz kapcsolat a User_Gyakorlat táblával	
edzesek	Egy a többhöz kapcsolat az Edzes_Gyakorlat táblával	
izomcsoportok	Egy a többhöz kapcsolat a Gyakorlat_Izomcsoport táblával	

User_Gyakorlat tábla ↗

Oszlop neve	Tárolt adat típus	Egyéb
user_id	Egész szám	
gyakorlat_id	Egész szám	
personal_best	Lebegőpontos szám	
last_weight	Lebegőpontos szám	
last_reps	Egész szám	
total_sets	Egész szám	
history	Egy a többhöz kapcsolat a User_Gyakorlat_History táblával	
user	Több az egyhez kapcsolat a User táblával	
gyakorlat	Több az egyhez kapcsolat a Gyakorlat táblával	

User_Gyakorlat_History tábla ↗

Oszlop neve	Tárolt adat típus	Egyéb
id	Egész szám	automatikusan generálódik
user_id	Egész szám	

gyakorlat_id	Egész szám	
weight	Lebegőpontos szám	
reps	Egész szám	
date	DateTime	Alapértelmezett (most)
edzes_id	Egész szám	
user_gyakorlat	Több az egyhez kapcsolat a user_gyakorlat táblával	
edzes	Több az egyhez kapcsolat az edzes táblával	

Edzés tábla ↗

Oszlop neve	Tárolt adat típus	Egyéb
edzes_id	Egész szám	automatikusan generálódik
edzes_neve	Szöveg	
datum	DateTime	Alapértelmezett (most)
isTemplate	Logikai	Alapértelmezett (false)
user_id	Egész szám	
ido	Egész szám	alapértelmezett (0)
isFinalized	logikai	alapértelmezett (false)
isFavorite	logikai	alapértelmezett (false)
user	Egy a töbhöz kapcsolat a User táblával	
gyakorlatok	Több a töbhöz kapcsolat az Edzes_gyakorlat táblával	
history	Több az egyhez kapcsolat a User_Gyakorlat táblához	

Edzes_Gyakorlat tábla ↗

Oszlop neve	Tárolt adat típus	Egyéb
edzes_id	Egész szám	
gyakorlat_id	Egész szám	
createdAt	DateTime	alapértelmezett (most)
edzes	Kapcsolat az edzes táblával	
gyakorlat	Kapcsolat a Gyakorlat táblával	

szettek	Egy a többhöz kapcsolat az Edzes_Gyakorlat_Set táblával	
---------	---	--

Edzes_Gyakorlat_Set tábla ↗

Oszlop neve	Tárolt adat típus	Egyéb
id	Egész szám	automatikusan generálódik
edzes_id	Egész szám	
gyakorlat_id	Egész szám	
set_szam	Egész szám	
weight	Lebegőpontos szám	
reps	Lebegőpontos szám	
edzes_gyakorlat	Több az egyhez kapcsolat az Edzes_Gyakorlat táblával	

Izompocsort tábla ↗

Oszlop neve	Tárolt adat típus	Egyéb
izomcsoport_id	Egész szám	automatikusan generálódik
nev	Szöveg	
gyakorlatok	Egy a többhöz kapcsolat a Gyakorlat_Izomcsoport táblával	

Gyakorlat_Izomcsoport Tábla ↗

Oszlop neve	Tárolt adat típus	Egyéb
gyakorlat_id	Egész szám	
izomcsoport_id	Egész szám	
gyakorlat	Több az egyhez kapcsolat a Gyakorlat táblával	
izomcsoport	Több az egyhez kapcsolat az Izomcsoport táblával	

2.1.3 Projektstruktúra [BACKEND]

Ez a dokumentáció bemutatja a backend projekt struktúráját és az egyes mappák funkciót, ami ezáltal segít a kódbázison belüli eligazodásban és annak megértésében.

-  minden resourcenak és annak végpontjainak swaggerben található a dokumentálása, ezért ebben a részben azok nem lesznek részletezve.

Projektstruktúra

```
1 Vizsgaremek-Backend/
2   |-- dist/
3   |-- flashpost-tests/
4   |-- node_modules/
5   |-- prisma/
6     |-- migrations/
7     |-- seeds/
8     |-- schema.prisma
9   |-- public/
10  |-- scripts/
11  |-- src/
12    |-- auth/
13    |-- common/
14    |-- docs/
15    |-- edzes/
16    |-- gyakorlat/
17    |-- izomcsoport/
18    |-- user-gyakorlat/
19    |-- users/
20    |-- app.controller.spec.ts
21    |-- app.controller.ts
22    |-- app.module.ts
23    |-- app.service.ts
24    |-- main.ts
25    |-- prisma.service.ts
26  |-- test/
27  |-- views/
```

Mappák és Fájlok

prisma/

- `migrations/` : Az adatbázis migrációs fájljai.
- `seeds/` : Az adatbázis feltöltésére szolgáló adatok.
- `schema.prisma` : A Prisma ORM adatmodellje.

src/

A fő backend kódbázis mappája.

Resource-ok (Végpontokhoz tartozó modulok)

- `auth/` : Hitelesítési modul (bejelentkezés, regisztráció, tokenek kezelése).
- `edzes/` : Edzésekkel kapcsolatos végpontok.
- `gyakorlat/` : Gyakorlatok kezelése.
- `izomcsoport/` : Izomcsoportokhoz kapcsolódó végpontok.
- `user-gyakorlat/` : Felhasználókhoz köthető gyakorlatok kezelése.
- `users/` : Felhasználói adatok kezelése.

Egyéb mappák

- `common/` : Közös segédfüggvények és middleware-ek.
- `docs/` : Dokumentációval kapcsolatos fájlok (pl. Swagger beállítások).
- `scripts/` : Különböző segédscript-ek (pl. adatfeltöltés).

Fő TypeScript Fájlok

- `app.controller.ts` : Az alkalmazás fő vezérlője.
- `app.module.ts` : Az alkalmazás fő modulja.
- `app.service.ts` : Az alkalmazás fő szolgáltatásai.
- `main.ts` : Az alkalmazás belépési pontja.
- `prisma.service.ts` : Prisma ORM szolgáltatás kezelése.

test/

Tesztelési fájlok a backend validálásához.

e2e teszt esetek, végig megy a projekt különböző részein és leteszeli annak alapvető mechanikáját

views/

Esetleges statikus fájlok vagy renderelt nézetek (ha szükséges). EBBEN A PROJEKTBEN NINCSEN



2.1.3.1 Scripts

A projekt kettő scripttel rendelkezik

- `run-seeds.sh` és `uploadApiDocument.ts`

A `run-seeds.sh` seedelésért felel:

- Izomcsoportok egy statikus tábla, ami a gyakorlatok bekategorizálását segíti
- gyakorlatok seed 873 különböző fajta gyakorlattal tölti fel az adatbázisunkat
- users seed néhány teszt felhasználót generál le admin és user jogosultságokkal
- edzesek seed legenerál minden felhasználó számára néhány edzést minden hétre bizonyos időközönként és biztosítja, hogy az applikáció jól tudja mérni az esetleges fejlődést vagy hanyatlást

A `uploadApiDocument.ts` swagger dokumentáció fájlba való kiszervezéséért felelős.

- első soron ez a postmanbe való beimportálás miatt volt fontos, hogy tudjuk a végpontokat tesztelni.

`uploadApiDocument.ts` futtatása:

```
cd ..\Vizsgaremek-Backend\
```

```
.\dist\scripts\uploadApiDocument.js
```

ez a kód létre hoz a docs könyvtárba egy `api.yaml` nevű fájl-t

👉 2.1.3.2 Common

- helpers
 - `pagination.helper.ts` segít az oldalazásban
 - `bmi.ts` bmi kategorizálása
- dto
 - különböző dto-ok az újrahasznosítás érdekében, mint pl
 - error
 - not-found
 - pagination
 - success



2.1.3.3 Docs

Bevezetés [🔗](#)

A backend végpontok swaggerrel, vannak dokumentálva, ez az oldal segít, a legfrissebb dokumentáció lekérésében, és megtekintésében.

Lépések [🔗](#)

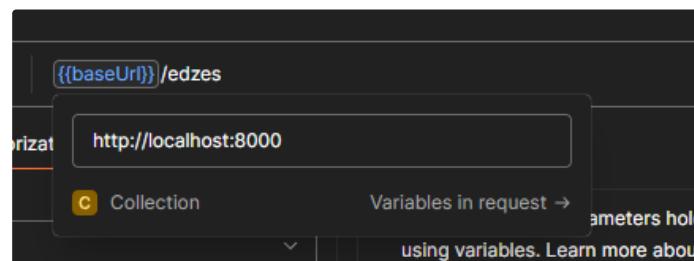
Elsőnek ezt a kódot futtassuk le.

```
1 node dist/scripts/uploadApiDocument.js
```

Ekkor, a docs mappánkban meg fog jelenni egy api.yaml fájl, amit, ha beimportálunk a postman-be, akkor le is tudjuk tesztelni, hogy hogy működnek a végpontjaink, és mikor milyen hibaüzenetek dobnak vissza, fontos hogy a postmant setupolni kell. 2 dolgot kell megtennünk mielőtt használni tudnánk, a végpontokat

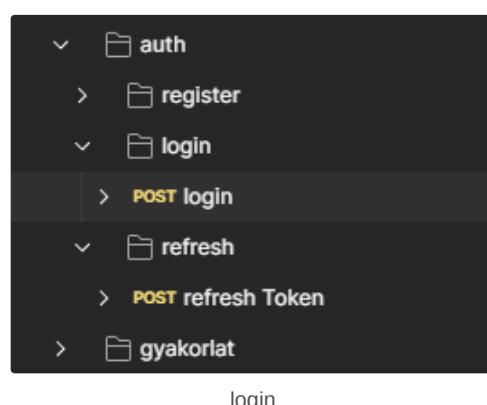
1. baseURL-nek be kell állítani

```
1 localhost:8000
```



Itt lehet átírni, ha az egerünket rávisszük a baseURL változóra.

2. A loginnál, bejelentkezünk egy fiókkal, például az alap generált adminnal, és válaszban kapunk egy access tokenet



login

A body-ba az alábbi módon kell megadni az adatokat. (ebben az esetben egy admin fiókkal jelentkezünk be, hogy minden végpont elérhetővé váljon.

The screenshot shows the Postman interface with a successful API call. At the top, the 'Body' tab is selected, showing the raw JSON sent in the request:

```

1  {
2   "email": "admin@gmail.com",
3   "password": "Admin123."
4 }

```

Below the request, the 'Body' tab is still active, showing the detailed JSON response received:

```

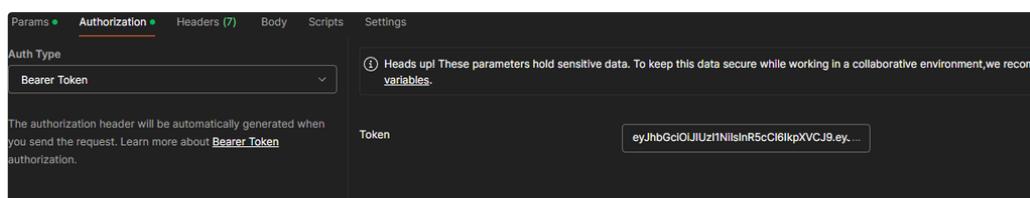
1  {
2   "user": {
3     "user_id": 17,
4     "username": "Admin",
5     "email": "admin@gmail.com",
6     "isAdmin": true,
7     "suly": 80,
8     "magassag": 180
9   },
10  "backendTokens": {
11    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImFkbWluQGdtYWlsLmNvbSIsImlzQWRtaW4iOnRydWUsInVzZ45WDn-1yemUvYRpmpXj7YcClfJPq6y-wsp3DiXPhGtE",
12    "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImFkbWluQGdtYWlsLmNvbSIsImlzQWRtaW4iOnRydWUsInVzZMtPJqZCVPcMwKfAYWMJu9dcIp40-IVyrg5YWH0hbdT0",
13    "expiresIn": 1744636587511
14  }
15 }

```

A response és a body

Ezután a login body-jában visszakapott accessToken-t kell használnunk, hogy a többi levédett végpontot is elérhessük.

A kapott accessTokenet az alábbi "Authorization" fülben tudjuk hasznosítani. Először állítsuk át az "Auth Type"-ot "Bearer Token"-re és a jobb oldalon látható mezőbe másoljuk be a korábban megszerzett accessTokenünket, fontos, hogy az idézőjelek a token elejéről és végéről ne kerüljenek bele.



authorizáció

i Fontos! Ezt mindenkor be kell állítani, ha olyan végpontot akarunk használni, amit még korábban nem. Illetve, ha az AccessTokenünk lejárt (1 nap a lejárási ideje).

Amennyiben nem szeretnénk, tesztelni a végpontok működését és csak, megszeretnénk tekinteni, a végpontokat, akkor megtudjuk nyitni a swaggett, mindenkor el kell indítanunk a backendet, az alábbi paranccsal.

```
1 npm run start:dev
```

És beírni a böngészőbe az alábbi url-t.

```
1 localhost:8000/api
```

i Swaggerről részletesebben az erre szánt részben tudhat meg többet.

Összegzés

A swaggerrel, meg lehet tekinteni, az összes végpontot, láthatjuk az elvárt kimeneteket, és hibaüzenetet.

A script lefuttatásával, pedig, a végpontok, egy export fájlba kerülnek, amit, ha valamilyen tesztelő alkalmazásba, mint például a postman, le tudunk futtatni, és láthatjuk a tényleges működést.

A script szépsége, hogy a tesztelő alkalmazást, nem kell manuálisan frissítgetni, mivel amikor új végpontokat hozunk létre, akkor csak le kell futtatni, majd újra importálni, és már láthatóak is a változtatások.

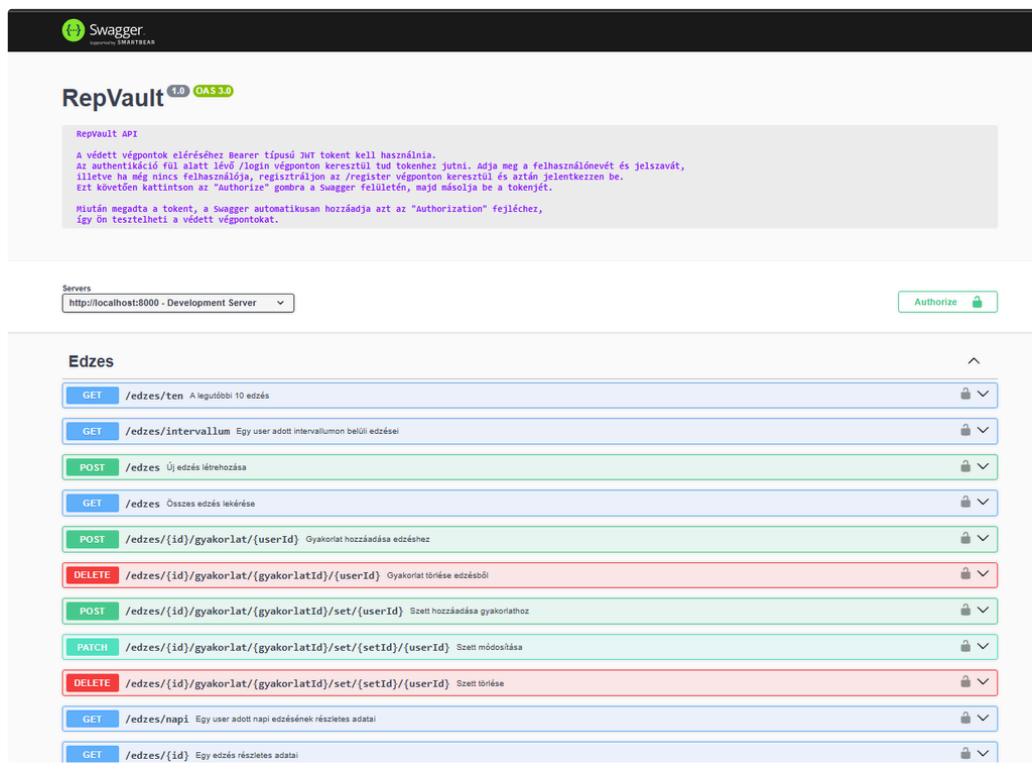
2.1.4 Swagger dokumentáció

Először el kell indítanunk a backendet az alábbi parancssal.

```
1 npm run start:dev
```

Ha ez megvan akkor, az alábbi url-en lesz elérhető a swagger dokumentáció.

```
1 localhost:8000/api
```



The screenshot shows the Swagger UI interface for the RepVault API. At the top, there's a navigation bar with the RepVault logo and the text "RepVault 1.0 OAS 3.0". Below the header, there's a detailed note about JWT token usage for authentication. The main content area is titled "Edzes" (Workouts) and lists several API endpoints:

- GET /edzes/ten: A legutóbbi 10 edzés
- GET /edzes/intervallum: Egy user adott intervallumon belüli edzései
- POST /edzes: Új edzés létrehozása
- GET /edzes: Összes edzés lekérése
- POST /edzes/{id}/gyakorlat/{userId}: Gyakorlat hozzáadása edzéshez
- DELETE /edzes/{id}/gyakorlat/{gyakorlatId}/{userId}: Gyakorlat törlése edzésből
- POST /edzes/{id}/gyakorlat/{gyakorlatId}/set/{userId}: Szett hozzáadása gyakorlathoz
- PATCH /edzes/{id}/gyakorlat/{gyakorlatId}/set/{setId}/{userId}: Szett módosítása
- DELETE /edzes/{id}/gyakorlat/{gyakorlatId}/set/{setId}/{userId}: Szett törlése
- GET /edzes/napi: Egy user adott napi edzésének részletes adatai
- GET /edzes/{id}: Egy edzés részletes adatai

On the right side of the interface, there's an "Authorize" button with a lock icon. The bottom right corner has the word "swagger".

Itt ha legörgetünk az authentikációhoz, azon belül is a bejelentkezéshez.

POST /auth/login Bejelentkezés

Felhasználi bejelentkezése

Parameters

No parameters

Request body **required**

application/json

Felhasználi bejelentkezési adatai

Example Value | Schema

```
{
  "email": "example@example.com",
  "password": "example123"
}
```

Responses

Code	Description	Links
200	Sikeres bejelentkezés	No links
401	Hibás felhasználónév vagy jelszó	No links

Bejelentkezés

Ezt láthatjuk és itt a try it out gombra majd a lentebb megjelenő executera kattintva kapunk egy, access tokenet, amit ha kimásolunk és az oldal tetején, az authorize gombnál bemásoljuk, akkor tudjuk magunkat validálni. Ezután használhatóak a többi végpontok, az előbb említett módon.



2.1.5 Authentikáció és Authorizáció

Használt technológiák és módszerek

Backend

- **JWT (JSON Web Token)** token-alapú authentikáció és authorizáció
- **bcrypt** jelszó titkosítás (hash-elés)

Frontend

- **NextAuth.js** authentikáció kezelésére
- **Formik** és **Yup** űrlapok validációjához

Authentikációs folyamat lépései

1. Regisztráció (POST /auth/register)

- A felhasználó elküldi regisztrációs adatait (felhasználónév, email, jelszó).
- A rendszer ellenőrzi, hogy az email cím egyedi-e.
- bcrypt segítségével a rendszer biztonságosan hash-eli a jelszót.
- Sikeres ellenőrzés és hash-elés után az új felhasználót az adatbázisba menti.
- Konfliktus esetén (email már használatban van), 409-es hibakódot dob.

2. Bejelentkezés (POST /auth/login)

- A felhasználó megadja a bejelentkezési adatait (email és jelszó).
- A rendszer bcrypt segítségével validálja a jelszót a tárolt hash-sel.
- Sikeres validáció esetén JWT-tokent generál:
 - **Access Token** (érvényesség: 1 nap)
 - **Refresh Token** (érvényesség: 7 nap)
- Sikertelen bejelentkezés esetén 401-es hibakódot ad vissza.

3. JWT-alapú hitelesítés (JwtGuard)

- minden védett útvonalhoz szükséges az Access Token érvényessége.
- A rendszer a HTTP fejlécből (Bearer token) kiolvassa az Access Tokent.
- Ellenőrzi a JWT token hitelességét és lejáratát.
- Ha a token érvénytelen vagy lejárt, a rendszer 401-es hibát dob.

4. Token frissítése (POST /auth/refresh)

- Ha az Access Token lejár, a felhasználó Refresh Token segítségével új Access Tokent kérhet.
- A Refresh Token a HTTP fejlécben `Refresh` típussal kerül továbbításra.
- A rendszer validálja a Refresh Tokent és új Access és Refresh tokent generál.
- Sikertelen ellenőrzés esetén Unauthorized (401) hibát dob.

További guardok és védelem

Admin Guard

- Ellenőrzi, hogy a felhasználó rendelkezik-e admin jogosultsággal.
- Csak adminok számára elérhető endpointokat véd, például jogosultságok kiosztása vagy visszavonása.
- Jogosultság hiányában 403-as (Forbidden) hibát ad vissza.

CreateEdzesOwnerGuard

- Ellenőrzi, hogy egy felhasználó csak saját részére hozhat-e létre edzést.
- Összehasonlítja a kérelemben szereplő felhasználói azonosítót a tokenből kiolvasott azonosítóval.
- Jogosulatlan kísérlet esetén 403-as (Forbidden) hibát dob.

EdzesOwnerGuard

- Biztosítja, hogy a felhasználók kizárolag a saját edzéseiket érhessék el.
- Admin felhasználók bármilyen edzést megtekinthetnek.
- Jogosulatlan hozzáférés esetén 403-as (Forbidden) hibát ad vissza.

UserIdMatchGuard

- Garantálja, hogy a felhasználó kizárolag saját adatait láthassa és módosíthassa.
- Admin jogosultságú felhasználók bármelyik felhasználó adatait kezelhetik.
- Jogosulatlan hozzáférési próbálkozás esetén 401-es (Unauthorized) hibát dob.

Frontend authentikációs folyamat lépései

1. Regisztráció

- Formik és Yup segítségével validált űrlap beküldése.
- Adatok elküldése a backend `/auth/register` végpontjára.
- Sikeres regisztráció esetén átirányítás a bejelentkezés oldalra.

2. Bejelentkezés

- Formik és Yup validált űrlap beküldése.
- NextAuth.js a backend `/auth/login` végpontjára küldi a bejelentkezési kérelmet.
- Sikeres bejelentkezés után JWT-tokenek tárolása és felhasználó átirányítása a dashboardra.
- Sikertelen bejelentkezés esetén hibaüzenet (Toast).

3. JWT-kezelés és token frissítés

- NextAuth.js kezeli a JWT-tokeneket a sessionben.
- Ha lejárt az Access Token, automatikusan frissíti azt a backend `/auth/refresh` endpoint segítségével.

Frontend Middleware

- Next.js middleware biztosítja, hogy csak authentikált felhasználók érhessenek el bizonyos oldalakat.
- Admin jogosultság ellenőrzése, és jogosulatlan hozzáférés esetén automatikus átirányítás történik.

Frontend komponensek

- **WelcomeLogin:** bejelentkezési űrlap.

- **Register:** regisztrációs űrlap.
- **ReactSessionProvider:** NextAuth.js session szolgáltató a teljes app számára.



2.2 Frontend

Frontend setupolása

- Klónozzuk le a repot githubról.
- 1 git clone https://github.com/Almos-Pal/Vizsgaremek-Frontend
- nyissuk meg a mappát

```
1 cd .\Vizsgaremek-Frontend\frontend\
```

- töltük le a dependenciákat

```
1 npm install
```

- Futtassuk a szervert

```
1 npm run dev
```

Szerver futtatását követően valami hasonlót fogunk kapni a terminálunkban:

```
▲ Next.js 15.1.6 (Turbopack)
- Local:          http://localhost:3000
- Network:        http://192.168.1.43:3000
- Environments:  .env.local
```

Az értékek nagyvalószínűséggel mások lesznek

Viszont ekkor még nem teljesen használatra kész a programunk. Még egy teendőnk van az pedig az, hogy a `.env.local`-ből hozzunk létre és másoljuk át az adatokat egy `.env.local` fájlba majd cseréljük ki az alábbi sorokat.

A NEXTAUTH_URL-t két módon is beállíthatjuk.

Az első megoldás, ha a localhost:3000-es portot használjuk, ez teljesen működő képes addig amíg csak azon az eszközön akarjuk futtatni a programot amelyiken fut a kódunk, ez így nézne ki:

```
1 NEXTAUTH_URL=http://localhost:3000
```

Viszont, ha szeretnénk telefonon is tesztelni a programunkat, a korábban futtatott npm run dev által kapott Network ip-t kell használnunk (ez az ip nagyvalószínűséggel más lesz mindenkinél). Ez így nézhet ki például:

```
1 NEXTAUTH_URL=http://192.168.1.43:3000
```

ⓘ Telefonos futtatásról az arról szóló részben tudhatunk meg többet.

A NEXT_PUBLIC_BACKEND_URL beállítása a következő:

Ez a sor kód a backend elérését biztosítja. Ahhoz, hogy ezt megtehessük ugyan azt a Network ip-t kell használnunk, mint a nextauth_url-nél csak most a 8000-es portot használva. Ez így nézhet ki például:

⚠ Fontos, hogy itt nem használhatjuk a localhost portot. Kötelező megadni a networkot

```
1 NEXT_PUBLIC_BACKEND_URL=http://192.168.1.43:8000
```

NEXTAUTH_SECRET beállítása:

Ez a sor kód az authentikációban segít, mivel a programunk csak lokálisan fut ezért mindegy, hogy itt mit adunk meg. Viszont az fontos, hogy legalább valami meg legyen adva.

```
1 NEXTAUTH_SECRET=32131241dsqwadfafdasfasfdasbfdasuisufdsa
```

A végső **.env.local** fájlnak valami hasonlóan kell kinéznie:

```
1 NEXTAUTH_URL=http://192.168.1.43:3000
2 NEXT_PUBLIC_BACKEND_URL=http://192.168.1.43:8000
3 NEXTAUTH_SECRET=32131241dsqwadfafdasfasfdasbfdasuisufdsa
```

Ha ezzel minden megvagyunk akkor készen áll a programunk a használatra. A programot az npm run dev után kapott **Local** vagy **Network** címen érhetjük el.

2.2.1 Projektstruktúra [FRONTEND]

Ez a dokumentáció bemutatja a projekt alapvető mappastruktúráját és a mappák funkcióit.

Mappastruktúra

```
1 src/
2   |-- app/
3   |-- components/
4   |   |-- client/
5   |   |-- server/
6   |-- hooks/
7   |-- lib/
8   |-- styles/
9   |-- types/
10  |-- utils/
11
```

Mappák részletes ismertetése

src/

A projekt fő forráskódja ebben a mappában található.

app/

A fő Next.js alkalmazás mappa, ahol az oldalak és az applikáció logikája helyezkedik el.

components/

Az alkalmazás komponenseit tartalmazza.

- `client/` : A kliensoldali komponensek, amelyek a böngészőben futnak.
- `server/` : A szerveroldali komponensek, amelyek a backend logikához kapcsolódnak.

hooks/

Egyedi React hook-ok gyűjteménye, amelyeket az alkalmazás különböző részein lehet használni.

lib/

Külső könyvtárak és segédfüggvények, amelyek az alkalmazás központi logikáját kezelik.

Ebbe tartozik még az api is

styles/

Globális és moduláris stílusfájlok (SCSS, CSS, Tailwind stb.).

types/

Típusdefiníciók TypeScript-hez, amelyeket a projektben használunk.

utils/

Hasznos segédfüggvények és helper metódusok, amelyek segítik az ismétlődő kódok csökkentését.

Ez a struktúra segít a projekt könnyebb karbantarthatóságában és átláthatóbb szervezésében.

2.2.2 Indítás Mobilról

A projektet, el lehet érni mobilról is, a könnyebb tesztelhetőség érdekében.

Ahhoz, hogy ezt meg tudjuk csinálni, szükségünk lesz az alábbiakra.

- A számítógépünknek (amiről futtatjuk a projektet) és a mobilunknak egy hálózaton kell lenniük.
- Ezután a projekt frontendjét el kell indítani a szokásos “npm run dev” parancsal, ezáltal kapjuk meg az alábbihoz hasonló táblázatot a terminálunkban

```
▲ Next.js 15.1.6 (Turbopack)
- Local:      http://localhost:3000
- Network:    http://192.168.1.43:3000
- Environments: .env.local
```

- Ezt követően az előbb megkapott táblázatból a network utánit kell beillesztenünk az env.local fájlunkba.
- Illetve fontos, hogy a NEXTAUTH_URL “:3000”-re végződjön, a NEXT_PUBLIC_BACKEND_URL pedig “8000”-re (mivel 8000-es porton fut a backendünk).

```
NEXTAUTH_URL=http://192.168.1.43:3000
NEXT_PUBLIC_BACKEND_URL=http://192.168.1.43:8000
```

 Fontos, hogy a megfelelő működéshez az .env fájlunk nevét .env.local.example-ről, .env.local -ra írjuk át, ha még nem tettük meg, mivel csak így lesz elérhető a program számára

Ha pedig mind ezt megtettük, készen áll a programunk ahoz, hogy mobilon is tesztelhető legyen. Mindössze a korábban az indítás után megkapott Network ip-címet kell beírni a mobil telefonunk böngészőjébe, és már használható a program. (Fontos, hogy ne felejtsük el a <http://> az elejéről)



2.2.3 Seedelés által generált teszt fiókok

Ha backenden sikeresen lefuttattuk a seedelést a `bash run-seeds.sh` parancs segítségével, létrejött pár felhasználói fiók, ami segíthet a program egyes funkció tesztelésében.

Admin fiók:

email	admin@gmail.com
jelszó	Admin123.
felhasználónév	Admin
admin jogosultság	rendelkezik admin jogosultsággal

Az admin fiókot használva elérhetővé válik számunkra:

- az admin felületek használata a weboldalon
 - Gyakorlatok szerkesztése
 - Új gyakorlat hozzáadása
 - Felhasználók törlése, admin jogosultság nyújtása
- Más felhasználók adatai megtekintése, végpontok segítségével
- az extra admin jogosultságok mellett a fiók kapott edzés előzményeket is

Szimpla felhasználó fiókok:

email	jelszó	felhasználónév	admin jogosultság
user@gmail.com	User123.	BasicUser	nem rendelkezik admin jogosultsággal
pepesdenes@gmail.com	Pepes123.	PepesDenes	nem rendelkezik admin jogosultsággal
veghbela@gmail.com	Vegh123.	VeghBela	nem rendelkezik admin jogosultsággal

Ezek a felhasználói fiókok átadják a normál felhasználói élményt. Részletes edzés előzményekkel és ezáltal statisztikákkal, amik jól jöhetnek tesztelés során.



3. Felhasználói dokumentáció

Bevezetés

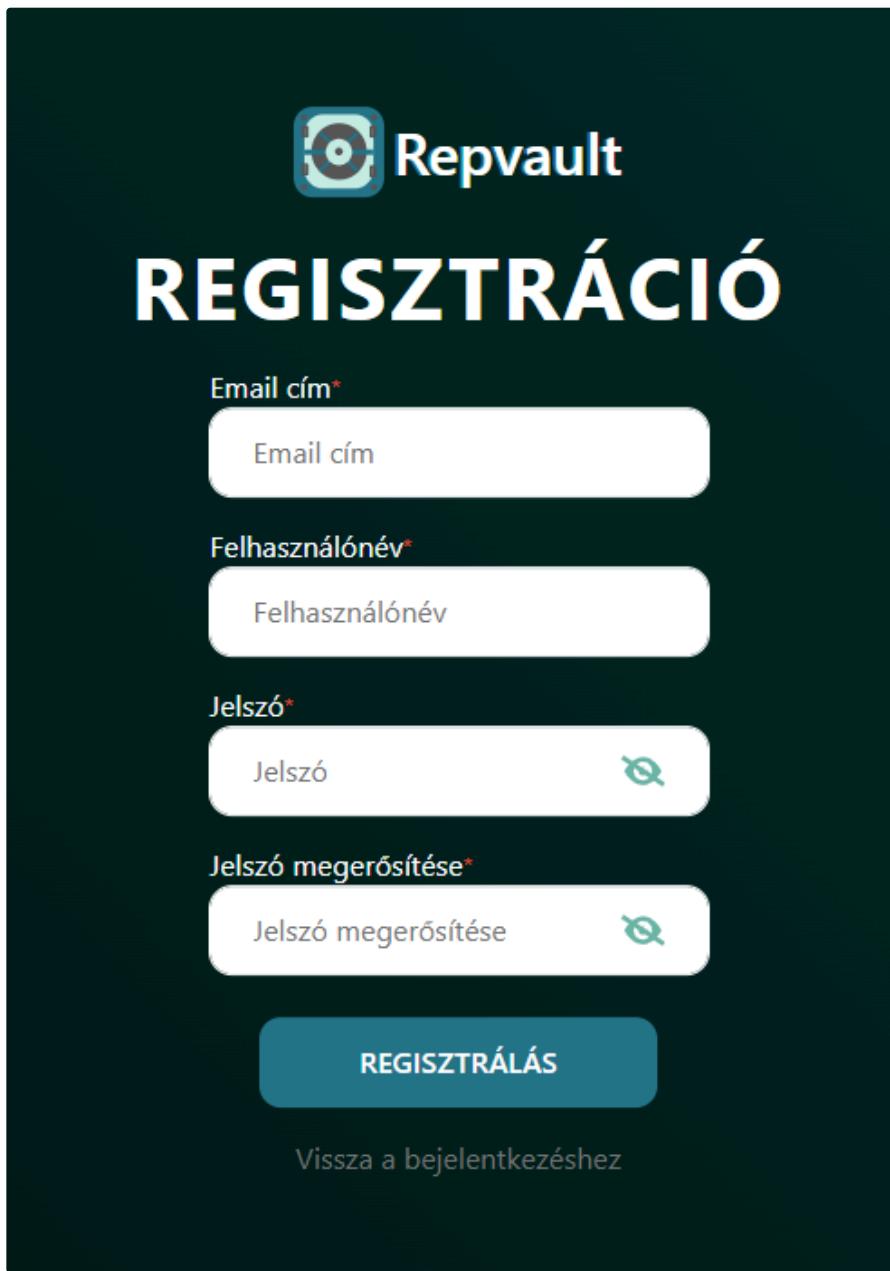
A repvault projekt átfogó felhasználói dokumentációja, amely segítségével bárki könnyedén elnavigálhat a webalkalmazás számos oldalán, és fényt deríthet minden esetlegesen felmerülő hibaüzenet jelentésére, és arra, hogy az egyes beviteli mezőknek milyen értékeket adhat meg.

3.1 Regisztráció/Bejelentkezés

Összefoglaló

Az alábbi pontokban, A regisztráció és bejelentkezés lépéseiiről szerezhetünk átfogó tudást.

3.1.1 Regisztráció



The screenshot shows the registration page for Repvault. At the top is the Repvault logo, which consists of a blue square icon with a white target symbol followed by the word "Repvault" in blue and orange. Below the logo is a large, bold title "REGISZTRÁCIÓ". The form contains four input fields: "Email cím*" (Email address), "Felhasználónév*" (Username), "Jelszó*" (Password), and "Jelszó megerősítése*" (Password confirmation). Each input field has a placeholder text and a small circular icon with a checkmark or question mark to its right. Below the input fields is a large blue button labeled "REGISZTRÁLÁS" (Register). At the bottom of the form is a link "Vissza a bejelentkezéshez" (Back to login).

regisztráló felület

Lépések ↗

- Felhasználó regisztrációja
- kötelező mezők:
 - felhasználónév (legalább 3 karakter)
 - email cím (egyedi egy email-hez egy felhasználó tartozhat, az email címnek valósnak kell lennie)

- jelszó legalább (8 karakter legalább 1-1 kisbetű, nagybetű, különleges karakter, szám)
- jelszó megismétlése (a két jelszónak egyeznie kell)
- Az egyes beviteli mezők piros körvonalra váltanak ha nem megfelelő a bevitt érték.
- Sikeres regisztráció esetén, sikeres regisztrálás üzenetet kapunk, és visszakerülünk a bejelentkezés oldalra.
- Sikertelen regisztráció esetén 2 hibaüzenetet kaphatunk.
 - Már rendelkezik felhasználó ezzel az email címmel, ami azt jelenti hogy már van ezzel az email címmel regisztrálva fiók.
 - A másik hiba a hiba a regisztráció során, ez valamilyen szerver hibát jelent, ha ezt az üzenetet kapjuk valószínűleg valamilyen háttér hiba lépett fel.

3.1.2 Bejelentkezés



Lépések ☈

- A korábban regisztrált fiókunk adatait megadjuk.
 - email: a regisztrált email cím
 - jelszó: a fiókhöz tartozó jelszó
- ha jól adtuk meg az adatainkat és létezik a felhasználói fiókunk, a bejelentkezés gombra kattintva, üzenetet kapunk a sikeres bejelentkezésről, és a program átnavigál minket a főoldalra.
- Ha valamit nem jól adtunk meg, és úgy kattintunk a bejelentkezés gombra az oldal hibaüzenetet küld és ottmaradunk a bejelentkezés oldalon.

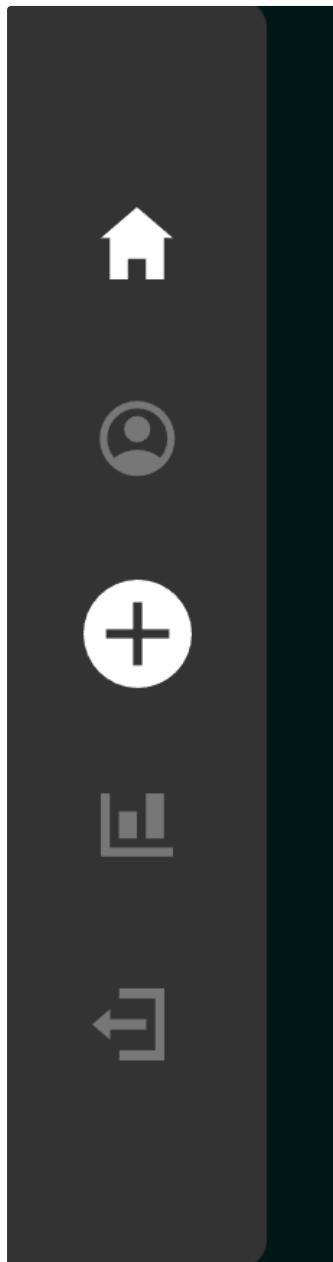
! A főoldal és az aloldalak csak akkor elérhetőek ha be vagyunk jelentkezve.

 Bejelenetkezéstől számított 1 hézig, nem kell újra bejelentkeznünk, ha nem jelentkezünk ki manuálisan.

3.2 Navigációs sáv

Bevezetés [🔗](#)

Az alábbiakban a navigációs panel különböző gomjairól olvashatunk.



Az oldalsó navigációs sáv



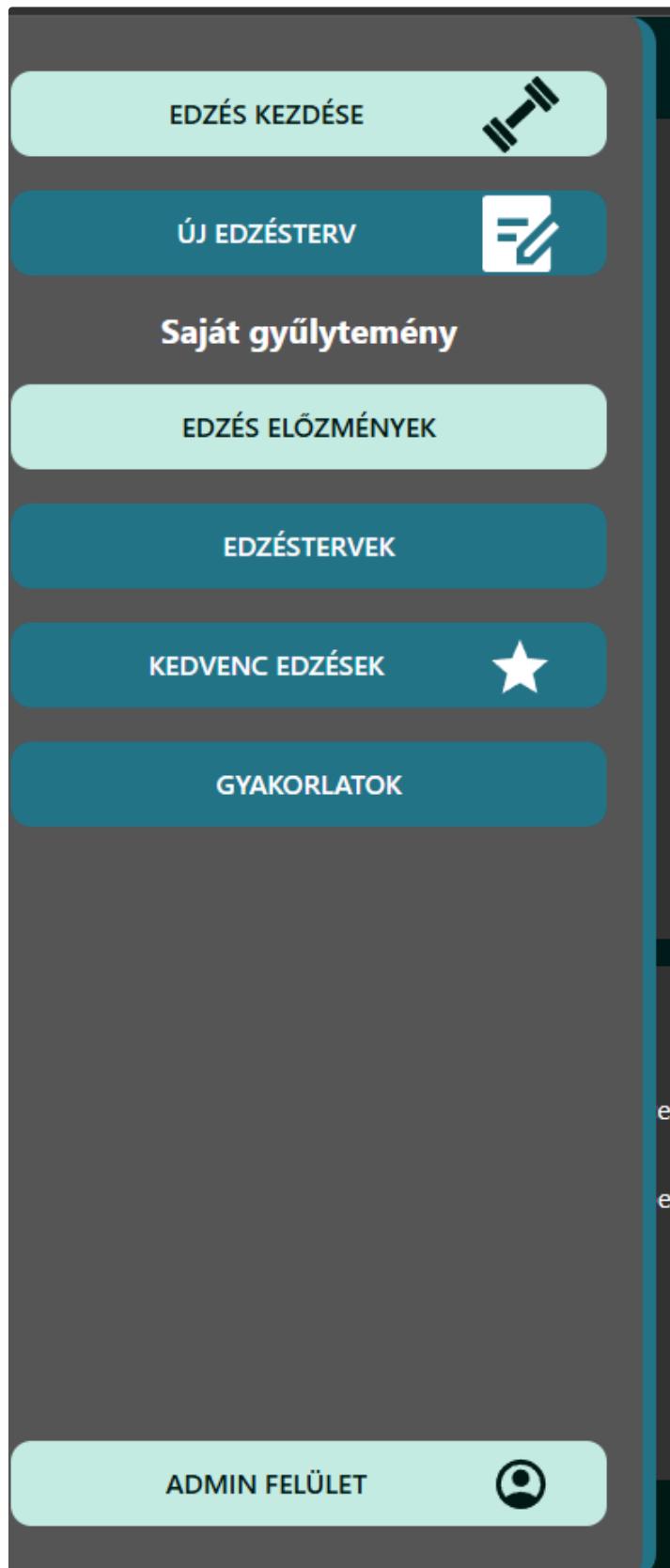
Navigációs sáv mobil eszköz nézeten

3.2.1 gombok



navigációs
sáv

- házikó ikon átvisz minket a főoldalra
- Emberke ikon átvisz minket a profil oldalra
- diagram ikon átvisz a statisztika oldalra
- kijárat ikon kijelentkeztet bennünket és átvisz a bejelentkezés oldalra
- pluszjel ikon felhoz egy plussz menüt



plussz menü

- Edzés Kezdése/edzés folytatása: elindítja a mai edzsünket, vagy ha már elkezdtük, de még nem végezték, akkor folytatni tudjuk, ha az edzést már véglegesítettük, információs üzenetet kapunk arról, hogy a mai edzsünket már megcsináltuk.
- Új edzésterv :Először nevet kell adnunk az új edzéstervhöz, majd átnavigál az edzésterv készítő edzéstervhöz oldalra.
- Edzés előzmények: átvavigál az edzések oldalra.

- Edzéstervezek: átnavigál az edzéstervezek oldalra.
- Kedvenc edzések: átnavigál a kedvenc edzések oldalra.
- gyakorlatok: átnavigál a gyakorlatok oldalra.
- (adminoknak) Admin felület: átnavigál az admin oldalra.

3.3 Főoldal/dashboard

Bevezetés

Az alkalmazás főoldala, itt 5 különböző modullal találjuk szembe magunkat ezek:

- napi edzés
- heti edzés
- naptár
- rekordok
- statisztika

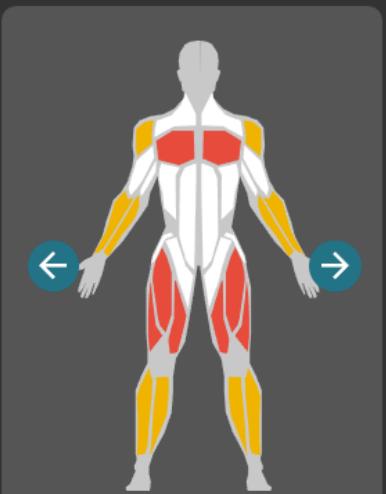
Ezek a modulok mint katinthatóak, és legtöbbüknek megvan a saját teljes képernyős aloldala amiről a későbbiekben olvashatunk.

3.3.1 Napi edzés

Mai Edzés

Barbell Deadlift	3x10
Barbell Full Squat	3x12
Barbell Bench Press - Medium Grip	3x8

EDZÉS MEGTEKINTÉSE



Mai edzés modul példaértékekkel

Baloldalt láthatók a gyakorlataink, és hogy hányszor hány ismétlést csináltunk/tervezünk végrehajtani a mai edzésen.

Alatta az edzés megtekintése gombra kattintva eljuthatunk ennek az edzésnek az oldalára, ott több adatot tudhatunk meg az edzésről.

A bal oldalon látható figura szemlélteti hogy milyen izmainkat fogjuk megozgatni/mozgattuk meg a mai edzésen, a kétoldali kis nyilacsákkal a figurát szemből, és hátulról is láthatjuk.

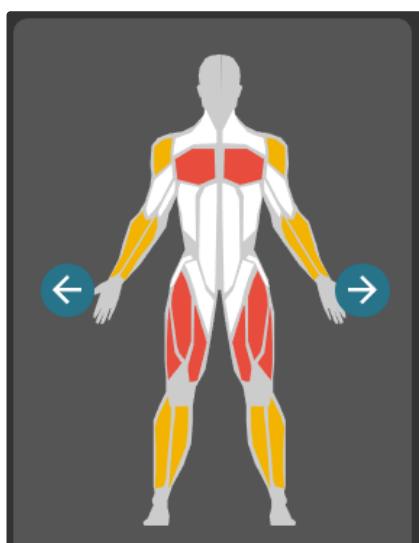


figura előlről

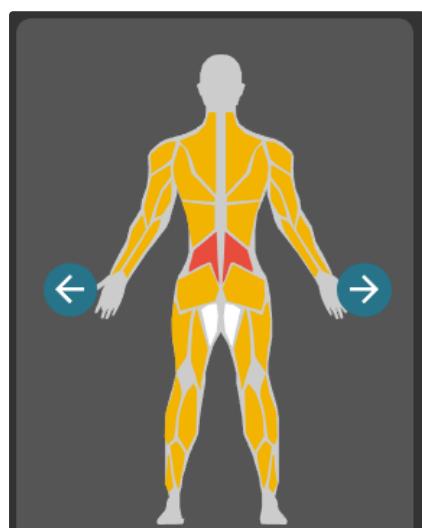


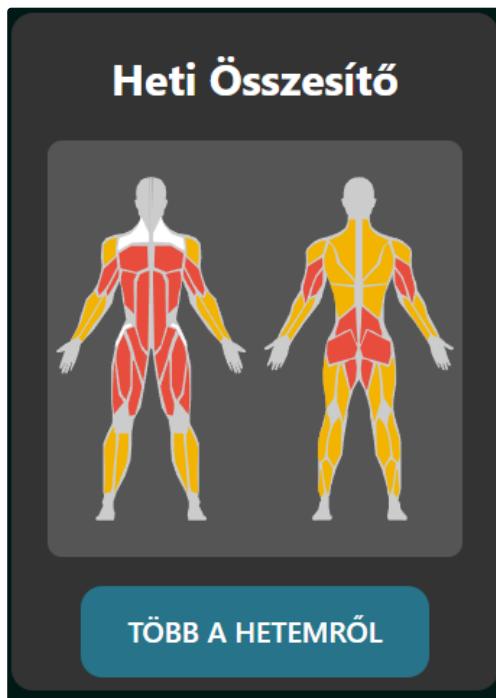
figura hátulról

Ha esetleg még nincs a mai napra edzésünk, az alkalmazás így jelzi.

Még nincs edzés a mai napra

Nincs edzés példa

3.3.2 Heti edzés



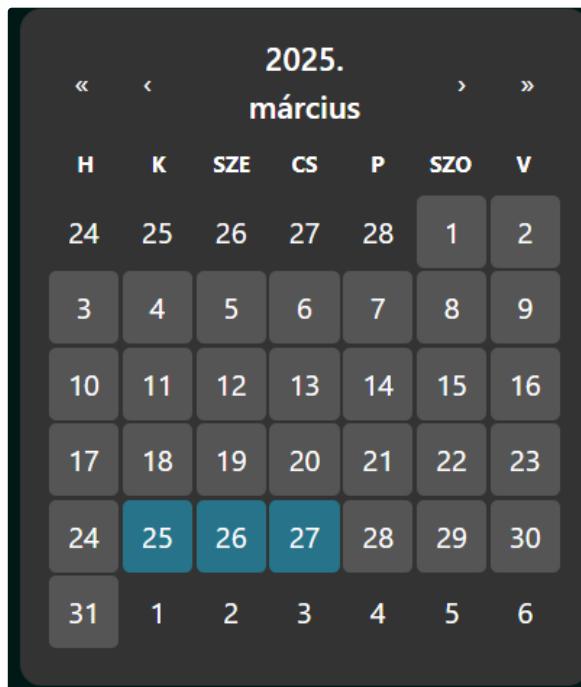
Heti edzés modul

Ezen a modulon látható, az adott héten lévő edzéseink összesítése.

A figuráról és hátulról is látható, és beszíneződnek rajta azok az izomcsoportok, amelyeket leadzünk és amelyeket várhatóan le fogunk edzeni, ha valamely edzésünk módosul a héten az itt is megjelenik.

A "Több A Hetemről" gomb, átnavigál minket egy külön oldalra, ahol az adott heti edzéseink jelennek meg részletesen.

3.3.3 Naptár

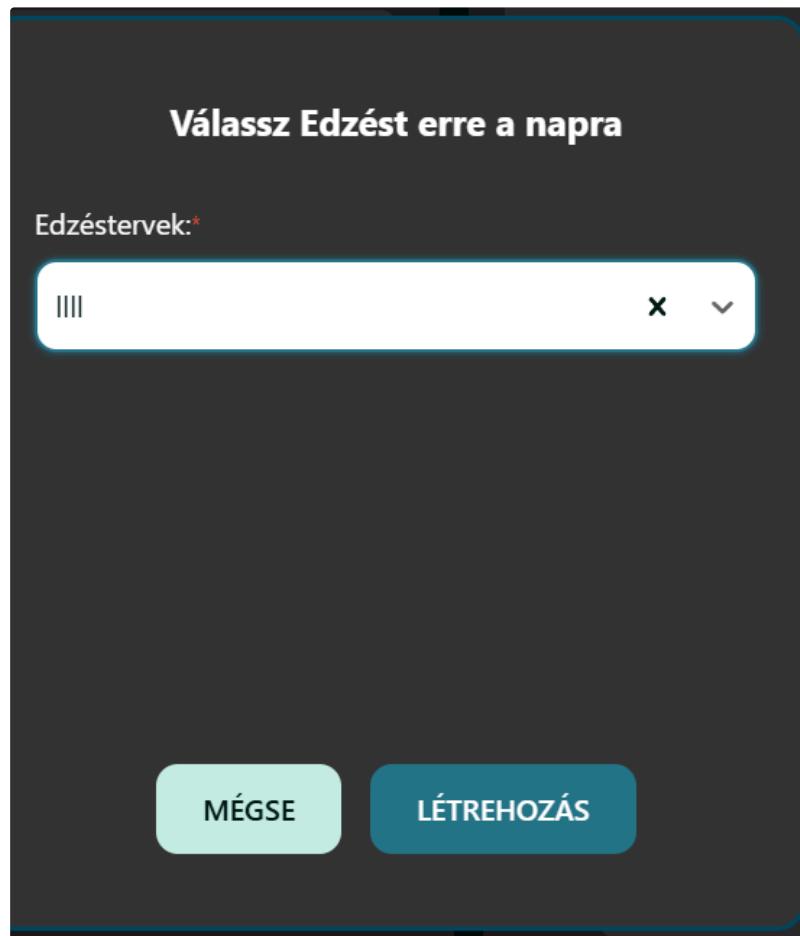


naptár

A naptáron látható, hogy mely napokon vannak edzéseink.

Amennyiben olyan napra kattintunk amelyiken, van edzésünk, megjelennek az az napi edzéseink részletei.

Különben felugrik egy ablak, ahonnan kiválaszthatjuk, hogy melyik edzéstervünket szeretnénk arra a napra tenni.



Felugró ablak

A mégse gombra kattintva bezárul az ablak.

A létrehozás gombra kattintva, létrejön az új edzés, és a jobb felső sarokban megjelenik egy értesítés, hogy létrejött az edzés, és hogy pontosan milyen napon.

Ha nem sikerült valamiért létrehozni az edzést, akkor ugyanott hibaüzenet jelenik meg.

3.3.4 Rekordok

Rekordok

Hasprés Gép:	85.3 kg	Váltott Kalapács	84 kg
		Bicepsz:	
Haladó Kettlebell	82.6 kg	Adduktor:	81.8 kg
Szélmalom:			
Air Bike:	78.8 kg	3/4 Felülés:	74.3 kg

TOVÁBBI REKORDOK 

Rekordok

Ezen a modulon láthatóak a rekordjaink, a 6 (mobileszköz nézetben 3) legnagyobb súlyú gyakorlatunk jelenik meg.

A további rekordok gombra kattintva, átnavigál minket a rekordok oldalra, ahol az összes rekordunk látható.

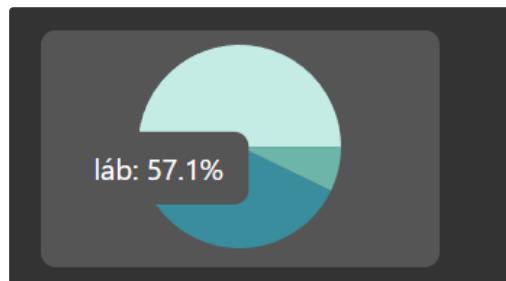
3.3.5 Statisztika



Statisztikák

A statisztika modul, Az adott heti összesített értékeinket mutatja meg, a bal oldalt látható a héten összes megmozgatott súly az edzéseink során.

A jobb oldalt pedig a megmozgatott izomcsoportjaink százalékos eloszlása.



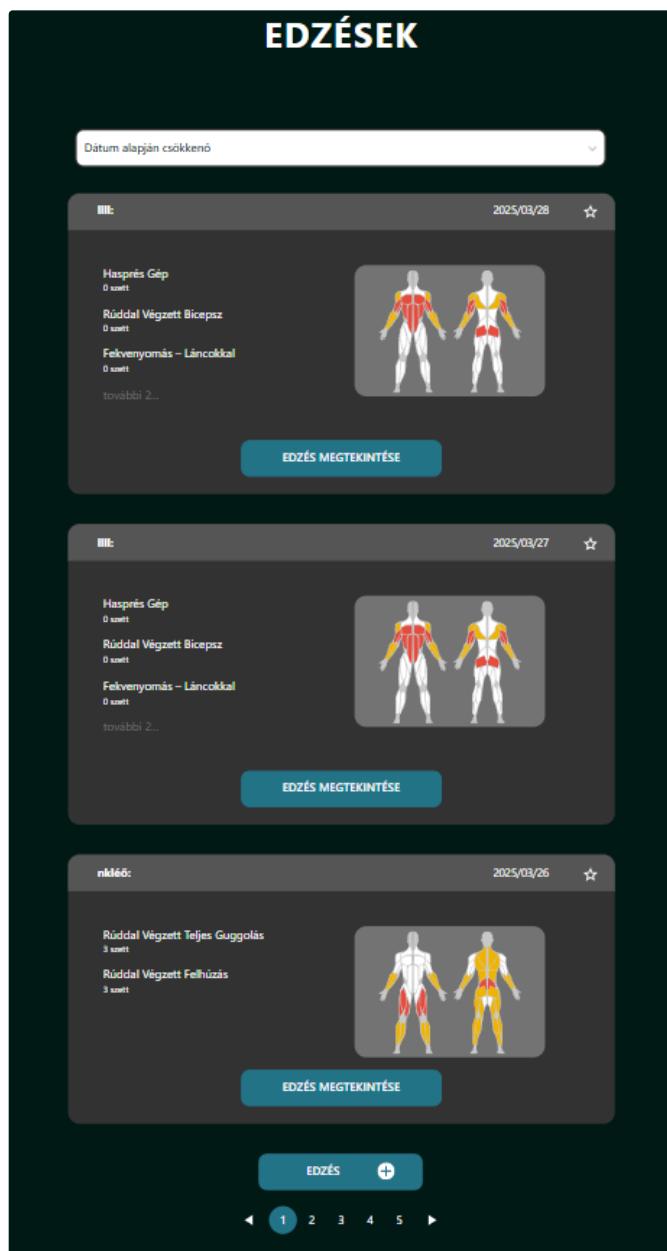
százalékos eloszlás

3.4 Edzések

Bevezetés

Ebben a fejezetben betekintünk az edzések és edzéstervezek részleteibe.

3.4.1 Edzések Oldal

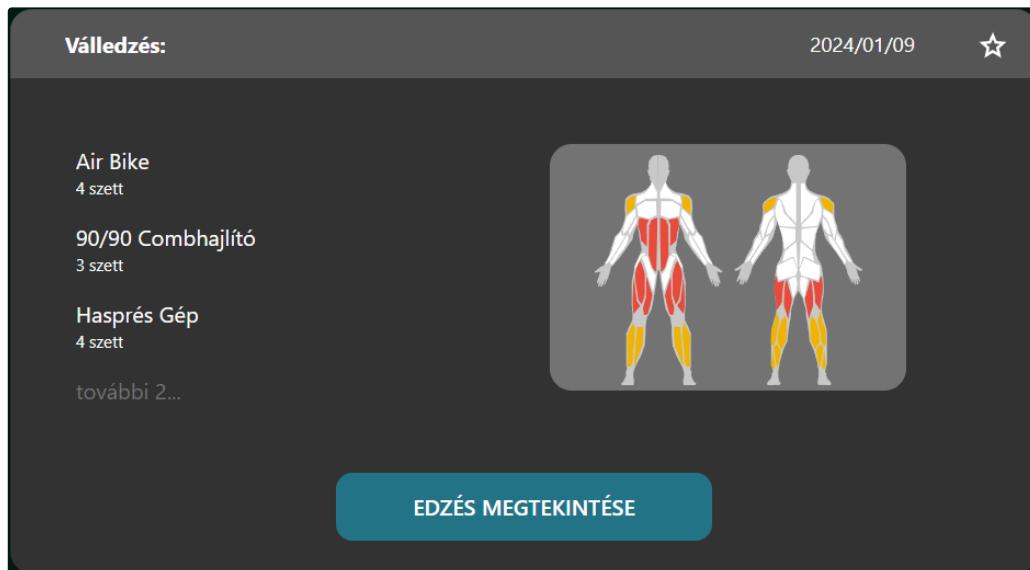


Az edzések oldalon, az összes edzésünket tudjuk kezelni, a felső szűrőben a kiíratás sorrendjét tudjuk megváltoztatni.

Három lehetőség közül választhatunk

- Kedvencek alapján (A kedvenc edzéseket állítja előre)
- Dátum alapján növekvő (Az edzéseket dátum alapján rendezzi, régiektől az újakig)
- Dátum alapján csökkenő (Az edzéseket dátum alapján rendezé újaktól a régiekig, ez az alapértelmezett rendezés)

Az oldalon egyszerre csak 3 edzés kerül kiírásra, a többi edzést az oldal alján található lapozóval érhetjük el.

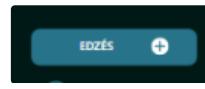


Egy edzés

A különböző edzéseket ilyen kártyákon láthatjuk, itt látszik, hogy miből mennyit csináltunk, mikor volt/lesz ez az edzés, és a jobb felső sarokban lévő kis csillag segítségével, kedvenccé tudjuk tenni.

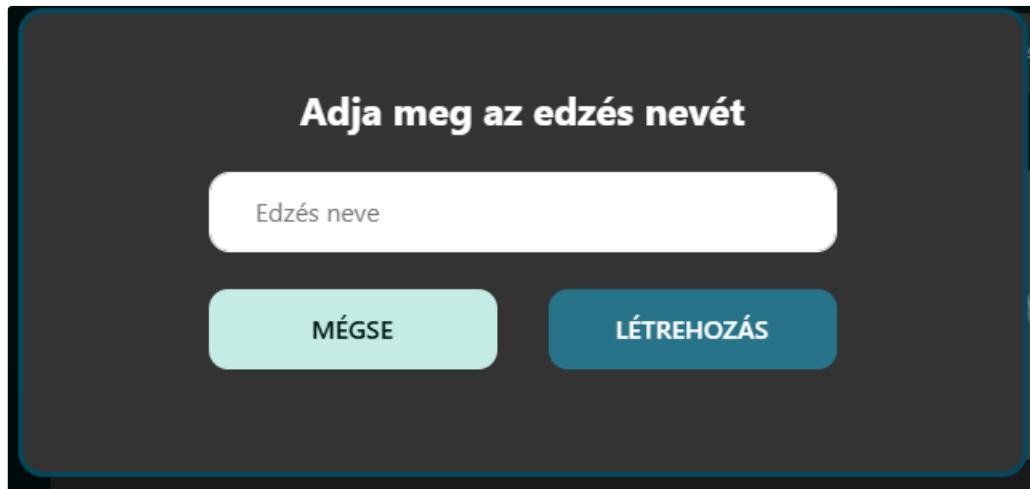
Ha teli a csillag, az azt jelenti, hogy az edzésünk már kedvenc, ha rákattintunk akkor azzal kivesszük a kedvencek közül az adott edzést.

Az edzés megtekintése gombra kattintva, több részletet tudhatunk meg az edzésünkről, ha ma van az edzés akkor elindíthatjuk, és akár törölhetjük is.



edzés hozzáadása
gomb

Ezzel a gombbal új edzést kezdhetünk, ha még nincs a mai nap edzésünk.



felugró ablak

Ha rákkantintunk erre a gombra, megjelenik egy ablak ahol, megadhatjuk új edzésünk nevét, majd létrehozhatjuk azt a létrehozás gombbal, ha esetleg meggondolnánk magunkat, akkor a mégse gombbal bezárthatjuk, a felugró ablakot.

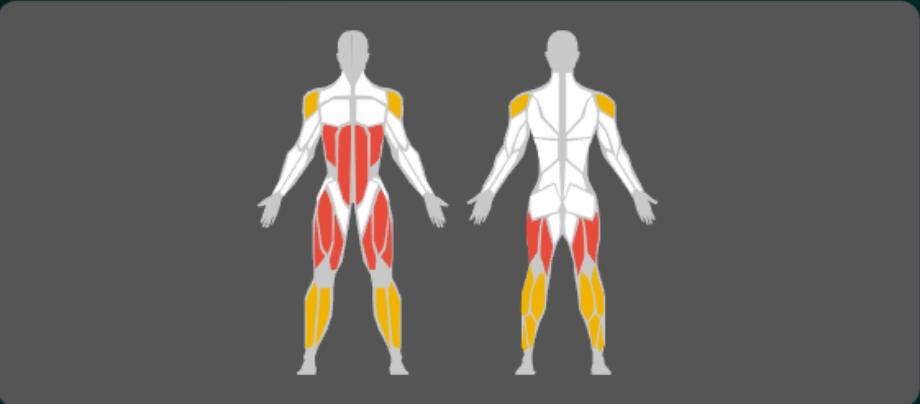
3.4.2 Egy edzés részletei Oldal

VÁLLEDZÉS

01:17:00

ÚJ EDZÉS KEZDÉSE ▶

VISSZA TÖRLÉS



Gyakorlatok

Hasgörgő:					
Előző Alkalom			Most		
Szett	KG	Ism.	Szett	KG	Ism.
1	58.7	8	1	68.9	9
2	57.5	11	2	64.5	12
3	62.3	9	3	64.4	12
			4	66.9	8
			5	65.8	9

lezárt edzés

Itt látható egy edzés, nézzük meg miből áll, legfelül van az edzés neve, alatt látható, hogy mennyi ideig tartott, alatta, ha szeretnénk, kezdhetünk egy új edzést, ami ugyanazokat a gyakorlatokat fogja tartalmazni, mint ez az edzés,

alatta látható a vissza gombunk, ezzel visszatérhetünk a lista oldalra, a mellette lévő gombbal pedig törölhetjük az edzésünket,

ez alatt látható a jól ismert figuránk, amelyen láthatjuk mely izomcsoportokat mozgattuk meg,

ez alatt pedig láthatóak, a gyakorlataink, hogy mennyit csináltunk belőle az ezt megelőző alkalommal, amikor ugyanezt a gyakorlatok hajtottuk végre.

Látható, hogy meg vannak számozva a szettjeink. és ha a múltkori alkalomhoz képest, egy szettben jobban teljesítettünk, akkor zölddel jelenik meg az súly, és az ismétlésszám, ha esetleg rosszabbul teljesítettünk volna, akkor pirossal jelenne meg.

Bár a képen csak egy gyakorlat látható, az edzésben ledolgozott összes gyakorlatot láthatjuk, mint a képen az az egyet.

3.4.3 Aktív edzés

VÁLLEDZÉS

00:00:30

Edzés neve*

Gyakorlatok

Hasgörgő:		Előző alkalom				Most			
		KG	Ism.	KG	Ism.				
1		68.9	9	55	2	-			
2		64.5	12	0	0	-	SZETT	+	
3		64.4	12	-	-				
4		66.9	8	-	-				
5		65.8	9	-	-				

Air Bike:

SZETT +

Aktav edzés

Itt látható, az az edzés, amit éppen csinálunk, felül a név, hasonló módon mint az egy edzésnél, alatt látható a név, amit bármikor megváltoztathatunk, egy dogra kell figyelnünk, hogy az edzés neve nem lehet 3 karakternél rövidebb.

A beviteli mező alatt pedig láthatóak a gyakorlatink.

Hasgörgő:			
Előző alkalom		Most	
	KG	Ism.	KG
1	68.9	9	55
2	64.5	12	0
3	64.4	12	-
4	66.9	8	-
5	65.8	9	-

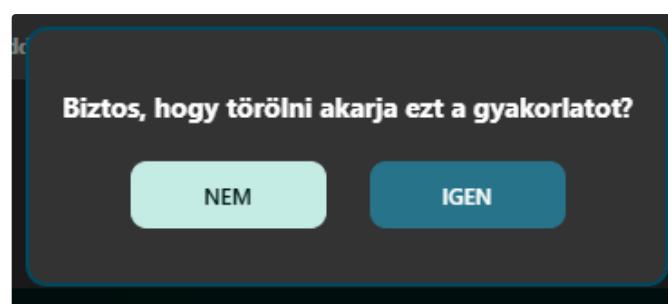
Egy gyakorlat

Itt látható, egy gyakorlat, ismételten láthatóak, az előző alkalom szetjei, most azonban, jobb oldalon megjelent egy új gomb, a szett + gomb, ezzel szetteket adhatunk a gyakorlathoz, és most felirat alatt, súlyt és imzsítést, adhatunk hozzájuk

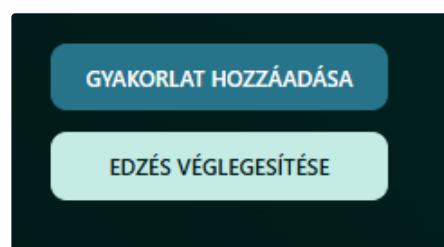
Ezeknek a beviteli mezőknek nem adhatunk meg negatív értéket.

A mellettük lévő kis minusz gombbal kitörölhetjük, ekkor üzenetet kapunk, hogy "Szett törlőve."

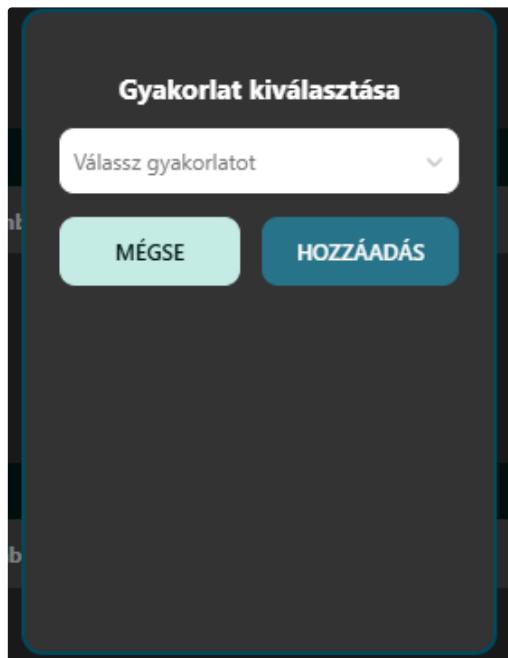
A gyakorlat jobb felső sarkában található, kis x-el kitörölhetjük a gyakorlatot, rákattintva felugrok az alábbi ablak.



A nem gombra kattintva bezárul az ablak, az igen gombra kattintva, szintén bezárul az ablak, és kitörődik a gyakorlat, majd üzenetet kapunk: "Gyakorlat törlőve." felirattal.



Az oldal alján, található két gomb, a gyakorlat hozzáadása gombra kattintva, megjelenik egy felugró ablak



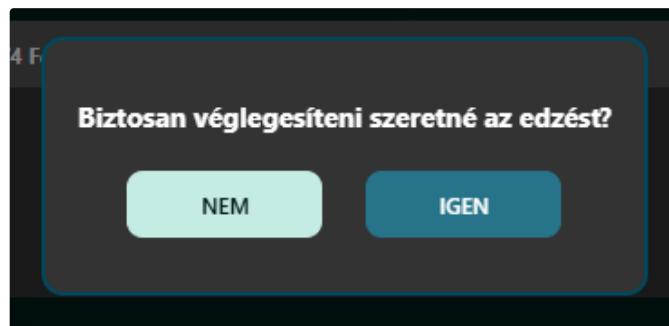
Itt kiválaszthatjuk melyik gyakorlatot szeretnénk hozzáadni az edzésünkhez, a beviteli mezőbe kattintva, megjelennek gyakorlataink, és tudunk köztük böngészni, azonban ha mi egy kimondott gyakorlatra vagyunk kíváncsiak, elkezdhetjük begépelni a nevét, és a program automatikusan kiszűri, nekünk, a gyakorlatot.

Ha a hozzáadás gombra kattintuk, és nincs kiválasztva gyakorlat, nem történik semmi, ellenben ha van gyakorlat kiválasztva, eltűnik a mező, üzenetet kapunk, "gyakorlat hozzáadva" és az új gyakorlat a gyakorlataink listájának a végére kerül.

A mégse gombbal, bezárul az ablak.



Ezzel a gombbal, vélegesíthetjük az edzést, ha így teszünk, már nem módosíthatunk a gyakorlatinkon, és azon se amit beléjük írtunk vagy nem írtunk. Az számláló is megáll, és az lesz az edzésünk ideje, de mint minden más nagy döntésünknél az oldalon. I

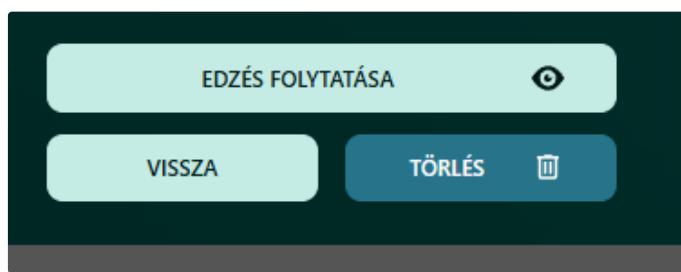


Itt is felugrik egy ablak, ahol megerősíthetjük elkötelezettségünk a vélegesítés iránt, amennyiben meggondolnánk magunkat, a nem gombra kattintva bezárul az ablak.

Amennyiben az igen gombra kattintunk az edzés vélegesítésre kerül és "Edzés befejezve" üzenettel, a mostan edzésünk részletei oldalra.

Amennyiben, véletlenül vagy direkt, elhagytuk az edzés oldalát, az edzéseink között megtaláljuk, és új edzés kezdése helyett, edzés folytatása gomb üdvözöl bennünket, ha erre rákattintuk, visszakerülünk a szerkesztőfeleületre

- Fontos megemlíteni, hogy amíg nem vélegesítjük az edzésünket, addig az időmérő nem áll meg.



3.5 Edzéstervezek

Bevezetés

Ebben a fejezetben, megnézzük, hogy milyen módon hozhatunk létre edzéstervezeket, és ezeket hogyan hasznosíthatjuk.

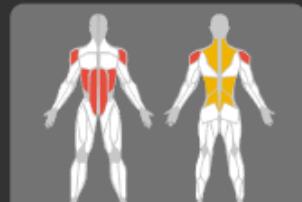
3.5.1 Edzéstervezek Oldal

EDZÉSTERVEK

Dátum alapján növekvő

Vállalás:

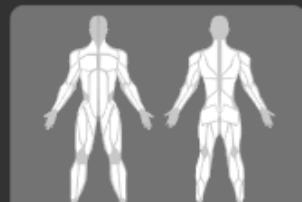
- Hasprás Gép has
- Váltott Sarkirányás has
- Döntött Alacsony Kábel Oldalemlés vall hit



[EDZÉSTERV MEGTEKINTÉSE](#)

Teszt/Edzés1:

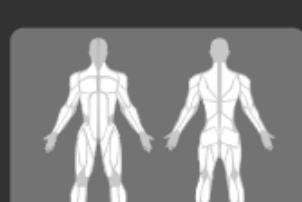
Nincsenek gyakorlatok



[EDZÉSTERV MEGTEKINTÉSE](#)

Teszt/Edzés2:

Nincsenek gyakorlatok



[EDZÉSTERV MEGTEKINTÉSE](#)

[EDZÉSTERV](#) +

◀ 1 ▶

edzésterv oldal

Az edzéstervezek ugyanúgy, működik mint az edzések oldal, ugyanaz a filterező van rajta, ugyanúgy 3 kártyát ír ki egy oldalra, ugyanaz a folyamat, ha újat szeretnénk létrehozni, és a lapozó és ugyanúgy működik. Az egyetlen lényeges különbség magukban a kártyákban rejlik.



edzésterv kártya

Mint látható, itt nem szettek jelennek meg, hanem, hogy egyes gyakorlatok mely izomcsoportokat mozgatják meg, itt is megfigyelhető jobb oldalt a kis figura, ami összesíti, hogy az edzésterv mely izomcsoportokat mozgatja meg. Edzésterveket is lehet kedvenccé tenni a csillag ikonnal.

Az edzésterv megtekintése átvisz az adott edzésterv oldalára.

3.5.2 Egy edzésterv oldal

EDZÉSTERV SZERKESZTÉSE

Válledzés

Edzés neve*

Válledzés

Érintett izomcsoportok

Gyakorlatok

Hasprés Gép:

Válasszon fényellenállást, és üljen le az ab gépre, tegye a lábat a mellékelt párnák alá, és fogja meg a felső fogantyúkat. A karjait 90 fokos szögben kell hajlítani, miközben a tricepszt a mellékelt párnára támasztja. Ez lesz a kiindulási pozíció. Ugyanakkor kezdje el felemelni a lábakat, mik...

Több

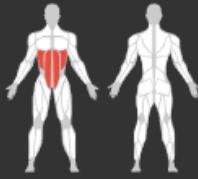
Felül látható, az edzésterv neve, az ez alatt lévő mezővel, megváltoztathatjuk ezt, ennek is legalább 3 karakterből kell állnia, úgy mint az edzsések. Allatt látható a kis figuránk szokásosan.

Ez alatt vannak a gyakorlatok.

A fejlécben található a gyakorlat neve amire ha rákattintunk, átnavigál bennünket, az adott gyakorlat oldalára, az x gombbal pedig törölhetjük a gyakorlatot, az edzéstervünkön.

A fő részben a jobb oldalon látható, hogy a gyakorlat mely izmokat mozgatja meg, a bal odralt pedig, a gyakorlat részletes leírása. A leírás alatt, ha rákattintunk a "Több" gombra, az egész leírás megjelenik.

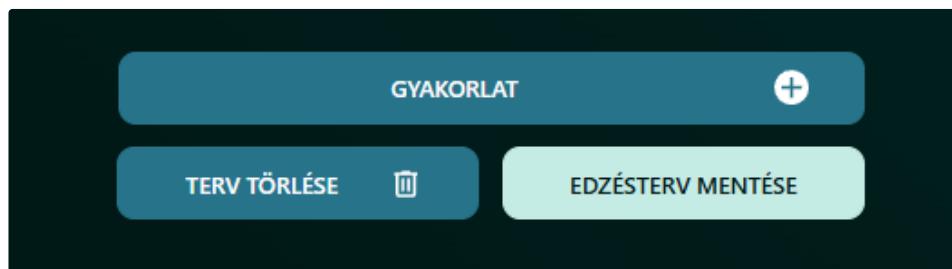
Hasprés Gép:



Válasszon fényellenállást, és üljön le az ab gépre, tegye a lábat a mellékelt párnák alá, és fogja meg a felső fogantyúkat. A karjait 90 fokos szögben kell hajlítani, miközben a tricepsz a mellékelt párnákról támasztja. Ez lesz a kiindulási pozíció. Ugyanakkor kezdje el felemelni a lábakat, miközben megragadja a felső törzset. Lélegezzen ki, miközben ezt a mozgást hajtja végre. Tipp: Ügyeljen arra, hogy lassú és szabályozott mozgást használjon. Koncentráljon arra, hogy hasmadával mozgassa a súlyt, miközben lazítja a lábat és a lábát. Egy második szünet után lassan térjen vissza a kiindulási helyzetbe, miközben belélegzel. Ismételje meg a mozgást az előírt mennyiségű ismétléshez.

Kevesebb

Itt látható, hogy az egész leírás kiírásra került, ha itt rányomunk a kevesebb gombra, akkor a gyakorlat visszatér eredeti állapotába.



A felsorolt gyakorlatok után, ez a három gom látható, a gyakorlat+ ugyanazt a felugró ablakot dobja fel, mint az edzésekknél, és ugyanazt az üzenetet küldi, amikor hozzáadunk egy gyakorlatot az edzéstervünkhez.

A terv törlése, megkérdezi, hogy biztos szeretnénk-e törölni az edzést, ha igenre kattintunk, akkor kitörli a tervet, visszadob minket a lista oldalra, és üzenetet küld a sikeres törlésről. Ha nemre kattintunk akkor bezárul.

Az edzésterv mentése, elmenti a változtatásainkat, de előtte, egy felugró ablakkal, megkérdezi, hogy biztos szeretnénk-e menteni, ha nemre kattintunk, bezáródik, ha igenre kattintunk, akkor elmenti az edzéstervet és visszadob minket a listaoldalra, "Edzésterv Elmentve." üzenettel.

- ! Ellentétben az edzéssel, itt a mentés után, bármikor módosíthatjuk edzéstervünk, az viszont figyelembe kell vennünk, hogy ez a változás csak a módosítást követően létrehozott edzéseinkre lesz kihatással, a módosítás előtt, de még le nem edzéseink is maradnak a régi edzésterv szerintiek. Ha ezeket szeretnénk változtatni, ki kell törölünk a szóbanforgó edzéseket, és újra létre kell hoznunk őket, a már módosított edzéstervünkkel.

3.6 Gyakorlatok

Bevezetés

Ebben a fejezetben a gyakorlatokról kapunk átfogó információt, és hogy miként tudjuk/nem tudjuk manipulálni a gyakorlatokat.

3.6.1 Gyakorlatok Oldal

GYAKORLATOK

Filterek

Gyakorlat neve:

Régiókategória:

Választható kategóriák:

Izomcsoportok:

Választható izomcsoportok:

Gedők csoport:

KERESÉS

[ÚJ GYAKORLAT](#)

Gyakorlatok

Hipopláti Gép		TÖBB INFO
Hangúgá		TÖBB INFO
Adduktoryi/Légyük		TÖBB INFO
Haladt Kettőbeli Számlálás		TÖBB INFO
Air Bike		TÖBB INFO
Mélykötésű Cserélyijász		TÖBB INFO
Váltott Kalapács Röplap		TÖBB INFO
Váltott Sarkárlatás		TÖBB INFO
Váltott Ferde Padas Kád Röplap		TÖBB INFO
Váltott Átlós Lébagás		TÖBB INFO

◀ 1 ▶

Ezen az oldalon látható az összes gyakorlatunk, a gyakorlatok oldalakra vannak szedve, egy oldalon 10 gyakorlat látható.

GYAKORLATOK

Filterek

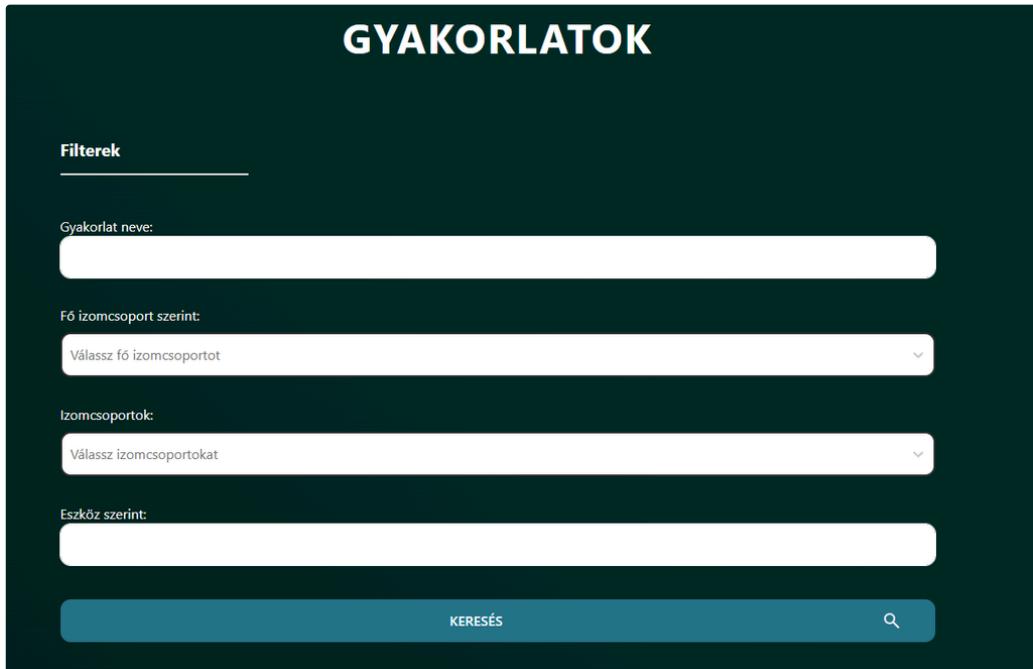
Gyakorlat neve:

Fő izomcsoport szerint:

Izomcsoportok:

Eszköz szerint:

KERESÉS 



Gyakorlat filter

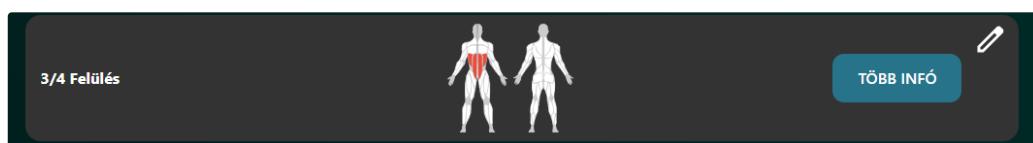
Az oldal tetjén található szűrővel tudunk gyakorlatokra szűrni,

- Gyakorlat neve (ide be tudjuk írni a gyakorlat nevét, részben vagy egészben, és azokat a gyakorlatokat írja ki amelyekben szerepel a beírt szöveg)
- Fő izomcsoport szerint (itt egy egy izomcsoportot tudunk, kiválasztani, és azokat a gyakorlatokat írja ki, amelyeknek ez a fő (figurán pirossal jelölt) izomcsoportja).
- Izomcsoportok (itt több izomcsoportot is ki tudunk választani, és aztokat a gyakorlatokat kapjuk válaszként, amelyekben szerepel az összes kiválasztott, mellék (figurán sárgával jelölt) izomcsoport).
- Eszköz szerint (ide, be tudjuk gépelni, az eszköz nevét, részben vagy egészben, és amely eszköz névében szerepel a begépelt szöveg, azokat a gyakorlatokat adja vissza)

 Egyszerre több szűrővel is kereshetünk

A keresés gombra kattintva megtörténik a szűrés, és utána az oldal, már csak a szűrőnek megfelelő gyakorlatokat írja ki.

(Adminoknak, a keresés gomb alatt, van egy új gyakorlat gomb, ami átnavigál az új gyakorlat készítése oldalra.)



Gyakorlat kártya

Egy-egy gyakorlat kártyája így néz ki, baloldalt látható a gyakorlat neve,

középen a figura, hogy mely izmokat edzi meg,

jobb oldalt pedig a "TÖBB INFÓ" gomb, amire rákattintva, átkerülünk, az adott gyakorlat oldalára, ahol több részletet tudhatunk meg a gyakorlatról.

 Adminisztrátorként, a jobb felső sarokban megjelenik egy plussz gomb, a szerkesztés gomb, ez átvisz minket a gyakorlat szerkesztése oldalra.

3.6.2 Egy gyakorlat Oldal

3/4 FELÜLÉS

[VISSZA](#) [REKORDOK](#) [EDZÉS ELŐZMÉNYEK](#) [DIAGRAMOK](#)

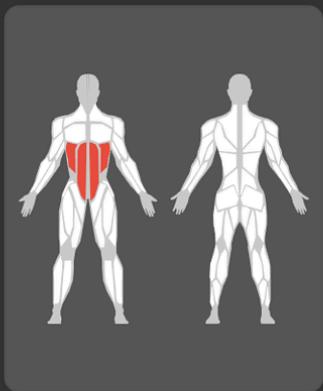
Eszköz: saját testsúlyos

Elsődleges izom: Hasizom

Másodlagos izmok: Hasizom

Gyakorlat leírása :
Feküdjön le a padlóra, és rögzítse a lábat. A lábat térdre kell hajlítani. Helyezze a kezét a feje mögé vagy oldalára. Kezdje hátradal a földön. Ez lesz a kiindulási pozíció. Hajlítsa meg a csípőjét és a gerincét, hogy a törzset a térd felé emelje. A összehúzódás tetején a törzsnek merőlegesnek kell lennie a talajra. Fordítsa meg a mozgást csak az út részén haladunk le. Ismételje meg az ajánlott ismétlés mennyiségéhez.

Elsődleges izom ■ **Másodlagos izmok** ■



Itt láthatóak egy gyakorlat részletes adatai.

A vissza gombra kattintva, visszakerülünk a lista oldalra.

- Ha volt szűrőnk beállítva, akkor ismét a megszűrt lista lesz látható számunkra.

A rekordok gombra kattintva átkerülünk a rekordok oldalra.

Az edzés előzmények, gombra kattintva, átkerülünk az edzések oldalra, egy speciális szűrővel, így csak azok az edzéseink lesznek láthatóak, amelyekben szerepel az adott gyakorlat.

AZ ALÁBBI EDZÉSEK TARTALMAZZÁK A KERESETT GYAKORLATOT: ADDUKTOR

Dátum alapján csökkenő

Lábedzés: 2025/03/30 ☆

Adduktor 0 szett
Hasgörgő 0 szett
Haladó Kettlebell Szélmalom 0 szett
további 2...

EDZÉS MEGTEKINTÉSE

Melledzés: 2024/02/27 ☆

Air Bike

edzés előzmények

i A címből, lehet megkülönböztetni a sima edzések oldaltól

A diagramok oldal átvisz a statisztika oldalra.

Eszköz: saját testsúlyos

Elsődleges izom: Hasizom

Másodlagos izmok: Hasizom

Gyakorlat leírása :
Feküdjön le a padlón, és rögzítse a lábat. A lábat térdre kell hajlítani. Helyezze a kezét a feje mögé vagy oldalára. Kezdeje hátradal a földön. Ez lesz a kiindulási pozíció. Hajlítsa meg a csípőjét és a gerincét, hogy a törzset a térd felé emelje. A összehúzódás tetején a törzsnek merőlegesnek kell lennie a talajra. Fordítsa meg a mozgást csak az út ? részén haladunk le. Ismételje meg az ajánlott ismétlés mennyiségehez.

Elsődleges izom ■ Másodlagos izmok ■

A gyakorlat részletei

itt Látható

az eszköz,

az elsődleges izom,

a másodlagos izmok,

a gyakorlat teljes leírása,
jobb oldalt a figura, beszinezett izmokkal.

 Itt is megjelenik a szerkesztés ikon, persze kizárálag Adminisztrátoroknak.

3.7 Profil

Bevezetés

Ebben a fejezetben, megnézzük, mit tudunk tenni a profil oldalon

3.7.1 Profil Oldal

The screenshot shows a dark-themed web application interface. At the top center is the title 'ADMIN ADATAI'. Below it are three navigation tabs: 'ADMIN FELÜLET' (highlighted in blue), 'KEDVENC EDZÉSEK' (with a star icon), and 'EDZÉSTERVEZŐ' (with a pencil icon). The main content area is divided into two sections. On the left, under 'Adatok', there is a table with columns 'Testsúly', 'Magasság', and 'BMI', showing values 80kg, 180cm, and 24.69 respectively. Below this is a form titled 'Adatok módosítása' with input fields for 'Testsúly' (80) and 'Magasság' (180), and a 'MENTÉS' (Save) button. On the right, under 'Testtömeg index (BMI)', there is a large green '24.69' and a table of BMI categories:

Súlyos soványsság	<16
Mérsékelt soványsság	16,0 - 16,99
Enyhe soványsság	17,0 - 18,49
Normális testsúly	18,5 - 24,99
Túlsúly	25,0 - 29,99
I. fokú elhízás	30,0 - 34,99
II. fokú elhízás	35,0 - 39,99
III. fokú elhízás	>40

Az oldalon a nevünk alatt 2 gomb látható (3 ha adminisztrátorok vagyunk)

A kedvenc edzések, ami átnavigál, egy külön oldalra, csak a kedvenc edzéseknek fenntartva

Az edzéstervező ami átnavigál az edzéstervező oldalra,

Ha adminok vagyunk akkor az Admin felület ami átnavigál az admin oldalra

Alább láthatóak az adataink.

Alatta tudjuk módosítani az adatainkat

- testsúly: kilogrammban értendő, ide egy számot kell beírnunk, aminek minimum 20-nak kell lennie és maximum 635 lehet.
- Magasság: centiméterben értendő, ide is egy számot kell beírnunk ami minimum 50-kell hogy legyen, és maximum 255 lehet.

3.7.2 Kedvenc edzések

EDZÉSEK

Lábedzés:

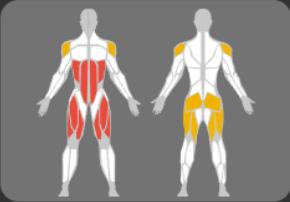
Hasgörgő 0 szett

Haladó Kettlebell Szélmalom 0 szett

Négykézláb Combnyújtás 0 szett

további 1...

2025/03/30 ★



EDZÉS MEGTEKINTÉSE

Válledzés:

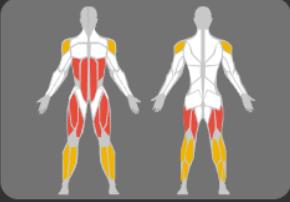
Hasgörgő 5 szett

Air Bike 4 szett

90/90 Combhajlító 3 szett

további 1...

2024/03/04 ★



Kedvenc edzések

Nagyon hasonló az edzések oldalhoz, itt nincs sorrend állító, és csak a kedvenc edzéseinket jeleníti meg az oldalon, de ugyanúgy néznek ki a kártyák és ugyanúgy egyszerre három edzés jelenik meg, ha rákattintunk, a csillagra akkor innen eltűnik az edzés, és csak az edzések közül, tudjuk újra kedvenccé tenni.

3.8 Admin

Bevezetés

Itt az adminisztrátorok, plussz jogusulságait, lehet megtekinteni, amik itt le vannak írva, az egy mezei felhasználó, nem tudja megtenni, és ezeket az oldalakat nem is éri el.

3.8.1 Gyakorlat Szerkesztése Oldal

GYAKORLAT SZERKESZTÉSE

Gyakorlat neve*
3/4 Felülés

Fő izomcsoport*
Hasizom

Izomcsoportok*
Hasizom x

Eszköz
saját testsúlyos

Gyakorlat leírása
Feküdjön le a padlóra, és rögzítse a lábat. A lábat térdre kell hajlítani. Helyezze a kezét a feje mögé vagy oldalára. Kezdeje háadal a földön. Ez lesz a kiindulási pozíciód. Hajlitsa meg a csípjét és a gerincét, hogy a törzs a térd felé emelje. A összehúzódás tetején a törzsnek merőlegesnek kell lennie a talajra. Forditsa meg a mozgást csak az út ? részén haladunk le. Ismételje meg az ajánlott ismétlés mennyiségezhet.

TÖRLÉS MÉGSE MENTÉS

A szerkesztés ikonra kattintva, ez a látvány tárul elénk.

⚠️ Figyeljünk mert a nagy erő nagy felelősséggel jár, hiszen az itt tett változtatásaink minden felhasználó számára láthatóak, és ha törlünk egy gyakorlatot, az minden felhasználó, minden edzéséből törlődik.

- Gyakorlat neve: megváltoztathatjuk a gyakorlat nevét, megadása kötelező.
- Fő izomcsoport: legördülő listából kiválaszthatjuk, a fő izomcsoportot, megadása kötelező.
- Izomcsoportok: legördülő listából kiválaszthatjuk, az izomcsoportokat, legalább egy kötelező. A kis x-re kattintva kitörölhetjük, a hozzáadott izomcsoportokat.
- Eszköz: beírhatjuk az eszközt, nem kötelező.
- Gyakorlat leírása: beírhatjuk a leírást, nem kötelező

A törlés gombra kattintva felugrik egy megerősítő ablak, ha nemre kattintunk bezárul, ha igenre kattintunk visszakerülünk, a gyakorlatok oldalra, kitörlődik a gyakorlat és üzenetet kapunk a sikeres törlésről.

A mégse gombra kattintva visszakerülünk a gyakorlat részleteinek az oldalára, és a változtatásaink törlésre kerülnek.

A mentés gombra kattintva, a változtatásaink elmentődnek, visszakerülünk a gyakorlat részleteinek az oldalára, és üzenetet kapunk, hogy az adott gyakorlat sikeresen megváltozott.

3.8.2 Új gyakorlat hozzáadása

ÚJ GYAKORLAT LÉTREHOZÁSA

Gyakorlat neve*

Fő izomcsoport*

Izomcsoportok*

Eszköz

Gyakorlat leírása

MÉGSE LÉTREHOZÁS

Új gyakorlat form

Itt lehet új gyakorlatokat létrehozni.

Gyakorlat neve(kötelező): be kell írni, hogy mi lesz a gyakorlat neve.

Fő izomcsoport(kötelező): ide egy izomcsoportot kell megadni, (többet nem lehet).

Izomcsoportok(kötelező): Ide több izomcsoportot is meg lehet adni.

Eszköz(nem kötelező):ide meg lehet adni, hogy milyen eszközzel, történik a gyakorlat.

Gyakorlat leírása(nem kötelező): Ide részletes leírást lehet írni, a gyakorlat végrehajtásáról

Mégse gombra kattintva, visszakerülünk az előző oldalra.

Létrehozás gombra kattintva, visszakerülünk az előző oldalra, és üzenetet kapunk, a sikeres létrehozásról.

3.8.3 Admin felület

ADMIN FELÜLET

Filterek

Felhasználónév:

Email:

Admin jog:

KERESÉS 🔍

admin@gmail.com	Admin	Saját felhasználó
user@gmail.com	BasicUser	ADMIN HOZZÁFÉRÉS +

admin felület

Az admin felületen, láthatjuk az összes felhasználót, a felső szűrővel tudunk keresni a felhasználók között.

- felhasználónév: Ha a beírt szöveget tartalmazza bármelyik létező felhasználó, akkor kiírja azokat
- Email: Ha a beírt szöveget tartalmazza bármelyik létező felhasználó, akkor kiírja azokat
- Admin jog: kiválasztható, hogy admin vagy felhasználó, és amelyiket választjuk, olyan tipusú fiókokat jelenít meg.

admin@gmail.com	Admin	Saját felhasználó
user@gmail.com	BasicUser	ADMIN HOZZÁFÉRÉS +
pepesdenes@gmail.com	PepesDenes	ADMIN HOZZÁFÉRÉS -

felhasználó kártyák

A kártyák egyszerűen épülnek fel, bal oldalt az email cím

középen a felhasználónév

jobb oldalt pedig az admin jog megváltoztatása. Értelemszerűen saját magunktól, nem tudjuk elvenni a jogot.

Másoknak viszont tudunk adni, és el is tudjuk tőlük venni, az "admin hozzáférés" gombra kattintva, megjelenik egy megerősítő ablak, hogy biztos szeretnénk-e módosítani az adott felhasználó jogát, igenre kattintva ez megtörténik, és üzenetet kapunk a változásról, nemre kattintva bezárul az ablak.

A jobb oldalt lévő kis kukával törölhetjük a felhasználót, itt is megjelenik a megerősítő ablak, igenre kattintva, kitörlődik a felhasználó és üzenetet kapunk a törlésről.

3.9 Statisztikák

Bevezetés

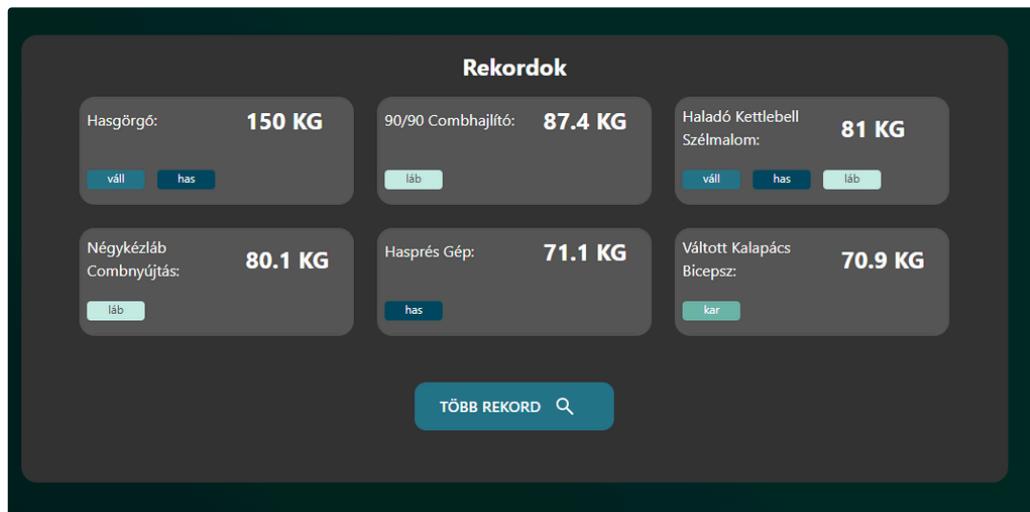
Ebben a fejezetben a különböző kimutatásokról, tudhatunk meg többet.

3.9.1 Statisztikák oldal



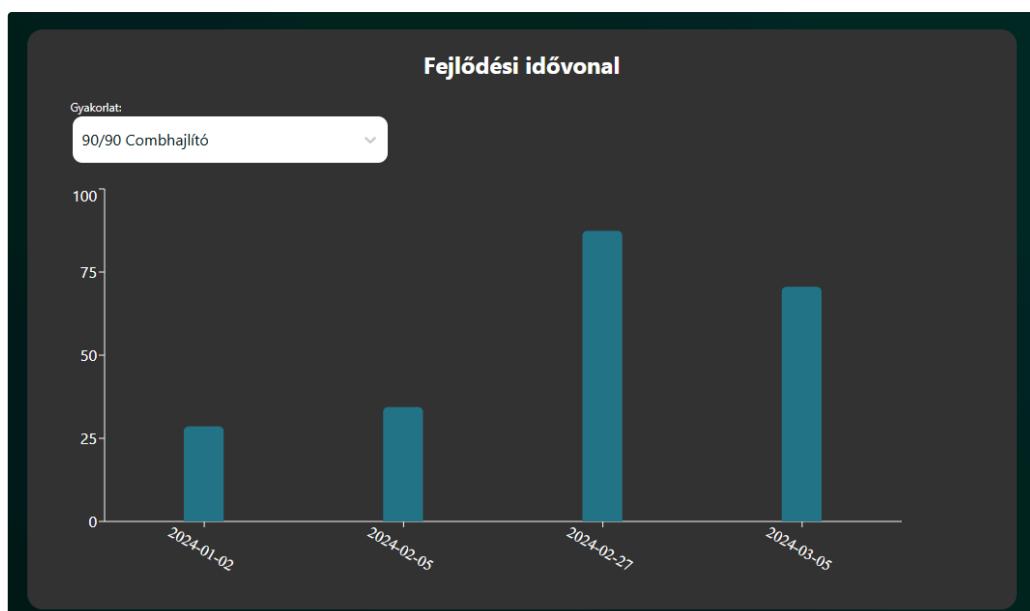
Statisztikák oldal

Itt látható a statisztikák oldal



Rekordok modul

Itt látható néhány rekord. A "több rekord" gombra kattintva átnavigál a rekordok oldalra.



Oszlopdiagraam

A fejlődési idővonal, egy oszlopdiagramot jelenít meg.

A legördülő menüből, csak azok közül a gyakorlatok közül választhatunk, amelyek már voltak az edzéseinkben, az oszlopdiagram, maximum a legutóbbi 10 edzésünket jeleníti meg, ahogy a képen látszik ha kevesebb mint 10 edzésünkben szerepel ez a gyakorlat akkor kevesebb edzést, jelenít meg.

Minden oszlop, az adott edzésben, elért legnagyobb súlyt írja ki.



Itt, az összes megmozgatott súly, és az izomcsoportok százalékos eloszlása látható. Az időtartamnál 4 lehetőség közül választhatunk

- hét (7 napra visszamenően írja ki az edzések összesítését)
- hónap (30 napra visszamenően írja ki az edzések összesítését)
- félév (180 napra visszamenően írja ki az edzések összesítését)
- összes (Az összes edzést összesíti)

3.9.2 Rekordok Oldal

The screenshot shows a search interface for exercises. At the top, there is a search bar labeled 'Gyakorlat neve:' (Exercise name) and a teal 'KERESÉS' (Search) button with a magnifying glass icon. Below the search bar, the results are displayed in a grid format:

Gyakorlat neve:	Hasgörgő:	90/90 Combhajlító:	Haladó Kettlebell Szélmalom:
	150 KG váll has láb	87.4 KG láb	81 KG váll has láb
Négykézláb Combnyújtás:	80.1 KG láb	Hasprés Gép:	71.1 KG has
		Váltott Kalapács Bicepsz:	70.9 KG kar
Air Bike:	67.7 KG has	Adduktor/Lágyék:	62.7 KG láb

At the bottom center of the page, there is a navigation indicator showing '1' between two arrows.

A rekordok oldalon, láthatjuk azokat a gyakorlatainkat, amelyeket valaha végrehajtottunk, és hogy maximum, mekkora súlytal csináltuk ezt, a gyakorlat neve szűrővel kereshetünk adott gyakorlatokra.

Az oldal egyszerre 9 gyakorlatot jelenít meg.



4. Teszt Dokumentáció

Bevezetés

A Teszt esetek egy NestJS alapú alkalmazás részei, amelyek az aplikáció adatainak kezeléséért felelnek. A **Service** réteg biztosítja az adatok lekérdezését, míg a **Controller** a HTTP kéréseket kezeli. A tesztek célja, hogy minden komponens megfelelően működjön és helyes választ küldjön a különböző kérésekre.

Tesztkörnyezet

A tesztek **Jest** és **@nestjs/testing** segítségével készültek. A **PrismaService** és az **AuthGuard** működését mock objektumokkal szimuláljuk, hogy izoláltan tudjuk tesztelni az üzleti logikát és a vezérlő működését.

4.1 Edzés Teszt esetek

Futtatás:

```
npm run test edzes.service.spec.ts
```

A tesztek **Jest** és **@nestjs/testing** keretrendszerben lettek írva.. A **PrismaService** adatbáziskezelést mock objektumokkal utánozzuk (**jest-mock-extended** segítségével).

Mock Adatok

A tesztekben a következő mock adatok szerepelnek:

- **mockUser**: Egy felhasználó adatstruktúrája.
- **mockEdzes**: Egy edzés adatstruktúrája.
- **mockGyakorlat**: Egy gyakorlat adatstruktúrája.
- **mockEdzesGyakorlat**: Egy edzéshez hozzárendelt gyakorlat adatstruktúrája.
- **mockPreviousHistory**: Egy felhasználó előző gyakorlatai.

Tesztesetek

1. Edzés létrehozása

- **Leírás**: Egy edzés sikeres létrehozásának ellenőrzése.
- **Mock beállítások**:
 - **user.findUnique**: A felhasználó létezik.
 - **edzes.findMany**: Az adott nevű edzés még nem létezik.
 - **edzes.create**: Az edzés sikeresen létrejön.
- **Elvárt eredmény**: Az edzés adatai megfelelnek a várt értékeknek.

2. Egy edzés lekérdezése

- **Leírás**: Egy edzés lekérdezésekor annak gyakorlatai is szerepeljenek.
- **Mock beállítások**:
 - **edzes.findUnique**: Az edzés létezik.
 - **user_Gyakorlat_History.findFirst**: A felhasználónak van előzménye.
 - **user_Gyakorlat_History.findMany**: Az előzmények listája visszatér.
- **Elvárt eredmény**: Az edzés adatstruktúrája tartalmazza a gyakorlati listát.

3. Gyakorlat hozzáadása egy edzéshez

- **Leírás**: Ellenőrzi, hogy egy edzéshez gyakorlat hozzáadható-e.
- **Tesztesetek**:
 - Edzés nem létezik → NotFoundException** dobása.

b. Gyakorlat már hozzá van adva → **ConflictException** dobása.

4. Gyakorlat eltávolítása egy edzésből

- **Leírás:** Ellenőrzi, hogy egy gyakorlat sikeresen törölhető-e egy edzésből.
- **Mock beállítások:**
 - **edzes.findUnique:** Az edzés létezik.
 - **edzes_Gyakorlat.findUnique:** A gyakorlat szerepel az edzésben.
 - **\$transaction:** Az adatbázis tranzakció sikeresen lefut.
- **Elvárt eredmény:** Sikeres törlés esetén egy "Gyakorlat sikeresen törölve az edzésből" üc.

5. Gyakorlat törlés sikertelen esetek

- **Edzés nem létezik** → **BadRequestException** dobása.
- **Gyakorlat nincs az edzésben** → **BadRequestException** dobása.

```
> nestjs-keret-2024@0.0.1 test
> jest --config jest.config.js edzes.service.spec.ts

PASS  src/edzes/edzes.service.spec.ts
  EdzesService
    create
      ✓ should create an edzes successfully (14 ms)
    findOne
      ✓ should include gyakorlatok in the response (3 ms)
    addGyakorlatToEdzes
      ✓ should throw NotFoundException when edzes not found (28 ms)
      ✓ should throw ConflictException when gyakorlat already exists in edzes (5 ms)
    deleteGyakorlatFromEdzes
      ✓ should delete gyakorlat from edzes successfully (3 ms)
      ✓ should throw BadRequestException when edzes not found (3 ms)
      ✓ should throw BadRequestException when gyakorlat not found in edzes (1 ms)

Test Suites: 1 passed, 1 total
Tests:       7 passed, 7 total
Snapshots:  0 total
Time:        1.282 s, estimated 4 s
```

4.2 IzomcsoportService és IzomcsoportController

Futtatás:

```
npm run test gyakorlat.controller.spec.ts
```

```
> nestjs-keret-2024@0.0.1 test
> jest --config jest.config.js izomcsoport.service.spec.ts

PASS  src/izomcsoport/izomcsoport.service.spec.ts
  IzomcsoportService
    ✓ should be defined (10 ms)
    findAll
      ✓ should return an array of izomcsoportok (3 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        1.102 s, estimated 4 s
Ran all test suites matching /izomcsoport.service.spec.ts/i.
```

```
npm run test gyakorlat.service.spec.ts
```

```
> nestjs-keret-2024@0.0.1 test
> jest --config jest.config.js izomcsoport.controller.spec.ts

PASS  src/izomcsoport/izomcsoport.controller.spec.ts
  IzomcsoportController
    ✓ should be defined (11 ms)
    findAll
      ✓ should return an array of izomcsoportok (3 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        1.419 s, estimated 5 s
Ran all test suites matching /izomcsoport.controller.spec.ts/i.
```

Mock Adatok

A tesztekhez az alábbi mock adatok kerültek felhasználásra:

- **mockIzomcsoportok**: Az adatbázisból lekérhető izomcsoportok listája, amely több tesztadatot tartalmaz.
- **mockAuthGuard**: Egy egyszerűsített autentikációs őr, amely lehetővé teszi a vezérlő metódusainak tesztelését.

Teszesetek

1. Szerviz inicializálása

- **Leírás**: Ellenőrzi, hogy az **IzomcsoportService** megfelelően inicializálható-e.
- **Elvárt eredmény**: A szerviz példánya megfelelően létezik és használható.

2. Izomcsoportok lekérdezése a szerviz rétegen

- **Leírás**: Ellenőrzi, hogy a szerviz vissza tudja adni az összes elérhető izomcsoportot.

- **Mock beállítások:**
 - **izomcsoport.findMany**: Egy előre definiált tömböt ad vissza az adatbázisból.
- **Elvárt eredmény**: Az eredmény tartalmazza az összes várt izomcsoportot a megfelelő struktúrával.

3. Vezérlő inicializálása ↗

- **Leírás**: Ellenőrzi, hogy az **IzomcsoportController** megfelelően inicializálható-e.
- **Elvárt eredmény**: A vezérlő példánya megfelelően létezik és használható.

4. Izomcsoportok lekérdezése a vezérlő rétegen keresztül ↗

- **Leírás**: Ellenőrzi, hogy az API végpont megfelelően visszaadja az izomcsoportok listáját.
- **Mock beállítások**
 - **izomcsoportService.findAll**: Egy előre definiált tömböt ad vissza.
- **Elvárt eredmény**: Az API válasza tartalmazza az összes elérhető izomcsoportot.

4.3 Felhasználó Teszt esetek - UserService és UsersController

Bevezetés

A **UserService** és **UsersController** felelős a felhasználók kezeléséért az API végpontjain keresztül. A tesztek biztosítják, hogy minden szolgáltatás helyesen működik.

Futtatás:

```
npm run test users.service.spec.ts
```

```
> nestjs-keret-2024@0.0.1 test
> jest --config jest.config.js users.service.spec.ts

PASS  src/users/users.service.spec.ts
  UserService
    ✓ should be defined (10 ms)
    create
      ✓ should create a user successfully (3 ms)
      ✓ should throw ConflictException if username exists (12 ms)
    findOne
      ✓ should return a user (2 ms)
      ✓ should throw NotFoundException if user not found (4 ms)
    findAll
      ✓ should return all users for admin (2 ms)
      ✓ should return filtered users for non-admin (2 ms)
    update
      ✓ should update user data (2 ms)
      ✓ should throw NotFoundException when updating non-existent user (2 ms)
    remove
      ✓ should delete a user (2 ms)
      ✓ should throw NotFoundException when deleting non-existent user (1 ms)

Test Suites: 1 passed, 1 total
Tests:       11 passed, 11 total
S_snapshots: 0 total
```

```
npm run test users.controller.spec.ts
```

```
> nestjs-keret-2024@0.0.1 test
> jest --config jest.config.js users.controller.spec.ts

PASS  src/users/users.controller.spec.ts
  UsersController
    ✓ should be defined (10 ms)
    findOne
      ✓ should return a user (4 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
S_snapshots: 0 total
Time:       1.568 s, estimated 5 s
Run all test suites matching "/users/controller.spec.ts/i"
```

Teszkörnyezet

A tesztek **Jest** és **@nestjs/testing** segítségével készültek. A következő komponensek kerültek mockolásra:

- **UsersService**: A felhasználókkal kapcsolatos üzleti logika kezelése.
 - **UsersController**: Az API végpontokat kezelő vezérlő.
 - **PrismaService**: Az adatbázis műveletek szimulálása.
 - **JwtService**: A JWT tokenek generálása és ellenőrzése.
 - **AuthGuard**: Az autentikációs és jogosultságkezelési logika mockolása.
-

Mock Adatok

A tesztekben az alábbi mock adatok szerepelnek:

- **mockUser**: Egy alapértelmezett felhasználó adatstruktúrája.
 - **mockAdminUser**: Egy admin jogokkal rendelkező felhasználó.
 - **mockUserService**: A **UsersService** metódusai mockolva vannak, hogy a vezérlő megfelelően hívja meg azokat.
-

Teszesetek

1. Szerviz inicializálása

- **Leírás**: Ellenőrzi, hogy a **UsersService** megfelelően inicializálható-e.
 - **Elvárt eredmény**: A szerviz példánya megfelelően létezik és használható.
-

2. Felhasználó létrehozása

- **Leírás**: Ellenőrzi, hogy a szerviz képes-e új felhasználót létrehozni.
 - **Mock beállítások**:
 - **user.findUnique**: Ellenőrzi, hogy a felhasználó már létezik-e.
 - **bcrypt.hash**: A jelszó hash-elése.
 - **user.create**: A felhasználó adatainak mentése.
 - **Elvárt eredmény**: A létrehozott felhasználó adatainak megfelelő visszaadása.
 - **Hibakezelés**:
 - Ha a felhasználónév már létezik → **ConflictException** dobása.
-

3. Felhasználó lekérdezése

- **Leírás**: Ellenőrzi, hogy egy adott felhasználó lekérhető-e az ID alapján.
 - **Mock beállítások**:
 - **user.findUnique**: Visszaadja a felhasználó adatait.
 - **Elvárt eredmény**: A felhasználó adatai sikeresen visszatérnek.
 - **Hibakezelés**:
 - Ha a felhasználó nem létezik → **NotFoundException** dobása.
-

4. Összes felhasználó lekérdezése

- **Leírás**: Ellenőrzi, hogy a szerviz vissza tudja adni az összes elérhető felhasználót.
- **Mock beállítások**:
 - **user.findMany**: Egy előre definiált felhasználói listát ad vissza.

- **Elvárt eredmény:** Az összes várt felhasználó a megfelelő struktúrával.
-

5. Felhasználó adatainak frissítése

- **Leírás:** Ellenőrzi, hogy egy felhasználó adatai sikeresen frissíthetők-e.
 - **Mock beállítások:**
 - **user.findUnique:** Ellenőrzi, hogy a felhasználó létezik-e.
 - **user.update:** A módosított adatok mentése.
 - **Elvárt eredmény:** A frissített felhasználó adatai megfelelően visszatérnek.
 - **Hibakezelés:**
 - Ha a felhasználó nem létezik → **NotFoundException** dobása.
-

6. Felhasználó törlése

- **Leírás:** Ellenőrzi, hogy egy felhasználó sikeresen törölhető-e.
 - **Mock beállítások:**
 - **user.findUnique:** Ellenőrzi, hogy a felhasználó létezik-e.
 - **user.delete:** A törlési folyamat végrehajtása.
 - **Elvárt eredmény:** A törölt felhasználó adatainak visszatérése.
 - **Hibakezelés:**
 - Ha a felhasználó nem létezik → **NotFoundException** dobása.
-

Vezérlő Tesztesetek (UsersController)

7. UsersController inicializálása

- **Leírás:** Ellenőrzi, hogy a **UsersController** megfelelően inicializálható-e.
 - **Elvárt eredmény:** A vezérlő példánya megfelelően létezik és használható.
-

8. Egy felhasználó lekérdezése

- **Leírás:** Ellenőrzi, hogy a vezérlő megfelelően hívja meg a **findOne** metódust a szerviz rétegben.
 - **Mock beállítások:**
 - **findOne:** A mockUser adat visszaadása.
 - **Elvárt eredmény:** A megfelelő felhasználó adatainak visszaadása.
-

9. Összes felhasználó lekérdezése

- **Leírás:** Ellenőrzi, hogy a vezérlő vissza tudja adni az összes felhasználót.
 - **Mock beállítások:**
 - **findAll:** Egy tömb visszaadása több felhasználóval.
 - **Elvárt eredmény:** Az összes elérhető felhasználó megfelelő visszaadása.
-

10. Új felhasználó létrehozása

- **Leírás:** Ellenőrzi, hogy a vezérlő képes-e új felhasználót létrehozni.

- **Mock beállítások:**
 - **create:** A mockUser visszaadása a létrehozási metódusból.
 - **Elvárt eredmény:** A létrehozott felhasználó adatainak visszaadása.
-

11. Felhasználó adatainak frissítése

- **Leírás:** Ellenőrzi, hogy a vezérlő megfelelően meghívja az update metódust.
 - **Mock beállítások:**
 - **update:** A frissített felhasználó adatainak visszaadása.
 - **Elvárt eredmény:** A módosított felhasználói adatok visszaadása.
-

12. Felhasználó törlése

- **Leírás:** Ellenőrzi, hogy a vezérlő megfelelően meghívja a remove metódust.
- **Mock beállítások:**
 - **remove:** Egy sikeres törlési válasz visszaadása.
- **Elvárt eredmény:** A törölt felhasználó adatai visszatérnek.

4.4 UserGyakorlatController és UserGyakorlatService Teszt Dokumentáció

1. Bevezetés [🔗](#)

A dokumentum bemutatja a `UserGyakorlatController` és `UserGyakorlatService` tesztelési stratégiáját. A cél az, hogy ellenőrizzük a funkcionális helyes működését és az esetleges hibák kezelését.

2. Tesztkörnyezet [🔗](#)

A tesztkörnyezet a következő eszközöket használja:

- **Jest**: Tesztelési keretrendszer
- **NestJSTestingModule**: A modulok és szolgáltatások izolált teszteléséhez
- **Mockok**: Függőségek helyettesítése

3. UserGyakorlatController Tesztelése [🔗](#)

3.0. Futtatás [🔗](#)

```
npm run test user-gyakorlat.controller.spec.ts
```

```
> nestjs-keret-2024@0.0.1 test
> jest --config jest.config.js user-gyakorlat.controller.spec.ts

PASS  src/user-gyakorlat/user-gyakorlat.controller.spec.ts
UserGyakorlatController
  ✓ should be defined (10 ms)
    create
      ✓ should create a user gyakorlat (3 ms)
    createHistory
      ✓ should create a history entry (2 ms)
    getUserGyakorlatok
      ✓ should return user gyakorlatok (2 ms)
    getUserGyakorlatDetails
      ✓ should return user gyakorlat details (3 ms)
    deleteUserGyakorlat
      ✓ should delete a user gyakorlat (4 ms)

Test Suites: 1 passed, 1 total
Tests:       6 passed, 6 total
Snapshots:  0 total
Time:        1.436 s, estimated 2 s
All tests passed!
```

3.1. Példányosítás [🔗](#)

A tesztelés előtt a `UserGyakorlatController` példányosítása történik mockolt szolgáltatásokkal.

3.2. Tesztesetek [🔗](#)

3.2.1. create [🔗](#)

Cél: Ellenőrizni, hogy a `createUserGyakorlat` metódus megfelelően létrehozza a felhasználói gyakorlatot.

- Bemenet: `CreateUserGyakorlatDto`
- Kimenet: Létrehozott objektum

- Ellenőrzések:
 - A metódus meghívása a megfelelő paraméterekkel
 - A visszatérési érték helyessége

3.2.2. `createHistory` ↗

Cél: Ellenőrizni, hogy a történeti bejegyzés létrehozása megfelelően működik.

- Bemenet: `CreateUserGyakorlatHistoryDto`
- Kimenet: Létrehozott bejegyzés
- Ellenőrzések:
 - A megfelelő paraméterekkel történő hívás
 - A visszatérési érték tartalma

3.2.3. `getUserGyakorlatok` ↗

Cél: Ellenőrizni, hogy a gyakorlatok listázása működik.

- Bemenet: `GetUserGyakorlatokQueryDto`
- Kimenet: Gyakorlatok listája és metaadatok
- Ellenőrzések:
 - A szolgáltatás megfelelő meghívása
 - A visszatérési érték formátuma

3.2.4. `getUserGyakorlatDetails` ↗

Cél: Egy adott gyakorlat részleteinek lekérése.

- Ellenőrzés:
 - Létező és nem létező gyakorlat esetén
 - Hibakezelés

3.2.5. `deleteUserGyakorlat` ↗

Cél: Egy felhasználói gyakorlat törlésének ellenőrzése.

- Ellenőrzések:
 - Sikeres törlés
 - Nem létező gyakorlat esetén hiba dobása

4. UserGyakorlatService Tesztelése ↗

4.0. Futtatás ↗

```
npm run test user-gyakorlat.controller.spec.ts
```

```

> nestjs-keret-2024@0.0.1 test
> jest --config jest.config.js user-gyakorlat.service.spec.ts

  PASS  src/user-gyakorlat/user-gyakorlat.service.spec.ts
    UserGyakorlatService
      ✓ should be defined (10 ms)
      createUserGyakorlat
        ✓ should create a user gyakorlat successfully (3 ms)
        ✓ should throw BadRequestException if user gyakorlat already exists (11 ms)
        ✓ should throw NotFoundException if user or gyakorlat not found (2 ms)
      createUserGyakorlatHistory
        ✓ should create a history entry successfully (4 ms)
        ✓ should create user gyakorlat if it does not exist (2 ms)
      getUserGyakorlatok
        ✓ should return user gyakorlatok with pagination (2 ms)
        ✓ should return user gyakorlatok with search filter (1 ms)
      getUserGyakorlatDetails
        ✓ should return user gyakorlat details (2 ms)
        ✓ should throw NotFoundException if user gyakorlat not found (1 ms)
      deleteUserGyakorlat
        ✓ should delete user gyakorlat and related records (4 ms)
        ✓ should throw NotFoundException if user gyakorlat not found (1 ms)

      Test Suites: 1 passed, 1 total
      Tests:       12 passed, 12 total
      Snapshots:  0 total
      Time:        1.208 s, estimated 2 s
      Ran all test suites matching /user-gyakorlat.service.spec.ts/i

```

4.1. Példányosítás

A szolgáltatás PrismaService mockolt verziójával kerül tesztelésre.

4.2. Tesztesetek

4.2.1. createUserGyakorlat

Cél: Egy új felhasználói gyakorlat létrehozásának ellenőrzése.

- Ellenőrzések:
 - Létező gyakorlat esetén BadRequestException
 - Hiányzó gyakorlat vagy felhasználó esetén NotFoundException
 - Sikeres létrehozás

4.2.2. createUserGyakorlatHistory

Cél: Egy új történeti bejegyzés létrehozása.

- Ellenőrzések:
 - Létező gyakorlat és edzés esetén sikeres mentés
 - Ha a gyakorlat nem létezik, automatikusan létrejön

4.2.3. getUserGyakorlatok

Cél: A gyakorlatok lekérése lapozással és keresési szűrővel.

- Ellenőrzések:
 - Az adatok helyes visszaadása
 - Keresési feltételek működése

4.2.4. getUserGyakorlatDetails

Cél: Egy konkrét gyakorlat részleteinek lekérése.

- Ellenőrzések:
 - Létező és nem létező rekord esetén

- Hibakezelés

4.2.5. **deleteUserGyakorlat** ↗

Cél: Egy felhasználói gyakorlat törlésének ellenőrzése.

- Ellenőrzések:
 - Sikeres törlés
 - Kapcsolódó adatok törlése
 - Nem létező rekord esetén `NotFoundException`

4.5 GyakorlatController és GyakorlatService Teszt Dokumentáció

1. Bevezetés

A dokumentáció célja a **GyakorlatController** és **GyakorlatService** modulok tesztelésének bemutatása. A tesztek a NestJS framework segítségével készültek.

2. Tesztelési környezet

- **Framework:** NestJS
- **Tesztelési könyvtár:** Jest
- **Mockok:** Jest.fn(), fake PrismaService
- **Adatforrás:** Prisma ORM (mockolva)

3. GyakorlatController Tesztjei

3.0. Futtatás:

```
npm run test gyakorlat.controller.spec.ts
```

3.1. Létrehozás (create)

Leírás: Egy új gyakorlat hozzáadása.

- **Sikeres létrehozás:** A függvény sikeresen elmenti az adatbázisba és visszaadja a létrehozott objektumot.
- **Érvénytelen adat:** Ha az adatok hiányosak, BadRequestException keletkezik.

3.2. Lekérdezés (findAll)

Leírás: Gyakorlatok listázása szűrőkkel és oldalszámítással.

- **Összes gyakorlat lekérése:** A függvény visszaadja az összes gyakorlatot paginációval.
- **Szűrés név alapján:** Ha van szűrés, csak az adott névvel rendelkező gyakorlatokat adja vissza.

3.3. Egyetlen gyakorlat lekérdezése (findOne)

Leírás: Egy adott ID alapján keres gyakorlatot.

- **Létező ID:** A megfelelő gyakorlat visszatérése.
- **Nem létező ID:** NotFoundException kivétel dobása.

3.4. Módosítás (update)

Leírás: Meglévő gyakorlat módosítása.

- **Sikeres frissítés:** A módosított adatok visszatérése.
- **Nem létező ID:** NotFoundException kivétel dobása.

3.5. Törlés (remove)

Leírás: Gyakorlat törlése ID alapján.

- **Létező ID:** A törlés végrehajtása és sikeres válasz.
- **Nem létező ID:** NotFoundException kivétel dobása.

```
> nestjs-keret-2024@0.0.1 test
> jest --config jest.config.js gyakorlat.controller.spec.ts

PASS  src/user-gyakorlat/user-gyakorlat.controller.spec.ts
PASS  src/gyakorlat/gyakorlat.controller.spec.ts

Test Suites: 2 passed, 2 total
Tests:       16 passed, 16 total
Snapshots:   0 total
Time:        2.091 s, estimated 5 s
Ran all test suites matching /gyakorlat_controller_spec_ts/i
```

4. GyakorlatService Tesztjei

4.0. Futtatás:

```
npm run test gyakorlat.service.spec.ts
```

4.1. findAll

Leírás: Összes gyakorlat listázása Prisma ORM használatával.

- **Sikeres lekérdezés:** Az összes gyakorlat visszatérése paginációval.
- **Üres adatbázis:** Üres lista visszatérése.

4.2. findOne

Leírás: Egy adott gyakorlat lekérése ID alapján.

- **Létező gyakorlat:** Az adott ID-jú gyakorlat visszaadása.
- **Nem létező ID:** NotFoundException dobása.

4.3. create

Leírás: Új gyakorlat hozzáadása az adatbázishoz.

- **Sikeres létrehozás:** A létrehozott gyakorlat visszatérése.
- **Érvénytelen adatok:** BadRequestException kivétel dobása.

4.4. update

Leírás: Egy meglévő gyakorlat módosítása.

- **Sikeres módosítás:** A frissített gyakorlat visszaadása.
- **Nem létező ID:** NotFoundException kivétel dobása.

4.5. remove

Leírás: Egy adott ID-jú gyakorlat törlése.

- **Létező ID:** Sikeres törlés.
- **Nem létező ID:** NotFoundException kivétel dobása.

```
> nestjs-keret-2024@0.0.1 test
> jest --config jest.config.js gyakorlat.service.spec.ts

PASS  src/gyakorlat/gyakorlat.service.spec.ts
PASS  src/user-gyakorlat/user-gyakorlat.service.spec.ts

Test Suites: 2 passed, 2 total
Tests:       14 passed, 14 total
Snapshots:   0 total
Time:        1.767 s, estimated 4 s
```



4.6 Frontend manuális tesztelés

TC ID	Funkció/Modul	Teszt leírása	Bemenet	Elvárt eredmény	Tényleges eredmény	Teszt státusz
TC001	Regisztráció	üres mezők beküldése		Üres mezők a hiányosságot jelzik	Üres mezők a hiányosságot jelzik	Sikeres ✓
TC002	Regisztráció	Helytelen email formátum megadása	Email cím: usergmail.com	Email megadása kötelező hibaüzenet	Email megadása kötelező hibaüzenet	Sikeres ✓
TC003	Regisztráció	Gyenge jelszó megadása	Jelszó: jelszócska21	A jelszónak tartalmaznia kell legalább egy nagybetű → és speciális karaktert hibaüzenet	A jelszónak tartalmaznia kell legalább egy nagybetű → és speciális karaktert hibaüzenet	Sikeres ✓
TC004	Regisztráció	Jelszó megerősítése mező helytelen kitöltése	Jelszó: Jelszócska21. Jelszó megerősítése: Jelszócska321.	A jelszavaknak egyezniük kell hibaüzenet	A jelszavaknak egyezniük kell hibaüzenet	Sikeres ✓
TC005	Regisztráció	Helyes adatokkal való regisztrálás	Email cím: jani@gmail.com Felhasználónév: János Jelszó: Jelszó123. Jelszó megerősítése: Jelszó123.	Sikeres Regiszráció Toast. Regisztráció megtörténése Átirányítás a bejelentkezés oldalra	Sikeres Regiszráció Toast.	Sikeres ✓
TC006	Regisztráció	Már használt emai címmel való regisztráció	Email cím: jani@gmail.com Felhasználónév: János Jelszó: Jelszó123. Jelszó megerősítése: Jelszó123.	Az email cím már használatban van Toast Nem regisztrál új felhasználót	Az email cím már használatban van Toast Nem regisztrál új felhasználót	Sikeres ✓

TC007	Bejelentkezés	Üres mezőkkel való bejelentkezés		Email megadása kötelező Jelszó megadása kötelező hibaüzenetek	Email megadása kötelező Jelszó megadása kötelező hibaüzenetek	Sikeres ✓
TC008	Bejelentkezés	Helytelen email formátum megadása	Email cím: janigmail.com	Helytelen email formátum hibaüzenet	Helytelen email formátum hibaüzenet	Sikeres ✓
TC009	Bejelentkezés	Hibás adatokkal való bejelentkezés	Email cím: jani@gmail.com Jelszó: HibásJelszó213	Hibás email vagy jelszó! Ellenőrizze a beírt adatokat. hibaüzenet	Hibás email vagy jelszó! Ellenőrizze a beírt adatokat. hibaüzenet	Sikeres ✓
TC009	Bejelentkezés	Helyes adatokkal való bejelentkezés	Email cím: jani@gmail.com Jelszó: Jelszó123.	Sikeres bejelentkezés Átirányítás a dashboard oldalra	Sikeres bejelentkezés Átirányítás a dashboard oldalra	Sikeres ✓
TC010	Kijelentkezés	Menüsor kijelentkezés gombjára nyomva való kijelentkezés	Kijelentkezés gomb majd Igen választása a megerősítő felugró ablakban	Fiók kijelentkezve Bejelentkezés oldalra átdobás	Fiók kijelentkezve Bejelentkezés oldalra átdobás	Sikeres ✓
TC011	Engedélyeztetlen belépés	Kijelentkezett állapotban a weboldal egyik lapjának megnyitása	Url mezőbe /dashboard beírása	Visszairányít a bejelentkezés oldalra	Visszairányít a bejelentkezés oldalra	Sikeres ✓
TC011	Új Edzés kezdése menüsorból	Menüsor plusz gombjára kattintva megnyiló menusávba edzés kezdése gomb megnyomása, és helytelen adatok megadása	Menüsor plusz gomb > edzés kezdése > Edzés neve: ab	Az edzés névének legalább 3 karakter hosszúnak kell lennie hibaüzenet	Az edzés névének legalább 3 karakter hosszúnak kell lennie hibaüzenet	Sikeres ✓
TC012	Új Edzés kezdése menüsorból	Menüsor plusz gombjára kattintva megnyiló menusávba edzés kezdése gomb megnyomása, és helyes	Menüsor plusz gomb > edzés kezdése > Edzés neve: lábazás > Létrehozás	Edzés elkezdése, számláló elindul, edzesek/[edzesID]/szerkeszt oldalra való átirányítás	Edzés elkezdése, számláló elindul, edzesek/[edzesID]/szerkeszt oldalra való átirányítás	Sikeres ✓

		adatok megadása				
TC013	Edzés folytatása menüsorból	Miután elindítottunk egy edzést viszont még nem nyomtunk a vélegesítése gombra, menüsort plusz gombon belüli, edzés folytatása gomb használata	Menüsort plusz gomb > Edzés Folytatása	Visszadob a már elindított edzés oldalára, a számláló továbbra is fut	Visszadob a már elindított edzés oldalára, a számláló továbbra is fut	Sikeres ✓
TC014	Edzés kezdése a menüsorból	Menüsort plusz gombjára kattintva megnyiló menusávba edzés kezdése gomb megnyomása, ha a mai nap már volt edzésünk	Menüsort plusz gomb > edzés kezdése	A mai napi edzés már befejeződött Toast üzenet	A mai napi edzés már befejeződött Toast üzenet	Sikeres ✓
TC015	Edzés kezdése a menüsorból	Ha a mai napra már van beütemezve edzés és az még nem lett befejezve, a menüsori edzés kezdése gombra nyomva az edzés elindítható	Menüsort plusz gomb > edzés kezdése > Igen választása a megerősítő felugró ablakban	A mai napra betervezett edzés elkezdése és átirányítás az szerkesztés oldalára, számláló elindul	A mai napra betervezett edzés elkezdése és átirányítás az szerkesztés oldalára, számláló elindul	Sikeres ✓
TC016	Edzés szerkesztése	Gyakorlat hozzáadása az edzéshez	Gyakorlat Hozzáadása gomb > felugró ablakban egy gyakorlat kiválasztása > Hozzáadás	Gyakorlat hozzáadva az edzéshez	Gyakorlat hozzáadva az edzéshez	Sikeres ✓
TC017	Edzés szerkesztése	Egy gyakorlat törlése az edzésből	Gyakorlat jobb sarkában található X gomb megnyomása >	Gyakorlat törlése az edzésből	Gyakorlat törlése az edzésből	Sikeres ✓

			felugró ablak Igen gomb			
TC018	Edzés szerkesztése	Az edzéshez hozzáadott gyakorlathoz szett hozzáadása	Gyakorlat Szett + gomb megnyomása	Új üres szett hozzáadva	Új üres szett hozzáadva	Sikeres <input checked="" type="checkbox"/>
TC019	Edzés szerkesztése	Az edzéshez hozzáadott gyakorlathoz hozzáadott szett törlése	Szett melletti minusz gomb megnyomása	Szett törölve	Szett törölve	Sikeres <input checked="" type="checkbox"/>
TC020	Edzés szerkesztése	Az edzéshez hozzáadott gyakorlat nevére kattintás átvezet a gyakorlat oldalára	Gyakorlat nevére kattintás	Átirányít a gyakorlat oldalára	Átirányít a gyakorlat oldalára	Sikeres <input checked="" type="checkbox"/>
TC021	Edzés szerkesztése	Edzés véglegesítése hibás szett mezővel	Egy korábban hozzáadott szett ismétlés mező adatának kitörlése és ezután edzés véglegesítés gombra nyomás	Hiba: Ellenőrid az ürlap mezőit toast üzenet	Hiba: Ellenőrid az ürlap mezőit toast üzenet	Sikeres <input checked="" type="checkbox"/>
TC022	Edzés szerkesztése	Edzés nevének hibás szerkesztése üres mezővel	Edzés mező üresen hagyása	Az edzlés nevének megadása kötelező, figyelmeztető üzenet	Az edzlés nevének megadása kötelező, figyelmeztető szöveg	Sikeres <input checked="" type="checkbox"/>
TC023	Edzés szerkesztése	Edzés nevének hibás szerkesztése rossz adattal	Rossz adat megadása Edzés nevéhez: ab	Az edzés nevének legalább 3 karakter hosszúnak kell lennie, figyelmeztető üzenet	Az edzés nevének legalább 3 karakter hosszúnak kell lennie, figyelmeztető üzenet	Sikeres <input checked="" type="checkbox"/>
TC024	Edzés szerkesztése	Edzés véglegesítése hibás edzés neve mezővel	Edzés neve: ab Edzés véglegesítése gomb megnyomása	Hiba: Ellenőrid az ürlap mezőit toast üzenet	Hiba: Ellenőrid az ürlap mezőit toast üzenet	Sikeres <input checked="" type="checkbox"/>
TC025	Edzés szerkesztése	Edzés véglegesítése helyes mezőkkel	Edzés véglegesítése gombra nyomás	Edzés befejezése, Edzés értékelő oldalra való átirányítás	Edzés befejezése, Edzés értékelő oldalra való átirányítás	Sikeres <input checked="" type="checkbox"/>

TC026	Edzés áttekintése oldal	Annak az edzésnek az áttekintés oldalon vagyunk amely edzés még tart		a számláló továbbra is megy, edzés folytatása van kiirva a legfelső gombra	a számláló továbbra is megy, edzés folytatása van kiirva a legfelső gombra	Sikeres ✓
TC027	Edzés áttekintése oldal	Edzés folytatása gomb megnyomása ha éppen tart az edzés amelyet éppen áttekintünk	edzés folytatása gomb megnyomása	Annak az edzésnek a szerkesztő oldalára átirányít amelyet az imént áttekintettünk	Annak az edzésnek a szerkesztő oldalára átirányít amelyet az imént áttekintettünk	Sikeres ✓
TC028	Edzés áttekintése oldal	Ha az éppen áttekintés alatt lévő edzés már befejeződött és ma már volt edzésünk	Edzés újrahasználása gomb megnyomása	A mai nap már van edzés Toast üzenet nem indul új edzés	A mai nap már van edzés Toast üzenet nem indul új edzés	Sikeres ✓
TC029	Edzés áttekintése oldal	Ha az éppen áttekintés alatt lévő edzés még nem lett elkezdve/befejezve. Elkezdhető az edzés kezdése gomb használatával	Edzés kezdése gomb megnyomása	Edzés elkezdése, számláló elindul, edzesek/[edzesID]/szerkeszt oldalra való átirányítás	Edzés elkezdése, számláló elindul, edzesek/[edzesID]/szerkeszt oldalra való átirányítás	Sikeres ✓
TC030	Edzés áttekintése oldal	Edzés áttekintése oldal vissza gomb használata	Vissza gomb megnyomása	Visszavisz az ezelőtti oldalra	Visszavisz az ezelőtti oldalra	Sikeres ✓
TC031	Edzés áttekintése oldal	Edzés törlése	Törlés gomb megnyomása > Igen választása a megerősítő felugró ablakban	Edzés törölve	Edzés törölve	Sikeres ✓
TC032	Edzés előzmények	Kilistázott edzések megtekintése gomb megnyomásával megtekinthető az edzés	Edzés megtekintése gomb	Átvisz annak az edzésnek az áttekintés oldalára amelyen a gomb található volt	Átvisz annak az edzésnek az áttekintés oldalára amelyen a gomb található volt	Sikeres ✓

TC033	Edzés előzmények	Egy kilistázott edzés csillag iconjára kattintva kedvencek közé kerül az az edzés	csillag iconra kattintás	Kedvencek közé kerül az adott edzés	Kedvencek közé kerül az adott edzés	Sikeres ✓
TC034	Edzés előzmények	Listázási opciók használata az oldalon. Listázás kedvencek alapján	Kedvencek alapján kiválasztása	Az edzések kilistázásban továbbra is dátum alapján csökkenő a listázás, viszont azon belül a kedvenceket helyezi előre, és visszadob az első oldalra	Az edzések kilistázásban továbbra is dátum alapján csökkenő a listázás, viszont azon belül a kedvenceket helyezi előre, és visszadob az első oldalra	Sikeres ✓
TC034	Edzés előzmények	Listázási opciók használata az oldalon. Listázás dátum alapján növekvő	Dátum alapján növekvő kiválasztása	Dátum alapján növekvő sorrenbe listázza ki az edzéseket, és visszadob az első oldalra	Dátum alapján növekvő sorrenbe listázza ki az edzéseket, és visszadob az első oldalra	Sikeres ✓
TC034	Edzés előzmények	Listázási opciók használata az oldalon. Listázás dátum alapján csökkenő	Dátum alapján csökkenő kiválasztása	Dátum alapján csökkenő sorrenbe listázza ki az edzéseket, és visszadob az első oldalra	Dátum alapján csökkenő sorrenbe listázza ki az edzéseket, és visszadob az első oldalra	Sikeres ✓
TC034	Edzés előzmények, új edzés kezdése funkció	Ezen az oldalon is jelen van egy edzés kezdése gomb, ha a mai nap még nincs edzés és a megadott edzés neve hosszabb 3 karakternél akkor létrehozható egy új edzés	Edzés + gomb > Edzés neve: lábazás > Létrehozás	Edzés elkezdése, számláló elindul, edzesek/[edzesID]/szerkeszt oldalra való átirányítás	Edzés elkezdése, számláló elindul, edzesek/[edzesID]/szerkeszt oldalra való átirányítás	Sikeres ✓
TC034	Edzés előzmények, új edzés kezdése funkció	Ha mai napon már volt edzés	Edzés + gomb > Edzés neve: lábazás > Létrehozás	A mai nap már van edzés	A mai nap már van edzés	Sikeres ✓

TC035	Edzéstervezek oldal	Listázási opciók használata az oldalon. Listázás kedvencek alapján	Kedvencek alapján kiválasztása	Kedvenc edzéstervezeket adja vissza a lista elejére	Kedvenc edzéstervezeket adja vissza a lista elejére	Sikeres ✓
TC036	Edzéstervezek oldal	Listázási opciók használata az oldalon. Listázás dátum alapján növekvő	Dátum alapján növekvő kiválasztása	Dátum alapján növekvő sorrenbe listázza ki az edzéstervezeket, és visszadob az első oldalra	Dátum alapján növekvő sorrenbe listázza ki az edzéstervezeket, és visszadob az első oldalra	Sikeres ✓
TC037	Edzéstervezek oldal	Listázási opciók használata az oldalon. Listázás dátum alapján csökkenő	Dátum alapján csökkenő kiválasztása	Dátum alapján csökkenő sorrenbe listázza ki az edzéstervezeket, és visszadob az első oldalra	Dátum alapján csökkenő sorrenbe listázza ki az edzéstervezeket, és visszadob az első oldalra	Sikeres ✓
TC038	Edzéstervezek oldal	Edzéster szerkesztése az edzéster szerkesztése gomb segítségével	Edzéster megtekintése gomb megnyomása	Átirányít az adott edzéster szerkesztő oldalára	Átirányít az adott edzéster szerkesztő oldalára	Sikeres ✓
TC039	Edzéstervezek oldal	Edzéster létérehozása, helyes adatokkal	Edzéster + gomb > felugró ablakon belül > Edzéster neve: Teljes test edzés > létérehozás	Létrehoz egy edzéstervet és átirányít az edzéster szerkesztése oldalra	Létrehoz egy edzéstervet és átirányít az edzéster szerkesztése oldalra	Sikeres ✓
TC040	Edzéstervezek oldal	Edzéster létérehozása, hibás adatokkal	Edzéster + gomb > felugró ablakon belül > Edzéster neve: a	Az edzéster nevének legalább 3 karakter hosszúnak kell lennie figyelmeztető üzenet	Az edzéster nevének legalább 3 karakter hosszúnak kell lennie figyelmeztető üzenet	Sikeres ✓
TC041	Edzéster szerkesztése	Gyakorlat hozzáadása edzéstervezhez	Gyakorlat + gomb megnyomása > Gyakorlat kiválasztása > Hozzáadás	Gyakorlat hozzáadódik az edzéstervezhez, a gyakorlatban mozgatott izomcsoportok megjelennek a felső izomzat ábrán	Gyakorlat hozzáadódik az edzéstervezhez, a gyakorlatban mozgatott izomcsoportok megjelennek a felső izomzat ábrán	Sikeres ✓
TC042	Edzéster szerkesztése	Gyakorlat törlése edzésterverből	Hozzáadott gyakorlat, X gombra kattintása > felugró ablakon > Igen választása	Gyakorlat kitörlődik az edzésterverből, a gyakorlatban mozgatott izomcsoportok eltünnek a felső izomzat ábráról	Gyakorlat kitörlődik az edzésterverből, a gyakorlatban mozgatott izomcsoportok eltünnek a felső izomzat ábráról	Sikeres ✓

TC043	Edzésterv szerkesztése	Edzésterv mentése	Edzésterv mentése gomb megnyomása > felugró ablak > Igen választása	Az oldal visszairányít az edzéstervek oldalra és menti az edzésterven végrehajtott változtatásokat	Az oldal visszairányít az edzéstervek oldalra és menti az edzésterven végrehajtott változtatásokat	Sikeres ✓
TC043	Edzésterv szerkesztése	Edzésterv törlése	Terv törlése gomb megnyomása > felugró ablak > Igen választása	Az oldal visszairányít az edzéstervek oldalra, és kitörli az edzéstervet	Az oldal visszairányít az edzéstervek oldalra, és kitörli az edzéstervet	Sikeres ✓
TC044	Gyakorlatok szűrése	Gyakorlatok szűrése név alapján	Gyakorlat neve: fekvő	Kilistázza az összes gyakorlatot amik tartalmazzák a megadott gyakorlat nevet	Kilistázza az összes gyakorlatot amik tartalmazzák a megadott gyakorlat nevet	Sikeres ✓
TC045	Gyakorlatok szűrése	Gyakorlatok szűrése fő izomcsoport alapján	Fő izomcsoport szerint: Combefeszítő izom	Kilistázza az összes gyakorlatot amiknél egyezik a megadott főizomcsoport	Kilistázza az összes gyakorlatot amiknél egyezik a megadott főizomcsoport	Sikeres ✓
TC046	Gyakorlatok szűrése	Gyakorlatok szűrése izomcsoportok szerint	Izomcsoportok: Hasizom, Vádli izom, Mellizom	Kilistázza az összes gyakorlatot amik egyeznek a megadott izomcsoportokkal	Kilistázza az összes gyakorlatot amik egyeznek a megadott izomcsoportokkal	Sikeres ✓
TC047	Gyakorlatok szűrése	Gyakorlatok szűrése eszköz szerint	Eszköz szerint: kettlebell	Kilistázza az összes gyakorlatot ahol a megadott eszköz van használatban	Kilistázza az összes gyakorlatot ahol a megadott eszköz van használatban	Sikeres ✓
TC048	Profil oldal, BMI számító	Mezők üresen hagyása	Üres mezők > Mentés	A testsúly megadása kötelező A testmagasság megadása kötelező figyelmeztető üzenetek	A testsúly megadása kötelező A testmagasság megadása kötelező figyelmeztető üzenet	Sikeres ✓
TC049	Profil oldal, BMI számító	Magasság mező üresen hagyása	Testsúly: 80 Magasság: >Mentés	A testmagasság megadása kötelező figyelmeztető üzenet	A testmagasság megadása kötelező figyelmeztető üzenet	Sikeres ✓
TC050	Profil oldal, BMI számító	Helytelen testsúly megadása	Testsúly: 2 Magasság: 120 >Mentés	A testsúly nem lehet kevesebb, mint 20 kg figyelmeztető üzenet	A testsúly nem lehet kevesebb, mint 20 kg figyelmeztető üzenet	Sikeres ✓
TC051	Profil oldal, BMI számító	Helytelen testsúly megadása	Testsúly: 700 Magasság: 120 >Mentés	A testsúly nem lehet több, mint 635 kg figyelmeztető üzenet	A testsúly nem lehet több, mint 635 kg figyelmeztető üzenet	Sikeres ✓
TC052	Profil oldal, BMI számító	Helytelen magasság	Testsúly: 80 Magasság: 45	A testmagasság nem lehet kevesebb, mint 50	A testmagasság nem lehet kevesebb, mint 50	Sikeres ✓

		megadása	>Mentés	cm figyelmeztető üzenet	cm figyelmeztető üzenet	
TC053	Profil oldal, BMI számító	Helytelen magasság megadása	Testsúly: 80 Magasság: 260 >Mentés	A testmagasság nem lehet több, mint 255 cm figyelmeztető üzenet	A testmagasság nem lehet több, mint 255 cm figyelmeztető üzenet	Sikeress <input checked="" type="checkbox"/>
TC054	Profil oldal, BMI számító	Helyes adatok megadása	Testsúly: 80 Magasság: 180 >Mentés	Bmi kiszámítva, elmentve a felhasználó számára, Bmi megjelenítve	Bmi kiszámítva, elmentve a felhasználó számára, Bmi megjelenítve	Sikeress <input checked="" type="checkbox"/>
TC055	Statisztika oldal	Fejlődési idővonal szűrése	Gyakorlat kiválasztása (az opciónak változóak aszerint hogy az adott felhasználó milyen gyakorlatot végzett eddig)	A megadott gyakorlat alapú szűrés	A megadott gyakorlat alapú szűrés	Sikeress <input checked="" type="checkbox"/>
TC056	Statisztika oldal	Időtartam alapú szűrés	időtartam: hónap	Edzett izmok aránya, leedzett súly a megadott időtartam adatait jeleníti meg	Edzett izmok aránya, leedzett súly a megadott időtartam adatait jeleníti meg	Sikeress <input checked="" type="checkbox"/>

Admin funkciók

TC ID	Funkció/Modu l	Teszt leírása	Bemenet	Elvárt eredmény	Tényleges eredmény	Teszt státusz
TC057	Gyakorlat szerkesztése gomb	Ha admin fiókkal vagyunk bejelentkezve egy ceruza ikonnak kell megjelennie a gyakorlat jobb felső sarkában	Gyakorlatok oldal/gyakorlat megtekintése oldal	Ceruza ikon gomb megjelenik a gyakorlatok jobb felső sarkában	Ceruza ikon gomb megjelenik a gyakorlatok jobb felső sarkában	Sikeress <input checked="" type="checkbox"/>
TC058	Gyakorlat szerkesztése	Gyakorlat helytelen szerkesztése, gyakorlat neve kitörlése	gyakorlat neve mező üresen hagyása	A gyakorlat neve kötelező figyelmeztető üzenet	A gyakorlat neve kötelező figyelmeztető üzenet	Sikeress <input checked="" type="checkbox"/>
TC059	Gyakorlat szerkesztése	Gyakorlat helytelen szerkesztése, Izomcsoport mező üresen hagyása	Izomcsoport mező üresen hagyása	Legalább egy izomcsoportot ki kell választani figyelmeztető üzenet	Legalább egy izomcsoportot ki kell választani figyelmeztető üzenet	Sikeress <input checked="" type="checkbox"/>

TC060	Gyakorlat szerkesztése	Gyakorlat helyes szerkesztése	Gyakorlat neve: Gugolás Fő izomcsoport: Combhajlító izom Izomcsoportok: Farizom Eszköz: saját testsúly Gyakorlat leírása: természetes gugolás	A gyakorlat frissítődik az összes felhasználónak, visszadob a gyakorlat megtekintése oldalra	A gyakorlat frissítődik az összes felhasználónak, visszadob a gyakorlat megtekintése oldalra	Sikeress ✓
TC061	Gyakorlat szerkesztése	Gyakorlat törlése	Törlés gomb > felugró ablak > Igen választása	Gyakorlat kitörlődik az összes felhasználó számára, visszadob a gyakorlatok kilistázása oldalra	Gyakorlat kitörlődik az összes felhasználó számára, visszadob a gyakorlatok kilistázása oldalra	Sikeress ✓
TC061	Admin felület, gomb a menüsoron	Ha adminfiókkal vagyunk bejelentkezve, Admin felület gomb jelenik meg a menüsor alján	menüsor	Admin felület gomb megjelenik	Admin felület gomb megjelenik	Sikeress ✓
TC062	Admin felület, gomb a profil oldalon	Ha adminfiókkal vagyunk bejelentkezve, Admin felület gomb jelenik meg a profil oldalon	profil oldal	Admin felület gomb megjelenik	Admin felület gomb megjelenik	Sikeress ✓
TC062	Admin felület	Felhasználó alapú szűrés	Felhasználónév: User	Felhasználók kilistázása ahol szerepel a megadott név	Felhasználók kilistázása ahol szerepel a megadott név	Sikeress ✓
TC063	Admin felület	Email alapú szűrés	Email: user@gmail.com	Felhasználók kilistázása ahol szerepel a megadott email	Felhasználók kilistázása ahol szerepel a megadott email	Sikeress ✓
TC064	Admin felület	Admin jog alapú szűrés	Admin jog: Felhasználó	Felhasználók kilistázása ahol a felhasználó felhasználó jogosultságu	Felhasználók kilistázása ahol a felhasználó felhasználó jogosultságu	Sikeress ✓

TC065	Admin felület	Admin jogosultságadása	Felhasználónál aki jelenleg még nem admin > Admin hozzáférés + gomb megnyomása > felugró ablak > Igen választása	A felhasználó admin jogosultá válik	A felhasználó admin jogosultá válik	Sikeres ✓
TC066	Admin felület	Admin jogosultság megvonása	Felhasználónál aki jelenleg admin > Admin hozzáférés - gomb megnyomása > felugró ablak > Igen választása	A felhasználó admin jogosultságának törlése	A felhasználó admin jogosultságának törlése	Sikeres ✓
TC067	Admin felület	Felhasználó törlése	Kuka icon gomb megnyomása > felugró albak > Igen	A felhasználó törlődik	A felhasználó törlődik	Sikeres ✓

4.7 e2e teszt

 futtatás: `npm run test:e2e`

FIGYELEM! (Éles adatbázis használat)

Mivel nem tartottuk előnyösnek egy külön tesztadatbázis létrehozását, és a mockolt adatok alkalmazása ellentmondana az E2E tesztelés céljának, ezért az éles adatbázison hajtjuk végre a lekérdezéseket.

Az éles adatbázis használatának fő indoka, hogy a tesztadatok könnyen újra seedelhetőek, így biztosítjuk a konzisztens tesztelési környezetet. Mivel egy kisebb projektről van szó, amely bemutatásra kerül, úgy döntöttünk, hogy ennél a megoldásnál maradunk, elkerülve a felesleges adatbázis-migrációs és karbantartási feladatokat. Ez lehetővé teszi, hogy az E2E tesztek valós adatstruktúrán és kapcsolatokon fussanak, biztosítva a hitelesebb eredményeket. Áttekintés

seedelést elvégző bash file : `run-seeds.sh`

részletes leírás:  [2.1 Backend](#)

Ez a teszt szkript egy NestJS alapú alkalmazás "EdzesController" komponensének végpontjait vizsgálja ami lényegében az applikációnk gerince. A cél, hogy validálja a műveletek helyes működését, beleértve a hitelesítést, adatvalidációt és a különböző üzleti szabályok érvényesítését. A tesztek során a Prisma ORM segítségével a tesztadatbázis állapotát is kezeljük (törölés, létrehozás).

Tesztkörnyezet beállítása

- **NestJS alkalmazás inicializálása:**

A teszt modul inicializálásakor egy teljes NestJS alkalmazást hozunk létre, melyre globális ValidationPipe van beállítva.

- **Tesztadatbázis kezelése:**

Minden egyes teszt előtt a releváns adatbázis táblákat töröljük (például edzés, gyakorlat, felhasználó adatok), majd létrehozunk egy teszt felhasználót.

- **JWT token generálás:**

A hitelesítés teszteléséhez a Prisma által létrehozott teszt felhasználó adataival egy JWT token kerül generálásra, amelyet a kérésekhez csatolunk.

Tesztelt Végpontok és Funkcionalitások

1. POST /edzes – Új edzés létrehozása

- **Sikeres edzés létrehozás:** A megadott érvényes adatokkal (edzés neve, időtartam, dátum, stb.) létrejön egy új edzés, amely visszaadja az edzés azonosítóját és adatait.
- **Hitelesítési hibák:** A hitelesítetlen kérések esetén 401-es státusz kódot várunk.
- **Hibás felhasználói azonosító:** Amennyiben a request-ben szereplő `user_id` nem egyezik a tokenben szereplő felhasználóval, 403-as hibát kapunk.
- **Adatvalidációs hibák:** Hibás adatok (például üres edzés neve, negatív idő) esetén 400-as hibakódöt kapunk, és a hibaüzenetek tömb formátumúak.

2. GET /edzes – Felhasználó edzéseinek lekérése

- **Sikeres lekérés:** A hitelesített felhasználó edzései tömbben visszatérnek, ellenőrizve, hogy a lekért elemek rendelkeznek az edzés azonosítóval és megfelelő nevével.
- **Hitelesítés hiánya:** Ha a kérés nem hitelesített, 401-es hibakódot kapunk.

3. GET /edzes/:id – Konkrét edzés lekérése

- **Sikeres lekérés:** A megadott azonosítóhoz tartozó edzés adatai (pl. név, azonosító) visszakapók.
- **Nem létező edzés:** Ha a megadott edzés nem található, 404-es hibakódot várunk.
- **Hitelesítés hiánya:** Hitelesítés nélkül a kérés 401-es kódot eredményez.

4. PATCH /edzes/:id/finalize/:userId – Edzés lezárása

- **Sikeres lezárás:** A felhasználó által létrehozott edzés lezáródik (isFinalized értéke true lesz).
- **Jogosultság hiba:** Ha egy másik felhasználó edzését próbáljuk lezárni, a rendszer 403-as hibát ad vissza.

5. POST /edzes/:id/gyakorlat/:userId – Gyakorlat hozzáadása egy edzéshez

- **Sikeres hozzáadás:** Egy létező gyakorlat hozzáadódik az adott edzéshez, és az eredmény tartalmazza a hozzáadott gyakorlatok listáját.
- **Duplikált hozzáadás:** Ha már hozzá lett adva az adott gyakorlat, a rendszer 409-es hibával jelzi, hogy a gyakorlat már szerepel az edzésben.

6. DELETE /edzes/:id/gyakorlat/:gyakorlatId/:userId – Gyakorlat eltávolítása egy edzésből

- **Sikeres eltávolítás:** Az előzőleg hozzáadott gyakorlat eltávolításra kerül az edzésből, ezt követően a lekérés már nem tartalmazza az adott gyakorlatot.

Összefoglalás

Ez az e2e teszt szkript lefedi az edzés-kezelő végpontok teljes funkcionálitását, beleértve az alábbiakat:

- **CRUD műveletek:** Edzések létrehozása, lekérése, módosítása (lezárás) és törlése (gyakorlat eltávolítás).
- **Hitelesítés és jogosultság kezelés:** Teszteli, hogy csak a megfelelő jogosultságokkal rendelkező felhasználók tudják elvégezni az egyes műveleteket.
- **Adatvalidáció:** Ellenőrzi, hogy a bemeneti adatok megfelelnek-e az elvárt formátumnak és szabályoknak.
- **Üzleti logika:** Ellenőrzi a felhasználói azonosítók egyezőségét és a duplikált adatok kezelését.

Ezzel a tesztelési stratégiával biztosítjuk, hogy a "EdzesController" végpontok megbízhatóan és biztonságosan működnek az alkalmazásban.

```
> nestjs-keret-2024@0.0.1 test:e2e
> jest --config ./test/jest-e2e.json

PASS  test/app.e2e-spec.ts (6.337 s)
PASS  test/edzes.e2e-spec.ts (6.796 s)
● Console

  console.log
    107 37

    at EdzesService.changeEdzesFinalizedStatus (./src/edzes/edzes.service.ts:1123:13)

  console.log
    108 39

    at EdzesService.changeEdzesFinalizedStatus (./src/edzes/edzes.service.ts:1123:13)

Test Suites: 2 passed, 2 total
Tests:       15 passed, 15 total
Snapshots:  0 total
Time:        7.296 s
```