# Tartalomjegyzék

[Tartalomjegyzék 1](#_Toc190513538)

[Bevezetés 2](#_Toc190513539)

[Témaválasztás 2](#_Toc190513540)

[Tervezés, drótvázak 4](#_Toc190513541)

[Felhasználói dokumentáció 5](#_Toc190513542)

[A program célja és lényegesebb funkciói 5](#_Toc190513543)

[Futtatási környezet 5](#_Toc190513544)

[A program fő funkcióinak leírása 6](#_Toc190513545)

[Fejlesztői dokumentáció 8](#_Toc190513546)

[Fejlesztői környezet 8](#_Toc190513547)

[Program struktúra 8](#_Toc190513548)

[Adatbázis 8](#_Toc190513549)

[Leírás 8](#_Toc190513550)

[Funkciók 9](#_Toc190513551)

[Szerkezet 9](#_Toc190513552)

[Egyed kapcsolat diagram 10](#_Toc190513553)

[Kapcsolati ábra 10](#_Toc190513554)

[Táblák szerkezete a tábla szintű megszorításokkal 11](#_Toc190513555)

[Fogalmak 11](#_Toc190513556)

[Hasznos lekérdezések 11](#_Toc190513557)

[Beépítendő megszorítások 11](#_Toc190513558)

[Megvalósítás 11](#_Toc190513559)

[Scriptek 11](#_Toc190513560)

[Teszt adatok 11](#_Toc190513561)

[Továbbfejlesztési ötletek 11](#_Toc190513562)

[Backend 11](#_Toc190513563)

[Modellek és Controllerek 11](#_Toc190513564)

[API végpontok 11](#_Toc190513565)

[Frontend 12](#_Toc190513566)

[Reszponzivitás 12](#_Toc190513567)

[Tesztelés 13](#_Toc190513568)

[Fejlesztési lehetőségek 14](#_Toc190513569)

[Irodalomjegyzék 15](#_Toc190513570)

# Bevezetés

## Témaválasztás

A szakdolgozat témájának a magánklinika online foglalási rendszer fejlesztését választottuk, mert ez egy valós problémára ad digitális megoldást, illetve javíthatja a klinikák működését és a páciensek élményét.

Az egészségügyi szektor egyre inkább digitalizálódik, és egy hatékony, felhasználóbarát foglalási rendszer nagy előnyt jelenthet a magánklinikák számára. Az emberek hozzászoktak az online időpontfoglaláshoz, és elvárják, hogy gyorsan és kényelmesen intézhessék ügyeiket.

Szakdolgozatunk egy valós életben is használható rendszert mutat be, amelyet egy valódi klinika is bevezethetne.

A projekt során többféle technológiát kell alkalmazni, például:

* Backend fejlesztés (adatbázis-kezelés, API-k, autentikáció)
* Frontend fejlesztés (felhasználóbarát UI, reszponzivitás)
* Mobilbarát megoldások (mivel a páciensek sok esetben mobilról foglalnak)
* Tesztelési folyamatok (automatizált és manuális tesztelés a megbízhatóság érdekében)

Ezek hozzájárulnak ahhoz, hogy egy komplex, fejlesztői és tesztelői kihívást oldjunk meg.

# Feladat specifikáció

## A program célja és lényegesebb funkciói

A feladat célja egy orvos-páciens menedzsment rendszer létrehozása, amely lehetőséget biztosít a látogatóknak a regisztrációra, az orvosi vizsgálatok közötti keresésre, valamint a vizsgálatok időpontfoglalására és kezelésére.

A páciensek bejelentkezés után egyszerűen megtekinthetik, módosíthatják és lemondhatják a jövőbeni időpontjaikat, valamint kiválaszthatják a szabad időpontok közül azokat, amelyek az általuk kívánt eljárásokhoz elérhetőek, tehát a páciens egy szakon belül lévő vizsgálatra kiválaszt egy szabad időpontot, és látja a hozzá tartozó orvos(oka)t. Az orvos kiválasztását követően le tudja foglalni a vizsgálatot.

Az orvosok saját profiljukon keresztül kezelhetik rendelési idejüket és nyomon követhetik a páciensek értékeléseit. Egy orvoshoz egy szak tartozhat.

Az adminisztrátorok hozzáféréssel rendelkeznek a felhasználói nem szenzitív adatokhoz és a foglalások kezeléséhez, emellett orvosokat regisztrálhatnak.

## Futtatási környezet

Az ötletelés fázisában gyorsan felmerült, hogy időpontfoglaló applikáció készítése illeszkedik leginkább az elképzelésekhez, amelyet webalkalmazás formájában kívánunk véghez vinni. Ennek oka, hogy kiemelten fontosnak tartottuk a felhasználóbarátságot, és résztvevők sokrétűsége miatti általános megtekinthetőséget. Ezáltal telefonon, vagy nagy képernyő előtt számítógépen ugyanúgy megtekinthető a felület külön alkalmazás telepítése nélkül, amely bizonyos felhasználók esetén nehézséget okozna. Emellett fontosnak tartottuk a platform független működést, így a felhasználó számára is egyszerűbb egy weboldal megtekintése, mint egy alkalmazás futtatása (esetlegesen frissítést követően), emellett nem követel meg átlagon felüli sávszélességet. A megtekintésre a webböngészők friss verziói a legalkalmasabbak (Microsoft Edge, Google Chrome, Mozilla Firefox, Opera).

## A program fő funkcióinak leírása

A program fő célja:

A szakdolgozat keretében fejlesztett rendszer egy magánklinika online időpontfoglaló és menedzsment rendszerének létrehozása, amelynek célja, hogy egyszerűsítse és digitalizálja a klinika működésével kapcsolatos legfontosabb folyamatokat.

A rendszer fő funkciói a következők:

1. Könnyű és gyors időpontfoglalás
   * A páciensek online, néhány kattintással foglalhatnak időpontot a megfelelő szakterület orvosához.
   * Az elérhető időpontokat valós időben frissíti a rendszer.
2. Orvosok és adminisztrátorok munkájának megkönnyítése
   * Az orvosok megadhatják rendelési idejüket, szabadságukat, és kezelhetik a foglalásokat.
   * Az adminisztrátorok szükség esetén módosíthatják az időpontokat és kezelhetik a felhasználókat.
3. Felhasználóbarát és mobilbarát felület
   * A rendszer reszponzív dizájnnal készül, így asztali gépen és mobiltelefonon is könnyen használható.
   * Letisztult és intuitív felület biztosítja a gördülékeny navigációt.
4. Értesítések és emlékeztetők
   * A páciensek automatikus értesítést kapnak foglalásukról és annak közelgő időpontjáról.
   * Az értesítések csökkentik az elfelejtett időpontok számát, optimalizálva az orvosok időbeosztását.
5. Biztonságos adatkezelés és jogosultságkezelés
   * A rendszer különböző szerepköröket használ (páciens, orvos, admin), hogy mindenki csak a számára releváns adatokhoz férjen hozzá.
   * Az érzékeny egészségügyi adatok védelméről modern titkosítási és biztonsági eljárások gondoskodnak.
6. Statisztikák és jelentések
   * Az adminisztrátorok és orvosok betekintést nyerhetnek a foglalási statisztikákba, például a leggyakoribb szolgáltatásokba vagy a kihasználtsági arányokba.
   * Az adatelemzés segíthet a klinika működésének optimalizálásában.

# Drótvázak

# Tervezés

## Fejlesztői környezet

A létrehozott magánklinika alkalmazás szerver-kliens alapon működik, segítve az erőforrás optimálisabb használatát. Adatbázis szempontjából Microsoft SQL Servert alkalmaztunk a tervezés során. A kivitelezés MySQL segítségével történt. Szerver oldalon PHP programozási nyelvet használtunk, illetve RESTful mentén Laravel keretrendszert használtunk. Kliens oldali megvalósítás Javascript segítségével történt React használatával. A fejlesztést Visual Studio Code-ban végeztünk, verziókövetésre GitHub-ot használtunk.

## Adatbázis

### Leírás

A Magánklinika adatbázis a klinika betegfoglalásait, orvosi szakrendeléseit és felhasználóit kezeli.

Az adatbázisban kezeljük a felhasználók adatait, akik szerepkörök alapján vannak besorolva. Ezen szerepkörök határozzák meg a rendszerbeli jogosultságaikat, amik 3 részre válnak szét. Lehetnek páciensek, orvosok vagy adminisztrátorok.

A tárolt felhasználók azon adatait, amik megegyeznek az orvosok és páciensek esetén, egy közös táblában kezelnénk. A nem közös adatokat pedig külön az orvosoknak és pácienseknek létrehozandó táblában tárolnánk. Az adminisztrátoroknak nem tervezünk külön táblát, mivel nekik speciális adatra nincs szükség, ami a felhasználóknak létrehozandó táblában nem tárolható.

Minden orvos rendelkezik egy adott szakosodással, amit későbbiekben külön táblában szeretnénk tárolni.

A rendszerben a kezelések és azok időtartama is nyilvántartva kell, hogy legyen. Minden kezeléshez tartoznia kell egy adott szakosodásnak, amely biztosítja, hogy csak megfelelő szaktudással rendelkező orvosok végezhetik el az adott beavatkozásokat.

Mivel a páciensek időpontokat szeretnének foglalni az orvosokhoz, ezért ezen időpontokat szeretnénk táblában tárolni. A beazonosíthatóság miatt egy foglalás tartalmazza az orvos azonosítóját, a beteg azonosítóját, a foglalás kezdeti időpontját, a kiválasztott kezelést és annak állapotát. Az időpontokat az orvosok szabad kapacitásai alapján lehet lefoglalni.

A rendszerben szeretnénk lehetőséget biztosítani az orvosok értékelésére is, amely segíti a pácienseket a megfelelő szakember kiválasztásában.

### Funkciók

- Felhasználók (páciensek, orvosok, adminisztrátorok) felvétele, módosítása, törlése

- Páciensek időpontfoglalása, módosítása, törlése

- Orvosi szakrendelések kezelése

- Kezelések nyilvántartása, módosítása, törlése

- Orvosok szakosodásának kezelése

- Navigációs menüpontok és jogosultságok kezelése

- Adminisztrátori műveletek: felhasználók, kezelések és foglalások menedzselése

- Orvosi értékelések kezelése és megjelenítése

- Statisztikák és riportok generálása

### Szerkezet

Jelölés:   
Tábla magyar neve (TÉNYLEGES TÁBLA NEVE)  
(kulcs, leíró, *kapcsolathordozó*)  
Rövid leírás

#### Funkcionális táblák:

Felhasználói szerepkör (ROLE)

{role\_id, name}  
A rendszerben minden felhasználó egy meghatározott szerepkörbe tartozik, amely meghatározza a jogosultságait. Például az adminisztrátorok kezelhetik a felhasználókat, míg az orvosok csak a saját időpontjaikat láthatják és módosíthatják. A szerepkörök előre definiáltak, és minden egyes felhasználónak pontosan egy szerepkör van hozzárendelve.

Felhasználó (USER)  
{id, name, phone\_number, email, email\_verified\_at, password, *role\_id*, remember\_token}

Ez a tábla tárolja a rendszer összes regisztrált felhasználóját, függetlenül attól, hogy páciensekről, orvosokról vagy adminisztrátorokról van szó. Minden felhasználónak meg kell adnia a személyes adatait, beleértve az elérhetőségeit és egy biztonságos jelszót. A rendszer biztosítja, hogy egy felhasználónak mindig legyen érvényes szerepköre.

Páciens (PATIENT)  
{user\_id, taj\_number, birth\_date, address}  
Ez a tábla tartalmazza a klinikára regisztrált páciensek adatait. Minden páciens rendelkezik egy egyedi azonosítóval, amelyet a rendszerben való azonosításra használunk. Az egészségügyi ellátás biztosítása érdekében itt tároljuk a társadalombiztosítási számukat, a születési dátumukat és lakcímüket is.

Orvosi szakterület (SPECIALIZATION)  
{specialization\_id, specialization\_name}  
A klinikán dolgozó orvosok különböző szakterületekre specializálódnak, például belgyógyászat, sebészet vagy ortopédia. Ebben a táblában vannak felsorolva ezek a szakterületek.

Orvos (DOCTOR)  
{user\_id, *specialization\_id*}  
Az orvosok adatai ebben a táblában vannak tárolva. Minden orvos egyedi azonosítóval rendelkezik, amely a rendszer többi táblájában is használható azonosításra. Az orvosok csak olyan kezeléseket végezhetnek, amelyek a saját szakterületükhöz tartoznak.

Kezelés (TREATMENT)  
{treatment\_id, *specialization\_id*, treatment\_name, treatment\_length, price}  
A klinika által nyújtott különböző kezelések listája ebben a táblában található. Minden kezeléshez meghatározott időtartam és ár tartozik, valamint egy szakirány, amely biztosítja, hogy csak a megfelelő szaktudással rendelkező orvosok végezhessék el az adott beavatkozást.

Orvosi időpontfoglalás (DOCTOR\_APPOINTMENT)  
{*doctor\_id*, start\_time, *user\_id*, *treatment\_id*, status, rating}

A páciensek a rendszerben foglalhatnak időpontokat az orvosokhoz. Minden foglalás tartalmazza az érintett orvos és páciens azonosítóját, valamint az időpont kezdetét. A páciens azonosítója akkor kerül a táblába, ha a foglalás megtörténik, ezzel a státusza „b” (foglalt) lesz. Az időpont státusza lehet „v” (szabad), „b” (foglalt), „d” (elvégzett), „c” (orvos által törölt) vagy „p” (páciens által törölt). A rendszer lehetőséget biztosít a kezelések értékelésére is, így a páciensek visszajelzést adhatnak az ellátás minőségéről.

### Funkcionális táblák egyed kapcsolat diagram

A képen szöveg, diagram, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

### Technikai táblák leírása:

A (NAVIGATION) és (NAVIGATION\_ROLE) táblák kezelik a rendszer menüpontjait és azt, hogy egy adott szerepkör milyen felületeket érhet el. Az adminisztrátorok jogosultak kezelni a felhasználókat, orvosokat, kezeléseket és a páciensek foglalásait.

### Technikai táblák szerkezete:

Navigációs menüpont (NAVIGATION)  
{navigation\_id, name, URL, component\_name, parent}

A rendszer különböző funkcióit menüpontokon keresztül érhetik el a felhasználók. Minden menüpontnak van egy neve és egy URL-címe, amely meghatározza, hogy melyik oldalra vezet. Egyes menüpontok önállóan léteznek, míg mások egy főmenü alá tartoznak.

Menüpont-hozzáférési jogosultság (NAVIGATION\_ROLE)

{navigationRole\_id, ranking, *role\_id*, *navigation\_id*}

Nem minden menüpont érhető el minden felhasználói szerepkör számára. Ebben a táblában van meghatározva, hogy mely szerepkörök milyen menüpontokat láthatnak és használhatnak. Így például a páciensek nem férhetnek hozzá az adminisztrációs funkciókhoz, míg az adminisztrátorok teljes hozzáféréssel rendelkeznek.

### Technikai táblák egyed kapcsolat diagram:

A képen szöveg, képernyőkép, Betűtípus, sor látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A felhasználói szerepkör tábla ugyan nem technikai, de közvetlen kacsolatban van a menüpont-hozzáérési jogosultság táblával. Az áttekinthetőség érdekében a Felhasználói szerepkör táblát is szerepeltetjük ezen ábrán.

### Kapcsolati ábra:

A képen szöveg, szám, Betűtípus, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

### Táblák szerkezete a tábla szintű megszorításokkal

#### Funkcionális táblák:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SZEREP (ROLE)** | | | | |
| Mező magyarul | Mező valós | Adattípus | Szerep | Korlátozás |
| jogosultság azonosító | role\_id | bigint(20) | K |  |
| név | name | varchar(255) | kötelező, egyedi |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FELHASZNÁLÓ (USER)** | | | | |
| Mező magyarul | Mező valós | Adattípus | Szerep | Korlátozás |
| azonosító | id | bigint(20) | K |  |
| név | name | varchar(255) | kötelező |  |
| telefonszám | phone\_number | varchar(255) | kötelező, egyedi |  |
| email | email | varchar(255) | kötelező, egyedi |  |
| jelszó | password | varchar(255) | kötelező |  |
| jogosultság azonosító | role\_id | bigint(20) | KK(Role) | role\_id > 0 & role\_id < role tála méret |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SPECIALIZÁCIÓ (SPECIALIZATION)** | | | | |
| Mező magyarul | Mező valós | Adattípus | Szerep | Korlátozás |
| specializáció azonosító | specialization\_id | bigint(20) | K |  |
| specializáció név | specializaton\_name | varchar(255) | kötelező, egyedi |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ORVOS (DOCTOR)** | | | | |
| Mező magyarul | Mező valós | Adattípus | Szerep | Korlátozás |
| felhasználó azonosító | user\_id | bigint(20) | K, KK (USER) |  |
| specializáció azonosító | specialization\_id | bigint(20) | KK (SPECIALIZATION) |  |

További megszorítások:

* Egy orvos csak egy szakkal rendelkezhez

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PÁCIENS (PATIENT)** | | | | |
| Mező magyarul | Mező valós | Adattípus | Szerep | Korlátozás |
| felhasználó azonosító | user\_id | bigint(20) | K, KK (USER) |  |
| tajszám | taj\_number | int(11) | kötelező, egyedi | taj\_number > 0 |
| születési idő | birth\_date | date | kötelező | birth\_date > mai dátum |
| cím | address | varchar(255) | kötelező |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KEZELÉS (TREATMENT)** | | | | |
| Mező magyarul | Mező valós | Adattípus | Szerep | Korlátozás |
| kezelés azonosító | treatment\_id | bigint(20) | K |  |
| specializáció azonosító | specialization\_id | bigint(20) | KK (SPECIALIZATION) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ORVOSI IDŐPONTFOGLALÁSOK (DOCTOR\_APPOINTMENT)** | | | | |
| Mező magyarul | Mező valós | Adattípus | Szerep | Korlátozás |
| orvos azonosító | doctor\_id | bigint(20) | K, KK része (DOCTOR) |  |
| kezdés ideje | start\_time | datetime | K | kezdés idő > mai nap |
| páciens azonosító | patient\_id | bigint(20) | KK (PATIENT) |  |
| kezelés azonosító | treatment\_id | bigint(20) | KK (TREATMENT) |  |
| státusz | status | enum | default vacant |  |
| értékelés | rating | int(11) |  | rating > 0 & rating < 6 |

További megszorítások:

* ugyanarra a kezelésre adott páciensnek egy 'booked'(foglalt) státuszú kezelése legyen
* ha a kezelés státusza nem "d", akkor nem lehetséges értékelni
* ugyanazon kezelésre adott páciens csak egy értékelést tudjon adni

#### Technikai táblák:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAVIGÁCIÓ (NAVIGATION)** | | | | |
| Mező magyarul | Mező valós | Adattípus | Szerep | Korlátozás |
| navigáció azonosító | navigation\_id | bigint(20) | K |  |
| megnevezés | name | varchar(255) | kötelező |  |
| URL | URL | varchar(255) | kötelező |  |
| komponens neve | component\_name | varchar(255) | kötelező |  |
| szülő | parent | int(11) |  |  |

További megszorítások:

* hurokvizsgálat (parent)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAVIGÁCIÓS SZEREPKÖRÖK (NAVIGATION\_ROLE)** | | | | |
| Mező magyarul | Mező valós | Adattípus | Szerep | Korlátozás |
| navigációs szerepkör azonosító | navigationRole\_id | bigint(20) | K |  |
| sorszám | ranking | int(11) | kötelező | ranking > 0 |
| jogosultság azonosító | role\_id | bigint(20) | KK (ROLE) |  |
| navigáció azonosító | navigation\_id | bigint(20) | KK (NAVIGATION) |  |

További megszorítások:

* szerepkörönként nem ismétlődhet a sorszám

### Hasznos lekérdezések

1. Adott orvosnak hány elvégzett kezelése volt az előző hónapban?

Ez a lekérdezés egy adott orvos által az előző hónapban végzett sikeres kezelések számát adja vissza. Csak azokat az időpontokat veszi figyelembe, amelyek sikeresen lezajlottak (status = 'done'). Az időintervallum meghatározására a DATEADD függvényeket használja, így mindig az aktuális hónaphoz képest előző hónapot vizsgálja.

1. Adott Páciensnek hány kezelése volt az előző hónapban?

Ez a lekérdezés egy adott páciens számára az előző hónapban elvégzett kezelések számát határozza meg. A páciens egyedi TAJ-száma alapján keres, és csak a lezárt, sikeres kezeléseket számolja (status = 'done'). Az időtartomány az előző hónap teljes időszakára vonatkozik.

1. Hány Páciensnek volt elvégzett kezelése?

Ez a lekérdezés azt határozza meg, hogy hány különböző páciens vett részt legalább egy lezárt (status = 'done') kezelésen. A COUNT(DISTINCT) operátor biztosítja, hogy minden páciens csak egyszer szerepeljen a számlálásban, még akkor is, ha több kezelése volt.

1. Adott orvosnak hány törölt időpontja volt az adott hónapban/évben?

Ez a lekérdezés egy adott orvos által törölt időpontok számát adja vissza egy adott hónapban vagy évben. A törölt időpontokat a status mezőben tárolt adatok alapján lehet azonosítani (status = 'cancelled'). A lekérdezés szűrhető egy adott hónapra vagy évre, így az orvos munkájának elemzésében segíthet.

1. Melyik három orvos kapta a legtöbb értékelést tavaly?

Ez a lekérdezés megmutatja, hogy melyik három orvos kapta a legtöbb visszajelzést (rating IS NOT NULL) az előző évben. Az orvosok értékelései az appointments táblában találhatók, és a COUNT(\*) segítségével összesítjük, hány értékelés érkezett az adott orvoshoz. Az eredmény csökkenő sorrendben jelenik meg, hogy a legtöbb értékelést kapott orvosok kerüljenek előre.

1. Melyik három orvos kapta a legrosszabb értékelést tavaly?

Ez a lekérdezés az előző év legrosszabbul értékelt három orvosát azonosítja. Az orvosok értékeléseinek átlaga (AVG(da.rating)) alapján sorba rendezi az orvosokat növekvő sorrendben, így az első három eredmény a leggyengébben értékelt orvosokat mutatja.

1. Mely userek nem lépett be a rendszerbe több mint 3 éve?

Ez a lekérdezés azokat a felhasználókat keresi, akik legalább három éve nem jelentkeztek be a rendszerbe. Az utolsó bejelentkezési dátum (last\_login) alapján ellenőrzi, hogy a dátum több mint három évvel ezelőtti-e. Ez segíthet az inaktív felhasználók kiszűrésében vagy figyelmeztetés küldésében.

1. Mely szakon történt a legtöbb elvégzett vizsgálat/kezelés?

Ez a lekérdezés azonosítja, hogy melyik orvosi szakterületen végezték a legtöbb sikeres kezelést (status = 'done'). Az egyes kezelésekhez kapcsolódó szakterületeket (specialisation\_name) összesíti, és az eredményt csökkenő sorrendben rendezi, hogy a legnépszerűbb szakterület jelenjen meg elsőként.

1. Mely páciensek mondták le az időpontjaikat legalább 10 alkalommal tavaly?

Ez a lekérdezés azoknak a pácienseknek a listáját adja vissza, akik az előző évben legalább tíz alkalommal lemondták a foglalt időpontjaikat (status = 'cancelled\_by\_patient'). Az eredmény a páciensek egyedi TAJ-számával és nevükkel együtt jelenik meg, lehetőséget biztosítva a rendszer adminisztrátorainak a gyakori időpontlemondások elemzésére.

1. Mely Vizsgálatok hozták a legtöbb pénzt adott évben vagy hónapban?

Ez a lekérdezés az adott időszakban (évben vagy hónapban) legtöbb bevételt termelő vizsgálatokat és kezeléseket határozza meg. A SUM(t.price) segítségével az egyes kezelésekből származó teljes bevételt számolja össze, és az eredményt csökkenő sorrendben rendezi, így az első néhány sor a legnagyobb bevételt hozó kezeléseket mutatja.

1. Melyik páciens volt, aki a volt vizsgálaton, de nem értékelt?

Ez a lekérdezés azoknak a pácienseknek a listáját adja vissza, akik részt vettek egy vizsgálaton, de nem adtak értékelést (rating IS NULL). Az eredményben szerepel a páciens neve, elérhetősége, a kapott kezelés neve és az időpont dátuma, lehetőséget biztosítva arra, hogy az intézmény visszajelzést kérjen a páciensektől.

### Beépítendő megszorítások

* patients.taj\_number 9 számjegy hosszú kell, hogy legyen
* appointments.status csak előre meghatározott értékeket vehet fel
* appointments.rating csak 1 és 5 közötti szám lehet
* treatments.price > 0 (A kezelés ára nem lehet negatív vagy nulla)
* appointments.start\_time nem lehet múltbeli időpont az új foglalásoknál
* appointments.end\_time nagyobb kell, hogy legyen, mint appointments.start\_time
* users.last\_login nem lehet nagyobb, mint az aktuális dátum
* appointments.status alapértelmezett értéke 'scheduled'
* users.role\_id alapértelmezett értéke '3' (ha nincs megadva, akkor páciens lesz)
* appointments.rating alapértelmezett értéke NULL, mivel a páciens nem ad rögtön értékelést

### Scriptek

Az adatbázis scriptjei gyorshivatkozásként láthatóak az alábbiakban.

Táblák scriptje:  
[privateclinic\_table.sql](https://szamalkszalezihu-my.sharepoint.com/:u:/g/personal/bordacs_bela_diak_szamalk-szalezi_hu/EfZ95R_QS5ZHoDiyxQn9dPQB941qmTIQwcL4b1sereHvcg?e=18bh0b)  
  
Példa adatok scriptje:  
[privateclinic\_teszt\_adatok.sql](https://szamalkszalezihu-my.sharepoint.com/:u:/g/personal/bordacs_bela_diak_szamalk-szalezi_hu/Ea9uKNG08n9IhE7qPtk5ilQBSpDgydALNMlHyDn27rbA-A?e=GAlwgM)

Hasznos lekérdezések scriptje:  
[privateclinic\_lekerdezesek.sql](https://szamalkszalezihu-my.sharepoint.com/:u:/g/personal/bordacs_bela_diak_szamalk-szalezi_hu/EZ7m75_9Y2VNkEOgASHskdsBhkEAqS0dZ8NbsdFwNIObug?e=y8KYNr)

### Továbbfejlesztési ötletek

## Backend

#### Modellek és Controllerek

#### API végpontok

## Frontend

### Reszponzivitás

# Implementáció

## Program struktúra

## Adatbázis

## Backend

## Frontend

# Tesztelés

## Adatbázis

## Backend

## Frontend

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Az összes teszteset** | **Tesztelési egység** | | | |
|  | | | |
| Teszteset / Bemeneti paraméterek | Várt eredmény | Kapott eredmény | Státusz |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Fejlesztési lehetőségek

# Irodalomjegyzék