BGSzC Pestszentlőrinci Közgazdasági és Informatikai Szakgimnáziuma

1184 Budapest Hengersor 34.

1. Záró dolgozat

Személyi Edző Kereső

Konzulens tanár: Készítette:

név Saját név

Tartalom

[1 Bevezetés 3](#_Toc85723173)

[1.1 Feladat leírás 3](#_Toc85723174)

[1.2 A felhasznált ismeretek 3](#_Toc85723175)

[1.3 A felhasznált szoftverek 3](#_Toc85723176)

[2 Felhasználói dokumentáció 4](#_Toc85723177)

[2.1 A program általános specifikációja 4](#_Toc85723178)

[2.2 Rendszerkövetelmények 4](#_Toc85723179)

[2.2.1 Hardver követelmények 4](#_Toc85723180)

[2.2.2 Szoftver követelmények 4](#_Toc85723181)

[2.3 3. A program telepítése 4](#_Toc85723182)

[2.4 A program használatának a részletes leírása 5](#_Toc85723183)

[3 Fejlesztői dokumentáció 6](#_Toc85723184)

[3.1 Az alkalmazott fejlesztői eszközök 6](#_Toc85723185)

[3.2 Adatmodell leírása 6](#_Toc85723186)

[3.3 Részletes feladatspecifikáció, algoritmusok 6](#_Toc85723187)

[3.4 Tesztelési dokumentáció 7](#_Toc85723188)

[4 Összefoglalás 8](#_Toc85723189)

[4.1 Önértékelés 8](#_Toc85723190)

[4.2 Továbbfejlesztési lehetőségek 8](#_Toc85723191)

[5 Felhasznált irodalom 9](#_Toc85723192)

[6 Ábrajegyzék 10](#_Toc85723193)

# Bevezetés

## Feladat leírás

A szakdolgozatunk keretében egy olyan webalkalmazást fejlesztettünk, amely hidat képez a személyi edzők és a potenciális ügyfelek között. A projekt célja egy olyan platform létrehozása volt, ahol az edzők könnyedén bemutathatják szolgáltatásaikat, az ügyfelek pedig egyszerűen megtalálhatják a számukra legmegfelelőbb edzőt.

## A felhasznált ismeretek

A projekt során a következő területeken szerzett ismereteinket alkalmaztuk:

- Frontend fejlesztési alapelvek

- Modern JavaScript és TypeScript programozás

- React alapú komponens fejlesztés

- Reszponzív felhasználói felület tervezés

- CSS-in-JS és Tailwind CSS használata

- Git verziókezelés

- UI/UX tervezési alapelvek

## A felhasznált szoftverek

A fejlesztés során a következő technológiákat és eszközöket használtuk:

- React 18.3.1 - Azért ezt használjuk, mert ez egy gyors, modern frontend keretrendszer és hatékonyan tudtunk vele dolgozni.

- TypeScript - Azért ezt választottuk mert erősebben típusos mint a javascript így lehetővé teszi a hibák gyorsabb felismerését, segíti a fejlesztői élményt és alkalmas számos fejlesztői könyvtár integrálására.

- Vite - Gyors fejlesztői környezet és build eszköz.

- Tailwind CSS - Utility-first CSS keretrendszer. Rengeteg előreelkészített komponenst tartalmaz, ami felgyórsították a fejlesztési folyamatot.

- Shadcn/ui - React komponens könyvtár. Könnyen integrálható react környezetbe.

- Lucide-react - Modern ikon könyvtár

# Felhasználói dokumentáció

## A program általános specifikációja

Az alkalmazás négy fő modulból épül fel:

- Landing oldal: A platform szolgáltatásainak és előnyeinek bemutatása

- Edzők kezelése: Részletes edzői kártyák és az edzők bemutatása, keresési és részletes szűrési funkciók funkciók

- foglalási rendszer

## Rendszerkövetelmények

### Hardver követelmények

### Szoftver követelmények

## 3. A program telepítése

A fejlesztői környezet beállítása:

* Repository klónozása:  
  git clone <repository-url>
* Függőségek telepítése:  
  npm install
* Fejlesztői szerver indítása:  
  npm run dev
* 

1. kép Ez a módosító ablak a programban

Ajánlott terjedelem: 2 -4 oldal, ábrákkal együtt.

## A program használatának a részletes leírása

Az alkalmazás főbb funkciói és használatuk:

1. Edzők keresése:

- Oldal részletes bemutatása

- Időpontfoglalás

2. Edzők keresése:

- Szűrés szakterület szerint

- Részletes profil megtekintése

- Időpontfoglalás

# Fejlesztői dokumentáció

## Az alkalmazott fejlesztői eszközök

A fejlesztés során használt eszközök és környezetek:

* - Visual Studio Code - Elsődleges kódszerkesztő
* - Git - Verziókezelő rendszer
* - Node.js - JavaScript futtatókörnyezet
* - npm - Csomagkezelő
* - Chrome DevTools - Fejlesztői eszközök
* PostgreSql - Adatbázis kezelés
* 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.

## Adatmodell leírása

Az alkalmazás főbb adatstruktúrái:

interface Coach {  
 id: number;  
 name: string;  
 specialty: string;  
 rating: number;  
 reviews: number;  
 image: string;  
 price: string;  
 experience: string;  
 location: string;  
 certifications: string[];  
 availability: string;  
 activeClients: number;  
 successStories: number;  
 description: string;  
 languages: string[];  
}  
  
interface MealPlan {  
 id: number;  
 title: string;  
 description: string;  
 duration: string;  
 mealsPerDay: number;  
 price: string;  
 image: string;  
 category: string;  
}  
  
interface Post {  
 id: number;  
 author: {  
 name: string;  
 role: string;  
 image: string;  
 verified: boolean;  
 };  
 content: string;  
 image?: string;  
 likes: number;  
 comments: number;  
 time: string;  
 isCoach: boolean;  
}

## Részletes feladatspecifikáció, algoritmusok

A fejlesztés során implementált főbb komponensek és funkciók:

1. Navigációs rendszer:

- React Router alapú útvonalkezelés

- Dinamikus navigáció

- Védett útvonalak kezelése

2. Komponens architektúra:

- Újrafelhasználható UI komponensek

- Shadcn/ui integráció

- 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

## Tesztelési dokumentáció

A fejlesztés során alkalmazott tesztelési módszerek:

1. Komponens tesztek:

- Működési logika ellenőrzése

- Props validáció

- Event handling tesztelés

2. UI tesztek:

- Reszponzivitás ellenőrzése

- Különböző képernyőméretek tesztelése

- Böngésző kompatibilitás

3. Teljesítmény tesztek:

- Betöltési idők optimalizálása

- Komponens renderelés

- Memória használat

# Összefoglalás

## Önértékelés

A projekt során sikeresen:

- Implementáltunk egy modern, reszponzív webalkalmazást

- Kialakítottunk újrafelhasználható komponens struktúrát

- Megvalósítottunk komplex felhasználói interakciókat

- Optimalizáltuk a teljesítményt és felhasználói élményt

## Továbbfejlesztési lehetőségek

* Fizetési rendszer integrálása
* Mobilalkalmazás fejlesztése

# Felhasznált irodalom

1. React dokumentáció (<https://reactjs.org/docs>)
2. TypeScript kézikönyv (<https://www.typescriptlang.org/docs>)
3. Tailwind CSS dokumentáció (<https://tailwindcss.com/docs>)
4. Web Development Best Practices - MDN ([https://developer.mozilla.org](https://developer.mozilla.org/))
5. Modern JavaScript Tutorial ([https://javascript.info](https://javascript.info/))

# Ábrajegyzék

[1. kép Ez a módosító ablak a programban 5](#_Toc63250394)