

# ZÁRÓDOLGOZAT

Bordács Béla

Karsai Bence

2025

## **Magánklinika**

# **SZÁMALK-Szalézi Technikum és Szakgimnázium**

## **Szoftverfejlesztő és tesztelő**

## **Nyilatkozat a projektfeladat eredetiségéről**

Alulírott Karsai Bence (Oktatási azonosító szám: 73090679367) büntetőjogi és fegyelmi felelősségem tudatában kijelentem és aláírásommal igazolom, hogy a projektfeladatra leadott produktum saját munkám eredménye. A felhasznált irodalmi és egyéb információs forrásokat az előírásoknak megfelelően kezeltem, a projektfeladat készítésre vonatkozó szabályokat betartottam.

Kijelentem, hogy ahol mások eredményeit, szavait vagy gondolatait idéztem, azt minden esetben, beazonosítható módon feltüntettem, a fotók és ábrák közlésével pedig mások szerzői jogait nem sértem.

Kijelentem, hogy a munkám elektronikus változata (Moodle oldalra feltöltés: <https://vizsga.szamalk-szalezi.hu/>) teljes egészében megegyezik a nyomtatott formával. Hozzájárulok ahhoz, hogy az érvényben lévő jogszabályok és a Számalk-Szalézi Technikum és Szakgimnázium belső szabályzata alapján az iskola Moodle oldalán megtekinthető (olvasható) legyen a projektfeladatra leadott munkám.

Budapest, 2025. 04 29.



.....  
Tanuló aláírása

## **Nyilatkozat a projektfeladat eredetiségéről**

Alulírott Bordács Béla (Oktatási azonosító szám: 79041066908) büntetőjogi és fegyelmi felelősségem tudatában kijelentem és aláírásommal igazolom, hogy a projektfeladatra leadott produktum saját munkám eredménye. A felhasznált irodalmi és egyéb információs forrásokat az előírásoknak megfelelően kezeltem, a projektfeladat készítésre vonatkozó szabályokat betartottam.

Kijelentem, hogy ahol mások eredményeit, szavait vagy gondolatait idéztem, azt minden esetben, beazonosítható módon feltüntettem, a fotók és ábrák közlésével pedig mások szerzői jogait nem sértem.

Kijelentem, hogy a munkám elektronikus változata (Moodle oldalra feltöltés:  
<https://vizsga.szamalk-szalezi.hu/>) teljes egészében megegyezik a nyomtatott formával.  
Hozzájárulok ahhoz, hogy az érvényben lévő jogszabályok és a Számalk-Szalézi Technikum és Szakgimnázium belső szabályzata alapján az iskola Moodle oldalán megtekinthető (olvasható) legyen a projektfeladatra leadott munkám.

Budapest, 2025. 04.29.

*Bordács Béla*

.....  
Tanuló aláírása

# 1. Tartalomjegyzék

1.	Tartalomjegyzék .....	5
2.	Ábrajegyzék.....	7
3.	GitHub Repository elérhetősége:.....	10
4.	Bevezetés .....	10
4.1.	Témaválasztás.....	10
5.	Felhasználói dokumentáció .....	11
5.1.	A program célja és lényegesebb funkciói.....	11
5.2.	Futtatási környezet .....	11
5.3.	A program fő funkcióinak leírása.....	12
5.3.1.	A program fő célja.....	12
5.3.2.	A rendszer fő funkciói a következők.....	12
5.3.3.	A program fő funkcióinak bemutatása .....	13
6.	Tervezés, drótvázak.....	27
7.	Fejlesztői dokumentáció .....	31
7.1.	Fejlesztői környezet .....	31
7.2.	Adatbázis .....	32
7.2.1.	Leírás.....	32
7.2.2.	Funkciók .....	33
7.2.3.	Adatbázis terv .....	34
7.2.4.	Technikai táblák .....	37
7.2.5.	Táblák szerkezete a tábla szintű megszorításokkal.....	39
7.2.6.	Kapcsolati ábra .....	42

7.2.7.	Hasznos lekérdezések .....	43
7.2.8.	Beépítendő megszorítások.....	45
7.2.9.	Adatbázis Script .....	46
7.3.	Backend.....	47
7.4.	Frontend .....	55
7.4.1.	Axios.....	56
7.4.2.	Contextek .....	56
7.4.3.	Routolás és navigáció .....	63
7.5.	Reszponzivitás .....	70
7.5.1.	Főoldal .....	70
7.5.2.	Doktorok .....	71
7.5.3.	Időpont foglalás .....	71
7.5.4.	Időpont foglalás – naptár .....	72
8.	Tesztelés .....	73
8.1.	Backend tesztelés .....	73
8.2.	Frontend tesztelés.....	74
9.	Fejlesztési lehetőségek.....	76
9.1.	DÁP integráció.....	76
9.2.	Fizetési lehetőségek .....	76
9.3.	Hozzászólások .....	76
9.4.	Statisztikák .....	76
9.5.	Hírfolyam .....	76
9.6.	Live chat .....	77
10.	Irodalomjegyzék.....	78
11.	Mellékletek .....	79
11.1.	Program struktúra .....	79

## 2. Ábrajegyzék

1. ábra - Főoldal .....	14
2. ábra - Rólunk .....	14
3. ábra - Doktorok.....	15
4. ábra - Doktor részletek .....	15
5. ábra - Kapcsolat.....	15
6. ábra - Bejelentkezés .....	16
7. ábra - Regisztráció doktor.....	16
8. ábra - Regisztráció páciens.....	16
9. ábra - Regisztráció visszaigazoló weboldal .....	17
10. ábra - Regisztráció visszaigazoló e-mail.....	17
11. ábra - Időpont létrehozás calendar.....	17
12. ábra - Időpont létrehozása részletek.....	18
13. ábra - Időpontok feltöltése.....	18
14. ábra - Időpont törlése .....	19
15. ábra - Foglalt időpont törlése .....	19
16. ábra - Foglalt időpont törlése visszaigazoló e-mail .....	20
17. ábra - Vizsgálat .....	20
18. ábra - Vizsgálat részletek .....	21
19. ábra - Saját páciensek .....	21
20. ábra - Összes páciens .....	21
21. ábra - Páciens részletek.....	22
22. ábra - Időpont foglalás doktorok alapján .....	22
23. ábra - Időpont foglalás visszaigazolás .....	23
24. ábra - Időpont foglalás visszaigazoló e-mail.....	23
25. ábra - Időpont foglalás doktorok alapján visszaigazolás .....	23
26. ábra - Időpont foglalás kezelések alapján .....	23
27. ábra - Időpont törlése páciens által .....	24

---

28. ábra - Időpont törlése páciens által visszaigazolás.....	24
29. ábra - Időpont törlése páciens által visszaigazoló e-mail .....	24
30. ábra - Időpont értékelése.....	24
31. ábra - Nav elemek szerkesztése .....	25
32. ábra - Felhasználó deaktiválása .....	25
33. ábra - Felhasználó deaktiválása visszaigazolás.....	25
34. ábra - Profil szerkesztése .....	26
35. ábra - Profilkép feltöltése .....	26
36. ábra - Profil módosítás mentése .....	26
37. ábra - Drótváz főoldal .....	27
38. ábra - Drótváz Bejelentkezés.....	27
39. ábra - Drótváz Regisztráció páciens.....	28
40. ábra - Drótváz Regisztráció orvos .....	28
41. ábra - Drótváz doktorok listája .....	29
42. ábra - Drótváz időpontok listája orvosok alapján.....	29
43. ábra - Drótváz időpontok listája kezelések alapján .....	30
44. ábra – Egyed kapcsolat diagram .....	36
45. ábra – Egyed kapcsolat diagram technikai táblák .....	38
46. ábra - Kapcsolati ábra .....	42
47. ábra - Modellek listája .....	48
48. ábra - Controllerek listája .....	48
49. ábra - Trigger .....	53
50. ábra - Trigger Controller .....	54
51. ábra - Bejelentkezés kód.....	57
52. ábra - useEffect.....	61
53. ábra - Context kapcsolatok .....	62
54. ábra - Context használat példa a komponensben .....	62
55. ábra - Api végpont hívás.....	62
56. ábra - komponensek és contextek felépítése .....	62
57. ábra - Routolás .....	63
58. ábra - Navigáció kód.....	64
59. ábra - Nav lista módosítás .....	64

60. ábra - Reszponzivitás főoldal iphone 12 pro .....	70
61. ábra - Reszponzivitás főoldal ipad pro .....	70
62. ábra - Reszponzivitás Doktorok ipad mini .....	71
63. ábra - Reszponzivitás Doktorok FullHD monitor .....	71
64. ábra - Reszponzivitás Időpont foglalás FullHD monitor .....	71
65. ábra - Reszponzivitás Időpont foglalás iphone 12 pro .....	71
66. ábra - Reszponzivitás Időpont foglalás naptár iphone 12 pro .....	72
67. ábra - Reszponzivitás Időpont foglalás naptár QHD monitor .....	72
68. ábra - Tesztelés Backend 2.....	73
69. ábra - Tesztelés Backend 1.....	73

### 3. GitHub Repository elérhetősége:

<https://github.com/Maganklinika/maganklinika.git>

## 4. Bevezetés

### 4.1. 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 alkalmazunk, 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.

## 5. Felhasználói dokumentáció

A különböző szerepkörökhez tartozó belépési adatok:

Felhasználó neve	email cím	jelszó	szerepkör
patienttest	patienttest@test.com	password	páciens
doctortest	doctortest@test.com	password	doktor
admintest	admintest@test.com	password	admin

### 5.1. 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 elérhetők, tehát a páciens egy szakon belül lévő vizsgálatra kiválaszt egy szabad időpontot, és látja a hozzá tartozó orvos(okat). 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.

### 5.2. Futtatási környezet

Az ötletelés fázisában gyorsan felmerült, hogy időpontfoglaló applikáció készítése illeszkedik leginkább az elképzélésekhez, amelyet webalkalmazás formájában kívánunk véghez vinni. Ennek oka, hogy kiemelten fontosnak tartottuk a felhasználóbarátságot, és a 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 - 120.0.2210.91, Google Chrome - 135.0.7049.96, Mozilla Firefox – 137.0.2, Opera - 118.0.5461.60, Vivaldi – 7.3).

## 5.3. A program fő funkcióinak leírása

### 5.3.1. 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.

### 5.3.2. 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.

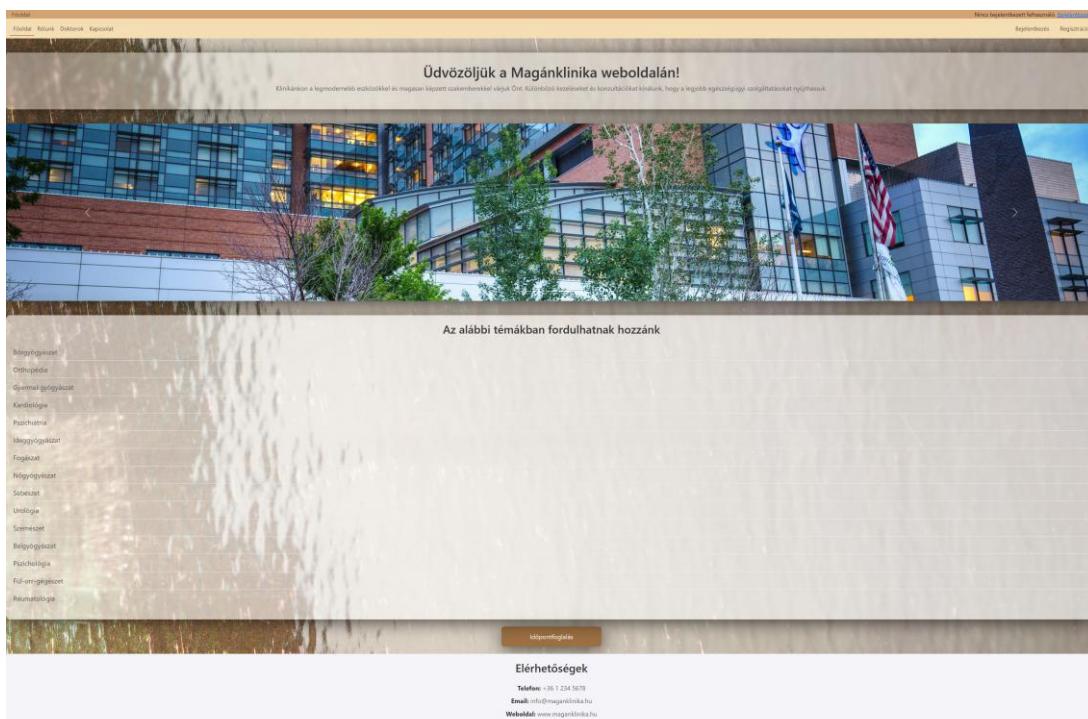
#### 6. Statisztikák és jelentések

- Az adminisztrátorok és orvosok betekintést nyerhetnek a foglalási statisztikákba, például a leggyakoribb szolgáltatásokba vagy a kihasználtsági arányokba.
- Az adatalemzés segíthet a klinika működésének optimalizálásában.

### 5.3.3.A program fő funkcióinak bemutatása

#### 5.3.3.1. Főoldal

Az oldallal elsőként a főoldalon keresztül találkozunk, amely betekintést enged a magánklinika életébe automatikusan váltakozó képekkel, valamint az elérhető kezelési témaik (specializációk) felsorolásával. Innen közvetlenül navigálhatunk az időpont foglalásra, hiszen a főoldal megtekintésének nagy részét a páciensek teszik ki, akik a kezelésekre történő időpontok miatt lépnek fel a weboldalra. Végül pedig a magánklinika elérhetőségét jelenítjük meg a honlap alján. Ez a felület mindenki számára bejelentkezés nélkül elérhető, de az oldal jobb felső sarkában van lehetőség átnavigálni a bejelentkező, valamint regisztráció képernyőre. Ez azért fontos, mert az oldal legtöbb funkciója bejelentkezással védett.



1. ábra - Főoldal

### 5.3.3.2. Rólunk

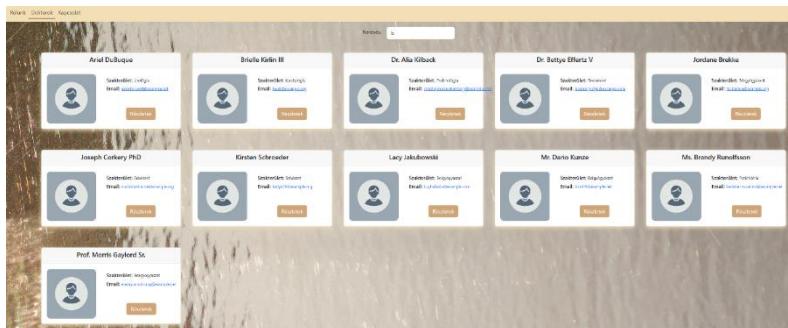
Bejelentkezés nélkül elérhető oldalakon tovább haladva a következő menüpontban egy leírás látható a klinika háttérinformációiról.

2. ábra - Rólunk

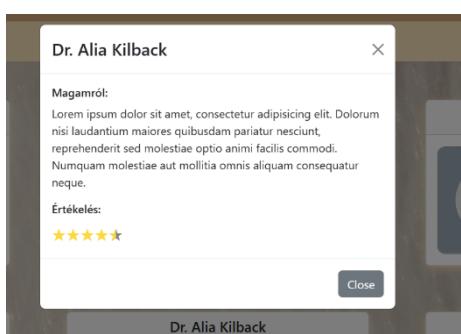
### 5.3.3.3. Doktorok

A doktorok menüpontban láthatóak a magánklinikán dolgozó (regisztrált) orvosok. Amennyiben feltöltötték profilképet, itt megjelenik. Továbbá az orvoshoz tartozó kártyán látható a szakterülete, valamint e-mail címe.

Az oldal tetején található keresés alapján, ha tudjuk az Orvos nevét, rá tudunk szűrni.



3. ábra - Doktorok

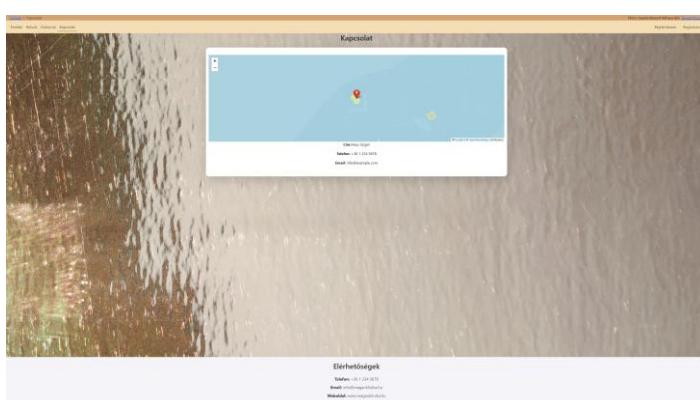


4. ábra - Doktor részletek

A részletek gombra kattintva megjelenik az orvosok által saját magukról írt bemutató, valamint az eddigi kezelések alapján összesített értékelés.

### 5.3.3.4. Kapcsolat

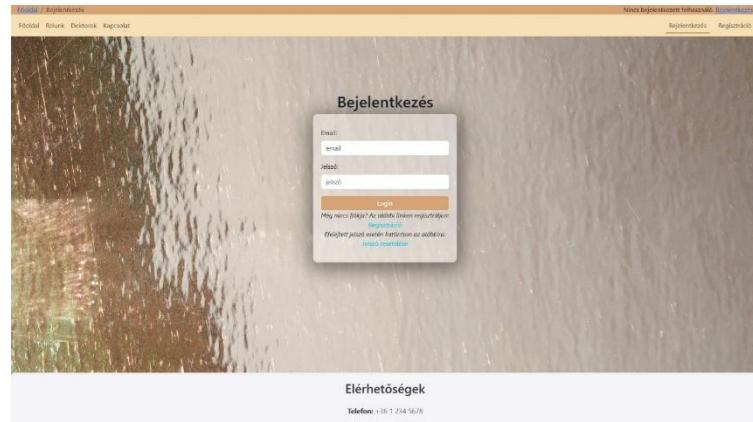
A kapcsolat menüpontban látható a magánklinika térképen elhelyezve.



5. ábra - Kapcsolat

### 5.3.3.5. Bejelentkezés

Bejelentkezés az összes megkülönböztetett felhasználó típus számára kötelező (páciens, orvos, admin). A bejelentkezés jelenleg email cím és jelszóval lehetséges. Ezen a felületen kapott helyet a jelszó visszaállítás, valamint át tudunk navigálni a



6. ábra - Bejelentkezés regisztráció oldalra. Ezen egymásba ágyazás oka, hogy nem akartuk a főoldal képét több menüponttal sűríteni, valamint a regisztráció és bejelentkezés (valamint elfelejtett jelszó visszaállítása) szorosan összefüggő menüpontok. Itt azonban megjelenítjük menüpont szinten is a fejlécnél ezen két oldalt.

### 5.3.3.6. Regisztráció

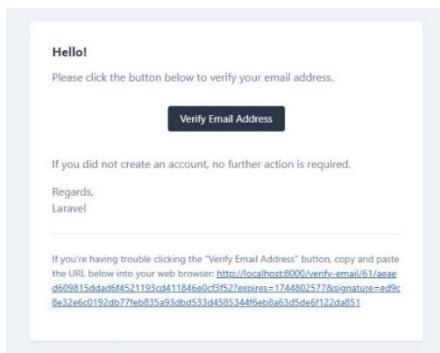
Regisztráció minden felhasználó számára szükséges, kivéve az első adminisztrátort, aki alapértelmezetten rögzítve van az adatbázis szintjén. A regisztráció során ki kell választani, hogy páciens vagy orvos szeretne regisztrálni, majd ennek megfelelően különböző adatokat szükséges megadni a folyamat során.

8. ábra - Regisztráció páciens

7. ábra - Regisztráció doktor

Doktorok számára azonban szükséges megadni az orvosi licensz számát, amellyel a valós orvosi engedély ellenőrzését szimuláljuk.

A szükséges adatok megadását követően, majd a Regisztráció gombra kattintva, a megadott e-mail címre egy megerősítő kód kerül kiküldésre, valamint erről a felület is információt ad.



10. ábra - Regisztráció visszaigazoló e-mail

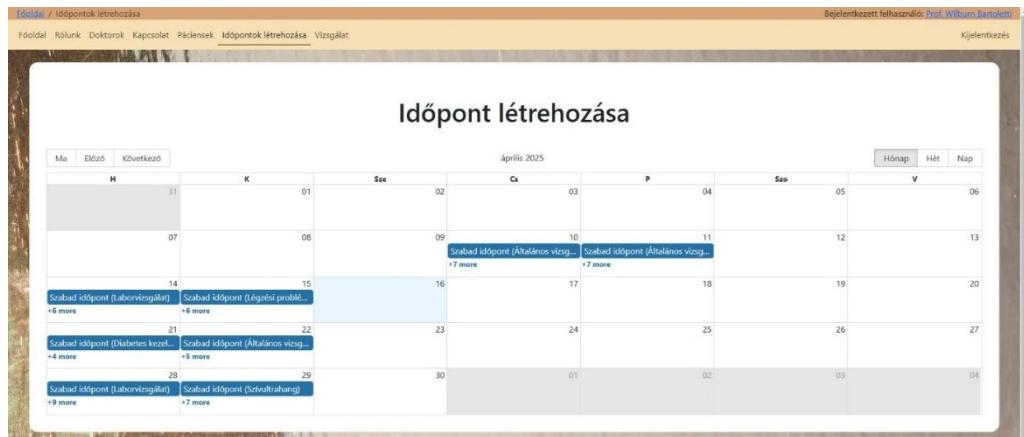


9. ábra - Regisztráció visszaigazoló weboldal

Bejelentkezést követően már a jogosultságnak megfelelő menüpontok láthatóak a navigációs listában. Ez az alapértelmezett lista módosulhat (erről bővebben az admin menüszerkesztés résznél).

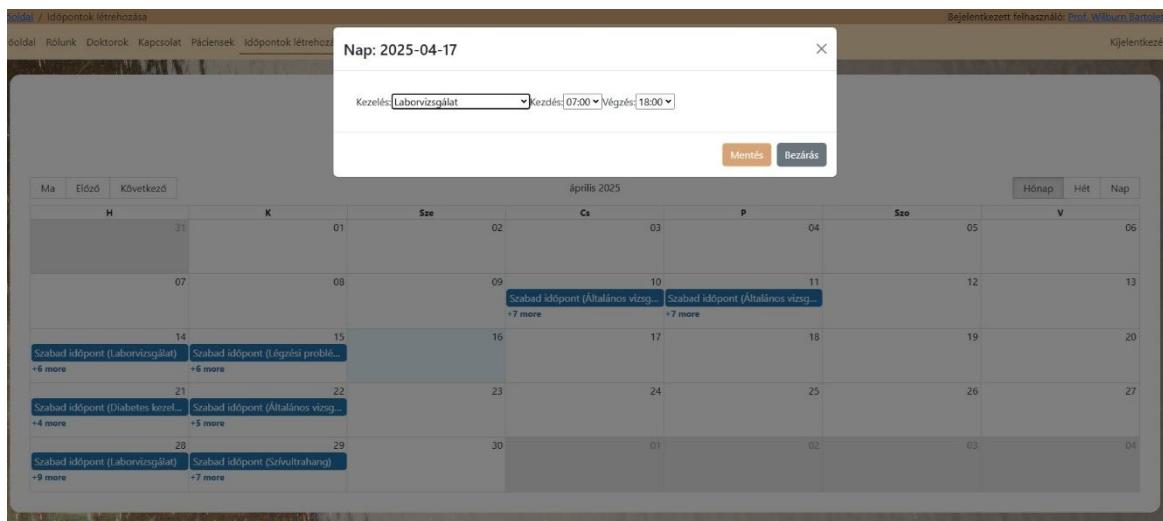
#### 5.3.3.7. Időpont létrehozás

Logikai felépítést tekintve a következő funkció az orvosok számára elérhető időpont létrehozás. Ebben a menüpontban egy calendar nézetben, az adott napra navigálva, fel tudja tölteni foglalható időpontokkal az orvos a kijelölt napot. Időpontok létrehozását visszamenőleges napokra nem enged a rendszer létrehozni. A navigáláshoz a calendar jobb oldalán található nézet váltó, valamint bal oldalon található léptetők adnak segítséget.

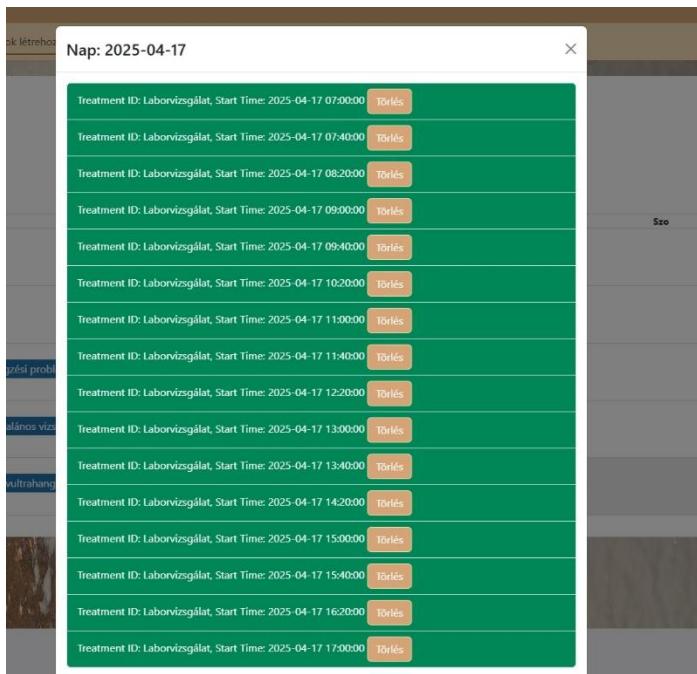


11. ábra - Időpont létrehozás calendar

A megfelelő napra kattintva felugrik egy ablak, amelyben a belépett orvos megadhatja, hogy saját specializációjához elérhető kezelések közül melyiket végezné adott napon, illetve kezdő és befejező időpontot tud megadni.



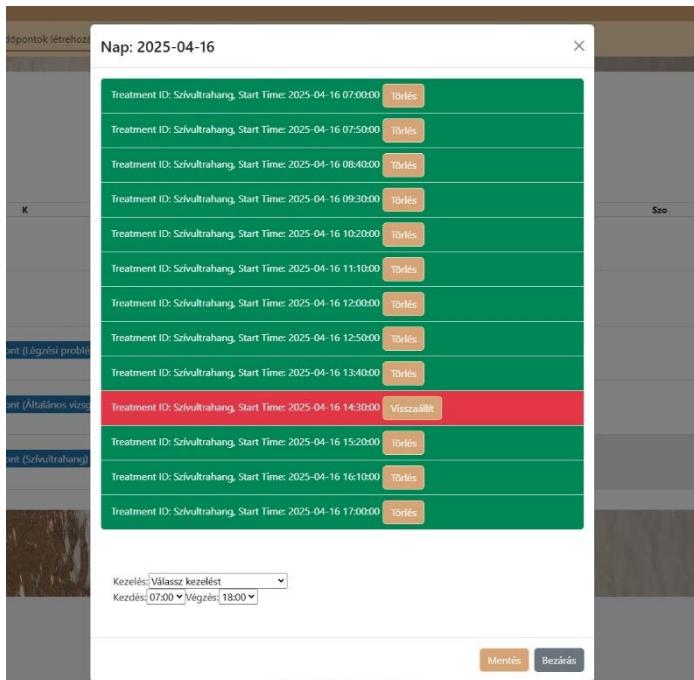
12. ábra - Időpont létrehozása részletek



13. ábra - Időpontok feltöltése

Ezután a rendszer automatikusan feltölti elérhető időpontokkal a kezelés hossza alapján, valamint 10 perces köztes szünetekkel a kezdéstől a befejezés idejéig a napot. Ha még el tudna kezdeni egy új időpontot, de a befejezés már túl nyúlni a végzés idején, akkor az az időpont már nem kerül be az időpontok listájába. Ezzel szerettük volna felhasználó barátibbá, és könnyebbé tenni az időpont létrehozást az orvosok számára.

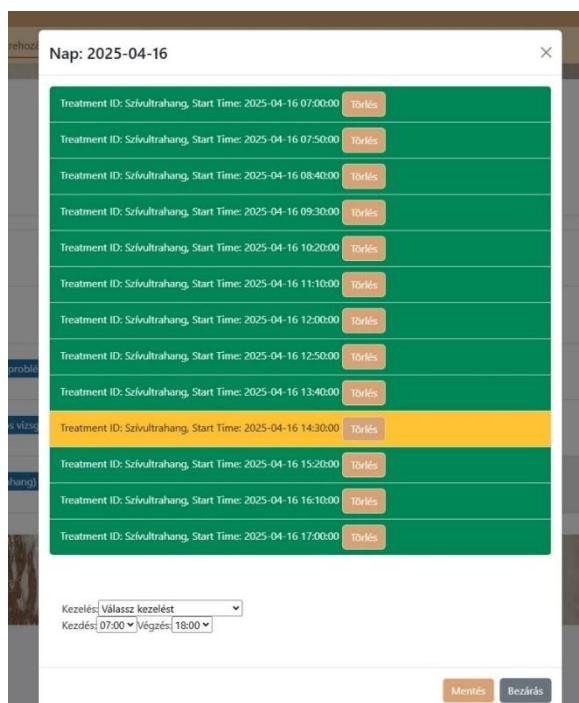
### 5.3.3.8. Időpont törlése



14. ábra - Időpont törlése

Egy már létrehozott időpontot az orvos az előzőekben látható oldalon a Törlés gombra kattintva tud törölni. Ennek sikerességét a rendszer piros háttérrel jelzi. Lehetőség van a törölt időpont visszaállítására, ekkor újra elérhetővé és foglalhatóvá válik a korábbiakban törölt kezelés.

### 5.3.3.9. Foglalt időpont törlése



15. ábra - Foglalt időpont törlése

Hasonló színezés alapú jelzéssel mutatja a rendszer, hogy ha adott napont már foglalás történt egy időpontra, jelezve ezzel, hogy a törlés olyan kezelést érint, amelyre egy felhasználó már jelentkezett, azonban ilyen esetben is lehetőség van az időpont törlésére.

### 5.3.3.10. Foglalt időpont törlése esetén e-mail értesítés

Amennyiben korábban már lefoglalt időpontra történik az orvos részéről törlés, úgy az



The screenshot shows an email from Laravel. At the top left is the Laravel logo with the text "Laravel". Below it says "Címzett: Ön". Underneath are three buttons: "Köszönöm.", "Kérlek törlje. Köszönöm.", and "Köszönöm hogy értesített.". To the right of the email body, there is a toolbar with icons for reply, forward, etc., and the date "2025. 04. 16, Sze, 15:58". The main content of the email reads:

**Kedves Karsai Bence!**

Az Ön által foglalt időpont állapotát megváltoztatta.

**Időpont:** 2025-04-16 14:30:00

**Státusz:** Törölve orvos által

Ha bármilyen kérdése van, forduljon hozzánk bizalommal.

Üdvözlettel,  
Az Ön egészségügyi csapata

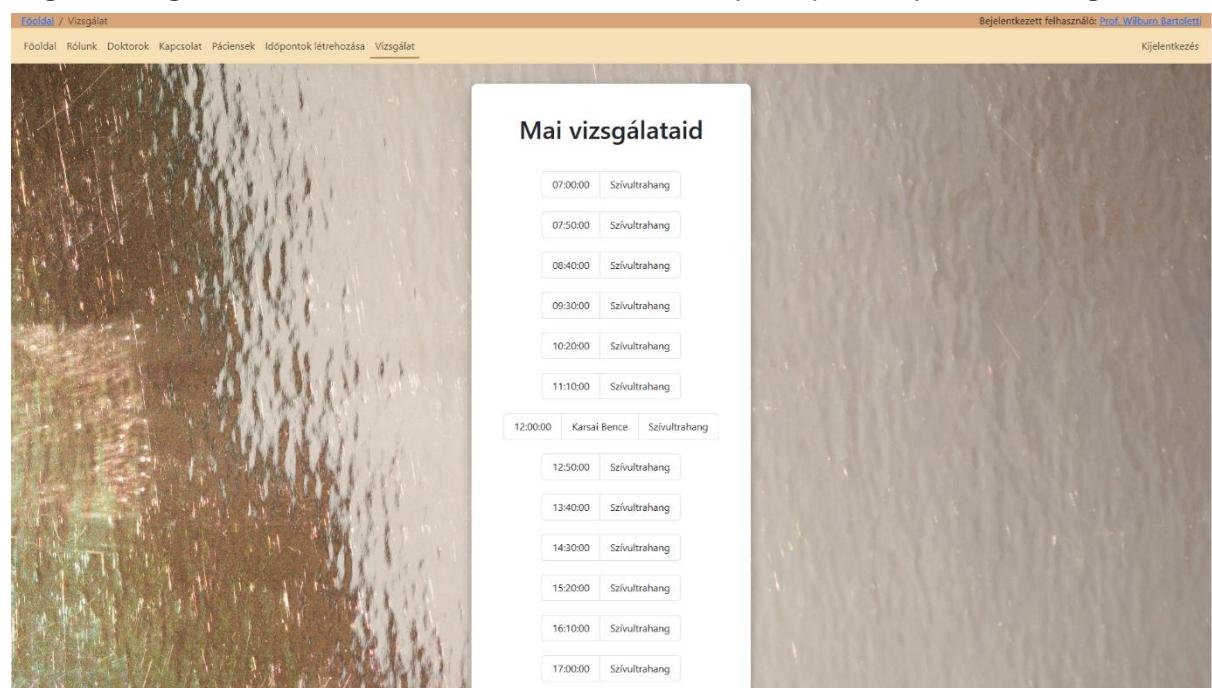
Below the email body are two buttons: "Válasz" and "Továbbítás".

**időpont törléséről  
e-mail értesítés  
került kiküldésre a  
páciens számára, a  
regisztrációjákor  
megadott e-mail  
címre.**

16. ábra - Foglalt időpont törlése visszaigazoló e-mail

### 5.3.3.11. Vizsgálat

Az orvosok a vizsgálat menüpontban listaszerűen megtekinthetik, hogy adott napon milyen kezelésre vannak időpontjaik létrehozva, valamint ezekhez milyen páciensek végeztek foglalást. minden esetben csak az adott napi időpontok jelennek meg ebben a

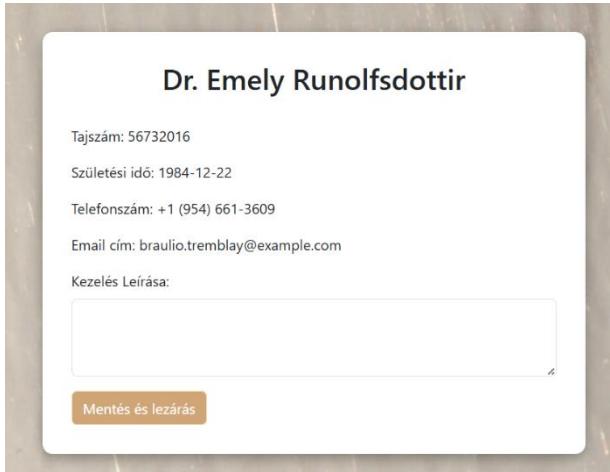


The screenshot shows a web application interface. At the top, there is a navigation bar with links: "Főoldal", "Rólunk", "Doktorok", "Kapcsolat", "Páciensek", "Időpontok létrehozása", and "Vizsgálat". On the right side of the header, it says "Bejelentkezett felhasználó: Prof. Wilburn Bartoletti" and "Kijelentkezés". The main content area has a title "Mai vizsgálataid" (Today's examinations). Below it is a table listing appointments for today:

07:00:00	Szívultrahang
07:50:00	Szívultrahang
08:40:00	Szívultrahang
09:30:00	Szívultrahang
10:20:00	Szívultrahang
11:10:00	Szívultrahang
12:00:00	Karsai Bence Szívultrahang
12:50:00	Szívultrahang
13:40:00	Szívultrahang
14:30:00	Szívultrahang
15:20:00	Szívultrahang
16:10:00	Szívultrahang
17:00:00	Szívultrahang

17. ábra - Vizsgálat

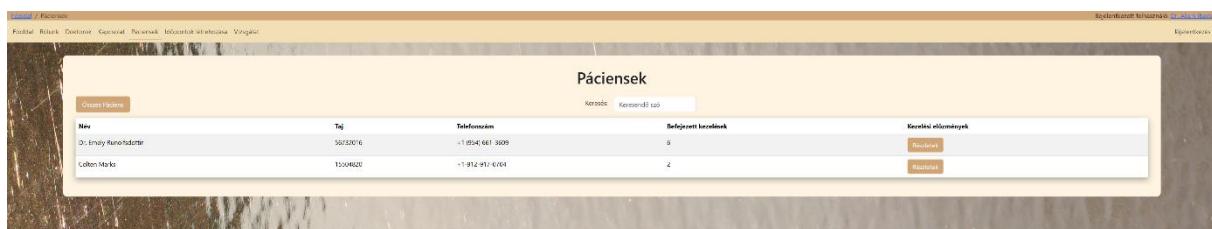
menüpontban. Ezen menüpont célja, hogy gyorsan meg tudja nézni az orvos mikor melyik (esetleg ismert) páciensével fog kezelést végezni.



18. ábra - Vizsgálat részletek

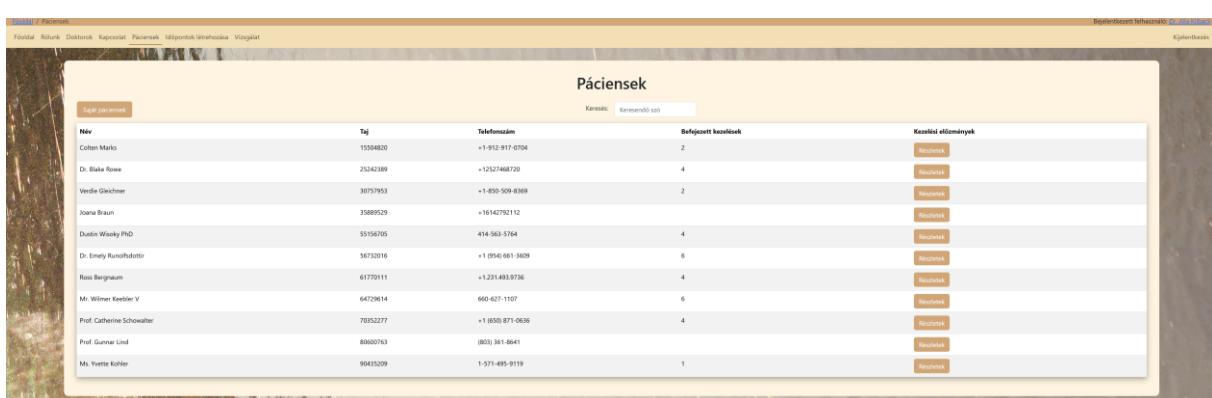
### 5.3.3.12. Páciensek

Az orvosok számára egy további menüpont érhető el, ahol alapértelmezetten saját pácienseinek listáját látják, ugyanis legtöbbször az orvosok ezen személyekre fognak keresni, információkat vizsgálni. A listában láthatóak a páciens személyes adatai, valamint befejezettségeinek listája.



19. ábra - Saját páciensek

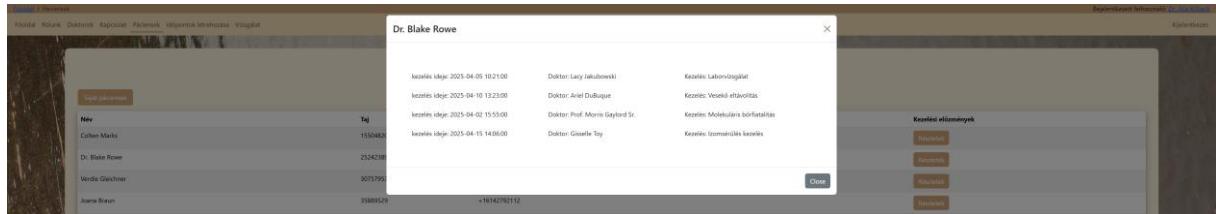
Az összes páciens gombra kattintva az összes regisztrált páciens megtekinthető.



20. ábra - Összes páciens

Sok páciens esetén javasolt a kereső mező használata, amely a páciensek neve alapán szűri a találatokat.

A részletek gombra kattintva megtekinthetőek az adott páciens korábbi kezelései listaszerűen.

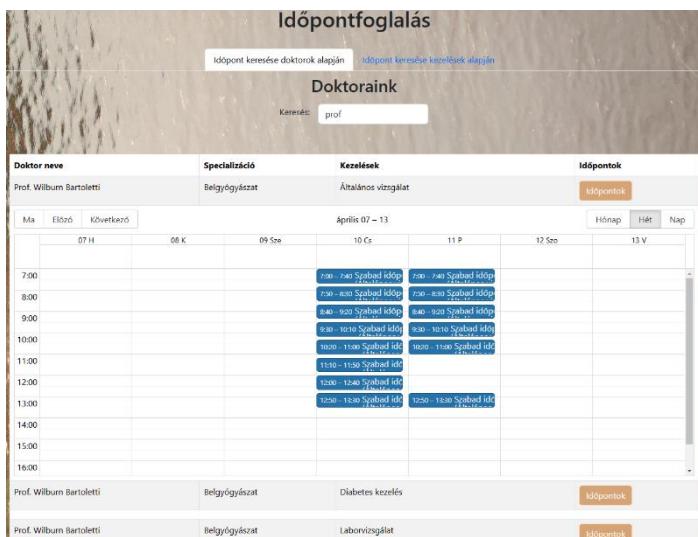


21. ábra - Páciens részletek

#### 5.3.3.13. Időpont foglalása

Időpontokra keresését, majd adott időpontra foglalást a páciensek két módon tudják megtenni:

Doktorok illetve kezelések alapján.



22. ábra - Időpont foglalás doktorok alapján

Doktorok alapján történő keresésnél amennyiben tudjuk az orvos nevét, célszerű a szűrőt használni a találatok leszűkítése érdekében.

A preferált orvos mellett található Időpontok gombra kattintva lenyílik a naptár, amelyben havi / heti / napi lebontásban látjuk a foglalható időpontjait. A bal oldalon látható navigációs gombokkal tudunk léptetni, vagy a mai dátumra visszaugrani.

Az aktuális időpontra kattintva egy felugró ablakban láthatjuk az aktuális vizsgálat részleteit, majd az Igen gombra kattintva a vizsgálat lefoglalásra kerül. Ezzel egyidejűleg automatikusan átnavigál a felület a Profil oldalra, valamint visszaigazoló e-mailt küld a sikeres foglalásról.

Laravel  
Címzett: Ön

Válasz kezdése a következővel:

2025. 04. 16, Sze, 14:22

**Kedves Karsai Bence!**

Az Ön által foglalt időpont státusza megváltozott.

**Időpont:** 2025-04-10 08:40:00

**Státusz:** Foglalva

Ha bármilyen kérdése van, forduljon hozzánk bizalommal.

Üdvözlettel,  
Az Ön egészségügyi csapata

24. ábra - Időpont foglalás visszaigazoló e-mail



23. ábra - Időpont foglalás visszaigazolás

Kezelések alapján történő keresésnél legördülő listából tudjuk a kívánt kezelést kiválasztani. Ezt követően a preferált időpont mellett a foglalok gombra kattintás után felugró ablakban tudjuk visszaigazolni foglalási szándékunkat.

**Foglalási Időpontok**

Válasszon kezelést

Szorongás kezelés

Időpont:	2025-04-21 10:30:00	<input type="button" value="Foglalok"/>
Időpont:	2025-04-21 11:40:00	<input type="button" value="Foglalok"/>
Időpont:	2025-04-21 12:50:00	<input type="button" value="Foglalok"/>
Időpont:	2025-04-22 13:00:00	<input type="button" value="Foglalok"/>
Időpont:	2025-04-22 14:10:00	<input type="button" value="Foglalok"/>
Időpont:	2025-04-22 15:20:00	<input type="button" value="Foglalok"/>
Időpont:	2025-04-22 16:30:00	<input type="button" value="Foglalok"/>
Időpont:	2025-05-19 07:00:00	<input type="button" value="Foglalok"/>

26. ábra - Időpont foglalás kezelések alapján



25. ábra - Időpont foglalás doktorok alapján visszaigazolás

A két nézet után a folyamat további része megegyezik.

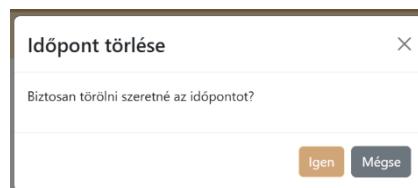
### 5.3.3.14. Időpont törlése

A profilkok oldalon a bejelentkezett felhasználó a foglalt időpontjait tudja megtekinteni.

Időpont	Orvos	Kezelés	Törles	Értékelés
2025-04-19 07:00:00	Dr. Alia Kilback	Szorongás kezelés	Már nem törölhető	★★★★★
2025-04-20 08:15:00	Dr. Alia Kilback	Szorongás kezelés	Már nem törölhető	★★★★★
2025-04-21 09:20:00	Dr. Alia Kilback	Szorongás kezelés	<b>Töröl</b>	A kezelés folyamatban
2025-03-27 05:00:00	Károly Wolfi	Vérnyomás kezelés	Már nem törölhető	★★★★★
2025-04-02 12:50:00	Veda Schuler	Alvászavar kezelés	Már nem törölhető	★★★★★
2025-03-30 12:27:00	Deejan Soliman	Pánikrövör kezelés	Már nem törölhető	★★★★★
2025-03-31 12:00:00	Johanna Juhász	Laparoskopikus műtét	Már nem törölhető	★★★★★

27. ábra - Időpont törlése páciens által

A kezelések listájában a lefoglalt, de még orvos által nem lezárt időpont törlésére van lehetőség a Törlés gombbal, amelyre kattintva, felugró ablakban tudjuk visszaigazolni a műveletet.



28. ábra - Időpont törlése páciens által visszaigazolás

Sikerességtől függően visszaigazoló e-mail érkezik a műveletről, benne a vizsgálat időpontjával.

Válasz kezdése a következővel:

**Kedves Karsai Bence!**

Az Ön által foglalt időpont státusa megváltozott.

**Időpont:** 2025-04-16 14:30:00

**Státusz:** Visszamondva

Ha bármilyen kérdése van, forduljon hozzánk bizalommal.

Üdvözlettel,  
Az Ön egészségügyi csapata

29. ábra - Időpont törlése páciens által visszaigazoló e-mail

### 5.3.3.15. Időpont értékelése

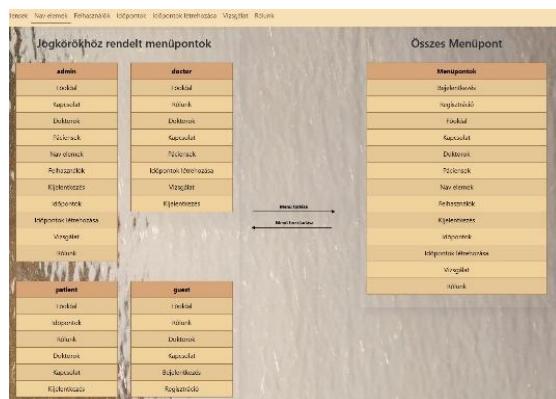
Időpont	Orvos	Kezelés	Törles	Értékelés
2025-04-19 07:00:00	Dr. Alia Kilback	Szorongás kezelés	Már nem törölhető	Az értékelést: ★★★★★
2025-04-20 08:10:00	Dr. Alia Kilback	Szorongás kezelés	Már nem törölhető	Az értékelést: ★★★★★
2025-04-21 09:20:00	Dr. Alia Kilback	Szorongás kezelés	<b>Töröl</b>	A kezelés folyamatban

30. ábra - Időpont értékelése

Ugyanezen résznél lehetséges az orvos által már lezárt (tehát lezajlott) időpontok esetén értékelést adni az orvosról, az adott időpont esetén. Ezen értékelések átlaga szerepel az adott Orvos részleteinél.

### 5.3.3.16. Navigációs elemek szerkesztése

Adminok részére lehetőség van a különböző jogosultsági szintekhez egyedi menülistát beállítani, amelyet a Navigációs elemek szerkesztése menüpontból tud az admin elérni.



31. ábra - Nav elemek szerkesztése

Ezen menüpontban jobb oldalt láthatjuk az összes elérhető menüt, amit drag and drop segítségével hozzá tudunk adni a bal oldalon található jogosultsági szintek valamelyikéhez. Ugyanígy el is távolíthatunk menüpontot adott role esetén, egyszerűen megfogva és jobb oldalra áthúzva a menüpontot.

### 5.3.3.17. Felhasználó deaktiválása



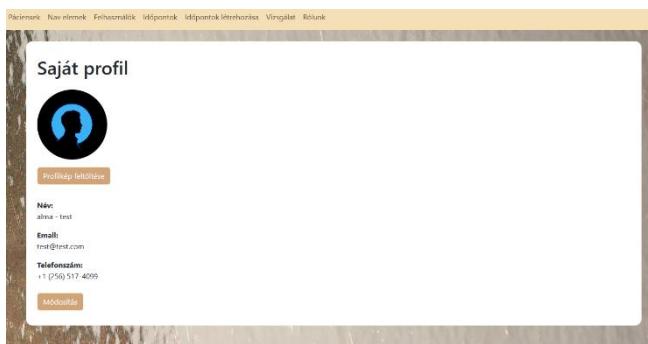
32. ábra - Felhasználó deaktiválása

A felhasználók menüpont alatt találjuk az összes regisztrált felhasználót, és a hozzá tartozó jelenlegi jogosultsági szintet. A jogosultság oszlopban tudjuk deaktiválni a felhasználót, amivel „guest” jogosultsága lesz, valamint erről a felület tájékoztatja az érintett felhasználót a következő belépési kísérlet során.



33. ábra - Felhasználó deaktiválása visszaigazolás

### 5.3.3.18. Profil szerkesztése



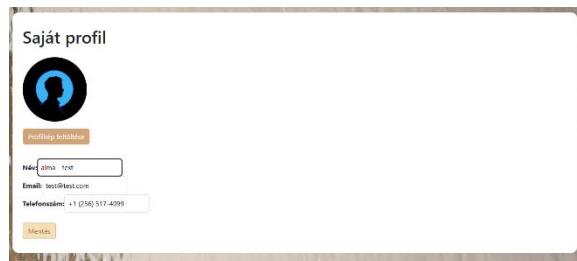
34. ábra - Profil szerkesztése



35. ábra - Profilkép feltöltése

A profil menüpontban a bejelentkezett felhasználó profilképet tud feltölteni, valamint bizonyos adatait tudja szerkeszteni a Módosítás gombra kattintva.

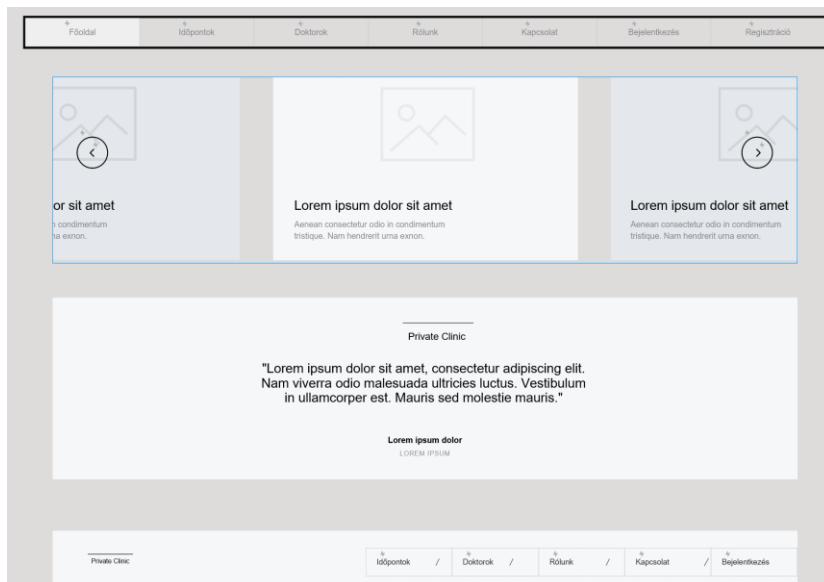
A módosítást követően a Mentés gombra kattintva frissülnek az adatok.



36. ábra - Profil módosítás mentése

## 6. Tervezés, drótvázak

Főoldal:



37. ábra - Drótváz főoldal

Bejelentkezés:

38. ábra - Drótváz Bejelentkezés

## Regisztráció páciensek esetén:

The screenshot shows the registration form for patients. At the top, there is a navigation bar with links: Főoldal, Időpontok, Doktorok, Rólunk, Kapcsolat, Bejelentkezés, and Regisztráció. Below the navigation bar, the main title is "Regisztráció". There are two radio button options: "Páciens" (selected) and "Orvos". The form fields include:

- Email:** Input field with error message "The input field must contain @".
- Jelszó:** Input field with error message "The text field cannot be empty".
- Jelszó megerősítés:** Input field with error message "The text field cannot be empty".
- Név:** Input field with error message "Input".
- Telefonszám:** Input field with error message "Input".
- Születési idő:** Input field with error message "ÉÉÉÉ.hh/nn".
- Lakcím:** Input field with error message "Input".
- TAJ szám:** Input field with error message "Input".

At the bottom of the form are two buttons: "Regisztrálok" (Registration) and "Bejelentkezés" (Login). The footer contains the text "Private Clinic" and a navigation bar with links: Időpontok, Doktorok, Rólunk, Kapcsolat, and Bejelentkezés.

39. ábra - Drótváz Regisztráció páciens

## Regisztráció orvosok esetén:

The screenshot shows the registration form for doctors. At the top, there is a navigation bar with links: Főoldal, Időpontok, Doktorok, Rólunk, Kapcsolat, Bejelentkezés, and Regisztráció. Below the navigation bar, the main title is "Regisztráció". There are two radio button options: "Páciens" and "Orvos" (selected). The form fields include:

- Email:** Input field with error message "The input field must contain @".
- Jelszó:** Input field with error message "The text field cannot be empty".
- Jelszó megerősítés:** Input field with error message "The text field cannot be empty".
- Név:** Input field with error message "Input".
- Szak kiválasztása:** A dropdown menu showing "Kardiológia".
- Orvos Azonosító:** Input field with error message "Input".
- Telefonszám:** Input field with error message "Input".

At the bottom of the form are two buttons: "Regisztrálok" (Registration) and "Bejelentkezés" (Login). The footer contains the text "Private Clinic" and a navigation bar with links: Időpontok, Doktorok, Rólunk, Kapcsolat, and Bejelentkezés.

40. ábra - Drótváz Regisztráció orvos

## Doktorok listája:

The wireframe shows a header with navigation links: Főoldal, Időpontok, Doktorok, Rólunk, Kapcsolat, Bejelentkezés, and Regisztráció. Below the header is a grid of six doctor profiles, each consisting of a placeholder image, the doctor's name (Szaktelektér Doktor neve), a short description (Aenean consectetur odio in condimentum tristique. Nam hendrerit urna ex, non pretium erat pellentesque eget. Sed ut risus nec augue sagittis convallis.), and a review section with 'Részletek' and 'Értékelés: ★★★★☆'. At the bottom of the grid is a footer with links: Időpontok / Doktorok / Rólunk / Kapcsolat / Bejelentkezés.

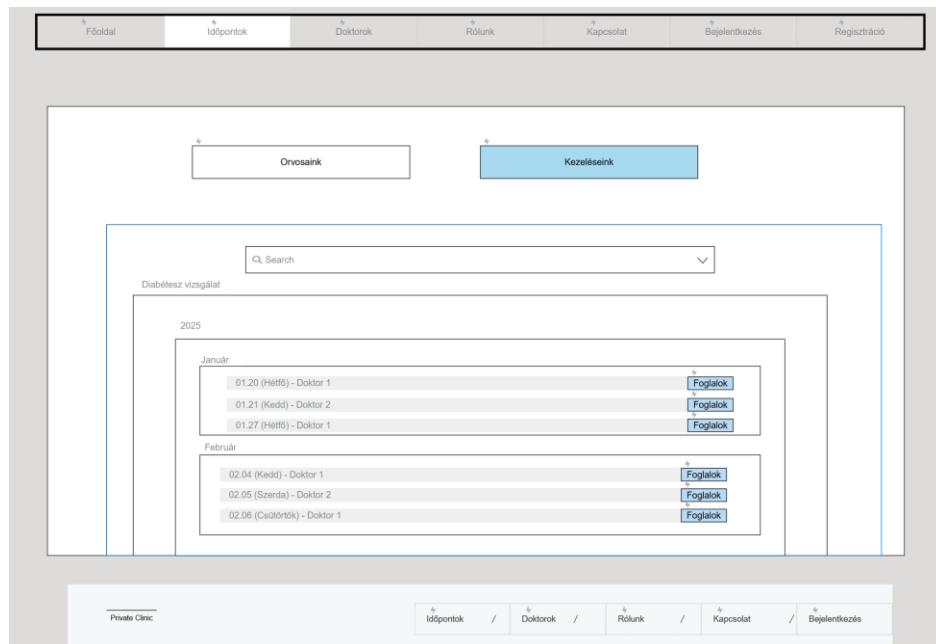
41. ábra - Drótváz doktorok listája

## Időpontok orvosok szerint:

The wireframe shows a header with navigation links: Főoldal, Időpontok, Doktorok, Rólunk, Kapcsolat, Bejelentkezés, and Regisztráció. Below the header is a grid of appointment slots. Each slot is divided into two sections: 'Orvosaink' (with placeholder names like 'Doktor 1 Szak') and 'Kezeléseink' (with a placeholder 'Kezelés'). Each slot also includes a 'Szak' dropdown and a 'Időpontok' button. At the bottom of the grid is a footer with links: Időpontok / Doktorok / Rólunk / Kapcsolat / Bejelentkezés.

42. ábra - Drótváz időpontok listája orvosok alapján

## Időpontok kezelések szerint:



43. ábra - Drótváz időpontok listája kezelések alapján

## 7. Fejlesztői dokumentáció

### 7.1. 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 MariaDB segítségével történt. Szerver oldalon PHP programozási nyelvet használtunk, illetve RESTful mentén Laravel keretrendszer használtunk. Kliens oldali megvalósítás Javascript segítségével történt React használatával. A fejlesztést Visual Studio Code-ban végeztünk, verziókövetésre GitHub-ot használtunk.

Használt rendszerek és verzióik:

- MariaDB 10.4.32
- PHP 8.2.12
- Laravel 11.41.3
- React 19.0.0
  
- VSCode (fejlesztéshez) 1.99.3
- Xampp (virtualizációs környezet) 3.3.0
- Justinmind (drótváz készítéshez) 10.7.3

## 7.2. Adatbázis

### 7.2.1. Leírás

A Magánklinika adatbázis a klinika betegfoglalásait, orvosi szakrendeléseit és felhasználót 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 kezeljük. A nem közös adatokat pedig külön az orvosoknak és pácienseknek létrehozandó táblában tároljuk. 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. Továbbá egy mezőt, ami tartalmazza a doktor által megírt kezelés zárójelentését, Emellett egy értékelés mező is bekerül, amely segíti a pácienseket a megfelelő szakember kiválasztásában. Az időpontokat az orvosok szabad kapacitásai alapján lehet lefoglalni.

### 7.2.2. 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 és menülista menedzselése
- Orvosi értékelések kezelése és megjelenítése

### 7.2.3. Adatbázis terv

Jelölés:

Tábla magyar neve (TÉNYLEGES TÁBLA NEVE)

(kulcs, leíró, *kapcsolathordozó*)

Rövid leírás

#### 7.2.3.1. Adattáblák leírása

Felhasználói szerepkör

{jogosultság azonosító, név}

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

{azonosító, név, telefonszám, email, jelszó, jogosultság azonosító, profilkép}

Ez a tábla tárolja a rendszer összes regisztrált felhasználóját, függetlenül attól, hogy pácienskről, 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 minden legyen érvényes szerepköre. Tovább a felhasználóhoz opcionálisan feltölthető a default profilképtől eltérő kép.

Páciens

{felhasználó azonosító, TAJ szám, születési idő, cím}

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

{specializáció azonosító, specializáció név}

A klinikán dolgozó orvosok különböző szakterületekre specializálódnak, például belgyógyászat, sebészet vagy ortopédia. Ebben a táblában vannak felsorolva ezek a szakterületek.

## Orvos

{felhasználó azonosító, specializáció azonosító}

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 szakterületükhez tartoznak.

## Kezelés

{kezelés azonosító, specializáció azonosító, kezelés neve, kezelés hossza, ár}

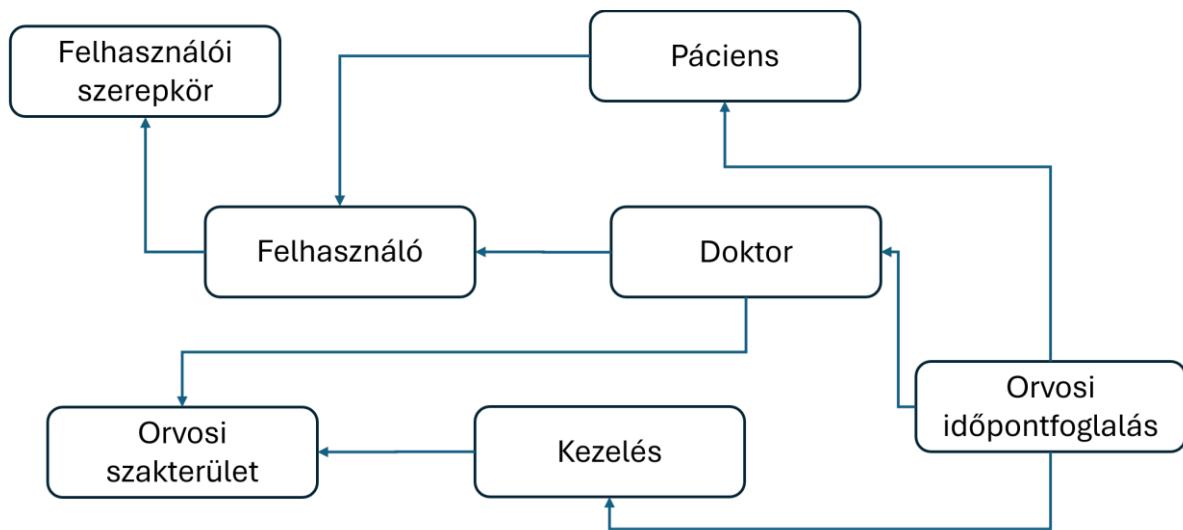
A klinika által nyújtott különböző kezelések listája ebben a táblában található. minden kezeléshez meghatározott időtartam és ár tartozik, valamint egy szakirány, amely biztosítja, hogy csak a megfelelő szaktudással rendelkező orvosok végezhessék el az adott beavatkozást.

## Orvosi időpontfoglalás

{időpont azonosító, orvos azonosító, kezdés ideje, páciens azonosító, kezelés azonosító, leírás, státusz, értékelés}

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. Emellett az érintett kezelés aktuális időpontjára az orvos leírást tud adni.

### 7.2.3.2. Egyed kapcsolat diagram



44. ábra – Egyed kapcsolat diagram

## 7.2.4. Technikai táblák

A Navigációs menüpont és Menüpont-hozzáférési jogosultság 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. A Doktor licensz tábla tartalmazza a licensz számát, illetve, hogy már aktiválásra került-e.

### 7.2.4.1. *Technikai táblák leírása*

Navigációs menüpont

{navigáció azonosító, név, URL, komponens neve, szülő}

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

{navigációs szerepkör azonosító, sorszám, *jogosultság azonosító*, *navigáció azonosító*}

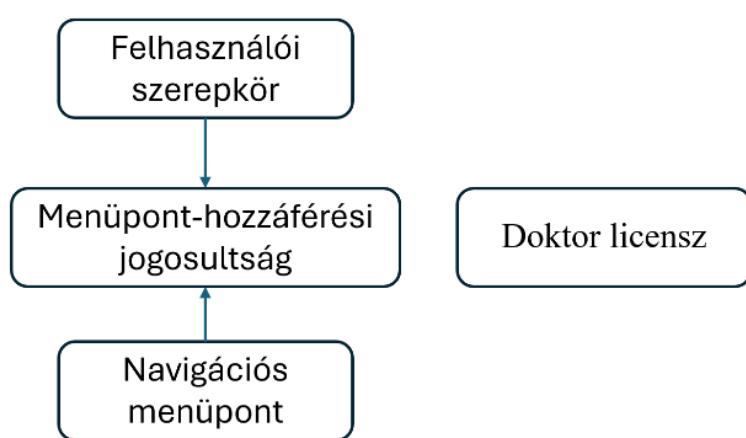
Nem minden menüpont érhető el minden felhasználói szerepkör számára. Ebben a táblában van meghatározva, hogy mely szerepkörök milyen menüpontokat láthatnak és használhatnak. Így például a páciensek nem férhetnek hozzá az adminisztrációs funkciókhoz, míg az adminisztrátorok teljes hozzáféréssel rendelkeznek.

Doktor licensz

{licenzsz azonosító, licensz sorszám, aktivált}

Elképzelésünk egy izolált tábla, amely az adatbázis szerkezetén kívül helyezkedik el. Ez a tábla az orvos regisztrációjához szükséges.

#### 7.2.4.2. Technikai táblák egyed kapcsolat diagram



A felhasználói szerepkör tábla ugyan nem technikai, de közvetlen kapcsolatban van a menüpont-hozzáférési jogosultság táblával. Az áttekinthetőség érdekében a Felhasználói szerepkör táblát is szerepeltetjük ezen ábrán.

45. ábra – Egyed kapcsolat diagram technikai táblák

## 7.2.5. Táblák szerkezete a tábla szintű megszorításokkal

### 7.2.5.1. Adattáblák

<b>SZEREP (ROLE)</b>				
Mező magyarul	Mező valós	Adattípus	Szerep	Korlátozás
jogosultság azonosító	role_id	bigint(20)	K	
név	name	varchar(255)	kötelező, egyedi	

<b>FELHASZNÁLÓ (USER)</b>				
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 tábla méret
profil kép	profile_picture	varchar(255)		

<b>SPECIALIZÁCIÓ (SPECIALIZATION)</b>				
Mező magyarul	Mező valós	Adattípus	Szerep	Korlátozás
specializáció azonosító	specialization_id	bigint(20)	K	
specializáció név	specializaton_name	varchar(255)	kötelező, egyedi	

<b>ORVOS (DOCTOR)</b>				
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 rendelkezhet

<b>PÁCIENS (PATIENT)</b>				
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	

<b>KEZELÉS (TREATMENT)</b>				
Mező magyarul	Mező valós	Adattípus	Szerep	Korlátozás
kezelés azonosító	treatment_id	bigint(20)	K	
specializáció azonosító	specialization_id	bigint(20)	KK (SPECIALIZATION)	
kezelés neve	treatment_name	varchar(255)	kötelező )	
kezelés hossza	treatment_length	time	kötelező	kezelés hossza > 0
ár	price	int(10)	kötelező	ár >=0

<b>ORVOSI IDŐPONTFOGLALÁSOK (DOCTOR_APPOINTMENT)</b>				
Mező magyarul	Mező valós	Adattípus	Szerep	Korlátozás
időpont azonosító	id	bigint(20)	K	
orvos azonosító	doctor_id	bigint(20)	KK része (DOCTOR)	
kezdés ideje	start_time	datetime	kötelező	kezdés idő > mai nap
páciens azonosító	patient_id	bigint(20)	KK (PATIENT)	
kezelés azonosító	treatment_id	bigint(20)	KK (TREATMENT)	
leírás	description	varchar(255)		
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
- adott orvoshoz, csupán a hozzá tartozó kezelések közül tudjon a páciens foglalni időpontot

### 7.2.5.2. 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)

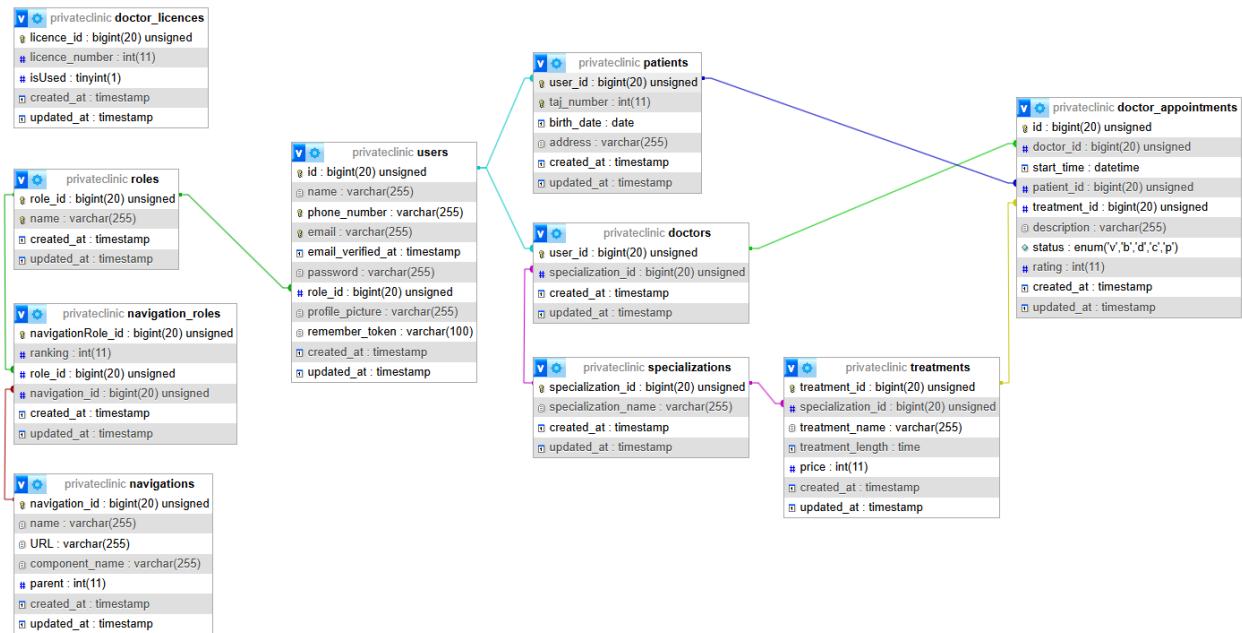
NAVIGÁCIÓS SZEREPKÖRÖK (NAVIGATION_ROLE)				
Mező magyarul	Mező valós	Adattípus	Szerep	Korlátozás
navigációs szerepkör azonosító	navigationRole_id	bigint(20)	K	
sorszám	ranking	int(11)	kötelező	ranking > 0
jogosultság azonosító	role_id	bigint(20)	KK (ROLE)	
navigáció azonosító	navigation_id	bigint(20)	KK (NAVIGATION)	

További megszorítások:

- szerepkörönként nem ismétlődhet a sorszám

DOKTOR LICENSZ (DOCTOR_LICENCE)				
Mező magyarul	Mező valós	Adattípus	Szerep	Korlátozás
licenzs azonosító	licence_id	bigint(20)	K	
licenzs sorszám	licence_number	int(11)	kötelező	
használatban	isUsed	boolean		

## 7.2.6. Kapcsolati ábra



46. ábra - Kapcsolati ábra

## 7.2.7. 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 mindenkor az aktuális hónaphoz képest előző hónapot vizsgálja.

### 2. 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.

### 3. Hány Páciensnek volt elvégzett kezelése?

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

### 4. 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 hónapról vagy évről, így az orvos munkájának elemzésében segíthet.

## 5. 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.

## 6. 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 legyengébben értékelt orvosokat mutatja.

## 7. 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 bejelentkezási dátum (last\_login) alapján ellenőri, hogy a dátum több mint három ével ezelőtti-e. Ez segíthet az inaktív felhasználók kiszűrésében vagy figyelmeztetés küldésében.

## 8. 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.

## 9. Mely páciensek mondta le az időpontjaikat legalább 10 alkalommal tavaly?

Ez a lekérdezés azoknak a pácienseknek a listáját adja vissza, akik az előző évben legalább tíz alkalommal lemondta 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.

#### 10. 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 rendezzi, így az első néhány sor a legnagyobb bevételt hozó kezeléseket mutatja.

#### 11. 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.

### 7.2.8. 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

- Egy adott orvos (doctor\_id) egy adott időpontra (start\_time) csak olyan kezelést (treatment\_id) adhat ki, amely az Ő szakterületéhez (specialization\_id) tartozik.
- Az orvos csak akkor tudjon regisztrálni, ha a doctor\_licence táblában található isUsed állapot false.

### 7.2.9. Adatbázis Script

Az adatbázis scriptje az alábbi linken található, amely a GitHub repositoryban foglal helyet:

[maganklinika\\_adatbazis.sql](#)

### 7.3. Backend

Az adatbázis megtervezése után a következő lépés a backend, tehát a háttérrendszer kialakítása. Ez felel az üzleti logika megvalósításáért és az adatok kezeléséért. A mai modern webalkalmazások nagyrésze ehhez API-kat használ, amelyek lehetővé teszik a frontend számára, hogy strukturált módon, jellemzően JSON formátumban kommunikáljon a backenddel.

A szakdolgozatunk készítése során ilyen API végpontokat hoztunk létre, amelyhez Laravel keretrendszert használtunk. Ez a módszer az MVC (Model – View – Controller) tervezési mintát követi.

A Model kezeli az egyes erőforrásokkal kapcsolatos adatokat.

A View réteg a megjelenésért és a felület kinézetéért felelős. Emiatt erre bővebben a Frontend esetén fogunk kitérni.

A Controller réteg köti össze a Modelt és Viewt. Így folyamatos kommunikációt végez a Model és View rétegekkel, azokat vezérelve.

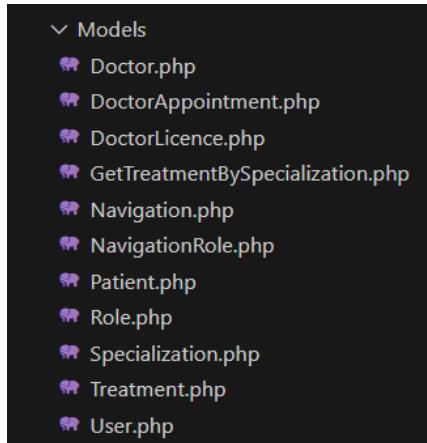
Backend esetén a route-ok segítségével határoztuk meg, hogy az éppen aktuális kérés feldolgozásáért melyik Controller, és azon belül melyik metódus felelős.

Laravel esetén telepített függőségek:

- Laravel/Sanctum 4.0.8
- Laravel/Breeze 2.3.3

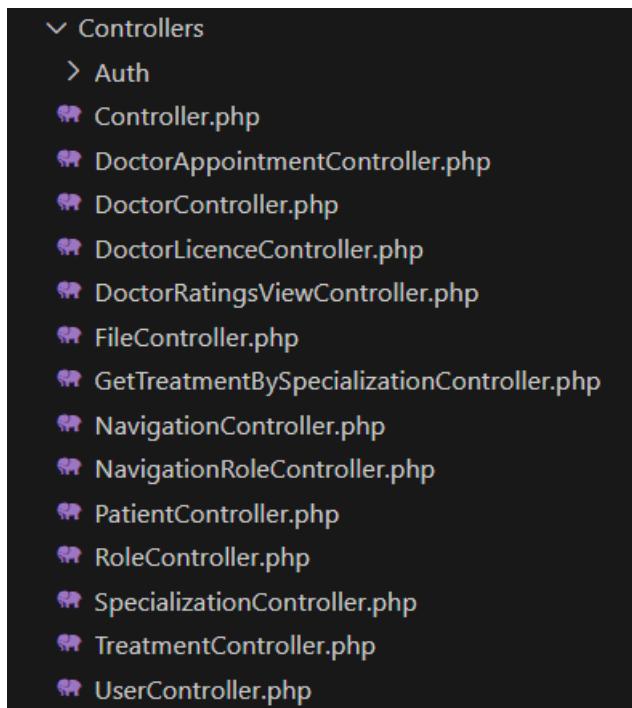
### 7.3.1.1. Modellek és Controllerek

Modellek az alábbiakban láthatóak VSCode alapján:



47. ábra - Modellek listája

Controllerek megjelenése VSCode-ban:



48. ábra - Controllerek listája

### 7.3.1.2. API végpontok

Az API végpontok listáját a következőkben szétbontjuk a különböző jogosultsági körökre (middleware):

Jogosultság nélkül elérhető végpontok:

Kérés Típusa	URI	LEÍRÁS
GET	/nav-items	Lekéri a navigációs elemeket
GET	/get-avg-ratings-by-doctors	Lekéri a doktorok átlag értékelését
GET	/treatments	Kilistázza az összes kezelést
GET	/specializations	Visszaadja az összes specializációt
GET	/checkLicenceByld/{id}	Ellenőrzi, hogy az orvosi licensz létezik-e, vagy már használatban van, vagy szabad
GET	/get-available-appointments	Kilistázza a szabad időpontok listáját

Páciens jogosultsággal rendelkező végpontok:

Kérés típusa	URI	LEÍRÁS
GET	/get-appointments-by-doctor	Lekéri az adott orvoshoz tartozó időpontokat
GET	/get-treatments-by-specialization	Lekéri a specializációhoz tartozó kezeléseket
GET	/test-get-tbs	Az összes specializációhoz tartozó kezeléseket adja vissza
GET	/appointments	Kilistázza az összes időpontot
GET	/doctors-with-spec	Visszaadja a doktorokat és hozzájuk tartozó specializációkat
GET	/get-treatment/{id}	Lekéri az adott azonosítójú kezelést
GET	/user-data	A bejelentkezett user tábla összekapcsolásáért felel a páciens vagy orvos táblával, a jogosultság alapján
GET	/get-appointments-by-patients/{id}	Lekéri a megadott páciens időpontjait
PUT	/profile/rate/{id}	Értékelést ad az adott időponthoz
PUT	/booking-appointment/{id}	Lefoglal egy létező és foglalható időpontot
GET	/get-all-appointment-by-doctor	Lekéri az adott orvos összes időpontját
PUT	/cancel-appointment-by-patient/{id}	Az adott beteg törli az adott időpontfoglalását
PUT	/change-user-info/{id}	Módosítja a felhasználó adatait
POST	/file-upload	A profilkép feltöltéséért felel

Orvos jogosultsággal rendelkező végpontok:

Kérés típusa	URI	LEÍRÁS
POST	/create-appointments	Új időpontot hoz létre orvos számára
GET	/get-all-patients-with-name	Lekéri az összes páciens nevét és egyéb adatait
GET	/get-patients-to-auth-doctor	Lekéri az autentikált orvoshoz tartozó betegeket
GET	/get-appointments-count	Lekéri az időpontok számát páciensenként
PUT	/appointment-delete-by-doctor/{id}	Az orvos törli az adott időpontot
PUT	/appointment-cancel-delete-by-doctor/{id}	Az orvos visszavonja az általa törölt időpontot
GET	/get-today-appointments/{id}	Lekéri a megadott orvos mai időpontjait
GET	/get-patient-data/{id}	Lekéri a megadott beteg részletes adatait
PUT	/finish-appointment/{id}	Az adott időpontot lezárja, befejezettnek jelöli

Admin jogosultsággal rendelkező végpontok:

Kérés típusa	URI	LEÍRÁS
GET	/roles	Kilistázza az összes szerepkört
PUT	/update-nav	Frissíti a navigációs elemek sorrendjét
GET	/get-nav-items-with-roles	Lekéri a navigációs elemeket a szerepkörökkel együtt
GET	/navs	Kilistázza az összes navigációs elemet
POST	/add-nav-to-role	Hozzáad egy navigációs elemet a szerepkörhöz
POST	/check-nav-assigned-to-role	Ellenőrzi, hogy adott szerepkörhöz már van-e rendelve ilyen navigációs elem
DELETE	/remove-nav-from-role/{id}	Eltávolítja a navigációs elemet a szerepkörből
GET	/users	Kilistázza a felhasználókat és azok szerepköreit
PUT	/update-user-role/{id}	Frissíti a felhasználó szerepkörét

### 7.3.1.3. Autentikáció / Hitelesítés

A projektünk során használunk gyárilag beépített végpontokat is a hitelesítéshez, jelszó visszaállításhoz, regisztrációt követő verifikációs email küldéséhez is:

Kérés típusa	URI	LEÍRÁS
POST	/register	Regisztrál egy új felhasználót
POST	/login	Bejelentkezik egy felhasználó
POST	/forgot-password	Elküldi a jelszóhelyreállító linket a megadott email címre
POST	/reset-password	Beállítja az új jelszót a megadott adatokat használva
GET	/verify-email/{id}/{hash}	Ellenőrzi a felhasználó email címét a megadott azonosítóval
POST	/email/verification-notification	Új email megerősítő linket küld a felhasználónak
POST	/logout	Kijelentkezik a felhasználó

### 7.3.1.4. Létrehozott Trigger

Az általunk létrehozott Trigger az időpont létrehozásának ütközéseit vizsgálja. minden esetben az új időpont beszúrása előtt fut le, és amennyiben időpont ütközést talál, nem engedi beszűrni az új rekordot. Ehhez elsőként lekéri a kezelés hosszát percben egy adott 'treatment\_id' alapján, majd kiszámolja az új időpont végét. Ezt követően lekérdezi, hogy az orvosnak van-e másik időpontja, ami ütközik a jelenlegivel (meglévő időpont eleje vagy vége beleesik az új időpontba, illetve a meglévő teljesen lefedi-e az új időpontot).

Ha talál ütközést, hibaüzenetet küld, és megakadályozza a beszúrást.

```
DB:::unprepared"
CREATE TRIGGER check_appointment_overlap
BEFORE INSERT ON doctor_appointments
FOR EACH ROW
BEGIN
    DECLARE treatment_length_minutes INT;
    DECLARE new_end_time DATETIME;

    -- A kezelés hosszának lekérése
    SELECT TIME_TO_SEC(treatment_length) / 60
    INTO treatment_length_minutes
    FROM treatments
    WHERE treatment_id = NEW.treatment_id;

    -- Az új időpont végének kiszámítása
    SET new_end_time = DATE_ADD(NEW.start_time, INTERVAL treatment_length_minutes MINUTE);

    -- Ütközés ellenőrzése
    IF EXISTS (
        SELECT 1 FROM doctor_appointments
        WHERE doctor_id = NEW.doctor_id
        AND (
            (start_time BETWEEN NEW.start_time AND new_end_time)
            OR
            (DATE_ADD(start_time, INTERVAL (
                |   SELECT TIME_TO_SEC(treatment_length) / 60 FROM treatments WHERE treatment_id = doctor_appointments.treatment_id
            ) MINUTE) BETWEEN NEW.start_time AND new_end_time)
            OR
            (start_time < NEW.start_time AND DATE_ADD(start_time, INTERVAL (
                |   SELECT TIME_TO_SEC(treatment_length) / 60 FROM treatments WHERE treatment_id = doctor_appointments.treatment_id
            ) MINUTE) > new_end_time)
        )
    ) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Az orvosnak már van időpontja ebben az időszakban';
    END IF;
END;
");
```

49. ábra - Trigger

### 7.3.1.5. A Triggerhez tartozó időpont létrehozás

A DoctorAppointmentController.php fájlban található metódus végzi az időpontok létrehozását.

Elsőként megtörténik a bemenetek validációja. Ezt követően lekéri a bejelentkezett orvos azonosítóját, majd megkeresi az adatbázisban a megadott nevű kezelést (amennyiben nem találja 404-es hibakóddal 'Kezelés nem található' üzenetet kapunk). Megtörténik a kezdő és záró időpontok Carbon objektumokká alakítása. A kezelések hosszát percekben átszámolja, majd ciklus segítségével időpontokat generál, a végükön 10 perc szünetekkel. A ciklus addig fut, amíg a kezdő időpont + kezelés hossza túl nem lépi a megadott végidőt. Ha az utolsó elem túlnyúlna, eltávolítja a hozzáadandó tömbből. Végül beszúrja az összes időpontot az adatbázisba, és visszaad egy üzenetet 200-as kóddal 'Kezelések sikeresen létrehozva' szöveggel.

```
public function createAppointments(Request $request)
{
    $request->validate([
        'start_time' => 'required|date',
        'end_time' => 'required|date',
        'treatment_name' => 'required|string',
    ]);

    $doctor = Auth::user()->id;

    $treatment = Treatment::where('treatment_name', $request->treatment_name)->first();
    if (!$treatment) {
        return response()->json(['error' => 'Kezelés nem található'], 404);
    }

    $startTime = Carbon::parse($request->start_time);
    $endTime = Carbon::parse($request->end_time);

    $treatmentLength = Carbon::parse($treatment->treatment_length);
    $treatmentLengthMinutes = $treatmentLength->hour * 60 + $treatmentLength->minute;

    $appointments = [];
    $currentStartTime = $startTime;

    while (true) {
        $currentEndTime = $currentStartTime->copy()->addMinutes($treatmentLengthMinutes);

        if ($currentEndTime->greaterThan($endTime)) {
            break;
        }

        $appointments[] = [
            'doctor_id' => $doctor,
            'start_time' => $currentStartTime,
            'treatment_id' => $treatment->treatment_id,
            'patient_id' => null,
            'status' => 'V',
        ];

        $currentStartTime = $currentEndTime->addMinutes(10);
    }

    if ($currentStartTime->greaterThan($endTime)) {
        array_pop($appointments);
    }

    DoctorAppointment::insert($appointments);
    return response()->json(['message' => 'Kezelések sikeresen létrehozva'], 200);
}
```

50. ábra - Trigger Controller

## 7.4. Frontend

A háttérrendszer megvalósítását követően (backend), a következő lépés a felhasználói felület kialakítása, amelyet a frontend testesít meg. Ezen szakaszban az MVC-ből a View réteg kap fő szerepet. Célunk, hogy a backend által létrehozott adatok alapán a felhasználók számára esztétikus és megfelelő formában jelenjenek meg a kívánt adatok. A frontend fejlesztéséhez React JavaScript könyvtárat használtunk, ami biztosította számunkra a dinamikus, komponens alapú felületek létrehozását. A backenddel való megfelelő szinkronizációt az Axios használatával értük el, ami az aszinkron http kérések optimális és hatékony kezelését biztosítja.

React esetén telepített függőségek:

- @hello-pangea/dnd 18.0.1
- Axios 1.8.3
- Bootstrap 5.3.3
- CRA-template 1.2.0
- Date-fns 2.30.0
- Font-awesome 4.7.0
- Leaflet 1.9.4
- Leaflet.awesome-markers 2.0.5
- Moment-timezone 0.5.47
- React-big-calendar 1.18.0
- React-Bootstrap 2.10.9
- React-calendar 5.1.0
- React-leaflet 5.0.0
- React-rating-stars-component 2.2.0
- React-router-dom 7.3.0
- React-slick 0.30.3
- Slick-carousel 1.8.1

### 7.4.1.Axios

Az Axioson keresztül történik a végpontok meghívása, amelyek az Api.php-ban találhatóak.

Létrehoztunk az Axios fájlban egy myAxios változót, amelyben eltároltuk a végpont elérési útját. A későbbiekben ezen myAxios változón keresztül tudjuk elérni a különböző Backend végpontokat.

Axios fájlunk az alábbi útvonalon érhető el: /src/api/Axios.jsx

### 7.4.2.Contextek

A React beépített eszközei közé tartoznak.

A contextek globális állapotot biztosítanak a React alkalmazáson belül, tehát lehetővé teszik, hogy egy adatot a teljes alkalmazáson keresztül elérhetővé tegyük anélkül, hogy paramétereiken keresztül minden szinten tovább kellene őket adni. A végpont hívások nagy része a contextekben kerül megfogalmazásra.

#### 7.4.2.1. Auth Context

Ezen fájlban olyan állapotokat tárolunk, amik az autentikációhoz és a fő adatok backendről való lekérésére szolgál. Célja, hogy biztosítsa a hitelesítéssel, regisztrációval, és bejelentkezett felhasználó adataival kapcsolatos funkciókat.

Elérési útvonala: /src/context/AuthContext.jsx

Elérhető metódusok:

Metódus neve	Végpont elérési útvonal	Leírás
csrf	/sanctum/csrf-cookie	Lekéri a CSRF cookie-t a biztonságos űrlapküldéshez.
getUser	/api/user	Lekéri a bejelentkezett felhasználó alapadatait.
getUserData	/api/user-data	Részletes felhasználói adatokat kér le a felhasználói profilhoz.
login	/login	Bejelentkezteti a felhasználót az e-mail és jelszó alapján.

logout	/logout	Kijelentkezteti a felhasználót a rendszerből.
reg	/register	Regisztrálja az új felhasználót a megadott adatokkal.
fetchNavigation	/api/nav-items	Lekéri a szerepkör alapú navigációs elemeket.
fetchSpecializations	/api/specializations	Lekéri az elérhető orvosi szakterületeket.
fetchAVGDoctorsRatings	/api/get-avg-ratings-by-doctors	Visszaadja az orvosok értékelési átlagait.
fetchEmailStatus	/api/user/email-status	Ellenörzi, hogy a felhasználó emailje hitelesítve van-e.
checkDoctorLicence	/api/checkLicenceByld/{licence}	Lekérdezi, hogy egy adott orvosi licenc érvényes-e.
fetchUploadProfilePicture	/api/file-upload (GET)	Előkészíti a profilkép-feltöltéshez szükséges adatokat.
fetchHandleUpload	/api/file-upload (POST)	Feltölti a kiválasztott profilképet az adatbázisba.
changeUserInfo	/api/change-user-info/{id}	Módosítja a felhasználó adatait a megadott id alapján.

```
const login = async ({ ...adat }) => {
  await csrf();

  try {
    const response = await myAxios.post("/login", adat);
    await getUser();
    if (response.data.success) {
      navigate("/");
    }
  } catch (error) {
    console.log(error);
    if (error.response.status === 422) {
      setErrors(error.response.data.errors);
    }
  }
};
```

51. ábra - Bejelentkezés kód

Jelen képen látható egy bejelentkezés. Lekérjük a CSRF tokent, amely elengedhetetlen a bejelentkezéshez. Meghívjuk myAxios POST-tal a '/login' végpontot, átadva paraméternek az adattípush (autentikációs adatok).

Ennek eredményét elmentjük a response változóba. Meghívjuk késleltetve a getUser metódust, majd ha sikeresen megtörtént a user lekérése. Ha jók voltak a hitelesítő adatok, bejelentkezteti a user-t, és elnavigál a főoldalra. Egyéb esetben egy változóban eltároljuk a hibát.

#### 7.4.2.2. Patient Context

Célja, hogy a páciensekhez kapcsolódó adatokat és műveleteket kezelje. Többek között biztosítja a kezelések és időpontok lekérdezését, a foglalások kezelését (lefoglalás és foglalás törlése), illetve a foglalásokhoz tartozó értékelést.

Elérési útvonala: /src-contexts/PatientContext.jsx

Elérhető metódusok:

Metódus neve	Végpont elérési útvonal	Leírás
fetchAppointments	/api/appointments	Lekéri az összes páciens időpontot, és dátum alapján csoportosítja őket.
fetchAvailableAppointments	/api/get-available-appointments?treatment_id={id}	Visszaadja az elérhető időpontokat egy adott kezelés alapján.
fetchDoctorAppointments	/api/get-appointments-by-doctor?doctor_id={id}	Lekéri az adott orvoshoz tartozó időpontokat.
groupAppointmentsByDate	– (belül segédfüggvény)	Csoportosítja az időpontokat dátum szerint.
fetchTreatmentsBySpecialization	/api/get-treatments-by-specialization	Lekéri a szakterületekhez tartozó kezeléseket.
fetchTreatments	/api/get-treatments	Lekéri az összes kezelés elérhető opciót.
fetchPatientData	– (összetett metódus, több API-hívással)	Elindítja az összes szükséges adat betöltését.
bookingAppointment	/api/booking-appointment/{id}	Lefoglal egy kiválasztott időpontot a páciens részére.
fetchAppointmentRating	/api/profile/rate/{id}	Értékelést küld az adott időponthoz.
fetchCancelAppointmentByPatient	/api/cancel-appointment-by-patient/{id}	Lemondja a páciens által foglalt időpontot.

#### 7.4.2.3. Doctor Context

Ezen Context biztosítja az orvosok számára a betegek és időpontok kezelését,

Elérési útvonala: /src/context/DoctorContext.jsx

Elérhető metódusok:

Metódus neve	Végpont elérési útvonal	Leírás
appointmentDeleteByDoctor	/api/appointment-delete-by-doctor/{id}	Törli az orvos által kezelt időpontot.
appointmentCancelDeleteByDoctor	/api/appointment-cancel-delete-by-doctor/{id}	Visszavonja az időpont törlését.
fetchAllPatients	/api/get-all-patients-with-name	Lekéri az összes beteget név, és egyéb adatok szerint.
fetchAppointmentByPatients	/api/get-appointments-by-patients/{pat_id}	Lekéri a betegekhez tartozó időpontokat.
fetchAppointmentsCount	/api/get-appointments-count	Lekéri az elvégzett időpontok számát páciensenként csoportosítva.
fetchPatientsByAppointments	/api/get-patients-to-auth-doctor	Lekéri a bejelentkezett orvoshoz tartozó pácienseket.
fetchDoctorsWithSpec	/api/doctors-with-spec	Lekéri az orvosokat szakirány szerint.
fetchDoctorData	– (összetett metódus, több API-hívással)	Lekéri a doktor adatait bejelentkezést követően.
fetchAppointmentsByDoctor	/api/get-appointments-by-doctor	Lekéri az orvoshoz tartozó foglalható időpontokat.
fetchAllAppointmentsByDoctor	/api/get-all-appointment-by-doctor	Lekéri az összes időpontot egy orvoshoz.
fetchGetTodayAppointments	/api/get-today-appointments/{id}	Lekéri az orvos számára a mai napra vonatkozó időpontokat.
fetchGetPatientData	/api/get-patient-data/{id}	Lekéri a beteg adatokat.
fetchFinishAppointment	/api/finish-appointment/{id}	Lezársa az időpontot egy leírás hozzáadásával.

#### 7.4.2.4. Admin Context

Ezen Context biztosítja az Admin jogosultsággal rendelkező felhasználók számára a navigációs elemek, felhasználók és szerepkörök kezelését, valamint admin joghoz tartozó alap adatok lekérdezését.

Elérési útvonala: /src-contexts/AdminContext.jsx

Elérhető metódusok:

Metódus neve	Végpont elérési útvonal	Leírás
getUsers	/api/users	Lekéri az összes felhasználó adatát.
fetchNavRoleInfo	/api/get-nav-items-with-roles	Lekéri a navigációs elemeket és azok szerepköreit.
fetchAdminData	/api/roles, /api/navs, /api/users, /api/get-nav-items-with-roles	Lekéri az adminisztrációs adatokat, beleértve a szerepköröket, navigációs elemeket és felhasználókat.
setFilteredUsersList	– (belépő segédfüggvény)	Beállítja a szűrt felhasználók listáját.

#### 7.4.2.5. Contextek összekapcsolása

Az oldalak betöltésére a React useEffect-et használ, amelyet lehet paraméterezni, és ezen paraméter a frissítés gyakoriságát szolgálja.

```
useEffect(() => {
  const storedIsLoggedIn = JSON.parse(sessionStorage.getItem("isLoggedIn"));
  if (storedIsLoggedIn) {
    getUser().then(() => fetchEmailStatus());
  }
}, []);

useEffect(() => {
  if (user && !isVerified) {
    navigate("/verify-email");
  }
}, [isVerified]);

useEffect(() => {
  if (user && user.role_id <= 3 && isVerified) {
    fetchPatientData();
    fetchDoctorsWithSpec();
    fetchNavigation();
    if (user && user.role_id <= 2 && isVerified) {
      fetchDoctorData();
      fetchAppointmentsByDoctor(user.id);
      fetchAllAppointmentsByDoctor(user.id);
      if (user && user.role_id === 1 && isVerified) {
        fetchAdminData();
      }
    }
  } else if (user && !isVerified) {
    navigate("/verify-email");
  } else if (user && user.role_id === 4) {
    navigate("/disabled-user");
  } else {
    fetchNavigation();
    fetchAVGDoctorsRatings();
    fetchDoctorsWithSpec();
  }
}, [user]);

```

52. ábra - useEffect

A harmadik useEffect a jogosultságok vizsgálatára szolgál. Ha van user, és annak szerepköre 3 vagy az alatti, illetve megerősített e-mail címe van, akkor lefutnak a pácienshez tartozó metódusok. Ezt követően hasonló vizsgálattal megnézzük, hogy a szerepkör 2 vagy az alatti, majd az adott orvos szerepkörhöz tartozó metódusok futnak le. Végül ellenőrizzük, hogy a szerepekér értéke 1, és betölt az admin jogosultsághoz tartozó metódus készlet.

Ha van user, de nincs megerősítve, akkor átnavigáljuk az e-mail visszaigazoló oldalra.

Amennyiben van user, de a szerepköre 4, akkor átnavigáljuk az inaktivált felhasználók oldalra. Egyéb esetben töltse be a mindenki számára elérhető funkciókat.

Ez a user változó változásakor fut le minden alkalommal.

Sorrendben haladva a képen látható useEffectek esetén az első kimenti a session-ból, hogy van-e bejelentkezett felhasználó. Amennyiben van, újra lekérjük az adatait. Ez a useEffect egyszer fut le az oldal betöltésekor.

A második arra szolgál, hogy ha van felhasználó, de nincs megerősítve az e-mail, akkor átnavigáljon az e-mail megerősítő oldalra. Ez akkor fut le, ha változik az 'isVerified' változó.

Az alábbi képen látható a Contextjeink hierarchiai felépítése.

```
<BrowserRouter>
  <PatientProvider>
    <DoctorProvider>
      <AdminProvider>
        <AuthProvider>
          <App />
        </AuthProvider>
      </AdminProvider>
    </DoctorProvider>
  </PatientProvider>
</BrowserRouter>
```

53. ábra - Context kapcsolatok

A Browser Router a navigációért felel, és ez helyezkedik el a legfelső szinten. Belsőbb szinteken az adott Contexthez tartozó Providerek találhatóak, amelyek az App irányába biztosítanak adatokat és függvényeket.

Az alábbi képen látható egy konkrét Context hívás a ReservationAppointment komponensben, amely a PatientContext szükséges változóit és metódusait hívja meg.

```
const { fetchAvailableAppointments, treatmentOptions, bookingAppointment } = usePatientContext();
```

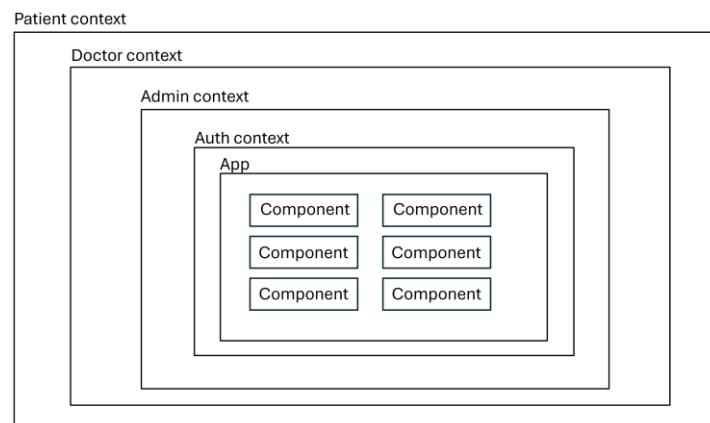
54. ábra - Context használat példa a komponensben

Alább látható a PatientContext-ben található bookingAppointment függvény hívás.

```
const bookingAppointment = async (id) => {
  const response = await myAxios.put(`api/booking-appointment/${id}`)
  setBookingStatusChange(response.data)
}
```

55. ábra - Api végpont hívás

#### 7.4.2.6. Komponensek és contextek felépítése



56. ábra - komponensek és contextek felépítése

### 7.4.3. Routolás és navigáció

A navigáció az adatbázisból kerül betöltésre. Ennek célja, hogy szerepkörönként dinamikusan szabályozhassuk az elérhető menüpontok listáját.

A React Router DOM egy olyan extension, amely lehetővé teszi a React alapú weboldalon az aloldalak kezelését anélkül, hogy az egész oldal újratöltődésére lenne szükség. Segítségével gyorsabb az oldal betöltődése, valamint kevesebb az adatmozgás. Ezen keresztül lehetséges többek között a navigáció létrehozása.

```
<Routes>
  <Route path="/" element={<VendegLayout />} />
  <Route path="/disabled-user" element={<DisabledUser />} />
  <Route path="/verify-email/*" element={<VerifyEmailPage />} />
  <Route path="/forgot-password" element={<ForgotPassword />} />
  <Route path="/password-reset/*" element={<ResetPassword />} />
  <Route path="/selected-treatment" element={<SelectedTreatment />} />
  <Route path="/profile" element={<Profile />} />
  <Route index element={<Fooldal />} />
  {navigation ? (
    navigation.map((e, index) => {
      const Component = ComponentsMap[e.component_name];

      if (!Component) {
        console.error(`Component ${e.component_name} not found.`);
        return null;
      }

      return (
        <Route
          key={index}
          path={urls[index]}
          element={<Component />}
        />
      );
    })
    : (
      <h1>Loading</h1>
    )
  )}
</Route>
</Routes>
```

57. ábra - Routolás

A képen látható VendegLayout kezeli a gyermek komponenseket. A route-ok között az 'index' a Föoldal töltődik be alapértelmezetten. Néhányat manuálisan adtunk meg, a többi pedig backend oldalról töltünk be. Ehhez segítségül szükséges volt egy ComponentsMap.jsx fájl létrehozása, melyben egy lista található a komponensek neveivel. Ezeket a neveket használjuk fel a dinamikus route feltöltésekhez.

```

<nav className="navbar navbar-expand-sm fixed-top">
  <div className="container-fluid">
    <ul className="navbar-nav" ref=[navRef]>
      {navigation ? (
        navigation.map((item, i) => {
          const isLogout = item.url === "/logout";
          return (
            <li
              key={i}
              onClick={isLogout ? logout : undefined}
              ref={isLogout ? logoutRef : undefined}>
              <Link className="nav-link" to={item.url}>
                {item.name}
              </Link>
            </li>
          );
        })
      : (
        <li className="navbar-item">
          <span className="nav-link">Loading...</span>
        </li>
      )
    </ul>
  </div>
</nav>

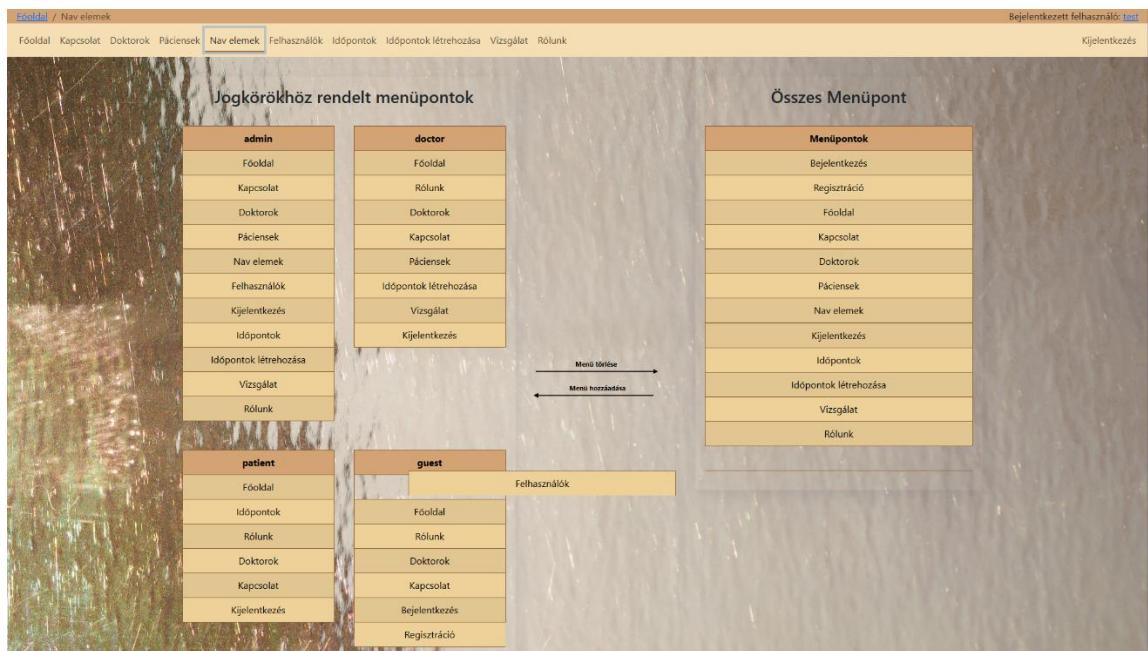
```

58. ábra - Navigáció kód

En a képen látható, hogy a backendtől kapott listán megyünk végig, majd egyesével betöljük az egyesével kapott navigációs elemeket.

Az applikációnkban funkcióként beépítettük a menüpontok dinamikus beállíthatóságát a különböző szerepkörökhez. Ez megába foglalja a menüpontok hozzáadását és eltávolítását. A végleges állapotban ezt drag 'n drop segítségével érjük el, amelynek irányában határozza meg, hogy a menüpont hozzáadódik vagy törlődik az adott szerepkör esetén.

A kód komplexitása és mérete miatt nem helyeztünk el képet a kódról, azonban a felületen látható megjelenés az alábbi képen látható:



59. ábra - Nav lista módosítás

#### 7.4.3.1. Időpont létrehozás

A komponens biztosítja az orvos számára az időpont létrehozás lehetőségét, amelyhez az alábbi metódusokat használjuk.

Metódus neve	Leírás
handleViewChange(view)	Beállítja a naptár aktuális nézetét (hónap, hét, nap, stb.). A setView(view) segítségével a nézet frissül a felhasználó választása alapján.
handleNavigate(date)	Frissít az aktuálisan megjelenített dátumot a naptárban. Ez lehetővé teszi az előre-hátra navigálást.
handleDateClick(value)	Beállítja a kiválasztott dátumot és megjeleníti az időpontfelvételi űrlapot. A dátum sv-SE (ÉÉÉÉ:HH:NN) formátumban kerül eltárolásra.
createStartTimes()	7:00 és 14:00 között enged 30 percenként időpontot létrehozni. Ezeket az időpontfoglaláshoz használja az időválasztó.
handleSubmit()	Összeállítja és elküldi az új időpontokat a backend API-nak. A sikeres mentés után frissíti az orvos időpontjait.
createEndTimes(startTime)	A kezdési idő alapján kiszámolja az elérhető zárási időpontokat 2 órás ráhagyással. Biztosítja, hogy a vége ne nyúljon túl 18:00 időponton.
handleStartTimeChange(event)	Frissíti a kiválasztott kezdési időt, és automatikusan új végidőt generál. Ez biztosítja a konzisztens időintervallumokat.
handleChange(event)	Frissíti a kezelés nevét (treatment) a beviteli mező változásakor. Ez az adat is elküldésre kerül az időpont létrehozásakor.
getTreatmentsByDoctor()	A páciens foglalásokból kiszűri az adott orvos által végzett kezelések nevét. Csak az aktuális orvoshoz tartozó kezelések kerülnek visszaadásra.
generateEvents(appointments)	Átalakítja az időpontlistát naptári eseményekké, amelyek tartalmazzák a kezdési-végi időt és a páciens nevét. Ezek jelennek meg vizuálisan a naptárban.

#### 7.4.3.2. Időpont keresés és foglalás

A páciens kezelés alapán, vagy konkrét doktorra keresve tud időpontokra keresni és a preferált időpontot lefoglalni.

#### 7.4.3.3. Kezelés alapján

Metódus neve	Leírás
handleTreatmentChange(event)	Frissíti a kiválasztott kezelés nevét és betölti az elérhető időpontokat a backendból a fetchAvailableAppointments segítségével, amint a felhasználó új opciót választ.
handleBooking(appointmentId)	Elküldi a foglalási kérést a context bookingAppointment metódusával, majd a navigate('/profile') hívással átirányítja a felhasználót a profiloldalra. Ez biztosítja, hogy a felhasználó azonnal lássa a lefoglalt időpont részleteit.

##### 7.4.3.3.1. Doktor alapján

Metódus neve	Leírás
handleShow()	Megnyitja a Modal ablakot az aktuálisan kiválasztott időpont részleteinek megjelenítéséhez. A setShow(true) állapotfrissítés indítja el a megjelenítést.
handleClose()	Bezárja a Modal ablakot az időpont részleteinek megjelenítése után. Ez a setShow(false) hívással gondoskodik arról, hogy a felhasználó visszatérjen a naptár nézetéhez.
toggleAppointments(doctorId, treatmentId)	A megadott orvos és kezelés kombinációjához tartozó időpontok listáját nyitja meg vagy zárja be. Ha ugyanazt a párost kattintják újra, visszaállítja a szűrési állapotot, különben lekéri az orvos időpontjait a backendről.
generateEvents(appointments)	Az appointments listából kiszűri a toggleAppointments által beállított orvos–kezelés pároshoz tartozó elemeket. Ezeket átalakítja naptári eseményekké, megadva a kezdési és zárási időpontot, valamint a címként a páciens nevét.
handleViewChange(view)	Módosítja a naptár nézetét (month, week, day, stb.) a felhasználó választása alapján. A setView(view) hívás gondoskodik arról, hogy a naptár az új nézetet jelenítse meg.
handleNavigate(date)	Frissíti az aktuális dátumot a naptár nézetében, amikor a páciens lapoz a hónapok vagy hetek között, így a setCurrentDate(date) beállítja, hogy melyik időszakot lássuk.
handleEventClick(event)	Amikor a naptár eseményére kattintanak, megtalálja a appointmentsDoctor listában az esemény objektumát az id

	alapján. Ezután beállítja a selectedAppointment állapotot és megnyitja a Modal-t (handleShow()).
bookAppointmentClick(e)	Elindítja az időpontfoglalást a bookingAppointment(e.id) hívással, majd átirányítja a felhasználót a profiloldalra (navigate('/profile')). Így a felhasználó azonnal láthatja a lefoglalt időpont részleteit.

#### 7.4.3.4. Profil és lefoglalt időpontok

Ebben a komponensben helyezkednek el a felhasználó saját adatai, valamint páciensek esetén az általuk korábban lefoglalt időpontok listája.

##### 7.4.3.4.1. Profilért felelős metódusok listája

Metódus neve	Leírás
handleFileChange(event)	Profilkép feltöltéséhez használható. A kiválasztott képfájlt (event.target.files[0]) eltárolja, majd előnézeti képet generál belőle a FileReader segítségével, amit a setPreview állapotkezelő frissít.
useEffect	Akkor fut le, ha a userData változik. Feltölti a form adatait (formData) a felhasználó adataival, és lekéri a hozzá tartozó időpontokat az orvos szemszögéből a fetchAppointmentByPatients segítségével.
handleChange(e)	A formData állapotot frissíti, amikor a felhasználó módosítja a mezők tartalmát. minden beviteli mező a saját name attribútuma alapján frissül.
handleEdit()	Átvált szerkesztési módba. Az isEditing állapotot true-ra állítja, ezzel engedélyezve bizonyos form mezők módosítását.
handleSave()	A szerkesztett adatokat elmenti: meghívja a changeUserInfo(userId, formData) függvényt, majd kilép a szerkesztési módból (isEditing = false).
handleUploadPicture()	Feltölти a kiválasztott profilképet a szerverre. Ha isUploadSelected = true, akkor végrehajtja a fetchHandleUpload() hívást, majd frissíti a felhasználói adatokat (getUserData()), végül váltogatja az isUploadSelected értékét.

#### 7.4.3.4.2. Pofilban található foglalt időpontok listájának metódusai

Metódus neve	Leírás
handleShow()	Megnyitja a Modal ablakot, azaz a show állapotot true-ra állítja.
handleClose()	Bezárja a Modal ablakot (pl. lemondás megerősítő popup), azaz a show állapotot false-ra állítja.
handleRatingChange(appointmentId, newRating)	Egy adott időpont értékelését frissíti. Először módosítja a ratings állapotot, majd meghívja a fetchAppointmentRating() függvényt az új értékkel, végül frissíti a rating értékét.
handleClickCancelAppointmentByPatient()	Egy időpontot lemond a páciens. Meghívja a fetchCancelAppointmentByPatient() metódust az időpont azonosítójával, majd frissíti a páciens időpontlistáját, végül bezárja a modalt (handleClose()).

#### 7.4.3.5. Orvoshoz tartozó adott vizsgálat időpontja

Az adott orvoshoz tartozó aktuális napra szóló időpontokat listázza. A lista adott elemét kiválasztva megjelennek az időponthoz tartozó releváns információk.

Metódus neve	Leírás
useEffect	Az oldal betöltésekor fut le. A DoctorContextból érkező metódust futtatja, amely lekéri az adott orvos aznapi időpontjait.
ListGroupTreatments	Funkcionális komponens, amely a props objektumból kapott adatokat jeleníti meg. Az alapvető funkciója, hogy kezelje a kezeléseket és navigáljon a kiválasztott kezelésre.
handleOnClick	A handleOnClick metódus a kattintás eseményét kezeli, és a navigate függvény segítségével átirányít a "/selected-treatment" útvonalra, átadva a kiválasztott kezelés adatait. Ezzel az interakcióval biztosítja, hogy a felhasználó a megfelelő kezelés oldalára kerüljön.

#### 7.4.3.6. Az aktuális vizsgálat

A folyamatban lévő kezelés lezárásához szükséges komponens, ahol leírást lehet adni a kezelés eredményéről.

Metódus neve	Leírás
useEffect	Akkor fut le, ha a data.p_id értéke változik, és meghívja a fetchGetPatientData függvényt az adott páciens adatainak lekérésére. Így biztosítja, hogy a megfelelő páciens adatai jelenjenek meg.
handleSubmit	Ez a metódus a kezelési űrlap elküldéséért felel, valamint megakadályozza az alapértelmezett form viselkedést, majd meghívja a fetchFinishAppointment függvényt, és visszanavigál a kezelés listájához. A felhasználó így egy sikeres művelet után azonnal a kezelések áttekintő oldalára jut.
handleChange	Ez a metódus frissíti a description állapotot az input mező értéke alapján. Ezzel lehetővé teszi, hogy a felhasználó szabad szöveget írjon a kezeléshez kapcsolódóan.

## 7.5. Reszponzivitás

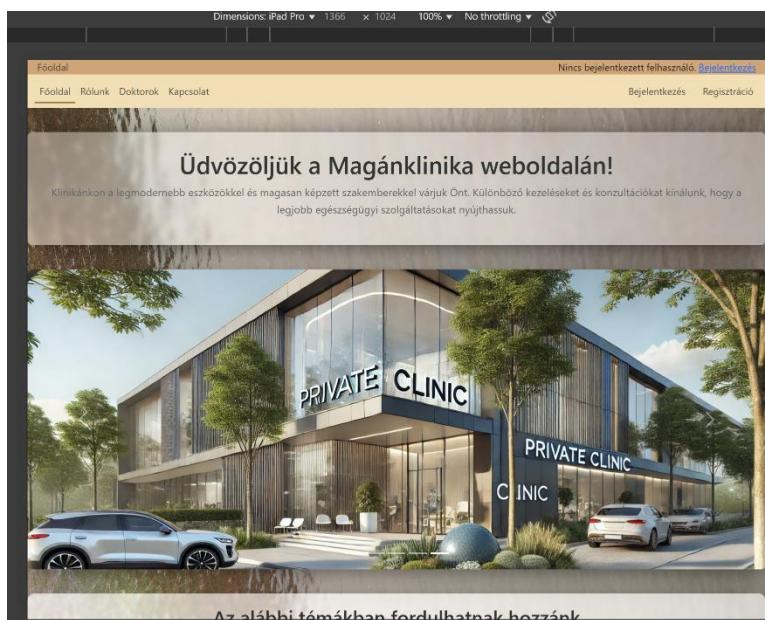
A stílusok kialakításához Bootstrap keretrendszer használtuk, ami segített a reszponzív és gyorsan kialakítható kinézet létrehozásában.

Az alábbi képeken látható az oldal a különböző képernyőméretek használatával.

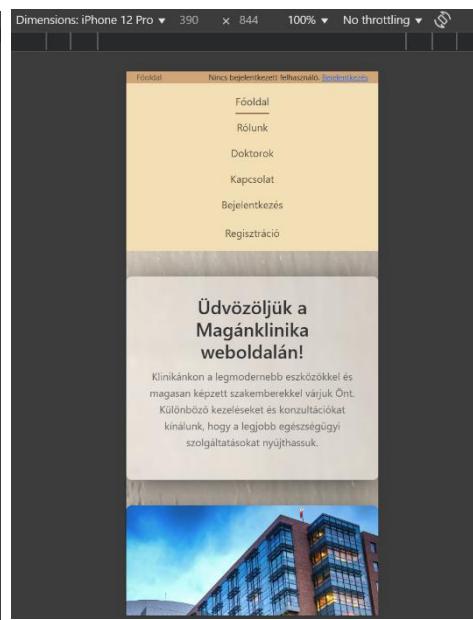
### 7.5.1. Főoldal

A főoldal esetén 768px-nél van töréspont.

iPad pro (fekvő) illetve iPhone 12 pro (álló) képernyőményet esetén.



61. ábra - Reszponzivitás főoldal ipad pro

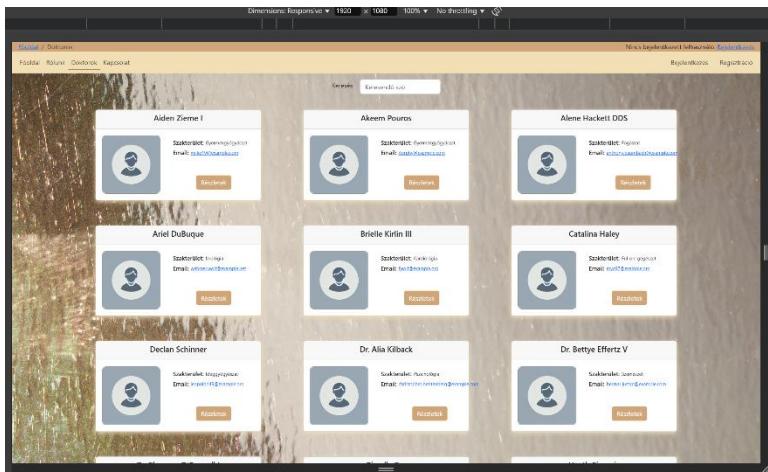


60. ábra - Reszponzivitás főoldal iphone 12 pro

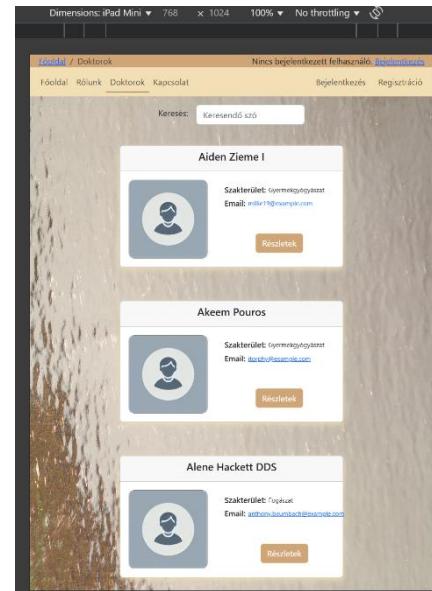
## 7.5.2. Doktorok

Ezen az oldalon 375px-re állítottunk be töréspontot.

FullHD monitoron, illetve iPad Mini (álló) esetén



63. ábra - Reszponzivitás Doktorok FullHD monitor



62. ábra - Reszponzivitás Doktorok ipad mini

## 7.5.3. Időpont foglalás

FullHD monitoron, illetve iPhone 12 Pro (álló) esetén

The screenshot shows a responsive web page for booking appointments. At the top, there's a navigation bar with links for 'Főoldal', 'Rölkö', 'Doktorok', 'Kapcsolat', and 'Bejelentkezés'. Below the navigation, there's a search bar with placeholder text 'Időpont keresése időpont alapján' and a button 'Keresés'. The main content area shows a table titled 'Foglalási Időpontok' with two columns: 'Időpont' and 'Foglalások'. The 'Időpont' column lists various times from 2025-04-21 13:00:00 to 2025-04-21 15:40:00. The 'Foglalások' column contains green rectangular buttons, each labeled 'Foglalás' and corresponding to a specific time slot. The background of the page features a blurred image of a forest.

64. ábra - Reszponzivitás Időpont foglalás FullHD monitor

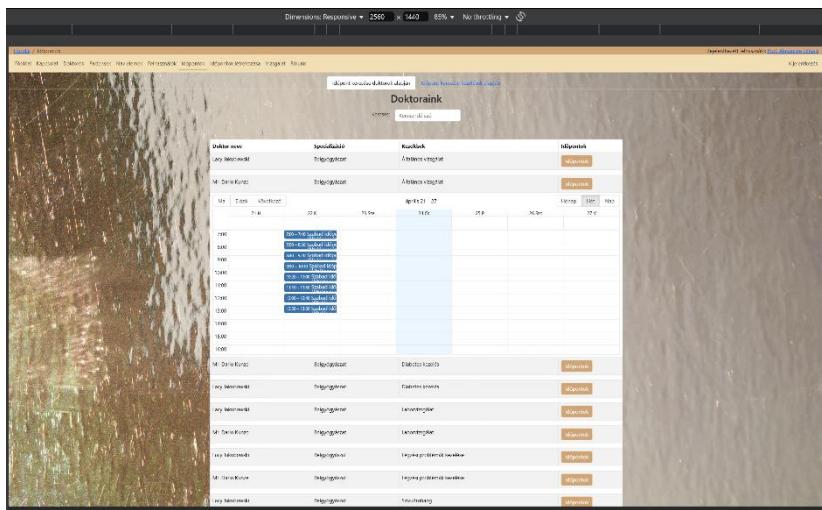
This screenshot shows the same appointment booking page as above, but viewed on an iPhone 12 Pro with a smaller screen. It zooms in on the list of time slots. The visible part of the page includes the header 'Időpontok', the search bar, and the 'Foglalási Időpontok' section. The table shows the same list of time slots and green 'Foglalás' buttons. The background is a blurred forest image.

65. ábra - Reszponzivitás Időpont foglalás iphone 12 pro

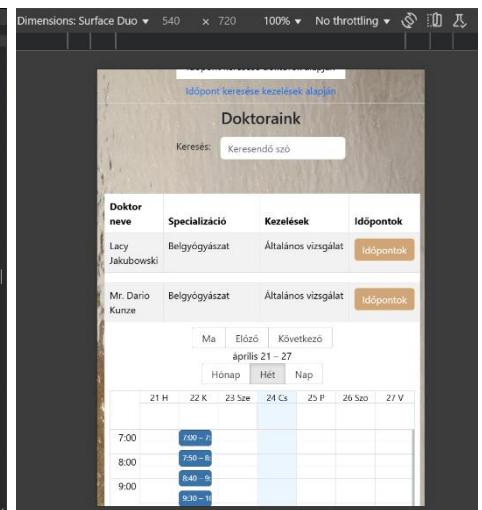
71

## 7.5.4.Időpont foglalás – naptár

QHD monitoron, illetve iPhone 12 Pro (álló) esetén



67. ábra - Reszponzivitás Időpont foglalás naptár QHD monitor



66. ábra - Reszponzivitás Időpont foglalás naptár iphone 12 pro

Ezen oldalakon túl globálisan 575 px-nél törik meg az oldal.

A navigációs menü 768px-nél, illetve 575px-nél törik meg.

A list group 650px-nél törik meg.

A jelszó visszaállítási oldal 650px-nél törik meg.

Továbbá a bootstrap alapértelmezett töréspontjai szabályozzák a weboldal töréseit.

## 8. Tesztelés

### 8.1. Backend tesztelés

Unit teszt eredmények az alábbi mappaszerkezetben érhetőek el:

<https://github.com/Maganklinika/maganklinika/tree/359abe659a897915c590788a1774e34f7b2f6a2b/maganklinika/backend/tests>

Laravel keretrendszert használtunk a Unit tesztekhez elvégzéséhez. A tesztelés során a teszt nevek megadásakor törekedtünk az átláthatóságra, és a végpontok beazonosíthatóságára.

```
ps E:\programming\maganklinika\maganklinika\backend> php artisan test
PASS Tests\Unit\ExampleTest
✓ that true is true

PASS Tests\Feature\Auth\AuthenticationTest
✓ users can authenticate using the login screen
✓ users can not authenticate with invalid password
✓ users can logout

PASS Tests\Feature\Auth\EmailVerificationTest
✓ email can be verified
✓ email is not verified with invalid hash

PASS Tests\Feature\Auth>PasswordResetTest
✓ reset password link can be requested
✓ password can be reset with valid token

PASS Tests\Feature\Auth\RegistrationTest
✓ new users can register

PASS Tests\Feature\DoctorAppointmentTest
✓ get available appointments by treatment
✓ get appointment by doctor
✓ get all appointment by doctor
✓ create appointments
✓ get appointments count
✓ get appointments by patients
✓ booking appointment
✓ fail booking appointment
✓ get today appointments
✓ finish appointment

PASS Tests\Feature\DoctorLicenceTest
✓ doctor can register with unused licence
✓ doctor can register with used licence

PASS Tests\Feature\DoctorRatingsViewTest
✓ get average ratings by doctors

PASS Tests\Feature\DoctorTest
✓ list doctors with specialization

PASS Tests\Feature\ExampleTest
✓ the application returns a successful response
```

68. ábra - Tesztelés Backend 1

```
PASS Tests\Feature\NavigationTest
✓ example
✓ update nav
✓ get nav items with roles
✓ get navigation
✓ add nav to role
✓ check nav assigned to role
✓ remove nav from role

PASS Tests\Feature\PatientTest
✓ get all patients with name
✓ get patients to auth doctor
✓ get patient data
✓ cancel appointment by patient
✓ fail to cancel appointment by patient

PASS Tests\Feature\RoleTest
✓ example

PASS Tests\Feature\SpecializationTest
✓ get specializations
✓ get treatment by spec

PASS Tests\Feature\TreatmentTest
✓ get treatments

PASS Tests\Feature\UserTest
✓ users
✓ users auth
✓ update user role
✓ user data
✓ change user info

Tests: 45 passed (62 assertions)
Duration: 129.52s
```

69. ábra - Tesztelés Backend 2

A képeken látható, hogy a lefuttatott tesztek sikeresek voltak, amelyek lefedik az útvonalak nagyrészét.

## 8.2. Frontend tesztelés

Frontend esetén manuális tesztelést végeztünk.

Bejelentkezés		
Teszteset	Elvárt eredmény	Kapott eredmény
Bejelentkezés helyes adatokkal	Az oldal tovább irányít a bejelentkezett felhasználó jogosultságához tartozó főoldalra, valamint betölti a megfelelő menülistát	Az elvárt eredménnyel megegyezik
Bejelentkezés hibás e-mail címmel	A bejelentkezés nem történik meg, 'Hibás e-mail cím vagy jelszó.' szöveget kapjuk.	Az elvárt eredménnyel megegyezik
Bejelentkezés hibás jelszóval	A bejelentkezés nem történik meg, 'Hibás e-mail cím vagy jelszó.' szöveget kapjuk.	Az elvárt eredménnyel megegyezik
Bejelentkezási kísérlet sok próbálkozással	A bejelentkezés nem történik meg. 'Túl sok bejelentkezási próbálkozás. Kérlek próbáld újra XX másodperc múlva.' szöveget kapjuk	Az elvárt eredménnyel megegyezik
Efelejtett jelszó kérése helyes e-mail címmel	A jelszó-visszaállító linket elküldtük üzenetet kapjuk, valamint kiküldésre került a visszaállító e-mail	Az elvárt eredménnyel megegyezik
Efelejtett jelszó kérése hibás e-mail címmel	Hiba történt: Nem található felhasználó ezzel az e-mail címmel' üzenetet kapjuk, e-mail nem kerül kiküldésre.	Az elvárt eredménnyel megegyezik
Elfelejtett jelszó úra kérése 1 percen belül 3 próbálkozás után	Hiba történt: Újra kiküldés csak 1 perc múlva lehetséges.' szöveget kapjuk.	Az elvárt eredménnyel megegyezik

Regisztráció		
Teszteset	Elvárt eredmény	Kapott eredmény
Páciens esetén a megfelelő adatok megadását követően regisztrálás	Sikeress regisztráció, visszaigazoló e-mail küldése	Az elvárt eredménnyel megegyezik
Doktor esetén a megfelelő adatok megadását követően regisztrálás	Sikeress regisztráció, visszaigazoló e-mail küldése	Az elvárt eredménnyel megegyezik
Hibás telefonszám megadása (betűk) regisztráció során	A telefonszám csak számokat, illetve +, -, / karaktereket tartalmazhat.' üzenet megjelenése és sikertelen regisztráció.	Az elvárt eredménnyel megegyezik
Nem megfelelő hosszúságú karakter megadás TAJ szám esetén regisztráció során	A TAJ szám kizárolag 9 számjegyű lehet.' üzenet megjelenése és sikertelen regisztráció.	Az elvárt eredménnyel megegyezik
Jövőbeni születési dátum megadása regisztráció során	Csak mai napnál korábbi dátum adható meg.' üzenet megjelenése és sikertelen regisztráció.	Az elvárt eredménnyel megegyezik
Doktor esetén nem létező licensz megadása	Nincs ilyen licensz a rendszerben' üzenet megjelenése és sikertelen regisztráció.	Az elvárt eredménnyel megegyezik

<b>Időpontfoglalás</b>		
Teszteset	Elvárt eredmény	Kapott eredmény
Időpont foglalás menüpont megnyitása	Menüpont betölt, megjelenik az Időpont keresése kezelések alapján és Időpont keresése doktorok alapján lapfül	Az elvárt eredménnyel megegyezik
Időpont keresése kezelések alapján fülre kattintás	Legördülő lista betölt, benne az elérhető kezelésekkel	Az elvárt eredménnyel megegyezik
Legördülő listában kezelés kiválasztása	Adott kezeléshez tartozó szabad időpontok jelennek meg az összes érvényes orvos esetén	Az elvárt eredménnyel megegyezik
Foglalok gombra kattintás egy időpontnál	Felugró ablak jelenik meg a kezelés nevével, kezdés időpontjával, és foglalási információval, igen és mégse gombokkal.	Az elvárt eredménnyel megegyezik
Mégse gombra kattintás	Nem történik foglalás, a felugró ablak bezáródik	Az elvárt eredménnyel megegyezik
Igen gombra kattintás	Megtörténik a foglalás, az oldal átirányít a profilok lapra	Az elvárt eredménnyel megegyezik
Időpont keresése doktorok alapján fülre kattintás	Doktorok listája megjelenik a megfelelő specializációval, kezelésekkel és Időpontok gombbal	Az elvárt eredménnyel megegyezik
Keresés szövegdoboznál orvos nevének részletére keresés	Kilistázza az orvosokat, akik nevében szerepel a beírt betűkombináció	Az elvárt eredménnyel megegyezik
Időpontok gombra kattintás időponttal rendelkező Orvos esetén	Megjelenik a naptár, az adott orvoshoz tartozó foglalható időpontokkal.	Az elvárt eredménnyel megegyezik
Időpontok gombra kattintás időponttal nem rendelkező Orvos esetén	Ha nincs szabad időpont, akkor 'Nincs elérhető időpont' szöveg jelenik meg.	Az elvárt eredménnyel megegyezik
Naptár esetén napi nézetben előző/következő gombra kattintás	Előző gombra az előző napra, következő gombra kattintva a következő napra ugrik a naptár.	Az elvárt eredménnyel megegyezik
Naptár esetén heti nézetben előző/következő gombra kattintás	Előző gombra az előző hétre, következő gombra kattintva a következő hétre ugrik a naptár.	Az elvárt eredménnyel megegyezik
Naptár esetén havi nézetben előző/következő gombra kattintás	Előző gombra az előző hónapra, következő gombra kattintva a következő hónapra ugrik a naptár.	Az elvárt eredménnyel megegyezik
Ma gombra kattintás	A nézetnek megfelelően aktuális napra / hétre / hónapra ugrik a naptár	Az elvárt eredménnyel megegyezik
Szabad időpontra kattintás	Felugró ablakban megjelenik az időpont neve, kezdési ideje, hossza és ára, valamint igen és mégse gomb.	Az elvárt eredménnyel megegyezik
Mégse gombra kattintás	A felugró ablak bezáródik, időpont nem foglalódik le.	Az elvárt eredménnyel megegyezik
Igen gombra kattintás	A felugró ablak bezáródik, az oldal átnavigál a Profilra és lefoglalódik az időpont.	Az elvárt eredménnyel megegyezik
Korábbi időpontra foglalás	Korábbi időpont nem jelenik meg foglalhatóként.	Az elvárt eredménnyel megegyezik

## 9. Fejlesztési lehetőségek

A felület jelenlegi állapotában rendelkezik az orvosok és páciensek számára szükséges fő funkciókkal, azonban a rendszer további bővítésére az alábbi ötleteink vannak:

### 9.1. DÁP integráció

A felületen a későbbi gördülékenyebb azonosításra és belépésre, valamint az orvosi dokumentumok kezelésére a DÁP biztosítana lehetőséget.

### 9.2. Fizetési lehetőségek

Jelenleg is tároljuk a kezelések árait, amelyekhez különböző fizetési lehetőségek bevezetésével közvetlenül rendezni tudják a páciensek a kezelések díját.

### 9.3. Hozzászólások

A jelenlegi értékelési rendszer mellett szabadszöveges hozzászólások segítségével még inkább nyomon követhetővé válik a lezajlott kezelések minősége.

### 9.4. Statisztikák

Első sorban adminisztrátorok, de később orvosok és páciensek számára is hasznos lehet különböző statisztikai adatok, diagramok megjelenítése a számukra hasznos információkkal.

### 9.5. Hírfolyam

Hasznos hírek létrehozása, kezelése az adminisztrátorok számára, amely az orvosok és páciensek számára szolgálnak információval.

## 9.6. Live chat

Páciensek kérdései esetén gyorsan távolról tudnak segíteni az ügyintézők a megoldandó problémával kapcsolatban.

## 10. Irodalomjegyzék

W3Schools	<a href="https://www.w3schools.com/">https://www.w3schools.com/</a>
Laravel dokumentáció	<a href="https://laravel.com/docs/11.x">https://laravel.com/docs/11.x</a>
Axios dokumentáció	<a href="https://axios-http.com/docs/intro">https://axios-http.com/docs/intro</a>
React Bootstrap	<a href="https://react-bootstrap.netlify.app/">https://react-bootstrap.netlify.app/</a>
Bootstrap	<a href="https://getbootstrap.com/docs/5.3/getting-started/introduction/">https://getbootstrap.com/docs/5.3/getting-started/introduction/</a>
GitHub	<a href="https://docs.github.com/en">https://docs.github.com/en</a>
HelloPangea	<a href="https://dnd.hellopangea.com/">https://dnd.hellopangea.com/</a>
BigCalendar	<a href="https://jquense.github.io/react-big-calendar/examples/?path=/story/about-big-calendar--page">https://jquense.github.io/react-big-calendar/examples/?path=/story/about-big-calendar--page</a>
StackOverflow	<a href="https://stackoverflow.com/questions">https://stackoverflow.com/questions</a>
NPM	<a href="https://docs.npmjs.com/getting-started">https://docs.npmjs.com/getting-started</a>
NodeJS	<a href="https://nodejs.org/docs/latest/api/">https://nodejs.org/docs/latest/api/</a>
React	<a href="https://react.dev/reference/react">https://react.dev/reference/react</a>
ChatGPT	<a href="https://chatgpt.com/">https://chatgpt.com/</a>
React Leaflet	<a href="https://react-leaflet.js.org/docs/start-introduction/">https://react-leaflet.js.org/docs/start-introduction/</a>
React Rating Stars	<a href="https://www.npmjs.com/package/react-simple-star-rating">https://www.npmjs.com/package/react-simple-star-rating</a>
Unsplash	<a href="https://unsplash.com/">https://unsplash.com/</a>
Coolors	<a href="https://coolors.co/">https://coolors.co/</a>

# 11. Mellékletek

## 11.1. Program struktúra

- .env
- .gitignore
- .gitlab-ci.yml
- docker-compose.yml
- mysql.cnf
- package-lock.json
- tree.txt
  
- backend/
  - .dockerignore
  - .editorconfig
  - .env
  - .env.example
  - .gitattributes
  - .gitignore
  - .phpunit.result.cache
  - artisan
  - composer.json
  - composer.lock
  - Dockerfile
  - phpunit.xml
  - README.md
  
- app/
  - Http/
    - Controllers/
      - Controller.php
      - DoctorAppointmentController.php
      - DoctorController.php
      - DoctorLicenceController.php
      - DoctorRatingsViewController.php
      - FileController.php
      - GetTreatmentBySpecializationController.php
      - NavigationController.php
      - NavigationRoleController.php
      - PatientController.php
      - RoleController.php
      - SpecializationController.php
      - TreatmentController.php
      - UserController.php
    - LAuth/
      - AuthenticatedSessionController.php
      - EmailVerificationNotificationController.php
      - NewPasswordController.php
      - PasswordResetLinkController.php
      - RegisteredUserController.php
      - VerifyEmailController.php
    - Middleware/
      - Admin.php
      - Doctor.php

- EnsureEmailsVerified.php
- Patient.php
- Mail/
  - AppointmentStatusUpdated.php
- Models/
  - Doctor.php
  - DoctorAppointment.php
  - DoctorLicence.php
  - GetTreatmentBySpecialization.php
  - Navigation.php
  - NavigationRole.php
  - Patient.php
  - Role.php
  - Specialization.php
  - Treatment.php
  - User.php
- database/
  - .gitignore
  - factories/
    - DoctorAppointmentFactory.php
    - DoctorFactory.php
    - DoctorLicenceFactory.php
    - NavigationFactory.php
    - NavigationRoleFactory.php
    - PatientFactory.php
    - RoleFactory.php
    - SpecializationFactory.php
    - TreatmentFactory.php
    - UserFactory.php
  - migrations/
    - 0000\_02\_08\_090408\_create\_roles\_table.php
    - 0001\_01\_01\_000000\_create\_users\_table.php
    - 0001\_01\_01\_000001\_create\_cache\_table.php
    - 0001\_01\_01\_000002\_create\_jobs\_table.php
    - 2025\_02\_08\_090242\_create\_personal\_access\_tokens\_table.php
    - 2025\_02\_08\_090657\_create\_navigations\_table.php
    - 2025\_02\_08\_090723\_create\_navigation\_roles\_table.php
    - 2025\_02\_08\_090735\_create\_patients\_table.php
    - 2025\_02\_08\_090759\_create\_specializations\_table.php
    - 2025\_02\_08\_090812\_create\_doctors\_table.php
    - 2025\_02\_08\_090822\_create\_treatments\_table.php
    - 2025\_02\_08\_090846\_create\_doctor\_appointments\_table.php
    - 2025\_02\_14\_111542\_get\_treatment\_by\_specialization.php
    - 2025\_02\_19\_102106\_doctors\_ratings\_view.php
    - 2025\_02\_21\_111958\_add\_appointment\_overlap\_trigger.php
    - 2025\_03\_08\_073623\_create\_doctor\_licences\_table.php
  - seeders/
    - DatabaseSeeder.php
    - DoctorAppointmentSeeder.php
    - DoctorLicenceSeeder.php
    - DoctorSeeder.php
    - NavigationRoleSeeder.php
    - NavigationSeeder.php
    - PatientSeeder.php
    - RoleSeeder.php
    - SpecializationSeeder.php

- TreatmentSeeder.php
- public/
  - .htaccess
  - favicon.ico
  - index.php
  - robots.txt
- storage/
  - profile\_pictures/
    - 1743430056\_ben-sweet-2LowviVHZ-E-unsplash.jpg
    - 1743430347\_ben-sweet-2LowviVHZ-E-unsplash.jpg
    - 1743526946\_1.jpg
    - 1743526975\_1.jpg
    - 1743527130\_1.jpg
- resources/
  - lang/
    - hu/
      - auth.php
      - passwords.php
      - validation.php
  - views/
    - .gitkeep
    - appointment-status-updated.blade.php
    - fileUpload.blade.php
- routes/
  - api.php
  - auth.php
  - console.php
  - web.php
- tests/
  - TestCase.php
  - TestCaseWithSeed.php
  - Feature/
    - DoctorAppointmentTest.php
    - DoctorLicenceTest.php
    - DoctorRatingsViewTest.php
    - DoctorTest.php
    - ExampleTest.php
    - NavigationTest.php
    - PatientTest.php
    - RoleTest.php
    - SpecializationTest.php
    - TreatmentTest.php
    - UserTest.php
  - LAuth/
    - AuthenticationTest.php
    - EmailVerificationTest.php
    - PasswordResetTest.php
    - RegistrationTest.php
  - Unit/
    - ExampleTest.php
- frontend/

- .gitignore
- default.conf
- Dockerfile
- package-lock.json
- package.json
- README.md
  
- public/
  - favicon.ico
  - index.html
  - logo192.png
  - logo512.png
  - manifest.json
  - robots.txt
  
- images/
  - clinic1.jpg
  - clinic2.jpg
  - clinic3.jpg
  - clinic4.jpg
  - clinic5.jpg
  
- backgrounds/
  - aboutUs\_bg.jpg
  - rolunk\_bg.jpg
  
- profile\_picture/
  - default\_profil\_picture.webp
  
- src/
  - App.css
  - App.js
  - App.test.js
  - index.css
  - index.js
  - logo.svg
  - reportWebVitals.js
  - setupTests.js
  - variables.css
  
- api/
  - Axios.jsx
  
- components/
  - appointments/
    - DoctorAppointments.jsx
    - ReservationAppointments.css
    - ReservationAppointments.jsx
  - componentsmap/
    - ComponentsMap.jsx
  
- doktorok/
  - doctors.css
  - Doktorok.jsx
  - doctors\_with\_spec/
    - DoctorsWithSpec.jsx
  
- filters/
  - text\_filter\_top/

- textfilter.css
  - TextFilter.jsx
  - listgroup/
  - ListGroupRow.jsx
  - nav/
  - nav.css
  - navElemek.css
  - NavElemek.jsx
  - Navigacio.jsx
  - paciensek/
  - Paciensek.jsx
  - PatientsRow.jsx
  - userlist.css
  - tables/
  - roletables.css
  - TableRow.jsx
  - Tables.jsx
  - UsersTableRow.jsx
  - userdivlist/
  - ListHeader.jsx
  - UserDivList.jsx
- 
- contexts/
  - AdminContext.jsx
  - AuthContext.jsx
  - DoctorContext.jsx
  - PatientsContext.jsx
- 
- layouts/
  - VendegLayout.jsx
  - footer/
  - footer.css
  - Footer.jsx
  - infobar/
  - infobar.css
  - InfoBar.jsx
- 
- pages/
  - Appointments.jsx
  - Bejelentkezes.jsx
  - Fooldal.css
  - Fooldal.jsx
  - ForgotPassword.jsx
  - Kijelentkezes.jsx
  - regisztracio.css
  - Regisztracio.jsx
- 
- contact/
  - contact.css
  - Kapcsolat.jsx
- 
- create\_appointment/
  - appointment.css
  - CreateAppointmentsByDoctor.jsx
- 
- disabledUser/
  - disabledUser.css

- DisabledUser.jsx
- profiles/
  - PatientAppointmentTableRow.jsx
  - profile.css
  - Profile.jsx
- reset\_password/
  - resetPassword.css
  - ResetPassword.jsx
- rolunk/
  - rolunk.css
  - Rolunk.jsx
- treatment/
  - listgroup.css
  - ListGroupTreatments.jsx
  - SelectedTreatment.jsx
  - Treatment.jsx
- users/
  - Felhasznalok.jsx
  - users.css
- verify\_email/
  - verifyEmail.css
  - VerifyEmailPage.jsx