BGSzC Pestszentlőrinci Közgazdasági és Informatikai Szakgimnáziuma

1184 Budapest Hengersor 34.

Záró dolgozat

Személyi Edző Kereső

Konzulens tanár: Dobrocsi Csilla Készítette: Chen Wen Jie, Pribelszki Dávid

Tartalom

[1 Bevezetés 3](#_Toc195175706)

[1.1 Feladat leírás 3](#_Toc195175707)

[1.2 A felhasznált ismeretek 3](#_Toc195175708)

[1.3 A felhasznált szoftverek 4](#_Toc195175709)

[2 Felhasználói dokumentáció 5](#_Toc195175710)

[2.1 A program általános specifikációja 5](#_Toc195175711)

[2.2 Rendszerkövetelmények 5](#_Toc195175712)

[Hardver követelmények 5](#_Toc195175713)

[Szoftver követelmények 5](#_Toc195175714)

[2.3 A program telepítése 5](#_Toc195175715)

[2.4 A program használatának a részletes leírása 7](#_Toc195175716)

[3 Fejlesztői dokumentáció 9](#_Toc195175717)

[3.1 Az alkalmazott fejlesztői eszközök 9](#_Toc195175718)

[3.2 Az adatmodell működésének leírása 10](#_Toc195175719)

[3.3 Részletes feladatspecifikáció, algoritmusok 13](#_Toc195175720)

[5. Foglalási logika: 14](#_Toc195175721)

[3.4 Tesztelési dokumentáció 15](#_Toc195175722)

[1. Teszteset: Edzők regisztrációja és adatmegjelenítése 15](#_Toc195175723)

[2. Teszteset: Időpontfoglalási rendszer működése 16](#_Toc195175724)

[3. Teszteset: Közösségi oldalon megjelenő bejegyzések 18](#_Toc195175725)

[4. Teszteset: Edzők profiljának módosítása 20](#_Toc195175726)

[4 Összefoglalás 22](#_Toc195175727)

[4.1 Önértékelés 22](#_Toc195175728)

[5 Felhasznált irodalom 25](#_Toc195175729)

# Bevezetés

## Feladat leírás

Szakdolgozatunk keretében egy "FitConnect" nevű webalkalmazást fejlesztettünk, amely hidat képez a személyi edzők és a potenciális ügyfelek között. A projekt célja egy olyan platform létrehozása volt, ahol az edzők hatékonyan bemutathatják szolgáltatásaikat, az ügyfelek pedig egyszerűen megtalálhatják a számukra legmegfelelőbb edzőt, kapcsolatba léphetnek velük emailen keresztül, és időpontot foglalhatnak egy integrált rendszer segítségével. A fejlesztést ketten végeztük, megosztva a frontend, backend és dizájn feladatköröket, hogy egy modern, reszponzív és felhasználóbarát alkalmazást hozzunk létre.

A "FitConnect" egy olyan platform, amely két fő célcsoportot szolgál ki: a személyi edzőket és az ügyfeleket. Az edzők részletes profilt hozhatnak létre, amely tartalmazza szolgáltatásaikat, áraikat, elérhetőségüket és szakmai hátterüket. Az ügyfelek szűrők segítségével kereshetnek edzőket (pl. helyszín, szakterület, ár), megtekinthetik profiljaikat, és időpontot foglalhatnak egy beépített naptárrendszeren keresztül. A platform egyszerű és intuitív kezelést biztosít mindkét fél számára, támogatva a hatékony kommunikációt és az edzéstervezést.

## A felhasznált ismeretek

A projekt során a következő területeken szerzett ismereteinket alkalmaztuk:

* **Frontend fejlesztés**: Modern JavaScript/TypeScript, React komponensek fejlesztése.
* **UI/UX tervezés**: Reszponzív felületek, Tailwind CSS és Shadcn/ui használata.
* **Backend alapok**: Adatmodellezés, API integráció (pl. email értesítések).
* **Verziókezelés**: Git és GitHub közös munkafolyamatokhoz.
* **Projektmenedzsment**: Feladatok megosztása és koordináció két fejlesztő között.

A munkát megosztottuk: Pribelszki Dávid a frontend fejlesztésre és dizájnra fókuszált, míg Chen Wen Jie az adatmodellezést és a funkcionális logikát támogatta.

## A felhasznált szoftverek

A fejlesztés során a következő technológiákat és eszközöket használtuk:

* **React 18.3.1**: Gyors és modern frontend keretrendszer, hatékony komponenskezeléshez.
* **TypeScript**: Erősen típusos programozás a hibák csökkentésére és jobb fejlesztői élményre.
* **Vite**: Gyors build eszköz és fejlesztői szerver a hatékony munkához.
* **Tailwind CSS**: Utility-first CSS keretrendszer a dizájn gyorsítására.
* **Shadcn/ui**: Könnyen integrálható React komponens könyvtár az egységes UI-hoz.
* **Lucide-react**: Modern ikonok az esztétikus megjelenésért.
* **Node.js és npm**: Futtatókörnyezet és csomagkezelés.
* **PostgreSQL**: Adatbázis az edzői és foglalási adatok tárolására.
* **Git**: Verziókezelés a közös munkához.
* **ChatGPT**: [ChatGPT | OpenAI](https://openai.com/chatgpt/overview/) A mesterséges intelligencia is sokat segített a projekt elkészítésében. Minden folyamatban hasznos segítséget nyújtott, ha elakadtunk és tanácstalanok lettünk, segített utat mutatni, így hatékonyak és produktívak tudtunk maradni mindig.

# Felhasználói dokumentáció

## A program általános specifikációja

Az alkalmazás négy fő modulból áll:

1. **Landing oldal**: Bemutatja a platformot, kiemeli előnyeit, és gyors keresést kínál.
2. **Edzők kezelése**: Edzői profilok, keresési és szűrési funkciók (helyszín, szakterület, ár).
3. **Foglalási rendszer**: Naptár alapú időpontfoglalás emailes visszaigazolással.
4. **Felhasználói fiók**: Edzők és ügyfelek számára külön kezelőfelület.

## Rendszerkövetelmények

## Hardver követelmények

* **Hardver**: Minimum 4 GB RAM, 2 GHz processzor, stabil internetkapcsolat.
* **Szoftver**: Modern böngésző (Chrome, Firefox, Edge), Node.js 18+ a fejlesztéshez.

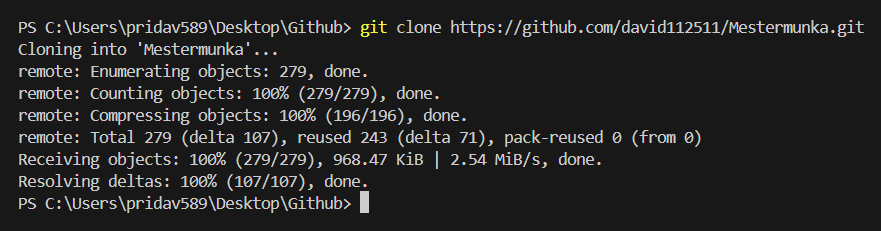
## Szoftver követelmények

* Visual studio code
* Windows 10/11

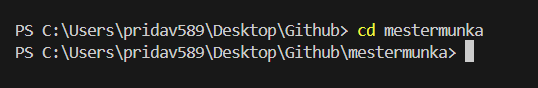
## A program telepítése

A fejlesztői környezet beállítása:

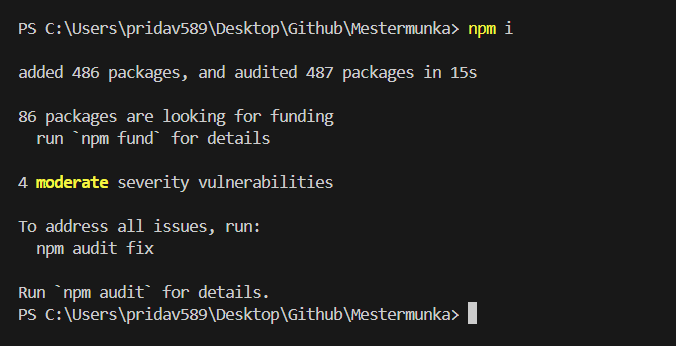
1. Repository klónozása: git clone <repository-url>

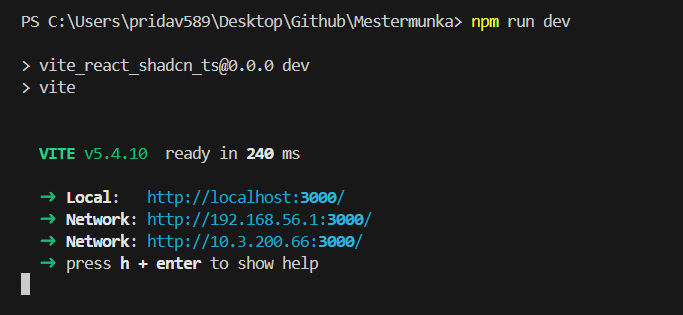


1. Belépés a letöltött mappába: cd mestermunka



1. Függőségek telepítése: npm install



1. Fejlesztői szerver indítása: npm run dev  
   

## A program használatának a részletes leírása

Az alkalmazás főbb funkciói és használatuk:

1. **Bejelentkezés/Regisztráció**

* A regisztráció két fiókkal történhet:
  + Felhasználó
  + Edző

1. **Edzők keresése:**
   * Az ügyfelek a főoldalon vagy a keresési oldalon szűrhetik az edzőket (pl. "Budapest, jóga, 5000 Ft/óra").
   * Az edzői profil részletes adatokat mutat (név, tapasztalat, árak), és lehetőséget ad emailes kapcsolatfelvételre vagy foglalásra.
   * Funkció: Szűrők alkalmazása, profil megtekintése, "Foglalás" gomb.
2. **Időpontfoglalás:** 
   * Az edző profilján a naptárban láthatóak a szabad időpontok.
   * Az ügyfél kiválaszt egy időpontot, majd a rendszer emailben értesíti mindkét felet.
3. **Profil módosítása:**

* A felhasználók a navigációs sávon keresztül a profiljukra kattintva a profilom menüponton belűl a személyes adataim menüben tudják kezelni a nyilvános adataikat a profil táblából.
* Az edzőknek külön elérhető egy edzői profil beállítás, ahol az trainers táblában lévő adataikat tudják frissíteni.
* A fiók beállítások menüben pedig az fiók azonosításához szükséges adatokat lehet módosítani.

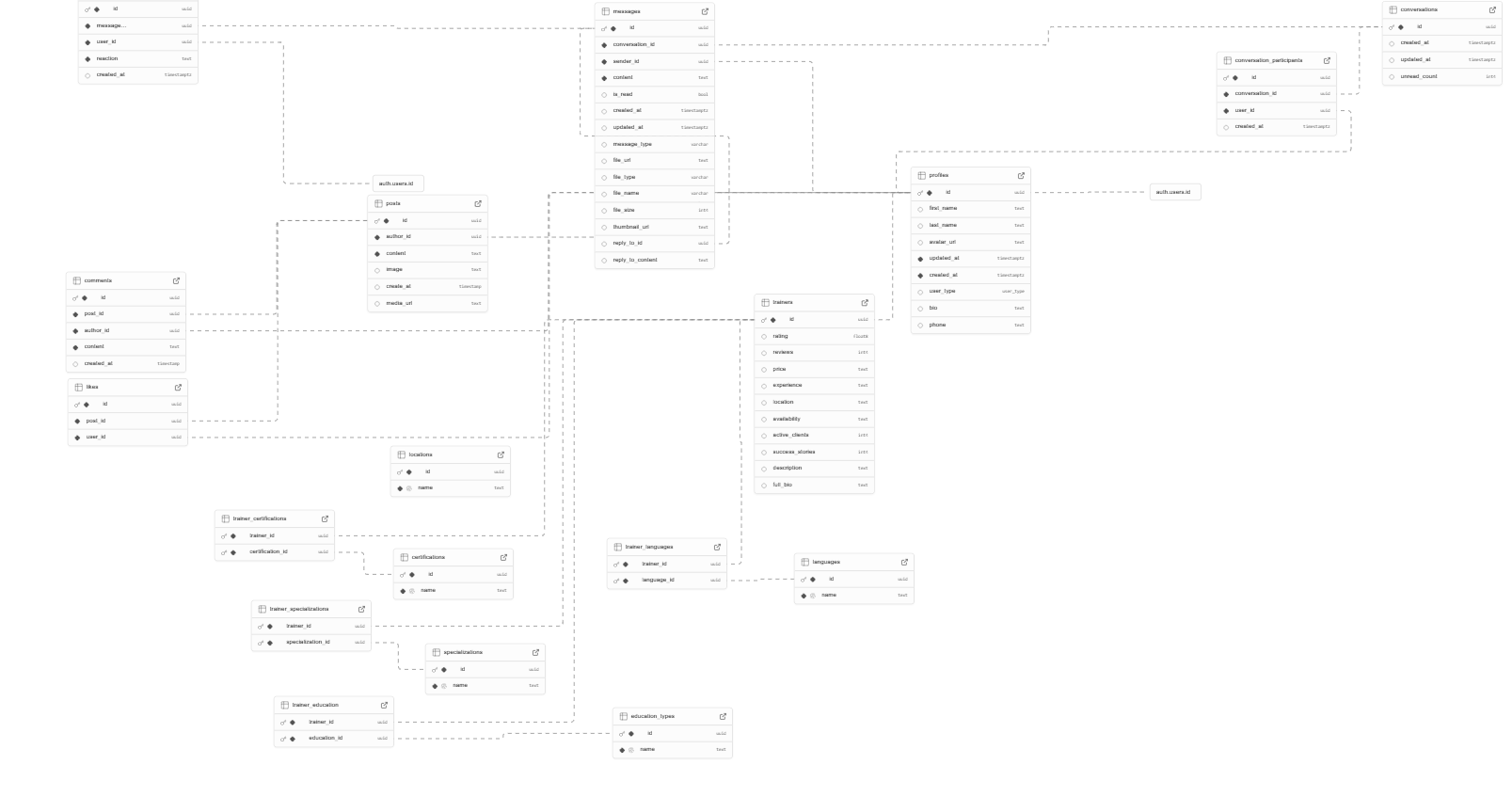
# Fejlesztői dokumentáció

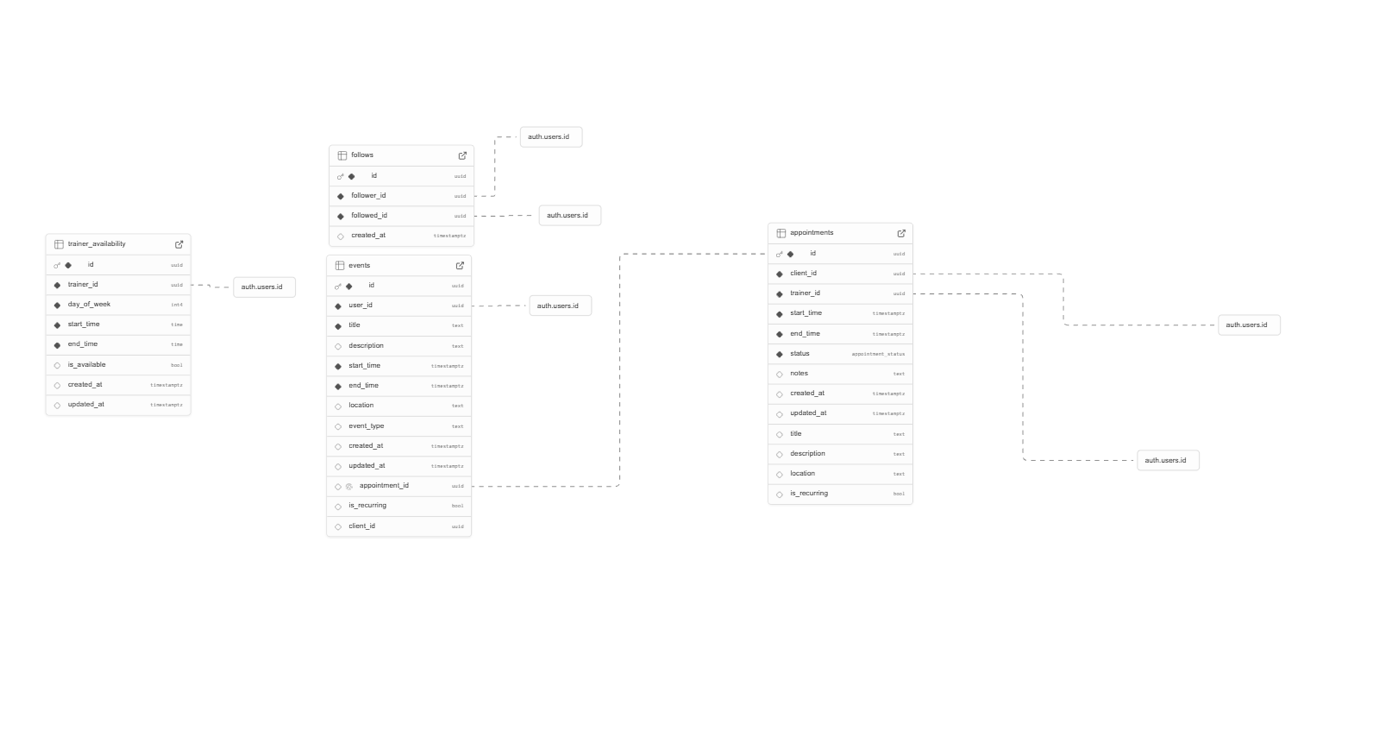
## Az alkalmazott fejlesztői eszközök

**A fejlesztés során használt eszközök és környezetek:**

* - Visual Studio Code - Elsődleges kódszerkesztő
* - Git/GitHub: Verziókezelés és kollaboráció.
* - Node.js - JavaScript futtatókörnyezet
* - npm - Csomagkezelő
* - Chrome DevTools - Fejlesztői eszközök
* Supabase - Adatbázis kezelés [Supabase | The Open Source Firebase Alternative](https://supabase.com/)
* Shadcn-ui – Hogy valamivel rugalmasabb dizájn biztosítsunk helyenként [Build your component library - shadcn/ui](https://ui.shadcn.com/)
* Tailwind-CSS - [Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.](https://tailwindcss.com/)
* React
* TypeScript
* Vite
* Lucide-react - Kevésbé rugalmas, de egységes dizájnt biztosít.

## Az adatmodell működésének leírása





A FitConnect alkalmazás adatmodellje egy olyan adatbázis-struktúrát és kapcsolatrendszert biztosít, amely lehetővé teszi a felhasználók, edzők és közösségi interakciók hatékony kezelését. Az alábbiakban bemutatom, hogyan működik ez a rendszer a gyakorlatban, a főbb egységekre bontva.

#### **1. Felhasználói adatok kezelése**

A rendszer alapja a felhasználók azonosítása és adataik tárolása:

* **Regisztráció és autentikáció**: Amikor egy felhasználó regisztrál, adatai az auth.users táblába kerülnek, amely az azonosításhoz szükséges alapvető információkat (pl. e-mail, jelszó) tartalmazza.
* **Profiladatok**: A részletes személyes adatok – mint a név, avatar vagy bio – a profiles táblában tárolódnak. Ez a tábla az auth.users táblához kapcsolódik az id mezőn keresztül, így minden felhasználóhoz egyedi profil tartozik.

Ez a kettős struktúra lehetővé teszi, hogy az authentikáció és a profiladatok külön kezelhetők legyenek, miközben összekapcsolhatók maradnak.

#### **2. Edzői adatok és szolgáltatások**

Az edzők speciális szerepet töltenek be, ezért adataik és szaktudásuk részletesen nyilvántartott:

* **Edzői profil**: Az edzők szakmai információi (pl. értékelés, tapasztalat, működési hely) a trainers táblában találhatók. Ez a tábla a profiles táblához kapcsolódik az id mezővel, vagyis minden edző egyben felhasználó is.
* **Szakterületek és képesítések**: Az edzők tudása több szempontból is kategorizálható:
  + A trainer\_specializations tábla az edzőket a specializations táblával köti össze, így látható, hogy ki milyen edzéstípusban (pl. jóga, erőnlét) jártas.
  + A trainer\_languages a languages táblával mutatja meg, milyen nyelveken vállalnak edzést.
  + A trainer\_education és trainer\_certifications táblák az edzők végzettségeit (education\_types) és tanúsítványait (certifications) rögzítik. Ezek mind sok-a-sokhoz kapcsolatok, amelyeket köztes táblák kezelnek, így egy edző több szakterülettel, nyelvtudással vagy végzettséggel is rendelkezhet.
* **Helyszínek**: Az edzők működési helyeit a locations tábla tartalmazza, amely a trainers táblához location\_id-val kapcsolódik. Ez alapján a felhasználók láthatják, hol érhető el az edző.

Ez a rendszer rugalmasan támogatja az edzők keresését és szűrését, például hely, szakterület vagy nyelvtudás alapján.

#### **3. Közösségi interakciók**

A FitConnect közösségi funkciókat is kínál, ahol a felhasználók posztokat hozhatnak létre és interakcióba léphetnek egymással:

* **Posztok**: A posts tábla tárolja a felhasználók által létrehozott tartalmakat (pl. edzéstippek, élmények). Minden poszt szerzője az author\_id révén a profiles táblához kötött, így egyértelmű, ki tette közzé.
* **Kommentek**: A posztokhoz fűzött hozzászólások a comments táblában vannak, ahol a post\_id a posztra, az author\_id pedig a kommentelő profiljára utal. Ez lehetővé teszi a beszélgetések követését.
* **Like-ok**: A posztok kedveléseit a likes tábla rögzíti, a post\_id a posztra, a user\_id pedig a kedvelő profiljára hivatkozik. Így mérhető egy poszt népszerűsége.

Ez a struktúra egy dinamikus közösségi felületet biztosít, ahol a felhasználók tartalmat oszthatnak meg, reagálhatnak egymásra, és nyomon követhetik az interakciókat.

#### **Hogyan működik együtt az egész?**

* Egy új felhasználó regisztrál, adatai az auth.users és profiles táblákba kerülnek. Ha edzővé válik, a trainers táblában rögzítik szakmai adatait, és megadhatja szakterületeit, nyelveit, végzettségeit a megfelelő táblákon keresztül.
* Az edzők keresésekor a rendszer a trainers, locations és a szakterület-táblák adatait használja fel, hogy a felhasználók megtalálják a számukra megfelelő szakembert.
* Közben a felhasználók posztokat írnak, kommentelnek és like-olnak, amit a posts, comments és likes táblák kezelnek, összekapcsolva a profilokkal.

Ez az adatmodell biztosítja, hogy a FitConnect egyszerre legyen hatékony edzőkereső platform és aktív közösségi tér, miközben az adatok integritása és a kapcsolatok logikája megmarad.

## Részletes feladatspecifikáció, algoritmusok

A fejlesztés során implementált főbb komponensek és funkciók:

1. Navigációs rendszer:

- React Router-rel kezelt útvonalak (pl. /coaches/:id az edzői profilhoz).

- Dinamikus navigáció

- Védett útvonalak a bejelentkezéshez kötött funkciókhoz.

2. Komponens architektúra:

- Újrafelhasználható komponensek (pl. CoachCard, BookingForm).

- Shadcn/ui integráció a konzisztens dizájnért.

- Egyedi hook-ok fejlesztése

3. Állapotkezelés:

- React hooks használata

- Lokális komponens állapotok

- Globális alkalmazás állapot

4. UI/UX megvalósítás:

- Tailwind CSS alapú styling

- Reszponzív dizájn

- Animációk és átmenetek

## 5. Foglalási logika:

- Algoritmus: Az edző availability tömbjéből szűri a szabad időpontokat,

majd a booking objektumot menti az adatbázis

## Tesztelési dokumentáció

## 1. Teszteset: Edzők regisztrációja és adatmegjelenítése

**Cél:** Biztosítani, hogy az edzők regisztrációja után adataik helyesen mentődjenek és megjelenjenek az edzők oldalon.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Lépés | Tevékenység | Várt eredmény | Kapott eredmény | Probléma leírása | Megjegyzés és megoldás |
| 1 | Új edző regisztrálása: Kitöltjük a regisztrációs űrlapot és elküldjük. | Az edző regisztrálódik, és profilja megjelenik az edzők oldalon. | Az edző regisztrálódott, de nem jelent meg az edzők oldalon. | Az oldal nem mutatta az új edzőket, mert az RLS (Row-Level Security) szabályok korlátozták az adatokat. | Megvizsgáltuk az adatbázist, és rájöttünk, hogy az RLS hibás beállítása a gond. Átállítottuk az RLS-t, hogy az autentikált felhasználók számára olvasható legyen az adat. |
| 2 | Újabb edző regisztrálása és ellenőrzés az edzők oldalon. | Az edző megjelenik az edzők oldalon, minden adatával együtt. | Az edző megjelent, de a neve és profilkép nem volt látható. | A név és profilkép nem mentődött megfelelően az adatbázisba. | Ellenőriztük a regisztrációs űrlapot, és kiderült, hogy ezek a mezők nem voltak helyesen leképezve az adatbázisra. Javítottuk az űrlap kezelését, hogy a név és profilkép is mentésre kerüljön. |
| 3 | Harmadik edző regisztrálása és az összes adat ellenőrzése. | Minden adat (név, profilkép stb.) helyesen megjelenik az edzők oldalon. | Az adatok végül helyesen megjelentek, miután javítottuk az adatlehívást. | Kezdetben az adatlehívó lekérdezés nem hozta fel az összes szükséges mezőt. | Módosítottuk a lekérdezést, hogy minden releváns adatot lekérjen az edzők táblából, így megoldódott a probléma. |

### 2. Teszteset: Időpontfoglalási rendszer működése

**Cél:** Ellenőrizni, hogy az időpontfoglalási rendszer helyesen kezeli a foglalásokat, értesítéseket küld, és frissíti a naptárat.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Lépés | Tevékenység | Várt eredmény | Kapott eredmény | Probléma leírása | Megjegyzés és megoldás |
| 1 | Időpont foglalása: Kliensként kiválasztunk egy edzőt és lefoglalunk egy időpontot. | Az időpont lefoglalódik, az edző emailben és az oldalon is értesítést kap. | Az időpont lefoglalódott, de az edző nem kapott emailt. | Az email értesítés nem indult el a foglalás után. | Megnéztük az email szolgáltatás integrációját, és hibás konfigurációt találtunk. Javítottuk az email beállításokat, hogy az értesítések elküldésre kerüljenek. |
| 2 | Edzőként bejelentkezve ellenőrizzük az oldalon belüli értesítéseket. | Az edző látja az új foglalásról szóló értesítést a fiókjában. | Nem jelent meg értesítés az edző felületén. | Az oldalon belüli értesítési rendszer nem volt implementálva. | Hozzáadtunk egy értesítési komponenst az edzői műszerfalhoz, és összekötöttük a foglalási rendszerrel. |
| 3 | Ellenőrizzük az edző naptárát a foglalás után. | A lefoglalt időpont nem foglalható újra, a naptár frissül. | A naptár nem frissült, az időpont szabad maradt. | A naptárfrissítési logika hibásan működött. | Átdolgoztuk a naptárkezelő kódot, hogy a lefoglalt időpontok helyesen jelölődjenek, így megoldódott a probléma. |

### 

### 3. Teszteset: Közösségi oldalon megjelenő bejegyzések

**Cél:** Biztosítani, hogy a közösségi oldalon a bejegyzések helyesen jelenjenek meg.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Lépés | Tevékenység | Várt eredmény | Kapott eredmény | Probléma leírása | Megjegyzés és megoldás |
| 1 | Új bejegyzés létrehozása: Felhasználóként szöveget és képet teszünk közzé. | A bejegyzés mentésre kerül és megjelenik az oldalon. | A bejegyzés mentésre került, de nem jelent meg az oldalon. | A bejegyzéseket lekérdező lekérdezés nem működött megfelelően. | Megvizsgáltuk a lekérdezést, és hiányzott egy csatlakozás a profilok táblához. Frissítettük a lekérdezést, hogy helyesen működjön. |
| 2 | Az oldal frissítése és a bejegyzés ellenőrzése. | A bejegyzés látható, a szerző nevével és tartalommal együtt. | A bejegyzés megjelent, de a szerző neve hiányzott. | A szerző neve nem került lehívásra az adatbázisból. | Módosítottuk a lekérdezést, hogy a profilok táblából a szerző nevét is lekérje. |
| 3 | A bejegyzés képének ellenőrzése. | A kép látható a bejegyzésben. | A kép nem töltött be. | A kép URL-je hibásan lett generálva. | Javítottuk a kép URL-ének előállítását, hogy a helyes tárolási helyre mutasson, így a kép is megjelent. |

### 4. Teszteset: Edzők profiljának módosítása

**Cél:** Ellenőrizni, hogy az edzők profiljának módosításai helyesen mentődjenek és megjelenjenek az adatbázisban.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Lépés | TEVÉKENYSÉg | Várt eredmény | Kapott eredmény | Probléma leírása | Megjegyzés és megoldás |
| 1 | Profil szerkesztése: Edzőként módosítjuk az életrajzot és új tanúsítványt adunk hozzá. | A változtatások mentésre kerülnek és láthatóak a profiloldalon. | A módosítások nem jelentek meg a profiloldalon. | Az űrlap nem küldte el helyesen az adatokat az adatbázisba. | Ellenőriztük az űrlap beküldését, és javítottuk, hogy minden mező adata eljusson az adatbázisba. |
| 2 | Adatbázis ellenőrzése a módosítások után. | Az adatbázisban látható az új életrajz és tanúsítvány. | Az életrajz frissült, de a tanúsítvány nem került bele. | A tanúsítvány mentési logikája hiányzott. | Bevezettük a tanúsítványok mentésére szolgáló logikát a profilfrissítési folyamatba. |
| 3 | Profiloldal frissítése és ellenőrzése. | Az életrajz és a tanúsítvány is látható a profiloldalon. | A javítás után minden adat helyesen megjelent. | A probléma megoldódott a logika implementálásával. | Nincs további teendő, a frissítések most már megfelelően működnek. |

# Összefoglalás

## Önértékelés

A "FitConnect" projekt fejlesztése során mi, Pribelszki Dávid és Chen Wen Jie, egy igazi csapatként dolgoztunk együtt, és mindketten beleadtuk szívünket-lelkünket a munkába. Számomra (Dávid) a frontend világa, a React és a Tailwind CSS használata volt a legizgalmasabb, míg Chen az adatbázisok és a backend logika terén érezte magát otthon. A közös munka tele volt lendülettel, de nem tagadom, néha kaotikusnak éreztük, hogy mindent összehangoljunk.

#### **Hogyan éltük meg?**

Az elején hatalmas lelkesedéssel vágtunk bele, de hamar rájöttünk, hogy a tervezés és a valóság nem mindig találkozik. A Supabase integrációja például váratlanul sok fejtörést okozott, és az időpontfoglaló rendszer összekötése a frontenddel is próbára tette a türelmünket. Mégis, ezek a nehézségek hozták ki belőlünk azt a pluszt, hogy ne adjuk fel, hanem keressünk megoldást együtt.

#### **Milyen hibákkal szembesültünk, és hogyan oldottuk meg?**

Technikai gondok akadtak bőven: a Supabase néha nem úgy működött, ahogy vártuk, és a GitHub-on is voltak kisebb kavarodások a commit-ok miatt. A kommunikációnk sem volt mindig tökéletes, főleg, amikor siettünk, és nem egyeztettük elég pontosan a feladatokat. Ezeket végül úgy hidaltuk át, hogy sűrűbben tartottunk rövid egyeztetéseket, és alaposabban átnéztük a dokumentációt. A hibák nem voltak leküzdhetetlenek, csak időt és figyelmet igényeltek.

#### **Mit tanultunk?**

A "FitConnect" projekt fejlesztése során rengeteget fejlődtünk szakmailag, és mindketten, Pribelszki Dávid és Chen Wen Jie, jelentős szintet léptünk a frontend és backend fejlesztés terén. Ez volt az első teljes értékű fullstack projektünk, amelynek köszönhetően mélyebb betekintést nyertünk az MVC (Model-View-Controller) struktúra működésébe, és megtanultuk, hogyan lehet a gyakorlatban létrehozni egy modern, jól működő webalkalmazást, amely hatékonyan kiszolgálja a felhasználók igényeit.

A fejlesztési folyamat egy igazi utazás volt számunkra, amely során nemcsak technikai tudásunkat bővítettük, hanem azt is felismertük, hogy a megfelelő eszközök használatával és egy összehangolt csapattal szinte bármilyen kihívást képesek vagyunk leküzdeni.

A projekt során mindketten elmélyedtünk a React alapú frontend fejlesztésben, amelyet TypeScripttel támogattunk meg, hogy a kódunk robusztusabb és hibamentesebb legyen. Dávid számára a frontend fejlesztés, a reszponzív dizájn és a Tailwind CSS alkalmazása jelentette a legnagyobb örömöt és tanulási lehetőséget, míg Chen az adatmodellezés, a PostgreSQL adatbázis-struktúrák kialakítása és a backend logika kidolgozása terén kamatoztatta tudását. Az olyan modern eszközök, mint a Vite, a Shadcn/ui és a Lucide-react használata lehetővé tette, hogy gyorsan és hatékonyan építsünk egy esztétikus, felhasználóbarát felületet. A backend oldalon a Supabase integrációja és az API-k kezelése új kihívásokat hozott, de ezek megoldása tovább növelte kompetenciáinkat.

A fejlesztői környezet kialakításában és a közös munkában kulcsfontosságú szerepet játszottak a verziókezelő eszközök, mint a Git és a GitHub, amelyekkel megtanultuk a kollaboráció alapjait, például a pull requestek kezelését és a konfliktusok feloldását. A Visual Studio Code-ban dolgoztunk, amelyet kiegészítettünk olyan bővítményekkel, mint a Prettier és az ESLint, hogy a kódolás gördülékeny és egységes legyen. Az npm csomagkezelővel gyorsan telepítettük és kezeltük a projekt függőségeit, míg a Chrome DevTools segítségével valós időben teszteltük és finomhangoltuk az alkalmazást.

Külön kiemelnénk a mesterséges intelligencia, konkrétan a ChatGPT szerepét a fejlesztési folyamatban. Ez az eszköz nélkülözhetetlen segítséget nyújtott, amikor elakadtunk, legyen szó kódolási problémákról, logikai kihívásokról vagy akár a dokumentáció pontosításáról. Ha tanácstalanok voltunk, a ChatGPT gyorsan iránymutatást adott, legyen az egy algoritmus megértése, egy hiba elhárítása vagy egy új technológia alkalmazása. Ezáltal nemcsak hatékonyabbak maradtunk, hanem produktívabbak is, hiszen ahelyett, hogy órákig keresgéltünk volna megoldásokat, célzott válaszokat kaptunk, amelyek előrébb lendítettek minket.

A közös munka során az összedolgozáshoz szükséges eszközökhöz is alkalmazkodtunk, és megtanultunk hatékonyan kommunikálni egymással. Az időnként kaotikusnak tűnő feladatok összehangolását rövid, célzott egyeztetésekkel oldottuk meg, így az időt is optimálisan ki tudtuk használni. Ez a projekt nem csupán technikai tudást adott, hanem azt is megmutatta, hogy a csapatmunka ereje a rugalmasságban, a nyílt kommunikációban és a közös problémamegoldásban rejlik. A "FitConnect" fejlesztése során szerzett tapasztalatokkal magabiztosabban állunk a jövőbeli projektek elé, és biztosak vagyunk benne, hogy a tanultakat kamatoztatva még jobb fejlesztőkké válhatunk.

#### **Saját hibáink felismerése**

Visszatekintve látjuk, hogy a tervezésnél jobban is felkészülhettünk volna – néha túl gyorsan ugrottunk a kódolásba, és emiatt később kellett visszalépnünk. A tesztelést is elhanyagoltuk az elején, ami pár kisebb hibát okozott, de ezeket gyorsan javítottuk. Ezek nem voltak súlyos problémák, és nem veszélyeztették a projekt sikerét, viszont megmutatták, hol tudunk még csiszolni a munkánkon a jövőben.

#### **Zárás**

A "FitConnect" végül egy olyan alkalmazás lett, amire büszkék vagyunk. Modernt, hasznosat alkottunk, és közben jobb fejlesztőkké váltunk. A hibáinkból tanultunk, de ezek inkább lépcsőfokok voltak, mint akadályok. Biztos vagyok benne, hogy a következő projektünkben még precízebbek és hatékonyabbak leszünk, miközben ugyanilyen lelkesedéssel dolgozunk majd együtt.

# Felhasznált irodalom

React dokumentáció (<https://reactjs.org/docs>) – 2024.04.10.

TypeScript kézikönyv (<https://www.typescriptlang.org/docs>) – 2024.04.10.

Tailwind CSS dokumentáció (<https://tailwindcss.com/docs>) – 2024.04.10.

Web Development Best Practices - MDN (<https://developer.mozilla.org>) – 2024.04.10.

**TypeScript Full Course for Beginners | Complete All-in-One Tutorial | 8 Hour** - Dave Gray: <https://youtu.be/gieEQFIfgYc?si=Ulghu9yKfgwHFDeY> - 2024.12.23.

**React Native Full 8 Hours Course (Expo, Expo Router, Supabase)** -notJust.dev:<https://youtu.be/rIYzLhkG9TA?si=eysIERkReMA4Z625> - 2024.12.10.