

JEGYZŐKÖNYV

Szakmai Gyakorlat I.

Tervezési szakasz

Készítette: Orosz Kristóf

Neptunkód: EYZWG9

Dátum: 2025. május 30.

Orosz Kristóf

Orosz Kristóf

Hallgató

Dédesi Péter

Szakmai konzulens

Tartalomjegyzék:

Bevezetés.....	3
Fájlok tartalmának leírása.....	3
 1. Feladatrészek	4
1.1 Funkciólista összeállítása	4
1.2 Oldalszerkezet - Menüpontok tervezése.....	5
1.3 Oldalstruktúra ábra készítése.....	6
1.4 Adatbázis szerkezet tervezése	7
1.5 Adatbázis modellek rajzolása	8
1.6 Technológiai terv készítése	10
1.7 Felhasználói szerepkörök és jogosultságok tervezése	12

Bevezetés

A szakmai gyakorlatom tervezési szakaszában egy webalapú időpontfoglaló rendszer részletes előkészítésén dolgoztam, amelynek célja, hogy a felhasználók egyszerűen és átlátható módon vehessenek igénybe különböző szolgáltatásokat. A rendszer többféle szerepkört támogat: vendégek, regisztrált felhasználók és adminisztrátorok különböző jogosultságokat kapnak, így biztosítva a biztonságos és célszerű használatot. A tervezés során külön figyelmet fordítottam a funkciók strukturált dokumentálására, a felhasználói felületek felépítésére, az adatbázis logikai és fizikai modelljére, valamint a technológiai háttér megtervezésére is. Az elkészült adatmodell öt fő táblára épül, amelyek közötti kapcsolatok egyértelműen definiáltak. A REST API végpontok tervezése és az admin funkciók elkülönítése is hangsúlyt kapott. A projekt jövőbeni bővíthetősége szempontjából megfontolásra került egy értékelési funkció bevezetése is, amely M:N kapcsolatot igényelne a felhasználók és szolgáltatók között. Ez a bővítés a későbbi fejlesztési szakaszban kerülhet implementálásra, amennyiben az alapfunkciók stabilan működnek. A jelen dokumentum összefoglalja a tervezési fázis fő lépéseit és annak eredményeit.

Fájlok tartalmának leírása

- Bevezetés.pdf	[A mappa tartalmának leírása.]
- Funkciólista összeállítása.pdf	[Rendszer funkcióinak megtervezése.]
- Oldalszerkezet – Menüpontok.pdf	[HTML oldalak menüpontjainak megtervezése.]
- Oldalstruktúra ábra készítése.pdf	[Oldalstruktúra ábrázolása.]
- Adatbázis szerkezet tervezése.pdf	[Az adatbázis tábláinak részletes leírása]
- Adatbázis modellek rajzolása.pdf	[Az ER és relációs modellek vizuális ábrázolása.]
- ER modell.draw.io	[ER modell forrásfájlja.]
- RM modell.draw.io	[Relációs modell forrásfájlja.]
- Technológiai terv készítése.pdf	[Kiválasztott technológiák leírása.]
- Szerepkörök és jogosultságok.pdf	[A rendszer felhasználóinak, jogosultságainak leírása.]
- JK_Tervezési szakasz.pdf	[A teljes tervezési szakasz jegyzőkönyve.]

1.Feladatrészek

Az alábbi pontokban röviden összefoglalom, hogy a tervezési szakasz során az egyes feladatrészekhez milyen konkrét tevékenységeket végeztem.

1.1 Funkciólista összeállítása

A funkciólista összeállításakor célom az volt, hogy a teljes rendszer minden felhasználói és adminisztrátori funkcióját egyértelműen, strukturáltan rögzítsem. A dokumentumban három fő szakaszban írtam le a funkciókat: felhasználói, adminisztrátori és rendszerfejlesztési szempontból. A felhasználói funkciókhoz hozzátartozik a regisztráció, bejelentkezés, valamint az időpontok kezelése (foglalás, módosítás, törlés). Az adminisztrátorok számára külön funkciókat definiáltam a felhasználók, szolgáltatók és foglalások kezelésére.

1. Funkciók

- **Regisztrációs rendszer**
 - Felhasználók regisztrációja, bejelentkezése, kijelentkezése.
 - Adminisztrátori felület kezelése.
- **Felhasználói funkciók - Időpontfoglalás**
 - Szolgáltatók kiválasztása.
 - Időpontok keresése és megjelenítése.
 - Időpont lefoglalása, törlése, módosítása.
- **Adminisztrátori funkciók**
 - Felhasználók, szolgáltatók, foglalások kezelése.
 - Adatok, időpontok hozzáadása megtekintése, szerkesztése, törlése.

2. Backend fejlesztési szakasz

Funkciók backend megvalósítása (PHP + adatbázis + REST API):

- Felhasználók kezelése (regisztráció, bejelentkezés).
- Szolgáltatók kezelése (hozzáadás, törlés, szerkesztés, lista megjelenítése).
- Időpontfoglalás kezelése (CRUD műveletek, foglalások kezelése).
- REST API létrehozása (végpontok definiálása, dokumentálása).

3. Frontend fejlesztési szakasz

Funkciók frontend megvalósítása (HTML, CSS, JS):

- Regisztráció, bejelentkezés, kijelentkezés felületének elkészítése.

- Szolgáltató lista, részletező oldalak.
- Időpontfoglalás felületének elkészítése (időpontok választása, módosítása, törlése).
- Adminisztrátori felület frontend elkészítése.
- Egyszerű UX/UI dizájn (átláthatóság, mobilbarát kialakítás).

4. Frontend-Backend integrációs szakasz

- Frontend REST API-val való összekapcsolása (adatok küldése, fogadása).

1.2 Oldalszerkezet - Menüpontok tervezése

Megtervezésre kerültek a felhasználói, adminisztrátori és vendég szerepkörhöz tartozó menüpontok és azok funkciói. Minden menüpont mellé rövid leírás tartozik a várható viselkedésről, egyértelműsítve, hogy az adott oldal milyen célt szolgál. A tervezés eredményeként elkészült egy részletes oldalszerkezet, amely a funkciókhoz konkrét HTML fájlokat rendel hozzá. Az adminisztrátori felületek felépítése, különös tekintettel a felhasználók, szolgáltatók, időpontok és foglalások kezelésére kiemelt figyelmet fordítok. Ez a struktúra megalapozza a jogosultsági rendszer kialakítását, valamint az adminisztrátori funkciók felépítését.

1. Főoldali menüpontok

MENÜPONT	LEÍRÁS
Kezdőlap	Általános információk az oldalról.
Bejelentkezés	Belépés felhasználóként vagy adminként.
Regisztráció	Új felhasználói fiók létrehozása.
Szolgáltatók	Elérhető szolgáltatók megtekintése.

2. Felhasználói menüpontok

MENÜPONT	LEÍRÁS
Foglalás	Szabad időpontok listázása, foglalás.
Saját foglalásaim	A felhasználó aktuális foglalásai.
Profilom	Személyes adatok megtekintése.
Kijelentkezés	Kilépés a fiókból.

3. Admin menüpontok

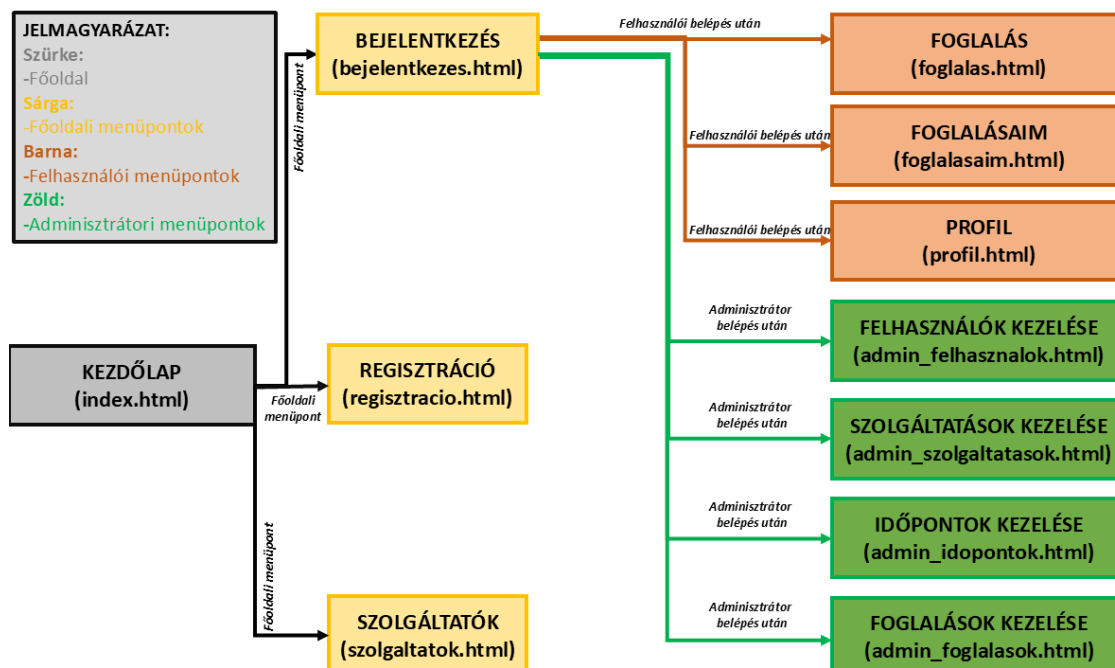
MENÜPONT	LEÍRÁS
Felhasználók kezelése	Felhasználói lista, szerkesztése, törlése.
Szolgáltatók kezelése	Új szolgáltató felvétele, módosítása, törlése.
Időpontok kezelése	Szolgáltatók időpontjainak kezelése.
Foglalások kezelése	Összes foglalás megtekintése.
Kijelentkezés	Admin kilépés.

4. Oldalszerkezet:

OLDAL	TARTALOM
index.html	Kezdőlap.
szolgaltatok.html	Szolgáltatók listája.
regisztracio.html	Regisztrációs oldal.
bejelentkezes.html	Bejelentkezési oldal.
foglalas.html	Időpontfoglalás, időpont lista.
foglalasaim.html	Felhasználói foglalások
profil.html	Felhasználói adatok megtekintése.
admin_felhasznalok.html	Admin: felhasználók kezelése.
admin_szolgaltatok.html	Admin: szolgáltatók kezelése.
admin_idopontok.html	Admin: időpontok kezelése.
admin_foglalasok.html	Admin: foglalások kezelése.

1.3 Oldalstruktúra ábra készítése

Az oldalstruktúra ábrájának elkészítése során arra törekedtem, hogy vizuálisan is átláthatóvá tegyem a webalkalmazás logikai felépítését. A főbb oldalakat (főoldal, regisztráció, bejelentkezés, szolgáltatók listája stb.) és azok közötti kapcsolódási pontokat hierarchikusan ábrázoltam. Ez a struktúra segítséget nyújt a frontend oldali navigáció és menüszerkezet tervezésében.



1.4 Adatbázis szerkezet tervezése

Ez a kezdetleges adatbázisterv részletesen tartalmazza az időpontfoglaló rendszer összes tábláját és azok mezőit. Minden mezőhöz leírtam a szerepét, típusát és az adatbázisban betöltött funkcióját. A táblák között szerepel a felhasználók, szolgáltatók, időpontok, foglalások és szolgáltatástípusok táblája. A mezőkhöz hozzárendeltem az elsődleges kulcsokat, a külső kulcsokat. Táblázatosan is bemutattam az egyes entitások közötti kapcsolatokat. Rögzítésre került, hogy mely táblák között van 1:N vagy 1:1 kapcsolat, és milyen mezők szolgálnak összekötő kulcsként.

Egyedek:

Felhasznalok [id PK, nev, szerepkor, email, jelszo, reg_datum]

Szolgáltatok [id PK, Szolgáltatás_típusok_id FK, nev, leiras, aktiv]

Idopontok [id PK, Szolgáltatok_id FK, datum, ido, foglalható]

Foglalások [id PK, Felhasznalok_id FK, Idopontok_id FK, foglalasi_ido, állapot, megjegyzes]

Szolgáltatás_típusok [id PK, nev, leiras]

Kapcsolatok:

TÁBLA 1	TÁBLA 2	KAPCSOLAT TÍPUSA	KAPCSOLAT NÉV	LEÍRÁS
Felhasznalok	Foglalások	1:N	F-F	Egy felhasználónak több foglalása lehet.
Szolgáltatok	Idopontok	1:N	SZ-I	Egy szolgáltató több időpontot kínálhat.
Foglalások	Idopontok	1:1	I-F	Egy időpont egy foglaláshoz tartozik.
Szolgáltatás_típusok	Szolgáltatok	1:N	SZ_T-SZ	Egy típushoz több szolgáltató tartozhat.

Táblák, mezők leírásának magyarázata:

Felhasznalok

MEZŐ	SZEREPE
id	Azonosító, elsődleges kulcs. (PK)
nev	Teljes név.
szerepkor	Felhasználók és adminok megkülönböztetése.
email	Email cím.
jelszo	Jelszó a belépéshez.
reg_datum	Regisztráció dátuma.

Szolgáltatók

MEZŐ	SZEREPE
id	Szolgáltató azonosítója. (PK)
Szolgáltatás_tipusok_id	Szolgáltatás típusa. (FK)
nev	Szolgáltató neve.
leiras	Rövid leírás a szolgáltatásról.
aktiv	Aktív-e a szolgáltató.

Időpontok

MEZŐ	SZEREPE
id	Egyedi időpont-azonosító. (PK)
Szolgáltatók_id	Kapcsolat a szolgáltatóhoz (FK)
datum	Az időpont napja.
ido	Óra-perc formátumú idő.
foglalható	Szabad-e az időpont.

Foglalások

MEZŐ	SZEREPE
id	Foglalás azonosító (PK)
Felhasználók_id	Ki foglalt. (Kapcsolat a 'felhasználók' táblához – FK)
Időpontok_id	Melyik időpontra vonatkozik. (Kapcsolat az 'időpontok' táblához – FK)
foglalási_idő	Mikor történt a foglalás.
allapot	Lefoglalt, lemondva, teljesítve.
megjegyzés	Opcionális megjegyzés a foglaláshoz.

Szolgáltatás típusok

MEZŐ	SZEREPE
id	Azonosító, elsődleges kulcs. (PK)
nev	A szolgáltatástípus megnevezése.
leiras	Rövid leírás a típusról.

1.5 Adatbázis modellek rajzolása

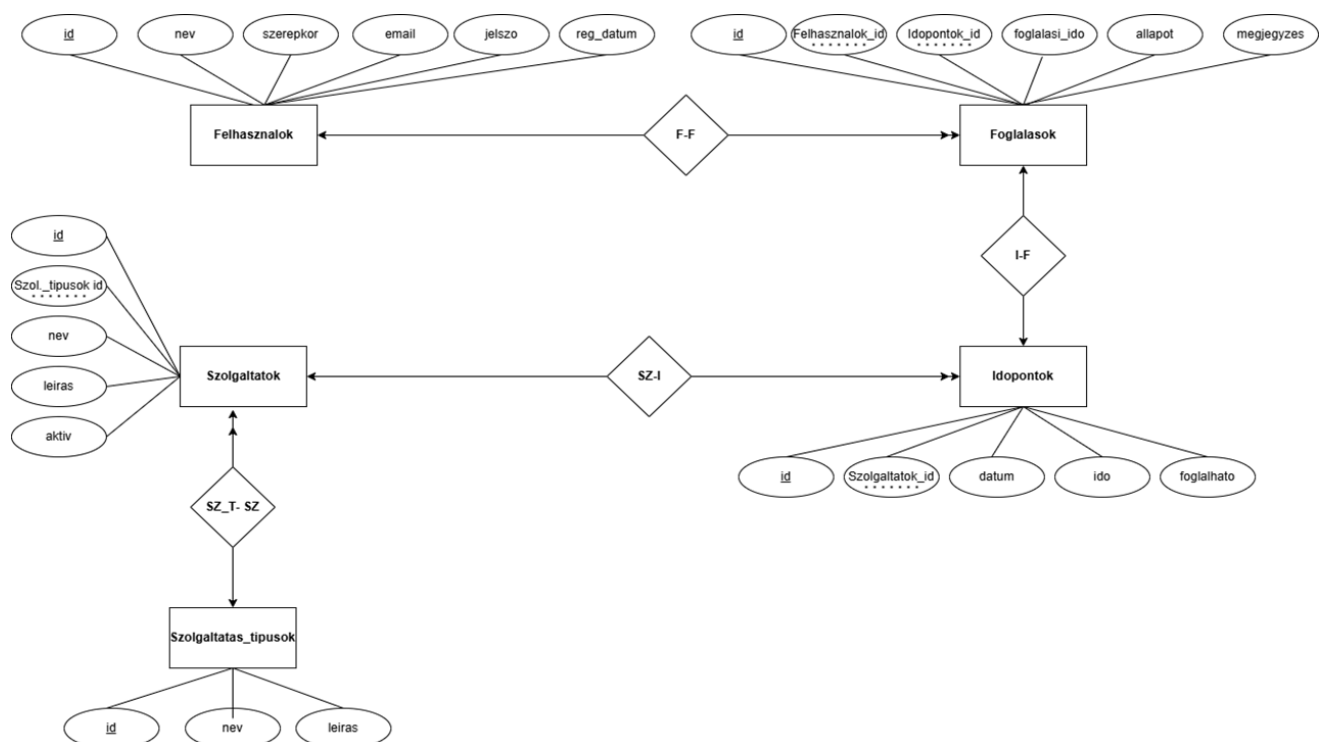
E feladat során vizuálisan ábrázoltam az időpontfoglaló rendszer ER és relációs modelljét. Az ER diagram öt egyedet tartalmaz: felhasználók, szolgáltatók, időpontok, foglalások és szolgáltatástípusok. A modellezés során az egyedek közötti kapcsolattípusokat is feltüntettem, például 1:N és 1:1 kapcsolat formájában. A relációs modell már tartalmazza az adattípusokat,

A mezők mellett megadtam az elsődleges kulcsokat (PK) és az idegen kulcsokat (FK). Amennyiben stabil lesz az alrendszerem, sikerül a backenddel megfelelően összekapcsolni adatbázisom és még marad időm, később bővíteni tervezem adatbázisom értékelési funkcióval, amely M:N kapcsolatot valósítana meg a felhasználók és szolgáltatók között.

ER modell rajzolása

Az ER diagramon öt fő egyed szerepel: **felhasználók**, **szolgáltatók**, **időpontok**, **foglalások**, valamint **szolgáltatástípusok**.

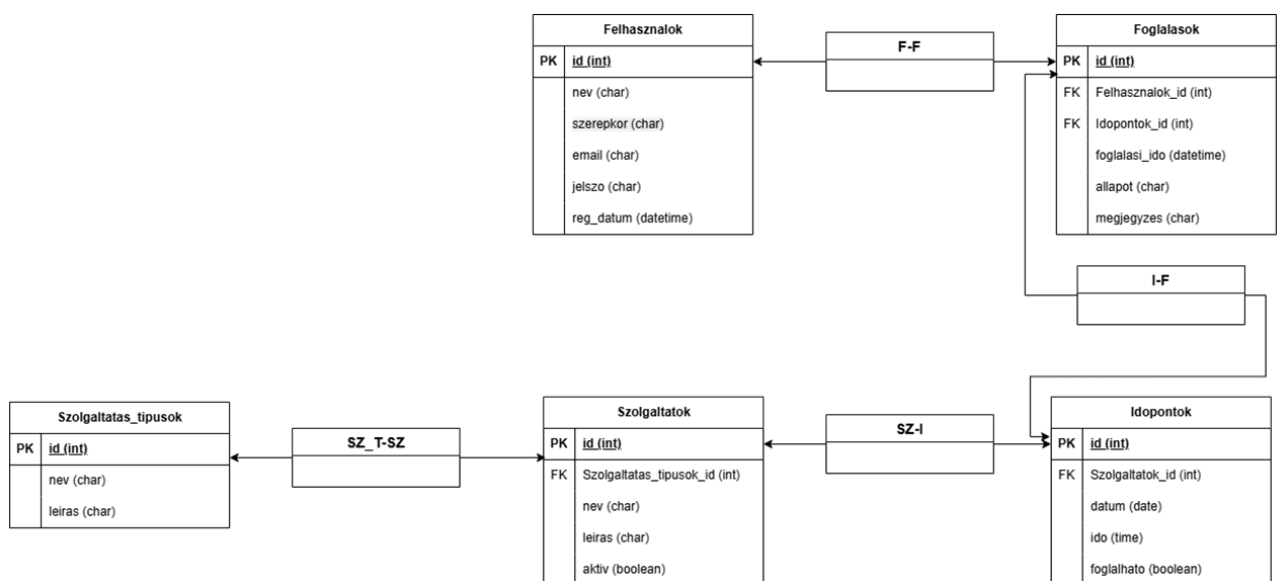
A felhasználók és a foglalások között **1:N kapcsolat** áll fenn, mivel egy felhasználónak több foglalása is lehet. A szolgáltatók és időpontok szintén **1:N kapcsolatban** állnak egymással, hiszen egy szolgáltató több időpontot is kínálhat. Az időpontok és foglalások között **1:1 kapcsolatot** határoztam meg, mivel egy időpont csak egyszer foglalható. A szolgáltatástípusok és szolgáltatók között pedig szintén **1:N kapcsolat** jött létre, amely lehetővé teszi, hogy több szolgáltató ugyanazon típushoz tartozzon.



Relációs modell rajzolása

Az elkészített relációs modell a rendszer teljes adatbázisának logikai szerkezetét mutatja be, az egyedek és azok kapcsolatai alapján. A relációs modellen jelölve vannak az **elsődleges kulcsok (PK)**, a **idegen kulcsok (FK)**, valamint a mezőtípusok. A fő táblák között szerepelnek a **Felhasznalok**, **Foglalasok**, **Idopontok**, **Szolgáltatok** és **Szolgáltatás_típusok**. A relációs modell esetében minden kapcsolat külső kulcsokon keresztül történik.

A **Foglalasok** tábla például egyszerre kapcsolódik a **Felhasznalok** és az **Idopontok** táblához, így modellezi, hogy mely felhasználó melyik időpontot foglalta le. A **Szolgáltatok** tábla a **Szolgáltatás_típusok** táblához kapcsolódik, lehetővé téve a szolgáltatók kategorizálását.



1.6 Technológiai terv készítése

A technológiai terv során kiválasztottam azokat az eszközöket és nyelveket, amelyekkel a webes alkalmazás fejlesztése történik. A frontend oldalon HTML-t használok az oldalak strukturált felépítéséhez, valamint CSS-t a megjelenés és elrendezés kialakításához. A dinamikus működést JavaScript biztosítja, különösen a felhasználói interakciók és API-hívások terén. A backend fejlesztését PHP nyelven valósítom meg, amely alkalmas REST API-k létrehozására és XAMPP környezetben jól működik. Az adatbáziskezelésre MariaDB-t használok, mivel ez kompatibilis a MySQL-lel és a XAMPP része. Biztonsági tartalékként felkészültem az Oracle APEX használatára is, amennyiben a MariaDB környezet nem működne megfelelően.

TECHNOLÓGIA	HASZNÁLATI TERÜLET	LEÍRÁS
HTML	Frontend	A HTML-t használom az oldalak strukturált felépítéséhez és alapvázának kialakításához.
CSS	Frontend	A megjelenést, stílust, színeket és az oldal reszponzív elrendezését CSS segítségével szabályozom.
JavaScript	Frontend	A felhasználói interakciókat, az adatok betöltését és a fetch-alapú API-kommunikációt JavaScript segítségével valósítom meg.
PHP	Backend	A szerveroldali működést PHP nyelven fejlesztem, ezzel valósítom meg a REST API végpontokat.
JSON	Frontend - Backend	A frontend és a backend közötti adatkommunikáció JSON formátumon keresztül valósítom meg. A JavaScript fetch() metódus POST vagy GET kéréseket küld a szerver felé, amely PHP nyelven feldolgozza az adatokat, majd JSON formátumban válaszol.
MariaDB	Adatbázis	Elsődlegesen MariaDB-t használok az adatbáziskezeléshez, amely MySQL-kompatibilis és része a XAMPP csomagnak.
Oracle APEX	Adatbázis	Amennyiben a MariaDB környezet valamilyen okból nem működik megfelelően, Oracle APEX környezetet fogom használni.
Postman	API tesztelés	A készülő REST API végpontokat Postman segítségével tesztelem és hibakeresem.
XAMPP	Fejlesztői környezet	XAMPP-ot használok helyi Apache, PHP és MariaDB környezetként.
Visual Studio Code	Fejlesztői környezet	A fejlesztést Visual Studio Code-ban végzem, bővítmények segítségével.
GitHub	Verziókezelés	A projekt kódját GitHub-on tárolom, így nyomon követem a változásokat és biztonságosan mentem a forráskódokat.

1.7 Felhasználói szerepkörök és jogosultságok tervezése

A rendszerben három fő felhasználói szerepkört különböztettem meg: **vendég**, **regisztrált felhasználó** és **admin**.

A **vendég** szerepkörhöz nem tartozik bejelentkezés, így csak a főoldal, a szolgáltatók listája, valamint a regisztráció és bejelentkezés érhető el. Vendégként nem engedélyezek hozzáférést a foglalási rendszerhez és a profiladatokhoz.

A **regisztrált felhasználók** már bejelentkezhetnek, időpontot foglalhatnak, illetve megtekinthetik saját profiljukat és foglalásaikat. Ők nem férhetnek hozzá más felhasználók adataihoz és az admin funkciókhoz.

Az **adminisztrátor** szerepkörrel rendelkező felhasználók teljes jogosultságot élveznek: kezelhetik az összes rendszerkomponenst, beleértve a felhasználókat, szolgáltatókat, időpontokat és foglalásokat.

A szerepkörök technikai ellenőrzése minden REST API hívásnál backend oldalon történik, a szerepkör mező értéke alapján.

SZEREPKÖR	ELÉRHETŐ FUNKCIÓK	KORLÁTOZÁSOK	TECHNIKAI KEZELÉS
Vendég	A vendégek számára elérhető a kezdőlap, a szolgáltatók listája, valamint a regisztrációs és bejelentkezési lehetőség.	Vendégként nem biztosítok hozzáférést a foglalási rendszerhez, és nem jelenítem meg számukra a profilhoz tartozó adatokat.	Vendégként nincs bejelentkezett állapot, az eléréseket backend oldalon ellenőrzöm.
Regisztrált felhasználó	A regisztrált felhasználók időpontot foglalhatnak, megtekinthetik saját profiljukat és foglalásaikat.	Nem engedem számukra más felhasználók adatainak vagy az admin funkciók elérését.	A bejelentkezett állapotban szereplő felhasználóknál a 'szerepkör' mező értéke 'felhasznalo', ezt minden REST API hívásnál figyelembe veszem.
Adminisztrátor	Adminisztrátorként teljes jogosultságot elérhető: Kezelhetőek a felhasználók, szolgáltatók, időpontok, foglalások, valamint hozzáférés a szolgáltatástípusokhoz is.	Az adminisztrátor szerepkör nem tartalmaz korlátozásokat, minden rendszerfunkció elérhető.	Az admin felhasználókat a 'felhasznalok' táblában szereplő 'admin' érték azonosítja a 'szerepkör' mezőben, és ezt backend oldalon szűröm.